

موسسه بابان

انتشارات بابان و انتشارات راهیان ارشد

درس و کنکور ارشد

پایگاه داده‌ها

(حل تشریحی سوالات کنکور ارشد ۱۳۷۰ تا ۱۴۰۱)

ویژه‌ی داوطلبان کنکور کارشناسی ارشد مهندسی کامپیوتر و IT

براساس کتب مرجع

راما کریشنان، آبراهام سیلبرشاتز و رامز المصری

ارسطو خلیلی فر

تقدیم به:

تمامی کسانی که برای پیشرفت و سعادت خود و بشریت

تلاش می‌کنند.

ارسطو خلیلی فر

فهرست مطالب

۹.....	تست‌های فصل اول: مفاهیم اولیه
۱۵.....	پاسخ تست‌های فصل اول: مفاهیم اولیه.....
۲۵.....	تست‌های فصل دوم: مدل نهاد و رابطه.....
۲۶.....	پاسخ تست‌های فصل دوم: مدل نهاد و رابطه.....
۳۱.....	تست‌های فصل سوم: مدل رابطه‌ای.....
۴۱.....	پاسخ تست‌های فصل سوم: مدل رابطه‌ای.....
۱۱۳.....	تست‌های فصل چهارم: جبر رابطه‌ای.....
۱۲۷.....	پاسخ تست‌های فصل چهارم: جبر رابطه‌ای.....
۱۹۳.....	تست‌های فصل پنجم: حساب رابطه‌ای.....
۱۹۶.....	پاسخ تست‌های فصل پنجم: حساب رابطه‌ای.....
۲۰۵.....	تست‌های فصل ششم: SQL دستورات DML.....
۲۳۶.....	پاسخ تست‌های فصل ششم: SQL دستورات DML.....
۳۹۹.....	تست‌های فصل هفتم: SQL دستورات DDL و DCL.....
۴۰۸.....	پاسخ تست‌های فصل هفتم: SQL دستورات DDL و DCL.....
۴۵۱.....	تست‌های فصل هشتم: وابستگی تابعی.....
۴۵۸.....	پاسخ تست‌های فصل هشتم: وابستگی تابعی.....
۵۰۵.....	تست‌های فصل نهم: نرمال‌سازی.....
۵۱۶.....	پاسخ تست‌های فصل نهم: نرمال‌سازی.....
۵۸۹.....	تست‌های فصل دهم: شاخص‌گذاری اطلاعات.....
۵۹۰.....	پاسخ تست‌های فصل دهم: شاخص‌گذاری اطلاعات.....

تست‌های فصل اول: مفاهیم اولیه

- ۱- مزایای یک DATA BASE نسبت به فایل‌های متعارف چیست؟ (کارت‌شناسی ارشد-آزاد ۷۶ و ۷۱)
- دستیابی مشترک (SHARED ACCESS) به داده‌ها
 - اطمینان از صحت داده‌ها (DATA VALIDATION) کم‌تر مورد نیاز است.
 - ۱ و ۴
 - کنترل حساب شده مقدار تکرار داده‌ها در DATA BASE

- ۲- کدام یک از موارد زیر جزء وظایف DBA نمی‌باشد؟ (کارت‌شناسی ارشد-آزاد ۷۲)
- نوشتن Data Dictionary برای پایگاه داده‌ها
 - نظارت بر عملکرد پایگاه داده‌ها (Performance Monitoring)
 - تهیه رویه‌ها و استراتژی تهیه Back up و نحوه احیای پایگاه داده‌ها
 - تهیه Schema برای پایگاه داده‌ها

- ۳- در یک سیستم مدیریت پایگاه داده‌ها (DBMS) کدام یک از امکانات زیر جزء عناصر اصلی تشکیل‌دهنده DMBS محسوب نمی‌شوند؟ (کارت‌شناسی ارشد-دولتی ۷۲)
- امکان پردازش زبان طبیعی برای کار با پایگاه
 - امکان کار با داده‌ها به کمک یک DSL (Data Sub Language)
 - امکان تأمین جامعیت و بی‌نقصی (integrity) پایگاه
 - امکان تأمین ایمنی پایگاه

- ۴- دو مرحله از مراحل طراحی یک بانک اطلاعاتی عبارتند از طراحی ادراکی (Conceptual Design) و طراحی منطقی (Logical Design) این دو چه تفاوت اساسی با هم دارند؟ (کارت‌شناسی ارشد-دولتی ۷۵)
- طراحی ادراکی به مدل خاصی مربوط می‌شود و پس از انتخاب مدل صورت می‌گیرد ولی طراحی منطقی به مدل خاصی بستگی ندارد.
 - طراحی ادراکی مکمل طراحی منطقی است و پس از آن انجام می‌گیرد.
 - طراحی منطقی به صورت کلی به سیستم می‌نگرد و با روش‌هایی مانند ER انجام می‌گیرد.
 - طراحی منطقی به مدل خاصی مربوط می‌شود و پس از انتخاب مدل صورت می‌گیرد ولی طراحی ادراکی به مدل خاصی بستگی ندارد.

- ۵- کدام یک از گزاره‌های زیر در مورد کاتالوگ (System Catalog) در بانک اطلاعاتی صحیح است؟ (مهندس کامپیوتر-دولتی ۷۴)
- کاتالوگ فقط با احکام DML قابل تغییر است.
 - کاتالوگ فقط توسط برخی از احکام DDL قابل تغییر است.
 - کاتالوگ فقط با احکام DML و برخی از احکام DDL قابل تغییر است.
 - کاتالوگ فقط با احکام DDL قابل تغییر است.

۶- استقلال منطقی در یک پایگاه داده رابطه‌ای شامل چه مورد (یا مواردی) نیست؟ (کارشناسی ارشد- آزاد ۷۱)

(۱) تعریف یک رابطه جدید در SCHEMA

(۲) حذف یک رابطه از SCHEMA

(۳) جایگزینی یک رابطه با دو یا چند رابطه کوچک‌تر معادل

(۴) ۱ و ۳

۷- در مورد کاتالوگ سیستم کدام یک از گزاره‌های زیر همواره صحیح است؟ (کارشناسی ارشد- دولتی ۷۳)

(۱) استفاده از کاتالوگ از مزایای سیستم مدیریت بانک اطلاعاتی رابطه‌ای بر سیستم مدیریت بانک اطلاعاتی شبکه‌ای است.

(۲) استفاده از کاتالوگ باعث کاهش سرعت ذخیره اطلاعات می‌شود.

(۳) استفاده از کاتالوگ باعث افزایش استقلال از داده (Data Independence) می‌شود.

(۴) استفاده از کاتالوگ باعث افزایش سرعت بازیابی اطلاعات می‌شود.

۸- در صورتی که زمان خواندن کل رکوردهای یک جدول به ترتیب کلید اصلی ملاک ارزیابی باشد، DMBS باید کدام یک از ساختارهای ذیل را در سطح داخلی پایگاه داده مورد استفاده قرار دهد؟ (مهندسی IT- دولتی ۸۳)

(۱) Indexed- Sequential file

(۲) Hash file

(۳) Multi Indexed file

(۴) Pile file

۹- استقلال داده فیزیکی دارای کدام یک از مزایای زیر است؟ (مهندسی کامپیوتر- دولتی ۸۵)

(۱) سهولت بیان پرس و جوها

(۲) سرعت پاسخ به پرس و جوها

(۳) کاهش حجم بانک اطلاعاتی

(۴) سهولت تغییر مدل دیسک‌های بانک اطلاعاتی

۱۰- استفاده از انواع مختلف قفل‌ها در کنترل همزمانی دارای کدام یک از مزایای زیر است؟ (کارشناسی ارشد- دولتی ۸۲)

(۱) کاهش حافظه اصلی مورد نیاز

(۲) سهولت بهینه‌سازی پرس و جو

(۳) افزایش احتمال بن‌بست تراکنش‌ها

(۴) افزایش سرعت عملیات روی بانک اطلاعات

۱۱- کدام یک از ویژگی‌های ذیل از خصوصیات دید ادراکی (Conceptual View) می‌باشد؟ (مهندسی IT- آزاد ۸۴)

(۱) دید طراح از داده‌های ذخیره شده در بانک است.

(۲) به صورت انتزاعی می‌باشد.

(۳) نشأت گرفته از نمودار ER می‌باشد.

(۴) هر سه مورد

۱۲- اگر اطلاعات مربوط به پایگاه داده یعنی مشخصات کاربران، مجوزهای دسترسی آن‌ها به پایگاه داده، عملیات انجام شده و ... نگهداری شود به آن، چه می‌گویند؟ (مهندسی IT- آزاد ۸۶)

(۱) فرهنگ داده‌ها

(۲) کاتالوگ سیستم

(۳) زبان تعریف داده‌ها DDL

(۴) زبان زیر داده DSL

۱۳- در یک نظام بانک اطلاعات کارآمد کدام گزینه ممکن است رخ دهد؟ (در حالت کار عادی سیستم - بدون رخ دادن هیچ اشکالی)

(مهندسی IT - آزاد ۸۷)

- ۱) برخی از دستورات تراکنش اجرا گردند و برخی دیگر خیر.
- ۲) مدتی پس از اتمام تراکنش و commit شدن آن، DBMS آن را Undo می‌کند.
- ۳) در اجرای همزمان تراکنش‌ها، تراکنشی باعث غلط شدن نتیجه تراکنش دیگر شود.
- ۴) درمیان اجرای تراکنش برخی قوانین جامعیتی به طور موقت نقض گردند.

۱۴- کدام مورد زیر می‌تواند از معایب سیستم بانک اطلاعاتی نسبت به سیستم پرونده‌ای باشد؟

(مهندسی IT - دولتی ۸۸)

- ۱) کاهش همزمانی عملیات
- ۲) کاهش امنیت اطلاعات
- ۳) افزایش زمان اجرای برنامه‌های کاربردی
- ۴) افزایش تکرار اطلاعات (Redundancy)

۱۵- کدام یک از گزینه‌های زیر نادرست است؟

(مهندسی IT - آزاد ۸۸)

- ۱) یک تراکنش نمی‌تواند قوانین جامعیتی را نقض کند.
- ۲) در یک پایگاه داده، تراکنش‌ها همزمان می‌توانند به داده‌های مورد نظر خود آزادانه دسترسی داشته باشند و کنترل خاصی در این زمینه لازم نیست.
- ۳) یا همه دستورات یک تراکنش با موفقیت اجرا می‌شود یا هیچ یک اجرا نمی‌شود.
- ۴) نتیجه یک تراکنش پس از تکمیل و اتمام آن نباید در اثر حوادث غیرمترقبه از بین برود.

۱۶- کدام گزینه مفهوم ناسازگاری داده‌ای (Data Inconsistency) را به درستی بیان می‌کند؟

(مهندسی IT - آزاد ۸۸)

- ۱) مقدار داده صفت با نوع آن متفاوت باشد.
- ۲) داده‌های پایگاه داده با اندیس (Index) آن سازگار نباشد.
- ۳) داده‌های ذخیره شده در مورد یک چیز در قسمت‌های مختلف پایگاه داده با یکدیگر متفاوت باشند.
- ۴) مقدار داده صفت با مقدار ذخیره شده در اندیس (Index) ساخته شده روی آن صفت یکسان نباشد.

۱۷- کدام مورد مزیت سیستم پرونده‌ای بر سیستم بانک اطلاعاتی است؟

(مهندسی IT - دولتی ۸۹)

- ۱) سهولت دسترسی همزمان به اطلاعات
- ۲) امنیت فیزیکی اطلاعات، بیش‌تر
- ۳) سازگاری اطلاعات، بیش‌تر
- ۴) سهولت پاسخ به سوالات پیش‌بینی نشده

۱۸- کدام مورد باعث عدم برقراری استقلال داده منطقی است؟

(مهندسی IT - دولتی ۸۹)

- ۱) تغییر ساختمان داده‌ها در سطح منطقی
- ۲) تغییر محتوی داده‌ها در سطح منطقی
- ۳) تغییر دیسک حاوی بانک اطلاعاتی
- ۴) تغییر محتوی داده‌ها در سطح فیزیکی

۱۹- یکپارچگی (Integration) باعث کدام مزیت در سیستم بانک اطلاعات نسبت به سیستم پرونده‌ای است؟

(مهندسی IT - دولتی ۸۹)

- ۱) سرعت بیش‌تر در دسترسی به اطلاعات
- ۲) سهولت کار برای کاربر نهایی
- ۳) امنیت بیش‌تر اطلاعات
- ۴) کاهش تکرار اطلاعات (Redundancy)

- ۲۰- کدام یک جزء وظایف DBMS نمی باشد؟
 (مهندسی کامپیوتر- آزاد ۸۹ گروه الف)
 (۱) حفظ جامعیت و امنیت داده‌ها
 (۲) به روز رسانی کاتالوگ سیستم
 (۳) نظارت بر کارایی و پاسخ به تغییر نیازها
 (۴) ترمیم و حفظ سازگاری داده‌ها
-
- ۲۱- اطلاعات ذخیره شده در کاتالوگ سیستم شامل کدام گزینه نیست؟
 (مهندسی کامپیوتر- آزاد ۸۹ گروه الف)
 (۱) مقادیر داده‌های درون جداول
 (۲) نام موجودیت‌ها و ارتباطات بین آن‌ها
 (۳) نام ساختارهای داده‌ای
 (۴) نام صفات خاصه هر موجودیت
-
- ۲۲- کدام مورد از ویژگی‌های معماری توزیع شده در پایگاه داده‌ها نیست؟ (مهندسی کامپیوتر- آزاد ۸۹ گروه ب)
 (۱) داده‌ها به بخش‌هایی تقسیم و در سایت‌ها توزیع شده‌اند.
 (۲) سیستم چنان عمل می‌کند که از نظر کاربران شبیه به یک پایگاه داده متمرکز به نظر می‌رسد.
 (۳) داده‌های ذخیره شده در هر سایت تحت کنترل یک سیستم مدیریت پایگاه داده‌ها هستند.
 (۴) سیستم بانک اطلاعاتی به صورت یک ساختار دو بخشی در نظر گرفته می‌شود.
-
- ۲۳- به چه دلیل معماری سه سطحی ANSI در پایگاه داده اهمیت دارد؟ (مهندسی کامپیوتر- آزاد ۸۹ گروه ب)
 (۱) معماری ANSI یک معماری مرجع برای تمام DBMS ها، فروشندگان و توسعه‌دهندگان آن‌ها به وجود آورده است.
 (۲) معماری ANSI استقلال داده‌ای فیزیکی و منطقی را تا حد امکان تضمین می‌نماید.
 (۳) معماری ANSI یک روش مشترک برای کاربران در مشخص کردن استفاده از داده‌ها، مفاهیم داده‌ای و ذخیره‌سازی آن‌ها فراهم نموده است.
 (۴) معماری ANSI توسط مؤسسه استاندارد ANSI تعریف شده است.
-
- ۲۴- کدام یک از پایگاه داده‌های ذیل با موارد دیگر متفاوت است؟ (مهندسی کامپیوتر- آزاد ۸۹ گروه ب)
 Desktop (۱) Enterprise (۲) Workgroup (۳) Multi-User (۴)
-
- ۲۵- برای تضمین صحت و جامعیت تراکنش‌ها کدام یک از موارد زیر لازم است؟ (مهندسی IT- آزاد ۸۹)
 (۱) یکپارچگی (Atomicity)
 (۲) هم‌خوانی (Consistency)
 (۳) پایداری (Durability)
 (۴) همه موارد
-
- ۲۶- کدام یک از موارد زیر از مزایای استفاده از کاتالوگ سیستم می‌باشد؟ (مهندسی IT- آزاد ۸۹)
 (۱) استقلال داده‌ای (Data Independence)
 (۲) امنیت (Security)
 (۳) جامعیت (Integrity)
 (۴) حفظ سازگاری داده‌ها (Consistency)
-
- ۲۷- ساختن شاخص (Index) باعث ایجاد تغییر در کدام یک می‌شود؟ (مهندسی IT- آزاد ۸۹)
 (۱) رابطه (Relation)
 (۲) کاتالوگ سیستم (System Cataloge)
 (۳) شما (Schema)
 (۴) دید (View)

۲۸- ساختن دید (View) باعث ایجاد تغییر در کدام یک می‌شود؟ (مهندسی IT- آزاد ۹۰)

- (۱) رابطه (Relation) (۲) کاتالوگ سیستم (System Cataloge)
(۳) شما (Schema) (۴) دید (View)

۲۹- کدام یک از موارد زیر مزیت سیستم بانک اطلاعاتی شی‌گرا بر سیستم بانک اطلاعاتی رابطه‌ای محسوب می‌شود؟ (مهندسی کامپیوتر- دولتی ۹۱)

- (۱) افزایش هم‌زمانی عملیات (۲) اشتراک داده‌ها و دیدها بین کاربران
(۳) کاهش حجم بانک اطلاعات (۴) اشتراک عملیات جدید بین کاربران

۳۰- کدام مورد شامل جزئی از اجزاء معماری پایگاه داده، طبق معماری ANSI نیست؟ (مهندسی کامپیوتر- آزاد ۹۱)

- (۱) کاربر، زبان داده‌ای (۲) شبکه و سیستم عامل
(۳) سیستم عامل و مدیر پایگاه داده (۴) مدیر پایگاه داده و زبان داده‌ای

۳۱- کدام یک از اطلاعات زیر در مورد لغت‌نامه داده‌ها (Data Dictionary) پایگاه داده درست است؟ (مهندسی کامپیوتر- آزاد ۹۱)

- (۱) در لغت‌نامه داده‌ها اطلاعاتی در مورد جداول و دیدها (view) نگهداری می‌شود.
(۲) در لغت‌نامه داده‌ها اطلاعات مربوط به سطح دسترسی و مالک داده نگهداری می‌شود.
(۳) در لغت‌نامه داده‌ها اطلاعات مربوط به آمار استفاده از داده‌ها ذخیره می‌شود.
(۴) هر سه مورد صحیح است.

۳۲- کدام یک از عبارات زیر صحیح است؟ (مهندسی کامپیوتر- آزاد ۹۲)

- الف) هر شما (Schema) متشکل از تعدادی کاتالوگ است.
ب) هر شما متشکل از تعدادی Data Object است.
ج) هر کاتالوگ متشکل از تعدادی شما است.
د) هر Data Object می‌تواند در چند شما باشد.
(۱) هر ۴ مورد درست است.
(۲) ب، ج و د
(۳) الف، ب و د
(د) ب و ج

۳۳- مدیریت ترمیم (Recovery management) در DBMS، عهده‌دار انجام کدام خاصیت زیر می‌باشد؟ (مهندسی کامپیوتر- آزاد ۹۳)

- (۱) Atomicity (۲) Consistency (۳) Isolation (۴) Durability

۳۴- کدام گزینه نادرست است؟ (مهندسی کامپیوتر- آزاد ۹۳)

- (۱) Commit توسط کاربر صادر گردیده و DBMS نمی‌تواند آن را لغو نماید.
(۲) Commit توسط کاربر صادر گردیده و DBMS می‌تواند آن را به تعویق بیندازد.
(۳) Commit در یک تراکنش می‌تواند توسط کاربر صادر گردد، لیکن DBMS عهده‌دار تأیید نهایی آن می‌باشد.

۴) یک تراکنش دارای دستورالعمل‌های مختلفی می‌باشد که Commit و Abort نیز جزء آن محسوب می‌گردد.

۳۵- کدام یک از گزینه‌های زیر جزء مزایای استقلال فیزیکی داده‌ها محسوب می‌گردد؟

(مهندسی کامپیوتر- آزاد ۹۳)

- ۱) کاهش زمان پاسخ پرس و جوها.
- ۲) کاهش پیچیدگی در بیان پرس و جوها.
- ۳) کاهش حجم پایگاه داده‌ها.
- ۴) سهولت در تغییرات مدل دیسک‌های پایگاه داده‌ها.

۳۶- در کدام یک از دستورات تراکنش، DBMS یک شناسه منحصر به فرد به آن تراکنش نسبت می‌دهد؟

(مهندسی کامپیوتر- آزاد ۹۳)

- start (۱) abort و commit (۲) فقط commit (۳) input و output (۴)

۳۷- کنترل همروندی (Concurrency Control) به عنوان واحدی در DBMS عهده‌دار رسیدگی به کدام خاصیت است؟

(مهندسی IT- آزاد ۹۳)

- Atomicity (۱) Isolation (۲) Consistency (۳) Durability (۴)

۳۸- مصونیت دید کاربران در قبال تغییراتی که در سطح ادراکی پدید می‌آید بر کدام گزینه تأکید می‌نماید؟

(مهندسی IT - آزاد ۹۳)

- ۱) ترمیم داده‌ها ۲) سازگاری
۳) استقلال داده‌ای منطقی ۴) استقلال داده‌ای فیزیکی

۳۹- تراکنش نهایی نشده Uncommit تراکنشی است که:

(مهندسی کامپیوتر- آزاد ۹۱)

- ۱) ساقط (abort) شده است. ۲) یا فعال (active) باشد یا ساقط شده باشد.
۳) فعال نیست. ۴) دستور Start آن صادر نشده است.

۴۰- دید طراح بانک از داده‌های ذخیره شده در بانک به کدام لایه در معماری پیشنهادی ANSI باز می‌گردد؟

(مهندسی IT - آزاد ۹۳)

- ۱) دید خارجی ۲) دید ادراکی
۳) دید داخلی ۴) دید ادراکی و دید خارجی

پاسخ تست‌های فصل اول: مفاهیم اولیه

۱- گزینه (۳) صحیح است.

در پایگاه داده نسبت به سیستم فایل معمولی (فایل‌های متعارف یا سیستم پرونده‌ای) افزونگی داده‌ها بسیار کم‌تر است و کنترل آن امکان‌پذیر است، از طرف دیگر پایگاه داده امکان به اشتراک گذاشتن داده‌ها را فراهم می‌آورد، هم‌چنین در پایگاه داده‌ها اطمینان از صحت داده‌ها بسیار مهم است.

۲- گزینه (۱) صحیح است.

ایجاد و بروز رسانی خودکار Data Dictionary بر عهده DBMS است. منظور از تهیه Schema برای پایگاه داده، همان طراحی ساختار کلی پایگاه داده است.

۳- گزینه (۱) صحیح است.

پردازش زبان‌های طبیعی (رویه‌ای یا میزان) مانند زبان C و پاسکال بر عهده کامپایلر و پردازش زبان‌های بیانی (میهمان) مانند SQL بر عهده DBMS (SQL Server) می‌باشد.

۴- گزینه (۴) صحیح است.

در طراحی ادراکی یا لایه ادراکی عامل (فاز تحلیل)، مدل مورد استفاده می‌تواند ساخت یافته (مانند ER) یا شی‌گرا (مانند UML) باشد. بنابراین طراحی ادراکی به مدل خاصی بستگی ندارد. اما طراحی منطقی یا لایه ادراکی خاص (فاز طراحی) به مدل به کار رفته در طراحی ادراکی بستگی دارد. یعنی اگر مدل‌های فاز تحلیل ساخت یافته بود، فازهای طراحی و پیاده‌سازی نیز با رویکرد ساخت یافته پیش خواهد رفت و اگر مدل‌های فاز تحلیل شی‌گرا بود، فازهای طراحی و پیاده‌سازی نیز با رویکرد شی‌گرا پیش خواهد رفت.

۵- گزینه (۴) صحیح است.

کاتالوگ حاوی اطلاعات سیستمی پایگاه داده از قبیل مشخصات جداول، شاخص‌ها، دیدها و سطوح دسترسی کاربران است. از آن‌جا که تمامی دستورات DDL، با ساختار پایگاه داده سر و کار دارند، حال ایجاد یک ساختار و یا حذف ساختار، بنابراین در هر شرایطی دستورات DDL موجب تغییر در کاتالوگ خواهند شد. پس گزینه دوم نادرست، هم‌چنین دستورات DML، با محتوای پایگاه داده سر و کار دارند، برخی دستورات مانند Insert اندازه جداول را تغییر می‌دهد، بنابراین موجب تغییر در کاتالوگ می‌شوند، اما برخی از دستورات مانند select مثلاً فقط منجر به خواندن از پایگاه داده می‌شود بدون آنکه محتوای پایگاه داده تغییر کند، بنابراین بهتر است بگوییم برخی دستورات DML موجب تغییر در کاتالوگ خواهند شد و برخی دیگر خیر. بنابراین گزینه اول و سوم نیز نادرست و گزینه چهارم درست است.

۶- گزینه (۲) صحیح است.

حذف یک رابطه از شما جزء استقلال منطقی نیست زیرا با حذف یک رابطه کاربران و برنامه‌های کاربری دیگر به اطلاعات آن دسترسی ندارند. اما تعریف رابطه جدید، جایگزینی یک جدول با چند جدول دیگر و پخش داده‌های موجود در یک جدول در جداول جدید، جزء استقلال منطقی است.

۷- گزینه (۳) صحیح است.

کاتالوگ سیستم در تمام مدل‌های بانک اطلاعاتی مانند رابطه‌ای و شبکه‌ای کاربرد دارد. در واقع بحث کاتالوگ پایگاه داده مستقل از بحث پایگاه داده است. کاتالوگ سیستم پس از هر ذخیره و بازیابی اطلاعات مربوط به بانک اطلاعات بروزسانی می‌گردد. پس کاتالوگ سیستم دخالتی در افزایش یا کاهش سرعت ذخیره و بازیابی اطلاعات نخواهد داشت. نگهداری مشخصات سیستمی بانک اطلاعات در کاتالوگ سیستم و در فضایی جداگانه و مستقل از محتوای بانک اطلاعات باعث افزایش استقلال داده‌های محیط عملیاتی از محتویات کاتالوگ سیستم می‌گردد.

۸- گزینه (۱) صحیح است.

زمان خواندن کل فایل در فایل‌های Pile، Hash و Multi Indexed به دلیل نامرتب بودن رکوردهای فایل بسیار بالا است. تنها فایل‌های Indexed Sequential هستند که فایل اصلی بر اساس کلید اصلی ترتیبی بوده، بنابراین زمان خواندن کل آن، نسبت به فایل‌های دیگر کم‌تر است.

۹- گزینه (۴) صحیح است.

استقلال داده‌ها به دو صورت فیزیکی و منطقی بیان می‌شود. در صورت وجود استقلال فیزیکی داده‌ها می‌توان مدل دیسک‌های بانک اطلاعات را مستقل از دید منطقی (برنامه‌های کاربردی) تغییر داد. به بیان دیگر اگر تغییری در ذخیره‌سازی داده‌ها انجام گیرد (مثلاً نوع دیسک عوض شود) برنامه‌های کاربردی هیچ تغییری نمی‌کنند. در استقلال منطقی داده تغییر تصویر ادراکی بانک از دید کاربر و برنامه‌های آن‌ها مخفی می‌ماند. البته تغییری که منجر به حذف جدول گردد، استقلال منطقی داده را نقض می‌کند.

۱۰- گزینه (۴) صحیح است.

وجود قفل‌ها سبب می‌شود که در اجرای هم‌روند چند تراکنش، یک داده به صورت هم‌زمان توسط دو تراکنش مورد استفاده قرار نگیرد. (برقراری شرط انحصار متقابل) این کار سرعت عملیات را افزایش می‌دهد زیرا داده اشتراکی به صورت انحصاری فقط در اختیار یک تراکنش است. البته تراکنش‌های دیگر که به داده اشتراکی کاری ندارند می‌توانند از سایر بخش‌های بانک اطلاعات بهره ببرند. یکی دیگر از معایب استفاده از قفل‌ها نیز، احتمال به وجود آمدن بن‌بست در اجرای تراکنش‌ها است. زیرا قفل‌گذاری شرط لازم برای وقوع بن‌بست است که همان انحصار متقابل است را برقرار می‌کند، که در ادامه ممکن است هر سه شرط دیگر یعنی انحصاری بودن، نگهداری و انتظار و سیکل چرخشی به عنوان شرط کافی نیز برقرار گردند که منجر به وقوع بن‌بست می‌گردد. بخشی از حافظه اصلی نیز برای نگهداری اطلاعات قفل‌ها به کار رفته می‌شود که منجر به سربار هزینه حافظه نیز خواهد شد. بهینه‌سازی پرس و جوها در مواردی همچون سرعت و مصرف حافظه بستگی به نحوه نوشتن دستورها دارد.

۱۱- گزینه (۲) صحیح است.

در دید ادراکی یا همان دید ادراکی عام به صورت کاملاً انتزاعی کل نیازمندی‌ها و موجودیت‌ها مستقل از پیاده‌سازی شناسایی و توسط مدل ER مدل‌سازی می‌شود. خروجی دید ادراکی عام به عنوان ورودی دید ادراکی خاص تلقی می‌گردد. به بیان دیگر دید ادراکی خاص

نشأت گرفته از نمودار ER می‌باشد. دید طراح از داده‌های ذخیره شده در بانک اطلاعات مربوط به دید ادراکی خاص می‌باشد که توسط مدل رابطه‌ای بیان می‌گردد.

۱۲- گزینه (۲) صحیح است.

کاتالوگ سیستم و فرهنگ داده‌ها در کتب دیت و سیلبرشاتز به یک معنای یکسان بیان شده‌اند. اما از نگاه برخی کتب دیگر اطلاعات مربوط به مشخصات کاربران، حق دستیابی افراد به داده‌ها، تاریخ ایجاد و بروز در آوردن داده‌ها و اندازه جداول در کاتالوگ سیستم نگهداری می‌شود. در آن کتب لغت‌نامه داده‌ها زیر مجموعه کاتالوگ سیستم در نظر گرفته شده است. اما کاتالوگ سیستم به دلیل کاربرد ویژه‌ای که دارد مجزا شده است.

۱۳- گزینه (۴) صحیح است.

در یک نظام بانک اطلاعات کارآمد چهار کنترل یکپارچگی، هم‌خوانی، انزوا و پایایی روی تمامی تراکنش‌ها اعمال می‌گردد تا جامعیت (صحت و سازگاری) آن تضمین شود. گزینه اول: قاعده جامعیت یکپارچگی (atomicity) مانع می‌شود. گزینه دوم: قاعده جامعیت پایایی (durability) مانع می‌شود. گزینه سوم: قاعده انزوا (isolation) مانع می‌شود. گزینه چهارم: در حین اجرای یک تراکنش ممکن است جامعیت (سازگاری) به طور موقت نقض شود ولی در انتهای کار تراکنش باید جامعیت (سازگاری) برقرار باشد.

۱۴- گزینه (۳) صحیح است.

گزینه اول: در سیستم‌های پایگاه داده همزمانی عملیات، به دلیل مجتمع بودن و اشتراکی شدن داده‌ها، افزایش یافته است. گزینه دوم: در سیستم‌های پایگاه داده امنیت اطلاعات به دلیل تعریف سطوح دسترسی کاربران و مکانیزم‌های پشتیبان‌گیری افزایش یافته است. گزینه سوم: در سیستم‌های پایگاه داده زمان اجرای برنامه‌های کاربردی به دلیل کنترل مسائل امنیتی حقوق دستیابی (عبور از گیت امنیتی) افزایش یافته است، در واقع سرعت اجرا کاهش می‌یابد. گزینه چهارم: در سیستم‌های پایگاه داده به دلیل مجتمع شدن داده‌های ذخیره شده و وحدت ذخیره‌سازی پدیده افزونگی کاهش یافته است.

۱۵- گزینه (۲) صحیح است.

گزینه اول: قاعده همخوانی است. گزینه دوم: ناقص شرط انزوا در اجرای تراکنش‌های همروند است. بر طبق قاعده جامعیت انزوا، همروندی تراکنش‌ها باید کنترل شود تا اثر مخرب روی هم نداشته باشند. گزینه سوم: قاعده یکپارچگی است. گزینه چهارم: قاعده پایایی است.

۱۶- گزینه (۳) صحیح است.

گزینه اول: خطای ناسازگاری نوع است. (Type Miss Match)

گزینه سوم: ناسازگاری داده به مفهوم عدم تطابق داده‌های مربوط به یک نمونه موجودیت در بخش‌های مختلف پایگاه داده است. ناسازگاری داده‌ها هنگامی بروز می‌کند که بنابر دلایلی یک فقره اطلاع در بیش از یک نقطه از بانک ذخیره شود و لازم باشد که به هنگام درآید. اگر عمل بهنگام‌سازی در تمامی نقاطی که آن فقره اطلاع وجود دارد، توسط سیستم مدیریت بانک انجام نشود، ناهمگونی در اطلاعات و به عبارت دیگر پدیده ناسازگاری داده‌ای بروز می‌کند.

گزینه دوم و چهارم: داده‌های مربوط به شاخص یا اندیس (Index) ایجاد شده روی ستون‌های یک جدول به طور خودکار توسط DBMS بروزرسانی می‌گردد.

۱۷- گزینه (۲) صحیح است.

در یک DBMS، امکان دسترسی همزمان به اطلاعات به راحتی وجود دارد. هم‌چنین DBMS، سازگاری اطلاعات را تا جای ممکن کنترل می‌کند. به علاوه امکان پاسخ‌دهی به سوالات متنوعی وجود دارد. پس گزینه‌های اول، سوم و چهارم اشتباه هستند.

در سیستم‌های پرونده‌ای، اطلاعات ممکن است در مکان‌های مختلف با توجه به نوع برنامه‌های کاربردی ذخیره شوند، توجه کنید که در سیستم پرونده‌ای وحدت ذخیره‌سازی اطلاعات وجود ندارد. بنابراین هر برنامه کاربردی، اطلاعات داده‌ای خاص خود را دارد، مثلاً مشخصات یک شخص مانند نام و نام‌خانوادگی و آدرس ممکن است هم در سیستم حساب جاری باشد و هم در سیستم حساب قرص‌الحسنه. حال اگر اطلاعات یک شخص در حساب جاری از بین برود، در صورت وجود می‌توان اطلاعات شخص را از روی سیستم قرص‌الحسنه استخراج نمود. اما در سیستم‌های بانک اطلاعات، به دلیل وحدت ذخیره‌سازی، تکرار اطلاعات به مانند سیستم‌های پرونده‌ای وجود ندارد. بنابراین امنیت فیزیکی اطلاعات در سیستم‌های پرونده‌ای نسبت به سیستم‌های بانک اطلاعاتی، بیش‌تر است، اما امنیت منطقی اطلاعات در سیستم‌های بانک اطلاعات به دلیل وجود مکانیزم‌های حفاظتی که DBMS فراهم می‌کند، مانند تعریف سطوح دسترسی کاربران و مکانیزم پشتیبان‌گیری نسبت به سیستم‌های پرونده‌ای بیش‌تر است.

۱۸- گزینه (۱) صحیح است.

تغییر محتوای داده‌ها (چه در سطح منطقی و چه در سطح فیزیکی) تأثیری روی استقلال داده‌ی منطقی ندارد. هم‌چنین تغییر دیسک نیز بر روی استقلال داده‌ی منطقی بی‌تأثیر است پس گزینه‌های دوم، سوم و چهارم اشتباه‌اند. اما با تغییر ساختار منطقی داده‌ها ممکن است استقلال منطقی داده‌ها از بین برود. مثلاً اگر این تغییر منجر به حذف یک جدول گردد، حتماً استقلال منطقی نقض می‌گردد.

۱۹- گزینه (۴) صحیح است.

یکپارچگی (Integration) که به معنی وحدت ذخیره‌سازی در بانک اطلاعات است. در گام ابتدایی منجر به کاهش افزونگی طبیعی (محتوایی) و به معنای دیگر کاهش تکرار اطلاعات می‌گردد، اما عیب آن این است که امنیت فیزیکی اطلاعات را کاهش می‌دهد. البته در سیستم‌های بانک اطلاعات به دلیل وجود DBMS به عنوان محافظ و نگهدارنده داده‌ها و وجود مکانیزم‌های حفاظتی، مانند تعریف سطوح دسترسی کاربران و

مکانیزم پشتیبان‌گیری، امنیت منطقی اطلاعات نسبت به سیستم‌های پرونده‌ای افزایش می‌یابد که این مزیت یکپارچگی برای سیستم‌های بانک اطلاعات است.

هدف اصلی و ابتدایی از یکپارچگی در سیستم‌های بانک اطلاعات، کاهش افزونگی طبیعی (محتوایی) است حال این‌که این هدف در مقصد مزایا و معایبی در امنیت فیزیکی و منطقی برای اطلاعات به ارمغان می‌آورد، جزء نتیجه‌های رسیدن به این هدف است، و نه خود هدف. بنابراین گزینه چهارم درست و گزینه سوم نادرست است.

سرعت بیش‌تر در دسترسی به اطلاعات، امکاناتی است که DBMS فراهم می‌کند، مانند تعریف شاخص‌ها. البته توجه کنید که سرعت دسترسی به اطلاعات در سیستم‌های بانک اطلاعات بدون در نظر گرفتن تعریف شاخص‌ها، به دلیل عملیات حفاظتی در DBMS، نسبت به سیستم‌های پرونده‌ای کم‌تر است. سهولت کار، برای کاربر نهایی، پس از ساخت بانک اطلاعات، در فاز پیاده‌سازی براساس سلیقه مشتری و سازنده ایجاد می‌گردد و مسائل یکپارچه‌سازی و نرمال‌سازی فدای راحتی کاربر نهایی نمی‌گردد.

۲۰- گزینه (۳) صحیح است.

بررسی و نظارت سیستم و پاسخ به تغییر نیازها یا نیازهای جدید، از وظایف طراح و مدیر (admin) بانک اطلاعاتی است نه DMBS. بنابراین گزینه سوم جزء وظایف DBMS نیست. گزینه‌های دیگر از وظایف DBMS هستند.

۲۱- گزینه (۱) صحیح است.

کاتالوگ سیستم شامل مقادیر داده‌های درون جداول نیست، بلکه شامل «داده‌هایی درباره‌ی داده» است. گزینه‌های دوم، سوم و چهارم نمونه‌هایی از «داده‌هایی درباره‌ی داده» هستند و در کاتالوگ سیستم ذخیره می‌شوند.

۲۲- گزینه (۴) صحیح است.

هر سیستم پایگاه داده توزیع شده شامل مجموعه‌ای از سایت‌ها است که هر یک، یک سیستم پایگاه داده‌ی محلی را در خود نگه می‌دارند. (پس گزینه‌های اول و سوم درست‌اند.) سیستم پایگاه داده‌ی توزیع شده باید از دید کاربر مشابه یک پایگاه داده‌ی متمرکز عمل کند و اگر برای اجرای یک پرس و جو به داده‌های چند سایت نیاز است، عمل تجزیه پرس و جو و ارسال آن به سایت‌ها و سپس دریافت پاسخ‌ها و ترکیب آن‌ها باید از دید کاربر مخفی بماند. (پس گزینه دوم نیز درست است.) سیستم پایگاه توزیع شده می‌تواند شامل هر تعداد سایت باشد، پس گزینه‌ی چهارم عبارت نادرستی است.

۲۳- گزینه (۳) صحیح است.

معماری ANSI برای پایگاه داده‌ها شامل سه سطح زیر است:

۱- **Physical Level**: به ذخیره‌سازی فیزیکی داده‌ها می‌پردازد (در کجا و چگونه ذخیره شوند، اندیس‌گذاری و ...)

۲- **Conceptual Level**: ساختار منطقی پایگاه داده‌ها و مفاهیم داده‌ای را نشان می‌دهد که معمولاً همان مدل رابطه‌ی 3NF می‌باشد.

۳- **External Level**: دید (View) هایی است که توسط کاربران نهایی سیستم دیده می‌شود.

در بین سطوح فوق، سطح اول به ذخیره‌سازی داده، سطح دوم به مفاهیم داده‌ای و سطح سوم به استفاده از داده‌ها مربوط می‌شود.

۲۴- گزینه (۱) صحیح است.

گزینه‌ی اول یک پایگاه داده‌ی تک کاربره را نشان می‌دهد، در صورتی که گزینه‌های دیگر پایگاه داده‌های چند کاربره هستند.

۲۵- گزینه (۴) صحیح است.

برای تضمین جامعیت تراکنش‌ها باید چهار خاصیت (که به خواص ACID معروفند) وجود داشته باشد:

a. خاصیت Atomicity (یک‌پارچگی): هر تراکنش یا به طور کامل اجرا شود یا اصلاً اجرا نشود.

b. خاصیت Consistency (هم‌خوانی یا سازگاری): هر تراکنش باید بانک اطلاعاتی را از یک وضعیت صحیح به وضعیت صحیح دیگر منتقل کند.

خاصیت Isolation (انزوا): اثر چند تراکنش هم‌زمان باید همانند زمانی باشد که آن تراکنش‌ها به طور غیر هم‌زمان اجرا شوند.

c. خاصیت Durability (پایایی) پس از انجام هر تراکنش باید اثر آن برای همیشه باقی بماند و به هیچ صورت، حتی با زلزله، سرقت، آتش‌سوزی و ... از بین نرود.

۲۶- گزینه (۱) صحیح است.

استفاده از کاتالوگ سیستم باعث افزایش استقلال داده‌ای می‌شود.

۲۷- گزینه (۲) صحیح است.

مشخصات شاخص، یکی از انواع اطلاعاتی است که در کاتالوگ سیستم ذخیره می‌شود. به خاطر داشته باشید که کاتالوگ سیستم حاوی «داده درباره‌ی داده» است. یعنی اطلاعاتی (مثل نوع فیلدها، کلید بودن برخی فیلدها و ...) درباره‌ی بانک اطلاعاتی را در خود نگه می‌دارد. مشخصات شاخص نیز یک نوع «داده درباره‌ی داده» است و در کاتالوگ سیستم ذخیره می‌شود. پس با ساختن شاخص جدید، کاتالوگ سیستم تغییر می‌کند.

۲۸- گزینه (۲) صحیح است.

در کاتالوگ سیستم، اسامی جداول، ستون‌های آن‌ها، نوع ستون‌ها و غیره ذخیره می‌شوند. با تعریف یک جدول مجازی نیز باید مشخصات مربوط به آن در کاتالوگ سیستم ذخیره شوند. پس گزینه دوم صحیح است.

به طور کلی کلیه مشخصات مربوط به جداول، شاخص‌ها، دیدها و اطلاعات مربوط به کاربران در کاتالوگ سیستم ذخیره می‌گردد.

۲۹- گزینه (۴) صحیح است.

سیستم بانک اطلاعاتی شی‌گرا OODBMS، سیستم بانک اطلاعاتی است که اجازه می‌دهد تا اطلاعات به شکل اشیاء در برنامه‌نویسی شی‌گرا مورد استفاده قرار گیرد. این سیستم‌ها نسبت به بانک اطلاعاتی رابطه‌ای،

RDBMS نمایش اشیاء مختلط و مرکب را بسیار آسان کرده است، هم‌چنین به کاربر امکان می‌دهد که ضمن استفاده از انواع ساده داده‌ها مانند عدد صحیح، عدد حقیقی، تاریخ و غیره خود نیز به طور دلخواه به تعریف داده بپردازد. در این نوع بانک‌های اطلاعاتی مفاهیمی از قبیل وراثت، چند ریختی و کلاس، کاربرد مناسبی در زبان‌های برنامه‌نویسی شی‌گرا دارد. افزایش هم‌زمانی عملیات، اشتراک داده‌ها، دیده‌ها بین کاربران و کاهش حجم بانک اطلاعات در سیستم بانک اطلاعات رابطه‌ای (RDBMS) هم امکان‌پذیر است و فقط مختص به سیستم‌های بانک اطلاعاتی شی‌گرا (OODBMS) نیست. اما در سیستم بانک اطلاعات شی‌گرا (OODBMS) با استفاده از مفاهیم شی‌گرا می‌توان بین کاربران عملیات جدید تعریف نمود، مانند پردازش تصویر و صدا.

۳۰- گزینه (۲) صحیح است.

معماری ANSI برای پایگاه داده‌ها شامل این اجزاء است: کاربر، زبان میزبان، زبان داده‌ای فرعی، فایل‌های فیزیکی DBMS، مدیر پایگاه داده. تنها گزینه‌ای که هیچ یک از اجزاء فوق را ندارد گزینه دوم است.

۳۱- گزینه (۴) صحیح است.

۳۲- گزینه (۴) صحیح است.

هر پایگاه داده شامل یک یا چند شیما (جدول) است. هر کاتالوگ شامل چند شیما می‌تواند باشد. هر شیما شامل چند Data Object (ستون) است و هر Data Object فقط متعلق به یک شیمای خاص است. البته در بحث کلید خارجی، Data Object یا ستون مشترک، در دو جدول هم‌نام (نه لزوماً) و هم نوع هستند، اما هر یک متعلق به شیما یا جدول خودشان هستند.

۳۳- گزینه (۴) صحیح است.

بر اساس خاصیت پایایی (Durability)، تراکنش‌هایی که به مرحله انجام (Commit) برسند، اثرشان ماندنی است و هرگز به طور تصادفی از بین نمی‌روند. به بیان دیگر تأثیر اجرای موفق یک تراکنش باید به طور دائمی در پایگاه داده ذخیره شود و خرابی‌های بعدی نباید موجب از دست رفتن اثر آن شود. واحد مدیریت بازیابی (ترمیم) وظیفه بازیابی اطلاعات و اطمینان از پایایی و از دست نرفتن آن‌ها را بر عهده دارد.

۳۴- گزینه (۱) صحیح است.

اگر اجرای یک تراکنش مطابق قوانین DBMS باشد، آن تراکنش توسط اجرای دستور Commit از سوی DBMS تأیید می‌گردد. و اگر اجرای یک تراکنش مطابق قوانین DBMS نباشد، آن تراکنش توسط اجرای دستور abort از سوی DBMS لغو می‌گردد. کاربر با اجرای یک تراکنش، خواستار Commit و پذیرش اجرای تراکنش خود می‌باشد. اما در نهایت، DBMS در تأیید یا لغو درخواست کاربر، تصمیم می‌گیرد.

۳۵- گزینه (۴) صحیح است.

استقلال فیزیکی داده‌ها به معنی مصونیت برنامه‌های کاربردی در قبال تغییراتی که در سطح فیزیکی (رسانه

ذخیره‌سازی) پایگاه داده‌ها بروز می‌کند می‌باشد. یعنی اگر تغییری در ذخیره‌سازی داده‌ها انجام گیرد (برای مثال نوع دیسک عوض شود) برنامه‌های کاربردی هیچ تغییری نکنند.

۳۶- گزینه (۱) صحیح است.

با ایجاد و شروع هر تراکنش، یک شناسه توسط DBMS به آن نسبت داده می‌شود.

۳۷- گزینه (۲) صحیح است.

در بانک اطلاعات ممکن است تراکنش‌های همروند وجود داشته باشد (مثل سیستم‌های چند برنامه‌ای) بر طبق خاصیت انزوا، همروندی تراکنش‌ها باید کنترل شود تا اثر مخرب بر روی هم نداشته باشند، به بیان دیگر اثر تراکنش‌های همروند روی یکدیگر چنان است که گویا هر کدام در انزوا انجام می‌شود. به تعریفی دیگر تراکنش‌ها جدا از یکدیگر هستند. اگر چند تراکنش به طور همزمان اجرا شوند، به‌هنگام سازی‌های هر کدام از یکدیگر مخفی می‌مانند تا به اتمام برسند. به عبارتی دیگر، برای دو تراکنش مجزای A و B، تراکنش A می‌تواند به‌هنگام‌سازی‌ها B را پس از پذیرفته شدن آن (Commit) یا B می‌تواند به‌هنگام‌سازی‌های A را پس از پذیرفته شدن A ببیند. اما این دو تراکنش به طور همزمان نمی‌توانند به‌هنگام‌سازی‌های یکدیگر را ببینند. این کنترل‌ها توسط بخشی از DBMS به نام واحد کنترل همروندی (Concurrency control) انجام می‌شود.

۳۸- گزینه (۳) صحیح است.

استقلال منطقی داده‌ها، به معنی مصونیت برنامه‌های کاربردی در قبال تعاریف و تغییراتی که در سطح ادراکی پایگاه داده بروز می‌کند، می‌باشد. یعنی تعریف و تغییر تصویر ادراکی بانک از دید برنامه‌های کاربردی آن‌ها مخفی بماند.

۳۹- گزینه (۲) صحیح است.

یک تراکنش از لحظه شروع به اجرا تا لحظه خاتمه برای پذیرش (commit) یا عدم پذیرش (abort) از سوی DBMS یک تراکنش فعال یا تراکنش نهایی نشده (uncommit) محسوب می‌گردد. همچنین اگر تراکنش پس از خاتمه، مورد تأیید DBMS قرار نگیرد، تراکنش مورد نظر از سوی DBMS یک تراکنش نهایی نشده (uncommit) یا ساقط شده (abort) محسوب می‌گردد.

۴۰- گزینه (۲) صحیح است.

معماری ANSI برای پایگاه داده شامل سه لایه زیر است:

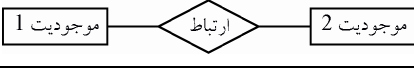
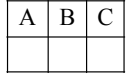

۱- لایه خارجی.

۲- لایه ادراکی (شامل زیر لایه‌های ادراکی عام (طراحی ادراکی) و ادراکی خاص (طراحی منطقی)).

۳- لایه داخلی (فیزیکی).

توجه: به جای کلمه «لایه»، کلمه «تصویر» نیز مورد استفاده قرار می‌گیرد.

به طور کلی برای ایجاد بانک اطلاعات یک محصول نرم‌افزاری مرحله مدل‌سازی بانک شامل مراحل تحلیل (تصویر ادراکی عام)، طراحی (تصویر ادراکی خاص) و مرحله پیاده‌سازی (تصویر فیزیکی) مطابق شکل زیر انجام می‌گردد:

دیدهای کاربران مختلف (view)	کاربر 1 ... کاربر 2 ... کاربر n	تصویر خارجی	
کل بانک بدون توجه به مدل خاصی	ER مدل 	تصویر ادراکی عام (طراحی ادراکی)	تحلیل
کل بانک در قالب مدل انتخابی	مدل رابطه‌ای 	تصویر ادراکی خاص (طراحی منطقی)	طراحی
کل بانک روی رسانه	SQL 	تصویر فیزیکی	پیاپی سازی

لایه خارجی

در این لایه، میزان دسترسی کاربران به پایگاه داده، مشخص و مدیریت می‌گردد. لایه خارجی، تنها لایه‌ای است که به کاربران مربوط می‌شود. هر کاربر با بخشی از بانک سر و کار دارد و فقط اجازه دسترسی به بخشی از بانک به او داده می‌شود. برای مثال در بانک اطلاعات مربوط به بانک مرکزی، کاربر نهایی (کارمندان) بخش ارز مسافری، حق دسترسی به بخش‌های دیگر بانک مثل حساب افراد و وام‌ها را ندارند.

قانون «پنهان‌سازی اطلاعات» که می‌گوید «به هر کس به همان اندازه اطلاعات بده که نیاز دارد، نه بیشتر» در اینجا صادق است. بنابراین در لایه خارجی دیدهای مختلف کاربران مطرح است و هر کدام بخشی از بانک را می‌بینند.

توجه: لایه خارجی لایه‌ای است که کاربران با آن سر و کار دارند، لایه‌های دیگر به مدیر و برنامه‌سازان بانک اطلاعات مربوط می‌شود.

لایه ادراکی

برای ایجاد یک بانک اطلاعات، پس از شناسایی نیازمندی‌ها، در ادامه در فعالیت تحلیل (تصویر ادراکی عام) موجودیت‌های محیط عملیاتی مدل سازی می‌شود. یکی از معروف‌ترین مدل‌هایی که در این لایه مورد استفاده قرار می‌گیرد، مدل ER می‌باشد. خروجی فعالیت تحلیل ERD می‌باشد که به عنوان ورودی فعالیت طراحی تلقی می‌گردد.

در ادامه و در فعالیت طراحی، ERD، توسط مدل رابطه‌ای به تصویر ادراکی خاص بدل می‌گردد و در نهایت در فاز پیاده‌سازی با توجه یک DBMS خاص مانند SQL Server، تصویر ادراکی خاص به تصویر فیزیکی تبدیل می‌شود.

توجه: طراحی ادراکی به مدل خاصی وابستگی ندارد چون در شروع فعالیت‌ها قرار دارد، اما طراحی منطقی به مدل خاصی بستگی دارد، بسته به اینکه در فعالیت طراحی ادراکی چه مدلی انتخاب شود، طراحی منطقی باید تبعیت کند. اگر طراحی ادراکی ساخت یافته بود، طراحی منطقی نیز باید ساخت یافتگی را دنبال کند و اگر طراحی ادراکی شی گرا بود، طراحی منطقی نیز باید شی‌گرایی را دنبال کند.

لایه داخلی (فیزیکی)

پس از مرحله مدل‌سازی بانک یعنی مراحل فعالیت تحلیل (تحلیل موجودیت‌ها توسط مدل ER و تحلیل پرس و جوها توسط حساب رابطه‌ای) و فعالیت طراحی (طراحی جدول‌ها توسط مدل رابطه‌ای و طراحی پرس و جوها توسط جبر رابطه‌ای)، نوبت به مرحله پیاده‌سازی می‌رسد. در این مرحله جداول و پرس و جوهای مورد نیاز توسط یک زبان بیانی (Declarative) مانند SQL پیاده‌سازی می‌گردند.

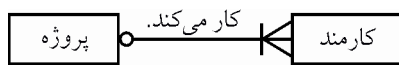
توجه: SQL شامل مجموعه دستورات DDL (خلق جداول، ساختارها و ظرف‌ها)، DML (خلق پرس و جوها و محتوا) و DCL (خلق امنیت) می‌باشد که به اجتماع این دستورات DSL (Data Sub Language) یا QL (Query Language) گفته می‌شود.

تست‌های فصل دوم: مدل نهاد و رابطه

۱- در نمودار موجودیت ارتباط (EER) تعریف زیر معرف کدام نوع خصیصه (Attribute) است؟
(مهندسی کامپیوتر- آزاد ۸۵)

«خصیصه‌ای که در موجودیت و در نهایت در پایگاه داده برای آن فیلدی تعریف نشده است.»
(۱) چند مقداری (۲) مشتق (۳) ساده (۴) مرکب

۲- با توجه به ERD روبه‌رو کدام جمله زیر نادرست است؟ (مهندسی IT - دولتی ۸۷)

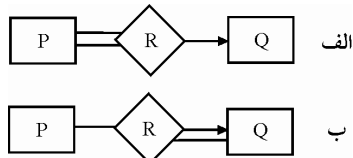


- (۱) ممکن است پروژه‌ای کارمند نداشته باشد.
(۲) هر کارمند حداکثر در یک پروژه کار می‌کند.
(۳) ممکن است کارمندی در هیچ پروژه‌ای کار نکند.
(۴) در یک پروژه ممکن است چندین کارمند کار کنند.

۳- کدام یک از موارد زیر برای متمایز کردن نمونه موجودیت‌ها از یکدیگر مورد استفاده قرار می‌گیرد؟ (مهندسی کامپیوتر- آزاد ۸۸)

(۱) ترتیب صفات خاصه (۲) اسامی صفات خاصه (۳) مقادیر صفات خاصه (۴) ترکیب صفات خاصه

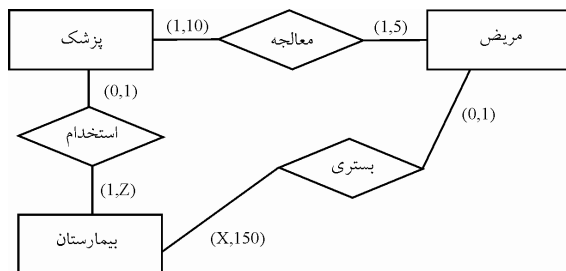
۴- نمودارهای ER زیرا را در نظر بگیرید این عبارت که تعداد نمونه موجودیت‌های مجموعه موجودیت P بایستی بزرگ‌تر یا مساوی با تعداد نمونه موجودیت‌های مجموعه موجودیت Q باشند را می‌توان توسط کدام یک از دو نمودار زیر بیان نمود؟ (مهندسی کامپیوتر- آزاد ۹۰ گروه ب)



- (۱) فقط نمودار الف
(۲) فقط نمودار ب
(۳) هر دو
(۴) هیچ کدام

۵- یک دیاگرام ER، ناهمگون خوانده می‌شود اگر هیچ امکانی برای ارضای تمامی محدودیت‌های آن وجود نداشته باشد، کدام یک از پاسخ‌های زیر موجب می‌شود تا دیاگرام زیر ناهمگون گردد:

(مهندسی IT - دولتی ۹۱)



- (۱) $X=1; Z=10$
(۲) $X=100; Z=5$
(۳) $X=1; Z=50$
(۴) $X=100; Z=100$

۶- در مدل ER حداکثر درجه ارتباط میان موجودیت‌ها چه مقداری می‌باشد؟ (مهندسی IT - آزاد ۹۳)

(۱) با توجه به شرایط مسأله مشخص گردیده و حداکثری برای آن نمی‌توان در نظر گرفت.
(۲) 3 (۳) 4 (۴) 5

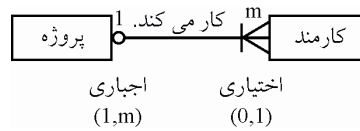
پاسخ تست‌های فصل دوم: مدل نهاد و رابطه

۱- گزینه (۲) صحیح است.

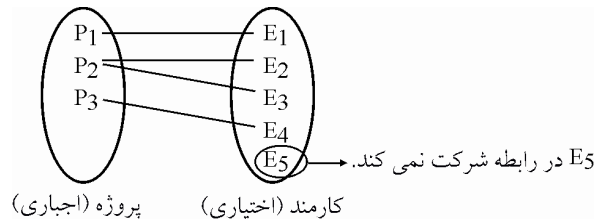
صفت مشتق در موجودیت وجود خارجی ندارد و در نمودار ER به صورت یک بیضی خط‌چین ترسیم می‌شود. صفت مشتق صفتی است که به کمک صفت‌های دیگر می‌توان آن را محاسبه کرد. مثلاً سن، یک صفت مشتق است که با توجه به تاریخ تولد قابل محاسبه می‌باشد. تصمیم‌گیری در مورد صفت مشتق به عهده طراح است مثلاً معدل کل برای دانشجو بهتر است مشتق باشد زیرا مرتباً با گذراندن دروس بیش‌تر، این معدل تغییر می‌کند ولی برای فارغ‌التحصیل معدل کل بهتر است بخشی از پدیده باشد.

۲- گزینه (۱) صحیح است.

با توجه به نمودار، حضور کارمند در رابطه اختیاری و حضور پروژه در رابطه اجباری می‌باشد.



شکل زیر گویای نمودار ER است: (به طور مثال)



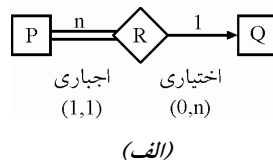
با توجه به ارتباط یک به چند پروژه با کارمند و اجباری بودن حضور نمونه موجودیت‌های پروژه قید (1,m) برای پروژه در نظر گرفته می‌شود، بنابراین رابطه زیر را داریم:
 یک پروژه حداقل یک و حداکثر m کارمند دارد. بنابراین گزینه اول نادرست و گزینه چهارم درست است.
 همچنین با توجه به ارتباط چند به یک کارمند با پروژه و اختیاری بودن حضور نمونه موجودیت‌های کارمند، قید (0,1) برای کارمند در نظر گرفته می‌شود، بنابراین رابطه زیر را داریم:
 یک کارمند حداقل هیچ (مانند E5) و حداکثر یک پروژه دارد. بنابراین گزینه‌های دوم و سوم درست است.

۳- گزینه (۳) صحیح است.

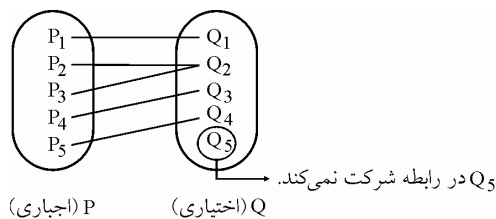
وجه تمایز موجودیت‌ها (اسامی عام) در صفات آنها است.
 مانند موجودیت‌های استاد و دانشجو که دارای صفات متفاوت مانند شماره استادی و شماره دانشجویی هستند و وجه تمایز نمونه موجودیت‌ها (اسامی خاص) در مقادیر آنها است.
 مانند مقادیر متفاوت داده‌های مربوط به هر یک از دانشجویان. مثلاً برای موجودیت دانشجو مقادیر صفات خاصه مثل نام و نام‌خانوادگی نمونه‌ها را از یک‌دیگر متمایز می‌کند.

۴- گزینه (۳) صحیح است.

با توجه به نمودار (الف)، حضور نمونه‌های P در رابطه اجباری و حضور نمونه‌های Q در رابطه اختیاری می‌باشد.



شکل زیر گویای نمودار ER است: (به طور مثال)

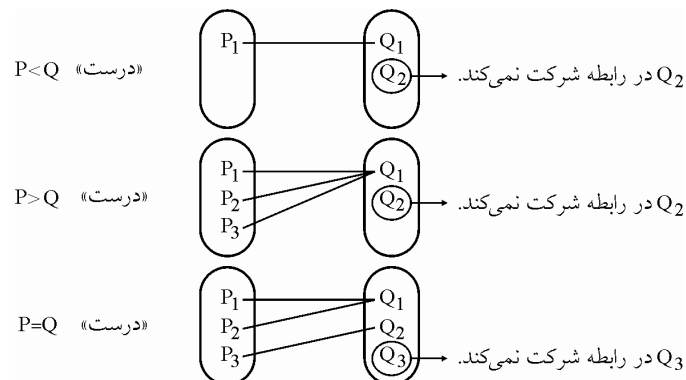


با توجه به ارتباط چند به یک P با Q و اجباری بودن حضور نمونه موجودیت‌های P، قید (1,1) برای P در نظر گرفته می‌شود. بنابراین رابطه زیر را داریم:

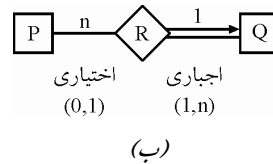
یک نمونه موجودیت P حداقل با یک و حداکثر با یک نمونه موجودیت از Q رابطه دارد.

همچنین با توجه به ارتباط یک به چند P با Q و اختیاری بودن حضور نمونه موجودیت‌های Q، قید (0,n) برای Q در نظر گرفته می‌شود. بنابراین رابطه زیر را داریم:

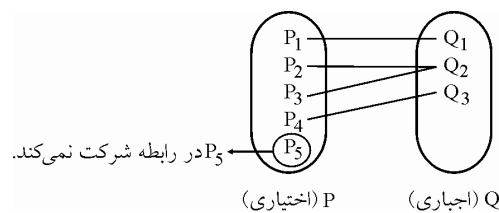
یک نمونه موجودیت Q حداقل هیچ (مانند Q5) و حداکثر با n نمونه موجودیت از P رابطه دارد. بنابراین تعداد نمونه موجودیت‌های P کوچک‌تر یا بزرگ‌تر یا مساوی با تعداد نمونه موجودیت‌های Q است. چون ارتباط از P اجباری به Q اختیاری چند به یک است و نه یک به چند.



با توجه به نمودار (ب)، حضور نمونه‌های P در رابطه اختیاری و حضور نمونه‌های Q در رابطه اجباری می‌باشد.

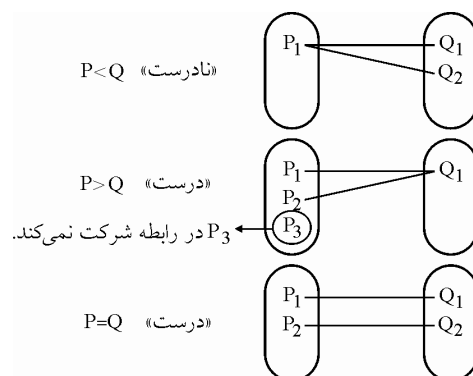


شکل زیر گویای نمودار ER است: (به طور مثال)



با توجه به ارتباط چند به یک P با Q و اختیاری بودن حضور نمونه موجودیت‌های P، قید (0,1) برای P در نظر گرفته می‌شود بنابراین رابطه زیر را داریم:
یک نمونه موجودیت P حداقل هیچ (مانند P₅) و حداکثر با یک نمونه موجودیت از Q رابطه دارد، یعنی ممکن است نمونه موجودیت‌هایی از P وجود داشته باشند که با هیچ نمونه‌ای از موجودیت Q رابطه ندارند. (مانند P₅)

هم‌چنین با توجه به ارتباط یک به چند P با Q و اجباری بودن حضور نمونه موجودیت‌های Q، قید (1,n) برای Q در نظر گرفته می‌شود. بنابراین رابطه زیر را داریم:
یک نمونه موجودیت Q حداقل با یک و حداکثر با n نمونه موجودیت از P رابطه دارد. بنابراین حتماً تعداد نمونه موجودیت‌های P بزرگ‌تر یا مساوی با تعداد نمونه موجودیت‌های Q است. چون ارتباط از P اختیاری به Q اجباری چند به یک است و نه یک به چند، حالت کوچک‌تر بودن P نسبت به Q امکان‌پذیر نیست.



۵- گزینه (۲) صحیح است.

یکی از ویژگی‌های ارتباطات در نمودار ER، حد (محدودیت) می‌باشد که در پایین خط ارتباط مقدار حداقل و حداکثر آن در پرانتز نوشته می‌شود.

حال اگر این حدود یا محدودیت‌ها تحت هیچ شرایطی با یکدیگر سازگار نباشند، نمودار ER را ناهمگون گویند.

با توجه به دیاگرام:

مقدار x حداقل تعداد بیماران در بیمارستان

و مقدار z حداکثر تعداد پزشکان در بیمارستان است.

برای یافتن پاسخ، گزینه‌های مختلف را با توجه به مقادیر x و z آزمایش می‌کنیم:

گزینه اول:

- ۱- هر بیمارستان می‌تواند حداقل 1 ($x=1$) و حداکثر 150 بیمار داشته باشد.
 - ۲- هر بیمارستان می‌تواند حداقل 1 و حداکثر 10 ($z=10$) پزشک در استخدام داشته باشد.
 - ۳- هر پزشک می‌تواند حداقل در هیچ و حداکثر در 1 بیمارستان استخدام باشد.
 - ۴- هر پزشک می‌تواند حداقل 1 و حداکثر 10 بیمار داشته باشد.
 - ۵- هر بیمار می‌تواند حداقل بستری نباشد و حداکثر در 1 بیمارستان بستری باشد.
 - ۶- هر بیمار می‌تواند حداقل 1 و حداکثر 5 پزشک معالج داشته باشد.
- زمانی که حداقل تعداد بیماران 1 نفر ($x=1$) و حداکثر تعداد پزشکان استخدام شده در بیمارستان 10 نفر ($z=10$) باشد، تمامی محدودیت‌ها ارضا شده، بنابراین نمودار ER ناهمگون نخواهد بود.
- با توجه به مورد 4، برای معالجه حداقل 1 نفر بیمار ($x=1$) نیاز به 1 نفر پزشک داریم و با توجه به مورد 2، حداقل 1 و حداکثر 10 نفر پزشک ($z=10$) داریم که محدودیت ارضا می‌شود. با 10 نفر پزشک می‌توان حداکثر 100 نفر بیمار را معالجه کرد و با توجه به حداکثر 150 نفر بیمار بنابراین با حداکثر 100 نفر بیمار این امکان وجود دارد که این محدودیت هم ارضا شود. چون حالت حداکثری است و امکان ارضا دارد.

گزینه دوم:

- ۱- هر بیمارستان می‌تواند حداقل 100 ($x=100$) و حداکثر 150 بیمار داشته باشد.
 - ۲- هر بیمارستان می‌تواند حداقل 1 و حداکثر 5 ($z=5$) پزشک در استخدام داشته باشد.
 - ۳- هر پزشک می‌تواند حداقل در هیچ و حداکثر در 1 بیمارستان استخدام باشد.
 - ۴- هر پزشک می‌تواند حداقل 1 و حداکثر 10 بیمار داشته باشد.
 - ۵- هر بیمار می‌تواند حداقل بستری نباشد و حداکثر در 1 بیمارستان بستری باشد.
 - ۶- هر بیمار می‌تواند حداقل 1 و حداکثر 5 پزشک معالج داشته باشد.
- زمانی که حداقل تعداد بیماران 100 نفر ($x=100$) و حداکثر تعداد پزشکان استخدام شده در بیمارستان 5 نفر ($z=5$) باشد، تمامی محدودیت‌ها ارضا نشده، بنابراین نمودار ER ناهمگون خواهد بود.
- با توجه به مورد 4، برای معالجه حداقل 100 نفر بیمار نیاز به 10 پزشک داریم، در حالی که با توجه به مورد 2، حداقل 1 و حداکثر 5 نفر پزشک ($z=5$) در استخدام بیمارستان داریم و با این تعداد تنها 50 بیمار را می‌توانیم معالجه کنیم و نه 100 بیمار را. بنابراین تحت هیچ شرایطی این امکان وجود ندارد که این محدودیت ارضا شود و این یعنی ناهمگونی.

گزینه سوم:

- ۱- هر بیمارستان می‌تواند حداقل 1 ($x=1$) و حداکثر 150 بیمار داشته باشد.

۲- هر بیمارستان می‌تواند حداقل 1 و حداکثر 50 ($z=50$) پزشک در استخدام داشته باشد.
 ۳- هر پزشک می‌تواند حداقل در هیچ و حداکثر در 1 بیمارستان استخدام باشد.
 ۴- هر پزشک می‌تواند حداقل 1 و حداکثر 10 بیمار داشته باشد.
 ۵- هر بیمار می‌تواند حداقل بستری نباشد و حداکثر در 1 بیمارستان بستری باشد.
 ۶- هر بیمار می‌تواند حداقل 1 و حداکثر 5 پزشک معالج داشته باشد.
 زمانی که حداقل تعداد بیماران 1 نفر ($x=1$) و حداکثر تعداد پزشکان استخدام شده در بیمارستان 50 نفر ($z=50$) باشد، تمامی محدودیت‌ها ارضا شده بنابراین نمودار ER ناهمگون نخواهد بود.
 با توجه به مورد 4، برای معالجه حداقل 1 نفر بیمار ($x=1$) نیاز به 1 نفر پزشک داریم و با توجه به مورد 2، حداقل 1 و حداکثر 50 نفر پزشک ($z=50$) داریم که محدودیت ارضا می‌شود. با 50 نفر پزشک حداکثر می‌توان 500 نفر بیمار را معالجه کرد که این محدودیت هم ارضا می‌شود.

گزینه چهارم:

۱- هر بیمارستان می‌تواند حداقل 100 ($x=100$) و حداکثر 150 بیمار داشته باشد.
 ۲- هر بیمارستان می‌تواند حداقل 1 و حداکثر 100 ($z=100$) پزشک در استخدام داشته باشد.
 ۳- هر پزشک می‌تواند حداقل در هیچ و حداکثر در 1 بیمارستان استخدام باشد.
 ۴- هر پزشک می‌تواند حداقل 1 و حداکثر 10 بیمار داشته باشد.
 ۵- هر بیمار می‌تواند حداقل بستری نباشد و حداکثر در 1 بیمارستان بستری باشد.
 ۶- هر بیمار می‌تواند حداقل 1 و حداکثر 5 پزشک معالج داشته باشد.
 زمانی که حداقل تعداد بیماران 100 نفر ($x=100$) و حداکثر تعداد پزشکان استخدام شده در بیمارستان 100 نفر ($z=100$) باشد، تمامی محدودیت‌ها ارضا شده بنابراین نمودار ER ناهمگون نخواهد بود.
 با توجه به مورد 4، برای معالجه حداقل 100 نفر بیمار ($x=100$) نیاز به 10 نفر پزشک داریم، و با توجه به مورد 2، حداقل 1 و حداکثر 100 نفر پزشک ($z=100$) داریم که محدودیت ارضا می‌شود. با 100 نفر پزشک حداکثر می‌توان 1000 نفر بیمار را معالجه نمود که این محدودیت هم ارضا می‌شود.

۶- گزینه (۳) صحیح است.

به تعداد موجودیت‌هایی که در یک ارتباط مشارکت دارند، درجه ارتباط گفته می‌شود. درجه ارتباط در مدل ER عددی صحیح و کوچکتر از 5 است. ارتباط‌های درجه 1، 2 و 3 معمول، ارتباط درجه 4 کمیاب و غیر معمول است و ارتباط بالاتر از درجه 4 قابل رسم کردن نیست.

تست‌های فصل سوم: مدل رابطه‌ای

(ک(رشناسی ارشد- دولتی ۷۳)

۱- کدام یک از چهار گزینه زیر کلید جدول رابطه‌ای زیر می‌باشد؟

C ₁	C ₂	C ₃
S	B	D ₁
T	J	D ₁
M	B	D ₁
T	C	D ₂
S	K	D ₂

(۱) {C₁}

(۲) {C₂, C₃}

(۳) {C₁, C₃}

(۴) {C₂}

(ک(رشناسی ارشد- دولتی ۸۲)

۲- کدام یک از کلیدهای زیر می‌تواند مقدار NULL را بگیرد؟

(۲) کلید خارجی

(۱) کلید جستجو

(۴) نامزد کلیدی (Candidate Key)

(۳) کلید hash

۳- جدولی با n ستون دارای یک کلید اصلی و یک کلید کاندید دیگر هر کدام شامل یک صفت می‌باشد. این

(مهندسی کامپیوتر- دولتی ۸۴)

جدول چند ابرکلید (Super Key) دارد؟

(۴) 2^{n-1}

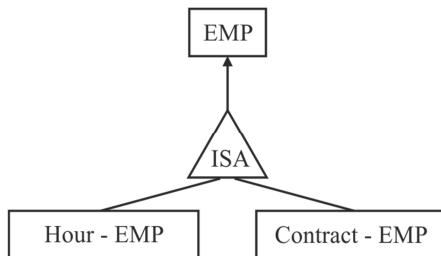
(۳) $3 \times 2^{n-3}$

(۲) 2^{n-2}

(۱) $3 \times 2^{n-2}$

۴- رابطه ISA (ارث‌بری) در نمودار ER زیر را چگونه می‌توان در مدل رابطه‌ای پیاده‌سازی نمود؟

(مهندسی کامپیوتر- دولتی ۸۴)



(۱) رابطه ISA در مدل رابطه‌ای قابل پیاده‌سازی

نیست.

(۲) همه ویژگی‌های EMP را در Hour-EMP و

Contract-EMP کپی می‌نمایم.

(۳) کلید اصلی EMP را در Hour-EMP و

Contract-EMP به عنوان کلید اصلی و کلید

خارجی مجدداً تعریف می‌کنیم.

(۴) موارد ۲ و ۳ هر دو صحیح می‌باشند.

۵- برای پیاده‌سازی رابطه IS-A در یک نمودار ER به مدل رابطه‌ای کدام یک از گزینه‌های زیر بهترین است؟

(مهندسی کامپیوتر- دولتی ۸۵)

(۱) کلید اصلی موجودیت پدر به عنوان کلید خارجی به موجودیت‌های فرزند آن موجودیت اضافه شود.

(۲) توسط موجودیت پلی (Bridge) ایجاد می‌شود که شامل کلید اصلی موجودیت پدر و موجودیت‌های فرزندان آن می‌باشد.

(۳) بخشی از ویژگی‌های موجودیت پدر که در موجودیت‌های فرزند نیاز است را به موجودیت‌های فرزند اضافه می‌کند.

(۴) کلید اصلی موجودیت‌های فرزند را به عنوان کلید خارجی به موجودیت پدر اضافه می‌کند.

۶- جداول S تهیه کنندگان و SP تولید، مطابق زیر مفروض هستند که در آن‌ها S# شماره تهیه کننده، Sname نام و Age سن تهیه کننده، Pname نام قطعه و QTY تعداد قطعه تولید شده توسط S# است. آن‌گاه در جداول زیر کدام مشکل وجود دارد؟ (مهندسی IT- آزاد ۸۵)

SP			S		
S#	Pname	QTY	S#	Sname	Age
S4	مهره	20	S1	احمدی	28
S2	---	50	S5	حسینی	28
S4	پیچ	50	S2	جلالی	34
S5	مهره	20	S4	---	32

(۱) نقض جامعیت دامنه خصیصه (Domain/Attribute)

(۲) نقض جامعیت ارجاع (Referential)

(۳) نقض جامعیت موجودیت (Entity Integrity)

(۴) نقض جامعیت درون رابطه‌ای (Intra- relation Integrity)

۷- کدام گزینه از خواص رابطه (Relation) نیست؟ (مهندسی IT- آزاد ۸۶)

(۱) در یک رابطه مشخص تاپل‌ها بدون ترتیب هستند.

(۲) در یک رابطه مشخص تاپل‌های تکراری وجود ندارد.

(۳) در یک رابطه همه مقادیر خصیصه‌ها غیرقابل تجزیه هستند.

(۴) در یک رابطه مشخص خصیصه‌ها دارای ترتیب هستند.

۸- کدام مورد در مدل رابطه‌ای جایز نیست؟ (مهندسی IT- دولتی ۸۶)

(۱) دامنه مرکب (Composite Domain)

(۲) رابطه تهی

(۳) ارتباط $m:n$ و $n > 1$ و $m > 1$

(۴) ارتباط $1:n$ و $n > 1$

۹- کدام جمله در مورد کلید خارجی صحیح نیست؟ (مهندسی کامپیوتر- آزاد ۸۷)

(۱) اگر صفتی کلید خارجی باشد، کلید کاندید هم هست.

(۲) مقدار NULL برای آن مجاز است.

(۳) در جدول دیگر کلید اصلی است.

(۴) معمولاً برای Join جداول به کار می‌رود.

۱۰- کدام عبارت، در مورد کلید خارجی در مدل رابطه‌ای صحیح است؟ (مهندسی نرم‌افزار- دولتی ۸۸)

(۱) کلید خارجی یکی از کلیدهای کاندید همان رابطه است.

(۲) کلید خارجی بایستی کلید اصلی رابطه دیگری باشد. (نباید کلید فرعی باشد)

(۳) کلید خارجی بایستی خصیصه ساده باشد.

(۴) کلید خارجی یک رابطه می‌تواند متناظر با مقادیر کلید کاندید همان رابطه باشد.

۱۱- اگر FK کلید خارجی رابطه R1 متناظر با نامزد کلیدی CK از رابطه R2 باشد، کدام عبارت صحیح است؟ (مهندسی IT- دولتی ۸۸)

- (۱) FK نمی‌تواند NULL باشد.
- (۲) روابط R1 و R2 متفاوت هستند.
- (۳) هر مقدار غیر NULL برای FK برابر یک مقدار CK است.
- (۴) هر مقدار غیر NULL برای CK برابر یک مقدار FK است.

۱۲- کدام قانون مرجعیتی بیان می‌کند که کلید خارجی نمی‌تواند مقداری داشته باشد که در جدول مرجع وجود ندارد؟ (مهندسی کامپیوتر- آزاد ۸۸)

- (۱) جامعیت صفت (Attribute Integrity)
- (۲) جامعیت ارجاع (Referential Integrity)
- (۳) جامعیت دامنه‌ای (Domain Integrity)
- (۴) جامعیت درون رابطه‌ای (Intra-relational Integrity)

۱۳- کدام جمله در مورد کلید خارجی صحیح نیست؟ (مهندسی کامپیوتر- آزاد ۸۸)

- (۱) در یک رابطه کلید اصلی و در رابطه دیگر یک قلم اطلاعاتی ساده می‌باشد.
- (۲) کلید خارجی یکی از کلیدهای کاندید می‌باشد.
- (۳) نقش کلید خارجی از نظر مدل رابطه‌ای ایجاد ارتباط بین جداول است.
- (۴) یک فیلد مشترک بین دو یا چند رابطه است که در Join رابطه‌ها به کار گرفته می‌شود.

۱۴- کدام کلید با بقیه متفاوت است؟ (مهندسی IT- آزاد ۸۸)

- (۱) کلید اصلی
- (۲) کلید کاندید
- (۳) کلید جانشین
- (۴) کلید خارجی

۱۵- کدام گزینه نادرست است؟ (مهندسی IT- آزاد ۸۹)

- (۱) جامعیت ارجاع یعنی کلید خارجی نمی‌تواند مقداری داشته باشد که در جدول مرجع وجود ندارد.
- (۲) جامعیت دامنه‌ای یعنی مقادیر کلید خارجی باید از درون مقادیر کلید اصلی معادلش انتخاب شود.
- (۳) جامعیت موجودیتی یعنی هیچ جزء تشکیل‌دهنده کلید اصلی نباید دارای مقدار هیچ (NULL Value) باشد.
- (۴) جامعیت درون رابطه‌ای یعنی رابطه عضو تکراری نداشته باشد و کلیدهایش مقدار تکراری نداشته باشد.

۱۶- برای تعیین کلید اصلی یک رابطه توجه به کدام یک از موارد زیر ضروری است؟

(مهندسی کامپیوتر- دولتی ۹۰)

- (۱) معنی خصیصه‌های رابطه در دنیای خارج
- (۲) عملیات مورد نیاز روی رابطه
- (۳) مقادیر ابرکلیدهای رابطه
- (۴) مقادیر خصیصه‌های رابطه در زمان بار کردن رابطه

۱۷- رابطه‌های r و s را در نظر بگیرید.

(مهندسی کامپیوتر- دولتی ۹۰)

$r(R_1 : D_1, R_2 : D_2, R_3 : D_3)$

$s(R_2 : D_2, R_3 : D_3, R_4 : D_4)$

کدام زیر مجموعه از صفات r می تواند کلید خارجی باشد؟

- (۱) R_1, R_2 (۲) R_3 (۳) R_2 (۴) R_2, R_3

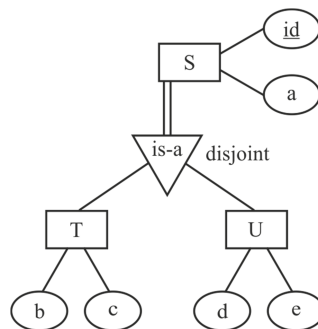
۱۸- تضمین سازگاری (Consistency) بانک‌های اطلاعاتی به کدام مورد مربوط نمی‌شود؟

(مهندسی کامپیوتر- آزاد ۹۱)

- (۱) بانک‌های اطلاعاتی مبتنی بر مدل‌های داده‌ای ایجاد می‌شوند که دارای قواعد جامعیتی هستند.
 (۲) در هر سیستم پایگاه داده مکانیزم‌هایی برای کنترل همروندی و اجرای اتمیک تراکنش‌ها وجود دارد.
 (۳) در پایگاه‌های داده مکانیزم‌های ترمیم تعبیه شده است.
 (۴) هر رابطه حداقل یک کلید کاندید (نامزد) دارد.

(مهندسی IT- دولتی ۹۲)

۱۹- قطعه زیر از یک نمودار موجودیت- رابطه را در نظر بگیرید:



کدام یک از مجموعه رابطه‌های زیر برای تبدیل این قطعه به مدل رابطه‌ای مناسب‌ترین است؟

- (۱) $T(id, a, b, c)$, $U(id, a, d, e)$
 (۲) $S(id, a)$, $T(id, b, c)$, $U(id, d, e)$
 (۳) مقدار $R(id, a, b, c, d, e, Type)$ ، مقدار $Type$ یکی از سه مقدار S, T یا U می‌باشد.
 (۴) $S(id, a, b, c, d, e, isT, isU)$ ، دو مقدار isU و isT مقادیر $True$ یا $False$ را می‌پذیرند.

(مهندسی IT- دولتی ۹۲)

۲۰- با توجه به دو گزاره زیر کدام یک از گزینه‌های زیر صحیح است؟

- (۱) اگر K_1 و K_2 کلیدهای کاندید در رابطه r باشند، آنگاه $K_1 \cup K_2$ نیز یک کلید کاندید در رابطه r است.
 (۲) اگر K_1 و K_2 ابرکلیدهای (super key) در رابطه r باشند، آنگاه $K_1 \cap K_2$ نیز یک کلید کاندید در رابطه r است.

- (۱) هر دو گزاره صحیح هستند.
 (۲) هر دو گزاره غلط هستند.
 (۳) تنها گزاره دوم صحیح است.
 (۴) تنها گزاره اول صحیح است.

۲۱- کدام یک از موارد زیر بیانگر جامعیت ارجاعی (Referential Integrity) در پایگاه داده‌ها می‌باشد؟

(مهندسی کامپیوتر- آزاد ۹۲)

- (۱) هر مقدار برای مجموعه‌ای از صفات در یک رابطه عیناً برای آن صفات در رابطه مقابل موجود باشد.
 (۲) کلید اصلی در یک رابطه همان کلید کاندید در رابطه مقابل باشد.
 (۳) کلید اصلی در یک رابطه همان مقادیر کلید ثانویه در رابطه مقابل باشد.
 (۴) همه موارد.

۲۲- کدام کلید در بانک اطلاعات، کمینه (minimal) نمی‌باشد؟ (مهندسی کامپیوتر- آزاد ۹۳)

- (۱) کلید اصلی (۲) ابرکلید (Super key) (۳) کلید خارجی (۴) کلید کاندید

۲۳- «قرار داشتن کلید خارجی یک رابطه در رابطه دیگر به صورت کلید» کدام نوع جامعیت (integrity) را

مورد توجه قرار می‌دهد؟ (مهندسی کامپیوتر- آزاد ۹۳)

- (۱) جامعیت دامنه‌ای (۲) جامعیت درون رابطه‌ای
(۳) جامعیت ارجاع (۴) خصیصه فوق، جامعیت محسوب نمی‌گردد.

۲۴- کدام گزینه بر «صحیح بودن هر رابطه به تنهایی» تأکید می‌نماید؟ (مهندسی IT - آزاد ۹۳)

- (۱) جامعیت دامنه‌ای (۲) جامعیت درون رابطه‌ای
(۳) جامعیت ارجاع (۴) استقلال داده‌ای

۲۵- در مدل رابطه‌ای پایگاه داده یک سیستم آموزشی، ارتباط یک به چند میان درس و پیش نیازهای آن، ارتباط

چند به چند میان درس و دانشکده، و ارتباط یک به یک ریاست میان گروه آموزشی و استاد وجود دارد.

کدام عبارت نمایش مناسبتر هر رابطه را به ترتیب بیان می‌کند؟ (مهندسی IT - دولتی ۹۴)

(۱) درج کلید اصلی درس در موجودیت درس به عنوان کلید خارجی، درج کلیدهای اصلی دانشکده و درس در یک موجودیت جدید به عنوان کلیدهای خارجی، درج کلید اصلی استاد در موجودیت گروه آموزشی به عنوان کلید خارجی.

(۲) درج کلیدهای اصلی درس و پیش‌نیازهای آن در یک موجودیت جدید به عنوان کلیدهای خارجی، درج کلید اصلی دانشکده در موجودیت درس به عنوان کلید خارجی، درج کلیدهای اصلی گروه آموزشی و استاد در یک موجودیت جدید به عنوان کلیدهای خارجی.

(۳) درج کلیدهای اصلی درس و پیش‌نیازهای آن در یک موجودیت جدید به عنوان کلیدهای خارجی، درج کلیدهای اصلی دانشکده و درس در یک موجودیت جدید به عنوان کلیدهای خارجی، و درج کلیدهای اصلی استاد و گروه آموزشی در یک موجودیت جدید به عنوان کلیدهای خارجی.

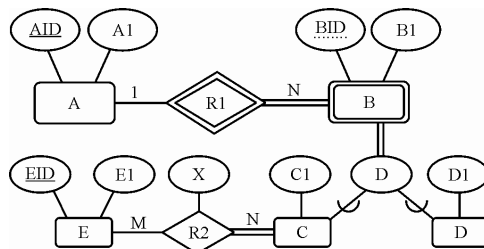
(۴) درج کلید اصلی درس در موجودیت درس به عنوان کلید خارجی، درج کلیدهای اصلی دانشکده و درس در یک موجودیت جدید به عنوان کلیدهای خارجی، درج کلید اصلی گروه آموزشی در موجودیت استاد به عنوان کلید خارجی.

۲۶- با فرض وجود نمودار EER زیر کدام یک از رابطه‌های زیر به درستی طراحی نشده است؟

(مهندسی فناوری اطلاعات-دولتی ۹۵)

A (AID, A1) B (BID, B1) E (EID, E1, AID, BID)

C (AID, BID, B1, C1) D (AID, BID, B1, D1) R2 (AID, BID, EID, X)



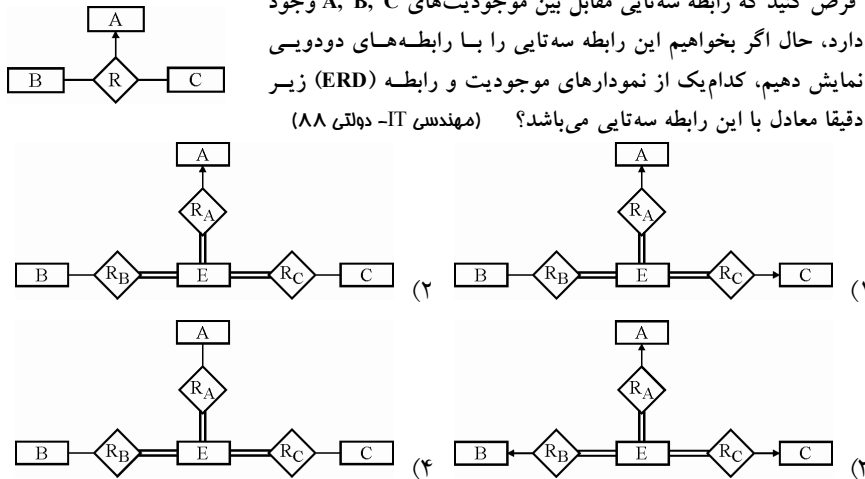
A, B (۱)

B, E (۲)

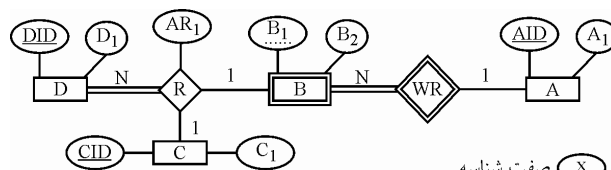
C, D (۳)

E, C (۴)

۲۷- فرض کنید که رابطه سه تایی مقابل بین موجودیت های A, B, C وجود دارد، حال اگر بخواهیم این رابطه سه تایی را با رابطه های دودویی نمایش دهیم، کدام یک از نمودارهای موجودیت و رابطه (ERD) زیر دقیقاً معادل با این رابطه سه تایی می باشد؟ (مهندسی IT- دولتی ۸۸)



۲۸- کدام مورد، شامل طراحی منطقی درست ارتباط R در نمودار ER زیر است؟ (مهندسی IT- دولتی ۹۶)



○ X صفت شناسه

○ X صفت ممیزه موجودیت ضعیف

(۱) $R(\underline{DID}, AID, B_1, CID, AR_1)$ به صورت یک جدول مستقل P.K.

(۲) مستتر در جدول موجودیت B $B(\underline{B_1}, B_2, DID, CID, AR_1)$ P.K. F.K. F.K.

(۳) مستتر در جدول موجودیت C $C(\underline{CID}, C_1, DID, AID, B_1, AR_1)$ P.K. F.K. F.K.

(۴) مستتر در جدول موجودیت D $D(\underline{DID}, D_1, CID, AID, B_1, AR_1)$ P.K. F.K. F.K.

(مهندسی IT- دولتی ۹۶)

۲۹- کدام مورد، درست نیست؟

- (۱) می توان از سه ارتباط دوگانی یک ارتباط سه گانی را نتیجه گرفت.
- (۲) موجودیت ضعیف وابسته به یک موجودیت قوی (مانند E₁) می تواند خود با موجودیت قوی دیگری (مانند E₂) رابطه داشته باشد.
- (۳) برای حذف یک ارتباط سه گانی می توان به جای آن یک موجودیت ضعیف وابسته به هر سه موجودیت مشارکت کننده در ارتباط، در نظر گرفت.

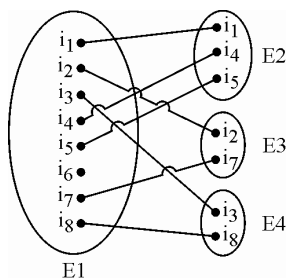
۴) اگر یک نوع موجودیت (مانند E_1) تنها یک صفت داشته باشد و تنها با یک نوع موجودیت دیگر (مانند E_2) در ارتباط باشد، می‌توانیم آن را حذف و به عنوان صفت موجودیت دوم (یعنی E_2) در نظر بگیریم.

۳۰- با توجه به گزاره‌های زیر کدام مورد درست است؟ (مهندسی کامپیوتر-دولتی ۹۷)

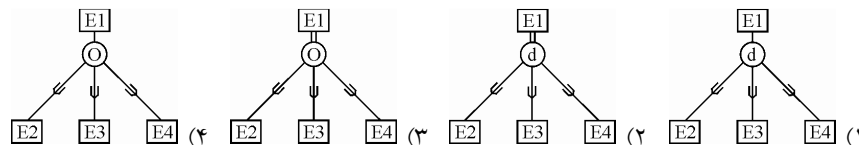
- (a) مدل مفهومی پایگاه داده، استقلال داده‌ای را افزایش می‌دهد.
 (b) اگر K_1 و K_2 ابرکلیدهای رابطه R باشند، آنگاه $K_1 \cap K_2$ یک ابرکلید R است.
 (c) برای اتصال به پایگاه داده، در $Connection String$ نحوه احراز هویت کاربر مشخص می‌شود.
 (d) در رابطه R با تعداد خصیصه بزرگتر از یک، تعداد ابرکلیدها همواره از تعداد کلیدهای کاندید بیشتر است.
- (۱) a درست، b درست (۲) a درست، c درست (۳) d درست، b نادرست (۴) b نادرست، c نادرست

۳۱- با توجه به گزاره‌های زیر، کدام مورد درست است؟ (مهندسی کامپیوتر-دولتی ۹۷)

- (a) اگر رابطه R دارای $n=2k$ خصیصه باشد، آنگاه تعداد کلیدهای کاندید آن حداقل یک و حداکثر $\binom{n}{k}$ است.
 (b) اگر رابطه R دارای n خصیصه باشد، آنگاه تعداد ابرکلیدهای این رابطه حداکثر $\binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{n}$ است.
 (c) تعداد ابرکلیدهای یک رابطه همواره بیشتر از تعداد کلیدهای کاندید آن رابطه است.
 (d) هر صفت مرکب، لزوماً تک مقداری است.
- (۱) یک گزاره نادرست است. (۲) دو گزاره نادرست است.
 (۳) سه گزاره نادرست است. (۴) چهار گزاره نادرست است.



۳۲- نمونه‌هایی از چهار نوع موجودیت E_1, E_2, E_3 و E_4 در شکل مقابل نمایش داده شده است. کدام مورد بهترین نمودار EER معرف محیط است؟ (مهندسی کامپیوتر-دولتی ۹۸)



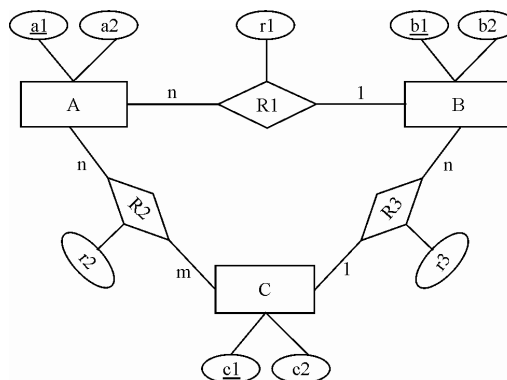
۳۳- در نمودار $E-R$ ، اگر رابطه $is-A$ از نوع منفصل و کامل باشد، کدام مورد برای تبدیل نمودار به مدل رابطه‌ای مناسب‌تر است؟ (مهندسی IT-دولتی ۹۸)

(۱) کلید اصلی موجودیت پدر را به عنوان کلید خارجی به موجودیت‌های فرزند اضافه می‌کنیم.

۲) کلید اصلی موجودیت‌های فرزند را به عنوان کلید خارجی به موجودیت پدر اضافه می‌کنیم.
 ۳) برای موجودیت پدر، رابطه جداگانه ایجاد نمی‌کنیم و ویژگی‌های رابطه پدر را به موجودیت‌های فرزند اضافه می‌کنیم.
 ۴) موجودیت جداگانه‌ای ایجاد می‌شود که شامل کلید اصلی موجودیت پدر و کلید اصلی موجودیت‌های فرزند است.

(مهندسی IT - دولتی ۹۸)

۳۴- مدل رابطه‌ای متناظر با نمودار ER زیر کدام است؟



۱) $A(\underline{a1}, a2) B(\underline{b1}, b2) C(\underline{c1}, c2) R1(\underline{a1}, \underline{b1}, r1) R2(\underline{a1}, \underline{c1}, r2) R3(\underline{b1}, \underline{c1}, r3)$

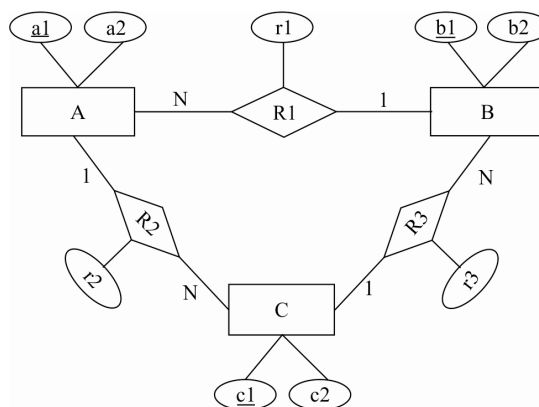
۲) $A(\underline{a1}, a2) B(\underline{b1}, b2, a1, r1) C(\underline{c1}, c2, b1, r3) R1(a1, b1, r1) R3(b1, c1, r3)$

۳) $A(\underline{a1}, a2, b1) B(\underline{b1}, b2, c1, r1) C(\underline{c1}, c2, r3) R2(\underline{a1}, \underline{c1}, r2)$

۴) $A(\underline{a1}, a2, b1, r1) B(\underline{b1}, b2, c1, r3) C(\underline{c1}, c2) R2(\underline{a1}, \underline{c1}, r2)$

(مهندسی کامپیوتر - دولتی ۹۹)

۳۵- مدل رابطه‌ای متناظر با نمودار ER زیر کدام است؟



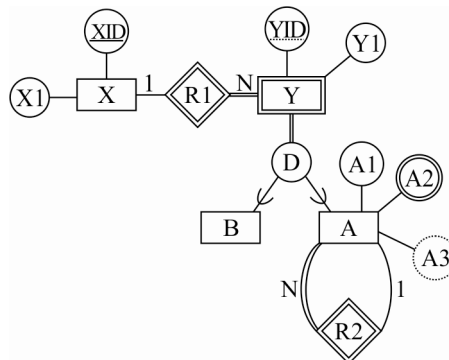
۱) $A(\underline{a1}, a2, b1, r1) B(\underline{b1}, b2, c1, r3) C(\underline{c1}, c2, a1, r2)$

۲) $A(\underline{a1}, a2, b1, r1) B(\underline{b1}, b2, c1, r3) C(\underline{c1}, c2, a1, r2)$

$$A(a_1, a_2) \quad B(b_1, b_2, a_1, \eta) \quad C(c_1, c_2, b_1, r_3) \quad R_1(a_1, b_1, \eta) \quad R_3(b_1, c_1, r_3) \quad (۳)$$

$$A(a_1, a_2) \quad B(b_1, b_2) \quad C(c_1, c_2) \quad R_1(a_1, b_1, \eta) \quad R_2(a_1, c_1, r_2) \quad R_3(b_1, c_1, r_3) \quad (۴)$$

۳۶- در نمودار EER زیر، طراحی منطقی رابطه یا رابطه‌های مربوط به موجودیت A، کدام مورد خواهد بود؟
 (در گزینه‌های داده شده، زیر خط ممتد به معنای کلید اصلی و خط چین به معنای کلید خارجی است.)
 (مهندسی کامپیوتر- دولتی ۹۹)

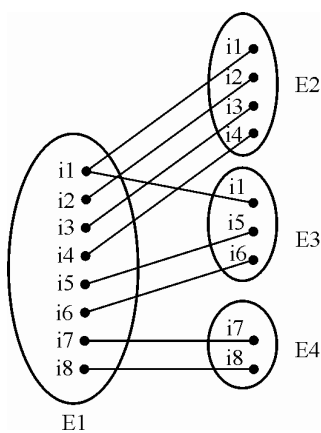


- (۱) $A(\underline{YID}, \underline{YIDR2}, A1, A2, A3)$
 (۲) $A(\underline{YID}, \underline{YIDR2}, A1, A2)$
 (۳) $AA2(\underline{YID}, A2) \quad A(\underline{YID}, Y1, \underline{YIDR2}, A1)$
 (۴) $AA2(\underline{XID}, \underline{YID}, A2) \quad A(\underline{XID}, YID, Y1, \underline{XIDR}, \underline{YIDR2}, A1)$

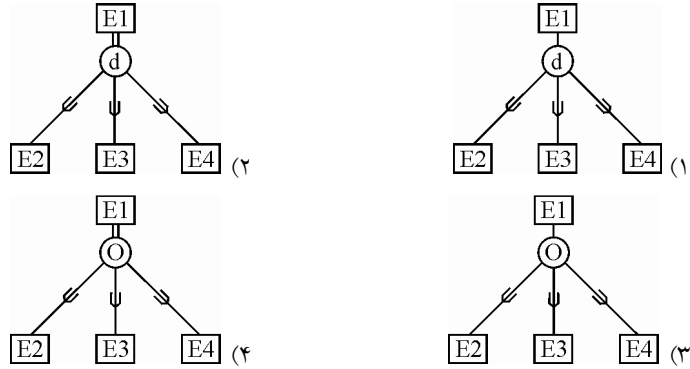
۳۷- چه تعداد از گزاره‌های زیر درست است؟ (مهندسی کامپیوتر-دولتی ۹۹)

- گزاره اول: تعداد کلیدهای کاندید یک رابطه از تعداد سوپر کلیدهای آن رابطه همواره کمتر است.
 گزاره دوم: کلید خارجی یک رابطه، می‌تواند جزئی از کلید اصلی آن باشد.
 گزاره سوم: استفاده از View می‌تواند استقلال داده‌ای را افزایش دهد.
 گزاره چهارم: مدیریت تراکنش‌های همروند به عهده DBMS است.

- (۱) یک گزاره (۲) دو گزاره (۳) سه گزاره (۴) چهار گزاره



۳۸- نمونه‌هایی از چهار موجودیت E1, E2, E3 و E4 در شکل مقابل نمایش داده شده است. کدام گزینه بهترین نمودار EER معرف محیط است؟ (مهندسی IT- دولتی ۹۹)



پاسخ تست‌های فصل سوم: مدل رابطه‌ای

۱- گزینه (۳) صحیح است.

کلید باید خاصیت یکتایی در جدول داشته باشد.
 در ستون C_1 مقدار S و T بیش از یک بار تکرار شده است.
 در ستون C_2 مقدار B بیش از یک بار تکرار شده است.
 در ستون C_3 مقدار D_1 و D_2 بیش از یک بار تکرار شده است.
 بنابراین هیچ یک از ستون‌های C_1 ، C_2 و C_3 خاصیت یکتایی ندارند.
 ترکیب ستون‌های $\{C_2, C_3\}$ خاصیت یکتایی ندارد زیرا سطر BD_1 بیش از یک بار تکرار شده است.
 ترکیب ستون‌های $\{C_1, C_3\}$ خاصیت یکتایی دارد. بنابراین گزینه سوم درست است.

۲- گزینه (۲) صحیح است.

کلید کاندید نمی‌تواند مقدار NULL داشته باشد. هم‌چنین اگر منظور از کلید جستجو همان کلید اصلی باشد، این کلید هم نمی‌تواند مقدار NULL داشته باشد. اما طبق تعریف، کلید خارجی می‌تواند مقدار NULL داشته باشد.

۳- گزینه (۱) صحیح است.

راه‌حل اول: با کنار گذاشتن کلید اصلی و کلید کاندید، $(n-2)$ صفت باقی می‌ماند که تعداد حالات کنار هم قرار گرفتن آن‌ها برابر است با:

$$2^{n-2} \text{ (عدد 2 در پایه به معنی وجود یا عدم وجود یک صفت است)}$$

هر یک از حالات بالا با کلید اصلی یا کلید کاندید یا هر دو کلید، ابرکلید را ایجاد می‌کند. بنابراین داریم:

$$3 \times 2^{n-2}$$

راه‌حل دوم: در این رابطه تعداد کل ابرکلیدها برابر است با:

$$\left[\binom{2}{1} + \binom{2}{2} \right] \times \left[\binom{n-2}{0} + \binom{n-2}{1} + \binom{n-2}{2} + \dots + \binom{n-2}{m} \right]$$

$$(2^2 - 1) \times 2^{n-2} = 3 \times 2^{n-2}$$

زیرا در ابرکلیدها حتماً حداقل یکی از کلیدها (کلید اصلی یا کلید کاندید) یا هر دو حضور دارد.

$$\left[\binom{2}{1} + \binom{2}{2} \right]$$

و از $n-2$ ستون باقی مانده هر زیر مجموعه‌ای (حتی تهی) را می‌توان به ترکیب کلیدها (کلید اصلی یا کلید کاندید یا هر دو کلید) اضافه کرد.

راه‌حل سوم:

مثال: جدول زیر با یک کلید اصلی و یک کلید کاندید هر کدام با یک صفت در نظر بگیرید:

این جدول چند ابرکلید دارد؟

کلید اصلی	کلید کاندید	صفات غیرکلیدی			
A	B	C	D	E	

ابركليد از حاصل جمع كليد كانديد يا كليد اصلي يا هر دو به علاوه صفات غيركليدي ايجاد مي‌گردد.
 [صفات غيركليد] + [كليد كانديد و كليد اصلي] يا [كليد كانديد] يا [كليد اصلي] = ابركليد
توجه: كليد كانديد يا كليد اصلي يا هر دو در ساخت ابركليد نمي‌تواند تهی باشد.
توجه: صفات غيركليد حتي مي‌تواند تهی باشد.
 با توجه به مطالب فوق داريم:

كليد اصلي		كليد كانديد		
A		B		
0	0	0	0	ϕ
0	0	0	1	B
0	1	0	0	A
0	1	0	1	AB

$\Rightarrow 2^2 - 1 = 3$

صفات غير كليد			
C	D	E	
0	0	0	ϕ
0	0	1	E
0	1	0	D
0	1	1	DE
1	0	0	C
1	0	1	CE
1	1	0	CD
1	1	1	CDE

$\Rightarrow 2^3 = 8$

بنابراين تعداد ابركليدها برابر حاصل ضرب دو مقدار فوق مي‌باشد. $3 \times 8 = 24$ هم‌چنين در يك رابطه‌ي كلي خواهيم داشت:

$$(2^2 - 1) \times 2^{n-2} \Rightarrow 3 \times 2^{n-2}$$

۴- گزینه (۴) صحيح است.

دو حل براي نگاهت رابطه ارث‌بري (IS-A) به مدل رابطه‌اي وجود دارد:

۱- تمام فيلدهاي پدر (EMP) در جدول‌هاي فرزند Contract-EMP و Hour-EMP اضافه شود. در اين حالت ديگر نيازي به جدول EMP نيست.

۲- به ازاي هر يك از موجوديت‌هاي پدر و فرزندان يك جدول در مدل رابطه‌اي تعريف شود. سپس فيلدهاي مشترك فرزندان در جدول EMP ذخيره شود و كليد اصلي جدول EMP در جدول‌هاي Contract-EMP و Hour-EMP به عنوان كليد خارجي اضافه شود.

البته گزاره بيان شده در گزینه سوم بايد كمی اصلاح شود. به اين ترتيب كه ضرورتي ندارد كليد اصلي EMP در فرزندان به عنوان كليد اصلي تعريف شود بلكه كافي است به صورت كليد خارجي تعريف شود.

۵- گزینه (۱) صحيح است.

در رابطه IS-A معمولاً صفات مشترك بين چند رابطه را يافته و به عنوان پدر آن روابط در نظر مي‌گيرند. بنابراين چون تمام اطلاعات رابطه پدر براي فرزندان لازم است، بهترين پياده‌سازي با حداقل تکرار حالتی

است که کلید اصلی رابطه پدر را به عنوان کلید خارجی برای کلیه روابط فرزند در نظر بگیریم. گزینه دوم: موجودیت پل برای ارتباط چند به چند کاربرد دارد. در اینجا ارتباط جدول پدر و جدول فرزند، ارتباطی یک به چند است. گزینه سوم: افزونگی دارد. گزینه چهارم: از آنجا که به ازای یک سطر از موجودیت پدر، چند سطر در موجودیت فرزند وجود دارد. بنابراین، این گزینه نادرست است.

۶- گزینه (۳) صحیح است.

در جدول S، S# کلید اصلی و در جدول SP، S# کلید خارجی است. در جدول SP، S# نمی‌تواند کلید اصلی باشد. چون دارای تکرار است، سایر صفات یعنی Pname و QTY نیز نمی‌تواند به تنهایی کلید اصلی باشند. در نتیجه (S#، Pname) کلید اصلی است. در جدول SP برای S2 فیلد Pname، NULL می‌باشد. طبق قاعده جامعیت موجودیت هیچ جزء تشکیل‌دهنده کلید اصلی نباید مقدار NULL داشته باشد. بنابراین در فیلد Pname که جزء کلید می‌باشد قاعده جامعیت موجودیت نقض شده است.

۷- گزینه (۴) صحیح است.

از آنجا که رابطه (جدول) بر پایه جبر مجموعه‌ها در ریاضی می‌باشد، بنابراین رابطه (جدول) دارای خواص مجموعه می‌باشد. در مجموعه عضو تکراری و نظم اعظا وجود ندارد، پس در رابطه عضو تکراری و نظم در سطر (تاپل) و ستون (خصیصه) معنا ندارد. خصیصه‌های رابطه (جدول) باید اتومیک (غیرقابل تجزیه) باشد، یعنی صفت چند مقداری (مانند مدارک استاد) و مرکب (مانند آدرس) نباشند.

۸- گزینه (۱) صحیح است.

در مدل رابطه‌ای جداول می‌توانند دارای روابط یک به یک، یک به چند و چند به چند باشند. در مدل رابطه‌ای جدول می‌تواند تهی (بدون سطر) باشد. در مدل رابطه‌ای صفات باید اتومیک (غیرقابل تجزیه) باشند، یعنی چند مقداری (مانند مدارک استاد) و مرکب (مانند آدرس) نباشند.

۹- گزینه (۱) صحیح است.

طبق تعریف، کلید خارجی برای برقراری ارتباط بین جداول مورد استفاده قرار می‌گیرد (گزینه چهارم) و می‌تواند مقدار NULL داشته باشد (گزینه دوم)، همچنین کلید خارجی باید به کلید کاندید یک جدول مرجع ارجاع نماید (گزینه سوم) توجه کنید که گزینه سوم، گزاره کامل نیست و باید به شکل «در جدول دیگری کلید کاندید (اصلی یا فرعی)» اصلاح گردد. الزامی بر کلید کاندید بودن کلید خارجی در یک جدول وجود ندارد.

۱۰- گزینه (۴) صحیح است.

کلید خارجی باید به کلید کاندید همان رابطه یا رابطه دیگر ارجاع کند، در واقع لزومی ندارد روابط فرضی R1 و R2 از هم مجزا باشند تا کلید خارجی در یک رابطه باشد و کلید کاندید در رابطه‌ای دیگر، زیرا هم کلید کاندید و هم کلید خارجی می‌توانند در یک رابطه باشند، یعنی R1 و R2 می‌توانند یک رابطه باشند،

بنابراین کلید خارجی یک رابطه می‌تواند متناظر با کلید کاندید همان رابطه باشد. در واقع ممکن است دیاگرام ارجاعی دارای طوقه باشد به این معنی که مبدا و مقصد کلید خارجی با هم یکی باشد.

شماره کارمندی	نام	رئیس
1	A	1
2	B	1
3	C	1
4	D	3
5	E	2
6	F	5
کلید کاندید		کلید خارجی

مثال: نام رئیس D چیست؟

شماره رئیس D برابر 3 است و نام شماره 3 برابر C است.

پس گزینه‌های اول و دوم نادرست هستند و گزینه چهارم درست است.

مجموعه‌ای از صفات رابطه R_2 را کلید خارجی این رابطه می‌گویند هرگاه در همان رابطه یا رابطه دیگر مثل R_1 کلید کاندید (اصلی یا فرعی) باشد، بنابراین هیچ الزامی برای این‌که کلید خارجی ترکیبی نباشد و ساده باشد وجود ندارد، بنابراین کلید خارجی می‌تواند یک کلید ترکیبی باشد یعنی بیش از یک جزء داشته باشد، لذا اگر کلید خارجی ترکیبی باشد حتماً کلید کاندید متناظر نیز ترکیبی است.

توجه کنید که اگر کلید خارجی ترکیبی باشد (می‌تواند بیش از یک جزء داشته باشد) یا تماماً NULL است یا تماماً غیر NULL به عبارت دیگر هیچ‌گاه جزئی از یک کلید خارجی نمی‌تواند مقدار NULL بگیرد. در گزینه اول آمده است که کلید خارجی یکی از کلیدهای کاندید همان رابطه است، در حالی که کلید اصلی یکی از کلیدهای کاندید همان رابطه است، بنابراین گزینه اول، نادرست است.

۱۱- گزینه (۳) صحیح است.

طبق تعریف، کلید خارجی می‌تواند مقدار NULL داشته باشد. همچنین کلید خارجی یک رابطه می‌تواند به کلید کاندید همان رابطه ارجاع نماید، بنابراین الزام ذکر شده در گزینه دوم نادرست است. مطابق قانون جامعیت ارجاعی برای برقرار بودن جامعیت ارجاع باید به ازای هر مقدار برای کلید خارجی یک مقدار برای کلید کاندید در جدول مرجع وجود داشته باشد اما نه بالعکس. بنابراین گزینه سوم درست و گزینه چهارم نادرست است.

۱۲- گزینه (۲) صحیح است.

مطابق قانون جامعیت ارجاعی هیچ‌گاه کلید خارجی نباید ارجاع NULL داشته باشد، بنابراین مقادیر موجود در کلید خارجی باید زیر مجموعه‌ای از مقادیر کلید کاندید در جدول مرجع باشد.

۱۳- گزینه (۲) صحیح است.

طبق تعریف، کلید خارجی وظیفه برقراری ارتباط بین جداول را دارد و باید به کلید کاندید (اصلی یا فرعی) جدول مرجع ارجاع نماید. کلید خارجی به عنوان ستون مشترک در پیوند جداول مورد استفاده قرار می‌گیرد. بنابراین گزینه‌های اول، سوم و چهارم گزاره‌های درست را بیان می‌کنند.

البته در مورد گزینه اول باید تذکر داده شود که کلید خارجی در یک رابطه کلید کاندید (اصلی یا فرعی) است و ممکن است در همان رابطه به عنوان قلم اطلاعاتی ساده تکرار شود (کلید خارجی یک جدول می‌تواند به کلید کاندید همان جدول نیز ارجاع نماید) به نظر می‌رسد که این مطلب در طرح سؤال مورد توجه قرار نگرفته است.

گزینه دوم نادرست است، زیرا کلید اصلی یکی از کلیدهای کاندید می‌باشد و نه کلید خارجی. البته از یک منظر دیگر می‌توان این‌گونه بیان کرد که کلید خارجی در جدول خود می‌تواند کلید کاندید باشد ولی نه لزوماً.

مثال:

کلید اصلی کد ملی	کلید فرعی شماره دانشجویی	کلید خارجی شماره دانشجویی	نام خانوادگی	نام
A		B		

شماره دانشجویی در جدول B کلید اصلی (کلید کاندید) و هم‌چنین کلید خارجی برای ارتباط با جدول A است. صفت شماره دانشجویی در جدول A کلید فرعی (کلید کاندید) است.

۱۴- گزینه (۴) صحیح است.

طبق تعریف کلید اصلی و کلید جانشین (فرعی یا جایگزین) از بین کلیدهای کاندید انتخاب می‌شوند، بنابراین همه ویژگی‌های آن را هم دارا هستند. کلیدهای اصلی، جانشین و کاندید نباید مقدار تکراری داشته باشند و هم‌چنین هیچ بخشی از آن‌ها نباید NULL باشد. اما هیچ یک از این دو محدودیت، در مورد کلید خارجی صدق نمی‌کند.

۱۵- گزینه (۲) صحیح است.

جامعیت دامنه‌ای به این معنی است که مقدار هر فیلد باید در دامنه‌ی آن فیلد باشد. مثلاً اگر فیلدی از نوع integer تعریف شده باشد نمی‌تواند مقادیر اعشاری به خود بگیرد. عبارت نوشته شده در گزینه‌ی دوم، تعریف جامعیت ارجاع است نه جامعیت دامنه‌ای.

۱۶- گزینه (۱) صحیح است.

برای تعیین کلیدهای یک جدول باید به معنی صفات در دنیای خارج توجه کرد نه مقادیر فعلی آن‌ها در جدول، هم‌چنین کلید اصلی باید دارای قدرت تفکیک باشد و مقادیر تکراری نداشته باشد. توجه کنید که کلید اصلی از بین کلیدهای کاندید انتخاب می‌شود و نه از بین ابرکلیدها.

۱۷- گزینه (۳) صحیح است.

ستون R₂ در جدول r یک کلید خارجی است به جدول s.

برای این‌که ستونی از یک جدول بتواند کلید خارجی به جدول دیگر باشد باید نوع آن با نوع کلید اصلی جدول دیگر برابر باشد. هم‌چنین کلید خارجی یک رابطه باید در رابطه دیگر کلید کاندید (کلید اصلی یا فرعی) باشد. کلید کاندید (اصلی) رابطه S خصیصه R₂ است بنابراین خصیصه R₂ در رابطه r کلید خارجی آن است.

۱۸- گزینه (۲) صحیح است.

خاصیت consistency (از خواص ACID) تضمین می‌کند که هر تراکنش، بانک اطلاعاتی را از یک وضعیت صحیح به وضعیت صحیح دیگر منتقل می‌کند. برای صحیح بودن وضعیت بانک اطلاعاتی، داده‌های نوشته شده در جداول باید منطبق بر تمامی قواعد تعیین شده (مانند کلیدهای اصلی و خارجی، کلیدهای فرعی، Constraint ها و trigger ها) باشند. با توجه به این توضیحات، گزینه‌های اول و چهارم مرتبط با خاصیت سازگاری هستند.

هم‌چنین، برای تضمین سازگاری، اگر تراکنشی قواعد تعیین شده برای سازگاری را رعایت نکرده باشد. باید آن تراکنش rollback شود و بانک اطلاعاتی ترمیم (recover) گردد. پس گزینه سوم نیز مرتبط با سازگاری است. اما گزینه دوم مربوط به خاصیت‌های یکپارچگی و انزوا است.

به طور کلی برقراری کلیه قوانین جامعیت بانک اطلاعات، توسط بخش Consistency کنترل می‌گردد.

۱۹- گزینه (۱) صحیح است.

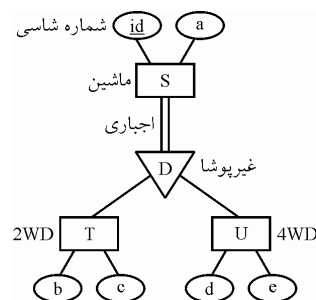
مدل EER رسم شده، یک رابطه ارث‌بری total (کامل) و disjoint (تخصیص مجزا) را نشان می‌دهد (ارتباط دو خطی بین A و S نشان‌دهنده total بودن است) در یک رابطه ارث‌بری total، هر عضو از کلاس پدر باید حتماً متعلق به یکی از دو کلاس فرزند باشد. برای مثال، در این سؤال، نمی‌توان شیء‌ای از کلاس S داشت که در هیچ یک از کلاس‌های U و T نباشد. هم‌چنین در یک رابطه ارث‌بری disjoint، کلاس‌های فرزند نمی‌توانند شیء مشترکی داشته باشند (یا به عبارت دیگر، یک شیء از کلاس پدر نمی‌تواند از نوع هر دو فرزند باشد). یک ماشین با یک شماره شاسی خاص نمی‌تواند هم دو درب باشد و هم چهار درب یا یک شماره دانشجویی خاص نمی‌تواند هم دانشجوی کارشناسی باشد و هم کارشناسی ارشد.

با توجه به شرایط total بودن، نیازی نیست جدولی جداگانه برای S داشته باشیم (مثل گزینه دوم) و با توجه به شرایط disjoint بودن نیازی نیست یک جدول برای نگهداری تمام رکودها داشته باشیم (مثل گزینه‌های سوم و چهارم) چون رکوردهای اشتراکی بین کلاس‌ها نداریم.

به طور کلی در شرایط total و disjoint بودن رابطه ارث‌بری، باید برای هر یک از فرزندان جدولی جدا در نظر گرفت که صفات پدر را هم در خود دارند و نیازی به جدولی برای پدر نیست (مشابه گزینه اول).

مثال: رابطه ISA بر دو نوع پوشا (Overlap) و غیر پوشا (Disjoint) می‌باشد، مرسوم است که اگر رابطه پوشا باشد، داخل مثلث O و اگر غیر پوشا باشد داخل مثلث D قرار بگیرد. اگر ماشینی باشد، حتماً یک شماره شاسی دارد که حتماً برای آن باید یک ماشین باشد. یعنی شماره شاسی با نوع ماشین رابطه اجباری (Total) دارد. اما یک ماشین با یک شماره شاسی خاص نمی‌تواند هم دو محور باشد (2WD) و هم چهار محور (4WD) باشد. این یعنی غیر پوشایی (Disjoint) بین انواع ماشین‌ها.

مدل تحلیل (نمودار (ISA))



مدل طراحی (مدل رابطه‌ای)

توجه: عمل نگاشت، نمودار ISA از فاز تحلیل به مدل رابطه‌ای در فاز طراحی به صورت زیر است:

id	a	b	c	id	a	d	E
1	X	Y	Z	2	M	P	Q

جدول T

جدول U

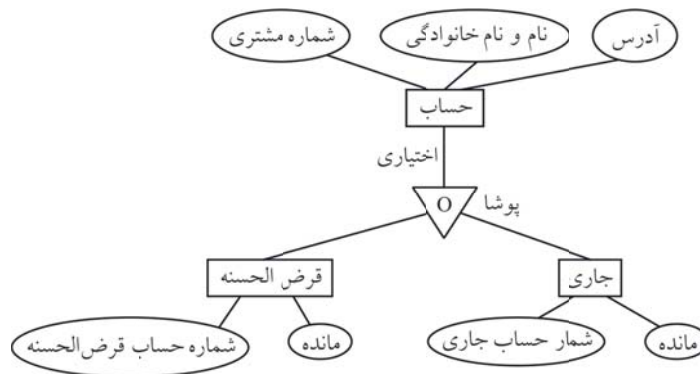
توجه: چون رابطه ماشین با نوع ماشین اجباری است، و به ازای هر شماره شاسی حتماً یک ماشین وجود دارد و به تبع عدم مقادیر NULL جلوی شماره شاسی و عدم نیاز به عمل الحاق، نگاشت فوق به عنوان یک تبدیل ایده‌آل توصیه می‌گردد.

اگر شماره مشتری در یک بانک باشد، ممکن است برای آن حسابی (قرض الحسنه یا جاری) موجود نباشد، البته اگر نظر طراح بانک اطلاعات بر این باشد که شماره مشتری‌های فاقد حساب در بانک اطلاعات باقی بمانند.

توجه: در بانکداری ایران هیچ‌گاه شماره مشتری‌ها (حتی فاقد حساب) پاک نمی‌شوند، هم‌چنین حتی پس از بستن حساب از آن‌جا که شماره مشتری‌ها باقی می‌مانند، تمامی تراکنش‌های انجام شده، همچنان موجود و قابل دسترسی برای بانکداران می‌باشد.

در این شرایط شماره مشتری با انواع حساب (قرض الحسنه یا جاری) رابطه اختیاری (Partial) دارد. اما اگر یک شماره مشتری بخواهد صاحب حساب هم باشد، می‌تواند چندین نوع حساب (قرض الحسنه یا جاری) داشته باشد. این یعنی پوشایی (Overlap) بین انواع حساب‌ها.

۱- مدل تحلیل (نمودار ISA)



۲- مدل طراحی (مدل رابطه‌ای)

توجه: عمل نگاشت، نمودار ISA از فاز تحلیل به مدل رابطه‌ای در فاز طراحی به صورت زیر است:

شماره مشتری	نام و نام خانوادگی	آدرس

جدول حساب

مانده	شماره حساب جاری	شماره مشتری	مانده	شماره حساب قرض الحسنه	شماره مشتری
-------	-----------------	-------------	-------	-----------------------	-------------

جدول حساب قرض الحسنه

جدول حساب جاری

توجه: چون رابطه حساب با انواع حساب (قرض الحسنه یا جاری) اختیاری است، و به ازای یک شماره مشتری ممکن است حساب قرض الحسنه یا جاری باشد یا نباشد، بنابراین برای جلوگیری از درج مقادیر NULL جلوی شماره مشتری های فاقد حساب، به جای مدل دو جدولی در حالت قبل، مدل سه جدولی فوق اما با تحمل سربار حاصل از عمل الحاق، به عنوان یک تبدیل ایده آل توصیه می گردد.

توجه: در این نوع نگاشت، به ازای هر یک از موجودیت های پدر و فرزند یک جدول در مدل رابطه ای تعریف می شود و کلید اصلی جدول پدر به عنوان کلید خارجی در جداول فرزند تعریف می شود.

توجه: موجودیت حساب به شکل اختیاری در رابطه شرکت می کند، بدین معنی که به ازای هر شماره مشتری حساب، یک شماره حساب قرض الحسنه یا جاری ممکن است موجود باشد یا نباشد. اما موجودیت های جاری و قرض الحسنه به شکل اجباری در رابطه شرکت می کنند، بدین معنی که اگر یک شماره حساب قرض الحسنه یا جاری موجود باشد حتماً به یک شماره مشتری تعلق دارد.

توجه: جداول فوق را می توان به این صورت تفسیر معنایی نمود که، هنگام ایجاد یک حساب قرض الحسنه یا جاری در پایگاه داده، اطلاعات بسته به نوع حساب به دو بخش اطلاعات عمومی و اطلاعات اختصاصی تقسیم می شود. البته اطلاعات عمومی حساب در جدول حساب درج می شود، سپس اطلاعات اختصاصی در یکی از جداول قرض الحسنه یا جاری متناسب با نوع حساب درج می شود و در نهایت دو سطر درج شده در دو جدول پدر و فرزند از طریق شماره مشتری، که کلید اصلی جدول حساب و کلید خارجی جداول قرض الحسنه یا جاری است به یکدیگر مرتبط می شوند. به این ترتیب می توان گفت که اطلاعات عمومی حساب در جدول حساب و اطلاعات اختصاصی حساب قرض الحسنه یا جاری در جداول حساب قرض الحسنه و حساب جاری قرار دارند.

توجه: نگاشت فوق، یک راه حل غیربهمینه و ایده آل نیز دارد، بدین شکل که برای موجودیت پدر، جدول در نظر گرفته نشود و فقط برای موجودیت های فرزند جدول در نظر گرفته شود و تمام صفات پدر در جدول های فرزند درج گردند. این رویکرد، ساده ترین راه حل محسوب می گردد، اما چنانچه گفته شد، راه حل غیربهمینه و دارای تکرار اطلاعات و به تبع افزونگی است.

توجه: عمل نگاشت، نمودار ISA از فاز تحلیل به مدل رابطه ای در فاز طراحی در راه حل غیربهمینه به صورت زیر است:

مانده	شماره حساب جاری	آدرس	نام و نام خانوادگی	شماره مشتری
-------	-----------------	------	--------------------	-------------

جدول حساب جاری

مانده	شماره حساب قرض الحسنه	آدرس	نام و نام خانوادگی	شماره مشتری
-------	-----------------------	------	--------------------	-------------

جدول حساب قرض الحسنه

۲۰- گزینه (۲) صحیح است.

جدول stud (stno, stname, avg) را در نظر بگیرید. stno (شماره دانشجویی) یک کلید کاندید است. فرض کنید stname (اسم دانشجو) منحصر به فرد و در نتیجه یک کلید کاندید باشد. اجتماع این دو می‌شود: (stno, stname) که هر چند ابرکلید است ولی یک کلید کاندید نیست (توجه کنید که کلید کاندید باید کمینه باشد). هم‌چنین اشتراک این دو برابر مجموعه تهی است که نمی‌تواند کلید باشد.

۲۱- گزینه (۱) صحیح است.

اگر از رابطه A به رابطه B کلید خارجی وجود داشته باشد، باید مقادیر صفت‌های کلید خارجی در رابطه A عیناً در رابطه B (به عنوان مقادیر کلید اصلی) وجود داشته باشند. مقادیر کلید خارجی در A زیر مجموعه مقادیر کلید کاندید در B باشد.

۲۲- گزینه (۲) صحیح است.

انواع کلید

ابرکلید (Super key)

هر ترکیبی از صفات (ستون‌ها) که خاصیت کلیدی داشته باشد، یک ابرکلید است. دقت کنید که خاصیت کلیدی بر اساس محیط خارج مشخص می‌گردد و نه محتوی جدول.

تعریف خاصیت کلیدی: اگر بر اساس آن مورد، جستجو انجام شود، فقط و فقط یک مورد را به صورت یکتا برگرداند. در واقع قدرت تفکیک داشته باشد.

مثال: شماره دانشجویی یا شماره ملی

توجه: هر ترکیبی از ابرکلید با سایر ستون‌های جدول مورد نظر نیز یک ابرکلید است. بنابراین ابرکلید می‌تواند عضو زائد داشته باشد و کمینه (minimal) نباشد، پس گزینه دوم پاسخ سؤال است.

مثال:

S#	Sname	city
S ₁	Sn ₁	C ₁
S ₂	Sn ₂	C ₂
S ₃	Sn ₃	C ₂
S ₄	Sn ₄	C ₂

S# : ابرکلید است.

Sname : ابرکلید نیست.

(Sname, City) : ابرکلید نیست.

(S#, Sname) : ابرکلید است.

(S#, City) : ابرکلید است.

مثال: تعداد ابرکلیدهای مثال فوق چند عدد است؟

پاسخ: برابر 4 عدد می‌باشد.

S# +	Sname	City	ابر کلید
0	0	0 →	(S#)
0	1	1 →	(S#, City)
1	0	0 →	(S#, Sname)
1	1	1 →	(S#, Sname, City)

خاصیت

کلیدی دارد

بنابراین ابر

کلید است.

کلید کاندید (Candidate key)

ابرکلیدی که عضو زائد نداشته باشد، کلید کاندید است.

مثال:

S#	Sname	City	S#	P#	QTY	P#	Pname	Color
S ₁	Sn ₁	C ₁	S ₁	P ₁	10	P ₁	Pn ₁	Red
S ₂	Sn ₂	C ₂	S ₁	P ₂	20	P ₂	Pn ₂	Blue
S ₃	Sn ₃	C ₂	S ₂	P ₁	30	P ₃	Pn ₃	Blue

جدول S

جدول SP

جدول P

S# : ابرکلید است. کلید کاندید نیز هست. (در جدول S)
 (S#,Sname) : ابرکلید است، زیرا خاصیت کلیدی دارد، اما کلید کاندید نیست، زیرا عضو زائد Sname را دارد. در واقع صفت S#، به تنهایی خاصیت کلیدی دارد، بنابراین صفت Sname، عضو زائد است. (در جدول S)
 (S#,P#) : ابرکلید است. کلید کاندید نیز هست. (در جدول SP).

مثال:

شماره ملی: ابرکلید است. کلید کاندید نیز هست.
 (شماره ملی و نام خانوادگی): ابرکلید است. زیرا خاصیت کلیدی دارد، اما کلید کاندید نیست، زیرا عضو زائد نام خانوادگی را دارد. در واقع صفت شماره ملی، به تنهایی خاصیت کلیدی دارد، بنابراین صفت نام خانوادگی، عضو زائد است.
 توجه: یک جدول می تواند چندین کلید کاندید داشته باشد.

مثال:

شماره دانشجویی	شماره ملی	نام خانوادگی	نام
کلید کاندید (کلید اصلی)	کلید کاندید (کلید فرعی)		

توجه: در مدل رابطه‌ای، هر رابطه حتماً حداقل یک کلید کاندید دارد، زیرا در بدترین شرایط، همه صفات با هم کلید کاندید می شوند، که به این رابطه تمام کلید (All key) گفته می شود.
 توجه: یک رابطه، تحت هیچ شرایطی نمی تواند به دلیل استفاده از خاصیت مجموعه‌ای بودن، سطر تکراری داشته باشد. بنابراین یک رابطه، حداقل یک کلید کاندید دارد.
 مثال: یک جدول تمام کلید.

a	b	c
1	2	3
1	6	3
1	2	7
8	2	3

کلید اصلی (Primary key)

کلید اصلی، یکی از کلیدهای کاندید است که بر اساس سلیقه طراح پایگاه داده، انتخاب می شود.
 توجه: مطابق قوانین مدل رابطه‌ای، هر جدول حتماً باید یک کلید اصلی داشته باشد.

توجه: می‌توان مقادیر کلید خارجی را برابر NULL قرار داد، به شرطی که قوانین بالا دستی بر کلید خارجی حاکم نباشد. در مثال فوق S# و P# در جدول SP به عنوان کلید خارجی نمی‌توانند NULL باشند، زیرا (S#,P#) با هم قبل از اینکه به تنهایی کلید خارجی باشند، در جدول SP کلید اصلی جدول SP بوده‌اند. بنابراین قانون بالا دستی **جامعیت موجودیت**، بدین معنی که هیچگاه نباید تمام یا بخشی از کلید اصلی NULL باشد، جلوی NULL شدن S# و P# به عنوان کلید خارجی را در جدول SP می‌گیرد.

توجه: مطابق تعریف کلید خارجی، کلید خارجی باید به کلید کاندید همان رابطه یا رابطه دیگر ارجاع کند، در واقع لزومی ندارد که روابط فرضی R₁ و R₂ از هم مجزا باشند تا کلید خارجی در یک رابطه باشد و کلید کاندید در رابطه‌ای دیگر. زیرا هم کلید کاندید و هم کلید خارجی می‌توانند در یک رابطه باشند، یعنی R₁ و R₂ می‌توانند یک رابطه باشند، بنابراین کلید خارجی یک رابطه می‌تواند متناظر با کلید کاندید همان رابطه باشد. در واقع ممکن است، دیاگرام ارجاعی دارای طوقه باشد، به این معنی که مبداء و مقصد کلید خارجی با هم یکی باشد.

مثال: نام رئیس D چیست؟

شماره کارمندی	نام	رئیس
1	A	1
2	B	1
3	C	1
4	D	3
5	E	2
<u>6</u>	F	<u>5</u>

کلید خارجی ← کلید کاندید

طوقه

پاسخ: شماره رئیس D برابر 3 است و نام شماره 3 برابر C است.

توجه: مجموعه‌ای از صفات رابطه R₂ را کلید خارجی این رابطه می‌گویند، هرگاه در همان رابطه یا رابطه دیگر مثل R₁ کلید کاندید (اصلی یا فرعی) باشد، بنابراین هیچ الزامی برای اینکه کلید خارجی ترکیبی نباشد و ساده باشد، وجود ندارد، بنابراین کلید خارجی می‌تواند یک کلید ترکیبی باشد، یعنی بیش از یک صفت داشته باشد، لذا اگر کلید خارجی ترکیبی باشد حتماً کلید کاندید متناظر نیز ترکیبی است.

مثال:

کلید خارجی			کلید کاندید		
a	b	c	b	c	d
5	1	2	1	2	8
6	3	4	3	4	9
4	NULL	NULL	5	6	3
			7	2	4
			1	8	1

توجه: اگر کلید خارجی ترکیبی باشد (یعنی بیش از یک صفت داشته باشد) یا تماماً NULL است یا تماماً

غیر NULL، به عبارت دیگر هیچگاه بخشی از یک کلید خارجی ترکیبی نمی‌تواند مقدار NULL بگیرد.

نکته: برای اینکه یک رابطه، در مدل رابطه‌ای صدق کند، باید دارای دو شرط زیر باشد:

۱- اتومیک باشد.

یعنی صفت چندمقداری و مرکب نباشد. به عبارت دیگر یعنی اینکه هیچ یک از سلول‌های این رابطه دارای بیش از یک مقدار نباشد.

۲- رابطه باید حتماً دارای حداقل یک کلید کاندید باشد.

توجه: از آنجاییکه رابطه در مدل رابطه‌ای یک مجموعه است، در بدترین حالت تمام صفات یک سطر با هم، باید تشکیل یک کلید را بدهد، در غیر این صورت تعریف مجموعه نقض شده است. (مجموعه تاپل تکراری ندارد.)

نکته: یک رابطه در مدل رابطه‌ای دارای حداقل یک کلید کاندید است. اما می‌تواند، بیش از یک کلید کاندید نیز داشته باشد.

۲۳- گزینه (۳) صحیح است.

مطابق قانون جامعیت ارجاعی (Referential Integrity Rule)، هیچگاه نباید کلید خارجی، دچار ارجاع NULL گردد، بنابراین مطابق تعریف کلید خارجی ابتدا می‌بایست از نظر ساختاری، اگر ستون(هایی) در یک جدول به عنوان کلید خارجی تعریف می‌شود باید در همان جدول یا جدول دیگری به عنوان کلید کاندید مطرح باشد. همچنین از نظر محتوایی نیز مقادیر کلید خارجی باید همواره زیر مجموعه مقادیر کلید کاندید باشد که این مورد را DBMS گارانتی می‌کند.

۲۴- گزینه (۲) صحیح است.

مطابق قانون جامعیت درون رابطه‌ای (Intra - Relation Integrity Rule)، هر رابطه (جدول) باید به تنهایی درست باشد.

یعنی صفات رابطه اتومیک باشد (چند مقداری و مرکب نباشد) و همچنین هر رابطه باید حتماً حداقل دارای یک کلید کاندید باشد.

۲۵- گزینه (۱) صحیح است.

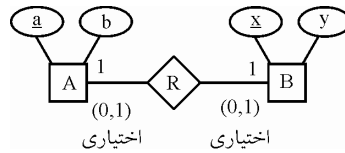
به طور کلی در مدل رابطه‌ای، هر موجودیت شناسایی شده در نمودار ER (فعالیت تحلیل) هنگام نگاشت به مدل رابطه‌ای (فعالیت طراحی) به یک جدول تبدیل می‌شود. همچنین صفات‌های موجودیت پس از نگاشت آن در مدل رابطه‌ای به صورت ستون‌های جدول بیان می‌شوند. همچنین ارتباط بین جدول از طریق کلید خارجی برقرار می‌گردد.

نگاشت رابطه یک به یک بین دو موجودیت به مدل رابطه‌ای

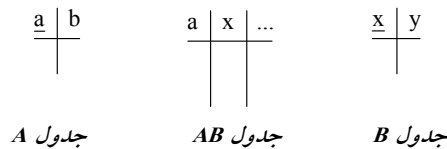
حالت اول: اگر هر دو طرف رابطه اختیاری باشد. هر موجودیت به یک جدول تبدیل می‌گردد. و یک جدول ارتباط نیز به عنوان ارتباط دهنده دو جدول مورد استفاده قرار می‌گیرد. همچنین کلید کاندید جدول ارتباط یکی از کلیدهای دو جدول دیگر می‌باشد.

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



مدل طراحی:



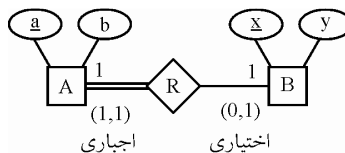
توجه: کلید کاندید جدول AB، ستون a یا ستون x می‌باشد. زیرا هر یک به تنهایی قدرت تفکیک سطرها را دارند. ترکیب ستون‌های a و x با هم به عنوان ax، ابرکلید ایجاد می‌کند.

توجه: ستون a در جدول AB به عنوان کلید خارجی تعریف می‌گردد که به جدول A ارجاع می‌کند. همچنین ستون x در جدول AB به عنوان کلید خارجی تعریف می‌گردد که به جدول B ارجاع می‌کند.

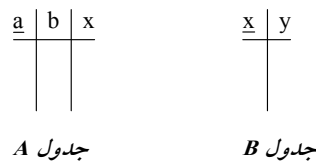
حالت دوم: اگر یک طرف رابطه اختیاری و طرف دیگر رابطه اجباری باشد، هر موجودیت به یک جدول تبدیل می‌گردد. و کلید کاندید جدول موجودیت اختیاری در جدول موجودیت اجباری به عنوان کلید خارجی تعریف می‌گردد.

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



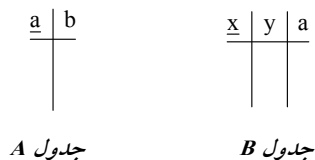
مدل طراحی:



توجه: این نگاشت منطقی هم هست، هرگاه سطری در جدول موجودیت اختیاری وجود داشت، توسط کلید خارجی به جدول موجودیت اجباری می‌تواند رجوع کند.

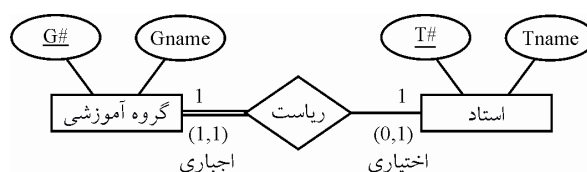
توجه: اگر بالعکس نگاشت فوق عمل کنیم و کلید کاندید جدول موجودیت اجباری را در جدول موجودیت اختیاری به عنوان کلید خارجی تعریف کنیم، آنگاه جدول موجودیت اختیاری با مقادیر زیاد NULL مواجه می‌گردد.

مدل طراحی نادرست:



مثال:

مدل تحلیل:



توجه: رابطه ریاست بین استاد و گروه آموزشی به صورت فوق است، چون هر گروه آموزشی حتماً می‌بایست داشته باشد، اما یک استاد ممکن است رئیس یک گروه آموزشی باشد یا نباشد.

مدل طراحی:

<u>G#</u>	Gname	T#
G ₁	Gn1	T ₁
G ₂	Gn2	T ₂

<u>T#</u>	Tname
T ₁	Tn1
T ₂	Tn2
T ₃	Tn3
T ₄	Tn4

جدول گروه آموزشی

جدول استاد

بنابراین تا همین جا واضح است که گزینه اول درست است.

مدل طراحی نادرست:

<u>G#</u>	Gname
G ₁	Gn1
G ₂	Gn2

<u>T#</u>	Tname	G#
T ₁	Tn1	G1
T ₂	Tn2	G2
T ₃	Tn3	NULL
T ₄	Tn4	NULL

جدول گروه آموزشی

جدول استاد

توجه: مقادیر NULL در ستون G# قابل مشاهده است، زیرا بسیاری از اساتید، رئیس هیچ گروه آموزشی نیستند!

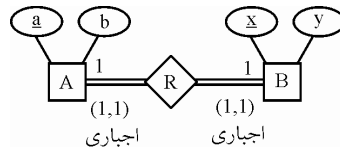
بنابراین گزینه‌های دوم، سوم و چهارم نادرست هستند.

حالت سوم: اگر هر دو طرف رابطه اجباری باشد، هر دو موجودیت در قالب یک جدول قرار می‌گیرند. و

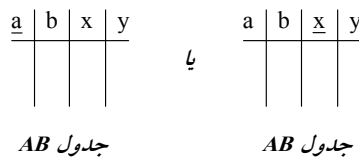
کلید کاندید جدول حاصل، کلید کاندید هر یک از موجودیت‌ها می‌تواند باشد.

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



مدل طراحی:



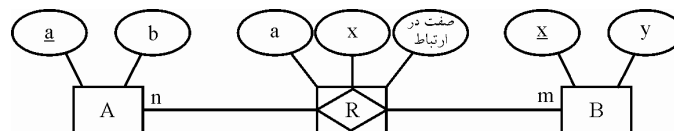
توجه: در مواقع اختیاری، با ایجاد جداول دیگر، از تکرار بی‌رویه مقادیر NULL، جلوگیری می‌کنیم، ولی در موارد اجباری مانند حالت سوم، ایجاد جداول دیگر، معنا ندارد.

نگاشت رابطه چند به چند بین دو موجودیت به مدل رابطه‌ای

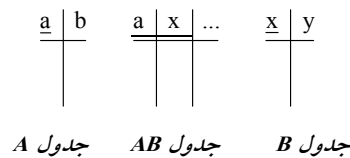
مستقل از اختیاری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد و یک جدول پل (Bridge) نیز به عنوان ارتباط دهنده دو جدول مورد استفاده قرار می‌گیرد. همچنین کلید کاندید جدول پل از ترکیب کلید کاندید دو جدول دیگر ایجاد می‌گردد.

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



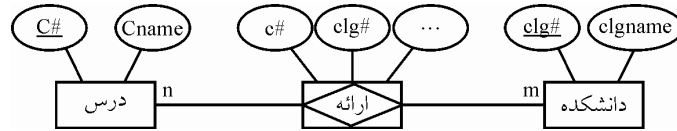
مدل طراحی:



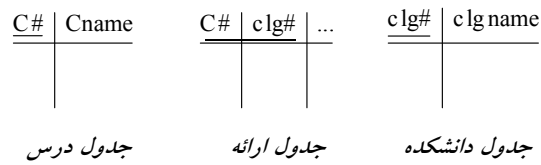
توجه: ستون a در جدول AB به عنوان کلید خارجی تعریف می‌گردد که به جدول A ارجاع می‌کند. همچنین ستون x در جدول AB به عنوان کلید خارجی تعریف می‌گردد که به جدول B ارجاع می‌کند.

مثال:

مدل تحلیل:



مدل طراحی:



توجه: مثال فوق مطابق گزینه اول است.

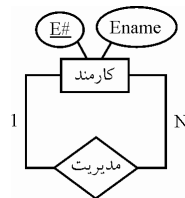
نگاشت رابطه یک به چند در یک موجودیت به مدل رابطه‌ای

مستقل از اختیاری یا اجباری بودن موجودیت، موجودیت به یک جدول تبدیل می‌گردد.

مثال: فرض کنید در یک سازمان هر کارمند، تحت مدیریت یک کارمند دیگر باشد.

توجه: یک رئیس چندین کارمند دارد ولی هر کارمند حداکثر یک رئیس دارد.

مدل تحلیل:



مدل طراحی:

<u>E#</u>	Sname	M#
S1	Sn1	NULL
S2	Sn2	S1
S3	Sn3	S1
S4	Sn4	S3
S5	Sn5	S2
S6	Sn6	<u>S5</u>

کلید خارجی (شماره مدیر)
کلید کاندید (شماره کارمندی)

↑
طوقه

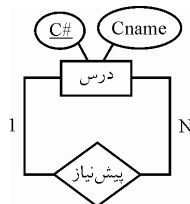
توجه: سطر مربوط به مدیر کل، که مدیر بالاتری ندارد، مقدار NULL در فیلد M# خود می‌گیرد.

توجه: شماره مدیر، برای شخصی با شماره کارمندی S4 برابر S3 است و نام S3 برابر Sn3 است.
توجه: مطابق تعریف کلید خارجی، کلید خارجی باید به کلید کاندید همان رابطه یا رابطه دیگر، ارجاع کند، در واقع لزومی ندارد که روابط فرضی R1 و R2 از هم مجزا باشند، تا کلید خارجی در یک رابطه باشد و کلید کاندید در رابطه‌ای دیگر. زیرا هم کلید کاندید و هم کلید خارجی می‌توانند در یک رابطه باشند، یعنی R1 و R2 می‌توانند یک رابطه باشند، بنابراین کلید خارجی یک رابطه می‌تواند متناظر با کلید کاندید همان رابطه باشد. در واقع دیاگرام ارجاعی دارای طوقه است به این معنی که مبداء و مقصد کلید خارجی با هم یکی است.

مثال: فرض کنید در یک دانشگاه، هر درس، پیش‌نیازهای دیگری دارد.

توجه: در رابطه یک به چند میان درس و پیش‌نیازهای آن، یک درس چندین پیش‌نیاز دارد ولی هر پیش‌نیاز، حداکثر پیش‌نیاز یک درس است.

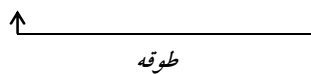
مدل تحلیل:



مدل طراحی:

<u>C#</u>	Cname	<u>CC#</u>
C1	Cn1	NULL
C2	Cn2	C1
C3	Cn3	C1
C4	Cn4	C3
C5	Cn5	C2
C6	Cn6	C5

کلید خارجی (شماره درس) کلید کاندید (پیش‌نیازها)



توجه: C2 و C3 پیش‌نیازهای درس C1 هستند و C6 پیش‌نیاز درس C5 و C1 پیش‌نیاز هیچ درسی نیست.
توجه: مطابق گزینه اول، کلید اصلی درس، به عنوان کلید خارجی در جدول فوق تعریف شده است.

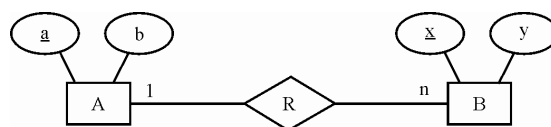
۲۶- گزینه (۲) صحیح است.

به طور کلی در مدل رابطه‌ای، هر موجودیت شناسایی شده در نمودار ER (مدل تحلیل) هنگام نگاشت به مدل رابطه‌ای (مدل طراحی) به یک جدول تبدیل می‌شود. همچنین صفت‌های موجودیت پس از نگاشت آن در مدل رابطه‌ای به صورت ستون‌های جدول بیان می‌شوند. همچنین ارتباط بین جدول از طریق کلید خارجی برقرار می‌گردد.

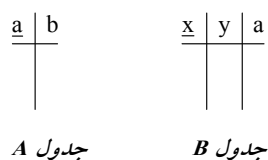
نگاشت رابطه یک به چند بین دو موجودیت به مدل رابطه‌ای مستقل از اختیاری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد. و کلید کاندید جدول یک در جدول چند به عنوان کلید خارجی تعریف می‌گردد.

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



مدل طراحی:



وابستگی وجودی

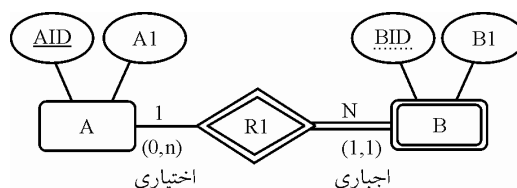
اگر در یک بانک اطلاعاتی، وجود یک موجودیت، وابسته به موجودیت دیگری باشد که در صورت حذف و تغییر موجودیت اصلی یعنی موجودیت قوی این موجودیت نیز تغییر کند، این نوع وابستگی را وابستگی وجودی گفته و به پدیده وابسته، موجودیت ضعیف گویند. همچنین موجودیت ضعیف کلید موجودیت قوی را در بر دارد تا هرگونه تغییر یا حذف در موجودیت قوی به موجودیت ضعیف اعمال شود.

توجه: موجودیت ضعیف با دو مستطیل تو در تو نمایش داده می‌شود.

بخشی از مدل EER رسم شده در صورت سوال، یک رابطه یک به چند بین دو موجودیت قوی و ضعیف را نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را شرح می‌دهیم.

حالت اجباری و اختیاری

مدل تحلیل:



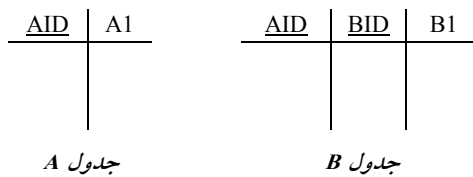
توجه: در شکل فوق صفت کلید موجودیت قوی A و صفت BID، صفت ممیزه موجودیت ضعیف B است.

توجه: نماد خط مضاعف افقی نشانه اجباری بودن موجودیت چسبیده به آن است، اما نماد | به معنی یک و الزام شرکت در رابطه نشانه اجباری بودن موجودیت طرف مقابل است.

توجه: نماد خط افقی نشانه اختیاری بودن موجودیت چسبیده به آن است، اما نماد دایره کوچک توخالی به معنی صفر و عدم الزام شرکت در رابطه نشانه اختیاری بودن موجودیت طرف مقابل است.

توجه: قید (0,N) نشان می‌دهد که هر نمونه موجودیت از A حداقل با صفر و حداکثر با N نمونه موجودیت از B ارتباط دارد و قید (1,1) نشان می‌دهد که هر نمونه موجودیت از B حداقل با یک و حداکثر با یک نمونه موجودیت از A ارتباط دارد.

مدل طراحی:



توجه: کلید کاندید جدول یک یعنی موجودیت قوی در جدول چند یعنی موجودیت ضعیف به عنوان کلید خارجی تعریف می‌گردد.

توجه: کلید کاندید جدول چند یعنی موجودیت ضعیف برابر ترکیب کلید خارجی و صفت ممیزه در جدول موجودیت ضعیف است. یعنی کلید کاندید جدول B برابر (AID,BID) است.

توجه: صفت ممیزه یا کلید جزئی به طور سراسری در یک موجودیت ضعیف یکتا نیست، بلکه فقط در بین نمونه‌ها یا دسته‌هایی که با موجودیت قوی ارتباط دارند، یکتا است.

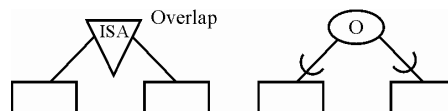
توجه: یک موجودیت ضعیف همیشه در ارتباطش با موجودیت قوی رابطه اجباری دارد.

توجه: همانطور که واضح است، طراحی جدول B در صورت سوال به صورت (BID,B1) در نظر گرفته شده است، که مطابق آنچه بیان کردیم، طراحی درست آن به صورت (AID,BID,B1) است. همچنین طراحی جدول A در صورت سوال به صورت (AID,A1) در نظر گرفته شده است، که مطابق آنچه بیان کردیم، طراحی درست آن هم به صورت (AID,A1) است. بنابراین گزینه‌های اول، سوم و چهارم را کنار می‌گذاریم، پس تا همینجا واضح است که گزینه دوم پاسخ سوال است.

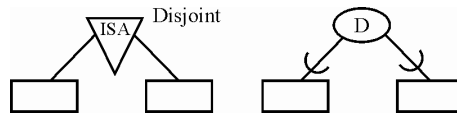
نگاشت رابطه ISA به مدل رابطه‌ای

در رابطه ISA رابطه پدر با فرزندان به دو صورت رابطه اختیاری یا جزئی (Partial) با نماد خط عمودی و رابطه اجباری یا کلی (Total) با نماد خط مضاعف عمودی است و رابطه فرزندان با پدر به دو صورت رابطه پوشا یا تخصیص غیرمجزا (Overlap) و رابطه غیرپوشا یا تخصیص مجزا (Disjoint) می‌باشد.

رابطه پوشا یا تخصیص غیرمجزا (Overlap) مابین فرزندان و پدر به دو شیوه زیر نشان داده می‌شود:



رابطه غیرپوشا یا تخصیص مجزا (Disjoint) مابین فرزندان و پدر به دو شیوه زیر نشان داده می‌شود:



بخشی از مدل EER رسم شده در صورت سوال، یک رابطه اجباری یا کلی (Total) مابین موجودیت B و موجودیت‌های C و D و یک رابطه غیرپوشا یا تخصیص مجزا (Disjoint) را مابین موجودیت‌های C و D و موجودیت B نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را بیان می‌کنیم.

در یک رابطه اجباری یا کلی (Total)، هر نمونه از موجودیت پدر حتماً می‌بایست با یکی از نمونه موجودیت‌های فرزند در ارتباط باشد. برای مثال در این سؤال، هر نمونه از موجودیت B حتماً می‌بایست با یکی از نمونه موجودیت‌های C یا D در ارتباط باشد. به عبارت دیگر نمی‌توان نمونه‌ای از موجودیت B داشت که با هیچ یک از نمونه موجودیت‌های C یا D در ارتباط نیست.

هم‌چنین در یک رابطه غیرپوشا یا تخصیص مجزا (Disjoint)، نمونه موجودیت‌های فرزند نمی‌توانند به طور همزمان با نمونه‌ای از موجودیت پدر در ارتباط باشند. برای مثال در این سؤال، نمونه موجودیت‌های C و D نمی‌توانند به طور همزمان با نمونه‌ای از موجودیت B در ارتباط باشند. به عبارت دیگر نمی‌توان نمونه‌هایی از موجودیت‌های C و D داشت که به طور همزمان با نمونه‌ای از موجودیت B در ارتباط هستند.

به عنوان مثالی دیگر هر نمونه از موجودیت ماشین که شماره شاسی یکتا و منحصر به فرد خود را دارد حتماً می‌بایست با یکی از نمونه موجودیت‌های نوع ماشین که دومحور (2WD) یا چهارمحور (4WD) است در ارتباط باشد. به عبارت دیگر نمی‌توان نمونه‌ای از موجودیت ماشین که شماره شاسی یکتا و منحصر به فرد خود را دارد داشت که با هیچ یک از نمونه موجودیت‌های نوع ماشین که دومحور (2WD) یا چهارمحور (4WD) است در ارتباط نباشد. این یعنی رابطه اجباری یا کلی (Total).

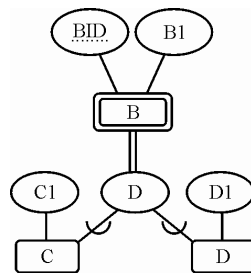
همچنین نمونه موجودیت‌های نوع ماشین که دومحور (2WD) یا چهارمحور (4WD) است نمی‌توانند به طور همزمان با نمونه‌ای از موجودیت ماشین که شماره شاسی یکتا و منحصر به فرد خود را دارد در ارتباط باشند، به عبارت دیگر نمی‌توان نمونه‌هایی از موجودیت‌های نوع ماشین که دومحور (2WD) یا چهارمحور (4WD) است داشت که به طور همزمان با نمونه‌ای از موجودیت ماشین که شماره شاسی یکتا و منحصر به فرد خود را دارد در ارتباط باشند. این یعنی رابطه غیرپوشا یا تخصیص مجزا (Disjoint).

از آنجاکه رابطه مابین موجودیت B و موجودیت‌های C و D یک رابطه اجباری یا کلی (Total) است، پس رکوردهای حاوی محتوای مقدار NULL در ستون‌های مربوط به موجودیت B در طراحی به شکل مدل دو جدولی در جداول C و D به ازای یک نمونه موجودیت از B به دلیل عدم ارتباط با برخی از نمونه موجودیت‌های C و D ایجاد نمی‌گردد، که باعث شود این محتوای NULL در جداول C و D حاصل از عدم ارتباط برخی از نمونه موجودیت‌های موجودیت B با نمونه موجودیت‌های C و D در جدول B، به شکل مدل سه جدولی نگهداری شود. در حالت رابطه اجباری مابین موجودیت B و موجودیت‌های C و D به ازای هر نمونه از موجودیت B، حتماً نمونه موجودیتی از C یا D وجود دارد که با B رابطه برقرار کند، پس در این حالت طراحی بهینه این است که کل صفات موجودیت B در دو جدول موجودیت‌های C و D قرار داده شود و یک طراحی به شکل مدل دو جدولی ایجاد گردد، همچنین از آنجاکه رابطه مابین موجودیت‌های C و D و موجودیت B یک رابطه غیرپوشا یا تخصیص مجزا (Disjoint) است، پس رکوردهای تکراری در جداول C و D به ازای یک نمونه موجودیت از موجودیت B ایجاد نمی‌گردد، که

افزونگی حاصل از تکرار رکوردها در جداول C و D سبب شود رکورد نمونه موجودیت‌های B در جدول B نگهداری شود و یک مدل سه جدولی ایجاد گردد. پس در این حالت طراحی بهینه این است که کل صفات موجودیت B در دو جدول موجودیت‌های C و D قرار داده شود و یک طراحی به شکل مدل دو جدولی ایجاد گردد.

در ادامه فرآیند نگاشت نمودار (ISA) به مدل رابطه‌ای را شرح می‌دهیم:

مدل تحلیل (نمودار (ISA))



مدل طراحی (مدل رابطه‌ای)

همانطور که در مدل طراحی قبل تر گفتیم، مدل طراحی درست جدول B به صورت (AID, BID, B1) در نظر گرفته شد.



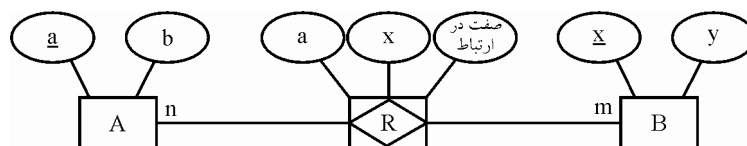
توجه: چون رابطه موجودیت B با موجودیت‌های C و D اجباری است، و رابطه موجودیت‌های C و D با موجودیت B از نوع Disjoint است. و به ازای هر نمونه موجودیت از B حتماً یک نمونه موجودیت از C و یا D وجود دارد و به تبع عدم مقادیر NULL جلوی نمونه موجودیت‌های B و عدم نیاز به عمل‌الحاق، نگاشت فوق به عنوان یک تبدیل ایده‌آل توصیه می‌گردد.

نگاشت رابطه چند به چند بین دو موجودیت به مدل رابطه‌ای

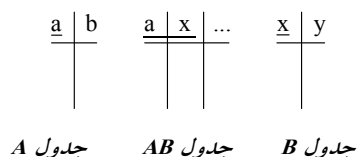
مستقل از اختیاری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد و یک جدول پُل (Bridge) نیز به عنوان ارتباط دهنده دو جدول مورد استفاده قرار می‌گیرد. همچنین کلید کاندید جدول پُل از ترکیب کلید کاندید دو جدول دیگر ایجاد می‌گردد.

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:

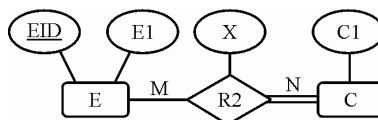


مدل طراحی:

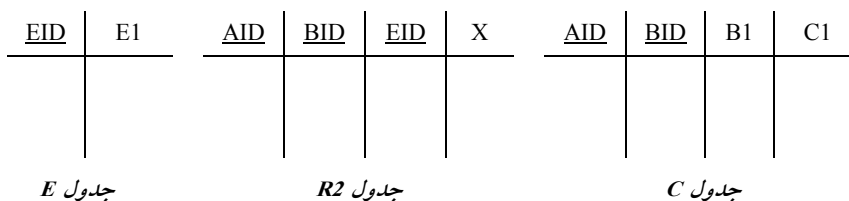


توجه: ستون a در جدول AB به عنوان کلید خارجی تعریف می‌گردد که به جدول A ارجاع می‌کند. همچنین ستون x در جدول AB به عنوان کلید خارجی تعریف می‌گردد که به جدول B ارجاع می‌کند. بخشی از مدل EER رسم شده در صورت سوال، یک رابطه چند به چند بین دو موجودیت را نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را شرح می‌دهیم.

مدل تحلیل:



مدل طراحی:



توجه: جدول C در نگاشت مرحله قبل ایجاد شده است که در اینجا دقیقاً به همان شکل و همان مشخصات، استفاده شده است.

توجه: همانطور که واضح است، طراحی جدول E در صورت سوال به صورت (EID, E1, AID, BID) در نظر گرفته شده است، که مطابق آنچه بیان کردیم، طراحی درست آن به صورت (EID, E1) است. بنابراین پرواضح است که گزینه دوم پاسخ سوال است.

۲۷- گزینه (۳) صحیح است.

به طور کلی در مدل رابطه‌ای، هر موجودیت شناسایی شده در نمودار ER (مدل تحلیل) هنگام نگاشت به مدل رابطه‌ای (مدل طراحی) به یک جدول تبدیل می‌شود. همچنین صفت‌های موجودیت پس از نگاشت آن در مدل رابطه‌ای به صورت ستون‌های جدول بیان می‌شوند. همچنین ارتباط بین جدول از طریق کلید خارجی برقرار می‌گردد.

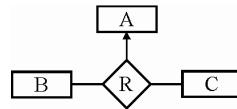
مدل ER رسم شده در صورت سوال، یک رابطه چند به چند و یک به چند بین سه موجودیت را نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را شرح می‌دهیم.

نگاشت رابطه چند به چند و یک به چند بین سه موجودیت به مدل رابطه‌ای

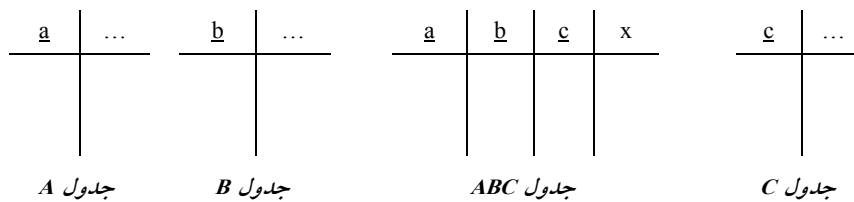
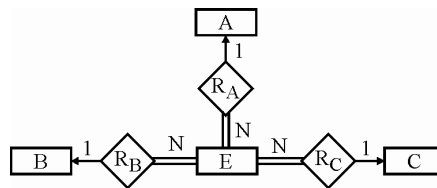
مستقل از اجباری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد و یک جدول

پُل (Bridge) نیز به عنوان ارتباط دهنده سه جدول مورد استفاده قرار می‌گیرد. همچنین کلید کاندید جدول پُل از ترکیب کلید کاندید سه جدول دیگر ایجاد می‌گردد. همچنین صفات متصل به رابطه در صورت وجود، درون جدول پُل مستتر می‌شود.

مدل تحلیل:



مدل طراحی:



توجه: ستون a در جدول ABC به عنوان کلید خارجی تعریف می‌گردد که به جدول A ارجاع می‌کند. همچنین ستون b در جدول ABC به عنوان کلید خارجی تعریف می‌گردد که به جدول B ارجاع می‌کند. همچنین ستون c در جدول ABC به عنوان کلید خارجی تعریف می‌گردد که به جدول C ارجاع می‌کند. توجه: کلید کاندید جدول ABC برابر abc است.

توجه: ستون x، می‌تواند به عنوان یک صفت در ارتباط در جدول ABC باشد.

توجه: نتیجه اینکه رابطه درجه سوم یا ارتباط سه‌گانه R، به سه رابطه درجه دوم یا ارتباط دوگانه توسط تعریف کلید خارجی تبدیل شد.

توجه: همانطور که واضح است، مدل طراحی در گزینه سوم به صورت شکل فوق در نظر گرفته شده‌است، که مطابق آنچه بیان کردیم، طراحی درستی است. بنابراین پرواضح است که گزینه سوم پاسخ سوال است.

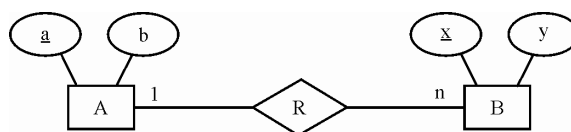
۲۸- گزینه (۴) صحیح است.

به طور کلی در مدل رابطه‌ای، هر موجودیت شناسایی شده در نمودار ER (مدل تحلیل) هنگام نگاشت به مدل رابطه‌ای (مدل طراحی) به یک جدول تبدیل می‌شود. همچنین صفات‌های موجودیت پس از نگاشت آن در مدل رابطه‌ای به صورت ستون‌های جدول بیان می‌شوند. همچنین ارتباط بین جدول از طریق کلید خارجی برقرار می‌گردد.

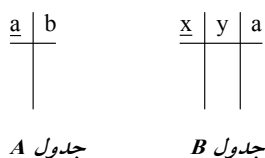
نگاشت رابطه یک به چند بین دو موجودیت به مدل رابطه‌ای مستقل از اختیاری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد. و کلید کاندید جدول یک در جدول چند به عنوان کلید خارجی تعریف می‌گردد.

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



مدل طراحی:

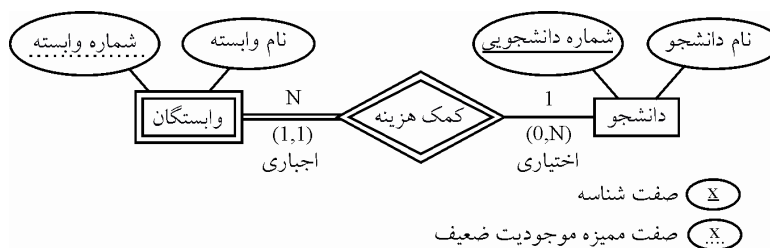


وابستگی وجودی

اگر در یک بانک اطلاعاتی، وجود یک موجودیت، وابسته به موجودیت دیگری باشد که در صورت حذف و تغییر موجودیت اصلی یعنی موجودیت قوی این موجودیت نیز تغییر کند، این نوع وابستگی را وابستگی وجودی گفته و به پدیده وابسته، موجودیت ضعیف گویند. همچنین موجودیت ضعیف کلید موجودیت قوی را در بر دارد تا هرگونه تغییر یا حذف در موجودیت قوی به موجودیت ضعیف اعمال شود. توجه: موجودیت ضعیف با دو مستطیل تو در تو نمایش داده می‌شود.

مثال: حالت اجباری و اختیاری

مدل تحلیل:



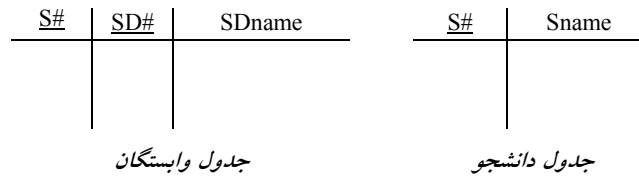
توجه: نماد خط مضاعف افقی نشانه اجباری بودن موجودیت چسبیده به آن است، اما نماد | به معنی یک و الزام شرکت در رابطه نشانه اجباری بودن موجودیت طرف مقابل است.

توجه: نماد خط افقی نشانه اختیاری بودن موجودیت چسبیده به آن است، اما نماد دایره کوچک توخالی به معنی صفر و عدم الزام شرکت در رابطه نشانه اختیاری بودن موجودیت طرف مقابل است.

توجه: قید (0,N) نشان می‌دهد که هر نمونه موجودیت از دانشجو حداقل با صفر و حداکثر با N نمونه

موجودیت از وابستگی ارتباط دارد و قید (1,1) نشان می‌دهد که هر نمونه موجودیت از وابستگی حداقل با یک و حداکثر با یک نمونه موجودیت از دانشجو ارتباط دارد.

مدل طراحی:



توجه: کلید کاندید جدول یک یعنی موجودیت قوی در جدول چند یعنی موجودیت ضعیف به عنوان کلید خارجی تعریف می‌گردد.

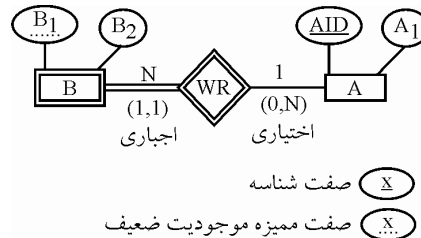
توجه: کلید کاندید جدول چند یعنی موجودیت ضعیف برابر ترکیب کلید خارجی و صفت ممیزه در جدول موجودیت ضعیف است. یعنی کلید کاندید جدول وابستگی برابر (S#,SD#) است.

توجه: صفت ممیزه یا کلید جزئی به طور سراسری در یک موجودیت ضعیف یکتا نیست، بلکه فقط در بین نمونه‌ها یا دسته‌هایی که با موجودیت قوی ارتباط دارند، یکتا است.

توجه: یک موجودیت ضعیف همیشه در ارتباطش با موجودیت قوی رابطه اجباری دارد. بخشی از مدل EER رسم شده در صورت سوال، یک رابطه یک به چند بین دو موجودیت قوی و ضعیف را نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را شرح می‌دهیم.

حالت اجباری و اختیاری

مدل تحلیل:

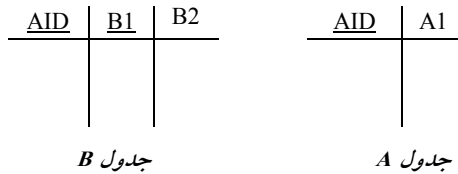


توجه: نماد خط مضاعف افقی نشانه اجباری بودن موجودیت چسبیده به آن است، اما نماد | به معنی یک و الزام شرکت در رابطه نشانه اجباری بودن موجودیت طرف مقابل است.

توجه: نماد خط افقی نشانه اختیاری بودن موجودیت چسبیده به آن است، اما نماد دایره کوچک توخالی به معنی صفر و عدم الزام شرکت در رابطه نشانه اختیاری بودن موجودیت طرف مقابل است.

توجه: قید (0,N) نشان می‌دهد که هر نمونه موجودیت از A حداقل با صفر و حداکثر با N نمونه موجودیت از B ارتباط دارد و قید (1,1) نشان می‌دهد که هر نمونه موجودیت از B حداقل با یک و حداکثر با یک نمونه موجودیت از A ارتباط دارد.

مدل طراحی:



توجه: کلید کاندید جدول یک یعنی موجودیت قوی در جدول چند یعنی موجودیت ضعیف به عنوان کلید خارجی تعریف می‌گردد.

توجه: کلید کاندید جدول چند یعنی موجودیت ضعیف برابر ترکیب کلید خارجی و صفت ممیزه در جدول موجودیت ضعیف است. یعنی کلید کاندید جدول B برابر (AID,B1) است.

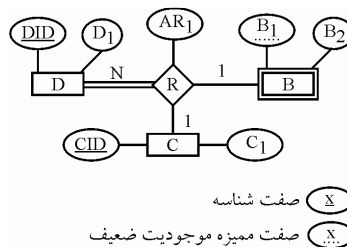
توجه: صفت ممیزه یا کلید جزئی به طور سراسری در یک موجودیت ضعیف یکتا نیست، بلکه فقط در بین نمونه‌ها یا دسته‌هایی که با موجودیت قوی ارتباط دارند، یکتا است.

توجه: یک موجودیت ضعیف همیشه در ارتباطش با موجودیت قوی رابطه اجباری دارد. همچنین در ادامه نگاشت، بخشی از مدل EER رسم شده در صورت سوال، یک رابطه یک به چند و یک به یک بین سه موجودیت را نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را شرح می‌دهیم.

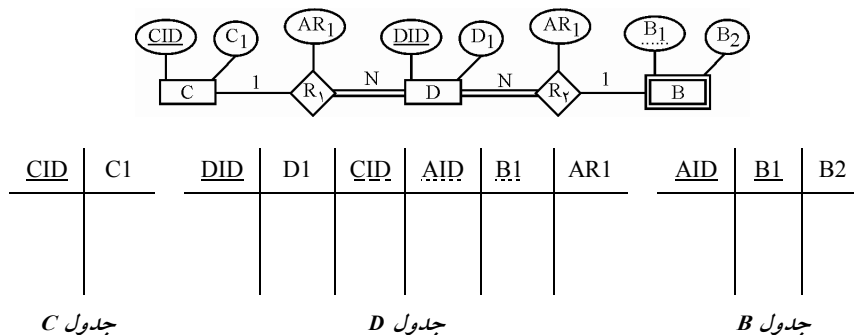
نگاشت رابطه یک به چند و یک به یک به یک بین سه موجودیت به مدل رابطه‌ای

مستقل از اختیاری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد. و کلید کاندید جداول یک در جدول چند به عنوان کلید خارجی تعریف می‌گردند. همچنین صفات متصل به رابطه، درون جدول چند مستتر می‌شود.

مدل تحلیل:



مدل طراحی:



توجه: ستون CID در جدول D به عنوان کلید خارجی تعریف می‌گردد که به جدول C ارجاع می‌کند. همچنین ستون‌های (AID, B1) در جدول D به عنوان کلید خارجی تعریف می‌گردد که به جدول B ارجاع می‌کند.

توجه: کلید کاندید جدول D فقط و فقط کلید کاندید موجودیت D یعنی صفت DID است.

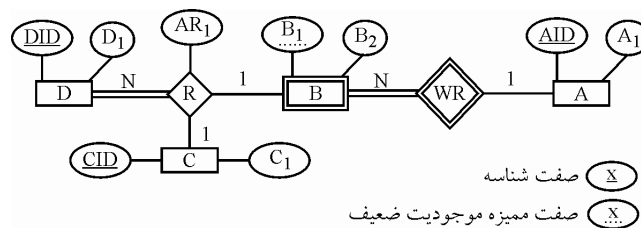
توجه: جدول B در نگاهت مرحله قبل ایجاد شده است که در اینجا دقیقاً به همان شکل و همان مشخصات، استفاده شده است.

توجه: نتیجه اینکه رابطه درجه سوم یا ارتباط سه‌گانه R، به دو رابطه درجه دوم یا ارتباط دوگانه توسط تعریف کلید خارجی تبدیل شد.

توجه: همانطور که واضح است، طراحی جدول D در گزینه چهارم بصورت (DID, D1, CID, AID, B1, AR1) در نظر گرفته شده است، که مطابق آنچه بیان کردیم، طراحی درستی است. بنابراین پرواضح است که گزینه چهارم پاسخ سوال است.

۲۹- گزینه (۱) صحیح است.

مثال: طراحی منطقی درست ارتباط R در نمودار ER زیر به چه شکل است؟



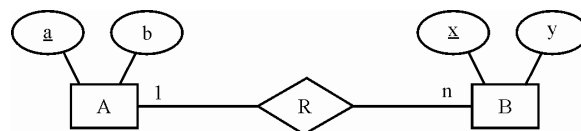
پاسخ: به طور کلی در مدل رابطه‌ای، هر موجودیت شناسایی شده در نمودار ER (مدل تحلیل) هنگام نگاهت به مدل رابطه‌ای (مدل طراحی) به یک جدول تبدیل می‌شود. همچنین صفت‌های موجودیت پس از نگاهت آن در مدل رابطه‌ای به صورت ستون‌های جدول بیان می‌شوند. همچنین ارتباط بین جدول از طریق کلید خارجی برقرار می‌گردد.

نگاشت رابطه یک به چند بین دو موجودیت به مدل رابطه‌ای

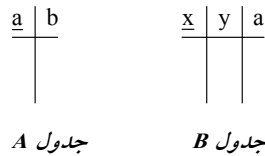
مستقل از اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد. و کلید کاندید جدول یک در جدول چند به عنوان کلید خارجی تعریف می‌گردد.

روال کلی نگاهت در این حالت به صورت زیر است:

مدل تحلیل:



مدل طراحی:

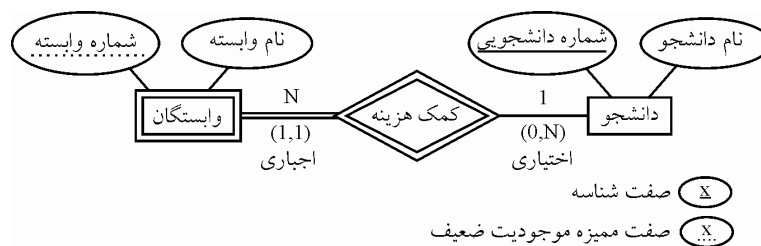


وابستگی وجودی

اگر در یک بانک اطلاعاتی، وجود یک موجودیت، وابسته به موجودیت دیگری باشد که در صورت حذف و تغییر موجودیت اصلی یعنی موجودیت قوی این موجودیت نیز تغییر کند، این نوع وابستگی را وابستگی وجودی گفته و به پدیده وابسته، موجودیت ضعیف گویند. همچنین موجودیت ضعیف کلید موجودیت قوی را در بر دارد تا هرگونه تغییر یا حذف در موجودیت قوی به موجودیت ضعیف اعمال شود. توجه: موجودیت ضعیف با دو مستطیل تو در تو نمایش داده می‌شود.

مثال: حالت اجباری و اختیاری

مدل تحلیل:

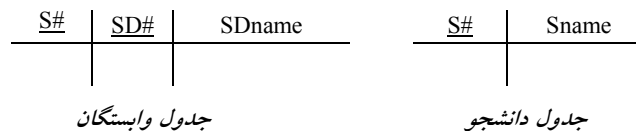


توجه: نماد خط مضاعف افقی نشانه اجباری بودن موجودیت چسبیده به آن است، اما نماد | به معنی یک و الزام شرکت در رابطه نشانه اجباری بودن موجودیت طرف مقابل است.

توجه: نماد خط افقی نشانه اختیاری بودن موجودیت چسبیده به آن است، اما نماد دایره کوچک توخالی به معنی صفر و عدم الزام شرکت در رابطه نشانه اختیاری بودن موجودیت طرف مقابل است.

توجه: قید (0,N) نشان می‌دهد که هر نمونه موجودیت از دانشجو حداقل با صفر و حداکثر با N نمونه موجودیت از وابستگی ارتباط دارد و قید (1,1) نشان می‌دهد که هر نمونه موجودیت از وابستگی حداقل با یک و حداکثر با یک نمونه موجودیت از دانشجو ارتباط دارد.

مدل طراحی:



توجه: کلید کاندید جدول یک یعنی موجودیت قوی در جدول چند یعنی موجودیت ضعیف به عنوان کلید خارجی تعریف می‌گردد.

توجه: کلید کاندید جدول چند یعنی موجودیت ضعیف برابر ترکیب کلید خارجی و صفت ممیزه در جدول موجودیت ضعیف است. یعنی کلید کاندید جدول وابستگان برابر $(S\#,SD\#)$ است.

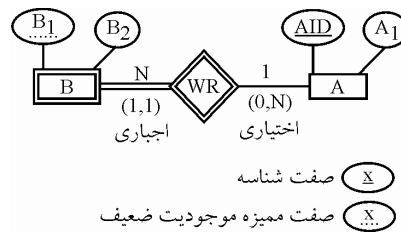
توجه: صفت ممیزه یا کلید جزئی به طور سراسری در یک موجودیت ضعیف یکتا نیست، بلکه فقط در بین نمونه‌ها یا دسته‌هایی که با موجودیت قوی ارتباط دارند، یکتا است.

توجه: یک موجودیت ضعیف همیشه در ارتباطش با موجودیت قوی رابطه اجباری دارد.

بخشی از مدل ER رسم شده در صورت سوال، یک رابطه یک به چند بین دو موجودیت قوی و ضعیف را نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را شرح می‌دهیم.

حالت اجباری و اختیاری

مدل تحلیل:

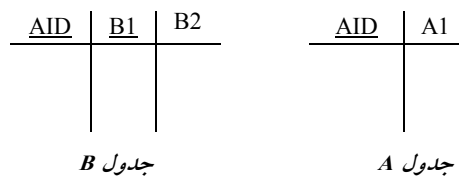


توجه: نماد خط مضاعف افقی نشانه اجباری بودن موجودیت چسبیده به آن است، اما نماد | به معنی یک و الزام شرکت در رابطه نشانه اجباری بودن موجودیت طرف مقابل است.

توجه: نماد خط افقی نشانه اختیاری بودن موجودیت چسبیده به آن است، اما نماد دایره کوچک توخالی به معنی صفر و عدم الزام شرکت در رابطه نشانه اختیاری بودن موجودیت طرف مقابل است.

توجه: قید $(0,N)$ نشان می‌دهد که هر نمونه موجودیت از A حداقل با صفر و حداکثر با N نمونه موجودیت از B ارتباط دارد و قید $(1,1)$ نشان می‌دهد که هر نمونه موجودیت از B حداقل با یک و حداکثر با یک نمونه موجودیت از A ارتباط دارد.

مدل طراحی:



توجه: کلید کاندید جدول یک یعنی موجودیت قوی در جدول چند یعنی موجودیت ضعیف به عنوان کلید خارجی تعریف می‌گردد.

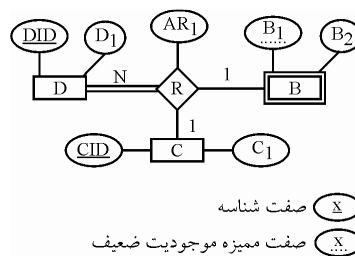
توجه: کلید کاندید جدول چند یعنی موجودیت ضعیف برابر ترکیب کلید خارجی و صفت ممیزه در جدول موجودیت ضعیف است. یعنی کلید کاندید جدول B برابر $(AID,B1)$ است.

توجه: صفت ممیزه یا کلید جزئی به طور سراسری در یک موجودیت ضعیف یکتا نیست، بلکه فقط در بین نمونه‌ها یا دسته‌هایی که با موجودیت قوی ارتباط دارند، یکتا است.

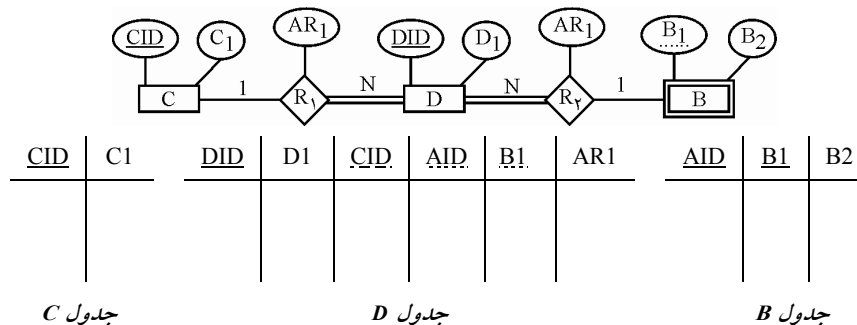
توجه: یک موجودیت ضعیف همیشه در ارتباطش با موجودیت قوی رابطه اجباری دارد. همچنین در ادامه نگاشت، بخشی از مدل ER رسم شده در صورت سوال، یک رابطه یک به چند و یک به یک بین سه موجودیت را نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را شرح می‌دهیم.

نگاشت رابطه یک به چند و یک به یک بین سه موجودیت به مدل رابطه‌ای مستقل از اجباری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد. و کلید کاندید جداول یک در جدول چند به عنوان کلید خارجی تعریف می‌گردند. همچنین صفات متصل به رابطه، درون جدول چند مستتر می‌شود.

مدل تحلیل:



مدل طراحی:



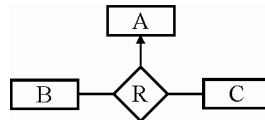
توجه: ستون CID در جدول D به عنوان کلید خارجی تعریف می‌گردد که به جدول C ارجاع می‌کند. همچنین ستون‌های (AID, B1) در جدول D به عنوان کلید خارجی تعریف می‌گردد که به جدول B ارجاع می‌کند.

توجه: کلید کاندید جدول D فقط و فقط کلید کاندید موجودیت D یعنی صفت DID است.

توجه: جدول B در نگاشت مرحله قبل ایجاد شده است که در اینجا دقیقاً به همان شکل و همان مشخصات، استفاده شده است.

توجه: نتیجه اینکه رابطه درجه سوم یا ارتباط سه گانه R، به دو رابطه درجه دوم یا ارتباط دوگانه توسط تعریف کلید خارجی تبدیل شد. بنابراین مطابق آنچه گفتیم می‌توان از یک ارتباط سه گانه (رابطه درجه سوم)، دو ارتباط دوگانه (رابطه درجه دوم) را نتیجه گرفت.

مثال: فرض کنید که رابطه سه‌تایی زیر بین موجودیت‌های A, B, C وجود دارد، حال اگر بخواهیم این رابطه سه‌تایی را با رابطه‌های دودویی نمایش دهیم، نمودار موجودیت و رابطه (ERD) آن به چه شکل خواهد بود؟



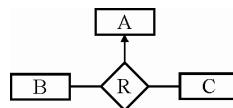
پاسخ: به طور کلی در مدل رابطه‌ای، هر موجودیت شناسایی شده در نمودار ER (مدل تحلیل) هنگام نگاشت به مدل رابطه‌ای (مدل طراحی) به یک جدول تبدیل می‌شود. همچنین صفت‌های موجودیت پس از نگاشت آن در مدل رابطه‌ای به صورت ستون‌های جدول بیان می‌شوند. همچنین ارتباط بین جدول از طریق کلید خارجی برقرار می‌گردد.

مدل ER رسم شده در صورت سوال، یک رابطه چند به چند و یک به چند بین سه موجودیت را نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را شرح می‌دهیم.

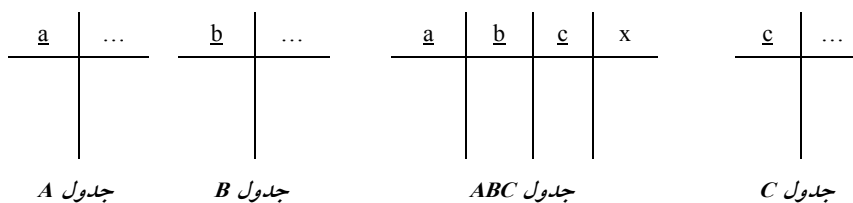
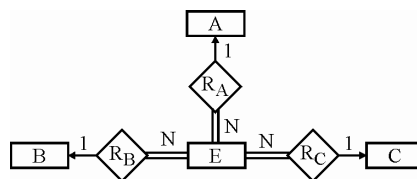
نگاشت رابطه چند به چند و یک به چند بین سه موجودیت به مدل رابطه‌ای

مستقل از اجباری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد و یک جدول پل (Bridge) نیز به عنوان ارتباط دهنده سه جدول مورد استفاده قرار می‌گیرد. همچنین کلید کاندید جدول پل از ترکیب کلید کاندید سه جدول دیگر ایجاد می‌گردد. همچنین صفات متصل به رابطه در صورت وجود، درون جدول پل مستتر می‌شود.

مدل تحلیل:



مدل طراحی:



توجه: ستون a در جدول ABC به عنوان کلید خارجی تعریف می‌گردد که به جدول A ارجاع می‌کند.

همچنین ستون b در جدول ABC به عنوان کلید خارجی تعریف می‌گردد که به جدول B ارجاع می‌کند. همچنین ستون c در جدول ABC به عنوان کلید خارجی تعریف می‌گردد که به جدول C ارجاع می‌کند. توجه: کلید کاندید جدول ABC برابر abc است.

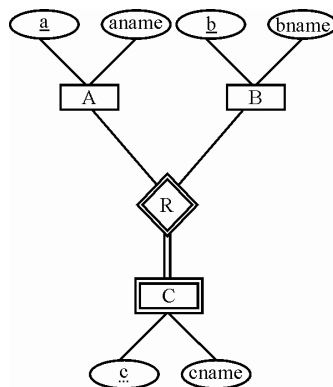
توجه: ستون x، به عنوان یک صفت در ارتباط در جدول ABC به صورت مستتر قرار می‌گیرد. زیرا همانطور که گفتیم صفات متصل به رابطه در صورت وجود، درون جدول پُل مستتر می‌شود.

توجه: نتیجه اینکه رابطه درجه سوم یا ارتباط سه‌گانی R، به سه رابطه درجه دوم یا ارتباط دوگانی توسط تعریف کلید خارجی تبدیل شد. بنابراین مطابق آنچه گفتیم می‌توان از یک ارتباط سه‌گانی (رابطه درجه سوم)، سه ارتباط دوگانی (رابطه درجه دوم) را نتیجه گرفت اما عکس آن صادق نیست. بنابراین پرواضح است که گزینه اول عبارت نادرستی است و به تبع پاسخ سوال است و گزینه سوم عبارت درستی است و به تبع پاسخ سوال نیست.

گزینه دوم عبارت درستی است، بنابراین پاسخ سوال نیست. گزینه دوم به این شکل بیان شده است. «وجودیت ضعیف وابسته به یک موجودیت قوی (مانند E1) می‌تواند خود با موجودیت قوی دیگری (مانند E2) رابطه داشته باشد.» که عبارت درستی است.

مثال: فرض کنید که موجودیت ضعیف C، با دو موجودیت قوی A و B رابطه داشته باشد. نمودار موجودیت و رابطه (ERD) یعنی مدل تحلیل و مدل طراحی آن به چه شکل خواهد بود؟

مدل تحلیل:



توجه: در شکل فوق صفت a کلید موجودیت قوی A و صفت b کلید موجودیت قوی B است، همچنین صفت c، صفت ممیزه موجودیت ضعیف C است.

توجه: نماد خط مضاعف افقی نشانه اجباری بودن موجودیت چسبیده به آن است، اما نماد | به معنی یک و الزام شرکت در رابطه نشانه اجباری بودن موجودیت طرف مقابل است.

توجه: نماد خط افقی نشانه اختیاری بودن موجودیت چسبیده به آن است، اما نماد دایره کوچک توخالی به معنی صفر و عدم الزام شرکت در رابطه نشانه اختیاری بودن موجودیت طرف مقابل است.

مدل طراحی:

a	aname	...	a	b	c	cname	b	bname	...

جدول A جدول C جدول B

توجه: کلیدهای کاندید دو جدول موجودیت قوی در جدول موجودیت ضعیف به عنوان کلید خارجی تعریف می‌گردند.

توجه: ستون a در جدول C به عنوان کلید خارجی تعریف می‌گردد که به جدول A ارجاع می‌کند. همچنین ستون b در جدول C به عنوان کلید خارجی تعریف می‌گردد که به جدول B ارجاع می‌کند.

توجه: کلید کاندید جدول موجودیت ضعیف برابر ترکیب کلیدهای خارجی و صفت ممیزه در جدول موجودیت ضعیف است. یعنی کلید کاندید جدول C برابر (a,b,c) است.

توجه: صفت ممیزه یا کلید جزئی به طور سراسری در یک موجودیت ضعیف یکتا نیست، بلکه فقط در بین نمونه‌ها یا دسته‌هایی که با موجودیت قوی ارتباط دارند، یکتا است.

توجه: یک موجودیت ضعیف همیشه در ارتباطش با موجودیت قوی رابطه اجباری دارد. گزینه چهارم عبارت درستی است، بنابراین پاسخ سوال نیست. گزینه چهارم به این شکل بیان شده است. «اگر یک نوع موجودیت (مانند E₁) تنها یک صفت داشته باشد و تنها با یک نوع موجودیت دیگر (مانند E₂) در ارتباط باشد، می‌توانیم آن را حذف و به عنوان صفت موجودیت دوم (یعنی E₂) در نظر بگیریم.» بهترین راه شناسایی موجودیت‌های یک محیط عملیاتی، بررسی مقادیری است که قرار است ذخیره شود، ذخیره‌سازی مقادیر داخل صفات (ستون‌ها) انجام می‌شود و این صفات قطعاً مرتبط با یک موجودیت هستند. بنابراین هر موجودیت باید دارای چندین صفت باشد. پس موجودیت فاقد صفت معنا ندارد. حتی در اغلب مواقع، موجودیت تک خصیصه‌ای نیز قابل قبول نمی‌باشد و به احتمال فراوان آن صفت مرتبط به یک موجودیت دیگر است که به خطا به عنوان یک موجودیت تک خصیصه‌ای شناخته شده است.

۳۰- گزینه (۲) صحیح است.

صورت سوال به این شکل است:

با توجه به گزاره‌های زیر کدام مورد درست است؟

(a) مدل مفهومی پایگاه داده، استقلال داده‌ای را افزایش می‌دهد.

گزاره a درست است، زیرا یکی از مهم‌ترین مزایای تکنولوژی پایگاه داده‌ها (مدل مفهومی پایگاه داده)، بلکه مهمترین هدف آن تأمین و افزایش استقلال داده‌ای است، به معنی وابسته نبودن برنامه‌های کاربردی به داده‌های ذخیره شده.

(b) اگر K₁ و K₂ ابرکلیدهای رابطه R باشند، آنگاه $K_1 \cap K_2$ یک ابرکلید R است.

گزاره b نادرست است، زیرا جدول (nano, stno, avg) stud را در نظر بگیرید. nano (کد ملی) یک کلید کاندید و به تبع یک ابرکلید است. همچنین stno (شماره دانشجویی) یک کلید کاندید و به تبع یک ابرکلید

دیگر باشد. اجتماع این دو می‌شود: (nano, stno) که هر چند ابرکلید است ولی یک کلید کاندید نیست (توجه کنید که کلید کاندید باید کمینه باشد و عضو زائد نداشته باشد). هم‌چنین اشتراک این دو برابر مجموعه تهی است که نمی‌تواند نه ابرکلید و نه کلید کاندید باشد. به صورت زیر:

$$K1 \cap K2 = \{\text{nano}\} \cap \{\text{stno}\} = \{\}$$

(c) برای اتصال به پایگاه داده، در **Connection String** نحوه‌ی احراز هویت کاربر مشخص می‌شود. گزاره c درست است، زیرا یک برنامه‌ی کاربردی نوشته شده به یک زبان روالی سطح بالا پس از ارتباط با یک سرویس‌دهنده زبان بیانی مانند SQL Server از طریق مکانیزم‌های ویژه (Connection String) اقدام به استفاده از بانک اطلاعات می‌نماید. همچنین برای اتصال به پایگاه داده، در رشته‌ی اتصال (Connection String) نحوه احراز هویت (authentication) کاربر مشخص می‌شود.

(d) در رابطه R با تعداد خصیصه بزرگتر از یک، تعداد ابرکلیدها همواره از تعداد کلیدهای کاندید بیشتر است.

گزاره d نادرست است، زیرا در جدول تمام کلید، یک جدول فقط و فقط یک ابرکلید دارد و فقط و فقط هم یک کلید کاندید دارد. رابطه تمام کلید مثلا ممکن است سه ستون داشته باشد، در این حالت یک جدول فقط و فقط یک ابرکلید دارد و فقط و فقط هم یک کلید کاندید دارد. یعنی حداقل یک ابرکلید و یک کلید کاندید دارد و حداکثر هم یک ابرکلید و یک کلید کاندید دارد.

۳۱- گزینه (۳) صحیح است.

صورت سوال به این شکل است:

با توجه به گزاره‌های زیر، کدام مورد درست است؟

(a) اگر رابطه R، دارای $n=2k$ خصیصه باشد، آنگاه تعداد کلیدهای کاندید آن حداقل یک و حداکثر $\binom{n}{k}$ است.

گزاره a درست است، زیرا هر جدولی حتما حداقل یک کلید کاندید دارد. و در حالت حداکثر با توجه به مفروضات گزاره a، همواره رابطه زیر برقرار است:

$$\max\left(\binom{n}{1}, \binom{n}{2}, \dots, \binom{n}{k-1}, \binom{n}{k} + \binom{n}{k+1}, \dots, \binom{n}{n}\right) = \binom{n}{k}$$

مثال: اگر $k=1$ آنگاه $n=2k=2$

$$\max\left(\binom{2}{1}, \binom{2}{2}\right) = \binom{2}{1}$$

مثال: اگر $k=2$ آنگاه $n=2k=4$

$$\max\left(\binom{4}{1}, \binom{4}{2}, \binom{4}{3}, \binom{4}{4}\right) = \binom{4}{2}$$

مثال: اگر $k=3$ آنگاه $n=2k=6$

$$\max\left(\binom{6}{1}, \binom{6}{2}, \binom{6}{3}, \binom{6}{4}, \binom{6}{5}, \binom{6}{6}\right) = \binom{6}{3}$$

توجه: به وضوح حالت قرینه‌ی مقادیر قابل مشاهده است، که همان حد وسط یعنی $\binom{6}{3}$ ، بیشترین مقدار

است. برای مثال حالت $\binom{6}{4}$ قرینه‌ی حالت $\binom{6}{2}$ است. و حالت $\binom{6}{5}$ قرینه‌ی حالت $\binom{6}{1}$ است به جز حالت آخر $\binom{6}{6}$ که قرینه ندارد. هر کدام از ترکیب‌ها مثل $\binom{6}{1}$ یا $\binom{6}{2}$ یا $\binom{6}{3}$ تا $\binom{6}{6}$ می‌تواند تعداد کلیدهای کاندید باشد، اما بیشترین تعداد آن ترکیب $\binom{6}{3}$ است. دقت کنید که نباید ترکیب‌ها را جمع کنید، حاصل جمع ترکیب‌ها، تعداد ابرکلیدها را می‌شمارد.

(b) اگر رابطه R ، دارای n خصیصه باشد، آنگاه تعداد ابرکلیدهای این رابطه حداکثر $\binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{n}$ است.

گزاره b نادرست است، زیرا در حالت کلی، اگر رابطه R ، دارای n خصیصه باشد، آنگاه تعداد ابرکلیدهای آن حداقل یک و حداکثر $2^n - 1$ است. در جدول تمام کلید، یک جدول فقط و فقط یک ابرکلید دارد و فقط هم یک کلید کاندید دارد. رابطه تمام کلید مثلا ممکن است سه ستون داشته باشد، در این حالت یک جدول فقط و فقط یک ابرکلید دارد و فقط هم یک کلید کاندید دارد. یعنی حداقل یک ابرکلید و یک کلید کاندید دارد و حداکثر هم یک ابرکلید و یک کلید کاندید دارد. در گزاره اول حالت کلی مورد بررسی قرار گرفته است. دقت کنید که حداقل یک ابرکلید و حداکثر $2^n - 1$ ابرکلید، حالت خاص جدول تمام کلید هم پوشش می‌دهد چون بیان حداکثر $2^n - 1$ مقادیر کوچکتر و برابر خودش را پوشش می‌دهد. اگر در یک رابطه با n خصیصه، تک تک خصیصه‌ها به تنهایی کلید کاندید باشد، آنگاه رابطه دارای n کلید کاندید است. بنابراین هر زیرمجموعه غیر تهی از خصیصه‌های این رابطه یک ابرکلید است. که در این حالت تعداد ابرکلیدهای یک رابطه با n خصیصه برابر با $2^n - 1$ است که بیشترین مقدار ممکن در تعداد ابرکلیدهای یک رابطه با n خصیصه است. ابرکلید بدون صفت نداریم، بنابراین حالت $\binom{n}{0}$ اضافه است. همانطور که گفتیم در حالت کلی، یک رابطه دارای n خصیصه، شرایط مختلفی را در تعداد ابرکلید می‌تواند تجربه کند، که حداکثر تعداد ابرکلیدهایی که می‌تواند تجربه کند برابر $\binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{n}$ یا $2^n - 1$ است. در واقع عبارت زیر برقرار است:

$$2^n - 1 = \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n}$$

(c) تعداد ابرکلیدهای یک رابطه همواره بیشتر از تعداد کلیدهای کاندید آن رابطه است.

گزاره c نادرست است، زیرا در جدول تمام کلید، یک جدول فقط و فقط یک ابرکلید دارد و فقط هم یک کلید کاندید دارد. رابطه تمام کلید مثلا ممکن است سه ستون داشته باشد، در این حالت یک جدول فقط و فقط یک ابرکلید دارد و فقط هم یک کلید کاندید دارد. یعنی حداقل یک ابرکلید و یک کلید کاندید دارد و حداکثر هم یک ابرکلید و یک کلید کاندید دارد.

(d) هر صفت مرکب، لزوماً تک مقداری است.

گزاره d نادرست است، زیرا صفت مرکب صفتی است که از چند صفت ساده تشکیل شده باشد به گونه‌ای که تجزیه‌شدنی باشند و اجزاء حاصله خود صفات ساده باشند. مانند صفت آدرس که از اجزاء نام استان، نام شهر، نام خیابان، نام کوچه، شماره پلاک و کدپستی تشکیل شده است. بعضی از صفات چه ساده و چه مرکب فقط می‌توانند یک مقدار را بگیرند که به این صفات، صفت تک‌مقداری می‌گویند. مانند شماره دانشجویی که نمی‌تواند بیش از یک مقدار داشته باشد. صفاتی وجود دارند که می‌توانند چندین مقدار را

بگیرند مانند صفت مدرک در موجودیت استاد که می‌تواند مقادیر لیسانس، فوق لیسانس و یا دکتری را در خود بگیرد. به مثال‌های زیر توجه کنید:

صفت ساده تک‌مقداری: مانند کدملی

صفت ساده چندمقداری: مانند مدرک تحصیلی

صفت مرکب تک‌مقداری: مانند تاریخ تولد

صفت مرکب چندمقداری: مانند آدرس

۳۲- گزینه (۱) صحیح است.

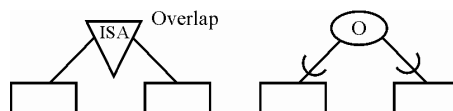
نگاشت رابطه ISA یا وراثت به مدل رابطه‌ای

در رابطه ISA رابطه پدر با فرزندان به دو صورت رابطه اختیاری یا جزئی یا بخشی (Partial) با نماد خط عمودی و رابطه اجباری یا کلی یا کامل (Total) با نماد خط مضاعف عمودی است و رابطه فرزندان با پدر به دو صورت رابطه متصل یا پوشا یا تخصیص غیرمجزا (Overlap) و رابطه منفصل یا غیرپوشا یا تخصیص مجزا (Disjoint) می‌باشد.

توجه: در یک رابطه اجباری یا کلی (Total)، هر نمونه از موجودیت پدر حتماً می‌بایست با یکی از نمونه موجودیت‌های فرزند در ارتباط باشد.

توجه: در یک رابطه اختیاری یا جزئی (Partial)، هر نمونه از موجودیت پدر می‌تواند با یکی از نمونه موجودیت‌های فرزند در ارتباط باشد یا نباشد.

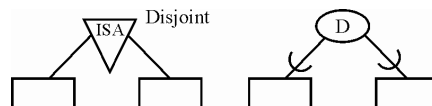
رابطه متصل یا پوشا یا تخصیص غیرمجزا (Overlap) مابین فرزندان و پدر به دو شیوه زیر نشان داده می‌شود:



توجه: در رابطه متصل یا پوشا یا تخصیص غیرمجزا (Overlap) ارتباط پدر با فرزندان می‌تواند یک به یک باشد و همچنین می‌تواند یک به چند باشد. و همچنین اشتراک نمونه موجودیت‌ها میان موجودیت‌های فرزند با یک نمونه موجودیت از پدر می‌تواند تهي باشد.

توجه: در یک رابطه پوشا یا تخصیص غیرمجزا (Overlap)، نمونه موجودیت‌های فرزند می‌توانند به طور همزمان با نمونه‌ای از موجودیت پدر در ارتباط باشند که این یعنی همان رابطه یک به چند میان پدر و فرزندان.

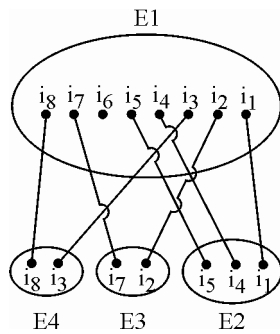
رابطه منفصل یا غیرپوشا یا تخصیص مجزا (Disjoint) مابین فرزندان و پدر به دو شیوه زیر نشان داده می‌شود:



توجه: در رابطه منفصل یا غیرپوشا یا تخصیص مجزا (Disjoint) ارتباط پدر با فرزندان فقط و فقط یک به یک است. و همچنین اشتراک نمونه موجودیت‌ها میان موجودیت‌های فرزند با یک نمونه موجودیت از پدر همواره تهي است.

توجه: در یک رابطه غیرپوشا یا تخصیص مجزا (Disjoint)، نمونه موجودیت‌های فرزند نمی‌توانند به طور همزمان با نمونه‌ای از موجودیت پدر در ارتباط باشند که این یعنی همان رابطه یک به یک میان پدر و فرزندان.

صورت سوال به این شکل است:



نمونه‌هایی از چهار نوع موجودیت E1، E2، E3 و E4 در شکل مقابل نمایش داده شده است، کدام مورد بهترین نمودار EER معرفی محیط است؟

توجه: در صورت سوال باید شرایط محیط عملیاتی به دقت گفته می‌شد، تعدد نمونه موجودیت‌ها بیان شده در نمودار صورت سوال، گویای شرایط فعلی محیط عملیاتی است و گویای آینده تعدد نمونه موجودیت‌های شرایط آتی محیط عملیاتی نیست، ضمن اینکه نمودار مطرح شده در صورت سوال نشانه تعدد لحظه‌ای یک محیط عملیاتی است و گویای آینده محیط عملیاتی نیست، و خود نمودار ER همواره گویای محیط عملیاتی است که طراح خواسته از تعدد به نمودار ER برسد آن هم بدون ذکر شرایط دقیق محیط عملیاتی. که تفکر نادرستی بوده است. سناریوی دقیق شرایط محیط عملیاتی می‌بایست در صورت سوال مطرح می‌شد. که نشده است.

بطور کلی خواص رابطه به سه شکل زیر وجود دارد:

الف) درجه ارتباط

ب) کاردینالیته ارتباط

ج) اجباری و اختیاری بودن ارتباط

توجه: همه روابط سه خصیصه فوق را به طور همزمان دارند، اما مقادیر این خصیصه‌ها در روابط مختلف، متفاوت است.

الف) درجه ارتباط

به تعداد موجودیت‌هایی که در یک رابطه مشارکت دارند، درجه ارتباط گفته می‌شود. درجه در مدل ER عددی صحیح و کوچکتر از 5 است. ارتباط‌های درجه 1، 2 و 3 معمول، ارتباط درجه 4 کمیاب و غیر معمول است و ارتباط بالاتر از درجه 4 قابل رسم کردن نیست.

ب) کاردینالیته ارتباط

کاردینالیته ارتباط بر سه نوع است: یک به یک، یک به چند، چند به چند.

ج) اجباری و اختیاری بودن ارتباط

این نوع رابطه به دو دسته کلی زیر تقسیم می‌شود:

۱- اجباری یا کلی (Total)

یک رابطه اجباری است، اگر و تنها اگر تمام نمونه‌های موجودیت در رابطه شرکت کرده باشند. اجباری بودن رابطه ISA در نمودار EER با نماد خط مضاعف عمودی از موجودیت پدر به سمت موجودیت‌های فرزند نشان داده می‌شود.

۲- اختیاری یا جزئی (Partial)

یک رابطه اختیاری است، اگر و تنها اگر حداقل یکی از نمونه‌های موجودیت در رابطه شرکت نکرده باشد. اختیاری بودن رابطه ISA در نمودار EER با نماد خط عمودی از موجودیت پدر به سمت موجودیت‌های فرزند نشان داده می‌شود.

همانطور که گفتیم در رابطه ISA رابطه پدر با فرزندان به دو صورت رابطه اختیاری یا جزئی یا بخشی (Partial) با نماد خط عمودی و رابطه اجباری یا کلی یا کامل (Total) با نماد خط مضاعف عمودی است. همچنین همانطور که گفتیم یک رابطه اختیاری است، اگر و تنها اگر حداقل یکی از نمونه‌های موجودیت پدر در رابطه شرکت نکرده باشد. اختیاری بودن رابطه ISA در نمودار EER با نماد خط عمودی نشان داده می‌شود.

همانطور که در شکل صورت سوال واضح و مشخص است، نمونه موجودیت z6 در موجودیت E1 یعنی پدر با هیچ یک از نمونه موجودیت‌های فرزند یعنی موجودیت‌های E2، E3 و E4 وارد رابطه نشده است. و از آنجاکه اگر و تنها اگر حداقل یکی از نمونه‌های موجودیت پدر در رابطه با فرزندان شرکت نکرده باشد، آن رابطه اختیاری است، بنابراین نمونه‌های موجودیت E1 به صورت **اختیاری** وارد رابطه با فرزندان یعنی موجودیت‌های E2، E3 و E4 شده‌اند. بنابراین گزینه‌های دوم و سوم پاسخ سوال نیستند.

توجه: اگر فرض کنیم نمودار مطرح شده در صورت سوال، یک لحظه‌ی خاص از تعریف محیط عملیاتی نباشد، بلکه حالت کلی، ایستا، پایدار و همیشگی محیط عملیاتی باشد، آنگاه از آنجا که ارتباط پدر با فرزندان یک به یک است. و همچنین اشتراک نمونه موجودیت‌ها میان موجودیت‌های فرزند با یک نمونه موجودیت از پدر تهی است، می‌توان رابطه فرزندان با پدر را حالت منفصل یا غیرپوشا یا تخصیص مجزا (Disjoint) در نظر گرفت، که در این شرایط گزینه اول پاسخ سوال است. که طراح محترم هم همین گزینه را به عنوان بهترین نمودار EER برای نمودار صورت سوال در نظر گرفته است.

توجه: در یک رابطه غیرپوشا یا تخصیص مجزا (Disjoint)، نمونه موجودیت‌های فرزند نمی‌توانند به طور همزمان با نمونه‌ای از موجودیت پدر در ارتباط باشند که این یعنی همان رابطه یک به یک میان پدر و فرزندان.

توجه: اما اگر فرض کنیم نمودار مطرح شده در صورت سوال، یک لحظه‌ی خاص از تعریف محیط عملیاتی باشد، یعنی حالت جزئی، پویا، ناپایدار و غیرهمیشگی محیط عملیاتی باشد، آنگاه از آنجا که ارتباط پدر با فرزندان می‌تواند یک به یک باشد و همچنین می‌تواند یک به چند باشد در آینده. و همچنین اشتراک نمونه موجودیت‌ها میان موجودیت‌های فرزند با یک نمونه موجودیت از پدر می‌تواند تهی باشد و همچنین می‌تواند غیرتهی باشد، می‌توان رابطه فرزندان با پدر را حالت متصل یا پوشا یا تخصیص غیرمجزا (Overlap) در نظر گرفت، که در این شرایط گزینه چهارم پاسخ سوال است.

توجه: در یک رابطه پوشا یا تخصیص غیرمجزا (Overlap)، نمونه موجودیت‌های فرزند می‌توانند به طور همزمان با نمونه‌ای از موجودیت پدر در ارتباط باشند که این یعنی همان رابطه یک به چند میان پدر و

فرزندان.

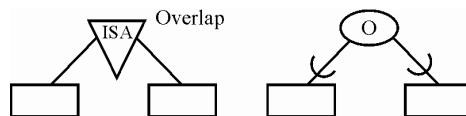
توجه: سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه اول را به عنوان پاسخ اعلام کرده بود.

۳۳- گزینه (۳) صحیح است.

نگاشت رابطه ISA یا وراثت به مدل رابطه‌ای

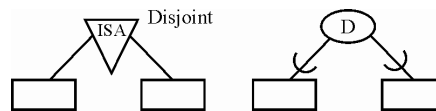
در رابطه ISA رابطه پدر با فرزندان به دو صورت رابطه اختیاری یا جزئی یا بخشی (Partial) با نماد خط عمودی و رابطه اجباری یا کلی یا کامل (Total) با نماد خط مضاعف عمودی است و رابطه فرزندان با پدر به دو صورت رابطه متصل یا پوشا یا تخصیص غیرمجزا (Overlap) و رابطه منفصل یا غیرپوشا یا تخصیص مجزا (Disjoint) می‌باشد.

رابطه متصل یا پوشا یا تخصیص غیرمجزا (Overlap) مابین فرزندان و پدر به دو شیوه زیر نشان داده می‌شود:



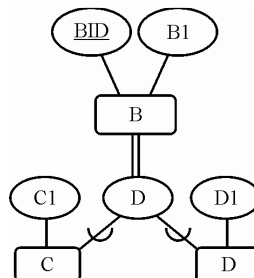
توجه: در رابطه متصل یا پوشا یا تخصیص غیرمجزا (Overlap) ارتباط پدر با فرزندان می‌تواند یک به یک باشد و همچنین می‌تواند یک به چند باشد. و همچنین اشتراک نمونه موجودیت‌ها میان موجودیت‌های فرزند با یک نمونه موجودیت از پدر می‌تواند تهی باشد و همچنین می‌تواند غیرتهی باشد.

رابطه منفصل یا غیرپوشا یا تخصیص مجزا (Disjoint) مابین فرزندان و پدر به دو شیوه زیر نشان داده می‌شود:



توجه: در رابطه منفصل یا غیرپوشا یا تخصیص مجزا (Disjoint) ارتباط پدر با فرزندان فقط و فقط یک به یک است. و همچنین اشتراک نمونه موجودیت‌ها میان موجودیت‌های فرزند با یک نمونه موجودیت از پدر همواره تهی است.

مثال: نمودار نهاد و رابطه زیر را در نظر بگیرید:



مدل EER رسم شده در شکل فوق، یک رابطه اجباری یا کلی یا کامل (Total) مابین موجودیت B و موجودیت‌های C و D و یک رابطه منفصل یا غیرپوشا یا تخصیص مجزا (Disjoint) را مابین موجودیت‌های C و D و موجودیت B نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را بیان می‌کنیم.

در یک رابطه اجباری یا کلی یا کامل (Total)، هر نمونه از موجودیت پدر حتماً می‌بایست با یکی از نمونه موجودیت‌های فرزند در ارتباط باشد. برای نمونه در این مثال، هر نمونه از موجودیت B حتماً می‌بایست با یکی از نمونه موجودیت‌های C یا D در ارتباط باشد. به عبارت دیگر نمی‌توان نمونه‌ای از موجودیت B داشت که با هیچ یک از نمونه موجودیت‌های C یا D در ارتباط نیست.

هم‌چنین در یک رابطه منفصل یا غیرپوشا یا تخصیص مجزا (Disjoint)، نمونه موجودیت‌های فرزند نمی‌توانند به طور همزمان با نمونه‌ای از موجودیت پدر در ارتباط باشند. برای نمونه در این مثال، نمونه موجودیت‌های C و D نمی‌توانند به طور همزمان با نمونه‌ای از موجودیت B در ارتباط باشند. به عبارت دیگر نمی‌توان نمونه‌هایی از موجودیت‌های C و D داشت که به طور همزمان با نمونه‌ای از موجودیت B در ارتباط هستند. به بیان دیگر همانطور که گفتیم در رابطه منفصل یا غیرپوشا یا تخصیص مجزا (Disjoint) ارتباط پدر با فرزندان فقط و فقط یک به یک است. و همچنین اشتراک نمونه موجودیت‌ها میان موجودیت‌های فرزند با یک نمونه موجودیت پدر همواره تهی است.

به عنوان مثالی دیگر هر نمونه از موجودیت ماشین که شماره شاسی یکتا و منحصر به فرد خود را دارد حتماً می‌بایست با یکی از نمونه موجودیت‌های نوع ماشین که دومحور (2WD) یا چهارمحور (4WD) است در ارتباط باشد. به عبارت دیگر نمی‌توان نمونه‌ای از موجودیت ماشین که شماره شاسی یکتا و منحصر به فرد خود را دارد داشت که با هیچ یک از نمونه موجودیت‌های نوع ماشین که دومحور (2WD) یا چهارمحور (4WD) است در ارتباط نباشد. این یعنی رابطه اجباری یا کلی یا کامل (Total).

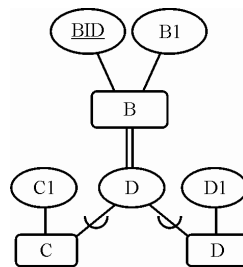
همچنین نمونه موجودیت‌های نوع ماشین که دومحور (2WD) یا چهارمحور (4WD) است نمی‌توانند به طور همزمان با نمونه‌ای از موجودیت ماشین که شماره شاسی یکتا و منحصر به فرد خود را دارد در ارتباط باشند، به عبارت دیگر نمی‌توان نمونه‌هایی از موجودیت‌های نوع ماشین که دومحور (2WD) یا چهارمحور (4WD) است داشت که به طور همزمان با نمونه‌ای از موجودیت ماشین که شماره شاسی یکتا و منحصر به فرد خود را دارد در ارتباط باشند. این یعنی رابطه منفصل یا غیرپوشا یا تخصیص مجزا (Disjoint).

از آنجاکه رابطه مابین موجودیت B و موجودیت‌های C و D یک رابطه اجباری یا کلی یا کامل (Total) است، پس رکوردهای حاوی محتوای مقدار NULL در ستون‌های مربوط به موجودیت B در طراحی به شکل مدل دو جدولی در جداول C و D به ازای یک نمونه موجودیت از B به دلیل عدم ارتباط با برخی از نمونه موجودیت‌های C و D ایجاد نمی‌گردد، که باعث شود این محتوای NULL در جداول C و D حاصل از عدم ارتباط برخی از نمونه موجودیت‌های موجودیت B با نمونه موجودیت‌های C و D در جدول B، به شکل مدل سه جدولی نگهداری شود.

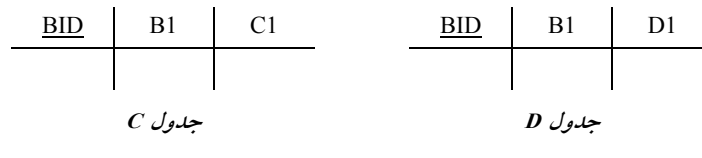
در حالت رابطه اجباری مابین موجودیت B و موجودیت‌های C و D به ازای هر نمونه از موجودیت B، حتماً نمونه موجودیتی از C یا D وجود دارد که با B رابطه برقرار کند، پس در این حالت طراحی بهینه این است که کل صفات موجودیت B در دو جدول موجودیت‌های C و D قرار داده شود و یک طراحی به شکل مدل دو جدولی ایجاد گردد، همچنین از آنجاکه رابطه مابین موجودیت‌های C و D و موجودیت B یک رابطه منفصل یا غیرپوشا یا تخصیص مجزا (Disjoint) است، پس رکوردهای تکراری در جداول C و D به ازای یک نمونه

موجودیت از موجودیت B ایجاد نمی‌گردد، که افزونگی حاصل از تکرار رکوردها در جداول C و D سبب شود رکورد نمونه موجودیت‌های B در جدول B نگهداری شود و یک مدل سه جدولی ایجاد گردد. پس در این حالت طراحی بهینه این است که کل صفات موجودیت B در دو جدول موجودیت‌های C و D قرار داده شود و یک طراحی به شکل مدل دو جدولی ایجاد گردد. به عبارت دیگر برای موجودیت پدر، رابطه جداگانه ایجاد نمی‌کنیم و ویژگی‌های رابطه پدر را به موجودیت‌های فرزند اضافه می‌کنیم. بنابراین گزینه‌های اول، دوم و چهارم را بطور کامل کنار می‌گذاریم، پس پُر واضح است که گزینه سوم پاسخ سوال است. بهمین سادگی در ادامه فرآیند نگاشت نمودار (ISA) به مدل رابطه‌ای را شرح می‌دهیم:

مدل تحلیل (نمودار ISA)



مدل طراحی (مدل رابطه‌ای)



۳۴- گزینه (۴) صحیح است.

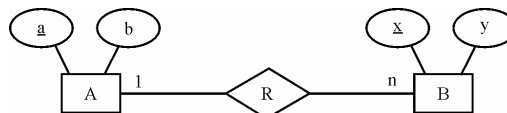
به طور کلی در مدل رابطه‌ای، هر موجودیت شناسایی شده در نمودار ER (مدل تحلیل) هنگام نگاشت به مدل رابطه‌ای (مدل طراحی) به یک جدول تبدیل می‌شود. همچنین صفات‌های موجودیت پس از نگاشت آن در مدل رابطه‌ای به صورت ستون‌های جدول بیان می‌شوند. همچنین ارتباط بین جدول از طریق کلید خارجی برقرار می‌گردد.

نگاشت رابطه یک به چند بین دو موجودیت به مدل رابطه‌ای

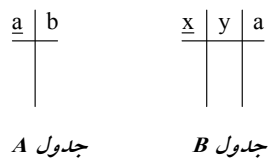
مستقل از اختیاری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد. و کلید کاندید جدول یک در جدول چند به عنوان کلید خارجی تعریف می‌گردد. همچنین صفات متصل به رابطه، درون جدول چند مستتر می‌شود.

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



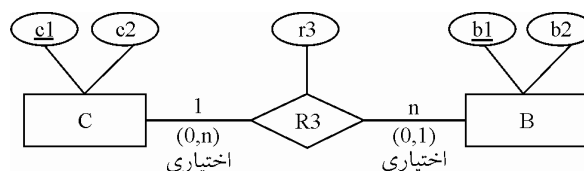
مدل طراحی:



بخشی از مدل EER رسم شده در صورت سوال، یک رابطه یک به چند بین دو موجودیت را نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را شرح می‌دهیم.

حالت اختیاری

مدل تحلیل:

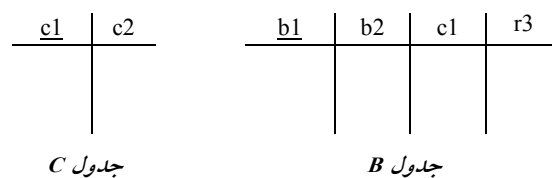


توجه: در شکل فوق صفت کلید $c1$ موجودیت C و صفت کلید $b1$ موجودیت B است.

توجه: نماد خط افقی نشانه اختیاری بودن موجودیت چسبیده به آن است.

توجه: قید $(0,N)$ نشان می‌دهد که هر نمونه موجودیت از C حداقل با صفر و حداکثر با N نمونه موجودیت از B ارتباط دارد و قید $(0,1)$ نشان می‌دهد که هر نمونه موجودیت از B حداقل با صفر و حداکثر با یک نمونه موجودیت از C ارتباط دارد.

مدل طراحی:



توجه: کلید کاندید جدول یک یعنی موجودیت C در جدول چند یعنی موجودیت B به عنوان کلید خارجی تعریف می‌گردد.

توجه: همچنین صفات متصل ($r3$) به رابطه، درون جدول چند یعنی موجودیت B مستتر می‌شود.

توجه: کلید کاندید جدول یک یعنی موجودیت C برابر همان کلید کاندید سابق در جدول موجودیت C است. یعنی کلید کاندید جدول C برابر ($c1$) است.

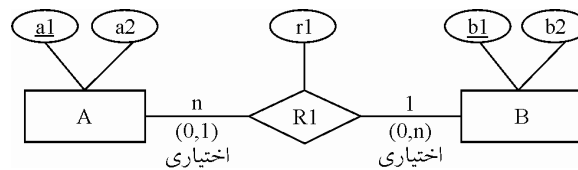
توجه: کلید کاندید جدول چند یعنی موجودیت B برابر همان کلید کاندید سابق در جدول موجودیت B است. یعنی کلید کاندید جدول B برابر ($b1$) است.

توجه: همانطور که واضح است، طراحی جدول B در گزینه چهارم به صورت $B(\underline{b1}, b2, c1, r3)$ در نظر گرفته شده است، همچنین طراحی جدول C در گزینه چهارم به صورت $C(\underline{c1}, c2)$ در نظر گرفته شده است

که مطابق آنچه بیان کردیم، طراحی درستی است. بنابراین گزینه‌های اول، دوم و سوم را کنار می‌گذاریم، پس تا همینجا پُر واضح است که گزینه چهارم پاسخ سوال است. به طور مجدد بخشی از مدل EER رسم شده در صورت سوال، یک رابطه یک به چند بین دو موجودیت را نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را شرح می‌دهیم.

حالت اختیاری

مدل تحلیل:

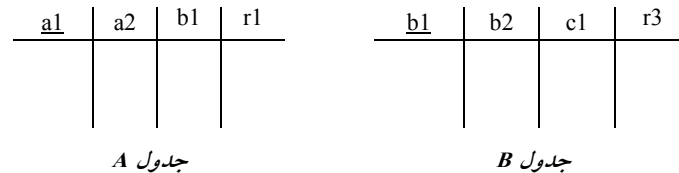


توجه: در شکل فوق صفت کلید $a1$ موجودیت A و صفت کلید $b1$ موجودیت B است.

توجه: نماد خط افقی نشانه اختیاری بودن موجودیت چسبیده به آن است.

توجه: قید $(0,1)$ نشان می‌دهد که هر نمونه موجودیت از A حداقل با صفر و حداکثر با یک نمونه موجودیت از B ارتباط دارد و قید $(0,N)$ نشان می‌دهد که هر نمونه موجودیت از B حداقل با صفر و حداکثر با N نمونه موجودیت از A ارتباط دارد.

مدل طراحی:



توجه: کلید کاندید جدول یک یعنی موجودیت B در جدول چند یعنی موجودیت A به عنوان کلید خارجی تعریف می‌گردد.

توجه: همچنین صفات متصل $(r1)$ به رابطه، درون جدول چند یعنی موجودیت A مستتر می‌شود.

توجه: کلید کاندید جدول یک یعنی موجودیت B برابر همان کلید کاندید سابق در جدول موجودیت B است. یعنی کلید کاندید جدول B برابر $(b1)$ است.

توجه: کلید کاندید جدول چند یعنی موجودیت A برابر همان کلید کاندید سابق در جدول موجودیت A است. یعنی کلید کاندید جدول A برابر $(a1)$ است.

توجه: جدول B در نگاشت مرحله قبل ایجاد شده است که در اینجا دقیقاً به همان شکل و همان مشخصات، استفاده شده است.

توجه: همانطور که واضح است، طراحی جدول B در گزینه چهارم به صورت $B(b1, b2, c1, r3)$ در نظر گرفته شده است، همچنین طراحی جدول A در گزینه چهارم به صورت $A(a1, a2, b1, r1)$ در نظر گرفته شده است که مطابق آنچه بیان کردیم، طراحی درستی است. بنابراین گزینه‌های اول، دوم و سوم را کنار

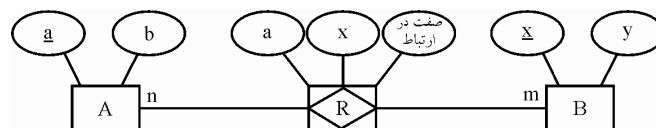
می‌گذاریم، پس تا همینجا پُر واضح است که گزینه چهارم پاسخ سوال است.

نگاشت رابطه چند به چند بین دو موجودیت به مدل رابطه‌ای

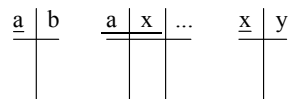
مستقل از اختیاری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد و یک جدول پُل (Bridge) نیز به عنوان ارتباط دهنده دو جدول مورد استفاده قرار می‌گیرد. همچنین کلید کاندید جدول پُل از ترکیب کلید کاندید دو جدول دیگر ایجاد می‌گردد. همچنین صفات متصل به رابطه، درون جدول پُل مستتر می‌شود.

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



مدل طراحی:

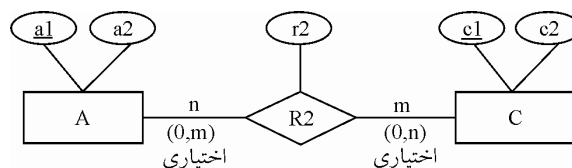


جدول A جدول AB جدول B

توجه: ستون a در جدول AB به عنوان کلید خارجی تعریف می‌گردد که به جدول A ارجاع می‌کند. همچنین ستون x در جدول AB به عنوان کلید خارجی تعریف می‌گردد که به جدول B ارجاع می‌کند. بخشی از مدل EER رسم شده در صورت سوال، یک رابطه چند به چند بین دو موجودیت را نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را شرح می‌دهیم.

حالت اختیاری

مدل تحلیل:



توجه: در شکل فوق صفت a1 کلید موجودیت A و صفت c1 کلید موجودیت C است.

توجه: نماد خط افقی نشانه اختیاری بودن موجودیت چسبیده به آن است.

توجه: قید (0,m) نشان می‌دهد که هر نمونه موجودیت از A حداقل با صفر و حداکثر با M نمونه موجودیت از C ارتباط دارد و قید (0,n) نشان می‌دهد که هر نمونه موجودیت از C حداقل با صفر و حداکثر با N نمونه موجودیت از A ارتباط دارد.

مدل طراحی:

<u>a1</u>	a2	b1	r1
جدول A			

<u>a1</u>	<u>c1</u>	r2
جدول r2		

<u>c1</u>	c2
جدول C	

توجه: ستون a1 در جدول r2 به عنوان کلید خارجی تعریف می‌گردد که به جدول A ارجاع می‌کند. همچنین ستون c1 در جدول r2 به عنوان کلید خارجی تعریف می‌گردد که به جدول C ارجاع می‌کند.

توجه: همچنین صفات متصل (r2) به رابطه، درون جدول پُل یعنی جدول r2 مستتر می‌شود.

توجه: همچنین کلید کاندید جدول پُل از ترکیب کلید کاندید دو جدول دیگر ایجاد می‌گردد. یعنی کلید کاندید جدول r2 برابر (a1,c1) است.

توجه: کلید کاندید جدول چند چپ یعنی موجودیت A برابر همان کلید کاندید سابق در جدول موجودیت A است. یعنی کلید کاندید جدول A برابر (a1) است.

توجه: کلید کاندید جدول چند راست یعنی موجودیت C برابر همان کلید کاندید سابق در جدول موجودیت C است. یعنی کلید کاندید جدول C برابر (c1) است.

توجه: جدول A در نگاهت مرحله قبل ایجاد شده است که در اینجا دقیقاً به همان شکل و همان مشخصات، استفاده شده است.

توجه: جدول C در نگاهت مرحله قبل ایجاد شده است که در اینجا دقیقاً به همان شکل و همان مشخصات، استفاده شده است.

توجه: همانطور که واضح است، طراحی جدول C در گزینه چهارم به صورت C(c1,c2) در نظر گرفته شده است، همچنین طراحی جدول A در گزینه چهارم به صورت A(a1,a2,b1,r1) در نظر گرفته شده است، همچنین طراحی جدول r2 در گزینه چهارم به صورت R2(a1,c1,r2) در نظر گرفته شده است که مطابق آنچه بیان کردیم، طراحی درستی است. بنابراین گزینه‌های اول، دوم و سوم را به طور کامل کنار می‌گذاریم، پس پرواضح است که گزینه چهارم پاسخ سوال است. به همین سادگی.

۳۵- گزینه (۱) صحیح است.

به طور کلی در مدل رابطه‌ای، هر موجودیت شناسایی شده در نمودار ER (مدل تحلیل) هنگام نگاهت به مدل رابطه‌ای (مدل طراحی) به یک جدول تبدیل می‌شود. همچنین صفات‌های موجودیت پس از نگاهت آن در مدل رابطه‌ای به صورت ستون‌های جدول بیان می‌شوند. همچنین ارتباط بین جدول از طریق کلید خارجی برقرار می‌گردد.

توجه: جهت تبدیل مدل تحلیل به مدل رابطه‌ای از سمت و موجودیتی شروع کنید که ورودی صفت از موجودیت دیگری نداشته باشد.

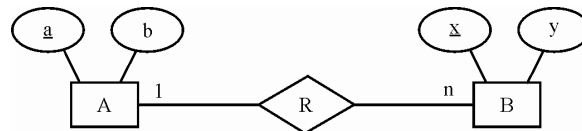
نگاشت رابطه یک به چند بین دو موجودیت به مدل رابطه‌ای

مستقل از اختیاری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد. و کلید

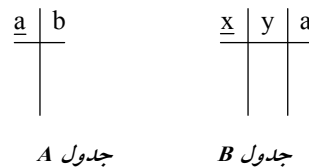
کاندید جدول یک در جدول چند به عنوان کلید خارجی تعریف می‌گردد. همچنین صفات متصل به رابطه، درون جدول چند مستتر می‌شود.

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



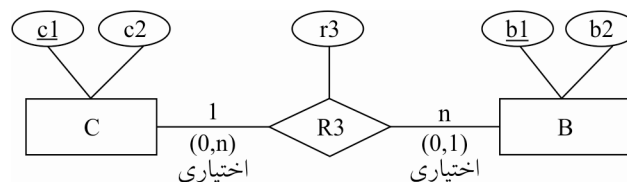
مدل طراحی:



بخشی از مدل EER رسم شده در صورت سوال، یک رابطه یک به چند بین دو موجودیت را نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را شرح می‌دهیم.

حالت اختیاری

مدل تحلیل:

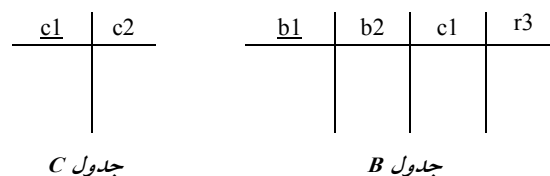


توجه: در شکل فوق صفت c_1 کلید موجودیت C و صفت b_1 کلید موجودیت B است.

توجه: نماد خط افقی نشانه اختیاری بودن موجودیت چسبیده به آن است.

توجه: قید $(0,N)$ نشان می‌دهد که هر نمونه موجودیت از C حداقل با صفر و حداکثر با N نمونه موجودیت از B ارتباط دارد و قید $(0,1)$ نشان می‌دهد که هر نمونه موجودیت از B حداقل با صفر و حداکثر با یک نمونه موجودیت از C ارتباط دارد.

مدل طراحی:



توجه: کلید کاندید جدول یک یعنی موجودیت C در جدول چند یعنی موجودیت B به عنوان کلید خارجی تعریف می‌گردد.

توجه: همچنین صفات متصل (r_3) به رابطه، درون جدول چند یعنی موجودیت B مستتر می‌شود.

توجه: کلید کاندید جدول یک یعنی موجودیت C برابر همان کلید کاندید سابق در جدول موجودیت C است. یعنی کلید کاندید جدول C برابر (c_1) است.

توجه: کلید کاندید جدول چند یعنی موجودیت B برابر همان کلید کاندید سابق در جدول موجودیت B است. یعنی کلید کاندید جدول B برابر (b_1) است.

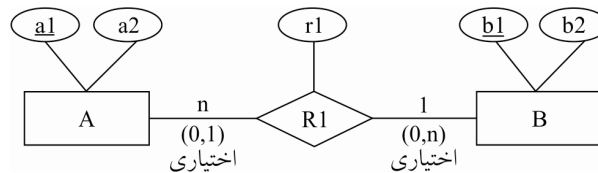
توجه: همانطور که واضح است، طراحی جدول B در گزینه اول به صورت $B(b_1, b_2, c_1, r_3)$ در نظر گرفته شده است، که مطابق آنچه بیان کردیم، طراحی درستی است. بنابراین گزینه‌های دوم، سوم و چهارم را کنار می‌گذاریم، پس تا همینجا پُر واضح است که گزینه اول پاسخ سوال است.

توجه: دقت کنید که در گزینه دوم طراحی جدول B به صورت $B(b_1, b_2, c_1, r_3)$ در نظر گرفته شده است که طراحی نادرستی است، زیرا کلید جدول B را ترکیب دو ستون (b_1, r_3) در نظر گرفته است که مطابق آنچه بیان کردیم، طراحی نادرستی است.

به طور مجدد بخشی از مدل EER رسم شده در صورت سوال، یک رابطه یک به چند بین دو موجودیت را نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را شرح می‌دهیم.

حالت اختیاری

مدل تحلیل:

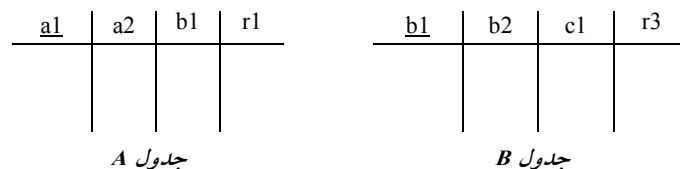


توجه: در شکل فوق صفت کلید a_1 موجودیت A و صفت b_1 کلید موجودیت B است.

توجه: نماد خط افقی نشانه اختیاری بودن موجودیت چسبیده به آن است.

توجه: قید (0,1) نشان می‌دهد که هر نمونه موجودیت از A حداقل با صفر و حداکثر با یک نمونه موجودیت از B ارتباط دارد و قید (0,N) نشان می‌دهد که هر نمونه موجودیت از B حداقل با صفر و حداکثر با N نمونه موجودیت از A ارتباط دارد.

مدل طراحی:



توجه: کلید کاندید جدول یک یعنی موجودیت B در جدول چند یعنی موجودیت A به عنوان کلید

خارجی تعریف می‌گردد.

توجه: همچنین صفات متصل ($r1$) به رابطه، درون جدول چند یعنی موجودیت A مستتر می‌شود.

توجه: کلید کاندید جدول یک یعنی موجودیت B برابر همان کلید کاندید سابق در جدول موجودیت B است. یعنی کلید کاندید جدول B برابر ($b1$) است.

توجه: کلید کاندید جدول چند یعنی موجودیت A برابر همان کلید کاندید سابق در جدول موجودیت A است. یعنی کلید کاندید جدول A برابر ($a1$) است.

توجه: جدول B در نگاهت مرحله قبل ایجاد شده است که در اینجا دقیقاً به همان شکل و همان مشخصات، استفاده شده است.

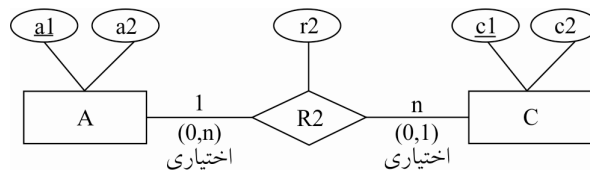
توجه: همانطور که واضح است، طراحی جدول B در گزینه اول به صورت $B(b1, b2, c1, r3)$ در نظر گرفته شده است، همچنین طراحی جدول A در گزینه اول به صورت $A(a1, a2, b1, r1)$ در نظر گرفته شده است که مطابق آنچه بیان کردیم، طراحی درستی است. بنابراین گزینه‌های دوم، سوم و چهارم را کنار می‌گذاریم، پس تا همینجا پُر واضح است که گزینه اول پاسخ سوال است.

توجه: دقت کنید که در گزینه دوم طراحی جدول A به صورت $A(a1, a2, b1, r1)$ در نظر گرفته شده است که طراحی نادرستی است زیرا کلید جدول A را ترکیب دو ستون $(a1, r1)$ در نظر گرفته است که مطابق آنچه بیان کردیم، طراحی نادرستی است.

و در نهایت به طور مجدد بخشی از مدل EER رسم شده در صورت سوال، یک رابطه یک به چند بین دو موجودیت را نشان می‌دهد. که در ادامه فرآیند نگاهت آن به مدل رابطه‌ای را شرح می‌دهیم.

حالت اختیاری

مدل تحلیل:

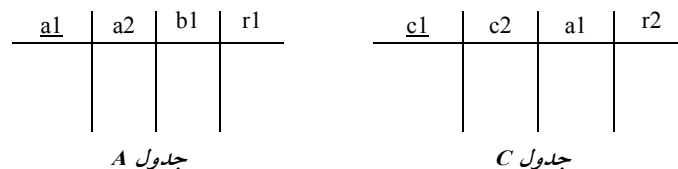


توجه: در شکل فوق صفت $a1$ کلید موجودیت A و صفت $c1$ کلید موجودیت C است.

توجه: نماد خط افقی نشانه اختیاری بودن موجودیت چسبیده به آن است.

توجه: قید $(0,n)$ نشان می‌دهد که هر نمونه موجودیت از A حداقل با صفر و حداکثر با N نمونه موجودیت از C ارتباط دارد و قید $(0,1)$ نشان می‌دهد که هر نمونه موجودیت از C حداقل با صفر و حداکثر با 1 نمونه موجودیت از A ارتباط دارد.

مدل طراحی:



توجه: کلید کاندید جدول یک یعنی موجودیت A در جدول چند یعنی موجودیت C به عنوان کلید خارجی تعریف می‌گردد.

توجه: همچنین صفات متصل (r2) به رابطه، درون جدول چند یعنی موجودیت C مستتر می‌شود.

توجه: کلید کاندید جدول یک یعنی موجودیت A برابر همان کلید کاندید سابق در جدول موجودیت A است. یعنی کلید کاندید جدول A برابر (a1) است.

توجه: کلید کاندید جدول چند یعنی موجودیت C برابر همان کلید کاندید سابق در جدول موجودیت C است. یعنی کلید کاندید جدول C برابر (c1) است.

توجه: جدول A در نگاشت مرحله قبل ایجاد شده است که در اینجا دقیقاً به همان شکل و همان مشخصات، استفاده شده است.

توجه: همانطور که واضح است، طراحی جدول C در گزینه اول به صورت $C(c_1, c_2, a_1, r_2)$ در نظر گرفته شده است، همچنین طراحی جدول A در گزینه اول به صورت $A(a_1, a_2, b_1, r_1)$ در نظر گرفته شده است و در نهایت طراحی جدول B در گزینه اول به صورت $B(b_1, b_2, c_1, r_3)$ در نظر گرفته شده است که مطابق آنچه بیان کردیم، طراحی درستی است. بنابراین گزینه‌های دوم، سوم و چهارم را به طور کامل کنار می‌گذاریم، پس پُر واضح است که گزینه اول پاسخ سوال است. به همین سادگی.

توجه: دقت کنید که در گزینه دوم طراحی جدول C به صورت $C(c_1, c_2, a_1, r_2)$ در نظر گرفته شده است که طراحی نادرستی است زیرا کلید جدول C را ترکیب دو ستون (c_1, r_2) در نظر گرفته است که مطابق آنچه بیان کردیم، طراحی نادرستی است.

۳۶- گزینه (۴) صحیح است.

به طور کلی در مدل رابطه‌ای، هر موجودیت شناسایی شده در نمودار ER (مدل تحلیل) هنگام نگاشت به مدل رابطه‌ای (مدل طراحی) به یک جدول تبدیل می‌شود. همچنین صفات‌های موجودیت پس از نگاشت آن در مدل رابطه‌ای به صورت ستون‌های جدول بیان می‌شوند. همچنین ارتباط بین جدول از طریق کلید خارجی برقرار می‌گردد.

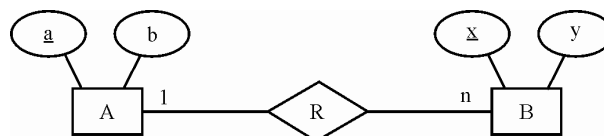
توجه: جهت تبدیل مدل ER (مدل تحلیل) به مدل مدل رابطه‌ای (مدل طراحی یا طراحی منطقی) از سمت و موجودیتی شروع کنید که ورودی صفت از موجودیت دیگری نداشته باشد.

نگاشت رابطه یک به چند بین دو موجودیت به مدل رابطه‌ای

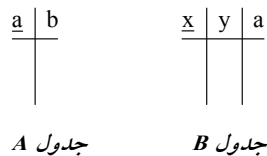
مستقل از اختیاری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد. و کلید کاندید جدول یک در جدول چند به عنوان کلید خارجی تعریف می‌گردد.

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



مدل طراحی:



وابستگی وجودی

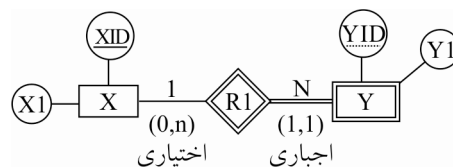
اگر در یک بانک اطلاعاتی، وجود یک موجودیت، وابسته به موجودیت دیگری باشد که در صورت حذف و تغییر موجودیت اصلی یعنی موجودیت قوی این موجودیت نیز تغییر کند، این نوع وابستگی را وابستگی وجودی گفته و به پدیده وابسته، **موجودیت ضعیف** گویند. همچنین **موجودیت ضعیف کلید موجودیت قوی** را در بر دارد تا هرگونه تغییر یا حذف در موجودیت قوی به موجودیت ضعیف اعمال شود.

توجه: موجودیت ضعیف با دو مستطیل تو در تو نمایش داده می‌شود.

بخشی از مدل EER رسم شده در صورت سوال، یک رابطه یک به چند بین دو موجودیت قوی و ضعیف را نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را شرح می‌دهیم.

حالت اجباری و اختیاری

مدل تحلیل:



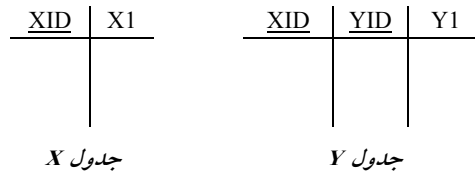
توجه: در شکل فوق صفت XID کلید موجودیت قوی X و صفت YID، صفت ممیزه موجودیت ضعیف Y است. صفت ممیزه موجودیت ضعیف توسط نقطه چین نشان داده می‌شود.

توجه: نماد خط مضاعف افقی نشانه اجباری بودن موجودیت چسبیده به آن است، اما نماد | به معنی یک و الزام شرکت در رابطه نشانه اجباری بودن موجودیت طرف مقابل است.

توجه: نماد خط افقی نشانه اختیاری بودن موجودیت چسبیده به آن است، اما نماد دایره کوچک توخالی به معنی صفر و عدم الزام شرکت در رابطه نشانه اختیاری بودن موجودیت طرف مقابل است.

توجه: قید (0,N) نشان می‌دهد که هر نمونه موجودیت از X حداقل با صفر و حداکثر با N نمونه موجودیت از Y ارتباط دارد و قید (1,1) نشان می‌دهد که هر نمونه موجودیت از Y حداقل با یک و حداکثر با یک نمونه موجودیت از X ارتباط دارد.

مدل طراحی:



توجه: کلید کاندید جدول یک یعنی موجودیت قوی در جدول چند یعنی موجودیت ضعیف به عنوان کلید خارجی تعریف می‌گردد.

توجه: کلید کاندید جدول چند یعنی موجودیت ضعیف برابر ترکیب کلید خارجی و صفت ممیزه در جدول موجودیت ضعیف است. یعنی کلید کاندید جدول Y برابر (XID, YID) است.

توجه: صفت ممیزه یا کلید جزئی به طور سراسری در یک موجودیت ضعیف یکتا نیست، بلکه فقط در بین نمونه‌ها یا دسته‌هایی که با موجودیت قوی ارتباط دارند، یکتا است.

توجه: یک موجودیت ضعیف همیشه در ارتباطش با موجودیت قوی رابطه اجباری دارد.

توجه: جهت تبدیل مدل ER (مدل تحلیل) به مدل مدل رابطه‌ای (مدل طراحی یا طراحی منطقی) از سمت و موجودیتی شروع کنید که ورودی صفت از موجودیت دیگری نداشته باشد.

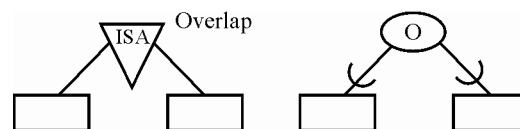
نگاشت رابطه ISA یا وراثت به مدل رابطه‌ای

در رابطه ISA رابطه پدر با فرزندان به دو صورت رابطه اختیاری یا جزئی یا بخشی (Partial) با نماد خط عمودی و رابطه اجباری یا کلی یا کامل (Total) با نماد خط مضاعف عمودی است و رابطه فرزندان با پدر به دو صورت رابطه متصل یا پوشا یا تخصیص غیرمجزا (Overlap) و رابطه منفصل یا غیرپوشا یا تخصیص مجزا (Disjoint) می‌باشد.

توجه: در یک رابطه اجباری یا کلی (Total)، هر نمونه از موجودیت پدر حتماً می‌بایست با یکی از نمونه موجودیت‌های فرزند در ارتباط باشد.

توجه: در یک رابطه اختیاری یا جزئی (Partial)، هر نمونه از موجودیت پدر می‌تواند با یکی از نمونه موجودیت‌های فرزند در ارتباط باشد یا نباشد.

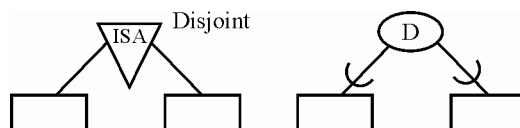
رابطه متصل یا پوشا یا تخصیص غیرمجزا (Overlap) مابین فرزندان و پدر به دو شیوه زیر نشان داده می‌شود:



توجه: در رابطه متصل یا پوشا یا تخصیص غیرمجزا (Overlap) ارتباط پدر با فرزندان می‌تواند یک به یک باشد و همچنین می‌تواند یک به چند باشد. و همچنین اشتراک نمونه موجودیت‌ها میان موجودیت‌های فرزند با یک نمونه موجودیت از پدر می‌تواند تهی باشد و همچنین می‌تواند غیرتهی باشد.

توجه: در یک رابطه متصل یا پوشا یا تخصیص غیرمجزا (Overlap)، نمونه موجودیت‌های فرزند می‌توانند

به طور همزمان با نمونه‌ای از موجودیت پدر در ارتباط باشند که این یعنی همان رابطه یک به چند میان پدر و فرزندان.
رابطه منفصل یا غیرپوشا یا تخصیص مجزا (Disjoint) مابین فرزندان و پدر به دو شیوه زیر نشان داده می‌شود:



توجه: در رابطه منفصل یا غیرپوشا یا تخصیص مجزا (Disjoint) ارتباط پدر با فرزندان فقط و فقط یک به یک است. و همچنین اشتراک نمونه موجودیت‌ها میان موجودیت‌های فرزند با یک نمونه موجودیت از پدر همواره تهی است.

توجه: در یک رابطه منفصل یا غیرپوشا یا تخصیص مجزا (Disjoint)، نمونه موجودیت‌های فرزند نمی‌توانند به طور همزمان با نمونه‌ای از موجودیت پدر در ارتباط باشند که این یعنی همان رابطه یک به یک میان پدر و فرزندان.

بخشی از مدل EER رسم شده در صورت سوال، یک رابطه اجباری یا کلی یا کامل (Total) مابین موجودیت Y و موجودیت‌های A و B و یک رابطه منفصل یا غیرپوشا یا تخصیص مجزا (Disjoint) را مابین موجودیت‌های A و B و موجودیت Y نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را بیان می‌کنیم.

توجه: در یک رابطه اجباری یا کلی یا کامل (Total)، هر نمونه از موجودیت پدر حتماً می‌بایست با یکی از نمونه موجودیت‌های فرزند در ارتباط باشد. برای مثال در این سؤال، هر نمونه از موجودیت Y حتماً می‌بایست با یکی از نمونه موجودیت‌های A یا B در ارتباط باشد. به عبارت دیگر نمی‌توان نمونه‌ای از موجودیت Y داشت که با هیچ یک از نمونه موجودیت‌های A یا B در ارتباط نیست.

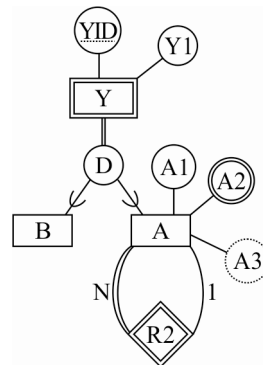
همچنین در یک رابطه منفصل یا غیرپوشا یا تخصیص مجزا (Disjoint)، نمونه موجودیت‌های فرزند نمی‌توانند به طور همزمان با نمونه‌ای از موجودیت پدر در ارتباط باشند. برای مثال در این سؤال، نمونه موجودیت‌های A و B نمی‌توانند به طور همزمان با نمونه‌ای از موجودیت Y در ارتباط باشند. به عبارت دیگر نمی‌توان نمونه‌هایی از موجودیت‌های A و B داشت که به طور همزمان با نمونه‌ای از موجودیت Y در ارتباط هستند. به بیان دیگر همانطور که گفتیم در رابطه منفصل یا غیرپوشا یا تخصیص مجزا (Disjoint) ارتباط پدر با فرزندان فقط و فقط یک به یک است. و همچنین اشتراک نمونه موجودیت‌ها میان موجودیت‌های فرزند با یک نمونه موجودیت از پدر همواره تهی است.

توجه: از آنجاکه رابطه مابین موجودیت Y و موجودیت‌های A و B یک رابطه اجباری یا کلی یا کامل (Total) است، پس رکوردهای حاوی محتوای مقدار NULL در ستون‌های مربوط به موجودیت Y در طراحی به شکل مدل دو جدولی در جداول A و B به ازای یک نمونه موجودیت از Y به دلیل عدم ارتباط با برخی از نمونه موجودیت‌های A و B ایجاد نمی‌گردد، که باعث شود این محتوای NULL در جداول A و B حاصل از عدم ارتباط برخی از نمونه موجودیت‌های موجودیت Y با نمونه موجودیت‌های A و B در جدول Y، به شکل مدل سه جدولی نگهداری شود.

توجه: در حالت رابطه اجباری مابین موجودیت Y و موجودیت‌های A و B به ازای هر نمونه از موجودیت Y، حتماً نمونه موجودیتی از A یا B وجود دارد که با Y رابطه برقرار کند، پس در این حالت طراحی بهینه این است که کل صفات موجودیت Y در دو جدول موجودیت‌های A و B قرار داده شود و یک طراحی به شکل مدل دو جدولی ایجاد گردد، همچنین از آنجاکه رابطه مابین موجودیت‌های A و B و موجودیت Y یک رابطه منفصل یا غیرپوشا یا تخصیص مجزا (Disjoint) است، پس رکوردهای تکراری در جداول A و B به ازای یک نمونه موجودیت از موجودیت Y ایجاد نمی‌گردد، که افزونگی حاصل از تکرار رکوردها در جداول A و B سبب شود رکورد نمونه موجودیت‌های Y در جدول Y نگهداری شود و یک مدل سه جدولی ایجاد گردد. پس در این حالت طراحی بهینه این است که کل صفات موجودیت Y در دو جدول موجودیت‌های A و B قرار داده شود و یک طراحی به شکل مدل دو جدولی ایجاد گردد. به عبارت دیگر برای موجودیت پدر، رابطه جداگانه ایجاد نمی‌کنیم و کل ویژگی‌های رابطه پدر را به موجودیت‌های فرزند اضافه می‌کنیم. در یک عبارت ساده‌تر و واضح‌تر در شرایط پدر اجباری (Total) و فرزندان غیرپوشا (Disjoint)، کل صفات باباشو بیارید جلوی چشم بچه‌هاش، به همین سادگی.

در ادامه فرآیند نگاشت نمودار (ISA) به مدل رابطه‌ای را شرح می‌دهیم:

مدل تحلیل (نمودار (ISA))



مدل طراحی (مدل رابطه‌ای)

همانطور که در مدل طراحی قبل تر گفتیم، مدل طراحی جدول Y به صورت (XID, YID, Y1) در نظر گرفته شد.

XID	YID	Y1	...	XID	YID	Y1	A1	A2
جدول B				جدول غیرنرمال A				

توجه: چون رابطه موجودیت Y با موجودیت‌های A و B اجباری است، و رابطه موجودیت‌های A و B با موجودیت Y از نوع Disjoint است. و به ازای هر نمونه موجودیت از Y حتماً یک نمونه موجودیت از A و یا B وجود دارد و به تبع عدم مقادیر NULL جلوی نمونه موجودیت‌های Y و عدم نیاز به عمل‌الحاق، نگاشت دو جدولی فوق به عنوان یک تبدیل بهینه توصیه می‌گردد.

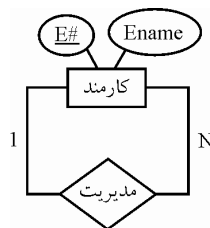
نگاشت رابطه یک به چند در یک موجودیت به مدل رابطه‌ای

مستقل از اختیاری یا اجباری بودن موجودیت، موجودیت به یک جدول تبدیل می‌گردد.

مثال: فرض کنید در یک سازمان هر کارمند، تحت مدیریت یک کارمند دیگر باشد.

توجه: یک رئیس چندین کارمند دارد ولی هر کارمند حداکثر یک رئیس دارد.

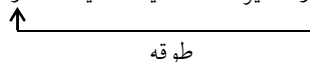
مدل تحلیل:



مدل طراحی:

<u>E#</u>	Sname	M#
S1	Sn1	NULL
S2	Sn2	S1
S3	Sn3	S1
S4	Sn4	S3
S5	Sn5	S2
S6	Sn6	S5

کلید خارجی (شماره مدیر) کلید کاندید (شماره کارمندی)

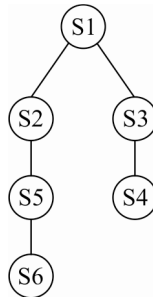


توجه: سطر مربوط به مدیر کل، که مدیر بالاتری ندارد، مقدار NULL در فیلد M# خود می‌گیرد.

توجه: شماره مدیر، برای شخصی با شماره کارمندی S4 برابر S3 است و نام S3 برابر Sn3 است.

توجه: مطابق تعریف کلید خارجی، کلید خارجی باید به کلید کاندید همان رابطه یا رابطه دیگر، ارجاع کند، در واقع لزومی ندارد که روابط فرضی R1 و R2 از هم مجزا باشند، تا کلید خارجی در یک رابطه باشد و کلید کاندید در رابطه‌ای دیگر. زیرا هم کلید کاندید و هم کلید خارجی می‌توانند در یک رابطه باشند، یعنی R1 و R2 می‌توانند یک رابطه باشند، بنابراین کلید خارجی یک رابطه می‌تواند متناظر با کلید کاندید همان رابطه باشد. در واقع دیاگرام ارجاعی دارای طوقه است به این معنی که مبداء و مقصد کلید خارجی با هم یکی است.

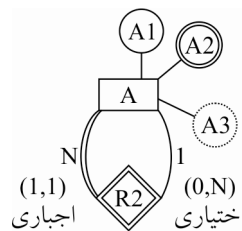
توجه: مدل ریاضی نگاشت رابطه یک به چند در یک موجودیت به مدل رابطه‌ای یک ساختار درختی است، شامل گره ریشه (مدیر کل)، پدر(ناظر) و فرزندان (زیردست).



بخشی از مدل EER رسم شده در صورت سوال، یک رابطه یک به چند در یک موجودیت را نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را شرح می‌دهیم.

حالت اجباری و اختیاری

مدل تحلیل:



توجه: در شکل فوق و بر اساس نوع نمایش و نشانه‌گذاری‌ها صفت A2 چند مقداری و صفت A3 مشتق (پویا) است. همانطور که قبل‌تر هم گفتیم کلید کاندید موجودیت A صفات (XID, YID) است.

توجه: نماد خط مضعف افقی نشانه اجباری بودن موجودیت چسبیده به آن است، اما نماد | به معنی یک و الزام شرکت در رابطه نشانه اجباری بودن موجودیت طرف مقابل است.

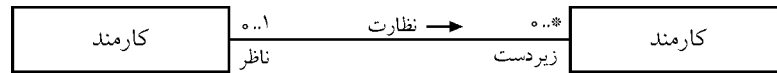
توجه: نماد خط افقی نشانه اختیاری بودن موجودیت چسبیده به آن است، اما نماد دایره کوچک توخالی به معنی صفر و عدم الزام شرکت در رابطه نشانه اختیاری بودن موجودیت طرف مقابل است.

مدل‌سازی رابطه خودانجمنی (Self-Association)

یک رابطه‌ی خودانجمنی، رابطه‌ای است که یک موجودیت با خودش دارد. لذا اگر نمونه موجودیت‌های یک موجودیت با یکدیگر در ارتباط باشند، ارتباط مذکور از نوع خودانجمنی است. به عبارت دیگر، اگر ابتدا و انتهای یک رابطه انجمنی به یک موجودیت اشاره داشته باشد، در این صورت رابطه‌ی مذکور از نوع خودانجمنی خواهد بود. در شکل زیر، دو مثال برای رابطه خودانجمنی (بازتابی) ارائه شده است. توجه داشته باشید که در صورت مشخص کردن نقش دو طرف رابطه، ذکر نام رابطه، ضروری نیست.



توجه: برای سادگی در نحوه خواندن رابطه خودانجمنی توصیه می‌کنیم، رابطه را به شکل خطی ایجاد کنید، سپس رابطه را بخوانید:

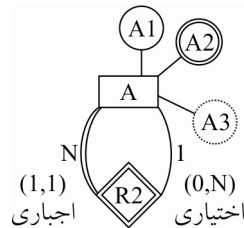


شکل فوق گویای رابطه خودانجمنی (Self Association) است. به این معنی که یک کارمند هیچ زیردستی (فرزند) ندارد یعنی برگ درخت است (چون مدیر نیست) یا چندین زیردست (فرزند) دارد (چون مدیر است) همچنین یک کارمند هیچ ناظری (پدر) ندارد (چون مدیر کل است) یا حداکثر یک ناظر (پدر) دارد (چون مدیر کل نیست).

توجه: یک مدیر (رئیس) چندین کارمند دارد ولی هر کارمند حداکثر یک رئیس دارد.

توجه: جهت درک بیشتر در ساختار درختی، یکبار بالا به پایین نگاه کنید، ببیند یک گره مورد نظر، فرزند (زیر دست) دارد یا ندارد، اگر یک گره مورد نظر، فرزند نداشت پس کارمند است، اگر فرزند داشت پس مدیر است. حالا یکبار پایین به بالا نگاه کنید، ببینید یک گره مورد نظر، پدر (ناظر) دارد یا ندارد، اگر یک گره مورد نظر، پدر (ناظر) نداشت پس ریشه و مدیر کل است، اگر پدر (ناظر) داشت پس فرزند است و مدیر کل نیست یعنی کارمند یا مدیر است.

توجه: در مدل مطرح شده در صورت سوال که به صورت زیر است:



قید (0,N) نشان می‌دهد که هر نمونه موجودیت از A حداقل با صفر و حداکثر با N نمونه موجودیت از A ارتباط دارد و قید (1,1) نشان می‌دهد که هر نمونه موجودیت از A حداقل با یک و حداکثر با یک نمونه موجودیت از A ارتباط دارد.

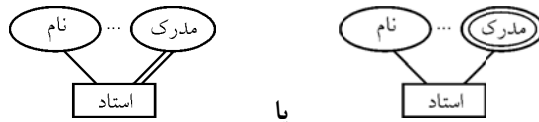
توجه: تعریف دقیق‌تر، قید (0,N) به این معنی است که هر گره (نمونه موجودیت) مورد نظر، حداقل صفر فرزند (زیر دست) و حداکثر N فرزند دارد و قید (1,1) به این معنی است که هر گره (نمونه موجودیت) مورد نظر، حداقل یک و حداکثر یک پدر (ناظر) دارد، در واقع در شکل فوق، ناظر مدیر کل خود مدیر کل است. و سایر گره‌های مورد نظر نیز حداقل یک و حداکثر یک ناظر را مطابق فرض سوال و خط مضاعف افقی به طور اجباری دارند.

صفت تک مقداری و چند مقداری

بعضی از صفات چه ساده و چه مرکب فقط می‌توانند یک مقدار را بگیرند که به این صفات، صفت تک‌مقداری می‌گویند. مانند شماره دانشجویی که نمی‌تواند بیش از یک مقدار داشته باشد. این صفات در نمودار ER بصورت معمول نمایش داده می‌شوند.

صفاتی وجود دارند که می‌توانند چندین مقدار را بگیرند مانند صفت مدرک در موجودیت استاد که می‌تواند مقادیر لیسانس، فوق لیسانس و یا دکتری را در خود بگیرد. صفت چند مقداری در نمودار ER بصورت دو دایره مضاعف و یا خط مضاعف متصل شده به آن نمایش داده می‌شود.

مثال:



یا

توجه: به مثال‌های زیر توجه کنید.

صفت ساده تک‌مقداری: مانند کدملی

صفت ساده چندمقداری: مانند مدرک تحصیلی

صفت مرکب تک‌مقداری: مانند تاریخ تولد

صفت مرکب چندمقداری: مانند آدرس

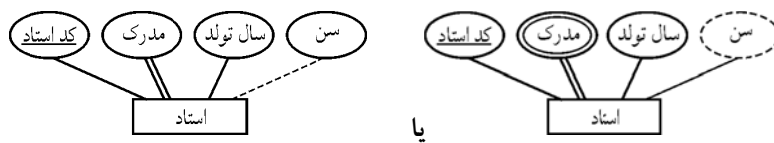
توجه: در بانک اطلاعاتی مبتنی بر مدل رابطه‌ای (جدولی) صفت چندمقداری نداریم.

صفت مشتق (پویا)

صفتی است که در موجودیت وجود خارجی ندارد ولی در صورت لزوم می‌توان آنرا بدست آورد. صفتی که مقادیر آن مدام در حال تغییر و تحول باشد، صفت پویا یا مشتق محسوب می‌گردد. بنابراین به دلیل تغییرات مداوم، توصیه می‌گردد صفت پویا در جداول بانک اطلاعات مورد استفاده قرار نگیرد و مقدار آن از طریق صفت مرتبط با آن محاسبه گردد. برای مثال برای محاسبه صفت سن، می‌توان صفت تاریخ تولد را در نظر گرفت و از روی این صفت، سن را محاسبه نمود.

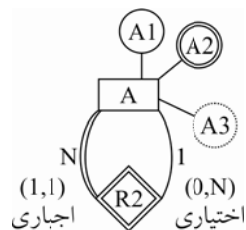
توجه: صفت مشتق را در نمودار ER با نقطه چین به موجودیت مورد نظر متصل می‌کنند.

مثال:



یا

همانطور که گفتیم بخشی از مدل EER رسم شده در صورت سوال به صورت زیر است:



توجه: در شکل فوق و بر اساس نوع نمایش و نشانه‌گذاری‌ها صفت A2 چند مقداری و صفت A3 مشتق (پویا) است.

مدل طراحی غیرنرمال:

<u>XID</u>	<u>YID</u>	Y1	A1	A2	<u>XIDR</u>	<u>YIDR2</u>
------------	------------	----	----	----	-------------	--------------

جدول غیرنرمال A

توجه: کلید کاندید جدول A برابر (XID,YID) است.

توجه: همانطور که گفتیم کلید خارجی یک رابطه می‌تواند متناظر با کلید کاندید همان رابطه باشد. در واقع دیگرام ارجاعی دارای طوقه است به این معنی که مبداء و مقصد کلید خارجی با هم یکی است. همانطور که گفتیم در نگاشت رابطه یک به چند در یک موجودیت به مدل رابطه‌ای مستقل از اختیاری یا اجباری بودن موجودیت، موجودیت به یک جدول تبدیل می‌گردد، بنابراین در جدول A صفات (XIDR,YIDR2) به عنوان کلید خارجی به کلید کاندید (XID,YID) از همان جدول A ارجاع می‌کند. شاید این سوال در ذهن شما شکل بگیرد که در کلید خارجی، XIDR چرا XIDR2 نوشته نشده است، پاسخ این است که خوب بهتر بود همین XIDR2 نوشته می‌شد که احتمالاً خطای حرفچینی سوال بوده است ولی مهم هم نیست چون مهمه نام عوض شود که عوض شده است.

توجه: همانطور که گفتیم توصیه می‌گردد صفت مشتق (پویا) در جداول بانک اطلاعات مورد استفاده قرار نگیرد و مقدار آن از طریق صفت مرتبط با آن محاسبه گردد، بنابراین صفت مشتق A3 در مدل EER در جدول A به عنوان ستون نیامده است. در جدول A در گزینه چهارم هم صفت مشتق A3 نیامده است.

توجه: در بانک اطلاعاتی مبتنی بر مدل رابطه‌ای (جدولی) صفت چندمقداری نداریم، صفت A2 در مدل EER مطرح شده در صورت سوال بر اساس نوع نمایش و نشانه‌گذاری صفت چند مقداری است و همین عامل باعث شده است که بگوییم جدول A غیرنرمال است. که برای حل این مساله و حذف صفت چند مقداری A2 از جدول A، این جدول باید در سطح اول نرمال قرار بگیرد.

نرمال فرم اول

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم اول را به صورت زیر بیان کرد:

- دارای حداقل یک کلید کاندید باشد.
- همه خصیصه‌های آن غیرقابل تجزیه باشند (جدول باید فاقد خصیصه‌های مرکب باشد)
- همه خصیصه‌های آن تک مقداری باشند (جدول باید فاقد خصیصه‌های چند مقداری باشد)

واضح است که جدول مطرح شده یعنی جدول A در فرم اول نرمال قرار ندارد. و باید با حذف خصیصه‌های چند مقداری آن یعنی صفت A2 در نرمال فرم اول قرار بگیرد.

مدل طراحی غیرنرمال:

<u>XID</u>	<u>YID</u>	Y1	A1	A2	<u>XIDR</u>	<u>YIDR2</u>
------------	------------	----	----	----	-------------	--------------

جدول غیرنرمال A

توجه: یکی از روش‌های حذف خصیصه‌های چند مقداری از یک جدول غیرنرمال و تبدیل آن به جداول نرمال به اینصورت است که جدول غیرنرمال به دو جدول تجزیه شود و کلید کاندید جدول غیرنرمال به عنوان کلید خارجی در جدول دوم دیگری تعریف شود و صفت چند مقداری نیز در آن درج شود، به صورت زیر:

مدل طراحی نرمال:

XID	YID	A2	XID	YID	Y1	A1	XIDR	YIDR2	
جدول نرمال AA2			جدول نرمال A						

توجه: در حال حاضر جدول A در نرمال فرم اول قرار دارد و همچنین جدول AA2 در نرمال فرم اول قرار دارد.

توجه: کلید کاندید جدول A صفات (XID, YID) است.

توجه: کلید کاندید جدول AA2 صفات (XID, YID, A2) یعنی تمام کلید است.

توجه: تجزیه در نرمال‌سازی یعنی دوری و دوستی، دور هستن یعنی دو جدول شدن، اما دوست هم هستن یعنی توسط تعریف کلید خارجی الحاق پذیر هم هستن و توسط عمل الحاق همان جدول پایه ایجاد می‌گردد.

توجه: همانطور که واضح است، طراحی جدول A در گزینه چهارم به صورت $A(XID, YID, Y1, XIDR, YIDR, A1)$ در نظر گرفته شده است، همچنین طراحی جدول AA2 در گزینه چهارم به صورت $AA2(XID, YID, A2)$ در نظر گرفته شده است که مطابق آنچه بیان کردیم، طراحی درستی است. بنابراین گزینه‌های اول، دوم و سوم را به طور کامل کنار می‌گذاریم، پس پُر واضح است که گزینه چهارم پاسخ سوال است. به همین سادگی.

توجه: نقطه چین بودن صفت XID در گزینه چهارم، به این دلیل است که در نگاشت بالاتر یعنی چند به یک بین دو موجودیت X و Y کلید کاندید جدول X یعنی XID به عنوان کلید خارجی در جدول Y تعریف شده بود. در جداول کلید خارجی توسط نقطه چین یا خط چین نشان داده می‌شود.

۳۷- گزینه (۳) صحیح است.

صورت سوال به این شکل است:

چه تعداد از گزاره‌های داده شده درست است؟

گزاره اول: تعداد کلیدهای کاندید یک رابطه از تعداد سوپرکلیدهای آن رابطه همواره کمتر است.
گزاره اول نادرست است، زیرا در حالت کلی، اگر رابطه R، دارای n خصیصه باشد، آنگاه تعداد ابرکلیدهای آن حداقل یک و حداکثر $2^n - 1$ است. در جدول تمام کلید، یک جدول فقط و فقط یک ابرکلید دارد و فقط هم یک کلید کاندید دارد. رابطه تمام کلید مثلاً ممکن است سه ستون داشته باشد، در این حالت یک جدول فقط و فقط یک ابرکلید دارد و فقط و فقط هم یک کلید کاندید دارد. یعنی حداقل یک ابرکلید و یک کلید کاندید دارد و حداکثر هم یک ابرکلید و یک کلید کاندید دارد. در گزاره اول حالت

کلی مورد بررسی قرار گرفته است. دقت کنید که حداقل یک ابرکلید و حداکثر $2^n - 1$ ابرکلید، حالت خاص جدول تمام کلید هم پوشش می‌دهد چون بیان حداکثر $2^n - 1$ مقادیر کوچکتر و برابر خودش را پوشش می‌دهد. اگر در یک رابطه با n خصیصه، تک تک خصیصه‌ها به تنهایی کلید کاندید باشد، آنگاه رابطه دارای n کلید کاندید است. بنابراین هر زیرمجموعه غیر تهی از خصیصه‌های این رابطه یک ابرکلید است. که در این حالت تعداد ابرکلیدهای یک رابطه با n خصیصه برابر با $2^n - 1$ است که بیشترین مقدار ممکن در تعداد ابرکلیدهای یک رابطه با n خصیصه است. ابرکلید بدون صفت نداریم، بنابراین حالت $\binom{n}{0}$ در نظر گرفته نمی‌شود. همانطور که گفتیم در حالت کلی، یک رابطه دارای n خصیصه، شرایط مختلفی را در تعداد ابرکلید می‌تواند تجربه کند، که حداکثر تعداد ابرکلیدهایی که می‌تواند تجربه کند برابر $2^n - 1 = \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n}$ است. در واقع عبارت زیر برقرار است:

$$2^n - 1 = \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n}$$

توجه: در گزاره اول اگر کلمه مساوی اضافه شود آنگاه گزاره درستی می‌بود، یعنی بشود تعداد کلیدهای کاندید یک رابطه از تعداد سوپرکلیدهای آن رابطه همواره کمتر یا مساوی است.

توجه: ابرکلیدی که عضو زائد نداشته باشد، کلید کاندید (Candidate key) است، به عبارت دیگر ابرکلید کمینه را کلید کاندید می‌گویند. منظور از ابرکلید کمینه، ابرکلیدی نیست که کمترین تعداد صفت را داشته باشد، بلکه منظور ابرکلیدی است که صفت زائد نداشته باشد.

مثال:

S#	Sname	City	S#	P#	QTY	P#	Pname	Color
S1	Sn1	C1	S1	P1	10	P1	Pn1	Red
S2	Sn2	C2	S1	P2	20	P2	Pn2	Blue
S3	Sn3	C2	S2	P1	30	P3	Pn3	Blue

جدول S

جدول SP

جدول P

S# : ابرکلید است. کلید کاندید نیز هست. (در جدول S)

(S#, Sname) : ابرکلید است، زیرا خاصیت کلیدی دارد، اما کلید کاندید نیست، زیرا عضو زائد Sname را دارد. در واقع صفت S#، به تنهایی خاصیت کلیدی دارد، بنابراین صفت Sname، عضو زائد است. (در جدول S)

(S#, P#) : ابرکلید است. کلید کاندید نیز هست. (در جدول SP).

مثال:

شماره ملی: ابرکلید است. کلید کاندید نیز هست.

(شماره ملی و نام خانوادگی): ابرکلید است. زیرا خاصیت کلیدی دارد، اما کلید کاندید نیست، زیرا عضو زائد نام خانوادگی را دارد. در واقع صفت شماره ملی، به تنهایی خاصیت کلیدی دارد، بنابراین صفت نام خانوادگی، عضو زائد است.

توجه: یک جدول می‌تواند چندین کلید کاندید داشته باشد.

مثال:

شماره دانشجویی	شماره ملی	نام خانوادگی	نام
کلید کاندید (کلید اصلی)	کلید کاندید (کلید فرعی)		

توجه: در مدل رابطه‌ای، هر رابطه حتماً حداقل یک کلید کاندید دارد، زیرا در بدترین شرایط، همه صفات با هم کلید کاندید می‌شوند، که به این رابطه تمام کلید (All key) گفته می‌شود.

توجه: یک رابطه، تحت هیچ شرایطی نمی‌تواند به دلیل استفاده از خاصیت مجموعه‌ای بودن، سطر تکراری داشته باشد. بنابراین یک رابطه، حداقل یک کلید کاندید دارد.

مثال: یک جدول تمام کلید.

a	b	c
1	2	3
1	6	3
1	2	7
8	2	3

نتیجه اینکه مجموعه کلیدهای کاندید همواره زیر مجموعه ابرکلیدها است، بنابراین تعداد کلیدهای کاندید یک رابطه از تعداد ابرکلیدهای آن رابطه همواره کمتر یا مساوی است.

گزاره دوم: کلید خارجی یک رابطه، می‌تواند جزیی از کلید اصلی آن باشد.

گزاره دوم درست است، زیرا از نظر ساختاری کلید خارجی یک رابطه، می‌تواند جزیی از کلید اصلی آن باشد.

مثال:

S#	Sname	City	S#	P#	QTY	P#	Pname	Color
S ₁	S _{n1}	C ₁	S ₁	P ₁	10	P ₁	P _{n1}	Red
S ₂	S _{n2}	C ₂	S ₁	P ₂	20	P ₂	P _{n2}	Blue
S ₃	S _{n3}	C ₂	S ₂	P ₁	30	P ₃	P _{n3}	Blue

جدول S

جدول SP

جدول P

S#: ابرکلید است. کلید کاندید نیز هست. (در جدول S)

(S#, Sname): ابرکلید است، زیرا خاصیت کلیدی دارد، اما کلید کاندید نیست، زیرا عضو زائد Sname را دارد. در واقع صفت S#، به تنهایی خاصیت کلیدی دارد، بنابراین صفت Sname، عضو زائد است. (در جدول S)

(S#, P#): ابرکلید است. کلید کاندید و کلید اصلی نیز هست. (در جدول SP)

S#: در جدول SP کلید خارجی است و به ستون S# در جدول S ارجاع می‌کند.

P#: در جدول SP کلید خارجی است و به ستون P# در جدول P ارجاع می‌کند.

گزاره سوم: استفاده از Veiw می‌تواند استقلال داده‌ای را افزایش دهد.

گزاره سوم درست است، زیرا معماری ANSI برای پایگاه داده‌ها شامل سه لایه زیر است:

- ۱- لایه خارجی.
- ۲- لایه ادراکی شامل زیر لایه‌های مدل تحلیل (طراحی ادراکی یا ادراکی عام) و مدل طراحی (طراحی منطقی یا ادراکی خاص).
- ۳- لایه داخلی (فیزیکی).

یک محصول نرم‌افزاری به واسطه فرآیند تولید نرم‌افزار که شامل فعالیت‌های مدل تحلیل، مدل طراحی، پیاده‌سازی و تست می‌باشد، ایجاد می‌گردد. کاربران نهایی در لایه خارجی، مدل تحلیل و مدل طراحی در لایه ادراکی و فعالیت پیاده‌سازی در لایه داخلی قرار دارند.

زبان‌های پیاده‌سازی

یک محصول نرم‌افزاری از دو وجه عملکرد (برنامه کاربردی) و داده (بانک اطلاعات) تشکیل می‌شود. انواع زبان‌های برنامه‌سازی به صورت زیر است:

زبان پیاده‌سازی برنامه کاربردی (وجه عملکرد)

برنامه کاربردی نیز مانند بخش داده، حاصل مراحل تحلیل، طراحی و پیاده‌سازی می‌باشد. مرحله پیاده‌سازی برنامه کاربردی توسط یکی از زبان‌های برنامه‌نویسی سطح بالا انجام می‌شود.
توجه: به زبان‌های سطح بالا، زبان میزبان یا زبان روالی (Procedural) نیز گفته می‌شود.

زبان پیاده‌سازی بانک اطلاعات (وجه داده)

در بانک اطلاعات از زبان‌های بیانی (Declarative) که به آنها زبان پرس‌وجو (Query Language) نیز گفته می‌شود، استفاده می‌شود. در زبان‌های بیانی کاربر برنامه‌ساز کفایت بگوید چه چیزی لازم دارد تا سیستم برای او ایجاد (مثل جدول) یا استخراج (مثل پرس و جوها) کند. در واقع چگونگی ایجاد جدول یا استخراج پرس‌وجوها از دید کاربر برنامه‌ساز و کاربر نهایی مخفی است.

استقلال داده‌ای

یکی از مهم‌ترین مزایای تکنولوژی پایگاه داده‌ها (مدل مفهومی پایگاه داده)، بلکه مهم‌ترین هدف آن تأمین و افزایش استقلال داده‌ای است، به معنی وابسته نبودن برنامه‌های کاربردی به داده‌های ذخیره شده. استقلال داده‌ای بر دو نوع می‌باشد:

۱- استقلال فیزیکی داده‌ها

به معنی مصونیت برنامه‌های کاربردی در قبال تغییراتی که در سطح فیزیکی (رسانه ذخیره‌سازی) پایگاه داده‌ها بروز می‌کند. یعنی اگر تغییری در ذخیره‌سازی داده‌ها انجام گیرد (برای مثال نوع دیسک عوض شود) برنامه‌های کاربردی هیچ تغییری نکنند.

۲- استقلال منطقی داده‌ها

به معنی مصونیت برنامه‌های کاربردی در قبال تعاریف و تغییراتی که در سطح مدل طراحی (مدل رابطه‌ای) پایگاه داده بروز می‌کند. یعنی تعریف و تغییر مدل طراحی بانک (ادراکی خاص یا طراحی منطقی) از دید برنامه‌های کاربردی آنها مخفی بماند.

برای مثال مدل رابطه‌ای از تجربیدی به نام جدول استفاده می‌کند و داده‌ها هر چه باشند در قالب چند جدول ریخته می‌شوند و نحوه ذخیره‌سازی داده‌ها روی رسانه‌ها از دید برنامه کاربردی مخفی است. در حالی که

در روش فایلینگ تعاریف مربوط به فایل‌های داده‌ای، در فایل برنامه کاربردی می‌آید. از آنجاکه برنامه‌های کاربردی براساس مدل طراحی بانک (ادراکی خاص یا طراحی منطقی) تعریف می‌شوند، بنابراین به طور بالقوه در معرض تأثیرپذیری از تغییرات در مدل طراحی بانک (ادراکی خاص یا طراحی منطقی) قرار دارند. **توجه:** در سیستم‌های امروزی، این نوع استقلال هم تا حدی (و نه صددرصد) تأمین شده است.

انواع تغییر در مدل طراحی (طراحی منطقی یا ادراکی خاص)

۱- رشد پایگاه داده‌ها به دلیل مطرح شدن نیازهای جدید مشتری: مانند درج جدول جدید، ترکیب جداول، تجزیه جداول.

۲- سازماندهی مجدد: مانند تغییر در نوع صفات خاصه، تغییر در اندازه صفات.

مثال: اگر جدولی دارای چهار ستون باشد و ستون پنجمی نیز به آن اضافه گردد، در صورتی که برنامه کاربردی سابق نیاز به دستکاری و تغییر نداشته باشد، استقلال منطقی داده‌ها براساس تغییرات نیز لحاظ شده است.

توجه: از آن جا که با حذف جداول، داده‌ها هم از بین می‌رود، بنابراین برنامه‌های کاربردی نسبت به حذف جداول هیچگاه استقلال منطقی نخواهند داشت.

همانطور که گفتیم یک محصول نرم‌افزاری از دو وجه **عملکرد** (برنامه کاربردی) و **داده** (بانک اطلاعات) تشکیل شده است، بخش داده (بانک اطلاعات) که با SQL پیاده‌سازی می‌شود به مفاهیم استقلال داده‌ای مرتبط است. ساختار وجه داده توسط دستورات DDL نظیر Create Table، Create View و Create Index و دیگر دستورات آن ایجاد و مدیریت می‌گردد. و مقادیر وجه داده توسط دستورات DML نظیر Insert، Update و Delete و دیگر دستورات آن ایجاد و مدیریت می‌گردد.

دستور Create Table با ساخت مفهوم جدول، کمک به برقراری استقلال داده‌ای از نوع **استقلال فیزیکی داده‌ها** میان یک برنامه کاربردی و داده‌ها می‌کند، به معنی وابسته نبودن برنامه‌های کاربردی به داده‌های ذخیره شده، یعنی همانطور که گفتیم، مدل رابطه‌ای از تجزیدی به نام جدول استفاده می‌کند و داده‌ها هر چه باشند در قالب چند جدول ریخته می‌شوند و نحوه ذخیره‌سازی داده‌ها روی رسانه‌ها از دید برنامه کاربردی مخفی است. دقت کنید که Table بخشی از وجه داده است. در واقع بخش داده از بخش‌های مختلف Table، View و Index تشکیل شده است.

دستور Create View با ساخت مفهوم دید، تا حدی کمک به برقراری استقلال داده‌ای از نوع **استقلال منطقی داده‌ها** میان یک برنامه کاربردی و داده‌ها می‌کند، به معنی وابسته نبودن برنامه‌های کاربردی به داده‌های ذخیره شده، یعنی همانطور که گفتیم، اگر جدولی دارای چهار ستون باشد و ستون پنجمی نیز به آن اضافه گردد، در صورتی که برنامه کاربردی سابق نیاز به دستکاری و تغییر نداشته باشد، استقلال منطقی داده‌ها براساس تغییرات نیز لحاظ شده است. از آنجاکه View روی ساختار قدیم شامل نام جدول قدیم و ستون‌های قدیم ایجاد می‌شود، اگر جدولی دارای چهار ستون باشد و ستون پنجمی نیز به آن اضافه گردد، آنگاه بدون تغییرات در ساختار View بخش داده و به تبع تغییرات در ساختار بخش عملکرد (برنامه کاربردی)، امکان حیات برنامه کاربردی بدون اشکال همچنان وجود دارد و این یعنی View حافظ استقلال داده‌ای از نوع استقلال منطقی داده‌ها است. دقت کنید که View بخشی از وجه داده است. در واقع بخش داده از بخش‌های مختلف Table، View و Index تشکیل شده است.

مثال: پیاده‌سازی جدول S به صورت زیر را در نظر بگیرید:

```
CREATE TABLE S
(
  S# char (5),
  Sname char (20)
  Primary key (S#)
)
```

جدول S با مقادیر زیر را در نظر بگیرید:

S#	Sname
S1	Sn1
S2	Sn2

جدول S

پرس و جوی زیر را در نظر بگیرید:

```
SELECT *
FROM S
```

خروجی پرس و جو به صورت زیر است:

S#	Sname
S1	Sn1
S2	Sn2

پیاده‌سازی View با نام SV را روی جدول S به صورت زیر در نظر بگیرید:

```
CREATE VIEW SV
AS SELECT S.S# AS SID, S.Sname AS SN
FROM S
```

پرس و جوی زیر بر روی VIEW با نام VS را به صورت زیر در نظر بگیرید:

```
SELECT *
FROM SV
```

خروجی پرس و جو بر روی VIEW با نام VS به صورت زیر است:

SID	SN
S1	Sn1
S2	Sn2

تغییرات: اضافه‌شدن ستون City به جدول S را به صورت زیر در نظر بگیرید:

```
ALTER TABLE S
ADD City char (15)
```

ساختار و مقادیر جدول S پس از اضافه‌شدن ستون City به صورت زیر است:

S#	Sname	City
S1	Sn1	NULL
S2	Sn2	NULL

مجددا پرس و جوی زیر بر روی VIEW با نام VS را به صورت زیر در نظر بگیرید:

```
SELECT *
FROM VS
```

خروجی پرس و جو بر روی VIEW با نام VS به صورت زیر است:

SID	SN
S1	Sn1
S2	Sn2

توجه: همانطور که واضح است، از آنجاکه View روی ساختار قدیم شامل نام جدول قدیم و ستون‌های قدیم ایجاد می‌شود، اگر جدولی دارای دو ستون باشد و ستون سومی نیز به آن اضافه گردد، آنگاه بدون تغییرات در ساختار View بخش داده و به تبع تغییرات در ساختار بخش عملکرد (برنامه کاربردی)، امکان حیات برنامه کاربردی بدون اشکال همچنان وجود دارد و این یعنی View حافظ استقلال داده‌ای از نوع استقلال منطقی داده‌ها است.

همانطور که گفتیم استفاده از View می‌تواند استقلال داده‌ای را افزایش دهد.

گزاره چهارم: مدیریت تراکنش‌های همروند به عهده DBMS است.

گزاره چهارم درست است، زیرا هر برنامه‌ای که در محیط بانک اطلاعاتی توسط کاربر اجرا گردد یک تراکنش نام دارد. (مانند عملیات کارت به کارت در یک دستگاه خودپرداز بانک) تراکنش واحد کار DBMS است. به طور کلی هر عملیاتی در پایگاه داده در قالب یک تراکنش تعریف و اجرا می‌شود. هر تراکنش شامل دو یا چند دستور SQL است. تفاوت اصلی یک تراکنش با یک برنامه معمولی در محیط غیربانکی این است که تراکنش همواره به DBMS تسلیم می‌شود و DBMS در اعمال هر گونه کنترل و حتی به تعویق انداختن و ساقط کردن آن آزادی عمل دارد. هدف اصلی از اینگونه کنترل‌ها، حذف‌ها و تعویق‌ها، حفظ جامعیت داخلی و خارجی بانک اطلاعات است.

تضمین جامعیت داخلی و خارجی بانک اطلاعات

چهار کنترل زیر لازم است روی تمامی تراکنش‌ها در بانک اطلاعات اعمال گردد تا جامعیت داخلی و خارجی یعنی رعایت اصل سازگاری آن تضمین شود. این کنترل‌ها به خواص ACID معروفند:

۱- یکپارچگی یا تجزیه‌ناپذیری (Atomicity)

این خاصیت به همه یا هیچ موسوم است. منظور این است که یا تمام دستورات یک تراکنش باید اجرا شود یا هیچکدام از آنها نباید اجرا شود. این به معنی تجزیه‌ناپذیر بودن بخش‌های مختلف یک تراکنش است. برای مثال تراکنش انتقال پول از حساب A به حساب B از دو بخش جداگانه تشکیل یافته است:

الف) بخش اول (برداشت پول)

پول را از حساب A برداشت می‌کند.

ب) بخش دوم (واریز پول)

همان پول را به حساب B واریز می‌کند.

بخش اول حساب A را بدهکار و بخش دوم حساب B را بستانکار می‌کند. در شروع و پایان یک تراکنش سیستم باید سازگار باشد ولی در اثنای اجرای تراکنش ممکن است موقتاً نیاز به ناسازگاری باشد. برای مثال هنگام واریز پول از حساب A به B، پس از برداشت پول از حساب A سیستم به طور موقت ناسازگار است و پس از واریز آن به حساب B دوباره سیستم سازگار می‌شود. لذا برنامه برداشت از حساب A یا واریز به حساب B به تنهایی تراکنش نیستند.

این دو بخش ممکن است روی دو کامپیوتر جداگانه اجرا شوند. فرض کنید بخش اول تراکنش اجرا شود اما ناگهان ارتباط با ماشین دوم قطع گردد. و بخش دوم قابل انجام نباشد. بدیهی است که در این حالت باید پول برداشت شده دوباره به همان حساب اول بازگردانده شود تا جامعیت بانک اطلاعات حفظ شود. این عمل معادل این است که بگوییم هیچ دستورالعملی از تراکنش انجام نشده است.

به عنوان مثالی دیگر، هنگام خرید اینترنتی با کارت عضو شتاب ممکن است پول از حساب شما کسر گردد، اما به حساب فروشگاه مورد نظر واریز نگردد، بنابراین خرید شما ناموفق اعلام می گردد، که در این حالت پول حداکثر تا ۴۸ ساعت دیگر به حساب شما بازمی گردد.

در بیانی دیگر تراکنش را می توان اینگونه تعریف کرد، تراکنش مجموعه ای از دستورات تعریف و دستکاری داده هاست که DBMS تضمین می کند یا همه آن دستورات اجرا شوند و یا هیچکدام از آن دستورات اجرا نشوند. برای محقق کردن خاصیت یکپارچگی (Atomicity) هر تراکنش می بایست بین دو دستور Begin Transaction و End Transaction قرار گیرد. به طور کلی هر عملیاتی در پایگاه داده در قالب یک تراکنش تعریف و اجرا می شود. پس تراکنش واحد کار DBMS است. هر تراکنش شامل دو یا چند دستور SQL است. بر این اساس می توان گفت که هیچ پرس و جویی برای پایگاه داده هویت مستقل ندارد، بلکه DBMS فقط تراکنش ها را می شناسد و اجرا می کند. به این ترتیب باید گفت که هر پرس و جویی در پایگاه داده ابتدا به یک تراکنش تبدیل می شود و سپس اجرا می گردد. در SQL برای نمایش ابتدای یک تراکنش از دستور Begin Transaction و برای نمایش خاتمه یک تراکنش از دستور End Transaction استفاده می شود. تراکنش با اجرای Begin Transaction شروع می گردد و در صورت اجرای موفق commit و در صورت عدم موفقیت یعنی عدم اجرای همه بخش های مختلف تراکنش با اجرای دستور Rollback خاتمه می یابد. با اجرای این دستور کلیه تغییراتی که تراکنش روی پایگاه داده اعمال نموده است، ابطال می شود و وضعیت پایگاه داده به آخرین وضعیت قبل از اجرای تراکنش برگردانده می شود.

۲- سازگاری (Consistency)

به طور کلی جامعیت در سیستم های بانکی به دو طبقه جامعیت داخلی و خارجی تقسیم می گردد. به حفظ قوانین مطرح شده از سوی مدل رابطه ای و DBMS در سطح پیاده سازی، جامعیت داخلی و به حفظ قوانین مطرح شده از سوی طراحان و برنامه نویسان بانک در سطح پیاده سازی، جامعیت خارجی گفته می شود. در صورتی که در یک بانک جامعیت داخلی و خارجی هر دو توأم باهم برقرار باشد، در آن بانک، اصل سازگاری برقرار شده است. به عبارت دیگر سازگاری، به معنی رعایت قوانین داخلی و خارجی در بانک است. DBMS مسئول کنترل قوانین داخلی و خارجی در بانک است و هرگونه عاملی که باعث نقض قوانین داخلی و خارجی و به تبع سازگاری بانک گردد را رد می کند. قوانین داخلی بانک شامل قانون جامعیت درون رابطه ای، قانون جامعیت موجودیت، قانون جامعیت ارجاعی و قانون جامعیت دامنه ای است، این موارد در فصل مدل رابطه ای بررسی خواهد شد. قوانین خارجی بانک هم شامل هر قانونی است که طراحان و برنامه نویسان بانک آنرا وضع می کنند، مانند تعریف زیردامنه برای ورود اطلاعات، تعریف بازه 0 تا 20 برای نمرات، تعریف بازه 0 تا بینهایت برای حساب های بانکی به معنی عدم وجود موجودی منفی در حساب های بانکی...

در سیستم های بانکی کنترل جامعیت داخلی و خارجی به صورت خودکار توسط مکانیزم های موجود در DBMS انجام می گردد.

حال یکبار دیگر کد تعریف جدول SP را در نظر بگیرید:

```
Create Table SP
(
  S# char (5),
  P# char (5),
  QTY numeric (10),
  Primary key (S#, P#),
  Foreign key (S#) References S(S#)
    on delete cascade
    on update cascade,
  Foreign key (P#) References P(P#)
    on delete cascade
    on update cascade,
  Check (QTY>1 AND QTY<1000)
)
```

کارکرد قطعه کد زیر از کد تعریف جدول SP فوق به صورت زیر است:

```
Foreing key (S#) References S(S#)
on delete cascade
on update cascade
```

این قطعه کد، سبب می‌گردد تا به طور خودکار هرگونه تغییری در ستون S# در جدول S به ستون S# در جدول SP نیز اعمال گردد. بنابراین **جامعیت داخلی** از نوع جامعیت ارجاعی نقض نمی‌گردد. یا به طور مشابه، کارکرد قطعه کد زیر از کد تعریف جدول SP فوق به صورت زیر است:

```
Foreing key (P#) References P(P#)
on delete cascade
on update cascade
```

این قطعه کد، سبب می‌گردد تا به طور خودکار هرگونه تغییری در ستون P# در جدول P به ستون P# در جدول SP نیز اعمال گردد. بنابراین **جامعیت داخلی** از نوع جامعیت ارجاعی نقض نمی‌گردد. کارکرد قطعه کد زیر از کد تعریف جدول SP فوق به صورت زیر است:

```
Check (QTY>1 AND QTY<1000)
```

این قطعه کد، سبب می‌گردد تا به طور خودکار هرگونه مقداردهی در ستون QTY از جدول SP در بازه 1 تا 1000 باشد، بنابراین **جامعیت خارجی** نقض نمی‌گردد.

توجه: همانطور که واضح است DBMS در حفظ جامعیت داخلی و خارجی به دقت نظارت دارد. خاصیت سازگاری (Consistency) بیانگر این است که اگر یک تراکنش در محیط بانک اطلاعات انجام شود باید بانک اطلاعات را از حالتی سازگار به حالت سازگار دیگری منتقل کند. به بیان دیگر هر تراکنش باید تمامی قوانین جامعیت داخلی و خارجی بانک اطلاعات را رعایت کند. خاصیت سازگاری می‌گوید انجام تراکنشی از سوی DBMS باید پذیرفته شود که جامعیت داخلی و خارجی به معنی حفظ قوانین داخلی و خارجی پایگاه داده را رعایت کند، یعنی پس از انجام تراکنش‌ها اصل سازگاری برقرار باشد، در غیراینصورت انجام تراکنش رد شود.

بنابراین تا به اینجا مشاهده شد که تراکنش ممکن است دو نوع پایان داشته باشد:
الف) پایان موفق که آن را انجام (commit) می‌نامند.
ب) پایان ناموفق که آن را سقوط (abort) می‌نامند.

۳- انزوا یا جداسازی (Isolation)

همزمانی در دسترسی به داده‌ها موجب بهبود کارایی و کاهش زمان پاسخ‌گویی سیستم می‌گردد. و این امر تسریع عملکرد برنامه‌ها را در پی دارد. بسیاری از سیستم‌ها اجازه می‌دهند که چندین کاربر به صورت هم‌روند به داده‌ها دسترسی یابند و تغییرات مورد نظر خود را بر روی آنها اعمال نمایند. در چنین محیط‌هایی تغییرات هم‌روند ایجاد شده بر روی داده ممکن است منجر به ایجاد ناسازگاری در داده‌ها گردد. در سیستم‌های بانکی کاربران مختلف می‌توانند به صورت هم‌زمان با بانک کار کنند. بنابراین اگر یک داده خاص بین کاربران مختلف به صورت اشتراکی مورد بازیابی و دستکاری قرار گرفت سیستم پایگاه داده باید محیطی را ایجاد نماید که مانع از بروز مشکلات و یا ایجاد نتایج نامطلوب گردد، مانند مطالب مربوط به ناحیه بحرانی در فرآیندهای هم‌روند در درس سیستم عامل.

مثال: فرض کنید کاربر A و B به ترتیب در تهران و شیراز هم‌زمان قصد برداشت وجه از حساب آقای 6037 از طریق برگ چک را دارند. بنابراین روال‌های زیر را خواهیم داشت، از آنجا که برداشت وجه از یک رکورد مشترک (عامل مشترک) صورت می‌گیرد، به تبع وقوع پدیده هم‌زمانی برای هر دو تراکنش رخ می‌دهد. روال کار بدین صورت است که هر دو تراکنش مبلغ موجودی حساب که برابر مقدار 500 هزار تومان می‌باشد را خوانده و با توجه به مبلغ برگ چک A و B به ترتیب مبلغ 100 و 50 هزار تومان را از حساب کسر می‌کنند و بر حسب اینکه کدام تراکنش آخرین بروزرسانی را انجام دهد مبلغ مانده حساب 400 تا 450 هزار تومان ذخیره می‌گردد. که در هر دو صورت اطلاعات نادرستی ذخیره شده است. در حالی که مبلغ 350 هزار تومان باید جهت مبلغ مانده حساب ذخیره می‌شد!

موجودی	نام خانوادگی	نام	شماره حساب	شماره مشتری
500	Alavi	Ali	0313	6037

Read (A)

A \square 500

R=500-100

R \square 400

Write (R) \square 400

Read (B)

B \square 500

R=500-50

R \square 450

Write (R) \square 450

بدین ترتیب مقادیر نادرستی توسط برنامه‌ها ذخیره شده است که این نادرستی به دلیل تداخل عملیات دو برنامه هم‌روند است. برای جلوگیری از ایجاد چنین نتایج نامطلوبی سیستم باید نوعی نظارت بر عملکرد برنامه‌های هم‌روند داشته باشد. مانند روش قفل‌گذاری.

در بانک اطلاعات ممکن است تراکنش‌های هم‌روند وجود داشته باشد (مثل سیستم‌های چند برنامه‌ای) بر طبق خاصیت انزوا هم‌روندی تراکنش‌ها باید کنترل شود تا اثر مخرب بر روی هم نداشته باشند به بیان دیگر اثر تراکنش‌های هم‌روند روی یکدیگر چنان است که گویا هر کدام در انزوا انجام می‌شود.

به تعریفی دیگر تراکنش‌ها جدا از یکدیگر هستند. اگر چند تراکنش به طور هم‌زمان اجرا شوند، به

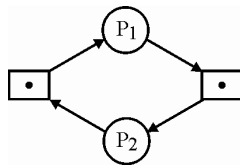
هنگام سازی های هر کدام از یکدیگر مخفی می مانند تا به اتمام برسند. به عبارتی دیگر، برای دو تراکنش مجزای A و B، تراکنش A می تواند بهنگام سازی های B را پس از پذیرفته شدن آن (commit) یا B می تواند بهنگام سازی های A را پس از پذیرفته شدن A ببیند. اما این دو تراکنش به طور همزمان نمی توانند، بهنگام سازی های یکدیگر ببینند.

توجه: کنترل همروندی توسط بخشی از DBMS بنام واحد کنترل همروندی (concurrency control) انجام می شود.

روش قفل گذاری

یکی از روش های اعمال خاصیت جداسازی در تراکنش ها و جلوگیری از اثر مخرب تراکنش های همروند بر روی یکدیگر، روش قفل گذاری است. در این روش هنگامی که تراکنش به داده ای نیاز داشته باشد. تقاضای قفل کردن آن را می دهد و در این حالت بقیه تراکنش ها تا اتمام کار آن نمی توانند از آن استفاده کنند. در واقع استفاده از انواع قفل ها در مکانیسم قفل گذاری، روش های دسترسی به فیلدها و یا داده های اشتراکی را در کاربردهای دیگر، امکان پذیر می سازد. وجود قفل ها سبب می شود که در اجرای همروند چند تراکنش، یک رکورد مشترک (عامل مشترک) به صورت همزمان توسط دو تراکنش مورد استفاده قرار نگیرد. این کار سرعت عملیات را افزایش می دهد زیرا داده اشتراکی به صورت انحصاری فقط در اختیار یک تراکنش است. ولی احتمال بروز بن بست را به دلیل برقراری شرط انحصار متقابل زیاد می کند.

بر اساس قاعده کافمن شرایط وقوع بن بست به صورت زیر است:



۱- انحصار متقابل

۲- انحصاری بودن

۳- نگهداری و انتظار

۴- انتظار چرخشی

۴- پایایی یا ماندگاری (Durability)

بر اساس این خاصیت تراکنش هایی که به مرحله انجام (commit) برسند اثرشان ماندنی است و هرگز به طور تصادفی از بین نمی روند. برای مثال اگر مبلغی به حسابی واریز شود و تراکنش مربوطه انجام یافته (commit) اعلام شود حتی در صورت وقوع حادثه در آن شعبه بانک، مشتری نباید متضرر شود، برای مثال عمل واریز قبل از اعلام انجام موفق (commit) باید در جای دیگر نیز ثبت شده باشد، مثل دیسک اصلی یا دیسک پشتیبان. یعنی تأثیرات تراکنش در پایگاه داده، ماندگار باشد. هنگامی که یک تراکنش دستور commit را اجرا می کند نتایج اجرای آن به نسخه ای اصلی پایگاه داده در دیسک منتقل می شود. بنابراین می توان گفت تا زمانی که یک تراکنش دستور commit را اجرا نکرده است یا به اصطلاح تثبیت نشده است، ویژگی ماندگاری در مورد آن تراکنش تضمین شده نیست. اما پس از اجرای دستور ماندگاری نتایج تراکنش تضمین می شود.

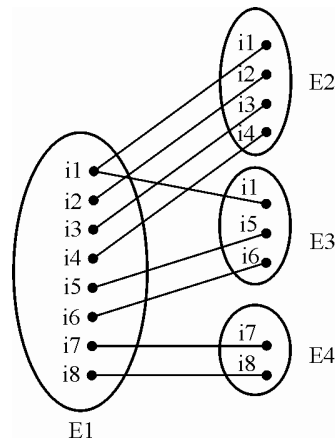
توجه: دو خاصیت یکپارچگی و پایایی توسط واحدی از DBMS به نام واحد مدیریت بازگرد (Recovery management) کنترل می گردد.

توجه: سازمان سنجش آموزش کشور، در کلید اولیه خود، گزینه چهارم را به عنوان پاسخ اعلام نمود، اما در کلید نهایی گزینه سوم را به عنوان پاسخ نهایی اعلام نمود، که کار درستی بوده است.

۳۸- گزینه (۴) صحیح است.

صورت سوال به این شکل است:

نمونه‌هایی از چهار موجودیت E1، E2، E3 و E4 در شکل زیر نمایش داده شده است، کدام گزینه بهترین نمودار EER معرف محیط است؟



در رابطه ISA رابطه پدر با فرزندان به دو صورت رابطه اختیاری یا جزئی یا بخشی (Partial) با نماد خط عمودی و رابطه اجباری یا کلی یا کامل (Total) با نماد خط مضاعف عمودی است. یک رابطه اجباری است، اگر و تنها اگر تمام نمونه موجودیت‌های پدر در رابطه شرکت کرده باشند. اجباری بودن رابطه ISA در نمودار EER با نماد خط مضاعف عمودی از موجودیت پدر به سمت موجودیت‌های فرزند نشان داده می‌شود.

همانطور که در شکل صورت سوال واضح و مشخص است، همه نمونه موجودیت‌های i1 تا i8 در موجودیت E1 یعنی پدر با نمونه موجودیت‌های فرزند یعنی موجودیت‌های E2، E3 و E4 وارد رابطه شده‌اند. و از آنجاکه اگر و تنها اگر تمام نمونه موجودیت‌های پدر در رابطه شرکت کرده باشند، آن رابطه اجباری است، بنابراین نمونه موجودیت‌های E1 یعنی پدر به صورت اجباری وارد رابطه با فرزندان یعنی موجودیت‌های E2، E3 و E4 شده‌اند. بنابراین گزینه‌های اول و سوم پاسخ سوال نیستند.

توجه: از آنجا که مطابق شکل صورت سوال ارتباط پدر با فرزندان هم یک به یک و هم یک به چند (نمونه موجودیت i1 در E1 یعنی پدر با نمونه موجودیت i1 در E2 یعنی فرزند و نمونه موجودیت i1 در E3 یعنی فرزند به طور مشترک رابطه دارد) است. و همچنین اشتراک نمونه موجودیت‌ها میان موجودیت‌های فرزند با یک نمونه موجودیت از پدر هم تهی و هم غیرتهی (نمونه موجودیت i1 در E2 یعنی فرزند و نمونه موجودیت i1 در E3 یعنی فرزند با نمونه موجودیت i1 در E1 یعنی پدر به طور مشترک رابطه دارند) است، بنابراین رابطه فرزندان با پدر حالت متصل یا پوشا یا تخصیص غیرمجزا (Overlap) است، که پُر واضح است که گزینه چهارم پاسخ سوال است.

توجه: در یک رابطه پوشا یا تخصیص غیرمجزا (Overlap)، نمونه موجودیت‌های فرزند می‌توانند به طور همزمان با نمونه‌ای از موجودیت پدر در ارتباط باشند که این یعنی همان رابطه یک به چند میان پدر و فرزندان.

توجه: سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه چهارم را به عنوان پاسخ اعلام کرده بود.

تست‌های فصل چهارم: جبر رابطه‌ای

۱- اگر ستون FK کلید خارجی در جدول a باشد، که به جدول b ارجاع می‌دهد، کدام گزینه معادل

$$\prod_{a.*} \left(\begin{array}{l} \sigma(a \times b) \\ \theta \wedge a.FK = b.FK \end{array} \right) \text{ است؟} \quad \text{(کارتداسی ارشد- دولتی ۸۳)}$$

$$\sigma_{\theta}(a \times b) \quad (۱) \quad \sigma_{\theta}(a \bowtie b) \quad (۲) \quad \prod_{a.*} (a \bowtie b) \quad (۳) \quad \prod_{b.*} (\sigma_{\theta}(a \times b)) \quad (۴)$$

۲- رابطه حرارت (کد ناحیه، نام، تاریخ، حرارت بالا، حرارت پایین) برای ثبت درجه حرارت‌های بیشینه و کمینه در نواحی مختلف در زمان‌های متفاوت به کار می‌رود. کلید این رابطه < کد ناحیه، تاریخ > می‌باشد. اگر روابط جبر رابطه‌ای زیر را در نظر بگیرید:

$$\begin{aligned} R_1 & \text{ (کد ناحیه، تاریخ، H) = (حرارت) حرارت بالا، تاریخ، کد ناحیه } \Pi \\ R_2 & \text{ (کد ناحیه، تاریخ، L) = (حرارت) حرارت پایین، تاریخ، کد ناحیه } \Pi \\ R_3 & \text{ (کد ناحیه) = (حرارت) } R_1 \bowtie \text{ کد ناحیه } \Pi \\ & \text{ حرارت بالا } H < \\ R_4 & \text{ (کد ناحیه) = (حرارت) } R_2 \bowtie \text{ کد ناحیه } \Pi \\ & \text{ حرارت پایین } L > \end{aligned}$$

آن‌گاه کدام گزینه نام نواحی با درجه حرارت بیشینه و کمینه را به ما می‌دهد؟ (مهندسی IT- دولتی ۸۴)

$$\begin{aligned} (۱) & \Pi \text{ نام } (R_3 \bowtie R_4) \\ (۲) & \Pi \text{ نام } (R_3 \cup R_4) \\ (۳) & ((R_4 - R_3) \cup \Pi \text{ کد ناحیه } (R_3 - R_4)) \bowtie \Pi \text{ نام } (R_3) \\ (۴) & ((R_4 \cap R_3) \cap \Pi \text{ کد ناحیه } (R_3 - R_4)) \bowtie \Pi \text{ نام } (R_3) \end{aligned}$$

۳- دو رابطه زیر را در نظر بگیرید: (مهندسی کامپیوتر- آزاد ۷۲)

STUDENT	STU-ID	NAME
	S001	AHMADI
	S002	TAVANA
	S003	DIBADJ

COURSE	COURSE-ID	STU-ID	TITLE
	CSC100	S001	PASCAL PROG.
	CSC100	S002	PASCAL PROG.
	CSC200	S001	COMPILER WRITING
	CSC300	S004	DATA STRUCTURE

رابطه‌ای که حاصل الحاق طبیعی دو رابطه بالا باشد دارای چند ستون و چند تاپل خواهد بود؟
 (۱) 4 و 3 (۲) 5 و 3 (۳) 4 و 6 (۴) 4 و 5

۴- رابطه‌ای که حاصل OUTER JOIN دو رابطه سؤال قبل باشد دارای چند تاپل خواهد بود؟
(مهندسی کامپیوتر- آزاد ۷۲)

- (۱) 4 (۲) 5 (۳) 3 (۴) هیچ کدام

۵- کدام یک از گزینه‌های زیر غلط است؟
(مهندسی کامپیوتر- دولتی ۷۵)

(توجه: \cup : Union و \cap : Intersection, \bowtie : semijoin, \Join : join)

$$R \cap S = R - (R - S) \quad (۲) \quad R \bowtie_f S = \sigma_f(R \cup S) \quad (۱)$$

$$R \bowtie_f S = S \bowtie_f R \quad (۴) \quad R \times_f S \neq S \times_f R \quad (۳)$$

۶- فرض کنید R_1 و R_2 دو رابطه باشند و $R_3 = R_1 \cup R_2$ ، کدام یک از گزاره‌های زیر صحیح است؟
(مهندسی کامپیوتر- دولتی ۸۲)

- (۱) کلید اصلی R_3 اجتماع کلیدهای اصلی R_1 و R_2 است.
(۲) کلید اصلی R_3 تمام خصیصه‌های R_1 و R_2 است.
(۳) کلید اصلی R_3 کلید اصلی R_1 یا کلید اصلی R_2 است.
(۴) کلید اصلی R_3 تقاطع تمام خصیصه‌های R_1 و R_2 است.

۷- اگر A یک رابطه دارای n خصیصه (ستون) باشد، تعداد تصاویر A که تهی نیستند برابر است با:

- (مهندسی کامپیوتر- دولتی ۸۲)
- (۱) $2n-1$ (۲) $n!-1$ (۳) n^2-1 (۴) 2^n-1

۸- عبارت زیر در چه صورت صحیح است؟
(مهندسی کامپیوتر- دولتی ۸۲)

$$\sigma_p(\prod_{a_1, a_2, \dots, a_n}(R)) = \prod_{a_1, a_2, \dots, a_n}(\sigma_p(R))$$

σ : انتخاب
 Π : تصویر

- (۱) شرط p حداقل برای یک سطر رابطه R برقرار باشد.
(۲) شرط p فقط ستون‌های a_1, \dots, a_n را در بر گیرد.
(۳) شرط p همه ستون‌های a_1, \dots, a_n را در بر گیرد.
(۴) هیچ کدام

۹- اگر مجموعه عنوان (Heading) در رابطه R و S با هم یکسان باشند، آنگاه پیوند طبیعی این دو رابطه معادل با کدام عمل، روی رابطه‌های R و S خواهد بود؟
(مهندسی IT- دولتی ۸۳)

- (۱) ضرب (۲) اجتماع (۳) اشتراک (۴) تفاضل

برای پاسخ به دو پرسش بعدی، جداول زیر را در نظر بگیرید:

هتل (کد هتل، نام هتل، آدرس، شماره تلفن)

رزرو اتاق (کد رزرو، کد هتل، نام، آدرس، شماره تلفن، تاریخ تولد، تاریخ ورود، مدت اقامت، تاریخ رزرو)

اتاق (کد هتل، شماره اتاق، امکانات)

وضعیت اتاق (کد هتل، شماره اتاق، کد رزرو، تاریخ ورود، تاریخ خروج)

جدول هتل دارای 35 سطر، رزرو اتاق 25000، اتاق 1500 و وضعیت اتاق R سطر دارد.

اطلاعات مشتریان در جدول رزرو اتاق ذخیره می‌شود.

۱۰- اطلاعات قبلی و قدیمی تا مدتی در پایگاه داده ذخیره می‌شود. اگر کاردینالیته را معادل تعداد تاپل‌های حاصله تعریف کنیم آن‌گاه:

(مهندسی IT- دولتی ۸۴)

کاردینالیته اتاق \bowtie هتل (اتاق. کد هتل \neq هتل. کد هتل) برابر کدام مقدار است؟

(۱) $1500 \times R$ (۲) 25000 (۳) 26500 (۴) 51000

۱۱- با توجه به مساله قبلی کاردینالیته وضعیت اتاق \bowtie هتل (وضعیت اتاق. کد هتل = هتل. کد هتل) برابر است با:

(مهندسی IT- دولتی ۸۴)

(۱) R (۲) $35 \times R$ (۳) 25000 (۴) $52500 \times R$

۱۲- در بانک اطلاعاتی تولید کنندگان و قطعات دستور جبر رابطه‌ای زیر کدام خروجی را تولید می‌کند؟

(مهندسی کامپیوتر- دولتی ۸۵)

$\Pi_{S\#}(\sigma_{CITY='PARIS'}(S \bowtie SP))$

(۱) شماره تمام تولیدکنندگان پاریس.

(۲) شماره اولین تولیدکننده پاریس.

(۳) شماره یکی از تولیدکنندگان پاریس.

(۴) شماره تولیدکنندگان پاریس که قطعه‌ای را تولید می‌کنند.

۱۳- اگر رابطه $R(a, b)$ دارای تعداد r تاپل بوده و رابطه $S(a, b)$ دارای تعداد s تاپل آنگاه تعداد کمینه و بیشینه تاپل‌های $R \cup S$ برابر است با:

(مهندسی IT- دولتی ۸۵)

(۱) $r + s$ و $\min(r, s)$ (۲) $r - s$ و $\max(r, s)$
(۳) $r + s$ و $\max(r, s)$ (۴) $\min(r, s)$ و $\max(r, s)$

۱۴- کدام یک از گزینه‌های زیر درست است؟

(مهندسی IT- دولتی ۸۵)

(۱) $\sigma_{c-d}(R) \equiv \sigma_c(\sigma_d(R))$ (۲) $\sigma_{c \wedge d}(R) \equiv \sigma_c(R) - \sigma_d(R)$
(۳) $\sigma_{c \vee d}(R) \equiv \sigma_c(R) \cap \sigma_d(R)$ (۴) $\sigma_c(R_1 \times R_2) \equiv R_1 \bowtie_c R_2$

۱۵- از دیدگاه کاد، تنویسین مدل رابطه‌ای، کدام یک از عملگرهای ذیل برای سیستم‌های شبه رابطه‌ای لازم می‌باشد؟

(مهندسی IT- آزاد ۸۴)

(۱) Project (۲) Select (۳) Join (۴) هر سه مورد

۱۶- کدام یک از عملگرهای جبر رابطه‌ای به عنوان عملگر اولیه نیست و می‌توان آن را براساس دیگر عملگرهای اولیه به دست آورد؟

(مهندسی IT- آزاد ۸۵)

(۱) Union (۲) Times (۳) Minus (۴) Intersect

۱۷- اگر A و B دو رابطه با عنوان یکسان باشند، عبارت زیر معادل با کدام گزینه است؟ (مهندسی IT- آزاد ۸۶)

$(A \text{ MINUS } (A \text{ MINUS } B)) \text{ UNION } (B \text{ MINUS } (B \text{ MINUS } A))$

(۱) $A \text{ INTERSECT } B$ (۲) $A \text{ DIVIDE } B$
(۳) $A \text{ JOIN } B$ (۴) $A \text{ MINUS } B$

۱۸- پایگاه داده‌ای دارای سه جدول S و L و G به صورت زیر است:

$S(S\#, Sname)$, $L(L\#, Lname)$, $G(S\#, L\#, grade)$

که در آن جدول S جدول دانشجو با خصیصه‌های شماره دانشجویی و نام دانشجو، جدول L جدول درس با خصیصه‌های شماره درس و نام درس و جدول G جدول نمره با خصیصه‌های شماره دانشجویی، شماره درس و نمره درس است. با توجه به این پایگاه داده به سؤال زیر پاسخ دهید.

جفت شماره دانشجویی و شماره درس‌هایی را مشخص کنید که برای دانشجوی مورد نظر نمره‌ای بابت درس مورد اشاره منظور نشده است. (مهندسی IT- آزاد ۸۶)

$$(1) (S \text{ JOIN } G) [S\#] \text{ TIMES } (L \text{ JOIN } G) [L\#]$$

$$(2) (S \text{ TIMES } L) [S\#, L\#] \text{ MINUS } G [S\#, L\#]$$

$$(3) (((S[S\#] \text{ MINUS } (S \text{ JOIN } G)[S\#]) \text{ JOIN } G) \text{ JOIN } L) [S\#, L\#]$$

$$(4) ((S \text{ TIMES } L) \text{ JOIN } G) [S\#, L\#]$$

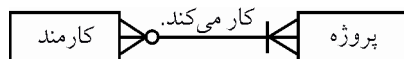
۱۹- نمودار ER روبه‌رو را در نظر بگیرید. با فرض آن‌که در موجودیت کارمند 200 رکورد و در موجودیت پروژه نیز 5 رکورد موجود باشد، حداقل و حداکثر تعداد رکوردها در پیوند طبیعی این دو موجودیت چیست؟ (مهندسی IT- دولتی ۸۵)

(۱) 0 و 1000

(۲) 5 و 200

(۳) 5 و 1000

(۴) 200 و 1000



۲۰- اگر A و B دو رابطه دارای اسامی خصیصه‌های یکسان باشند $A \bowtie B$ معادل با کدام یک از عبارات زیر است؟ (مهندسی کامپیوتر- دولتی ۸۷)

(۱) $A \times B$ (۲) $A \cap B$ (۳) $A \cup B$ (۴) هیچ کدام

۲۱- اگر r و s دو رابطه باشند و S و R به ترتیب Heading این دو رابطه باشند و با فرض آن‌که $S \subseteq R$ باشد، کدام یک از عبارات‌های زیر معادل با $r \div s$ است؟ (مهندسی IT- دولتی ۸۷)

$$(1) \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times s))$$

$$(2) \Pi_{R-S}(r) \cap \Pi_{R-S}((\Pi_{R-S}(r) \times s))$$

$$(3) \Pi_{R-S}((\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r))$$

$$(4) \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r))$$

۲۲- با توجه به جداول زیر کدام گزینه معادل سایر گزینه‌ها نیست؟ (مهندسی کامپیوتر- آزاد ۸۷)

prof (pnum, pname, esp, depnum)
dep (depnum, depname)

$$(1) \Pi_{pname, depname}(\sigma_{esp="DB"}(\text{prof} \bowtie \text{dep}))$$

$$(2) \Pi_{pname, depname} \left[\sigma_{esp="DB"} \left(\begin{array}{c} \text{prof} \times \text{dep} \\ \text{prof.depnum} = \text{dep.depnum} \end{array} \right) \right]$$

$$(3) \Pi_{pname, depname}(\sigma_{esp="DB"}(\text{prof} \times \text{dep}))$$

$$(4) \Pi_{pname, depname}(\sigma_{esp="DB"} \wedge \text{prof.depnum} = \text{dep.depnum}(\text{prof} \times \text{dep}))$$

۲۳- اگر T_1 و T_2 حاوی اطلاعات زیر باشند، $T_1 \bowtie T_2$ چند سطر خواهد داشت؟ (مهندسی IT- آزاد ۸۷)

A	B	C
1	5	1
2	7	3
2	7	5
2	2	7

A	C	E
1	1	3
2	3	3
2	3	5
2	2	7

16 (۴)

10 (۳)

3 (۲)

4 (۱)

۲۴- با توجه به جداول زیر مطلوب است دستور جبر رابطه‌ای برای تعیین مشخصات کامل داوطلبانی که در همه‌ی آزمون‌ها شرکت کرده‌اند؟ (مهندسی کامپیوتر- دولتی ۸۸)

داوطلب $S(S\#, Sname, address)$

آزمون $T(T\#, Tname, no - of - Ques)$

شرکت در آزمون $ST(S\#, T\#, Date, Time, Code)$

$$[\Pi_{S\#, T\#}(ST) \div \Pi_{T\#}(T)] \times S \quad (۲) \quad [\Pi_{S\#, T\#}(ST) \div \Pi_{T\#}(T)] \bowtie S \quad (۱)$$

$$S \bowtie [\Pi_{S\#, T\#}(ST) \div \Pi_{T\#, Tname}(T)] \quad (۳) \quad \Pi_{S\#, T\#}(ST) \div \Pi_{T\#, Tname}(T) \quad (۴)$$

۲۵- دو رابطه S و SL مفروض هستند، حاصل پرس و جوی زیر کدام است؟ (مهندسی کامپیوتر- آزاد ۸۵)

$S(S\#, sname)$ $SL(S\#, L\#, term\#, grade)$

$SL[S\#] \text{Minus } (SL \text{ where } grade < 20)[S\#]$

(۱) شماره دانشجویانی که حداقل یک نمره 20 دارند.

(۲) شماره دانشجویانی که تمام نمره‌های آن‌ها 20 است.

(۳) شماره دانشجویانی که حداقل یک نمره 20 ندارند.

(۴) شماره دانشجویانی که حداکثر یک نمره 20 دارند.

۲۶- رابطه $R(a, b, c)$ و دو عبارت جبری زیر را در نظر بگیرید:

$$Q_1 : \Pi_{b,c}(\sigma_{b=c}(R))$$

$$Q_2 : \Pi_{a,b}(\sigma_{a=b}(R))$$

(مهندسی کامپیوتر- دولتی ۸۹)

کدام عبارت همواره تعریف شده است؟

$$Q_1 \div Q_2 \quad (۱) \quad Q_1 \bowtie Q_2 \quad (۲) \quad Q_1 \cap Q_2 \quad (۳) \quad (۴) \text{ موارد ۱ و ۲}$$

۲۷- اگر A و B دو رابطه با یک قالب (Schema) باشند، آنگاه $A \bowtie B$ معادل کدام عبارت خواهد بود؟

(مهندسی IT- دولتی ۸۹)

$$A \times B \quad (۱) \quad A \times B \quad (۲) \quad A \cap B \quad (۳) \quad A \times B \quad (۴)$$

(مهندسی کامپیوتر- آزاد ۸۹ گروه الف)

۲۸- خروجی دستور زیر چند سطر خواهد داشت؟

$$\Pi_{T1.B}(T_1 \times T_2)$$

$$T_1.B = T_2.B$$

T1	
A	B
a1	b1
a2	b2
a3	b3

T2	
B	C
b1	c1
b1	c2
b2	c2
b2	c3
b2	c4

5 (۱)
2 (۲)
3 (۳)
15 (۴)

۲۹- با توجه به عبارت جبر رابطه‌ای دو سؤال query1 و query2 بر روی رابطه‌های S(b,c) و R(a,b) (جبر رابطه‌ای) کدام گزینه صحیح است؟ (مهندسی کامپیوتر- آزاد ۸۹ گروه ب)

Query1: $\Pi_{a,b}(R \bowtie S)$

Query2: $R \cap (\rho_{t(a)}(\Pi_C(s)) \times \Pi_b(s))$

- (۱) پاسخ‌های Query1 زیر مجموعه پاسخ‌های Query2 است.
 (۲) پاسخ‌های Query1 کاملاً متفاوت با پاسخ‌های Query2 است.
 (۳) پاسخ‌های Query2 زیر مجموعه پاسخ‌های Query1 است.
 (۴) هر دو Query پاسخ یکسانی را می‌دهند.

۳۰- سه جدول تهیه‌کننده S(S#, Sname, City) و قطعه P(P#, Pname, Color, City) محموله SP(S#, P#, QTY) را در نظر گرفته و به سؤال زیر پاسخ دهید.

کدام گزینه معادل پرس‌وجوی «اسامی همه تهیه‌کنندگانی را بیابید که حداقل یک قطعه آبی رنگ تهیه می‌کنند» می‌باشد. (مهندسی IT- آزاد ۸۹)

- (۱) $\Pi_{Sname}(\Pi_{S\#}(\Pi_{P\#}(\sigma_{Color="Blue"}(P)) \bowtie SP) \bowtie S)$
 (۲) $\Pi_{Sname}(\Pi_{S\#}(\Pi_{P\#}(P \div \sigma_{Color="Blue"}(P)) \bowtie SP) \bowtie S)$
 (۳) $\Pi_{Sname}(\Pi_{S\#}(\Pi_{P\#}(\sigma_{Color="Blue"}(P)) \bowtie SP) \div S)$
 (۴) $\Pi_{Sname}(\Pi_{S\#}(\Pi_{P\#}(P - \sigma_{Color="Blue"}(P)) \bowtie SP) \bowtie S)$

۳۱- کدام عبارت در مورد جبر رابطه‌ای نادرست است؟ (مهندسی IT- آزاد ۸۹)

- (۱) اگر A و B دو رابطه باشند کلید اصلی A MINUS B، کلید اصلی A می‌شود.
 (۲) در عملگر \bowtie همتایی عملوند (Same arity) الزامی است.
 (۳) عملگرهای اشتراک، تقسیم و پیوند طبیعی را می‌توان به وسیله عملگرهای دیگر در جبر رابطه‌ای شبیه‌سازی کرد.
 (۴) اگر رابطه A از درجه n باشد، 2^n تصویر مختلف از A وجود دارد.

۳۲- رابطه‌های R(A,B) و S(A,C,D) زیر مفروض‌اند: (مهندسی کامپیوتر- دولتی ۹۰)

R	
A	B
1	2
3	4

S		
A	C	D
1	2	3
1	3	4
2	4	5

$R \bowtie S$ چند تاپل خواهد داشت؟ ($S \bowtie R$: فرایوند طبیعی کامل - Natural full out or join)

2 (۱) 3 (۲) 4 (۳) 5 (۴)

۳۳- خروجی دستور زیر چند سطر خواهد داشت؟ (مهندسی کامپیوتر- آزاد ۹۰ گروه الف)

$\Pi_B(T_1 \bowtie T_2)$

T ₁		T ₂	
A	B	B	C
a ₁	b ₁	b ₁	c ₁
a ₂	b ₂	b ₁	c ₂
a ₃	b ₃	b ₂	c ₂
		b ₂	c ₃
		b ₂	c ₄

2 (۱) 5 (۲) 3 (۳) 15 (۴)

۳۴- دو رابطه $R(A,B)$ و $S(C,D)$ را در نظر بگیرید. کدام گزینه هم ارزی جبر رابطه‌ای‌ها را در دو رابطه درست نشان می‌دهد؟ (مهندسی کامپیوتر- آزاد ۹۰ گروه ب)

$$(۱) \Pi_{A,B,D}(R \bowtie S) = R \bowtie \rho_{T(B,D)}(S)$$

$$(۲) R - \rho_{T(A,B)}(S) = \rho_{T(A,B)}(S - \rho_{u(C,D)}(R))$$

$$(۳) \Pi_{A,B}(R \times S) = R$$

(۴) هیچ کدام

۳۵- سه جدول تهیه کننده $S(S\#, Sname, City)$ و قطعه $P(P\#, Pname, Color, City)$ و محموله $SP(S\#, P\#, QTY)$ را در نظر گرفته و به سؤال زیر پاسخ دهید. کدام گزینه پاسخ پرس و جوی «اسامی همه تهیه کنندگانی را بیاید که هیچ قطعه آبی رنگی تهیه نمی‌کنند» می‌باشد؟ (مهندسی IT- آزاد ۹۰)

$$(۱) \Pi_{Sname}(S - \Pi_{S\#}(\Pi_{P\#}(\sigma_{Color='Blue'}(P)) \bowtie SP) \bowtie S)$$

$$(۲) \Pi_{Sname}(\Pi_{S\#}(\Pi_{P\#}(P \div \sigma_{Color='Blue'}(P)) \bowtie SP) \bowtie S)$$

$$(۳) \Pi_{Sname}(\Pi_{S\#}(\Pi_{P\#}(\sigma_{Color='Blue'}(P)) \bowtie SP) \div S)$$

$$(۴) \Pi_{Sname}(\Pi_{S\#}(\Pi_{P\#}(\sigma_{Color='Blue'}(P)) \bowtie SP) \bowtie S)$$

با توجه به جدول زیر به دو سؤال بعد پاسخ دهید.

parent	(pid	,	ppid	,	sex)		
پدر یا مادر	شناسه شخص		شناسه پدر یا مادر		جنسیت		
person	(pid	,	pname	,	age	,	DOB)
شخص	شناسه شخص		نام		سن		تاریخ تولد

۳۶- پاسخ «شناسه پدر «آرش کمانگیر» در جبر رابطه کدام است؟ (\times یعنی نیم پیوند)

(مهندسی کامپیوتر- دولتی ۹۱)

(۱) $\Pi_{ppid}(\sigma_{sex="مرد"}(parent) \bowtie \sigma_{pname="آرش\ کمانگیر"}(person))$

(۲) $\Pi_{ppid}(\sigma_{pname="آرش\ کمانگیر"}(person) \times \sigma_{sex="مرد"}(parent))$

(۳) $\Pi_{ppid}(\sigma_{pname="آرش\ کمانگیر"}(person) \times \sigma_{sex="مرد"}(parent))$

(۴) $\Pi_{ppid}(\sigma_{sex="مرد"}(parent) \bowtie \sigma_{pname="آرش\ کمانگیر"}(person))$

۳۷- به کدام گزینه در مدل رابطه‌ای نمی‌توان پاسخ داد؟ (مهندسی کامپیوتر- دولتی ۹۱)

(۱) همسر فردی به نام «آرش کمانگیر» (۲) اجداد فردی به نام «آرش کمانگیر»

(۳) افرادی که همسر آنها بالای ۹۰ سال دارند. (۴) افرادی که سن آنها اشتباه محاسبه شده است.

۳۸- کدام یک از گزینه‌های ذیل به ازای هر دو رابطه R_1 و R_2 و رابطه $R_3 = R_1 \text{ op } R_2$ همواره درست

(مهندسی کامپیوتر- آزاد ۹۱)

است؟

(۱) اگر $op = \text{UNION}$ (اجتماع) و $R_1 = R_2$ باشد، آن‌گاه کلید اصلی R_3 اجتماع کلیدهای R_1 و R_2 است.

(۲) اگر $op = \text{DIV}$ (تقسیم) باشد آن‌گاه تمامی صفات خاصه R_3 جزو کلیدهای اصلی هستند.

(۳) اگر $op = \text{Intersection}$ (اشتراک) باشد آن‌گاه کلید اصلی برابر کلیه کلیدهای R_1 و R_2 است.

(۴) $op = *$ (ضرب دکارتی) باشد رابطه R_3 رابطه‌ای تمام کلید است (تمامی صفات R_3 جزو کلید اصلی هستند)

۳۹- کدام یک از معادله‌های ذیل در جبر رابطه‌ای نادرست است؟ (مهندسی کامپیوتر- آزاد ۹۱)

(۱) $R_1 \text{ intersect } R_2 = R_1 \text{ Minus } (R_1 \text{ Minus } R_2)$

(۲) $R_1 \text{ intersect } R_2 = R_2 \text{ Minus } (R_2 \text{ Minus } R_1)$

(۳) $R_1 \text{ intersect } R_2 = (R_1 \text{ Union } R_2) \text{ Minus } [(R_1 \text{ Minus } R_2) \text{ Minus } (R_2 \text{ Minus } R_1)]$

(۴) $R_1 \text{ Semiminus } R_2 = R_1 \text{ Minus } (R_1 \text{ Semijoin } R_2)$

۴۰- رابطه‌های: (مهندسی کامپیوتر- آزاد ۹۱)

(نام دانشجو STName، شماره دانشجویی STUD) STU : دانشجو

(نام درس: CTit، کد درس: CID) CRS : درس

(کد درس: CID، شماره دانشجویی: STID) SC : درس - دانشجو

را در نظر بگیرید. پرس‌وجوی ذیل چه چیزی را بر می‌گرداند؟ $(STU \text{ Times } CRS) [STID, CID] \text{ Minus } SC$

(۱) جفت شماره دانشجو و دروسی که در هر جفت، دانشجو درس را انتخاب نکرده است.

(۲) جفت شماره دانشجو و دروسی که ارائه شده را می‌دهد.

(۳) جفت شماره دانشجو و دروسی که ارائه نشده را می‌دهد.

(۴) تمام جفت شماره دانشجو و دروسی را می‌دهد که تمام دانشجویان درس را انتخاب کرده‌اند.

با توجه به بانک زیر، به سؤال زیر پاسخ دهید.

Person (pID , pName , Sex , DoB)

اطلاعات فرد شامل شناسه pID، نام pName، جنسیت sex و تاریخ تولد DoB.

Rel (pID1 , pID2 , pRel)

اطلاعات ارتباط بین افراد- رابطه pRel بین شخص pID1 و pID2 برقرار است.
به طور خاص، pRel=1 به این معنی است که شخص با شناسه pID2 پدر شخص با شناسه pID1 است.

۴۱- عبارت جبر رابطه‌ای زیر معادل کدام گزینه است؟ (مهندسی کامپیوتر- دولتی ۹۲)

$$(\Pi_{pID1} (\sigma_{pRel=1} (Rel))) \bowtie_{pID1=pID} Person$$

- (۱) فرزندان (۲) پدران
(۳) فرزندان کسانی که پدر دارند. (۴) پدران کسانی که فرزند دارند.

۴۲- کدام یک از موارد زیر بهترین تعریف برای بیان اتصال طبیعی (Natural Join) است:

(مهندسی کامپیوتر- آزاد ۹۲)

- (۱) یک عمل باینری است که از تعریف عبارت select و ضرب کارتیزین شکل می‌گیرد.
(۲) یک عمل باینری است که از ترکیب کامل اطلاعات دو رابطه به وجود می‌آید.
(۳) یک عمل ترکیبی است که از فقدان اطلاعات دو رابطه (Data Loss) جلوگیری می‌کند.
(۴) هیچ کدام

۴۳- کدام یک از موارد ذیل در رابطه با مقدار NULL (هیچ مقدار) نادرست است؟ (مهندسی کامپیوتر- آزاد ۹۲)

- (۱) چنانچه دستور select با آن مواجه شود به نتیجه اضافه می‌گردد.
(۲) در دستور projection با آن همانند سایر مقادیر رفتار می‌گردد.
(۳) در دستور union با آن همانند سایر مقادیر رفتار می‌گردد.
(۴) همه موارد

۴۴- رابطه زیر را در نظر بگیرید:

Course (cname, TA)

وجود تاپل (c,t) در این جدول نشان می‌دهد که t «کمک استاد» در درس c است.

می‌خواهیم همه زوج‌های (TA1,TA2) را به دست آوریم که TA1 و TA2 از هم متمایزند و حداقل در یک درس به عنوان کمک استاد با هم همکاری می‌کنند. می‌خواهیم هر زوج یک بار نوشته شود، به این معنی که (x,y) و (y,x) هم‌زمان در نتیجه آورده نشوند. کدام یک از عبارات جبر رابطه‌ای زیر این درخواست را به درستی توصیف می‌کنند؟ $\rho_a \rightarrow b$ نشان دهنده تغییر نام یک صفت از a به b است. (مهندسی کامپیوتر- دولتی ۹۳)

$$\Pi_{TA1,TA2}(\rho_{TA \rightarrow TA1}(Course) \bowtie \rho_{TA \rightarrow TA2}(Course)) \quad (1)$$

$$\Pi_{TA1,TA2}[\sigma_{TA1 <> TA2}(\rho_{TA \rightarrow TA1}(Course) \bowtie \rho_{TA \rightarrow TA2}(Course))] \quad (2)$$

$$\Pi_{TA1,TA2}[\sigma_{TA1 < TA2}(\rho_{TA \rightarrow TA1}(Course) \bowtie \rho_{TA \rightarrow TA2}(Course))] \quad (3)$$

(۴) هیچ کدام

۴۵- اگر رابطه r روی مجموعه صفت‌های X و رابطه s روی مجموعه صفت‌های Y تعریف شده باشند و اگر نیم‌الحاق (r semi-join s) را به صورت زیر تعریف کنیم، کدام گزینه زیر صحیح است؟

(مهندسی IT- دولتی ۹۳)

$$r \bowtie s := \Pi_X(r \bowtie s)$$

$$I: r \bowtie s := r \bowtie \Pi_{X \cap Y}(S)$$

$$II: r \bowtie s := (r \bowtie s) \bowtie s$$

$$III: r \bowtie s := ((r \bowtie s) \bowtie (s \bowtie r))$$

- (۱) تنها گزاره II صحیح است.
 (۲) گزاره‌های I و II صحیح هستند.
 (۳) گزاره‌های I و III صحیح هستند.
 (۴) هر سه گزاره صحیح است.

۴۶- کدام یک از روابط زیر در جبر رابطه‌ای صحیح نیست؟ (r و s رابطه هستند). (مهندسی IT - دولتی ۹۳)

$$(۱) r \times s = s \times r$$

$$(۲) r \bowtie \text{condition} s = s \bowtie \text{condition} r$$

$$(۳) \Pi_{\text{list1}}(\Pi_{\text{list2}} r) = \Pi_{\text{list2}}(\Pi_{\text{list1}} r)$$

$$(۴) \sigma_{\text{condition1}}(\sigma_{\text{condition2}} r) = \sigma_{\text{condition2}}(\sigma_{\text{condition1}} r)$$

۴۷- با داشتن رابطه‌های products و orderDetails که مربوط به کالاها و اقلام سفارش داده شده مشتریان است و هم چنین دو رابطه r و s کدام گزینه می‌تواند شناسه سفارش (ID) مشتریانی را بدهد که هیچ کالایی در گروه food نباشد که نخریده باشند؟ (مهندسی IT - دولتی ۹۳)

Products (Prod_ID, Group, UnitPrice)

orderDetails (ID, Prod_ID, Quantity)

r(ID, Prod_ID) = $\Pi_{\text{ID, Prod_ID}}$ (order Details)

s(Prod_ID) = $\Pi_{\text{Prod_ID}}(\sigma_{\text{Group}='food'} \text{products})$

$$(۱) r \div s \quad (۲) r \bowtie s \quad (۳) r \Rightarrow s \quad (۴) r \Leftarrow s$$

۴۸- کدام یک از عملگرهای زیر (عملگرهای جبر رابطه‌ای) در بانک اطلاعاتی نامتمرکز، کاربرد بیشتری دارد؟ (مهندسی کامپیوتر - آزاد ۹۳)

$$(۱) \text{ضرب دکارتی} \times \quad (۲) \text{نیم پیوند} \bowtie \quad (۳) \text{پرتو} \Pi \quad (۴) \text{گزینه‌ش} \sigma$$

۴۹- در جبر رابطه‌ای عملگر اضافی را می‌توان با ترکیب عملگرهای اصلی جایگزین نمود، کدام یک از عملگرهای زیر اضافی است؟ (مهندسی کامپیوتر - آزاد ۹۳)

$$(۱) - \quad (۲) \cup \quad (۳) \cap \quad (۴) \sigma$$

۵۰- چنانچه R_1 و R_2 دو رابطه با اسامی صفات یکسانی باشند، نتیجه الحاق R_1 و R_2 کدام گزینه است؟ (مهندسی IT - آزاد ۹۳)

$$(۱) R_1 - R_2 \quad (۲) R_1 \times R_2 \quad (۳) R_1 \cup R_2 \quad (۴) R_1 \cap R_2$$

۵۱- در جبر رابطه‌ای کدام یک از عملگرهای زیر نیاز به زمان و حافظه زیادی دارد؟ (مهندسی IT - آزاد ۹۳)

$$(۱) \text{نیم پیوند} \bowtie \quad (۲) \text{پیوند طبیعی} \bowtie \quad (۳) \text{پیوند شرطی} \theta \times \quad (۴) \text{ضرب دکارتی} \times$$

۵۲- چنانچه R یک رابطه با k صفت (ستون) باشد، تعداد تصاویر R که تهی نمی‌باشد برابر با کدام گزینه است؟ (مهندسی IT - آزاد ۹۳)

$$(۱) 2^k \quad (۲) 2^k - 1 \quad (۳) 2^k + 1 \quad (۴) k!$$

۵۳- چنانچه R_1 و R_2 دو رابطه در پایگاه داده باشند، عبارت $R_1 - (R_1 - R_2)$ برابر با کدام گزینه است؟ (مهندسی IT - آزاد ۹۳)

$$(۱) R_1 \cup R_2 \quad (۲) R_1 - R_2 \quad (۳) R_2 \quad (۴) R_1 \cap R_2$$

(مهندسی IT - آزاد ۹۳)

۵۴- کدام فرمول در جبر رابطه‌ای نادرست است؟

$$\begin{aligned} \sigma_p(a - b) &= \sigma_p(a) - \sigma_p(b) & (۲) & \sigma_p(a - b) = \sigma_p(a) - b & (۱) \\ a - b &\neq b - a & (۴) & \sigma_p(a \cap b) &= a - b & (۳) \end{aligned}$$

۵۵- سه رابطه students (دانشجویان)، courses (درس‌ها) و take (اخذ درس) را در نظر بگیرید:

students (sid, sname)**courses (cid, cname, dept)****take (sid, cid, grade)**

می‌خواهیم شماره دانشجویانی را پیدا کنیم که همه دروس ارائه شده در دانشکده 'CS' را اخذ کرده‌اند. کدام مورد دستور جبر رابطه‌ای صحیح برای این درخواست است؟ (مهندسی کامپیوتر - دولتی ۹۴)

$$\Pi_{sid}[take \div \sigma_{dept='CS'}(courses)] \quad (۱)$$

$$\Pi_{sid}[take \div \Pi_{cid}(\sigma_{dept='CS'}(courses))] \quad (۲)$$

$$\Pi_{sid}\left[\left(\Pi_{sid}(students) \times \Pi_{cid}(\sigma_{dept='CS'}(courses))\right) - \Pi_{sid,cid}(take)\right] \quad (۳)$$

$$\Pi_{sid}(students) - \Pi_{sid}\left[\left(\Pi_{sid}(students) \times \Pi_{cid}(\sigma_{dept='CS'}(courses))\right) - \Pi_{sid,cid}(take)\right] \quad (۴)$$

۵۶- (کار دستور $\rho_y(x)$ ، ایجاد یک کپی (یک نام مستعار) از رابطه x با نام y است.) اگر رابطه Emp شامل لیستی از کارمندان باشد که در این رابطه ویژگی Salary حقوق کارمند را نشان دهد، کار دستور زیر کدام است؟ (مهندسی IT - دولتی ۹۴)

$$\Pi_{salary}(Emp) - \Pi_{first.salary}(\sigma_{first.salary > second.salary}(\rho_{first}(Emp) \times \rho_{second}(Emp)))$$

(۱) لیست حقوق کارمندانی که حقوق مشابه آنها را کسی ندارد.

(۲) لیست حقوق کارمندان با حذف مقادیر تکراری

(۳) محاسبه ماکزیمم حقوق کارمندان

(۴) محاسبه مینیمم حقوق کارمندان

۵۷- اگر درس‌ها و پیش‌نیازهای آنها در جدول‌های زیر تعریف شده باشند، کدام یک از پیوندهای زیر می‌تواند وجود مشکلی در اطلاعات ثبت شده در سیستم را نشان دهد؟ (مهندسی IT - دولتی ۹۴)

Course (course-id, title, units)**Section (course-id, sec-id, semester, year)****Prereg (course-id, prereg-id)**

(prereg-id کلید خارجی مرتبط با فیلد course-id از جدول Course می‌باشد.)

$$Course \bowtie Prereg \quad (۲) \quad Course \Rightarrow \Leftarrow Section \quad (۱)$$

$$Section \Rightarrow \Leftarrow Prereg \quad (۴) \quad Course \Rightarrow \Leftarrow Prereg \quad (۳)$$

۵۸- با توجه به پایگاه داده زیر، عبارت جبر رابطه‌ای معادل پرس و جوی زیر کدام است؟

«نام تمامی اعضای را بیابید که همه کتاب‌های منتشر شده توسط McGraw-Hill را به امانت برده‌اند.»

Member (mem-no, name, age)**Book (ISBN, title, author, publisher)****Borrowed (mem-no, ISBN, date)**

(مهندسی IT - دولتی ۹۵)

$$\Pi_{\langle \text{name} \rangle} \left(\left(\Pi_{\langle \text{isbn}, \text{name}, \text{mem-no} \rangle} \text{Borrowed} \bowtie \text{Member} \right) \div \sigma_{\text{publisher} = \text{'McGraw-Hill' Book}} \right) \quad (۱)$$

$$\Pi_{\langle \text{name} \rangle} \left(\left(\text{Borrowed} \div \sigma_{\text{publisher} = \text{'McGraw-Hill' Book}} \right) \bowtie \text{Member} \right) \quad (۲)$$

$$\Pi_{\langle \text{name} \rangle} \left(\left(\Pi_{\langle \text{mem-no}, \text{isbn} \rangle} \text{Borrowed} \div \Pi_{\langle \text{isbn} \rangle} \left(\sigma_{\text{publisher} = \text{'McGraw-Hill' Book}} \right) \right) \bowtie \text{Member} \right) \quad (۳)$$

$$\Pi_{\langle \text{name} \rangle} \left(\text{Borrowed} \bowtie \text{Member} \right) \div \Pi_{\langle \text{isbn} \rangle} \left(\sigma_{\text{publisher} = \text{'McGraw-Hill' Book}} \right) \quad (۴)$$

۵۹- عبارت جبر رابطه‌ای «عنوان کتاب‌هایی از انتشارات wiley که توسط هیچ عضوی امانت گرفته نشده» معادل

(مهندسی کامپیوتر- دولتی ۹۴)

کدام است؟

کتاب‌ها: **Book (ISBN, Title, Author, publisher)**

اعضاء: **Member (MID, Name, MDate)**

کتاب: **Borrow (MID, ISBN, BDate, Duration)**

$$\Pi_{\langle \text{Title} \rangle} \left(\sigma_{\text{publisher} = \text{'wiley' (Book)}} - \Pi_{\langle \text{ISBN} \rangle} (\text{Borrow}) \right) \quad (۱)$$

$$\Pi_{\langle \text{Title, ISBN} \rangle} \left(\sigma_{\text{publisher} = \text{'wiley' (Book)}} \right) \div \Pi_{\langle \text{ISBN} \rangle} (\text{Borrow}) \quad (۲)$$

$$\Pi_{\langle \text{Title} \rangle} \left(\left(\Pi_{\langle \text{ISBN, MID} \rangle} (\text{Borrow}) \div \Pi_{\langle \text{MID} \rangle} (\text{Member}) \right) \bowtie \text{Borrow} \right) \quad (۳)$$

$$\Pi_{\langle \text{Title} \rangle} \left(\left(\Pi_{\langle \text{ISBN} \rangle} \left(\sigma_{\text{publisher} = \text{'wiley' (Book)}} \right) - \Pi_{\langle \text{ISBN} \rangle} (\text{Borrow}) \right) \bowtie \text{Book} \right) \quad (۴)$$

۶۰- بر فرض اینکه R_1 و R_2 دو رابطه باشند که هیچ صفت مشترکی نداشته باشند، اگر CK_1 تنها کلید

کандید رابطه R_1 و CK_2 تنها کلید رابطه R_2 باشد، آنگاه کلید کاندید رابطه حاصل از ضرب کارتیزین

(مهندسی کامپیوتر- دولتی ۹۴)

R_1 و R_2 (یعنی $R_1 \times R_2$)، کدام است؟

$$CK_1 \cap CK_2 \quad (۱)$$

$$CK_1 \cup CK_2 \quad (۲)$$

$$CK_1 \times CK_2 \quad (۳)$$

$$CK_1, CK_2 \text{ هر دو کلید کاندید رابطه حاصل هستند.} \quad (۴)$$

۶۱- پایگاه داده رابطه‌ای زیر را در نظر بگیرید. می‌خواهیم نام اعضای را بیابیم که همه کتاب‌های منتشر شده

توسط McGraw-Hill را به امانت برده‌اند. کدام مورد، دستور جبر رابطه‌ای درست برای این درخواست

(مهندسی IT- دولتی ۹۴)

است؟

Member (**ID**, Name, age)

Book (**ISBN**, Title, publisher)

Borrowed (**ID**, ISBN, Date)

$$\Pi_{\langle \text{Name} \rangle} \left[\text{Member} \bowtie \Pi_{\langle \text{ID} \rangle} \left(\text{Borrowed} \div \left(\sigma_{\text{publisher} = \text{'McGraw-Hill' Book}} \right) \right) \right] \quad (۱)$$

$$\Pi_{\langle \text{Name} \rangle} \left[\text{Member} \times \Pi_{\langle \text{ID} \rangle} \left(\text{Borrowed} \div \left(\sigma_{\text{publisher} = \text{'McGraw-Hill' Book}} \right) \right) \right] \quad (۲)$$

$$\Pi_{\langle \text{Name} \rangle} \left[\text{Member} \bowtie \left(\left(\Pi_{\langle \text{ID, ISBN} \rangle} \text{Borrowed} \right) \div \left(\Pi_{\langle \text{ISBN} \rangle} \left(\sigma_{\text{publisher} = \text{'McGraw-Hill' Book}} \right) \right) \right) \right] \quad (۳)$$

$$\Pi_{\langle \text{Name} \rangle} \left[\text{Member} \times \left(\left(\Pi_{\langle \text{ID, ISBN} \rangle} \text{Borrowed} \right) \div \left(\Pi_{\langle \text{ISBN} \rangle} \left(\sigma_{\text{publisher} = \text{'McGraw-Hill' Book}} \right) \right) \right) \right] \quad (۴)$$

(مهندسی IT- دولتی ۹۸)

۶۲- پایگاه داده زیر را در نظر بگیرید:

node (NID, Name, Color, Description) :گره‌ها

اطلاعات موجود در جدول **node** شامل شماره، نام، رنگ و شرح مربوط به هر گره است.

یال‌ها: $\text{edge}(\text{NID1}, \text{NID2}, \text{EdgeType})$

هر سطر از جدول **edge**، نشان دهنده وجود یک یال جهت‌دار از نوع **EdgeType** از گره با شماره **NID1** به گره با شماره **NID2** است.

عبارت جبر رابطه‌ای زیر معادل کدام مورد است؟ (عملگر $\rho_{R2}(R1)$ ، نام رابطه **R1** را به **R2** تغییر می‌دهد.)

$$\Pi_{E1.NID1} \left(\sigma_{E1.NID2 = E2.NID1} \left(\sigma_{E1.EdgeType = T2} \left(\rho_{E1(Edge)} \times \rho_{E2(Edge)} \right) \right) \right)$$

(۱) شماره گره‌هایی که حداقل یک یال از نوع **T2** از آنها خارج شده است.

(۲) شماره گره‌هایی که حداقل یک یال خروجی از نوع **T2** به یک گره مانند **g** دارند و گره **g** حداقل یک یال خروجی دارد.

(۳) شماره گره‌هایی که حداقل یک یال ورودی از نوع **T2** از یک گره مانند **g** دریافت می‌کنند و گره **g** حداقل یک یال خروجی دارد.

(۴) شماره گره‌هایی که حداقل یک یال خروجی از نوع **T2** به یک گره مانند **g** دارند و گره **g** حداقل یک یال خروجی از نوع **T2** دارد.

۶۳- با توجه به جدول ارائه شده، کدام عبارت جبری درست است؟

(مهندسی کامپیوتر- دولتی ۹۹)
STUD(S#, Sname, City, avg, clg#)

(شماره دانشکده، معدل، محل تولد، نام، شماره دانشجویی) جدول دانشجو

$$\Pi_{City}(\sigma_{avg>16}(STUD)) = \sigma_{avg>16}(\Pi_{City}(STUD)) \quad (۱)$$

$$\sigma_{City}(\Pi_{avg>16}(STUD)) = \Pi_{avg>16}(\sigma_{City}(STUD)) \quad (۲)$$

$$\Pi_{City}(\sigma_{avg>16}(STUD)) \quad (۳)$$

$$\sigma_{avg>16}(\Pi_{City}(STUD)) \quad (۴)$$

۶۴- کار عملگر $\rho_{second}(Marks)$ این است که یک کپی از رابطه‌ی **Marks** با نام مستعار **Second** ایجاد کند.

(مهندسی کامپیوتر- دولتی ۹۹)

کار دستور زیر در جبر رابطه‌ای، کدام است؟

$$\Pi_{mark}(Marks) - \Pi_{Marks.mark}(Marks \times \rho_{second}(Marks))$$

$$Marks.mark > Second.mark$$

(۱) یک لیست تهی بر می‌گرداند.

(۲) نمره مینیمم را از لیست نمرات استخراج می‌کند.

(۳) نمره ماکزیمم را از لیست نمرات استخراج می‌کند.

(۴) نمره‌هایی که در جدول **Marks** موجود نیستند را نشان می‌دهد.

۶۵- فرض کنید $pk(R)$ کلید اصلی رابطه **R** را نشان می‌دهد. در مورد یک ارتباط چند به یک که بین دو رابطه

(مهندسی IT- دولتی ۹۹)

R1 و **R2** وجود دارد، کدام گزینه صحیح است؟

$$pk(R1) \rightarrow pk(R2) \quad (۲)$$

$$pk(R2) \rightarrow pk(R1) \quad (۱)$$

$$pk(R1) \rightarrow R1 \cap R2 \quad (۴)$$

$$pk(R2) \rightarrow R1 \cap R2 \quad (۳)$$

۶۶- شمای رابطه‌ای زیر، پایگاه داده موسسات آموزش هنر است.
 در این پایگاه داده اسامی هنرجویانی که در هر موسسه عضو هستند ذخیره شده است.
 جدول هنرهای مورد علاقه، نام رشته‌های هنری مورد علاقه هر هنرجو را نشان می‌دهد.
 جدول دوره‌های هنری نشان می‌دهد در هر موسسه چه رشته‌های هنری ارائه می‌شود.

(مهندسی کامپیوتر-دولتی-۱۴۰۰)

Student (SID , Name)

Institute (IID , IName , IAddress)

Membership (SID, IID)

Faviorate Field (SID , Field)

Offered Field (IID , Field)

کدام جبر رابطه‌ای لیست تمام هنرجوها را می‌دهد که فقط در موسسه‌هایی عضوند که هیچ رشته هنری خارج از علاقه‌مندی آنها را ارائه نمی‌دهد؟

$$\Pi_{SID}(\text{Faviorate Field} \bowtie \text{Membership} \bowtie \text{Offered Field}) \quad (۱)$$

$$\Pi_{SID}(\text{Membership}) - \Pi_{SID}(\text{Faviorate Field} \bowtie \text{Membership} \bowtie \text{Offered Field}) \quad (۲)$$

$$\Pi_{SID}(\text{Membership}) - \Pi_{SID}(\text{Membership} - \Pi_{SID, IID}(\text{Faviorate Field} \bowtie \text{Offered Field})) \quad (۳)$$

$$\Pi_{SID}(\text{Membership}) - \Pi_{SID}(\Pi_{SID, Field}(\text{Membership} \bowtie \text{Offered Field}) - \text{Faviorate Field}) \quad (۴)$$

۶۷- هم‌ارزی‌های جبر رابطه‌ای زیر را در نظر بگیرید. این هم‌ارزی‌ها ممکن است همواره درست باشند، در بعضی شرایط درست باشند، یا همواره نادرست باشند. در این عبارت‌ها، R یک رابطه (Relation)، ciها شرط‌هایی بر روی R و aiها زیر مجموعه‌هایی از صفت‌های R هستند. کدام هم‌ارزی همواره درست است؟
 (مهندسی کامپیوتر-دولتی-۱۴۰۰)

$$\Pi_{a1}(\Pi_{a2}(R)) \equiv \Pi_{a2}(\Pi_{a1}(R)) \quad (۲) \quad \sigma_{c1}(\sigma_{c2}(R)) \equiv \sigma_{c2}(\sigma_{c1}(R)) \quad (۱)$$

$$\Pi_{a1}(\Pi_{a2}(R)) \equiv \Pi_{a1}(R) \quad (۴) \quad \Pi_{a1}(\sigma_{c1}(R)) \equiv \sigma_{c1}(\Pi_{a1}(R)) \quad (۳)$$

۶۸- رابطه $R = (a, b, c)$ و عبارات جبر رابطه‌ای زیر را در نظر بگیرید: (مهندسی IT- دولتی ۱۴۰۰)

$$Q_1 : \rho(S(a, b_1, c_1), R) \quad Q_2 : \Pi_{b, c}(\sigma_{b=c} R)$$

$$\rho(T, S \bowtie R)$$

$$\Pi_{b_1, c}(\sigma_{b_1=c} T)$$

- (۱) Q_1 و Q_2 پاسخ‌های یکسان تولید می‌کنند. (۲) Q_1 و Q_2 پاسخ‌های متفاوت تولید می‌کنند.
 (۳) پاسخ Q_2 زیر مجموعه‌ای از پاسخ Q_1 است. (۴) پاسخ Q_1 زیر مجموعه‌ای از پاسخ Q_2 است.

باسخ تست‌های فصل چهارم: جبر رابطه‌ای

۱- گزینه (۱) صحیح است.

گزینه اول درست است. زیرا، عملگر نیم پیوند در گزینه اول، پرس و جوی مطرح شده در صورت سؤال را شبیه‌سازی می‌کند.

$$\Pi_{a.*}(\theta \wedge_{a.FK=b.FK} \sigma(a \times b)) = \Pi_{a.*}(\sigma_{\theta}(a.FK=b.FK))$$

جابه‌جایی (اگر چه مشروط است) همه ستون‌های جدول a

باید شرط σ زیر مجموعه Π باشد، $a.*$ همه مؤلفه‌های θ را شامل می‌شود. پس جابه‌جایی امکان‌پذیر است.

$$\sigma_{\theta} \left[\Pi_{a.*} \left[\sigma(a \times b) \right]_{a.FK=b.FK} \right] \Rightarrow \sigma_{\theta}(a \times b)$$

تعریف نیم پیوند چپ

گزینه دوم نادرست است، زیرا، با استفاده از عملگر الحاق طبیعی علاوه بر ستون‌های رابطه a، ستون‌های رابطه b را نیز تولید می‌کند.

گزینه سوم نادرست است، زیرا، شرط θ را بیان نکرده است، هر چند ستون‌های جدول a را فقط تولید می‌کند.

گزینه چهارم نادرست است، زیرا، دارای خطای نحوی است، خروجی حاصل از عملگر نیم پیوند چپ در پرانتز داخلی فقط ستون‌های جدول a را تولید می‌کند. حال چطور می‌توان در پرانتز خارجی ستون‌های جدول b را تولید نمود!

۲- گزینه (۳) صحیح است.

به مثال عددی زیر توجه کنید:

جدول حرارت :

کد ناحیه	نام	تاریخ	حرارت بالا	حرارت پایین
1	تهران	x	30	1
2	اهواز	y	50	20
3	شیراز	z	40	10

رابطه‌ی R_1 : حرارت‌های بالای تمامی نواحی مختلف در تاریخ‌های متفاوت را نشان می‌دهد. هم‌چنین نام ستون حرارت بالا به H تغییر کرده است.

R_1 :

H	تاریخ	کد ناحیه
30	X	1
50	Y	2
40	Z	3

رابطه‌ی R_2 : حرارت‌های پایین تمامی نواحی مختلف در تاریخ‌های متفاوت را نشان می‌دهد. هم‌چنین نام ستون حرارت پایین به L تغییر کرده است.

L	تاریخ	کد ناحیه
1	x	1
20	y	2
10	z	3

رابطه R_3 : سطرهایی از دو جدول R_1 و حرارت را انتخاب می‌کند که شرط «حرارت بالا $H <$ » در آن برقرار باشد. به عبارت دیگر رابطه R_3 ، کد گروهی از نواحی در رابطه R_1 را می‌دهد که بیش‌ترین حرارت آن‌ها از بیش‌ترین حرارت یکی از نواحی دیگر در جدول حرارت کم‌تر است.

H	حرارت بالا	کد ناحیه
30	40	1
40	50	3

 $\xrightarrow{R_3}$

کد ناحیه
1
3

رابطه R_4 : سطرهایی از دو جدول R_2 و حرارت را انتخاب می‌کند که شرط «حرارت پایین $L >$ » در آن برقرار باشد. به عبارت دیگر رابطه R_4 ، کد گروهی از نواحی در رابطه R_2 را می‌دهد که کم‌ترین حرارت آن‌ها از کم‌ترین حرارت یکی از نواحی دیگر در جدول حرارت بیش‌تر است.

L	حرارت پایین	کد ناحیه
20	1	2
10	1	3

 $\xrightarrow{R_4}$

کد ناحیه
2
3

در نتیجه عبارت $R_3 - R_4$ (حرارت) - کد ناحیه Π ، کد دسته‌ای از نواحی را می‌دهد که حرارت بیشینه دارند.

کد ناحیه
1
2
3

 $-$

کد ناحیه
1
3

 $=$

کد ناحیه
2

هم‌چنین عبارت $R_4 - R_3$ (حرارت) - کد ناحیه Π ، کد دسته‌ای از نواحی را می‌دهد که حرارت کمینه دارند.

کد ناحیه
1
2
3

 $-$

کد ناحیه
2
3

 $=$

کد ناحیه
1

به این ترتیب اجتماع آن‌ها و الحاق طبیعی با جدول حرارت، نام نواحی با درجه حرارت بیشینه و کمینه را به عنوان پاسخ نهایی می‌دهد. بنابراین تهران و اهواز در خروجی نمایش داده می‌شود.

۳- گزینه (۱) صحیح است.

نتیجه الحاق طبیعی این دو رابطه به صورت زیر خواهد بود. چنان‌که ملاحظه می‌شود چون در الحاق طبیعی ستون مشترک یک بار در خروجی ظاهر می‌شود، خروجی این عبارت چهار ستون دارد. هم‌چنین فقط

سطرهایی در خروجی ظاهر می‌شوند که مقدارشان در ستون مشترک (STU-ID) برابر باشد. به این ترتیب حاصل الحاق طبیعی دو رابطه سه سطر خواهد داشت.

STU-ID	NAME	COURSE-ID	TITLE
S001	AHMADI	CSC100	PASCAL PROC.
S001	AHMADI	CSC200	COMPILER WRITING
S002	TAVANA	CSC100	PASCAL PROC.

۴- گزینه (۲) صحیح است.

نتیجه الحاق خارجی راست و الحاق خارجی چپ این دو رابطه مطابق تعریف این دو عملگر و محتویات داده شده برای دو رابطه، چهار سطر خواهد داشت. هم‌چنین نتیجه الحاق خارجی کامل این دو رابطه پنج سطر دارد. زیرا در الحاق خارجی کامل علاوه بر تاپل‌های حاصل از الحاق طبیعی، تاپل‌های آویزان مثل S003 و S004 به نتیجه نهایی اضافه می‌گردد. توجه کنید که در صورت سؤال مشخص نشده است که منظور از الحاق خارجی کدام نوع الحاق خارجی است. از آن‌جا که در برخی منابع، منظور از الحاق خارجی همان الحاق خارجی کامل است. گزینه دوم به عنوان جواب انتخاب می‌شود.

۵- گزینه (۱) صحیح است.

گزینه دوم تعریف عملگر اشتراک را بیان می‌کند. عملگر \bowtie دارای خاصیت جابه‌جایی است اما عملگر \times دارای خاصیت جابه‌جایی نمی‌باشد. بنابراین گزینه‌های دوم تا چهارم عبارات درستی را در جبر رابطه‌ای بیان می‌کنند. اما گزینه اول با توجه به تعریف عملگرهای اجتماع، انتخاب و الحاق طبیعی، در جبر رابطه‌ای عبارتی درست تلقی نمی‌شود.

۶- گزینه‌های (۲ و ۴).

هنگامی دو رابطه اجتماع‌پذیرند که شروط سازگاری به معنی خصیصه‌های یکسان (تیرهای مشابه) برای دو رابطه برقرار باشد. طبق تعریف اگر R و S دو رابطه باشند کلید RUS معادل است با اجتماع تمام خصیصه‌های R یا S. بنابراین گزینه دوم البته با اصلاحی جزئی درست است.

مثال:

R:	S:	=																		
<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr><td style="padding: 2px 5px;"><u>a</u></td><td style="padding: 2px 5px;"><u>b</u></td><td style="padding: 2px 5px;"><u>c</u></td></tr> <tr><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">2</td><td style="padding: 2px 5px;">4</td></tr> <tr><td style="padding: 2px 5px;">2</td><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">5</td></tr> </table>	<u>a</u>	<u>b</u>	<u>c</u>	1	2	4	2	1	5	U	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr><td style="padding: 2px 5px;"><u>a</u></td><td style="padding: 2px 5px;"><u>b</u></td><td style="padding: 2px 5px;"><u>c</u></td></tr> <tr><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">2</td><td style="padding: 2px 5px;">6</td></tr> <tr><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">3</td><td style="padding: 2px 5px;">7</td></tr> </table>	<u>a</u>	<u>b</u>	<u>c</u>	1	2	6	1	3	7
<u>a</u>	<u>b</u>	<u>c</u>																		
1	2	4																		
2	1	5																		
<u>a</u>	<u>b</u>	<u>c</u>																		
1	2	6																		
1	3	7																		
		<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr><td style="padding: 2px 5px;"><u>a</u></td><td style="padding: 2px 5px;"><u>b</u></td><td style="padding: 2px 5px;"><u>c</u></td></tr> <tr><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">2</td><td style="padding: 2px 5px;">4</td></tr> <tr><td style="padding: 2px 5px;">2</td><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">5</td></tr> <tr><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">2</td><td style="padding: 2px 5px;">6</td></tr> <tr><td style="padding: 2px 5px;">1</td><td style="padding: 2px 5px;">3</td><td style="padding: 2px 5px;">7</td></tr> </table>	<u>a</u>	<u>b</u>	<u>c</u>	1	2	4	2	1	5	1	2	6	1	3	7			
<u>a</u>	<u>b</u>	<u>c</u>																		
1	2	4																		
2	1	5																		
1	2	6																		
1	3	7																		

توجه کنید که اجتماع کلیدهای دو جدول R و S یعنی (a,b) خاصیت کلیدی ندارد! با توجه به یکسان بودن صفات دو رابطه R و S برای عملگر اجتماع، مجموعه صفات حاصل از تقاطع یا اشتراک صفات دو جدول، کل صفات را شامل می‌شود و جواب مشابه گزینه دوم خواهد بود.

۷- گزینه (۴) صحیح است.

همان‌طور که می‌دانید یک مجموعه با n عضو، دارای 2^n زیر مجموعه است که البته یکی از آن‌ها مجموعه

تهی است. بنابراین اگر رابطه‌ای دارای n خصیصه باشد، تعداد کل تصاویر (به معنی عملگر پرتو یا Π) آن، 2^n خواهد بود که اگر تصویر تهی آن را کنار بگذاریم برابر $2^n - 1$ تصویر غیرتهی می‌شود.
مثال: جدول زیر را با سه ستون در نظر بگیرید، تعداد تصاویر غیرتهی برابر چیست؟

X	Y	Z	تصویر
0	0	0	تهی
0	0	1	Z
0	1	0	Y
0	1	1	YZ
1	0	0	X
1	0	1	XZ
1	1	0	XY
1	1	1	XYZ

که جدول بدون ستون وجود ندارد. \Rightarrow

بنابراین تعداد تصاویر غیرتهی برابر مقدار $2^3 - 1 = 7$ خواهد بود، که در یک عبارت کلی رابطه $2^n - 1$ را که n برابر تعداد ستون‌ها است، خواهیم داشت.

همچنین در یک راه حل دیگر داریم:

هر تصویر غیرتهی از جدول A، شامل 1، 2، 3، ... یا n صفت از جدول A است. بنابراین تعداد آنها برابر است با:

$$\binom{n}{1} + \binom{n}{2} + \binom{n}{3} + \dots + \binom{n}{n} = 2^n - 1$$

۸- گزینه (۲) صحیح است.

در یک قاعده کلی اگر R یک رابطه، L مجموعه‌ای از صفت‌ها و θ مجموعه‌ای از شروط باشد، آنگاه تساوی:
 $\Pi_L(\sigma_\theta(R)) = \sigma_\theta(\Pi_L(R))$
برقرار است. اگر و فقط اگر تمامی ستون‌های θ زیرمجموعه L باشند، یا به عبارت دیگر شرط θ فقط ستون‌های موجود در مجموعه L را در بر گیرد.

۹- گزینه (۳) صحیح است.

پیوند طبیعی عملگری است که دو رابطه را بر مبنای یک یا چند صفت مشترک به هم پیوند می‌دهد. در پیوند طبیعی دو رابطه R و S یعنی $R \bowtie S$ اگر R و S صفت مشترکی نداشته باشند، آنگاه تبدیل به $R \times S$ خواهد شد و اگر روابط R و S تماماً دارای صفات یکسانی باشند، آنگاه $R \bowtie S$ معادل $R \cap S$ خواهد شد. مثال زیر گویای مطلب می‌باشد:

$$\begin{array}{c|c|c} a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \end{array} \bowtie \begin{array}{c|c|c} a & b & c \\ \hline 1 & 2 & 3 \\ \hline 1 & 2 & 3 \end{array} = \begin{array}{c|c|c} a & b & c \\ \hline 1 & 2 & 3 \\ \hline 1 & 2 & 3 \end{array}$$

$$\begin{array}{c|c|c} a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \end{array} \cap \begin{array}{c|c|c} a & b & c \\ \hline 1 & 2 & 3 \\ \hline & & \end{array} = \begin{array}{c|c|c} a & b & c \\ \hline 1 & 2 & 3 \\ \hline & & \end{array}$$

۱۰- گزینه (۴) صحیح است.

جدول هتل 35 سطر و جدول اتاق 1500 سطر دارد، پس تعداد رکوردهای حاصل از ضرب دکارتی دو جدول برابر با ضرب تعداد رکوردهای دو جدول در یکدیگر است. بنابراین حاصل ضرب دکارتی برابر $35 \times 1500 = 52500$ سطر خواهد بود. در ضرب دکارتی قاعده‌ی زیر برقرار است:

$$R \times S = (R \bowtie S) + (R \bowtie S)$$

با شرط نابرابری با شرط برابری

پس بنا بر رابطه‌ی فوق داریم:

$$\text{اتاق} \times \text{هتل} = \left(\begin{array}{c} \text{اتاق} \bowtie \text{هتل} \\ \text{کد هتل. اتاق} = \text{کد هتل. هتل} \end{array} \right) + \left(\begin{array}{c} \text{اتاق} \bowtie \text{هتل} \\ \text{کد هتل. اتاق} \neq \text{کد هتل. هتل} \end{array} \right)$$

اما به خروجی $\left(\begin{array}{c} \text{اتاق} \bowtie \text{هتل} \\ \text{کد هتل. اتاق} = \text{کد هتل. هتل} \end{array} \right)$ توجه کنید، در جدول هتل، اطلاعات مربوط به هتل، همچون کد هتل وجود دارد، در جدول اتاق نیز، توسط آوردن کد هتل، مشخص شده است که هر اتاق مربوط به کدام هتل است، بنابراین در هر سطر موجود در جدول اتاق، این که کدام اتاق، از کدام هتل است، مشخص شده است. بنابراین پیوند شرطی حاصل از دو جدول هتل و اتاق با شرط برابری کد هتل در خصیصه مشترک کد هتل، برابر با تعداد سطرهای جدول اتاق خواهد شد. به این مطالب کمی فکر کنید، زیرا در این حالت همه کدهای هتل موجود در جدول اتاق حتماً زیر مجموعه کدهای هتل، در جدول هتل خواهد بود. کدام اتاق از کدام هتل. بنابراین حاصل این پیوند شرطی برابر 1500 سطر خواهد بود، یعنی به تعداد سطرهای جدول اتاق.

توجه کنید که در جدول اتاق، اطلاعات مربوط به همه اتاق‌های همه هتل‌ها آورده شده است، خواه اقامتی در آن اتاق باشد، خواه نباشد. اطلاعات اقامت مسافران در اتاق، در جدول وضعیت اتاق قرار دارد.

$$52500 = 1500 + \left(\begin{array}{c} \text{اتاق} \bowtie \text{هتل} \\ \text{کد هتل. اتاق} \neq \text{کد هتل. هتل} \end{array} \right)$$

$$\left(\begin{array}{c} \text{اتاق} \bowtie \text{هتل} \\ \text{کد هتل. اتاق} \neq \text{کد هتل. هتل} \end{array} \right) = 52500 - 1500 = 51000$$

۱۱- گزینه (۱) صحیح است.

در صورت سوال خروجی $\left(\begin{array}{c} \text{وضعیت اتاق} \bowtie \text{هتل} \\ \text{کد هتل. وضعیت اتاق} = \text{کد هتل. هتل} \end{array} \right)$ خواسته شده است، هم‌چنین، اطلاعات اقامت مسافران در اتاق، در جدول وضعیت اتاق قرار دارد. اما مطابق آنچه پیش از این در پاسخ قبل گفتیم، در جدول هتل، اطلاعات مربوط به هتل، همچون کد هتل وجود دارد، در جدول وضعیت اتاق نیز، توسط آوردن کد هتل، مشخص شده است که هر اتاق دارای اقامت مسافر، مربوط به کدام هتل است، بنابراین هر سطر موجود در جدول وضعیت اتاق، این که کدام اتاق، از کدام هتل است، مشخص شده است. بنابراین

پیوند شرطی حاصل از دو جدول هتل و وضعیت اتاق با شرط برابری کد هتل در خصیصه مشترک کد هتل برابر با تعداد سطرهای جدول وضعیت اتاق خواهد شد. به این مطلب باز هم کمی فکر کنید، زیرا در این حالت هم کدهای هتل موجود در جدول وضعیت اتاق حتماً زیر مجموعه کدهای هتل، در جدول هتل خواهد بود، کدام اتاق از کدام هتل. بنابراین حاصل این پیوند شرطی برابر R سطر خواهد بود، یعنی به تعداد سطرهای جدول وضعیت اتاق.

۱۲- گزینه (۴) صحیح است.

گزینه دوم و سوم نادرست است، زیرا، به طور کلی در جبر رابطه‌ای مفاهیمی همچون اولین، آخرین و یکی از، معنا و مفهومی ندارد. در واقع اگر شرط انتخاب سطر، برای سطرهای برقرار باشد، سطر مورد نظر در خروجی قرار می‌گیرد. جداول زیر را در نظر بگیرید:

S#	sname	city	S#	P#	QTY	P#	Pname	...
S ₁	S _{n1}	LONDON	S ₁	P ₁	10	P ₁		
S ₂	S _{n2}	PARIS	S ₁	P ₂	20	P ₁		
S ₃	S _{n3}	PARIS	S ₂	P ₁	30	P ₂		
S ₄	S _{n4}	PARIS	S ₂	P ₂	40			
			S ₃	P ₁	50			

جدول S

جدول SP

جدول P

مطابق پرس و جوی مطرح شده در صورت سؤال داریم:

$$\Pi_{S\#}(\sigma_{city='PARIS'}(S \bowtie SP))$$

ابتدا، در داخلی‌ترین پرانتز، مشخصات تولیدکنندگانی که حداقل یک قطعه تولید کرده‌اند، ایجاد می‌گردد. سپس، در ادامه، توسط شرط انتخاب سطر، فقط، مشخصات تولیدکنندگانی که حداقل یک قطعه تولید کرده‌اند و اهل شهر پاریس نیز هستند، ایجاد می‌گردد. در آخر، توسط عملگر پرتو، فقط، شماره تولیدکنندگانی که حداقل یک قطعه تولید کرده‌اند و اهل شهر پاریس نیز هستند، ایجاد می‌گردد. مطابق جداول مذکور، خروجی به صورت زیر خواهد بود:

S#
S ₂
S ₃

عبارت جبر رابطه‌ای گزینه اول به صورت زیر می‌باشد:

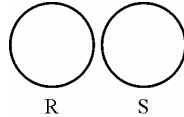
$$\Pi_{S\#}(\sigma_{city='PARIS'}(S))$$

مطابق جداول مذکور، خروجی به صورت زیر خواهد بود:

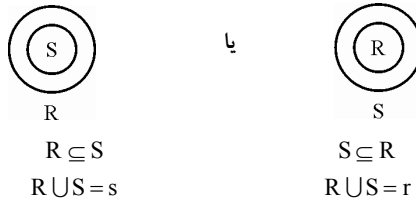
S#
S ₂
S ₃
S ₄

۱۳- گزینه (۳) صحیح است.

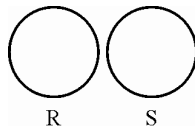
به طور کلی اجتماع دو رابطه با تیرهای یکسان (سازگار) به صورت زیر است:
برای دو رابطه $R(a,b)$ دارای r تاپل و رابطه $S(a,b)$ دارای s تاپل داریم:
اگر $R \cap S = \emptyset$ باشد، یعنی:



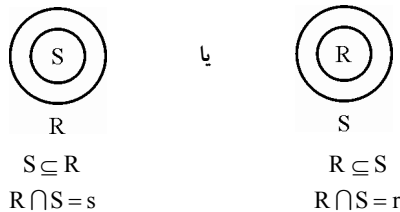
آنگاه، تعداد تاپل‌های $R \cup S$ ، برابر $r+s$ است که بیشترین مقدار ممکن خواهد بود.
اگر $R \subseteq S$ یا $S \subseteq R$ باشد، یعنی:



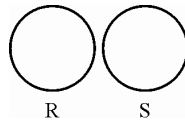
آنگاه، تعداد تاپل‌های $R \cup S$ ، برابر $\max(r,s)$ است که کمترین مقدار ممکن خواهد بود.
همچنین اشتراک دو رابطه با تیرهای یکسان (سازگار) به صورت زیر است:
برای دو رابطه $R(a,b)$ دارای r تاپل و رابطه $S(a,b)$ دارای s تاپل داریم:
اگر $R \cap S = \emptyset$ باشد، یعنی:



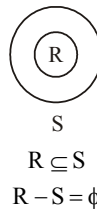
آنگاه، تعداد تاپل‌های $R \cap S$ ، برابر تهی است که کمترین مقدار ممکن خواهد بود.
اگر $R \subseteq S$ یا $S \subseteq R$ باشد، یعنی:



آنگاه، تعداد تاپل‌های $R \cap S$ ، برابر $\min(r,s)$ است که بیشترین مقدار ممکن خواهد بود.
همچنین تفاضل دو رابطه با تیرهای یکسان (سازگار) به صورت زیر است:
برای دو رابطه $R(a,b)$ دارای r تاپل و رابطه $S(a,b)$ دارای s تاپل داریم:
اگر $R \cap S = \emptyset$ باشد، یعنی:



آنگاه، تعداد تاپل‌های $R - S$ ، برابر Γ است که بیشترین مقدار ممکن خواهد بود. اگر $R \subseteq S$ باشد، یعنی:



آنگاه، تعداد تاپل‌های $R - S$ ، برابر تهی است که کمترین مقدار ممکن خواهد بود.

۱۴- گزینه (۴) صحیح است.

گزینه اول نادرست است، زیرا معادله‌های آن به صورت زیر است:

$$\sigma_{c-d}(R) \equiv \sigma_{c \wedge \sim d}(R) \equiv \sigma_c(\sigma_{\sim d}(R))$$

$$\sigma_{c \wedge d} \equiv \sigma_c(\sigma_d(R))$$

گزینه دوم نادرست است، زیرا معادله‌های آن به صورت زیر است:

$$\sigma_{c \wedge d}(R) \equiv \sigma_c(\sigma_d(R))$$

$$\sigma_c(R) - \sigma_d(R) \equiv \sigma_{c-d}(R) \equiv \sigma_c(\sigma_{\sim d}(R))$$

$$\sigma_c(\sigma_{\sim d}(R)) \equiv \sigma_{c \wedge \sim d}(R) \equiv \sigma_c(R) \cap \sigma_{\sim d}(R)$$

گزینه سوم نادرست است، زیرا معادله‌های آن به صورت زیر است:

$$\sigma_{c \vee d}(R) \equiv \sigma_c(R) \cup \sigma_d(R)$$

$$\sigma_{c \wedge d}(R) \equiv \sigma_c(R) \cap \sigma_d(R)$$

گزینه چهارم درست است، زیرا دو طرف با یکدیگر معادل هستند. در هر دو عبارت تاپل‌هایی از دو رابطه در خروجی ظاهر می‌شوند که شرط c در مورد آن‌ها صدق می‌کند.

۱۵- گزینه (۴) صحیح است.

اگر فرض کنیم منظور از join در این سؤال همان ضرب دکارتی است. بنابراین هر سه عملگر عنوان شده در گزینه‌های اول تا سوم جزء عملگرهای اصلی جبر رابطه‌ای بوده و وجودشان برای تمامی سیستم‌های رابطه‌ای و شبه رابطه‌ای ضروری است.

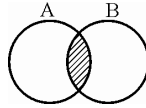
۱۶- گزینه (۴) صحیح است.

Times عملگر ضرب دکارتی و Minus عملگر تفاضل در جبر رابطه‌ای است. عملگرهای اجتماع، ضرب دکارتی و تفاضل جزء عملگرهای اصلی جبر رابطه‌ای هستند و عملگرهایی که با استفاده از عملگرهای

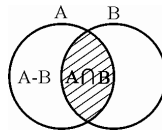
اصلی قابل ایجاد هستند، عملگرهای فرعی هستند، مانند عملگر اشتراک و تقسیم.

$$A \cap B = A - (A - B) = B - (B - A)$$

مثال:



با توجه به شکل زیر، مشاهده می‌شود که با کنار گذاشتن ناحیه $A - B$ از ناحیه A ، قسمت باقی مانده برابر $A \cap B$ است.



۱۷- گزینه (۱) صحیح است.

عبارت داده شده اشتراک دو رابطه را ایجاد می‌کند:

$$[A - (A - B)] \cup [B - (B - A)] = [A \cap B] \cup [A \cap B] = A \cap B$$

۱۸- گزینه (۲) صحیح است.

جدول زیر را در نظر بگیرید:

S#	Sname	S#	L#	Grade	L#	Lname
S ₁	S _{n1}	S ₁	L ₁	10	L ₁	L _{n1}
S ₂	S _{n2}	S ₁	L ₂	15	L ₂	L _{n2}
		S ₂	20	20		جدول درس (L)

جدول دانشجو (S)

جدول نمره (G)

از جداول فوق مشخص است که دانشجویی با شماره دانشجویی S₂ از درس L₂ نمره‌ای در جدول نمره ندارد. یعنی با توجه به پرس و جوی مطرح شده در صورت سؤال عبارت (S₂, L₂) پاسخ پرس و جو خواهد بود.

برای تولید عبارت فوق با استفاده از جبر رابطه‌ای پرواضح است که ابتدا باید جدول دانشجو و جدول درس را در هم ضرب دکارتی نمود تا حاصل آن برابر تمامی حالت‌های همه دانشجویها با همه درس‌ها شود. سپس تفاضل این حاصل را با جدول نمره محاسبه نمود. اما برای برقراری شروط سازگاری (تیتراهی یکسان) در عملگر تفاضل باید دستورات زیر را نوشت:

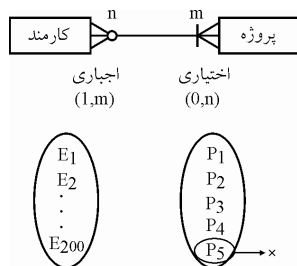
$$[\Pi_{S\#,L\#}(S \times L)] - [\Pi_{S\#,L\#}(G)]$$

عبارت فوق معادل عبارت زیر می‌باشد و پاسخ سؤال نیز می‌باشد:

$$(S \text{ TIMES } L) [S\#, L\#] \text{ MINUS } G[S\#, L\#]$$

۱۹- گزینه (۴) صحیح است.

بر اساس نمودار ER موجود در صورت سؤال اطلاعات زیر از آن قابل برداشت می‌باشد:



پروژه P_5 به دلیل اختیاری بودن موجودیت پروژه می‌تواند در ارتباط با نمونه موجودیت‌های کارمند شرکت نکند. اما همه کارمندان از E_1 تا E_{200} به دلیل اجباری بودن موجودیت کارمند باید در ارتباط با نمونه موجودیت‌های پروژه شرکت کنند.

با توجه به رابطه ذکر شده، یک پروژه می‌تواند هیچ کارمندی نداشته باشد، چون در رابطه می‌تواند شرکت نکند مانند P_5 و یا اگر یک پروژه در رابطه شرکت کرد می‌تواند حداکثر 200 کارمند داشته باشد. در طرف مقابل یک کارمند باید حداقل یک پروژه داشته باشد چون حضور کارمند اجباری است و حداکثر می‌تواند 5 پروژه داشته باشد.

برای یافتن حداکثر تعداد رکوردها در پیوند طبیعی این دو موجودیت باید حالتی را در نظر گرفت که همه کارمندان در اجرای همه پروژه‌ها مشارکت داشته‌اند. در این حالت تعداد سطرهای خروجی برابر $1000 = 200 \times 5$ خواهد بود. هم‌چنین برای یافتن حداقل تعداد رکوردها در پیوند طبیعی این دو موجودیت باید حالتی را در نظر گرفت که هر کارمند فقط در اجرای یک پروژه مشارکت داشته است. در این حالت تعداد سطرهای خروجی برابر با $200 = 200 \times 1$ خواهد بود.

۲۰- گزینه (۲) صحیح است.

پیوند طبیعی عملگری است که دو رابطه را بر مبنای یک یا چند صفت مشترک به هم پیوند می‌دهد. در پیوند طبیعی دو رابطه A و B یعنی $A \bowtie B$ اگر A و B صفت مشترکی نداشته باشند $A \bowtie B$ تبدیل به $A \times B$ خواهد شد و اگر روابط A و B تماماً دارای صفات یکسانی باشند، آن‌گاه $A \bowtie B$ معادل $A \cap B$ خواهد شد. مثال زیر گویای مطلب می‌باشد:

a	b	c	\bowtie	a	b	c	=	a	b	c
1	2	3		1	2	3		1	2	3
4	5	6		1	2	3		1	2	3

a	b	c	\cap	a	b	c	=	a	b	c
1	2	3		1	2	3		1	2	3
4	5	6		1	2	3		1	2	3

۲۱- گزینه (۴) صحیح است.

مطابق تعریف عملگر تقسیم در جبر رابطه‌ای فقط گزینه چهارم عملگر تقسیم را بیان می‌کند. توجه کنید که عبارت $\Pi_{R-S}(r)$ معادل r می‌باشد. یعنی عبارت زیر نیز عملگر تقسیم را تعریف می‌کند:

$$\Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times S) - r)$$

۲۲- گزینه (۳) صحیح است.

گزینه سوم نادرست است، زیرا، دارای خطای نحوی است. در گزینه سوم، در ابتدا توسط عملگر نیم پیوند، و شرط انتخاب سطر، مشخصات اساتیدی استخراج می‌گردد که متخصص DB هستند. دقت کنید که در عملگر نیم پیوند چپ، فقط ستون‌های جدول سمت چپ در خروجی قرار می‌گیرد، در اینجا ستون‌های

جدول prof در خروجی قرار می‌گیرد. حال چطور ممکن است، در پرانتز خارجی توسط عملگر پرتو، ستون depname استخراج گردد، در حالی که این ستون در جدول prof قرار ندارد. گزینه‌های اول، دوم و چهارم یک خروجی یکسان، اما با شیوه‌های متفاوت را تولید می‌کنند. به عبارت دیگر این گزینه‌ها نام استاد و نام دپارتمان اساتیدی که متخصص DB هستند را در خروجی نمایش می‌دهد.

۲۳- گزینه (۲) صحیح است.

طبق تعریف عملگر الحاق طبیعی در جبر رابطه‌ای ستون یا ستون‌های مشترک فقط یک بار در خروجی ظاهر می‌شود. در اینجا دو جدول دارای دو ستون مشترک یعنی A و C هستند، پس خروجی عبارت $T_1 \bowtie T_2$ دارای چهار ستون خواهد بود. هم‌چنین فقط سطرهایی از دو جدول در خروجی ظاهر می‌شوند که مقدارشان در ستون‌های مشترک مساوی باشد، بنابراین خروجی عبارت داده شده به صورت زیر خواهد بود. ملاحظه می‌شود که عبارت $T_1 \bowtie T_2$ دارای سه سطر است.

A	B	C	E
1	5	1	3
2	7	3	3
2	7	3	5

۲۴- گزینه (۱) صحیح است.

جداول زیر را در نظر بگیرید:

S#	Sname	...
S ₁	S _{n1}	
S ₂	S _{n2}	
S ₃	S _{n3}	
S ₄	S _{n4}	

S#	T#	...
S ₁	T ₁	
S ₁	T ₂	
S ₂	T ₁	
S ₂	T ₂	
S ₃	T ₁	

T#	Tname	...
T ₁	T _{n1}	
T ₂	T _{n2}	

جدول آزمون (T) جدول شرکت در آزمون (ST) جدول داوطلب (S)

مطابق پرس و جوی مطرح شده در گزینه اول داریم: ابتدا در پرانتز داخلی توسط عملگر تقسیم، شماره داوطلبانی که در همه آزمون‌ها شرکت کرده‌اند، استخراج می‌گردد، به صورت زیر:

$$\frac{S\# \mid T\#}{S_1 \mid T_1} \div \frac{T\#}{T_1} = \frac{S\#}{S_1}$$

$$\frac{S\# \mid T\#}{S_1 \mid T_2} \div \frac{T\#}{T_2} = \frac{S\#}{S_2}$$

$$\frac{S\# \mid T\#}{S_2 \mid T_1} \div \frac{T\#}{T_1} = \frac{S\#}{S_2}$$

$$\frac{S\# \mid T\#}{S_2 \mid T_2} \div \frac{T\#}{T_2} = \frac{S\#}{S_2}$$

$$\frac{S\# \mid T\#}{S_3 \mid T_1} \div \frac{T\#}{T_1} = \frac{S\#}{S_3}$$

سپس خروجی فوق با جدول S الحاق طبیعی می‌گردد و در نهایت مشخصات داوطلبانی که در همه آزمون‌ها شرکت کرده‌اند، در خروجی قرار می‌گیرد. به صورت زیر:

S#	Sname	...
S ₁	S _{n1}	
S ₂	S _{n2}	

گزینه دوم نادرست است.

مطابق پرس و جوی مطرح شده در گزینه دوم داریم: $(\Pi_{S\#,T\#}(ST) \div \Pi_{T\#}(T)) \times S$
ابتدا در پراتنز داخلی توسط عملگر تقسیم، شماره داوطلبانی که در همه آزمون‌ها شرکت کرده‌اند، استخراج می‌گردد، به صورت زیر:

S#	T#	÷	T#	=	S#
S ₁	T ₁		T ₁		S ₁
S ₁	T ₂		T ₂		S ₂
S ₂	T ₁				
S ₂	T ₂				
S ₃	T ₁				

سپس خروجی فوق با جدول S، نیم پیوند چپ می‌گردد و در نهایت مطابق عملگر نیم پیوند چپ فقط ستون‌های جدول سمت چپ در خروجی قرار می‌گیرد. یعنی فقط شماره داوطلبانی که در همه آزمون‌ها شرکت کرده‌اند، به عنوان خروجی نمایش داده می‌شود. به صورت زیر:

S#	×	S#	Sname	...	S#
S ₁		S ₁	S _{n1}		S ₁
S ₂		S ₂	S _{n2}		S ₂
		S ₃	S _{n3}		
		S ₄	S _{n4}		

در حالی که در صورت سؤال، مشخصات داوطلبانی که در همه آزمون‌ها شرکت کرده‌اند، مورد پرس و جو قرار گرفته است.

گزینه‌های سوم و چهارم نادرست هستند. زیرا دارای خطای نحوی هستند. مطابق قوانین عملگر تقسیم در جبر رابطه‌ای، ستون‌های مقسوم‌علیه، باید زیرمجموعه ستون‌های مقسوم باشد. اما همانطور که در این دو گزینه مشاهده می‌شود، ستون Tname در مقسوم‌علیه، زیرمجموعه مقسوم نیست.

۲۵- گزینه (۲) صحیح است.

جدول زیر را در نظر بگیرید:

S#	Sname	S#	L#	Term#	Grade
S ₁	S _{n1}	S ₁	L ₁	92	20
S ₂	S _{n2}	S ₁	L ₂	91	20
S ₃	S _{n3}	S ₁	L ₃	90	20
		S ₂	L ₁	92	20
		S ₂	L ₂	90	18
		S ₃	L ₁	92	19

SL[S#]Minus(SL where grade <> 20)[S#]

جدول SL

جبر رابطه‌ای عبارت فوق به صورت مقابل است: $\Pi_{S\#}(SL) - \Pi_{S\#}(\sigma_{\text{grade} \neq 20}(SL))$
بخش اول عبارت فوق، شماره دانشجویانی را می‌دهد که تاکنون حداقل یک نمره برای آن‌ها ثبت شده است.

و بخش دوم عبارت فوق، شماره دانشجویانی را می‌دهد که حداقل یک نمره غیر بیست برای آن‌ها ثبت شده است.

بنابراین حاصل تفاضل این دو بخش شماره دانشجویانی را می‌دهد که هیچ نمره غیر بیستی ندارند یا به عبارت دیگر شماره دانشجویانی را می‌دهد که تمام نمرات آن‌ها بیست است.

$$\begin{array}{r} \underline{S\#} \quad \underline{S\#} \quad \underline{S\#} \\ S_1 - S_2 = S_1 \\ S_2 \quad S_3 \\ S_3 \end{array}$$

۲۶- گزینه (۲) صحیح است.

گزینه‌ی اول اشتباه است چون در رابطه‌ی تقسیم باید رابطه‌ی سمت چپ تقسیم شامل تمام ستون‌های رابطه‌ی سمت راست تقسیم باشد. به عبارت دیگر صفات مقسوم‌علیه زیر مجموعه صفات مقسوم باشد. گزینه‌ی سوم اشتباه است چون در رابطه اشتراک باید ستون‌های متناظر یکسان باشد. (تپترهای مشابه)

۲۷- گزینه (۳) صحیح است.

عملگر \bowtie ، به طور کلی، سطرهایی از جداول را در خروجی می‌آورد که صفات‌های همنام آن‌ها مقدار یکسانی داشته باشند. به علاوه، ستون‌های همنام فقط یک بار در خروجی ظاهر می‌شوند. بنابراین عبارت $A \bowtie B$ ، سطرهایی از دو جدول A و B که مقادیر تمام صفات‌هایشان با هم برابر باشد را بر می‌گرداند. پس خروجی، همان اشتراک A و B است.

مطابق یک قاعده کلی اگر تمامی صفات دو رابطه A و B یکسان باشند، آن‌گاه $A \bowtie B = A \cap B$. هم‌چنین اگر هیچ صفت مشترکی بین دو رابطه A و B نباشند، آن‌گاه $A \bowtie B = A \times B$. بنابر آن‌چه در صورت سؤال مطرح شده است، مبنی بر یکسان بودن صفات دو رابطه A و B پس می‌توان نتیجه گرفت که گزینه سوم درست و گزینه دوم نادرست است. البته به این مسأله نیز توجه کنید که خروجی عبارت $A \bowtie B$ با $A \times B$ در گزینه اول و همچنین با $A \times B$ در گزینه چهارم به دلیل یکسان بودن تمام صفات A و B برابر است! که طراح محترم این مسأله را لحاظ نکرده‌اند و گزینه سوم را به عنوان پاسخ درست اعلام کرده‌اند!

مثال‌های زیر گویای مطلب است:

$$\begin{array}{ccc} \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline \end{array} & \bowtie & \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline \end{array} \\ \\ \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline \end{array} & \times & \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline \end{array} \\ \\ \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline \end{array} & \bowtie & \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline \end{array} \\ \\ \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline \end{array} & \cap & \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline \end{array} \end{array}$$

۲۸- گزینه (۲) صحیح است.

ابتدا خروجی دستور $T1 \times T2$ را تعیین می‌کنیم: با توجه به شرط $T1.B = T2.B$ ، باید ببینیم.

$$T1.B = T2.B$$

هر سطر از جدول $T1$ با چند سطر از جدول $T2$ تطابق دارد. (منظور از «تطابق» برابری مقدار فیلد B در دو جدول است). سطر اول از $T1$ با سطرهای اول و دوم از $T2$ تطابق دارد. پس به ازای آن‌ها، دو سطر در خروجی تولید می‌شود. سطر دوم از $T1$ با سطرهای سوم، چهارم و پنجم از $T2$ تطابق دارد. پس به ازای آن‌ها سه سطر در خروجی تولید می‌شود. سطر سوم از $T1$ با هیچ سطری از $T2$ تطابق ندارد و به ازای آن هیچ سطری در خروجی تولید نمی‌شود. با توجه به این توضیحات، خروجی دستور $T1 \times T2$ به شکل زیر است:

$$T1.B = T2.B$$

A	T1.B	T2.B	C
a1	b1	b1	c1
a1	b1	b1	c2
a2	b2	b2	c2
a2	b2	b2	c3
a2	b2	b2	c4

حال باید حاصل انجام $\Pi_{T1.B}$ روی خروجی فوق را تعیین کنیم. واضح است که این دستور ستون $T1.B$ از خروجی فوق را استخراج می‌کند. تنها نکته‌ای که باید به آن توجه کنیم این است که در خروجی عملگر Π ، سطرهای تکراری وجود ندارد، بنابراین خروجی نهایی به صورت زیر است:

T1.B
b1
b2

پس خروجی دستور داده شده، فقط دو سطر دارد.

۲۹- گزینه (۳) صحیح است.

Query1 را در نظر بگیرید. عبارت $R(S)$ سطرهایی از رابطه‌های R و S که مقدار فیلد b آن‌ها یکسان است را با هم پیوند می‌دهد. سپس عملگر پرتوی $\Pi_{a,b}$ فقط ستون‌های a و b (که مربوط به رابطه‌ی R هستند) را انتخاب می‌کند. پس می‌توان خروجی نهایی را این‌گونه توصیف کرد:
Query1 سطرهایی از رابطه‌ی R را بر می‌گرداند که مقدار صفت b آن‌ها با مقدار صفت b در یکی از سطرهای رابطه‌ی S برابر باشد.

حال Query2 را در نظر بگیرید. دستور $\Pi_C(s)$ ستون C از رابطه‌ی S را انتخاب می‌کند. سپس دستور نامگذاری $\rho_t(a)$ ، جدول خروجی دستور $\Pi_C(s)$ را برابر t و نام تنها ستون آن را برابر a قرار می‌دهد. (این تغییر نام برای فراهم سازی امکان اجرای عملگر اشتراک انجام شده است، زیرا برای انجام اشتراک باید صفات دو رابطه یکسان باشند). دستور $\Pi_b(s)$ ستون b از رابطه‌ی S را انتخاب می‌کند. خروجی این دستور در خروجی دستور قبلی ضرب دکارتی می‌شود و جدولی با شمای زیر حاصل می‌شود.

$$\rho_t(a)(\Pi_C(s)) \times \Pi_b(s)$$

S.c	S.b

 \rightarrow

a	b

سطرهای جدول فوق، شامل هر یک از مقادیر مختلف ستون c از رابطه‌ی S است که در کنار هر یک از مقادیر مختلف ستون b از رابطه‌ی S قرار گرفته باشد. به بیان دیگر براساس محتویات رابطه S همه ترکیبات مختلف برای مقادیر دو صفت b و c در جدول فوق قرار دارد.

در نهایت، اشتراک جدول فوق با جدول R محاسبه می‌شود. می‌دانیم که در خروجی اشتراک، سطرهایی از دو جدول ظاهر می‌شوند که مقدار صفت‌های آن‌ها به ترتیب با هم برابر باشد. یعنی خروجی نهایی شامل سطرهایی از R است که صفت a از آن با صفت S.c (که S.c به a تغییر نام داده است) و صفت b از آن با صفت S.b از جدول فوق برابر باشد.

اگر خروجی Query2 را با خروجی Query1 مقایسه کنید می‌بیند که در خروجی Query1 سطرهایی از R ظاهر می‌شوند که مقدار صفت b آن‌ها در S.b وجود داشته باشد اما در خروجی Query2 علاوه بر شرط تساوی مقادیر صفت‌های b از R و S، شرط تساوی صفت a از R با صفت c از S (که c به a تغییر نام داده است) نیز در نظر گرفته شده است. پس خروجی Query2 محدودتر از Query1 است و زیر مجموعه‌ی آن می‌باشد.

مثال: دو جدول $R(a,b)$ و $S(b,c)$ را با مقادیر زیر در نظر بگیرید:

a	b
1	2
3	4
6	7

جدول R

b	c
2	1
4	5

جدول S

Query1: $\Pi_{a,b}(R \bowtie S)$

$$\text{Query1: } \Pi_{a,b} \left(\begin{array}{c|c} a & b \\ \hline 1 & 2 \\ 3 & 4 \\ 6 & 7 \end{array} \bowtie \begin{array}{c|c} b & c \\ \hline 2 & 1 \\ 4 & 5 \end{array} \right) = \begin{array}{c|c} a & b \\ \hline 1 & 2 \\ 3 & 4 \end{array}$$

Query2: $R \cap (\rho_{t(a)}(\Pi_c(S)) \times \Pi_b(S))$

$$\begin{array}{c} \downarrow \quad \downarrow \\ \left(\begin{array}{c|c} a & b \\ \hline 1 & 2 \\ 5 & 4 \end{array} \right) \\ \downarrow \\ \begin{array}{c|c} a & b \\ \hline 1 & 2 \\ 3 & 4 \\ 6 & 7 \end{array} \cap \begin{array}{c|c} a & b \\ \hline 1 & 4 \\ 5 & 2 \\ 5 & 4 \end{array} = \begin{array}{c|c} a & b \\ \hline 1 & 2 \end{array} \end{array}$$

۳۰- گزینه (۱) صحیح است.

قبل از هر چیز توجه کنید که کلمه‌ی «همه» در عبارت داده شده زاید است و بنابراین نیازی به استفاده از عملگر \neq نیست.

برای نوشتن عبارت جبر رابطه‌ای مربوطه، مراحل زیر را به ترتیب انجام می‌دهیم:

(۱) از جدول قطعه (P) سطرهای مربوط به قطعات آبی رنگ را به دست می‌آوریم: $\sigma_{\text{color}="Blue"}(P)$

(۲) از خروجی مرحله‌ی قبل، فقط ستون «شماره قطعه» ($P\#$) را انتخاب می‌کنیم:

$$\Pi_{P\#}(\sigma_{\text{color}="Blue"}(P))$$

با اجرای دستور فوق، شماره‌ی قطعات آبی رنگ به دست می‌آید.

(۳) خروجی مرحله‌ی ۲ را با جدول محموله (SP) پیوند طبیعی می‌زنیم تا سطرهایی از جدول SP که $P\#$ آن‌ها برابر یکی از مقادیر خروجی مرحله‌ی ۲ است را به دست آوریم. به عبارت دیگر، با این کار، سطرهایی از جدول SP که متناظر با قطعات آبی رنگ هستند به دست می‌آیند:

$\Pi_{P\#}(\sigma_{\text{color}="Blue"}(P)) \bowtie SP$

(۴) از خروجی مرحله‌ی قبل، فقط ستون «شماره‌ی تهیه‌کننده» ($S\#$) را انتخاب می‌کنیم:

$$\Pi_{S\#}(\Pi_{P\#}(\sigma_{\text{color}="Blue"}(P)) \bowtie SP)$$

با اجرای دستور فوق، شماره‌ی تهیه‌کننده‌هایی که قطعه‌ی آبی رنگ تولید می‌کنند به دست می‌آید.

(۵) خروجی مرحله‌ی قبل را با جدول تهیه‌کننده (S) پیوند طبیعی می‌زنیم تا سطرهایی از جدول S که متناظر با تهیه‌کننده‌های قطعه‌های آبی رنگ هستند به دست آید:

$$\Pi_{S\#}(\Pi_{P\#}(\sigma_{\text{color}="Blue"}(P)) \bowtie SP) \bowtie S$$

(۶) از خروجی مرحله‌ی قبل، فقط ستون «اسم تهیه‌کننده» (Sname) را انتخاب می‌کنیم:

$$\Pi_{\text{Sname}}(\Pi_{S\#}(\Pi_{P\#}(\sigma_{\text{color}="Blue"}(P)) \bowtie SP) \bowtie S)$$

۳۱- گزینه (۲) صحیح است.

گزینه‌ی اول عبارت درستی است. در جدول حاصل از انجام A-B، کلید اصلی A به عنوان کلید اصلی در نظر گرفته می‌شود.

گزینه‌ی دوم عبارت نادرستی است: در \bowtie نیازی نیست شروط سازگاری (تیرهای مشابه) در دو رابطه برقرار باشد.

گزینه‌ی سوم عبارت درستی است: عملگرهای جبر رابطه‌ای به دو دسته‌ی اصلی و فرعی تقسیم می‌شوند. عملگرهای اصلی عبارتند از گزینش (σ)، پرتو (Π)، اجتماع (\cup)، تفاضل ($-$)، ضرب دکارتی (\times) و جایگزینی (\leftarrow). عملگرهای فرعی را می‌توان از روی عملگرهای اصلی به دست آورد. برای مثال، اشتراک و پیوند طبیعی را می‌توان به صورت زیر به دست آورد:

$$A \cap B = A - (A - B)$$

$$A \bowtie B = A \times B$$

شرط تساوی

ستون‌های همنام

هم‌چنین اگر صفات مشترک دو رابطه‌ی A و B را Y و بقیه‌ی صفات رابطه‌ی A را X بنامیم آن‌گاه $A \div B$ را می‌توان به صورت مقابل نوشت:

$$\Pi_X(A) - \Pi_X(\Pi_X(A) \times \Pi_Y(B) - A)$$

گزینه‌ی چهارم عبارت (تقریباً) درستی است: اگر رابطه‌ای n صفت داشته باشد با عملگر پرتو می‌توان $(2^n - 1)$ تصویر مختلف از آن به دست آورد: یکی از n ستون را انتخاب کنیم یا دو تا از n ستون را

انتخاب کنیم یا یا تمام n ستون را انتخاب کنیم، که مجموع تعداد این انتخاب‌ها برابر است با:

$$\binom{n}{1} + \binom{n}{2} + \binom{n}{3} + \dots + \binom{n}{n} = 2^n - 1$$

۳۲- گزینه (۳) صحیح است.

چون از full outer join استفاده شده پس تمام سطرهای دو جدول در خروجی ظاهر می‌شوند و سطرهایی که امکان Natural Join آن‌ها وجود دارد (یعنی ستون‌های هم نامشان، یعنی A ، مقدار یکسان دارند) با هم قرار می‌گیرند. پس خروجی به صورت زیر است:

A	B	C	D
1	2	2	3
1	2	3	4
3	4	NULL	NULL
2	NULL	4	5

۳۳- گزینه (۱) صحیح است.

پیوند طبیعی (\bowtie) سطرهایی از دو جدول که ستون‌های هم نامشان مقادیر یکسانی دارند را با هم ترکیب می‌کند (ستون‌های هم نام یک بار در خروجی می‌آیند). بنابراین حاصل پیوند طبیعی به صورت زیر است:

T_1		T_2		$T_1 \bowtie T_2$		
A	B	B	C	A	B	C
a1	b1	b1	c1	a1	b1	c1
a2	b2	b1	c2	a1	b1	c2
a3	b3	b2	c2	a2	b2	c2
		b2	c3	a2	b2	c3
		b2	c4	a2	b2	c4

بنابراین با انجام عملگر پرتو (Π_B) فقط ستون B انتخاب و در خروجی ظاهر می‌شود:

B
b1
b2

توجه کنید که چون خروجی عملگر Π یک مجموعه است و در مجموعه‌ها عضو تکراری معنی ندارد، سطرهای تکراری در خروجی وجود نخواهد داشت. بنابراین فقط دو سطر در خروجی وجود دارد.

۳۴- گزینه (۳) صحیح است.

گزینه اول اشتباه است. برای اثبات این موضوع، فرض کنید مقادیر زیر در جدول‌های R و S وجود داشته باشند (حروف بزرگ نام ستون‌ها و حروف کوچک مقدار فیلدها هستند).

A	B	C	D
a	b	c	d

جدول R جدول S

بنابراین $S \bowtie R$ به صورت زیر خواهد بود (توجه کنید که چون R و S ستون همنامی ندارند پیوند طبیعی معادل ضرب دکارتی خواهد بود):

A	B	C	D
a	b	c	d

بنابراین خروجی $\Pi_{A,B,D}(R \bowtie S)$ به صورت زیر است:

A	B	D
a	b	d

اما خروجی $\rho_{T(B,D)}(S)$ به صورت زیر است:

B	D
c	d

و بنابراین حاصل $R \bowtie \rho_{T(B,D)}(S)$ هیچ سطری در بر ندارد! (چون ستون‌های همنام در جدول R و جدول قبلی مقادیر مشترکی ندارند و بنابراین پیوند طبیعی \bowtie هیچ سطری بر نمی‌گرداند) گزینه دوم اشتباه است. سمت چپ تساوی، حاصل $R-S$ را بر می‌گرداند در صورتی که سمت راست تساوی حاصل $S-R$ را به ما می‌دهد.

گزینه سوم صحیح است. با انجام ضرب دکارتی $R \times S$ ، هر سطر از R در کنار تمام سطرها S قرار می‌گیرد. اما با انجام $\Pi_{A,B}$ فقط ستون‌های A و B (ستون‌های R) در خروجی می‌آیند. توجه کنید که در خروجی Π ، سطر تکراری وجود ندارد، بنابراین خروجی به دست آمده معادل R می‌باشد.

۳۵- گزینه (۱) صحیح است.

برای پاسخگویی به این سؤال باید به صورت زیر از تفریق استفاده کنیم:

{تهیه کنندگانی که قطعه آبی تولید می‌کنند} - {تمام تهیه کنندگان}

و سپس با اعمال Π_{name} ، فقط نام تهیه کنندگان را انتخاب کنیم:

$\Pi_{\text{name}}(\text{تهیه کنندگانی که قطعه آبی تولید می‌کنند} - \text{تمام تهیه کنندگان})$

مجموعه «تمام تهیه کنندگان» همان جدول S است. برای به دست آوردن مجموعه «تهیه کنندگانی که قطعه آبی تولید می‌کنند» به صورت زیر عمل می‌کنیم: از جدول قطعه (P) ، سطرها P را انتخاب می‌کنیم:

$\sigma_{\text{Color}='Blue'}(P)$

$\Pi_P(\sigma_{\text{Color}='Blue'}(P))$

و سپس، از حاصل آن، فقط شماره قطعه $(P\#)$ را بر می‌داریم:

سپس باید در جدول محموله‌ها (SP) ، محموله‌هایی را بیابیم که مربوط به شماره قطعه‌های فوق باشند. برای این منظور، جدول SP را با نتیجه قبلی، پیوند طبیعی می‌کنیم:

$\Pi_P(\sigma_{\text{Color}='Blue'}(P)) \bowtie SP$

سپس از حاصل فوق، فقط شماره تهیه‌کننده $(S\#)$ را بر می‌داریم:

$\Pi_{S\#}(\Pi_P(\sigma_{\text{Color}='Blue'}(P)) \bowtie SP)$

حاصل عبارت فوق، شماره تهیه‌کنندگانی است که قطعه آبی تولید می‌کنند. برای به دست آوردن نام این تهیه‌کنندگان، باید عبارت فوق را به جدول تهیه‌کنندگان (S) پیوند طبیعی کنیم:

$\Pi_{S\#}(\Pi_P(\sigma_{\text{Color}='Blue'}(P)) \bowtie SP) \bowtie S$

۳۶- گزینه (۱ و ۴) صحیح است.

توجه کنید که گزینه‌های اول و چهارم و همچنین گزینه‌های دوم و سوم یکسان هستند!

برای پاسخ به این سؤال باید جدول Person و Parent (یا زیر مجموعه‌هایی از آنها) روی فیلد Pid با هم پیوند زده شوند. این کار در گزینه‌های اول و چهارم با عملگر \bowtie انجام شده است (توجه کنید که عملگر \bowtie دو جدول را روی فیلدهای هم‌نامشان پیوند می‌زند و Pid تنها فیلد هم‌نام این دو جدول است). شرط‌های "مرد" =sex و "آرش کمانگیر" =Pname در عملگرهای σ باعث انتخاب سطرهای لازم می‌شوند. در نهایت عملگر Π ppid ستون شناسه پدر را بر می‌گرداند.

توجه کنید که در گزینه‌های دوم و سوم عملگر \bowtie باعث می‌شود فقط ستون‌های جدول سمت چپ پیوند (یعنی ستون‌های Person) در خروجی پیوند ظاهر شود. از آنجا که در جدول Person ستون Ppid وجود ندارد، عملگر Π ppid به دلیل وجود خطای نحوی، قابل انجام نخواهد بود! پس این دو گزینه اشتباه هستند.

۳۷- گزینه (۲) صحیح است.

برای به دست آوردن تمام اجداد یک نفر تا هر تعداد نسل قبل، لازم است جدول Person به طور بازگشتی با جدول Parent پیوند زده شود. اما query بازگشتی در جبر رابطه‌ای جایی ندارد. بنابراین گزینه دوم در جبر رابطه‌ای قابل پاسخ‌گویی نیست.

جدول Person و Parent را با مقادیر زیر در نظر بگیرید:

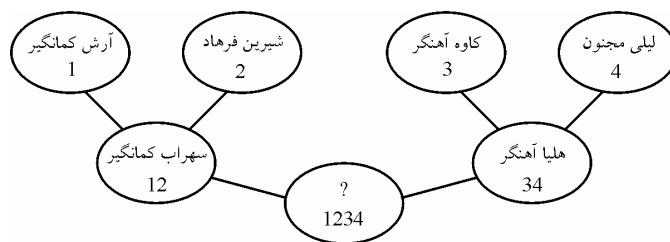
pid	pname	age	DOB	pid	ppid	Sex
1	آرش کمانگیر	93	1300	12	1	مرد
2	شیرین فرهاد	91	1302	12	2	زن
12	سهراب کمانگیر	63	1330	34	3	مرد
3	کاوه آهنگر	53	1340	34	4	زن
4	لیلی مجنون	43	1350			
34	هلیا آهنگر	28	1365			

جدول Parent

جدول Person

توجه: فرض کنید در سال ۱۳۹۳ قرار داریم.

درخت زیر وجود دارد:



توجه: واضح است که نام خانوادگی شماره 1234 پس از تولد "کمانگیر" خواهد بود.

برای پاسخ به گزینه اول با پرس و جوی «همسر فردی به نام "آرش کمانگیر"» روال زیر را باید دنبال کرد: برای این کار ابتدا باید pid آرش کمانگیر را از جدول person به صورت زیر شناسایی نمود.

$$\Pi_{pid}(\sigma_{pname = \text{"آرش کمانگیر"}}(\text{person})) \rightarrow \frac{pid}{1}$$

سپس از جدول Parent فردی که یکی از والدین او "آرش کمانگیر" است را شناسایی نمود.

$$\Pi_{pid}(\sigma_{ppid=1}(\text{parent})) \rightarrow \frac{pid}{12}$$

سپس از جدول Parent، ppid یکی دیگر از والدین فردی با pid برابر مقدار 12 و جنسیت برابر زن را می‌توان شناسایی نمود.

$$\Pi_{ppid}(\sigma_{pid=12 \wedge sex="زن"}(\text{parent})) \rightarrow \frac{ppid}{2}$$

سپس از جدول person نام فردی با pid برابر مقدار 2 را می‌توان شناسایی نمود.

$$\Pi_{pname}(\sigma_{pid=2}(\text{person})) \rightarrow \frac{pname}{\text{شیرین فرهاد}}$$

در واقع شیرین فرهاد، همسر آرش کمانگیر است.

برای پاسخ به گزینه سوم با پرس و جوی «افرادی که همسر آنها بالای 90 سال دارند» روال زیر را باید دنبال کرد:

ابتدا باید همسر هر فردی را مطابق آنچه پیش از این نیز گفتیم شناسایی کنید. مثلاً «شیرین فرهاد» همسر «آرش کمانگیر» است، سپس بررسی می‌شود که سن «شیرین فرهاد» آیا بالای 90 سال است، اگر شرط برقرار بود، «آرش کمانگیر» فردی است که همسر آن بالای 90 سال است و در خروجی نمایش داده می‌شود. این روال باید برای تک تک اعضای جدول person انجام گردد. اگر فرض کنیم که پدر و مادری که دارای فرزند هستند، هر دو در قید حیات و با هم در حال زندگی نیز هستند، می‌توان نتیجه‌های زیر را نیز از جدول person و parent استخراج نمود:

$$\Pi_{ppid}(\text{parent}) \rightarrow \frac{ppid}{\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix}}$$

شماره شناسایی افراد دارای فرزند (پدرها و مادرها)

شماره شناسایی افراد بدون فرزند:

شماره شناسایی افراد دارای فرزند - کل افراد = شماره شناسایی افراد بدون فرزند

$$\text{شماره شناسایی افراد بدون فرزند} = [\Pi_{pid}(\text{person})] \text{Minus} [\rho_T(pid)(\Pi_{ppid}(\text{parent}))]$$

pid	pid	pid
1	1	
2	2	12
12	= 3	- 34
3	4	
4		
34		

توجه: تغییر نام توسط عملگر ρ، برای برقراری شروط سازگاری (تیتراهای مشابه) در عمل تفاضل انجام گرفته است.

گزینه چهارم به راحتی با کسر کردن تاریخ تولد هر فرد از تاریخ روز و مقایسه حاصل با سن آن فرد قابل پاسخ‌گویی است. توجه داشته باشید که صفت سن چون پویا و در حال تغییر است (مشتق) اغلب در پایگاه داده ذخیره نمی‌شود.

۳۸- گزینه (۳) صحیح است.

گزینه اول اشتباه است: دو رابطه فرضی $R_1(A, B)$ و $R_2(A, B)$ و اجتماع آن‌ها واضح است که A_3 (که اجتماع کلیدهای R_1 و R_2 است) کلید نیست.

R_1	
A	B
2	a
2	b

R_2	
A	B
1	x
2	y

R_3	
A	B
1	a
1	x
2	b
2	y

گزینه دوم اشتباه است: دو رابطه فرضی $R_1(A, B, C, D, E)$ و $R_2(D, E)$ را در نظر بگیرید. حاصل تقسیم $R_1 \div R_2$ یک رابطه است که ستون‌های آن شامل ستون‌هایی از R_1 است که در R_2 نیست:

$R_1 \div R_2 = R_3(A, B, C)$ در جدول R_3 ، ستون‌های A و B با هم کلید اصلی‌اند (با اضافه کردن صفت C به صفات A و B ، مجموعه $\{A, B, C\}$ هر چند یک کلید است ولی کلید کمینه نیست). کلید اصلی از میان کلیدهای کاندید انتخاب می‌شود و کلید کاندید حتماً کمینه است.

گزینه سوم صحیح است: اشتراک دو رابطه، شامل رکوردهایی است که در هر دو رابطه بوده‌اند و بنابراین وابستگی‌های تابعی در هر دو رابطه را رعایت کرده‌اند. بنابراین هر یک از کلیدهای کاندید R_1 و R_2 باز هم یک کلید کاندید در R_3 هستند و می‌توانند به عنوان کلید اصلی در نظر گرفته شوند.

گزینه چهارم اشتباه است: دو رابطه فرضی $R_1(A, B)$ و $R_2(C, D)$ را در نظر بگیرید. در حاصل ضرب دکارتی آن دو، یعنی $R_1 \times R_2 = R_3(A, B, C, D)$ با توجه به وابستگی‌های تابعی موجود، مجموعه صفات $\{A, C\}$ کلید هستند. پس نیازی نیست تمام صفات را روی هم به عنوان کلید اصلی گرفت.

۳۹- گزینه (۳) صحیح است.

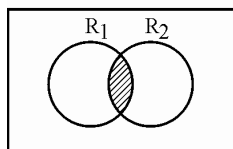
گزینه‌های اول و دوم روابط معروف بین عملگرهای \cap و $-$ هستند. گزینه چهارم هم تعریف عملگر Semiminus است.

گزینه سوم اشتباه است. ابتدا توجه کنید که $(R_1 - R_2)$ و $(R_2 - R_1)$ هیچ اشتراکی با هم ندارند. پس:

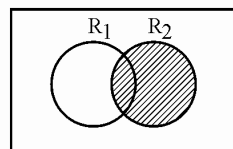
$$(R_1 - R_2) - (R_2 - R_1) = (R_1 - R_2)$$

حال با توجه به عدم تساوی قسمت‌های هاشور خورده در نمودارهای ون زیر معلوم می‌شود تساوی گزینه سوم برقرار نیست:

$$R_1 \cap R_2$$



$$(R_1 \cup R_2) - (R_1 - R_2) = R_2$$



۴۰- گزینه (۱) صحیح است.

عبارت STU Times CRS جدول دانشجو را با جدول دروس ضرب دکارتی می‌کند. حاصل ضرب این ضرب دکارتی، جدولی است که در آن، هر دانشجو با تمام دروس (چه آن درس را گرفته باشد چه نگرفته باشد) ظاهر شده است. سپس عبارت [STID, CID] عمل پرتو را انجام می‌دهد و فقط دو ستون STID و CID را باقی می‌گذارد (برای این که عملگر MINUS امکان‌پذیر شود). در جدول SC، معلوم است هر دانشجو چه درسی را گرفته است. عملگر MINUS باعث می‌شود از جدول حاصل ضرب دکارتی، فقط رکوردهای هر دانشجو با دروسی که نگرفته است باقی بماند.

۴۱- گزینه (۱) صحیح است.

عبارت $\sigma_{pRel=1}(Rel)$ مجموعه سطرهایی از جدول Rel را می‌دهد که یک رابطه پدر-فرزندی را دارند. سپس با اعمال عملگر Π ، فقط ستون pID1 (یعنی شناسه فرزندان) از مجموعه فوق انتخاب می‌شود. یعنی کل عبارت سمت چپ π_{pID1} ، مجموعه شناسه‌های فرزندان را به ما می‌دهد. این مجموعه با جدول Person با شرط $pID1 = pid$ پیوند می‌شود، پس، از جدول Person سطرهایی را به دست می‌آورد که شناسه آن‌ها (pID آن‌ها) برابر pID1 ها (یعنی شناسه‌های فرزندان) باشد. پس در نهایت سطرهایی از جدول Person به دست می‌آید که مربوط به فرزندان است.

۴۲- گزینه (۱) صحیح است.

عمل اتصال طبیعی روی دو جدول معادل است با انجام ضرب دکارتی روی آن دو جدول و سپس انجام عمل select با شرط برابری مقدار ستون‌های همنام دو جدول. بنابراین گزینه اول صحیح است. گزینه دوم اشتباه است، چون با انجام پیوند طبیعی بخشی از اطلاعات جداول حذف می‌شوند. در واقع، رکوردهایی از هر یک از جداول که رکورد متناظری در جدول دیگر ندارند حذف می‌شوند. گزینه سوم نیز اشتباه است، چون با توجه به توضیحات فوق برای گزینه دوم، برخی از اطلاعات از دست می‌روند.

توجه: منظور از عمل باینری یا دودویی، یک عملگر با دو عملوند است.

۴۳- گزینه (۱) صحیح است.

دستور select در جبر رابطه‌ای یک شرط (یعنی یک یا چند عمل مقایسه) را روی سطرهای جدول بررسی می‌کند. هرگاه مقدار داده‌ای در یک عمل مقایسه برابر NULL باشد نتیجه مقایسه برابر UNKNOWN خواهد شد (یعنی TURE نمی‌شود) و در نتیجه، سطر مربوطه در خروجی ظاهر نمی‌شود. اما دستور projection (که ستون‌هایی از یک جدول را باقی می‌گذارد) با مقادیر داخل جدول کاری ندارد و بنابراین با NULL نیز مانند سایر مقادیر برخورد می‌کند. هم‌چنین دستور UNION هم اگر مقدار NULL را در یکی از دو جدول ببیند، این مقدار را (همانند مقادیر دیگر) در خروجی می‌آورد. پس گزینه‌های دوم و سوم عبارات درستی‌اند.

۴۴- گزینه (۳) صحیح است.

برای درک بهتر، جدول course، با مقادیر زیر را در نظر بگیرید:

course (Cname, TA)	
Cname	TA
C ₁	T ₁
C ₁	T ₂
C ₂	T ₁
C ₃	T ₃

که براساس $\rho_{TA_1 \rightarrow TA_2}(\text{course})$ و $\rho_{TA_2 \rightarrow TA_1}(\text{course})$ الحاق دو جدول زیر را فقط بر روی ستون مشترک Cname به شکل زیر خواهیم داشت:

Cname	TA1	Cname	TA2	Cname	TA1	TA2
C1	T1	C1	T1	C1	T1	T1
C1	T2	C1	T2	C1	T1	T2
C2	T1	C2	T1	C1	T2	T1
C3	T3	C3	T3	C1	T2	T2
				C2	T1	T1
				C3	T3	T3

در مسأله مطرح شده است که می‌خواهیم همه زوج‌های (TA1 و TA2) را به دست آوریم که TA1 و TA2 از هم متمایزند و با هم تفاوت مقداری دارند و حداقل در یک درس به عنوان کمک استاد با هم همکاری می‌کنند، مانند کمک استادهای T1 و T2 که هر دو در درس C1 به عنوان کمک استاد با هم همکاری می‌کنند. هم‌چنین در سؤال مطرح شده است که هر زوج یک بار نوشته شود، به این معنی که (y و x) و (x و y) هم‌زمان در نتیجه آورده نشوند، مانند (T1 و T2) و (T2 و T1) در سطرهای دوم و سوم جدول حاصل از الحاق دو جدول. بنابراین با توجه به دو شرط مطرح شده در صورت سؤال، خروجی حاصل از الحاق دو جدول به شکل زیر خواهد بود:

	Cname	TA1	TA2		Cname	TA1	TA2
* TA1, TA2 مقادیر غیر متمایز	C1	T1	T1	\Rightarrow	C1	T1	T2
✓ TA1 < TA2 مقادیر متمایز	C1	T1	T2				
* TA1 > TA2 مقادیر متمایز	C1	T2	T1				
* TA1, TA2 مقادیر غیر متمایز	C1	T2	T2				
* TA1 < TA2 مقادیر غیر متمایز	C2	T1	T1				
* TA1 > TA2 مقادیر غیر متمایز	C3	T3	T3				

توجه: برای برقراری شرط دوم که تأکید شده است که هر زوج با مقادیر متمایز یک بار نوشته شوند، باید شرط $TA_1 < TA_2$ یا $TA_1 > TA_2$ رعایت گردد، ما شرط $TA_1 < TA_2$ که در گزینه سوم هم آمده است، برای حل این مثال در نظر گرفته‌ایم.

عبارت گزینه اول هیچ شرطی روی TAها نگذاشته است و بنابراین نه تنها زوج‌های متقارن را در خروجی می‌آورد بلکه هر کمک استاد را با خودش هم در خروجی نشان می‌دهد.

در عبارت گزینه دوم، شرط $TA_1 < TA_2$ باعث می‌شود که یک کمک استاد با خودش در خروجی نیاید اما از نمایش زوج‌های متقارن جلوگیری نمی‌کند.

اما در عبارت گزینه سوم، شرط $TA_1 < TA_2$ باعث می‌شود فقط یکی از زوج‌های متقارن در خروجی ظاهر شود چون همواره اسم (یا کُد) دو کمک استاد با هم متفاوت خواهد بود. این شرط هم‌چنین از نمایش یک کمک استاد با خودش هم جلوگیری می‌کند.

۴۵- گزینه (۴) صحیح است.

در گزاره صورت سؤال، عمل نیم الحاق (یا نیم پیوند) همان کار پیوند طبیعی (\bowtie) را انجام می‌دهد با این تفاوت که در خروجی، فقط صفت‌های عملوند سمت چپ ظاهر می‌شوند. به عبارت دیگر، عبارت $R \times S$ سطرهای متناظر از R و S را به هم ربط می‌دهد اما در خروجی فقط ستون‌های R ظاهر می‌شوند. (منظور از «متناظر» بودن دو سطر از دو جدول این است که ستون‌های هم‌نامشان در دو جدول مقادیر یکسانی داشته باشند).

در گزاره I: عبارت $\Pi_X \Pi_Y(S)$ فقط ستون‌هایی از S که همنام با ستون‌های R باشند را باقی می‌گذارد. بنابراین با انجام پیوند طبیعی روی حاصل این عبارت با R همان نتیجه $R \times S$ حاصل می‌شود. در گزاره II: عبارت $R \times S$ سطرهایی از R که سطرهای متناظری در S دارند را به ما می‌دهد. اما فقط ستون‌های R ظاهر می‌شوند. با انجام پیوند طبیعی روی حاصل این عبارت با S مجدداً سطرهای متناظر از S و R به هم ربط داده می‌شوند اما ستون‌های S نیز در خروجی ظاهر می‌شوند. در گزاره III: عبارت $R \times S$ سطرهایی از R که سطر متناظری در S دارند را به ما می‌دهد اما فقط ستون‌های R در خروجی می‌آیند. به طور مشابه، عبارت $S \times R$ سطرهایی از S که سطر متناظر در R دارند را به ما می‌دهد اما فقط ستون‌های S در خروجی می‌آیند. بنابراین با انجام پیوند طبیعی روی حاصل این دو عبارت، همان نتیجه $R \bowtie S$ به دست می‌آید.

۴۶- گزینه (۳) صحیح است.

علمگرهای ضرب دکارتی (\times) و پیوند طبیعی (\bowtie) خاصیت جابه‌جایی (تقارنی) دارند. پس گزینه‌های اول و دوم صحیح‌اند. هم‌چنین در گزینه چهارم، ترتیب اعمال شرط‌های condition 1 و condition 2 تأثیری در نتیجه نهایی ندارند و هر دو معادل $\sigma_{\text{condition 1} \& \text{condition 2}}(R)$ هستند. اما عملگر پرتو (Π) همواره خاصیت جابه‌جایی ندارند. توجه کنید که عبارت $\Pi_{\text{list 2}}(\Pi_{\text{list 1}}(R))$ فقط در صورتی قابل انجام است که $\text{list 2} \subseteq \text{list 1}$ باشد که در آن صورت لزوماً $\text{list 1} \subseteq \text{list 2}$ نیست (مگر در حالت استثنایی‌ای که $\text{list 1} = \text{list 2}$ باشد)، بنابراین عبارت $\Pi_{\text{list 1}}(\Pi_{\text{list 2}}(R))$ در حالت کلی و در همه شرایط قابل انجام نیست.

۴۷- گزینه (۱) صحیح است.

در این سؤال، باید شناسه سفارش مشتریانی را بیابیم که همه کالاهای گروه food را خریده‌اند. برای نوشتن جبر رابطه‌ای برای پرس‌وجوهایی مشابه این که در آن‌ها کلمه «همه» آمده است مقسوم‌علیه (یعنی عبارت بعد از «همه») و مقسوم (یعنی عبارت قبل از «همه») را می‌نویسیم. مقسوم باید شامل صفت‌های مقسوم‌علیه و صفت‌های خواسته شده در خروجی باشد. در نهایت، مقسوم را بر مقسوم‌علیه تقسیم می‌کنیم. به سادگی دیده می‌شود که رابطه‌های R و S به ترتیب مقسوم و مقسوم‌علیه هستند. بنابراین، پاسخ سؤال $R \div S$ خواهد بود.

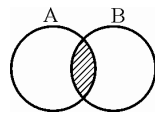
۴۸- گزینه (۲) صحیح است.

در بانک اطلاعات نامتمرکز، عملگر نیم‌پیوند، مانع از انتقال حجم زیادی از اطلاعات نامرتبط از یک سرور به سرور دیگر می‌شود.

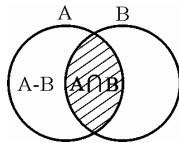
۴۹- گزینه (۳) صحیح است.

عملگرهای اجتماع، ضرب دکارتی و تفاضل، جزء عملگرهای اصلی جبر رابطه‌ای هستند و عملگرهایی که با استفاده از عملگرهای اصلی قابل ایجاد هستند، عملگرهای فرعی هستند، مانند عملگر اشتراک و تقسیم.

مثال:



$$A \cap B = A - (A - B) = B - (B - A)$$



با توجه به شکل مقابل، مشاهده می‌شود که با کنار گذاشتن ناحیه $A - B$ از ناحیه A ، قسمت باقی مانده برابر $A \cap B$ است.

۵۰- گزینه (۴) صحیح است.

پیوند طبیعی، عملگری است که دو رابطه را بر مبنای یک یا چند صفت مشترک به هم پیوند می‌دهد. در پیوند طبیعی دو رابطه R_1 و R_2 یعنی $R_1 \bowtie R_2$ اگر R_1 و R_2 صفت مشترکی نداشته باشند، $R_1 \bowtie R_2$ تبدیل به $R_1 \times R_2$ خواهد شد و اگر روابط R_1 و R_2 تماماً دارای صفات یکسانی باشند، آن‌گاه $R_1 \bowtie R_2$ معادل $R_1 \cap R_2$ خواهد شد.

مثال زیر گویای مطلب می‌باشد:

$$\begin{array}{c|c|c} a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \end{array} \bowtie \begin{array}{c|c|c} a & b & c \\ \hline 1 & 2 & 3 \\ \hline 1 & 2 & 3 \end{array} = \begin{array}{c|c|c} a & b & c \\ \hline 1 & 2 & 3 \\ \hline 1 & 2 & 3 \end{array}$$

$$\begin{array}{c|c|c} a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \end{array} \cap \begin{array}{c|c|c} a & b & c \\ \hline 1 & 2 & 3 \\ \hline 1 & 2 & 3 \end{array} = \begin{array}{c|c|c} a & b & c \\ \hline 1 & 2 & 3 \\ \hline 1 & 2 & 3 \end{array}$$

۵۱- گزینه (۴) صحیح است.

عملگر ضرب دکارتی، تمام رکوردهای یک جدول را در کنار تمام رکوردهای یک جدول دیگر قرار می‌دهد و در خروجی نمایش می‌دهد. بنابراین ضرب دکارتی زمان و حافظه بیشتری نسبت به پیوند شرطی، طبیعی و نیم‌پیوند می‌خواهد.

۵۲- گزینه (۲) صحیح است.

همانطور که می‌دانید یک مجموعه با k عضو، دارای 2^k زیر مجموعه است که البته یکی از آن‌ها، مجموعه تهی است. بنابراین اگر رابطه‌ای دارای k خصیصه باشد، تعداد کل تصاویر (به معنی عملگر پرتو یا Π) آن، 2^k خواهد بود که اگر تصویر تهی آن را کنار بگذاریم برابر $2^k - 1$ تصویر غیر تهی می‌شود.

مثال: جدول زیر را با سه ستون در نظر بگیرید، تعداد تصاویر غیر تهی برابر چیست؟

X	Y	Z

X	Y	Z	تصویر
0	0	0	تهی \Rightarrow که جدول بدون ستون وجود ندارد.
0	0	1	Z
0	1	0	Y
0	1	1	YZ
1	0	0	X
1	0	1	XZ
1	1	0	XY
1	1	1	XYZ

بنابراین تعداد تصاویر غیرتهی برابر مقدار $2^3 - 1 = 7$ خواهد بود، که در یک عبارت کلی رابطه $2^k - 1$ را که k برابر تعداد ستون‌ها است، خواهیم داشت.

همچنین در یک راه‌حل دیگر داریم:

هر تصویر غیرتهی از جدول R شامل 1، 2، 3، ... یا k صفت از جدول R است. بنابراین تعداد آنها برابر است با:

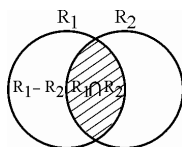
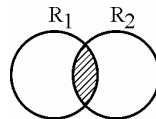
$$\binom{k}{1} + \binom{k}{2} + \binom{k}{3} + \dots + \binom{k}{k} = 2^k - 1$$

۵۳- گزینه (۴) صحیح است.

عملگرهای اجتماع، ضرب دکارتی و تفاضل جزء عملگرهای اصلی جبر رابطه‌ای هستند و عملگرهایی که با استفاده از عملگرهای اصلی قابل ایجاد هستند، عملگرهای فرعی هستند، مانند عملگر اشتراک و تقسیم.

$$R_1 \cap R_2 = R_1 - (R_1 - R_2) = R_2 - (R_2 - R_1)$$

مثال:



با توجه به شکل مقابل، مشاهده می‌شود که با کنار گذاشتن ناحیه $R_1 - R_2$ از ناحیه R_1 ، قسمت باقی مانده برابر $R_1 \cap R_2$ است.

۵۴- گزینه (۳) صحیح است.

جدول زیر را در نظر بگیرید:

X	Y
1	5
3	2
3	4

جدول a

X	Y
1	2
3	2
5	6
3	7

جدول b

توجه: دقت کنید که هر طرف رابطه به تنهایی باید درست در نظر گرفته شود و برابری آن با طرف درست مقابل مقایسه گردد، در واقع هر یک از طرفین مقایسه فاقد خطای نحوی هستند. برای مثال شروط

سازگاری برای عملگر تفاضل به معنای ستون‌های مشابه، باید در نظر گرفته شود.
گزینه اول درست است.

$$\sigma_p(a - b) = \sigma_p(a) - b$$

مثال:

مقدار شرط p را برای جدول a برابر $X=3$ در نظر بگیرید، بنابراین داریم:

$$\sigma_{X=3}(a - b) = \sigma_{X=3}(a) - b$$

$$\sigma_{X=3} \left(\begin{array}{c|c} X & Y \\ \hline 1 & 5 \\ 3 & 2 \\ 3 & 4 \end{array} - \begin{array}{c|c} X & Y \\ \hline 1 & 2 \\ 3 & 2 \\ 5 & 6 \\ 3 & 7 \end{array} \right) = \sigma_{X=3} \left(\begin{array}{c|c} X & Y \\ \hline 1 & 5 \\ 3 & 2 \\ 3 & 4 \end{array} \right) - \begin{array}{c|c} X & Y \\ \hline 1 & 2 \\ 3 & 2 \\ 5 & 6 \\ 3 & 7 \end{array}$$

$$\downarrow \qquad \qquad \qquad \downarrow$$

$$\begin{array}{c|c} X & Y \\ \hline 3 & 4 \end{array} \qquad \qquad \qquad \begin{array}{c|c} X & Y \\ \hline 3 & 4 \end{array}$$

گزینه دوم درست است.

$$\sigma_p(a - b) = \sigma_p(a) - \sigma_p(b)$$

مثال:

مقدار شرط p را برای جدول a و b برابر $X=3$ در نظر بگیرید، بنابراین داریم:

$$\sigma_{X=3} \left(\begin{array}{c|c} X & Y \\ \hline 1 & 5 \\ 3 & 2 \\ 3 & 4 \end{array} - \begin{array}{c|c} X & Y \\ \hline 1 & 2 \\ 3 & 2 \\ 5 & 6 \\ 3 & 7 \end{array} \right) = \sigma_{X=3} \left(\begin{array}{c|c} X & Y \\ \hline 1 & 5 \\ 3 & 2 \\ 3 & 4 \end{array} \right) - \sigma_{X=3} \left(\begin{array}{c|c} X & Y \\ \hline 1 & 2 \\ 3 & 2 \\ 5 & 6 \\ 3 & 7 \end{array} \right)$$

$$\downarrow \qquad \qquad \qquad \downarrow$$

$$\begin{array}{c|c} X & Y \\ \hline 3 & 4 \end{array} \qquad \qquad \qquad \begin{array}{c|c} X & Y \\ \hline 3 & 4 \end{array}$$

گزینه سوم نادرست است.

عبارت $\sigma_p(a \cap b)$ ، زیر مجموعه‌ای از عناصر موجود در $a \cap b$ را به ما می‌دهد، اما $a - b$ ، عناصری از a را می‌دهد که در b و در نتیجه در $a \cap b$ وجود ندارد.

$$\sigma_p(a \cap b) = a - b$$

مثال:

مقدار شرط p را برای جدول a و b برابر $X=3$ در نظر بگیرید، بنابراین داریم:

$$\sigma_{X=3} \left(\begin{array}{c|c} X & Y \\ \hline 1 & 5 \\ 3 & 2 \\ 3 & 4 \end{array} \cap \begin{array}{c|c} X & Y \\ \hline 1 & 2 \\ 3 & 2 \\ 5 & 6 \\ 3 & 7 \end{array} \right) \neq \begin{array}{c|c} X & Y \\ \hline 1 & 5 \\ 3 & 2 \\ 3 & 4 \end{array} - \begin{array}{c|c} X & Y \\ \hline 1 & 2 \\ 3 & 2 \\ 5 & 6 \\ 3 & 7 \end{array}$$

$$\downarrow \qquad \qquad \qquad \downarrow$$

$$\begin{array}{c|c} X & Y \\ \hline 3 & 2 \end{array} \qquad \qquad \qquad \begin{array}{c|c} X & Y \\ \hline 1 & 5 \\ 3 & 4 \end{array}$$

گزینه چهارم درست است.

$$a - b \neq b - a$$

مثال:

عبارت گزینه سوم مطابق اصل جابه‌جانیپذیری عملگر تفاضل، درست است.

۵۵- گزینه (۴) صحیح است.

سه جدول students، take و courses با مقادیر زیر را در نظر بگیرید:

sid	sname	sid	cid	grade	cid	cname	dept
s1	sn1	s1	c1	12	c1	cn1	CS
s2	sn2	s1	c2	14	c2	cn2	CS
s3	sn3	s2	c1	16	c3	cn3	CE
s4	sn4	s3	c3	18	c4	cn4	IT
s5	sn5	s4	c4	20			

جدول students

جدول take

جدول courses

گزینه اول نادرست است. جدول take که می‌گوید هر دانشجویی چه درسی را اخذ کرده و چه نمره‌ای گرفته است، مقسوم (مؤلفه اول) است و جدول courses با شرط «دروس ارائه شده در دانشکده CS» مقسوم علیه (مؤلفه دوم) است. مطابق تعریف عملگر تقسیم در جبر رابطه‌ای، باید تمام خصیصه‌های مقسوم علیه عملگر تقسیم در مقسوم باشد. به عبارت دیگر مجموعه صفات مقسوم علیه، زیر مجموعه، مجموعه صفات مقسوم باشد، اما همانطور که در این گزینه مشاهده می‌شود، صفات cname و dept در مقسوم علیه وجود دارند اما در مقسوم وجود ندارند. بنابراین اجرای عملگر تقسیم در گزینه اول، به دلیل وجود خطای نحوی امکان‌پذیر نیست.

گزینه دوم نادرست است. مطابق تعریف عملگر تقسیم در جبر رابطه‌ای، اجرای عملگر تقسیم در گزینه دوم امکان‌پذیر است، اما خروجی مطرح شده در صورت سؤال را استخراج نمی‌کند.

$$\Pi_{sid} \left[\begin{array}{c} \text{take} \\ \downarrow \end{array} \div \frac{\Pi_{cid}(\sigma_{dept='cs'}(\text{courses}))}{\downarrow} \right]$$

sid	cid	grade	
s1	c1	12	÷
s1	c2	14	
s2	c1	16	
s3	c3	18	
s4	c4	20	

$$\Pi_{sid} \left[\begin{array}{c|c} \text{sid} & \text{grade} \\ \hline & \end{array} \right] \rightarrow \underline{\text{sid}}$$

توجه: به ازای سطر (s1,12) مقدار c1 و هم c2 برقرار نیست،

همچنین به ازای سطر $(s_1, 14)$ مقدار c_1 و هم c_2 برقرار نیست،
 همچنین به ازای سطر $(s_2, 16)$ مقدار c_1 و هم c_2 برقرار نیست،
 همچنین به ازای سطر $(s_3, 18)$ مقدار c_1 و هم c_2 برقرار نیست،
 و در نهایت به ازای سطر $(s_4, 20)$ مقدار c_1 و هم c_2 برقرار نیست، نتیجه اینکه حاصل تقسیم تهی خواهد بود.

گزینه چهارم درست است. اگر r و s دو جدول (رابطه) به ترتیب با مجموعه ستون‌های R و S باشند، آنگاه $r \div s$ به صورت زیر قابل تعریف است اگر و فقط اگر $S \subseteq R$ باشد.

$$r \div s = \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times S) - r)$$

مثال: دو جدول زیر را در نظر بگیرید:

a	b	b
5	7	1
5	1	2
4	2	جدول s
4	3	
5	2	

جدول r

$$R = \{a, b\} = \text{مجموعه ستون‌های جدول } r$$

$$S = \{b\} = \text{مجموعه ستون‌های جدول } s$$

$$R - S = \{a, b\} - \{b\} = \{a\}$$

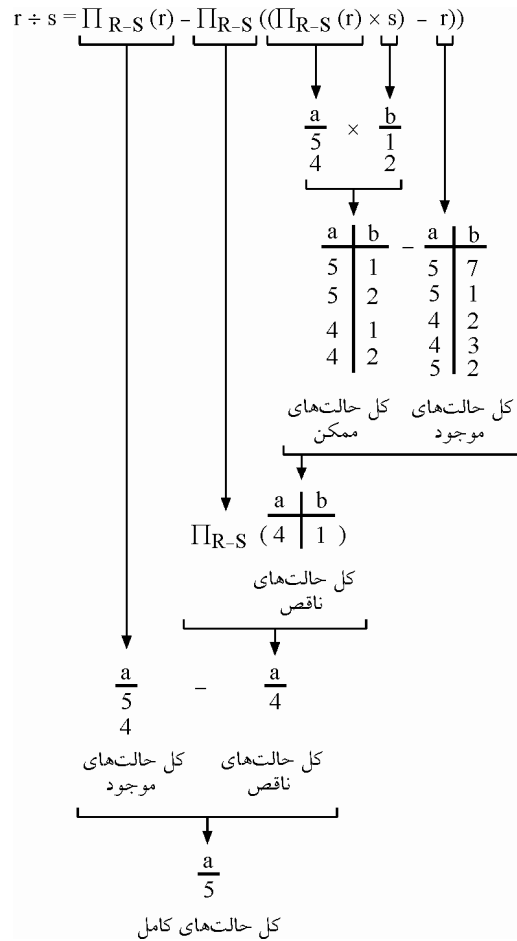
توجه: می‌توان به جای مولفه r در انتهای رابطه‌ی فوق از مؤلفه $\Pi_{R-S, S}(r)$ نیز استفاده نمود، زیرا دو مؤلفه برابر هستند.

توجه: مطابق تعریف عملگر تقسیم، ستون‌های حاصل از تقسیم دو جدول برابر تفاضل ستون‌های دو جدول است. همچنین سطرهایی از جدول r در خروجی ظاهر می‌گردد که تمام اعضای جدول s را در کنار خود داشته باشد. بنابراین داریم:

a	b	\div	$\frac{b}{1} = \frac{a}{5}$
5	7		2
5	1		2
4	2		جدول s
4	3		
5	2		

جدول r

همچنین براساس رابطه تعریف شده داریم:



توجه: یکی از عملگرهای اصلی جبر رابطه‌ای، عملگر تفاضل است. اگر R و S دو رابطه باشند، منظور از R - S مجموعه کلیه سطرهایی است که عضو R هستند، اما در S حضور ندارند.

توجه: در جبر رابطه‌ای تفاضل هر دو رابطه دلخواه امکان‌پذیر نیست. تفاضل دو رابطه در جبر رابطه‌ای زمانی امکان‌پذیر است که اگر و فقط اگر شروط سازگاری در مورد آن‌ها برقرار باشد.

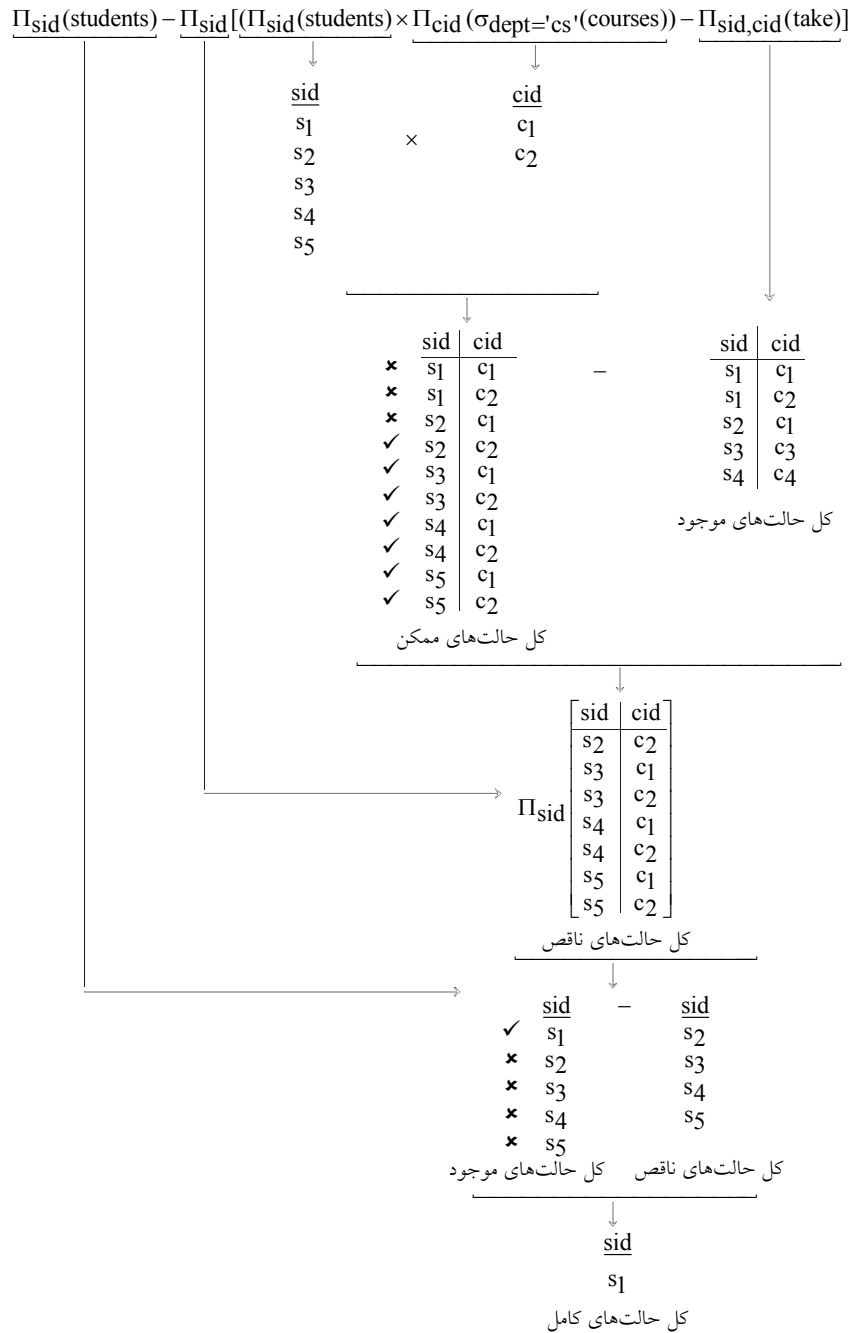
شروط سازگاری: در جبر رابطه‌ای دو شرط به عنوان شروط سازگاری مطرح است:

شرط اول: تعداد ستون‌های دو جدول یکسان باشد، به عبارت دیگر دو جدول هم درجه باشند.

شرط دوم: نوع یا دامنه ستون‌های متناظر در دو جدول یکسان باشد.

توجه: رابطه تقسیم قبل را مجدداً مشاهده نمایید، همانطور که واضح است، هرگاه از عملگر تفاضل استفاده شده است در دو طرف عملگر، شروط سازگاری رعایت شده است.

توجه: براساس مطالب فوق، رابطه موجود در گزینه چهارم را مطابق جداول مطرح شده در نظر بگیرید:



توجه: مشاهده می‌شود که شماره دانشجویی s1، همه دروس ارائه شده در دانشکده 'cs' را اخذ کرده است.

گزینه سوم نادرست است. زیرا مؤلفه $\Pi_{sid}(\text{students})$ را نسبت به گزینه چهارم برای پاسخ‌دهی به پرس‌وجوی موردنظر ندارد. در واقع خروجی گزینه سوم به صورت زیر است:

sid
s2
s3
s4
s5

یعنی شماره دانشجویانی را استخراج می‌کند که همه دروس ارائه شده در دانشکده 'cs' را اخذ نکرده‌اند.

۵۶- گزینه (۴) صحیح است.

مطابق فرض صورت سؤال، دستور $\rho_Y(X)$ ، ایجاد یک کپی (یک نام مستعار) از رابطه X با نام Y است. جدول زیر را در نظر بگیرید:

<u>E#</u>	<u>Salary</u>
S ₁	10
S ₂	20
S ₃	30

جدول Emp

پرس‌وجوی مطرح شده در صورت سؤال به صورت زیر است:

$\Pi_{\text{salary}}(\text{Emp}) - \Pi_{\text{first.salary}}(\sigma_{\text{first.salary} > \text{second.salary}}(\rho_{\text{first}}(\text{Emp}) \times \rho_{\text{second}}(\text{Emp})))$

توجه: یکی از عملگرهای اصلی جبر رابطه‌ای، عملگر تفاضل است. اگر R و S دو رابطه باشند، منظور از $R-S$ مجموعه کلیه سطرهایی است که عضو R هستند، اما در S حضور ندارند.

توجه: در جبر رابطه‌ای تفاضل هر دو رابطه دلخواه امکان‌پذیر نیست. تفاضل دو رابطه در جبر رابطه‌ای زمانی امکان‌پذیر است که اگر و فقط اگر شروط سازگاری در مورد آن‌ها برقرار باشد. شروط سازگاری: در جبر رابطه‌ای دو شرط به عنوان شروط سازگاری مطرح است: شرط اول: تعداد ستون‌های دو جدول یکسان باشد به عبارت دیگر دو جدول هم درجه باشند.

شرط دوم: نوع یا دامنه ستون‌های دو جدول یکسان باشد.

توجه: در پرس‌وجوی مطرح شده در هر دو طرف عملگر تفاضل، شروط سازگاری رعایت شده است.

براساس پرس‌وجوی مطرح شده در صورت سؤال داریم:

در ابتدا توسط عبارت $\Pi_{\text{salary}}(\text{Emp})$ در سمت چپ عملگر تفاضل حقوق کارمندان از جدول Emp به صورت زیر استخراج می‌گردد:

salary
10
20
30

همچنین براساس بخش سمت راست عملگر تفاضل داریم:
 عبارت $(\rho_{\text{first}}(\text{Emp}) \times \rho_{\text{second}})$ ضرب دکارتی جدول Emp در خودش است. حاصل این ضرب دکارتی، جدولی است که در آن، رکورد اطلاعاتی هر کارمند جدول first در کنار رکوردهای اطلاعاتی همه کارمندان جدول second قرار می‌گیرد.

first.E#	first.salary	second.E#	second.salary
S ₁	10	S ₁	10
S ₁	10	S ₂	20
S ₁	10	S ₃	30
S ₂	20	S ₁	10
S ₂	20	S ₂	20
S ₂	20	S ₃	30
S ₃	30	S ₁	10
S ₃	30	S ₂	20
S ₃	30	S ₃	30

سپس شرط $\sigma_{\text{first.salary} > \text{second.salary}}$ ، سطریهایی که کارمند جدول first حقوقش حداقل از یک نفر دیگر بیشتر باشد را انتخاب می‌کند (یعنی کارندهایی که حقوقشان کمترین حقوق نیست). بنابراین، پس از اعمال شرط داریم:

first.E#	first.salary	second.E#	second.Salary
S ₂	20	S ₁	10
S ₃	30	S ₁	10
S ₃	30	S ₂	20

در ادامه، توسط عملگر $\Pi_{\text{first.salary}}$ ، حقوق این نوع از کارمندان، به صورت زیر استخراج می‌گردد.

<u>first.salary</u>
20
30

توجه: در جبر رابطه‌ای، سطریهای تکراری در خروجی قرار نمی‌گیرند.
 در انتها، با انجام عملگر تفاضل، حقوق کارمندان فوق، از مجموعه حقوق کلیه کارمندان کنار گذاشته می‌شوند. آنچه باقی می‌ماند، حقوق کارمند یا کارمندانی است که کمتر از همه حقوق می‌گیرند. به عبارت دیگر مینیمم حقوق کارمندان استخراج می‌گردد.
 خروجی نهایی پرس‌وجوی مطرح شده به صورت زیر است:

<u>salary</u>	<u>first.salary</u>	<u>salary</u>
10	20	10
20	30	
30		

توجه: اگر شرط موجود در پرس‌وجوی مطرح شده عکس گردد، آنگاه گزینه سوم ایجاد می‌گردد.

۵۷- گزینه (۲) صحیح است.

عملگر الحاق طبیعی

عملگر الحاق طبیعی سطرهایی از دو رابطه را بر مبنای ستون‌های مشترک (هم نام و هم دامنه)، که مقدارشان در همه ستون‌های مشترک مساوی است، به هم پیوند می‌زند و در خروجی نمایش می‌دهد، ستون‌های مشترک فقط یکبار در خروجی ظاهر می‌شوند. به بیان دیگر ستون‌های جدول خروجی، برابر اجتماع ستون‌های جدول‌های ورودی هستند.

مثال: یک ستون مشترک.

$$\begin{array}{c|c} a & b \\ \hline 2 & 5 \\ 8 & 1 \\ 4 & 2 \\ 7 & 5 \\ 1 & 6 \end{array} \bowtie \begin{array}{c|c} b & c \\ \hline 6 & 1 \\ 5 & 2 \\ 4 & 7 \\ 8 & 9 \\ 1 & 6 \end{array} = \begin{array}{c|c|c} a & b & c \\ \hline 2 & 5 & 2 \\ 8 & 1 & 6 \\ 4 & 2 & 7 \\ 7 & 5 & 9 \\ 1 & 6 & 6 \end{array}$$

مثال: دو ستون مشترک.

$$\begin{array}{c|c|c} a & b & c \\ \hline 2 & 4 & 1 \\ 3 & 1 & 4 \\ 2 & 5 & 2 \\ 6 & 1 & 1 \end{array} \bowtie \begin{array}{c|c|c} b & c & d \\ \hline 2 & 3 & 6 \\ 1 & 6 & 1 \\ 1 & 4 & 2 \\ 4 & 2 & 1 \end{array} = \begin{array}{c|c|c|c} a & b & c & d \\ \hline 3 & 1 & 4 & 2 \\ 2 & 3 & 6 & 1 \\ 1 & 4 & 2 & 1 \\ 4 & 2 & 1 & 2 \end{array}$$

عملگر الحاق خارجی راست (right outer join)

این عملگر نیز مانند الحاق طبیعی ستون‌های مشترک را فقط یکبار در خروجی قرار می‌دهد. همچنین کلیه سطرهای پیوندپذیر را در خروجی قرار می‌دهد اما علاوه بر آن کلیه سطرهای پیوندناپذیر جدول سمت راست را نیز در خروجی قرار می‌دهد و در این حالت برای ستون‌های غیرمشترک جدول سمت چپ مقدار NULL قرار می‌دهد.

به بیان دیگر ابتدا دو جدول ورودی با یکدیگر الحاق طبیعی می‌شوند، سپس سطرهایی از جدول سمت راست که به دلیل عدم تطابق در الحاق طبیعی شرکت نکرده‌اند، به حاصل الحاق اضافه می‌گردد و برای ستون‌های غیرمشترک جدول سمت چپ مقدار NULL درج می‌گردد.

توجه: الحاق خارجی راست میان دو رابطه، با نماد \bowtie نشان داده می‌شود.

مثال: یک ستون مشترک.

$$\begin{array}{c|c} a & b \\ \hline 2 & 5 \\ 8 & 1 \\ 4 & 2 \\ 7 & 5 \end{array} \bowtie \begin{array}{c|c} b & c \\ \hline \times 6 & 1 \\ \checkmark 5 & 2 \\ \times 4 & 7 \\ \times 8 & 9 \\ \checkmark 1 & 6 \end{array} = \begin{array}{c|c|c} a & b & c \\ \hline 2 & 5 & 2 \\ 8 & 1 & 6 \\ 7 & 5 & 2 \\ \hline NULL & 6 & 1 \\ NULL & 4 & 7 \\ NULL & 8 & 9 \end{array}$$

مثال: دو ستون مشترک.

a	b	c	b	c	d	a	b	c	d
2	4	1	× 2	3	6	3	1	4	2
3	1	4	× 1	6	1	NULL	2	3	6
2	5	2	✓ 1	4	2	NULL	1	6	1
6	1	1	× 4	2	1	NULL	4	2	1

تعریف کلید خارجی

کلید خارجی برای ارتباط میان جداول مورد استفاده قرار می‌گیرد. اگر ستون(هایی) در یک جدول به عنوان کلید خارجی تعریف شود، این ستون(ها) باید در همان جدول یا جدول دیگری به عنوان کلید کاندید (اصلی یا فرعی) وجود داشته باشد.

توجه: صرف وجود ستون مشترک، به معنای وجود کلید خارجی در یک جدول نیست، بلکه برای یک جدول، کلید خارجی باید تعریف گردد.

توجه: کلید خارجی در صورت وجود برای یک جدول باید هم نوع با کلید کاندید باشد، اما الزامی ندارد که با کلید کاندید هم نام باشد.

توجه: به ازای هر مقدار موجود در یک کلید خارجی، باید دقیقاً یک مقدار متناظر در کلید کاندید متناظر آن وجود داشته باشد، در غیر این صورت می‌گوییم، کلید خارجی دارای ارجاع NULL است. به بیان دیگر، مقادیر کلید خارجی همواره باید زیرمجموعه مقادیر کلید کاندید باشد.

توجه: یک کلید خارجی، در یک رابطه هیچگاه نباید ارجاع NULL داشته باشد، این مسأله را به عنوان یک قانون جامعیتی در مدل رابطه‌ای می‌شناسیم و آن را قانون جامعیت ارجاعی می‌نامیم.

مثال:

کلید کاندید			کلید خارجی			کلید خارجی			کلید کاندید		
S#	Sname	city	S#	P#	QTY	P#	Pname	color			
S ₁	Sn1	C ₁	S ₁	P ₁	10	P ₁	Pn1	Red			
S ₂	Sn2	C ₂	S ₁	P ₂	20	P ₂	Pn2	Blue			
S ₃	Sn3	C ₃	S ₂	P ₁	30	P ₃	Pn3	Blue			
			NULL ←	S ₄	P ₃	40					

جدول S

جدول SP

جدول P

توجه: S#، در جدول S، کلید کاندید است، پس عناصرش یکتا است.

توجه: S# در جدول SP، به عنوان کلید خارجی، خاصیت کلیدی ندارد، پس ممکن است، مقدار تکراری داشته باشد.

توجه: درج رکورد (S₄, P₃, 40) در جدول SP، غیرمجاز و غیرممکن است، زیرا سبب ارجاع NULL می‌گردد.

مثال: فرض کنید در یک نوع پیاده‌سازی کشف ارجاع NULL، قبل از هر بار درج رکورد در جدول SP، الحاق طبیعی دو جدول به صورت زیر اجرا گردد و تعداد سطرهاى جواب حاصل شمارش گردد. بنابراین

داریم:

S#	Sname	city	S#	P#	QTY	S#	Sname	city	P#	QTY
S ₁	Sn1	C1	S ₁	P ₁	10	S ₁	Sn1	C ₁	P ₁	10
S ₂	Sn2	C2	S ₁	P ₂	20	S ₁	Sn2	C ₁	P ₂	20
S ₃	Sn3	C3	S ₂	P ₁	30	S ₂	Sn2	C ₂	P ₁	30

جدول S

جدول SP

توجه: تعداد سطرهای پرس وجوی حاصل برابر 3 می باشد.

حال اگر پرس از هر بار درج رکورد در جدول SP (مثلاً درج رکورد (S₄, P₃, 40))، الحاق خارجی راست به صورت زیر اجرا گردد و تعداد سطرهای حاصل از این پرس وجو بیشتر از تعداد سطرهای حاصل از الحاق طبیعی باشد، این نتیجه بدین معنی خواهد بود که در درج اخیر ارجاع NULL رخ داده است و باید از درج رکورد مورد نظر در جدول SP، ممانعت به عمل آید.

S#	Sname	city	S#	P#	QTY	S#	Sname	city	P#	QTY
S ₁	Sn1	C ₁	S ₁	P ₁	10	S ₁	Sn1	C ₁	P ₁	10
S ₂	Sn2	C ₂	S ₁	P ₂	20	S ₁	Sn1	C ₁	P ₂	20
S ₃	Sn3	C ₃	S ₂	P ₁	30	S ₂	Sn2	C ₂	P ₁	30
			S ₄	P ₃	40	S ₄	NULL	NULL	P ₃	40

NULL ←

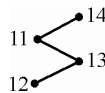
توجه: تعداد سطرهای پرس وجوی حاصل برابر 4 است. پس ارجاع NULL رخ داده است. براساس جدول مطرح شده در صورت سؤال، جداول course و prereg را با مقادیر زیر در نظر بگیرید:

course-id	title	units	course-id	prereg-id
11	t ₁	2	13	11
12	t ₂	2	13	12
13	t ₃	3	14	11
14	t ₄	3		

جدول course

جدول prereg

توجه: یک درس مانند 13 می تواند دروس 11 و 12 را به عنوان پیش نیاز، نیاز داشته باشد و یک درس مانند 11، می تواند پیش نیاز دروس 13 و 14 باشد. به شکل زیر توجه نمایید:



پیاده سازی دو جدول course و prereg در SQL می تواند به صورت زیر باشد.

course - id	title	units	course - id	course - id	title	units
11	t ₁	2	13	13	t ₃	3
12	t ₂	2	14	14	t ₄	3
13	t ₃	3	NULL ← 16	16	NULL	NULL
14	t ₄	3				

جدول course

توجه: تعداد سطرهای پرس وجوی حاصل برابر 3 است. پس ارجاع NULL رخ داده است.
نتیجه: پس آزمون اول ارجاع NULL را کشف کرد.

آزمون دوم:

$$\text{course} \bowtie (\rho_{t(\text{course-id})}(\Pi_{\text{prereg-id}(\text{prereg})}))$$

توجه: دستور نام گذاری $\rho_{t(\text{course-id})}$ ، جدول خروجی دستور $\Pi_{\text{prereg-id}(\text{prereg})}$ را برابر t و نام تنها ستون آن را برابر course-id قرار می دهد.
این تغییر نام برای فراهم سازی امکان اجرای عملگر الحاق طبیعی (دارا بودن ستون مشترک هم نام و هم نوع) انجام شده است.

course - id	title	units	course - id	course - id	title	units
11	t ₁	2	11	11	t ₁	2
12	t ₂	2	12	12	t ₂	2
13	t ₃	3				
14	t ₄	3				

توجه: تعداد سطرهای پرس وجوی حاصل برابر 2 است.

حال اگر پس از هر بار درج رکورد در جدول prereg (مثلاً درج رکورد (15 و 16))، الحاق خارجی راست به صورت زیر اجرا گردد و تعداد سطرهای حاصل از این پرس وجو بیشتر از تعداد سطرهای حاصل از الحاق طبیعی باشد، این نتیجه بدین معنی خواهد بود که در درج اخیر ارجاع NULL رخ داده است و باید از درج رکورد مورد نظر در جدول prereg، ممانعت به عمل آید.

$$\text{course} \bowtie (\rho_{t(\text{course-id})}(\Pi_{\text{prereg-id}(\text{prereg})}))$$

course - id	title	units	course - id	course - id	title	units
11	t ₁	2	11	11	t ₁	2
12	t ₂	2	12	12	t ₂	2
13	t ₃	3	NULL ← 15	15	NULL	NULL
14	t ₄	3				

توجه: تعداد سطرهای پرس وجوی حاصل برابر 3 است. پس ارجاع NULL رخ داده است.
نتیجه: پس آزمون دوم نیز ارجاع NULL را کشف کرد.

توجه: اگر حداقل یکی از آزمون‌های اول و دوم موفق به کشف ارجاع NULL گردد، باید از درج رکورد مورد نظر در جدول prereg، ممانعت به عمل آید.

توجه: اگر پرس‌وجوی مطرح شده در گزینه دوم اجرا گردد، مطابق تعریف الحاق خارجی، الحاق بر روی ستون مشترک یعنی course-id انجام می‌گردد، در این حالت پرس‌وجو فقط قادر خواهد بود آزمون نوع اول مطرح شده را انجام دهد و فقط خطاهای درج منجر به ارجاع NULL را در ستون course-id از جدول prereg کشف کند. در صورت سؤال هم گفته شده است که کدام گزینه می‌تواند وجود مشکلی در اطلاعات ثبت شده در سیستم را کشف کند، بنابراین گزینه دوم درست است.
بر اساس الحاق طبیعی دو جدول prereg و course داریم:

course - id	title	units	course - id	prereg - id	course - id	prereg - id	title	units
11	t ₁	2	13	11	13	11	t ₃	3
12	t ₂	2	13	12	13	12	t ₃	3
13	t ₃	3	14	11	14	11	t ₄	3
14	t ₄	3	NULL	16	16	15		

جدول course

جدول prereg

توجه: تعداد سطرهای پرس‌وجوی حاصل برابر 3 است.

حال اگر پس از هر بار درج رکورد در جدول prereg (مثلاً درج رکورد (15 و 16)) الحاق خارجی راست به صورت زیر اجرا گردد و تعداد سطرهای حاصل از این پرس‌وجو بیشتر از تعداد سطرهای حاصل از الحاق طبیعی باشد، این نتیجه بدین معنی خواهد بود که در درج اخیر ارجاع NULL رخ داده است و باید از درج رکورد مورد نظر در جدول prereg، ممانعت به عمل آید.

بر اساس الحاق خارجی راست دو جدول prereg و course داریم:

course - id	title	units	course - id	prereg - id	course - id	prereg - id	title	units
11	t ₁	2	13	11	13	11	t ₃	3
12	t ₂	2	13	12	13	12	t ₃	3
13	t ₃	3	14	11	14	11	t ₄	3
14	t ₄	3	NULL	16	16	15	NULL	NULL

جدول course

جدول prereg

توجه: تعداد سطرهای پرس‌وجوی حاصل برابر 4 است. پس ارجاع NULL رخ داده است.

توجه: گزینه دوم مطرح شده در صورت سؤال، قادر به کشف ارجاع NULL در ستون prereg-id در جدول prereg نیست.

بر اساس الحاق طبیعی دو جدول prereg و course داریم:

course - id	title	units	course - id	prereq - id	course - id	prereq - id	title	units
11	t ₁	2	13	11	13	11	t ₃	3
12	t ₂	2	13	12	13	12	t ₃	3
13	t ₃	3	14	11	13	15	t ₃	3
14	t ₄	3	13	15	14	11	t ₄	3

جدول course

جدول prereq

توجه: تعداد سطرهای پرس و جوی حاصل برابر 4 است. حال اگر پس از هر بار درج رکورد در جدول prereq (مثلاً رکورد (15 و 13)) الحاق خارجی راست به صورت زیر اجرا گردد و تعداد سطرهای حاصل از این پرس و جو بیشتر از تعداد سطرهای حاصل از الحاق طبیعی باشد. این نتیجه بدین معنی خواهد بود که در درج اخیر ارجاع NULL رخ داده است و باید از درج رکورد مورد نظر در جدول prereq، ممانعت به عمل آید. بر اساس الحاق خارجی راست دو جدول prereq و course داریم:

course - id	title	units	course - id	prereq - id	course - id	prereq - id	title	units
11	t ₁	2	13	11	13	11	t ₃	3
12	t ₂	2	13	12	13	12	t ₃	3
13	t ₃	3	14	11	13	15	t ₃	3
14	t ₄	3	13	15	14	11	t ₄	3

جدول course

جدول prereq

توجه: تعداد سطرهای پرس و جوی حاصل باز هم برابر 4 شد، در حالی که ارجاع NULL به واسطه مقدار 15 در ستون prereq-id از جدول prereq وجود دارد، دقت کنید که مطابق تعریف کلید خارجی برای عدم وجود ارجاع NULL، باید مقادیر هر یک از ستون‌های course-id و prereq-id از جدول prereq، زیرمجموع مقادیر ستون course-id از جدول course باشد. اما مطابق آنچه گفتیم، پرس و جوی مطرح شده در گزینه دوم قادر به کشف ارجاع NULL در ستون prereq-id از جدول prereq نیست.

۵۸- گزینه (۳) صحیح است.

جداول زیر را در نظر بگیرید:

mem-no	name	age	mem-no	isbn	date	isbn	...	publisher
m1	mn1	21	m1	is1	11	is1		McGraw-Hill
m2	mn2	22	m1	is2	12	is2		McGraw-Hill
m3	mn3	23	m2	is1	13	is3		Penguin
	Member		m2	is2	۱۴	is4		Penguin
			m2	is3	۱۵			Book
								Borrowed

مطابق پرس و جوی مطرح شده در صورت سوال داریم:

«نام تمامی اعضایی را بیابید که همه کتاب‌های منتشرشده توسط McGraw-Hill را به امانت برده‌اند.»
توجه: کلمه «تمامی» در پرس و جوی مطرح شده در صورت سوال بی‌تاثیر است، چه این کلمه باشد و چه نباشد، تمامی اعضایی که همه کتاب‌های منتشرشده توسط McGraw-Hill را به امانت برده‌اند، استخراج می‌گردند.

پرس و جوی فوق توسط عملگر تقسیم در جبر رابطه‌ای قابل پیاده‌سازی است. مطابق تعریف عملگر تقسیم در جبر رابطه‌ای، باید تمام خصیصه‌های مقسوم علیه در مقسوم باشد. به عبارت دیگر مجموعه صفات مقسوم علیه، زیر مجموعه، مجموعه صفات مقسوم باشد. همچنین ستون‌های حاصل از عملگر تقسیم برابر تفاضل ستون‌های مقسوم و مقسوم علیه است. همچنین سطرهایی از جدول r در خروجی ظاهر می‌گردد که تمام اعضای جدول s را در کنار خود داشته باشد.

مثال:

a	b	\div	$\frac{b}{1}$	$=$	$\frac{a}{5}$
5	7		2		
5	1				
4	2				s جدول
4	3				
5	2				

جدول r

جدول Borrowed که می‌گوید هر عضوی چه کتابی را به امانت برده‌است، مقسوم است و جدول Book با شرط «کلیه‌ی کتاب‌های منتشرشده توسط انتشارات McGraw-Hill» مقسوم علیه است. گزینه اول نادرست است. زیرا، ستون Publisher در سمت مقسوم علیه وجود دارد، اما در سمت مقسوم وجود ندارد. بنابراین اجرای عملگر تقسیم در گزینه اول، به دلیل وجود خطای نحوی امکان‌پذیر نیست.

$$\Pi(\text{name}) \left(\left(\Pi(\text{isbn}, \text{name}, \text{mem-no}) \text{Borrowed} \bowtie \text{Member} \right) \div \sigma_{\text{publisher} = \text{'McGraw-Hill' Book}} \right)$$

گزینه دوم نادرست است. زیرا به دلیل مشابه گزینه اول، ستون Publisher در سمت مقسوم علیه وجود دارد، اما در سمت مقسوم وجود ندارد. بنابراین اجرای عملگر تقسیم در گزینه دوم نیز، به دلیل وجود خطای نحوی امکان‌پذیر نیست.

$$\Pi(\text{name}) \left(\left(\text{Borrowed} \div \sigma_{\text{publisher} = \text{'McGraw-Hill' Book}} \right) \bowtie \text{Member} \right)$$

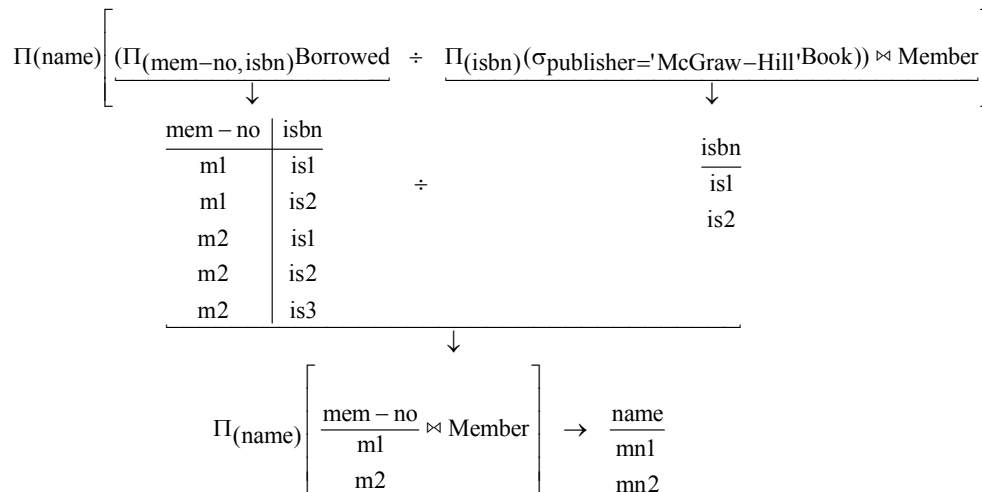
گزینه چهارم نادرست است. زیرا، ستون isbn در سمت مقسوم علیه وجود دارد، اما در سمت مقسوم وجود ندارد. بنابراین اجرای عملگر تقسیم در گزینه چهارم نیز، به دلیل وجود خطای نحوی امکان‌پذیر نیست. دقت کنید که در پرس و جوی گزینه چهارم ابتدا عبارت زیر انجام می‌شود:

$$\Pi(\text{name}) (\text{Borrowed} \bowtie \text{Member})$$

که باعث می‌شود مقسوم، فاقد ستون isbn گردد.

$$\Pi(\text{name}) (\text{Borrowed} \bowtie \text{Member}) \div \Pi(\text{isbn}) \left(\sigma_{\text{publisher} = \text{'McGraw-Hill' Book}} \right)$$

گزینه سوم درست است. مطابق تعریف عملگر تقسیم در جبر رابطه‌ای، اجرای عملگر تقسیم در گزینه سوم امکان‌پذیر است.



$\Pi(\text{name}) \left(\left(\Pi(\text{mem-no, isbn})\text{Borrowed} \div \Pi(\text{isbn})(\sigma_{\text{publisher}='McGraw-Hill'}\text{Book}) \right) \bowtie \text{Member} \right)$

توجه: مشاهده می‌شود که اعضای mn1 و mn2، همه کتاب‌های منتشرشده توسط انتشارات McGraw-Hill را به امانت برده‌اند.

۵۹- گزینه (۴) صحیح است.

جداول زیر را در نظر بگیرید:

MID	Name	Mdate	MID	ISBN	BDate	...	ISBN	Title	...	publisher
m1	mn1	21	m1	is1	11		is1	tit1		McGraw-Hill
m2	mn2	22	m1	is2	12		is2	tit2		McGraw-Hill
m3	mn3	23	m2	is1	13		is3	tit3		Wiley
Member			m2	is2	۱۴		is4	tit4		Wiley
			m2	is3	۱۵		Book			
			m3	is1	۱۶					
Borrow										

مطابق پرس و جوی مطرح شده در صورت سوال داریم:

«عنوان کتاب‌هایی از انتشارات wiley که توسط هیچ عضوی امانت گرفته نشده»

پرس و جوی فوق توسط عملگر تفاضل در جبر رابطه‌ای قابل پیاده‌سازی است.

مطابق تعریف عملگر تفاضل در جبر رابطه‌ای، تفاضل هر دو رابطه دلخواه امکان‌پذیر نیست. مگر اینکه شروط سازگاری در مورد آنها برقرار باشد. در جبر رابطه‌ای دو شرط به عنوان سازگاری مطرح است:

شرط اول: تعداد ستون‌های دو جدول یکسان باشد، به عبارت دیگر دو رابطه (جدول) هم درجه باشند.

شرط دوم: نوع یا دامنه ستون‌های متناظر در دو جدول یکسان باشد.
 اگر بخواهیم دو شرط فوق را در یک جمله بیان کنیم، اینطور خواهد بود، شروط سازگاری یعنی تیتراها در دو رابطه (جدول) یکسان باشد.
 به طور کلی، اگر عبارت نفی در پرس و جو استفاده شده باشد، این امر نشانه‌ای برای ضرورت استفاده از عملگر تفاضل است. دقت کنید که در انجام عمل تفاضل برقراری شروط سازگاری الزامی است که در پرس و جوی گزینه چهارم این الزام برقرار است.
 جدول Borrow می‌گوید هر عضوی چه کتابی را به امانت برده‌است. و جدول Book با شرط انتشارات wiley «کلیه‌ی کتاب‌های منتشرشده توسط انتشارات wiley» را بیان می‌کند.
 گزینه اول نادرست است. زیرا، شروط سازگاری در عملگر تفاضل برقرار نیست. بنابراین اجرای عملگر تفاضل در گزینه اول، به دلیل وجود خطای نحوی امکان‌پذیر نیست.

$$\Pi_{\langle \text{Title} \rangle} (\sigma_{\text{publisher} = 'wiley'}(\text{Book})) - \Pi_{\langle \text{ISBN} \rangle}(\text{Borrow})$$

گزینه دوم نادرست است. زیرا، از عملگر تفاضل یعنی عبارت نفی در پرس و جو استفاده نشده است.

$$\Pi_{\langle \text{Title}, \text{ISBN} \rangle} (\sigma_{\text{publisher} = 'wiley'}(\text{Book})) \div \Pi_{\langle \text{ISBN} \rangle}(\text{Borrow})$$

گزینه سوم نادرست است. زیرا، از عملگر تفاضل یعنی عبارت نفی در پرس و جو استفاده نشده است. مطابق تعریف عملگر تقسیم در جبر رابطه‌ای، باید تمام خصیصه‌های مقسوم علیه در مقسوم باشد. به عبارت دیگر مجموعه صفات مقسوم علیه، زیر مجموعه، مجموعه صفات مقسوم باشد. همچنین ستون‌های حاصل از عملگر تقسیم برابر تفاضل ستون‌های مقسوم و مقسوم علیه است. همچنین سطرهایی از جدول r در خروجی ظاهر می‌گردد که تمام اعضای جدول s را در کنار خود داشته باشد.

مثال:

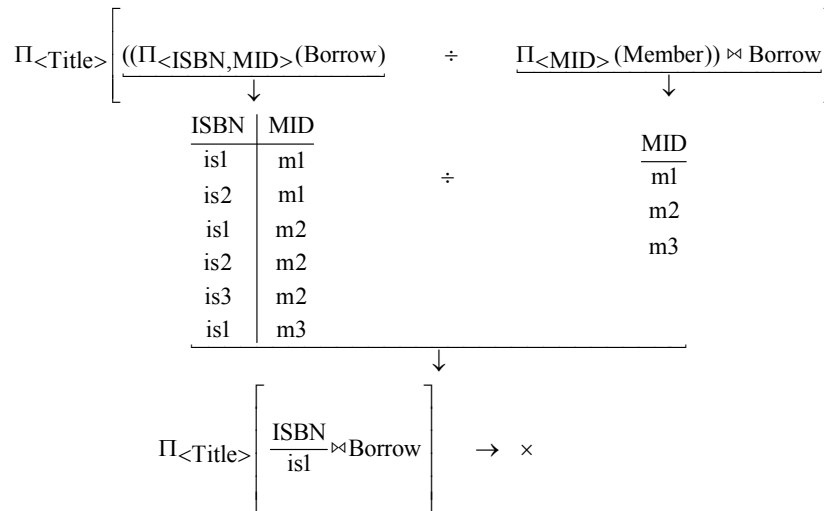
a	b	÷	$\frac{b}{1} = \frac{a}{5}$
5	7		2
5	1		جدول s
4	2		
4	3		
5	2		

جدول r

مطابق پرس و جوی مطرح شده در گزینه چهارم داریم:

$$\Pi_{\langle \text{Title} \rangle} \left(\left(\Pi_{\langle \text{ISBN}, \text{MID} \rangle}(\text{Borrow}) \div \Pi_{\langle \text{MID} \rangle}(\text{Member}) \right) \bowtie \text{Borrow} \right)$$

جدول Borrow که می‌گوید هر عضوی چه کتابی را به امانت برده‌است، مقسوم است و جدول Member که اعضاء را نشان می‌دهد مقسوم علیه است. مطابق تعریف عملگر تقسیم در جبر رابطه‌ای، اجرای عملگر تقسیم در گزینه سوم امکان‌پذیر است.



توجه: مشاهده می‌شود که ISBN استخراج شده یعنی is1، توسط همه اعضا به امانت برده شده است. یعنی پرس و جوی داخلی، ISBN کتبی را استخراج می‌کند که توسط همه اعضا به امانت برده شده است. اما در ادامه و حرکت به سمت خارج و اجرای عملگر الحاق طبیعی، ستون Title استخراج نمی‌شود، بنابراین اجرای عملگر پرتو در گزینه سوم، به دلیل وجود خطای نحوی امکان‌پذیر نیست. گزینه چهارم درست است. زیرا، مطابق تعریف عملگر تفاضل در جبر رابطه‌ای، اجرای عملگر تفاضل در گزینه چهارم امکان‌پذیر است. مطابق پرس و جوی مطرح شده در گزینه چهارم داریم:

$$\Pi_{\langle \text{Title} \rangle} \left(\left(\Pi_{\langle \text{ISBN} \rangle} (\sigma_{\text{publisher} = \text{'wiley'}} (\text{Book})) - \Pi_{\langle \text{ISBN} \rangle} (\text{Borrow}) \right) \bowtie \text{Book} \right)$$

در پرس و جوی سمت چپ عملگر تفاضل یعنی در جدول Book مشخصات کلیه کتاب‌های منتشرشده قرار دارد، که پس از اجرای عملگر انتخاب مشخصات کلیه کتاب‌های منتشر شده توسط انتشارات wiley استخراج می‌گردد، در نهایت پس از اجرای عملگر پرتو، ISBN کلیه کتاب‌های منتشر شده توسط انتشارات wiley استخراج می‌شود، به صورت زیر:

ISBN
is3
is4

همچنین در پرس و جوی سمت راست عملگر تفاضل یعنی در جدول Borrow مشخصات کلیه کتاب‌هایی که تا کنون به امانت گرفته شده‌اند قرار دارد، که در نهایت پس از اجرای عملگر پرتو، ISBN کلیه کتاب‌هایی که تا کنون به امانت گرفته شده‌اند استخراج می‌شود، به صورت زیر:

ISBN
is1
is2
is3

در واقع در پرس و جوی فوق کلیه کتاب‌های منتشرشده توسط انتشارات Wiley منهای کلیه کتاب‌های به امانت گرفته شده تاکنون شده است.

که در قدم بعدی پس از اجرای عملگر تفاضل، خروجی پرس و جوی داخلی به صورت زیر خواهد بود:

$$\frac{\text{ISBN}}{\text{is3}} \quad - \quad \frac{\text{ISBN}}{\text{is1}} \quad = \quad \frac{\text{ISBN}}{\text{is4}}$$

$$\text{is4} \qquad \qquad \text{is2} \qquad \qquad \text{is3}$$

همچنین از آنجاکه در پرس و جوی مطرح شده در صورت سوال، عنوان کتاب‌هایی از انتشارات wiley که توسط هیچ عضوی امانت گرفته نشده درخواست شده است، بنابراین در نهایت پس از اجرای عملگر الحاق طبیعی، خروجی نهایی پرس و جوی به صورت زیر خواهد بود:

$$\Pi \langle \text{Title} \rangle \left[\left(\frac{\text{ISBN}}{\text{is3}} \right) - \left(\frac{\text{ISBN}}{\text{is1}} \right) \right] \bowtie \text{Book}$$

$$\frac{\text{ISBN}}{\text{is3}} \quad - \quad \frac{\text{ISBN}}{\text{is1}}$$

$$\text{is4} \qquad \qquad \text{is2}$$

$$\text{is3}$$

$$\Pi \langle \text{Title} \rangle \left[\frac{\text{ISBN}}{\text{is4}} \right] \bowtie \text{Book} \rightarrow \frac{\text{Title}}{\text{tit4}}$$

$$\Pi(\text{name}) \left(\left(\Pi(\text{mem-no, isbn}) \text{Borrowed} \div \Pi(\text{isbn}) (\sigma_{\text{publisher} = \text{'McGraw-Hill'}} \text{Book}) \right) \bowtie \text{Member} \right)$$

توجه: مشاهده می‌شود که عنوان tit4، از انتشارات wiley توسط هیچ عضوی امانت گرفته نشده است.

۶۰- گزینه (۲) صحیح است.

عملگر ضرب دکارتی یا ضرب کارتیزین توسط نماد \times نمایش داده می‌شود. عملگر \times یک عملگر اصلی است. عملگر ضرب دکارتی گاهی با نماد Times نیز نشان داده می‌شود. عملگر \times جهت اتصال کلیه سطرهای دو جدول که در هم ضرب می‌شود، مورد استفاده قرار می‌گیرد. در جبر رابطه‌ای ضرب دکارتی هر دو رابطه دلخواه امکان‌پذیر است و شرط خاصی برای ضرب ندارد.

کلید کاندید رابطه مقصد حاصل از عملگر \times مابین دو رابطه R_1 و R_2 همواره برابر هر ترکیبی از کلیدهای کاندید روابط مبدا یعنی هر ترکیبی از یک کلید کاندید R_1 و یک کلید کاندید R_2 می‌باشد، به صورت زیر:

$$C.K.(R_3) = C.K.(R_1) \cup C.K.(R_2)$$

مثال: جداول R_1 و R_2 را به صورت زیر در نظر بگیرید:

a	b
1	2
3	2

R_1

c	d	e
5	6	7
8	9	10
11	12	13
14	6	7

R_2

$$R_3 = R_1 \times R_2$$

پرس و جوی مقابل را در نظر بگیرید:

خروجی پرس و جوی فوق به صورت زیر است:

<u>a</u>	b	×	<u>c</u>	d	e	=	<u>a</u>	b	<u>c</u>	d	e
1	2		5	6	7		1	2	5	6	7
3	2		8	9	10		1	2	8	9	10
			11	12	13		1	2	11	12	13
			14	6	7		1	2	14	6	7
							3	2	5	6	7
							3	2	8	9	10
							3	2	11	12	13
							3	2	14	6	7

از آنجا که کلید کاندید رابطه مقصد حاصل از عملگر \times مابین دو رابطه R_1 و R_2 همواره برابر هر ترکیبی از کلیدهای کاندید روابط مبدا یعنی هر ترکیبی از یک کلید کاندید R_1 و یک کلید کاندید R_2 می‌باشد، پس کلید کاندید رابطه مقصد یعنی R_3 برابر ترکیبی از یک کلید کاندید R_1 و یک کلید کاندید R_2 است، به صورت زیر:

$$C.K.(R_3) = \{a\} \cup \{c\} = \{a, c\}$$

$$C.K.(R_3) = \{a\} \times \{c\} = \{a, c\}$$

بنابراین رابطه زیر برقرار خواهد بود:

$$C.K.(R_3) = C.K.(R_1) \cup C.K.(R_2)$$

$$C.K.(R_3) = C.K.(R_1) \times C.K.(R_2)$$

مثال: جداول R_1 و R_2 را به صورت زیر در نظر بگیرید:

<u>a</u>	b	c		<u>d</u>	e	f	g
1	1	2		5	5	6	7
3	3	2		8	8	9	10
1	3	2		11	11	12	13
	R ₁			5	8	6	7
					R ₂		

$$R_3 = R_1 \times R_2$$

پرس و جوی مقابل را در نظر بگیرید:

خروجی پرس و جوی فوق به صورت زیر است:

<u>a</u>	b	c	×	<u>d</u>	e	f	g	=	<u>a</u>	b	c	d	e	f	g
1	1	2		5	5	6	7		1	1	2	5	5	6	7
3	3	2		8	8	9	10		1	1	2	8	8	9	10
1	3	2		11	11	12	13		1	1	2	11	11	12	13
				5	8	6	7		1	1	2	5	8	6	7
									3	3	2	5	5	6	7
									3	3	2	8	8	9	10
									3	3	2	11	11	12	13
									3	3	2	5	8	6	7
									1	3	2	5	5	6	7
									1	3	2	8	8	9	10
									1	3	2	11	11	12	13
									1	3	2	5	8	6	7

از آنجا که کلید کاندید رابطه مقصد حاصل از عملگر \times مابین دو رابطه R_1 و R_2 همواره برابر هر ترکیبی از کلیدهای کاندید روابط مبدا یعنی هر ترکیبی از یک کلید کاندید R_1 و یک کلید کاندید R_2 می‌باشد، پس کلید کاندید رابطه مقصد یعنی R_3 برابر ترکیبی از یک کلید کاندید R_1 و یک کلید کاندید R_2 است، به صورت زیر:

$$C.K.(R_3) = \{a, b\} \cup \{d, e\} = \{a, b, d, e\}$$

$$C.K.(R_3) \neq \{a, b\} \cap \{d, e\} = \{ \}$$

$$C.K.(R_3) \neq \{a, b\} \times \{d, e\} = \{(a, d), (a, e), (b, d), (b, e)\}$$

واضح است که در حالت کلی اگر جداول مبدا هیچ صفت مشترکی نداشته باشند، آنگاه کلید کاندید حاصل از ضرب دکارتی برابر اجتماع دو کلید جدول مبدا است و نه برابر اشتراک یا ضرب دکارتی دو کلید جدول مبدا. بنابراین گزینه اول، سوم و چهارم نادرست و گزینه دوم درست است.

$$C.K.(R_3) = C.K.(R_1) \cup C.K.(R_2)$$

بنابراین رابطه مقابل به طور کلی برقرار خواهد بود:

دقت کنید که در صورت سوال بیان شده است که جداول مبدا هیچ صفت مشترکی نداشته باشند، و این فرض در صورت سوال در نظر گرفته شده است، حال سوال این است که اگر کلیدهای کاندید جداول مبدا صفت مشترک داشته باشند، آنگاه کلید کاندید حاصل از ضرب دکارتی جداول مبدا به چه صورت است؟ پاسخ این است که صفات مشترک میان کلیدهای کاندید جداول مبدا، می‌بایست در کلید کاندید حاصل از ضرب دکارتی جداول مبدا، دوبار تکرار شوند.

مثال: جداول R_1 و R_2 را به صورت زیر در نظر بگیرید:

a	b
1	2
3	2

R_1

a	c	d
5	6	7
8	9	10
11	12	13
14	6	7

R_2

$$R_3 = R_1 \times R_2$$

پرس و جوی مقابل را در نظر بگیرید:

خروجی پرس و جوی فوق به صورت زیر است:

a	b	×	a	c	d	=	R ₁ .a	b	R ₂ .a	c	d
1	2		5	6	7		1	2	5	6	7
3	2		8	9	10		1	2	8	9	10
			11	12	13		1	2	11	12	13
			14	6	7		1	2	14	6	7
							3	2	5	6	7
							3	2	8	9	10
							3	2	11	12	13
							3	2	14	6	7

از آنجا که کلید کاندید رابطه مقصد حاصل از عملگر \times مابین دو رابطه R_1 و R_2 همواره برابر هر ترکیبی از کلیدهای کاندید روابط مبدا یعنی هر ترکیبی از یک کلید کاندید R_1 و یک کلید کاندید R_2 می‌باشد،

پس کلید کاندید رابطه مقصد یعنی R_3 برابر ترکیبی از یک کلید کاندید R_1 و یک کلید کاندید R_2 است، به صورت زیر:

$$C.K.(R_3) = \{R_1.a\} \cup \{R_2.a\} = \{R_1.a, R_2.a\}$$

بنابراین رابطه زیر برقرار خواهد بود:

$$C.K.(R_3) = C.K.(R_1) \cup C.K.(R_2)$$

۶۱- گزینه (۳) صحیح است.

جداول زیر را در نظر بگیرید:

ID	Name	age	ID	ISBN	Date	ISBN	Title	publisher
m1	mn1	21	m1	is1	11	is1	tit1	McGraw-Hill
m2	mn2	22	m1	is2	12	is2	tit2	McGraw-Hill
Member	mn3	23	m2	is1	13	is3	tit3	Wiley
			m2	is2	14	is4	tit4	Wiley
			m2	is3	15			
			m3	is1	16			
								Book

Borrowed

مطابق پرس و جوی مطرح شده در صورت سوال داریم:

«نام اعضای را بیابید که همه کتاب‌های منتشر شده توسط McGraw-Hill را به امانت برده‌اند.»

پرس و جوی فوق توسط عملگر تقسیم در جبر رابطه‌ای قابل پیاده‌سازی است.

مطابق تعریف عملگر تقسیم در جبر رابطه‌ای، باید تمام خصیصه‌های مقسوم علیه در مقسوم باشد. به عبارت دیگر مجموعه صفات مقسوم علیه، زیر مجموعه، مجموعه صفات مقسوم باشد. همچنین ستون‌های حاصل از عملگر تقسیم برابر تفاضل ستون‌های مقسوم و مقسوم علیه است. همچنین سطرهایی از جدول r در خروجی ظاهر می‌گردد که تمام اعضای جدول s را در کنار خود داشته باشد.

مثال:

a	b	$\div \frac{b}{1} = \frac{a}{5}$
5	7	2
5	1	
4	2	جدول s
4	3	
5	2	

جدول r

جدول Borrowed که می‌گوید هر عضوی چه کتابی را به امانت برده‌است، مقسوم است و جدول Book با شرط «کلیه کتاب‌های منتشر شده توسط انتشارات McGraw-Hill» مقسوم علیه است.

گزینه اول نادرست است. زیرا، ستون Publisher در سمت مقسوم علیه وجود دارد، اما در سمت مقسوم وجود ندارد. بنابراین اجرای عملگر تقسیم در گزینه اول، به دلیل وجود خطای نحوی امکان‌پذیر نیست.

$$\Pi_{\langle Name \rangle} \left[Member \bowtie \Pi_{\langle ID \rangle} \left(Borrowed \div \left(\sigma_{publisher = 'McGraw-Hill'} Book \right) \right) \right]$$

گزینه دوم نادرست است. زیرا به دلیل مشابه گزینه اول، ستون Publisher در سمت مقسوم علیه وجود دارد، اما در سمت مقسوم وجود ندارد. بنابراین اجرای عملگر تقسیم در گزینه دوم نیز، به دلیل وجود خطای نحوی امکان‌پذیر نیست.

$$\Pi_{\langle \text{Name} \rangle} \left[\text{Member} \times \Pi_{\langle \text{ID} \rangle} \left(\text{Borrowed} \div \left(\sigma_{\text{publisher} = \text{'McGraw-Hill' Book}} \right) \right) \right]$$

گزینه سوم درست است. مطابق تعریف عملگر تقسیم در جبر رابطه‌ای، اجرای عملگر تقسیم در گزینه سوم امکان‌پذیر است.

$$\Pi_{\langle \text{Name} \rangle} \left[\frac{\text{Member} \bowtie (\Pi_{\langle \text{ID, ISBN} \rangle} \text{Borrowed})}{\Pi_{\langle \text{ISBN} \rangle} (\sigma_{\text{publisher} = \text{'McGraw-Hill' Book}})} \right]$$

ID	ISBN	÷	ISBN
m1	is1		is1
m1	is2		is2
m2	is1		
m2	is2		
m2	is3		
m3	is1		

$$\Pi_{\langle \text{Name} \rangle} \left[\frac{\text{ID} \bowtie \text{Member}}{\text{m1}} \right] \rightarrow \frac{\text{Name}}{\text{mn1}}$$

$$\Pi_{\langle \text{Name} \rangle} \left[\text{Member} \bowtie \left(\left(\Pi_{\langle \text{ID, ISBN} \rangle} \text{Borrowed} \right) \div \left(\Pi_{\langle \text{ISBN} \rangle} \left(\sigma_{\text{publisher} = \text{'McGraw-Hill' Book}} \right) \right) \right) \right]$$

توجه: مشاهده می‌شود که اعضای mn1 و mn2، همه کتاب‌های منتشرشده توسط انتشارات McGraw-Hill را به امانت برده‌اند.

گزینه چهارم نادرست است. زیرا، تا بخشی همانند گزینه سوم است اما در نهایت از عملگر ضرب دکارتی استفاده می‌کند که باعث می‌شود سطرهای پیوندپذیر یعنی معتبر و هم سطرهای پیوندناپذیر یعنی نامعتبر در خروجی قرار گیرد.

$$\Pi_{\langle \text{Name} \rangle} \left[\frac{\text{ID} \times \text{Member}}{\text{m1}} \right] \rightarrow \frac{\text{Name}}{\text{mn1}}$$

توجه: مشاهده می‌شود که اعضای mn1 و mn2، همه کتاب‌های منتشرشده توسط انتشارات McGraw-Hill را به امانت برده‌اند. اما عضو mn3 نیز که همه کتاب‌های منتشرشده توسط انتشارات McGraw-Hill را به امانت نبرده‌است نیز در خروجی قرار گرفته است.

۶۲ - گزینه (۲) صحیح است.

دو جدول Node و Edge با مقادیر زیر را در نظر بگیرید:

<u>NID</u>	Name	Color	Description	<u>NID1</u>	<u>NID2</u>	EdgeType
N1	NN1	C1	D1	N1	N2	T1
N2	NN2	C2	D2	N1	N3	T2
N3	NN3	C3	D3	N4	N1	T3
N4	NN4	C4	D4	N3	N2	T4
				N4	N3	T5

جدول Node

جدول Edge

توجه: کلید کاندید جدول Node ستون NID است و جدول Edge دارای دو کلید خارجی NID1 و NID2 است که هر دو به کلید کاندید جدول Node ارجاع می‌کنند.

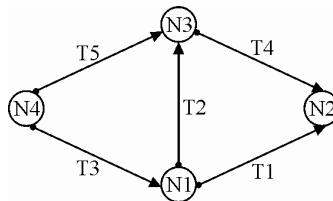
توجه: ستون NID1 در جدول Edge به عنوان کلید خارجی به ستون NID از جدول Node ارجاع می‌کند.

توجه: ستون NID2 در جدول Edge به عنوان کلید خارجی به ستون NID از جدول Node ارجاع می‌کند.

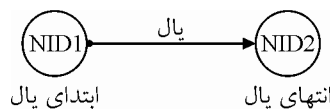
توجه: مقادیر کلید خارجی یعنی ستون‌های NID1 و NID2 از جدول Edge همواره باید زیرمجموعه مقادیر کلید کاندید یعنی ستون NID از جدول Node باشد.

توجه: هر تاپل یا سطر یا رکورد از جدول Edge نشانه یک ارتباط است، پس هر تاپل نشانه یک یال هم هست، که مجموع این یال‌ها یک گراف را ایجاد می‌کند. خطی که از یک گره به گره بعدی رسم می‌شود یک یال است. هر گراف G شامل دو مجموعه V و E است. V مجموعه محدود و غیرتهی از رئوس است و E مجموعه‌ای محدود و احتمالاً تهی از لبه‌ها می‌باشد. $V(G)$ و $E(G)$ مجموعه رئوس و لبه‌های گراف G را نمایش می‌دهند. برای نمایش یک گراف می‌توانیم بنویسیم $G=(V, E)$. در یک گراف بدون جهت، زوج رئوس، زوج مرتب نیستند. بنابراین زوج‌های (V_0, V_1) و (V_1, V_0) باهم یکسان هستند. گراف حداقل یک راس دارد و نمی‌تواند کاملاً تهی باشد. در یک گراف جهت دار، هر لبه با زوج مرتب $\langle V_0, V_1 \rangle$ نمایش داده می‌شود، که پیکانی از V_0 به V_1 ترسیم می‌شود. بنابراین $\langle V_1, V_0 \rangle$ و $\langle V_0, V_1 \rangle$ دو لبه متفاوت را نمایش می‌دهند. گراف جهت دار را به صورت $G=\langle V, E \rangle$ نمایش می‌دهیم.

توجه: گراف جهت دار جدول Edge به صورت زیر است.



گراف جدول Edge



هر تاپل از جدول Edge نشانه یک یال از گراف جدول Edge است.

توجه: همانطور که واضح است گره‌های موجود در ستون NID1 از جدول Edge یا گراف Edge، یال‌های خروجی دارند.

$$NID1 = \{N1, N3, N4\}$$

یعنی از گره‌های $N1$ ، $N3$ و $N4$ یال خارج شده است. توجه: همانطور که واضح است گره‌های موجود در ستون $NID2$ از جدول $Edge$ یا گراف $Edge$ ، یال‌های ورودی دارند.

$$NID2 = \{N1, N2, N3\}$$

یعنی به گره‌های $N1$ ، $N2$ و $N3$ یال وارد شده است.

نتیجه:

استخراج گره‌هایی که هم یالی از آن خارج شده است و هم یالی به آن وارد شده است، به صورت زیر است:

$$NID1 \cap NID2 = \{N1, N3, N4\} \cap \{N1, N2, N3\} = \{N1, N3\}$$

استخراج گره‌هایی که یالی از آن خارج شده است ولی یالی به آن وارد نشده است، به صورت زیر است:

$$NID1 - NID2 = \{N1, N3, N4\} - \{N1, N2, N3\} = \{N4\}$$

استخراج گره‌هایی که یالی به آن وارد شده است ولی یالی از آن خارج نشده است، به صورت زیر است:

$$NID2 - NID1 = \{N1, N2, N3\} - \{N1, N3, N4\} = \{N2\}$$

مطابق پرس و جوی مطرح شده در صورت سوال، داریم:

$$\Pi_{E1.NID1} \left(\sigma_{E1.NID2 = E2.NID1} \left(\sigma_{E1.EdgeType = 'T2'} (\rho_{E1}(Edge)) \times \rho_{E2}(Edge) \right) \right)$$

ابتدا در داخلی‌ترین پراتز، عملوند سمت چپ عملگر ضرب دکارتی اجرا می‌شود، که خروجی آن بر اساس عملگر ρ_{E1} به جدول $E1$ نام‌گذاری می‌شود، به صورت زیر:

$$\sigma_{E1.EdgeType = 'T2'} (\rho_{E1}(Edge))$$

معادل پرس و جوی جبر رابطه‌ای فوق به زبان SQL به صورت زیر است:

```
(select *
from edge
where EdgeType = 'T2') E1
```

خروجی قطعه کد فوق به صورت زیر است:

NID1	NID2	EdgeType
N1	N3	T2

جدول E1

توجه: قطعه پرس و جوی فوق یال‌های نوع $T2$ را استخراج می‌کند. در ادامه در داخلی‌ترین پراتز، عملوند سمت راست عملگر ضرب دکارتی اجرا می‌شود، که خروجی آن بر اساس عملگر ρ_{E2} به جدول $E2$ نام‌گذاری می‌شود، به صورت زیر:

$$\rho_{E2}(Edge)$$

معادل پرس و جوی جبر رابطه‌ای فوق به زبان SQL به صورت زیر است:

```
(select *
from edge) E2
```

خروجی قطعه کد فوق به صورت زیر است:

NID1	NID2	EdgeType
N1	N2	T1
N1	N3	T2
N4	N1	T3
N3	N2	T4
N4	N3	T5

جدول E2

در ادامه در داخلی ترین پراتز، عملوند سمت چپ عملگر ضرب دکارتی (T1) و عملوند سمت راست عملگر ضرب دکارتی (T2) در هم ضرب دکارتی می شوند، که خروجی آن بر اساس عملگر $\sigma_{E1.NID2 = E2.NID1}$ ، به صورت زیر است:

$$\sigma_{E1.NID2 = E2.NID1} \left(\sigma_{E1.EdgeType = T2} (\rho_{E1}(Edge)) \times \rho_{E2}(Edge) \right)$$

معادل پرس و جوی جبر رابطه ای فوق به زبان SQL به صورت زیر است:

```
select *
from (select *
      from edge
      where EdgeType= 'T2') E1 ,
(select *
  from edge) E2
where E1.NID2 = E2.NID1
```

خروجی قطعه کد فوق به صورت زیر است:

E1.NID1	E1.NID2	E1.EdgeType	E2.NID1	E2.NID2	E2.EdgeType
N1	N3	T2	N1	N2	T1
N1	N3	T2	N1	N3	T2
N1	N3	T2	N4	N1	T3
N1	N3	T2	N3	N2	T4
N1	N3	T2	N4	N3	T5

که خروجی آن پس از اجرای عملگر $\sigma_{E1.NID2 = E2.NID1}$ ، به صورت زیر است:

E1.NID1	E1.NID2	E1.EdgeType	E2.NID1	E2.NID2	E2.EdgeType
N1	N3	T2	N3	N2	T4

در نهایت و حرکت به سمت خارجی ترین پراتز، عملگر Project اجرا می شود، که خروجی آن بر اساس عملگر $\Pi_{E1.NID1}$ ، به صورت زیر است:

$$\Pi_{E1.NID1} \left(\sigma_{E1.NID2 = E2.NID1} \left(\sigma_{E1.EdgeType = T2} (\rho_{E1}(Edge)) \times \rho_{E2}(Edge) \right) \right)$$

معادل پرس و جوی جبر رابطه ای فوق به زبان SQL به صورت زیر است:

```

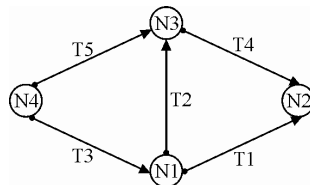
select E1.NID1
from (select *
      from edge
      where EdgeType='T2') E1 ,
      (select *
      from edge) E2
where E1.NID2 = E2.NID1

```

که خروجی نهایی آن پس از اجرای عملگر Π E1.NID1، به صورت زیر است:

$$\frac{E1.NID1}{N1}$$

یعنی «شماره گره‌هایی که حداقل یک یال خروجی از نوع T2 به یک گره مانند g دارند و گره g حداقل یک یال خروجی دارد.» بنابراین پُر واضح است که گزینه‌ی دوم پاسخ سوال است. شکل زیر گویای مطلب است:



گراف جدول Edge

توجه: سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه دوم را به عنوان پاسخ اعلام کرده بود.

۶۳- گزینه (۳) صحیح است.

صورت سوال به صورت زیر است:

با توجه به جدول ارائه شده، کدام عبارت جبری درست است؟

STUD(S#, Sname, City, avg, clg#)

(شماره دانشکده، معدل، محل تولد، نام، شماره دانشجویی) جدول دانشجو

$$\Pi_{City}(\sigma_{avg>16}(STUD)) = \sigma_{avg>16}(\Pi_{City}(STUD)) \quad (۱)$$

گزینه اول نادرست است، زیرا دو عملگر σ و Π به صورت مشروط دارای خاصیت جابه‌جایی هستند. به طور کلی اگر R یک رابطه، L زیر مجموعه‌ای از ستون‌ها و θ مجموعه‌ای از شروط بر روی سطرها باشد، آنگاه تساوی زیر زمانی برقرار است که ستون‌های عملگر σ زیر مجموعه ستون‌های عملگر Π باشد. یعنی

$$\sigma \subseteq \Pi$$

$$\Pi_L(\sigma_\theta(R)) = \sigma_\theta(\Pi_L(R))$$

مثال: جدول STUD را به صورت زیر در نظر بگیرید:

S#	Sname	City	avg	clg#
S1	Sn1	C1	14	clg1
S2	Sn2	C2	17	clg2
S3	Sn3	C3	18	clg3

پرس و جوی زیر را در نظر بگیرید:

$$\Pi_{\text{City}}(\sigma_{\text{avg}>16}(\text{STUD}))$$

خروجی پرس و جوی پرائنتز داخلی به صورت زیر است:

S#	Sname	City	avg	clg#
S2	Sn2	C2	17	Clg2
S3	Sn3	C3	18	Clg3

که در نهایت پس از حرکت به سمت خارج و انجام عملگر پرتو، خروجی نهایی پرس و جوی به صورت زیر خواهد بود:

City
C2
C3

همچنین پرس و جوی زیر را در نظر بگیرید:

$$\sigma_{\text{avg}>16}(\Pi_{\text{City}}(\text{STUD}))$$

خروجی پرس و جوی پرائنتز داخلی به صورت زیر است:

City
C1
C2
C3

که در نهایت پس از حرکت به سمت خارج، انجام عملگر انتخاب امکان پذیر نخواهد بود، زیرا ستون‌های عملگر σ یعنی avg زیر مجموعه ستون‌های عملگر Π یعنی City نیست، یعنی $\text{avg} \notin \text{City}$. در واقع ستون avg در پرائنتز داخلی توسط عملگر Π انتخاب نشده است، بنابراین اجرای عملگر σ بر روی ستون avg امکان پذیر نیست.

پس رابطه زیر برقرار است:

$$\Pi_{\text{City}}(\sigma_{\text{avg}>16}(\text{STUD})) \neq \sigma_{\text{avg}>16}(\Pi_{\text{City}}(\text{STUD}))$$

بنابراین همانطور که واضح است، گزینه اول نادرست است.

مثال: جدول STUD را به صورت زیر در نظر بگیرید:

S#	Sname	City	avg	clg#
S1	Sn1	C1	14	clg1
S2	Sn2	C2	17	clg2
S3	Sn3	C3	18	clg3

پرس و جوی زیر را در نظر بگیرید:

$$\Pi_{\text{City,avg}}(\sigma_{\text{avg}>16}(\text{STUD}))$$

خروجی پرس و جوی پرائنتز داخلی به صورت زیر است:

S#	Sname	City	avg	clg#
S2	Sn2	C2	17	clg2
S3	Sn3	C3	18	clg3

که در نهایت پس از حرکت به سمت خارج و انجام عملگر پرتو، خروجی نهایی پرس و جو به صورت زیر خواهد بود:

City	avg
C2	17
C3	18

همچنین پرس و جوی زیر را در نظر بگیرید:

$$\sigma_{avg>16}(\Pi_{City,avg}(STUD))$$

خروجی پرس و جوی پرائتز داخلی به صورت زیر است:

City	avg
C1	14
C2	17
C3	18

که در نهایت پس از حرکت به سمت خارج و انجام عملگر پرتو، خروجی نهایی پرس و جو به صورت زیر خواهد بود:

City	avg
C2	17
C3	18

پس رابطه زیر برقرار است:

$$\Pi_{City,avg}(\sigma_{avg>16}(STUD)) = \sigma_{avg>16}(\Pi_{City,avg}(STUD))$$

پس همانطور که گفتیم دو عملگر σ و Π به صورت مشروط دارای خاصیت جابه‌جایی هستند.

$$\sigma_{City}(\Pi_{avg>16}(STUD)) = \Pi_{avg>16}(\sigma_{City}(STUD)) \quad (۲)$$

گزینه دوم نادرست است، زیرا از نظر ساختار پرس و جو نویسی در جبر رابطه‌ای خطای نحوی دارد و پرس و جو اصلاً قابلیت اجرا ندارد. عملگر σ جهت انتخاب سطر در یک جدول مورد استفاده قرار می‌گیرد.

فرم کلی عملگر σ به صورت زیر است:

$$R_2 = \sigma_{\theta}(R_1)$$

در بخش θ شرط انتخاب سطر به عنوان ملاک انتخاب سطر مشخص می‌گردد. همچنین در این بخش علائم ریاضی $=, >, \geq, <, \leq,$ و عملگرهای منطقی \neg, \vee, \wedge می‌تواند مورد استفاده قرار گیرد. عبارت σ_{City} در گزینه دوم نادرست است و خطای نحوی دارد.

عملگر Π جهت انتخاب ستون در یک جدول مورد استفاده قرار می‌گیرد.

فرم کلی عملگر Π به صورت زیر است:

$$R_2 = \Pi_L(R_1)$$

در بخش L مدل انتخاب ستون به عنوان ستون‌های مورد نیاز مشخص می‌گردد. عبارت $\Pi_{avg>16}$ در گزینه دوم نادرست است و خطای نحوی دارد.

$$\Pi_{City}(\sigma_{avg>16}(STUD)) \quad (3)$$

گزینه سوم درست است.

پرس و جوی زیر را در نظر بگیرید:

$$\Pi_{City}(\sigma_{avg>16}(STUD))$$

خروجی پرس و جوی پرائتز داخلی به صورت زیر است:

S#	Sname	City	avg	clg#
S2	Sn2	C2	17	clg2
S3	Sn3	C3	18	clg3

که در نهایت پس از حرکت به سمت خارج و انجام عملگر پرتو، خروجی نهایی پرس و جوی به صورت زیر خواهد بود:

City
C2
C3

$$\sigma_{avg>16}(\Pi_{City}(STUD)) \quad (4)$$

گزینه چهارم نادرست است.

پرس و جوی زیر را در نظر بگیرید:

$$\sigma_{avg>16}(\Pi_{City}(STUD))$$

خروجی پرس و جوی پرائتز داخلی به صورت زیر است:

City
C1
C2
C3

که در نهایت پس از حرکت به سمت خارج، انجام عملگر انتخاب امکان پذیر نخواهد بود، زیرا ستون‌های عملگر σ یعنی avg زیر مجموعه ستون‌های عملگر Π یعنی City نیست، یعنی $avg \notin City$. در واقع ستون avg در پرائتز داخلی توسط عملگر Π انتخاب نشده است، بنابراین اجرای عملگر σ بر روی ستون avg امکان پذیر نیست.

۶۴- گزینه (۲) صحیح است.

مطابق فرض صورت سؤال، کار عملگر $\rho_{second}(Marks)$ این است که یک کپی از رابطه‌ی Marks با نام مستعار Second ایجاد کند.

سه جدول Marks, Students, و Courses با مقادیر زیر را در نظر بگیرید:

<u>sid</u>	sname	<u>sid</u>	<u>cid</u>	mark	<u>cid</u>	cname	dept
s1	sn1	s1	c1	12	c1	cn1	CS
s2	sn2	s2	c2	14	c2	cn2	CS
s3	sn3	s3	c3	16	c3	cn3	CE
s4	sn4				c4	cn4	IT
s5	sn5						

جدول *Sudents*جدول *Marks*جدول *Courses*

پرس‌وجوی مطرح شده در صورت سؤال به صورت زیر است:

$$\Pi_{\text{mark}}(\text{Marks}) - \Pi_{\text{Marks.mark}}(\text{Marks} \times \rho_{\text{second}}(\text{Marks}))$$

$$\text{Marks.mark} > \text{Second.mark}$$

توجه: یکی از عملگرهای اصلی جبر رابطه‌ای، عملگر تفاضل است. اگر R و S دو رابطه باشند، منظور از R-S مجموعه کلیه سطرهایی است که عضو R هستند، اما در S حضور ندارند.

توجه: در جبر رابطه‌ای تفاضل هر دو رابطه دلخواه امکان‌پذیر نیست. تفاضل دو رابطه در جبر رابطه‌ای زمانی امکان‌پذیر است که اگر و فقط اگر شروط سازگاری در مورد آن‌ها برقرار باشد. به طور کلی، اگر عملگر **تفاضل** در پرس و جو استفاده شود، این امر به معنی عبارت **نفی** در پرس و جو است. دقت کنید که در انجام عمل تفاضل برقراری شروط سازگاری الزامی است که در پرس و جو مطرح شده در صورت سوال این الزام برقرار است.

شروط سازگاری: در جبر رابطه‌ای دو شرط به عنوان شروط سازگاری مطرح است:

شرط اول: تعداد ستون‌های دو جدول یکسان باشد به عبارت دیگر دو جدول هم درجه باشند.

شرط دوم: نوع یا دامنه ستون‌های دو جدول یکسان باشد.

توجه: در پرس‌وجوی مطرح شده در هر دو طرف عملگر تفاضل، شروط سازگاری رعایت شده است.

براساس پرس‌وجوی مطرح شده در صورت سؤال داریم:

در ابتدا توسط عبارت $\Pi_{\text{mark}}(\text{Marks})$ در سمت چپ عملگر تفاضل نمره کلیه دانشجویان از جدول Marks به صورت زیر استخراج می‌گردد:

mark
12
14
16

همچنین براساس بخش سمت راست عملگر تفاضل داریم:

عبارت $(\text{Marks} \times \rho_{\text{second}}(\text{Marks}))$ ، ضرب دکارتی جدول Marks در خودش است. حاصل این ضرب $\text{Marks.mark} > \text{Second.mark}$

دکارتی، جدولی است که در آن، رکورد اطلاعاتی هر دانشجو از جدول Marks در کنار رکوردهای اطلاعاتی همه دانشجویان جدول second قرار می‌گیرد.

Marks.sid	Marks.mark	Second.sid	Second.mark
S ₁	12	S ₁	12
S ₁	12	S ₂	14
S ₁	12	S ₃	16
S ₂	14	S ₁	12
S ₂	14	S ₂	14
S ₂	14	S ₃	16
S ₃	16	S ₁	12
S ₃	16	S ₂	14
S ₃	16	S ₃	16

سپس شرط $\text{Marks.mark} > \text{Second.mark}$ ، سطرهایی که دانشجوی جدول Marks نمره‌اش حداقل از یک نفر دیگر بیشتر باشد را انتخاب می‌کند (یعنی دانشجویانی که نمره‌شان کمترین نمره نیست). بنابراین، پس از اعمال شرط داریم:

Marks.sid	Marks.mark	Second.sid	Second.mark
S ₂	14	S ₁	12
S ₃	16	S ₁	12
S ₃	16	S ₂	14

در ادامه، توسط عملگر Π Marks.mark، نمره این نوع از دانشجویان، به صورت زیر استخراج می‌گردد.

Marks.mark

14

16

توجه: در جبر رابطه‌ای، سطرهای تکراری در خروجی قرار نمی‌گیرند. در انتها، با انجام عملگر تفاضل، نمره دانشجویان فوق، از مجموعه نمره کلیه دانشجویان کنار گذاشته می‌شوند. آنچه باقی می‌ماند، نمره دانشجو یا دانشجویانی است که کمتر از همه نمره گرفته‌اند. به عبارت دیگر مینیمم نمره دانشجویان استخراج می‌گردد. بنابراین واضح است که پاسخ گزینه دوم است. یعنی «نمره مینیمم را از لیست نمرات استخراج می‌کند».

خروجی نهایی پرس‌وجوی مطرح شده به صورت زیر است:

$$\frac{\text{mark}}{12} - \frac{\text{Marks.mark}}{14} = \frac{\text{mark}}{12}$$

$$\frac{14}{14} - \frac{16}{14} = \frac{16}{14}$$

$$\frac{16}{14}$$

توجه: اگر شرط موجود در پرس‌وجوی مطرح شده عکس گردد، آنگاه گزینه سوم ایجاد می‌گردد.

۶۵- گزینه (۲) صحیح است.

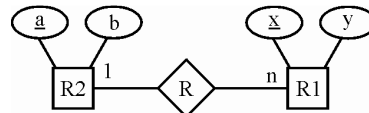
مطابق فرض سوال و ترتیب مطرح شده در آن، رابطه چند همان R1 و رابطه یک همان R2 است، همچنین pk(R) کلید اصلی رابطه R است.

نگاشت رابطه یک به چند بین دو موجودیت به مدل رابطه‌ای

مستقل از اختیاری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد. و کلید کاندید جدول یک در جدول چند به عنوان کلید خارجی تعریف می‌گردد.

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



مدل طراحی:

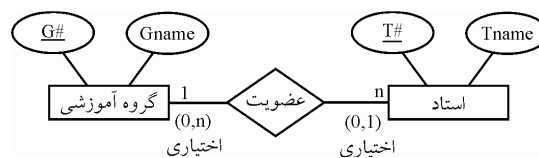
<u>a</u>	b	<u>x</u>	y	a
a1	b1	x1	y1	a1
a2	b2	x2	y2	a1
a3	b3	x3	y3	a2
		x4	y4	a2

جدول R2 جدول R1

توجه: در جدول R1 ستون x کلید اصلی $pk(R1) = x$ است، در نتیجه ستون x به عنوان کلید اصلی جدول R1 تمام ستون‌های جدول R1 را تولید می‌کند یعنی $pk(R1) \rightarrow R1$ یا $x \rightarrow R1$ یا $x \rightarrow (x, y, a)$ همچنین کلید کاندید جدول R2 در سمت یک یعنی ستون a به عنوان کلید خارجی در جدول R1 در سمت چند تعریف شده است. بنابراین همانطور گفتیم در جدول R1 داریم $x \rightarrow a$ یا $pk(R1) \rightarrow a$ یا $pk(R1) \rightarrow pk(R2)$ بنابراین پُر واضح است که گزینه دوم پاسخ سوال است.

مثال: حالت اختیاری

مدل تحلیل:



مدل طراحی:

<u>G#</u>	Tname	<u>T#</u>	Tname	G#
G1	Gn1	T1	Tn1	G1
G2	Gn2	T2	Tn2	G1
G3	Gn3	T3	Tn3	G2
		T4	Tn4	NULL

جدول گروه آموزشی جدول استاد

توجه: اگر استادی عضو گروه آموزشی نباشد، ستون G# برای آن مقدار NULL می‌گیرد، مانند T4 در جدول استاد.

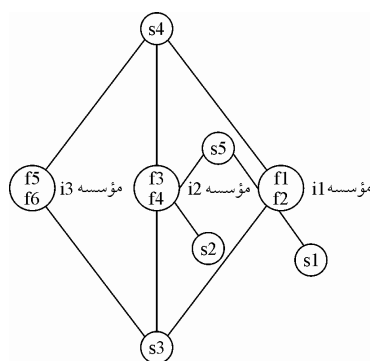
توجه: سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه دوم را بعنوان پاسخ اعلام کرده بود.

۶۶- گزینه (۴) صحیح است.

سه جدول Faviorate Field, Offered Field و Membership با مقادیر زیر را در نظر بگیرید:

SID	Field	SID	IID	IID	Field
s1	f1	s1	i1	i1	f1
s1	f2	s2	i2	i1	f2
s2	f3	s3	i3	i2	f3
s2	f4	s3	i1	i2	f4
s3	f5	s3	i2	i3	f5
s3	f6	s4	i3	i3	f6
s3	f1	s4	i1	Offered Field	
s3	f3	s4	i2		
s4	f5	s5	i1		
s4	f1	s5	i2		
s4	f3	Membership			
s5	f1				
s5	f2				
s5	f3				
s5	f4				
Faviorate Field					

شکل زیر روابط بین سه جدول فوق را نمایش می‌دهد:



مطابق پرس و جوی مطرح شده در گزینه چهارم، داریم:

$$\Pi_{SID}(\text{Membership}) - \Pi_{SID}(\Pi_{SID, Field}(\text{Membership} \bowtie \text{Offered Field}) - \text{Faviorate Field})$$

SID
s1
s2
s3
s4
s5

ابتدا توسط عبارت $\Pi_{SID}(\text{Membership})$ در سمت چپ عملگر تفاضل شماره کلیه هنرجویان از جدول Membership به صورت مقابل استخراج می‌گردد:

توجه: در جبر رابطه‌ای، سطرهای تکراری در خروجی قرار نمی‌گیرند.

همچنین براساس بخش سمت راست عملگر تفاضل داریم:

در عبارت $\Pi_{SID}(\Pi_{SID,Field}(\text{Membership} \bowtie \text{Offered Field}) - \text{Favirate Field})$ ، الحاق طبیعی پرائنتز داخلی یعنی $\Pi_{SID,Field}(\text{Membership} \bowtie \text{Offered Field})$ به صورت زیر انجام می‌شود:

SID	IID	Feild
s1	i1	f1
s1	i1	f2
s2	i2	f3
s2	i2	f4
s3	i3	f5
s3	i3	f6
s3	i1	f1
s3	i1	f2
s3	i2	f3
s3	i2	f4
s4	i3	f5
s4	i3	f6
s4	i1	f1
s4	i1	f2
s4	i2	f3
s4	i2	f4
s5	i1	f1
s5	i1	f2
s5	i2	f3
s5	i2	f4

سپس در ادامه، توسط عملگر $\Pi_{SID,Field}$ ، شماره هر هنرجو در کنار تمام فیلهای هنری هر موسسه‌ای که هنرجو در آن حضور دارد به صورت زیر استخراج می‌گردد:

SID	Feild
s1	f1
s1	f2
s2	f3
s2	f4
s3	f5
s3	f6
s3	f1
s3	f2
s3	f3
s3	f4
s4	f5
s4	f6
s4	f1
s4	f2
s4	f3
s4	f4
s5	f1
s5	f2
s5	f3
s5	f4

سپس در ادامه با انجام عملگر تفاضل، خروجی به صورت زیر استخراج می‌گردد:

SID	Feild	-	SID	Feild	=	SID	Feild
s1	f1		s1	f1		s3	f2
s1	f2		s1	f2		s3	f4
s2	f3		s2	f3		s4	f6
s2	f4		s2	f4		s4	f2
s3	f5		s3	f5		s4	f4
s3	f6		s3	f6			
s3	f1		s3	f1			
s3	f2		s3	f3			
s3	f3		s4	f5			
s3	f4		s4	f1			
s4	f5		s4	f3			
s4	f6		s5	f1			
s4	f1		s5	f2			
s4	f2		s5	f3			
s4	f3		s5	f4			
s4	f4						
s5	f1						
s5	f2						
s5	f3						
s5	f4						

Faviorate Field

سپس در ادامه، توسط عملگر Π SID، خروجی به صورت زیر استخراج می‌گردد:

SID
s3
s4

در انتها، با انجام عملگر تفاضل نهایی، شماره هنرجویان فوق، از مجموعه شماره کلیه هنرجویان کنار گذاشته می‌شود. آنچه باقی می‌ماند، شماره یا لیست تمام هنرجویانی است که فقط در موسسه‌هایی عضوند که هیچ رشته هنری خارج از علاقه‌مندی آنها را ارائه نمی‌دهد. دقت کنید که بر اساس صورت سوال، مهم نیست که یک هنرجو فقط عضو یک موسسه باشد یا نباشد، اما اگر عضو هر موسسه‌ای بود باید در تمام دوره‌های هنری موسسه شرکت کرده باشد. بنابراین واضح است که پاسخ گزینه چهارم است. خروجی نهایی پرس‌وجوی مطرح شده به صورت زیر است:

SID	-	SID	=	SID
s1		s3		s1
s2		s4		s2
s3				s5
s4				
s5				

توجه: در سایر گزینه‌ها اطلاعات نامرتبط و نامعتبر استخراج می‌گردد.

۶۷- گزینه (۱) صحیح است.

$$\sigma_{c1}(\sigma_{c2}(R)) \equiv \sigma_{c2}(\sigma_{c1}(R)) \quad (۱)$$

گزینه اول درست است، زیرا ترتیب اعمال شرط‌های c_1 و c_2 تأثیری در نتیجه نهایی ندارد و هر دو طرف عبارت هم‌ارزی معادل $\sigma_{c1} \& c_2(R)$ می‌باشد.

$$\Pi_{a1}(\Pi_{a2}(R)) \equiv \Pi_{a2}(\Pi_{a1}(R)) \quad (۲)$$

گزینه دوم نادرست است، زیرا عملگر پرتو (Π) همواره خاصیت جابه‌جایی ندارد. توجه کنید که عبارت $\Pi_{a1}(\Pi_{a2}(R))$ فقط در صورتی قابل انجام است که ستون‌های a_1 زیر مجموعه ستون‌های a_2 باشد یعنی $a_1 \subseteq a_2$ باشد که در آن صورت لزوماً $a_2 \subseteq a_1$ نیست (مگر در حالت استثنایی که $a_1 = a_2$ باشد)، بنابراین عبارت $\Pi_{a1}(\Pi_{a2}(R))$ در حالت کلی و در همه شرایط قابل انجام نیست و همچنین همواره معادل عبارت $\Pi_{a2}(\Pi_{a1}(R))$ هم نیست.

$$\Pi_{a1}(\sigma_{c1}(R)) \equiv \sigma_{c1}(\Pi_{a1}(R)) \quad (۳)$$

گزینه سوم نادرست است، زیرا دو عملگر σ و Π به صورت مشروط دارای خاصیت جابه‌جایی هستند. به طور کلی اگر R یک رابطه، L زیر مجموعه‌ای از ستون‌ها و θ مجموعه‌ای از شروط بر روی سطرها باشد، آنگاه تساوی زیر زمانی برقرار است که ستون‌های عملگر σ زیر مجموعه ستون‌های عملگر Π باشد. یعنی $\sigma \subseteq \Pi$

$$\Pi_{a1}(\Pi_{a2}(R)) \equiv \Pi_{a1}(R) \quad (۴)$$

گزینه چهارم نادرست است، زیرا $\Pi_{a1}(\Pi_{a2}(R)) \equiv \Pi_{a1}(R)$ فقط در شرایطی هم‌ارز است که ستون‌های a_1 زیر مجموعه ستون‌های a_2 باشد یعنی $a_1 \subseteq a_2$ باشد.

۶۸- گزینه (۳) صحیح است.

جدول $R(a,b,c)$ را با مقادیر زیر در نظر بگیرید:

a	b	c
1	2	2
3	4	4
5	6	7
5	7	6

جدول R

بخش اول از پرس و جوی Q_1 :

$$Q_1 : \rho(S(a, b_1, c_1), R)$$

در بخش اول پرس و جوی Q_1 ، جدول R به جدول S با ستون‌های a, b_1, c_1 نام‌گذاری مجدد می‌شود، به صورت زیر:

a	b_1	c_1
1	2	2
3	4	4
5	6	7
5	7	6

جدول S

در بخش دوم از پرس و جوی Q_1 :

$$\rho(T, S \bowtie R)$$

در بخش دوم پرس و جوی Q_1 ، جدول S و R الحاق طبیعی می شود و خروجی به جدول T با ستون های سابق نام گذاری مجدد می شود، به صورت زیر:

a	b ₁	c ₁	a	b	c	a	b ₁	c ₁	b	c
1	2	2	1	2	2	1	2	2	2	2
3	4	4	3	4	4	3	4	4	4	4
5	6	7	5	6	7	5	6	7	6	7
5	7	6	5	7	6	5	6	7	7	6
			5	7	6	5	7	6	6	7
			5	7	6	5	7	6	7	6

جدول S

جدول R

جدول T

توجه: ستون مشترک در الحاق طبیعی فوق فقط و فقط ستون a است. چون قبلا ستون b به b₁ و ستون c به c₁ نام گذاری مجدد شده است.

در بخش سوم از پرس و جوی Q_1 :

$$\Pi_{b_1, c}(\sigma_{b_1=c} T)$$

در بخش سوم پرس و جوی Q_1 ، ابتدا سطرهایی از جدول T بر اساس عملگر و شرط انتخاب سطر $\sigma_{b_1=c}$ انتخاب می شود، به صورت زیر:

a	b ₁	c ₁	b	c
1	2	2	2	2
3	4	4	4	4
5	6	7	7	6
5	7	6	6	7

سپس بر اساس عملگر و مدل انتخاب ستون $\Pi_{b_1, c}$ ، خروجی نهایی پرس و جوی Q_1 مشخص می شود، به صورت زیر:

b ₁	c
2	2
4	4
6	6
7	7

پرس و جوی Q_2 :

$$Q_2: \Pi_{b, c}(\sigma_{b=c} R)$$

در پرس و جوی Q_2 ، ابتدا سطرهایی از جدول R بر اساس عملگر و شرط انتخاب سطر $\sigma_{b=c}$ انتخاب می شود، به صورت زیر:

a	b	c
1	2	2
3	4	4

سپس بر اساس عملگر و مدل انتخاب ستون $\Pi_{b, c}$ ، خروجی نهایی پرس و جوی Q_2 مشخص می شود، به

صورت زیر:

$$\begin{array}{c|c} b & c \\ \hline 2 & 2 \\ 4 & 4 \end{array}$$

واضح است که مابین پرس و جوی Q_1 و Q_2 رابطه $Q_2 \subseteq Q_1$ برقرار است، به صورت زیر:

$$\begin{array}{c|c} b & c \\ \hline 2 & 2 \\ 4 & 4 \end{array} \subseteq \begin{array}{c|c} b_1 & c \\ \hline 2 & 2 \\ 4 & 4 \\ 6 & 6 \\ 7 & 7 \end{array}$$

Q_2 Q_1

بنابراین پُرواضح است که گزینه سوم پاسخ سوال است.

تست‌های فصل پنجم: حساب رابطه‌ای

۱- کدام گزینه پاسخ پرس‌وجوی «اسامی دانشکده‌هایی که همه دانشجویان آن معدل الف دارند» به روی DRC می‌باشد؟
(مهندسی IT - آزاد ۸۸)

$$\{ \langle c, \ln \mid \exists c, ci, pn (\langle c, \ln, ci, pn \rangle \in clg \wedge \forall s, sn, ci2, a, cl2 (\langle s, sn, ci2, a, cl2 \rangle \in stud \wedge a \geq 17)) \} \quad (۱)$$

$$\{ \langle c, \ln \mid \exists c, ci, pn (\langle c, \ln, ci, pn \rangle \in clg \wedge \exists s, sn, ci2, a, cl2 (\langle s, sn, ci2, a, cl2 \rangle \in stud \wedge a \geq 17)) \} \quad (۲)$$

$$\{ \langle c, \ln \mid \exists c, ci, pn (\langle c, \ln, ci, pn \rangle \in clg \wedge \exists s, sn, ci2, a (\langle s, sn, ci2, a, cl \rangle \in stud \wedge a \geq 17)) \} \quad (۳)$$

$$\{ \langle c, \ln \mid \exists c, ci, pn (\langle c, \ln, ci, pn \rangle \in clg \wedge \forall s, sn, ci2, a (\langle s, sn, ci2, a, cl \rangle \in stud \wedge a \geq 17)) \} \quad (۴)$$

۲- اگر جداول دانشجو و دانشکده و درس به صورت زیر باشند:

stud(s# , sname , city , avg , clg#)

crs(c# , cname , unit , clg#)

clg(clg# , clgname , city , pname)

در حساب رابطه‌ای دامنه‌ای، در کدام گزینه صورت پرس‌وجو و پاسخ با هم مطابقت ندارد؟

(مهندسی کامپیوتر - آزاد ۸۹ گروه الف)

(۱) شماره و نام دانشجویانی که معدل الف دارند یا متولد زاهدان هستند.

$$\{ \langle s, sn \rangle \mid \exists ci, a, cl (\langle s, sn, ci, a, cl \rangle \in stud \wedge (a \geq 17 \vee ci = "Zahedan")) \}$$

(۲) اسامی دانشکده‌هایی که همه دانشجویان آن معدل الف دارند.

$$\{ \langle c, \ln \rangle \mid \exists c, ci, pn (\langle c, \ln, ci, pn \rangle \in clg \wedge \forall s, sn, ci2, a (\langle s, sn, ci2, a, cl \rangle \in stud \wedge a \geq 17)) \}$$

(۳) شماره و نام دانشکده‌هایی که دروس چهار واحدی ارائه می‌کنند.

$$\{ \langle c, \ln \rangle \mid \exists ci, pn (\langle c, \ln, ci, pn \rangle \in clg \wedge \exists c, cn, u (\langle c, cn, u, cl \rangle \in crs \wedge u = 4)) \}$$

(۴) شماره و نام دانشجویانی که در دانشکده کامپیوتر تحصیل می‌کنند.

$$\{ \langle s, sn \rangle \mid \exists c, \ln, ci, pn (\langle c, \ln, ci, pn \rangle \in clg \wedge \exists a (\langle s, sn, ci, a, cl \rangle \in stud \wedge c, \ln = "computer")) \}$$

۳- کدام یک از گزاره‌های زیر درست است؟ (مهندسی IT - دولتی ۹۰)

(۱) قدرت بیان‌کنندگی (expressive power) حساب رابطه‌ای دامنه‌ای و حساب رابطه‌ای تاپلی با هم برابر است اگر عبارات در هر دو حساب به عبارات امن (safe expression) محدود گردند.

(۲) زبان QBE بر مبنای تئوری حساب رابطه‌ای تاپلی استوار شده است.

(۳) قدرت بیان‌کنندگی (expressive power) حساب رابطه‌ای تاپلی و جبر رابطه‌ای با هم برابر است.

(۴) SQL یک زبان رویه‌ای است.

۴- اگر جدول دانشجوی و دانشکده و درس به صورت زیر باشند:

stud(s# , sname , city , avg , clg#)

crs(c# , cname , unit , clg#)

clg(clg#clgname , city , pname)

در حساب رابطه‌ای دامنه‌ای، در کدام گزینه صورت پرس‌وجو و پاسخ با هم مطابقت دارد؟

(۱) شماره و نام دانشجویانی که معدل الف دارند یا متولد زاهدان هستند.

(۲) اسامی دانشکده‌هایی که همه دانشجویان آن معدل الف دارند.

$$\{ \langle cln \rangle \mid \exists cl, ci, pn (\langle cl, cln, ci, pn \rangle \in clg \wedge \forall s, sn, ci2, a (\langle s, sn, ci2, a, cl \rangle \in stud \wedge a \geq 17)) \}$$

(۳) شماره و نام دانشکده‌هایی که دروس چهار واحدی ارائه می‌کنند.

$$\{ \langle cl, cln \rangle \mid \exists ci, pn (\langle cl, cln, ci, pn \rangle \in clg \wedge \exists c, cn, u (\langle c, cn, u, cl \rangle \in crs \wedge u = 4)) \}$$

(۴) همه موارد

۵- براساس جداول S, P, J و SPJ به صورت زیر، در کدام یک از موارد زیر، نتیجه حاصل از اجرای عبارت

جبر رابطه‌ای با نتیجه حاصل از اجرای عبارت حساب رابطه‌ای تاپلی یکسان است؟ (مهندسی IT - دولتی ۹۴)

S(S#, SNAME, STATUS, CITY) جدول تولید کنندگان

P(P#, PNAME, WEIGHT, CITY) جدول قطعات

J(J#, JNAME, CITY) جدول پروژه‌ها

SPJ(S#, P#, J#, QTY) جدول تعداد قطعات تولید شده

توسط تولید کنندگان در پروژه‌ها

1) $\{ t \mid \exists spjx \in SPJ, \exists sx \in S (t[SNAME] = sx[SNAME] \wedge spjx[S\#] = sx[S\#] \wedge spjx[J\#] = 'j1') \}$

$\Pi_{S.SNAME} \sigma_{SPJ.J='j1'} (S \bowtie SPJ)$

2) $\{ t \mid \forall px \in P (\exists spjx \in SPJ, \exists sx \in S (spjx[S\#] = sx[S\#] \wedge spjx[P\#] = px[P\#] \wedge t[SNAME] = sx[SNAME])) \}$

$\Pi_{S.SNAME} (S \bowtie (\Pi_{S\#, P\#} SPJ \div \Pi_{P\#} P))$

(۱) فقط 1

(۲) فقط 2

(۳) 1 و 2

(۴) در هر دو مورد یکسان نیست.

۶- معادل حساب رابطه‌ای یا جبر رابطه‌ای پرس و جوی «نام دانشجویانی که حداقل یک درس عملی - منظور

درسی که COTYPE آن برابر 'p' باشد - در ترم 2 سال 94-95 انتخاب کرده باشند» کدام مورد زیر است؟

دانشجویان (STID, STNAME, STLEVEL, STMAJOR)

درس‌ها (COID, CONAME, COTYPE, CREDIT)

درس‌های انتخاب شده توسط دانشجویان (STID, COID, YR, TR, GRADE)

RANGVAR STX OVER ST

RANGVAR COX OVER CO

RANGVAR STCOX OVER STCO

A) STX.STNAME WHERE EXISTS STCOX (STCOX.YR='94-95' AND STCOX.TR='2' AND STCOX.STID=STX.STID AND EXIST COX (COX.COID=STCOX.COID AND COX.COTYPE='P '))

B) $\Pi(\text{STNAME}) \left(\text{ST} \bowtie (\sigma_{\text{YR}='94-95' \wedge \text{TR}='2'}(\text{STCO}) \bowtie \sigma_{\text{COTYPE}='P'}(\text{CO})) \right)$

C) STX.STNAME WHERE NOT EXISTS STCOX (STCOX.YR='94-95' AND STCOX.TR='2' AND STCOX.STID=STX.STID AND NOT EXIST COX (COX.COID=STCOX.COID AND COX.COTYPE='P '))

D) $\Pi(\text{STNAME}) \left(\sigma_{\text{YR}='94-95' \wedge \text{TR}='2' \wedge \text{COTYPE}='P'}(\text{ST} \bowtie (\text{STCO} \bowtie \text{CO})) \right)$

(مهندسی کامپیوتر-دولتی ۹۵)

(۱) فقط C (۲) فقط A و B (۳) فقط C و D (۴) A و B و D

۷- کدام مورد، عبارت حساب رابطه‌ای معادل «نام دانشجویانی که تمام دروس 4 واحدی را اخذ کرده‌اند.» است؟ (مهندسی کامپیوتر-دولتی ۹۶)

ST (STID, STName,...): دانشجویان

CO (COID, COName, Credit,...): دروس

STCO (STID, COID, Semester, Grade): دروس انتخاب‌شده توسط دانشجویان

Rangevar STX Over ST

Rangevar COX Over CO

Rangevar STCOX Over STCO

STX.STNAME WHERE FORALL COX (COX.CREDIT=4 AND EXISTS STCOX (STCOX.STID=STX.STID AND STCOX.COID=COX.COID))

STX.STNAME WHERE FORALL COX (COX.CREDIT !=4 OR NOT EXISTS STCOX(STCOX.STID=STX.STID AND STCOX.COID=COX.COID))

STX.STNAME WHERE NOT EXISTS COX(COX.CREDIT=4 AND NOT EXISTS STCOX(STCOX.STID=STX.STID AND STCOX.COID=COX.COID))

STX.STNAME WHERE EXISTS STCOX(STCOX.STID=STX.STID AND EXISTS COX(COX.COID=STCOX.COID AND COX.CREDIT=4))

پاسخ تست‌های فصل پنجم: حساب رابطه‌ای

۱- گزینه (۴) صحیح است.

DRC مخفف حساب رابطه‌ای دامنه‌ای (Domain Relation Calculus) است. از آن‌جا که صورت سؤال همه دانشجویان را مورد پرس و جو قرار داده است. پس باید از سور عمومی استفاده گردد. بنابراین گزینه‌های دوم و سوم را کنار می‌گذاریم. برای یافتن دانشجویان هم دانشکده باید دو رابطه دانشجویان و دانشکده براساس ستون مشترک cl به هم متصل شوند. بنابراین فقط باید تاپل‌هایی از دو رابطه بررسی شوند که مقدارشان در صفت مشترک یکسان است. اما در گزینه اول الزامی به یکسان بودن ستون مشترک cl نشده است. بر این اساس این گزینه نمی‌تواند پاسخ مورد نظر را تولید نماید. اما این نکته در گزینه چهارم رعایت شده است. اگر با دقت دو گزینه اول و چهارم بررسی شود مشخص خواهد شد که تنها تفاوت این دو گزینه در الزام به یکسان بودن cl در گزینه چهارم و عدم این الزام در گزینه اول است.

۲- گزینه (۴) صحیح است.

در عبارت حساب رابطه‌ای نوشته شده، در گزینه چهارم جدول clg با جدول stud روی دو فیلد #clg و city پیوند خورده است:

$$\{ \langle s, sn \rangle \mid \exists cl, cln, ci, pn (\langle cl, cln, ci, pn \rangle \in clg \wedge \exists a (\langle s, sn, ci, a, cl \rangle \in stud \wedge cln = \text{"computer"})) \}$$

پیوند روی #clg (به همراه شرط cln = "computer") باعث می‌شود فقط دانشجویان دانشکده‌ی کامپیوتر در خروجی ظاهر شوند و پیوند روی city باعث می‌شود دانشجویانی در خروجی ظاهر شوند که شهر محل زندگی‌شان با شهر محل تحصیل‌شان برابر باشد. بنابراین، خروجی این عبارت حساب رابطه‌ای عبارتست از: «شماره و نام دانشجویانی که در دانشکده‌ی کامپیوتر تحصیل می‌کنند و در شهر محل تحصیل خود زندگی می‌کنند.»

برای این‌که در خروجی، «شماره و نام دانشجویانی که در دانشکده‌ی کامپیوتر تحصیل می‌کنند» ظاهر شود باید دو جدول clg و stud فقط روی فیلد #clg پیوند زده شوند و پیوند روی فیلد city حذف شود. برای این منظور، کافی است در جدول stud برای فیلد city، نام دیگری به جز ci (مثلاً ci2) انتخاب کنیم و اسم این متغیر را جلوی نماد \exists مربوط به جدول stud ذکر کنیم:

$$\{ \langle s, sn \rangle \mid \exists cl, cln, ci, pn (\langle cl, cln, ci, pn \rangle \in clg \wedge \exists a, ci2 (\langle s, sn, ci2, a, cl \rangle \in stud \wedge cln = \text{"computer"})) \}$$

۳- گزینه (۱) صحیح است.

زبان QBE براساس حساب رابطه‌ای دامنه‌ای استوار است. قدرت بیان‌کنندگی حساب رابطه‌ای دامنه‌ای و حساب رابطه‌ای تاپلی با هم برابر است، اگر و تنها اگر در صورتی که عبارات به عبارات امن (مجموعه مقادیر از پیش تعریف شده) محدود شوند. قدرت بیان‌کنندگی حساب رابطه‌ای تاپلی از جبر رابطه‌ای بیش‌تر است. زیرا حساب رابطه‌ای تاپلی و دامنه‌ای مربوط به تحلیل پرس و جوها و مربوط به فاز تحلیل (ادراکی عام) و جبر رابطه‌ای مربوط به طراحی پرس و جوها و مربوط به فاز طراحی (ادراکی خاص) است، فاز تحلیل انتزاعی‌تر و به زبان محیط عملیاتی و زبان انسان نزدیک‌تر است و فاز طراحی به زبان

پیاده‌سازی در حال نزدیک شدن است.

SQL یک زبان بیانی (declarative) است نه یک زبان رویه‌ای یا دستوری، البته نسخه‌های رویه‌ای از SQL نیز توسعه یافته‌اند (مثل PL-SQL).

۴- گزینه (۴) صحیح است.

۵- گزینه (۱) صحیح است.

فعالیت‌های مربوط به تولید سیستم‌های نرم‌افزاری شامل مراحل تحلیل، طراحی و پیاده‌سازی می‌باشد. حساب رابطه‌ای (دامنه‌ای و تاپلی) برای تحلیل پرس‌وجو در فعالیت تحلیل مورد استفاده قرار می‌گیرد که خروجی این فعالیت به عنوان ورودی فعالیت طراحی برای طراحی پرس‌وجو توسط جبر رابطه‌ای مورد استفاده قرار می‌گیرد. حساب رابطه‌ای برای مدل‌سازی عالم خارج (زبان انسان) مورد استفاده قرار می‌گیرد. اما جبر رابطه‌ای برای مدل‌سازی عالم داخل (زبان ماشین) مورد استفاده قرار می‌گیرد.

توجه: متغیر سطری (تاپلی) $spjx$ و sx به عنوان دو متغیر به ترتیب بر روی دو جدول spj و s تعریف شده‌اند. متغیر سطری سطر به سطر بر روی جدولی که بر روی آن تعریف شده است حرکت می‌کند. به بیان دیگر دامنه حرکت یک متغیر سطری، جدولی است که بر روی آن تعریف شده است.

هر دو عبارت گزاره اول، نام تولیدکنندگان حاضر در پروژه iz را استخراج می‌کنند. در واقع نام تولیدکنندگانی را استخراج می‌کند که برای پروژه iz ، حداقل یک قطعه تولید کرده‌اند.

عبارت جبر رابطه‌ای داده شده در گزاره دوم، نام تولیدکنندگانی را استخراج می‌کند که همه قطعات را تولید کرده‌اند.

فرم صحیح عبارت حساب رابطه‌ای داده شده در گزاره دوم، برای استخراج نام تولیدکنندگانی که همه قطعات را تولید کرده‌اند، به صورت زیر است:

$$\{t \mid \exists sx \in s (t[sname] = sx[sname]) \text{ AND}$$

$$\forall px \in p (\exists spjx \in spj (spjx[s\#] = sx[s\#] \text{ AND } spjx[p\#] = px[p\#]))\}$$

بنابراین فقط در گزاره اول نتیجه حاصل از اجرای عبارت جبر رابطه‌ای با نتیجه حاصل از اجرای عبارت حساب رابطه‌ای تاپلی یکسان است.

۶- گزینه (۴) صحیح است.

سه جدول ST، CO و STCO با مقادیر زیر را در نظر بگیرید:

STID	STNAME	STLEVEL	STMAJOR	COID	CONAME	COTYPE	CREDIT
S1	Sn1			C1	Cn1	P	
S2	Sn2			C2	Cn2	P	
S3	Sn3			C3	Cn3	Q	
S4	Sn4			C4	Cn4	Q	
S5	Sn5						

جدول ST

جدول CO

STID	COID	YR	TR	GRADE
S1	C1	94-95	2	19
S1	C3	94-95	2	20
S2	C2	94-95	2	18
S3	C2	94-95	1	17
S4	C3	94-95	2	16

جدول STCO

مطابق پرس و جوی مطرح شده در عبارت B، داریم:

$$\Pi(\text{STNAME}) \left(\text{ST} \bowtie (\sigma_{\text{YR}='94-95' \wedge \text{TR}='2'}(\text{STCO})) \bowtie \sigma_{\text{COTYPE}='p'}(\text{CO}) \right)$$

ابتدا، در داخلی‌ترین پراتنز، بخش سمت چپ الحاق طبیعی، مشخصات درس‌های انتخاب شده توسط دانشجویان در سال 94-95 در ترم 2، از جدول STCO استخراج می‌گردد، به صورت زیر:

STID	COID	YR	TR	GRADE
S1	C1	94-95	2	19
S1	C3	94-95	2	20
S2	C2	94-95	2	18
S4	C3	94-95	2	16

سپس، خروجی فوق با سطرهایی از جدول CO که ستون COTYPE آن برابر مقدار P است، الحاق طبیعی می‌گردد.

سپس، در ادامه، در داخلی‌ترین پراتنز، بخش سمت راست الحاق طبیعی، مشخصات درس‌های عملی که ستون COTYPE آن برابر مقدار P است، از جدول CO استخراج می‌گردد، به صورت زیر:

COID	CONAME	COTYPE	CREDIT
C1	Cn1	P	
C2	Cn2	P	

در نهایت دو بخش سمت چپ و راست در داخلی‌ترین پراتنز، بر روی ستون مشترک COID، مطابق قوانین عملگر الحاق طبیعی، الحاق طبیعی می‌گردند.

بنابراین نتیجه داخلی‌ترین پراتنز، برابر STID یا شماره دانشجویانی خواهد بود که حداقل یک درس عملی (منظور درسی که COTYPE آن برابر 'p' باشد) در سال 94-95 در ترم 2 انتخاب کرده‌اند، به صورت زیر:

STID	COID	YR	TR	GRADE	CONAME	COTYPE	CREDIT
S1	C1	94-95	2	19	Cn1	P	
S2	C2	94-95	2	18	Cn2	P	

توجه: طبق تعریف عملگر الحاق طبیعی، در جبر رابطه‌ای ستون یا ستون‌های مشترک فقط یکبار در خروجی ظاهر می‌شود. در اینجا دو جدول STCO و CO دارای یک ستون مشترک یعنی COID هستند. پس خروجی الحاق طبیعی در داخلی‌ترین پراتنز دارای هشت ستون خواهد بود. در ادامه، نتیجه داخلی‌ترین پراتنز، با جدول ST (جدول دانشجویان) بر روی ستون مشترک STID، مطابق قوانین عملگر الحاق طبیعی، الحاق طبیعی می‌گردد، به صورت زیر:

STID	STNAME	COID	YR	TR	GRADE	CONAME	COTYPE	...
S1	Sn1			C1	94-95	2	19	Cn1	P	
S2	Sn2			C2	94-95	2	18	Cn2	P	

در نهایت، با اجرای عملگر Π ، نام دانشجویانی که حداقل یک درس عملی (منظور درسی که COTYPE آن برابر 'p' باشد) در ترم 2 سال 94-95 انتخاب کرده باشند، استخراج می‌گردد، به صورت زیر:

STNAME

Sn1

Sn2

که مطابق پرس و جوی مطرح شده در صورت سوال است. مطابق پرس و جوی مطرح شده در عبارت D، داریم:

$$\Pi(\text{STNAME}) \left(\sigma_{\text{YR}='94-95' \wedge \text{TR}='2' \wedge \text{COTYPE}='p'} (\text{ST} \bowtie (\text{STCO} \bowtie \text{CO})) \right)$$

ابتدا، جداول CO، STCO و ST با یکدیگر الحاق طبیعی می‌شوند، سپس عملگر انتخاب سطر (σ) اجرا می‌گردد. که خروجی نهایی آن به صورت زیر خواهد بود:

STID	STNAME	COID	YR	TR	GRADE	CONAME	COTYPE	...
S1	Sn1			C1	94-95	2	19	Cn1	P	
S2	Sn2			C2	94-95	2	18	Cn2	P	

در نهایت، با اجرای عملگر Π ، نام دانشجویانی که حداقل یک درس عملی (منظور درسی که COTYPE آن برابر 'p' باشد) در ترم 2 سال 94-95 انتخاب کرده باشند، استخراج می‌گردد، به صورت زیر:

STNAME

Sn1

Sn2

که مطابق پرس و جوی مطرح شده در صورت سوال است.

توجه: عبارت D معادل عبارت B است، با این تفاوت که در عبارت B ابتدا عملگر انتخاب روی جداول اجرا می‌شود و سپس نتایج با یکدیگر الحاق طبیعی می‌شوند، اما در عبارت D ابتدا جداول با یکدیگر الحاق طبیعی می‌شوند و سپس عملگر انتخاب اجرا می‌شود. مطابق پرس و جوی مطرح شده در عبارت A، داریم:

RANGVAR STX OVER ST

RANGVAR COX OVER CO

RANGVAR STCOX OVER STCO

STX.STNAME WHERE EXISTS STCOX (STCOX.YR='94-95' AND STCOX.TR='2' AND STCOX.STID=STX.STID AND EXIST COX (COX.COID=STCOX.COID AND COX.COTYPE='P'))

پرس و جوی مطرح شده در عبارت A به صورت زیر است:
 نام دانشجویانی در خروجی ظاهر گردد که به ازای آن وجود داشته باشد (EXISTS اول)، درس اخذ شده‌ای در ترم 2 سال 94-95 که عملی باشد (EXISTS دوم).
 به عبارت دیگر به ازای یک دانشجوی خاص، آیا وجود دارد درس اخذ شده‌ای در ترم 2، سال 94-95 که عملی باشد، اگر وجود داشت، نام آن دانشجو پاسخ مورد نظر پرس و جو خواهد بود. زیرا به ازای یک دانشجوی خاص وجود دارد حداقل درس اخذ شده‌ای در ترم 2، سال 94-95 که عملی باشد. مطابق پرس و جوی مطرح شده در عبارت A، ما به دنبال نام دانشجویانی هستیم که حداقل یکی از دروس اخذ شده آنها در ترم 2، سال 94-95 عملی باشد.
 خروجی پرس و جوی عبارت A به صورت زیر است:

STNAME

Sn1

Sn2

پرس و جو از جدول STX که همان جدول ST است، آغاز می‌گردد. سپس به ازای هر یک از سطرهاى جدول STX، کل شرایط مطرح شده، جلوی EXISTS اول و دوم به طور همزمان بررسی می‌گردد، در حال حاضر در سطر اول جدول STX قرار داریم، مطابق شرایط مطرح شده جلوی EXISTS اول که مطابق عبارت STX.STID=STCOX.STID می‌باشد. مقدار STX.STID در سطر اول جدول STX که برابر S1 می‌باشد را بردارید، حال بگردید داخل جدول STCOX و به دنبال STCOX.STID هایی باشید که برابر مقدار S1 باشد و هم در آن سطرها دو شرط 'STCOX.TR=2' AND 'STCOX.YR='94-95' برقرار باشد که دو سطر اول و دوم از جدول STCOX در این شرایط وجود دارد.

حال به ازای سطر اول و دوم از جدول STCOX، کل شرایط مطرح شده، جلوی EXISTS دوم بررسی می‌گردد. در حال حاضر در سطر اول جدول STCOX قرار داریم، مطابق شرایط مطرح شده جلوی EXISTS دوم که مطابق عبارت STCOX.COID=COX.COID می‌باشد، مقدار STCOX.COID در سطر اول جدول STCOX که برابر C1 می‌باشد را بردارید، حال بگردید داخل جدول COX و به دنبال COX.COID ای باشید که برابر مقدار C1 باشد و هم در آن سطر (ستون COID کلید اصلی است، بنابراین مقدار تکراری ندارد و دارای خاصیت یکتایی است) شرط 'COX.COTYPE='P' برقرار باشد که سطر اول از جدول COX در این شرایط قرار دارد.

حال در سطر دوم جدول STCOX قرار می‌گیریم، مطابق شرایط مطرح شده جلوی EXISTS دوم که مطابق عبارت STCOX.COID=COX.COID می‌باشد، مقدار STCOX.COID در سطر دوم جدول STCOX که برابر C3 می‌باشد را بردارید، حال بگردید داخل جدول COX و به دنبال COX.COID ای باشید که برابر مقدار C3 باشد و هم در آن سطر، شرط 'COX.COTYPE='P' برقرار باشد که همچنین سطر اول از جدول COX در این شرایط وجود ندارد.

حال اگر به ازای سطر اول از جدول STX با STID برابر مقدار S1 که پرس و جو را نیز با آن آغاز کردیم،

حداقل یکی از دو سطر اول و دوم از جدول STCOX که STID آن‌ها نیز برابر مقدار S1 است، شرط جلوی EXISTS دوم برقرار شود، آنگاه STNAME مربوط به سطر اول از جدول STX مطابق دستور STX.STNAME در خروجی قرار می‌گیرد. که مطابق آنچه پیش‌تر نیز گفتیم، شرایط مذکور برای سطر اول از جدول STCOX برقرار بود و برای سطر دوم از جدول STCOX برقرار نبود. از آنجاکه حداقل یک سطر در شرایط مذکور قرار دارد، پس خروجی پرس و جو به ازای حرکت در سطر اول از جدول STX به صورت زیر خواهد بود:

```
STNAME
-----
Sn1
```

این روند به ازای کلیه سطرها بعدی جدول STX ادامه دارد، که در صورت ادامه پرس و جو خروجی نهایی به صورت زیر خواهد بود:

```
STNAME
-----
Sn1
Sn2
```

به عبارت دیگر، نام دانشجویانی که حداقل یک درس عملی (منظور درسی که COTYPE آن برابر 'p' باشد) در ترم 2، سال 94-95 انتخاب کرده باشند، استخراج می‌گردد، به صورت زیر:

```
STNAME
-----
Sn1
Sn2
```

که مطابق پرس و جو مطرح شده در صورت سوال است. **توجه:** مطابق تعاریف حساب رابطه‌ای تاپلی یا سطری، STX متغیر سطری می‌باشد که بر اساس تعریف آن، سطر به سطر بر روی جدول ST حرکت می‌کند، برای سادگی در متن فوق، از STX به عنوان جدولی معادل ST یاد کردیم. مطابق پرس و جو مطرح شده در عبارت C، داریم:

```
RANGVAR STX OVER ST
RANGVAR COX OVER CO
RANGVAR STCOX OVER STCO
STX.STNAME WHERE NOT EXISTS STCOX (STCOX.YR='94-95' AND STCOX.TR='2'
AND STCOX.STID=STX.STID AND NOT EXIST COX (COX.COID=STCOX.COID AND
COX.COTYPE='P'))
```

پرس و جو مطرح شده در عبارت C، به صورت زیر است: نام دانشجویی در خروجی ظاهر گردد که به ازای آن وجود نداشته باشد (NOT EXISTS اول)، درس اخذ شده‌ای در ترم 2، سال 94-95 که عملی نباشد. (NOT EXISTS دوم) به عبارت دیگر نام دانشجویی در خروجی ظاهر گردد که همه دروس اخذ شده آن در ترم 2، سال 94-95 عملی باشد.

توجه: برای خوانایی بهتر، هرگاه دو not exist پشت سرهم در پرس و جوها دیدید، not exists اول را

exist بخوانید، خواندن را که تمام کردید، که ساده‌تر هم هست، مجدداً همان معنای not exist را در نظر بگیرید. برای مثال در اینجا، ببینید که به ازای یک دانشجوی خاص، آیا وجود دارد درس اخذ شده‌ای در ترم 2، سال 94-95 که عملی نباشد. اگر وجود داشت، نام آن دانشجو پاسخ مورد نظر پرس و جو نخواهد بود. زیرا به ازای یک دانشجوی خاص وجود دارد درس اخذ شده‌ای در ترم 2، سال 94-95 که عملی نیست. مطابق پرس و جو مطرح شده در عبارت C، ما به دنبال نام دانشجویانی هستیم که همه دروس اخذ شده آنها در ترم 2، سال 94-95، عملی باشد. خروجی پرس و جو عبارت C به صورت زیر است:

STNAME

Sn2

که مطابق پرس و جو مطرح شده در صورت سوال نیست. پس عبارات A، B و D مطابق پرس و جو مطرح شده در صورت سوال است، بنابراین گزینه چهارم پاسخ سوال است.

۷- گزینه (۳) صحیح است.

سه جدول ST، CO و STCO با مقادیر زیر را در نظر بگیرید:

<u>STID</u>	<u>STNAME</u>	<u>COID</u>	<u>CONAME</u>	<u>CREDIT</u>
S1	Sn1			C1	Cn1	4
S2	Sn2			C2	Cn2	3
S3	Sn3			C3	Cn3	4
S4	Sn4			C4	Cn4	3
S5	Sn5					

جدول ST

جدول CO

<u>STID</u>	<u>COID</u>	<u>SEMESTER</u>	<u>GRADE</u>
S1	C1		19
S1	C3		20
S2	C2		18
S3	C2		17
S4	C3		16

جدول STCO

مطابق پرس و جو مطرح شده در گزینه سوم، داریم:

```
STX.STNAME
WHERE
NOT EXISTS COX
(COX.CREDIT=4 AND
NOT EXISTS STCOX
(STCOX.STID=STX.STID AND STCOX.COID=COX.COID))
```

پرس و جو مطرح شده در گزینه سوم را همانند سورها به صورت زیر بخوانید:

«نام دانشجویانی که، وجود نداشته باشد (NOT EXISTS) درس 4 واحدی که، اخذ نشده باشد (NOT EXISTS). یعنی نام دانشجویانی که همه دروس 4 واحدی را اخذ کرده‌اند.»
خروجی پرس و جوی گزینه‌ی سوم به صورت زیر است:

STNAME

Sn1

که مطابق پرس و جوی مطرح شده در صورت سوال است.
مطابق پرس و جوی مطرح شده در گزینه‌ی چهارم، داریم:

```
STX.STNAME
WHERE
EXISTS STCOX
(STCOX.STID=STX.STID AND
EXISTS COX
(COX.COID=STCOX.COID AND COX.CREDIT=4))
```

پرس و جوی مطرح شده در گزینه‌ی چهارم را همانند سورها به صورت زیر بخوانید:
«نام دانشجویانی که، وجود داشته باشد (EXISTS) درس اخذ شده‌ای که، 4 واحدی باشد، (EXISTS).
یعنی نام دانشجویانی که حداقل دروس 4 واحدی را اخذ کرده‌اند.»
خروجی پرس و جوی گزینه‌ی چهارم به صورت زیر است:

STNAME

Sn1

Sn4

که مطابق پرس و جوی مطرح شده در صورت سوال نیست.

تست‌های فصل ششم: SQL دستورات DML

۱- کدام یک از عبارات زیر نادرست است؟ (مهندسی IT- دولتی ۸۵)

- (۱) استفاده از Correlated Query به جای Join موجب پایین آمدن کارایی (Performance) می‌شود.
 (۲) هر Query که قابل بیان به صورت یک Natural Join می‌باشد می‌تواند به صورت یک Subquery نیز بیان شود.
 (۳) هر Query که قابل بیان به صورت یک Subquery می‌باشد می‌تواند به صورت یک Equi Join نیز بیان شود.
 (۴) هر Query که قابل بیان به صورت یک Equi Join می‌باشد می‌تواند به صورت یک Natural Join نیز بیان شود.

۲- پرس و جوی ذیل در یک پایگاه داده مطرح است: (مهندسی IT-دولتی ۸۵)

«اسامی کارمندانی که مدیر پروژه نیز می‌باشند را لیست نمایید.»

کدام یک از عبارات ذیل برای فرموله کردن پرس و جوی فوق نادرست است؟

- | | |
|--|--|
| <pre>select ename from emps e where one projs p satisfies p.pmgr = e.empno</pre> <p style="text-align: right;">(۲)</p> | <pre>select ename from emps e where some projs p satisfies p.pmgr = e.empno</pre> <p style="text-align: right;">(۱)</p> |
| <pre>select ename from emps e where exists(select * from projs p weher p.pmgr = e.empno)</pre> <p style="text-align: right;">(۴)</p> | <pre>select ename from emps e where e.empno = any (select p.pmgr from projs p)</pre> <p style="text-align: right;">(۳)</p> |

۳- کدام یک از عبارات SQL زیر با بقیه عبارات معادل نمی‌باشد؟ (مهندسی کامپیوتر-دولتی ۸۵)

- (۱)

```
select P.*
from part P
where P.P# IN (select SP.P# from supply SP where SP.charge>40000)
```
- (۲)

```
select P.*
from part P
where P.P#= ALL (select S.P# from supply S where S.charge>40000)
```
- (۳)

```
Select P.*
from part P
where Exists (select * from supply SP where P.P#=SP.P# and SP.charge>40000)
```
- (۴)

```
select P.*
from part P
where P.P#=(select SP.P# from supply SP where P.P#=SP.P# and SP. charge> 40000)
```

۴- رابطه **Person (no, name, father_no)** شامل شماره شناسایی، نام افراد و شماره شناسایی پدر آنهاست. کدام گزینه نام همه پدرهای موجود در رابطه را می‌دهد؟ (مهندسی کامپیوتر-دولتی ۷۵)

<pre>select P1.name from person P1 , Person P2 where P1.father_no = P2.father_no</pre>	(۲)	<pre>select name from person where no = father_no</pre>	(۱)
<pre>select P1.name from Person P1, Person P2 where P1.no = P2.father_no</pre>	(۴)	<pre>select name from person where name IN (select name from person where no = father_no)</pre>	(۳)

۵- جدول A دارای ستون **Price** و **Pcode** است. کدام گزینه در مورد عبارت زیر صحیح است؟ (مهندسی کامپیوتر-دولتی ۷۵)

```
select Pcode, Price
from A
where Price = Max (Price)
```

(۱) معادل عبارت مقابل است:

```
select Pcode, Price
from A
where Price = (select Max (Price) from A)
```

(۲) اشکال منطقی دارد زیرا محاسبه ماکزیمم در مرحله جداگانه‌ای انجام نمی‌شود.
 (۳) همواره یک سطر چاپ می‌کند که غلط است.
 (۴) معادل عبارت مقابل است:

```
select Pcode, Max (Price)
from A
where Max (Price) = select Price from A)
```

۶- جداول زیر برای یک بانک مفروض است: (کارشناسی ارشد- دولتی ۸۲)

مشتری **CUSTOMER (CUSTNAME , ADDRESS)**
 حساب **ACCOUNT (ACCNO , CUSTNAME)**
 وام **LOAN (LOANNO , CUSTNAME)**

دستور SQL زیر برای تعیین مشتریانی که وام نگرفته‌اند کافی است.

(۱)

```
SELECT *FROM CUSTOMER , LOAN
WHERE CUSTOMER.CUSTNAME <> LOAN.CUSTNAME
```

(۲)

```
SELECT *FROM CUSTOMER, LOAN
WHERE CUSTOMER.CUSTNAME = LOAN.CUSTNAME
```

(۳)

```
SELECT * FROM CUSTOMER
WHERE CUSTNAME NOT IN
(SELECT CUSTNAME FROM LOAN)
```

(۴) هیچ کدام

۷- ضد نیم اتصال دو رابطه R و S بنا بر تعریف شامل تاپل‌های آویزان R می‌شود. به این معنا که تاپل‌هایی که متناظری در S و هم‌چنین در هیچ رابطه دیگری ندارند. روابط R(A,B,C) و S(A,B,D) را در نظر بگیرید که براساس صفات مشترک A و B متصل می‌شوند و فرض کنید که <A, B> کلیدی برای هر دو رابطه می‌باشد. کدام گزینه یک ضد نیم اتصال را به طور صحیح پیاده‌سازی می‌کند؟ (مهندسی IT-دولتی ۸۴)

Select R.*from R , S where R.A <> S.A AND R.B <>S.B (۱)

Select*from R where not exists (Select*from S where S.A= R.A OR S.B=R.B) (۲)

Select*from R where not exists (Select*from S where S.A=R.A AND S.B=R.B) (۳)

Select*from R where A<>any (Select A from S) AND B<>any (Select B from S) (۴)

۸- رابطه تولیدکننده S(S#,City,Status) مفروض است. برای بازایی Status و S# مربوط به تولیدکنندگان شهر کرمان براساس ترتیب نزولی Status کدام یک از دستورات SQL زیر صحیح می‌باشد؟

(مهندسی کامپیوتر-دولتی ۷۴)

Select S# , Status	(۲)	Select S# , Status	(۱)
Where City = 'Kerman'		From S	
From S		Where City = 'Kerman'	
Order By Status		Order By Status Desc	
Order By Status		Select Status, S#	
From S		Where City = 'Kerman'	(۳)
Select Status, S#	(۴)	Order By Status Desc	
Where City = 'Kerman'		From S	

۹- در صورتی که مفهوم رابطه <SPJ<S#,P#,J#,QTY> به شکل ذیل بیان گردد:
«تهیه‌کننده S#، قطعه P# را برای پروژه J# به تعداد QTY تهیه نموده است» و کلید اصلی رابطه S#,P#,J# باشد: برای فرموله کردن پرس‌وجوی زیر به زبان SQL کدام قسمت تکمیلی باید به دستور SQL زیر ملحق شود؟ (مهندسی IT-دولتی ۸۳)

پرس‌وجو:	دستور SQL:
شماره پروژه‌هایی را معین کنید که تمام قطعات آن‌ها را یک تهیه‌کننده، تهیه کرده باشد.	Select J # From SPJ

Group by P#	(۲)	Group by S#	(۱)
Having count(*) = 1		Having Cont(*) = 1	
Group by J#	(۴)	Group by J# , P#	(۳)
Having Count (Distinct S#) = 1		Having Count (S#) = 1	

(مهندسی IT - دولتی ۸۳)

۱۰- جدول محصول به صورت زیر مفروض است:

goods(P#,Pname,Price)

خروجی دستور زیر کدام گزینه است؟

Select Price form goods

Where Price >= Max (Price)

(۲) کل جدول

(۱) NULL

(۴) ماکزیمم قیمت در جدول

(۳) بستگی به داده‌ها دارد.

(مهندسی IT - دولتی ۸۳)

۱۱- کدام یک از گزینه‌ها نادرست است؟

- (۱) قانون اول و دوم جامعیت داده‌ای به وسیله تعریف دامنه (domain) برای ویژگی‌ها قابل انجام است.
 (۲) برای اعمال قانون اول جامعیت داده‌ای در SQL کافی است در تعریف ویژگی که به عنوان کلید اصلی است not NULL و unique قید شود.
 (۳) برای اعمال قانون اول جامعیت داده‌ای در SQL می‌بایست در ایجاد جدول کلید اصلی به وسیله primary key تعریف شود.
 (۴) برای اعمال قانون سوم جامعیت داده‌ای در SQL می‌بایست در ایجاد جدول کلید خارجی به وسیله foreign key تعریف شود.

۱۲- اگر employee (empno, empname, dept, salary)

شماره کارمند: empno ، نام کارمند: empname ، واحد: dept و حقوق: salary باشد، آنگاه جواب پرسش زیر کدام یک از گزینه‌ها می‌باشد؟
 پرسش: مشخصات کارکنانی که حقوقشان از متوسط حقوق کلیه کارکنان سازمان بیش تر است را لیست نمایید.

(۱)

```
select *
from employee
where salary > avg (salary)
```

(۲)

```
select *
from employee
having salary > avg (salary)
```

(۳)

```
select *
from employee
where salary > (select avg (salary) from employee)
```

(۴)

```
select *
from employee
group by dept having salary > avg (salary)
```

۱۳- دستور SQL زیر در بانک اطلاعاتی تولیدکنندگان و قطعات چه کار می کند؟

(مهندسی IT-دولتی ۸۶ و آزاد ۸۸)

```
UPDATE P
SET COLOR='Orange'
WHERE COLOR = 'Red'
```

- (۱) تمامی قطعات قرمز رنگ را به رنگ نارنجی در می آورد.
- (۲) تمامی قطعات نارنجی رنگ را به رنگ قرمز در می آورد.
- (۳) اولین قطعه قرمز رنگ را به رنگ نارنجی در می آورد.
- (۴) آخرین قطعه قرمز رنگ را به رنگ نارنجی در می آورد.

۱۴- اگر رابطه های $S(S\#,SNAME,STATUS,CITY)$ و $SP(S\#,P\#,QTY)$ مفروض باشد، معادل پرسش زیر کدام گزینه است؟

(مهندسی IT-دولتی ۸۶)

```
SELECT SNAME, CITY
FROM S
WHERE S# IN (SELECT S#
              FROM SP
              WHERE P#='P1')
```

(۱)

```
SELECT SNAME, CITY
FROM S
WHERE S# = ANY (SELECT S# FROM SP WHERE P# = 'P1')
```

(۲)

```
SELECT SNAME, CITY
FROM S
WHERE EXISTS (SELECT * FROM SP WHERE SP.P#='P1'
              AND SP.S#=S.S#)
```

(۳)

```
SELECT SNAME, CITY
FROM S,SP
WHERE S.S#=SP.S# AND SP.P# = 'P1'
```

(۴) هر سه معادل می باشند.

۱۵- دو جدول x و y مفروض هستند. در صورت اجرای دستور SQL زیر کادرینالیتی جدول حاصل کدام است؟

(مهندسی IT- آزاد ۸۵)

	x		y		
	F	B	A	B	C
Select *	f1	b1	a1	b1	MK
From y Right outer join x	f2	b2	a2	b4	MTK
On y.B=x.B			a1	b2	LN
Where y.c Like 'M%'			a2	b1	RP

3 (۴)

4 (۳)

1 (۲)

2 (۱)

پایگاه داده ای دارای سه جدول S، L و G به صورت زیر است:

$S(S\#,Sname),L(L\#,Lname),G(S\#,L\#,grade)$

که در آن جدول S یا جدول دانشجو با خصیصه‌های شماره دانشجویی و نام دانشجو، جدول L یا جدول درس با خصیصه‌های شماره درس و نام درس و جدول G یا جدول نمره با خصیصه‌های شماره دانشجویی، شماره درس و نمره درس است، با توجه به این پایگاه داده به دو سؤال بعدی پاسخ دهید.

(مهندسی IT-آزاد ۸۶)

۱۶- خروجی دستور SQL زیر کدام است؟

```
SELECT DISTINCT Sname
FROM S
WHERE NOT EXISTS (SELECT S# FROM G WHERE L# = 'L1')
```

- ۱) نام تمام دانشجویانی را مشخص می‌کند که از درس L₁ نمره دارند.
- ۲) نام بعضی از دانشجویانی را مشخص می‌کند که از درس L₁ نمره ندارند.
- ۳) نام تمام دانشجویانی را مشخص می‌کند که از درس L₁ نمره ندارند.
- ۴) نام تمام دانشجویانی را مشخص می‌کند که از درس L₁ حداکثر یک نمره دارند.

(مهندسی IT-آزاد ۸۶)

۱۷- خروجی عبارت SQL زیر کدام است؟

```
SELECT DISTINCT S#
FROM G
WHERE grade = (SELECT MAX (grade) FROM G)
```

- ۱) شماره تمام دانشجویانی را مشخص می‌کند که بیش‌ترین نمره را اخذ نموده‌اند.
- ۲) شماره بعضی از دانشجویانی را مشخص می‌کند که بیش‌ترین نمره را اخذ نموده‌اند.
- ۳) شماره دانشجویانی را مشخص می‌کند که نمره بیشتری اخذ نموده است.
- ۴) این دستور مشکل گرامری دارد.

(مهندسی IT-دولتی ۸۵)

۱۸- در عبارت

```
SELECT State, COUNT (State)
FORM Customer_T
GROUP BY State
HAVING COUNT (State) >1
```

- ۱) می‌توانیم از یک Subquery استفاده کنیم.
- ۲) می‌توانیم از WHERE به جای HAVING استفاده کنیم.
- ۳) می‌توانیم شرط $COUNT (State) < 10$ را با شرط فعلی ترکیب کنیم.
- ۴) می‌توانیم از $COUNT (*)$ به جای $COUNT (State)$ استفاده کنیم.

۱۹- با اجرای دستور SQL زیر روی بانک اطلاعاتی تولید کنندگان قطعات:

```
INSERT INTO S (S # , SNAME, CITY)
VALUES ('S10' , 'Smith', 'New York')
```

(مهندسی کامپیوتر-دولتی ۸۷)

کدام یک از گزینه‌های زیر صحیح است؟

- ۱) دستور با خطای اجرا مواجه می‌شود.
- ۲) دستور با خطای ترجمه مواجه می‌شود.
- ۳) اعتبار (STATUS) تولید کننده Smith مقدار NULL خواهد شد.
- ۴) اعتبار (STATUS) تولید کننده Smith مقدار قبلی خود را حفظ می‌کند.

۲۰- نتیجه دستور SQL زیر چیست؟

(مهندسی کامپیوتر-دولتی ۸۷)

```
SELECT P.*
FROM S, P, SP
WHERE S. CITY= 'LONDON'
AND S.S# = SP.S#
AND P.P# = SP.P#
```

(۱) دستور دارای خطای نحوی است.

(۲) کلیه مشخصات قطعاتی که توسط تولیدکنندگان لندن تولید می‌شود.

(۳) کلیه مشخصات قطعاتی که حداقل توسط یکی از تولیدکنندگان لندن تولید می‌شود.

(۴) کلیه مشخصات قطعاتی که دقیقاً توسط یکی از تولیدکنندگان لندن تولید می‌شود.

۲۱- دو جدول زیر را در نظر بگیرید:

(مهندسی IT-دولتی ۸۷)

loan			borrower	
loan-number	branch-name	amount	customer-name	loan-number
L-170	Downtown	3000	Jones	L-170
L-230	Redwood	4000	Smith	L-230
L-260	Perryridge	1700	Hayes	L-155

حاصل اجرای کدام یک از عبارات‌ها به شکل زیر است؟

loan-number	branch-name	amount	customer-name
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	NULL
L-155	NULL	NULL	Hayes

(۱)

```
select *
from loan natural left outer join borrower
```

(۲)

```
select *
from loan full outer join borrower using (loan-number)
```

(۳)

```
select *
from loan natural inner join borrower
```

(۴)

```
select *
from loan right outer join borrower
on (loan. loan-number = borrower. Loan- number)
```

(مهندسی کامپیوتر-آزاد ۸۷)

۲۲- کدام عبارت SQL معادل عبارت جبر رابطه‌ای زیر است؟

```
 $\Pi_{sname}[(stud \times \sigma_{clg\ name='computer'}(clg))]
stud.clg\# = clg.clg\#$ 
```

(۱)

```
select sname
from stud, clg
where clgname = 'computer'
```

(۲)

```
select sname
from stud
where clg#in (select clg#
              from clg
              where clgname = 'computer')
```

(۳)

```
select sname
from stud
where clgname = 'computer'
AND clg# = (select clg#
            from clg)
```

(۴)

```
select sname
from stud
group by clg#
having clg#contains (select clg#
                    from clg
                    where clgname = 'computer')
```

(مهندسی کامپیوتر-آزاد ۸۷)

۲۳- کدام گزینه معادل سایر گزینه‌ها نیست؟

(۱)

```
select ename
from employee
where not exists (select depnum
                 from dep
                 where depname = 'sale' and dep.depnum = employee.depnum)
```

(۲)

```
select ename
from employee
where depnum in (select depnum
                from dep
                where depname='sale')
```

(۳)

```
select ename
from employee
where depnum= any (select depnum
                  from dep
                  where depname='sale')
```

(۴)

```
select ename
from employee, dep
where employee. depnum = dep.depnum and depname = 'sale'
```

(مهندسی IT-آزاد ۸۷)

۲۴- جدول ثبت‌نام به صورت زیر تعریف شده است:

(نمره score، نیمسال term، شماره دانشجویی studnum، شماره گروه درسی enroll (secnum))

که در ابتدای ترم، ثبت نام همه دانشجویان در آن وارد می شود. فرض کنید استادی در انتهای ترم نمرات دانشجویان را در یک گروه درسی (مثلاً 1577) اعلام کرده است. برای وارد کردن نمرات در این جدول از کدام دستور باید استفاده نمود؟

Insert (۱) Update (۲) Delete (۳) Select (۴)

۲۵- فرض کنید یک بانک اطلاعاتی شامل سه جدول به صورت زیر باشد:

T1 (A, B, C, FK) T2 (D, E, W, FK) T3 (FK, X, Y, Z)

که FK در جدول T3 کلید اصلی و در جدول T1 و T2 کلید خارجی است. T1 و T2 به جز FK کلید خارجی دیگری ندارند. فرض کنید به هنگام تعریف جداول، برای FK در T1 عبارت `on update cascade` ذکر شده باشد، اما برای FK در T2 ذکر نشده باشد، اگر دستور زیر را اجرا کنیم چه اتفاقی خواهد افتاد؟ (مهندسی IT-آزاد ۸۷)

```
update T3
set FK=15
where FK=10
```

- (۱) مقدار FK فقط در جدول T3 به روز می شود.
 (۲) مقدار FK در جداول T1 و T3 به روز می شود.
 (۳) مقدار FK در هیچ یک از جداول به روز نمی شود.
 (۴) مقدار FK در هر سه جدول به روز می شود.

۲۶- با توجه به جداول داده شده، کدام عبارت SQL معادل عبارت جبر رابطه ای زیر است؟

(مهندسی IT-آزاد ۸۷)

prof (pnum, pname, esp, depnum) dep (depnum, depname)

$\Pi_{pname}(\sigma_{depname='computer'}(prof \bowtie dep))$

(۱)

```
select pname
from prof
where depnum in (select depnum
                  from dep
                  where depname= 'computer')
```

(۲)

```
select pname
from prof, dep
where depname= 'computer'
```

(۳)

```
select pname
from prof
where prof.depnum=dep.depnum and depname in (select depname
                                              from dep
                                              where depname= 'computer')
```

(۴)

```
select pname
from prof
group by depnum
having depnum= (select depnum
                from dep
                where depname= 'computer')
```

۲۷- در SQL شرط EXISTS R چه موقع صحیح (True) می‌شود؟ (مهندسی کامپیوتر- دولتی ۸۷)

- (۱) اگر R یک دید (view) باشد.
 (۲) اگر R یک رابطه پایه (base relation) باشد.
 (۳) اگر R حداقل یک سطر داشته باشد.
 (۴) اگر R حداقل یک ستون داشته باشد.

۲۸- اگر عنوان رابطه $R = (A, B, C, D)$ و $S = (C, D)$ و رابطه‌های $r(R)$ و $s(S)$ را داشته باشیم. عمل تقسیم $r \div s$ توسط کدام مورد محقق می‌گردد؟ (مهندسی کامپیوتر-دولتی ۸۸)

(select A, B from r)	
except	with rl (A, B) as
(with rl(A, B) as	select A, B
select A, B	from r
from r	select A, B
with r2 (A, B, C, D) as (۲)	from rl (۱)
select *	where exists (select C, D
from rl, s	from s
select A, B	where (A, B, C, D) in r)
from (r2 except r))	
(select A, B from r)	with rl (A, B) as
except	select A, B
(with rl(A, B) as	from r
select A, B	select A, B
from r	from rl (۳)
with r2(A, B, C, D) as (۴)	where not exists (select C, D
select *	form s
froml rl, s	where (A, B, C, D) in r)
select A, B	except s)
from (r intersect r2))	

۲۹- کدام گزینه پاسخ پرس و جوی «ترم‌هایی که دانشکده کامپیوتر درس ارائه کرده است» می‌باشد؟

(مهندسی کامپیوتر-آزاد ۸۸)

select term	select distinct term
from sec, clg, crs	from sec, clg
wehre clg.clg# = crs.clg# (۲)	where clg.pname = sec.pname (۱)
and crs.c# = sec.c#	and clgname = "کامپیوتر"
group by term	
select distinct term	select term
from sec, clg, crs	from clg, crs
where clg.clg# = crs.clg# (۴)	where clg.clg# = crs.clg# (۳)
and crs.c# = sec.c#	having clgname = "کامپیوتر"
and clgname = "کامپیوتر"	

۳۰- خروجی دستور زیر کدام است؟

(مهندسی کامپیوتر-آزاد ۸۸)

```
select sname
from stud
where avg > AVG (avg)
```

(۱) دانشجویانی که معدل آن‌ها از نیمی از معدل‌ها بیش تر است.

(۲) این دستور غلط است.

(۳) دانشجویانی که معدل آن‌ها از میانگین همه معدل‌ها بیش تر است.

(۴) دانشجویانی که معدل ترم جاری آن‌ها از میانگین همه معدل‌ها بیش تر است.

۳۱- خروجی پرس‌وجوی زیر چیست؟

(مهندسی IT-آزاد ۸۸)

```
select distinct essn
from work-on
where pno in (select pno
              from work-on
              where name = 'Parvizi')
```

(۱) اسامی کارفرمایان پروژه‌هایی که پرویزی روی آن‌ها کار می‌کند.

(۲) اسامی کارفرمایان کارمندانی که روی یکی از پروژه‌هایی کار می‌کنند که پرویزی روی آن‌ها کار می‌کند.

(۳) اسامی کارفرمایان کارمندانی که دقیقاً روی همان پروژه‌هایی کار می‌کنند که پرویزی روی آن‌ها کار می‌کند.

(۴) اسامی کارفرمایان کارمندانی که روی تمام پروژه‌هایی که پرویزی روی آن‌ها کار می‌کند، کار می‌کند.

۳۲- کدام گزینه پاسخ پرس‌وجوی «مشخصات دانشجویانی که معدل‌شان از متوسط معدل دانشجویان

(مهندسی IT-آزاد ۸۸)

دانشکده‌شان بیش تر است» می‌باشد؟

(۱)

```
select *
from stud
having average > AVG (dept.average)
```

(۲)

```
select *
from stud
group by dept
having average > AVG (average)
```

(۳)

```
select *
from stud S1
where average > (select AVG (average)
                from stud S2
                where S1.dept=S2.dept)
```

(۴)

```
select *
from stud
where average > (select AVG (average)
                from stud)
```

۳۳- در بانک زیر کدام گزینه پاسخ پرس و جوی «نام ملوان‌هایی که همه قایق‌ها را رزرو کرده‌اند» می‌باشد؟

(مهندس کامپیوتر-دولتی ۸۴)

Sailor (sid, sname, rating, age) ملوان

Boats (bid, bname, color) قایق

Reserves (sid, bid, day) رزرو

select S.sname from sailor S where not Exists ((select B.bid from Boats B) Except

(select R.bid from Reserves R (۱)

where S.sid=R.sid))

select S.name from Sailor S where not Exists (select B.bid from Boats B where

not Exists (select R.bid from Reserves R where (۲

R.bid=B.bid and R.sid=S.sid))

(۳ موارد ۱ و ۲

(۴ هیچ کدام

۳۴- در مورد عبارت SQL زیر کدام یک از گزینه‌ها صحیح است؟ (مهندس کامپیوتر-دولتی ۸۵)

select avg (balance) from account where branch-name='melli' (الف)

select branch-name, avg (balance) from account group by (ب)

branch-name having avg (balance)> 1200

select branch-name from Branch where assets > Any (select assets from (ج)

Branch where branch- city='ISFAHAN')

(۱ فقط عبارت الف صحیح است. (۲ فقط عبارت الف و ب صحیح است.

(۳ هر سه عبارت الف، ب و ج صحیح است. (۴ لااقل یکی از عبارات الف، ب و ج غلط است.

۳۵- دستور SQL زیر در بانک اطلاعاتی تولیدکنندگان و قطعات چه کار می‌کند؟ (مهندس کامپیوتر-دولتی ۸۶)

select SP.p# , sum (SP.QTY) As TOTQTY

from SP

group by SP.p#

(۱ شماره هر قطعه و تعداد کل قطعات تولید شده را می‌دهد.

(۲ شماره هر قطعه و تعداد کل آن قطعه تولید شده را می‌دهد.

(۳ شماره هر قطعه و وزن کل قطعات تولید شده را می‌دهد.

(۴ شماره هر قطعه و وزن کل قطعات تولید شده توسط تولیدکننده آن قطعه را می‌دهد.

۳۶- دستور SQL زیر در بانک اطلاعاتی تولیدکنندگان و قطعات چه بر می‌گرداند؟ (مهندس کامپیوتر-دولتی ۸۶)

select S.name from S where S.s# in (select SP.s# from SP where SP.p# = 'p2')

(۱ اسامی تولیدکنندگانی که قطعه شماره 'p2' را تولید می‌کنند. (احتمالاً با تکرار)

(۲ اسامی تولیدکنندگانی که قطعه شماره 'p2' را تولید می‌کنند (بدون تکرار).

(۳ اسامی تولیدکنندگانی که فقط قطعه 'p2' را تولید می‌کنند.

(۴ اسامی تولیدکنندگانی که علاوه بر قطعه شماره 'p2' قطعه دیگری نیز تولید می‌کنند.

۳۷- پایگاه داده‌ای با دستورات زیر ایجاد شده است:

```
create table Books (isbn chr (20) primary key,
                  bname char (50),
                  type char (9) check (type in ('Techincal', 'fiction', 'self-help'))
                  );
```

```
create table Authors (ssn char (9),
                    isbn char (20),
                    primary key (ssn, isbn));
```

کدام یک از گزینه‌های زیر برای پیاده‌سازی پرس‌وجوی «شماره ssn را برای نویسندگانی که هیچ کتابی از نوع fiction نوشته‌اند بیابید» با استفاده از SQL نادرست است؟ (مهندسی کامپیوتر-دولتی ۸۴)

(۱)

```
select ssn from Authors
Except (select ssn from Authors A, Books B
       where A.isbn=B.isbn and B.type='fiction')
```

(۲)

```
select ssn from Authors, Books
where Authors.isbn=Books. isbn and type <> 'fiction'
```

(۳)

```
select ssn from Authors A
where Not Exists (select * from Books B
                 where A.isbn=B.isbn and B.type='fiction')
```

(۴)

```
select ssn from Authors where ssn Not in
(select ssn from Authors A, Books B
 where A.isbn= B.isbn and B.type='fiction')
```

۳۸- پایگاه داده‌ای با جداول زیر را در نظر بگیرید. حاصل کدام یک از پرس‌وجوهای زیر با بقیه متفاوت است؟ (مهندسی IT-دولتی ۸۷)

```
account (acc#, bname , balance)      depositor (cname , acc#)
```

```
select cname
from depositor as T, account
where T.acc# = account.acc#and
bname = 'perryridge'
group by cname
having count (T.acc#) <= 1
```

(۲)

```
select cname
from depositor as T, account
where T.acc# = account.acc#and
bname = 'perryridge'
group by cname
having count (*) <= 1
```

(۱)

```
select T.cname
from depositor as T where unique
(select R.cname
from account, depositor as R
where T.cname = R.cname and
R.acc#= account.acc#and
bname = 'perryridge')
```

(۴)

```
select cname
from depositor as T, account
where T.acc#= account.acc#and
bname = 'perryridge'
group by cname
having count (Distinct bname)<=1
```

(۳)

۳۹- پایگاه داده‌ای با جداول زیر ایجاد شده است: (مهندسی IT- دولتی ۸۷)

موجودیت شعب بانک Branch (branch - name , branch - city , assets)

موجودیت حساب پس‌انداز Account (account - number , branch - number , balance)

ارتباط حساب پس‌انداز با نام مشتریان depositor(customer - name , account - number)

آنگاه شرح مناسب برای Query روبه‌رو کدام گزینه می‌باشد؟

```
select distinct S.customer- name
from depositor as S
where not exists (
  (select branch-name
   from Branch
   where branch-city= 'Brooklyn')
Except
(select R.branch-name
 from depositor as T, account as R
 where T.account-number= R.account-number
 and S.customer-name= T.customer-name))
```

- ۱) اسامی مشتریانی را می‌دهد که در تمام شعب شهر Brooklyn حساب باز کرده‌اند.
- ۲) اسامی مشتریانی را می‌دهد که در هیچ یک از شعب شهر Brooklyn حساب باز نکرده‌اند.
- ۳) اسامی مشتریانی را می‌دهد که در شعب شهر Brooklyn به صورت تکراری حساب باز کرده‌اند.
- ۴) اسامی مشتریانی را می‌دهد که در شعب شهر Brooklyn به صورت غیر تکراری حساب باز کرده‌اند.

۴۰- رابطه R(a,b,c) را در نظر بگیرید: (مهندسی کامپیوتر- دولتی ۸۹)

۱) پاسخ Q₁ زیر مجموعه‌ای از پاسخ Q₂ است.

۲) Q₁ و Q₂ پاسخ‌های یکسان تولید می‌کنند.

۳) پاسخ Q₂ زیر مجموعه‌ای از پاسخ Q₁ است.

۴) Q₁ و Q₂ پاسخ‌های متفاوت تولید می‌کنند.

بانک اطلاعاتی تولیدکنندگان و قطعات شامل روابط زیر است:

S(S#, SNAME , STATUS , CITY) تولیدکننده
 S#: شماره تولیدکننده
 SNAME: اسم تولیدکننده
 STATUS: اعتبار تولیدکننده
 CITY: شهر تولیدکننده
 P(P#, PNAME , COLOR , WEIGHT , CITY) قطعه
 P#: شماره قطعه
 PNAME: اسم قطعه
 COLOR: رنگ قطعه
 WEIGHT: وزن قطعه
 CITY: شهر محل انبار کردن قطعه
 SP(S#, P#, QTY) تولید

QTY تعداد قطعه شماره P# تولید شده توسط تولید کننده شماره S# است.

۴۱- جمله SQL زیر در بانک اطلاعاتی تولید کنندگان و قطعات چه کار می‌کند؟ (مهندسی IT-دولتی ۸۹)

```
SELECT DISTINCT SNAME
FROM S
WHERE S.S# IN (SELECT SP.S#
                FROM SP
                WHERE SP.P# IN (SELECT P.P#
                                FROM P
                                WHERE PCOLOR= 'RED'))
```

(۱) اسامی تولید کنندگانی که بیش از یک قطعه قرمز رنگ تولید می‌کنند.

(۲) اسامی تولید کنندگانی که حداکثر یک قطعه قرمز رنگ تولید می‌کنند.

(۳) اسامی تولید کنندگانی که حداقل یک قطعه قرمز رنگ تولید می‌کنند.

(۴) هیچ کدام

۴۲- جمله SQL زیر در بانک اطلاعاتی تولید کنندگان و قطعات چه کار می‌کند؟ (مهندسی IT-دولتی ۸۹)

```
SELECT S.SNAME
FROM S
WHERE EXISTS (SELECT *
              FROM SP
              WHERE SP.S# = S.S# AND SP.P# = 'P2')
```

(۱) اسامی تولید کنندگانی که قطعه‌ای غیر از 'P2' هم تولید می‌کنند.

(۲) اسامی تولید کنندگانی که قطعه 'P2' را تولید می‌کنند.

(۳) اسامی تولید کنندگانی که قطعه 'P2' را تولید نمی‌کنند.

(۴) اسامی تولید کنندگانی که قطعه 'P2' را تولید می‌کنند بدون تکرار.

۴۳- کدام عبارت پاسخ پرس و جوی زیر را به درستی بیان می‌کند: (مهندسی IT-آزاد ۸۹)

I. select * مشخصات قطعاتی که با دور 1000 یا 2000 کار می‌کنند؟

```
from product
where spin = 1000 or spin= 2000
```

II. select *
from product
where spin in (1000, 2000)

III. select *
from product
where spin=1000
union
select *
from product
where spin=2000

I, II, III (۴)

III, II (۳)

I,II (۲)

I (۱)

(مهندسی IT - آزاد ۸۹)

۴۴- خروجی دستور زیر چند سطر خواهد داشت؟

```
select T1.B
from T1,T2
where T1.B=T2.B
```

T1	
A	B
a1	b1
a2	b2
a3	b3

T2	
B	C
b1	c1
b1	c2
b2	c2
b2	c3
b2	c4

15 (۴)

5 (۳)

3 (۲)

2 (۱)

(مهندسی IT - آزاد ۸۹)

۴۵- خروجی پرس وجوی زیر چیست؟

```
select distinct pname
from enroll
where crsno in (select crsno
from enroll
where sname= 'Kazemi')
```

(۱) اسامی استادانی که یکی از درس هایی را تدریس می کنند که کاظمی در آن درس ها ثبت نام کرده است.

(۲) اسامی استادان دانشجویی به نام کاظمی

(۳) اسامی استادانی که دقیقاً همان درس هایی را تدریس می کنند که کاظمی در آن درس ها ثبت نام کرده است.

(۴) اسامی استادانی که تمام درس هایی را تدریس می کنند که کاظمی در آن درس ها ثبت نام کرده است.

۴۶- فرض کنید که $R = (A, B)$ و $S = (A, C)$ و فرض کنید $r(R)$ و $s(S)$ دو رابطه باشند. عبارت $\{ \langle a \rangle \mid \exists c (\langle a, c \rangle \in s \wedge \exists b_1, b_2 (\langle a, b_1 \rangle \in r \wedge \langle c, b_2 \rangle \in r \wedge b_1 > b_2)) \}$ در حساب رابطه ای

دامنه ای (domain relational calculus)، معادل کدام یک از پرس وجوهای SQL زیر نیست؟

(مهندسی IT - دولتی ۹۰)

```
SELECT DISTINCT A
FROM s
WHERE (A,C) IN {
  SELECT r1.A, r2.A
  FROM r as r1, r as r2
  WHERE r1.B > r2.B
}
```

(۱)

```
SELECT DISTINCT A
FROM s
WHERE EXIST {
  SELECT *
  FROM r as r1, r as r2
  WHERE r1.A = s.C AND r2.A = s.A
  AND r2.B > r1.B
}
```

(۲)

```
SELECT DISTINCT A
FROM (r NATURAL INNER JOIN s) as
res (A,B,C)
WHERE EXIST {
  SELECT *
  FROM r
  WHERE r.A = res.C AND r.B > res.B
}
```

(۳)

```
SELECT DISTINCT r1.A
FROM r as r1, r as r2
WHERE r1.B > r2.B AND
(r1.A, r2.A) IN {
  SELECT *
  FROM s
}
```

(۴)

۴۷- فرض کنید در یک دفتر مسافرتی نام مسافری به نام «اکبرزاده» به اشتباه «اکبرزاده» وارد جدول مشخصات مسافران شده است، با کدام یک از دستورهایی زیر می‌توان این اشتباه را تصحیح کرد؟ (مهندسی IT - آزاد ۹۰)

Insert (۱) Update (۲) Delete (۳) Select (۴)

۴۸- اگر جدول ثبت‌نام به صورت زیر باشد: Enroll (s#, c#, term) کدام گزینه پاسخ پرس‌وجوی «تعداد دروسی که در نیم‌سال 892 ارائه شده‌اند به روش SQL می‌باشد؟» (مهندسی IT - آزاد ۹۰)

<pre>select distinct (count c#) from Enroll where term = 892</pre>	<pre>select count (distinct c#) from Enroll where term = 892</pre>
(۲)	(۱)
<pre>select distinct (c#) from Enroll where term = 892</pre>	<pre>select count (c#) from Enroll where term = 892</pre>
(۴)	(۳)

۴۹- کدام عبارت پاسخ پرس‌وجوی زیر را به درستی بیان می‌کند؟ «مشخصات تولید کنندگانی که در شهر همدان یا تبریز مستقرند.» (مهندسی IT - آزاد ۹۰)

I. select *
from producer
where city= 'Hamadan' or city= 'Tabriz'

II. select *
from producer
where city in ('Hamadan', 'Tabriz')

III. select *
from producer
where city= 'Hamadan'
union
select *
from producer
where city= 'Tabriz'

I, II, III (۴) I (۳) II (۲) I, II (۱)

۵۰- خروجی دستور زیر چند سطر خواهد داشت:

(مهندسی IT - آزاد ۹۰)

```
Select T1.B
from T1 left Inner Join T2
where T1.B=T2.B
```

T1	T2	
A	B	C
a ₁	b ₁	c ₁
a ₂	b ₂	c ₂
a ₃	b ₃	c ₂
	b ₂	c ₃
	b ₂	c ₄

6 (۱)

2 (۲)

5 (۳)

15 (۴)

۵۱- خروجی پرس و جوی زیر چیست؟

(مهندسی IT - آزاد ۹۰)

```
select distinct pname
from enroll
where crsno in (select crsno
from enroll
where sname = 'zakeri')
```

(۱) اسم فقط یکی از استادان دانشجویی به نام ذاکری.

(۲) اسامی استادانی که یکی از درس‌هایی را تدریس می‌کنند که ذاکری در آن درس‌ها ثبت‌نام کرده است.

(۳) اسامی استادانی که دقیقاً همان درس‌هایی را تدریس می‌کنند که ذاکری در آن درس‌ها ثبت‌نام کرده است.

(۴) اسامی استادانی که تمام درس‌هایی را تدریس می‌کنند که ذاکری در آن درس‌ها ثبت‌نام کرده است.

۵۲- با توجه به شش رابطه زیر:

Branch-schema = (branch-name, branch-city, assets)

Customer- schema = (customer-name, customer-street, customer - city)

Loan-schema= (loan-number, branch- name, amount)

Borrower-schema= (customer-name loan- number)

Account-schema= (account-number, branch-name, balance)

Depositor-schema= (customer-name, account-number)

برای پرس و جوی «نام همه مشتریانی که در همه شعب Brooklyn حساب دارند.» برای این که SQL زیر

جواب درستی برای سؤال مطرح شده باشد، کدام یک از عملگرهای SQL گزینه‌های درست برای متغیرهای

(مهندسی IT - دولتی ۹۱)

XX و YY در کد SQL زیر هستند؟

```
select distinct S.customer-name
from depositor as S
where XX (
(select branch-name
from branch
where branch-city= 'Brooklyn')
YY
```

```
(select R.branch-name
from depositor as T, account as R
where T.account-number= R.account-number
and S.customer-name= T.customer-name)
)
```

XX = exists, YY = except (۲) XX = unique, YY = union (۱)

XX = not unique, YY = union (۴) XX = not exists, YY = except (۳)

۵۳- فرض کنید بانک اطلاعاتی یک اپراتور مخابراتی حاوی جداول ذیل است.

(نام کارمند: empname، کد پرسنلی: emp#) emp کارمندان

(نام صاحب خط: custName، شماره تلفن ثابت: #file) fixtel تلفن های ثابت

(نام صاحب خط: custName، شماره تلفن موبایل: #Mtel) Mobtel تلفن های موبایل

کدام یک از دستورات SQL زیر برای استخراج «لیست کارمندانی از شرکت که تلفن موبایل آن اپراتور (شرکت) را ندارند» درست است؟ (مهندسی کامپیوتر- آزاد ۹۱)

(۱)

```
select * from emp where empname
not in (select custName from Mobtel)
```

(۲)

```
select * from emp, Mobtel
where emp.empname < > Mobtel.custName
```

(۳)

```
select * from emp, Mobtel
where emp.empname= Mobtel.custName
```

(۴)

```
select * from emp, fixtel
where emp.empname < > fixtel.custName
```

با توجه به بانک زیر، به سؤال زیر پاسخ دهید.

Person (pID , pName , sex, DoB)

اطلاعات فرد شامل شناسه pID، نام pName، جنسیت sex و تاریخ تولد DoB

Rel (pID1, pID2 , pRel)

اطلاعات ارتباط بین افراد- رابطه pRel بین شخص pID1 و pID2 برقرار است.

به طور خاص، pRel=1 به این معنی است که شخص با شناسه pID2 پدر شخص با شناسه pID1 است.

(مهندسی کامپیوتر- دولتی ۹۲)

۵۴- دستور SQL زیر تاریخ تولد کدام گزینه را می دهد؟

```
SELECT DoB
FROM Person
WHERE pName = x AND pID IN (SELECT pID2
FROM Rel
WHERE pRel=1)
```

(۱) پدر x

(۲) فرزند x

(۳) x در صورتی که پدر داشته باشد.

(۴) x در صورتی که فرزند داشته باشد.

۵۵- شمای پایگاه داده زیر را در نظر بگیرید:

کارمندان : **Employee(eid, NN, ename)**

(هر کارمند یک شماره یکتا (eid) دارد و برای هر کارمند شماره ملی (NN) و نام (ename) او را نگه می‌داریم.)

بخش‌ها : **Department(did, dname, address)**

(هر بخش یک شماره یکتا (did) دارد و برای هر بخش نام (dname) و آدرس (address) آن را نگه می‌داریم.)

کارمندان هر بخش : **WorksIn(eid, did, salary)**

(هر کارمند در چه بخشی کار می‌کند و حقوق (salary) او چقدر است.)

پرس‌وجوی SQL زیر را در نظر بگیرید.

```
SELECT AVG (R.s)
FROM (SELECT SUM (W.salary) AS s
FROM Employee E , WorksIn W
WHERE E.eid=W.eid
GROUP BY W.did
UNION ALL
SELECT 0 AS s
FROM Department D
WHERE NOT EXISTS (SELECT *
FROM WorksIn W1
WHERE W1.did =D.did)) AS R;
```

نتیجه اجرای این پرس‌وجو بر نمونه زیر از جدول‌های Employee و Department، WorksIn چه خواهد بود؟ (مهندسی کامپیوتر- دولتی ۹۳)

Employee			Department			WorksIn		
eid	NN	ename	did	dname	address	eid	did	Salary
e1	111111	John	d1	Books	a1	e1	d1	1
e2	222222	Jack	d2	Electronics	a2	e2	d1	3
e3	333333	Bob	d3	Health	a3	e3	d1	4
e4	444444	Joe				e4	d3	4

2 (۴) 3 (۳) 4 (۲) 6 (۱)

۵۶- با داشتن رابطه **person (personID, Name, MotherID, FatherID)** که اطلاعات افراد در آن ذخیره می‌شود و صفت‌های **FatherID** و **MotherID** که به رکوردهای همین رابطه اشاره می‌کنند، پرس‌وجوی SQL زیر چه جوابی را بر می‌گرداند؟ (مهندسی IT- دولتی ۹۳)

```
SELECT C.Name AS Name, F.Name AS Father, M.Name AS Mother
FROM person AS C LEFT OUTER JOIN person AS F ON C.FatherID=F.PersonID
INNER JOIN person AS M ON C.MotherID=M.PersonID
```

(۱) همه پدرهایی که برای آن‌ها فرزند ثبت نشده است.

- (۲) همه مادرهایی که برای آنها فرزند ثبت نشده است.
 (۳) همه افراد، حتی آنهایی که مادر برای آنها ثبت نشده است.
 (۴) همه افراد، حتی آنهایی که پدر برای آنها ثبت نشده است.

۵۷- هر تاپل در رابطه `supplier (sid, pid, sdate)` نشان می‌دهد که تهیه کننده `sid` قطعه `pid` را در تاریخ `sdate` تهیه کرده است. کلید اصلی این رابطه ترکیب هر سه صفت تشکیل دهنده آن است. کدام پرس و جوی SQL زیر معادل این درخواست است؟
 (مهندسی IT - دولتی ۹۳)

«`sid` تهیه کنندگانی که بعضی قطعات را بیش از ده بار تهیه کرده‌اند.»

<pre>select distinct sid from supplier where count(*) > 10</pre>	<pre>select distinct sid from supplier group by sid, pid having count(*) > 10</pre>
---	--

<pre>select distinct sid from supplier where count(distinct sdate) > 10 group by sid, pid</pre>	<pre>select distinct s1.sid from supplier as s1, supplier as s2 where count(*) > 10 and s1.pid = s2.pid</pre>
--	--

۵۸- سه رابطه `suppliers` (تهیه کنندگان)، `parts` (قطعات) و `catalog` (کاتالوگ) را در نظر بگیرید:

`suppliers (sid, sname)`

`parts (pid, pname)`

`catalog (sid, pid)`

`sid` در `catalog` کلید خارجی و ارجاع دهنده به `suppliers` و `pid` در `catalog` کلید خارجی و ارجاع دهنده به `parts` می‌باشد. وجود رکورد `(s1, p1)` در جدول `catalog` نشان دهنده این است که تهیه کننده `s1` قطعه `p1` را تولید می‌کند.
 (مهندسی کامپیوتر - دولتی ۹۴)

اگر بخواهیم دستور SQL زیر مشخصات قطعاتی را بدست آورد که توسط بعضی از تهیه کنندگان تولید نمی‌شود، در دو جای خالی (A) و (B) چه عملگرهایی باید قرار گیرد؟

```
SELECT *
FROM parts P
WHERE .....(A).....(SELECT S.sid
FROM suppliers S
WHERE ....(B)..(SELECT *
FROM catalog C
WHERE S.sid=C.sid AND P.pid = C.pid))
```

EXISTS : (B)	NOT EXISTS : (A) (۱)
EXISTS : (B)	EXISTS : (A) (۲)
NOT EXISTS : (B)	EXISTS : (A) (۳)
NOT EXISTS : (B)	NOT EXISTS : (A) (۴)

۵۹- با فرض وجود جداول S, P, J و SPJ به صورت زیر:
 (مهندسی IT - دولتی ۹۴)
 جدول تولید کنندگان
 S(S#, SNAME, STATUS, CITY)
 جدول قطعات
 P(P#, PNAME, WEIGHT, CITY)
 جدول پروژه‌ها
 J(J#, JNAME, CITY)
 جدول تعداد قطعات تولید شده توسط تولیدکنندگان در پروژه‌ها
 SPJ(S#, P#, J#, QTY)
 نتیجه اجرای پرس‌وجوی SQL زیر بر روی جداول فوق، کدام یک از عبارات زیر است؟

```
SELECT S.SNAME
FROM S NATURAL JOIN SPJ
WHERE SPJ.J#='j2'
GROUP BY S#, S.SNAME
HAVING SUM (QTY)<=All (SELECT SUM(QTY)
FROM SPJ
WHERE SPJ.J#='j2'
GROUP BY S#)
```

- (۱) نام تولیدکنندگانی که تعداد قطعه تولیدی آنها در پروژه کد j2 از مجموعه همه قطعات تولید شده توسط دیگر تولیدکنندگان بیشتر است.
 (۲) نام تولیدکنندگانی که کمترین تعداد قطعه (از تمام انواع قطعات) را برای پروژه کد j2 تولید کرده‌اند.
 (۳) نام تولیدکنندگانی که بیشترین تعداد قطعه (از تمام انواع قطعات) را برای پروژه کد j2 تولید کرده‌اند.
 (۴) نام تولیدکنندگانی که هیچ قطعه‌ای از پروژه کد j2 تولید نکرده‌اند.

۶۰- با توجه به پایگاه داده زیر، دستور SQL برای یافتن «نام اعضای که همه کتاب‌های منتشر شده توسط MC Graw-Hill را به امانت برده‌اند»، کدام است؟
 (مهندسی کامپیوتر-دولتی ۹۵)

Member (mem-no, name, age)
 Book (ISBN, title, auther, publisher)
 Borrowed (mem-no, ISBN, date)

(۱)
 SELECT A.name FROM Member A WHERE
 NOT EXISTS (SELECT isbn FROM Book, Borrowed B WHERE publisher = MCGraw - Hill
 AND B.mem - no ≠ A.mem - no)
 (۲)

(۳)
 SELECT A.name FROM Member A WHERE
 NOT EXISTS (SELECT isbn FROM Book, Borrowed B WHERE publisher = MCGraw - Hill
 AND B.mem - no = A.mem - no)
 (۴)

(۴)
 SELECT A.name FROM Member A WHERE
 NOT EXISTS ((SELECT isbn FROM Book
 WHERE publishe = McGraw - Hill) EXCEPT (SELECT isbn FROM Borrowed B
 WHERE B.mem - no = A.mem - no))

(۴)
 SELECT A.name FROM Member A WHERE
 NOT EXISTS ((SELECT isbn FROM Borrowed B
 WHERE B.mem - no = A.mem - no) EXCEPT (SELECT isbn FROM Book
 WHERE publishe = McGraw - Hill))

۶۱- جدول S جدول تولیدکنندگان، جدول P جدول قطعات و جدول SP جدول ارتباط دهنده آن دو جدول است که مشخص می‌کند چه تولیدکننده‌ای چه قطعه‌ای تولید کرده‌است. کدام گزینه شماره تولیدکنندگانی را می‌دهد که همه قطعات را تولید کرده‌اند؟ (مهندسی IT-دولتی ۹۵)

S (S#, SNAME, CITY)

P (P#, PNAME, COLOR)

SP (S#, P#)

(۱)

SELECT S# FROM S WHERE S# IN (SELECT S# FROM SP)

(۲)

SELECT S# FROM SP GROUP BY S# HAVING COUNT(P#) =
(SELECT COUNT(P#) FROM SP)

(۳)

SELECT S# FROM S WHERE EXISTS (SELECT P# FROM P
WHERE EXISTS (SELECT * FROM SP
WHERE S.S# = SP.S# AND P.P# = SP.P#))

(۴)

SELECT S# FROM S WHERE NOT EXISTS (SELECT P# FROM P
WHERE NOT EXISTS (SELECT * FROM SP
WHERE S.S# = SP.S# AND P.P# = SP.P#))

۶۲- سه رابطه‌ی قایق، قایقران و رزرو را در نظر بگیرید. می‌خواهیم نام قایقران‌هایی را بیابیم که همه قایق‌ها را رزرو کرده‌اند. کدام مورد، دستور SQL درست برای این درخواست است؟ (مهندسی کامپیوتر-دولتی ۹۶)

Sailors (sid, sname) : قایقران

Reservations (sid, bid, day) : رزرو

Boats (bid, bname, color) : قایق

(۱)

Select sname from Sailors S where not exists
((Select bid from Boats B) except
(Select bid from Reservations R where R.bid = B.bid AND R.sid = S.sid))

(۲)

Select sname from Sailors S where exists
((Select bid from Boats B) except
(Select bid from Reservations R where R.bid = B.bid AND R.sid = S.sid))

(۳)

Select sname from Sailors S where not exists
(Select bid from Boats B where not exists
(Select * from Reservations R where R.bid = B.bid AND R.sid = S.sid))

(۴)

Select sname from Sailors S where exists
(Select bid from Boats B where exists
(Select * from Reservations R where R.bid = B.bid AND R.sid = S.sid))

۶۳- کدام مورد، لیست نام دانشجویان به همراه معدل وزنی نمرات دروس اخذ شده آنها را در ترم 1-95-96 می دهد؟
(مهندسی IT-دولتی ۹۶)

ST (STID, STName,...): دانشجویان

CO (COID, CName, Credit,...): دروس

STCO (STID, COID, Semester, Grade): دروس انتخاب شده توسط دانشجویان

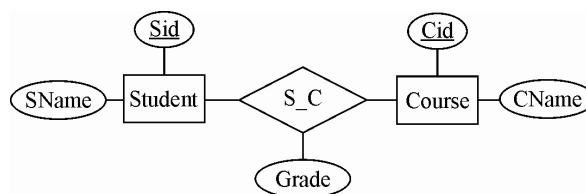
```
Select STName, SUM(Grade * Credit) / SUM(Credit)
From ST JOIN STCO JOIN CO
Group By STID, STName, Semester
Having Semester = '95 - 96 - 1' (۱)
```

```
Select STName, AVG(Grade)
From ST JOIN STCO JOIN CO
Group By STID, STName, Semester
Having Semester = '95 - 96 - 1' (۲)
```

```
Select STName, AVG(Grade * Credit)
From ST JOIN STCO JOIN CO
Where Semester = '95 - 96 - 1' (۳)
Group By STID, STName
```

```
Select STName, SUM(Grade * Credit) / SUM(Credit)
From ST JOIN STCO JOIN CO
Where Semester = '95 - 96 - 1' (۴)
Group By STID, Semester
```

۶۴- با توجه به نمودار ER داده شده، کدام مورد برای پرس و جو «نام دانشجویانی که معدل آنها از میانگین نمرات همه دروس دانشگاه بیشتر است» نادرست است؟
(مهندسی کامپیوتر-دولتی ۹۷)



```
Select SName
From Student T1, (Select S_C.Sid
From S_C
group By S_C.Sid
having AVG(grade) > (Select AVG(grade)
From S_C)) T2 (۱)
```

Where T1.Sid = T2.Sid

```

Select SName
From Student T1
Where Exists (Select '1'
              From S_C
              Where T1.Sid = S_C.Sid
              group by S_C.Sid
              having AVG(grade) > (Select AVG(grade)
                                   From S_C))

```

```

Select SName
From Student
Where Sid in (Select Sid
              From S_C
              group by Sid
              having AVG(grade) > (Select AVG(grade)
                                   From S_C))

```

```

Select SName
From Student, S_C
Where Student.Sid = S_C.Sid AND AVG(grade) > (Select AVG(grade)
                                               From S_C)

```

۶۵- رابطه‌های جدول داده شده را در نظر بگیرید. با اجرای دستور زیر حداقل و حداکثر تعداد سطرهاى خروجی کدام است؟

(مهندسی کامپیوتر-دولتی ۹۷)

نام جدول	تعداد سطرها
Student (Stid, StName,...)	$K > 0$
Course (Cid, CName,...)	$N > 0$
StudentCourse (Stid, Cid, grade)	$M > 0$

(۱) حداقل K و حداکثر $K+N-1$

(۲) حداقل ۱ و حداکثر $K+N-1$

(۳) حداقل N و حداکثر $K+N$

(۴) حداقل K و حداکثر $K+N$

۶۶- مطابق جدول، با اجرای دستور زیر، تعداد سطرهاى خروجی کدام است؟

(مهندسی IT-دولتی ۹۷)

```

select distinct Student.* from student, studentCourse

```

نام جدول	تعداد سطرها
Student (Stid, StName,...)	K
Course (Cid, CName,...)	N
StudentCourse (Stid, Cid, grade)	M

(۱) K

(۲) 0

(۳) $K.N$

(۴) گزینه‌های ۱ و ۲ می‌توانند صحیح باشند.

۶۷- با توجه به گزاره‌های داده شده، کدام مورد درست است؟

(مهندسی IT-دولتی ۹۷)

(a) اگر رابطه R ، دارای n خصیصه باشد، آنگاه تعداد ابرکلیدهای آن حداقل یک و حداکثر $2^n - 1$ است.

(b) صفت مرکب، صفتی است که مقدار آن از مقدار سایر صفت‌ها محاسبه می‌شود.
 (c) تعداد تاپل‌های عبارت $\Pi_{R,\alpha}(R \times T)$ همواره برابر با تعداد سطرهای α From R است.

$$\sigma_c(\Pi_\alpha(R)) = \Pi_\alpha(\sigma_c(R)) \quad (d)$$

- (۱) یک گزاره نادرست است. (۲) دو گزاره نادرست است.
 (۳) سه گزاره نادرست است. (۴) همه گزاره‌ها نادرست هستند.

۶۸- با توجه به پایگاه داده زیر، کدام کد SQL داده شده، لیست تمام کتاب‌هایی را نمایش می‌دهد که تمام نسخه‌های آنها امانت گرفته شده است؟ (مهندسی کامپیوتر- دولتی ۹۸)

user (UID , name , Contact)
 BOOK (BID , Title , Publisher , TotalNumber)
 BORROWING (UID , BID , StartDate , EndDate)

وقتی کتابی هنوز در امانت است تاریخ خاتمه NULL است و به محض تحویل پُر می‌شود.

- a) with tbl as (
 Select BORROWING.BID, COUNT(BORROWING.BID) as cnt
 From BORROWING, BOOK
 Where BOOK.BID = BORROWING.BID and EndDate is NULL
 Group by BORROWING.BID)
 Select tbl.BID
 From tbl, BOOK
 Where tbl.BID = BOOK.BID and BOOK.TotalNumber = cnt
- b) Select BID
 From BOOK b1
 Where (select COUNT (UID)
 From BORROWING
 Where BORROWING.BID = b1.BID and
 BORROWING.EndDate is NULL) = b1.TotalNumber
- c) Select BID, TotalNumber
 From BORROWING, BOOK
 Where BOOK.BID = BORROWING.BID and EndDate is NULL
 Group by BORROWING.BID
 Having COUNT(BID) = TotalNumber

c , b , a (۴) c , b (۳) c , a (۲) b , a (۱)

۶۹- بانک اطلاعاتی ماشین‌ها و تصادفات روبه‌رو را در نظر بگیرید: (مهندسی کامپیوتر- دولتی ۹۸)

Person (SSN , name , address)
 Car (License , year , model)
 Accident (License , accident - date , driver , damage - amount)
 Owns (SSN , License)

با توجه به بانک اطلاعاتی ماشین‌ها و تصادفات، کدام راننده پرهزینه‌ترین تصادف را داشته است؟
(نام راننده و میزان خسارت برگردانده شود به زبان SQL)

```
(select driver, damage - amount
from Accident) Except (select a.driver, a.damage - amount
from Accident a, Accident b
where a.damage - amount < b.damage - amount and a.driver <> b.driver) (۱)
```

```
select driver, damage - amount
from Accident
where damage - amount in (select MAX(damage - amount)
from Accident) (۲)
```

```
select driver, damage - amount
from Accident
where damage - amount = MAX(damage - amount) (۳)
```

(۴) همه موارد صحیح است.

۷۰- بانک اطلاعاتی ماشین‌ها و تصادفات روبه‌رو را در نظر بگیرید: (مهندسی کامپیوتر- دولتی ۹۸)

Person (SSN , name , address)

Car (License , year , model)

Accident (License , accident - date , driver , damage - amount)

Owns (SSN , License)

با توجه به بانک اطلاعاتی ماشین‌ها و تصادفات، کدام شماره پلاک ماشین در بیش از یک تصادف، درگیر بوده است؟

(پاسخ به زبان SQL و سطرهای تکراری فقط یک‌بار نشان داده شوند.)

```
select distinct A.License
from Accident A
where A.License in (select B.License
from Accident B
where A.accident - date <> B.accident - date) (۱)
```

```
select License
from Accident
group by License
having count (accident - date) > 1 (۲)
```

```
select A.License
from Accident A, Accident B
where A.License = B.License and A.accident - date <> B.accident - date (۳)
```

(۴) موارد ۱ و ۲ صحیح است.

۷۱- پایگاه داده زیر را در نظر بگیرید: (مهندسی IT- دولتی ۹۸)

اطلاعات موجود در جدول node شامل شماره، نام، رنگ و شرح مربوط به هر گره است. node (NID , Name , Color , Description): گره‌ها

edge (NID1, NID2, EdgeType): یال‌ها

هر سطر از جدول edge، نشان دهنده وجود یک یال جهت‌دار از نوع EdgeType از گره با شماره NID1 به گره با شماره NID2 است.

در خصوص پرس و جوهای SQL زیر کدام گزینه درست است؟

پرس و جوی اول	<pre>select distinct NID from node, edge where NID=edge.NID2 AND not exists (select * from edge where edge.NID1= NID)</pre>
پرس و جوی دوم	<pre>select T1.NID from (select count (NID1) as cnt , NID from node left outer join edge on edge.NID1=NID group by NID) T1 , (select count (NID2) as cnt , NID from node left outer join edge on edge.NID2=NID group by NID) T2 where T1.NID = T2.NID and T1.cnt < T2.cnt</pre>

- (۱) پرس و جوی اول، شماره گره‌هایی را می‌دهد که یال خروجی دارند اما یال ورودی ندارند.
 (۲) پرس و جوی دوم، شماره گره‌هایی را می‌دهد که درجه ورودی آنها کمتر از درجه خروجی آنها است.
 (۳) پرس و جوی دوم، شماره گره‌هایی را می‌دهد که درجه خروجی آنها کمتر از درجه ورودی آنها است.
 (۴) گزینه‌های ۱ و ۳ صحیح هستند.

۷۲- با توجه به جداول داده شده زیر خروجی به چه صورت است؟

(مهندسی کامپیوتر- دولتی ۹۹)

City = شهر تولیدکننده

Color = رنگ قطعه

P# = شماره قطعه

S# = شماره تولیدکننده

QTY = تعداد تولیدات

Supplier = تولیدکننده

Supply = تولید

Part = قطعه

Supplier(S#, Sname , City)

Supply(S#, P#, QTY)

Part(P#, Color)

SELECT S#

FROM Supplier

WHERE S# = P#;

- ۱) شماره تولیدکنندگان و قطعات را که باهم یکسان هستند در خروجی ظاهر می‌کند.
- ۲) اجرا نمی‌شود، زیرا دو دامنه (Domain) متفاوت باهم مقایسه شده است.
- ۳) تمام تولیدکنندگانی که قطعه‌های یکسان تولید می‌کنند را بر می‌گرداند.
- ۴) تمام قطعات با شماره یکسان را بر می‌گرداند.

۷۳- با توجه به پایگاه داده زیر در مورد پرس و جوهای SQL زیر کدام مورد درست است؟

(مهندسی کامپیوتر- دولتی ۹۹)

جدول گره‌ها	Node(NID , Name , Color , Description) اطلاعات موجود در جدول گره‌ها شامل شماره، نام، رنگ و شرح مربوط به هر گره است.
جدول یال‌ها	Edge(NID1, NID2, EdgeType) هر سطر از جدول Edge، نشان دهنده وجود یک یال جهت‌دار از نوع EdgeType از گره با شماره NID1 به گره با شماره NID2 است.

پرس و جو اول	<pre>select distinct NID from node, edge where NID=edge.NID2 AND not exists (select * from edge where edge.NID1=NID)</pre>
پرس و جو دوم	<pre>select T1.NID from (select count (NID1) as cnt , NID from node left outer join edge on edge.NID1=NID group by NID) T1 , (select count (NID2) as cnt , NID from node left outer join edge on edge.NID2=NID group by NID) T2 where T1.NID = T2.NID and T1.cnt < T2.cnt</pre>

- ۱) پرس و جو اول، شماره گره‌هایی را می‌دهد که یال خروجی دارند اما یال ورودی ندارند.
- ۲) پرس و جو دوم، شماره گره‌هایی را می‌دهد که درجه خروجی آنها کمتر از درجه ورودی آنها است.
- ۳) پرس و جو دوم، شماره گره‌هایی را می‌دهد که درجه ورودی آنها کمتر از درجه خروجی آنها است.
- ۴) گزینه‌های ۱ و ۲ صحیح هستند.

۷۴- با توجه به پایگاه داده زیر، پرس و جوی «نام اعضای که همه کتاب‌های انتشارات Springer را امانت گرفته‌اند» را در نظر بگیرید. چند مورد از دستوره‌های SQL (الف، ب و ج) برای این پرس و جو درست است؟ (مهندسی کامپیوتر-دولتی ۹۹)

کتاب‌ها	Book (ISBN, Title, Auther, Publisher)
اعضا	Member (MID, Name, MDate)
امانت کتاب	Borrow (MID, ISBN, BDate, Duration)

(الف)

```

SELECT Name
FROM (SELECT Member.MID , Member.Name , count (Book.ISBN) as BorrowCount
      FROM Member , Borrow , Book
      WHERE Member.MID - Borrow.MID and Book.ISBN - Borrow.ISBN
      and Publisher - 'Springer'
      GROUP BY Member.MID , Member.Name) as M
WHERE BorrowCount - (SELECT count(*)
                     FROM Book
                     WHERE Publisher - 'Springer')

```

(ب)

```

SELECT Name
FROM Member
WHERE NOT EXISTS ((SELECT ISBN
                   FROM Book
                   WHERE Publisher = 'Springer')
                  EXCEPT
                  (SELECT ISBN
                   FROM Borrow
                   WHERE Borrow.MID = Member.MID))

```

(ج)

```

SELECT Name
FROM Member
WHERE NOT EXISTS (SELECT 1
                  FROM Book
                  WHERE Publisher = 'Springer'
                  And
                  NOT EXISTS (SELECT ISBN
                              FROM Borrow
                              WHERE Borrow.MID = Member.MID
                              And Borrow.ISBN = Book.ISBN))

```

(۱) صفر (۲) یک (۳) دو (۴) سه

۷۵- رابطه‌های $R1(A,B)$ و $R2(C,D)$ را در نظر بگیرید. کدام گزینه برقرار باشد، خروجی عبارت SQL زیر همان رابطه $R1$ است؟ (مهندسی IT- دولتی ۹۹)

```

Select Distinct A,B
From R1,R2

```

- (۱) هیچ تاپل تکراری ندارد و $R2$ تهی نیست.
- (۲) هیچ تاپل تکراری ندارد و $R1$ تهی نیست.
- (۳) $R1$ هیچ تاپل تکراری ندارد و $R2$ تهی است.
- (۴) هم $R1$ و هم $R2$ هیچ تاپل تکراری ندارند.

۷۶- این شمای پایگاه داده را در نظر بگیرید:

(مهندس کامپیوتر-دولتی ۱۴۰۰)

Student(sid, sname, age)

Course(cid, cname, credits)

Takes(sid, cid, grade)

می‌خواهیم sid دانشجویانی را پیدا کنیم که هم در درس Database و هم در درس Math ثبت نام کرده‌اند. کدام پرس و جوی SQL برای این منظور مناسب است؟

I. SELECT T1.sid
FROM Course C1, Takes T1
WHERE C1.cid = T1.cid AND C1.cname = 'Database'

INTERSECT

SELECT T2.sid
FROM Course C2, Takes T2
WHERE C2.cid = T2.cid AND C2.cname = 'Math'

II. SELECT T1.sid
FROM Course C1, Takes T1
WHERE C1.cid = T1.cid AND C1.cname = 'Database'
AND T1.sid IN (SELECT T2.sid
FROM Course C2, Takes T2
WHERE C2.cid = T2.cid AND C2.cname = 'Math')

III. SELECT T1.sid
FROM Course C1, Takes T1
WHERE C1.cid = T1.cid AND C1.cname = 'Database'
AND EXISTS (SELECT *
FROM Course C2, Takes T2
WHERE C2.cid = T2.cid AND C2.cname = 'Math'
AND C2.sid = C1.sid)

(۱) فقط I (۲) فقط II (۳) فقط I و II (۴) I و II و III

پاسخ تست‌های فصل ششم: SQL دستورات DML

۱- گزینه (۲) صحیح است.

برای پاسخ به این سؤال ابتدا لازم است انواع join مورد بحث و بررسی قرار گیرد:
جداول Employee و Department را در نظر بگیرید:

Emp#	EmpName	Salary	Dep#
101	Alavi	2	12
102	Hassani	4	12
103	Hosseini	6	13
104	Sajjadi	8	13
105	Asgari	10	14
106	Akbari	12	14

جدول Employee

Dep#	DepName
12	حسابداری
13	فروش
14	مهندسی

جدول Department

CROSS JOIN (ضرب دکارتی)

مثال اول:

```
select *
from employee CROSS JOIN department
```

مثال دوم:

```
select *
from employee, department
```

توجه: در CROSS JOIN همه سطرهای دو جدول در کنار هم قرار می‌گیرند.

Employee				Department	
Emp#	EmpName	Salary	Dep#	Dep#	DepName
⋮	⋮	⋮	⋮	⋮	⋮

INNER JOIN (ضرب دکارتی دارای شرط)

این عملگر در واقع همان ضرب دکارتی است که می‌تواند همراه با شرط مورد نظر برای پیوند جداول مورد استفاده قرار بگیرد.

مثال اول:

```
select *
from employee INNER JOIN department
on employee.Dep# = department.Dep#
```

توجه: در INNER JOIN می‌توان از =، >، < و >=، <= و <> استفاده نمود.

مثال دوم: (شبیه‌سازی)

```
select *
from employee, department
where employee.Dep# = department.Dep#
```

توجه: مطابق پرس وجوهای فوق، ابتدا دو جدول با یکدیگر ضرب دکارتی می شوند و سپس شرط داده شده روی سطرها اعمال می گردد. شرط داده شده در واقع همان شرط ضرب است. در خروجی پرس وجوهای فوق ستون مشترک دو بار ظاهر می گردد.

Employee				Department	
Emp#	EmpName	Salary	Dep#	Dep#	DepName
101	Alavi	2	12	12	حسابداری
102	Hassani	4	12	12	حسابداری
103	Hosseini	6	13	13	فروش
104	Sajjadi	8	13	13	فروش
105	Asgari	10	14	14	مهندسی
106	Akbari	12	14	14	مهندسی

EQUI JOIN (ضرب دکارتی دارای شرط فقط برابری)

این عملگر در واقع همان ضرب دکارتی است که می تواند همراه با شرط فقط برابری برای پیوند جداول مورد استفاده قرار بگیرد.

مثال اول:

```
select *
from employee JOIN department
on employee.Dep# = department.Dep#
```

توجه: در JOIN فقط = مورد استفاده قرار می گیرد و استفاده از <، >، = و <> مجاز نیست.

مثال دوم: (شبیه سازی)

```
select *
from employee, department
where employee.Dep# = department.Dep#
```

مثال سوم: (شبیه سازی)

```
select *
from employee INNER JOIN department USING (Dep#)
```

مثال چهارم: (شبیه سازی)

```
select *
from employee INNER JOIN department
on employee.Dep# = department.Dep#
```

توجه: در INNER JOIN می توان از =، <، >، = و <> استفاده نمود.

توجه: مطابق پرس و جوهای فوق، ابتدا دو جدول با یکدیگر ضرب دکارتی می شوند و سپس شرط برابری روی سطرها اعمال می گردد. شرط برابری در واقع همان شرط ضرب است. در خروجی پرس وجوهای فوق ستون مشترک دو بار ظاهر می شود.

Employee				Department	
Emp#	EmpName	Salary	Dep#	Dep#	DepName
101	Alavi	2	12	12	حسابداری
102	Hassani	4	12	12	حسابداری
103	Hosseini	6	13	13	فروش
104	Sajjadi	8	13	13	فروش
105	Asgari	10	14	14	مهندسی
106	Akbari	12	14	14	مهندسی

NATURAL JOIN (الحاق طبیعی)

این عملگر، الحاق طبیعی دو جدول را انجام می‌دهد.

مثال:

**select *
from employee NATURAL JOIN department**

توجه: مطابق پرس و جوی فوق، فقط سطرهایی از پیوند دو جدول در خروجی ظاهر می‌شوند که مقدارشان در ستون مشترک برابر باشد. (در واقع همان تعریف الحاق طبیعی در جبر رابطه‌ای). در خروجی پرس و جوی فوق ستون مشترک فقط یک بار ظاهر می‌شود.

Employee				Department	
Emp#	EmpName	Salary	Dep#	Dep#	DepName
101	Alavi	2	12	12	حسابداری
102	Hassani	4	12	12	حسابداری
103	Hosseini	6	13	13	فروش
104	Sajjadi	8	13	13	فروش
105	Asgari	10	14	14	مهندسی
106	Akbari	12	14	14	مهندسی

انواع پرس و جوهای تو در تو (nested query)

جداول زیر را در نظر بگیرید:

S#	Sname	City
S1	Sn1	C1
S2	Sn2	C2
S3	Sn3	C2

جدول s

S#	P#	QTY
S1	P1	10
S1	P2	20
S2	P1	30

جدول sp

Normal Subquery (الف)

مثال: نام تولیدکنندگانی که حداقل یک قطعه تولید کرده‌اند.

```

select Sname
from S
where S# in (select S#
             from SP
             where S# = SP.S#)

```

خروجی ⇒

Sname
Sn1
Sn2

ویژگی‌ها

۱- تکرار

ابتدا زیر پرس‌وجوی داخلی یک بار و برای همیشه اجرا می‌گردد، سپس پرس‌وجوی خارجی به ازای حرکت در هر یک از سطرهاى خود، از مقادیر زیر پرس‌وجوی داخلی استفاده می‌کند.

۲- وابستگی (outer to inner)

پرس‌وجوی خارجی، به زیر پرس‌وجوی داخلی وابستگی دارد.

۳- کارایی (Performance)

Normal Subquery کارایی بالاتری را نسبت به Correlated Subquery دارد.

۴- مرتبه اجرایی $O(N+M)$

ب) Correlated Subquery

مثال اول: نام تولید کنندگانی که حداقل یک قطعه تولید کرده‌اند:

```

select Sname
from S
where exists (select *
             from SP
             where S.S# = SP.S#)

```

خروجی ⇒

Sname
Sn1
Sn2

ویژگی‌ها

۱- تکرار

به ازای حرکت در هر یک از سطرهاى پرس‌وجوی خارجی، یک بار به طور کامل از ابتدا تا انتها زیر پرس‌وجوی داخلی اجرا و براساس شرطی که زیر پرس‌وجوی داخلی را به پرس‌وجوی خارجی متصل می‌کند، بررسی انجام می‌شود. مانند دو حلقه تو در تو، که به ازای هر بار اجرای حلقه خارجی، یک بار به طور کامل حلقه داخلی اجرا می‌گردد.

۲- وابستگی (Inner to Outer)

پرس‌وجوی داخلی توسط یک ستون به پرس‌وجوی خارجی متصل و وابسته می‌گردد.

۳- کارایی (Performance)

Correlated Subquery کارایی پایین‌تری را نسبت به Normal Subquery دارد.

بنابراین گزینه اول درست است.

۴- مرتبه اجرایی $O(M \times N)$

جداول Employee و Department را در نظر بگیرید:

Emp#	EmpName	Salary	Dep#
101	Alavi	2	12
102	Hassani	4	12
103	Hosseini	6	13
104	Sajjadi	8	13
105	Asgari	10	14
106	Akbari	12	14

جدول Employee

Dep#	DepName
12	حسابداری
13	فروش
14	مهندسی

جدول Department

مثال دوم: مشخصات کارمندانی که حقوقشان، از متوسط حقوق، کارمندان بخششان بیش تر است:

```
select *
```

```
from employee emp1
```

```
where salary > (select AVG (salary)
```

```
from employee emp2
```

```
where emp1.Dep# = emp2.Dep#)
```

توجه: زیر پرس وجوی فوق، یک Correlated Subquery است که زیر پرس وجوی مورد نظر را توسط ستون Dep# به پرس وجوی خارجی متصل می کند.

توجه: در پرس وجوی فوق، به ازای حرکت در هر یک از سطرها پرس وجوی خارجی، یک بار به طور کامل از ابتدا تا انتها، زیر پرس وجوی داخلی اجرا و براساس شرطی که زیر پرس وجوی داخلی را به پرس وجوی خارجی متصل می کند، بررسی انجام می شود.

توجه: پرس وجوی فوق سطر به سطر توسط عملگر where بر روی جدول emp1 (جدول employee) حرکت می کند و برای هر سطر از جدول emp1، شرط جلوی where را چک می کند، اگر شرط جلوی where برقرار بود، سطر مورد نظر از جدول emp1 انتخاب می گردد و این رویه تا به انتهای جدول emp1 ادامه پیدا می کند.

فرض کنید در سطر اول جدول emp1 قرار داریم....

Emp#	EmpName	Salary	Dep#
101	Alavi	2	12

با توجه به مقدار Dep# برابر 12 از سطر اول جدول emp1، پرس وجوی select داخلی، یعنی ...

```
select AVG (Salary)
```

```
from employee emp2
```

```
where emp1.Dep# = emp2.Dep#
```

12

میانگین حقوق کارمندانی را از جدول emp2 محاسبه می کند که dep# آنها برابر 12 باشد، به بیان دیگر زیر پرس وجوی فوق میانگین حقوق کارمندان بخش شماره 12 را محاسبه می کند. دو جدول emp1 و emp2 که

مطابق جدول employee هستند و پرس و جوی فوق را تجسم کنید. با توجه به مشخصات کارمندان بخش شماره 12، پرواضح است که میانگین حقوق کارمندان بخش شماره 12، برابر $\frac{2+4}{2} = \frac{6}{2} = 3$ می باشد. بنابراین کارمند شماره 101، برای نمایش در خروجی، انتخاب نمی گردد. زیرا:

```
select *
from employee emp1
where salary > (select AVG (salary)
                from employee emp2
                where emp1. Dep # = emp2. Dep #)
```

$$\frac{2+4}{2} = \frac{6}{2} = 3$$

مشاهده می کنید که شرط فوق برای سطر اول جدول emp1 برقرار نیست. بنابراین سطر اول از جدول emp1 حائز شرایط نیست و برای نمایش در خروجی انتخاب نمی گردد. این رویه تا انتهای جدول emp1 به صورت سطر به سطر، ادامه پیدا می کند. برای مثالی دیگر، سطر آخر جدول emp1 را هم در نظر بگیرید.

با توجه به مشخصات کارمندان بخش شماره 14، پرواضح است که میانگین حقوق کارمندان بخش شماره 14، برابر $\frac{10+12}{2} = \frac{22}{2} = 11$ می باشد. بنابراین کارمند شماره 106 برای نمایش در خروجی، انتخاب می گردد. زیرا:

```
select *
from employee emp1
where salary > (select AVG (salary)
                from employee emp2
                where emp1. Dep # = emp2. Dep #)
```

$$\frac{10+12}{2} = \frac{22}{2} = 11$$

مشاهده می کنید که شرط فوق برای سطر آخر جدول emp1 برقرار است. بنابراین سطر آخر از جدول emp1 حائز شرایط است و برای نمایش در خروجی، انتخاب می گردد. با توجه به مشخصات کارمندان بخش 13، پرواضح است که میانگین حقوق کارمندان بخش شماره 13 برابر $\frac{6+8}{2} = \frac{14}{2} = 7$ است. بنابراین خروجی پرس و جوی مذکور به صورت زیر خواهد بود:

Emp#	EmpName	Salary	Dep#
102	Hassani	4	12
104	Sajjadi	8	13
106	Akbari	12	14

گزینه دوم نادرست است. زیرا، هر Query که قابل بیان به صورت یک Natural Join می باشد، همواره این طور نیست که به صورت یک Subquery نیز قابل بیان باشد.

مثال: جداول زیر را در نظر بگیرید:

a	b
1	2
3	4
6	7

جدول P

b	c
2	5
2	7
4	8

جدول Q

توسط الحاق طبیعی:

```
select *
from P NATURAL JOIN Q
```

a	b	c
1	2	5
1	2	7
3	4	8

خروجی ⇒

توسط subquery :

```
select *
from P
where b in (select b
            from Q)
```

a	b
1	2
3	4

خروجی ⇒

~~2,2,4~~

توجه: زیرا subquery، یک زیر پرس وجوی داخلی است که مقادیر لازم برای پرس وجوی خارجی را فراهم می کند، اما در نهایت مشخصات پرس وجوی خارجی براساس شرط مورد نظر، راهی خروجی می گردد. در حالی که الحاق طبیعی، اجتماع ستون های دو جدول مورد نظر، در مقادیر برابر در ستون مشترک را راهی خروجی می کند.

گزینه سوم درست است. زیرا، هر Query که قابل بیان به صورت یک subquery می باشد، می تواند به صورت یک Equi Join نیز بیان شود. مثال: جداول زیر را در نظر بگیرید:

a	b
1	2
3	4
6	7

جدول P

b	c
2	5
2	7
4	8

جدول Q

توسط subquery :

```
select *
from P
where b in (select b from Q)
```

a	b
1	2
3	4

خروجی ⇒

توسط Equi Join :

```

select Distinct P.a , P.b
from P Join Q
on P.b = Q.b

```

خروجی \Rightarrow

a	b
1	2
3	4

گزینه چهارم درست است. زیرا، هر Query که قابل بیان به صورت یک Equi Join باشد می تواند به صورت یک Natural Join نیز بیان شود.
مثال: جداول زیر را در نظر بگیرید.

a	b
1	2
3	4
6	7

جدول P

b	c
2	5
2	7
4	8

جدول Q

توسط Equi Join :

```

select P.a, P.b , Q.c
from P Join Q
on P.b = Q.b

```

خروجی \Rightarrow

P.a	P.b	Q.b	Q.c
1	2	2	5
1	2	2	7
3	4	4	8

توسط Natural Join :

```

select *
from P NATURAL JOIN Q

```

خروجی \Rightarrow

a	b	c
1	2	5
1	2	7
3	4	8

۲- گزینه (۲) صحیح است.

گزینه های اول، سوم و چهارم معادل هستند و اسامی کارمندانی که مدیر پروژه نیز می باشند را استخراج می کند، اما گزینه دوم اسامی کارمندانی که فقط مدیر یک پروژه می باشند را استخراج می کند. گزینه دوم بیانگر این پرس و جو است که تنها یک پروژه موجود باشد که شرط $p.mgr=e.empno$ را تحقق بخشد که جهت پیاده سازی پرس و جوی مورد نظر مناسب نیست. به عبارت دیگر گزینه دوم نام کارمندانی را می دهد که فقط مدیر یک پروژه هستند، در حالی که ممکن است کارمندی مدیریت چند پروژه را بر عهده داشته باشد.

توجه: satisfies, یک کلمه شبه کد، و به معنای ارضای شرط مقابل می باشد.

۳- گزینه (۲) صحیح است.

همه گزینه ها به جز گزینه دوم، تاپل هایی از جدول P را بر می گرداند که P# آن ها با P# حداقل یکی از تاپل های جدول SP که مقدار شارژ آن بیش تر از 40000 است، برابر باشد. به عبارت دیگر اطلاعات قطعاتی که توسط تهیه کننده ای با شارژ بزرگتر از 40000 تولید شده است، استخراج می گردد.
گزینه دوم خطای نحوی هم دارد که اگر درست شود، با توجه به کلمه All این گزینه از سایر گزینه ها

متمايز می‌گردد. عبارت جلوی All باید به شکل زیر اصلاح گردد:

```
select SP.P# from supply SP where SP.charge > 40000
```

همچنین در گزینه دوم =All معنی ندارد و نمادهای <، >، <=، >= و <> می‌توانست قرار بگیرد. البته در گزینه چهارم نیز استفاده از زیر پرس وجو به صورت نادرستی انجام گرفته است. در این گزینه زیر پرس وجو به صورت کامل از پرس وجوی اصلی مستقل است. بنابراین نمی‌توان از شرط P.P#=SP.P# در زیر پرس وجو استفاده کرد. گزینه چهارم به طور کامل دارای ساختار نادرست و خطای نحوی و غیر قابل اصلاح می‌باشد و معنای درست آن در همان ساختار گزینه سوم قرار دارد. سازمان سنجش آموزش کشور گزینه دوم را به عنوان پاسخ درست اعلام کرده بود.

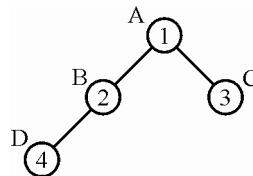
۴- گزینه (۴) صحیح است.

به مثال زیر توجه کنید:

no	name	Father-no
1	A	NULL
2	B	1
3	C	1
4	D	2

جدول person

توجه: درخت روابط پدری و فرزندی جدول فوق به شکل گویاتر به صورت زیر است:



توجه: واضح است که گره‌های 1 و 2 با نام‌های A و B پدر هستند و گره‌های 3 و 4 با نام‌های C و D به عنوان برگ‌ها پدر نیستند.

توجه: قطعه دستور from person p1 و person p2، دو کپی مجزا از جدول person را ایجاد می‌کند و در هم ضرب دکارتی می‌کند، به صورت زیر:

p ₁ -no	p ₁ -name	p ₁ -father-no	p ₂ -no	p ₂ -name	p ₂ -father-no
1	A	NULL	1	A	NULL
①	A	NULL	2	B	①
①	A	NULL	3	C	①
1	A	NULL	4	D	2
2	B	1	1	A	NULL
2	B	1	2	B	1
2	B	1	3	C	1
②	B	1	4	D	②

p ₁ -no	p ₁ -name	p ₁ -father-no	p ₂ -no	p ₂ -name	p ₂ -father-no
3	C	1	1	A	NULL
3	C	1	2	B	1
3	C	1	3	C	1
3	C	1	4	D	2
4	D	2	1	A	NULL
4	D	2	2	B	1
4	D	2	3	C	1
4	D	2	4	D	2

گزینه اول: نام تمام افرادی را می‌دهد که شماره شناسایی آن‌ها با شماره شناسایی پدرشان یکی باشد که بی‌معناست، کد ملی پدر و فرزند نمی‌شود یکسان باشد.

گزینه دوم: نام تمام افراد را استخراج می‌کند. در جدول فوق واضح است.

گزینه سوم: معادل گزینه اول است و نام تمام افرادی را می‌دهد که شماره شناسایی آن‌ها با شماره شناسایی پدرشان یکی باشد که بی‌معناست.

گزینه چهارم: نام تمام افرادی که شماره آن‌ها، در یکی از سطرها به عنوان شماره پدر درج شده است به عنوان خروجی داده می‌شود. به عبارت دیگر این گزینه نام همه پدرهای موجود در رابطه را می‌دهد. این مطلب در جدول فوق مشخص شده است. گره‌های ۱ و ۲ با نام‌های A و B پدر هستند.

توجه: در یک قاعده کلی، یکی از راه‌حل‌های مقایسه مقادیر یک ستون از یک جدول با مقادیر یک ستون دیگر از جدول دیگر، استفاده از ضرب دکارتی به همراه شرط ضرب می‌باشد. در این مساله هر دو ستون مورد مقایسه در یک جدول بودند که با ایجاد دو کپی مجزا از جدول person و ضرب دکارتی و بیان شرط ضرب، پرس‌وجوی گزینه چهارم ایجاد شده بود.

۵- گزینه (۲) صحیح است.

در صورتی که تابع عددی به شکل مستقیم مقابل دستور where قرار گیرد، پرس‌وجو دارای خطای نحوی خواهد بود. راه‌حل درست این پرس‌وجو در گزینه اول آمده است. نتیجه این که هرگاه تابع آماری بخواهد مقابل دستور where قرار بگیرد باید داخل یک select مستقل قرار بگیرد تا محاسبات آن به شکل مستقل انجام گردد. بهتر بود در گزینه دوم اشکال منطقی به صورت اشکال نحوی بیان می‌شد.

۶- گزینه (۳) صحیح است.

گزینه اول: اطلاعات سطرهای بی اعتبار در حاصل ضرب دکارتی دو جدول وام و مشتری را می‌دهد (چون فقط سطرهایی را در خروجی می‌دهد که مقدارشان در ستون مشترک مساوی نیست).

گزینه دوم: اطلاعات مشتریانی را می‌دهد که وام گرفته‌اند.

گزینه سوم: داخلی‌ترین پراتن نام مشتریانی را می‌دهد که وام گرفته‌اند و پراتن بیرونی مشخصات کامل مشتریانی را می‌دهد که وام نگرفته‌اند.

۷- گزینه (۳) صحیح است.

ضد نیم اتصال R و S تاپل‌هایی از رابطه R می‌باشد که در $R \bowtie S$ وجود ندارد. انتخاب تاپل‌هایی که در

$R \bowtie S$ وجود دارند یعنی تاپل‌های پیوندپذیر، بنا به تساوی هر دو فیلد A و B می‌باشد، یعنی $R.A=S.A$ and $R.B=S.B$ نشان دهنده تاپل‌های متناظر در R و S می‌باشند که در $R \bowtie S$ وجود دارند و طبق تعریف موجود در صورت سؤال، ضد نیم اتصال، رکوردهایی از R است که در این مجموعه تاپل‌ها (پیوندپذیر) وجود نداشته باشد. گزینه سوم تاپل‌های آویزان (پیوند ناپذیر) از رابطه R را در خروجی نشان می‌دهد.

هم‌چنین گزینه اول الزام می‌کند که هم مقدار خصیصه A و هم مقدار خصیصه B باید در دو رابطه متفاوت باشد در حالی که متفاوت بودن یکی از آن‌ها برای آویزان بودن تاپل کافی است. در واقع تاپل‌هایی از دو رابطه توسط عملگر ضد نیم اتصال به عنوان خروجی داده می‌شوند که مقدار خصیصه A یا B آن‌ها در هر دو رابطه R و S به صورت هم‌زمان برابر نباشند. به عبارت دیگر تاپل‌هایی باید در خروجی نهایی ظاهر شوند که یا مقدار خصیصه A آن‌ها در دو رابطه متفاوت است یا مقدار خصیصه B آن‌ها در دو رابطه متفاوت است. گزینه‌های دوم و چهارم این شرط را نقض می‌کنند، بنابراین پاسخ سؤال نیستند. دو جدول R و S با مقادیر زیر را در نظر بگیرید:

	A	B	C
→ پیوندپذیر	①	②	3
→ پیوندناپذیر (تاپل آویزان)	⑤	6	8
→ پیوندناپذیر (تاپل آویزان)	2	④	5
→ پیوندناپذیر (تاپل آویزان)	3	7	2

جدول R

	A	B	D
	①	②	4
	⑤	8	9
	7	④	1
	4	5	6

جدول S

سطرهای پیوندپذیر جدول R به صورت پرس و جوی زیر استخراج می‌شود:

```
select R.*
from R,S
```

```
where R.A=S.A AND R.B=S.B
```

با توجه به جداول فوق، خروجی نهایی پرس و جوی فوق پس از انجام عملگر ضرب دکارتی و شرط اتصال برابری در دو ستون A و B یعنی سطرهای پیوندپذیر به صورت زیر است:

A	B	C
1	2	3

مطابق پرس و جوی مطرح شده در گزینه اول، داریم:

```
select R.*
from R,S
```

```
where R.A<>S.A AND R.B<>S.B
```

با توجه به جداول فوق، خروجی نهایی پرس و جوی فوق پس از انجام عملگر ضرب دکارتی و شرط اتصال نابرابری در هر دو ستون A و B ، به صورت زیر است:

A	B	C
3	7	2

بنابراین پرس و جوی گزینه اول همه تاپل‌های آویزان (پیوندناپذیر) مربوط به جدول R را استخراج نمی‌کند.

مطابق پرس و جوی مطرح شده در گزینه دوم، داریم:

```
select *
from R
where not exists (select *
                  from S
                  where S.A=R.A OR S.B=R.B)
```

با توجه به جداول فوق، خروجی نهایی پرس و جوی فوق به صورت زیر است:

A	B	C
3	7	2

در واقع در پرس و جوی فوق، فقط به ازای تاپل (3,7,2) از جدول R، عبارت جلوی not exists می‌گردد که منجر به TRUE شدن where در پرس و جوی خارجی و به تبع انتخاب تاپل از جدول R می‌گردد.

بنابراین پرس و جوی گزینه دوم نیز همه تاپل‌های آویزان (پیوندناپذیر) مربوط به جدول R را استخراج نمی‌کند.

مطابق پرس و جوی مطرح شده در گزینه سوم، داریم:

```
select *
from R
where not exists (select *
                  from S
                  where S.A=R.A AND S.B=R.B)
```

با توجه به جداول فوق، خروجی نهایی پرس و جوی فوق به صورت زیر است:

A	B	C
5	6	8
2	4	5
3	7	2

در واقع در پرس و جوی فوق، فقط به ازای تاپل‌های (5,6,8)، (2,4,5) و (3,7,2) از جدول R، عبارت جلوی not exists می‌گردد که منجر به TRUE شدن where در پرس و جوی خارجی و به تبع انتخاب تاپل از جدول R می‌گردد.

بنابراین پرس و جوی گزینه سوم همه تاپل‌های آویزان (پیوندناپذیر) مربوط به جدول R را استخراج می‌کند، که پاسخ سؤال نیز همین گزینه سوم است.

مطابق پرس و جوی مطرح شده در گزینه چهارم، داریم:

```
select *
from R
where A <> any (select A AND B <> any (select B
                                     from S)
               from S)
```

با توجه به جداول فوق، خروجی نهایی پرس و جوی فوق به صورت زیر است:

A	B	C
3	7	2

بنابراین پرس و جوی گزینه چهارم نیز همه تاپل‌های آویزان (پیوندناپذیر) مربوط به جدول R را استخراج نمی‌کند. اگر در گزینه چهارم به جای AND از OR استفاده می‌شد، آنگاه گزینه چهارم نیز می‌توانست پاسخ سؤال باشد.

۸- گزینه (۱) صحیح است.

ساختاری نحوی دستور select به صورت زیر است:
 SELECT ستون‌ها
 FROM جداول
 [WHERE شرایط]
 [ORDER BY شرایط]
 [GROUP BY ستون‌ها]
 [HAVING شرایط]

در گزینه اول به دلیل استفاده از دستور DESC خروجی به صورت نزولی نشان داده خواهد شد، اما در سایر گزینه‌ها خطای نحوی وجود دارد.

۹- گزینه (۴) صحیح است.

ابتدا با دستور Group by J#، اطلاعات جدول براساس J# دسته‌بندی می‌شود، سپس با استفاده از شرط Having و تابع count و دستور Distinct عمل شمارش S# در داخل هر پروژه انجام می‌شود، اگر این مقدار برابر یک بود، به این معنی است که یک تهیه‌کننده در آن شماره پروژه، به تنهایی تمام قطعات مورد نیاز پروژه را تهیه کرده است.
 به مثال زیر توجه کنید:

S#	P#	J#	
S1	P1	J1	select J#
S1	P2	J1	from SPJ
S1	P3	J1	Group by J#
S1	P1	J2	Having count (Distinct S#) = 1
S2	P2	J2	

از آن‌جا که هدف، شمارش تهیه‌کننده‌ها به تفکیک پروژه‌ها است، باید گروه‌بندی براساس شماره پروژه انجام شود، از طرفی استفاده از دستور Distinct باعث می‌شود که یک تهیه‌کننده فقط یک بار با تابع count شمرده شود.

پرس‌وجوی فوق گروه‌های زیر را ایجاد می‌کند:

J1		J2		
S1	P1	S1	P1	در Group by سرگروه در خروجی نمایش داده می‌شود.
S1	P2	S2	P2	
S1	P3			
				J#
				J1
				J2

در گروه J1 شرط Count (Distinct S#) = 1 برقرار است. پس این گروه توسط Having انتخاب می‌شود.

J#
J1

همان‌طور که واضح است، فقط در پروژه J1 است که همه قطعات آن را یک تهیه‌کننده به نام S1 تهیه می‌کند، اما در پروژه J2، دو تهیه‌کننده S1 و S2 به ترتیب قطعات P1 و P2 را تهیه می‌کنند. که

count(Distinct S#) در آن برابر مقدار 2 می‌باشد.

۱۰- گزینه () صحیح است.

پرس‌وجوی داده شده، به دلیل استفاده از تابع عددی در محل نادرست، خطای نحوی دارد و توسط کامپایلر قابل اجرا نیست. متأسفانه این پاسخ در بین گزینه‌های داده شده وجود ندارد.

۱۱- گزینه (۲) صحیح است.

طبق قانون اول جامعیت (جامعیت درون رابطه‌ای) هر رابطه (جدول) باید به تنهایی درست باشد، یعنی صفات آن چند مقداری و مرکب نباشند و حداقل یک کلید کاندید داشته باشد. بنابراین گزینه سوم درست است.

مطابق قانون دوم جامعیت (جامعیت موجودیت) مقادیر همه مؤلفه‌های تاپل‌ها در یک رابطه باید در دامنه تعریف شده برای آن مؤلفه قرار داشته باشند و همچنین کلید اصلی یا بخش‌های مختلف آن نباید مقدار NULL داشته باشند. بنابراین گزینه اول درست است.

قانون سوم جامعیت (جامعیت ارجاع) به مسئله کلید خارجی مربوط می‌شود. بنابراین گزینه چهارم نیز به درستی چگونگی برقراری صحت ارجاع در SQL را بیان می‌کند. اما مسئله not NULL بودن، اول این که به قانون دوم جامعیت مربوط می‌شود و دوم این که وقتی یک صفت به عنوان کلید اصلی تعریف می‌شود به صورت پیش تعریف از دو خاصیت not NULL و شاخص unique برخوردار است و لازم نیست این دو خاصیت به صورت صریح برای آن ذکر شود.

۱۲- گزینه (۳) صحیح است.

گزینه اول نادرست است: زیرا تابع عددی avg در محل نادرست استفاده شده است، بنابراین این پرس‌وجو خطای نحوی دارد و از سوی کامپایلر اجرا نمی‌گردد. تابع عددی جلوی where حتماً باید داخل select باشد.

گزینه دوم نادرست است: زیرا بدون Group by از having استفاده شده است، بنابراین این پرس‌وجو خطای نحوی دارد و از سوی کامپایلر اجرا نمی‌گردد.

گزینه چهارم نادرست است: زیرا شرط سازگاری صفات در جلوی Group by و select رعایت نشده است، بنابراین این پرس‌وجو خطای نحوی دارد و از سوی کامپایلر اجرا نمی‌گردد. گزینه سوم درست است.

جدول زیر را در نظر بگیرید:

empno	empname	dep	salary
e1	en1	amoozesh	10
e2	en2	amoozesh	20
e3	en3	amoozesh	30
e4	en4	maly	40
e5	en5	maly	50
e6	en6	maly	60
e7	en7	honari	70
e8	en8	honari	80

جدول employee

پرس وجو: مشخصات کارکنانی که حقوقشان از متوسط حقوق کلیه کارکنان سازمان بیش تر است را لیست نمایید. (مطابق گزینه سوم)

```
select *
from employee
where salary > (select avg(salary) from employee)
```

45

پس مقدار میانگین حقوق کلیه کارمندان، برابر 45 می باشد. بنابراین خروجی پرس وجوی فوق به صورت زیر خواهد بود:

empno	empname	dep	salary
e5	en5	maly	50
e6	en6	maly	60
e7	en7	honari	70
e8	en8	honari	80

در واقع در گزینه سوم ابتدا در پراتنز داخلی میانگین حقوق کلیه کارمندان سازمان محاسبه می شود. سپس در پرس وجوی بیرونی مشخصات کامل کارکنانی که حقوقشان از متوسط حقوق کلیه کارمندان سازمان بیش تر است به عنوان خروجی نهایی تولید می شود.

۱۳- گزینه (۱) صحیح است.

بر اساس پرس وجوی مطرح شده در صورت سؤال، تمامی قطعات قرمز رنگ، به رنگ نارنجی توسط دستور update به روزرسانی می گردند.

۱۴- گزینه (۴) صحیح است.

پرس وجوی مطرح شده، نام و شهر تولیدکنندگانی را استخراج می کند که تاکنون قطعه P1 را تولید کرده اند. در گزینه های اول، دوم و سوم نحوه بیان این پرس وجو به شیوه های متفاوت، اما با خروجی یکسان بیان شده است.

۱۵- گزینه (۲) صحیح است.

A	B	C	F	B	A	B	C	F
a1	b1	MK	f1	b1	a1	b1	MK	f1
a2	b4	MTK	f2	b2	a2	b1	RP	f2
a1	b2	LN			a1	b2	LN	f2
a2	b1	RP						

Right Outer Join

جدول x

جدول y

توجه کنید که پس از پیوند طبیعی، تاپلی از جدول x (جدول سمت راست) باقی نمی ماند که بخواهد اضافه شود (تاپل آویزان در جدول سمت راست وجود ندارد). بنابراین با توجه به جداول مذکور، حاصل الحاق خارجی راست، برابر الحاق طبیعی است.

پس از اعمال شرط زیر

A	B	C	F
a1	b1	MK	f1

where y.c like 'M%'

۱۶- گزینه (۳) صحیح است.

پرس و جوی مطرح شده در صورت سؤال خطای نحوی دارد. در زیر پرس و جوهای متصل به خارج، باید نوع اتصال، مابین زیر پرس و جوی داخلی و پرس و جوی خارجی مشخص گردد. شکل اصلاح شده پرس و جوی مطرح شده در صورت سؤال به صورت زیر است:

```
select Distinct sname
from S
where not exists (select *
from G
where G.S# = S.S# AND L#='L1')
```

جدول زیر را در نظر بگیرید:

S#	Sname
S1	S _{n1}
S2	S _{n2}
S3	S _{n3}
S4	S _{n5}
S5	S _{n5}
S6	S _{n6}

S#	L#	grade
S1	L1	10
S1	L2	12
S2	L1	14
S3	L2	16
S4	L2	18
S5	L2	20

L#	Lname
L1	L _{n1}
L2	L _{n2}

جدول S

جدول G

جدول L

در پرس و جوی اصلاح شده صورت سؤال، به ازای حرکت در هر سطر از جدول S، سطری از جدول G که شماره دانشجویی آنها با شماره دانشجویی جدول S برابر باشد، بررسی می‌گردد که به ازای شماره دانشجویی مورد نظر در جدول S هیچ نمره‌ای از درس L1 به ازای شماره دانشجویی مورد نظر در جدول G وجود نداشته باشد. اگر هیچ سطری به ازای شماره دانشجویی موجود در جدول S، در جدول G با توجه به شرط مطرح شده موجود نباشد، سطر مورد نظر از جدول S انتخاب می‌گردد. و در نهایت نام دانشجویانی در خروجی قرار می‌گیرد که از درس L1 نمره‌ای ندارند.

در پرس و جوی فوق، به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول G بررسی می‌گردد، تا مشخص گردد هیچ سطری در جدول G قرار ندارد که شرط موجود در select داخلی را برقرار کند، اگر هیچ سطری وجود نداشته باشد، سطر مورد نظر از جدول S، در خروجی نمایش داده می‌شود.

پرس و جوی فوق در واقع نام دانشجویانی را نمایش می‌دهد که برای آنها هیچ نمره‌ای از درس L1 در جدول G ثبت نشده است. البته چون به دلیل کلید نبودن Sname، ممکن است اسامی برخی دانشجویان شبیه هم باشد، امکان دارد، اسامی تکراری در خروجی این پرس و جو مشاهده شود. بنابراین خروجی Sname با توجه به استفاده از دستور Distinct در پرس و جوی مطرح شده، به صورت زیر خواهد بود:

Sname
S _{n3}
S _{n5}

توجه: SQL به طور پیش فرض سطرهای تکراری را حذف نمی‌کند، مگر از دستور Distinct استفاده شود. گزینه دوم نادرست است. زیرا، در SQL مفاهیمی همچون اولین، آخرین و یکی از، معنا و مفهومی ندارد. در واقع اگر شرط انتخاب سطر، برای سطرهای برقرار باشد، سطر مورد نظر در خروجی قرار می‌گیرد. **توجه:** اگر به جای دستور not exists از دستور exists در پرس و جوی اصلاح شده صورت سؤال، استفاده گردد، آنگاه پرس و جوی مطرح شده در گزینه اول ایجاد می‌گردد.

۱۷- گزینه (۱) صحیح است.

پراتنز داخلی یعنی (select MAX (grade) from G) ماکزیمم نمره ثبت شده را استخراج می‌کند. پراتنز بیرونی تر شماره تمام دانشجویانی را می‌دهد که نمره ماکزیمم کسب کرده‌اند. گزینه سوم بیانگر شماره دانشجویانی است که تعداد نمره بیشتری دریافت کرده‌اند که با پرس و جوی مطرح شده مطابقت ندارد. دقت کنید که گزینه دوم با دستورات SQL قابل تولید نیست. زیرا SQL با مفاهیمی همچون اولین، یکی از، بعضی و آخرین، بیگانه است. اگر شرط انتخاب سطر در دستور select برقرار باشد، همه سطرهای حائز شرط، در خروجی ظاهر خواهند شد.

۱۸- گزینه (۳) صحیح است.

گزینه اول نادرست است. زیرا، پس از استفاده از دستور Group By نمی‌توان از Subquery استفاده کرد. اصولاً Group By در پرس و جوی تو در تو کاربرد ندارد. گزینه دوم نادرست است. زیرا، در دستور Group By شرط با Having بیان می‌شود و نمی‌توان از where استفاده نمود چون خطای نحوی است. گزینه چهارم نادرست است. زیرا، Count(*) در جلوی دستور select باید تنها باشد. همچنین Count(*) نمی‌تواند جایگزین Count(state) شود زیرا Count(*) مواردی را که مقدار آن‌ها NULL است را نیز مورد شمارش قرار می‌دهد در حالی که Count (state) مواردی را که مقدار آن‌ها NULL است، نمی‌شمارد. گزینه سوم درست است. زیرا، امکان ترکیب شرط $Count(state) < 10$ با شرط فعلی بدون هیچ محدودیتی امکان‌پذیر است.

۱۹- گزینه (۳) صحیح است.

با اجرای دستور مطرح شده در صورت سؤال، و با توجه به وجود جمله S(S#, Sname, city) در دستور، این دستور بدون خطا اجرا می‌شود و مقدار NULL در ستون status قرار می‌گیرد. البته به شرطی که ستون status در زمان تعریف ساختار جدول S، به صورت not NULL تعریف نشده باشد. توجه کنید که اگر ساختار دستور به صورت مقابل می‌بود:
Insert into S values ('S10', Smith 'New York')
دستور فوق به دلیل نقض شروط سازگاری از نظر تعداد و نوع با خطا مواجه می‌گردید. و اجرای دستور از سوی DBMS لغو می‌گردید، زیرا جدول تولیدکننده دارای ستون چهارمی به نام status است که در دستور، مقداری برای آن در نظر نگرفته شده است.

۲۰- گزینه (۳) صحیح است.

دستور SQL داده شده جداول S، P و SP را به هم پیوند می‌زند. یک قطعه در جدول P ممکن است توسط هیچ، یک، دو یا چند تولید کننده ساخته شده باشد. با توجه به خط where s.city='London' اگر قطعه‌ای

حداقل توسط یک تولید کننده لندن ساخته شده باشد، کلیه مشخصات آن نشان داده می‌شود. برای ایجاد گزینه دوم باید تمامی قطعات تولید شده توسط هر یک از تولید کنندگان لندن را استخراج نمود، سپس اشتراک گرفت، حاصل، کلیه مشخصات قطعاتی خواهد بود که توسط تمام تولید کنندگان لندن تولید می‌شود.

گزینه چهارم می‌گوید، کلیه مشخصات قطعاتی که دقیقاً توسط یکی از تولید کنندگان لندن تولید می‌شود، فرض کنید S1، S2 و S3 تهیه کننده لندن هستند، اما قطعه P2 فقط توسط S2 تولید شده است، بنابراین P2 در خروجی می‌آید. اما قطعه P1 در خروجی نمی‌آید، زیرا هم S1 آن را تولید کرده است و هم S2.

S#	Sname	City	S#	P#	QTY	P#	Pname	Color
S1	Sn1	London	S1	P1	10	P1	Pn1	Red
S2	Sn2	London	S2	P1	20	P2	Pn2	Blue
S3	Sn3	London	S2	P2	30	P3	Pn3	Red
S4	Sn4	London						

جدول s

جدول sp

جدول p

با توجه به جدول S، تهیه کننده S3، لندن هست اما با توجه به جدول SP، قطعه‌ای تولید نکرده است، بنابراین تهیه کننده S3، تولید کننده هیچ قطعه‌ای نیست. خروجی گزینه سوم به صورت زیر است:

P#	Pname	Color
P1	Pn1	Red
P2	Pn2	Blue

خروجی گزینه دوم به صورت زیر است:

P#	Pname	Color
P1	Pn1	Red

خروجی گزینه چهارم به صورت زیر است:

P#	Pname	Color
P2	Pn2	Blue

پرس و جوی گزینه سوم را مجدداً در نظر بگیرید:

«کلیه مشخصات قطعاتی که حداقل توسط یکی از تولید کنندگان لندن تولید می‌شود.»

حال سراغ جدول p بروید، سطر اول را انتخاب کنید، که p1 مشاهده می‌شود، حال این p1 را ببرید داخل جدول sp، ببینید S# آن کیست، یعنی چه کسی p1 را در جدول sp تولید کرده است، می‌بینید که s1 و s2 تولید کرده‌اند، حال s1 یا s2 را ببرید داخل جدول s ببینید که این s1 یا s2 تهیه کننده لندن هستند، می‌بینید که هستند، پس سطر اول از جدول p یعنی (p1, pn1, Red) در خروجی نمایش داده می‌شود، زیرا نه تنها که حداقل توسط یکی از تولید کنندگان لندن تولید شده است که بلکه توسط دو تولید کننده لندن یعنی s1 و s2 تولید شده است، سطر دوم از جدول p هم، یعنی (p2, pn2, Blue) در خروجی نمایش داده می‌شود، مانند روال سطر قبل، خروجی را دنبال کنید، اما سطر سوم یعنی (p3, pn3, Red) در خروجی نمایش داده نمی‌شود، چون p3 را که ببریم داخل جدول p، اصلاً p3 ای وجود ندارد که روال را دنبال کنیم. در یک قاعده کلی، یکی از راه‌حل‌های مقایسه سطرهای یک جدول با خودش یا دو جدول با هم یا چند جدول با هم، استفاده از ضرب دکارتی و بیان نوع اتصال و شرط ضرب می‌باشد. به این مطلب کمی فکر کنید، ساده

است، برای مقایسه یک سطر از یک جدول مبدأ با یک سطر از یک جدول مقصد، دو جدول ابتدا باید ضرب دکارتی شوند. تا همه‌ی سطرها کنار هم قرار گیرند، سپس شرط ضرب اعمال گردد. مثال واقعی مانند زمانی است که استادی (جدول مبدأ) سؤال می‌پرسد و دانشجویی (جدول مقصد) پاسخ را می‌دهد که اگر حداقل یک دانشجو پاسخ دهد، کافی است. در واقع سؤال در جدول مبدأ قرار دارد و پاسخ در جدول مقصد.

مثال: نام تولید کنندگانی که حداقل یک قطعه تولید می‌کنند.

کدام تولید کننده در جدول مبدأ قرار دارد (سؤال) و قطعه در جدول مقصد قرار دارد (پاسخ)

حل: راهکار در استفاده از ضرب دکارتی است، تا سطرهاى دو جدول کنار هم قرار بگیرند. شرط ضرب هم $sp.s\# = s.s\#$ است، چون وقتی که $s1$ از جدول مبدأ به عنوان سؤال مطرح می‌شود، در جدول مقصد فقط برای $s1$ باید دنبال پاسخ بود! $s1$ را از جدول s (مبدأ) انتخاب کنید، بربید داخل جدول sp (مقصد) ببینید حداقل یک قطعه را تولید می‌کند، بله $p1$ را تولید می‌کند، حال $s2$ را از جدول s انتخاب کنید، بربید داخل جدول sp ، ببینید حداقل یک قطعه را تولید می‌کند، بله، $p1$ و $p2$ را تولید می‌کند، حال $s3$ و $s4$ را از جدول s به ترتیب، بربید داخل جدول sp نه، قطعه‌ای تولید نمی‌کنند، این حرف‌ها که گفتیم به زبان SQL به صورت زیر نوشته می‌شود:

```
select  sname      sname
from    s , sp    →  sn1
where  s.s#=sp.s#  sn2
```

۲۱- گزینه (۲) صحیح است.

در دو جدول loan و borrower ستون loan-number به عنوان ستون مشترک تلقی می‌شود. چنان‌که ملاحظه می‌شود، کلیه سطرهاى پیوندپذیر حاصل از مقدار L-170 و K-230 در ستون مشترک و هم‌چنین سطرهاى پیوندناپذیر چپ حاصل از مقدار L-260 و سطرهاى پیوندناپذیر راست حاصل از مقدار L-155 در هر دو جدول در خروجی نمایش داده شده است، بنابراین، خروجی داده شده با خروجی الحاق خارجی کامل تطبیق دارد.

۲۲- گزینه (۲) صحیح است.

عبارت جبر رابطه‌ای داده شده نام دانشجویان دانشکده کامپیوتر را به عنوان خروجی می‌دهد. گزینه اول نادرست است، زیرا نوع اتصال بیان نشده است. درست آن براساس جبر رابطه‌ای مذکور به صورت زیر است:

```
select sname
from stud , clg
where stud.clg#=clg.clg# AND clg.clgname='computer'
```

گزینه دوم درست است و معادل پرس‌وجوی اصلاح شده فوق و جبر رابطه‌ای مطرح شده است.

گزینه سوم نادرست است، زیرا در جلوی دستور where مربوط به جدول stud از ستون clgname برای شرط استفاده شده است، که اصلاً جدول stud، ستون clgname ندارد، ستون clgname مربوط به جدول clg است.

گزینه چهارم نادرست است، زیرا خطای نحوی دارد، ستون جلوی group by با ستون جلوی select باید یکسان باشد، مگر ستون جلوی select با توابع آماری همراه گردد.

۲۳- گزینه (۱) صحیح است.

گزینه‌های دوم تا چهارم نام کارمندانی را می‌دهند که در واحد فروش کار می‌کنند. در گزینه‌های دوم و سوم دو دستور any و in مانند یک‌دیگر عمل می‌کنند. هم‌چنین در گزینه چهارم با استفاده از ضرب دکارتی و شرط ضرب دقیقاً همان خروجی تولید می‌شود. اما گزینه اول درست معکوس خروجی سه گزینه دیگر را می‌دهد، نام کارمندانی را می‌دهد که در واحد فروش کار نمی‌کنند. اگر در گزینه اول به جای not exists از exists استفاده می‌شد، آن‌گاه گزینه اول هم معادل سایر گزینه‌ها می‌شد.

۲۴- گزینه (۲) صحیح است.

از آن‌جا که رکورد دانشجویان قرار است، در آخر ترم تغییر کند و نمره دانشجویان وارد شود، دستور update باید اجرا شود. دقت کنید که در فرض مطرح شده در صورت سؤال، وارد کردن نمرات دانشجویان، درج سطر جدید در جدول enroll تلقی نمی‌شود، بلکه وارد کردن نمرات به منزله بروز رسانی ستون score در سطرهای مربوط به درسی است که می‌خواهیم نمرات آن را ثبت کنیم. بر این اساس از دستور update برای این کار استفاده می‌شود. دستور Insert یک رکورد کامل جدید به جدول اضافه می‌کند، در حالی که ما می‌خواهیم ستون score را اصلاح کنیم، لذا از دستور update استفاده می‌کنیم.

۲۵- گزینه () صحیح است.

جداول زیر را داریم:

	کلید خارجی	کلید اصلی																								
	↑	↑																								
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <th style="width: 25%;"><u>A</u></th> <th style="width: 25%;">B</th> <th style="width: 25%;">C</th> <th style="width: 25%;">FK</th> </tr> <tr> <td>1</td> <td>-</td> <td>-</td> <td>10</td> </tr> </table>	<u>A</u>	B	C	FK	1	-	-	10	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <th style="width: 25%;"><u>D</u></th> <th style="width: 25%;"><u>E</u></th> <th style="width: 25%;">W</th> <th style="width: 25%;">FK</th> </tr> <tr> <td>1</td> <td>2</td> <td>-</td> <td>10</td> </tr> </table>	<u>D</u>	<u>E</u>	W	FK	1	2	-	10	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <th style="width: 25%;"><u>FK</u></th> <th style="width: 25%;">X</th> <th style="width: 25%;">Y</th> <th style="width: 25%;">Z</th> </tr> <tr> <td>10</td> <td>-</td> <td>-</td> <td>-</td> </tr> </table>	<u>FK</u>	X	Y	Z	10	-	-	-
<u>A</u>	B	C	FK																							
1	-	-	10																							
<u>D</u>	<u>E</u>	W	FK																							
1	2	-	10																							
<u>FK</u>	X	Y	Z																							
10	-	-	-																							
جدول T1	جدول T2	جدول T3																								

```
create table T1
(A char (5)
B char (5)
C char (5)
FK char (5)
primary key (A)
foreign key (FK) References T3 (FK)
```

```
on update option {
no action (restrict)
cascade ✓
set null
}
```

از آن‌جا که مطابق فرض مسأله، گزینه on update cascade برای جدول T2 ذکر نشده است، بنابراین، برای جدول T2 گزینه no action (به معنی صرف‌نظر از عملیات در صورتی که بر اثر عملیات حذف، درج و بروزسانی، قوانین جامعیت نقض گردد) در نظر گرفته می‌شود. در واقع no action گزینه پیش فرض است و هیچ عملی انجام نمی‌شود.

از آن‌جا که ستون FK در جدول T3 کلید اصلی و مورد مراجعه سطرهای دو جدول T1 و T2 است، امکان

تغییر دادن بی قید و شرط مقدار FK در جدول T3 وجود ندارد.

فرض اول: اگر مقدار 10 در ستون FK در جدول T3 باشد و مقدار 10 در ستون FK در دو جدول T2 و T1 نباشد. در این صورت مقدار 10 در ستون FK به مقدار 15 به روز رسانی می شود. در واقع فقط جدول T3 به روز رسانی می گردد.

فرض دوم: اگر مقدار 10 در ستون FK در جدول T3 باشد و مقدار 10 در ستون FK در جدول T1 هم باشد و مقدار 10 در ستون FK در جدول T2 نباشد. در این صورت مقدار 10 در ستون FK در دو جدول T1 و T3 به روز رسانی می شود در واقع فقط جدول T3 و T1 به روز رسانی می گردد.

فرض سوم: اگر مقدار 10 در ستون FK در جدول T3 باشد و مقدار 10 در ستون FK در دو جدول T2 و T1 هم باشد. در این صورت از آن جا که تغییر مقدار 10 در ستون FK در جدول T3 به مقدار 15 سبب ارجاع NULL در جدول T2 می گردد، عملیات به روزرسانی برای هر سه جدول لغو می گردد، هر چند جدول T1، دارای on update cascade می باشد. اما از آن جا که برای جدول T2 گزینه on update no action فعال است، اگر تغییر در جدول T3 سبب ارجاع NULL در جدول T2 گردد، عملیات به روزرسانی در جدول T3 لغو می گردد. در واقع مقدار FK در هیچ یک از جداول به روز نمی شود.

۲۶- گزینه (۱) صحیح است.

عبارت جبری رابطه ای داده شده نام اساتیدی را به عنوان خروجی می دهد که در دپارتمان کامپیوتر مشغول کار هستند.

گزینه اول درست است، زیرا در پراتز داخلی شماره دپارتمان کامپیوتر شناسایی می شود و در پراتز بیرونی تر نام اساتیدی که شماره دپارتمان آن ها همان شماره دپارتمان کامپیوتر است به عنوان خروجی نهایی تولید می شود، به عبارت دیگر گزینه اول همانند عبارت جبر رابطه ای داده شده نام اساتیدی را به عنوان خروجی می دهد که در دپارتمان کامپیوتر مشغول کار هستند.

گزینه دوم نادرست است، زیرا نوع اتصال ذکر نشده است و بنابراین خروجی خواسته شده را نمی دهد.

شکل صحیح گزینه دوم مطابق با جبر رابطه ای ذکر شده به صورت زیر است:

```
select pname
from prof, dep
where prof.depnum=dep.depnum and depname='computer'
```

گزینه سوم نادرست است: زیرا شرط ضرب آمده است اما عملیات ضرب انجام نشده است، زیرا نام جدول dep در مقابل from نیست، پس این گزینه هم خروجی خواسته شده را نمی دهد.

شکل صحیح گزینه سوم مطابق با جبر رابطه ای ذکر شده به صورت زیر است:

```
select pname
from prof, dep
where prof.depnum=dep.depnum
and depname in (select depname from dep
where depname='computer')
```

گزینه چهارم نادرست است، زیرا چون از group by استفاده شده است باید ستون مقابل select همان ستونی باشد که براساس آن گروه بندی انجام شده، اما در این گزینه گروه بندی براساس depnum صورت گرفته است، در حالی که در مقابل select نام ستون pname آمده است. به همین دلیل این گزینه هم پاسخ

مورد نظر را به دست نمی‌دهد، در واقع خطای نحوی دارد.

۲۷- گزینه (۳) صحیح است.

در یک قاعده، شرط Exists هنگامی درست می‌شود که R تهی نباشد. R هنگامی تهی نیست که حداقل یک سطر داشته باشد.

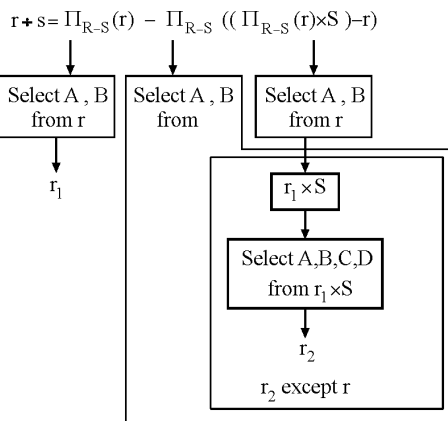
۲۸- گزینه (۲) صحیح است.

در یک قاعده کلی عملگر تقسیم به صورت $r \div s = \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times S) - r)$ پیاده‌سازی می‌شود. توجه کنید که شرط زیر مجموعه بودن مقسوم‌علیه به مقسوم هم برقرار است، با توجه به ستون‌های جداول r و s داریم:

$$R = \{A, B, C, D\} \quad S = \{C, D\}$$

$$R - S = \{A, B\}$$

بنابراین، جمله (select A, B from r Except) باید در ابتدای پرس و جو باشد، پس گزینه‌های اول و سوم توانایی پیاده‌سازی عملگر تقسیم را ندارند، لذا این گزینه‌ها را کنار می‌گذاریم. در گزینه چهارم از عملگر اشتراک استفاده شده است، در حالی که عملگر intersect در تعریف عملگر تقسیم کاربرد ندارد. بنابراین گزینه چهارم را هم کنار می‌گذاریم. اما گزینه دوم پیاده‌سازی درستی از عملگر تقسیم را ارائه می‌کند. اطلاعات زیر گویای این مطلب است:



۲۹- گزینه (۴) صحیح است.

اگر فرض کنیم جداول crs و sec و clg به صورت زیر باشد:

crs(c#, cname, unit, clg#)

(شماره دانشکده ارائه دهنده، تعداد واحد، نام درس، شماره درس) درس

sec(sec#, c#, s#, term, pname, score)

(نمره، نام استاد، ترم، شماره دانشجویی، شماره درس، شماره گروه) گروه درس

clg(clg#, clgname, city, pname)

(نام رئیس، نام شهر، نام دانشکده، شماره دانشکده) دانشکده

توجه: از جدول crs برای پیوند جدول‌ها استفاده می‌شود.

توجه: اطلاعات ترم، درس و نام دانشکده (جستجو براساس دانشکده کامپیوتر) در سه جدول sec، crs و clg قرار دارند که تنها گزینه‌های دوم و چهارم از اتصال این سه جدول استفاده کرده‌اند. جداول زیر را در نظر بگیرید:

clg#	clgname	city	pname	c#	cname	unit	clg#	sec#	c#	s#	term	pname	score
clg1	کامپیوتر	تهران	احمدی	c1	ساختمان داده	3	clg1	sec1	c1	s1	881	علوی	20
clg2	برق	شیراز	اکبری	c2	مدار منطقی	3	clg1	sec2	c2	s2	881	اصغری	18
				c3	مدار الکتریکی	3	clg2	sec3	c2	s3	881	جعفری	17
								sec4	c2	s4	882	حسینی	19
								sec5	c3	s5	883	سجادی	15

جدول دانشکده (clg)

جدول درس (crs)

جدول گروه درس (sec)

پرس‌وجوی مطرح شده در سؤال را در نظر بگیرید:

«ترم‌هایی که دانشکده کامپیوتر درس ارائه کرده است.»

حال سراغ جدول clg بروید، سطری را انتخاب کنید که نام دانشکده آن برابر کامپیوتر باشد ("کامپیوتر"=clgname)، که با توجه به شرط مطرح شده، سطر اول انتخاب می‌گردد که clg# آن برابر clg1 می‌باشد. حال این clg1 را ببرید داخل جدول crs، ببینید چه درس‌هایی در دانشکده clg1 ارائه شده است. می‌بینید که شماره درس‌های c1 و c2 ارائه شده است. حال این شماره درس‌های c1 و c2 را ببرید داخل جدول sec، ببینید درس‌های c1 و c2 در چه ترم‌هایی ارائه شده است. می‌بینید که در ترم‌های 881 و 882 ارائه شده است. بنابراین 881 و 882 در خروجی نمایش داده می‌شود. در یک قاعده کلی، یکی از راه‌حل‌های مقایسه سطرهای یک جدول با خودش یا دو جدول با هم یا چند جدول با هم، استفاده از ضرب دکارتی و بیان نوع اتصال و شرط ضرب می‌باشد. این حرف‌ها که گفتیم به زبان SQL به صورت زیر نوشته می‌شود:

```
select distinct term
from sec, clg, crs
where clg.clg # = crs.clg # and crs.c # = sec.c # and clgname = 'کامپیوتر'
```

گزینه اول نادرست است، زیرا اول این که از جدول crs استفاده نشده است، و دوم این‌که، پرس‌وجوی مطرح شده در گزینه اول، شماره ترم‌هایی که اساتیدی هم نام با رئیس دانشکده کامپیوتر دارند را ارائه می‌دهد.

گزینه دوم نادرست است، زیرا شرط مطرح شده در بخش having نمی‌تواند پاسخ مورد نظر را تولید کند، فرم اصلاح شده‌ی گزینه دوم برای برآورده کردن پرس‌وجوی مطرح شده در سؤال به صورت زیر است:

```
select term
from sec, clg, crs
where clg.clg # = crs.clg # and crs.c # = sec.c # and clgname = 'کامپیوتر'
```

group by term

توجه: به خاطر بسپارید که یکی از راه‌کارهای شبیه‌سازی دستور distinct، استفاده از دستور group by می‌باشد.

گزینه سوم نادرست است، زیرا having باید با group by همراه باشد، بنابراین خطای نحوی دارد و از سوی

کامپایلر اجرا نمی‌گردد.

۳۰- گزینه (۲) صحیح است.

در دستور داده شده تابع عددی AVG در محل نادرست مورد استفاده قرار گرفته است، توابع عددی در جلوی where باید داخل select قرار بگیرند. بنابراین پرس وجو دارای خطای نحوی است و از سوی کامپایلر اجرا نمی‌گردد.
جدول زیر را در نظر بگیرید:

std #	sname	avg
s1	sn1	12
s2	sn2	14
s3	sn3	16
s4	sn4	18

جدول stud

اگر پرس وجوی مطرح شده به شکل زیر بیان شود:

```
select *
from stud
where avg > (select avg(avg) from stud)
```

آنگاه پرس وجوی فوق، مشخصات دانشجویانی را از جدول stud استخراج می‌کند که معدل آن‌ها، از میانگین معدل تمامی دانشجویان بالاتر باشد. بنابراین خروجی پرس وجوی فوق به صورت زیر است:

std #	sname	avg
s3	sn3	16
s4	sn4	18

۳۱- گزینه (۱) صحیح است.

جدول work-on در صورت سؤال داده نشده است. اگر جدول زیر را فرض کنیم:

کارفرما	شماره پروژه	کارمند پروژه (مجری پروژه)
Essn	pno	name
Alavi	123	Akbari
Alavi	123	Parvizi
Hassani	456	Sajadi

هر سطر جدول فوق بیان می‌کند که کدام پروژه توسط کدام کارمند اجرا شده و کارفرمای آن پروژه چه کسی بوده است، می‌توان گفت که پراتنز داخلی:

(select pno from work-on where name='parvizi') شماره کلیه پروژه‌هایی را می‌دهد که مجری آن‌ها پرویزی بوده و پراتنز بیرونی اسامی همه کارفرمایانی را می‌دهد که پروژه‌های آن‌ها توسط پرویزی اجرا شده است. بنابراین خروجی پرس وجوی مطرح شده به صورت زیر خواهد بود.

```
essn
Alavi
```

۳۲- گزینه (۳) صحیح است.

جدول زیر را در نظر بگیرید:

s #	sname	dept #	average
s1	sn1	dep1	9
s2	sn2	dep1	10
s3	sn3	dep2	11
s4	sn4	dep2	12
s5	sn5	dep2	13
s6	sn6	dep3	14
s7	sn7	dep3	15
s8	sn8	dep3	17
s9	sn9	dep3	18

جدول stud

گزینه اول نادرست است، زیرا بدون group by از having استفاده شده است، بنابراین پرس وجو دارای خطای نحوی است و از سوی کامپایلر اجرا نمی گردد.

گزینه دوم نادرست است، زیرا شروط سازگاری صفات در جلوی group by و select رعایت نشده است، همچنین در خط `having average>avg (average)` به جای `average` سمت چپ باید یک مقدار عددی قرار بگیرد. بنابراین این پرس وجو، خطای نحوی دارد و از سوی کامپایلر اجرا نمی گردد. اگر به طور مثال، گزینه دوم به صورت زیر اصلاح گردد:

```
select dept #
from stud
group by dept #
having avg (average)>15
```

آن گاه، پرس وجوی فوق، شماره دانشکده هایی را می دهد که میانگین معدل آن دانشکده از 15 بالاتر است. پرس وجوی فوق ابتدا براساس dept # به صورت زیر گروه بندی می شود و در ادامه براساس (average) avg، میانگین هر گروه محاسبه می گردد.

dept 1	dept 2
s1 sn1 9	s3 sn3 11
s2 sn2 10	s4 sn4 12
_____	s5 sn5 13
گروه اول	گروه دوم

$$\text{avg(average)} = \frac{19}{2} = 9.5 \quad \text{avg(average)} = \frac{36}{3} = 12$$

dept 3
s6 sn6 14
s7 sn7 15
s8 sn8 17
s9 sn9 18

گروه سوم

$$\text{avg(average)} = \frac{64}{4} = 16$$

سپس براساس دستور $\text{having avg (average)} > 15$ و استفاده از having به عنوان شرط انتخاب گروه، تنها گروه سوم حائز شرط می‌باشد و انتخاب می‌گردد. بنابراین خروجی پرس‌وجو به صورت زیر خواهد بود.

توجه: دستور group by ، سرگروه‌ها را، راهی خروجی می‌کند.

```
dept #
dep3
```

گزینه سوم درست است. به پرس‌وجوی آن توجه کنید:

```
select *
from stud s1
where average > (select AVG (average)
                 from stud s2
                 where s1.dept # = s2.dept #)
```

پرس‌وجوی فوق سطر به سطر توسط عملگر where بر روی جدول $s1$ (جدول stud) حرکت می‌کند، و برای هر سطر از جدول $s1$ ، شرط جلوی where را چک می‌کند، اگر شرط جلوی where برقرار بود، سطر مورد نظر از جدول $s1$ انتخاب می‌گردد و این رویه تا به انتهای جدول $s1$ ادامه پیدا می‌کند. فرض کنید در سطر اول جدول $s1$ قرار داریم...

s #	sname	dept #	average
s1	sn1	dep1	9

با توجه به مقدار dept \# برابر dep1 از سطر جدول $s1$ ، پرس‌وجوی select داخلی، یعنی

```
select AVG (average)
from stud s2
where s1. dept# = s2.dept#)
      dep1
```

میانگین معدل دانشجویانی را از جدول $s2$ محاسبه می‌کند که dept \# آن‌ها برابر dep1 باشد، به بیان دیگر زیر پرس‌وجوی فوق میانگین معدل دانشجویان دانشکده dep1 را محاسبه می‌کند. دو جدول $s1$ و $s2$ که مطابق جدول stud هستند و پرس‌وجوی فوق را تجسم کنید. با توجه به اطلاعات دانشجویان دانشکده

dep1 ، پرواضح است که میانگین معدل دانشجویان دانشکده dep1 برابر $9.5 = \frac{9+10}{2} = \frac{19}{2}$ می‌باشد.

بنابراین دانشجو $s1$ برای نمایش در خروجی انتخاب نمی‌گردد. زیرا:

```
select *
from stud s1
where average > (select AVG (average)
                 from studs2
                 where s1. dept# = s2.dept #)
      9
      9.5      dep1
```

مشاهده می‌کنید که شرط فوق برای سطر اول جدول $s1$ برقرار نیست. بنابراین سطر اول از جدول $s1$ حائز شرایط نیست و برای نمایش در خروجی انتخاب نمی‌گردد. این رویه تا انتهای جدول $s1$ به صورت سطر به سطر، ادامه پیدا می‌کند.

برای مثالی دیگر سطر آخر جدول $S1$ را هم در نظر بگیرید.

با توجه به اطلاعات دانشجویان دانشکده dep3، پرواضح است که میانگین دانشجویان دانشکده dep3 برابر $\frac{14+15+17+18}{4} = \frac{64}{4} = 16$ می‌باشد. بنابراین دانشجو s9 برای نمایش در خروجی انتخاب می‌گردد. زیرا:

```
select *
from stud s1
where average > (select AVG (average)
                 from studS2
                 where s1.dept # = s2.dept #)
16 dep3
```

مشاهده می‌کنید که شرط فوق برای سطر آخر جدول s1 برقرار است. بنابراین سطر آخر از جدول s1 حائز شرایط است و برای نمایش در خروجی، انتخاب می‌گردد. با توجه به اطلاعات دانشجویان دانشکده dep2،

پرواضح است که میانگین معدل دانشجویان دانشکده dep2 برابر $\frac{11+12+13}{3} = \frac{36}{3} = 12$ است. بنابراین خروجی پرس‌وجوی مذکور در گزینه سوم به صورت زیر خواهد بود:

s #	sname	dept #	average
s2	sn2	dep1	10
s5	sn5	dep2	13
s8	sn8	dep3	17
s9	sn9	dep3	18

گزینه چهارم نادرست است، زیرا خطای معنایی دارد. این پرس‌وجو خطای نحوی ندارد و از سوی کامپایلر اجرا می‌گردد. اما پاسخ مورد نظر را برآورده نمی‌سازد.

```
select *
from stud
where average > (select AVG (average) from stud)
```

پرس‌وجوی فوق مشخصات دانشجویانی که معدلشان از متوسط معدل کلیه دانشجویان بیش‌تر است را لیست می‌نماید.

با توجه به اطلاعات جدول stud، پرس‌وجوی فوق به صورت زیر خواهد بود.

```
select *
from stud
where average > (select AVG (average) from stud)
13
```

بنابراین میانگین معدل کلیه دانشجویان، برابر 13 می‌شود. البته اگر نوع داده‌ای ستون average صحیح باشد، برابر 13 می‌شود، توجه کنید که اگر تابع عددی AVG بر روی ستونی با نوع داده‌ای صحیح میانگین انجام دهد و حاصل اعشاری گردد، تابع AVG پاسخ را گرد نمی‌کند، بلکه فقط قسمت صحیح را جدا می‌کند و به خروجی انتقال می‌دهد. هم‌چنین اگر نوع داده‌ای ستون average اعشاری باشد، میانگین برابر 13.2 می‌شود. ما نوع داده‌ای average را در اینجا صحیح در نظر گرفته‌ایم.

بنابراین حاصل پرس‌وجوی فوق به صورت زیر خواهد بود:

s #	sname	dept #	average
s6	sn6	dep3	14
s7	sn7	dep3	15
s8	sn8	dep3	17
s9	sn9	dep3	18

۳۳- گزینه (۳) صحیح است.

راه حل اول:

جداول زیر را در نظر بگیرید:

S#	Sname	Level	..	S#	P#	QTY	P#	...
s1	sn1	6		s1	p1	10	p1	
s2	sn2	2		s1	p2	20	p2	
s3	sn3	3		s2	p1	30		

S تولیدکنندگان
SP تولید
P قطعات

الگوهای چهارگانه (همه، هیچ، بعضی و حداقل) زیر را در نظر بگیرید:

الگوی اول (00):

```
SELECT S#
FROM S
WHERE NOT EXISTS (SELECT *
                   FROM P
                   WHERE NOT EXISTS (SELECT *
                                     FROM SP
                                     WHERE S.S# = SP.S#
                                     AND P.P# = SP.P#))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«شماره تولیدکنندگانی که، وجود نداشته باشد (NOT EXISTS) قطعه‌ای که، تولید نشده باشد (NOT EXISTS). یعنی شماره تولیدکنندگانی که همه قطعات را تولید کرده‌اند.»
با توجه به جداول فوق خروجی پرس و جوی الگوی اول به صورت زیر است:

S#
s1

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول P بررسی می‌گردد، تا مشخص گردد آیا هیچ سطری در جدول P وجود ندارد که شرط اتصال S.S# = SP.S# AND P.P# = SP.P# موجود در select داخلی را برقرار نکند، اگر وجود نداشته باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

توجه: شرط لازم برای تبعیت از الگوی اول وجود NOT EXISTS اول و NOT EXISTS دوم است و شرط کافی برای تبعیت از الگوی اول وجود شروط اتصال متناسب و سازگار است.

توجه: عملگر تقسیم در جبر رابطه‌ای توسط الگوی اول در SQL قابل پیاده‌سازی است. عملگر تقسیم در جبر رابطه‌ای زمانی مورد استفاده قرار می‌گیرد که پرس و جو بخواهد همه حالت‌های یک اتفاق را بررسی کند. مانند شماره تولید تولیدکنندگانی که همه قطعات را تولید کرده‌اند. که تولیدکردن قطعات توسط تولیدکنندگان، اتفاق و مقسوم است و همه قطعات، حالت‌ها و مقسوم علیه است.

الگوی دوم (01):

```

SELECT S#
FROM S
WHERE NOT EXISTS (SELECT *
                  FROM P
                  WHERE EXISTS (SELECT *
                                FROM SP
                                WHERE S.S# = SP.S#
                                AND P.P# = SP.P#))

```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«شماره تولیدکنندگانی که، وجود نداشته باشد (NOT EXISTS) قطعه‌ای که، تولید شده باشد (EXISTS).
یعنی شماره تولیدکنندگانی که هیچ قطعه‌ای را تولید نکرده‌اند.»
با توجه به جداول فوق خروجی پرس و جوی الگوی دوم به صورت زیر است:

S#
s3

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول P بررسی می‌گردد، تا مشخص گردد آیا هیچ سطری در جدول P وجود ندارد که شرط اتصال S.S# = SP.S# AND P.P# = SP.P# موجود در select داخلی را برقرار کند، اگر وجود نداشته باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

توجه: شرط لازم برای تبعیت از الگوی دوم وجود NOT EXISTS اول و EXISTS دوم است و شرط کافی برای تبعیت از الگوی دوم وجود شروط اتصال متناسب و سازگار است.

الگوی سوم (10):

```

SELECT S#
FROM S
WHERE EXISTS (SELECT *
              FROM P
              WHERE NOT EXISTS (SELECT *
                                FROM SP
                                WHERE S.S# = SP.S#
                                AND P.P# = SP.P#))

```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«شماره تولیدکنندگانی که، وجود داشته باشد (EXISTS) قطعه‌ای که، تولید نشده باشد (NOT EXISTS).
یعنی شماره تولیدکنندگانی که بعضی قطعات را تولید نکرده‌اند.»
با توجه به جداول فوق خروجی پرس و جوی الگوی سوم به صورت زیر است:

S#
s2
s3

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول P بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول P وجود دارد که شرط اتصال S.S# =

SP.S# AND P.P# = SP.P# موجود در select داخلی را برقرار نکند، اگر وجود داشته باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

توجه: شرط لازم برای تبعیت از الگوی سوم وجود **EXISTS** اول و **NOT EXISTS** دوم است و شرط کافی برای تبعیت از الگوی سوم وجود **شرط اتصال** متناسب و سازگار است.

الگوی چهارم (11):

```
SELECT S#
FROM S
WHERE EXISTS (SELECT *
              FROM P
              WHERE EXISTS (SELECT *
                            FROM SP
                            WHERE S.S# = SP.S#
                            AND P.P# = SP.P#))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«شماره تولیدکنندگانی که، وجود داشته باشد (**EXISTS**) قطعه‌ای که، تولید شده باشد (**EXISTS**). یعنی شماره تولیدکنندگانی که حداقل یک قطعه را تولید کرده‌اند.»

با توجه به جداول فوق خروجی پرس و جوی **الگوی چهارم** به صورت زیر است:

```
S#
s1
s2
```

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول P بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول P وجود دارد که شرط اتصال S.S# = SP.S# AND P.P# = SP.P# موجود در select داخلی را برقرار کند، اگر وجود داشته باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

توجه: شرط لازم برای تبعیت از الگوی چهارم وجود **EXISTS** اول و **EXISTS** دوم است و شرط کافی برای تبعیت از الگوی چهارم وجود **شرط اتصال** متناسب و سازگار است. پرس و جوی مطرح شده در گزینه دوم را در نظر بگیرید:

```
SELECT S.sname
FROM Sailor S
WHERE NOT EXISTS (SELECT B.bid
                  FROM Boats B
                  WHERE NOT EXISTS (SELECT R.bid
                                     FROM Reserves R
                                     WHERE S.sid = R.sid
                                     AND B.bid = R.bid))
```

توجه: پرس و جوی مطرح شده در **گزینه دوم**، مطابق **الگوی اول** می‌باشد.

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«نام ملوان‌هایی که، وجود نداشته باشد (**NOT EXISTS**) قایقی که، رزرو نشده باشد (**NOT EXISTS**). یعنی نام ملوان‌هایی که همه قایق‌ها را رزرو کرده‌اند.»

توجه: عملگر تقسیم در جبر رابطه‌ای به فرم زیر نیز در SQL قابل پیاده‌سازی است:

```
SELECT S.sname
FROM Sailor S
WHERE NOT EXISTS ((SELECT B.bid
                   FROM Boats B)
                  EXCEPT
                  (SELECT R.bid
                   FROM Reserves R
                   WHERE S.Sid = R.Sid))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«نام ملوان‌هایی که، وجود نداشته باشد (NOT EXISTS) قایقی که، رزرو نشده باشد (EXCEPT). یعنی نام ملوان‌هایی که همه قایق‌ها را رزرو کرده‌اند.»

بنابراین هر دو گزینه اول و دوم نام ملوان‌هایی که همه قایق‌ها را رزرو کرده‌اند، استخراج می‌کند، بنابراین پاسخ سوال گزینه سوم است.

راه حل دوم:

شماره‌ی تمامی قایق‌ها

```
select S.sname from Sailor S where not Exists (
  (select B.bid from Boats B)
  Except
  (select R.bid from Reserves R
   where S.sid = R.sid))
```

توجه: توسط دستور where در خط زیر:

```
select S.sname from Sailor S where not exists (...)
```

برای هر سطر از جدول Sailor شرط جلوی not exists حاصل یک تفاضل توسط عملگر except می‌باشد، محاسبه می‌گردد، اگر تهی بود، شرط جلوی where همان not exists است، TRUE می‌گردد و سطر مورد نظر از جدول Sailor انتخاب می‌گردد و این رویه برای تک تک سطرهای جدول Sailor، تا به انتهای جدول Sailor ادامه پیدا می‌کند. به بیان دیگر گزینه‌ی اول نام ملوان‌هایی را می‌دهد که پراتنز مقابل not exists برای آن‌ها تهی است. این پراتنز هنگامی تهی می‌شود که حاصل تفاضل بیان شده در این پراتنز تهی شود. حاصل این تفاضل در صورتی تهی می‌شود که تعداد کل قایق‌ها برابر تعداد قایق‌هایی باشد که آن ملوان تاکنون رزرو کرده است. به عبارت دیگر گزینه اول نام ملوان‌هایی را می‌دهد که تاکنون تمام قایق‌ها را رزرو کرده‌اند. می‌توانید برای تجسم بهتر جدول‌های زیر را در نظر بگیرید:

sid	sname	...	sid	Bid	...	bid	...
s1	sn1		s1	b1		b1	
s2	sn2		s1	b2		b2	
s3	sn3		s2	b1			

Sailor ملوان
S
Reserves رزرو
R
Boats قایق
B

توجه: همان‌طور که مشاهده می‌کنید، ملوان s1 همه‌ی قایق‌ها را رزرو کرده است. به بیان دیگر `select B.bid from Boats B where B.bid not exists` دستوری جلوی دستوری `select B.bid from Boats B` را نمایش می‌دهد. جدول قایق را نمایش می‌دهد. با توجه به جداول فوق داریم:

B.bid

b1

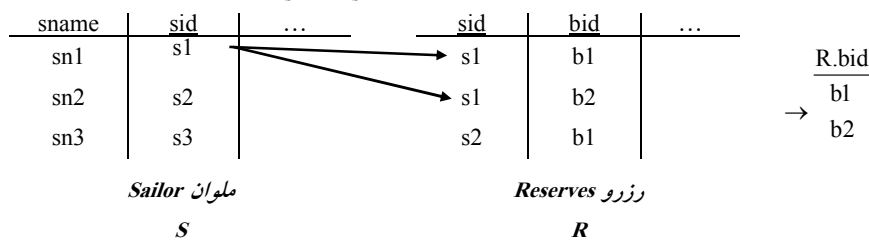
b2

هم‌چنین جمله‌ی بعد از دستور Except یعنی:

```
(select R.bid
from Reserves R
where S.sid=R.sid
```

توسط دستور `S.sid=R.sid` به محیط خارج یعنی جدول S یا Sailor متصل می‌گردد. حال به ازای حرکت در هر سطر از جدول Sailor، یک بار به طور کامل سطرهای جدول رزرو از ابتدا تا انتها با توجه به شرط اتصال بررسی می‌گردد. مطابق شکل زیر:

ابتدا برای سطر اول از جدول Sailor با توجه به شرط اتصال `S.sid = R.sid` داریم:



خروجی حاصل از عملگر تفاضل جلوی `not exists` به ازای سطر اول از جدول Sailor به صورت زیر است:

B.bid R.bid خروجی

b1 - b1 = تهی

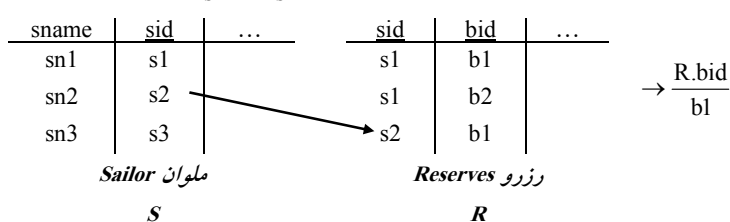
b2 b2

جلوی `not exists` برابر تهی گردید، بنابراین شرط `where not exists` در پشت `not exists` TRUE می‌گردد. بنابراین سطر اول از جدول Sailor به شکل زیر در خروجی نمایش داده می‌شود.

Sname

Sn1

حال برای سطر دوم از جدول Sailor با توجه به شرط اتصال `S.sid = R.sid` داریم:



خروجی حاصل از عملگر تفاضل جلوی `not exists` به ازای سطر دوم از جدول `Sailor` به صورت زیر است:

خروجی $\frac{B.bid}{b1} - \frac{R.bid}{b1} = \frac{b2}{b2}$

جلوی `not exists` برابر غیر تهی گردید، بنابراین شرط `where` در پشت `not exists` `FALSE` می‌گردد، بنابراین سطر دوم از جدول `Sailor` در خروجی نمایش داده نخواهد شد. در ادامه برای سطر سوم از جدول `Sailor` با توجه به شرط اتصال $S.sid = R.sid$ داریم:

sname	sid	...	sid	bid	...
sn1	s1		s1	b1	
sn2	s2		s1	b2	
sn3	s3	هیچ	s2	b1	

$\rightarrow \frac{R.bid}{تهی}$

Sailor ملوان *S*
Reserves رزرو *R*

خروجی حاصل از عملگر تفاضل جلوی `not exists` به ازای سطر سوم از جدول `Sailor S` به صورت زیر است:

خروجی $\frac{B.bid}{b1} - \frac{R.bid}{تهی} = \frac{b1}{b2}$

جلوی `not exists` برابر غیر تهی گردید، بنابراین شرط `where` در پشت `not exists` `FALSE` می‌گردد، بنابراین سطر سوم از جدول `Sailor` در خروجی نمایش داده نخواهد شد. بنابراین خروجی نهایی برای «نام ملوان‌هایی که همه قایق‌ها را رزرو کرده‌اند» به صورت زیر است:

Sname
Sn1

حال به بررسی گزینه دوم می‌پردازیم:

`select S.sname from sailor S where not exists (select B.bid from Boats B where not exists (select R.bid from Reserves R where R.bid=B.bid and R.sid=S.sid))`

الگوریتم فوق به این شکل حرکت می‌کند، به شرح الگوریتم دقت کنید:

ابتدا سطر اول از جدول ملوان بررسی می‌شود تا بررسی کامل در همین سطر توقف کنید، اگر جلوی `not exists` اول تهی باشد، نام ملوان از جدول `Sailor` در خروجی به عنوان ملوانی که همه قایق‌ها را رزرو کرده است، نمایش داده می‌شود. اما مساله این است که چه وقت جلوی `not exists` اول تهی می‌گردد، پاسخ این است که هیچ `B.bid` در جمله:

`select B.bid from Boats B where` موجود نباشد، واضح است که وقتی این اتفاق می‌افتد که شرط `where` در همان جمله برقرار نباشد. اما چه وقت شرط `where` برقرار نیست، پاسخ، وقتی که پرس‌وجوی داخلی

not exists دوم تھی نباشد، یعنی قایق رزرو شده باشد. در یک بیان ساده، ابتدا سطر اول از جدول ملوان را انتخاب می‌کنیم. مثلاً ملوان s1، سپس برای ملوان s1 از جدول ملوان بررسی می‌کنیم، که آیا همه قایق‌های موجود در جدول قایق برای ملوان s1 رزرو شده‌اند یا خیر. بنابراین برای یک ملوان خاص، توسط دستور select جلوی not exists اول، بر روی جدول قایق از ابتدا تا انتهای جدول به ازای یک ملوان خاص، حرکت می‌کنیم، اما به ازای حرکت در هر سطر از جدول قایق، توسط دستور select جلوی not exists دوم، بررسی می‌گردد که قایق موجود در جدول قایق، در جدول رزرو، رزرو شده است یا خیر. اگر رزرو شده باشد، توسط R.bid داخل دستور select جلوی not exists دوم نمایش داده می‌شود و توسط B.bid داخل دستور select جلوی not exists اول نمایش داده نمی‌شود به دلیل FALSE بودن شرط دستور where جلوی not exists اول ناشی از FALSE بودن not exists دوم به دلیل تھی نبودن جلوی not exists دوم. حال اگر مطابق روال فوق به ازای همه قایق‌های موجود در جدول قایق، برای یک ملوان خاص، همه قایق‌ها، رزرو شده باشند، باعث می‌شود تا B.bid داخل دستور select جلوی not exists اول تھی گردد و باعث شود، شرط دستور where در پشت not exists اول TRUE گردد، بنابراین آن ملوان خاص موجود در جدول ملوان در خروجی نمایش داده می‌شود.

پرسوجوی مطرح شده در گزینه دوم را مجدداً در نظر بگیرید:

```
select S.name from Sailor S where not exists (select B.bid from Boats B where
not exists (select R.bid from Reserves R where
```

```
R.bid = B.bid and 

|               |
|---------------|
| R.sid = S.sid |
| s1 s1         |


```

به کادر مستطیل شکل مشخص شده در پرسوجوی فوق توجه کنید به ازای حرکت در هر سطر جدول Sailor جمله R.sid=S.sid تغییر می‌کند. هم اکنون در سطر اول جدول Sailor هستیم. پس sid برابر s1 می‌باشد.

به قطعه کد زیر موجود در جلوی not exists اول، دقت کنید:

```

خروجی
not Exists (select B.bid from Bosts B where
↓
ابتدا سطر اول از جدول قایق یعنی b1 بررسی می‌شود.
not Exsist (select R.bid from Reserves R where
↓
خروجی
```

با شرط زیر بررسی می‌شود که ملوان s1، قایق b1 را رزرو کرده است یا خیر.

```
R.b1 = B.b1 and R.s1 = S.s1))


|      |      |
|------|------|
| TRUE | TRUE |
| TRUE |      |


```

شرط برقرار است، زیرا قایق b1 توسط ملوان s1 رزرو شده است، بنابراین داریم:

not Exists (select $R.bid$ from Reserves R where
 $b1$)

چون جلوی not exists دوم غیر تهی است، بنابراین شرط where موجود پشت not Exists دوم FALSE می‌گردد، بنابراین سطر اول جدول قایق یعنی قایق $b1$ در خروجی دستور select جلوی not exists اول ظاهر نمی‌شود. چون قایق $b1$ توسط ملوان $s1$ رزرو شده است. در ادامه سطر دوم از جدول قایق یعنی $b2$ ، همچنان برای ملوان $s1$ بررسی می‌شود. با شرط زیر بررسی می‌شود که ملوان $s1$ ، قایق $b2$ را رزرو کرده است یا خیر.

$R.b2 = B.b2$ and $R.s1 = S.s1$)
 $\frac{\text{TRUE} \quad \text{TRUE}}{\text{TRUE}}$

شرط برقرار است، زیرا قایق $b2$ توسط ملوان $s1$ رزرو شده است، بنابراین داریم:

not Exists (select $R.bid$ from Reserves R where
 $b2$)

چون جلوی not exists دوم غیر تهی است، بنابراین شرط where موجود پشت not exists دوم FALSE می‌گردد، بنابراین سطر دوم جدول قایق یعنی قایق $b2$ نیز در خروجی دستور select جلوی not exists اول ظاهر نمی‌شود. چون قایق $b2$ توسط ملوان $s1$ رزرو شده است. نتیجه این که ملوان $s1$ همه قایق‌ها را رزرو کرده است.

پس داریم: select s.sname from sailor S where not Exists (select B.bid from Boats B wehre
 در نهایت مقدار B.bid در خط فوق در جلوی not exists اول برای ملوان $s1$ تهی می‌گردد، چون به دلیل شرایط جلوی not exists دوم مشاهده نمودید که هیچ یک از سطرهاى جدول قایق در خروجی دستور select جلوی not exists اول ظاهر نشدند. چون قایق‌های $b1$ و $b2$ توسط ملوان $s1$ رزرو شده بودند. در یک بیان ساده اگر قایقی برای یک ملوان خاص رزرو شده باشد در خروجی select B.bid در خط فوق قرار نمی‌گیرد. رزرو قایق توسط دستور select در جلوی not exists دوم بررسی می‌گردد. جلوی not exists اول، برای سطر اول جدول ملوان، یعنی ملوان $s1$ ، تهی است، زیرا در خروجی select B.bid جلوی not exists اول، قایق‌های رزرو نشده، توسط یک ملوان خاص، خارج می‌گردند، از آن‌جا که پس از بررسی قایق‌های رزرو شده برای ملوان $s1$ توسط دستور select در جلوی not exists دوم، مشخص شد که ملوان $s1$ ، همه‌ی قایق‌ها را رزرو کرده است، بنابراین خروجی select B.bid جلوی not exists اول، تهی خواهد بود، چون همه‌ی قایق‌ها توسط ملوان $s1$ رزرو شده‌اند. پس شرط where پشت not exists اول TRUE است، پس سطر اول جدول ملوان انتخاب می‌گردد و S.sname با مقدار sn1 در خروجی قرار می‌گیرد، زیرا ملوان $s1$ ، همه قایق‌ها را رزرو کرده بود. حال سطر دوم از جدول ملوان بررسی می‌شود، تا بررسی کامل در همین سطر توقف کنید. به قطعه کد زیر، موجود در جلوی not exists اول، دقت کنید:

خروجی
 \nwarrow
 not Exists (select B.bid from Boats B where
 \downarrow

ابتدا سطر اول از جدول قایق یعنی $b1$ بررسی می‌شود.

not Exists (select R.bid from Reserves R where

خروجی

با شرط زیر بررسی می‌شود که ملوان s2، قایق b1 را رزرو کرده است یا خیر.

R.b1 = B.b1 and R.s2 = S.s2))

TRUE	and	TRUE
TRUE		

شرط برقرار است، زیرا قایق b1 توسط ملوان s2 رزرو شده است، بنابراین داریم:

not Exists (select ~~R.bid~~ from Reserves R where
b1

چون جلوی not exists دوم غیر تهی است، بنابراین شرط where موجود پشت not exists دوم FALSE می‌گردد. بنابراین سطر اول جدول قایق یعنی قایق b1 در خروجی دستور select جلوی not exists اول ظاهر نمی‌شود. چون قایق b1 توسط ملوان S2 رزرو شده است. در ادامه سطر دوم از جدول قایق یعنی b2، همچنان برای ملوان s2 بررسی می‌شود.

با شرط زیر بررسی می‌شود که ملوان s2، قایق b2 را رزرو کرده است یا خیر.

R.b2 = B.b2	and	R.s2 = S.s2
FALSE	and	TRUE
FALSE		

شرط برقرار نیست، زیرا قایق b2 توسط ملوان s2 رزرو نشده است، بنابراین داریم:

not exists (select ~~R.bid~~ from Reserves R where

تهی

چون جلوی not exists دوم تهی است، بنابراین شرط where موجود پشت not exists دوم TRUE می‌گردد. بنابراین سطر دوم جدول قایق یعنی قایق b2 در خروجی دستور select جلوی not exists اول ظاهر می‌شود. چون قایق b2 توسط ملوان s2 رزرو نشده است. نتیجه این که ملوان s2 همه قایق‌ها را رزرو نکرده است.

پس داریم: select s.sname from sailor S where not exists (select B.bid from Boots B where در نهایت مقدار B.bid در خط فوق جلوی not exists اول برای ملوان s2 غیر تهی می‌گردد و برابر مقدار b2، چون به دلیل شرایط جلوی not exists دوم مشاهده نمودید که سطر دوم از جدول قایق در خروجی دستور select جلوی not exists اول ظاهر شد، چون قایق b2 توسط ملوان s2 رزرو نشده بود. جلوی not exists اول، برای سطر دوم جدول ملوان، یعنی ملوان s2، غیر تهی است، پس شرط where در پشت not exists اول FALSE است، پس سطر دوم جدول ملوان انتخاب نمی‌گردد، زیرا ملوان s2 همه قایق‌ها را رزرو نکرده بود.

به همین ترتیب سطر سوم جدول ملوان نیز انتخاب نمی‌گردد، زیرا ملوان s3 همه قایق را رزرو نکرده است. نتیجه اینکه، گزینه دوم نیز نام ملوان‌هایی را می‌دهد که قایقی وجود ندارد که آن‌ها آن را رزرو نکرده باشند. به عبارت دیگر نام ملوان‌هایی را می‌دهد که تاکنون همه قایق‌ها را رزرو کرده‌اند.

۳۴- گزینه (۳) صحیح است.

عبارت الف درست است. زیرا، میانگین موجودی حساب‌های افتتاح شده در شعبه ملی را به عنوان خروجی

می‌دهد. هم‌چنین استفاده از تابع avg به درستی انجام شده است. عبارت ب درست است. زیرا، نام شعبه و میانگین موجودی حساب‌ها را در مورد شعبی که میانگین موجودی حساب‌های افتتاح شده در آن‌ها از 1200 بیش تر است، می‌دهد. هم‌چنین صفت همراه گروه‌بندی در خروجی آمده است. عبارت ج درست است. زیرا، نام کلیه شعبی را می‌دهد که موجودی آن‌ها حداقل از موجودی یکی از شعب شهر اصفهان بیش تر است. بنابراین هر سه عبارت صحیح است.

۳۵- گزینه (۲) صحیح است.

جدول زیر را در نظر بگیرید:

S#	P#	QTY
s1	p1	10
s1	p2	20
s1	p3	30
s2	p1	40
s2	p2	50
s3	p3	60

در اجرای این پرس‌وجو ابتدا کلیه سطرهای جدول SP براساس شماره قطعه، گروه‌بندی می‌شوند. سپس تابع عددی sum روی اعضای هر گروه به صورت جداگانه اجرا می‌شود.

p1	p2	p3
s1 10	s1 20	s1 30
<u>s2 40</u>	<u>s2 50</u>	<u>s3 60</u>
+50	+70	+90

در نهایت پرس‌وجوی مطرح شده، شماره و تعداد کل هر قطعه تولید شده را به عنوان خروجی نمایش می‌دهد.

P#	TOTQTY
p1	50
p2	70
p3	90

توجه: پس از گروه‌بندی همواره سر گروه‌ها در اینجا، p1، p2 و p3 در خروجی نشان داده می‌شوند.

۳۶- گزینه (۱) صحیح است.

به جداول زیر دقت کنید:

S#	Sname	...	S#	P#	QTY	P#	Pname	...
s1	sn1		s1	p1	10	p1		
s2	sn2		s1	p2	20	p2		
s3	sn3		s2	p2	30			
			s3	p2	40			

جدول s

جدول sp

جدول p

```
select S.snmae from S where S.s# in (select from SP where sp.p#='p2')
```

پرانتر داخلی شماره تولید کنندگانی را می‌دهد که قطعه p2 را تولید کرده‌اند. خروجی آن به صورت زیر است:

```
SP.s#
s1
s2
s3
```

select بیرونی نام تولید کنندگانی را می‌دهد که قطعه p2 را تولید کرده‌اند، البته چون به دلیل کلید نبودن Sname ممکن است، اسامی برخی تولید کنندگان شبیه هم باشد امکان دارد اسامی تکراری در خروجی این دستور مشاهده شود. بنابراین خروجی Sname به صورت زیر خواهد بود:

```
S.Sname
s1
s2
s3
```

۳۷- گزینه (۲) صحیح است.

جداول زیر را در نظر بگیرید:

<u>isbn</u>	bname	type	<u>ssn</u>	<u>isbn</u>
isb1	b1	technical	ss1	isb1
isb2	b2	fiction	ss1	isb2
isb3	b3	self-help	ss2	isb3
isb4	b4	technical	ss3	isb4

جدول کتاب *Books B*

جدول نویسنده *Authors A*

گزینه اول درست است. زیرا، ابتدا نویسندگانی که کتابی از نوع fiction نوشته‌اند، شناسایی می‌شوند و سپس حاصل تفاضل آن‌ها از کل نویسندگان، خروجی خواسته شده را نمایش می‌دهد.

```
select ssn (select ssn
from Authors Except from Authors A , Books B
where A.isbn=B.isbn and B.type = 'fiction')
```

```
ssn      ssn      ssn
ss1      ss1      ss2
ss1 Except = ss3
ss2
ss3
```

گزینه سوم درست است. زیرا، شماره نویسندگانی که هیچ کتابی از نوع fiction برای آن‌ها ثبت نشده است به عنوان خروجی داده می‌شود که همان خروجی مورد نظر است.

```
select ssn
from Authors A
where Not Exists (select *
from Books B
where A.isbn=B.isbn AND B.type= 'fiction')
```


روال حرکت به این صورت است که به ازای هر سطر از جدول Authors بررسی می‌شود که در پراتنز جلوی not exists با توجه به شرط آن سطری موجود است یا خیر، اگر سطری موجود نباشد، یعنی تهی باشد شرط جلوی where بیرونی، TRUE می‌گردد و سطر مورد نظر از جدول Authors انتخاب می‌گردد. گزینه چهارم درست است. زیرا، شماره نویسندگانی داده می‌شود که عضو مجموعه نویسندگان کتاب‌هایی از نوع fiction نیستند، بنابراین این گزینه نیز خروجی مورد نظر را می‌دهد.

```
select ssn
from Authors
where ssn not in (select ssn
                  from Authors A , Books B
                  where A.isbn= B.isbn AND B.type= 'fiction')
```

```
ssn not in ssn = ssn
ssl      ssl  ss2
ssl                      ss3
ss2
ss3
```

گزینه دوم نادرست است. زیرا، شماره نویسندگانی را می‌دهد که کتابی از غیر نوع fiction تألیف کرده‌اند. اما ممکن است نویسنده‌ای کتابی از نوع غیر fiction تألیف کرده باشد (پس در پاسخ این پرس‌وجو ظاهر می‌شود) و در عین حال کتابی از نوع fiction هم تألیف کرده باشد (پس نباید در خروجی ظاهر می‌شد که می‌شود). پس این گزینه نادرست است. در واقع انتظار این بود که شماره ssn برای نویسندگانی که هیچ کتابی از نوع fiction ننوشته‌اند در خروجی ظاهر شوند، در حالی که با توجه به پرس‌وجوی گزینه دوم شماره ssn با مقدار ssl در خروجی نمایش داده می‌شود که با پرس‌وجوی مطرح شده در صورت سؤال تناقض دارد.

```
select ssn
from Authors, Books
where Authors.isbn= Books. isbn AND type <> 'fiction'
```

پس از ضرب دکارتی دو جدول Authors و Books براساس نوع اتصال و شرط ضرب داریم:

ssn	Authors.isbn	Books.isbn	Type	...
✓ ss1	isb1	isb1	technical	
✗ ss1	isb2	isb2	fiction	
✓ ss2	isb3	isb3	self-help	
✓ ss3	isb4	isb4	technical	

خروجی به صورت زیر خواهد بود:

```
ssn
ssl
ss2
ss3
```

۳۸- گزینه (۳) صحیح است.

دو جدول Account و depositor با مقادیر زیر را در نظر بگیرید:

توجه: دستور GROUP BY، سرگروه‌ها را، راهی خروجی می‌کند. بنابراین مطابق پرس و جوی مطرح شده در گزینه‌ی اول، «نام مشتریانی استخراج می‌شود که در شعبه Perryidge فقط و فقط یک حساب بانکی دارند، نه کمتر و نه بیشتر»
مطابق پرس و جوی مطرح شده در گزینه‌ی دوم، داریم:

```
select cname
from depositor as T, account
where T.acc# = account.acc# and bname='perryidge'
group by cname
having count(T.acc#)<=1
```

با توجه به جداول فوق، ابتدا خروجی ضرب دکارتی و شرط اتصال به صورت زیر محاسبه می‌گردد:

cname	acc#	acc#	bname
alavi	1213	1213	Perryidge
hassani	1215	1215	Perryidge
hassani	1216	1216	Perryidge

همچنین در ادامه، پس از انجام دستور group by cname براساس ستون cname خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:

alavi	hassani
1213 1213 Perryidge	1215 1215 Perryidge
_____	1216 1216 Perryidge
گروه اول	گروه دوم

و در نهایت دستور $having\ count(T.acc\#)\leq 1$ برای هر گروه به طور مستقل اعمال می‌گردد. توجه: دستور HAVING بر روی گروه‌ها، اعمال می‌گردد.

alavi	hassani
1213 1213 Perryidge	1215 1215 Perryidge
_____	1216 1216 Perryidge
گروه اول	گروه دوم
$1 \leq 1,$	$2 \leq 1$

توجه: با توجه به شرط انتخاب گروه توسط دستور $having\ count(T.acc\#)\leq 1$ ، فقط گروه اول جهت نمایش در خروجی انتخاب می‌شود.

و در نهایت دستور cname داخل دستور select برای هر گروه انتخاب شده توسط دستور Having به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد، بنابراین خروجی نهایی پرس و جو به صورت زیر است:

cname
alavi

توجه: دستور GROUP BY، سرگروه‌ها را، راهی خروجی می‌کند. بنابراین مطابق پرس و جوی مطرح شده در گزینه‌ی دوم، «نام مشتریانی استخراج می‌شود که در شعبه

Perryidge فقط و فقط یک حساب بانکی دارند، نه کمتر و نه بیشتر» که مطابق پرس و جوی مطرح شده در گزینه اول است.

مطابق پرس و جوی مطرح شده در گزینه سوم، داریم:

```
select cname
from depositor as T, account
where T.acc# = account.acc# and bname='perryidge'
group by cname
having count (Distinct bname) <=1
```

با توجه به جداول فوق، ابتدا خروجی ضرب دکارتی و شرط اتصال به صورت زیر محاسبه می‌گردد:

cname	acc#	acc#	bname
alavi	1213	1213	Perryidge
hassani	1215	1215	Perryidge
hassani	1216	1216	Perryidge

همچنین در ادامه، پس از انجام دستور group by cname براساس ستون cname خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:

alavi	hassani
1213 1213 Perryidge	1215 1215 Perryidge
_____	1216 1216 Perryidge
گروه اول	گروه دوم

و در نهایت دستور $\text{having count}(\text{Distinct bname}) \leq 1$ برای هر گروه به طور مستقل اعمال می‌گردد. توجه: دستور HAVING بر روی گروه‌ها، اعمال می‌گردد.

alavi	hassani
1213 1213 Perryidge	1215 1215 Perryidge
_____	1216 1216 Perryidge
گروه اول	گروه دوم
$1 \leq 1,$	$1 \leq 1$

توجه: به عملکرد مربوط به عملگر Distinct و حذف تکرار در سطرهای مربوط به ستون bname دقت کنید.

توجه: با توجه به شرط انتخاب گروه توسط دستور $\text{having count}(\text{Distinct bname}) \leq 1$ ، گروه‌های اول و دوم جهت نمایش در خروجی انتخاب می‌شوند.

و در نهایت دستور cname داخل دستور select برای هر گروه انتخاب شده توسط دستور Having به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد، بنابراین خروجی نهایی پرس و جو به صورت زیر است:

cname
alavi
hassani

توجه: دستور GROUP BY، سرگروه‌ها را، راهی خروجی می‌کند.

بنابراین مطابق پرس و جوی مطرح شده در گزینه سوم، «نام مشتریانی استخراج می‌شود که در شعبه

Perryidge حداقل یک حساب بانکی دارند.» که البته پاسخ سوال هم همین گزینه است که حاصل پرس و جوی آن، با سایر گزینه‌ها متفاوت است.

مطابق پرس و جوی مطرح شده در گزینه‌ی چهارم، داریم:

```
select T.cname
from depositor as T
where UNIQUE (select R.cname
               from account, depositor as R
               where T.cname = R.cname and
                     R.acc#=account.acc# and bname='perryidge')
```

با توجه به وجود دستور UNIQUE، زیر پرس و جوی داخلی فوق یک Correlated Subquery است، یعنی به ازای حرکت در هریک از سطرهای پرس و جوی خارجی، یک بار به طور کامل از ابتدا تا انتها زیر پرس و جوی داخلی اجرا و بر اساس شرطی که زیر پرس و جوی داخلی را به پرس و جوی خارجی متصل می‌کند، بررسی انجام می‌شود. مانند دو حلقه تو در تو، که به ازای هربار اجرای حلقه خارجی، یک بار به طور کامل حلقه داخلی اجرا می‌گردد.

توسط دستور where در فرم زیر:

```
select T.cname
from depositor as T
where UNIQUE (...)
```

برای هر سطر از جدول depositor شرط جلوی UNIQUE که حاصل یک مقایسه می‌باشد، محاسبه می‌گردد، اگر فقط و فقط یک سطر وجود داشت یعنی نه کمتر و نه بیشتر، آنگاه شرط جلوی where که همان UNIQUE است، TRUE می‌گردد و سطر مورد نظر از جدول depositor انتخاب می‌گردد و این رویه برای تک تک سطرهای جدول depositor، تا به انتهای جدول depositor ادامه پیدا می‌کند، دقت کنید که دستور UNIQUE سور یکه است. به بیان دیگر این پرس و جو نام مشتریانی را می‌دهد که در پرائنز مقابل UNIQUE برای آن‌ها فقط و فقط یک سطر وجود داشته باشد. در این پرائنز هنگامی فقط و فقط یک سطر وجود دارد که در حاصل مقایسه بیان شده در این پرائنز فقط و فقط یک سطر وجود داشته باشد. به عبارت دیگر در پرس و جوی گزینه‌ی چهارم نام مشتریانی استخراج می‌شود که در شعبه Perryidge فقط و فقط یک حساب بانکی دارند، نه کمتر و نه بیشتر. که مطابق پرس و جوی مطرح شده در گزینه‌های اول و دوم است. بنابراین مطابق پرس و جوی مطرح شده در گزینه‌ی چهارم، «نام مشتریانی استخراج می‌شود که در شعبه Perryidge فقط و فقط یک حساب بانکی دارند، نه کمتر و نه بیشتر»

عبارت موجود در جلوی دستور UNIQUE که به صورت زیر است:

```
select R.cname
from account, depositor as R
where T.cname = R.cname and
      R.acc#=account.acc# and bname='perryidge')
```

توسط دستور 'T.cname = R.cname and R.acc# = account.acc# and bname = 'perryidge' به محیط خارج یعنی جدول depositor متصل می‌گردد. حال به ازای حرکت در هر سطر از جدول depositor، یک بار به طور کامل سطرهای حاصل از ضرب دکارتی دو جدول account و depositor از ابتدا تا انتها با توجه به شرط اتصال بررسی می‌گردد. مطابق شکل زیر:

ابتدا برای سطر اول از جدول depositor با توجه به شرط اتصال
 داریم: T.cname = R.cname and R.acc# = account.acc# and bname = 'perryidge'

T.cname	acc#	...
alavi	1213	
alavi	1214	
hassani	1215	
hassani	1216	
asgari	1217	

جدول مشتری depositor T

R.cname	R.acc#	account.acc#	bname
alavi	1213	1213	Perryidge
hassani	1215	1215	Perryidge
hassani	1216	1216	Perryidge

جلوی UNIQUE فقط و فقط یک سطر وجود دارد، بنابراین شرط where در پشت UNIQUE، TRUE می‌گردد. بنابراین سطر اول از جدول depositor در خروجی نمایش داده می‌شود.

در ادامه برای سطر دوم از جدول depositor با توجه به شرط اتصال
 داریم: T.cname = R.cname and R.acc# = account.acc# and bname = 'perryidge'

T.cname	acc#	...
alavi	1213	
alavi	1214	
hassani	1215	
hassani	1216	
asgari	1217	

جدول مشتری depositor T

R.cname	R.acc#	account.acc#	bname
alavi	1213	1213	Perryidge
hassani	1215	1215	Perryidge
hassani	1216	1216	Perryidge

جلوی UNIQUE فقط و فقط یک سطر وجود دارد، بنابراین شرط where در پشت UNIQUE، TRUE می‌گردد. بنابراین سطر دوم از جدول depositor در خروجی نمایش داده می‌شود.

در ادامه برای سطر سوم از جدول depositor با توجه به شرط اتصال
 داریم: T.cname = R.cname and R.acc# = account.acc# and bname = 'perryidge'

T.cname	acc#	...
alavi	1213	
alavi	1214	
hassani	1215	
hassani	1216	
asgari	1217	

جدول مشتری depositor T

R.cname	R.acc#	account.acc#	bname
alavi	1213	1213	Perryidge
hassani	1215	1215	Perryidge
hassani	1216	1216	Perryidge

جلوی UNIQUE فقط و فقط یک سطر وجود ندارد یعنی دو سطر وجود دارد، بنابراین شرط where در پشت UNIQUE، FALSE می‌گردد. بنابراین سطر سوم از جدول depositor در خروجی نمایش داده نمی‌شود.

در ادامه برای سطر چهارم از جدول depositor با توجه به شرط اتصال داریم: T.cname = R.cname and R.acc# = account.acc# and bname = 'perryidge'

T.cname	acc#	...
alavi	1213	
alavi	1214	
hassani	1215	
hassani	1216	
asgari	1217	

depositor T جدول مشتری

R.cname	R.acc#	account.acc#	bname
alavi	1213	1213	Perryidge
hassani	1215	1215	Perryidge
hassani	1216	1216	Perryidge

جلوی UNIQUE فقط و فقط یک سطر وجود ندارد یعنی دو سطر وجود دارد، بنابراین شرط where در پشت UNIQUE، FALSE می‌گردد. بنابراین سطر چهارم از جدول depositor در خروجی نمایش داده نمی‌شود.

در ادامه برای سطر پنجم از جدول depositor با توجه به شرط اتصال داریم: T.cname = R.cname and R.acc# = account.acc# and bname = 'perryidge'

T.cname	acc#	...
alavi	1213	
alavi	1214	
hassani	1215	
hassani	1216	
asgari	1217	

depositor T جدول مشتری

R.cname	R.acc#	account.acc#	bname
alavi	1213	1213	Perryidge
hassani	1215	1215	Perryidge
hassani	1216	1216	Perryidge

جای UNIQUE فقط و فقط یک سطر وجود ندارد یعنی هیچ سطری وجود ندارد، بنابراین شرط where در پشت UNIQUE، FALSE می‌گردد. بنابراین سطر پنجم از جدول depositor در خروجی نمایش داده نمی‌شود.
و در نهایت دستور T.cname داخل دستور select اعمال می‌گردد، بنابراین خروجی نهایی پرس و جو به صورت زیر است:

```
T.cname
-----
alavi
alavi
```

توجه: بهتر بود طراح محترم جهت حذف سطرها تکراری در پشت ستون T.cname از دستور Distinct استفاده می‌نمود، به صورت زیر:

```
select Distinct T.cname
from depositor as T
where UNIQUE (...)
```

که در اینصورت خروجی نهایی پرس و جو به صورت زیر می‌بود:

```
T.cname
-----
alavi
```

بنابراین مطابق پرس و جوی مطرح شده در گزینه‌ی چهارم، «نام مشتریانی استخراج می‌شود که در شعبه Perryidge فقط و فقط یک حساب بانکی دارند، نه کمتر و نه بیشتر» که مطابق پرس و جوی مطرح شده در گزینه‌های اول و دوم است.

توجه: اگر در پرس و جوی مطرح شده در گزینه چهارم به جای عملگر UNIQUE از عملگر Exists استفاده می‌شد، به صورت زیر:

```
select Distinct T.cname
from depositor as T
where Exists (select R.cname
              from account, depositor as R
              where T.cname = R.cname and
                    R.acc#=account.acc# and bname='perryidge')
```

آنگاه حاصل پرس و جو همانند گزینه‌ی سوم می‌شد، خروجی نهایی پرس و جو به صورت زیر است:

```
cname
-----
alavi
hassani
```

یعنی، «نام مشتریانی استخراج می‌شود که در شعبه Perryidge حداقل یک حساب بانکی دارند.»

۳۹- گزینه (۱) صحیح است.

راه حل اول: پرس و جوی مطرح شده در صورت سوال را در نظر بگیرید:

```
SELECT distinct S.customer-name
FROM depositor as S
WHERE NOT EXISTS ((SELECT branch-name
                   FROM Branch
```



```

WHERE branch-city='Brooklyn')
EXCEPT
(SELECT R.branch-name
FROM depositor as T, account as R
WHERE T.account-number = R.account-number
And S.customer-name=T.customer-name))

```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«نام مشتریانی که، وجود نداشته باشد (NOT EXISTS) شعبه‌ای از شهر Brooklyn که، حساب در آن باز نشده باشد (EXCEPT). یعنی نام مشتریانی که در همه شعب شهر Brooklyn حساب باز کرده‌اند.»

توجه: عملگر تقسیم در جبر رابطه‌ای به فرم فوق در SQL قابل پیاده‌سازی است.

راه حل دوم: در زیر پرس و جوی مقابل عملگر not exists ابتدا تفاضل اسامی کل شعب شهر Brooklyn و شعبی که مشتری در آنها حساب دارد، استخراج می‌گردد. سپس در پراتز بیرونی نام مشتریانی به عنوان خروجی داده می‌شود که نتیجه قبل در مورد آنها تهی باشد، یعنی نام مشتریانی که شعبه‌ای از Brooklyn وجود ندارد که آن‌ها در آن شعبه حساب ندارند. به این ترتیب به عنوان خروجی نهایی پرس و جوی داده شده نام مشتریانی داده می‌شود که در تمامی شعب شهر Brooklyn حساب دارند.

مثال: جداول Branch, Account, Depositor، با مقادیر زیر را در نظر بگیرید:

branch-name	branch-city	assets	account-number	branch-name	balance
A	Brooklyn	700	11	A	10
B	Brooklyn	500	11	B	20
C	Brooklyn	800	11	C	30
D	Tehran	900	11	D	40
			12	A	100
			12	B	90
			12	C	80
			13	A	50
			14	D	60

Branch شعب
Assets : دارایی شعبه است.

Account R موجودی

customer-name	account-number
Alavi	11
Hassani	12
Taghavei	13
Javadi	14

T و Depositor S مشتریان

با توجه به پرس و جوی مطرح شده، به ازای حرکت در هر سطر از جدول S تفاضل حاصل از اجرای کامل دو طرف عملگر except محاسبه می‌گردد، اگر حاصل تفاضل تهی گردد. شرط where در پشت

TRUE .not exists می گردد، و سطر مورد نظر از جدول S، در خروجی نمایش داده می شود، البته با توجه به ستون S.Customer-name در دستور select، نام مشتری در خروجی نمایش داده می شود.

توجه: دستور S.customer-name=T.customer-name نقطه اتصال جدول حاصل از ضرب دکارتی دو جدول T و R به جدول s در select خارجی به ازای حرکت در هر سطر از جدول s است و دستور T.account-number=R.account-number شرط و نوع اتصال ضرب دکارتی دو جدول T و R را مشخص می کند.

۴۰- گزینه (۲) صحیح است.

هر دو عبارت، دو ستون a و b را در خروجی می آورند.

عبارت Q1 سطرهایی از R که مقادیر دو ستون (a,b) آن ها غیرتکراری است را انتخاب کرده و فقط دو ستون a و b را در خروجی نشان می دهد.

عبارت Q2 نیز ابتدا رابطه ی R را برحسب مقادیر a,b گروه بندی می کند و از هر گروه فقط مقادیر a و b (که مطمئناً غیرتکراری هستند) را در خروجی نشان می دهد.

پس دو عبارت Q1 و Q2 پاسخ های یکسانی تولید می کنند.

مثال: جدول R(a,b,c) را با مقادیر زیر در نظر بگیرید:

a	b	c
1	2	3
1	2	4
3	4	5
3	4	6

Q1 : select distinct a,b
from R

a	b
1	2
3	4

Q2 : select a,b
from R
group by a,b

جدول R براساس (a,b) به صورت زیر گروه بندی می شود:

(1,2)	(3,4)
3	5
<u>4</u>	<u>6</u>
گروه اول	گروه دوم

توجه: در گروه بندی، سرگروه ها، راهی خروجی می شوند، بنابراین داریم:

a	b
1	2
3	4

همان طور که مشاهده می شود Q1 و Q2 پاسخ های یکسان تولید می کنند.

۴۱- گزینه (۳) صحیح است.

دستور select سوم، شماره قطعاتی را می‌دهد که رنگشان قرمز است. دستور select دوم، شماره‌ی تولید کنندگانی را می‌دهد که قطعه‌ای از مجموعه‌ی فوق (خروجی دستور select سوم) را تولید کنند، یعنی شماره‌ی تولید کنندگانی را می‌دهد که قطعه‌ای به رنگ قرمز تولید می‌کنند. بالاخره دستور select اول، نام تولید کنندگانی را می‌دهد که شماره‌ی آن‌ها در خروجی دستور select دوم باشد، یعنی نام تولید کنندگانی که قطعه‌ای (حداقل یکی) به رنگ قرمز تولید می‌کنند.

۴۲- گزینه (۲) صحیح است.

در پرس و جوی داده شده، به ازای هر سطر از جدول S، سطرهایی از جدول SP که شماره‌ی تولیدکننده (S#) آن‌ها با شماره‌ی تولیدکننده‌ی جدول S برابر باشد و قطعه‌ای با شماره‌ی 'P2' تولید کند را انتخاب می‌کند، اگر حداقل یک سطر از SP انتخاب شود، سطر متناظر از S در خروجی ظاهر می‌شود. در نهایت پرس و جوی داده شده، نام تولیدکنندگانی را می‌دهد که قطعه‌ی P2 را تولید می‌کنند. به جداول زیر دقت کنید:

s#	Sname	...	s#	p#	QTY	p#	pname
s1	sn1		s1	p1	10	p1	
s2	sn1		s1	p2	20	p2	
s3	sn3		s2	p2	30		
			s3	p2	40		

جدول S

جدول SP

جدول P

```
select S.sname
from s
where exists (select *
              from sp
              where S.s# = SP.s# AND SP.p# = 'p2')
```

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول s، یک بار به طور کامل از ابتدا تا انتهای جدول sp بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول sp قرار دارد که شرط موجود در select داخلی را برقرار کند، اگر وجود داشته باشد، سطر مورد نظر از جدول s در خروجی نمایش داده می‌شود.

پرس و جوی فوق در واقع نام تولید کنندگانی را نمایش می‌دهد که برای آن‌ها تولید قطعه 'p2' ثبت شده است (تولید قطعه 'p2' برای آن‌ها وجود دارد) به بیان دیگر پرس و جوی فوق نام تولید کنندگانی را می‌دهد که قطعه 'p2' را تولید کرده‌اند، در نهایت select بیرونی، نام تولید کنندگانی را می‌دهد که قطعه p2 را تولید کرده‌اند، البته چون به دلیل کلید نبودن sname، ممکن است اسامی برخی تولید کنندگان شبیه هم باشد، امکان دارد، اسامی تکراری در خروجی این پرس و جو مشاهده شود. بنابراین خروجی sname به صورت زیر خواهد بود.

S.sname
sn1
sn1
sn3

توجه: SQL به طور پیش فرض سطرهای تکراری را حذف نمی‌کند، مگر از دستور Distinct استفاده شود.

بنابراین گزینه چهارم نادرست است. زیرا عبارت «بدون تکرار» را لحاظ کرده است.

۴۳- گزینه (۴) صحیح است.

هر سه دستور sql نوشته شده، یک کار را انجام می دهند.

دستور select اول، مشخصات قطعاتی را می دهد که spin آنها برابر 1000 یا 2000 باشد.

دستور select دوم، مشخصات قطعاتی را می دهد که spin آنها عضو مجموعه‌ی {1000 و 2000} باشد.

دستور select سوم، اجتماع دو مجموعه را می دهد: مجموعه‌ی قطعاتی که spin آنها برابر 1000 است و مجموعه‌ی قطعاتی که spin آنها برابر 2000 است.

۴۴- گزینه (۳) صحیح است.

با توجه به شرط $T1.B = T2.B$ ، باید ببینیم هر سطر از جدول T1 با چند سطر از جدول T2 تطابق دارد. (منظور از «تطابق» برابری مقدار فیلد B در دو جدول است.) سطر اول از T1 با سطرهای اول و دوم از T2 تطابق دارد، پس به ازای آنها، دو سطر در خروجی تولید می شود. سطر دوم از T1 با سطرهای سوم، چهارم و پنجم از T2 تطابق دارد. پس به ازای آنها سه سطر در خروجی تولید می شود. سطر سوم از T1 با هیچ سطری از T2 تطابق ندارد و به ازای آن هیچ سطری در خروجی تولید نمی شود. پس تعداد کل سطرهای خروجی برابر است با: $2+3=5$

T1		T2	
A	B	B	C
a1	b1	b1	c1
a2	b2	b1	c2
a3	b3	b2	c2
		b2	c3
		b2	c4

خروجی دستور select نوشته شده به صورت زیر است:

B
b1
b1
b2
b2
b2

توجه کنید که خروجی‌های تکراری ایجاد می شوند چون جلوی select از distinct استفاده نشده است.

توجه: SQL به طور پیش فرض سطرهای تکراری را حذف نمی کند، مگر از دستور distinct استفاده شود.

۴۵- گزینه (۱) صحیح است.

در پرس و جوی نوشته شده از جدول enroll (به معنی «ثبت نام») استفاده شده است که حداقل شامل سه فیلد crsno، pname و sname است. این فیلدها به ترتیب، شماره‌ی درس، نام استاد آن درس و نام دانشجویی که در آن درس ثبت نام کرده است را نشان می دهد.

برای توصیف خروجی پرس و جوی داده شده بهتر است ابتدا خروجی دستور select داخلی را تعیین کنیم.

به راحتی می‌توان دید که دستور select داخلی، مجموعه‌ی شماره‌ی درس‌هایی که دانشجویی با نام کاظمی در آن‌ها ثبت‌نام کرده است را به ما می‌دهد. در نتیجه، select بیرونی، نام اساتیدی را می‌دهد که درسی را تدریس کنند که شماره‌ی آن درس متعلق به مجموعه‌ی فوق باشد. یعنی حداقل یکی از درس‌هایی را که کاظمی در آن‌ها ثبت‌نام کرده است، تدریس کرده‌اند. بنابراین کل عبارت پرس‌وجو، نام اساتیدی (غیرتکراری) را می‌دهد که حداقل یکی از درس‌هایی را تدریس می‌کنند که کاظمی در آن‌ها ثبت‌نام کرده است.

مثال: به جدول زیر دقت کنید:

Sname	pname	Crsno
✓ کاظمی	اکبری	12
✓ کاظمی	حسینی	13
✓ کاظمی	حسینی	14
* جوادی	حسینی	15
* عسگری	سجادی	16
* تقوی	اکبری	17

جدول enroll

براساس پرس‌وجوی مطرح شده داریم:

```
select distinct pname
from enroll
where crsno in (select crsno
                from enroll
                where sname = 'kazemi')
12,13,14
```

در پرس‌وجوی فوق به ازای حرکت در هر سطر از جدول enroll مجموعه جلوی in بررسی می‌گردد. آیا crsno داخل این مجموعه قرار دارد یا خیر. اگر قرار داشت سطر مورد نظر از جدول enroll در خروجی نمایش داده می‌شود.

در نهایت خروجی پرس‌وجوی فوق به صورت زیر است:

```
Pname
اکبری
حسینی
```

توجه: سطرهای تکراری به دلیل استفاده از دستور distinct حذف گردید.

گزینه دوم نادرست است. زیرا استادانی مدنظر است که در حال حاضر به کاظمی تدریس می‌کنند و نه استادان قبل‌تر.

توجه: جدول enroll، جدول ثبت نام است.

۴۶- گزینه (۳) صحیح است.

عبارت حساب رابطه‌ای داده شده، دو سطر (a, b_1) و (c, b_2) از رابطه‌ی r را پیدا می‌کند که در آن‌ها $b_1 > b_2$ باشد و سطر (a, c) در رابطه‌ی S باشد و به ازای چنین سطرهایی مقدار a را بر می‌گرداند. به بیان دیگر عبارت حساب رابطه‌ای داده شده مؤلفه اول $\langle a \rangle$ سطرهایی از رابطه s را نمایش می‌دهد که دو مؤلفه

آن‌ها $\langle a, c \rangle \in s$ مؤلفه‌های اول دو سطر در رابطه r را تشکیل می‌دهند و رابطه‌ی $b_1 > b_2$ میان مؤلفه‌های دوم سطرهای مورد نظر در رابطه r برقرار است.

A	C
a	c

A	B
a	b ₁
	b ₂

, $b_1 > b_2$

جدول r جدول s

گزینه‌های اول و دوم و چهارم نیز چنین کاری می‌کنند. در گزینه‌ی اول، قسمت زیر از دستورات را در نظر بگیرید:

```
from r as r1, r as r2
```

```
where r1.B > r2.B
```

عبارت `from`، هر سطر از r را با سطر دیگری از آن در نظر می‌گیرد و عبارت `where` سطرهایی را انتخاب می‌کند که فیلدهای B آن‌ها برابر نباشند (همان شرط $b_1 > b_2$).

سپس وجود سطر (a, c) در جدول S در دو خط زیر بررسی می‌شوند:

```
where (A,C) in {
```

```
select r1.A , r2.A
```

در گزینه‌ی دوم، عبارت `from r as r1` و `r as r2` هر دو سطری از r را با هم در نظر می‌گیرد و سپس شرط جلوی `where` دوم، سطرهایی که شرط $r2.B > r1.B$ (یا همان $b_1 > b_2$) و شرط

$r1.A = s.C$ AND $r2.A = s.A$ (معادل با وجود سطر (a, c) در S) را داشته باشند را انتخاب می‌کند.

در گزینه‌ی چهارم، شرط $r1.B > r2.B$ معادل شرط $b_1 > b_2$ است و ادامه‌ی شرط `where` (بعد از AND) معادل با وجود سطر (a, c) در S است.

با توجه دقیق به گزینه‌های اول و چهارم می‌توان دریافت که این دو گزینه دو سوی مختلف یک مطلب یکسان را بیان می‌کنند و با هم معادل هستند. گزینه دوم نیز پیاده‌سازی شده عملگر `in` را با استفاده از عملگر `exists` نشان می‌دهد و با گزینه اول و چهارم معادل است.

چنانکه ملاحظه می‌شود در عبارت حساب رابطه‌ای داده شده هیچ قیدی در مورد یکسان بودن مقدار فیلد A در سطرهایی از دو جدول r و s ذکر نشده است. اما در گزینه‌ی سوم، با انجام NATURAL INNER JOIN، سطرهایی از r و s که فیلدهای A در آن‌ها مقدار یکسانی دارند انتخاب می‌شوند که اشتباه است. در واقع گزینه سوم با عبارت حساب رابطه‌ای داده شده معادل نیست.

مثال: دو جدول $r(A, B)$ و $s(A, C)$ ، با مقادیر زیر را در نظر بگیرید:

A	C
a	c

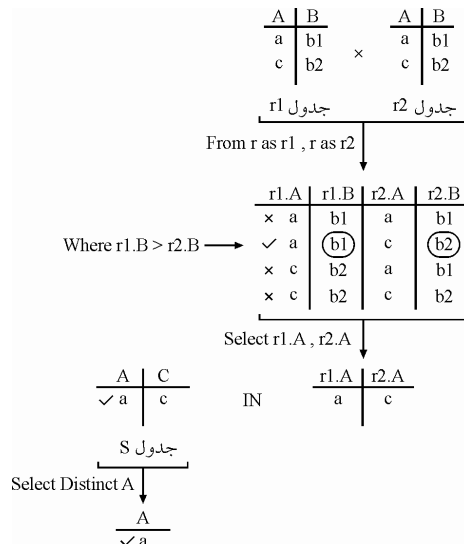
A	B
a	b ₁
	b ₂

جدول r جدول s

توجه: در جدول r ، مقدار b_1 بزرگ‌تر از مقدار b_2 در نظر گرفته شده است.

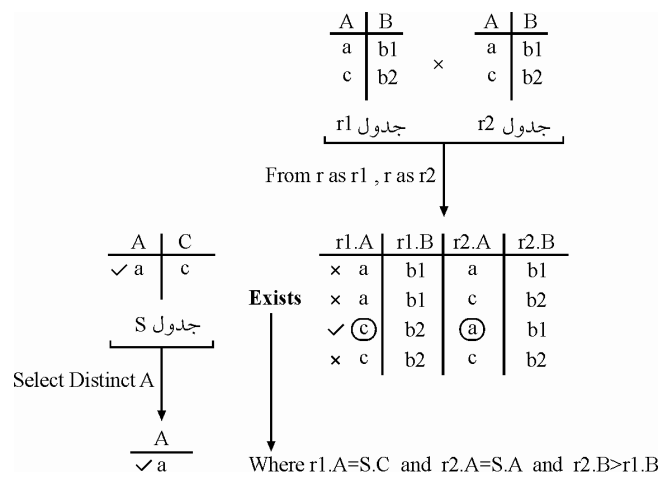
گزینه اول: در این گزینه به ازای حرکت در هر سطر از جدول s ، یک بار به طور کامل `select` داخلی اجرا می‌گردد. سپس عضویت مقادیر (A, C) از جدول s در محتوای حاصل از ضرب دکارتی مطرح شده با

شرایط ویژه‌اش بررسی می‌گردد. اگر عضویت برقرار بود، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.



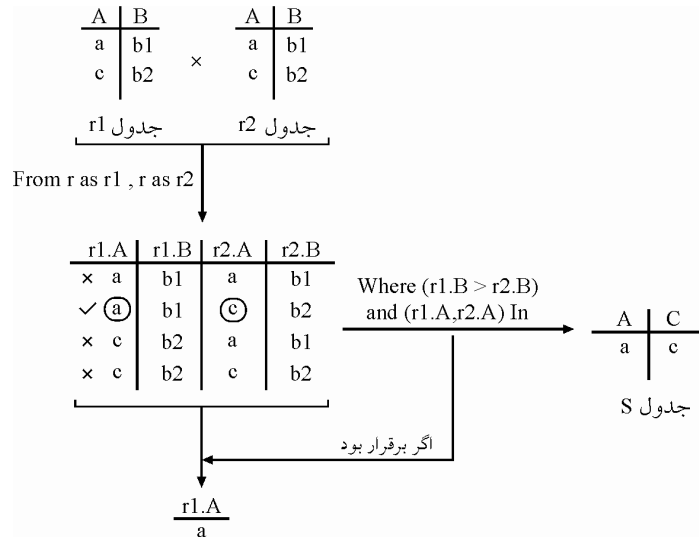
گزینه دوم: این گزینه پیاده‌سازی شده عملگر in ، با استفاده از عملگر $exists$ است.

در این گزینه به ازای حرکت در هر سطر از جدول S، یکبار به طور کامل در $select$ داخلی بررسی می‌گردد که آیا حداقل یک سطر وجود دارد که شرایط موجود در $select$ داخلی را برآورده سازد، اگر بود، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

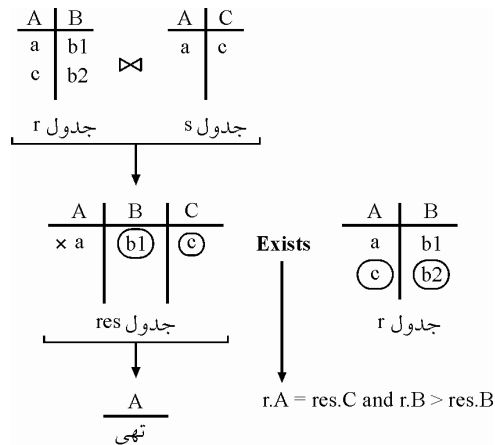


توجه: شیوهی جستجو در دستور $where$ با گزینه اول متفاوت است، اما نتیجه یکسان خواهد بود.

گزینه چهارم: این گزینه طرف مقابل گزینه اول اما با پاسخ یکسان را پیاده‌سازی می‌کند.



گزینه سوم: در این گزینه به دلیل استفاده از عملگر الحاقی طبیعی، سطریهایی از r و s که در ستون مشترک A، هم مقدار هستند، جهت نمایش در خروجی انتخاب می‌شوند. در حالی که در گزینه‌های اول، دوم و چهارم سطریهایی از جدول s انتخاب می‌شوند که مقادیر ستون A و C از جدول s برابر مقادیر ستون A در دو سطر از جدول r باشد که $b1 > b2$ برای ستون B در جدول r را نیز رعایت کند. بنابراین پرس‌وجوی گزینه سوم کاملاً متفاوت با پرس‌وجوهای سایر گزینه‌ها انجام می‌دهد.



۴۷- گزینه (۲) صحیح است.

دستور Update برای تغییر دادن و اصلاح کردن فیلدهای جداول می‌باشد. برای مثال، اگر اسم جدول مسافران، Passenger و اسم فیلد متناظر با نام مسافر، Pname باشد، برای تغییر تمام نام‌های «اکبرزاده» به «اکبرزاده» می‌نویسیم:

update Passenger

set Pname = «اکبرزاده»

where Pname = «اکبرزاده»

۴۸- گزینه (۱) صحیح است.

باید تعداد کد درس‌ها (#c) را در ترم 892 (یعنی term = 892) را بشماریم. اما توجه کنید که جدول Enroll جدول ثبت‌نام‌هاست (چه دانشجویی چه درسی را در چه ترمی با چه استادی گرفته است) نه جدول دروس ارائه شده. پس در جدول Enroll ممکن است یک کد درس (#c) چندین بار تکرار شده باشد (چون یک درس توسط دانشجویان مختلفی گرفته می‌شود و به ازای هر کدام یک سطر در جدول Enroll وجود دارد). پس باید از distinct استفاده کنیم تا موارد تکراری را نشمارد یا به عبارت دیگر، هر مقدار از #c را فقط یک بار بشمارد.

توجه کنید که کلمه distinct همان‌طور که در گزینه اول نوشته شده است، باید قبل از فیلد مربوطه (#c) بیابد و نمی‌تواند مانند گزینه دوم نوشته شود. گزینه دوم خطای نحوی دارد.

گزینه سوم اشتباه است چون از کلمه distinct استفاده نکرده است. در واقع گزینه سوم موارد تکراری را هم می‌شمارد. گزینه چهارم نیز اشتباه است چون از کلمه count استفاده نکرده و در نتیجه تعداد درس‌ها را به ما نمی‌دهد بلکه کد درس‌ها را می‌دهد.

۴۹- گزینه (۴) صحیح است.

هر سه عبارت پرس‌وجوی داده شده یک کار را انجام می‌دهند:

عبارت I، مشخصات تولیدکنندگانی را می‌دهد که شهر آن‌ها برابر Hamadan یا Tabriz باشد.

عبارت II، مشخصات تولیدکنندگانی را می‌دهد که شهر آن‌ها در مجموعه { 'Hamadan', 'Tabriz' }

وجود داشته باشد.

عبارت III، با استفاده از عملگر union اجتماع مجموعه «تولید کنندگان همدان» و مجموعه «تولیدکنندگان تبریز» را به ما می‌دهد.

۵۰- گزینه (۱) یا (۳) صحیح است.

صورت سؤال دو اشکال دارد:

(۱) پیوندی به نام 'Left inner join' وجود ندارد. منظور طراح سؤال می‌تواند پیوند 'inner join' یا 'Left outer join' بوده باشد.

(۲) پیوندهای inner join و outer join همواره با عبارت «شرط on» همراه‌اند که عبارت «شرط» نشان می‌دهد چه سطریایی از دو جدول باید با هم پیوند زده شوند.

بنابراین، دستور SQL داده شده باید به یکی از دو شکل زیر می‌بود:

```
select T1.B
from T1 inner join T2 on T1.B=T2.B
```

یا

```
select T1.B
from T1 Left outer join T2 on T1.B=T2.B
```

دستور اول، هر یک از سطریهای T1 را با تک تک سطریهای T2 مقایسه می‌کند، هر جا شرط T1.B=T2.B برقرار بود مقدار T1.B را در خروجی می‌نویسد. توجه کنید که چون از distinct استفاده نشده پس سطریهای تکراری در خروجی ممکن است چاپ شوند. بنابراین، سطر اول T1 با سطریهای اول و دوم T2 و سطر دوم از T1 با سطریهای سوم، چهارم و پنجم از T2 تطابق دارند و بنابراین پنج سطر زیر در خروجی

ظاهر می‌شود. (سطر سوم از T1 با هیچ سطری از T2 تطابق ندارد):

T1		T2		خروجی
A	B	B	C	T1.B
a1	b1	b1	c1	b1
a2	b2	b1	c2	b1
a3	b3	b2	c2	b2
		b2	c3	b2
		b2	c4	b2

در دستور select دوم، پیوند Left outer join باعث می‌شود علاوه بر خروجی‌هایی که توسط inner join تولید می‌شوند، سطرهایی از T1 (جدول سمت چپ پیوند) که با هیچ سطری از T2 تطابق ندارند نیز در خروجی پیوند ظاهر شوند (فیلدهای متناظر با T2 در خروجی برابر NULL خواهد بود). سپس فقط ستون T1.B در خروجی می‌آید. پس سطر سوم از T1 نیز در خروجی پیوند ظاهر می‌شود و فیلد T1.B از آن نیز

در خروجی می‌آید:

T1.B
b1
b1
b2
b2
b2
b3

۵۱- گزینه (۲) صحیح است.

در پرس‌وجوی نوشته شده از جدول enroll (به معنی «ثبت نام») استفاده شده است که حداقل شامل سه فیلد crsno، pname و sname است. این فیلدها به ترتیب، شماره‌ی درس، نام استاد آن درس و نام دانشجویی که در آن درس ثبت‌نام کرده است را نشان می‌دهد.

برای توصیف خروجی پرس‌وجوی داده شده بهتر است ابتدا خروجی دستور select داخلی را تعیین کنیم. به راحتی می‌توان دید که دستور select داخلی، مجموعه‌ی شماره‌ی درس‌هایی که دانشجویی با نام ذاکری در آن‌ها ثبت‌نام کرده است را به ما می‌دهد. در نتیجه، select بیرونی، نام اساتیدی را می‌دهد که درسی را تدریس کنند که شماره‌ی آن درس متعلق به مجموعه‌ی فوق باشد. یعنی حداقل یکی از درس‌هایی را که ذاکری در آن‌ها ثبت‌نام کرده است، تدریس کرده‌اند. بنابراین کل عبارت پرس‌وجو، نام اساتیدی (غیر تکراری) را می‌دهد که حداقل یکی از درس‌هایی را تدریس می‌کنند که ذاکری در آن‌ها ثبت‌نام کرده است.

مثال: به جدول زیر دقت کنید:

Sname	Pname	Crsno
✓ ذاکری	اکبری	12
✓ ذاکری	حسینی	13
✓ ذاکری	حسینی	14
* جوادی	حسینی	15
* عسگری	سجادی	16
* تقوی	اکبری	17

جدول enroll

براساس پرس وجوی مطرح شده داریم:

```
select distinct pname
from enroll
where crsno in
      (select crsno
       from enroll
       where sname = 'kazemi')
12,13,14
```

در پرس وجوی فوق به ازای حرکت در هر سطر از جدول enroll مجموعه جلوی in بررسی می‌گردد، آیا crsno داخل این مجموعه قرار دارد یا خیر. اگر قرار داشت سطر مورد نظر از جدول enroll در خروجی نمایش داده می‌شود.

در نهایت خروجی پرس وجوی فوق به صورت زیر است:

Pname
اکبری
حسینی

توجه: سطرهای تکراری به دلیل استفاده از دستور distinct حذف گردید.

توجه: جدول enroll، جدول ثبت نام است.

۵۲- گزینه (۳) صحیح است.

راه حل اول: پرس وجوی مطرح شده در صورت سوال را در نظر بگیرید:

«نام همه مشتریانی که در همه شعب Brooklyn حساب دارند.»

توجه: کلمه همه اول، در پرس وجوی فوق زائد است و بود و نبودش تفاوتی در خروجی پرس وجوی ایجاد نمی‌کند. چون خواه ناخواه همه مشتریان حائز شرط در خروجی قرار می‌گیرند و ذکر کلمه همه در ابتدای مشتریان بی معنی است. بنابراین پرس وجوی مطرح شده در صورت سوال را به فرم زیر می‌توان نوشت:

«نام مشتریانی که در همه شعب Brooklyn حساب دارند.»

توجه: عملگر تقسیم در جبر رابطه‌ای در SQL قابل پیاده‌سازی است. عملگر تقسیم در جبر رابطه‌ای زمانی مورد استفاده قرار می‌گیرد که پرس وجوی بخواهد همه حالت‌های یک اتفاق را بررسی کند. مانند نام مشتریانی که در همه شعب شهر Brooklyn حساب باز کرده‌اند. که باز شدن حساب توسط مشتریان، اتفاق و مقسوم است و همه شعب شهر Brooklyn، حالت‌ها و مقسوم علیه است.

توجه: عملگر تقسیم در جبر رابطه‌ای به فرم زیر در SQL قابل پیاده‌سازی است.

```
SELECT distinct S.customer-name
FROM depositor as S
WHERE NOT EXISTS ((SELECT branch-name
                   FROM Branch
                   WHERE branch-city='Brooklyn')
                  EXCEPT
                  (SELECT R.branch-name
                   FROM depositor as T, account as R
                   WHERE T.account-number = R.account-number
                   And S.customer-name=T.customer-name))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«نام مشتریانی که، وجود نداشته باشد (NOT EXISTS) شعبه‌ای از شهر Brooklyn که، حساب در آن باز نشده باشد (EXCEPT). یعنی نام مشتریانی که در همه شعب شهر Brooklyn حساب باز کرده‌اند.»

راه حل دوم: برای به دست آوردن «نام مشتریانی که در همه شعب Brooklyn حساب دارند» باید برای هر مشتری (با نام S.customer-name) از جدول depositor نام تمام شعبه‌هایی که آن مشتری در آن‌ها حساب دارد را از روی جداول depositor و account به دست آورد (این مجموعه را A می‌نامیم). همچنین باید مجموعه نام تمام شعب شهر Brooklyn را از جدول branch به دست آورد (این مجموعه را B می‌نامیم). اگر A با B برابر بود (یا به عبارت دیگر، اگر B-A هیچ رکوردی نداشت) نام مشتری مربوطه باید در خروجی بیاید. دستورات مربوط به به دست آوردن مجموعه‌های A و B در زیر نشان داده شده‌اند. بنابراین باید به جای دستور except قرار داده شود تا B-A محاسبه شود. همچنین برای بررسی این که B-A هیچ رکوردی نداشته باشد باید به جای XX دستور not exists قرار داده شود.

```
select distinct S.customer-name
from depositor as s
```

مجموعه‌ی B

```
where XX ((select branch - name
           from branch
           where branch - city = 'Brooklyn'))
```

YY

مجموعه‌ی A

```
(select R.branch - name
from depositor as T, account as R
where T.account - number = R.account - number and
S.customer - name = T.customer - name)
```

)

مثال: جداول Branch, Account و Depositor، با مقادیر زیر را در نظر بگیرید:

branch-name	branch-city	assets
A	Brooklyn	700
B	Brooklyn	500
C	Brooklyn	800
D	Tehran	900

Branch شعب

assets : دارایی شعبه است.

account-number	branch-name	balance
11	A	10
11	B	20
11	C	30
11	D	40
12	A	100
12	B	90
12	C	80
13	A	50
14	D	60

Account R موجودی

customer-name	account-number
Alavi	11
Hassani	12
Taghavi	13
Javadi	14

T و S Depositor مشتریان

با توجه به پرس و جوی مطرح شده، به ازای حرکت در هر سطر از جدول S تفاضل حاصل از اجرای کامل دو طرف عملگر except محاسبه می‌گردد، اگر حاصل تفاضل تهی گردد. شرط where در پشت not exists TRUE می‌گردد، و سطر مورد نظر از جدول S، در خروجی نمایش داده می‌شود، البته با توجه به ستون S.Customer-name در دستور select، نام مشتری در خروجی نمایش داده می‌شود.

توجه: دستور S.customer-name=T.customer-name اتصال جدول حاصل از ضرب دکارتی دو جدول T و R به جدول S در select خارجی به ازای حرکت در هر سطر از جدول S است و دستور T.account-number=R.Account-number شرط و نوع اتصال ضرب دکارتی دو جدول T و R را مشخص می‌کند.

۵۳- گزینه (۱) صحیح است.

در این سؤال فرض شده است که اسامی افراد منحصر به فرد (و در نتیجه کلید) هستند. برای پاسخ به این سؤال، باید از بین تمامی کارمندان، آن‌هایی که اسمشان در جدول Mobtel نیست کنار گذاشته شوند. این دقیقاً کاری است که گزینه اول انجام می‌دهد.

پرانتر داخلی در گزینه اول، نام کارمندانی را می‌دهد که تلفن موبایل آن اپراتور را دارند. با استفاده از not in در پرانتر بیرونی، نام کارمندانی به دست می‌آید که تلفن موبایل آن اپراتور را ندارند. گزینه چهارم به دلیل استفاده از جدول fixtel به تلفن ثابت مربوط می‌شود، بنابراین نادرست است.

گزینه سوم لیست کارمندانی را می‌دهد که تلفن موبایل آن شرکت را دارند که بر عکس خواسته سؤال است. گزینه دوم، سطرهای غیر معتبر حاصل از ضرب دکارتی دو جدول را می‌دهد.

۵۴- گزینه (۴) صحیح است.

دستور select داخلی، مجموعه شناسه‌های پدران را به ما می‌دهد (یعنی مردانی که فرزند دارند). سپس دستور select بیرونی، تاریخ تولد شخصی را می‌دهد که نامش برابر x باشد و شناسه‌اش در مجموعه فوق باشد (یعنی آن شخص پدر باشد، و این یعنی این که شخص فرزندی داشته باشد)

۵۵- گزینه (۲) صحیح است.

ابتدا قسمت زیر از پرس و جو را در نظر بگیرید:

```
SELECT SUM (W.salary) AS s
FROM Employee E, WorksIn W
WHERE E.eid=W.eid
GROUP BY W.did
```

این عبارت، سطرهای جدول WorksIn را برحسب مقدار did (شماره بخش) گروه‌بندی می‌کند (یعنی برای هر بخش یک گروه در نظر می‌گیرد) و مجموع حقوق کارمندان آن بخش را بر می‌گرداند. با توجه به مقادیر جدول WorksIn، دو گروه تشکیل می‌شود (یکی برای d1 و دیگری برای d3) و مجموع salary هر گروه

جداگانه محاسبه می‌شود (8 برای d1 و 4 برای d3). پس خروجی دستور select فوق، جدولی شبیه جدول زیر است:

S
8
4

حال قسمت زیر از پرس‌وجو را در نظر بگیرید:

```
SELECT 0 AS s
FROM Department D
WHERE NOT EXISTS (SELECT *
                  FROM WorksIn W1
                  WHERE W1.did=D.did)
```

این قسمت از پرس‌وجو، به ازای هر بخشی که هیچ کارمندی در آن کار نکند، یک عدد صفر در خروجی قرار می‌دهد.

با توجه به این که سه بخش d1، d2 و d3 در جدول Department تعریف شده است، اما در جدول WorksIn برای بخش d2 هیچ کارمندی درج نشده است پس این قسمت از پرس‌وجو یک جدول تک ستونه با یک سطر حاوی مقدار صفر به صورت زیر خواهد بود.

S
0

دستور UNION ALL اجتماع دو جدول فوق را به ما می‌دهد و در نهایت، دستور SELECT AVG (R.s) میانگین اعداد حاصل از اجتماع دو جدول فوق را به ما می‌دهد که برابر است با:

$$\frac{8+4+0}{3} = 4$$

توجه: به تفاوت union و union All دقت کنید، در union دو جدول مقادیر تکراری حذف می‌گردد و در union All دو جدول مقادیر تکراری حذف نمی‌شوند.

۵۶- گزینه (۴) صحیح است.

الحاق خارجی چپ (LEFT OUTER JOIN)، همه افرادی که پدری برای آنها ثبت شده باشد یا نشده باشد را تولید می‌کند. زیرا در الحاق خارجی چپ، تمام سطرهای جدول سمت چپ در خروجی ظاهر می‌شوند، حتی سطرهایی که سطر متناظری در جدول سمت راست ندارند. بنابراین با انجام این پیوند روی C و F، تمامی افراد در خروجی می‌آیند، حتی آنهایی که سطر متناظری در جدول پدران ندارند. در ادامه، الحاق داخلی (INNER JOIN)، افرادی را انتخاب می‌کند که مادری برای آنها ثبت شده باشد. بنابراین خروجی نهایی این پرس‌وجو به صورت زیر است:

همه افرادی که مادری برای آنها ثبت شده است، مستقل از اینکه پدری برای آنها ثبت شده باشد یا نشده باشد.

سازمان سنجش آموزش کشور، گزینه چهارم را به عنوان پاسخ در کلید نهایی اعلام نموده است. عبارت «همه افراد» در گزینه چهارم، بدین معنی است که افرادی که حتی مادر برای آنها ثبت نشده است نیز می‌تواند در خروجی قرار گیرد در حالی که با توجه به شرط الحاق داخلی، افرادی انتخاب می‌شوند که

مادری برای آنها ثبت شده باشد.

بنابراین بهتر بود گزینه چهارم به صورت زیر اصلاح می‌گردید:

«همه افرادی که مادر برای آنها ثبت شده است، حتی آنهایی که پدر برای آنها ثبت نشده است.»

برای درک بهتر، جدول Person با مقادیر زیر را در نظر بگیرید:

Person ID	Name	MotherID	FatherID
1	Sara	NULL	NULL
2	Saeed	NULL	NULL
12	Hamed	1	2
1122	Reza	1	NULL

جدول Person

پرس‌وجوی مطرح شده در صورت سؤال به صورت زیر است:

```
SELECT C.Name AS Name, F.Name AS Father, M.Name AS Mother
FROM person AS C LEFT OUTER JOIN person AS F ON C.FatherID=F.PersonID
INNER JOIN person AS M ON C.MotherID=M.PersonID
```

توجه: وجود دستور AS در جلوی دستور SELECT سبب می‌شود ستون‌های C.Name, F.Name و M.Name به ترتیب به صورت Name, Father و Mother در خروجی نهایی ظاهر گردند.

توجه: وجود دستور AS در جلوی دستور FROM سبب می‌شود، سه کپی مجزا به نام‌های C, F و M از روی جدول Person ایجاد گردد.

عملگر الحاق خارجی چپ در جبر رابطه‌ای

این عملگر، مانند الحاق طبیعی، ستون‌های مشترک را فقط یکبار در خروجی قرار می‌دهد. همچنین کلیه سطرهای پیوندپذیر را در خروجی قرار می‌دهد. اما علاوه بر آن کلیه سطرهای پیوندناپذیر جدول سمت چپ را نیز در خروجی قرار می‌دهد و در این حالت برای ستون‌های غیر مشترک جدول سمت راست مقدار NULL قرار می‌دهد.

عملگر الحاق خارجی چپ در SQL

این عملگر، ستون‌های مشترک را دوبار در خروجی قرار می‌دهد. همچنین کلیه سطرهای پیوندپذیر را در خروجی قرار می‌دهد. اما علاوه بر آن کلیه سطرهای پیوندناپذیر جدول سمت چپ را نیز در خروجی قرار می‌دهد و در این حالت برای تمام ستون‌های جدول سمت راست مقدار NULL قرار می‌دهد.

توجه: به تفاوت عملگر الحاق خارجی چپ در جبر رابطه‌ای و SQL دقت کنید.

خروجی الحاق خارجی چپ مستقل از بخش‌های دیگر پرس‌وجو به صورت زیر است:

```
Select *
FROM Person AS C LEFT OUTER JOIN Person AS F ON C.FatherID = F. PersonID
```

C.PersonID	C.Name	C.MotherID	C.FatherID	F.PersonID	F.Name	F.MotherID	F.FatherID
1	sara	NULL	NULL	NULL	NULL	NULL	NULL
2	saeed	NULL	NULL	NULL	NULL	NULL	NULL
12	hamed	1	2	2	saeed	NULL	NULL
1122	reza	1	NULL	NULL	NULL	NULL	NULL

توجه: مشاهده می‌شود که همه افراد، چه برای آن‌ها پدر ثبت شده باشد و چه نشده باشد، بنا به خاصیت الحاق خارجی چپ، در خروجی ظاهر شده‌اند. الحاق داخلی خروجی فوق با جدول M، مستقل از ستون‌های جلوی دستور SELECT به صورت زیر است:

C				F				M			
PersonID	Name	MotherID	FatherID	PersonID	Name	MotherID	FatherID	PersonID	Name	MotherID	FatherID
12	hamed	1	2	2	saeed	NULL	NULL	1	sara	NULL	NULL
1122	reza	1	NULL	NULL	NULL	NULL	NULL	1	sara	NULL	NULL

در انتها، خروجی نهایی، با دخالت دستور SELECT به صورت زیر خواهد بود:

Name	Father	Mother
hamed	saeed	sara
reza	NULL	sara

مشاهده می‌شود که، «همه افرادی که مادری برای آن‌ها ثبت شده است، مستقل از اینکه پدری برای آن‌ها ثبت شده باشد یا نشده باشد» در خروجی نشان داده شده است.

توجه: اگر هر دو پیوند، INNER JOIN می‌بود، همه افرادی استخراج می‌شدند که هم پدر و هم مادر برای آن‌ها ثبت شده است.

توجه: اگر هر دو پیوند، LEFT OUTER JOIN می‌بود، همه افراد استخراج می‌شدند، مستقل از اینکه پدر و یا مادری برای آن‌ها ثبت شده باشد یا نشده باشد.

توجه: اگر پیوند اول INNER JOIN و پیوند دوم LEFT OUTER JOIN می‌بود، همه افرادی استخراج می‌شدند که پدری برای آن‌ها ثبت شده باشد، مستقل از اینکه مادری برای آن‌ها ثبت شده باشد یا نشده باشد. که در این صورت گزینه سوم درست می‌بود.

۵۷- گزینه (۱) صحیح است.

پرس‌وجوی خواسته شده باید سطرهای جدول supplier را برحسب sid و pid گروه‌بندی کند (یعنی هر قطعه تولید شده توسط هر تهیه‌کننده یک گروه در نظر گرفته شود) و سپس تعداد رکوردهای هر گروه را بشمارد و گروه‌هایی که این تعداد برای آن‌ها بیش از 10 باشد را در خروجی نشان دهند. تنها گزینه‌ای که چنین کاری انجام می‌دهد گزینه اول است.

البته گزینه اول دارای خطای نحوی است، زیرا ستون جلوی select و group by باید یکسان باشند.

۵۸- گزینه (۳) صحیح است.

راه حل اول: پرس و جوی مطرح شده در صورت سوال را در نظر بگیرید:

«مشخصات قطعاتی را بدست آورید که توسط **بعضی** از تهیه‌کنندگان تولید نمی‌شود.»

توجه: این پرس و جو به فرم زیر توسط الگوی سوم (بعضی) در SQL قابل پیاده‌سازی است.

الگوی سوم (10):

```
SELECT *
FROM Parts P
WHERE EXISTS (SELECT S.sid
              FROM Suppliers S
              WHERE NOT EXISTS (SELECT *
                                FROM Catalog C
                                WHERE S.sid = C.sid
                                AND P.pid = C.pid))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«مشخصات قطعاتی که، وجود داشته باشد (EXISTS) تهیه‌کننده‌ای که، قطعه مورد نظر را تولید نکرده باشد (NOT EXISTS). یعنی مشخصات قطعاتی که توسط بعضی تهیه‌کنندگان تولید نشده است.»

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول Parts، یک بار به طور کامل از ابتدا تا انتهای جدول Suppliers بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول Suppliers وجود دارد که شرط اتصال $S.sid = C.sid$ AND $P.pid = C.pid$ موجود در select داخلی را برقرار نکند، اگر وجود داشته باشد، سطر مورد نظر از جدول Parts در خروجی نمایش داده می‌شود.

توجه: شرط لازم برای تبعیت از الگوی سوم وجود **EXISTS** اول و **NOT EXISTS** دوم است و شرط کافی برای تبعیت از الگوی سوم وجود **شرط اتصال** متناسب و سازگار است.

راه حل دوم: جداول زیر را در نظر بگیرید:

sid	sname	sid	pid	pid	pname
s1	sn1	s1	p1	p1	pn1
s2	sn2	s1	p2	p2	pn2
		s2	p1		

تهیه‌کنندگان *suppliers*

تولید *catalog*

قطعات *parts*

S

C

P

select *

form parts p

where(A).....(select s.sid

from suppliers s

where(B)....(select *

from catalog c

where s.sid=c.sid AND p.pid=c.pid))

پرس و جوی مطرح شده در صورت سؤال به صورت زیر است:

«مشخصات قطعاتی را بدست آورید که توسط بعضی از تهیه‌کنندگان تولید نمی‌شود»

به عبارت دیگر می‌توان گفت:

«مشخصات قطعه‌ای در خروجی ظاهر گردد که وجود دارد، تهیه‌کننده‌ای که آن قطعه را تولید نکرده باشد.»

جدول قطعات	exists	جدول تهیه‌کنندگان	جدول not exists
			تولید

وجود داشتن تهیه کننده با بکارگیری عملگر exists به جای A کنترل می شود و تولید نشدن آن قطعه توسط یک تهیه کننده نیز که معادل با وجود نداشتن رکوردی با مقادیر p.pid و s.sid مربوطه در جدول تولید است، با بکارگیری عملگر not exists به جای B کنترل می شود. بنابراین پرس وجوی فوق به صورت زیر خواهد بود:

```
select *
from parts p
where exists (select s.sid
              from suppliers s
              where not exists (select *
                                from catalog c
                                where s.sid=c.sid AND p.pid=c.pid))
```

برای عملکرد دقیق پرس وجوی فوق ادامه راه حل را دنبال کنید:

الگوریتم فوق به این شکل حرکت می کند، به شرح الگوریتم دقت کنید:

ابتدا سطر اول از جدول قطعات بررسی می شود تا بررسی کامل در همین سطر توقف کنید، اگر جلوی exists غیر تهی باشد، یعنی دارای مقدار باشد، نام قطعه مورد نظر از جدول قطعات در خروجی به عنوان قطعه ای که توسط برخی تهیه کنندگان تولید نشده است، نمایش داده می شود. اما مسأله این است که چه وقت جلوی exists غیر تهی می شود. پاسخ این است که شماره تهیه کننده ای (s.sid) از جمله select s.sid form suppliers s where where در همان جمله برقرار باشد. اما چه وقت شرط where برقرار است، پاسخی که پرس وجوی داخل not exists تهی باشد، یعنی قطعه مورد نظر، توسط تهیه کننده مورد بررسی، تولید نشده باشد. در یک بیان ساده، ابتدا سطر اول از جدول قطعات را انتخاب می کنیم. مثلاً قطعه p1، سپس برای قطعه p1 از جدول قطعات بررسی می کنیم، که آیا تهیه کنندگان موجود در جدول تهیه کنندگان قطعه p1 را تولید کرده اند یا خیر.

بنابراین برای قطعه مورد نظر، توسط دستور select جلوی exists، بر روی جدول تهیه کنندگان از ابتدا تا انتهای جدول به ازای قطعه مورد نظر، حرکت می کنیم، اما به ازای حرکت در هر سطر از جدول تهیه کنندگان، توسط دستور select جلوی not exists بررسی می گردد که تهیه کننده مورد بررسی در جدول تهیه کنندگان، در جدول تولید، به ازای قطعه مورد نظر، تولید داشته است، یا خیر. اگر تولید داشته باشد، توسط دستور select جلوی not exists نمایش داده می شود. و توسط s.sid داخل دستور select جلوی exists نمایش داده نمی شود به دلیل FALSE بودن شرط دستور where جلوی exists ناشی از FALSE بودن not exists به دلیل تهی نبودن جلوی not exists.

حال اگر مطابق روال فوق به ازای حداقل یکی از تهیه کنندگان موجود در جدول تهیه کنندگان، به ازای قطعه مورد نظر، حداقل یکی از تهیه کنندگان، قطعه مورد نظر را تولید نکرده باشد. باعث می شود تا s.sid داخل دستور select جلوی exists تهی نگردد و باعث شود، شرط دستور where در پشت exists، TRUE گردد، بنابراین قطعه مورد نظر موجود در جدول قطعات، به عنوان قطعه ای که حداقل یکی از تهیه کنندگان آن را تولید نکرده است، در خروجی نمایش داده می شود. پرس وجوی مطرح شده را مجدداً در نظر بگیرید:

```

select *
from parts p
where exists (select s.sid
              from suppliers s
              where not exists (select *
                                from catalog c
                                where s.sid=c.sid AND  $\begin{matrix} p.pid = c.pid \\ p_1 \quad p_1 \end{matrix}$  ) )

```

به کادر مستطیل شکل مشخص شده در پرس و جوی فوق توجه کنید، به ازای حرکت در هر سطر از جدول قطعات جمله $p.pid = c.pid$ تغییر می‌کند. هم اکنون در سطر اول جدول قطعات هستیم. پس pid برابر p_1 است. به قطعه که موجود در جلوی $exists$ ، دقت کنید:

خروجی
↑
exists (select s.sid
 form suppliers s
 where ...

ابتدا سطر اول از جدول تهیه کنندگان یعنی s_1 بررسی می‌شود. همچنین به قطعه که موجود در جلوی $not exists$ ، دقت کنید:

خروجی
↑
not exists (select *
 from catalog
 where $\frac{s_1 \quad s_1}{TRUE}$ AND $\frac{p_1 \quad p_1}{TRUE}$
 $\frac{\quad \quad}{TRUE}$

با شرط فوق، بررسی می‌شود که قطعه p_1 ، توسط تهیه کننده s_1 تولید شده است یا خیر. شرط برقرار است، زیرا تهیه کننده s_1 ، قطعه p_1 را تولید کرده است، بنابراین داریم:

(s_1, p_1)
↑
not exists (select *
 form catalog c

چون جلوی $not exists$ تهی نیست، بنابراین شرط $where$ موجود پشت $not exists$ FALSE می‌گردد، بنابراین سطر اول جدول تهیه کنندگان یعنی تهیه کننده s_1 در خروجی دستور $select$ جلوی $exists$ ظاهر نمی‌شود. چون قطعه p_1 توسط تهیه کننده s_1 تولید شده است. در ادامه سطر دوم از جدول تهیه کنندگان یعنی s_2 ، همچنان برای قطعه p_1 بررسی می‌شود. با شرط زیر بررسی می‌شود که قطعه p_1 ، توسط تهیه کننده s_2 تولید شده است یا خیر.

where $\frac{s_2 \quad s_2}{TRUE}$ AND $\frac{p_1 \quad p_1}{TRUE}$
 $\frac{\quad \quad}{TRUE}$

شرط برقرار است، زیرا تهیه کننده s2، قطعه p1 را تولید کرده است، بنابراین داریم:

```
(s2,p1)
  ↑
not exists (select *
            form catalog c
```

چون جلوی not exists تهی نیست، بنابراین شرط where موجود پشت not exists می‌گردد، بنابراین سطر دوم جدول تهیه کنندگان یعنی تهیه کننده s2 در خروجی دستور select جلوی exists ظاهر نمی‌شود.

چون قطعه p1 توسط تهیه کننده s2 تولید شده است. نتیجه اینکه، قطعه p1 توسط همه تولید کنندگان، تولید شده است. مطابق پرس و جوی مطرح شده داریم:

```
select *
form parts p
where exists (select s.sid
              from suppliers s
              where ...
```

در نهایت مقدار s.sid در خط فوق در جلوی exists برای قطعه p1 تهی می‌گردد، چون به دلیل شرایط جلوی not exists مشاهده نمودید که هیچ یک از سطرهای جدول تهیه کنندگان در خروجی دستور select جلوی exists ظاهر نشدند. چون قطعه p1 توسط تهیه کنندگان s1 و s2 تولید شده‌اند. در یک بیان ساده، اگر تهیه کننده‌ای، قطعه مورد نظر را تولید کرده باشد، در خروجی select s.sid در خط فوق، قرار نمی‌گیرد. تولیدات تهیه کننده توسط دستور select در جلوی not exists بررسی می‌گردد.

جلوی exists، برای سطر اول جدول قطعات، یعنی قطعه p1، تهی است، زیرا در خروجی select s.sid جلوی exists، تهیه کنندگانی که قطعه مورد نظر را تولید نکرده باشند، خارج می‌گردند، از آنجا که پس از بررسی تولیدات همه تهیه کنندگان برای قطعه p1 توسط دستور select در جلوی not exists مشخص شد که قطعه p1 توسط همه تهیه کنندگان تولید شده است، بنابراین خروجی select s.sid exists، تهی خواهد بود، چون همه تهیه کنندگان قطعه p1 را تولید کرده‌اند. پس شرط where در پشت exists، FALSE است، پس سطر اول جدول قطعات انتخاب نمی‌گردد و به تبع در خروجی نیز نمایش داده نمی‌شود، زیرا قطعه p1 توسط همه تهیه کنندگان، تولید شده است.

حال سطر دوم از جدول قطعات بررسی می‌شود، تا بررسی کامل در همین سطر توقف کنید. پس pid برابر p2 است.

به قطعه کد موجود در جلوی exists، دقت کنید:

```
خروجی
  ↑
exists (select s.sid
        form suppliers s
        where ...
```

ابتدا سطر اول از جدول تهیه کنندگان یعنی s1 بررسی می‌شود. همچنین به قطعه کد موجود در جلوی not exists، دقت کنید:

خروجی
↑

```
not exists (select *
            from catalog
            where s.sid = c.sid AND p.pid = c.pid
             $\frac{\frac{s_1}{TRUE} \quad \frac{s_1}{TRUE}}{TRUE} \quad \frac{\frac{p_2}{TRUE} \quad \frac{p_2}{TRUE}}{TRUE}$ )
```

با شرط فوق، بررسی می‌شود که قطعه p2، توسط تهیه کننده s1 تولید شده است یا خیر. شرط برقرار است، زیرا تهیه کننده s1، قطعه p2 را تولید کرده است، بنابراین داریم:

(s1, p2)
↑

```
not exists (select *
            form catalog c
```

چون جلوی not exists تهی نیست، بنابراین شرط where موجود پشت not exists FALSE می‌گردد، بنابراین سطر اول جدول تهیه کنندگان یعنی تهیه کننده s1 در خروجی دستور select جلوی exists ظاهر نمی‌شود. چون قطعه p2 توسط تهیه کننده s1 تولید شده است.

در ادامه سطر دوم از جدول تهیه کنندگان یعنی s2، همچنان برای قطعه p2 بررسی می‌شود. با شرط زیر بررسی می‌شود که قطعه p2، توسط تهیه کننده s2 تولید شده است. یا خیر.

```
where s.sid  $\neq$  c.sid AND p.pid = c.pid
 $\frac{\frac{s_2}{FALSE} \quad \times}{FALSE} \quad \frac{\frac{p_2}{FALSE} \quad \times}{FALSE}$ 
FALSE
```

شرط برقرار نیست، زیرا تهیه کننده s2، قطعه p2 را تولید نکرده است، بنابراین داریم:

تهی
↑

```
not exists (select *
            form catalog c
```

چون جلوی not exists تهی است، بنابراین شرط where موجود پشت not exists TRUE می‌گردد، بنابراین سطر دوم جدول تهیه کنندگان یعنی تهیه کننده s2، در خروجی دستور select جلوی exists ظاهر می‌شود. چون قطعه p2 توسط تهیه کننده s2 تولید نشده است.

نتیجه اینکه، قطعه p2 توسط همه تولید کنندگان، تولید نشده است، به عبارت دیگر قطعه p2، توسط بعضی از تهیه کنندگان تولید نشده است.

مطابق پرس و جوی مطرح شده داریم:

```
select *
form parts p
where exists (select s.sid
              from suppliers s
              where ...
```

در نهایت مقدار s.sid در خط فوق در جلوی exists برای قطعه p2 غیرتهی می‌گردد و برابر مقدار s2، چون به دلیل شرایط جلوی not exists مشاهده نمودید که سطر دوم از جدول تهیه‌کنندگان در خروجی دستور select جلوی not exists ظاهر شد، چون قطعه p2 توسط تهیه‌کننده s2 تولید نشده بود. جلوی exists، برای سطر دوم جدول قطعات، یعنی قطعه p2، غیرتهی است، زیرا در خروجی select s.sid جلوی exists، تهیه‌کنندگانی که قطعه مورد نظر را تولید نکرده باشند، خارج می‌گردند، از آنجا که پس از بررسی تولیدات همه تهیه‌کنندگان برای قطعه p2 توسط دستور select در جلوی not exists مشخص شد که قطعه p2 توسط برخی تهیه‌کنندگان تولید نشده است، بنابراین خروجی select s.sid جلوی exists، غیرتهی خواهد بود، چون برخی تهیه‌کنندگان قطعه p2 را تولید نکرده‌اند. پس شرط where در پشت exists، TRUE است، پس سطر دوم جدول قطعات انتخاب می‌گردد، زیرا قطعه p2 توسط تهیه‌کننده s2 تولید نشده است.

۵۹- گزینه (۲) صحیح است.

جدول زیر را در نظر بگیرید:

S#	Sname	Status	City
S1	Sn1	1	C1
S2	Sn2	2	C2
S3	Sn3	3	C3
S4	Sn4	4	C4

جدول تهیه‌کنندگان S

S#	P#	J#	QTY
S1	P1	J1	10
S1	P2	J1	20
S1	P1	J2	30
S1	P2	J2	40
S2	P1	J1	50
S2	P1	J2	20
S2	P2	J2	50
S3	P1	J2	70
S3	P3	J2	80

جدول SPJ

J#	Jname	City
J1	Jn1	C1
J2	Jn2	C2

جدول پروژه J

P#	Pname	Weight	City
P1	Pn1	100	C1
P2	Pn2	200	C2
P3	Pn3	300	C3

جدول قطعه P

توجه: زیر پرس‌وجوی مطرح شده در جلوی دستور All، یک Normal Subquery است. بنابراین ابتدا زیر پرس‌وجوی داخلی یک بار و برای همیشه اجرا می‌گردد، سپس پرس‌وجوی خارجی به ازای حرکت در هر یک از گروه‌های خود (به دلیل اعمال گروه‌بندی) از مقادیر زیر پرس‌وجوی داخلی استفاده می‌کند. بنابراین در ادامه برای زیر پرس‌وجوی داخلی داریم:

```
(select SUM(QTY)
form SPJ
where SPJ.J#='J2'
group by S#)
```

روال کار بدین صورت است که ابتدا سراغ جدول SPJ می‌رویم، سپس شرط 'J2'=SPJ.J# بر روی جدول SPJ اعمال می‌گردد، در ادامه دستور Group by بر روی ستون S# اعمال می‌گردد، در انتها دستور SUM(QTY) برای هر گروه به طور مستقل محاسبه می‌گردد و در خروجی پرس‌وجو قرار می‌گیرد. بنابراین در ادامه داریم:

S#	P#	J#	QTY
S ₁	P ₁	J ₂	30
S ₁	P ₂	J ₂	40
S ₂	P ₁	J ₂	20
S ₂	P ₂	J ₂	50
S ₃	P ₁	J ₂	70
S ₃	P ₃	J ₂	80

from SPJ
where SPJ.J#='J2' ⇒

حال در ادامه براساس زیر پرس‌وجوی مطرح شده، ابتدا دستور Group by بر روی ستون S# اعمال می‌گردد و سپس دستور SUM(QTY) بر روی ستون QTY در هر گروه به طور مستقل انجام می‌گردد. و خروجی‌های آن جهت استفاده پرس‌وجوی خارجی ذخیره می‌گردد، بنابراین در ادامه داریم:

(S ₁)	(S ₂)	(S ₃)
P ₁ J ₂ 30	P ₁ J ₂ 20	P ₁ J ₂ 70
P ₂ J ₂ 40	P ₂ J ₂ 50	P ₃ J ₂ 80
گروه اول	گروه دوم	گروه سوم
SUM(QTY) = 70	SUM(QTY) = 70	SUM(QTY) = 150

در واقع زیر پرس‌وجوی داخلی مطرح شده در جلوی دستور All، برای هر تولیدکننده حاضر در پروژه J₂، تعداد کل قطعات تولید شده (از تمام انواع قطعات) توسط آن تولیدکننده در پروژه J₂ را می‌دهد. بنابراین داریم:

... <= All(70,70,150)

حال در ادامه، پرس‌وجوی خارجی به ازای حرکت در هر یک از گروه‌های خود، از مقادیر زیر پرس‌وجوی داخلی استفاده می‌کند.

توجه: دستور having، حرکت بر روی گروه‌ها را میسر می‌کند، همانطور که دستور where، حرکت بر روی سطرها را میسر می‌کند.

الحاق طبیعی دو جدول S و SPJ، تهیه کنندگانی که حداقل یک قطعه برای یک پروژه را تولید کرده‌اند، استخراج می‌کند، همچنین پس از اعمال شرط 'J2'=SPJ.J# داریم:

S#	Sname	Status	City	P#	J#	QTY
S ₁	Sn1	1	C1	P ₁	J ₂	30
S ₁	sn1	1	C1	P ₂	J ₂	40
S ₂	Sn2	2	C2	P ₁	J ₂	20
S ₂	Sn2	2	C2	P ₂	J ₂	50
S ₃	Sn3	3	C3	P ₁	J ₂	70
S ₃	Sn3	3	C3	P ₃	J ₂	80

S natural join SPJ
where SPJ.J#='J2' ⇒

در ادامه برای سادگی از ستون‌های status و city صرف نظر می‌کنیم، بنابراین داریم:

S#	Sname	P#	J#	QTY
S1	Sn1	P1	J2	30
S1	Sn1	P2	J2	40
S2	Sn2	P1	J2	20
S2	Sn2	P2	J2	50
S3	Sn3	P1	J2	70
S3	Sn3	P3	J2	80

حال، براساس پرس و جوی مورد نظر، دستور Sname و S# by Group اجرا می‌گردد، بنابراین داریم:
(S# , Sname)

(S1,Sn1)	(S2,Sn2)	(S3,Sn3)
P1 J2 30	P1 J2 20	P1 J2 70
<u>P2 J2 40</u>	<u>P2 J2 50</u>	<u>P3 J2 80</u>
گروه اول	گروه دوم	گروه سوم

حال در ادامه، براساس پرس و جوی خارجی، دستور SUM(QTY) having بر روی هر گروه به طور جداگانه اعمال می‌گردد:

توجه: دستور having بر روی گروه‌ها، اعمال می‌گردد.

(S1,Sn1)	(S2,Sn2)	(S3,Sn3)
P1 J2 30	P1 J2 20	P1 J2 70
<u>P2 J2 40</u>	<u>P2 J2 50</u>	<u>P3 J2 80</u>
گروه اول	گروه دوم	گروه سوم
SUM(QTY) = 70	SUM(QTY) = 70	SUM(QTY) = 150

در واقع، پرس و جوی خارجی، برای هر تولید کننده‌ای که در پروژه J2 شرکت داشته است، اگر تعداد کل قطعات تولید شده (از تمام انواع قطعات) توسط وی در پروژه J2 کمتر از تعداد کل قطعات تولید شده (از تمام انواع قطعات) توسط تک تک تولید کنندگان در پروژه J2 باشد، نام آن تولید کننده را در خروجی می‌آورد.

بنابراین در ادامه داریم:

$70 \leq \text{All}(70, 70, 150)$

گام اول: بررسی گروه اول.

پاسخ: SUM(QTY) با مقدار 70 در گروه اول از همه مقادیر (70,70,150) کوچکتر یا مساوی است.

بنابراین تهیه کننده Sn1، به عنوان تولید کننده‌ای که کمترین تعداد قطعه (از تمام انواع قطعات) را برای پروژه J2 تولید کرده است، در خروجی قرار می‌گیرد.

$70 \leq \text{All}(70, 70, 150)$

گام دوم: بررسی گروه دوم.

پاسخ: SUM(QTY) با مقدار 70 در گروه دوم از همه مقادیر (70,70,150) کوچکتر یا مساوی است.

بنابراین تهیه کننده Sn2، به عنوان تولید کننده‌ای که کمترین تعداد قطعه (از تمام انواع قطعات) را برای پروژه J2 تولید کرده است، در خروجی قرار می‌گیرد.

گام سوم: بررسی گروه سوم.
 $150 \leq \text{All}(70, 70, 150)$
 پاسخ: SUM(QTY) با مقدار 150 در گروه سوم از همه مقادیر (70,70,150) کوچکتر یا مساوی نیست.
 بنابراین تهیه کننده Sn3، به عنوان تولید کننده‌ای که کمترین تعداد قطعه (از تمام قطعات) را برای پروژه J2 تولید کرده باشد، در خروجی قرار نمی‌گیرد.
 بنابراین پرس‌وجوی مطرح شده در صورت سؤال، نام تولید کنندگانی که کمترین تعداد قطعه (از تمام انواع قطعات) را برای پروژه کد J2 تولید کرده‌اند را به صورت زیر در خروجی نمایش می‌دهد.

Sname

Sn1

Sn2

توجه: البته پرس‌وجوی مطرح شده دارای کمی خطای نحوی است، ستون‌های جلوی دستور select و group by در پرس‌وجوی خارجی باید یکسان باشند، یا اینکه ستون موجود در جلوی select که در مقابل group by قرار ندارد، در جلوی select با توابع آماری همراه گردد. اگر جلوی دستور select، ستون S# نیز علاوه بر ستون Sname اضافه گردد، خطای نحوی مرتفع می‌گردد.

۶۰- گزینه (۳) صحیح است.

راه حل اول:

جداول زیر را در نظر بگیرید:

<u>mem-no</u>	name	age	<u>mem-no</u>	<u>isbn</u>	date	<u>isbn</u>	...	Publisher
m1	mn1	21	m1	is1	11	is1		McGraw-Hill
m2	mn2	22	m1	is2	12	is2		McGraw-Hill
m3	mn3	23	m2	is1	13	is3		Penguin
Member A			m2	is2	14	is4		Penguin
			m2	is3	15			Book
			Borrowed B					

مطابق پرس و جوی مطرح شده در صورت سوال داریم:

«نام اعضای که همه کتاب‌های منتشرشده توسط McGraw-Hill را به امانت برده‌اند.»

پرس و جوی فوق توسط عملگر تقسیم در جبر رابطه‌ای قابل پیاده‌سازی است، اما در SQL می‌بایست عملگر تقسیم شبیه‌سازی گردد.

در ادامه، این پرس و جو را به سه فرم زیر می‌نویسیم:

فرم اول:

```
SELECT A.name
FROM Member A, Borrowed B
WHERE A.mem-no=B.mem-no
GROUP BY A.mem-no
HAVING COUNT (B.isbn) = (SELECT COUNT (isbn)
FROM Book
WHERE Publisher=McGraw-Hill)
```

فرم دوم:

```

SELECT A.name
FROM Member A
WHERE NOT EXISTS ((SELECT isbn
                    FROM Book
                    WHERE publisher = McGraw-Hill)
                   EXCEPT
                   (SELECT isbn
                    FROM Borrowed B
                    WHERE A.mem-no = B.mem-no))

```

کارکرد قطعه پرس و جوی زیر از پرس و جوی فوق به صورت زیر است:

```

SELECT isbn
FROM Book
WHERE publisher = McGraw-Hill

```

«شماره‌ی همه‌ی کتاب‌های منتشرشده توسط انتشارات McGraw-Hill»
همچنین کارکرد قطعه پرس و جوی زیر از پرس و جوی فوق به صورت زیر است:

```

SELECT isbn
FROM Borrowed B
WHERE A.mem-no = B.mem-no

```

«شماره کتاب‌های به امانت برده شده توسط یک عضو»

در پرس و جوی فوق، در پراتز داخلی کل کتاب‌های چاپ شده، توسط انتشارات McGraw-Hill منهای کلیه کتاب‌های به امانت برده شده توسط یک عضو می‌شود و در صورتی که حاصل این تفاضل تهی باشد نام عضو مورد جستجو در خروجی ظاهر می‌شود.

بنابراین پرس و جوی فوق، نام اعضای که همه کتاب‌های منتشرشده توسط انتشارات McGraw-Hill را به امانت برده‌اند، نشان می‌دهد.

توجه: فرم دوم مطرح شده، مطابق گزینه سوم می‌باشد، بنابراین گزینه سوم پاسخ سوال خواهد بود.

حال به تشریح نحوه‌ی کارکرد پرس و جو می‌پردازیم:

توسط دستور where در فرم زیر:

```

SELECT A.name
FROM Member A
WHERE NOT EXISTS (...)

```

برای هر سطر از جدول Member شرط جلوی not exists که حاصل یک تفاضل توسط عملگر except می‌باشد، محاسبه می‌گردد، اگر تهی بود، شرط جلوی where که همان not exists است، TRUE می‌گردد و سطر مورد نظر از جدول Member انتخاب می‌گردد و این رویه برای تک تک سطرهای جدول Member، تا به انتهای جدول Member ادامه پیدا می‌کند. به بیان دیگر این پرس و جو نام اعضای را می‌دهد که پراتز مقابل not exists برای آن‌ها تهی است. این پراتز هنگامی تهی می‌شود که حاصل تفاضل بیان شده در این پراتز تهی شود. حاصل این تفاضل در صورتی تهی می‌شود که تعداد کل کتاب‌های منتشرشده توسط انتشارات McGraw-Hill برابر تعداد کتاب‌هایی باشد که آن عضو تاکنون به امانت برده است. به عبارت دیگر این پرس و جو نام اعضای را می‌دهد که تاکنون تمام کتاب‌های منتشرشده توسط انتشارات

McGraw-Hill را به امانت برده‌اند.

توجه: همان‌طور که مشاهده می‌کنید، عضو m1 و m2، همه‌ی کتاب‌های منتشرشده توسط انتشارات McGraw-Hill را به امانت برده‌اند.

به بیان دیگر عبارت موجود در جلوی دستور not exists که به صورت زیر است:

```
SELECT isbn
FROM Book
WHERE publisher = McGraw-Hill
```

شماره همه‌ی کتاب‌های منتشرشده توسط انتشارات McGraw-Hill موجود در جدول Book را نمایش می‌دهد.

با توجه به جداول فوق داریم:

```
isbn
is1
is2
```

هم‌چنین عبارت بعد از دستور Except یعنی:

```
SELECT isbn
FROM Borrowed B
WHERE A.mem-no = B.mem-no
```

توسط دستور $A.mem - no = B.mem - no$ به محیط خارج یعنی جدول A یا Member متصل می‌گردد. حال به ازای حرکت در هر سطر از جدول Member، یک بار به طور کامل سطرهای جدول Borrowed از ابتدا تا انتها با توجه به شرط اتصال بررسی می‌گردد. مطابق شکل زیر:

ابتدا برای سطر اول از جدول Member با توجه به شرط اتصال $A.mem - no = B.mem - no$ داریم:

m1			m1			→	isbn	
name	mem-no	...	mem-no	isbn	...		is1	is2
mn1	m1		m1	is1		→	is1	
mn2	m2		m1	is2			is2	
mn3	m3		m2	is1				
Member A			m2	is2				
			m2	is3				
			Borrowed B					

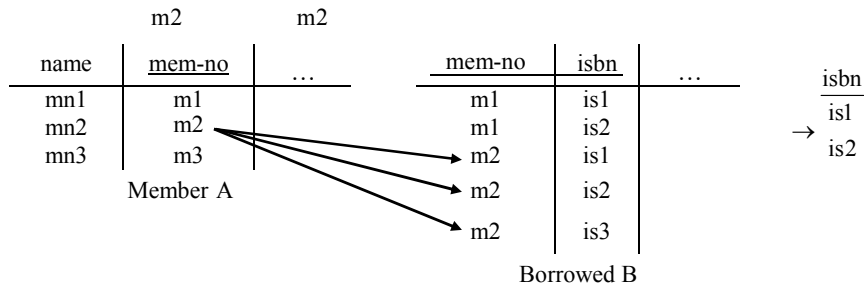
خروجی حاصل از عملگر تفاضل جلوی not exists به ازای سطر اول از جدول Member به صورت زیر است:

```
isbn  isbn  خروجی
is1  _  is1  =  تهی
is2  is2
```

جلوی not exists برابر تهی گردید، بنابراین شرط where در پشت not exists TRUE می‌گردد. بنابراین سطر اول از جدول Member به شکل زیر در خروجی نمایش داده می‌شود.

name
mn1

حال برای سطر دوم از جدول Member با توجه به شرط اتصال $A.\text{mem-no} = B.\text{mem-no}$ داریم:



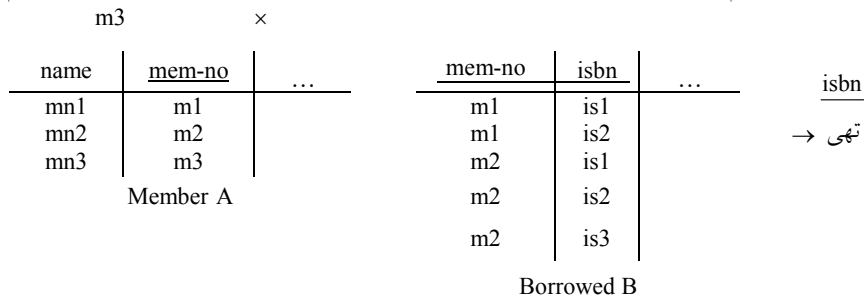
خروجی حاصل از عملگر تفاضل جلوی not exists به ازای سطر دوم از جدول Member به صورت زیر است:

isbn isbn خروجی
is1 _ is1 = تهی
is2 is2
is3

جلوی not exists برابر تهی گردید، بنابراین شرط where در پشت not exists TRUE می‌گردد. بنابراین سطر دوم از جدول Member به شکل زیر در خروجی نمایش داده می‌شود.

name
mn2

در ادامه برای سطر سوم از جدول Member با توجه به شرط اتصال $A.\text{mem-no} = B.\text{mem-no}$ داریم:



خروجی حاصل از عملگر تفاضل جلوی not exists به ازای سطر سوم از جدول Member به صورت زیر است:

isbn isbn خروجی
is1 _ تهی = is1
is2 is2

جلوی not exists برابر غیر تهی گردید، بنابراین شرط where در پشت not exist FALSE می‌گردد،

بنابراین سطر سوم از جدول Member در خروجی نمایش داده نخواهد شد.
بنابراین خروجی نهایی برای «نام اعضایی که همه کتاب‌های منتشرشده توسط انتشارات McGraw-Hill را به امانت برده‌اند.» به صورت زیر است:

```
name
mn1
mn2
```

فرم سوم:

```
SELECT A.name
FROM Member A
WHERE NOT EXISTS (SELECT isbn
                   FROM Book
                   WHERE publisher = McGraw-Hill
                   And
                   NOT EXISTS (SELECT isbn
                               FROM Borrowed B
                               WHERE A.mem-no = B.mem-no
                               And Book.isbn = B.isbn))
```

بررسی گزینه چهارم:

```
SELECT A.name
FROM Member A
WHERE NOT EXISTS ((SELECT isbn
                   FROM Borrowed B
                   WHERE A.mem-no = B.mem-no)
                  EXCEPT
                  (SELECT isbn
                   FROM Book
                   WHERE publisher = McGraw-Hill))
```

کارکرد قطعه پرس و جوی زیر از پرس و جوی فوق به صورت زیر است:

```
SELECT isbn
FROM Borrowed B
WHERE A.mem-no = B.mem-no
```

«شماره کتاب‌های به امانت برده شده توسط یک عضو»

همچنین کارکرد قطعه پرس و جوی زیر از پرس و جوی فوق به صورت زیر است:

```
SELECT isbn
FROM Book
WHERE publisher = McGraw-Hill
```

«شماره‌ی همه‌ی کتاب‌های منتشرشده توسط انتشارات McGraw-Hill»

در پرس و جوی فوق، در پراتز داخلی کل کتاب‌های به امانت برده شده توسط یک عضو منهای کلیه کتاب‌های چاپ شده، توسط انتشارات McGraw-Hill می‌شود و در صورتی که حاصل این تفاضل تهی باشد نام عضو مورد جستجو در خروجی ظاهر می‌شود.

بنابراین پرس و جوی فوق، نام اعضایی که فقط کتاب‌های منتشرشده توسط انتشارات

McGraw-Hill را به امانت برده‌اند، نشان می‌دهد. یعنی نباید کتاب‌های منتشر شده توسط انتشارات‌های دیگر را به امانت برده باشند، اینکه همه کتاب‌های منتشر شده توسط انتشارات McGraw-Hill را به امانت برده‌اند یا نبرده‌اند مهم نیست، اما این مهم است آنچه به امانت برده شده است، حتماً و فقط و فقط از انتشارات McGraw-Hill باشد.

بررسی گزینه‌های اول و دوم:

مطابق آنچه گفتیم، استفاده از دستورات `not exists` و `except` یکی از فرم‌های شبیه‌سازی عملگر تقسیم در SQL است. نکته حائز اهمیت این است که، در هیچ یک از فرم‌های شبیه‌سازی عملگر تقسیم، دستور `not exists` به تنهایی مورد استفاده قرار نمی‌گیرد. بنابراین گزینه‌های اول و دوم مطابق پرس و جوی مطرح شده در صورت سوال نیستند. گزینه‌های اول و دوم صرفاً ضرب دکارتی دو جدول `Book` و `Borrowed` را انجام می‌دهند و براساس شروط مطرح شده سطرهایی را انتخاب می‌کنند، و بسته به شرایط جلوی `not exists` سطرهایی از جدول `Member` انتخاب می‌گردد، خروجی استخراج شده از پرس و جویهای مطرح شده در گزینه‌های اول و دوم معتبر نیستند.

راه حل دوم:

پرس و جوی مطرح شده در صورت سوال را در نظر بگیرید:

«نام اعضای که همه کتاب‌های منتشرشده توسط McGraw-Hill را به امانت برده‌اند.»

توجه: عملگر تقسیم در جبر رابطه‌ای در SQL قابل پیاده‌سازی است. عملگر تقسیم در جبر رابطه‌ای زمانی مورد استفاده قرار می‌گیرد که پرس و جو بخواهد همه حالت‌های یک اتفاق را بررسی کند. مانند نام اعضای که همه کتاب‌های منتشر شده توسط McGraw-Hill را به امانت برده‌اند. که به امانت بردن کتاب‌ها توسط اعضا، اتفاق و مقسوم است و همه کتاب‌های منتشر شده توسط McGraw-Hill، حالت‌ها و مقسوم علیه است.

توجه: عملگر تقسیم در جبر رابطه‌ای به فرم زیر در SQL قابل پیاده‌سازی است.

```
SELECT A.name
FROM Member A
WHERE NOT EXISTS ((SELECT isbn
                    FROM Book
                    WHERE publisher=McGraw-Hill)
EXCEPT
(SELECT isbn
FROM Borrowed B
WHERE A.mem-no=B.mem-no))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«نام اعضای که، وجود نداشته باشد (`NOT EXISTS`) کتابی از McGraw-Hill که، به امانت برده نشده باشد (`EXCEPT`). یعنی نام اعضای که همه کتاب‌های منتشر شده توسط McGraw-Hill را به امانت برده‌اند.»

۶۱- گزینه (۴) صحیح است.

راه حل اول:

جداول زیر را در نظر بگیرید:

S#	Sname	City	S#	P#	QTY	P#	Pname	Color
S1	Sn1	C1	S1	P1	10	P1	Pn1	Red
S2	Sn2	C2	S1	P2	20	P2	Pn2	Blue
S3	Sn3	C2	S2	P1	30			

جدول S

جدول SP

مطابق پرس و جوی مطرح شده در صورت سوال داریم:

«شماره تولیدکنندگانی که همه قطعات را تولید کرده‌اند.»

پرس و جوی فوق توسط عملگر تقسیم در جبر رابطه‌ای قابل پیاده‌سازی است، اما در SQL می‌بایست عملگر تقسیم شبیه‌سازی گردد.

در ادامه، این پرس و جو را به سه فرم زیر شبیه‌سازی می‌کنیم:

فرم اول:

```
SELECT S#
FROM SP
GROUP BY S#
HAVING COUNT (P#) = (SELECT COUNT (P#)
FROM P)
```

پرس و جوی فوق، شماره تولیدکنندگانی را می‌دهد که انواع قطعاتی که تولید کرده‌اند برابر تعداد کل قطعات موجود در جدول قطعه باشد.

در پرس و جوی فوق ابتدا توسط زیر پرس و جوی زیر تعداد کل قطعات موجود در جدول قطعه شمارش و محاسبه می‌گردد.

```
SELECT COUNT (P#)
FROM P
```

که حاصل زیر پرس و جوی فوق برابر مقدار 2 است، که این مقدار برابر تعداد کل قطعات موجود در جدول P است، که شامل قطعات P1 و P2 است.

در ادامه، پرس و جوی اصلی توسط دستور GROUP BY S# براساس S# به صورت زیر گروه‌بندی می‌شود و در ادامه براساس COUNT (P#)، تعداد انواع قطعات تولیدشده در هر گروه محاسبه می‌گردد.

S1	S2
P1 10	P1 10
P2 20	گروه دوم
گروه اول	
COUNT (P#) = 2	COUNT (P#) = 1

سپس توسط دستور HAVING COUNT (P#)=2 به عنوان شرط انتخاب گروه، گروه‌هایی که COUNT (P#) آن برابر مقدار 2 است انتخاب می‌گردند. بنابراین مشاهده می‌شود که شرط مطرح شده فقط

برای گروه اول برقرار است، پس گروه اول انتخاب می‌گردد. به صورت زیر:

$\frac{S\#}{SI}$

توجه: دستور GROUP BY، سرگروه‌ها را، راهی خروجی می‌کند.

توجه: در گزینه دوم اگر دستور FROM SP موجود در SELECT دوم به FROM P تغییر کند، آنگاه فرم اول ایجاد می‌شود.

فرم دوم:

```
SELECT S#
FROM S
WHERE NOT EXISTS ((SELECT P#
                    FROM P)
                  EXCEPT
                  (SELECT P#
                   FROM SP
                   WHERE S.S# = SP.S#))
```

کارکرد قطعه پرس و جوی زیر از پرس و جوی فوق به صورت زیر است:

```
SELECT P#
FROM P
```

«شماره‌ی همه‌ی قطعات موجود در جدول قطعات»

همچنین کارکرد قطعه پرس و جوی زیر از پرس و جوی فوق به صورت زیر است:

```
SELECT P#
FROM SP
WHERE S.S# = SP.S#
```

«شماره قطعات تولیدشده توسط یک تولیدکننده»

در پرس و جوی فوق، در پراتنز داخلی کل قطعات موجود در جدول قطعه، منهای کلیه قطعات تولیدشده توسط یک تولیدکننده می‌شود و در صورتی که حاصل این تفاضل تهی باشد نام تولیدکننده مورد جستجو در خروجی ظاهر می‌شود.

بنابراین پرس و جوی فوق، شماره تولیدکنندگانی که همه قطعات موجود در جدول قطعه را تولید کرده‌اند، نشان می‌دهد.

حال به تشریح نحوه‌ی کارکرد پرس و جوی می‌پردازیم:

توسط دستور where در فرم زیر:

```
SELECT S#
FROM S
WHERE NOT EXISTS (...)
```

برای هر سطر از جدول S شرط جلوی not exists که حاصل یک تفاضل توسط عملگر except می‌باشد، محاسبه می‌گردد، اگر تهی بود، شرط جلوی where که همان not exists است، TRUE می‌گردد و سطر مورد نظر از جدول S انتخاب می‌گردد و این رویه برای تک تک سطرهای جدول S، تا به انتهای جدول S ادامه پیدا می‌کند. به بیان دیگر این پرس و جوی شماره تولیدکنندگانی را می‌دهد که پراتنز مقابل not exists برای آن‌ها تهی است. این پراتنز هنگامی تهی می‌شود که حاصل تفاضل بیان شده در این پراتنز تهی شود.

حاصل این تفاضل در صورتی تهی می شود که تعداد کل قطعات موجود در جدول قطعه برابر تعداد قطعاتی باشد که آن تولیدکننده تاکنون تولید کرده است. به عبارت دیگر این پرس و جو شماره تولیدکنندگانی را می دهد که تاکنون تمام قطعات موجود در جدول قطعه را تولید کرده اند.
توجه: همان طور که مشاهده می کنید، تولیدکننده S1 همه ی قطعات موجود در جدول قطعه را تولید کرده است.

به بیان دیگر عبارت موجود در جلوی دستور not exists که به صورت زیر است:
SELECT P#
FROM P

شماره ی همه ی قطعات موجود در جدول قطعات را نمایش می دهد.
با توجه به جداول فوق داریم:

P#
P1
P2

هم چنین عبارت بعد از دستور except یعنی:

SELECT P#
FROM SP
WHERE S.S# = SP.S#

توسط دستور S.S# = SP.S# به محیط خارج یعنی جدول S متصل می گردد. حال به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل سطرهای جدول SP از ابتدا تا انتها با توجه به شرط اتصال بررسی می گردد. مطابق شکل زیر:

ابتدا برای سطر اول از جدول S با توجه به شرط اتصال S.S# = SP.S# داریم:

Sname	S#	City	S#	P#	QTY	P#
Sn1	S1		S1	P1	10	P1
Sn2	S2		S1	P2	20	→ P2
Sn3	S3		S2	P1	30	

S SP

خروجی حاصل از عملگر تفاضل جلوی not exists به ازای سطر اول از جدول S به صورت زیر است:

P# P# خروجی
P1 - P1 = تهی
P2 P2

جلوی not exists برابر تهی گردید، بنابراین شرط where در پشت not exists TRUE می گردد. بنابراین سطر اول از جدول S به شکل زیر در خروجی نمایش داده می شود.

S#
S1

حال برای سطر دوم از جدول S با توجه به شرط اتصال S.S# = SP.S# داریم:

Sname	S#	City	S#	P#	QTY	P#
Sn1	S1		S1	P1	10	P1
Sn2	S2		S1	P2	20	
Sn3	S3		S2	P1	30	→ P1

S SP

این است که هیچ P# در جمله:

SELECT P# FROM P WHERE where
در همان جمله برقرار نباشد. اما چه وقت شرط where برقرار نیست، پاسخ، وقتی که پرس وجوی داخل
not exists دوم تهی نباشد، یعنی قطعه تولید شده باشد.

در یک بیان ساده، ابتدا سطر اول از جدول S را انتخاب می‌کنیم. برای مثال تولیدکننده S1، سپس برای
تولیدکننده S1 از جدول S بررسی می‌کنیم، که آیا همه قطعات موجود در جدول قطعه برای تولیدکننده S1
تولید شده‌اند یا خیر. بنابراین برای یک تولیدکننده خاص، توسط دستور select جلوی not exists اول، بر
روی جدول P از ابتدا تا انتهای جدول به ازای یک تولیدکننده خاص، حرکت می‌کنیم، اما به ازای حرکت
در هر سطر از جدول P، توسط دستور select جلوی not exists دوم، بررسی می‌گردد که قطعه موجود در
جدول قطعه، در جدول SP، تولید شده است یا خیر. اگر تولید شده باشد، توسط * داخل دستور select
جلوی not exists دوم نمایش داده می‌شود و توسط P# داخل دستور select جلوی not exists اول نمایش
داده نمی‌شود به دلیل FALSE بودن شرط دستور where جلوی not exists اول ناشی از FALSE بودن
not exists دوم به دلیل تهی نبودن جلوی not exists دوم. حال اگر مطابق روال فوق به ازای همه قطعات
موجود در جدول قطعه، برای یک تولیدکننده خاص، همه قطعات، تولید شده باشند، باعث می‌شود تا P#
داخل دستور select جلوی not exists اول تهی گردد و باعث شود، شرط دستور where در پشت
not exists اول TRUE گردد، بنابراین آن تولیدکننده خاص موجود در جدول S در خروجی نمایش داده
می‌شود.

پرس وجوی مطرح شده در فرم سوم را مجدداً در نظر بگیرید:

```
SELECT S#
FROM S
WHERE NOT EXISTS (SELECT P#
FROM P
WHERE NOT EXISTS (SELECT *
FROM SP
WHERE S.S# = SP.S#
And P.P# = SP.P#))
```

به شرط اتصال S.S# = SP.S# موجود در پرس وجوی فوق توجه کنید به ازای حرکت در هر سطر جدول S
مقدار شرط اتصال S.S# = SP.S# تغییر می‌کند. هم اکنون در سطر اول جدول S هستیم. پس S# برابر S1
می‌باشد.

به قطعه کد زیر موجود در جلوی not exists اول، دقت کنید:

```
NOT EXISTS ((SELECT P#
FROM P
WHERE
```

ابتدا سطر اول از جدول P یعنی قطعه P1 بررسی می‌شود.

با شرط زیر بررسی می‌شود که تولیدکننده S1، قطعه P1 را تولید کرده است یا خیر.

S.S# = SP.S#	and	SP.P# = P.P#
S1		P1
TRUE		TRUE
TRUE		

شرط برقرار است، زیرا قطعه P1 توسط تولیدکننده S1 تولید شده است، بنابراین خروجی فعلی قطعه پرس و جوی زیر:

```
NOT EXISTS (SELECT *
            FROM SP
            WHERE S.S# = SP.S#
            And P.P# = SP.P#)
```

برابر مقدار P1 (سطر P1 به دلیل استفاده از عملگر *) می باشد.

چون جلوی not exists دوم غیر تهی است، بنابراین شرط where موجود پشت not exists دوم FALSE می گردد، بنابراین سطر اول جدول قطعه یعنی قطعه P1 در خروجی دستور select جلوی not exists اول ظاهر نمی شود. چون قطعه P1 توسط تولیدکننده S1 تولید شده است.

در ادامه سطر دوم از جدول قطعه یعنی قطعه P2، همچنان برای تولیدکننده S1 بررسی می شود. با شرط زیر بررسی می شود که تولیدکننده S1، قطعه P2 را تولید کرده است یا خیر.

$S.S\# = SP.S\#$	and	$SP.P\# = P.P\#$
S1 S1		P2 P2
TRUE		TRUE
TRUE		

شرط برقرار است، زیرا قطعه P2 توسط تولیدکننده S1 تولید شده است، بنابراین خروجی فعلی قطعه پرس و جوی زیر:

```
NOT EXISTS (SELECT *
            FROM SP
            WHERE S.S# = SP.S#
            And P.P# = SP.P#)
```

برابر مقدار P2 (سطر P2 به دلیل استفاده از عملگر *) می باشد.

چون جلوی not exists دوم غیر تهی است، بنابراین شرط where موجود پشت not exists دوم FALSE می گردد، بنابراین سطر دوم جدول قطعه یعنی قطعه P2 نیز در خروجی دستور select جلوی not exists اول ظاهر نمی شود. چون قطعه P2 توسط تولیدکننده S1 تولید شده است.

نتیجه این که تولیدکننده S1 همه قطعات موجود در جدول قطعه را تولید کرده است. حال یکبار دیگر پرس و جوی مطرح شده در فرم سوم را در نظر بگیرید:

```
SELECT S#
FROM S
WHERE NOT EXISTS ((SELECT P#
                   FROM P
                   WHERE NOT EXISTS (SELECT *
                                     FROM SP
                                     WHERE S.S# = SP.S#
                                     And P.P# = SP.P#)
```

در نهایت خروجی قطعه پرس و جوی زیر:

```
SELECT P#
FROM P
WHERE NOT EXISTS (...)
```

برای تولیدکننده S1 موجود در سطر اول جدول تولیدکننده برابر مقدار تهی می‌شود. چون به دلیل شرایط جلوی not exists دوم مشاهده نمودید که هیچ یک از سطرهای جدول قطعه در خروجی دستور select جلوی not exists اول ظاهر نشدند، زیرا در خروجی P# select جلوی not exists اول، قطعات تولیدنشده، توسط یک تولیدکننده خاص، خارج می‌گردند، از آن‌جا که پس از بررسی قطعات تولیدشده برای تولیدکننده S1 توسط دستور select در جلوی not exists دوم، مشخص شد که تولیدکننده S1، همه‌ی قطعات را تولید کرده است، بنابراین خروجی P# select جلوی not exists اول، تهی خواهد شد. بنابراین مطالب فوق شرط where در پشت not exists اول برای تولیدکننده S1 برابر مقدار TRUE خواهد بود، پس سطر اول جدول تولیدکننده انتخاب می‌گردد و S# با مقدار S1 در خروجی قرار می‌گیرد. توجه: در یک بیان ساده اگر قطعه‌ای برای یک تولیدکننده خاص تولیدشده باشد در خروجی قطعه پرس و جوی فوق قرار نمی‌گیرد. تولید یک قطعه توسط یک تولیدکننده خاص توسط دستور select در جلوی not exists دوم بررسی می‌گردد. حال سطر دوم از جدول تولیدکننده بررسی می‌شود، تا بررسی کامل در همین سطر توقف کنید. به قطعه کد زیر موجود در جلوی not exists اول، دقت کنید:

```
NOT EXISTS ((SELECT P#
             FROM P
             WHERE
```

ابتدا سطر اول از جدول P یعنی قطعه P1 بررسی می‌شود.

با شرط زیر بررسی می‌شود که تولیدکننده S2، قطعه P1 را تولید کرده است یا خیر.

S.S# = SP.S#	and	SP.P# = P.P#
S2		P1
S2		P1
TRUE		TRUE
TRUE		

شرط برقرار است، زیرا قطعه P1 توسط تولیدکننده S2 تولیدشده است، بنابراین خروجی فعلی قطعه پرس و جوی زیر:

```
NOT EXISTS (SELECT *
            FROM SP
            WHERE S.S# = SP.S#
            And P.P# = SP.P#)
```

برابر مقدار P1 (سطر P1 به دلیل استفاده از عملگر *) می‌باشد.

چون جلوی not exists دوم غیر تهی است، بنابراین شرط where موجود پشت not exists دوم FALSE می‌گردد، بنابراین سطر اول جدول قطعه یعنی قطعه P1 در خروجی دستور select جلوی not exists اول ظاهر نمی‌شود. چون قطعه P1 توسط تولیدکننده S2 تولیدشده است. در ادامه سطر دوم از جدول قطعه یعنی قطعه P2، همچنان برای تولیدکننده S2 بررسی می‌شود. با شرط زیر بررسی می‌شود که تولیدکننده S2، قطعه P2 را تولید کرده است یا خیر.

S.S# = SP.S#	and	SP.P# = P.P#
S2	×	P2
S2		P2
FALSE		FALSE
FALSE		

شرط برقرار نیست، زیرا قطعه P2 توسط تولیدکننده S2 تولید نشده است، بنابراین خروجی فعلی قطعه پرس و جوی زیر:

```
NOT EXISTS (SELECT *
            FROM SP
            WHERE S.S# = SP.S#
            And P.P# = SP.P#)
```

برابر مقدار تهی می‌باشد.

چون جلوی not exists دوم تهی است، بنابراین شرط where موجود پشت not exists دوم TRUE می‌گردد، بنابراین سطر دوم جدول قطعه یعنی قطعه P2 در خروجی دستور select جلوی not exists اول ظاهر می‌شود. چون قطعه P2 توسط تولیدکننده S2 تولید نشده است. نتیجه این که تولیدکننده S2 همه قطعات موجود در جدول قطعه را تولید نکرده است.

```
SELECT S#
FROM S
WHERE NOT EXISTS ((SELECT P#
                   FROM P
                   WHERE NOT EXISTS (SELECT *
                                     FROM SP
                                     WHERE S.S# = SP.S#
                                     And P.P# = SP.P#)
```

در نهایت خروجی قطعه پرس و جوی زیر:

```
SELECT P#
FROM P
WHERE NOT EXISTS (...)
```

برای تولیدکننده S2 موجود در سطر دوم جدول تولیدکننده برابر مقدار غیر تهی و برابر مقدار P2 می‌شود. چون به دلیل شرایط جلوی not exists دوم مشاهده نمودید که سطر دوم از جدول قطعه در خروجی دستور select جلوی not exists اول ظاهر شد، زیرا در خروجی P# select جلوی not exists اول، قطعات تولید نشده، توسط یک تولیدکننده خاص، خارج می‌گردند، از آنجا که پس از بررسی قطعات تولید شده برای تولیدکننده S2 توسط دستور select در جلوی not exists دوم، مشخص شد که تولیدکننده S2، قطعه P2 را تولید نکرده است، بنابراین خروجی P# select جلوی not exists اول، غیر تهی خواهد شد. بنابراین مطالب فوق شرط where در پشت not exists اول برای تولیدکننده S2 برابر مقدار FALSE خواهد بود، پس سطر دوم جدول تولیدکننده انتخاب نمی‌گردد و S# با مقدار S2 در خروجی قرار نمی‌گیرد. در نهایت سطر سوم از جدول تولیدکننده بررسی می‌شود، که نتیجه‌ای همچون سطر دوم از جدول تولیدکننده خواهد داشت.

نتیجه اینکه، پرس و جوی فرم سوم شماره تولیدکنندگانی را می‌دهد که قطعه‌ای وجود ندارد که آنها آنرا تولید نکرده باشند. به عبارت دیگر شماره تولیدکنندگانی را می‌دهد که همه قطعات موجود در جدول قطعه را تولید کرده‌اند.

بررسی گزینه‌های اول و سوم:

مطابق پرس و جوهای مطرح شده در گزینه‌های اول و سوم داریم:
 «شماره تولیدکنندگانی که حداقل یک قطعه را تولید کرده‌اند.»
 در ادامه، این پرس و جو را به چهار فرم زیر می‌نویسیم:

فرم اول:

```
SELECT distinct (S.S#)
FROM S, SP
WHERE S.S# = SP.S#
```

فرم دوم:

```
SELECT S#
FROM S
WHERE S# IN (SELECT S#
             FROM SP)
```

ابتدا زیر پرس وجوی داخلی یک بار و برای همیشه اجرا می‌گردد، سپس پرس وجوی خارجی به ازای حرکت در هر یک از سطرهاى خود، از مقادیر زیر پرس وجوی داخلی استفاده می‌کند. در پرس وجوی فوق به ازای حرکت در هر سطر از جدول S مجموعه جلوی in بررسی می‌گردد. آیا S# داخل این مجموعه قرار دارد یا خیر. اگر قرار داشت سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

توجه: فرم دوم مطرح شده، مطابق گزینه اول می‌باشد، که مطابق پرس و جوی مطرح شده در صورت سوال نمی‌باشد، بنابراین پاسخ سوال نیز نمی‌باشد.

فرم سوم:

```
SELECT S#
FROM S
WHERE EXISTS (SELECT P#
              FROM SP
              WHERE S.S# = SP.S#)
```

به ازای حرکت در هر یک از سطرهاى پرس وجوی خارجی، یک بار به طور کامل از ابتدا تا انتها زیر پرس وجوی داخلی اجرا و براساس شرطی که زیر پرس وجوی داخلی را به پرس وجوی خارجی متصل می‌کند، بررسی انجام می‌شود. مانند دو حلقه تو در تو، که به ازای هر بار اجرای حلقه خارجی، یک بار به طور کامل حلقه داخلی اجرا می‌گردد.

در پرس وجوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول SP بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول SP وجود دارد که شرط اتصال S.S# = SP.S# موجود در select داخلی را برقرار کند، اگر وجود داشته باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

فرم چهارم:

```
SELECT S#
FROM S
WHERE EXISTS (SELECT P#
              FROM P
              WHERE EXISTS (SELECT *
                            FROM SP
                            WHERE S.S# = SP.S# AND SP.P# = P.P#))
```

در پرس وجوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول P بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول P وجود دارد که شرط اتصال $S.S\# = SP.S\# \text{ AND } SP.P\# = P.P\#$ موجود در select داخلی را برقرار کند، اگر وجود داشته باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

توجه: فرم چهارم مطرح شده، مطابق گزینه سوم می‌باشد، که مطابق پرس و جوی مطرح شده در صورت سوال نمی‌باشد، بنابراین پاسخ سوال نیز نمی‌باشد.

راه حل دوم:

پرس و جوی مطرح شده در صورت سوال را در نظر بگیرید:

«شماره تولیدکنندگانی که همه قطعات را تولید کرده‌اند.»

توجه: عملگر تقسیم در جبر رابطه‌ای توسط الگوی اول در SQL قابل پیاده‌سازی است. عملگر تقسیم در جبر رابطه‌ای زمانی مورد استفاده قرار می‌گیرد که پرس و جو بخواهد همه حالت‌های یک اتفاق را بررسی کند. مانند شماره تولید تولیدکنندگانی که همه قطعات را تولید کرده‌اند. که تولیدکردن قطعات توسط تولیدکنندگان، اتفاق و مقسوم است و همه قطعات، حالت‌ها و مقسوم علیه است.

توجه: این پرس و جو به فرم زیر توسط الگوی اول (همه) در SQL قابل پیاده‌سازی است.

الگوی اول (00):

```
SELECT S#
FROM S
WHERE NOT EXISTS (SELECT P#
                  FROM P
                  WHERE NOT EXISTS (SELECT *
                                    FROM SP
                                    WHERE S.S# = SP.S#
                                    AND P.P# = SP.P#))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«شماره تولیدکنندگانی که، وجود نداشته باشد (NOT EXISTS) قطعه‌ای که، تولید نشده باشد (NOT EXISTS). یعنی شماره تولیدکنندگانی که همه قطعات را تولید کرده‌اند.»

در پرس وجوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول P بررسی می‌گردد، تا مشخص گردد آیا هیچ سطری در جدول P وجود ندارد که شرط اتصال $S.S\# = SP.S\# \text{ AND } P.P\# = SP.P\#$ موجود در select داخلی را برقرار نکند، اگر وجود نداشته باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

توجه: شرط لازم برای تبعیت از الگوی اول وجود NOT EXISTS اول و NOT EXISTS دوم است و شرط کافی برای تبعیت از الگوی اول وجود شروط اتصال متناسب و سازگار است.

۶۲- گزینه (۳) صحیح است.

راه حل اول:

جداول زیر را در نظر بگیرید:

<u>sid</u>	sname	...	<u>sid</u>	<u>bid</u>	...	<u>bid</u>	...
s1	sn1		s1	b1		b1	
s2	sn2		s1	b2		b2	
s3	sn3		s2	b1			

قایقران Sailor
S
رزرو Reservation
R
قایق Boats
B

مطابق پرس و جوی مطرح شده در صورت سوال داریم:

«نام قایقران‌هایی که همه قایق‌ها را رزرو کرده‌اند.»

پرس و جوی فوق توسط عملگر تقسیم در جبر رابطه‌ای قابل پیاده‌سازی است. اما در SQL می‌بایست عملگر تقسیم شبیه‌سازی گردد.

در ادامه، این پرس و جو را به دو فرم زیر شبیه‌سازی می‌کنیم:

فرم اول:

```
SELECT Sname
FROM Sailors S
WHERE NOT EXISTS ((SELECT bid
FROM Boats B)
EXCEPT
(SELECT bid
FROM Reservation R
WHERE R.sid = S.sid))
```

توجه: گزینه اول خطای نحوی دارد و به دلیل داشتن دستور اضافه $R.bid = B.bid$ از سوی کامپایلر کامپایل نخواهد شد. دقت کنید که جدول Boats در پرس و جوی بعد از دستور `except` قابل شناخت نیست و به تبع حضور دستور $R.bid = B.bid$ در پرس و جوی بعد از دستور `except` از سوی کامپایلر خطای کامپایلری محسوب می‌گردد، فرم صحیح گزینه اول به شکل فرم اول مطرح شده می‌باشد. بنابراین گزینه اول پاسخ سوال نمی‌باشد.

کارکرد قطعه پرس و جوی زیر از پرس و جوی فوق به صورت زیر است:

```
SELECT bid
FROM Boats B
```

«شماره‌ی همه‌ی قایق‌های موجود در جدول قایق»

همچنین کارکرد قطعه پرس و جوی زیر از پرس و جوی فوق به صورت زیر است:

```
SELECT bid
FROM Reservation R
WHERE R.sid = S.sid
```

«شماره قایق‌های رزرو شده توسط یک قایقران»

در پرس و جوی فوق، در پرانتز داخلی کل قایق‌های موجود در جدول قایق، منهای کلیه قایق‌های رزرو شده توسط یک قایقران می‌شود و در صورتی که حاصل این تفاضل تهی باشد نام قایقران مورد جستجو در خروجی ظاهر می‌شود.

بنابراین پرس و جوی فوق، نام قایقران‌هایی که همه قایق‌های موجود در جدول قایق را رزرو کرده‌اند، نشان می‌دهد.

حال به تشریح نحوه‌ی کارکرد پرس و جو می‌پردازیم:
توسط دستور where در فرم زیر:

```
SELECT Sname
FROM Sailors S
WHERE NOT EXISTS (...)
```

به بیان دیگر این پرس و جو شماره تولیدکنندگانی را می‌دهد که پُرانتز مقابل not exists برای آن‌ها تهی است. این پُرانتز هنگامی تهی می‌شود که حاصل تفاضل بیان شده در این پُرانتز تهی شود. حاصل این تفاضل در صورتی تهی می‌شود که تعداد کل قطعات موجود در جدول قطعه برابر تعداد قطعاتی باشد که آن تولیدکننده تاکنون تولید کرده است. به عبارت دیگر این پرس و جو شماره تولیدکنندگانی را می‌دهد که تاکنون تمام قطعات موجود در جدول قطعه را تولید کرده‌اند.

برای هر سطر از جدول Sailor شرط جلوی not exists که حاصل یک تفاضل توسط عملگر except می‌باشد، محاسبه می‌گردد، اگر تهی بود، شرط جلوی where که همان not exists است، TRUE می‌گردد و سطر مورد نظر از جدول Sailor انتخاب می‌گردد و این رویه برای تک تک سطرهای جدول Sailor، تا به انتهای جدول Sailor ادامه پیدا می‌کند. به بیان دیگر پرس و جوی فوق نام قایقران‌هایی را می‌دهد که پُرانتز مقابل not exists برای آن‌ها تهی است. این پُرانتز هنگامی تهی می‌شود که حاصل تفاضل بیان شده در این پُرانتز تهی شود. حاصل این تفاضل در صورتی تهی می‌شود که تعداد کل قایق‌ها برابر تعداد قایق‌هایی باشد که آن قایقران تاکنون رزرو کرده است. به عبارت دیگر پرس و جوی فوق نام قایقران‌هایی را می‌دهد که تاکنون تمام قایق‌ها را رزرو کرده‌اند.

توجه: همان‌طور که مشاهده می‌کنید، قایقران s1، همه‌ی قایق‌های موجود در جدول قایق را رزرو کرده است.

به بیان دیگر عبارت موجود در جلوی دستور not exists که به صورت زیر است:

```
SELECT Sname
FROM Sailors S
```

شماره‌ی همه‌ی قطعات موجود در جدول قطعات را نمایش می‌دهد.
با توجه به جداول فوق داریم:

```
bid
b1
b2
```

هم‌چنین عبارت بعد از دستور except یعنی:

```
SELECT bid
FROM Reservation R
WHERE R.sid = S.sid
```

توسط دستور S.sid=R.sid به محیط خارج یعنی جدول S یا Sailor متصل می‌گردد. حال به ازای حرکت در هر سطر از جدول Sailor، یک بار به طور کامل سطرهای جدول Reservation از ابتدا تا انتها با توجه به شرط اتصال بررسی می‌گردد. مطابق شکل زیر:

ابتدا برای سطر اول از جدول Sailor با توجه به شرط اتصال $S.sid = R.sid$ داریم:

sname	<u>sid</u>	...	<u>sid</u>	<u>bid</u>	...
sn1	s1		s1	b1	
sn2	s2		s1	b2	
sn3	s3		s2	b1	

Sailor قایقران S رزرو Reservation R

$\frac{R.bid}{b1}$
 \rightarrow $\frac{b1}{b2}$

خروجی حاصل از عملگر تفاضل جلوی $not\ exists$ به ازای سطر اول از جدول Sailor به صورت زیر است:

خروجی $\frac{bid}{b1} _ \frac{bid}{b2}$
 $\frac{b1}{b2} =$ تهی

جلوی $not\ exists$ برابر تهی گردید، بنابراین شرط $where$ در پشت $not\ exists$ TRUE می‌گردد. بنابراین سطر اول از جدول Sailor به شکل زیر در خروجی نمایش داده می‌شود.

Sname
 Sn1

حال برای سطر دوم از جدول Sailor با توجه به شرط اتصال $S.sid = R.sid$ داریم:

sname	<u>sid</u>	...	<u>sid</u>	<u>bid</u>	...
sn1	s1		s1	b1	
sn2	s2		s1	b2	
sn3	s3		s2	b1	

Sailor قایقران S رزرو Reservation R

\rightarrow $\frac{bid}{b1}$

خروجی حاصل از عملگر تفاضل جلوی $not\ exists$ به ازای سطر دوم از جدول Sailor به صورت زیر است:

خروجی $\frac{bid}{b1} _ \frac{bid}{b2}$
 $\frac{b1}{b2} =$ b2

جلوی $not\ exists$ برابر غیر تهی گردید، بنابراین شرط $where$ در پشت $not\ exists$ FALSE می‌گردد، بنابراین سطر دوم از جدول Sailor در خروجی نمایش داده نخواهد شد.

در ادامه برای سطر سوم از جدول Sailor با توجه به شرط اتصال $S.sid = R.sid$ داریم:

S3 ×

sname	<u>sid</u>	...	<u>sid</u>	<u>bid</u>	...	<u>bid</u>
sn1	s1		s1	b1		→
sn2	s2		s1	b2		تهی
sn3	s3	هیچ →	s2	b1		

قایقران S
رزرو R

خروجی حاصل از عملگر تفاضل جلوی `not exists` به ازای سطر سوم از جدول `Sailor S` به صورت زیر است:

<u>bid</u>	<u>bid</u>	خروجی
b1	b1	تهی =
b2	b2	

جلوی `not exists` برابر غیر تهی گردید، بنابراین شرط `where` در پشت `not exists` `FALSE` می‌گردد، بنابراین سطر سوم از جدول `Sailor` در خروجی نمایش داده نخواهد شد. بنابراین خروجی نهایی برای «نام قایقران‌هایی که همه قایق‌ها را رزرو کرده‌اند» به صورت زیر است:

<u>Sname</u>
Sn1

فرم دوم:

```

SELECT Sname
FROM Sailors S
WHERE NOT EXISTS (SELECT bid
                  FROM Boats B
                  WHERE NOT EXISTS (SELECT *
                                    FROM Reservation R
                                    WHERE R.bid = B.bid
                                    AND R.sid = S.sid))

```

توجه: فرم دوم مطرح شده، مطابق گزینه سوم می‌باشد، بنابراین گزینه سوم پاسخ سوال خواهد بود.

حال به بررسی فرم دوم می‌پردازیم:

الگوریتم فوق به این شکل حرکت می‌کند، به شرح الگوریتم دقت کنید:

ابتدا سطر اول از جدول قایقران بررسی می‌شود تا بررسی کامل در همین سطر توقف کنید، اگر جلوی `not exists` اول تهی باشد، نام قایقران از جدول `Sailor` در خروجی به عنوان قایقرانی که همه قایق‌ها را رزرو کرده است، نمایش داده می‌شود. اما مساله این است که چه وقت جلوی `not exists` اول تهی می‌گردد، پاسخ این است که هیچ `bid` در جمله:

`select bid from Boats B where` موجود نباشد، واضح است که وقتی این اتفاق می‌افتد که شرط `where` در همان جمله برقرار نباشد. اما چه وقت شرط `where` برقرار نیست، پاسخ، وقتی که پرس‌وجوی داخل `not exists` دوم تهی نباشد، یعنی قایق رزرو شده باشد.

در یک بیان ساده، ابتدا سطر اول از جدول قایقران را انتخاب می‌کنیم. برای مثال قایقران `S1`، سپس برای قایقران `S1` از جدول قایقران بررسی می‌کنیم، که آیا همه قایق‌های موجود در جدول قایق برای قایقران `S1`

رزرو شده‌اند یا خیر. بنابراین برای یک قایقران خاص، توسط دستور select جلوی not exists اول، بر روی جدول قایق از ابتدا تا انتهای جدول به ازای یک قایقران خاص، حرکت می‌کنیم، اما به ازای حرکت در هر سطر از جدول قایق، توسط دستور select جلوی not exists دوم، بررسی می‌گردد که قایق موجود در جدول قایق، در جدول رزرو، رزرو شده است یا خیر. اگر رزرو شده باشد، توسط bid داخل دستور select جلوی not exists دوم نمایش داده می‌شود و توسط bid داخل دستور select جلوی not exists اول نمایش داده نمی‌شود به دلیل FALSE بودن شرط دستور where جلوی not exists اول ناشی از FALSE بودن not exists دوم به دلیل تهی نبودن جلوی not exists دوم. حال اگر مطابق روال فوق به ازای همه‌ی قایق‌های موجود در جدول قایق، برای یک قایقران خاص، همه قایق‌ها، رزرو شده باشند، باعث می‌شود تا bid داخل دستور select جلوی not exists اول تهی گردد و باعث شود، شرط دستور where در پشت not exists اول TRUE گردد، بنابراین آن قایقران خاص موجود در جدول قایقران در خروجی نمایش داده می‌شود.

پرس‌وجوی مطرح شده در فرم دوم را مجدداً در نظر بگیرید:

```
SELECT Sname
FROM Sailors S
WHERE NOT EXISTS (SELECT bid
                   FROM Boats B
                   WHERE NOT EXISTS (SELECT *
                                     FROM Reservation R
                                     WHERE R.bid = B.bid
                                     AND R.sid = S.sid))
```

به شرط اتصال R.sid = S.sid موجود در پرس‌وجوی فوق توجه کنید به ازای حرکت در هر سطر جدول Sailor جمله R.sid=S.sid تغییر می‌کند. هم اکنون در سطر اول جدول Sailor هستیم. پس sid برابر s1 می‌باشد.

به قطعه کد زیر موجود در جلوی not exists اول، دقت کنید:

```
NOT EXISTS ((SELECT bid
             FROM Boats B
             WHERE
```

ابتدا سطر اول از جدول قایق یعنی قایق b1 بررسی می‌شود.

با شرط زیر بررسی می‌شود که قایقران s1، قایق b1 را رزرو کرده است یا خیر.

$R.bid = B.bid$	and	$R.sid = S.sid$
b1 b1		s1 s1
TRUE		TRUE
 TRUE		

شرط برقرار است، زیرا قایق b1 توسط قایقران s1 رزرو شده است، بنابراین خروجی فعلی قطعه پرس و جوی زیر:

```
NOT EXISTS (SELECT *
            FROM Reservation R
            WHERE R.bid = B.bid
            AND R.sid = S.sid))
```

برابر مقدار b1 (سطر b1 به دلیل استفاده از عملگر *) می‌باشد. چون جلوی not exists دوم غیر تهی است، بنابراین شرط where موجود پشت not exists دوم FALSE می‌گردد، بنابراین سطر اول جدول قایق یعنی قایق b1 در خروجی دستور select جلوی not exists ظاهر نمی‌شود. چون قایق b1 توسط قایقران s1 رزرو شده است. در ادامه سطر دوم از جدول قایق یعنی b2، همچنان برای قایقران s1 بررسی می‌شود. با شرط زیر بررسی می‌شود که قایقران s1، قایق b2 را رزرو کرده است یا خیر.

R.bid = B.bid	and	R.sid = S.sid
b2 b2		s1 s1
TRUE		TRUE
└──────────────────┘		
TRUE		

شرط برقرار است، زیرا قایق b2 توسط قایقران s1 رزرو شده است، بنابراین خروجی فعلی قطعه پرس و جوی زیر:

```
NOT EXISTS (SELECT *
             FROM Reservation R
             WHERE R.bid = B.bid
             AND R.sid = S.sid))
```

برابر مقدار b2 (سطر b2 به دلیل استفاده از عملگر *) می‌باشد. چون جلوی not exists دوم غیر تهی است، بنابراین شرط where موجود پشت not exists دوم FALSE می‌گردد، بنابراین سطر دوم جدول قایق یعنی قایق b2 نیز در خروجی دستور select جلوی not exists ظاهر نمی‌شود. چون قایق b2 توسط قایقران s1 رزرو شده است. نتیجه این که قایقران s1 همه قایق‌های موجود در جدول قایق را رزرو کرده است. حال یکبار دیگر پرس و جوی مطرح شده در فرم دوم را در نظر بگیرید:

```
SELECT Sname
FROM Sailors S
WHERE NOT EXISTS (SELECT bid
                  FROM Boats B
                  WHERE NOT EXISTS (SELECT *
                                     FROM Reservation R
                                     WHERE R.bid = B.bid
                                     AND R.sid = S.sid))
```

در نهایت خروجی قطعه پرس و جوی زیر:

```
SELECT bid
FROM Boats B
WHERE NOT EXISTS (...)
```

برای قایقران s1 موجود در سطر اول جدول قایقران برابر مقدار تهی می‌شود. چون به دلیل شرایط جلوی not exists دوم مشاهده نمودید که هیچ یک از سطرهای جدول قایق در خروجی دستور select جلوی not exists اول ظاهر نشدند، زیرا در خروجی select bid جلوی not exists اول، قایق‌های رزرو نشده، توسط یک قایقران خاص، خارج می‌گردند، از آن‌جا که پس از بررسی قایق‌های رزرو شده برای قایقران s1

توسط دستور select در جلوی not exists دوم، مشخص شد که قایقران s1، همه‌ی قایق‌ها را رزرو کرده است، بنابراین خروجی select bid جلوی not exists اول، تهی خواهد شد. بنابراین مطالب فوق شرط where در پشت not exists اول برای قایقران s1 برابر مقدار TRUE خواهد بود، پس سطر اول جدول قایقران انتخاب می‌گردد و sname با مقدار sn1 در خروجی قرار می‌گیرد. توجه: در یک بیان ساده اگر قایقی برای یک قایقران خاص رزورده باشد در خروجی قطعه پرس و جوی فوق قرار نمی‌گیرد. رزرو یک قایق توسط یک قایقران خاص توسط دستور select در جلوی not exists دوم بررسی می‌گردد. حال سطر دوم از جدول تولیدکننده بررسی می‌شود، تا بررسی کامل در همین سطر توقف کنید. به قطعه کد زیر موجود در جلوی not exists اول، دقت کنید:

```
NOT EXISTS ((SELECT bid
             FROM Boats B
             WHERE
```

ابتدا سطر اول از جدول قایق یعنی قایق b1 بررسی می‌شود.

با شرط زیر بررسی می‌شود که قایقران s2، قایق b1 را رزرو کرده است یا خیر.

$$\begin{array}{ccccccc} R.\text{bid} & = & B.\text{bid} & \text{ and } & R.\text{sid} & = & S.\text{sid} \\ b1 & & b1 & & s2 & & s2 \\ \hline & & \text{TRUE} & & \text{TRUE} & & \\ & & & & \text{TRUE} & & \end{array}$$

شرط برقرار است، زیرا قایق b1 توسط قایقران s2 رزرو شده است، بنابراین خروجی فعلی قطعه پرس و جوی زیر:

```
NOT EXISTS (SELECT *
            FROM Reservation R
            WHERE R.bid = B.bid
            AND R.sid = S.sid)
```

برابر مقدار b1 (سطر b1 به دلیل استفاده از عملگر *) می‌باشد.

چون جلوی not exists دوم غیر تهی است، بنابراین شرط where موجود پشت not exists دوم FALSE می‌گردد، بنابراین سطر اول جدول قایق یعنی قایق b1 در خروجی دستور select جلوی not exists اول ظاهر نمی‌شود. چون قایق b1 توسط قایقران s2 رزرو شده است.

در ادامه سطر دوم از جدول قایق یعنی قایق b2، همچنان برای قایقران s2 بررسی می‌شود.

با شرط زیر بررسی می‌شود که قایقران s2، قایق b2 را رزرو کرده است یا خیر.

$$\begin{array}{ccccccc} R.\text{bid} & = & B.\text{bid} & \text{ and } & R.\text{sid} & = & S.\text{sid} \\ \times & & b2 & & \times & & s2 \\ \hline & & \text{FALSE} & & \text{FALSE} & & \\ & & & & \text{FALSE} & & \end{array}$$

شرط برقرار نیست، زیرا قایق b2 توسط قایقران s2 رزرو نشده است، بنابراین خروجی فعلی قطعه پرس و جوی زیر:

```
NOT EXISTS (SELECT *
FROM Reservation R
WHERE R.bid = B.bid
AND R.sid = S.sid))
```

برابر مقدار تهی می‌باشد.

چون جلوی not exists دوم تهی است، بنابراین شرط where موجود پشت not exists دوم TRUE می‌گردد، بنابراین سطر دوم جدول قایق یعنی قایق b2 در خروجی دستور select جلوی not exists اول ظاهر می‌شود. چون قایق b2 توسط قایقران s2 رزرو نشده است. نتیجه این که قایقران s2 همه قایق‌های موجود در جدول قایق را رزرو نکرده است. حال یکبار دیگر پرس و جوی مطرح شده در فرم دوم را در نظر بگیرید:

```
SELECT Sname
FROM Sailors S
WHERE NOT EXISTS (SELECT bid
FROM Boats B
WHERE NOT EXISTS (SELECT *
FROM Reservation R
WHERE R.bid = B.bid
AND R.sid = S.sid))
```

در نهایت خروجی قطعه پرس و جوی زیر:

```
SELECT bid
FROM Boats B
WHERE NOT EXISTS (...)
```

برای قایقران s2 موجود در سطر دوم جدول قایقران برابر مقدار غیر تهی و برابر مقدار b2 می‌شود. چون به دلیل شرایط جلوی not exists دوم مشاهده نمودید که سطر دوم از جدول قایق در خروجی دستور select جلوی not exists اول ظاهر شد، زیرا در خروجی select bid جلوی not exists اول، قایق‌های رزرو نشده، توسط یک قایقران خاص، خارج می‌گردند، از آنجا که پس از بررسی قایق‌های رزرو شده برای قایقران s2 توسط دستور select در جلوی not exists دوم، مشخص شد که قایقران s2، قایق b2 را رزرو نکرده است، بنابراین خروجی select bid جلوی not exists اول، غیر تهی خواهد شد.

بنابراین مطالب فوق شرط where در پشت not exists اول برای قایقران s2 برابر مقدار FALSE خواهد بود، پس سطر دوم جدول قایقران انتخاب نمی‌گردد و sname با مقدار s2 در خروجی قرار نمی‌گیرد. در نهایت سطر سوم از جدول قایقران بررسی می‌شود، که نتیجه‌ای همچون سطر دوم از جدول قایقران خواهد داشت.

نتیجه اینکه، پرس و جوی فرم دوم نام قایقران‌هایی را می‌دهد که قایقی وجود ندارد که آنها آنرا رزرو نکرده باشند. به عبارت دیگر نام قایقران‌هایی را می‌دهد که همه قایق‌های موجود در جدول قایق را رزرو کرده‌اند.

بررسی گزینه دوم:

مطابق پرس و جوی مطرح شده در گزینه دوم داریم:

«نام قایقران‌هایی که حداقل یک قطعه را تولید نکرده‌اند.»

در ادامه، این پرس و جو را به دو فرم زیر شبیه‌سازی می‌کنیم:

فرم اول:

```
SELECT Sname
FROM Sailors S
WHERE EXISTS ((SELECT bid
                FROM Boats B)
              EXCEPT
              (SELECT bid
               FROM Reservation R
               WHERE R.sid = S.sid)))
```

توجه: گزینه دوم خطای نحوی دارد و به دلیل داشتن دستور اضافه $R.bid = B.bid$ از سوی کامپایلر کامپایل نخواهد شد. دقت کنید که جدول Boats در پرس و جوی بعد از دستور except قابل شناخت نیست و به تبع حضور دستور $R.bid = B.bid$ در پرس و جوی بعد از دستور except از سوی کامپایلر خطای کامپایلری محسوب می‌گردد، فرم صحیح گزینه دوم به شکل فرم اول مطرح شده می‌باشد. بنابراین گزینه دوم پاسخ سوال نمی‌باشد.

فرم دوم:

```
SELECT Sname
FROM Sailors S
WHERE EXISTS (SELECT bid
              FROM Boats B
              WHERE NOT EXISTS (SELECT *
                                FROM Reservation R
                                WHERE R.bid = B.bid
                                AND R.sid = S.sid))
```

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول قایقران، یک بار به طور کامل از ابتدا تا انتهای جدول قایق بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول قایق وجود دارد که شرط اتصال $R.bid = B.bid$ AND $R.sid = S.sid$ موجود در select داخلی را برقرار نکند، اگر وجود داشته باشد، سطر مورد نظر از جدول قایقران در خروجی نمایش داده می‌شود.

بررسی گزینه چهارم:

مطابق پرس و جوی مطرح شده در گزینه چهارم داریم:
«نام قایقران‌هایی که حداقل یک قطعه را تولید کرده‌اند.»
در ادامه، این پرس و جو را به چهار فرم زیر می‌نویسیم:

فرم اول:

```
SELECT distinct (S.sname)
FROM Sailors S, Reservation R
WHERE S.sid = R.sid
```

فرم دوم:

```
SELECT S.sname
FROM Sailors S
WHERE sid IN (SELECT sid
              FROM Reservation R)
```

ابتدا زیر پرس وجوی داخلی یک بار و برای همیشه اجرا می‌گردد، سپس پرس وجوی خارجی به ازای حرکت در هر یک از سطرهای خود، از مقادیر زیر پرس وجوی داخلی استفاده می‌کند. در پرس وجوی فوق به ازای حرکت در هر سطر از جدول قایقران مجموعه جلوی in بررسی می‌گردد. آیا sid داخل این مجموعه قرار دارد یا خیر. اگر قرار داشت سطر مورد نظر از جدول قایقران در خروجی نمایش داده می‌شود.

فرم سوم:

```
SELECT S.sname
FROM Sailors S
WHERE EXISTS (SELECT bid
              FROM Reservation R
              WHERE S.sid = R.sid)
```

به ازای حرکت در هر یک از سطرهای پرس وجوی خارجی، یک بار به طور کامل از ابتدا تا انتها زیر پرس وجوی داخلی اجرا و براساس شرطی که زیر پرس وجوی داخلی را به پرس وجوی خارجی متصل می‌کند، بررسی انجام می‌شود. مانند دو حلقه تو در تو، که به ازای هر بار اجرای حلقه خارجی، یک بار به طور کامل حلقه داخلی اجرا می‌گردد.

در پرس وجوی فوق به ازای حرکت در هر سطر از جدول قایقران، یک بار به طور کامل از ابتدا تا انتهای جدول رزرو بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول رزرو وجود دارد که شرط اتصال S.sid = R.sid موجود در select داخلی را برقرار کند، اگر وجود داشته باشد، سطر مورد نظر از جدول قایقران در خروجی نمایش داده می‌شود.

فرم چهارم:

```
SELECT Sname
FROM Sailors S
WHERE EXISTS (SELECT bid
              FROM Boats B
              WHERE EXISTS (SELECT *
                            FROM Reservation R
                            WHERE R.bid = B.bid AND R.sid = S.sid))
```

در پرس وجوی فوق به ازای حرکت در هر سطر از جدول قایقران، یک بار به طور کامل از ابتدا تا انتهای جدول قایق بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول قایق وجود دارد که شرط اتصال R.bid = B.bid AND R.sid = S.sid موجود در select داخلی را برقرار کند، اگر وجود داشته باشد، سطر مورد نظر از جدول قایقران در خروجی نمایش داده می‌شود.

توجه: فرم چهارم مطرح شده، مطابق گزینه چهارم می‌باشد، که مطابق پرس و جوی مطرح شده در صورت سوال نمی‌باشد، بنابراین پاسخ سوال نیز نمی‌باشد.

توجه: سازمان سنجش آموزش کشور در کلید اولیه خود ابتدا گزینه اول را به عنوان پاسخ اعلام نمود، سپس در کلید نهایی نظر خود را عوض کرد و گزینه سوم را به عنوان پاسخ اعلام کرد، که عمل درستی را انجام داده است.

راه حل دوم:

پرس و جوی مطرح شده در صورت سوال را در نظر بگیرید:

«نام قایقران‌هایی که همه قایق‌ها را رزرو کرده‌اند.»

توجه: عملگر تقسیم در جبر رابطه‌ای توسط الگوی اول در SQL قابل پیاده‌سازی است. عملگر تقسیم در جبر رابطه‌ای زمانی مورد استفاده قرار می‌گیرد که پرس و جو بخواهد همه حالت‌های یک اتفاق را بررسی کند. مانند نام قایقران‌هایی که همه قایق‌ها را رزرو کرده‌اند. که رزرو کردن قایق‌ها توسط قایقران‌ها، اتفاق و مقسوم است و همه قایق‌ها، حالت‌ها و مقسوم علیه است.

توجه: این پرس و جو به فرم زیر توسط الگوی اول (همه) در SQL قابل پیاده‌سازی است.

الگوی اول (00):

```
SELECT S.sname
FROM Sailor S
WHERE NOT EXISTS (SELECT bid
                  FROM Boats B
                  WHERE NOT EXISTS (SELECT *
                                    FROM Reserves R
                                    WHERE S.sid = R.sid
                                    AND B.bid = R.bid))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:
 «نام قایقران‌هایی که، وجود نداشته باشد (NOT EXISTS) قایقی که، رزرو نشده باشد (NOT EXISTS).
 یعنی نام قایقران‌هایی که همه قایق‌ها را رزرو کرده‌اند.»
 در پرس و جوی فوق به ازای حرکت در هر سطر از جدول Sailor، یک بار به طور کامل از ابتدا تا انتهای جدول Boats بررسی می‌گردد، تا مشخص گردد آیا هیچ سطری در جدول Boats وجود ندارد که شرط اتصال $S.sid = R.sid$ AND $B.bid = R.bid$ موجود در select داخلی را برقرار نکند، اگر وجود نداشته باشد، سطر مورد نظر از جدول Sailor در خروجی نمایش داده می‌شود.
توجه: شرط لازم برای تبعیت از الگوی اول وجود NOT EXISTS اول و NOT EXISTS دوم است و شرط کافی برای تبعیت از الگوی اول وجود شروط اتصال متناسب و سازگار است.

۶۳- گزینه (۱) صحیح است.

سه جدول ST، CO و STCO با مقادیر زیر را در نظر بگیرید:

STID	STNAME	COID	CONAME	CREDIT
S1	Sn1			C1	Cn1	2
S2	Sn2			C2	Cn2	3
S3	Sn3			C3	Cn3	4
S4	Sn4					

جدول ST

جدول CO

STID	COID	SEMESTER	GRADE
S1	C1	95-96-1	20
S1	C2	95-96-1	15
S1	C3	92-92-1	17
S2	C1	95-96-1	19
S2	C2	95-96-1	18
S2	C3	93-93-1	12
S3	C1	94-94-1	16

جدول STCO

مطابق پرس و جوی مطرح شده در گزینه‌ی اول، داریم:

```
Select STName, SUM(Grade*Credit)/SUM(Credit)
From ST JOIN STCO JOIN CO
Group By STID,STName,Semester
Having Semester='95-96-1'
```

که البته پرس و جوی مطرح شده در گزینه‌ی اول کمی خطای نحوی دارد، که فرم اصلاح شده آن به صورت زیر است:

```
Select STName, SUM(Grade*Credit)/SUM(Credit)
From ST JOIN STCO
on ST.STID=STCO.STID JOIN CO on CO.COID=STCO.COID
Group By ST.STID,STName,Semester
Having Semester='95-96-1'
```

توجه: در فرم اصلاح شده‌ی گزینه‌ی اول، شرط اتصال و عملگر ON و پشت ستون STID جدول ST قرار گرفت. پرس و جوی گزینه‌ی اول دو خطای نحوی داشت که اصلاح شد. با توجه به جداول فوق، خروجی پرس و جوی گزینه‌ی اول پس از انجام عملگر JOIN به صورت زیر است:

STID	STNAME	STID	COID	SEMESTER	GRADE	COID	CONAME	CREDIT
S1	Sn1	S1	C1	95-96-1	20	C1	Cn1	2
S1	Sn1	S1	C2	95-96-1	15	C2	Cn2	3
S1	Sn1	S1	C3	92-92-1	17	C3	Cn3	4
S2	Sn2	S2	C1	95-96-1	19	C1	Cn1	2
S2	Sn2	S2	C2	95-96-1	18	C2	Cn2	3
S2	Sn2	S2	C3	93-93-1	12	C3	Cn3	4
S3	Sn3	S3	C1	94-94-1	16	C1	Cn1	2

همچنین در ادامه، پس از انجام دستور Group By ST.STID,STName,Semester براساس ستون‌های ST.STID,STName,Semester خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:

S1,Sn1,95-96-1	S1,Sn1,92-92-1
C1 20 2 ...	C3 17 4 ...
C2 15 3	
گروه اول	گروه دوم
S2,Sn2,95-96-1	S2,Sn2,93-93-1
C1 19 2 ...	C3 12 4 ...
C2 18 3	
گروه سوم	گروه چهارم
S3,Sn3,94-94-1	
C1 16 2 ...	
گروه پنجم	

و در نهایت دستور '95-96-1' Having Semester برای هر گروه به طور مستقل اعمال می‌گردد.
توجه: دستور HAVING بر روی گروه‌ها، اعمال می‌گردد.

S1,Sn1,95-96-1
C1 20 2 ...
C2 15 3
گروه اول

S1,Sn1,92-92-1
C3 17 4 ...
گروه دوم

Semester = '95-96-1'

Semester = '95-96-1'

S2,Sn2,95-96-1
C1 19 2 ...
C2 18 3
گروه سوم

S2,Sn2,93-93-1
C3 12 4 ...
گروه چهارم

Semester = '95-96-1'

Semester = '95-96-1'

S3,Sn3,94-94-1
C1 16 2 ...
گروه پنجم

Semester = '95-96-1'

توجه: با توجه به شرط انتخاب گروه توسط دستور '95-96-1' Having Semester، فقط گروه‌های اول و سوم جهت نمایش در خروجی انتخاب می‌شوند.

و در نهایت دستور SUM(Grade*Credit)/SUM(Credit) داخل دستور SELECT برای هر گروه انتخاب شده توسط دستور Having به طور مستقل محاسبه می‌گردد و در خروجی پرس و جو قرار می‌گیرد.
معدل وزنی نمرات دروس اخذ شده گروه اول یعنی دانشجو Sn1 به صورت زیر محاسبه می‌گردد:

$$\frac{(20 \times 2) + (15 \times 3)}{5} = \frac{(40) + (45)}{5} = \frac{85}{5} = 17.00$$

معدل وزنی نمرات دروس اخذ شده گروه سوم یعنی دانشجو Sn2 به صورت زیر محاسبه می‌گردد:

$$\frac{(19 \times 2) + (18 \times 3)}{5} = \frac{(38) + (54)}{5} = \frac{92}{5} = 18.40$$

بنابراین خروجی نهایی پرس و جو گزینه‌ی اول به صورت زیر است:

STName	(No column name)
Sn1	17.00
Sn2	18.40

توجه: دستور GROUP BY، سرگروه‌ها را، راهی خروجی می‌کند.

که مطابق پرس و جو مطرح شده در صورت سوال است. یعنی لیست نام دانشجویان به همراه معدل وزنی نمرات دروس اخذ شده آن‌ها را در ترم 95-96-1.

توجه: گزینه‌ی اول را می‌توان بدون استفاده از دستور JOIN و به شکل کلاسیک آن یعنی با استفاده از ضرب دکارتی و شرط اتصال برابری نیز بازنویسی کرد.

فرم کلاسیک گزینه‌ی اول:

```
Select STName, SUM(Grade*Credit)/SUM(Credit)
From ST , STCO, CO
Where ST.STID=STCO.STID AND
STCO.COID=CO.COID
Group Bby ST.STID,ST.STName,Semester
Having Semester='95-96-1'
```

بنابراین خروجی نهایی فرم کلاسیک پرس و جوی گزینه‌ی اول نیز به صورت زیر است:

STName	(No column name)
Sn1	17.00
Sn2	18.40

مطابق پرس و جوی مطرح شده در گزینه‌ی چهارم، داریم:

```
Select STName, SUM(Grade*Credit)/SUM(Credit)
From ST JOIN STCO JOIN CO
Where Semester='95-96-1'
Group By STID,Semester
```

که البته پرس و جوی مطرح شده در گزینه‌ی چهارم کمی خطای نحوی دارد، که فرم اصلاح شده آن به صورت زیر است:

```
Select STName, SUM(Grade*Credit)/SUM(Credit)
From ST JOIN STCO
on ST.STID=STCO.STID JOIN CO on CO.COID=STCO.COID
Where Semester='95-96-1'
Group By ST.STID,STName,Semester
```

توجه: در فرم اصلاح شده‌ی گزینه‌ی چهارم، شرط اتصال و عملگر ON و پشت ستون STID جدول ST قرار گرفت. پرس و جوی گزینه‌ی چهارم تا به اینجا دو خطای نحوی داشت که اصلاح شد. اما یک **خطای نحوی پررنگ‌تری** هم دارد، اینکه همواره به غیر از توابع آماری، همه ستون‌های جلوی دستور Select باید زیرمجموعه یا مساوی ستون‌های دستور Group By باشد، بنابراین در فرم اصلاح شده‌ی گزینه‌ی چهارم، ستون STName جلوی دستور Group By قرار گرفت. نتیجه اینکه گزینه‌ی چهارم سه خطای نحوی داشت که اصلاح شد.

توجه: اگر به فرم‌های اصلاح شده گزینه اول و چهارم باشد، آنگاه هر دو گزینه‌ی اول و چهارم می‌توانند پاسخ سوال باشند.

توجه: گزینه‌ی اول، دو خطای نحوی دارد و گزینه‌ی چهارم، سه خطای نحوی دارد، اما طراح محترم گزینه‌ی اول را در کلید اولیه و نهایی اعلام کرده است. درحالی‌که هر دو گزینه‌ی اول و چهارم خطای نحوی دارد. اگر ملاک سنجش خطای کمتر است، آنگاه گزینه‌ی اول پاسخ سوال است. و اگر ملاک سنجش پاسخ واقعا درست است، این سوال پاسخ ندارد و باید سوال حذف می‌شد. و اگر ملاک سنجش عدالت است، آنگاه گزینه‌ی اول و چهارم باید پاسخ می‌بود. بنابراین اگر منظور طراح خطای کمتر باشد، آنگاه

گزینه‌ی اول پاسخ سؤال خواهد بود. همانطور که گفتیم سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه‌ی اول را به عنوان پاسخ اعلام کرده بود. که کار نادرستی بوده است. با توجه به جداول فوق، خروجی پرس و جوی گزینه‌ی چهارم پس از انجام عملگر JOIN به صورت زیر است:

STID	STNAME	STID	COID	SEMESTER	GRADE	COID	CONAME	CREDIT
S1	Sn1	S1	C1	95-96-1	20	C1	Cn1	2
S1	Sn1	S1	C2	95-96-1	15	C2	Cn2	3
S1	Sn1	S1	C3	92-92-1	17	C3	Cn3	4
S2	Sn2	S2	C1	95-96-1	19	C1	Cn1	2
S2	Sn2	S2	C2	95-96-1	18	C2	Cn2	3
S2	Sn2	S2	C3	93-93-1	12	C3	Cn3	4
S3	Sn3	S3	C1	94-94-1	16	C1	Cn1	2

در ادامه، پس از انجام دستور '95-96-1'=Semester Where فقط سطرهایی که شرط '95-96-1'=Semester را برقرار می‌کنند، انتخاب می‌شوند، به صورت زیر:

STID	STNAME	STID	COID	SEMESTER	GRADE	COID	CONAME	CREDIT
S1	Sn1	S1	C1	95-96-1	20	C1	Cn1	2
S1	Sn1	S1	C2	95-96-1	15	C2	Cn2	3
S2	Sn2	S2	C1	95-96-1	19	C1	Cn1	2
S2	Sn2	S2	C2	95-96-1	18	C2	Cn2	3

همچنین در ادامه، پس از انجام دستور Group By ST.STID,STName,Semester براساس ستون‌های ST.STID,STName,Semester خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:

S1,Sn1,95-96-1	S2,Sn2,95-96-1
C1 20 2 ...	C1 19 2 ...
C2 15 3	C2 18 3
گروه اول	گروه دوم

و در نهایت دستور SUM(Grade*Credit)/SUM(Credit) داخل دستور SELECT برای هر گروه به طور مستقل محاسبه می‌گردد و در خروجی پرس و جو قرار می‌گیرد.

معدل وزنی نمرات دروس اخذ شده گروه اول یعنی دانشجو Sn1 به صورت زیر محاسبه می‌گردد:

$$\frac{(20 \times 2) + (15 \times 3)}{5} = \frac{(40) + (45)}{5} = \frac{85}{5} = 17.00$$

معدل وزنی نمرات دروس اخذ شده گروه سوم یعنی دانشجو Sn2 به صورت زیر محاسبه می‌گردد:

$$\frac{(19 \times 2) + (18 \times 3)}{5} = \frac{(38) + (54)}{5} = \frac{92}{5} = 18.40$$

بنابراین خروجی نهایی پرس و جوی گزینه‌ی چهارم به صورت زیر است:

STName	(No column name)
Sn1	17.00
Sn2	18.40

توجه: دستور GROUP BY، سرگروه‌ها را، راهی خروجی می‌کند. که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی لیست نام دانشجویان به همراه معدل وزنی نمرات دروس اخذ شده آن‌ها را در ترم 95-96-1. توجه: گزینه‌ی چهارم را می‌توان بدون استفاده از دستور JOIN و به شکل کلاسیک آن یعنی با استفاده از ضرب دکارتی و شرط اتصال برابری نیز بازنویسی کرد.

فرم کلاسیک گزینه‌ی چهارم:

```
Select STName, SUM(grade*credit)/SUM(credit)
From ST , STCO, CO
Where ST.STID=STCO.STID and
STCO.COID=CO.COID and Semester='95-96-1'
Ggroup By ST.STID,STName,Semester
```

بنابراین خروجی نهایی فرم کلاسیک پرس و جوی گزینه‌ی چهارم نیز به صورت زیر است:

STName	(No column name)
Sn1	17.00
Sn2	18.40

۶۴- گزینه (۴) صحیح است.

سه جدول Student، Course و S_C با مقادیر زیر را در نظر بگیرید:

Sid	SName	Sid	Cid	grade	Cid	CName
s1	sn1	s1	c1	10	c1	cn1
s2	sn2	s1	c2	12	c2	cn2
s3	sn3	s2	c1	14	Course	
Student		s2	c2	16		
		s3	c1	18		
		s3	c2	20		
		S C				

مطابق پرس و جوی مطرح شده در گزینه‌ی اول، داریم:

```
Select SName
From Student T1,(Select S_C.Sid
From S_C
Group By S_C.Sid
Having AVG(grade) > (Select AVG(grade)
From S_C)) T2
```

Where T1.Sid=T2.Sid

با توجه به جداول فوق، ابتدا خروجی داخلی‌ترین زیر پرس و جوی داخلی بر اساس جدول S_C به صورت زیر محاسبه می‌گردد:

Sid	Cid	grade
s1	c1	10
s1	c2	12
s2	c1	14
s2	c2	16
s3	c1	18
s3	c2	20

Select AVG(grade)
From S_C

که مقدار آن برابر 15 می‌شود، به صورت زیر:

$$\frac{10+12+14+16+18+20}{2} = \frac{90}{6} = 15$$

همچنین در ادامه با توجه به جداول فوق، خروجی زیر پرس و جوی داخلی در جدول S_C به صورت زیر است:

Sid	Cid	grade
s1	c1	10
s1	c2	12
s2	c1	14
s2	c2	16
s3	c1	18
s3	c2	20

همچنین در ادامه، پس از انجام دستور Group By S_C.Sid براساس ستون S_C.Sid خروجی زیر پرس و جوی داخلی به صورت زیر گروه‌بندی می‌شود:

S1	S2	S3
C1 10	C1 14	C1 18
<u>C2 12</u>	<u>C2 16</u>	<u>C2 20</u>
گروه اول	گروه دوم	گروه سوم

و در نهایت دستور Having AVG(grade) > (Select AVG(grade) From S_C) می‌گردد.

توجه: دستور HAVING بر روی گروه‌ها، اعمال می‌گردد.

S1	S2	S3
C1 10	C1 14	C1 18
<u>C2 12</u>	<u>C2 16</u>	<u>C2 20</u>
گروه اول	گروه دوم	گروه سوم

$$\frac{10+12}{2} = 11 > 15, \quad \frac{14+16}{2} = 15 > 15, \quad \frac{18+20}{2} = 19 > 15$$

توجه: با توجه به شرط انتخاب گروه توسط دستور Having AVG(grade) > (Select AVG(grade) From S_C) فقط

گروه سوم جهت نمایش در خروجی انتخاب می‌شود.

و در نهایت دستور S_C.Sid داخل دستور SELECT برای هر گروه انتخاب شده توسط دستور Having به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد، حاصل نهایی زیر پرس و جوی داخلی یک جدول است که مطابق تعاریف زیر پرس و جوی داخلی T2 نامگذاری شده است، بنابراین خروجی

نهایی زیر پرس و جوی داخلی به صورت زیر است:

$$\frac{\text{Sid}}{\text{S3}}$$

توجه: دستور GROUP BY، سرگروه‌ها را، راهی خروجی می‌کند.

بنابراین در ادامه پرس و جوی زیر را خواهیم داشت:

```
Select SName
From Student T1, T2
Where T1.Sid=T2.Sid
```

با توجه به جداول فوق، خروجی نهایی پرس و جوی فوق پس از انجام عملگر ضرب دکارتی و شرط اتصال برابری یعنی سطرهای پیوندپذیر به صورت زیر است:

$$\frac{\text{SName}}{\text{sn3}}$$

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «نام دانشجویانی که معدل آنها از میانگین نمرات همه دروس دانشگاه بیشتر است.»

مطابق پرس و جوی مطرح شده در گزینه‌ی دوم، داریم:

```
Select SName
From Student T1
Where Exists (Select '1'
              From S_C
              Where T1.Sid=S_C.Sid
              Group by S_C.Sid
              Having AVG(grade) > (Select AVG(grade)
                                   From S_C))
```

با توجه به وجود دستور Exists، زیر پرس و جوی داخلی فوق یک Correlated Subquery است، یعنی به ازای حرکت در هریک از سطرهای پرس و جوی خارجی، یک بار به طور کامل از ابتدا تا انتها زیر پرس و جوی داخلی اجرا و بر اساس شرطی که زیر پرس و جوی داخلی را به پرس و جوی خارجی متصل می‌کند، بررسی انجام می‌شود. مانند دو حلقه تو در تو، که به ازای هربار اجرای حلقه خارجی، یک بار به طور کامل حلقه داخلی اجرا می‌گردد. توسط دستور where در فرم زیر:

```
SELECT SName
FROM Student T1
WHERE EXISTS (...)
```

برای هر سطر از جدول Student شرط جلوی Exists که حاصل یک مقایسه می‌باشد، محاسبه می‌گردد، اگر غیرتهی بود، شرط جلوی where که همان Exists است، TRUE می‌گردد و سطر مورد نظر از جدول Student انتخاب می‌گردد و این رویه برای تک تک سطرهای جدول Student، تا به انتهای جدول Student ادامه پیدا می‌کند. به بیان دیگر این پرس و جو نام دانشجویانی را می‌دهد که پرانتز مقابل Exists برای آنها غیرتهی است. این پرانتز هنگامی غیرتهی می‌شود که حاصل مقایسه بیان شده در این پرانتز غیرتهی شود. حاصل این مقایسه در صورتی غیرتهی می‌شود که معدل دانشجوی مورد نظر از میانگین نمرات همه دروس دانشگاه بیشتر باشد. به عبارت دیگر پرس و جوی گزینه‌ی دوم نام دانشجویانی که معدل آنها از میانگین

نمرات همه دروس دانشگاه بیشتر است را استخراج می‌کند. که مطابق پرس و جوی مطرح شده در صورت سوال است.

توجه: همان‌طور که مشاهده می‌کنید، معدل دانشجو Sn3، از میانگین نمرات همه دروس دانشگاه بیشتر است. به بیان دیگر عبارت موجود در جلوی دستور exists که به صورت زیر است:

```
Select AVG(grade)
From S_C
```

میانگین نمرات همه دروس دانشگاه را محاسبه می‌کند.

با توجه به جداول فوق داریم، که مقدار آن برابر 15 می‌شود، به صورت زیر:

$$\frac{10 + 12 + 14 + 16 + 18 + 20}{2} = \frac{90}{6} = 15$$

همچنین عبارت موجود در جلوی دستور Exists که به صورت زیر است:

```
Select '1'
From S_C
Where T1.Sid=S_C.Sid
Group by S_C.Sid
Having AVG(grade) > (Select AVG(grade)
From S_C)
```

توسط دستور T1.Sid = S_C.Sid به محیط خارج یعنی جدول Student متصل می‌گردد. حال به ازای حرکت در هر سطر از جدول Student، یک بار به طور کامل سطرهای جدول S_C از ابتدا تا انتها با توجه به شرط اتصال بررسی می‌گردد. مطابق شکل زیر:

ابتدا برای سطر اول از جدول Student با توجه به شرط اتصال T1.Sid = S_C.Sid داریم:

<u>Sid</u>	SName	<u>Sid</u>	<u>Cid</u>	grade	<u>Cid</u>	CName
s1	sn1	s1	c1	10	c1	cn1
s2	sn2	s1	c2	12	c2	cn2
s3	sn3	s2	c1	14	Course	
		s2	c2	16		
		s3	c1	18	Course	
		s3	c2	20		

S_C

همچنین در ادامه، پس از انجام دستور Group By S_C.Sid براساس ستون S_C.Sid خروجی زیر پرس و جوی داخلی به صورت زیر گروه‌بندی می‌شود:

```
SI
C1 10
C2 12
```


با توجه به جداول فوق، خروجی نهایی پرس و جوی فوق به صورت زیر است:

SName
sn3

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «نام دانشجویانی که معدل آنها از میانگین نمرات همه دروس دانشگاه بیشتر است.»

مطابق پرس و جوی مطرح شده در گزینه سوم، داریم:

```
Select SName
From Student
Where Sid in (Select Sid
              From S_C
              Group by sid
              Having avg(grade) > (Select AVG(grade)
                                   From S_C))
```

با توجه به وجود دستور in، زیر پرس و جوی داخلی فوق یک Normal Subquery است، یعنی ابتدا زیر پرس و جوی داخلی یک بار و برای همیشه اجرا می‌گردد، سپس پرس و جوی خارجی به ازای حرکت در هر یک از سطرهاى خود، از مقادیر زیر پرس و جوی داخلی استفاده می‌کند.

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول Student مجموعه جلوی in بررسی می‌گردد که آیا Sid داخل این مجموعه قرار دارد یا خیر. اگر قرار داشت سطر مورد نظر از جدول Student در خروجی نمایش داده می‌شود.

توسط دستور where در فرم زیر:

```
SELECT SName
FROM Student
WHERE in Sid (...)
```

برای هر سطر از جدول Student کل مجموعه‌ی جلوی in که حاصل یک مقایسه می‌باشد، بررسی می‌گردد، اگر Sid موجود در هر سطر، داخل Sidهای مجموعه جلوی in بود، آنگاه شرط جلوی where که همان in است، TRUE می‌گردد و سطر مورد نظر از جدول Student انتخاب می‌گردد و این رویه برای تک تک سطرهاى جدول Student، تا به انتهای جدول Student ادامه پیدا می‌کند. به بیان دیگر این پرس و جوی نام دانشجویانی از جدول Student را می‌دهد که در پراتر مقابل in قرار دارند. به عبارت دیگر پرس و جوی گزینه سوم نام دانشجویانی که معدل آنها از میانگین نمرات همه دروس دانشگاه بیشتر است را استخراج می‌کند. که مطابق پرس و جوی مطرح شده در صورت سوال است.

توجه: همان‌طور که مشاهده می‌کنید، معدل دانشجو Sn3، از میانگین نمرات همه دروس دانشگاه بیشتر است. با توجه به جداول فوق، ابتدا خروجی داخلی‌ترین زیر پرس و جوی داخلی بر اساس جدول S_C به صورت زیر محاسبه می‌گردد:

Sid	Cid	grade
s1	c1	10
s1	c2	12
s2	c1	14
s2	c2	16
s3	c1	18
s3	c2	20

Select AVG(grade)
From S_C

که مقدار آن برابر 15 می‌شود، به صورت زیر:

$$\frac{10+12+14+16+18+20}{2} = \frac{90}{6} = 15$$

همچنین در ادامه با توجه به جداول فوق، خروجی زیر پرس و جوی داخلی در جدول S_C به صورت زیر است:

Sid	Cid	grade
s1	c1	10
s1	c2	12
s2	c1	14
s2	c2	16
s3	c1	18
s3	c2	20

همچنین در ادامه، پس از انجام دستور Group By Sid براساس ستون Sid خروجی زیر پرس و جوی داخلی به صورت زیر گروه‌بندی می‌شود:

S1	S2	S3
C1 10	C1 14	C1 18
<u>C2 12</u>	<u>C2 16</u>	<u>C2 20</u>
گروه اول	گروه دوم	گروه سوم

و در نهایت دستور (Select AVG(grade) > Having AVG(grade) >) برای هر گروه به طور مستقل اعمال می‌گردد.

توجه: دستور HAVING بر روی گروه‌ها، اعمال می‌گردد.

S1	S2	S3
C1 10	C1 14	C1 18
<u>C2 12</u>	<u>C2 16</u>	<u>C2 20</u>
گروه اول	گروه دوم	گروه سوم

$$\frac{10+12}{2} = 11 > 15, \quad \frac{14+16}{2} = 15 > 15, \quad \frac{18+20}{2} = 19 > 15$$

توجه: با توجه به شرط انتخاب گروه توسط دستور (Having AVG(grade) > (Select AVG(grade) >))
From S_C)

فقط گروه سوم جهت نمایش در خروجی انتخاب می‌شود.

و در نهایت دستور Sid داخل دستور SELECT برای هر گروه انتخاب شده توسط دستور Having به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد، حاصل نهایی زیر پرس و جوی داخلی یک جدول یا مجموعه است، بنابراین خروجی نهایی زیر پرس و جوی داخلی به صورت زیر است:

Sid
s3

توجه: دستور GROUP BY، سرگروه‌ها را، راهی خروجی می‌کند. بنابراین در ادامه پرس و جوی زیر را خواهیم داشت:

```
Select SName
From Student
Where Sid in (S3)
```

همانطور که گفتیم برای هر سطر از جدول Student کل مجموعه‌ی جلوی in که حاصل یک مقایسه می‌باشد، بررسی می‌گردد، اگر Sid موجود در هر سطر، داخل Sidهای مجموعه جلوی in بود، آنگاه شرط جلوی where که همان in است، TRUE می‌گردد و سطر مورد نظر از جدول Student انتخاب می‌گردد و این رویه برای تک تک سطرهای جدول Student، تا به انتهای جدول Student ادامه پیدا می‌کند. به بیان دیگر این پرس و جو نام دانشجویانی از جدول Student را می‌دهد که در پراتز مقابل in قرار دارند. با توجه به جداول فوق، خروجی نهایی پرس و جوی فوق پس از انجام عملگر in به ازای هر سطر جدول Student به صورت زیر است:

SName
sn3

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «نام دانشجویانی که معدل آنها از میانگین نمرات همه دروس دانشگاه بیشتر است.»
مطابق پرس و جوی مطرح شده در گزینه‌ی چهارم، داریم:

```
Select SName
From Student,S_C
Where Student.Sid = S_C.Sid AND AVG(grade) > (Select AVG(grade)
From S_C)
```

در پرس و جوی گزینه‌ی چهارم تابع عددی AVG اول از سمت چپ در محل نادرست مورد استفاده قرار گرفته است، به طور کلی توابع عددی در جلوی Where باید داخل Select قرار بگیرند. بنابراین پرس و جوی گزینه‌ی چهارم دارای خطای نحوی است و از سوی کامپایلر اجرا نمی‌گردد. فرم اصلاح شده‌ی گزینه‌ی چهارم می‌تواند به فرم‌های گزینه‌های اول، دوم و یا سوم باشد.

۶۵- گزینه (۱ و ۴) صحیح است.

عملگر الحاق خارجی چپ در جبر رابطه‌ای

این عملگر، مانند الحاق طبیعی، ستون‌های مشترک را فقط یکبار در خروجی قرار می‌دهد. همچنین کلیه سطرهای پیوندپذیر را در خروجی قرار می‌دهد. اما علاوه بر آن کلیه سطرهای پیوندناپذیر جدول سمت چپ را نیز در خروجی قرار می‌دهد و در این حالت برای ستون‌های غیر مشترک جدول سمت راست مقدار NULL قرار می‌دهد.

عملگر الحاق خارجی چپ در SQL

این عملگر، ستون‌های مشترک را دوبار در خروجی قرار می‌دهد. همچنین کلیه سطرهای پیوندپذیر را در خروجی قرار می‌دهد. اما علاوه بر آن کلیه سطرهای پیوندناپذیر جدول سمت چپ را نیز در خروجی قرار

می‌دهد و در این حالت برای تمام ستون‌های جدول سمت راست مقدار NULL قرار می‌دهد.
توجه: به تفاوت عملگر الحاق خارجی چپ در جبر رابطه‌ای و SQL دقت کنید.
 برای فرم **حداقل** جداول زیر را در نظر بگیرید:

Stid	StName	...	Stid	Cid	grade	Cid	CName
s1	sn1		s1	c1	10	c1	cn1
s2	sn2		StudentCourse			c2	cn2
s3	sn3		M>0			Course	
Student						N>0	
K>0							

توجه: مطابق فرض سؤال، تعداد سطرهای جداول باید بیشتر از صفر باشد، یعنی جداول تهی نباشند.
 مطابق پرس و جوی مطرح شده در صورت سؤال، داریم:

```
select *
from student left outer join studentCourse
```

که البته پرس و جوی مطرح شده در صورت سؤال کمی خطای نحوی دارد، که فرم اصلاح شده آن به صورت زیر است:

```
select *
from student left outer join studentCourse
on student.stid= studentCourse.Stid
```

توجه: در فرم اصلاح شده‌ی پرس و جوی مطرح شده در صورت سؤال، شرط اتصال و عملگر ON قرار گرفت. پرس و جوی مطرح شده در صورت سؤال یک خطای نحوی داشت که اصلاح شد.
 با توجه به جداول فوق، خروجی پرس و جوی مطرح شده در صورت سؤال در حالت **حداقل** پس از انجام عملگر **Left Outer Join** به صورت زیر است:

Stid	StName	Stid	Cid	grade
s1	sn1	s1	c1	10
s2	sn2	NULL	NULL	NULL
s3	sn3	NULL	NULL	NULL

بنابراین رابطه زیر برقرار خواهد بود:

$$\text{Cardinality}_{\min}(\text{student left outer join studentCourse}) = K = 3$$

توجه: دقت کنید که در خروجی فوق ستون مشترک stid، دو بار در خروجی قرار گرفت.
 برای فرم **حداکثر** جداول زیر را در نظر بگیرید:

Stid	StName	...	Stid	Cid	grade	Cid	CName
s1	sn1		s1	c1	10	c1	cn1
s2	sn2		s1	c2	20	c2	cn2
s3	sn3		s2	c1	30	Course	
Student			s2	c2	40	N>0	
K>0			s3	c1	50		
			s3	c2	60		
			StudentCourse				
			M>0				

با توجه به جداول فوق، خروجی پرس و جوی مطرح شده در صورت سوال در حالت حداکثر پس از انجام عملگر Left Outer Join به صورت زیر است:

Stid	StName	Stid	Cid	grade
s1	sn1	s1	c1	10
s1	sn1	s1	c2	20
s2	sn2	s2	c1	30
s2	sn2	s2	c2	40
s3	sn3	s3	c1	50
s3	sn3	s3	c2	60

بنابراین رابطه زیر برقرار خواهد بود:

$$\text{Cardinality}_{\max}(\text{student left outer join studentCourse}) = K \times N = 3 \times 2 = 6$$

توجه: سازمان سنجش آموزش کشور، در کلید اولیه خود، ابتدا گزینه اول را به عنوان پاسخ اعلام نمود، سپس در کلید نهایی نظر خود را عوض کرد و گزینه اول و چهارم را به عنوان پاسخ با تأثیر مثبت اعلام کرد، که عمل نادرستی را انجام داده است.

توجه: در کلید نهایی این سؤال به دو دلیل واضح باید حذف می‌شد، دلیل اول وجود خطای کامپایلری در صورت سوال و دلیل دوم عدم وجود گزینه صحیح در گزینه‌های صورت سوال.

۶۶- گزینه (۴) صحیح است.

CROSS JOIN (ضرب دکارتی)

فرم اول: `select *`

from Student **CROSS JOIN** StudentCourse

فرم دوم:

`select *`
from Student, StudentCourse

توجه: در CROSS JOIN همه سطرهای دو جدول در کنار هم قرار می‌گیرند.

برای فرم **حداقل**، جداول زیر را در نظر بگیرید:

<u>Stid</u>	StName	...	<u>Stid</u>	<u>Cid</u>	grade	<u>Cid</u>	CName
s1	sn1					c1	cn1
s2	sn2					c2	cn2
s3	sn3						

Student

<u>Stid</u>	<u>Cid</u>	grade

StudentCourse

<u>Cid</u>	CName

Course

توجه: مطابق فرض سؤال، الزامی بر تعداد سطرهای جداول وجود ندارد، یعنی تعداد سطرهای جداول می‌تواند صفر (تهی) یا بیشتر (غیرتهی) باشد.

خروجی پرس و جوی مطرح شده در صورت سوال در حالت **حداقل** به صورت زیر است:

`select distinct Student.*`
from student, studentCourse

<u>Stid</u>	StName	...

بنابراین رابطه زیر برقرار خواهد بود:

$$\text{Cardinality}_{\min}(\text{student}, \text{studentCourse}) = 0$$

برای فرم حداکثر جداول زیر را در نظر بگیرید:

<u>Stid</u>	StName	...	<u>Stid</u>	<u>Cid</u>	grade	<u>Cid</u>	CName
s1	sn1		s1	c1	10	c1	cn1
s2	sn2		s1	c2	20	c2	cn2
s3	sn3		s2	b1	30	Course	
Student			s2	b2	40		
			s3	b1	50		
			s3	B2	60		
StudentCourse							

خروجی پرس و جوی مطرح شده در صورت سوال در حالت حداکثر به صورت زیر است:

```
select distinct Student.*
from student, studentCourse
```

<u>Stid</u>	StName	...
s1	sn1	
s2	sn2	
s3	sn3	

توجه: در SQL هنگام استفاده از عملگر `distinct`، رکوردهای تکراری فقط یک بار در خروجی ظاهر می‌شوند.

$$\text{Cardinality}_{\max}(\text{student}, \text{studentCourse}) = K = 3$$

بنابراین رابطه مقابل برقرار خواهد بود:

۶۷- گزینه (۳) صحیح است.

صورت سوال به این شکل است:

با توجه به گزاره‌های داده شده، کدام مورد درست است؟

(a) اگر رابطه R ، دارای n خصیصه باشد، آنگاه تعداد ابرکلیدهای آن حداقل یک و حداکثر $2^n - 1$ است.

گزاره اول درست است، زیرا در حالت کلی، اگر رابطه R ، دارای n خصیصه باشد، آنگاه تعداد ابرکلیدهای آن حداقل یک و حداکثر $2^n - 1$ است. در جدول تمام کلید، یک جدول فقط و فقط یک ابرکلید دارد و فقط هم یک کلید کاندید دارد. رابطه تمام کلید مثلاً ممکن است سه ستون داشته باشد، در این حالت یک جدول فقط و فقط یک ابرکلید دارد و فقط و فقط هم یک کلید کاندید دارد. یعنی حداقل یک ابرکلید و یک کلید کاندید دارد و حداکثر هم یک ابرکلید و یک کلید کاندید دارد. در گزاره اول حالت کلی مورد بررسی قرار گرفته است. دقت کنید که حداقل یک ابرکلید و حداکثر $2^n - 1$ ابرکلید، حالت خاص جدول

تمام کلید هم پوشش می‌دهد چون بیان حداکثر $2^n - 1$ مقادیر کوچکتر و برابر خودش را پوشش می‌دهد. اگر در یک رابطه با n خصیصه، تک تک خصیصه‌ها به تنهایی کلید کاندید باشد، آنگاه رابطه دارای n کلید کاندید است. بنابراین هر زیرمجموعه غیر تهی از خصیصه‌های این رابطه یک ابرکلید است. که در این حالت تعداد ابرکلیدهای یک رابطه با n خصیصه برابر با $2^n - 1$ است که بیشترین مقدار ممکن در تعداد ابرکلیدهای یک رابطه با n خصیصه است. ابرکلید بدون صفت نداریم، بنابراین حالت $\binom{n}{0}$ اضافه است. همانطور که گفتیم در حالت کلی، یک رابطه دارای n خصیصه، شرایط مختلفی را در تعداد ابرکلید می‌تواند تجربه کند، که حداکثر تعداد ابرکلیدهایی که می‌تواند تجربه کند برابر $\binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n}$ یا $2^n - 1$ است. در واقع عبارت زیر برقرار است:

$$2^n - 1 = \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n}$$

(b) صفت مرکب، صفتی است که مقدار آن از مقدار سایر صفت‌ها محاسبه می‌شود.

گزاره دوم نادرست است، زیرا صفت مشتق (پویا) صفتی است که مقدار آن از مقدار سایر صفت‌ها محاسبه می‌شود. صفت مشتق (پویا) صفتی است که در موجودیت وجود خارجی ندارد ولی در صورت لزوم می‌توان آنرا بدست آورد. صفتی که مقادیر آن مدام در حال تغییر و تحول باشد، صفت پویا یا مشتق محسوب می‌گردد. بنابراین به دلیل تغییرات مداوم، توصیه می‌گردد صفت پویا در جداول بانک اطلاعات مورد استفاده قرار نگیرد و مقدار آن از طریق صفت مرتبط با آن محاسبه گردد. برای مثال برای محاسبه صفت سن، می‌توان صفت تاریخ تولد را در نظر گرفت و از روی این صفت، سن را محاسبه نمود. همچنین صفت ساده صفتی است که مقدار آن از لحاظ معنایی ساده یا اتومیک یا تجزیه نشدنی باشد، به این معنا که اگر مقدار آن را به اجزایی تجزیه کنیم، مقادیر جزئی حاصله فاقد معنا باشد. برای مثال صفت درس و شماره دانشجویی یک صفت ساده است. صفت مرکب صفتی است که از چند صفت ساده تشکیل شده باشد به گونه‌ای که تجزیه شدنی باشند و اجزاء حاصله خود صفات ساده باشند. مانند صفت آدرس که از اجزاء نام استان، نام شهر، نام خیابان، نام کوچه، شماره پلاک و کدپستی تشکیل شده است.

(c) تعداد تاپل‌های عبارت $\Pi_{R,a}(R \times T)$ همواره برابر با تعداد سطرهای R است.

گزاره سوم نادرست است. جداول R و T را به صورت زیر در نظر بگیرید:

a	b	c	d	e
1	2			
3	4			

R

c	d	e

T

$\Pi_{R,a}(R \times T)$

پرس و جوی مقابل را در نظر بگیرید:

خروجی پرس و جوی پرانتز داخلی به صورت زیر است:

a	b	c	d	e
1	2			
3	4			

×

c	d	e

=

a	b	c	d	e

که در نهایت پس از حرکت به سمت خارج و انجام عملگر پرتو، خروجی نهایی پرس و جو به صورت زیر خواهد بود:

<u>a</u>
تهی

توجه: در جبر رابطه‌ای هنگام استفاده از عملگر Π ، تاپل‌های تکراری فقط یک بار در خروجی ظاهر می‌شوند.

همچنین پرس و جوی زیر را در نظر بگیرید:

Select distinct R.a
From R

خروجی پرس و جو به صورت زیر است:

<u>a</u>
1
3

توجه: در SQL هنگام استفاده از عملگر distinct، رکوردهای تکراری فقط یک بار در خروجی ظاهر می‌شوند.

$\Pi_{R,\alpha}(R \times T) \neq \text{Select distinct } \alpha \text{ From } R$

بنابراین رابطه مقابل برقرار خواهد بود:

$$\sigma_c(\Pi_\alpha(R)) = \Pi_\alpha(\sigma_c(R)) \quad (d)$$

گزاره چهارم نادرست است، زیرا دو عملگر Π و σ به صورت مشروط دارای خاصیت جابه‌جایی هستند. به طور کلی اگر R یک رابطه، α زیر مجموعه‌ای از ستون‌ها و c مجموعه‌ای از شروط بر روی سطرها باشد، آنگاه تساوی زیر زمانی برقرار است که ستون‌های عملگر σ زیر مجموعه ستون‌های عملگر Π باشد. یعنی $\sigma \subseteq \Pi$

$$\Pi_\alpha(\sigma_c(R)) = \sigma_c(\Pi_\alpha(R))$$

مثال: جدول S را به صورت زیر در نظر بگیرید:

<u>S#</u>	Sname	City
S1	Sn1	C1
S2	Sn2	C2
S3	Sn3	C2

$$R_2 = \Pi_{S\#,City}(\sigma_{City='C2'}(S))$$

پرس و جوی مقابل را در نظر بگیرید:

خروجی پرس و جوی پرانتز داخلی به صورت زیر است:

<u>S#</u>	Sname	City
S2	Sn2	C2
S3	Sn3	C2

که در نهایت پس از حرکت به سمت خارج و انجام عملگر پرتو، خروجی نهایی پرس و جو به صورت زیر خواهد بود:

<u>S#</u>	City
S2	C2
S3	C2

همچنین پرس و جوی زیر را در نظر بگیرید:

$$R_2 = \sigma_{City='C2'}(\Pi_{S\#,City}(S))$$

خروجی پرس و جوی پراتنز داخلی به صورت زیر است:

S#	City
S1	C1
S2	C2
S3	C2

که در نهایت پس از حرکت به سمت خارج و انجام عملگر انتخاب، خروجی نهایی پرس و جو به صورت زیر خواهد بود:

S#	City
S2	C2
S3	C2

$$\Pi_{S\#,City}(\sigma_{City='C2'}(S)) = \sigma_{City='C2'}(\Pi_{S\#,City}(S))$$

پس رابطه مقابل برقرار است:

مثال: جدول S را به صورت زیر در نظر بگیرید:

S#	Sname	City
S1	Sn1	C1
S2	Sn2	C2
S3	Sn3	C2

$$R_2 = \Pi_{S\#}(\sigma_{City='C2'}(S))$$

پرس و جوی مقابل را در نظر بگیرید:

خروجی پرس و جوی پراتنز داخلی به صورت زیر است:

S#	Sname	City
S2	Sn2	C2
S3	Sn3	C2

که در نهایت پس از حرکت به سمت خارج و انجام عملگر پرتو، خروجی نهایی پرس و جو به صورت زیر خواهد بود:

S#
S2
S3

$$R_2 = \sigma_{City='C2'}(\Pi_{S\#}(S))$$

همچنین پرس و جوی مقابل را در نظر بگیرید:

خروجی پرس و جوی پراتنز داخلی به صورت زیر است:

S#
S1
S2
S3

که در نهایت پس از حرکت به سمت خارج، انجام عملگر انتخاب امکان پذیر نخواهد بود، زیرا ستون‌های عملگر σ یعنی City زیر مجموعه ستون‌های عملگر Π یعنی S# نیست، یعنی $City \not\subseteq S\#$. در واقع ستون City در پراتنز داخلی توسط عملگر Π انتخاب نشده است، بنابراین اجرای عملگر σ بر روی ستون City امکان‌پذیر نیست.

پس رابطه زیر برقرار است:

$$\Pi_{S\#}(\sigma_{City='C2'}(S)) \neq \sigma_{City='C2'}(\Pi_{S\#}(S))$$

۶۸- گزینه (۱) صحیح است.

سه جدول USER، BOOK و BORROWING با مقادیر زیر را در نظر بگیرید:

UID	NAME	BID	TITLE	PUBLISHER	TOTALNUMBER
U1	Un1			B1	Tn1	Pub1	1
U2	Un2			B2	Tn2	Pub2	2
U3	Un3			B3	Tn3	Pub3	3
U4	Un4						
U5	UN5						

جدول USER

جدول BOOK

UID	BID	StartDate	EndDate
U1	B1	95-01-1	NULL
U2	B2	95-02-1	NULL
U3	B2	95-03-1	NULL
U4	B3	95-04-1	NULL
U5	B3	95-05-1	NULL

جدول BORROWING

مطابق پرس و جوی مطرح شده در گزاره‌ی a، داریم:

a) with tbl as (
 Select BORROWING.BID, COUNT(BORROWING.BID) as cnt
 From BORROWING, BOOK
 Where BOOK.BID = BORROWING.BID and EndDate is NULL
 Group by BORROWING.BID)
 Select tbl.BID
 From tbl, BOOK
 Where tbl.BID = BOOK.BID and BOOK.TotalNumber = cnt

توجه: دستور with کل پرس و جوی مقابلش را tbl نام گذاری می‌کند. که در From دوم به طور خلاصه‌تر، ساده‌تر و خواناتر مورد استفاده قرار می‌گیرد.

با توجه به جداول فوق، خروجی بخش اول پرس و جوی گزاره‌ی a پس از انجام عملگر ضرب دکارتی به صورت زیر است:

UID	BID	StartDate	EndDate	BID	TITLE	PUBLISHER	TOTALNUMBER
U1	B1	95-01-1	NULL	B1	Tn1	Pub1	1
U2	B2	95-02-1	NULL	B2	Tn2	Pub2	2
U3	B2	95-03-1	NULL	B2	Tn2	Pub2	2
U4	B3	95-04-1	NULL	B3	Tn3	Pub3	3
U5	B3	95-05-1	NULL	B3	Tn3	Pub3	3

همچنین در ادامه، پس از انجام دستور `GROUP BY BORROWING.BID` براساس ستون `BORROWING.BID` خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:

<p>B1</p> <p><u>U1 95-01-1 Tn1 ...</u></p> <p style="text-align: center;">گروه اول</p>	<p>B2</p> <p>U2 95-02-1 Tn2 ...</p> <p><u>U3 95-03-1 Tn2 ...</u></p> <p style="text-align: center;">گروه دوم</p>
<p>B3</p> <p>U4 95-04-1 Tn3 ...</p> <p><u>U5 95-05-1 Tn3 ...</u></p> <p style="text-align: center;">گروه سوم</p>	

در ادامه دستور `COUNT(BORROWING.BID)` داخل دستور `SELECT` برای هر گروه به طور مستقل محاسبه می‌گردد و در خروجی پرس و جو قرار می‌گیرد. بنابراین خروجی نهایی بخش اول پرس و جوی گزاره‌ی `a` به صورت زیر است:

BID	cnt
B1	1
B2	2
B3	2

توجه: دستور `GROUP BY`، سرگروه‌ها را، راهی خروجی می‌کند.

توجه: نتیجه اینکه خروجی بخش اول پرس و جوی گزاره‌ی `a` پس از انجام عملگر ضرب دکارتی و اجرای دستور `GROUP BY` شامل دو ستون کد کتاب (`BID`) و تعداد نسخه‌های به امانت رفته کتاب است که توسط دستور `with` به جدول `tb1` نام‌گذاری شده است. بخش دوم پرس و جوی گزاره‌ی `a` را نظر بگیرید:

```
Select tb1.BID
From tb1,BOOK
Where tb1.BID = BOOK.BID and BOOK.TotalNumber = cnt
```

با توجه به جداول فوق، خروجی بخش دوم پرس و جوی گزاره‌ی `a` پس از انجام عملگر ضرب دکارتی به صورت زیر است:

BID	cnt	BID	TITLE	PUBLISHER	TOTALNUMBER
B1	1	B1	Tn1	Pub1	1
B2	2	B2	Tn2	Pub2	2
B3	2	B3	Tn3	Pub3	3

توجه: پس از انجام عمل ضرب دکارتی مابین جدول `tb1` و جدول `BOOK`، شرط اصلی پرس و جو اجرا می‌گردد که منجر به استخراج کد کتاب‌هایی می‌گردد که تعداد نسخه‌های به امانت رفته آنها برابر با تعداد

کل نسخه‌های آن کتاب است. در واقع لیست تمام کتاب‌هایی را نمایش می‌دهد که تمام نسخه‌های آنها امانت گرفته شده است.

در ادامه پس از اجرای شرط اصلی پرس و جو یعنی `BOOK.TotalNumber = cnt` خروجی پرس و جو به صورت زیر خواهد بود:

BID	cnt	BID	TITLE	PUBLISHER	TOTALNUMBER
B1	1	B1	Tn1	Pub1	1
B2	2	B2	Tn2	Pub2	2

و در نهایت پس از اجرای دستور `Select tbl.BID` خروجی پرس و جو به صورت زیر خواهد بود:

BID
B1
B2

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «لیست تمام کتاب‌هایی را نمایش می‌دهد که تمام نسخه‌های آنها امانت گرفته شده است.»

مطابق پرس و جوی مطرح شده در گزاره b، داریم:

b) Select BID

From BOOK b1

Where (select COUNT (UID)

From BORROWING

Where BORROWING.BID = b1.BID and

BORROWING.EndDate is NULL) = b1.TotalNumber

با توجه به وجود عملگر تساوی، زیر پرس و جوی داخلی فوق یک `Correlated Subquery` است، یعنی به ازای حرکت در هریک از سطرها پرس و جوی خارجی، یک بار به طور کامل از ابتدا تا انتها زیر پرس و جوی داخلی اجرا و بر اساس شرطی که زیر پرس و جوی داخلی را به پرس و جوی خارجی متصل می‌کند، بررسی انجام می‌شود. مانند دو حلقه تو در تو، که به ازای هریک از حلقه خارجی، یک بار به طور کامل حلقه داخلی اجرا می‌گردد.

SELECT BID

توسط دستور where در فرم زیر:

FROM Book b1

WHERE (...)

برای هر سطر از جدول Book شرط تساوی که حاصل یک مقایسه می‌باشد، محاسبه می‌گردد، اگر برابر بود، شرط جلوی where که همان تساوی است، TRUE می‌گردد و سطر مورد نظر از جدول Book انتخاب می‌گردد و این رویه برای تک تک سطرها جدول Book، تا به انتهای جدول Book ادامه پیدا می‌کند. به بیان دیگر این پرس و جو شماره کتاب‌هایی را می‌دهد که پراتنژ مقابل where برای آنها TRUE است. این پراتنژ هنگامی TRUE می‌شود که حاصل مقایسه بیان شده در این پراتنژ برابر شود. حاصل این مقایسه در صورتی برابر می‌شود که تعداد نسخه‌های به امانت رفته کتاب مورد بررسی برابر کل نسخه‌های کتاب مورد بررسی باشد. به عبارت دیگر پرس و جوی گزاره b لیست تمام کتاب‌هایی را نمایش می‌دهد که تمام نسخه‌های آنها امانت گرفته شده است. که مطابق پرس و جوی مطرح شده در صورت سوال است.

به بیان دیگر عبارت موجود در جلوی دستور where که به صورت زیر است:

```
Select BID
From BOOK b1
Where (select COUNT (UID)
      From BORROWING
      Where BORROWING.BID = b1.BID and
            BORROWING.EndDate is NULL) = b1.TotalNumber
```

توسط دستور count (UID) تعداد کتاب‌های به امانت رفته را می‌شمارد و محاسبه می‌کند. همچنین عبارت موجود در جلوی دستور where که به صورت زیر است:

```
Select BID
From BOOK b1
Where (select COUNT (UID)
      From BORROWING
      Where BORROWING.BID = b1.BID and
            BORROWING.EndDate is NULL) = b1.TotalNumber
```

توسط دستور BORROWING.BID = b1.BID به محیط خارج یعنی جدول Book متصل می‌گردد. حال به ازای حرکت در هر سطر از جدول Book، یک بار به طور کامل سطرهای جدول BORROWING از ابتدا تا انتها با توجه به شرط اتصال بررسی می‌گردد. مطابق شکل زیر:

ابتدا برای سطر اول از جدول Book با توجه به شرط اتصال BORROWING.BID = b1.BID داریم:

B1	B1	UID	BID	...	EndDate	UID	...
B1	1	U1	B1		NULL	U1	
B2	2	U2	B2		NULL	U2	
B3	3	U3	B2		NULL	U3	
BOOK		U4	B3		NULL	U4	
		U5	B3		NULL	U5	
		BORROWING			USER		

که پرس و جوی زیر برای آن اجرا می‌شود:

```
Select COUNT (UID)
From BORROWING
Where BORROWING.BID = b1.BID and
      BORROWING.EndDate is NULL
```

توجه: مقدار count (UID) برای سطر اول جدول Book یعنی B1 برابر مقدار 1 است. که رابطه‌ی برابری برای آن برقرار است، به صورت زیر:

$$\frac{\text{count (UID)}}{1} = \frac{\text{b1.TotalNumber}}{1}$$

شرط تساوی جلوی where برقرار است، بنابراین شرط where TRUE می‌گردد. بنابراین سطر اول از جدول Book در خروجی نمایش داده می‌شود.

بنابراین کد کتاب **B1** جهت نمایش در خروجی انتخاب می‌شود.

حال برای سطر دوم از جدول Book با توجه به شرط اتصال $BORROWING.BID = b1.BID$ داریم:

B2			B2			B2		
<u>BID</u>	...	TOTALNUMBER	<u>UID</u>	<u>BID</u>	...	EndDate	<u>UID</u>	...
B1		1	U1	B1		NULL	U1	
B2		2	U2	B2		NULL	U2	
B3		3	U3	B2		NULL	U3	
BOOK			U4	B3		NULL	U4	
			U5	B3		NULL	U5	
			BORROWING			USER		

که پرس و جوی زیر برای آن اجرا می‌شود:

```
Select COUNT (UID)
From BORROWING
Where BORROWING.BID = b1.BID and
BORROWING.EndDate is NULL
```

توجه: مقدار count (UID) برای سطر دوم جدول Book یعنی B2 برابر مقدار 2 است. که رابطه‌ی برابری برای آن برقرار است، به صورت زیر:

$$\frac{\text{count (UID)}}{2} = \frac{b1.TotalNumber}{2}$$

شرط تساوی جلوی where برقرار است، بنابراین شرط where TRUE می‌گردد. بنابراین سطر دوم از جدول Book در خروجی نمایش داده می‌شود.

بنابراین کد کتاب **B2** جهت نمایش در خروجی انتخاب می‌شود.

حال برای سطر سوم از جدول Book با توجه به شرط اتصال $BORROWING.BID = b1.BID$ داریم:

B3			B3			B3		
<u>BID</u>	...	TOTALNUMBER	<u>UID</u>	<u>BID</u>	...	EndDate	<u>UID</u>	...
B1		1	U1	B1		NULL	U1	
B2		2	U2	B2		NULL	U2	
B3		3	U3	B2		NULL	U3	
BOOK			U4	B3		NULL	U4	
			U5	B3		NULL	U5	
			BORROWING			USER		

که پرس و جوی زیر برای آن اجرا می‌شود:

```
Select count (UID)
From BORROWING
Where BORROWING.BID = b1.BID and
BORROWING.EndDate is NULL
```

توجه: مقدار count (UID) برای سطر سوم جدول Book یعنی B3 برابر مقدار 2 است. که رابطه‌ی برابری برای آن برقرار نیست، به صورت زیر:

$$\frac{\text{count}(UID)}{2} \neq \frac{b1.TotalNumber}{3}$$

شرط تساوی جلوی where برقرار نیست، بنابراین شرط where FALSE می‌گردد. بنابراین سطر سوم از جدول Book در خروجی نمایش داده نمی‌شود.

بنابراین کد کتاب B3 جهت نمایش در خروجی انتخاب نمی‌شود.

و در نهایت پس از اجرای کامل دستور Select BID خروجی پرس و جو به صورت زیر خواهد بود:

BID
B1
B2

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «لیست تمام کتاب‌هایی را نمایش می‌هد که تمام نسخه‌های آنها امانت گرفته شده است.»
مطابق پرس و جوی مطرح شده در گزاره c، داریم:

```
c) Select BID, TotalNumber
From BORROWING, BOOK
Where BOOK.BID = BORROWING.BID and EndDate is NULL
Group by BORROWING.BID
Having COUNT(BID) = TotalNumber
```

که البته پرس و جوی مطرح شده در گزاره c کمی خطای نحوی دارد، که فرم اصلاح شده آن به صورت زیر است:

```
Select BORROWING.BID, TotalNumber
From BORROWING, BOOK
Where BOOK.BID = BORROWING.BID and EndDate is NULL
Group by BORROWING.BID, TotalNumber
Having COUNT(BORROWING.BID) = TotalNumber
```

توجه: در فرم اصلاح شده‌ی گزاره c، نام جدول BORROWING پشت ستون BID در دستور select قرار گرفت، که کامپایلر متوجه شود ستون BID جدول BOOK یا BORROWING را انتخاب کند، پرس و جوی گزاره c تا به اینجا یک خطای نحوی داشت که اصلاح شد. اما یک خطای نحوی پررنگ‌تری هم دارد، اینکه همواره به غیر از توابع آماری، همه ستون‌های جلوی دستور Select باید زیرمجموعه یا مساوی ستون‌های دستور Group By باشد، بنابراین در فرم اصلاح شده‌ی گزاره c، ستون TotalNumber جلوی

دستور Group By قرار گرفت. گزاره c تا به اینجا دو خطای نحوی داشت که اصلاح شد. همچنین نام جدول BORROWING پشت ستون BID در دستور having قرار گرفت، که کامپایلر متوجه شود ستون BID جدول BOOK یا BORROWING را انتخاب کند، نتیجه اینکه پرس و جوی گزاره c سه خطای نحوی داشت که اصلاح شد.

با توجه به جداول فوق، خروجی بخش اول پرس و جوی گزاره ی C پس از انجام عملگر ضرب دکارتی به صورت زیر است:

UID	BID	StartDate	EndDate	BID	TITLE	PUBLISHER	TOTALNUMBER
U1	B1	95-01-1	NULL	B1	Tn1	Pub1	1
U2	B2	95-02-1	NULL	B2	Tn2	Pub2	2
U3	B2	95-03-1	NULL	B2	Tn2	Pub2	2
U4	B3	95-04-1	NULL	B3	Tn3	Pub3	3
U5	B3	95-05-1	NULL	B3	Tn3	Pub3	3

همچنین در ادامه، پس از انجام دستور TotalNumber , Group By BORROWING.BID براساس ستونهای BORROWING.BID و TotalNumber خروجی پرس و جو به صورت زیر گروه بندی می شود:

B1, 1	B2, 2
<u>U1 95-01-1 NULL Tn1 Pub1</u>	<u>U2 95-02-1 NULL Tn2 Pub2</u>
گروه اول	<u>U3 95-03-1 NULL Tn2 Pub2</u>
	گروه دوم
B3, 3	
<u>U4 95-04-1 NULL Tn3 Pub3</u>	
<u>U5 95-05-1 NULL Tn3 Pub3</u>	
گروه سوم	

و در نهایت دستور $\text{having count(BORROWING.BID) = TotalNumber}$ برای هر گروه به طور مستقل اعمال می گردد.

توجه: دستور HAVING بر روی گروه ها، اعمال می گردد.

B1, 1	B2, 2
<u>U1 95-01-1 NULL Tn1 Pub1</u>	<u>U2 95-02-1 NULL Tn2 Pub2</u>
گروه اول	<u>U3 95-03-1 NULL Tn2 Pub2</u>
$1 = 1,$	گروه دوم
	$2 = 2$
B3, 3	
<u>U4 95-04-1 NULL Tn3 Pub3</u>	
<u>U5 95-05-1 NULL Tn3 Pub3</u>	
گروه سوم	
$2 \neq 3$	

توجه: با توجه به شرط انتخاب گروه توسط دستور $\text{having count(BORROWING.BID) = TotalNumber}$ ، فقط گروه اول و دوم جهت نمایش در خروجی انتخاب می‌شود. و در نهایت دستور $\text{BORROWING.BID, TotalNumber}$ داخل دستور select برای هر گروه انتخاب شده توسط دستور Having به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد، بنابراین خروجی نهایی پرس و جو به صورت زیر است:

BID	TOTALNUMBER
B1	1
B2	2

توجه: دستور GROUP BY ، سرگروه‌ها را، راهی خروجی می‌کند.

که مطابق پرس و جو مطرح شده در صورت سوال نیست. یعنی «لیست تمام کتاب‌هایی را نمایش می‌هد که تمام نسخه‌های آنها امانت گرفته شده است.» در واقع پرس و جو گزاره C به دو دلیل مطابق پرس و جو مطرح شده در صورت سوال نیست؛ اول اینکه خطای نحوی داشت که جهت اجرا می‌بایست اصلاح می‌شد که ما به فرم اصلاح شده آنرا بررسی کردیم، هرچند که در My-SQL این فرم خطای نحوی صادر نمی‌شود. دوم اینکه در صورت سوال صرفاً شماره کتاب‌هایی خواسته شده که همه نسخه‌های آنها به امانت گرفته شده است، درحالی‌که در این پرس و جو علاوه بر شماره کتاب (BID)، تعداد کل نسخه‌های هر کتاب هم در خروجی ظاهر می‌شود که این برخلاف صورت سوال است. بنابراین فقط و فقط گزاره‌های a و b مطابق پرس و جو صورت سوال است که به تبع گزینه‌های دوم، سوم و چهارم را به طور کامل کنار می‌گذاریم، پس پرواضح است که گزینه اول پاسخ سوال است.

۶۹- گزینه (۲) صحیح است.

چهار جدول Person ، Car ، Accident و Owns با مقادیر زیر را در نظر بگیرید:

<u>SSN</u>	name	address	<u>License</u>	year	model

جدول Person

جدول Car

<u>License</u>	<u>accident-date</u>	driver	damage-amount
L1	95-01-1	d1	10
L2	95-02-1	d2	20
L3	95-03-1	d3	30

جدول Accident

<u>SSN</u>	<u>License</u>

جدول Owns

توجه: دقت کنید که مطابق فرض سوال، ترکیب دو ستون License و accident-date هر دو باهم به عنوان کلید کاندید جدول Accident مشخص شده است، بنابراین نباید ترکیب این دو ستون باهم سطرهای تکراری داشته باشد، پس یک خودرو (License) در یک تاریخ تصادف (accident-date) نمی‌تواند بیش از یکبار تکرار شود، یعنی نباید دوبار و در دو سطر و بیشتر تکرار شود.

مطابق پرس و جوی مطرح شده در گزینه‌ی اول، داریم:

```
(select driver, damage – amount
from Accident) Except (select a.driver, a.damage – amount
from Accident a, Accident b
where a.damage – amount < b.damage – amount and a.driver <> b.driver
```

با توجه به جداول فوق، خروجی بخش دوم پرس و جوی گزینه‌ی اول پس از انجام عملگر ضرب دکارتی به صورت زیر است:

License	accident-date	driver	damage-amount	License	accident-date	driver	damage-amount
L1	95-01-1	d1	10	L1	95-01-1	d1	10
L1	95-01-1	d1	10	L2	95-02-1	d2	20
L1	95-01-1	d1	10	L3	95-03-1	d3	30
L2	95-02-1	d2	20	L1	95-01-1	d1	10
L2	95-02-1	d2	20	L2	95-02-1	d2	20
L2	95-02-1	d2	20	L3	95-03-1	d3	30
L3	95-03-1	d3	30	L1	95-01-1	d1	10
L3	95-03-1	d3	30	L2	95-02-1	d2	20
L3	95-03-1	d3	30	L3	95-03-1	d3	30

خروجی بخش دوم پرس و جوی گزینه‌ی اول پس از انجام عملگر ضرب دکارتی و اجرای شرط where a.damage – amount < b.damage – amount and a.driver <> b.driver به صورت زیر است:

License	accident-date	driver	damage-amount	License	accident-date	driver	damage-amount
L1	95-01-1	d1	10	L2	95-02-1	d2	20
L1	95-01-1	d1	10	L3	95-03-1	d3	30
L2	95-02-1	d2	20	L3	95-03-1	d3	30

خروجی بخش دوم پرس و جوی گزینه‌ی اول پس از انجام عملگر ضرب دکارتی و اجرای شرط where و اجرای دستور select a.driver, a.damage – amount به صورت زیر است:

driver	damage-amount
d1	10
d1	10
d2	20

توجه: شرط where مطرح شده، نام رانندگانی را مشخص می‌کند که هزینه تصادف آنها از هزینه تصادف حداقل یک راننده دیگر کمتر باشد، به عبارت دیگر شرط where نام رانندگان با هزینه تصادف MAX را حذف می‌کند. در واقع نام رانندگانی که هزینه تصادف آنها MAX یعنی بیشینه نیست استخراج می‌شود.

همچنین در ادامه، خروجی بخش اول پرس و جوی گزینه‌ی اول به صورت زیر است:

```
select driver, damage – amount
from Accident
```

driver	damage-amount
d1	10
d2	20
d3	30

توجه: در پراتز سمت راست عملگر تفاضل (Except) نام رانندگانی مشخص می‌شود که هزینه تصادف آنها از هزینه تصادف حداقل یک راننده دیگر کمتر بوده است. همچنین در پراتز سمت چپ عملگر تفاضل (Except) نام کلیه رانندگانی که هزینه تصادفی داشته‌اند، مشخص می‌شود. پس از انجام عملگر تفاضل، نام رانندگانی در خروجی پرس و جو استخراج می‌شوند که هزینه تصادفی داشته‌اند و هزینه‌ی تصادف آنها از هزینه تصادف حداقل یک راننده دیگر کمتر نبوده‌است. به عبارت دیگر در خروجی پرس و جوی گزینه‌ی اول نام رانندگانی استخراج می‌شود که هزینه تصادف آنها MAX و بیشینه بوده است.

در نهایت پس از انجام عملگر Except خروجی پرس و جو به صورت زیر خواهد بود:

driver	damage-amount	Except	driver	damage-amount	=	driver	damage-amount
d1	10		d1	10		d3	30
d2	20		d1	10			
d3	30		d2	20			

عملگر تفاضل (Except)

این عملگر توسط دستور Except نمایش داده می‌شود. عملگر Except یک عملگر اصلی است. عملگر Except جهت تفاضل سطرهای دو جدول مورد استفاده قرار می‌گیرد. اگر R_1 و R_2 دو رابطه باشند، منظور از R_1 Except R_2 مجموعه کلیه تاپل‌هایی است که عضو R_1 هستند اما در R_2 حضور ندارند. در جبر رابطه‌ای و SQL تفاضل هر دو رابطه دلخواه امکان‌پذیر نیست. مگر اینکه شروط سازگاری در مورد آنها برقرار باشد. در جبر رابطه‌ای و SQL دو شرط به عنوان شروط سازگاری مطرح است:

شرط اول: تعداد ستون‌های دو جدول یکسان باشد، به عبارت دیگر دو رابطه (جدول) هم درجه باشند.

شرط دوم: نوع یا دامنه ستون‌های متناظر در دو جدول یکسان باشد.

اگر بنخواهیم دو شرط فوق را در یک جمله بیان کنیم، اینطور خواهد بود، شروط سازگاری یعنی تیتراها در دو رابطه (جدول) یکسان باشد.

فرم کلی عملگر Except به صورت زیر است:

$$R_3 = R_1 \text{ Except } R_2$$

توجه: در پرس و جوی گزینه‌ی اول شروط سازگاری در طرفین عملگر تفاضل (Except) برقرار است.

همچنین جدول Accident با مقادیر زیر را در نظر بگیرید:

License	accident-date	driver	damage-amount
L1	95-01-1	d1	10
L1	95-02-1	d1	20
L1	95-03-1	d1	30

جدول Accident

توجه: در جدول فوق، فقط تصادف‌های یک راننده بررسی شده است.

مطابق پرس و جوی مطرح شده در گزینه‌ی اول، داریم:

```
(select driver, damage – amount
from Accident) Except (select a.driver, a.damage – amount
from Accident a, Accident b
where a.damage – amount < b.damage – amount and a.driver <> b.driver
```

با توجه به جداول فوق، خروجی بخش دوم پرس و جوی گزینه‌ی اول پس از انجام عملگر ضرب دکارتی به صورت زیر است:

License	accident-date	driver	damage-amount	License	accident-date	driver	damage-amount
L1	95-01-1	d1	10	L1	95-01-1	d1	10
L1	95-01-1	d1	10	L1	95-02-1	d1	20
L1	95-01-1	d1	10	L1	95-03-1	d1	30
L1	95-02-1	d1	20	L1	95-01-1	d1	10
L1	95-02-1	d1	20	L1	95-02-1	d1	20
L1	95-02-1	d1	20	L1	95-03-1	d1	30
L1	95-03-1	d1	30	L1	95-01-1	d1	10
L1	95-03-1	d1	30	L1	95-02-1	d1	20
L1	95-03-1	d1	30	L1	95-03-1	d1	30

خروجی بخش دوم پرس و جوی گزینه‌ی اول پس از انجام عملگر ضرب دکارتی و اجرای شرط where a.damage – amount < b.damage – amount and a.driver <> b.driver به صورت زیر است:

License	accident-date	driver	damage-amount	License	accident-date	driver	damage-amount

توجه: علت تهی شدن خروجی فوق این است که در شرط where شاید a.damage – amount < b.damage – amount برقرار باشد، اما a.driver <> b.driver برقرار نیست، چون فقط یک راننده وجود دارد که مقدار driver آنها مخالف هم نیستند.

خروجی بخش دوم پرس و جوی گزینه‌ی اول پس از انجام عملگر ضرب دکارتی و اجرای شرط where و اجرای دستور select a.driver, a.damage – amount به صورت زیر است:

driver	damage-amount

همچنین در ادامه، خروجی بخش اول پرس و جوی گزینه‌ی اول به صورت زیر است:

```
select driver, damage – amount
from Accident
```

driver	damage-amount
d1	10
d1	20
d1	30

در نهایت پس از انجام عملگر Except خروجی پرس و جو به صورت زیر خواهد بود:

driver	damage-amount	Except	driver	damage-amount	=	driver	damage-amount
d1	10					d1	10
d1	20					d1	20
d1	30					d1	30

توجه: نتیجه اینکه گزینه اول در بررسی هزینه‌های تصادف چند راننده مختلف، حالت هزینه بیشینه تصادف را میان رانندگان مختلف استخراج می‌کند، اما در بررسی هزینه‌های تصادف یک راننده، هزینه بیشینه تصادف را مشخص نمی‌کند. بنابراین گزینه اول پاسخ سوال نیست.

مطابق پرس و جوی مطرح شده در گزینه‌ی دوم، داریم:

```
select driver, damage – amount
from Accident
where damage – amount in (select MAX(damage – amount)
from Accident)
```

با توجه به وجود دستور in، زیر پرس و جوی داخلی فوق یک Normal Subquery است، یعنی ابتدا زیر پرس و جوی داخلی یک بار و برای همیشه اجرا می‌گردد، سپس پرس و جوی خارجی به ازای حرکت در هر یک از سطرها، از مقادیر زیر پرس و جوی داخلی استفاده می‌کند. در پرس و جوی فوق به ازای حرکت در هر سطر از جدول Accident مقدار جلوی in بررسی می‌گردد که آیا damage-amount برابر MAX(damage-amount) است یا خیر. اگر برابر بود سطر مورد نظر از جدول Accident در خروجی نمایش داده می‌شود.

توسط دستور where در فرم زیر:

```
SELECT driver , damage-amount
FROM Accident
WHERE damage-amount in (...)
```

برای هر سطر از جدول Accident مقدار جلوی in که حاصل یک تابع آماری به صورت MAX(damage-amount) است، بررسی می‌گردد، اگر damage-amount موجود در هر سطر، برابر MAX(damage-amount) جلوی in بود، آنگاه شرط جلوی where که همان in است، TRUE می‌گردد و سطر مورد نظر از جدول Accident انتخاب می‌گردد و این رویه برای تک تک سطرها، جدول Accident، تا به انتهای جدول Accident ادامه پیدا می‌کند. به بیان دیگر این پرس و جو نام رانندگانی از جدول Accident را می‌دهد که در شرط پراتنز مقابل in قرار دارند.

به عبارت دیگر در خروجی پرس و جوی گزینه‌ی دوم نام رانندگانی استخراج می‌شود که هزینه تصادف آنها MAX و بیشینه بوده است. که مطابق پرس و جوی مطرح شده در صورت سوال است.
توجه: همان‌طور که مشاهده می‌کنید، هزینه تصادف راننده d3، از هزینه تصادف تمام رانندگان دیگر بیشتر است.

<u>License</u>	<u>accident-date</u>	driver	damage-amount
L1	95-01-1	d1	10
L2	95-02-1	d2	20
L3	95-03-1	d3	30

با توجه به جداول فوق، ابتدا خروجی زیر پرس و جوی داخلی بر اساس جدول Accident به صورت زیر محاسبه می‌گردد:

<u>License</u>	<u>accident-date</u>	driver	damage-amount
L1	95-01-1	d1	10
L2	95-02-1	d2	20
L3	95-03-1	d3	30

Select MAX(damage-amount)
From Accident

که مقدار آن برابر 30 می‌شود، به صورت زیر:

$30 > 20 > 10$

بنابراین در ادامه پرس و جوی زیر را خواهیم داشت:

SELECT driver , damage-amount
FROM Accident
WHERE damage-amount in (30)

همانطور که گفتیم برای هر سطر از جدول Accident مقدار جلوی in که حاصل یک تابع آماری به صورت MAX(damage-amount) است، بررسی می‌گردد، اگر damage-amount موجود در هر سطر، برابر MAX(damage-amount) جلوی in بود، آنگاه شرط جلوی where که همان in است، TRUE می‌گردد و سطر مورد نظر از جدول Accident انتخاب می‌گردد و این رویه برای تک تک سطرهای جدول Accident، تا به انتهای جدول Accident ادامه پیدا می‌کند. به بیان دیگر این پرس و جو نام رانندگانی از جدول Accident را می‌دهد که در شرط پراتنز مقابل in قرار دارند.
با توجه به جداول فوق، خروجی نهایی پرس و جوی فوق پس از انجام عملگر in به ازای هر سطر جدول Accident به صورت زیر است:

driver	damage-amount
d3	30

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «نام رانندگانی استخراج می‌شود که هزینه تصادف آنها MAX و بیشینه بوده است.»

همچنین جدول Accident با مقادیر زیر را در نظر بگیرید:

<u>License</u>	<u>accident-date</u>	driver	damage-amount
L1	95-01-1	d1	10
L1	95-02-1	d1	20
L1	95-03-1	d1	30

جدول Accident

با توجه به جداول فوق، خروجی نهایی پرس و جوی فوق پس از انجام عملگر in به ازای هر سطر جدول Accident به صورت زیر است:

driver	damage-amount
d3	30

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «نام رانندگانی استخراج می‌شود که هزینه تصادف آنها MAX و بیشینه بوده است.»

توجه: نتیجه اینکه گزینه دوم در بررسی هزینه‌های تصادف چند راننده مختلف، حالت هزینه بیشینه تصادف را میان رانندگان مختلف استخراج می‌کند و همچنین در بررسی هزینه‌های تصادف یک راننده هم، هزینه بیشینه تصادف را مشخص می‌کند. بنابراین گزینه دوم پاسخ سوال است.

توجه: در پرس و جوی گزینه‌ی دوم چون خروجی select داخلی فقط یک مقدار و حاصل تابع آماری MAX است، می‌توان از عملگر = به جای عملگر in نیز استفاده نمود، بنابراین فرم زیر همانند فرم گزینه دوم است.

```
select driver, damage - amount
from Accident
where damage - amount = (select MAX(damage - amount)
from Accident)
```

مطابق پرس و جوی مطرح شده در گزینه‌ی سوم، داریم:

```
select driver, damage - amount
from Accident
where damage - amount = MAX(damage - amount)
```

در پرس و جوی گزینه‌ی سوم تابع عددی MAX در محل نادرست مورد استفاده قرار گرفته است، به طور کلی توابع عددی در جلوی Where باید داخل Select قرار بگیرند. بنابراین پرس و جوی گزینه‌ی سوم دارای خطای نحوی است و از سوی کامپایلر اجرا نمی‌گردد. فرم اصلاح شده‌ی گزینه‌ی سوم می‌تواند به فرم گزینه‌ی دوم باشد. بنابراین گزینه سوم پاسخ سوال نیست.

توجه: سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه دوم را به عنوان پاسخ اعلام کرده بود.

۷۰- گزینه (۴) صحیح است.

چهار جدول Person، Car، Accident و Owns با مقادیر زیر را در نظر بگیرید:

<u>SSN</u>	name	address	<u>License</u>	year	model

جدول Person

جدول Car

License	accident-date	driver	damage-amount
L1	95-01-1	d1	10
L1	95-01-2	d1	15
L2	95-02-1	d2	20

جدول Accident

SSN	License

جدول Owns

توجه: دقت کنید که مطابق فرض سوال، ترکیب دو ستون License و accident-date هر دو باهم به عنوان کلید کاندید جدول Accident مشخص شده است، بنابراین نباید ترکیب این دو ستون باهم سطرهای تکراری داشته باشد، پس یک خودرو (License) در یک تاریخ تصادف (accident-date) نمی تواند بیش از یکبار تکرار شود، یعنی نباید دوبار و در دو سطر و بیشتر تکرار شود.

مطابق پرس و جوی مطرح شده در گزینه ی اول، داریم:

```
select distinct A.License
from Accident A
where A.License in (select B.License
                    from Accident B
                    where A.accident - date <> B.accident - date)
```

با توجه به وجود عملگر in و البته شرط اتصال زیر پرس و جوی داخلی به پرس و جوی خارجی، زیر پرس و جوی داخلی فوق یک Correlated Subquery است، یعنی به ازای حرکت در هریک از سطرهای پرس و جوی خارجی، یک بار به طور کامل از ابتدا تا انتها زیر پرس و جوی داخلی اجرا و بر اساس شرطی که زیر پرس و جوی داخلی را به پرس و جوی خارجی متصل می کند، بررسی انجام می شود. مانند دو حلقه تو در تو، که به ازای هربار اجرای حلقه خارجی، یک بار به طور کامل حلقه داخلی اجرا می گردد. توسط دستور where در فرم زیر:

```
SELECT distinct A.License
FROM Accident A
WHERE A.License IN (...)
```

برای هر سطر از جدول Accident شرط جلوی IN که حاصل یک مقایسه می باشد، محاسبه می گردد، اگر غیرتهی بود، شرط جلوی where که همان IN است، TRUE می گردد و سطر مورد نظر از جدول Accident انتخاب می گردد و این رویه برای تک تک سطرهای جدول Accident، تا به انتهای جدول Accident ادامه پیدا می کند. به بیان دیگر این پرس و جو شماره پلاک ماشین هایی را می دهد که پراتنز مقابل IN برای آنها غیرتهی است. این پراتنز هنگامی غیرتهی می شود که حاصل مقایسه بیان شده در این پراتنز غیرتهی شود. حاصل این مقایسه در صورتی غیرتهی می شود که شماره پلاک ماشین مورد بررسی در بیش از یک تصادف، درگیر باشد. به عبارت دیگر پرس و جوی گزینه ی اول شماره پلاک ماشین هایی را استخراج می کند که در بیش از یک تصادف، درگیر هستند. که مطابق پرس و جوی مطرح شده در صورت سوال است.

حال برای سطر سوم از جدول Accident با توجه به شرط اتصال $A.\text{accident-date} = B.\text{accident-date}$
 $L2$ $L2$
 داریم:

<u>License</u>	<u>Accident.date</u>	...	<u>License</u>	<u>Accident.date</u>	...
L1	95-01-1		L1	95-01-1	
L1	95-01-2		L1	95-01-2	
L2	95-02-1		L2	95-02-1	

Accident A *Accident B*

که پرس و جوی زیر برای آن اجرا می‌شود:

```
select distinct A.License
from Accident A
where A.License in (select B.License
from Accident B
where A.accident - date <> B.accident - date)
```

توجه: خروجی زیر پرس و جوی داخلی در تاریخ تصادف‌های مختلف به ازای سطر سوم جدول Accident A، به صورت زیر:

B.License
 نهی
 Accident B

جلوی in برابر تهی است، بنابراین شرط where در پشت in FALSE می‌گردد. بنابراین سطر سوم از جدول Accident A در خروجی نمایش داده نمی‌شود.

بنابراین شماره پلاک ماشین **L2** از آنجاکه در بیش از یک تصادف در تاریخ‌های مختلف، درگیر نبوده است، جهت نمایش در خروجی انتخاب نمی‌شود، به صورت زیر:

B.License
 نهی
 Accident A

و در نهایت پس از اجرای کامل دستور `Select distinct A.License` خروجی پرس و جو به صورت زیر خواهد بود:

A.License
L1
L1
 Accident A

که پس از اجرای دستور `distinct` سطرهای تکراری حذف می‌شود، به صورت زیر:

A.License
L1
 Accident A

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «شماره پلاک ماشین‌هایی که در بیش از یک تصادف در تاریخ‌های مختلف درگیر بوده‌اند.»

فرم دوم گزینه اول با استفاده از دستور exists به صورت زیر است:

```
select distinct A.License
from Accident A
where exists(select B.License
from Accident B
where A.License=B.License and A.accident - date <> B.accident - date)
```

که پس از اجرای دستور distinct سطرهای تکراری حذف می‌شود، به صورت زیر:

A.License

L1
Accident A

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «شماره پلاک ماشین‌هایی که در بیش از یک تصادف در تاریخ‌های مختلف درگیر بوده‌اند.»

مطابق پرس و جوی مطرح شده در گزینه‌ی دوم، داریم:

```
select License
from Accident
group by License
having count(accident - date)>1
```

با توجه به جداول فوق، پس از انجام دستور group by License بر اساس ستون License خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:

L1	L2
95-01-1 dl 10	95-02-1 d2 20
95-01-2 dl 15	-----
گروه اول	گروه دوم

توجه: هر گروه شامل تصادف‌های ثبت شده در تاریخ‌های مختلف برای یک شماره پلاک است.

و در نهایت دستور $\text{having count}(\text{accident} - \text{date}) > 1$ برای هر گروه به طور مستقل اعمال می‌گردد.

توجه: دستور HAVING بر روی گروه‌ها، اعمال می‌گردد.

L1	L2
95-01-1 dl 10	95-02-1 d2 20
95-01-2 dl 15	-----
گروه اول	گروه دوم
$2 > 1,$	$1 \ngtr 1$

توجه: با توجه به شرط انتخاب گروه توسط دستور $\text{having count}(\text{accident} - \text{date}) > 1$ ، فقط گروه اول جهت نمایش در خروجی انتخاب می‌شود.

و در نهایت دستور select License داخل دستور select برای هر گروه انتخاب شده توسط دستور Having

به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد، بنابراین خروجی نهایی پرس و جو به صورت زیر است:

```
License
-----
L1
```

توجه: دستور GROUP BY، سرگروه‌ها را، راهی خروجی می‌کند، که به تبع سرگروه، سطر تکراری ندارد. که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «شماره پلاک ماشین‌هایی که در بیش از یک تصادف در تاریخ‌های مختلف درگیر بوده‌اند.»
مطابق پرس و جوی مطرح شده در گزینه‌ی سوم، داریم:

```
select A.License
from Accident A, Accident B
where A.License = B.License and A.accident - date <> B.accident - date
```

با توجه به جداول فوق، خروجی پرس و جوی گزینه‌ی سوم پس از انجام عملگر ضرب دکارتی به صورت زیر است:

License	accident-date	driver	accident-amount	License	accident-date	driver	accident-amount
L1	95-01-1	d1	10	L1	95-01-2	d1	15
L1	95-01-2	d1	15	L1	95-01-1	d1	10

و در نهایت پس از اجرای دستور select A.License خروجی پرس و جو به صورت زیر خواهد بود:

```
A.License
-----
L1
L1
Accident A
```

که مطابق پرس و جوی مطرح شده در صورت سوال نیست، زیرا در صورت سوال این فرض قید شده است که پاسخ به زبان SQL باشد و سطرهای تکراری فقط یک‌بار نشان داده شوند، که در پرس و جوی گزینه‌ی سوم به دلیل عدم استفاده از دستور distinct در خروجی دستور select جهت حذف سطرهای تکراری، فرض مطرح شده در صورت سوال نقض شده است. بنابراین گزینه‌ی سوم پاسخ سوال نیست.

توجه: البته اگر پرس و جوی گزینه‌ی سوم به فرم زیر اصلاح شود آنگاه خروجی گزینه‌ی سوم، همانند گزینه اول و دوم می‌شود، به عبارت دیگر فرم اصلاح شده گزینه‌ی سوم، همان فرم کلاسیک گزینه‌های اول و دوم با استفاده از ضرب دکارتی است، فرم اصلاح شده گزینه‌ی سوم به فرم زیر است:

```
select distinct A.License
from Accident A, Accident B
where A.License = B.License and A.accident - date <> B.accident - date
```

و در نهایت پس از اجرای کامل دستور Select distinct A.License خروجی پرس و جو به صورت زیر خواهد بود:

A.License

L1

L1

Accident A

که پس از اجرای دستور distinct سطرهای تکراری حذف می‌شود، به صورت زیر:

A.License

L1

Accident A

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «شماره پلاک ماشین‌هایی که در بیش از یک تصادف در تاریخ‌های مختلف درگیر بوده‌اند».

توجه: سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه چهارم را به عنوان پاسخ اعلام کرده بود.

۷۱- گزینه (۳) صحیح است.

دو جدول Node و Edge با مقادیر زیر را در نظر بگیرید:

<u>NID</u>	Name	Color	Description	<u>NID1</u>	<u>NID2</u>	EdgeType
N1	NN1	C1	D1	N1	N2	T1
N2	NN2	C2	D2	N1	N3	T2
N3	NN3	C3	D3	N4	N1	T3
N4	NN4	C4	D4	N3	N2	T4
				N4	N3	T5

جدول Node

جدول Edge

توجه: کلید کاندید جدول Node ستون NID است و جدول Edge دارای دو کلید خارجی NID1 و NID2 است که هر دو به کلید کاندید جدول Node ارجاع می‌کنند.

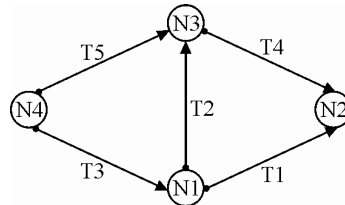
توجه: ستون NID1 در جدول Edge به عنوان کلید خارجی به ستون NID از جدول Node ارجاع می‌کند.

توجه: ستون NID2 در جدول Edge به عنوان کلید خارجی به ستون NID از جدول Node ارجاع می‌کند.

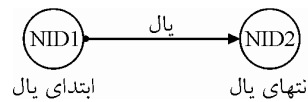
توجه: مقادیر کلید خارجی یعنی ستون‌های NID1 و NID2 از جدول Edge همواره باید زیرمجموعه مقادیر کلید کاندید یعنی ستون NID از جدول Node باشد.

توجه: هر تاپل یا سطر یا رکورد از جدول Edge نشانه یک ارتباط است، پس هر تاپل نشانه یک یال هم هست، که مجموع این یال‌ها یک گراف را ایجاد می‌کند. خطی که از یک گره به گره بعدی رسم می‌شود یک یال است. هر گراف G شامل دو مجموعه V و E است. V مجموعه محدود و غیرتهی از رئوس است و E مجموعه‌ای محدود و احتمالاً تهی از لبه‌ها می‌باشد. $V(G)$ و $E(G)$ مجموعه رئوس و لبه‌های گراف G را نمایش می‌دهند. برای نمایش یک گراف می‌توانیم بنویسیم $G=(V, E)$. در یک گراف بدون جهت، زوج رئوس، زوج مرتب نیستند. بنابراین زوج‌های (V_0, V_1) و (V_1, V_0) باهم یکسان هستند. گراف حداقل یک راس دارد و نمی‌تواند کاملاً تهی باشد. در یک گراف جهت دار، هر لبه با زوج مرتب $\langle V_0, V_1 \rangle$ نمایش داده می‌شود، که پیکانی از V_0 به V_1 ترسیم می‌شود. بنابراین $\langle V_1, V_0 \rangle$ و $\langle V_0, V_1 \rangle$ دو لبه متفاوت را

نمایش می‌دهند. گراف جهت دار را به صورت $G = \langle V, E \rangle$ نمایش می‌دهیم.
توجه: گراف جهت دار جدول Edge به صورت زیر است.



گراف جدول Edge



هر تاپل از جدول Edge نشانه یک پال از گراف جدول Edge است.

توجه: همانطور که واضح است گره‌های موجود در ستون NID1 از جدول Edge یا گراف Edge، پال‌های خروجی دارند.

$$NID1 = \{N1, N3, N4\}$$

یعنی از گره‌های N1، N3 و N4 پال خارج شده است.

فرم اول پرس و جوی آن به صورت زیر است:

```
select NID
from node
where EXISTS (select *
              from edge
              where node.NID=edge.NID1)
```

فرم دوم پرس و جوی آن به صورت زیر است:

```
select NID
from node
where NID IN (select NID1
             from edge)
```

فرم سوم پرس و جوی آن به صورت زیر است:

```
select distinct NID
from node , edge
where node.NID=edge.NID1
```

توجه: در فرم سوم یا کلاسیک با استفاده از ضرب دکارتی، جهت حذف سطرهای تکراری می‌بایست از دستور distinct استفاده نمود.

خروجی پرس و جویها به صورت زیر است:

<u>NID</u>
N1
N3
N4

توجه: همانطور که واضح است گره‌های موجود در ستون NID2 از جدول Edge یا گراف Edge. یال‌های ورودی دارند.

$NID2 = \{N1, N2, N3\}$

یعنی به گره‌های N1، N2 و N3 یال وارد شده است.
فرم اول پرس و جوی آن به صورت زیر است:

```
select NID
from node
where EXISTS (select *
              from edge
              where node.NID=edge.NID2)
```

فرم دوم پرس و جوی آن به صورت زیر است:

```
select NID
from node
where NID IN (select NID2
             from edge)
```

فرم سوم پرس و جوی آن به صورت زیر است:

```
select distinct NID
from node , edge
where node.NID=edge.NID2
```

توجه: در فرم سوم یا کلاسیک با استفاده از ضرب دکارتی، جهت حذف سطرهای تکراری می‌بایست از دستور distinct استفاده نمود.

خروجی پرس و جویها به صورت زیر است:

<u>NID</u>
N1
N2
N3

نتیجه: استخراج گره‌هایی که هم یالی از آن خارج شده است و هم یالی به آن وارد شده است، به صورت زیر است:

$$NID1 \cap NID2 = \{N1, N3, N4\} \cap \{N1, N2, N3\} = \{N1, N3\}$$

فرم اول پرس و جوی آن به صورت زیر است:

```
(select NID
 from node
 where EXISTS (select *
              from edge
              where node.NID=edge.NID1)) INTERSECT (select NID
 from node
 where EXISTS (select *
              from edge
              where node.NID=edge.NID2))
```

فرم دوم پرس و جوی آن به صورت زیر است:

```
select NID
from node
where EXISTS (select *
              from edge
              where node.NID=edge.NID1) AND EXISTS (select *
              from edge
              where node.NID=edge.NID2))
```

خروجی پرس و جویها به صورت زیر است:

NID
N1
N3

استخراج گرههایی که یالی از آن خارج شده است ولی یالی به آن وارد نشده است، به صورت زیر است:

$$NID1 - NID2 = \{N1, N3, N4\} - \{N1, N2, N3\} = \{N4\}$$

فرم اول پرس و جوی آن به صورت زیر است:

```
(select NID
from node
where EXISTS (select *
              from edge
              where node.NID=edge.NID1)) EXCEPT (select NID
from node
where EXISTS (select *
              from edge
              where node.NID=edge.NID2))
```

فرم دوم پرس و جوی آن به صورت زیر است:

```
select NID
from node
where EXISTS (select *
              from edge
              where node.NID=edge.NID1) AND NOT EXISTS (select *
              from edge
              where node.NID=edge.NID2)
```

خروجی پرس و جویها به صورت زیر است:

NID
N1
N3

استخراج گرههایی که یالی به آن وارد شده است ولی یالی از آن خارج نشده است، به صورت زیر است:

$$NID2 - NID1 = \{N1, N2, N3\} - \{N1, N3, N4\} = \{N2\}$$

فرم اول پرس و جوی آن به صورت زیر است:

```
(select NID
from node
where EXISTS (select *
               from edge
               where node.NID=edge.NID2)) EXCEPT (select NID
                                                    from node
                                                    where EXISTS (select *
                                                           from edge
                                                           where node.NID=edge.NID1))
```

فرم دوم پرس و جوی آن به صورت زیر است:

```
select NID
from node
where EXISTS (select *
               from edge
               where node.NID=edge.NID2) AND NOT EXISTS (select *
                                                           from edge
                                                           where node.NID=edge.NID1)
```

فرم سوم پرس و جوی آن به صورت زیر است:

```
select distinct NID
from node , edge
where node.NID=edge.NID2 AND NOT EXISTS (select *
                                           from edge
                                           where node.NID = edge.NID1)
```

توجه: می‌توان دستور Exists اول از فرم دوم را به صورت ضرب دکارتی و کلاسیک نیز نوشت.

توجه: که دقیقا همان فرم پرس و جوی اول مطرح شده در صورت سوال است.

خروجی پرس و جویها به صورت زیر است:

$$\frac{NID}{N2}$$

مطابق پرس و جوی اول مطرح شده در صورت سوال، داریم:

```
select distinct NID
from node , edge
where node.NID=edge.NID2 AND NOT EXISTS (select *
                                           from edge
                                           where node.NID = edge.NID1)
```

توجه: از آنجاییکه ستون‌های NID و NID1 در دو جدول مختلف تکراری نیستند، بنابراین الزامی بر وجود نام جداول پشت این ستون‌ها وجود ندارد.

با توجه به وجود عملگر not exists، زیر پرس و جوی داخلی فوق یک Correlated Subquery است، یعنی به ازای حرکت در هریک از سطرهای پرس و جوی خارجی، یک بار به طور کامل از ابتدا تا انتها زیر پرس و جوی داخلی اجرا و بر اساس شرطی که زیر پرس و جوی داخلی را به پرس و جوی خارجی متصل می‌کند، بررسی انجام می‌شود. مانند دو حلقه تو در تو، که به ازای هربار اجرای حلقه خارجی، یک بار به طور کامل حلقه داخلی اجرا می‌گردد.

```
SELECT distinct NID
FROM node , edge
WHERE node.NID=edge.NID2 AND NOT EXISTS (...)
```

توسط دستور where در فرم زیر:

برای هر سطر از جدول حاصل از ضرب دکارتی $node, edge$ به طور همزمان شرط $node.NID=edge.NID2$ (شرط اول) و شرط جلوی $NOT EXISTS$ (شرط دوم) بررسی می‌شود. اگر به طور همزمان شرط $node.NID=edge.NID2$ و همچنین شرط جلوی $NOT EXISTS$ برقرار بود، آنگاه سطر مورد نظر از جدول حاصل از ضرب دکارتی $node, edge$ انتخاب می‌گردد و این رویه برای تک تک سطرهای جدول حاصل از ضرب دکارتی $node, edge$ ، تا به انتهای جدول حاصل از ضرب دکارتی $node, edge$ ادامه پیدا می‌کند. دقت کنید که برای هر سطر از جدول حاصل از ضرب دکارتی $node, edge$ شرط جلوی $NOT EXISTS$ که حاصل یک مقایسه می‌باشد، محاسبه می‌گردد، اگر تهی بود، شرط جلوی $NOT EXISTS$ (شرط دوم) که همان $NOT EXISTS$ است، $TRUE$ می‌گردد.

به بیان دیگر این پرس و جو شماره گره‌هایی را می‌دهد که NID آن، عضو ستون $NID2$ یعنی مجموعه گره‌های دارای **یال وارد شونده** باشد و همچنین به طور همزمان پرائنتز مقابل $NOT EXISTS$ برای آن‌ها تهی باشد. این پرائنتز هنگامی تهی می‌شود که حاصل مقایسه بیان شده در این پرائنتز تهی شود. حاصل این مقایسه در صورتی تهی می‌شود که شماره گره مورد بررسی، NID آن، عضو ستون $NID1$ یعنی مجموعه گره‌های دارای **یال خارج شونده** نباشد. به عبارت دیگر پرس و جو اول مطرح شده در صورت سوال شماره گره‌هایی را استخراج می‌کند که **یال ورودی** دارند اما **یال خروجی** ندارند. یعنی استخراج گره‌هایی که یالی به آن وارد شده است ولی یالی از آن **خارج** نشده است، که مطابق پرس و جو اول مطرح شده در صورت سوال است که در هیچکدام از گزینه‌های صورت سوال، این حالت پرس و جو وجود ندارد. به بیان دیگر عبارت موجود در جلوی دستور $not exists$ که به صورت زیر است:

```
select distinct NID
from node , edge
where node.NID=edge.NID2 AND NOT EXISTS (select *
                                         from edge
                                         where node.NID = edge.NID1)
```

توسط دستور $node.NID = edge.NID1$ به محیط خارج یعنی جدول حاصل از ضرب دکارتی $node, edge$ متصل می‌گردد. حال به ازای حرکت در هر سطر از جدول حاصل از ضرب دکارتی $node, edge$ ، یک بار به طور کامل سطرهای جدول $edge$ از ابتدا تا انتها با توجه به شرط اتصال بررسی می‌گردد. مطابق شکل زیر:

ابتدا برای سطر اول از جدول حاصل از ضرب دکارتی $node, edge$ البته با اعمال شرط $node.NID=edge.NID2$ با توجه به شرط اتصال $node.NID = edge.NID1$ داریم:

N2					×			N1		
NID	...	NID1	NID2	EdgeType	NID1	NID2	EdgeType	NID1	NID2	EdgeType
N2		N1	N2	T1	N1	N2	T1	N1	N3	T2
N3		N1	N3	T2	N1	N3	T2	N4	N1	T3
N1		N4	N1	T3	N4	N1	T3	N3	N2	T4
N3		N4	N3	T5	N3	N2	T4	N4	N3	T5
N2		N3	N2	T4						

Node , Edge

Edge

که پرس و جو زیر برای آن اجرا می‌شود:

```
select distinct NID
from node , edge
where node.NID=edge.NID2 AND NOT EXISTS (select *
                                         from edge
                                         where node.NID = edge.NID1)
```

توجه: خروجی زیر پرس و جوی داخلی در کنترل عضویت شماره گره مورد نظر یعنی N2 از NID در ستون NID1 یعنی مجموعه گره‌های دارای یال **خارج** شونده. به ازای سطر اول جدول حاصل از ضرب دکارتی node , edge البته با اعمال شرط node.NID=edge.NID2 به صورت زیر است:

NID1	NID2	EdgeType
------	------	----------

تهی

Edge

ججوی not exists برابر تهی است، بنابراین شرط where در not exists (شرط دوم) ، TRUE می‌گردد. شرط where در node.NID=edge.NID2 (شرط اول) هم که از قبل برقرار و TRUE بود. که نتیجه کلی شرط where به دلیل وجود عملگر AND می‌شود TRUE. بنابراین سطر اول از جدول حاصل از ضرب دکارتی node , edge با اعمال شرط node.NID=edge.NID2 توجه به شرط اتصال در خروجی نمایش داده می‌شود. بنابراین گره شماره **N2** از آنجاکه که یال ورودی دارد اما یال خروجی ندارد. یعنی یالی به آن وارد شده است ولی یالی از آن **خارج** نشده است، جهت نمایش در خروجی انتخاب می‌شود، به صورت زیر:

NID
N2

حال برای سطر دوم از جدول حاصل از ضرب دکارتی node , edge البته با اعمال شرط node.NID=edge.NID2 با توجه به شرط اتصال $node.NID = edge.NID1$ داریم:

NID	...	NID1	NID2	EdgeType	NID1	NID2	EdgeType
N2		N1	N2	T1	N1	N2	T1
N3		N1	N3	T2	N1	N3	T2
N1		N4	N1	T3	N4	N1	T3
N3		N4	N3	T5	N3	N2	T4
N2		N3	N2	T4	N4	N3	T5

Node , Edge**Edge**

که پرس و جوی زیر برای آن اجرا می‌شود:

```
select distinct NID
from node , edge
where node.NID=edge.NID2 AND NOT EXISTS (select *
                                         from edge
                                         where node.NID = edge.NID1)
```

توجه: خروجی زیر پرس و جوی داخلی در کنترل عضویت شماره گره مورد نظر یعنی N3 از NID در ستون NID1 یعنی مجموعه گره‌های دارای یال **خارج** شونده. به ازای سطر دوم جدول حاصل از ضرب

دکارتی $node, edge$ البته با اعمال شرط $node.NID=edge.NID2$ به صورت زیر است:

NID1	NID2	EdgeType
N3	N2	T4

Edge

جلوی $not\ exists$ برابر غیرتهی است، بنابراین شرط $where$ در $not\ exists$ (شرط دوم)، $FALSE$ می‌گردد. شرط $where$ در $node.NID=edge.NID2$ (شرط اول) هم که از قبل برقرار و $TRUE$ بود. که نتیجه کلی شرط $where$ به دلیل وجود عملگر AND می‌شود $FALSE$. بنابراین سطر دوم از جدول حاصل از ضرب دکارتی $node, edge$ با اعمال شرط $node.NID=edge.NID2$ با توجه به شرط اتصال در خروجی نمایش داده نمی‌شود. بنابراین گره شماره **N3** از آنجاکه که یال ورودی دارد و یال خروجی هم دارد. یعنی یالی به آن وارد شده است و یالی هم از آن خارج شده است، جهت نمایش در خروجی انتخاب نمی‌شود، به صورت زیر:

NID
N2

حال برای سطر سوم ($N1$)، چهارم ($N3$) و پنجم ($N2$) از جدول حاصل از ضرب دکارتی $node, edge$ با اعمال شرط $node.NID=edge.NID2$ نیز همین روال تکرار می‌شود، همچنین به واسطه وجود دستور $distinct$ در دستور $select$ سطرهای تکراری نیز در خروجی پرس و جو حذف می‌شود. با توجه به جداول فوق، خروجی نهایی پرس و جوی فوق به صورت زیر است:

NID
N2

که مطابق پرس و جوی اول مطرح شده در صورت سوال است. یعنی «شماره گره‌هایی را استخراج می‌کند که یال ورودی دارند اما یال خروجی ندارند. یعنی استخراج گره‌هایی که یالی به آن وارد شده است ولی یالی از آن خارج نشده است، که در هیچکدام از گزینه‌های صورت سوال، این حالت پرس و جو وجود ندارد.

مطابق پرس و جوی دوم مطرح شده در صورت سوال، داریم:

```
select T1.NID
from (select count (NID1) as cnt , NID
      from node left outer join edge on edge.NID1=NID
      group by NID) T1 ,
      (select count (NID2) as cnt, NID
      from node left outer join edge on edge.NID2=NID
      group by NID) T2
where T1.NID=T2.NID and T1.cnt < T2.cnt
```

عملگر الحاق خارجی چپ در جبر رابطه‌ای

این عملگر، مانند الحاق طبیعی، ستون‌های مشترک را فقط یکبار در خروجی قرار می‌دهد. همچنین کلیه سطرهای پیوندپذیر را در خروجی قرار می‌دهد. اما علاوه بر آن کلیه سطرهای پیوندناپذیر جدول سمت چپ را نیز در خروجی قرار می‌دهد و در این حالت برای ستون‌های غیر مشترک جدول سمت راست

مقدار NULL قرار می‌دهد.

عملگر الحاق خارجی چپ در SQL (Left Outer Join)

این عملگر، ستون‌های مشترک را دوبار در خروجی قرار می‌دهد. همچنین کلیه سطرهای پیوندپذیر را در خروجی قرار می‌دهد. اما علاوه بر آن کلیه سطرهای پیوندناپذیر جدول سمت چپ را نیز در خروجی قرار می‌دهد و در این حالت برای تمام ستون‌های جدول سمت راست مقدار NULL قرار می‌دهد.

توجه: به تفاوت عملگر الحاق خارجی چپ در جبر رابطه‌ای و SQL دقت کنید.

توجه: برای شمارش درجه خروجی گره‌ها یعنی شمارش یال‌های خارج شونده از هر گره باید جدول Node در Edge الحاق خارجی چپ شود که نتیجه آن T1 نام‌گذاری شده است، تا تعداد تکرار یال‌های خارج شونده از هر گره شمارش شود، به صورت زیر:

```
(select count (NID1) as cnt , NID
from node left outer join edge on edge.NID1=NID
group by NID) T1
```

NID	name	color	Description	NID1	NID2	EdgeType
N1	NN1	C1	D1	N1	N2	T1
N1	NN1	C1	D1	N1	N3	T2
N2	NN2	C2	D2	NULL	NUUL	NULL
N3	NN3	C3	D3	N3	N2	T4
N4	NN4	C4	D4	N4	N1	T3
N4	NN4	C4	D4	N4	N3	T5

با توجه به جدول فوق، پس از انجام دستور group by NID براساس ستون NID خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:

N1	
NN1 C1 D1 N1 N2 T1	N2
NN1 C1 D1 N1 N3 T2	NN2 C2 D2 NULL NULL NULL
	گروه دوم
گروه اول	
N3	N4
NN3 C3 D3 N3 N2 T4	NN4 C4 D4 N4 N1 T3
	NN4 C4 D4 N4 N3 T5
گروه سوم	گروه چهارم

توجه: هر گروه شامل تعداد یال‌های خارج شونده از هر گره است، چون شرط اتصال روی edge.NID1=NID در node left outer join edge است.

و در نهایت دستور select count (NID1) as cnt , NID داخل دستور select برای هر گروه به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد، بنابراین خروجی نهایی پرس و جو به صورت زیر است:

T1.cnt	T1.NID
2	N1
0	N2
1	N3
2	N4

توجه: دستور GROUP BY، سرگروه‌ها را، راهی خروجی می‌کند، که به تبع سرگروه، سطر تکراری ندارد.
توجه: تابع آماری count (NID1) برای هر گروه به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد.

توجه: تابع آماری count (NID1)، ستون NULL را نادیده می‌گیرد و نمی‌شمارد.

توجه: برای شمارش درجه ورودی گره‌ها یعنی شمارش **یال‌های وارد شونده** به هر گره باید جدول Node در Edge الحاق خارجی چپ شود که نتیجه آن T2 نام‌گذاری شده است، تا تعداد تکرار یال‌های وارد شونده به هر گره شمارش شود، به صورت زیر:

```
(select count (NID2) as cnt , NID
from node left outer join edge on edge.NID2=NID
group by NID) T2
```

NID	name	color	Description	NID1	NID2	EdgeType
N1	NN1	C1	D1	N4	N1	T3
N2	NN2	C2	D2	N1	N2	T1
N2	NN2	C2	D2	N3	N2	T4
N3	NN3	C3	D3	N1	N3	T2
N3	NN3	C3	D3	N4	N3	T5
N4	NN4	C4	D4	NULL	NULL	NULL

با توجه به جدول فوق، پس از انجام دستور group by NID براساس ستون NID خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:

N1 <u>NN1 C1 D1 N4 N1 T3</u> گروه اول	N2 NN2 C2 D2 N1 N2 T1 <u>NN2 C2 D2 N3 N2 T4</u> گروه دوم
N3 NN3 C3 D3 N1 N3 T2 <u>NN3 C3 D3 N4 N3 T5</u> گروه سوم	N4 <u>NN4 C4 D4 NULL NULL NULL</u> گروه چهارم

توجه: هر گروه شامل تعداد **یال‌های وارد شونده** به هر گره است، چون شرط اتصال روی edge.NID2=NID در node left outer join است.

و در نهایت دستور count (NID2) as cnt , NID داخل دستور select برای هر گروه به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد، بنابراین خروجی نهایی پرس و جو به صورت زیر است:

T2.cnt	T2.NID
1	N1
2	N2
2	N3
0	N4

توجه: دستور GROUP BY، سرگروه‌ها را، راهی خروجی می‌کند، که به تبع سرگروه، سطر تکراری ندارد.
توجه: تابع آماری count (NID2) برای هر گروه به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد.

توجه: تابع آماری count (NID2)، ستون NULL را نادیده می‌گیرد و نمی‌شمارد.

توجه: در ادامه T1 و T2 در هم ضرب دکارتی می‌شوند که می‌شود 16 سطر که با اعمال شرط T1.NID=T2.NID در دستور where خروجی زیر را خواهیم داشت:

T1.cnt	T1.NID	T2.cnt	T2.NID
2	N1	1	N1
0	N2	2	N2
1	N3	2	N3
2	N4	0	N4

توجه: در ادامه با اعمال شرط T1.cnt < T2.cnt در دستور where خروجی زیر را خواهیم داشت:

T1.cnt	T1.NID	T2.cnt	T2.NID
0	N2	2	N2
1	N3	2	N3

توجه: در نهایت با اعمال دستور select T1.NID در دستور select خروجی زیر را خواهیم داشت:

T1.NID
N2
N3

که مطابق پرس و جو دوم مطرح شده در صورت سوال است. یعنی «شماره گره‌هایی را می‌دهد که درجه خروجی آنها کمتر از درجه ورودی آنها است.» یعنی استخراج گره‌هایی که **یال‌های خارج شونده** از آن کمتر از **یال‌های وارد شونده** به آن است، که در گزینه سوم این حالت پرس و جو وجود دارد. بنابراین پُر واضح است که گزینه‌ی سوم پاسخ سوال است.

توجه: اگر در شرط where به جای T1.cnt < T2.cnt از T1.cnt > T2.cnt استفاده می‌شد، آنگاه شماره گره‌هایی را می‌داد که درجه خروجی آنها بیشتر از درجه ورودی آنها است.» یعنی استخراج گره‌هایی که **یال‌های خارج شونده** از آن بیشتر از **یال‌های وارد شونده** به آن است.

توجه: اگر در شرط where به جای T1.cnt < T2.cnt از T1.cnt = T2.cnt استفاده می‌شد، آنگاه شماره گره‌هایی را می‌داد که درجه خروجی آنها برابر درجه ورودی آنها است.» یعنی استخراج گره‌هایی که **یال‌های خارج شونده** از آن برابر **یال‌های وارد شونده** به آن است.

توجه: سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه سوم را به عنوان پاسخ اعلام کرده بود.

۷۲- گزینه (۲) صحیح است.

اگر فرض کنیم منظور طراح محترم از دو دامنه (Domain) متفاوت، ستون‌های مختلف دو جدول مختلف باشد، آنگاه از آنجا که جداول مربوط به ستون P# یعنی جدول Supply و Part در پرس و جو مطرح شده در صورت سوال وجود ندارد، بنابراین پرس و جو اجرا نمی‌شود و خطای فقدان و نامعتبر بودن نام

ستون (Invalid column name) از سوی کامپایلر صادر می‌شود و پرس و جو اجرا نمی‌شود و گزینه دوم پاسخ سوال است، اما این فرض نادرست است و بدآموزی دارد. اصولاً خطای مقایسه دو دامنه (Domain) مختلف مربوط به کنترل نوع دو ستون مختلف در یک جدول است و نه ستون‌های مختلف دو جدول مختلف. در واقع بخش اول گزینه دوم درست است، اما بخش دوم گزینه دوم نادرست است. یعنی باید به جای عبارت «زیرا دو دامنه (Domain) متفاوت باهم مقایسه شده است» عبارت «زیرا ستون‌ها نامعتبر هستند (Invalid column name)» در گزینه دوم مطرح می‌شد، چون ستون P# داخل جدول Supplier اصلاً تعریف نشده است و نامعتبر است.

مثال: جهت درک بیشتر، جداول و پرس و جو زیر را در نظر بگیرید:

پیاده‌سازی جدول تولیدکنندگان (Supplier)

Create Table Supplier

```
(
  S# char (5),
  Sname char (20),
  City char (15),
  Primary key (S#)
)
```

پیاده‌سازی جدول قطعات (Part)

Create Table Part

```
(
  P# char (5),
  Color char (10),
  Primary key (P#)
)
```

پیاده‌سازی جدول تولید (Supply)

Create Table Supply

```
(
  S# char (5),
  P# char (5),
  QTY numeric (10),
  Primary key (S#, P#),
  Foreign key (S#) References Supplier(S#)
  on delete cascade
  on update cascade,
  Foreign key (P#) References Part(P#)
  on delete cascade
  on update cascade,
  Check (QTY>1 AND QTY<1000)
)
```

مثال: پرس و جوی زیر را در نظر بگیرید:

```
SELECT S#
FROM Supply
WHERE S# = QTY;
```

توجه: در پرس و جوی فوق هر دو ستون S# و QTY در جدول تعریف شده در پرس و جو یعنی Supply آماده است و خطای فقدان و نامعتبر بودن نام ستون (Invalid column name) از سوی کامپایلر صادر نمی‌شود، اما از آنجا که نوع ستون S# از نوع char(5) و نوع ستون QTY از نوع numeric(10) است خطای قیاس دو دامنه (Domain) متفاوت (Error converting data type) یعنی خطای عدم توانایی در تبدیل numeric(10) و char(5) به هم از سوی کامپایلر صادر می‌شود و پرس و جو اجرا نمی‌شود.

مثال: پرس و جوی زیر را در نظر بگیرید:

```
SELECT S#
FROM Supply
WHERE S# = P#;
```

توجه: در پرس و جوی فوق هر دو ستون S# و P# در جدول تعریف شده در پرس و جو یعنی Supply آماده است و خطای فقدان و نامعتبر بودن نام ستون (Invalid column name) از سوی کامپایلر صادر نمی‌شود، همچنین از آنجا که نوع ستون S# از نوع char(5) و نوع ستون P# از نوع char(5) است خطای قیاس دو دامنه (Domain) متفاوت (Error converting data type) یعنی خطای عدم توانایی در تبدیل char(5) و char(5) به هم نیز از سوی کامپایلر صادر نمی‌شود و پرس و جو به درستی اجرا می‌شود. و شماره تولیدکنندگان و قطعات را که باهم یکسان هستند در خروجی ظاهر می‌کند که در این حالت گزینه اول پاسخ سوال است.

مثال: پرس و جوی زیر را در نظر بگیرید:

```
SELECT S#
FROM Supplier
WHERE S# = Sname;
```

توجه: در پرس و جوی فوق هر دو ستون S# و Sname در جدول تعریف شده در پرس و جو یعنی Supplier آماده است و خطای فقدان و نامعتبر بودن نام ستون (Invalid column name) از سوی کامپایلر صادر نمی‌شود، همچنین از آنجا که نوع ستون S# از نوع char(5) و نوع ستون Sname از نوع char(20) است خطای قیاس دو دامنه (Domain) متفاوت (Error converting data type) یعنی خطای عدم توانایی در تبدیل char(20) و char(5) به هم نیز از سوی کامپایلر صادر نمی‌شود و پرس و جو به درستی اجرا می‌شود. **توجه:** جهت حفظ حقوق دانشجویانی که یکسال تلاش کرده‌اند، این سوال بهتر بود حذف می‌شد و یا همان گزینه دوم اما با تاثیر مثبت می‌شد.

توجه: سازمان سنجش آموزش کشور، در کلید اولیه خود، گزینه دوم را به عنوان پاسخ اعلام نمود، و همچنین در کلید نهایی بازهم گزینه دوم را به عنوان پاسخ نهایی اعلام نمود، بدون تاثیر مثبت، که کار خیلی درستی نبوده است.

۷۳- گزینه (۲) صحیح است.

این سوال تکراری است و پاسخ آن دقیقاً همان پاسخ سوال ۷۱ فصل ششم است.
توجه: سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه دوم را به عنوان پاسخ اعلام کرده بود.

۷۴- گزینه (۳) صحیح است.

جداول زیر را در نظر بگیرید:

MID	Name	MDate	MID	ISBN	BDate	Duration	ISBN	...	Publisher
m1	mn1	21	m1	is1	990101	11	is1		Springer
m2	mn2	22	m1	is2	990202	12	is2		Springer
m3	mn3	23	m2	is1	990303	13	is3		Penguin
Member			m2	is2	990404	14	is4		Penguin
			m2	is3	990505	15		Book	
			m3	is1	990606	16			

Borrow

مطابق پرس و جوی مطرح شده در گزاره الف داریم:

```
SELECT Name
FROM (SELECT Member.MID , Member.Name , count (Book.ISBN) as BorrowCount
      FROM Member , Borrow , Book
      WHERE Member.MID - Borrow.MID and Book.ISBN - Borrow.ISBN
      and Publisher - 'Springer'
      GROUP BY Member.MID , Member.Name) as M
WHERE BorrowCount - (SELECT count(*)
                     FROM Book
                     WHERE Publisher - 'Springer')
```

که البته پرس و جوی مطرح شده در گزاره الف کمی خطای نحوی دارد، که فرم اصلاح شده آن به صورت زیر است:

```
SELECT Name
FROM (SELECT Member.MID , Member.Name , count (Book.ISBN) as BorrowCount
      FROM Member , Borrow , Book
      WHERE Member.MID = Borrow.MID and Book.ISBN = Borrow.ISBN
      and Publisher = 'Springer'
      GROUP BY Member.MID , Member.Name) as M
WHERE BorrowCount = (SELECT count(*)
                     FROM Book
                     WHERE Publisher = 'Springer')
```

توجه: در فرم اصلاح شده‌ی گزاره الف، عملگر = جایگزین عملگر - شده است.
توجه: گزاره الف به فرم اصلاح شده نشده، خطای نحوی دارد و از سوی کامپایلر اجرا نمی‌شود، دقت کنید که جلوی دستور WHERE باید عملگر منطقی قرار گیرد که حاصل آن هم منطقی باشد. که این موضوع نقض شده است. بنابراین گزاره الف مطابق پرس و جوی مطرح شده در صورت سوال نیست. با توجه به جداول فوق، خروجی پرس و جوی داخلی گزاره‌ی الف پس از انجام عملگر ضرب دکارتی و اعمال شرط اتصال به صورت زیر است:

MID	Name	MDate	MID	ISBN	BDate	Duration	ISBN	...	Publisher
m1	mn1	21	m1	is1	990101	11	is1	...	Springer
m1	mn1	21	m1	is2	990202	12	is2	...	Springer
m2	mn2	22	m2	is1	990303	13	is1	...	Springer
m2	mn2	22	m2	is2	990404	14	is2	...	Springer
m3	mn3	23	m3	is1	990606	16	is1	...	Springer

توجه: شرط اتصال زیر:

WHERE Member.MID = Borrow.MID and Book.ISBN = Borrow.ISBN
 and Publisher = 'Springer'

شماره کتاب‌های انتشارات Springer را جلوی امانت گیرنده آن قرار می‌دهد.

همچنین در ادامه، پس از انجام دستور GROUP BY Member.MID , Member.Name براساس ستون‌های Member.MID , Member.Name , count (Book.ISBN) as BorrowCount خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:

m1, mn1	m2, mn2
21 m1 is1 ... 11 is1 ... Springer	22 m2 is1 ... 13 is1 ... Springer
21 m1 is2 ... 12 is2 ... Springer	22 m2 is2 ... 14 is2 ... Springer
گروه اول	گروه دوم

m3, mn3
23 m3 is1 ... 16 is1 ... Springer
گروه سوم

توجه: دستور GROUP BY، سرگروه‌ها را، راهی خروجی می‌کند. و در نهایت دستور SELECT برای هر گروه به طور مستقل اعمال می‌گردد.

MID	Name	BorrowCount
m1	mn1	2
m2	mn2	2
m3	mn3	1

توجه: دستور count (Book.ISBN) as BorrowCount بر روی گروه‌ها، اعمال می‌گردد.

توجه: خروجی قطعه پرس و جوی مطرح شده در دستور WHERE خارجی به صورت زیر است:

```
2 = (SELECT count(*)
FROM Book
WHERE Publisher = 'Springer')
```

توجه: با توجه به شرط انتخاب سطر توسط دستور (SELECT count(*) WHERE BorrowCount = ... ، فقط سطر اول و دوم جهت نمایش در خروجی انتخاب می‌شود.

```
WHERE BorrowCount = (SELECT count(*)
FROM Book
WHERE Publisher = 'Springer')
```

و در نهایت ستون Name داخل دستور SELECT خارجی برای هر سطر انتخاب شده توسط دستور WHERE خارجی به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد، بنابراین خروجی نهایی پرس و جو به صورت زیر است:

Name
mn1
mn2

بنابراین پرس و جوی فوق، نام اعضای که همه کتاب‌های منتشرشده توسط انتشارات Springer را به امانت برده‌اند، نشان می‌دهد. که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «نام اعضای که همه کتاب‌های انتشارات Springer را امانت گرفته‌اند.»
توجه: دقت کنید که فرم اصلاح نشده گزاره الف خطای نحوی و کامپایلری دارد و مطابق پرس و جوی مطرح شده در صورت سوال نیست؛ البته فرم اصلاح شده آنرا بررسی کردیم.
مطابق پرس و جوی مطرح شده در گزاره ب داریم:

راه حل اول:

```
SELECT Name
FROM Member
WHERE NOT EXISTS ((SELECT ISBN
FROM Book
WHERE Publisher = 'Springer')
EXCEPT
(SELECT ISBN
FROM Borrow
WHERE Borrow.MID = Member.MID))
```

کارکرد قطعه پرس و جوی زیر از پرس و جوی فوق به صورت زیر است:

```
SELECT ISBN
FROM Book
WHERE Publisher = 'Springer'
```

«شماره‌ی همه‌ی کتاب‌های منتشر شده توسط انتشارات Springer»

همچنین کارکرد قطعه پرس و جوی زیر از پرس و جوی فوق به صورت زیر است:

```
SELECT ISBN
FROM Borrow
WHERE Borrow.MID = Member.MID
```

«شماره کتاب‌های به امانت برده شده توسط یک عضو»

در پرس و جوی فوق، در پرانتز داخلی کل کتاب‌های چاپ شده، توسط انتشارات Springer منهای کلیه کتاب‌های به امانت برده شده توسط یک عضو می‌شود و در صورتی که حاصل این تفاضل تهی باشد نام عضو مورد جستجو در خروجی ظاهر می‌شود.

```
name
mn1
mn2
```

راه حل دوم:

پرس و جوی مطرح شده در صورت سوال را در نظر بگیرید:

« نام اعضای که همه کتاب‌های انتشارات Springer را امانت گرفته‌اند.»

توجه: عملگر تقسیم در جبر رابطه‌ای در SQL قابل پیاده‌سازی است. عملگر تقسیم در جبر رابطه‌ای زمانی مورد استفاده قرار می‌گیرد که پرس و جو بخواهد همه حالت‌های یک اتفاق را بررسی کند. مانند نام اعضای که همه کتاب‌های منتشر شده توسط Springer را به امانت برده‌اند. که به امانت بردن کتاب‌ها توسط اعضا، اتفاق و مقسوم است و همه کتاب‌های منتشر شده توسط Springer، حالت‌ها و مقسوم علیه است.

توجه: عملگر تقسیم در جبر رابطه‌ای به فرم زیر در SQL قابل پیاده‌سازی است.

```
SELECT Name
FROM Member
WHERE NOT EXISTS ((SELECT ISBN
FROM Book
WHERE Publisher = 'Springer')
EXCEPT
(SELECT ISBN
FROM Borrow
WHERE Borrow.MID = Member.MID))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«نام اعضای که، وجود نداشته باشد (NOT EXISTS) کتابی از انتشارات Springer که، به امانت برده نشده باشد (EXCEPT). یعنی نام اعضای که همه کتاب‌های انتشارات Springer را امانت گرفته‌اند.»

بنابراین پرس و جوی فوق، نام اعضای که همه کتاب‌های منتشرشده توسط انتشارات Springer را به امانت برده‌اند، نشان می‌دهد. که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «نام اعضای که همه کتاب‌های انتشارات Springer را امانت گرفته‌اند.»
مطابق پرس و جوی مطرح شده در گزاره ج داریم:

```
SELECT Name
FROM Member
WHERE NOT EXISTS (SELECT 1
                   FROM Book
                   WHERE Publisher = 'Springer'
                   And
                   NOT EXISTS (SELECT ISBN
                               FROM Borrow
                               WHERE Borrow.MID = Member.MID
                               And Borrow.ISBN = Book.ISBN))
```

پرس و جوی مطرح شده در صورت سوال را در نظر بگیرید:

« نام اعضای که همه کتاب‌های انتشارات Springer را امانت گرفته‌اند.»

توجه: عملگر تقسیم در جبر رابطه‌ای به فرم زیر در SQL قابل پیاده‌سازی است.

توجه: این پرس و جو به فرم زیر توسط الگوی اول (همه) در SQL قابل پیاده‌سازی است.

الگوی اول (00):

```
SELECT Name
FROM Member
WHERE NOT EXISTS (SELECT 1
                   FROM Book
                   WHERE Publisher = 'Springer'
                   And
                   NOT EXISTS (SELECT ISBN
                               FROM Borrow
                               WHERE Borrow.MID = Member.MID
                               And Borrow.ISBN = Book.ISBN))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«نام اعضای که، وجود نداشته باشد (NOT EXISTS) کتابی از انتشارات Springer که، به امانت برده نشده

باشد (NOT EXISTS). یعنی نام اعضای که همه کتاب‌های انتشارات Springer را امانت گرفته‌اند.»

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول Member، یک بار به طور کامل از ابتدا تا انتهای

جدول Book بررسی می‌گردد، تا مشخص گردد هیچ سطری در جدول Book از انتشارات Springer وجود

ندارد که شرط اتصال Borrow.MID = Member.MID AND Borrow.ISBN = Book.ISBN موجود در

select داخلی را برقرار نکرده باشد، اگر وجود نداشته باشد، سطر مورد نظر از جدول Member در خروجی

نمایش داده می‌شود.

توجه: بعضی آدم‌ها در دنیا هستن که کاری نیست (NOT EXISTS) که، انجام نداده باشن (NOT EXISTS)، یعنی هر کاری بگی انجام دادن. قایقی نیست که سوار نشده باشن، تفریحی نیست که انجام نداده باشن و ... این افراد به عرض زندگی بیشتر توجه داشتن تا طول زندگی!

توجه: شرط لازم برای تبعیت از الگوی اول وجود NOT EXISTS اول و NOT EXISTS دوم است و شرط کافی برای تبعیت از الگوی اول وجود شرط اتصال متناسب و سازگار است.

بنابراین پرس و جوی فوق، نام اعضای که همه کتاب‌های منتشرشده توسط انتشارات Springer را به امانت برده‌اند، نشان می‌دهد. که مطابق پرس و جوی مطرح شده در صورت سوال است.

یعنی «نام اعضای که همه کتاب‌های انتشارات Springer را امانت گرفته‌اند.»

بنابراین فقط و فقط گزاره‌های ب و ج مطابق پرس و جوی صورت سوال است، پس پرواضح است که گزینه سوم پاسخ سوال است.

توجه: سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه سوم را به عنوان پاسخ اعلام کرده بود.

۷۵- گزینه (۱) صحیح است.

دو جدول $R1(A,B)$ و $R2(C,D)$ را در نظر بگیرید:

A	B

جدول $R1$

C	D

جدول $R2$

مطابق فرض مطرح شده در گزینه‌ی اول، داریم:

$R1$ هیچ تاپل تکراری ندارد و $R2$ تهی نیست.

توجه: جدولی که تاپل تکراری ندارد، می‌تواند تهی باشد.

توجه: جدولی که تهی نیست، نمی‌تواند تاپل تکراری داشته باشد.

Select Distinct A,B

From R1,R2

A	B
1	2
3	4

جدول $R1$

C	D
5	6
7	8

جدول $R2$

با توجه به مقادیر جداول فوق، خروجی پرس و جوی گزینه‌ی اول پس از انجام عملگر ضرب دکارتی به صورت زیر است:

A	B	C	D
1	2	5	6
1	2	7	8
3	4	5	6
3	4	7	8

خروجی نهایی پرس و جوی گزینه‌ی اول پس از اجرای دستور `Select Distinct A,B` به صورت زیر است:

A	B
1	2
3	4

توجه: SQL به طور پیش فرض سطرهای تکراری را حذف نمی‌کند، مگر از دستور `Distinct` استفاده شود.

توجه: همانطور که مشاهده می‌شود خروجی عبارت SQL فوق دقیقاً همان رابطه R1 است، بنابراین پُر واضح است که گزینه اول پاسخ سوال است. مطابق فرض مطرح شده در گزینه‌ی دوم، داریم:

R2 هیچ تاپل تکراری ندارد و R1 تهی نیست.

توجه: جدولی که تاپل تکراری ندارد، می‌تواند تهی باشد.

توجه: جدولی که تهی نیست، نمی‌تواند تاپل تکراری داشته باشد.

`Select Distinct A,B`

From R1,R2

A	B	C	D
1	2		
3	4		

جدول R1

جدول R2

با توجه به مقادیر جداول فوق، خروجی پرس و جوی گزینه‌ی دوم پس از انجام عملگر ضرب دکارتی به صورت زیر است:

A	B	C	D

خروجی نهایی پرس و جوی گزینه‌ی دوم پس از اجرای دستور `Select Distinct A,B` به صورت زیر است:

A	B

توجه: همانطور که مشاهده می‌شود خروجی عبارت SQL فوق همان رابطه R1 نیست، بنابراین گزینه دوم

پاسخ سوال نیست.

مطابق فرض مطرح شده در گزینه‌ی سوم، داریم:

R1 هیچ تاپل تکراری ندارد و R2 تهی است.

توجه: جدولی که تاپل تکراری ندارد، می‌تواند تهی باشد.

توجه: جدولی که تهی نیست، نمی‌تواند تاپل تکراری داشته باشد.

Select Distinct A,B

From R1,R2

A	B
1	2
3	4

جدول R1

C	D

جدول R2

با توجه به مقادیر جداول فوق، خروجی پرس و جوی گزینه‌ی سوم پس از انجام عملگر ضرب دکارتی به صورت زیر است:

A	B	C	D

خروجی نهایی پرس و جوی گزینه‌ی سوم پس از اجرای دستور Select Distinct A,B به صورت زیر است:

A	B

توجه: همانطور که مشاهده می‌شود خروجی عبارت SQL فوق همان رابطه R1 نیست، بنابراین گزینه سوم پاسخ سوال نیست.

مطابق فرض مطرح شده در گزینه‌ی چهارم، داریم:

هم R1 و هم R2 هیچ تاپل تکراری ندارند.

توجه: جدولی که تاپل تکراری ندارد، می‌تواند تهی باشد.

توجه: جدولی که تهی نیست، نمی‌تواند تاپل تکراری داشته باشد.

Select Distinct A,B

From R1,R2

A	B
1	2
3	4

جدول R1

C	D

جدول R2

با توجه به مقادیر جداول فوق، خروجی پرس و جوی گزینه‌ی چهارم پس از انجام عملگر ضرب دکارتی به صورت زیر است:

A	B	C	D

خروجی نهایی پرس و جوی گزینه‌ی چهارم پس از اجرای دستور `Select Distinct A, B` به صورت زیر است:

A	B

توجه: همانطور که مشاهده می‌شود خروجی عبارت SQL فوق همان رابطه R1 نیست، بنابراین گزینه چهارم پاسخ سوال نیست.
توجه: سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه اول را به عنوان پاسخ اعلام کرده بود.

۷۶- گزینه (۳) صحیح است.

سه جدول Student، Course و Takes با مقادیر زیر را در نظر بگیرید:

Sid	Sname	age	Sid	Cid	Grade	Cid	CName	Credits
s1	sn1		s1	c1	10	c1	Database	1
s2	sn2		s1	c2	12	c2	Math	2
s3	sn3		s2	c1	14	Course C1		
Student			s2	c2	16			
			s3	c1	18			
			Takes T1					

مطابق پرس و جوی مطرح شده در گزاره I، داریم:

```
I. SELECT T1.sid
FROM Course C1 , Takes T1
WHERE C1.cid = T1.cid AND C1.cname = 'Database'
```

INTERSECT

```
SELECT T2.sid
FROM Course C2 , Takes T2
WHERE C2.cid = T2.cid AND C2.cname = 'Math'
```

با توجه به جداول فوق، خروجی بخش اول پرس و جوی گزاره‌ی I پس از انجام عملگر ضرب دکارتی و اعمال شرط به صورت زیر است:

Sid	Cid	Grade	Cid	CName	Credits
s1	c1	10	c1	Database	1
s2	c1	14	c1	Database	1
s3	c1	18	c1	Database	1

و در نهایت پس از اعمال دستور `SELECT T1.sid` خروجی نهایی بخش اول پرس و جوی گزاره‌ی I به صورت زیر است:

Sid
s1
s2
s3

با توجه به جداول فوق، خروجی بخش دوم پرس و جوی گزاره‌ی I پس از انجام عملگر ضرب دکارتی و اعمال شرط به صورت زیر است:

Sid	Cid	Grade	Cid	CName	Credits
s1	c2	12	c2	Math	2
s2	c2	16	c2	Math	2

و در نهایت پس از اعمال دستور `SELECT T2.sid` خروجی نهایی بخش دوم پرس و جوی گزاره‌ی I به صورت زیر است:

Sid
s1
s2

در نهایت پس از انجام عملگر `INTERSECT` خروجی پرس و جو به صورت زیر خواهد بود:

Sid	INTERSECT	Sid	=	Sid
s1		s1		s1
s2		s2		s2
s3				

بنابراین گزاره‌ی I مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «شماره دانشجویی، دانشجویانی که هم در درس Database و هم در درس Math ثبت نام کرده‌اند را استخراج می‌کند.»
مطابق پرس و جوی مطرح شده در گزاره II، داریم:

```

II. SELECT T1.sid
FROM Course C1, Takes T1
WHERE C1.cid = T1.cid AND C1.cname = 'Database'
AND T1.sid IN (SELECT T2.sid
FROM Course C2 , Takes T2
WHERE C2.cid = T2.cid AND C2.cname = 'Math')

```

با توجه به وجود دستور `in`، زیر پرس و جوی داخلی فوق یک `Normal Subquery` است، یعنی ابتدا زیر پرس و جوی داخلی یک بار و برای همیشه اجرا می‌گردد، سپس پرس و جوی خارجی به ازای حرکت در هر یک از سطرها، از مقادیر زیر پرس و جوی داخلی استفاده می‌کند.

در پرس وجوی فوق به ازای حرکت در هر سطر از جدول حاصل از ضرب دکارتی Course C1, Takes T1 و شرط اعمال شده در پرس وجوی خارجی مجموعه جلوی in بررسی می گردد که آیا Sid داخل این مجموعه قرار دارد یا خیر. اگر قرار داشت سطر مورد نظر از جدول مذکور در پرس و جوی خارجی در خروجی نمایش داده می شود.

توسط دستور where در فرم زیر:

```
SELECT T1.sid
FROM Course C1, Takes T1
WHERE C1.cid = T1.cid AND C1.cname = 'Database'
AND T1.sid IN (...)
```

برای هر سطر از جدول مذکور کل مجموعه جلوی in که حاصل یک مقایسه می باشد، بررسی می گردد، اگر sid موجود در هر سطر، داخل sidهای مجموعه جلوی in بود، آنگاه شرط جلوی where که همان in است، TRUE می گردد و سطر مورد نظر از جدول مذکور انتخاب می گردد و این رویه برای تک تک سطرهای جدول مذکور، تا به انتهای جدول ادامه پیدا می کند. به بیان دیگر این پرس وجو شماره دانشجویانی از جدول مذکور را می دهد که در پرانتز مقابل in قرار دارند. به عبارت دیگر پرس وجوی گزاره‌ی II شماره دانشجویی، دانشجویانی که هم در درس Database و هم در درس Math ثبت نام کرده اند را استخراج می کند.

با توجه به جداول فوق، خروجی زیر پرس و جوی داخلی پس از انجام عملگر ضرب دکارتی و اعمال شرط به صورت زیر است:

Sid	Cid	Grade	Cid	CName	Credits
s1	c2	12	c2	Math	2
s2	c2	16	c2	Math	2

و در نهایت پس از اعمال دستور SELECT T2.sid خروجی به صورت زیر است:

Sid
s1
s2

بنابراین در ادامه پرس و جوی زیر را خواهیم داشت:

```
SELECT T1.sid
FROM Course C1, Takes T1
WHERE C1.cid = T1.cid AND C1.cname = 'Database'
AND T1.sid IN (s1,s2)
```

همانطور که گفتیم برای هر سطر از جدول مذکور کل مجموعه جلوی in که حاصل یک مقایسه می باشد، بررسی می گردد، اگر Sid موجود در هر سطر، داخل Sidهای مجموعه جلوی in بود، آنگاه شرط جلوی where که همان in است، TRUE می گردد و سطر مورد نظر از جدول مذکور انتخاب می گردد و این رویه برای تک تک سطرهای جدول، تا به انتهای جدول ادامه پیدا می کند. به بیان دیگر این پرس و جوی شماره دانشجویانی از جدول مذکور را می دهد که در پرانتز مقابل in قرار دارند. با توجه به جداول فوق، خروجی جدول مذکور به صورت زیر است:

Sid	Cid	Grade	Cid	CName	Credits
s1	c1	10	c1	Database	1
s2	c1	14	c1	Database	1
s3	c1	18	c1	Database	1

خروجی نهایی پرس و جوی فوق پس از انجام عملگر in به ازای هر سطر جدول مذکور به صورت زیر است:

Sid	Cid	Grade	Cid	Cname	Credit	IN	Sid =
s1	c1	10	c1	Database	1		s1
s2	c1	14	c1	Database	1		s2
s3	c1	18	c1	Database	1		

Sid	Cid	Grade	Cid	Cname	Credit
s1	c1	10	c1	Database	1
s2	c1	14	c1	Database	1

و در نهایت پس از اعمال دستور SELECT T1.sid خروجی نهایی به صورت زیر است:

Sid
s1
s2

بنابراین گزاره II مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «شماره دانشجویی، دانشجویانی که هم در درس Database و هم در درس Math ثبت نام کرده‌اند را استخراج می‌کند.»

مطابق پرس و جوی مطرح شده در گزاره III، داریم:

```

III. SELECT T1.sid
FROM Course C1 , Takes T1
WHERE C1.cid = T1.cid AND C1.cname = 'Database'
AND EXISTS (SELECT *
FROM Course C2 , Takes T2
WHERE C2.cid = T2.cid AND C2.cname = 'Math'
AND C2.sid = C1.sid)

```

که البته پرس و جوی مطرح شده در گزاره III خطای نحوی دارد زیرا در $C2.sid = C1.sid$ وجود C1 و C2 به معنی جدول Course است که اصلاً در جدول Course ستون sid وجود ندارد، که فرم اصلاح شده آن به صورت زیر است:

```

III. SELECT T1.sid
FROM Course C1 , Takes T1
WHERE C1.cid = T1.cid AND C1.cname = 'Database'
AND EXISTS (SELECT *
FROM Course C2 , Takes T2
WHERE C2.cid = T2.cid AND C2.cname = 'Math'
AND T2.sid = T1.sid)

```

توجه: در فرم اصلاح شده‌ی گزاره III، شرط اتصال $T2.sid = T1.sid$ جایگزین شرط اتصال $C2.sid = C1.sid$ شده است.

توجه: گزاره III به فرم اصلاح نشده، خطای نحوی دارد و از سوی کامپایلر اجرا نمی‌شود، بنابراین گزاره III مطابق پرس و جوی مطرح شده در صورت سوال نیست.

با توجه به وجود دستور Exists، زیر پرس و جوی داخلی فوق یک Correlated Subquery است، یعنی به ازای حرکت در هریک از سطرهای پرس و جوی خارجی، یکبار به طور کامل از ابتدا تا انتها زیر پرس و جوی داخلی اجرا و بر اساس شرطی که زیر پرس و جوی داخلی را به پرس و جوی خارجی متصل می‌کند، بررسی انجام می‌شود. مانند دو حلقه تو در تو، که به ازای هربار اجرای حلقه خارجی، یکبار به طور کامل حلقه داخلی اجرا می‌گردد. توسط دستور where در فرم زیر:

```
SELECT T1.sid
FROM Course C1, Takes T1
WHERE C1.cid = T1.cid AND C1.cname = 'Database'
AND EXISTS (...)
```

برای هر سطر از جدول حاصل از ضرب دکارتی Course C1, Takes T1 و شرط اعمال شده شرط جلوی Exists که حاصل یک مقایسه می‌باشد، محاسبه می‌گردد، اگر غیرتهی بود، شرط جلوی where که همان Exists است، TRUE می‌گردد و سطر مورد نظر از جدول مذکور انتخاب می‌گردد و این رویه برای تک تک سطرهای جدول مذکور، تا به انتهای جدول ادامه پیدا می‌کند. به بیان دیگر این پرس و جو شماره دانشجویانی را می‌دهد که پرائتز مقابل Exists برای آن‌ها غیرتهی است. این پرائتز هنگامی غیرتهی می‌شود که حاصل مقایسه بیان شده در این پرائتز غیرتهی شود. حاصل این مقایسه در صورتی غیرتهی می‌شود که شماره دانشجویی که درس Database ثبت نام کرده است، درس Math هم ثبت نام کرده باشد. به عبارت دیگر فرم اصلاح شده پرس و جوی گزاره‌ی III شماره دانشجویی، دانشجویانی که هم در درس Database و هم در درس Math ثبت نام کرده‌اند را استخراج می‌کند. که مطابق پرس و جوی مطرح شده در صورت سوال است.

عبارت موجود در جلوی دستور Exists که به صورت زیر است:

```
SELECT *
FROM Course C2 , Takes T2
WHERE C2.cid = T2.cid AND C2.cname = 'Math'
AND T2.sid = T1.sid)
```

توسط دستور $T1.Sid = T2.Sid$ به محیط خارج یعنی جدول حاصل از ضرب دکارتی Course C1, Takes T1 و شرط اعمال شده متصل می‌گردد. حال به ازای حرکت در هر سطر از جدول مذکور، یک بار به طور کامل سطرهای حاصل از ضرب دکارتی دو جدول Course C2 , Takes T2 از ابتدا تا انتها با توجه به شرط اتصال بررسی می‌گردد.

ابتدا برای سطر اول از جدول حاصل از ضرب دکارتی Course C1, Takes T1 با توجه به شرط اتصال داریم:

$T1.Sid = T2.Sid$
SI SI

Sid	Cid	Grade	Cid	Cname	Credit	EXISTS
s1	c1	10	c1	Database	1	
s2	c1	14	c1	Database	1	
s3	c1	18	c1	Database	1	

Sid	Cid	Grade	Cid	CName	Credits
s1	c2	12	c2	Math	2
s2	c2	16	c2	Math	2

جلیوی EXISTS برابر غیرتهی گردید، بنابراین شرط where در پشت EXISTS، TRUE می‌گردد. بنابراین سطر اول از جدول حاصل از ضرب دکارتی Course C1, Takes T1 در خروجی نمایش داده می‌شود. در ادامه برای سطر دوم از جدول حاصل از ضرب دکارتی Course C1, Takes T1 با توجه به شرط اتصال داریم:

$$S2 = S2$$

Sid	Cid	Grade	Cid	Cname	Credit	EXISTS
s1	c1	10	c1	Database	1	
s2	c1	14	c1	Database	1	
s3	c1	18	c1	Database	1	

Sid	Cid	Grade	Cid	CName	Credits
s1	c2	12	c2	Math	2
s2	c2	16	c2	Math	2

جلیوی EXISTS برابر غیرتهی گردید، بنابراین شرط where در پشت EXISTS، TRUE می‌گردد. بنابراین سطر دوم از جدول حاصل از ضرب دکارتی Course C1, Takes T1 در خروجی نمایش داده می‌شود. در ادامه برای سطر سوم از جدول حاصل از ضرب دکارتی Course C1, Takes T1 با توجه به شرط اتصال داریم:

$$S3 = \times$$

Sid	Cid	Grade	Cid	Cname	Credit	EXISTS
s1	c1	10	c1	Database	1	
s2	c1	14	c1	Database	1	
s3	c1	18	c1	Database	1	

Sid	Cid	Grade	Cid	CName	Credits
s1	c2	12	c2	Math	2
s2	c2	16	c2	Math	2

جلیوی EXISTS برابر تهی گردید، بنابراین شرط where در پشت EXISTS، FALSE می‌گردد. بنابراین سطر سوم از جدول حاصل از ضرب دکارتی Course C1, Takes T1 در خروجی نمایش داده نمی‌شود. بنابراین خروجی پرس‌وجو به صورت زیر خواهد بود:

Sid	Cid	Grade	Cid	Cname	Credit	EXISTS
s1	c1	10	c1	Database	1	
s2	c1	14	c1	Database	1	

و در نهایت پس از اعمال دستور SELECT T1.sid خروجی نهایی به صورت زیر است:

Sid
s1
s2

بنابراین فرم اصلاح شده **گزاره‌ی III** مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «شماره دانشجویی، دانشجویانی که هم در درس Database و هم در درس Math ثبت نام کرده‌اند را استخراج می‌کند.»

توجه: دقت کنید که فرم اصلاح نشده گزاره III خطای نحوی و کامپایلری دارد و مطابق پرس و جوی مطرح شده در صورت سوال نیست؛ البته فرم اصلاح شده آنرا بررسی کردیم.

توجه: سازمان سنجش آموزش کشور در کلید اولیه خود ابتدا گزینه چهارم را به عنوان پاسخ اعلام نمود، سپس در کلید نهایی نظر خود را عوض کرد و گزینه سوم را به عنوان پاسخ اعلام کرد، که عمل درستی را انجام داده است.

تست‌های فصل هفتم: SQL دستورات DDL و DCL

۱- روش مناسب برای تعریف محدودیت‌های جامعیت عمومی در SQL چیست؟ (مهندسی IT - دولتی ۸۵)

۱) Create Table ۲) Create Domain ۳) Create Assertion ۴) Triggered Procedures

۲- اگر روی رابطه R (A, B, C, D) دید (View) زیر تعریف شود، کدام یک از گزاره‌ها درست است؟

(مهندسی کامپیوتر - دولتی ۷۲)

Create View V (A, B, C)

As select A, B, sum (D)

From R

Group By A, B

۱) از این دید می‌توان عملیات ذخیره‌سازی را انجام داد.

۲) از این دید نمی‌توان عملیات ذخیره‌سازی را انجام داد.

۳) از این دید می‌توان عملیات ذخیره‌سازی انجام داد ولی عوارض نامطلوب دارد.

۴) از این دید می‌توان عملیات ذخیره‌سازی انجام داد ولی استقلال داده‌ای تضعیف می‌شود.

۳- اگر رابطه ساعت (Watches) به طریق زیر تعریف شده باشد: (مهندسی IT - دولتی ۸۴)

CREATE TABLE Watches (name Char (30), Manf char (30) References Manfs (Manf))

برای آن‌که دستور DML زیر را اجرا کنیم کدام یک از دسترسی‌های ذکر شده لازم است:

INSERT INTO Watches VALUES (Tissot, 'swatch')

SELECT ON Watches (۲)

SELECT ON Manfs (۱)

REFERENCES ON Manfs (۴)

UPDATE ON Watches (۳)

۴- دو رابطه R(A, B, C, D) و R'(A, B, C', D') در بانک اطلاعاتی رابطه‌ای مفروض است و دیدگاه (View) به

نام RR' را به شکل زیر تعریف می‌کنیم: (مهندسی کامپیوتر - دولتی ۷۴)

Create View RR' As

(Select A, B from R)

Union

(Select A, B from R')

کدام گزینه با تعریف فوق صادق است؟

۱) RR'(C', C) ۲) RR'(A, B, A, B) ۳) RR'(A, B) ۴) RR'(C', C)

۵- با توجه به جدول employee، دستور زیر چه کاری انجام می‌دهد؟ (مهندسی کامپیوتر - دولتی ۸۳)

employee (empno, empname, dept, salary): نام کارمند، شماره کارمند، واحد

Grant select, update (empno, empname) on employee to Roya

۱) این دستور مشکل ساختاری دارد.

۲) اجازه به روز در آوردن و نیز انتخاب ستون‌های جدول employee را به Roya می‌دهد.

۳) اجازه به روز در آوردن و نیز انتخاب ستون‌های جدول را از Roya پس می‌گیرد.

۴) اجازه به روز در آوردن empname و empno و نیز انتخاب ستون‌های جدول employee را به Roya

می‌دهد.

۶- در بانک اطلاعاتی تولید کنندگان و قطعات، دستور SQL زیر باعث.... (مهندسی IT - دولتی ۸۶)

```
CREATE VIEW LONDONSUP
As SELECT *
FROM S
WHERE CITY= 'LONDON'
```

- (۱) افزایش سرعت پاسخگویی به برخی از پرس و جوها می شود.
- (۲) خطای زمان اجرا می شود.
- (۳) خطای نحوی (Syntax) می شود.
- (۴) سهولت بیان برخی از پرس و جوها می شود.

۷- کدام دستور با معنای داده شده، در SQL2 وجود ندارد؟ (مهندسی IT - دولتی ۸۶)

- | | |
|----------------------------------|-----------------------------------|
| (۱) Update برای تغییر view | (۲) Delete برای از بین بردن جداول |
| (۳) Revoke برای لغو اجازه دسترسی | (۴) Update برای تغییر جداول پایه |

۸- دستور زیر چه کاری انجام می دهد؟ (مهندسی کامپیوتر - آزاد ۸۷)

```
Revoke update on stud
From Amiri
```

- (۱) اجازه انجام update روی stud را به کاربری به نام Amiri می دهد.
- (۲) فایل سابقه (log file)، دستورات انجام شده روی stud از طرف کاربری به نام Amiri را به روز می کند.
- (۳) باعث می شود سابقه دستورات انجام شده از طرف کاربری به نام Amiri روی stud از نو ساخته شود.
- (۴) اجازه انجام update روی stud را از کاربری به نام Amiri بازپس می گیرد.

۹- دستور زیر چه کاری انجام می دهد؟ (مهندسی IT - آزاد ۸۷)

```
Grant update on stud
To Ahmadi
```

- (۱) اجازه انجام update روی stud را به کاربری به نام Ahmadi می دهد.
- (۲) اجازه انجام update روی stud را از کاربری به نام Ahmadi بازپس می گیرد.
- (۳) فایل شاخص را روی stud برای کاربری به نام Ahmadi به روز می کند.
- (۴) باعث می شود فایل سابقه (log file) برای دستورات انجام شده از طرف کاربری به نام Ahmadi به روز شود.

۱۰- کدام یک از عبارات زیر صحیح است؟ (مهندسی IT - آزاد ۸۷)

- (۱) جدول پایه (Base Table) جدولی است که توسط کاربران ایجاد و حذف می شود.
 - (۲) جدول میانی (Intermediate Table)، جدولی است که توسط سیستم ایجاد، مدیریت و حذف می شود و کاربر دخالتی در آن ندارد.
 - (۳) دید (View)، جدولی است که وجود خارجی ندارد ولی می توان روی آن پرس و جو انجام داد.
- (۱ و ۲) (۲ و ۳) (۱ و ۳) (۱ و ۴)

۱۱- در صورتی که یک جدول Drop شود، DBMS به طور خودکار کدام عمل را انجام می دهد؟

(مهندسی کامپیوتر - دولتی ۸۴)

۱) حذف دید (View) های مربوط به آن جدول ۲) حذف کلیدهای خارجی مربوط به آن جدول
 ۳) اصلاح لغت‌نامه داده‌ها (data dictionary) ۴) گزینه‌های ۱ و ۲

۱۲- در صورتی که هنگام تعریف جدول SL دستورات زیر نوشته شده باشد و با توجه به جدول L و SL پس از اجرای دستور زیر آن‌گاه کاردینالیته جدول L کدام است؟ (مهندسی IT - آزاد ۸۵)

Delete from L where unit=3

Create Table SL (sname char[20] not NULL, Lname char [20] not NULL,

primary key (Sname, Lname),

foreign key (Lname) references L (Lname),

on delete restrict

on update cascade)

Sname	Lname	Lname	unit
احمدی	ریاضی	ریاضی	4
حسینی	فیزیک	پایگاه	3
رحمانی	پایگاه	تربیت بدنی	1
عباسی	پایگاه	شبکه	3
		فیزیک	3

جدول SL	جدول L
4 (۴)	2 (۲)
3 (۳)	5 (۱)

۱۳- جدول Customers (name, age) و دستور زیر را در نظر بگیرید:

Create Assertion Check-Customer Check

(Not Exists ((Select * From Customers) Except

(Select * From Customers Where name= name OR age = age)))

(مهندسی کامپیوتر - دولتی ۸۹)

Assertion بالا چه چیزی را بررسی می‌کند؟

۱) جدول Customers خالی نباشد.

۲) همه تاپل‌های Customers، یا name یکسان داشته باشند یا age یکسان

۳) در هیچ تاپلی از Customers، فیلدهای name یا age مقدار NULL نداشته باشند.

۴) در هیچ تاپلی از Customers، فیلدهای name و age هم‌زمان مقدار NULL نداشته باشند.

۱۴- کدام گزینه در مورد جداول مجازی درست نیست؟ (مهندسی کامپیوتر - آزاد ۸۹ گروه الف)

۱) جدول مجازی وجود خارجی و فیزیکی ندارد.

۲) می‌توان به جدول مجازی دسترسی داشت و آن را تغییر داد.

۳) تغییرات جدول مجازی مستقل از تغییرات جدول اصلی است.

۴) هدف جدول مجازی ایجاد جدول خلاصه از اطلاعات موجود و محدود کردن دید کاربران است.

۱۵- کدام یک از گزینه‌های زیر در مورد جداول مجازی درست است؟ (مهندسی IT - دولتی ۹۶)

۱) تغییرات جدول مجازی از طریق تغییرات در جداول اصلی رخ می‌دهد.

۲) جداول مجازی در حافظه وجود خارجی دارند ولی در دیسک سخت ذخیره نمی‌شوند.

۳) بروزرسانی جداول مجازی در همه شرایط مستقل از نوع تعریف جدول مجازی، امکان‌پذیر است.

۴) هدف از ایجاد جدول مجازی تنها ساخت جداول خلاصه برای افزایش سرعت جستجو است.

۱۶- کدام یک از گزینه‌های زیر برای چک کردن این که آیا وابستگی تابعی $c \rightarrow b$ در رابطه $R(a,b,c,d)$ رعایت می‌شود غلط است؟
(مهندسی IT - دولتی ۹۳)

(۱)

```
create assertion b-to-c check (
not exists
(select b
from r
group by b
having count (c)>1
)
)
```

(۲)

```
create assertion b-to-c check (
not exists
(select b
from r as r1 , r as r2
where r1.b=r2.b AND not (r1.c=r2.c)
)
)
```

(۳) کافی است که در زمان ایجاد جدول (create table)، به شرط آن که b کلید رابطه R نیز باشد، این تعریف را نیز اضافه کنیم: primary key (b)
(۴) کافی است که یک materialized view روی دو صفت b و c تعریف کنیم و مشخص کنیم که b کلید است.

۱۷- کدام یک از موارد زیر تفاوت مفاهیم Type و Domain در پایگاه داده را بیان می‌کند:

(مهندسی کامپیوتر- آزاد ۹۲)

(۱) Domain می‌تواند دارای مقادیر پیش فرض و نیز constraint باشد.

(۲) Type نمی‌تواند دارای مقادیر پیش فرض و نیز constraint باشد.

(۳) Domain دارای خاصیت Strongly Typed است.

(۴) همه موارد

۱۸- دستوراتی که به عنوان عکس‌العمل یک عمل در پایگاه داده به فرم خودکار اجرا می‌شود، نامیده می‌شود.
(مهندسی کامپیوتر- آزاد ۹۲)

(۱) Grant (۲) Assertion (۳) Trigger (۴) Stored procedure

۱۹- کدام یک از عبارات زیر در رابطه با مفهوم (View) نادرست می‌باشد؟
(مهندسی کامپیوتر- آزاد ۹۲)

(۱) اجرای عملیات select و insert روی view امکان‌پذیر است.

(۲) تعریف یک view روی یک view دیگر امکان‌پذیر است.

(۳) هر view دارای ماهیت فیزیکی بوده و می‌تواند روی روابط پایه ایجاد شود.

(۴) تعریف view به صورت بازگشتی امکان‌پذیر نیست.

۲۰- کدام یک از موارد ذیل جزو انواع متنوع مجوزدهی (Authorisation) در سطح پایگاه داده نیست؟

(مهندسی کامپیوتر- آزاد ۹۲)

(۱) مجوز درج داده‌ها (۲) مجوز بروزرسانی داده‌ها

(۳) مجوز کپی کردن داده‌ها (۴) مجوز حذف داده‌ها

۲۱- دو جدول customers (مشتریان) و orders (سفارش‌ها) که با دستورهای زیر ایجاد شده‌اند را در نظر بگیرید:

```
CREATE TABLE customers (cid CHAR(4),
                        cname CHAR(20),
                        PRIMARY KEY (cid))
CREATE TABLE orders (oid CHAR(6),
                    odate CHAR(6),
                    custid CHAR(4),
                    PRIMARY KEY (oid),
                    FOREIGN KEY (custid) REFERENCES customers(cid)
                    ON DELETE CASCADE)
```

در صورتی که تعداد رکوردهای مشتریان برابر 10 و تعداد رکوردهای سفارش‌ها برابر 15 باشد، پس از اجرای دستور زیر:

```
DELETE
FROM customers C
WHERE C.cid='2000'
```

مجموع تعداد رکوردهای دو جدول customers و orders حداکثر و حداقل چه تعداد خواهد بود؟
(مهندسی کامپیوتر - دولتی ۹۴)

(۱) حداکثر: 24 حداقل: 9
(۲) حداکثر: 25 حداقل: 9
(۳) حداکثر: 24 حداقل: 18
(۴) حداکثر: 25 حداقل: 18

۲۲- با وجود رابطه‌های ST، CO و STCO در مورد دیدهای تعریف شده V1 و V2، کدام گزینه از لحاظ تئوری صحیح است؟
(مهندسی IT - دولتی ۹۵)

دانشجویان (STID, STNAME, STLEVEL, STMAJOR)

درس‌ها (COID, CONAME, CREDIT)

درس‌های انتخاب شده توسط دانشجویان (STID, COID, GRADE)

CREATE VIEW V1 AS SELECT ST.STID, STNAME, COID FROM ST JOIN STCO	CREATE VIEW V2 AS SELECT STID, AVG(GRADE) FROM STCO GROUP BY STID
--	--

(۱) دید V1 و دید V2 هر دو پذیرای عملیات ذخیره‌سازی (Updatable) هستند.

(۲) دید V1 و دید V2 هر دو ناپذیرا (Non-Updatable) هستند.

(۳) دید V1 پذیرا ولی دید V2 ناپذیرا است.

(۴) دید V1 ناپذیرا ولی دید V2 پذیرا است.

۲۳- فرض کنید که دید V1 بر روی جداول S و SP بصورت زیر تعریف شده باشد. (مهندسی کامپیوتر - دولتی ۹۶)

S (SID, SName, Level, City)

P (PID, PName, Weight)

SP (SID, PID, QTY)

CREATE VIEW V1

```
AS SELECT SID AS ID, SUM(QTY) AS SQTY
FROM SP JOIN S
WHERE LEVEL > 10
GROUP BY SID
```

دستور زیر در سطح خارجی با نگاشت E/C تبدیل به چه دستوری در سطح ادراکی می‌شود؟

```
SELECT ID, SQTY
FROM V1
WHERE SQTY > 1000
```

(۱)

```
SELECT ID, SQTY
FROM SP JOIN S WHERE LEVEL > 10
GROUP BY SID HAVING SQTY > 1000
```

(۲)

```
SELECT ID, SQTY
FROM SP JOIN S WHERE LEVEL > 10 AND SQTY > 1000
GROUP BY ID
```

(۳)

```
SELECT SID AS ID, SUM(QTY) AS SQTY
FROM SP JOIN S WHERE LEVEL > 10
GROUP BY SID HAVING SUM(QTY) > 1000
```

(۴)

```
SELECT SID AS ID, SUM(QTY) AS SQTY
FROM SP JOIN S WHERE LEVEL > 10
AND SUM(QTY) > 1000
GROUP BY SID
```

۲۴- دستور Assertion زیر چه محدودیتی را بیان می‌کند؟ (مهندس کامپیوتر-دولتی ۹۶)

S (SID, SName, Level, City): تولیدکنندگان

P (PID, PName, Weight): قطعات

SP (SID, PID, QTY): قطعات تولیدی توسط تولیدکنندگان

```
CREATE ASSERTION MYCONST
CHECK (NOT EXISTS (SELECT *
FROM S
WHERE S.LEVEL < 5 AND EXISTS (SELECT *
FROM SP
WHERE SP.SID = S.SID
AND SP.PID = 'p1'))
```

(۱) تولیدکنندگان با سطح کمتر از 5 نباید قطعات با شناسه p1 را تولید کنند.

(۲) فقط تولیدکنندگان با سطح کمتر از 5 می‌توانند قطعات با شناسه p1 را تولید کنند.

(۳) تولیدکنندگان با سطح کمتر از 5 باید حداقل یک قطعه با شناسه p1 را تولید نمایند.

(۴) هر تولیدکننده‌ای که قطعه با شناسه p1 را تولید می‌کند نباید سطحی بیشتر از 5 داشته باشد.

۲۵- با توجه به جدول دانشجو با مشخصات شماره‌ی دانشجویی، نام، رشته و دانشکده، ASSERTION زیر چه محدودیت جامعی را توصیف می‌نماید؟ (مهندسی IT- دولتی ۹۶)

STUDENT (STID, Name, Major, Department)

CREATE ASSERTION MYASSERT

CHECK (NOT EXISTS (SELECT DEPARTMENT
FROM STUDENT
GROUP BY DEPARTMENT
HAVING COUNT (DISTINCT (MAJOR))>1))

(۱) هر دانشجو در چند رشته می‌تواند تحصیل کند.

(۲) در هر دانشکده فقط یک رشته ارائه می‌شود.

(۳) هر رشته فقط در یک دانشکده ارائه می‌شود.

(۴) هر دانشجو یک نام منحصر به فرد دارد.

۲۶- پس از اجرای دستورات زیر (به ترتیب مشخص شده) تعداد سطرهای جدول Person کدام است؟

(مهندسی کامپیوتر-دولتی ۹۷)

Create Table person (personID int, managerID int, level int, Primary key (personID), Foreign key (managerID) References person(personID) on delete CASCADE on update RESTRICT,)	دستور اول
insert into person values (14,NULL,1), (12,14,2), (11,12,3), (13,12,3), (15,12,3), (10,14,2), (17,10,3), (18,10,3)	دستور دوم
Delete from person where level = 2	دستور سوم

0 (۴)

1 (۳)

8 (۲)

6 (۱)

۲۷- فرض کنید اسکیمای جدول‌های E1، E2 و E3 مطابق با دستورات زیر بوده و تعداد سطرهای هر کدام به ترتیب، M، N و K باشد. پس از اجرای دستور زیر، تعداد سطرهای جدول‌های E1، E2 و E3 به ترتیب M'، N' و K' است. کدام مورد نمی‌تواند رخ دهد؟ (مهندسی IT-دولتی ۹۷)

Delete from E1 where id1=5

create table E1(id1 int, primary key(id1))	دستور اول
create table E2(id2 int, a int, primary key(id2), foreign key(a) references E1(id1) on delete cascade)	دستور دوم
create table E3(id3 int, b int, primary key(id3), foreign key(b) references E2(id2) on delete restrict)	دستور سوم

$$M' = M, N' = N, K' = K \quad (۲) \quad M' = M-1, N' = N-1, K' = K-1 \quad (۱)$$

$$M' < M, N' = N, K' = K \quad (۴) \quad M' < M, N' < N, K' = K \quad (۳)$$

۲۸- چه تعداد از گزاره‌های داده شده درست است؟ (مهندسی کامپیوتر-دولتی ۹۸)

الف) تعداد کلیدهای کاندید یک رابطه از تعداد ابرکلیدهای آن رابطه همواره کمتر یا مساوی است.

ب) اگر رابطه R دارای n خصیصه باشد، آنگاه تعداد ابرکلیدهای این رابطه حداکثر 2ⁿ است.

ج) دستور ALTER Table در SQL، کاتالوگ سیستم را به روز می‌کند.

د) استفاده از View و Index می‌تواند استقلال داده‌ای را افزایش دهد.

۱) یک گزاره (۲) دو گزاره (۳) سه گزاره (۴) چهار گزاره

۲۹- یک فروشگاه زنجیره‌ای را در نظر بگیرید که در سطح شهر شعب مختلف و هر شعبه تعدادی کارمند دارد.

اسکیمای (شِما) جدول‌های شعبه و کارمند داده شده است. فرض کنید، کلید خارجی در جدول کارمند با

حالت On Delete Cascade و On Update Cascade تعریف شده باشد. در حال حاضر شعب با شماره‌های

3، 4، 6 و 8 در پایگاه داده تعریف شده است. مدیریت فروشگاه قصد دارد کارمندان شعبه 3 و 4 را ادغام و

در یک شعبه جدید به شماره 5 منتقل نماید. (مهندسی IT- دولتی ۹۸)

شعبه: Branch (BranchNo, BranchName, Tel, Address)

کارمند: Staff (StaffNo, FName, LName, BranchNo)

دستورات زیر به چه ترتیبی باید اجرا شوند تا کارمندان فعلی شعب 3 و 4 به شعبه 5 منتقل گردند؟ (ترتیب

گزینه‌ها از چپ به راست است.)

نام دستور	دستور SQL
a	Insert into Branch values (5, 'شادی', 021222324, 'تهران')
b	Update Staff set BranchNo=5 where BranchNo=3
c	Delete From Branch where BranchNo=4
d	Update Staff set BranchNo=5 where BranchNo=4
e	Delete From Branch where BranchNo=3

۱) abdce (۲) acebd (۳) bdace (۴) bedca

۳۰- دستور ON UPDATE CASCADE، باعث می‌شود در مورد کدام گزینه اطمینان حاصل کنیم؟

(مهندسی IT - دولتی ۹۹)

(۱) دیدهای ذخیره شده (۲) نرمال بودن داده‌ها (۳) یکپارچگی داده‌ها (۴) تمام موارد بالا

پاسخ تست‌های فصل هفتم: SQL دستورات DDL و DCL

۱- گزینه (۳) صحیح است.

گزینه اول نادرست است. زیرا از دستور Create Table برای تعریف جداول پایگاه داده استفاده می‌شود. گزینه دوم نادرست است. زیرا از دستور Create Domain برای ایجاد دامنه‌های مورد نیاز بهره‌گیری می‌شود. گزینه سوم درست است. زیرا از دستور Create Assertion برای تعریف محدودیت‌های جامعیت عمومی استفاده می‌شود. این دستور به همراه دستوراتی مثل Check، Unique و not NULL برای تعیین محدودیت‌های جامعیتی مورد استفاده قرار می‌گیرد. گزینه چهارم نادرست است. زیرا از دستور Triggered Procedures برای مدل‌سازی عملیات مورد نیاز روی داده‌ها براساس نظر طراح پایگاه داده مورد استفاده قرار می‌گیرد. این دستورات در شرایط خاص جداول از سوی DBMS به طور اتوماتیک اجرا می‌شوند در حالی که Stored Procedure ها از سوی کاربران فراخوانی می‌شوند.

۲- گزینه (۲) صحیح است.

به طور کلی می‌توان محتویات یک جدول مجازی را تغییر داد. (درج، حذف و بروزرسانی) اگر شرایط زیر برقرار باشد:

- جدول مجازی فقط روی یک جدول پایه تعریف شده باشد.
- در تعریف جدول مجازی فقط از ستون‌های جدول پایه استفاده شده باشد (از زیر پرس‌وجوهای تو در تو استفاده نشده باشد).
- ستون‌هایی از جدول پایه که در جدول مجازی حضور ندارند، امکان پذیرش مقدار NULL را داشته باشند.
- در تعریف جدول مجازی از دستور Select Distinct استفاده نشده باشد.
- در تعریف جدول مجازی از توابع آماری استفاده نشده باشد.
- در تعریف جدول مجازی از دستور Group by استفاده نشده باشد.

بنابراین وجود توابع آماری و گروه‌بندی در دید مطرح شده، باعث می‌شود تا نتوان تغییرات دید را در جدول پایه ذخیره نمود.

۳- گزینه (۴) صحیح است.

در SQL می‌توان پنج نوع مجوز را به کاربران اعطاء نمود:

۱- Select: مجوز خواندن اطلاعات

۲- Insert: مجوز درج اطلاعات جدید

۳- Update: مجوز بروزرسانی اطلاعات

۴- Delete: مجوز حذف اطلاعات

۵- References: مجوز ارجاع کلید خارجی یک جدول به کلید کاندید یک جدول مبدأ

برای درج اطلاعات در جدول Watches لازم است تا کاربر مورد نظر مجوز درج را به شکل Insert ON Watches داشته باشد. اما هیچ یک از گزینه‌ها به مجوز درج در جدول Watches اشاره نکرده است. گزینه اول مربوط به مجوز خواندن از جدول Manfs و گزینه دوم مربوط به مجوز خواندن از جدول Watches است. هم‌چنین گزینه سوم مربوط به مجوز بروزرسانی در جدول Watches است.

اما علاوه بر مجوز درج، چون در تعریف جدول Watches به جدول Manfs ارجاع می‌شود باید مجوز ارجاع به Manfs نیز وجود داشته باشد. بنابراین برای اجرای دستور درج داده شده، هم مجوز درج و هم مجوز ارجاع لازم است.
این مجوزها توسط دستور زیر به کاربر مورد نظر اعطا می‌گردد:

```
grant References on Manfs to user name
grant Insert on Watches to user name
```

۴- گزینه (۳) صحیح است.

دستور داده شده ستون‌های A و B از دو جدول R و R' را جدا کرده و با هم اجتماع می‌کند و دید مجازی به نام RR' با دو ستون A و B ایجاد می‌کند. به بیان دیگر دید RR' اجتماع ستون‌های A و B از دو جدول R و R' را استخراج می‌کند. توجه کنید که شرایط سازگاری (تیتراهای مشابه) برای عملگر اجتماع برقرار است. زیرا در صورت سؤال ستون‌های A و B دو جدول R و R' یکسان هستند. البته ستون‌های C و D در جدول R و D' و C' در جدول R' متفاوت هستند که ارتباطی به عملگر اجتماع در دید تعریف شده ندارند.

۵- گزینه (۴) صحیح است.

به طور کلی دستور Grant برای واگذاری مجوز به کاربران مورد استفاده قرار می‌گیرد. ساختار کلی این دستور به صورت زیر است:

<نام کاربر> to <نام جدول> on <فهرست مجوزها> Grant

با توجه به دستور مطرح شده مجوز خواندن اطلاعات (Select) و بروزرسانی اطلاعات (Update)، بر روی ستون‌های empno و empname از جدول employee به کاربری به نام Roya اعطا می‌گردد.

۶- گزینه (۴) صحیح است.

گزینه دوم و سوم نادرست هستند. زیرا دستورات مطرح شده فاقد خطای نحوی و خطای زمان اجرا هستند. گزینه اول نادرست است. زیرا، تعریف و استفاده از جداول مجازی می‌تواند موجب افت کارایی و کاهش سرعت پاسخگویی به پرس‌وجوها شود، زیرا جداول مجازی استقلال وجودی ندارند و وابسته به جداول پایه هستند. در هنگام نیاز به این جداول، DBMS براساس تعریف آن‌ها و محتویات جداول پایه ابتدا آن‌ها را بازسازی می‌کند و سپس مورد استفاده قرار می‌دهد. به این ترتیب سرعت پاسخگویی به پرس‌وجوهایی که به جداول مجازی ارجاع می‌کنند کاهش می‌یابد.
گزینه چهارم درست است. زیرا، دید، ساختار پرس‌وجوها را ساده‌تر و درک آن‌ها را آسان‌تر می‌کند و باعث سهولت در بیان برخی از پرس‌وجوها می‌شود.

۷- گزینه (۲) صحیح است.

برای حذف فیزیکی جداول در SQL2، از دستور Drop Table استفاده می‌گردد، هم‌چنین در حذف فیزیکی جداول، اطلاعات سیستمی مربوط به جداول که در کاتالوگ سیستم (فرهنگ لغات داده‌ها) قرار دارند نیز پاک می‌گردد. برای حذف سطرهای جداول، دستور Delete مورد استفاده قرار می‌گیرد. گزینه‌های دیگر جهت انجام فعالیت‌های ذکر شده در SQL2 وجود دارند.

۸- گزینه (۴) صحیح است.

به طور کلی دستور Revoke برای بازپس‌گیری مجوز از کاربران مورد استفاده قرار می‌گیرد. ساختار کلی این دستور به صورت زیر است:

< نام کاربر > from < نام جدول > on < فهرست مجوزها > Revoke

با توجه به دستور مطرح شده، مجوز بروزرسانی اطلاعات (Update)، بر روی جدول stud از کاربری به نام Amiri بازپس گرفته می‌شود.

۹- گزینه (۱) صحیح است.

به طور کلی دستور Grant برای واگذاری مجوز به کاربران مورد استفاده قرار می‌گیرد. ساختار کلی این دستور به صورت زیر است:

< نام کاربر > to < نام جدول > on < فهرست مجوزها > Grant

با توجه به دستور مطرح شده، مجوز بروزرسانی اطلاعات (Update)، بر روی جدول stud به کاربری به نام Ahmadi اعطا می‌گردد.

۱۰- گزینه (۲) صحیح است.

طبق تعریف، جدولی که توسط طراح پایگاه داده طراحی و با استفاده از دستور Create Table، ایجاد شده است، جدول پایه نامیده می‌شود.

هم‌چنین امکان هرگونه پردازش مثل درج، حذف و بروزرسانی بر روی جدول پایه برای کاربران در چارچوب قواعد امنیتی امکان‌پذیر است. بنابراین گزاره اول، عبارت درستی است. همچنین نتیجه هر پرس‌وجوی SQL توسط DBMS در یک جدول به نام جدول میانی (Intermediat Table) ذخیره می‌شود. مدیریت جداول میانی بر عهده DBMS است و کاربران فقط امکان خواندن از جداول میانی را دارند. انجام عملیات درج، حذف و بروزرسانی بر روی جداول میانی توسط کاربران امکان‌پذیر نیست. پس گزاره دوم نیز، عبارت درستی است.

مثال:

```
select *
from (select *
      from s
      where city = 'c2')
```

خروجی حاصل از select داخلی جهت استفاده در select خارجی، در یک جدول میانی توسط DBMS ذخیره می‌گردد.

در مورد جدول مجازی می‌توان گفت که جدولی که توسط طراح پایگاه داده طراحی گردیده و با استفاده از دستور Create View ...as ایجاد شده است، جدول مجازی نامیده می‌شود. امکان خواندن از جدول مجازی برای کاربر فراهم است. اما هرگونه پردازش مثل درج، حذف و بروزرسانی بر روی جدول مجازی تابع پیش شرط‌هایی می‌باشد که در همه شرایط امکان‌پذیر نیست. بر این اساس گزاره سوم نیز عبارت درستی است.

۱۱- گزینه (۳) صحیح است.

اطلاعات سیستمی مربوط به جداول، مانند نام جدول، نوع ستون‌های جداول در لغت‌نامه داده‌ها ذخیره

می‌گردد. حذف فیزیکی یک جدول توسط دستور Drop باعث می‌شود که به طور خودکار اطلاعات سیستمی (شناسنامه جدول) آن جدول از لغت‌نامه حذف گردد و در نتیجه لغت‌نامه داده‌ها اصلاح شود.

۱۲- گزینه (۴) صحیح است.

برای رفتار ستون کلید خارجی در یک جدول مقصد، در قبال تغییرات کلید کاندید از یک جدول مبدأ گزینه‌های زیر وجود دارد:

نام جدول Create Table

(
نام ستون‌ها

:

foreign key...references...

on delete option

on update option

:

)

فیلد option می‌تواند یکی از موارد زیر باشد:

(restrict) no action

گزینه پیش فرض است و هیچ عملی انجام نمی‌شود. یعنی اگر بر اثر عملیات حذف، درج و بروزرسانی، قوانین جامعیت نقض گردد، این اعمال انجام نمی‌گردد. در واقع زمانی عملیات حذف، درج و بروزرسانی انجام می‌گردد که قوانین جامعیت نقض نگردند.

cascade

اگر سطرهای جدول مرجع حذف یا بروزرسانی شود، کلید خارجی جدولی که به آن ارجاع کرده است نیز حذف یا بروزرسانی خواهد شد.

set NULL

همان روش NULLIFY است که اگر سطرهای جدول مرجع حذف یا بروزرسانی شود، کلید خارجی جدولی که به آن ارجاع کرده است با مقدار NULL پُر می‌شود. با توجه به توضیحات فوق و دستور مطرح شده به شکل زیر:

```
Delete
from L
where unit=3
```

داریم:

در تعریف جدول SL، ستون Lname به عنوان کلید خارجی تعریف شده است که به جدول L ارجاع می‌کند. چون در تعریف جدول SL از دستور on delete restrict استفاده شده است، در عمل امکان حذف هیچ یک از سطرهای جدول L که مورد مراجعه سطرهای جدول SL هستند وجود ندارد. در دستور SQL داده شده حذف دروس سه واحدی از جدول L خواسته شده است. بررسی محتویات جدول L نشان می‌دهد که سه درس سه واحدی (پایگاه، شبکه و فیزیک) در این جدول وجود دارد. اما فقط درس شبکه مورد مراجعه سطرهای جدول SL نیست. بنابراین فقط امکان حذف همین سطر وجود دارد و دو سطر دیگر را نمی‌توان حذف کرد. به این ترتیب بعد از اجرای دستور حذف، چهار سطر، در جدول L باقی خواهد ماند. در صورتی که دستور on delete cascade در تعریف جدول SL ذکر می‌شد، دروس پایگاه، شبکه و

فیزیک از جدول L حذف می‌شدند و کلیه سطرهای ارجاع کننده به این دروس در جدول SL هم حذف می‌شد و در نهایت دو سطر در جدول L باقی می‌ماند.

۱۳- گزینه (۴) صحیح است.

نکته‌ای که در حل سؤال باید از آن استفاده کرد این است که حاصل مقایسه‌ی $NULL=NULL$ مساوی با FALSE است.

با توجه به نکته‌ی فوق، خروجی عبارت select دوم، تمام سطرهای جدول customers به جز سطرهایی است که فیلدهای name و age آنها همزمان NULL است.

دستور select اول نیز تمام سطرهای جدول customers را بر می‌گرداند.

تفاضل خروجی‌های دو دستور select، سطرهایی هستند که فیلدهای name و age آنها همزمان NULL است. اگر چنین سطرهایی در جدول وجود نداشته باشد دستور not exists مقدار TRUE بر می‌گرداند.

۱۴- گزینه (۳) صحیح است.

هر جدول مجازی (یا View) یک دستور select روی یک یا چند جدول است که خلاصه‌ای از اطلاعات آن جداول را به دست می‌دهد و اگر به کاربر فقط اجازه‌ی دسترسی به این جدول مجازی را بدهیم کاربر به بقیه‌ی اطلاعات موجود در جداول اصلی دسترسی ندارد، یعنی دید کاربر محدود می‌شود. (گزینه چهارم) از آنجایی که هر جدول مجازی معادل یک دستور select است، نیازی به ذخیره‌سازی سطرهای خروجی آن دستور select نیست، چون در هر زمان می‌توان آن‌ها را از روی جداول اصلی مربوطه به دست آورد. به علاوه، با هر تغییر در اطلاعات جداول اصلی مربوطه، باید اطلاعات متناظر با آن‌ها در جدول مجازی نیز تغییر کند. بنا به این دلایل، نیازی به ذخیره‌سازی اطلاعات جداول مجازی نیست. پس جداول مجازی وجود خارجی و فیزیکی ندارند. (گزینه اول)

همانند جداول اصلی بانک اطلاعاتی، می‌توان به جداول مجازی نیز دسترسی داشت و اطلاعات آن‌ها را تغییر داد یا آن‌ها را با جداول اصلی یا مجازی دیگر پیوند زد و روی آن‌ها عمل select انجام داد (گزینه دوم). از آنجایی که جداول مجازی وجود خارجی ندارند و خلاصه‌ی اطلاعاتی هستند که از روی جداول اصلی مربوطه به دست می‌آیند، هر تغییر بر روی یک جدول مجازی بر روی جداول مربوط به آن اعمال می‌شود. پس عبارت گزینه سوم اشتباه است.

۱۵- گزینه (۱) صحیح است.

گزینه اول صحیح است: جدول مجازی داده‌های ذخیره شده خاص خود ندارد، بنابراین با انجام تغییرات در جدول یا جداول اصلی، اطلاعات جدول مجازی هم تغییر می‌کند.

گزینه دوم غلط است: جداول مجازی معمولاً وجود خارجی ندارند (یعنی رکوردهای آن‌ها در جایی - نه هاردیسک و نه حافظه RAM - ذخیره نمی‌شوند). هرگونه دسترسی به جداول مجازی توسط DBMS به جداول اصلی مرتبط می‌شود.

گزینه سوم غلط است: در صورتی می‌توان جداول مجازی را بروزرسانی کرد که انجام تغییرات مربوطه روی جداول اصلی بدون ابهام باشد. برخی نسخه‌های SQL این کار را به طور کلی غیرمجاز می‌دانند. اما در SQL2 در شرایط خاصی این کار قابل انجام است (جدول مجازی روی یک جدول باشد، در ساختن آن از Select distinct استفاده نشده باشد، و جدول مربوطه در subquery نیامده باشد).

گزینه چهارم غلط است: هدف اصلی از ایجاد جداول مجازی، ایجاد جداول خلاصه از اطلاعات موجود و محدود کردن دید کاربران است.

۱۶- گزینه (۱) صحیح است.

وابستگی تابعی $c \rightarrow b$ به این معنی است که به هر مقدار از b فقط یک مقدار از c مرتبط باشد. برای مثال در جدول زیر به هر مقدار b فقط یک مقدار c مرتبط است و بنابراین وابستگی تابعی $c \rightarrow b$ رعایت شده است (عدد $b=2$ فقط با $c=3$ مرتبط است و عدد $b=3$ فقط با $c=1$ مرتبط است).

a	b	c	d
1	2	3	4
2	2	3	5
3	3	1	6
4	3	1	7
5	3	1	8

گزینه اول جدول r را برحسب فیلد b گروه‌بندی می‌کند و برای هر گروه (یعنی برای هر مقدار b) تعداد مقادیر c را می‌شمارد. توجه کنید که در دستور `count` جلوی `having` از `distinct` استفاده نشده است و بنابراین مقادیر تکراری چند بار شمرده می‌شوند.

در نتیجه این‌طور تصور می‌شود که برای $b=2$ دو مقدار از c و برای $b=3$ سه مقدار از c وجود دارد و بنابراین به اشتباه تصور می‌شود که وابستگی تابعی $c \rightarrow b$ رعایت نشده است.

اگر دستور `count` با `Distinct` بود. این گزینه صحیح بود: `Having count (distinct c)>1`
گزینه دوم چک می‌کند که در جدول r دو سطر با مقدار b یکسان و مقدار c متفاوت وجود نداشته باشد. یعنی چک می‌کند هر مقدار b به یک مقدار از c مرتبط باشد. پس وابستگی تابعی $c \rightarrow b$ را به درستی بررسی می‌کند.

گزینه سوم صفت b را کلید می‌کند و بنابراین مقادیر b تکراری نخواهند بود، در نتیجه مطمئناً یک مقدار از b نمی‌تواند دوبار ظاهر شود و با دو مقدار متفاوت از c مرتبط شود.

`materialized view` شی‌ای در بانک اطلاعاتی است که حاصل یک `query` را در خود نگه می‌دارد با ایجاد چنین شی‌ای روی صفات b و c و کلید کردن صفت b ، مقادیر b تکراری نخواهند بود و بنابراین هر مقدار b فقط یک بار ظاهر می‌شود و مطمئناً هر مقدار b با یک مقدار c مرتبط خواهد بود.

۱۷- گزینه (۴) صحیح است.

در هنگام تعریف یک `Domain` (با دستور `CREATE DOMAIN`)، بر طبق گرامر `SQL`، می‌توان برای آن `Domain` مقدار پیش فرض و هم‌چنین `constraint` تعریف کرد. بعلاوه، در دستور `CREATE DOMAIN` لازم است نوع مقادیر آن `DOMAIN` حتماً تعیین شوند. بنابراین، `Domain`ها، `Strongly-Typed` هستند. اما در هنگام تعریف یک `Type` (با دستور `CREATE TYPE`)، طبق گرامر `SQL`، امکان تعریف مقدار پیش فرض یا `Constraint` وجود ندارد.

۱۸- گزینه (۳) صحیح است.

`stored procedure` توسط کاربر و `trigger` براساس برقراری شرط‌هایی از قبل تعیین شده به شکل خودکار اجرا می‌شوند.

۱۹- گزینه (۳) صحیح است.

روی view می توان select زد. اما عمل insert را نمی توان روی هر view انجام داد. مثلاً وقتی view روی چند جدول زده شود امکان insert وجود ندارد. یا زمانی که دستور insert منجر به NULL شدن مقدار کلید اصلی یا بخشی از کلید اصلی گردد که سبب نقض قانون جامعیت موجودیت می گردد. بنابراین گزینه اول درست است.

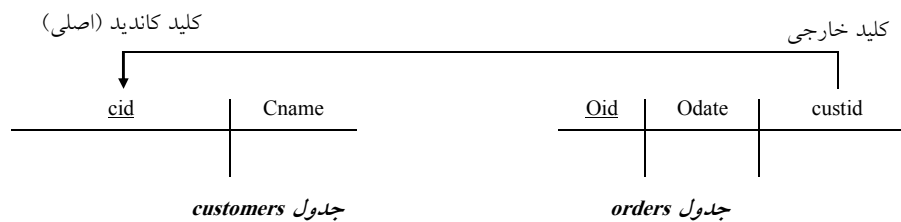
هر view روی روابط پایه یا روی view های دیگر بنا می شود اما viewها ماهیت فیزیکی و وجود خارجی ندارند و رکوردهای آنها به طور فیزیکی ذخیره نمی شوند بلکه در هنگام اجرا و استفاده از آن view، دستورات select مربوط به آن view اجرا می شوند و اطلاعات از روی جداول پایه واکنشی می شوند. بنابراین گزینه سوم نیز نادرست است. گزینه های دوم و چهارم عبارات صحیحی هستند.

۲۰- گزینه (۳) صحیح است.

با دستور GRANAT در زبان SQL می توان مجوزهای Insert، Update، Select و Delete را به یک نفر داد، اما مجوزی برای Copy وجود ندارد.

۲۱- گزینه (۲) صحیح است.

دو جدول customers و orders، با مقادیر زیر را در نظر بگیرید:



مطابق دستورات Create Table برای دو جدول customers و orders ستون cid توسط دستور Primary key(cid) به عنوان کلید اصلی برای جدول customers انتخاب می گردد. همچنین ستون oid توسط دستور Primary key(oid) به عنوان کلید اصلی برای جدول orders انتخاب می گردد. همچنین ستون custid توسط دستور Foreign key (custid) References customers (cid) ON Delete Cascade به عنوان کلید خارجی برای جدول orders انتخاب می گردد. ستون custid از جدول orders به عنوان کلید خارجی به ستون cid از جدول customers مطابق ساختار تعریف جدول orders، به شکل On Delete Cascade ارجاع می کند.

برای رفتار ستون کلید خارجی در یک جدول مقصد، در قبال تغییرات کلید کاندید از یک جدول مبدا گزینه های زیر وجود دارد:

```

Create Table   نام جدول
(
نام ستون ها
:
foreign key ... references ...
on delete option
on update option
:
)
  
```

فیلد option می تواند یکی از موارد زیر باشد:

(restrict) no action

گزینه پیش فرض است و هیچ عملی انجام نمی شود. یعنی اگر بر اثر عملیات حذف، درج و بروزرسانی، قوانین جامعیت نقض گردند، این اعمال انجام نمی گردند. در واقع زمانی عملیات حذف، درج و بروزرسانی انجام می گردند که قوانین جامعیت نقض نگردند.

Cascade

اگر سطرهای جداول مرجع حذف یا بروزرسانی شود، کلید خارجی جدولی که به آن ارجاع کرده است نیز حذف یا بروزرسانی خواهد شد.

Set NULL

همان روش NULLIFY است که اگر سطرهای جدول مرجع حذف یا بروزرسانی شود، کلید خارجی جدولی که به آن ارجاع کرده است با مقدار NULL پُر می شود.

توجه: مطابق فرض سوال تعداد رکوردهای جدول customers برابر 10 و تعداد رکوردهای جدول orders برابر 15 است.

دستور Delete داده شده، ممکن است صفر یا حداکثر یک رکورد را از جدول customers حذف کند (چون ممکن است هیچ رکوردی در جدول customers با 'cid=2000' وجود نداشته باشد. اگر هم رکوردی با 'cid=2000' وجود داشته باشد، چون cid کلید اصلی است، پس فقط همان یک رکورد وجود دارد).

در صورتیکه هیچ رکوردی از جدول customers حذف نشود (یعنی رکوردی با 'cid=2000' در جدول customers وجود نداشته باشد) مطمئناً رکوردی هم از جدول orders حذف نخواهد شد. چون بنابر قانون جامعیت ارجاعی هیچ رکوردی با 'custid=2000' نمی تواند وجود داشته باشد، چون مطابق قانون جامعیت ارجاعی همواره باید مقادیر کلید خارجی زیر مجموعه کلید کاندید (اصلی) در جدول مرجع باشد. بنابراین مجموع تعداد رکوردها ثابت باقی می ماند و در حداکثر مقادیر خود برابر $10 + 15 = 25$ می گردد. از سوی دیگر، در صورتیکه رکوردی با 'cid=2000' در جدول customers وجود داشته باشد با اجرای دستور Delete آن یک رکورد حذف می شود و در جدول customers تعداد 9 رکورد باقی می ماند. حال سه وضعیت می تواند رخ دهد:

وضعیت اول: حالتی که هیچ یک از رکوردهای جدول orders مقدار 'custid=2000' نداشته باشد، در این شرایط هیچ رکوردی از جدول orders حذف نمی شود، بنابراین مجموع رکوردهای باقی مانده برابر $9 + 15 = 24$ خواهد شد.

وضعیت دوم: حالتی که فقط یک رکورد از جدول orders مقدار 'custid=2000' داشته باشد، در این شرایط فقط یک رکورد از جدول orders حذف می شود، بنابراین مجموع رکوردهای باقی مانده برابر $9 + 14 = 23$ خواهد شد.

وضعیت سوم: حالتی که همه رکوردها از جدول orders، مقدار 'custid=2000' داشته باشند، در این شرایط همه رکوردهای جدول orders حذف می شوند، بنابراین مجموع رکوردهای باقی مانده برابر $9 + 0 = 9$ خواهد شد.

پس به طور کلی، مجموع تعداد رکوردهای دو جدول customers و orders در حداکثر مقدار برابر 25 و در حداقل مقدار برابر 9 خواهد بود.

۲۲- گزینه (۳) صحیح است.

جداول مجازی (دیدها) تصویری از جداول حقیقی (پایه) هستند، یعنی توسط سیستم به جداول پایه متصل می‌شوند و وجود خارجی ندارند. دسترسی به آنها از دید کاربر مستقیم ولی از دید سیستم غیرمستقیم است، یعنی سیستم، هرگونه استخراج اطلاعات را از روی جداول پایه انجام می‌دهد. محتوای دید در لحظه اجرای دید، تولید می‌شود. این محتوای تولید شده از سوی دید، در جایی ذخیره نمی‌گردد، بلکه در هر لحظه بر اساس ساختار دید از روی جداول پایه، استخراج می‌گردد.

به طور کلی دو عمل **ذخیره‌سازی** (درج، حذف و بروزرسانی) و **بازیابی** بر روی دیدها انجام می‌شود. که عمل ذخیره‌سازی با قید و شرط انجام می‌شود یعنی نه همیشه، اما عمل بازیابی بی‌قید و شرط انجام می‌شود یعنی همیشه. بنابراین مشکل اصلی در دیدها، عمل ذخیره‌سازی (درج، حذف و بروزرسانی) است که قید و شرط دارد و همیشه قابل انجام نیست. به طور کلی پذیرا بودن ذخیره‌سازی (درج، حذف و بروزرسانی) بر روی دیدها به سه دسته زیر تقسیم می‌گردد:

پذیرا از نظر تئوری و عملی

در این حالت دید از نظر **تئوری** پذیرای ذخیره‌سازی است، یعنی قوانین مدل رابطه‌ای همچون داشتن حداقل یک کلید کاندید و داشتن ستون‌های واقعی و نه مجازی حفظ می‌گردد و همچنین تناظر سطر به سطر مابین دید و جدول پایه حفظ می‌گردد. و از نظر **عملی** هم پذیرای ذخیره‌سازی است به دلیل رعایت قوانین جامعیت بانک در مدل رابطه‌ای در حین اجرا.

موارد زیر پذیرا بودن ذخیره‌سازی بر روی دید را از نظر تئوری تضمین می‌کند:

- در تعریف دید فقط از یک جدول پایه با حداقل یک کلید کاندید استفاده شده باشد.
- اگر رعایت نشود حفظ قوانین مدل رابطه‌ای همچون داشتن حداقل یک کلید کاندید و حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.
- در تعریف دید از دستور `select distinct` استفاده نشده باشد.
- اگر رعایت نشود حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.
- در تعریف دید از دستور `group by` استفاده نشده باشد.
- اگر رعایت نشود حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.
- در تعریف دید از توابع آماری استفاده نشده باشد.
- اگر رعایت نشود حفظ قوانین مدل رابطه‌ای همچون داشتن ستون‌های واقعی و نه مجازی نقض می‌شود.
- در تعریف دید از زیرپرس و جوه‌های تو در تو استفاده نشده باشد.
- اگر رعایت نشود حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.
- مورد زیر پذیرا بودن ذخیره‌سازی بر روی دید را از نظر عملی تضمین می‌کند:**
- در هنگام اجرای ذخیره‌سازی بر روی دید، قوانین جامعیت بانک نقض نگردد.
- سه جدول `ST`، `CO` و `STCO` با مقادیر زیر را در نظر بگیرید:

STID	STNAME	STLEVEL	STMAJOR	COID	CONAME	CREDIT
S1	Sn1			C1	Cn1	
S2	Sn2			C2	Cn2	
S3	Sn3			C3	Cn3	

جدول *ST*جدول *CO*

STID	COID	GRADE
S1	C1	18
S1	C3	20
S2	C2	16
S2	C3	14

جدول STCO

مثال: دید زیر را در نظر بگیرید:

```
Create view GSTCO
AS Select STID, COID
From STCO
Where GRADE>10
```

در تعریف دید GSTCO از یک جدول پایه استفاده شده است، بدنه دید GSTCO به صورت زیر است:

STID	COID
S1	C1
S1	C3
S2	C2
S2	C3

آنچه واضح است، این است که ساختار دید GSTCO از نظر تئوری پذیرای ذخیره‌سازی است، یعنی قوانین مدل رابطه‌ای همچون داشتن حداقل یک کلید کاندید و داشتن ستون‌های واقعی و نه مجازی حفظ می‌گردد و همچنین تناظر سطر به سطر مابین دید و جدول پایه حفظ می‌گردد.

فرض کنید کاربر با اجرای دستور زیر بخواهد یک سطر جدید را در دید GSTCO درج نماید:

```
Insert into GSTCO
Value ('S3', 'C3', 20)
```

از آنجا که دید GSTCO استقلال وجودی ندارد، عملیات درج فوق در واقع روی جدول پایه STCO انجام می‌گردد.

در واقع دستور فوق به هنگام اجرا، به دستور زیر تبدیل می‌گردد:

```
Insert into STCO
Value ('S3', 'C3', 20)
```

در هنگام ذخیره‌سازی عملی بر روی دید GSTCO قوانین جامعیت بانک رعایت می‌شود. پس اجرای دستور ذخیره‌سازی بر روی دید GSTCO به صورت عملی از سوی DBMS پذیرفته می‌شود. بنابراین ذخیره‌سازی بر روی دید GSTCO به صورت عملی قابل انجام است به دلیل رعایت قوانین جامعیت بانک در مدل رابطه‌ای.

پذیرا از نظر تئوری و غیرپذیرا از نظر عملی

در این حالت دید از نظر تئوری پذیرای ذخیره‌سازی است، یعنی قوانین مدل رابطه‌ای همچون داشتن حداقل یک کلید کاندید و داشتن ستون‌های واقعی و نه مجازی حفظ می‌گردد و همچنین تناظر سطر به سطر مابین دید و جدول پایه حفظ می‌گردد. اما از نظر عملی پذیرای ذخیره‌سازی نیست به دلیل عدم رعایت قوانین جامعیت بانک در مدل رابطه‌ای در حین اجرا.

موارد زیر پذیرا بودن ذخیره‌سازی بر روی دید را از نظر تئوری تضمین می‌کند:

- در تعریف دید فقط از یک جدول پایه با حداقل یک کلید کاندید استفاده شده باشد.

اگر رعایت نشود حفظ قوانین مدل رابطه‌ای همچون داشتن حداقل یک کلید کاندید و حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.

- در تعریف دید از دستور `select distinct` استفاده نشده باشد.

اگر رعایت نشود حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.

- در تعریف دید از دستور `group by` استفاده نشده باشد.

اگر رعایت نشود حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.

- در تعریف دید از توابع آماری استفاده نشده باشد.

اگر رعایت نشود حفظ قوانین مدل رابطه‌ای همچون داشتن ستون‌های واقعی و نه مجازی نقض می‌شود.

- در تعریف دید از زیرپرس و جوه‌های تو در تو استفاده نشده باشد.

اگر رعایت نشود حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.

مورد زیر پذیرا بودن ذخیره‌سازی بر روی دید را از نظر عملی تضمین می‌کند:

- در هنگام اجرای ذخیره‌سازی بر روی دید، قوانین جامعیت بانک نقض نگردد.

سه جدول `ST`، `CO` و `STCO` با مقادیر زیر را در نظر بگیرید:

<u>STID</u>	<u>STNAME</u>	<u>STLEVEL</u>	<u>STMAJOR</u>	<u>COID</u>	<u>CONAME</u>	<u>CREDIT</u>
S1	Sn1			C1	Cn1	
S2	Sn2			C2	Cn2	
S3	Sn3			C3	Cn3	

جدول *ST*جدول *CO*

<u>STID</u>	<u>COID</u>	<u>GRADE</u>
S1	C1	18
S1	C3	20
S2	C2	16
S2	C3	14

جدول *STCO*

مثال: دید زیر را در نظر بگیرید:

```
Create view GSTCO
AS Select STID, COID
From STCO
Where GRADE>10
```

در تعریف دید `GSTCO` از یک جدول پایه استفاده شده است، بدنه دید `GSTCO` به صورت زیر است:

<u>STID</u>	<u>COID</u>
S1	C1
S1	C3
S2	C2
S2	C3

آنچه واضح است، این است که ساختار دید `GSTCO` از نظر تئوری پذیرای ذخیره‌سازی است، یعنی قوانین مدل رابطه‌ای همچون داشتن حداقل یک کلید کاندید و داشتن ستون‌های واقعی و نه مجازی حفظ می‌گردد

و همچنین تناظر سطر به سطر مابین دید و جدول پایه حفظ می‌گردد.

فرض کنید کاربر با اجرای دستور زیر بخواهد یک سطر جدید را در دید GSTCO درج نماید:

```
Insert into GSTCO
```

```
Value ('S3', 'C3')
```

از آنجا که دید GSTCO استقلال وجودی ندارد، عملیات درج فوق در واقع روی جدول پایه STCO انجام می‌گردد. اما جدول STCO دارای سه ستون است، درحالیکه، فقط دو ستون در دستور فوق مربوط به جدول GSTCO مقداردهی شده‌است. بنابراین برای ستون GRADE مقدار NULL در نظر گرفته می‌شود. فرض کنید در هنگام تعریف جدول STCO برای ستون GRADE خاصیت NOT NULL در نظر گرفته شده باشد، یعنی حتماً می‌بایست ستون GRADE دارای مقدار باشد و مقدار NULL برای آن قابل پذیرش نیست. بنابراین مطابق قانون جامعیت بانک نباید ستون GRADE با NULL مقداردهی شود.

درواقع دستور فوق به هنگام اجرا، به دستور زیر تبدیل می‌گردد:

```
Insert into STCO
```

```
Value ('S3', 'C3', NULL)
```

در هنگام ذخیره‌سازی عملی بر روی دید GSTCO مطابق قانون جامعیت بانک در جداول نمی‌توان در ستون GRADE از جدول STCO که خاصیت NOT NULL برای آن در نظر گرفته شده‌است، مقدار NULL را درج کرد، پس اجرای دستور ذخیره‌سازی بر روی دید GSTCO به صورت عملی از سوی DBMS پذیرفته نمی‌شود و رد می‌شود. بنابراین ذخیره‌سازی بر روی دید GSTCO به صورت عملی قابل انجام نیست به دلیل عدم رعایت قوانین جامعیت بانک در مدل رابطه‌ای.

همچنین فرض کنید کاربر با اجرای دستور زیر بخواهد یک سطر جدید را در دید GSTCO درج نماید:

```
Insert into GSTCO
```

```
Value ('S10', 'C10', 20)
```

از آنجا که دید GSTCO استقلال وجودی ندارد، عملیات درج فوق در واقع روی جدول پایه STCO انجام می‌گردد.

درواقع دستور فوق به هنگام اجرا، به دستور زیر تبدیل می‌گردد:

```
Insert into STCO
```

```
Value ('S10', 'C10', 20)
```

در هنگام ذخیره‌سازی عملی بر روی دید GSTCO هر چند هر سه ستون جدول STCO به واسطه دستور درج، مقداردهی شده‌اند، اما این بار قانون جامعیت ارجاعی نقض شده است، زیرا ستون‌های STID و COID به عنوان کلیدهای خارجی در جدول STCO تعریف شده‌اند و مطابق تعریف محتوایی کلید خارجی و قانون جامعیت ارجاعی، مجموعه مقادیر کلید خارجی باید همواره زیر مجموعه، مجموعه مقادیر کلید کاندید معادل خود باشد تا ارجاع NULL ایجاد نگردد، یعنی مجموعه مقادیر کلید خارجی STID در جدول STCO باید همواره زیر مجموعه، مجموعه مقادیر کلید کاندید معادل خود یعنی STID در جدول ST باشد و مجموعه مقادیر کلید خارجی COID در جدول STCO باید همواره زیر مجموعه، مجموعه مقادیر کلید کاندید معادل خود یعنی COID در جدول CO باشد، که درج مقادیر S10 و C10 در جدول STCO این قاعده را نقض می‌کند، پس اجرای دستور ذخیره‌سازی بر روی دید GSTCO به صورت عملی از سوی DBMS پذیرفته نمی‌شود و رد می‌شود. بنابراین ذخیره‌سازی بر روی دید GSTCO به صورت عملی قابل انجام نیست به دلیل عدم رعایت قوانین جامعیت بانک در مدل رابطه‌ای.

همچنین فرض کنید کاربر با اجرای دستور زیر بخواهد یک سطر جدید را در دید GSTCO درج نماید:

```
Insert into GSTCO
```

```
Value ('S1', 'C1', 20)
```

از آنجا که دید GSTCO استقلال وجودی ندارد، عملیات درج فوق در واقع روی جدول پایه STCO انجام می‌گردد.

درواقع دستور فوق به هنگام اجرا، به دستور زیر تبدیل می‌گردد:

```
Insert into STCO
```

```
Value ('S1', 'C1', 20)
```

در هنگام ذخیره‌سازی عملی بر روی دید GSTCO هر چند هر سه ستون جدول STCO به واسطه دستور درج، مقداره‌ی شده‌اند، اما این‌بار قانون جامعیت درون رابطه‌ای نقض شده است، زیرا ستون‌های STID و COID باهم به عنوان کلید کاندید در جدول STCO تعریف شده‌اند و مطابق تعریف محتوایی کلید کاندید، مقادیر کلید کاندید باید یکتا باشد، که درج مجدد مقادیر S1 و C1 به طور همزمان در قالب یک سطر در جدول STCO این قاعده را نقض می‌کند، پس اجرای دستور ذخیره‌سازی بر روی دید GSTCO به صورت عملی از سوی DBMS پذیرفته نمی‌شود و رد می‌شود. بنابراین ذخیره‌سازی بر روی دید GSTCO به صورت عملی قابل انجام نیست به دلیل عدم رعایت قوانین جامعیت بانک در مدل رابطه‌ای.

دید V1 مطرح شده در صورت سوال به صورت زیر است:

```
CREATE VIEW V1
```

```
AS SELECT ST.STID, STNAME, COID
```

```
FROM ST JOIN STCO
```

در تعریف دید V1 از الحاق طبیعی استفاده شده است، بدنه دید V1 به صورت زیر است:

ST.STID	STNAME	COID
S1	Sn1	C1
S1	Sn1	C3
S2	Sn2	C2
S2	Sn2	C3

آنچه واضح است، این است که ساختار دید V1 از نظر تئوری پذیرای ذخیره‌سازی است، یعنی قوانین مدل رابطه‌ای همچون داشتن حداقل یک کلید کاندید و داشتن ستون‌های واقعی و نه مجازی حفظ می‌گردد و همچنین تناظر سطر به سطر مابین دید و جدول پایه حفظ می‌گردد.

فرض کنید کاربر با اجرای دستور زیر بخواهد یک سطر جدید را در دید V1 درج نماید:

```
Insert into V1
```

```
Value ('S4', 'Sn4', 'C4')
```

از آنجا که دید V1 استقلال وجودی ندارد، عملیات درج فوق در واقع روی جداول پایه ST و STCO انجام می‌گردد.

درواقع دستور فوق به هنگام اجرا، به دستورات زیر می‌بایست تبدیل گردد:

```
Insert into ST
```

```
Value ('S4', 'Sn4', NULL, NULL)
```

Insert into STCO
Value ('S4', 'C4', NULL)

Insert into CO
Value ('C4', NUUL, NULL)

در هنگام ذخیره‌سازی عملی بر روی دید V1، برای مثال برای حفظ قوانین جامعیت بانک می‌بایست در چند جای مختلف از بانک درجهایی براساس سه دستور فوق انجام گردد، که SQL در عمل توانایی انجام چنین کاری را ندارد، شناخت و اجرای دستورات فوق برای SQL امکان‌پذیر نیست، از آنجاکه عدم اجرای هر یک از دستورات فوق جامعیت بانک را برهم می‌زند، و SQL هم توانایی شناخت و اجرای همه دستورات فوق را ندارد، پس اجرای دستور فوق یعنی ذخیره‌سازی بر روی دید V1 به صورت عملی از سوی DBMS پذیرفته نمی‌شود و رد می‌شود. بنابراین ذخیره‌سازی بر روی دید V1 به صورت عملی قابل انجام نیست به دلیل عدم رعایت قوانین جامعیت بانک در مدل رابطه‌ای.

غیر پذیرا از نظر تئوری و عملی

در این حالت دید از نظر تئوری پذیرای ذخیره‌سازی نیست، یعنی قوانین مدل رابطه‌ای همچون داشتن حداقل یک کلید کاندید و داشتن ستون‌های واقعی و نه مجازی حفظ نمی‌گردد و همچنین تناظر سطر به سطر مابین دید و جدول پایه حفظ نمی‌گردد. همچنین از نظر عملی هم پذیرای ذخیره‌سازی نیست به دلیل عدم پذیرا بودن ذخیره‌سازی در دید از نظر تئوری.

موارد زیر پذیرا بودن ذخیره‌سازی بر روی دید را از نظر تئوری تضمین می‌کند:

- در تعریف دید فقط از یک جدول پایه با حداقل یک کلید کاندید استفاده شده باشد.
- اگر رعایت نشود حفظ قوانین مدل رابطه‌ای همچون داشتن حداقل یک کلید کاندید و حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.
- در تعریف دید از دستور select distinct استفاده نشده باشد.
- اگر رعایت نشود حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.
- در تعریف دید از دستور group by استفاده نشده باشد.
- اگر رعایت نشود حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.
- در تعریف دید از توابع آماری استفاده نشده باشد.
- اگر رعایت نشود حفظ قوانین مدل رابطه‌ای همچون داشتن ستون‌های واقعی و نه مجازی نقض می‌شود.
- در تعریف دید از زیرپرس و جوه‌های تو در تو استفاده نشده باشد.
- اگر رعایت نشود حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.
- مورد زیر پذیرا بودن ذخیره‌سازی بر روی دید را از نظر عملی تضمین می‌کند:
- در هنگام اجرای ذخیره‌سازی بر روی دید، قوانین جامعیت بانک نقض نگردد.
- سه جدول ST، CO و STCO با مقادیر زیر را در نظر بگیرید:

STID	STNAME	STLEVEL	STMAJOR	COID	CONAME	CREDIT
S1	Sn1			C1	Cn1	
S2	Sn2			C2	Cn2	
S3	Sn3			C3	Cn3	

جدول ST

جدول CO

STID	COID	GRADE
S1	C1	18
S1	C3	20
S2	C2	16
S2	C3	14

جدول STCO

دید V2 مطرح شده در صورت سوال به صورت زیر است:

```
CREATE VIEW V2
AS SELECT STID, AVG (GRADE)
FROM STCO
GROUP BY STID
```

در تعریف دید V2 از یک جدول پایه، دستور group by و تابع آماری avg استفاده شده است، بدنه دید V2 به صورت زیر است:

STID	no column name
S1	19
S2	15

آنچه واضح است، این است که دید V2 از نظر تئوری پذیرای ذخیره‌سازی نیست، یعنی قوانین مدل رابطه‌ای همچون داشتن حداقل یک کلید کاندید و داشتن ستون‌های واقعی و نه مجازی حفظ نمی‌گردد و همچنین تناظر سطر به سطر مابین دید و جدول پایه حفظ نمی‌گردد. فرض کنید کاربر با اجرای دستور زیر بخواهد یک سطر جدید را در دید V2 درج نماید:

```
Insert into V2
Value ('S3', 12)
```

در هنگام ذخیره‌سازی عملی بر روی دید V2، برای مثال، مقصد مقدار 12 مشخص نیست، زیرا مقصد آن یک ستون مجازی است و نه یک ستون واقعی، اگر به فرض محال هم درج شود میانگین نمرات دانشجو S3 برابر 12 است حال آنکه دروس آن مشخص نیست، پس اجرای دستور ذخیره‌سازی بر روی دید V2 به صورت عملی از سوی DBMS پذیرفته نمی‌شود و رد می‌شود. بنابراین ذخیره‌سازی بر روی دید V2 به صورت عملی قابل انجام نیست به دلیل عدم پذیرا بودن ذخیره‌سازی در دید V2 از نظر تئوری.

۲۳- گزینه (۳) صحیح است.

جداول مجازی (دیدها) تصویری از جداول حقیقی (پایه) هستند، یعنی توسط سیستم به جداول پایه متصل می‌شوند و وجود خارجی ندارند. دسترسی به آنها از دید کاربر مستقیم ولی از دید سیستم غیرمستقیم است، یعنی سیستم، هرگونه استخراج اطلاعات را از روی جداول پایه انجام می‌دهد. محتوای دید در لحظه‌ای اجرای دید، تولید می‌شود. این محتوای تولید شده از سوی دید، در جایی ذخیره نمی‌گردد، بلکه در هر لحظه بر اساس ساختار دید از روی جداول پایه، استخراج می‌گردد.

VIEW مطرح شده در صورت سوال به صورت زیر است:

```
CREATE VIEW V1
AS SELECT SID AS ID, SUM(QTY) AS SQTY
FROM SP JOIN S
WHERE LEVEL>10
GROUP BY SID
```

مطابق تعریف فوق، محتویات جدول مجازی V1 کاملاً متکی به جداول S و SP است. در واقع می‌توان گفت در هر زمان محتویات V1 عبارتند از نتیجه‌ی اجرای پرس و جوی زیر:

```
SELECT SID AS ID, SUM(QTY) AS SQTY
FROM SP JOIN S
WHERE LEVEL>10
GROUP BY SID
```

که البته پرس و جوی مطرح شده در صورت سوال کمی خطای نحوی دارد، که فرم اصلاح شده آن به صورت زیر است:

```
SELECT S.SID AS ID, SUM(QTY) AS SQTY
FROM SP JOIN S
on S.SID=SP.SID
WHERE LEVEL>10
GROUP BY S.SID
```

جداول زیر را در نظر بگیرید:

SID	SName	Level	City	SID	PID	QTY	PID	...
s1	sn1	10	c1	s1	p1	500	p1	
s2	sn2	20	c2	s1	p2	600	p2	
s3	sn3	30	c3	s2	p1	700		
				s2	p2	800		
				s3	p1	300		
				s3	p2	400		

تولیدکنندگان S

تولید SP

قطعات P

با توجه به جداول فوق، خروجی پرس و جو پس از انجام عملگر JOIN به صورت زیر است:

SID	SName	Level	City	SID	PID	QTY
s1	sn1	10	c1	s1	p1	500
s1	sn1	10	c1	s1	p2	600
s2	sn2	20	c2	s2	p1	700
s2	sn2	20	c2	s2	p2	800
s3	sn3	30	c3	s3	p1	300
s3	sn3	30	c3	s3	p2	400

در ادامه پس از انجام دستور WHERE LEVEL>10 خروجی پرس و جو به صورت زیر است:

SID	SName	Level	City	SID	PID	QTY
s2	sn2	20	c2	s2	p1	700
s2	sn2	20	c2	s2	p2	800
s3	sn3	30	c3	s3	p1	300
s3	sn3	30	c3	s3	p2	400

همچنین در ادامه، پس از انجام دستور GROUP BY S.SID براساس S.SID خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:

S2	S3
P1 700 ...	P1 300 ...
P2 800 ...	P2 400 ...
<u> </u>	<u> </u>
گروه اول	گروه دوم
SUM (QTY)=1500	SUM (QTY)= 700

و در نهایت دستور SUM(QTY) داخل دستور SELECT برای هر گروه به طور مستقل محاسبه می‌گردد و در خروجی پرس و جو قرار می‌گیرد، بنابراین خروجی نهایی جدول مجازی V1 به صورت زیر است:

SID	SQTY
s2	1500
s3	700

توجه: دستور GROUP BY، سرگروه‌ها را، راهی خروجی می‌کند.

در صورت سوال مطرح شده است که دستور زیر در سطح خارجی با نگاشت E/C تبدیل به چه دستوری در سطح ادراکی می‌شود؟

```
SELECT ID, SQTY
FROM V1
WHERE SQTY>1000
```

خروجی پرس و جو فوق پس از انجام دستور WHERE SQTY>1000 بر روی جدول مجازی V1 به صورت زیر است:

SID	SQTY
s2	1500

هنگامی که در یک پرس و جو از جدول مجازی استفاده می‌شود، ابتدا تعریف جدول مجازی مربوطه، در آن پرس و جو جایگذاری می‌شود و سپس پرس و جو اجرا می‌گردد، در واقع می‌توان گفت پرس و جو فوق، عبارتند از نتیجه‌ی اجرای پرس و جو زیر:

```
SELECT S.SID AS ID, SUM(QTY) AS SQTY
FROM SP JOIN S
on S.SID=SP.SID
WHERE LEVEL>10
GROUP BY S.SID
HAVING SUM(QTY)> 1000
```

دقت کنید که دستور WHERE SQTY>1000 در پرس و جو زیر بر روی سطرهای جدول مجازی V1 انجام می‌شود، به عبارت دقیق‌تر دستور WHERE SQTY>1000 در پرس و جو زیر بر روی حاصل جمع تعداد قطعات در گروه‌های حاصل از دستور GROUP BY S.SID موجود در جدول مجازی V1 انجام می‌شود.

توجه: پرس و جو زیر از نظر نحوی صحیح است، زیرا خروجی جدول مجازی V1 تعدادی سطر است و دستور WHERE SQTY>1000 می‌تواند بر روی سطرها اعمال گردد.

```
SELECT ID, SQTY
FROM V1
WHERE SQTY>1000
```

و خروجی زیر نیز حاصل می‌شود:

SID	SQTY
s2	1500

اما هنگامی که قرار است، ساختار کلی پرس و جو بدون جدول مجازی V1 مشاهده شود، از دستور `HAVING SUM(QTY) > 1000` به جای `WHERE SQTY > 1000` برای انتخاب گروه‌ها استفاده می‌شود.

خروجی پرس و جوی زیر به صورت زیر است:

```
SELECT S.SID AS ID, SUM (QTY) AS SQTY
FROM SP JOIN S
on S.SID=SP.SID
WHERE LEVEL>10
GROUP BY S.SID
HAVING SUM (QTY) > 1000
```

همانطور که گفتیم، پس از انجام دستور `GROUP BY S.SID` براساس `S.SID` خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود.

S2	S3
P1 700 ...	P1 300 ...
P2 800 ...	P2 400 ...
<hr/>	<hr/>
گروه اول	گروه دوم

و در نهایت دستور `HAVING SUM(QTY) > 1000` برای هر گروه به طور مستقل اعمال می‌گردد.
توجه: دستور `HAVING` بر روی گروه‌ها، اعمال می‌گردد.

S2	S3
P1 700 ...	P1 300 ...
P2 800 ...	P2 400 ...
<hr/>	<hr/>
گروه اول	گروه دوم
SUM (QTY) = 1500	SUM (QTY) = 700

توجه: دستور `GROUP BY`، سرگروه‌ها را، راهی خروجی می‌کند.

در واقع پرس و جوی فوق، شماره تولیدکنندگانی را در خروجی قرار می‌دهد، که هم سطح آنها بالای 10 باشد و هم مجموع شماره قطعات تولیدی آنها بالای 1000 باشد، خروجی نهایی پرس و جو به صورت زیر است:

SID	SQTY
s2	1500

۲۴- گزینه (۱) صحیح است.

جداول زیر را در نظر بگیرید:

<u>SID</u>	<u>SName</u>	<u>Level</u>	..	<u>SID</u>	<u>PID</u>	<u>QTY</u>	<u>PID</u>	...	
s1	sn1	6		s1	p1	10	p1		
s2	sn2	2		s1	p2	20	p2		
s3	sn3	3		s2	p2	30			
تولیدکنندگان S				تولید SP			قطعات P		

ASSERTION مطرح شده در صورت سوال به صورت زیر است:

```
CREATE ASSERTION MYCONST
CHECK (NOT EXISTS (SELECT *
                   FROM S
                   WHERE S.LEVEL<5 AND EXISTS (SELECT *
                                                FROM SP
                                                WHERE SP.SID = S.SID
                                                AND SP.PID = 'p1')))
```

الگوهای چهارگانه (همه، هیچ، بعضی و حداقل) زیر را در نظر بگیرید:
الگوی اول (00):

```
SELECT SID
FROM S
WHERE NOT EXISTS (SELECT *
                  FROM P
                  WHERE NOT EXISTS (SELECT *
                                    FROM SP
                                    WHERE S.SID = SP.SID
                                    AND P.PID = SP.PID))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:
«شماره تولیدکنندگانی که، وجود نداشته باشد (NOT EXISTS) قطعه‌ای که، تولید نشده باشد (NOT EXISTS). یعنی شماره تولیدکنندگانی که همه قطعات را تولید کرده‌اند.»
با توجه به جداول فوق خروجی پرس و جوی الگوی اول به صورت زیر است:

```
SID
sl
```

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول P بررسی می‌گردد، تا مشخص گردد آیا هیچ سطری در جدول P وجود ندارد که شرط اتصال S.SID = SP.SID AND P.PID = SP.PID موجود در select داخلی را برقرار نکند، اگر وجود نداشته باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

الگوی دوم (01):

```
SELECT SID
FROM S
WHERE NOT EXISTS (SELECT *
                  FROM P
                  WHERE EXISTS (SELECT *
                                FROM SP
                                WHERE S.SID = SP.SID
                                AND P.PID = SP.PID))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«شماره تولیدکنندگانی که، وجود نداشته باشد (**NOT EXISTS**) قطعه‌ای که، تولید شده باشد (**EXISTS**). یعنی شماره تولیدکنندگانی که هیچ قطعه‌ای را تولید نکرده‌اند.»
با توجه به جداول فوق خروجی پرس و جوی الگوی دوم به صورت زیر است:

SID
s3

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول P بررسی می‌گردد، تا مشخص گردد آیا هیچ سطر در جدول P وجود ندارد که شرط اتصال $S.SID = SP.SID \text{ AND } P.PID = SP.PID$ موجود در select داخلی را برقرار کند، اگر وجود نداشته باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

الگوی سوم (10):

```
SELECT SID
FROM S
WHERE EXISTS (SELECT *
              FROM P
              WHERE NOT EXISTS (SELECT *
                                FROM SP
                                WHERE S.SID = SP.SID
                                AND P.PID = SP.PID))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:
«شماره تولیدکنندگانی که، وجود داشته باشد (**EXISTS**) قطعه‌ای که، تولید نشده باشد (**NOT EXISTS**). یعنی شماره تولیدکنندگانی که بعضی قطعات را تولید نکرده‌اند.»
با توجه به جداول فوق خروجی پرس و جوی الگوی سوم به صورت زیر است:

SID
s2
s3

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول P بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول P وجود دارد که شرط اتصال $S.SID = SP.SID \text{ AND } P.PID = SP.PID$ موجود در select داخلی را برقرار نکند، اگر وجود داشته باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

الگوی چهارم (11):

```
SELECT SID
FROM S
WHERE EXISTS (SELECT *
              FROM P
              WHERE EXISTS (SELECT *
                            FROM SP
                            WHERE S.SID = SP.SID
                            AND P.PID = SP.PID))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:
 «شماره تولیدکنندگانی که، وجود داشته باشد (EXISTS) قطعه‌ای که، تولید شده باشد (EXISTS). یعنی شماره تولیدکنندگانی که حداقل یک قطعه را تولید کرده‌اند.»
 با توجه به جداول فوق خروجی پرس و جوی الگوی چهارم به صورت زیر است:

SID

s1

s2

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول P بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول P وجود دارد که شرط اتصال S.SID = SP.SID AND P.PID = SP.PID موجود در select داخلی را برقرار کند، اگر وجود داشته باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

ASSERTION مطرح شده در صورت سوال را مجدداً در نظر بگیرید:

```
CREATE ASSERTION MYCONST
CHECK (NOT EXISTS (SELECT *
FROM S
WHERE S.LEVEL < 5 AND EXISTS (SELECT *
FROM SP
WHERE SP.SID = S.SID
AND SP.PID = 'p1'))
```

توجه: ASSERTION مطرح شده در صورت سوال، مطابق الگوی دوم می‌باشد.
 توجه: دقت کنید که ASSERTIONها دقیقاً همانطور که خوانده می‌شوند عمل می‌کنند، و دقیقاً به همان شکل که خوانده می‌شوند اجازه ذخیره‌سازی داده‌ها را به جداول می‌دهند.

ASSERTION فوق را همانند سورها به صورت زیر بخوانید:
 «تولیدکنندگانی با سطح کمتر از 5 وجود نداشته باشد (NOT EXISTS) که، قطعه‌ای با شناسه p1 را تولید کرده باشد (EXISTS). یعنی تولیدکنندگانی با سطح کمتر از 5 که هیچ قطعه‌ای با شناسه p1 را تولید نکرده‌اند.» برای مثال تولیدکننده s2 با سطح 2 و تولیدکننده s3 با سطح 3 حق تولید قطعه p1 را ندارند.

۲۵- گزینه (۲) صحیح است.

جدول زیر را در نظر بگیرید:

<u>STID</u>	Name	Major	Department
s1	sn1	m1	d1
s2	sn2	m1	d1
s3	sn3	m2	d2
s4	sn4	m2	d2
s5	sn5	m3	d3
s6	sn6	m3	d3

جدول Student

ASSERTION مطرح شده در صورت سوال به صورت زیر است:

```
CREATE ASSERTION MYASSERT
CHECK (NOT EXISTS (SELECT DEPARTMENT
FROM STUDENT
GROUP BY DEPARTMENT
HAVING COUNT (DISTINCT (MAJOR))>1))
```

توجه: دقت کنید که ASSERTION ها دقیقاً همانطور که خواننده می‌شوند عمل می‌کنند، و دقیقاً به همان شکل که خواننده می‌شوند اجازه ذخیره‌سازی داده‌ها را به جداول می‌دهند. در هر درج و بروزرسانی ASSERTION فوق بر روی ورود داده‌ها در جدول STUDENT نظارت می‌کند تا **جامعیت داخلی** تعریف شده توسط طراح بانک اطلاعات برقرار گردد. به این نحو که پس از ورود داده‌ها ابتدا پرس و جوی زیر اجرا می‌شود:

```
SELECT DEPARTMENT
FROM STUDENT
GROUP BY DEPARTMENT
HAVING COUNT (DISTINCT (MAJOR))>1
```

پس از انجام دستور GROUP BY DEPARTMENT براساس DEPARTMENT خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:

d1	d2	d3
s1 sn1 m1	s3 sn3 m2	s5 sn5 m3
<u>s2 sn2 m1</u>	<u>s4 sn4 m2</u>	<u>s6 sn6 m3</u>
گروه اول	گروه دوم	گروه سوم

در نهایت دستور $HAVING\ COUNT\ (DISTINCT\ (MAJOR))>1$ برای هر گروه به طور مستقل اعمال می‌گردد.

توجه: دستور HAVING بر روی گروه‌ها، اعمال می‌گردد.

d1	d2	d3
s1 sn1 m1	s3 sn3 m2	s5 sn5 m3
<u>s2 sn2 m1</u>	<u>s4 sn4 m2</u>	<u>s6 sn6 m3</u>
گروه اول	گروه دوم	گروه سوم

$COUNT(DISTINCT(MAJOR)) = 1$ $COUNT(DISTINCT(MAJOR)) = 1$ $COUNT(DISTINCT(MAJOR)) = 1$

توجه: دستور GROUP BY، سرگروه‌ها را، راهی خروجی می‌کند.

توجه: دستور Distinct، سطرهای تکراری در ستون MAJOR را حذف می‌کند.

در واقع پرس و جوی فوق، شماره دانشکده‌هایی (DEPARTMENT) را در خروجی قرار می‌دهد که بیشتر از یک رشته (MAJOR) ارائه می‌کنند، که مطابق آنچه واضح است در حال حاضر هیچ یک از دانشکده‌ها بیشتر از یک رشته ارائه نکرده‌اند. بنابراین خروجی نهایی پرس و جو یک جدول تهی به صورت زیر است:

DEPARTMENT

حال اگر در ادامه دستور NOT EXISTS انجام گردد، از آنجا که جلوی NOT EXISTS تهی است، در

نتیجه مقدار NOT EXISTS برابر TRUE می‌شود و این بدین معنی است که ASSERTION در حالت امن قرار دارد، یعنی جامعیت داخلی تعریف شده توسط طراح بانک اطلاعات برقرار است. حال اگر دستور درج رکورد ('s7', 'sn7', 'm4', 'd3') از سوی کاربر بر روی جدول STUDENT به صورت زیر انجام شود:

```
INSERT INTO STUDENT (STID,NAME,MAJOR,DEPARTMENT)
VALUES ('s7', 'sn7', 'm4', 'd3')
```

آنگاه ASSERTION تعریف شده توسط طراح بانک اطلاعات جلوی درج رکورد ('s7', 'sn7', 'm4', 'd3') بر روی جدول STUDENT را می‌گیرد، زیرا در این شرایط دانشکده d3 دو رشته m3 و m4 را دارد که این موضوع خلاف جامعیت داخلی تعریف شده توسط طراح بانک اطلاعات است و بانک اطلاعات را ناامن کرده است، بنابراین اجرای دستور درج مذکور، از سوی DBMS پذیرفته نمی‌شود و رد می‌شود. جدول زیر را پس از دستور درج مذکور در نظر بگیرید:

STID	Name	Major	Department
s1	sn1	m1	d1
s2	sn2	m1	d1
s3	sn3	m2	d2
s4	sn4	m2	d2
s5	sn5	m3	d3
s6	sn6	m3	d3
s7	sn7	m4	d3

جدول Student

ASSERTION مطرح شده در صورت سوال به صورت زیر است:

```
CREATE ASSERTION MYASSERT
CHECK (NOT EXISTS (SELECT DEPARTMENT
FROM STUDENT
GROUP BY DEPARTMENT
HAVING COUNT (DISTINCT (MAJOR))>1))
```

در هر درج و بروزرسانی ASSERTION فوق بر روی ورود داده‌ها در جدول STUDENT نظارت می‌کند تا جامعیت داخلی تعریف شده توسط طراح بانک اطلاعات برقرار گردد. به این نحو که پس از ورود داده‌ها ابتدا پرس و جوی زیر اجرا می‌شود:

```
SELECT DEPARTMENT
FROM STUDENT
GROUP BY DEPARTMENT
HAVING COUNT (DISTINCT (MAJOR))>1
```

پس از انجام دستور GROUP BY DEPARTMENT براساس DEPARTMENT خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:

d1	d2	d3
s1 sn1 m1	s3 sn3 m2	s5 sn5 m3
s2 sn2 m1	s4 sn4 m2	s6 sn6 m3
		s7 sn7 m4
گروه اول	گروه دوم	گروه سوم

در نهایت دستور $HAVING COUNT (DISTINCT (MAJOR)) > 1$ برای هر گروه به طور مستقل اعمال می‌گردد.

توجه: دستور $HAVING$ بر روی گروه‌ها، اعمال می‌گردد.

d1	d2	d3
s1 sn1 m1	s3 sn3 m2	s5 sn5 m3
s2 sn2 m1	s4 sn4 m2	s6 sn6 m3
		s7 sn7 m4
گروه اول	گروه دوم	گروه سوم
$COUNT(DISTINCT(MAJOR)) = 1$	$COUNT(DISTINCT(MAJOR)) = 1$	$COUNT(DISTINCT(MAJOR)) = 2$

توجه: دستور $GROUP BY$ ، سرگروه‌ها را، راهی خروجی می‌کند.

توجه: دستور $Distinct$ ، سطرهای تکراری در ستون $MAJOR$ را حذف می‌کند.

در واقع پرس و جوی فوق، شماره دانشکده‌هایی ($DEPARTMENT$) را در خروجی قرار می‌دهد که بیشتر از یک رشته ($MAJOR$) ارائه می‌کنند، که مطابق آنچه واضح است در حال حاضر دانشکده‌ی $d3$ بیشتر از یک رشته ارائه کرده‌است. بنابراین خروجی نهایی پرس و جو به صورت زیر است:

<u>DEPARTMENT</u>
d3

حال اگر در ادامه دستور $NOT EXISTS$ انجام گردد، از آنجا که جلوی $NOT EXISTS$ غیر تهی است، در نتیجه مقدار $NOT EXISTS$ برابر $FALSE$ می‌شود و این بدین معنی است که $ASSERTION$ در حالت ناامن قرار دارد، یعنی جامعیت داخلی تعریف شده توسط طراح بانک اطلاعات برقرار نیست. بنابراین اجرای دستور درج مذکور، از سوی $DBMS$ پذیرفته نمی‌شود و رد می‌شود. و بانک اطلاعات به حالت امن باز می‌گردد. به همین سادگی ...

۲۶- گزینه (۳) صحیح است.

جدول زیر را در نظر بگیرید.

کلید خارجی	کلید کاندید
↓	↓
PersonID	ManagerID
Level	
<i>Person</i>	

کلید خارجی برای ارتباط میان جداول یا ارتباط درون یک جدول مورد استفاده قرار می‌گیرد. به ازای هر مقدار موجود در یک کلید خارجی، باید دقیقاً یک مقدار متناظر در کلید کاندید متناظر آن وجود داشته باشد، در غیر این صورت می‌گوییم، کلید خارجی دارای ارجاع NULL است. به بیان دیگر، مقادیر کلید خارجی همواره باید زیرمجموعه مقادیر کلید کاندید باشد. یک کلید خارجی در یک رابطه هیچگاه نباید ارجاع NULL داشته باشد، این مسأله را به عنوان یک قانون جامعیتی داخلی در مدل رابطه‌ای، می‌شناسیم و آن را **قانون جامعیت ارجاعی** می‌نامیم. هر مقداری که در کلید خارجی وجود دارد، باید دارای مقدار متناظر در کلید کاندید مقصد باشد ولی عکس آن صادق نیست.

دستور زیر بر روی جدول Person تعریف شده است:

Foreign key (managerID) References person (personID)

on delete CASCADE

یعنی کلید خارجی جدول Person یعنی ستون ManagerID به تغییرات (حذف) کلید کاندید جدول Person یعنی ستون PersonID به فرم cascade حساس باشد و واکنش نشان دهد، دقت کنید که درج در جدول Person باید با رعایت و حفظ قانون جامعیت ارجاعی انجام شود، همچنین حذف در جدول Person به واسطه‌ی تعریف کلید خارجی در جدول Person باعث می‌شود در جهت حفظ قانون جامعیت ارجاعی از جدول Person به Person، جدول Person همواره به فرم cascade به تغییرات (حذف) در جدول Person حساس باشد. اما این حساسیت در جدول Person از جنس cascade است، یعنی اگر رکوردی در جدول Person حذف شود که منجر به حذف رکورد دیگری در جدول Person شود، آن حذف رکورد دیگری نیز در جدول Person نیز در جهت حفظ قانون جامعیت ارجاعی پذیرفته می‌شود. همچنین دستور زیر بر روی جدول Person تعریف شده است:

Foreign key (managerID) References person(personID)

on update RESTRICT

یعنی کلید خارجی جدول Person یعنی ستون ManagerID به تغییرات (بروزرسانی) کلید کاندید جدول Person یعنی ستون PersonID به فرم restrict(no action) حساس باشد و واکنش نشان دهد، دقت کنید که درج در جدول Person باید با رعایت و حفظ قانون جامعیت ارجاعی انجام شود، همچنین بروزرسانی در جدول Person به واسطه‌ی تعریف کلید خارجی در جدول Person باعث می‌شود در جهت حفظ قانون جامعیت ارجاعی از جدول Person به Person، جدول Person همواره به فرم restrict به تغییرات (بروزرسانی) در جدول Person حساس باشد. اما این حساسیت در جدول Person از جنس restrict است، یعنی اگر رکوردی در جدول Person بروزرسانی شود که منجر به بروزرسانی رکورد دیگری در جدول Person شود، آن بروزرسانی رکورد دیگری در جدول Person در جهت حفظ قانون جامعیت ارجاعی پذیرفته نمی‌شود و رد می‌شود.

فرم جدول و تعداد رکوردهای جدول Person قبل از انجام دستور Delete به صورت زیر است:

Delete from person where level = 2

جدول زیر را در نظر بگیرید.

کلید خارجی کلید کاندید

↓

<u>PersonID</u>	ManagerID	Level
14	NULL	1
12	14	2
11	12	3
13	12	3
15	12	3
10	14	2
17	10	3
18	10	3

Person

Card(Person) = 8

فرم جدول و تعداد رکوردهای جدول Person پس از انجام دستور Delete به صورت زیر است:

کلید خارجی کلید کاندید

↓

<u>PersonID</u>	ManagerID	Level
14	NULL	1

Person

Card(Person) = 1

پس از اجرای دستور Delete اگر سطر (12,14,2) از جدول Person حذف شود، آنگاه به دلیل فرم حساسیت cascade جدول Person به Person، باید سطرهایی که در ستون ManagerID مقدار 12 دارند شامل سطرهای (11,12,3)، (13,12,3) و (15,12,3)، نیز از جدول Person حذف شود و می‌شود. همچنین پس از اجرای دستور Delete اگر سطر (10,14,2) از جدول Person حذف شود، آنگاه به دلیل فرم حساسیت cascade جدول Person به Person، باید سطرهایی که در ستون ManagerID مقدار 10 دارند شامل سطرهای (17,10,3) و (18,10,3) نیز از جدول Person حذف شود و می‌شود و آنچه در انتها می‌ماند به صورت زیر است:

کلید خارجی کلید کاندید

↓

<u>PersonID</u>	ManagerID	Level
14	NULL	1

Person

۲۷- گزینه (۱) صحیح است.

صورت سوال به این شکل است:

فرض کنید اسکیمای جدول‌های E1، E2 و E3 مطابق با دستورات زیر بوده و تعداد سطرهای هر کدام به

ترتیب، M، N و K باشد. پس از اجرای دستور زیر، تعداد سطرهای جدول‌های E1، E2 و E3 به ترتیب M'، N' و K' است. کدام مورد نمی‌تواند رخ دهد؟

Delete from E1 where id1=5

$$(1) \quad M' = M - 1, N' = N - 1, K' = K - 1$$

گزینه اول نمی‌تواند رخ دهد. جداول زیر را در نظر بگیرید.

کلید خارجی	کلید کاندید	کلید خارجی												
<table style="width: 100%; border-collapse: collapse;"> <tr><td style="border-bottom: 1px solid black; padding: 2px;">id1</td></tr> <tr><td style="padding: 2px;">5</td></tr> <tr><td style="padding: 2px;">3</td></tr> <tr><td style="padding: 2px;">1</td></tr> <tr><td style="padding: 2px;">E1</td></tr> </table>	id1	5	3	1	E1	<table style="width: 100%; border-collapse: collapse;"> <tr><td style="border-bottom: 1px solid black; padding: 2px;">a id2</td></tr> <tr><td style="padding: 2px;">5 8</td></tr> <tr><td style="padding: 2px;">3 6</td></tr> <tr><td style="padding: 2px;">E2</td></tr> </table>	a id2	5 8	3 6	E2	<table style="width: 100%; border-collapse: collapse;"> <tr><td style="border-bottom: 1px solid black; padding: 2px;">b id3</td></tr> <tr><td style="padding: 2px;">8 2</td></tr> <tr><td style="padding: 2px;">E3</td></tr> </table>	b id3	8 2	E3
id1														
5														
3														
1														
E1														
a id2														
5 8														
3 6														
E2														
b id3														
8 2														
E3														

کلید خارجی برای ارتباط میان جداول مورد استفاده قرار می‌گیرد. به ازای هر مقدار موجود در یک کلید خارجی، باید دقیقاً یک مقدار متناظر در کلید کاندید متناظر آن وجود داشته باشد، در غیر این صورت می‌گوییم، کلید خارجی دارای ارجاع NULL است. به بیان دیگر، مقادیر کلید خارجی همواره باید زیرمجموعه مقادیر کلید کاندید باشد. یک کلید خارجی در یک رابطه هیچگاه نباید ارجاع NULL داشته باشد، این مسأله را به عنوان یک قانون جامعیتی داخلی در مدل رابطه‌ای، می‌شناسیم و آن را **قانون جامعیت ارجاعی** می‌نامیم. هر مقداری که در کلید خارجی وجود دارد، باید دارای مقدار متناظر در کلید کاندید مقصد باشد ولی عکس آن صادق نیست.

دستور زیر بر روی جدول E2 تعریف شده است:

foreign key(a) references E1(id1)
on delete cascade

یعنی کلید خارجی جدول E2 یعنی ستون a به تغییرات (حذف) کلید کاندید جدول E1 یعنی ستون id1 به فرم cascade حساس باشد و واکنش نشان دهد، دقت کنید که درج در جدول E1 به خودش ربط دارد و نیاز به واکنش جدول دیگری نیست، اما حذف در جدول E1 به واسطه‌ی تعریف کلید خارجی در جدول E2 باعث می‌شود در جهت حفظ قانون جامعیت ارجاعی از جدول E2 به E1، جدول E2 همواره به فرم cascade به تغییرات (حذف) در جدول E1 حساس باشد. اما این حساسیت در جدول E2 از جنس cascade است، یعنی اگر رکوردی در جدول E1 حذف شود که منجر به حذف در جدول E2 شود، آن حذف در جدول E2 نیز در جهت حفظ قانون جامعیت ارجاعی پذیرفته می‌شود. همچنین دستور زیر بر روی جدول E3 تعریف شده است:

foreign key(b) references E2(id2)
on delete restrict

یعنی کلید خارجی جدول E3 یعنی ستون b به تغییرات (حذف) کلید کاندید جدول E2 یعنی ستون id2 به فرم restrict(no action) حساس باشد و واکنش نشان دهد، دقت کنید که درج در جدول E2 به خودش ربط دارد و نیاز به واکنش جدول دیگری نیست، اما حذف در جدول E2 به واسطه‌ی تعریف کلید

خارجی در جدول E3 باعث می‌شود در جهت حفظ قانون جامعیت ارجاعی از جدول E3 به E2، جدول E3 همواره به فرم restrict به تغییرات (حذف) در جدول E2 حساس باشد. اما این حساسیت در جدول E3 از جنس restrict است، یعنی اگر رکوردی در جدول E2 حذف شود که منجر به حذف در جدول E3 شود، آن حذف در جدول E2 در جهت حفظ قانون جامعیت ارجاعی پذیرفته نمی‌شود و رد می‌شود.

فرم جداول و تعداد رکوردهای جداول E1، E2 و E3 قبل از انجام دستور Delete به صورت زیر است:

Delete from E1 where id1 = 5

	کلید خارجی	کلید خارجی
↓	↓	↓
<u>id1</u>	a <u>id2</u>	b <u>id3</u>
5	5 8	8 2
3	3 6	
1		
<i>E1</i>	<i>E2</i>	<i>E3</i>
$\text{Card}(E1) = M = 3, \quad \text{Card}(E2) = N = 2, \quad \text{Card}(E3) = K = 1$		

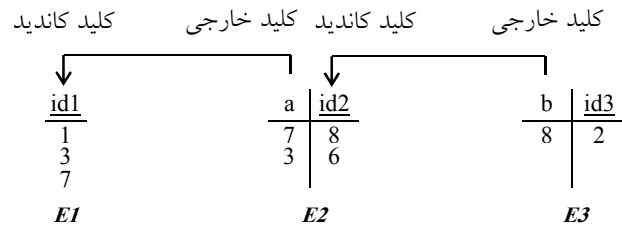
فرم جداول و تعداد رکوردهای جداول E1، E2 و E3 پس از انجام دستور Delete به صورت زیر است:

	کلید خارجی	کلید خارجی
↓	↓	↓
<u>id1</u>	a <u>id2</u>	b <u>id3</u>
5	5 8	8 2
3	3 6	
1		
<i>E1</i>	<i>E2</i>	<i>E3</i>
$\text{Card}(E1) = M' = M = 3, \quad \text{Card}(E2) = N' = N = 2, \quad \text{Card}(E3) = K' = K = 1$		

پس از اجرای دستور Delete هیچ رکوردی حذف نمی‌شود و تغییری در تعداد رکوردهای هیچ یک از جداول ایجاد نمی‌شود. در حالی که قرار بود سطر اول همه‌ی جداول حذف شود تا یک سطر از همه‌ی جداول کم شود. اگر سطر (5) از جدول E1 حذف شود، آنگاه به دلیل فرم حساسیت cascade جدول E2 به E1، باید سطر (5,8) نیز از جدول E2 حذف شود. همچنین اگر سطر (5,8) از جدول E2 حذف شود، آنگاه به دلیل فرم حساسیت restrict جدول E3 به E2، سطر (8,2) از جدول E3 حذف نمی‌شود. و در جهت حفظ قانون جامعیت ارجاعی به شکل بازگشتی، سطر (5,8) هم از جدول E2 حذف نمی‌شود، همچنین در جهت حفظ قانون جامعیت ارجاعی سطر (5) هم از جدول E1 حذف نمی‌شود.

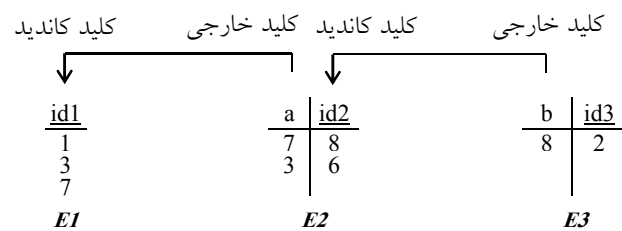
$$M' = M, N' = N, K' = K \quad (۲)$$

گزینه دوم می‌تواند رخ دهد. جداول زیر را در نظر بگیرید.



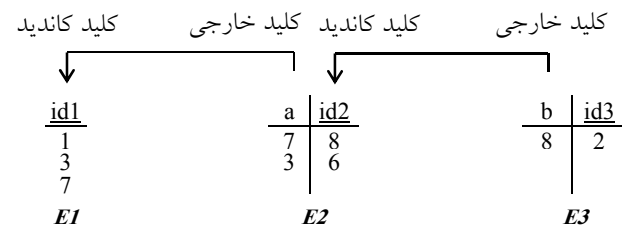
فرم جداول و تعداد رکوردهای جداول E1, E2 و E3 قبل از انجام دستور Delete به صورت زیر است:

Delete from E1 where id1=5



$$\text{Card}(E1) = M = 3, \quad \text{Card}(E2) = N = 2, \quad \text{Card}(E3) = K = 1$$

فرم جداول و تعداد رکوردهای جداول E1, E2 و E3 پس از انجام دستور Delete به صورت زیر است:

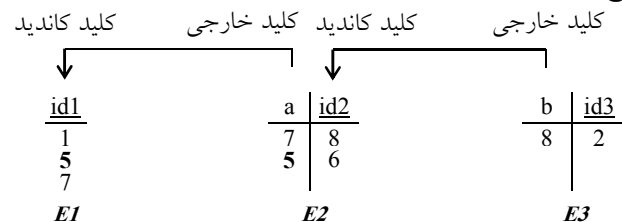


$$\text{Card}(E1) = M' = M = 3, \quad \text{Card}(E2) = N' = N = 2, \quad \text{Card}(E3) = K' = K = 1$$

پس از اجرای دستور Delete هیچ رکوردی حذف نمی‌شود و تغییری در تعداد رکوردهای هیچ یک از جداول ایجاد نمی‌شود. چون جدول E1 اصلاً سطری با مقدار (5) ندارد که بخواهد حذف شود.

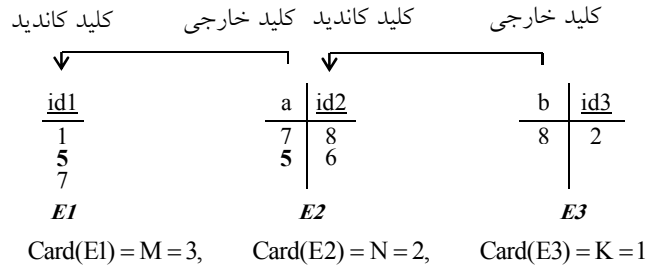
$$M' < M, N' < N, K' = K \quad (۳)$$

گزینه سوم می‌تواند رخ دهد. جداول زیر را در نظر بگیرید.

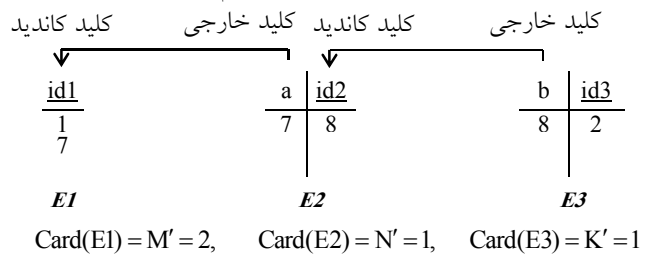


فرم جداول و تعداد رکوردهای جداول E1, E2 و E3 قبل از انجام دستور Delete به صورت زیر است:

Delete from E1 where id1=5



فرم جداول و تعداد رکوردهای جداول E1, E2 و E3 پس از انجام دستور Delete به صورت زیر است:

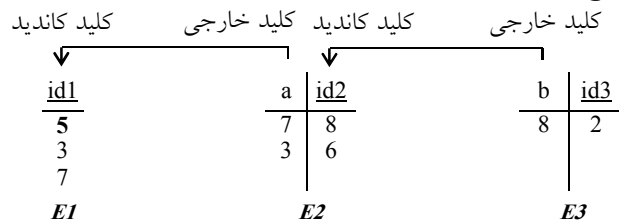


پس از اجرای دستور Delete اگر سطر (5) از جدول E1 حذف شود، آنگاه به دلیل فرم حساسیت cascade جدول E2 به E1، باید سطر (5,6) نیز از جدول E2 حذف شود و می‌شود. همچنین اگر سطر (5,6) از جدول E2 حذف شود، این حذف، حساسیتی بر روی جدول E3 ایجاد نمی‌کند.

$$Card(E1) = M' < M, \quad Card(E2) = N' < N, \quad Card(E3) = K' = K$$

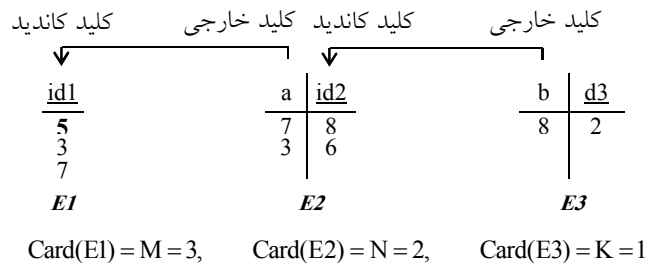
$$M' < M, N' = N, K' = K \quad (۴)$$

گزینه چهارم می‌تواند رخ دهد. جداول زیر را در نظر بگیرید.



فرم جداول و تعداد رکوردهای جداول E1, E2 و E3 قبل از انجام دستور Delete به صورت زیر است:

Delete from E1 where id1=5



فرم جداول و تعداد رکوردهای جداول E1, E2 و E3 پس از انجام دستور Delete به صورت زیر است:

کلید خارجی	کلید کاندید	کلید خارجی
$\frac{id1}{3}$ 7 E1	a $\frac{id2}{8}$ 7 6 3 E2	b $\frac{id3}{2}$ 8 E3

$$Card(E1) = M' = 2, \quad Card(E2) = N' = 2, \quad Card(E3) = K' = 1$$

پس از اجرای دستور Delete اگر سطر (5) از جدول E1 حذف شود، آنگاه این حذف، حساسیتی بر روی جدول E2 ایجاد نمی‌کند.

$$Card(E1) = M' < M, \quad Card(E2) = N' = N, \quad Card(E3) = K' = K$$

۲۸- گزینه (۲) صحیح است.

صورت سوال به این شکل است:

چه تعداد از گزاره‌های داده شده درست است؟

الف) تعداد کلیدهای کاندید یک رابطه از تعداد ابرکلیدهای آن رابطه همواره کمتر یا مساوی است. گزاره اول درست است، زیرا در حالت کلی، اگر رابطه R، دارای n خصیصه باشد، آنگاه تعداد ابرکلیدهای آن حداقل یک و حداکثر $2^n - 1$ است. در جدول تمام کلید، یک جدول فقط و فقط یک ابرکلید دارد و فقط و فقط هم یک کلید کاندید دارد. رابطه تمام کلید مثلاً ممکن است سه ستون داشته باشد، در این حالت یک جدول فقط و فقط یک ابرکلید دارد و فقط و فقط هم یک کلید کاندید دارد. یعنی حداقل یک ابرکلید و یک کلید کاندید دارد و حداکثر هم یک ابرکلید و یک کلید کاندید دارد. در گزاره اول حالت کلی مورد بررسی قرار گرفته است. دقت کنید که حداقل یک ابرکلید و حداکثر $2^n - 1$ ابرکلید، حالت خاص جدول تمام کلید هم پوشش می‌دهد چون بیان حداکثر $2^n - 1$ مقادیر کوچکتر و برابر خودش را پوشش می‌دهد. اگر در یک رابطه با n خصیصه، تک تک خصیصه‌ها به تنهایی کلید کاندید باشد، آنگاه رابطه دارای n کلید کاندید است. بنابراین هر زیرمجموعه غیر تهی از خصیصه‌های این رابطه یک ابرکلید است. که در این حالت تعداد ابرکلیدهای یک رابطه با n خصیصه برابر با $2^n - 1$ است که بیشترین مقدار ممکن در تعداد ابرکلیدهای یک رابطه با n خصیصه است. ابرکلید بدون صفت نداریم، بنابراین حالت $\binom{n}{0}$ اضافه است. همانطور که گفتیم در حالت کلی، یک رابطه دارای n خصیصه، شرایط مختلفی را در تعداد ابرکلید می‌تواند تجربه کند، که حداکثر تعداد ابرکلیدهایی که می‌تواند تجربه کند برابر $\binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n}$ یا $2^n - 1$ است. در واقع عبارت زیر برقرار است:

$$2^n - 1 = \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n}$$

توجه: ابرکلیدی که عضو زائد نداشته باشد، کلید کاندید (**Candidate key**) است، به عبارت دیگر ابرکلید کمینه را کلید کاندید می‌گویند. منظور از ابرکلید کمینه، ابرکلیدی نیست که کمترین تعداد صفت را داشته باشد، بلکه منظور ابرکلیدی است که صفت زائد نداشته باشد.

مثال:

S#	Sname	City	S#	P#	QTY	P#	Pname	Color
S ₁	Sn ₁	C ₁	S ₁	P ₁	10	P ₁	Pn ₁	Red
S ₂	Sn ₂	C ₂	S ₁	P ₂	20	P ₂	Pn ₂	Blue
S ₃	Sn ₃	C ₂	S ₂	P ₁	30	P ₃	Pn ₃	Blue

جدول S

جدول SP

جدول P

S# : ابرکلید است. کلید کاندید نیز هست. (در جدول S)
 (S#,Sname) : ابرکلید است، زیرا خاصیت کلیدی دارد، اما کلید کاندید نیست، زیرا عضو زائد Sname را دارد. در واقع صفت S#، به تنهایی خاصیت کلیدی دارد، بنابراین صفت Sname، عضو زائد است. (در جدول S)
 (S#,P#) : ابرکلید است. کلید کاندید نیز هست. (در جدول SP).

مثال:

شماره ملی: ابرکلید است. کلید کاندید نیز هست.
 (شماره ملی و نام خانوادگی): ابرکلید است. زیرا خاصیت کلیدی دارد، اما کلید کاندید نیست، زیرا عضو زائد نام خانوادگی را دارد. در واقع صفت شماره ملی، به تنهایی خاصیت کلیدی دارد، بنابراین صفت نام خانوادگی، عضو زائد است.
توجه: یک جدول می‌تواند چندین کلید کاندید داشته باشد.

مثال:

شماره دانشجویی	شماره ملی	نام خانوادگی	نام
کلید کاندید (کلید اصلی)	کلید کاندید (کلید فرعی)		

توجه: در مدل رابطه‌ای، هر رابطه حتماً حداقل یک کلید کاندید دارد، زیرا در بدترین شرایط، همه صفات با هم کلید کاندید می‌شوند، که به این رابطه تمام کلید (All key) گفته می‌شود.
توجه: یک رابطه، تحت هیچ شرایطی نمی‌تواند به دلیل استفاده از خاصیت مجموعه‌ای بودن، سطر تکراری داشته باشد. بنابراین یک رابطه، حداقل یک کلید کاندید دارد.
مثال: یک جدول تمام کلید.

a	b	c
1	2	3
1	6	3
1	2	7
8	2	3

نتیجه اینکه مجموعه کلیدهای کاندید همواره زیر مجموعه ابرکلیدها است، بنابراین تعداد کلیدهای کاندید یک رابطه از تعداد ابرکلیدهای آن رابطه همواره کمتر یا مساوی است.

ب) اگر رابطه R دارای n خصیصه باشد، آنگاه تعداد ابرکلیدهای این رابطه حداکثر 2^n است. گزاره دوم نادرست است، زیرا در حالت کلی، اگر رابطه R دارای n خصیصه باشد، آنگاه تعداد ابرکلیدهای آن حداقل یک و حداکثر $2^n - 1$ است.

ج) دستور ALTER Table در SQL، کاتالوگ سیستم را به روز می‌کند.

گزاره سوم درست است، زیرا در یک سیستم بانک اطلاعات، اسامی زیادی مورد استفاده قرار می‌گیرند، از آنجایی که افراد زیاد و متفاوتی در یک مجموعه بانک اطلاعات درگیر هستند، مرجعی برای ایجاد یکنواختی و هماهنگی در نام داده‌ها و معنای آنها ضروری است. این مرجع، دیکشنری داده‌ها نام دارد. در اختصار کاتالوگ سیستم یا دیکشنری داده‌ها شامل تمامی اطلاعات سیستمی مربوط به پایگاه داده‌ها همچون مشخصات سیستمی جداول و سطوح دسترسی کاربران می‌باشد که به طور خودکار توسط DBMS بروزرسانی می‌گردد. به بیان کامل‌تر هرگونه تغییرات حاصل از دستورات DDL در پایگاه داده همچون ایجاد جدول، تغییر جدول، حذف جدول، ایجاد شاخص، حذف شاخص، ایجاد View، حذف View و یا هرگونه تغییرات حاصل از دستورات DML در پایگاه داده همچون درج، حذف و بروزرسانی رکوردها که منجر به تغییر تعداد رکوردها و به تبع آن تغییر اندازه جدول پایگاه داده‌ها می‌شود و یا هرگونه تغییرات حاصل از ایجاد و تغییر سطوح دسترسی توسط دستورات DCL در پایگاه داده همچون تخصیص سطوح دسترسی به کاربران و هر آنچه که مربوط به مشخصات سیستمی پایگاه داده باشد در کاتالوگ سیستم نگهداری می‌شود. در واقع شناسنامه بانک اطلاعات، دیکشنری داده‌ها یا کاتالوگ سیستم است. دیکشنری داده در جایگاه خود، پایگاه داده‌ای سیستمی شامل اطلاعاتی سیستمی در مورد پایگاه داده یک محیط عملیاتی می‌باشد که حاوی «داده‌هایی درباره داده‌ها» است که گاهی اوقات به نام «فراداده» یا «دادگان» به معنی داده در مورد داده معرفی می‌گردد.

کاتالوگ سیستم به واسطه اجرای تمامی دستورات DDL و برخی دستورات DML مانند Delete و Insert و نه Select و Update دستخوش تغییر می‌گردد. دستور ALTER Table یکی از دستورات DDL برای اعمال تغییرات روی جدول پایگاه داده است، دستورات DDL در SQL دستورات ظرف ساز و ساختارساز در پایگاه داده هستند. از آنجا که تمامی دستورات DDL منجر به تغییرات در پایگاه داده می‌شود، بنابراین کاتالوگ سیستم به واسطه اجرای تمامی دستورات DDL دستخوش تغییر می‌گردد، نتیجه اینکه دستور ALTER Table در SQL، کاتالوگ سیستم را به روز می‌کند.

د) استفاده از View و Index می‌تواند استقلال داده‌ای را افزایش دهد.

گزاره چهارم نادرست است، زیرا معماری ANSI برای پایگاه داده‌ها شامل سه لایه زیر است:
۱- لایه خارجی.

۲- لایه ادراکی شامل زیر لایه‌های مدل تحلیل (طراحی ادراکی یا ادراکی عام) و مدل طراحی (طراحی منطقی یا ادراکی خاص).

۳- لایه داخلی (فیزیکی).

یک محصول نرم‌افزاری به واسطه فرآیند تولید نرم‌افزار که شامل فعالیت‌های مدل تحلیل، مدل طراحی،

پیاده‌سازی و تست می‌باشد، ایجاد می‌گردد. کاربران نهایی در لایه خارجی، مدل تحلیل و مدل طراحی در لایه ادراکی و فعالیت پیاده‌سازی در لایه داخلی قرار دارند.

زبان‌های پیاده‌سازی

یک محصول نرم‌افزاری از دو وجه عملکرد (برنامه کاربردی) و داده (بانک اطلاعات) تشکیل می‌شود. انواع زبان‌های برنامه‌سازی به صورت زیر است:

زبان پیاده‌سازی برنامه کاربردی (وجه عملکرد)

برنامه کاربردی نیز مانند بخش داده، حاصل مراحل تحلیل، طراحی و پیاده‌سازی می‌باشد. مرحله پیاده‌سازی برنامه کاربردی توسط یکی از زبان‌های برنامه‌نویسی سطح بالا انجام می‌شود. توجه: به زبان‌های سطح بالا، زبان میزبان یا زبان روالی (Procedural) نیز گفته می‌شود.

زبان پیاده‌سازی بانک اطلاعات (وجه داده)

در بانک اطلاعات از زبان‌های بیانی (Declarative) که به آنها زبان پرس‌وجو (Query Language) نیز گفته می‌شود، استفاده می‌شود. در زبان‌های بیانی کاربر برنامه‌ساز کفایت بگوید چه چیزی لازم دارد تا سیستم برای او ایجاد (مثل جداول) یا استخراج (مثل پرس و جوها) کند. در واقع چگونگی ایجاد جداول یا استخراج پرس‌وجوها از دید کاربر برنامه‌ساز و کاربر نهایی مخفی است.

استقلال داده‌ای

یکی از مهم‌ترین مزایای تکنولوژی پایگاه داده‌ها (مدل مفهومی پایگاه داده)، بلکه مهم‌ترین هدف آن تأمین و افزایش استقلال داده‌ای است، به معنی وابسته نبودن برنامه‌های کاربردی به داده‌های ذخیره شده. استقلال داده‌ای بر دو نوع می‌باشد:

۱- استقلال فیزیکی داده‌ها

به معنی مصونیت برنامه‌های کاربردی در قبال تغییراتی که در سطح فیزیکی (رسانه ذخیره‌سازی) پایگاه داده‌ها بروز می‌کند. یعنی اگر تغییری در ذخیره‌سازی داده‌ها انجام گیرد (برای مثال نوع دیسک عوض شود) برنامه‌های کاربردی هیچ تغییری نکند.

۲- استقلال منطقی داده‌ها

به معنی مصونیت برنامه‌های کاربردی در قبال تعاریف و تغییراتی که در سطح مدل طراحی (مدل رابطه‌ای) پایگاه داده بروز می‌کند. یعنی تعریف و تغییر مدل طراحی بانک (ادراکی خاص یا طراحی منطقی) از دید برنامه‌های کاربردی آنها مخفی بماند.

برای مثال مدل رابطه‌ای از تجربیدی به نام جدول استفاده می‌کند و داده‌ها هر چه باشند در قالب چند جدول ریخته می‌شوند و نحوه ذخیره‌سازی داده‌ها روی رسانه‌ها از دید برنامه کاربردی مخفی است. در حالی که در روش فایلینگ تعاریف مربوط به فایل‌های داده‌ای، در فایل برنامه کاربردی می‌آید. از آنجاکه برنامه‌های کاربردی براساس مدل طراحی بانک (ادراکی خاص یا طراحی منطقی) تعریف می‌شوند، بنابراین به طور بالقوه در معرض تأثیرپذیری از تغییرات در مدل طراحی بانک (ادراکی خاص یا طراحی منطقی) قرار دارند. توجه: در سیستم‌های امروزی، این نوع استقلال هم تا حدی (و نه صددرصد) تأمین شده است.

انواع تغییر در مدل طراحی (طراحی منطقی یا ادراکی خاص)

۱- رشد پایگاه داده‌ها به دلیل مطرح شدن نیازهای جدید مشتری: مانند درج جدول جدید، ترکیب جداول، تجزیه جداول.

۲- سازماندهی مجدد: مانند تغییر در نوع صفات خاصه، تغییر در اندازه صفات.

مثال: اگر جدولی دارای چهار ستون باشد و ستون پنجمی نیز به آن اضافه گردد، در صورتی که برنامه کاربردی سابق نیاز به دستکاری و تغییر نداشته باشد، استقلال منطقی داده‌ها براساس تغییرات نیز لحاظ شده است.

توجه: از آن جا که با حذف جداول، داده‌ها هم از بین می‌رود، بنابراین برنامه‌های کاربردی نسبت به حذف جداول **هیچگاه** استقلال منطقی نخواهند داشت.

همانطور که گفتیم یک محصول نرم‌افزاری از دو وجه **عملکرد** (برنامه کاربردی) و **داده** (بانک اطلاعات) تشکیل شده است، بخش داده (بانک اطلاعات) که با SQL پیاده‌سازی می‌شود به مفاهیم استقلال داده‌ای مرتبط است. ساختار وجه داده توسط دستورات DDL نظیر Create Table، Create View و Create Index و دیگر دستورات آن ایجاد و مدیریت می‌گردد. و مقادیر وجه داده توسط دستورات DML نظیر Insert، Update و Delete و دیگر دستورات آن ایجاد و مدیریت می‌گردد.

دستور Create Table با ساخت مفهوم جدول، کمک به برقراری استقلال داده‌ای از نوع **استقلال فیزیکی داده‌ها** میان یک برنامه کاربردی و داده‌ها می‌کند، به معنی وابسته نبودن برنامه‌های کاربردی به داده‌های ذخیره شده، یعنی همانطور که گفتیم، مدل رابطه‌ای از تجزیدی به نام جدول استفاده می‌کند و داده‌ها هر چه باشند در قالب چند جدول ریخته می‌شوند و نحوه ذخیره‌سازی داده‌ها روی رسانه‌ها از دید برنامه کاربردی مخفی است. دقت کنید که Table بخشی از وجه داده است. در واقع بخش داده از بخش‌های مختلف Table، View و Index تشکیل شده است.

دستور Create View با ساخت مفهوم دید، تا حدی کمک به برقراری استقلال داده‌ای از نوع **استقلال منطقی داده‌ها** میان یک برنامه کاربردی و داده‌ها می‌کند، به معنی وابسته نبودن برنامه‌های کاربردی به داده‌های ذخیره شده، یعنی همانطور که گفتیم، اگر جدولی دارای چهار ستون باشد و ستون پنجمی نیز به آن اضافه گردد، در صورتی که برنامه کاربردی سابق نیاز به دستکاری و تغییر نداشته باشد، استقلال منطقی داده‌ها براساس تغییرات نیز لحاظ شده است. از آنجاکه View روی ساختار قدیم شامل نام جدول قدیم و ستون‌های قدیم ایجاد می‌شود، اگر جدولی دارای چهار ستون باشد و ستون پنجمی نیز به آن اضافه گردد، آنگاه بدون تغییرات در ساختار View بخش داده و به تبع تغییرات در ساختار بخش عملکرد (برنامه کاربردی)، امکان حیات برنامه کاربردی بدون اشکال همچنان وجود دارد و این یعنی View حافظ استقلال داده‌ای از نوع استقلال منطقی داده‌ها است. دقت کنید که View بخشی از وجه داده است. در واقع بخش داده از بخش‌های مختلف Table، View و Index تشکیل شده است.

دستور Create Index با ساخت مفهوم شاخص، هیچ کمکی به برقراری استقلال داده‌ای از نوع **استقلال فیزیکی داده‌ها و استقلال منطقی داده‌ها** میان یک برنامه کاربردی و داده‌ها نمی‌کند، به معنی وابسته نبودن برنامه‌های کاربردی به داده‌های ذخیره شده. مهمترین کاربرد شاخص، افزایش سرعت جستجو و بازیابی اطلاعات است و این مفهوم و کاربرد هیچ ارتباطی به مفهوم استقلال داده‌ای ندارد.

شاخص فقط باعث بالا رفتن سرعت دستیابی به اطلاعات می‌گردد. شاخص‌ها برای بهبود فرآیند جستجو و بازیابی اطلاعات در جداول ایجاد می‌شوند. به عبارت دیگر شاخص‌ها به فرآیند جستجو و بازیابی

اطلاعات، سرعت می‌بخشند. شاخص‌ها باعث می‌شوند موتور جستجوی پایگاه داده کل یک جدول را برای پیدا کردن رکورد یا رکوردهای مورد نظر به طور کامل نگردد. اساساً شاخص‌ها بر روی ستون‌هایی باید تنظیم شود که بیشتر مورد جستجو قرار می‌گیرند.

دو هدف اصلی سیستم ذخیره و بازیابی اطلاعات در پایگاه داده‌ها، اول سرعت عملیات در ذخیره و بازیابی اطلاعات و دوم صرفه‌جویی در مصرف حافظه است. برای مثال کاهش افزونگی محتوایی (طبیعی) توسط نرمال‌سازی جداول منجر به کاهش میزان حافظه مصرفی می‌شود. عمل واکنشی تک تک رکوردها وقت‌گیر است، برای رفع این عیب، شاخص یا Index ابداع شد. برای اینکه جستجو و بازیابی داده‌ها با سرعت و کارایی بیشتر صورت گیرد، از شاخص استفاده می‌شود. شاخص ساختمان داده‌ای است که سیستم مدیریت پایگاه داده‌ها به کمک آن رکوردهای مورد نظر را در یک فایل با سرعت بسیار زیاد پیدا می‌کند و به این ترتیب سرعت پاسخ به پرس و جوها افزایش می‌یابد.

توجه: تعریف و نگهداری شاخص موجب تحمیل سربار حافظه‌ای به سیستم می‌شود. شاخص بر روی هارد دیسک نگهداری می‌شود و هنگام استفاده به حافظه اصلی آورده می‌شود، بنابراین اعمال سیاست شاخص‌گذاری، بر افزایش حجم اطلاعات ذخیره شده بر روی حافظه اصلی و هارد دیسک تاثیر دارد.

توجه: عملیات درج، حذف و بروزرسانی در جدولی که شاخص دارد یعنی خود جدول پایه، نسبت به جدولی که شاخص ندارد زمان بیشتری مصرف می‌کند و به تبع این عملیات کندتر خواهد بود. زیرا شاخص‌ها نیز همگام با جداول پایه خود نیاز به بروزرسانی دارند. بنابراین تنها روی ستون‌هایی شاخص ایجاد می‌گردد، که به تناوب روی آنها جستجو انجام می‌شود. بنابراین هرچه تعداد شاخص‌های یک جدول بیشتر باشد، سرعت Insert، Update و Delete در آن جدول کمتر می‌شود. اما خود شاخص عامل جستجو و بازیابی سریع اطلاعات از یک جدول پایه است.

توجه: شاخص‌گذاری، افزونگی تکنیکی دارد و مقداری از فضای حافظه اصلی و هارد دیسک را اشغال می‌کند. به عبارت دیگر تعریف هر شاخص روی یک جدول هزینه‌ی زیادی را به پایگاه داده تحمیل می‌کند و اگر نرخ تغییرات محتوایی پایگاه داده زیاد باشد این هزینه به صورت قابل توجهی افزایش می‌یابد. بنابراین طراحان پایگاه داده ترجیح می‌دهند که روی هر جدول بیش از یک شاخص تعریف نکنند.

توجه: شاخص‌گذاری، منجر به افزایش سرعت جستجو و بازیابی اطلاعات و به تبع کاهش زمان جستجو و بازیابی اطلاعات می‌شود، نبود شاخص باعث کندی سرعت جستجو و زیادی آن باعث کندی فرآیندهای درج، حذف بروزرسانی رکوردها در جداول پایه می‌شود. بنابراین راز نه در افراط است و نه در تفریط بلکه راز در تعادل است، گاهی هم استفاده‌ی مکرر از گزینه‌های خوب، نتیجه‌ی بد هم می‌تواند به همراه داشته باشد! و به قول معروف در حوزه‌ی سلامتی نیاکان ما گفته‌اند که کم بخور، همیشه بخور...

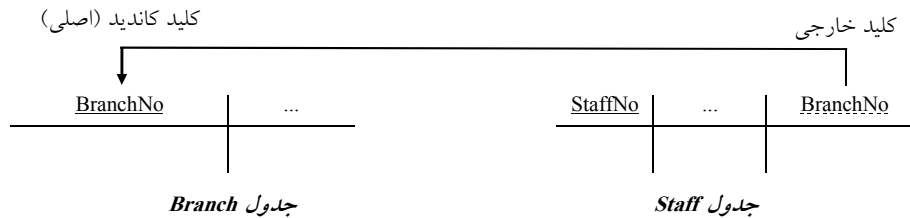
توجه: شاخص (Index) با هدف افزایش سرعت جستجو و بازیابی اطلاعات و به تبع کاهش زمان جستجو و بازیابی اطلاعات ایجاد می‌گردد. و به عنوان نمونه‌ی دیگری از فراداده‌ها، مشخصات سیستمی شاخص در کاتالوگ سیستم نگهداری می‌شود.

دقت کنید که Index بخشی از وجه داده است. در واقع بخش داده از بخش‌های مختلف Table، View و Index تشکیل شده است.

همانطور که گفتیم استفاده از View می‌تواند استقلال داده‌ای را افزایش دهد. اما استفاده از Index به دلیل نامرتب بودن آن با مفهوم استقلال داده‌ای نمی‌تواند استقلال داده‌ای را افزایش یا کاهش دهد و اثری بر استقلال داده‌ای ندارد.

۲۹- گزینه (۱) صحیح است.

دو جدول Branch و Staff را به صورت زیر در نظر بگیرید:



مطابق دستورات Create Table برای دو جدول Branch و Staff ستون BranchNo توسط دستور Primary key(BranchNo) به عنوان کلید اصلی برای جدول Branch انتخاب می‌گردد. همچنین ستون StaffNo توسط دستور Primary key(StaffNo) به عنوان کلید اصلی برای جدول Staff انتخاب می‌گردد. همچنین ستون BranchNo توسط دستور Foreign key(BranchNo) References Branch (BranchNo) به عنوان کلید خارجی برای جدول Staff انتخاب می‌گردد. به شکل زیر:

Foreign key (BranchNo) References Branch (BranchNo)

On delete CASCADE

On update CASCADE

به عنوان کلید خارجی برای جدول Staff انتخاب می‌گردد. ستون BranchNo از جدول Staff به عنوان کلید خارجی به ستون BranchNo از جدول Branch مطابق ساختار تعریف جدول Staff، به شکل On Update Cascade و On Delete Cascade ارجاع می‌کند.

برای رفتار ستون کلید خارجی در یک جدول مقصد، در قبال تغییرات کلید کاندید از یک جدول مبدا گزینه‌های زیر وجود دارد:

Create Table نام جدول

(
 نام ستون‌ها
 :
 foreign key ... references ...
 on delete option
 on update option
 :
)

فیلد option می‌تواند یکی از موارد زیر باشد:

(restrict) no action

گزینه پیش فرض است و هیچ عملی انجام نمی‌شود. یعنی اگر بر اثر عملیات حذف، درج و بروزرسانی، قوانین جامعیت نقض گردند، این اعمال انجام نمی‌گردند. در واقع زمانی عملیات حذف، درج و بروزرسانی انجام می‌گردند که قوانین جامعیت نقض نگردند.

Cascade

اگر سطرهای جداول مرجع حذف یا بروزرسانی شود، کلید خارجی جدولی که به آن ارجاع کرده است نیز حذف یا بروزرسانی خواهد شد.

Set NULL

همان روش NULLIFY است که اگر سطرهای جدول مرجع حذف یا بروزرسانی شود، کلید خارجی

جدولی که به آن ارجاع کرده است با مقدار NULL پُر می‌شود.
توجه: کلید خارجی برای ارتباط میان جداول مورد استفاده قرار می‌گیرد.
 به ازای هر مقدار موجود در یک کلید خارجی، باید دقیقاً یک مقدار متناظر در کلید کاندید متناظر آن وجود داشته باشد، در غیر این صورت می‌گوییم، کلید خارجی دارای ارجاع NULL است. به بیان دیگر، مقادیر کلید خارجی همواره باید زیرمجموعه مقادیر کلید کاندید باشد.
 یک کلید خارجی در یک رابطه هیچگاه نباید ارجاع NULL داشته باشد، این مسأله را به عنوان یک قانون جامعیتی داخلی در مدل رابطه‌ای، می‌شناسیم و آن را **قانون جامعیت ارجاعی** می‌نامیم.
 هر مقداری که در کلید خارجی وجود دارد، باید دارای مقدار متناظر در کلید کاندید مقصد باشد ولی عکس آن صادق نیست.
 دستور زیر بر روی جدول Staff تعریف شده است:

Foreign key (BranchNo) References Branch (BranchNo)

On delete CASCADE

یعنی کلید خارجی جدول Staff یعنی ستون BranchNo به تغییرات (حذف) کلید کاندید جدول Branch یعنی ستون BranchNo به فرم cascade حساس باشد و واکنش نشان دهد، دقت کنید که درج در جدول Branch به خودش ربط دارد و نیاز به واکنش جدول دیگری نیست، اما حذف در جدول Branch به واسطه‌ی تعریف کلید خارجی در جدول Staff باعث می‌شود در جهت حفظ قانون جامعیت ارجاعی از جدول Staff به Branch، جدول Staff همواره به فرم cascade به تغییرات (حذف) در جدول Branch حساس باشد. اما این حساسیت در جدول Staff از جنس cascade است، یعنی اگر رکوردی در جدول Branch حذف شود که منجر به حذف در جدول Staff شود، آن حذف در جدول Staff نیز در جهت حفظ قانون جامعیت ارجاعی پذیرفته می‌شود.
 همچنین دستور زیر بر روی جدول Staff تعریف شده است:

Foreign key (BranchNo) References Branch (BranchNo)

On update CASCADE

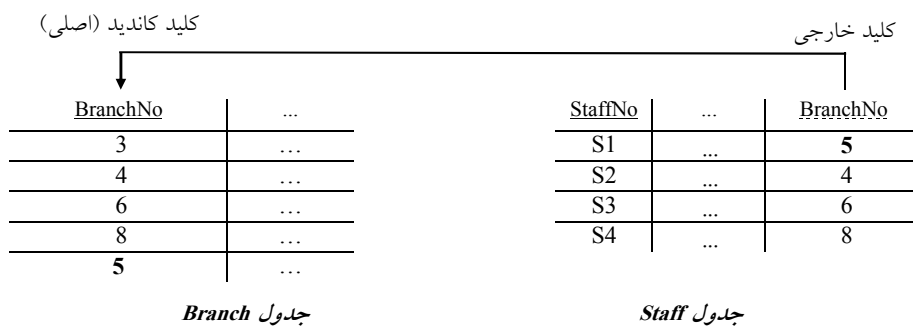
یعنی کلید خارجی جدول Staff یعنی ستون BranchNo به تغییرات (بروزرسانی) کلید کاندید جدول Branch یعنی ستون BranchNo به فرم cascade حساس باشد و واکنش نشان دهد، دقت کنید که درج در جدول Branch به خودش ربط دارد و نیاز به واکنش جدول دیگری نیست، اما بروزرسانی در جدول Branch به واسطه‌ی تعریف کلید خارجی در جدول Staff باعث می‌شود در جهت حفظ قانون جامعیت ارجاعی از جدول Staff به Branch، جدول Staff همواره به فرم cascade به تغییرات (بروزرسانی) در جدول Branch حساس باشد. اما این حساسیت در جدول Staff از جنس cascade است، یعنی اگر رکوردی در جدول Branch بروزرسانی شود که منجر به بروزرسانی در جدول Staff شود، آن بروزرسانی در جدول Staff نیز در جهت حفظ قانون جامعیت ارجاعی پذیرفته می‌شود.

اگر دستورات مطرح شده در صورت سوال به ترتیب زیر اجرا شوند، آنگاه کارمندان فعلی شعب 3 و 4 به شعبه 5 منتقل می‌گردند:

همانطور که گفتیم به ازای هر مقدار موجود در یک کلید خارجی، باید دقیقاً یک مقدار متناظر در کلید کاندید متناظر آن وجود داشته باشد، در غیر این صورت می‌گوییم، کلید خارجی دارای ارجاع NULL است. به بیان دیگر، مقادیر کلید خارجی همواره باید زیرمجموعه مقادیر کلید کاندید باشد.

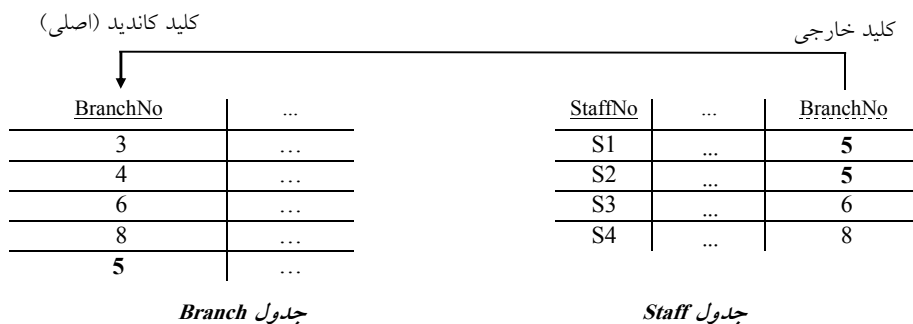
فرم جداول و تعداد رکوردهای جداول Branch و Staff پس از انجام دستور Update به صورت زیر است:

نام دستور	دستور SQL
b	Update Staff set BranchNo=5 where BranchNo=3



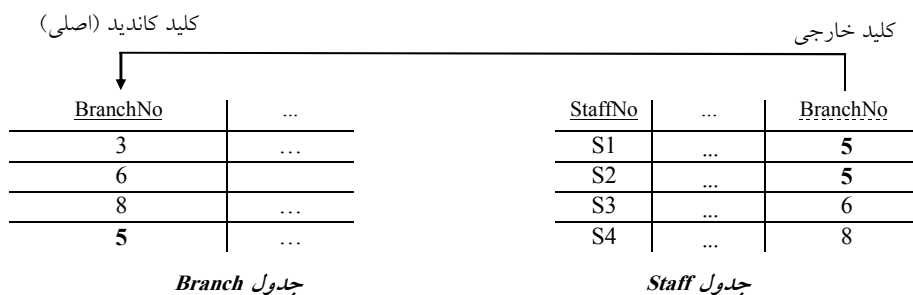
فرم جداول و تعداد رکوردهای جداول Branch و Staff پس از انجام دستور Update به صورت زیر است:

نام دستور	دستور SQL
d	Update Staff set BranchNo=5 where BranchNo=4



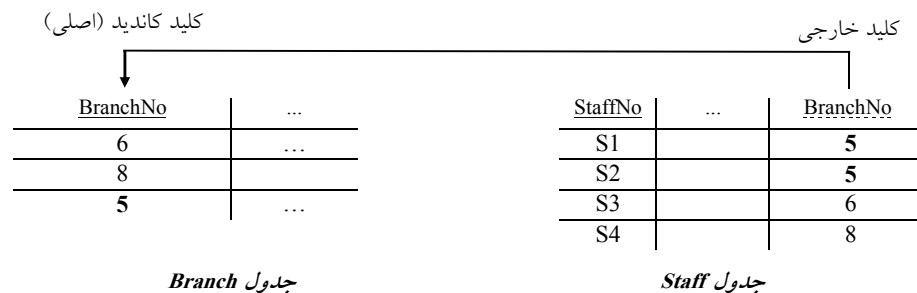
فرم جداول و تعداد رکوردهای جداول Branch و Staff پس از انجام دستور Delete به صورت زیر است:

نام دستور	دستور SQL
c	Delete From Branch where BranchNo=4



فرم جداول و تعداد رکوردهای جداول Branch و Staff پس از انجام دستور Delete به صورت زیر است:

نام دستور	دستور SQL
e	Delete From Branch where BranchNo=3



توجه: علاوه بر ترتیب abdce که در گزینه اول آمده است و پاسخ سوال نیز می‌باشد سایر ترتیب‌های abedc و adcbe و abdec و adbce و adbce نیز درست و امکان پذیر است. بنابراین گزینه‌های دوم، سوم و چهارم را به طور کامل کنار می‌گذاریم، پس پرواضح است که گزینه اول پاسخ سوال است.

۳۰- گزینه (۳) صحیح است.

صورت سوال به این شکل است:

دستور ON UPDATE CASCADE باعث می‌شود در مورد کدام گزینه اطمینان حاصل کنیم؟

(۱) دیدهای ذخیره شده

گزاره اول نادرست است، زیرا دستور ON UPDATE CASCADE ارتباطی به اطمینان حاصل کردن از خروجی داده‌های دید ندارد، ضمن اینکه خروجی دیدها جایی ذخیره نمی‌شود بلکه در یک لحظه به واسطه فراخوانی و شکل ساختاری دید، داده‌های جدول پایه را در خروجی نمایش می‌دهد.
مثال: پیاده‌سازی جدول S به صورت زیر را در نظر بگیرید:

```
CREATE TABLE S
(
  S# char (5),
  Sname char (20)
  Primary key (S#)
)
```

جدول S با مقادیر زیر را در نظر بگیرید:

S#	Sname
S1	Sn1
S2	Sn2

جدول S

پیاده‌سازی View با نام SV را روی جدول S به صورت زیر در نظر بگیرید:

```
CREATE VIEW SV
AS SELECT S.S# AS SID, S.Sname AS SN
FROM S
```

پرس و جوی زیر بر روی VIEW با نام VS را به صورت زیر در نظر بگیرید:

```
SELECT *
FROM SV
```

خروجی پرس و جو بر روی VIEW با نام VS به صورت زیر است:

SID	SN
S1	Sn1
S2	Sn2

۲) نرمال بودن داده‌ها

گزاره دوم نادرست است، زیرا دستور ON UPDATE CASCADE ارتباطی به اطمینان حاصل کردن از نرمال بودن داده‌ها ندارد، نرمال بودن داده‌ها و حذف افزونگی طبیعی داده‌ها بر عهده فرآیند کاهش و نرمال‌سازی است.

۳) یکپارچگی داده‌ها

گزاره سوم درست است، زیرا دستور ON UPDATE CASCADE باعث اطمینان حاصل کردن از یکپارچگی داده‌ها می‌شود. برای رفتار ستون کلید خارجی در یک جدول مقصد، در قبال تغییرات کلید کاندید از یک جدول مبدأ گزینه‌های زیر وجود دارد:

نام جدول Create Table

```
(
نام ستون‌ها
:
foreign key...references...
on delete option
on update option
:
)
```

توجه: فیلد option می‌تواند یکی از موارد (restrict) no action، cascade و set NULL باشد.

توجه: در cascade اگر سطرهای جدول مرجع حذف یا بروزرسانی شود، کلید خارجی جدولی که به آن ارجاع کرده است نیز حذف یا بروزرسانی خواهد شد.

توجه: به تفاوت یکپارچگی یا تجزیه‌ناپذیری (Atomicity) تراکنش‌ها و «یکپارچگی داده‌ها» دقت نمایید.

۴) تمام موارد بالا

گزاره چهارم نادرست است، زیرا گزینه سوم پاسخ سوال است.

توجه: سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه سوم را به عنوان پاسخ اعلام کرده بود.

تست‌های فصل هشتم: وابستگی تابعی

۱- کدام گزینه، معادل مجموعه وابستگی‌های تابعی مقابل است؟ (مهندسی کامپیوتر- دولتی ۸۳)

$$\{(X, Y, Z) \rightarrow ABCD, A \rightarrow B, X \rightarrow Y\}$$

(۱) $\{(X, Y) \rightarrow ACD, A \rightarrow B, X \rightarrow Y\}$

(۲) $\{(X, Y, Z) \rightarrow ABCD, XA \rightarrow BY\}$

(۳) $\{(X, Z) \rightarrow ACD, A \rightarrow B, X \rightarrow Y\}$

(۴) $\{(Y, Z) \rightarrow ACD, A \rightarrow B, X \rightarrow Y\}$

۲- جدولی با پنج ستون A, B, C, D, E دارای وابستگی‌های زیر است. کدام گزینه، کلید کاندید این جدول است؟ (مهندسی IT - دولتی ۸۳)

$$\{A \rightarrow BC, B \rightarrow D, D \rightarrow A\}$$

(۱) (D, E) (۲) (A, E) (۳) (A, D) (۴) (A, D, E)

۳- در بانک زیر کلید کاندید کدام است؟ (مهندسی کامپیوتر- دولتی ۸۴)

$$R = (A, B, C, D, E, H)$$

$$F = \{A \rightarrow B, B \rightarrow C, D \rightarrow E, H \rightarrow A\}$$

(۱) EH (۲) HD (۳) ABD (۴) ADH

۴- رابطه $R(A, B, C, D, E, F, G, H)$ را با وابستگی‌های کاربردی زیر در نظر بگیرید:

کدام یک از وابستگی‌های کاربردی زیر می‌تواند حذف شود، بدون آن‌که کلید اصلی R تغییری بکند؟

(مهندسی IT - دولتی ۸۴)

$$A \rightarrow BCD$$

(۱) $AD \rightarrow E$

$$AD \rightarrow E$$

(۲) $F \rightarrow GH$

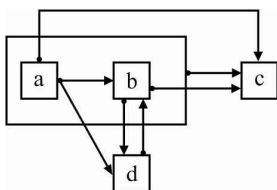
$$EFG \rightarrow H$$

(۳) $EFG \rightarrow H$

$$F \rightarrow GH$$

(۴) $A \rightarrow BCD$

۵- نمودار FD زیر را در نظر بگیرید. کدام FD متعلق به مجموعه حداقل FD ها است؟ (مهندسی IT - دولتی ۸۵)



(۱) $b \rightarrow c$

(۲) $a \rightarrow c$

(۳) $a \rightarrow d$

(۴) $ab \rightarrow c$

۶- حداقل مجموعه وابستگی‌های تابعی S' که معادل مجموعه وابستگی‌های تابعی S می‌باشد، کدام است؟

(مهندسی IT - دولتی ۸۶)

$$S = \{A \rightarrow B, A \rightarrow C, A \rightarrow E, B \rightarrow C, B \rightarrow E, C \rightarrow E\}$$

(۱) $S' = \{A \rightarrow B, A \rightarrow C, C \rightarrow E, B \rightarrow E\}$

$$S' = \{A \rightarrow B, B \rightarrow C, C \rightarrow E\} \quad (۲)$$

$$S' = \{A \rightarrow B, A \rightarrow E, B \rightarrow E\} \quad (۳)$$

$$S' = \{A \rightarrow B, A \rightarrow C, A \rightarrow E, C \rightarrow E\} \quad (۴)$$

۷- رابطه $R(A, B, C, D, E, F)$ با مجموعه وابستگی های FD مفروض است. کلید کاندید در رابطه R کدام است؟ (مهندسی IT - آزاد ۸۵)

$$FD = \{AB \rightarrow C, DB \rightarrow E, C \rightarrow F, B \rightarrow CE, A \rightarrow BE, D \rightarrow C\}$$

$$(A, B, D) \quad (۴) \quad (A, D) \quad (۳) \quad (B, D) \quad (۲) \quad (A, B) \quad (۱)$$

۸- مجموعه غیر قابل کاهش وابستگی ها برای مجموعه وابستگی FD شامل چند وابستگی است؟ (مهندسی IT - آزاد ۸۵)

$$FD = \{AC \rightarrow BD, B \rightarrow DE, A \rightarrow D, A \rightarrow C, BC \rightarrow E, D \rightarrow E, A \rightarrow E\}$$

$$6 \quad (۴) \quad 5 \quad (۳) \quad 4 \quad (۲) \quad 3 \quad (۱)$$

۹- اگر مجموعه وابستگی زیر در رابطه $R(A, B, C, D, E, F, G)$ مفروض باشند، آن گاه کدام وابستگی زیر را می توان نتیجه گرفت: (مهندسی IT - آزاد ۸۶)

$$\{A \rightarrow B, CD \rightarrow F, B \rightarrow D, B \rightarrow C, BE \rightarrow G, G \rightarrow F, AG \rightarrow B\}$$

$$A \rightarrow F \quad (۴) \quad G \rightarrow D \quad (۳) \quad G \rightarrow B \quad (۲) \quad B \rightarrow G \quad (۱)$$

۱۰- در رابطه با $R(x, y, z, w)$ اگر $x \rightarrow z$ و $y \rightarrow w$ برقرار باشد، نامزد کلیدی رابطه R کدام است؟ (مهندسی کامپیوتر - دولتی ۸۷)

$$(۱) \text{ نامزد کلیدی } (x, y) \text{ است.} \quad (۲) \text{ نامزد کلیدی فقط } x \text{ یا فقط } y \text{ است.}$$

$$(۳) \text{ نامزد کلیدی } (x, y, z, w) \text{ است.} \quad (۴) \text{ } x \text{ یا } y \text{ هر یک نامزد کلیدی است.}$$

۱۱- فرض کنید AB کلید اصلی شمای $R(A, B, C, D, E, H, K)$ باشد، کدام یک از وابستگی های تابعی زیر وابستگی جزئی است؟ (مهندسی کامپیوتر - آزاد ۸۷)

$$DEH \rightarrow K \quad (۴) \quad D \rightarrow EH \quad (۳) \quad A \rightarrow CD \quad (۲) \quad AB \rightarrow E \quad (۱)$$

۱۲- اگر A یک مجموعه از صفات در شمای R باشد و داشته باشیم $A \xrightarrow{FFD} R$ (R وابستگی تابعی کامل به A دارد) در این صورت A شمای R است. (مهندسی IT - آزاد ۸۷)

$$(۱) \text{ کلید کاندید} \quad (۲) \text{ کلید اصلی} \quad (۳) \text{ کلید فرعی} \quad (۴) \text{ کلید خارجی}$$

۱۳- رابطه $R(A, B, C, D, E, H, K, L)$ با وابستگی های تابعی زیر داده شده است. کلید کاندید R کدام است؟ (مهندسی IT - آزاد ۸۷)

$$F = \{AB \rightarrow L, C \rightarrow H, CD \rightarrow E, EH \rightarrow B, K \rightarrow C\}$$

$$ADK \quad (۴) \quad ABDHK \quad (۳) \quad K \quad (۲) \quad ABK \quad (۱)$$

۱۴- اگر در جدول ST داشته باشیم $Code \rightarrow T\#$ ، آن گاه این جدول چند نامزد کلیدی (Candidate Key) دارد؟ (مهندسی کامپیوتر - دولتی ۸۸)

ST(S#, T#, Date, Time, Code)

(۱) یک (۲) دو (۳) سه (۴) چهار

۱۵- کدام مورد، درباره وابستگی‌های تابعی (Functional Dependencies) غلط می‌باشد؟ (مهندسی IT- دولتی ۸۸)

(۱) $AB \rightarrow C, D \rightarrow AE, AF \rightarrow C, EF \rightarrow GA \Rightarrow AD \rightarrow C$ (۲) $AB \rightarrow C, D \rightarrow AE, AE \rightarrow C, EF \rightarrow GA \Rightarrow DF \rightarrow G$ (۳) $X \rightarrow Y \wedge YZ \rightarrow W \Rightarrow XZ \rightarrow W$ (۴) $X \rightarrow Y \wedge Y \rightarrow Z \wedge X \rightarrow W \Rightarrow XZ \rightarrow W$

۱۶- رابطه R با وابستگی‌های تابعی F داده شده است. کلید اصلی R کدام است؟ (مهندسی IT- آزاد ۸۸)

 $R(A, B, C, D, E, F, G, H)$ $F = \{ABC \rightarrow DE, G \rightarrow DE, E \rightarrow F, H \rightarrow E, F \rightarrow G\}$

(۱) ABH (۲) ABCH (۳) ABE (۴) ABC

۱۷- فرض کنید AB کلید اصلی شمای $R(A, B, C, D, E, H)$ باشد. کدام یک از وابستگی‌های تابعی زیر،

وابستگی انتقالی است؟ (مهندسی IT- آزاد ۸۷)

(۱) $AB \rightarrow C$ (۲) $A \rightarrow DEH$ (۳) $D \rightarrow EH$ (۴) $B \rightarrow D$

۱۸- فرض کنید $R = (U)$ عنوان یک رابطه باشد که روی مجموعه صفات U تعریف شده است و وابستگی‌های تابعی

موجود در مجموعه F در مورد آن اعمال شده است. با فرض این که R دارای یک کلید منحصر به فرد K است.

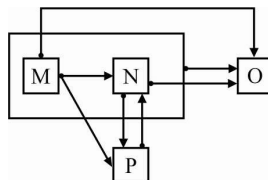
کدام یک از این نتیجه‌ها در مورد گزاره زیر صادق است؟ (مهندسی IT- دولتی ۸۸)

 $\forall X \subseteq U \forall A \in U : X \rightarrow A \in F^+ \wedge A \notin X$ (۱) $X \subset K$ (۲) $X \not\subset K$ (۳) A جزئی از کلید اصلی است. ($A \in K$)(۴) A جزئی از کلید اصلی نیست. ($A \notin K$)۱۹- رابطه $R(A, B, C, D, E)$ و وابستگی‌های تابعی زیر را در نظر بگیرید. $A \rightarrow B$ $AB \rightarrow CD$ $D \rightarrow ABC$

کدام گزینه کلید رابطه است؟ (مهندسی کامپیوتر- دولتی ۸۹)

(۱) AD (۲) AE (۳) AB (۴) ABD

۲۰- نمودار وابستگی‌های تابعی (FD) زیر را در نظر بگیرید:



کدام FD متعلق به مجموعه حداقل FD ها است؟ (مهندسی کامپیوتر- آزاد ۸۹ گروه ب)

(۱) $N \rightarrow O$ (۲) $(MN) \rightarrow O$ (۳) $M \rightarrow P$ (۴) $M \rightarrow O$

۲۱- رابطه $R(A, B, C, D, E)$ با وابستگی‌های تابعی زیر را در نظر بگیرید:

$A \rightarrow B, B \rightarrow C, C \rightarrow A, D \rightarrow E, E \rightarrow D$

تعداد ابرکلید (فوق کلید) ها و کلیدهای کاندید رابطه (به ترتیب از راست به چپ) چندتا است؟

(مهندسی کامپیوتر- آزاد ۸۹ گروه ب)

(۱) ابرکلید= 8 و کلید کاندید= 2 (۲) ابرکلید= 21 و کلید کاندید= 6

(۳) ابرکلید= 21 و کلید کاندید= 8 (۴) ابرکلید= 18 و کلید کاندید= 7

۲۲- رابطه $R(A, B, C, D, E)$ و وابستگی‌های تابعی (Functional Dependency) زیر را در نظر بگیرید. فرض

کنید نمی‌دانیم \square چیست و \square می‌تواند هر زیر مجموعه غیرتهی از صفات R باشد.

$A \rightarrow BC$

$CD \rightarrow E$

$\square \rightarrow D$

(مهندسی کامپیوتر- دولتی ۹۰)

کدام یک از گزاره‌های زیر مستقل از \square صحیح است؟

(I) هر کلید کاندید R شامل A است.

(II) هیچ کلید کاندیدی از R شامل C نیست.

(III) بعضی از کلیدهای کاندید R شامل C هستند و بعضی دیگر شامل C نمی‌باشند.

(۱) I و II (۲) فقط II (۳) فقط I (۴) I و III

۲۳- فرض کنید رابطه‌ای به صورت $R(V)$ که V مجموعه‌ای از صفات است و با وابستگی‌های تابعی

$F = \{x_1 \rightarrow y_1, x_2 \rightarrow y_2, \dots, x_n \rightarrow y_n\}$ داده شده باشد که در آن x_i و y_i مجموعه‌ای از صفات که

زیر مجموعه V هستند، می‌باشند. کدام یک از گزینه‌های زیر صحیح است؟ (U اجتماع مجموعه‌ای است).

(مهندسی IT- دولتی ۹۰)

(۱) اگر $\bigcup_{i=1}^n (x_i - y_i)$ یک ابرکلید باشد، آن‌گاه R تنها یک کلید کاندید خواهد داشت.

(۲) اگر $V - \bigcup_{i=1}^n (x_i - y_i)$ یک ابرکلید باشد، آن‌گاه R تنها یک کلید کاندید خواهد داشت.

(۳) اگر $V - \bigcup_{i=1}^n (y_i - x_i)$ یک ابرکلید باشد، آن‌گاه R حداقل یک کلید کاندید خواهد داشت.

(۴) اگر $V - \bigcup_{i=1}^n (y_i - x_i)$ یک ابرکلید باشد، آن‌گاه R تنها یک کلید کاندید خواهد داشت.

۲۴- رابطه R با وابستگی‌های تابعی F به صورت زیر مفروض است. کلید اصلی R کدام است؟

(مهندسی کامپیوتر- آزاد ۹۰ گروه الف)

$R(A, B, C, D, E, F, G, H, I, J)$

$F\{ABD \rightarrow EG, C \rightarrow DG, E \rightarrow FG, I \rightarrow H, H \rightarrow J, AB \rightarrow C, G \rightarrow F\}$

(۱) ABI (۲) AB (۳) ABD (۴) ABDI

۲۵- رابطه $R(A, B, C, D, E)$ با وابستگی‌های تابعی زیر را در نظر بگیرید: (مهندسی کامپیوتر- آزاد ۹۰ گروه ب)

$$ABC \rightarrow DE$$

$$D \rightarrow AB$$

تعداد ابرکلیدهای (Superkey) رابطه چند تا است؟

2 (۱) 7 (۲) 10 (۳) 12 (۴)

۲۶- رابطه $R(A, B, C, D, E)$ و وابستگی‌های تابعی زیر را در نظر بگیرید:

$$A \rightarrow B$$

$$BC \rightarrow D$$

$$E \rightarrow C$$

کدام یک از وابستگی‌های تابعی زیر لزوماً در R برقرار نیست؟ (مهندسی IT- دولتی ۹۱)

CE \rightarrow D (۱) AE \rightarrow C (۲) BC \rightarrow B (۳) AC \rightarrow D (۴)

۲۷- فرض کنید سه رابطه $R_1(A, B)$ ، $R_2(B, C)$ و $R_3(A, C)$ را داریم. اگر FD های $A \rightarrow B$ ، $B \rightarrow C$ و

$A \rightarrow C$ را داشته باشیم کدام وابستگی (FD) می‌تواند افزونه باشد؟ (مهندسی کامپیوتر- آزاد ۹۱)

A \rightarrow B (۱) A \rightarrow C (۲)

B \rightarrow C (۳) (۴) رابطه‌ها جدا هستند و رابطه افزونه وجود ندارد.

۲۸- رابطه $R(A, B, C, D, E)$ و وابستگی‌های تابعی (Functional Dependencies) زیر را در نظر بگیرید. کدام

یک از این وابستگی‌های تابعی حتماً در R برقرار است؟ (مهندسی کامپیوتر- دولتی ۹۲)

A \rightarrow E (۱) B \rightarrow A (۲) AD \rightarrow E

D \rightarrow E (۳) C \rightarrow E (۴) C \rightarrow AB

$$B \rightarrow D$$

۲۹- در یک رابطه شش وابستگی تابعی به صورت ذیل داریم:

1) A \rightarrow (B, C) 2) A \rightarrow D 3) A \rightarrow K
4) K \rightarrow C 5) B \rightarrow D 6) (B, C) \rightarrow D

پس از کمینه کردن مجموعه وابستگی‌ها، این رابطه چند وابستگی تابعی دارد؟ (مهندسی کامپیوتر- آزاد ۹۲)

4 (۱) 5 (۲) 3 (۳) (۴) هیچ کدام

۳۰- رابطه $R(A, B, C, D)$ را در نظر بگیرید. کدام یک از گزینه‌های زیر صحیح نیست؟

(مهندسی کامپیوتر- دولتی ۹۳)

(۱) اگر $A \rightarrow B$ و $AB \rightarrow C$ آن‌گاه $A \rightarrow C$ (۲) اگر $A \rightarrow B$ و $BC \rightarrow D$ آن‌گاه $AC \rightarrow D$

(۳) اگر $A \rightarrow B$ و $AC \rightarrow D$ آن‌گاه $AC \rightarrow BD$ (۴) اگر $AB \rightarrow BC$ و $C \rightarrow D$ آن‌گاه $A \rightarrow D$

۳۱- با داشتن رابطه $r(A, B, C, D, E)$ و مجموعه وابستگی‌های F به صورت زیر، کدام عبارت صحیح است؟

(مهندسی IT- دولتی ۹۳)

$$F = \{AB \rightarrow C, C \rightarrow A, C \rightarrow BD, D \rightarrow E\}$$

(۱) A و AC کلید هستند.

(۲) AB و AC کلید هستند.

(۳) AC و AD کلید هستند.

(۴) AD و CD کلید هستند.

۳۲- رابطه $R(A,B,C,D,E)$ و مجموعه وابستگی‌های تابعی F را در نظر بگیرید.

$$F = \{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$$

کدام یک از گزینه‌ها نمی‌تواند یک کلید کاندید برای R باشد؟ (مهندسی کامپیوتر-دولتی ۹۵)

B (۱) E (۲) BC (۳) CD (۴)

۳۳- رابطه $R(A,B,C,D,E)$ و مجموعه وابستگی‌های تابعی زیر را در نظر بگیرید. کلید کاندید این رابطه

کدام است؟ $FD = \{(A,B) \rightarrow C, B \rightarrow D, (D,E) \rightarrow C\}$ (مهندسی کامپیوتر-دولتی ۹۶)

{(A,B,D)} (۱) {(A,B,E)} (۲)

{(A,B,C,E)} (۳) {(A,B,D,E)} (۴)

۳۴- در رابطه R با مجموعه صفات $\{A,B,C,D\}$ ، اگر وابستگی‌های تابعی (FD) به صورت

$F = \{(A,B) \rightarrow (C,D), D \rightarrow B\}$ باشد، کلید(های) کاندید این رابطه کدام است؟ (مهندسی IT-دولتی ۹۶)

D فقط (۱) فقط (A,B) (۲) فقط (A,D) (۳) (A,D) و (A,B) (۴)

۳۵- با توجه به رابطه $R(A,B,C,D,E,F)$ و مجموعه وابستگی‌های تابعی زیر، رابطه R چند کلید کاندید

دارد؟ $F = \{A \rightarrow BCD, BC \rightarrow DE, B \rightarrow D, D \rightarrow A\}$ (مهندسی کامپیوتر-دولتی ۹۷)

1 (۴) 2 (۳) 3 (۲) 4 (۱)

۳۶- با توجه به رابطه $R(A,B,C,D,E)$ و مجموعه وابستگی‌های تابعی زیر، کدام مورد نادرست است؟

$F = \{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$ (مهندسی IT-دولتی ۹۷)

{E} و {C,B} هر دو کلید کاندید هستند. (۱) {B} و {C,D} هر دو کلید کاندید هستند.

{E} و {C,D} هر دو کلید کاندید هستند. (۳) {A} و {E} هر دو کلید کاندید هستند. (۴)

۳۷- رابطه $R(A,B,C,D,E)$ و وابستگی‌های تابعی زیر را در نظر بگیرید:

$$f = \{A \rightarrow B, AB \rightarrow CD, D \rightarrow ABC\}$$

کلید کاندید رابطه کدام است؟ (مهندسی کامپیوتر-دولتی ۹۸)

AB (۱) AD (۲) AE (۳) BE (۴)

۳۸- با در نظر گرفتن رابطه $R(A,B,C,D,E)$ و مجموعه وابستگی‌های تابعی

$F = \{AB \rightarrow C, CD \rightarrow E, DE \rightarrow B\}$ ، این رابطه چند ابرکلید دارد؟ (مهندسی IT-دولتی ۹۸)

3 (۱) 7 (۲) 10 (۳) 12 (۴)

۳۹- با در نظر گرفتن رابطه $R(A,B,C,D,E)$ و مجموعه وابستگی‌های تابعی $F = \{AB \rightarrow C, CD \rightarrow E\}$ ،

این رابطه چند ابرکلید دارد؟ (مهندسی کامپیوتر-دولتی ۹۹)

1 (۱) 3 (۲) 4 (۳) 7 (۴)

۴۰- رابطه $R(A,B,C,D,E,F)$ را در نظر بگیرید. فرض کنیم A و (B,C) دو کلید کاندید این رابطه و تنها

کلیدهای کاندید باشند. این رابطه چند ابرکلید دارد؟ (مهندسی کامپیوتر-دولتی ۹۹)

8 (۱) 24 (۲) 32 (۳) 40 (۴)

۴۱- در مورد رابطه $R(A, B, C, D, E)$ مجموعه وابستگی‌های تابعی زیر به دست آمده است:

$$F = \{A \rightarrow B, A \rightarrow C, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$$

با توجه به این مجموعه وابستگی‌های تابعی، کدام وابستگی تابعی زیر را نمی‌توان استنتاج کرد؟

(مهندسی IT- دولتی ۹۹)

$$AC \rightarrow BC \quad (۴) \quad BC \rightarrow CD \quad (۳) \quad CD \rightarrow AC \quad (۲) \quad BD \rightarrow CD \quad (۱)$$

۴۲- رابطه $R(A, B, C, D, E)$ و این وابستگی‌های تابعی را در نظر بگیرید، تعداد کلیدهای کاندید R چندتاست؟

$$AB \rightarrow CDE, E \rightarrow BC$$

(مهندسی کامپیوتر-دولتی ۱۴۰۰)

$$4 \quad (۴) \quad 3 \quad (۳) \quad 2 \quad (۲) \quad 1 \quad (۱)$$

(مهندسی IT- دولتی ۱۱۴۰۰)

۴۳- در رابطه $R = (A, B, C, D, E, F)$ وابستگی‌های زیر برقرار است:

$$A \rightarrow F$$

$$B \rightarrow C$$

$$ABC \rightarrow E$$

در این رابطه چند سوپر کلید وجود دارد؟

$$8 \quad (۴) \quad 6 \quad (۳) \quad 3 \quad (۲) \quad 1 \quad (۱)$$

۴۴- فرض کنید در یک جدول $N \geq 3$ خصیصه داریم. در این جدول دو کلید کاندید وجود دارد. یک کلید

کاندید دارای دو خصیصه است و یک کلید کاندید دیگر دارای یک خصیصه. بین خصیصه‌های این دو

کلید همپوشانی وجود ندارد. تعداد کل سوپر کلیدها در این جدول کدام است؟ (مهندسی IT- دولتی ۱۴۰۰)

$$5 \times 2^{N-2} \quad (۴) \quad 5 \times 2^{N-3} \quad (۳) \quad 3 \times 2^{N-3} \quad (۲) \quad 3 \times 2^{N-2} \quad (۱)$$

پاسخ تست‌های فصل هشتم: وابستگی تابعی

۱- گزینه (۳) صحیح است.

به طور کلی دو مجموعه وابستگی تابعی F_1 و F_2 معادل هستند اگر و فقط اگر $F_1^+ = F_2^+$. ابتدا باید مجموعه وابستگی‌های مطرح شده مورد بررسی قرار گیرد.

با کمی دقت می‌توان موارد زائد را حذف کرد، از آن‌جا که $X \rightarrow Y$ ، پس می‌توان Y را در سمت چپ از کنار X حذف کرد و به جای $(X, Y, Z) \rightarrow ABCD$ نوشت $(X, Z) \rightarrow ABCD$

$$\left. \begin{array}{l} XYZ \rightarrow ABCD \\ X \rightarrow Y \end{array} \right\} XZ \rightarrow ABCD$$

هم‌چنین چون $A \rightarrow B$ پس می‌توان B را در سمت راست وابستگی از کنار A حذف کرد و به جای $(X, Z) \rightarrow ACD$ نوشت $(X, Z) \rightarrow ABCD$.

$$\left. \begin{array}{l} XZ \rightarrow ABCD \\ A \rightarrow B \end{array} \right\} XZ \rightarrow ACD$$

بنابراین داریم:

$$\{(X, Z) \rightarrow ACD, A \rightarrow B, X \rightarrow Y\}$$

پرواضح است که مجموعه فوق، حاصل از حذف موارد زائد در مجموعه وابستگی‌های مطرح شده دقیقاً مطابق گزینه سوم است.

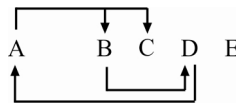
سایر گزینه‌ها شرایط لازم برای معادل بودن با مجموعه داده شده را دارا نیستند.

۲- گزینه‌های (۱ و ۲) صحیح است.

وابستگی‌های زیر را برای جدول $R(A, B, C, D, E)$ داریم:

$$\begin{array}{l} A \rightarrow BC \\ B \rightarrow D \\ D \rightarrow A \end{array}$$

دیاگرام وابستگی‌های فوق به صورت زیر است:



گزینه سوم نادرست است. زیرا، ستون E در سمت راست هیچ یک از وابستگی‌های تابعی نیامده است، بنابراین ستون E حتماً عضو کلید کاندید باید باشد.

با کمی بررسی در نمودار وابستگی‌ها، پرواضح است که ستون A ، ستون‌های B و C و به واسطه ستون B و رابطه تعدی، ستون A ، ستون D را هم می‌دهد. بنابراین ستون A ، صفت‌های B ، C و D را می‌دهد.

پس از آن‌جا که A ، ستون‌های B ، C و D را می‌دهد و ستون E هم حتماً باید عضو کلید کاندید باشد، پس AE کلید کاندید این جدول تلقی می‌شود. از طرف دیگر چون وابستگی $D \rightarrow A$ برقرار است، و به تبع ستون A هم ستون‌های B و C را می‌دهد، می‌توان نتیجه گرفت که DE هم، کلید کاندید دیگری برای این جدول است.

از طرفی دیگر هم، چون وابستگی $B \rightarrow D$ برقرار است و وابستگی $D \rightarrow A$ هم برقرار است و به تبع ستون A هم ستون‌های B و C را می‌دهد، می‌توان نتیجه گرفت که BE کلید کاندید دیگری برای این جدول است. بنابراین جدول $R(A, B, C, D, E)$ ، دارای سه کلید کاندید AE ، DE و BE می‌باشد و کلید کاندید دیگری هم ندارد.

با توجه به وابستگی‌های مطرح شده:

$$A \rightarrow BC$$

$$B \rightarrow D$$

$$D \rightarrow A$$

بستار کلیدهای کاندید AE ، DE و BE به صورت زیر است:

$$\{AE\}^+ = \{A, E, B, C, D\}$$

$$\{DE\}^+ = \{D, E, A, B, C\}$$

$$\{BE\}^+ = \{B, E, D, A, C\}$$

بنابراین گزینه‌های اول و دوم درست هستند.

گزینه چهارم نادرست است. زیرا ADE ابرکلید است، اما کلید کاندید نیست، زیرا ADE ، عضو زائدی مثل A یا D دارد، چون وجود دو ستون A و D به صورت با هم در ساخت یک کلید کاندید با توجه به وابستگی‌های مطرح شده، ضروری نیست و وجود هر یک به تنهایی به همراه ستون E ، برای ساخت کلید کاندید، لازم و کافی است.

۳- گزینه (۲) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

توجه: ما نام این روابط را به عنوان مبدع آن «قوانین ارسطو» نام‌گذاری کردیم، این قوانین به «قوانین چهارگانه ارسطو» نیز موسوم است. این قوانین می‌تواند جایگزین مناسبی برای قوانین آرمسترانگ باشد.

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیربدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

توجه: عبارت $[x_i \text{ (چپ)} - y_i \text{ (راست)}]$ به طور مستقل بر روی تک تک وابستگی‌ها انجام می‌گردد.

$$A \rightarrow BC \Rightarrow BC - A = BC$$

مثال:

توجه: استفاده از روش اول مستلزم گام ابتدایی حذف وابستگی‌های بدیهی است، اگر در حذف

وابستگی‌های بدیهی دچار خطا می‌شوید، از روش دوم استفاده نمایید.

توجه: وابستگی بدیهی: اگر در یک وابستگی، بخش سمت راست، زیر مجموعه بخش سمت چپ باشد آن وابستگی را، وابستگی بدیهی می‌نامند.

$$AB \rightarrow A$$

مثال:

با توجه به وابستگی‌های مطرح شده داریم:

$$A \rightarrow B$$

$$B \rightarrow C$$

$$D \rightarrow E$$

$$H \rightarrow A$$

$$ABCDEH - ABCE = DH$$

بنابر رابطه فوق صفات DH حتماً باید عضو کلید کاندید باشد.

بستار صفات DH به صورت زیر است:

$$\{DH\}^+ = \{D, H, A, B, C\}$$

بر اساس بستار فوق، صفات DH، همه ستون‌ها را تولید می‌کند، پس صفات DH کلید کاندید می‌باشد.

قانون دوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم) همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

گزینه اول نادرست است. زیرا، عضو کلید کاندید DH را ندارد.

گزینه سوم نادرست است. زیرا، عضو کلید کاندید DH را ندارد.

گزینه چهارم نادرست است. زیرا، عضو زائد A را دارد، هر چند عضو کلید کاندید DH را دارد. صفات DH همه ستون‌ها را به تنهایی با استفاده از وابستگی‌های مطرح شده تولید می‌کنند، بنابراین صفت A، عضو زائد است. مجموعه صفات ADH در واقع ابرکلید است.

۴- گزینه (۳) صحیح است.

طراح محترم: در مباحث پایگاه داده، مفهومی، تحت عنوان وابستگی کاربردی وجود ندارد، بنابراین، عبارت وابستگی کاربردی به وابستگی تابعی باید اصلاح گردد.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیربدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E, F, G, H)$ داریم:

$$A \rightarrow BCD$$

$$AD \rightarrow E$$

$$EFG \rightarrow H$$

$$F \rightarrow GH$$

$$ABCDEFGH - BCDEGH = AF$$

بنابر رابطه فوق صفات AF حتماً باید عضو کلید کاندید باشد.

بستار صفات AF به صورت زیر است:

$$\{AF\}^+ = \{A, F, B, C, D, E, G, H\}$$

بر اساس بستار فوق، صفات AF ، همه ستون‌ها را تولید می‌کند، پس صفات AF کلید کاندید می‌باشد.

قانون دوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم) همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

مجدداً وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E, F, G, H)$ را در نظر بگیرید:

$$A \rightarrow BCD$$

$$AD \rightarrow E$$

$$EFG \rightarrow H$$

$$F \rightarrow GH$$

حال اگر وابستگی $EFG \rightarrow H$ ، از مجموعه وابستگی‌ها حذف گردد، تأثیری در کلید کاندید رابطه R نخواهد داشت. زیرا صفت H در سمت راست وابستگی $EFG \rightarrow H$ توسط وابستگی $F \rightarrow GH$ تولید می‌گردد. در واقع بود و نبود وابستگی $EFG \rightarrow H$ در یافتن کلید کاندید رابطه R تأثیری نخواهد داشت. بنابراین بدون وابستگی $EFG \rightarrow H$ داریم:

$$A \rightarrow BCD$$

$$AD \rightarrow E$$

$$F \rightarrow GH$$

$$ABCDEFGH - BCDEGH = AF$$

بنابراین با حذف وابستگی $EFG \rightarrow H$ ، صفات AF ، هم چنان عضو کلید کاندید رابطه R خواهد بود. بستار صفات AF ، بر اساس وابستگی‌های فوق و بدون استفاده از وابستگی $EFG \rightarrow H$ ، به صورت زیر است:

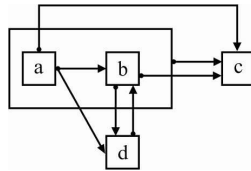
$$\{AF\}^+ = \{A, F, B, C, D, E, G, H\}$$

بر اساس بستار فوق، صفات AF ، همه ستون‌ها را تولید می‌کند. پس صفات AF هم چنان کلید کاندید می‌باشد.

۵- گزینه (۱) صحیح است.

وابستگی‌های تابعی متناظر با نمودار مطرح شده به صورت زیر است:

$a \rightarrow b$
 $a \rightarrow c$
 $b \rightarrow c$
 $a \rightarrow d$
 $b \rightarrow d$
 $d \rightarrow b$
 $ab \rightarrow c$



راه حل اول:

به دست آوردن مجموعه حداقل FDها براساس قوانین زیر:

- ۱- سمت راست همه وابستگی‌ها توسط عمل تجزیه، تک خصیصه‌ای باشد.
- ۲- سمت چپ همه وابستگی‌ها در صورت امکان کمینه باشد.
- ۳- همه وابستگی‌های زائد حذف گردد.

وابستگی $ab \rightarrow c$ به دلیل وجود وابستگی $a \rightarrow b$ قابل تغییر به وابستگی $a \rightarrow c$ است. در ادامه پس از حذف وابستگی‌های زائد داریم:
حذف وابستگی $a \rightarrow c$ به دلیل زیر:

$$\begin{cases} a \rightarrow b \\ b \rightarrow c \end{cases} \Rightarrow a \rightarrow c$$

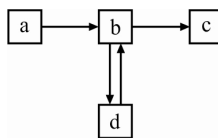
حذف وابستگی $a \rightarrow d$ به دلیل زیر:

$$\begin{cases} a \rightarrow b \\ b \rightarrow d \end{cases} \Rightarrow a \rightarrow d$$

بنابراین مجموعه حداقل FDها به صورت زیر است:

$a \rightarrow b$
 $b \rightarrow c$
 $b \rightarrow d$
 $d \rightarrow b$

هم‌چنین نمودار وابستگی‌های مجموعه حداقل FDها به صورت زیر است:



بنابراین وابستگی $b \rightarrow c$ مطرح شده در گزینه اول، متعلق به مجموعه حداقل FDها است.

راه حل دوم: (حذف گزینه)

اگر وابستگی متعلق به مجموعه حداقل FDها باشد، نباید توسط وابستگی‌های دیگر تولید گردد. گزینه اول درست است. زیرا، وابستگی $b \rightarrow c$ متعلق به مجموعه حداقل FDها است، زیرا توسط وابستگی‌های دیگر قابل تولید نیست. یعنی اگر صفت b را داشته باشیم، نمی‌توان از صفت b بدون استفاده از وابستگی $b \rightarrow c$ و با استفاده از وابستگی‌های دیگر به صفت c رسید. پُر واضح است که نمی‌شود. گزینه دوم نادرست است. زیرا وابستگی $a \rightarrow c$ قابل حذف است، چون اگر وابستگی $a \rightarrow c$ حذف گردد،

از صفت a توسط وابستگی‌های دیگر می‌توان به صفت c رسید، بنابراین وابستگی $a \rightarrow c$ زائد است.

$$\begin{cases} a \rightarrow b \\ b \rightarrow c \end{cases} \Rightarrow a \rightarrow c$$

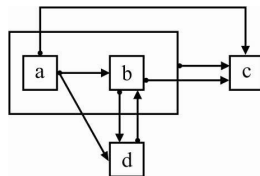
گزینه سوم نادرست است. زیرا، وابستگی $a \rightarrow b$ قابل حذف است، چون اگر وابستگی $a \rightarrow d$ حذف گردد، از صفت a توسط وابستگی‌های دیگر می‌توان به صفت d رسید. بنابراین وابستگی $a \rightarrow d$ زائد است.

$$\begin{cases} a \rightarrow b \\ b \rightarrow d \end{cases} \Rightarrow a \rightarrow d$$

گزینه چهارم نادرست است. زیرا، وابستگی $ab \rightarrow c$ به دلیل وجود وابستگی $a \rightarrow b$ قابل تغییر به وابستگی $a \rightarrow c$ می‌باشد و مطابق آنچه پیش از این گفتیم وابستگی $a \rightarrow c$ زائد است.

راه‌حل سوم: (با استفاده از نمودار)

نمودار مطرح شده را در نظر بگیرید:



گزینه اول درست است. زیرا، وابستگی $b \rightarrow c$ متعلق به مجموعه حداقل FD ها است، زیرا توسط مسیرهای دیگر قابل دسترسی نیست.

یعنی اگر مطابق نمودار فوق، مربع صفت b را داشته باشیم، هیچ مسیر دیگری به مربع صفت c وجود ندارد. گزینه دوم نادرست است. زیرا، وابستگی $a \rightarrow c$ قابل حذف است، چون اگر وابستگی $a \rightarrow c$ حذف گردد، از مربع صفت a توسط مسیر $a \rightarrow b \rightarrow c$ می‌توان به مربع صفت c دسترسی پیدا کرد. بنابراین وابستگی $a \rightarrow c$ زائد است.

گزینه سوم نادرست است. زیرا، وابستگی $a \rightarrow d$ قابل حذف است، چون اگر وابستگی $a \rightarrow d$ حذف گردد، از مربع صفت a توسط مسیر $a \rightarrow b \rightarrow d$ می‌توان به مربع صفت d دسترسی پیدا کرد. بنابراین وابستگی $a \rightarrow d$ زائد است.

گزینه چهارم نادرست است، زیرا، وابستگی $ab \rightarrow c$ قابل حذف است، چون اگر وابستگی $ab \rightarrow c$ حذف گردد، از مربع صفت a توسط مسیر $a \rightarrow b \rightarrow c$ می‌توان به مربع صفت c دسترسی پیدا کرد. بنابراین وابستگی $ab \rightarrow c$ زائد است.

۶- گزینه (۲) صحیح است.

وابستگی‌های مطرح شده به صورت زیر است:

- $A \rightarrow B$
- $A \rightarrow C$
- $A \rightarrow E$
- $B \rightarrow C$
- $B \rightarrow E$
- $C \rightarrow E$

مجموعه حداقل FD ها براساس قوانین زیر ایجاد می‌گردد:

۱- سمت راست همه وابستگی‌ها توسط عمل تجزیه تک خصیصه‌ای باشد.

۲- سمت چپ همه وابستگی‌ها در صورت امکان کمینه باشد.

۳- همه وابستگی‌های زائد حذف گردد.

حذف وابستگی $A \rightarrow C$ به دلیل زیر:

$$\begin{cases} A \rightarrow B \\ B \rightarrow C \end{cases} \Rightarrow A \rightarrow C$$

حذف وابستگی $A \rightarrow E$ به دلیل زیر:

$$\begin{cases} A \rightarrow B \\ B \rightarrow E \end{cases} \Rightarrow A \rightarrow E$$

حذف وابستگی $B \rightarrow E$ به دلیل زیر:

$$\begin{cases} B \rightarrow C \\ C \rightarrow E \end{cases} \Rightarrow B \rightarrow E$$

بنابراین مجموعه حداقل FD ها به صورت زیر است:

$A \rightarrow B$

$B \rightarrow C$

$C \rightarrow E$

۷- گزینه (۳) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیربدهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A,B,C,D,E,F)$ داریم:

$AB \rightarrow C$

$DB \rightarrow E$

$C \rightarrow F$

$B \rightarrow CE$

$A \rightarrow BE$

$D \rightarrow C$

$ABCDEF - BCEF = AD$

بنابر رابطه فوق صفات AD حتماً باید عضو کلید کاندید باشد. بستار صفات AD به صورت زیر است:

$$\{AD\}^+ = \{A, D, B, C, E, F\}$$

براساس بستار فوق، صفات AD، همه ستون‌ها را تولید می‌کند، پس صفات AD کلید کاندید می‌باشد.

قانون دوم ارسطو

هرگاه عضو کلید کاندید حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.
گزینه اول نادرست است. زیرا، عضو کلید کاندید AD را ندارد.
گزینه دوم نادرست است. زیرا، عضو کلید کاندید AD را ندارد.
گزینه سوم درست است. زیرا، عضو کلید کاندید AD، به تنهایی کلید کاندید است.
گزینه چهارم نادرست است. زیرا، عضو زائد B را دارد، هر چند عضو کلید کاندید AD را دارد. صفات AD همه ستون‌ها را به تنهایی با استفاده از وابستگی‌های مطرح شده تولید می‌کند، بنابراین صفت B، عضو زائد است. مجموعه صفات ABD در واقع ابرکلید است.

۸- گزینه (۲) صحیح است.

وابستگی‌های مطرح شده به صورت زیر است:

$$AC \rightarrow BD$$

$$B \rightarrow DE$$

$$A \rightarrow D$$

$$A \rightarrow C$$

$$BC \rightarrow E$$

$$D \rightarrow E$$

$$A \rightarrow E$$

مجموعه حداقل FD ها براساس قوانین زیر ایجاد می‌گردد:

۱- سمت راست همه وابستگی‌ها توسط عمل تجزیه، تک خصیصه‌ای باشد.

۲- سمت چپ همه وابستگی‌ها در صورت امکان کمینه باشد.

۳- همه وابستگی‌های زائد حذف گردد.

بنابراین، براساس قوانین فوق داریم:

(1) $AC \rightarrow B$

(2) $AC \rightarrow D$

(3) $B \rightarrow D$

(4) $B \rightarrow E$

(5) $A \rightarrow D$

(6) $A \rightarrow C$

(7) $BC \rightarrow E$

(8) $D \rightarrow E$

(9) $A \rightarrow E$

وابستگی $AC \rightarrow B$ به دلیل وجود وابستگی $A \rightarrow C$ قابل تغییر به وابستگی $A \rightarrow B$ است.

وابستگی $AC \rightarrow D$ به دلیل وجود وابستگی $A \rightarrow C$ قابل تغییر به وابستگی $A \rightarrow D$ است و چون وابستگی حاصل یعنی $A \rightarrow D$ تکراری است، این وابستگی حذف می‌گردد.

حذف وابستگی $BC \rightarrow E$ به دلیل زیر:

هدف رسیدن به صفت E است، از آن جا که صفت B در سمت راست وابستگی $BC \rightarrow E$ ، توسط وابستگی $B \rightarrow E$ ، به تنهایی قادر به تولید صفت E است، پس صفت C در سمت چپ وابستگی $BC \rightarrow E$ ، زائد است و وابستگی $B \rightarrow E$ باقی می‌ماند و چون وابستگی حاصل یعنی $B \rightarrow E$ تکراری است، این وابستگی حذف می‌گردد.

حذف وابستگی $A \rightarrow E$ به دلیل زیر:

$$\begin{cases} A \rightarrow D \\ D \rightarrow E \end{cases} \Rightarrow A \rightarrow E$$

حذف وابستگی $A \rightarrow D$ به دلیل زیر:

$$\begin{cases} A \rightarrow B \\ B \rightarrow D \end{cases} \Rightarrow A \rightarrow D$$

حذف وابستگی $B \rightarrow E$ به دلیل زیر:

$$\begin{cases} B \rightarrow D \\ D \rightarrow E \end{cases} \Rightarrow B \rightarrow E$$

بنابراین مجموعه حداقل FD ها به صورت زیر است:

$A \rightarrow B$
 $B \rightarrow D$
 $A \rightarrow C$
 $D \rightarrow E$

۹- گزینه (۴) صحیح است.

وابستگی‌های مطرح شده به صورت زیر است:

$A \rightarrow B$
 $CD \rightarrow F$
 $B \rightarrow D$
 $B \rightarrow C$
 $BE \rightarrow G$
 $G \rightarrow F$
 $AG \rightarrow B$

{سؤال: آیا می‌توان از صفت B بدون استفاده از وابستگی $B \rightarrow G$ و با استفاده از وابستگی‌های دیگر به

$$\{B \rightarrow G\}^+ = \{ \text{صفت } G \text{ رسید؟} \}$$

$$\{B\}^+ = \{B, C, D\}$$

پاسخ: خیر، نمی‌توان رسید، زیرا خصیصه B با همراهی خصیصه E می‌تواند خصیصه G را تولید کند و به تنهایی قادر به این کار نیست. بنابراین گزینه اول نادرست است.

{سؤال: آیا می‌توان از صفت G بدون استفاده از وابستگی $G \rightarrow B$ و با استفاده از وابستگی‌های دیگر به

$$\{G \rightarrow B\}^+ = \{\text{صفت } B \text{ رسید؟}\}$$

$$\{G\}^+ = \{G, F\}$$

پاسخ: خیر، نمی توان رسید، زیرا خصیصه G می تواند خصیصه F را تولید کند، اما خصیصه F در سمت چپ هیچ یک از وابستگی های تابعی نیامده است. بنابراین هیچ خصیصه دیگری را نمی تواند تولید کند. بنابراین گزینه دوم نادرست است.

سؤال: آیا می توان از صفت G بدون استفاده از وابستگی $G \rightarrow D$ و با استفاده از وابستگی های دیگر به

$$\{G \rightarrow D\}^+ = \{\text{صفت } D \text{ رسید؟}\}$$

$$\{G\}^+ = \{G, F\}$$

پاسخ: خیر، نمی توان رسید، زیرا خصیصه G می تواند خصیصه F را تولید کند، اما خصیصه F در سمت چپ هیچ یک از وابستگی های تابعی نیامده است. بنابراین هیچ خصیصه دیگری را نمی تواند تولید کند. بنابراین گزینه سوم نادرست است.

سؤال: آیا می توان از صفت A بدون استفاده از وابستگی $A \rightarrow F$ و با استفاده از وابستگی های دیگر به

$$\{A \rightarrow F\}^+ = \{\text{صفت } F \text{ رسید؟}\}$$

$$\{A\}^+ = \{A, B, C, D, \boxed{F}\}$$

پاسخ: بله، می توان رسید، روال زیر را در نظر بگیرید:

$$A \rightarrow B$$

$$\begin{cases} B \rightarrow C \\ B \rightarrow D \end{cases} \Rightarrow B \rightarrow CD$$

$$\begin{cases} B \rightarrow CD \\ CD \rightarrow F \end{cases} \Rightarrow B \rightarrow F$$

$$\begin{cases} A \rightarrow B \\ B \rightarrow F \end{cases} \Rightarrow A \rightarrow F$$

توجه: نیازی نیست، روال فوق را دنبال کنید، کافی است، با چشم، بر روی وابستگی های اصلی مطرح شده حرکت کنید. A خود را می دهد، A می دهد B را، B می دهد C و D را، C و D را داریم، پس می دهد F را، بنابراین از A به F بدون استفاده از وابستگی $A \rightarrow F$ رسیدیم، به همین سادگی، همین.

۱۰- گزینه (۱) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه های سمت راست وابستگی های غیر بدیهی - تمام خصیصه های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کانديد} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(x, y, z, w)$ داریم:

$$x \rightarrow z$$

$$y \rightarrow w$$

$$xyzw - zw = xy$$

بنابر رابطه فوق صفات xy حتماً باید عضو کلید کانديد باشد.

$$\{xy\}^+ = \{x, y, z, w\}$$

بستار صفات xy به صورت روبه‌رو است:

بر اساس بستار فوق، صفات xy ، همه ستون‌ها را تولید می‌کند، پس صفات xy کلید کانديد می‌باشد.

قانون دوم ارسطو

هرگاه عضو کلید کانديد، حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کانديد، تنها کلید کانديد جدول خواهد بود.

گزینه اول درست است. زیرا، عضو کلید کانديد xy ، به تنهایی کلید کانديد است.

گزینه دوم و چهارم نادرست است، زیرا، عضو کلید کانديد xy را ندارد. صفت x یا y به تنهایی خاصیت کلیدی ندارد. xy هر دو با هم عضو کلید کانديد هستند.

گزینه سوم نادرست است. زیرا، اعضاء زائد z و w را دارد، هر چند عضو کلید کانديد xy را دارد. صفات xy همه ستون‌ها را به تنهایی و با استفاده از وابستگی‌های مطرح شده تولید می‌کنند، بنابراین صفات w و z اعضا زائد هستند. مجموعه صفات $xyzw$ در واقع ابرکلید است.

۱۱- گزینه (۲) صحیح است.

طبق تعریف وابستگی بخشی (جزئی)، عبارت است از وابستگی یک عنصر غیرکلیدی به بخشی از کلید کانديد. از آن‌جا که مطابق صورت سؤال AB کلید اصلی (کانديد) است. پس $A \rightarrow CD$ وابستگی دو عنصر غیرکلیدی C و D به A که بخشی از کلید است، را نشان می‌دهد. بنابراین می‌توان دریافت که $A \rightarrow CD$ یک وابستگی بخشی (جزئی) تلقی می‌شود. سایر گزینه‌ها با تعریف وابستگی بخشی انطباق ندارد.

۱۲- گزینه (۱) صحیح است.

مطابق مفروضات مطرح شده در صورت سؤال، چون کلیه خصیصه‌های R وابستگی تابع کامل به A دارند، پس A حتماً کلید کانديد این رابطه است، اما این که حتماً کلید اصلی آن باشد مشخص نیست، زیرا انتخاب کلید اصلی بر عهده طراح بانک و بر اساس مورد کاربرد، از یکی از کلیدهای کانديد انتخاب می‌گردد.

۱۳- گزینه (۴) صحیح است.

به طور کلی کلید کانديد باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کانديد از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E, H, K, L)$ داریم:

$$AB \rightarrow L$$

$$C \rightarrow H$$

$$CD \rightarrow E$$

$$EH \rightarrow B$$

$$K \rightarrow C$$

$$ABCDEHKL - BCEHL = ADK$$

بنابر رابطه فوق صفات ADK حتماً باید عضو کلید کاندید باشد.

$$\{ADK\}^+ = \{A, D, K, C, E, H, B, L\}$$

بستار صفات ADK به صورت مقابل است:

براساس بستار فوق، صفات ADK، همه ستون‌ها را تولید می‌کند، پس صفات ADK کلید کاندید می‌باشد.

قانون دوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

گزینه اول نادرست است. زیرا، عضو کلید کاندید ADK را ندارد.

گزینه دوم نادرست است. زیرا، عضو کلید کاندید ADK را ندارد.

گزینه سوم نادرست است. زیرا، اعضا زائد B و H را دارد، هر چند عضو کلید کاندید ADK را دارد، صفات ADK همه ستون‌ها را به تنهایی و با استفاده از وابستگی‌های مطرح شده تولید می‌کنند، بنابراین صفات B و H، اعضاء زائد هستند. مجموعه صفات ABDHK در واقع ابرکلید است

گزینه چهارم درست است. زیرا، عضو کلید کاندید ADK، به تنهایی کلید کاندید است.

۱۴- گزینه (۲) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$S\# \text{ عضو کلید کاندید} = R - \bigcup_{i=1}^n [(چپ) x_i - (راست) y_i]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $ST(S\#, T\#, Date, Time, Code)$ و اعلام $(S\#, T\#)$ به عنوان کلید کاندید از طرف طراح سؤال، داریم:

$S\#, T\# \rightarrow Date$

$S\#, T\# \rightarrow Time$

$S\#, T\# \rightarrow Code$

$Code \rightarrow T\#$

$S\# \cancel{T\#} \cancel{Date} \cancel{Time} \cancel{Code} - T\# \cancel{Date} \cancel{Time} \cancel{Code} = S\#$

بنابر رابطه فوق صفت $S\#$ حتماً باید عضو کلید کاندید باشد. بستار صفت $S\#$ به صورت زیر است:

$$\{S\#\}^+ = \{S\#\}$$

براساس بستار فوق، صفت $S\#$ ، فقط ستون $S\#$ را تولید می‌کند، پس صفت $S\#$ فقط عضو کلید کاندید می‌باشد و کلید کاندید نمی‌باشد.

قانون سوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول، برخی از ستون‌ها را تولید کند، بدین معنی است که، جدول موردنظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت.

با کمی دقت و بررسی، پُر واضح است که اگر صفت $T\#$ در کنار عضو کلید $S\#$ قرار بگیرد، همکاری صفات $(S\#, T\#)$ می‌تواند، همه ستون‌ها را تولید کند، بنابراین مطابق آنچه در صورت سؤال هم آمده است، $(S\#, T\#)$ یک کلید کاندید این جدول است.

بستار صفات $(S\#, T\#)$ به صورت زیر است:

$$\{S\#, T\#\}^+ = \{S\#, T\#, Date, Time, Code\}$$

براساس بستار فوق، صفات $(S\#, T\#)$ ، همه ستون‌ها را تولید می‌کنند، پس صفات $(S\#, T\#)$ کلید کاندید است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد.

هم‌چنین از آن‌جا که $Code \rightarrow T\#$. پس می‌توان ترکیب دو خصیصه $(S\#, Code)$ را هم کلید کاندید دیگری برای این جدول تلقی کرد. چون وقتی $(S\#, T\#)$ کلید کاندید است و همه ستون‌ها را تولید می‌کند، پس $(S\#, Code)$ هم کلید کاندید است و همه ستون‌ها را تولید می‌کند، زیرا در نهایت طبق وابستگی $Code \rightarrow T\#$ ، صفت $Code$ ، صفت $T\#$ را می‌دهد و $(S\#, Code)$ به $(S\#, T\#)$ تبدیل می‌گردد.

بستار صفات $(S\#, Code)$ به صورت زیر است:

$$\{S\#, Code\}^+ = \{S\#, Code, T\#, Date, Time\}$$

بر اساس بستار فوق، صفات $(S\#, Code)$ ، همه ستون‌ها را تولید می‌کنند، پس صفات $(S\#, Code)$ کلید کاندید است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده در مجموع دارای دو کلید کاندید است، که مطابق قانون سوم ارسطو عضو کلید کاندید $S\#$ در بین هر دو کلید کاندید به طور مشترک قرار دارد.

توجه: طراح بانک، کلید کاندید $(S\#, T\#)$ را با رسم خط، به عنوان کلید اصلی انتخاب کرده است، بنابراین کلید کاندید $(S\#, Code)$ به عنوان کلید فرعی تلقی می‌گردد. این نتیجه مهم بود.

۱۵- گزینه (۱) صحیح است.

گزینه اول نادرست است. زیرا، وابستگی‌های مطرح شده به صورت زیر است:

$$\begin{aligned} AB &\rightarrow C \\ D &\rightarrow AE \\ AF &\rightarrow C \\ EF &\rightarrow GA \end{aligned}$$

{سؤال: آیا می‌توان از صفات AD با استفاده از وابستگی‌های فوق به صفت C رسید؟}

$$\{AD \rightarrow C\}^+ = \{AD\}^+ = \{A, D, E\}$$

پاسخ: خیر، نمی‌توان رسید. زیرا A ، A را می‌دهد و D ، D را می‌دهد و D ، A و E را می‌دهد و کار تمام می‌شود. به بیان دیگر، وابستگی تابعی $AD \rightarrow C$ قابل استنتاج از وابستگی‌های تابعی داده شده در گزینه اول نیست.

گزینه دوم درست است. زیرا، وابستگی‌های مطرح شده به صورت زیر است:

$$\begin{aligned} AB &\rightarrow C \\ D &\rightarrow AE \\ AE &\rightarrow C \\ EF &\rightarrow GA \end{aligned}$$

{سؤال: آیا می‌توان از صفات DF با استفاده از وابستگی‌های فوق به صفت G رسید؟}

$$\{DF \rightarrow G\}^+ = \{DF\}^+ = \{D, F, A, E, \boxed{G}\}$$

پاسخ: بله، می‌توان رسید، زیرا D ، D را می‌دهد، F ، F را می‌دهد و D ، A و E را می‌دهد و E و F هم G و A را می‌دهد.

راه حل کلاسیک:

$$\left. \begin{aligned} D &\rightarrow AE \Rightarrow D \rightarrow E \\ EF &\rightarrow GA \end{aligned} \right\} \Rightarrow DF \rightarrow GA \Rightarrow DF \rightarrow G$$

گزینه سوم درست است. زیرا، وابستگی‌های مطرح شده به صورت زیر است:

$$\begin{aligned} X &\rightarrow Y \\ YZ &\rightarrow W \end{aligned}$$

{سؤال: آیا می‌توان از صفات XZ با استفاده از وابستگی‌های فوق به صفت W رسید؟}

$$\{XZ \rightarrow W\}^+ = \{XZ\}^+ = \{X, Z, Y, \boxed{W}\}$$

پاسخ: بله، می توان رسید، زیرا X, X را می دهد و Z, Z را می دهد و Y, X را می دهد و Z و Y هم W را می دهد.

راه حل کلاسیک:

$$\left. \begin{array}{l} X \rightarrow Y \\ YZ \rightarrow W \end{array} \right\} \Rightarrow XZ \rightarrow W$$

گزینه چهارم درست است. زیرا، وابستگی های مطرح شده به صورت زیر است:

$$\begin{array}{l} X \rightarrow Y \\ Y \rightarrow Z \\ X \rightarrow W \end{array}$$

{سؤال: آیا می توان از صفات XZ با استفاده از وابستگی های فوق به صفت W رسید؟}

$$\{XZ \rightarrow W\}^+ = \{XZ\}^+ = \{X, Z, Y, \overline{W}\}$$

پاسخ: بله، می توان رسید، زیرا X, X را می دهد و Z, Z را می دهد و Y, X را می دهد.

راه حل کلاسیک:

$$\left\{ \begin{array}{l} X \rightarrow Y \\ Y \rightarrow Z \end{array} \right\} \Rightarrow X \rightarrow Z$$

$$\left\{ \begin{array}{l} X \rightarrow Z \\ X \rightarrow W \end{array} \right\} \Rightarrow XZ \rightarrow W$$

۱۶- گزینه (۲) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه های سمت راست وابستگی های غیر بدیهی - تمام خصیصه های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

با توجه به وابستگی های مطرح شده برای رابطه $R(A, B, C, D, E, F, G, H)$ داریم:

$$ABC \rightarrow DE$$

$$G \rightarrow DE$$

$$E \rightarrow F$$

$$H \rightarrow E$$

$$F \rightarrow G$$

$$ABCDEF GH - DEFG = ABCH$$

بنابر رابطه فوق صفات ABCH حتماً باید عضو کلید کاندید باشد. بستار صفات ABCH به صورت زیر است:

$$\{ABCH\}^+ = \{A, B, C, H, D, E, F, G\}$$

براساس بستار فوق، صفات ABCH، همه ستون‌ها را تولید می‌کند، پس صفات ABCH کلید کاندید می‌باشد.

قانون دوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

گزینه اول نادرست است. زیرا، عضو کلید کاندید ABCH را ندارد.

گزینه دوم درست است. زیرا، عضو کلید کاندید ABCH، به تنهایی کلید کاندید است.

گزینه سوم نادرست است. زیرا، عضو کلید کاندید ABCH را ندارد.

گزینه چهارم نادرست است. زیرا، عضو کلید کاندید ABCH را ندارد.

۱۷- گزینه (۳) صحیح است.

طبق تعریف، وابستگی یک عنصر غیرکلیدی به یک عنصر غیرکلیدی دیگر وابستگی انتقالی، وابستگی یک عنصر غیرکلیدی به کلید کاندید، وابستگی تابعی کامل و وابستگی یک عنصر غیرکلیدی به بخشی از کلید کاندید، وابستگی بخشی خوانده می‌شود.

گزینه اول نادرست است. زیرا، $AB \rightarrow C$ ، وابستگی تابعی کامل است.

گزینه دوم نادرست است. زیرا، $A \rightarrow DEH$ ، وابستگی بخشی است.

گزینه سوم درست است. زیرا $D \rightarrow EH$ ، وابستگی انتقالی است.

گزینه چهارم نادرست است. زیرا، $B \rightarrow D$ ، وابستگی بخشی است.

۱۸- گزینه (۴) صحیح است.

مثال: جدول $R(a, b, c, d)$ را با وابستگی‌های زیر در نظر بگیرید:

$$a \rightarrow b$$

$$b \rightarrow c$$

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(a,b,c,d)$ داریم:

$$a \neq b \vee d - bc = ad$$

بنابر رابطه فوق صفات ad حتماً باید عضو کلید کاندید باشد. بستار صفات ad به صورت زیر است:

$$\{ad\}^+ = \{a, d, b, c\}$$

بر اساس بستار فوق، صفات ad ، همه ستون‌ها را تولید می‌کند، پس صفات ad کلید کاندید می‌باشد.

قانون دوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود. یک کلید کاندید وجود دارد که کلید اصلی هم خواهد بود و بدون کلید فرعی. در یک قاعده کلی، صفات سمت راست وابستگی‌های غیربدیهی، جزء کلید کاندید (اصلی) نخواهند بود.

حال گزاره مطرح شده را در نظر بگیرید:

$$\forall X \subseteq U \forall A \in U : X \rightarrow A \in F^+ \wedge A \notin X$$

شرط گزاره

گزاره فوق بدین معنی است که X زیر مجموعه‌ای از صفات U است و A صفتی از صفات U است.

حال اگر مطابق شرط $X \rightarrow A \in F^+$ بدین معنی که زیر مجموعه صفاتی، صفت A را بدهد، بدین معنی است که A به X وابستگی تابعی دارد. شرط بعدی یعنی $A \notin X$ بیان می‌کند که A متعلق به X نباشد، چون در این صورت وابستگی بدیهی ایجاد می‌کند. مانند $ab \rightarrow b$ که ab ، صفت b را می‌دهد و b زیر مجموعه ab است. بنابراین منظور از عبارت $X \rightarrow A \in F^+ \wedge A \notin X$ وابستگی‌هایی است که بدیهی نباشد.

مانند وابستگی‌های غیربدیهی زیر برای $R(a,b,c,d)$ به عبارت دیگر غیربدیهی باشند.

$$\begin{array}{c} X \\ \bar{a} \rightarrow \bar{b} \end{array}$$

$$\begin{array}{c} X \\ \bar{b} \rightarrow \bar{c} \end{array}$$

همان‌طور که پیش از این بیان شد، کلید کاندید جدول فوق ad می‌باشد، که صورت سوال، کلید کاندید را K نامگذاری کرده است.

$$ad \equiv k$$

پرواضح است که صفات b و c جزء کلید کاندید (اصلی) نیستند. زیرا در یک قاعده کلی صفات سمت راست وابستگی‌های غیربدیهی، جزء کلید کاندید (اصلی) نخواهند بود.

گزینه اول نادرست است. زیرا، $X \subseteq K$ (به معنی هر زیر مجموعه‌ای از صفات جدول R ، با نام X زیر مجموعه کلید کاندید است) همواره برقرار نیست.

مثال: $\bar{X} \not\subseteq \bar{K}$ ، یعنی b زیر مجموعه ad نیست.

گزینه دوم نادرست است، زیرا $X \not\subseteq K$ (به معنی هر زیر مجموعه‌ای از صفات جدول R با نام X ، زیر مجموعه کلید کاندید نیست) همواره برقرار نیست.

مثال: $\bar{X} \not\subseteq K$ ، یعنی a زیر مجموعه ad نیست.
گزینه سوم نادرست است. زیرا، اگر جدولی مطابق آنچه گفتیم، فقط یک کلید کاندید داشته باشد، امکان ندارد، صفات سمت راست وابستگی‌ها جزء کلید کاندید (اصلی) جدول باشد. بنابراین گزینه چهارم درست است.

۱۹- گزینه (۲) صحیح است.

مجموعه‌ی پوششی A (یعنی مجموعه‌ی تمام صفت‌هایی که با A قابل دسترسی‌اند) عبارت است از:
 $\{A\}^* = \{A, B, C, D\}$
 ملاحظه می‌شود که فقط صفت E در مجموعه‌ی فوق نیست پس A و E به همراه هم کلیدند.
 به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:
 ۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.
 ۲- عضو زائد نداشته باشد.
 به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i - (چپ) y_i \text{ (راست)}]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E)$ داریم:

$$A \rightarrow B$$

$$AB \rightarrow CD$$

$$D \rightarrow ABC$$

$$ABCDE - ABCD = E$$

بنابر رابطه فوق صفت E حتماً باید عضو کلید کاندید باشد. بستر صفت E به صورت زیر است:

$$\{E\}^+ = \{E\}$$

براساس بستر فوق، صفت E ، فقط ستون E را تولید می‌کند، پس صفت E فقط عضو کلید کاندید می‌باشد و کلید کاندید نمی‌باشد.

قانون سوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، برخی از ستون‌ها را تولید کند، بدین معنی است که، جدول موردنظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت.

با کمی دقت و بررسی، پُر واضح است که اگر صفت A در کنار عضو کلید کاندید E قرار بگیرد، همکاری صفات (A و E) می‌تواند، همه ستون‌ها را تولید کند، بنابراین صفات AE کلید کاندید جدول R خواهد بود. بستار صفات AE به صورت زیر است:

$$\{AE\}^+ = \{A, E, B, C, D\}$$

بر اساس بستار فوق، صفات AE، همه ستون‌ها را تولید می‌کنند، پس صفات AE، کلید کاندید است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد.

همچنین از آن‌جا که $D \rightarrow A$ ، پس می‌توان ترکیب دو خصیصه (D, E) را هم کلید کاندید دیگری برای این جدول تلقی کرد.

چون وقتی (A, E) کلید کاندید است و همه ستون‌ها را تولید می‌کند، پس (D, E) هم کلید کاندید است و همه ستون‌ها را تولید می‌کند، زیرا در نهایت طبق وابستگی $D \rightarrow A$ ، صفت D، صفت A را می‌دهد و (D, E) به (A, E) تبدیل می‌گردد.

بستار صفات (D, E) به صورت زیر است:

$$\{D, E\}^+ = \{D, E, A, B, C\}$$

بر اساس بستار فوق، صفات (D, E)، همه ستون‌ها را تولید می‌کنند، پس صفات (D, E) کلید کاندید است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده، در مجموع، دارای دو کلید کاندید است که مطابق قانون سوم ارسطو، عضو کلید کاندید E در بین هر دو کلید کاندید به طور مشترک قرار دارد.

گزینه اول نادرست است زیرا، عضو کلید کاندید E را ندارد.

گزینه دوم درست است. زیرا، صفات AE کلید کاندید جدول R است.

گزینه سوم نادرست است. زیرا، عضو کلید کاندید E را ندارد.

گزینه چهارم نادرست است. زیرا، عضو کلید کاندید E را ندارد.

۲۰- گزینه (۱) صحیح است.

با توجه به نمودار FD داده شده، وابستگی‌های زیر تعریف شده‌اند:

- × (1) $M \rightarrow N$
- × (2) $M \rightarrow O$
- × (3) $M \rightarrow P$
- ✓ (4) $N \rightarrow O$
- ✓ (5) $N \rightarrow P$
- ✓ (6) $P \rightarrow N$
- × (7) $MN \rightarrow O$

می‌توان دید که وابستگی (7) از روی وابستگی‌های (2) و (4) قابل نتیجه‌گیری است، پس می‌توان وابستگی (7) را کنار گذاشت. بنابراین گزینه دوم نادرست است.

هم‌چنین وابستگی (2) از روی وابستگی‌های (1) و (4) به دست می‌آید، پس وابستگی (2) را نیز کنار می‌گذاریم. بنابراین گزینه چهارم نادرست است.

وابستگی (3) نیز از روی وابستگی‌های (1) و (5) به دست می‌آید، پس وابستگی (3) را نیز کنار می‌گذاریم. بنابراین گزینه سوم نادرست است.

وابستگی (1) نیز از روی وابستگی‌های (3) و (6) به دست می‌آید. پس وابستگی (1) را نیز کنار می‌گذاریم.

توجه: وقتی یک وابستگی، کنار گذاشته می‌شود، به معنی عدم استفاده از آن نیست، زیرا توسط وابستگی‌های دیگر مجدداً قابل تولید است. بنابراین می‌توان از آن استفاده نمود.

وابستگی‌های باقیمانده (یعنی مجموعه‌ی حداقل FDها) به صورت زیر خواهد بود.

$$(4) \quad N \rightarrow O$$

$$(5) \quad N \rightarrow P$$

$$(6) \quad P \rightarrow N$$

۲۱- گزینه (۲) صحیح است.

با توجه به سه وابستگی تابعی اول (از سمت چپ)، مشاهده می‌شود که از بین صفت‌های A و B و C با داشتن هر کدام می‌توان مقدار دوتای دیگر را تعیین کرد. هم‌چنین با توجه به دو وابستگی تابعی آخر، مشاهده می‌شود که با داشتن هر یک از صفات D و E می‌توان دیگری را به دست آورد.

برای ساختن یک ابرکلید باید از بین سه صفت A و B و C حداقل یکی و از بین دو صفت D و E نیز حداقل یکی را انتخاب کرد. برای انتخاب حداقل یک صفت از بین سه صفت A و B و C یا باید یکی از آن‌ها را انتخاب کنیم یا دو تا یا سه تا را. تعداد حالات ممکن برای انتخاب حداقل یک صفت از بین سه صفت A و B و C برابر است با:

$$\binom{3}{1} + \binom{3}{2} + \binom{3}{3} = 7$$

به طور مشابه، تعداد حالات ممکن برای انتخاب حداقل یک صفت از بین دو صفت D و E برابر است با:

$$\binom{2}{1} + \binom{2}{2} = 3$$

بنابراین:

$$7 \times 3 = 21 = \text{تعداد ابرکلیدهای ممکن}$$

برای ساختن یک کلید کاندید، باید از بین سه صفت A و B و C دقیقاً یک صفت و از بین دو صفت D و E نیز دقیقاً یک صفت انتخاب کنیم. تعداد حالات ممکن برابر خواهد بود با:

$$\binom{3}{1} + \binom{2}{1} = 3 \times 2 = 6 = \text{تعداد کلیدهای کاندید ممکن}$$

۲۲- گزینه (۱) صحیح است.

از آن‌جا که حرف A در سمت راست هیچ یک از وابستگی‌های تابعی وجود ندارد. بنابراین حرف A حتماً جزئی از کلید کاندید است. بنابراین $\{A, \dots\}$ کلید کاندید است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A,B,C,D,E)$ داریم:

$$A \rightarrow BC$$

$$CD \rightarrow E$$

$$\square \rightarrow D$$

$$ABCDE - BCDE = A$$

بنابر رابطه فوق صفت A حتماً باید عضو کلید کاندید باشد. بنابراین گزاره I درست است. بستار صفت A به صورت زیر است:

$$\{A\}^+ = \{A, B, C, \dots\}$$

توجه: چند نقطه (...) برای این است که نمی‌دانیم سمت چپ وابستگی $\square \rightarrow D$ چیست. اما آنچه که پر واضح است، این است که کلید کاندید تحت هیچ شرایطی شامل صفات B و C نخواهد بود، زیرا توسط صفت A ، به عنوان عضو کلید کاندید قابل تولید است. بنابراین گزاره II درست و گزاره III نادرست است. در نهایت مشخص می‌شود که گزینه اول درست است.

۲۳- گزینه (۴) صحیح است.

توجه: گزینه چهارم، همان قانون اول و دوم ارسطو می‌باشد.

جدول $R = (A, B, C, D, E, H)$ با وابستگی‌های $F = \{A \rightarrow B, B \rightarrow C, D \rightarrow E, H \rightarrow A\}$ را در نظر بگیرید: مطابق رابطه گزینه چهارم داریم:

$$V - \bigcup_{i=1}^n (y_i - x_i)$$

$$y_1 - x_1 = A \rightarrow B \Rightarrow B - A = B$$

$$y_2 - x_2 = B \rightarrow C \Rightarrow C - B = C$$

$$y_3 - x_3 = D \rightarrow E \Rightarrow E - D = E$$

$$y_4 - x_4 - H \rightarrow A \Rightarrow A - H = A$$

بنابراین عبارت $\bigcup_{i=1}^n (y_i - x_i)$ در یک مجموعه وابستگی تابعی، صفاتی که سمت راست وابستگی‌های

تابعی هستند و جزء دترمینانها (سمت چپ رابطه) نیستند را می‌دهد. سپس عبارت $V - \sum_{i=1}^n (y_i - x_i)$ این صفات را از مجموعه صفات حذف می‌کند. صفاتی که از مجموعه V باقی می‌مانند، عضو کلید کاندید رابطه خواهند بود.

$$(A, B, C, D, E, H) - (ABCE) = DH$$

بنابر رابطه فوق صفات DH باید عضو کلید کاندید باشد.

بستار صفات DH به صورت زیر است:

$$\{DH\}^+ = \{D, H, E, A, B, C\}$$

براساس بستار فوق، صفات DH همه ستون‌ها را تولید می‌کند، پس صفات DH کلید کاندید می‌باشد.

قانون دوم ارسطو

هرگاه عضو کلید کاندید حاصل از تفاضل قانون اول (روش اول یا دوم) همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

۲۴- گزینه (۱) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \sum_{i=1}^n [(چپ) x_i - (راست) y_i]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E, F, G, H, I, J)$ داریم:

$$ABD \rightarrow EG$$

$$C \rightarrow DG$$

$$E \rightarrow FG$$

$$I \rightarrow H$$

$$H \rightarrow J$$

$$AB \rightarrow C$$

$$G \rightarrow F$$

$$ABCDEFGHIJ - CDEFGHJ = ABI$$

بنابر رابطه فوق صفات ABI حتماً باید عضو کلید کاندید باشد. بستار صفات ABI به صورت زیر است:

$$\{ABI\}^+ = \{A, B, I, C, H, D, G, E, F, J\}$$

بر اساس بستار فوق، صفات ABI، همه ستون‌ها را تولید می‌کند، پس صفات ABI کلید کاندید می‌باشد.

قانون دوم ارسطو

هرگاه عضو کلید کاندید حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کاندید جدول خواهد بود.

گزینه اول درست است. زیرا، عضو کلید کاندید ABI، به تنهایی کلید کاندید است.

گزینه دوم نادرست است. زیرا، عضو کلید کاندید ABI را ندارد.

گزینه سوم نادرست است. زیرا، عضو کلید کاندید ABI را ندارد.

گزینه چهارم نادرست است. زیرا، عضو زائد D را دارد، هر چند عضو کلید کاندید ABI را دارد.

صفات ABI، همه ستون‌ها را به تنهایی با استفاده از وابستگی‌های مطرح شده تولید می‌کنند، بنابراین صفت D، عضو زائد است. مجموعه صفات ABDI در واقع ابرکلید است.

۲۵- گزینه (۳) صحیح است.

با کمی بررسی می‌توان دید که رابطه‌ی داده شده، با توجه به وابستگی‌های تابعی آن، دو کلید کاندید دارد: DC و ABC. (توجه کنید که کلید کاندید باید کمینه باشد یعنی زیر مجموعه‌ی محضی از صفات آن، کلید نباشد) برای به دست آوردن ابرکلیدها، باید به هر یک از دو کلید کاندید فوق، صفر یا بیش‌تر از صفات‌های دیگر را اضافه کرد. بنابراین باید به کلید کاندید DC، صفر، یک، دو یا سه مورد از صفات‌های دیگر (E, B, A) اضافه کرد. برای این کار ۸ حالت، وجود دارد:

$$\binom{3}{0} + \binom{3}{1} + \binom{3}{2} + \binom{3}{3} = 2^3 = 8$$

به طور مشابه، باید به کلید کاندید ABC، صفر، یک یا دو صفت از بین صفات‌های دیگر (E, D) اضافه کرد. برای این کار ۴ حالت وجود دارد:

$$\binom{2}{0} + \binom{2}{1} + \binom{2}{2} = 2^2 = 4$$

ترکیب اول: هیچ یا ترکیبی از صفات باقی مانده + کلید کاندید DC = ابرکلید

DC +	A B E	ترکیبات صفات باقی مانده	ابرکلیدها
	0 0 0	تهی	DC
	0 0 1	E	DCE
	0 1 0	B	DCB
	0 1 1	BE	DCBE
	1 0 0	A	DCA
	1 0 1	AE	DCAE
	1 1 0	AB	DCAB
	1 1 1	ABE	DCABE

توجه: واضح است که 8 ابرکلید، ایجاد می‌گردد. (2^3)
ترکیب دوم: هیچ یا ترکیبی از صفات باقیمانده + کلید کاندید ABC = ابرکلید

ED	ترکیبات صفات باقی مانده	ابرکلیدها
00	تهی	ABC
01	D	ABCD
10	E	ABCE
11	ED	ABCED

توجه: واضح است که 4 ابرکلید، ایجاد می‌گردد. (2^2)
بنابراین حاصل جمع ابرکلیدها 12 عدد خواهد بود، که از این مجموعه حاصل، ابرکلیدهای ABCDE و ABCD در دو مجموعه تکراری هستند، بنابراین با کنار گذاشتن ابرکلیدهای تکراری، در نهایت 10 ابرکلید خواهیم داشت. به همین سادگی.

۲۶- گزینه (۱) صحیح است.

راه حل اول: کلاسیک

گزینه اول: برقرار نیست، چون از روی E و C نمی‌توان به D رسید.
گزینه دوم درست است، چون $E \rightarrow C$ ، پس واضح است که $AE \rightarrow C$.
گزینه سوم درست است.
چون صفت سمت راست \rightarrow زیر مجموعه صفات سمت چپ \rightarrow است. (وابستگی بدیهی)
گزینه چهارم درست است، چون از دو وابستگی اول و دوم به دست می‌آید. $AC \rightarrow D$.
راه حل دوم: وابستگی‌های مطرح شده به صورت زیر است:

$$A \rightarrow B$$

$$BC \rightarrow D$$

$$E \rightarrow C$$

گزینه اول درست است، زیرا:

{سؤال: آیا می‌توان از صفات CE با استفاده از وابستگی‌های فوق به صفت D رسید؟}

$$\{CE \rightarrow D\}^+ = \{CE\}^+ = \{C, E\}$$

پاسخ: خیر نمی‌توان رسید، زیرا، صفات CE، C و E را می‌دهد و کار تمام می‌شود.

گزینه دوم نادرست است، زیرا:

{سؤال: آیا می‌توان از صفات AE با استفاده از وابستگی‌های فوق به صفت C رسید؟}

$$\{AE \rightarrow C\}^+ = \{AE\}^+ = \{A, E, B, C, D\}$$

پاسخ: بله می‌توان رسید، زیرا، صفات AE، A و E را می‌دهد. A، B را می‌دهد و E، C را می‌دهد و BC را داریم پس D را هم می‌دهد.

گزینه سوم نادرست است. زیرا، وابستگی بدیهی است و واضح است که حتماً BC ، B را تولید می‌کند. گزینه چهارم نادرست است. زیرا:

{سؤال: آیا می‌توان از صفات AC با استفاده از وابستگی‌های فوق به صفت D رسید؟}

$$\{AC \rightarrow D\}^+ = \{C\}^+ = \{A, C, B, \overline{D}\}$$

پاسخ: بله، می‌توان رسید، زیرا AC ، A و C را تولید می‌کند. A ، B را می‌دهد، BC را داریم، پس D را هم می‌دهد.

۲۷- گزینه (۲) صحیح است.

رابطه $A \rightarrow C$ را می‌توان از روی روابط $A \rightarrow B$ و $B \rightarrow C$ به دست آورد، پس افزونه است.

۲۸- گزینه (۴) صحیح است.

از روی وابستگی تابعی $C \rightarrow AB$ می‌توان نتیجه گرفت، $C \rightarrow B$ و $C \rightarrow A$.

از $C \rightarrow B$ و $B \rightarrow D$ می‌توان نتیجه گرفت: $C \rightarrow D$.

از $C \rightarrow A$ و $C \rightarrow D$ نتیجه می‌گیریم: $C \rightarrow AD$.

از $C \rightarrow AD$ و $AD \rightarrow E$ نتیجه می‌گیریم: $C \rightarrow E$.

۲۹- گزینه (۱) صحیح است.

ابتدا وابستگی‌های تابعی را بازنویسی می‌کنیم. طوری که سمت راست هر وابستگی فقط یک صفت باشد:

$$(1) A \rightarrow B \quad (2) A \rightarrow C \quad (3) A \rightarrow D \quad (4) A \rightarrow K$$

$$(5) K \rightarrow C \quad (6) B \rightarrow D \quad (7) BC \rightarrow D$$

دید می‌شود که با داشتن وابستگی (6) می‌توان وابستگی (7) را نتیجه گرفت پس وابستگی (7) اضافی است و باید حذف شود.

هم‌چنین از وابستگی‌های (1) و (6) می‌توان وابستگی (3) را نتیجه گرفت پس وابستگی (3) اضافی است و حذف می‌شود. به طور مشابه، از وابستگی‌های (4) و (5) می‌توان وابستگی (2) را نتیجه گرفت پس وابستگی (2) اضافی است و باید حذف شود. بنابراین، وابستگی‌های زیر باقی می‌مانند:

$$(1) A \rightarrow B \quad (4) A \rightarrow K \quad (5) K \rightarrow C \quad (6) B \rightarrow D$$

۳۰- گزینه (۴) صحیح است.

{سؤال: آیا می‌توان از صفت A با استفاده از وابستگی‌های زیر به صفت C رسید؟}

$$\{A \rightarrow C\}^+ =$$

$$A \rightarrow B$$

$$AB \rightarrow C$$

$$\{A\}^+ = \{A, B, \overline{C}\}$$

پاسخ: بله، می‌توان رسید، A خود را می‌دهد، A ، B را می‌دهد و AB را داریم، پس C را می‌دهد. بنابراین گزینه اول درست است.

{سؤال: آیا می‌توان از صفات AC با استفاده از وابستگی‌های زیر به صفت D رسید؟}

$$\{AC \rightarrow D\}^+ =$$

$$A \rightarrow B$$

$$BC \rightarrow D$$

$$\{AC\}^+ = \{A, C, B, \boxed{D}\}$$

پاسخ: بله، می‌توان رسید، A و C خود را می‌دهند، A، B را می‌دهد و BC را داریم، پس D را می‌دهد. بنابراین گزینه دوم درست است.

{سؤال: آیا می‌توان از صفات AC با استفاده از وابستگی‌های زیر به صفات BD رسید؟}

$$\{AC \rightarrow BD\}^+ =$$

$$A \rightarrow B$$

$$AC \rightarrow D$$

$$\{AC\}^+ = \{A, C, B, D\}$$

پاسخ: بله، می‌توان رسید، A و C خود را می‌دهند، A، B را می‌دهد و AC را داریم، پس D را می‌دهد. بنابراین گزینه سوم درست است.

{سؤال: آیا می‌توان از صفت A با استفاده از وابستگی‌های زیر به صفت D رسید؟}

$$\{A \rightarrow D\}^+ =$$

$$AB \rightarrow BC$$

$$C \rightarrow D$$

$$\{A\}^+ = \{A\}$$

پاسخ: خیر، نمی‌توان رسید، A خودش را می‌دهد و کار تمام می‌شود.

۳۱- گزینه (۲) صحیح است.

به راحتی می‌توان مجموعه پوششی هر یک از مجموعه صفات داده شده را محاسبه کرد.

$$\{A\}^* = \{A\}, C^* = \{A, B, C, D, E\}, \{AC\}^* = \{A, B, C, D, E\}$$

$$\{AD\}^* = \{A, D, E\}, \{AB\}^* = \{A, B, C, D, E\}, \{CD\}^* = \{A, B, C, D, E\}$$

مشاهده می‌شود که تمام صفت‌ها از C، AC، AB و CD قابل دسترسی‌اند. پس این مجموعه از صفت‌ها ابرکلید هستند. CD ابرکلید است، اما کلید کاندید نیست زیرا صفت D عضو زائد است، چون صفت C به تنهایی همه ستون‌ها را تولید می‌کند. هم‌چنین AC ابرکلید است، اما کلید کاندید نیست زیرا صفت A عضو زائد است چون صفت C به تنهایی همه ستون‌ها را تولید می‌کند. C و AB ابرکلید هستند و کلید کاندید هم هستند چون عضو زائد ندارند.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E)$ داریم:

$$AB \rightarrow C$$

$$C \rightarrow A$$

$$C \rightarrow BD$$

$$D \rightarrow E$$

$$ABCDE - ABCDE = \text{تهی}$$

قانون چهارم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، تهی گردد، بدین معنی است که، جدول فوق چندین کلید کاندید دارد، که هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید آن وجود ندارد. بنابراین باید کلید کاندید با بررسی دقیق بر روی مجموعه وابستگی کشف گردد.

$$\{C\}^+ = \{A, B, C, D, E\}$$

بر اساس بستار فوق، صفت C، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفت C، کلید کاندید می‌باشد.

$$\{AB\}^+ = \{A, B, C, D, E\}$$

بر اساس بستار فوق، صفات AB، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفات AB، کلید کاندید می‌باشد.

توجه: همان‌طور که مشاهده می‌شود، مطابق قانون چهارم ارسطو، هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید فوق وجود ندارد.

۳۲- گزینه (۱) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E)$ داریم:

$$A \rightarrow BC$$

$$CD \rightarrow E$$

$$B \rightarrow D$$

$$E \rightarrow A$$

$$ABCDE - ABCDE = \text{تهی}$$

قانون چهارم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، تهی گردد، بدین معنی است که، جدول فوق چندین کلید کاندید دارد، که هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید آن وجود ندارد. بنابراین باید کلید کاندید با بررسی دقیق بر روی مجموعه وابستگی کشف گردد.

$$\{B\}^+ = \{B, D\}$$

بر اساس بستار فوق، صفت B، فقط ستون‌های B و D را تولید می‌کند، پس صفت B، کلید کاندید نمی‌باشد. بنابراین گزینه اول پاسخ سوال خواهد بود.

$$\{A\}^+ = \{A, B, C, D, E\}$$

بر اساس بستار فوق، صفت A، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفت A، کلید کاندید می‌باشد.

$$\{E\}^+ = \{E, A, B, C, D\}$$

بر اساس بستار فوق، صفت E، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفت E، کلید کاندید می‌باشد.

$$\{BC\}^+ = \{B, C, D, E, A\}$$

بر اساس بستار فوق، صفات BC، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفات BC، کلید کاندید می‌باشد.

$$\{CD\}^+ = \{C, D, E, A, B\}$$

بر اساس بستار فوق، صفات CD، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفات CD، کلید کاندید می‌باشد.

توجه: همان‌طور که مشاهده می‌شود، مطابق قانون چهارم ارسطو، هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید فوق وجود ندارد.

۳۳- گزینه (۲) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیربدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E)$ داریم:

$$(A, B) \rightarrow C$$

$$B \rightarrow D$$

$$(D, E) \rightarrow C$$

$$ABCDE - CD = ABE$$

بنابر رابطه فوق صفات ABE حتماً باید عضو کلید کاندید باشد. بستر صفات ABE به صورت زیر است:

$$\{ABE\}^+ = \{A, B, E, C, D\}$$

بر اساس بستر فوق، صفات ABE ، همه ستون‌ها را تولید می‌کند، پس صفات ABE کلید کاندید می‌باشد.

قانون دوم ارسطو

هرگاه عضو کلید کاندید حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

گزینه اول نادرست است. زیرا، عضو کلید کاندید ABE را ندارد.

گزینه دوم درست است. زیرا، عضو کلید کاندید ABE ، به تنهایی کلید کاندید است.

گزینه سوم نادرست است. زیرا، عضو زائد C را دارد، هر چند عضو کلید کاندید ABE را دارد. صفات ABE همه ستون‌ها را به تنهایی با استفاده از وابستگی‌های مطرح شده تولید می‌کند، بنابراین صفت C ، عضو زائد است. مجموعه صفات $ABCE$ در واقع ابرکلید است.

گزینه چهارم نادرست است. زیرا، عضو زائد D را دارد، هر چند عضو کلید کاندید ABE را دارد. صفات ABE همه ستون‌ها را به تنهایی با استفاده از وابستگی‌های مطرح شده تولید می‌کند، بنابراین صفت D ، عضو زائد است. مجموعه صفات $ABDE$ در واقع ابرکلید است.

۳۴- گزینه (۴) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [(چپ) x_i - (راست) y_i]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D)$ داریم:

$$AB \rightarrow CD$$

$$D \rightarrow B$$

$$ABC \cap D - BCD = A$$

بنابر رابطه فوق صفت A حتماً باید عضو کلید کاندید باشد. بستار صفت A به صورت زیر است:

$$\{A\}^+ = \{A\}$$

براساس بستار فوق، صفت A ، فقط ستون A را تولید می‌کند، پس صفت A فقط عضو کلید کاندید می‌باشد و کلید کاندید نمی‌باشد.

قانون سوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، برخی از ستون‌ها را تولید کند، بدین معنی است که، جدول موردنظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت.

با کمی دقت و بررسی، پُر واضح است که اگر صفت B در کنار عضو کلید کاندید A قرار بگیرد، همکاری صفات (A و B) می‌تواند، همه ستون‌ها را تولید کند، بنابراین صفات AB کلید کاندید جدول R خواهد بود. بستار صفات AB به صورت زیر است:

$$\{AB\}^+ = \{A, B, C, D\}$$

براساس بستار فوق، صفات AB ، همه ستون‌ها را تولید می‌کنند، پس صفات AB ، کلید کاندید است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد.

همچنین از آن‌جا که $D \rightarrow B$ ، پس می‌توان ترکیب دو خصیصه (A, D) را هم کلید کاندید دیگری برای این جدول تلقی کرد.

چون وقتی (A, B) کلید کاندید است و همه ستون‌ها را تولید می‌کند، پس (A, D) هم کلید کاندید است و همه ستون‌ها را تولید می‌کند، زیرا در نهایت طبق وابستگی $D \rightarrow B$ ، صفت D ، صفت B را می‌دهد و (A, D) به (A, B) تبدیل می‌گردد.

بستار صفات AD به صورت زیر است:

$$\{A, D\}^+ = \{A, D, B, C\}$$

براساس بستار فوق، صفات (A, D)، همه ستون‌ها را تولید می‌کنند، پس صفات (A, D) کلید کاندید است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده، در مجموع،

دارای دو کلید کاندید است که مطابق قانون سوم ارسطو، عضو کلید کاندید A در بین هر دو کلید کاندید به طور مشترک قرار دارد.

گزینه اول نادرست است زیرا، عضو کلید کاندید A را ندارد.

گزینه دوم نادرست است. زیرا، صفات AD هم کلید کاندید جدول R است.

گزینه سوم نادرست است. زیرا، صفات AB هم کلید کاندید جدول R است.

گزینه چهارم درست است. زیرا، صفات AB و AD هر دو کلیدهای کاندید جدول R هستند.

۳۵- گزینه (۲) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [(x_i \text{ (چپ)}) - (y_i \text{ (راست)})]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A,B,C,D,E,F)$ داریم:

$$A \rightarrow BCD$$

$$BC \rightarrow DE$$

$$B \rightarrow D$$

$$D \rightarrow A$$

$$ABCDEF - ABCDE = F$$

بنابر رابطه فوق صفت F حتماً باید عضو کلید کاندید باشد. بستار صفات F به صورت زیر است:

$$\{F\}^+ = \{F\}$$

بر اساس بستار فوق، صفت F، فقط ستون F را تولید می‌کند، پس صفت F فقط عضو کلید کاندید می‌باشد و کلید کاندید نمی‌باشد.

قانون سوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، برخی از ستون‌ها را تولید کند، بدین معنی است که، جدول موردنظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید

جایگاهی نخواهند داشت.

با کمی دقت و بررسی، پُر واضح است که اگر صفت A در کنار عضو کلید کاندید F قرار بگیرد، همکاری صفات $(A$ و $F)$ می‌تواند، همه ستون‌ها را تولید کند، بنابراین صفات AF کلید کاندید جدول R خواهد بود. بستار صفات AF به صورت زیر است:

$$\{AF\}^+ = \{A, F, B, C, D, E\}$$

براساس بستار فوق، صفات AF ، همه ستون‌ها را تولید می‌کنند، پس صفات AF ، کلید کاندید است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد.

همچنین از آن‌جا که $D \rightarrow A$ ، پس می‌توان ترکیب دو خصیصه (D, F) را هم کلید کاندید دیگری برای این جدول تلقی کرد.

چون وقتی (A, F) کلید کاندید است و همه ستون‌ها را تولید می‌کند، پس (D, F) هم کلید کاندید است و همه ستون‌ها را تولید می‌کند، زیرا در نهایت طبق وابستگی $D \rightarrow A$ ، صفت D ، صفت A را می‌دهد و (D, F) به (A, F) تبدیل می‌گردد.

بستار صفات (D, F) به صورت زیر است:

$$\{D, F\}^+ = \{D, F, A, B, C, E\}$$

براساس بستار فوق، صفات (D, F) ، همه ستون‌ها را تولید می‌کنند، پس صفات (D, F) کلید کاندید است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد.

همچنین از آن‌جا که $B \rightarrow D$ ، پس می‌توان ترکیب دو خصیصه (B, F) را هم کلید کاندید دیگری برای این جدول تلقی کرد.

چون وقتی (D, F) کلید کاندید است و همه ستون‌ها را تولید می‌کند، پس (B, F) هم کلید کاندید است و همه ستون‌ها را تولید می‌کند، زیرا در نهایت طبق وابستگی $B \rightarrow D$ ، صفت B ، صفت D را می‌دهد و (B, F) به (D, F) تبدیل می‌گردد.

بستار صفات (B, F) به صورت زیر است:

$$\{B, F\}^+ = \{B, F, D, A, C, E\}$$

براساس بستار فوق، صفات (B, F) ، همه ستون‌ها را تولید می‌کنند، پس صفات (B, F) کلید کاندید است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده، در مجموع، دارای سه کلید کاندید است که مطابق قانون سوم ارسطو، عضو کلید کاندید F در بین هر سه کلید کاندید به طور مشترک قرار دارد.

۳۶- گزینه (۲) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E)$ داریم:

$$A \rightarrow BC$$

$$CD \rightarrow E$$

$$B \rightarrow D$$

$$E \rightarrow A$$

$$ABCDE - ABCDE = \text{تهی}$$

قانون چهارم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، تهی گردد، بدین معنی است که، جدول فوق چندین کلید کاندید دارد، که هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید آن وجود ندارد. بنابراین باید کلید کاندید با بررسی دقیق بر روی مجموعه وابستگی کشف گردد.

$$\{B\}^+ = \{B, D\}$$

براساس بستار فوق، صفت B، فقط ستون‌های B و D را تولید می‌کند، پس صفت B، کلید کاندید نمی‌باشد. بنابراین گزینه دوم پاسخ سوال خواهد بود.

$$\{A\}^+ = \{A, B, C, D, E\}$$

براساس بستار فوق، صفت A، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفت A، کلید کاندید می‌باشد.

در ادامه به شکل بازگشتی جهت کشف مابقی کلیدهای کاندید داریم:

صفت E، ستون A را تولید می‌کند. پس صفت E کلید کاندید است، به صورت زیر:

$$\{E\}^+ = \{E, A, B, C, D\}$$

براساس بستار فوق، صفت E، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفت E، کلید کاندید می‌باشد.

صفات CD، ستون E را تولید می‌کند. پس صفات CD کلید کاندید است، به صورت زیر:

$$\{CD\}^+ = \{C, D, E, A, B\}$$

براساس بستار فوق، صفات CD، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفات CD، کلید کاندید می‌باشد.

صفت B، ستون D را تولید می‌کند. پس صفات CB کلید کاندید است، به صورت زیر:

$$\{CB\}^+ = \{C, B, D, E, A\}$$

بر اساس بستار فوق، صفات CB، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفات CB، کلید کاندید می‌باشد.

توجه: همان‌طور که مشاهده می‌شود، مطابق قانون چهارم ارسطو، هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید فوق وجود ندارد.

۳۷- گزینه (۳) صحیح است.

مجموعه‌ی پوششی A (یعنی مجموعه‌ی تمام صفت‌هایی که با A قابل دسترسی اند) عبارت است از:

$$\{A\}^* = \{A, B, C, D\}$$

ملاحظه می‌شود که فقط صفت E در مجموعه‌ی فوق نیست پس A و E به همراه هم کلیدند.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E)$ داریم:

$$A \rightarrow B$$

$$AB \rightarrow CD$$

$$D \rightarrow ABC$$

$$ABCDE - ABCD = E$$

بنابر رابطه فوق صفت E حتماً باید عضو کلید کاندید باشد. بستار صفت E به صورت زیر است:

$$\{E\}^+ = \{E\}$$

بر اساس بستار فوق، صفت E، فقط ستون E را تولید می‌کند، پس صفت E فقط عضو کلید کاندید می‌باشد و کلید کاندید نمی‌باشد.

قانون سوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، برخی از ستون‌ها را تولید کند، بدین معنی است که، جدول موردنظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت.

براساس بستار فوق، صفت E به عنوان عضو کلید کاندید همه ستون‌ها را تولید نمی‌کند، بنابراین مطابق قانون سوم ارسطو باید صفاتی در کنار صفت E قرار گیرد تا کلید کاندید تولید گردد. این صفات کناری از صفات باقی مانده به غیر از صفات E انتخاب می‌گردند، صفات باقی مانده عبارتند از D و C و B و A البته از این مجموعه صفات C و B را هم کنار می‌گذاریم، زیرا کمکی در تولید صفات دیگر نمی‌کنند. چون در سمت چپ هیچ یک از وابستگی‌های تابعی به تنهایی نیامده‌اند. بنابراین با ترکیب صفات A یا D با صفت E کلیدهای کاندید تولید می‌گردند.

$$\{AE\}^+ = \{A, E, B, C, D\}$$

$$\{DE\}^+ = \{D, E, A, B, C\}$$

پس ترکیبات صفات AE و DE کلیدهای کاندید جدول R هستند.

به بیان دیگر با کمی دقت و بررسی، پُر واضح است که اگر صفت A در کنار عضو کلید کاندید E قرار بگیرد، همکاری صفات (A و E) می‌تواند، همه ستون‌ها را تولید کند، بنابراین صفات AE کلید کاندید جدول R خواهد بود.

بستار صفات AE به صورت زیر است:

$$\{AE\}^+ = \{A, E, B, C, D\}$$

براساس بستار فوق، صفات AE، همه ستون‌ها را تولید می‌کنند، پس صفات AE، کلید کاندید است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد.

همچنین از آن‌جا که $D \rightarrow A$ ، پس می‌توان ترکیب دو خصیصه (D, E) را هم کلید کاندید دیگری برای این جدول تلقی کرد.

چون وقتی (A, E) کلید کاندید است و همه ستون‌ها را تولید می‌کند، پس (D, E) هم کلید کاندید است و همه ستون‌ها را تولید می‌کند، زیرا در نهایت طبق وابستگی $D \rightarrow A$ ، صفت D، صفت A را می‌دهد و (D, E) به (A, E) تبدیل می‌گردد.

بستار صفات (D, E) به صورت زیر است:

$$\{DE\}^+ = \{D, E, A, B, C\}$$

براساس بستار فوق، صفات (D, E)، همه ستون‌ها را تولید می‌کنند، پس صفات (D, E) کلید کاندید است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده، در مجموع، دارای دو کلید کاندید است که مطابق قانون سوم ارسطو، عضو کلید کاندید E در بین هر دو کلید کاندید به طور مشترک قرار دارد.

گزینه اول نادرست است زیرا، عضو کلید کاندید E را ندارد.

گزینه دوم نادرست است. زیرا، عضو کلید کاندید E را ندارد.

گزینه سوم درست است. زیرا، صفات AE کلید کاندید جدول R است.

گزینه چهارم نادرست است. زیرا، کلیدهای کاندید فقط و فقط AE و DE هستند. به بیان دیگر بستار BE

$$\{B, E\}^+ = \{B, E\}$$

همه ستون‌ها رو تولید نمی‌کند:

۳۸- گزینه (۲) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E)$ داریم:

$$AB \rightarrow C$$

$$CD \rightarrow E$$

$$DE \rightarrow B$$

$$ABCDE - BCE = AD$$

بنابر رابطه فوق صفات AD حتماً باید عضو کلید کاندید باشد. بستر صفات AD به صورت زیر است:

$$\{AD\}^+ = \{A, D\}$$

براساس بستر فوق، صفات AD ، فقط ستون‌های A و D را تولید می‌کند، پس صفات AD فقط عضو کلید کاندید می‌باشد و کلید کاندید نمی‌باشد.

قانون سوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، برخی از ستون‌ها را تولید کند، بدین معنی است که، جدول موردنظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت.

براساس بستر فوق، صفات AD به عنوان عضو کلید کاندید همه ستون‌ها را تولید نمی‌کند، بنابراین مطابق قانون سوم ارسطو باید صفاتی در کنار صفات AD قرار گیرد تا کلید کاندید تولید گردد. این صفات کناری از صفات باقی مانده به غیر از صفات AD انتخاب می‌گردند، صفات باقی مانده عبارتند از E و C و B که از این مجموعه هیچ صفت دیگری را کنار نمی‌گذاریم، زیرا در تولید صفات دیگر کمک می‌کنند. بنابراین با ترکیب صفات E و C و B با صفات AD کلیدهای کاندید تولید می‌گردند.

$$\{ADE\}^+ = \{A, D, E, B, C\}$$

$$\{ADC\}^+ = \{A, D, C, E, B\}$$

$$\{ADB\}^+ = \{A, D, B, C, E\}$$

پس ترکیبات صفات ADE و ADC و ADB کلیدهای کاندید جدول R هستند. به بیان دیگر با کمی دقت و بررسی، پُر واضح است که اگر صفت E در کنار عضو کلید کاندید AD قرار بگیرد، همکاری صفات (A و D و E) می‌تواند، همه ستون‌ها را تولید کند، بنابراین صفات ADE کلید کاندید جدول R خواهد بود. بستار صفات ADE به صورت زیر است:

$$\{ADE\}^+ = \{A, D, E, B, C\}$$

براساس بستار فوق، صفات ADE، همه ستون‌ها را تولید می‌کنند، پس صفات ADE، کلید کاندید است. توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. در ادامه به شکل بازگشتی جهت کشف مابقی کلیدهای کاندید داریم: همچنین از آن‌جا که $CD \rightarrow E$ ، پس می‌توان ترکیب سه خصیصه (A, D, C) را هم کلید کاندید دیگری برای این جدول تلقی کرد. چون وقتی (A, D, E) کلید کاندید است و همه ستون‌ها را تولید می‌کند، پس (A, D, C) هم کلید کاندید است و همه ستون‌ها را تولید می‌کند، زیرا در نهایت طبق وابستگی $CD \rightarrow E$ ، صفات CD، صفت E را می‌دهد و (A, D, C) به (A, D, E) تبدیل می‌گردد. بستار صفات (A, D, C) به صورت زیر است:

$$\{ADC\}^+ = \{A, D, C, E, B\}$$

براساس بستار فوق، صفات (A, D, C)، همه ستون‌ها را تولید می‌کنند، پس صفات (A, D, C) کلید کاندید است. توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. همچنین از آن‌جا که $AB \rightarrow C$ ، پس می‌توان ترکیب سه خصیصه (A, D, B) را هم کلید کاندید دیگری برای این جدول تلقی کرد. چون وقتی (A, D, C) کلید کاندید است و همه ستون‌ها را تولید می‌کند، پس (A, D, B) هم کلید کاندید است و همه ستون‌ها را تولید می‌کند، زیرا در نهایت طبق وابستگی $AB \rightarrow C$ ، صفات AB، صفت C را می‌دهد و (A, D, B) به (A, D, C) تبدیل می‌گردد. بستار صفات (A, D, B) به صورت زیر است:

$$\{ADB\}^+ = \{A, D, B, C, E\}$$

براساس بستار فوق، صفات (A, D, B)، همه ستون‌ها را تولید می‌کنند، پس صفات (A, D, B) کلید کاندید است. توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده، در مجموع، دارای سه کلید کاندید است که مطابق قانون سوم ارسطو، عضو کلید کاندید AD در بین هر سه کلید کاندید به طور مشترک قرار دارد.

ترکیب اول: هیچ یا ترکیبی از صفات باقی مانده + کلید کاندید ADE = ابرکلید

ADE+	BC	ترکیبات صفات باقی مانده	ابرکلیدها
	00	تهی	ADE
	01	C	ADEC
	10	B	ADEB
	11	BC	ADEBC

توجه: واضح است که 4 ابرکلید، ایجاد می‌گردد. (2^2)

ترکیب دوم: هیچ یا ترکیبی از صفات باقیمانده + کلید کاندید ADC = ابرکلید

ADC+	BE	ترکیبات صفات باقی مانده	ابرکلیدها
	00	تهی	ADC
	01	D	ADCE
	10	E	ADCB
	11	ED	ADCE

توجه: واضح است که 4 ابرکلید، ایجاد می‌گردد. (2^2)

ترکیب سوم: هیچ یا ترکیبی از صفات باقیمانده + کلید کاندید ADB = ابرکلید

ADB+	CE	ترکیبات صفات باقی مانده	ابرکلیدها
	00	تهی	ADB
	01	E	ADBE
	10	C	ADBC
	11	CE	ADBCE

توجه: واضح است که 4 ابرکلید، ایجاد می‌گردد. (2^2)

بنابراین حاصل جمع ابرکلیدها 12 عدد خواهد بود، که از این مجموعه حاصل، ابرکلیدهای ADCE و ADCBE و ADBE و ADBC و ADBCE در سه مجموعه تکراری هستند، بنابراین با کنار گذاشتن ابرکلیدهای تکراری، در نهایت 7 ابرکلید خواهیم داشت. به همین سادگی.

۳۹- گزینه (۳) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}] = \text{عضو کلید کاندید}$$

توجه: عبارت $[x_i \text{ (چپ)} - y_i \text{ (راست)}]$ به طور مستقل بر روی تک تک وابستگی‌ها انجام می‌گردد.

$$A \rightarrow BC \Rightarrow BC - A = BC$$

مثال:

توجه: استفاده از روش اول مستلزم گام ابتدایی حذف وابستگی‌های بدیهی است، اگر در حذف وابستگی‌های بدیهی دچار خطا می‌شوید، از روش دوم استفاده نمایید.
با توجه به وابستگی‌های مطرح شده برای رابطه $R(A,B,C,D,E)$ داریم:

$$AB \rightarrow C$$

$$CD \rightarrow E$$

$$ABCDE - CE = ABD$$

بنابر رابطه فوق صفات ABD حتماً باید عضو کلید کاندید باشد. بستار صفات ABD به صورت زیر است:

$$\{ABD\}^+ = \{A, B, D, C, E\}$$

بر اساس بستار فوق، صفات ABD ، همه ستون‌ها را تولید می‌کند، پس صفات ABD کلید کاندید می‌باشد.

قانون دوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم) همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

ساخت ابرکلید: هیچ یا ترکیبی از صفات باقیمانده + کلید کاندید ABD = ابرکلید

$ABD +$	CE	ترکیبات صفات باقی مانده	ابرکلیدها
00		تهی	ABD
01		E	ABDE
10		C	ABDC
11		CE	ABDCE

توجه: واضح است که 4 ابرکلید، ایجاد می‌گردد. (2^2)

بنابراین حاصل جمع ابرکلیدها 4 عدد خواهد بود، ابرکلیدهای ABD ، $ABDE$ ، $ABDC$ و $ABDCE$ ، به همین سادگی.

۴۰- گزینه (۴) صحیح است.

برای درک سوال و راه حل آن به مثال زیر دقت نمایید:

مثال: با توجه به رابطه $R(A,B,C,D,E)$ و مجموعه وابستگی‌های تابعی زیر، کدام مورد نادرست است؟

$$F = \{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$$

(۱) $\{E\}$ و $\{C, B\}$ هر دو کلید کاندید هستند. (۲) $\{B\}$ و $\{C, D\}$ هر دو کلید کاندید هستند.

(۳) $\{C, D\}$ و $\{E\}$ هر دو کلید کاندید هستند. (۴) $\{A\}$ و $\{E\}$ هر دو کلید کاندید هستند.

پاسخ: گزینه (۲) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

توجه: عبارت $[x_i \text{ (چپ)} - y_i \text{ (راست)}]$ به طور مستقل بر روی تک تک وابستگی‌ها انجام می‌گردد.

$$A \rightarrow BC \Rightarrow BC - A = BC$$

مثال:

توجه: استفاده از روش اول مستلزم گام ابتدایی حذف وابستگی‌های بدیهی است، اگر در حذف

وابستگی‌های بدیهی دچار خطا می‌شوید، از روش دوم استفاده نمایید.

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E)$ داریم:

$$A \rightarrow BC$$

$$CD \rightarrow E$$

$$B \rightarrow D$$

$$E \rightarrow A$$

$$ABCDE - ABCDE = \text{تهی}$$

قانون چهارم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، تهی گردد، بدین معنی است که،

جدول فوق چندین کلید کاندید دارد، که هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید آن

وجود ندارد. بنابراین باید کلید کاندید با بررسی دقیق بر روی مجموعه وابستگی کشف گردد.

$$\{B\}^+ = \{B, D\}$$

بر اساس بستار فوق، صفت B، فقط ستون‌های B و D را تولید می‌کند، پس صفت B، کلید کاندید نمی‌باشد.

بنابراین گزینه دوم پاسخ سوال خواهد بود.

$$\{A\}^+ = \{A, B, C, D, E\}$$

بر اساس بستار فوق، صفت A، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفت A، کلید کاندید

می‌باشد.

در ادامه به شکل بازگشتی جهت کشف مابقی کلیدهای کاندید داریم:

صفت E، ستون A را تولید می‌کند. پس صفت E کلید کاندید است، به صورت زیر:

$$\{E\}^+ = \{E, A, B, C, D\}$$

براساس بستار فوق، صفت E، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفت E، کلید کاندید می‌باشد.

صفات CD، ستون E را تولید می‌کند. پس صفات CD کلید کاندید است، به صورت زیر:

$$\{CD\}^+ = \{C, D, E, A, B\}$$

براساس بستار فوق، صفات CD، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفات CD، کلید کاندید می‌باشد.

صفت B، ستون D را تولید می‌کند. پس صفات CB کلید کاندید است، به صورت زیر:

$$\{CB\}^+ = \{C, B, D, E, A\}$$

براساس بستار فوق، صفات CB، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفات CB، کلید کاندید می‌باشد.

توجه: همان‌طور که مشاهده می‌شود، مطابق قانون چهارم ارسطو، هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید فوق وجود ندارد.

توجه: پس صورت سوال مطرح شده در شرایط قانون چهارم ارسطو قرار دارد، زیرا اول اینکه بیشتر از یک کلید کاندید دارد و دوم اینکه هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید آن وجود ندارد. ترکیب اول: هیچ یا ترکیبی از صفات باقی مانده + کلید کاندید A = ابرکلید

A +	BCDEF	ترکیبات صفات باقی مانده	ابرکلیدها
	00000	تهی	A
	00001	F	AF
	00010	E	AE
	00011	EF	AEF

	11111	BCDEF	ABCDEF

توجه: واضح است که 32 ابرکلید، ایجاد می‌گردد. (2^5)

ترکیب دوم: هیچ یا ترکیبی از صفات باقیمانده + کلید کاندید BC = ابرکلید

BC +	ADEF	ترکیبات صفات باقی مانده	ابرکلیدها
	0000	تهی	BC
	0001	F	BCF
	0010	E	BCE
	0011	EF	BCEF

	1111	ADEF	BCADEF

توجه: واضح است که 16 ابرکلید، ایجاد می‌گردد. (2^4)
 بنابراین حاصل جمع ابرکلیدها $32+16=48$ عدد خواهد بود، که از این مجموعه حاصل، ابرکلیدهای زیر در دو مجموعه فوق تکراری هستند:

ترکیب تکراری‌ها: هیچ یا ترکیبی از صفات باقیمانده + ابرکلید ABC = ابرکلید
توجه: ترکیب دو کلید کاندید یعنی کلید کاندید A و کلید کاندید BC باهم می‌شود، ابرکلید.

ABC +	DEF	ترکیبات صفات باقی مانده	ابرکلیدها
	000	تهی	ABC
	001	F	ABC F
	010	E	ABCE
	011	EF	ABCEF
	100	D	ABCD
	101	DF	ABCDF
	110	DE	ABCDE
	111	DEF	ABCDEF

توجه: واضح است که 8 ابرکلید تکراری، میان دو مجموعه ترکیب اول و دوم وجود دارد. (2^3)
توجه: بنابراین با کنار گذاشتن ابرکلیدهای تکراری، در نهایت 40 ابرکلید خواهیم داشت. به همین سادگی.

۴۱- گزینه (۱) صحیح است.

وابستگی‌های مطرح شده به صورت زیر است:

- $A \rightarrow B$
- $A \rightarrow C$
- $CD \rightarrow E$
- $B \rightarrow D$
- $E \rightarrow A$

گزینه اول پاسخ سوال است، زیرا:

{سؤال: آیا می‌توان از صفات BD با استفاده از وابستگی‌های فوق به صفات CD رسید؟}

$$\{BD \rightarrow CD\}^+ = \{BD\}^+ = \{B, D\}$$

پاسخ: خیر نمی‌توان رسید، زیرا، صفات BD، B و D را می‌دهد و کار تمام می‌شود. بنابراین پُر واضح است که گزینه اول پاسخ سوال است.

گزینه دوم پاسخ سوال نیست، زیرا:

{سؤال: آیا می‌توان از صفات CD با استفاده از وابستگی‌های فوق به صفات AC رسید؟}

$$\{CD \rightarrow AC\}^+ = \{CD\}^+ = \{\overline{C}, D, E, \overline{A}, B\}$$

پاسخ: بله می‌توان رسید، زیرا، صفات CD، C و D را می‌دهد. CD، E را می‌دهد و E، A را می‌دهد و A را

داریم پس B را هم می دهد.

گزینه سوم پاسخ سوال نیست، زیرا:

{سؤال: آیا می توان از صفات BC با استفاده از وابستگی های فوق به صفات CD رسید؟}

$$\{BC \rightarrow CD\}^+ = \{BC\}^+ = \{B, \boxed{C}, \boxed{D}, E, A\}$$

پاسخ: بله می توان رسید، زیرا، صفات BC، B و C را می دهد. B، D را می دهد و CD، E را می دهد و E را

داریم پس A را هم می دهد.

گزینه چهارم پاسخ سوال نیست. زیرا:

{سؤال: آیا می توان از صفات AC با استفاده از وابستگی های فوق به صفات BC رسید؟}

$$\{AC \rightarrow BC\}^+ = \{AC\}^+ = \{A, \boxed{C}, \boxed{B}, D, E\}$$

پاسخ: بله می توان رسید، زیرا AC، A و C را تولید می کند. A، B را می دهد، B، D را می دهد و CD را

داریم، پس E را هم می دهد.

توجه: سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه اول را به عنوان پاسخ اعلام کرده

بود.

۴۲- گزینه (۲) صحیح است.

با توجه به وابستگی های مطرح شده برای رابطه $R(A, B, C, D, E)$ داریم:

$$AB \rightarrow CDE$$

$$E \rightarrow BC$$

$$ABCDE - BCDE = A$$

بنابر رابطه فوق صفت A حتماً باید عضو کلید کاندید باشد. بستر صفات A به صورت زیر است:

$$\{A\}^+ = \{A\}$$

براساس بستر فوق، صفت A، فقط ستون A را تولید می کند، پس صفت A فقط عضو کلید کاندید می باشد و کلید کاندید نمی باشد.

قانون سوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، برخی از ستون ها را تولید کند، بدین معنی است که، جدول مورد نظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت.

براساس بستر فوق، صفت A به عنوان عضو کلید کاندید همه ستون ها را تولید نمی کند، بنابراین مطابق قانون سوم ارسطو باید صفاتی در کنار صفت A قرار گیرد تا کلید کاندید تولید گردد. این صفات کناری از صفات باقی مانده انتخاب می گردند، صفات باقی مانده عبارتند از B و C و D و E البته از این مجموعه صفات C و D را هم کنار می گذاریم، زیرا کمکی در تولید صفات دیگر نمی کنند. چون در سمت چپ هیچ

یک از وابستگی‌های تابعی نیامده‌اند. بنابراین با ترکیب صفات B یا E با صفت A کلیدهای تولید می‌گردند.

$$\{AB\}^+ = \{A, B, C, D, E\}$$

$$\{AE\}^+ = \{A, E, B, C, D\}$$

پس ترکیبات صفات AB و AE کلیدهای جدول R هستند. بستار صفات AB به صورت زیر است:

$$\{AB\}^+ = \{A, B, C, D, E\}$$

بر اساس بستار فوق، صفات AB، همه ستون‌ها را تولید می‌کنند، پس صفات AB، کلید کلید است. همچنین از آنجا که مطابق وابستگی‌ها $E \rightarrow B$ ، پس می‌توان ترکیب دو خصیصه (A, E) را هم کلید کلید دیگری برای این جدول تلقی کرد. چون وقتی (A, B) کلید کلید است و همه ستون‌ها را تولید می‌کند، پس (A, E) هم کلید کلید است و همه ستون‌ها را تولید می‌کند، زیرا در نهایت طبق وابستگی $E \rightarrow B$ ، صفت E، صفت B را می‌دهد و (A, E) به (A, B) تبدیل می‌گردد. بستار صفات (A, E) به صورت زیر است:

$$\{AE\}^+ = \{A, E, B, C, D\}$$

بر اساس بستار فوق، صفات (A, E)، همه ستون‌ها را تولید می‌کنند، پس صفات (A, E) کلید کلید است. توجه: دقت کنید که هیچ‌گاه، کلید کلید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده، در مجموع، دارای دو کلید کلید است که مطابق قانون سوم ارسطو، عضو کلید کلید A در بین هر دو کلید کلید به طور مشترک قرار دارد.

۴۳- گزینه (۴) صحیح است.

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E, F)$ داریم:

$$A \rightarrow F$$

$$B \rightarrow C$$

$$ABC \rightarrow E$$

$$ABCDEF - CEF = ABD$$

بنابر رابطه فوق صفات ABD حتماً باید عضو کلید کلید باشد. بستار صفات ABD به صورت زیر است:

$$\{ABD\}^+ = \{A, B, D, F, C, E\}$$

بر اساس بستار فوق، صفات ABD، همه ستون‌ها را تولید می‌کند، پس صفات ABD کلید کلید می‌باشد.

قانون دوم ارسطو

هرگاه عضو کلید کلید، حاصل از تفاضل قانون اول (روش اول یا دوم) همه ستون‌ها را تولید کند، آن عضو کلید کلید، تنها کلید کلید جدول خواهد بود.

ساخت ابرکلید: هیچ یا ترکیبی از صفات باقیمانده + کلید کلید ABD = ابرکلید

ABD +	CEF	ترکیبات صفات باقی مانده	ابرکلیدها
	000	تهی	ABD
	001	F	ABDF
	010	E	ABDE
	011	EF	ABDEF
	100	C	ABDC
	101	CF	ABDCF
	110	CE	ABDCE
	111	CEF	ABDCEF

توجه: واضح است که 8 ابرکلید، ایجاد می‌گردد. (2^3)

بنابراین حاصل جمع ابرکلیدها 8 عدد خواهد بود، به صورت زیر:

ابرکلیدهای ABD، ABDF، ABDE، ABDEF، ABDC، ABDCF، ABDCE و ABDCEF

۴۴- گزینه (۳) صحیح است.

قانون چهارم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، تهی گردد، بدین معنی است که، جدول فوق چندین کلید کاندید دارد، که هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید آن وجود ندارد. بنابراین باید کلید کاندید با بررسی دقیق بر روی مجموعه وابستگی کشف گردد.

توجه: مطابق فرض سوال، بین خصیصه‌های دو کلید کاندید همپوشانی وجود ندارد. بنابراین صورت سوال مطرح شده در شرایط قانون چهارم ارسطو قرار دارد، زیرا اول اینکه بیشتر از یک کلید کاندید دارد و دوم اینکه هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید آن وجود ندارد.

توجه: فرض کنید کلید کاندید تک خصیصه برابر A و کلید کاندید دو خصیصه برابر BC باشد.

توجه: در یک جدول با $N \geq 3$ خصیصه داریم:

ترکیب اول: هیچ یا ترکیبی از صفات باقی مانده + کلید کاندید A = ابرکلید

A +	BCDEF...	ترکیبات صفات باقی مانده	ابرکلیدها
	00000...	تهی	A

	11111...	BCDEF...	ABCDEF...

توجه: واضح است که در این حالت (2^{n-1}) ابرکلید، ایجاد می‌گردد.

ترکیب دوم: هیچ یا ترکیبی از صفات باقیمانده + کلید کاندید BC = ابرکلید

BC +	ADEF...	ترکیبات صفات باقی مانده	ابرکلیدها
	0000...	تهی	BC

	1111...	ADEF...	BCADEF...

توجه: واضح است که در این حالت (2^{n-2}) ابرکلید، ایجاد می‌گردد.

بنابراین حاصل جمع ابرکلیدها $\left[(2^{n-1}) + (2^{n-2}) \right]$ عدد خواهد بود، که از این مجموعه حاصل،

ابرکلیدهای زیر در دو مجموعه فوق تکراری هستند:

ترکیب تکراری‌ها: هیچ یا ترکیبی از صفات باقیمانده + ابرکلید ABC = ابرکلید
توجه: ترکیب دو کلید کاندید یعنی کلید کاندید A و کلید کاندید BC باهم می‌شود، ابرکلید.

ABC +	DEF...	ترکیبات صفات باقی مانده	ابرکلیدها
	0 0 0 ...	تهی	ABC

	1 1 1 ...	DEF...	ABCDEF...

توجه: واضح است که (2^{n-3}) ابرکلید تکراری، میان دو مجموعه ترکیب اول و دوم وجود دارد.
توجه: بنابراین با کنار گذاشتن ابرکلیدهای تکراری، در نهایت $5 \times (2^{n-3})$ ابرکلید خواهیم داشت، به صورت زیر:

$$\left[(2^{n-1}) + (2^{n-2}) \right] - \left[(2^{n-3}) \right] = 2^{n-3} (2^2 + 2^1 - 1) = 5 \times (2^{n-3})$$

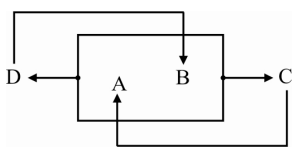
تست‌های فصل نهم: نرمال‌سازی

۱- کدام یک از گزاره‌های زیر در رابطه با «عمده‌ترین اهداف نرمال‌سازی رابطه‌ها» نادرست است؟
(مهندسی کامپیوتر- دولتی ۷۶)

- ۱) کاهش بعضی انواع افزونگی (Redundancy).
- ۲) کاهش سربار (Overhead) سیستم در پاسخگویی در عمل بازیابی.
- ۳) تسهیل در اعمال برخی محدودیت‌های جامعیتی (Integrity Constraints).
- ۴) اجتناب از بعضی از انواع آنومالی در عملیات روی پایگاه.

۲- کدام گزینه از اهداف کلی فرآیند نرمال‌سازی (normalization) نیست؟ (مهندسی کامپیوتر- دولتی ۷۵)

- ۱) تسهیل اعمال بعضی از قواعد جامعیتی (integrity rules)
- ۲) تسهیل پیاده‌سازی دید کاربر (user view)
- ۳) حذف بعضی از انواع قواعد وابستگی تابعی (functional dependency)
- ۴) کاهش بعضی از انواع افزونگی (redundancy)



۳- اگر رابطه $R(A, B, C, D)$ با توابع وابستگی شکل مقابل مفروض باشد، کدام یک از گزینه‌ها نادرست است؟

(مهندسی کامپیوتر- دولتی ۸۳)

- ۱) (A, B) و (C, D) کلید کاندید رابطه R می‌باشند.
- ۲) رابطه R نرمال فرم BCNF می‌باشد، اما نرمال فرم 3NF نمی‌باشد.
- ۳) رابطه R نرمال فرم 2NF و نیز نرمال فرم 3NF می‌باشد.
- ۴) رابطه R نرمال فرم 3NF می‌باشد اما نرمال فرم BCNF نمی‌باشد.

۴- کدام گزینه صحیح است؟ (مهندسی IT- دولتی ۸۳)

- ۱) جدولی که BCNF باشد، حتماً 3NF است.
- ۲) اگر جدولی 3NF باشد، وابستگی انتقالی ندارد.
- ۳) اگر جدولی وابستگی انتقالی نداشته باشد، 3NF است.
- ۴) ۱ و ۲.

۵- جدول $R(X, Y, Z)$ به دو جدول $R_1(X, Y)$ و $R_2(X, Z)$ تجزیه و نرمال شده است. کدام گزینه از شرط‌های صحت این عمل می‌باشد؟ ($X \rightarrow R_1$ یعنی X کلید R_1 است). (مهندسی کامپیوتر- دولتی ۸۴)

- ۱) $X \rightarrow R_1$
- ۲) $X \rightarrow R_2$
- ۳) $X \rightarrow R_1$ و $X \rightarrow R_2$
- ۴) $X \rightarrow R_1$ یا $X \rightarrow R_2$

۶- رابطه XYZ به شکل روبه‌رو را در نظر بگیرید. این رابطه در چه سطحی از نرمال بودن است؟ (مهندسی IT- دولتی ۸۵)

X	Y	Z
x1	y1	z2
x1	y2	z1
x2	y1	z1
x1	y1	z1

- ۱) 1NF
- ۲) 2NF
- ۳) 3NF
- ۴) BCNF

۷- با نرمال‌سازی رابطه R تا سطح 3NF تعداد جداول حاصل چند عدد می‌شود؟ مجموعه وابستگی‌ها در FD نمایش داده شده است. (مهندسی IT- آزاد ۸۵)

$R(A, B, C, D, E, F(G, H), I)$

$FD = \{BC \rightarrow A, B \rightarrow G, BCD \rightarrow I, G \rightarrow H, E \rightarrow I, BCD \rightarrow E, BCD \rightarrow A\}$

4 (۱) 6 (۲) 3 (۳) 5 (۴)

۸- اگر در رابطه $R(A, B, C, D, E, F)$ کلید اصلی BCD باشد، آن‌گاه کدام وابستگی تابعی زیر شرط 2NF بودن رابطه R را نقض می‌کند؟ (مهندسی IT- آزاد ۸۶)

$BC \rightarrow E$ (۱) $DCB \rightarrow A$ (۲) $E \rightarrow F$ (۳) $A \rightarrow EF$ (۴)

۹- رابطه $R(A, B, C, D, E, F, G)$ با کلید اصلی BCD و با مجموعه وابستگی تابعی زیر مفروض است:

$FD = \{BCD \rightarrow E, BCD \rightarrow B, BCD \rightarrow F, BD \rightarrow A, F \rightarrow G, E \rightarrow F, C \rightarrow E\}$

در صورت تبدیل آن به 3NF چند رابطه ایجاد می‌شود؟

6 (۱) 4 (۲) 3 (۳) 5 (۴)

۱۰- اگر وابستگی چند مقداری به شکل $X \twoheadrightarrow Y$ برقرار باشد، و هم‌چنین داشته باشیم $V \subseteq W$ آن‌گاه طبق قانون افزایشی (augmentation) می‌توان درستی کدام گزینه زیر را نتیجه گرفت؟ (مهندسی IT- دولتی ۸۷)

$VX \twoheadrightarrow WY$ (۱) $VX \rightarrow VY$ (۲) $WX \twoheadrightarrow VY$ (۳) $WX \twoheadrightarrow WY$ (۴)

۱۱- در یک پایگاه داده محیط آموزشی دانشگاه برای جدول دانشجو سه صفت شماره دانشجویی، کد پستی و شهر محل سکونت با وابستگی‌های تابعی زیر وجود دارند:

{شهر محل سکونت \rightarrow کد پستی محل سکونت \rightarrow شماره دانشجویی}

با توجه به این که رابطه شهرها و کدپستی ثابت است، سطح نرمال مناسب برای تجزیه این جدول چیست؟

1NF (۱) 2NF (۲) 3NF (۳) BCNF (۴)

۱۲- مجموعه وابستگی‌های تابعی (FD) روبه‌رو را در نظر بگیرید (هر حرف یک صفت است). برای این مجموعه پایگاه داده رابطه‌ای با سه جدول AIE، ABCDH و EFG طراحی شده است. این پایگاه داده در چه سطح نرمالی قرار دارد؟ (مهندسی IT- دولتی ۸۷)

$A \rightarrow IEG$ 1NF (۱)

$AB \rightarrow IECD$ 2NF (۲)

$ABE \rightarrow CDFGH$ 3NF (۳)

$AC \rightarrow IEC$ BCNF (۴)

$CDE \rightarrow DE$

$E \rightarrow FG$

$F \rightarrow E$

۱۳- در طراحی بانک اطلاعاتی رابطه‌ای، کدام یک از معیارها مهم‌تر است؟ (مهندسی IT- دولتی ۸۸)

۱) رعایت شرط Nonloss Join ۲) داشتن کلید اصلی در رابطه‌ها

۳) حفظ وابستگی‌های تابعی ۴) کاهش تعداد خصیصه‌ها در رابطه‌ها

- ۱۴- کدام مورد از خصوصیات شکل نرمال اول نیست؟ (مهندسی IT- دولتی ۸۸)
- (۱) هیچ رابطه تو در تو (Nested) وجود ندارد.
 - (۲) هیچ صفت مرکب (Composite Attribute) وجود ندارد.
 - (۳) هیچ صفت چند مقداری (Multivalued Attribute) وجود ندارد.
 - (۴) هیچ وابستگی تابعی تراگذاری (Transitive Functional Dependency) وجود ندارد.

- ۱۵- کدام گزینه در خصوص نرمال سازی نادرست است؟ (مهندسی کامپیوتر- آزاد ۸۸)
- (۱) نرمال سازی باعث افزایش سرعت پاسخ گویی به پرس و جو می شود.
 - (۲) نرمال سازی تا سطح 3NF اجباری است.
 - (۳) نرمال سازی ممکن است باعث افزایش افزونگی شود.
 - (۴) حل مشکل مقادیر پوچ (NULL) یکی از اهداف نرمال سازی است.

- ۱۶- با توجه به عبارات زیر، کدام گزینه صحیح است؟ (مهندسی IT- آزاد ۸۸)
- (۱) هر رابطه ای 2NF باشد، حتماً 1NF نیز هست.
 - (۲) هر رابطه ای 1NF باشد، حتماً 2NF نیز هست.
 - (۳) هر رابطه ای BCNF باشد، حتماً 3NF نیز هست.
 - (۴) هر رابطه ای 3NF باشد، حتماً BCNF نیز هست.
- (۴) 1, 2, 3 و 4 (۳) 1 و 4 (۲) 1 و 3 (۱) 2 و 3

- ۱۷- در رابطه $R(A, B, C, D, E)$ با مجموعه وابستگی های S کدام یک از گزینه های زیر درست است؟ (مهندسی کامپیوتر- آزاد ۸۸)
- $$S = \{A \rightarrow C, B \rightarrow D, AB \rightarrow E\}$$
- (۱) رابطه R نرمال 2NF می باشد.
 - (۲) رابطه R نرمال 3NF می باشد.
 - (۳) رابطه R نرمال 3NF و BCNF می باشد.
 - (۴) اگر رابطه R به روابط $R_1(A, B, E)$ و $R_2(B, D)$ و $R_3(A, C)$ تفکیک شود، هر سه رابطه حاصل نرمال 3NF و BCNF می باشند.

- ۱۸- اگر جدولی در فرم نرمال BCNF باشد، ممکن است در کدام فرم نرمال دیگر نباشد؟ (مهندسی کامپیوتر- آزاد ۸۸)
- (۱) 1NF (۲) 2NF (۳) 3NF (۴) 4NF

- ۱۹- کدام گزینه از اهداف کلی فرآیند نرمال سازی نیست؟ (مهندسی IT- آزاد ۸۸)
- (۱) تسهیل در پیاده سازی دید کاربر
 - (۲) تسهیل اعمال برخی از قوانین جامعیتی
 - (۳) حذف برخی از مقادیر NULL
 - (۴) کاهش برخی از انواع افزونگی

- ۲۰- اگر در جدول ST وابستگی تابعی $Code \rightarrow S\#$ را داشته باشیم، آنگاه این جدول در کدام فرم نرمال نمی باشد؟ (مهندسی IT- دولتی ۸۹)
- ST(S#, T#, Date, Time, Code)
- (۱) 3NF (۲) 2NF (۳) CCNF (۴) BCNF

۲۱- اگر تمامی اطلاعات جداول را در یک جدول بریزیم با چه مشکلاتی روبه‌رو می‌شویم؟

(مهندسی کامپیوتر- آزاد ۸۹ گروه الف)

- (۱) افزونگی داده‌ها (Data Redundancy) (۲) بی‌نظمی (Anomaly)
(۳) مقادیر تهی (NULL Values) (۴) همه موارد

۲۲- کدام گزینه از عمده‌ترین اهداف نرمال‌سازی نیست؟

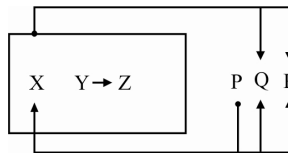
(مهندسی IT- آزاد ۸۹)

- (۱) کاهش بعضی از انواع افزونگی‌ها
(۲) حذف بعضی از انواع وابستگی تابعی
(۳) تسهیل اعمال بعضی از قواعد جامعیتی
(۴) کاهش سربار (Over head) سیستم در پاسخگویی در عمل بازیابی

۲۳- رابطه $A(X, Y, Z, P, Q, R)$ با وابستگی‌های تابعی (Functional Dependency) شکل زیر را در نظر بگیرید.

(مهندسی کامپیوتر- دولتی ۹۰)

این رابطه در کدام سطح نرمال است؟



- (۱) 3NF
(۲) 2NF
(۳) 1NF
(۴) BCNF

۲۴- کدام یک از گزینه‌های زیر نادرست است؟

(مهندسی IT- دولتی ۹۰)

(۱) در SQL، عملگرهای $(\neq \text{all})$ و not in هم ارز هستند.

(۲) رابطه R با وابستگی تابعی F در BCNF است اگر و فقط اگر رابطه R با وابستگی‌های تابعی F^+ (بستار F) در BCNF باشد.

(۳) در رابطه‌ای با اسکیمای $R(A, B, C, D)$ به طور منطقی می‌توان از وجود وابستگی چند مقداری $BC \twoheadrightarrow A$ وابستگی‌های منطقی $A \twoheadrightarrow B$ و $A \twoheadrightarrow C$ را نتیجه‌گیری کرد.

(۴) دو مجموعه وابستگی‌های زیر هم ارز (معادل) نیستند:

$$F_1 = \{AB \rightarrow E, A \rightarrow CD, A \rightarrow E, E \rightarrow D, D \rightarrow A\}$$

$$F_2 = \{AB \rightarrow D, E \rightarrow AD, A \rightarrow CD, D \rightarrow AE, EC \rightarrow B\}$$

۲۵- فرض کنید رابطه R با مجموعه وابستگی‌های تابعی F داده شده است. صفت $A \in R$ را ناهنجار

(abnormal) می‌گوییم اگر و فقط اگر $(\exists x \subseteq R) : x \rightarrow R \notin F^+ \wedge A \in (x^+ - x)$ که نمایانگر بستار

(closure) مجموعه صفات x تحت وابستگی‌های تابعی F می‌باشد. کدام یک از گزینه‌های زیر نادرست

(مهندسی IT- دولتی ۹۰)

است؟

(۱) اگر رابطه R در 3NF باشد، آن‌گاه هیچ صفت غیرکلیدی در رابطه R ناهنجار نخواهد بود.

(۲) اگر رابطه R در BCNF باشد، آن‌گاه هیچ صفت ناهنجاری در R وجود نخواهد داشت.

(۳) اگر هیچ صفت ناهنجاری در R وجود نداشته باشد، آن‌گاه رابطه R در BCNF خواهد بود.

(۴) هیچ کدام

۲۶- رابطه $R(A, B, C, D, E)$ و وابستگی های تابعی زیر را در نظر بگیرید:

$$A \rightarrow B, BC \rightarrow D, E \rightarrow C$$

اگر نگاشت R بر روی $S(B, C, D, E)$ را در نظر بگیریم، کدام یک از وابستگی های تابعی در S برقرار است و شرط BCNF را برای S نقض نمی کند؟

(مهندسی IT - دولتی ۹۰)

$$B \rightarrow E \text{ (۴)} \quad BC \rightarrow D \text{ (۳)} \quad E \rightarrow C \text{ (۲)} \quad BE \rightarrow D \text{ (۱)}$$

۲۷- جدول $R(A, B, C, D, E, F, H, I)$ در طی فرآیند نرمال سازی به جدول $R_1(A, B)$ ، $R_2(B, C, F)$ ، $R_3(D, E)$ ، $R_4(C, F, J, I)$ شکسته شده است.

(مهندسی کامپیوتر - آزاد ۹۰ گروه الف)

این شکستن قطعاً غلط است.

(۲) این کار به شرطی درست است که A کلید اصلی جدول R باشد.

(۳) این کار به شرطی درست است که AB کلید اصلی جدول R باشد.

(۴) بدون داشتن وابستگی های تابعی نمی توان اظهار نظری کرد.

۲۸- کدام گزینه در خصوص نرمال سازی غلط است؟

(مهندسی کامپیوتر - آزاد ۹۰ گروه الف)

(۱) نرمال سازی عموماً به افزایش سرعت پرس و جوها منجر می شود.

(۲) وجود مقادیر پوچ (NULL Values) می تواند فرآیند نرمال سازی را با مشکل مواجه نماید.

(۳) نرمال سازی می تواند باعث کاهش بی نظمی (Anomaly) در پایگاه داده شود.

(۴) نرمال سازی تا سطح 3NF عموماً حجم پایگاه داده را کاهش می دهد.

۲۹- جدول $R(A, B, C, D, E, F)$ با وابستگی های تابعی $F = \{AB \rightarrow CF, C \rightarrow DE, E \rightarrow F\}$ در طی فرآیند

(مهندسی کامپیوتر - آزاد ۹۰ گروه الف)

نرمال سازی به سه جدول R_1 و R_2 و R_3 شکسته شده است.

(۱) در صورتی که وابستگی های تابعی جداول به ترتیب $F_1 = \{AB \rightarrow C\}$ ، $F_2 = \{C \rightarrow DE\}$ و $F_3 = \{E \rightarrow F\}$ باشد، این شکستن درست است.

(۲) در صورتی که وابستگی های تابعی جداول به ترتیب $F_1 = \{AB \rightarrow CF\}$ ، $F_2 = \{C \rightarrow D\}$ و $F_3 = \{C \rightarrow E\}$ باشد، این شکستن درست است.

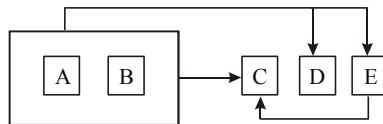
(۳) این شکستن قطعاً غلط است و جدول R باید به چهار جدول شکسته شد.

(۴) این شکستن قطعاً غلط است و جدول R باید به دو جدول شکسته شود.

۳۰- با توجه به نمودار وابستگی های تابعی داده شده رابطه $R(A, B, C, D, E)$ در چه سطحی از نرمال قرار

(مهندسی کامپیوتر - آزاد ۹۰ گروه ب)

دارد؟



(۲) سطح دوم نرمال (2NF)

(۱) سطح سوم نرمال (3NF)

(۴) سطح نرمال (1NF)

(۳) سطح نرمال BCNF

(مهندسی کامپیوتر - آزاد ۹۰ گروه ب)

۳۱- رابطه $R(A, B, C, D)$ و وابستگی های تابعی زیر مفروضند:

$$A \rightarrow B, B \rightarrow C, BC \rightarrow A$$

این رابطه:

- (۱) 3NF هست ولی BCNF نیست. (۲) 3NF نیست.
 (۳) BCNF هست ولی 4NF نیست. (۴) 4NF هست.

۳۲- یک طراحی منطقی خوب و مناسب یک پایگاه داده باید کدام یک از خصوصیات زیر را داشته باشد؟

(مهندسی کامپیوتر- آزاد ۹۰ گروه ب)

- (۱) نمایش جامع و واضحی از محیط دنیای واقعی داشته باشد و در برگیرنده تمام خصوصیات مورد نیاز کاربران، به گونه‌ای که تولید برنامه‌های کاربردی به راحتی امکان‌پذیر باشد.
 (۲) تمام محدودیت‌های (قواعد) جامعیتی، که قابل اعمال در هر مرحله از طراحی جامعیتی باشند، در طراحی منظور شده است.
 (۳) کم‌ترین میزان اختلاط اطلاعاتی را داشته باشد و کم‌ترین مقدار هیچ مقدار (NULL) در پایگاه داده ایجاد گردد.
 (۴) همه موارد

۳۳- کدام گزینه در خصوص نرمال‌سازی صحیح است؟

(مهندسی IT- آزاد ۹۰)

- (۱) نرمال‌سازی نکردن می‌تواند به افزایش سرعت برخی پرس‌وجوها منجر شود.
 (۲) نرمال‌سازی نکردن می‌تواند به کاهش سرعت برخی پرس‌وجوها منجر شود.
 (۳) نرمال‌سازی تا سطح 3NF عموماً باعث کاهش افزونگی می‌شود.
 (۴) هر سه مورد

۳۴- جدول $R(A, B, C, D, E, F)$ با وابستگی‌های تابعی $F = \{AB \rightarrow CD, C \rightarrow DE, E \rightarrow F\}$ در طی فرآیند

(مهندسی IT- آزاد ۹۰)

- نرمال‌سازی به سه جدول R_1 و R_2 و R_3 شکسته شده است.
 (۱) در صورتی که وابستگی‌های تابعی جداول به ترتیب $F_1 = \{AB \rightarrow C\}$ و $F_2 = \{C \rightarrow DE\}$ و $F_3 = \{E \rightarrow F\}$ باشد، این شکستن درست است.
 (۲) در صورتی که وابستگی‌های تابعی جداول به ترتیب $F_1 = \{AB \rightarrow CD\}$ و $F_2 = \{C \rightarrow E\}$ و $F_3 = \{E \rightarrow F\}$ باشد این شکستن درست است.
 (۳) این شکستن قطعاً غلط است و جدول R باید به دو جدول شکسته شود.
 (۴) این شکستن قطعاً غلط است و جدول R باید به چهار جدول شکسته شود.

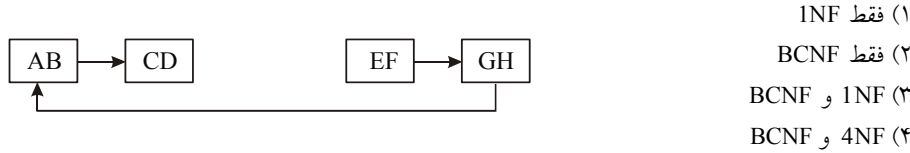
۳۵- جدول $R(A, B, C, D, E, F)$ در طی فرآیند نرمال‌سازی به جداول $R_1(A, B, C)$ و $R_2(B, F)$ و

(مهندسی IT- آزاد ۹۰)

- $R_3(D, E)$ شکسته شده است.
 (۱) این کار به شرطی درست است که A کلید اصلی جدول R باشد.
 (۲) این کار به شرطی درست است که AB کلید اصلی جدول R باشد.
 (۳) این شکستن قطعاً غلط است.
 (۴) بدون داشتن وابستگی‌های تابعی نمی‌توان اظهارنظر کرد.

۳۶- اگر کوتاه‌ترین کلید را کلید اصلی بگیریم آن‌گاه بانک اطلاعات زیر در کدام فرم نرمال است؟

(مهندسی کامپیوتر- دولتی ۹۱)



۳۷- رابطه $R(A, B, C, D, E, G)$ با وابستگی‌های تابعی F را در نظر بگیرید:

$$F = \{A \rightarrow BC, BE \rightarrow G, G \rightarrow CD, AD \rightarrow BG, AE \rightarrow G\}$$

(مهندسی IT- دولتی ۹۱)

کدام یک از گزینه‌های زیر صحیح نیست؟

(۱) صفت B در وابستگی تابعی $AD \rightarrow BG$ اضافی (extraneous) است.

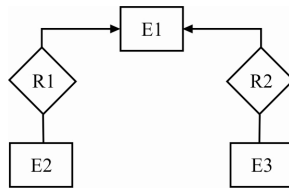
(۲) تنها کلید کاندید رابطه R ، $\{AE\}$ است.

(۳) وابستگی تابعی $AE \rightarrow G$ تکراری (redundant) است.

(۴) رابطه R در نرمال فرم دوم است.

۳۸- فرض کنید که بین سه مجموعه موجودیت $E1$ ، $E2$ و $E3$ مجموعه رابطه‌هایی به شکل زیر برقرار باشد

(مهندسی IT- دولتی ۹۱)



ولی در زمان طراحی دیاگرام ER، طراح به اشتباه یک رابطه سه گانه میان این سه مجموعه موجودیت در نظر گرفته است. حال پس از تبدیل دیاگرام ER به مدل رابطه‌ای، رابطه‌ای (جدولی) برای رابطه سه گانه در نظر گرفته شده است. آیا می‌توان با اعمال نرمال فرم‌ها به طراحی درست رسید؟

(۱) بله، با استفاده از 4NF می‌توان رسید، البته به شرط داشتن وابستگی‌های تابعی و چند مقداری موجود.

(۲) بله، با استفاده از BCNF می‌توان رسید، البته به شرط داشتن وابستگی‌های تابعی موجود.

(۳) خیر، به علت اشتباه اولیه در طراحی دیاگرام ER، بازیابی رابطه‌های درست دو تایی میسر نیست.

(۴) بله، با استفاده از 3NF می‌توان رسید، البته به شرط داشتن وابستگی‌های تابعی موجود.

۳۹- فرض کنید رابطه $R(A1, A2, A3, \dots, An)$ را داشته باشیم. فرض کنید Z ، Y ، X و W زیر مجموعه‌هایی از

(مهندسی IT- دولتی ۹۱)

R باشند. دو گزاره زیر را در نظر بگیرید:

گزاره اول: اگر داشته باشیم $X \rightarrow Y$ و $Z \subseteq W$ آن‌گاه می‌توان به طور منطقی نتیجه گرفت $WX \rightarrow YZ$.

گزاره دوم: اگر $X \rightarrow Y$ وجود داشته باشد و زیر مجموعه W دارای خواص زیر باشد $W \cap Y = \emptyset$ و

$W \rightarrow Z$ و $Z \subseteq Y$ آن‌گاه می‌توان به طور منطقی نتیجه گرفت $X \rightarrow Z$.

(۱) فقط گزاره اول صحیح است.

(۲) فقط گزاره دوم صحیح است.

(۳) هر دو گزاره صحیح هستند.

(۴) هر دو گزاره غلط هستند.

۴۰- رابطه $R = (A, B, C, D, E)$ و وابستگی‌های تابعی (FD) را به صورت ذیل در نظر بگیرید:

$$FD = \{A \rightarrow D, A \rightarrow E, D \rightarrow E, (A, B) \rightarrow D, (A, B) \rightarrow E, (A, B) \rightarrow C\}$$

کدام تجزیه بهتر و حداقل سطح نرمالیتی آن تجزیه کدام است؟ (مهندسی کامپیوتر- آزاد ۹۱)

$$2NF, R''(A, D, E), R'(A, B, C) \quad (۱)$$

$$3NF, R'''(D, E), R''(A, D), R'(A, B, C) \quad (۲)$$

$$3NF, R'''(A, E), R''(A, D), R'(A, B, C) \quad (۳)$$

$$3NF, R'''(A, B, E), R''(A, B, D), R'(A, B, C) \quad (۴)$$

۴۱- رابطه $R(A, B, C, D)$ و مجموعه وابستگی‌های تابعی (Functional Dependencies) F را در نظر بگیرید.

$$F = \{A \rightarrow B, B \rightarrow C, C \rightarrow D\}$$

کدام یک از این تجزیه‌ها بدون گمشدگی (Lossless) نیست؟ (مهندسی کامپیوتر- دولتی ۹۲)

یادآوری: رابطه R و مجموعه وابستگی‌های تابعی F روی R را در نظر بگیرید. تجزیه R به دو رابطه با مجموعه صفات X و Y یک تجزیه «بدون گمشدگی» با توجه به F است. اگر به ازای هر نمونه r از R که با F مطابقت دارد، $\Pi_X(r) \bowtie \Pi_Y(r) = r$ ، به عبارت دیگر رابطه اولیه را می‌توان از روی رابطه‌های جدید به دست آورد.

$$R_1(A, B), R_2(B, C), R_3(C, D) \quad (۱)$$

$$R_1(A, B), R_2(A, C), R_3(A, D) \quad (۲)$$

$$R_1(A, D), R_2(B, D), R_3(C, D) \quad (۳)$$

(۴) هیچ کدام- به عبارت دیگر همه این تجزیه‌ها بدون گمشدگی هستند.

۴۲- کدام یک از گزینه‌های زیر برای رابطه **Booking** و وابستگی‌های تابعی داده شده آن، که اطلاعات رزرو

اتاق در یک هتل را نگهداری می‌کنند، غلط است؟ (مهندسی IT- دولتی ۹۲)

Booking (guestID, guestName, creditCard, roomNo, roomCat, from, to)

$F = \{$

1) guestID \rightarrow guestName, creditCard

2) roomNo \rightarrow roomCat

3) roomNo, from \rightarrow guestID, to

4) roomNo, to \rightarrow guestID, from

$\}$

(۱) این رابطه در هیچ یک از نرمال فرم‌های 3NF, 2NF و BCNF نیست.

(۲) دو وابستگی تابعی سوم و چهارم باعث نقض شدن 3NF در رابطه نمی‌شود.

(۳) اگر این رابطه به سه رابطه R_1 (guestID, roomNo, from, to), R_2 (roomNo, roomCat) و R_3 (creditCard, roomNo, from)

تجزیه شود، هر سه رابطه در 3NF و BCNF هستند و تجزیه بدون

اتلاف (lossless decomposition) است.

(۴) اگر این رابطه به سه رابطه R_1 (guestID, guestName, creditCard), R_2 (roomNo, roomCat) و R_3 (guestID, roomNo, from, to)

تجزیه شود، هر سه رابطه در 3NF هستند ولی در BCNF نیستند.

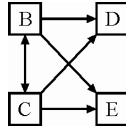
۴۳- رابطه زیر و مجموعه وابستگی‌های تابعی آن را در نظر بگیرید: (مهندسی IT- دولتی ۹۲)

$$R(A, B, C, D, E, F, G, H) \quad F = \{D \rightarrow EF, F \rightarrow C, DG \rightarrow AB\}$$

کدام یک از گزینه‌های زیر غلط است؟

- (۱) این رابطه ۳۲ ابرکلید دارد.
 (۲) تنها کلید کاندید DGH است.
 (۳) رابطه فوق در فرم نرمال دوم است.
 (۴) رابطه فوق در نرمال فرم سوم نیست.

۴۴- نمودار وابستگی تابعی رابطه‌ای به صورت ذیل است. B و C کلید کاندید هستند. این رابطه در کدام سطح نرمالیتی است؟
 (مهندسی کامپیوتر- آزاد ۹۲)



- (۱) 3NF
 (۲) BCNF
 (۳) 1NF
 (۴) 2NF

۴۵- رابطه دانشجو (کد دانشجو، نام، نام خانوادگی، درس، استاد) در کدام سطح نرمال قرار دارد؟ با فرض آن که یک استاد می‌تواند چندین درس را ارائه نماید و یک درس می‌تواند توسط چندین استاد ارائه شود؟
 (مهندسی کامپیوتر- آزاد ۹۲)

- (۱) 2NF (۲) 3NF (۳) 1NF (۴) 4NF

۴۶- با داشتن رابطه‌ی $R(A, B, C, D, E, F, G, H)$ و مجموعه وابستگی‌های تابعی $F = \{BE \rightarrow GH, G \rightarrow FA, D \rightarrow C, F \rightarrow B\}$ کدام گزینه نادرست است؟ (مهندسی کامپیوتر- دولتی ۹۳)
 (۱) رابطه R نرمال 3NF نیست.
 (۲) رابطه R نرمال BCNF نیست.
 (۳) کلیدی وجود ندارد که شامل A نباشد.
 (۴) کلیدی وجود ندارد که شامل D نباشد.

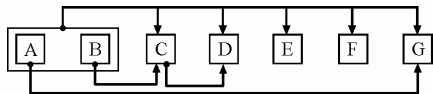
۴۷- رابطه $R(A, B, C, D, E)$ و مجموعه وابستگی‌های تابعی F را در نظر بگیرید.

$$F = \{BC \rightarrow A, A \rightarrow D, D \rightarrow C, D \rightarrow E\}$$

کدام یک از وابستگی‌های تابعی زیر BCNF را نقض نمی‌کند؟
 (مهندسی کامپیوتر - دولتی ۹۴)

- (۱) $BC \rightarrow A$ (۲) $D \rightarrow C$ (۳) $A \rightarrow D$ (۴) $D \rightarrow E$

۴۸- رابطه $R(A, B, C, D, E, F, G)$ که نمودار وابستگی‌های تابعی آن به صورت زیر می‌باشد مفروض است. اگر R بخواهد تا سطح 2NF نرمال سازی شود، جداول حاصل کدامند؟
 (مهندسی IT - دولتی ۹۴)

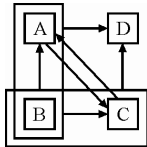


- (۱) BC, AG, ABDEF
 (۲) BC, CD, AG, ABEF
 (۳) BCD, AG, ABEF
 (۴) ABCDG, ABEF

۴۹- کدام یک از موارد زیر صحیح است؟
 (مهندسی کامپیوتر-دولتی-۹۵)

- (۱) رابطه‌ای نرمال است که هیچ یک از صفات ساده‌اش چندمقداری نباشند.
 (۲) کلید کاندید رابطه می‌تواند کاهش پذیر باشد.
 (۳) تاپل‌های یک رابطه نظم دارند.
 (۴) رابطه تاپل تکراری ندارد.

۵۰- فرض کنید در رابطه $R(A,B,C,D)$ ، کلید اصلی (AB) و کلید کاندید و روابط وابستگی تابعی به صورت زیر باشد. کدام گزینه در خصوص این رابطه صحیح است؟ (مهندسی IT-دولتی ۹۵)



(۱) رابطه R در فرم نرمال 3NF است ولی BCNF نیست.

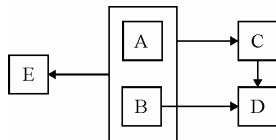
(۲) رابطه R در فرم نرمال BCNF است.

(۳) رابطه R در فرم نرمال 3NF نیست.

(۴) رابطه R در فرم نرمال 2NF نیست.

۵۱- اگر $R(A,B,C,D,E)$ یک رابطه در فرم نرمال BCNF و ABC تنها کلید R باشد، کدام یک از وابستگی‌های تابعی زیر حتماً برای R برقرار است؟ (مهندسی IT-دولتی ۹۵)

(۱) $BCE \rightarrow D$ (۲) $ACE \rightarrow D$ (۳) $ABD \rightarrow E$ (۴) $ABCD \rightarrow E$



۵۲- اگر مجموعه وابستگی‌های تابعی (FDهای) یک رابطه به

صورت شکل مقابل و کلید اصلی آن (A,B) باشد، این رابطه

در چه سطح نرمالی است؟ (مهندسی IT-دولتی ۹۶)

(۱) 1NF (۲) 2NF (۳) 3NF (۴) BCNF

۵۳- اگر در رابطه R ، تعداد ابرکلیدها با تعداد کلیدهای کاندید برابر باشد، آنگاه کدام مورد در خصوص رابطه R نادرست است؟ (مهندسی IT-دولتی ۹۷)

(۱) تنها یک خصیصه دارد.

(۲) در فرم نرمال 3NF است.

(۳) همه خصیصه‌های R ، NOT NULL هستند.

(۴) تعداد وابستگی‌های تابعی نابدهی R ، صفر هستند.

۵۴- رابطه و وابستگی‌های تابعی زیر را در نظر بگیرید.

$R(X,Y,Z)$

1) $Y \rightarrow Z$ 2) $XZ \rightarrow Y$ 3) $X \rightarrow Z$

(مهندسی کامپیوتر-دولتی ۹۸)

با توجه به رابطه فوق کدام عبارت نادرست است؟

(۱) X کلید کاندید است.

(۲) این رابطه در فرم نرمال BCNF نیست.

(۳) وابستگی سوم، اضافه و قابل حذف است.

(۴) X در وابستگی دوم، مشخصه اضافه و قابل حذف است.

۵۵- شمای رابطه‌ای $R(A,B,C,D,E,P,G)$ را در نظر بگیرید که در آن وابستگی‌های تابعی زیر برقرار است:

$F = \{AB \rightarrow CD, DE \rightarrow P, C \rightarrow E, P \rightarrow C, B \rightarrow G\}$

با توجه به وابستگی‌های تابعی فوق، کدام گزینه در مورد رابطه R برقرار است؟ (مهندسی IT-دولتی ۹۹)

(۱) رابطه R در شکل نرمال 2NF است، اما به شکل 3NF نیست.

(۲) رابطه R در شکل نرمال 3NF است، اما به شکل BCNF نیست.

(۳) رابطه R در شکل نرمال 2NF نیست.

(۴) رابطه R در شکل نرمال BCNF است.

۵۶- رابطه $R(A, B, C, D)$ و این وابستگی های تابعی را در نظر بگیرید. کدام گزینه در مورد رابطه R درست است؟
 $B \rightarrow C, CD \rightarrow B$ (مهندسی کامپیوتر-دولتی ۱۴۰۰)

(۱) R در 2NF نیست. (۲) R در BCNF است.
 (۳) R در 2NF است، اما در 3NF نیست. (۴) R در 3NF است، اما در BCNF نیست.

۵۷- حاصل تجزیه رابطه زیر بر اساس 3NF چند رابطه خواهد بود؟ (مهندسی کامپیوتر-دولتی ۱۴۰۰)

$R(A, B, C, D, E)$
 $A \rightarrow B, C$
 $B, C \rightarrow A, D$
 $D \rightarrow E$

(۱) رابطه 1 (۲) رابطه 2 (۳) رابطه 3 (۴) رابطه 4

۵۸- کدام مورد در خصوص رابطه درست است؟ (مهندسی IT- دولتی ۱۴۰۰)

(۱) رابطه ای نرمال است که هیچ یک از صفات ساده اش چند مقداری نباشند.
 (۲) کلید کاندید رابطه می تواند کاهش پذیر باشد.
 (۳) تاپل های یک رابطه نظم مکانی دارند.
 (۴) رابطه تاپل تکراری ندارد.

۵۹- رابطه $R = (A, B, C, D, E, F)$ را در نظر بگیرید. اگر مجموعه وابستگی های تابعی F روی R برقرار باشد، کدام یک از تجزیه های زیر دارای گمشدگی (lossless) است؟ (مهندسی IT- دولتی ۱۴۰۰)

$F = \{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$

(۱) $R_1(A, B, C), R_2(C, D, E)$ (۲) $R_1(A, B, C), R_2(A, D, E)$
 (۳) $R_1(E, B, C), R_2(E, D, A)$ (۴) $R_1(A, B, C), R_2(B, C, D, E)$

۶۰- رابطه $R = (A, B, C, D, E, F, G)$ را با وابستگی های زیر در نظر بگیرید: (مهندسی IT- دولتی ۱۴۰۰)

$AF \rightarrow BE$
 $FC \rightarrow DE$
 $F \rightarrow CD$
 $D \rightarrow E$
 $C \rightarrow A$

حاصل تجزیه 3NF این رابطه چند رابطه خواهد بود؟

(۱) 2 (۲) 3 (۳) 4 (۴) 5

پاسخ تست‌های فصل نهم: نرمال‌سازی

۱- گزینه (۲) صحیح است.

جدول studclg با مقادیر آن به شکل زیر را در نظر بگیرید:

S#	Sname	Clg#	Clgname
8621	Ali	12	Computer
8442	Reza	12	Computer
8731	Abbas	NULL	NULL

افزونگی

در جدول studclg

- این اطلاعات که «کد دانشکده‌ی کامپیوتر برابر 12 است.» در سطرهای اول و دوم تکرار شده است، پس **افزونگی داده** وجود دارد.
- اگر شخصی بخواهد که دانشکده‌ی کامپیوتر را به 22 تغییر دهد و به اشتباه، فقط مقدار clg# در سطر اول جدول studclg را عوض کند، از این به بعد دانشکده کامپیوتر دو کد خواهد داشت: 12 و 22. این موضوع یعنی **بی‌نظمی یا آنومالی** به معنی نقض جامعیت (صحت و سازگاری) در بانک اطلاعات است.
- در سطر آخر از این جدول، مقادیر NULL وجود دارد. به طور کلی، به ازای هر سطر از جدول studclg که clg# آن برابر NULL باشد یک سطر در جدول studclg خواهیم داشت که مقادیر دو ستون آخر آن NULL خواهد بود. پس مشکل مقادیر NULL نیز وجود دارد. حال اگر جدول studclg را به دو جدول clg و stud به شکل زیر تجزیه و نرمال کنیم:

S#	Sname	Clg#	Clg#	Clgname
8621	Ali	12	12	Computer
8442	Reza	12		
8731	Abbas	NULL		

جدول Clg

جدول Stud

مشکلات فوق مرتفع و نتایج زیر حاصل خواهد شد.

- کاهش افزونگی طبیعی (محتوایی) مانند تکرار بی‌دلیل سطر (12, computer)
- افزایش افزونگی تکنیکی به دلیل تعریف کلید خارجی (در قالب ستون های مشترک) بنابراین کاهش بعضی از انواع افزونگی، طبیعی کاهش و تکنیکی افزایش می‌یابد. بنابراین گزینه اول درست است.
- جداول تجزیه شده را مجدداً در نظر بگیرید:

S#	Sname	Clg#	Clg#	Clgname
8621	Ali	12	12	Computer
8442	Reza	12		
8731	Abbas	NULL		

جدول Clg

جدول Stud

حال پرس وجوی زیر را در نظر بگیرید:
شماره دانشجویی همه دانشجویان را ارائه دهید.

```
select S#
from stud
```

پرواضح است که این پرس وجو نیازی به الحاق دو جدول stud و clg ندارد. بنابراین این پرس وجو سربار حاصل از الحاق دو جدول را ندارد.

حال اگر پرس وجوی فوق را بر روی جدول studclg در حالت غیرنرمال اجرا کنیم. از آنجا که طول رکورد در جدول studclg نسبت به جدول stud بزرگتر است، واضح است که سرعت اجرای پرس وجوی فوق بر روی جدول stud بیش تر است. به بیان دیگر زمان اجرای پرس وجوی فوق بر روی جدول stud کاهش می یابد. مطابق رابطه ی زیر:

$$\text{طول رکورد (L)} = \text{نرخ انتقال (R)} \times \text{زمان انتقال رکورد (T}_p\text{)}$$

حال پرس وجوی زیر را در نظر بگیرید:
مشخصات کامل همه دانشجویان دانشکده کامپیوتر را ارائه دهید:

```
select *
from stud, clg
where stud.clg# = clg.clg# AND clgname = 'computer'
```

واضح است که این پرس وجو نیازی به الحاق دو جدول stud و clg دارد.
بنابراین این پرس وجو سربار حاصل از الحاق دو جدول را دارد.

در حالی که اگر پرس وجوی فوق بر روی جدول studclg در حالت غیرنرمال به شکل زیر اجرا می گردید.

```
select *
from studclg
where clgname = 'computer'
```

نیاز به عمل الحاق و سربار حاصل از آن نبود.

توجه: طول رکورد در پرس وجوی اخیر، در هر دو حالت جدول نرمال نشده و جداول نرمال شده برابر است، اما در حالت جداول نرمال شده، برای داشتن تمام ستون ها نیاز به یک عمل اضافه و دارای سربار به نام الحاق وجود دارد.

بنابراین از آنجا که نرمال سازی باعث تجزیه جداول می شود گاهی اوقات برای بازیابی نیاز به پیوند مجدد این جداول داریم که باعث افزایش سربار سیستم می گردد، البته نه همیشه مانند پرس وجوی قبل که نیاز به الحاق نداشت.

بنابراین گزینه دوم نادرست است، زیرا نرمال سازی رابطه ها گاهی موجب افزایش سربار سیستم در عمل بازیابی به دلیل نیاز به الحاق جداول می شود و در نتیجه سرعت پاسخگویی به پرس وجوها را کاهش می دهد. بنابراین افزایش یا کاهش سرعت پرس وجوها بر روی جداول نرمال شده، بستگی به نوع پرس وجوها دارد.

گزینه سوم و چهارم درست است. زیرا برقراری جامعیت (صحت و سازگاری) در جداول نرمال شده، ساده تر و قابل کنترل تر می باشد. مانند استفاده از دستور on update cascade برای حفظ سازگاری ستون clg# در دو جدول stud و clg در برابر تغییرات. در حالی که در جدول غیرنرمال شده studclg، تغییر 12 به

22 به صورت دستی و توسط انسان انجام می‌شود و خطای انسانی در بر دارد و ممکن است، تغییر 12 به 22 در سطر دوم فراموش گردد! انسان است و جایزالخطا بودنش. همچنین حل مشکل مقادیر بوج (NULL)، یکی دیگر از اهداف نرمال‌سازی است و این که حل پدیده آنومالی (بی‌نظمی و ناسازگاری)، یکی دیگر از اهداف نرمال‌سازی است.

۲- گزینه (۲) صحیح است.

گزینه‌های اول و چهارم به صورت مشخص جزء اهداف فرآیند نرمال‌سازی محسوب می‌شوند. گزینه دوم ارتباطی به بحث نرمال‌سازی ندارد و به عنوان یکی از اهداف اصلی نرمال‌سازی محسوب نمی‌شود، زیرا اهداف نرمال‌سازی کاهش افزونگی داده‌ها (کاهش افزونگی طبیعی یا محتوایی) مقابله با بی‌نظمی، آنومالی و ناسازگاری و سهولت در برقراری جامعیت (صحت و سازگاری) و کاهش مقادیر تهی (NULL) است. اینها اهداف نرمال‌سازی هستند، اما در این مسیر و رسیدن به این اهداف لازم است برخی وابستگی‌های افزونگی ساز، مثل وابستگی بخشی (وابستگی صفت غیرکلید به جزئی از کلید کاندید)، وابستگی انتقالی (وابستگی صفت غیرکلید به صفت غیرکلید) حذف گردند، که این مسیر است و نه هدف. که طراح محترم گزینه سوم را به عنوان نرمال‌سازی در نظر گرفته‌اند. البته مفهومی تحت عنوان انواع قواعد وابستگی تابعی در مباحث پایگاه داده وجود ندارد که حال نرمال‌سازی موجب حذف برخی از این قوانین گردد، بهتر است گزینه سوم به «حذف بعضی از وابستگی‌های تابعی» اصلاح گردد که حتی در این صورت نیز به عنوان یکی از اهداف کلی فرآیند نرمال‌سازی محسوب نمی‌شود، بلکه حذف بعضی از وابستگی‌های تابعی، مسیر رسیدن به اهداف نرمال‌سازی است. پس از آن‌که نرمال‌سازی با اهداف مذکور انجام شد، پیاده‌سازی دید کاربر هم در فاز پیاده‌سازی انجام می‌گردد، نرمال‌سازی و کاهش افزونگی‌ها فدای راحتی برای پیاده‌سازی دید کاربر نمی‌گردد. البته متذکر می‌شویم که تسهیل پیاده‌سازی دید کاربر هیچ جایگاهی در اهداف نرمال‌سازی ندارد، نرمال‌سازی که انجام شد، پیاده‌سازی دید کاربر هم براساس نیازها و سلیقه‌ی کاربر انجام خواهد شد.

۳- گزینه (۲) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

توجه: ما نام این روابط را به عنوان مبدع آن «قوانین ارسطو» نام‌گذاری کردیم، این قوانین به «قوانین چهارگانه ارسطو» نیز موسوم است.

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیربديهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [(چپ) x_i - (راست) y_i]$$

توجه: عبارت $[(چپ) x_i - (راست) y_i]$ به طور مستقل بر روی تک تک وابستگی‌ها انجام می‌گردد.

$$A \rightarrow BC \Rightarrow BC - A = BC$$

مثال:

توجه: استفاده از روش اول مستلزم گام ابتدایی حذف وابستگی های بدیهی است، اگر در حذف وابستگی های بدیهی دچار خطا می شوید، از روش دوم استفاده نمایید.
با توجه به وابستگی های مطرح شده برای رابطه $R(A, B, C, D)$ داریم:

$$AB \rightarrow C$$

$$AB \rightarrow D$$

$$C \rightarrow A$$

$$D \rightarrow B$$

$$ABCD - ABCD = \text{تهی}$$

قانون چهارم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، تهی گردد، بدین معنی است که، جدول فوق چندین کلید کاندید دارد، که هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید آن وجود ندارد. بنابراین باید کلید کاندید با بررسی دقیق بر روی مجموعه وابستگی کشف گردد.

$$\{AB\}^+ = \{A, B, C, D\}$$

بر اساس بستار فوق، صفات AB، همه ستون ها را بدون عضو زائد تولید می کند، پس صفات AB، کلید کاندید می باشد.

حال کلید کاندید AB را در نظر بگیرید، از آنجا که مطابق وابستگی $C \rightarrow A$ ، صفت C، صفت A را می دهد. می توان به جای صفت A در کلید کاندید AB، صفت C را قرار داد. چون مجدداً مطابق وابستگی $C \rightarrow A$ ، صفت A را می دهد که منجر به ایجاد کلید کاندید CB می گردد.

مطابق الگوی زیر:

$$\begin{array}{c} AB \\ \uparrow \Rightarrow \\ CB \\ C \end{array}$$

$$\{CB\}^+ = \{C, B, A, D\}$$

بر اساس بستار فوق، صفات CB، همه ستون ها را بدون عضو زائد تولید می کند، پس صفات CB کلید کاندید می باشد.

حال مجدداً کاندید AB را در نظر بگیرید، از آنجا که مطابق وابستگی $D \rightarrow B$ ، صفت D، صفت B را می دهد. می توان به جای صفت B در کلید کاندید AB، صفت D را قرار داد. چون مجدداً مطابق وابستگی $D \rightarrow B$ ، صفت B را می دهد که منجر به ایجاد کلید کاندید AD می گردد.

مطابق الگوی زیر:

$$\begin{array}{c} AB \\ \uparrow \Rightarrow \\ AD \\ D \end{array}$$

$$\{AD\}^+ = \{A, D, B, C\}$$

بر اساس بستار فوق، صفات AD، همه ستون ها را بدون عضو زائد تولید می کند، پس صفات AD کلید کاندید است.

حال کلید کاندید CB یا AD را در نظر بگیرید، براساس وابستگی‌های $C \rightarrow A$ و $D \rightarrow B$ و مطابق الگوی زیر داریم:

$$CB \Rightarrow CD$$

↑

D

$$AD \Rightarrow CD$$

↑

C

$$\{CD\}^+ = \{C, D, A, B\}$$

براساس بستار فوق، صفات CD، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفات CD کلید کاندید است.

بررسی مجموعه وابستگی‌های مطرح شده نشان داد که (A, B) و (C, B) و (A, D) و (C, D) به عنوان کلیدهای کاندید برای رابطه R محسوب می‌شوند. بنابراین گزینه اول گزاره‌ای درست است.

توجه: همان‌طور که مشاهده می‌شود، مطابق قانون چهارم ارسطو، هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید فوق وجود ندارد.

مستقل از متن سؤال گزینه دوم گزاره‌ای نادرست را بیان می‌کند، زیرا طبق تعریف هر رابطه‌ای در BCNF باشد در 3NF هم هست.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم اول را به صورت زیر بیان کرد:

□ دارای حداقل یک کلید کاندید باشد.

□ همه خصیصه‌های آن غیرقابل تجزیه باشند (جدول باید فاقد خصیصه‌های مرکب باشد)

□ همه خصیصه‌های آن تک مقداری باشند (جدول باید فاقد خصیصه‌های چند مقداری باشد)

واضح است که جدول مطرح شده در فرم اول نرمال قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

□ جدول باید در نرمال فرم اول باشد.

□ جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مولفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می‌نامند.

مولفه غیرکلید: هر صفتی که عضو هیچ کلید کاندیدی نباشد، به عنوان مؤلفه غیرکلیدی نامیده می‌شود.

مؤلفه جزء کلید کاندید: هر صفتی که عضو حداقل یک کلید کاندید باشد، به عنوان مؤلفه جزء کلید نامیده می‌شود.

وابستگی‌های مطرح شده را در نظر بگیرید:

$$\text{کلید کاندید } \overline{AB} \longrightarrow \text{جزء کلید } \overline{C}$$

$$\text{کلید کاندید } \overline{AB} \longrightarrow \text{جزء کلید } \overline{D}$$

$$\text{جزء کلید } \overline{C} \longrightarrow \text{جزء کلید } \overline{A}$$

$$\text{جزء کلید } \overline{D} \longrightarrow \text{جزء کلید } \overline{B}$$

در وابستگی های فوق، وابستگی بخشی وجود ندارد. بنابراین جدول مربوطه در نرمال فرم دوم هم قرار دارد.

به طور کلی می توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

□ جدول باید در نرمال فرم دوم باشد.

□ جدول باید فاقد وابستگی انتقالی باشد.

وابستگی انتقالی: وابستگی یک مؤلفه غیرکلیدی به یک مؤلفه غیرکلیدی دیگر را وابستگی انتقالی می نامند.

در وابستگی های فوق، وابستگی انتقالی وجود ندارد. بنابراین جدول مربوطه در نرمال فرم سوم هم قرار دارد.

به طور کلی می توان شرط قرار داشتن یک جدول در نرمال فرم BCNF را به صورت زیر بیان کرد:

□ سمت چپ همه مجموعه وابستگی ها، ابرکلید باشد.

در وابستگی های فوق، سمت چپ وابستگی اول و دوم در مجموعه داده شده ابرکلید است. اما سمت چپ وابستگی های سوم و چهارم ابرکلید نیست.

بنابراین این جدول به دلیل نقض شرایط مربوطه، در BCNF واقع نیست. در نتیجه گزینه های سوم و چهارم نیز گزاره های درستی را بیان می کنند.

۴- گزینه (۴) صحیح است.

جدولی که BCNF باشد، حتماً 3NF است، بنابراین گزینه اول درست است به طور کلی می توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

□ جدول باید در نرمال فرم دوم باشد.

□ جدول باید فاقد وابستگی انتقالی باشد.

وابستگی انتقالی: وابستگی یک مؤلفه غیرکلیدی به یک مؤلفه غیرکلیدی دیگر را وابستگی انتقالی می نامند.

بنابراین گزینه دوم درست است و گزینه سوم نادرست است. زیرا عدم وابستگی انتقالی برای یک جدول نشانه 3NF بودن یک جدول نیست بلکه جدول باید در نرمال فرم دوم هم باشد.

مثال: جدول زیر را در نظر بگیرید:

a	b	c
1	2	3
3	6	5
7	2	6
1	4	8

در جدول فوق ab کلید کاندید است و صفت غیرکلید c با عضو کلید کاندید یعنی b، وابستگی بخشی دارد. اما نمی توان ادعا کرد که چون جدول فوق فاقد وابستگی انتقالی است، پس 3NF است. زیرا همان طور که گفتیم جدولی در 3NF است که علاوه بر عدم وابستگی انتقالی، 2NF هم باشد، که جدول فوق به دلیل وجود وابستگی بخشی، 2NF نیست، پس جدول فوق 3NF هم نیست.

۵- گزینه (۴) صحیح است.

جدول $R(\underline{X}, Y, Z)$ تنها یک کلید کاندید دارد، این جدول فاقد وابستگی بخشی (وابستگی غیرکلید به عضو

کلید کاندید) و وابستگی انتقالی (وابستگی غیرکلید به غیرکلید) می‌باشد. و تنها دارای وابستگی تابعی کامل است.

به صورت زیر:

$R(\underline{X}, Y, Z)$

وابستگی تابعی کامل $X \rightarrow Y$

وابستگی تابعی کامل $X \rightarrow Z$

در جدول R، صفت X، کلید کاندید است.

جدول R به دلیل عدم وجود وابستگی بخشی و وابستگی انتقالی در نرمال فرم سوم (3NF) قرار دارد. هم‌چنین از آن‌جا که سمت چپ تمام وابستگی‌ها، ابرکلید است، بنابراین جدول فوق در سطح BCNF نیز قرار دارد. هم‌چنین اگر جدولی در BCNF باشد و حداقل یک کلید کاندید تک خصیصه‌ای داشته باشد، حتماً در نرمال فرم چهارم (4NF) هم واقع است. بنابراین جدول فوق در سطح 4NF نیز قرار دارد. هم‌چنین اگر جدولی در نرمال فرم چهارم باشد و همه کلیدهای کاندید آن تک خصیصه‌ای باشند آن جدول در نرمال فرم پنجم واقع است، بنابراین جدول فوق در سطح 5NF نیز قرار دارد. بنابراین جدول R نیاز به نرمال‌سازی ندارد. اما نظر طراح محترم، اصرار بر نرمال‌سازی است! مطابق ضوابط ریسانن تجزیه R به دو رابطه R1 و R2 مطلوب است. اگر و فقط اگر R1 و R2 مستقل باشند، یعنی:

۱- شرط لازم: دو جدول الحاق‌پذیر باشند (صفت مشترک در دو جدول، حداقل در یکی کلید کاندید باشد)

۲- شرط کافی: تمام وابستگی‌های تابعی R قابل استنتاج از روی وابستگی‌های تابعی R1 و R2 باشد یا در آن‌ها موجود باشد.

در صورت سؤال مطرح شده است که «کدام گزینه از شرط‌های صحت این عمل می‌باشد؟»

بنابراین گزینه چهارم با گزاره اول (شرط لازم) ضوابط ریسانن تطابق دارد.

بنابراین گزینه چهارم، پاسخ این تست خواهد بود.

اما برای رسیدن به تجزیه مطلوب باید هر دو شرط از ضوابط ریسانن برقرار باشد.

با توجه به این که صفت X در جدول R کلید کاندید است، و به تبع وابستگی‌های $X \rightarrow Y$ و $X \rightarrow Z$ در این جدول وجود دارد.

با نرمال‌سازی جدول R به R1 و R2 نیز، وابستگی‌های $X \rightarrow Y$ و $X \rightarrow Z$ باید حفظ گردند.

بنابراین برای برقراری گزاره دوم (شرط کافی) ضوابط ریسانن باید صفت X در دو جدول R1 و R2 کلید کاندید باشد.

$R(\underline{X}, Y, Z)$

$X \rightarrow Y$

$X \rightarrow Z$

پس از تجزیه داریم:

$R1(\underline{X}, Y)$ $R2(\underline{X}, Z)$

$X \rightarrow Y$ $X \rightarrow Z$

۶- گزینه (۴) صحیح است.

جدول XYZ، $2^3 - 1 = 7$ حالت مختلف، می تواند به صورت زیر کلید کاندید داشته باشد.

X Y Z	صفات	تکرار
0 0 0	تهی	
0 0 1	Z	سطر تکراری دارد
0 1 0	Y	سطر تکراری دارد
0 1 1	Y Z	سطر تکراری دارد
1 0 0	X	سطر تکراری دارد
1 0 1	X Z	سطر تکراری دارد
1 1 0	X Y	سطر تکراری دارد
1 1 1	X Y Z	سطر تکراری ندارد

با توجه به جدول فوق و بررسی دقیق محتویات جدول داده شده، واضح است که جدول داده شده تمام کلید است، یعنی کلید کاندید آن XYZ است. طبق تعریف، جدولی که تمام کلید است، در نرمال فرم BCNF قرار دارد. توجه کنید که جدولی که در BCNF است، حتماً در نرمال فرم های اول، دوم و سوم هم قرار دارد، بنابراین گزینه های اول تا سوم هم درست هستند. بهتر بود بالاترین سطح نرمال این جدول مورد سؤال قرار می گرفت.

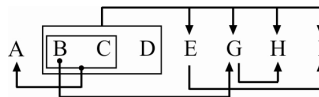
۷- گزینه (۴) صحیح است.

جدولی در سطح نرمال فرم اول قرار دارد که صفت چند مقداری و مرکب نداشته باشد. از آن جا که در رابطه R، صفت F یک صفت مرکب است، بنابراین این جدول در نرمال فرم اول قرار ندارد. برای نرمال سازی این جدول ابتدا صفت مرکب F باید حذف و اجزای آن در جدول R جایگزین گردد.

به این ترتیب جدول R به شکل زیر در نرمال فرم اول قرار می گیرد:

$R(A, \underline{B}, \underline{C}, \underline{D}, E, G, H, I)$

نمودار وابستگی های جدول به صورت زیر است:



به طور کلی می توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

جدول باید در نرمال فرم اول باشد.

جدول باید فاقد وابستگی بخشی باشد.

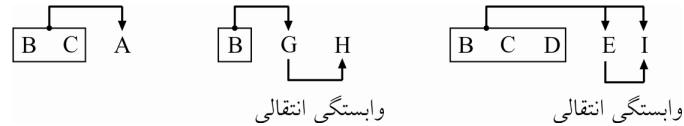
وابستگی بخشی: وابستگی یک مولفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می نامند.

با توجه به این که ترکیب صفات BCD به عنوان کلید کاندید رابطه R معرفی شده است، وابستگی های تابعی زیر:

$B \rightarrow G$

$BC \rightarrow A$

به عنوان وابستگی بخشی محسوب می‌گردند. بنابراین جداول R در نرمال فرم دوم قرار ندارد. بنابراین برای قرار دادن جداول R، در نرمال فرم دوم، باید وابستگی‌های بخشی از جداول R خارج گردند.



بنابراین جدول R به سه جدول کوچک‌تر تجزیه می‌گردد:

$$R3(\underline{B}, C, D, E, I) \quad R2(\underline{B}, G, H) \quad R1(\underline{B}, C, A)$$

با توجه به کلید کاندید جدول R1 می‌توان دریافت که این جدول فاقد وابستگی بخشی و انتقالی است، بنابراین این جدول در نرمال فرم سوم قرار دارد. همچنین از آنجا که سمت چپ تمام وابستگی‌های جدول R1 ابرکلید است، بنابراین جدول R1 در سطح BCNF قرار دارد.

به‌طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

جدول باید در نرمال فرم دوم باشد.

جدول باید فاقد وابستگی انتقالی باشد.

وابستگی انتقالی: وابستگی یک مؤلفه غیرکلیدی به یک مؤلفه غیرکلیدی دیگر را وابستگی انتقالی می‌نامند.

در جدول R2 و R3 وابستگی‌های تابعی زیر:

$$G \rightarrow H$$

$$E \rightarrow I$$

به عنوان وابستگی انتقالی محسوب می‌گردند. بنابراین جداول R2 و R3 در نرمال فرم سوم قرار ندارند. بنابراین برای قرار دادن R2 و R3، در نرمال فرم سوم، باید وابستگی‌های انتقالی از جدول R2 و R3 خارج گردند.



بنابراین جداول R2 و R3 به جداول کوچک‌تر تجزیه می‌گردند:

$$R21(\underline{B}, G) \quad R22(\underline{G}, H) \quad R31(\underline{B}, C, D, E) \quad R32(\underline{E}, I)$$

با توجه به کلیدهای کاندید در جداول R21, R22, R31 و R32 می‌توان دریافت که این جداول فاقد وابستگی بخشی و انتقالی هستند و بنابراین در نرمال فرم سوم قرار دارند. همچنین از آنجا که سمت چپ تمام وابستگی‌های جداول R21, R22, R31 و R32 ابرکلید است، بنابراین جداول R21, R22, R31 و R32 در سطح BCNF نیز قرار دارند. به این ترتیب با تجزیه جدول R به پنج جدول R1, R21, R22, R31 و R32 می‌توان جدول R را در نرمال فرم سوم و سطح BCNF قرار داد.

۸- گزینه (۱) صحیح است.

به‌طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

جدول باید در نرمال فرم اول باشد.

جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مؤلفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می‌نامند.

با توجه به این که ترکیب صفات BCD به عنوان کلید کاندید رابطه R معرفی شده است، وابستگی های زیر را داریم:

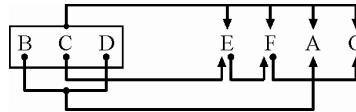
B	C	D	وابستگی ها
0	0	0	ϕ
0	0	1	$D \rightarrow X$
0	1	0	$C \rightarrow X$
0	1	1	$CD \rightarrow X$
1	0	0	$B \rightarrow X$
1	0	1	$BD \rightarrow X$
1	1	0	$BC \rightarrow X$
1	1	1	$BCD \rightarrow R$

} وابستگی بخشی
 X: مؤلفه غیر کلیدی
 } وابستگی تابعی کامل
 گزینۀ اول \Rightarrow

بنابراین فرم نرمال دوم جدول R به واسطه وابستگی بخشی $BC \rightarrow E$ موجود در گزینۀ اول، نقض می گردد.

۹- گزینۀ (۴) صحیح است.

جدولی در سطح نرمال فرم اول قرار دارد که صفت چند مقداری و مرکب نداشته باشد. از آن جا که در جدول R، صفت چند مقداری و مرکب وجود ندارد، بنابراین این جدول در نرمال فرم اول قرار دارد. نمودار وابستگی های جدول R به صورت زیر است:



به طور کلی می توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

جدول باید در نرمال فرم اول باشد.

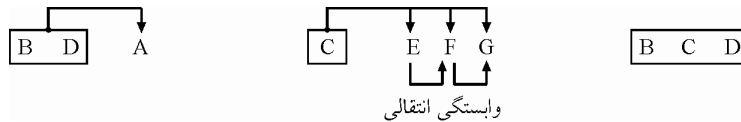
جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مؤلفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می نامند. با توجه به این که ترکیب صفات BCD به عنوان کلید رابطه R معرفی شده است، وابستگی های تابعی زیر:

$BD \rightarrow A$

$C \rightarrow E$

به عنوان وابستگی بخشی محسوب می گردند. بنابراین جدول R در نرمال فرم دوم قرار ندارد. بنابراین برای قرار دادن جدول R در نرمال فرم دوم، باید وابستگی های بخشی از جدول R خارج گردند.



بنابراین جدول R به سه جدول کوچکتر تجزیه می گردد:

$R1(\underline{B}, \underline{D}, A)$ $R2(\underline{C}, E, F, G)$ $R3(\underline{B}, \underline{C}, \underline{D})$

با توجه به کلید کاندید در دو جدول R1 و R3 می توان دریافت که این دو جدول فاقد وابستگی بخشی و انتقالی هستند، بنابراین در نرمال فرم سوم قرار دارند. هم چنین از آن جا که سمت چپ تمام وابستگی های دو جدول R1 و R3، ابرکلید است، بنابراین دو جدول R1 و R3 در سطح BCNF نیز قرار دارند.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

□ جدول باید در نرمال فرم دوم باشد.

□ جدول باید فاقد وابستگی انتقالی باشد.

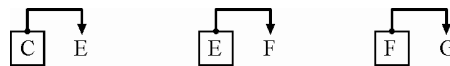
وابستگی انتقالی: وابستگی یک مؤلفه غیرکلیدی به یک مؤلفه غیرکلیدی دیگر را وابستگی انتقالی می‌نامند.

در جدول R2 وابستگی‌های تابعی زیر:

$E \rightarrow F$

$F \rightarrow G$

به عنوان وابستگی انتقالی محسوب می‌گردند. بنابراین جدول R2 در نرمال فرم سوم قرار ندارد. بنابراین برای قرار دادن جدول R2، در نرمال فرم سوم، باید وابستگی‌های انتقالی از جدول R2 خارج گردند.



بنابراین جدول R2 به سه جدول کوچک‌تر تجزیه می‌گردد.

$R_{21}(C,E)$ $R_{22}(E,F)$ $R_{23}(F,G)$

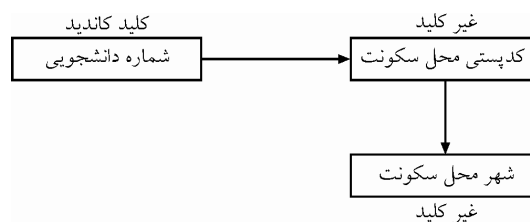
با توجه به کلیدهای کاندید در دو جدول R21، R22 و R23 می‌توان دریافت که این سه جدول فاقد وابستگی بخشی و انتقالی هستند و بنابراین در نرمال فرم سوم قرار دارند. همچنین از آنجا که سمت چپ تمام وابستگی‌های سه جدول R21، R22 و R23 ابرکلید است، بنابراین سه جدول R21، R22 و R23 در سطح BCNF نیز قرار دارند. به این ترتیب با تجزیه جدول R به پنج جدول R1، R21، R22، R23 و R3 می‌توان جدول R را در نرمال فرم سوم و سطح BCNF قرار داد.

۱۰- گزینه (۳) صحیح است.

قانون افزایشی به این صورت بیان می‌شود که اگر وابستگی چند مقداری $X \rightarrow Y$ موجود باشد و $V \subseteq W$ آن‌گاه می‌توان نتیجه گرفت $WX \rightarrow VY$.

به بیان دیگر با توجه به این که $V \subseteq W$ است، می‌توان نتیجه گرفت که $W \rightarrow V$ است و از طرفی دیگر با توجه به فرض مسأله $X \rightarrow Y$ می‌توان نتیجه گرفت $XW \rightarrow VY$.

۱۱- گزینه (۴) صحیح است.



از آنجا که در جدول دانشجویی، صفت چند مقداری و مرکب وجود ندارد، بنابراین این جدول در نرمال فرم اول قرار دارد.

به طور کلی می توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

جدول باید در نرمال فرم اول باشد.

جدول باید فاقد وابستگی بخشی باشد.

جدول دانشجو در حال حاضر 2NF است، زیرا کلید کاندید آن شماره دانشجویی است و وابستگی بخشی وجود ندارد.

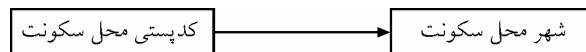
به طور کلی می توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

جدول باید در نرمال فرم دوم باشد.

جدول باید فاقد وابستگی انتقالی باشد.

وابستگی انتقالی: وابستگی یک مولفه غیرکلیدی به یک مولفه غیرکلیدی دیگر را، وابستگی انتقالی می نامند.

در جدول دانشجو، وابستگی تابعی زیر:



به عنوان وابستگی انتقالی محسوب می گردد. بنابراین جدول دانشجو در نرمال فرم سوم قرار ندارند. بنابراین برای قرار دادن جدول دانشجو، در نرمال فرم سوم، باید وابستگی های انتقالی از جدول دانشجو خارج کردند.



بنابراین جدول دانشجو به دو جدول کوچکتر تجزیه می گردد.

(شهر محل سکونت و کد پستی محل سکونت) stu2 (کد پستی محل سکونت و شماره دانشجویی) stu1 با توجه به کلیدهای کاندید در دو جدول stu1 و stu2 می توان دریافت که این دو جدول فاقد وابستگی بخشی و انتقالی هستند، بنابراین در نرمال فرم سوم قرار دارند. هم چنین از آن جا که سمت چپ تمام وابستگی های دو جدول stu1 و stu2 ابرکلید است، بنابراین این دو جدول در سطح BCNF نیز قرار دارند. به این ترتیب با تجزیه جدول دانشجو به دو جدول stu1 و stu2 می توان جدول دانشجو را در نرمال فرم سوم و سطح BCNF قرار داد.

البته این دو جدول در نرمال فرم های چهارم و پنجم نیز قرار دارند، اما مطابق آنچه در گزینه ها آمده است، گزینه چهارم به عنوان پاسخ خواهد بود.

۱۲- گزینه (۴) صحیح است.

راه حل اول:

فرم کمینه وابستگی های مطرح شده به صورت زیر است:

۱- گام اول: تجزیه همه صفت های سمت راست.

A → I

A → E

A → G

$AB \rightarrow I$
 $AB \rightarrow E$
 $AB \rightarrow C$
 $AB \rightarrow D$
 $ABE \rightarrow C$
 $ABE \rightarrow D$
 $ABE \rightarrow F$
 $ABE \rightarrow G$
 $ABE \rightarrow H$
 $AC \rightarrow I$
 $AC \rightarrow E$

C زیر مجموعه AC است، بنابراین این وابستگی بدیهی بوده و حذف می‌گردد.

$AC \rightarrow C$ *

DE زیر مجموعه CDE است، بنابراین این وابستگی بدیهی بوده و حذف می‌گردد.

$CDE \rightarrow DE$ *

$E \rightarrow F$
 $E \rightarrow G$
 $F \rightarrow E$

۲- گام دوم: خلاصه‌سازی صفتهای سمت چپ تا حد امکان

$A \rightarrow I$
 $A \rightarrow E$
 $A \rightarrow G$
 $AB \rightarrow I$
 $AB \rightarrow E$
 $AB \rightarrow C$
 $AB \rightarrow D$

زیرا توسط وابستگی $AB \rightarrow E$ ، صفات AB، صفات E را می‌دهد.

$ABE \rightarrow C \Rightarrow AB \rightarrow C$

$ABE \rightarrow D \Rightarrow AB \rightarrow D$

$ABE \rightarrow F \Rightarrow AB \rightarrow F$

$ABE \rightarrow G \Rightarrow AB \rightarrow G$

$ABE \rightarrow H \Rightarrow AB \rightarrow H$

$AC \rightarrow I$

$AC \rightarrow E$

$E \rightarrow F$

$E \rightarrow G$

$F \rightarrow E$

۳- گام سوم: حذف وابستگی های زائدی که از روی سایر وابستگی ها قابل تولید است.

$A \rightarrow I$ ✓

$A \rightarrow E$ ✓

$A \rightarrow G$ ✗

با داشتن صفت A بدون وابستگی $A \rightarrow G$ ، توسط وابستگی های $A \rightarrow E$ و $E \rightarrow G$ می توان به صفت G رسید.

بنابراین وابستگی $A \rightarrow G$ حذف می گردد. $\{A\}^+ = \{A, E, \boxed{G}\}$.

$AB \rightarrow I$ ✗

با داشتن صفات AB بدون وابستگی $AB \rightarrow I$ ، توسط وابستگی $A \rightarrow I$ می توان به صفت I رسید. بنابراین

وابستگی $AB \rightarrow I$ حذف می گردد. $\{AB\}^+ = \{A, B, \boxed{I}\}$

$AB \rightarrow E$ ✗

با داشتن صفات AB بدون وابستگی $AB \rightarrow E$ ، توسط وابستگی $A \rightarrow E$ می توان به صفت E رسید.

بنابراین وابستگی $AB \rightarrow E$ حذف می گردد. $\{AB\}^+ = \{A, B, \boxed{E}\}$

$AB \rightarrow C$ ✓

$AB \rightarrow D$ ✓

$AB \rightarrow C$ ✗ تکراری

$AB \rightarrow D$ ✗ تکراری

$AB \rightarrow F$ ✗

با داشتن صفات AB بدون وابستگی $AB \rightarrow F$ ، توسط وابستگی $A \rightarrow E$ و $E \rightarrow F$ می توان به صفت F رسید. بنابراین وابستگی $AB \rightarrow F$ حذف می گردد.

$\{AB\}^+ = \{A, B, E, \boxed{F}\}$

$AB \rightarrow G$ ✗

با داشتن صفات AC بدون وابستگی $AB \rightarrow G$ ، توسط وابستگی $A \rightarrow G$ می توان به صفت G رسید.

بنابراین وابستگی $AB \rightarrow G$ حذف می گردد. $\{AB\}^+ = \{A, B, \boxed{G}\}$

$AB \rightarrow H$ ✓

$AC \rightarrow I$ ✗

با داشتن صفات AC بدون وابستگی $AC \rightarrow I$ ، توسط وابستگی $A \rightarrow I$ ، می توان به صفت I رسید.

بنابراین وابستگی $AC \rightarrow I$ حذف می گردد. $\{AC\}^+ = \{A, C, \boxed{I}\}$

$AC \rightarrow E$ ✗

با داشتن صفات AC بدون وابستگی $AC \rightarrow E$ ، توسط وابستگی $A \rightarrow E$ ، می توان به صفت E رسید.

بنابراین وابستگی $AC \rightarrow E$ حذف می گردد. $\{AC\}^+ = \{A, C, \boxed{E}\}$

$E \rightarrow F$ ✓

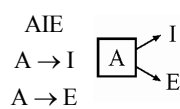
$E \rightarrow G$ ✓

$F \rightarrow E$ ✓

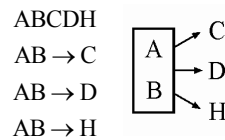
بنابراین وابستگی‌های کمینه، مجموعه وابستگی‌های مطرح شده به صورت زیر خواهد بود:

$A \rightarrow I$
 $A \rightarrow E$
 $AB \rightarrow C$
 $AB \rightarrow D$
 $AB \rightarrow H$
 $E \rightarrow F$
 $E \rightarrow G$
 $F \rightarrow E$

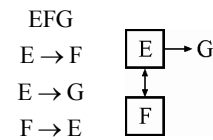
حال با توجه به وابستگی‌های مربوط به هر جدول، سطح نرمال مربوط به هر جدول را می‌توان مشخص نمود:



کلید کاندید A می‌باشد.



کلید کاندید AB می‌باشد.



کلید کاندید E یا F می‌باشد.

همان‌طور که مشاهده می‌شود سمت چپ تمام وابستگی‌های تابعی در هر سه جدول ابرکلید است، بنابراین هر سه جدول در BCNF قرار دارند. بنابراین پایگاه داده در BCNF قرار دارد.

راه‌حل دوم: پیدا کردن کلید کاندید هر جدول (بدون نیاز به کمینه‌سازی وابستگی‌ها) به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

- ۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.
- ۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیربدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [(\text{چپ}) x_i - (\text{راست}) y_i]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R_1(A, I, E)$ داریم:

$A \rightarrow IEG$
 $AB \rightarrow IEC D$
 $ABE \rightarrow CDFGH$
 $AC \rightarrow IEC$

* وابستگی بدیهی، کنار گذاشته می شود. $CDE \rightarrow DE \Rightarrow$

$$E \rightarrow FG$$

$$F \rightarrow E$$

$$AIE - IEGCDFH = A$$

بنابر رابطه فوق صفت A حتماً باید عضو کلید کاندید باشد.

بستار صفت A فقط برای جدول $R1(A, I, E)$ به صورت زیر است:

$$\{A\}^+ = \{A, I, E\}$$

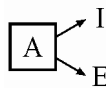
براساس بستار فوق، صفت A، همه ستون های جدول $R1(\underline{A}, I, E)$ را تولید می کند، پس صفت A کلید کاندید می باشد.

قانون دوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

براساس وابستگی های مطرح شده جدول $R1(A, I, E)$ با کلید کاندید A، فاقد هر گونه وابستگی بخشی (وابستگی غیر کلید به عضو کلید کاندید) و انتقالی (وابستگی غیر کلید به غیر کلید) است. بنابراین این جدول در فرم نرمال 3NF قرار دارد.

بنابراین نمودار وابستگی تابعی جدول $R1(\underline{A}, I, E)$ به صورت زیر خواهد بود:



همان طور که مشاهده می شود سمت چپ تمام وابستگی های تابعی جدول $R1(\underline{A}, I, E)$ ابرکلید است، بنابراین جدول $R1(\underline{A}, I, E)$ در BCNF قرار دارد.

هم چنین با توجه به وابستگی های مطرح شده برای جدول $R2(A, B, C, D, H)$ داریم:

$$A \rightarrow IEG$$

$$AB \rightarrow IECD$$

$$ABE \rightarrow CDFGH$$

$$AC \rightarrow IEC$$

* وابستگی بدیهی، کنار گذاشته می شود $CDE \rightarrow DE \Rightarrow$

$$E \rightarrow FG$$

$$F \rightarrow E$$

$$ABCDH - IEGCDFH = AB$$

بنابر رابطه فوق صفات AB حتماً باید عضو کلید کاندید باشد.

بستار صفات AB فقط برای جدول $R2(A, B, C, D, H)$ به صورت زیر است:

$$\{AB\}^+ = \{A, B, C, D, H\}$$

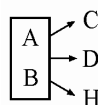
براساس بستار فوق، صفات AB، همه ستون های جدول $R2(\underline{A}, B, C, D, H)$ را تولید می کند، پس صفات AB کلید کاندید می باشد.

قانون دوم ارسطو

هرگاه عضو کلید کاندید حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

براساس وابستگی‌های مطرح شده جدول $R2(A, B, C, D, H)$ با کلید کاندید AB ، فاقد هرگونه وابستگی بخشی (وابستگی غیرکلید به عضو کلید کاندید) و انتقالی (وابستگی غیرکلید به غیرکلید) است. بنابراین جدول در فرم نرمال 3NF قرار دارد.

بنابراین نمودار وابستگی تابعی جدول $R2(A, B, C, D, H)$ به صورت زیر خواهد بود:



همان‌طور که مشاهده می‌شود سمت چپ تمام وابستگی‌های تابعی جدول $R2(A, B, C, D, H)$ ابرکلید است، بنابراین جدول $R2(A, B, C, D, H)$ در BCNF قرار دارد.

هم‌چنین با توجه به وابستگی‌های مطرح شده برای جدول $R3(E, F, G)$ داریم:

$$A \rightarrow IEG$$

$$AB \rightarrow IECD$$

$$ABE \rightarrow CDFGH$$

$$AC \rightarrow IEC$$

وابستگی بدیهی، کنار گذاشته می‌شود $CDE \rightarrow DE$

$$E \rightarrow FG$$

$$F \rightarrow E$$

$$EFG - IEGCDFH = \text{تهی}$$

قانون چهارم ارسطو

هرگاه عضو کلید کاندید حاصل از تفاضل قانون اول (روش اول یا دوم)، تهی گردد، بدین معنی است که، جدول فوق چندین کلید کاندید دارد، که هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید آن وجود ندارد. بنابراین باید کلید کاندید با بررسی دقیق بر روی مجموعه وابستگی کشف گردد.

با کمی دقت و بررسی پُر واضح است که براساس وابستگی‌های مطرح شده، در جدول $R3(\underline{E}, \underline{F}, G)$ صفت E و F کلید کاندید هستند. در واقع جدول $R3$ دو کلید کاندید دارد.

$$\{E\}^+ = \{E, F, G\}$$

براساس بستار فوق صفت E ، همه ستون‌های جدول $R3$ را تولید می‌کند، پس صفت E کلید کاندید است.

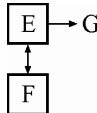
$$\{F\}^+ = \{F, E, G\}$$

براساس بستار فوق صفت F ، همه ستون‌های جدول $R3$ را تولید می‌کند، پس صفت F کلید کاندید است.

توجه: همان‌طور که مشاهده می‌شود، مطابق قانون چهارم ارسطو، هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید فوق وجود ندارد.

بر اساس وابستگی های مطرح شده جدول $R3(\underline{E}, F, G)$ با دو کلید کاندید E و F، فاقد هر گونه وابستگی بخشی (وابستگی غیرکلید به عضو کلید کاندید) و انتقالی (وابستگی غیرکلید به غیرکلید) است. بنابراین این جدول در فرم نرمال 3NF قرار دارد.

بنابراین نمودار وابستگی های تابعی جدول $R3(\underline{E}, F, G)$ به صورت زیر خواهد بود.



همانطور که مشاهده می شود سمت چپ تمام وابستگی های تابعی جدول $R3(\underline{E}, F, G)$ ابرکلید است، بنابراین جدول $R3(\underline{E}, F, G)$ در BCNF هم قرار دارد.

۱۳- گزینه (۱) صحیح است.

زمانی برای جداول یک بانک اطلاعاتی کلید اصلی تعیین می گردد که جداول یک بانک اطلاعاتی با هدف کاهش افزونگی های طبیعی تا حد BCNF، مقابله با بی نظمی ها و ناسازگاری ها و مقادیر تهی نرمال سازی شده باشند.

پس در طراحی بانک اطلاعاتی رابطه ای، پدیده نرمال سازی (روال تجزیه یا کاهش) در اولویت اول قرار دارد.

شرایط تجزیه مطلوب به صورت زیر است:

- ۱- شرط لازم (Nonloss join): دو جدول الحاق پذیر باشند (صفت مشترک در دو جدول، حداقل در یکی کلید کاندید باشد) این شرط، مهم ترین شرط نرمال سازی است، زیرا شرط لازم است.
 - ۲- شرط کافی (حفظ وابستگی های تابعی): وابستگی های تابعی جدول پایه قابل استنتاج از روی وابستگی های تابعی جداول حاصل از تجزیه باشد یا در آنها موجود باشد.
- در ادامه پس از نرمال سازی جداول یک بانک اطلاعاتی، برای هر یک از جداول، یک کلید اصلی توسط طراح بانک اطلاعات، مشخص می گردد.

۱۴- گزینه (۴) صحیح است.

عملیات نرمال سازی یک عملیات چند مرحله ای است که با طی آنها اهداف نرمال سازی به تدریج حاصل می شود. نخستین مرحله نرمال سازی تبدیل جداول به نرمال فرم اول است. برای قرار گرفتن یک جدول در نرمال فرم اول، آن جدول باید در شروط خاصی صدق کند. یک جدول (رابطه) برای برقرار گرفتن در نرمال فرم اول باید دارای شروط زیر باشد: (حفظ قوانین جامعیت درون رابطه ای)

- ۱- دارای حداقل یک کلید کاندید باشد.
 - ۲- جدول تو جدول یا جدول تو در تو نباشد (یعنی جدول فاقد صفت مرکب و چند مقداری باشد)
 - ۱-۲- همه صفات آن غیرقابل تجزیه (Atomic) باشند. (جدول باید فاقد صفت مرکب باشد)
 - ۲-۲- همه صفات جدول تک مقداری باشند. (جدول باید فاقد صفت چند مقداری باشد).
- بنابراین گزینه های اول تا سوم درست هستند.
- به طور کلی می توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:
- جدول باید در نرمال فرم دوم باشد.
 - جدول باید فاقد وابستگی انتقالی (تراگذری) باشد.

وابستگی انتقالی (تراگذری): وابستگی یک مؤلفه غیر کلیدی به یک مؤلفه غیر کلیدی دیگر را وابستگی انتقالی می‌نامند.

بنابراین گزینه چهارم ارتباطی به نرمال فرم اول ندارد و پاسخ این سؤال خواهد بود.

۱۵- گزینه (۱) صحیح است.

جدول studclg با مقادیر آن به شکل زیر را در نظر بگیرید:

S#	Sname	Clg#	Clgname
8621	Ali	12	Computer
8442	Reza	12	Computer
8731	Abbas	NULL	NULL

افزونگی

در جدول studclg:

- این اطلاعات که «کد دانشکده‌ی کامپیوتر برابر 12 است» در سطرهاى اول و دوم تکرار شده است، پس **افزونگی داده** وجود دارد.
- اگر شخصی بخواهد کد دانشکده‌ی کامپیوتر را به 22 تغییر دهد و به اشتباه، فقط مقدار clg# در سطر اول جدول studclg را عوض کند، از این به بعد دانشکده کامپیوتر دو کد خواهد داشت: 12 و 22. این موضوع یعنی **بی‌نظمی یا آنومالی** به معنی نقض جامعیت (صحت و سازگاری) در بانک اطلاعات است.
- در سطر آخر از این جدول، مقادیر NULL وجود دارد. به طور کلی، به ازای هر سطر از جدول studclg که clg# آن برابر NULL باشد یک سطر در جدول studclg خواهیم داشت که مقادیر دو ستون آخر آن NULL خواهد بود. پس مشکل مقادیر NULL نیز وجود دارد. حالا اگر جدول studclg را به دو جدول clg و stud به شکل زیر تجزیه و نرمال کنیم:

S#	Sname	Clg#	Clg#	Clgname
8621	Ali	12	12	Computer
8442	Reza	12		
8731	Abbas	NULL		

جدول Clg

جدول Stud

مشکلات فوق مرتفع و نتایج زیر حاصل خواهد شد.

- کاهش افزونگی طبیعی (محتوایی) مانند تکرار بی‌دلیل سطر (12, computer)
- افزایش افزونگی تکنیکی به دلیل تعریف کلید خارجی (در قالب ستون‌های مشترک) بنابراین کاهش بعضی از انواع افزونگی، طبیعی کاهش و تکنیکی افزایش می‌یابد. بنابراین گزینه سوم درست است.

جداول تجزیه شده را مجدداً در نظر بگیرید:

S#	Sname	Clg#	Clg#	Clgname
8621	Ali	12	12	Computer
8442	Reza	12		
8731	Abbas	NULL		

جدول Clg

جدول Stud

حال پرس وجوی زیر را در نظر بگیرید:
شماره دانشجویی همه دانشجویان را ارائه دهید:

```
select S#
from stud
```

پرواضح است که این پرس وجو نیازی به الحاق دو جدول stud و clg ندارد. بنابراین این پرس وجو سربار حاصل از الحاق دو جدول را ندارد. حال اگر پرس وجوی فوق را بر روی جدول studclg در حالت غیر نرمال اجرا کنیم. از آنجا که طول رکورد در جدول studclg نسبت به جدول stud بزرگتر است، واضح است که سرعت اجرای پرس وجوی فوق بر روی جدول stud افزایش می یابد، به بیان دیگر زمان اجرای پرس وجوی فوق بر روی جدول stud کاهش می یابد. مطابق رابطه ی زیر:

$$\text{طول رکورد} (L) = \text{نرخ انتقال} (R) \times \text{زمان انتقال رکورد} (T_p)$$

حال پرس وجوی صفحه بعد را در نظر بگیرید:
مشخصات کامل همه دانشجویان دانشکده کامپیوتر را ارائه دهید:

```
select *
from stud, clg
where stud.clg# = clg.clg# AND clgname= 'computer'
```

واضح است که این پرس وجو و جو نیاز به الحاق دو جدول stud و clg دارد. بنابراین این پرس وجو سربار حاصل از الحاق دو جدول را دارد. در حالی که اگر پرس وجوی فوق بر روی جدول studclg در حالت غیر نرمال به شکل زیر اجرا می گردید.

```
select *
from studclg
where clgname= 'computer'
```

نیاز به عمل الحاق و سربار حاصل از آن نبود.

توجه: طول رکورد در پرس وجوی اخیر، در هر دو حالت جدول نرمال نشده و جداول نرمال شده برابر است، اما در حالت جداول نرمال شده، برای داشتن تمام ستون ها، نیاز به یک عمل اضافه و دارای سربار به نام الحاق وجود دارد.

بنابراین از آنجا که نرمال سازی باعث تجزیه جداول می شود، گاهی اوقات برای بازیابی نیاز به پیوند مجدد این جداول داریم که باعث افزایش سربار سیستم می گردد، البته نه همیشه مانند پرس وجوی قبل که نیاز به الحاق نداشت.

بنابراین گزینه اول نادرست است، زیرا نرمال سازی رابطه ها گاهی موجب افزایش سربار سیستم در عمل بازیابی به دلیل نیاز به الحاق جداول می شود و در نتیجه سرعت پاسخگویی به پرس وجوها را کاهش می دهد. بنابراین افزایش یا کاهش سرعت پرس وجوها بر روی جداول نرمال شده، بستگی به نوع پرس وجوها دارد.

نرمال سازی تا سطح 3NF و در صورت نیاز تا سطح BCNF اجباری است بنابراین گزینه دوم هم درست است.

همچنین حل مشکل مقادیر پوچ (NULL)، یکی دیگر از اهداف نرمال‌سازی است. بنابراین گزینه چهارم هم درست است.

و این که حل پدیده آنومالی (بی‌نظمی و ناسازگاری) یکی دیگر از اهداف نرمال‌سازی است.

۱۶- گزینه (۲) صحیح است.

طبق تعریف هر جدولی که در نرمال فرم دوم باشد، حتماً در نرمال فرم اول نیز قرار دارد (گزاره اول درست است)، اما عکس این مطلب همواره برقرار نیست. (گزاره دوم نادرست است)، همچنین طبق تعریف هر جدولی که در BCNF باشد. حتماً در نرمال فرم سوم نیز قرار دارد (گزاره سوم درست است)، اما عکس این مطلب همواره برقرار نیست (گزاره چهارم نادرست است)

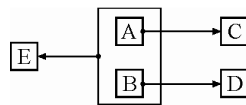
نرمال‌سازی مانند درجات علمی است.

دیپلم، فوق دیپلم، لیسانس، فوق لیسانس، دکتری، فوق دکتری

1NF, 2NF, 3NF, 4NF, BCNF, 5NF

۱۷- گزینه (۴) صحیح است.

شکل زیر، نمودار وابستگی‌های تابعی صفات را نشان می‌دهد. پُر واضح است که کلید این جدول صفات AB است. بنابراین بدیهی است که جدول 2NF نیست به دلیل وجود وابستگی‌های بخشی $A \rightarrow C$ و $B \rightarrow D$ (وابستگی غیرکلید به عضو کلید کاندید) پس جدول مذکور 3NF نیز نیست. پس سه گزینه اول نادرست است. بنابراین گزینه چهارم درست است.



به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیربدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E)$ داریم:

$$A \rightarrow C$$

$$B \rightarrow D$$

$$AB \rightarrow E$$

$$ABCDE - CDE = AB$$

بنابر رابطه فوق صفت AB حتماً باید عضو کلید کاندید باشد.
بستار صفت AB به صورت زیر است:

$$\{AB\}^+ = \{A, B, C, D, E\}$$

براساس بستار فوق، صفات AB، همه ستون‌ها را تولید می‌کند، پس صفات AB، کلید کاندید می‌باشد.

قانون دوم ارسطو

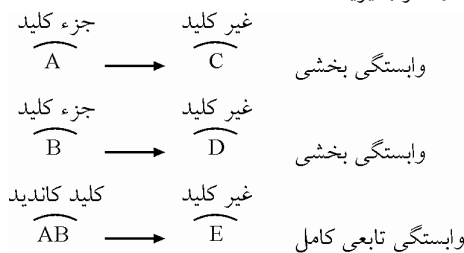
هرگاه عضو کلید کاندید حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

□ جدول باید در نرمال فرم اول باشد. (که هست، زیرا صفت چند مقداری و مرکب ندارد).

□ جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مؤلفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می‌نامند.
وابستگی‌های مطرح شده را در نظر بگیرید:



در وابستگی‌های فوق، وابستگی بخشی $A \rightarrow C$ و $B \rightarrow D$ وجود دارد، بنابراین جدول مربوطه در نرمال فرم دوم قرار ندارد، بنابراین رویه نرمال سازی (تجزیه مطلوب) جهت قرارگیری جدول R در نرمال فرم دوم توسط عمل تجزیه جدول R می‌بایست انجام گردد.

شرایط تجزیه مطلوب به صورت زیر است:

۱- شرط لازم: دو جدول الحاق‌پذیر باشند (صفت مشترک در دو جدول، حداقل در یکی کلید کاندید باشد)

شرط الحاق‌پذیری به دو شکل زیر انجام می‌گردد:

۱-۱- برای حذف وابستگی بخشی از تئوری هیث استفاده می‌گردد.

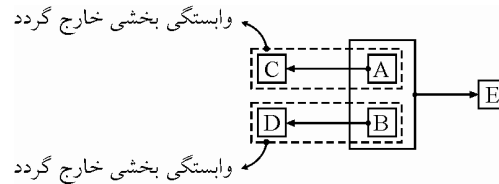
تئوری هیث: این تئوری می‌گوید، وابستگی بخشی (وابستگی غیرکلید به جز کلید کاندید) را از جدول پایه خارج کنید.

۱-۲- برای حذف وابستگی انتقالی از تئوری ریسانن استفاده می‌گردد.

تئوری ریسانن: این تئوری می‌گوید، وابستگی انتقالی (وابستگی غیرکلید به غیرکلید) را از جدول پایه خارج کنید.

۲- شرط کافی: وابستگی‌های تابعی جدول پایه قابل استنتاج از روی وابستگی‌های تابعی جداول حاصل از تجزیه باشد یا در آن‌ها موجود باشد.

با توجه به نمودار وابستگی‌های جدول $R(A, B, C, D, E)$ داریم:



بنابراین، پس از انجام مرحله نرمال‌سازی فرم دوم داریم:

$R1(A, B, E)$	$R2(B, D)$	$R3(A, C)$
$AB \rightarrow E$	$B \rightarrow D$	$A \rightarrow C$

توجه: در این تجزیه شرط کافی هم برقرار است، زیرا وابستگی‌های تابعی جدول پایه، قابل استنتاج از روی وابستگی‌های تابعی جداول حاصل از تجزیه می‌باشد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

□ جدول باید در نرمال فرم دوم باشد (که هست، زیرا وابستگی‌های بخشی از جدول پایه خارج گردید).

□ جدول باید فاقد وابستگی انتقالی باشد.

وابستگی انتقالی: وابستگی یک مولفه غیرکلیدی به یک مولفه غیرکلیدی دیگر را وابستگی انتقالی می‌نامند. همان‌طور که مشاهده می‌شود در هیچ یک از وابستگی‌های تابعی جداول $R1, R2$ و $R3$ وابستگی انتقالی وجود ندارد. بنابراین این پایگاه داده در نرمال فرم سوم هم قرار دارد. هم‌چنین همان‌طور که واضح است، سمت چپ تمام وابستگی‌های تابعی در هر سه جدول ابرکلید است و بنابراین هر سه جدول در BCNF قرار دارند، بنابراین این پایگاه داده در BCNF قرار دارد.

۱۸- گزینه (۴) صحیح است.

طبق تعریف رابطه‌ای که در BCNF است، حتماً در 3NF است و رابطه‌ای که در 3NF است حتماً در 2NF نیز واقع است. هم‌چنین رابطه‌ای که در 2NF است حتماً در 1NF هم قرار دارد. در نتیجه می‌توان گفت که رابطه‌ای که در BCNF است حتماً در نرمال فرم‌های اول، دوم و سوم هم قرار دارد. اما رابطه‌ای که در BNCF است ممکن است در 4NF باشد و البته ممکن است در 4NF نباشد.

۱۹- گزینه (۱) صحیح است.

گزینه‌های دوم تا چهارم به صورت مشخص جزء اهداف فرآیند نرمال‌سازی محسوب می‌شوند. گزینه اول ارتباطی به بحث نرمال‌سازی ندارد و به عنوان یکی از اهداف اصلی نرمال‌سازی محسوب نمی‌شود. زیرا اهداف نرمال‌سازی کاهش افزونگی داده‌ها (کاهش افزونگی طبیعی یا محتوایی)، مقابله با بی‌نظمی، آنومالی و ناسازگاری و سهولت در برقراری قوانین جامعیت (صحت و سازگاری) و کاهش مقادیر تهی (NULL) است. اینها اهداف نرمال‌سازی هستند، اما در این مسیر و رسیدن به این اهداف لازم است برخی وابستگی‌های افزونگی ساز، مثل وابستگی بخشی (وابستگی صفت غیرکلید به جزئی از کلید کاندید) و وابستگی انتقالی (وابستگی صفت غیرکلید به صفت غیرکلید) حذف گردند، که این مسیر است و نه هدف. پس از آن‌که نرمال‌سازی با اهداف مذکور انجام شد، پیاده‌سازی دید کاربر هم در فاز پیاده‌سازی انجام می‌گردد. نرمال‌سازی و کاهش افزونگی‌ها فدای راحتی برای پیاده‌سازی دید کاربر نمی‌گردد. البته متذکر

می‌شویم که تسهیل پیاده‌سازی دید کاربر هیچ جایگاهی در اهداف نرمال سازی ندارد، نرمال‌سازی که انجام شد، پیاده‌سازی دید کاربر هم براساس نیازها و سلیقه‌ی کاربر انجام خواهد شد.

۲۰- گزینه (۴) صحیح است.

بر طبق تعریف، جدولی در BCNF است که ستون‌های آن فقط به کلیدهای کاندیدش وابستگی تابعی داشته باشند، اما در این جدول، وابستگی $Code \rightarrow S\#$ یک وابستگی به ستون‌های غیرکلیدی است.

۲۱- گزینه (۴) صحیح است.

برای مشاهده‌ی مشکلات، به مثال ساده‌ی زیر توجه کنید:

فرض کنید بانک اطلاعات شامل دو جدول stud و clg به صورت زیر باشد:

Stud			Clg	
S#	Sname	clg#	clg#	clgName
8621	Ali	12	12	Computer
8442	Reza	12		
8731	Abbas	NULL		

اگر اطلاعات دو جدول فوق را در یک جدول بریزیم جدول زیر به دست می‌آید:

studclg			
S#	Sname	clg#	Clgname
8621	Ali	12	Computer
8442	Reza	12	Computer
8731	Abbas	NULL	NULL

در جدول studclg :

- این اطلاعات که «کد دانشکده‌ی کامپیوتر برابر 12 است» در سطرهای اول و دوم تکرار شده است، پس افزونگی داده وجود دارد.
- اگر شخصی بخواهد کد دانشکده‌ی کامپیوتر را به 22 تغییر دهد و به اشتباه، فقط مقدار clg# در سطر اول جدول studclg را عوض کند، از این به بعد دانشکده‌ی کامپیوتر دو کد خواهد داشت: 12 و 22. این موضوع یعنی بی‌نظمی در بانک اطلاعات.
- در سطر آخر از این جدول، مقادیر NULL وجود دارد. به طور کلی، به ازای هر سطر از جدول stud که clg# آن برابر NULL باشد یک سطر در جدول studclg خواهیم داشت که مقادیر دو ستون آخر آن NULL خواهد بود. پس مشکل مقادیر NULL نیز وجود دارد.

۲۲- گزینه (۴) صحیح است.

اگر نرمال‌سازی صورت نگیرد و تمام اطلاعات در یک جدول ذخیره شود مشکلات افزونگی داده‌ها، وجود وابستگی‌های تابعی و دشوار بودن کنترل صحت و جامعیت اطلاعات وجود دارد اما مزیت آن این است که در هنگام بازیابی اطلاعات، نیازی به پیوند زدن چند جدول نیست. در طرف مقابل، با انجام نرمال‌سازی، اطلاعات در چند جدول تقسیم می‌شوند و مشکلات فوق برطرف می‌شوند. اما برای بازیابی اطلاعات،

معمولاً لازم خواهد بود چند جدول را با هم پیوند بزنیم، از آنجایی که پیوند زدن جداول عمل زمانگیری است، این عمل باعث ایجاد سربار و در نتیجه کند شدن بازیابی می شود.

۲۳- گزینه (۳) صحیح است.

با کمی بررسی می توان دید که PY کلید اصلی است. رابطه‌ای در فرم $2NF$ است که اگر و فقط اگر $1NF$ باشد و هر صفت غیرکلید با جزئی از کلید اصلی وابستگی تابعی نداشته باشد. از آنجا که وابستگی $Y \rightarrow Z$ یک وابستگی به زیر مجموعه‌ای از کلید اصلی (یعنی Y) است. پس این رابطه در $2NF$ نیست، بنابراین در $3NF$ و $BCNF$ هم نیست.

۲۴- گزینه (۳) صحیح است.

گزینه اول عبارت درستی است: دو عبارت زیر را در نظر بگیرید: (x یک مقدار و S یک مجموعه است):

$$x \text{ not in } S, \quad x \neq \text{all}(S)$$

عبارت سمت راست می گوید مقدار x با هیچ یک از اعضای مجموعه‌ی S برابر نیست و عبارت سمت چپ می گوید مقدار x در مجموعه‌ی S نیست. این دو عبارت با هم یکسانند.

گزینه دوم عبارت درستی است: اگر رابطه‌ای در $BCNF$ باشد صفت‌های آن فقط به کلیدهای کاندیدش وابستگی تابعی دارند، یعنی به صفت‌های غیرکلید وابستگی ندارند و این یعنی وابستگی انتقالی وجود ندارد، مطمئناً با به دست آوردن مجموعه‌ی پوششی مربوطه (F^+) یا با به دست آوردن هر مجموعه‌ی وابستگی دیگری معادل با مجموعه‌ی وابستگی اولیه، هیچ وابستگی انتقالی به وجود نمی آید. پس رابطه کماکان در $BCNF$ خواهد بود.

گزینه سوم عبارت نادرستی است: اگر $A \rightarrow B$ و $A \rightarrow C$ و آن گاه $A \rightarrow BC$.

گزینه چهارم نیز عبارت درستی است: دو مجموعه‌ی وابستگی هم ارزند اگر مجموعه‌ی پوششی آن‌ها برابر باشد. چون به دست آوردن مجموعه‌ی پوششی زمان بر است عدم هم ارزی F_1 و F_2 را با یک مثال نشان می دهیم:

وابستگی $EC \rightarrow B$ در F_2 به هیچ صورتی از وابستگی‌های موجود در F_1 قابل استنتاج نیست، چون صفت B در هیچ یک از وابستگی‌های F_1 در سمت راست " \rightarrow " نیامده است.

۲۵- گزینه (۴) صحیح است.

شرط $(\exists x \subseteq R): x \rightarrow R \notin F^+$ به این معنی است که اگر، x ای (متعلق به مجموعه صفات رابطه R) موجود باشد که نتوان از آن توسط تمام وابستگی‌های تابعی رابطه R ، تمام صفات رابطه R را به دست آورد، این بدین معنی خواهد بود که x کلید نیست.

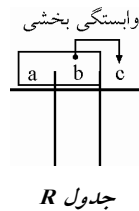
حال اگر این x کلید نباشد ولی با x بتوان به برخی صفات دیگر دسترسی پیدا کرد، این چه معنی خواهد داشت، واضح است که این معنی استنباط می شود که در دل این x صفاتی قرار دارند که وابستگی بخشی یا وابستگی انتقالی و یا وابستگی معکوس، ایجاد کرده‌اند.

عبارت $A \in (x^+ - x)$ ، بدین معنی است که، صفتی مانند A وجود داشته باشد، که از مجموعه صفات غیر کلیدی x قابل دستیابی باشد، یعنی با صفاتی از x بتوان به صفت A رسید. این بدین معنی است که صفت A به مجموعه صفات غیر کلید x وابسته است و این یعنی وجود وابستگی بخشی یا وابستگی انتقالی و یا وابستگی معکوس در رابطه R .

بنابراین اگر حاصل تفاضل $(x^+ - x)$ تهی گردد، نتیجه این است که در رابطه R وابستگی بخشی، انتقالی و معکوس وجود ندارد و اگر حاصل تفاضل $(x^+ - x)$ غیر تهی گردد، نتیجه این است که در رابطه R وابستگی بخشی یا انتقالی یا معکوس وجود دارد.

بنابراین صفت ناهنجار A ، زمانی در رابطه R قرار دارد که حاصل تفاضل $(x^+ - x)$ غیر تهی گردد. یعنی در رابطه R وابستگی بخشی یا انتقالی یا معکوس وجود داشته باشد.

مثال: وابستگی بخشی.



وابستگی بخشی: وابستگی یک مؤلفه غیر کلیدی به جزئی از کلید کاندید را وابستگی بخشی می‌نامند. توجه: در جدول R ، صفات ab کلید کاندید است و صفت c با صفت b وابستگی بخشی دارد. بنابراین اگر مقدار x را برابر b در نظر بگیریم (صفت b کلید نیست، پس شرط $x \rightarrow R \notin F^+$ برقرار است)، با توجه به وابستگی بخشی $b \rightarrow c$ ، آنگاه خواهیم داشت:

$$x = b$$

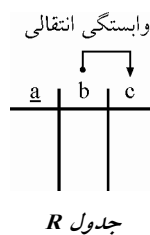
$$\{x\}^+ = \{b\}^+ = \{b, c\}$$

$$A \in (x^+ - x) \rightarrow (\{b, c\} - \{b\}) = \{c\}$$

حاصل تفاضل $(x^+ - x)$ غیر تهی شد. چون در رابطه R وابستگی بخشی وجود دارد. پس صفت c یک صفت ناهنجار است، زیرا از صفت b عضو کلید کاندید می‌توان به آن رسید.

توجه: صفت c یک صفت غیرکلید ناهنجار است.

مثال: وابستگی انتقالی.



وابستگی انتقالی: وابستگی یک مؤلفه غیر کلیدی، به یک مؤلفه غیر کلیدی دیگر را وابستگی انتقالی می‌نامند.

توجه: در جدول R ، صفت a کلید کاندید است و صفت c با صفت b وابستگی انتقالی دارد. بنابراین اگر مقدار x را برابر b در نظر بگیریم (صفت b کلید نیست، پس شرط $x \rightarrow R \notin F^+$ برقرار است)، با توجه به وابستگی انتقالی $b \rightarrow c$ ، آنگاه خواهیم داشت:

$$x = b$$

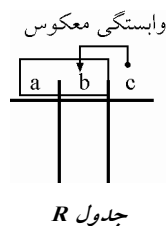
$$\{x\}^+ = \{b\}^+ = \{b, c\}$$

$$A \in (x^+ - x) \rightarrow (\{b, c\} - \{b\}) = \{c\}$$

حاصل تفاضل $(x^+ - x)$ غیر تهی شد. چون در رابطه R وابستگی انتقالی وجود دارد. پس صفت c یک صفت ناهنجار است، زیرا از صفت b غیر کلید می‌توان به آن رسید.

توجه: صفت c یک صفت غیر کلید ناهنجار است.

مثال: وابستگی معکوس.



وابستگی معکوس: وابستگی یک عضو کلید کاندید به عضو یک کلید کاندید دیگر یا مؤلفه غیر کلیدی را، وابستگی معکوس می‌نامند.

عضو کلید کاندید \rightarrow عضو کلید کاندید

عضو کلید کاندید \rightarrow غیر کلید

توجه: در جدول R ، صفات ab کلید کاندید است و صفت b با صفت c وابستگی معکوس دارد.

بنابراین اگر مقدار x را برابر c در نظر بگیریم (صفت c کلید نیست، پس شرط $x \rightarrow R \notin F^+$ برقرار است)، با توجه به وابستگی معکوس $c \rightarrow b$ ، آنگاه خواهیم داشت:

$$x = b$$

$$\{x\}^+ = \{c\}^+ = \{c, b\}$$

$$A \in (x^+ - x) \rightarrow (\{c, b\} - \{c\}) = \{b\}$$

حاصل تفاضل فوق $(x^+ - x)$ غیر تهی شد. چون در رابطه R وابستگی معکوس وجود دارد. پس صفت b یک صفت ناهنجار است، زیرا از صفت c غیر کلید می‌توان به آن رسید.

توجه مهم: صفت b یک صفت عضو کلید کاندید ناهنجار است.

از آنجا که در جدول R ، صفت چند مقداری و مرکب وجود ندارد، بنابراین این جدول در نرمال فرم اول قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

جدول باید در نرمال فرم اول باشد.

جدول باید فاقد وابستگی بخشی باشد.

گزینه اول درست است.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

جدول باید در نرمال فرم دوم باشد.

□ جدول باید فاقد وابستگی انتقالی باشد.

مطابق گزینه اول، اگر رابطه R در 3NF باشد، آنگاه وابستگی بخشی و وابستگی انتقالی نخواهد داشت، پس هیچ صفت غیرکلیدی ناهنجاری را هم نخواهد داشت. هر چند رابطه R وابستگی بخشی و انتقالی ندارد، اما ممکن است وابستگی معکوس را داشته باشد، یعنی رابطه در نرمال فرم سوم باشد اما در BCNF نباشد. که در این حالت رابطه R دارای صفت عضو کلید کاندید ناهنجار است.

گزینه اول را مجدداً بخوانید «اگر رابطه R در 3NF باشد، آنگاه هیچ صفت غیرکلیدی در رابطه R ناهنجار نخواهد بود»، این گزینه درست است. زیرا امکان ندارد صفات غیرکلیدی در ناهنجار باشد، به دلیل عدم وابستگی بخشی و انتقالی، هر چند ممکن است صفت عضو کلید کاندید ناهنجار داشته باشد، به دلیل وجود وابستگی معکوس، اما گزینه گفته است هیچ صفت غیر کلیدی در رابطه R ناهنجار نخواهد بود، که درست است.

گزینه دوم درست است.

به طور کلی می توان شروط قرار داشتن یک جدول در BCNF را به صورت زیر بیان کرد:

□ جدول باید در نرمال فرم سوم باشد.

□ جدول باید فاقد وابستگی معکوس باشد.

توجه: در یک نگاه دیگر برای BCNF می توان گفت، جدولی در BCNF قرار دارد که همه شروع های وابستگی ها، ابرکلید باشد. به بیان دیگر هر گاه سمت چپ همه وابستگی ها، ابرکلید باشد، آن گاه آن جدول در BCNF قرار دارد.

مطابق گزینه دوم، اگر رابطه R در BCNF باشد، آنگاه وابستگی بخشی، وابستگی انتقالی و وابستگی معکوس نخواهد داشت، پس هیچ صفت ناهنجاری (اعم از صفت غیرکلید ناهنجار حاصل از وابستگی بخشی و وابستگی انتقالی به دلیل عدم وجود وابستگی بخشی و انتقالی و اعم از صفت عضو کلید کاندید ناهنجار حاصل از وابستگی معکوس به دلیل عدم وجود وابستگی معکوس) نخواهد داشت.

گزینه سوم درست است.

مطابق گزینه سوم اگر هیچ صفت ناهنجاری در R وجود نداشته باشد، (اعم از صفت غیر کلید ناهنجار حاصل از وابستگی بخشی و وابستگی انتقالی به دلیل عدم وجود وابستگی بخشی و انتقالی و اعم از صفت عضو کلید کاندید ناهنجار حاصل از وابستگی معکوس به دلیل عدم وجود وابستگی معکوس) این نتیجه حاصل خواهد شد که رابطه R در BCNF قرار دارد.

۲۶- گزینه (۱) صحیح است.

با کمی بررسی می توان دید که کلید کاندید در S برابر است با BE، برای این که جدول S در BCNF باشد باید تمام صفت های آن فقط به کلید کاندید وابستگی تابعی داشته باشند، گزینه های دوم و سوم وابستگی تابعی به غیرکلید کاندید را نشان می دهند. وابستگی تابعی گزینه ی چهارم نیز اصلاً وجود ندارد اما وابستگی تابعی گزینه ی اول یک وابستگی تابعی به کلید کاندید است.

۲۷- گزینه (۱) صحیح است.

در هنگام نرمال سازی جداول، برای این که جدول R در نرمال فرم باشد باید ابتدا برای آن، کلیدهای کاندید را به دست آورد و یکی از کلیدهای کاندید را به عنوان کلید اصلی در نظر گرفت. دو نکته در مورد کلید اصلی وجود دارد:

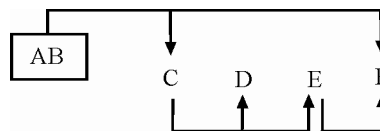
(۱) تمام فیلدهای دیگر به کلید اصلی وابستگی تابعی دارند.
 (۲) حتی پس از انجام مراحل نرمال‌سازی (3NF, 2NF, ...) می‌توان از روی کلید اصلی (که در یکی از جداول حاصل از نرمال‌سازی قرار دارد) به فیلدهای دیگر (که در جداول مختلف حاصل از نرمال‌سازی هستند) دسترسی پیدا کرد، این دسترسی از طریق کلیدهای خارجی صورت می‌گیرد.
 حال به جدول R3 توجه کنید. فیلدهای آن (یعنی D و E) در هیچ جدول دیگری وجود ندارد. بنابراین بین جدول R3 و جداول دیگر هیچ ارتباطی وجود ندارد. پس نه از صفت‌های DE می‌توان صفات دیگر را پیدا کرد و نه از صفت‌های دیگر می‌توان صفات‌های DE را یافت. پس نرمال‌سازی انجام شده قطعاً اشتباه است. چون هیچ کلید اصلی‌ای نمی‌توان برای رابطه‌ی اولیه‌ی R متصور شد.

۲۸- گزینه (۲) صحیح است.

گزینه‌ی اول صحیح است، با انجام نرمال‌سازی، افزونگی‌ها عموماً حذف می‌شوند و عموماً حجم جداول کوچک‌تر می‌شود، بنابراین جستجوها با سرعت بیشتری صورت می‌پذیرد.
 گزینه‌ی دوم غلط است، فرآیند نرمال‌سازی به اطلاعات درون جداول کاری ندارد و فقط براساس وابستگی‌های تعریف شده روی ستون‌های جداول عمل می‌کند پس NULL بودن مقادیر تأثیری بر فرآیند نرمال‌سازی ندارد.
 گزینه‌ی سوم صحیح است، یکی از اهداف نرمال‌سازی کاهش بی‌نظمی در پایگاه داده است.
 گزینه‌ی چهارم صحیح است، با انجام نرمال‌سازی، عموماً افزونگی حذف و در نتیجه حجم پایگاه داده کم می‌شود.

۲۹- گزینه (۱) صحیح است.

وابستگی‌های ذکر شده در شکل زیر آمده‌اند:

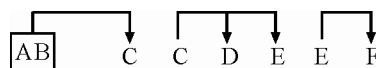


مراحل نرمال‌سازی به شرح زیر است:

تنها کلید کاندید، AB است:

$$\{AB\}^+ = \{A, B, C, D, E, F\}$$

پس آن را به عنوان کلید اصلی در نظر می‌گیریم. تا اینجا جدول R در 1NF هست. چون وابستگی به زیر مجموعه‌ای از کلید اصلی وجود ندارد. پس جدول R در 2NF هم هست.
 برای این که جدول به 3NF برود، باید وابستگی‌های انتقالی را حذف کنیم. با انجام این کار جداول زیر به دست می‌آیند.



جداول فوق نرمال هستند و به نرمال‌سازی بیشتر نیاز ندارند. با توجه به جداول به دست آمده و وابستگی‌های آنها گزینه اول صحیح است.

۳۰- گزینه (۲) صحیح است.

با توجه به شکل می توان دید که AB کلید اصلی است:

$$\{AB\}^+ = \{A, B, C, D, E\}$$

کلید کاندید دیگری نیز وجود ندارد. پس جدول در 1NF هست.

از آنجایی که وابستگی تابعی به زیر مجموعه ای از کلید اصلی وجود ندارد پس جدول در 2NF نیز هست. اما چون وابستگی انتقالی زیر وجود دارد جدول در 3NF نیست:

$$AB \rightarrow E, E \rightarrow C$$

۳۱- گزینه (۲) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه های سمت راست وابستگی های غیربدهی - تمام خصیصه های جدول = عضو کلید کاندید

روش دوم

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

با توجه به وابستگی های مطرح شده برای رابطه $R(A, B, C, D)$ داریم:

$$A \rightarrow B$$

$$B \rightarrow C$$

$$BC \rightarrow A$$

$$\cancel{A} \cancel{B} \cancel{C} D - ABC = D$$

بنابر رابطه فوق صفت D حتماً باید عضو کلید کاندید باشد. بستر صفت D به صورت زیر است:

$$\{D\}^+ = \{D\}$$

براساس بستر فوق، صفت D، فقط ستون D را تولید می کند، پس صفت D فقط عضو کلید کاندید می باشد و کلید کاندید نمی باشد.

قانون سوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول، برخی از ستون ها را تولید کند، بدین معنی است که، جدول مورد نظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفت دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهد داشت.

با کمی دقت و بررسی، پُر واضح است که اگر صفت A در کنار عضو کلید کاندید D قرار بگیرد، همکاری صفات (A و D) می‌تواند، همه ستون‌ها را تولید کند، بنابراین صفات AD کلید کاندید جدول R خواهد بود.

بستار صفات AD به صورت مقابل است:

بر اساس بستار فوق، صفات AD ، همه ستون‌ها را تولید می‌کنند، پس صفات AD ، کلید کاندید است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد.

همچنین از آن جا که $BC \rightarrow A$ و $B \rightarrow C$ ، پس می‌توان ترکیب دو خصیصه (B, D) را هم کلید کاندید دیگری برای این جدول تلقی کرد، دقت کنید که صفت C جزء کلید کاندید نیست، زیرا توسط صفت B قابل دسترسی است.

چون وقتی (A, D) کلید کاندید است و همه ستون‌ها را تولید می‌کند، پس (B, D) هم، کلید کاندید است و همه ستون‌ها را تولید می‌کند، زیرا در نهایت طبق وابستگی $BC \rightarrow A$ و $B \rightarrow C$ ، صفت B ، صفت A را می‌دهد و (B, D) به (A, D) تبدیل می‌گردد.

بستار صفات (B, D) به صورت مقابل است:

بر اساس بستار فوق، صفات (B, D) ، همه ستون‌ها را تولید می‌کنند، پس صفات (B, D) کلید کاندید است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده، در مجموع، دارای دو کلید کاندید است که مطابق قانون سوم ارسطو، عضو کلید کاندید D در بین هر دو کلید کاندید به طور مشترک قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم اول را به صورت زیر بیان کرد:

دارای حداقل یک کلید کاندید باشد.

همه خصیصه‌های آن غیرقابل تجزیه باشند (جدول باید فاقد خصیصه‌های مرکب باشد)

همه خصیصه‌های آن تک مقداری باشند (جدول باید فاقد خصیصه‌های چند مقداری باشد)

واضح است که جدول مطرح شده در نرمال فرم اول قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

جدول باید در نرمال فرم اول باشد.

جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مولفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می‌نامند.

مولفه غیرکلیدی: هر صفتی که عضو هیچ کلید کاندید نباشد، به عنوان مولفه غیرکلیدی نامیده می‌شود.

مولفه جزء کلید کاندید: هر صفتی که عضو حداقل یک کلید کاندید باشد، به عنوان مولفه جزء کلید کاندید نامیده می‌شود.

وابستگی‌های مطرح شده را در نظر بگیرید:

جزء کلید $A \rightarrow B$

غیرکلید $B \rightarrow C$ وابستگی بخشی:

$BC \rightarrow A$

در وابستگی های فوق، وابستگی بخشی وجود دارد. بنابراین جدول مربوطه در نرمال فرم دوم قرار ندارد، پس در نرمال فرم سوم، BCNF و نرمال فرم چهارم هم قرار ندارد.

۳۲- گزینه (۴) صحیح است.

۳۳- گزینه (۴) صحیح است.

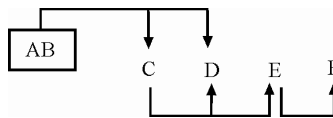
با انجام نرمال سازی، چند جدول به دست می آید که با کلیدهای خارجی به هم وصل اند. برای استخراج اطلاعات از روی چند جدول نیاز است جداول با هم پیوند بخورند. پیوند زدن جداول زمان بر است در نتیجه اگر جداول نرمال نشوند در برخی موارد سرعت پرس و جوها بیشتر خواهد بود. پس گزینه اول صحیح است.

از سوی دیگر، اگر بانک اطلاعاتی نرمال سازی نشود و تمام اطلاعات در یک جدول ذخیره شود، تعداد سطرهای این جدول بسیار زیاد خواهد بود. در نتیجه اگر به دنبال پرس و جوی اطلاعات از یک موجودیت (که در حالت نرمال شده یک جدول می بود) باشیم، باید در تعداد سطرهای زیادی به دنبال آن بگردیم در صورتی که در حالت نرمال شده، تعداد سطرهای جدول مربوط به آن موجودیت بسیار کم تر می بود، پس گزینه دوم نیز صحیح است.

گزینه سوم نیز صحیح است. ممکن است جداولی که در 3NF هستند باز هم افزونگی داشته باشند که در آن صورت از نرمال سازی سطح 4NF یا 5NF یا BCNF استفاده می شود.

۳۴- گزینه (۱) صحیح است.

وابستگی های ذکر شده در سؤال به صورت زیرند:

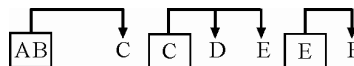


با کمی بررسی می توان دید که AB تنها کلید کاندید و در نتیجه کلید اصلی است:

$$\{A, B\}^+ = \{A, B, C, D, E, F\}$$

پس جدول فوق در 1NF هست.

جدول فوق در 2NF هم هست چون هیچ صفتی به زیر مجموعه ای از کلید اصلی وابسته نیست. ولی جدول فوق در 3NF نیست چون دو وابستگی انتقالی وجود دارد: یکی $\{AB \rightarrow C, C \rightarrow DE\}$ و دیگری $\{C \rightarrow E, E \rightarrow F\}$ با تبدیل جدول فوق به 3NF، جداول زیر به دست می آیند.



جداول به دست آمده دیگر افزونگی ندارند و بنابراین به نرمال سازی بیشتر احتیاج نیست. با توجه به جداول به دست آمده، گزینه اول صحیح است.

۳۵- گزینه (۳) صحیح است.

در هنگام نرمال سازی جداول، برای این که جدول R در نرمال 1NF باشد ابتدا برای آن، کلیدهای کاندید را به دست آورد و یکی از کلیدهای کاندید را به عنوان کلید اصلی در نظر گرفت.

دو نکته در مورد کلید اصلی وجود دارد:

(۱) تمام فیلدهای دیگر به کلید اصلی وابستگی تابعی دارند.
 (۲) حتی پس از انجام مراحل نرمال‌سازی (2NF, 3NF و...) می‌توان از روی کلید اصلی (که در یکی از جداول حاصل از نرمال‌سازی قرار دارد) به فیلدهای دیگر (که در جداول مختلف حاصل از نرمال‌سازی هستند) دسترسی پیدا کرد. این دسترسی از طریق کلیدهای خارجی صورت می‌گیرد.
 حال به جداول R2 توجه کنید. فیلدهای آن (یعنی B و F) در هیچ جدول دیگری وجود ندارد. بنابراین بین جدول R2 و جداول دیگر هیچ ارتباطی وجود ندارد. پس نه از صفت‌های BF می‌توان به صفات دیگر دسترسی داشت و نه از صفت‌های دیگر می‌توان به BF دسترسی یافت. پس نرمال‌سازی انجام شده اشتباه است. چون هیچ کلید اصلی‌ای نمی‌توان برای رابطه اولیه R متصور شد.

۳۶- گزینه (۱) صحیح است.

این بانک اطلاعاتی در BCNF نیست چون صفت‌های آن به صفت‌هایی غیر از کلیدهای کاندید وابستگی تابعی دارند (وابستگی‌های $AB \rightarrow CD$ و $H \rightarrow AB$)
 این بانک اطلاعاتی همچنین در 3NF نیست چون وابستگی انتقالی (یعنی وابستگی به صفت‌های غیرکلیدی) وجود دارد، (مثال همان وابستگی‌های $AB \rightarrow CD$, $H \rightarrow AB$). اما این بانک اطلاعاتی در 2NF هست چون صفت‌های آن به زیر مجموعه‌های کلید اصلی وابستگی تابعی ندارند (کلید اصلی در اینجا EF است چون طبق صورت سؤال، کوتاه‌ترین کلید ممکن را کلید اصلی فرض کرده است). این بانک در 1NF هم هست (چون کلیدهای آن قابل تعیین‌اند و صفت‌ها از دامنه تو در تو نیستند)
 توجه نمایید که طراح پاسخ مناسبی را تعیین نکرده است اگر گزینه اول فقط 2NF می‌بود پاسخ بهتری بود.

۳۷- گزینه (۴) صحیح است.

گزینه اول صحیح است، چون با توجه به وابستگی $A \rightarrow BC$ با داشتن A می‌توان B را به دست آورد. پس با حذف B از وابستگی $AD \rightarrow BG$ اطلاعاتی از بین نمی‌رود.
 گزینه دوم صحیح است، توجه کنید که صفت‌های A و E در سمت راست هیچ یک از وابستگی‌های تابعی نیستند پس از روی صفات دیگر قابل دستیابی نیستند و بنابراین حتماً باید در کلید کاندید حضور داشته باشند. بعلاوه، با داشتن E و A می‌توان تمام صفت‌های دیگر را به دست آورد، پس $\{AE\}$ تنها کلید کمینه است و بنابراین تنها کلید کاندید است.
 گزینه سوم نیز صحیح است، چون با ترکیب وابستگی‌های $A \rightarrow BC$ و $BE \rightarrow G$ می‌توان $AE \rightarrow G$ را نتیجه گرفت. اما گزینه چهارم صحیح نیست چون در رابطه داده شده، وابستگی انتقالی وجود دارد، مثلاً $CD \rightarrow G$ و $AE \rightarrow G$ ، یا $G \rightarrow CD$ و $BE \rightarrow G$ بنابراین این رابطه در 2NF نیست.

۳۸- گزینه (۱) صحیح است.

با توجه به شکل رسم شده، رابطه‌ی R1 یک رابطه‌ی یک به چند بین E1 و E2 را نشان می‌دهد به طوری که هر تاپل از E2 با یک تاپل از E1 مرتبط است و به عبارت مهم‌تر، «هر تاپل E1 مرتبط با چند تاپل از E2 است.» این عبارت بیان‌کننده یک وابستگی چند مقداری بین E1 و E2 است. ما این وابستگی را به صورت $e1 \rightarrow e2$ نشان می‌دهیم که در آن e1 و e2 به ترتیب کلیدهای E1 و E2 هستند.

به طور مشابه، یک وابستگی چند مقداری بین $E1$ و $E3$ به صورت $e1 \rightarrow e3$ وجود دارد. وقتی به جای ارتباطهای $R1$ و $R2$ یک ارتباط سه گانه قرار داده شود و اگر این رابطه سه گانه را R در نظر بگیریم. بنابراین تئوری فاگین (Fagin)، تجزیه‌ای مطلوب است که:

$$\left. \begin{array}{l} R(A,B,C) \\ A \rightarrow B \\ A \rightarrow C \end{array} \right\} \Rightarrow \left. \begin{array}{l} R1(A,B) \\ R2(A,C) \end{array} \right\} 4NF \text{ فرم نرمال}$$

بنابراین با توجه به دو وابستگی چندمقداری $e1 \rightarrow e2$ و $e1 \rightarrow e3$ تجزیه مطلوب در فرم نرمال 4NF به صورت زیر خواهد بود:

$$R1(e1,e2) , R2(e1,e3)$$

باید کلیدهای سه موجودیت $E1$ ، $E2$ و $E3$ به جدول متناظر با این رابطه سه گانه اضافه شود. بنابراین در این جدول، دو وابستگی چند مقداری $e1 \rightarrow e2$ و $e1 \rightarrow e3$ وجود خواهد داشت. برای رفع این وابستگی‌های چند مقداری باید جدول را به 4NF برد که البته برای این کار باید وابستگی‌های تابعی و چند مقداری را دانست.

۳۹- گزینه (۳) صحیح است.

قبل از هر چیز توجه کنید که W, X, Y و Z مجموعه‌هایی از صفات $A1$ تا An هستند، بنابراین از شرط $Z \subseteq W$ در گزاره اول می‌توان نتیجه گرفت که $W \rightarrow Z$ ، یعنی از روی مجموعه صفات W می‌توان صفات موجود در مجموعه Z را نتیجه گرفت. برای مثال اگر $W = \{A1, A2, A3\}$ و $Z = \{A1, A2\}$ باشند (و در نتیجه $Z \subseteq W$ باشد) بدیهی است که داریم:

$$\{A1, A2, A3\} \rightarrow \{A1, A2\} \text{، یعنی از روی صفات } \{A1, A2, A3\} \text{ می‌توان صفات } \{A1, A2\} \text{ را به دست آورد.}$$

با توجه به توضیحات فوق، گزاره اول اگر $X \rightarrow Y$ می‌بود و همچنین داریم $Z \subseteq W$ که از روی آن می‌فهمیم که $W \rightarrow Z$. بنابراین طبق قاعده ترکیب داریم:

$$\left. \begin{array}{l} X \rightarrow Y \\ W \rightarrow Z \end{array} \right\} \Rightarrow WX \rightarrow YZ$$

گزاره دوم اگر $X \rightarrow Y$ می‌بود، از سوی دیگر از $Z \subseteq Y$ در می‌یابیم که $Y \rightarrow Z$ بنابراین طبق قاعده انتقال داریم:

$$\left. \begin{array}{l} X \rightarrow Y \\ Y \rightarrow Z \end{array} \right\} \Rightarrow X \rightarrow Z$$

اما در گزاره اول داریم:

$$X \rightarrow Y , Z \subseteq W \Rightarrow WX \rightarrow YZ$$

مطابق قاعده افزایشی داریم:

$$\left. \begin{array}{l} X \rightarrow Y \\ Z \subseteq W \end{array} \right\} \left. \begin{array}{l} X \rightarrow Y \\ W \rightarrow Z \end{array} \right\} \Rightarrow XW \rightarrow YZ$$

بنابراین گزاره اول عبارت درستی است.

هم‌چنین در گزاره دوم داریم:

$$X \rightarrow Y, W \cap Y, W \rightarrow Z, Z \subseteq Y \Rightarrow X \rightarrow Z$$

مطابق قاعده انعقاد داریم:

$$\left. \begin{array}{l} X \rightarrow Y \\ W \cap Y = \phi \\ W \rightarrow Z \\ Z \subseteq Y \end{array} \right\} X \rightarrow Z$$

بنابراین گزاره دوم هم عبارت درستی است.

۴۰- گزینه (۲) صحیح است.

برای پاسخگویی به این سؤال باید ابتدا مجموعه وابستگی پوششی FD^+ و سپس مجموعه وابستگی بهینه FD_{opt} را به دست آوریم و بعد از آن مراحل نرمال‌سازی را انجام دهیم. با کمی بررسی دیده می‌شود که مجموعه داده شده همان FD^+ است. پس با انجام قواعد لازم، FD_{opt} به صورت زیر به دست می‌آید:

$$FD_{opt} = \{A \rightarrow D, D \rightarrow E, AB \rightarrow C\}$$

با توجه به FD_{opt} مشاهده می‌شود که تنها کلید کاندید AB است. تا اینجا، جدول مربوطه در $1NF$ است. با بردن جداول به $2NF$ (یعنی با حذف وابستگی‌های تابعی به زیر مجموعه‌های کلید اصلی) دو جدول زیر، با وابستگی‌های تابعی نشان داده شده به دست می‌آیند.



جدول 1 را نمی‌توان بیش از این نرمال کرد اما جدول 2 وابستگی انتقالی دارد و باید به $3NF$ برود:



بنابراین سه جدول (1) و (3) و (4) به دست می‌آیند که در $3NF$ هستند و بیش از این قابل نرمال‌سازی نیستند. البته جداول در سطح $BCNF$ نیز می‌باشند.

۴۱- گزینه (۳) صحیح است.

در مورد گزینه اول، با توجه به وابستگی‌های تابعی داده شده، با کمی بررسی می‌توان دید که از پیوند طبیعی سه جدول R_3, R_2, R_1 جدول R به دست می‌آید.

در مورد گزینه دوم، چون کلید اصلی جدول R (یعنی A) در هر سه جدول R_3, R_2, R_1 آمده است از پیوند طبیعی سه جدول مذکور جدول R به دست می‌آید.

اما در مورد گزینه سوم، نمی‌توان با پیوند طبیعی R_3, R_2, R_1 به جدول R رسید.

یکی از شرایط در تجزیه بدون گمشدگی (LossLess) اینست که صفت مشترک در یکی از روابط کلید باشد که گزینه سوم این شرط را دارا نیست.

توجه: اگر سؤال، تجزیه مطلوب را در نظر داشت، گزینه اول، پاسخ می‌بود، زیرا هم الحاق‌پذیر است و هم وابستگی‌های تابعی جدول پایه را حفظ کرده است.

۴۲- گزینه (۴) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

با توجه به وابستگی‌های مطرح شده برای رابطه:

Booking (guestID, guestName, creditCard, roomNo, roomCat, from, to)

داریم:

guestID → questName, creditCard

roomNo → roomCat

roomNo, from → guestID, to

roomNo, to → guestID, from

questID, questName, creditCard, roomNo, roomCat, from, to -

questID, questName, creditCard, roomCat, from, to = roomNo

بنابر رابطه فوق صفت roomNo حتماً باید عضو کلید کاندید باشد.

$$\{\text{roomNo}\}^+ = \{\text{roomNo}, \text{roomCat}\}$$

بستار صفت roomNo به صورت روبه‌رو است:

قانون سوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول، برخی از ستون‌ها را تولید کند، بدین معنی است که، جدول موردنظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت. بنابراین صفت roomCat که توسط عضو کلید کاندید یعنی roomNo تولید می‌گردد، در کلید کاندید جایگاهی نخواهد داشت.

بر اساس بستار فوق، صفت roomNo به عنوان عضو کلید کاندید همه ستون‌ها را تولید نمی‌کند، بنابراین مطابق قانون سوم ارسطو باید صفاتی در کنار صفت roomNo قرار گیرد تا کلید کاندید تولید گردد. این صفات کناری از صفات باقی مانده انتخاب می‌گردند، صفات باقی مانده عبارتند از guestID, guestName, creditCard, from, to البته از این مجموعه صفات guestName, creditCard را هم کنار می‌گذاریم، زیرا کمکی در تولید صفات دیگر نمی‌کنند. چون در سمت چپ هیچ یک از وابستگی‌های تابعی نیامده است. بنابراین با ترکیب صفات from یا to با صفت roomNo کلیدهای کاندید تولید می‌گردند.

$$\{\text{roomNo, from}\}^+ = \{\text{roomNo, from, guestID, to, guestName, creditCard, roomCat}\}$$

$$\{\text{roomNo, to}\}^+ = \{\text{roomNo, to, guestID, from, guestName, creditCard, roomCat}\}$$

پس ترکیب صفات $\{\text{roomNo, from}\}$ و $\{\text{roomNo, to}\}$ کلیدهای کاندید جدول Booking هستند. توجه: دقت کنید که ترکیب صفات $\{\text{roomNo, guestID}\}$ نمی تواند به عنوان کلید کاندید محسوب گردد، چون همه صفات را تولید نمی کند. بستر صفات $\{\text{roomNo, guestID}\}$ به صورت زیر است:

$$\{\text{roomNo, guestID}\}^+ = \{\text{roomNo, guestID, guestName, creditCard, roomCat}\}$$

توجه: کلید اصلی، یکی از کلیدهای کاندید است که بر اساس سلیقه طراح پایگاه داده، انتخاب می شود. مطابق قوانین مدل رابطه ای، هر جدول حتماً باید یک کلید اصلی داشته باشد. همچنین هر جدول فقط می تواند یک کلید اصلی داشته باشد. هر کلید کاندیدی که به عنوان کلید اصلی انتخاب نشده است، یک کلید فرعی است. فرض کنید که ترکیب صفات $\{\text{roomNo, from}\}$ کلید اصلی و ترکیب صفات $\{\text{roomNo, to}\}$ کلید فرعی جدول Booking باشد.

توجه: دقت کنید که هیچ گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده در مجموع دارای دو کلید کاندید است. که مطابق قانون سوم ارسطو، عضو کلید کاندید roomNo ، در بین هر دو کلید کاندید به طور مشترک قرار دارد.

حال یک بار دیگر وابستگی های مطرح شده برای رابطه:

Booking (guestID, guestName, creditCard, roomNo, roomCat, from, to)

را در نظر بگیرید:

وابستگی انتقالی	غیر کلید guestID → questName
وابستگی انتقالی	غیر کلید guestID → creditCard
وابستگی بخشی	غیر کلید roomNo → roomCat
وابستگی کامل	غیر کلید roomNo, from → guestID
وابستگی کامل	عضو کلید کاندید roomNo, from → to
وابستگی کامل	غیر کلید roomNo, to → guestID
وابستگی کامل	عضو کلید کاندید roomNo, to → from

به طور کلی می توان شروط قرار داشتن یک جدول در نرمال فرم اول را به صورت زیر بیان کرد:

دارای حداقل یک کلید کاندید باشد.

همه خصیصه های آن غیر قابل تجزیه باشند (جدول باید فاقد خصیصه های مرکب باشد)

همه خصیصه های آن تک مقداری باشند (جدول باید فاقد خصیصه های چند مقداری باشد)

واضح است که جدول مطرح شده در فرم اول نرمال قرار دارد.

به طور کلی می توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

جدول باید در نرمال فرم اول باشد.

جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مؤلفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می‌نامند. **مؤلفه غیرکلید:** هر صفتی که عضو هیچ کلید کاندیدی نباشد، به عنوان مؤلفه غیرکلیدی نامیده می‌شود. **مؤلفه جزء کلید کاندید:** هر صفتی که عضو حداقل یک کلید کاندید باشد، به عنوان مؤلفه جزء کلید کاندید نامیده می‌شود.

در وابستگی‌های فوق، وابستگی بخشی وجود دارد. بنابراین جدول مربوطه در نرمال فرم دوم قرار ندارد.

وابستگی بخشی
غیرکلید $roomNo \rightarrow roomCat$ عضو کلید کاندید
به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:
 جدول باید در نرمال فرم دوم باشد.
 جدول باید فاقد وابستگی انتقالی باشد.

وابستگی انتقالی: وابستگی یک مؤلفه غیرکلیدی به یک مؤلفه غیرکلیدی دیگر را وابستگی انتقالی می‌نامند. در وابستگی‌های فوق، وابستگی بخشی و انتقالی وجود دارد. بنابراین جدول مربوطه در نرمال فرم سوم هم قرار ندارد.

وابستگی بخشی
غیرکلید $roomNo \rightarrow roomCat$ عضو کلید کاندید
وابستگی انتقالی
غیرکلید $questID \rightarrow questName$
وابستگی انتقالی
غیرکلید $questID \rightarrow creditCard$
به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم BCNF را به صورت زیر بیان کرد:
 جدول باید در نرمال فرم سوم باشد.
 جدول باید فاقد وابستگی معکوس باشد.

وابستگی معکوس: وابستگی یک عضو کلید کاندید به عضو یک کلید کاندید دیگر یا مؤلفه غیرکلیدی را، وابستگی معکوس می‌نامند.

عضو کلید کاندید \rightarrow عضو کلید کاندید

غیرکلید \rightarrow عضو کلید کاندید

در وابستگی‌های فوق، وابستگی معکوس وجود ندارد. اما وابستگی‌های بخشی و انتقالی وجود دارد، یعنی همانطور که گفتیم جدول مربوطه در نرمال فرم دوم و سوم قرار ندارد، بنابراین جدول مربوطه در نرمال فرم BCNF هم قرار ندارد.

در یک نگاه دیگر برای نرمال فرم BCNF می‌توان گفت، جدولی در نرمال فرم BCNF قرار دارد که همه‌ی شروع‌های وابستگی‌ها، ابرکلید باشد. به بیان دیگر هرگاه سمت چپ همه وابستگی‌ها، ابرکلید باشد، آن‌گاه آن جدول در BCNF قرار دارد که در وابستگی‌های فوق این چنین نیست. پس BCNF هم نیست. در وابستگی‌های فوق، سمت چپ وابستگی‌های کامل، ابرکلید است. اما سمت چپ وابستگی‌های بخشی و انتقالی، ابرکلید نیست.

بنابراین این جدول به دلیل نقض شرایط مربوطه، در نرمال فرم BCNF قرار ندارد. در نتیجه گزینه اول عبارت درستی است و به تبع پاسخ سوال هم نیست.

همچنین دو وابستگی تابعی سوم و چهارم مطرح شده در صورت سوال یعنی، وابستگی‌های زیر:

وابستگی کامل
غیرکلید $roomNo, from \rightarrow guestID$ کلید کاندید

واستگی کامل	عضو کلید کاندید to ← roomNo, from ← کلید کاندید
واستگی کامل	غیر کلید roomNo, to ← guestID کلید کاندید
واستگی کامل	عضو کلید کاندید from ← roomNo, to ← کلید کاندید
باعث نقض شدن نرمال فرم سوم یعنی 3NF نمی‌شود. زیرا وابستگی‌های بخشی و انتقالی باعث نقض شدن نرمال فرم سوم یعنی 3NF می‌شود. یعنی وابستگی‌های اول و دوم مطرح شده در صورت سوال یعنی، وابستگی‌های زیر:	
وابستگی انتقالی	غیر کلید questName ← guestID غیر کلید
وابستگی انتقالی	غیر کلید creditCard ← guestID غیر کلید
وابستگی بخشی	غیر کلید roomCat ← roomNo عضو کلید کاندید

در نتیجه گزینه دوم عبارت درستی است و به تبع پاسخ سوال هم نیست.

در وابستگی‌های فوق، وابستگی بخشی roomCat ← roomNo وجود دارد، بنابراین جدول مربوطه در نرمال فرم دوم قرار ندارد، بنابراین رویه نرمال‌سازی (تجزیه مطلوب) جهت قرارگیری جدول Booking در نرمال فرم دوم توسط عمل تجزیه جدول Booking می‌بایست انجام گردد.

شرایط تجزیه مطلوب به صورت زیر است:

۱- شرط لازم (Nonloss Join): دو جدول الحاق‌پذیر باشند (صفت مشترک در دو جدول، حداقل در یکی کلید کاندید باشد)

شرط الحاق‌پذیری به دو شکل زیر انجام می‌گردد:

۱-۱- برای حذف وابستگی بخشی از تئوری هیث استفاده می‌گردد.

تئوری هیث: این تئوری می‌گوید، وابستگی بخشی (وابستگی غیر کلید به جز کلید کاندید) را از جدول پایه خارج کنید.

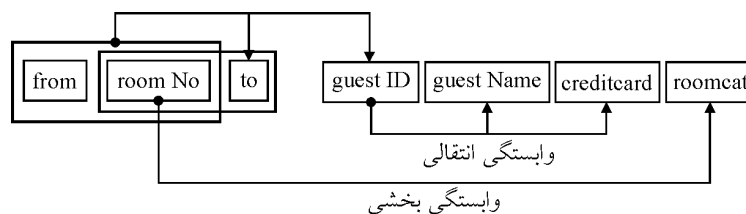
۱-۲- برای حذف وابستگی انتقالی از تئوری ریسانن استفاده می‌گردد.

تئوری ریسانن: این تئوری می‌گوید، وابستگی انتقالی (وابستگی غیر کلید به غیر کلید) را از جدول پایه خارج کنید.

۲- شرط کافی (حفظ وابستگی‌های تابعی): وابستگی‌های تابعی جدول پایه قابل استنتاج از روی وابستگی‌های تابعی جداول حاصل از تجزیه باشد یا در آن‌ها موجود باشد.
با توجه به نمودار وابستگی‌های جدول:

Booking (guestID, guestName, creditCard, roomNo, roomCat, from, to)

داریم:



توجه: کلید اصلی، یکی از کلیدهای کاندید است که بر اساس سلیقه طراح پایگاه داده، انتخاب می شود. مطابق قوانین مدل رابطه‌ای، هر جدول حتماً باید یک کلید اصلی داشته باشد. همچنین هر جدول فقط می تواند یک کلید اصلی داشته باشد. هر کلید کاندیدی که به عنوان کلید اصلی انتخاب نشده است، یک کلید فرعی است. فرض کنید که ترکیب صفات $\{roomNo, from\}$ **کلید اصلی** و ترکیب صفات $\{roomNo, to\}$ **کلید فرعی** جدول Booking باشد.

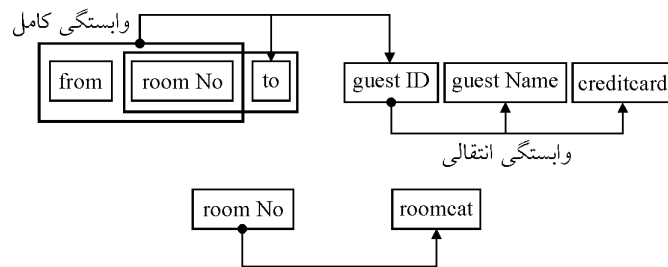
به طور کلی می توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

□ جدول باید در نرمال فرم اول باشد.

□ جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مؤلفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می نامند. با توجه به این که ترکیب صفات $\{roomNo, from\}$ و $\{roomNo, to\}$ به عنوان کلیدهای کاندید رابطه Booking معرفی شده است، وابستگی تابعی زیر:

وابستگی بخشی
غیرکلید $roomNo \rightarrow roomCat$ عضو کلید کاندید
به عنوان وابستگی بخشی محسوب می گردد. بنابراین جداول Booking در نرمال فرم دوم قرار ندارد. بنابراین برای قرار دادن جداول Booking، در نرمال فرم دوم، باید وابستگی بخشی از جداول Booking خارج گردد.



بنابراین جدول Booking به دو جدول کوچک تر تجزیه می گردد:

R1 (from, roomNo, to, guestID, guestName, creditCard)

R2 (roomNo, roomCat)

با توجه به کلید کاندید جدول R1 می توان دریافت که این جدول فاقد وابستگی بخشی است، بنابراین این جدول در نرمال فرم دوم قرار دارد. همچنین با توجه به کلید کاندید جدول R2 می توان دریافت که این جدول فاقد وابستگی بخشی و انتقالی است، بنابراین این جدول در نرمال فرم دوم و سوم قرار دارد. همچنین از آن جا که سمت چپ تمام وابستگی های جدول R2 ابرکلید است، بنابراین جدول R2 در سطح BCNF هم قرار دارد.

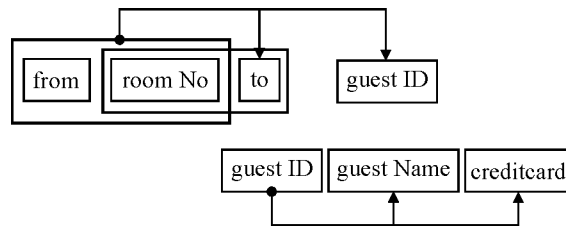
به طور کلی می توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

□ جدول باید در نرمال فرم دوم باشد.

□ جدول باید فاقد وابستگی انتقالی باشد.

وابستگی انتقالی: وابستگی یک مؤلفه غیرکلیدی به یک مؤلفه غیرکلیدی دیگر را وابستگی انتقالی می نامند. در جدول R1 وابستگی تابعی زیر:

وابستگی انتقالی
 غیرکلید $questID \rightarrow \{questName, creditCard\}$
 به عنوان وابستگی انتقالی محسوب می‌گردد. بنابراین جدول R1 در نرمال فرم سوم قرار ندارد.
 بنابراین برای قرار دادن R1، در نرمال فرم سوم، باید وابستگی انتقالی از جدول R1 خارج گردد.



بنابراین جدول R1 به جداول کوچک‌تر تجزیه می‌گردد:

R11(from, roomNo, to, guestID)

R12(guestID, guestName, creditCard)

با توجه به کلیدهای کاندید در جداول R11 و R12 می‌توان دریافت که این جداول فاقد وابستگی بخشی و انتقالی هستند و بنابراین در نرمال فرم دوم و سوم قرار دارند. هم‌چنین از آن‌جا که سمت چپ تمام وابستگی‌های جداول R11 و R12، ابرکلید است، بنابراین جداول R11 و R12 در سطح BCNF نیز قرار دارند. به این ترتیب با تجزیه جدول Booking به سه جدول R11، R12 و R2 می‌توان جدول Booking را در نرمال فرم دوم، سوم و سطح BCNF قرار داد.

توجه: آنچه تا به اینجا مطرح شد، مربوط به تجزیه مطلوب جدول Booking بود.

توجه: در گزینه‌ی سوم مطرح شده است که، اگر رابطه Booking به سه رابطه:

۱- R1 (guestID, roomNo, from, to)

۲- R2 (roomNo, roomCat) و

۳- R3 (guestName, creditCard, roomNo, from)

تجزیه شود، هر سه رابطه در 3NF و BCNF هستند و تجزیه بدون اتلاف (lossless decomposition) است. در واقع گزینه‌ی سوم به دنبال تجزیه‌ی مطلوب و برقراری شرط لازم و شرط کافی نیست. بلکه فقط برقراری شرط لازم یعنی تجزیه بدون اتلاف (lossless decomposition) و به تبع تجزیه نامطلوب را مدنظر قرار داده است. دقت کنید که تجزیه مطلوب پیش‌تر بررسی شد. شرط لازم به صورت زیر است:
شرط لازم (Nonloss Join): دو جدول الحاق‌پذیر باشند (صفت مشترک در دو جدول، حداقل در یکی کلید کاندید باشد)

با توجه به وابستگی‌های مرتبط برای رابطه R1 (guestID, roomNo, from, to) داریم:

$roomNo, from \rightarrow guestID$

$roomNo, from \rightarrow to$

$roomNo, to \rightarrow guestID$

$roomNo, to \rightarrow from$

$guestID, roomNo, from, to = guestID, from, to = roomNo$

بنابر رابطه فوق صفت roomNo حتماً باید عضو کلید کاندید باشد.
 بستار صفت roomNo برای جدول R1(guestID, roomNo, from, to) به صورت زیر است:

$$\{\text{roomNo}\}^+ = \{\text{roomNo}\}$$

قانون سوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول، برخی از ستون‌ها را تولید کند، بدین معنی است که، جدول موردنظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت.

بر اساس بستار فوق، صفت roomNo به عنوان عضو کلید کاندید همه ستون‌ها را تولید نمی‌کند، بنابراین مطابق قانون سوم ارسطو باید صفاتی در کنار صفات roomNo قرار گیرد تا کلید کاندید تولید گردد. این صفات کناری از صفات باقی مانده انتخاب می‌گردند، صفات باقی مانده عبارتند از guestID, from, to البته از این مجموعه صفات guestID را هم کنار می‌گذاریم، زیرا کمکی در تولید صفات دیگر نمی‌کند. چون در سمت چپ هیچ یک از وابستگی‌های تابعی نیامده است. بنابراین با ترکیب صفات from یا to با صفت roomNo کلیدهای کاندید تولید می‌گردند.

$$\{\text{roomNo, from}\}^+ = \{\text{roomNo, from, guestID, to}\}$$

$$\{\text{roomNo, to}\}^+ = \{\text{roomNo, to, guestID, from}\}$$

پس ترکیب صفات $\{\text{roomNo, from}\}$ و $\{\text{roomNo, to}\}$ کلیدهای کاندید جدول R1 هستند.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه R1 در مجموع دارای دو کلید کاندید است. که مطابق قانون سوم ارسطو، عضو کلید کاندید roomNo، در بین هر دو کلید کاندید به طور مشترک قرار دارد.

حال یک‌بار دیگر وابستگی‌های مطرح شده برای رابطه R1(guestID, roomNo, from, to) را در نظر بگیرید:

وابستگی کامل ← guestID ← roomNo, from ← کلید کاندید

وابستگی کامل ← to ← roomNo, from ← عضو کلید کاندید

وابستگی کامل ← guestID ← roomNo, to ← غیرکلید

وابستگی کامل ← from ← roomNo, to ← عضو کلید کاندید

در وابستگی‌های فوق، وابستگی بخشی، انتقالی و معکوس وجود ندارد. بنابراین جدول R1 در نرمال فرم دوم، سوم و BCNF قرار دارد.

با توجه به وابستگی‌های مرتبط برای رابطه R2(roomNo, roomCat) داریم:

$$\text{roomNo} \longrightarrow \text{roomCat}$$

$roomNo, roomCat - roomCat = roomNo$

بنابر رابطه فوق صفت $roomNo$ حتماً باید عضو کلید کاندید باشد.

بستار صفت $roomNo$ برای جدول $R2(roomNo, roomCat)$ به صورت زیر است:

$\{roomNo\}^+ = \{roomNo, roomCat\}$

قانون دوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

حال یک‌بار دیگر وابستگی‌های مطرح شده برای رابطه $R2(roomNo, roomCat)$ را در نظر بگیرید:

وابستگی کامل $roomNo \longrightarrow roomCat$ غیرکلید

در وابستگی‌های فوق، وابستگی بخشی، انتقالی و معکوس وجود ندارد. بنابراین جدول $R2$ در نرمال فرم دوم، سوم و $BCNF$ قرار دارد.

با توجه به وابستگی‌های مرتبط برای رابطه $R3(guestName, creditCard, roomNo, from)$ داریم:

{

در وابستگی‌های فوق، وابستگی بخشی، انتقالی و معکوس وجود ندارد. بنابراین جدول $R3$ در نرمال فرم دوم، سوم و $BCNF$ قرار دارد. در واقع جدول $R3$ یک جدول تمام کلید است. و یک جدول تمام کلید همواره در سطح $BCNF$ نرمال قرار دارد.

توجه: همانطور که گفتیم، در گزینه‌ی سوم مطرح شده است که، اگر رابطه $Bookin$ به سه رابطه:

۱- $R1(guestID, roomNo, from, to)$

۲- $R2(roomNo, roomCat)$ و

۳- $R3(guestName, creditCard, roomNo, from)$

تجزیه شود، هر سه رابطه در $3NF$ و $BCNF$ هستند و تجزیه بدون اتلاف (lossless decomposition) است. در واقع گزینه‌ی سوم به دنبال تجزیه‌ی مطلوب و برقراری شرط لازم و شرط کافی نیست. بلکه فقط برقراری شرط لازم یعنی تجزیه بدون اتلاف (lossless decomposition) و به تبع تجزیه نامطلوب را مدنظر قرار داده است. دقت کنید که تجزیه مطلوب پیش‌تر بررسی شد. شرط لازم به صورت زیر است:

شرط لازم (Nonloss Join): دو جدول الحاق‌پذیر باشند (صفت مشترک در دو جدول، حداقل در یکی کلید کاندید باشد)

در گزینه‌ی سوم صفت مشترک در دو جدول، حداقل در یکی کلید کاندید است، بنابراین شرط لازم برقرار است و تجزیه بدون اتلاف (lossless decomposition) است. به صورت زیر:

$R1(guestID, roomNo, from, to)$

$R2(roomNo, roomCat)$

$R3(guestName, creditCard, roomNo, from)$

همچنین همانطور که گفتیم هر سه جدول $R1$ ، $R2$ و $R3$ مطرح شده در گزینه‌ی سوم، در نرمال فرم دوم، سوم و $BCNF$ قرار دارند. در نتیجه گزینه سوم عبارت درستی است و به تبع پاسخ سوال هم نیست.

توجه: در گزینه‌ی چهارم مطرح شده است که، اگر رابطه Booking به سه رابطه:

۱- $R1 (guestID, guestName, creditCard)$

۲- $R2 (roomNo, roomCat)$ و

۳- $R3 (guestID, roomNo, from, to)$

تجزیه شود، هر سه رابطه در 3NF هستند ولی در BCNF نیستند. در واقع گزینه‌ی چهارم به دنبال تجزیه‌ی مطلوب و برقراری شرط لازم و شرط کافی نیست. بلکه فقط برقراری سطوح نرمال را مد نظر قرار داده است.

با توجه به وابستگی‌های مرتبط برای رابطه $R1 (guestID, guestName, creditCard)$ داریم:

$guestID \longrightarrow guestName, creditCard$

$guestID, guestName, creditCard - guestName, creditCard = guestID$

بنابر رابطه فوق صفت $guestID$ حتماً باید عضو کلید کاندید باشد.

بستار صفت $guestID$ برای جدول $R1 (guestID, guestName, creditCard)$ به صورت زیر است:

$\{guestID\}^+ = \{guestID, guestName, creditCard\}$

قانون دوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود. حال یک‌بار دیگر وابستگی‌های مطرح شده برای رابطه $R1 (guestID, guestName, creditCard)$ را در نظر بگیرید:

وابستگی کامل
 غیرکلید $guestID \longrightarrow guestName, creditCard$ کلید کاندید
 در وابستگی فوق، وابستگی بخشی، انتقالی و معکوس وجود ندارد. بنابراین جدول $R1$ در نرمال فرم دوم، سوم و BCNF قرار دارد.

با توجه به وابستگی‌های مرتبط برای رابطه $R2 (roomNo, roomCat)$ داریم:

$roomNo \longrightarrow roomCat$

$roomNo, roomCat - roomCat = roomNo$

بنابر رابطه فوق صفت $roomNo$ حتماً باید عضو کلید کاندید باشد.

بستار صفت $roomNo$ برای جدول $R2 (roomNo, roomCat)$ به صورت زیر است:

$\{roomNo\}^+ = \{roomNo, roomCat\}$

قانون دوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

حال یک‌بار دیگر وابستگی‌های مطرح شده برای رابطه $R2 (roomNo, roomCat)$ را در نظر بگیرید:

وابستگی کامل
 غیرکلید $roomNo \longrightarrow roomCat$ کلید کاندید

در وابستگی های فوق، وابستگی بخشی، انتقالی و معکوس وجود ندارد. بنابراین جدول R2 در نرمال فرم دوم، سوم و BCNF قرار دارد.

با توجه به وابستگی های مرتبط برای رابطه R3(guestID, roomNo, from, to) داریم:

roomNo, from \longrightarrow guestID

roomNo, from \longrightarrow to

roomNo, to \longrightarrow guestID

roomNo, to \longrightarrow from

guestID, roomNo, from, to = roomNo

بنابر رابطه فوق صفت roomNo حتماً باید عضو کلید کاندید باشد.

بستار صفت roomNo برای جدول R3(guestID, roomNo, from, to) به صورت زیر است:

$$\{\text{roomNo}\}^+ = \{\text{roomNo}\}$$

قانون سوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول، برخی از ستون ها را تولید کند، بدین معنی است که، جدول مورد نظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت.

بر اساس بستار فوق، صفت roomNo به عنوان عضو کلید کاندید همه ستون ها را تولید نمی کند، بنابراین مطابق قانون سوم ارسطو باید صفاتی در کنار صفت roomNo قرار گیرد تا کلید کاندید تولید گردد. این صفات کناری از صفات باقی مانده انتخاب می گردند، صفات باقی مانده عبارتند از guestID, from, to البته از این مجموعه صفات guestID را هم کنار می گذاریم، زیرا کمکی در تولید صفات دیگر نمی کند. چون در سمت چپ هیچ یک از وابستگی های تابعی نیامده است. بنابراین با ترکیب صفات from یا to با صفت roomNo کلیدهای کاندید تولید می گردند.

$$\{\text{roomNo, from}\}^+ = \{\text{roomNo, from, guestID, to}\}$$

$$\{\text{roomNo, to}\}^+ = \{\text{roomNo, to, guestID, from}\}$$

پس ترکیب صفات {roomNo, from} و {roomNo, to} کلیدهای کاندید جدول R3 هستند.

توجه: دقت کنید که هیچ گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه R3 در مجموع دارای دو کلید کاندید است. که مطابق قانون سوم ارسطو، عضو کلید کاندید roomNo، در بین هر دو کلید کاندید به طور مشترک قرار دارد.

حال یک بار دیگر وابستگی های مطرح شده برای رابطه R3(guestID, roomNo, from, to) را در نظر بگیرید:

وابستگی کامل غیر کلید roomNo, from \longrightarrow guestID

وابستگی کامل عضو کلید کاندید roomNo, from \longrightarrow to

وابستگی کامل
غیرکلید $roomNo, to \rightarrow guestID$ کلید کاندید

وابستگی کامل
عضو کلید کاندید $roomNo, to \rightarrow from$ کلید کاندید

در وابستگی های فوق، وابستگی بخشی، انتقالی و معکوس وجود ندارد. بنابراین جدول R3 در نرمال فرم دوم، سوم و BCNF قرار دارد.

توجه: همانطور که گفتیم، در گزینه ی چهارم مطرح شده است که، اگر رابطه Booking به سه رابطه:

۱- $R1 (guestID, guestName, creditCard)$

۲- $R2 (roomNo, roomCat)$ و

۳- $R3 (guestID, roomNo, from, to)$

تجزیه شود، هر سه رابطه در 3NF هستند ولی در BCNF نیستند. اما همانطور که گفتیم هر سه جدول R1، R2 و R3 مطرح شده در گزینه ی چهارم، در نرمال فرم دوم، سوم و BCNF قرار دارند. در نتیجه گزینه چهارم عبارت نادرستی است و به تبع پاسخ سوال است.

۴۳- گزینه (۳) صحیح است.

با کمی بررسی می توان دید تنها کاندید DGH است. پس گزینه دوم صحیح است. برای به دست آوردن یک ابرکلید باید از بین پنج صفتی که در کلید کاندید نیستند تعداد صفر، یک، دو، ... یا پنج صفت را به دلخواه انتخاب و به صفات DGH اضافه کنیم. تعداد انتخاب های ممکن به صورت زیر به دست می آید:

$$\binom{5}{0} + \binom{5}{1} + \binom{5}{2} + \binom{5}{3} + \binom{5}{4} + \binom{5}{5} = 2^5 = 32$$

پس این رابطه 32 ابرکلید دارد. پس گزینه اول صحیح است. رابطه داده شده در 2NF نیست چون وابستگی تابعی $D \rightarrow EF$ یک وابستگی به زیر مجموعه ای از صفات کلید اصلی است. پس گزینه سوم غلط است. چون رابطه داده شده در 2NF نیست پس در 3NF هم نیست (البته وابستگی انتقالی $D \rightarrow F$ و $F \rightarrow C$ هم وجود دارد). پس گزینه چهارم صحیح است.

۴۴- گزینه (۲) صحیح است.

این رابطه در 1NF هست چون همه کلیدهای آن معلوم اند، تمام صفات با کلید اصلی وابستگی تابعی دارند (هر یک از دو کلید کاندید را به دلخواه کلید اصلی فرض کنید)، و صفات آن از دامنه های تو در تو (چند مقداری و مرکب) نیستند.

این رابطه هم چنین در 2NF هست چون هیچ یک از صفات غیرکلیدی آن به زیر مجموعه ای از کلید کاندید وابسته نیستند. (وابستگی بخشی ندارد).

این رابطه هم چنین در 3NF هست چون صفات غیرکلیدی آن (یعنی D و E) به هم وابستگی ندارند. (وابستگی انتقالی ندارد).

هم چنین این جدول در BCNF هم هست چون ستون های آن فقط به کلیدهای کاندید وابستگی تابعی دارند. هرگاه همه شروع های مجموعه وابستگی ها، ابرکلید بودند، آن جدول BCNF است. به بیان دیگر هرگاه سمت چپ همه وابستگی ها، ابرکلید باشد، آن گاه آن جدول در BCNF قرار دارد.

۴۵- گزینه (۲) صحیح است.

این رابطه در 1NF هست چون کلیدهای آن معلوم اند (کد دانشجو)، صفت‌های آن به کلید اصلی (کد دانشجو) وابسته‌اند، و صفت‌ها از دامنه‌های تو در تو نیستند.

این رابطه در 2NF هست چون هیچ صفتی به زیر مجموعه‌ای از کلید کاندید اصلی وابستگی ندارد.

این رابطه در 3NF هم هست چون وابستگی به صفت‌های غیرکلیدی نداریم (توجه کنید که از روی نام درس نمی‌توان نام استاد آن را به صورت یکتا تعیین کرد).

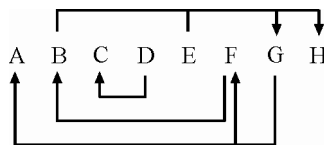
بنابراین وابستگی استاد \rightarrow درس وجود ندارد. به دلیل مشابه، وابستگی درس \rightarrow استاد هم وجود ندارد)

این جدول در BCNF هم هست چون ستون‌های آن فقط به کلیدهای کاندیدش وابسته‌اند.

اما این رابطه در 4NF نیست چون وابستگی‌های چند مقداری غیربدیهی درس $\rightarrow \rightarrow$ استاد و استاد $\rightarrow \rightarrow$ درس وجود دارند که در آن‌ها «درس» و «استاد» ابرکلید نیستند.

۴۶- گزینه (۳) صحیح است.

نمودار وابستگی تابعی رابطه R به صورت زیر است:



به سادگی دیده می‌شود که صفت A توسط برخی از صفات دیگر قابل دستیابی است (توسط G به طور مستقیم و توسط B و F به طور واسطه). بنابراین هر کلید که شامل این صفات باشد نیازی نیست شامل صفت A باشد. پس گزینه سوم عبارت نادرستی است و پاسخ این سؤال است.

همچنین مشاهده می‌شود که صفت D توسط هیچ یک از صفات دیگر قابل دستیابی نیست. بنابراین هر کلیدی باید حتماً شامل D باشد. پس گزینه چهارم عبارت درستی است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیربدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E, F, G, H)$ داریم:

$$BE \rightarrow GH$$

$$G \rightarrow FA$$

$$D \rightarrow C$$

$$F \rightarrow B$$

$$AB\bar{C}DE\bar{F}GH - ABCFGH = DE$$

بنابر رابطه فوق صفت DE حتماً باید عضو کلید کاندید باشد. بنابراین گزینه چهارم درست است و گزینه سوم نادرست است.

بستار صفات DE به صورت روبه‌رو است:

$$\{DE\}^+ = \{D, E, C\}$$

قانون سوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول، برخی از ستون‌ها را تولید کند، بدین معنی است که، جدول موردنظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت.

براساس بستار فوق، صفات DE به عنوان عضو کلید کاندید همه ستون‌ها را تولید نمی‌کند، بنابراین مطابق قانون سوم ارسطو باید صفاتی در کنار صفات DE قرار گیرد تا کلید کاندید تولید گردد. این صفات کناری از صفات باقی مانده به غیر از صفات CDE انتخاب می‌گردند، صفات باقی مانده عبارتند از H و G و F و B و A البته از این مجموعه صفات A و H را هم کنار می‌گذاریم، زیرا کمکی در تولید صفات دیگر نمی‌کنند. چون در سمت چپ هیچ یک از وابستگی‌های تابعی نیامده‌اند. بنابراین با ترکیب صفات B یا F یا G با صفات DE کلیدهای کاندید تولید می‌گردند.

$$\{BDE\}^+ = \{B, D, E, G, H, C, F, A\}$$

$$\{FDE\}^+ = \{F, D, E, B, C, G, H, A\}$$

$$\{GDE\}^+ = \{G, D, E, C, F, A, B, H\}$$

پس ترکیبات صفات BDE، FDE و GDE کلیدهای کاندید جدول R هستند.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده در مجموع دارای سه کلید کاندید است. که مطابق قانون سوم ارسطو، عضو کلید کاندید DE، در بین هر سه کلید کاندید به طور مشترک قرار دارد.

حال یک بار دیگر وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E, F, G, H)$ را در نظر بگیرید:

عضو کلید کاندید $BE \rightarrow G$	وابستگی معکوس
غیرکلید $BE \rightarrow H$	وابستگی بخشی
عضو کلید کاندید $G \rightarrow F$	وابستگی معکوس
غیرکلید $G \rightarrow A$	وابستگی بخشی
عضو کلید کاندید $D \rightarrow C$	وابستگی بخشی

وابستگی معکوس عضو کلید کاندید $B \rightarrow F$ عضو کلید کاندید
وابستگی معکوس: وابستگی یک عضو کلید کاندید به عضو یک کلید کاندید دیگر یا مؤلفه غیرکلیدی راه، وابستگی معکوس می‌نامند.

عضو کلید کاندید \rightarrow عضو کلید کاندید

عضو کلید کاندید \rightarrow غیرکلید

در وابستگی‌های فوق، وابستگی بخشی (غیرکلید به عضو کلید کاندید) وجود دارد، بنابراین جدول در نرمال فرم دوم قرار ندارد، و در نتیجه در فرم نرمال سطح سوم و BCNF هم قرار ندارد. بنابراین گزینه اول و دوم درست هستند.

در یک نگاه دیگر برای BCNF می‌توان گفت، جدولی در BCNF قرار دارد که همه‌ی شروع‌های وابستگی‌ها، ابرکلید باشد. به بیان دیگر هرگاه سمت چپ همه وابستگی‌ها، ابرکلید باشد، آن‌گاه آن جدول در BCNF قرار دارد که در وابستگی‌های فوق این چنین نیست. پس BCNF هم نیست.

۴۷- گزینه (۱) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیربدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n (x_i \text{ (چپ)} - y_i \text{ (راست)})$$

توجه: وابستگی بدیهی: اگر در یک وابستگی، بخش سمت راست، زیرمجموعه بخش سمت چپ باشد، آن وابستگی راه، وابستگی بدیهی می‌نامند.

مثال: $AB \rightarrow A$

با توجه به وابستگی‌های مطرح شده برای رابطه (E و D و C و B و A) داریم:

$BC \rightarrow A$

$A \rightarrow D$

$D \rightarrow C$

$D \rightarrow E$

$ACDE = B$

بنابر رابطه فوق صفت B حتماً باید عضو کلید کاندید باشد.

بستار صفت B به صورت زیر است:

$$\{B\}^+ = \{B\}$$

بر اساس بستار فوق، صفت B، فقط ستون B را تولید می کند، پس صفت B فقط عضو کلید کاندید می باشد و کلید کاندید نمی باشد.

قانون سوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، برخی از ستون ها را تولید کند، بدین معنی است که، جدول مورد نظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد. همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت.

صفات همراه عضو کلید کاندید، از صفات باقی مانده به غیر از صفت B انتخاب می گردند، صفات باقی مانده عبارتند از E و D و C و A. البته از این مجموعه، صفت E را هم کنار می گذاریم، زیرا کمکی در تولید صفات دیگر نمی کند. چون در سمت چپ هیچ یک از وابستگی های تابعی نیامده است. بنابراین با ترکیب صفات A یا C یا D با صفت B کلیدهای کاندید تولید می گردند.

$$\{AB\}^+ = \{A, B, D, C, E\}$$

$$\{CB\}^+ = \{C, B, A, D, E\}$$

$$\{DB\}^+ = \{D, B, C, A, E\}$$

بر اساس بستارهای فوق، صفات AB، CB و DB، هر یک به تنهایی همه ستون ها را تولید می کنند، پس صفات AB، CB و DB، کلیدهای کاندید جدول R هستند.

حال یکبار دیگر وابستگی های مطرح شده برای رابطه $R(A, B, C, D, E)$ را در نظر بگیرید:

وابستگی کامل عضو کلید کاندید $BC \rightarrow A$ کلید کاندید

وابستگی معکوس عضو کلید کاندید $A \rightarrow D$ عضو کلید کاندید

وابستگی معکوس عضو کلید کاندید $D \rightarrow C$ عضو کلید کاندید

وابستگی بخشی غیرکلید $D \rightarrow E$ عضو کلید کاندید

وابستگی معکوس: وابستگی یک عضو کلید کاندید به یک عضو کلید کاندید دیگر یا مؤلفه غیرکلیدی را، وابستگی معکوس می نامند.

عضو کلید کاندید \rightarrow عضو کلید کاندید

عضو کلید کاندید \rightarrow غیرکلید

در وابستگی های فوق، وابستگی بخشی (غیرکلید به عضو کلید کاندید) وجود دارد، بنابراین جدول در نرمال فرم دوم قرار ندارد، و در نتیجه در فرم نرمال سطح سوم و BCNF هم قرار ندارد.

در یک نگاه دیگر، برای BCNF می توان گفت، جدولی در BCNF قرار دارد که همه شروع های وابستگی ها، ابرکلید باشد. به بیان دیگر هرگاه سمت چپ همه وابستگی ها، ابرکلید باشد، آنگاه آن جدول در BCNF قرار دارد که در وابستگی های فوق این چنین نیست. پس BCNF هم نیست.

۴۸- گزینه (۳) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر بدست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیربدهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E, F, G)$ داریم:

$$AB \rightarrow CDEFG$$

$$A \rightarrow G$$

$$B \rightarrow C$$

$$C \rightarrow D$$

$$AB \rightarrow CDEFG - CDEFG = AB$$

بنابر رابطه فوق صفات AB حتماً باید عضو کلید کاندید باشد. بستر صفات AB به صورت زیر است:

$$\{AB\}^+ = \{A, B, C, D, E, F, G\}$$

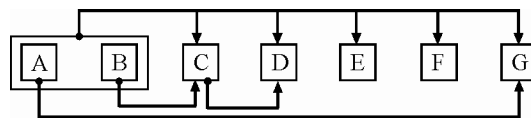
بر اساس بستر فوق، صفات AB ، همه ستون‌ها را تولید می‌کند، پس صفات AB کلید کاندید می‌باشد.

قانون دوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

جدولی در سطح نرمال فرم اول قرار دارد که صفت چند مقداری و مرکب نداشته باشد. از آنجا که در جدول R ، صفت چند مقداری و مرکب وجود ندارد، بنابراین این جدول در نرمال فرم اول قرار دارد.

نمودار وابستگی‌های جدول R به صورت زیر است:



به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

□ جدول باید در نرمال فرم اول باشد.

□ جدول باید فاقد وابستگی بخشی باشد.

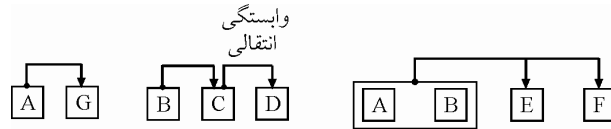
وابستگی بخشی: وابستگی یک مؤلفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می‌نامند. با

توجه به این که ترکیب صفات AB کلید کاندید رابطه R می‌باشد، وابستگی‌های تابعی زیر:

$$A \rightarrow G$$

$$B \rightarrow C$$

به عنوان وابستگی بخشی محسوب می گردند. بنابراین جدول R در نرمال فرم دوم قرار ندارد. بنابراین برای قرار دادن جدول R، در نرمال فرم دوم، باید وابستگی های بخشی از جدول R خارج گردند.



بنابراین جدول R به سه جدول کوچکتر تجزیه می گردد:

$$R_1(\underline{A}, G) \quad R_2(\underline{B}, C, D) \quad R_3(\underline{A, B}, E, F)$$

با توجه به کلید کاندید در دو جدول R_1 و R_3 می توان دریافت که این دو جدول فاقد وابستگی بخشی و انتقالی هستند، بنابراین در نرمال فرم سوم قرار دارند. همچنین از آن جا که سمت چپ تمام وابستگی های دو جدول R_1 و R_3 ابرکلید است، بنابراین دو جدول R_1 و R_3 در سطح BCNF نیز قرار دارند. بنابراین گزینه سوم درست است. اما راه حل را تا سطح BCNF برای جدول R_2 نیز ادامه می دهیم. به طور کلی می توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

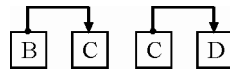
□ جدول باید در نرمال فرم دوم باشد.

□ جدول باید فاقد وابستگی انتقالی باشد.

وابستگی انتقالی: وابستگی یک مؤلفه غیرکلیدی به یک مؤلفه غیرکلیدی دیگر را وابستگی انتقالی می نامند. در جدول R_2 وابستگی تابعی زیر:

$$C \rightarrow D$$

به عنوان وابستگی انتقالی محسوب می گردد. بنابراین جدول R_2 در نرمال فرم سوم قرار ندارد. بنابراین برای قرار دادن جدول R_2 ، در نرمال فرم سوم، باید وابستگی های انتقالی از جدول R_2 خارج گردند.



بنابراین جدول R_2 به دو جدول کوچکتر تجزیه می گردد.

$$R_{21}(\underline{B}, C) \quad R_{22}(\underline{C}, D)$$

با توجه به کلیدهای کاندید در دو جدول R_{21} و R_{22} می توان دریافت که این دو جدول فاقد وابستگی بخشی و انتقالی هستند و بنابراین در نرمال فرم سوم قرار دارند. همچنین از آن جا که سمت چپ تمام وابستگی های سه جدول R_{21} و R_{22} ابرکلید است، بنابراین دو جدول R_{21} و R_{22} در سطح BCNF نیز قرار دارند. به این ترتیب با تجزیه جدول R به چهار جدول R_1 ، R_2 ، R_{21} ، R_{22} می توان جدول R را در نرمال فرم سوم و سطح BCNF قرار داد.

۴۹- گزینه (۴) صحیح است.

گزینه اول نادرست است. عملیات نرمال سازی یک عملیات چند مرحله ای است که با طی آن ها اهداف نرمال سازی به تدریج حاصل می شود. نخستین مرحله نرمال سازی تبدیل جداول پایه به نرمال فرم اول است. برای قرار گرفتن یک جدول در نرمال فرم اول، شروط خاصی باید برقرار باشد. یک جدول (رابطه) برای قرار گرفتن در نرمال فرم اول باید دارای شروط زیر باشد: (حفظ قوانین جامعیت درون رابطه ای)

۱- دارای حداقل یک کاندید باشد.

۲- جدول تو جدول یا جدول تو در تو نباشد (یعنی جدول فاقد صفت مرکب و چند مقداری باشد)

۱-۲- همه صفات آن غیر قابل تجزیه (Atomic) باشند. (جدول باید فاقد صفت مرکب باشد)
 ۲-۲- همه صفات جدول تک مقداری باشند. (جدول باید فاقد صفت چند مقداری باشد).
 گزینه اول برای تحقق سطح اول نرمال سازی، یعنی نرمال فرم اول، فقط به یکی از سه شرط فوق اشاره کرده است.
 گزینه دوم نادرست است. زیرا، ابرکلیدی که عضو زائد نداشته باشد، کلید کاندید است. به عبارت دیگر ابرکلیدی که کاهش ناپذیر باشد، کلید کاندید است.

ابرکلید (Supper key)

هر ترکیبی از صفات (ستون ها) که خاصیت کلیدی داشته باشد، یک ابرکلید است. دقت کنید که خاصیت کلیدی بر اساس محیط خارج مشخص می گردد و نه محتوی جدول.
تعریف خاصیت کلیدی: اگر بر اساس آن مورد، جستجو انجام شود، فقط و فقط یک مورد را به صورت یکتا برگرداند. در واقع قدرت تفکیک داشته باشد.

مثال: شماره دانشجویی یا شماره ملی

توجه: هر ترکیبی از ابرکلید با سایر ستون های جدول مورد نظر نیز یک ابرکلید است. بنابراین ابرکلید می تواند عضو زائد داشته باشد و کمینه (minimal) نباشد، پس گزینه دوم پاسخ سؤال است.

مثال:

S#	Sname	city
S ₁	Sn ₁	C ₁
S ₂	Sn ₂	C ₂
S ₃	Sn ₃	C ₂
S ₄	Sn ₄	C ₂

S#: ابرکلید است.

Sname: ابرکلید نیست.

(Sname, City): ابرکلید نیست.

(S#, Sname): ابرکلید است.

(S#, City): ابرکلید است.

مثال: تعداد ابرکلیدهای مثال فوق چند عدد است؟

پاسخ: برابر 4 عدد می باشد.

S#	+	Sname	City	ابر کلید
	0		0 →	(S#)
خاصیت	0		1 →	(S#, City)
کلیدی دارد	1		0 →	(S#, Sname)
بنابراین	1		1 →	(S#, Sname, City)
ابرکلید است.				

کلید کاندید (Candidate key)

ابرکلیدی که عضو زائد نداشته باشد، کلید کاندید است.

مثال:

S#	Sname	City	S#	P#	QTY	P#	Pname	Color
S ₁	Sn ₁	C ₁	S ₁	P ₁	10	P ₁	Pn ₁	Red
S ₂	Sn ₂	C ₂	S ₁	P ₂	20	P ₂	Pn ₂	Blue
S ₃	Sn ₃	C ₂	S ₂	P ₁	30	P ₃	Pn ₃	Blue

جدول S

جدول SP

جدول P

S#: ابرکلید است. کلید کاندید نیز هست. (در جدول S)
 (S#, Sname): ابرکلید است، زیرا خاصیت کلیدی دارد، اما کلید کاندید نیست، زیرا عضو زائد Sname را دارد. در واقع صفت S#، به تنهایی خاصیت کلیدی دارد، بنابراین صفت Sname، عضو زائد است. (در جدول S)
 (S#, P#): ابرکلید است. کلید کاندید نیز هست. (در جدول SP).

مثال:

شماره ملی: ابرکلید است. کلید کاندید نیز هست.
 (شماره ملی و نام خانوادگی): ابرکلید است. زیرا خاصیت کلیدی دارد، اما کلید کاندید نیست، زیرا عضو زائد نام خانوادگی را دارد. در واقع صفت شماره ملی، به تنهایی خاصیت کلیدی دارد، بنابراین صفت نام خانوادگی، عضو زائد است.
 توجه: یک جدول می تواند چندین کلید کاندید داشته باشد.

مثال:

شماره دانشجویی	شماره ملی	نام خانوادگی	نام
کلید کاندید (کلید اصلی)	کلید کاندید (کلید فرعی)		

توجه: در مدل رابطه ای، هر رابطه حتماً حداقل یک کلید کاندید دارد، زیرا در بدترین شرایط، همه صفات با هم کلید کاندید می شوند، که به این رابطه تمام کلید (All key) گفته می شود.
 توجه: یک رابطه، تحت هیچ شرایطی نمی تواند به دلیل استفاده از خاصیت مجموعه ای بودن، سطر تکراری داشته باشد. بنابراین یک رابطه، حداقل یک کلید کاندید دارد.
 مثال: یک جدول تمام کلید.

a	b	c
1	2	3
1	6	3
1	2	7
8	2	3

ویژگی های رابطه در مدل رابطه ای به صورت زیر است:

- ۱- در رابطه، تاپل تکراری وجود ندارد. زیرا رابطه یک مجموعه است و مجموعه در ریاضیات طبق تعریف تاپل تکراری ندارد. بنابراین در جدول هم رکوردهای تکراری نداریم. بنابراین گزینه چهارم درست است.
- ۲- تاپل ها در رابطه نظم مکانی ندارند. زیرا مجموعه نظم ندارد. بنابراین ترتیب رکوردها در جدول اهمیت ندارد. بنابراین گزینه سوم نادرست است.

۳- ترتیب صفات خاصه در یک تاپل مهم نیست. البته به شرطی که ترتیب صفات خاصه در تمام تاپل‌های موجود در یک رابطه با یکدیگر سازگار باشند. بنابراین ترتیب فیلدها در یک جدول نیز مهم نیست، نظم ندارد و می‌توان آنرا جابه‌جا کرد. البته به شرطی که ترتیب فیلدها در تمام رکوردهای موجود در یک جدول با یکدیگر سازگار باشند.

۴- همهی مقادیر صفات خاصه در تاپل‌ها تجزیه‌ناپذیرند. به عبارت دیگر در مدل رابطه‌ای نمی‌توان فیلد مرکبی که از فیلدهای ساده تشکیل شده‌است، تعریف نمود. یعنی فیلدهای موجود در جداول مدل رابطه‌ای نباید مرکب مثل تاریخ تولد و آدرس باشد و نباید چندمقداری مثل مدرک تحصیلی باشد.

۵۰- گزینه (۱) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیربدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i (\text{چپ}) - y_i (\text{راست})]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D)$ داریم:

$$AB \rightarrow C$$

$$AB \rightarrow D$$

$$BC \rightarrow A$$

$$BC \rightarrow D$$

$$A \rightarrow C$$

$$C \rightarrow A$$

$$ABC \rightarrow D - ACD = B$$

بنابر رابطه فوق صفت B حتماً باید عضو کلید کاندید باشد.

$$\{B\}^+ = \{B\}$$

بستار صفات B به صورت روبه‌رو است:

قانون سوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول، برخی از ستون‌ها را تولید کند، بدین معنی است که، جدول موردنظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد. همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت.

بر اساس بستار فوق، صفت B به عنوان عضو کلید کاندید همه ستون‌ها را تولید نمی‌کند، بنابراین مطابق قانون سوم ارسطو باید صفاتی در کنار صفت B قرار گیرد تا کلید کاندید تولید گردد. این صفات کناری از صفات باقی مانده انتخاب می‌گردند، صفات باقی مانده عبارتند از D و C و A البته از این مجموعه صفت D را هم کنار می‌گذاریم، زیرا کمکی در تولید صفات دیگر نمی‌کند. چون در سمت چپ هیچ یک از وابستگی‌های تابعی نیامده است. بنابراین با ترکیب صفات A یا C با صفت B کلیدهای کاندید تولید می‌گردند.

$$\{AB\}^+ = \{A, B, C, D\}$$

$$\{BC\}^+ = \{B, C, A, D\}$$

پس ترکیبات صفات AB و BC کلیدهای کاندید جدول R هستند.

توجه: کلید اصلی، یکی از کلیدهای کاندید است که بر اساس سلیقه طراح پایگاه داده، انتخاب می‌شود. مطابق قوانین مدل رابطه‌ای، هر جدول حتماً باید یک کلید اصلی داشته باشد. همچنین هر جدول فقط می‌تواند یک کلید اصلی داشته باشد. هر کلید کانیدی که به عنوان کلید اصلی انتخاب نشده است، یک کلید فرعی است. مطابق فرض سوال AB کلید اصلی و کلید BC کلید کاندید (کلید فرعی) در نظر گرفته شده است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده در مجموع دارای دو کلید کاندید است. که مطابق قانون سوم ارسطو، عضو کلید کاندید B، در بین هر دو کلید کاندید به طور مشترک قرار دارد.

حال یک‌بار دیگر وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D)$ را در نظر بگیرید:

وابستگی کامل	عضو کلید کاندید C \rightarrow AB کلید کاندید
وابستگی کامل	غیر کلید D \rightarrow AB کلید کاندید
وابستگی کامل	عضو کلید کاندید A \rightarrow BC کلید کاندید
وابستگی کامل	غیر کلید D \rightarrow BC کلید کاندید
وابستگی معکوس	عضو کلید کاندید C \rightarrow A عضو کلید کاندید
وابستگی معکوس	عضو کلید کاندید A \rightarrow C عضو کلید کاندید

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم اول را به صورت زیر بیان کرد:

دارای حداقل یک کلید کاندید باشد.

همه خصیصه‌های آن غیر قابل تجزیه باشند (جدول باید فاقد خصیصه‌های مرکب باشد)

همه خصیصه‌های آن تک مقداری باشند (جدول باید فاقد خصیصه‌های چند مقداری باشد)

واضح است که جدول مطرح شده در فرم اول نرمال قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

جدول باید در نرمال فرم اول باشد.

جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مولفه غیر کلیدی، به جزئی از کلید کاندید را وابستگی بخشی می‌نامند.

مولفه غیر کلید: هر صفتی که عضو هیچ کلید کانیدی نباشد، به عنوان مؤلفه غیر کلیدی نامیده می‌شود.

مؤلفه جزء کلید کاندید: هر صفتی که عضو حداقل یک کلید کاندید باشد، به عنوان مؤلفه جزء کلید نامیده می‌شود.

در وابستگی‌های فوق، وابستگی بخشی وجود ندارد. بنابراین جدول مربوطه در نرمال فرم دوم هم قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

□ جدول باید در نرمال فرم دوم باشد.

□ جدول باید فاقد وابستگی انتقالی باشد.

وابستگی انتقالی: وابستگی یک مؤلفه غیرکلیدی به یک مؤلفه غیرکلیدی دیگر را وابستگی انتقالی می‌نامند.

در وابستگی‌های فوق، وابستگی انتقالی وجود ندارد. بنابراین جدول مربوطه در نرمال فرم سوم هم قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم BCNF را به صورت زیر بیان کرد:

□ جدول باید در نرمال فرم سوم باشد.

□ جدول باید فاقد وابستگی معکوس باشد.

وابستگی معکوس: وابستگی یک عضو کلید کاندید به عضو یک کلید کاندید دیگر یا مؤلفه غیرکلیدی را،

وابستگی معکوس می‌نامند. عضو کلید کاندید → عضو کلید کاندید

عضو کلید کاندید → غیرکلید

در وابستگی‌های فوق، وابستگی معکوس وجود دارد. بنابراین جدول مربوطه در نرمال فرم BCNF قرار ندارد.

در یک نگاه دیگر برای نرمال فرم BCNF می‌توان گفت، جدولی در نرمال فرم BCNF قرار دارد که همه‌ی

شروع‌های وابستگی‌ها، ابرکلید باشد. به بیان دیگر هرگاه سمت چپ همه وابستگی‌ها، ابرکلید باشد، آن‌گاه

آن جدول در BCNF قرار دارد که در وابستگی‌های فوق این چنین نیست. پس BCNF هم نیست.

در وابستگی‌های فوق، سمت چپ وابستگی‌های اول، دوم، سوم و چهارم ابرکلید است. اما سمت چپ

وابستگی‌های پنجم و ششم ابرکلید نیست.

بنابراین این جدول به دلیل نقض شرایط مربوطه، در نرمال فرم BCNF قرار ندارد. در نتیجه گزینه اول

پاسخ سوال خواهد بود.

۵۱- گزینه (۴) صحیح است.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم BCNF را به صورت زیر بیان کرد:

□ جدول باید در نرمال فرم سوم باشد.

□ جدول باید فاقد وابستگی معکوس باشد.

وابستگی معکوس: وابستگی یک عضو کلید کاندید به عضو یک کلید کاندید دیگر یا مؤلفه غیرکلیدی را،

وابستگی معکوس می‌نامند. عضو کلید کاندید → عضو کلید کاندید

عضو کلید کاندید → غیرکلید

در یک نگاه دیگر برای نرمال فرم BCNF می‌توان گفت، جدولی در نرمال فرم BCNF قرار دارد که همه‌ی

شروع‌های وابستگی‌ها، ابرکلید باشد. به بیان دیگر هرگاه سمت چپ همه وابستگی‌ها، ابرکلید باشد، آن‌گاه

آن جدول در BCNF قرار دارد.

طبق فرض سوال جدول $R(A, B, C, D, E)$ در نرمال فرم BCNF قرار دارد و داری تنها یک کلید کاندید به

صورت ABC می‌باشد، بنابراین هر ترکیبی از کلید کاندید ABC با صفات باقی‌مانده ابرکلید می‌سازد. برای

به‌دست آوردن ابرکلیدها، باید به کلید کاندید ABC، صفر یا بیش‌تر، از صفات باقی‌مانده را اضافه کرد.

بنابراین باید به کلید کاندید ABC، صفر، یک، یا دو مورد از صفات دیگر (D, E) اضافه کرد. برای این

کار 4 حالت، وجود دارد:

$$\binom{2}{0} + \binom{2}{1} + \binom{2}{2} = 2^2 = 4$$

ساخت ابرکلید: هیچ یا ترکیبی از صفات باقیمانده + کلید کاندید ABC = ابرکلید

DE	ترکیبات صفات باقی مانده	ابرکلیدها
0 0	تهی	ABC
0 1	E	ABCE
1 0	D	ABCD
1 1	DE	ABCDE

توجه: واضح است که 4 ابرکلید، ایجاد می‌گردد. (2^2)

حال وابستگی‌های مطرح شده در گزینه‌های صورت سوال را در نظر بگیرید:

BCE → D

ACE → D

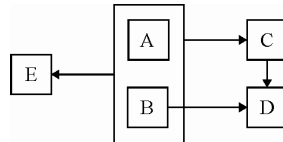
ABD → E

ABCD → E

بر اساس وابستگی‌های فوق، سمت چپ وابستگی‌های گزینه‌های اول، دوم و سوم ابرکلید نیست. اما سمت چپ وابستگی گزینه چهارم ابرکلید است، موجود در سطر سوم از جدول فوق. بنابراین از وابستگی‌های فوق، تنها سمت چپ وابستگی گزینه چهارم ابرکلید قرار دارد که پاسخ سوال نیز خواهد بود، به همین سادگی...

۵۲- گزینه (۱) صحیح است.

شکل زیر، نمودار وابستگی‌های تابعی صفات را نشان می‌دهد. پُر واضح است که کلید این جدول صفات AB است. بنابراین بدیهی است که جدول 2NF نیست به دلیل وجود وابستگی‌های بخشی $B \rightarrow D$ (وابستگی غیرکلید به عضو کلید کاندید) پس جدول مذکور 3NF و BCNF نیز نیست. پس گزینه‌های دوم، سوم و چهارم پاسخ سوال نیستند. بنابراین گزینه اول پاسخ سوال است.



به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیربدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E)$ داریم:

$$C \rightarrow D$$

$$B \rightarrow D$$

$$AB \rightarrow CE$$

$$ABCDE - CDE = AB$$

بنابر رابطه فوق صفت AB حتماً باید عضو کلید کاندید باشد.

بستار صفت AB به صورت زیر است:

$$\{AB\}^+ = \{A, B, C, D, E\}$$

بر اساس بستار فوق، صفات AB ، همه ستون‌ها را تولید می‌کند، پس صفات AB ، کلید کاندید می‌باشد.

قانون دوم ارسطو

هرگاه عضو کلید کاندید حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

از آن‌جا که در جدول مطرح شده، صفت چند مقداری و مرکب وجود ندارد، بنابراین این جدول در نرمال فرم اول قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

□ جدول باید در نرمال فرم اول باشد. (که هست، زیرا صفت چند مقداری و مرکب ندارد).

□ جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مؤلفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می‌نامند.

وابستگی‌های مطرح شده را در نظر بگیرید:

غیر کلید \widehat{C}	→	غیر کلید \widehat{D}	وابستگی انتقالی
جزء کلید \widehat{B}	→	غیر کلید \widehat{D}	وابستگی بخشی
کلید کاندید \widehat{AB}	→	غیر کلید \widehat{CE}	وابستگی تابعی کامل

در وابستگی‌های فوق، وابستگی بخشی $B \rightarrow D$ وجود دارد، بنابراین جدول مربوطه در نرمال فرم دوم قرار ندارد و به تبع در نرمال فرم سوم و BCNF هم قرار ندارد.

۵۳- گزینه (۱) صحیح است.

صورت سوال به این شکل است:

اگر در رابطه R ، تعداد ابرکلیدها با تعداد کلیدهای کاندید برابر باشد، آنگاه کدام مورد در خصوص رابطه R نادرست است؟

(۱) تنها یک خصیصه دارد.

گزینه اول نادرست است، زیرا در جدول تمام کلید، یک جدول فقط و فقط یک ابرکلید دارد و فقط و فقط هم یک کلید کاندید دارد. رابطه تمام کلید مثلاً ممکن است سه ستون داشته باشد، در این حالت یک جدول

فقط و فقط یک ابرکلید دارد و فقط و فقط هم یک کلید کاندید دارد. یعنی حداقل یک ابرکلید و یک کلید کاندید دارد و حداکثر هم یک ابرکلید و یک کلید کاندید دارد. همچنین جدولی هم که تنها یک خصیصه دارد حداقل یک ابرکلید و یک کلید کاندید دارد و حداکثر هم یک ابرکلید و یک کلید کاندید دارد، اما این حالت فقط برای یک جدول تک خصیصه‌ای نیست و همانطور که گفتیم در جدول تمام کلید تعداد ابرکلید و کلید کاندید یکسان، و برابر مقدار یک است. هر ترکیبی از صفات (ستون‌ها) که خاصیت کلیدی داشته باشد، یک ابرکلید است. و ابرکلیدی که عضو زائد نداشته باشد، کلید کاندید است.

۲) در فرم نرمال 3NF است.

گزینه دوم درست است، زیرا در حالتی که تعداد ابرکلیدها با تعداد کلیدهای کاندید برابر باشد، آن جدول یا تک خصیصه‌ای است و یا تمام کلید است که در هر دو حالت در فرم نرمال 3NF قرار دارد. به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

□ جدول باید در نرمال فرم دوم باشد.

□ جدول باید فاقد وابستگی انتقالی باشد.

وابستگی انتقالی: وابستگی یک مؤلفه غیرکلیدی به یک مؤلفه غیرکلیدی دیگر را وابستگی انتقالی می‌نامند. در حالت جدول تک خصیصه‌ای یا جدول تمام کلید، وابستگی بخشی وجود ندارد، پس نرمال فرم دوم برقرار است. همچنین در حالت جدول تک خصیصه‌ای یا جدول تمام کلید، وابستگی انتقالی هم وجود ندارد، پس نرمال فرم سوم هم برقرار است.

۳) همه خصیصه‌های R، NOT NULL هستند.

گزینه سوم درست است، زیرا در حالتی که تعداد ابرکلیدها با تعداد کلیدهای کاندید برابر باشد، آن جدول یا تک خصیصه‌ای است و یا تمام کلید است که در هر دو حالت همه خصیصه‌های جدول R باید NOT NULL باشد. مطابق تعریف قانون جامعیت موجودیت، هیچگاه نباید تمام یا بخشی از کلید کاندید مقدار NULL داشته باشد.

۴) تعداد وابستگی‌های تابعی نابدیهی R، صفر هستند.

گزینه چهارم درست است، زیرا در حالتی که تعداد ابرکلیدها با تعداد کلیدهای کاندید برابر باشد، آن جدول یا تک خصیصه‌ای است و یا تمام کلید است که در هر دو حالت تعداد وابستگی‌های تابعی نابدیهی R، صفر هستند. در جدول تک خصیصه‌ای یا تمام کلید، هیچ وابستگی غیربیدیهی (نابدیهی) همچون وابستگی تابعی بخشی، وابستگی تابعی انتقالی و وابستگی تابعی معکوس وجود ندارد. یعنی تعداد وابستگی‌های تابعی نابدیهی R، صفر هستند.

۵۴- گزینه (۳) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بیدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i - (\text{چپ}) - y_i - (\text{راست})]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(X, Y, Z)$ داریم:

$$Y \rightarrow Z$$

$$XZ \rightarrow Y$$

$$X \rightarrow Z$$

$$XYZ - YZ = X$$

بنابر رابطه فوق صفت X حتماً باید عضو کلید کاندید باشد. بستار صفت X به صورت زیر است:

$$\{X\}^+ = \{X, Y, Z\}$$

بر اساس بستار فوق، صفت X ، همه ستون‌ها را تولید می‌کند، پس صفت X ، کلید کاندید می‌باشد.

قانون دوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

حال یک‌بار دیگر وابستگی‌های مطرح شده برای رابطه $R(X, Y, Z)$ را در نظر بگیرید:

وابستگی انتقالی $Y \longrightarrow Z$ غیرکلید

وابستگی تابعی $XZ \longrightarrow Y$ غیرکلید

وابستگی کامل $X \longrightarrow Z$ غیرکلید

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم اول را به صورت زیر بیان کرد:

□ دارای حداقل یک کلید کاندید باشد.

□ همه خصیصه‌های آن غیرقابل تجزیه باشند (جدول باید فاقد خصیصه‌های مرکب باشد)

□ همه خصیصه‌های آن تک مقداری باشند (جدول باید فاقد خصیصه‌های چند مقداری باشد)

واضح است که جدول مطرح شده در نرمال فرم اول قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

□ جدول باید در نرمال فرم اول باشد.

□ جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مؤلفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می‌نامند.

مؤلفه غیرکلید: هر صفتی که عضو هیچ کلید کاندید نباشد، به عنوان مؤلفه غیرکلیدی نامیده می‌شود.

مؤلفه جزء کلید کاندید: هر صفتی که عضو حداقل یک کلید کاندید باشد، به عنوان مؤلفه جزء کلید نامیده می‌شود.

در وابستگی‌های فوق، وابستگی بخشی وجود ندارد. بنابراین جدول مربوطه در نرمال فرم دوم هم قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

□ جدول باید در نرمال فرم دوم باشد.

□ جدول باید فاقد وابستگی انتقالی باشد.

وابستگی انتقالی: وابستگی یک مؤلفه غیرکلیدی به یک مؤلفه غیرکلیدی دیگر را وابستگی انتقالی می‌نامند.

در وابستگی های فوق، وابستگی انتقالی وجود دارد. بنابراین جدول مربوطه در نرمال فرم سوم قرار ندارد. و به تبع در نرمال فرم BCNF هم قرار ندارد.

بطور کلی می توان شروط قرار داشتن یک جدول در نرمال فرم BCNF را به صورت زیر بیان کرد:

□ جدول باید در نرمال فرم سوم باشد.

□ جدول باید فاقد وابستگی معکوس باشد.

وابستگی معکوس: وابستگی یک عضو کلید کاندید به عضو یک کلید کاندید دیگر یا مؤلفه غیرکلیدی راه،

وابستگی معکوس می نامند. عضو کلید کاندید → عضو کلید کاندید

عضو کلید کاندید → غیرکلید

در وابستگی های فوق، وابستگی معکوس وجود ندارد. اما جدول فوق در نرمال فرم سوم قرار ندارد، بنابراین جدول مربوطه در نرمال فرم BCNF هم قرار ندارد.

در یک نگاه دیگر برای نرمال فرم BCNF می توان گفت، جدولی در نرمال فرم BCNF قرار دارد که همگی شروع های وابستگی ها، ابرکلید باشد. به بیان دیگر هرگاه سمت چپ همه وابستگی ها، ابرکلید باشد، آن گاه آن جدول در BCNF قرار دارد که در وابستگی های فوق این چنین نیست. پس BCNF هم نیست.

در وابستگی های فوق، سمت چپ وابستگی های دوم و سوم ابرکلید است. اما سمت چپ وابستگی اول ابرکلید نیست.

بنابراین این جدول به دلیل نقض شرایط مربوطه، در نرمال فرم BCNF قرار ندارد. در نتیجه گزینه دوم گزاره درستی است.

گزینه اول گزاره درستی است، زیرا صفت X در سمت راست هیچ یک از وابستگی های تابعی مطرح شده در صورت سوال نیست، پس از روی صفات دیگر قابل دستیابی نیست و بنابراین حتماً باید در کلید کاندید حضور داشته باشد. بعلاوه، با داشتن X می توان تمام صفات های دیگر را به دست آورد، پس $\{X\}$ تنها کلید کمینه است و بنابراین ستون X تنها کلید کاندید رابطه R است.

گزینه دوم نیز گزاره درستی است، زیرا با توجه به وابستگی $Y \rightarrow Z$ وابستگی انتقالی یعنی وابستگی غیرکلید به غیرکلید وجود دارد، بنابراین رابطه R در سطح نرمال فرم سوم قرار ندارد و به تبع همین موضوع در سطح نرمال فرم BCNF هم قرار ندارد.

گزینه سوم گزاره نادرستی است، زیرا وابستگی سوم یعنی وابستگی $X \rightarrow Z$ اضافه و قابل حذف نیست، چون با داشتن X نمی توان از روی وابستگی های تابعی باقی مانده به Z رسید. در واقع وابستگی $X \rightarrow Z$ یک وابستگی اصلی از مجموعه وابستگی های بهینه است و یک وابستگی فرعی نیست.

گزینه چهارم نیز گزاره نادرستی است، زیرا ستون X در وابستگی دوم، مشخصه اضافه و قابل حذف نیست. چون ستون X توسط ستون دیگری تولید نمی شود یعنی در سمت راست هیچکدام از وابستگی های مطرح شده در صورت سوال قرار ندارد که بتوانیم ستون X را حذف کنیم. به عبارت دیگر اگر ستون X توسط ستون Z تولید می شد، آنگاه ستون X در وابستگی دوم، مشخصه اضافه و قابل حذف می بود که اینطور نیست.

توجه: البته اگر طراح محترم در گزینه چهارم به جای ستون X از ستون Z استفاده می کرد که ماهم فکر می کنیم همین بوده است اما در مرحله حروفچینی سوال، ستون Z به صورت X حروفچینی شده است. آنگاه گزینه چهارم هم گزاره درستی می بود و همان گزینه سوم در بین چهار گزینه فقط گزاره ای نادرست می شد و پاسخ سوال هم همان گزینه سوم و مد نظر طراح محترم می شد.

توجه: دقت کنید که با توجه به وابستگی $X \rightarrow Z$ با داشتن X می‌توان Z را به دست آورد. پس با حذف Z از وابستگی $XZ \rightarrow Y$ اطلاعاتی از بین نمی‌رود و حالت بهینه و کمینه وابستگی دوم به فرم $X \rightarrow Y$ می‌باشد. دقت کنید که عامل حذف ستون X در وابستگی دوم، وجود و حضور وابستگی اصلی سوم است. وابستگی اصلی سوم تحت هیچ شرایطی قابل حذف نیست.

توجه: سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه سوم را به عنوان پاسخ اعلام کرده بود.

۵۵- گزینه (۳) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیربدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [x_i \text{ (چپ)} - y_i \text{ (راست)}]$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E, P, G)$ داریم:

$$AB \rightarrow CD$$

$$DE \rightarrow P$$

$$C \rightarrow E$$

$$P \rightarrow C$$

$$B \rightarrow G$$

$$ABCDEPG - CDEPG = AB$$

بنابر رابطه فوق صفت AB حتماً باید عضو کلید کاندید باشد.

$$\{AB\}^+ = \{A, B, C, D, E, P, G\}$$

بستار صفت AB به صورت مقابل است:

براساس بستار فوق، صفات AB ، همه ستون‌ها را تولید می‌کند، پس صفات AB ، کلید کاندید می‌باشد.

قانون دوم ارسطو

هرگاه عضو کلید کاندید حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

از آن‌جا که در جدول مطرح شده، صفت چند مقداری و مرکب وجود ندارد، بنابراین این جدول در نرمال فرم اول قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

□ جدول باید در نرمال فرم اول باشد. (که هست، زیرا صفت چند مقداری و مرکب ندارد).

□ جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مؤلفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می نامند. حال یک بار دیگر وابستگی های مطرح شده برای رابطه $R(A, B, C, D, E, P, G)$ را در نظر بگیرید:

وابستگی کامل	غیرکلید $AB \rightarrow CD$
وابستگی انتقالی	غیرکلید $DE \rightarrow P$
وابستگی انتقالی	غیرکلید $C \rightarrow E$
وابستگی انتقالی	غیرکلید $P \rightarrow C$
وابستگی بخشی	غیرکلید $B \rightarrow G$

در وابستگی های فوق، وابستگی بخشی $B \rightarrow G$ وجود دارد، بنابراین جدول مربوطه در نرمال فرم دوم قرار ندارد و به تبع در نرمال فرم سوم و BCNF هم قرار ندارد.

توجه: سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه سوم را به عنوان پاسخ اعلام کرده بود.

۵۶- گزینه (۴) صحیح است.

با توجه به وابستگی های مطرح شده برای رابطه $R(A, B, C, D)$ داریم:

$$B \rightarrow C$$

$$CD \rightarrow B$$

$$ABC \rightarrow D$$

بنابر رابطه فوق صفات AD حتماً باید عضو کلید کاندید باشد. بستر صفات AD به صورت زیر است:

$$\{A, D\}^+ = \{A, D\}$$

براساس بستر فوق، صفات AD ، فقط ستون های AD را تولید می کند، پس صفات AD فقط عضو کلید کاندید می باشد و کلید کاندید نمی باشد.

قانون سوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول، برخی از ستون ها را تولید کند، بدین معنی است که، جدول مورد نظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت.

براساس بستر فوق، صفات AD به عنوان عضو کلید کاندید همه ستون ها را تولید نمی کند، بنابراین مطابق قانون سوم ارسطو باید صفاتی در کنار صفت AD قرار گیرد تا کلید کاندید تولید گردد. این صفات کناری از صفات باقی مانده انتخاب می گردند، صفات باقی مانده عبارتند از B و C بنابراین با ترکیب صفات B یا C با صفت AD کلیدهای کاندید تولید می گردند.

$$\{ADB\}^+ = \{A, D, B, C\}$$

$$\{ADC\}^+ = \{A, D, C, B\}$$

پس ترکیبات صفات ADB و ADC کلیدهای کاندید جدول R هستند.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده در مجموع دارای دو کلید کاندید است. که مطابق قانون سوم ارسطو، عضو کلید کاندید AD، در بین هر دو کلید کاندید به طور مشترک قرار دارد.

حال یک‌بار دیگر وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D)$ را در نظر بگیرید:

وابستگی معکوس عضو کلید کاندید $C \rightarrow B$ عضو کلید کاندید

وابستگی معکوس عضو کلید کاندید $B \rightarrow CD$ عضو کلید کاندید

در وابستگی‌های فوق، وابستگی بخشی وجود ندارد. بنابراین جدول مربوطه در نرمال فرم دوم هم قرار دارد.

در وابستگی‌های فوق، وابستگی انتقالی وجود ندارد. بنابراین جدول مربوطه در نرمال فرم سوم هم قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم BCNF را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم سوم باشد.
- جدول باید فاقد وابستگی معکوس باشد.

وابستگی معکوس: وابستگی یک عضو کلید کاندید به عضو یک کلید کاندید دیگر یا مؤلفه غیرکلیدی را، وابستگی معکوس می‌نامند.

عضو کلید کاندید \rightarrow عضو کلید کاندید

عضو کلید کاندید \rightarrow غیرکلید

در وابستگی‌های فوق، وابستگی معکوس وجود دارد. بنابراین جدول مربوطه در نرمال فرم BCNF قرار ندارد.

در یک نگاه دیگر برای نرمال فرم BCNF می‌توان گفت، جدولی در نرمال فرم BCNF قرار دارد که هم‌هی شروع‌های وابستگی‌ها، ابرکلید باشد. به بیان دیگر هرگاه سمت چپ همه وابستگی‌ها، ابرکلید باشد، آن‌گاه آن جدول در BCNF قرار دارد که در وابستگی‌های فوق این چنین نیست. پس BCNF هم نیست.

در وابستگی‌های فوق، سمت چپ وابستگی‌های اول و دوم ابرکلید نیست. بنابراین این جدول به دلیل نقض شرایط مربوطه، در نرمال فرم BCNF قرار ندارد. در نتیجه گزینه چهارم پاسخ سوال است.

۵۷- گزینه (۲) صحیح است.

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E)$ داریم:

$$A \rightarrow B, C$$

$$B, C \rightarrow A, D$$

$$D \rightarrow E$$

$$ABCDE - ABCDE = \text{تهی}$$

قانون چهارم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول، تهی گردد، بدین معنی است که، جدول فوق چندین

کلید کاندید دارد، که هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید آن وجود ندارد. بنابراین باید کلید کاندید با بررسی دقیق بر روی مجموعه وابستگی کشف گردد.

$$\{A\}^+ = \{A, B, C, D, E\}$$

بر اساس بستار فوق، صفت A، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفت A، کلید کاندید می‌باشد.

حال کلید کاندید A را در نظر بگیرید، از آنجا که مطابق وابستگی $BC \rightarrow A$ ، صفات BC، صفت A را می‌دهد. می‌توان به جای صفت A در کلید کاندید A، صفات BC را قرار داد. چون مجدداً مطابق وابستگی $BC \rightarrow A$ ، صفت A را می‌دهد که منجر به ایجاد کلید کاندید BC می‌گردد. مطابق الگوی زیر:

$$A \Rightarrow BC$$

↑

BC

$$\{BC\}^+ = \{B, C, A, D, E\}$$

بر اساس بستار فوق، صفات BC، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفات BC کلید کاندید می‌باشد.

بررسی مجموعه وابستگی‌های مطرح شده نشان داد که (A) و (B,C) به عنوان کلیدهای کاندید برای رابطه R محسوب می‌شوند.

توجه: همان‌طور که مشاهده می‌شود، مطابق قانون چهارم ارسطو، هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید فوق وجود ندارد.

واضح است که جدول مطرح شده در فرم اول نرمال قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم اول باشد.
- جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مولفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می‌نامند.

مولفه غیرکلیدی: هر صفتی که عضو هیچ کلید کاندیدی نباشد، به عنوان مؤلفه غیرکلیدی نامیده می‌شود.

مؤلفه جزء کلید کاندید: هر صفتی که عضو حداقل یک کلید کاندید باشد، به عنوان مؤلفه جزء کلید نامیده می‌شود.

وابستگی‌های مطرح شده را در نظر بگیرید:

وابستگی کامل کلید کاندید $A \rightarrow BC$

وابستگی کامل کلید کاندید $BC \rightarrow A$

وابستگی کامل غیرکلید $BC \rightarrow D$

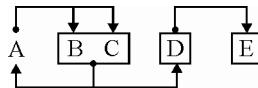
وابستگی انتقالی غیرکلید $D \rightarrow E$

در وابستگی‌های فوق، وابستگی بخشی وجود ندارد. بنابراین جدول مربوطه در نرمال فرم دوم هم قرار دارد.

به طور کلی می توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم دوم باشد.
- جدول باید فاقد وابستگی انتقالی باشد.

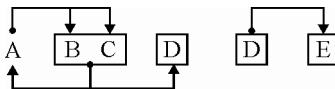
وابستگی انتقالی: وابستگی یک مؤلفه غیرکلیدی به یک مؤلفه غیرکلیدی دیگر را وابستگی انتقالی می نامند. در جدول زیر:



وابستگی تابعی زیر:

$D \rightarrow E$

به عنوان وابستگی انتقالی محسوب می گردد. بنابراین جدول R در نرمال فرم سوم قرار ندارد. بنابراین برای قرار دادن جدول R، در نرمال فرم سوم، باید وابستگی های انتقالی از جدول R خارج گردند.



بنابراین جدول R به دو جدول کوچکتر تجزیه می گردد.

$R_1(A, B, C, D)$ $R_2(D, E)$

با توجه به کلیدهای کاندید در دو جدول R_1 و R_2 می توان دریافت که این دو جدول فاقد وابستگی بخشی و انتقالی هستند و بنابراین در نرمال فرم سوم قرار دارند. همچنین از آن جا که سمت چپ تمام وابستگی های دو جدول R_1 و R_2 ، ابرکلید است، بنابراین دو جدول R_1 و R_2 در سطح BCNF نیز قرار دارند. به این ترتیب با تجزیه جدول R به دو جدول R_1 و R_2 می توان جدول R را در نرمال فرم سوم و سطح BCNF قرار داد.

۵۸- گزینه (۴) صحیح است.

ویژگی های رابطه در مدل رابطه ای

- ۱- در رابطه، تاپل تکراری وجود ندارد. زیرا رابطه یک مجموعه است و مجموعه در ریاضیات طبق تعریف تاپل تکراری ندارد. بنابراین در جدول هم رکوردهای تکراری نداریم.
 - ۲- تاپل ها در رابطه نظم مکانی ندارند. زیرا مجموعه نظم ندارد. بنابراین ترتیب رکوردها در جدول اهمیت ندارد.
 - ۳- ترتیب صفات خاصه در یک تاپل مهم نیست. البته به شرطی که ترتیب صفات خاصه در تمام تاپل های موجود در یک رابطه با یکدیگر سازگار باشند. بنابراین ترتیب فیلدها در یک جدول نیز مهم نیست، نظم ندارد و می توان آنرا جابه جا کرد. البته به شرطی که ترتیب فیلدها در تمام رکوردهای موجود در یک جدول با یکدیگر سازگار باشند.
- برای مثال رابطه R_3 به عنوان یک زیر مجموعه دلخواه از حاصلضرب دکارتی سه دامنه D_1 ، D_2 و D_3 را در نظر بگیرید:

$R_3 = \{(a, x, 1), (a, x, 2), (b, x, 1), (b, x, 2)\}$

اگر ترتیب تاپل های رابطه R_7 به صورت زیر باشد:

$$R_7 = \{(x, 1, a), (x, 2, a), (x, 1, b), (x, 2, b)\}$$

رابطه R_7 معادل رابطه R_3 است، زیرا ترتیب صفات خاصه در تمام تاپل های موجود در رابطه R_7 با یکدیگر سازگار است. ($R_3 = R_7$)

اگر ترتیب تاپل های رابطه R_8 به صورت زیر باشد:

$$R_8 = \{(x, 1, a), (2, x, a), (1, b, x), (x, 2, b)\}$$

رابطه R_8 معادل رابطه R_3 نیست، زیرا ترتیب صفات خاصه در تمام تاپل های موجود در رابطه R_8 با یکدیگر سازگار نیست. ($R_3 \neq R_8$)

۴- همه ی مقادیر صفات خاصه در تاپل ها تجزیه ناپذیرند. به عبارت دیگر در مدل رابطه ای نمی توان فیلد مرکبی که از فیلدهای ساده تشکیل شده است، تعریف نمود. یعنی فیلدهای موجود در جداول مدل رابطه ای نباید مرکب مثل تاریخ تولد و آدرس باشد و نباید چندمقداری مثل مدرک تحصیلی باشد. به عبارت دیگر هیچ رابطه ای نمی تواند شامل تاپل های تو در تو، تاپل تو تاپل یا جدول تو جدول باشد، به بیان دیگر مولفه های هر یک از تاپل های رابطه نباید مرکب و چندمقداری باشد، به عبارت ساده تر هر مولفه یک تاپل نمی تواند خودش یک تاپل باشد.

توجه: بهترین راه نمایش و پیاده سازی رابطه استفاده از جدول است، بنابراین جدول معادل رابطه در نظر گرفته می شود.

گزینه اول نادرست است. زیرا، اگر هیچ یک از صفات ساده یک رابطه چندمقداری و مرکب نباشد، این مربوط به فقط و فقط نرمال فرم اول (Inf) است. در حالی که نرمال سازی تا سطح سوم (3nf) اجباری است. **گزینه دوم نادرست است.** زیرا، ابرکلیدی که عضو زائد نداشته باشد، کلید کاندید است، به عبارت دیگر ابرکلید کمینه را کلید کاندید می گویند. منظور از ابرکلید کمینه، ابرکلیدی نیست که کمترین تعداد صفت را داشته باشد، بلکه منظور ابرکلیدی است که صفت زائد نداشته باشد. بنابراین کلید کاندید یک رابطه خودش به طور ذاتی کمینه است و نمی تواند کاهش پذیر باشد.

گزینه سوم نادرست است. زیرا، تاپل ها در رابطه نظم مکانی ندارند. زیرا مجموعه نظم ندارد. بنابراین ترتیب رکوردها در جدول اهمیت ندارد.

گزینه چهارم درست است. زیرا، در رابطه، تاپل تکراری وجود ندارد. زیرا رابطه یک مجموعه است و مجموعه در ریاضیات طبق تعریف تاپل تکراری ندارد. بنابراین در جدول هم رکوردهای تکراری نداریم.

۵۹- گزینه (۱) صحیح است.

صورت سوال ابهام و خطا دارد، کلمه فارسی «گمشدگی» معادل کلمه انگلیسی «lossless» نیست، معنی درست و فارسی کلمه «lossless» عبارت «بدون گمشدگی» است. و دیگر اینکه صفت F موجود در جدول R در هیچ یک از گزینه ها نیامده است. بنابراین سوال قابل حل نیست.

توجه: سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه اول را به عنوان پاسخ اعلام کرده بود. که کار درستی نبوده است و سوال باید حذف می شد که نشد.

شرایط تجزیه مطلوب به صورت زیر است:

۱- شرط لازم (Nonloss join): دو جدول الحاق‌پذیر باشند (صفت مشترک در دو جدول، حداقل در یکی کلید کاندید باشد) این شرط، مهم‌ترین شرط نرمال‌سازی است، زیرا شرط لازم است.

۲- شرط کافی (حفظ وابستگی‌های تابعی): وابستگی‌های تابعی جدول پایه قابل استنتاج از روی وابستگی‌های تابعی جداول حاصل از تجزیه باشد یا در آن‌ها موجود باشد.

توجه: در تجزیه بدون گمشدگی (Lossless)، صفت مشترک در یکی از روابط باید کلید کاندید باشد. همچنین هنگامی که در فرآیند نرمال‌سازی یک جدول تجزیه می‌شود، این تجزیه باید به گونه‌ای انجام شود که امکان بازسازی محتوایی و ساختاری جدول اولیه بر اساس جداول کوچکتر فراهم باشد، به این صورت که با الحاق جداول کوچکتر به یکدیگر دقیقاً جدول اولیه (پایه) ایجاد گردد (نه سطر یا ستونی از جدول اولیه از دست برود و نه سطر یا ستونی به جدول اولیه اضافه شود). چنین تجزیه‌ای را تجزیه بدون گمشدگی (Lossless) می‌گویند.

اگر صورت سوال را به فرم زیر اصلاح کنیم، یعنی صفت F از جدول R حذف شود و معنی «lossless» را بدون گمشدگی معنی کنیم:

رابطه $R = (A, B, C, D, E)$ را در نظر بگیرید. اگر مجموعه وابستگی‌های تابعی F روی R برقرار باشد، کدام یک از تجزیه‌های زیر بدون گمشدگی (lossless) است؟

آنگاه بررسی گزینه‌ها بر اساس وابستگی‌های مطرح شده به صورت زیر خواهد بود:

$A \rightarrow BC$
 $CD \rightarrow E$
 $B \rightarrow D$
 $E \rightarrow A$

گزینه اول:

$R_1(A, B, C)$
 $\{C\}^+ = \{C, \text{Stop}\}$
 $R_2(C, D, E)$
 $\{C\}^+ = \{C, \text{Stop}\}$

در گزینه اول، ستون مشترک C در $R_1(A, B, C)$ یا $R_2(C, D, E)$ کلید کاندید نیست. بنابراین «دارای گمشدگی» است.

گزینه دوم:

$R_1(A, B, C)$
 $A \rightarrow BC$
 $\{A\}^+ = \{A, B, C\}$
 $R_2(A, D, E)$
 $\{A\}^+ = \{A, \text{Stop}\}$

در گزینه دوم، ستون مشترک A در $R_1(A, B, C)$ یا $R_2(A, D, E)$ در جدول R_1 کلید کاندید است. بنابراین

«بدون گمشدگی» (Lossless) است.

گزینه سوم:

$$R_1(E, B, C)$$

$$\{E\}^+ = \{E, \text{Stop}\}$$

$$R_2(E, D, A)$$

$$E \rightarrow A$$

$$\{E\}^+ = \{E, A, \text{Stop}\}$$

در گزینه سوم، ستون مشترک E در $R_1(E, B, C)$ یا $R_2(E, D, A)$ کلید کاندید نیست. بنابراین «دارای گمشدگی» است.

گزینه چهارم:

$$R_1(A, B, C)$$

$$\{BC\}^+ = \{B, C, \text{Stop}\}$$

$$R_2(B, C, D, E)$$

$$CD \rightarrow E$$

$$B \rightarrow D$$

$$\{BC\}^+ = \{B, C, D, E\}$$

در گزینه چهارم، ستون مشترک BC در $R_1(A, B, C)$ یا $R_2(B, C, D, E)$ در جدول R_2 کلید کاندید است. بنابراین «بدون گمشدگی» (Lossless) است.

۶۰- گزینه (۲ و ۳) صحیح است.

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E, F, G)$ داریم:

$$AF \rightarrow BE$$

$$FC \rightarrow DE$$

$$F \rightarrow CD$$

$$D \rightarrow E$$

$$C \rightarrow A$$

$$ABCDEF - ABCDE = FG$$

بنابر رابطه فوق صفات FG حتماً باید عضو کلید کاندید باشد. بستار صفات FG به صورت زیر است:

$$\{FG\}^+ = \{F, G, C, D, E, A, B\}$$

براساس بستار فوق، صفات FG، همه ستون‌ها را تولید می‌کند، پس صفات FG کلید کاندید می‌باشد.

قانون دوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

جدولی در سطح نرمال فرم اول قرار دارد که صفت چند مقداری و مرکب نداشته باشد. از آنجا که در جدول R، صفت چند مقداری و مرکب وجود ندارد، بنابراین این جدول در نرمال فرم اول قرار دارد. با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E, F, G)$ داریم:

$$AF \rightarrow BE$$

$$FC \rightarrow DE$$

$$F \rightarrow CD$$

$$D \rightarrow E$$

$$C \rightarrow A$$

توجه: به طور کلی در نرمال‌سازی پس از کشف کلید کاندید، جهت کشف وابستگی بخشی و انتقالی و کنترل بهتر وابستگی‌ها می‌توان وابستگی‌ها را بهینه و کمینه کرد، اما با عدم حذف وابستگی‌های اصلی و عدم درج وابستگی‌ها فرعی. در اغلب موارد خود وابستگی‌ها گویای وابستگی بخشی و انتقالی هستند و نیاز به کمینه‌سازی وابستگی‌ها هم نیست. مهمترین عنصر نرمال‌سازی صحیح، کشف دقیق و درست کلید کاندید توسط قوانین چهارگانه است.

توجه: در وابستگی $AF \rightarrow BE$ از وابستگی‌های $F \rightarrow A$ $\left\{ \begin{array}{l} F \rightarrow C \\ C \rightarrow A \end{array} \right.$ ستون F می‌دهد A را، پس در سمت

چپ وابستگی $AF \rightarrow BE$ ستون A زائد است، چون F می‌تواند خود A را تولید کند، پس به جای وابستگی $AF \rightarrow BE$ می‌توان وابستگی کمینه $F \rightarrow BE$ را جایگزین کرد.

توجه: در وابستگی $FC \rightarrow DE$ از وابستگی $F \rightarrow CD$ ستون F می‌دهد C را، پس در سمت چپ وابستگی $FC \rightarrow DE$ ستون C زائد است، چون F می‌تواند خود C را تولید کند، پس به جای وابستگی $FC \rightarrow DE$ می‌توان وابستگی کمینه $F \rightarrow DE$ را جایگزین کرد.

پس از حذف وابستگی‌های فرعی و قابل تولید از سایر وابستگی‌ها، وابستگی‌های کمینه بصورت زیر است:

$$F \rightarrow B$$

$$F \rightarrow C$$

$$F \rightarrow D$$

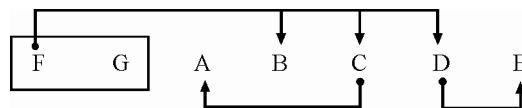
$$D \rightarrow E$$

$$C \rightarrow A$$

توجه: بر اساس وابستگی‌های فوق واضح است که صفت F همه ستون‌ها را به طور مستقیم و غیرمستقیم بجز صفت G را تولید می‌کند، بستار صفت F به صورت زیر است:

$$\{F\}^+ = \{F, C, D, E, A, B\}$$

نمودار وابستگی‌های جدول R به صورت زیر است:



به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

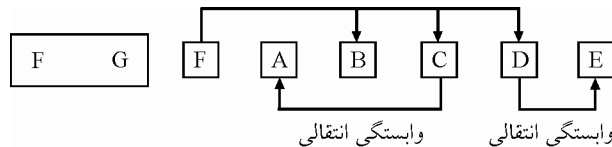
جدول باید در نرمال فرم اول باشد.

جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مؤلفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می نامند. با توجه به این که ترکیب صفات FG کلید کاندید رابطه R می باشد، وابستگی های تابعی زیر:

$F \rightarrow B$
 $F \rightarrow C$
 $F \rightarrow D$

به عنوان وابستگی بخشی محسوب می گردند. بنابراین جدول R در نرمال فرم دوم قرار ندارد. بنابراین برای قرار دادن جدول R، در نرمال فرم دوم، باید وابستگی های بخشی از جدول R خارج گردند.



بنابراین جدول R به دو جدول کوچکتر تجزیه می گردد:

$R_1(F, G)$ $R_2(\underline{F}, A, B, C, D, E)$

با توجه به کلید کاندید FG در جدول R_1 می توان دریافت که این جدول فاقد وابستگی بخشی و انتقالی است، بنابراین در نرمال فرم سوم قرار دارد. همچنین از آن جا که سمت چپ تمام وابستگی های جدول R_1 ، ابرکلید است، بنابراین جدول R_1 در سطح BCNF نیز قرار دارد. از نگاه دیگر می توان گفت جدول R_1 تمام کلید است پس BCNF است، از منظری دیگر نیز می توان گفت هر جدول با 2 ستون همواره و تحت هر شرایطی قطعاً BCNF است. مطابق خواسته سوال راه حل را تا سطح 3NF برای جدول R_2 ادامه می دهیم.

به طور کلی می توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

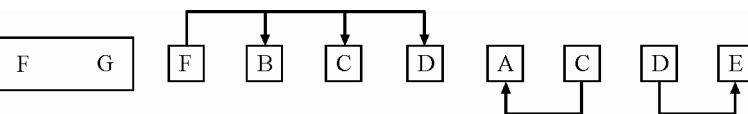
جدول باید در نرمال فرم دوم باشد.

جدول باید فاقد وابستگی انتقالی باشد.

وابستگی انتقالی: وابستگی یک مؤلفه غیرکلیدی به یک مؤلفه غیرکلیدی دیگر را وابستگی انتقالی می نامند. در جدول R_2 وابستگی تابعی زیر:

$D \rightarrow E$

$C \rightarrow A$



بنابراین جدول R_2 به سه جدول کوچکتر تجزیه می گردد.

$R_{21}(\underline{F}, B, C, D)$ $R_{22}(\underline{C}, A)$ $R_{23}(\underline{D}, E)$

با توجه به کلیدهای کاندید در سه جدول R_{21} و R_{22} و R_{23} می توان دریافت که این سه جدول فاقد وابستگی بخشی و انتقالی هستند و بنابراین در نرمال فرم سوم قرار دارند. همچنین از آن جا که سمت چپ تمام وابستگی های سه جدول R_{21} و R_{22} و R_{23} ، ابرکلید است، بنابراین سه جدول R_{21} و R_{22} و R_{23} در سطح BCNF نیز قرار دارند. به این ترتیب با تجزیه جدول R به چهار جدول R_1 ، R_{21} ، R_{22} و R_{23}

R23 می‌توان جدول R را در نرمال فرم سوم و سطح BCNF قرار داد.
توجه: سازمان سنجش آموزش کشور در کلید اولیه خود ابتدا گزینه دوم را به عنوان پاسخ اعلام نمود، سپس در کلید نهایی نظر خود را عوض کرد و گزینه دوم و سوم را به عنوان پاسخ اعلام کرد، که عمل نادرستی را انجام داده است. چون فقط و فقط همان گزینه سوم پاسخ سوال و درست است.

تست‌های فصل دهم: شاخص گذاری اطلاعات

- ۱- جدول T را در نظر بگیرید که روی ستون a، Clustered Index شده است. کدام مورد درست است؟
(مهندس کامپیوتر- دولتی ۹۷)
- ۱) اعمال سیاست شاخص گذاری، تاثیری بر حجم اطلاعات ذخیره شده بر روی دیسک ندارد.
 - ۲) با اعمال سیاست شاخص گذاری، پاسخ به Range Query های مرتبط به a، با سرعت بیشتری انجام می شود.
 - ۳) با اعمال سیاست شاخص گذاری، پاسخ به Equality Query های مرتبط به a، با سرعت کمتری انجام می شود.
 - ۴) همه موارد درست هستند.

- ۲- در مورد شاخص گذاری (Indexing)، کدام مورد نادرست است؟
(مهندسی IT-دولتی ۹۷)
- ۱) B⁺ Tree index برای پاسخ به Range Query مفید است.
 - ۲) هر جدول می تواند حداکثر یک شاخص کلاستر شده داشته باشد.
 - ۳) شاخص های از نوع Hash برای پاسخ به Range Query مفید هستند.
 - ۴) هرچه تعداد شاخص های یک جدول بیشتر باشد، سرعت Insert در آن جدول کمتر می شود.

پاسخ تست‌های فصل دهم: شاخص‌گذاری اطلاعات

۱- گزینه (۲) صحیح است.

دو هدف اصلی سیستم ذخیره و بازیابی اطلاعات در پایگاه داده‌ها، اول سرعت عملیات در ذخیره و بازیابی اطلاعات و دوم صرفه‌جویی در مصرف حافظه است. برای مثال کاهش افزونگی محتوایی (طبیعی) توسط نرمال‌سازی جداول منجر به کاهش میزان حافظه مصرفی می‌شود. عمل واکنشی تک تک رکوردها وقت‌گیر است، برای رفع این عیب شاخص یا Index ابداع شد. برای اینکه بازیابی داده‌ها با سرعت و کارایی بیشتر صورت گیرد، از شاخص استفاده می‌شود. شاخص ساختمان داده‌ای است که سیستم مدیریت پایگاه داده‌ها به کمک آن رکوردهای خاص را در یک فایل با سرعت زیاد پیدا می‌کند و به این ترتیب سرعت پاسخ به پرس و جوها افزایش می‌یابد. هر ساختار شاخص، حاوی رکوردهایی است که در هر رکورد یک مقدار کلیدی (کلید جستجو) و آدرس منطقی رکوردهای فایل داده‌ای نگهداری می‌شود.

اغلب سیستم‌های مدیریت پایگاه داده‌ها، از ساختار درخت برای ایجاد شاخص‌ها استفاده می‌کنند. عمق درخت، بیشترین تعداد سطوح از ریشه به برگ است. عمق ممکن است در مسیرهای مختلف از ریشه تا برگ متفاوت باشد. و همچنین عمق ممکن است در مسیرهای مختلف از ریشه تا برگ یکسان باشد، که در این شرایط با درخت متوازن و B-Tree و B⁺-Tree مواجه هستیم. هرچه درجه‌ی گره‌های درخت بیشتر شود، درخت پهن‌تر و کم‌عمق‌تری ایجاد می‌شود. از آنجاییکه زمان دسترسی در یک درخت، بیشتر وابسته به عمق درخت است تا پهنای آن، پس ساخت درخت پهن و کم‌عمق در ایجاد شاخص باعث افزایش سرعت جستجو می‌شود، ساختارهای B-Tree و B⁺-Tree درخت‌هایی با عمق کم و پهنای زیاد هستند. ساختار B⁺ Tree index برای پاسخ به Range Query و Equality Query مفید است.

صورت سوال به این شکل است:

جدول T را در نظر بگیرید که روی ستون a، Clustered Index شده است. کدام مورد درست است؟

۱) اعمال سیاست شاخص‌گذاری، تاثیری بر حجم اطلاعات ذخیره‌شده بر روی دیسک ندارد.

گزینه اول نادرست است، زیرا تعریف و نگهداری شاخص موجب تحمیل سربار حافظه‌ای به سیستم می‌شود. شاخص بر روی دیسک نگهداری می‌شود و هنگام استفاده به حافظه اصلی آورده می‌شود، بنابراین اعمال سیاست شاخص‌گذاری، بر افزایش حجم اطلاعات ذخیره‌شده بر روی حافظه اصلی و هارد دیسک تاثیر دارد.

۲) با اعمال سیاست شاخص‌گذاری، پاسخ به Range Query های مرتبط به a، با سرعت بیشتری انجام می‌شود.

گزینه دوم درست است، زیرا Clustered Index توسط ساختار B⁺ Tree پیاده‌سازی می‌شود. در ساختار B⁺ Tree جستجو و بازیابی نقطه‌ای (Equality Query) از مرتبه $O(\log_t n)$ و جستجو و بازیابی بازه‌ای (Range Query) از مرتبه $O(\log_t n + k)$ است. بنابراین اعمال سیاست شاخص‌گذاری توسط

ساختار B^+ Tree باعث می شود Range Query ها و Equality Query های مرتبط، با سرعت بیشتری انجام شود.

(۳) با اعمال سیاست شاخص گذاری، پاسخ به Equality Query های مرتبط به a ، با سرعت کمتری انجام می شود.

گزینه سوم نادرست است، زیرا Clustered Index توسط ساختار B^+ Tree پیاده سازی می شود. در ساختار B^+ Tree جستجو و بازیابی نقطه ای (Equality Query) از مرتبه $O(\log n)$ و جستجو و بازیابی بازه ای (Range Query) از مرتبه $O(\log n + k)$ است. بنابراین اعمال سیاست شاخص گذاری توسط ساختار B^+ Tree باعث می شود Range Query ها و Equality Query های مرتبط، با سرعت بیشتری انجام شود.

(۴) همه موارد درست هستند.

گزینه چهارم نادرست است، زیرا فقط گزینه دوم درست است.

۲- گزینه (۳) صحیح است.

دو هدف اصلی سیستم ذخیره و بازیابی اطلاعات در پایگاه داده ها، اول سرعت عملیات در ذخیره و بازیابی اطلاعات و دوم صرفه جویی در مصرف حافظه است. برای مثال کاهش افزونگی محتوایی (طبیعی) توسط نرمال سازی جداول منجر به کاهش میزان حافظه مصرفی می شود. عمل واکنشی تک تک رکوردها وقت گیر است، برای رفع این عیب شاخص یا Index ابداع شد. برای اینکه بازیابی داده ها با سرعت و کارایی بیشتر صورت گیرد، از شاخص استفاده می شود. شاخص ساختمان داده ای است که سیستم مدیریت پایگاه داده ها به کمک آن رکوردهای خاص را در یک فایل با سرعت زیاد پیدا می کند و به این ترتیب سرعت پاسخ به پرس و جوها افزایش می یابد. هر ساختار شاخص، حاوی رکوردهایی است که در هر رکورد یک مقدار کلیدی (کلید جستجو) و آدرس منطقی رکوردهای فایل داده ای نگهداری می شود.

اغلب سیستم های مدیریت پایگاه داده ها، از ساختار درخت برای ایجاد شاخص ها استفاده می کنند. عمق درخت، بیشترین تعداد سطوح از ریشه به برگ است. عمق ممکن است در مسیرهای مختلف از ریشه تا برگ متفاوت باشد. و همچنین عمق ممکن است در مسیرهای مختلف از ریشه تا برگ یکسان باشد، که در این شرایط با درخت متوازن و B -Tree و B^+ -Tree مواجه هستیم. هرچه درجه ی گره های درخت بیشتر شود، درخت پهن تر و کم عمق تری ایجاد می شود. از آنجاییکه زمان دسترسی در یک درخت، بیشتر وابسته به عمق درخت است تا پهنای آن، پس ساخت درخت پهن و کم عمق در ایجاد شاخص باعث افزایش سرعت جستجو می شود، ساختارهای B -Tree و B^+ -Tree درخت هایی با عمق کم و پهنای زیاد هستند. ساختار B^+ Tree index برای پاسخ به Range Query و Equality Query مفید است.

صورت سوال به این شکل است:

در مورد شاخص گذاری (Indexing)، کدام مورد نادرست است؟

(۱) B^+ Tree index برای پاسخ به Range Query مفید است.

گزینه اول درست است، زیرا Clustered Index توسط ساختار B^+ Tree پیاده‌سازی می‌شود. در ساختار B^+ Tree جستجو و بازیابی نقطه‌ای (Equality Query) از مرتبه $O(\log n)$ و جستجو و بازیابی بازه‌ای (Range Query) از مرتبه $O(\log n + k)$ است. بنابراین اعمال سیاست شاخص‌گذاری توسط ساختار B^+ Tree باعث می‌شود Range Query و Equality Query های مرتبط، با سرعت بیشتری انجام شود.

(۲) هر جدول می‌تواند حداکثر یک شاخص کلاستر شده داشته باشد.

گزینه دوم درست است، زیرا هر جدول می‌تواند حداکثر یک clustered Index داشته باشد.

(۳) شاخص‌های از نوع Hash برای پاسخ به Range Query مفید هستند.

گزینه سوم نادرست است، زیرا در ساختار فایل مستقیم (تکنیک درهم‌سازی) جستجو و بازیابی نقطه‌ای (Equality Query) از مرتبه $O(1)$ است. دستیابی از مرتبه $O(1)$ به جدول پایه، به این معناست که بدون توجه، به اندازه جدول پایه، جهت دسترسی به یک رکورد دلخواه، همواره تعداد اندکی پیگرد نیاز است. در حالت ایده‌آل با یک حرکت می‌توان به رکورد مورد نظر دسترسی پیدا کرد. همچنین در ساختار فایل مستقیم (تکنیک درهم‌سازی) جستجو و بازیابی بازه‌ای (Range Query) می‌بایست برای تک تک اعضای بازه به طور مستقل تابع hash اجرا شود که این امر مستلزم صرف وقت و هزینه زیادی است. فایل مستقیم بی‌نظم است و امکان پردازش سریالی و ترتیبی را ندارد، زیرا فایل مستقیم بر حسب کلید مرتب نشده است. به عبارت دیگر به علت درهم بودن فایل، واکنشی رکورد بعدی مشابه واکنشی یک رکورد جدید است، پس امکان پردازش سریالی و ترتیبی در آن وجود ندارد، ساختار فایل مستقیم مناسب محیط‌هایی است که دستیابی سریع به رکوردها مورد نیاز است و پردازش‌ها ترتیبی و سریالی نباشد. بنابراین برای جستجو و بازیابی بازه‌ای (Range Query)، شاخص‌گذاری مرتب شده با ساختار B^+ Tree با مرتبه $O(\log n + k)$ مناسب‌تر است. و شاخص‌های از نوع Hash برای پاسخ به Range Query ها مفید نیست.

(۴) هرچه تعداد شاخص‌های یک جدول بیشتر باشد، سرعت Insert در آن جدول کمتر می‌شود.

گزینه چهارم درست است، زیرا عملیات درج، حذف و بروزرسانی در جدولی که شاخص دارد یعنی خود جدول پایه، نسبت به جدولی که شاخص ندارد زمان بیشتری مصرف می‌کند و به تبع این عملیات کندتر خواهد بود. زیرا شاخص‌ها نیز همگام با جداول پایه خود نیاز به بروزرسانی دارند. بنابراین تنها روی ستون‌هایی شاخص ایجاد می‌گردد، که به تناوب روی آنها جستجو انجام می‌شود. بنابراین هرچه تعداد شاخص‌های یک جدول بیشتر باشد، سرعت Insert، Update و Delete در آن جدول کمتر می‌شود. اما خود شاخص عامل جستجو و بازیابی سریع اطلاعات از یک جدول پایه است.

مشاوره تخصصی رشته کامپیوتر و IT

در راستای رسالت مؤسسه فرهنگی و انتشاراتی بابان مبنی بر ارتقای سطح علم و دانش کشور و کمک همه جانبه به دانشجویان و داوطلبان گرامی، در جهت قبولی در کنکور کارشناسی ارشد و دکتری مهندسی کامپیوتر و IT دو طرح زیر را پایه‌ریزی کرده‌ایم:

- ۱) ارائه مشاوره تخصصی حضوری و غیرحضوری (تلفنی و آنلاین)
 - ۲) برگزاری کلاس‌های حضوری و غیرحضوری (فیلم آموزشی و کلاس آنلاین)
- برای آشنایی بیشتر با خدمات ارائه شده توسط مؤسسه بابان به وب سایت khalilifar.ir یا کانال تلگرام [@arastookhalilifar](https://t.me/arastookhalilifar) مراجعه فرمایید.

تلفن دفتر مرکزی مؤسسه بابان: ۰۲۱-۷۷۹۷۲۸۶۸

تلفن دفتر فروشگاه انتشارات بابان: ۰۲۱-۷۷۹۷۳۳۸۶

پایگاه اطلاع رسانی مؤسسه بابان: www.baban.ir



گروه بابان
BABAN.IR

مشاوره

آزمون

کتاب

کلاس

جهت تهیه فیلم های آموزشی مجموعه کتاب های
راهیان ارشد به سایت موسسه بابان مراجعه نمایید.

⊕ BABAN.IR

کلاس‌های آنلاین و آفلاین ویژه کنکور کارشناسی ارشد ودکتری

با حضور اساتید بابان و مولفین راهیان ارشد



استاد زارع
(مولف کتب راهیان ارشد)



استاد کتیرایی
(مولف کتب راهیان ارشد)



استاد گلیک
(مولف کتب راهیان ارشد)



استاد خلیلی فر
(مولف کتب راهیان ارشد)

در موسسه بابان

رشته‌های: مهندسی کامپیوتر مهندسی IT علوم کامپیوتر

بهترین منابع کنکور ارشد کامپیوتر

منابع	عنوان درس	
کلاس آنلاین یا فیلم استاد زارع یا کتاب راهیان ارشد	زبان انگلیسی	زبان عمومی و تخصصی
کلاس آنلاین یا فیلم استاد گیلک یا جزوه بابان	ریاضی عمومی ۱ و ۲	ریاضیات
کلاس آنلاین یا فیلم استاد گیلک یا جزوه بابان	آمار و احتمال	
کلاس آنلاین یا فیلم استاد گیلک یا کتاب راهیان ارشد	ریاضیات گسسته	
کلاس آنلاین یا فیلم استاد گیلک یا کتاب راهیان ارشد	ساختمان داده‌ها	دروس تخصصی
کلاس آنلاین یا فیلم استاد گیلک یا کتاب راهیان ارشد	طراحی الگوریتم	
کلاس آنلاین یا فیلم استاد خلیلی‌فر یا کتاب راهیان ارشد	سیستم عامل	
کلاس آنلاین یا فیلم استاد خلیلی‌فر یا کتاب راهیان ارشد	پایگاه داده‌ها	
کلاس آنلاین یا فیلم استاد کتیرایی یا کتاب راهیان ارشد	مدارهای منطقی	
کلاس آنلاین یا فیلم استاد کتیرایی یا کتاب راهیان ارشد	معماری کامپیوتر	
کلاس آنلاین یا فیلم استاد گیلک یا کتاب راهیان ارشد	هوش مصنوعی	
کلاس آنلاین یا فیلم استاد خلیلی‌فر یا کتاب راهیان ارشد	شبکه‌های کامپیوتری	
کلاس آنلاین یا فیلم استاد گیلک یا کتاب علوم رایانه استاد شاپوری	نظریه زبان و ماشین	
کلاس آنلاین یا فیلم استاد کتیرایی یا کتاب راهیان ارشد	الکترونیک دیجیتال	
کلاس آنلاین یا فیلم استاد تقدسی یا کتاب نصیر	سیگنال سیستم	

بهترین منابع

کنکور ارشد فناوری اطلاعات

منابع	عنوان درس	
کلاس آنلاین یا فیلم استاد زارع یا کتاب راهیان ارشد	زبان انگلیسی	زبان عمومی و تخصصی
کلاس آنلاین یا فیلم استاد گیلک یا کتاب راهیان ارشد	ریاضیات گسسته	درس مشترک
کلاس آنلاین یا فیلم استاد گیلک یا کتاب راهیان ارشد	ساختمان داده‌ها	
کلاس آنلاین یا فیلم استاد گیلک یا کتاب راهیان ارشد	طراحی الگوریتم	
کلاس آنلاین یا فیلم استاد خلیلی‌فر یا کتاب راهیان ارشد	مهندسی نرم‌افزار	
کلاس آنلاین یا فیلم استاد خلیلی‌فر یا کتاب راهیان ارشد	شبکه‌های کامپیوتری	درس تخصصی
کلاس آنلاین یا فیلم استاد خلیلی‌فر یا کتاب راهیان ارشد	پایگاه داده‌ها	
کلاس آنلاین یا فیلم استاد گیلک یا کتاب راهیان ارشد	هوش مصنوعی	
کلاس آنلاین یا فیلم استاد خلیلی‌فر یا کتاب راهیان ارشد	سیستم عامل	
مباحث مدیریت و مباحث رفقا از سازمانی استیضاف را بریزید زیر کتاب راهیان ارشد.	اصول و مباحث مدیریت	

بهترین منابع کنکور ارشد علوم کامپیوتر

منابع	عنوان درس	
کلاس آنلاین یا فیلم استاد زارع یا کتاب راهیان ارشد	زبان انگلیسی	زبان عمومی و تخصصی
کلاس آنلاین یا فیلم استاد گیلک یا جزوه بابان	ریاضی عمومی ۱ و ۲	دروس پایه
کتاب‌های بن‌ویژین ترجمه عمید، رسولیان فصل‌های اول، پنجم و هفتم کلاس آنلاین ریاضیات گسسته یا فیلم استاد گیلک یا کتاب ریاضیات گسسته راهیان ارشد	مبانی علوم ریاضی	
کتاب جبر خطی هافمن یا کتاب جبر خطی اونان یا کتاب جبر خطی پیام‌نور	مبانی ماتریس‌ها و جبر خطی	
اصول آنالیز ریاضی رودین فصل یک تا شش یا کتاب آنالیز ریاضی یک پیام‌نور	مبانی آنالیز ریاضی	
کتاب راهیان ارشد بهزاد خداکرمی یا کتاب آنالیز عددی بابلیان کلاس آنلاین یا فیلم استاد گیلک یا جزوه بابان کتاب آمار و احتمال شلدون راس	مبانی آنالیز عددی مبانی احتمال	
کلاس آنلاین یا فیلم استاد گیلک یا کتاب راهیان ارشد	ساختمان داده	دروس مشترک
کلاس آنلاین یا فیلم استاد گیلک یا کتاب راهیان ارشد	طراحی الگوریتم	
کلاس آنلاین یا فیلم استاد گیلک یا کتاب علوم رایانه استاد شاپوری	مبانی نظریه محاسبه	
کلاس آنلاین یا فیلم استاد گیلک یا جزوه بابان	مبانی منطق و نظریه مجموعه‌ها	
کلاس آنلاین یا فیلم استاد گیلک یا کتاب راهیان ارشد	ریاضیات گسسته و مبانی ترکیبیات	