

موسسه بابان

انتشارات بابان و انتشارات راهیان ارشد

درس و کنکور ارشد

پایگاه داده‌ها

(حل تشریحی سوالات دولتی ۱۳۹۷)

ویژه‌ی داوطلبان کنکور کارشناسی ارشد مهندسی کامپیوتر و IT

براساس کتب مرجع

راما کریشنن، آبراهام سیلبرشاتز و رامز المصری

ارسطو خلیلی فر

تست‌های فصل سوم: مدل رابطه‌ای

- ۱- با توجه به گزاره‌های زیر کدام مورد درست است؟ (مهندسی کامپیوتر-دولتی ۹۷)
- (a) مدل مفهومی پایگاه داده، استقلال داده‌ای را افزایش می‌دهد.
(b) اگر $K1$ و $K2$ ابرکلیدهای رابطه R باشند، آنگاه $K1 \cap K2$ یک ابرکلید R است.
(c) برای اتصال به پایگاه داده، در **Connection String** نحوه احراز هویت کاربر مشخص می‌شود.
(d) در رابطه R با تعداد خصیصه بزرگتر از یک، تعداد ابرکلیدها همواره از تعداد کلیدهای کاندید بیشتر است.
- (۱) a درست، b درست
(۲) a درست، c درست
(۳) d درست، b نادرست
(۴) b نادرست، c نادرست
-

- ۲- با توجه به گزاره‌های زیر، کدام مورد درست است؟ (مهندسی کامپیوتر-دولتی ۹۷)
- (a) اگر رابطه R دارای $n=2k$ خصیصه باشد، آنگاه تعداد کلیدهای کاندید آن حداقل یک و حداکثر $\binom{n}{k}$ است.
(b) اگر رابطه R دارای n خصیصه باشد، آنگاه تعداد ابرکلیدهای این رابطه حداکثر $\binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{n}$ است.
(c) تعداد ابرکلیدهای یک رابطه همواره بیشتر از تعداد کلیدهای کاندید آن رابطه است.
(d) هر صفت مرکب، لزوماً تک مقداری است.
- (۱) یک گزاره نادرست است.
(۲) دو گزاره نادرست است.
(۳) سه گزاره نادرست است.
(۴) چهار گزاره نادرست است.
-

پاسخ تست‌های فصل سوم: مدل رابطه‌ای

۱- گزینه (۲) صحیح است.

صورت سوال به این شکل است:

با توجه به گزاره‌های زیر کدام مورد درست است؟

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

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

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

گزاره b نادرست است، زیرا جدول (nano, stno, avg) stud را در نظر بگیرید. nano (کد ملی) یک کلید کاندید و به تبع یک ابرکلید است. همچنین stno (شماره دانشجویی) یک کلید کاندید و به تبع یک ابرکلید دیگر باشد. اجتماع این دو می‌شود: (nano, stno) که هر چند ابرکلید است ولی یک کلید کاندید نیست (توجه کنید که کلید کاندید باید کمینه باشد و عضو زائد نداشته باشد). همچنین اشتراک این دو برابر مجموعه تهی است که نمی‌تواند نه ابرکلید و نه کلید کاندید باشد. به صورت زیر:

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

(c) برای اتصال به پایگاه داده، در **Connection String** نحوه‌ی احراز هویت کاربر مشخص می‌شود.

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

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

گزاره d نادرست است، زیرا در جدول تمام کلید، یک جدول فقط و فقط یک ابرکلید دارد و فقط و فقط هم یک کلید کاندید دارد. رابطه تمام کلید 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.

۲- گزینه (۳) صحیح است.

صورت سوال به این شکل است:

با توجه به گزاره‌های زیر، کدام مورد درست است؟

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

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

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

مثال: اگر $k=1$ آنگاه $n=2k=2$

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

مثال: اگر $k=2$ آنگاه $n=2k=4$

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

مثال: اگر $k=3$ آنگاه $n=2k=6$

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

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

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

گزاره b نادرست است، زیرا ابرکلید بدون صفت نداریم، بنابراین حالت $\binom{n}{0}$ اضافه است. در حالت کلی یک رابطه دارای n خصیصه، شرایط مختلفی را در تعداد ابرکلید می‌تواند تجربه کند، که حداکثر تعداد ابرکلیدهایی که می‌تواند تجربه کند برابر $\binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n}$ است.

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

گزاره c نادرست است، زیرا در جدول تمام کلید، یک جدول فقط و فقط یک ابرکلید دارد و فقط 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.

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

گزاره d نادرست است، زیرا صفت مرکب صفتی است که از چند صفت ساده تشکیل شده باشد به گونه‌ای که تجزیه‌شدنی باشند و اجزاء حاصله خود صفات ساده باشند. مانند صفت آدرس که از اجزاء نام استان، نام شهر، نام خیابان، نام کوچه، شماره پلاک و کدپستی تشکیل شده است. بعضی از صفات چه ساده و چه مرکب فقط می‌توانند یک مقدار را بگیرند که به این صفات، صفت تک‌مقداری می‌گویند. مانند شماره دانشجویی که نمی‌تواند بیش از یک مقدار داشته باشد. صفاتی وجود دارند که می‌توانند چندین مقدار را بگیرند مانند صفت مدرک در موجودیت استاد که می‌تواند مقادیر لیسانس، فوق لیسانس و یا دکتری را در خود بگیرد. به مثال‌های زیر توجه کنید:

صفت ساده تک‌مقداری: مانند کدملی

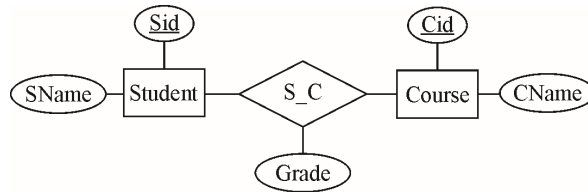
صفت ساده چندمقداری: مانند مدرک تحصیلی

صفت مرکب تک‌مقداری: مانند تاریخ تولد

صفت مرکب چندمقداری: مانند آدرس

تست‌های فصل ششم: SQL دستورات DML

۱- با توجه به نمودار ER داده شده، کدام مورد برای پرس و جو «نام دانشجویانی که معدل آنها از میانگین نمرات همه دروس دانشگاه بیشتر است» نا درست است؟ (مهندسی کامپیوتر-دولتی ۹۷)



```

Select SName
From Student T1, (Select S_C.Sid
                  From S_C
                  group By S_C.Sid
                  having AVG(grade) > (Select AVG(grade)
                                       From S_C)) T2
Where T1.Sid = T2.Sid
    
```

```

Select SName
From Student T1
Where Exists (Select '1'
              From S_C
              Where T1.Sid = S_C.Sid
              group by S_C.Sid
              having AVG(grade) > (Select AVG(grade)
                                    From S_C))
    
```

```

Select SName
From Student
Where Sid in (Select Sid
              From S_C
              group by Sid
              having AVG(grade) > (Select AVG(grade)
                                    From S_C))
    
```

Select SName
 From Student, S_C
 Where Student.Sid = S_C.Sid AND AVG(grade) > (Select AVG(grade)
 From S_C) (۴)

۲- رابطه‌های جدول داده شده را در نظر بگیرید. با اجرای دستور زیر حداقل و حداکثر تعداد سطرهای خروجی کدام است؟
 select * from student left outer join studentCourse

(مهندسی کامپیوتر-دولتی ۹۷)

نام جدول	تعداد سطرها
Student (<u>Stid</u> , StName,...)	$K > 0$
Course (<u>Cid</u> , CName,...)	$N > 0$
StudentCourse (<u>Stid</u> , <u>Cid</u> , grade)	$M > 0$

(۱) حداقل K و حداکثر $K+N-1$
 (۲) حداقل ۱ و حداکثر $K+N-1$
 (۳) حداقل N و حداکثر $K+N$
 (۴) حداقل K و حداکثر $K+N$

۳- مطابق جدول، با اجرای دستور زیر، تعداد سطرهای خروجی کدام است؟ (مهندسی IT-دولتی ۹۷)

select distinct Student.* from student, studentCourse

نام جدول	تعداد سطرها
Student (<u>Stid</u> , StName,...)	K
Course (<u>Cid</u> , CName,...)	N
StudentCourse (<u>Stid</u> , <u>Cid</u> , grade)	M

(۱) K
 (۲) 0
 (۳) $K.N$
 (۴) گزینه‌های ۱ و ۲ می‌توانند صحیح باشند.

۴- با توجه به گزاره‌های داده شده، کدام مورد درست است؟ (مهندسی IT-دولتی ۹۷)

(a) اگر رابطه R ، دارای n خصیصه باشد، آنگاه تعداد ابرکلیدهای آن حداقل یک و حداکثر $2^n - 1$ است.

(b) صفت مرکب، صفتی است که مقدار آن از مقدار سایر صفتها محاسبه می‌شود.

(c) تعداد تاپل‌های عبارت $\Pi_{R,\alpha}(R \times T)$ همواره برابر با تعداد سطرهای $\text{Select distinct } \alpha \text{ From } R$ است.

(d) $\sigma_c(\Pi_\alpha(R)) = \Pi_\alpha(\sigma_c(R))$

(۱) یک گزاره نادرست است.
 (۲) دو گزاره نادرست است.
 (۳) سه گزاره نادرست است.
 (۴) همه گزاره‌ها نادرست هستند.

پاسخ تست‌های فصل ششم: SQL دستورات DML

۱- گزینه (۴) صحیح است.

سه جدول Student، Course و S_C با مقادیر زیر را در نظر بگیرید:

Sid	SName	Sid	Cid	grade	Cid	CName
s1	sn1	s1	c1	10	c1	cn1
s2	sn2	s1	c2	12	c2	cn2
s3	sn3	s2	c1	14	Course	
Student		s2	c2	16		
		s3	c1	18		
		s3	c2	20		
S_C						

مطابق پرس و جوی مطرح شده در گزینه‌ی اول، داریم:

```
Select SName
From Student T1,(Select S_C.Sid
                  From S_C
                  Group By S_C.Sid
                  Having AVG(grade) > (Select AVG(grade)
                                       From S_C)) T2
```

Where T1.Sid=T2.Sid

با توجه به جداول فوق، ابتدا خروجی داخلی‌ترین زیر پرس و جوی داخلی بر اساس جدول S_C به صورت زیر محاسبه می‌گردد:

Sid	Cid	grade
s1	c1	10
s1	c2	12
s2	c1	14
s2	c2	16
s3	c1	18
s3	c2	20

```
Select AVG(grade)
```

```
From S_C
```

که مقدار آن برابر 15 می‌شود، به صورت زیر:

$$\frac{10+12+14+16+18+20}{2} = \frac{90}{6} = 15$$

همچنین در ادامه با توجه به جداول فوق، خروجی زیر پرس و جوی داخلی در جدول S_C به صورت زیر است:

<u>Sid</u>	<u>Cid</u>	grade
s1	c1	10
s1	c2	12
s2	c1	14
s2	c2	16
s3	c1	18
s3	c2	20

همچنین در ادامه، پس از انجام دستور `Group By S_C.Sid` براساس ستون `S_C.Sid` خروجی زیر پرس و جوی داخلی به صورت زیر گروه‌بندی می‌شود:

S1	S2	S3
C1 10	C1 14	C1 18
<u>C2 12</u>	<u>C2 16</u>	<u>C2 20</u>
گروه اول	گروه دوم	گروه سوم

و در نهایت دستور `(Select AVG(grade) > Having AVG(grade) > From S_C)` برای هر گروه به طور

مستقل اعمال می‌گردد.

توجه: دستور `HAVING` بر روی گروه‌ها، اعمال می‌گردد.

S1	S2	S3
C1 10	C1 14	C1 18
<u>C2 12</u>	<u>C2 16</u>	<u>C2 20</u>
گروه اول	گروه دوم	گروه سوم

$$\frac{10+12}{2} = 11 > 15, \quad \frac{14+16}{2} = 15 > 15, \quad \frac{18+20}{2} = 19 > 15$$

توجه: با توجه به شرط انتخاب گروه توسط دستور `(Select AVG(grade) > Having AVG(grade) > From S_C)`

فقط گروه سوم جهت نمایش در خروجی انتخاب می‌شود.

و در نهایت دستور `S_C.Sid` داخل دستور `SELECT` برای هر گروه انتخاب شده توسط دستور `Having` به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد، حاصل نهایی زیر پرس و جوی داخلی یک جدول است که مطابق تعاریف زیر پرس و جوی داخلی `T2` نامگذاری شده است، بنابراین خروجی نهایی زیر پرس و جوی داخلی به صورت زیر است:

Sid
s3

توجه: دستور `GROUP BY`، سرگروه‌ها را، راهی خروجی می‌کند.

بنابراین در ادامه پرس و جوی زیر را خواهیم داشت:

```
Select SName
From Student T1, T2
Where T1.Sid=T2.Sid
```

با توجه به جداول فوق، خروجی نهایی پرس و جوی فوق پس از انجام عملگر ضرب دکارتی و شرط اتصال برابری یعنی سطرهای پیوندپذیر به صورت زیر است:

$$\frac{SName}{sn3}$$

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «نام دانشجویانی که معدل آنها از میانگین نمرات همه دروس دانشگاه بیشتر است.»

مطابق پرس و جوی مطرح شده در گزینه‌ی دوم، داریم:

```
Select SName
From Student T1
Where Exists (Select '1'
              From S_C
              Where T1.Sid=S_C.Sid
              Group by S_C.Sid
              Having AVG(grade) > (Select AVG(grade)
                                   From S_C))
```

با توجه به وجود دستور Exists، زیر پرس و جوی داخلی فوق یک Correlated Subquery است، یعنی به ازای حرکت در هریک از سطرهای پرس و جوی خارجی، یک بار به طور کامل از ابتدا تا انتها زیر پرس و جوی داخلی اجرا و بر اساس شرطی که زیر پرس و جوی داخلی را به پرس و جوی خارجی متصل می‌کند، بررسی انجام می‌شود. مانند دو حلقه تو در تو، که به ازای هربار اجرای حلقه خارجی، یک بار به طور کامل حلقه داخلی اجرا می‌گردد. توسط دستور where در فرم زیر:

```
SELECT SName
FROM Student T1
WHERE EXISTS (...)
```

برای هر سطر از جدول Student شرط جلوی Exists که حاصل یک مقایسه می‌باشد، محاسبه می‌گردد، اگر غیرتهی بود، شرط جلوی where که همان Exists است، TRUE می‌گردد و سطر مورد نظر از جدول Student انتخاب می‌گردد و این رویه برای تک تک سطرهای جدول Student، تا به انتهای جدول Student ادامه پیدا می‌کند. به بیان دیگر این پرس و جو نام دانشجویانی را می‌دهد که پرائنتز مقابل Exists برای آنها غیرتهی است. این پرائنتز هنگامی غیرتهی می‌شود که حاصل مقایسه بیان شده در این پرائنتز غیرتهی شود. حاصل این مقایسه در صورتی غیرتهی می‌شود که معدل دانشجوی مورد نظر از میانگین نمرات همه دروس دانشگاه بیشتر باشد. به عبارت دیگر پرس

و جوی گزینه‌ی دوم نام دانشجویانی که معدل آنها از میانگین نمرات همه دروس دانشگاه بیشتر است را استخراج می‌کند. که مطابق پرس و جوی مطرح شده در صورت سوال است.
توجه: همان‌طور که مشاهده می‌کنید، معدل دانشجو Sn3، از میانگین نمرات همه دروس دانشگاه بیشتر است.

به بیان دیگر عبارت موجود در جلوی دستور exists که به صورت زیر است:

```
Select AVG(grade)
From S_C
```

میانگین نمرات همه دروس دانشگاه را محاسبه می‌کند.

با توجه به جداول فوق داریم، که مقدار آن برابر 15 می‌شود، به صورت زیر:

$$\frac{10+12+14+16+18+20}{6} = \frac{90}{6} = 15$$

همچنین عبارت موجود در جلوی دستور Exists که به صورت زیر است:

```
Select '1'
From S_C
Where T1.Sid=S_C.Sid
Group by S_C.Sid
Having AVG(grade) > (Select AVG(grade)
From S_C)
```

توسط دستور T1.Sid = S_C.Sid به محیط خارج یعنی جدول Student متصل می‌گردد. حال به ازای حرکت در هر سطر از جدول Student، یک بار به طور کامل سطرهای جدول S_C از ابتدا تا انتها با توجه به شرط اتصال بررسی می‌گردد. مطابق شکل زیر:

ابتدا برای سطر اول از جدول Student با توجه به شرط اتصال T1.Sid = S_C.Sid داریم:

S1 S1

<u>Sid</u>	SName	<u>Sid</u>	<u>Cid</u>	grade	<u>Cid</u>	CName
s1	sn1	s1	c1	10	c1	cn1
s2	sn2	s1	c2	12	c2	cn2
s3	sn3	s2	c1	14	Course	
Student		s2	c2	16		
		s3	c1	18		
S_C		s3	c2	20		

همچنین در ادامه، پس از انجام دستور Group By S_C.Sid براساس ستون S_C.Sid خروجی زیر پرس و جوی داخلی به صورت زیر گروه‌بندی می‌شود:

S1
 C1 10
C2 12

و در نهایت دستور $(\text{Select AVG(grade) > Having AVG(grade)})$ اعمال می‌گردد.
 From S_C)

توجه: دستور HAVING بر روی گروه، اعمال می‌گردد.

S1
 C1 10
C2 12

$$\frac{10+12}{2} = 11 > 15$$

توجه: با توجه به شرط انتخاب گروه توسط دستور $(\text{Select AVG(grade) > Having AVG(grade)})$ From S_C)

گروه S1 جهت نمایش در خروجی انتخاب نمی‌شود.

جلوی Exists برابر تهی گردید، بنابراین شرط where در پشت exists، FALSE می‌گردد. بنابراین سطر اول از جدول Student در خروجی نمایش داده نمی‌شود.

حال برای سطر دوم از جدول Student با توجه به شرط اتصال $T1.Sid = S_C.Sid$ داریم:

S2		S2			Course	
<u>Sid</u>	SName	<u>Sid</u>	<u>Cid</u>	grade	<u>Cid</u>	CName
s1	sn1	s1	c1	10	c1	cn1
s2	sn2	s1	c2	12	c2	cn2
s3	sn3	s2	c1	14	Course	
		s2	c2	16		
		s3	c1	18	Course	
		s3	c2	20		

S_C

همچنین در ادامه، پس از انجام دستور Group By S_C.Sid براساس ستون S_C.Sid خروجی زیر پرس و جوی داخلی به صورت زیر گروه‌بندی می‌شود:

S2
 C1 14
C2 16

و در نهایت دستور $(\text{Select AVG(grade) > Having AVG(grade)})$ اعمال می‌گردد.
 From S_C)

توجه: دستور HAVING بر روی گروه، اعمال می‌گردد.

S2
 C1 14
C2 16

$$\frac{14+16}{2} = 15 > 15$$

توجه: با توجه به شرط انتخاب گروه توسط دستور `Having AVG(grade) > (Select AVG(grade) From S_C)`

گروه S2 جهت نمایش در خروجی انتخاب نمی‌شود.

جلوی `Exists` برابر تهی گردید، بنابراین شرط `where` در پشت `exists`، `FALSE` می‌گردد. بنابراین سطر دوم از جدول `Student` در خروجی نمایش داده نمی‌شود.

در ادامه برای سطر سوم از جدول `Student` با توجه به شرط اتصال `T1.Sid = S_C.Sid` داریم:

S3		S3			Course	
<u>Sid</u>	SName	<u>Sid</u>	<u>Cid</u>	grade	<u>Cid</u>	CName
s1	sn1	s1	c1	10	c1	cn1
s2	sn2	s1	c2	12	c2	cn2
s3	sn3	s2	c1	14		
	Student	s2	c2	16		
		s3	c1	18		
		s3	c2	20		
		S_C				

همچنین در ادامه، پس از انجام دستور `Group By S_C.Sid` بر اساس ستون `S_C.Sid` خروجی زیر پرس و جوی داخلی به صورت زیر گروه‌بندی می‌شود:

S3
 C1 18
C2 20

و در نهایت دستور `Having AVG(grade) > (Select AVG(grade) From S_C)` اعمال می‌گردد.

توجه: دستور `HAVING` بر روی گروه، اعمال می‌گردد.

S3
 C1 18
C2 20

$$\frac{18+20}{2} = 19 > 15$$

توجه: با توجه به شرط انتخاب گروه توسط دستور (Select AVG(grade) > Having AVG(grade) > From S_C) گروه S3 جهت نمایش در خروجی انتخاب می‌شود. دستور '1' داخل دستور SELECT برای گروه انتخاب شده توسط دستور Having به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد، بنابراین خروجی نهایی زیر پرس و جوی داخلی به ازای گروه S3 به صورت زیر است:

(no column name)

1

جلوی Exists برابر غیر تهی گردید، بنابراین شرط where در پشت exists، TRUE می‌گردد. بنابراین سطر سوم از جدول Student در خروجی نمایش داده می‌شود. با توجه به جداول فوق، خروجی نهایی پرس و جوی فوق به صورت زیر است:

SName

sn3

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «نام دانشجویانی که معدل آنها از میانگین نمرات همه دروس دانشگاه بیشتر است.»
مطابق پرس و جوی مطرح شده در گزینه‌ی سوم، داریم:

```
Select SName
From Student
Where Sid in (Select Sid
              From S_C
              Group by sid
              Having avg(grade) > (Select AVG(grade)
                                   From S_C))
```

با توجه به وجود دستور in، زیر پرس و جوی داخلی فوق یک Normal Subquery است، یعنی ابتدا زیر پرس و جوی داخلی یک بار و برای همیشه اجرا می‌گردد، سپس پرس و جوی خارجی به ازای حرکت در هر یک از سطرهاى خود، از مقادیر زیر پرس و جوی داخلی استفاده می‌کند. در پرس و جوی فوق به ازای حرکت در هر سطر از جدول Student مجموعه جلوی in بررسی می‌گردد که آیا Sid داخل این مجموعه قرار دارد یا خیر. اگر قرار داشت سطر مورد نظر از جدول Student در خروجی نمایش داده می‌شود.

توسط دستور where در فرم زیر:

```
SELECT SName
FROM Student
WHERE in Sid (...)
```

برای هر سطر از جدول Student کل مجموعه‌ی جلوی in که حاصل یک مقایسه می‌باشد، بررسی

می‌گردد، اگر Sid موجود در هر سطر، داخل Sidهای مجموعه جلوی in بود، آنگاه شرط جلوی where که همان in است، TRUE می‌گردد و سطر مورد نظر از جدول Student انتخاب می‌گردد و این رویه برای تک تک سطرهای جدول Student، تا به انتهای جدول Student ادامه پیدا می‌کند. به بیان دیگر این پرس و جو نام دانشجویانی از جدول Student را می‌دهد که در پراکنش مقابل in قرار دارند. به عبارت دیگر پرس و جوی گزینه‌ی سوم نام دانشجویانی که معدل آنها از میانگین نمرات همه دروس دانشگاه بیشتر است را استخراج می‌کند. که مطابق پرس و جوی مطرح شده در صورت سوال است.

توجه: همان‌طور که مشاهده می‌کنید، معدل دانشجو Sn3، از میانگین نمرات همه دروس دانشگاه بیشتر است.

با توجه به جداول فوق، ابتدا خروجی داخلی‌ترین زیر پرس و جوی داخلی بر اساس جدول S_C به صورت زیر محاسبه می‌گردد:

Sid	Cid	grade
s1	c1	10
s1	c2	12
s2	c1	14
s2	c2	16
s3	c1	18
s3	c2	20

Select AVG(grade)

From S_C

که مقدار آن برابر 15 می‌شود، به صورت زیر:

$$\frac{10+12+14+16+18+20}{2} = \frac{90}{6} = 15$$

همچنین در ادامه با توجه به جداول فوق، خروجی زیر پرس و جوی داخلی در جدول S_C به صورت زیر است:

Sid	Cid	grade
s1	c1	10
s1	c2	12
s2	c1	14
s2	c2	16
s3	c1	18
s3	c2	20

همچنین در ادامه، پس از انجام دستور Group By Sid بر اساس ستون Sid خروجی زیر پرس و جوی داخلی به صورت زیر گروه‌بندی می‌شود:

S1	S2	S3
C1 10	C1 14	C1 18
<u>C2 12</u>	<u>C2 16</u>	<u>C2 20</u>
گروه اول	گروه دوم	گروه سوم

و در نهایت دستور $\text{Having AVG(grade) > (Select AVG(grade) From S_C)}$ برای هر گروه به طور مستقل اعمال می‌گردد.

توجه: دستور HAVING بر روی گروه‌ها، اعمال می‌گردد.

S1	S2	S3
C1 10	C1 14	C1 18
<u>C2 12</u>	<u>C2 16</u>	<u>C2 20</u>
گروه اول	گروه دوم	گروه سوم

$$\frac{10+12}{2} = 11 > 15, \quad \frac{14+16}{2} = 15 > 15, \quad \frac{18+20}{2} = 19 > 15$$

توجه: با توجه به شرط انتخاب گروه توسط دستور $\text{Having AVG(grade) > (Select AVG(grade) From S_C)}$

فقط گروه سوم جهت نمایش در خروجی انتخاب می‌شود.

و در نهایت دستور Sid داخل دستور SELECT برای هر گروه انتخاب شده توسط دستور Having به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد، حاصل نهایی زیر پرس و جوی داخلی یک جدول یا مجموعه است، بنابراین خروجی نهایی زیر پرس و جوی داخلی به صورت زیر است:

$$\frac{\text{Sid}}{\text{s3}}$$

توجه: دستور GROUP BY، سرگروه‌ها را، راهی خروجی می‌کند.

بنابراین در ادامه پرس و جوی زیر را خواهیم داشت:

```
Select SName
From Student
Where Sid in (S3)
```

همانطور که گفتیم برای هر سطر از جدول Student کل مجموعه‌ی جلوی in که حاصل یک مقایسه می‌باشد، بررسی می‌گردد، اگر Sid موجود در هر سطر، داخل Sidهای مجموعه جلوی in بود، آنگاه شرط جلوی where که همان in است، TRUE می‌گردد و سطر مورد نظر از جدول Student انتخاب می‌گردد و این رویه برای تک تک سطرهای جدول Student، تا به انتهای جدول Student ادامه پیدا می‌کند. به بیان دیگر این پرس و جو نام دانشجویانی از جدول Student را

می‌دهد که در پرانتز مقابل in قرار دارند. با توجه به جداول فوق، خروجی نهایی پرس و جوی فوق پس از انجام عملگر in به ازای هر سطر جدول Student به صورت زیر است:

<u>SName</u>
sn3

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «نام دانشجویانی که معدل آنها از میانگین نمرات همه دروس دانشگاه بیشتر است.»

مطابق پرس و جوی مطرح شده در گزینه‌ی چهارم، داریم:

```
Select SName
From Student,S_C
Where Student.Sid = S_C.Sid AND AVG(grade) > (Select AVG(grade)
From S_C)
```

در پرس و جوی گزینه‌ی چهارم تابع عددی AVG اول از سمت چپ در محل نادرست مورد استفاده قرار گرفته است، به طور کلی توابع عددی در جلوی Where باید داخل Select قرار بگیرند. بنابراین پرس و جوی گزینه‌ی چهارم دارای خطای نحوی است و از سوی کامپایلر اجرا نمی‌گردد. فرم اصلاح شده‌ی گزینه‌ی چهارم می‌تواند به فرم‌های گزینه‌های اول، دوم و یا سوم باشد.

۲- گزینه (۱ و ۴) صحیح است.

عملگر الحاق خارجی چپ در جبر رابطه‌ای

این عملگر، مانند الحاق طبیعی، ستون‌های مشترک را فقط یکبار در خروجی قرار می‌دهد. همچنین کلیه سطرهای پیوندپذیر را در خروجی قرار می‌دهد. اما علاوه بر آن کلیه سطرها‌ی پیوندناپذیر جدول سمت چپ را نیز در خروجی قرار می‌دهد و در این حالت برای ستون‌های غیر مشترک جدول سمت راست مقدار NULL قرار می‌دهد.

عملگر الحاق خارجی چپ در SQL

این عملگر، ستون‌های مشترک را دوبار در خروجی قرار می‌دهد. همچنین کلیه سطرهای پیوندپذیر را در خروجی قرار می‌دهد. اما علاوه بر آن کلیه سطرهای پیوندناپذیر جدول سمت چپ را نیز در خروجی قرار می‌دهد و در این حالت برای تمام ستون‌های جدول سمت راست مقدار NULL قرار می‌دهد.

توجه: به تفاوت عملگر الحاق خارجی چپ در جبر رابطه‌ای و SQL دقت کنید.

برای فرم حداقل جداول زیر را در نظر بگیرید:

<u>Stid</u>	StName	...	<u>Stid</u>	<u>Cid</u>	grade	<u>Cid</u>	CName
s1	sn1		s1	c1	10	c1	cn1
s2	sn2		StudentCourse			c2	cn2
s3	sn3		M>0			Course	
Student						N>0	
K>0							

توجه: مطابق فرض سؤال، تعداد سطرهاى جداول باید بیشتر از صفر باشد، یعنی جداول تهی نباشند.

مطابق پرس و جوی مطرح شده در صورت سؤال، داریم:

```
select *
from student left outer join studentCourse
```

که البته پرس و جوی مطرح شده در صورت سؤال کمی خطای نحوی دارد، که فرم اصلاح شده آن به صورت زیر است:

```
select *
from student left outer join studentCourse
on student.stid= studentCourse.Stid
```

توجه: در فرم اصلاح شده‌ی پرس و جوی مطرح شده در صورت سؤال، شرط اتصال و عملگر ON قرار گرفت. پرس و جوی مطرح شده در صورت سؤال یک خطای نحوی داشت که اصلاح شد.

با توجه به جداول فوق، خروجی پرس و جوی مطرح شده در صورت سؤال در حالت حداقل پس از انجام عملگر Left Outer Join به صورت زیر است:

Stid	StName	Stid	Cid	grade
s1	sn1	s1	c1	10
s2	sn2	NULL	NULL	NULL
s3	sn3	NULL	NULL	NULL

بنابراین رابطه زیر برقرار خواهد بود:

$$\text{Cardinality}_{\min}(\text{student left outer join studentCourse}) = K = 3$$

توجه: دقت کنید که در خروجی فوق ستون مشترک stid، دو بار در خروجی قرار گرفت.

برای فرم حداکثر جداول زیر را در نظر بگیرید:

<u>Stid</u>	StName	...	<u>Stid</u>	<u>Cid</u>	grade	<u>Cid</u>	CName
s1	sn1		s1	c1	10	c1	cn1
s2	sn2		s1	c2	20	c2	cn2
s3	sn3		s2	c1	30	Course	
Student			s2	c2	40	N>0	
K>0			s3	c1	50		
			s3	c2	60		
			StudentCourse				
			M>0				

با توجه به جداول فوق، خروجی پرس و جوی مطرح شده در صورت سوال در حالت حداکثر پس از انجام عملگر Left Outer Join به صورت زیر است:

Stid	StName	Stid	Cid	grade
s1	sn1	s1	c1	10
s1	sn1	s1	c2	20
s2	sn2	s2	c1	30
s2	sn2	s2	c2	40
s3	sn3	s3	c1	50
s3	sn3	s3	c2	60

بنابراین رابطه زیر برقرار خواهد بود:

$$\text{Cardinality}_{\max}(\text{student left outer join studentCourse}) = K \times N = 3 \times 2 = 6$$

توجه: سازمان سنجش آموزش کشور، در کلید اولیه خود، ابتدا گزینه اول را به عنوان پاسخ اعلام نمود، سپس در کلید نهایی نظر خود را عوض کرد و گزینه اول و چهارم را به عنوان پاسخ با تأثیر مثبت اعلام کرد، که عمل نادرستی را انجام داده است.

توجه: در کلید نهایی این سؤال به دو دلیل واضح باید حذف می‌شد، دلیل اول وجود خطای کامپایلری در صورت سوال و دلیل دوم عدم وجود گزینه صحیح در گزینه‌های صورت سوال.

۳- گزینه (۴) صحیح است.

CROSS JOIN (ضرب دکارتی)

فرم اول:

```
select *
from Student CROSS JOIN StudentCourse
```

فرم دوم:

```
select *
from Student, StudentCourse
```

توجه: در CROSS JOIN همه سطرهای دو جدول در کنار هم قرار می گیرند.

برای فرم **حداقل**، جداول زیر را در نظر بگیرید:

<u>Stid</u>	StName	...	<u>Stid</u>	<u>Cid</u>	grade	<u>Cid</u>	CName
s1	sn1					c1	cn1
s2	sn2		StudentCourse			c2	cn2
s3	sn3					Course	

Student

توجه: مطابق فرض سؤال، الزامی بر تعداد سطرهای جداول وجود ندارد، یعنی تعداد سطرهای جداول می تواند صفر (تهی) یا بیشتر (غیرتهی) باشد.

خروجی پرس و جوی مطرح شده در صورت سوال در حالت **حداقل** به صورت زیر است:

```
select distinct Student.*
from student, studentCourse
```

<u>Stid</u>	StName	...

بنابراین رابطه زیر برقرار خواهد بود:

$$\text{Cardinality}_{\min}(\text{student}, \text{studentCourse}) = 0$$

برای فرم **حداکثر** جداول زیر را در نظر بگیرید:

<u>Stid</u>	StName	...	<u>Stid</u>	<u>Cid</u>	grade	<u>Cid</u>	CName
s1	sn1		s1	c1	10	c1	cn1
s2	sn2		s1	c2	20	c2	cn2
s3	sn3		s2	b1	30	Course	
	Student		s2	b2	40		
			s3	b1	50		
			s3	B2	60		

StudentCourse

خروجی پرس و جوی مطرح شده در صورت سوال در حالت **حداکثر** به صورت زیر است:

```
select distinct Student.*
from student, studentCourse
```

<u>Stid</u>	StName	...
s1	sn1	
s2	sn2	
s3	sn3	

توجه: در SQL هنگام استفاده از عملگر `distinct`، رکوردهای تکراری فقط یک بار در خروجی ظاهر می‌شوند.

بنابراین رابطه زیر برقرار خواهد بود:

$$\text{Cardinality}_{\max}(\text{student}, \text{studentCourse}) = K = 3$$

۴- گزینه (۳) صحیح است.

صورت سوال به این شکل است:

با توجه به گزاره‌های داده شده، کدام مورد درست است؟

(a) اگر رابطه `R`، دارای `n` خصیصه باشد، آنگاه تعداد ابرکلیدهای آن حداقل یک و حداکثر $2^n - 1$ است.

گزاره اول نادرست است، زیرا در جدول تمام کلید، یک جدول فقط و فقط یک ابرکلید دارد و فقط و فقط هم یک کلید کاندید دارد. رابطه تمام کلید مثلا ممکن است سه ستون داشته باشد، در این حالت یک جدول فقط و فقط یک ابرکلید دارد و فقط و فقط هم یک کلید کاندید دارد. یعنی حداقل یک ابرکلید و یک کلید کاندید دارد و حداکثر هم یک ابرکلید و یک کلید کاندید دارد. اما در حالت عادی به جز حالت یک جدول تمام کلید، اگر رابطه `R`، دارای `n` خصیصه باشد، آنگاه تعداد ابرکلیدهای آن حداقل یک و حداکثر $2^n - 1$ است.

(b) صفت مرکب، صفتی است که مقدار آن از مقدار سایر صفت‌ها محاسبه می‌شود.

گزاره دوم نادرست است، زیرا صفت مشتق (پویا) صفتی است که مقدار آن از مقدار سایر صفت‌ها محاسبه می‌شود. صفت مشتق (پویا) صفتی است که در موجودیت وجود خارجی ندارد ولی در صورت لزوم می‌توان آنرا بدست آورد. صفتی که مقادیر آن مدام در حال تغییر و تحول باشد، صفت پویا یا مشتق محسوب می‌گردد. بنابراین به دلیل تغییرات مداوم، توصیه می‌گردد صفت پویا در جداول بانک اطلاعات مورد استفاده قرار نگیرد و مقدار آن از طریق صفت مرتبط با آن محاسبه گردد. برای مثال برای محاسبه صفت سن، می‌توان صفت تاریخ تولد را در نظر گرفت و از روی این صفت، سن را محاسبه نمود. همچنین صفت ساده صفتی است که مقدار آن از لحاظ معنایی ساده یا اتومیک یا تجزیه نشدنی باشد، به این معنا که اگر مقدار آن را به اجزایی تجزیه کنیم، مقادیر جزئی حاصله فاقد معنا باشد. برای مثال صفت درس و شماره دانشجویی یک صفت ساده است. صفت مرکب صفتی است که از چند صفت ساده تشکیل شده باشد به گونه‌ای که تجزیه شدنی باشند و اجزاء حاصله خود صفات ساده باشند. مانند صفت آدرس که از اجزاء نام استان، نام شهر، نام خیابان، نام کوچه، شماره پلاک و کدپستی تشکیل شده است.

(c) تعداد تاپل‌های عبارت $\Pi_{R,\alpha}(R \times T)$ همواره برابر با تعداد سطرهای `Select distinct α From R` است.

گزاره سوم درست است. جداول `R` و `T` را به صورت زیر در نظر بگیرید:

a	b	c	d	e
1	2	5	6	7
3	4	8	9	10
		11	12	13
		14	15	16

R

T

پرس و جوی زیر را در نظر بگیرید:

 $\Pi_{R,a}(R \times T)$

خروجی پرس و جوی پرائتز داخلی به صورت زیر است:

a	b	c	d	e	=	a	b	c	d	e
1	2	5	6	7		1	2	5	6	7
3	4	8	9	10		1	2	8	9	10
		11	12	13		1	2	11	12	13
		14	15	16		1	2	14	15	16
						3	4	5	6	7
						3	4	8	9	10
						3	4	11	12	13
						3	4	14	15	16

که در نهایت پس از حرکت به سمت خارج و انجام عملگر پرتو، خروجی نهایی پرس و جوی به صورت زیر خواهد بود:

a
1
3

توجه: در جبر رابطه‌ای هنگام استفاده از عملگر Π ، تاپل‌های تکراری فقط یک بار در خروجی ظاهر می‌شوند.

همچنین پرس و جوی زیر را در نظر بگیرید:

Select distinct R.a

From R

خروجی پرس و جوی به صورت زیر است:

a
1
3

توجه: در SQL هنگام استفاده از عملگر distinct، رکوردهای تکراری فقط یک بار در خروجی ظاهر می‌شوند.

بنابراین رابطه زیر برقرار خواهد بود:

$\Pi_{R,a}(R \times T) = \text{Select distinct } \alpha \text{ From } R$

$$\sigma_c(\Pi_\alpha(R)) = \Pi_\alpha(\sigma_c(R)) \quad (d)$$

گزاره چهارم نادرست است، زیرا دو عملگر Π و σ به صورت مشروط دارای خاصیت جابه‌جایی هستند. به طور کلی اگر R یک رابطه، α زیر مجموعه‌ای از ستون‌ها و c مجموعه‌ای از شروط بر روی سطرها باشد، آنگاه تساوی زیر زمانی برقرار است که ستون‌های عملگر σ زیر مجموعه ستون‌های عملگر Π باشد. یعنی $\sigma \subseteq \Pi$:

$$\Pi_\alpha(\sigma_c(R)) = \sigma_c(\Pi_\alpha(R))$$

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

<u>S#</u>	Sname	City
S1	Sn1	C1
S2	Sn2	C2
S3	Sn3	C2

پرس و جوی زیر را در نظر بگیرید:

$$R_2 = \Pi_{S\#,City}(\sigma_{City='C2'}(S))$$

خروجی پرس و جوی پراتنز داخلی به صورت زیر است:

<u>S#</u>	Sname	City
S2	Sn2	C2
S3	Sn3	C2

که در نهایت پس از حرکت به سمت خارج و انجام عملگر پرتو، خروجی نهایی پرس و جوی به صورت زیر خواهد بود:

<u>S#</u>	City
S2	C2
S3	C2

همچنین پرس و جوی زیر را در نظر بگیرید:

$$R_2 = \sigma_{City='C2'}(\Pi_{S\#,City}(S))$$

خروجی پرس و جوی پراتنز داخلی به صورت زیر است:

<u>S#</u>	City
S1	C1
S2	C2
S3	C2

که در نهایت پس از حرکت به سمت خارج و انجام عملگر انتخاب، خروجی نهایی پرس و جوی به صورت زیر خواهد بود:

<u>S#</u>	City
S2	C2
S3	C2

پس رابطه زیر برقرار است:

$$\Pi_{S\#,City}(\sigma_{City='C2'}(S)) = \sigma_{City='C2'}(\Pi_{S\#,City}(S))$$

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

<u>S#</u>	Sname	City
S1	Sn1	C1
S2	Sn2	C2
S3	Sn3	C2

پرس و جوی زیر را در نظر بگیرید:

$$R_2 = \Pi_{S\#}(\sigma_{City='C2'}(S))$$

خروجی پرس و جوی پراتنز داخلی به صورت زیر است:

<u>S#</u>	Sname	City
S2	Sn2	C2
S3	Sn3	C2

که در نهایت پس از حرکت به سمت خارج و انجام عملگر پرتو، خروجی نهایی پرس و جوی به صورت زیر خواهد بود:

<u>S#</u>
S2
S3

همچنین پرس و جوی زیر را در نظر بگیرید:

$$R_2 = \sigma_{City='C2'}(\Pi_{S\#}(S))$$

خروجی پرس و جوی پراتنز داخلی به صورت زیر است:

<u>S#</u>
S1
S2
S3

که در نهایت پس از حرکت به سمت خارج، انجام عملگر انتخاب امکان پذیر نخواهد بود، زیرا ستون‌های عملگر σ یعنی City زیر مجموعه ستون‌های عملگر Π یعنی S# نیست، یعنی $City \not\subseteq S\#$. در واقع ستون City در پراتنز داخلی توسط عملگر Π انتخاب نشده است، بنابراین اجرای عملگر σ بر روی ستون City امکان پذیر نیست.

پس رابطه زیر برقرار است:

$$\Pi_{S\#}(\sigma_{City='C2'}(S)) \neq \sigma_{City='C2'}(\Pi_{S\#}(S))$$

تست‌های فصل هفتم: SQL دستورات DDL و DCL

- ۱- جدول T را در نظر بگیرید که روی ستون a، **Clustered Index** شده است. کدام مورد درست است؟
 (مهندس کامپیوتر-دولتی ۹۷)
- ۱) اعمال سیاست شاخص گذاری، تاثیری بر حجم اطلاعات ذخیره شده بر روی دیسک ندارد.
 ۲) با اعمال سیاست شاخص گذاری، پاسخ به Range Query های مرتبط به a، با سرعت بیشتری انجام می شود.
 ۳) با اعمال سیاست شاخص گذاری، پاسخ به Equality Query های مرتبط به a، با سرعت کمتری انجام می شود.
 ۴) همه موارد درست هستند.

- ۲- پس از اجرای دستورات زیر (به ترتیب مشخص شده) تعداد سطرهای جدول **Person** کدام است؟
 (مهندسی کامپیوتر-دولتی ۹۷)

<pre>Create Table person (personID int, managerID int, level int, Primary key (personID), Foreign key (managerID) References person(personID) on delete CASCADE on update RESTRICT,)</pre>	دستور اول
<pre>insert into person values (14,NULL,1), (12,14,2), (11,12,3), (13,12,3), (15,12,3), (10,14,2), (17,10,3), (18,10,3)</pre>	دستور دوم
<pre>Delete from person where level = 2</pre>	دستور سوم
0 (۴) 1 (۳) 8 (۲) 6 (۱)	

- ۳- فرض کنید اسکیمای جدول های E1، E2 و E3 مطابق با دستورات زیر بوده و تعداد سطرهای هر کدام به ترتیب، M، N و K باشد. پس از اجرای دستور زیر، تعداد سطرهای جدول های E1، E2 و E3 به ترتیب M'، N' و K' است. کدام مورد نمی تواند رخ دهد؟
 (مهندسی IT-دولتی ۹۷)

Delete from E1 where id1=5

create table E1(id1 int, primary key(id1))	دستور اول
create table E2(id2 int, a int, primary key(id2), foreign key(a) references E1(id1) on delete cascade)	دستور دوم
create table E3(id3 int, b int, primary key(id3), foreign key(b) references E2(id2) on delete restrict)	دستور سوم

$$M' = M, N' = N, K' = K \quad (۲) \quad M' = M - 1, N' = N - 1, K' = K - 1 \quad (۱)$$

$$M' < M, N' = N, K' = K \quad (۴) \quad M' < M, N' < N, K' = K \quad (۳)$$

۴- در مورد شاخص گذاری (Indexing)، کدام مورد نادرست است؟ (مهندسی IT-دولتی ۹۷)

(۱) B⁺ Tree index برای پاسخ به Range Query مفید است.

(۲) هر جدول می تواند حداکثر یک شاخص کلاستر شده داشته باشد.

(۳) شاخص های از نوع Hash برای پاسخ به Range Query مفید هستند.

(۴) هرچه تعداد شاخص های یک جدول بیشتر باشد، سرعت Insert در آن جدول کمتر می شود.

پاسخ تست‌های فصل هفتم: SQL دستورات DDL و DCL

۱- گزینه (۲) صحیح است.

دو هدف اصلی سیستم ذخیره و بازیابی اطلاعات در پایگاه داده‌ها، اول سرعت عملیات در ذخیره و بازیابی اطلاعات و دوم صرفه‌جویی در مصرف حافظه است. برای مثال کاهش افزونگی محتوایی (طبیعی) توسط نرمال‌سازی جداول منجر به کاهش میزان حافظه مصرفی می‌شود. عمل واکنشی تک تک رکوردها وقت‌گیر است، برای رفع این عیب شاخص یا Index ابداع شد. برای اینکه بازیابی داده‌ها با سرعت و کارایی بیشتر صورت گیرد، از شاخص استفاده می‌شود. شاخص ساختمان داده‌ای است که سیستم مدیریت پایگاه داده‌ها به کمک آن رکوردهای خاص را در یک فایل با سرعت زیاد پیدا می‌کند و به این ترتیب سرعت پاسخ به پرس و جوها افزایش می‌یابد. هر ساختار شاخص، حاوی رکوردهایی است که در هر رکورد یک مقدار کلیدی (کلید جستجو) و آدرس منطقی رکوردهای فایل داده‌ای نگهداری می‌شود.

اغلب سیستم‌های مدیریت پایگاه داده‌ها، از ساختار درخت برای ایجاد شاخص‌ها استفاده می‌کنند. عمق درخت، بیشترین تعداد سطوح از ریشه به برگ است. عمق ممکن است در مسیرهای مختلف از ریشه تا برگ متفاوت باشد. و همچنین عمق ممکن است در مسیرهای مختلف از ریشه تا برگ یکسان باشد، که در این شرایط با درخت متوازن و B-Tree و B⁺-Tree مواجه هستیم. هرچه درجه‌ی گره‌های درخت بیشتر شود، درخت پهن‌تر و کم‌عمق‌تری ایجاد می‌شود. از آنجاییکه زمان دسترسی در یک درخت، بیشتر وابسته به عمق درخت است تا پهنای آن، پس ساخت درخت پهن و کم‌عمق در ایجاد شاخص باعث افزایش سرعت جستجو می‌شود، ساختارهای B-Tree و B⁺-Tree درخت‌هایی با عمق کم و پهنای زیاد هستند. ساختار B⁺ Tree index برای پاسخ به Range Query و Equality Query مفید است. شاخص‌ها بر روی هارد دیسک ذخیره می‌شوند و حافظه مصرف می‌کنند.

۲- گزینه (۳) صحیح است.

جدول زیر را در نظر بگیرید.

کلید خارجی	کلید کاندید	
	<u>PersonID</u>	ManagerID
		Level

Person

کلید خارجی برای ارتباط میان جداول یا ارتباط درون یک جدول مورد استفاده قرار می‌گیرد. به ازای هر مقدار موجود در یک کلید خارجی، باید دقیقاً یک مقدار متناظر در کلید کاندید متناظر آن وجود داشته باشد، در غیر این صورت می‌گوییم، کلید خارجی دارای ارجاع NULL است. به بیان دیگر، مقادیر کلید خارجی همواره باید زیرمجموعه مقادیر کلید کاندید باشد. یک کلید خارجی در یک رابطه هیچگاه نباید ارجاع NULL داشته باشد، این مسأله را به عنوان یک قانون جامعیتی داخلی در مدل رابطه‌ای، می‌شناسیم و آن را **قانون جامعیت ارجاعی** می‌نامیم. هر مقداری که در کلید خارجی وجود دارد، باید دارای مقدار متناظر در کلید کاندید مقصد باشد ولی عکس آن صادق نیست.

دستور زیر بر روی جدول Person تعریف شده است:

Foreign key (managerID) References person (personID)

on delete CASCADE

یعنی کلید خارجی جدول Person یعنی ستون ManagerID به تغییرات (حذف) کلید کاندید جدول Person یعنی ستون PersonID به فرم cascade حساس باشد و واکنش نشان دهد، دقت کنید که درج در جدول Person باید با رعایت و حفظ قانون جامعیت ارجاعی انجام شود، همچنین حذف در جدول Person به واسطه‌ی تعریف کلید خارجی در جدول Person باعث می‌شود در جهت حفظ قانون جامعیت ارجاعی از جدول Person به Person، جدول Person همواره به فرم cascade به تغییرات (حذف) در جدول Person حساس باشد. اما این حساسیت در جدول Person از جنس cascade است، یعنی اگر رکوردی در جدول Person حذف شود که منجر به حذف رکورد دیگری در جدول Person شود، آن حذف رکورد دیگری نیز در جدول Person نیز در جهت حفظ قانون جامعیت ارجاعی پذیرفته می‌شود.

همچنین دستور زیر بر روی جدول Person تعریف شده است:

Foreign key (managerID) References person(personID)

on update RESTRICT

یعنی کلید خارجی جدول Person یعنی ستون ManagerID به تغییرات (بروزرسانی) کلید کاندید جدول Person یعنی ستون PersonID به فرم restrict(no action) حساس باشد و واکنش نشان دهد، دقت کنید که درج در جدول Person باید با رعایت و حفظ قانون جامعیت ارجاعی انجام شود، همچنین بروزرسانی در جدول Person به واسطه‌ی تعریف کلید خارجی در جدول Person باعث می‌شود در جهت حفظ قانون جامعیت ارجاعی از جدول Person به Person، جدول Person همواره به فرم restrict به تغییرات (بروزرسانی) در جدول Person حساس باشد. اما این حساسیت در جدول Person از جنس restrict است، یعنی اگر رکوردی در جدول Person بروزرسانی شود که منجر به بروزرسانی رکورد دیگری در جدول Person شود، آن بروزرسانی رکورد دیگری در جدول Person در جهت حفظ قانون جامعیت ارجاعی پذیرفته نمی‌شود و رد می‌شود.

فرم جدول و تعداد رکوردهای جدول Person قبل از انجام دستور Delete به صورت زیر است:
Delete from person where level = 2

جدول زیر را در نظر بگیرید.

کلید خارجی کلید کاندید

↓

<u>PersonID</u>	ManagerID	Level
14	NULL	1
12	14	2
11	12	3
13	12	3
15	12	3
10	14	2
17	10	3
18	10	3

Person

Card(Person) = 8

فرم جدول و تعداد رکوردهای جدول Person پس از انجام دستور Delete به صورت زیر است:

کلید خارجی کلید کاندید

↓

<u>PersonID</u>	ManagerID	Level
14	NULL	1

Person

Card(Person) = 1

پس از اجرای دستور Delete اگر سطر (12,14,2) از جدول Person حذف شود، آنگاه به دلیل فرم حساسیت cascade جدول Person به Person، باید سطرهایی که در ستون ManagerID مقدار 12 دارند شامل سطرهای (11,12,3)، (13,12,3) و (15,12,3)، نیز از جدول Person حذف شود و می‌شود. همچنین پس از اجرای دستور Delete اگر سطر (10,14,2) از جدول Person حذف شود، آنگاه به دلیل فرم حساسیت cascade جدول Person به Person، باید سطرهایی که در ستون ManagerID مقدار 10 دارند شامل سطرهای (17,10,3) و (18,10,3) نیز از جدول Person حذف شود و می‌شود و آنچه در انتها می‌ماند به صورت زیر است:

کلید خارجی	کلید کاندید	
PersonID	ManagerID	Level
14	NULL	1

Person

۳- گزینه (۱) صحیح است.

صورت سوال به این شکل است:

فرض کنید اسکیمای جدول‌های E1، E2 و E3 مطابق با دستورات زیر بوده و تعداد سطرهای هر کدام به ترتیب، M، N و K باشد. پس از اجرای دستور زیر، تعداد سطرهای جدول‌های E1، E2 و E3 به ترتیب M'، N' و K' است. کدام مورد نمی‌تواند رخ دهد؟

Delete from E1 where id1=5

$$(1) \quad M' = M - 1, N' = N - 1, K' = K - 1$$

گزینه اول نمی‌تواند رخ دهد. جداول زیر را در نظر بگیرید.

کلید کاندید	کلید خارجی	کلید کاندید	کلید خارجی
<u>id1</u>	a	<u>id2</u>	b
5	5	8	8
3	3	6	
1			2

E1 *E2* *E3*

کلید خارجی برای ارتباط میان جداول مورد استفاده قرار می‌گیرد.

به ازای هر مقدار موجود در یک کلید خارجی، باید دقیقاً یک مقدار متناظر در کلید کاندید متناظر آن وجود داشته باشد، در غیر این صورت می‌گوییم، کلید خارجی دارای ارجاع NULL است. به بیان دیگر، مقادیر کلید خارجی همواره باید زیرمجموعه مقادیر کلید کاندید باشد.

یک کلید خارجی در یک رابطه هیچگاه نباید ارجاع NULL داشته باشد، این مسأله را به عنوان یک قانون جامعیتی داخلی در مدل رابطه‌ای، می‌شناسیم و آن را **قانون جامعیت ارجاعی** می‌نامیم. هر مقداری که در کلید خارجی وجود دارد، باید دارای مقدار متناظر در کلید کاندید مقصد باشد ولی عکس آن صادق نیست.

دستور زیر بر روی جدول E2 تعریف شده است:

foreign key(a) references E1(id1)
on delete cascade

یعنی کلید خارجی جدول E2 یعنی ستون a به تغییرات (حذف) کلید کاندید جدول E1 یعنی

ستون id1 به فرم cascade حساس باشد و واکنش نشان دهد، دقت کنید که درج در جدول E1 به خودش ربط دارد و نیاز به واکنش جدول دیگری نیست، اما حذف در جدول E1 به واسطه‌ی تعریف کلید خارجی در جدول E2 باعث می‌شود در جهت حفظ قانون جامعیت ارجاعی از جدول E2 به E1، جدول E2 همواره به فرم cascade به تغییرات (حذف) در جدول E1 حساس باشد. اما این حساسیت در جدول E2 از جنس cascade است، یعنی اگر رکوردی در جدول E1 حذف شود که منجر به حذف در جدول E2 شود، آن حذف در جدول E2 نیز در جهت حفظ قانون جامعیت ارجاعی پذیرفته می‌شود.

همچنین دستور زیر بر روی جدول E3 تعریف شده است:

foreign key(b) references E2(id2)
on delete restrict

یعنی کلید خارجی جدول E3 یعنی ستون b به تغییرات (حذف) کلید کاندید جدول E2 یعنی ستون id2 به فرم restrict(no action) حساس باشد و واکنش نشان دهد، دقت کنید که درج در جدول E2 به خودش ربط دارد و نیاز به واکنش جدول دیگری نیست، اما حذف در جدول E2 به واسطه‌ی تعریف کلید خارجی در جدول E3 باعث می‌شود در جهت حفظ قانون جامعیت ارجاعی از جدول E3 به E2، جدول E3 همواره به فرم restrict به تغییرات (حذف) در جدول E2 حساس باشد. اما این حساسیت در جدول E3 از جنس restrict است، یعنی اگر رکوردی در جدول E2 حذف شود که منجر به حذف در جدول E3 شود، آن حذف در جدول E2 در جهت حفظ قانون جامعیت ارجاعی پذیرفته نمی‌شود و رد می‌شود.

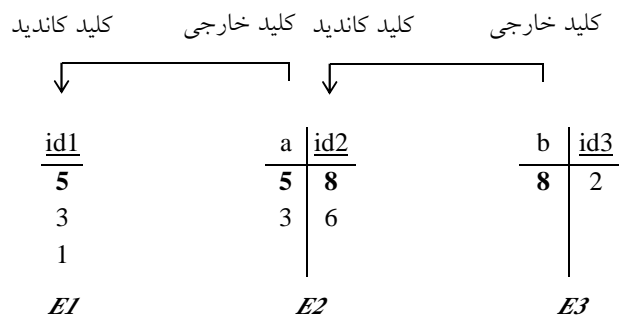
فرم جداول و تعداد رکوردهای جداول E1، E2 و E3 قبل از انجام دستور Delete به صورت زیر است:

Delete from E1 where id1 = 5

↓	↓	↓														
<table border="1" style="border-collapse: collapse; margin: auto;"> <thead> <tr> <th style="padding: 2px 5px;"><u>id1</u></th> </tr> </thead> <tbody> <tr><td style="padding: 2px 5px;">5</td></tr> <tr><td style="padding: 2px 5px;">3</td></tr> <tr><td style="padding: 2px 5px;">1</td></tr> </tbody> </table>	<u>id1</u>	5	3	1	<table border="1" style="border-collapse: collapse; margin: auto;"> <thead> <tr> <th style="padding: 2px 5px;">a</th> <th style="padding: 2px 5px;"><u>id2</u></th> </tr> </thead> <tbody> <tr><td style="padding: 2px 5px;">5</td><td style="padding: 2px 5px;">8</td></tr> <tr><td style="padding: 2px 5px;">3</td><td style="padding: 2px 5px;">6</td></tr> </tbody> </table>	a	<u>id2</u>	5	8	3	6	<table border="1" style="border-collapse: collapse; margin: auto;"> <thead> <tr> <th style="padding: 2px 5px;">b</th> <th style="padding: 2px 5px;"><u>id3</u></th> </tr> </thead> <tbody> <tr><td style="padding: 2px 5px;">8</td><td style="padding: 2px 5px;">2</td></tr> </tbody> </table>	b	<u>id3</u>	8	2
<u>id1</u>																
5																
3																
1																
a	<u>id2</u>															
5	8															
3	6															
b	<u>id3</u>															
8	2															
<i>E1</i>	<i>E2</i>	<i>E3</i>														

$$\text{Card}(E1) = M = 3, \quad \text{Card}(E2) = N = 2, \quad \text{Card}(E3) = K = 1$$

فرم جداول و تعداد رکوردهای جداول E1، E2 و E3 پس از انجام دستور Delete به صورت زیر است:

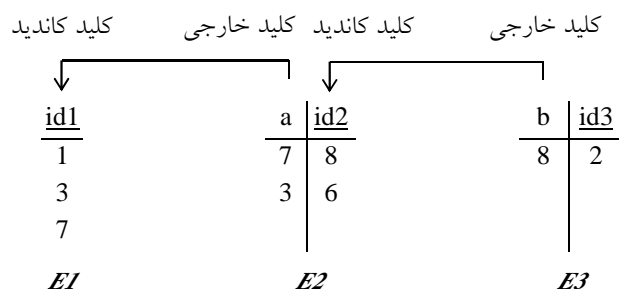


$$\text{Card}(E1) = M' = M = 3, \text{Card}(E2) = N' = N = 2, \text{Card}(E3) = K' = K = 1$$

پس از اجرای دستور Delete هیچ رکوردی حذف نمی‌شود و تغییری در تعداد رکوردهای هیچ یک از جداول ایجاد نمی‌شود. در حالی که قرار بود سطر اول همه‌ی جداول حذف شود تا یک سطر از همه‌ی جداول کم شود. اگر سطر (5) از جدول E1 حذف شود، آنگاه به دلیل فرم حساسیت cascade جدول E2 به E1، باید سطر (5,8) نیز از جدول E2 حذف شود. همچنین اگر سطر (5,8) از جدول E2 حذف شود، آنگاه به دلیل فرم حساسیت restrict جدول E3 به E2، سطر (8,2) از جدول E3 حذف نمی‌شود. و در جهت حفظ قانون جامعیت ارجاعی به شکل بازگشتی، سطر (5,8) هم از جدول E2 حذف نمی‌شود، همچنین در جهت حفظ قانون جامعیت ارجاعی سطر (5) هم از جدول E1 حذف نمی‌شود.

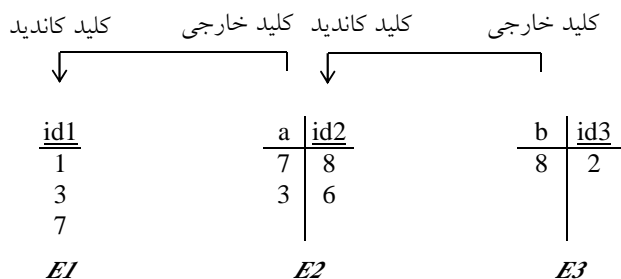
$$M' = M, N' = N, K' = K \quad (۲)$$

گزینه دوم می‌تواند رخ دهد. جداول زیر را در نظر بگیرید.



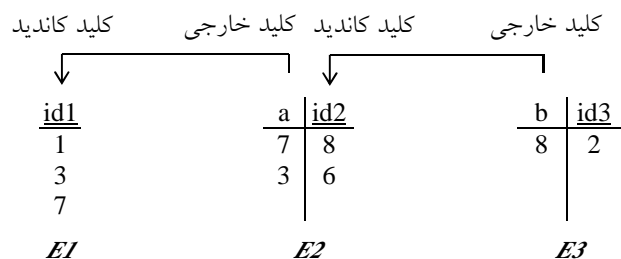
فرم جداول و تعداد رکوردهای جداول E1، E2 و E3 قبل از انجام دستور Delete به صورت زیر است:

Delete from E1 where id1=5



$$\text{Card}(E1) = M = 3, \quad \text{Card}(E2) = N = 2, \quad \text{Card}(E3) = K = 1$$

فرم جداول و تعداد رکوردهای جداول E1، E2 و E3 پس از انجام دستور Delete به صورت زیر است:

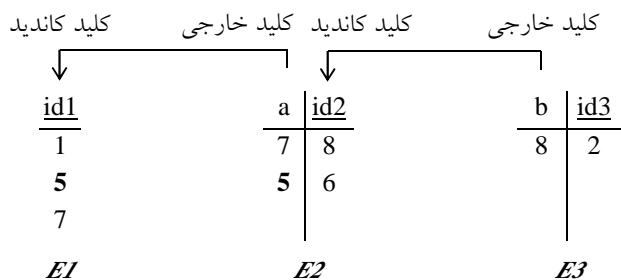


$$\text{Card}(E1) = M' = M = 3, \quad \text{Card}(E2) = N' = N = 2, \quad \text{Card}(E3) = K' = K = 1$$

پس از اجرای دستور Delete هیچ رکوردی حذف نمی‌شود و تغییری در تعداد رکوردهای هیچ یک از جداول ایجاد نمی‌شود. چون جدول E1 اصلاً سطری با مقدار (5) ندارد که بخواهد حذف شود.

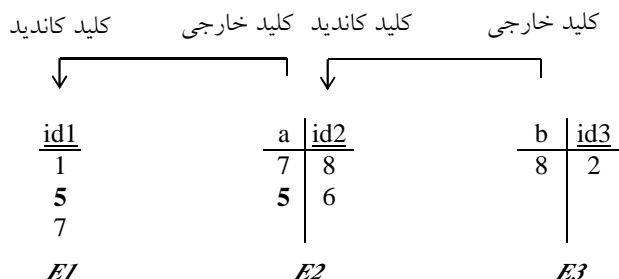
$$M' < M, N' < N, K' = K \quad (۳)$$

گزینه سوم می‌تواند رخ دهد. جداول زیر را در نظر بگیرید.



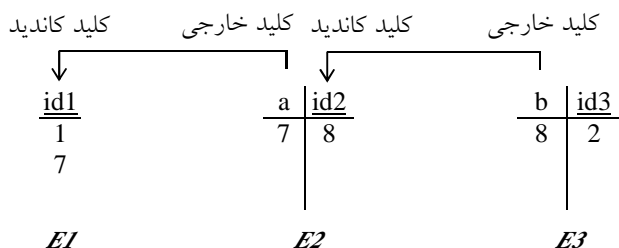
فرم جداول و تعداد رکوردهای جداول E1، E2 و E3 قبل از انجام دستور Delete به صورت زیر است:

Delete from E1 where id1=5



$$\text{Card}(E1) = M = 3, \quad \text{Card}(E2) = N = 2, \quad \text{Card}(E3) = K = 1$$

فرم جداول و تعداد رکوردهای جداول E1، E2 و E3 پس از انجام دستور Delete به صورت زیر است:



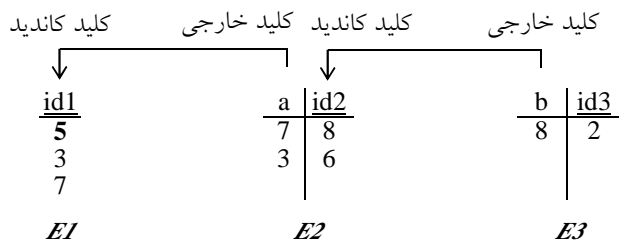
$$\text{Card}(E1) = M' = 2, \quad \text{Card}(E2) = N' = 1, \quad \text{Card}(E3) = K' = 1$$

پس از اجرای دستور Delete اگر سطر (5) از جدول E1 حذف شود، آنگاه به دلیل حساسیت cascade جدول E2 به E1، باید سطر (5,6) نیز از جدول E2 حذف شود و می‌شود. همچنین اگر سطر (5,6) از جدول E2 حذف شود، این حذف، حساسیتی بر روی جدول E3 ایجاد نمی‌کند.

$$\text{Card}(E1) = M' < M, \quad \text{Card}(E2) = N' < N, \quad \text{Card}(E3) = K' = K$$

$$M' < M, N' = N, K' = K \quad (۴)$$

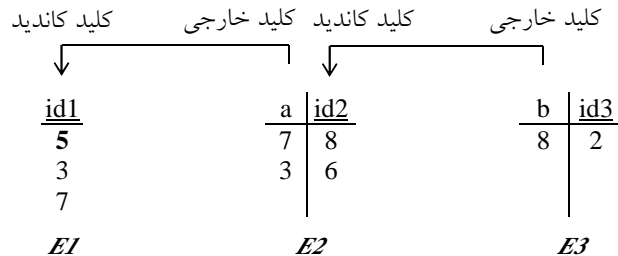
گزینه چهارم می‌تواند رخ دهد. جداول زیر را در نظر بگیرید.



فرم جداول و تعداد رکوردهای جداول E1، E2 و E3 قبل از انجام دستور Delete به صورت زیر

است:

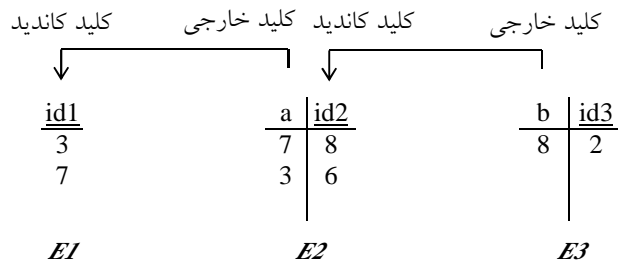
Delete from E1 where id1=5



$$\text{Card}(E1) = M = 3, \quad \text{Card}(E2) = N = 2, \quad \text{Card}(E3) = K = 1$$

فرم جداول و تعداد رکوردهای جداول E1، E2 و E3 پس از انجام دستور Delete به صورت زیر

است:



$$\text{Card}(E1) = M' = 2, \quad \text{Card}(E2) = N' = 2, \quad \text{Card}(E3) = K' = 1$$

پس از اجرای دستور Delete اگر سطر (5) از جدول E1 حذف شود، آنگاه این حذف، حساسیتی بر روی جدول E2 ایجاد نمی‌کند.

$$\text{Card}(E1) = M' < M, \quad \text{Card}(E2) = N' = N, \quad \text{Card}(E3) = K' = K$$

۴- گزینه (۳) صحیح است.

دو هدف اصلی سیستم ذخیره و بازیابی اطلاعات در پایگاه داده‌ها، اول سرعت عملیات در ذخیره و بازیابی اطلاعات و دوم صرفه‌جویی در مصرف حافظه است. برای مثال کاهش افزونگی محتوایی (طبیعی) توسط نرمال‌سازی جداول منجر به کاهش میزان حافظه مصرفی می‌شود. عمل واکنشی تک تک رکوردها وقت‌گیر است، برای رفع این عیب شاخص یا Index ابداع شد. برای اینکه بازیابی داده‌ها با سرعت و کارایی بیشتر صورت گیرد، از شاخص استفاده می‌شود. ساختار داده‌ای است که سیستم مدیریت پایگاه داده‌ها به کمک آن رکوردهای خاص را در یک فایل با سرعت زیاد پیدا می‌کند و به این ترتیب سرعت پاسخ به پرس و جوها افزایش می‌یابد. هر ساختار شاخص، حاوی رکوردهایی است که در هر رکورد یک مقدار کلیدی (کلید جستجو) و آدرس منطقی رکوردهای فایل داده‌ای نگهداری می‌شود.

اغلب سیستم‌های مدیریت پایگاه داده‌ها، از ساختار درخت برای ایجاد شاخص‌ها استفاده می‌کنند. عمق درخت، بیشترین تعداد سطوح از ریشه به برگ است. عمق ممکن است در مسیرهای مختلف از ریشه تا برگ متفاوت باشد. و همچنین عمق ممکن است در مسیرهای مختلف از ریشه تا برگ یکسان باشد، که در این شرایط با درخت متوازن و B-Tree و B⁺-Tree مواجه هستیم. هرچه درجه‌ی گره‌های درخت بیشتر شود، درخت پهن‌تر و کم‌عمق‌تری ایجاد می‌شود. از آنجاییکه زمان دسترسی در یک درخت، بیشتر وابسته به عمق درخت است تا پهنای آن، پس ساخت درخت پهن و کم‌عمق در ایجاد شاخص باعث افزایش سرعت جستجو می‌شود، ساختارهای B-Tree و B⁺-Tree درخت‌هایی با عمق کم و پهنای زیاد هستند. ساختار B⁺ Tree index برای پاسخ به Equality Query و Range Query مفید است. شاخص‌ها بر روی هارد دیسک ذخیره می‌شوند و حافظه مصرف می‌کنند.

تست‌های فصل هشتم: وابستگی تابعی

۱- با توجه به رابطه $R(A,B,C,D,E,F)$ و مجموعه وابستگی‌های تابعی زیر، رابطه R چند کلید کاندید دارد؟
(مهندسی کامپیوتر-دولتی ۹۷)

$$F = \{A \rightarrow BCD, BC \rightarrow DE, B \rightarrow D, D \rightarrow A\}$$

- ۱) 4 ۲) 3 ۳) 2 ۴) 1
-

۲- با توجه به رابطه $R(A,B,C,D,E)$ و مجموعه وابستگی‌های تابعی زیر، کدام مورد نادرست است؟
(مهندسی IT-دولتی ۹۷)

$$F = \{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$$

- 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 تست‌های فصل هشتم: وابستگی تابعی

۱- 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 $R(A, B, C, D, E, F)$ داریم:

$$A \rightarrow BCD$$

$$BC \rightarrow DE$$

$$B \rightarrow D$$

$$D \rightarrow A$$

$$ABCDEF - ABCDE = F$$

بنابر رابطه فوق صفت F حتماً باید عضو کلید کاندید باشد. بستار صفات F به صورت زیر است:

$$\{F\}^+ = \{F\}$$

بر اساس بستار فوق، صفت F فقط ستون F را تولید می‌کند، پس صفت F فقط عضو کلید کاندید می‌باشد و کلید کاندید نمی‌باشد.

قانون سوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، برخی از ستون‌ها را تولید کند، بدین معنی است که، جدول موردنظر، چندین کلید کاندید دارد، که این عضو کلید

کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد. همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت.

با کمی دقت و بررسی، پُر واضح است که اگر صفت A در کنار عضو کلید کاندید F قرار بگیرد، همکاری صفات (A و F) می‌تواند، همه ستون‌ها را تولید کند، بنابراین صفات AF کلید کاندید جدول R خواهد بود.

بستار صفات AF به صورت زیر است:

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

براساس بستار فوق، صفات AF، همه ستون‌ها را تولید می‌کنند، پس صفات AF، کلید کاندید است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد.

همچنین از آن‌جا که $D \rightarrow A$ ، پس می‌توان ترکیب دو خصیصه (D, F) را هم کلید کاندید دیگری برای این جدول تلقی کرد.

چون وقتی (A, F) کلید کاندید است و همه ستون‌ها را تولید می‌کند، پس (D, F) هم کلید کاندید است و همه ستون‌ها را تولید می‌کند، زیرا در نهایت طبق وابستگی $D \rightarrow A$ ، صفت D، صفت A را می‌دهد و (D, F) به (A, F) تبدیل می‌گردد.

بستار صفات (D, F) به صورت زیر است:

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

براساس بستار فوق، صفات (D, F)، همه ستون‌ها را تولید می‌کنند، پس صفات (D, F) کلید کاندید است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد.

همچنین از آن‌جا که $B \rightarrow D$ ، پس می‌توان ترکیب دو خصیصه (B, F) را هم کلید کاندید دیگری برای این جدول تلقی کرد.

چون وقتی (D, F) کلید کاندید است و همه ستون‌ها را تولید می‌کند، پس (B, F) هم کلید کاندید است و همه ستون‌ها را تولید می‌کند، زیرا در نهایت طبق وابستگی $B \rightarrow D$ ، صفت B، صفت D را می‌دهد و (B, F) به (D, F) تبدیل می‌گردد.

بستار صفات (B, F) به صورت زیر است:

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

براساس بستار فوق، صفات (B, F)، همه ستون‌ها را تولید می‌کنند، پس صفات (B, F) کلید کاندید است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده، در مجموع، دارای سه کلید کاندید است که مطابق قانون سوم ارسطو، عضو کلید کاندید F در بین هر سه کلید کاندید به طور مشترک قرار دارد.

۲- گزینه (۲) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

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

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

$$A \rightarrow BC \Rightarrow BC - A = BC$$

مثال:

توجه: استفاده از روش اول مستلزم گام ابتدایی حذف وابستگی‌های بدیهی است، اگر در حذف وابستگی‌های بدیهی دچار خطا می‌شوید، از روش دوم استفاده نمایید.

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E)$ داریم:

$$A \rightarrow BC$$

$$CD \rightarrow E$$

$$B \rightarrow D$$

$$E \rightarrow A$$

$$ABCDE - ABCDE = \text{تهی}$$

قانون چهارم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، تهی گردد، بدین معنی است که، جدول فوق چندین کلید کاندید دارد، که هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید آن وجود ندارد. بنابراین باید کلید کاندید با بررسی دقیق بر روی مجموعه وابستگی کشف گردد.

$$\{B\}^+ = \{B, D\}$$

بر اساس بستار فوق، صفت B، فقط ستون‌های B و D را تولید می‌کند، پس صفت B، کلید کاندید نمی‌باشد. بنابراین گزینه دوم پاسخ سوال خواهد بود.

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

بر اساس بستار فوق، صفت A، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفت A، کلید کاندید می‌باشد.

در ادامه به شکل بازگشتی جهت کشف مابقی کلیدهای کاندید داریم:

صفت E، ستون A را تولید می‌کند. پس صفت E کلید کاندید است، به صورت زیر:

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

بر اساس 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:

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

بر اساس بستار فوق، صفات CD، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفات CD، کلید کاندید می‌باشد.

صفت B، ستون D را تولید می‌کند. پس صفات CB کلید کاندید است، به صورت زیر:

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

بر اساس بستار فوق، صفات CB، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفات CB، کلید کاندید می‌باشد.

توجه: همان‌طور که مشاهده می‌شود، مطابق قانون چهارم ارسطو، هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید فوق وجود ندارد.

تست‌های فصل نهم: نرمال‌سازی

- ۱- اگر در رابطه R، تعداد ابرکلیدها با تعداد کلیدهای کاندید برابر باشد، آنگاه کدام مورد در خصوص رابطه R نادرست است؟
- (۱) تنها یک خصیصه دارد.
- (۲) در فرم نرمال 3NF است.
- (۳) همه خصیصه‌های R، NOT NULL هستند.
- (۴) تعداد وابستگی‌های تابعی نابدیهی R، صفر هستند.
-

پاسخ تست‌های فصل نهم: نرمال‌سازی

۱- گزینه (۱) صحیح است.

صورت سوال به این شکل است:

اگر در رابطه R، تعداد ابرکلیدها با تعداد کلیدهای کاندید برابر باشد، آنگاه کدام مورد در خصوص رابطه R نادرست است؟

(۱) تنها یک خصیصه دارد.

گزینه اول نادرست است، زیرا در جدول تمام کلید، یک جدول فقط و فقط یک ابرکلید دارد و فقط و فقط هم یک کلید کاندید دارد. رابطه تمام کلید مثلا ممکن است سه ستون داشته باشد، در این حالت یک جدول فقط و فقط یک ابرکلید دارد و فقط هم یک کلید کاندید دارد. یعنی حداقل یک ابرکلید و یک کلید کاندید دارد و حداکثر هم یک ابرکلید و یک کلید کاندید دارد. همچنین جدولی هم که تنها یک خصیصه دارد حداقل یک ابرکلید و یک کلید کاندید دارد و حداکثر هم یک ابرکلید و یک کلید کاندید دارد، اما این حالت فقط برای یک جدول تک خصیصه‌ای نیست و همانطور که گفتیم در جدول تمام کلید تعداد ابرکلید و کلید کاندید یکسان، و برابر مقدار یک است. هر ترکیبی از صفات (ستون‌ها) که خاصیت کلیدی داشته باشد، یک ابرکلید است. و ابرکلیدی که عضو زائد نداشته باشد، کلید کاندید است.

(۲) در فرم نرمال 3NF است.

گزینه دوم درست است، زیرا در حالتی که تعداد ابرکلیدها با تعداد کلیدهای کاندید برابر باشد، آن جدول یا تک خصیصه‌ای است و یا تمام کلید است که در هر دو حالت در فرم نرمال 3NF قرار دارد.

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

- جدول باید در نرمال فرم دوم باشد.
- جدول باید فاقد وابستگی انتقالی باشد.

وابستگی انتقالی: وابستگی یک مؤلفه غیرکلیدی به یک مؤلفه غیرکلیدی دیگر را وابستگی انتقالی می‌نامند.

در حالت جدول تک خصیصه‌ای یا جدول تمام کلید، وابستگی بخشی وجود ندارد، پس نرمال فرم دوم برقرار است. همچنین در حالت جدول تک خصیصه‌ای یا جدول تمام کلید، وابستگی انتقالی هم وجود ندارد، پس نرمال فرم سوم هم برقرار است.

(۳) همه خصیصه‌های R، NOT NULL هستند.

گزینه سوم درست است، زیرا در حالتی که تعداد ابرکلیدها با تعداد کلیدهای کاندید برابر باشد، آن جدول یا تک خصیصه‌ای است و یا تمام کلید است که در هر دو حالت همه خصیصه‌های جدول

R باید NOT NULL باشد. مطابق تعریف قانون جامعیت موجودیت، هیچگاه نباید تمام یا بخشی از کلید کاندید مقدار NULL داشته باشد.

۴) تعداد وابستگی‌های تابعی نابدیهی R، صفر هستند.

گزینه چهارم درست است، زیرا در حالتی که تعداد ابرکلیدها با تعداد کلیدهای کاندید برابر باشد، آن جدول یا تک خصیصه‌ای است و یا تمام کلید است که در هر دو حالت تعداد وابستگی‌های تابعی نابدیهی R، صفر هستند. در جدول تک خصیصه‌ای یا تمام کلید، هیچ وابستگی غیربدیهی (نابدیهی) همچون وابستگی تابعی بخشی، وابستگی تابعی انتقالی و وابستگی تابعی معکوس وجود ندارد. یعنی تعداد وابستگی‌های تابعی نابدیهی R، صفر هستند.
