

موسسه بابان

انتشارات بابان و انتشارات راهیان ارشد

درس و کنکور ارشد

پایگاه داده‌ها

(حل تشریحی سوالات دولتی ۱۳۹۹)

ویژه‌ی داوطلبان کنکور کارشناسی ارشد مهندسی کامپیوتر و IT

براساس کتب مرجع

راما کریشنان، آبراهام سیلبرشاتز و رامز المصری

ارسطو خلیلی فر

سؤالات کنکور کارشناسی ارشد سال ۱۳۹۹

۱- 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 می کند.
- ۲) اجرا نمی شود، زیرا دو دامنه (Domain) متفاوت باهم مقایسه شده است.
- ۳) تمام تولیدکنندگانی که قطعه های یکسان تولید می کنند را بر می گرداند.
- ۴) تمام قطعات با شماره یکسان را بر می گرداند.

۲- 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

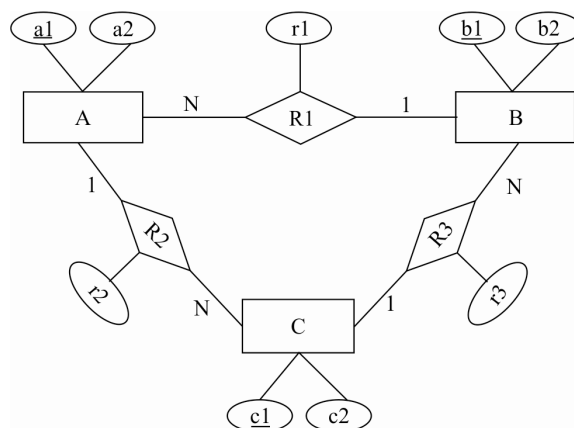
$$\Pi_{City}(\sigma_{avg>16}(STUD)) = \sigma_{avg>16}(\Pi_{City}(STUD)) \quad (۱)$$

$$\sigma_{City}(\Pi_{avg>16}(STUD)) = \Pi_{avg>16}(\sigma_{City}(STUD)) \quad (۲)$$

$$\Pi_{City}(\sigma_{avg>16}(STUD)) \quad (۳)$$

$$\sigma_{avg>16}(\Pi_{City}(STUD)) \quad (۴)$$

۳- مدل رابطه‌ای متناظر با نمودار ER زیر کدام است؟ (مهندسی کامپیوتر- دولتی ۹۹)



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

۴- 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)

- (۱) 1
- (۲) 3
- (۳) 4
- (۴) 7

۵- 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)

- (۱) 8
- (۲) 24
- (۳) 32
- (۴) 40

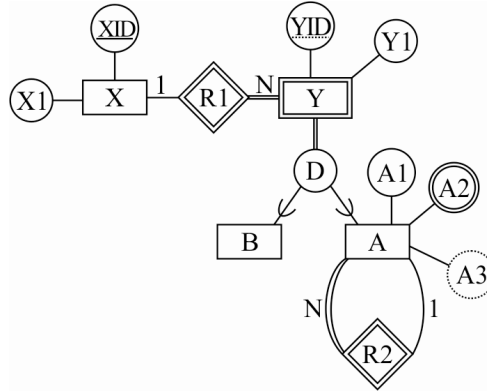
۶- کار عملگر $\rho_{second}(Marks)$ این است که یک کپی از رابطه‌ی Marks با نام مستعار Second ایجاد کند. کار دستور زیر در جبر رابطه‌ای، کدام است؟ (مهندسی کامپیوتر- دولتی ۹۹)

$$\Pi_{mark}(Marks) - \Pi_{Marks.mark}(Marks \times \rho_{second}(Marks))$$

$$Marks.mark > Second.mark$$

- (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.

۷- 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) $AA2(\underline{YID}, A2) \quad A(\underline{YID}, Y1, YIDR2, A1)$

۴) $AA2(\underline{XID}, \underline{YID}, A2) \quad A(\underline{XID}, \underline{YID}, Y1, \underline{XIDR}, \underline{YIDR2}, A1)$

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

- گزاره اول: تعداد کلیدهای کاندید یک رابطه از تعداد سوپرکلیدهای 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

۹- 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 سوم، شماره گره‌هایی را می‌دهد که درجه ورودی آنها کمتر از درجه خروجی آنها است.
- ۴) گزینه‌های ۱ و ۲ صحیح هستند.
- ۱۰- 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 دستورهای SQL (الف، ب و ج) برای این پرس و جو درست است؟ (مهندسی کامپیوتر-دولتی ۹۹)

کتاب‌ها	Book (<u>ISBN</u> , Title, Auther, Publisher)
اعضا	Member (<u>MID</u> , Name, MDate)
امانت کتاب	Borrow (<u>MID</u> , 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

۱۱- فرض کنید $pk(R)$ کلید اصلی رابطه R را نشان می‌دهد. در مورد یک ارتباط چند به یک که بین دو رابطه R_1 و R_2 وجود دارد، کدام گزینه صحیح است؟ (مهندسی IT- دولتی ۹۹)

(۱) $pk(R_2) \rightarrow pk(R_1)$ (۲) $pk(R_1) \rightarrow pk(R_2)$
 (۳) $pk(R_2) \rightarrow R_1 \cap R_2$ (۴) $pk(R_1) \rightarrow R_1 \cap R_2$

۱۲- 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)

(۱) $BD \rightarrow CD$ (۲) $CD \rightarrow AC$ (۳) $BC \rightarrow CD$ (۴) $AC \rightarrow BC$

۱۳- 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.

۱۴- دستور **ON UPDATE CASCADE**، باعث می‌شود در مورد کدام گزینه اطمینان حاصل کنیم؟

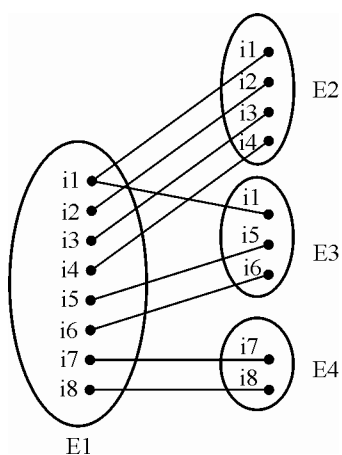
(مهندسی IT- دولتی ۹۹)

(۱) دیدهای ذخیره شده (۲) نرمال بودن داده‌ها (۳) یکپارچگی داده‌ها (۴) تمام موارد بالا

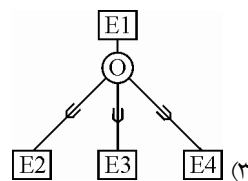
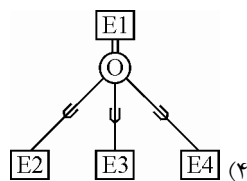
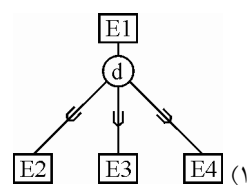
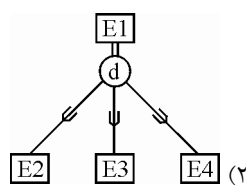
۱۵- رابطه‌های $R1(A,B)$ و $R2(C,D)$ را در نظر بگیرید. کدام گزینه برقرار باشد، خروجی عبارت SQL زیر همان رابطه $R1$ است؟
(مهندسی IT- دولتی ۹۹)

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) هیچ تاپل تکراری ندارد و $R2$ تهی است.
- ۴) هم $R1$ و هم $R2$ هیچ تاپل تکراری ندارند.



۱۶- 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 سوالات کنکور کارشناسی ارشد سال ۱۳۹۹

۱- 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 جوی 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 می‌شود، چون ستون P# داخل جدول Supplier اصلاً تعریف نشده است و نامعتبر است.

مثال: جهت درک بیشتر، جداول و پرس و جوی 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 جوی 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 جوی 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 جوی 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 جوی 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 جوی 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

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.

۲- 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 دانشجوی

$$\Pi_{City}(\sigma_{avg>16}(STUD)) = \sigma_{avg>16}(\Pi_{City}(STUD)) \quad (1)$$

گزینه اول نادرست است، زیرا دو عملگر Π و σ 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:

$$\Pi_{City}(\sigma_{avg>16}(STUD))$$

خروجی پرس و جوی پرانتز 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
C2
C3

همچنین پرس و جوی زیر را در نظر بگیرید:

$$\sigma_{avg>16}(\Pi_{City}(STUD))$$

خروجی پرس و جوی پرانتز 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 پرانتز داخلی توسط عملگر Π انتخاب نشده است، بنابراین اجرای عملگر σ بر روی ستون avg امکان پذیر نیست.

پس رابطه زیر برقرار است:

$$\Pi_{City}(\sigma_{avg>16}(STUD)) \neq \sigma_{avg>16}(\Pi_{City}(STUD))$$

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:

$$\Pi_{City,avg}(\sigma_{avg>16}(STUD))$$

خروجی پرس و جوی پرانتز 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 پرائنز 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:

$$\Pi_{\text{City,avg}}(\sigma_{\text{avg}>16}(\text{STUD})) = \sigma_{\text{avg}>16}(\Pi_{\text{City,avg}}(\text{STUD}))$$

پس همانطور که گفتیم دو عملگر σ و Π به صورت مشروط دارای خاصیت جابه‌جایی هستند.

$$\sigma_{\text{City}}(\Pi_{\text{avg}>16}(\text{STUD})) = \Pi_{\text{avg}>16}(\sigma_{\text{City}}(\text{STUD})) \quad (۲)$$

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 \wedge, \vee, \neg 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 $\Pi_{\text{avg}>16}$ در گزینه دوم نادرست است و خطای نحوی دارد.

$$\Pi_{\text{City}}(\sigma_{\text{avg}>16}(\text{STUD})) \quad (۳)$$

گزینه سوم درست است.

پرس و جوی زیر را در نظر بگیرید:

$$\Pi_{\text{City}}(\sigma_{\text{avg}>16}(\text{STUD}))$$

خروجی پرس و جوی پراتنز 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
C2
C3

(۴) $\sigma_{avg>16}(\Pi_{City}(STUD))$

گزینه چهارم نادرست است.

پرس و جوی زیر را در نظر بگیرید:

$\sigma_{avg>16}(\Pi_{City}(STUD))$

خروجی پرس و جوی پراتنز 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 پراتنز داخلی توسط عملگر Π انتخاب نشده است، بنابراین اجرای عملگر σ بر روی ستون avg امکان پذیر نیست.

۳- گزینه (۱) 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.

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

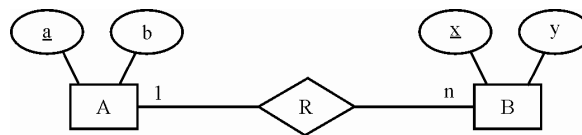
نگاشت رابطه یک به چند بین دو موجودیت به مدل رابطه‌ای

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

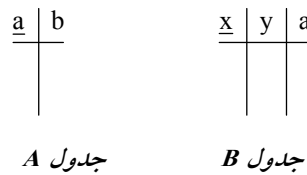
کاندید جدول یک در جدول چند به عنوان کلید خارجی تعریف می‌گردد. همچنین صفات متصل به رابطه، درون جدول چند مستتر می‌شود.

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



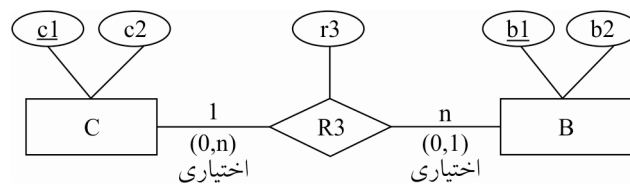
مدل طراحی:



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:

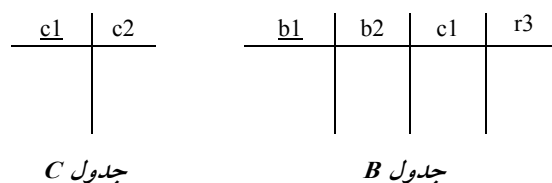


توجه: در شکل فوق صفت c_1 کلید موجودیت C و صفت b_1 کلید موجودیت B است.

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 (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.

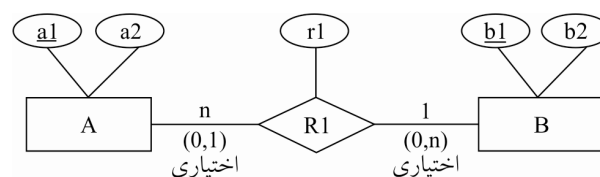
توجه: همانطور که واضح است، طراحی جدول B در گزینه اول به صورت $B(b_1, b_2, c_1, r_3)$ در نظر گرفته شده است، که مطابق آنچه بیان کردیم، طراحی درستی است. بنابراین گزینه‌های دوم، سوم و چهارم را کنار می‌گذاریم، پس تا همینجا پُر واضح است که گزینه اول پاسخ سوال است.

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:

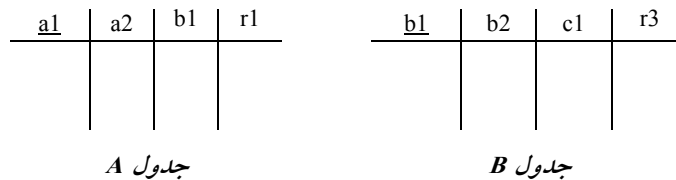


توجه: در شکل فوق صفت a_1 کلید موجودیت A و صفت b_1 کلید موجودیت B است.

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

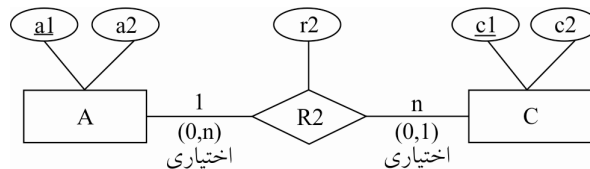
توجه: همانطور که واضح است، طراحی جدول B در گزینه اول به صورت $B(b_1, b_2, c_1, r_3)$ در نظر گرفته شده است، همچنین طراحی جدول A در گزینه اول به صورت $A(a_1, a_2, b_1, r_1)$ در نظر گرفته شده است که مطابق آنچه بیان کردیم، طراحی درستی است. بنابراین گزینه‌های دوم، سوم و چهارم را کنار می‌گذاریم، پس تا همینجا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

توجه: دقت کنید که در گزینه دوم طراحی جدول A به صورت $A(a_1, a_2, b_1, r_1)$ در نظر گرفته شده است که طراحی نادرستی است زیرا کلید جدول A را ترکیب دو ستون (a_1, r_1) 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 نگاشت آن به مدل رابطه‌ای را شرح می‌دهیم.

حالت اختیاری

مدل تحلیل:



توجه: در شکل فوق صفت a_1 کلید موجودیت A و صفت c_1 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:

<u>a1</u>	a2	b1	r1

جدول A

<u>c1</u>	c2	a1	r2

جدول C

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 $C(c1, c2, a1, r2)$ در نظر گرفته شده است، همچنین طراحی جدول A در گزینه اول به صورت $A(a1, a2, b1, r1)$ در نظر گرفته شده است و در نهایت طراحی جدول B در گزینه اول به صورت $B(b1, b2, c1, r3)$ 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 پرواضح است که گزینه اول پاسخ سوال است. به همین سادگی.

توجه: دقت کنید که در گزینه دوم طراحی جدول C به صورت $C(c1, c2, a1, r2)$ در نظر گرفته شده است که طراحی نادرستی است زیرا کلید جدول C را ترکیب دو ستون $(c1, r2)$ در نظر گرفته است که مطابق آنچه بیان کردیم، طراحی نادرستی است.

۴- 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:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [(چپ) x_i - (راست) y_i]$$

توجه: عبارت $[(چپ) x_i - (راست) y_i]$ به طور مستقل بر روی تک تک وابستگی‌ها انجام می‌گردد.

$$\text{مثال: } A \rightarrow BC \Rightarrow BC - A = BC$$

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:

$$AB \rightarrow C$$

$$CD \rightarrow E$$

$$ABCD E - CE = ABD$$

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.

۵- 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) $\{E\}$ و $\{C, D\}$ 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:

$$A \rightarrow BC$$

$$CD \rightarrow E$$

$$B \rightarrow D$$

$$E \rightarrow A$$

$$\cancel{A} \cancel{B} \cancel{C} \cancel{D} E - ABCDE = \text{تهی}$$

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:

$$\{CB\}^+ = \{C, B, D, E, A\}$$

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	ABCF
	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.

۶- گزینه (۲) صحیح است.

مطابق فرض صورت سؤال، کار عملگر psecond(Marks) این است که یک کپی از رابطه‌ی Marks با نام مستعار Second ایجاد کند.

سه جدول Students، Marks و Courses با مقادیر زیر را در نظر بگیرید:

sid	sname	sid	cid	mark	cid	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 Students

جدول Marks

جدول Courses

پرس وجوی 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 سؤال 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 پرس وجوی مطرح شده در هر دو طرف عملگر تفاضل، شروط سازگاری رعایت شده است.

براساس پرس وجوی 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 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 $\text{Marks.mark} > \text{Second.mark}$ ، سطریهایی که دانشجوی جدول Marks نمره اش حداقل از یک نفر دیگر بیشتر باشد را انتخاب می کند (یعنی دانشجویانی که نمره شان 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 می گردد.

<u>Marks.mark</u>
14
16

توجه: در جبر رابطه ای، سطریهای تکراری در خروجی قرار نمی گیرند. در انتها، با انجام عملگر تفاضل، نمره دانشجویان فوق، از مجموعه نمره کلیه دانشجویان کنار گذاشته می شوند. آنچه باقی می ماند، نمره دانشجو یا دانشجویانی است که کمتر از همه نمره گرفته اند. به عبارت دیگر مینیمم نمره دانشجویان استخراج می گردد. بنابراین واضح است که پاسخ گزینه سوم است. یعنی «نمره مینیمم را از لیست نمرات استخراج می کند.»

خروجی نهایی پرس وجوی مطرح شده به صورت زیر است:

$\frac{\text{mark}}{12}$	$-$	$\frac{\text{Marks.mark}}{14}$	$=$	$\frac{\text{mark}}{12}$
14		16		
16				

توجه: اگر شرط موجود در پرس وجوی مطرح شده عکس گردد، آنگاه گزینه سوم ایجاد می گردد.

۷- 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 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.

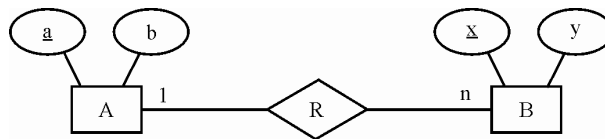
توجه: جهت تبدیل مدل ER (مدل تحلیل) به مدل مدل رابطه‌ای (مدل طراحی یا طراحی منطقی) از سمت و موجودیتی شروع کنید که ورودی صفت از موجودیت دیگری نداشته باشد.

نگاشت رابطه یک به چند بین دو موجودیت به مدل رابطه‌ای

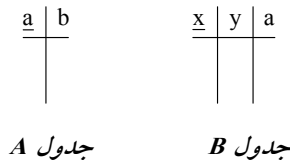
مستقل از اجباری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد. و کلید کاندید جدول یک در جدول چند به عنوان کلید خارجی تعریف می‌گردد.

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



مدل طراحی:



وابستگی وجودی

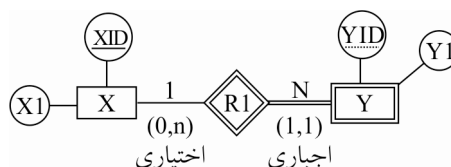
اگر در یک بانک اطلاعاتی، وجود یک موجودیت، وابسته به موجودیت دیگری باشد که در صورت حذف و تغییر موجودیت اصلی یعنی موجودیت قوی این موجودیت نیز تغییر کند، این نوع وابستگی را وابستگی وجودی گفته و به پدیده وابسته، موجودیت ضعیف گویند. همچنین موجودیت ضعیف کلید موجودیت قوی را در بر دارد تا هرگونه تغییر یا حذف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



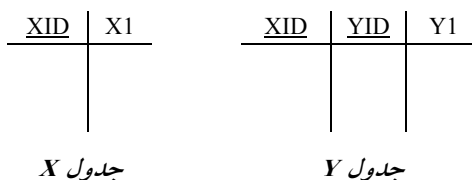
توجه: در شکل فوق صفت XID کلید موجودیت قوی X و صفت YID، صفت ممیزه موجودیت ضعیف Y است. صفت ممیزه موجودیت ضعیف توسط نقطه چین نشان داده می‌شود.

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 رابطه‌ای (مدل طراحی یا طراحی منطقی) از سمت و موجودیتی شروع کنید که ورودی صفت از موجودیت دیگری نداشته باشد.

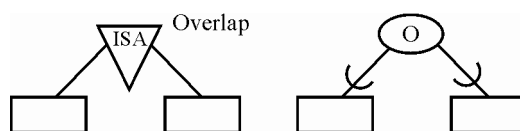
نگاشت رابطه ISA یا وراثت به مدل رابطه‌ای

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.

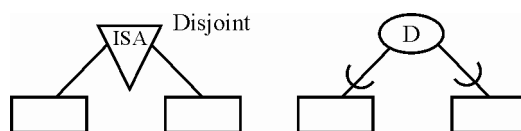
رابطه متصل یا پوشا یا تخصیص غیرمجزا (Overlap) مابین فرزندان و پدر به دو شیوه زیر نشان داده می‌شود:



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 تهي باشد و همچنین می‌تواند غیرتھی 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 تھی 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.

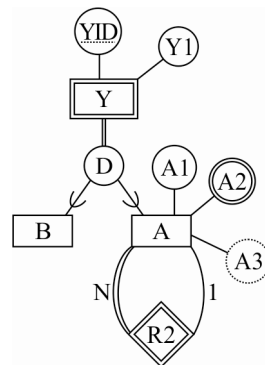
هم‌چنین در یک رابطه منفصل یا غیرپوشا یا تخصیص مجزا (Disjoint)، نمونه موجودیت‌های فرزند نمی‌توانند به طور همزمان با نمونه‌ای از موجودیت پدر در ارتباط باشند. برای مثال در این سؤال، نمونه موجودیت‌های A و B نمی‌توانند به طور همزمان با نمونه‌ای از موجودیت Y در ارتباط باشند. به عبارت دیگر نمی‌توان نمونه‌هایی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

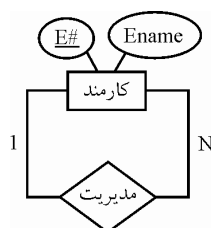
<u>XID</u>	<u>YID</u>	Y1	...	<u>XID</u>	<u>YID</u>	Y1	A1	A2
جدول B				جدول غیرنرمال A				

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

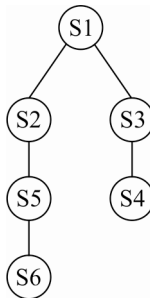
کلید خارجی (شماره مدیر) کلید کاندید (شماره کارمندی)
 ↑
 طوقه

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.

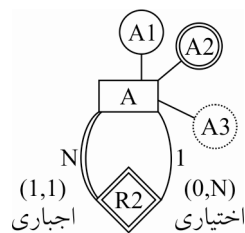
توجه: مدل ریاضی نگاشت رابطه یک به چند در یک موجودیت به مدل رابطه‌ای یک ساختار درختی است، شامل گره ریشه (مدیر کل)، پدر(ناظر) و فرزندان (زیردست).



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:



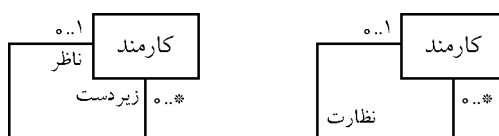
توجه: در شکل فوق و بر اساس نوع نمایش و نشانه‌گذاری‌ها صفت A2 چند مقداری و صفت A3 مشتق (پویا) است. همانطور که قبل‌تر هم گفتیم کلید کاندید موجودیت A صفات (XID,YID) است.

توجه: نماد خط 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.

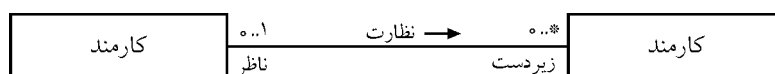
توجه: نماد خط افقی نشانه اختیاری بودن موجودیت چسبیده به آن است، اما نماد دایره کوچک توخالی به معنی صفر و عدم الزام شرکت در رابطه نشانه اختیاری بودن موجودیت طرف مقابل است.

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:

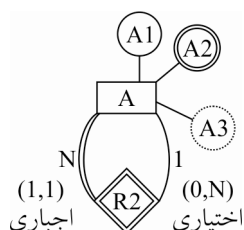


شکل فوق گویای رابطه خودانجمنی (Self Association) است. به این معنی که یک کارمند هیچ زیردستی (فرزند) ندارد یعنی برگ درخت است (چون مدیر نیست) یا چندین زیردست (فرزند) دارد (چون مدیر است) همچنین یک کارمند هیچ ناظری (پدر) ندارد (چون مدیر کل است) یا حداکثر یک ناظر (پدر) دارد (چون مدیر کل نیست).

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، ببیند 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.

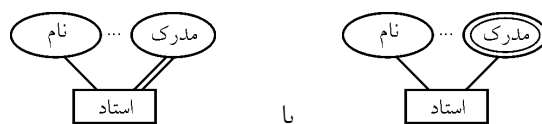
توجه: تعریف دقیق‌تر، قید (0,N) به این معنی است که هر گره (نمونه موجودیت) مورد نظر، حداقل صفر فرزند (زیر دست) و حداکثر N فرزند دارد و قید (1,1) به این معنی است که هر گره (نمونه موجودیت) مورد نظر، حداقل یک و حداکثر یک پدر (ناظر) دارد، در واقع در شکل فوق، ناظر مدیر کل خود مدیر کل است. و سایر گره‌های مورد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



توجه: به مثال‌های زیر توجه کنید.

صفت ساده تک‌مقداری: مانند کد ملی

صفت ساده چندمقداری: مانند مدرک تحصیلی

صفت مرکب تک‌مقداری: مانند تاریخ تولد

صفت مرکب چندمقداری: مانند آدرس

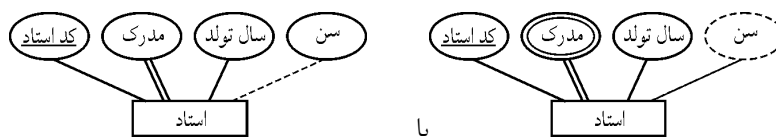
توجه: در بانک اطلاعاتی مبتنی بر مدل رابطه‌ای (جدولی) صفت چندمقداری نداریم.

صفت مشتق (پویا)

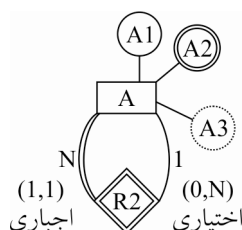
صفتی است که در موجودیت وجود خارجی ندارد ولی در صورت لزوم می‌توان آنرا بدست آورد. صفتی که مقادیر آن مدام در حال تغییر و تحول باشد، صفت پویا یا مشتق محسوب می‌گردد. بنابراین به دلیل تغییرات مداوم، توصیه می‌گردد صفت پویا در جداول بانک اطلاعات مورد استفاده قرار نگیرد و مقدار آن از طریق صفت مرتبط با آن محاسبه 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:



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 غیر نرمال:

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

جدول غیر نرمال 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 دیاگرام 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 خب بهتر بود همین XIDR2 نوشته می‌شد که احتمالاً خطای حروفچینی 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 غیر نرمال 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:

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

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 پرواضح 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.

۸- 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 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 $2^n - 1 = \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n}$ 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 می تواند چندین کلید کاندید داشته باشد.

مثال:

شماره ملی	شماره دانشجویی	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

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، می تواند جزئی از کلید اصلی آن باشد.
 گزاره دوم درست است، زیرا از نظر ساختاری کلید خارجی یک رابطه، می تواند جزئی از کلید اصلی آن 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 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 View 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 کفایست 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 پرس و جوها 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 تجریدی 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 View، Table، 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 View، Table، 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:

<u>S#</u>	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:

<u>S#</u>	Sname
S1	Sn1
S2	Sn2

پایه‌سازی 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 سوم 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 دستورات عملی 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 می‌گردد، که در این حالت پول حداکثر تا ۴۸ ساعت دیگر به حساب شما بازمی‌گردد.

در بیانی دیگر تراکنش را می‌توان اینگونه تعریف کرد، تراکنش مجموعه‌ای از دستورات تعریف و دستکاری داده‌هاست که DBMS تضمین می‌کند یا همه آن دستورات اجرا شوند و یا هیچکدام از آن دستورات اجرا نشوند. برای محقق کردن خاصیت یکپارچگی (Atomicity) هر تراکنش می‌بایست 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 می‌شود. پس تراکنش واحد کار DBMS است. هر تراکنش شامل دو یا چند دستور SQL است. بر این اساس می‌توان گفت که هیچ پرس و جویی برای پایگاه داده هویت مستقل ندارد، بلکه DBMS فقط تراکنش‌ها را می‌شناسد و اجرا می‌کند. به این ترتیب باید گفت که هر پرس و جویی در پایگاه داده ابتدا به یک تراکنش تبدیل می‌شود و سپس اجرا می‌گردد. در SQL برای نمایش ابتدای یک تراکنش از

دستور Begin Transaction و برای نمایش خاتمه یک تراکنش از دستور End Transaction استفاده می‌شود. تراکنش با اجرای Begin Transaction شروع می‌گردد و در صورت اجرای موفق commit و در صورت عدم موفقیت یعنی عدم اجرای همه بخش‌های مختلف تراکنش با اجرای دستور Rollback خاتمه می‌یابد. با اجرای این دستور کلیه تغییراتی که تراکنش روی پایگاه داده اعمال نموده است، ابطال می‌شود و وضعیت پایگاه داده به آخرین وضعیت قبل از اجرای تراکنش برگردانده می‌شود.

۲- سازگاری (Consistency)

به طور کلی جامعیت در سیستم‌های بانکی به دو طبقه جامعیت داخلی و خارجی تقسیم می‌گردد. به حفظ قوانین مطرح شده از سوی مدل رابطه‌ای و DBMS در سطح پیاده‌سازی، جامعیت داخلی و به حفظ قوانین مطرح شده از سوی طراحان و برنامه‌نویسان بانک در سطح پیاده‌سازی، جامعیت خارجی گفته می‌شود. 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 می‌کند. قوانین داخلی بانک شامل قانون جامعیت درون رابطه‌ای، قانون جامعیت موجودیت، قانون جامعیت ارجاعی و قانون جامعیت دامنه‌ای است، این موارد در فصل مدل رابطه‌ای بررسی خواهد شد. قوانین خارجی بانک هم شامل هر قانونی است که طراحان و برنامه‌نویسان بانک آنرا وضع می‌کنند، مانند تعریف زیردامنه برای ورود اطلاعات، تعریف بازه 0 تا 20 برای نمرات، تعریف بازه 0 تا بینهایت برای حساب‌های بانکی به معنی عدم وجود موجودی منفی در حساب‌های بانکی...

در سیستم‌های بانکی کنترل جامعیت داخلی و خارجی به صورت خودکار توسط مکانیزم‌های موجود در DBMS انجام می‌گردد.

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:

Foreign 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:

Foreign 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 500

R=500-100

R 400Write (R) 400

Read (B)

B 500

R=500-50

R 450Write (R) 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.

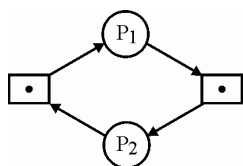
به تعریفی دیگر تراکنش‌ها جدا از یکدیگر هستند. اگر چند تراکنش به طور همزمان اجرا شوند، به هنگام سازی‌های هر کدام از یکدیگر مخفی می‌مانند تا به اتمام برسند. به عبارتی دیگر، برای دو تراکنش مجزای A و B، تراکنش A می‌تواند بهنگام سازی‌های B را پس از پذیرفته شدن آن (commit) یا B می‌تواند بهنگام سازی‌های A را پس از پذیرفته شدن A ببیند. اما این دو تراکنش به طور همزمان نمی‌توانند، بهنگام سازی‌های یکدیگر ببینند.

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

روش قفل‌گذاری

یکی از روش‌های اعمال خاصیت جداسازی در تراکنش‌ها و جلوگیری از اثر مخرب تراکنش‌های همروند بر روی یکدیگر، روش قفل‌گذاری است. در این روش هنگامی که تراکنش به داده‌ای نیاز داشته باشد. تقاضای قفل کردن آن را می‌دهد و در این حالت بقیه تراکنش‌ها تا اتمام کار آن نمی‌توانند از آن استفاده کنند. در واقع استفاده از انواع قفل‌ها در مکانیسم قفل‌گذاری، روش‌های دسترسی به فیلدها و یا داده‌های اشتراکی را در کاربردهای دیگر، امکان‌پذیر می‌سازد. وجود قفل‌ها سبب می‌شود 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 ی 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.

۹- 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.

۱۰- 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 شده 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 گزاره 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:

<p>m1, mn1</p> <p>21 m1 is1 ... 11 is1 ... Springer</p> <p>21 m1 is2 ... 12 is2 ... Springer</p> <hr style="width: 100%;"/> <p style="text-align: center;">گروه اول</p>	<p>m2, mn2</p> <p>22 m2 is1 ... 13 is1 ... Springer</p> <p>22 m2 is2 ... 14 is2 ... Springer</p> <hr style="width: 100%;"/> <p style="text-align: center;">گروه دوم</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------

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 اعضایی 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 اعضایی 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 تمامی کتاب‌های منتشرشده 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.

<u>name</u>
mn1
mn2

راه حل دوم:

پرس و جوی مطرح شده در صورت سوال را در نظر بگیرید:

«نام اعضایی 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

اعضایی 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 منتشر شده 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 اعضایی که، وجود نداشته باشد (NOT EXISTS) کتابی از انتشارات Springer که، به امانت برده نشده باشد (NOT EXISTS). یعنی نام اعضایی 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 هستند که کاری نیست (NOT EXISTS) که، انجام نداده باشند (NOT EXISTS)، یعنی هر کاری بگی انجام دادن. قایقی نیست که سوار نشده باشند، تفریحی نیست که انجام نداده باشند 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 اعضایی 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 اعضایی 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.

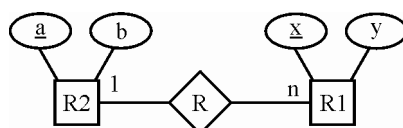
۱۱- گزینه (۲) صحیح است.

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

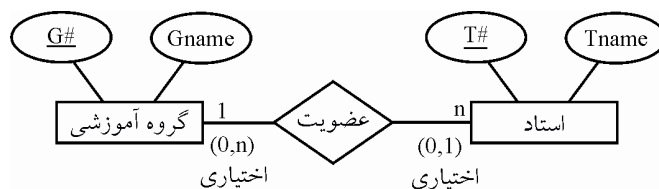
جدول R2

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:

G#	Tname	T#	Tname	G#
G ₁	Gn1	T ₁	Tn1	G1
G ₂	Gn2	T ₂	Tn2	G1
G ₃	Gn3	T ₃	Tn3	G2
		T ₄	Tn4	NULL

جدول استاد جدول گروه آموزشی

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 به عنوان پاسخ اعلام کرده بود.

۱۲- گزینه (۱) صحیح است.

وابستگی‌های مطرح شده به صورت زیر است:

A → B
A → C
CD → E
B → D
E → 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؟}

$$\{CD \rightarrow AC\}^+ = \{CD\}^+ = \{\underline{C}, D, E, \underline{A}, B\}$$

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 می‌دهد.

گزینه سوم پاسخ سوال نیست، زیرا:

{سؤال: آیا می‌توان از صفات BC با استفاده از وابستگی‌های فوق به صفات CD رسید؟}

$$\{BC \rightarrow CD\}^+ = \{BC\}^+ = \{B, \underline{C}, \underline{D}, E, A\}$$

پاسخ: بله می‌توان رسید، زیرا، صفات BC، B و C را می‌دهد. B، D را می‌دهد و CD، E را می‌دهد و E را داریم پس A را هم می‌دهد.

گزینه چهارم پاسخ سوال نیست. زیرا:

{سؤال: آیا می‌توان از صفات AC با استفاده از وابستگی‌های فوق به صفات BC رسید؟}

$$\{AC \rightarrow BC\}^+ = \{AC\}^+ = \{A, \boxed{C}, \boxed{B}, D, E\}$$

پاسخ: بله می‌توان رسید، زیرا AC، A و C را تولید می‌کند. A، B را می‌دهد، B، D را می‌دهد و CD را داریم، پس E را هم می‌دهد.

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

۱۳- 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 $R(A, B, C, D, E, P, G)$ 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.

بنابر صفت AB به صورت زیر است:

$$\{AB\}^+ = \{A, B, C, D, E, P, G\}$$

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.

حال یک‌بار دیگر وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E, P, G)$ را در نظر بگیرید:

وابستگی کامل $AB \rightarrow CD$ غیرکلید

وابستگی انتقالی $DE \rightarrow P$ غیرکلید

وابستگی انتقالی $C \rightarrow E$ غیرکلید

وابستگی انتقالی $P \rightarrow C$ غیرکلید

وابستگی بخشی $B \rightarrow G$ غیرکلید عضو کلید کاندید

در وابستگی‌های 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.

۱۴- 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:

```
Create Table جدول نام
(
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.

۱۵- 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 پرواضح 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

R1 جدول

C	D

R2 جدول

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

R1 جدول

C	D

R2 جدول

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

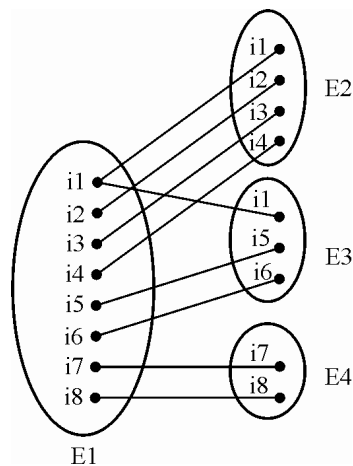
توجه: همانطور که مشاهده می‌شود خروجی عبارت SQL فوق همان رابطه R1 نیست، بنابراین گزینه چهارم پاسخ سوال نیست.

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

۱۶- 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 آنجا که 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.