

موسسه بابان

انتشارات بابان و انتشارات راهیان ارشد

درس و کنکور ارشد

پایگاه داده‌ها

(حل تشریحی سوالات کنکور ارشد ۷۰ تا ۹۹)

ویژه‌ی داوطلبان کنکور کارشناسی ارشد مهندسی کامپیوتر و IT

براساس کتب مرجع

rama Krishnan, Abraham Silberschatz و Ramz Almossry

ارسطو خلیلی‌فر

کلیه‌ی حقوق مادی و معنوی این اثر در سازمان استناد و کتابخانه‌ی ملی ایران به ثبت رسیده است.

تقدیم به:

تمامی آنلای که برای پیشرفت و سعادت خود و بشریت
تلash می کنند.

ارسطو خلیلی فر

فهرست مطالب

۹.....	تست‌های فصل اول: مفاهیم اولیه
۱۵.....	پاسخ تست‌های فصل اول: مفاهیم اولیه
۲۵.....	تست‌های فصل دوم: مدل نهاد و رابطه.....
۲۶.....	پاسخ تست‌های فصل دوم: مدل نهاد و رابطه
۳۱.....	تست‌های فصل سوم: مدل رابطه‌ای.....
۴۰.....	پاسخ تست‌های فصل سوم: مدل رابطه‌ای
۱۱۱.....	تست‌های فصل چهارم: جبر رابطه‌ای.....
۱۲۴.....	پاسخ تست‌های فصل چهارم: جبر رابطه‌ای.....
۱۸۳.....	تست‌های فصل پنجم: حساب رابطه‌ای.....
۱۸۶.....	پاسخ تست‌های فصل پنجم: حساب رابطه‌ای
۱۹۵.....	تست‌های فصل ششم: SQL دستورات DML
۲۲۵.....	پاسخ تست‌های فصل ششم: SQL دستورات DML
۳۷۹.....	تست‌های فصل هفتم: SQL دستورات DDL و DCL
۳۸۷.....	پاسخ تست‌های فصل هفتم: SQL دستورات DDL و DCL
۴۲۹.....	تست‌های فصل هشتم: وابستگی تابعی
۴۳۵.....	پاسخ تست‌های فصل هشتم: وابستگی تابعی
۴۷۷.....	تست‌های فصل نهم: نرمال‌سازی
۴۸۷.....	پاسخ تست‌های فصل نهم: نرمال‌سازی
۵۴۹.....	تست‌های فصل دهم: شاخص‌گذاری اطلاعات
۵۵۰.....	پاسخ تست‌های فصل دهم: شاخص‌گذاری اطلاعات

تست‌های فصل اول: مفاهیم اولیه

(کارشناسی ارشد-آزاد ۷۶۹۷۱)

۱- مزایای یک DATA BASE نسبت به فایل‌های متعارف چیست؟

(۱) دستیابی مشترک (SHARED ACCESS) به داده‌ها

(۲) اطمینان از صحت داده‌ها (DATA VALIDATION) کم‌تر مورد نیاز است.

(۳) و ۴

(۴) کنترل حساب شده مقدار تکرار داده‌ها در DATA BASE

(کارشناسی ارشد-آزاد ۷۶۲)

۲- کدام یک از موارد زیر جزء وظایف DBA نمی‌باشد؟

(۱) نوشتمن Data Dictionary برای پایگاه داده‌ها

(۲) نظارت بر عملکرد پایگاه داده‌ها (Performance Monitoring)

(۳) تهیه رویه‌ها و استراتژی تهیه Back up و نحوه احیای پایگاه داده‌ها

(۴) تهیه Schema برای پایگاه داده‌ها

۳- در یک سیستم مدیریت پایگاه داده‌ها (DBMS) کدام یک امکانات زیر جزء عناصر اصلی تشکیل‌دهنده

(کارشناسی ارشد-دولتی ۷۶)

DMBS محسوب نمی‌شوند؟

(۱) امکان پردازش زبان طبیعی برای کار با پایگاه

(۲) امکان کار با داده‌ها به کمک یک Data Sub Language (DSL)

(۳) امکان تأمین جامعیت و بی‌نقصی (integrity) پایگاه

(۴) امکان تأمین ایمنی پایگاه

۴- دو مرحله از مراحل طراحی یک بانک اطلاعاتی عبارتند از طراحی ادراکی (Conceptual Design) و

طراحی منطقی (Logical Design) این دو چه تفاوت اساسی با هم دارند؟ (کارشناسی ارشد-دولتی ۷۵)

(۱) طراحی ادراکی به مدل خاصی مربوط می‌شود و پس از انتخاب مدل صورت می‌گیرد ولی طراحی منطقی به مدل خاصی بستگی ندارد.

(۲) طراحی ادراکی مکمل طراحی منطقی است و پس از آن انجام می‌گیرد.

(۳) طراحی منطقی به صورت کلی به سیستم می‌نگرد و با روش‌هایی مانند ER انجام می‌گیرد.

(۴) طراحی منطقی به مدل خاصی مربوط می‌شود و پس از انتخاب مدل صورت می‌گیرد ولی طراحی ادراکی به مدل خاصی بستگی ندارد.

۵- کدام یک از گزاره‌های زیر در مورد کاتالوگ (System Catalog) در بانک اطلاعاتی صحیح است؟

(مهندس کامپیوتر-دولتی ۷۶۴)

(۱) کاتالوگ فقط با احکام DML قابل تغییر است.

(۲) کاتالوگ فقط توسط برخی از احکام DDL قابل تغییر است.

(۳) کاتالوگ فقط با احکام DML و برخی از احکام DDL قابل تغییر است.

(۴) کاتالوگ فقط با احکام DDL قابل تغییر است.

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

(۱) تعریف یک رابطه جدید در SCHEMA

(۲) حذف یک رابطه از SCHEMA

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

(۴) ۱ و ۳

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

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

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

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

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

۸- در صورتی که زمان خواندن کل رکوردهای یک جدول به ترتیب کلید اصلی ملاک ارزیابی باشد، DMBS باید کدام

یک از ساختارهای ذیل را در سطح داخلی پایگاه داده مورد استفاده قرار دهد؟ (مهندسی IT- دولتی ۸۱)

(۱) Hash file (۲) Indexed- Sequential file

(۳) Pile file (۴) Multi Indexed file

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

(۱) سهولت بیان پرس و جوها (۲) سرعت پاسخ به پرس و جوها

(۳) کاهش حجم بانک اطلاعاتی (۴) سهولت تغییر مدل دیسک‌های بانک اطلاعاتی

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

(۱) کاهش حافظه اصلی مورد نیاز

(۲) سهولت بهینه‌سازی پرس و جو

(۳) افزایش احتمال بنبست تراکنش‌ها (۴) افزایش سرعت عملیات روی بانک اطلاعات

۱۱- کدام یک از ویژگی‌های ذیل از خصوصیات دید ادراکی (Conceptual View) می‌باشد؟ (مهندسی IT- آزاد ۸۱)

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

(۲) به صورت انتزاعی می‌باشد.

(۳) نشأت گرفته از نمودار ER می‌باشد.

(۴) هر سه مورد

۱۲- اگر اطلاعات مربوط به پایگاه داده یعنی مشخصات کاربران، مجوزهای دسترسی آن‌ها به پایگاه داده،

عملیات انجام شده و ... نگهداری شود به آن، چه می‌گویند؟ (مهندسی IT- آزاد ۸۶)

(۱) فرهنگ داده‌ها

(۲) کاتالوگ سیستم

(۳) زبان تعريف داده‌ها DDL (۴) زبان زیر داده DSL

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

۱) برخی از دستورات تراکنش اجرا گردند و برخی دیگر خیر.

۲) مدتی پس از اتمام تراکنش و commit شدن آن، DBMS آن را Undo می کند.

۳) در اجرای همزمان تراکنش ها، تراکنشی باعث غلط شدن نتیجه تراکنش دیگر شود.

۴) در میانه اجرای تراکنش برخی قوانین جامعیتی به طور وقت نقض گردند.

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

(مهندسى IT- دولتى ۸۸)

۱) کاهش هم زمانی عملیات

۲) کاهش امنیت اطلاعات

۳) افزایش زمان اجرای برنامه های کاربردی (Redundancy)

(مهندسى IT - آزاد ۸۸)

۱۵- کدام یک از گزینه های زیر نادرست است؟

۱) یک تراکنش نمی تواند قوانین جامعیتی را نقض کند.

۲) در یک پایگاه داده، تراکنش ها هم زمان می توانند به داده های مورد نظر خود آزادانه دسترسی داشته باشند و کنترل خاصی در این زمینه لازم نیست.

۳) یا همه دستورات یک تراکنش با موفقیت اجرا می شود یا هیچ یک اجرا نمی شود.

۴) نتیجه یک تراکنش پس از تکمیل و اتمام آن نباید در اثر حوادث غیر مترقبه از بین برود.

۱۶- کدام گزینه مفهوم ناسازگاری داده ای (Data Inconsistency) را به درستی بیان می کند؟

(مهندسى IT- آزاد ۸۸)

۱) مقدار داده صفت با نوع آن متفاوت باشد.

۲) داده های پایگاه داده با اندیس (Index) آن سازگار نباشد.

۳) داده های ذخیره شده در مورد یک چیز در قسمت های مختلف پایگاه داده با یکدیگر متفاوت باشند.

۴) مقدار داده صفت با مقدار ذخیره شده در اندیس (Index) ساخته شده روی آن صفت یکسان نباشد.

(مهندسى IT- دولتى ۸۹)

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

۱) سهولت دسترسی هم زمان به اطلاعات

۲) امنیت فیزیکی اطلاعات، بیشتر

۳) سازگاری اطلاعات، بیشتر

(مهندسى IT- دولتى ۸۹)

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

۱) تغییر ساختمان داده ها در سطح منطقی

۲) تغییر محتوی داده ها در سطح منطقی

۳) تغییر دیسک حاوی بانک اطلاعاتی

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

(مهندسى IT- دولتى ۸۹)

۱) سرعت بیشتر در دسترسی به اطلاعات

۲) سهولت کار برای کاربر نهایی

۳) امنیت بیشتر اطلاعات

۲۰- کدام یک جزء وظایف DBMS نمی‌باشد؟ (مهندسی کامپیووتر- آزاد ۸۹ گروه الف)

- (۱) حفظ جامعیت و امنیت داده‌ها
- (۲) به روز رسانی کاتالوگ سیستم
- (۳) ناظر بر کارایی و پاسخ به تغییر نیازها
- (۴) ترمیم و حفظ سازگاری داده‌ها

۲۱- اطلاعات ذخیره شده در کاتالوگ سیستم شامل کدام گزینه نیست؟ (مهندسی کامپیووتر- آزاد ۸۹ گروه الف)

- (۱) مقادیر داده‌های درون جداول
- (۲) نام موجودیت‌ها و ارتباطات بین آن‌ها
- (۳) نام صفات خاصه هر موجودیت
- (۴) نام ساختارهای داده‌ای

۲۲- کدام مورد از ویژگی‌های معماری توزیع شده در پایگاه داده‌ها نیست؟ (مهندسی کامپیووتر- آزاد ۸۹ گروه ب)

- (۱) داده‌ها به بخش‌هایی تقسیم و در سایت‌ها توزیع شده‌اند.

(۲) سیستم چنان عمل می‌کند که از نظر کاربران شیوه به یک پایگاه داده متمرکز به نظر می‌رسد.

(۳) داده‌های ذخیره شده در هر سایت تحت کنترل یک سیستم مدیریت پایگاه داده‌ها هستند.

(۴) سیستم بانک اطلاعاتی به صورت یک ساختار دو بخشی در نظر گرفته می‌شود.

۲۳- به چه دلیل معماری سه سطحی ANSI در پایگاه داده اهمیت دارد؟ (مهندسی کامپیووتر- آزاد ۸۹ گروه ب)

(۱) معماری ANSI یک معماری مرجع برای تمام DBMS‌ها، فروشنده‌گان و توسعه‌دهندگان آن‌ها به وجود آورده است.

(۲) معماری ANSI استقلال داده‌ای فیزیکی و منطقی را تا حد امکان تضمین می‌نماید.

(۳) معماری ANSI یک روش مشترک برای کاربران در مشخص کردن استفاده از داده‌ها، مفاهیم داده‌ای و ذخیره‌سازی آن‌ها فراهم نموده است.

(۴) معماری ANSI توسط مؤسسه استاندارد ANSI تعریف شده است.

۲۴- کدام یک از پایگاه داده‌های ذیل با موارد دیگر متفاوت است؟ (مهندسی کامپیووتر- آزاد ۸۹ گروه ب)

- | | | | |
|----------------|---------------|----------------|-------------|
| Multi-User (۴) | Workgroup (۳) | Enterprise (۲) | Desktop (۱) |
|----------------|---------------|----------------|-------------|

۲۵- برای تضمین صحت و جامعیت تراکنش‌ها کدام یک از موارد زیر لازم است؟ (مهندسی IT- آزاد ۸۹)

- | | |
|----------------------------|--------------------------|
| (۱) یکپارچگی (Consistency) | (۲) هم خوانی (Atomicity) |
| (۳) پایانی (Durability) | (۴) همه موارد |

۲۶- کدام یک از موارد زیر از مزایای استفاده از کاتالوگ سیستم می‌باشد؟ (مهندسی IT- آزاد ۸۹)

- | | |
|---|---------------------------------------|
| (۱) استقلال داده‌ای (Data Independence) | (۲) امنیت (Security) |
| (۳) جامعیت (Integrity) | (۴) حفظ سازگاری داده‌ها (Consistency) |

۲۷- ساختن شاخص (Index) باعث ایجاد تغییر در کدام یک می‌شود؟ (مهندسی IT- آزاد ۸۹)

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

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

- | | |
|--------------------|------------|
| (System Catalogue) | (Relation) |
| (View) | (Schema) |
| ۲ | ۱ |
| ۴ | ۳ |

۲۹- کدام یک از موارد زیر مزیت سیستم بانک اطلاعاتی شی گرا بر سیستم بانک اطلاعاتی رابطه‌ای محسوب می شود؟
 (مهندسی کامپیووتر- دولتی ۹۱)

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

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

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

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

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

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

- | |
|---|
| الف) هر شیما (Schema) مشکل از تعدادی کاتالوگ است. |
| ب) هر شیما مشکل از تعدادی Data Object است. |
| ج) هر کاتالوگ مشکل از تعدادی شیما است. |
| د) هر Data Object می تواند در چند شیما باشد. |

- | |
|-------------------------|
| (۱) هر ۴ مورد درست است. |
| (۲) ب، ج و د |
| (۳) الف، ب و د |
| د) ب و ج |

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

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

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

- | |
|--|
| (۱) Commit توسط کاربر صادر گردیده و DBMS نمی تواند آن را لغو نماید. |
| (۲) Commit توسط کاربر صادر گردیده و DBMS می تواند آن را به تعویق بیندازد. |
| (۳) در یک تراکنش می تواند توسط کاربر صادر گردد، لیکن DBMS عهده‌دار تأیید نهایی آن می باشد. |

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

۳۵- کدام یک از گزینه‌های زیر جزء مزایای استقلال فیزیکی داده‌ها محسوب می‌گردد؟
(مهندسی کامپیوتر-آزاد ۹۳)

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

۳۶- در کدام یک از دستورات تراکنش، DBMS یک شناسه منحصر به فرد به آن تراکنش نسبت می‌دهد؟
(مهندسی کامپیوتر-آزاد ۹۳)

- | | | | | |
|--------|-------|--------|------------------|-------|
| outout | input | commit | commit and abort | start |
| ۴ | ۳ | ۲ | ۱ | |

۳۷- کنترل همروندي (Concurrency Control) به عنوان واحدی در DBMS عهده‌دار رسیدگی به کدام خاصیت است؟
(مهندسی IT-آزاد ۹۳)

- | | | | |
|------------|-------------|-----------|-----------|
| Durability | Consistency | Isolation | Atomicity |
| ۴ | ۳ | ۲ | ۱ |

۳۸- مصنونیت دید کاربران در قبال تغییراتی که در سطح ادراکی پدید می‌آید بر کدام گزینه تأکید می‌نماید؟
(مهندسی IT-آزاد ۹۳)

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

۳۹- تراکنش نهایی نشده Uncommit تراکنشی است که:
(مهندسی کامپیوتر-آزاد ۹۱)

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

۴۰- دید طراح بانک از داده‌های ذخیره شده در بانک به کدام لایه در معماری پیشنهادی ANSI باز می‌گردد؟
(مهندسی IT-آزاد ۹۳)

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

پاسخ تست‌های فصل اول: مفاهیم اولیه

۱- گزینه (۳) صحیح است.

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

۲- گزینه (۱) صحیح است.

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

۳- گزینه (۱) صحیح است.

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

۴- گزینه (۴) صحیح است.

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

۵- گزینه (۴) صحیح است.

کاتالوگ حاوی اطلاعات سیستمی پایگاه داده از قبیل مشخصات جداول، شاخص‌ها، دیدها و سطوح دسترسی کاربران است.

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

۶- گزینه (۲) صحیح است.

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

- گزینه (۳) صحیح است.

کاتالوگ سیستم در تمام مدل‌های بانک اطلاعاتی مانند رابطه‌ای و شبکه‌ای کاربرد دارد. در واقع بحث کاتالوگ پایگاه داده مستقل از بحث پایگاه داده است.

کاتالوگ سیستم پس از هر ذخیره و بازیابی اطلاعات مربوط به بانک اطلاعات بروزرسانی می‌گردد. پس کاتالوگ سیستم دخالتی در افزایش یا کاهش سرعت ذخیره و بازیابی اطلاعات نخواهد داشت.

نگهداری مشخصات سیستمی بانک اطلاعات در کاتالوگ سیستم و در فضایی جداگانه و مستقل از محتوای بانک اطلاعات باعث افزایش استقلال داده‌های محیط عملیاتی از محتویات کاتالوگ سیستم می‌گردد.

- گزینه (۱) صحیح است.

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

- گزینه (۴) صحیح است.

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

- گزینه (۴) صحیح است.

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

- گزینه (۲) صحیح است.

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

نشأت گرفته از نمودار ER می باشد.
دید طراح از داده های ذخیره شده در بانک اطلاعات مربوط به دید ادراکی خاص می باشد که توسط مدل رابطه ای بیان می گردد.

۱۲- گزینه (۲) صحیح است.

کاتالوگ سیستم و فرهنگ داده ها در کتب دیت و سیلبرشاتز به یک معنای یکسان بیان شده اند. اما از نگاه برخی کتب دیگر اطلاعات مربوط به مشخصات کاربران، حق دستیابی افراد به داده ها، تاریخ ایجاد و بروز در آوردن داده ها و اندازه جداول در کاتالوگ سیستم نگهداری می شود.
در آن کتب لغت نامه داده ها زیر مجموعه کاتالوگ سیستم در نظر گرفته شده است. اما کاتالوگ سیستم به دلیل کاربرد ویژه ای که دارد مجرزا شده است.

۱۳- گزینه (۴) صحیح است.

در یک نظام بانک اطلاعات کارآمد چهار کترل یکپارچگی، هم خوانی، انزوا و پایابی روی تمامی تراکنش ها اعمال می گردد تا جامعیت (صحت و سازگاری) آن تضمین شود.
گزینه اول: قاعده جامعیت یکپارچگی (atomicity) مانع می شود.
گزینه دوم: قاعده جامعیت پایابی (durability) مانع می شود.
گزینه سوم: قاعده انزوا (isolation) مانع می شود.
گزینه چهارم: در حین اجرای یک تراکنش ممکن است جامعیت (سازگاری) به طور موقت نقض شود ولی در انتهای کار تراکنش باید جامعیت (سازگاری) برقرار باشد.

۱۴- گزینه (۳) صحیح است.

گزینه اول: در سیستم های پایگاه داده همزمانی عملیات، به دلیل مجتمع بودن و اشتراکی شدن داده ها، افزایش یافته است.
گزینه دوم: در سیستم های پایگاه داده امنیت اطلاعات به دلیل تعریف سطوح دسترسی کاربران و مکانیزم های پشتیبان گیری افزایش یافته است.
گزینه سوم: در سیستم های پایگاه داده زمان اجرای برنامه های کاربردی به دلیل کترل مسائل امنیتی حقوق دستیابی (عبور از گیت امنیتی) افزایش یافته است، در واقع سرعت اجرا کاهش می یابد.
گزینه چهارم: در سیستم های پایگاه داده به دلیل مجتمع شدن داده های ذخیره شده و وحدت ذخیره سازی پدیده افزونگی کاهش یافته است.

۱۵- گزینه (۲) صحیح است.

گزینه اول: قاعده همخوانی است.
گزینه دوم: ناقص شرط انزوا در اجرای تراکنش های همرونند است. بر طبق قاعده جامعیت انزوا، همرونندی تراکنش ها باید کترل شود تا اثر مخرب روی هم نداشته باشند.
گزینه سوم: قاعده یکپارچگی است.
گزینه چهارم: قاعده پایابی است.

۱۶- گزینه (۳) صحیح است.

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

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

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

۱۷- گزینه (۲) صحیح است.

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

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

۱۸- گزینه (۱) صحیح است.

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

۱۹- گزینه (۴) صحیح است.

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

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

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

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

۲۰- گزینه (۳) صحیح است.

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

۲۱- گزینه (۱) صحیح است.

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

۲۲- گزینه (۴) صحیح است.

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

۲۳- گزینه (۳) صحیح است.

معماری ANSI برای پایگاه داده‌ها شامل سه سطح زیر است:

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

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

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

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

۲۴- گزینه (۱) صحیح است.

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

۲۵- گزینه (۴) صحیح است.

برای تضمین جامعیت تراکنش‌ها باید چهار خاصیت (که به خواص ACID معروفند) وجود داشته باشد:
a. خاصیت Atomicity (یک‌پارچگی): هر تراکنش یا به طور کامل جرا شود یا اصلاً اجرا نشود.

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

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

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

۲۶- گزینه (۱) صحیح است.

استفاده از کاتالوگ سیستم باعث افزایش استقلال داده‌ای می‌شود.

۲۷- گزینه (۲) صحیح است.

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

۲۸- گزینه (۲) صحیح است.

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

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

۲۹- گزینه (۴) صحیح است.

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

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

۳۰- گزینه (۲) صحیح است.

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

۳۱- گزینه (۴) صحیح است.

۳۲- گزینه (۴) صحیح است.

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

۳۳- گزینه (۴) صحیح است.

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

۳۴- گزینه (۱) صحیح است.

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

۳۵- گزینه (۴) صحیح است.

استقلال فیزیکی داده‌ها به معنی مصوبنیت برنامه‌های کاربردی در مقابل تغییراتی که در سطح فیزیکی (رسانه

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

۳۶- گزینه (۱) صحیح است.
با ایجاد و شروع هر تراکشن، یک شناسه توسط DBMS به آن نسبت داده می‌شود.

۳۷- گزینه (۲) صحیح است.
در بانک اطلاعات ممکن است تراکشن‌های همروند وجود داشته باشد (مثل سیستم‌های چند برنامه‌ای) بر طبق خاصیت انزوا، همروندی تراکشن‌ها باید کنترل شود تا اثر مخرب بر روی هم نداشته باشند، به بیان دیگر اثر تراکشن‌های همروند روی یکدیگر چنان است که گویا هر کدام در انزوا انجام می‌شود.
به تعریفی دیگر تراکشن‌ها جدا از یکدیگر هستند. اگر چند تراکشن به طور همزمان اجرا شوند، به‌هنگام سازی‌های هر کدام از یکدیگر مخفی می‌مانند تا به اتمام برسند.
به عبارتی دیگر، برای دو تراکشن مجزای A و B، تراکنش A می‌تواند بهنگام‌سازی‌ها B را پس از پذیرفته شدن آن (Commit) یا B می‌تواند بهنگام‌سازی‌های A را پس از پذیرفته شدن A ببیند. اما این دو تراکشن به طور همزمان نمی‌توانند بهنگام‌سازی‌های یکدیگر را ببینند. این کنترل‌ها توسط بخشی از DBMS به نام واحد کنترل هم‌روندی (Concurrency control) انجام می‌شود.

۳۸- گزینه (۳) صحیح است.
استقلال منطقی داده‌ها، به معنی مصونیت برنامه‌های کاربردی در قبال تعاریف و تغییراتی که در سطح ادراکی پایگاه داده بروز می‌کند، می‌باشد. یعنی تعریف و تغییر تصویر ادراکی بانک از دید برنامه‌های کاربردی آن‌ها مخفی بماند.

۳۹- گزینه (۲) صحیح است.
یک تراکشن از لحظه شروع به اجرا تا لحظه خاتمه برای پذیرش (commit) یا عدم پذیرش (abort) از سوی DBMS یک تراکشن فعال یا تراکشن نهایی نشده (uncommit) محسوب می‌گردد. همچنین اگر تراکشن پس از خاتمه، مورد تأیید DBMS قرار نگیرد، تراکشن مورد نظر از سوی DBMS یک تراکشن نهایی نشده (uncommit) یا ساقط شده (abort) محسوب می‌گردد.

۴۰- گزینه (۲) صحیح است.
معماری ANSI برای پایگاه داده شامل سه لایه زیر است:
۱- لایه خارجی.
۲- لایه ادراکی (شامل زیر لایه‌های ادراکی عام (طراحی ادراکی) و ادراکی خاص (طراحی منطقی)).
۳- لایه داخلی (فیزیکی).

توجه: به جای کلمه «لایه»، کلمه «تصویر» نیز مورد استفاده قرار می‌گیرد.
به طور کلی برای ایجاد بانک اطلاعات یک محصول نرم‌افزاری مرحله مدل‌سازی بانک شامل مراحل تحلیل (تصویر ادراکی عام)، طراحی (تصویر ادراکی خاص) و مرحله پیاده‌سازی (تصویر فیزیکی) مطابق شکل زیر انجام می‌گردد:

دیدهای کاربران (view) مختلف	کاربر ۱ ... کاربر ۲ کاربر ۱	تصویر خارجی
کل بانک بدون توجه به مدل خاصی	<p>مدل ER</p>	تصویر ادراکی عام (طراحی ادراکی)
کل بانک در قالب مدل انتخابی	<p>مدل رابطه‌ای</p>	تصویر ادراکی خاص (طراحی منطقی)
کل بانک روی رسانه	<p>SQL</p>	تصویر فیزیکی

تحلیل

طراحی

پیاده‌سازی

لایه خارجی

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

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

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

لایه ادراکی

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

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

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

لایه داخلی (فیزیکی)

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

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

تست‌های فصل دوم: مدل نهاد و رابطه

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

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

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

- (۱) ممکن است پروژه‌ای کارمند نداشته باشد.

 (۲) هر کارمند حداقل در یک پروژه کار می‌کند.
 (۳) ممکن است کارمندی در هیچ پروژه‌ای کار نکند.
 (۴) در یک پروژه ممکن است چندین کارمند کار کنند.

۳- کدام یک از موارد زیر برای متمازیز کردن نمونه موجودیت‌ها از یک‌دیگر مورد استفاده قرار می‌گیرد؟
 (مهندسى کامپیووتر-آزاد ۸۸)

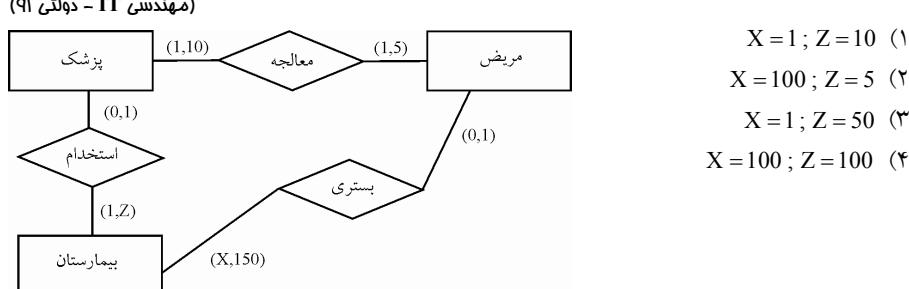
- (۱) ترتیب صفات خاصه (۲) اسمای صفات خاصه (۳) مقادیر صفات خاصه (۴) ترکیب صفات خاصه

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

- الف (۱) فقط نمودار الف

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

۵- یک دیاگرام ER، ناهمگون خوانده می‌شود اگر هیچ امکانی برای ارضای تمامی محدودیت‌های آن وجود نداشته باشد، کدام یک از پاسخ‌های زیر موجب می‌شود تا دیاگرام زیر ناهمگون گردد:
 (مهندسى IT - دولتی ۹۱)



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

- (۱) با توجه به شرایط مسئله مشخص گردیده و حداقلی برای آن نمی‌توان در نظر گرفت.

- 5 (۴) 4 (۳) 3 (۲)

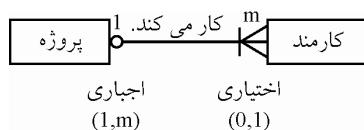
پاسخ تست‌های فصل دوم: مدل نهاد و رابطه

۱- گزینه (۲) صحیح است.

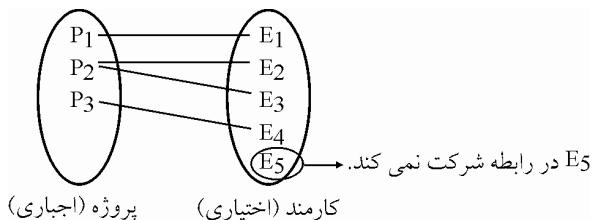
صفت مشتق در موجودیت وجود خارجی ندارد و در نمودار ER به صورت یک بیضی خط‌چین ترسیم می‌شود. صفت مشتق صفتی است که به کمک صفات‌های دیگر می‌توان آن را محاسبه کرد. مثلاً سن، یک صفت مشتق است که با توجه به تاریخ تولد قابل محاسبه می‌باشد. تصمیم‌گیری در مورد صفت مشتق به عهده طراح است مثلاً معدل کل برای دانشجو بهتر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

۲- گزینه (۱) صحیح است.

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



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



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

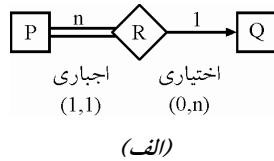
۳- گزینه (۳) صحیح است.

وجه تمایز موجودیت‌ها (سامی عام) در صفات آن‌ها است.

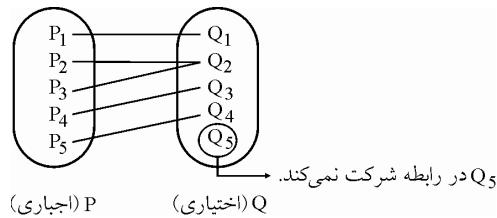
مانند موجودیت‌های استاد و دانشجو که دارای صفات متفاوت مانند شماره استادی و شماره دانشجویی هستند و وجه تمایز نمونه موجودیت‌ها (سامی خاص) در مقادیر آن‌ها است. مانند مقادیر متفاوت داده‌های مربوط به هر یک از دانشجویان. مثلاً برای موجودیت دانشجو مقادیر صفات خاصه مثل نام و نام خانوادگی نمونه‌ها را از یک‌دیگر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

۴- گزینه (۳) صحیح است.

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



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

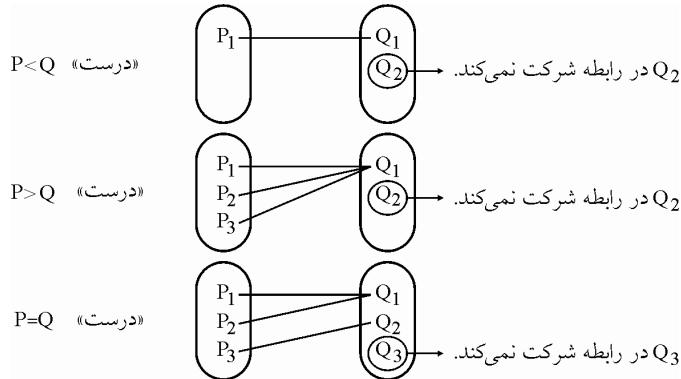


با توجه به ارتباط چند به یک P با Q و اجباری بودن حضور نمونه موجودیت‌های P ، قید $(1,1)$ برای P در نظر گرفته می‌شود. بنابراین رابطه زیر را داریم:

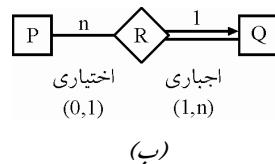
یک نمونه موجودیت P حداقل با یک و حداقل با یک نمونه موجودیت از Q رابطه دارد.

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

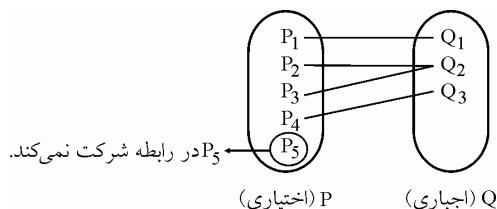
یک نمونه موجودیت Q حداقل هیچ (مانند Q_5) و حداقل با n نمونه موجودیت از P رابطه دارد. بنابراین تعداد نمونه موجودیت‌های P کوچک‌تر یا بزرگ‌تر با مساوی با تعداد نمونه موجودیت‌های Q است. چون ارتباط از P اجباری به Q اختیاری چند به یک است و نه یک به چند.



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

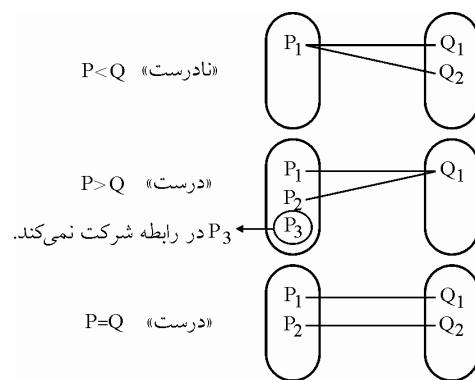


شکل زیر گویای نمودار ER است: (به طور مثال)



با توجه به ارتباط چند به یک P با Q و اختیاری بودن حضور نمونه موجودیت‌های P , قید $(0,1)$ برای P در نظر گرفته می‌شود بنابراین رابطه زیر را داریم:
یک نمونه موجودیت P حداقل هیچ (مانند P_5) و حداقل با یک نمونه موجودیت از Q رابطه دارد، یعنی ممکن است نمونه موجودیت‌هایی از P وجود داشته باشند که با هیچ نمونه‌ای از موجودیت Q رابطه ندارند.
(مانند (P_5))

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



۵- گزینه (۲) صحیح است.

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

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

با توجه به دیاگرام:

مقدار x حداقل تعداد بیماران در بیمارستان

و مقدار z حداکثر تعداد پزشکان در بیمارستان است.

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

گزینه اول:

۱- هر بیمارستان می‌تواند حداقل ۱ ($x=1$) و حداکثر ۱۵۰ بیمار داشته باشد.

۲- هر بیمارستان می‌تواند حداقل ۱ و حداکثر ۱۰ ($z=10$) پزشک در استخدام داشته باشد.

۳- هر پزشک می‌تواند حداقل در هیچ و حداکثر در ۱ بیمارستان استخدام باشد.

۴- هر پزشک می‌تواند حداقل ۱ و حداکثر ۱۰ بیمار داشته باشد.

۵- هر بیمار می‌تواند حداقل بسترهای نباشد و حداکثر در ۱ بیمارستان بسترهای باشد.

۶- هر بیمار می‌تواند حداقل ۱ و حداکثر ۵ پزشک معالج داشته باشد.

زمانی که حداقل تعداد بیماران ۱ نفر ($x=1$) و حداکثر تعداد پزشکان استخدام شده در بیمارستان ۱۰ نفر ($z=10$) باشد، تمامی محدودیت‌ها ارضاء شده، بنابراین نمودار ER ناهمگون خواهد بود.

با توجه به مورد ۴ برای معالجه حداقل ۱ نفر بیمار ($x=1$) نیاز به ۱ نفر پزشک داریم و با توجه به مورد ۲ حداقل ۱ و حداکثر ۱۰ نفر پزشک ($z=10$) داریم که محدودیت ارضاء می‌شود. با ۱۰ نفر پزشک می‌توان حداکثر ۱۰۰ نفر بیمار را معالجه کرد و با توجه به حداکثر ۱۵۰ نفر بیمار بنابراین با حداکثر ۱۰۰ نفر بیمار این امکان وجود دارد که این محدودیت هم ارضاء شود. چون حالت حداکثری است و امکان ارضاء دارد.

گزینه دوم:

۱- هر بیمارستان می‌تواند حداقل ۱۰۰ ($x=100$) و حداکثر ۱۵۰ بیمار داشته باشد.

۲- هر بیمارستان می‌تواند حداقل ۱ و حداکثر ۵ ($z=5$) پزشک در استخدام داشته باشد.

۳- هر پزشک می‌تواند حداقل در هیچ و حداکثر در ۱ بیمارستان استخدام باشد.

۴- هر پزشک می‌تواند حداقل ۱ و حداکثر ۱۰ بیمار داشته باشد.

۵- هر بیمار می‌تواند حداقل بسترهای نباشد و حداکثر در ۱ بیمارستان بسترهای باشد.

۶- هر بیمار می‌تواند حداقل ۱ و حداکثر ۵ پزشک معالج داشته باشد.

زمانی که حداقل تعداد بیماران ۱۰۰ نفر ($x=100$) و حداکثر تعداد پزشکان استخدام شده در بیمارستان ۵ نفر ($z=5$) باشد، تمامی محدودیت‌ها ارضاء نشده، بنابراین نمودار ER ناهمگون خواهد بود.

با توجه به مورد ۴ برای معالجه حداقل ۱۰۰ نفر بیمار نیاز به ۱۰ پزشک داریم، در حالی که با توجه به مورد ۲ حداقل ۱ و حداکثر ۵ نفر پزشک ($z=5$) در استخدام بیمارستان داریم و با این تعداد تنها ۵۰ بیمار را می‌توانیم معالجه کنیم و نه ۱۰۰ بیمار را. بنابراین تحت هیچ شرایطی این امکان وجود ندارد که این محدودیت ارضاء شود و این یعنی ناهمگونی.

گزینه سوم:

۱- هر بیمارستان می‌تواند حداقل ۱ ($x=1$) و حداکثر ۱۵۰ بیمار داشته باشد.

۲- هر بیمارستان می‌تواند حداقل ۱ و حداکثر $50(z=50)$ پزشک در استخدام داشته باشد.

۳- هر پزشک می‌تواند حداقل در هیچ و حداکثر در ۱ بیمارستان استخدام باشد.

۴- هر پزشک می‌تواند حداقل ۱ و حداکثر ۱۰ بیمار داشته باشد.

۵- هر بیمار می‌تواند حداقل بستره نباشد و حداکثر در ۱ بیمارستان بستره باشد.

۶- هر بیمار می‌تواند حداقل ۱ و حداکثر ۵ پزشک معالج داشته باشد.

زمانی که حداقل تعداد بیماران ۱ نفر ($x=1$) و حداکثر تعداد پزشکان استخدام شده در بیمارستان ۵۰ نفر

($z=50$) باشد، تمامی محدودیت‌ها ارضاء شده بنابراین نمودار ER ناهمگون نخواهد بود.

با توجه به مورد ۴ برای معالجه حداقل ۱ نفر بیمار ($x=1$) نیاز به ۱ نفر پزشک داریم و با توجه به مورد ۲

حداقل ۱ و حداکثر ۵۰ نفر پزشک ($z=50$) داریم که محدودیت ارضاء می‌شود. با ۵۰ نفر پزشک حداکثر

می‌توان ۵۰۰ نفر بیمار را معالجه کرد که این محدودیت هم ارضاء می‌شود.

گزینه چهار:

۱- هر بیمارستان می‌تواند حداقل $100(x=100)$ و حداکثر 150 بیمار داشته باشد.

۲- هر بیمارستان می‌تواند حداقل ۱ و حداکثر $100(z=100)$ پزشک در استخدام داشته باشد.

۳- هر پزشک می‌تواند حداقل در هیچ و حداکثر در ۱ بیمارستان استخدام باشد.

۴- هر پزشک می‌تواند حداقل ۱ و حداکثر ۱۰ بیمار داشته باشد.

۵- هر بیمار می‌تواند حداقل بستره نباشد و حداکثر در ۱ بیمارستان بستره باشد.

۶- هر بیمار می‌تواند حداقل ۱ و حداکثر ۵ پزشک معالج داشته باشد.

زمانی که حداقل تعداد بیماران ۱۰۰ نفر ($x=100$) و حداکثر تعداد پزشکان استخدام شده در بیمارستان ۱۰۰

نفر ($z=100$) باشد، تمامی محدودیت‌ها ارضاء شده بنابراین نمودار ER ناهمگون نخواهد بود.

با توجه به مورد ۴ برای معالجه حداقل ۱۰۰ نفر بیمار ($x=100$) نیاز به ۱۰ نفر پزشک داریم، و با توجه به

مورد ۲، حداقل ۱ و حداکثر ۱۰۰ نفر پزشک ($z=100$) داریم که محدودیت ارضاء می‌شود. با ۱۰۰ نفر

پزشک حداکثر می‌توان ۱۰۰۰ نفر بیمار را معالجه نمود که این محدودیت هم ارضاء می‌شود.

۶- گزینه (۳) صحیح است.

به تعداد موجودیت‌هایی که در یک ارتباط مشارکت دارند، درجه ارتباط گفته می‌شود. درجه ارتباط در

مدل ER عددی صحیح و کوچکتر از ۵ است. ارتباط‌های درجه ۱، ۲ و ۳ معمول، ارتباط درجه ۴ کمیاب و

غیر معمول است و ارتباط بالاتر از درجه ۴ قابل رسم کردن نیست.

تست‌های فصل سوم: مدل رابطه‌ای

(۵) (شناسی ارشد - دولتی ۷۳)

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

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

(۱) {C₁}

(۲) {C₂, C₃}

(۳) {C₁, C₃}

(۴) {C₂}

(کارشناسی ارشد - دولتی ۸۲)

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

(۱) کلید جستجو

(۲) کلید خارجی

(۳) کلید hash

(۴) نامزد کلیدی

- ۳- جدولی با n ستون دارای یک کلید اصلی و یک کلید کاندید دیگر هر کدام شامل یک صفت می‌باشد. این جدول چند ابرکلید (Super Key) دارد؟ (مهندسي کامپيوتر - دولتی ۸۱۴)

(۱) 2^{n-1}

(۲) $3 \times 2^{n-3}$

(۳) 2^{n-2}

(۴) $3 \times 2^{n-2}$

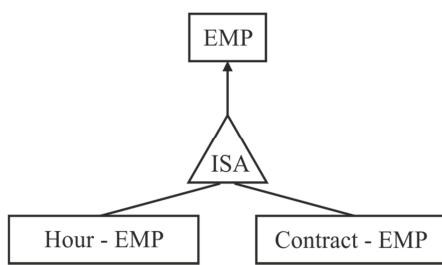
- ۴- رابطه ISA (ارث‌بری) در نمودار ER زیر را چگونه می‌توان در مدل رابطه‌ای پیاده‌سازی نمود؟ (مهندسي کامپيوتر - دولتی ۸۱۴)

(۱) رابطه ISA در مدل رابطه‌ای قابل پیاده‌سازی نیست.

(۲) همه ویژگی‌های EMP را در Hour-EMP و Contract-EMP کمی نماییم.

(۳) کلید اصلی Hour-EMP را در Hour-EMP و Contract-EMP به عنوان کلید اصلی و کلید خارجی مجددًا تعریف می‌کنیم.

(۴) موارد ۲ و ۳ هر دو صحیح می‌باشند.



- ۵- برای پیاده‌سازی رابطه IS-A در یک نمودار ER به مدل رابطه‌ای کدام یک از گزینه‌های زیر بهترین است؟ (مهندسي کامپيوتر - دولتی ۸۵)

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

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

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

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

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

SP		
S#	Pname	QTY
S ₄	مهره	20
S ₂	—	50
S ₄	بیچ	50
S ₅	مهره	20

S		
S#	Sname	Age
S ₁	احمدی	28
S ₅	حسینی	28
S ₂	جلالی	34
S ₄	—	32

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

(۲) نقض جامعیت ارجاع (Referential)

(۳) نقش جامعیت موجودیت (Entity Integrity)

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

(مهندسی IT-آزاد ۸۶)

۷- کدام گزینه از خواص رابطه (Relation) نیست؟

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

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

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

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

(مهندسی IT-دولتی ۸۶)

۸- کدام مورد در مدل رابطه‌ای جایز نیست؟

(۱) دامنه مرکب (Composite Domain)

(۲) رابطه تهی (Empty Relation)

(۳) ارتباط n:1 و ۱:n

(۴) ارتباط m:n و 1:n و 1:m

(مهندسی کامپیووت-آزاد ۸۷)

۹- کدام جمله در مورد کلید خارجی صحیح نیست؟

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

(۲) مقدار NULL برای آن مجاز است.

(۳) در جدول دیگر کلید اصلی است.

(۴) معمولاً برای Join جداول به کار می‌رود.

(مهندسی ذرمه‌افزار-دولتی ۸۸)

۱۰- کدام عبارت، در مورد کلید خارجی در مدل رابطه‌ای صحیح است؟

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

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

(۳) کلید خارجی بایستی خصیصه ساده باشد.

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

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

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

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

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

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

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

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

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

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

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

۱۶- برای تعیین کلید اصلی یک رابطه توجه به کدام یک از موارد زیر ضروری است؟
(مهندسی کامپیووتر- دولتی ۹۰)

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

۱۷- رابطه‌های r و s را در نظر بگیرید.

$r(\underline{R_1} : D_1, R_2 : D_2, R_3 : D_3)$

$s(\underline{R_2} : D_2, R_3 : D_3, R_4 : D_4)$

کدام زیر مجموعه از صفات r می‌تواند کلید خارجی باشد؟

R₂, R₃ (۴)

R₂ (۳)

R₃ (۲)

R₁, R₂ (۱)

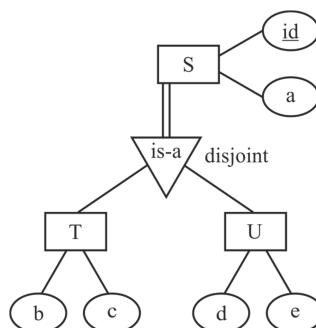
۱۸- تضمین سازگاری (Consistency) بانک‌های اطلاعاتی به کدام مورد مربوط نمی‌شود؟

(مهندسی کامپیوترا- آزاد ۹۱)

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

(مهندسی IT- دولتی ۹۱)

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



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

T(id,a,b,c) , U(id,a,d,e) (۱)

S(id,a) , T(id,b,c) , U(id,d,e) (۲)

R(id,a,b,c,d,e,Type) (۳) مقدار Type یکی از سه مقدار S,T یا U می‌باشد.

S(id,a,b,c,d,e,isT) و isU مقداری Flase یا True را می‌پذیرند. (۴)

(مهندسی IT- دولتی ۹۱)

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

(۱) اگر K₁ و K₂ کلیدهای کاندید در رابطه r باشند، آن‌گاه $K_1 \cup K_2$ نیز یک کلید کاندید در رابطه r است.

(۲) اگر K₁ و K₂ ابرکلیدهای (super key) در رابطه r باشند، آن‌گاه $K_1 \cap K_2$ نیز یک کلید کاندید در رابطه r است.

(۳) هر دو گزاره صحیح هستند.

(۴) تنها گزاره اول صحیح است.

(۱) هر دو گزاره صحیح هستند.

(۲) تنها گزاره دوم صحیح است.

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

(مهندسی کامپیوترا- آزاد ۹۱)

(۱) هر مقدار برای مجموعه‌ای از صفات در یک رابطه عیناً برای آن صفات در رابطه مقابله موجود باشد.

(۲) کلید اصلی در یک رابطه همان کلید کاندید در رابطه مقابله باشد.

(۳) کلید اصلی در یک رابطه همان مقدار کلید ثانویه در رابطه مقابله باشد.

(۴) همه موارد.

۲۲- کدام کلید در بانک اطلاعات، کمینه (minimal) نمی‌باشد؟

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

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

- (۱) جامعیت دامنه‌ای (۲) جامعیت درون رابطه‌ای
 (۳) خصیصه فرق، جامعیت محسوب نمی‌گردد.

۲۴- کدام گزینه بر «صحیح بودن هر رابطه به تنها یک تأکید می‌نماید؟

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

۲۵- در مدل رابطه‌ای پایگاه داده یک سیستم آموزشی، ارتباط یک به چند میان درس و پیش نیازهای آن، ارتباط چند به چند میان درس و دانشکده، و ارتباط یک به یک ریاست میان گروه آموزشی و استاد وجود دارد.

کدام عبارت نمایش مناسبتر هر رابطه را به ترتیب بیان می‌کند؟

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

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

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

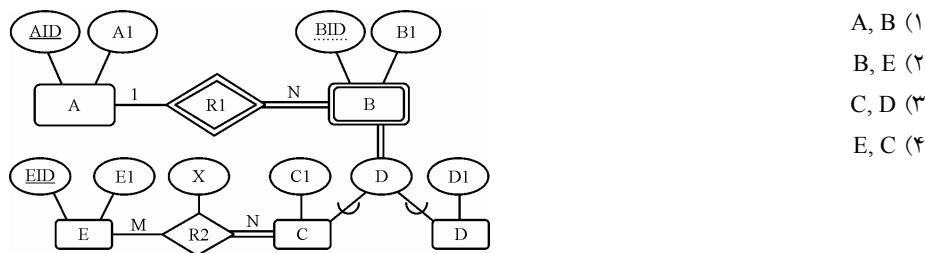
(۴) درج کلید اصلی درس در موجودیت درس به عنوان کلید خارجی، درج کلیدهای اصلی دانشکده و درس در یک موجودیت جدید به عنوان کلیدهای خارجی، درج کلید اصلی گروه آموزشی در موجودیت استاد به عنوان کلید خارجی.

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

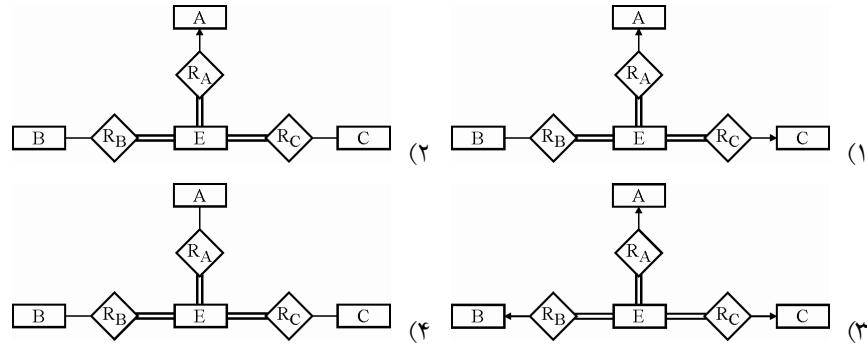
(مهندسو فناوری اطلاعات-دولتی ۹۵)

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

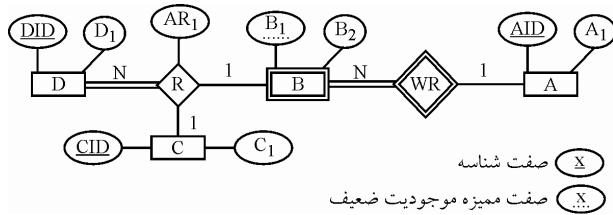
C (AID, BID, B1, C1) D (AID, BID, B1, D1) R2 (AID, BID, EID, X)



-۲۷- فرض کنید که رابطه سه‌تایی مقابل بین موجودیت‌های C, A, B وجود دارد، حال آگر بخواهیم این رابطه سه‌تایی را با رابطه‌های دودویی نمایش دهیم، کدام یک از نمودارهای موجودیت و رابطه (ERD) زیر دقیقاً معادل با این رابطه سه‌تایی می‌باشد؟ (مهندسی IT- دولتی ۸۸)



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



R(DID, AID, B1, CID, AR1) (۱)
P.K.

B(B1, B2, DID, CID, AR1) (۲)
P.K. F.K. F.K.

C(CID, C1, DID, AID, B1, AR1) (۳)
P.K. F.K. F.K.

D(DID, D1, CID, AID, B1, AR1) (۴)
P.K. F.K. F.K.

(مهندسی IT- دولتی ۹۶)

-۲۹- کدام مورد، درست نیست؟

(۱) می‌توان از سه ارتباط دوگانی یک ارتباط سه‌گانی را نتیجه گرفت.

(۲) موجودیت ضعیف وابسته به یک موجودیت قوی (مانند E₁) می‌تواند خود با موجودیت قوی دیگری (مانند E₂) رابطه داشته باشد.

(۳) برای حذف یک ارتباط سه‌گانی می‌توان به جای آن یک موجودیت ضعیف وابسته به هر سه موجودیت مشارکت‌کننده در ارتباط، در نظر گرفت.

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

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

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

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

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

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

۱) a) درست، b) درست ۲) a) درست، c) درست ۳) d) درست، b) نادرست ۴) b) نادرست، c) نادرست

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

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

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

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

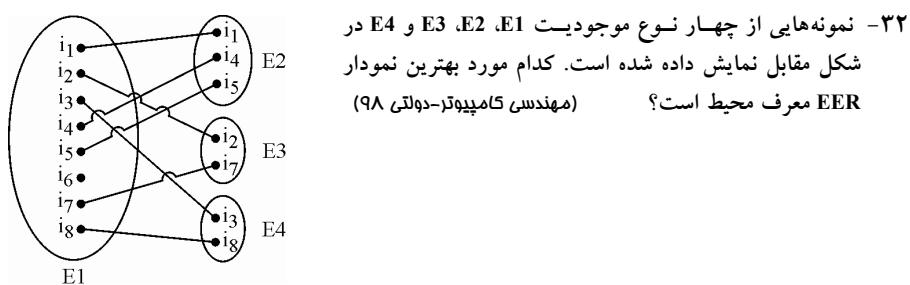
d) هر صفت مرکب، لزوماً تک مقداری است.

۱) یک گزاره نادرست است.

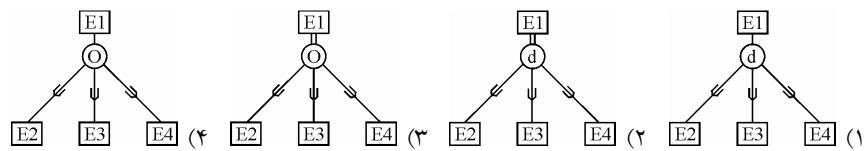
۲) دو گزاره نادرست است.

۳) سه گزاره نادرست است.

۴) چهار گزاره نادرست است.



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

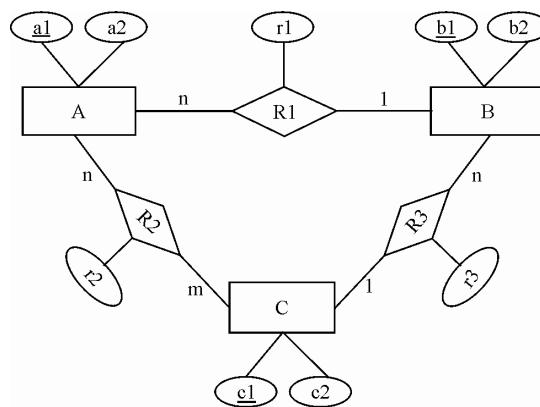


۳۳- در نمودار E-R، اگر رابطه is-A از نوع منفصل و کامل باشد، کدام مورد برای تبدیل نمودار به مدل رابطه‌ای مناسب‌تر است؟
(مهندسی IT- دولتی ۹۸)

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

(مهندسی IT - دولتی ۹۸)

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



A(a\underline{1},a2) B(b\underline{1},b2) C(c\underline{1},c2) R1(a\underline{1},b\underline{1},r1) R2(a\underline{1},c\underline{1},r2) R3(b\underline{1},c\underline{1},r3) (۱)

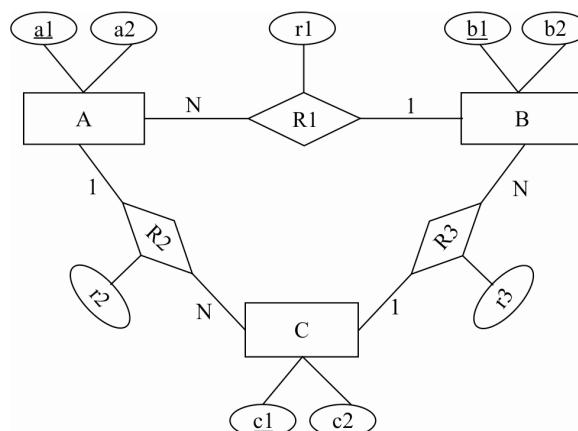
A(a\underline{1},a2) B(b\underline{1},b2,a\underline{1},r1) C(c\underline{1},c2,b\underline{1},r3) R1(a\underline{1},b\underline{1},r1) R3(b\underline{1},c\underline{1},r3) (۲)

A(a\underline{1},a2,b\underline{1}) B(b\underline{1},b2,c\underline{1},r1) C(c\underline{1},c2,b\underline{1},r3) R2(a\underline{1},c\underline{1},r2) (۳)

A(a\underline{1},a2,b\underline{1},r1) B(b\underline{1},b2,c\underline{1},r3) C(c\underline{1},c2) R2(a\underline{1},c\underline{1},r2) (۴)

(مهندسی کامپیوتر - دولتی ۹۹)

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

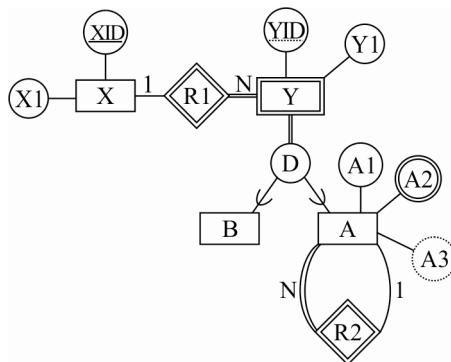


$$\begin{aligned}
 & A(\underline{a}_1, a_2, b_1, \eta) B(\underline{b}_1, b_2, c_1, r_3) C(\underline{c}_1, c_2, a_1, r_2) \quad (1) \\
 & A(\underline{a}_1, a_2, b_1, \eta) B(\underline{b}_1, b_2, c_1, r_3) C(\underline{c}_1, c_2, a_1, \underline{r}_2) \quad (2) \\
 & A(\underline{a}_1, a_2) B(\underline{b}_1, b_2, a_1, \eta) C(\underline{c}_1, c_2, b_1, r_3) R_1(a_1, \underline{b}_1, \eta) R_3(b_1, \underline{c}_1, r_3) \quad (3) \\
 & A(\underline{a}_1, a_2) B(\underline{b}_1, b_2) C(\underline{c}_1, c_2) R_1(a_1, b_1, \eta) R_2(a_1, \underline{c}_1, r_2) R_3(b_1, \underline{c}_1, r_3) \quad (4)
 \end{aligned}$$

- ۳۶- در نمودار EER زیر، طراحی منطقی رابطه یا رابطه‌های مربوط به موجودیت A، کدام مورد خواهد بود؟

(در گزینه‌های داده شده، زیر خط ممتد به معنای کلید اصلی و خط چین به معنای کلید خارجی است.)

(مهندسی کامپیووتر-دولتی ۹۹)



$$\begin{aligned}
 & A(YID, \underline{YIDR}2, A1, A2, A3) \quad (1) \\
 & A(YID, YIDR2, A1, A2) \quad (2) \\
 & AA2(YID, A2) \quad A(YID, Y1, YIDR2, A1) \quad (3) \\
 & AA2(XID, YID, A2) \quad A(XID, YID, Y1, XIDR, YIDR2, A1) \quad (4)
 \end{aligned}$$

(مهندسی کامپیووتر-دولتی ۹۹)

- ۳۷- چه تعداد از گزاره‌های زیر درست است؟

گزاره اول: تعداد کلیدهای کاندید یک رابطه از تعداد سویر کلیدهای آن رابطه همواره کمتر است.

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

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

گزاره چهارم: مدیریت تراکنش‌های همروند به عهده DBMS است.

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

پاسخ تست‌های فصل سوم: مدل رابطه‌ای

۱- گزینه (۳) صحیح است.

کلید باید خاصیت یکتاوی در جدول داشته باشد.

در ستون C_1 مقدار S و T بیش از یک بار تکرار شده است.

در ستون C_2 مقدار B بیش از یک بار تکرار شده است.

در ستون C_3 مقدار D_1 و D_2 بیش از یک بار تکرار شده است.

بنابراین هیچ یک از ستون‌های C_1 ، C_2 و C_3 خاصیت یکتاوی ندارند.

ترکیب ستون‌های $\{C_2, C_3\}$ خاصیت یکتاوی ندارد زیرا سطر BD_1 بیش از یک بار تکرار شده است.

ترکیب ستون‌های $\{C_1, C_3\}$ خاصیت یکتاوی دارد. بنابراین گزینه سوم درست است.

۲- گزینه (۲) صحیح است.

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

۳- گزینه (۱) صحیح است.

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

2^{n-2} (عدد ۲ در پایه به معنی وجود یا عدم وجود یک صفت است)

هر یک از حالات بالا با کلید اصلی یا کلید کاندید یا هر دو کلید، ابرکلید را ایجاد می‌کند. بنابراین داریم:
 $3 \times 2^{n-2}$

راه حل دوم: در این رابطه تعداد کل ابرکلیدها برابر است با:

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

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

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

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

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

راه حل سوم:

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

این جدول چند ابرکلید دارد؟

کلید اصلی		کلید کاندید		صفات غیرکلیدی		
A	B	C	D	E		

ابرکلید از حاصل جمع کلید کاندید یا کلید اصلی یا هر دو به علاوه صفات غیرکلیدی ایجاد می‌گردد.
 [صفات غیرکلید] + [کلید کاندید و کلید اصلی] یا [کلید کاندید] یا [کلید اصلی] = ابرکلید
 توجه: کلید کاندید یا کلید اصلی یا هر دو در ساخت ابرکلید نمی‌توانند تهی باشد.
 توجه: صفات غیرکلید حتی می‌توانند تهی باشد.
 با توجه به مطالب فوق داریم:

کلید اصلی		کلید کاندید		
A	B	0	1	
0	0	0		ϕ
0	1	1		B
1	0	0		A
1	1	1		AB

$$\Rightarrow 2^2 - 1 = 3$$

صفات غیر کلید				
C	D	E		
0	0	0	ϕ	
0	0	1	E	
0	1	0	D	
0	1	1	DE	
1	0	0	C	$\Rightarrow 2^3 = 8$
1	0	1	CE	
1	1	0	CD	
1	1	1	CDE	

بنابراین تعداد ابرکلیدها برابر حاصل ضرب دو مقدار فوق می‌باشد. $3 \times 8 = 24$ همچنین در یک رابطه‌ی کلی خواهیم داشت:

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

۴- گزینه (۴) صحیح است.

دو حل برای نگاشت رابطه ارثبری (IS-A) به مدل رابطه‌ای وجود دارد:

۱- تمام فیلد های پدر (EMP) در جدول های فرزند Contract-EMP و Hour-EMP اضافه شود. در این حالت دیگر نیازی به جدول EMP نیست.

۲- به ازای هر یک از موجودیت های پدر و فرزندان یک جدول در مدل رابطه‌ای تعریف شود. سپس فیلد های مشترک فرزندان در جدول EMP ذخیره شود و کلید اصلی جدول EMP در جدول های Contract-EMP و Hour-EMP اضافه شود.

البته گزاره بیان شده در گزینه سوم باید کمی اصلاح شود. به این ترتیب که ضرورتی ندارد کلید اصلی EMP در فرزندانش به عنوان کلید خارجی اضافه شود.

شود.

۵- گزینه (۱) صحیح است.

در رابطه IS-A معمولاً صفات مشترک بین چند رابطه را یافته و به عنوان پدر آن روابط در نظر می‌گیرند. بنابراین چون تمام اطلاعات رابطه پدر برای فرزندان لازم است، بهترین پیاده‌سازی با حداقل تکرار حالتی

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

۶- گزینه (۳) صحیح است.

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

۷- گزینه (۴) صحیح است.

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

۸- گزینه (۱) صحیح است.

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

۹- گزینه (۱) صحیح است.

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

۱۰- گزینه (۴) صحیح است.

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

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

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

مثال: نام رئیس D چیست؟

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

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

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

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

۱۱- گزینه (۳) صحیح است.

طبق تعریف، کلید خارجی می‌تواند مقدار NULL داشته باشد. هم‌چنان کلید خارجی یک رابطه می‌تواند به کلید کاندید همان رابطه ارجاع نماید، بنابراین الزام ذکر شده در گزینه دوم نادرست است.

مطابق قانون جامعیت ارجاعی برای برقرار بودن جامعیت ارجاع باید به ازای هر مقدار برای کلید خارجی یک مقدار برای کلید کاندید در جدول مرجع وجود داشته باشد اما نه بالعکس. بنابراین گزینه سوم درست و گزینه چهارم نادرست است.

۱۲- گزینه (۲) صحیح است.

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

۱۳- گزینه (۲) صحیح است.

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

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

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

مثال:

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

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

۱۴- گزینه (۴) صحیح است.

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

۱۵- گزینه (۲) صحیح است.

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

۱۶- گزینه (۱) صحیح است.

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

۱۷- گزینه (۳) صحیح است.

ستون R2 در جدول ۲ یک کلید خارجی است به جدول S. برای این‌که ستونی از یک جدول بتواند کلید خارجی به جدول دیگر باشد باید نوی آن با نوع کلید اصلی جدول دیگر برابر باشد. همچنین کلید خارجی یک رابطه باید در رابطه دیگر کلید کاندید (کلید اصلی یا فرعی) باشد. کلید کاندید (اصلی) رابطه S خصیصه R2 است بنابراین خصیصه R2 در رابطه S کلید خارجی آن است.

۱۸- گزینه (۲) صحیح است.

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

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

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

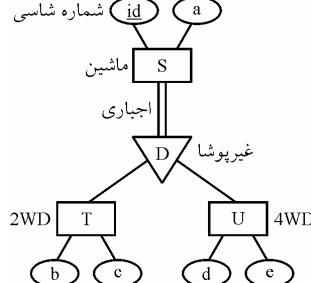
۱۹- گزینه (۱) صحیح است.

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

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

به طور کلی در شرایط total و disjoint بودن رابطه ارثبری، باید برای هر یک از فرزندان جدولی جدا در نظر گرفت که صفات پدر را هم در خود دارند و نیازی به جدولی برای پدر نیست (مشابه گزینه اول). مثال: رابطه ISA بر دو نوع پوشش (Overlap) و غیر پوشش (Disjoint) می‌باشد، مرسوم است که اگر رابطه پوشش باشد، داخل مثلث O و اگر غیر پوشش باشد داخل مثلث D قرار بگیرد. اگر ماشینی باشد، حتماً یک شماره شناسی دارد که حتماً برای آن باید یک ماشین باشد. یعنی شماره شناسی با نوع ماشین رابطه اجباری (Total) دارد. اما یک ماشین با یک شماره شناسی خاص نمی‌تواند هم دو محور باشد (2WD) و هم چهار محور (4WD) باشد. این یعنی غیر پوشایی (Disjoint) بین انواع ماشین‌ها.

مدل تحلیل (نمودار (ISA))



مدل طراحی (مدل رابطه‌ای)

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

<u>id</u>	a	b	c	
1	X	Y	Z	

T جدول

<u>id</u>	a	d	E
2	M	P	Q

U جدول

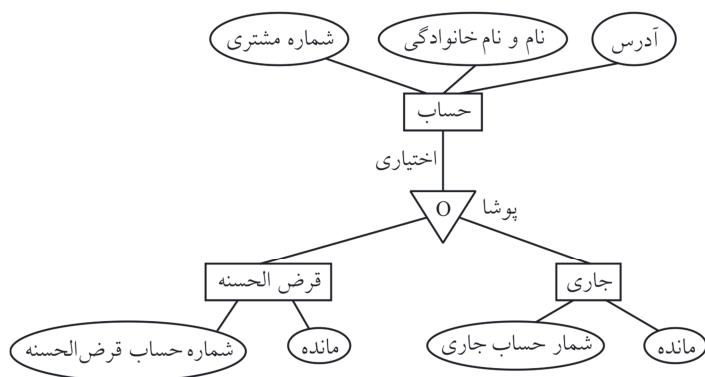
توجه: چون رابطه ماشین با نوع ماشین اجباری است، و به ازای هر شماره شاسی حتماً یک ماشین وجود دارد و به تبع عدم مقادیر NULL جلوی شماره شاسی و عدم نیاز به عمل الحق، نگاشت فوق به عنوان یک تبدیل ایده‌آل توصیه می‌گردد.

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

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

در این شرایط شماره مشتری با انواع حساب (قرض الحسنه یا جاري) رابطه اختياری (Partial) دارد. اما اگر یک شماره مشتری بخواهد صاحب حساب هم باشد، می تواند چندین نوع حساب (قرض الحسنه یا جاري) داشته باشد. این یعنی پوشایی (Overlap) بین انواع حساب ها.

۱- مدل تحلیل (نمودار ISA)



۲- مدل طراحی (مدل رابطه‌ای)

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

شماره مشتری	نام و نام خانوادگی	آدرس

مانده	شماره حساب جاری	شماره مشتری	مانده	شماره حساب قرض‌الحسنه	شماره حساب جاری
جدول حساب قرض‌الحسنه					جدول حساب جاری

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

توجه: در این نوع نگاشت، به ازای هر یک از موجودیت‌های پدر و فرزند یک جدول در مدل رابطه‌ای تعریف می‌شود و کلید اصلی جدول پدر به عنوان کلید خارجی در جداول فرزند تعریف می‌شود.

توجه: موجودیت حساب به شکل اختیاری در رابطه شرکت می‌کند، بدین معنی که به ازای هر شماره مشتری حساب، یک شماره حساب قرض‌الحسنه یا جاری ممکن است موجود باشد یا نباشد. اما موجودیت‌های جاری و قرض‌الحسنه به شکل اجباری در رابطه شرکت می‌کنند، بدین معنی که اگر یک شماره حساب قرض‌الحسنه یا جاری موجود باشد حتماً به یک شماره مشتری تعلق دارد.

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

توجه: نگاشت فوق، یک راه حل غیربهینه و ایده‌آل نیز دارد، بدین شکل که برای موجودیت پدر، جدول در نظر گرفته نشود و فقط برای موجودیت‌های فرزند جدول در نظر گرفته شود و تمام صفات پدر در جدول‌های فرزند درج گردند. این رویکرد، ساده‌ترین راه حل محاسبه می‌گردد، اما چنان‌چه گفته شد، راه حل غیربهینه و دارای تکرار اطلاعات و به تبع افزونگی است.

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

مانده	شماره حساب جاری	آدرس	نام و نامخانوادگی	شماره مشتری
جدول حساب جاری				

مانده	شماره حساب قرض‌الحسنه	آدرس	نام و نامخانوادگی	شماره مشتری
جدول حساب قرض‌الحسنه				

۲۰- گزینه (۲) صحیح است.

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

۲۱- گزینه (۱) صحیح است.

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

۲۲- گزینه (۲) صحیح است.

أنواع كلييد

(Supper key)

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

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

مثال: شماره دانشجویی یا شماره ملی

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

مثال:

S#	Sname	city
S1	Sn1	C1
S2	Sn2	C2
S3	Sn3	C2
S4	Sn4	C2

S#: ابرکلید است.

Sname: ابرکلید نیست.

(Sname, City): ابرکلید نیست.

(S#, Sname): ابرکلید است.

(S#, City): ابرکلید است.

مثال: تعداد ابرکلیدهای مثال فوق چند عدد است؟

پاسخ: برابر 4 عدد می‌باشد.

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

كلييد كانديد (Candidate key)

ابرکلیدی که عضو زائد نداشته باشد، کلید کاندید است.

مثال:

S#	Sname	City	S#	P#	QTY	P#	Pname	Color
S ₁	Sn ₁	C ₁	S ₁	P ₁	10	P ₁	Pn ₁	Red
S ₂	Sn ₂	C ₂	S ₁	P ₂	20	P ₂	Pn ₂	Blue
S ₃	Sn ₃	C ₂	S ₂	P ₁	30	P ₃	Pn ₃	Blue
جدول <i>S</i>			جدول <i>SP</i>			جدول <i>P</i>		

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

مثال:

شماره ملی: ابرکلید است. کلید کاندید نیز هست.

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

توجه: یک جدول می‌تواند چندین کلید کاندید داشته باشد.

مثال:

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

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

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

مثال: یک جدول تمام کلید.

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

کلید اصلی (Primary key)

کلید اصلی، یکی از کلیدهای کاندید است که بر اساس سلیقه طراح پایگاه داده، انتخاب می‌شود.

توجه: مطابق قوانین مدل رابطه‌ای، هر جدول حتماً باید یک کلید اصلی داشته باشد.

توجه: از آنجا که وظیفه کلید اصلی ایجاد تمايز میان سطرهای مختلف یک جدول است، هیچگاه کلید اصلی نمی‌تواند مقدار NULL داشته باشد.

توجه: اگر چندین کلید کاندید در یک جدول وجود داشته باشد، در انتخاب کلید اصلی همواره اولویت با کلید کاندیدی است که کمترین تعداد صفت را دارد.

توجه: هر جدول فقط می‌تواند یک کلید اصلی داشته باشد.

کلید فرعی (Alternative key)

هر کلید کاندیدی که به عنوان کلید اصلی انتخاب نشده است، یک کلید فرعی است.

مثال:

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

کلید خارجی (Foreign key)

کلید خارجی برای ارتباط میان جداول مورد استفاده قرار می‌گیرد.

تعریف: اگر صفت(هایی) در یک جدول به عنوان کلید خارجی تعریف شود.

اول اینکه: این صفت(ها) در جدول خودش شرط خاصی ندارد. و

دوم اینکه: این صفت(ها) در همان جدول یا جدول دیگری، کلید کاندید (اصلی یا فرعی) باید باشد.

مثال:

کلید اصلی			کلید خارجی			کلید خارجی			کلید اصلی		
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

مثال:

شماره دانشجویی			شماره ملی			شماره ملی			نام خانوادگی			نام		
کلید اصلی	کلید فرعی	کلید خارجی	کلید اصلی	کلید خارجی	کلید خارجی	کلید اصلی	کلید خارجی	کلید خارجی	کلید اصلی	کلید خارجی	کلید خارجی	کلید اصلی	کلید خارجی	کلید خارجی
R ₁			R ₂											

توجه: در مثال فوق، از آنجا که شماره ملی در جدول R₂، به طور ذاتی خاصیت کلیدی دارد، بنابراین کلید خارجی، خاصیت کلیدی دارد، اما الزاماً بر این شرط نبوده است.

توجه: # در جدول S، کلید کاندید است، پس عناصرش یکتاست.

توجه: # در جدول SP، به عنوان کلید خارجی، خاصیت کلیدی ندارد، پس ممکن است، مقادیر تکراری داشته باشد.

توجه: به ازای هر مقدار موجود در یک کلید خارجی، باید دقیقاً یک مقدار متناظر در کلید کاندید متناظر آن

وجود داشته باشد، در غیر این صورت می گوییم، کلید خارجی دارای ارجاع NULL است. به بیان دیگر، مقادیر کلید خارجی همه ادوار پایه \neq مجموعه مقادیر کلید کاندید باشد.

توجه: یک کلید خارجی در یک رابطه هیچگاه نباید ارجاع NULL داشته باشد، این مسئله را به عنوان یک قانون معتبر در مالکیت این انتشار می‌دانیم.

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 ₃	S ₂	P ₁	30	P ₃	Pn ₃	Blue
			S ₄	P ₃	40			
رجوع S			رجوع SP			رجوع P		
NULL			x ←			ارجاع		

درج رکورد (S4,P3,40) در جدول SP، غیر مجاز و غیر ممکن است، زیرا سبب ارجاع NULL می‌گردد.

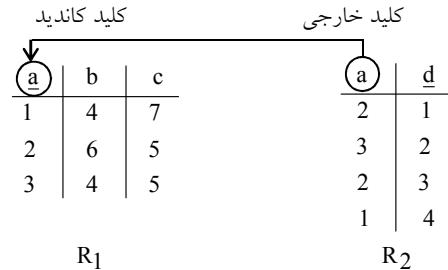
توجه: هر مقداری که در کلید خارجی وجود دارد، باید دارای مقدار متناظر در کلید کاندید مقصد باشد و لی عکس آن صادق نیست.

مثال: مانند S_3 که در جدول S قرار دارد ولی لزومی ندارد در جدول SP هم باشد. اما باید همه مقادیر

کلید خارجی در کلید کاندید باشد، از بالا به پایین می‌بارد، از پایین به بالا که نمی‌بارد!

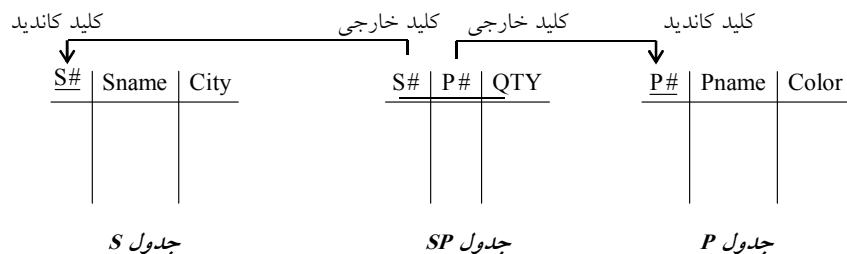
توجه: اگرچه کلید خارجی هیچگاه نباید، ارجاع NULL داشته باشد، اما می‌تواند مقدار NULL داشته باشد، البته به شرطی که کلید خارجی در شرایط قوانین بالا دستی قرار نگیرد.

مثال:



توجه: می‌توان مقادیر ستون a در جدول R₂ را NULL قرار داد.

مثال:



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

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

مثال: نام رئیس D چیست؟

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

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

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

مثال:

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

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

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

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

۱- اتومیک باشد.

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

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

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

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

۲۳- گزینه (۳) صحیح است.

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

۲۴- گزینه (۲) صحیح است.

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

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

۲۵- گزینه (۱) صحیح است.

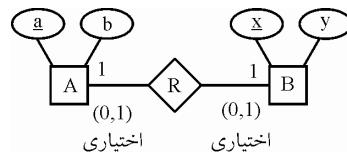
به طور کلی در مدل رابطه‌ای، هر موجودیت شناسایی شده در نمودار ER (فعالیت تحلیل) هنگام نگاشت به مدل رابطه‌ای (فعالیت طراحی) به یک جدول تبدیل می‌شود. همچنین صفات‌های موجودیت پس از نگاشت آن در مدل رابطه‌ای به صورت ستون‌های جدول بیان می‌شوند. همچنین ارتباط بین جدول از طریق کلید خارجی برقرار می‌گردد.

نگاشت رابطه یک به یک بین دو موجودیت به مدل رابطه‌ای

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

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



مدل طراحی:

$\underline{a} \mid b$	$a \mid x \mid \dots$	$\underline{x} \mid y$
جدول <i>A</i>	جدول <i>AB</i>	جدول <i>B</i>

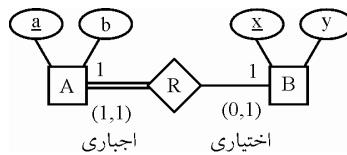
توجه: کلید کاندید جدول AB، ستون a یا ستون x می‌باشد. زیرا هر یک به تنها یک قدرت تفکیک سطرها را دارند. ترکیب ستون‌های a و x با هم به عنوان ax ایجاد کلید ایجاد می‌کنند.

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

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

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



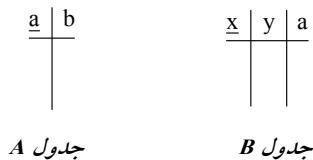
مدل طراحی:

<u>a</u>		b		x		y

توجه: این نگاشت منطقی هم هست، هرگاه سطحی در جدول موجودیت اختیاری وجود داشت، توسط کارخانه های ایرانی معرفی شده، تراجه ام و تواند بجهه کارخانه های ایرانی معرفی شده باشد.

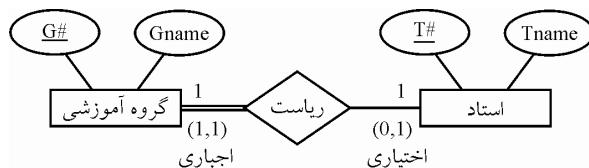
توجه: اگر بالعکس نگاشت فوق عمل کنیم و کلید کاندید جدول موجودیت اجرایی را در جدول موجودیت اختیاری به عنوان کلید خارجی تعریف کنیم، آنگاه جدول موجودیت اختیاری با مقادیر زیاد NULL مواجه می‌گردد.

مدل طراحی نادرست:



مثال:

مدل تحلیل:



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

مدل طراحی:

G #	Gname	T #
G ₁	Gn ₁	T ₁
G ₂	Gn ₂	T ₂

جدول گروه آموزشی

T #	Tname
T ₁	Tn ₁
T ₂	Tn ₂
T ₃	Tn ₃
T ₄	Tn ₄

جدول استاد

بنابراین تا همینجا واضح است که گزینه اول درست است.

مدل طراحی نادرست:

G #	Gname
G ₁	Gn ₁
G ₂	Gn ₂

جدول گروه آموزشی

T #	Tname	G #
T ₁	Tn ₁	G ₁
T ₂	Tn ₂	G ₂
T ₃	Tn ₃	NULL
T ₄	Tn ₄	NULL

جدول استاد

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

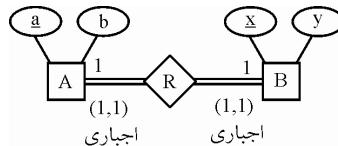
بنابراین گزینه‌های دوم، سوم و چهارم نادرست هستند.

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

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

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



مدل طراحی:

a	b	x	y

جدول AB

a	b	<u>x</u>	y

جدول AB

یا

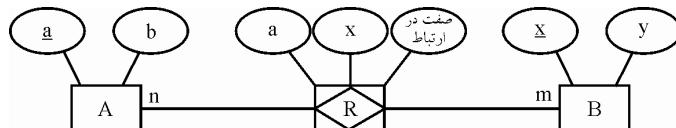
توجه: در موقع اختیاری، با ایجاد جداول دیگر، از تکرار بی‌رویه مقادیر NULL، جلوگیری می‌کنیم، ولی در موارد اجباری مانند حالت سو، ایجاد جداول دیگر، معنا ندارد.

نگاشت رابطه چند به چند بین دو موجودیت به مدل رابطه‌ای

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

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



مدل طراحی:

a	b

جدول A

a	x	...

جدول AB

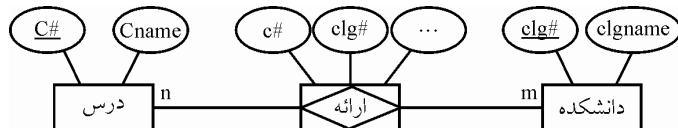
x	y

جدول B

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

مثال:

مدل تحلیل:



مدل طراحی:

C# Cname	C# c lg# ...	c lg# c lg name
جدول درس	جدول ارائه	جدول دانشکده

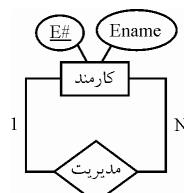
توجه: مثال فوق مطابق گزینه اول است.

نگاشت رابطه یک به چند در یک موجودیت به مدل رابطه‌ای مستقل از اختیاری یا اجباری بودن موجودیت، موجودیت به یک جدول تبدیل می‌گردد.

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

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

مدل تحلیل:



مدل طراحی:

E# Sname M#
S ₁ Sn ₁ NULL
S ₂ Sn ₂ S ₁
S ₃ Sn ₃ S ₁
S ₄ Sn ₄ S ₃
S ₅ Sn ₅ S ₂
S ₆ Sn ₆ S ₅

کلید خارجی (شماره مدیر)

↑ طوفه

کلید کارمند (شماره کارمندی)

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

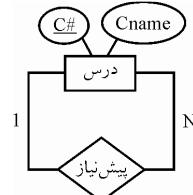
توجه: شماره مدیر، برای شخصی با شماره کارمندی S4 برابر S3 است و نام S3 برابر Sn3 است.

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

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

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

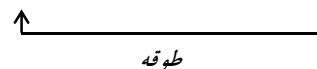
مدل تحلیل:



مدل طراحی:

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

کلید خارجی (شماره درس) کلید کاندید (پیش‌نیازها)



توجه: C2 و C3 پیش‌نیازهای درس C1 هستند و C6 پیش‌نیاز درس C5 و C1 پیش‌نیاز هیچ درسی نیست.

توجه: مطابق گزینه اول، کلید اصلی درس، به عنوان کلید خارجی در جدول فوق تعریف شده است.

_____ ۲۶ - گزینه (۲) صحیح است.

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

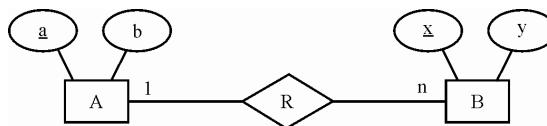
خارجی برقرار می‌گردد.

نگاشت رابطه یک به چندین 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:

$a b$	$x y a$
جدول A	جدول B

وابستگی وجودی

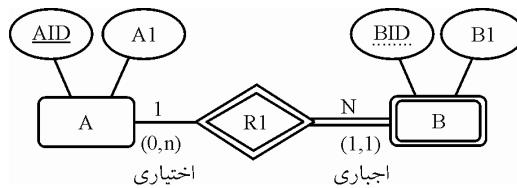
اگر در یک بانک اطلاعاتی، وجود یک موجودیت، وابسته به موجودیت دیگری باشد که در صورت حذف و تغییر موجودیت اصلی یعنی موجودیت قوی این موجودیت نیز تغییر کند، این نوع وابستگی را وجودی گفته و به پدیده وابسته، موجودیت ضعیف گویند. همچنین موجودیت ضعیف کلید موجودیت قوی را در بر دارد تا هرگونه تغییر یا حذف در موجودیت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

توجه: نماد خط مضاعف افقی نشانه اجباری بودن موجودیت چسبیده به آن است، اما نماد | به معنی یک و

الزام شرکت در رابطه نشانه اجباری بودن موجودیت طرف مقابل است.

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

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

مدل طراحی:

<u>AID</u>	A1	<u>AID</u>	<u>BID</u>	B1
جدول A		جدول B		

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

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

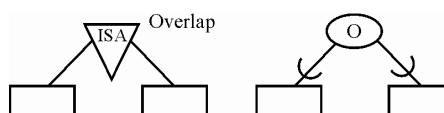
توجه: صفت ممیزه یا کلید جزئی به طور سراسری در یک موجودیت ضعیف یکتا نیست، بلکه فقط در بین نمونه‌ها یا دسته‌هایی که با موجودیت قوی ارتباط دارند، یکتا است.

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

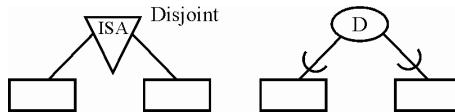
توجه: همانطور که واضح است، طراحی جدول B در صورت سوال به صورت (BID,B1) در نظر گرفته شده است، که مطابق آنچه بیان کردیم، طراحی درست آن به صورت (B1,AID,BID) است. همچنین طراحی جدول A در صورت سوال به صورت (AID,A1) در نظر گرفته شده است، که مطابق آنچه بیان کردیم، طراحی درست آن هم به صورت (AID,A1) است. بنابراین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

نگاشت رابطه ISA به مدل رابطه‌ای

در رابطه ISA رابطه پدر با فرزندان به دو صورت رابطه اختیاری یا جزئی (Partial) با نماد خط عمودی و رابطه اجباری یا کلی (Total) با نماد خط مضاعف عمودی است و رابطه فرزندان با پدر به دو صورت رابطه پوشای تخصیص غیرمجزا (Overlap) و رابطه غیرپوشای تخصیص مجزا (Disjoint) می‌باشد. رابطه پوشای تخصیص غیرمجزا (Overlap) مابین فرزندان و پدر به دو شیوه زیر نشان داده می‌شود:



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



بخشی از مدل EER رسم شده در صورت سوال، یک رابطه اجباری یا کلی (Total) مابین موجودیت B و موجودیت‌های C و D و یک رابطه غیرپوشای تخصیص مجزا (Disjoint) را مابین موجودیت‌های C و D و موجودیت B نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را بیان می‌کنیم. در یک رابطه اجباری یا کلی (Total)، هر نمونه از موجودیت پدر حتماً می‌بایست با یکی از نمونه موجودیت‌های فرزند در ارتباط باشد. برای مثال در این سؤال، هر نمونه از موجودیت B حتماً می‌بایست با یکی از نمونه موجودیت‌های C و D در ارتباط باشد. به عبارت دیگر نمی‌توان نمونه‌ای از موجودیت B داشت که با هیچ یک از نمونه موجودیت‌های C یا D در ارتباط نیست.

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

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

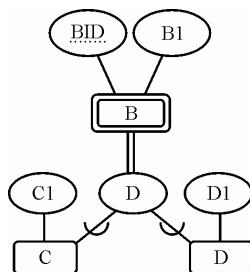
همچنین نمونه موجودیت‌های نوع ماشین که دو محور (2WD) یا چهارمحور (4WD) است نمی‌تواند به طور همزمان با نمونه‌ای از موجودیت ماشین که شماره شاسی یکتا و منحصر به فرد خود را دارد در ارتباط باشند، به عبارت دیگر نمی‌توان نمونه‌ای از موجودیت‌های نوع ماشین که دو محور (2WD) یا چهارمحور (4WD) است داشت که به طور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

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

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

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

مدل تحلیل (نمودار (ISA))



مدل طراحی (مدل رابطه‌ای)

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



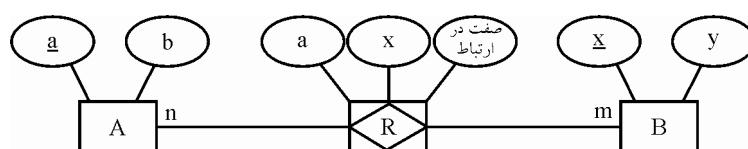
توجه: چون رابطه موجودیت B با موجودیت‌های C و D اجباری است، و رابطه موجودیت‌های C و D با موجودیت B از نوع Disjoint است. و به ازای هر نمونه موجودیت از B حتماً یک نمونه موجودیت از C و یا D وجود دارد و به تبع عدم مقادیر NULL جلوی نمونه موجودیت‌های B و عدم نیاز به عمل الحاق، نگاشت فوق به عنوان یک تبدیل ایده‌آل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

نگاشت رابطه چند به چند بین دو موجودیت به مدل رابطه‌ای

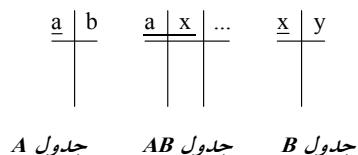
مستقل از اختیاری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد و یک جدول پل (Bridge) نیز به عنوان ارتباط دهنده دو جدول مورد استفاده قرار می‌گیرد. همچنین کلید کاندید جدول پل از ترکیب کلید کاندید دو جدول دیگر ایجاد می‌گردد.

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:

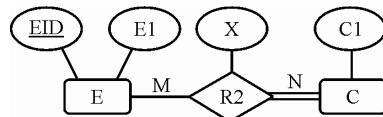


مدل طراحی:



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

مدل تحلیل:



مدل طراحی:

<u>EID</u>	<i>E1</i>	<u>AID</u>	<u>BID</u>	<u>EID</u>	<i>X</i>	<u>AID</u>	<u>BID</u>	<i>B1</i>	<i>C1</i>

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

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

۲۷- گزینه (۳) صحیح است.

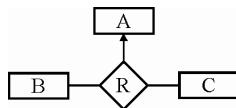
به طور کلی در مدل رابطه‌ای، هر موجودیت شناسایی شده در نمودار ER (مدل تحلیل) هنگام نگاشت به مدل رابطه‌ای (مدل طراحی) به یک جدول تبدیل می‌شود. همچنین صفت‌های موجودیت پس از نگاشت آن در مدل رابطه‌ای به صورت ستون‌های جدول بیان می‌شوند. همچنین ارتباط بین جدول از طریق کلید خارجی برقرار می‌گردد.

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

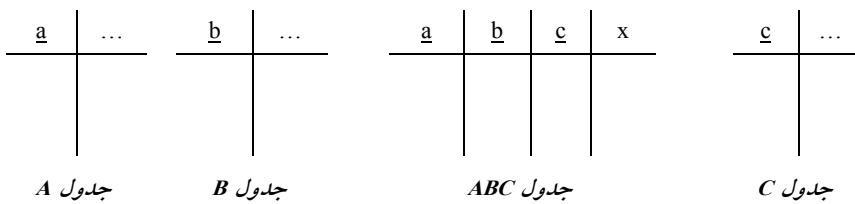
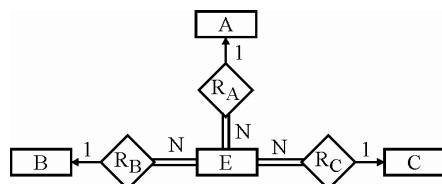
نگاشت رابطه‌چند به چند و یک به چند بین سه موجودیت به مدل رابطه‌ای مستقل از اختیاری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد و یک جدول

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

مدل تحلیل:



مدل طراحی:



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

همچنین ستون b در جدول ABC به عنوان کلید خارجی تعریف می‌گردد که به جدول B ارجاع می‌کند.

همچنین ستون c در جدول ABC به عنوان کلید خارجی تعریف می‌گردد که به جدول C ارجاع می‌کند.

توجه: کلید کاندید جدول ABC برابر abc است.

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

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

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

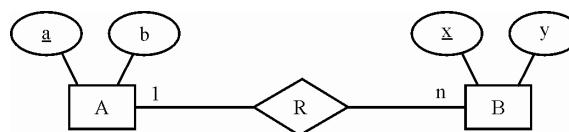
۲۸ - گزینه (۴) صحیح است.

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

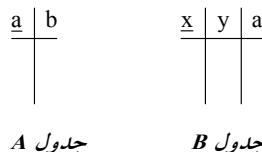
نگاشت رابطه یک به چند بین دو موجودیت به مدل رابطه‌ای مستقل از اختیاری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد. و کلید کاندید جدول یک در جدول چند به عنوان کلید خارجی تعریف می‌گردد.

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



مدل طراحی:



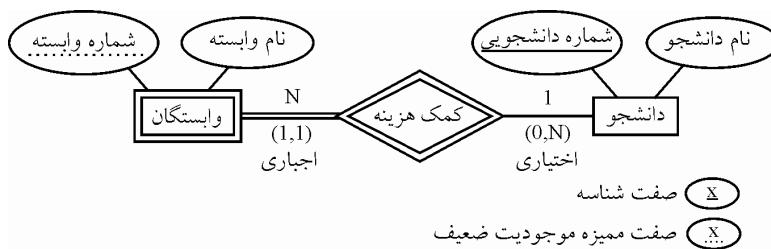
وابستگی وجودی

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

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

مثال: حالت اجباری و اختیاری

مدل تحلیل:



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

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

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

مدل طراحی:

<u>S#</u>	<u>SD#</u>	SDname	<u>S#</u>	Sname
		جدول وابستگان		جدول دانشجو

جدول دانشجو

جدول وابستگان

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

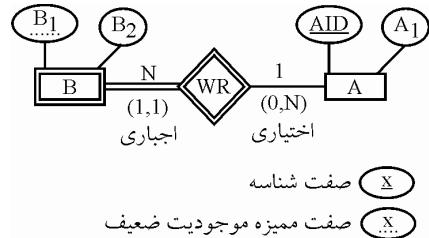
توجه: کلید کاندید جدول چند یعنی موجودیت ضعیف برابر ترکیب کلید خارجی و صفت ممیزه در جدول موجودیت ضعیف است. یعنی کلید کاندید جدول وابستگان برابر (S#,SD#) است.

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

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

حالات اجباری و اختیاری

مدل تحلیل:



صفت شناسه

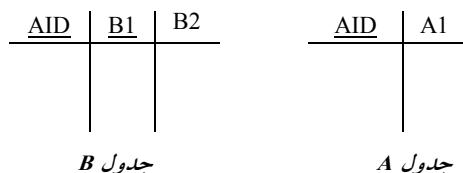
صفت ممیزه موجودیت ضعیف

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 توانایی به معنی صفر و عدم الزام شرکت در رابطه نشانه اختیاری بودن موجودیت طرف مقابل است.

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

مدل طراحی:



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

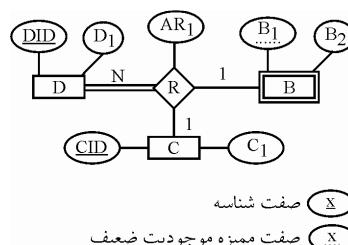
توجه: کلید کاندید جدول چند یعنی موجودیت ضعیف برای ترکیب کلید خارجی و صفت ممیزه در جدول موجودیت ضعیف است. یعنی کلید کاندید جدول B برای (AID,B1) است.

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

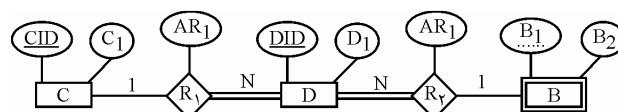
توجه: یک موجودیت ضعیف همیشه در ارتباطش با موجودیت قوی رابطه اجباری دارد. همچنین در ادامه نگاشت، بخشی از مدل EER رسم شده در صورت سوال، یک رابطه یک به چند و یک به یک بین سه موجودیت را نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را شرح می‌دهیم.

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

مدل تحلیل:



مدل طراحی:



<u>CID</u>	C1	<u>DID</u>	D1	<u>CID</u>	AID	B1	AR1	<u>AID</u>	B1	B2
	جدول C				جدول D				جدول B	

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

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

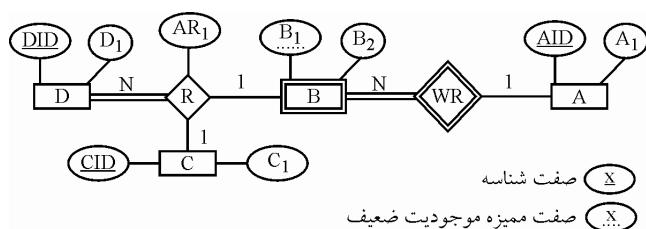
توجه: جدول B در نگاشت مرحله قبل ایجاد شده است که در اینجا دقیقاً به همان شکل و همان مشخصات، استفاده شده است.

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

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

۲۹- گزینه (۱) صحیح است.

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

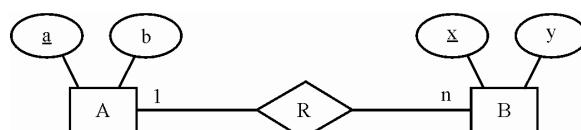


پاسخ: به طور کلی در مدل رابطه‌ای، هر موجودیت شناسایی شده در نمودار ER (مدل تحلیل) هنگام نگاشت به مدل رابطه‌ای (مدل طراحی) به یک جدول تبدیل می‌شود. همچنین صفت‌های موجودیت پس از نگاشت آن در مدل رابطه‌ای به صورت ستون‌های جدول بیان می‌شوند. همچنین ارتباط بین جدول از طریق کلید خارجی برقرار می‌گردد.

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

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



مدل طراحی:

$\begin{array}{ c c } \hline a & b \\ \hline & \\ \hline \end{array}$	$\begin{array}{ c c c } \hline x & y & a \\ \hline & & \\ \hline \end{array}$
جدول A	جدول B

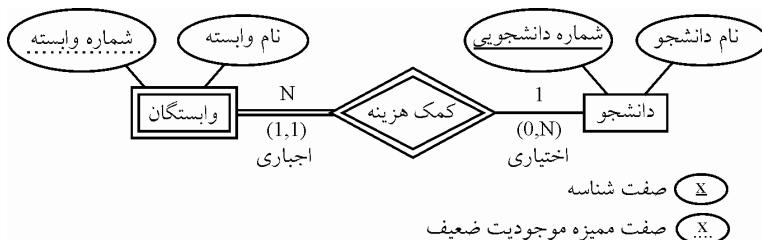
وابستگی وجودی

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

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

مثال: حالت اجباری و اختیاری

مدل تحلیل:



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

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

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

مدل طراحی:

S#	SD#	SDname	S#	Sname
		جدول وابستگان		جدول دانشجو

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 وایستگان برابر $(S\#, SD\#)$ است.

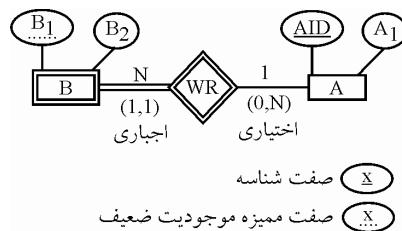
توجه: صفت ممیزه یا کلید جزئی به طور سراسری در یک موجودیت ضعیف یکتا نیست، بلکه فقط در بین نمونه‌ها یا دسته‌هایی که با موجودیت قوی ارتباط دارند، یکتا است.

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

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

حالت اجباری و اختیاری

مدل تحلیل:



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

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

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

• 196 •

<u>AID</u>	B1	B2	<u>AID</u>	A1
جدول	جدول	جدول	جدول	جدول

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

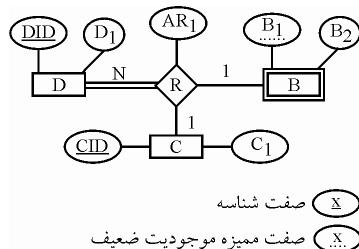
توجه: کلید کاندید جدول چند یعنی موجودیت ضعیف برابر ترکیب کلید خارجی و صفت ممیزه در جدول موجودیت ضعیف است. یعنی، کلید کاندید جدول B برابر (AID,B1) است.

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

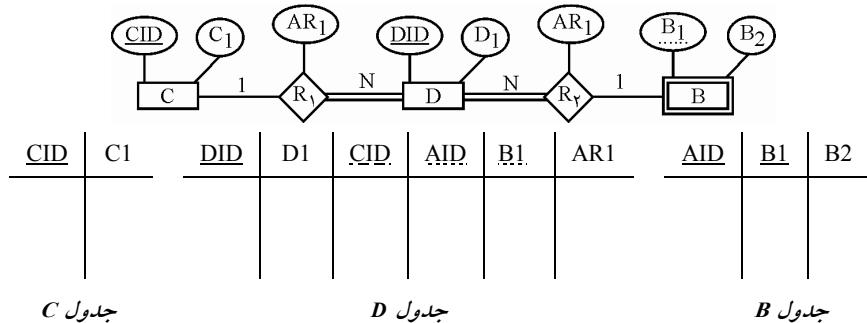
توجه: یک موجودیت ضعیف همیشه در ارتباطش با موجودیت قوی رابطه اجباری دارد. همچنین در ادامه نگاشت، بخشی از مدل ER رسم شده در صورت سوال، یک رابطه یک به چند و یک به یک بین سه موجودیت را نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را شرح می‌دهیم.

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

مدل تحلیل:



مدل طراحی:



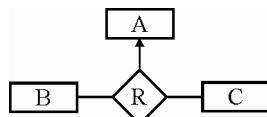
توجه: ستون CID در جدول D به عنوان کلید خارجی تعریف می‌گردد که به جدول C ارجاع می‌کند. همچنین ستون‌های (AID,B1) در جدول D به عنوان کلید خارجی تعریف می‌گردد که به جدول B ارجاع می‌کند.

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

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

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

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

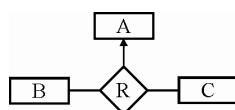


پاسخ: به طور کلی در مدل رابطه‌ای، هر موجودیت شناسایی شده در نمودار ER (مدل تحلیل) هنگام نگاشت به مدل رابطه‌ای (مدل طراحی) به یک جدول تبدیل می‌شود. همچنین صفت‌های موجودیت پس از نگاشت آن در مدل رابطه‌ای به صورت ستون‌های جدول بیان می‌شوند. همچنین ارتباط بین جداول از طریق کلید خارجی برقرار می‌گردد.

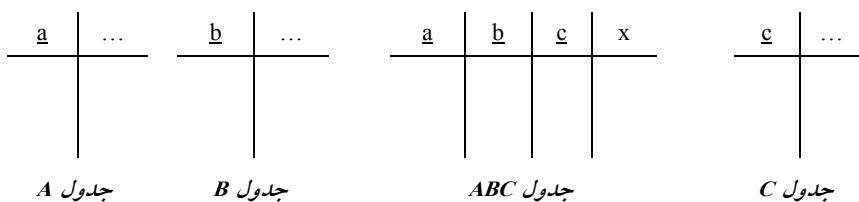
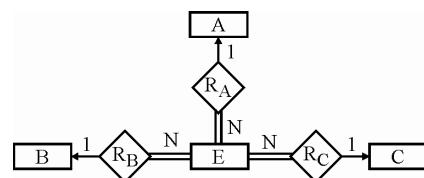
مدل ER رسم شده در صورت سوال، یک رابطه چند به چند و یک به چند بین سه موجودیت را نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را شرح می‌دهیم.

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

مدل تحلیل:



مدل طراحی:



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

همچنین ستون b در جدول ABC به عنوان کلید خارجی تعریف می‌گردد که به جدول B ارجاع می‌کند.

همچنین ستون c در جدول ABC به عنوان کلید خارجی تعریف می‌گردد که به جدول C ارجاع می‌کند.

توجه: کلید کاندید جدول ABC برابر abc است.

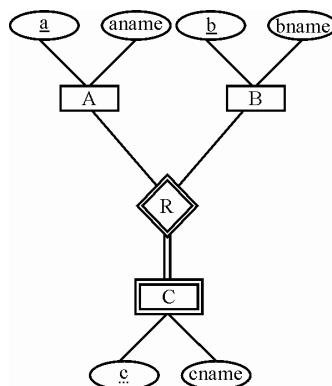
توجه: ستون a ، به عنوان یک صفت در ارتباط در جدول ABC به صورت مستر قرار می‌گیرد. زیرا همانطور که گفته شده متصل به رابطه در صورت وجود، درون جدول پُل مستر می‌شود.

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

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

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

مدل تحلیل:



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

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

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

مدل طراحی:

<u>a</u>	aname	...	<u>a</u>	<u>b</u>	<u>c</u>	cname	<u>b</u>	bname	...

جدول A

جدول C

جدول B

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

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

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

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

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

توجه: یک موجودیت ضعیف همیشه در ارتباطش با موجودیت قوی رابطه اجباری دارد. گزینه چهارم عبارت درستی است، بنابراین پاسخ سوال نیست. گزینه چهارم به این شکل بیان شده است.

«اگر یک نوع موجودیت (مانند E₁) تنها یک صفت داشته باشد و تنها با یک نوع موجودیت دیگر (مانند E₂) در ارتباط باشد، می‌توانیم آن را حذف و به عنوان صفت موجودیت دوم (یعنی E₂) در نظر بگیریم.» بهترین راه شناسایی موجودیت‌های یک محیط عملیاتی، بررسی مقادیری است که قرار است ذخیره شود، ذخیره‌سازی مقادیر داخل صفات (ستون‌ها) انجام می‌شود و این صفات قطعاً مرتبط با یک موجودیت هستند. بنابراین هر موجودیت باید دارای چندین صفت باشد. پس موجودیت فاقد صفت معنا ندارد. حتی در اغلب مواقع، موجودیت تک خصیصه‌ای نیز قابل قبول نمی‌باشد و به احتمال فراوان آن صفت مرتبط به یک موجودیت دیگر است که به خطاب به عنوان یک موجودیت تک خصیصه‌ای شناخته شده است.

۳۰ - گزینه (۲) صحیح است.

صورت سوال به این شکل است:

با توجه به گزاره‌های زیر کدام مورد درست است؟

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

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

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

گزاره b نادرست است، زیرا جدول stud (nano, stno, avg) 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). همچنان اشتراک این دو برابر مجموعه تهی است که نمی‌تواند نه ابرکلید و نه کلید کاندید باشد. به صورت زیر:

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

c) برای اتصال به پایگاه داده، در **Connection String** نحوه 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 حداقل هم یک ابرکلید و یک کلید کاندید دارد.

- ۳۱ - گزینه (۳) صحیح است.

صورت سوال به این شکل است:

با توجه به گزاره‌های زیر، کدام مورد درست است؟

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

گزاره a درست است، زیرا هر جدولی حتماً حداقل یک کلید کاندید دارد. و در حالت حداقل 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 $n = 2k = 2$ آنگاه $k = 1$

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

مثال: اگر $n = 2k = 4$ آنگاه $k = 2$

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

مثال: اگر $n = 2k = 6$ آنگاه $k = 3$

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

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

است. برای مثال حالت $\binom{6}{4}$ قرینهٔ حالت $\binom{6}{1}$ است. و حالت $\binom{6}{5}$ قرینهٔ حالت $\binom{6}{2}$ است. آخر $\binom{6}{6}$ که قرینه ندارد. هر کدام از ترکیب‌ها مثل $\binom{6}{1}$ یا $\binom{6}{2}$ یا $\binom{6}{3}$ 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 حداقل $\binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{n}$ است.

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

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

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

گزاره c نادرست است، زیرا در جدول تمام کلید، یک جدول فقط و فقط یک ابرکلید دارد و فقط 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 حداقل 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 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. 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 کلامی
- صفت ساده چندمقداری: مانند مدرک تحصیلی
- صفت مرکب تک‌مقداری: مانند تاریخ تولد
- صفت مرکب چندمقداری: مانند آدرس

۳۲- گزینه (۱) صحیح است.

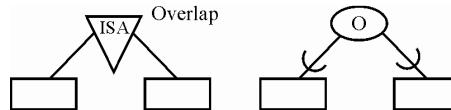
نگاشت رابطه ISA یا وراثت به مدل رابطه‌ای

در رابطه ISA رابطه پدر با فرزندان به دو صورت رابطه اختیاری یا جزئی یا پخشی (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 پوشای تخصیص غیرمجزا (Overlap) و رابطه منفصل یا غیرپوشای تخصیص مجزا (Disjoint) می‌باشد.

توجه: در یک رابطه اجباری یا کلی (Total)، هر نمونه از موجودیت پدر حتماً می‌باشد 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.

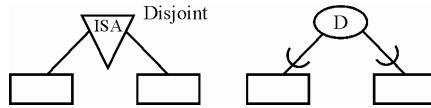
رابطه متصل یا پوشای تخصیص غیرمجزا (Overlap) مابین فرزندان و پدر به دو شیوه زیر نشان داده می‌شود:



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

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

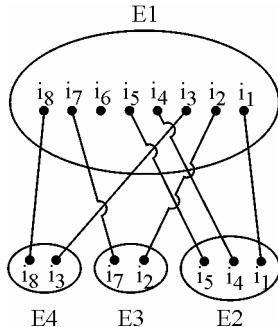
رابطه منفصل یا غیرپوشای تخصیص مجزا (Disjoint) مابین فرزندان و پدر به دو شیوه زیر نشان داده می‌شود:



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

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

صورت سوال به این شکل است:



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

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

بطور کلی خواص رابطه به سه شکل زیر وجود دارد:

الف) درجه ارتباط

ب) کاردینالیتی ارتباط

ج) اجباری و اختیاری بودن ارتباط

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

الف) درجه ارتباط

به تعداد موجودیت‌هایی که در یک رابطه مشارکت دارند، درجه ارتباط گفته می‌شود. درجه در مدل ER عددی صحیح و کوچکتر از ۵ است. ارتباط‌های درجه ۱، ۲ و ۳ معمول، ارتباط درجه ۴ کمیاب و غیر معمول است و ارتباط بالاتر از درجه ۴ قابل رسم نیست.

ب) کاردینالیتی ارتباط

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

ج) اجباری و اختیاری بودن ارتباط

این نوع رابطه به دو دسته کلی زیر تقسیم می‌شود:

۱- اجباری یا کلی (Total)

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

۲- اختیاری یا جزئی (Partial)

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

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

همانطور که در شکل صورت سوال واضح و مشخص است، نمونه موجودیت 6^j در موجودیت E1 یعنی پدر با هیچ یک از نمونه موجودیت‌های فرزند یعنی موجودیت‌های E2، E3 و E4 وارد رابطه نشده است. و از آنجاکه اگر و تنها اگر حداقل یکی از نمونه‌های موجودیت پدر در رابطه با فرزندان شرکت نکرده باشد، آن رابطه اختیاری است، بنابراین نمونه‌های موجودیت E1 به صورت اختیاری وارد رابطه با فرزندان یعنی موجودیت‌های E2، E3 و E4 شده‌اند. بنابراین گزینه‌های دوم و سوم پاسخ سوال نیستند.

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

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

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

توجه: در یک رابطه پوشای 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

فرزنдан.

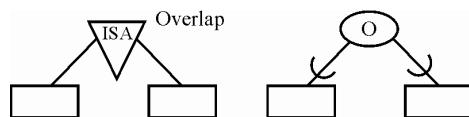
توجه: سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه اول را به عنوان پاسخ اعلام کرده بود.

۳۳- گزینه (۳) صحیح است.

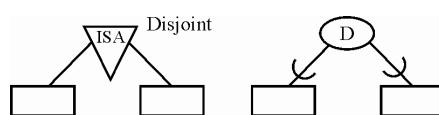
نگاشت رابطه ISA یا وراثت به مدل رابطه‌ای

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

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

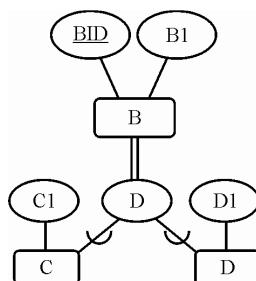


توجه: در رابطه متصل یا پوشای تخصیص غیرمجزا (Overlap) ارتباط پدر با فرزندان می‌تواند یک به یک باشد و همچنین می‌تواند یک به چند باشد. همچنین اشتراک نمونه موجودیت‌ها میان موجودیت‌های فرزند با یک نمونه موجودیت از پدر می‌تواند تهی باشد و همچنین می‌تواند غیرتهی باشد. رابطه منفصل یا غیرپوشای تخصیص مجزا (Disjoint) مابین فرزندان و پدر به دو شیوه زیر نشان داده می‌شود:



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

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



مدل EER رسم شده در شکل فوق، یک رابطه اجباری یا کلی یا کامل (Total) مابین موجودیت B و

موجودیت‌های C و D و یک رابطه منفصل یا غیرپوشای تخصیص مجزا (Disjoint) را مابین موجودیت‌های C و D و موجودیت B نشان می‌دهد. که در ادامه فرآیند نگاشت آن به مدل رابطه‌ای را بیان می‌کیم. در یک رابطه اجباری یا کلی یا کامل (Total)، هر نمونه از موجودیت پدر حتماً می‌باشد با یکی از نمونه موجودیت‌های فرزند در ارتباط باشد. برای نمونه در این مثال، هر نمونه از موجودیت B حتماً می‌باشد با یکی از نمونه موجودیت‌های C یا D در ارتباط باشد. به عبارت دیگر نمی‌توان نمونه‌ای از موجودیت B داشت که با هیچ یک از نمونه موجودیت‌های C یا D در ارتباط نیست.

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

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

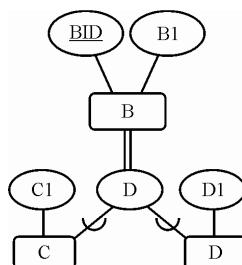
همچنین نمونه موجودیت‌های نوع ماشین که دو محور (2WD) یا چهارمحور (4WD) است نمی‌تواند به طور همزمان با نمونه‌ای از موجودیت ماشین که شماره شاسی یکتا و منحصر به فرد خود را دارد در ارتباط باشند، به عبارت دیگر نمی‌توان نمونه‌ای از موجودیت‌های نوع ماشین که دو محور (2WD) یا چهارمحور (4WD) است داشت که به طور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

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

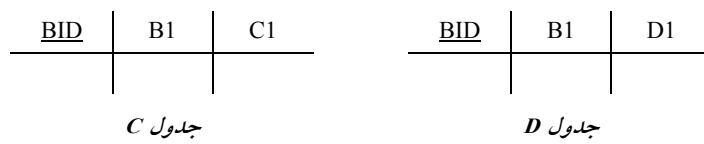
در حالت رابطه اجباری مابین موجودیت B و موجودیت‌های C و D به ازای هر نمونه از موجودیت B، حتماً نمونه موجودیتی از C یا D وجود دارد که با رابطه برقرار کند، پس در این حالت طراحی بهینه این است که کل صفات موجودیت B در دو جدول موجودیت‌های C و D قرار داده شود و یک طراحی به شکل مدل دو جدولی ایجاد گردد، همچنین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

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

مدل تحلیل (نمودار ISA)



مدل طراحی (مدل رابطه‌ای)



۳۴ - گزینه (۴) صحیح است.

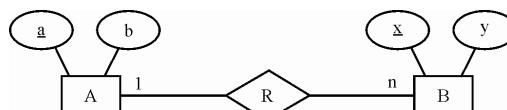
به طور کلی در مدل رابطه‌ای، هر موجودیت شناسایی شده در نمودار ER (مدل تحلیل) هنگام نگاشت به مدل رابطه‌ای (مدل طراحی) به یک جدول تبدیل می‌شود. همچنین صفت‌های موجودیت پس از نگاشت آن در مدل رابطه‌ای به صورت ستون‌های جدول بیان می‌شوند. همچنین ارتباط بین جدول از طریق کلید خارجی برقرار می‌گردد.

نگاشت رابطه یک به چند بین دو موجودیت به مدل رابطه‌ای

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

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



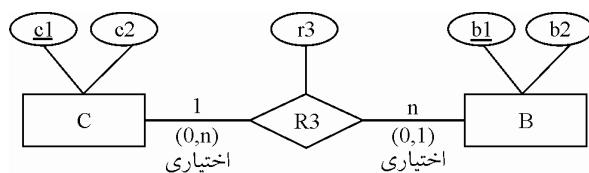
مدل طراحی:

$\underline{a} \mid b$ 	$\underline{x} \mid y \mid a$ 
A جدول	B جدول

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.

تو حه: نماد خط افقی نشانه اختباری یو دن موجودت چسیده 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:

<u>c1</u>	c2	<u>b1</u>	b2	c1	r3
<i>جدول C</i>			<i>جدول B</i>		

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

توجه: همچنین صفات متصل (r3) به رابطه، درون جدول چند یعنی موجودیت B مستر می شود.

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

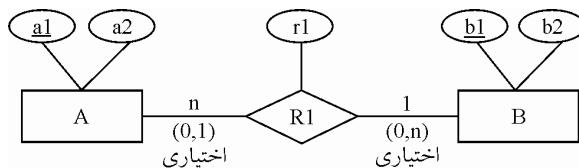
توجه: کلید کاندید جدول چند یعنی موجودیت B برابر همان کلید کاندید سابق در جدول موجودیت B است. یعنی کلید کاندید جدول B برابر (b1) است.

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

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

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

مدل طراحی:

<u><u>a1</u></u>	a2	b1	r1	<u><u>b1</u></u>	b2	c1	r3

جدول A

جدول B

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

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

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

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

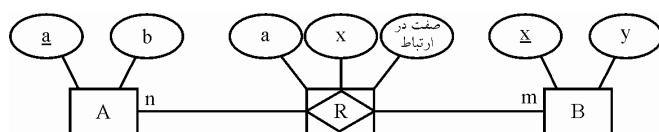
توجه: جدول B در نگاشت مرحله قبل ایجاد شده است که در اینجا دقیقاً به همان شکل و همان مشخصات، استفاده شده است.

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

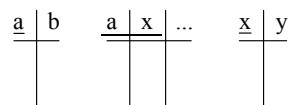
نگاشت رابطه چند به چند بین دو موجودیت به مدل رابطه‌ای مستقل از اختیاری یا اجباری بودن موجودیت‌ها، هر موجودیت به یک جدول تبدیل می‌گردد و یک جدول پل (Bridge) نیز به عنوان ارتباط دهنده دو جدول مورد استفاده قرار می‌گیرد. همچنین کلید کاندید جدول پل از ترکیب کلید کاندید دو جدول دیگر ایجاد می‌گردد. همچنین صفات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

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



مدل طراحی:

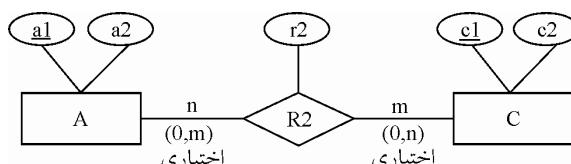


جدول A جدول AB جدول B

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

حالت اختیاری

مدل تحلیل:



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

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

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

مدل طراحی:

<u>a1</u>	a2	b1	r1	<u>a1</u>	<u>c1</u>	r2	<u>c1</u>	c2
			جدول			جدول		جدول

توجه: ستون a1 در جدول 2 به عنوان کلید خارجی تعریف می‌گردد که به جدول A ارجاع می‌کند.

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

توجه: همچنین صفات متصل (r2) به رابطه، درون جدول پل یعنی جدول 2 مستتر می شود.

توجه: همچنین کلید کاندید جدول پُل از ترکیب کلید کاندید دو جدول دیگر ایجاد می‌گردد. یعنی کلید کاندید جدول $t2$ برابر $(a1, c1)$ است.

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

موجودیت A است. یعنی کلید کاندید جدول A برابر (a1) است.

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

موجودیت C است. یعنی کلید کاندید جدول C برابر (c1) است.

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

مشخصات، استفاده شده است.

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

مشخصات، استفاده شده است.

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

۳۵ - گزینه (۱) صحیح است.

به طور کلی در مدل رابطه‌ای، هر موجودیت شناسایی شده در نمودار ER (مدل تحلیل) هنگام 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 وجودیتی شروع کنید که ورودی صفت از موجودیت دیگری نداشته باشد.

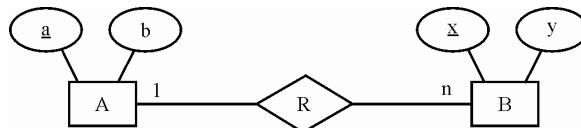
نگاشت رابطه یک به چند بین دو موجودیت به مدل رابطه‌ای

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

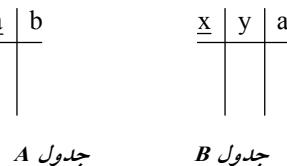
کاندید جدول یک در جدول چند به عنوان کلید خارجی تعریف می‌گردد. همچنین صفات متصل به رابطه، درون جدول چند مستتر می‌شود.

روال کلی نگاشت در این حالت به صورت زیر است:

مدل تحلیل:



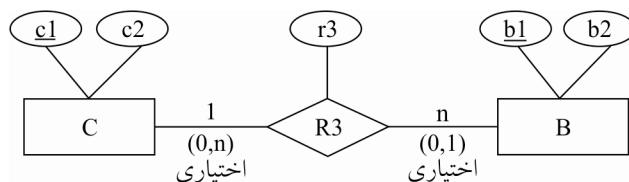
مدل طراحی:



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:

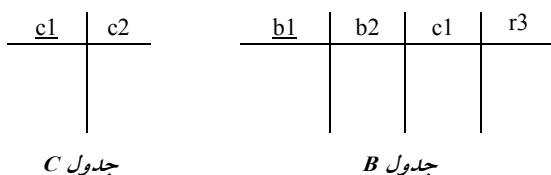


توجه: در شکل فوق صفت c_1 کلید موجودیت C و صفت b_1 کلید موجودیت B است.

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 حداقل با N نمونه موجودیت از B ارتباط دارد و قید $(0,1)$ نشان می‌دهد که هر نمونه موجودیت از B حداقل با صفر و حداقل 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 (g3) به رابطه، درون جدول چند یعنی موجودیت B مستتر می شود.

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.

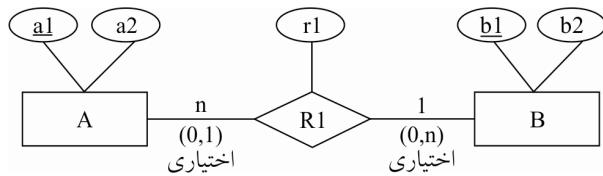
توجه: همانطور که واضح است، طراحی جدول B در گزینه اول به صورت (b_1, b_2, c_1, r_3) در نظر گرفته شده است، که مطابق آنچه بیان کردیم، طراحی درستی است. بنابراین گزینه‌های دوم، سوم و چهارم را کنار می‌گذاریم، پس تا همینجا پر واضح است که گزینه اول پاسخ سوال است.

توجه: دقت کنید که در گزینه دوم طراحی جدول B به صورت (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:

<u>a1</u>	a2	b1	r1

A جدول

<u>b1</u>	b2	c1	r3

B جدول

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

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 پُر واضح است که گزینه اول پاسخ سوال است. به همین سادگی.

توجه: دقت کنید که در گزینه دوم طراحی جدول C به صورت C(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.

۳۶- گزینه (۴) صحیح است.

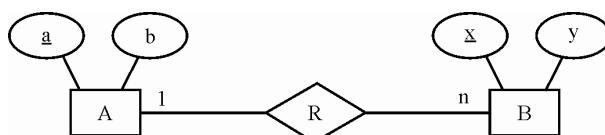
به طور کلی در مدل رابطه‌ای، هر موجودیت شناسایی شده در نمودار ER (مدل تحلیل) هنگام نگاشت به مدل رابطه‌ای (مدل طراحی) به یک جدول تبدیل می‌شود. همچنین صفت‌های موجودیت پس از نگاشت آن در مدل رابطه‌ای به صورت ستون‌های جدول بیان می‌شوند. همچنین ارتباط بین جدول از طریق کلید خارجی برقرار می‌گردد.

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 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:

$\begin{array}{c c} a & b \\ \hline & \end{array}$	$\begin{array}{c c c} x & y & a \\ \hline & & \end{array}$
جدول <i>A</i>	جدول <i>B</i>

وابستگی وجودی

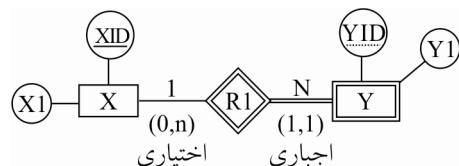
اگر در یک بانک اطلاعاتی، وجود یک موجودیت، وابسته به موجودیت دیگری باشد که در صورت حذف و تغییر موجودیت اصلی یعنی موجودیت قوی این موجودیت نیز تغییر کند، این نوع وابستگی را وابستگی وجودی گفته و به پدیده وابسته، موجودیت ضعیف گویند. همچنین موجودیت ضعیف کلید موجودیت قوی را در بر دارد تا هرگونه تغییر یا حذف 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 حداقل با *N* نمونه موجودیت از *Y* ارتباط دارد و قید (1,1) نشان می‌دهد که هر نمونه موجودیت از *Y* حداقل با یک و حداقل با یک نمونه موجودیت از *X* ارتباط دارد.

مدل طراحی:

<u>XID</u>	X1	<u>XID</u>	YID	Y1	
جدول X			جدول Y		
X			Y		

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

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

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

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

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

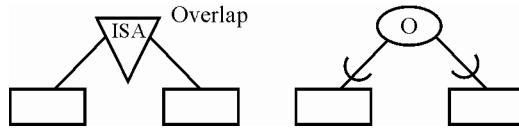
نگاشت رابطه ISA یا وراثت به مدل رابطه‌ای

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 پوشای تخصیص غیرمجزا (Overlap) و رابطه منفصل یا غیرپوشای تخصیص مجزا (Disjoint) می‌باشد.

توجه: در یک رابطه اجباری یا کلی (Total)، هر نمونه از موجودیت پدر حتماً می‌باشد 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 پوشای تخصیص غیرمجزا (Overlap) مابین فرزندان و پدر به دو شیوه زیر نشان داده می‌شود:

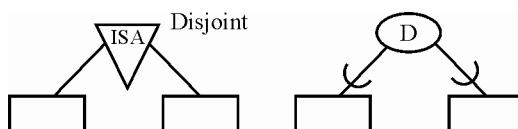


توجه: در رابطه متصل یا پوشای تخصیص غیرمجزا (Overlap) ارتباط پدر با فرزندان می‌تواند یک به یک باشد و همچنین می‌تواند یک به چند باشد. و همچنین اشتراک نمونه موجودیت‌ها میان موجودیت‌های فرزند با یک نمونه موجودیت از پدر می‌تواند تهی باشد و همچنین می‌تواند غیرتهی باشد.

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

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



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

توجه: در یک رابطه منفصل یا غیرپوشای 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 غیرپوشای 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 کامل (Total)، هر نمونه از موجودیت پدر حتماً می‌باشد با یکی از نمونه موجودیت‌های فرزند در ارتباط باشد. برای مثال در این سؤال، هر نمونه از موجودیت Y حتماً می‌باشد 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.

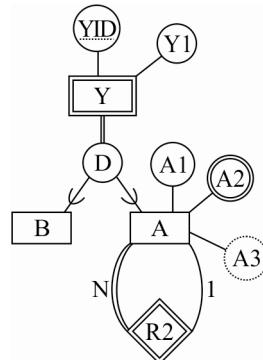
همچنین در یک رابطه منفصل یا غیرپوشای تخصیص مجزا (Disjoint)، نمونه موجودیت‌های فرزند نمی‌توانند به طور همزمان با نمونه‌ای از موجودیت پدر در ارتباط باشند. برای مثال در این سؤال، نمونه موجودیت‌های A و B نمی‌توانند به طور همزمان با نمونه‌ای از موجودیت Y در ارتباط باشند. به عبارت دیگر نمی‌توان نمونه‌هایی از موجودیت‌های A و B داشت که به طور همزمان با نمونه‌ای از موجودیت Y در ارتباط هستند. به بیان دیگر همانطور که گفته شد، در رابطه منفصل یا غیرپوشای 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 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 لحاظ 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 آنچاکه رابطه مابین موجودیت‌های A و B و موجودیت Y یک رابطه منفصل یا غیرپوشای 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 غیرپوشای (Disjoint)، کل صفات باشو 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)

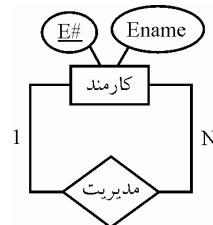
همانطور که در مدل طراحی قبل‌تر گفتیم، مدل طراحی جدول Y به صورت (XID,YID,Y1) در نظر گرفته شد.

XID	YID	Y1	...	XID	YID	Y1	A1	A2
جدول B				جدول غیرنرم 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 عمل الحاق، 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 حداقل یک رئیس دارد.

مدل تحلیل:



مدل طراحی:

E#	Sname	M#
S ₁	Sn1	NULL
S ₂	Sn2	S ₁
S ₃	Sn3	S ₁
S ₄	Sn4	S ₃
S ₅	Sn5	S ₂
S ₆	Sn6	S ₅

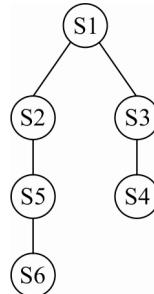
کلید خارجی (شماره مدیر)
کلید کاندید (شماره کارمند)
طوقه

توجه: سطر مریوط به مدیر کل، که مدیر بالاتری ندارد، مقدار M# در فیلد NULL 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 مبدأ و مقصد کلید خارجی با هم یکی است.

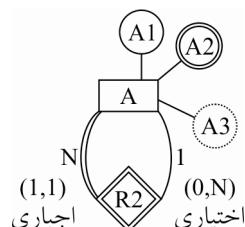
توجه: مدل ریاضی نگاشت رابطه یک به چند در یک موجودیت به مدل رابطه‌ای یک ساختار درختی است، شامل گره ریشه (مدیر کل)، پدر(ناظر) و فرزندان (زیردست).



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.

حالات 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 قبل تر هم گفته‌یم کلید کاندید موجودیت A صفات (XID,YID) است.

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

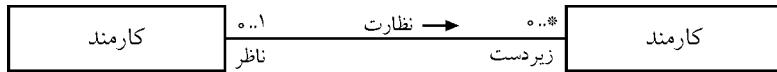
توجه: نماد خط افقی نشانه اختیاری بودن موجودیت چسییده به آن است، اما نماد دایره کوچک توخالی به معنی صفر و عدم الزام شرکت در رابطه نشانه اختیاری بودن موجودیت طرف مقابل است.

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 (بارتابی) 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:

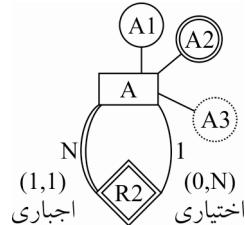


شکل فوق گویای رابطه خودانجمنی (Self Association) است. به این معنی که یک کارمند هیچ زیردستی (فرزنده) ندارد یعنی برگ درخت است (چون مدیر نیست) یا چندین زیردست (فرزنده) دارد (چون مدیر است) همچنین یک کارمند هیچ ناظری (پدر) ندارد (چون مدیر کل است) یا حداکثر یک ناظر (پدر) دارد (چون مدیر کل نیست).

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

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

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



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

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

صفت تک مقداری و چند مقداری

بعضی از صفات چه ساده و چه مرکب فقط می‌توانند یک مقدار را بگیرند که به این صفات، صفت تک مقداری 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 کدلی

صفت ساده چندمقداری: مانند مدرک تحصیلی

صفت مرکب تک‌مقداری: مانند تاریخ تولد

صفت مرکب چندمقداری: مانند آدرس

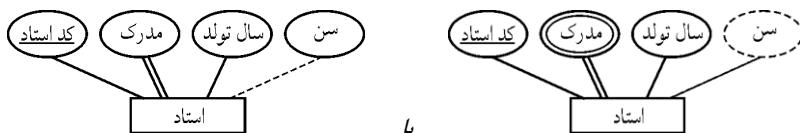
توجه: در بانک اطلاعاتی مبتنی بر مدل رابطه‌ای (جدولی) صفت چندمقداری نداریم.

صفت مشتق (پویا)

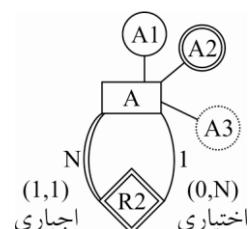
صفتی است که در موجودیت وجود خارجی ندارد ولی در صورت لزوم می‌توان آنرا بدست آورد. صفتی که مقادیر آن مدام در حال تغییر و تحول باشد، صفت پویا یا مشتق محسوب می‌گردد. بنابراین به دلیل تغییرات مداوم، توصیه می‌گردد صفت پویا در جداول بانک اطلاعات مورد استفاده قرار نگیرد و مقدار آن از طریق صفت مرتبط با آن محاسبه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



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

مدل طراحی غیرنرم‌الوایع

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

توجه: همانطور که گفته شد کلید خارجی یک رابطه می‌تواند متناظر با کلید کاندید همان رابطه باشد. در واقع دیاگرام ارجاعی دارای طوره است به این معنی که مبداء و مقصد کلید خارجی با هم یکی است. همانطور که گفته شد 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 بپرورد همین XIDR2 نوشته می‌شد که احتملاً خطای حروفچینی سوال بوده است ولی مهم هم نیست چون مهمه نام عوض شود که عوض شده است.

توجه: همانطور که گفته شد صفت مشتق (پیو) 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 گزینه چهارم هم صفت مشتق A3 نیامده است.

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

نرمال فرم اول

- دارای حداقل یک کلید کاندید باشد.

- 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:

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

مدل طراحی نرمال:

XID	YID	A2	XID	YID	Y1	A1	XIDR	YIDR2
AA2								

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 الحق پذیر هم هستن و توسط عمل الحق 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 پُرواضح است که گزینه چهارم پاسخ سوال است. به همین سادگی.

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

۳۷- گزینه (۳) صحیح است.

صورت سوال به این شکل است:

چه تعداد از گزاره‌های داده شده درست است؟

گزاره اول: تعداد کلیدهای کاندید یک رابطه از تعداد سوپرکلیدهای آن رابطه همواره کمتر است.
گزاره اول نادرست است، زیرا در حالت کلی، اگر رابطه R، دارای n خصیصه باشد، آنگاه تعداد ابرکلیدهای آن حداقل یک و حداقل $2^n - 1$ 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 حداقل 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 حداقل $1 - 2^n$ ابرکلید، حالت خاص جدول تمام کلید هم پوشش می‌دهد چون بیان حداقل $1 - 2^n$ 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 $1 - 2^n$ 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 حداقل تعداد ابرکلیدهایی که می‌توانند تجربه کند برابر $\binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n}$ یا $1 - 2^n$ است. در واقع عبارت زیر برقرار است:

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

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

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

مثال:

S#	Sname	City	S#	P#	QTY	P#	Pname	Color
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
<i>S</i> جدول			<i>SP</i> جدول			<i>P</i> جدول		

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

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

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

مثال:

شماره ملی: ابرکلید است. کلید کاندید نیز هست.

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

توجه: یک جدول می‌تواند چندین کلید کاندید داشته باشد.

مثال:

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

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

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

مثال: یک جدول تمام کلید.

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

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

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

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

مثال:

S#	Sname	City	S#	P#	QTY	P#	Pname	Color
S ₁	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 #، به تنها ی 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 # در جدول S ارجاع می‌کند.

P#: در جدول SP کلید خارجی است و به ستون # 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 کافیست بگوید چه چیزی لازم دارد تا سیستم برای او ایجاد (مثل جداول) یا استخراج (مثل پرس و جوهای) کند. در واقع چگونگی ایجاد جداول یا استخراج پرس‌وجوها از دید کاربر برنامه‌ساز و کاربر نهایی مخفی است.

استقلال داده‌ای

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

۱- استقلال فیزیکی داده‌ها

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

۲- استقلال منطقی داده‌ها

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

برای مثال مدل رابطه‌ای از تحریدی 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 می‌آمد. از آنجاکه برنامه‌های کاربردی براساس مدل طراحی بانک (ادراسی خاص یا طراحی منطقی) تعریف می‌شوند، بنابراین به طور بالقوه در معرض تأثیرپذیری از تغییرات در مدل طراحی بانک (ادراسی خاص یا طراحی منطقی) قرار دارند.

توجه: در سیستم‌های امروزی، این نوع استقلال هم تا حدی (و نه صدرصد) تأمین شده است.

أنواع تغيير در مدل طراحى منطقى يا ادراسى 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 **هيچگاه استقلال منطقى 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 Index و Create View و Create Table و دیگر دستورات آن ایجاد و مدیریت می‌گردد. و مقادیر وجه داده توسط دستورات Insert و DML نظیر 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 **فيزيكى** 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 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. دقت کنید که Table بخشی از وجه داده است. در واقع بخش داده از بخش‌های مختلف View و Index تشکیل شده است.

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

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

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

جدول S با مقادیر زیر را در نظر بگیرید:

S#	Sname
S1	Sn1
S2	Sn2

جدول S

پرس و جوی زیر را در نظر بگیرید:

```
SELECT *
FROM S
```

خروجی پرس و جو به صورت زیر است:

S#	Sname
S1	Sn1
S2	Sn2

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

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

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

```
SELECT *
FROM SV
```

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

SID	SN
S1	Sn1
S2	Sn2

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

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

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

S#	Sname	City
S1	Sn1	NULL
S2	Sn2	NULL

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

```
SELECT *
FROM VS
```

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

SID	SN
S1	Sn1
S2	Sn2

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

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

گزاره چهارم: مدیریت تراکنش‌های همرونده 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 اندختن و ساقط کردن آن آزادی عمل دارد. هدف اصلی از اینگونه کنترل‌ها، حذف‌ها و تعویق‌ها، حفظ جامعیت داخلی و خارجی بانک اطلاعات است.

تضمين جامعيت داخلی و خارجي 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 تضمين شود.

این کنترل‌ها به خواص ACID معروفند:

۱- یکپارچگی یا تجزیه‌ناپذیری (Atomicity)

این خاصیت به همه یا هیچ موسوم است. م perpetrator 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 بدھکار 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 موقعیّتی 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. بدینهی است که در این حالت باید پول برداشت شده دوباره به همان حساب اول بازگردانده شود تا جامعیت بانک اطلاعات حفظ شود. این عمل معادل این است که بگوییم هیچ دستورالعملی از تراکنش انجام نشده است.

به عنوان مثالی دیگر، هنگام خرید اینترنتی با کارت عضو شتاب ممکن است پول از حساب شما کسر گردد، اما به حساب فروشگاه مورد نظر واریز نگردد، بنابراین خرید شما ناموفق اعلام می‌گردد، که در این حالت پول حداقل 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 محقق‌کردن خاصیت یکپارچگی (Atomicity) هر تراکنش می‌بایست بین دو دستور تراکنش تعریف و اجرا می‌شود. پس تراکنش شامل دو یا چند دستور SQL است. بر این اساس می‌توان گفت که هیچ پرس و جوابی برای پایگاه داده هویت مستقل ندارد، بلکه DBMS فقط تراکنش‌ها را می‌شناسد و اجرا می‌کند. به این ترتیب باید گفت که هر پرس و جوابی 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 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 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 طبقه‌ی 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 ۰ تا ۲۰ برای نمرات، تعریف بازه ۰ 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:

Foreing key (S#) References S(S#)

```
on delete cascade  
on update cascadec
```

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

Foreing key (P#) References P(P#)

```
on delete cascade  
on update cascadec
```

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 حالت سازگار به حالت سازگار دیگری متغیر 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 تسريع عملکرد برنامه‌ها را در پی دارد. بسیاری از سیستم‌ها اجازه می‌دهند که چندین کاربر به صورت همرونده داده‌ها دسترسی یابند و تغییرات مورد نظر خود را بر روی آنها اعمال نمایند. در چنین محیط‌هایی تغییرات همرونند ایجاد شده بر روی داده ممکن است منجر به ایجاد ناسازگاری در داده‌ها گردد. در سیستم‌های بانکی کاربران مختلف می‌توانند به صورت همزمان با یکدیگر 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 همرونند 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 رکورд 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 مبالغ برگ چک A و B به ترتیب مبالغ 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!

نام خانوادگی	مواردی	نام	شماره حساب	شماره مشتری
	500	Ali	0313	6037

Read (A)	Read (B)
$A \leftarrow 500$	$B \leftarrow 500$
$R=500-100$	$R=500-50$
$R \leftarrow 400$	$R \leftarrow 450$
Write (R) $\leftarrow 400$	Write (R) $\leftarrow 450$

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

در بانک اطلاعات ممکن است تراکنش‌های همرونده وجود داشته باشد (مثل سیستم‌های چند برنامه‌ای) بر طبق خاصیت انزوا همرونده تراکنش‌ها باید کترل شود تا اثر مخرب بر روی هم نداشته باشند به بیان دیگر اثر تراکنش‌های همرونده 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 مجرای A و B، تراکش 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.

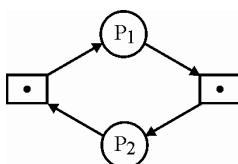
توجه: کنترل همروندي توسط بخشی از DBMS بنام واحد کنترل همروندي (concurrency control) انجام می شود.

روش قفل گذاري

يکي از روش هاي اعمال خاصيت جداسازی در تراکشن ها و جلوگيري از اثر مخرب تراکشن هاي همروندي بر روی يكديگر، روش قفل گذاري است. در اين روش هنگامي که تراکشن به داده اى نياز داشته باشد. تقاضاي قفل کردن آن را می دهد و در اين حالت بقیه تراکشن ها تا اتمام کار آن نمي توانند از آن استفاده کنند. در واقع استفاده از انواع قفل ها در مکانيسم قفل گذاري، روش هاي دسترسی به فيلدها و يا داده هاي اشتراکي را در كاربردهاي ديگر، امكان پذير می سازد. وجود قفل ها سبب می شود که در اجرای همروندي چند تراکشن، يك رکورد مشترک (عامل مشترک) به صورت همزمان توسيط دو تراکشن مورد استفاده قرار نگيرد. اين کار سرعت عملیات را افزایش می دهد زيرا داده اشتراکي به صورت انحصاری فقط در اختیار يك تراکشن است. ولی احتمال بروز بنيست را به دليل برقراری شرط انحصار متقابل زياد می کند.

بر اساس قاعده کافمن شرایط وقوع بنيست به صورت زير است:

- ۱- انحصار متقابل
- ۲- انحصاری بودن
- ۳- نگهداري و انتظار
- ۴- انتظار چرخشي



۴- پايائي يا ماندگاري (Durability)

براساس اين خاصيت تراکشن هايي که به مرحله انجام (commit) برسند اثريان ماندني است و هرگز به طور تصادفي از بين نمي روند. برای مثال اگر مبلغی به حسابي واريز شود و تراکشن مربوطه انجام یافته (commit) اعلام شود حتی در صورت وقوع حادثه در آن شعبه بانک، مشتری نباید متضرر شود، برای مثال عمل واريز قبل از اعلام انجام موفق (commit) باید در جای ديگر نيز ثبت شده باشد، مثل ديسک اصلی يا ديسک پشتيبان. يعني تأثيرات تراکشن در پايگاه داده، ماندگار باشد. هنگامي که يك تراکشن دستور commit را اجرا می کند نتایج اجرای آن به نسخه اصلی پايگاه داده در ديسک منتقل می شود. بنابراین می توان گفت تا زمانی که يك تراکشن دستور commit را اجرا نکرده است یا به اصطلاح تثبيت نشده است، و پيچگي ماندگاري در مورد آن تراکشن تضمین شده نیست. اما پس از اجرای دستور ماندگاري نتایج تراکشن تضمین می شود.

توجه: دو خاصيت يکپارچگي و پايائي توسط واحدی از DBMS به نام واحد مدیریت بازگرد (Recovery management) کنترل می گردد.

توجه: سازمان سنجش آموزش کشور، در کليد اوليه خود، گزينه چهارم را به عنوان پاسخ اعلام نمود، اما در کليد نهايی گزينه سوم را به عنوان پاسخ نهايی اعلام نمود، که کار درستی بوده است.

تست‌های فصل چهارم: جبر رابطه‌ای

۱- اگر ستون FK کلید خارجی در جدول a باشد، که به جدول b ارجاع می‌دهد، کدام گزینه معادل

(کارشناسی ارشد- دولتی ۸۱۶)

$$\prod_{a.}^* \left(\theta \wedge a.FK = b.FK \right) \quad \sigma(a \times b)$$

$$\prod_{b.*} (\sigma_\theta(a \times b)) \quad (4)$$

$$\prod_{a.*} (a \times b) \quad (3)$$

$$\sigma_\theta(a \times b) \quad (2)$$

$$\sigma_\theta(a \times b) \quad (1)$$

۲- رابطه حرارت (کد ناحیه، نام، تاریخ، حرارت بالا، حرارت پایین) برای ثبت درجه حرارت‌های بیشینه و کمینه در نواحی مختلف در زمان‌های متفاوت به کار می‌رود. کلید این رابطه < کد ناحیه، تاریخ < می‌باشد. اگر روابط جبر رابطه‌ای زیر را در نظر بگیرید:

$$R_1 \quad (\text{کد ناحیه، تاریخ، } H) = (\text{حرارت}) \text{ حرارت بالا، تاریخ، کد ناحیه}$$

$$R_2 \quad (\text{کد ناحیه، تاریخ، } L) = (\text{حرارت}) \text{ حرارت پایین، تاریخ، کد ناحیه}$$

$$R_3 \quad (\text{کد ناحیه}) = (\text{حرارت}) \text{ } R_1 \bowtie \text{ کد ناحیه}$$

$$\text{حرارت بالا} < H$$

$$R_4 \quad (\text{کد ناحیه}) = (\text{حرارت}) \text{ } R_2 \bowtie \text{ کد ناحیه}$$

$$\text{حرارت پایین} > L$$

آنگاه کدام گزینه نام نواحی با درجه حرارت بیشینه و کمینه را به ما می‌دهد؟ (مهندسی IT- دولتی ۸۱۶)

$$\Pi \quad \text{حرارت} \bowtie (R_3 \bowtie R_4) \quad (1)$$

$$\Pi \quad \text{حرارت} \bowtie (R_3 \cup R_4) \quad (2)$$

$$\Pi \quad (\text{حرارت}) \text{ کد ناحیه } \Pi \bowtie ((\text{حرارت}) \text{ نام } -R_3) \text{ } -R_4 \quad (3)$$

$$\Pi \quad ((\text{حرارت}) \text{ کد ناحیه } \Pi \cap R_3) - ((\text{حرارت}) \text{ کد ناحیه } \Pi \cap R_4) \quad (4)$$

(مهندسی کامپیووتر- آزاد ۷۷)

۳- دو رابطه زیر را در نظر بگیرید:

STUDENT	STU-ID	NAME
S001	AHMADI	
S002	TAVANA	
S003	DIBADJ	

COURSE	COURSE-ID	STU-ID	TITLE
CSC100	S001		PASCAL PROG.
CSC100	S002		PASCAL PROG.
CSC200	S001		COMPILER WRITING
CSC300	S004		DATA STRUCTURE

رابطه‌ای که حاصل الحق طبیعی دو رابطه بالا باشد دارای چند ستون و چند تاپل خواهد بود؟

۵ و ۴

۶ و ۴

۲ و ۳

۱ و ۳

۴- رابطه‌ای که حاصل OUTER JOIN دو رابطه سؤال قبل باشد دارای چند تاپل خواهد بود؟
(مهندسی کامپیوتر- آزاد ۷۷)

- ۴) هیچ‌کدام ۳) ۳ ۵) ۲ ۴) ۱

۵- کدام یک از گزینه‌های زیر غلط است؟
(\cup : Union, \cap : Intersection, \bowtie : semijoin, \bowtie : join)
(مهندسی کامپیوتر- دولتی ۷۵)

$$\begin{array}{ll} R \cap S = R - (R - S) & R \bowtie_f S = \sigma_f(R \cup S) \\ R \bowtie_f S = S \bowtie_f R & R \bowtie_f S \neq S \bowtie_f R \end{array}$$

۶- فرض کنید R_1 و R_2 دو رابطه باشند و $R_3 = R_1 \text{ UNION } R_2$ ، کدام یک از گزاره‌های زیر صحیح است؟
(مهندسی کامپیوتر- دولتی ۸۲)

- ۱) کلید اصلی R_3 اجتماع کلیدهای اصلی R_1 و R_2 است.
۲) کلید اصلی R_3 اجتماع تمام خصیصه‌های R_1 و R_2 است.
۳) کلید اصلی R_3 کلید اصلی R_1 یا کلید اصلی R_2 است.
۴) کلید اصلی R_3 تقاطع تمام خصیصه‌های R_1 و R_2 است.

۷- اگر A یک رابطه دارای n خصیصه (ستون) باشد، تعداد تصاویر A که تهی نیستند برابر است با:
(مهندسی کامپیوتر- دولتی ۸۲)

$$2^n - 1$$

$$n^2 - 1$$

$$n! - 1$$

$$2n - 1$$

۸- عبارت زیر در چه صورت صحیح است؟

$\sigma_p(\Pi_{a_1, a_2, \dots, a_n}(R)) = \Pi_{a_1, a_2, \dots, a_n}(\sigma_p(R))$
σ: انتخاب
Π: تصویر

- ۱) شرط p حداقل برای یک سطر رابطه R برقرار باشد.
۲) شرط p فقط ستون‌های a_1, \dots, a_n را در بر گیرد.
۳) شرط p همه ستون‌های a_1, \dots, a_n را در بر گیرد.
۴) هیچ‌کدام

۹- اگر مجموعه عنوان (Heading) در رابطه R و S با هم یکسان باشد، آنگاه پیوند طبیعی این دو رابطه معادل با کدام عمل، روی رابطه‌های R و S خواهد بود؟
(مهندسی IT- دولتی ۸۳)

- ۱) ضرب ۲) اجتماع ۳) اشتراک ۴) تفاضل

برای پاسخ به دو پرسش بعدی، جداول زیر را در نظر بگیرید:
هتل (کد هتل، نام هتل، آدرس، شماره تلفن)

رزرو اتاق (کد رزرو، کد هتل، نام، آدرس، شماره تلفن، تاریخ تولد، تاریخ ورود، مدت اقامت، تاریخ رزرو)
اتاق (کد هتل، شماره اتاق، امکانات)

وضعیت اتاق (کد هتل، شماره اتاق، کد رزرو، تاریخ ورود، تاریخ خروج)

جدول هتل دارای 35 سطر، رزرو اتاق 25000، اتاق 1500 و وضعیت اتاق R سطر دارد.
اطلاعات مشتریان در جدول رزرو اتاق ذخیره می‌شود.

۱۰- اطلاعات قبلی و قدیمی تا مدتی در پایگاه داده ذخیره می‌شود. اگر کاردینالیتی را معادل تعداد تاپل‌های حاصله تعریف کنیم آن‌گاه:

کاردینالیتی اتفاق \bowtie هتل (اتفاق. کد هتل ≠ هتل. کد هتل) برابر کدام مقدار است؟

$$1) 1500 \times R \quad 2) 25000 \quad 3) 26500 \quad 4) 51000$$

۱۱- با توجه به مساله قبلی کاردینالیتی وضعیت اتفاق \bowtie هتل (وضعیت اتفاق. کد هتل = هتل. کد هتل) برابر است با:

$$1) R \quad 2) R \times R \quad 3) 35 \times R \quad 4) 25000$$

۱۲- در بانک اطلاعاتی تولید کنندگان و قطعات دستور جبر رابطه‌ای زیر کدام خروجی را تولید می‌کند؟ (مهندنسی کامپیووتر- دولتی- ۸۵)

$$\Pi_{S\#}(\sigma_{CITY='PARIS'}(S \bowtie SP))$$

۱) شماره تمام تولید کنندگان پاریس.

۲) شماره اولین تولید کننده پاریس.

۳) شماره یکی از تولید کنندگان پاریس.

۴) شماره تولید کنندگان پاریس که قطعه‌ای را تولید می‌کنند.

۱۳- اگر رابطه $R(a,b)$ دارای تعداد r تاپل بوده و رابطه $S(a,b)$ دارای تعداد s تاپل آنگاه تعداد کمینه و بیشینه تاپل‌های $R \cup S$ برابر است با:

$$1) r+s \quad 2) \max(r,s) \quad 3) \min(r,s) \quad 4) \max(r+s, \min(r,s))$$

۱۴- کدام یک از گزینه‌های زیر درست است؟ (مهندنسی IT- دولتی- ۸۵)

$$\sigma_c \wedge d(R) \equiv \sigma_c(R) - \sigma_d(R) \quad 1) \quad \sigma_{c-d}(R) \equiv \sigma_c(\sigma_d(R))$$

$$\sigma_c(R_1 \times R_2) \equiv R_1 \bowtie_c R_2 \quad 2) \quad \sigma_{c \vee d}(R) \equiv \sigma_c(R) \cap \sigma_d(R)$$

۱۵- از دیدگاه کاد، تئوری‌سین مدل رابطه‌ای، کدام یک از عملگرهای ذیل برای سیستم‌های شبه رابطه‌ای لازم می‌باشد؟ (مهندنسی IT- آزاد- ۸۴)

$$1) Project \quad 2) Select \quad 3) Join \quad 4) هر سه مورد$$

۱۶- کدام یک از عملگرهای جبر رابطه‌ای به عنوان عملگر اولیه نیست و می‌توان آن را براساس دیگر

عملگرهای اولیه به دست آورد؟ (مهندنسی IT- آزاد- ۸۵)

$$1) Union \quad 2) Times \quad 3) Minus \quad 4) Intersect$$

۱۷- اگر A و B دو رابطه با عنوان یکسان باشند، عبارت زیر معادل با کدام گزینه است؟ (مهندنسی IT- آزاد- ۸۶)

$$(A MINUS (A MINUS B)) UNION (B MINUS (B MINUS A))$$

$$1) A INTERSECT B \quad 2) A DIVIDE B \quad 3) A JOIN B$$

$$1) A MINUS B \quad 2) Minus \quad 3) Times$$

- ۱۸- پایگاه داده‌ای دارای سه جدول S و L و G به صورت زیر است:

$S(S\#, Sname)$, $L(L\#, Lname)$, $G(S\#, L\#, grade)$

که در آن جدول S جدول دانشجو با خصیصه‌های شماره دانشجویی و نام دانشجو، جدول L جدول درس با خصیصه‌های شماره درس و نام درس و جدول G جدول نمره با خصیصه‌های شماره دانشجویی، شماره درس و نمره درس است. با توجه به این پایگاه داده به سؤال زیر پاسخ دهید.

جفت شماره دانشجویی و شماره درس‌هایی را مشخص کند که برای دانشجوی مورد نظر نمره‌ای بابت درس (مهندسی IT-آزاد ۸۶)

مورد اشاره منظور نشده است.

$$(S \text{ JOIN } G)[S\#] \text{ TIMES } (L \text{ JOIN } G)[L\#] \quad (۱)$$

$$(S \text{ TIMES } L)[S\#, L\#] \text{ MINUS } G[S\#, L\#] \quad (۲)$$

$$(((S[S\#] \text{ MINUS } (S \text{ JOIN } G)[S\#])) \text{ JOIN } G) \text{ JOIN } L[S\#, L\#] \quad (۳)$$

$$((S \text{ TIMES } L) \text{ JOIN } G)[S\#, L\#] \quad (۴)$$

- ۱۹- نمودار ER روبرو را در نظر بگیرید. با فرض آن‌که در موجودیت کارمند ۲۰۰ رکورد و در موجودیت

پروژه نیز ۵ رکورد موجود باشد، حداقل و حداکثر تعداد رکوردها در پیوند طبیعی این دو موجودیت

(مهندسی IT-دولتی ۸۵)

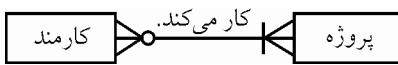
چیست؟

۱) ۰ و ۱۰۰۰

۲) ۵ و ۲۰۰

۳) ۵ و ۱۰۰۰

۴) ۲۰۰ و ۱۰۰۰



- ۲۰- اگر A و B دو رابطه دارای اسامی خصیصه‌های یکسان باشند $A \bowtie B$ معادل با کدام یک از عبارات زیر است؟

(مهندسی کامپیووترا- دولتی ۸۷)

۴) هیچ کدام

۳) $A \cup B$

۲) $A \cap B$

۱) $A \times B$

- ۲۱- اگر s و r دو رابطه باشند و S و R به ترتیب Heading این دو رابطه باشند و با فرض آن‌که $S \subseteq R$ باشد، کدام یک از عبارات‌های زیر معادل با $r \div s$ است؟

(مهندسی IT-دولتی ۸۷)

$$\Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times s)) \quad (۱)$$

$$\Pi_{R-S}(r) \cap \Pi_{R-S}((\Pi_{R-S}(r) \times s)) \quad (۲)$$

$$\Pi_{R-S}((\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r)) \quad (۳)$$

$$\Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r)) \quad (۴)$$

(مهندسی کامپیووترا- آزاد ۸۷)

با توجه به جداول زیر کدام گزینه معادل سایر گزینه‌ها نیست؟

$$\Pi_{pname, depname}(\sigma_{esp="DB"}(prof \bowtie dep)) \quad (۱)$$

$$\Pi_{pname, depname} \left[\sigma_{esp="DB"} \left(\begin{array}{c} prof \times dep \\ prof.depnum = dep.depnum \end{array} \right) \right] \quad (۲)$$

$$\Pi_{pname, depname}(\sigma_{esp="DB"}(prof \ltimes dep)) \quad (۳)$$

$$\Pi_{pname, depname}(\sigma_{esp="DB"} \wedge prof.depnum = dep.depnum(prof \times dep)) \quad (۴)$$

prof(pnum, pname, esp, depnum)
dep(depnum, depname)

۲۳- اگر T_1 و T_2 حاوی اطلاعات زیر باشند، $T_1 \bowtie T_2$ چند سطر خواهد داشت؟ (مهندسی IT-آزاد ۸۷)

T1			T2		
A	B	C	A	C	E
1	5	1	1	1	3
2	7	3	2	3	3
2	7	5	2	3	5
2	2	7	2	2	7

16 (۴) 10 (۳) 3 (۲) 4 (۱)

۲۴- با توجه به جداول زیر مطلوب است دستور جبر رابطه‌ای برای تعیین مشخصات کامل داوطلبانی که در همهی آزمون‌ها شرکت کرده‌اند؟ (مهندسی کامپیووتر- دولتی ۸۸)

داوطلب $S(S\#, Sname, address)$

آزمون $T(T\#, Tname, no_of_Ques)$

آزمون ST($S\#, T\#, Date, Time, Code$)

$$[\Pi_{S\#}, T\#(ST) \div \Pi_{T\#}(T)] \bowtie S \quad (۱)$$

$$\Pi_{S\#}, T\#(ST) \div \Pi_{T\#}, Tname(T) \quad (۴) \quad S \bowtie [\Pi_{S\#}, T\#(ST) \div \Pi_{T\#}, Tname(T)] \quad (۳)$$

۲۵- دو رابطه S و SL مفروض هستند، حاصل پرس و جوی زیر کدام است؟ (مهندسی کامپیووتر- آزاد ۸۵)

$S(S\#, sname)$ $SL(S\#, L\#, term\#, grade)$

$SL[S\#] Minus (SL where grade <> 20)[S\#]$

(۱) شماره دانشجویانی که حداقل یک نمره 20 دارند.

(۲) شماره دانشجویانی که تمام نمره‌های آنها 20 است.

(۳) شماره دانشجویانی که حداقل یک نمره 20 ندارند.

(۴) شماره دانشجویانی که حداکثر یک نمره 20 دارند.

۲۶- رابطه $R(a,b,c)$ و دو عبارت جبری زیر را در نظر بگیرید:

$Q_1 : \Pi_{b,c}(\sigma_{b=c}(R))$

$Q_2 : \Pi_{a,b}(\sigma_{a=b}(R))$

(مهندسی کامپیووتر- دولتی ۸۹)

کدام عبارت همواره تعریف شده است؟

$$(۱) Q_1 \div Q_2 \quad (۲) Q_1 \bowtie Q_2 \quad (۳) Q_1 \cap Q_2 \quad (۴) \text{موارد ۱ و ۲}$$

۲۷- اگر A و B و دو رابطه با یک قالب (Schema) باشند، آنگاه $A \bowtie B$ معادل کدام عبارت خواهد بود؟

(مهندسی IT- آزاد ۸۹)

$$A \bowtie B \quad (۱) \quad A \cap B \quad (۳) \quad A \times B \quad (۲) \quad A \times B \quad (۴)$$

(مهندسی کامپیووتر- آزاد ۸۹)

۲۸- خروجی دستور زیر چند سطر خواهد داشت؟

$$\Pi_{T1.B}(T_1 \times T_2)$$

$$T_1.B = T_2.B$$

T1		T2		
A	B	B	C	5 (۱)
a1	b1	b1	c1	2 (۲)
a2	b2	b1	c2	3 (۳)
a3	b3	b2	c2	15 (۴)
		b2	c3	
		b2	c4	

۲۹- با توجه به عبارت جبر رابطه‌ای دو سؤال query1 و query2 بر روی رابطه‌های $S(b,c)$ و $R(a,b)$ (جبر رابطه‌ای) کدام گزینه صحیح است؟ (مهندسی کامپیوترا- آزاد ۸۹ گروه ب)

$$\text{Query1: } \Pi_{a,b}(R \bowtie S)$$

$$\text{Query2: } R \cap (\rho_t(a)(\Pi_c(s)) \times \Pi_b(s))$$

(۱) پاسخ‌های Query1 زیر مجموعه پاسخ‌های Query2 است.

(۲) پاسخ‌های Query1 کاملاً متفاوت با پاسخ‌های Query2 است.

(۳) پاسخ‌های Query2 زیر مجموعه پاسخ‌های Query1 است.

(۴) هر دو پاسخ یکسانی را می‌دهند.

۳۰- سه جدول تهیه کننده $P(P\#, \text{Pname}, \text{Color}, \text{City})$ و قطعه $S(S\#, \text{Sname}, \text{Color}, \text{City})$ محموله $SP(S\#, P\#, QTY)$ را در نظر گرفته و به سؤال زیر پاسخ دهید.

کدام گزینه معادل پرس‌وجوی «اسامی همه تهیه کنندگانی را باید که حداقل یک قطعه آبی رنگ تهیه می‌کنند» می‌باشد. (مهندسی IT- آزاد ۸۹)

$$\Pi_{Sname}(\Pi_{S\#}(\Pi_{P\#}(\sigma_{Color="Blue"}(P)) \bowtie SP) \bowtie S) \quad (۱)$$

$$\Pi_{Sname}(\Pi_{S\#}(\Pi_{P\#}(P \div \sigma_{Color="Blue"}(P)) \bowtie SP) \bowtie S) \quad (۲)$$

$$\Pi_{Sname}(\Pi_{S\#}(\Pi_{P\#}(\sigma_{Color="Blue"}(P)) \bowtie SP) \div S) \quad (۳)$$

$$\Pi_{Sname}(\Pi_{S\#}(\Pi_{P\#}(P - \sigma_{Color="Blue"}(P)) \bowtie SP) \bowtie S) \quad (۴)$$

۳۱- کدام عبارت در مورد جبر رابطه‌ای نادرست است؟ (مهندسی IT- آزاد ۸۹)

(۱) اگر A و B دو رابطه باشند کلید اصلی A MINUS B، کلید اصلی A می‌شود.

(۲) در عملگر \bowtie همتایی عملوند (Same arity) الزامی است.

(۳) عملگرهای اشتراک، تقسیم و پیوند طبیعی را می‌توان به وسیله عملگرهای دیگر در جبر رابطه‌ای شبیه‌سازی کرد.

(۴) اگر رابطه A از درجه n باشد، 2^n تصویر مختلف از A وجود دارد.

۳۲- رابطه‌های $S(A,C,D)$ و $R(A,B)$ زیر مفروض‌اند: (مهندسی کامپیوترا- دولتی ۹۰)

R		S		
A	B	A	C	D
1	2	1	2	3
3	4	1	3	4
		2	4	5

R چند تاپل خواهد داشت؟ ($\Leftrightarrow S$: فرآیند طبیعی کامل - Natural full out or join)

5 (۴)

4 (۳)

3 (۲)

2 (۱)

(مهندسی کامپیووتر- آزاد ۹۰ گروه الف)

-۳۳- خروجی دستور زیر چند سطر خواهد داشت؟

 $\Pi_B(T_1 \bowtie T_2)$

T ₁		T ₂	
A	B	B	C
a ₁	b ₁	b ₁	c ₁
a ₂	b ₂	b ₁	c ₂
a ₃	b ₃	b ₂	c ₂
		b ₂	c ₃
		b ₂	c ₄

15 (۴)

3 (۳)

5 (۲)

2 (۱)

-۳۴- دو رابطه R(A,B) و S(C,D) را در نظر بگیرید. کدام گزینه هم ارزی جبر رابطه‌ای‌ها را در دو رابطه

(مهندسی کامپیووتر- آزاد ۹۰ گروه ب)

درست نشان می‌دهد؟

$$\Pi_{A,B,D}(R \bowtie S) = R \bowtie \rho_T(B,D)(S) \quad (۱)$$

$$R - \rho_T(A,B)(S) = \rho_T(A,B)(S - \rho_u(C,D)(R)) \quad (۲)$$

$$\Pi_{A,B}(R \times S) = R \quad (۳)$$

۴) هیچ کدام

-۳۵- سه جدول تهیه کننده S(S#, Sname, City) و قطعه P(P#, Pname, Color, City) و محموله

SP(S#, P#, QTY) را در نظر گرفته و به سؤال زیر پاسخ دهید. کدام گزینه پاسخ پرس‌وجوی «اسامي

همه تهیه کننده‌گانی را بیابید که هیچ قطعه آبی رنگی تهیه نمی‌کنند» می‌باشد؟ (مهندسی IT- آزاد ۹۰)

$$\Pi_{Sname}(S - \Pi_{S\#}(\Pi_{P\#}(\sigma_{Color='Blue'}(P)) \bowtie SP) \bowtie S) \quad (۱)$$

$$\Pi_{Sname}(\Pi_{S\#}(\Pi_{P\#}(P \div \sigma_{Color='Blue'}(P)) \bowtie SP) \bowtie S) \quad (۲)$$

$$\Pi_{Sname}(\Pi_{S\#}(\Pi_{P\#}(\sigma_{Color='Blue'}(P)) \bowtie SP) \div S) \quad (۳)$$

$$\Pi_{Sname}(\Pi_{S\#}(\Pi_{P\#}(\sigma_{Color='Blue'}(P)) \bowtie SP) \bowtie S) \quad (۴)$$

با توجه به جدول زیر به دو سؤال بعد پاسخ دهید.

parent (pid , ppid , sex)

جنسیت شناسه پدر یا مادر شناسه شخص

person (pid , pname , age , DOB)

تاریخ تولد سن نام شناسه شخص

-۳۶- پاسخ (شناسه پدر «آرش کمانگیر») در جبر رابطه کدام است؟ (\bowtie یعنی نیم پیوند)

(مهندسی کامپیووتر- دولتش ۹۱)

$$\Pi_{ppid}(\sigma_{sex="مرد"}(parent) \bowtie \sigma_{pname}=(person)) \quad (1)$$

$$\Pi_{ppid}(\sigma_{pname}="مرد" \bowtie \sigma_{sex}=(parent)) \quad (2)$$

$$\Pi_{ppid}(\sigma_{pname}="مرد" \bowtie \sigma_{sex}=(parent)) \quad (3)$$

$$\Pi_{ppid}(\sigma_{sex}="مرد" \bowtie \sigma_{pname}=(person)) \quad (4)$$

۳۷- به کدام گزینه در مدل رابطه‌ای نمی‌توان پاسخ داد؟
(مهندسه کامپیوتر- دولتی ۹۱)

(۱) همسر فردی به نام «آرش کمانگیر» (۲) اجداد فردی به نام «آرش کمانگیر»

(۳) افرادی که همسر آن‌ها بالای ۹۰ سال دارند. (۴) افرادی که سن آن‌ها اشتباه محاسبه شده است.

۳۸- کدام یک از گزینه‌های ذیل به ازای هر دو رابطه R_1 و R_2 و رابطه $R_3 = R_1 \text{ op } R_2$ همواره درست است؟
(مهندسه کامپیوتر- آزاد ۹۱)

(۱) اگر $\text{op} = \text{UNION}$ و $R_1 = R_2$ باشد، آن‌گاه کلید اصلی R_3 اجتماع کلیدهای R_1 و R_2 است.

(۲) اگر $\text{op} = \text{DIV}$ (تقسیم) باشد آن‌گاه تمامی صفات خاصه R_3 جزو کلیدهای اصلی هستند.

(۳) اگر $\text{op} = \text{Intersection}$ (اشتراک) باشد آن‌گاه کلید اصلی برابر کلیدهای R_1 و R_2 است.

(۴) $\text{op} = *$ (ضرب دکارتی) باشد رابطه R_3 رابطه‌ای تمام کلید است (تمامی صفات R_3 جزو کلید اصلی هستند)

۳۹- کدام یک از معادلهای ذیل در جبر رابطه‌ای نادرست است؟
(مهندسه کامپیوتر- آزاد ۹۱)

$$R_1 \text{ intersect } R_2 = R_1 \text{ Minus } (R_1 \text{ Minus } R_2) \quad (1)$$

$$R_1 \text{ intersect } R_2 = R_2 \text{ Minus } (R_2 \text{ Minus } R_1) \quad (2)$$

$$R_1 \text{ intersect } R_2 = (R_1 \text{ Union } R_2) \text{ Minus } [(R_1 \text{ Minus } R_2) \text{ Minus } (R_2 \text{ Minus } R_1)] \quad (3)$$

$$R_1 \text{ Semiminus } R_2 = R_1 \text{ Minus } (R_1 \text{ Semijoin } R_2) \quad (4)$$

۴۰- رابطه‌های:
(نام دانشجو، شماره دانشجویی STU : STUD : دانشجو)

(نام درس: CTit، کد درس: CRS (CID : دروس

(کد درس: CID، شماره دانشجویی: SC (STID : درس - دانشجو

را در نظر بگیرید. پرس‌وجوی ذیل چه چیزی را بر می‌گردد؟

(۱) جفت شماره دانشجو و دروسی که در هر جفت، دانشجو درس را انتخاب نکرده است.

(۲) جفت شماره دانشجو و دروسی که ارائه شده را می‌دهد.

(۳) جفت شماره دانشجو و دروسی که ارائه نشده را می‌دهد.

(۴) تمام جفت شماره دانشجو و دروسی را می‌دهد که تمام دانشجویان درس را انتخاب کرده‌اند.

با توجه به بانک زیر، به سؤال زیر پاسخ دهید.

Person (pID , pName , Sex , DoB)

اطلاعات فرد شامل شناسه pID، نام pName، جنسیت sex و تاریخ تولد DoB

Rel (pID1 , pID2 , pRel)

اطلاعات ارتباط بین افراد - رابطه $pRel$ بین شخص $pID1$ و $pID2$ برقرار است.
به طور خاص، $pRel=1$ به این معنی است که شخص با شناسه $pID2$ پدر شخص با شناسه $pID1$ است.

۴۱- عبارت جبر رابطه‌ای زیر معادل کدام گزینه است؟
(مهندسی کامپیوتر- دولتی ۹۶)

- $(\Pi_{pID1} (\sigma_{pRel=1} (Rel))) \bowtie_{pID1=pID} Person$
- (۱) فرزندان
(۲) پدران
(۳) فرزندان کسانی که پدر دارند.
(۴) پدران کسانی که فرزند دارند.

۴۲- کدام یک از موارد زیر بهترین تعریف برای بیان اتصال طبیعی (Natural Join) است:
(مهندسی کامپیوتر- آزاد ۹۶)

- (۱) یک عمل باینری است که از تعریف عبارت $select$ و ضرب کارتزین شکل می‌گیرد.
(۲) یک عمل باینری است که از ترکیب کامل اطلاعات دو رابطه به وجود می‌آید.
(۳) یک عمل ترکیبی است که از فقدان اطلاعات دو رابطه (Data Loss) جلوگیری می‌کند.
(۴) هیچ کدام

۴۳- کدام یک از موارد ذیل در رابطه با مقدار NULL (هیچ مقدار) نادرست است؟
(مهندسی کامپیوتر- آزاد ۹۶)

- (۱) چنان‌چه دستور $select$ با آن مواجه شود به نتیجه اضافه می‌گردد.
(۲) در دستور $projection$ با آن همانند سایر مقادیر رفتار می‌گردد.
(۳) در دستور $union$ با آن همانند سایر مقادیر رفتار می‌گردد.
(۴) همه موارد

۴۴- رابطه زیر را در نظر بگیرید:

Course (*cname*, TA)

وجود تاپل (c,t) در این جدول نشان می‌دهد که t «کمک استاد» در درس c است.
می‌خواهیم همه زوج‌های (TA1,TA2) را به دست آوریم که TA1 و TA2 از هم متمایزند و حداقل در یک درس
به عنوان کمک استاد با هم همکاری می‌کنند. می‌خواهیم هر زوج یک بار نوشته شود، به این معنی که (x,y) و
(y,x) هم‌زمان در نتیجه آورده نشوند. کدام یک از عبارات جبر رابطه‌ای زیر این درخواست را به درستی
توصیف می‌کنند؟ $b \rightarrow p_a \rightarrow p_b$ نشان دهنده تغییر نام یک صفت از a به b است.
(مهندسی کامپیوتر- دولتی ۹۶)

$$\Pi_{TA1,TA2}(\rho_{TA \rightarrow TA1}(Course) \bowtie \rho_{TA \rightarrow TA2}(Course)) \quad (1)$$

$$\Pi_{TA1,TA2}[\sigma_{TA1 <> TA2}(\rho_{TA \rightarrow TA1}(Course) \bowtie \rho_{TA \rightarrow TA2}(Course))] \quad (2)$$

$$\Pi_{TA1,TA2}[\sigma_{TA1 < TA2}(\rho_{TA \rightarrow TA1}(Course) \bowtie \rho_{TA \rightarrow TA2}(Course))] \quad (3)$$

(۴) هیچ کدام

۴۵- اگر رابطه r روی مجموعه صفت‌های X و رابطه s روی مجموعه صفت‌های Y تعریف شده باشند و اگر
نیم‌الحاق (r) semi-join با s را به صورت زیر تعریف کنیم، کدام گزینه زیر صحیح است؟
(مهندسی IT- دولتی ۹۶)

$$r \bowtie s := \Pi_X(r \bowtie s)$$

$$I : r \bowtie s := r \bowtie \Pi_{X \cap Y}(S)$$

$$II : r \bowtie s := (r \bowtie s) \bowtie s$$

$$III : r \bowtie s := ((r \bowtie s) \bowtie (s \bowtie r))$$

۱) تنها گزاره II صحیح است.

۲) گزاره های I و III صحیح هستند.

۳) هر سه گزاره صحیح است.

۴۶- کدام یک از روابط زیر در جبر رابطه ای صحیح نیست؟ (r و s رابطه هستند). (مهندسى IT- دولتى ۹۳)

$$r \times s = s \times r \quad (1)$$

$$r \bowtie \text{condition}^s = s \bowtie \text{condition}^r \quad (2)$$

$$\Pi \text{list1}(\Pi \text{list2}r) = \Pi \text{list2}(\Pi \text{list1}r) \quad (3)$$

$$\sigma_{\text{condition1}}(\sigma_{\text{condition2}} r) = \sigma_{\text{condition2}}(\sigma_{\text{condition1}} r) \quad (4)$$

۴۷- با داشتن رابطه های products و orderDetails که مربوط به کالاها و اقلام سفارش داده شده مشتریان است و هم چنین دو رابطه r و s کدام گزینه می تواند شناسه سفارش (ID) مشتریانی را بدهد که هیچ کالایی در گروه food نباشد که نخربده باشند؟ (مهندسى IT- دولتى ۹۳)

Products(Prod_ID, Group, UnitPrice)

orderDetails(ID, Prod_ID, Quantity)

r(ID, Prod_ID) = $\Pi_{ID, Prod_ID}$ (order Details)

s(Prod_ID) = Π_{Prod_ID} ($\sigma_{Group='food'}$ products)

$$r \bowtie s \quad (4)$$

$$r \Rightarrow s \quad (3)$$

$$r \bowtie s \quad (2)$$

$$r \div s \quad (1)$$

۴۸- کدام یک از عملگرهای زیر (عملگرهای جبر رابطه ای) در بانک اطلاعاتی نامتمرکز، کاربرد بیشتری دارد؟ (مهندسى کامپیووتر- آزاد ۹۳)

$$\sigma \quad (4)$$

$$\Pi \quad (3)$$

$$\times \quad (2)$$

$$\div \quad (1)$$

۴۹- در جبر رابطه ای عملگر اضافی را می توان با ترکیب عملگرهای اصلی جایگزین نمود، کدام یک از عملگرهای زیر اضافی است؟ (مهندسى کامپیووتر- آزاد ۹۳)

$$\sigma \quad (4)$$

$$\cap \quad (3)$$

$$\cup \quad (2)$$

$$- \quad (1)$$

۵۰- چنانچه R₂ و R₁ دو رابطه با اسامی صفات یکسانی باشند، نتیجه الحق R₁ و R₂ کدام گزینه است؟ (مهندسى IT - آزاد ۹۳)

$$R_1 \cap R_2 \quad (4)$$

$$R_1 \cup R_2 \quad (3)$$

$$R_1 \times R_2 \quad (2)$$

$$R_1 - R_2 \quad (1)$$

۵۱- در جبر رابطه ای کدام یک از عملگرهای زیر نیاز به زمان و حافظه زیادی دارد؟ (مهندسى IT - آزاد ۹۳)

$$\times \quad (1)$$

$$\bowtie \quad (2)$$

$$\bowtie \quad (3)$$

$$\times \quad (4)$$

۵۲- چنانچه R یک رابطه با k صفت (ستون) باشد، تعداد تصاویر R که تهی نمی باشد برابر با کدام گزینه است؟ (مهندسى IT - آزاد ۹۳)

$$k! \quad (4)$$

$$2^k + 1 \quad (3)$$

$$2^k - 1 \quad (2)$$

$$2^k \quad (1)$$

۵۳- چنانچه R₂ و R₁ دو رابطه در پایگاه داده باشند، عبارت R₁ - (R₁ - R₂) برابر با کدام گزینه است؟ (مهندسى IT - آزاد ۹۳)

$$R_1 \cap R_2 \quad (4)$$

$$R_2 \quad (3)$$

$$R_1 - R_2 \quad (2)$$

$$R_1 \cup R_2 \quad (1)$$

(مهندسى IT - آزاد ۹۳)

$$\sigma_p(a - b) = \sigma_p(a) - \sigma_p(b) \quad (۱)$$

$$a - b \neq b - a \quad (۲)$$

$$\sigma_p(a \cap b) = a - b \quad (۳)$$

- ۵۴- کدام فرمول در جبر رابطه‌ای نادرست است؟
 سه رابطه students (دانشجویان)، courses (درس‌ها) و take (اخذ درس) را در نظر گیرید:

students (sid, sname)
courses (cid, cname, dept)

take (sid, cid, grade)

می‌خواهیم شماره دانشجویانی را پیدا کنیم که همه دروس ارائه شده در دانشکده 'CS' را اخذ کرده‌اند. کدام مورد دستور جبر رابطه‌ای صحیح برای این درخواست است؟
 (مهندسى کامپیووتر - دولتی ۹۴)

$$\Pi_{\text{sid}}[\text{take} \div \sigma_{\text{dept}='cs'}(\text{courses})] \quad (۱)$$

$$\Pi_{\text{sid}}\left[\text{take} \div \Pi_{\text{cid}}\left(\sigma_{\text{dept}='cs'}(\text{courses})\right)\right] \quad (۲)$$

$$\Pi_{\text{sid}}\left[\left(\Pi_{\text{sid}}(\text{students}) \times \Pi_{\text{cid}}\left(\sigma_{\text{dept}='cs'}(\text{courses})\right)\right) - \Pi_{\text{sid}, \text{cid}}(\text{take})\right] \quad (۳)$$

$$\Pi_{\text{sid}}(\text{students}) - \Pi_{\text{sid}}\left[\left(\Pi_{\text{sid}}(\text{students}) \times \Pi_{\text{cid}}\left(\sigma_{\text{dept}='cs'}(\text{courses})\right)\right) - \Pi_{\text{sid}, \text{cid}}(\text{take})\right] \quad (۴)$$

- ۵۵- (کار دستور (x, y, p_y، ایجاد یک کپی (یک نام مستعار) از رابطه x با نام y است). اگر رابطه Emp شامل لیستی از کارمندان باشد که در این رابطه ویژگی Salary حقوق کارمند را نشان دهد، کار دستور زیر کدام است؟
 (مهندسى IT - دولتی ۹۴)

$$\Pi_{\text{salary}}(\text{Emp}) - \Pi_{\text{first.salary}}(\sigma_{\text{first.salary} > \text{second.salary}}(\rho_{\text{first}}(\text{Emp}) \times \rho_{\text{second}}(\text{Emp}))$$

(۱) لیست حقوق کارمندانی که حقوق مشابه آنها را کسی ندارد.

(۲) لیست حقوق کارمندان با حذف مقادیر تکراری

(۳) محاسبه ماکزیمم حقوق کارمندان

(۴) محاسبه مینیمم حقوق کارمندان

- ۵۶- اگر درس‌ها و پیش‌نیاز‌های آنها در جدول‌های زیر تعریف شده باشد، کدام یک از پیوندهای زیر می‌تواند وجود مشکلی در اطلاعات ثبت شده در سیستم را نشان دهد؟
 (مهندسى IT - دولتی ۹۴)

Course (course-id , title, units)

Section (course-id , sec-id , semester, year)

Prereg (course-id, prereg-id)

کلید خارجی مرتبه با فیلد course-id course از جدول Course می‌باشد.)

$$\text{Course} \bowtie \text{Prereg} \quad (۱)$$

$$\text{Section} \bowtie \text{Prereg} \quad (۲)$$

$$\text{Course} \Rightarrow\!\! \bowtie \text{Section} \quad (۳)$$

$$\text{Course} \Rightarrow\!\! \bowtie \text{Prereg} \quad (۴)$$

- ۵۷- با توجه به پایگاه داده زیر، عبارت جبر رابطه‌ای معادل پرس و جوی زیر کدام است؟

«نام تمامی اعضایی را بباید که همه کتاب‌های منتشر شده توسط McGraw-Hill را به امانت بردۀ‌اند.»

Member (mem-no, name, age)

Book (ISBN, title, author, publisher)

Borrowed (mem-no, ISBN, date)

(مهندسى IT - دولتی ۹۴)

$$\begin{aligned} \Pi_{(name)}\left(\left(\Pi_{(isbn, name, mem-no)} \text{ Borrowed} \bowtie \text{Member}\right) \div \sigma_{\text{publisher}} = 'McGraw-Hill' \text{ Book}\right) & (۱) \\ \Pi_{(name)}\left(\left(\text{Borrowed} \div \sigma_{\text{publisher}} = 'McGraw-Hill' \text{ Book}\right) \bowtie \text{Member}\right) & (۲) \\ \Pi_{(name)}\left(\left(\Pi_{(mem-no, isbn)} \text{ Borrowed} \div \Pi_{(isbn)} (\sigma_{\text{publisher}} = 'McGraw-Hill' \text{ Book})\right) \bowtie \text{Member}\right) & (۳) \\ \Pi_{(name)}\left(\text{Borrowed} \bowtie \text{Member}\right) \div \Pi_{(isbn)}\left(\sigma_{\text{publisher}} = 'McGraw-Hill' \text{ Book}\right) & (۴) \end{aligned}$$

۵۹- عبارت جبرابطه‌ای «عنوان کتاب‌هایی از انتشارات wiley که توسط هیچ عضوی امانت گرفته نشده» معادل کدام است؟
(مهندسی کامپیوترا- دولتی ۹۶)

Book (ISBN, Title, Author, publisher)

Member (MID, Name, MDate)

Borrow (MID, ISBN, BDate, Duration)

$$\begin{aligned} \Pi_{<\text{Title}>}(\sigma_{\text{publisher}} = 'wiley'(\text{Book})) - \Pi_{<\text{ISBN}>}(\text{Borrow}) & (۱) \\ \Pi_{<\text{Title}, \text{ISBN}>}(\sigma_{\text{publisher}} = 'wiley'(\text{Book})) \div \Pi_{<\text{ISBN}>}(\text{Borrow}) & (۲) \\ \Pi_{<\text{Title}>} \left(\left(\Pi_{<\text{ISBN}, \text{MID}>}(\text{Borrow}) \div \Pi_{<\text{MID}>}(\text{Member}) \right) \bowtie \text{Borrow} \right) & (۳) \\ \Pi_{<\text{Title}>} \left(\left(\Pi_{<\text{ISBN}>}(\sigma_{\text{publisher}} = 'wiley'(\text{Book})) - \Pi_{<\text{ISBN}>}(\text{Borrow}) \right) \bowtie \text{Book} \right) & (۴) \end{aligned}$$

۶۰- بر فرض اینکه R_1 و R_2 دو رابطه باشند که هیچ صفت مشترکی نداشته باشند، اگر CK_1 تنها کلید کاندید رابطه R_1 و CK_2 تنها کلید رابطه R_2 باشد. آنگاه کلید کاندید رابطه حاصل از ضرب کارتزین R_1 و R_2 (یعنی $R_1 \times R_2$)، کدام است؟
(مهندسی کامپیوترا- دولتی ۹۶)

$$CK_1 \cap CK_2 \quad (۱)$$

$$CK_1 \cup CK_2 \quad (۲)$$

$$CK_1 \times CK_2 \quad (۳)$$

۴) هر دو کلید کاندید رابطه حاصل هستند.

۶۱- پایگاه داده رابطه‌ای زیر را در نظر بگیرید. می‌خواهیم نام اعضایی را بیاییم که همه کتاب‌های منتشر شده توسط McGraw-Hill را به امانت بردہ‌اند. کدام مورد، دستور جبرابطه‌ای درست برای این درخواست است؟
(مهندسی IT- دولتی ۹۶)

Member (ID, Name, age)

Book (ISBN, Title, publisher)

Borrowed (ID, ISBN, Date)

$$\begin{aligned} \Pi_{<\text{Name}>} \left[\text{Member} \bowtie \Pi_{<\text{ID}>} \left(\text{Borrowed} \div \left(\sigma_{\text{publisher}} = 'McGraw-Hill' \text{ Book} \right) \right) \right] & (۱) \\ \Pi_{<\text{Name}>} \left[\text{Member} \times \Pi_{<\text{ID}>} \left(\text{Borrowed} \div \left(\sigma_{\text{publisher}} = 'McGraw-Hill' \text{ Book} \right) \right) \right] & (۲) \\ \Pi_{<\text{Name}>} \left[\text{Member} \bowtie \left(\left(\Pi_{<\text{ID}, \text{ISBN}>} \text{ Borrowed} \right) \div \left(\Pi_{<\text{ISBN}>} \left(\sigma_{\text{publisher}} = 'McGraw-Hill' \text{ Book} \right) \right) \right) \right] & (۳) \\ \Pi_{<\text{Name}>} \left[\text{Member} \times \left(\left(\Pi_{<\text{ID}, \text{ISBN}>} \text{ Borrowed} \right) \div \left(\Pi_{<\text{ISBN}>} \left(\sigma_{\text{publisher}} = 'McGraw-Hill' \text{ Book} \right) \right) \right) \right] & (۴) \end{aligned}$$

(مهندسی IT- دولتی ۹۸)

۶۲- پایگاه داده زیر را در نظر بگیرید:

node (NID, Name, Color, Description)

اطلاعات موجود در جدول **node** شامل شماره، نام، رنگ و شرح مربوط به هر گره است.

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: یال‌ها

عبارت جبر رابطه‌ای زیر معادل کدام مورد است؟ (عملگر ρ_{R2} ، نام رابطه $R1$ را به $R2$ تغییر می‌دهد).

$$\Pi_{E1.NID1} \left(\sigma_{E1.NID2 = E2.NID1} \left(\sigma_{E1.EdgeType = T2} (\rho_{E1}(Edge)) \times \rho_{E2}(Edge) \right) \right)$$

۱) شماره گرهایی که حداقل یک یال از نوع $T2$ از آنها خارج شده است.

۲) شماره گرهایی که حداقل یک یال خروجی از نوع $T2$ به یک گره مانند g دارند و گره g حداقل یک یال خروجی دارد.

۳) شماره گرهایی که حداقل یک یال ورودی از نوع $T2$ از یک گره مانند g دریافت می‌کنند و گره g حداقل یک یال خروجی دارد.

۴) شماره گرهایی که حداقل یک یال خروجی از نوع $T2$ به یک گره مانند g دارند و گره g حداقل یک یال خروجی از نوع $T2$ دارد.

۶۳- با توجه به جدول ارائه شده، کدام عبارت جبری درست است؟ (مهندسی کامپیوترا- دولتی ۹۹)

STUD(S#, Sname, City, avg, clg#)

(شماره دانشکده، معدل، محل تولد، نام، شماره دانشجویی) جدول دانشجو

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

$$\sigma_{City} (\Pi_{avg > 16}(STUD)) = \Pi_{avg > 16} (\sigma_{City}(STUD)) \quad (2)$$

$$\Pi_{City} (\sigma_{avg > 16}(STUD)) \quad (3)$$

$$\sigma_{avg > 16} (\Pi_{City}(STUD)) \quad (4)$$

۶۴- کار عملگر $\rho_{Second}(Marks)$ این است که یک کپی از رابطه **Marks** با نام مستعار **Second** ایجاد کند.

کار دستور زیر در جبر رابطه‌ای، کدام است؟ (مهندسی کامپیوترا- دولتی ۹۹)

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

$Marks.mark > Second.mark$

۱) یک لیست تنهی 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.

پاسخ تست‌های فصل چهارم: جبر رابطه‌ای

۱- گزینه (۱) صحیح است.

گزینه اول درست است. زیرا، عملگر نیم پیوند در گزینه اول، پرس و جوی مطرح شده در صورت سئوال را شبیه‌سازی می‌کند.

$$\Pi_{a.*}(\theta \wedge a.FK = b.FK) = \Pi_{a.*}(\sigma_\theta(a.FK = b.FK))$$

↑ ↑
 همه ستون‌های جدول a جابه‌جایی (اگر چه مشروط است)

باید شرط σ زیر مجموعه Π باشد، a . همه مؤلفه‌های θ را شامل می‌شود. پس جابه‌جایی امکان‌پذیر است.

$$\sigma_\theta \left[\Pi_{a.*} \left[a.FK = b.FK \right] \right] \Rightarrow \sigma_\theta(a \ltimes b)$$

تعريف نیم پیوند چپ

گزینه دوم نادرست است، زیرا، با استفاده از عملگر الحق طبیعی علاوه بر ستون‌های رابطه a ، ستون‌های رابطه b را نیز تولید می‌کند.

گزینه سوم نادرست است، زیرا، شرط θ را بیان نکرده است، هر چند ستون‌های جدول a را فقط تولید می‌کند.

گزینه چهارم نادرست است، زیرا، دارای خطای نحوی است، خروجی حاصل از عملگر نیم پیوند چپ در پرانتز داخلی فقط ستون‌های جدول a را تولید می‌کند. حال چطور می‌توان در پرانتز خارجی ستون‌های جدول b را تولید نمود!

۲- گزینه (۳) صحیح است.

به مثال عددی زیر توجه کنید:

: جدول حرارت

کد ناحیه	نام	تاریخ	حرارت بالا	حرارت پایین
1	تهران	x	30	1
2	اهواز	y	50	20
3	شیراز	z	40	10

رابطه‌ی R_1 : حرارت‌های بالای تمامی نواحی مختلف در تاریخ‌های متفاوت را نشان می‌دهد. هم‌چنین نام ستون حرارت بالا به H تغییر کرده است.

H	تاریخ	کد ناحیه
30	X	1
50	Y	2
40	Z	3

رابطه‌ی R_2 : حرارت‌های پایین تمامی نواحی مختلف در تاریخ‌های متفاوت را نشان می‌دهد. هم‌چنین نام ستون حرارت پایین به L تغییر کرده است.

L	تاریخ	کد ناحیه
1	x	1
20	y	2
10	z	3

رابطه‌ی R_3 : سطرهایی از دو جدول R_1 و حرارت را انتخاب می‌کند که شرط «حرارت بالا» در آن برقرار باشد. به عبارت دیگر رابطه R_3 ، کد گروهی از نواحی در رابطه R_1 را می‌دهد که بیشترین حرارت آنها از بیشترین حرارت یکی از نواحی دیگر در جدول حرارت کمتر است.

H	حرارت بالا	کد ناحیه		کد ناحیه
30	40	1	$\xrightarrow{R_3}$	1
40	50	3		3

رابطه‌ی R_4 : سطرهایی از دو جدول R_2 و حرارت را انتخاب می‌کند که شرط «حرارت پایین» در آن برقرار باشد. به عبارت دیگر رابطه R_4 ، کد گروهی از نواحی در رابطه R_2 را می‌دهد که کمترین حرارت آنها از کمترین حرارت یکی از نواحی دیگر در جدول حرارت بیشتر است.

L	حرارت پایین	کد ناحیه		کد ناحیه
20	1	2	$\xrightarrow{R_4}$	2
10	1	3		3

در نتیجه عبارت R_3 -حرارت) کد ناحیه Π ، کد دسته‌ای از نواحی را می‌دهد که حرارت بیشینه دارد.

کد ناحیه	-	کد ناحیه	=	کد ناحیه
1	-	1	=	2
2		3		
3				

هم‌چنین عبارت R_4 -حرارت) کد ناحیه Π ، کد دسته‌ای از نواحی را می‌دهد که حرارت کمینه دارد.

کد ناحیه	-	کد ناحیه	=	کد ناحیه
1	-	2	=	1
2		3		
3				

به این ترتیب اجتماع آنها و الحق طبیعی با جدول حرارت، نام نواحی با درجه حرارت بیشینه و کمینه را به عنوان پاسخ نهایی می‌دهد. بنابراین تهران و اهواز در خروجی نمایش داده می‌شود.

۳- گزینه (۱) صحیح است.

نتیجه الحق طبیعی این دو رابطه به صورت زیر خواهد بود. چنان‌که ملاحظه می‌شود چون در الحق طبیعی ستون مشترک یک بار در خروجی ظاهر می‌شود، خروجی این عبارت چهار ستون دارد. هم‌چنین فقط

سطرهایی در خروجی ظاهر می‌شوند که مقدارشان در ستون مشترک (STU-ID) برابر باشد. به این ترتیب حاصل الحق طبیعی دو رابطه سه سطر خواهد داشت.

STU-ID	NAME	COURSE-ID	TITLE
S001	AHMADI	CSC100	PASCAL PROC.
S001	AHMADI	CSC200	COMPILER WRITING
S002	TAVANA	CSC100	PASCAL PROC.

۴- گزینه (۲) صحیح است.

نتیجه الحق خارجی راست و الحق خارجی چپ این دو رابطه مطابق تعریف این دو عملگر و محتویات داده شده برای دو رابطه، چهار سطر خواهد داشت. هم‌چنین نتیجه الحق خارجی کامل این دو رابطه پنج سطر دارد. زیرا در الحق خارجی کامل علاوه بر تاپل‌های حاصل از الحق طبیعی، تاپل‌های آویزان مثل S003 و S004 به نتیجه نهایی اضافه می‌گردد. توجه کنید که در صورت سؤال مشخص نشده است که منظور از الحق خارجی کدام نوع الحق خارجی است. از آن‌جا که در برخی منابع، منظور از الحق خارجی همان الحق خارجی کامل است. گزینه دوم به عنوان جواب انتخاب می‌شود.

۵- گزینه (۱) صحیح است.

گزینه دوم تعریف عملگر اشتراک را بیان می‌کند. عملگر \bowtie دارای خاصیت جابه‌جایی است اما عملگر \bowtie دارای خاصیت جابه‌جایی نمی‌باشد. بنابراین گزینه‌های دوم تا چهارم عبارات درستی را در جبر رابطه‌ای بیان می‌کنند. اما گزینه اول با توجه به تعریف عملگرهای اجتماع، انتخاب و الحق طبیعی، در جبر رابطه‌ای عبارتی درست تلقی نمی‌شود.

۶- گزینه‌های (۲ و ۴).

هنگامی دو رابطه اجتماع پذیرند که شروط سازگاری به معنی خصیصه‌های یکسان (تیترهای مشابه) برای دو رابطه برقرار باشد. طبق تعریف اگر R و S دو رابطه باشند کلید کاندید US معادل است با اجتماع تمام خصیصه‌های R یا S. بنابراین گزینه دوم البته با اصلاحی جزئی درست است. مثال:

$$\begin{array}{c}
 \begin{array}{c} R: \\ \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 4 \\ \hline 2 & 1 & 5 \\ \hline \end{array} \end{array} \quad \cup \quad \begin{array}{c} S: \\ \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 6 \\ \hline 1 & 3 & 7 \\ \hline \end{array} \end{array} = \begin{array}{c} \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 4 \\ \hline 2 & 1 & 5 \\ \hline 1 & 2 & 6 \\ \hline 1 & 3 & 7 \\ \hline \end{array} \end{array}
 \end{array}$$

توجه کنید که اجتماع کلیدهای دو جدول R و S یعنی (a,b) خاصیت کلیدی ندارد! با توجه به یکسان بودن صفات دو رابطه R و S برای عملگر اجتماع، مجموعه صفات حاصل از تقاطع یا اشتراک صفات دو جدول، کل صفات را شامل می‌شود و جواب مشابه گزینه دوم خواهد بود.

۷- گزینه (۴) صحیح است.

همان‌طور که می‌دانید یک مجموعه با n^2 زیر مجموعه است که البته یکی از آن‌ها مجموعه

تهی است. بنابراین اگر رابطه‌ای دارای n خصیصه باشد، تعداد کل تصاویر (به معنی عملگر پرتو یا Π) آن، 2^n خواهد بود که اگر تصویر تهی آن را کنار بگذاریم برابر $1 - 2^n$ تصویر غیرتهی می‌شود.
مثال: جدول زیر را با سه ستون در نظر بگیرید، تعداد تصاویر غیرتهی برابر چیست؟

X	Y	Z	
X	Y	Z	تصویر
0	0	0	تهی
0	0	1	Z
0	1	0	Y
0	1	1	YZ
1	0	0	X
1	0	1	XZ
1	1	0	XY
1	1	1	XYZ

که جدول بدون ستون وجود ندارد. \Rightarrow

بنابراین تعداد تصاویر غیرتهی برابر مقدار $7 - 1 = 6$ خواهد بود، که در یک عبارت کلی رابطه $1 - 2^n$ را که n برابر تعداد ستون‌ها است، خواهیم داشت.
همچنین در یک راه حل دیگر داریم:
هر تصویر غیرتهی از جدول A، شامل 1، 2، 3 ... یا n صفت از جدول A است. بنابراین تعداد آنها برابر است با:

$$\binom{n}{1} + \binom{n}{2} + \binom{n}{3} + \dots + \binom{n}{n} = 2^n - 1$$

- گزینه (۲) صحیح است.

در یک قاعده کلی اگر R یک رابطه، L مجموعه‌ای از صفت‌ها و \emptyset مجموعه‌ای از شروط باشد، آنگاه $\Pi_L(\sigma_\emptyset(R)) = \sigma_\emptyset(\Pi_L(R))$ تساوی: برقرار است. اگر و فقط اگر تمامی ستون‌های \emptyset زیرمجموعه L باشند، یا به عبارت دیگر شرط \emptyset فقط ستون‌های موجود در مجموعه L را در بر گیرد.

- گزینه (۳) صحیح است.

پیوند طبیعی عملگری است که دو رابطه را برابر مبنای یک یا چند صفت مشترک به هم پیوند می‌دهد. در پیوند طبیعی دو رابطه R و S یعنی $R \bowtie S$ اگر R و S صفت مشترکی نداشته باشند، آنگاه تبدیل به $R \times S$ خواهد شد و اگر روابط R و S تماماً دارای صفات یکسانی باشند، آنگاه $R \bowtie S$ معادل $R \cap S$ خواهد شد.
مثال زیر گویای مطلب می‌باشد:

$$\begin{array}{c|c|c} a & b & c \\ \hline 1 & 2 & 3 \\ 4 & 5 & 6 \end{array} \bowtie \begin{array}{c|c|c} a & b & c \\ \hline 1 & 2 & 3 \\ 1 & 2 & 3 \end{array} = \begin{array}{c|c|c} a & b & c \\ \hline 1 & 2 & 3 \end{array}$$

$$\begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline \end{array} \cap \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline \end{array}$$

- ۱۰- گزینه (۴) صحیح است.

جدول هتل 35 سطر و جدول اتاق 1500 سطر دارد، پس تعداد رکوردهای حاصل از ضرب دکارتی دو جدول برابر با ضرب تعداد رکوردهای دو جدول در یکدیگر است. بنابراین حاصل ضرب دکارتی برابر $35 \times 1500 = 52500$ سطر خواهد بود. در ضرب دکارتی قاعده‌ی زیر برقرار است:

$$R \times S = (R \bowtie S) + (R \bowtie S)$$

با شرط نابرابری با شرط برابری

پس بنابر رابطه‌ی فوق داریم:

$$\text{اتاق } \bowtie \text{ هتل} = \left(\begin{array}{c} \text{اتاق } \bowtie \text{ هتل} \\ \text{کد هتل. اتاق } \neq \text{ کد هتل. هتل} \end{array} \right) + \left(\begin{array}{c} \text{اتاق } \times \text{ هتل} \\ \text{کد هتل. اتاق } = \text{ کد هتل. هتل} \end{array} \right)$$

اما به خروجی $\left(\begin{array}{c} \text{اتاق } \bowtie \text{ هتل} \\ \text{کد هتل. اتاق } = \text{ کد هتل. هتل} \end{array} \right)$ توجه کنید، در جدول هتل، اطلاعات مربوط به هتل، همچون کد هتل وجود دارد، در جدول اتاق نیز، توسط آوردن کد هتل، مشخص شده است که هر اتاق مربوط به کدام هتل است، بنابراین در هر سطر موجود در جدول اتاق، این‌که کدام اتاق، از کدام هتل است، مشخص شده است. بنابراین پیوند شرطی حاصل از دو جدول هتل و اتاق با شرط برابری کد هتل در خصیصه مشترک کد هتل، برابر با تعداد سطرهای جدول اتاق خواهد شد. به این مطلب کمی فکر کنید، زیرا در این حالت همه کدهای هتل موجود در جدول اتاق حتماً زیر مجموعه کدهای هتل، در جدول هتل خواهد بود. کدام اتاق از کدام هتل. بنابراین حاصل این پیوند شرطی برابر 1500 سطر خواهد بود، یعنی به تعداد سطرهای جدول اتاق.

توجه کنید که در جدول اتاق، اطلاعات مربوط به همه اتاق‌های همه هتل‌ها آورده شده است، خواه اقامتی در آن اتاق باشد، خواه نباشد. اطلاعات اقامت مسافران در اتاق، در جدول وضعیت اتاق قرار دارد.

$$\frac{52500}{35 \times 1500} = 1500 + \left(\begin{array}{c} \text{اتاق } \bowtie \text{ هتل} \\ \text{کد هتل. اتاق } \neq \text{ کد هتل. هتل} \end{array} \right)$$

$$\left(\begin{array}{c} \text{اتاق } \bowtie \text{ هتل} \\ \text{کد هتل. اتاق } \neq \text{ کد هتل. هتل} \end{array} \right) = 52500 - 1500 = 51000$$

- ۱۱- گزینه (۱) صحیح است.

در صورت سوال خروجی $\left(\begin{array}{c} \text{وضعیت اتاق } \bowtie \text{ هتل} \\ \text{کد هتل. وضعیت اتاق } = \text{ کد هتل. هتل} \end{array} \right)$ خواسته شده است، هم‌چنین، اطلاعات اقامت مسافران در اتاق، در جدول وضعیت اتاق قرار دارد. اما مطابق آن‌چه پیش از این در پاسخ قبل گفتیم، در جدول هتل، اطلاعات مربوط به هتل، هم‌چون کد هتل وجود دارد، در جدول وضعیت اتاق نیز، توسط آوردن کد هتل، مشخص شده است که هر اتاق دارای اقامت مسافر، مربوط به کدام هتل است، بنابراین در هر سطر موجود در جدول وضعیت اتاق، این‌که کدام اتاق، از کدام هتل است، مشخص شده است. بنابراین

پیوند شرطی حاصل از دو جدول هتل و وضعیت اتاق با شرط برابری کد هتل برابر با تعداد سطرهای جدول وضعیت اتاق خواهد شد. به این مطلب باز هم کمی فکر کنید، زیرا در این حالت هم کدهای هتل موجود در جدول وضعیت اتاق حتماً زیر مجموعه کدهای هتل، در جدول هتل خواهد بود، کدام اتاق از کدام هتل. بنابراین حاصل این پیوند شرطی برابر R سطر خواهد بود، یعنی به تعداد سطرهای جدول وضعیت اتاق.

۱۲- گزینه (۴) صحیح است.

گزینه دوم و سوم نادرست است، زیرا، به طور کلی در جبر رابطه‌ای مفاهیمی همچون اولین، آخرین و یکی از، معنا و مفهومی ندارد. در واقع اگر شرط انتخاب سطر، برای سطری برقرار باشد، سطر مورد نظر در خروجی قرار می‌گیرد.
جداول زیر را در نظر بگیرید:

S			SP			P		
S#	sname	city	S#	P#	QTY	P#	Pname	...
S ₁	S _{n1}	LONDON	S ₁	P ₁	10	P ₁		
S ₂	S _{n2}	PARIS	S ₁	P ₂	20			
S ₃	S _{n3}	PARIS	S ₂	P ₁	30	P ₂		
S ₄	S _{n4}	PARIS	S ₂	P ₂	40			
			S ₃	P ₁	50			

مطابق پرس و جوی مطرح شده در صورت سؤال داریم:

$$\Pi_{S\#}(\sigma_{\text{city}='PARIS'}(S \bowtie SP))$$

ابتدا، در داخلی ترین پرانتز، مشخصات تولیدکنندگانی که حداقل یک قطعه تولید کرده‌اند، ایجاد می‌گردد. سپس، در ادامه، توسط شرط انتخاب سطر، فقط، مشخصات تولیدکنندگانی که حداقل یک قطعه تولید کرده‌اند و اهل شهر پاریس نیز هستند، ایجاد می‌گردد. در آخر، توسط عملگر پرتو، فقط، شماره تولیدکنندگانی که حداقل یک قطعه تولید کرده‌اند و اهل شهر پاریس نیز هستند، ایجاد می‌گردد. مطابق جدول‌های مذکور، خروجی به صورت زیر خواهد بود:

S#
S ₂
S ₃

عبارت جبر رابطه‌ای گزینه اول به صورت زیر می‌باشد:

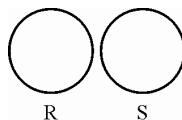
$$\Pi_{S\#}(\sigma_{\text{city}='PARIS'}(S))$$

مطابق جدول‌های مذکور، خروجی به صورت زیر خواهد بود:

S#
S ₂
S ₃
S ₄

۱۳- گزینه (۳) صحیح است.

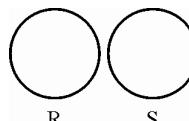
به طور کلی اجتماع دو رابطه با تیترهای یکسان (سازگار) به صورت زیر است:
برای دو رابطه $R(a,b)$ دارای r تاپل و رابطه $S(a,b)$ دارای s تاپل داریم:
اگر $R \cap S = \emptyset$ باشد، یعنی:



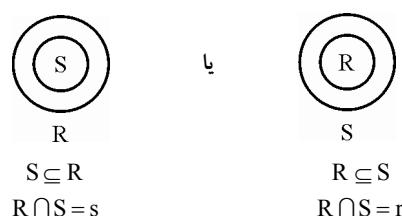
آنگاه، تعداد تاپلهای $R \cup S$ ، برابر $r+s$ است که بیشترین مقدار ممکن خواهد بود.
اگر $S \subseteq R$ یا $R \subseteq S$ باشد، یعنی:



آنگاه، تعداد تاپلهای $R \cup S$ ، برابر $\max(r,s)$ است که کمترین مقدار ممکن خواهد بود.
همچنین اشتراک دو رابطه با تیترهای یکسان (سازگار) به صورت زیر است:
برای دو رابطه $R(a,b)$ دارای r تاپل و رابطه $S(a,b)$ دارای s تاپل داریم:
اگر $R \cap S = \emptyset$ باشد، یعنی:



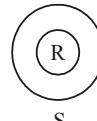
آنگاه، تعداد تاپلهای $R \cap S$ ، برابر تهی است که کمترین مقدار ممکن خواهد بود.
اگر $S \subseteq R$ یا $R \subseteq S$ باشد، یعنی:



آنگاه، تعداد تاپلهای $R \cap S$ ، برابر $\min(r,s)$ است که بیشترین مقدار ممکن خواهد بود.
همچنین تفاضل دو رابطه با تیترهای یکسان (سازگار) به صورت زیر است:
برای دو رابطه $R(a,b)$ دارای r تاپل و رابطه $S(a,b)$ دارای s تاپل داریم:
اگر $R \cap S = \emptyset$ باشد، یعنی:



آنگاه، تعداد تاپل‌های $S - R$ ، برابر r است که بیشترین مقدار ممکن خواهد بود.
اگر $R \subseteq S$ باشد، یعنی:



$$\begin{aligned} R &\subseteq S \\ R - S &= \emptyset \end{aligned}$$

آنگاه، تعداد تاپل‌های $S - R$ ، برابر تهی است که کمترین مقدار ممکن خواهد بود.

- گزینه (۴) صحیح است.

گزینه اول نادرست است، زیرا معادلهای آن به صورت زیر است:

$$\begin{aligned} \sigma_{c-d}(R) &\equiv \sigma_{c \wedge \sim d}(R) \equiv \sigma_c(\sigma_{\sim d}(R)) \\ \sigma_{c \wedge d} &\equiv \sigma_c(\sigma_d(R)) \end{aligned}$$

گزینه دوم نادرست است، زیرا معادلهای آن به صورت زیر است:

$$\begin{aligned} \sigma_{c \wedge d}(R) &\equiv \sigma_c(\sigma_d(R)) \\ \sigma_c(R) - \sigma_d(R) &\equiv \sigma_{c-d}(R) \equiv \sigma_c(\sigma_{\sim d}(R)) \\ \sigma_c(\sigma_{\sim d}(R)) &\equiv \sigma_{c \wedge \sim d}(R) \equiv \sigma_c(R) \cap \sigma_{\sim d}(R) \end{aligned}$$

گزینه سوم نادرست است، زیرا معادلهای آن به صورت زیر است:

$$\begin{aligned} \sigma_{c \vee d}(R) &\equiv \sigma_c(R) \cup \sigma_d(R) \\ \sigma_{c \wedge d}(R) &\equiv \sigma_c(R) \cap \sigma_d(R) \end{aligned}$$

گزینه چهارم درست است، زیرا دو طرف با یکدیگر معادل هستند. در هر دو عبارت تاپل‌هایی از دو رابطه در خروجی ظاهر می‌شوند که شرط c در مورد آنها صدق می‌کند.

- گزینه (۴) صحیح است.

اگر فرض کنیم منظور از join در این سؤال همان ضرب دکارتی است. بنابراین هر سه عملگر عنوان شده در گزینه‌های اول تا سوم جزو عملگرهای اصلی جبر رابطه‌ای بوده و وجودشان برای تمامی سیستم‌های رابطه‌ای و شبیه رابطه‌ای ضروری است.

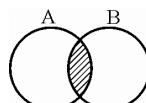
- گزینه (۴) صحیح است.

Times عملگر ضرب دکارتی و Minus عملگر تفاضل در جبر رابطه‌ای است. عملگرهای اجتماع، ضرب دکارتی و تفاضل جزو عملگرهای اصلی جبر رابطه‌ای هستند و عملگرهایی که با استفاده از عملگرهای

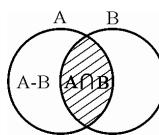
اصلی قابل ایجاد هستند، عملگرهای فرعی هستند، مانند عملگر اشتراک و تقسیم.

$$A \cap B = A - (A - B) = B - (B - A)$$

مثال:



با توجه به شکل زیر، مشاهده می‌شود که با کنار گذاشتن ناحیه $A - B$ از ناحیه A ، قسمت باقی مانده برابر $A \cap B$ است.



۱۷- گزینه (۱) صحیح است.

عبارت داده شده اشتراک دو رابطه را ایجاد می‌کند:

$$[A - (A - B)] \cup [B - (B - A)] = [A \cap B] \cup [A \cap B] = A \cap B$$

۱۸- گزینه (۲) صحیح است.

جدوال زیر را در نظر بگیرید:

S#	Sname	S#	L#	Grade	L#	Lname
S ₁	Sn ₁	S ₁	L ₁	10	L ₁	Ln ₁
S ₂	Sn ₂	S ₁	L ₂	15	L ₂	Ln ₂
		S ₂	20	20	(L)	جدول درس
جدول دانشجو (S)		جدول نمره (G)				

از جداول فوق مشخص است که دانشجویی با شماره دانشجویی S₂ از درس L₂ نمره‌ای در جدول نمره ندارد. یعنی با توجه به پرس‌وجوهی مطرح شده در صورت سؤال عبارت (S₂, L₂) پاسخ پرس‌وجوه خواهد بود.

برای تولید عبارت فوق با استفاده از جبر رابطه‌ای پردازش است که ابتدا باید جدول دانشجو و جدول درس را در هم ضرب دکارتی نمود تا حاصل آن برابر تمامی حالت‌های همه دانشجوها با همه درس‌ها شود. سپس تفاضل این حاصل را با جدول نمره محاسبه نمود. اما برای برقراری شروط سازگاری (تیترهای یکسان) در عملگر تفاضل باید دستورات زیر را نوشت:

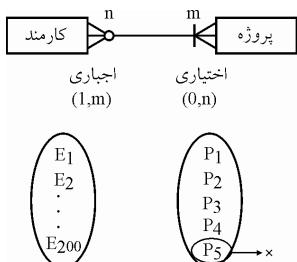
$$[\Pi_{S\#, L\#}(S \times L)] - [\Pi_{S\#, L\#}(G)]$$

عبارت فوق معادل عبارت زیر می‌باشد و پاسخ سؤال نیز می‌باشد:

$$(S \text{ TIMES } L)[S\#, L\#] \text{ MINUS } G[S\#, L\#]$$

۱۹- گزینه (۴) صحیح است.

براساس نمودار ER موجود در صورت سؤال اطلاعات زیر از آن قابل برداشت می‌باشد:



پروژه P5 به دلیل اختیاری بودن موجودیت پروژه می‌تواند در ارتباط با نمونه موجودیت‌های کارمند شرکت نکند. اما همه کارمندان از E1 تا E200 به دلیل اجباری بودن موجودیت کارمند باید در ارتباط با نمونه موجودیت‌های پروژه شرکت کنند.

با توجه به رابطه ذکر شده، یک پروژه می‌تواند هیچ کارمندی نداشته باشد، چون در رابطه می‌تواند شرکت نکند مانند P5 و یا اگر یک پروژه در رابطه شرکت کرد می‌تواند حداقل 200 کارمند داشته باشد. در طرف مقابل یک کارمند باید حداقل یک پروژه داشته باشد چون حضور کارمند اجباری است و حداقل می‌تواند 5 پروژه داشته باشد.

برای یافتن حداقل تعداد رکوردها در پیوند طبیعی این دو موجودیت باید حالت را در نظر گرفت که همه کارمندان در اجرای همه پروژه‌ها مشارکت داشته‌اند. در این حالت تعداد سطرهای خروجی برابر $5 \times 200 = 1000$ خواهد بود. همچنین برای یافتن حداقل تعداد رکوردها در پیوند طبیعی این دو موجودیت باید حالت را در نظر گرفت که هر کارمند فقط در اجرای یک پروژه مشارکت داشته است. در این حالت تعداد سطرهای خروجی برابر با $1 \times 200 = 200$ خواهد بود.

- ۲۰- گزینه (۲) صحیح است.

پیوند طبیعی عملگری است که دو رابطه را برابر مبنای یک یا چند صفت مشترک به هم پیوند می‌دهد. در پیوند طبیعی دو رابطه A و B یعنی $A \bowtie B$ اگر A و B صفت مشترکی نداشته باشند $A \bowtie B$ تبدیل به $A \times B$ خواهد شد و اگر روابط A و B تماماً دارای صفات یکسانی باشند، آنگاه $A \bowtie B$ معادل $B \cap A$ خواهد شد. مثال زیر گویای مطلب می‌باشد:

$$\begin{array}{c|c|c} a & b & c \\ \hline 1 & 2 & 3 \\ 4 & 5 & 6 \end{array} \bowtie \begin{array}{c|c|c} a & b & c \\ \hline 1 & 2 & 3 \end{array} = \begin{array}{c|c|c} a & b & c \\ \hline 1 & 2 & 3 \end{array}$$

$$\begin{array}{c|c|c} a & b & c \\ \hline 1 & 2 & 3 \\ 4 & 5 & 6 \end{array} \cap \begin{array}{c|c|c} a & b & c \\ \hline 1 & 2 & 3 \end{array} = \begin{array}{c|c|c} a & b & c \\ \hline 1 & 2 & 3 \end{array}$$

- ۲۱- گزینه (۴) صحیح است.

مطابق تعریف عملگر تقسیم در جبر رابطه‌ای فقط گزینه چهارم عملگر تقسیم را بیان می‌کند.

توجه کنید که عبارت $\Pi_{R-S, S}(r)$ معادل r می‌باشد. یعنی عبارت زیر نیز عملگر تقسیم را تعریف می‌کند:

$$\Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times S) - r)$$

- ۲۲- گزینه (۳) صحیح است.

گزینه سوم نادرست است، زیرا، دارای خطای نحوی است. در گزینه سوم، در ابتدا توسط عملگر نیم پیوند، و شرط انتخاب سطر، مشخصات اساتیدی استخراج می‌گردد که متخصص DB هستند. دقت کنید که در عملگرد نیم پیوند چپ، فقط ستون‌های جدول سمت چپ در خروجی قرار می‌گیرد، در اینجا ستون‌های

جدول prof در خروجی قرار می‌گیرد. حال چطور ممکن است، در پرانتز خارجی توسط عملگر پرتو، ستون depname استخراج گردد، در حالی که این ستون در جدول prof قرار ندارد. گزینه‌های اول، دوم و چهارم یک خروجی یکسان، اما با شیوه‌های مختلف را تولید می‌کنند. به عبارت دیگر این گزینه‌ها نام استاد و نام دپارتمان اساتیدی که متخصص DB هستند را در خروجی نمایش می‌دهد.

۲۳- گزینه (۲) صحیح است.

طبق تعریف عملگر الحق طبیعی در جبر رابطه‌ای ستون یا ستون‌های مشترک فقط یک بار در خروجی ظاهر می‌شود. در اینجا دو جدول دارای دو ستون مشترک یعنی A و C هستند، پس خروجی عبارت $T_1 \bowtie T_2$ دارای چهار ستون خواهد بود. همچنین فقط سطرهایی از دو جدول در خروجی ظاهر می‌شوند که مقدارشان در ستون‌های مشترک مساوی باشد، بنابراین خروجی عبارت داده شده به صورت زیر خواهد بود. ملاحظه می‌شود که عبارت $T_1 \bowtie T_2$ دارای سه سطر است.

A	B	C	E
1	5	1	3
2	7	3	3
2	7	3	5

۲۴- گزینه (۱) صحیح است.

جداول زیر را در نظر بگیرید:

S#	Sname	...	S#	T #	...	T #	Tname	...
S ₁	S _{n1}		S ₁	T ₁		T ₁	T _{n1}	
S ₂	S _{n2}		S ₁	T ₂		T ₂	T _{n2}	
S ₃	S _{n3}		S ₂	T ₁				
S ₄	S _{n4}		S ₂	T ₂				
			S ₃	T ₁				

جدول داوطلب (S)

جدول شرکت در آزمون (ST)

جدول آزمون (T)

مطلوب پرس و جوی مطرح شده در گزینه اول داریم: $(\Pi_{S\#, T\#}(ST) \div \Pi_{T\#}(T)) \bowtie S$ ابتدا در پرانتز داخلی توسط عملگر تقسیم، شماره داوطلبانی که در همه آزمون‌ها شرکت کرده‌اند، استخراج می‌گردد، به صورت زیر:

S#	T #		T #		S#
S ₁	T ₁		T ₁		S ₁
S ₁	T ₂		T ₂		S ₂
S ₂	T ₁				
S ₂	T ₂				
S ₃	T ₁				

سپس خروجی فوق با جدول S الحق طبیعی می‌گردد و در نهایت مشخصات داوطلبانی که در همه آزمون‌ها شرکت کرده‌اند، در خروجی قرار می‌گیرد. به صورت زیر:

S#	Sname	...
S ₁	S _{n1}	
S ₂	S _{n2}	

گزینه دوم نادرست است.

مطابق پرس و جوی مطرح شده در گزینه دوم داریم:

ابتدا در پرانتز داخلی توسط عملگر تقسیم، شماره داوطلبانی که در همه آزمون‌ها شرکت کرده‌اند، استخراج می‌گردد، به صورت زیر:

S#	T#		T#		S#
S ₁	T ₁	÷	T ₁	=	S ₁
S ₁	T ₂		T ₂		S ₂
S ₂	T ₁				
S ₂	T ₂				
S ₃	T ₁				

سپس خروجی فوق با جدول S، نیم پیوند چپ می‌گردد و در نهایت مطابق عملگر نیم پیوند چپ فقط ستون‌های جدول سمت چپ در خروجی قرار می‌گیرد. یعنی فقط شماره داوطلبانی که در همه آزمون‌ها شرکت کرده‌اند، به عنوان خروجی نمایش داده می‌شود. به صورت زیر:

S#	S#	Sname	...	S#
S ₁	S ₁	S _{n1}		S ₁
S ₂	S ₂	S _{n2}		S ₂
S ₃	S ₃	S _{n3}		
S ₄	S ₄	S _{n4}		

در حالی که در صورت سؤال، مشخصات داوطلبانی که در همه آزمون‌ها شرکت کرده‌اند، مورد پرس و جو قرار گرفته است.

گزینه‌های سوم و چهارم نادرست هستند. زیرا دارای خطای نحوی هستند. مطابق قوانین عملگر تقسیم در جبر رابطه‌ای، ستون‌های مقسوم علیه، باید زیرمجموعه ستون‌های مقسوم باشد. اما همانطور که در این دو گزینه مشاهده می‌شود، ستون Tname در مقسوم علیه، زیرمجموعه مقسوم نیست.

۲۵- گزینه (۲) صحیح است.

جداول زیر را در نظر بگیرید:

S#	Sname	S#	L#	Term#	Grade
S ₁	Sn ₁	S ₁	L ₁	92	20
S ₂	Sn ₂	S ₁	L ₂	91	20
S ₃	Sn ₃	S ₁	L ₃	90	20
جدول S		S ₂	L ₁	92	20
جدول L		S ₂	L ₂	90	18
جدول Term		S ₃	L ₁	92	19

SL[S#]Minus(SL where grade < 20)[S#]

جدول SL

جبر رابطه‌ای عبارت فوق به صورت مقابله است:

$\Pi_{S\#}(SL) - \Pi_{S\#}(\sigma_{grade < 20}(SL))$ بخش اول عبارت فوق، شماره دانشجویانی را می‌دهد که تاکنون حداقل یک نمره برای آن‌ها ثبت شده است.

و بخش دوم عبارت فوق، شماره دانشجویانی را می‌دهد که حداقل یک نمره غیر بیست برای آن‌ها ثبت شده است.

بنابراین حاصل تفاضل این دو بخش شماره دانشجویانی را می‌دهد که هیچ نمره غیر بیستی ندارند یا به عبارت دیگر شماره دانشجویانی را می‌دهد که تمام نمرات آن‌ها بیست است.

$$\begin{array}{c} \underline{\text{S}\#} \quad \underline{\text{S}\#} \quad \underline{\text{S}\#} \\ \text{S}_1 - \text{S}_2 = \text{S}_1 \\ \text{S}_2 \quad \text{S}_3 \\ \text{S}_3 \end{array}$$

- گزینه (۲) صحیح است.

گزینه اول اشتباه است چون در رابطه تقسیم باید رابطه سمت چپ تقسیم شامل ستون‌های رابطه سمت راست تقسیم باشد. به عبارت دیگر صفات مقسوم‌علیه زیر مجموعه صفات مقسوم باشد.

گزینه سوم اشتباه است چون در رابطه اشتراک باید ستون‌های متناظر یکسان باشد. (تیترهای مشابه)

- گزینه (۳) صحیح است.

عملگر \bowtie ، به طورکلی، سطرهایی از جداول را در خروجی می‌آورد که صفت‌های همنام آن‌ها مقدار یکسانی داشته باشند. به علاوه، ستون‌های همنام فقط یک بار در خروجی ظاهر می‌شوند. بنابراین عبارت $A \bowtie B$ ، سطرهایی از دو جدول A و B که مقادیر تمام صفت‌هایشان با هم برابر باشد را بر می‌گرداند. پس خروجی، همان اشتراک $A \bowtie B$ است.

مطابق یک قاعده کلی اگر تمامی صفات دو رابطه A و B یکسان باشند، آن‌گاه $A \bowtie B = A \cap B$. همچنان اگر هیچ صفت مشترکی بین دو رابطه A و B نباشد، آن‌گاه $A \bowtie B = A \times B$. بنابر آن‌چه در صورت سؤال مطرح شده است، مبنی بر یکسان بودن صفات دو رابطه A و B پس می‌توان نتیجه گرفت که گزینه سوم درست و گزینه دوم نادرست است. البته به این مسئله نیز توجه کنید که خروجی عبارت $A \bowtie B$ با $A \times B$ برابر است! که در گزینه اول و همچنین با $A \times B$ در گزینه چهارم به دلیل یکسان بودن تمام صفات A و B برابر است!

طراح محترم این مسئله را لحاظ نکرده‌اند و گزینه سوم را به عنوان پاسخ درست اعلام کرده‌اند!

مثال‌های زیر گویای مطلب است:

$$\begin{array}{c} \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline \end{array} \bowtie \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline \end{array} \\[10pt] \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline \end{array} \bowtie \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline \end{array} \\[10pt] \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline \end{array} \bowtie \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline \end{array} \\[10pt] \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline \end{array} \cap \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline \end{array} \end{array}$$

۲۸- گزینه (۲) صحیح است.

ابتدا خروجی دستور $T1 \times T2$ را تعیین می‌کنیم: با توجه به شرط $T1.B = T2.B$, باید بینیم.

$$T1.B = T2.B$$

هر سطر از جدول $T1$ با چند سطر از جدول $T2$ تطابق دارد. (منظور از «تطابق» برابری مقدار فیلد B در دو جدول است). سطر اول از $T1$ با سطرهای اول و دوم از $T2$ تطابق دارد. پس به ازای آنها، دو سطر در خروجی تولید می‌شود. سطر دوم از $T1$ با سطرهای سوم، چهارم و پنجم از $T2$ تطابق دارد. پس به ازای آنها سه سطر در خروجی تولید می‌شود. سطر سوم از $T1$ با هیچ سط्रی از $T2$ تطابق ندارد و به ازای آن هیچ سطري در خروجی تولید نمی‌شود. با توجه به این توضیحات، خروجی دستور $T1 \times T2$ به شکل زیر است:

$$T1.B = T2.B$$

A	T1.B	T2.B	C
a1	b1	b1	c1
a1	b1	b1	c2
a2	b2	b2	c2
a2	b2	b2	c3
a2	b2	b2	c4

حال باید حاصل انجام $\Pi_{T1.B}$ روی خروجی فوق را تعیین کنیم. واضح است که این دستور ستون $T1.B$ از خروجی فوق را استخراج می‌کند. تنها نکته‌ای که باید به آن توجه کنیم این است که در خروجی عملگر Π ، سطرهای تکراری وجود ندارد، بنابراین خروجی نهایی به صورت زیر است:

T1.B
b1
b2

پس خروجی دستور داده شده، فقط دو سطر دارد.

۲۹- گزینه (۳) صحیح است.

Query1 را در نظر بگیرید. عبارت S سطرهایی از رابطه‌های R و S که مقدار فیلد b آنها یکسان است را با هم پیوند می‌دهد. سپس عملگر پرتوی $\Pi_{a,b}$ فقط ستون‌های a و b (که مربوط به رابطه R هستند) را انتخاب می‌کند. پس می‌توان خروجی نهایی را این‌گونه توصیف کرد:

Query1 سطرهایی از رابطه R را بر می‌گرداند که مقدار صفت b آنها با مقدار صفت b در یکی از سطرهای رابطه S برابر باشد.

حال Query2 را در نظر بگیرید. دستور $\Pi_c(s)$ ستون C از رابطه S را انتخاب می‌کند. سپس دستور نامگذاری $Pt(a)$ ، جدول خروجی دستور $\Pi_t(s)$ را برابر t و نام تنها ستون آن را برابر a قرار می‌دهد. (این تغییر نام برای فراهم سازی امکان اجرای عملگر اشتراک انجام شده است، زیرا برای انجام اشتراک باید صفات دو رابطه یکسان باشند). دستور $\Pi_b(s)$ ستون b از رابطه S را انتخاب می‌کند. خروجی این دستور در خروجی دستور قبلی ضرب دکارتی می‌شود و جدولی با شمای زیر حاصل می‌شود.

$$Pt(a)(\Pi_c(s)) \times \Pi_b(s)$$

S.c	S.b

a	b



سطرهای جدول فوق، شامل هر یک از مقادیر مختلف ستون c از رابطه S است که در کنار هر یک از مقادیر مختلف ستون b از رابطه S قرار گرفته باشد. به بیان دیگر براساس محتویات رابطه S همه ترکیبات مختلف برای مقادیر دو صفت b و c در جدول فوق قرار دارد.

در نهایت، اشتراک جدول فوق با جدول R محاسبه می‌شود. می‌دانیم که در خروجی اشتراک، سطرهایی از دو جدول ظاهر می‌شوند که مقدار صفت‌های آن‌ها به ترتیب با هم برابر باشد. یعنی خروجی نهایی شامل سطرهایی از R است که صفت a از آن با صفت S.c (که S به a تغییر نام داده است) و صفت b از آن با صفت S.b از جدول فوق برابر باشد.

اگر خروجی 2 Query را با خروجی 1 Query مقایسه کنید می‌بیند که در خروجی 1 سطرهایی از ظاهر می‌شوند که مقدار صفت b آن‌ها در S.b وجود داشته باشد اما در خروجی 2 علاوه بر شرط تساوی مقادیر صفت‌های b از R و S، شرط تساوی صفت a از R با صفت c از S (که c به a تغییر نام داده است) نیز در نظر گرفته شده است. پس خروجی 2 محدودتر از Query1 است و زیر مجموعه‌ی آن می‌باشد.

مثال: دو جدول R(a,b,c) و S(b,c) را با مقادیر زیر در نظر بگیرید:

a	b
1	2
3	4
6	7

جدول R

b	c
2	1
4	5

جدول S

Query1: $\Pi_{a,b}(R \bowtie S)$

$$\text{Query1: } \Pi_{a,b} \left(\begin{array}{c|c} a & b \\ \hline 1 & 2 \\ 3 & 4 \\ 6 & 7 \end{array} \bowtie \begin{array}{c|c} b & c \\ \hline 2 & 1 \\ 4 & 5 \end{array} \right) = \begin{array}{c|c} a & b \\ \hline 1 & 2 \end{array}$$

Query2: $R \cap (\rho_t(a)(\Pi_c(S)) \times \Pi_b(S))$

$$\downarrow \quad \quad \quad \downarrow$$

$$\left(\begin{array}{c|c} a & b \\ \hline 1 & 2 \\ 5 & 4 \end{array} \times \begin{array}{c|c} b & c \\ \hline 2 & 1 \\ 4 & 5 \end{array} \right)$$

$$\begin{array}{c|c} a & b \\ \hline 1 & 2 \\ 3 & 4 \\ 6 & 7 \end{array} \cap \begin{array}{c|c} a & b \\ \hline 1 & 2 \\ 1 & 4 \\ 5 & 2 \end{array} = \begin{array}{c|c} a & b \\ \hline 1 & 2 \end{array}$$

۳۰- گزینه (۱) صحیح است.

قبل از هر چیز توجه کنید که کلمه‌ی «همه» در عبارت داده شده زاید است و بنابراین نیازی به استفاده از عملگر \neq نیست.

برای نوشتن عبارت جبر رابطه‌ای مربوطه، مراحل زیر را به ترتیب انجام می‌دهیم:

(۱) از جدول قطعه (P) سطرهای مربوط به قطعات آبی رنگ را به دست می‌آوریم:

(۲) از خروجی مرحله‌ی قبل، فقط ستون «شماره قطعه» (#P) را انتخاب می‌کنیم:

$\Pi_{P\#}(\sigma_{color="Blue"}(P))$

با اجرای دستور فوق، شماره‌ی قطعات آبی رنگ به دست می‌آید.

(۳) خروجی مرحله‌ی ۲ را با جدول محموله (SP) پیوند طبیعی می‌زنیم تا سطرهایی از جدول SP که # آنها برابر یکی از مقادیر خروجی مرحله‌ی ۲ است را به دست آوریم. به عبارت دیگر، با این کار، سطرهایی از جدول SP که متناظر با قطعات آبی رنگ هستند به دست می‌آیند:

(۴) از خروجی مرحله‌ی قبل، فقط ستون «شماره‌ی تهیه‌کننده» (S#) را انتخاب می‌کنیم:

$\Pi_{S\#}(\Pi_{P\#}(\sigma_{color="Blue"}(P)) \bowtie SP)$

با اجرای دستور فوق، شماره‌ی تهیه‌کننده‌هایی که قطعه‌ی آبی رنگ تولید می‌کنند به دست می‌آید.

(۵) خروجی مرحله‌ی قبل را با جدول تهیه‌کننده (S) پیوند طبیعی می‌زنیم تا سطرهایی از جدول S که متناظر با تهیه‌کننده‌های قطعه‌های آبی رنگ هستند به دست آید:

$\Pi_{S\#}(\Pi_{P\#}(\sigma_{color="Blue"}(P)) \bowtie SP) \bowtie S$

(۶) از خروجی مرحله‌ی قبل، فقط ستون «اسم تهیه‌کننده» (Sname) را انتخاب می‌کنیم:

$\Pi_{Sname}(\Pi_{S\#}(\Pi_{P\#}(\sigma_{color="Blue"}(P)) \bowtie SP) \bowtie S$

۳۱- گزینه (۲) صحیح است.

گزینه‌ی اول عبارت درستی است. در جدول حاصل از انجام A-B، کلید اصلی A به عنوان کلید اصلی در نظر گرفته می‌شود.

گزینه‌ی دوم عبارت نادرستی است: در \bowtie نیازی نیست شروط سازگاری (تیترهای مشابه) در دو رابطه برقرار باشد.

گزینه‌ی سوم عبارت درستی است: عملگرهای جبر رابطه‌ای به دو دسته‌ی اصلی و فرعی تقسیم می‌شوند. عملگرهای اصلی عبارتند از گزینش (۵)، پرتو (Π)، اجتماع (U)، تفاضل (-)، ضرب دکارتی (\times) و جایگزینی (\leftarrow). عملگرهای فرعی را می‌توان از روی عملگرهای اصلی به دست آورد. برای مثال، اشتراک $A \cap B = A - (A - B)$ و پیوند طبیعی را می‌توان به صورت زیر به دست آورد:

$$A \bowtie B = A \times B$$

شرط تساوی

ستون‌های همنام

هم‌چنین اگر صفات مشترک دو رابطه‌ی A و B را Y و بقیه‌ی صفات رابطه‌ی A را X بنامیم آنگاه $A \div B$

$\Pi_X(A) - \Pi_X(\Pi_X(A) \times \Pi_Y(B) - A)$

را می‌توان به صورت مقابل نوشت:

گزینه‌ی چهارم عبارت (تقریباً) درستی است: اگر رابطه‌ای n صفت داشته باشد با عملگر پرتو می‌توان

(۱) تصویر مختلف از آن به دست آورد: یکی از n ستون را انتخاب کنیم یا دو تا از n ستون را انتخاب کنیم یا یا تمام n ستون را انتخاب کنیم، که مجموع تعداد این انتخاب‌ها برابر است با:

$$\binom{n}{1} + \binom{n}{2} + \binom{n}{3} + \dots + \binom{n}{n} = 2^n - 1$$

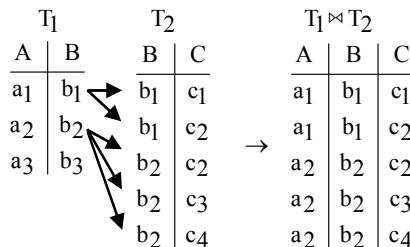
- گزینه (۳) صحیح است.

چون از full outer join استفاده شده پس تمام سطرهای دو جدول در خروجی ظاهر می‌شوند و سطرهایی که امکان Natural Join آن‌ها وجود دارد (یعنی ستون‌های هم نامشان، یعنی A ، مقدار یکسان دارند) با هم قرار می‌گیرند. پس خروجی به صورت زیر است:

A	B	C	D
1	2	2	3
1	2	3	4
3	4	NULL	NULL
2	NULL	4	5

- گزینه (۱) صحیح است.

پیوند طبیعی (自然连接) سطرهایی از دو جدول که ستون‌های هم نامشان مقادیر یکسانی دارند را با هم ترکیب می‌کند (ستون‌های هم نام یک بار در خروجی می‌آیند). بنابراین حاصل پیوند طبیعی به صورت زیر است:



بنابراین با انجام عملگر پرتو (Π_B) فقط ستون B انتخاب و در خروجی ظاهر می‌شود:

B
b1
b2

توجه کنید که چون خروجی عملگر Π یک مجموعه است و در مجموعه‌ها عضو تکراری معنی ندارد، سطرهای تکراری در خروجی وجود نخواهد داشت. بنابراین فقط دو سطر در خروجی وجود دارد.

- گزینه (۳) صحیح است.

گزینه اول اشتباه است. برای اثبات این موضوع، فرض کنید مقادیر زیر در جدول‌های R و S وجود داشته باشند (حروف بزرگ نام ستون‌ها و حروف کوچک مقادیر فیلدها هستند).

A	B	C	D
a	b	c	d

R جدول S جدول

بنابراین $R \bowtie S$ به صورت زیر خواهد بود (توجه کنید که چون R و S ستون همنامی ندارند پیوند طبیعی معادل ضرب دکارتی خواهد بود):

A	B	C	D
a	b	c	d

بنابراین خروجی $\Pi_{A,B,D}(R \bowtie S)$ به صورت زیر است:

A	B	D
a	b	d

اما خروجی $(S) \rho_{T(B,D)}$ به صورت زیر است:

B	D
c	d

و بنابراین حاصل $(S) \rho_{T(B,D)} R \bowtie S$ هیچ سط्रی در بر ندارد! (چون ستون‌های همنام در جدول R و جدول قبلی مقادیر مشترکی ندارند و بنابراین پیوند طبیعی \bowtie هیچ سطري برنمی‌گردداند) گزینه دوم اشتباه است. سمت چپ تساوی، حاصل $S-R$ را بر می‌گرداند در صورتی که سمت راست تساوی حاصل $R-S$ را به ما می‌دهد. گزینه سوم صحیح است. با انجام ضرب دکارتی $S \times R$ ، هر سطر از R در کنار تمام سطرهای S قرار می‌گیرد. اما با انجام $\Pi_{A,B}$ فقط ستون‌های A و B (ستون‌های R) در خروجی می‌آیند. توجه کنید که در خروجی Π ، سطر تکراری وجود ندارد، بنابراین خروجی به دست آمده معادل R می‌باشد.

۳۵- گزینه (۱) صحیح است.

برای پاسخگویی به این سؤال باید به صورت زیر از تفیریق استفاده کنیم:

{نهیه کنندگانی که قطعه آبی تولید می‌کنند} - {تمام تهیه کننده‌ها}

و سپس با اعمال Π_{sname} ، فقط نام تهیه کننده‌ها را انتخاب کنیم:

{نهیه کنندگانی که قطعه آبی تولید می‌کنند} - {تمام تهیه کننده‌ها}

مجموعه «تمام تهیه کننده‌ها» همان جدول S است. برای به دست آوردن مجموعه «تهیه کنندگانی که قطعه آبی تولید می‌کنند» به صورت زیر عمل می‌کنیم: از جدول قطعه (P)، سطرهای متناظر با قطعه آبی را انتخاب می‌کنیم: $\sigma_{Color='Blue'}(P)$

و سپس، از حاصل آن، فقط شماره قطعه (P) را بر می‌داریم:

سپس باید در جدول محموله‌ها (SP)، محموله‌هایی را بیابیم که مربوط به شماره قطعه‌های فوق باشند. برای

این منظور، جدول SP را با نتیجه قبلی، پیوند طبیعی می‌کنیم:

$\Pi_{P\#}(\sigma_{Color='Blue'}(P)) \bowtie SP$

سپس از حاصل فوق، فقط شماره تهیه کننده (S#) را بر می‌داریم:

$\Pi_{S\#}(\Pi_{P\#}(\sigma_{Color='Blue'}(P)) \bowtie SP)$

حاصل عبارت فوق، شماره تهیه کنندگانی است که قطعه آبی تولید می‌کنند. برای به دست آوردن نام این

تهیه کننده‌ها، باید عبارت فوق را به جدول تهیه کننده‌ها (S) پیوند طبیعی کنیم:

$\Pi_{S\#}(\Pi_{P\#}(\sigma_{Color='Blue'}(P)) \bowtie SP) \bowtie S$

۳۶- گزینه (۱ و ۴) صحیح است.

توجه کنید که گزینه‌های اول و چهارم و همچنین گزینه‌های دوم و سوم یکسان هستند!

برای پاسخ به این سؤال باید جدول Person و Parent (یا زیر مجموعه‌هایی از آن‌ها) روى فیلد Pid با هم پیوند زده شوند. این کار در گزینه‌های اول و چهارم با عملگر \bowtie انجام شده است (توجه کنید که عملگر \bowtie دو جدول را روی فیلدی همانشان پیوند می‌زند و Pid تنها فیلد همنام این دو جدول است). شرط‌های "مرد" =sex و "آرش کمانگیر" =Pname در عملگرهای σ باعث انتخاب سطرهای لازم می‌شوند. در نهایت عملگر Π_{ppid} ستون شناسه پدر را بر می‌گرداند.

توجه کنید که در گزینه‌های دوم و سوم عملگر \bowtie باعث می‌شود فقط ستون‌های جدول سمت چپ پیوند (یعنی ستون‌های Person) در خروجی پیوند ظاهر شود. از آن‌جا که در جدول Person ستون Ppid وجود ندارد، عملگر Π_{ppid} به دلیل وجود خطای نحوی، قابل انجام نخواهد بود! پس این دو گزینه اشتباه هستند.

۳۷ - گزینه (۲) صحیح است.

برای به دست آوردن تمام اجداد یک نفر تا هر تعداد نسل قبل، لازم است جدول Person به طور بازگشتی با جدول Parent پیوند زده شود. اما query بازگشتی در جبر رابطه‌ای جایی ندارد. بنابراین گزینه دوم در جبر رابطه‌ای قابل پاسخ‌گویی نیست.

جداول Person و Parent را با مقادیر زیر در نظر بگیرید:

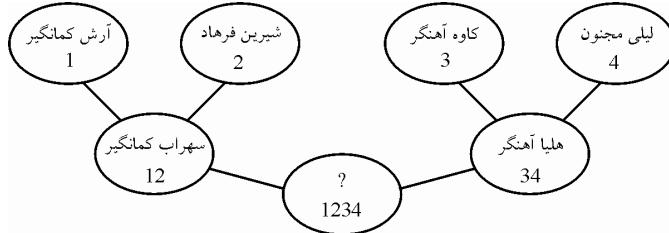
pid	pname	age	DOB	pid	ppid	Sex
1	آرش کمانگیر	93	1300	12	1	مرد
2	شیرین فرhad	91	1302	12	2	زن
12	سهراب کمانگیر	63	1330	34	3	مرد
3	کاوه آهنگر	53	1340	34	4	زن
4	لیلی مجذون	43	1350			
34	هلیا آهنگر	28	1365			

جدول Person

جدول Parent

توجه: فرض کنید در سال ۱۳۹۳ قرار داریم.

درخت زیر وجود دارد:



توجه: واضح است که نام خانوادگی شماره 1234 پس از تولد "کمانگیر" خواهد بود.

برای پاسخ به گزینه اول با پرس و جوی «همسر فردی به نام "آرش کمانگیر"» روال زیر را باید دنبال کرد:

برای این کار ابتدا باید pid آرش کمانگیر را از جدول person به صورت زیر شناسایی نمود.

$$\Pi_{pid}(\sigma_{pname} = "آرش کمانگیر") \rightarrow \frac{pid}{1}$$

سپس از جدول Parent فردی که یکی از والدین او "آرش کمانگیر" است را شناسایی نمود.

$$\Pi_{pid}(\sigma_{ppid=1}(parent) \rightarrow \frac{pid}{12}$$

سپس از جدول Parent، ppid یکی دیگر از والدین فردی با pid برابر مقدار 12 و جنسیت برابر زن را می‌توان شناسایی نمود.

$$\Pi_{ppid}(\sigma_{pid=12 \wedge sex = "زن"}(parent) \rightarrow \frac{ppid}{2}$$

سپس از جدول person نام فردی با pid برابر مقدار 2 را می‌توان شناسایی نمود.

$$\Pi_{pname}(\sigma_{pid=2(person)} \rightarrow \frac{pname}{شیرین فرهاد}$$

در واقع شیرین فرهاد، همسر آرش کمانگیر است.

برای پاسخ به گزینه سوم با پرس و جوی «افرادی که همسر آن‌ها بالای 90 سال دارند» روال زیر را باید دنبال کرد:

ابتدا باید همسر هر فردی را مطابق آنچه پیش از این نیز گفتیم شناسایی کنید. مثلاً "شیرین فرهاد" همسر "آرش کمانگیر" است، سپس بررسی می‌شود که سن "شیرین فرهاد" آیا بالای 90 سال است، اگر شرط برقرار بود، «آرش کمانگیر» فردی است که همسر آن بالای 90 سال است و در خروجی نمایش داده می‌شود. این روال باید برای تک تک اعضای جدول person انجام گردد. اگر فرض کنیم که پدر و مادری که دارای فرزند هستند، هر دو در قيد حیات و با هم در حال زندگی نیز هستند، می‌توان نتیجه‌های زیر را نیز از جدول person و parent استخراج نمود:

$$\Pi_{ppid}(parent) \rightarrow \frac{ppid}{\begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \end{array}}$$

شماره شناسایی افراد دارای فرزند (پدرها و مادرها)

شماره شناسایی افراد بدون فرزند:

شماره شناسایی افراد دارای فرزند - کل افراد = شماره شناسایی افراد بدون فرزند

$$[\Pi_{pid}(person)] - [\rho T(pid)(\Pi_{ppid}(parent))]$$

<u>pid</u>	<u>pid</u>	<u>pid</u>
1	1	
2	2	12
12	=	3
3		- 34
4		
34		

توجه: تغییر نام توسط عملگر ρ ، برای برقراری شروط سازگاری (تیترهای مشابه) در عمل تفاضل انجام گرفته است.

گزینه چهارم به راحتی با کسر کردن تاریخ تولد هر فرد از تاریخ روز و مقایسه حاصل با سن آن فرد قابل پاسخ‌گویی است. توجه داشته باشید که صفت سن چون پویا و در حال تغییر است (مشتق) اغلب در پایگاه داده ذخیره نمی‌شود.

- ۳۸- گزینه (۳) صحیح است.

گزینه اول اشتباه است: دو رابطه فرضی $R_1(A,B)$ و $R_2(A,B)$ و اجتماع آنها واضح است که A_3 (که اجتماع کلیدهای R_1 و R_2 است) کلید نیست.

R_1	
A	B
2	a
2	b

R_2	
A	B
1	x
2	y

R_3	
A	B
1	a
1	x
2	b
2	y

گزینه دوم اشتباه است: دو رابطه فرضی $R_1(A,B,C,D,E)$ و $R_2(D,E)$ را در نظر بگیرید. حاصل تقسیم $R_1 \div R_2$ یک رابطه است که ستون‌های آن شامل ستون‌هایی از R_1 است که در R_2 نیست:

$R_1 \div R_2 = R_3(A,B,C)$ در جدول R_3 ، ستون‌های B و A با هم کلید اصلی‌اند (با اضافه کردن صفت C به صفات A و B، مجموعه $\{A,B,C\}$ هر چند یک کلید است ولی کلید کمینه نیست). کلید اصلی از میان کلیدهای کاندید انتخاب می‌شود و کلید کاندید حتماً کمینه است.

گزینه سوم صحیح است: اشتراک دو رابطه، شامل رکوردهایی است که در هر دو رابطه بوده‌اند و بنابراین وابستگی‌های تابعی در هر دو رابطه را رعایت کرده‌اند. بنابراین هر یک از کلیدهای کاندید R_1 و R_2 باز هم یک کلید کاندید در R_3 هستند و می‌توانند به عنوان کلید اصلی در نظر گرفته شوند.

گزینه چهارم اشتباه است: دو رابطه فرضی $R_1(A,B)$ و $R_2(C,D)$ را در نظر بگیرید. در حاصل ضرب دکارتی آن دو، یعنی $R_1 \times R_2 = R_3(A,B,C,D)$ با توجه به وابستگی‌های تابعی موجود، مجموعه صفات $\{A,C\}$ کلید هستند. پس نیازی نیست تمام صفات را روی هم به عنوان کلید اصلی گرفت.

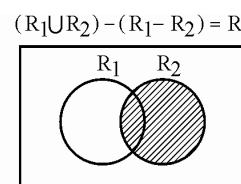
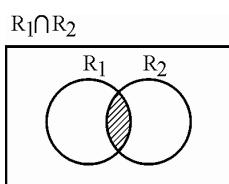
- ۳۹- گزینه (۳) صحیح است.

گزینه‌های اول و دوم روابط معروف بین عملگرهای \cap و $-$ هستند. گزینه چهارم هم تعریف عملگر Semiminus است.

گزینه سوم اشتباه است. ابتدا توجه کنید که $(R_1 - R_2)$ و $(R_2 - R_1)$ هیچ اشتراکی با هم ندارند. پس:

$$(R_1 - R_2) - (R_2 - R_1) = (R_1 - R_2)$$

حال با توجه به عدم تساوی قسمت‌های هاشور خورده در نمودارهای ون زیر معلوم می‌شود تساوی گزینه سوم برقرار نیست:



۴۰- گزینه (۱) صحیح است.

عبارت STU Times CRS جدول دانشجو را با جدول دروس ضرب دکارتی می‌کند. حاصل ضرب این ضرب دکارتی، جدولی است که در آن، هر دانشجو با تمام دروس (چه آن درس را گرفته باشد چه نگرفته باشد) ظاهر شده است. سپس عبارت [STID,CID] عمل پرتو را انجام می‌دهد و فقط دو ستون STID و CID را باقی می‌گذارد (برای این که عملگر MINUS امکان‌پذیر شود). در جدول SC، معلوم است هر دانشجو چه درسی را گرفته است. عملگر MINUS باعث می‌شود از جدول حاصل ضرب دکارتی، فقط رکوردهای هر دانشجو با دروسی که نگرفته است باقی بماند.

۴۱- گزینه (۱) صحیح است.

عبارت $\sigma_{pRel=1}(Rel)$ مجموعه سطرهایی از جدول Rel را می‌دهد که یک رابطه پدر - فرزندی را دارند. سپس با اعمال عملگر Π ، فقط ستون pID1 (یعنی شناسه فرزندان) از مجموعه فوق انتخاب می‌شود. یعنی کل عبارت سمت چپ Π ، مجموعه شناسه‌های فرزندان را به ما می‌دهد. این مجموعه با جدول Person با شرط $pID1 = pID$ پیوند می‌شود، پس، از جدول Person سطرهایی را به دست می‌آورد که شناسه آن‌ها (pID آن‌ها) برابر pID1 ها (یعنی شناسه‌های فرزندان) باشد. پس در نهایت سطرهایی از جدول Person به دست می‌آید که مربوط به فرزندان است.

۴۲- گزینه (۱) صحیح است.

عمل اتصال طبیعی روی دو جدول معادل است با انجام ضرب دکارتی روی آن دو جدول و سپس انجام عمل select با شرط برابری مقدار ستون‌های همنام دو جدول. بنابراین گزینه اول صحیح است. گزینه دوم اشتباه است، چون با انجام پیوند طبیعی بخشی از اطلاعات جداول حذف می‌شوند. در واقع، رکوردهایی از هر یک از جداول که رکورد متناظری در جدول دیگر ندارند حذف می‌شوند. گزینه سوم نیز اشتباه است، چون با توجه به توضیحات فوق برای گزینه دوم، برخی از اطلاعات از دست می‌روند.

توجه: منظور از عمل بازنی یا دودویی، یک عملگر با دو عملوند است.

۴۳- گزینه (۱) صحیح است.

دستور select در جبر رابطه‌ای یک شرط (یعنی یک یا چند عمل مقایسه) را روی سطرهای جدول بررسی می‌کند. هرگاه مقدار داده‌ای در یک عمل مقایسه برابر NULL باشد نتیجه مقایسه برابر UNKNOWN خواهد شد (یعنی TURE نمی‌شود) و در نتیجه، سطر مربوطه در خروجی ظاهر نمی‌شود. اما دستور projection (که ستون‌هایی از یک جدول را باقی می‌گذارد) با مقادیر داخل جدول کاری ندارد و بنابراین با NULL نیز مانند سایر مقادیر برحورд می‌کند. همچنین دستور UNION هم اگر مقدار NULL را در یکی از دو جدول بییند، این مقدار را (همانند مقادیر دیگر) در خروجی می‌آورد. پس گزینه‌های دوم و سوم عبارات درستی‌اند.

۴۴- گزینه (۳) صحیح است.

برای درک بهتر، جدول course، با مقادیر زیر را در نظر بگیرید:

course(Cname,TA)

Cname	TA
C ₁	T ₁
C ₁	T ₂
C ₂	T ₁
C ₃	T ₃

که براساس $\rho_{TA \rightarrow TA_1}$ و $\rho_{TA \rightarrow TA_2}$ الحاق دو جدول زیر را فقط بر روی ستون مشترک Cname به شکل زیر خواهیم داشت:

Cname	TA1	Cname	TA2	Cname	TA1	TA2		
C1	T1	C1	T1	C1	T1	T1		
C1	T2	C1	T2	C1	T1	T2		
C2	T1	x	C2	T1	=	C1	T2	T1
C3	T3		C3	T3		C1	T2	T2
						C2	T1	T1
						C3	T3	T3

در مسئله مطرح شده است که می خواهیم همه زوج های (TA1 و TA2) را به دست آوریم که TA1 و TA2 از هم متمایزند و با هم تفاوت مقداری دارند و حداقل در یک درس به عنوان کمک استاد با هم همکاری می کنند، مانند کمک استادهای T1 و T2 که هر دو در درس C1 به عنوان کمک استاد با هم همکاری می کنند. همچنین در سؤال مطرح شده است که هر زوج یک بار نوشته شود، به این معنی که (y و x) و (x و y) هم زمان در نتیجه آورده نشوند، مانند (T1 و T2) و (T2 و T1) در سطرهای دوم و سوم جدول حاصل از الحاق دو جدول. بنابراین با توجه به دو شرط مطرح شده در صورت سؤال، خروجی حاصل از الحاق دو جدول به شکل زیر خواهد بود:

Cname	TA1	TA2	Cname	TA1	TA2
TA1 , TA2 ✗	C1	T1	T1	C1	
مقادیر غیرمتمايز					
TA1 < TA2 ✓	C1	T1	T2		
مقادیر متمايز					
TA1 > TA2 ✗	C1	T2	T1		
مقادیر متمايز					
TA1 , TA2 ✗	C1	T2	T2		
مقادیر غیرمتمايز					
TA1 < TA2 ✗	C2	T1	T1		
مقادیر غیرمتمايز					
TA1>TA2 ✗	C3	T3	T3		
مقادیر غیرمتمايز					

توجه: برای برقراری شرط دوم که تأکید شده است که هر زوج با مقادیر متمايز یک بار نوشته شوند، باید شرط $TA1 < TA2$ یا $TA1 > TA2$ رعایت گردد، ما شرط $TA1 < TA2$ که در گزینه سوم هم آمده است، برای حل این مثال در نظر گرفته ایم.

عبارت گزینه اول هیچ شرطی روی TA ها نگذاشته است و بنابراین نه تنها زوج های متقارن را در خروجی می آورد بلکه هر کمک استاد را با خودش هم در خروجی نشان می دهد.

در عبارت گزینه دوم، شرط $TA1 \leftrightarrow TA2$ باعث می شود که یک کمک استاد با خودش در خروجی نیاید اما از نمایش زوج های متقارن جلوگیری نمی کند.

اما در عبارت گزینه سوم، شرط $TA1 < TA2$ باعث می شود فقط یکی از زوج های متقارن در خروجی ظاهر شود چون همواره اسم (یا کُد) دو کمک استاد با هم متفاوت خواهد بود. این شرط هم چنین از نمایش یک کمک استاد با خودش هم جلوگیری می کند.

۴۵- گزینه (۴) صحیح است.

در گزاره صورت سؤال، عمل نیم الحق (یا نیم پیوند) همان کار پیوند طبیعی (۲۴) را انجام می‌دهد با این تفاوت که در خروجی، فقط صفت‌های عملوند سمت چپ ظاهر می‌شوند. به عبارت دیگر، عبارت $r \times s$ سطرهای متناظر از r و s را به هم ربط می‌دهد اما در خروجی فقط ستون‌های r ظاهر می‌شوند. (منظور از «متناظر» بودن دو سطر از دو جدول این است که ستون‌های همنامشان در دو جدول مقادیر یکسانی داشته باشند).

در گزاره I: عبارت $(s \Delta X \cap Y)$ فقط ستون‌هایی از s که همنام با ستون‌های r باشند را باقی می‌گذارد. بنابراین با انجام پیوند طبیعی روی حاصل این عبارت با r ، همان نتیجه $r \times s$ حاصل می‌شود.

در گزاره II: عبارت $s \times r$ سطرهایی از r که سطرهای متناظری در s دارند را به ما می‌دهد. اما فقط ستون‌های r ظاهر می‌شوند. بنابراین پیوند طبیعی روی حاصل این عبارت با r مجددًا سطرهای متناظر از s و r به هم ربط داده می‌شوند اما ستون‌های s نیز در خروجی ظاهر می‌شوند.

در گزاره III: عبارت $r \times s$ سطرهایی از r که سطر متناظری در s دارند را به ما می‌دهد اما فقط ستون‌های r در خروجی می‌آیند. به طور مشابه، عبارت $s \times r$ سطرهایی از s که سطر متناظر در r دارند را به ما می‌دهد اما فقط ستون‌های s در خروجی می‌آیند. بنابراین با انجام پیوند طبیعی روی حاصل این دو عبارت، همان نتیجه $r \times s$ به دست می‌آید.

۴۶- گزینه (۳) صحیح است.

علمگرهاي ضرب دكارتی (\times) و پيوند طبیعی (۲۴) خاصیت جابه‌جایی (تقارنی) دارند. پس گزینه‌های اول و دوم صحیح‌اند. هم‌چنین در گزینه چهارم، ترتیب اعمال شرط‌های condition1 و condition2 تأثیری در نتیجه نهایی ندارند و هر دو معادل $(r \times s)$ هستند.

اما عملگر پرتو (Π) همواره خاصیت جابه‌جایی ندارند. توجه کنید که عبارت $\Pi_{list2}(\Pi_{list1}r)$ فقط در صورتی قابل انجام است که در آن صورت $list1 \subseteq list2$ استنایی ای که $list1 = list2$ باشد، بنابراین عبارت $\Pi_{list1}(\Pi_{list2}r)$ در حالت کلی و در همه شرایط قابل انجام نیست.

۴۷- گزینه (۱) صحیح است.

در این سؤال، باید شناسه سفارش مشتریانی را بیاییم که همه کالاهای گروه food را خریده‌اند. برای نوشتن جبر رابطه‌ای برای پرس‌وجوهایی مشابه این که در آنها کلمه «همه» آمده است مقسوم‌علیه (یعنی عبارت بعد از «همه») و مقسوم (یعنی عبارت قبل از «همه») را می‌نویسیم. (مقسوم باید شامل صفت‌های مقسوم‌علیه و صفت‌های خواسته شده در خروجی باشد). در نهایت، مقسوم را بر مقسوم‌علیه تقسیم می‌کنیم. به سادگی دیده می‌شود که رابطه‌های r و s به ترتیب مقسوم و مقسوم‌علیه هستند. بنابراین، پاسخ سؤال $r \div s$ خواهد بود.

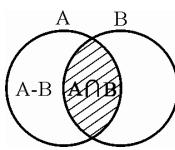
۴۸- گزینه (۲) صحیح است.

در بانک اطلاعات نامتمرکز، عملگر نیم‌پیوند، مانع از انتقال حجم زیادی از اطلاعات نامرتبط از یک سرور به سرور دیگر می‌شود.

- گزینه (۳) صحیح است.

عملگرهای اجتماع، ضرب دکارتی و تفاضل، جزء عملگرهای اصلی جبر رابطه‌ای هستند و عملگرهایی که با استفاده از عملگرهای اصلی قابل ایجاد هستند، عملگرهای فرعی هستند، مانند عملگر اشتراک و تقسیم. مثال:

$$\begin{array}{c} A \cap B = A - (A - B) = B - (B - A) \\ \text{---} \\ \text{---} \end{array}$$



با توجه به شکل مقابل، مشاهده می‌شود که با کنار گذاشتن ناحیه $A - B$ از ناحیه A ، قسمت باقی مانده برابر $A \cap B$ است.

- گزینه (۴) صحیح است.

پیوند طبیعی، عملگری است که دو رابطه را بر مبنای یک یا چند صفت مشترک به هم پیوند می‌دهد. در پیوند طبیعی دو رابطه R_1 و R_2 یعنی $R_1 \bowtie R_2$ اگر R_1 و R_2 صفت مشترکی نداشته باشند، $R_1 \bowtie R_2$ تبدیل به $R_1 \times R_2$ خواهد شد و اگر روابط R_1 و R_2 تماماً دارای صفات یکسانی باشند، آن‌گاه معادل $R_1 \bowtie R_2$ $R_1 \cap R_2$ خواهد شد.

مثال زیر گویای مطلب می‌باشد:

$$\begin{array}{c} \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline \end{array} \bowtie \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline & & \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline & & \\ \hline \end{array} \\ \text{---} \\ \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline \end{array} \cap \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline & & \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & 2 & 3 \\ \hline & & \\ \hline \end{array} \end{array}$$

- گزینه (۴) صحیح است.

عملگر ضرب دکارتی، تمام رکوردهای یک جدول را در کنار تمام رکوردهای یک جدول دیگر قرار می‌دهد و در خروجی نمایش می‌دهد. بنابراین ضرب دکارتی زمان و حافظه بیشتری نسبت به پیوند شرطی، طبیعی و نیم‌پیوند می‌خواهد.

- گزینه (۲) صحیح است.

همانطور که می‌دانید یک مجموعه با k^k عضو، دارای 2^k زیر مجموعه است که البته یکی از آن‌ها، مجموعه تهی است. بنابراین اگر رابطه‌ای دارای k خصیصه باشد، تعداد کل تصاویر (به معنی عملگر پرتو یا Π) آن، 2^k خواهد بود که اگر تصویر تهی آن را کنار بگذاریم برابر $1 - 2^k$ تصویر غیر تهی می‌شود.

مثال: جدول زیر را با سه ستون در نظر بگیرید، تعداد تصاویر غیر تهی برابر چیست؟

X	Y	Z

X	Y	Z	تصویر
0	0	0	تهی
0	0	1	Z
0	1	0	Y
0	1	1	YZ
1	0	0	X
1	0	1	XZ
1	1	0	XY
1	1	1	XYZ

که جدول بدون ستون وجود ندارد. \Rightarrow

بنابراین تعداد تصاویر غیرتهی برابر مقدار $7 = 1 - 2^3$ خواهد بود، که در یک عبارت کلی رابطه $1 - 2^k$ را

که k برابر تعداد ستون‌ها است، خواهیم داشت.

همچنین در یک راه حل دیگر داریم:

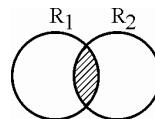
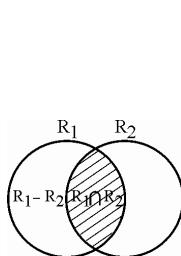
هر تصویر غیرتهی از جدول R شامل $1, 2, 3 \dots$ یا k صفت از جدول R است. بنابراین تعداد آنها برابر است با:

$$\binom{k}{1} + \binom{k}{2} + \binom{k}{3} + \dots + \binom{k}{k} = 2^k - 1$$

۵۳- گزینه (۴) صحیح است.

عملگرهای اجتماعی، ضرب دکارتی و تفاضل جزء عملگرهای اصلی جبر رابطه‌ای هستند و عملگرهایی که با استفاده از عملگرهای اصلی قابل ایجاد هستند، عملگرهای فرعی هستند، مانند عملگر اشتراک و تقسیم.

مثال: $R_1 \cap R_2 = R_1 - (R_1 - R_2) = R_2 - (R_2 - R_1)$



با توجه به شکل مقابل، مشاهده می‌شود که با کنار گذاشتن ناحیه $R_1 - R_2$ از ناحیه R_1 ، قسمت باقی مانده برابر $R_1 \cap R_2$ است.

۵۴- گزینه (۳) صحیح است.

جدول زیر را در نظر بگیرید:

X	Y
1	5
3	2
3	4

X	Y
1	2
3	2
5	6
3	7

جدول a

جدول b

توجه: دقت کنید که هر طرف رابطه به تنهایی باید درست در نظر گرفته شود و برابری آن با طرف درست مقابله مقایسه گردد، در واقع هر یک از طرفین مقایسه قادر خطا نحوی هستند. برای مثال شروط

سازگاری برای عملگر تفاضل به معنای ستون‌های مشابه، باید در نظر گرفته شود.
گزینه اول درست است.

$$\sigma_p(a - b) = \sigma_p(a) - b$$

مقدار شرط p را برای جدول a برابر $X = 3$ در نظر بگیرید، بنابراین داریم:

$$\sigma_{X=3}(a - b) = \sigma_{X=3}(a) - b$$

$$\sigma_{X=3} \left(\begin{array}{c|c} X & Y \\ \hline 1 & 5 \\ 3 & 2 \\ 3 & 4 \end{array} \right) - \left(\begin{array}{c|c} X & Y \\ \hline 1 & 2 \\ 3 & 2 \\ 5 & 6 \\ 3 & 7 \end{array} \right) = \sigma_{X=3} \left(\begin{array}{c|c} X & Y \\ \hline 1 & 5 \\ 3 & 2 \\ 3 & 4 \end{array} \right) - \left(\begin{array}{c|c} X & Y \\ \hline 1 & 2 \\ 3 & 2 \\ 5 & 6 \\ 3 & 7 \end{array} \right)$$

گزینه دوم درست است.

$$\sigma_p(a - b) = \sigma_p(a) - \sigma_p(b)$$

مقدار شرط p را برای جدول a و b بیاید، بنابراین داریم:

$$\sigma_{X=3} \begin{pmatrix} X & Y \\ \hline 1 & 5 \\ 3 & 2 \\ 3 & 4 \end{pmatrix} - \begin{pmatrix} X & Y \\ \hline 1 & 2 \\ 3 & 2 \\ 5 & 6 \\ 3 & 7 \end{pmatrix} = \sigma_{X=3} \begin{pmatrix} X & Y \\ \hline 1 & 5 \\ 3 & 2 \\ 3 & 4 \end{pmatrix} - \sigma_{X=3} \begin{pmatrix} X & Y \\ \hline 1 & 2 \\ 3 & 2 \\ 5 & 6 \\ 3 & 7 \end{pmatrix}$$

\downarrow \downarrow

$$\begin{array}{c} X \mid Y \\ \hline 3 \mid 4 \end{array} \qquad \qquad \qquad \begin{array}{c} X \mid Y \\ \hline 3 \mid 4 \end{array}$$

گزینه سوم نادرست است.

عبارت $\sigma_p(a \cap b)$, زیر مجموعه‌ای از عناصر موجود در $a \cap b$ را به ما می‌دهد, اما $a - b$, عناصری از a مدد که در b و دار نباشند را می‌دهد.

$$\sigma_p(a \cap b) = a - b$$

مقدار شرط p را برای جدول a و b برای $X = 3$ در نظر بگیرید، بنابراین داریم:

$$\sigma X = 3 \left(\begin{array}{c|c} X & Y \\ \hline 1 & 5 \\ 3 & 2 \\ 3 & 4 \end{array} \right) \cap \left(\begin{array}{c|c} X & Y \\ \hline 1 & 2 \\ 3 & 2 \\ 5 & 6 \\ 3 & 7 \end{array} \right) \neq \left(\begin{array}{c|c} X & Y \\ \hline 1 & 5 \\ 2 & -3 \\ 4 & 5 \\ 3 & 7 \end{array} \right)$$

گزینه چهارم درست است.

$$a - b \neq b - a$$

مثال: عبارت گزینه سوم مطابق اصل جابه‌جاناپذیری عملگر تفاضل، درست است.

۵۵- گزینه (۴) صحیح است.

سه جدول students، take و courses با مقادیر زیر را در نظر بگیرید:

<u>sid</u>	lname	<u>sid</u>	<u>cid</u>	grade	<u>cid</u>	cname	dept
s1	sn1	s1	c1	12	c1	cn1	CS
s2	sn2	s1	c2	14	c2	cn2	CS
s3	sn3	s2	c1	16	c3	cn3	CE
s4	sn4	s3	c3	18	c4	cn4	IT
s5	sn5	s4	c4	20			

students جدول take جدول courses جدول

گزینه اول نادرست است. جدول take که می‌گوید هر دانشجویی چه درسی را اخذ کرده و چه نمره‌ای گرفته است، مقسوم (مؤلفه اول) است و جدول courses با شرط «دروس ارائه شده در دانشکده CS» مقسوم عليه (مؤلفه دوم) است. مطابق تعریف عملگر تقسیم در جبر رابطه‌ای، باید تمام خصیصه‌های مقسوم عليه عملگر تقسیم در مقسوم باشد. به عبارت دیگر مجموعه صفات مقسوم عليه، زیر مجموعه، مجموعه صفات مقسوم باشد، اما همانطور که در این گزینه مشاهده می‌شود، صفات cname و dept در مقسوم عليه وجود دارند اما در مقسوم وجود ندارند. بنابراین اجرای عملگر تقسیم در گزینه اول، به دلیل وجود خطای نحوی امکان‌پذیر نیست.

گزینه دوم نادرست است. مطابق تعریف عملگر تقسیم در جبر رابطه‌ای، اجرای عملگر تقسیم در گزینه دوم امکان‌پذیر است، اما خروجی مطرح شده در صورت سؤال را استخراج نمی‌کند.

$$\Pi_{sid} \left[\begin{array}{c} \text{take} \\ \downarrow \end{array} \right] \div \left[\begin{array}{c} \Pi_{cid} \left(\sigma_{dept='cs'}(\text{courses}) \right) \\ \downarrow \end{array} \right]$$

<u>sid</u>	<u>cid</u>	grade
s1	c1	12
s1	c2	14
s2	c1	16
s3	c3	18
s4	c4	20

$$\Pi_{sid} \left[\begin{array}{c|c} sid & grade \\ \hline & \end{array} \right] \rightarrow \underline{sid}$$

توجه: به ازای سطر (s1,12) مقدار c1 و هم c2 برقرار نیست،

همچنین به ازای سطر (s_{1,14}) مقدار c_1 و هم c_2 برقرار نیست، همچنین به ازای سطر (s_{2,16}) مقدار c_1 و هم c_2 برقرار نیست، همچنین به ازای سطر (s_{3,18}) مقدار c_1 و هم c_2 برقرار نیست، و در نهایت به ازای سطر (s_{4,20}) مقدار c_1 و هم c_2 برقرار نیست، نتیجه اینکه حاصل تقسیم تهی خواهد بود.

گزینه چهارم درست است. اگر r و s دو جدول (رابطه) به ترتیب با مجموعه ستون‌های R و S باشند، آنگاه $r \div s$ به صورت زیر قابل تعریف است اگر و فقط اگر $S \subseteq R$ باشد.

$$r \div s = \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times S) - r)$$

مثال: دو جدول زیر را در نظر بگیرید:

a	b		b
5	7		1
5	1		2
4	2	جدول s	
4	3		
5	2		

جدول r

$$r = \text{مجموعه ستون‌های جدول } r = \{a, b\}$$

$$s = \text{مجموعه ستون‌های جدول } s = \{b\}$$

$$R - S = \{a, b\} - \{b\} = \{a\}$$

توجه: می‌توان به جای مؤلفه r در انتهای رابطهٔ فوق از مؤلفه $\Pi_{R-S}(r)$ نیز استفاده نمود، زیرا دو مؤلفه برابر هستند.

توجه: مطابق تعریف عملگر تقسیم، ستون‌های حاصل از تقسیم دو جدول برابر تفاصل ستون‌های دو جدول است. همچنین سطرهایی از جدول r در خروجی ظاهر می‌گردد که تمام اعضای جدول s را در کنار خود داشته باشد. بنابراین داریم:

a	b	\div	b	=	a
5	7		1		5
5	1		2		
4	2	جدول s			
4	3				
5	2				

جدول r

همچنین براساس رابطه تعریف شده داریم:

$$\begin{aligned}
 r \div s &= \underbrace{\Pi_{R-S}(r)}_{\text{کل حالت‌های ممکن}} - \underbrace{\Pi_{R-S}((\Pi_{R-S}(r) \times s) - r)}_{\text{کل حالت‌های ناقص}}
 \\
 &\quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\
 &\quad \frac{a}{5} \times \frac{b}{2} \\
 &\quad \downarrow \quad \downarrow \\
 &\quad \begin{array}{c|c} a & b \\ \hline 5 & 1 \\ 5 & 2 \\ 4 & 1 \\ 4 & 2 \end{array} - \begin{array}{c|c} a & b \\ \hline 5 & 7 \\ 5 & 1 \\ 4 & 2 \\ 4 & 3 \end{array} \\
 &\quad \downarrow \quad \downarrow \\
 &\quad \text{کل حالت‌های ممکن} \\
 &\quad \downarrow \\
 &\quad \Pi_{R-S}(4 \mid 1) \\
 &\quad \downarrow \quad \downarrow \\
 &\quad \text{کل حالت‌های ناقص} \\
 &\quad \downarrow \\
 &\quad \frac{a}{5} - \frac{a}{4} \\
 &\quad \downarrow \quad \downarrow \\
 &\quad \text{کل حالت‌های موجود} \\
 &\quad \downarrow \\
 &\quad \frac{a}{5}
 \end{aligned}$$

کل حالت‌های کامل

توجه: یکی از عملگرهای اصلی جبر رابطه‌ای، عملگر تفاضل است. اگر R و S دو رابطه باشند، منظور از $R - S$ مجموعه کلیه سطرهایی است که عضو R هستند، اما در S حضور ندارند.

توجه: در جبر رابطه‌ای تفاضل هر دو رابطه دلخواه امکان‌پذیر نیست. تفاضل دو رابطه در جبر رابطه‌ای زمانی امکان‌پذیر است که اگر و فقط اگر شروط سازگاری در مورد آنها برقرار باشد.

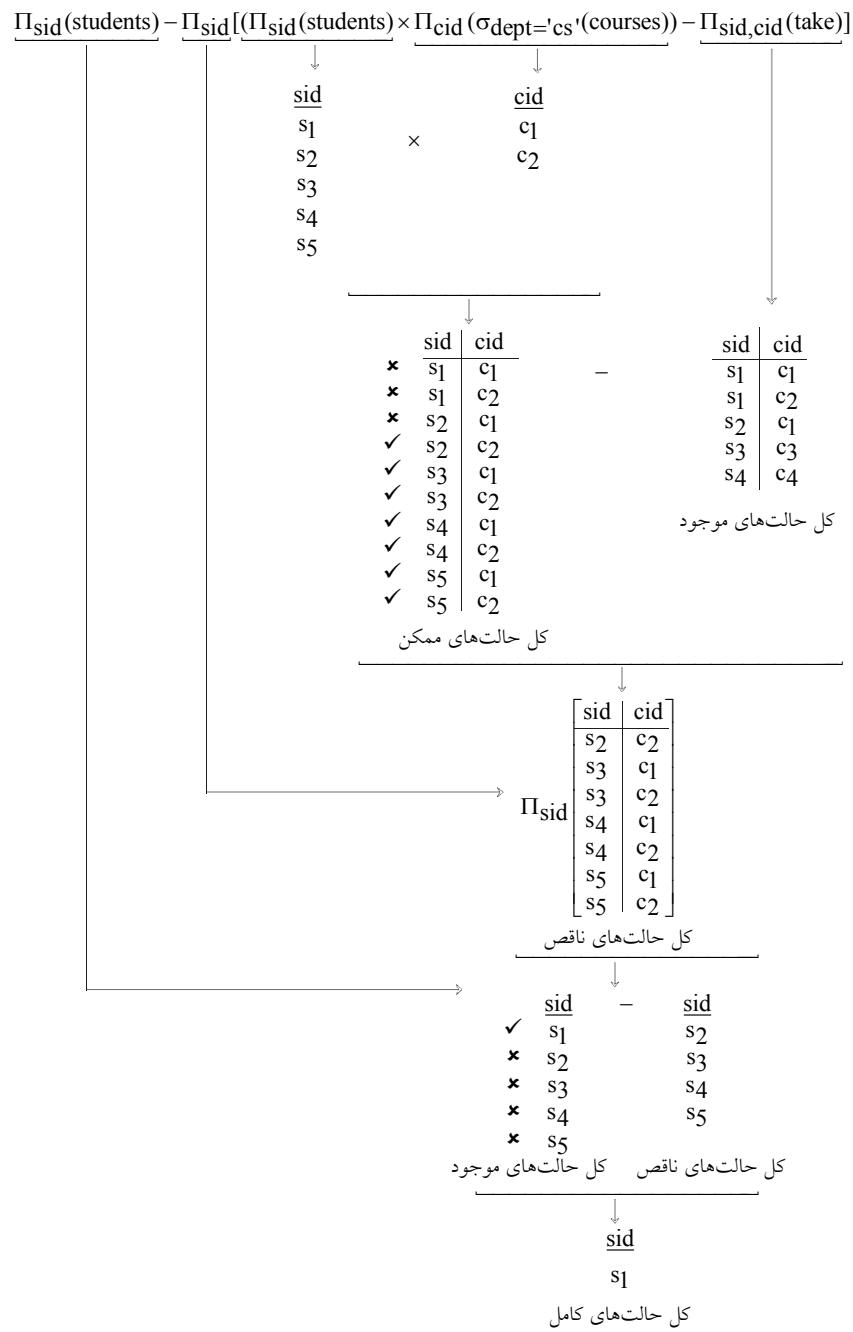
شروط سازگاری: در جبر رابطه‌ای دو شرط به عنوان شروط سازگاری مطرح است:

شرط اول: تعداد ستون‌های دو جدول یکسان باشد، به عبارت دیگر دو جدول هم درجه باشند.

شرط دوم: نوع یا دامنه ستون‌های متناظر در دو جدول یکسان باشد.

توجه: رابطه تقسیم قبل را مجدداً مشاهده نماید، همانطور که واضح است، هرگاه از عملگر تفاضل استفاده شده است در دو طرف عملگر، شروط سازگاری رعایت شده است.

توجه: براساس مطالب فوق، رابطه موجود در گزینه چهارم را مطابق جداول مطرح شده در نظر بگیرید:



توجه: مشاهده می‌شود که شماره دانشجویی s₁, همه دروس ارائه شده در دانشکده 'cs' را اخذ کرده است.

گزینه سوم نادرست است. زیرا مؤلفه $\Pi_{\text{sid}}(\text{students})$ را نسبت به گزینه چهارم برای پاسخ‌دهی به

پرس‌وجوی موردنظر ندارد.
در واقع خروجی گزینه سوم به صورت زیر است:

sid
s2
s3
s4
s5

يعنى شماره دانشجویانی را استخراج می‌کند که همه دروس ارائه شده در دانشکده 'cs' را اخذ نکرده‌اند.

۵۶ - گزینه (۴) صحیح است.

مطابق فرض صورت سؤال، دستور $\rho_y(x)$ ، ایجاد یک کپی (یک نام مستعار) از رابطه x با نام y است.
جدول زیر را در نظر بگیرید:

E#	Salary
S1	10
S2	20
S3	30

جدول Emp

پرس‌وجوی مطرح شده در صورت سؤال به صورت زیر است:

$\Pi_{\text{salary}}(\text{Emp}) - \Pi_{\text{first.salary}}(\sigma_{\text{first.salary} > \text{second.salary}}(\rho_{\text{first}}(\text{Emp}) \times \rho_{\text{second}}(\text{Emp}))$

توجه: یکی از عملگرهای اصلی جبر رابطه‌ای، عملگر تفاضل است. اگر R و S دو رابطه باشند، منظور از $R-S$ مجموعه کلیه سطرهایی است که عضو R هستند، اما در S حضور ندارند.

توجه: در جبر رابطه‌ای تفاضل هر دو رابطه دلخواه امکان‌پذیر نیست. تفاضل دو رابطه در جبر رابطه‌ای زمانی امکان‌پذیر است که اگر و فقط اگر شروط سازگاری در مورد آن‌ها برقرار باشد.

شرط سازگاری: در جبر رابطه‌ای دو شرط به عنوان شروط سازگاری مطرح است:

شرط اول: تعداد ستون‌های دو جدول یکسان باشد به عبارت دیگر دو جدول هم درجه باشند.

شرط دوم: نوع یا دامنه ستون‌های دو جدول یکسان باشد.

توجه: در پرس‌وجوی مطرح شده در هر دو طرف عملگر تفاضل، شروط سازگاری رعایت شده است.

براساس پرس‌وجوی مطرح شده در صورت سؤال داریم:

در ابتدا توسط عبارت $\Pi_{\text{salary}}(\text{Emp})$ در سمت چپ عملگر تفاضل حقوق کلیه کارمندان از جدول Emp به صورت زیر استخراج می‌گردد:

salary
10
20
30

همچنین براساس بخش سمت راست عملگر تفاضل داریم:

عبارت $\rho_{\text{first}}(\text{Emp}) \times \rho_{\text{second}}$ ضرب دکارتی جدول Emp در خودش است. حاصل این ضرب دکارتی، جدولی است که در آن، رکورد اطلاعاتی هر کارمند جدول first در کنار رکوردهای اطلاعاتی همه کارمندان جدول second قرار می‌گیرد.

first.E #	first.salary	second.E #	second.salary
S ₁	10	S ₁	10
S ₁	10	S ₂	20
S ₁	10	S ₃	30
S ₂	20	S ₁	10
S ₂	20	S ₂	20
S ₂	20	S ₃	30
S ₃	30	S ₁	10
S ₃	30	S ₂	20
S ₃	30	S ₃	30

سپس شرط $\sigma_{\text{first.salary} > \text{second.salary}}$ کارمند جدول first حقوقش حداقل از یک نفر دیگر بیشتر باشد را انتخاب می‌کند (یعنی کارمندهایی که حقوقشان کمترین حقوق نیست). بنابراین، پس از اعمال شرط داریم:

first.E #	first.salary	second.E #	second.Salary
S ₂	20	S ₁	10
S ₃	30	S ₁	10
S ₃	30	S ₂	20

در ادامه، توسط عملگر $\Pi_{\text{first.salary}}$ حقوق این نوع از کارمندان، به صورت زیر استخراج می‌گردد.

$$\begin{array}{c} \text{first.salary} \\ \hline 20 \\ 30 \end{array}$$

توجه: در جبر رابطه‌ای، سطرهای تکراری در خروجی قرار نمی‌گیرند. در انتها، با انجام عملگر تفاضل، حقوق کارمندان فوق، از مجموعه حقوق کلیه کارمندان کنار گذاشته می‌شوند. آنچه باقی می‌ماند، حقوق کارمند یا کارمندانی است که کمتر از همه حقوق می‌گیرند. به عبارت دیگر مینیمم حقوق کارمندان استخراج می‌گردد.

خروجی نهایی پرس‌وجوی مطرح شده به صورت زیر است:

$$\begin{array}{c} \text{salary} - \text{first.salary} = \text{salary} \\ \hline 10 - 20 = 10 \\ 20 - 30 \\ 30 \end{array}$$

توجه: اگر شرط موجود در پرس‌وجوی مطرح شده عکس گردد، آنگاه گزینه سوم ایجاد می‌گردد.

۵۷- گزینه (۲) صحیح است.

عملگر الحق طبیعی

عملگر الحق طبیعی سطرهایی از دو رابطه را بر مبنای ستون‌های مشترک (هم نام و هم دامنه)، که مقدارشان در همه ستون‌های مشترک مساوی است، به هم پیوند می‌زند و در خروجی نمایش می‌دهد، ستون‌های مشترک فقط یکبار در خروجی ظاهر می‌شوند. به بیان دیگر ستون‌های جدول خروجی، برابر اجتماع ستون‌های جدول‌های ورودی هستند.

مثال: یک ستون مشترک.

a	b		b	c	
2	5	✖	6	1	
8	1		5	2	
4	2		4	7	
7	5		8	9	
	1			6	

مثال: دو ستون مشترک.

a	b	c		b	c	d		a	b	c	d
2	4	1	✖	2	3	6		3	1	4	2
3	1	4		1	6	1					
2	5	2		1	4	2					
6	1	1		4	2	1					

عملگر الحق خارجی راست (right outer join)

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

به بیان دیگر ابتدا دو جدول ورودی با یکدیگر الحق طبیعی می‌شوند، سپس سطرهایی از جدول سمت راست که به دلیل عدم تطابق در الحق طبیعی شرکت نکرده‌اند، به حاصل الحق اضافه می‌گردد و برای ستون‌های غیرمشترک جدول سمت چپ مقدار NULL درج می‌گردد.

توجه: الحق خارجی راست میان دو رابطه، با نماد \bowtie نشان داده می‌شود.

مثال: یک ستون مشترک.

a	b		b	c		a	b	c
2	5		✖ 6	1		2	5	2
8	1		✓ 5	2		8	1	6
4	2	✖	✖ 4	7	=	7	5	2
7	5		✖ 8	9		NULL	6	1
			✓ 1	6		NULL	4	7
						NULL	8	9

مثال: دو ستون مشترک.

a	b	c		b	c	d		a	b	c	d	
2	4	1		✗	2	3	6	3	1	4	2	
3	1	4	⇒	✗	1	6	1	=	NULL	2	3	6
2	5	2		✓	1	4	2	NULL	1	6	1	
6	1	1		✗	4	2	1	NULL	4	2	1	

تعریف کلید خارجی

کلید خارجی برای ارتباط میان جداول مورد استفاده قرار می‌گیرد. اگر ستون(هایی) در یک جدول به عنوان کلید خارجی تعریف شود، این ستون(ها) باید در همان جدول یا جدول دیگری به عنوان کلید کاندید (اصلی یا فرعی) وجود داشته باشد.

توجه: صرف وجود ستون مشترک، به معنای وجود کلید خارجی در یک جدول نیست، بلکه برای یک جدول، کلید خارجی باید تعریف گردد.

توجه: کلید خارجی در صورت وجود برای یک جدول باید هم نوع با کلید کاندید باشد، اما الزامی ندارد که با کلید کاندید هم نام هم باشد.

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

توجه: یک کلید خارجی، در یک رابطه هیچگاه نباید ارجاع NULL داشته باشد، این مسئله را به عنوان یک قانون جامعیتی در مدل رابطه‌ای می‌شناسیم و آن را قانون جامعیت ارجاعی می‌نامیم.

مثال:

جدول S			جدول SP			جدول P		
کلید کاندید			کلید خارجی			کلید کاندید		
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	C3	S2	P1	30	P3	Pn3	Blue
NULL ←			S4	P3	40			

توجه: S#, در جدول S، کلید کاندید است، پس عناصرش یکتا است.

توجه: S# در جدول SP، به عنوان کلید خارجی، خاصیت کلیدی ندارد، پس ممکن است، مقدار تکراری داشته باشد.

توجه: درج رکورد (S4,P3,40) در جدول SP، غیرمجاز و غیرممکن است، زیرا سبب ارجاع NULL می‌گردد.

مثال: فرض کنید در یک نوع پیاده‌سازی کشف ارجاع NULL، قبل از هر بار درج رکورد در جدول SP، الحاق طبیعی دو جدول به صورت زیر اجرا گردد و تعداد سطرهای جواب حاصل شمارش گردد. بنابراین

داریم:

S#	Sname	city	S#	P#	QTY	S#	Sname	city	P#	QTY
S1	Sn1	C1	S1	P1	10	S1	Sn1	C1	P1	10
S2	Sn2	C2	S1	P2	20	S1	Sn2	C1	P2	20
S3	Sn3	C3	S2	P1	30	S2	Sn2	C2	P1	30

جدول *S*جدول *SP*

توجه: تعداد سطرهای پرس‌وجوی حاصل برابر ۳ می‌باشد.

حال اگر پس از هر بار درج رکورد در جدول *SP* (مثلاً درج رکورد $(S_4, P_3, 40)$)، الحاق خارجی راست به صورت زیر اجرا گردد و تعداد سطرهای حاصل از این پرس‌وجو بیشتر از تعداد سطرهای حاصل از الحاق طبیعی باشد، این نتیجه بدین معنی خواهد بود که در درج اخیر ارجاع NULL رخ داده است و باید از درج رکورد مورد نظر در جدول *SP*، ممانعت به عمل آید.

S#	Sname	city	S#	P#	QTY	S#	Sname	city	P#	QTY
S1	Sn1	C1	S1	P1	10	S1	Sn1	C1	P1	10
S2	Sn2	C2	S1	P2	20	S1	Sn1	C1	P2	20
S3	Sn3	C3	S2	P1	30	S2	Sn2	C2	P1	30
NULL	←		S4	P3	40	S4	NULL	NULL	P3	40

توجه: تعداد سطرهای پرس‌وجوی حاصل برابر ۴ است. پس ارجاع NULL رخ داده است.

براساس جدول مطرح شده در صورت سؤال، جداول *course* و *prereg* را با مقادیر زیر در نظر بگیرید:

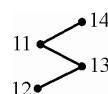
course-id	title	units	کلید خارجی	
			course-id	prereg-id
11	t1	2	13	11
12	t2	2	13	12
13	t3	3	14	11
14	t4	3		

جدول *course*جدول *prereg*

توجه: یک درس مانند ۱۳ می‌تواند دروس ۱۱ و ۱۲ را به عنوان پیش‌نیاز، نیاز داشته باشد و یک درس مانند

۱۱، می‌تواند پیش‌نیاز دروس ۱۳ و ۱۴ باشد.

به شکل زیر توجه نمایید:



پیاده‌سازی دو جدول *course* و *prereg* در SQL می‌تواند به صورت زیر باشد.

```
Create Table course
(
course-id int,
title nchar (10),
units int,
Primary key (course-id)
)
```

```
Create Table Prereg
(
course-id int,
prereg-id int,
Primary key (course-id, prereg-id),
Foreign key(course-id) references course(course-id),
Foreign key (prereg-id) references course (course-id)
)
```

توجه: مجموعه اعداد موجود در جدول `prereg` چه ستون `course-id` و چه ستون `prereg-id` باید همواره زیر مجموعه ستون `course-id` به عنوان کلید کاندید از جدول `course` باشد، این موضوع در دو خط آخر دستورات ایجاد جدول `prereg` بیان شده است.
جدول `course` و `prereg` را مجدداً در نظر بگیرید:

<i>course</i>			<i>prereg</i>	
<u>course - id</u>	title	units	<u>course - id</u>	prereg - id
11	t ₁	2	13	11
12	t ₂	2	13	12
13	t ₃	3	14	11
14	t ₄	3	NULL ←	16
				15

مثال: فرض کنید در یک نوع پیاده‌سازی کشف ارجاع `NULL`، قبل از هر بار درج رکورد در جدول `prereg` دو آزمون زیر به صورت زیر اجرا گردد و تعداد سطرهای جواب حاصل شمارش گردد. بنابراین داریم:
آزمون اول:

$\text{course} \bowtie (\Pi_{\text{course}-\text{id}}(\text{prereg}))$

<u>course - id</u>	title	units	\bowtie	<u>course - id</u>	<u>course - id</u>	title	units
11	t ₁	2		13	13	t ₃	3
12	t ₂	2		14	14	t ₄	3
13	t ₃	3					
14	t ₄	3					

course

توجه: تعداد سطرهای پرس‌وجوه حاصل برابر ۲ است.

حال اگر پس از هر بار درج رکورد در جدول `prereg` (مثلاً درج رکورد (15 و 16)، الحق خارجی راست به صورت زیر اجرا گردد و تعداد سطرهای حاصل از این پرس‌وجوه بیشتر از تعداد سطرهای حاصل از الحق طبیعی باشد، این نتیجه بدین معنی خواهد بود که در درج اخیر ارجاع `NULL` رخ داده است و باید از درج رکورد مورد نظر در جدول `prereg`، ممانعت به عمل آید.

$\text{course} \bowtie\!\!\!= (\Pi_{\text{course}-\text{id}}(\text{prereg}))$

course-id	title	units	course-id	course-id	title	units
11	t ₁	2	13	13	t ₃	3
12	t ₂	2	14	14	t ₄	3
13	t ₃	3	NULL ← 16	16	NULL	NULL
14	t ₄	3				

جدول course

توجه: تعداد سطرهای پرس‌وجوی حاصل برابر 3 است. پس ارجاع NULL رخ داده است.

نتیجه: پس آزمون اول ارجاع NULL را کشف کرد.

آزمون دوم:

$$\text{course} \bowtie (\rho_t(\text{course-id})(\Pi_{\text{prereg-id}}(\text{prereg})))$$

توجه: دستور نام‌گذاری (ρ_t)، جدول خروجی دستور ($\Pi_{\text{prereg-id}}$) (prereg) را برابر t و نام تنها ستون آن را برابر course-id قرار می‌دهد.

این تغییر نام برای فراهم‌سازی امکان اجرای عملگر الحق طبیعی (دارا بودن ستون مشترک هم نام و هم نوع) انجام شده است.

course-id	title	units	course-id	course-id	title	units
11	t ₁	2	11	11	t ₁	2
12	t ₂	2	12	12	t ₂	2
13	t ₃	3				
14	t ₄	3				

توجه: تعداد سطرهای پرس‌وجوی حاصل برابر 2 است.

حال اگر پس از هر بار درج رکورد در جدول prereg (مثلاً درج رکورد (15 و 16))، الحق خارجی راست به صورت زیر اجرا گردد و تعداد سطرهای حاصل از این پرس‌وجو بیشتر از تعداد سطرهای حاصل از الحق طبیعی باشد، این نتیجه بدین معنی خواهد بود که در درج اخیر ارجاع NULL رخ داده است و باید از درج رکورد مورد نظر در جدول prereg، ممانعت به عمل آید.

$$\text{course} \bowtie (\rho_t(\text{course-id})(\Pi_{\text{prereg-id}}(\text{prereg})))$$

course-id	title	units	course-id	course-id	title	units
11	t ₁	2	11	11	t ₁	2
12	t ₂	2	12	12	t ₂	2
13	t ₃	3	NULL ← 15	15	NULL	NULL
14	t ₄	3				

توجه: تعداد سطرهای پرس‌وجوی حاصل برابر 3 است. پس ارجاع NULL رخ داده است.

نتیجه: پس آزمون دوم نیز ارجاع NULL را کشف کرد.

توجه: اگر حداقل یکی از آزمون‌های اول و دوم موفق به کشف ارجاع NULL گردد، باید از درج رکورد مورد نظر در جدول *prereg*، ممانعت به عمل آید.

توجه: اگر پرس‌وجوی مطرح شده در گزینه دوم اجرا گردد، مطابق تعریف الحق خارجی، الحق بر روی ستون مشترک یعنی *course-id* انجام می‌گردد، در این حالت پرس‌وجو فقط قادر خواهد بود آزمون نوع اول مطرح شده را انجام دهد و فقط خطاهای درج منجر به ارجاع NULL را در ستون *course-id* از جدول *prereg* کشف کند. در صورت سؤال هم گفته شده است که کدام گزینه می‌تواند وجود مشکلی در اطلاعات ثبت شده در سیستم را کشف کند، بنابراین گزینه دوم درست است.

بر اساس الحق طبیعی دو جدول *prereg* و *course* داریم:

<i>course-id</i>	<i>title</i>	<i>units</i>	<i>course-id</i>	<i>prereg-id</i>	<i>course-id</i>	<i>prereg-id</i>	<i>title</i>	<i>units</i>
11	t ₁	2	13	11	13	11	t ₃	3
12	t ₂	2	13	12	13	12	t ₃	3
13	t ₃	3	14	11	14	11	t ₄	3
14	t ₄	3	NULL	16	15			

جدول *course*جدول *prereg*

توجه: تعداد سطرهای پرس‌وجوی حاصل برابر 3 است.

حال اگر پس از هر بار درج رکورد در جدول *prereg* (مثلاً درج رکورد (15 و 16)) الحق خارجی راست به صورت زیر اجرا گردد و تعداد سطرهای حاصل از این پرس‌وجو بیشتر از تعداد سطرهای حاصل از الحق طبیعی باشد، این نتیجه بدین معنی خواهد بود که در درج اخیر ارجاع NULL رخ داده است و باید از درج رکورد مورد نظر در جدول *prereg*، ممانعت به عمل آید.

بر اساس الحق خارجی راست دو جدول *prereg* و *course* داریم:

<i>course-id</i>	<i>title</i>	<i>units</i>	<i>course-id</i>	<i>prereg-id</i>	<i>course-id</i>	<i>prereg-id</i>	<i>title</i>	<i>units</i>
11	t ₁	2	13	11	13	11	t ₃	3
12	t ₂	2	13	12	13	12	t ₃	3
13	t ₃	3	14	11	14	11	t ₄	3
14	t ₄	3	NULL	16	15	16	NULL	NULL

جدول *course*جدول *prereg*

توجه: تعداد سطرهای پرس‌وجوی حاصل برابر 4 است. پس ارجاع NULL رخ داده است.

توجه: گزینه دوم مطرح شده در صورت سؤال، قادر به کشف ارجاع NULL در ستون *prereg-id* در جدول *prereg* نیست.

بر اساس الحق طبیعی دو جدول *prereg* و *course* داریم:

course - id	title	units	course - id	prereg - id	course - id	prereg - id	title	units
11	t ₁	2	13	11	13	11	t ₃	3
12	t ₂	2	13	12	13	12	t ₃	3
13	t ₃	3	14	11	13	15	t ₃	3
14	t ₄	3	NULL	13	15	14	t ₄	3

جدول course

جدول prereg

توجه: تعداد سطرهای پرس‌وجوی حاصل برابر 4 است.

حال اگر پس از هر بار درج رکورد در جدول prereg (مثلاً رکورد (15 و 13)) الحاق خارجی راست به صورت زیر اجرا گردد و تعداد سطرهای حاصل از این پرس‌وجو بیشتر از تعداد سطرهای حاصل از الحاق طبیعی باشد. این نتیجه بدین معنی خواهد بود که در درج اخیر ارجاع NULL رخ داده است و باید از درج رکورد مورد نظر در جدول prereg ممانعت به عمل آید.

بر اساس الحاق خارجی راست دو جدول prereg و course داریم:

course - id	title	units	course - id	prereg - id	course - id	prereg - id	title	units
11	t ₁	2	13	11	13	11	t ₃	3
12	t ₂	2	13	12	13	12	t ₃	3
13	t ₃	3	14	11	13	15	t ₃	3
14	t ₄	3	NULL	13	15	14	t ₄	3

جدول course

جدول prereg

توجه: تعداد سطرهای پرس‌وجوی حاصل باز هم برابر 4 شد، در حالی که ارجاع NULL به واسطه مقدار 15 در ستون prereg-id از جدول prereg وجود دارد، دقت کنید که مطابق تعریف کلید خارجی برای عدم وجود ارجاع NULL، باید مقادیر هر یک از ستون‌های course-id و prereg-id از جدول course مجموع مقادیر ستون course-id از جدول prereg باشد.

اما مطابق آنچه گفتیم، پرس‌وجوی مطرح شده در گزینه دوم قادر به کشف ارجاع NULL در ستون prereg-id از جدول prereg نیست.

- گزینه (۳) صحیح است.

جداویل زیر را در نظر بگیرید:

mem-no	name	age	mem-no	isbn	date	isbn	...	publisher
m1	mn1	21	m1	is1	11	is1		McGraw-Hill
m2	mn2	22	m1	is2	12	is2		McGraw-Hill
m3	mn3	23	m2	is1	13	is3		Penguin
Member			m2	is2	14	is4		Penguin
Borrowed			m2	is3	15	Book		

مطابق پرس و جوی مطرح شده در صورت سوال داریم:

«نام تمامی اعضاي را بیابید که همه کتاب‌های منتشرشده توسط McGraw-Hill را به امانت برده‌اند.»

توجه: کلمه «تمامی» در پرس و جوی مطرح شده در صورت سوال بی‌تأثیر است، چه این کلمه باشد و چه نباشد، تمامی اعضاي که همه کتاب‌های منتشرشده توسط McGraw-Hill را به امانت برده‌اند، استخراج می‌گردد.

پرس و جوی فوق توسط عملگر تقسیم در جبر رابطه‌ای قابل پیاده‌سازی است.

مطابق تعریف عملگر تقسیم در جبر رابطه‌ای، باید تمام خصیصه‌های مقسوم علیه در مقسوم باشد. به عبارت دیگر مجموعه صفات مقسوم علیه، زیر مجموعه، مجموعه صفات مقسوم باشد. همچنین ستون‌های حاصل از عملگر تقسیم برابر تفاضل ستون‌های مقسوم و مقسوم علیه است. همچنین سطرهایی از جدول r در خروجی ظاهر می‌گردد که تمام اعضاي جدول s را در کنار خود داشته باشد.

مثال:

a	b	\div	$\frac{b}{1} = \frac{a}{5}$
5	7		
5	1		2
4	2		جدول s
4	3		
5	2		

جدول r

جدول Borrowed که می‌گوید هر عضوی چه کتابی را به امانت برده‌است، مقسوم است و جدول Book با شرط «کلیه کتاب‌های منتشرشده توسط انتشارات McGraw-Hill» مقسوم علیه است.

گزینه اول نادرست است. زیرا، ستون Publisher در سمت مقسوم علیه وجود دارد، اما در سمت مقسوم وجود ندارد. بنابراین اجرای عملگر تقسیم در گزینه اول، به دلیل وجود خطای نحوی امکان‌پذیر نیست.

$$\Pi(name) \left((\Pi(isbn, name, mem-no) \text{ Borrowed} \bowtie \text{Member}) \div \sigma_{\text{publisher}} = 'McGraw-Hill' \text{ Book} \right)$$

گزینه دوم نادرست است. زیرا به دلیل مشابه گزینه اول، ستون Publisher در سمت مقسوم علیه وجود دارد، اما در سمت مقسوم وجود ندارد. بنابراین اجرای عملگر تقسیم در گزینه دوم نیز، به دلیل وجود خطای نحوی امکان‌پذیر نیست.

$$\Pi(name) \left(\left(\text{Borrowed} \div \sigma_{\text{publisher}} = 'McGraw-Hill' \text{ Book} \right) \bowtie \text{Member} \right)$$

گزینه چهارم نادرست است. زیرا، ستون isbn در سمت مقسوم علیه وجود دارد، اما در سمت مقسوم وجود ندارد. بنابراین اجرای عملگر تقسیم در گزینه چهارم نیز، به دلیل وجود خطای نحوی امکان‌پذیر نیست. دقت کنید که در پرس و جوی گزینه چهارم ابتدا عبارت زیر انجام می‌شود:

$$\Pi(name) (\text{Borrowed} \bowtie \text{Member})$$

که باعث می‌شود مقسوم، فاقد ستون isbn گردد.

$$\Pi(name) (\text{Borrowed} \bowtie \text{Member}) \div \Pi(isbn) \left(\sigma_{\text{publisher}} = 'McGraw-Hill' \text{ Book} \right)$$

گزینه سوم درست است. مطابق تعریف عملگر تقسیم در جبر رابطه‌ای، اجرای عملگر تقسیم در گزینه سوم امکان‌پذیر است.

$$\begin{aligned} \Pi(\text{name}) & \left[\frac{(\Pi(\text{mem-no}, \text{isbn}) \text{Borrowed})}{\downarrow} \div \frac{\Pi(\text{isbn})(\sigma_{\text{publisher}='\text{McGraw-Hill}' \text{Book}}) \bowtie \text{Member}}{\downarrow} \right] \\ & \begin{array}{c|c} \text{mem-no} & \text{isbn} \\ \hline m1 & is1 \\ m1 & is2 \\ m2 & is1 \\ m2 & is2 \\ m2 & is3 \end{array} \quad \begin{array}{c} \text{isbn} \\ \hline is1 \\ is2 \end{array} \\ & \downarrow \\ \Pi(\text{name}) & \left[\frac{\frac{\text{mem-no}}{m1} \bowtie \text{Member}}{m1} \right] \rightarrow \begin{array}{c} \text{name} \\ mn1 \\ mn2 \end{array} \end{aligned}$$

$$\Pi(\text{name}) \left((\Pi(\text{mem-no}, \text{isbn}) \text{Borrowed} \div \Pi(\text{isbn})(\sigma_{\text{publisher} = 'McGraw-Hill' \text{ Book}}) \bowtie \text{Member}) \right)$$

توجه: مشاهده می‌شود که اعضای $mn1$ و $mn2$ ، همه کتاب‌های منتشر شده توسط انتشارات McGraw-Hill را به امانت برده‌اند.

- ۵۹- گزینه (۴) صحیح است.

جداول زیر را در نظر بگیرید:

MID	Name	Mdate	MID	ISBN	BDate	...	ISBN	Title	...	publisher
m1	mn1	21	m1	is1	11		is1	tit1		McGraw-Hill
m2	mn2	22	m1	is2	12		is2	tit2		McGraw-Hill
m3	mn3	23	m2	is1	13		is3	tit3		Wiley
Member			m2	is2	14		is4	tit4		Wiley
			m2	is3	15		Book			
			m3	is1	16					
Borrow										

مطابق پرس و جوی مطرح شده در صورت سوال داریم:

«عنوان کتاب‌هایی از انتشارات wiley که توسط هیچ عضوی امانت گرفته نشده» پرس و جوی فوق توسط عملگر **تفاضل** در جبر رابطه‌ای قابل پیاده‌سازی است.

مطابق تعریف عملگر **تفاضل** در جبر رابطه‌ای، تفاضل هر دو رابطه دلخواه امکان‌پذیر نیست. مگر اینکه شروط سازگاری در مورد آنها برقرار باشد. در جبر رابطه‌ای دو شرط به عنوان شروط سازگاری مطرح است:

شرط اول: تعداد ستون‌های دو جدول یکسان باشد، به عبارت دیگر دو رابطه (جدول) هم درجه باشند.

شرط دوم: نوع یا دامنه ستون‌های متناظر در دو جدول یکسان باشد.

اگر بخواهیم دو شرط فوق را در یک جمله بیان کنیم، اینطور خواهد بود، شروط سازگاری یعنی تیترها در دو رابطه (جدول) یکسان باشد.

به طور کلی، اگر عبارت نفی در پرس و جو استفاده شده باشد، این امر نشانه‌ای برای ضرورت استفاده از عملگر تفاضل است. وقت کنید که در انجام عمل تفاضل برقراری شروط سازگاری الزامی است که در پرس و جوی گزینه چهارم این الزام برقرار است.

جدول Borrow می‌گوید هر عضوی چه کتابی را به امانت برده است. و جدول Book با شرط انتشارات wiley «کلیه کتاب‌های منتشرشده توسط انتشارات wiley» را بیان می‌کند. گزینه اول نادرست است. زیرا، شروط سازگاری در عملگر تفاضل برقرار نیست. بنابراین اجرای عملگر تفاضل در گزینه اول، به دلیل وجود خطای نحوی امکان‌پذیر نیست.

$$\Pi_{\langle \text{Title} \rangle} (\sigma_{\text{publisher} = \text{'wiley'}}(\text{Book})) - \Pi_{\langle \text{ISBN} \rangle} (\text{Borrow})$$

گزینه دوم نادرست است. زیرا، از عملگر تفاضل یعنی عبارت نفی در پرس و جو استفاده نشده است.

$$\Pi_{\langle \text{Title}, \text{ISBN} \rangle} (\sigma_{\text{publisher} = \text{'wiley'}}(\text{Book})) \div \Pi_{\langle \text{ISBN} \rangle} (\text{Borrow})$$

گزینه سوم نادرست است. زیرا، از عملگر تفاضل یعنی عبارت نفی در پرس و جو استفاده نشده است. مطابق تعریف عملگر تقسیم در جبر رابطه‌ای، باید تمام خصیصه‌های مقسوم علیه در مقسوم باشد. به عبارت دیگر مجموعه صفات مقسوم علیه، زیر مجموعه، مجموعه صفات مقسوم باشد. همچنین ستون‌های حاصل از عملگر تقسیم برابر تفاضل ستون‌های مقسوم و مقسوم علیه است. همچنین سطرهایی از جدول r در خروجی ظاهر می‌گردد که تمام اعضای جدول s را در کنار خود داشته باشد.

مثال:

a	b	$\div \frac{b}{1} = \frac{a}{5}$
5	7	
5	1	2
4	2	جدول s
4	3	
5	2	

جدول r

مطابق پرس و جوی مطرح شده در گزینه چهارم داریم:

$$\Pi_{\langle \text{Title} \rangle} \left(\left(\Pi_{\langle \text{ISBN}, \text{MID} \rangle} (\text{Borrow}) \div \Pi_{\langle \text{MID} \rangle} (\text{Member}) \right) \bowtie \text{Borrow} \right)$$

جدول Borrow که می‌گوید هر عضوی چه کتابی را به امانت برده است، مقسوم است و جدول Member که اعضاء را نشان می‌دهد مقسوم علیه است. مطابق تعریف عملگر تقسیم در جبر رابطه‌ای، اجرای عملگر تقسیم در گزینه سوم امکان‌پذیر است.

$$\Pi_{\langle \text{Title} \rangle} \left[\frac{((\Pi_{\langle \text{ISBN}, \text{MID} \rangle} (\text{Borrow}))}{\downarrow} \quad \div \quad \frac{\Pi_{\langle \text{MID} \rangle} (\text{Member}) \bowtie \text{Borrow}}{\downarrow} \right]$$

ISBN	MID		MID
is1	m1	÷	m1
is2	m1		m2
is1	m2		m3
is2	m2		
is3	m2		
is1	m3		

\downarrow

$$\Pi_{\langle \text{Title} \rangle} \left[\frac{\text{ISBN}}{\text{is1}} \bowtie \text{Borrow} \right] \rightarrow \times$$

توجه: مشاهده می‌شود که ISBN استخراج شده یعنی is1، توسط همه اعضاء به امانت برد شده است. یعنی پرس و جوی داخلی، ISBN کتبی را استخراج می‌کند که توسط همه اعضاء به امانت برد شده است. اما در ادامه و حرکت به سمت خارج و اجرای عملگر الحق طبیعی، ستون Title استخراج نمی‌شود، بنابراین اجرای عملگر پرتو در گزینه سوم، به دلیل وجود خطای نحوی امکان‌پذیر نیست.
 گزینه چهارم درست است. زیرا، مطابق تعریف عملگر تفاضل در جبر رابطه‌ای، اجرای عملگر تفاضل در گزینه چهارم امکان‌پذیر است.
 مطابق پرس و جوی مطرح شده در گزینه چهارم داریم:

$$\Pi_{\langle \text{Title} \rangle} \left((\Pi_{\langle \text{ISBN} \rangle} (\sigma_{\text{publisher} = 'wiley'} (\text{Book})) - \Pi_{\langle \text{ISBN} \rangle} (\text{Borrow})) \bowtie \text{Book} \right)$$

در پرس و جوی سمت چپ عملگر تفاضل یعنی در جدول Book مشخصات کلیه کتاب‌های منتشر شده قرار دارد، که پس از اجرای عملگر انتخاب مشخصات کلیه کتاب‌های منتشر شده توسط انتشارات wiley استخراج می‌گردد، در نهایت پس از اجرای عملگر پرتو، ISBN کلیه کتاب‌های منتشر شده توسط انتشارات wiley استخراج می‌شود، به صورت زیر:

ISBN
is3
is4

همچنین در پرس و جوی سمت راست عملگر تفاضل یعنی در جدول Borrow مشخصات کلیه‌ی کتاب‌هایی که تا کنون به امانت گرفته شده‌اند قرار دارد، که در نهایت پس از اجرای عملگر پرتو، ISBN کلیه کتاب‌هایی که تا کنون به امانت گرفته شده‌اند استخراج می‌شود، به صورت زیر:

ISBN
is1
is2
is3

در واقع در پرس و جوی فوق کلیه کتاب‌های منتشر شده توسط انتشارات Wiley منهای کلیه کتاب‌های به امانت گرفته شده تاکنون شده است.

که در قدم بعدی پس از اجرای عملگر تفاضل، خروجی پرس و جوی داخلی به صورت زیر خواهد بود:

$$\begin{array}{c} \text{ISBN} \\ \hline \text{is3} \\ \text{is4} \end{array} - \begin{array}{c} \text{ISBN} \\ \hline \text{is1} \\ \text{is2} \\ \text{is3} \end{array} = \begin{array}{c} \text{ISBN} \\ \hline \text{is4} \end{array}$$

همچنین از آنجاکه در پرس و جوی مطرح شده در صورت سوال، عنوان کتاب‌هایی از انتشارات wiley که توسط هیچ عضوی امانت گرفته نشده درخواست شده است، بنابراین در نهایت پس از اجرای عملگر الحق طبیعی، خروجی نهایی پرس و جوی به صورت زیر خواهد بود:

$$\begin{aligned} \Pi < \text{Title} > & \left[\frac{((\Pi < \text{ISBN} > (\sigma_{\text{publisher}} = 'wiley') (\text{Book})) - \Pi < \text{ISBN} > (\text{Borrow})) \bowtie \text{Book}}{\downarrow} \right] \\ & \begin{array}{c} \text{ISBN} \\ \hline \text{is3} \\ \text{is4} \end{array} - \begin{array}{c} \text{ISBN} \\ \hline \text{is1} \\ \text{is2} \\ \text{is3} \end{array} \\ & \xrightarrow{\quad} \\ \Pi < \text{Title} > & \left[\frac{\text{ISBN}}{\text{is4}} \bowtie \text{Book} \right] \rightarrow \frac{\text{Title}}{\text{tit4}} \end{aligned}$$

$$\Pi(\text{name}) \left(\left(\Pi(\text{mem-no}, \text{isbn}) \text{Borrowed} \div \Pi(\text{isbn}) (\sigma_{\text{publisher}} = 'McGraw-Hill') \text{ Book} \right) \bowtie \text{Member} \right)$$

توجه: مشاهده می‌شود که عنوان tit4 از انتشارات wiley توسط هیچ عضوی امانت گرفته نشده است.

۶۰- گزینه (۲) صحیح است.

عملگر ضرب دکارتی یا ضرب کارتزین توسط نماد \times نمایش داده می‌شود. عملگر \times یک عملگر اصلی است. عملگر ضرب دکارتی گاهای با نماد Times نیز نشان داده می‌شود. عملگر \times جهت اتصال کلیه سطرهای دو جدول که در هم ضرب می‌شود، مورد استفاده قرار می‌گیرد. در جبر رابطه‌ای ضرب دکارتی هر دو رابطه دلخواه امکان‌پذیر است و شرط خاصی برای ضرب ندارد.

کلید کاندید روابطه مقصد حاصل از عملگر \times مابین دو رابطه R₁ و R₂ همواره برابر هر ترکیبی از کلیدهای کاندید روابط مبدا یعنی هر ترکیبی از یک کلید کاندید R₁ و یک کلید کاندید R₂ می‌باشد، به صورت زیر:

$$\text{C.K.}(R_3) = \text{C.K.}(R_1) \cup \text{C.K.}(R_2)$$

مثال: جداول R₁ و R₂ را به صورت زیر در نظر بگیرید:

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

$$R_3 = R_1 \times R_2$$

پرس و جوی مقابل را در نظر بگیرید:

خروجی پرس و جوی فوق به صورت زیر است:

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

از آنجا که کلید کاندید رابطه مقصد حاصل از عملگر \times مابین دو رابطه R_1 و R_2 همواره برابر هر ترکیبی از کلیدهای کاندید روابط مبدا یعنی هر ترکیبی از یک کلید کاندید R_1 و یک کلید کاندید R_2 می‌باشد، پس کلید کاندید رابطه مقصد یعنی R_3 برابر ترکیبی از یک کلید کاندید R_1 و یک کلید کاندید R_2 است، به صورت زیر:

$$C.K.(R_3) = \{a\} \cup \{c\} = \{a, c\}$$

$$C.K.(R_3) = \{a\} \times \{c\} = \{a, c\}$$

بنابراین رابطه زیر برقرار خواهد بود:

$$C.K.(R_3) = C.K.(R_1) \cup C.K.(R_2)$$

$$C.K.(R_3) = C.K.(R_1) \times C.K.(R_2)$$

مثال: جداول R_1 و R_2 را به صورت زیر در نظر بگیرید:

a	b	c	d	e	f	g
1	1	2	5	5	6	7
3	3	2	8	8	9	10
1	3	2	11	11	12	13
			5	8	6	7

R_1 R_2

$$R_3 = R_1 \times R_2$$

پرس و جوی مقابل را در نظر بگیرید:

خروجی پرس و جوی فوق به صورت زیر است:

a	b	c	d	e	f	g	=	a	b	c	d	e	f	g
1	1	2	5	5	6	7		1	1	2	5	5	6	7
3	3	2	8	8	9	10		1	1	2	8	8	9	10
1	3	2	11	11	12	13		1	1	2	11	11	12	13
			5	8	6	7		1	1	2	5	8	6	7
								3	3	2	5	5	6	7
								3	3	2	8	8	9	10
								3	3	2	11	11	12	13
								3	3	2	5	8	6	7
								1	3	2	5	5	6	7
								1	3	2	8	8	9	10
								1	3	2	11	11	12	13
								1	3	2	5	8	6	7

از آنجا که کلید کاندید رابطه مقصد حاصل از عملگر \times مابین دو رابطه R_1 و R_2 همواره برابر هر ترکیبی از کلیدهای کاندید روابط مبدا یعنی هر ترکیبی از یک کلید کاندید R_1 و یک کلید کاندید R_2 می‌باشد، پس کلید کاندید رابطه مقصد یعنی R_3 برابر ترکیبی از یک کلید کاندید R_1 و یک کلید کاندید R_2 است، به صورت زیر:

$$C.K.(R_3) = \{a, b\} \cup \{d, e\} = \{a, b, d, e\}$$

$$C.K.(R_3) \neq \{a, b\} \cap \{d, e\} = \{\}$$

$$C.K.(R_3) \neq \{a, b\} \times \{d, e\} = \{(a, d), (a, e), (b, d), (b, e)\}$$

واضح است که در حالت کلی اگر جداول مبدا هیچ صفت مشترکی نداشته باشند، آنگاه کلید کاندید حاصل از ضرب دکارتی برابر اجتماع دو کلید جدول مبدا است و نه برابر اشتراک یا ضرب دکارتی دو کلید جدول مبدا. بنابراین گزینه اول، سوم و چهارم نادرست و گزینه دوم درست است.

$$C.K.(R_3) = C.K.(R_1) \cup C.K.(R_2)$$

دقت کنید که در صورت سوال بیان شده است که جداول مبدا هیچ صفت مشترکی نداشته باشند، و این فرض در صورت سوال در نظر گرفته شده است، حال سوال این است که اگر کلیدهای کاندید جداول مبدا صفت مشترک داشته باشند، آنگاه کلید کاندید حاصل از ضرب دکارتی جداول مبدا به چه صورت است؟ پاسخ این است که صفات مشترک میان کلیدهای کاندید جداول مبدا، می‌بایست در کلید کاندید حاصل از ضرب دکارتی جداول مبدا، دوبار تکرار شوند.

مثال: جداول R_1 و R_2 را به صورت زیر در نظر بگیرید:

a	b	a	c	d
1	2	5	6	7
3	2	8	9	10
R_1		11	12	13
R_2		14	6	7

$$R_3 = R_1 \times R_2$$

پرس و جوی مقابل را در نظر بگیرید:

خروجی پرس و جوی فوق به صورت زیر است:

a	b	x	a	c	d	=	R _{1.a}	b	R _{2.a}	c	d
1	2		5	6	7		1	2	5	6	7
3	2		8	9	10		1	2	8	9	10
			11	12	13		1	2	11	12	13
			14	6	7		1	2	14	6	7
							3	2	5	6	7
							3	2	8	9	10
							3	2	11	12	13
							3	2	14	6	7

از آنجا که کلید کاندید رابطه مقصد حاصل از عملگر \times مابین دو رابطه R_1 و R_2 همواره برابر هر ترکیبی از کلیدهای کاندید روابط مبدا یعنی هر ترکیبی از یک کلید کاندید R_1 و یک کلید کاندید R_2 می‌باشد،

پس کلید کاندید رابطه مقصد یعنی R_3 برابر ترکیبی از یک کلید کاندید R_1 و یک کلید کاندید R_2 است، به صورت زیر:

$$C.K.(R_3) = \{R_1.a\} \cup \{R_2.a\} = \{R_1.a, R_2.a\}$$

بنابراین رابطه زیر برقرار خواهد بود:

$$C.K.(R_3) = C.K.(R_1) \cup C.K.(R_2)$$

۶۱- گزینه (۳) صحیح است.

جداول زیر را در نظر بگیرید:

ID	Name	age	ID	ISBN	Date	ISBN	Title	publisher
m1	mn1	21	m1	is1	11	is1	tit1	McGraw-Hill
m2	mn2	22	m1	is2	12	is2	tit2	McGraw-Hill
m3	mn3	23	m2	is1	13	is3	tit3	Wiley
Member			m2	is2	14	is4	tit4	Wiley
			m2	is3	15	Book		
			m3	is1	16	Borrowed		

مطابق پرس و جوی مطرح شده در صورت سوال داریم:

«نام اعضاي را بیابید که همه کتاب‌های منتشرشده توسط McGraw-Hill را به امانت برده‌اند.»

پرس و جوی فوق توسط عملگر تقسیم در جبر رابطه‌ای قابل پیاده‌سازی است.

مطابق تعریف عملگر تقسیم در جبر رابطه‌ای، باید تمام خصیصه‌های مقسوم علیه در مقسوم باشد. به عبارت دیگر مجموعه صفات مقسوم علیه، زیر مجموعه، مجموعه صفات مقسوم باشد. همچنین ستون‌های حاصل از عملگر تقسیم برابر تفاضل ستون‌های مقسوم و مقسوم علیه است. همچنین سطرهایی از جدول r در خروجی ظاهر می‌گردد که تمام اعضاي جدول s را در کنار خود داشته باشد.

مثال:

a	b	÷	$\frac{b}{1}$	=	$\frac{a}{5}$
5	7				
5	1		2		
4	2	s			
4	3				
5	2				

جدول r

جدول r که می‌گوید هر عضوی چه کتابی را به امانت برده‌است، مقسوم است و جدول Book با شرط «کلیه‌ی کتاب‌های منتشرشده توسط انتشارات McGraw-Hill» مقسوم علیه است.

گزینه اول نادرست است. زیرا، ستون Publisher در سمت مقسوم علیه وجود دارد، اما در سمت مقسوم وجود ندارد. بنابراین اجرای عملگر تقسیم در گزینه اول، به دلیل وجود خطای نحوی امکان‌پذیر نیست.

$$\Pi_{<\text{Name}>} \left[\text{Member} \bowtie \Pi_{<\text{ID}>} \left(\text{Borrowed} \div (\sigma_{\text{publisher}} = 'McGraw-Hill' \text{ Book}) \right) \right]$$

گزینه دوم نادرست است. زیرا به دلیل مشابه گزینه اول، ستون Publisher در سمت مقسوم علیه وجود دارد، اما در سمت مقسوم وجود ندارد. بنابراین اجرای عملگر تقسیم در گزینه دوم نیز، به دلیل وجود خطای نحوی امکان‌پذیر نیست.

$$\Pi_{<\text{Name}>} \left[\text{Member} \times \Pi_{<\text{ID}>} \left(\text{Borrowed} \div (\sigma_{\text{publisher}} = 'McGraw-Hill' \text{ Book}) \right) \right]$$

گزینه سوم درست است. مطابق تعریف عملگر تقسیم در جبر رابطه‌ای، اجرای عملگر تقسیم در گزینه سوم امکان‌پذیر است.

$$\begin{aligned} \Pi_{<\text{Name}>} & \left[\frac{\text{Member} \bowtie ((\Pi_{<\text{ID}, \text{ISBN}>} \text{Borrowed}) \div (\Pi_{<\text{ISBN}>} (\sigma_{\text{publisher}} = 'McGraw-Hill' \text{ Book})))}{\begin{array}{c|c} \text{ID} & \text{ISBN} \\ \hline m1 & is1 \\ m1 & is2 \\ m2 & is1 \\ m2 & is2 \\ m2 & is3 \\ m3 & is1 \end{array} \quad \div \quad \begin{array}{c} \text{ISBN} \\ \hline is1 \\ is2 \end{array}} \right] \\ & \Pi_{<\text{Name}>} \left[\frac{\text{ID}}{\begin{array}{c} \text{ID} \\ \hline m1 \\ m2 \end{array}} \bowtie \text{Member} \right] \rightarrow \frac{\text{Name}}{\begin{array}{c} \text{mn1} \\ \text{mn2} \end{array}} \end{aligned}$$

$$\Pi_{<\text{Name}>} \left[\text{Member} \bowtie \left((\Pi_{<\text{ID}, \text{ISBN}>} \text{Borrowed}) \div (\Pi_{<\text{ISBN}>} (\sigma_{\text{publisher}} = 'McGraw-Hill' \text{ Book})) \right) \right]$$

توجه: مشاهده می‌شود که اعضای mn1 و mn2، همه کتاب‌های منتشر شده توسط انتشارات McGraw-Hill را به امانت برده‌اند.

گزینه چهارم نادرست است. زیرا، تا بخشی همانند گزینه سوم است اما در نهایت از عملگر ضرب دکارتی استفاده می‌کند که باعث می‌شود سطرهای پیوند پذیر یعنی معتر و هم سطرهای پیوند نداشته باشند یعنی نامعتبر در خروجی قرار گیرد.

$$\Pi_{<\text{Name}>} \left[\frac{\text{ID}}{\begin{array}{c} \text{m1} \\ \text{m2} \end{array}} \times \text{Member} \right] \rightarrow \frac{\text{Name}}{\begin{array}{c} \text{mn1} \\ \text{mn2} \\ \text{mn3} \end{array}}$$

توجه: مشاهده می‌شود که اعضای mn1 و mn2، همه کتاب‌های منتشر شده توسط انتشارات McGraw-Hill را به امانت برده‌اند. اما عضو mn3 نیز که همه کتاب‌های منتشر شده توسط انتشارات McGraw-Hill را به امانت نبرده است نیز در خروجی قرار گرفته است.

۶۲ - گزینه (۲) صحیح است.

دو جدول Node و Edge با مقادیر زیر را در نظر بگیرید:

NID	Name	Color	Description	NID1	NID2	EdgeType
N1	NN1	C1	D1	N1	N2	T1
N2	NN2	C2	D2	N1	N3	T2
N3	NN3	C3	D3	N4	N1	T3
N4	NN4	C4	D4	N3	N2	T4
				N4	N3	T5

جدول Node

جدول Edge

توجه: کلید کاندید جدول Node ستون NID است و جدول Edge دارای دو کلید خارجی NID1 و NID2 است که هر دو به کلید کاندید جدول Node ارجاع می‌کنند.

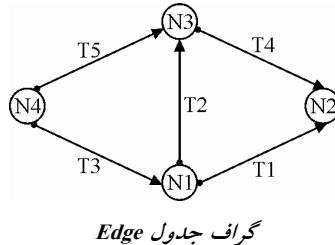
توجه: ستون NID1 در جدول Edge به عنوان کلید خارجی به ستون NID از جدول Node ارجاع می‌کند.

توجه: ستون NID2 در جدول Edge به عنوان کلید خارجی به ستون NID از جدول Node ارجاع می‌کند.

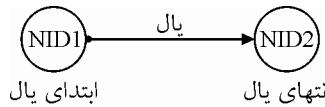
توجه: مقادیر کلید خارجی یعنی ستون‌های NID1 و NID2 از جدول Edge همواره باید زیرمجموعه مقادیر کلید کاندید یعنی ستون NID از جدول Node باشد.

توجه: هر تاپل یا سطر یا رکورد از جدول Edge نشانه یک ارتباط است، پس هر تاپل نشانه یک یال هم هست، که مجموع این یال‌ها یک گراف را ایجاد می‌کند. خطی که از یک گره به گره بعدی رسم می‌شود یک یال است. هر گراف G شامل دو مجموعه V و E است. V مجموعه محدود و غیرتنهی از رئوس است و E مجموعه‌ای محدود و احتمالاً تهی از لبه‌ها می‌باشد. $G = (V, E)$ یعنی G یک گراف بدون جهت، زوج رئوس، زوج مرتب نیستند. بنابراین زوج‌های (V_0, V_1) و (V_1, V_0) باهم یکسان هستند. گراف حداقل یک راس دارد و نمی‌تواند کاملاً تهی باشد. در یک گراف جهت دار، هر لبه با زوج مرتب $<V_0, V_1>$ نمایش داده می‌شود، که پیکانی از V_0 به V_1 ترسیم می‌شود. بنابراین $<V_1, V_0>$ و $<V_0, V_1>$ دو لبه متفاوت را نمایش می‌دهند. گراف جهت دار را به صورت $G = \langle V, E \rangle$ نمایش می‌دهیم.

توجه: گراف جهت دار جدول Edge به صورت زیر است.



گراف جدول Edge



هر تاپل از جدول Edge نشانه یک یال از گراف جدول Edge است.

توجه: همانطور که واضح است گره‌های موجود در ستون NID1 از جدول Edge یا گراف Edge، یال‌های خروجی دارند.

$$NID1 = \{N1, N3, N4\}$$

یعنی از گره‌های N1، N3 و N4 یال خارج شده است.

توجه: همانطور که واضح است گره‌های موجود در ستون NID2 از جدول Edge یا گراف Edge، یال‌های ورودی دارند.

$$NID2 = \{N1, N2, N3\}$$

یعنی به گره‌های N1 و N2 یال وارد شده است.

نتیجه:

استخراج گره‌هایی که هم یالی از آن خارج شده است و هم یالی به آن وارد شده است، به صورت زیر است:

$$NID1 \cap NID2 = \{N1, N3, N4\} \cap \{N1, N2, N3\} = \{N1, N3\}$$

استخراج گره‌هایی که یالی از آن خارج شده است ولی یالی به آن وارد نشده است، به صورت زیر است:

$$NID1 - NID2 = \{N1, N3, N4\} - \{N1, N2, N3\} = \{N4\}$$

استخراج گره‌هایی که یالی به آن وارد شده است ولی یالی از آن خارج نشده است، به صورت زیر است:

$$NID2 - NID1 = \{N1, N2, N3\} - \{N1, N3, N4\} = \{N2\}$$

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

$$\Pi_{E1.NID1} \left(\sigma_{E1.NID2 = E2.NID1} \left(\sigma_{E1.EdgeType = T2} (\rho_{E1}(Edge)) \times \rho_{E2}(Edge) \right) \right)$$

ابتدا در داخلی ترین پرانتز، عملوند سمت چپ عملگر ضرب دکارتی اجرا می‌شود، که خروجی آن بر اساس عملگر ρ_{E1} به جدول E1 نام‌گذاری می‌شود، به صورت زیر:

$$\sigma_{E1.EdgeType = T2} (\rho_{E1}(Edge))$$

معادل پرس و جوی جبر رابطه‌ای فوق به زبان SQL به صورت زیر است:

```
(select *
  from edge
  where EdgeType = 'T2') E1
```

خروجی قطعه کد فوق به صورت زیر است:

NID1	NID2	EdgeType
N1	N3	T2

جدول E1

توجه: قطعه پرس و جوی فوق یال‌های نوع T2 را استخراج می‌کند.

در ادامه در داخلی ترین پرانتز، عملوند سمت راست عملگر ضرب دکارتی اجرا می‌شود، که خروجی آن بر اساس عملگر ρ_{E2} به جدول E2 نام‌گذاری می‌شود، به صورت زیر:

$$\rho_{E2}(Edge)$$

معادل پرس و جوی جبر رابطه‌ای فوق به زبان SQL به صورت زیر است:

```
(select *
  from edge) E2
```

خروجی قطعه کد فوق به صورت زیر است:

NID1	NID2	EdgeType
N1	N2	T1
N1	N3	T2
N4	N1	T3
N3	N2	T4
N4	N3	T5

جدول E2

در ادامه در داخلی ترین پرانتز، عملوند سمت چپ عملگر ضرب دکارتی (T1) و عملوند سمت راست عملگر ضرب دکارتی (T2) در هم ضرب دکارتی می‌شوند، که خروجی آن بر اساس عملگر $\sigma_{E1.NID2 = E2.NID1}$ به صورت زیر است:

$$\sigma_{E1.NID2 = E2.NID1} \left(\sigma_{E1.EdgeType = T2} (\rho_{E1}(Edge)) \times \rho_{E2}(Edge) \right)$$

معادل پرس و جوی جبر رابطه‌ای فوق به زبان SQL به صورت زیر است:

```
select *
from (select *
       from edge
      where EdgeType= 'T2') E1 ,
(select *
       from edge) E2
where E1.NID2 = E2.NID1
```

خروجی قطعه کد فوق به صورت زیر است:

E1.NID1	E1.NID2	E1.EdgeType	E2.NID1	E2.NID2	E2.EdgeType
N1	N3	T2	N1	N2	T1
N1	N3	T2	N1	N3	T2
N1	N3	T2	N4	N1	T3
N1	N3	T2	N3	N2	T4
N1	N3	T2	N4	N3	T5

که خروجی آن پس از اجرای عملگر $\sigma_{E1.NID2 = E2.NID1}$ به صورت زیر است:

E1.NID1	E1.NID2	E1.EdgeType	E2.NID1	E2.NID2	E2.EdgeType
N1	N3	T2	N3	N2	T4

در نهایت و حرکت به سمت خارجی ترین پرانتز، عملگر Project اجرا می‌شود، که خروجی آن بر اساس عملگر $\Pi_{E1.NID1}$ به صورت زیر است:

$$\Pi_{E1.NID1} \left(\sigma_{E1.NID2 = E2.NID1} \left(\sigma_{E1.EdgeType = T2} (\rho_{E1}(Edge)) \times \rho_{E2}(Edge) \right) \right)$$

معادل پرس و جوی جبر رابطه‌ای فوق به زبان SQL به صورت زیر است:

```

select E1.NID1
from (select *
       from edge
      where EdgeType= 'T2') E1 ,
(select *
       from edge) E2
  where E1.NID2 = E2.NID1

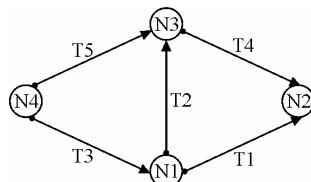
```

که خروجی نهایی آن پس از اجرای عملگر $\Pi_{E1.NID1}$ به صورت زیر است:

$$\frac{E1.NID1}{N1}$$

یعنی «شماره گرهایی که حداقل یک یال خروجی از نوع T2 به یک گره مانند g دارند و گره g حداقل یک یال خروجی دارد.» بنابراین پُر واضح است که گزینه دوم پاسخ سوال است.

شکل زیر گویای مطلب است:



گراف جدول Edge

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

۶۳- گزینه (۳) صحیح است.

صورت سوال به صورت زیر است:

با توجه به جدول ارائه شده، کدام عبارت جبری درست است؟

$STUD(S\#, Sname, City, avg, clg\#)$

(شماره دانشکده، محل تولد، نام، شماره دانشجویی) جدول دانشجو

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

گزینه اول نادرست است، زیرا دو عملگر σ و Π به صورت مشروط دارای خاصیت جابه‌جای هستند. به طور کلی اگر R یک رابطه، L زیر مجموعه‌ای از ستون‌ها و θ مجموعه‌ای از شروط بر روی سطرها باشد، آنگاه تساوی زیر زمانی برقرار است که ستون‌های عملگر σ زیر مجموعه ستون‌های عملگر Π باشد. یعنی

$$\sigma \subseteq \Pi$$

$$\Pi_L(\sigma_\theta(R)) = \sigma_\theta(\Pi_L(R))$$

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

S#	Sname	City	avg	clg#
S1	Sn1	C1	14	clg1
S2	Sn2	C2	17	clg2
S3	Sn3	C3	18	clg3

پرس و جوی زیر را در نظر بگیرید:

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

خروجی پرس و جوی 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:

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

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 نیست، یعنی $\text{avg} \not\subseteq \text{City}$. در واقع ستون avg در پرانتز داخلی توسط عملگر Π انتخاب نشده است، بنابراین اجرای عملگر σ بر روی ستون avg امکان‌پذیر نیست.

پس رابطه زیر برقرار است:

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

بنابراین همانطور که واضح است، گزینه اول نادرست است.

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

S#	Sname	City	avg	clg#
S1	Sn1	C1	14	clg1
S2	Sn2	C2	17	clg2
S3	Sn3	C3	18	clg3

پرس و جوی زیر را در نظر بگیرید:

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

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 پرانتز داخلی به صورت زیر است:

City	avg
C1	14
C2	17
C3	18

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

City	avg
C2	17
C3	18

پس رابطه زیر برقرار است:

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

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

$$\sigma_{City}(\Pi_{avg>16}(STUD)) = \Pi_{avg>16}(\sigma_{City}(STUD)) \quad (2)$$

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 $=, \neq, <, \leq, \geq, >$ و عملگرهای منطقی \wedge, \vee, \neg می‌توانند 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.

$$\Pi_{\text{City}}(\sigma_{\text{avg}>16}(\text{STUD})) \quad (3)$$

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 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

$$\sigma_{\text{avg}>16}(\Pi_{\text{City}}(\text{STUD})) \quad (4)$$

گزینه چهارم نادرست است.

پرس و جوی زیر را در نظر بگیرید:

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

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 avg یعنی σ زیر مجموعه ستون‌های عملگر Π یعنی City نیست، یعنی $\text{avg} \not\subseteq \text{City}$. در واقع ستون avg در پرانتز داخلی توسط عملگر Π انتخاب نشده است، بنابراین اجرای عملگر σ بر روی ستون avg امکان‌پذیر نیست.

_____ ۶۴- گزینه (۲) صحیح است.

مطابق فرض صورت سؤال، کار عملگر $\rho_{\text{second}}(\text{Marks})$ این است که یک کپی از رابطه‌ی Marks با نام Second ایجاد کند.

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

<i>Sid</i>	<i>sname</i>	<i>Sid</i>	<i>cid</i>	<i>mark</i>	<i>cid</i>	<i>cname</i>	<i>dept</i>
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						

Sudents جدول*Marks* جدول*Courses* جدول

پرس و جوی مطرح شده در صورت سؤال به صورت زیر است:

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

$$\text{Marks}.\text{mark} > \text{Second}.\text{mark}$$

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. وقت کنید که در انجام عمل تفاضل برقراری شروط سازگاری الزامی است که در پرس و جوی مطرح شده در صورت سوال این الزام برقرار است.

شروط سازگاری: در جبر رابطه‌ای دو شرط به عنوان شروط سازگاری مطرح است:

شرط اول: تعداد ستون‌های دو جدول یکسان باشد به عبارت دیگر دو جدول هم درجه باشند.

شرط دوم: نوع یا دامنه ستون‌های دو جدول یکسان باشد.

توجه: در پرس و جوی مطرح شده در هر دو طرف عملگر تفاضل، شروط سازگاری رعایت شده است.

براساس پرس و جوی 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:

<u>mark</u>
12
14
16

همچنین براساس بخش سمت راست عملگر تفاضل داریم:

عبارت $(\text{Marks} \times \rho_{\text{second}}(\text{Marks}))$ ، ضرب دکارتی جدول Marks در خودش است. حاصل این ضرب $\text{Marks}.\text{mark} > \text{Second}.\text{mark}$

دکارتی، جدولی است که در آن، رکورد اطلاعاتی هر دانشجو از جدول Marks در کنار رکوردهای اطلاعاتی همه دانشجویان جدول second قرار می‌گیرد.

Marks.sid	Marks.mark	Second.sid	Second.mark
S ₁	12	S ₁	12
S ₁	12	S ₂	14
S ₁	12	S ₃	16
S ₂	14	S ₁	12
S ₂	14	S ₂	14
S ₂	14	S ₃	16
S ₃	16	S ₁	12
S ₃	16	S ₂	14
S ₃	16	S ₃	16

سپس شرط $\text{Marks.mark} > \text{Second.mark}$ ، سطرهایی که دانشجویی جدول Marks نمره‌اش حداقل از یک نفر دیگر بیشتر باشد را انتخاب می‌کند (یعنی دانشجویانی که نمره‌شان کمترین نمره نیست). بنابراین، پس از اعمال شرط داریم:

Marks.sid	Marks.mark	Second.sid	Second.mark
S ₂	14	S ₁	12
S ₃	16	S ₁	12
S ₃	16	S ₂	14

در ادامه، توسط عملگر $\Pi_{\text{Marks.mark}}$ ، نمره این نوع از دانشجویان، به صورت زیر استخراج می‌گردد.

Marks.mark
14
16

توجه: در جبر رابطه‌ای، سطرهای تکراری در خروجی قرار نمی‌گیرند.

در انتهای، با انجام عملگر تفاضل، نمره دانشجویان فوق، از مجموعه نمره کلیه دانشجویان کنار گذاشته می‌شوند. آنچه باقی می‌ماند، نمره دانشجو یا دانشجویانی است که کمتر از همه نمره گرفته‌اند. به عبارت دیگر مینیمم نمره دانشجویان استخراج می‌گردد. بنابراین واضح است که پاسخ گزینه سوم است. یعنی «نمره مینیمم را از لیست نمرات استخراج می‌کند.»

خروجی نهایی پرس‌وجوی مطرح شده به صورت زیر است:

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

14	16
16	

توجه: اگر شرط موجود در پرس‌وجوی مطرح شده عکس گردد، آنگاه گزینه سوم ایجاد می‌گردد.

تست‌های فصل پنجم: حساب رابطه‌ای

۱- کدام گزینه پاسخ پرس‌وجوی «اسامی دانشکده‌هایی که همه دانشجویان آن معدل الف دارند» به روی DRC می‌باشد؟
(مهندسی IT - آزاد ۸۸ - ۸۹)

$$\{< \text{c ln} | \exists \text{cl}, \text{ci}, \text{pn} (< \text{cl}, \text{cln}, \text{ci}, \text{pn}) \in \text{clg} \wedge \forall \text{s}, \text{sn}, \text{ci2}, \text{a}, \text{cl2} & (1) \\ (< \text{s}, \text{sn}, \text{ci2}, \text{a}, \text{cl2}) \in \text{stud} \wedge \text{a} \geq 17))\}$$

$$\{< \text{c ln} | \exists \text{cl}, \text{ci}, \text{pn} (< \text{cl}, \text{cln}, \text{ci}, \text{pn}) \in \text{clg} \wedge \exists \text{s}, \text{sn}, \text{ci2}, \text{a}, \text{cl2} & (2) \\ (< \text{s}, \text{sn}, \text{ci2}, \text{a}, \text{cl2}) \in \text{stud} \wedge \text{a} \geq 17))\}$$

$$\{< \text{c ln} | \exists \text{cl}, \text{ci}, \text{pn} (< \text{cl}, \text{cln}, \text{ci}, \text{pn}) \in \text{clg} \wedge \exists \text{s}, \text{sn}, \text{ci2}, \text{a} & (3) \\ (< \text{s}, \text{sn}, \text{ci2}, \text{a}, \text{cl}) \in \text{stud} \wedge \text{a} \geq 17))\}$$

$$\{< \text{c ln} | \exists \text{cl}, \text{ci}, \text{pn} (< \text{cl}, \text{cln}, \text{ci}, \text{pn}) \in \text{clg} \wedge \forall \text{s}, \text{sn}, \text{ci2}, \text{a} & (4) \\ (< \text{s}, \text{sn}, \text{ci2}, \text{a}, \text{cl}) \in \text{stud} \wedge \text{a} \geq 17))\}$$

۲- اگر جداول دانشجو و دانشکده و درس به صورت زیر باشند:

stud(s#, sname, city, avg, clg#)

crs(c#, cname, unit, clg#)

clg(clg#, clgname, city, pname)

در حساب رابطه‌ای دامنه‌ای، در کدام گزینه صورت پرس‌وجو و پاسخ با هم مطابقت ندارد؟

(مهندسی کامپیوتر - آزاد ۸۹ - ۵۰۹ الف)

۱) شماره و نام دانشجویانی که معدل الف دارند یا متولد زاهدان هستند.

$$\{< \text{s}, \text{sn} > | \exists \text{ci}, \text{a}, \text{cl} (< \text{s}, \text{sn}, \text{ci}, \text{a}, \text{cl}) \in \text{stud} \wedge (\text{a} \geq 17 \vee \text{ci} = "Zahedan"))\}$$

۲) اسامی دانشکده‌هایی که همه دانشجویان آن معدل الف دارند.

$$\{< \text{cln} > | \exists \text{cl}, \text{ci}, \text{pn} (< \text{cl}, \text{cln}, \text{ci}, \text{pn}) \in \text{clg} \wedge \forall \text{s}, \text{sn}, \text{ci2}, \text{a} (< \text{s}, \text{sn}, \text{ci2}, \text{a}, \text{cl}) \in \text{stud} \wedge \text{a} \geq 17))$$

۳) شماره و نام دانشکده‌هایی که دروس چهار واحدی ارائه می‌کنند.

$$\{< \text{cl}, \text{cln} > | \exists \text{ci}, \text{pn} (< \text{cl}, \text{cln}, \text{ci}, \text{pn}) \in \text{clg} \wedge \exists \text{c}, \text{cn}, \text{u} (< \text{c}, \text{cn}, \text{u}, \text{cl}) \in \text{crs} \wedge \text{u} = 4)\}$$

۴) شماره و نام دانشجویانی که در دانشکده کامپیوتر تحصیل می‌کنند.

$$\{< \text{s}, \text{sn} > | \exists \text{cl}, \text{cln}, \text{ci}, \text{pn} (< \text{cl}, \text{cln}, \text{ci}, \text{pn}) \in \text{clg} \wedge \text{clg} \wedge \exists \text{a} (< \text{s}, \text{sn}, \text{ci}, \text{a}, \text{cl}) \in \text{stud} \wedge \text{cln} = "computer")\}$$

۳- کدام یک از گزاره‌های زیر درست است؟
(مهندسی IT - دولتی ۹۰)

۱) قدرت بیان کنندگی (expressive power) حساب رابطه‌ای دامنه‌ای و حساب رابطه‌ای تاپلی با هم برابر است اگر عبارات در هر دو حساب به عبارات امن (safe expression) محدود گردند.

۲) زبان QBE بر مبنای تئوری حساب رابطه‌ای تاپلی استوار شده است.

۳) قدرت بیان کنندگی (expressive power) حساب رابطه‌ای تاپلی و جبر رابطه‌ای با هم برابر است.
۴) یک زبان رویه‌ای است.

۴- اگر جدول دانشجو و دانشکده و درس به صورت زیر باشند: (مهندسی کامپیوتر- آزاده ۹۵- گروه الف)

$\text{stud}(\text{s}\#, \text{fname}, \text{city}, \text{avg}, \text{clg}\#)$
 $\text{crs}(\text{c}\#, \text{cname}, \text{unit}, \text{clg}\#)$
 $\text{clg}(\text{clg}\#\text{clgname}, \text{city}, \text{pname})$

در حساب رابطه‌ای دامنه‌ای، در کدام گزینه صورت پرس‌وجو و پاسخ با هم مطابقت دارد؟

(۱) شماره و نام دانشجویانی که معدل الف دارند یا متولد زاهدان هستند.

(۲) اسمی دانشکده‌هایی که همه دانشجویان آن معدل الف دارند.

$\{<\text{cln}>|\exists \text{cl}, \text{ci}, \text{pn}(<\text{cl}, \text{cln}, \text{ci}, \text{pn}> \in \text{clg} \wedge \forall \text{s}, \text{sn}, \text{ci2}, \text{a}(<\text{s}, \text{sn}, \text{ci2}, \text{a}, \text{cl}> \in \text{stud} \wedge \text{a} \geq 17))\}$

(۳) شماره و نام دانشکده‌هایی که دروس چهار واحدی ارائه می‌کنند.

$\{<\text{cl}, \text{cln}>|\exists \text{ci}, \text{pn}(<\text{cl}, \text{cln}, \text{ci}, \text{pn}> \in \text{clg} \wedge \exists \text{c}, \text{cn}, \text{u}(<\text{c}, \text{cn}, \text{u}, \text{cl}> \in \text{crs} \wedge \text{u} = 4))\}$

(۴) همه موارد

۵- براساس جداول S , P , J و SPJ به صورت زیر، در کدام یک از موارد زیر، نتیجه حاصل از اجرای عبارت جبر رابطه‌ای با نتیجه حاصل از اجرای عبارت حساب رابطه‌ای تاپلی یکسان است؟ (مهندسی IT - دولتی ۹۴)

$S(\underline{S\#}, \text{SNAME}, \text{STATUS}, \text{CITY})$

جدول تولید کنندگان

$P(\underline{P\#}, \text{PNAME}, \text{WEIGHT}, \text{CITY})$

جدول قطعات

$J(\underline{J\#}, \text{JNAME}, \text{CITY})$

جدول پروژه‌ها

$SPJ(\underline{S\#}, \underline{P\#}, \underline{J\#}, \text{QTY})$

جدول تعداد قطعات تولید شده

توسط تولید کنندگان در پروژه‌ها

۱)

$\{t | \exists \text{spjx} \in SPJ, \exists \text{sx} \in S (t[\text{SNAME}] = \text{sx}[\text{SNAME}] \wedge \text{spjx}[S\#] = \text{sx}[S\#] \wedge \text{spjx}[J\#] = 'jl')\}$

$\Pi_{S,SNAME} \sigma_{SPJ,J='jl'}(S \bowtie SPJ)$

(۱) فقط ۱

۲)

$\{t | \forall \text{px} \in P (\exists \text{spjx} \in SPJ, \exists \text{sx} \in S (\text{spjx}[S\#] = \text{sx}[S\#] \wedge \text{spjx}[P\#] = \text{px}[P\#] \wedge t[\text{SNAME}] = \text{sx}[\text{SNAME}]))\}$

$\Pi_{S,SNAME} (S \bowtie (\Pi_{S\#, P\#} SPJ \div \Pi_{P\#} P))$

(۲) فقط ۲

۳)

(۴) در هر دو مورد یکسان نیست.

(۳) ۱ و ۲

۶- معادل حساب رابطه‌ای یا جبر رابطه‌ای پرس و جوی «نام دانشجویانی که حداقل یک درس عملی - منظور

درسی که COTYPE آن برابر 'p' باشد - در ترم ۲ سال ۹۴-۹۵ انتخاب کرده باشند» کدام مورد زیر است؟

دانشجویان (STID, STNAME, STLEVEL, STMAJOR)

CO (COID, CONAME, COTYPE, CREDIT)

درس‌های انتخاب شده توسط دانشجویان (STCO (STID, COID, YR, TR, GRADE)

RANGVAR STX OVER ST

RANGVAR COX OVER CO

RANGVAR STCOX OVER STCO

A) $\text{STX.STNAME WHERE EXISTS STCOX (STCOX.YR='94-95' AND STCOX.TR='2' AND STCOX.STID=STX.STID AND EXIST COX (COX.COVID=STCOX.COVID AND COX.COTYPE='P'))}$

B) $\Pi_{(\text{STNAME})} \left(\text{ST} \bowtie (\sigma_{\text{YR}='94-95'} \wedge \text{TR}='2') (\text{STCO}) \bowtie \sigma_{\text{COTYPE}='P'} (\text{CO}) \right)$

C) $\text{STX.STNAME WHERE NOT EXISTS STCOX (STCOX.YR='94-95' AND STCOX.TR='2' AND STCOX.STID=STX.STID AND NOT EXIST COX (COX.COVID=STCOX.COVID AND COX.COTYPE='P'))}$

D) $\Pi_{(\text{STNAME})} \left(\sigma_{\text{YR}='94-95'} \wedge \text{TR}='2' \wedge \text{COTYPE}='P' (\text{ST} \bowtie (\text{STCO} \bowtie \text{CO})) \right)$

(مهندسى کامپیووتر-دولتى ۹۵)

D و B و A (۴)

فقط C و D (۳)

فقط A و B (۲)

فقط C (۱)

۷- کدام مورد، عبارت حساب رابطه‌ای معادل «نام دانشجویانی که تمام دروس ۴ واحدی را اخذ کرده‌اند» است؟
(مهندسى کامپیووتر-دولتى ۹۶)

دانشجویان : ST (STID, STName,...)

دروس : CO (COID, COName, Credit,...)

دانشجویان : دروس انتخاب شده توسط دانشجویان STCO(STID, COID, Semester, Grade)

Rangevar STX Over ST

Rangevar COX Over CO

Rangevar STCOX Over STCO

$\text{STX.STNAME WHERE FORALL COX (COX.CREDIT=4 AND EXISTS STCOX (STCOX.STID=STX.STID AND STCOX.COVID=COX.COVID))}$

$\text{STX.STNAME WHERE FORALL COX (COX.CREDIT != 4 OR NOT EXISTS (STCOX(STCOX.STID=STX.STID AND STCOX.COVID=COX.COVID)))}$

$\text{STX.STNAME WHERE NOT EXISTS COX(COX.CREDIT=4 AND NOT EXISTS (STCOX(STCOX.STID=STX.STID AND STCOX.COVID=COX.COVID)))}$

$\text{STX.STNAME WHERE EXISTS STCOX(STCOX.STID=STX.STID AND EXISTS COX(COX.COVID=STCOX.COVID AND COX.CREDIT=4))}$

پاسخ تست‌های فصل پنجم: حساب رابطه‌ای

۱- گزینه (۴) صحیح است.

DRC مخفف حساب رابطه‌ای دامنه‌ای (Domain Relation Calculus) است. از آنجا که صورت سؤال همه دانشجویان را مورد پرس و جو قرار داده است. پس باید از سور عمومی استفاده گردد. بنابراین گزینه‌های دوم و سوم را کنار می‌گذاریم. برای یافتن دانشجویان هم دانشکده باید دو رابطه دانشجویان و دانشکده براساس ستون مشترک cl به هم متصل شوند. بنابراین فقط باید تاپل‌هایی از دو رابطه بررسی شوند که مقدارشان در صفت مشترک cl یکسان است. اما در گزینه اول الزامی به یکسان بودن ستون مشترک cl نشده است. بر این اساس این گزینه نمی‌تواند پاسخ مورد نظر را تولید نماید. اما این نکته در گزینه چهارم رعایت شده است. اگر با دقت دو گزینه اول و چهارم بررسی شود مشخص خواهد شد که تنها تفاوت این دو گزینه در الزام به یکسان بودن cl در گزینه چهارم و عدم این الزام در گزینه اول است.

۲- گزینه (۴) صحیح است.

در عبارت حساب رابطه‌ای نوشته شده، در گزینه چهارم جدول clg با جدول stud روی دو فیلد $\# \text{clg}$ و city پیوند خورده است:

$$\{(s, sn) \mid \exists cl, cln, ci, pn \langle cl, cln, ci, pn \rangle \in clg \wedge \\ \exists a \langle s, sn, ci, a, cl \rangle \in stud \wedge cln = "computer"\}\}$$

پیوند روی $\# \text{clg}$ (به همراه شرط $\text{cln} = "computer"$) باعث می‌شود فقط دانشجویان دانشکده‌ی کامپیوتر در خروجی ظاهر شوند و پیوند روی city باعث می‌شود دانشجویانی در خروجی ظاهر شوند که شهر محل زندگی شان با شهر محل تحصیل شان برابر باشد. بنابراین، خروجی این عبارت حساب رابطه‌ای عبارتست از: «شماره و نام دانشجویانی که در دانشکده‌ی کامپیوتر تحصیل می‌کنند و در شهر محل تحصیل خود زندگی می‌کنند».

برای این‌که در خروجی، «شماره و نام دانشجویانی که در دانشکده‌ی کامپیوتر تحصیل می‌کنند» ظاهر شود باید دو جدول clg و stud فقط روی فیلد $\# \text{clg}$ پیوند زده شوند و پیوند روی فیلد city حذف شود. برای این‌منظور، کافی است در جدول stud برای فیلد city نام دیگری به جز ci (مثلاً ci2) انتخاب کنیم و اسم این متغیر را جلوی نماد \exists مربوط به جدول stud ذکر کنیم:

$$\{(s, sn) \mid \exists cl, cln, ci, pn \langle cl, cln, ci, pn \rangle \in clg \wedge \exists a, \underline{\text{ci2}} \\ \langle s, sn, \underline{\text{ci2}}, a, cl \rangle \in stud \wedge cln = "computer"\}\}$$

۳- گزینه (۱) صحیح است.

زبان QBE براساس حساب رابطه‌ای دامنه‌ای استوار است. قدرت بیان‌کنندگی حساب رابطه‌ای دامنه‌ای و حساب رابطه‌ای تاپلی با هم برابر است، اگر و تنها اگر در صورتی که عبارات به عبارات امن (مجموعه مقادیر از پیش تعریف شده) محدود شوند. قدرت بیان‌کنندگی حساب رابطه‌ای تاپلی از جبر رابطه‌ای بیش‌تر است. زیرا حساب رابطه‌ای تاپلی و دامنه‌ای مربوط به تحلیل پرس و جوها و مربوط به فاز تحلیل (ادرانکی عام) و جبر رابطه‌ای مربوط به طراحی پرس و جوها و مربوط به فاز طراحی (ادرانکی خاص) است، فاز تحلیل انتزاعی‌تر و به زبان محیط عملیاتی و زبان انسان نزدیک‌تر است و فاز طراحی به زبان

پیاده‌سازی در حال نزدیک شدن است.

SQL یک زبان بیانی (declarative) است نه یک زبان رویه‌ای یا دستوری، البته نسخه‌های رویه‌ای از SQL نیز توسعه یافته‌اند (مثل PL-SQL).

۴- گزینه (۴) صحیح است.

۵- گزینه (۱) صحیح است.

فعالیت‌های مربوط به تولید سیستم‌های نرم‌افزاری شامل مراحل تحلیل، طراحی و پیاده‌سازی می‌باشد. حساب رابطه‌ای (دامنه‌ای و تاپلی) برای تحلیل پرس‌وجو در فعالیت تحلیل مورد استفاده قرار می‌گیرد که خروجی این فعالیت به عنوان ورودی فعالیت طراحی برای طراحی پرس‌وجو توسط جبر رابطه‌ای مورد استفاده قرار می‌گیرد. حساب رابطه‌ای برای مدل‌سازی عالم خارج (زبان انسان) مورد استفاده قرار می‌گیرد. اما جبر رابطه‌ای برای مدل‌سازی عالم داخل (زبان ماشین) مورد استفاده قرار می‌گیرد.

توجه: متغیر سطّری (تاپلی) sx و $spjx$ به عنوان دو متغیر به ترتیب بر روی دو جدول jz و s تعریف شده‌اند. متغیر سطّری سطر بر روی جدولی که بر روی آن تعریف شده است حرکت می‌کند. به بیان دیگر دامنه حرکت یک متغیر سطّری، جدولی است که بر روی آن تعریف شده است.

هر دو عبارت گزاره اول، نام تولیدکنندگان حاضر در پروژه Jz را استخراج می‌کنند. در واقع نام تولیدکنندگانی را استخراج می‌کند که برای پروژه Jz ، حداقل یک قطعه تولید کرده‌اند.

عبارت جبر رابطه‌ای داده شده در گزاره دوم، نام تولیدکنندگانی را استخراج می‌کند که همه قطعات را تولید کرده‌اند.

فرم صحیح عبارت حساب رابطه‌ای داده شده در گزاره دوم، برای استخراج نام تولیدکنندگانی که همه قطعات را تولید کرده‌اند، به صورت زیر است:

$$\{t | \exists sx \in s(t[sname] = sx[sname] \text{ AND } \forall px \in p (\exists spjx \in spj(spjx[s\#] = sx[s\#] \text{ AND } spjx[p\#] = px[p\#])))\}$$

بنابراین فقط در گزاره اول نتیجه حاصل از اجرای عبارت جبر رابطه‌ای با نتیجه حاصل از اجرای عبارت حساب رابطه‌ای تاپلی یکسان است.

۶- گزینه (۴) صحیح است.

سه جدول ST ، CO و $STCO$ با مقادیر زیر را در نظر بگیرید:

STID	STNAME	STLEVEL	STMAJOR	COID	CONAME	COTYPE	CREDIT
S1	Sn1			C1	Cn1	P	
S2	Sn2			C2	Cn2	P	
S3	Sn3			C3	Cn3	Q	
S4	Sn4			C4	Cn4	Q	
S5	Sn5						

جدول ST

جدول CO

<u>STID</u>	<u>COID</u>	<u>YR</u>	<u>TR</u>	GRADE
S1	C1	94-95	2	19
S1	C3	94-95	2	20
S2	C2	94-95	2	18
S3	C2	94-95	1	17
S4	C3	94-95	2	16

جدول STCO

مطابق پرس و جوی مطرح شده در عبارت B ، داریم:

$$\Pi_{(STNAME)} \left(ST \bowtie (\sigma_{YR='94-95'} \wedge TR='2') (STCO) \bowtie \sigma_{COTYPE='P'} (CO)) \right)$$

ابتدا، در داخلی ترین پرانتز، بخش سمت چپ الحق طبیعی، مشخصات درس‌های انتخاب شده توسط دانشجویان در سال 94-95 در ترم 2 از جدول STCO استخراج می‌گردد، به صورت زیر:

<u>STID</u>	<u>COID</u>	<u>YR</u>	<u>TR</u>	GRADE
S1	C1	94-95	2	19
S1	C3	94-95	2	20
S2	C2	94-95	2	18
S4	C3	94-95	2	16

سپس، خروجی فوق با سطرهایی از جدول CO که ستون COTYPE آن برابر مقدار P است، الحق طبیعی می‌گردد.

سپس، در ادامه، در داخلی ترین پرانتز، بخش سمت راست الحق طبیعی، مشخصات درس‌های عملی که ستون COTYPE آن برابر مقدار P است، از جدول CO استخراج می‌گردد، به صورت زیر:

<u>COID</u>	CONAME	COTYPE	CREDIT
C1	Cn1	P	
C2	Cn2	P	

در نهایت دو بخش سمت چپ و راست در داخلی ترین پرانتز، بر روی ستون مشترک COID، مطابق قوانین عملگر الحق طبیعی، الحق طبیعی می‌گردد.

بنابراین نتیجه داخلی ترین پرانتز، برابر STID یا شماره دانشجویانی خواهد بود که حداقل یک درس عملی (منظور درسی که COTYPE آن برابر 'P' باشد) در سال 94-95 در ترم 2 انتخاب کرده‌اند، به صورت زیر:

<u>STID</u>	<u>COID</u>	<u>YR</u>	<u>TR</u>	GRADE	CONAME	COTYPE	CREDIT
S1	C1	94-95	2	19	Cn1	P	
S2	C2	94-95	2	18	Cn2	P	

توجه: طبق تعریف عملگر الحق طبیعی، در جبر رابطه‌ای ستون یا ستون‌های مشترک فقط یکبار در خروجی ظاهر می‌شود. در اینجا دو جدول CO و STCO دارای یک ستون مشترک یعنی COID هستند. پس خروجی الحق طبیعی در داخلی ترین پرانتز دارای هشت ستون خواهد بود. در ادامه، نتیجه داخلی ترین پرانتز، با جدول ST (جدول دانشجویان) بر روی ستون مشترک STID، مطابق قوانین عملگر الحق طبیعی، الحق طبیعی می‌گردد، به صورت زیر:

STID	STNAME	COID	YR	TR	GRADE	CONAME	COTYPE	...
S1	Sn1			C1	94-95	2	19	Cn1	P	
S2	Sn2			C2	94-95	2	18	Cn2	P	

در نهایت، با اجرای عملگر Π ، نام دانشجویانی که حداقل یک درس عملی (منظور درسی که آن برابر 'p' باشد) در ترم 2 سال 94-95 انتخاب کرده باشند، استخراج می‌گردد، به صورت زیر:

$$\begin{array}{c} \text{STNAME} \\ \hline \text{Sn1} \\ \text{Sn2} \end{array}$$

که مطابق پرس و جوی مطرح شده در صورت سوال است.
مطابق پرس و جوی مطرح شده در عبارت D، داریم:

$$\Pi(\text{STNAME}) \left(\sigma_{\text{YR}='94-95'} \wedge \text{TR}='2' \wedge \text{COTYPE}='p' (\text{ST} \bowtie (\text{STCO} \bowtie \text{CO})) \right)$$

ابتدا، جداول CO، STCO و ST با یکدیگر الحق طبیعی می‌شوند، سپس عملگر انتخاب سطر (σ) اجرا می‌گردد. که خروجی نهایی آن به صورت زیر خواهد بود:

STID	STNAME	COID	YR	TR	GRADE	CONAME	COTYPE	...
S1	Sn1			C1	94-95	2	19	Cn1	P	
S2	Sn2			C2	94-95	2	18	Cn2	P	

در نهایت، با اجرای عملگر Π ، نام دانشجویانی که حداقل یک درس عملی (منظور درسی که آن برابر 'p' باشد) در ترم 2 سال 94-95 انتخاب کرده باشند، استخراج می‌گردد، به صورت زیر:

$$\begin{array}{c} \text{STNAME} \\ \hline \text{Sn1} \\ \text{Sn2} \end{array}$$

که مطابق پرس و جوی مطرح شده در صورت سوال است.
توجه: عبارت D معادل عبارت B است، با این تفاوت که در عبارت B ابتدا عملگر انتخاب روی جداول اجرا می‌شود و سپس نتایج با یکدیگر الحق طبیعی می‌شوند، اما در عبارت D ابتدا جداول با یکدیگر الحق طبیعی می‌شوند و سپس عملگر انتخاب اجرا می‌شود.
مطابق پرس و جوی مطرح شده در عبارت A، داریم:

RANGVAR STX OVER ST
RANGVAR COX OVER CO
RANGVAR STCOX OVER STCO

STX.STNAME WHERE EXISTS STCOX (STCOX.YR='94-95' AND STCOX.TR='2' AND STCOX.STID=STX.STID AND EXIST COX (COX.COVID=STCOX.COVID AND COX.COTYPE='P'))

پرس و جوی مطرح شده در عبارت A به صورت زیر است:

نام دانشجویانی در خروجی ظاهر گردد که به ازای آن وجود داشته باشد (EXISTS اول)، درس اخذ شده‌ای در ترم 2 سال 94-95 که عملی باشد (EXISTS دوم).

به عبارت دیگر به ازای یک دانشجوی خاص، آیا وجود دارد درس اخذ شده‌ای در ترم 2، سال 94-95 که عملی باشد، اگر وجود داشت، نام آن دانشجو پاسخ موردنظر پرس و جو خواهد بود. زیرا به ازای یک دانشجوی خاص وجود دارد حداقل درس اخذ شده‌ای در ترم 2، سال 94-95 که عملی باشد. مطابق پرس و جوی مطرح شده در عبارت A، ما به دنبال نام دانشجویانی هستیم که حداقل یکی از دروس اخذ شده آنها در ترم 2 سال 94-95 عملی باشد.

خروجی پرس و جوی عبارت A به صورت زیر است:

STNAME
Sn1
Sn2

پرس و جو از جدول STX که همان جدول ST است، آغاز می‌گردد. سپس به ازای هر یک از سطرهای جدول STX، کل شرایط مطرح شده، جلوی EXISTS اول و دوم به طور همزمان بررسی می‌گردد، در حال حاضر در سطر اول جدول STX قرار داریم، مطابق شرایط مطرح شده جلوی EXISTS اول که مطابق عبارت STX.STID=STCOX.STID می‌باشد. مقدار STX.STID در سطر اول جدول STX که برابر S1 می‌باشد را بردارید، حال ببرید داخل جدول STCOX و به دنبال STCOX.STID هایی باشید که برابر مقدار S1 باشد و هم در آن سطرها دو شرط 'STCOX.YR='94-95' AND STCOX.TR='2' برقرار باشد که دو سطر اول و دوم از جدول STCOX در این شرایط وجود دارد.

حال به ازای سطر اول و دوم از جدول STCOX، کل شرایط مطرح شده، جلوی EXISTS دوم بررسی می‌گردد. در حال حاضر در سطر اول جدول STCOX قرار داریم، مطابق شرایط مطرح شده جلوی EXISTS دوم که مطابق عبارت STCOX.COVID=COX.COVID می‌باشد، مقدار STCOX.COVID در سطر اول جدول STCOX که برابر C1 می‌باشد را بردارید، حال ببرید داخل جدول COX و به دنبال COX.COVID ای باشید که برابر مقدار C1 باشد و هم در آن سطر (ستون COID کلید اصلی است، بنابراین مقدار تکراری ندارد و دارای خاصیت یکتاپی است) شرط 'COX.COTYPE='P'' برقرار باشد که سطر اول از جدول COX در این شرایط قرار دارد.

حال در سطر دوم جدول STCOX قرار می‌گیریم، مطابق شرایط مطرح شده جلوی EXISTS دوم که مطابق عبارت STCOX.COVID=COX.COVID می‌باشد، مقدار STCOX.COVID در سطر دوم جدول COX که برابر C3 می‌باشد را بردارید، حال ببرید داخل جدول COX و به دنبال COX.COVID ای باشید که برابر مقدار C3 باشد و هم در آن سطر، شرط 'COX.COTYPE='P'' برقرار باشد که همچین سطrix از جدول COX در این شرایط وجود ندارد.

حال اگر به ازای سطر اول از جدول STX با STID برابر مقدار S1 که پرس و جو را نیز با آن آغاز کردیم،

حداقل یکی از دو سطر اول و دوم از جدول STCOX که STID آن‌ها نیز برابر مقدار S1 است، شرط جلوی EXISTS دوم برقرار شود، آنگاه STNAME مربوط به سطر اول از جدول STX مطابق دستور STX.STNAME در خروجی قرار می‌گیرد. که مطابق آنچه پیش‌تر نیز گفتیم، شرایط مذکور برای سطر اول از جدول STCOX ببرقرار بود و برای سطر دوم از جدول STCOX ببرقرار نبود. از آنجاکه حداقل یک سطر در شرایط مذکور قرار دارد، پس خروجی پرس و جو به ازای حرکت در سطر اول از جدول STX به صورت زیر خواهد بود:

STNAME
Sn1

این روند به ازای کلیه سطرهای بعدی جدول STX ادامه دارد، که در صورت ادامه پرس و جو خروجی نهایی به صورت زیر خواهد بود:

STNAME
Sn1
Sn2

به عبارت دیگر، نام دانشجویانی که حداقل یک درس عملی (منظور درسی که COTYPE آن برابر 'P' باشد) در ترم ۲ سال ۹۴-۹۵ انتخاب کرده باشند، استخراج می‌گردد، به صورت زیر:

STNAME
Sn1
Sn2

که مطابق پرس و جوی مطرح شده در صورت سوال است.

توجه: مطابق تعاریف حساب رابطه‌ای تاپلی یا سطري، STX متغیر سطري می‌باشد که بر اساس تعریف آن سطر به سطر بر روی جدول ST حرکت می‌کند، برای سادگی در متن فوق، از STX به عنوان جدولی معادل ST یاد کردیم.

مطابق پرس و جوی مطرح شده در عبارت C، داریم:

RANGVAR STX OVER ST

RANGVAR COX OVER CO

RANGVAR STCOX OVER STCO

STX.STNAME WHERE NOT EXISTS STCOX (STCOX.YR='94-95' AND STCOX.TR='2'

AND STCOX.STID=STX.STID AND NOT EXIST COX (COX.COID=STCOX.COID AND

COX.COTYPE='P'))

پرس و جوی مطرح شده در عبارت C، به صورت زیر است:

نام دانشجویی در خروجی ظاهر گردد که به ازای آن وجود نداشته باشد (NOT EXISTS اول)، درس اخذ شده‌ای در ترم ۲ سال ۹۴-۹۵ که عملی نباشد. (NOT EXISTS دوم)

به عبارت دیگر نام دانشجویی در خروجی ظاهر گردد که همه دروس اخذ شده آن در ترم ۲ سال ۹۴-۹۵ عملی باشد.

توجه: برای خوانایی بهتر، هرگاه دو not exist پشت سرهم در پرس و جوها دیدید، not exists اول را

بخوانید، خواندن را که تمام کردید، که ساده‌تر هم هست، مجدداً همان معنای exist را در نظر بگیرید. برای مثال در اینجا، بینید که به ازای یک دانشجوی خاص، آیا وجود دارد درس اخذ شده‌ای در ترم ۲ سال ۹۴-۹۵ که عملی نباشد. اگر وجود داشت، نام آن دانشجو پاسخ مورد نظر پرس و جو نخواهد بود. زیرا به ازای یک دانشجوی خاص وجود دارد درس اخذ شده‌ای در ترم ۲ سال ۹۴-۹۵ که عملی نیست. مطابق پرس و جوی مطرح شده در عبارت C، ما به دنبال نام دانشجویانی هستیم که همه دروس اخذ شده آنها در ترم ۲ سال ۹۴-۹۵ عملی باشد.

خروجی پرس و جوی عبارت C به صورت زیر است:

STNAME
Sn2

که مطابق پرس و جوی مطرح شده در صورت سوال نیست.
پس عبارات A، B و D مطابق پرس و جوی مطرح شده در صورت سوال است، بنابراین گزینه چهارم پاسخ سوال است.

- گزینه (۳) صحیح است.

سه جدول ST، CO و STCO با مقادیر زیر را در نظر بگیرید:

STID	STNAME	COID	CONAME	CREDIT	
S1	Sn1			C1	Cn1	4	
S2	Sn2			C2	Cn2	3	
S3	Sn3			C3	Cn3	4	
S4	Sn4			C4	Cn4	3	
S5	Sn5						

جدول ST

جدول CO

STID	COID	SEMESTER	GRADE
S1	C1		19
S1	C3		20
S2	C2		18
S3	C2		17
S4	C3		16

جدول STCO

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

```
STX.STNAME
WHERE
NOT EXISTS COX
(COX.CREDIT=4 AND
NOT EXISTS STCOX
(STCOX.STID=STX.STID AND STCOX.COID=COX.COID))
```

پرس و جوی مطرح شده در گزینه‌ی سوم را همانند سورها به صورت زیر بخوانید:

«نام دانشجویانی که، وجود نداشته باشد (NOT EXISTS) درس 4 واحدی که، اخذ نشده باشد (NOT EXISTS). یعنی نام دانشجویانی که همه دروس 4 واحدی را اخذ کرده‌اند.» خروجی پرس و جوی گزینه‌ی سوم به صورت زیر است:

STNAME
Sn1

که مطابق پرس و جوی مطرح شده در صورت سوال است.
مطابق پرس و جوی مطرح شده در گزینه‌ی چهارم، داریم:

```
STX.STNAME
WHERE
EXISTS STCOX
(STCOX.STID=STX.STID AND
EXISTS COX
(COX.COID=STCOX.COID AND COX.CREDIT=4))
```

پرس و جوی مطرح شده در گزینه‌ی چهارم را همانند سورها به صورت زیر بخوانید:
«نام دانشجویانی که، وجود داشته باشد (EXISTS) درس اخذ شده‌ای که، 4 واحدی باشد، (EXISTS). یعنی نام دانشجویانی که حداقل دروس 4 واحدی را اخذ کرده‌اند.» خروجی پرس و جوی گزینه‌ی چهارم به صورت زیر است:

STNAME
Sn1
Sn4

که مطابق پرس و جوی مطرح شده در صورت سوال نیست.

تست‌های فصل ششم: SQL دستورات

(مهندسی IT- دولتی ۸۵)

۱- کدام یک از عبارات زیر نادرست است؟

۱) استفاده از Correlated Query به جای Join موجب پایین آمدن کارآیی (Performance) می‌شود.

۲) هر Query که قابل بیان به صورت یک Join Natural می‌باشد می‌تواند به صورت یک Subquery نیز بیان شود.

۳) هر Query که قابل بیان به صورت یک Subquery می‌باشد می‌تواند به صورت یک Equi Join نیز بیان شود.

۴) هر Query که قابل بیان به صورت یک Equi Join می‌باشد می‌تواند به صورت یک Natural Join نیز بیان شود.

(مهندسی IT- دولتی ۸۵)

۲- پرس و جوی ذیل در یک پایگاه داده مطرح است:

«اسامی کارمندانی که مدیر پروژه نیز می‌باشند را لیست نمایید.»

کدام یک از عبارات ذیل برای فرموله کردن پرس و جوی فوق نادرست است؟

select ename
from emps e
where one proj p
satisfies p.pmgr = e.empno

select ename
from emps e
where some proj p
satisfies p.pmgr = e.empno

select ename
from emps e
where exists(select * from proj p
weher p.pmgr = e.empno)

select ename
from emps e
where e.empno = any
(select p.pmgr from proj p)

(مهندسی کامپیوتر- دولتی ۸۵)

۳- کدام یک از عبارات SQL زیر با بقیه عبارات معادل نمی‌باشد؟

(۱)

select P.*
from part P
where P.P# IN (select SP.P# from supply SP where SP.charge>40000)

(۲)

select P.*
from part P
where P.P#= ALL (select S.P# from supply S where S.charge>40000)

(۳)

Select P.*
from part P
where Exists (select * from supply SP where P.P#=SP.P# and SP.charge>40000)

(۴)

select P.*
from part P
where P.P#=(select SP.P# from supply SP where P.P#=SP.P# and SP.charge> 40000)

۴- رابطه Person شامل شماره شناسایی، نام افراد و شماره شناسایی پدر آن هاست.
 کدام گزینه نام همه پدرهای موجود در رابطه را می دهد؟
 (مهندسی کامپیووتر-دولتی ۷۵)

```
select P1.name
from person P1,Person P2
where P1.father_no = P2.father_no
```

```
select name
from person
where no = father_no
```

```
select P1.name
from Person P1,Person P2
where P1.no = P2.father_no
```

```
select name
from person
where name IN (select name
from person where no = father_no)
```

۵- جدول A دارای ستون Price و Pcode است. کدام گزینه در مورد عبارت زیر صحیح است؟
 (مهندسی کامپیووتر-دولتی ۷۵)

```
select Pcode,Price
from A
where Price = Max (Price)
```

(۱) معادل عبارت مقابل است:

```
select Pcode,Price
from A
where Price = (select Max (Price) from A)
```

(۲) اشکال منطقی دارد زیرا محاسبه ماکریم در مرحله جداگانه‌ای انجام نمی‌شود.
 (۳) همواره یک سطر چاپ می‌کند که غلط است.

(۴) معادل عبارت مقابل است:

```
select Pcode, Max (Price)
from A
where Max (Price) = select Price from A
```

۶- جداول زیر برای یک بانک مفروض است:
 (کارشناسی ارشد- دولتی ۸۲)
 CUSTOMER (CUSTNAME , ADDRESS)
 ACCOUNT (ACCNO , CUSTNAME)
 LOAN(LOANNO , CUSTNAME)

دستور SQL زیر برای تعیین مشتریانی که وام نگرفته‌اند کافی است.

(۱)

```
SELECT * FROM CUSTOMER , LOAN
WHERE CUSTOMER.CUSTNAME <> LOAN.CUSTNAME
```

(۲)

```
SELECT * FROM CUSTOMER , LOAN
WHERE CUSTOMER.CUSTNAME = LOAN.CUSTNAME
```

(۳)

```
SELECT * FROM CUSTOMER
WHERE CUSTNAME NOT IN
(SELECT CUSTNAME FROM LOAN)
```

۴) هیچ کدام

- ۷- ضد نیم اتصال دو رابطه R و S بنابر تعریف شامل تاپل های آویزان R می شود. به این معنا که تاپل هایی که متناظری در S و هم چنین در هیچ رابطه دیگری ندارند. روابط (A,B,C) و (S,A,B,D) را در نظر بگیرید که براساس صفات مشترک A و B متصل می شوند و فرض کنید که $A \rightarrow B$, $A \rightarrow C$ کلیدی برای هر دو رابطه می باشد. کدام گزینه یک ضد نیم اتصال را به طور صحیح پیاده سازی می کند؟ (مهندسی IT-دولتی ۸۱)

(۱) Select R.* from R , S where R.A < \rightarrow S.A AND R.B < \rightarrow S.B

(۲) Select*from R where not exists (Select*from S where S.A= R.A OR S.B=R.B)

(۳) Select*from R where not exists (Select*from S where S.A=R.A AND S.B=R.B)

(۴) Select*from R where A< \rightarrow any (Select A from S) AND B< \rightarrow any (Select B from S)

- ۸- رابطه تولید کننده $S(S\#, City, Status)$ مفروض است. برای بازیابی Status و # مربوط به تولید کنندگان شهر کرمان براساس ترتیب نزولی Status کدام یک از دستورات SQL زیر صحیح می باشد؟

(مهندسی کامپیوترا-دولتی ۷۱)

Select S# , Status Where City = 'Kerman' From S Order By Status Order By Status From S Select Status, S# Where City = 'Kerman'	Select S# , Status From S Where City = 'Kerman' Order By Status Desc Select Status, S# Where City = 'Kerman' Order By Status Desc From S
---	---

- ۹- در صورتی که مفهوم رابطه $SPJ< S\#, P\#, J\#, QTY >$ به شکل ذیل بیان گردد:
«تهیه کننده S#, قطعه # P را برای پروژه # J به تعداد QTY تهیه نموده است» و کلید اصلی رابطه باشد: برای فرموله کردن پرس و جوی زیر به زبان SQL کدام قسمت تکمیلی باید به دستور SQL زیر ملحظ شود؟ (مهندسی IT-دولتی ۸۱)

پرس و جوی:	دستور SQL
شماره پروژه هایی را معین کنید که تمام قطعات آنها را یک تهیه کننده، تهیه کرده باشد.	Select J# From SPJ

Group by P#
Having count (*) = 1 (۲)

Group by S#
Having Cont(*) = 1 (۱)

Group by J#
Having Count (Distinct S#) = 1 (۴)

Group by J# , P#
Having Count (S#) = 1 (۳)

۱۰- جدول محصول به صورت زیر مفروض است:

(مهندسي IT- دولتي ۸۴) goods (P#, Pname, Price)

خروجی دستور زیر کدام گزینه است؟

Select Price form goods

Where Price >= Max (Price)

۲) کل جدول

۱) NULL

۴) ماکریم قیمت در جدول

۳) بستگی به داده ها دارد.

۱۱- کدام یک از گزینه ها نادرست است؟

(مهندسي IT- دولتي ۸۴) ۱) قانون اول و دوم جامعیت داده ای به وسیله تعریف دامنه (domain) برای ویژگی ها قابل انجام است.

۲) برای اعمال قانون اول جامعیت داده ای در SQL کافی است در تعریف ویژگی که به عنوان کلید اصلی است not NULL و unique قید شود.

۳) برای اعمال قانون اول جامعیت داده ای در SQL می بایست در ایجاد جدول کلید اصلی به وسیله primary key تعریف شود.

۴) برای اعمال قانون سوم جامعیت داده ای در SQL می بایست در ایجاد جدول کلید خارجی به وسیله foreign key تعریف شود.

۱۲- اگر employee (empno, empname, dept, salary)

شماره کارمند: empno ، نام کارمند: empname ، واحد: dept و حقوق: salary باشد، آن گاه جواب پرسش

زیر کدام یک از گزینه ها می باشد؟

(مهندسي IT- دولتي ۸۴)

پرسش: مشخصات کارکنانی که حقوقشان از متوسط حقوق کلیه کارکنان سازمان بیشتر است را لیست نمایید.

(۱)

select *

from employee

where salary > avg (salary)

(۲)

select *

from employee

having salary > avg (salary)

(۳)

select *

from employee

where salary >(select avg(salary) from employee)

(۴)

select *

from employee

group by dept having salary > avg (salary)

۱۳- دستور SQL زیر در بانک اطلاعاتی تولیدکنندگان و قطعات چه کار می‌کند؟
 (مهندسى IT-دولتى ۸۶ و آزاد ۸۸)

```
UPDATE P
SET COLOR='Orange'
WHERE COLOR = 'Red'
```

- (۱) تمامی قطعات قرمز رنگ را به رنگ نارنجی در می‌آورد.
- (۲) تمامی قطعات نارنجی رنگ را به رنگ قرمز در می‌آورد.
- (۳) اولین قطعه قرمز رنگ را به رنگ نارنجی در می‌آورد.
- (۴) آخرین قطعه قرمز رنگ را به رنگ نارنجی در می‌آورد.

۱۴- اگر رابطه‌های (SNAME,CITY) و (SP(S#,P#,QTY) مفروض باشد، معادل پرسش زیر کدام گزینه است؟
 (مهندسى IT-دولتى ۸۶)

```
SELECT SNAME, CITY
FROM S
WHERE S# IN (SELECT S#
               FROM SP
               WHERE P#='P1')
```

(۱)

```
SELECT SNAME, CITY
FROM S
WHERE S#=ANY (SELECT S# FROM SP WHERE P# = 'P1')
```

(۲)

```
SELECT SNAME, CITY
FROM S
WHERE EXISTS (SELECT * FROM SP WHERE SP.P#='P1'
AND SP.S#=S.S#)
```

(۳)

```
SELECT SNAME, CITY
FROM S,SP
WHERE S.S#=SP.S# AND SP.P# = 'P1'
```

(۴) هر سه معادل می‌باشند.

۱۵- دو جدول x و y مفروض هستند. در صورت اجرای دستور SQL زیر کادرینالیتی جدول حاصل کدام است؟
 (مهندسى IT-آزاد ۸۵)

Select *
 From y Right outer join x
 On y.B=x.B
 Where y.c Like 'M%'

x	
F	B
f1	b1
f2	b2

3 (۴)

y		
A	B	C
a1	b1	MK
a2	b4	MTK
a1	b2	LN
a2	b1	RP

1 (۲)

2 (۱)

پایگاه داده‌ای دارای سه جدول S, L و G به صورت زیر است:

S(S#,Sname), L(L#,Lname), G(S#,L#,grade)

که در آن جدول S یا جدول دانشجو با خصیصه‌های شماره دانشجویی و نام دانشجو، جدول L یا جدول درس با خصیصه‌های شماره درس و نام درس و جدول G یا جدول نمره با خصیصه‌های شماره دانشجویی، شماره درس و نمره درس است، با توجه به این پایگاه داده به دو سؤال بعدی پاسخ دهید.

(مهندسی IT-آزاد ۸۶)

۱۶- خروجی دستور SQL زیر کدام است؟

```
SELECT DISTINCT Sname
FROM S
WHERE NOT EXISTS (SELECT S# FROM G WHERE L# = 'L1')
```

- (۱) نام تمام دانشجویانی را مشخص می‌کند که از درس L1 نمره دارند.
- (۲) نام بعضی از دانشجویانی را مشخص می‌کند که از درس L1 نمره ندارند.
- (۳) نام تمام دانشجویانی را مشخص می‌کند که از درس L1 نمره ندارند.
- (۴) نام تمام دانشجویانی را مشخص می‌کند که از درس L1 حداکثر یک نمره دارند.

(مهندسی IT-آزاد ۸۶)

۱۷- خروجی عبارت SQL زیر کدام است؟

```
SELECT DISTINCT S#
FROM G
WHERE grade= (SELECT MAX (grade) FROM G)
```

- (۱) شماره تمام دانشجویانی را مشخص می‌کند که بیشترین نمره را اخذ نموده‌اند.
- (۲) شماره بعضی از دانشجویانی را مشخص می‌کند که بیشترین نمره را اخذ نموده‌اند.
- (۳) شماره دانشجویانی را مشخص می‌کند که نمره بیشتری اخذ نموده است.
- (۴) این دستور مشکل گرامری دارد.

(مهندسی IT-دولتی ۸۵)

۱۸- در عبارت در عبارت

```
SELECT State, COUNT (State)
FROM Customer_T
GROUP BY State
HAVING COUNT (State) >1
```

- (۱) می‌توانیم از یک Subquery استفاده کنیم.
- (۲) می‌توانیم از WHERE به جای HAVING استفاده کنیم.
- (۳) می‌توانیم شرط $COUNT (\text{State}) < 10$ را با شرط فعلی ترکیب کنیم.
- (۴) می‌توانیم از $COUNT (\text{State})$ به جای $COUNT (\text{State})$ استفاده کنیم.

۱۹- با اجرای دستور SQL زیر روی بانک اطلاعاتی تولید کنندگان قطعات:

```
INSERT INTO S (S # , SNAME, CITY)
VALUES ('S10' , 'Smith', 'New York')
```

(مهندسی کامپیوترا-دولتی ۸۷)

کدام یک از گزینه‌های زیر صحیح است؟

- (۱) دستور با خطای اجرا مواجه می‌شود.
- (۲) دستور با خطای ترجمه مواجه می‌شود.
- (۳) اعتبار (STATUS) تولید کننده Smith مقدار NULL خواهد شد.
- (۴) اعتبار (STATUS) تولید کننده Smith مقدار قبلی خود را حفظ می‌کند.

(مهندسی کامپیوٹر-دولتی ۸۷)

۲۰- نتیجه دستور SQL زیر چیست؟

```
SELECT P.*  
FROM S, P, SP  
WHERE S. CITY= 'LONDON'  
AND S.S# = SP.S#  
AND P.P# = SP.P#
```

(۱) دستور دارای خطای نحوی است.

(۲) کلیه مشخصات قطعاتی که توسط تولیدکنندگان لندن تولید می شود.

(۳) کلیه مشخصات قطعاتی که حداقل توسط یکی از تولیدکنندگان لندن تولید می شود.

(۴) کلیه مشخصات قطعاتی که دقیقاً توسط یکی از تولیدکنندگان لندن تولید می شود.

(مهندسی IT-دولتی ۸۷)

۲۱- دو جدول زیر را در نظر بگیرید:

loan			borrower	
loan-number	branch-name	amount	customer-name	loan-number
L-170	Downtown	3000	Jones	L-170
L-230	Redwood	4000	Smith	L-230
L-260	Perryridge	1700	Hayes	L-155

حاصل اجرای کدام یک از عبارت‌ها به شکل زیر است؟

loan-number	branch-name	amount	customer-name
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	NULL
L-155	NULL	NULL	Hayes

(۱)

```
select *  
from loan natural left outer join borrower
```

(۲)

```
select *  
from loan full outer join borrower using (loan-number)
```

(۳)

```
select *  
from loan natural inner join borrower
```

(۴)

```
select *  
from loan right outer join borrower  
on (loan. loan-number = borrower. Loan- number)
```

(مهندسی کامپیوٹر-آزاد ۸۷)

۲۲- کدام عبارت SQL معادل عبارت جبر رابطه‌ای زیر است؟

```
 $\Pi_{\text{surname}}[(\text{stud} \times \sigma_{\text{clg name} = 'computer'}(\text{clg}))$   
 $\text{stud.clg\#} = \text{clg.clg\#}$ 
```

(۱)

```
select surname  
from stud, clg  
where clgname = 'computer'
```

(۲)

```
select sname
from stud
where clg# in (select clg#
                  from clg
                  where clgname = 'computer')
```

(۳)

```
select sname
from stud
where clgname = 'computer'
AND clg# = (select clg#
              from clg)
```

(۴)

```
select sname
from stud
group by clg#
having clg# contains (select clg#
                           from clg
                           where clgname = 'computer')
```

(مهندسی کامپیوٹر-آزاد ۸۷)

۲۳ - کدام گزینه معادل سایر گزینه‌ها نیست؟

(۱)

```
select ename
from employee
where not exists (select depnum
                  from dep
                  where depname = 'sale' and dep.depnum = employee.depnum)
```

(۲)

```
select ename
from employee
where depnum in (select depnum
                  from dep
                  where depname = 'sale')
```

(۳)

```
select ename
from employee
where depnum = any (select depnum
                     from dep
                     where depname = 'sale')
```

(۴)

```
select ename
from employee, dep
where employee.depnum = dep.depnum and depnum = 'sale'
```

(مهندسی IT-آزاد ۸۷)

۲۴ - جدول ثبت‌نام به صورت زیر تعریف شده است:

(نمره score، نیمسال term، شماره دانشجویی studnum، شماره گروه درسی secnum)

که در ابتدای ترم، ثبت نام همه دانشجویان در آن وارد می‌شود. فرض کنید استادی در انتهای ترم نمرات دانشجویان را در یک گروه درسی (مثالاً ۱۵۷۷) اعلام کرده است. برای وارد کردن نمرات در این جدول از کدام دستور باید استفاده نمود؟

Select (۴)

Delete (۳)

Update (۲)

Insert (۱)

۲۵- فرض کنید یک بانک اطلاعاتی شامل سه جدول به صورت زیر باشد:

T1 (A,B,C,FK)

T2 (D,E,W,FK)

T3 (FK,X,Y,Z)

که FK در جدول T3 کلید اصلی و در جدول T1 و T2 کلید خارجی است. (T1 و T2 به جزء FK کلید خارجی دیگری ندارند). فرض کنید به هنگام تعریف جداول، برای FK در T1 عبارت `on update cascade` ذکر شده باشد، اما برای FK در T2 ذکر نشده باشد. اگر دستور زیر را اجرا کنیم چه اتفاقی خواهد افتاد؟ (مهندسی IT-آزاد ۸۷)

`update T3``set FK=15``where FK=10`

(۱) مقدار FK فقط در جدول T3 به روز می‌شود.

(۲) مقدار FK در جداول T1 و T3 به روز می‌شود.

(۳) مقدار FK در هیچ یک از جداول به روز نمی‌شود.

(۴) مقدار FK در هر سه جدول به روز می‌شود.

۲۶- با توجه به جداول داده شده، کدام عبارت SQL معادل عبارت جبر رابطه ای زیر است؟

(مهندسی IT-آزاد ۸۷)

`prof (pname, pname , esp, depnum) dep (depid, depname)``Πpname(σdepname='computer'(prof ⋈ dep))`

(۱)

```
select pname
from prof
where depnum in (select depnum
                  from dep
                  where depname= 'computer')
```

(۲)

```
select pname
from prof, dep
where depname = 'computer'
```

(۳)

```
select pname
from prof
where prof.depnum=dep.depid and depname in (select depname
                                              from dep
                                              where depname= 'computer')
```

(۴)

```
select pname
from prof
group by depnum
having depnum= (select depnum
                 from dep
                 where depname= 'computer')
```

۲۷- در SQL شرط **R EXISTS** چه موقع صحیح (True) می‌شود؟ (مهندسى کامپیوتر-دولتى ۸۷)

- (۱) اگر R یک دید (view) باشد.
- (۲) اگر R یک رابطه پایه (base relation) باشد.
- (۳) اگر R حداقل یک سطر داشته باشد.

۲۸- اگر عنوان رابطه r(S) و رابطه‌های $S = (A, B, C, D)$ و $r = (R)$ را داشته باشیم. عمل تقسیم

(مهندسى کامپیوتر-دولتى ۸۸) $r \div_S$ توسط کدام مورد محقق می‌گردد؟

(select A,B from r)	
except	with rl (A,B) as
(with rl(A,B) as	select A,B
select A,B	from r
from r	select A,B
with r2 (A,B,C,D) as	from rl
(۲)	where exists (select C,D
select *	from s
from rl,s	where (A,B,C,D) in r)
select A,B	
from (r2 except r))	
(select A,B from r)	with rl (A,B) as
except	select A,B
(with rl(A,B) as	from r
select A,B	select A,B
from r	from rl
with r2(A,B,C,D) as	where not exists (select C,D
(۴)	from s
select *	where (A,B,C,D) in r)
from rl,s	except s)
select A,B	
from (r intersect r2))	

۲۹- کدام گزینه پاسخ پرس‌وجوهی «ترم‌هایی که دانشکده کامپیوتر درس ارایه کرده است» می‌باشد؟

(مهندسى کامپیوتر-آزاد ۸۸)

select term	select distinct term
from sec,clg,crs	from sec, clg
where clg.clg# = crs.clg# (۲	where clg.pname = sec.pname
and crs.c# = sec.c#	and clgname = "کامپیوتر"
group by term	
select distinct term	select term
from sec,clg,crs	from clg,crs
where clg.clg# = crs.clg# (۴	where clg.clg# = crs.clg#
and crs.c# = sec.c#	having clgname = "کامپیوتر"
and clgname = "کامپیوتر"	

(مهندسی کامپیوتر-آزاد ۸۸)

۳۰- خروجی دستور زیر کدام است؟

```
select sname
from stud
where avg > AVG (avg)
```

(۱) دانشجویانی که معدل آنها از نیمی از معدل‌ها بیشتر است.

(۲) این دستور غلط است.

(۳) دانشجویانی که معدل آنها از میانگین همه معدل‌ها بیشتر است.

(۴) دانشجویانی که معدل ترم جاری آنها از میانگین همه معدل‌ها بیشتر است.

(مهندسی IT-آزاد ۸۸)

۳۱- خروجی پرس‌وجوی زیر چیست؟

```
select distinct essn
from work-on
where pno in (select pno
               from work-on
               where name = 'Parviz')
```

(۱) اسمی کارفرمایان پروژه‌هایی که پرویزی روی آنها کار می‌کند.

(۲) اسمی کارفرمایان کارمندانی که روی یکی از پروژه‌هایی کار می‌کنند که پرویزی روی آنها کار می‌کند.

(۳) اسمی کارفرمایان کارمندانی که دقیقاً روی همان پروژه‌هایی کار می‌کنند که پرویزی روی آنها کار می‌کند.

(۴) اسمی کارفرمایان کارمندانی که روی تمام پروژه‌هایی که پرویزی روی آنها کار می‌کند، کار می‌کند.

۳۲- کدام گزینه پاسخ پرس‌وجوی «مشخصات دانشجویانی که معدل‌شان از متوسط معدل دانشجویان

(مهندسی IT-آزاد ۸۸)

دانشکده‌شان بیشتر است» می‌باشد؟

(۱)

```
select *
from stud
having average > AVG (dept.average)
```

(۲)

```
select *
from stud
group by dept
having average > AVG (average)
```

(۳)

```
select *
from stud S1
where average > (select AVG (average)
                  from stud S2
                  where S1.dept=S2.dept)
```

(۴)

```
select *
from stud
where average > (select AVG (average)
                  from stud)
```

۳۳- در بانک زیر کدام گزینه پاسخ پرس و جوی «نام ملوان‌هایی که همه قایق‌ها را رزرو کرده‌اند» می‌باشد؟

(مهندسی کامپیوتر-دولتی ۸۱)

ملوان Sailor (sid , sname , rating , age)

قایق Boats (bid , bname , color)

رزرو Reserves(sid , bid , day)

select S.sname from sailor S where not Exists ((select B.bid from Boats B) Except

(select R.bid from Reserves R

(۱)

where S.sid=R.sid))

select S.name from Sailor S where not Exists (select B.bid from Boats B where

not Exists (select R.bid from Reserves R where

(۲)

R.bid=B.bid and R.sid=S.sid))

(۳) موارد ۱ و ۲

(۴) هیچ کدام

۳۴- در مورد عبارت SQL زیر کدام یک از گزینه‌ها صحیح است؟

(مهندسی کامپیوتر-دولتی ۸۵)

(الف)

select branch-name, avg (balance) from account group by

(ب)

branch-name having avg (balance)> 1200

(ج)

select branch-name from Branch where assets > Any (select assets from

Branch where branch-city= 'ISFAHAN')

(۱) فقط عبارت الف صحیح است.

(۲) فقط عبارت الف و ب صحیح است.

(۳) هر سه عبارت الف، ب و ج صحیح است. (۴) لااقل یکی از عبارات الف، ب و ج غلط است.

۳۵- دستور SQL زیر در بانک اطلاعاتی تولید کنندگان و قطعات چه کار می‌کند؟ (مهندسی کامپیوتر-دولتی ۸۶)

select SP.p# , sum (SP.QTY) As TOTQTY

from SP

group by SP.p#

(۱) شماره هر قطعه و تعداد کل قطعات تولید شده را می‌دهد.

(۲) شماره هر قطعه و تعداد کل آن قطعه تولید شده را می‌دهد.

(۳) شماره هر قطعه و وزن کل قطعات تولید شده را می‌دهد.

(۴) شماره هر قطعه و وزن کل قطعات تولید شده توسط تولیدکننده آن قطعه را می‌دهد.

۳۶- دستور SQL زیر در بانک اطلاعاتی تولیدکنندگان و قطعات چه بر می‌گرداند؟ (مهندسی کامپیوتر-دولتی ۸۶)

select S.name from S where S.s# in (select SP.s# from SP where SP.p#= 'p2')

(۱) اسمی تولیدکنندگانی که قطعه شماره 'p2' را تولید می‌کنند. (احتمالاً با تکرار)

(۲) اسمی تولیدکنندگانی که قطعه شماره 'p2' را تولید می‌کنند (بدون تکرار).

(۳) اسمی تولیدکنندگانی که فقط قطعه 'p2' را تولید می‌کنند.

(۴) اسمی تولیدکنندگانی که علاوه بر قطعه شماره 'p2' قطعه دیگری نیز تولید می‌کنند.

۳۷- پایگاه داده‌ای با دستورات زیر ایجاد شده است:

```
create table Books (isbn char (20) primary key,
                    bname char (50),
                    type char (9) check (type in ('Techincal', 'fiction', 'self-help'))
                    );
```

```
create table Authors (ssn char (9),
                     isbn char (20),
                     primary key (ssn, isbn));
```

کدام یک از گزینه‌های زیر برای پیاده‌سازی پرس‌وجوی «شماره ssn را برای نویسنده‌گانی که هیچ کتابی از نوع fiction نوشته‌اند بیابید» با استفاده از SQL نادرست است؟

(۱)

```
select ssn from Authors
Except (select ssn from Authors A, Books B
        where A.isbn=B.isbn and B.type= 'fiction')
```

(۲)

```
select ssn from Authors, Books
where Authors.isbn=Books. isbn and type <> 'fiction'
```

(۳)

```
select ssn from Authors A
where Not Exists (select * from Books B
                   where A.isbn=B.isbn and B.type= 'fiction')
```

(۴)

```
select ssn from Authors where ssn Not in
(select ssn from Authors A, Books B
where A.isbn=B.isbn and B.type= 'fiction')
```

۳۸- پایگاه داده‌ای با جداول زیر را در نظر بگیرید. حاصل کدام یک از پرس‌وجوهای زیر با بقیه متفاوت است؟

(۸۷) (مهندسی IT - دولتی)

```
account (acc#, bname , balance)      depositor (cname , acc#)
```

```
select cname
from depositor as T, account
where T.acc#=account.acc#and
      bname = 'perryridge'          (۲)
group by cname
having count (T.acc#) <= 1
```

```
select cname
from depositor as T,account
where T.acc#=account.acc#and
      bname = 'perryridge'          (۱)
group by cname
having count (*)<=1
```

```
select T.cname
from depositor as T where unique
(select R cname
from account, depositor as R           (۴)
where T.cname = R.cname and
      R.acc#=account.acc#and
      bname = 'perryridge')
```

```
select cname
from depositor as T, account
where T.acc#=account.acc#and
      bname = 'perryridge'          (۳)
group by cname
having count (Distinct bname)<=1
```

۳۹- پایگاه داده‌ای با جداول زیر ایجاد شده است:

Mهندسی IT- دولتی ۸۷
 Branch (branch-name , branch-city , assets)
 موجودیت شعب بانک
 Account (account-number , branch-number , balance)
 موجودیت حساب پس‌انداز
 depositor (customer-name , account-number)
 ارتباط حساب پس‌انداز با نام مشتریان
 آنگاه شرح مناسب برای Query روبرو کدام گزینه می‌باشد؟

```
select distinct S.customer-name
from depositor as S
where not exists (
  select branch-name
  from Branch
  where branch-city= 'Brooklyn')
Except
(select R.branch-name
from depositor as T, account as R
where T.account-number= R.account-number
and S.customer-name= T.customer-name))
```

- (۱) اسمی مشتریانی را می‌دهد که در تمام شعب شهر Brooklyn حساب باز کرده‌اند.
- (۲) اسمی مشتریانی را می‌دهد که در هیچ یک از شعب شهر Brooklyn حساب باز نکرده‌اند.
- (۳) اسمی مشتریانی را می‌دهد که در شعب شهر Brooklyn به صورت تکراری حساب باز کرده‌اند.
- (۴) اسمی مشتریانی را می‌دهد که در شعب شهر Brooklyn به صورت غیرتکراری حساب باز کرده‌اند.

(Mهندسی کامپیوتر- دولتی ۸۹)

۴۰- رابطه $R(a,b,c)$ را در نظر بگیرید:

- (۱) پاسخ Q_1 زیر مجموعه‌ای از پاسخ Q_2 است.
- (۲) Q_1 و Q_2 پاسخ‌های یکسان تولید می‌کنند.
- (۳) پاسخ Q_2 زیر مجموعه‌ای از پاسخ Q_1 است.
- (۴) Q_1 و Q_2 پاسخ‌های متفاوت تولید می‌کنند.

بانک اطلاعاتی تولیدکنندگان و قطعات شامل روابط زیر است:

S(S#, SNAME , STATUS , CITY) تولیدکننده

S# : شماره تولیدکننده

SNAME : اسم تولیدکننده

STATUS : اعتبار تولیدکننده

CITY : شهر تولیدکننده

P(P#, PNAME , COLOR , WEIGHT , CITY) قطعه

P# : شماره قطعه

PNAME : اسم قطعه

COLOR : رنگ قطعه

WEIGHT : وزن قطعه

CITY : شهر محل انبار کردن قطعه

SP(S#, P# , QTY) تولید

تعداد قطعه شماره $P\#$ تولید شده توسط تولیدکننده شماره $S\#$ است.

(مهندسي IT - دولتي ۸۹) ۴۱ - جمله SQL زير در بانک اطلاعاتی تولید کنندگان و قطعات چه کار می کند؟

```
SELECT DISTINCT SNAME
FROM S
WHERE S.S# IN (SELECT SP.S#
                FROM SP
                WHERE SP.P# IN (SELECT P.P#
                                FROM P
                                WHERE PCOLOR= 'RED'))
```

- (۱) اسمی تولید کنندگانی که بیش از یک قطعه قرمز رنگ تولید می کنند.
- (۲) اسمی تولید کنندگانی که حداقل یک قطعه قرمز رنگ تولید می کنند.
- (۳) اسمی تولید کنندگانی که حداقل یک قطعه قرمز رنگ تولید می کنند.
- (۴) هیچ کدام

(مهندسي IT - دولتي ۸۹) ۴۲ - جمله SQL زير در بانک اطلاعاتی تولید کنندگان و قطعات چه کار می کند؟

```
SELECT S.SNAME
FROM S
WHERE EXISTS (SELECT *
                FROM SP
                WHERE SP.S# = S.S# AND SP.P# = 'P2')
```

- (۱) اسمی تولید کنندگانی که قطعه‌ای غیر از 'P2' هم تولید می کنند.
- (۲) اسمی تولید کنندگانی که قطعه 'P2' را تولید می کنند.
- (۳) اسمی تولید کنندگانی که قطعه 'P2' را تولید نمی کنند.
- (۴) اسمی تولید کنندگانی که قطعه 'P2' را تولید می کنند بدون تکرار.

(مهندسي IT - آزاد ۸۹) ۴۳ - کدام عبارت پاسخ پرس و جوی زير را به درستی بيان می کند:

I. select * مشخصات قطعاتی که با دور 1000 یا 2000 کار می کنند؟

```
from product
where spin = 1000 or spin= 2000
```

```
II. select *
from product
where spin in (1000, 2000)
```

```
III. select *
from product
where spin=1000
union
select *
from product
where spin=2000
```

I , II , III (۴)

III , II (۳)

I,II (۲)

I (۱)

(مهندسی IT-آزاد ۸۹)

۴۴- خروجی دستور زیر چند سطر خواهد داشت؟

```
select T1.B
from T1,T2
where T1.B=T2.B
```

15 (۴)

A	B
a1	b1
a2	b2
a3	b3

5 (۳)

B	C
b1	c1
b1	c2
b2	c2
b2	c3
b2	c4

3 (۲)

2 (۱)

(مهندسی IT-آزاد ۸۹)

۴۵- خروجی پرس‌وجوی زیر چیست؟

```
select distinct pname
from enroll
where crsno in (select crsno
                  from enroll
                  where sname= 'Kazemi')
```

۱) اسمای استادانی که یکی از درس‌هایی را تدریس می‌کنند که کاظمی در آن درس‌ها ثبت‌نام کرده است.

۲) اسمای استادان دانشجویی به نام کاظمی

۳) اسمای استادانی که دقیقاً همان درس‌هایی را تدریس می‌کنند که کاظمی در آن درس‌ها ثبت‌نام کرده است.

۴) اسمای استادانی که تمام درس‌هایی را تدریس می‌کنند که کاظمی در آن درس‌ها ثبت‌نام کرده است.

۴۶- فرض کنید که $S=(A,C)$ و $R=(A,B)$ و $r(S)$ و $r(R)$ دو رابطه باشند. عبارت $\{<a> | \exists c (<a,c> \in S \wedge \exists b_1, b_2 (<a,b_1> \in r \wedge <c,b_2> \in r \wedge b_1 > b_2))\}$ در حساب رابطه‌ای

دامنه‌ای (domain relational calculus)، معادل کدام یک از پرس‌وجوهای SQL زیر نیست؟

(مهندسی IT-دولتی ۹۰)

```
SELECT DISTINCT A
FROM s
WHERE (A,C) IN {
    SELECT r1.A, r2.A
    FROM r as r1, r as r2
    WHERE r1.B > r2.B
}
```

(۱)

```
SELECT DISTINCT A
FROM s
WHERE EXIST {
    SELECT *
    FROM r as r1, r as r2
    WHERE r1.A = s.C AND r2.A = s.A
    AND r2.B > r1.B
}
```

(۲)

```

SELECT DISTINCT A
FROM(r NATURAL INNER JOIN s) as
res (A,B,C)
WHERE EXIST{
    SELECT *
    FROM r
    WHERE r.A = res.C AND r.B > res.B
}

```

(۳)

```

SELECT DISTINCT r1.A
FROM r as r1, r as r2
WHERE r1.B > r2.B AND
(r1.A, r2.A)IN {
    SELECT *
    FROM s
}

```

(۴)

-۴۷- فرض کنید در یک دفتر مسافرتی نام مسافری به نام «اکبرزاده» به اشتباه «اکبرزاده» وارد جدول مشخصات مسافران شده است، با کدام یک از دستورهای زیر می‌توان این اشتباه را تصحیح کرد؟ (مهندسی IT-آزاد ۹۰)

Select (۴)

Delete (۳)

Update (۲)

Insert (۱)

-۴۸- اگر جدول ثبت‌نام به صورت زیر باشد: Enroll (s#, c#, term) کدام گزینه پاسخ پرس‌وجوی «تعداد دروسی که در نیمسال ۸۹۲ ارائه شده‌اند به روش SQL می‌باشد؟ (مهندسی IT-آزاد ۹۰)

```

select distinct (count c#)
from Enroll
where term = 892

```

```

select distinct(c#)
from Enroll
where term = 892

```

```

select count (distinct c#)
from Enroll
where term = 892

```

```

select count(c#)
from Enroll
where term = 892

```

-۴۹- کدام عبارت پاسخ پرس‌وجوی زیر را به درستی بیان می‌کند؟ (مهندسی IT-آزاد ۹۰)
«مشخصات تولید کنندگانی که در شهر همدان یا تبریز مستقرند».

I. select *
from producer
where city= 'Hamadan' or city= 'Tabriz'
II. select *
from producer
where city in ('Hamadan', 'Tabriz')
III. select *
from producer
where city= 'Hamadan'
union
select *
from producer
where city= 'Tabriz'

I , II , III (۴)

I (۳)

II (۲)

I , II (۱)

۵۰- خروجی دستور زیر چند سطر خواهد داشت:

(مهندسی IT-آزاد ۹۰)
 Select T1.B
 from T1 left Inner Join T2
 where T1.B=T2.B

T1	T2			
A	B	B	C	
a ₁	b ₁	b ₁	c ₁	6 (۱)
a ₂	b ₂	b ₁	c ₂	2 (۲)
a ₃	b ₃	b ₁	c ₂	5 (۳)
		b ₂	c ₃	15 (۴)
		b ₂	c ₄	

(مهندسی IT-آزاد ۹۰)

۵۱- خروجی پرس و جوی زیر چیست؟

```
select distinct pname
from enroll
where crsno in (select crsno
                  from enroll
                  where sname = 'zakeri'
```

۱) اسم فقط یکی از استادان دانشجویی به نام ذاکری.

۲) اسمی استادانی که یکی از درس هایی را تدریس می کنند که ذاکری در آن درس ها ثبت نام کرده است.

۳) اسمی استادانی که دقیقاً همان درس هایی را تدریس می کنند که ذاکری در آن درس ها ثبت نام کرده است.

۴) اسمی استادانی که تمام درس هایی را تدریس می کنند که ذاکری در آن درس ها ثبت نام کرده است.

۵۲- با توجه به شش رابطه زیر:

Branch-schema = (branch-name, branch-city, assets)
 Customer- schema = (customer-name, customer-street, customer - city)
 Loan-schema= (loan-number, branch- name, amount)
 Borrower-schema= (customer-name loan- number)
 Account-schema= (account-number, branch-name, balance)
 Depositor-schema= (customer-name, account-number)

برای پرس و جوی «نام همه مشتریانی که در همه شعب Brooklyn حساب دارند». برای این که SQL زیر جواب درستی برای سؤال مطرح شده باشد، کدام یک از عملگرهای SQL گزینه های درست برای متغیرهای (مهندسی IT-دولتی ۹۱) XX و YY در کد SQL زیر هستند؟

```
select distinct S.customer-name
from depositor as S
where XX (
  select branch-name
  from branch
  where branch-city= 'Brooklyn')
YY
```

```
(select R.branch-name
from depositor as T, account as R
where T.account-number= R.account-number
and S.customer-name= T.customer-name)
)
```

XX = exists, YY = except (۲)

XX = unique, YY = union (۱)

XX = not unique , YY = union (۴)

XX = not exists, YY = except (۳)

۵۳- فرض کنید بانک اطلاعاتی یک اپراتور مخابراتی حاوی جداول ذیل است.

(نام کارمند: empname، کد پرسنلی: emp#) کارمندان

(نام صاحب خط: custName، شماره تلفن ثابت: #fixtel) تلفن های ثابت

(نام صاحب خط: custName، شماره تلفن موبایل: #Mtel) Mobtel تلفن های موبایل

کدام یک از دستورات SQL زیر برای استخراج «لیست کارمندانی از شرکت که تلفن موبایل آن اپراتور (مهندسی کامپیوتر-آزاد ۹۱) را ندارند» درست است؟

(۱)

select * from emp where empname
not in (select custName from Mobtel)

(۲)

select * from emp, Mobtel
where emp.empname <> Mobtel.custName

(۳)

select * from emp, Mobtel
where emp.empname= Mobtel.custName

(۴)

select * from emp, fixtel
where emp.empname <> fixtel.custName

با توجه به بانک زیر، به سؤال زیر پاسخ دهید.

Person (pID , pName , sex, DoB)

اطلاعات فرد شامل شناسه pID، نام pName، جنسیت sex و تاریخ تولد DoB

Rel (pID1, pID2 , pRel1)

اطلاعات ارتباط بین افراد- رابطه pRel بین شخص pID1 و pID2 برقرار است.

به طور خاص، pRel=1 به این معنی است که شخص با شناسه pID2 پدر شخص با شناسه pID1 است.

(مهندسی کامپیوتر- دولتی ۹۶)

۵۴- دستور SQL زیر تاریخ تولد کدام گزینه را می دهد؟

```
SELECT DoB
FROM Person
WHERE pName = x AND pID IN (SELECT pID2
                             FROM Rel
                             WHERE pRel=1)
```

x) فرزند (۲)

x) پدر (۱)

۴) x در صورتی که پدر داشته باشد.

۳) x در صورتی که فرزند داشته باشد.

۵۵- شمای پایگاه داده زیر را در نظر بگیرید:

Employee(eid,NN,ename) : کارمندان

(هر کارمند یک شماره یکتا (eid) دارد و برای هر کارمند شماره ملی (NN) و نام (ename) او را نگه می‌داریم.)

Department(did , dname, address)

(هر بخش یک شماره یکتا (did) دارد و برای هر بخش نام (dname) و آدرس (address) آن را نگه می‌داریم.)

WorksIn(eid , did , salary)

(هر کارمند در چه بخشی کار می‌کند و حقوق (salary) او چقدر است).
پرس‌وجوی SQL زیر را در نظر بگیرید.

SELECT AVG (R.s)

```
FROM (SELECT SUM (W.salary) AS s
      FROM Employee E , WorksIn W
     WHERE E.eid=W.eid
   GROUP BY W.did
UNION ALL
SELECT 0 AS s
FROM Department D
WHERE NOT EXISTS (SELECT *
                      FROM WorksIn W1
                     WHERE W1.did =D.did)) AS R;
```

نتیجه اجرای این پرس‌وجو بر نمونه زیر از جدول‌های Employee و Department و WorksIn چه خواهد بود؟

Employee		
eid	NN	ename
e1	111111	John
e2	222222	Jack
e3	333333	Bob
e4	444444	Joe

2 (۴)

Department		
did	dname	address
d1	Books	a1
d2	Electronics	a2
d3	Health	a3

3 (۳)

WorksIn		
eid	did	Salary
e1	d1	1
e2	d1	3
e3	d1	4
e4	d3	4

4 (۲)

6 (۱)

۵۶- با داشتن رابطه person که اطلاعات افراد در آن ذخیره می‌شود و صفت‌های MotherID و FatherID که به رکوردهای همین رابطه اشاره می‌کنند، پرس‌وجوی SQL زیر چه جوابی را برمی‌گرداند؟

```
SELECT C.Name AS Name, F.Name AS Father, M.Name AS Mother
  FROM person AS C LEFT OUTER JOIN person AS F ON C.FatherID=F.PersonID
 INNER JOIN person AS M ON C.MotherID=M.PersonID
```

۱) همه پدرهایی که برای آن‌ها فرزند ثبت نشده است.

- ۲) همه مادرهایی که برای آنها فرزند ثبت نشده است.
 ۳) همه افراد، حتی آنها باید که مادر برای آنها ثبت نشده است.
 ۴) همه افراد، حتی آنها باید که پدر برای آنها ثبت نشده است.

-۵۷- هر تاپل در رابطه supplier (sid, pid, sdate) نشان می‌دهد که تهیه کننده sid قطعه pid را در تاریخ sdate تهیه کرده است. کلید اصلی این رابطه ترکیب هر سه صفت تشکیل دهنده آن است. کدام پرس و جوی SQL زیر معادل این درخواست است؟
 (مهندسى IT - دولتى ۹۳)

» تهیه کنندگانی که بعضی قطعات را بیش از ده بار تهیه کرده‌اند.«

select distinct sid from supplier where count(*) > 10	select distinct sid from supplier group by sid,pid having count(*) > 10
---	--

select distinct sid from supplier where count(distinct sdate) > 10 group by sid,pid	select distinct s1.sid from supplier as s1,supplier as s2 where count(*) > 10 and s1.pid = s2.pid
--	---

-۵۸- سه رابطه suppliers (تهیه کنندگان)، parts (قطعات) و catalog (کاتالوگ) را در نظر بگیرید:

suppliers (sid, sname)

parts (pid, pname)

catalog (sid, pid)

sid در catalog کلید خارجی و ارجاع دهنده به suppliers و pid در catalog کلید خارجی و ارجاع دهنده به parts می‌باشد. وجود رکورد (s1 , p1) در جدول catalog نشان دهنده این است که تهیه کننده s1 قطعه p1 را تولید می‌کند.
 (مهندسى کامپیوترا - دولتى ۹۴)

اگر بخواهیم دستور SQL زیر مشخصات قطعاتی را بدست آورد که توسط بعضی از تهیه کنندگان تولید نمی‌شود، در دو جای خالی (A) و (B) چه عملگرهایی باید قرار گیرد؟

```
SELECT *
FROM parts P
WHERE .....(A).....(SELECT S.sid
                      FROM suppliers S
                      WHERE ....(B)..(SELECT *
                                    FROM catalog C
                                    WHERE S.sid=C.sid AND P.pid = C.pid))
```

EXISTS : (B) EXISTS : (B) NOT EXISTS : (B) NOT EXISTS : (B)	NOT EXISTS : (A) (۱) EXISTS : (A) (۲) EXISTS : (A) (۳) NOT EXISTS : (A) (۴)
--	--

۵۹- با فرض وجود جداول S , P , J و SPJ به صورت زیر:

$S(S\#, SNAME, STATUS, CITY)$

جدول تولید کنندگان

$P(P\#, PNAME, WEIGHT, CITY)$

جدول قطعات

$J(J\#, JNAME, CITY)$

جدول پروژه‌ها

$SPJ(S\#, P\#, J\#, QTY)$

جدول تعداد قطعات تولید شده توسط تولیدکنندگان در پروژه‌ها

نتیجه اجرای پرس‌وجوی SQL زیر بر روی جداول فوق، کدام یک از عبارات زیر است؟

```
SELECT S.SNAME
FROM S NATURAL JOIN SPJ
WHERE SPJ.J#='j2'
GROUP BY S#, S.SNAME
HAVING SUM (QTY)<=All (SELECT SUM(QTY)
                           FROM SPJ
                           WHERE SPJ.J#='j2'
                           GROUP BY S#)
```

(۱) نام تولیدکنندگانی که تعداد قطعه تولیدی آنها در پروژه کد 2 از مجموعه همه قطعات تولید شده توسط دیگر تولیدکنندگان بیشتر است.

(۲) نام تولیدکنندگانی که کمترین تعداد قطعه (از تمام انواع قطعات) را برای پروژه کد 2 تولید کرده‌اند.

(۳) نام تولیدکنندگانی که بیشترین تعداد قطعه (از تمام انواع قطعات) را برای پروژه کد 2 تولید کرده‌اند.

(۴) نام تولیدکنندگانی که هیچ قطعه‌ای از پروژه کد 2 تولید نکرده‌اند.

۶۰- با توجه به پایگاه داده زیر، دستور SQL برای یافتن «نام اعضا»ی که همه کتاب‌های منتشر شده توسط

McGraw-Hill را به امانت برده‌اند»، کدام است؟

Member (mem-no, name, age)

Book (ISBN, title, author, publisher)

Borrowed (mem-no, ISBN, date)

(۱)

SELECT A.name FROM Member A WHERE

NOT EXISTS $\left(\begin{array}{l} \text{SELECT isbn FROM Book, Borrowed B WHERE publisher = McGraw - Hill} \\ \text{AND B.mem - no} \neq A.mem - no \end{array} \right)$

(۲)

SELECT A.name FROM Member A WHERE

NOT EXISTS $\left(\begin{array}{l} \text{SELECT isbn FROM Book, Borrowed B WHERE publisher = McGraw - Hill} \\ \text{AND B.mem - no} = A.mem - no \end{array} \right)$

(۳)

SELECT A.name FROM Member A WHERE

NOT EXISTS $\left(\begin{array}{l} \left(\text{SELECT isbn FROM Book} \right) \text{EXCEPT } \left(\text{SELECT isbn FROM Borrowed B} \right) \\ \text{WHERE publishe = McGraw - Hill} \end{array} \right)$

(۴)

SELECT A.name FROM Member A WHERE

NOT EXISTS $\left(\begin{array}{l} \left(\text{SELECT isbn FROM Borrowed B} \right) \text{EXCEPT } \left(\text{SELECT isbn FROM Book} \right) \\ \text{WHERE B.mem - no} = A.mem - no \end{array} \right)$

۶۱- جدول S جدول تولیدکنندگان، جدول P جدول قطعات و جدول SP جدول ارتباط دهنده آن دو جدول است که مشخص می کند چه تولیدکننده‌ای چه قطعه‌ای تولید کرده است. کدام گزینه شماره تولیدکنندگانی را می‌دهد که همه قطعات را تولید کرده‌اند؟
 (مهندسی IT-دولتی ۹۵)

S (S#, SNAME, CITY)
 P (P#, PNAME, COLOR)
 SP (S#, P#)

(۱)

SELECT S# FROM S WHERE S# IN (SELECT S# FROM SP)

(۲)

SELECT S# FROM SP GROUP BY S# HAVING COUNT(P#) =
 (SELECT COUNT(P#) FROM SP)

(۳)

SELECT S# FROM S WHERE EXISTS (SELECT P# FROM P
 WHERE EXISTS (SELECT * FROM SP
 WHERE S.S# = SP.S# AND P.P# = SP.P#))

(۴)

SELECT S# FROM S WHERE NOT EXISTS (SELECT P# FROM P
 WHERE NOT EXISTS (SELECT * FROM SP
 WHERE S.S# = SP.S# AND P.P# = SP.P#))

۶۲- سه رابطه‌ی قایق، قایقران و رزرو را در نظر بگیرید. می‌خواهیم نام قایقران‌هایی را بیابیم که همه قایق‌ها را رزرو کرده‌اند. کدام مورد، دستور SQL درست برای این درخواست است؟
 (مهندسی کامپیوترا-دولتی ۹۶)

قایقران : Sailors (sid , sname)
 رزرو :Reservations(sid ,bid , day)
 قایق : Boats (bid , bname , color)

(۱)

Select sname from Sailors S where not exists
 ((Select bid from Boats B) except
 (Select bid from Reservations R where R.bid = B.bid AND R.sid = S.sid))

(۲)

Select sname from Sailors S where exists
 ((Select bid from Boats B) except
 (Select bid from Reservations R where R.bid = B.bid AND R.sid = S.sid))

(۳)

Select sname from Sailors S where not exists
 (Select bid from Boats B where not exists
 (Select * from Reservations R where R.bid = B.bid AND R.sid = S.sid))

(۴)

Select sname from Sailors S where exists
 (Select bid from Boats B where exists
 (Select * from Reservations R where R.bid = B.bid AND R.sid = S.sid))

۶۳- کدام مورد، لیست نام دانشجویان به همراه معدل وزنی نمرات دروس اخذ شده آنها را در ترم ۹۵-۹۶-۱ (مهندسی IT-دولتی) می دهد؟

دانشجویان : ST (STID, STName,...)

دروس :CO (COID, COName, Credit,...)

STCO (STID, COID, Semester, Grade) دروس انتخاب شده توسط دانشجویان

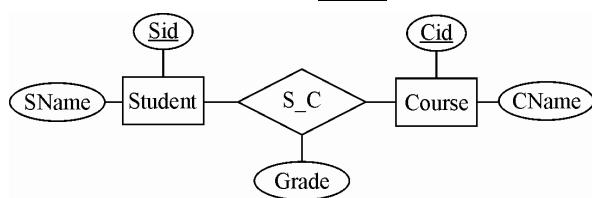
```
Select STName, SUM(Grade * Credit) / SUM(Credit)
From ST JOIN STCO JOIN CO
Group By STID, STName, Semester
Having Semester = '95 - 96 - 1'
```

Select STName, AVG(Grade)
From ST JOIN STCO JOIN CO
Group By STID, STName, Semester
Having Semester = '95 - 96 - 1' (2)

```
Select STName, AVG(Grade * Credit)  
From ST JOIN STCO JOIN CO  
Where Semester = '95 - 96 - 1'  
Group By STID, STName
```

```
Select STName, SUM(Grade * Credit) / SUM(Credit)
From ST JOIN STCO JOIN CO
Where Semester = '95 - 96 - 1'
Group By STID, Semester
```

۶۴- با توجه به نمودار ER داده شده، کدام مورد برای پرس و جو «نام دانشجویانی که معدل آنها از میانگین نمایند» دروس دانشگاه بسته است؟ نادرست است؟



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))

(۲)

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

(۳)

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

(۴)

۶۵- رابطه های جدول داده شده را در نظر بگیرید. با اجرای دستور زیر حداقل و حداکثر تعداد سطرهای خروجی کدام است؟

(مهندسی کامپیوتر-دولتی ۹۷)

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

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

۶۶- 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 (Stid, StName,...)	K
Course (Cid, CName,...)	N
StudentCourse (Stid, Cid, grade)	M

0 (۲) K (۱)

(۴) گزینه های ۱ و ۲ می توانند صحیح باشند.

K.N (۳)

۶۷- با توجه به گزاره های 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 $1 - 2^n$ 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 Select distinct α From R است.

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

(۱) دو گزاره نادرست است.

(۴) همه گزاره‌ها نادرست هستند.

(۱) یک گزاره نادرست است.

(۳) سه گزاره نادرست است.

۶۸- با توجه به پایگاه داده زیر، کدام کد SQL داده شده، لیست تمام کتاب‌هایی را نمایش می‌هد که تمام نسخه‌های آنها امانت گرفته شده است؟
(مهندسی کامپیوترا- دولتی ۹۸)

user (UID, name, Contact)

BOOK (BID, Title, Publisher, TotalNumber)

BORROWING (UID, BID, StartDate, EndDate)

وقتی کتابی هنوز در امانت است تاریخ خاتمه NULL است و به محض تحویل پُر می‌شود.

a) with tbl as (

Select BORROWING.BID, COUNT(BORROWING.BID) as cnt

From BORROWING, BOOK

Where BOOK.BID = BORROWING.BID and EndDate is NULL

Group by BORROWING.BID)

Select tb1.BID

From tb1, BOOK

Where tb1.BID = BOOK.BID and BOOK.TotalNumber = cnt

b) Select BID

From BOOK b1

Where (select COUNT (UID)

From BORROWING

Where BORROWING.BID = b1.BID and

BORROWING.EndDate is NULL) = b1.TotalNumber

c) Select BID, TotalNumber

From BORROWING, BOOK

Where BOOK.BID = BORROWING.BID and EndDate is NULL

Group by BORROWING.BID

Having COUNT(BID) = TotalNumber

c , b , a (۴

c , b (۳

c , a (۲

b , a (۱

۶۹- بانک اطلاعاتی ماشین‌ها و تصادفات روبه‌رو را در نظر بگیرید:
(مهندسی کامپیوترا- دولتی ۹۸)

Person (SSN, name, address)

Car (License, year, model)

Accident (License, accident-date, driver, damage-amount)

Owns (SSN, License)

با توجه به بانک اطلاعاتی ماشین‌ها و تصادفات، کدام راننده پرهزینه‌ترین تصادف را داشته است؟
 (نام راننده و میزان خسارت برگردانده شود به زبان SQL)

```
(select driver,damage – amount
from Accident)Except(select a.driver,a.damage – amount
from Accident a,Accident b
where a.damage – amount < b.damage – amount and a.driver <> b.driver)
```

```
select driver, damage – amount
from Accident
where damage – amount in(select MAX(damage – amount)
from Accident)
```

```
select driver, damage – amount
from Accident
where damage – amount = MAX(damage – amount)
```

۴) همه موارد صحیح است.

۷۰- بانک اطلاعاتی ماشین‌ها و تصادفات روبه‌رو را در نظر بگیرید:

Person (SSN , name , address)
 Car (License , year , model)
 Accident (License , accident – date , driver , damage – amount)
 Owns (SSN , License)

با توجه به بانک اطلاعاتی ماشین‌ها و تصادفات، کدام شماره پلاک ماشین در بیش از یک تصادف، درگیر بوده است؟

(پاسخ به زبان SQL و سطرهای تکراری فقط یکبار نشان داده شوند.)

```
select distinct A.License
from Accident A
where A.License in(select B.License
from Accident B
where A.accident – date <> B.accident – date)
```

```
select License
from Accident
group by License
having count(accident – date)>1
```

```
select A.License
from Accident A,Accident B
where A.License = B.License and A.accident – date <> B.accident – date
```

۴) موارد ۱ و ۲ صحیح است.

۷۱- پایگاه داده زیر را در نظر بگیرید:

node (NID , Name , Color , Description)

اطلاعات موجود در جدول node شامل شماره، نام، رنگ و شرح مربوط به هر گره است.

edge (NID1,NID2 , EdgeType)

هر سطر از جدول edge، نشان دهنده وجود یک یال جهت دار از نوع EdgeType از گره با شماره NID1 به گره با شماره NID2 است.

در خصوص پرس و جوهای SQL زیر کدام گزینه درست است؟

پرس و جوی اول	<pre>select distinct NID from node, edge where NID=edge.NID2 AND not exists (select * from edge where edge.NID1=NID)</pre>
پرس و جوی دوم	<pre>select T1.NID from (select count (NID1) as cnt , NID from node left outer join edge on edge.NID1=NID group by NID) T1 , (select count (NID2) as cnt , NID from node left outer join edge on edge.NID2=NID group by NID) T2 where T1.NID = T2.NID and T1.cnt < T2.cnt</pre>

(۱) پرس و جوی اول، شماره گرهایی را می دهد که یال خروجی دارند اما یال ورودی ندارند.

(۲) پرس و جوی دوم، شماره گرهایی را می دهد که درجه ورودی آنها کمتر از درجه خروجی آنها است.

(۳) پرس و جوی دوم، شماره گرهایی را می دهد که درجه خروجی آنها کمتر از درجه ورودی آنها است.

(۴) گزینه های ۱ و ۳ صحیح هستند.

(مهندسی کامپیوترا - دولتی ۹۹)

۷۲- با توجه به جداول داده شده زیر خروجی به چه صورت است؟

شهر = City

رنگ قطعه = Color

شماره قطعه = P#

شماره تولیدکننده = S#

تعداد تولیدات = QTY

تولیدکننده = Supplier

تولید = Supply

قطعه = Part

Supplier(S#, Sname , City)

Supply(S# , P# , QTY)

Part(P# , Color)

SELECT S#

FROM Supplier

WHERE S# = P#;

- (۱) شماره تولیدکنندگان و قطعات را که باهم یکسان هستند در خروجی ظاهر می‌کند.
- (۲) اجرا نمی‌شود، زیرا دو دامنه (Domain) متفاوت باهم مقایسه شده است.
- (۳) تمام تولیدکنندگانی که قطعه‌های یکسان تولید می‌کنند را برابر می‌گرداند.
- (۴) تمام قطعات با شماره یکسان را برابر می‌گرداند.

۷۳- با توجه به پایگاه داده زیر در مورد پرس و جوهای SQL زیر کدام مورد درست است؟
(مهندسی کامپیوتر- دولتی ۹۹)

جدول گره‌ها	Node(NID , Name , Color , Description) اطلاعات موجود در جدول گره‌ها شامل شماره، نام، رنگ و شرح مربوط به هر گره است.
جدول یال‌ها	Edge(NID1,NID2, EdgeType) هر سطر از جدول Edge، نشان دهنده وجود یک یال جهت‌دار از نوع EdgeType از گره با شماره NID1 به گره با شماره NID2 است.

پرس و جو اول	<pre>select distinct NID from node, edge where NID=edge.NID2 AND not exists (select * from edge where edge.NID1=NID)</pre>
پرس و جو دوم	<pre>select T1.NID from (select count (NID1) as cnt , NID from node left outer join edge on edge.NID1=NID group by NID) T1 , (select count (NID2) as cnt , NID from node left outer join edge on edge.NID2=NID group by NID) T2 where T1.NID = T2.NID and T1.cnt < T2.cnt</pre>

- (۱) پرس و جو اول، شماره گره‌هایی را می‌دهد که یال خروجی دارند اما یال ورودی ندارند.
- (۲) پرس و جو دوم، شماره گره‌هایی را می‌دهد که درجه خروجی آنها کمتر از درجه ورودی آنها است.
- (۳) پرس و جو دوم، شماره گره‌هایی را می‌دهد که درجه ورودی آنها کمتر از درجه خروجی آنها است.
- (۴) گزینه‌های ۱ و ۲ صحیح هستند.

۷۴- با توجه به پایگاه داده زیر، پرس و جوی «نام اعضایی 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 (ISBN, Title, Author, Publisher)
اعضا	Member (MID, Name, MDate)
امانت کتاب	Borrow (MID, ISBN, BDate, Duration)

(الف)

```

SELECT Name
FROM (SELECT Member.MID , Member.Name , count (Book.ISBN) as BorrowCount
      FROM Member , Borrow , Book
      WHERE Member.MID = Borrow.MID and Book.ISBN = Borrow.ISBN
            and Publisher = 'Springer'
      GROUP BY Member.MID , Member.Name) as M
WHERE BorrowCount < (SELECT count(*)
                        FROM Book
                        WHERE Publisher = 'Springer')

```

(ب)

```

SELECT Name
FROM Member
WHERE NOT EXISTS ((SELECT ISBN
                     FROM Book
                     WHERE Publisher = 'Springer')
                  EXCEPT
                  (SELECT ISBN
                     FROM Borrow
                     WHERE Borrow.MID = Member.MID))

```

(ج)

```

SELECT Name
FROM Member
WHERE NOT EXISTS (SELECT 1
                   FROM Book
                   WHERE Publisher = 'Springer'
                   And
                   NOT EXISTS (SELECT ISBN
                               FROM Borrow
                               WHERE Borrow.MID = Member.MID
                               And Borrow.ISBN = Book.ISBN))

```

۴ سے

۳ دو

۲ یک

۱) صفر

پاسخ تست‌های فصل ششم: SQL دستورات

۱- گزینه (۲) صحیح است.

برای پاسخ به این سؤال ابتدا لازم است انواع join مورد بحث و بررسی قرار گیرد:
جداول Employee و Department را در نظر بگیرید:

Emp#	EmpName	Salary	Dep#
101	Alavi	2	12
102	Hassani	4	12
103	Hosseini	6	13
104	Sajjadi	8	13
105	Asgari	10	14
106	Akbari	12	14

جدول Employee

Dep#	DepName
12	حسابداری
13	فروش
14	مهندسی

جدول Department

(ضرب دکارتی) CROSS JOIN

مثال اول:

```
select *  
from employee CROSS JOIN department
```

مثال دوم:

```
select *  
from employee, department
```

توجه: در CROSS JOIN همه سطوحی دو جدول در کنار هم قرار می‌گیرند.

Employee				Department	
Emp#	EmpName	Salary	Dep#	Dep#	DepName
:	:	:	:	:	:

(ضرب دکارتی دارای شرط) INNER JOIN

این عملگر در واقع همان ضرب دکارتی است که می‌تواند همراه با شرط مورد نظر برای پیوند جداول مورد استفاده قرار بگیرد.

مثال اول:

```
select *  
from employee INNER JOIN department  
on employee.Dep# = department.Dep#
```

توجه: در INNER JOIN می‌توان از =، >، <، >=، <= و <> استفاده نمود.

مثال دوم: (شبیه‌سازی)

```
select *  
from employee, department  
where employee.Dep# = department.Dep#
```

توجه: مطابق پرس و جوهای فوق، ابتدا دو جدول با یکدیگر ضرب دکارتی می‌شوند و سپس شرط داده شده روی سطرها اعمال می‌گردد. شرط داده شده در واقع همان شرط ضرب است. در خروجی پرس و جوهای فوق ستون مشترک دو بار ظاهر می‌گردد.

Employee				Department	
Emp#	EmpName	Salary	Dep#	Dep#	DepName
101	Alavi	2	12	12	حسابداری
102	Hassani	4	12	12	حسابداری
103	Hosseini	6	13	13	فروش
104	Sajjadi	8	13	13	فروش
105	Asgari	10	14	14	مهندسی
106	Akbari	12	14	14	مهندسی

EQUI JOIN (ضرب دکارتی دارای شرط فقط برابری)

این عملگر در واقع همان ضرب دکارتی است که می‌تواند همراه با شرط فقط برابری برای پیوند جداول مورد استفاده قرار بگیرد.

مثال اول:

```
select *
from employee JOIN department
on employee.Dep# = department.Dep#
```

توجه: در JOIN فقط = مورد استفاده قرار می‌گیرد و استفاده از <، >، =، <=، >= و <> مجاز نیست.

مثال دوم: (شبیه‌سازی)

```
select *
from employee, department
where employee.Dep# = department.Dep#
```

مثال سوم: (شبیه‌سازی)

```
select *
from employee INNER JOIN department USING (Dep#)
```

مثال چهارم: (شبیه‌سازی)

```
select *
from employee INNER JOIN department
on employee.Dep# = department.Dep#
```

توجه: در INNER JOIN می‌توان از =، >، < و <> استفاده نمود.

توجه: مطابق پرس و جوهای فوق، ابتدا دو جدول با یکدیگر ضرب دکارتی می‌شوند و سپس شرط برابری روی سطرها اعمال می‌گردد. شرط برابری در واقع همان شرط ضرب است. در خروجی پرس و جوهای فوق ستون مشترک دو بار ظاهر می‌شود.

Employee				Department	
Emp#	EmpName	Salary	Dep#	Dep#	DepName
101	Alavi	2	12	12	حسابداری
102	Hassani	4	12	12	حسابداری
103	Hosseini	6	13	13	فروش
104	Sajjadi	8	13	13	فروش
105	Asgari	10	14	14	مهندسی
106	Akbari	12	14	14	مهندسی

NATURAL JOIN (الحاق طبیعی)

این عملگر، الحاق طبیعی دو جدول را انجام می‌دهد.

مثال:

```
select *
from employee NATURAL JOIN department
```

توجه: مطابق پرس‌وجوی فوق، فقط سطرهایی از پیوند دو جدول در خروجی ظاهر می‌شوند که مقدارشان در ستون مشترک برابر باشد. (در واقع همان تعریف الحاق طبیعی در جبر رابطه‌ای). در خروجی پرس و جوی فوق ستون مشترک فقط یک بار ظاهر می‌شود.

Employee				Department
Emp#	EmpName	Salary	Dep#	DepName
101	Alavi	2	12	حسابداری
102	Hassani	4	12	حسابداری
103	Hosseini	6	13	فروش
104	Sajjadi	8	13	فروش
105	Asgari	10	14	مهندسی
106	Akbari	12	14	مهندسی

أنواع پرس‌وجوهای تو در تو (nested query)

جداول زیر را در نظر بگیرید:

S#	Sname	City
S1	Sn1	C1
S2	Sn2	C2
S3	Sn3	C2

جدول *s*

S#	P#	QTY
S1	P1	10
S1	P2	20
S2	P1	30

جدول *sp*

الف) Normal Subquery

مثال: نام تولیدکنندگانی که حداقل یک قطعه تولید کرده‌اند.

```

select Sname
from S
where S# in (select S#
              from SP
              where S# in
                    (select S#
                     from S1, S2
                     where S1.S# = S2.S#))
    
```

ویژگی ها**۱- تکرار**

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 سطرهای خود، از مقادیر زیر پرس و جوی داخلی استفاده می‌کند.

۲- وابستگی (outer to inner)

پرس و جوی خارجی، به زیر پرس و جوی داخلی وابستگی دارد.

۳- کارایی (Performance)

کارایی بالاتری را نسبت به Normal Subquery دارد.

۴- مرتبه اجرایی ($O(N+M)$)**ب) Correlated Subquery**

مثال اول: نام تولید کننده‌گانی که حداقل یک قطعه تولید کرده‌اند:

```

select Sname
from S
where exists (select *
              from SP
              where S.S# = SP.S#)
    
```

ویژگی ها**۱- تکرار**

به ازای حرکت در هر یک از سطرهای پرس و جوی خارجی، یک بار به طور کامل از ابتدا تا انتها زیر پرس و جوی داخلی اجرا و براساس 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 هر بار اجرای حلقه خارجی، یک بار به طور کامل حلقه داخلی اجرا می‌گردد.

۲- وابستگی (Inner to Outer)

پرس و جوی داخلی توسط یک ستون به پرس و جوی خارجی متصل و وابسته می‌گردد.

۳- کارایی (Performance)

کارایی پایین‌تری را نسبت به Normal Subquery دارد.

بنابراین گزینه اول درست است.

۴- مرتبه اجرایی $O(M \times N)$

جداول Employee و Department را در نظر بگیرید:

Emp#	EmpName	Salary	Dep#
101	Alavi	2	12
102	Hassani	4	12
103	Hosseini	6	13
104	Sajjadi	8	13
105	Asgari	10	14
106	Akbari	12	14

جدول Employee

Dep#	DepName
12	حسابداری
13	فروش
14	مهندسی

جدول Department

مثال دوم: مشخصات کارمندانی که حقوقشان، از متوسط حقوق، کارمندان بخشنده بیشتر است:

select *

```
from employee emp1
where salary > (select AVG (salary)
from employee emp2
where emp1.Dep# = emp2.Dep#)
```

توجه: زیر پرس‌وجوی فوق، یک Correlated Subquery است که زیر پرس‌وجوی مورد نظر را توسط ستون # Dep به پرس‌وجوی خارجی متصل می‌کند.

توجه: در پرس‌وجوی فوق، به ازای حرکت در هر یک از سطرهای پرس‌وجوی خارجی، یک بار به طور کامل از ابتدا تا انتهای، زیر پرس‌وجوی داخلی اجرا و برآسانش شرطی که زیر پرس‌وجوی داخلی را به پرس‌وجوی خارجی متصل می‌کند، بررسی انجام می‌شود.

توجه: پرس‌وجوی فوق سطر به سطر توسط عملگر where بر روی جدول emp1 (جدول employee) حرکت می‌کند و برای هر سطر از جدول emp1، شرط جلوی where را چک می‌کند، اگر شرط جلوی where برقرار بود، سطر مورد نظر از جدول emp1 انتخاب می‌گردد و این رویه تا به انتهای جدول emp1 ادامه پیدا می‌کند.

فرض کنید در سطر اول جدول emp1 قرار داریم....

Emp#	EmpName	Salary	Dep#
101	Alavi	2	12

با توجه به مقدار # Dep برابر 12 از سطر اول جدول emp1، پرس‌وجوی select داخلی، یعنی

```
select AVG (Salary)
from employee emp2
```

where emp1.Dep# = emp2.Dep#

12

میانگین حقوق کارمندانی را از جدول emp2 محاسبه می‌کند که # dep آنها برابر 12 باشد، به بیان دیگر زیر پرس‌وجوی فوق میانگین حقوق کارمندان بخش شماره 12 را محاسبه می‌کند. دو جدول emp1 و emp2 که

مطابق جدول employee هستند و پرس‌وجوی فوق را تجسم کنید. با توجه به مشخصات کارمندان بخش شماره 12، پُر واضح است که میانگین حقوق کارمندان بخش شماره 12، برابر $\frac{2+4}{2} = \frac{6}{2} = 3$ می‌باشد. بنابراین کارمند شماره 101، برای نمایش در خروجی، انتخاب نمی‌گردد. زیرا:

```
select *
from employee emp1
where salary > (select AVG(salary)
                  from employee emp2
                  where emp1.Dep# = emp2.Dep#)
```

$$\frac{2+4}{2} = \frac{6}{2} = 3$$

مشاهده می‌کنید که شرط فوق برای سطر اول جدول emp1 برقرار نیست. بنابراین سطر اول از جدول حائز شرایط نیست و برای نمایش در خروجی انتخاب نمی‌گردد. این رویه تا انتهای جدول به صورت سطر به سطر، ادامه پیدا می‌کند.

برای مثالی دیگر، سطر آخر جدول emp1 را هم در نظر بگیرید.

با توجه به مشخصات کارمندان بخش شماره 14، پُر واضح است که میانگین حقوق کارمندان بخش شماره 14، برابر $\frac{10+12}{2} = \frac{22}{2} = 11$ می‌باشد. بنابراین کارمند شماره 106 برای نمایش در خروجی، انتخاب می‌گردد. زیرا:

```
select *
from employee emp1
where salary > (select AVG(salary)
                  from employee emp2
                  where emp1.Dep# = emp2.Dep#)
```

$$\frac{10+12}{2} = \frac{22}{2} = 11$$

مشاهده می‌کنید که شرط فوق برای سطر آخر جدول emp1 برقرار است. بنابراین سطر آخر از جدول حائز شرایط است و برای نمایش در خروجی، انتخاب می‌گردد. با توجه به مشخصات کارمندان

بخش 13، پُر واضح است که میانگین حقوق کارمندان بخش شماره 13 برابر $\frac{6+8}{2} = \frac{14}{2} = 7$ است.

بنابراین خروجی پرس‌وجوی مذکور به صورت زیر خواهد بود:

Emp#	EmpName	Salary	Dep#
102	Hassani	4	12
104	Sajjadi	8	13
106	Akbari	12	14

گزینه دوم نادرست است. زیرا، هر Query که قابل بیان به صورت یک Natural Join می‌باشد، همواره این‌طور نیست که به صورت یک Subquery نیز قابل بیان باشد.

مثال: جداول زیر را در نظر بگیرید:

a	b
1	2
3	4
6	7

جدول P

b	c
2	5
2	7
4	8

جدول Q

توسط الحق طبیعی:

select *
from P NATURAL JOIN Q

خروجی \Rightarrow

a	b	c
1	2	5
1	2	7
3	4	8

: subquery توسط

select *
from P
where b in (select b
from Q)
 ^
 خروجی \Rightarrow
 a | b
 1 | 2
 3 | 4
 2,2,4

توجه: زیرا subquery یک زیر پرس‌وجوی داخلی است که مقادیر لازم برای پرس‌وجوی خارجی را فراهم می‌کند، اما در نهایت مشخصات پرس‌وجوی خارجی براساس شرط مورد نظر، راهی خروجی می‌گردد. در حالی که الحق طبیعی، اجتماع ستون‌های دو جدول مورد نظر، در مقادیر برابر در ستون مشترک را راهی خروجی می‌کند.

گزینه سوم درست است. زیرا، هر Query که قابل بیان به صورت یک subquery می‌باشد، می‌تواند به صورت یک Equi Join نیز بیان شود.

مثال: جداول زیر را در نظر بگیرید:

a	b
1	2
3	4
6	7

جدول P

b	c
2	5
2	7
4	8

جدول Q

: subquery توسط

select *
from P
where b in (select b from Q)

خروجی \Rightarrow

a	b
1	2
3	4
3	4

توسط : Equi Join

select Distinct P.a , P.b
from P Join Q
on P.b = Q.b

خروجی
 \Rightarrow

a	b
1	2
3	4

گزینه چهارم درست است. زیرا، هر Query که قابل بیان به صورت یک Equi Join باشد می‌تواند به صورت یک Natural Join نیز بیان شود.
مثال: جداول زیر را در نظر بگیرید.

a	b
1	2
3	4
6	7

جدول P

b	c
2	5
2	7
4	8

جدول Q

توسط : Equi Join

select P.a,P.b , Q.c
from P Join Q
on P.b = Q.b

خروجی
 \Rightarrow

P.a	P.b	Q.b	Q.c
1	2	2	5
1	2	2	7
3	4	4	8

توسط : Natural Join

select *
from P NATURAL JOIN Q

خروجی
 \Rightarrow

a	b	c
1	2	5
1	2	7
3	4	8

۲- گزینه (۲) صحیح است.

گزینه‌های اول، سوم و چهارم معادل هستند و اسامی کارمندانی که مدیر پروژه نیز می‌باشند را استخراج می‌کند، اما گزینه دوم اسامی کارمندانی که فقط مدیر یک پروژه می‌باشند را استخراج می‌کند. گزینه دوم بیانگر این پرس‌وجو است که تنها یک پروژه موجود باشد که شرط p.pmgr=e.empno را تحقق بخشید که جهت پیاده‌سازی پرس‌وجوی مورد نظر مناسب نیست. به عبارت دیگر گزینه دوم نام کارمندانی را می‌دهد که فقط مدیر یک پروژه هستند، در حالی که ممکن است کارمندی مدیریت چند پروژه را بر عهده داشته باشد.

توجه: *satisfies* یک کلمه شبیه کد، و به معنای ارضای شرط مقابله می‌باشد.

۳- گزینه (۲) صحیح است.

همه گزینه‌ها به جز گزینه دوم، تاپل‌هایی از جدول P را بر می‌گرداند که #P آنها با #SP حداقل یکی از تاپل‌های جدول SP که مقدار شارژ آن بیشتر از 40000 است، برابر باشد. به عبارت دیگر اطلاعات قطعاتی که توسط تهیه کننده‌ای با شارژ بزرگتر از 40000 تولید شده است، استخراج می‌گردد.
گزینه دوم خطای نحوی هم دارد که اگر درست شود، با توجه به کلمه All این گزینه از سایر گزینه‌ها

متمايز می‌گردد. عبارت جلوی All باید به شکل زیر اصلاح گردد:

```
select SP.P# from supply SP where SP.charge > 40000
```

همچنین در گزینه دوم $=All$ معنی ندارد و نمادهای $<$, $>$, $=$ و \neq می‌توانست قرار بگیرد. البته در گزینه چهارم نیز استفاده از زیر پرس‌وجو به صورت نادرستی انجام گرفته است. در این گزینه زیر پرس‌وجو به صورت کامل از پرس‌وجو اصلی مستقل است. بنابراین نمی‌توان از شرط $P.P\#=SP.P\#$ در زیر پرس‌وجو استفاده کرد. گزینه چهارم به طور کامل دارای ساختار نادرست و خطای نحوی و غیر قابل اصلاح می‌باشد و معنای درست آن در همان ساختار گزینه سوم قرار دارد. سازمان سنجش آموزش کشور گزینه دوم را به عنوان پاسخ درست اعلام کرده بود.

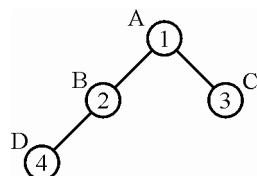
۴- گزینه (۴) صحیح است.

به مثال زیر توجه کنید:

no	name	Father-no
1	A	NULL
2	B	1
3	C	1
4	D	2

جدول person

توجه: درخت روابط پدری و فرزندی جدول فرق به شکل گویاگر به صورت زیر است:



توجه: واضح است که گره‌های 1 و 2 با نام‌های A و B پدر هستند و گره‌های 3 و 4 با نام‌های C و D به عنوان برگ‌ها پدر نیستند.

توجه: قطعه دستور `from person p1 person p2`, دو کپی مجزا از جدول person را ایجاد می‌کند و در هم ضرب دکارتی می‌کند، به صورت زیر:

p1-no	p1-name	p1-father-no	p2-no	p2-name	p2-father-no
1 ①	A	NULL	1	A	NULL
①	A	NULL	2	B	①
1	A	NULL	3	C	①
2	B	1	4	D	2
2	B	1	1	A	NULL
②	B	1	2	B	1
2	B	1	3	C	1
②	B	1	4	D	②

p1-no	p1-name	p1-father-no	p2-no	p2-name	p2-father-no
3	C	1	1	A	NULL
3	C	1	2	B	1
3	C	1	3	C	1
3	C	1	4	D	2
4	D	2	1	A	NULL
4	D	2	2	B	1
4	D	2	3	C	1
4	D	2	4	D	2

گزینه اول: نام تمام افرادی را می‌دهد که شماره شناسایی آن‌ها با شماره شناسایی پدرشان یکی باشد که بی معناست، کد ملی پدر و فرزند نمی‌شود یکسان باشد.

گزینه دوم: نام تمام افراد را استخراج می‌کند. در جدول فوق واضح است.

گزینه سوم: معادل گزینه اول است و نام تمام افرادی را می‌دهد که شماره شناسایی آن‌ها با شماره شناسایی پدرشان یکی باشد که بی معناست.

گزینه چهارم: نام تمام افرادی که شماره آن‌ها، در یکی از سطراها به عنوان شماره پدر درج شده است به عنوان خروجی داده می‌شود. به عبارت دیگر این گزینه نام همه پدرهای موجود در رابطه را می‌دهد. این مطلب در جدول فوق مشخص شده است. گرههای ۱ و ۲ با نام‌های A و B پدر هستند.

توجه: در یک قاعده کلی، یکی از راه حل‌های مقایسه مقادیر یک ستون از یک جدول با مقادیر یک ستون دیگر از جدول دیگر، استفاده از ضرب دکارتی به همراه شرط ضرب می‌باشد. در این مساله هر دو ستون مورد مقایسه در یک جدول بودند که با ایجاد دو کپی مجزا از جدول person و ضرب دکارتی و بیان شرط ضرب، پرس‌وجوی گزینه چهارم ایجاد شده بود.

۵- گزینه (۲) صحیح است.

در صورتی که تابع عددی به شکل مستقیم مقابل دستور where قرار گیرد، پرس‌وجو دارای خطای نحوی خواهد بود. راه حل درست این پرس‌وجو در گزینه اول آمده است. نتیجه این که هرگاه تابع آماری بخواهد مقابل دستور where قرار بگیرد باید داخل یک select مستقل قرار بگیرد تا محاسبات آن به شکل مستقل انجام گردد. بهتر بود در گزینه دوم اشکال منطقی به صورت اشکال نحوی بیان می‌شد.

۶- گزینه (۳) صحیح است.

گزینه اول: اطلاعات سطراهای بی اعتبار در حاصل ضرب دکارتی دو جدول وام و مشتری را می‌دهد (چون فقط سطراهایی را در خروجی می‌دهد که مقدارشان در ستون مشترک مساوی نیست).

گزینه دوم: اطلاعات مشتریانی را می‌دهد که وام گرفته‌اند.

گزینه سوم: داخلی ترین پرانتز نام مشتریانی را می‌دهد که وام گرفته‌اند و پرانتز بیرونی مشخصات کامل مشتریانی را می‌دهد که وام نگرفته‌اند.

۷- گزینه (۴) صحیح است.

ضد نیم اتصال R و S تاپل‌هایی از رابطه R می‌باشد که در $R \bowtie S$ وجود ندارد. انتخاب تاپل‌هایی که در

$R \bowtie S$ وجود دارند یعنی تاپل‌های پیوندپذیر، بنا به تساوی هر دو فیلد A و B می‌باشد، یعنی select*from R,S where R.A=S.A and R.B=S.B نشان دهنده تاپل‌های متناظر در R و S می‌باشد که در $R \bowtie S$ وجود دارند و طبق تعریف موجود در صورت سؤال، ضد نیم اتصال، رکوردهایی از R است که در این مجموعه تاپل‌ها (پیوندپذیر) وجود نداشته باشد. گزینه سوم تاپل‌های آویزان (پیوند ناپذیر) از رابطه R را در خروجی نشان می‌دهد.

هم‌چنین گزینه اول الزام می‌کند که هم مقدار خصیصه A و هم مقدار خصیصه B باید در دو رابطه متفاوت باشد در حالی که متفاوت بودن یکی از آن‌ها برای آویزان بودن تاپل کافی است. در واقع تاپل‌هایی از دو رابطه توسط عملگر ضد نیم اتصال به عنوان خروجی داده می‌شوند که مقدار خصیصه A یا B آن‌ها در هر دو رابطه R و S به صورت همزمان برابر نباشند. به عبارت دیگر تاپل‌هایی باید در خروجی نهايی ظاهر شوند که یا مقدار خصیصه A آن‌ها در دو رابطه متفاوت است یا مقدار خصیصه B آن‌ها در دو رابطه متفاوت است. گزینه‌های دوم و چهارم این شرط را نقض می‌کنند، بنابراین پاسخ سؤال نیستند. دو جدول R و S با مقادیر زیر را در نظر بگیرید:

جدول R			جدول S		
A	B	C	A	B	D
→ پیوندپذیر ①	②	3	①	②	4
→ پیوندناپذیر (تاپل آویزان) ⑤	6	8	⑤	8	9
→ پیوندناپذیر (تاپل آویزان) 2	④	5	7	④	1
→ پیوندناپذیر (تاپل آویزان) 3	7	2	4	5	6

سطرهای پیوندپذیر جدول R به صورت پرس و جوی زیر استخراج می‌شود:

select R.*

from R,S

where R.A=S.A AND R.B=S.B

با توجه به جداول فوق، خروجی نهايی پرس و جوی فوق پس از انجام عملگر ضرب دکارتی و شرط اتصال برابری در دو ستون A و B یعنی سطرهای پیوندپذیر به صورت زیر است:

A	B	C
1	2	3

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

select R.*

from R,S

where R.A<>S.A AND R.B<>S.B

با توجه به جداول فوق، خروجی نهايی پرس و جوی فوق پس از انجام عملگر ضرب دکارتی و شرط اتصال نابرابری در هر دو ستون A و B، به صورت زیر است:

A	B	C
3	7	2

بنابراین پرس و جوی گزینه اول همه تاپل‌های آویزان (پیوندناپذیر) مربوط به جدول R را استخراج نمی‌کند.

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

```
select *
from R
where not exists (select *
                   from S
                   where S.A=R.A OR S.B=R.B)
```

با توجه به جداول فوق، خروجی نهایی پرس و جوی فوق به صورت زیر است:

A	B	C
3	7	2

در واقع در پرس و جوی فوق، فقط به ازای تاپل (3,7,2) از جدول R، عبارت جلوی not exists تهی می‌گردد که منجر به شدن TRUE در پرس و جوی خارجی و به تبع انتخاب تاپل از جدول R می‌گردد.

بنابراین پرس و جوی گزینه دوم نیز همه تاپل‌های آویزان (پیوندانپذیر) مربوط به جدول R را استخراج نمی‌کند.

مطابق پرس و جوی مطرح شده در گزینه سوم، داریم:

```
select *
from R
where not exists (select *
                   from S
                   where S.A=R.A AND S.B=R.B)
```

با توجه به جداول فوق، خروجی نهایی پرس و جوی فوق به صورت زیر است:

A	B	C
5	6	8
2	4	5
3	7	2

در واقع در پرس و جوی فوق، فقط به ازای تاپل‌های (5,6,8)، (2,4,5) و (3,7,2) از جدول R، عبارت جلوی not exists تهی می‌گردد که منجر به شدن TRUE در پرس و جوی خارجی و به تبع انتخاب تاپل از جدول R می‌گردد.

بنابراین پرس و جوی گزینه سوم همه تاپل‌های آویزان (پیوندانپذیر) مربوط به جدول R را استخراج می‌کند، که پاسخ سؤال نیز همین گزینه سوم است.

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

```
select *
from R
where A<>any(select A AND B<>any(select B
                                             from S)
                                             from S)
```

با توجه به جداول فوق، خروجی نهایی پرس و جوی فوق به صورت زیر است:

A	B	C
3	7	2

بنابراین پرس و جوی گزینه چهارم نیز همه تاپل‌های آویزان (پیوندنایپذیر) مربوط به جدول R را استخراج نمی‌کند. اگر در گزینه چهارم به جای OR استفاده می‌شد، آنگاه گزینه چهارم نیز می‌توانست پاسخ سوال باشد.

۸- گزینه (۱) صحیح است.

SELECT ستون‌ها FROM جداول [شرط] [ORDER BY] [GROUP BY] [HAVING]

ساختاری نحوی دستور select به صورت زیر است:

در گزینه اول به دلیل استفاده از دستور DESC خروجی به صورت نزولی نشان داده خواهد شد، اما در سایر گزینه‌ها خطای نحوی وجود دارد.

۹- گزینه (۴) صحیح است.

ابتدا با دستور Group by J#، اطلاعات جدول براساس #Dسته‌بندی می‌شود، سپس با استفاده از شرط Having و تابع count و دستور Distinct عمل شمارش #S در داخل هر پروژه انجام می‌شود، اگر این مقدار برابر یک بود، به این معنی است که یک تهیه‌کننده در آن شماره پروژه، به تنهایی تمام قطعات مورد نیاز پروژه را تهیه کرده است.

به مثال زیر توجه کنید:

S#	P #	J #	
SI	P1	J1	select J#
SI	P2	J1	from SPJ
SI	P3	J1	Group by J#
SI	P1	J2	Having count(Distinct S#)=1
S2	P2	J2	

از آن‌جا که هدف، شمارش تهیه‌کننده‌ها به تفکیک پروژه‌ها است، باید گروه‌بندی براساس شماره پروژه انجام شود، از طرفی استفاده از دستور Distinct باعث می‌شود که یک تهیه‌کننده فقط یک بار با تابع count شمرده شود.

پرس‌وجوی فوق گروه‌های زیر را ایجاد می‌کند:

J1		J2		
SI	P1	SI	P1	در سرگروه در
SI	P2	S2	P2	خروچی نمایش داده می‌شود.
SI	P3			J#

در گروه J1 شرط Count(Distinct S#)=1 برقرار است. پس این گروه توسط Having انتخاب می‌شود.

J#
J1

همان‌طور که واضح است، فقط در پروژه J1 است که همه قطعات آن را یک تهیه‌کننده به نام SI تهیه می‌کند، اما در پروژه J2، دو تهیه‌کننده S1 و S2 به ترتیب قطعات P1 و P2 را تهیه می‌کنند. که

count(Distinct S#) در آن برابر مقدار 2 می‌باشد.

۱۰- گزینه (۱) صحیح است.

پرس‌وجوی داده شده، به دلیل استفاده از تابع عددی در محل نادرست، خطای نحوی دارد و توسط کامپایلر قابل اجرا نیست. متأسفانه این پاسخ در بین گزینه‌های داده شده وجود ندارد.

۱۱- گزینه (۲) صحیح است.

طبق قانون اول جامعیت (جامعیت درون رابطه‌ای) هر رابطه (جدول) باید به تنها یکی درست باشد، یعنی صفات آن چند مقداری و مرکب نباشند و حداقل یک کلید کاندید داشته باشد. بنابراین گزینه سوم درست است.

مطابق قانون دوم جامعیت (جامعیت موجودیت) مقادیر همه مؤلفه‌های تاپل‌ها در یک رابطه باید در دامنه تعریف شده برای آن مؤلفه قرار داشته باشند و هم‌چنین کلید اصلی یا بخش‌های مختلف آن باید مقدار NULL داشته باشند. بنابراین گزینه اول درست است.

قانون سوم جامعیت (جامعیت ارجاع) به مسئله کلید خارجی مربوط می‌شود. بنابراین گزینه چهارم نیز به درستی چگونگی برقراری صحت ارجاع در SQL را بیان می‌کند. اما مسئله not NULL بودن، اول این که به قانون دوم جامعیت مربوط می‌شود و دوم این که وقتی یک صفت به عنوان کلید اصلی تعریف می‌شود به صورت پیش تعریف از دو خاصیت not NULL و شاخص unique برخوردار است و لازم نیست این دو خاصیت به صورت صریح برای آن ذکر شود.

۱۲- گزینه (۳) صحیح است.

گزینه اول نادرست است: زیرا تابع عددی avg در محل نادرست استفاده شده است، بنابراین این پرس‌وجو خطای نحوی دارد و از سوی کامپایلر اجرا نمی‌گردد. تابع عددی جلوی where حتماً باید داخل select باشد.

گزینه دوم نادرست است: زیرا بدون Group by از having استفاده شده است، بنابراین این پرس‌وجو خطای نحوی دارد و از سوی کامپایلر اجرا نمی‌گردد.

گزینه چهارم نادرست است: زیرا شرط سازگاری صفات در جلوی Group by و select رعایت نشده است، بنابراین این پرس‌وجو خطای نحوی دارد و از سوی کامپایلر اجرا نمی‌گردد.

گزینه سوم درست است.

جدول زیر را در نظر بگیرید:

empno	empname	dep	salary
e1	en1	amoozesh	10
e2	en2	amoozesh	20
e3	en3	amoozesh	30
e4	en4	maly	40
e5	en5	maly	50
e6	en6	maly	60
e7	en7	honari	70
e8	en8	honari	80

جدول employee

پرس و جو: مشخصات کارکنانی که حقوقشان از متوسط حقوق کلیه کارکنان سازمان بیشتر است را لیست نمایید. (مطابق گزینه سوم)

```
select*
from employee
        45
where salary > (select avg(salary) from employee)
```

پس مقدار میانگین حقوق کلیه کارمندان، برابر 45 می باشد.

بنابراین خروجی پرس و جوی فوق به صورت زیر خواهد بود:

empno	empname	dep	salary
e5	en5	maly	50
e6	en6	maly	60
e7	en7	honari	70
e8	en8	honari	80

در واقع در گزینه سوم ابتدا در پرانتز داخلی میانگین حقوق کلیه کارمندان سازمان محاسبه می شود. سپس در پرس و جوی بیرونی مشخصات کامل کارکنانی که حقوقشان از متوسط حقوق کلیه کارمندان سازمان بیشتر است به عنوان خروجی نهایی تولید می شود.

۱۳- گزینه (۱) صحیح است.

براساس پرس و جوی مطرح شده در صورت سؤال، تمامی قطعات قرمز رنگ، به رنگ نارنجی توسط دستور update به روزرسانی می گردند.

۱۴- گزینه (۴) صحیح است.

پرس و جوی مطرح شده، نام و شهر تولیدکنندگانی را استخراج می کند که تاکنون قطعه P1 را تولید کرده اند. در گزینه های اول، دوم و سوم نحوه بیان این پرس و جو به شیوه های متفاوت، اما با خروجی یکسان بیان شده است.

۱۵- گزینه (۲) صحیح است.

A	B	C	F	B	A	B	C	F
a1	b1	MK	f1	b1	a1	b1	MK	f1
a2	b4	MTK	f2	b2	a2	b1	RP	f2
a1	b2	LN	Right Outer Join		a1	b2	LN	f2
a2	b1	RP	جدول x					

جدول y

توجه کنید که پس از پیوند طبیعی، تاپلی از جدول x (جدول سمت راست) باقی نمی ماند که بخواهد اضافه شود (تاپل آویزان در جدول سمت راست وجود ندارد). بنابراین با توجه به جداول مذکور، حاصل الحق خارجی راست، برابر الحق طبیعی است.

پس از اعمال شرط زیر	\rightarrow	A B C F
where $y.c \text{ like } 'M\%'$		a1 b1 MK f1

۱۶- گزینه (۳) صحیح است.

پرس و جوی مطرح شده در صورت سؤال خطای نحوی دارد. در زیر پرس و جوهای متصل به خارج، باید نوع اتصال، مابین زیر پرس و جوی داخلی و پرس و جوی خارجی مشخص گردد. شکل اصلاح شده پرس و جوی مطرح شده در صورت سؤال به صورت زیر است:

```
select Distinct sname
from S
where not exists (select *
                   from G
                   where G.S# = S.S# AND L#= 'L1')
```

جداول زیر را در نظر بگیرید:

S#	Sname	S#	L#	grade	L#	Lname
S1	S _{n1}	S1	L1	10	L1	L _{n1}
S2	S _{n2}	S1	L2	12	L2	L _{n2}
S3	S _{n3}	S2	L1	14		
S4	S _{n5}	S3	L2	16		
S5	S _{n5}	S4	L2	18		
S6	S _{n6}	S5	L2	20		

جدول S جدول G جدول L

در پرس و جوی اصلاح شده صورت سؤال، به ازای حرکت در هر سطر از جدول S، سطرهایی از جدول G که شماره دانشجویی آنها با شماره دانشجویی جدول S برابر باشد، بررسی می‌گردد که به ازای شماره دانشجویی مورد نظر در جدول S هیچ نمره‌ای از درس L₁ به ازای شماره دانشجویی مورد نظر در جدول G وجود نداشته باشد. اگر هیچ سطری به ازای شماره دانشجویی موجود در جدول S، در جدول G با توجه به شرط مطرح شده موجود نباشد، سطر مورد نظر از جدول S انتخاب می‌گردد. و در نهایت نام دانشجویانی در خروجی قرار می‌گیرد که از درس L₁ نمره‌ای ندارند.

در پرس و جوی فوق، به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول G بررسی می‌گردد، تا مشخص گردد هیچ سطری در جدول G قرار ندارد که شرط موجود در select داخلي را برقرار کند، اگر هیچ سطری وجود نداشته باشد، سطر مورد نظر از جدول S، در خروجی نمایش داده می‌شود.

پرس و جوی فوق در واقع نام دانشجویانی را نمایش می‌دهد که برای آنها هیچ نمره‌ای از درس L₁ در جدول G ثبت نشده است. البته چون به دلیل کلید نبودن Sname، ممکن است اسمی برخی دانشجویان شبیه هم باشد، امکان دارد، اسمی تکراری در خروجی این پرس و جو مشاهده شود. بنابراین خروجی Sname با توجه به استفاده از دستور Distinct در پرس و جوی مطرح شده، به صورت زیر خواهد بود:

Sname
S _{n3}
S _{n5}

توجه: SQL به طور پیش فرض سطرهای تکراری را حذف نمی کند، مگر از دستور Distinct استفاده شود.
گزینه دوم نادرست است. زیرا، در SQL مفاهیمی همچون اولین، آخرین و یکی از، معنا و مفهومی ندارد.
در واقع اگر شرط انتخاب سطر، برای سطری برقرار باشد، سطر مورد نظر در خروجی قرار می گیرد.
توجه: اگر به جای دستور not exists از دستور exists در پرس و جوی اصلاح شده صورت سوال، استفاده
گردد، آنگاه پرس و جوی مطرح شده در گزینه اول ایجاد می گردد.

۱۷- گزینه (۱) صحیح است.

پرانتز داخلی یعنی (select MAX (grade) from G) ماکریم نمره ثبت شده را استخراج می کند. پرانتز
بیرونی تر شماره تمام دانشجویانی را می دهد که نمره ماکریم کسب کرده اند.
گزینه سوم بیانگر شماره دانشجویانی است که تعداد نمره بیشتری دریافت کرده اند که با پرس و جوی مطرح
شده مطابقت ندارد. وقت کنید که گزینه دوم با دستورات SQL قابل تولید نیست. زیرا SQL با مفاهیمی
همچون اولین، یکی از، بعضی و آخرین، بیگانه است. اگر شرط انتخاب سطر در دستور select برقرار باشد،
همه سطرهای حائز شرط، در خروجی ظاهر خواهند شد.

۱۸- گزینه (۳) صحیح است.

گزینه اول نادرست است. زیرا، پس از استفاده از دستور Group By نمی توان از Subquery استفاده کرد.
اصولاً Group By در پرس و جوهای تو در تو کاربرد ندارد.
گزینه دوم نادرست است. زیرا، در دستور Group By شرط با Having بیان می شود و نمی توان از where استفاده نمود چون خطای نحوی است.
گزینه چهارم نادرست است. زیرا، Count(*) در جلوی دستور select باید تنها باشد. همچنین Count(*)
نمی تواند جایگزین Count(state) شود زیرا Count(state) مواردی را که مقدار state آنها NULL است را نیز
موردنده شمارش قرار می دهد در حالی که Count (state) مواردی را که مقدار state آنها NULL است،
نمی شمارد.
گزینه سوم درست است. زیرا، امکان ترکیب شرط <Count(state) > 10 با شرط فعلی بدون هیچ محدودیتی
امکان پذیر است.

۱۹- گزینه (۳) صحیح است.

با اجرای دستور مطرح شده در صورت سوال، و با توجه به وجود جمله S(S#, Sname, city) در دستور،
این دستور بدون خطا اجرا می شود و مقدار NULL در ستون status قرار می گیرد. البته به شرطی که ستون
status در زمان تعریف ساختار جدول S، به صورت not NULL تعریف نشده باشد. توجه کنید که اگر
ساختار دستور به صورت مقابل می بود:
Insert into S values ('S10', Smith 'New York')
دستور فوق به دلیل نقض شروط سازگاری از نظر تعداد و نوع با خطا مواجه می گردید. و اجرای دستور از
سوی DBMS لغو می گردید، زیرا جدول تولید کننده دارای ستون چهارمی به نام status است که در دستور،
مقداری برای آن در نظر نگرفته شده است.

۲۰- گزینه (۳) صحیح است.

دستور SQL داده شده جداول S و SP را به هم پیوند می زند. یک قطعه در جدول P ممکن است توسط
هیچ، یک، دو یا چند تولید کننده ساخته شده باشد. با توجه به خط 'where s.city= 'London' اگر قطعه ای

حداقل توسط یک تولید کننده لندن ساخته شده باشد، کلیه مشخصات آن نشان داده می‌شود. برای ایجاد گزینه دوم باید تمامی قطعات تولید شده توسط هر یک از تولید کنندگان لندنی را استخراج نمود، سپس اشتراک گرفت، حاصل، کلیه مشخصات قطعاتی خواهد بود که توسط تمام تولید کنندگان لندن تولید می‌شود.

گزینه چهارم می‌گوید، کلیه مشخصات قطعاتی که دقیقاً توسط یکی از تولید کنندگان لندن تولید می‌شود، فرض کنید S1 و S2 و S3 تهیه کننده لندنی هستند، اما قطعه P2 فقط توسط S2 تولید شده است، بنابراین P2 در خروجی می‌آید. اما قطعه P1 در خروجی نمی‌آید، زیرا هم S1 آن را تولید کرده است و هم S2

S#	Sname	City	S#	P#	QTY	P#	Pname	Color
S1	Sn1	London	S1	P1	10	P1	Pn1	Red
S2	Sn2	London	S2	P1	20	P2	Pn2	Blue
S3	Sn3	London	S2	P2	30	P3	Pn3	Red
S4	Sn4	London						

جدول s

جدول sp

جدول p

با توجه به جدول S، تهیه کننده S3، لندنی هست اما با توجه به جدول SP، قطعه‌ای تولید نکرده است، بنابراین تهیه کننده S3 تولید کننده هیچ قطعه‌ای نیست.
خروجی گزینه سوم به صورت زیر است:

P#	Pname	Color
P1	Pn1	Red
P2	Pn2	Blue

خروجی گزینه دوم به صورت زیر است:

P#	Pname	Color
P1	Pn1	Red

خروجی گزینه چهارم به صورت زیر است:

P#	Pname	Color
P2	Pn2	Blue

پرس‌وجوهی گزینه سوم را مجدداً در نظر بگیرید:

«کلیه مشخصات قطعاتی که حداقل توسط یکی از تولید کنندگان لندن تولید می‌شود.»

حال سراغ جدول p برویم، سطر اول را انتخاب کنید، که a1 p1 مشاهده می‌شود، حال این a1 را ببرید داخل جدول sp، ببینید #s آن کیست، یعنی چه کسی p1 را در جدول sp تولید کرده است، می‌بینید که s1 و s2 تولید کردۀ‌اند، حال s1 یا s2 را ببرید داخل جدول s ببینید که این s1 یا s2 تهیه کننده لندنی هستند، می‌بینید که هستند، پس سطر اول از جدول p یعنی (p1, pn1, Red) در خروجی نمایش داده می‌شود، زیرا نه تنها که حداقل توسط یکی از تولید کنندگان لندنی تولید شده است که بلکه توسط دو تولید کننده لندنی یعنی s1 و s2 تولید شده است، سطر دوم از جدول p هم، یعنی (p2, pn2, Blue) در خروجی نمایش داده می‌شود، مانند روای سطر قبل، خروجی را دنبال کنید، اما سطر سوم یعنی (p3, pn3, Red) در خروجی نمایش داده نمی‌شود، چون p3 را که ببریم داخل جدول p، اصلاً p3 ای وجود ندارد که روای را دنبال کنیم. در یک قاعده کلی، یکی از راه حل‌های مقایسه سطرهای یک جدول با خودش یا دو جدول با هم یا چند جدول با هم، استفاده از ضرب دکارتی و بیان نوع اتصال و شرط ضرب می‌باشد. به این مطلب کمی فکر کنید، ساده

است، برای مقایسه یک سطر از یک جدول مبدأ با یک سطر از یک جدول مقصد، دو جدول ابتدا باید ضرب دکارتی شوند. تا همه‌ی سطراها کنار هم قرار گیرند، سپس شرط ضرب اعمال گردد. مثال واقعی مانند زمانی است که استادی (جدول مبدأ) سؤال می‌پرسد و دانشجویی (جدول مقصد) پاسخ را می‌دهد که اگر حداقل یک دانشجو پاسخ دهد، کافی است. در واقع سؤال در جدول مبدأ قرار دارد و پاسخ در جدول مقصد.

مثال: نام تولید کنندگانی که حداقل یک قطعه تولید می‌کنند.

کدام تولید کننده در جدول مبدأ قرار دارد (سؤال) و قطعه در جدول مقصد قرار دارد (پاسخ)

حل: راهکار در استفاده از ضرب دکارتی است، تا سطراها دو جدول کنار هم قرار بگیرند. شرط ضرب هم $s.s\# = sp.s\#$ است، چون وقتی که s1 از جدول مبدأ به عنوان سؤال مطرح می‌شود، در جدول مقصد فقط برای s1 باید دنبال پاسخ بود! s1 را از جدول s (مبدأ) انتخاب کنید، ببرید داخل جدول sp (مقصد) ببینید حداقل یک قطعه را تولید می‌کند، بله p1 را تولید می‌کند، حال s2 را از جدول s انتخاب کنید، ببرید داخل جدول sp، ببینید حداقل یک قطعه را تولید می‌کند، بله، p1 و p2 را تولید می‌کند، حال s3 و s4 را از جدول s به ترتیب، ببرید داخل جدول sp نه، قطعه‌ای تولید نمی‌کنند، این حرف‌ها که گفتیم به زبان SQL به صورت زیر نوشته می‌شود:

```
select    sname      sname
from      s , sp     → sn1
where    s.s# = sp.s#   sn2
```

۲۱- گزینه (۲) صحیح است.

در دو جدول loan و borrower ستون loan-number به عنوان ستون مشترک تلقی می‌شود. چنان‌که ملاحظه می‌شود، کلیه سطراها پیوندپذیر حاصل از مقدار L-170 و K-230 در ستون مشترک و هم‌چنین سطراها پیوندپذیر چپ حاصل از مقدار L-260 و سطراها پیوندپذیر راست حاصل از مقدار L-155 در هر دو جدول در خروجی نمایش داده شده است، بنابراین، خروجی داده شده با خروجی الحاق خارجی کامل تطبیق دارد.

۲۲- گزینه (۲) صحیح است.

عبارت جبر رابطه‌ای داده شده نام دانشجویان دانشکده کامپیوتر را به عنوان خروجی می‌دهد. گزینه اول نادرست است، زیرا نوع اتصال بیان نشده است. درست آن براساس جبر رابطه‌ای مذکور به صورت زیر است:

```
select sname
from stud , clg
where stud.clg# = clg.clg# AND clg.clgname = 'computer'
```

گزینه دوم درست است و معادل پرس‌وجوی اصلاح شده فوق و جبر رابطه‌ای مطرح شده است. گزینه سوم نادرست است، زیرا در جلوی دستور where مربوط به جدول stud استون clgname برای شرط استفاده شده است، که اصلاً جدول stud، ستون clgname ندارد، ستون clgname مربوط به جدول clg است.

گزینه چهارم نادرست است، زیرا خطای نحوی دارد، ستون جلوی by با ستون جلوی select باید یکسان باشد، مگر ستون جلوی select با توابع آماری همراه گردد.

۲۳- گزینه (۱) صحیح است.

گزینه‌های دوم تا چهارم نام کارمندانی را می‌دهند که در واحد فروش کار می‌کنند. در گزینه‌های دوم و سوم دو دستور any و in مانند یک‌دیگر عمل می‌کنند. هم‌چنین در گزینه چهارم با استفاده از ضرب دکارتی و شرط ضرب دقیقاً همان خروجی تولید می‌شود. اما گزینه اول درست معکوس خروجی سه گزینه دیگر را می‌دهد، نام کارمندانی را می‌دهد که در واحد فروش کار نمی‌کنند. اگر در گزینه اول به جای not exists از exists استفاده می‌شود، آن‌گاه گزینه اول هم معادل سایر گزینه‌ها می‌شود.

۲۴- گزینه (۲) صحیح است.

از آنجا که رکورد دانشجویان قرار است، در آخر ترم تغییر کند و نمره دانشجویان وارد شود، دستور update باید اجرا شود. وقت که در فرض مطرح شده در صورت سؤال، وارد کردن نمرات دانشجویان، در سطر جدید در جدول enroll تلقی نمی‌شود، بلکه وارد کردن نمرات به منزله بروز رسانی ستون score در سطرهای مربوط به درسی است که می‌خواهیم نمرات آن را ثبت کنیم. بر این اساس از دستور update برای این کار استفاده می‌شود. دستور Insert یک رکورد کامل جدید به جدول اضافه می‌کند، در حالی که ما می‌خواهیم ستون score را اصلاح کنیم، لذا از دستور update استفاده می‌کنیم.

۲۵- گزینه (۳) صحیح است.

جداول زیر را داریم:

کلید خارجی				کلید خارجی				کلید اصلی			
A	B	C	FK	D	E	W	FK	FK	X	Y	Z
1	-	-	10	1	2	-	10	10	-	-	-
جدول T1				جدول T2				جدول T3			

create table T1

(A char (5)

B char (5)

C char (5)

FK char (5)

primary key (A)

foreign key (FK) References T3 (FK)

on update option $\begin{cases} \text{no action (restrict)} \\ \text{cascade } \checkmark \\ \text{set null} \end{cases}$)

از آنجا که مطابق فرض مسئله، گزینه on update cascade گزینه T2 ذکر نشده است، بنابراین، برای جدول T2 گزینه no action (به معنی صرف‌نظر از عملیات در صورتی که بر اثر عملیات حذف، درج و بروزرسانی، قوانین جامعیت نقض گردد) در نظر گرفته می‌شود. در واقع no action گزینه پیش فرض است و هیچ عملی انجام نمی‌شود.

از آنجا که ستون FK در جدول T3 کلید اصلی و مورد مراجعه سطرهای دو جدول T2 و T1 است، امکان

تغییر دادن بی قید و شرط مقدار FK در جدول T3 وجود ندارد.

فرض اول: اگر مقدار 10 در ستون FK در جدول T3 باشد و مقدار 10 در ستون FK در دو جدول T2 و T1 نباشد. در این صورت مقدار 10 در ستون FK به مقدار 15 به روز رسانی می شود. در واقع فقط جدول T3 به روز رسانی می گردد.

فرض دوم: اگر مقدار 10 در ستون FK در جدول T3 باشد و مقدار 10 در ستون FK در جدول T1 هم باشد و مقدار 10 در ستون FK در جدول T2 نباشد. در این صورت مقدار 10 در ستون FK در دو جدول T1 و T3 به روز رسانی می شود در واقع فقط جدول T3 و T1 به روز رسانی می گردد.

فرض سوم: اگر مقدار 10 در ستون FK در جدول T3 باشد و مقدار 10 در ستون FK در دو جدول T2 و T1 هم باشد. در این صورت از آنجا که تغییر مقدار 10 در ستون FK در جدول T3 به مقدار 15 سبب ارجاع NULL در جدول T2 می گردد، عملیات به روز رسانی برای هر سه جدول لغو می گردد، هر چند جدول T1، دارای on update cascade می باشد. اما از آنجا که برای جدول T2 گزینه on update no action فعال است، اگر تغییر در جدول T3 سبب ارجاع NULL در جدول T2 گردد، عملیات به روز رسانی در جدول T3 لغو می گردد. در واقع مقدار FK در هیچ یک از جداول به روز نمی شود.

۲۶- گزینه (۱) صحیح است.

عبارت جبری رابطه‌ای داده شده نام اساتیدی را به عنوان خروجی می دهد که در دپارتمان کامپیوتر مشغول کار هستند.

گزینه اول درست است، زیرا در پرانتز داخلی شماره دپارتمان کامپیوتر شناسایی می شود و در پرانتز بیرونی تر نام اساتیدی که شماره دپارتمان آن ها همان شماره دپارتمان کامپیوتر است به عنوان خروجی نهایی تولید می شود، به عبارت دیگر گزینه اول همانند عبارت جبر رابطه‌ای داده شده نام اساتیدی را به عنوان خروجی می دهد که در دپارتمان کامپیوتر مشغول کار هستند.

گزینه دوم نادرست است، زیرا نوع اتصال ذکر نشده است و بنابراین خروجی خواسته شده را نمی دهد.

شكل صحیح گزینه دوم مطابق با جبر رابطه‌ای ذکر شده به صورت زیر است:

```
select pname
from prof, dep
where prof.depnum=dep.depnum and demname= 'computer'
```

گزینه سوم نادرست است: زیرا شرط ضرب آمده است اما عملیات ضرب انجام نشده است، زیرا نام جدول dep در مقابل from نیست، پس این گزینه هم خروجی خواسته شده را نمی دهد.

شكل صحیح گزینه سوم مطابق با جبر رابطه‌ای ذکر شده به صورت زیر است:

```
select pname
from prof, dep
where prof.depnum=dep.depnum
and depname in (select depname from dep
where depname= 'computer')
```

گزینه چهارم نادرست است، زیرا چون از group by استفاده شده است باید ستون مقابله select همان ستونی باشد که براساس آن گروه‌بندی انجام شده، اما در این گزینه گروه‌بندی براساس depnum صورت گرفته است، در حالی که در مقابل select نام ستون pname آمده است. به همین دلیل این گزینه هم پاسخ

مورد نظر را به دست نمی‌دهد، در واقع خطای نحوی دارد.

- گزینه (۳) صحیح است.

در یک قاعده، شرط R هنگامی درست می‌شود که R تهی نباشد. R هنگامی تهی نیست که حداقل یک سطر داشته باشد.

- گزینه (۲) صحیح است.

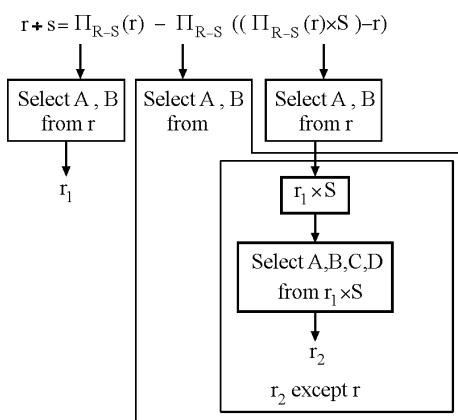
در یک قاعده کلی عملگر تقسیم به صورت $r \div s = \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times s) - r)$ پیاده‌سازی می‌شود. توجه کنید که شرط زیر مجموعه بودن مقسوم‌علیه به مقسوم هم برقرار است، با توجه به ستون‌های جداول r و s داریم:

$$R = \{A, B, C, D\} \quad S = \{C, D\}$$

$$R - S = \{A, B\}$$

بنابراین، جمله select A, B from r Except S (select A, B from r Except S) باید در ابتدای پرس و جو باشد، پس گزینه‌های اول و سوم توانایی پیاده‌سازی عملگر تقسیم را ندارند، لذا این گزینه‌ها را کنار می‌گذاریم. در گزینه چهارم از عملگر اشتراک استفاده شده است، در حالی که عملگر intersect در تعریف عملگر تقسیم کاربرد ندارد.

بنابراین گزینه چهارم را هم کنار می‌گذاریم. اما گزینه دوم پیاده‌سازی درستی از عملگر تقسیم را ارائه می‌کند. اطلاعات زیر گویای این مطلب است:



- گزینه (۴) صحیح است.

اگر فرض کنیم جداول crs و sec و clg به صورت زیر باشد:

crs(c#, cname, unit, clg#)

(شماره دانشکده ارائه دهنده، تعداد واحد، نام درس، شماره درس) درس

sec(sec#, c#, s#, term, pname, score)

(نمره، نام استاد، ترم، شماره دانشجویی، شماره درس، شماره گروه) گروه درس

clg(clg#, clgname, city, pname)

(نام رئیس، نام شهر، نام دانشکده، شماره دانشکده) دانشکده

توجه: از جدول crs برای پیوند جدول‌ها استفاده می‌شود.

توجه: اطلاعات ترم، درس و نام دانشکده (جستجو براساس دانشکده کامپیوتر) در سه جدول sec و clg قرار دارند که تنها گزینه‌های دوم و چهارم از اتصال این سه جدول استفاده کرده‌اند.
جداول زیر را در نظر بگیرید:

clg#	clgname	city	pname	c#	cname	unit	clg#	sec#	c#	s#	term	pname	score
clg1	کامپیوتر	تهران	احمدی	c1	ساختمان داده	3	clg1	sec1	c1	s1	881	علوی	20
clg2	برق	شهراز	اکبری	c2	مدادار منطقی	3	clg1	sec2	c2	s2	881	اصغری	18
				c3	مدادار الکترونیکی	3	clg2	sec3	c2	s3	881	جعفری	17
								sec4	c2	s4	882	حسنی	19
								sec5	c3	s5	883	سجادی	15

جدول دانشکده (clg)

جدول درس (crs)

جدول گروه درس (sec)

پرس‌وجوی مطرح شده در سؤال را در نظر بگیرید:

«ترم‌هایی که دانشکده کامپیوتر درس ارائه کرده است».

حال سراغ جدول clg بروید، سط्रی را انتخاب کنید که نام دانشکده آن برابر کامپیوتر باشد ("کامپیوتر"=clgname)، که با توجه به شرط مطرح شده، سطر اول انتخاب می‌گردد که # clg# آن برابر 1 می‌باشد. حال این clg1 را ببرید داخل جدول crs، ببینید چه درس‌هایی در دانشکده clg1 ارائه شده است. می‌بینید که شماره درس‌های c1 و c2 ارائه شده است. حال این شماره درس‌های c1 و c2 را ببرید داخل جدول sec، ببینید درس‌های c1 و c2 در چه ترم‌هایی ارائه شده است. می‌بینید که در ترم‌های 881 و 882 ارائه شده است. بنابراین 881 و 882 در خروجی نمایش داده می‌شود. در یک قاعده کلی، یکی از راه حل‌های مقایسه سطرهای یک جدول با خودش یا دو جدول با هم یا چند جدول با هم، استفاده از ضرب دکارتی و بیان نوع اتصال و شرط ضرب می‌باشد. این حرف‌ها که گفتیم به زبان SQL به صورت زیر نوشته می‌شود:

```
select distinct term
from sec, clg, crs
where clg.clg # = crs.clg # and crs.c # = sec.c # and clgname='کامپیوتر'
```

گزینه اول نادرست است، زیرا اول این که از جدول crs استفاده نشده است، و دوم این که، پرس‌وجوی مطرح شده در گزینه اول، شماره ترم‌هایی که اساتیدی هم نام با رئیس دانشکده کامپیوتر دارند را ارائه می‌دهد.

گزینه دوم نادرست است، زیرا شرط مطرح شده در بخش having نمی‌تواند پاسخ مورد نظر را تولید کند، فرم اصلاح شده‌ی گزینه دوم برای برآورده کردن پرس‌وجوی مطرح شده در سؤال به صورت زیر است:

```
select term
from sec,clg,crs
where clg.clg # =crs.clg # and crs.c # =sec.c # and clgname='کامپیوتر'
```

توجه: به خاطر بسپارید که یکی از راه‌کارهای شبیه‌سازی دستور distinct، استفاده از دستور group by term می‌باشد.

گزینه سوم نادرست است، زیرا having باید با group by همراه باشد، بنابراین خطای نحوی دارد و از سوی

کامپایلر اجرا نمی‌گردد.

۳۰- گزینه (۲) صحیح است.

در دستور داده شده تابع عددی AVG در محل نادرست مورد استفاده قرار گرفته است، توابع عددی در جلوی where باید داخل select قرار بگیرند. بنابراین پرس‌وجو دارای خطای نحوی است و از سوی کامپایلر اجرا نمی‌گردد.

جدول زیر را در نظر بگیرید:

std #	sname	avg
s1	sn1	12
s2	sn2	14
s3	sn3	16
s4	sn4	18

جدول stud

اگر پرس‌وجوی مطرح شده به شکل زیر بیان شود:

```
select *
from stud
      15
where avg > ( select avg(avg) from stud )
```

آنگاه پرس‌وجوی فوق، مشخصات دانشجویانی را از جدول stud استخراج می‌کند که معدل آن‌ها، از میانگین معدل تمامی دانشجویان بالاتر باشد. بنابراین خروجی پرس‌وجوی فوق به صورت زیر است:

std #	sname	avg
s3	sn3	16
s4	sn4	18

۳۱- گزینه (۱) صحیح است.

جدول work-on در صورت سؤال داده نشده است. اگر جدول زیر را فرض کنیم:

کارمند پروژه (مجری پروژه)	شماره پروژه	کارفرما
name	pno	Essn
Akbari	123	Alavi
Parvizi	123	Alavi
Sajadi	456	Hassani

هر سطر جدول فوق بیان می‌کند که کدام پروژه توسط کدام کارمند اجرا شده و کارفرمای آن پروژه چه کسی بوده است، می‌توان گفت که پرانتز داخلی:

(select pno from work-on where name='parvizi') پرس‌وجویی اسمای همه کارفرمایانی را می‌دهد که مجری آن‌ها پرویزی بوده و پرانتز بیرونی اسمای همه کارفرمایانی را می‌دهد که پروژه‌های آن‌ها توسط پرویزی اجرا شده است. بنابراین خروجی پرس‌وجوی مطرح شده به صورت زیر خواهد بود.

essn
Alavi

۳۲- گزینه (۳) صحیح است.

جدول زیر را در نظر بگیرید:

s #	sname	dept #	average
s1	sn1	dep1	9
s2	sn2	dep1	10
s3	sn3	dep2	11
s4	sn4	dep2	12
s5	sn5	dep2	13
s6	sn6	dep3	14
s7	sn7	dep3	15
s8	sn8	dep3	17
s9	sn9	dep3	18

جدول stud

گزینه اول نادرست است، زیرا بدون group by از having استفاده شده است، بنابراین پرس‌وجو دارای خطای نحوی است و از سوی کامپایلر اجرا نمی‌گردد.

گزینه دوم نادرست است، زیرا شروط سازگاری صفات در جلوی group by و select رعایت نشده است، همچنین در خط having average > avg (average) به جای average سمت چپ باید یک مقدار عددی قرار بگیرد. بنابراین این پرس‌وجو، خطای نحوی دارد و از سوی کامپایلر اجرا نمی‌گردد. اگر به طور مثال، گزینه select dept #

دوم به صورت زیر اصلاح گردد:

from stud

group by dept #

having avg (average) > 15

آن‌گاه، پرس‌وجوی فوق، شماره دانشکده‌هایی را می‌دهد که میانگین معدل آن دانشکده از 15 بالاتر است.

پرس‌وجوی فوق ابتدا براساس dept # به صورت زیر گروه‌بندی می‌شود و در ادامه براساس (average)

میانگین هر گروه محاسبه می‌گردد.

dept1	dept2
s1 sn1 9	s3 sn3 11
s2 sn2 10	s4 sn4 12
	s5 sn5 13

گروه اول

گروه دوم

$$\text{avg(average)} = \frac{19}{2} = 9.5 \quad \text{avg(average)} = \frac{36}{3} = 12$$

dept 3

s6 sn6 14

s7 sn7 15

s8 sn8 17

s9 sn9 18

گروه سوم

$$\text{avg(average)} = \frac{64}{4} = 16$$

سپس براساس دستور $\text{having avg (average) > 15}$ به عنوان شرط انتخاب گروه، تنها گروه سوم حائز شرط می‌باشد و انتخاب می‌گردد. بنابراین خروجی پرس‌وجو به صورت زیر خواهد بود.

توجه: دستور group by , سرگروه‌ها را، راهی خروجی می‌کند.

```
dept#
dep3
```

گزینه سوم درست است. به پرس‌وجو آن توجه کنید:

```
select *
from stud s1
where average > (select AVG (average)
                   from stud s2
                   where s1.dept # = s2.dept #)
```

پرس‌وجوی فوق سطر به سطر توسط عملگر where بر روی جدول $s1$ (جدول stud) حرکت می‌کند، و برای هر سطر از جدول $s1$ شرط جلوی where را چک می‌کند، اگر شرط جلوی where برقرار بود، سطر مورد نظر از جدول $s1$ انتخاب می‌گردد و این رویه تا به انتهای جدول $s1$ ادامه پیدا می‌کند. فرض کنید در سطر اول جدول $s1$ قرار داریم...

s #	sname	dept #	average
s1	sn1	dep1	9

با توجه به مقدار dept\# برابر $dep1$ از سطر جدول $s1$ ، پرس‌وجوی select داخلی، یعنی

```
select AVG (average)
from stud s2
where s1. dept# = s2.dept#)
```

$dep1$

میانگین معدل دانشجویانی را از جدول $s2$ محاسبه می‌کند که dept\# آنها برابر $dep1$ باشد، به بیان دیگر زیر پرس‌وجوی فوق میانگین معدل دانشجویان دانشکده $dep1$ را محاسبه می‌کند. دو جدول $s1$ و $s2$ که مطابق جدول stud هستند و پرس‌وجوی فوق را تجسم کنید. با توجه به اطلاعات دانشجویان دانشکده $dep1$ ، پر واضح است که میانگین معدل دانشجویان دانشکده $dep1$ ، برابر $\frac{9+10}{2} = \frac{19}{2} = 9.5$ می‌باشد.

بنابراین دانشجو $s1$ برای نمایش در خروجی انتخاب نمی‌گردد. زیرا:

```
select *
from stud s1
where average > (select AVG (average)
                     9
                     from stud s2
                     where s1. dept# = s2.dept#)
```

9.5 $dep1$

مشاهده می‌کنید که شرط فوق برای سطر اول جدول $s1$ برقرار نیست. بنابراین سطر اول از جدول $s1$ حائز شرایط نیست و برای نمایش در خروجی انتخاب نمی‌گردد. این رویه تا انتهای جدول $s1$ به صورت سطر به سطر، ادامه پیدا می‌کند.

برای مثالی دیگر سطر آخر جدول $s1$ را هم در نظر بگیرید.

با توجه به اطلاعات دانشجویان دانشکده dep3، پُر واضح است که میانگین دانشجویان دانشکده dep3، برابر $\frac{14+15+17+18}{4} = \frac{64}{4} = 16$ می باشد. بنابراین دانشجو s9 برای نمایش در خروجی انتخاب می گردد. زیرا:

```
select *
from stud s1
```

```
where average > (select AVG(average)
18      from stud S2
           where s1.dept # = s2.dept #)
16      dep3
```

مشاهده می کنید که شرط فوق برای سطر آخر جدول s1 بقرار است. بنابراین سطر آخر از جدول s1 حائز شرایط است و برای نمایش در خروجی، انتخاب می گردد. با توجه به اطلاعات دانشجویان دانشکده dep2

پُر واضح است که میانگین معدل دانشجویان دانشکده dep2 برابر $\frac{11+12+13}{3} = \frac{36}{3} = 12$ است.

بنابراین خروجی پرس و جوی مذکور در گزینه سوم به صورت زیر خواهد بود:

s #	sname	dept #	average
s2	sn2	dep1	10
s5	sn5	dep2	13
s8	sn8	dep3	17
s9	sn9	dep3	18

گزینه چهارم نادرست است، زیرا خطای معنایی دارد. این پرس و جو خطای نحوی ندارد و از سوی کامپایلر اجرا می گردد. اما پاسخ مورد نظر را برآورده نمی سازد.

```
select *
from stud
where average > (select AVG (average) from stud)
```

پرس و جوی فوق مشخصات دانشجویانی که معدلشان از متوسط معدل کلیه دانشجویان بیشتر است را لیست می نماید.

با توجه به اطلاعات جدول stud، پرس و جوی فوق به صورت زیر خواهد بود.

```
select *
from stud
where average > (select AVG (average) from stud)
```

13

بنابراین میانگین معدل کلیه دانشجویان، برابر 13 می شود. البته اگر نوع داده ای ستون average صحیح باشد، برابر 13 می شود، توجه کنید که اگر تابع عددی AVG بر روی ستونی با نوع داده ای صحیح میانگین انجام دهد و حاصل اعشاری گردد، تابع AVG پاسخ را گرد نمی کند، بلکه فقط قسمت صحیح را جدا می کند و به خروجی انتقال می دهد. همچنین اگر نوع داده ای ستون average اعشاری باشد، میانگین برابر 13.2 می شود. ما نوع داده ای average را در اینجا صحیح در نظر گرفته ایم.

بنابراین حاصل پرس و جوی فوق به صورت زیر خواهد بود:

s #	sname	dept #	average
s6	sn6	dep3	14
s7	sn7	dep3	15
s8	sn8	dep3	17
s9	sn9	dep3	18

۳۳- گزینه (۳) صحیح است.

راه حل اول:

جداول زیر را در نظر بگیرید:

<u>S#</u>	Sname	Level	..	<u>S#</u>	P#	QTY	<u>P#</u>	...
s1	sn1	6		s1	p1	10	p1	
s2	sn2	2		s1	p2	20	p2	
s3	sn3	3		s2	p1	30		

الگوهای چهارگانه (همه، هیچ، بعضی و حداقل) زیر را در نظر بگیرید:

الگوی اول (00):

SELECT S#
FROM S

WHERE NOT EXISTS (SELECT)

FROM P

WHERE NOT EXISTS (SELECT *

FROM SP

WHERE S.S# = SP.S#

AND P.P# = SP.P#))

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«شماره تولیدکنندگانی که، وجود نداشته باشد (NOT EXISTS) قطعه‌ای که، تولید نشده باشد

(**NOT EXISTS**). یعنی شماره تولیدکنندگانی که همه قطعات را تولیدکرده‌اند.»

با توجه به جداول فوق خروجی پرس و جوی الگوی اول به صورت زیر است:

S#
s1

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول P بررسی می‌گردد، تا مشخص گردد آیا هیچ سطروی در جدول P وجود ندارد که شرط اتصال = S.S# = SP.S# AND P.P# = SP.P# نظری از جدول S در خود نمایش داده می‌شود.

توجه: شرط لازم برای تبعیت از الگوی اول وجود NOT EXISTS اول و NOT EXISTS دوم است و شرط کافی برای تبعیت از الگوی اول وجود شرط اتصال متناسب و سازگار است.

الگوی دوم (01):

```

SELECT S#
FROM S
WHERE NOT EXISTS (SELECT *
                   FROM P
                   WHERE EXISTS (SELECT *
                                 FROM SP
                                 WHERE S.S# = SP.S#
                                   AND P.P# = SP.P#))

```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«شماره تولیدکننده کانی که، وجود نداشته باشد (NOT EXISTS) قطعه‌ای که، تولید شده باشد (EXISTS).
يعني شماره توليدکننده کانی که هیچ قطعه‌ای را تولید نکرده‌اند.»
با توجه به جداول فوق خروجی پرس و جوی الگوی دوم به صورت زیر است:

S#
s3

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول P بررسی می‌گردد، تا مشخص گردد آیا هیچ سطری در جدول P وجود ندارد که شرط اتصال = S.S# = SP.S# AND P.P# = SP.P# موجود در select داخلى را برقرار کند، اگر وجود نداشته باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

توجه: شرط لازم برای تبعیت از الگوی دوم وجود NOT EXISTS اول و EXISTS دوم است و شرط کافی برای تبعیت از الگوی دوم وجود شرط اتصال مناسب و سازگار است.

الگوی سوم (10):

```

SELECT S#
FROM S
WHERE EXISTS (SELECT *
               FROM P
               WHERE NOT EXISTS (SELECT *
                                 FROM SP
                                 WHERE S.S# = SP.S#
                                   AND P.P# = SP.P#))

```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«شماره تولیدکننده کانی که، وجود داشته باشد (EXISTS) قطعه‌ای که، تولید نشده باشد (NOT EXISTS).
يعني شماره توليدکننده کانی که بعضی قطعات را توليد نکرده‌اند.»
با توجه به جداول فوق خروجی پرس و جوی الگوی سوم به صورت زیر است:

S#
s2
s3

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول P بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول P وجود دارد که شرط اتصال = S.S# = SP.S# AND P.P# = SP.P# موجود در select داخلى را برقرار کند، اگر وجود نداشته باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

نظر از جدول S در خروجی نمایش داده می‌شود.

توجه: شرط لازم برای تبعیت از الگوی سوم وجود **NOT EXISTS** اول و **EXISTS** دوم است و شرط کافی برای تبعیت از الگوی سوم وجود شروط اتصال مناسب و سازگار است.

الگوی چهارم (11):

```
SELECT S#
FROM S
WHERE EXISTS (SELECT *
               FROM P
               WHERE EXISTS (SELECT *
                             FROM SP
                             WHERE S.S# = SP.S#
                             AND P.P# = SP.P#))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«شماره تولیدکنندگانی که، وجود داشته باشد (**EXISTS**) قطعه‌ای که، تولید شده باشد».

شماره تولیدکنندگانی که حداقل یک قطعه را تولید کرده‌اند».

با توجه به جداول فوق خروجی پرس و جوی الگوی چهارم به صورت زیر است:

S#

s1

s2

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول P بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول P وجود دارد که شرط اتصال S.S# = SP.S# AND P.P# = SP.P# موجود در select داشته باشد، اگر وجود داشته باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

توجه: شرط لازم برای تبعیت از الگوی چهارم وجود **EXISTS** اول و **NOT EXISTS** دوم است و شرط کافی برای تبعیت از الگوی چهارم وجود شروط اتصال مناسب و سازگار است.

پرس و جوی مطرح شده در گزینه دوم را در نظر بگیرید:

```
SELECT S.sname
FROM Sailor S
WHERE NOT EXISTS (SELECT B.bid
                   FROM Boats B
                   WHERE NOT EXISTS (SELECT R.bid
                                     FROM Reserves R
                                     WHERE S.sid = R.sid
                                     AND B.bid = R.bid))
```

توجه: پرس و جوی مطرح شده در گزینه دوم، مطابق الگوی اول می‌باشد.

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«نام ملوان‌هایی که، وجود نداشته باشد (**NOT EXISTS**) قایقی که، رزرو نشده باشد».

یعنی نام ملوان‌هایی که همه قایق‌ها را رزرو کرده‌اند».

توجه: عملگر تقسیم در جبرابطه‌ای به فرم زیر نیز در SQL قابل پیاده‌سازی است:

```
SELECT S.sname
FROM Sailor S
WHERE NOT EXISTS ((SELECT B.bid
                      FROM Boats B)
EXCEPT
(SELECT R.bid
  FROM Reserves R
 WHERE S.Sid = R.Sid))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«نام ملوان‌هایی که، وجود نداشته باشد (NOT EXISTS) قایقی که، رزرو نشده باشد (EXCEPT). یعنی نام ملوان‌هایی که همه قایق‌ها را رزرو کرده‌اند.»

بنابراین هر دو گزینه اول و دوم نام ملوان‌هایی که همه قایق‌ها را رزرو کرده‌اند، استخراج می‌کند، بنابراین پاسخ سوال گزینه سوم است.

راه حل دوم:

شماره‌ی همه‌ی قایق‌ها



```
select S.sname from Sailor S where not Exists
  
$$\left( \begin{array}{l} \text{(select B.bid from Boats B)} \\ \text{Except} \\ \text{(select R.bid from Reserves R)} \\ \text{where S.sid = R.sid) } \end{array} \right)$$

```

توجه: توسط دستور where در خط زیر:

select S.sname from Sailor S where not exists (....)

برای هر سطر از جدول Sailor شرط جلوی not exists که حاصل یک تفاضل توسط عملگر except می‌باشد، محاسبه می‌گردد، اگر تهی بود، شرط جلوی where که همان not exists است، TRUE می‌گردد و سطر مورد نظر از جدول Sailor انتخاب می‌گردد و این رویه برای تک تک سطرهای جدول Sailor، تا به انتهای جدول Sailor ادامه پیدا می‌کند. به بیان دیگر گزینه‌ی اول نام ملوان‌هایی را می‌دهد که پرانتز مقابل not exists برای آن‌ها تهی است. این پرانتز هنگامی تهی می‌شود که حاصل تفاضل بیان شده در این پرانتز تهی شود. حاصل این تفاضل در صورتی تهی می‌شود که تعداد کل قایق‌ها برابر تعداد قایق‌هایی باشد که آن ملوان تاکنون رزرو کرده است. به عبارت دیگر گزینه اول نام ملوان‌هایی را می‌دهد که تاکنون تمام قایق‌ها را رزرو کرده‌اند. می‌توانید برای تجسم بهتر جدول‌های زیر را در نظر بگیرید:

<u>sid</u>	sname	...	<u>sid</u>	<u>Bid</u>	...	<u>bid</u>	...
s1	sn1		s1	b1		b1	
s2	sn2		s1	b2		b2	
s3	sn3		s2	b1			
<i>Sailor</i> <i>S</i>			<i>Reserves</i> <i>R</i>			<i>Boats</i> <i>B</i>	
ملوان			رزرو			قایق	

توجه: همان طور که مشاهده می‌کنید، ملوان s_1 همه‌ی قایق‌ها را رزرو کرده است.
به بیان دیگر $B.bid$ from Boats select جلوی دستور not exists شماره همه‌ی قایق‌های موجود در جدول قایق را نمایش می‌دهد.
با توجه به جداول فوق داریم:

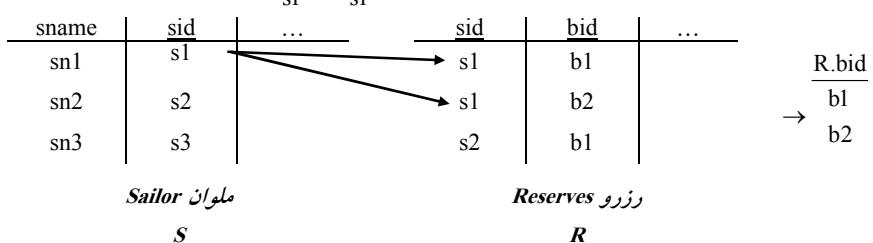
B.bid
b1
b2

هم‌چنان جمله‌ی بعد از دستور Except یعنی:

(select R.bid
from Reserves R
where S.sid=R.sid

توسط دستور $S.sid=R.sid$ به محیط خارج یعنی جدول S یا Sailor متصل می‌گردد. حال به ازای حرکت در هر سطر از جدول Sailor، یک بار به طور کامل سطرهای جدول رزرو از ابتدا تا انتها با توجه به شرط اتصال بررسی می‌گردد. مطابق شکل زیر:

ابتدا برای سطر اول از جدول Sailor با توجه به شرط اتصال $S.sid = R.sid$ داریم:



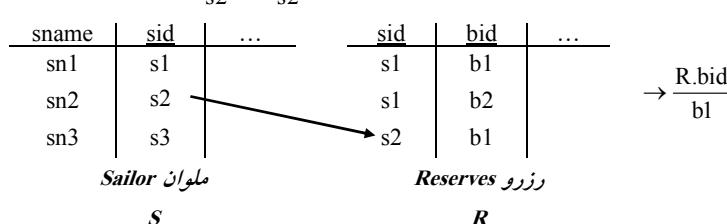
خروجی حاصل از عملگر تفاضل جلوی not exists به ازای سطر اول از جدول Sailor به صورت زیر است:

$$\frac{B.bid}{\begin{matrix} b1 \\ b2 \end{matrix}} - \frac{R.bid}{\begin{matrix} b1 \\ b2 \end{matrix}} = \frac{\text{خروجی}}{\text{تهی}}$$

جلوی not exists برابر تهی گردید، بنابراین شرط where در پشت not exists TRUE می‌گردد. بنابراین سطر اول از جدول Sailor به شکل زیر در خروجی نمایش داده می‌شود.

Sname
Sn1

حال برای سطر دوم از جدول Sailor با توجه به شرط اتصال $S.sid = R.sid$ داریم:



خروجی حاصل از عملگر تفاضل جلوی `not exists` به ازای سطر دوم از جدول Sailor به صورت زیر است:

$$\frac{B.bid}{b1} - \frac{R.bid}{b1} = \frac{\text{خروچی}}{b2}$$

جلوی `not exists` برابر غیر تهی گردید، بنابراین شرط where در پشت FALSE می‌گردد، بنابراین سطر دوم از جدول Sailor در خروجی نمایش داده نخواهد شد.

در ادامه برای سطر سوم از جدول Sailor با توجه به شرط اتصال $S.sid = R.sid$ داریم:

<i>sname</i>	<i>sid</i>	...	<i>sid</i>	<i>bid</i>	...	$\rightarrow \frac{R.bid}{\text{تهی}}$
sn1	s1		s1	b1		
sn2	s2		s1	b2		
sn3	s3	هیچ	s2	b1		

Sailor ملوان *Reserves* رزرو *R*

خروجی حاصل از عملگر تفاضل جلوی `not exists` به ازای سطر سوم از جدول S به صورت زیر است:

$$\frac{B.bid}{b1} - \frac{R.bid}{\text{تهی}} = \frac{\text{خروچی}}{b2}$$

جلوی `not exists` برابر غیر تهی گردید، بنابراین شرط where در پشت FALSE می‌گردد، بنابراین سطر سوم از جدول Sailor در خروجی نمایش داده نخواهد شد.

بنابراین خروجی نهایی برای «نام ملوان‌هایی که همه قایق‌ها را رزرو کرده‌اند» به صورت زیر است:

Sname
Sn1

حال به بررسی گزینه دوم می‌پردازیم:

select S.sname from sailor S where not exists (select B.bid from Boats B where not exists (select R.bid from Reserves R where

$R.bid=B.bid$ and $R.sid=S.sid$)

الگوریتم فوق به این شکل حرکت می‌کند، به شرح الگوریتم دقت کنید:

ابتدا سطر اول از جدول ملوان بررسی می‌شود تا بررسی کامل در همین سطر توقف کنید، اگر جلوی اول تهی باشد، نام ملوان از جدول Sailor در خروجی به عنوان ملوانی که همه قایق‌ها را رزرو کرده است، نمایش داده می‌شود. اما مساله این است که چه وقت جلوی `not exists` اول تهی می‌گردد، پاسخ این است که هیچ `B.bid` در جمله:

where select B.bid from Boats B where در همان جمله برقرار نباشد، واضح است که وقتی این اتفاق می‌افتد که شرط where برقرار نیست، پاسخ، وقتی که پرس‌وجوی داخل

دوم تهی نباشد، یعنی قایق رزرو شده باشد. در یک بیان ساده، ابتدا سطر اول از جدول ملوان را انتخاب می‌کنیم. مثلاً ملوان s_1 سپس برای ملوان s_1 از جدول ملوان بررسی می‌کنیم، که آیا همه قایق‌های موجود در جدول قایق برای ملوان s_1 رزرو شده‌اند یا خیر. بنابراین برای یک ملوان خاص، توسط دستور select جلوی not exists اول، بر روی جدول قایق از ابتدا تا انتهای جدول به ازای یک ملوان خاص، حرکت می‌کنیم، اما به ازای حرکت در هر سطر از جدول قایق، توسط دستور select جلوی not exists دوم، بررسی می‌گردد که قایق موجود در جدول قایق، در جدول رزرو، رزرو شده است یا خیر. اگر رزرو شده باشد، توسط R.bid داخل دستور select جلوی not exists دوم نمایش داده می‌شود و توسط B.bid داخل دستور select جلوی not exists اول نمایش داده نمی‌شود به دلیل FALSE بودن شرط دستور where اول ناشی از بودن not exists دوم به دلیل تهی نبودن جلوی not exists در جلوی not exists دوم. حال اگر مطابق روال فوق به ازای همه قایق‌های موجود در جدول قایق، برای یک ملوان خاص، همه قایق‌ها، رزرو شده باشند، باعث می‌شود تا داخل دستور select جلوی not exists اول تهی گردد و باعث شود، شرط دستور where در پشت B.bid اول TRUE not exists گردد، بنابراین آن ملوان خاص موجود در جدول ملوان در خروجی نمایش داده می‌شود.

پرس‌وجوی مطرح شده در گزینه دوم را مجدداً در نظر بگیرید:

select S.name from Sailor S where not exists (select B.bid from Boats B where not exists (select R.bid from Reserves R where

$$\begin{array}{l} R.bid = B.bid \text{ and } \\ \boxed{R.sid = S.sid} \\ sl \quad sl \end{array}$$

به کادر مستطیل شکل مشخص شده در پرس‌وجوی فوق توجه کنید به ازای حرکت در هر سطر جدول Sailor جمله R.sid=S.sid تغییر می‌کند. هم اکنون در سطر اول جدول Sailor هستیم. پس sid برابر s_1 می‌باشد.

به قطعه کد زیر موجود در جلوی not exists اول، دقت کنید:

خروجی

not Exists (select B.bid from Bots B where
↓

ابتدا سطر اول از جدول قایق یعنی b_1 بررسی می‌شود.

not Exist (select R.bid from Reserves R where

↓
خروجی

با شرط زیر بررسی می‌شود که ملوان s_1 ، قایق b_1 را رزرو کرده است یا خیر.

$$R.b1 = B.b1 \text{ and } R.sl = S.sl))$$

$$\begin{array}{c} \text{TRUE} \quad \text{TRUE} \\ \hline \text{TRUE} \end{array}$$

شرط برقرار است، زیرا قایق b_1 توسط ملوان s_1 رزرو شده است، بنابراین داریم:

not Exists (select ~~R.bid~~ from Reserves R where

b1

چون جلوی not exists دوم غیر تهی است، بنابراین شرط where موجود پشت not exists دوم می‌گردد، بنابراین سطر اول جدول قایق یعنی قایق b1 در خروجی دستور select جلوی not exists اول ظاهر نمی‌شود. چون قایق b1 توسط ملوان s1 رزرو شده است.

در ادامه سطر دوم از جدول قایق یعنی b2 همچنان برای ملوان s1 بررسی می‌شود. با شرط زیر بررسی می‌شود که ملوان s1، قایق b2 را رزرو کرده است یا خیر.

$R.b2 = B.b2 \text{ and } R.s1 = S.s1)$

$$\begin{array}{c} \text{TRUE} \quad \text{TRUE} \\ \hline \text{TRUE} \end{array}$$

شرط برقرار است، زیرا قایق b2 توسط ملوان s1 رزرو شده است، بنابراین داریم:

not Exists (select ~~R.bid~~ from Reserves R where
b2

چون جلوی not exists دوم غیر تهی است، بنابراین شرط where موجود پشت not exists دوم می‌گردد، بنابراین سطر دوم جدول قایق یعنی قایق b2 نیز در خروجی دستور select جلوی not exists اول ظاهر نمی‌شود. چون قایق b2 توسط ملوان s1 رزرو شده است. نتیجه این که ملوان s1 همه قایق‌ها را رزرو کرده است.

پس داریم: select s.sname from sailor S where not Exists (select B.bid from Boats B wehre

در نهایت مقدار B.bid در خط فوق در جلوی not exists اول برای ملوان s1 تهی می‌گردد، چون به دلیل شرایط جلوی not exists دوم مشاهده نمودید که هیچ یک از سطرهای جدول قایق در خروجی دستور select جلوی not exists اول ظاهر نشدند. چون قایق‌های b1 و b2 توسط ملوان s1 رزرو شده بودند. در یک بیان ساده اگر قایقی برای یک ملوان خاص رزرو شده باشد در خروجی select B.bid در خط فوق

قرار نمی‌گیرد. رزرو قایق توسط دستور select در جلوی not exists دوم بررسی می‌گردد.

جلوی not exists اول، برای سطر اول جدول ملوان، یعنی ملوان s1، تهی است، زیرا در خروجی select B.bid جلوی not exists اول، قایق‌های رزرو نشده، توسط یک ملوان خاص، خارج می‌گردند، از آنجا که پس از بررسی قایق‌های رزرو شده برای ملوان s1 توسط دستور select در جلوی not exists مشخص شد که ملوان s1، همه قایق‌ها را رزرو کرده است، بنابراین خروجی select B.bid جلوی not exists اول، تهی خواهد بود، چون همه قایق‌ها توسط ملوان s1 رزرو شده‌اند. پس شرط where در sn1 پشت not exists اول TRUE است، پس سطر اول جدول ملوان انتخاب می‌گردد و با مقدار

در خروجی قرار می‌گیرد، زیرا ملوان s1، همه قایق‌ها را رزرو کرده بود.

حال سطر دوم از جدول ملوان بررسی می‌شود، تا بررسی کامل در همین سطر توقف کنید.

به قطعه کد زیر، موجود در جلوی not exists اول، دقت کنید: خروجی

not Exists (select B.bid from Boats B where

↓

ابتدا سطر اول از جدول قایق یعنی b1 بررسی می‌شود.

not Exists (select R.bid from Reserves R where

↓
خرسچی

با شرط زیر بررسی می‌شود که ملوان 2، قایق b1 را رزرو کرده است یا خیر.

$R.b1 = B.b1 \text{ and } R.s2 = S.s2)$)

$\frac{\text{TRUE} \quad \text{TRUE}}{\text{TRUE}}$

شرط برقرار است، زیرا قایق b1 توسط ملوان 2 رزرو شده است، بنابراین داریم:

not Exists (select ~~R.bid~~ from Reserves R where
b1

چون جلوی not exists دوم غیر تهی است، بنابراین شرط where موجود پشت not exists دوم می‌گردد. بنابراین سطر اول جدول قایق یعنی قایق b1 در خرسچی دستور select جلوی not exists اول ظاهر نمی‌شود. چون قایق b1 توسط ملوان 2 رزرو شده است. در ادامه سطر دوم از جدول قایق یعنی b2 همچنان برای ملوان 2 بررسی می‌شود.

با شرط زیر بررسی می‌شود که ملوان 2، قایق b2 را رزرو کرده است یا خیر.

$\frac{R.b2 \leq B.b2 \text{ and } R.s2 = S.s2}{\frac{\text{FALSE} \quad \text{TRUE}}{\text{FALSE}}}$

شرط برقرار نیست، زیرا قایق b2 توسط ملوان 2 رزرو نشده است، بنابراین داریم:

not exists (select ~~R.bid~~ from Reserves R where
تهی

چون جلوی not exists دوم تهی است، بنابراین شرط where موجود پشت not exists دوم می‌گردد. بنابراین سطر دوم جدول قایق یعنی قایق b2 در خرسچی دستور select جلوی not exists اول ظاهر می‌شود. چون قایق b2 توسط ملوان 2 رزرو نشده است. نتیجه این که ملوان 2 همه قایق‌ها را رزرو نکرده است.

پس داریم:
select s.sname from sailor S where not exists (select B.bid from Boots B where
در نهایت مقدار B.bid در خط فوق جلوی not exists اول برای ملوان 2 غیر تهی می‌گردد و برابر مقدار b2، چون به دلیل شرایط جلوی not exists دوم مشاهده نمودید که سطر دوم از جدول قایق در خرسچی دستور select جلوی not exists اول ظاهر شد، چون قایق b2 توسط ملوان 2 رزرو نشده بود. جلوی not exists اول، برای سطر دوم جدول ملوان، یعنی ملوان 2، غیر تهی است، پس شرط where در پشت not exists اول FALSE است، پس سطر دوم جدول ملوان انتخاب نمی‌گردد، زیرا ملوان 2 همه‌ی قایق‌ها را رزرو نکرده بود.

به همین ترتیب سطرا سوم جدول ملوان نیز انتخاب نمی‌گردد، زیرا ملوان 3 همه قایق را رزرو نکرده است. نتیجه اینکه، گزینه دوم نیز نام ملوان‌هایی را می‌دهد که قایقی وجود ندارد که آن‌ها آن را رزرو نکرده باشند. به عبارت دیگر نام ملوان‌هایی را می‌دهد که تاکنون همه قایق‌ها را رزرو کرده‌اند.

۳۴- گزینه (۳) صحیح است.

عبارت الف درست است. زیرا، میانگین موجودی حساب‌های افتتاح شده در شعبه ملی را به عنوان خرسچی

می‌دهد. همچنین استفاده از تابع `avg` به درستی انجام شده است.
عبارت ب درست است. زیرا، نام شعبه و میانگین موجودی حساب‌ها را در مورد شعبی که میانگین موجودی حساب‌های افتتاح شده در آن‌ها از ۱۲۰۰ بیشتر است، می‌دهد. همچنین صفت همراه گروه‌بندی در خروجی آمده است.

عبارت ج درست است. زیرا، نام کلیه شعبی را می‌دهد که موجودی آن‌ها حداقل از موجودی یکی از شعب شهر اصفهان بیشتر است.
بنابراین هر سه عبارت صحیح است.

۳۵- گزینه (۲) صحیح است.

جدول زیر را در نظر بگیرید:

S#	P#	QTY
s1	p1	10
s1	p2	20
	p3	30
s2	p1	40
	p2	50
s3	p3	60

در اجرای این پرس‌وجو ابتدا کلیه سطرهای جدول SP براساس شماره قطعه، گروه‌بندی می‌شوند. سپس تابع عددی `sum` روی اعضای هر گروه به صورت جداگانه اجرا می‌شود.

$$\begin{array}{ccc}
 p1 & p2 & p3 \\
 \begin{array}{cc} s1 & 10 \\ s2 & 40 \\ \hline +50 \end{array} & \begin{array}{cc} s1 & 20 \\ s2 & 50 \\ \hline +70 \end{array} & \begin{array}{cc} s1 & 30 \\ s3 & 60 \\ \hline +90 \end{array}
 \end{array}$$

در نهایت پرس‌وجوی مطرح شده، شماره و تعداد کل هر قطعه تولید شده را به عنوان خروجی نمایش می‌دهد.

P#	TOTQTY
p1	50
p2	70
p3	90

توجه: پس از گروه‌بندی همواره سر گروه‌ها در اینجا، p1، p2 و p3 در خروجی نشان داده می‌شوند.

۳۶- گزینه (۱) صحیح است.

به جداول زیر دقت کنید:

S#	Sname	...	S#	P#	QTY	P#	Pname	...
s1	sn1		s1	p1	10	p1		
s2	sn2		s1	p2	20	p2		
s3	sn3		s2	p2	30			
			s3	p2	40	جدول p		
جدول s			جدول sp			جدول p		

select S.snmae from S where S.s# in (select from SP where sp.p#= 'p2')

پرانتز داخلی شماره تولید کنندگانی را می دهد که قطعه p2 را تولید کرده اند. خروجی آن به صورت زیر است:

SP.s#

s1

s2

s3

select بیرونی نام تولید کنندگانی را می دهد که قطعه p2 را تولید کرده اند، البته چون به دلیل کلید نبودن Sname ممکن است، اسمای برخی تولید کنندگان شبیه هم باشد امکان دارد اسمای تکراری در خروجی این دستور مشاهده شود. بنابراین خروجی Sname به صورت زیر خواهد بود:

S.Sname

s1

s2

s3

- گزینه (۲) صحیح است.

جداول زیر را در نظر بگیرید:

isbn	bname	type	ssn	isbn
isbn1	b1	technical	ss1	isbn1
isbn2	b2	fiction	ss1	isbn2
isbn3	b3	self-help	ss2	isbn3
isbn4	b4	technical	ss3	isbn4

جدول کتاب Books B

جدول نویسنده Authors A

گزینه اول درست است. زیرا، ابتدا نویسندهای که کتابی از نوع fiction نوشته اند، شناسایی می شوند و سپس حاصل تفاضل آنها از کل نویسندهای، خروجی خواسته شده را نمایش می دهد.

select ssn (select ssn
from Authors Except from Authors A , Books B
 where A.isbn=B.isbn and B.type = 'fiction')

ssn	ssn	ssn
ss1	ss1	ss2
ss1	Except	= ss3
ss2		
ss3		

گزینه سوم درست است. زیرا، شماره نویسندهای که هیچ کتابی از نوع fiction برای آنها ثبت نشده است به عنوان خروجی داده می شود که همان خروجی مورد نظر است.

select ssn
from Authors A
where Not Exists (select *
 from Books B
 where A.isbn=B.isbn AND B.type= 'fiction')

روال حرکت به این صورت است که به ازای هر سطر از جدول Authors برسی می‌شود که در پرانتز جلوی not exists با توجه به شرط آن سطري موجود است یا خیر، اگر سطري موجود نباشد، یعنی تهی باشد شرط جلوی where بیرونی، TRUE می‌گردد و سطر مورد نظر از جدول Authors انتخاب می‌گردد. گزینه چهارم درست است. زیرا، شماره نویسنده‌گانی داده می‌شود که عضو مجموعه نویسنده‌گان کتاب‌هایی از نوع fiction نیستند، بنابراین این گزینه نیز خروجی مورد نظر را می‌دهد.

```
select ssn
from Authors
where ssn not in (select ssn
                    from Authors A , Books B
                    where A.isbn=B.isbn AND B.type= 'fiction'
```

<u>ssn</u>	not in	<u>ssn</u>	=	<u>ssn</u>
ss1		ss1		ss2
ss1				ss3
ss2				
ss3				

گزینه دوم نادرست است. زیرا، شماره نویسنده‌گانی را می‌دهد که کتابی از غیر نوع fiction تألیف کرده‌اند. اما ممکن است نویسنده‌ای کتابی از نوع غیر fiction تألیف کرده باشد (پس در پاسخ این پرس‌وجو ظاهر می‌شود) و در عین حال کتابی از نوع fiction هم تألیف کرده باشد (پس نباید در خروجی ظاهر می‌شد که می‌شود). پس این گزینه نادرست است. در واقع انتظار این بود که شماره ssn برای نویسنده‌گانی که هیچ کتابی از نوع fiction ننوشته‌اند در خروجی ظاهر شوند، در حالی که با توجه به پرس‌وجو گزینه دوم شماره ssn با مقدار ss در خروجی نمایش داده می‌شود که با پرس‌وجو مطرح شده در صورت سؤال تناقض دارد.

```
select ssn
from Authors, Books
where Authors.isbn= Books. isbn AND type <> 'fiction'
```

پس از ضرب دکارتی دو جدول Authors و Books براساس نوع اتصال و شرط ضرب داریم:

ssn	Authors.isbn	Books.isbn	Type	...
✓ ss1	isbn1	isbn1	technical	
✗ ss1	isbn2	isbn2	fiction	
✓ ss2	isbn3	isbn3	self-help	
✓ ss3	isbn4	isbn4	technical	

خروجی به صورت زیر خواهد بود:

<u>ssn</u>
ss1
ss2
ss3

- ۳۸ - گزینه (۳) صحیح است.

دو جدول Account و depositor با مقادیر زیر را در نظر بگیرید:

acc#	bname	...	cname	acc#	...
1213	Perryridge		alavi	1213	
1214	Brooklyn		alavi	1214	
1215	Perryridge		hassani	1215	
1216	Perryridge		hassani	1216	
			asgari	1217	

جدول حساب Account

جدول مشتری depositor T یا R

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

```
select cname
from depositor as T, account
where T.acc# = account.acc# and bname='perryridge'
group by cname
having count(*)<=1
```

با توجه به جداول فوق، ابتدا خروجی ضرب دکارتی و شرط اتصال به صورت زیر محاسبه می‌گردد:

cname	acc#	acc#	bname
alavi	1213	1213	Perryridge
hassani	1215	1215	Perryridge
hassani	1216	1216	Perryridge

همچنین در ادامه، پس از انجام دستور group by cname ستون cname خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:

alavi 1213 1213 Perryridge	hassani 1215 1215 Perryridge 1216 1216 Perryridge
گروه اول	

و در نهایت دستور $\text{having count}(* <= 1)$ برای هر گروه به طور مستقل اعمال می‌گردد.
توجه: دستور HAVING بر روی گروه‌ها، اعمال می‌گردد.

alavi 1213 1213 Perryridge	hassani 1215 1215 Perryridge 1216 1216 Perryridge
گروه اول	
$1 <= 1$	$2 <= 1$

توجه: با توجه به شرط انتخاب گروه توسط دستور $\text{having count}(* <= 1)$ ، فقط گروه اول جهت نمایش در خروجی انتخاب می‌شود.

و در نهایت دستور cname داخل دستور select برای هر گروه انتخاب شده توسط دستور Having به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد، بنابراین خروجی نهایی پرس و جو به صورت زیر است:

cname
alavi

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

بنابراین مطابق پرس و جوی مطرح شده در گزینه‌ی اول، «نام مشتریانی استخراج می‌شود که در شعبه Perryidge فقط و فقط یک حساب بانکی دارند، نه کمتر و نه بیشتر»

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

```
select cname
from depositor as T, account
where T.acc# = account.acc# and bname='perryidge'
group by cname
having count(T.acc#)<=1
```

با توجه به جداول فوق، ابتدا خروجی ضرب دکارتی و شرط اتصال به صورت زیر محاسبه می‌گردد:

cname	acc#	acc#	bname
alavi	1213	1213	Perryidge
hassani	1215	1215	Perryidge
hassani	1216	1216	Perryidge

همچنین در ادامه، پس از انجام دستور group by cname براساس ستون cname خروجی پرس و جو به صورت زیر گروهبندی می‌شود:

alavi	hassani
1213 1213 Perryidge	1215 1215 Perryidge
	1216 1216 Perryidge
گروه اول	
گروه دوم	

و در نهایت دستور having count(T.acc#)<=1 برای هر گروه به طور مستقل اعمال می‌گردد. توجه: دستور HAVING بر روی گروهها، اعمال می‌گردد.

alavi	hassani
1213 1213 Perryidge	1215 1215 Perryidge
	1216 1216 Perryidge
گروه اول	
گروه دوم	
1<=1, 2<=1	

توجه: با توجه به شرط انتخاب گروه توسط دستور having count(T.acc#)<=1، فقط گروه اول جهت نمایش در خروجی انتخاب می‌شود.

و در نهایت دستور select داخل دستور Having برای هر گروه انتخاب شده توسط دستور Having به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد، بنابراین خروجی نهایی پرس و جو به صورت زیر است:

cname
alavi

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

بنابراین مطابق پرس و جوی مطرح شده در گزینه‌ی دوم، «نام مشتریانی استخراج می‌شود که در شعبه

فقط و فقط Perryridge یک حساب بانکی دارند، نه کمتر و نه بیشتر» که مطابق پرس و جوی مطرح شده داده اند.

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

```
select cname  
from depositor as T, account  
where T.acc# = account.acc# and bname='perryidge'  
group by cname  
having count(Distinct bname) <=1
```

با توجه به جداول فوق، ابتدا خروجی ضرب دکارتی و شرط اتصال به صورت زیر محاسبه می‌گردد:

cname	acc#	acc#	byname
alavi	1213	1213	Perryridge
hassani	1215	1215	Perryridge
hassani	1216	1216	Perryridge

همچنین در ادامه، پس از انجام دستور group by cname براساس ستون cname خروجی پرس و جو به صورت زیر گروهبندی می شود:

alavi	hassani
1213 1213 Perryidge	1215 1215 Perryidge
<hr/>	
گروه اول	گروه دوم

توجه: دستور HAVING پر روی گروهها، اعمال می‌گردد.
و در نهایت دستور $\text{having count(Distinct bname)} = 1$ برای هر گروه به طور مستقل اعمال می‌گردد.

alavi	hassani
1213 1213 Perryidge	1215 1215 Perryidge
<hr/>	
گروه اول	گروه دوم
1 <= 1,	1 <= 1

توجه: به عملکرد مربوط به عملگر Distinct و حذف تکرار در سطرهای مربوط به ستون bname دقت کنند.

توجه: با توجه به شرط انتخاب گروه توسط دستور `having count(Distinct bname) <= 1`، گروههای اول و دوم جهت نمایش در خروجی انتخاب می‌شوند.

و در نهایت دستور select داخل دستور cname برای هر گروه انتخاب شده توسط دستور Having به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد، بنابراین خروجی نهایی پرس و جو به صورت زیر است:

cname
alavi
hassani

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

بنابراین مطابق پرس و جوی مطرح شده در گزینه‌ی سوم، «نام مشتریانی استخراج می‌شود که در شعبه

Perryidge حداقل یک حساب بانکی دارند.» که البته پاسخ سوال هم همین گزینه است که حاصل پرس و جوی آن، با سایر گزینه‌ها متفاوت است.

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

```
select T cname
from depositor as T
where UNIQUE (select R cname
               from account, depositor as R
               where T cname = R cname and
                     R acc# = account acc# and bname='perryidge')
```

با توجه به وجود دستور UNIQUE، زیر پرس و جوی داخلی فوق یک Correlated Subquery است، یعنی به ازای حرکت در هریک از سطوح 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 هر بار اجرای حلقه خارجی، یک بار به طور کامل حلقه داخلی اجرا می‌گردد.
توسط دستور where در فرم زیر:

```
select T cname
from depositor as T
where UNIQUE (...)
```

برای هر سطر از جدول depositor شرط جلوی UNIQUE که حاصل یک مقایسه می‌باشد، محاسبه می‌گردد، اگر فقط و فقط یک سطر وجود داشت یعنی نه کمتر و نه بیشتر، آنگاه شرط جلوی where همان UNIQUE است، TRUE می‌گردد و سطر مورد نظر از جدول depositor انتخاب می‌گردد و این رویه برای تک تک سطوح‌های جدول depositor، تا به انتهای جدول depositor ادامه پیدا می‌کند، دقت کنید که دستور UNIQUE سور یکه است. به بیان دیگر این پرس و جو نام مشتریانی را می‌دهد که در پرانتز مقابل UNIQUE برای آن‌ها فقط و فقط یک سطر وجود داشته باشد. در این پرانتز هنگامی فقط و فقط یک سطر وجود دارد که در حاصل مقایسه بیان شده در این پرانتز فقط و فقط یک سطر وجود داشته باشد. به عبارت دیگر در پرس و جوی گزینه‌ی چهارم نام مشتریانی استخراج می‌شود که در شعبه Perryidge فقط و فقط یک حساب بانکی دارند، نه کمتر و نه بیشتر. که مطابق پرس و جوی مطرح شده در گزینه‌های اول و دوم است. بنابراین مطابق پرس و جوی مطرح شده در گزینه‌ی چهارم، «نام مشتریانی استخراج می‌شود که در شعبه Perryidge فقط و فقط یک حساب بانکی دارند، نه کمتر و نه بیشتر» عبارت موجود در جلوی دستور UNIQUE که به صورت زیر است:

```
select R cname
from account, depositor as R
where T cname = R cname and
      R acc# = account acc# and bname='perryidge')
```

توسط دستور 'T cname = R cname and R acc# = account acc# and bname = 'perryidge' به محیط خارج یعنی جدول depositor متصل می‌گردد. حال به ازای حرکت در هر سطر از جدول depositor، یک بار به طور کامل سطوح‌های حاصل از ضرب دکارتی دو جدول account و depositor از ابتدا تا انتها با توجه به شرط اتصال بررسی می‌گردد. مطابق شکل زیر:

ابتدا برای سطر اول از جدول depositor باتوجه به شرط اتصال $T.cname = R.cname$ and $R.acc\# = account.acc\#$ and $bname = 'perryidge'$ داریم:

T.cname	acc#	...
alavi	1213	
alavi	1214	
hassani	1215	
hassani	1216	
asgari	1217	

جدول مشتری depositor T

R.cname	R.acc#	account.acc#	bname
alavi	1213	1213	Perryidge
hassani	1215	1215	Perryidge
hassani	1216	1216	Perryidge

جلوی UNIQUE فقط و فقط یک سطر وجود دارد، بنابراین شرط where در پشت TRUE می‌گردد. بنابراین سطر اول از جدول depositor در خروجی نمایش داده می‌شود. در ادامه برای سطر دوم از جدول depositor باتوجه به شرط اتصال $T.cname = R.cname$ and $R.acc\# = account.acc\#$ and $bname = 'perryidge'$ داریم:

T.cname	acc#	...
alavi	1213	
alavi	1214	
hassani	1215	
hassani	1216	
asgari	1217	

جدول مشتری depositor T

R.cname	R.acc#	account.acc#	bname
alavi	1213	1213	Perryidge
hassani	1215	1215	Perryidge
hassani	1216	1216	Perryidge

جلوی UNIQUE فقط و فقط یک سطر وجود دارد، بنابراین شرط where در پشت TRUE می‌گردد. بنابراین سطر دوم از جدول depositor در خروجی نمایش داده می‌شود. در ادامه برای سطر سوم از جدول depositor باتوجه به شرط اتصال $T.cname = R.cname$ and $R.acc\# = account.acc\#$ and $bname = 'perryidge'$ داریم:

T.cname	acc#	...
alavi	1213	
alavi	1214	
hassani	1215	
hassani	1216	
asgari	1217	

جدول مشتری depositor T

R.cname	R.acc#	account.acc#	bname
alavi	1213	1213	Perryidge
hassani	1215	1215	Perryidge
hassani	1216	1216	Perryidge

جلوی UNIQUE فقط و فقط یک سطر وجود ندارد یعنی دو سطر وجود دارد، بنابراین شرط where در پشت UNIQUE می‌گردد. بنابراین سطر سوم از جدول depositor در خروجی نمایش داده نمی‌شود.

در ادامه برای سطر چهارم از جدول depositor با توجه به شرط اتصال T.cname = R.cname and R.acc# = account.acc# and bname = 'perryidge' داریم:

T.cname	acc#	...
alavi	1213	
alavi	1214	
hassani	1215	
hassani	1216	
asgari	1217	

جدول مشتری depositor T

R.cname	R.acc#	account.acc#	bname
alavi	1213	1213	Perryidge
hassani	1215	1215	Perryidge
hassani	1216	1216	Perryidge

جلوی UNIQUE فقط و فقط یک سطر وجود ندارد یعنی دو سطر وجود دارد، بنابراین شرط where در پشت UNIQUE می‌گردد. بنابراین سطر چهارم از جدول depositor در خروجی نمایش داده نمی‌شود.

در ادامه برای سطر پنجم از جدول depositor با توجه به شرط اتصال T.cname = R.cname and R.acc# = account.acc# and bname = 'perryidge' داریم:

T.cname	acc#	...
alavi	1213	
alavi	1214	
hassani	1215	
hassani	1216	
asgari	1217	

جدول مشتری depositor T

R.cname	R.acc#	account.acc#	bname
alavi	1213	1213	Perryidge
hassani	1215	1215	Perryidge
hassani	1216	1216	Perryidge

جلوی UNIQUE فقط یک سطر وجود ندارد یعنی هیچ سطري وجود ندارد، بنابراین شرط where در پشت FALSE می‌گردد. بنابراین سطر پنجم از جدول depositor در خروجی نمایش داده نمی‌شود.

و در نهایت دستور select اعمال می‌گردد، بنابراین خروجی نهایی پرس و جو به صورت زیر است:

T.cname
alavi
alavi

توجه: بهتر بود طراح محترم جهت حذف سطرهای تکراری در پشت ستون T.cname از دستور Distinct استفاده نمود، به صورت زیر:

```
select Distinct T.cname
from depositor as T
where UNIQUE (...)
```

که در اینصورت خروجی نهایی پرس و جو به صورت زیر می‌بود:

T.cname
alavi

بنابراین مطابق پرس و جوی مطرح شده در گزینه‌ی چهارم، «نام مشتریانی استخراج می‌شود که در شعبه Perryridge فقط و فقط یک حساب بانکی دارند، نه کمتر و نه بیشتر» که مطابق پرس و جوی مطرح شده در گزینه‌های اول و دوم است.

توجه: اگر در پرس و جوی مطرح شده در گزینه چهارم به جای عملگر UNIQUE از عملگر Exists استفاده نمود، به صورت زیر:

```
select Distinct T.cname
from depositor as T
where Exists (select R.cname
               from account, depositor as R
               where T.cname = R.cname and
                     R.acc#=account.acc# and bname='perryidge')
```

آنگاه حاصل پرس و جو همانند گزینه‌ی سوم می‌شد، خروجی نهایی پرس و جو به صورت زیر است:

cname
alavi
hassani

یعنی، «نام مشتریانی استخراج می‌شود که در شعبه Perryidge حداقل یک حساب بانکی دارند.»

۳۹ - گزینه (۱) صحیح است.

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

```
SELECT distinct S.customer-name
FROM depositor as S
WHERE NOT EXISTS ((SELECT branch-name
                   FROM Branch
```

```

WHERE branch-city='Brooklyn')
EXCEPT
(SELECT R.branch-name
FROM depositor as T, account as R
WHERE T.account-number = R.account-number
And S.customer-name=T.customer-name))

```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«نام مشتریانی که، وجود نداشته باشد (**NOT EXISTS**) شعبه‌ای از شهر Brooklyn که، حساب در آن باز نشده باشد (**EXCEPT**). یعنی نام مشتریانی که در همه شعب شهر Brooklyn حساب باز کرده‌اند.»

توجه: عملگر تقسیم در جبرابطه‌ای به فرم فوق در SQL قابل پیاده‌سازی است.

راه حل دوم: در زیر پرس و جوی مقابله عملگر **not exists** ابتدا تفاضل اسامی کل شعب شهر Brooklyn و شعبی که مشتری در آنها حساب دارد، استخراج می‌گردد. سپس در پرانتز بیرونی نام مشتریانی به عنوان خروجی داده می‌شود که نتیجه قبل در مورد آنها تهی باشد، یعنی نام مشتریانی که شعبه‌ای از Brookln وجود ندارد که آنها در آن شعبه حساب ندارند. به این ترتیب به عنوان خروجی نهایی پرس و جوی داده شده نام مشتریانی داده می‌شود که در تمامی شعب شهر Brooklyn حساب دارند.

مثال: جداول **Branch**, **Depositor**, **Account**, **Branch** و **Account** را در نظر بگیرید:

branch-name	branch-city	assets	account-number	branch-name	balance
A	Brooklyn	700	11	A	10
B	Brooklyn	500	11	B	20
C	Brooklyn	800	11	C	30
D	Tehran	900	11	D	40

شعب Branch	12	A	100
	12	B	90
	12	C	80
	13	A	50
	14	D	60

موجودی Account R

customer-name	account-number
Alavi	11
Hassani	12
Taghavei	13
Javadi	14

مشتریان Depositor S و T

با توجه به پرس و جوی مطرح شده، به ازای حرکت در هر سطر از جدول S تفاضل حاصل از اجرای کامل دو طرف عملگر **except** محاسبه می‌گردد، اگر حاصل تفاضل تهی گردد. شرط where در پشت

TRUE می‌گردد، و سطر مورد نظر از جدول S، در خروجی نمایش داده می‌شود، البته با توجه به ستون select در دستور S.Customer-name، نام مشتری در خروجی نمایش داده می‌شود.
توجه: دستور S.customer-name=T.customer-name نقطه اتصال جدول حاصل از ضرب دکارتی دو جدول T و R به جدول s در select خارجی به ازای حرکت در هر سطر از جدول s است و دستور T.account-number=R.account-number شرط و نوع اتصال ضرب دکارتی دو جدول T و R را مشخص می‌کند.

۴۰- گزینه (۲) صحیح است.

هر دو عبارت، دو ستون a و b را در خروجی می‌آورند.
عبارت Q1 سطراهایی از R که مقادیر دو ستون (a,b) آنها غیرتکراری است را انتخاب کرده و فقط دو ستون a و b را در خروجی نشان می‌دهد.
عبارت Q2 نیز ابتدا رابطه‌ی R را بر حسب مقادیر a,b گروه‌بندی می‌کند و از هر گروه فقط مقادیر a و b (که مطمئناً غیرتکراری هستند) را در خروجی نشان می‌دهد.
پس دو عبارت Q1 و Q2 پاسخ‌های یکسانی تولید می‌کنند.
مثال: جدول R(a,b,c) را با مقادیر زیر در نظر بگیرید:

a	b	c
1	2	3
1	2	4
3	4	5
3	4	6

Q1 : select distinct a,b
from R \rightarrow

a	b
1	2
3	4

Q2 : select a,b
from R
group by a,b

جدول R بر اساس (a,b) به صورت زیر گروه‌بندی می‌شود:

(1,2)	(3,4)
3	5
4	6
گروه اول	گروه دوم

توجه: در گروه‌بندی، سرگروه‌ها، راهی خروجی می‌شوند، بنابراین داریم:

a	b
1	2
3	4

همان‌طور که مشاهده می‌شود Q1 و Q2 پاسخ‌های یکسان تولید می‌کنند.

۴۱- گزینه (۳) صحیح است.

دستور select سوم، شماره قطعاتی را می‌دهد که رنگشان قرمز است.

دستور select دوم، شماره‌ی تولید کنندگانی را می‌دهد که قطعه‌ای از مجموعه‌ی فوق (خروجی دستور select سوم) را تولید کنند، یعنی شماره‌ی تولید کنندگانی را می‌دهد که قطعه‌ای به رنگ قرمز تولید می‌کنند. بالاخره دستور select اول، نام تولید کنندگانی را می‌دهد که شماره‌ی آنها در خروجی دستور select دوم باشد، یعنی نام تولید کنندگانی که قطعه‌ای (حداقل یکی) به رنگ قرمز تولید می‌کنند.

۴۲- گزینه (۲) صحیح است.

در پرس و جوی داده شده، به ازای هر سطر از جدول S، سطرهایی از جدول SP که شماره‌ی تولیدکننده (S#) آنها با شماره‌ی تولیدکننده‌ی جدول S برابر باشد و قطعه‌ای با شماره‌ی 'P2' تولید کند را انتخاب می‌کند، اگر حداقل یک سطر از SP انتخاب شود، سطر متناظر از S در خروجی ظاهر می‌شود. در نهایت پرس و جوی داده شده، نام تولیدکنندگانی را می‌دهد که قطعه‌ی P2 را تولید می‌کنند. به جداول زیر دقت کنید:

جدول S			جدول SP			جدول P	
s#	Sname	...	s#	p#	QTY	p#	pname
s1	sn1		s1	p1	10	p1	
s2	sn1		s1	p2	20	p2	
s3	sn3		s2	p2	30		
			s3	p2	40		

```
select S.sname
from s
where exists (select *
               from sp
               where S.s#=SP.s# AND SP.p#='p2')
```

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول sp بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول sp قرار دارد که شرط موجود در select داخلی را برقرار کند، اگر وجود داشته باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

پرس و جوی فوق در واقع نام تولید کنندگانی را نمایش می‌دهد که برای آنها تولید قطعه 'p2' ثبت شده است (تولید قطعه 'p2' برای آنها وجود دارد) به بیان دیگر پرس و جوی فوق نام تولید کنندگانی را می‌دهد که قطعه 'p2' را تولید کرده‌اند، در نهایت select بیرونی، نام تولید کنندگانی را می‌دهد که قطعه 'p2' را تولید کرده‌اند، البته چون به دلیل کلید نبودن Sname ممکن است اسمی برخی تولید کنندگان شیوه هم باشد، امکان دارد، اسمی تکراری در خروجی این پرس و جو مشاهده شود. بنابراین خروجی Sname به صورت زیر خواهد بود.

S.sname

sn1

sn1

sn3

توجه: SQL به طور پیش فرض سطرهای تکراری را حذف نمی‌کند، مگر از دستور Distinct استفاده شود.

بنابراین گزینه چهارم نادرست است. زیرا عبارت «بدون تکرار» را لحاظ کرده است.

۴- گزینه (۴) صحیح است.

هر سه دستور sql نوشته شده، یک کار را انجام می‌دهند.

دستور select اول، مشخصات قطعاتی را می‌دهد که spin آنها برابر 1000 یا 2000 باشد.

دستور select دوم، مشخصات قطعاتی را می‌دهد که spin آنها عضو مجموعه {2000 و 1000} باشد.

دستور select سوم، اجتماع دو مجموعه را می‌دهد: مجموعه قطعاتی که spin آنها برابر 1000 است و

مجموعه قطعاتی که spin آنها برابر 2000 است.

۵- گزینه (۳) صحیح است.

با توجه به شرط $T1.B = T2.B$ ، باید بینیم هر سطر از جدول T1 با چند سطر از جدول T2 تطابق دارد.

(منظور از «تطابق» برابری مقدار فیلد B در دو جدول است). سطر اول از T1 با سطراهای اول و دوم از

T2 تطابق دارد، پس به ازای آنها، دو سطر در خروجی تولید می‌شود. سطر دوم از T1 با سطراهای سوم،

چهارم و پنجم از T2 تطابق دارد. پس به ازای آنها سه سطر در خروجی تولید می‌شود. سطر سوم از T1 با هیچ سطري از T2 تطابق ندارد و به ازای آن هیچ سطري در خروجی تولید نمی‌شود. پس تعداد کل

سطرهای خروجی برابر است با: $5 + 3 = 8$

T1		T2	
A	B	B	C
a1	b1	b1	c1
a2	b2	b1	c2
a3	b3	b2	c2
		b2	c3
		b2	c4

خروجی دستور select نوشته شده به صورت زیر است:

B
b1
b1
b2
b2
b2

توجه کنید که خروجی‌های تکراری ایجاد می‌شوند چون جلوی select از distinct استفاده نشده است.

توجه: SQL به طور پیش فرض سطرهای تکراری را حذف نمی‌کند، مگر از دستور distinct استفاده شود.

۶- گزینه (۱) صحیح است.

در پرس‌وجوی نوشته شده از جدول enroll (به معنی «ثبت‌نام») استفاده شده است که حداقل شامل سه

فیلد crsno، cname و pname است. این فیلدها به ترتیب، شماره‌ی درس، نام استاد آن درس و نام

دانشجویی که در آن درس ثبت‌نام کرده است را نشان می‌دهد.

برای توصیف خروجی پرس‌وجوی داده شده بهتر است ابتدا خروجی دستور select داخلی را تعیین کنیم.

به راحتی می‌توان دید که دستور select داخلی، مجموعه‌ی شماره‌ی درس‌هایی که دانشجویی با نام کاظمی در آن‌ها ثبت‌نام کرده است را به ما می‌دهد. در نتیجه، select بیرونی، نام اساتیدی را می‌دهد که درسی را تدریس کنند که شماره‌ی آن درس متعلق به مجموعه‌ی فوق باشد. یعنی حداقل یکی از درس‌هایی را که کاظمی در آن‌ها ثبت‌نام کرده است، تدریس کرده‌اند. بنابراین کل عبارت پرس‌وجو، نام اساتیدی (غیرتکراری) را می‌دهد که حداقل یکی از درس‌هایی را تدریس می‌کنند که کاظمی در آن‌ها ثبت‌نام کرده است.

مثال: به جدول زیر دقت کنید:

Sname	pname	Crsno
✓ کاظمی	اکبری	12
✓ کاظمی	حسنی	13
✓ کاظمی	حسنی	14
* جوادی	حسنی	15
* عسگری	سجادی	16
* تقوی	اکبری	17

جدول *enroll*

براساس پرس‌وجوی مطرح شده داریم:

```
select distinct pname
```

```
from enroll
where crsno in
  (select crsno
   from enroll
   where sname = 'kazemi')
12,13,14
```

در پرس‌وجوی فوق به ازای حرکت در هر سطر از جدول *enroll* مجموعه جلوی *in* بررسی می‌گردد. آیا *crsno* داخل این مجموعه قرار دارد یا خیر. اگر قرار داشت سطر مورد نظر از جدول *enroll* در خروجی نمایش داده می‌شود.

در نهایت خروجی پرس‌وجوی فوق به صورت زیر است:

Pname
اکبری
حسنی

توجه: سطرهای تکراری به دلیل استفاده از دستور *distinct* حذف گردید.
گزینه دوم نادرست است. زیرا استادانی مدنظر است که در حال حاضر به کاظمی تدریس می‌کنند و نه استادان قبل‌تر.

توجه: جدول *enroll*، جدول ثبت نام است.

۴۶ - گزینه (۳) صحیح است.

عبارت حساب رابطه‌ای داده شده، دو سطر (b_1, b_2) و (a, c) از رابطه‌ی r را پیدا می‌کند که در آن‌ها $b_1 > b_2$ باشد و سطر (a, c) در رابطه‌ی S باشد و به ازای چنین سطرهایی مقدار a را برابر می‌گرداند. به بیان دیگر عبارت حساب رابطه‌ای داده شده مؤلفه اول a سطرهایی از رابطه S را نمایش می‌دهد که دو مؤلفه

آنها $s \in \{a, c\}$ مولفه‌های اول دو سطر در رابطه r را تشکیل می‌دهند و رابطه‌ی $b_1 > b_2$ میان مولفه‌های دوم سطرهای مورد نظر در رابطه r برقرار است.

A	C	A	B
a	c	a	b_1
c		c	b_2

جدول r جدول s

گزینه‌های اول و دوم و چهارم نیز چنین کاری می‌کنند. در گزینه‌ی اول، قسمت زیر از دستورات را در نظر بگیرید:

from r as $r1$, r as $r2$

where $r1.B > r2.B$

عبارت from، هر سطر از r را با سطر دیگری از آن در نظر می‌گیرد و عبارت where سطرهایی را انتخاب می‌کند که فیلهای B آنها برابر نباشند (همان شرط $b_1 > b_2$). سپس وجود سطر (a, c) در جدول S در دو خط زیر بررسی می‌شوند:

where $(A, C) \in \{$

select $r1.A, r2.A$

در گزینه‌ی دوم، عبارت from r as $r1$ و r as $r2$ هر دو سطري از r را با هم در نظر می‌گیرد و سپس شرط جلوی where دوم، سطرهایی که شرط $r2.B > r1.B$ (یا همان $b_1 > b_2$) را داشته باشند را انتخاب می‌کند.

در گزینه‌ی چهارم، شرط $r1.B > r2.B$ معادل شرط $b_1 > b_2$ است و ادامه‌ی شرط where (بعد از AND) معادل با وجود سطر (a, c) در S است.

با توجه دقیق به گزینه‌های اول و چهارم می‌توان دریافت که این دو گزینه دو سوی مختلف یک مطلب یکسان را بیان می‌کنند و با هم معادل هستند. گزینه دوم نیز پیاده‌سازی شده عملگر in را با استفاده از عملگر exists نشان می‌دهد و با گزینه اول و چهارم معادل است.

چنانکه ملاحظه می‌شود در عبارت حساب رابطه‌ای داده شده هیچ قیدی در مورد یکسان بودن مقدار فیله A در سطرهایی از دو جدول r و s ذکر نشده است. اما در گزینه‌ی سوم، با انجام NATRUAL INNER JOIN سطرهایی از r و s که فیلهای A در آنها مقدار یکسانی دارند انتخاب می‌شوند که اشتباه است. در واقع گزینه سوم با عبارت حساب رابطه‌ای داده شده معادل نیست.

مثال: دو جدول r و s با مقادیر زیر را در نظر بگیرید:

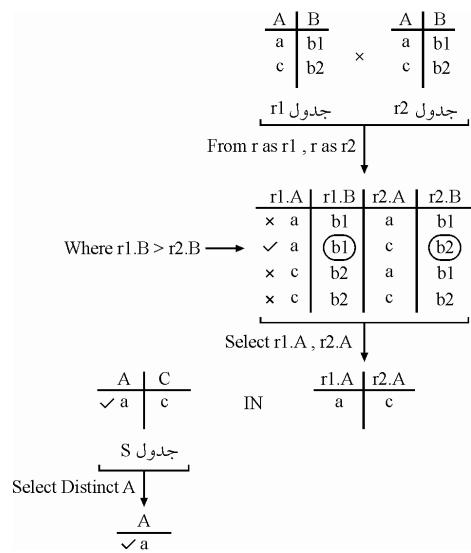
A	C	A	B
a	c	a	b_1
c		c	b_2

جدول r جدول s

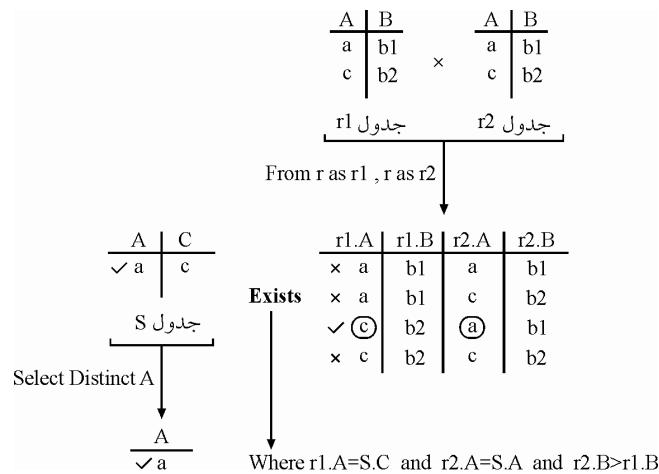
توجه: در جدول r مقدار b_1 بزرگ‌تر از مقدار b_2 در نظر گرفته شده است.

گزینه اول: در این گزینه به ازای حرکت در هر سطر از جدول s ، یک بار به طور کامل select داخلی اجرا می‌گردد. سپس عضویت مقادیر (A, C) از جدول s در محتوای حاصل از ضرب دکارتی مطرح شده با

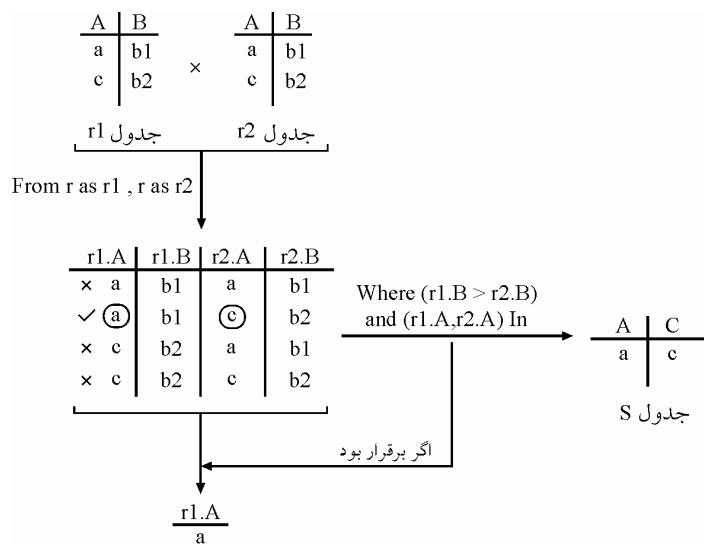
شرایط ویژه‌اش بررسی می‌گردد. اگر عضویت برقرار بود، سطر مورد نظر از جدول s در خروجی نمایش داده می‌شود.



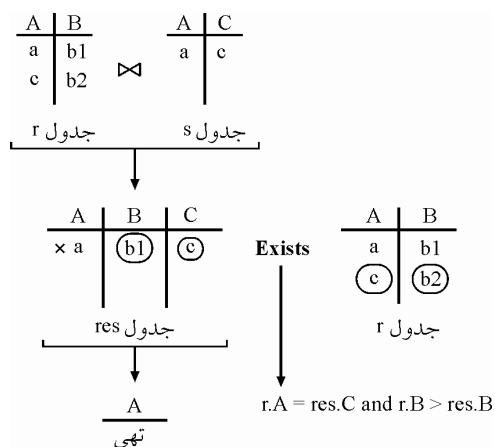
گزینه دوم: این گزینه پیاده‌سازی شده عملگر in ، با استفاده از عملگر $exists$ است.
در این گزینه به ازای حرکت در هر سطر از جدول s ، یکبار به طور کامل در $select$ داخلی بررسی می‌گردد که آیا حداقل یک سطر وجود دارد که شرایط موجود در $select$ داخلی را برآورده سازد، اگر بود، سطر مورد نظر از جدول s در خروجی نمایش داده می‌شود.



توجه: شیوه‌ی جستجو در دستور $where$ با گزینه اول متفاوت است، اما نتیجه یکسان خواهد بود.
گزینه چهارم: این گزینه طرف مقابل گزینه اول اما با پاسخ یکسان را پیاده‌سازی می‌کند.



گزینه سوم: در این گزینه به دلیل استفاده از عملگر الحقیقی طبیعی، سطرهایی از r و s که در ستون مشترک A هم مقدار هستند، جهت نمایش در خروجی انتخاب می‌شوند. در حالی که در گزینه‌های اول، دوم و چهارم سطرهایی از جدول s انتخاب می‌شوند که مقادیر ستون A و C از جدول s برابر مقادیر ستون A در دو سطر از جدول r باشد که $b1 > b2$ برای ستون B در جدول r را نیز رعایت کند. بنابراین پرس‌وجوی گزینه سوم عملی کاملاً متفاوت با پرس‌وجوهای سایر گزینه‌ها انجام می‌دهد.



۴۷- گزینه (۲) صحیح است.

دستور Update برای تغییر دادن و اصلاح کردن فیلد های جداول می باشد. برای مثال، اگر اسم جدول مسافران، Passenger و اسم فیلد متناظر با نام مسافر، Pname باشد، برای تغییر تمام نام های «اکبرزاده» به update Passenger «اکبرزاده» می نویسیم:

set Pname = «اکبرزاده»

where Pname = «اکبرزاده»

۴۸- گزینه (۱) صحیح است.

باید تعداد کد درس‌ها (#c#) ها در ترم 892 (یعنی $term = 892$) را بشماریم. اما توجه کنید که جدول Enroll جدول ثبت نام هاست (چه دانشجویی چه درسی را در چه ترمی با چه استادی گرفته است) نه جدول دروس ارائه شده. پس در جدول Enroll ممکن است یک کد درس (#c#) چندین بار تکرار شده باشد (چون یک درس توسط دانشجویان مختلفی گرفته می‌شود و به ازای هر کدام یک سطر در جدول Enroll وجود دارد). پس باید از distinct استفاده کنیم تا موارد تکراری را نشمرد یا به عبارت دیگر، هر مقدار از #c را فقط یک بار بشمرد.

توجه کنید که کلمه distinct، همان‌طور که در گزینه اول نوشته شده است، باید قبل از فیلد مربوطه (#c#) بیابد و نمی‌تواند مانند گزینه دوم نوشته شود. گزینه دوم خطای نحوی دارد.

گزینه سوم اشتباه است چون از کلمه distinct استفاده نکرده است. در واقع گزینه سوم موارد تکراری را هم می‌شمارد. گزینه چهارم نیز اشتباه است چون از کلمه count استفاده نکرده و در نتیجه تعداد درس‌ها را به ما نمی‌دهد بلکه کد درس‌ها را می‌دهد.

۴۹- گزینه (۴) صحیح است.

هر سه عبارت پرس‌وچوی داده شده یک کار را انجام می‌دهند:
 عبارت I، مشخصات تولید‌کنندگانی را می‌دهد که شهر آن‌ها برابر Hamadan یا Tabriz باشد.
 عبارت II، مشخصات تولید‌کنندگانی را می‌دهد که شهر آن‌ها در مجموعه {Hamadan, 'Tabriz'} وجود داشته باشد.

عبارت III با استفاده از عملگر union، اجتماع مجموعه «تولید‌کنندگان همدان» و مجموعه «تولید‌کنندگان تبریز» را به ما می‌دهد.

۵۰- گزینه (۱) یا (۳) صحیح است.

صورت سؤال دو اشکال دارد:

۱) پیوندی به نام 'join' وجود ندارد. منظور طراح سؤال می‌تواند پیوند 'Left outer join' یا 'inner join' باشد.

۲) پیوندهای inner join و outer join همواره با عبارت «شرط on» همراه‌اند که عبارت «شرط» نشان می‌دهد چه سطرهایی از دو جدول باید با هم پیوند زده شوند.

بنابراین، دستور SQL داده شده باید به یکی از دو شکل زیر می‌بود:

select T1.B

from T1 inner join T2 on T1.B=T2.B

یا

select T1.B

from T1 Left outer join T2 on T1.B=T2.B

دستور اول، هر یک از سطرهای T1 را با تک سطرهای T2 مقایسه می‌کند، هر جا شرط T1.B=T2.B برقرار بود مقدار T1.B را در خروجی می‌نویسد. توجه کنید که چون از distinct استفاده نشده پس سطرهای تکراری در خروجی ممکن است چاپ شوند. بنابراین، سطر اول T1 با سطرهای اول و دوم T2 و سطر دوم از T1 با سطرهای سوم، چهارم و پنجم از T2 تطابق دارند و بنابراین پنج سطر زیر در خروجی

ظاهر می‌شود. (سطر سوم از T1 با هیچ سطری از T2 تطابق ندارد):

T1		T2		خروجی
A	B	B	C	T1.B
a ₁	b ₁	b ₁	c ₁	b ₁
a ₂	b ₂	b ₁	c ₂	b ₁
a ₃	b ₃	b ₂	c ₂	b ₂
		b ₂	c ₃	b ₂
		b ₂	c ₄	b ₂

در دستور select دوم، پیوند Left outer join باعث می‌شود علاوه بر خروجی‌هایی که توسط inner join تولید می‌شوند، سطرهایی از T1 (جدول سمت چپ پیوند) که با هیچ سطری از T2 تطابق ندارند نیز در خروجی پیوند ظاهر شوند (فیلدهای متناظر با T2 در خروجی برابر NULL خواهد بود). سپس فقط ستون T1.B در خروجی می‌آید. پس سطر سوم از T1 نیز در خروجی پیوند ظاهر می‌شود و فیلد T1.B از آن نیز در خروجی می‌آید:

خروچی
T1.B
b ₁
b ₁
b ₂
b ₂
b ₂
b ₃

۵۱- گزینه (۲) صحیح است.

در پرس‌وجوی نوشته شده از جدول enroll (به معنی «ثبت نام») استفاده شده است که حداقل شامل سه فیلد crsno، cname و sname است. این فیلدها به ترتیب، شماره‌ی درس، نام استاد آن درس و نام دانشجویی که در آن درس ثبت‌نام کرده است را نشان می‌دهد.

برای توصیف خروجی پرس‌وجوی داده شده بهتر است ابتدا خروجی دستور select داخلی را تعیین کنیم. به راحتی می‌توان دید که دستور select داخلی، مجموعه‌ی شماره‌ی درس‌هایی درس‌هایی با نام ذاکری در آن‌ها ثبت‌نام کرده است را به ما می‌دهد. در نتیجه، select بیرونی، نام استادی را می‌دهد که درسی را تدریس کنند که شماره‌ی آن درس متعلق به مجموعه‌ی فوق باشد. یعنی حداقل یکی از درس‌هایی را که ذاکری در آن‌ها ثبت‌نام کرده است، تدریس کرده‌اند. بنابراین کل عبارت پرس‌وجو، نام استادی (غیر تکراری) را می‌دهد که حداقل یکی از درس‌هایی را تدریس می‌کنند که ذاکری در آن‌ها ثبت‌نام کرده است.

مثال: به جدول زیر دقت کنید:

Sname	Pname	Crsno
✓ ذاکری	اکبری	12
✓ ذاکری	حسنی	13
✓ ذاکری	حسنی	14
* جوادی	حسنی	15
* عسگری	سجادی	16
* تقی	اکبری	17

جدول enroll

براساس پرس‌وجوی مطرح شده داریم:

```
select distinct pname
from enroll
where crsno in
      (select crsno
       from enroll
       where sname = 'kazemi')
12,13,14
```

در پرس‌وجوی فوق به ازای حرکت در هر سطر از جدول `enroll` مجموعه جلوی `in` بررسی می‌گردد، آیا `crsno` داخل این مجموعه قرار دارد یا خیر. اگر قرار داشت سطر مورد نظر از جدول `enroll` در خروجی نمایش داده می‌شود.

در نهایت خروجی پرس‌وجوی فوق به صورت زیر است:

Pname
اکبری
حسنی

توجه: سطرهای تکراری به دلیل استفاده از دستور `distinct` حذف گردید.

توجه: جدول `enroll`، جدول ثبت‌نام است.

۵۲- گزینه (۳) صحیح است.

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

«نام همه مشتریانی که در همه شعب Brooklyn حساب دارند.»

توجه: کلمه همه اول، در پرس و جوی فوق زائد است و بود و نبودش تفاوتی در خروجی پرس و جو ایجاد نمی‌کند. چون خواه ناخواه همه مشتریان حائز شرط در خروجی قرار می‌گیرند و ذکر کلمه همه در ابتدای مشتریان بی معنی است. بنابراین پرس و جوی مطرح شده در صورت سوال را به فرم زیر نیز می‌توان نوشت:

«نام مشتریانی که در همه شعب Brooklyn حساب دارند.»

توجه: عملگر تقسیم در جبر رابطه‌ای در SQL قابل پیاده‌سازی است. عملگر تقسیم در جبر رابطه‌ای زمانی مورد استفاده قرار می‌گیرد که پرس و جو بخواهد همه حالت‌های یک اتفاق را بررسی کند. مانند نام مشتریانی که در همه شعب شهر Brooklyn حساب باز کرده‌اند. که بازشدن حساب توسط مشتریان، اتفاق و مقسوم است و همه شعب شهر Brooklyn، حالت‌ها و مقسوم علیه است.

توجه: عملگر تقسیم در جبر رابطه‌ای به فرم زیر در SQL قابل پیاده‌سازی است.

```
SELECT distinct S.customer-name
FROM depositor as S
WHERE NOT EXISTS ((SELECT branch-name
                     FROM Branch
                     WHERE branch-city='Brooklyn')
EXCEPT
(SELECT R.branch-name
FROM depositor as T, account as R
WHERE T.account-number = R.account-number
And S.customer-name=T.customer-name))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:
 «نام مشتریانی که، وجود نداشته باشد (NOT EXISTS) شعبه‌ای از شهر Brooklyn که، حساب در آن باز نشده باشد (EXCEPT). یعنی نام مشتریانی که در همه شعب شهر Brooklyn حساب باز کرده‌اند.»

راه حل دوم: برای به دست آوردن «نام مشتریانی که در همه شعب Brooklyn حساب دارند» باید برای هر مشتری (با نام S.customer-name) از جدول depositor نام تمام شعبه‌هایی که آن مشتری در آن ها حساب دارد را از روی جداول depositor و account به دست آورد (این مجموعه را A می‌نامیم). همچنین باید مجموعه نام تمام شعب شهر Brooklyn را از جدول branch به دست آورد (این مجموعه را B می‌نامیم). اگر A با B برابر بود (یا به عبارت دیگر، اگر B-A هیچ رکوردهای نداشت) نام مشتری مربوطه باید در خروجی بیاید. دستورات مربوط به به دست آوردن مجموعه‌های A و B در زیر نشان داده شده‌اند. بنابراین باید به جای YY دستور except قرار داده شود تا B-A محسابه شود. همچنین برای بررسی این که B-A هیچ رکوردهای نداشته باشد باید به جای XX دستور not exists قرار داده شود.

```
select distinct S.customer-name
from depositor as s
```

مجموعه‌ی B

```
where XX
((select branch-name
  from branch
  where branch-city = 'Brooklyn'))
```

مجموعه‌ی A

```
(select R.branch-name
  from depositor as T, account as R
  where T.account-number = R.account-number and
  S.customer-name = T.customer-name)
```

)

مثال: جداول Depositor و Account، با مقادیر زیر را در نظر بگیرید:

branch-name	branch-city	assets	account-number	branch-name	balance
A	Brokklyn	700	11	A	10
B	Brooklyn	500	11	B	20
C	Brooklyn	800	11	C	30
D	Tehran	900	11	D	40
شعب Branch			12	A	100
دارایی شعبه است.			12	B	90
: assets			12	C	80
			13	A	50
			14	D	60

موجودی Account R

customer-name	account-number
Alavi	11
Hassani	12
Taghavi	13
Javadi	14

مشتریان Depositor S و T

با توجه به پرس‌وجوی مطرح شده، به ازای حرکت در هر سطر از جدول S تفاضل حاصل از اجرای کامل دو طرف عملگر except محاسبه می‌گردد، اگر حاصل تفاضل تهی گردد. شرط where در پشت not exists می‌گردد، و سطر مورد نظر از جدول S، در خروجی نمایش داده می‌شود، البته با توجه به ستون TRUE در دستور select، نام مشتری در خروجی نمایش داده می‌شود.

توجه: دستور S.customer-name=T.customer-name نقطه اتصال جدول حاصل از ضرب دکارتی دو جدول T و R به جدول S در select خارجی به ازای حرکت در هر سطر از جدول s است و دستور شرط و نوع اتصال ضرب دکارتی دو جدول T و R را مشخص می‌کند.

۵۳- گزینه (۱) صحیح است.

در این سؤال فرض شده است که اسمی افراد منحصر به فرد (و در نتیجه کلید) هستند. برای پاسخ به این سؤال، باید از بین تمامی کارمندان، آن‌هایی که اسمشان در جدول Mobtel نیست کنار گذاشته شوند. این دقیقاً کاری است که گزینه اول انجام می‌دهد.

پرانتز داخلی در گزینه اول، نام کارمندانی را می‌دهد که تلفن موبایل آن اپراتور را دارند. با استفاده از not in در پرانتز بیرونی، نام کارمندانی به دست می‌آید که تلفن موبایل آن اپراتور را ندارند. گزینه چهارم به دلیل استفاده از جدول fixtel به تلفن ثابت مربوط می‌شود، بنابراین نادرست است.

گزینه سوم لیست کارمندانی را می‌دهد که تلفن موبایل آن شرکت را دارند که بر عکس خواسته سؤال است گزینه دوم، سطرهای غیر معتبر حاصل از ضرب دکارتی دو جدول را می‌دهد.

۵۴- گزینه (۴) صحیح است.

دستور select داخلي، مجموعه شناسه‌های پدران را به ما می‌دهد (يعني مردانی که فرزند دارند). سپس دستور select بیرونی، تاریخ تولد شخصی را می‌دهد که نامش برابر X باشد و شناسه‌اش در مجموعه فوق باشد (يعني آن شخص پدر باشد، و این يعني این که شخص فرزندی داشته باشد)

۵۵- گزینه (۲) صحیح است.

ابتدا قسمت زیر از پرس‌وجو را در نظر بگیرید:

```
SELECT SUM(W.salary) AS s
FROM Employee E, WorksIn W
WHERE E.eid=W.eid
GROUP BY W.did
```

این عبارت، سطرهای جدول WorksIn را بر حسب مقدار did (شماره بخش) گروه‌بندی می‌کند (يعني برای هر بخش یک گروه در نظر می‌گیرد) و مجموع حقوق کارمندان آن بخش را بر می‌گرداند. با توجه به مقادیر جدول WorksIn، دو گروه تشکیل می‌شود (یکی برای d1 و دیگری برای d3) و مجموع salary هر گروه

جداگانه محاسبه می شود (8 برای d1 و 4 برای d3). پس خروجی دستور select فوق، جدولی شبیه جدول زیر است:

S
8
4

حال قسمت زیر از پرس و جو را در نظر بگیرید:

```
SELECT 0 AS s
FROM Department D
WHERE NOT EXISTS (SELECT *
                   FROM WorksIn W1
                   WHERE W1.did=D.did)
```

این قسمت از پرس و جو، به ازای هر بخشی که هیچ کارمندی در آن کار نکند، یک عدد صفر در خروجی قرار می دهد.

با توجه به این که سه بخش d1، d2 و d3 در جدول Department تعریف شده است، اما در جدول WorksIn برای بخش d2 هیچ کارمندی درج نشده است پس این قسمت از پرس و جو یک جدول تک ستونه با یک سطر حاوی مقدار صفر به صورت زیر خواهد بود.

S
0

دستور UNION ALL اجتماع دو جدول فوق را به ما می دهد و در نهایت، دستور (R.s) میانگین اعداد حاصل از اجتماع دو جدول فوق را به ما می دهد که برابر است با:

$$\frac{8+4+0}{3} = 4$$

توجه: به تفاوت union و All union دقت کنید، در union دو جدول مقادیر تکراری حذف می گردد و در All union دو جدول مقادیر تکراری حذف نمی شوند.

۵۶- گزینه (۴) صحیح است.

الحق خارجی چپ (LEFT OUTER JOIN)، همه افرادی که پدری برای آنها ثبت شده باشد یا نشده باشد را تولید می کند. زیرا در الحق خارجی چپ، تمام سطرهای جدول سمت چپ در خروجی ظاهر می شوند، حتی سطرهایی که سطر متناظری در جدول سمت راست ندارند. بنابراین با انجام این پیوند روی C و F تمامی افراد در خروجی می آیند، حتی آنها که سطر متناظری در جدول پدران ندارند.

در ادامه، الحق داخلی (INNER JOIN)، افرادی را انتخاب می کند که مادری برای آنها ثبت شده باشد. بنابراین خروجی نهایی این پرس و جو به صورت زیر است:

همه افرادی که مادری برای آنها ثبت شده است، مستقل از اینکه پدری برای آنها ثبت شده باشد یا نشده باشد.

سازمان سنجش آموزش کشور، گزینه چهارم را به عنوان پاسخ در کلید نهایی اعلام نموده است. عبارت «همه افراد» در گزینه چهارم، بدین معنی است که افرادی که حتی مادر برای آنها ثبت نشده است نیز می توانند در خروجی قرار گیرد در حالی که با توجه به شرط الحق داخلی، افرادی انتخاب می شوند که

مادری برای آنها ثبت شده باشد.

بنابراین بهتر بود گرینه چهارم به صورت زیر اصلاح می‌گردید:

«همه افرادی که مادر برای آنها ثبت شده است، حتی آنها بی که پدر برای آنها ثبت نشده است.»

برای درک بهتر، جدول Person با مقادیر زیر را در نظر بگیرید:

Person ID	Name	MotherID	FatherID
1	Sara	NULL	NULL
2	Saeed	NULL	NULL
12	Hamed	1	2
1122	Reza	1	NULL

جدول Person

پرس‌وجوی مطرح شده در صورت سوال به صورت زیر است:

```
SELECT C.Name AS Name, F.Name AS Father, M.Name AS Mother
FROM person AS C LEFT OUTER JOIN person AS F ON C.FatherID=F.PersonID
INNER JOIN person AS M ON C.MotherID=M.PersonID
```

توجه: وجود دستور AS در جلوی دستور SELECT سبب می‌شود ستون‌های C.Name و F.Name به ترتیب به صورت C.Name و Father.Name و Mother در خروجی نهایی ظاهر گردد.

توجه: وجود دستور AS در جلوی دستور FROM سبب می‌شود، سه کمی مجزا به نام‌های C، F و M از روی جدول Person ایجاد گردد.

عملگر الحق خارجی چپ در جبر رابطه‌ای

این عملگر، مانند الحق 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 الحق 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 الحق خارجی چپ در جبر رابطه‌ای و SQL دقت کنید.

خروجی الحق خارجی چپ مستقل از بخش‌های دیگر پرس‌وجو به صورت زیر است:

Select *

```
FROM Person AS C LEFT OUTER JOIN Person AS F ON C.FatherID = F.PersonID
```

C.PersonID	C.Name	C.MotherID	C.FatherID	F.PersonID	F.Name	F.MotherID	F.FatherID
1	sara	NULL	NULL	NULL	NULL	NULL	NULL
2	saeed	NULL	NULL	NULL	NULL	NULL	NULL
12	hamed	1	2	2	saeed	NULL	NULL
1122	reza	1	NULL	NULL	NULL	NULL	NULL

توجه: مشاهده می شود که همه افراد، چه برای آنها پدر ثبت شده باشد و چه نشده باشد، بنا به خاصیت الحاق خارجی چپ، در خروجی ظاهر شده اند.
الحاق داخلی خروجی فوق با جدول M، مستقل از ستون های جلوی دستور SELECT به صورت زیر است:

C				F				M			
PersonID	Name	MotherID	FatherID	PersonID	Name	MotherID	FatherID	PersonID	Name	MotherID	FatherID
12	hamed	1	2	2	saeed	NULL	NULL	1	sara	NULL	NULL
1122	reza	1	NULL	NULL	NULL	NULL	NULL	1	sara	NULL	NULL

در انتها، خروجی نهایی، با دخالت دستور SELECT به صورت زیر خواهد بود:

Name	Father	Mother
hamed	saeed	sara
reza	NULL	sara

مشاهده می شود که، «همه افرادی که مادری برای آنها ثبت شده است، مستقل از اینکه پدری برای آنها ثبت شده باشد یا نشده باشد» در خروجی نشان داده شده است.

توجه: اگر هر دو پیوند، INNER JOIN می بود، همه افرادی استخراج می شدند که هم پدر و هم مادر برای آنها ثبت شده است.

توجه: اگر هر دو پیوند، LEFT OUTER JOIN می بود، همه افراد استخراج می شدند، مستقل از اینکه پدر و یا مادری برای آنها ثبت شده باشد یا نشده باشد.

توجه: اگر پیوند اول INNER JOIN و پیوند دوم LEFT OUTER JOIN می بود، همه افرادی استخراج می شدند که پدری برای آنها ثبت شده باشد، مستقل از اینکه مادری برای آنها ثبت شده باشد یا نشده باشد. که در این صورت گزینه سوم درست می بود.

۵۷- گزینه (۱) صحیح است.

پرس و جوی خواسته شده باید سطرهای جدول supplier را بر حسب sid و pid گروه بندی کند (یعنی هر قطعه تولید شده توسط هر تهیه کننده یک گروه در نظر گرفته شود) و سپس تعداد رکوردهای هر گروه را بشمارد و گروههایی که این تعداد برای آنها بیش از 10 باشد را در خروجی نشان دهند. تنها گزینه ای که چنین کاری انجام می دهد گزینه اول است.

البته گزینه اول دارای خطای نحوی است، زیرا ستون جلوی select و group by باید یکسان باشند.

۵۸- گزینه (۳) صحیح است.

راه حل اول: پرس و جوی مطرح شده در صورت سوال را در نظر بگیرید:
مشخصات قطعاتی را بدست آورید که توسط بعضی از تهیه کنندگان تولید نمی شود.

توجه: این پرس و جو به فرم زیر توسط الگوی سوم (بعضی) در SQL قابل پیاده‌سازی است.

الگوی سوم (10):

```
SELECT *
FROM Parts P
WHERE EXISTS (SELECT S.sid
               FROM Suppliers S
              WHERE NOT EXISTS (SELECT *
                                 FROM Catalog C
                                WHERE S.sid = C.sid
                                  AND P.pid = C.pid))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«مشخصات قطعاتی که، وجود داشته باشد (EXISTS) تهیه‌کننده‌ای که، قطعه مورد نظر را تولید نکرده باشد (NOT EXISTS). یعنی مشخصات قطعاتی که توسط بعضی تهیه‌کننده‌گان تولید نشده است.»

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول Parts، یک بار به طور کامل از ابتدا تا انتهای جدول Suppliers بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول Suppliers وجود دارد که شرط اتصال select S.sid = C.sid AND P.pid = C.pid داشته باشد، سطر مورد نظر از جدول Parts در خروجی نمایش داده می‌شود.

توجه: شرط لازم برای تبعیت از الگوی سوم وجود NOT EXISTS اول و دوم است و شرط کافی برای تبعیت از الگوی سوم وجود شرط اتصال متناسب و سازگار است.

راه حل دوم: جداول زیر را در نظر بگیرید:

	sid	sname	sid	pid	pid	pname
	s1	sn1	s1	p1	p1	pn1
	s2	sn2	s1	p2	p2	pn2
			s2	p1		

Tehيه کننده‌گان suppliers	توليد catalog	قطعات parts
S	C	P

select *

پرس و جوی مورد نظر را در نظر بگیرید:

form parts p
where(A).....(select s.sid
from suppliers s
where(B)....(select *

from catalog c
where s.sid=c.sid AND p.pid=c.pid))

پرس و جوی مطرح شده در صورت سؤال به صورت زیر است:

«مشخصات قطعاتی را بدست آورید که توسط بعضی از تهیه کننده‌گان تولید نمی‌شود»

به عبارت دیگر می‌توان گفت:

«مشخصات قطعه‌ای در خروجی ظاهر گردد که وجود دارد، تهیه کننده‌ای که آن قطعه را تولید نکرده باشد.
not exists جدول قطعات جدول تهیه‌کننده‌گان exists تویلید

وجود داشتن تهیه کننده با بکارگیری عملگر `exists` به جای A کنترل می‌شود و تولید نشدن آن قطعه توسط یک تهیه کننده نیز که معادل با وجود نداشتن رکوردی با مقادیر p.pid و s.sid مربوطه در جدول تولید است، با بکارگیری عملگر `not exists` به جای B کنترل می‌شود. بنابراین پرس‌وجوی فوق به صورت زیر خواهد بود:

```
select *
from parts p
where exists (select s.sid
               from suppliers s
               where not exists (select *
                                  from catalog c
                                  where s.sid=c.sid AND p.pid=c.pid))
```

برای عملکرد دقیق پرس‌وجوی فوق ادامه راه حل را دنبال کنید:

الگوریتم فوق به این شکل حرکت می‌کند، به شرح الگوریتم دقت کنید:

ابتدا سطر اول از جدول قطعات بررسی می‌شود تا بررسی کامل در همین سطر توقف کنید، اگر جلوی `exists` غیرتهی باشد، یعنی دارای مقدار باشد، نام قطعه مورد نظر از جدول قطعات در خروجی به عنوان قطعه‌ای که توسط برخی تهیه کنندگان تولید نشده است، نمایش داده می‌شود. اما مسأله این است که چه وقت جلوی `exists` غیرتهی می‌شود. پاسخ این است که شماره تهیه کننده‌ای (s.sid) از جمله select s.sid form suppliers s where استخراج گردد، واضح است که وقتی این اتفاق می‌افتد که شرط در همان جمله برقرار باشد. اما چه وقت شرط where برقرار است، پاسخ وقتی که پرس‌وجوی داخل not exists تهی باشد، یعنی قطعه مورد نظر، توسط تهیه کننده موجود بررسی، تولید نشده باشد. در یک بیان ساده، ابتدا سطر اول از جدول قطعات را انتخاب می‌کنیم. مثلاً قطعه p1، سپس برای قطعه p1 از جدول قطعات بررسی می‌کنیم، که آیا تهیه کنندگان موجود در جدول تهیه کنندگان قطعه p1 را تولید کرده‌اند یا خیر.

بنابراین برای قطعه مورد نظر، توسط دستور select جلوی `exists`، بر روی جدول تهیه کنندگان از ابتدا تا انتهای جدول به ازای قطعه مورد نظر، حرکت می‌کنیم، اما به ازای حرکت در هر سطر از جدول تهیه کنندگان، توسط دستور select جلوی `not exists` بررسی می‌گردد که تهیه کننده موجود بررسی در جدول تهیه کنندگان، در جدول تولید، به ازای قطعه موردنظر، تولید داشته است، یا خیر. اگر تولید داشته باشد، توسط دستور select جلوی `not exists` نمایش داده می‌شود. و توسط select s.sid داخل دستور `exists` جلوی نمایش داده نمی‌شود به دلیل شرط دستور where جلوی `exists` ناشی از FALSE بودن not exists به دلیل تهی بودن جلوی `.not exists`.

حال اگر مطابق روال فوق به ازای حداقل یکی از تهیه کنندگان موجود در جدول تهیه کنندگان، به ازای قطعه مورد نظر، حداقل یکی از تهیه کنندگان، قطعه موجود نظر را تولید نکرده باشد. باعث می‌شود تا s.sid داخل دستور select جلوی `exists` تهی نگردد و باعث شود، شرط دستور where در پشت TRUE، `exists` آن را تولید نکرده است، در خروجی نمایش داده می‌شود. بنابراین قطعه مورد نظر موجود در جدول قطعات، به عنوان قطعه‌ای که حداقل یکی از تهیه کنندگان

پرس‌وجوی مطرح شده را مجدداً در نظر بگیرید:

به کادر مستطیل شکل مشخص شده در پرس و جوی فوق توجه کنید، به ازای حرکت در هر سطر از جدول قطعات جمله $p.pid = c.pid$ تغییر می‌کند. هم اکنون در سطر اول جدول قطعات هستیم.

به قطعه کد موجود در جلوی `exists`, دقت کنید:

خروجی
↑
exists (select s.sid
form suppliers s
where ...)

ابتدا سطر اول از جدول تهیه کنندگان یعنی s_1 بررسی می‌شود.
همچنین به قطعه کد موجود در جلوی `not exists`, دقت کنید:

خروجی
↑
not exists (select *
from catalog
where s.sid = c.sid AND p.pid = c.pid
 $\frac{s_1 \quad s_1}{\text{TRUE}} \quad \frac{p_1 \quad p_1}{\text{TRUE}}$
 \underbrace{\hspace{1cm}}_{\text{TRUE}}}

با شرط فوق، بررسی می‌شود که قطعه p_1 ، توسط تهیه کننده S_1 تولید شده است یا خیر. شرط برقرار است، زیرا تهیه کننده S_1 ، قطعه p_1 را تولید کرده است، بنابراین داریم:

چون جلوی not exists تهی نیست، بنابراین شرط where موجود پشت FALSE not exists می‌گردد، بنابراین سطر اول جدول تهیه کنندگان یعنی تهیه کننده s_1 در خروجی دستور select جلوی ظاهر نمی‌شود. چون قطعه p_1 توسط تهیه کننده s_1 تولید شده است.

در ادامه سطر دوم از جدول تهیه کنندگان یعنی s_2 ، همچنان برای قطعه p_1 بررسی می‌شود. با شرط زیر پرسی می‌شود که قطعه p_1 ، توسط تهیه کننده s_2 تولید شده است یا خیر.

(s₁, p₁)
↑
not exists (select *
form catalog c)

$$\text{where } s.\text{sid} = c.\text{sid} \text{ AND } p.pid = c.pid$$

$$\frac{s_2}{\text{TRUE}} \qquad \frac{p_1}{\text{TRUE}}$$

TRUE

شرط برقرار است، زیرا تهیه کننده s_2 ، قطعه p_1 را تولید کرده است، بنابراین داریم:

```
(s2,p1)
↑
not exists (select *
form catalog c
```

چون جلوی not exists تهی نیست، بنابراین شرط where موجود پشت FALSE می‌گردد، بنابراین سطر دوم جدول تهیه کنندگان یعنی تهیه کننده s_2 در خروجی دستور select جلوی exists ظاهر نمی‌شود.

چون قطعه p_1 توسط تهیه کننده s_2 تولید شده است. نتیجه اینکه، قطعه p_1 توسط همه تولید کنندگان، تولید شده است. مطابق پرس‌وجوی مطرح شده داریم:

```
select *
form parts p
where exists (select s.sid
from suppliers s
where ...
```

در نهایت مقدار $s.sid$ در خط فوق در جلوی exists برای قطعه p_1 تهی می‌گردد، چون به دلیل شرایط جلوی مشاهده نمودید که هیچ یک از سطرهای جدول تهیه کنندگان در خروجی دستور select جلوی exists ظاهر نشدند. چون قطعه p_1 توسط تهیه کنندگان s_1 و s_2 تولید شده‌اند. در یک بیان ساده، اگر تهیه کننده‌ای، قطعه مورد نظر را تولید کرده باشد، در خروجی select $s.sid$ در خط فوق، قرار نمی‌گیرد. تولیدات تهیه کننده توسط دستور select در جلوی not exists بررسی می‌گردد.

جلوی exists برای سطر اول جدول قطعات، یعنی قطعه p_1 ، تهی است، زیرا در خروجی select $s.sid$ جلوی exists تهیه کنندگانی که قطعه مورد نظر را تولید نکرده باشند، خارج می‌گردند، از آنجا که پس از بررسی تولیدات همه تهیه کنندگان برای قطعه p_1 توسط دستور select در جلوی not exists مشخص شد که قطعه p_1 توسط همه تهیه کنندگان تولید شده است، بنابراین خروجی select $s.sid$ تهی خواهد بود، چون همه تهیه کنندگان قطعه p_1 را تولید کرده‌اند. پس شرط where در پشت exists است، پس سطر اول جدول قطعات انتخاب نمی‌گردد و به تبع در خروجی نیز نمایش داده نمی‌شود، زیرا قطعه p_1 توسط همه تهیه کنندگان، تولید شده است.

حال سطر دوم از جدول قطعات بررسی می‌شود، تا بررسی کامل در همین سطر توقف کنید. پس pid برابر p_2 است.

```
خروجی
↑
exists (select s.sid
form suppliers s
where ...
```

به قطعه کد موجود در جلوی exists، دقت کنید:

ابتدا سطر اول از جدول تهیه کنندگان یعنی s_1 بررسی می‌شود. همچنین به قطعه کد موجود در جلوی not exists دقت کنید:

خروجی
↑
not exists (select *
from catalog
where s.sid = c.sid AND p.pid = c.pid

$$\frac{s_1 \quad s_1}{\text{TRUE} \quad \text{TRUE}}$$

$$\frac{p_2 \quad p_2}{\text{TRUE} \quad \text{TRUE}}$$

$$\frac{}{\text{TRUE}}$$

با شرط فوق، بررسی می‌شود که قطعه p_2 ، توسط تهیه کننده s_1 تولید شده است یا خیر.

شرط برقرار است، زیرا تهیه کننده s_1 ، قطعه p_2 را تولید کرده است، بنابراین داریم:

(s_1, p_2)
↑
not exists (select *

form catalog c

چون جلوی not exists تهی نیست، بنابراین شرط where موجود پشت FALSE not exists بنا براین سطر اول جدول تهیه کنندگان یعنی تهیه کننده s_1 در خروجی دستور select جلوی ظاهر نمی‌شود. چون قطعه p_2 توسط تهیه کننده s_1 تولید شده است.

در ادامه سطر دوم از جدول تهیه کنندگان یعنی s_2 ، همچنان برای قطعه p_2 بررسی می‌شود.

با شرط زیر بررسی می‌شود که قطعه p_2 ، توسط تهیه کننده s_2 تولید شده است. یا خیر.

where s.sid ≠ c.sid AND p.pid = c.pid

$$\frac{s_2 \quad \times}{\text{FALSE}} \quad \frac{p_2 \quad \times}{\text{FALSE}}$$

$$\frac{}{\text{FALSE}}$$

شرط برقرار نیست، زیرا تهیه کننده s_2 ، قطعه p_2 را تولید نکرده است، بنابراین داریم:

تهی
↑
not exists (select *

form catalog c

چون جلوی not exists تهی است، بنابراین شرط where TRUE موجود پشت FALSE not exists بنا براین سطر دوم جدول تهیه کنندگان یعنی تهیه کننده s_2 ، در خروجی دستور select جلوی ظاهر می‌شود.

چون قطعه p_2 توسط تهیه کننده s_2 تولید نشده است.

نتیجه اینکه، قطعه p_2 توسط همه تولید کنندگان، تولید نشده است، به عبارت دیگر قطعه p_2 ، توسط

بعضی از تهیه کنندگان تولید نشده است.

مطابق پرس و جوی مطرح شده داریم:

select *

form parts p

where exists (select s.sid

from suppliers s

where ...

در نهایت مقدار s.sid در خط فوق در جلوی exists برای قطعه p2 غیرتهی می‌گردد و برابر مقدار s2، چون به دلیل شرایط جلوی not exists مشاهده نمودید که سطر دوم از جدول تهیه‌کنندگان در خروجی دستور select جلوی not exists ظاهر شد، چون قطعه p2 توسط تهیه‌کننده s2 تولید نشده بود.

جلوی exists، برای سطر دوم جدول قطعات، یعنی قطعه p2، غیرتهی است، زیرا در خروجی select s.sid جلوی exists، تهیه‌کنندگانی که قطعه مورد نظر را تولید نکرده باشند، خارج می‌گردند، از آنجا که پس از بررسی تولیدات همه تهیه‌کنندگان برای قطعه p2 توسط دستور select در جلوی not exists select s.sid جلوی exists تهیه‌کنندگان تولید نشده است، بنابراین خروجی select s.sid که قطعه p2 توسط برخی تهیه‌کنندگان قطعه p2 را تولید نکرده‌اند. پس شرط where، در پشت این است، پس سطر دوم جدول قطعات انتخاب می‌گردد، زیرا قطعه p2 توسط تهیه‌کننده s2 تولید نشده است.

۵۹- گزینه (۲) صحیح است.

جداول زیر را در نظر بگیرید:

S#	Sname	Status	City
S1	Sn1	1	C1
S2	Sn2	2	C2
S3	Sn3	3	C3
S4	Sn4	4	C4

S جدول تهیه‌کنندگان

S#	P#	J#	QTY
S1	P1	J1	10
S1	P2	J1	20
S1	P1	J2	30
S1	P2	J2	40

S2	P1	J1	50
S2	P1	J2	20
S2	P2	J2	50
S3	P1	J2	70
S3	P3	J2	80

SPJ جدول

J#	Jname	City
J1	Jn1	C1
J2	Jn2	C2

جدول پروژه *J*

P#	Pname	Weight	City
P1	Pn1	100	C1
P2	Pn2	200	C2
P3	Pn3	300	C3

P جدول قطعه

توجه: زیر پرس‌وجوی مطرح شده در جلوی دستور All، یک Normal Subquery است.

بنابراین ابتدا زیر پرس‌وجوی داخلی یک بار و برای همیشه اجرا می‌گردد، سپس پرس‌وجوی خارجی به ازای حرکت در هر یک از گروه‌های خود (به دلیل اعمال گروه‌بندی) از مقادیر زیر پرس‌وجوی داخلی استفاده می‌کند. بنابراین در ادامه برای زیر پرس‌وجوی داخلی داریم:

```
(select SUM(QTY)
form SPJ
where SPJ.J#=J2'
group by S#)
```

روال کار بدین صورت است که ابتدا سراغ جدول SPJ می‌رویم، سپس شرط 'SPJ.J#='J2' برا روی جدول SPJ اعمال می‌گردد، در ادامه دستور Group by اعمال S# بر روی ستون # اعمال می‌گردد، در انتها دستور SUM(QTY) برای هر گروه به طور مستقل محاسبه می‌گردد و در خروجی پرس‌وجو قرار می‌گیرد. بنابراین در ادامه داریم:

S#	P#	J#	QTY
S ₁	P ₁	J ₂	30
S ₁	P ₂	J ₂	40
S ₂	P ₁	J ₂	20
S ₂	P ₂	J ₂	50
S ₃	P ₁	J ₂	70
S ₃	P ₃	J ₂	80

حال در ادامه براساس زیر پرس‌وجوی مطرح شده، ابتدا دستور by Group بر روی ستون S# اعمال می‌گردد و سپس دستور SUM(QTY) بر روی ستون QTY در هر گروه به طور مستقل انجام می‌گردد. و خروجی‌های آن جهت استفاده پرس‌وجوی خارجی ذخیره می‌گردد، بنابراین در ادامه داریم:

(S ₁)	(S ₂)	(S ₃)
P ₁ J ₂ 30	P ₁ J ₂ 20	P ₁ J ₂ 70
P ₂ J ₂ 40	P ₂ J ₂ 50	P ₃ J ₂ 80
گروه اول	گروه دوم	گروه سوم
SUM(QTY)=70	SUM(QTY)=70	SUM(QTY)=150

در واقع زیر پرس‌وجوی داخلی مطرح شده در جلوی دستور All، برای هر تولید کننده حاضر در پروژه J₂، تعداد کل قطعات تولید شده (از تمام انواع قطعات) توسط آن تولید کننده در پروژه J₂ را می‌دهد. بنابراین داریم:

... <= All(70,70,150)

حال در ادامه، پرس‌وجوی خارجی به ازای حرکت در هر یک از گروه‌های خود، از مقادیر زیر پرس‌وجوی داخلی استفاده می‌کند.

توجه: دستور having حرکت بر روی گروه‌ها را میسر می‌کند، همانطور که دستور where، حرکت بر روی سطراها را میسر می‌کند.

الحاق طبیعی دو جدول S و SPJ، تهیه کننده‌گانی که حداقل یک قطعه برای یک پروژه را تولید کرده‌اند، استخراج می‌کند، همچنین پس از اعمال شرط 'SPJ.J#='J2' داریم:

S#	Sname	Status	City	P#	J#	QTY
S ₁	Sn1	1	C1	P ₁	J ₂	30
S ₁	sn1	1	C1	P ₂	J ₂	40
S ₂	Sn2	2	C2	P ₁	J ₂	20
S ₂	Sn2	2	C2	P ₂	J ₂	50
S ₃	Sn3	3	C3	P ₁	J ₂	70
S ₃	Sn3	3	C3	P ₃	J ₂	80

در ادامه برای سادگی از ستون‌های status و city صرف‌نظر می‌کنیم، بنابراین داریم:

S#	Sname	P#	J#	QTY
S ₁	Sn1	P ₁	J ₂	30
S ₁	Sn1	P ₂	J ₂	40
S ₂	Sn2	P ₁	J ₂	20
S ₂	Sn2	P ₂	J ₂	50
S ₃	Sn3	P ₁	J ₂	70
S ₃	Sn3	P ₃	J ₂	80

حال، براساس پرس‌وجوی مورد نظر، دستور Group by S# و Sname اجرا می‌گردد، بنابراین داریم:

(S ₁ , Sn1)	(S ₂ , Sn2)	(S ₃ , Sn3)
P ₁ J ₂ 30	P ₁ J ₂ 20	P ₁ J ₂ 70
P ₂ J ₂ 40	P ₂ J ₂ 50	P ₃ J ₂ 80
گروه اول	گروه دوم	گروه سوم

حال در ادامه، براساس پرس‌وجوی خارجی، دستور having SUM(QTY) بر روی هر گروه به طور جداگانه اعمال می‌گردد:

توجه: دستور having بر روی گروه‌ها، اعمال می‌گردد.

(S ₁ , Sn1)	(S ₂ , Sn2)	(S ₃ , Sn3)
P ₁ J ₂ 30	P ₁ J ₂ 20	P ₁ J ₂ 70
P ₂ J ₂ 40	P ₂ J ₂ 50	P ₃ J ₂ 80
گروه اول	گروه دوم	گروه سوم
SUM(QTY) = 70	SUM(QTY) = 70	SUM(QTY) = 150

در واقع، پرس‌وجوی خارجی، برای هر تولید کننده‌ای که در پروژه J₂ شرکت داشته است، اگر تعداد کل قطعات تولید شده (از تمام انواع قطعات) توسط وی در پروژه J₂ کمتر از تعداد کل قطعات تولید شده (از تمام انواع قطعات) توسط تک‌تک تولید کنندگان در پروژه J₂ باشد، نام آن تولید کننده را در خروجی می‌آورد.

بنابراین در ادامه داریم:

گام اول: بررسی گروه اول.

پاسخ: SUM(QTY) با مقدار 70 در گروه اول از همه مقادیر (70, 70, 150) کوچکتر یا مساوی است.

بنابراین تهیه کننده Sn1، به عنوان تولید کننده‌ای که کمترین تعداد قطعه (از تمام انواع قطعات) را برای پروژه J₂ تولید کرده است، در خروجی قرار می‌گیرد.

گام دوم: بررسی گروه دوم.

پاسخ: SUM(QTY) با مقدار 70 در گروه دوم از همه مقادیر (70, 70, 150) کوچکتر یا مساوی است.

بنابراین تهیه کننده Sn2، به عنوان تولید کننده‌ای که کمترین تعداد قطعه (از تمام انواع قطعات) را برای پروژه J₂ تولید کرده است، در خروجی قرار می‌گیرد.

$150 \leq \text{All}(70, 70, 150)$

گام سوم: بررسی گروه سوم.

پاسخ: $\text{SUM}(\text{QTY})$ با مقدار 150 در گروه سوم از همه مقادیر (70, 70, 150) کوچکتر یا مساوی نیست.

بنابراین تهیه کننده‌ای Sn3، به عنوان تولید کننده‌ای که کمترین تعداد قطعه (از تمام قطعات) را برای پرروزه J2 تولید کرده باشد، در خروجی قرار نمی‌گیرد.

بنابراین پرس‌وجوی مطرح شده در صورت سؤال، نام تولید کننده‌گانی که کمترین تعداد قطعه (از تمام انواع قطعات) را برای پرروزه کد J2 تولید کرده‌اند را به صورت زیر در خروجی نمایش می‌دهد.

Sname
Sn1
Sn2

توجه: البته پرس‌وجوی مطرح شده دارای کمی خطای نحوی است، ستون‌های جلوی دستور select و group by در پرس‌وجوی خارجی باید یکسان باشند، یا اینکه ستون موجود در جلوی select که در مقابل group by قرار ندارد، در جلوی select با تابع آماری همراه گردد. اگر جلوی دستور select، ستون #S نیز علاوه بر ستون Sname اضافه گردد، خطای نحوی مرتفع می‌گردد.

۶۰- گزینه (۳) صحیح است.

راه حل اول:

جداول زیر را در نظر بگیرید:

mem-no	name	age	mem-no	isbn	date	isbn	...	Publisher
m1	mn1	21	m1	is1	11	is1		McGraw-Hill
m2	mn2	22	m1	is2	12	is2		McGraw-Hill
m3	mn3	23	m2	is1	13	is3		Penguin
Member A			m2	is2	14	is4		Penguin
			m2	is3	15		Book	
Borrowed B								

مطابق پرس‌وجوی مطرح شده در صورت سوال داریم:

«نام اعضایی که همه کتاب‌های منتشرشده توسط McGraw-Hill را به امانت برده‌اند.»

پرس‌وجوی فوق توسط عملگر تقسیم در جبر رابطه‌ای قابل پیاده‌سازی است، اما در SQL می‌بایست عملگر تقسیم شبیه‌سازی گردد.

در ادامه، این پرس‌وجو را به سه فرم زیر می‌نویسیم:

فرم اول:

```

SELECT A.name
FROM Member A, Borrowed B
WHERE A.mem-no=B.mem-no
GROUP BY A.mem-no
HAVING COUNT (B.isbn)=(SELECT COUNT (isbn)
                        FROM Book
                        WHERE Publisher=McGraw-Hill)

```

فرم دوم:

```
SELECT A.name
FROM Member A
WHERE NOT EXISTS ((SELECT isbn
                     FROM Book
                     WHERE publisher = McGraw-Hill)
EXCEPT
(SELECT isbn
FROM Borrowed B
WHERE A.mem-no = B.mem-no))
```

کارکرد قطعه پرس و جوی زیر از پرس و جوی فوق به صورت زیر است:

```
SELECT isbn
FROM Book
WHERE publisher = McGraw-Hill
```

«شماره همه کتاب های منتشر شده توسط انتشارات McGraw-Hill همچنین کارکرد قطعه پرس و جوی زیر از پرس و جوی فوق به صورت زیر است:

```
SELECT isbn
FROM Borrowed B
WHERE A.mem-no = B.mem-no
```

«شماره کتاب های به امانت برده شده توسط یک عضو»

در پرس و جوی فوق، در پرانتز داخلی کل کتاب های چاپ شده، توسط انتشارات McGraw-Hill منهای کلیه کتاب های به امانت برده شده توسط یک عضو می شود و در صورتی که حاصل این تفاضل تهی باشد

نام عضو مورد جستجو در خروجی ظاهر می شود.

بنابراین پرس و جوی فوق، نام اعضا یی که همه کتاب های منتشر شده توسط انتشارات McGraw-Hill را به امانت برده اند، نشان می دهد.

توجه: فرم دوم مطرح شده، مطابق گزینه سوم می باشد، بنابراین گزینه سوم پاسخ سوال خواهد بود.

حال به تشریح نحوه کارکرد پرس و جوی پردازیم:

توسط دستور where در فرم زیر:

```
SELECT A.name
FROM Member A
WHERE NOT EXISTS (...)
```

برای هر سطر از جدول Member شرط جلوی not exists که حاصل یک تفاضل توسط عملگر except می باشد، محاسبه می گردد، اگر تهی بود، شرط جلوی where که همان not exists است، TRUE می گردد و سطر مورد نظر از جدول Member انتخاب می گردد و این رویه برای تک تک سطرهای جدول Member تا به انتهای جدول Member ادامه پیدا می کند. به بیان دیگر این پرس و جو نام اعضا یی را می دهد که پرانتز مقابل not exists برای آنها تهی است. این پرانتز هنگامی تهی می شود که حاصل تفاضل بیان شده در این پرانتز تهی شود. حاصل این تفاضل در صورتی تهی می شود که تعداد کل کتاب های منتشر شده توسط انتشارات McGraw-Hill برابر تعداد کتاب هایی باشد که آن عضو تاکنون به امانت برده است. به عبارت دیگر این پرس و جو نام اعضا یی را می دهد که تاکنون تمام کتاب های منتشر شده توسط انتشارات

McGraw-Hill را به امانت برده‌اند.

توجه: همان‌طور که مشاهده می‌کنید، عضو m₁ و m₂ همه‌ی کتاب‌های منتشرشده توسط انتشارات McGraw-Hill را به امانت برده‌اند.

به بیان دیگر عبارت موجود در جلوی دستور not exists که به صورت زیر است:

```
SELECT isbn
FROM Book
WHERE publisher = McGraw-Hill
```

شماره همه‌ی کتاب‌های منتشرشده توسط انتشارات McGraw-Hill موجود در جدول Book را نمایش می‌دهد.

با توجه به جداول فوق داریم:

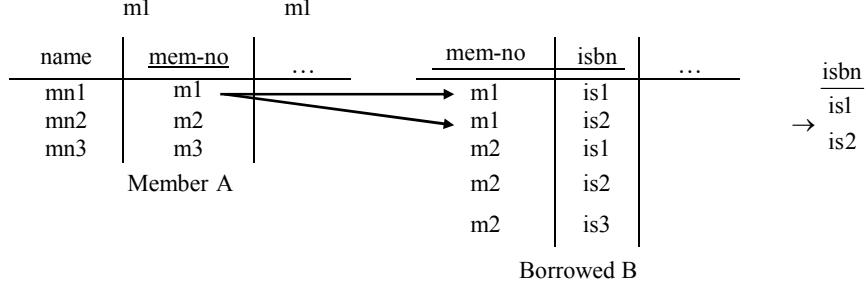
isbn
is1
is2

هم‌چنان عبارت بعد از دستور Except یعنی:

```
SELECT isbn
FROM Borrowed B
WHERE A.mem-no = B.mem-no
```

توسط دستور A.mem – no = B.mem – no Member به محیط خارج یعنی جدول A یا Member متصل می‌گردد. حال به ازای حرکت در هر سطر از جدول Member، یک بار به طور کامل سطرهای جدول Borrowed از ابتدا تا انتهای با توجه به شرط اتصال بررسی می‌گردد. مطابق شکل زیر:

ابتدا برای سطر اول از جدول Member با توجه به شرط اتصال A.mem – no = B.mem – no داریم:



خروجی حاصل از عملگر تفاضل جلوی not به ازای سطر اول از جدول Member به صورت زیر است:

isbn	isbn	خرجی
is1	– is1	= تهی
is2	is2	

جلوی not exists برابر تهی گردید، بنابراین شرط where در پشت not exists TRUE می‌گردد. بنابراین سطر اول از جدول Member به شکل زیر در خروجی نمایش داده می‌شود.

name
mn1

حال برای سطر دوم از جدول Member با توجه به شرط اتصال A.mem-no = B.mem-no داریم:

name	mem-no	...	mem-no	isbn	...	isbn
mn1	m1		m1	is1		
mn2	m2		m1	is2		
mn3	m3		m2	is1		
Member A			Borrowed B			
						→ is1 is2

خروجی حاصل از عملگر تفاضل جلوی not به ازای سطر دوم از جدول Member به صورت زیر است:

$$\begin{array}{ccc} \text{isbn} & \text{isbn} & \text{خروجی} \\ \text{is1} & - \text{is1} & = \quad \text{تهی} \\ & & \\ \text{is2} & \text{is2} & \\ & & \\ & \text{is3} & \end{array}$$

جلوی not برابر تهی گردید، بنابراین شرط where در پشت TRUE می‌گردد. بنابراین سطر دوم از جدول Member به شکل زیر در خروجی نمایش داده می‌شود.

name
mn2

در ادامه برای سطر سوم از جدول Member با توجه به شرط اتصال A.mem-no = B.mem-no داریم:

name	mem-no	...	mem-no	isbn	...	isbn
mn1	m1		m1	is1		
mn2	m2		m1	is2		
mn3	m3		m2	is1		
Member A			Borrowed B			
						→ تهی

خروجی حاصل از عملگر تفاضل جلوی not به ازای سطر سوم از جدول Member به صورت زیر است:

$$\begin{array}{ccc} \text{isbn} & \text{isbn} & \text{خروجی} \\ \text{is1} & - \text{is1} & = \quad \text{تهی} \\ \text{is2} & \text{is2} & \\ & & \end{array}$$

جلوی not برابر غیر تهی گردید، بنابراین شرط where در پشت FALSE not exists می‌گردد.

بنابراین سطر سوم از جدول Member در خروجی نمایش داده نخواهد شد.
بنابراین خروجی نهایی برای «نام اعضايی که همه کتاب‌های منتشرشده توسط انتشارات McGraw-Hill را به امانت برده‌اند.» به صورت زیر است:

name
mn1
mn2

فرم سوم:

```
SELECT A.name
FROM Member A
WHERE NOT EXISTS (SELECT isbn
                   FROM Book
                   WHERE publisher = McGraw-Hill
                   And
                   NOT EXISTS (SELECT isbn
                               FROM Borrowed B
                               WHERE A.mem-no = B.mem-no
                               And Book.isbn = B.isbn))
```

بررسی گزینه چهارم:

```
SELECT A.name
FROM Member A
WHERE NOT EXISTS ((SELECT isbn
                     FROM Borrowed B
                     WHERE A.mem-no = B.mem-no)
                     EXCEPT
                     (SELECT isbn
                     FROM Book
                     WHERE publisher = McGraw-Hill))
```

کارکرد قطعه پرس و جوی زیر از پرس و جوی فوق به صورت زیر است:

```
SELECT isbn
FROM Borrowed B
WHERE A.mem-no = B.mem-no
```

«شماره کتاب‌های به امانت برده شده توسط یک عضو»

همچنین کارکرد قطعه پرس و جوی زیر از پرس و جوی فوق به صورت زیر است:

```
SELECT isbn
FROM Book
WHERE publisher = McGraw-Hill
```

«شماره‌ی همه‌ی کتاب‌های منتشرشده توسط انتشارات McGraw-Hill در پرس و جوی فوق، در پرانتز داخلی کل کتاب‌های به امانت برده شده توسط یک عضو منهای کلیه کتاب‌های چاپ شده، توسط انتشارات McGraw-Hill می‌شود و در صورتی که حاصل این تفاضل تهی باشد نام عضو مورد جستجو در خروجی ظاهر می‌شود.

بنابراین پرس و جوی فوق، نام اعضايی که فقط کتاب‌های منتشرشده توسط انتشارات

McGraw-Hill را به امانت برده‌اند، نشان می‌دهد. یعنی نباید کتاب‌های منتشر شده توسط انتشارات‌های دیگر را به امانت برده باشند، اینکه همه کتاب‌های منتشر شده توسط انتشارات McGraw-Hill را به امانت برده‌اند یا نبرده‌اند مهم نیست، اما این مهم است آنچه به امانت برده شده است، حتماً و فقط و فقط از انتشارات McGraw-Hill باشد.

بررسی گزینه‌های اول و دوم:

مطابق آنچه گفته‌یم، استفاده از دستورات `not exists` یکی از فرم‌های شبیه‌سازی عملگر تقسیم در SQL است. نکته حائز اهمیت این است که، در هیچ یک از فرم‌های شبیه‌سازی عملگر تقسیم، دستور `not exists` به تنها یک مورد استفاده قرار نمی‌گیرد. بنابراین گزینه‌های اول و دوم مطابق پرس و جوی مطرح شده در صورت سوال نیستند. گزینه‌های اول و دوم صرفا ضرب دکارتی دو جدول Book و Borrowed را انجام می‌دهند و براساس شروط مطرح شده سطرهایی را انتخاب می‌کنند، و بسته به شرایط جلوی not exists سطرهایی از جدول Member انتخاب می‌گردد، خروجی استخراج شده از پرس و جوهای مطرح شده در گزینه‌های اول و دوم معتبر نیستند.

راه حل دوم:

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

«نام اعضايی که همه کتاب‌های منتشر شده توسط McGraw-Hill را به امانت برده‌اند.»

توجه: عملگر تقسیم در جبر رابطه‌ای در SQL قابل پیاده‌سازی است. عملگر تقسیم در جبر رابطه‌ای زمانی مورد استفاده قرار می‌گیرد که پرس و جو بخواهد همه حالت‌های یک اتفاق را بررسی کند. مانند نام اعضايی که همه کتاب‌های منتشر شده توسط McGraw-Hill را به امانت برده‌اند. که به امانت بردن کتاب‌ها توسط اعضاء، اتفاق و مقسوم است و همه کتاب‌های منتشر شده توسط McGraw-Hill، حالت‌ها و مقسوم علیه است.

توجه: عملگر تقسیم در جبر رابطه‌ای به فرم زیر در SQL قابل پیاده‌سازی است.

```
SELECT A.name
FROM Member A
WHERE NOT EXISTS ((SELECT isbn
                     FROM Book
                     WHERE publisher=McGraw-Hill)
EXCEPT
(SELECT isbn
FROM Borrowed B
WHERE A.mem-no=B.mem-no))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«نام اعضايی که، وجود نداشته باشد (**NOT EXISTS**) کتابی از McGraw-Hill که، به امانت برده نشده باشد (**EXCEPT**). یعنی نام اعضايی که همه کتاب‌های منتشر شده توسط McGraw-Hill را به امانت برده‌اند.»

۶۱- گزینه (۴) صحیح است.

راه حل اول:

جداول زیر را در نظر بگیرید:

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

مطلوب پرس و جوی مطرح شده در صورت سوال داریم:

«شماره تولیدکنندگانی که همه قطعات را تولید کرده‌اند.»

پرس و جوی فوق توسط عملگر تقسیم در جبر رابطه‌ای قابل پیاده‌سازی است، اما در SQL می‌بایست عملگر تقسیم شبیه‌سازی گردد.

در ادامه، این پرس و جو را به سه فرم زیر شبیه‌سازی می‌کنیم:

فرم اول:

```
SELECT S#
FROM SP
GROUP BY S#
HAVING COUNT (P#) = (SELECT COUNT (P#)
                      FROM P)
```

پرس و جوی فوق، شماره تولیدکنندگانی را می‌دهد که انواع قطعاتی که تولیدکرده‌اند برابر تعداد کل قطعات موجود در جدول قطعه باشد.

در پرس و جوی فوق ابتدا توسط زیر پرس و جوی زیر تعداد کل قطعات موجود در جدول قطعه شمارش و محاسبه می‌گردد.

```
SELECT COUNT (P#)
FROM P
```

که حاصل زیر پرس و جوی فوق برابر مقدار ۲ است، که این مقدار برابر تعداد کل قطعات موجود در جدول P است، که شامل قطعات P1 و P2 است.

در ادامه، پرس و جوی اصلی توسط دستور GROUP BY S# براساس #S به صورت زیر گروه‌بندی می‌شود و در ادامه براساس COUNT (P#). تعداد انواع قطعات تولیدشده در هر گروه محاسبه می‌گردد.

S1	S2
P1 10	P1 10
P2 20	گروه دوم
گروه اول	

COUNT (P#)=2 COUNT (P#)=1

سپس توسط دستور HAVING COUNT (P#)=2 به عنوان شرط انتخاب گروه، گروه‌هایی که آن برابر مقدار ۲ است انتخاب می‌گردند. بنابراین مشاهده می‌شود که شرط مطرح شده فقط

برای گروه اول برقرار است، پس گروه اول انتخاب می‌گردد. به صورت زیر:

S#
Sl

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

توجه: در گزینه دوم اگر دستور FROM SP موجود در SELECT دوم به SELECT FROM P تغییر کند، آنگاه فرم اول ایجاد می‌شود.

فرم دوم:

```
SELECT S#
FROM S
WHERE NOT EXISTS ((SELECT P#
                   FROM P)
                  EXCEPT
                  (SELECT P#
                   FROM SP
                   WHERE S.S# = SP.S#))
```

کارکرد قطعه پرس و جوی زیر از پرس و جوی فوق به صورت زیر است:

```
SELECT P#
FROM P
```

«شماره همه قطعات موجود در جدول قطعات»

همچنین کارکرد قطعه پرس و جوی زیر از پرس و جوی فوق به صورت زیر است:

```
SELECT P#
FROM SP
WHERE S.S# = SP.S#
```

«شماره قطعات تولیدشده توسط یک تولیدکننده»

در پرس و جوی فوق، در پرانتز داخلی کل قطعات موجود در جدول قطعه، منهاي کلیه قطعات تولیدشده توسط یک تولیدکننده می‌شود و در صورتی که حاصل این تفاضل تهی باشد نام تولیدکننده مورد جستجو در خروجی ظاهر می‌شود.

بنابراین پرس و جوی فوق، شماره تولیدکننده‌گانی که همه قطعات موجود در جدول قطعه را تولیدکرده‌اند، نشان می‌دهد.

حال به تشریح نحوی کارکرد پرس و جوی پردازیم:

توسط دستور where در فرم زیر:

```
SELECT S#
FROM S
WHERE NOT EXISTS (...)
```

برای هر سطر از جدول S شرط جلوی not exists که حاصل یک تفاضل توسط عملگر except می‌باشد، محاسبه می‌گردد، اگر تهی بود، شرط جلوی where که همان not exists not است، TRUE می‌گردد و سطر Mورد نظر از جدول S انتخاب می‌گردد و این رویه برای تک تک سطرهای جدول S، تا به انتهای جدول S ادامه پیدا می‌کند. به بیان دیگر این پرس و جو شماره تولیدکننده‌گانی را می‌دهد که پرانتز مقابل not exists برای آن‌ها تهی است. این پرانتز هنگامی تهی می‌شود که حاصل تفاضل بیان شده در این پرانتز تهی شود.

حاصل این تفاضل در صورتی تهی می‌شود که تعداد کل قطعات موجود در جدول قطعه برابر تعداد قطعاتی باشد که آن تولیدکننده تاکنون تولید کرده است. به عبارت دیگر این پرس و جو شماره تولیدکننگانی را می‌دهد که تاکنون تمام قطعات موجود در جدول قطعه را تولید کرده‌اند.

توجه: همان‌طور که مشاهده می‌کنید، تولیدکننده S1 همه‌ی قطعات موجود در جدول قطعه را تولید کرده است.

به بیان دیگر عبارت موجود در جلوی دستور not exists که به صورت زیر است:

شماره‌ی همه‌ی قطعات موجود در جدول قطعات را نمایش می‌دهد.
با توجه به جداول فوق داریم:

P#
P1
P2

هم‌چنان عبارت بعد از دستور except یعنی:

SELECT P#
FROM SP
WHERE S.S# = SP.S#

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

ابتدا برای سطر اول از جدول S با توجه به شرط اتصال $S.S\# = SP.S\#$ داریم:

SI	SI	
----	----	--

Sname	S#	City	S#	P#	QTY	P#
Sn1	S1		S1	P1	10	P1
Sn2	S2		S1	P2	20	
Sn3	S3		S2	P1	30	P2

SP

خروجی حاصل از عملگر تفاضل جلوی not exists به ازای سطر اول از جدول S به صورت زیر است:

P#	P#	خروجی
P1	P1	_____
_____	_____	تهی
P2	P2	

جلوی not exists برابر تهی گردید، بنابراین شرط where در پشت TRUE می‌گردد. بنابراین سطر اول از جدول S به شکل زیر در خروجی نمایش داده می‌شود.

S#

حال برای سطر دوم از جدول S با توجه به شرط اتصال $S.S\# = SP.S\#$ داریم:

S2	S2	
----	----	--

Sname	S#	City	S#	P#	QTY	P#
Sn1	S1		S1	P1	10	P1
Sn2	S2		S1	P2	20	
Sn3	S3		S2	P1	30	

SP

خروجی حاصل از عملگر تفاضل جلوی not exists به ازای سطر دوم از جدول S به صورت زیر است:

$$\frac{P\#}{P1} - \frac{P\#}{P1} = \frac{\text{خروجی}}{P2}$$

بنابراین سطر دوم از جدول S در خروجی نمایش داده نخواهد شد.

در ادامه برای سطر سوم از جدول S با توجه به شرط اتصال $S.S\# = SP.S\#$ داریم:

Sname	<u>S#</u>	City	<u>S#</u>	P	QTY	P#
Sn1	S1		S1	P1		
Sn2	S2		S1	P2		
Sn3	S3		S2	P1		

→ تبعی

خروجی حاصل از عملگر تفاضل جلوی not exists به ازای سطر سوم از جدول S به صورت زیر است:

$$\frac{P\#}{P1} - \frac{P\#}{تھی} = \frac{\text{خروجی}}{P1}$$

جلوی not exists برابر غیر تهی گردید، بنابراین شرط where در پشت not exsist می‌گردد، بنابراین سطر سوم از جدول S در خروجی نمایش داده نخواهد شد.

بنابراین خروجی نهایی برای «شماره تولیدکنندگانی» که همه قطعات موجود در جدول قطعه را تولید کرده‌اند به صورت زیر است:

S#
S1

فرم سوم:

```
SELECT S#
FROM S
WHERE NOT EXISTS (SELECT P#
                   FROM P
                   WHERE NC
```

And P.P# = SP.P#))

الله فـ دـان

که در این سطح حرفت می‌شود، به سرچ الگوریتم دست نمیدهد.
ابتدا سطر اول از جدول S بررسی می‌شود تا بررسی کامل در همین سطر توقف کنید، اگر جلوی not exists اول تهی باشد، شماره تولیدکننده از جدول S در خروجی به عنوان تولیدکننده‌ای که همه قطعات را تولید کرده است، نمایش داده می‌شود. اما مساله این است که چه وقت جلوی not exists باید باشند.

این است که هیچ $P\#$ در جمله:

`SELECT P# FROM P WHERE` موجود نباشد، واضح است که وقتی این اتفاق می‌افتد که شرط `where` در همان جمله برقرار نباشد. اما چه وقت شرط `where` برقرار نیست، پاسخ، وقتی که پرس‌وجوی داخل `not exists` دوم تهی نباشد، یعنی قطعه تولید شده باشد.

در یک بیان ساده، ابتدا سطر اول از جدول S را انتخاب می‌کنیم. برای مثال تولیدکننده $S1$. سپس برای تولیدکننده $S1$ از جدول S بررسی می‌کنیم، که آیا همه قطعات موجود در جدول قطعه برای تولیدکننده $S1$ تولید شده‌اند یا خیر. بنابراین یک تولیدکننده خاص، توسط دستور `select` جلوی `not exists` اول، بر روی جدول P از ابتدتا تا انتهای جدول به ازای یک تولیدکننده خاص، حرکت می‌کیم، اما به ازای حرکت در هر سطر از جدول P ، توسط دستور `select` جلوی `not exists` دوم، بررسی می‌گردد که قطعه موجود در جدول قطعه، در جدول SP ، تولیدشده است یا خیر. اگر تولیدشده باشد، توسط * داخل دستور `select` جلوی `not exists` دوم نمایش داده می‌شود و توسط $P\#$ داخل دستور `select` جلوی `not exists` اول نمایش داده نمی‌شود به دلیل بودن شرط دستور `where` `not exists` اول ناشی از `FALSE` بودن. داده نمایش داده به دلیل تهی بودن جلوی `not exists` دوم. حال اگر مطابق روال فوق به ازای همه قطعات موجود در جدول قطعه، برای یک تولیدکننده خاص، همه قطعات، تولیدشده باشند، باعث می‌شود تا $P\#$ داخل دستور `select` جلوی `not exists` اول تهی گردد و باعث شود، شرط دستور `where` در پشت اول `not exists` گردد، بنابراین آن تولیدکننده خاص موجود در جدول S در خروجی نمایش داده می‌شود.

پرس‌وجوی مطرح شده در فرم سوم را مجدداً در نظر بگیرید:

```
SELECT S#
FROM S
WHERE NOT EXISTS (SELECT P#
                   FROM P
                   WHERE NOT EXISTS (SELECT *
                                      FROM SP
                                      WHERE S.S# = SP.S#
                                         And P.P# = SP.P#))
```

به شرط اتصال $S.S\# = SP.S\#$ موجود در پرس‌وجوی فوق توجه کنید به ازای حرکت در هر سطر جدول S مقدار شرط اتصال $S.S\# = SP.S\#$ تغییر می‌کند. هم اکنون در سطر اول جدول S هستیم، پس $S\#$ برابر $S1$ می‌باشد.

به قطعه کد زیر موجود در جلوی `not exists` اول، دقت کنید:

```
NOT EXISTS ((SELECT P#
              FROM P
              WHERE
```

ابتدا سطر اول از جدول P یعنی قطعه $P1$ بررسی می‌شود.

با شرط زیر بررسی می‌شود که تولیدکننده $S1$ قطعه $P1$ را تولیدکرده است یا خیر.

$S.S\# = SP.S\#$	and	$SP.P\# = P.P\#$	
$S1$	$S1$	$P1$	$P1$
TRUE		TRUE	
TRUE			

شرط برقرار است، زیرا قطعه P1 توسط تولیدکننده S1 تولیدشده است، بنابراین خروجی فعلی قطعه پرس و جوی زیر:

```
NOT EXISTS (SELECT *
  FROM SP
  WHERE S.S# = SP.S#
  And P.P# = SP.P#)
```

برابر مقدار P1 (سطر P1 به دلیل استفاده از عملگر *) می باشد.

چون جلوی not exists دوم غیر تهی است، بنابراین شرط where موجود پشت not Exists دوم می گردد، بنابراین سطر اول جدول قطعه یعنی قطعه P1 در خروجی دستور select جلوی not exists اول ظاهر نمی شود. چون قطعه P1 توسط تولیدکننده S1 تولیدشده است.

در ادامه سطر دوم از جدول قطعه یعنی قطعه P2 همچنان برای تولیدکننده S1 بررسی می شود.
با شرط زیر بررسی می شود که تولیدکننده S1 قطعه P2 را تولیدکرده است یا خیر.

$S.S\# = SP.S\#$	and	$SP.P\# = P.P\#$	
S1	S1	P2	P2
TRUE		TRUE	
TRUE			

شرط برقرار است، زیرا قطعه P2 توسط تولیدکننده S1 تولیدشده است، بنابراین خروجی فعلی قطعه پرس و جوی زیر:

```
NOT EXISTS (SELECT *
  FROM SP
  WHERE S.S# = SP.S#
  And P.P# = SP.P#)
```

برابر مقدار P2 (سطر P2 به دلیل استفاده از عملگر *) می باشد.

چون جلوی not exists دوم غیر تهی است، بنابراین شرط where موجود پشت not Exists دوم می گردد، بنابراین سطر دوم جدول قطعه یعنی قطعه P2 نیز در خروجی دستور select جلوی not exists اول ظاهر نمی شود. چون قطعه P2 توسط تولیدکننده S1 تولیدشده است.

نتیجه این که تولیدکننده S1 همه قطعات موجود در جدول قطعه را تولیدکرده است.
حال یکبار دیگر پرس و جوی مطرح شده در فرم سوم را در نظر بگیرید:

```
SELECT S#
FROM S
WHERE NOT EXISTS ((SELECT P#
  FROM P
  WHERE NOT EXISTS (SELECT *
    FROM SP
    WHERE S.S# = SP.S#
    And P.P# = SP.P#)))
```

در نهایت خروجی قطعه پرس و جوی زیر:

```
SELECT P#
FROM P
WHERE NOT EXISTS (...)
```

برای تولیدکننده S1 موجود در سطر اول جدول تولیدکننده برابر مقدار تهی می‌شود. چون به دلیل شرایط جلوی not exists دوم مشاهده نمودید که هیچ یک از سطرهای جدول قطعه در خروجی دستور select جلوی not exists اول ظاهر نشدند، زیرا در خروجی select P# not exists اول، قطعات تولیدنشده، توسط یک تولیدکننده خاص، خارج می‌گردند، از آنجا که پس از بررسی قطعات تولیدشده برای تولیدکننده S1 توسط دستور select در جلوی not exists select P# اول، همهی قطعات را تولید کرده است، بنابراین خروجی select P# جلوی not exists اول، تهی خواهد شد.

بنابر مطالب فوق شرط where در پشت not exists اول برای تولیدکننده S1 برابر مقدار TRUE خواهد بود، پس سطر اول جدول تولیدکننده انتخاب می‌گردد و # با مقدار S1 در خروجی قرار می‌گیرد.

توجه: در یک بیان ساده اگر قطعه‌ای برای یک تولیدکننده خاص تولیدشده باشد در خروجی قطعه پرس و جوی فوق قرار نمی‌گیرد. تولید یک قطعه توسط یک تولیدکننده خاص توسط دستور select در جلوی not exists دوم بررسی می‌گردد.

حال سطر دوم از جدول تولیدکننده بررسی می‌شود، تا بررسی کامل در همین سطر توقف کنید.

به قطعه که زیر موجود در جلوی not exists اول، دقت کنید:

```
NOT EXISTS ((SELECT P#
              FROM P
              WHERE
```

ابتدا سطر اول از جدول P یعنی قطعه P1 بررسی می‌شود.
با شرط زیر بررسی می‌شود که تولیدکننده S2، قطعه P1 را تولیدکرده است یا خیر.

$$\begin{array}{ll} S.S\# = SP.S\# \text{ and } SP.P\# = P.P\# \\ \hline S2 & S2 \quad P1 & P1 \\ \text{TRUE} & \text{TRUE} \\ \hline & \text{TRUE} \end{array}$$

شرط برقرار است، زیرا قطعه P1 توسط تولیدکننده S2 تولیدشده است، بنابراین خروجی فعلی قطعه پرس و جوی زیر:

```
NOT EXISTS (SELECT *
              FROM SP
              WHERE S.S# = SP.S#
              And P.P# = SP.P#)
```

برابر مقدار P1 (سطر P1 به دلیل استفاده از عملگر *) می‌باشد.

چون جلوی not exists دوم غیر تهی است، بنابراین شرط where موجود پشت not exists دوم می‌گردد، بنابراین سطر اول جدول قطعه یعنی قطعه P1 در خروجی دستور select جلوی not exists اول ظاهر نمی‌شود. چون قطعه P1 توسط تولیدکننده S2 تولیدشده است.

در ادامه سطر دوم از جدول قطعه یعنی قطعه P2 همچنان برای تولیدکننده S2 بررسی می‌شود.
با شرط زیر بررسی می‌شود که تولیدکننده S2، قطعه P2 را تولیدکرده است یا خیر.

$$\begin{array}{ll} S.S\# = SP.S\# \text{ and } SP.P\# = P.P\# \\ \hline S2 & \times \quad \times \quad P2 \\ \text{FALSE} & \text{FALSE} \\ \hline & \text{FALSE} \end{array}$$

شرط برقرار نیست، زیرا قطعه P2 توسط تولیدکننده S2 تولیدنشده است، بنابراین خروجی فعلی قطعه پرس و جوی زیر:

```
NOT EXISTS (SELECT *
  FROM SP
  WHERE S.S# = SP.S#
  And P.P# = SP.P#)
```

برابر مقدار تهی می‌باشد.

چون جلوی not exists not exists دوم تهی است، بنابراین شرط where موجود پشت TRUE not exists دوم می‌گردد، بنابراین سطر دوم جدول قطعه یعنی قطعه P2 در خروجی دستور select جلوی not exists اول ظاهر می‌شود. چون قطعه P2 توسط تولیدکننده S2 تولید نشده است. نتیجه این که تولیدکننده S2 همه قطعات موجود در جدول قطعه را تولید نکرده است.

حال یکبار دیگر پرس و جوی مطرح شده در فرم سوم را در نظر بگیرید:

```
SELECT S#
  FROM S
 WHERE NOT EXISTS ((SELECT P#
    FROM P
    WHERE NOT EXISTS (SELECT *
      FROM SP
      WHERE S.S# = SP.S#
      And P.P# = SP.P#))
```

در نهایت خروجی قطعه پرس و جوی زیر:

SELECT P#
FROM P
WHERE NOT EXISTS (...)

برای تولیدکننده S2 موجود در سطر دوم جدول تولیدکننده برابر مقدار غیرتهی و برابر مقدار P2 می‌شود. چون به دلیل شرایط جلوی not exists دوم مشاهده نمودید که سطر دوم از جدول قطعه در خروجی دستور select جلوی not exists اول ظاهر شد، زیرا در خروجی select P# جلوی not exists select P# در خروجی select P# تولیدکننده، توسط یک تولیدکننده خاص، خارج می‌گردد، از آن‌جا که پس از بررسی قطعات تولیدشده برای تولیدکننده S2 توسط دستور select در جلوی not exists دوم، مشخص شد که تولیدکننده S2، قطعه P2 را تولید نکرده است، بنابراین خروجی select P# جلوی not exists اول، غیرتهی خواهد شد. بنابر مطالب فوق شرط where در پشت not exists اول برای تولیدکننده S2 برابر مقدار FALSE خواهد بود، پس سطر دوم جدول تولیدکننده انتخاب نمی‌گردد و S# با مقدار S2 در خروجی قرار نمی‌گیرد. در نهایت سطر سوم از جدول تولیدکننده بررسی می‌شود، که نتیجه‌ای همچون سطر دوم از جدول تولیدکننده خواهد داشت.

نتیجه اینکه، پرس و جوی فرم سوم شماره تولیدکننده‌گانی را می‌دهد که قطعه‌ای وجود ندارد که آنها آنرا تولید نکرده باشند. به عبارت دیگر شماره تولیدکننده‌گانی را می‌دهد که همه قطعات موجود در جدول قطعه را تولید نمی‌داند.

بررسی گزینه‌های اول و سوم:

مطابق پرس و جوهای مطرح شده در گزینه‌های اول و سوم داریم:

«شماره تولیدکننده‌گانی که حداقل یک قطعه را تولید کرده‌اند.»

در ادامه، این پرس و جو را به چهار فرم زیر می‌نویسیم:

فرم اول:

```
SELECT distinct (S.S#)
FROM S, SP
WHERE S.S# = SP.S#
```

فرم دوم:

```
SELECT S#
FROM S
WHERE S# IN (SELECT S#
               FROM SP)
```

ابتدا زیر پرس و جوی داخلی یک بار و برای همیشه اجرا می‌گردد، سپس پرس و جوی خارجی به ازای حرکت در هر یک از سطوحهای خود، از مقادیر زیر پرس و جوی داخلی استفاده می‌کند.

در پرس و جوی فوق به ازای حرکت در هر سطح از جدول S مجموعه جلوی in بررسی می‌گردد. آیا #S# داخل این مجموعه قرار دارد یا خیر. اگر قرار داشت سطح مورد نظر از جدول S در خروجی نمایش داده می‌شود.

توجه: فرم دوم مطرح شده، مطابق گزینه اول می باشد، که مطابق پرس و جوی مطرح شده در صورت سوال نمی باشد، بنابراین پاسخ سوال نیز نمی باشد.

فرم سوم:

```
SELECT S#
FROM S
WHERE EXISTS (SELECT P#
                FROM SP
                WHERE S.S# = SP.S#)
```

به ازای حرکت در هر یک از سطوحهای پرس‌وجوی خارجی، یک بار به طور کامل از ابتدا تا انتها زیر پرس‌وجوی داخلی اجرا و براساس شرطی که زیر پرس‌وجوی داخلی را به پرس‌وجوی خارجی متصل می‌کند، بررسی انجام می‌شود. مانند دو حلقه تو در تو، که به ازای هر بار اجرای حلقه خارجی، یک بار به طور کامل حلقه داخلی اجرا می‌گردد.

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول SP بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول SP وجود دارد که شرط اتصال S.# موجود در select داخلي را برقرار کند، اگر وجود داشته باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

فرم چهارم:

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول P بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول P وجود دارد که شرط اتصال $S.S\# = SP.S\# \text{ AND } SP.P\# = P.P\#$ موجود در select داخلی را برقرار کند، اگر وجود داشته باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

توجه: فرم چهارم مطرح شده، مطابق گزینه سوم می‌باشد، که مطابق پرس و جوی مطرح شده در صورت سوال نمی‌باشد، بنابراین پاسخ سوال نیز نمی‌باشد.

راه حل دوم:

پرس و جوی مطرح شده در صورت سوال را در نظر بگیرید:
«شماره تولیدکنندگانی که همه قطعات را تولید کرده‌اند.»

توجه: عملگر تقسیم در جبر رابطه‌ای توسط الگوی اول در SQL قابل پیاده‌سازی است. عملگر تقسیم در جبر رابطه‌ای زمانی مورد استفاده قرار می‌گیرد که پرس و جو بخواهد همه حالت‌های یک اتفاق را بررسی کند. مانند شماره تولیدکنندگانی که همه قطعات را تولیدکرده‌اند. که تولیدکردن قطعات توسط تولیدکنندگان، اتفاق و مقسوم است و همه قطعات، حالت‌ها و مقسم علیه است.

توجه: این پرس و جو به فرم زیر توسط الگوی اول (همه) در SQL قابل پیاده‌سازی است.

الگوی اول (۰۰):

```
SELECT S#
FROM S
WHERE NOT EXISTS (SELECT P#
                    FROM P
                    WHERE NOT EXISTS (SELECT *
                                      FROM SP
                                      WHERE S.S# = SP.S#
                                         AND P.P# = SP.P#))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«شماره تولیدکنندگانی که، وجود نداشته باشد (**NOT EXISTS**) قطعه‌ای که، تولید نشده باشد (**NOT EXISTS**). یعنی شماره تولیدکنندگانی که همه قطعات را تولید کرده‌اند.»

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول P بررسی می‌گردد، تا مشخص گردد آیا هیچ سطrix در جدول P وجود ندارد که شرط اتصال $S.S\# = SP.S\# \text{ AND } SP.P\# = P.P\#$ موجود در select داخلی را برقرار نکند، اگر وجود نداشته باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

توجه: شرط لازم برای تبعیت از الگوی اول وجود **NOT EXISTS** اول و **NOT EXISTS** دوم است و شرط کافی برای تبعیت از الگوی اول وجود شرط اتصال مناسب و سازگار است.

- گزینه (۳) صحیح است.

راه حل اول:

جداول زیر را در نظر بگیرید:

<u>sid</u>	sname	...	<u>sid</u>	<u>bid</u>	...	<u>bid</u>	...
s1	sn1		s1	b1		b1	
s2	sn2		s1	b2		b2	
s3	sn3		s2	b1			
قايقان Sailor S			رزو Reservation R			قايق Boats B	

مطابق پرس و جوی مطرح شده در صورت سوال داریم:

«نام قایقران‌هایی که همه قایق‌ها را رزرو کرده‌اند.»

پرس و جوی فوق توسط عملگر تقسیم در جبر رابطه‌ای قابل پیاده‌سازی است. اما در SQL می‌بایست عملگر تقسیم شبیه‌سازی گردد.

در ادامه، این پرس و جو را به دو فرم زیر شبیه‌سازی می‌کنیم:

فرم اول:

```
SELECT Sname
FROM Sailors S
WHERE NOT EXISTS ((SELECT bid
                     FROM Boats B)
EXCEPT
(SELECT bid
         FROM Reservation R
        WHERE R.sid = S.sid)))
```

توجه: گزینه اول خطای نحوی دارد و به دلیل داشتن دستور اضافه $=$ B.bid = R.bid از سوی کامپایلر کامپایل نخواهد شد. دقت کنید که جدول Boats در پرس و جوی بعد از دستور except قابل شناخت نیست و به تبع حضور دستور R.bid = B.bid در پرس و جوی بعد از دستور except از سوی کامپایلر خطای کامپایلری محسوب می‌گردد، فرم صحیح گزینه اول به شکل فرم اول مطرح شده می‌باشد. بنابراین گزینه اول پاسخ سوال نمی‌باشد.

کارکرد قطعه پرس و جوی زیر از پرس و جوی فوق به صورت زیر است:

```
SELECT bid
FROM Boats B
```

«شماره‌ی همه‌ی قایق‌های موجود در جدول قایق»

همچنین کارکرد قطعه پرس و جوی زیر از پرس و جوی فوق به صورت زیر است:

```
SELECT bid
FROM Reservation R
WHERE R.sid = S.sid
```

«شماره قایق‌های رزرو شده توسط یک قایقران»

در پرس و جوی فوق، در پرانتز داخلی کل قایق‌های موجود در جدول قایق، منهای کلیه قایق‌های رزرو شده توسط یک قایقران می‌شود و در صورتی که حاصل این تفاضل تهی باشد نام قایقران مورد جستجو در خروجی ظاهر می‌شود.

بنابراین پرس و جوی فوق، نام قایقران‌هایی که همه قایق‌های موجود در جدول قایق را رزرو کرده‌اند، نشان می‌دهد.

حال به تشریح نحوه کارکرد پرس و جوی پردازیم:
توسط دستور where در فرم زیر:

```
SELECT Sname
FROM Sailors S
WHERE NOT EXISTS (...)
```

به بیان دیگر این پرس و جو شماره تولیدکنندگانی را می‌دهد که پرانتز مقابل not برای آن‌ها تهی است. این پرانتز هنگامی تهی می‌شود که حاصل تفاضل بیان شده در این پرانتز تهی شود. حاصل این تفاضل در صورتی تهی می‌شود که تعداد کل قطعات موجود در جدول قطعه برابر تعداد قطعاتی باشد که آن تولیدکننده تاکنون تولید کرده است. به عبارت دیگر این پرس و جو شماره تولیدکنندگانی را می‌دهد که تاکنون تمام قطعات موجود در جدول قطعه را تولید کرده‌اند.

برای هر سطر از جدول شرط جلوی not که حاصل یک تفاضل توسط عملگر except می‌باشد، محاسبه می‌گردد، اگر تهی بود، شرط جلوی where که همان not exists است، TRUE می‌گردد و سطر مورد نظر از جدول Sailor انتخاب می‌گردد و این رویه برای تک تک سطرهای جدول Sailor، تا به انتهای جدول Sailor ادامه پیدا می‌کند. به بیان دیگر پرس و جوی فوق نام قایقران‌هایی را می‌دهد که پرانتز مقابل not برای آن‌ها تهی است. این پرانتز هنگامی تهی می‌شود که حاصل تفاضل بیان شده در این پرانتز تهی شود. حاصل این تفاضل در صورتی تهی می‌شود که تعداد کل قایق‌ها برابر تعداد قایق‌هایی باشد که آن قایقران تاکنون رزرو کرده است. به عبارت دیگر پرس و جوی فوق نام قایقران‌هایی را می‌دهد که تاکنون تمام قایق‌ها را رزرو کرده‌اند.

توجه: همان‌طور که مشاهده می‌کنید، قایقران ۱، همه‌ی قایق‌های موجود در جدول قایق را رزرو کرده است.

به بیان دیگر عبارت موجود در جلوی دستور not exists که به صورت زیر است:

```
SELECT Sname
FROM Sailors S
```

شماره‌ی همه‌ی قطعات موجود در جدول قطعات را نمایش می‌دهد.
با توجه به جداول فوق داریم:

bid
b1
b2

هم‌چنین عبارت بعد از دستور except یعنی:

```
SELECT bid
FROM Reservation R
WHERE R.sid = S.sid
```

توسط دستور $S.sid=R.sid$ به محیط خارج یعنی جدول S یا Sailor متصل می‌گردد. حال به ازای حرکت در هر سطر از جدول Sailor، یک بار به طور کامل سطرهای جدول Reservation از ابتدا تا انتها با توجه به شرط اتصال بررسی می‌گردد. مطابق شکل زیر:

ابتدا برای سطر اول از جدول Sailor با توجه به شرط اتصال $S.sid = R.sid$ داریم:

خروجی حاصل از عملگر تفاضل جلوی not exists به ازای سطر اول از جدول Sailor به صورت زیر است:

$$\frac{\text{bid}}{\text{b1}} - \frac{\text{bid}}{\text{b2}} = \frac{\text{خرожی تهی}}{\text{b2}}$$

جلوی not exists برایر تهی گردید، بنابراین شرط where در پشت not exists TRUE می‌گردد. بنابراین سطر اول از جدول Sailor به شکل زیر در خروجی نمایش داده می‌شود.

Sname
Sn1

حال برای سطر دوم از جدول Sailor با توجه به شرط اتصال $\text{sid} = R.\text{sid}$ داریم:

sname	<u>sid</u>	...	<u>sid</u>	<u>bid</u>	...
sn1	s1		s1	b1	
sn2	s2		s1	b2	
sn3	s3		s2	b1	

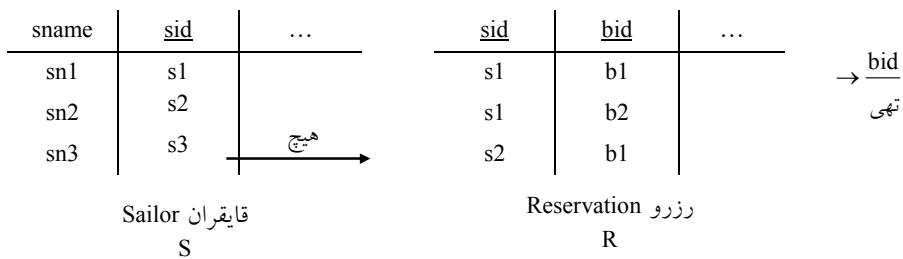
→ $\frac{\text{bid}}{\text{bl}}$

خروجی حاصل از عملگر تفاضل جلوی `exists not` به ازای سطر دوم از جدول Sailor به صورت زیر است:

$$\frac{\text{bid}}{\text{b1}} - \frac{\text{bid}}{\text{b1}} = \frac{\text{خروجی}}{\text{b2}}$$

بنابراین سطر دوم از جدول Sailors در خروجی نمایش داده نخواهد شد.

در ادامه برای سطر سوم از جدول Sailor با توجه به شرط اتصال $S.sid = R.sid$ داریم:



خروجی حاصل از عملگر تفاضل جلوی not exists به ازای سطر سوم از جدول S به صورت زیر است:

$$\frac{\text{bid}}{\text{bl}} - \frac{\text{bid}}{\text{تهنی}} = \frac{\text{خرجوی}}{\text{bl}}$$

$$\frac{\text{b2}}{\text{b2}} - \frac{\text{b2}}{\text{b2}} = \frac{\text{خرجوی}}{\text{bl}}$$

جلوی not exists برابر غیر تهنی گردید، بنابراین شرط where در پشت FALSE می‌گردد.
بنابراین سطر سوم از جدول Sailor در خروجی نمایش داده نخواهد شد.
بنابراین خروجی نهایی برای «نام قایقران‌هایی که همه قایق‌ها را رزرو کرده‌اند» به صورت زیر است:

Sname
Sn1

فرم دوم:

```
SELECT Sname
FROM Sailors S
WHERE NOT EXISTS (SELECT bid
                   FROM Boats B
                   WHERE NOT EXISTS (SELECT *
                                     FROM Reservation R
                                     WHERE R.bid = B.bid
                                     AND R.sid = S.sid))
```

توجه: فرم دوم مطرح شده، مطابق گزینه سوم می‌باشد، بنابراین گزینه سوم پاسخ سوال خواهد بود.
حال به بررسی فرم دوم می‌پردازیم:

الگوریتم فوق به این شکل حرکت می‌کند، به شرح الگوریتم دقت کنید:

ابتدا سطر اول از جدول قایقران بررسی می‌شود تا بررسی کامل در همین سطر توقف کنید، اگر جلوی not exists اول تهنی باشد، نام قایقران از جدول Sailor در خروجی به عنوان قایقرانی که همه قایق‌ها رزرو کرده است، نمایش داده می‌شود. اما مساله این است که چه وقت جلوی not exists اول تهنی می‌گردد، پاسخ این است که هیچ bid در جمله:

select bid from Boats B where همان جمله برقرار نباشد. اما چه وقت شرط where برقرار نیست، پاسخ، وقتی که پرس‌وجوی داخل not exists دوم تهنی نباشد، یعنی قایق رزرو شده باشد.

در یک بیان ساده، ابتدا سطر اول از جدول قایقران را انتخاب می‌کنیم. برای مثال قایقران S1 سپس برای قایقران S1 از جدول قایقران بررسی می‌کنیم، که آیا همه قایق‌های موجود در جدول قایق برای قایقران S1

رزرو شده‌اند یا خیر. بنابراین برای یک قایقران خاص، توسط دستور select جلوی not exists اول، بر روی جدول قایق از ابتدا تا انتهای جدول به ازای یک قایقران خاص، حرکت می‌کنیم، اما به ازای حرکت در هر سطر از جدول قایق، توسط دستور select جلوی not exists دوم، بررسی می‌گردد که قایق موجود در جدول قایق، در جدول رزرو، رزرو شده است یا خیر. اگر رزرو شده باشد، توسط bid دارای دستور select جلوی not exists دوم نمایش داده می‌شود و توسط bid دارای دستور select جلوی not exists داده نمی‌شود به دلیل FALSE بودن شرط دستور where جلوی not exists اول ناشی از بودن FALSE داده نمی‌شود به دلیل تهی نبودن جلوی not exists دوم. حال اگر مطابق روال فوق به ازای همه قایقهای موجود در جدول قایق، برای یک قایقران خاص، همه قایقهای رزرو شده باشند، باعث می‌شود تا دستور select جلوی not exists اول تهی گردد و باعث شود، شرط دستور where در پشت bid دارای دستور NOT EXISTS (SELECT * FROM Reservation R WHERE R.bid = B.bid AND R.sid = S.sid) می‌شود.

پرس‌وجوی مطرح شده در فرم دوم را مجدداً در نظر بگیرید:

```
SELECT Sname
FROM Sailors S
WHERE NOT EXISTS (SELECT bid
                   FROM Boats B
                   WHERE NOT EXISTS (SELECT *
                                      FROM Reservation R
                                      WHERE R.bid = B.bid
                                         AND R.sid = S.sid))
```

به شرط اتصال R.sid = S.sid موجود در پرس‌وجوی فوق توجه کنید به ازای حرکت در هر سطر جدول Sailor جمله R.sid=S.sid تغییر می‌کند. هم اکنون در سطر اول جدول Sailor هستیم. پس sid برابر s1 می‌باشد.

به قطعه کد زیر موجود در جلوی not exists اول، دقت کنید:

```
NOT EXISTS ((SELECT bid
               FROM Boats B
               WHERE
```

ابتدا سطر اول از جدول قایق یعنی قایق b1 بررسی می‌شود.

با شرط زیر بررسی می‌شود که قایقران s1، قایق b1 را رزرو کرده است یا خیر.

$$\begin{array}{cccc} R.bid & = & B.bid & \text{and} \\ b1 & & b1 & \\ \text{TRUE} & & \text{TRUE} & \\ \hline & & & \text{TRUE} \end{array}$$

شرط برقرار است، زیرا قایق b1 توسط قایقران s1 رزرو شده است، بنابراین خروجی فعلی قطعه پرس و جوی زیر:

```
NOT EXISTS (SELECT *
               FROM Reservation R
               WHERE R.bid = B.bid
                 AND R.sid = S.sid))
```

برابر مقدار b1 (سطر b1 به دلیل استفاده از عملگر *) می‌باشد.

چون جلوی not exists دوم غیر تهی است، بنابراین شرط where موجود پشت FALSE not Exists دوم می‌گردد، بنابراین سطر اول جدول قایق یعنی قایق b1 در خروجی دستور select جلوی not exists اول ظاهر نمی‌شود. چون قایق b1 توسط قایقران s1 رزرو شده است.

در ادامه سطر دوم از جدول قایق یعنی b2 همچنان برای قایقران s1 بررسی می‌شود.
با شرط زیر بررسی می‌شود که قایقران s1، قایق b2 را رزرو کرده است یا خیر.

$$\begin{array}{ll} R.bid = B.bid & \text{and } R.sid = S.sid \\ b2 & b2 \\ b2 & s1 \\ \hline \text{TRUE} & \text{TRUE} \\ \hline & \text{TRUE} \end{array}$$

شرط برقرار است، زیرا قایق b2 توسط قایقران s1 رزرو شده است، بنابراین خروجی فعلی قطعه پرس و جوی زیر:

NOT EXISTS (SELECT *)

```
FROM Reservation R
WHERE R.bid = B.bid
AND R.sid = S.sid))
```

برابر مقدار b2 (سطر b2 به دلیل استفاده از عملگر *) می‌باشد.

چون جلوی not exists دوم غیر تهی است، بنابراین شرط where موجود پشت FALSE not Exists دوم می‌گردد، بنابراین سطر دوم جدول قایق یعنی قایق b2 نیز در خروجی دستور select جلوی not exists اول ظاهر نمی‌شود. چون قایق b2 توسط قایقران s1 رزرو شده است.

نتیجه این که قایقران s1 همه قایقهای موجود در جدول قایق را رزرو کرده است.
حال یکبار دیگر پرس و جوی مطرح شده در فرم دوم را در نظر بگیرید:

```
SELECT Sname
FROM Sailors S
```

WHERE NOT EXISTS (SELECT bid

FROM Boats B

WHERE NOT EXISTS (SELECT *

FROM Reservation R

WHERE R.bid = B.bid

AND R.sid = S.sid))

در نهایت خروجی قطعه پرس و جوی زیر:

SELECT bid

FROM Boats B

WHERE NOT EXISTS (...

برای قایقران s1 موجود در سطر اول جدول قایقران برابر مقدار تهی می‌شود. چون به دلیل شرایط جلوی not exists دوم مشاهده نمودید که هیچ یک از سطرهای جدول قایق در خروجی دستور select جلوی اول ظاهر نشدنند، زیرا در خروجی bid جلوی not exists select bid اول، قایقهای رزرو شده، توسط یک قایقران خاص، خارج می‌گردند، از آنجا که پس از بررسی قایقهای رزرو شده برای قایقران s1

توسط دستور select در جلوی not exists دوم، مشخص شد که قایقران ۱، همهی قایق‌ها را رزرو کرده است، بنابراین خروجی select bid جلوی not exists اول، تهی خواهد شد.

بنابر مطلب فوق شرط where در پشت not exists اول برای قایقران s1 برابر مقدار TRUE خواهد بود، پس سطر اول جدول قایقران انتخاب می‌گردد و sname با مقدار sn1 در خروجی قرار می‌گیرد.

توجه: در یک بیان ساده اگر قایقی برای یک قایقران خاص رزرسدیده باشد در خروجی قطعه پرس و جوی فوق قرار نمی‌گیرد. رزرو یک قایق توسعه یک قایقران خاص توسط دستور select در جلوی not exists دوم بررسی می‌گردد.

حال سطر دوم از جدول تولیدکننده بررسی می‌شود، تا بررسی کامل در همین سطر توقف کنید.

به قطعه کد زیر موجود در جلوی not exists اول، دقت کنید:

```
NOT EXISTS ((SELECT bid
              FROM Boats B
              WHERE
```

ابتدا سطر اول از جدول قایق یعنی قایق b1 بررسی می‌شود.

با شرط زیر بررسی می‌شود که قایقران ۲، قایق b1 را رزرو کرده است یا خیر.

$$\begin{array}{ll} R.bid = B.bid & R.sid = S.sid \\ \hline b1 & b1 \\ & s2 \\ \hline & \text{TRUE} \\ & \text{TRUE} \\ \hline & \text{TRUE} \end{array}$$

شرط برقرار است، زیرا قایق ۱ توسط قایقران ۲ رزرو شده است، بنابراین خروجی فعلی قطعه پرس و جوی زیر:

NOT EXISTS (SELECT *

```
              FROM Reservation R
              WHERE R.bid = B.bid
                  AND R.sid = S.sid))
```

برابر مقدار b1 (سطر ۱ به دلیل استفاده از عملگر *) می‌باشد.

چون جلوی not exists دوم غیر تهی است، بنابراین شرط where موجود پشت not exists دوم می‌گردد، بنابراین سطر اول جدول قایق یعنی قایق b1 در خروجی دستور جلوی not exists اول ظاهر نمی‌شود. چون قایق b1 توسط قایقران ۲ رزرو شده است.

در ادامه سطر دوم از جدول قایق یعنی قایق ۲، همچنان برای قایقران ۲ بررسی می‌شود.

با شرط زیر بررسی می‌شود که قایقران ۲، قایق b2 را رزرو کرده است یا خیر.

$$\begin{array}{ll} R.bid = B.bid & R.sid = S.sid \\ \hline \times & b2 \\ & \times \\ \hline \text{FALSE} & \text{FALSE} \\ \hline & \text{FALSE} \end{array}$$

شرط برقرار نیست، زیرا قایق ۲ توسط قایقران ۲ رزرو نشده است، بنابراین خروجی فعلی قطعه پرس و جوی زیر:

NOT EXISTS (SELECT *

```
FROM Reservation R
WHERE R.bid = B.bid
AND R.sid = S.sid))
```

برابر مقدار تهی می باشد.

چون جلوی not exists دوم تهی است، بنابراین شرط where موجود پشت TRUE not exists دوم اول می گردد، بنابراین سطر دوم جدول قایق یعنی قایق b2 در خروجی دستور select جلوی not exists ظاهر می شود. چون قایق b2 توسط قایقران s2 رزرو شده است. نتیجه این که قایقران s2 همه قایقهای موجود در جدول قایق را رزرو نکرده است. حال یکبار دیگر پرس و جوی مطرح شده در فرم دوم را در نظر بگیرید:

```
SELECT Sname
FROM Sailors S
WHERE NOT EXISTS (SELECT bid
                   FROM Boats B
                   WHERE NOT EXISTS (SELECT *
                                      FROM Reservation R
                                      WHERE R.bid = B.bid
                                      AND R.sid = S.sid))
```

در نهایت خروجی قطعه پرس و جوی زیر:

```
SELECT bid
FROM Boats B
WHERE NOT EXISTS (...)
```

برای قایقران s2 موجود در سطر دوم جدول قایقران برابر مقدار غیرتهی و برابر مقدار b2 می شود. چون به دلیل شرایط جلوی not exists دوم مشاهده نمودید که سطر دوم از جدول قایق در خروجی دستور select جلوی not exists اول ظاهر شد، زیرا در خروجی select bid جلوی not exists اول، قایقهای رزرو شده، توسط یک قایقران خاص، خارج می گرددند، از آنجا که پس از بررسی قایقهای رزرو شده برای قایقران s2 توسط دستور select در جلوی not exists دوم، مشخص شد که قایقران b2 قایق b2 را رزرو نکرده است، بنابراین خروجی bid جلوی not exists اول، غیرتهی خواهد شد.

بنابر مطالب فوق شرط where در پشت not exists اول برای قایقران s2 برابر مقدار FALSE خواهد بود، پس سطر دوم جدول قایقران انتخاب نمی گردد و sname با مقدار s2 در خروجی قرار نمی گیرد.

در نهایت سطر سوم از جدول قایقران بررسی می شود، که نتیجه های همچون سطر دوم از جدول قایقران خواهد داشت.

نتیجه اینکه، پرس و جوی فرم دوم نام قایقران هایی را می دهد که قایقی وجود ندارد که آنها آنرا رزرو نکرده باشند. به عبارت دیگر نام قایقران هایی را می دهد که همه قایقهای موجود در جدول قایق را رزرو کرده اند.

بررسی گزینه دوم:

مطابق پرس و جوی مطرح شده در گزینه دوم داریم:
 «نام قایقران هایی که حداقل یک قطعه را تولید نکرده اند.»
 در ادامه، این پرس و جو را به دو فرم زیر شبیه سازی می کنیم:

فرم اول:

```
SELECT Sname
FROM Sailors S
WHERE EXISTS ((SELECT bid
                FROM Boats B)
              EXCEPT
              (SELECT bid
                FROM Reservation R
                WHERE R.sid = S.sid)))
```

توجه: گزینه دوم خطای نحوی دارد و به دلیل داشتن دستور اضافه R.bid = B.bid از سوی کامپایلر کامپایل نخواهد شد. دقت کنید که جدول Boats در پرس و جوی بعد از دستور except قابل شناخت نیست و به تبع حضور دستور R.bid = B.bid در پرس و جوی بعد از دستور except از سوی کامپایلر خطای کامپایلری محسوب می‌گردد، فرم صحیح گزینه دوم به شکل فرم اول مطرح شده می‌باشد. بنابراین گزینه دوم پاسخ سوال نمی‌باشد.

فرم دوم:

```
SELECT Sname
FROM Sailors S
WHERE EXISTS (SELECT bid
                FROM Boats B
                WHERE NOT EXISTS (SELECT *
                                  FROM Reservation R
                                  WHERE R.bid = B.bid
                                  AND R.sid = S.sid))
```

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول قایقران، یک بار به طور کامل از ابتدا تا انتهای جدول قایق بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول قایق وجود دارد که شرط اتصال select R.bid = B.bid AND R.sid = S.sid موجود در داخلی را برقرار نکند، اگر وجود داشته باشد، سطر مورد نظر از جدول قایقران در خروجی نمایش داده می‌شود.

بررسی گزینه چهارم:

مطابق پرس و جوی مطرح شده در گزینه چهارم داریم:
«نام قایقران‌هایی که حداقل یک قطعه را تولید کرده‌اند.»
در ادامه، این پرس و جو را به چهار فرم زیر می‌نویسیم:

فرم اول:

```
SELECT distinct (S.sname)
FROM Sailors S, Reservation R
WHERE S.sid = R.sid
```

فرم دوم:

```
SELECT S.sname
FROM Sailors S
WHERE sid IN (SELECT sid
               FROM Reservation R)
```

ابتدا زیر پرس‌وجوی داخلی یک بار و برای همیشه اجرا می‌گردد، سپس پرس‌وجوی خارجی به ازای حرکت در هر یک از سطرهای خود، از مقادیر زیر پرس‌وجوی داخلی استفاده می‌کند.
در پرس‌وجوی فوق به ازای حرکت در هر سطر از جدول قایقران مجموعه جلوی *in* بررسی می‌گردد. آیا sid داخل این مجموعه قرار دارد یا خیر. اگر قرار داشت سطر مورد نظر از جدول قایقران در خروجی نمایش داده می‌شود.

فرم سوم:

```
SELECT S.sname
FROM Sailors S
WHERE EXISTS (SELECT bid
               FROM Reservation R
               WHERE S.sid = R.sid)
```

به ازای حرکت در هر یک از سطرهای پرس‌وجوی خارجی، یک بار به طور کامل از ابتدا تا انتهای زیر پرس‌وجوی داخلی اجرا و براساس شرطی که زیر پرس‌وجوی داخلی را به پرس‌وجوی خارجی متصل می‌کند، بررسی انجام می‌شود. مانند دو حلقه تو در تو، که به ازای هر بار اجرای حلقه خارجی، یک بار به طور کامل حلقه داخلی اجرا می‌گردد.

در پرس‌وجوی فوق به ازای حرکت در هر سطر از جدول قایقران، یک بار به طور کامل از ابتدا تا انتهای جدول رزرو بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول رزرو وجود دارد که شرط اتصال select S.sid = R.sid موجود در select داخلی را برقرار کند، اگر وجود داشته باشد، سطر مورد نظر از جدول قایقران در خروجی نمایش داده می‌شود.

فرم چهارم:

```
SELECT Sname
FROM Sailors S
WHERE EXISTS (SELECT bid
               FROM Boats B
               WHERE EXISTS (SELECT *
                             FROM Reservation R
                             WHERE R.bid = B.bid AND R.sid = S.sid))
```

در پرس‌وجوی فوق به ازای حرکت در هر سطر از جدول قایقران، یک بار به طور کامل از ابتدا تا انتهای جدول قایق بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول قایق وجود دارد که شرط اتصال select R.bid = B.bid AND R.sid = S.sid موجود در select داخلی را برقرار کند، اگر وجود داشته باشد، سطر مورد نظر از جدول قایقران در خروجی نمایش داده می‌شود.

توجه: فرم چهارم مطرح شده، مطابق گزینه چهارم می‌باشد، که مطابق پرس و جوی مطرح شده در صورت سوال نمی‌باشد، بنابراین پاسخ سوال نیز نمی‌باشد.

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

راه حل دوم:

پرس و جوی مطرح شده در صورت سوال را در نظر بگیرید:
«نام قایقران‌هایی که همه قایق‌ها را رزرو کرده‌اند».

توجه: عملگر تقسیم در جبر رابطه‌ای توسط الگوی اول در SQL قابل پیاده‌سازی است. عملگر تقسیم در جبر رابطه‌ای زمانی مورد استفاده قرار می‌گیرد که پرس و جو بخواهد همه حالت‌های یک اتفاق را بررسی کند. مانند نام قایقران‌هایی که همه قایق‌ها را رزرو کرده‌اند. که رزرو کردن قایق‌ها توسط قایقران‌ها، اتفاق و مقسم است و همه قایق‌ها، حالت‌ها و مقسم علیه است.

توجه: این پرس و جو به فرم زیر توسط الگوی اول (همه) در SQL قابل پیاده‌سازی است.

الگوی اول (۰۰):

```
SELECT S.sname
FROM Sailor S
WHERE NOT EXISTS (SELECT bid
                   FROM Boats B
                   WHERE NOT EXISTS (SELECT *
                                     FROM Reserves R
                                     WHERE S.sid = R.sid
                                     AND B.bid = R.bid))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«نام قایقران‌هایی که، وجود نداشته باشد (NOT EXISTS) قایقی که، رزرو نشده باشد (NOT EXISTS). یعنی نام قایقران‌هایی که همه قایق‌ها را رزرو کرده‌اند.»

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول Sailor، یک بار به طور کامل از ابتدا تا انتهای جدول Boats بررسی می‌گردد، تا مشخص گردد آیا هیچ سطری در جدول Boats وجود ندارد که شرط اتصال موجود در select داخلي را برقرار نکند، اگر وجود نداشته باشد، سطر مورد نظر از جدول Sailor در خروجی نمایش داده می‌شود.

توجه: شرط لازم برای تبعیت از الگوی اول وجود NOT EXISTS اول و NOT EXISTS دوم است و شرط کافی برای تبعیت از الگوی اول وجود شروط اتصال متناسب و سازگار است.

۶۳- گزینه (۱) صحیح است.

سه جدول ST، CO و STCO با مقادیر زیر را در نظر بگیرید:

<u>STID</u>	<u>STNAME</u>	<u>COID</u>	<u>CONAME</u>	CREDIT	
S1	Sn1			C1	Cn1	2	
S2	Sn2			C2	Cn2	3	
S3	Sn3			C3	Cn3	4	
S4	Sn4						

جدول ST

جدول CO

<u>STID</u>	<u>COID</u>	SEMESTER	GRADE
S1	C1	95-96-1	20
S1	C2	95-96-1	15
S1	C3	92-92-1	17
S2	C1	95-96-1	19
S2	C2	95-96-1	18
S2	C3	93-93-1	12
S3	C1	94-94-1	16

جدول STCO

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

```
Select STName, SUM(Grade*Credit)/SUM(Credit)
From ST JOIN STCO JOIN CO
Group By STID,STName,Semester
Having Semester='95-96-1'
```

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

```
Select STName, SUM(Grade*Credit)/SUM(Credit)
From ST JOIN STCO
on ST.STID=STCO.STID JOIN CO on CO.COID=STCO.COID
Group By ST.STID,STName,Semester
Having Semester='95-96-1'
```

توجه: در فرم اصلاح شده‌ی گزینه‌ی اول، شرط اتصال و عملگر ON و پشت ستون STID جدول ST قرار گرفت. پرس و جوی گزینه‌ی اول دو خطای نحوی داشت که اصلاح شد.
با توجه به جداول فوق، خروجی پرس و جوی گزینه‌ی اول پس از انجام عملگر JOIN به صورت زیر است:

STID	STNAME	STID	COID	SEMESTER	GRADE	COID	CONAME	CREDIT
S1	Sn1	S1	C1	95-96-1	20	C1	Cn1	2
S1	Sn1	S1	C2	95-96-1	15	C2	Cn2	3
S1	Sn1	S1	C3	92-92-1	17	C3	Cn3	4
S2	Sn2	S2	C1	95-96-1	19	C1	Cn1	2
S2	Sn2	S2	C2	95-96-1	18	C2	Cn2	3
S2	Sn2	S2	C3	93-93-1	12	C3	Cn3	4
S3	Sn3	S3	C1	94-94-1	16	C1	Cn1	2

همچنین در ادامه، پس از انجام دستور Group By ST.STID,STName,Semester براساس ستون‌های خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:

$$\begin{array}{ll}
 \begin{array}{l}
 S1, Sn1, 95-96-1 \\
 C1 \ 20 \ 2 \dots \\
 C2 \ 15 \ 3 \dots \\
 \hline
 \text{گروه اول}
 \end{array} &
 \begin{array}{l}
 S1, Sn1, 92-92-1 \\
 C3 \ 17 \ 4 \dots \\
 \hline
 \text{گروه دوم}
 \end{array} \\
 \hline
 \begin{array}{l}
 S2, Sn2, 95-96-1 \\
 C1 \ 19 \ 2 \dots \\
 C2 \ 18 \ 3 \dots \\
 \hline
 \text{گروه سوم}
 \end{array} &
 \begin{array}{l}
 S2, Sn2, 93-93-1 \\
 C3 \ 12 \ 4 \dots \\
 \hline
 \text{گروه چهارم}
 \end{array} \\
 \hline
 \begin{array}{l}
 S3, Sn3, 94-94-1 \\
 C1 \ 16 \ 2 \dots \\
 \hline
 \text{گروه پنجم}
 \end{array} &
 \end{array}$$

و در نهایت دستور 'Having Semester='95-96-1' Having Semester هر گروه به طور مستقل اعمال می‌گردد.
توجه: دستور HAVING بر روی گروه‌ها، اعمال می‌گردد.

$$\begin{array}{r} S1, Sn1, 95-96-1 \\ C1 \quad 20 \quad 2 \dots \\ C2 \quad 15 \quad 3 \dots \\ \hline \text{گروه اول} \end{array}$$

$$\begin{array}{r} S1, Sn1, 92-92-1 \\ C3 \quad 17 \quad 4 \dots \\ \hline \text{گروه دوم} \end{array}$$

Semester = '95-96-1'

Semester = '95-96-1'

$$\begin{array}{r} S2, Sn2, 95-96-1 \\ C1 \quad 19 \quad 2 \dots \\ C2 \quad 18 \quad 3 \dots \\ \hline \text{گروه سوم} \end{array}$$

$$\begin{array}{r} S2, Sn2, 93-93-1 \\ C3 \quad 12 \quad 4 \dots \\ \hline \text{گروه چهارم} \end{array}$$

Semester = '95-96-1'

Semester = '95-96-1'

$$\begin{array}{r} S3, Sn3, 94-94-1 \\ C1 \quad 16 \quad 2 \dots \\ \hline \text{گروه پنجم} \end{array}$$

Semester = '95-96-1'

توجه: با توجه به شرط انتخاب گروه توسط دستور 'Having Semester='95-96-1', فقط گروه‌های اول و سوم جهت نمایش در خروجی انتخاب می‌شوند.

و در نهایت دستور SELECT SUM(Grade*Credit)/SUM(Credit) داخل دستور برای هر گروه انتخاب شده توسط دستور Having به طور مستقل محاسبه می‌گردد و در خروجی پرس و جو قرار می‌گیرد.
معدل وزنی نمرات دروس اخذ شده گروه اول یعنی دانشجو Sn1 به صورت زیر محاسبه می‌گردد:

$$\frac{(20 \times 2) + (15 \times 3)}{5} = \frac{(40) + (45)}{5} = \frac{85}{5} = 17.00$$

معدل وزنی نمرات دروس اخذ شده گروه سوم یعنی دانشجو Sn2 به صورت زیر محاسبه می‌گردد:

$$\frac{(19 \times 2) + (18 \times 3)}{5} = \frac{(38) + (54)}{5} = \frac{92}{5} = 18.40$$

بنابراین خروجی نهایی پرس و جوی گزینه‌ی اول به صورت زیر است:

STNAME	(No column name)
Sn1	17.00
Sn2	18.40

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

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی لیست نام دانشجویان به همراه معدل وزنی نمرات دروس اخذ شده آن‌ها را در ترم 95-96-1.

توجه: گزینه‌ی اول را می‌توان بدون استفاده از دستور JOIN و به شکل کلاسیک آن یعنی با استفاده از ضرب دکارتی و شرط اتصال برابری نیز بازنویسی کرد.
فرم کلاسیک گزینه‌ی اول:

```
Select STName, SUM(Grade*Credit)/SUM(Credit)
From ST , STCO, CO
Where ST.STID=STCO.STID AND
STCO.COID=CO.COID
Group By ST.STID,ST.STName,Semester
Having Semester='95-96-1'
```

بنابراین خروجی نهایی فرم کلاسیک پرس و جوی گزینه‌ی اول نیز به صورت زیر است:

STName	(No column name)
Sn1	17.00
Sn2	18.40

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

```
Select STName, SUM(Grade*Credit)/SUM(Credit)
From ST JOIN STCO JOIN CO
Where Semester='95-96-1'
Group By STID,Semester
```

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

```
Select STName, SUM(Grade*Credit)/SUM(Credit)
From ST JOIN STCO
on ST.STID=STCO.STID JOIN CO on CO.COID=STCO.COID
Where Semester='95-96-1'
Group By STID,STName,Semester
```

توجه: در فرم اصلاح شده‌ی گزینه‌ی چهارم، شرط اتصال و عملگر ON و پشت ستون STID جدول ST قرار گرفت. پرس و جوی گزینه‌ی چهارم تا به اینجا دو خطای نحوی داشت که اصلاح شد. اما یک خطای نحوی پرنگتری هم دارد، اینکه همواره به غیر از توابع آماری، همه ستون‌های جلوی دستور Select باید زیرمجموعه یا مساوی ستون‌های دستور By باشد، بنابراین در فرم اصلاح شده‌ی گزینه‌ی چهارم، ستون STName جلوی دستور By قرار گرفت. نتیجه اینکه گزینه‌ی چهارم سه خطای نحوی داشت که اصلاح شد.

توجه: اگر به فرم‌های اصلاح شده گزینه اول و چهارم باشد، آنگاه هر دو گزینه‌ی اول و چهارم می‌توانند پاسخ سوال باشند.

توجه: گزینه‌ی اول، دو خطای نحوی دارد و گزینه‌ی چهارم، سه خطای نحوی دارد، اما طراح محترم گزینه‌ی اول را در کلید اولیه و نهایی اعلام کرده است. درحالی که هر دو گزینه‌ی اول و چهارم خطای نحوی دارد. اگر ملاک سنجش خطای کمتر است، آنگاه گزینه‌ی اول پاسخ سوال است. و اگر ملاک سنجش پاسخ واقعاً درست است، این سوال پاسخ ندارد و باید سوال حذف می‌شد. و اگر ملاک سنجش عدالت است، آنگاه گزینه‌ی اول و چهارم باید پاسخ می‌بود. بنابراین اگر منظور طراح خطای کمتر باشد، آنگاه

گزینه‌ی اول پاسخ سؤال خواهد بود. همانطور که گفته‌یم سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه‌ی اول را به عنوان پاسخ اعلام کرده بود. که کار نادرستی بوده است. با توجه به جداول فوق، خروجی پرس و جوی گزینه‌ی چهارم پس از انجام عملگر JOIN به صورت زیر است:

STID	STNAME	STID	COID	SEMESTER	GRADE	COID	CONAME	CREDIT
S1	Sn1	S1	C1	95-96-1	20	C1	Cn1	2
S1	Sn1	S1	C2	95-96-1	15	C2	Cn2	3
S1	Sn1	S1	C3	92-92-1	17	C3	Cn3	4
S2	Sn2	S2	C1	95-96-1	19	C1	Cn1	2
S2	Sn2	S2	C2	95-96-1	18	C2	Cn2	3
S2	Sn2	S2	C3	93-93-1	12	C3	Cn3	4
S3	Sn3	S3	C1	94-94-1	16	C1	Cn1	2

در ادامه، پس از انجام دستور 'Where Semester='95-96-1' فقط سطرهایی که شرط Semester='95-96-1' را برقرار می‌کنند، انتخاب می‌شوند، به صورت زیر:

STID	STNAME	STID	COID	SEMESTER	GRADE	COID	CONAME	CREDIT
S1	Sn1	S1	C1	95-96-1	20	C1	Cn1	2
S1	Sn1	S1	C2	95-96-1	15	C2	Cn2	3
S2	Sn2	S2	C1	95-96-1	19	C1	Cn1	2
S2	Sn2	S2	C2	95-96-1	18	C2	Cn2	3

همچنین در ادامه، پس از انجام دستور Group By ST.STID,STName,Semester براساس ستون‌های ST.STID,STName,Semester خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:

ST1,Sn1,95-96-1	S2,Sn2,95-96-1
C1 20 2 ...	C1 19 2 ...
C2 15 3	C2 18 3
گروه اول	
گروه دوم	

و در نهایت دستور SELECT SUM(Grade*Credit)/SUM(Credit) داخل دستور برای هر گروه به طور مستقل محاسبه می‌گردد و در خروجی پرس و جو قرار می‌گیرد.

معدل وزنی نمرات دروس اخذ شده گروه اول یعنی دانشجو Sn1 به صورت زیر محاسبه می‌گردد:

$$\frac{(20 \times 2) + (15 \times 3)}{5} = \frac{(40) + (45)}{5} = \frac{85}{5} = 17.00$$

معدل وزنی نمرات دروس اخذ شده گروه سوم یعنی دانشجو Sn2 به صورت زیر محاسبه می‌گردد:

$$\frac{(19 \times 2) + (18 \times 3)}{5} = \frac{(38) + (54)}{5} = \frac{92}{5} = 18.40$$

بنابراین خروجی نهایی پرس و جوی گزینه‌ی چهارم به صورت زیر است:

STNAME	(No column name)
Sn1	17.00
Sn2	18.40

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

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی لیست نام دانشجویان به همراه معدل وزنی نمرات دروس اخذ شده آن‌ها را در ترم ۹۵-۹۶-۱.

توجه: گزینه‌ی چهارم را می‌توان بدون استفاده از دستور JOIN و به شکل کلاسیک آن یعنی با استفاده از ضرب دکارتی و شرط اتصال برابری نیز بازنویسی کرد.

فرم کلاسیک گزینه‌ی چهارم:

```
Select STName, SUM(grade*credit)/SUM(credit)
From ST , STCO, CO
Where ST.STID=STCO.STID and
STCO.COID=CO.COID and Semester='95-96-1'
Ggroup By ST.STID,STName,Semester
```

بنابراین خروجی نهایی فرم کلاسیک پرس و جوی گزینه‌ی چهارم نیز به صورت زیر است:

STNAME	(No column name)
Sn1	17.00
Sn2	18.40

۶۴- گزینه (۴) صحیح است.

سه جدول Student و Course 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		
	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 ۱۵ می‌شود، به صورت زیر:

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

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

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

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

S1	S2	S3
C1 10	C1 14	C1 18
C2 12	C2 16	C2 20
گروه اول	گروه دوم	گروه سوم

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

توجه: دستور HAVING بر روی گروه‌ها، اعمال می‌گردد.

S1	S2	S3
C1 10	C1 14	C1 18
C2 12	C2 16	C2 20
گروه اول	گروه دوم	گروه سوم

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

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

گروه سوم جهت نمایش در خروجی انتخاب می‌شود.

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

نهایی زیر پرس و جوی داخلی به صورت زیر است:

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

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

بنابراین در ادامه پرس و جوی زیر را خواهیم داشت:

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

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

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

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

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

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

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

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

برای هر سطر از جدول Student شرط جلوی Exists که حاصل یک مقایسه می‌باشد، محاسبه می‌گردد، اگر غیرتھی بود، شرط جلوی where همان TRUE است، Exists می‌گردد و سطر مورد نظر از جدول Student انتخاب می‌گردد و این رویه برای تک تک سطرهای جدول Student، تا به انتهای جدول Student ادامه پیدا می‌کند. به بیان دیگر این پرس و جو نام دانشجویانی را می‌دهد که پرانتز مقابل Exists برای آنها غیرتھی است. این پرانتز هنگامی غیرتھی می‌شود که حاصل مقایسه بیان شده در این پرانتز غیرتھی شود. حاصل این مقایسه در صورتی غیرتھی 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						

<u>Sid</u>	<u>SName</u>	<u>Sid</u>	<u>Cid</u>	<u>grade</u>	<u>Cid</u>	<u>CName</u>
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		
		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:

```
S1
C1 10
C2 12
```

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

<u>Sid</u>	<u>SName</u>	<u>Sid</u>	<u>Cid</u>	grade	<u>Cid</u>	<u>CName</u>
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 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 Having AVG(grade) > (Select AVG(grade) From S_C) اعمال می‌گردد.
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)

S2 جهت نمایش در خروجی انتخاب نمی‌شود.

جلوی Exists برابر تهی گردید، بنابراین شرط where در پشت exists 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, S_C.Sid = S_C.Sid داریم:

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				

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:

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 Having AVG(grade) > (Select 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 غیرتهی گردید، بنابراین شرط where در پشت exists 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:

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

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

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

با توجه به وجود دستور `in`، زیر پرس و جوی داخلی فوق یک Normal Subquery است، یعنی ابتدا زیر پرس و جوی داخلی یک بار و برای همیشه اجرا می‌گردد، سپس پرس و جوی خارجی به ازای حرکت در هر یک از سطرهای 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 `Sid`‌های مجموعه جلوی `in` بود، آنگاه شرط جلوی `where` که همان 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 جوی نام دانشجویانی از جدول Student را می‌دهد که در پرانتز مقابل `in` قرار دارند. به عبارت دیگر پرس و جوی گزینه سوم نام دانشجویانی که معدل آنها از میانگین نمرات همه دروس دانشگاه بیشتر است را استخراج می‌کند. که مطابق پرس و جوی مطرح شده در صورت سوال است.

توجه: همان‌طور که مشاهده می‌کنید، معدل دانشجو `Sn3` از میانگین نمرات همه دروس دانشگاه بیشتر است.
با توجه به جداول فوق، ابتدا خروجی داخلی ترین 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 ۱۵ می‌شود، به صورت زیر:

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

همچنان در ادامه با توجه به جداول فوق، خروجی زیر پرس و جوی داخلی در جدول 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

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

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

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

توجه: دستور HAVING بر روی گروهها، اعمال می‌گردد.

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

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

توجه: با توجه به شرط انتخاب گروه توسط دستور Having AVG(grade) > (Select AVG(grade)) From S_C فقط گروه سوم جهت نمایش در خروجی انتخاب می‌شود.

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

Sid
s3

توجه: دستور GROUP BY، سرگروه‌ها را، راهی خروجی می‌کند.
بنابراین در ادامه پرس و جوی زیر را خواهیم داشت:

Select SName
From Student
Where Sid in (S3)

همانطور که گفته‌یم برای هر سطر از جدول Student کل مجموعه‌ی جلوی in که حاصل یک مقایسه می‌باشد، بررسی می‌گردد، اگر Sid موجود در هر سطر، داخلSid های 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 سطراهای جدول Student، تا به انتهای جدول 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:

SName
sn3

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

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

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

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

۶۵- گزینه (۱ و ۴) صحیح است.

عملگر الحق خارجی چپ در جبر رابطه‌ای

این عملگر، مانند الحق طبیعی، ستون‌های مشترک را فقط یکبار در خروجی قرار می‌دهد. همچنین کلیه سطراهای پیوندپذیر را در خروجی قرار می‌دهد. اما علاوه بر آن کلیه سطراهای 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 الحق خارجی چپ در SQL

این عملگر، ستون‌های مشترک را دوبار در خروجی قرار می‌دهد. همچنین کلیه سطراهای پیوندپذیر را در خروجی قرار می‌دهد. اما علاوه بر آن کلیه سطراهای پیوندناپذیر جدول سمت چپ را نیز در خروجی قرار

می‌دهد و در این حالت تمام ستون‌های جدول سمت راست مقدار NULL قرار می‌دهد.

توجه: به تفاوت عملگر الحق خارجی چپ در جبر رابطه‌ای و SQL دقต کنید.

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

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

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

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

select *

from student left outer join studentCourse

که البته پرس و جوی مطرح شده در صورت سوال 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 شده پرس و جوی مطرح شده در صورت سوال، شرط اتصال و عملگر ON قرار گرفت. پرس و جوی مطرح شده در صورت سوال یک خطای نحوی داشت که اصلاح شد.
با توجه به جداول فوق، خروجی پرس و جوی مطرح شده در صورت سوال در حالت **حداقل** پس از انجام عملگر Join به صورت زیر است:

<u>Stid</u>	<u>StName</u>	<u>Stid</u>	<u>Cid</u>	grade
s1	sn1	s1	c1	10
s2	sn2	NULL	NULL	NULL
s3	sn3	NULL	NULL	NULL

بنابراین رابطه زیر برقرار خواهد بود:

Cardinality_{min}(student left outer join studentCourse) = K = 3

توجه: دقیت 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>	<u>StName</u>	...	<u>Stid</u>	<u>Cid</u>	grade	<u>Cid</u>	<u>CName</u>
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:

StdId	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 می شد، دلیل اول وجود خطای کامپایلری در صورت سوال و دلیل دوم وجود گزینه صحیح در گزینه های صورت سوال.

۶۶ - گزینه (۴) صحیح است.

CROSS JOIN (ضرب دکارتی)

```
select *
```

فرم اول:

from Student **CROSS JOIN** StudentCourse

فرم دوم:

```
select *
```

from Student, StudentCourse

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

برای فرم حداقل، جداول زیر را در نظر بگیرید:

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

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

خروجی پرس و جوی مطرح شده در صورت سوال در حالت حداقل به صورت زیر است:
select distinct Student.*
from student,studentCourse

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

بنابراین رابطه زیر برقرار خواهد بود:

Cardinality_{min}(student, 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 مقابله ممکن برقرار خواهد بود:

۶۷- گزینه (۳) صحیح است.

صورت سوال به این شکل است:

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

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

گزاره اول درست است، زیرا در حالت کلی، اگر رابطه R، دارای n خصیصه باشد، آنگاه تعداد ابرکلیدهای آن حداقل یک و حداقل $1 - 2^n$ 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 حداقل 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 حداقل $1 - 2^n$ ابرکلید، حالت خاص جدول

تمام کلید هم پوشش می دهد چون بیان حداکثر -1^{n-2} مقادیر کوچکتر و برابر خودش را پوشش می دهد. اگر در یک رابطه با n خصیصه، تک تک خصیصه ها به تنها یک کلید کاندید باشد، آنگاه رابطه دارای n کلید کاندید است. بنابراین هر زیرمجموعه غیر تهی از خصیصه های این رابطه یک ابرکلید است. که در این حالت تعداد ابرکلیدهای یک رابطه با n خصیصه برابر با -1^{n-2} است که بیشترین مقدار ممکن در تعداد ابرکلیدهای یک رابطه با n خصیصه است. ابرکلید بدون صفت نداریم، بنابراین حالت $\binom{n}{0}$ اضافه است. همانطور که گفتیم در حالت کلی، یک رابطه دارای n خصیصه، شرایط مختلفی را در تعداد ابرکلید می تواند تجربه کند، که حداکثر تعداد ابرکلیدهایی که می تواند تجربه کند برابر $\binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n} = -1^{n-2}$ است. در واقع عبارت زیر برقرار است:

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

(b) صفت مرکب، صفتی است که مقدار آن از مقدار سایر صفت‌ها محاسبه می‌شود. گزاره دوم نادرست است، زیرا صفت مشتق (پویا) صفتی است که مقدار آن از مقدار سایر صفت‌ها محاسبه می‌شود. صفت مشتق (پویا) صفتی است که در موجودیت وجود خارجی ندارد ولی در صورت لزوم می‌توان آنرا بدست آورد. صفتی که مقادیر آن مدام در حال تغییر و تحول باشد، صفت پویا یا مشتق محسوب می‌گردد. بنابراین به دلیل تغییرات مداوم، توصیه می‌گردد صفت پویا در جداول بانک اطلاعات مورد استفاده قرار نگیرد و مقدار آن از طریق صفت مرتبط با آن محاسبه گردد. برای مثال برای محاسبه صفت سن، می‌توان صفت تاریخ تولد را در نظر گرفت و از روی این صفت، سن را محاسبه نمود. همچنین صفت ساده صفتی است که مقدار آن از لحظه معنایی ساده یا اتومیک یا تجزیه شدنی باشد، به این معنا که اگر مقدار آن را به اجزایی تجزیه کنیم، مقادیر جزیی 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 کد پستی تشکیل شده است.

۴) تعداد تاپل‌های عبارت $\Pi_{R,\alpha}(R \times T)$ همواره برابر با تعداد سطرهای R است.

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

<u>a</u>	b	<u>c</u>	d	e
1	2			
3	4			

پرس و جوی مقابل را در نظر بگیرید:

خروجی پرس و جوی پرانتز داخلی به صورت زیر است:

$$\begin{array}{c|c} \underline{a} & b \\ \hline 1 & 2 \\ 3 & 4 \end{array} \times \begin{array}{c|c|c} \underline{c} & d & e \end{array} = \begin{array}{c|c|c|c|c} \underline{a} & b & \underline{c} & d & e \end{array}$$

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

$$\frac{a}{\text{نهی}}$$

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

همچنین پرس و جوی زیر را در نظر بگیرید:

Select distinct R.a

From R

خروجی پرس و جو به صورت زیر است:

$$\frac{a}{1 \\ 3}$$

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

بنابراین رابطه مقابله ممکن برقرار خواهد بود:

$$\sigma_c(\Pi_\alpha(R)) = \Pi_\alpha(\sigma_c(R)) \quad \text{(d)}$$

گزاره چهارم نادرست است، زیرا دو عملگر σ و Π به صورت مشروط دارای خاصیت جابه‌جایی هستند. به طور کلی اگر R یک رابطه، α زیرمجموعه‌ای از ستون‌ها و c مجموعه‌ای از شرط‌ها باشد، آنگاه تساوی زیر زمانی برقرار است که ستون‌های عملگر σ زیرمجموعه 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:

S#	Sname	City
S1	Sn1	C1
S2	Sn2	C2
S3	Sn3	C2

$$R_2 = \Pi_{S\#, \text{City}} (\sigma_{\text{City}='C2'}(S)) \quad \text{پرس و جوی مقابله را در نظر بگیرید:}$$

خروجی پرس و جوی پرانتز داخلی به صورت زیر است:

S#	Sname	City
S2	Sn2	C2
S3	Sn3	C2

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

S#	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 پرانتز داخلی به صورت زیر است:

S#	City
S1	C1
S2	C2
S3	C2

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

S#	City
S2	C2
S3	C2

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

پس رابطه مقابل برقرار است:

S#	Sname	City
S1	Sn1	C1
S2	Sn2	C2
S3	Sn3	C2

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

پرس و جوی مقابل را در نظر بگیرید:

خروجی پرس و جوی پرانتز داخلی به صورت زیر است:

S#	Sname	City
S2	Sn2	C2
S3	Sn3	C2

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

S#
S2
S3

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

همچنین پرس و جوی مقابل را در نظر بگیرید:

خروجی پرس و جوی پرانتز داخلی به صورت زیر است:

S#
S1
S2
S3

که در نهایت پس از حرکت به سمت خارج، انجام عملگر انتخاب امکان پذیر نخواهد بود، زیرا ستون‌های عملگر σ یعنی City زیر مجموعه ستون‌های عملگر Π نیست، یعنی $S\# \subsetneq City$. در واقع ستون City در پرانتز داخلی توسط عملگر Π انتخاب نشده است، بنابراین اجرای عملگر σ بر روی ستون City امکان‌پذیر نیست.

پس رابطه زیر برقرار است:

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

۶۸- گزینه (۱) صحیح است.

سه جدول USER، BOOK و BORROWING با مقادیر زیر را در نظر بگیرید:

<u>UID</u>	NAME	<u>BID</u>	TITLE	PUBLISHER	TOTALNUMBER
U1	Un1			B1	Tn1	Pub1	1
U2	Un2			B2	Tn2	Pub2	2
U3	Un3			B3	Tn3	Pub3	3
U4	Un4						
U5	UN5						

جدول USER

جدول BOOK

<u>UID</u>	<u>BID</u>	StartDate	EndDate
U1	B1	95-01-1	NULL
U2	B2	95-02-1	NULL
U3	B2	95-03-1	NULL
U4	B3	95-04-1	NULL
U5	B3	95-05-1	NULL

جدول BORROWING

مطابق پرس و جوی مطرح شده در گزاره‌ی a، داریم:

a) with tb1 as (

Select BORROWING.BID, COUNT(BORROWING.BID) as cnt

From BORROWING, BOOK

Where BOOK.BID = BORROWING.BID and EndDate is NULL

Group by BORROWING.BID)

Select tb1.BID

From tb1,BOOK

Where tb1.BID = BOOK.BID and BOOK.TotalNumber = cnt

توجه: دستور with کل پرس و جوی مقابلش را tb1 نام گذاری می‌کند. که در دوم به طور

خلاصه‌تر، ساده‌تر و خواناتر مورد استفاده قرار می‌گیرد.

با توجه به جداول فوق، خروجی بخش اول پرس و جوی گزاره‌ی a پس از انجام عملگر ضرب دکارتی

به صورت زیر است:

<u>UID</u>	<u>BID</u>	StartDate	EndDate	<u>BID</u>	TITLE	PUBLISHER	TOTALNUMBER
U1	B1	95-01-1	NULL	B1	Tn1	Pub1	1
U2	B2	95-02-1	NULL	B2	Tn2	Pub2	2
U3	B2	95-03-1	NULL	B2	Tn2	Pub2	2
U4	B3	95-04-1	NULL	B3	Tn3	Pub3	3
U5	B3	95-05-1	NULL	B3	Tn3	Pub3	3

همچنین در ادامه، پس از انجام دستور GROUP BY BORROWING.BID براساس ستون خروجی پرس و جو به صورت زیر گروهبندی می‌شود:

B1	B2
U1 95-01-1 Tn1 ...	U2 95-02-1 Tn2 ...
گروه اول	U3 95-03-1 Tn2 ... گروه دوم

B3
U4 95-04-1 Tn3 ...
U5 95-05-1 Tn3 ...
گروه سوم

در ادامه دستور SELECT COUNT(BORROWING.BID) داخل دستور GROUP BY هر گروه به طور مستقل محاسبه می‌گردد و در خروجی پرس و جو قرار می‌گیرد.
بنابراین خروجی نهایی بخش اول پرس و جوی گزاره‌ی a به صورت زیر است:

BID	cnt
B1	1
B2	2
B3	2

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

توجه: نتیجه اینکه خروجی بخش اول پرس و جوی گزاره‌ی a پس از انجام عملگر ضرب دکارتی و اجرای دستور Group By شامل دو ستون کد کتاب (BID) و تعداد نسخه‌های به امانت رفته کتاب است که توسط دستور with جدول tb1 نام‌گذاری شده است.
بخش دوم پرس و جوی گزاره‌ی a را نظر بگیرید:

```
Select tb1.BID
From tb1,BOOK
Where tb1.BID = BOOK.BID and BOOK.TotalNumber = cnt
```

با توجه به جداول فوق، خروجی بخش دوم پرس و جوی گزاره‌ی a پس از انجام عملگر ضرب دکارتی به صورت زیر است:

BID	cnt	BID	TITLE	PUBLISHER	TOTALNUMBER
B1	1	B1	Tn1	Pub1	1
B2	2	B2	Tn2	Pub2	2
B3	2	B3	Tn3	Pub3	3

توجه: پس از انجام عمل ضرب دکارتی مابین جدول tb1 و جدول BOOK، شرط اصلی پرس و جو اجرا می‌گردد که منجر به استخراج کد کتاب‌هایی می‌گردد که تعداد نسخه‌های به امانت رفته آنها برابر با تعداد کل نسخه‌های آن کتاب است. در واقع لیست تمام کتاب‌هایی را نمایش می‌هد که تمام نسخه‌های آنها امانت

گرفته شده است.

در ادامه پس از اجرای شرط اصلی پرس و جو یعنی $\text{BOOK.TOTALNUMBER} = \text{cnt}$ خروجی پرس و جو به صورت زیر خواهد بود:

BID	cnt	BID	TITLE	PUBLISHER	TOTALNUMBER
B1	1	B1	Tn1	Pub1	1
B2	2	B2	Tn2	Pub2	2

و در نهایت پس از اجرای دستور $\text{Select } \text{tbl.BID}$ خروجی پرس و جو به صورت زیر خواهد بود:

BID
B1
B2

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «لیست تمام کتاب‌هایی را نمایش می‌هد که تمام نسخه‌های آنها امانت گرفته شده است.»
مطابق پرس و جوی مطرح شده در گزاره b، داریم:

b) Select BID

From BOOK b1

Where (select COUNT (UID)

From BORROWING

Where $\text{BORROWING.BID} = \text{b1.BID}$ and

$\text{BORROWING.EndDate} \text{ is NULL} = \text{b1.TotalNumber}$

با توجه به وجود عملگر تساوی، زیر پرس و جوی داخلی فوق یک Correlated Subquery است، یعنی به ازای حرکت در هر یک از سطوح‌های 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 هر بار اجرای حلقه خارجی، یک بار به طور کامل حلقه داخلی اجرا می‌گردد.

SELECT BID

FROM Book b1

WHERE (...)

برای هر سطر از جدول Book شرط تساوی که حاصل یک مقایسه می‌باشد، محاسبه می‌گردد، اگر برابر بود، شرط جلوی where که همان تساوی است، TRUE می‌گردد و سطر مورد نظر از جدول Book انتخاب می‌گردد و این روش برای تک تک سطوح‌های جدول Book، تا به انتهای جدول Book ادامه پیدا می‌کند. به بیان دیگر این پرس و جو شماره کتاب‌هایی را می‌دهد که پرانتز مقابل where برای آنها TRUE است. این پرانتز هنگامی TRUE می‌شود که حاصل مقایسه بیان شده در این پرانتز برابر شود. حاصل این مقایسه در صورتی برابر می‌شود که تعداد نسخه‌های به امانت رفته کتاب مورد بررسی برابر کل نسخه‌های کتاب مورد بررسی باشد. به عبارت دیگر پرس و جوی گزاره b لیست تمام کتاب‌هایی را نمایش می‌هد که تمام نسخه‌های آنها امانت گرفته شده است. که مطابق پرس و جوی مطرح شده در صورت سوال است.

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

Select BID

From BOOK b1

Where (select COUNT (UID)

From BORROWING

Where BORROWING.BID = b1.BID and

BORROWING.EndDate is NULL) = b1.TotalNumber

توسط دستور (UID) count تعداد کتاب‌های به امانت رفته را می‌شمارد و محاسبه می‌کند.

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

Select BID

From BOOK b1

Where (select COUNT (UID)

From BORROWING

Where BORROWING.BID = b1.BID and

BORROWING.EndDate is NULL) = b1.TotalNumber

توسط دستور BORROWING.BID = b1.BID به محیط خارج یعنی جدول Book متصل می‌گردد. حال به ازای حرکت در هر سطر از جدول Book، یک بار به طور کامل سطرهای جدول BORROWING از ابتداء تا انتها با توجه به شرط اتصال بررسی می‌گردد. مطابق شکل زیر:

ابتدا برای سطر اول از جدول Book با توجه به شرط اتصال BORROWING.BID = b1.BID داریم:

B1	... BOOK	B1	... BORROWING	U1	... USER
BID	TOTALNUMBER	UID	BID	... EndDate	UID
B1	1	U1	B1	NULL	U1
B2	2	U2	B2	NULL	U2
B3	3	U3	B2	NULL	U3
		U4	B3	NULL	U4
		U5	B3	NULL	U5

که پرس و جوی زیر برای آن اجرا می‌شود:

Select COUNT (UID)

From BORROWING

Where BORROWING.BID = b1.BID and

BORROWING.EndDate is NULL

توجه: مقدار (UID) count برای سطر اول جدول Book یعنی B1 برابر مقدار 1 است. که رابطه‌ی برابری برای آن برقرار است، به صورت زیر:

count (UID) = b1.TotalNumber

1 1

شرط تساوی جلوی where برقرار است، بنابراین شرط TRUE می‌گردد. بنابراین سطر اول از جدول Book در خروجی نمایش داده می‌شود.

بنابراین کد کتاب **B1** جهت نمایش در خروجی انتخاب می‌شود.

حال برای سطر دوم از جدول Book با توجه به شرط اتصال $\text{BORROWING.BID} = \text{b1.BID}$ داریم:

B2	B2		UID	BID	...	EndDate	UID	...
B1		1	U1	B1		NULL	U1	
B2		2	U2	B2		NULL	U2	
B3		3	U3	B2		NULL	U3	
BOOK			U4	B3		NULL	U4	
			U5	B3		NULL	U5	
			BORROWING			USER		

که پرس و جوی زیر برای آن اجرا می‌شود:

Select COUNT (UID)

From BORROWING

Where BORROWING.BID = b1.BID and

BORROWING.EndDate is NULL

توجه: مقدار count (UID) برای سطر دوم جدول Book یعنی B2 برابر مقدار 2 است. که رابطه‌ی برابری برای آن برقرار است، به صورت زیر:

$$\underline{\text{count (UID)}} = \underline{\text{b1.TotalNumber}}$$

2 2

شرط تساوی جلوی where برقرار است، بنابراین شرط where TRUE می‌گردد. بنابراین سطر دوم از جدول Book در خروجی نمایش داده می‌شود.

بنابراین کد کتاب **B2** جهت نمایش در خروجی انتخاب می‌شود.

حال برای سطر سوم از جدول Book با توجه به شرط اتصال $\text{BORROWING.BID} = \text{b1.BID}$ داریم:

B3	B3		UID	BID	...	EndDate	UID	...
B1		1	U1	B1		NULL	U1	
B2		2	U2	B2		NULL	U2	
B3		3	U3	B2		NULL	U3	
BOOK			U4	B3		NULL	U4	
			U5	B3		NULL	U5	
			BORROWING			USER		

که پرس و جوی زیر برای آن اجرا می‌شود:

Select count (UID)

From BORROWING

Where BORROWING.BID = b1.BID and

BORROWING.EndDate is NULL

توجه: مقدار count (UID) برای سطر سوم جدول Book یعنی B3 برابر مقدار 2 است. که رابطه‌ی برابری برای آن برقرار نیست، به صورت زیر:

$$\underline{\text{count}(UID)} \neq b1.\underline{\text{TotalNumber}}$$

2	3
---	---

شرط تساوی جلوی where برقرار نیست، بنابراین شرط where FALSE می‌گردد. بنابراین سطر سوم از جدول Book در خروجی نمایش داده نمی‌شود.

بنابراین کد کتاب **B3** جهت نمایش در خروجی انتخاب نمی‌شود.
و در نهایت پس از اجرای کامل دستور Select BID خروجی پرس و جو به صورت زیر خواهد بود:

BID
B1
B2

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «لیست تمام کتاب‌هایی را نمایش می‌هد که تمام نسخه‌های آنها امانت گرفته شده است.»
مطابق پرس و جوی مطرح شده در گزاره **c**، داریم:

c) Select BID, TotalNumber

```
From BORROWING,BOOK
Where BOOK.BID = BORROWING.BID and EndDate is NULL
Group by BORROWING.BID
Having COUNT(BID) = TotalNumber
```

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

```
Select BORROWING.BID,TotalNumber
From BORROWING,BOOK
Where BOOK.BID = BORROWING.BID and EndDate is NULL
Group by BORROWING.BID, TotalNumber
Having COUNT(BORROWING.BID) = TotalNumber
```

توجه: در فرم اصلاح شده‌ی گزاره **c**، نام جدول BORROWING پشت ستون BID در دستور select قرار گرفت، که کامپایلر متوجه شود ستون BID جدول BOOK یا BORROWING را انتخاب کند، پرس و جوی گزاره **c** تا به اینجا یک خطای نحوی داشت که اصلاح شد. اما یک خطای نحوی پررنگتری هم دارد، اینکه همواره به غیر از توابع آماری، همه ستون‌های جلوی دستور Select باید زیرمجموعه یا مساوی ستون‌های دستور Group By باشد، بنابراین در فرم اصلاح شده‌ی گزاره **c**، ستون TotalNumber جلوی دستور Group By قرار گرفت. گزاره **c** تا به اینجا دو خطای نحوی داشت که اصلاح شد. همچنین نام جدول BORROWING پشت ستون BID در دستور having قرار گرفت، که کامپایلر متوجه شود ستون BID جدول BOOK یا BORROWING را انتخاب کند، نتیجه اینکه پرس و جوی گزاره **c** سه خطای نحوی داشت که اصلاح شد.

با توجه به جداول فوق، خروجی بخش اول پرس و جوی گزاره‌ی C پس از انجام عملگر ضرب دکارتی به صورت زیر است:

UID	BID	StartDate	EndDate	BID	TITLE	PUBLISHER	TOTALNUMBER
U1	B1	95-01-1	NULL	B1	Tn1	Pub1	1
U2	B2	95-02-1	NULL	B2	Tn2	Pub2	2
U3	B2	95-03-1	NULL	B2	Tn2	Pub2	2
U4	B3	95-04-1	NULL	B3	Tn3	Pub3	3
U5	B3	95-05-1	NULL	B3	Tn3	Pub3	3

همچنین در ادامه، پس از انجام دستور Group By BORROWING.BID، TotalNumber براساس ستون‌های TotalNumber و BORROWING.BID خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:

B1, 1	B2, 2
U1 95-01-1 NULL Tn1 Pub1	U2 95-02-1 NULL Tn2 Pub2
گروه اول	گروه دوم
B3, 3	
U4 95-04-1 NULL Tn3 Pub3	
U5 95-05-1 NULL Tn3 Pub3	
گروه سوم	

و در نهایت دستور having count(BORROWING.BID) = TotalNumber برای هر گروه به طور مستقل اعمال می‌گردد.

توجه: دستور HAVING بر روی گروه‌ها، اعمال می‌گردد.

B1, 1	B2, 2
U1 95-01-1 NULL Tn1 Pub1	U2 95-02-1 NULL Tn2 Pub2
گروه اول	گروه دوم
1 = 1,	2 = 2

B3, 3
U4 95-04-1 NULL Tn3 Pub3
U5 95-05-1 NULL Tn3 Pub3
گروه سوم

$2 \neq 3$

توجه: با توجه به شرط انتخاب گروه توسط دستور having count(BORROWING.BID) = TotalNumber فقط گروه اول و دوم جهت نمایش در خروجی انتخاب می‌شود. و در نهایت دستور select BORROWING.BID, TotalNumber برای هر گروه انتخاب شده

توسط دستور Having به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد، بنابراین خروجی نهایی پرس و جو به صورت زیر است:

BID	TOTALNUMBER
B1	1
B2	2

توجه: دستور GROUP BY، سرگروهها را، راهی خروجی می‌کند.
که مطابق پرس و جوی مطرح شده در صورت سوال نیست. یعنی «لیست تمام کتاب‌هایی را نمایش می‌هد که تمام نسخه‌های آنها امانت گرفته شده است». در واقع پرس و جوی گزاره ۵ به دو دلیل مطابق پرس و جوی مطرح شده در صورت سوال نیست؛ اول اینکه خطای نحوی داشت که جهت اجرا می‌باشد اصلاح می‌شد که ما به فرم اصلاح شده آنرا بررسی کردیم، هرچند که در MySQL این فرم خطای نحوی صادر نمی‌شود. دوم اینکه در صورت سوال صرفاً شماره کتاب‌هایی خواسته شده که همه نسخه‌های آنها به امانت گرفته شده است، در حالیکه در این پرس و جوی علاوه بر شماره کتاب (BID)، تعداد کل نسخه‌های هر کتاب هم در خروجی ظاهر می‌شود که این برخلاف صورت سوال است. بنابراین فقط و فقط گزاره‌های a و b مطابق پرس و جوی صورت سوال است که به تبع گزینه‌های دوم، سوم و چهارم را به طور کامل کنار می‌گذاریم، پس پُر واضح است که گزینه اول پاسخ سوال است.

۶۹- گزینه (۲) صحیح است.

چهار جدول Accident، Car، Person و Owns با مقادیر زیر را در نظر بگیرید:

SSN	name	address	License	year	model

جدول Person

جدول Car

License	accident-date	driver	damage-amount
L1	95-01-1	d1	10
L2	95-02-1	d2	20
L3	95-03-1	d3	30

جدول Accident

SSN	License

جدول Owns

توجه: دقت کنید که مطابق فرض سوال، ترکیب دو ستون License و accident-date هردو باهم به عنوان کلید کاندید جدول Accident مشخص شده است، بنابراین نباید ترکیب این دو ستون باهم سطرهای تکراری داشته باشد، پس یک خودرو (License) در یک تاریخ تصادف (accident-date) نمی‌تواند بیش از

یکبار تکرار شود، یعنی نباید دوبار و در دو سطر و بیشتر تکرار شود.

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

```
(select driver, damage - amount
from Accident) Except (select a.driver, a.damage - amount
from Accident a, Accident b
where a.damage - amount < b.damage - amount and a.driver <> b.driver)
```

با توجه به جداول فوق، خروجی بخش دوم پرس و جوی گزینه اول پس از انجام عملگر ضرب دکارتی به صورت زیر است:

License	accident-date	driver	damage-amount	License	accident-date	driver	damage-amount
L1	95-01-1	d1	10	L1	95-01-1	d1	10
L1	95-01-1	d1	10	L2	95-02-1	d2	20
L1	95-01-1	d1	10	L3	95-03-1	d3	30
L2	95-02-1	d2	20	L1	95-01-1	d1	10
L2	95-02-1	d2	20	L2	95-02-1	d2	20
L2	95-02-1	d2	20	L3	95-03-1	d3	30
L3	95-03-1	d3	30	L1	95-01-1	d1	10
L3	95-03-1	d3	30	L2	95-02-1	d2	20
L3	95-03-1	d3	30	L3	95-03-1	d3	30

خروجی بخش دوم پرس و جوی گزینه اول پس از انجام عملگر ضرب دکارتی و اجرای شرط where a.damage - amount < b.damage - amount and a.driver <> b.driver به صورت زیر است:

License	accident-date	driver	damage-amount	License	accident-date	driver	damage-amount
L1	95-01-1	d1	10	L2	95-02-1	d2	20
L1	95-01-1	d1	10	L3	95-03-1	d3	30
L2	95-02-1	d2	20	L3	95-03-1	d3	30

خروجی بخش دوم پرس و جوی گزینه اول پس از انجام عملگر ضرب دکارتی و اجرای شرط where a.damage - amount < b.damage - amount and a.driver <> b.driver به صورت زیر است:

driver	damage-amount
d1	10
d1	10
d2	20

توجه: شرط where مطرح شده، نام رانندگانی را مشخص می کند که هزینه تصادف آنها از هزینه تصادف حداقل یک راننده دیگر کمتر باشد، به عبارت دیگر شرط where نام رانندگان با هزینه تصادف MAX را حذف می کند. در واقع نام رانندگانی که هزینه تصادف آنها MAX یعنی بیشینه نیست استخراج می شود.

همچنین در ادامه، خروجی بخش اول پرس و جوی گزینه‌ی اول به صورت زیر است:

```
select driver, damage – amount
from Accident
```

driver	damage-amount
d1	10
d2	20
d3	30

توجه: در پرانتز سمت راست عملگر تفاضل (Except) نام رانندگانی مشخص می‌شود که هزینه تصادف آنها از هزینه تصادف حداقل یک راننده دیگر کمتر بوده است. همچنین در پرانتز سمت چپ عملگر تفاضل (Except) نام کلیه رانندگانی که هزینه تصادفی داشته‌اند، مشخص می‌شود. پس از انجام عملگر تفاضل، نام رانندگانی در خروجی پرس و جو استخراج می‌شوند که هزینه تصادفی داشته‌اند و هزینه‌ی تصادف آنها از هزینه تصادف حداقل یک راننده دیگر کمتر نبوده است. به عبارت دیگر در خروجی پرس و جوی گزینه‌ی اول نام رانندگانی استخراج می‌شود که هزینه تصادف آنها MAX و بیشینه بوده است.

در نهایت پس از انجام عملگر Except خروجی پرس و جو به صورت زیر خواهد بود:

driver	damage-amount	Except	driver	damage-amount	=	driver	damage-amount
d1	10	R1 Except R2	d1	10	=	d3	30
d2	20		d1	10			
d3	30		d2	20			

عملگر تفاضل (Except)

این عملگر توسط دستور Except نمایش داده می‌شود. عملگر Except یک عملگر اصلی است. عملگر Except جهت تفاضل سطرهای دو جدول مورد استفاده قرار می‌گیرد. اگر R_1 و R_2 دو رابطه باشند، منظور از $R_1 \text{ Except } R_2$ مجموعه کلیه تاپل‌هایی است که عضو R_1 هستند اما در R_2 حضور ندارند. در جبر رابطه‌ای و SQL تفاضل هر دو رابطه دلخواه امکان‌پذیر نیست. مگر اینکه شروط سازگاری در مورد آنها برقرار باشد. در جبر رابطه‌ای و SQL دو شرط به عنوان شروط سازگاری مطرح است:

شرط اول: تعداد ستون‌های دو جدول یکسان باشد، به عبارت دیگر دو رابطه (جدول) هم درجه باشند.

شرط دوم: نوع یا دامنه ستون‌های متناظر در دو جدول یکسان باشد.

اگر بخواهیم دو شرط فوق را در یک جمله بیان کنیم، اینطور خواهد بود، شروط سازگاری یعنی تیترها در دو رابطه (جدول) یکسان باشد.

فرم کلی عملگر Except به صورت زیر است:

$$R_3 = R_1 \text{ Except } R_2$$

توجه: در پرس و جوی گزینه‌ی اول شروط سازگاری در طرفین عملگر تفاضل (Except) برقرار است.

همچنین جدول Accident با مقادیر زیر را در نظر بگیرید:

License	accident-date	driver	damage-amount
L1	95-01-1	d1	10
L1	95-02-1	d1	20
L1	95-03-1	d1	30

جدول Accident

توجه: در جدول فوق، فقط تصادف‌های یک راننده بررسی شده است.

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

```
(select driver, damage - amount
from Accident) Except (select a.driver, a.damage - amont
from Accident a, Accident b
where a.damage - amount < b.damage - amount and a.driver <> b.driver)
```

با توجه به جداول فوق، خروجی بخش دوم پرس و جوی گزینه‌ی اول پس از انجام عملگر ضرب دکارتی به صورت زیر است:

License	accident-date	driver	damage-amount	License	accident-date	driver	damage-amount
L1	95-01-1	d1	10	L1	95-01-1	d1	10
L1	95-01-1	d1	10	L1	95-02-1	d1	20
L1	95-01-1	d1	10	L1	95-03-1	d1	30
L1	95-02-1	d1	20	L1	95-01-1	d1	10
L1	95-02-1	d1	20	L1	95-02-1	d1	20
L1	95-02-1	d1	20	L1	95-03-1	d1	30
L1	95-03-1	d1	30	L1	95-01-1	d1	10
L1	95-03-1	d1	30	L1	95-02-1	d1	20
L1	95-03-1	d1	30	L1	95-03-1	d1	30

خروجی بخش دوم پرس و جوی گزینه‌ی اول پس از انجام عملگر ضرب دکارتی و اجرای شرط where a.damage - amount < b.damage - amount and a.driver <> b.driver به صورت زیر است:

License	accident-date	driver	damage-amount	License	accident-date	driver	damage-amount

توجه: علت تهی شدن خروجی فوق این است که در شرط where a.driver <> b.driver برقرار باشد، اما a.damage - amount < b.damage - amount برقرار نیست، چون فقط یک راننده وجود دارد که مقدار driver آنها مخالف هم نیستند.

خروجی بخش دوم پرس و جوی گزینه‌ی اول پس از انجام عملگر ضرب دکارتی و اجرای شرط where select a.driver, a.damage - amont به صورت زیر است:

driver	damage-amount

همچنین در ادامه، خروجی بخش اول پرس و جوی گزینه‌ی اول به صورت زیر است:

```
select driver, damage – amount
from Accident
```

driver	damage-amount
d1	10
d1	20
d1	30

در نهایت پس از انجام عملگر Except خروجی پرس و جو به صورت زیر خواهد بود:

driver	damage-amount	Except	driver	damage-amount	=	driver	damage-amount
d1	10				=	d1	10
d1	20				=	d1	20
d1	30				=	d1	30

توجه: نتیجه اینکه گزینه اول در بررسی هزینه‌های تصادف چند راننده مختلف، حالت هزینه بیشینه تصادف را میان رانندگان مختلف استخراج می‌کند، اما در بررسی هزینه‌های تصادف یک راننده، هزینه بیشینه تصادف را مشخص نمی‌کند. بنابراین گزینه اول پاسخ سوال نیست.

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

```
select driver, damage – amount
from Accident
where damage – amount in(select MAX(damage – amount)
from Accident)
```

با توجه به وجود دستور in، زیر پرس و جوی داخلی فرق 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 سطرهای خود، از مقادیر زیر پرس و جوی داخلی استفاده می‌کند. در پرس و جوی فرق به ازای حرکت در هر سطر از جدول Accident مقدار جلوی in بررسی می‌گردد که آیا damage-amount برابر MAX(damage-amount) است یا خیر. اگر برابر بود سطر مورد نظر از جدول Accident در خروجی نمایش داده می‌شود.

توسط دستور where در فرم زیر:

```
SELECT driver , damage-amount
FROM Accident
WHERE damage-amount in (...)
```

برای هر سطر از جدول Accident مقدار جلوی in که حاصل یکتابع آماری به صورت MAX(damage-amount) است، بررسی می‌گردد، اگر damage-amount موجود در هر سطر، برابر MAX(damage-amount) جلوی in بود، آنگاه شرط جلوی where که همان in است، TRUE می‌گردد و سطر مورد نظر از جدول Accident انتخاب می‌گردد و این رویه برای تک تک سطرهای جدول Accident تا به انتهای جدول Accident ادامه پیدا می‌کند. به بیان دیگر این پرس و جو نام رانندگانی از جدول Accident را می‌دهد که در شرط پرانتز مقابل in قرار دارند.

به عبارت دیگر در خروجی پرس و جوی گزینه‌ی دوم نام رانندگانی استخراج می‌شود که هزینه تصادف آنها MAX و بیشینه بوده است. که مطابق پرس و جوی مطرح شده در صورت سوال است.

توجه: همان‌طور که مشاهده می‌کنید، هزینه تصادف راننده d3، از هزینه تصادف تمام رانندگان دیگر بیشتر است.

License	accident-date	driver	damage-amount
L1	95-01-1	d1	10
L2	95-02-1	d2	20
L3	95-03-1	d3	30

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

License	accident-date	driver	damage-amount
L1	95-01-1	d1	10
L2	95-02-1	d2	20
L3	95-03-1	d3	30

Select MAX(damage-amount)
From Accident

که مقدار آن برابر 30 می‌شود، به صورت زیر:

30 > 20 > 10

بنابراین در ادامه پرس و جوی زیر را خواهیم داشت:

SELECT driver , damage-amount
FROM Accident
WHERE damage-amount in (30)

همان‌طور که گفتیم برای هر سطر از جدول Accident مقدار جلوی in که حاصل یک تابع آماری به صورت MAX(damage-amount) است، بررسی می‌گردد، اگر damage-amount موجود در هر سطر، برابر MAX(damage-amount) جلوی in بود، آنگاه شرط جلوی where که همان in است، TRUE می‌گردد و سطر مورد نظر از جدول Accident انتخاب می‌گردد و این رویه برای تک تک سطرهای جدول Accident تا به انتهای جدول Accident ادامه پیدا می‌کند. به بیان دیگر این پرس و جو نام رانندگانی از جدول Accident را می‌دهد که در شرط 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 Accident به صورت زیر است:

driver	damage-amount
d3	30

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «نام رانندگانی استخراج می‌شود که هزینه تصادف آنها MAX و بیشینه بوده است.»

همچنین جدول Accident با مقادیر زیر را در نظر بگیرید:

<u>License</u>	<u>accident-date</u>	driver	damage-amount
L1	95-01-1	d1	10
L1	95-02-1	d1	20
L1	95-03-1	d1	30

جدول Accident

با توجه به جداول فوق، خروجی نهایی پرس و جوی فوق پس از انجام عملگر `in` به ازای هر سطر جدول `Accident` به صورت زیر است:

driver	damage-amount
d3	30

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «نام رانندگانی استخراج می‌شود که هزینه تصادف آنها MAX و بیشینه بوده است.»

توجه: نتیجه اینکه گزینه دوم در بررسی هزینه‌های تصادف چند راننده مختلف، حالت هزینه بیشینه تصادف را میان رانندگان مختلف استخراج می‌کند و همچنین در بررسی هزینه‌های تصادف یک راننده هم، هزینه بیشینه تصادف را مشخص می‌کند. بنابراین گزینه دوم پاسخ سوال است.

توجه: در پرس و جوی گزینه‌ی دوم چون خروجی `select` داخلی فقط یک مقدار و حاصل تابع آماری MAX است، می‌توان از عملگر `=` به جای عملگر `in` نیز استفاده نمود، بنابراین فرم زیر همانند فرم گزینه دوم است.

```
select driver, damage - amount
from Accident
where damage - amount = (select MAX(damage - amount)
                           from Accident)
```

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

```
select driver, damage - amount
from Accident
where damage - amount = MAX(damage - amount)
```

در پرس و جوی گزینه‌ی سوم تابع عددی MAX در محل نادرست مورد استفاده قرار گرفته است، به طور کلی تابع عددی در جلوی Where باید داخل Select قرار بگیرند. بنابراین پرس و جوی گزینه‌ی سوم دارای خطای نحوی است و از سوی کامپایلر اجرا نمی‌گردد. فرم اصلاح شده‌ی گزینه‌ی سوم می‌تواند به فرم گزینه‌ی دوم باشد. بنابراین گزینه سوم پاسخ سوال نیست.

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

- ۷۰- گزینه (۴) صحیح است.

چهار جدول `Person`, `Car`, `Person`, `Car`, `Accident` با مقادیر زیر را در نظر بگیرید:

<u>SSN</u>	<u>name</u>	<u>address</u>	<u>License</u>	<u>year</u>	<u>model</u>
<i>Person</i>			<i>Car</i>		

<u>License</u>	<u>accident-date</u>	<u>driver</u>	<u>damage-amount</u>
L1	95-01-1	d1	10
L1	95-01-2	d1	15
L2	95-02-1	d2	20

جدول Accident

<u>SSN</u>	<u>License</u>

جدول Owns

توجه: دقت کنید که مطابق فرض سوال، ترکیب دو ستون accident-date و License هردو باهم به عنوان کلید کاندید جدول Accident مشخص شده است، بنابراین نباید ترکیب این دو ستون باهم سطرهای تکراری داشته باشد، پس یک خودرو (License) در یک تاریخ تصادف (accident-date) نمی‌تواند بیش از یکبار تکرار شود، یعنی نباید دوبار و در دو سطر و بیشتر تکرار شود.

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

```
select distinct A.License
from Accident A
where A.License in (select B.License
                     from Accident B
                     where A.accident - date <> B.accident - date)
```

با توجه به وجود عملگر in و البته شرط اتصال زیر پرس و جوی داخلی به پرس و جوی خارجی، زیر پرس و جوی داخلی فوق یک Correlated Subquery است، یعنی به ازای حرکت در هریک از سطرهای پرس و جوی خارجی، یک بار به طور کامل از ابتدا تا انتها زیر پرس و جوی داخلی را 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 هر بار اجرای حلقه خارجی، یک بار به طور کامل حلقه داخلی اجرا می‌گردد. توسط دستور where در فرم زیر:

```
SELECT distinct A.License
FROM Accident A
WHERE A.License IN (...)
```

برای هر سطر از جدول Accident شرط جلوی IN که حاصل یک مقایسه می‌باشد، محاسبه می‌گردد، اگر غیرتھی بود، شرط جلوی where که همان IN است، TRUE می‌گردد و سطر مورد نظر از جدول Accident انتخاب می‌گردد و این رویه برای تک تک سطرهای جدول Accident، تا به انتهای جدول Accident ادامه پیدا می‌کند. به بیان دیگر این پرس و جو شماره پلاک ماشین‌هایی را می‌دهد که پرانتر مقابل IN برای آن‌ها غیرتھی است. این پرانتر هنگامی غیرتھی می‌شود که حاصل مقایسه بیان شده در این پرانتر غیرتھی شود. حاصل این مقایسه در صورتی غیرتھی می‌شود که شماره پلاک ماشین مورد بررسی در بیش از یک تصادف، درگیر باشد. به عبارت دیگر پرس و جوی گزینه‌ی اول شماره پلاک ماشین‌هایی را استخراج می‌کند که در بیش از یک تصادف، درگیر هستند. که مطابق پرس و جوی مطرح شده در صورت سوال است.

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

```
select distinct A.License
from Accident A
where A.License in(select B.License
                    from Accident B
                    where A.accident - date <> B.accident - date)
```

توسط دستور $A.accident - date <> B.accident - date$ به محیط خارج یعنی جدول Accident A متصل می‌گردد. حال به ازای حرکت در هر سطر از جدول A، یک بار به طور کامل سطراهای جدول B از ابتدا تا انتها با توجه به شرط اتصال بررسی می‌گردد. مطابق شکل زیر:

A.accident=date = B.accident=date با توجه به شرط اتصال

L1

L1

داریم:

License	Accident.date	...	License	Accident.date	...
L1	95-01-1		L1	95-01-1	
L1	95-01-2		L1	95-01-2	
L2	95-02-1		L2	95-02-1	

Accident A

Accident B

که پرس و جوی زیر برای آن اجرا می‌شود:

```
select distinct A.License
from Accident A
where A.License in(select B.License
                    from Accident B
                    where A.accident - date <> B.accident - date)
```

توجه: خروجی زیر پرس و جوی داخلی در تاریخ تصادفات مختلف به ازای سطر اول جدول Accident A به صورت زیر:

B.License
L1

Accident B

جلوی in برابر غیرتهی است، بنابراین شرط where در پشت in TRUE می‌گردد. بنابراین سطر اول از جدول Accident A در خروجی نمایش داده می‌شود.

بنابراین شماره پلاک ماشین **L1** از آنجاکه در بیش از یک تصادف در تاریخ‌های مختلف، درگیر بوده است، جهت نمایش در خروجی انتخاب می‌شود، به صورت زیر:

B.License
L1

Accident A

حال برای سطر دوم نیز دقیقا همان روال سطر اول تکرار می‌شود.

حال برای سطر سوم از جدول Accident با توجه به شرط اتصال $A.\underline{\text{accident_date}} = B.\underline{\text{accident_date}}$ داریم:

<u>License</u>	<u>Accident.date</u>	...	<u>License</u>	<u>Accident.date</u>	...
L1	95-01-1		L1	95-01-1	
L1	95-01-2		L1	95-01-2	
L2	95-02-1		L2	95-02-1	

*Accident A**Accident B*

که پرس و جوی زیر برای آن اجرا می‌شود:

```
select distinct A.License
from Accident A
where A.License in(select B.License
                    from Accident B
                    where A.accident - date <> B.accident - date)
```

توجه: خروجی زیر پرس و جوی داخلی در تاریخ تصادف‌های مختلف به ازای سطر سوم جدول Accident A به صورت زیر:

B.License

نهی

Accident B

جلوی in برابر تهی است، بنابراین شرط where در پشت in FALSE می‌گردد. بنابراین سطر سوم از جدول Accident A در خروجی نمایش نمی‌شود.

بنابراین شماره پلاک ماشین **L2** از آنجاکه در بیش از یک تصادف در تاریخ‌های مختلف، درگیر نبوده است، جهت نمایش در خروجی انتخاب نمی‌شود، به صورت زیر:

B.License

نهی

Accident A

و در نهایت پس از اجرای کامل دستور Select distinct A.License خروجی پرس و جو به صورت زیر خواهد بود:

A.License

L1

L1

Accident A

که پس از اجرای دستور distinct سطرهای تکراری حذف می‌شود، به صورت زیر:

A.License

L1

Accident A

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «شماره پلاک ماشین‌هایی که در بیش از یک تصادف در تاریخ‌های مختلف درگیر بوده‌اند.»

فرم دوم گزینه اول با استفاده از دستور exists به صورت زیر است:

```
select distinct A.License
from Accident A
where exists(select B.License
            from Accident B
            where A.License=B.License and A.accident - date < B.accident - date)
```

که پس از اجرای دستور distinct سطرهای تکراری حذف می‌شود، به صورت زیر:

A.License
L1
Accident A

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «شماره پلاک ماشین‌هایی که در بیش از یک تصادف در تاریخ‌های مختلف درگیر بوده‌اند.»

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

```
select License
from Accident
group by License
having count(accident - date)>1
```

با توجه به جداول فوق، پس از انجام دستور group by License براساس ستون License خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:

L1 95-01-1 d1 10 95-01-2 d1 15 <hr/> گروه اول	L2 95-02-1 d2 20 <hr/> گروه دوم
---	--

توجه: هر گروه شامل تصادفاتی ثبت شده در تاریخ‌های مختلف برای یک شماره پلاک است.
و در نهایت دستور having count(accident - date)>1 برای هر گروه به طور مستقل اعمال می‌گردد.
توجه: دستور HAVING بر روی گروه‌ها، اعمال می‌گردد.

L1 95-01-1 d1 10 95-01-2 d1 15 <hr/> گروه اول 2>1,	L2 95-02-1 d2 20 <hr/> گروه دوم 1>1
---	---

توجه: با توجه به شرط انتخاب گروه توسط دستور having count(accident - date)>1، فقط گروه اول جهت نمایش در خروجی انتخاب می‌شود.
و در نهایت دستور select داخل دستور Having برای هر گروه انتخاب شده توسط دستور Having

به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد، بنابراین خروجی نهایی پرس و جو به صورت زیر است:

License

L1

توجه: دستور GROUP BY، سرگروهها را، راهی خروجی می‌کند، که به تبع سرگروه، سطر تکراری ندارد.
که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «شماره پلاک ماشین‌هایی که در بیش از یک تصادف در تاریخ‌های مختلف درگیر بوده‌اند».
مطابق پرس و جوی مطرح شده در گزینه‌ی سوم، داریم:

```
select A.License
from Accident A, Accident B
where A.License = B.License and A.accident - date <> B.accident - date
```

با توجه به جداول فوق، خروجی پرس و جوی گزینه‌ی سوم پس از انجام عملکری ضرب دکارتی به صورت زیر است:

License	accident-date	driver	accident-amount	License	accident-date	driver	accident-amount
L1	95-01-1	d1	10	L1	95-01-2	d1	15
L1	95-01-2	d1	15	L1	95-01-1	d1	10

و در نهایت پس از اجرای دستور select A.License خروجی پرس و جو به صورت زیر خواهد بود:

A.License

L1

L1

Accident A

که مطابق پرس و جوی مطرح شده در صورت سوال نیست، زیرا در صورت سوال این فرض قید شده است که پاسخ به زبان SQL باشد و سطرهای تکراری فقط یکبار نشان داده شوند، که در پرس و جوی گزینه‌ی سوم به دلیل عدم استفاده از دستور distinct در خروجی دستور select جهت حذف سطرهای تکراری، فرض مطرح شده در صورت سوال نقض شده است. بنابراین گزینه‌ی سوم پاسخ سوال نیست.

توجه: البته اگر پرس و جوی گزینه‌ی سوم به فرم زیر اصلاح شود آنگاه خروجی گزینه‌ی سوم، همانند گزینه اول و دوم می‌شود، به عبارت دیگر فرم اصلاح شده گزینه‌ی سوم، همان فرم کلاسیک گزینه‌های اول و دوم با استفاده از ضرب دکارتی است، فرم اصلاح شده گزینه‌ی سوم به فرم زیر است:

```
select distinct A.License
from Accident A, Accident B
where A.License = B.License and A.accident - date <> B.accident - date
```

و در نهایت پس از اجرای کامل دستور Select distinct A.License خروجی پرس و جو به صورت زیر خواهد بود:

A.License

L1**L1**

Accident A

که پس از اجرای دستور distinct سطرهای تکراری حذف می‌شود، به صورت زیر:

A.License

L1

Accident A

که مطابق پرس و جوی مطرح شده در صورت سوال است. یعنی «شماره پلاک ماشین‌هایی که در بیش از یک تصادف در تاریخ‌های مختلف در گیر بوده‌اند.»
توجه: سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه چهارم را به عنوان پاسخ اعلام کرده بود.

۷۱ - گزینه (۳) صحیح است.

دو جدول Node و Edge با مقادیر زیر را در نظر بگیرید:

NID	Name	Color	Description	NID1	NID2	EdgeType
N1	NN1	C1	D1	N1	N2	T1
N2	NN2	C2	D2	N1	N3	T2
N3	NN3	C3	D3	N4	N1	T3
N4	NN4	C4	D4	N3	N2	T4
				N4	N3	T5

Node جدول

Edge جدول

توجه: کلید کاندید جدول Node ستون NID است و جدول Edge دارای دو کلید خارجی NID1 و NID2 است که هر دو به کلید کاندید جدول Node ارجاع می‌کنند.

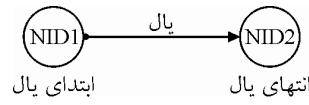
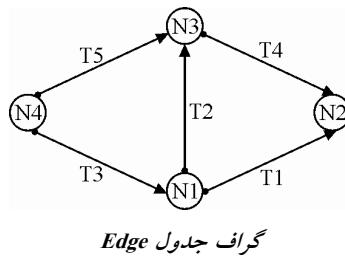
توجه: ستون NID1 در جدول Edge به عنوان کلید خارجی به ستون NID از جدول Node ارجاع می‌کند.

توجه: ستون NID2 در جدول Edge به عنوان کلید خارجی به ستون NID از جدول Node ارجاع می‌کند.

توجه: مقادیر کلید خارجی یعنی ستون‌های NID1 و NID2 از جدول Edge همواره باید زیرمجموعه مقادیر کلید کاندید یعنی ستون NID از جدول Node باشد.

توجه: هر تاپل یا سطر یا رکورد از جدول Edge نشانه یک ارتباط است، پس هر تاپل نشانه یک یال هم هست، که مجموع این یال‌ها یک گراف را ایجاد می‌کند. خطی که از یک گره به گره بعدی رسم می‌شود یک یال است. هر گراف G شامل دو مجموعه V و E است. V مجموعه محدود و غیرتهی از رئوس است و E مجموعه‌ای محدود و احتمالاً تهی از لبه‌ها می‌باشد. (V(G) و E(G) مجموعه رئوس و لبه‌های گراف G را نمایش می‌دهند. برای نمایش یک گراف می‌توانیم بنویسیم $(E, V) = G$. در یک گراف بدون جهت، زوج رئوس، زوج مرتب نیستند. بنابراین زوج‌های (V_0, V_1) و (V_1, V_0) باهم یکسان هستند. گراف حداقل یک راس دارد و نمی‌تواند کاملاً تهی باشد. در یک گراف جهت دار، هر لبه با زوج مرتب $<V_0, V_1>$ نمایش داده می‌شود، که پیکانی از V_0 به V_1 ترسیم می‌شود. بنابراین $<V_0, V_1>$ و $<V_1, V_0>$ دو لبه متفاوت را

نمایش می‌دهند. گراف جهت دار را به صورت $G = \langle V, E \rangle$ نمایش می‌دهیم.
توجه: گراف جهت دار جدول Edge به صورت زیر است.



هر تاپل از جدول Edge نشانه یک یال از گراف جدول Edge است.
توجه: همانطور که واضح است گره‌های موجود در ستون NID1 از جدول Edge یا گراف Edge، یال‌های خروجی دارند.

$NID1 = \{N1, N3, N4\}$

یعنی از گره‌های N1، N3 و N4 یال خارج شده است.
فرم اول پرس و جوی آن به صورت زیر است:

```
select NID
from node
where EXISTS (select *
               from edge
               where node.NID=edge.NID1)
```

فرم دوم پرس و جوی آن به صورت زیر است:

```
select NID
from node
where NID IN (select NID1
               from edge)
```

فرم سوم پرس و جوی آن به صورت زیر است:

```
select distinct NID
from node , edge
where node.NID=edge.NID1
```

توجه: در فرم سوم یا کلاسیک با استفاده از ضرب دکارتی، جهت حذف سطوحی تکراری می‌بایست از دستور distinct استفاده نمود.

خروجی پرس و جوها به صورت زیر است:

NID
N1
N3
N4

توجه: همانطور که واضح است گرههای موجود در ستون NID2 از جدول Edge یا گراف Edge، یالهای ورودی دارند.

$$NID2 = \{N1, N2, N3\}$$

یعنی به گره‌های N1، N2 و N3 یا وارد شده است.
فرم اول پرس و جوی آن به صورت زیر است:

select NID

```
from node  
where EXISTS (select *  
              from edge  
              where node.NID=edge.NID2)
```

فرم دوم پرس و جوی آن به صورت زیر است:

```
select NID  
from node  
where NID IN (select NID2  
               from edge)
```

فرم سوم پرس و جوی آن به صورت زیر است:

```
select distinct NID  
from node , edge  
where node.NID=edge.NID2
```

توجه: در فرم سوم یا کلاسیک با استفاده از ضرب دکارتی، جهت حذف سطرهای تکراری می‌باشد از دستور `distinct` استفاده نمود.

خر و جی پرس و جوها به صورت زیر است:

NID

نتیجه: استخراج گرهایی که هم یالی از آن خارج شده است و هم یالی به آن وارد شده است، به صورت زیر است:

$$\text{NID1} \cap \text{NID2} = \{\text{N1}, \text{N3}, \text{N4}\} \cap \{\text{N1}, \text{N2}, \text{N3}\} = \{\text{N1}, \text{N3}\}$$

فرم اول پرس و جوی آن به صورت زیر است:

(select NID

```
SELECT ALL  
from node  
where EXISTS (select *  
              from edge  
              where nod
```

فرم دوم پرس و جوی آن به صورت زیر است:

```
select NID
from node
where EXISTS (select *
               from edge
               where node.NID=edge.NID1) AND EXISTS (select *
                                                       from edge
                                                       where node.NID=edge.NID2))
```

خروجی پرس و جوها به صورت زیر است:

<u>NID</u>
N1
N3

استخراج گرههایی که يالی از آن خارج شده است ولی يالی به آن وارد نشده است، به صورت زیر است:

$NID1 - NID2 = \{N1, N3, N4\} - \{N1, N2, N3\} = \{N4\}$

فرم اول پرس و جوی آن به صورت زیر است:

```
(select NID
from node
where EXISTS (select *
               from edge
               where node.NID=edge.NID1)) EXCEPT (select NID
                                                 from node
                                                 where EXISTS (select *
                                                               from edge
                                                               where node.NID=edge.NID2))
```

فرم دوم پرس و جوی آن به صورت زیر است:

```
select NID
from node
where EXISTS (select *
               from edge
               where node.NID=edge.NID1) AND NOT EXISTS (select *
                                                       from edge
                                                       where node.NID=edge.NID2)
```

خروجی پرس و جوها به صورت زیر است:

<u>NID</u>
N1
N3

استخراج گرههایی که يالی به آن وارد شده است ولی يالی از آن خارج نشده است، به صورت زیر است:

$NID2 - NID1 = \{N1, N2, N3\} - \{N1, N3, N4\} = \{N2\}$

فرم اول پرس و جوی آن به صورت زیر است:

```
(select NID
from node
where EXISTS (select *
from edge
where node.NID=edge.NID2)) EXCEPT (select NID
from node
where EXISTS (select *
from edge
where node.NID=edge.NID1))
```

فرم دوم پرس و جوی آن به صورت زیر است:

```
select NID
from node
where EXISTS (select *
from edge
where node.NID=edge.NID2) AND NOT EXISTS (select *
from edge
where node.NID=edge.NID1)
```

فرم سوم پرس و جوی آن به صورت زیر است:

```
select distinct NID
from node , edge
where node.NID=edge.NID2 AND NOT EXISTS (select *
from edge
where node.NID = edge.NID1)
```

توجه: می‌توان دستور Exists اول از فرم دوم را به صورت ضرب دکارتی و کلاسیک نیز نوشت.

توجه: که دقیقاً همان فرم پرس و جوی اول مطرح شده در صورت سوال است.

خروجی پرس و جویها به صورت زیر است:

$$\frac{\text{NID}}{\text{N2}}$$

مطابق پرس و جوی اول مطرح شده در صورت سوال، داریم:

```
select distinct NID
from node , edge
where node.NID=edge.NID2 AND NOT EXISTS (select *
from edge
where node.NID = edge.NID1)
```

توجه: از آنجاییکه ستون‌های NID و 1 در دو جدول مختلف تکراری نیستند، بنابراین الزامی بر وجود نام جداول پشت این ستون‌ها وجود ندارد.

با توجه به وجود عملگر not exists، زیر پرس و جوی داخلی فوق یک Correlated Subquery است، یعنی با ازای حرکت در هریک از سطرهای پرس و جوی خارجی، یک بار به طور کامل از ابتدا تا انتهای 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 distinct NID
FROM node , edge
WHERE node.NID=edge.NID2 AND NOT EXISTS (...)
```

برای هر سطر از جدول حاصل از ضرب دکارتی $node \times edge$ به طور همزمان شرط $node.NID = edge.NID2$ (شرط اول) و شرط جلوی NOT EXISTS (شرط دوم) بررسی می‌شود. اگر به طور همزمان شرط $node.NID = edge.NID2$ و همچنین شرط جلوی NOT EXISTS برقرار بود، آنگاه سطر مورد نظر از جدول حاصل از ضرب دکارتی $node \times edge$ انتخاب می‌گردد و این رویه برای تک تک سطرهای جدول حاصل از ضرب دکارتی $node \times edge$ ، تا به انتهای جدول حاصل از ضرب دکارتی $node \times edge$ ادامه پیدا می‌کند. وقت کنید که برای هر سطر از جدول حاصل از ضرب دکارتی $node \times edge$ شرط جلوی NOT EXISTS که حاصل یک مقایسه می‌باشد، محسوبه می‌گردد، اگر تهی بود، شرط جلوی NOT EXISTS (شرط دوم) که همان TRUE است، NOT EXISTS می‌گردد.

به بیان دیگر این پرس و جو شماره گرهایی را می‌دهد که NID آن، عضو ستون NID2 یعنی مجموعه گرهای دارای یال وارد شونده باشد و همچنین به طور همزمان پرانتز مقابل NOT EXISTS برای آنها تهی باشد. این پرانتز هنگامی تهی می‌شود که حاصل مقایسه بیان شده در این پرانتز تهی شود. حاصل این مقایسه در صورتی تهی می‌شود که شماره گره مورد بررسی، NID آن، عضو ستون NID1 یعنی مجموعه گرهای دارای یال خارج شونده نباشد. به عبارت دیگر پرس و جوی اول مطرح شده در صورت سوال شماره گرهایی را استخراج می‌کند که یال ورودی دارند اما یال خروجی ندارند. یعنی استخراج گرهایی که یالی به آن وارد شده است ولی یالی از آن خارج نشده است، که مطابق پرس و جوی اول مطرح شده در صورت سوال است که در هیچکدام از گزینه‌های صورت سوال، این حالت پرس و جو وجود ندارد.

به بیان دیگر عبارت موجود در جلوی دستور not exists که به صورت زیر است:

```
select distinct NID
from node , edge
where node.NID=edge.NID2 AND NOT EXISTS (select *
                                              from edge
                                              where node.NID = edge.NID1)
```

توسط دستور $node.NID = edge.NID1$ به محیط خارج یعنی جدول حاصل از ضرب دکارتی $node \times edge$ متصل می‌گردد. حال به ازای حرکت در هر سطر از جدول حاصل از ضرب دکارتی $node \times edge$ ، یک بار به طور کامل سطرهای جدول edge از ابتدا تا انتها با توجه به شرط اتصال بررسی می‌گردد. مطابق شکل زیر:

ابتدا برای سطر اول از جدول حاصل از ضرب دکارتی $node \times edge$ البته با اعمال شرط

$node.NID = edge.NID1 \wedge edge.NID = edge.NID2$ با توجه به شرط اتصال داریم:

$N2 \times$				$Edge$			
NID	$...$	$NID1$	$NID2$	$EdgeType$	$NID1$	$NID2$	$EdgeType$
N2		N1	N2	T1	N1	N2	T1
N3		N1	N3	T2	N1	N3	T2
N1		N4	N1	T3	N4	N1	T3
N3		N4	N3	T5	N3	N2	T4
N2		N3	N2	T4	N4	N3	T5

Node , Edge

Edge

که پرس و جوی زیر برای آن اجرا می‌شود:

```

select distinct NID
from node , edge
where node.NID=edge.NID2 AND NOT EXISTS (select *
                                              from edge
                                              where node.NID = edge.NID1)

```

توجه: خروجی زیر پرس و جوی داخلی در کترل عضویت شماره گرده مورد نظر یعنی N2 از NID در ستون NID1 یعنی مجموعه گرههای دارای یال خارج شونده. به ازای سطر اول جدول حاصل از ضرب دکارتی node , edge با اعمال شرط node.NID=edge.NID2 به صورت زیر است:

NID1	NID2	EdgeType
Edge		
تهی		

جلوی not exists برابر تهی است، بنابراین شرط where not exists در (شرط دوم) ، TRUE می‌گردد. شرط در where node.NID=edge.NID2 (شرط اول) هم که از قبل برقرار و TRUE بود. که نتیجه کلی شرط where به دلیل وجود عملگر AND می‌شود. بنابراین سطر اول از جدول حاصل از ضرب دکارتی node , edge با اعمال شرط node.NID=edge.NID2 با توجه به شرط اتصال در خروجی نمایش داده می‌شود. بنابراین گره شماره N2 از آنجاکه که یال ورودی دارد اما یال خروجی ندارد. یعنی یالی به آن وارد شده است ولی یالی از آن خارج نشده است، جهت نمایش در خروجی انتخاب می‌شود، به صورت زیر:

NID
N2

حال برای سطر دوم از جدول حاصل از ضرب دکارتی node , edge با اعمال شرط

node.NID=edge.NID داریم:

N2 N2

NID	...	NID1	NID2	EdgeType	NID1	NID2	EdgeType
N2		N1	N2	T1	N1	N2	T1
N3		N1	N3	T2	N1	N3	T2
N1		N4	N1	T3	N4	N1	T3
N3		N4	N3	T5	N3	N2	T4
N2		N3	N2	T4	N4	N3	T5

Node , Edge

Edge

که پرس و جوی زیر برای آن اجرا می‌شود:

```

select distinct NID
from node , edge
where node.NID=edge.NID2 AND NOT EXISTS (select *
                                              from edge
                                              where node.NID = edge.NID1)

```

توجه: خروجی زیر پرس و جوی داخلی در کترل عضویت شماره گرده مورد نظر یعنی N3 از NID در ستون NID1 یعنی مجموعه گرههای دارای یال خارج شونده. به ازای سطر دوم جدول حاصل از ضرب

دکارتی node , edge البته با اعمال شرط node.NID=edge.NID2 به صورت زیر است:

NID1	NID2	EdgeType
N3	N2	T4
<i>Edge</i>		

جلوی not exists برابر غیرتهی است، بنابراین شرط where node.NID=edge.NID2 (شرط دوم) ، not exists در شرط where node.NID=edge.NID2 (شرط اول) هم که از قبل برقرار و TRUE بود. که نتیجه کلی شرط where به دلیل وجود عملگر AND می‌شود FALSE شرط دوم از جدول حاصل از ضرب دکارتی node.NID=edge.NID2 با اعمال شرط node.NID=edge.NID2 با بنابراین سطر دوم از جدول خروجی نمایش داده نمی‌شود. بنابراین گره شماره N3 از آنجاکه که یال ورودی دارد و یال خروجی هم دارد. یعنی یالی به آن وارد شده است و یالی هم از آن خارج شده است، جهت نمایش در خروجی انتخاب نمی‌شود، به صورت زیر:

NID
N2

حال برای سطر سوم (N1)، چهارم (N3) و پنجم (N2) از جدول حاصل از ضرب دکارتی node , edge با اعمال شرط node.NID=edge.NID2 نیز همین روال تکرار می‌شود، همچنین به واسطه وجود دستور select distinct در دستور سطرهای تکراری نیز در خروجی پرس و جو حذف می‌شود. با توجه به جداول فوق، خروجی نهایی پرس و جوی فوق به صورت زیر است:

NID
N2

که مطابق پرس و جوی اول مطرح شده در صورت سوال است. یعنی «شماره گرههایی را استخراج می‌کند که یال ورودی دارند اما یال خروجی ندارند. یعنی استخراج گرههایی که یالی به آن وارد شده است ولی یالی از آن خارج نشده است، که در هیچکدام از گزینه‌های صورت سوال، این حالت پرس و جو وجود ندارد.

مطابق پرس و جوی دوم مطرح شده در صورت سوال، داریم:

```
select T1.NID
from (select count (NID1) as cnt , NID
      from node left outer join edge on edge.NID1=NID
      group by NID) T1 ,
      (select count (NID2) as cnt , NID
      from node left outer join edge on edge.NID2=NID
      group by NID) T2
where T1.NID=T2.NID and T1.cnt < T2.cnt
```

عملگر الحق خارجی چپ در جبر رابطه‌ای

این عملگر، مانند الحق 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 الحق خارجی چپ در SQL (Left Outer Join)

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

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

توجه: برای شمارش درجه خروجی گره‌ها یعنی شمارش یال‌های خارج شونده از هر گره باید جدول Edge در Node در الحق خارجی چپ شود که نتیجه آن T1 نام‌گذاری شده است، تا تعداد تکرار یال‌های خارج شونده از هر گره شمارش شود، به صورت زیر:

```
(select count (NID1) as cnt , NID
from node left outer join edge on edge.NID1=NID
group by NID) T1
```

NID	name	color	Description	NID1	NID2	EdgeType
N1	NN1	C1	D1	N1	N2	T1
N1	NN1	C1	D1	N1	N3	T2
N2	NN2	C2	D2	NULL	NUUL	NULL
N3	NN3	C3	D3	N3	N2	T4
N4	NN4	C4	D4	N4	N1	T3
N4	NN4	C4	D4	N4	N3	T5

با توجه به جدول فوق، پس از انجام دستور group by NID براساس ستون NID خروجی پرس و جو به صورت زیر گروهبندی می‌شود:

N1	N2
NN1 C1 D1 N1 N2 T1	NN2 C2 D2 NULL NULL NULL
<u>NN1 C1 D1 N1 N3 T2</u>	
گروه اول	گروه دوم
N3	N4
NN3 C3 D3 N3 N2 T4	NN4 C4 D4 N4 N1 T3
<u>NN3 C3 D3 N3 N2 T4</u>	<u>NN4 C4 D4 N4 N3 T5</u>
گروه سوم	گروه چهارم

توجه: هر گروه شامل تعداد یال‌های خارج شونده از هر گره است، چون شرط اتصال روی و در نهایت دستور select count (NID1) as cnt , NID در node left outer join edge edge.NID1=NID select دستور داخل select count (NID1) as cnt , NID برای هر گروه به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد، بنابراین خروجی نهایی پرس و جو به صورت زیر است:

T1.cnt	T1.NID
2	N1
0	N2
1	N3
2	N4

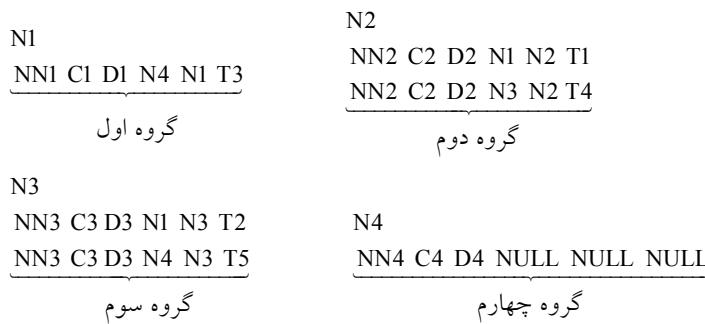
توجه: دستور GROUP BY، سرگروهها را، راهی خروجی می‌کند، که به تبع سرگروه، سطر تکراری ندارد.
توجه: تابع آماری count (NID1) برای هر گروه به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد.

توجه: تابع آماری count (NID1)، ستون NULL را نادیده می‌گیرد و نمی‌شمارد.
توجه: برای شمارش درجه ورودی گره‌ها یعنی شمارش یال‌های وارد شونده به هر گره باید جدول Node در Edge الحاق خارجی چپ شود که نتیجه آن T2 نام‌گذاری شده است، تا تعداد تکرار یال‌های وارد شونده به هر گره شمارش شود، به صورت زیر:

```
(select count (NID2) as cnt , NID
from node left outer join edge on edge.NID2=NID
group by NID) T2
```

NID	name	color	Description	NID1	NID2	EdgeType
N1	NN1	C1	D1	N4	N1	T3
N2	NN2	C2	D2	N1	N2	T1
N2	NN2	C2	D2	N3	N2	T4
N3	NN3	C3	D3	N1	N3	T2
N3	NN3	C3	D3	N4	N3	T5
N4	NN4	C4	D4	NULL	NULL	NULL

با توجه به جدول فوق، پس از انجام دستور group by NID براساس ستون NID خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:



توجه: هر گروه شامل تعداد یال‌های وارد شونده به هر گره است، چون شرط اتصال روی node left outer join edge در Edge.NID2=NID است.
و در نهایت دستور select count (NID2) as cnt , NID برای هر گروه به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد، بنابراین خروجی نهایی پرس و جو به صورت زیر است:

T2.cnt	T2.NID
1	N1
2	N2
2	N3
0	N4

توجه: دستور GROUP BY سرگروه‌ها را، راهی خروجی می‌کند، که به تبع سرگروه، سطر تکراری ندارد.
توجه: تابع آماری count برای هر گروه به طور مستقل اعمال می‌گردد و در خروجی پرس و جو قرار می‌گیرد.

توجه: تابع آماری count را نادیده می‌گیرد و نمی‌شمارد.
توجه: در ادامه T1 و T2 در هم ضرب دکارتی می‌شوند که می‌شود 16 سطر که با اعمال شرط T1.NID=T2.NID در دستور where خروجی زیر را خواهیم داشت:

T1.cnt	T1.NID	T2.cnt	T2.NID
2	N1	1	N1
0	N2	2	N2
1	N3	2	N3
2	N4	0	N4

توجه: در ادامه با اعمال شرط T1.cnt < T2.cnt در دستور where خروجی زیر را خواهیم داشت:

T1.cnt	T1.NID	T2.cnt	T2.NID
0	N2	2	N2
1	N3	2	N3

توجه: در نهایت با اعمال دستور select T1.NID در دستور select خروجی زیر را خواهیم داشت:

$$\begin{array}{c} \text{T1.NID} \\ \hline \text{N2} \\ \hline \text{N3} \end{array}$$

که مطابق پرس و جوی دوم مطرح شده در صورت سوال است. یعنی «شماره گره‌هایی را می‌دهد که درجه خروجی آنها کمتر از درجه ورودی آنها است.» یعنی استخراج گره‌هایی که یال‌های خارج شونده از آن کمتر از یال‌های وارد شونده به آن است، که در گزینه سوم این حالت پرس و جو وجود دارد. بنابراین پُر واضح است که گزینه سوم پاسخ سوال است.

توجه: اگر در شرط where به جای T1.cnt < T2.cnt از T1.cnt = T2.cnt استفاده می‌شد، آنگاه شماره گره‌هایی را می‌داد که درجه خروجی آنها بیشتر از درجه ورودی آنها است. یعنی استخراج گره‌هایی که یال‌های خارج شونده از آن بیشتر از یال‌های وارد شونده به آن است.

توجه: اگر در شرط where به جای T1.cnt < T2.cnt از T1.cnt = T2.cnt استفاده می‌شد، آنگاه شماره گره‌هایی را می‌داد که درجه خروجی آنها برابر درجه ورودی آنها است. یعنی استخراج گره‌هایی که یال‌های خارج شونده از آن برابر یال‌های وارد شونده به آن است.

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

۷۲- گزینه (۲) صحیح است.

اگر فرض کنیم منظور طراح محترم از دو دامنه (Domain) متفاوت، ستون‌های مختلف دو جدول مختلف باشد، آنگاه از آنجا که جداول مربوط به ستون #P یعنی جدول Supply و Part در پرس و 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 می‌شد، 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 معنی Supply آمده است و خطای فقدان و نامعتبر بودن نام ستون (Invalid column name) از سوی کامپایلر صادر نمی‌شود، اما از آنجا که نوع ستون S# از نوع char(5) و نوع ستون QTY از نوع numeric(10) است خطای قیاس دو دامنه (Domain) متفاوت (Error converting data type) یعنی خطای عدم توانایی در تبدیل numeric(10) و char(5) به هم از سوی کامپایلر صادر می‌شود و پرس و جو اجرا نمی‌شود.

مثال: پرس و جوی زیر را در نظر بگیرید:

```
SELECT S#
FROM Supply
WHERE S# = P#;
```

توجه: در پرس و جوی فوق هر دو ستون S# و P# در جدول تعریف شده در پرس و جوی معنی Supply آمده است و خطای فقدان و نامعتبر بودن نام ستون (Invalid column name) از سوی کامپایلر صادر نمی‌شود، همچنین از آنجا که نوع ستون S# از نوع char(5) و نوع ستون P# از نوع char(5) است خطای قیاس دو دامنه (Domain) متفاوت (Error converting data type) یعنی خطای عدم توانایی در تبدیل char(5) و char(5) به هم نیز از سوی کامپایلر صادر نمی‌شود و پرس و جو به درستی اجرا می‌شود. و شماره تولیدکنندگان و قطعات را که باهم یکسان هستند در خروجی ظاهر می‌کند که در این حالت گرینه 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 معنی Supplier آمده است و خطای فقدان و نامعتبر بودن نام ستون (Invalid column name) از سوی کامپایلر صادر نمی‌شود، همچنین از آنجا که نوع ستون S# از نوع char(5) و نوع ستون Sname از نوع char(20) است خطای قیاس دو دامنه (Domain) متفاوت (Error converting data type) یعنی خطای عدم توانایی در تبدیل char(20) و char(5) به هم نیز از سوی کامپایلر صادر نمی‌شود و پرس و جو به درستی اجرا می‌شود. توجه: جهت حفظ حقوق دانشجویانی که یکسال تلاش کرده‌اند، این سوال بهتر بود حذف می‌شد و یا همان گرینه 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 باز هم گزینه دوم را به عنوان پاسخ نهایی اعلام نمود، بدون تاثیر مثبت، که کار خیلی درستی نبوده است.

۷۳- گزینه (۲) صحیح است.

این سوال تکراری است و پاسخ آن دقیقاً همان پاسخ سوال ۷۱ فصل ششم است.

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

۷۴- گزینه (۳) صحیح است.

جداول زیر را در نظر بگیرید:

<u>MID</u>	Name	MDate	<u>MID</u>	ISBN	BDate	Duration	<u>ISBN</u>	...	Publisher
m1	mn1	21	m1	is1	990101	11	is1		Springer
m2	mn2	22	m1	is2	990202	12	is2		Springer
m3	mn3	23	m2	is1	990303	13	is3		Penguin
Member			m2	is2	990404	14	is4		Penguin
			m2	is3	990505	15	Book		
			m3	is1	990606	16			

Borrow

مطابق پرس و جوی مطرح شده در گزاره الف داریم:

```
SELECT Name
FROM (SELECT Member.MID , Member.Name , count (Book.ISBN) as BorrowCount
      FROM Member , Borrow , Book
      WHERE Member.MID = Borrow.MID and Book.ISBN = Borrow.ISBN
            and Publisher = 'Springer'
      GROUP BY Member.MID , Member.Name) as M
WHERE BorrowCount - (SELECT count(*)
                      FROM Book
                      WHERE Publisher = 'Springer')
```

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

```
SELECT Name
FROM (SELECT Member.MID , Member.Name , count (Book.ISBN) as BorrowCount
      FROM Member , Borrow , Book
      WHERE Member.MID = Borrow.MID and Book.ISBN = Borrow.ISBN
            and Publisher = 'Springer'
      GROUP BY Member.MID , Member.Name) as M
WHERE BorrowCount = (SELECT count(*)
                      FROM Book
                      WHERE Publisher = 'Springer')
```

توجه: در فرم اصلاح شده‌ی گزاره الف، عملگر = جایگزین عملگر - شده است.

توجه: گزاره الف به فرم اصلاح شده نشده، خطای نحوی دارد و از سوی کامپایلر اجرا نمی‌شود، دقت کنید که جلوی دستور WHERE باید عملگر منطقی قرار گیرد که حاصل آن هم منطقی باشد. که این موضوع نقض شده است. بنابراین گزاره الف مطابق پرس و جوی مطرح شده در صورت سوال نیست. با توجه به جداول فوق، خروجی پرس و جوی داخلی گزاره‌ی الف پس از انجام عملگر ضرب دکارتی و اعمال شرط اتصال به صورت زیر است:

MID	Name	MDate	MID	ISBN	BDate	Duration	ISBN	...	Publisher
m1	mn1	21	m1	is1	990101	11	is1	...	Springer
m1	mn1	21	m1	is2	990202	12	is2	...	Springer
m2	mn2	22	m2	is1	990303	13	is1	...	Springer
m2	mn2	22	m2	is2	990404	14	is2	...	Springer
m3	mn3	23	m3	is1	990606	16	is1	...	Springer

توجه: شرط اتصال زیر:

WHERE Member.MID = Borrow.MID and Book.ISBN = Borrow.ISBN

and Publisher = 'Springer'

شماره کتاب‌های انتشارات Springer را جلوی امانت گیرنده آن قرار می‌دهد.

همچنین در ادامه، پس از انجام دستور GROUP BY Member.MID , Member.Name براساس ستون‌های Member.MID , Member.Name , count (Book.ISBN) as BorrowCount خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:

m1, mn1	m2, mn2
21 m1 is1 ... 11 is1 ... Springer	22 m2 is1 ... 13 is1 ... Springer
21 m1 is2 ... 12 is2 ... Springer	22 m2 is2 ... 14 is2 ... Springer

گروه اول

گروه دوم

m3, mn3
23 m3 is1 ... 16 is1 ... Springer

گروه سوم

توجه: دستور GROUP BY، سرگروه‌ها را، راهی خروجی می‌کند. و در نهایت دستور SELECT برای هر گروه به طور مستقل اعمال می‌گردد.

MID	Name	BorrowCount
m1	mn1	2
m2	mn2	2
m3	mn3	1

توجه: دستور count (Book.ISBN) as BorrowCount بر روی گروه‌ها، اعمال می‌گردد.

توجه: خروجی قطعه پرس و جوی مطرح شده در دستور WHERE خارجی به صورت زیر است:

```
2 = (SELECT count(*)
      FROM Book
      WHERE Publisher = 'Springer')
```

توجه: با توجه به شرط انتخاب سطر توسط دستور (SELECT count(*)
WHERE BorrowCount = (SELECT count(*))
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 اعضايی 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 اعضايی 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'
```

«شماره همه 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.

name
mn1
mn2

راه حل دوم:

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

«نام اعضايی 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 پخواهد همه حالت‌های یک اتفاق را بررسی کند. مانند نام اعضايی که همه کتاب‌های منتشر شده توسط Springer را به امانت برده‌اند. که به امانت بردن کتاب‌ها توسط اعضاء، 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 اعضايی که، وجود نداشته باشد (**NOT EXISTS**) کتابی از انتشارات Springer که، به امانت برده نشده باشد (**EXCEPT**). یعنی نام اعضايی که همه کتاب‌های انتشارات Springer را امانت گرفته‌اند.»

بنابراین پرس و جوی فوق، نام اعضايی 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 اعضايی که همه کتاب‌های انتشارات Springer را امانت گرفته‌اند». مطابق پرس و جوی مطرح شده در گزاره ج داريم:

```
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 بگيريد:

«نام اعضايی که همه کتاب‌های انتشارات Springer را امانت گرفته‌اند.»

توجه: عملگر تقسيم در جبرابطه‌اي به فرم زير در SQL قابل پياده‌سازی است.

توجه: اين پرس و جو به فرم زير توسط الگوي اول (همه) در SQL قابل پياده‌سازی است.

الگوي 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 زير بخوانيد:

«نام اعضايی که، وجود نداشته باشد (**NOT EXISTS**) کتابی از انتشارات Springer که، به امانت برده نشده باشد (**NOT EXISTS**). یعنی نام اعضايی که همه کتاب‌های انتشارات Springer را امانت گرفته‌اند.»

در پرس و جوی فوق به ازاي حرکت در هر سطر از جدول Member، يك بار به طور كامل از ابتدا تا انتهای جدول Book بررسی می‌گردد، تا مشخص گردد هیچ سطري در جدول Book از انتشارات Springer وجود ندارد که شرط اتصال Borrow.MID = Member.MID AND Borrow.ISBN = Book.ISBN موجود در select داخلي را برقرار نکرده باشد، اگر وجود نداشته باشد، سطر مورد نظر از جدول Member در خروجي

نمایش داده می‌شود.

توجه: بعضی آدم‌ها در دنیا هستن که کاری نیست (NOT EXISTS) که، انجام نداده باشن (NOT EXISTS)، یعنی هر کاری بگی انجام دادن. قایقی نیست که سوار نشده باشن، تفریحی نیست که انجام نداده باشن و ... این افراد به عرض زندگی بیشتر توجه داشتن تا طول زندگی!

توجه: شرط لازم برای تبعیت از الگوی اول وجود **NOT EXISTS** اول و **NOT EXISTS** دوم است و شرط کافی برای تبعیت از الگوی اول وجود شروط اتصال مناسب و سازگار است.

بنابراین پرس و جوی فوق، نام اعضايی 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 اعضايی که همه کتاب‌های انتشارات Springer را امانت گرفته‌اند».

بنابراین فقط و فقط گزاره‌های ب و ج مطابق پرس و جوی صورت سوال است، پس پُرواضح 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.

تست‌های فصل هفتم: SQL دستورات DDL و DCL

۱- روش مناسب برای تعریف محدودیت‌های جامعیت عمومی در SQL چیست؟
(مهندسی IT - دولتی ۸۵)
Triggered Procedures (۴) Create Assertion (۳) Create Domain (۲) Create Table (۱)

۲- اگر روی رابطه (A,B,C,D) R دید (View) زیر تعریف شود، کدام یک از گزاره‌ها درست است؟
(مهندسی کامپیوتر - دولتی ۷۶)

Create View V (A ,B ,C)
As select A, B, sum (D)
From R
Group By A, B

- (۱) از این دید می‌توان عملیات ذخیره‌سازی را انجام داد.
- (۲) از این دید نمی‌توان عملیات ذخیره‌سازی را انجام داد.
- (۳) از این دید می‌توان عملیات ذخیره‌سازی انجام داد ولی عوارض نامطلوب دارد.
- (۴) از این دید می‌توان عملیات ذخیره‌سازی انجام داد ولی استقلال داده‌ای تضعیف می‌شود.

۳- اگر رابطه ساعت (Watches) به طریق زیر تعریف شده باشد:

CREATE TABLE Watches (name Char (30), Manf char (30) References Manfs (Manf))

برای آنکه دستور DML زیر را اجرا کنیم کدام یک از دسترسی‌های ذکر شده لازم است:

INSERT INTO Watches VALUES ('Tissot', 'swatch')

SELECT ON Watches (۲) SELECT ON Manfs (۱)
REFERENCES ON Manfs (۴) UPDATE ON Watches (۳)

۴- دو رابطه (A,B,C,D) و (B,C,D) در بانک اطلاعاتی رابطه‌ای مفروض است و دیدگاه (View) به نام 'RR' را به شکل زیر تعریف می‌کنیم:

Create View RR' As
(Select A , B from R)
Union
(Select A , B from R')

- کدام گزینه با تعریف فوق صادق است؟
- RR'(C',C) (۴) RR'(A,B) (۳) RR'(A,B,A,B) (۲) RR'(C',D',C,D) (۱)

۵- با توجه به جدول employee، دستور زیر چه کاری انجام می‌دهد؟
(مهندسی کامپیوتر - دولتی ۸۳)

employee (empno, empname, dept, salary)، نام کارمند: empno، شماره کارمند: salary

Grant select, update (empno, empname) on employee to Roya

- (۱) این دستور مشکل ساختاری دارد.
- (۲) اجازه به روز در آوردن و نیز انتخاب ستون‌های جدول employee را به Roya می‌دهد.
- (۳) اجازه به روز در آوردن و نیز انتخاب ستون‌های جدول را از Roya پس می‌گیرد.
- (۴) اجازه به روز در آوردن empno و empname و نیز انتخاب ستون‌های جدول employee را به Roya می‌دهد.

۶- در بانک اطلاعاتی تولید کنندگان و قطعات، دستور SQL زیر باعث.... (مهندسی IT - دولتی ۸۶)

```
CREATE VIEW LONDONSUP
As SELECT *
FROM S
WHERE CITY='LONDON'
```

- (۱) افزایش سرعت پاسخگویی به برخی از پرس‌وجوها می‌شود.
- (۲) خطای زمان اجرا می‌شود.
- (۳) خطای نحوی (Syntax) می‌شود.
- (۴) سهولت بیان برخی از پرس‌وجوها می‌شود.

۷- کدام دستور با معنای داده شده، در SQL وجود ندارد؟ (مهندسی IT - دولتی ۸۶)

- | | | |
|----------------------------------|--------------------------------|----------------------------------|
| (۱) Delete برای تغییر view | (۲) Update برای این بردن جداول | (۳) Revoke برای لغو اجازه دسترسی |
| (۴) Update برای تغییر جداول پایه | | |

۸- دستور زیر چه کاری انجام می‌دهد؟ (مهندسی کامپیوتر-آزاد ۸۷)

```
Revoke update on stud
From Amiri
```

- (۱) اجازه انجام update روی stud را به کاربری به نام Amiri می‌دهد.
- (۲) فایل سابقه (log file)، دستورات انجام شده روی stud از طرف کاربری به نام Amiri را به روز می‌کند.
- (۳) باعث می‌شود سابقه دستورات انجام شده از طرف کاربری به نام Amiri روی stud از نو ساخته شود.
- (۴) اجازه انجام update روی stud را از کاربری به نام Amiri بازپس می‌گیرد.

۹- دستور زیر چه کاری انجام می‌دهد؟ (مهندسی IT - آزاد ۸۷)

```
Grant update on stud
To Ahmadi
```

- (۱) اجازه انجام update روی stud را به کاربری به نام Ahmadi می‌دهد.
- (۲) اجازه انجام update روی stud را از کاربری به نام Ahmadi بازپس می‌گیرد.
- (۳) فایل شاخص را روی stud برای کاربری به نام Ahmadi به روز می‌کند.
- (۴) باعث می‌شود فایل سابقه (log file) برای دستورات انجام شده از طرف کاربری به نام Ahmadi به روز شود.

۱۰- کدام یک از عبارات زیر صحیح است؟ (مهندسی IT - آزاد ۸۷)

- (۱) جدول پایه (Base Table) جدولی است که توسط کاربران ایجاد و حذف می‌شود.
 - (۲) جدول میانی (Intermediate Table)، جدولی است که توسط سیستم ایجاد، مدیریت و حذف می‌شود و کاربر دخالتی در آن ندارد.
 - (۳) دید (View)، جدولی است که وجود خارجی ندارد ولی می‌توان روی آن پرس‌وجو انجام داد.
- | | | | |
|-------|-------|-------|-------|
| ۱ (۴) | ۳ (۳) | ۲ (۲) | ۱ (۱) |
|-------|-------|-------|-------|

۱۱- در صورتی که یک جدول Drop شود، DBMS به طور خودکار کدام عمل را انجام می‌دهد؟

(مهندسی کامپیوتر-دولتی ۸۶)

- (۱) حذف دید (View) های مربوط به آن جدول (۲) حذف کلیدهای خارجی مربوط به آن جدول
 (۳) اصلاح لغتنامه دادهها (data dictionary) (۴) گزینههای ۱ و ۲

۱۲- در صورتی که هنگام تعریف جدول SL دستورات زیر نوشته شده باشد و با توجه به جدول L و SL پس از اجرای دستور زیر آن گاه کاردينالیتی جدول L کدام است؟
 (مهندسی IT - آزاد ۸۵)

Delete from L where unit=3

Create Table SL (sname char[20] not NULL, Lname char [20] not NULL,

primary key (Sname, Lname),

foreign key (Lname) references L (Lname),

on delete restrict

on update cascade)

Sname	Lname	Lname	unit
احمدی	ریاضی	ریاضی	4
حسینی	فیزیک	پایگاه	3
رحمانی	پایگاه	تریبیت بدنه	1
عباسی	پایگاه	شبکه	3
		فیزیک	3
جدول SL		جدول L	
4 (۴)	3 (۳)	2 (۲)	5 (۱)

۱۳- جدول (Customers (name , age) و دستور زیر را در نظر بگیرید:

Create Assertion Check-Customer Check

(Not Exists ((Select * From Customers) Except

(Select * From Customers Where name= name OR age = age)))

(مهندسی کامپیوتر- دولتی ۸۹)

Assertion بالا چه چیزی را بررسی می کند؟

۱) جدول Customers خالی نباشد.

۲) همه تابلهای name، یا name یکسان داشته باشند یا age یکسان

۳) در هیچ تابلی از Customers، فیلدهای name یا age مقدار NULL نداشته باشند.

۴) در هیچ تابلی از Customers، فیلدهای name و age هم زمان مقدار NULL نداشته باشند.

۱۴- کدام گزینه در مورد جداول مجازی درست نیست؟
 (مهندسی کامپیوتر- آزاد ۸۹ گروه الف)

۱) جدول مجازی وجود خارجی و فیزیکی ندارد.

۲) می توان به جدول مجازی دسترسی داشت و آن را تغییر داد.

۳) تغییرات جدول مجازی مستقل از تغییرات جدول اصلی است.

۴) هدف جدول مجازی ایجاد جدول خلاصه از اطلاعات موجود و محدود کردن دید کاربران است.

۱۵- کدام یک از گزینههای زیر در مورد جداول مجازی درست است؟
 (مهندسی IT - دولتی ۹۶)

۱) تغییرات جدول مجازی از طریق تغییرات در جداول اصلی رخ می دهد.

۲) جداول مجازی در حافظه وجود خارجی دارند ولی در دیسک سخت ذخیره نمی شوند.

۳) بروزرسانی جداول مجازی در همه شرایط مستقل از نوع تعریف جدول مجازی، امكان پذیر است.

۴) هدف از ایجاد جدول مجازی تنها ساخت جداول خلاصه برای افزایش سرعت جستجو است.

- ۱۶- کدام یک از گزینه‌های زیر برای چک کردن این که آیا وابستگی تابعی $c \rightarrow b$ در رابطه $R(a,b,c,d)$ رعایت می‌شود غلط است؟
 (۱) مهندسی IT - دولتی (۹۳)

```
create assertion b-to-c check (
not exists
(select b
from r
group by b
having count (c)>1
)
)
```

(۲)

```
create assertion b-to-c check (
not exists
(select b
from r as r1 , r as r2
where r1.b=r2.b AND not (r1.c=r2.c)
)
)
```

(۳) کافی است که در زمان ایجاد جدول (create table)، به شرط آن که b کلید رابطه R نیز باشد، این تعريف را نیز اضافه کنیم: (b) primary key.

(۴) کافی است که یک materialized view روی دو صفت b و c تعريف کنیم و مشخص کنیم که b کلید است.

- ۱۷- کدام یک از موارد زیر تفاوت مفاهیم Type و Domain در پایگاه داده را بیان می‌کند:
 (۱) مهندسی کامپیوتر-آزاد (۹۲)

- (۱) Domain می‌تواند دارای مقادیر پیش فرض و نیز constraint باشد.
- (۲) Type نمی‌تواند دارای مقادیر پیش فرض و نیز constraint باشد.
- (۳) دارای خاصیت Strongly Typed است.
- (۴) همه موارد

- ۱۸- دستوراتی که به عنوان عکس العمل یک عمل در پایگاه داده به فرم خودکار اجرا می‌شود، نامیده می‌شود.
 (۱) مهندسی کامپیوتر-آزاد (۹۲)

Stored procedure (۴)	Trigger (۳)	Assertion (۲)	Grant (۱)
----------------------	-------------	---------------	-----------

- ۱۹- کدام یک از عبارات زیر در رابطه با مفهوم View (View) نادرست می‌باشد؟
 (۱) اجرای عملیات select و insert روی view امکان‌پذیر است.
 (۲) تعريف یک view روی یک view دیگر امکان‌پذیر است.
 (۳) هر view دارای ماهیت فیزیکی بوده و می‌تواند روی روابط پایه ایجاد شود.
 (۴) تعريف view به صورت بازگشتی امکان‌پذیر نیست.

- ۲۰- کدام یک از موارد ذیل جزو انواع متنوع مجوزدهی (Authorisation) در سطح پایگاه داده نیست?
 (۱) مهندسی کامپیوتر-آزاد (۹۲)

- | | |
|----------------------------|---------------------------|
| (۲) مجوز بروزرسانی داده‌ها | (۱) مجوز درج داده‌ها |
| (۴) مجوز حذف داده‌ها | (۳) مجوز کپی کردن داده‌ها |

-۲۱- دو جدول **customers** (مشتریان) و **orders** (سفارش‌ها) که با دستورهای زیر ایجاد شده‌اند را در نظر بگیرید:

```
CREATE TABLE customers (cid CHAR(4),
                        cname CHAR(20),
                        PRIMARY KEY (cid))
CREATE TABLE orders (oid CHAR(6),
                     odate CHAR(6),
                     custid CHAR(4),
                     PRIMARY KEY (oid),
                     FOREIGN KEY (custid) REFERENCES customers(cid)
                     ON DELETE CASCADE)
```

در صورتی که تعداد رکوردهای مشتریان برابر ۱۰ و تعداد رکوردهای سفارش‌ها برابر ۱۵ باشد، پس از اجرای دستور زیر:

DELETE

```
FROM customers C
WHERE C.cid='2000'
```

مجموع تعداد رکوردهای دو جدول **customers** و **orders** حداکثر و حداقل چه تعداد خواهد بود؟

(مهندسي کامپيوتر - دولتي ۹۴)

- | | |
|---------------|-----------|
| ۱) حداکثر: 24 | حداقل: 9 |
| ۲) حداکثر: 25 | حداقل: 4 |
| ۳) حداکثر: 24 | حداقل: 18 |
| ۴) حداکثر: 25 | حداقل: 18 |

-۲۲- با وجود رابطه‌های ST, CO و STCO در مورد دیدهای تعریف شده V1 و V2، کدام گزینه از لحظه تئوری صحیح است؟ (مهندسي IT - دولتي ۹۵)

ST (STID, STNAME, STLEVEL, STMAJOR) دانشجویان

CO (COID, CONAME, CREDIT) درس‌ها

STCO (STID, COID, GRADE) درس‌های انتخاب شده توسط دانشجویان

<pre>CREATE VIEW V1 AS SELECT ST.STID, STNAME, COID FROM ST JOIN STCO</pre>	<pre>CREATE VIEW V2 AS SELECT STID, AVG(GRADE) FROM STCO GROUP BY STID</pre>
---	--

(۱) دید V1 و دید V2 هر دو پذیرای عملیات ذخیره‌سازی (Updatable) هستند.

(۲) دید V1 و دید V2 هر دو ناپذیرا (Non-Updatable) هستند.

(۳) دید V1 پذیرا ولی دید V2 ناپذیرا است.

(۴) دید V1 ناپذیرا ولی دید V2 پذیرا است.

-۲۳- فرض کنید که دید V1 بر روی جداول S و SP بصورت زیر تعریف شده باشد. (مهندسي کامپيوتر-دولتي ۹۶)

S (SID, SName, Level, City)

P (PID, PName, Weight)

SP(SID, PID, QTY)

CREATE VIEW V1

```
AS SELECT SID AS ID, SUM(QTY) AS SQTY
      FROM SP JOIN S
     WHERE LEVEL>10
    GROUP BY SID
```

دستور زیر در سطح خارجی با نگاشت E/C تبدیل به چه دستوری در سطح ادراکی می‌شود؟

```
SELECT ID, SQTY
      FROM V1
     WHERE SQTY>1000
```

(۱)

```
SELECT ID,SQTY
      FROM SP JOIN S WHERE LEVEL >10
    GROUP BY SID HAVING SQTY >1000
```

(۲)

```
SELECT ID,SQTY
      FROM SP JOIN S WHERE LEVEL >10 AND SQTY >1000
    GROUP BY ID
```

(۳)

```
SELECT SID AS ID ,SUM(QTY) AS SQTY
      FROM SP JOIN S WHERE LEVEL >10
    GROUP BY SID HAVING SUM(QTY) >1000
```

(۴)

```
SELECT SID AS ID ,SUM(QTY) AS SQTY
      FROM SP JOIN S WHERE LEVEL >10
        AND SUM(QTY)>1000
    GROUP BY SID
```

(مهندسی کامپیووت-دولتی ۹۶)

- دستور Assertion زیر چه محدودیتی را بیان می‌کند؟

S (SID, SName, Level, City) : تولیدکنندگان

P (PID, PName, Weight) : قطعات

SP (SID, PID, QTY) : قطعات تولیدی توسط تولیدکنندگان

CREATE ASSERTION MYCONST

CHECK (**NOT EXISTS (SELECT ***
FROM S

WHERE S.LEVEL<5 AND EXISTS (SELECT *

FROM SP

WHERE SP.SID = S.SID

AND SP.PID = 'p1'))

۱) تولیدکنندگان با سطح کمتر از ۵ نباید قطعات با شناسه p1 را تولید کنند.

۲) فقط تولیدکنندگان با سطح کمتر از ۵ می‌توانند قطعات با شناسه p1 را تولید کنند.

۳) تولیدکنندگان با سطح کمتر از ۵ باید حداقل یک قطعه با شناسه p1 را تولید نمایند.

۴) هر تولیدکننده‌ای که قطعه با شناسه p1 را تولید می‌کند نباید سطوحی بیشتر از ۵ داشته باشد.

۲۵- با توجه به جدول دانشجو با مشخصات شماره‌ی دانشجویی، نام، رشته و دانشکده، زیر چه محدودیت جامعیتی را توصیف می‌نماید؟
 (مهندسی IT- دولتی ۹۶)

STUDENT (STID, Name, Major, Department)

CREATE ASSERTION MYASSERT

CHECK (NOT EXISTS (SELECT DEPARTMENT
 FROM STUDENT
 GROUP BY DEPARTMENT
 HAVING COUNT (DISTINCT (MAJOR))>1))

- ۱) هر دانشجو در چند رشته می‌تواند تحصیل کند.
- ۲) در هر دانشکده فقط یک رشته ارائه می‌شود.
- ۳) هر رشته فقط در یک دانشکده ارائه می‌شود.
- ۴) هر دانشجو یک نام منحصر به فرد دارد.

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

Create Table person (personID int, managerID int, level int, Primary key (personID), Foreign key (managerID) References person(personID) on delete CASCADE on update RESTRICT,)	دستور اول
insert into person values (14,NULL,1), (12,14,2), (11,12,3), (13,12,3), (15,12,3), (10,14,2), (17,10,3), (18,10,3)	دستور دوم
Delete from person where level = 2	دستور سوم

۰ (۴) ۱ (۳) ۸ (۲) ۶ (۱)

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

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

$$M' = M, N' = N, K' = K \quad (2) \quad M' = M - 1, N' = N - 1, K' = K - 1 \quad (1)$$

$$M' < M, N' = N, K' = K \quad (4) \quad M' < M, N' < N, K' = K \quad (3)$$

-۲۸- چه تعداد از گزاره‌های داده شده درست است؟ (مهندس کامپیووتر-دولتی ۹۸)

الف) تعداد کلیدهای کاندید یک رابطه از تعداد ابرکلیدهای آن رابطه همواره کمتر یا مساوی است.

ب) اگر رابطه R دارای n خصیصه باشد، آنگاه تعداد ابرکلیدهای این رابطه حداقل n^2 است.

ج) دستور ALTER Table در SQL، کاتالوگ سیستم را به روز می‌کند.

د) استفاده از View و Index می‌تواند استقلال داده‌ای را افزایش دهد.

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

-۲۹- یک فروشگاه زنجیره‌ای را در نظر بگیرید که در سطح شهر شعب مختلف و هر شعبه تعدادی کارمند دارد.

اسکیمای (شما) جداول‌های شعبه و کارمند داده شده است. فرض کنید، کلید خارجی در جدول کارمند با

حالت On Update Cascade و On Delete Cascade تعريف شده باشد. در حال حاضر شعب با شماره‌های

۳، ۴، ۶ و ۸ در پایگاه داده تعريف شده است. مدیریت فروشگاه قصد دارد کارمندان شعبه ۳ و ۴ را ادغام و

در یک شعبه جدید به شماره ۵ منتقل نماید. (مهندسی IT- دولتی ۹۸)

شعبه: Branch(BranchNo , BranchName , Tel , Address)

کارمند: Staff(StaffNo , FName , LName , BranchNo)

دستورات زیر به چه ترتیبی باید اجرا شوند تا کارمندان فعلی شعب ۳ و ۴ به شعبه ۵ منتقل گردند؟ (ترتیب

گزینه‌ها از چپ به راست است.)

نام دستور	SQL دستور
a	Insert into Branch values (5, 'تهران', 'سادی', 021222324)
b	Update Staff set BranchNo=5 where BranchNo=3
c	Delete From Branch where BranchNo=4
d	Update Staff set BranchNo=5 where BranchNo=4
e	Delete From Branch where BranchNo=3

bedca (۴)

bdace (۳)

acebd (۲)

abdce (۱)

پاسخ تست‌های فصل هفتم: SQL دستورات DCL و DDL

۱- گزینه (۳) صحیح است.

گزینه اول نادرست است. زیرا از دستور Create Table برای تعریف جداول پایگاه داده استفاده می‌شود. گزینه دوم نادرست است. زیرا از دستور Create Domain برای ایجاد دامنه‌های مورد نیاز بهره‌گیری می‌شود. گزینه سوم درست است. زیرا از دستور Create Assertion برای تعریف محدودیت‌های جامعیت عمومی استفاده می‌شود. این دستور به همراه دستوراتی مثل Unique، Check و not NULL برای تعیین محدودیت‌های جامعیتی مورد استفاده قرار می‌گیرد.

گزینه چهارم نادرست است. زیرا از دستور Triggered Procedures برای مدل‌سازی عملیات مورد نیاز روی داده‌ها براساس نظر طراح پایگاه داده مورد استفاده قرار می‌گیرد. این دستورات در شرایط خاص جداول از سوی DBMS به طور اتوماتیک اجرا می‌شوند در حالی که Stored Procedure ها از سوی کاربران فرآخوانی می‌شوند.

۲- گزینه (۲) صحیح است.

به طور کلی می‌توان محتويات یک جدول مجازی را تغییر داد. (درج، حذف و بروزرسانی) اگر شرایط زیر برقرار باشد:

- جدول مجازی فقط روی یک جدول پایه تعریف شده باشد.

- در تعریف جدول مجازی فقط از ستون‌های جدول پایه استفاده شده باشد (از زیر پرس و جوهای تو در تو استفاده نشده باشد).

ستون‌هایی از جدول پایه که در جدول مجازی حضور ندارند، امکان پذیرش مقدار NULL را داشته باشند.

- در تعریف جدول مجازی از دستور Select Distinct استفاده نشده باشد.

- در تعریف جدول مجازی از توابع آماری استفاده نشده باشد.

- در تعریف جدول مجازی از دستور Group by استفاده نشده باشد.

بنابراین وجود توابع آماری و گروه‌بندی در دید مطرح شده، باعث می‌شود تا نتوان تغییرات دید را در جدول پایه ذخیره نمود.

۳- گزینه (۴) صحیح است.

در SQL می‌توان پنج نوع مجوز را به کاربران اعطاء نمود:

۱- **Select**: مجوز خواندن اطلاعات

۲- **Insert**: مجوز درج اطلاعات جدید

۳- **Update**: مجوز بروزرسانی اطلاعات

۴- **Delete**: مجوز حذف اطلاعات

۵- **References**: مجوز ارجاع کلید خارجی یک جدول به کلید کاندید یک جدول مبدأ

برای درج اطلاعات در جدول Watches لازم است تا کاربر مورد نظر مجوز درج را به شکل Insert ON Watches داشته باشد. اما هیچ یک از گزینه‌ها به مجوز درج در جدول Watches اشاره نکرده است. گزینه اول مربوط به مجوز خواندن از جدول Manfs و گزینه دوم مربوط به مجوز خواندن از جدول Watches است. هم‌چنین گزینه سوم مربوط به مجوز بروزرسانی در جدول Watches است.

اما علاوه بر مجوز درج، چون در تعریف جدول Manfs Watches به جدول Manfs ارجاع می‌شود باید مجوز ارجاع به Manfs نیز وجود داشته باشد. بنابراین برای اجرای دستور درج داده شده، هم مجوز درج و هم مجوز ارجاع لازم است.

این مجوزها توسط دستور زیر به کاربر مورد نظر اعطا می‌گردد:

grant References on Manfs to user name
grant Insert on Watches to user name

۴- گزینه (۳) صحیح است.

دستور داده شده ستون‌های A و B از دو جدول R و R' را جدا کرده و با هم اجتماع می‌کند و دید مجازی به نام RR' با دو ستون A و B ایجاد می‌کند. به بیان دیگر دید RR' اجتماع ستون‌های A و B از دو جدول R و R' را استخراج می‌کند. توجه کنید که شرایط سازگاری (تیترهای مشابه) برای عملگر اجتماع برقرار است. زیرا در صورت سؤال ستون‌های B و A دو جدول R و R' یکسان هستند. البته ستون‌های D و C در جدول R و D' و C' در جدول R' متفاوت هستند که ارتباطی به عملگر اجتماع در دید تعریف شده ندارند.

۵- گزینه (۴) صحیح است.

به طور کلی دستور Grant برای واگذاری مجوز به کاربران مورد استفاده قرار می‌گیرد.
ساختار کلی این دستور به صورت زیر است:

<نام کاربر> to <نام جدول> on <فهرست مجوزها>

با توجه به دستور مطرح شده مجوز خواندن اطلاعات (Select) و بروزرسانی اطلاعات (Update)، بر روی ستون‌های empno و empname از جدول employee به کاربری به نام Roya اعطا می‌گردد.

۶- گزینه (۴) صحیح است.

گزینه دوم و سوم نادرست هستند. زیرا دستورات مطرح شده فاقد خطای نحوی و خطای زمان اجرا هستند. گزینه اول نادرست است. زیرا، تعریف و استفاده از جداول مجازی می‌تواند موجب افت کارایی و کاهش سرعت پاسخگویی به پرس‌وجوها شود، زیرا جداول مجازی استقلال وجودی ندانند و وابسته به جداول پایه هستند. در هنگام نیاز به این جداول، DBMS براساس تعریف آن‌ها و محتویات جداول پایه ابتدا آن‌ها را بازسازی می‌کند و سپس مورد استفاده قرار می‌دهد. به این ترتیب سرعت پاسخگویی به پرس‌وجوها بیکاری که به جداول مجازی ارجاع می‌کنند کاهش می‌یابد.

گزینه چهارم درست است. زیرا، دید، ساختار پرس‌وجوها را ساده‌تر و درک آن‌ها را آسان‌تر می‌کند و باعث سهولت در بیان برخی از پرس‌وجوها می‌شود.

۷- گزینه (۲) صحیح است.

برای حذف فیزیکی جداول در SQL2، از دستور Drop Table استفاده می‌گردد، هم‌چنین در حذف فیزیکی جداول، اطلاعات سیستمی مربوط به جداول که در کاتالوگ سیستم (فرهنگ لغات داده‌ها) قرار دارند نیز پاک می‌گردد. برای حذف سطرهای جداول، دستور Delete مورد استفاده قرار می‌گیرد. گزینه‌های دیگر جهت انجام فعالیت‌های ذکر شده در SQL2 وجود دارند.

۸- گزینه (۴) صحیح است.

به طور کلی دستور Revoke برای بازپس‌گیری مجوز از کاربران مورد استفاده قرار می‌گیرد.

ساختار کلی این دستور به صورت زیر است:

<نام کاربر> from <نام جدول> on <فهرست مجوزها>

با توجه به دستور مطرح شده، مجوز بروزرسانی اطلاعات (Update)، بر روی جدول stud از کاربری به نام Amiri بازپس گرفته می‌شود.

۹- گزینه (۱) صحیح است.

به طور کلی دستور Grant برای واگذاری مجوز به کاربران مورد استفاده قرار می‌گیرد.

ساختار کلی این دستور به صورت زیر است:

<نام کاربر> to <نام جدول> on <فهرست مجوزها>

با توجه به دستور مطرح شده، مجوز بروزرسانی اطلاعات (Update)، بر روی جدول stud به کاربری به نام Ahmadi اعطا می‌گردد.

۱۰- گزینه (۲) صحیح است.

طبق تعریف، جدولی که توسط طراح پایگاه داده طراحی و با استفاده از دستور Create Table، ایجاد شده است، جدول پایه نامیده می‌شود.

هم‌چنین امکان هرگونه پردازش مثل درج، حذف و بروزرسانی بر روی جدول پایه برای کاربران در چارچوب قواعد امنیتی امکان‌پذیر است. بنابراین گزاره اول، عبارت درستی است. همچنین نتیجه هر پرس‌وجوه SQL توسط DBMS در یک جدول به نام جدول میانی (Intermediat Table) ذخیره می‌شود. مدیریت جداول میانی بر عهده DBMS است و کاربران فقط امکان خواندن از جداول میانی را دارند. انجام عملیات درج، حذف و بروزرسانی بر روی جداول میانی توسط کاربران امکان‌پذیر نیست. پس گزاره دوم نیز، عبارت درستی است.

مثال:

```
select *
from (select *
       from s
      where city = 'c2')
```

خروجی حاصل از select داخلي جهت استفاده در select خارجي، در یک جدول میانی توسط DBMS ذخیره می‌گردد.

در مورد جدول مجازی می‌توان گفت که جدولی که توسط طراح پایگاه داده طراحی گردیده و با استفاده از دستور Create View ...as برای کاربر فراهم است. اما هرگونه پردازش مثل درج، حذف و بروزرسانی بر روی جدول مجازی تابع پیش شرط‌هایی می‌باشد که در همه شرایط امکان‌پذیر نیست. بر این اساس گزاره سوم عبارت درستی است.

۱۱- گزینه (۳) صحیح است.

اطلاعات سیستمی مربوط به جداول، مانند نام جداول، نوع ستون‌های جداول در لغتنامه داده‌ها ذخیره

می‌گردد. حذف فیزیکی یک جدول توسط دستور Drop باعث می‌شود که به طور خودکار اطلاعات سیستمی (شناسنامه جدول) آن جدول از لغت‌نامه حذف گردد و در نتیجه لغت‌نامه داده‌ها اصلاح شود.

۱۲- گزینه (۴) صحیح است.

برای رفتار ستون کلید خارجی در یک جدول مقصود، در قبال تغییرات کلید کاندید از یک جدول مبدأ Create Table نام جدول گزینه‌های زیر وجود دارد:

()

نام ستون‌ها

:

foreign key...references...

on delete option

on update option

:

)

فیلد option می‌تواند یکی از موارد زیر باشد:

(restrict) no action

گزینه پیش فرض است و هیچ عملی انجام نمی‌شود. یعنی اگر بر اثر عملیات حذف، درج و بروزرسانی، قوانین جامعیت نقض گردد، این اعمال انجام نمی‌گردد. در واقع زمانی عملیات حذف، درج و بروزرسانی انجام می‌گردد که قوانین جامعیت نقض نگردند.

cascade

اگر سطرهای جدول مرجع حذف یا بروزرسانی شود، کلید خارجی جدولی که به آن ارجاع کرده است نیز حذف یا بروزرسانی خواهد شد.

set NULL

همان روش NULLIFY است که اگر سطرهای جدول مرجع حذف یا بروزرسانی شود، کلید خارجی جدولی که به آن ارجاع کرده است با مقدار NULL پُر می‌شود. با توجه به توضیحات فوق و دستور مطرح شده به شکل زیر:

```
Delete
from L
where unit=3
```

داریم:

در تعریف جدول SL، ستون Lname به عنوان کلید خارجی تعریف شده است که به جدول L ارجاع می‌کند. چون در تعریف جدول SL از دستور on delete restrict استفاده شده است، در عمل امکان حذف هیچ یک از سطرهای جدول L که مورد مراجعه سطرهای جدول SL هستند وجود ندارد. در دستور SQL داده شده حذف دروس سه واحدی از جدول L خواسته شده است. بررسی محتویات جدول L نشان می‌دهد که سه درس سه واحدی (پایگاه، شبکه و فیزیک) در این جدول وجود دارد. اما فقط درس شبکه مورد مراجعه سطرهای جدول SL نیست. بنابراین فقط امکان حذف همین سطر وجود دارد و دو سطر دیگر را نمی‌توان حذف کرد. به این ترتیب بعد از اجرای دستور حذف، چهار سطر، در جدول L باقی خواهد ماند. در صورتی که دستور on delete cascade در تعریف جدول SL ذکر می‌شد، دروس پایگاه، شبکه و

فیزیک از جدول L حذف می‌شدن و کلیه سطرهای ارجاع کننده به این دروس در جدول SL هم حذف می‌شد و در نهایت دو سطر در جدول L باقی می‌ماند.

۱۳- گزینه (۴) صحیح است.

نکته‌ای که در حل سؤال باید از آن استفاده کرد این است که حاصل مقایسه $\text{NULL} = \text{NULL}$ مساوی با FALSE است.

با توجه به نکته‌ی فوق، خروجی عبارت select دوم، تمام سطرهای جدول customers به جز سطرهایی است که فیلدهای name و age آن‌ها هم‌zman NULL است.

دستور select اول نیز تمام سطرهای جدول customers را برابر می‌گرداند.

تفاضل خروجی‌های دو دستور select سطرهایی هستند که فیلدهای name و age آن‌ها هم‌zman NULL است. اگر چنین سطرهایی در جدول وجود نداشته باشد دستور not exists مقدار TRUE بر می‌گردد.

۱۴- گزینه (۳) صحیح است.

هر جداول مجازی (یا View) یک دستور select روی یک یا چند جدول است که خلاصه‌ای از اطلاعات آن جداول را به دست می‌دهد و اگر به کاربر فقط اجازه دسترسی به این جدول مجازی را بدهیم کاربر به بقیه‌ی اطلاعات موجود در جداول اصلی دسترسی ندارد، یعنی دید کاربر محدود می‌شود. (گزینه چهارم) از آنجایی که هر جدول مجازی معادل یک دستور select است، نیازی به ذخیره‌سازی سطرهای خروجی آن دستور select نیست، چون در هر زمان می‌توان آن‌ها را از روی جداول اصلی مربوطه به دست آورد. به علاوه، با هر تغییر در اطلاعات جداول اصلی مربوطه، باید اطلاعات متناظر با آن‌ها در جدول مجازی نیز تغییر کند. بنا به این دلایل، نیازی به ذخیره‌سازی اطلاعات جداول مجازی نیست. پس جداول مجازی وجود خارجی و فیزیکی ندارند. (گزینه اول)

همانند جداول اصلی بانک اطلاعاتی، می‌توان به جداول مجازی نیز دسترسی داشت و اطلاعات آن‌ها را تغییر داد یا آن‌ها را با جداول اصلی یا مجازی دیگر پیوند زد و روی آن‌ها عمل select انجام داد (گزینه دوم). از آنجایی که جداول مجازی وجود خارجی ندارند و خلاصه‌ی اطلاعاتی هستند که از روی جداول اصلی مربوطه به دست می‌آیند، هر تغییر بر روی یک جدول مجازی بر روی جداول مربوط به آن اعمال می‌شود. پس عبارت گزینه سوم اشتباه است.

۱۵- گزینه (۱) صحیح است.

گزینه اول صحیح است: جدول مجازی داده‌های ذخیره شده خاص خود ندارد، بنابراین با انجام تغییرات در جدول یا جداول اصلی، اطلاعات جدول مجازی هم تغییر می‌کند.

گزینه دوم غلط است: جداول مجازی معمولاً وجود خارجی ندارند (یعنی رکوردهای آن‌ها در جایی - نه هاردیسک و نه حافظه RAM - ذخیره نمی‌شوند). هرگونه دسترسی به جداول مجازی توسط DBMS به جداول اصلی مرتبط می‌شود.

گزینه سوم غلط است: در صورتی می‌توان جدول مجازی را بروزرسانی کرد که انجام تغییرات مربوطه روی جداول اصلی بدون ابهام باشد. برخی نسخه‌های SQL این کار را به طور کلی غیرمجاز می‌دانند. اما در SQL2 در شرایط خاصی این کار قابل انجام است (جدول مجازی روی یک جدول باشد، در ساختن آن از Select distinct استفاده نشده باشد، و جدول مربوطه در subquery نیامده باشد).

گزینه چهارم غلط است: هدف اصلی از ایجاد جداول مجازی، ایجاد جداول خلاصه از اطلاعات موجود و محدود کردن دید کاربران است.

۱۶- گزینه (۱) صحیح است.

وابستگی تابعی $c \rightarrow b$ به این معنی است که به هر مقدار از b فقط یک مقدار از c مرتبط باشد. برای مثال در جدول زیر به هر مقدار b فقط یک مقدار c مرتبط است و بنابراین وابستگی تابعی $c \rightarrow b$ رعایت شده است (عدد ۲ = $b = 3$ فقط است و عدد ۳ = $b = 1$ فقط با $c = 1$ مرتبط است).

a	b	c	d
1	2	3	4
2	2	3	5
3	3	1	6
4	3	1	7
5	3	1	8

گزینه اول جدول ۲ را بحسب فیلد b گروه‌بندی می‌کند و برای هر گروه (یعنی برای هر مقدار b) تعداد مقادیر c را می‌شمرد. توجه کنید که در دستور count جلوی having از distinct استفاده نشده است و بنابراین مقادیر تکراری چند بار شمرده می‌شوند.

در نتیجه این‌طور تصور می‌شود که برای $b = 2$ دو مقدار از c و برای $b = 3$ سه مقدار از c وجود دارد و بنابراین به اشتباه تصور می‌شود که وابستگی تابعی $c \rightarrow b$ رعایت نشده است.

اگر دستور count با Distinct می‌باشد. این گزینه صحیح بود:
Having count (distinct c)>1
گزینه دوم چک می‌کند که در جدول ۲ دو سطر با مقدار b یکسان و مقدار c متفاوت وجود نداشته باشد. یعنی چک می‌کند هر مقدار b به یک مقدار از c مرتبط باشد. پس وابستگی تابعی $c \rightarrow b$ را به درستی بررسی می‌کند.

گزینه سوم صفت b را کلید می‌کند و بنابراین مقادیر b تکراری نخواهد بود، در نتیجه مطمئناً یک مقدار از b نمی‌تواند دوبار ظاهر شود و با دو مقدار متفاوت از c مرتبط شود.
materialized view شیءای در بانک اطلاعاتی است که حاصل یک query را در خود نگه می‌دارد با ایجاد چنین شیءای روی صفات b و c و کلید کردن صفت b ، مقادیر b تکراری نخواهد بود و بنابراین هر مقدار b فقط یک بار ظاهر می‌شود و مطمئناً هر مقدار b با یک مقدار c مرتبط خواهد بود.

۱۷- گزینه (۴) صحیح است.

در هنگام تعریف یک Domain (با دستور CREATE DOMAIN)، بر طبق گرامر SQL، می‌توان برای آن Domain مقدار پیش فرض و هم‌چنین constraint تعیین کرد. بعلاوه، در دستور CREATE DOMAIN لازم است نوع مقادیر آن DOMAIN حتماً تعیین شوند. بنابراین، Domain، Strongly-Typed، اما در هنگام تعریف یک Type (با دستور CREATE TYPE)، طبق گرامر SQL، امکان تعریف مقدار پیش فرض یا Constraint وجود ندارد.

۱۸- گزینه (۳) صحیح است.

stored procedure توسط کاربر و trigger براساس شرط‌هایی از قبل تعیین شده به شکل خودکار اجرا می‌شوند.

۱۹- گزینه (۳) صحیح است.

روی view می‌توان select زد. اما عمل insert را نمی‌توان روی هر view انجام داد. مثلاً وقتی view روی چند جدول زده شود امکان insert وجود ندارد. یا زمانی که دستور insert منجر به NULL شدن مقدار کلید اصلی یا بخشی از کلید اصلی گردد که سبب نقض قانون جامعیت موجودیت می‌گردد. بنابراین گزینه اول درست است.

هر view روی روابط پایه یا روی view های دیگر بنا می‌شود اما viewها ماهیت فیزیکی و وجود خارجی ندارند و رکوردهای آن‌ها به طور فیزیکی ذخیره نمی‌شوند بلکه در هنگام اجرا و استفاده از آن view دستورات select مربوط به آن view اجرا می‌شوند و اطلاعات از روی جداول پایه واکنشی می‌شوند.

بنابراین گزینه سوم نیز نادرست است.
گزینه‌های دوم و چهارم عبارات صحیحی هستند.

۲۰- گزینه (۳) صحیح است.

با دستور GRANAT در زبان SQL می‌توان مجوزهای Insert، Update، Select و Delete را به یک نفر داد، اما مجوزی برای Copy وجود ندارد.

۲۱- گزینه (۲) صحیح است.

دو جدول orders و customers، با مقادیر زیر را در نظر بگیرید:

customers		orders		
کلید کاندید (اصلی)		کلید خارجی		
<u>cid</u>	Cname	<u>Oid</u>	Odate	custid
جدول		جدول		

مطابق دستورات Create Table برای دو جدول customers و orders ستون cid توسط دستور Primary key(cid) به عنوان کلید اصلی برای جدول customers انتخاب می‌گردد. همچنین ستون oid توسط دستور Primary key(oid) به عنوان کلید اصلی برای جدول orders انتخاب می‌گردد. همچنین ستون custid References customers (cid) ON Delete Cascade توسط دستور Foreign key (custid) به عنوان custid کلید خارجی برای جدول orders انتخاب می‌گردد. ستون custid از جدول orders به عنوان کلید خارجی به ستون cid از جدول customers مطابق ساختار تعريف جدول orders، به شکل On Delete Cascade ارجاع می‌کند.

برای رفتار ستون کلید خارجی در یک جدول مقصود، در قبال تغییرات کلید کاندید از یک جدول مبداء Create Table نام جدول گزینه‌های زیر وجود دارد:

```
(  
    نام ستون‌ها  
    :  
    foreign key ... references ...  
    on delete option  
    on update option  
    :  
)
```

فیلد option می‌تواند یکی از موارد زیر باشد:

(restrict) no action

گزینه پیش فرض است و هیچ عملی انجام نمی‌شود. یعنی اگر بر اثر عملیات حذف، درج و بروزرسانی، قوانین جامعیت نقض گردند، این اعمال انجام نمی‌گردند. در واقع زمانی عملیات حذف، درج و بروزرسانی انجام می‌گردند که قوانین جامعیت نقض نگردند.

Cascade

اگر سطح‌های جداول مرجع حذف یا بروزرسانی شود، کلید خارجی جدولی که به آن ارجاع کرده است نیز حذف یا بروزرسانی خواهد شد.

Set NULL

همان روش NULLIFY است که اگر سطح‌های جدول مرجع حذف یا بروزرسانی شود، کلید خارجی جدولی که به آن ارجاع کرده است با مقدار NULL پر می‌شود.

توجه: مطابق فرض سوال تعداد رکوردهای جدول customers برابر 10 و تعداد رکوردهای جدول orders برابر 15 است.

دستور Delete داده شده، ممکن است صفر یا حداقل یک رکورد را از جدول customers حذف کند (چون ممکن است هیچ رکوردی در جدول customers با 'cid = 2000' وجود نداشته باشد. اگر هم رکوردی با 'cid = 2000' وجود داشته باشد، چون کلید اصلی است، پس فقط همان یک رکورد وجود دارد).

در صورتیکه هیچ رکوردی از جدول customers حذف نشود (یعنی رکوردی با 'cid = 2000' در جدول customers وجود نداشته باشد)، مطمئناً رکوردی هم از جدول orders حذف نخواهد شد. چون بنابر قانون

جامعیت ارجاعی هیچ رکوردی با 'custid = 2000' نمی‌تواند وجود داشته باشد، چون مطابق قانون جامعیت ارجاعی همواره باید مقادیر کلید خارجی زیر مجموعه کلید کاندید (اصلی) در جدول مرجع باشد.

بنابراین مجموع تعداد رکوردهای ثابت باقی می‌مانند و در حداقل مقادیر خود برابر $25 - 10 - 15 = 0$ می‌گردد. از سوی دیگر، در صورتیکه رکوردی با 'cid = 2000' در جدول customers وجود داشته باشد با اجرای دستور Delete آن یک رکورد حذف می‌شود و در جدول customers تعداد 9 رکورد باقی می‌ماند. حال سه

وضعیت می‌تواند رخ دهد:

وضعیت اول: حالی که هیچ یک از رکوردهای جدول orders مقدار 'custid = 2000' نداشته باشد، در این شرایط هیچ رکوردی از جدول orders حذف نمی‌شود، بنابراین مجموع رکوردهای باقی مانده برابر $9 + 15 = 24$ خواهد شد.

وضعیت دوم: حالی که فقط یک رکورد از جدول orders مقدار 'custid = 2000' داشته باشد، در این شرایط فقط یک رکورد از جدول orders حذف می‌شود، بنابراین مجموع رکوردهای باقی مانده برابر $9 + 14 = 23$ خواهد شد.

وضعیت سوم: حالی که همه رکوردهای از جدول orders، مقدار 'custid = 2000' داشته باشند، در این شرایط همه رکوردهای جدول orders حذف می‌شوند، بنابراین مجموع رکوردهای باقی مانده برابر $9 + 0 = 9$ خواهد شد.

پس به طور کلی، مجموع تعداد رکوردهای دو جدول customers و orders در حداقل مقدار برابر 25 و در حداقل مقدار برابر 9 خواهد بود.

۲۲- گزینه (۳) صحیح است.

جداول مجازی (دیدها) تصویری از جداول حقیقی (پایه) هستند، یعنی توسط سیستم به جداول پایه متصل می‌شوند و وجود خارجی ندارند. دسترسی به آنها از دید کاربر مستقیم ولی از دید سیستم غیرمستقیم است، یعنی سیستم، هرگونه استخراج اطلاعات را از روی جداول پایه انجام می‌دهد. محتوای دید در لحظه اجرای دید، تولید می‌شود. این محتوای تولید شده از سوی دید، در جایی ذخیره نمی‌گردد، بلکه در هر لحظه بر اساس ساختار دید از روی جداول پایه، استخراج می‌گردد.

به طور کلی دو عمل ذخیره‌سازی (درج، حذف و بروزرسانی) و بازیابی بر روی دیدها انجام می‌شود. که عمل ذخیره‌سازی با قید و شرط انجام می‌شود یعنی نه همیشه، اما عمل بازیابی بسیار قید و شرط انجام می‌شود یعنی همیشه. بنابراین مشکل اصلی در دیدها، عمل ذخیره‌سازی (درج، حذف و بروزرسانی) است که قید و شرط دارد و همیشه قابل انجام نیست. به طور کلی پذیرا بودن ذخیره‌سازی (درج، حذف و بروزرسانی) بر روی دیدها به سه دسته زیر تقسیم می‌گردد:

پذیرا از نظر تئوری و عملی

در این حالت دید از نظر تئوری پذیرای ذخیره‌سازی است، یعنی قوانین مدل رابطه‌ای همچون داشتن حداقل یک کلید کاندید و داشتن ستون‌های واقعی و نه مجازی حفظ می‌گردد و همچنین تناظر سطر به سطر مابین دید و جدول پایه حفظ می‌گردد. و از نظر عملی هم پذیرای ذخیره‌سازی است به دلیل رعایت قوانین جامعیت بانک در مدل رابطه‌ای در حین اجرا.

موارد زیر پذیرا بودن ذخیره‌سازی بر روی دید را از نظر تئوری تضمین می‌کند:

- در تعریف دید فقط از یک جدول پایه با حداقل یک کلید کاندید استفاده شده باشد.

اگر رعایت نشود حفظ قوانین مدل رابطه‌ای همچون داشتن حداقل یک کلید کاندید و حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.

- در تعریف دید از دستور select distinct استفاده نشده باشد.

اگر رعایت نشود حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.

- در تعریف دید از دستور group by استفاده نشده باشد.

اگر رعایت نشود حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.

- در تعریف دید از توابع آماری استفاده نشده باشد.

اگر رعایت نشود حفظ قوانین مدل رابطه‌ای همچون داشتن ستون‌های واقعی و نه مجازی نقض می‌شود.

- در تعریف دید از زیرپرس و جوهای تو در تو استفاده نشده باشد.

اگر رعایت نشود حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.

مورد زیر پذیرا بودن ذخیره‌سازی بر روی دید را از نظر عملی تضمین می‌کند:

- در هنگام اجرای ذخیره‌سازی بر روی دید، قوانین جامعیت بانک نقض نگردد.

سه جدول ST، CO و STCO با مقادیر زیر را در نظر بگیرید:

STID	STNAME	STLEVEL	STMAJOR	COID	CONAME	CREDIT
S1	Sn1			C1	Cn1	
S2	Sn2			C2	Cn2	
S3	Sn3			C3	Cn3	

جدول ST

جدول CO

STID	COID	GRADE
S1	C1	18
S1	C3	20
S2	C2	16
S2	C3	14

جدول STCO

مثال: دید زیر را در نظر بگیرید:

```
Create view GSTCO
AS Select STID, COID
  From STCO
  Where GRADE>10
```

در تعریف دید GSTCO از یک جدول پایه استفاده شده است، بدنه دید GSTCO به صورت زیر است:

STID	COID
S1	C1
S1	C3
S2	C2
S2	C3

آنچه واضح است، این است که ساختار دید GSTCO از نظر تئوری پذیرای ذخیره‌سازی است، یعنی قوانین مدل رابطه‌ای همچون داشتن حداقل یک کلید کاندید و داشتن ستون‌های واقعی و نه مجازی حفظ می‌گردد و همچنین تناظر سطر به سطر مابین دید و جدول پایه حفظ می‌گردد.

فرض کنید کاربر با اجرای دستور زیر بخواهد یک سطر جدید را در دید GSTCO درج نماید:

```
Insert into GSTCO
```

```
Value ('S3', 'C3', 20)
```

از آنجا که دید GSTCO استقلال وجودی ندارد، عملیات درج فوق در واقع روی جدول پایه STCO انجام می‌گردد.

در واقع دستور فوق به هنگام اجرا، به دستور زیر تبدیل می‌گردد:

```
Insert into STCO
```

```
Value ('S3', 'C3', 20)
```

در هنگام ذخیره‌سازی عملی بر روی دید GSTCO قوانین جامعیت بانک رعایت می‌شود. پس اجرای دستور ذخیره‌سازی بر روی دید GSTCO به صورت عملی از سوی DBMS پذیرفته می‌شود. بنابراین ذخیره‌سازی بر روی دید GSTCO به صورت عملی قابل انجام است به دلیل رعایت قوانین جامعیت بانک در مدل رابطه‌ای.

پذیرا از نظر تئوری و غیرپذیرا از نظر عملی

در این حالت دید از نظر تئوری پذیرای ذخیره‌سازی است، یعنی قوانین مدل رابطه‌ای همچون داشتن حداقل یک کلید کاندید و داشتن ستون‌های واقعی و نه مجازی حفظ می‌گردد و همچنین تناظر سطر به سطر مابین دید و جدول پایه حفظ می‌گردد. اما از نظر عملی پذیرای ذخیره‌سازی نیست به دلیل عدم رعایت قوانین جامعیت بانک در مدل رابطه‌ای در حین اجرا.

موارد زیر پذیرا بودن ذخیره‌سازی بر روی دید را از نظر تئوری تضمین می‌کند:

- در تعریف دید فقط از یک جدول پایه با حداقل یک کلید کاندید استفاده شده باشد.

- اگر رعایت نشود حفظ قوانین مدل رابطه‌ای همچون داشتن حداقل یک کلید کاندید و حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.
- در تعریف دید از دستور `select distinct` استفاده نشده باشد.
 - اگر رعایت نشود حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.
 - در تعریف دید از دستور `group by` استفاده نشده باشد.
 - اگر رعایت نشود حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.
 - در تعریف دید از توابع آماری استفاده نشده باشد.
 - اگر رعایت نشود حفظ قوانین مدل رابطه‌ای همچون داشتن ستون‌های واقعی و نه مجازی نقض می‌شود.
 - در تعریف دید از زیرپرس و جوهای تو در تو استفاده نشده باشد.
 - اگر رعایت نشود حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.
 - مورد زیر پذیرا بودن ذخیره‌سازی بر روی دید را از نظر عملی تضمین می‌کند:
 - در هنگام اجرای ذخیره‌سازی بر روی دید، قوانین جامعیت بانک نقض نگردد.
- سه جدول STCO و CO با مقادیر زیر را در نظر بگیرید:

STID	STNAME	STLEVEL	STMAJOR	COID	CONAME	CREDIT
S1	Sn1			C1	Cn1	
S2	Sn2			C2	Cn2	
S3	Sn3			C3	Cn3	

جدول ST

جدول CO

STID	COID	GRADE
S1	C1	18
S1	C3	20
S2	C2	16
S2	C3	14

جدول STCO

مثال: دید زیر را در نظر بگیرید:

Create view GSTCO

AS Select STID, COID

From STCO

Where GRADE>10

در تعریف دید GSTCO از یک جدول پایه استفاده شده است، بدنه دید GSTCO به صورت زیر است:

STID	COID
S1	C1
S1	C3
S2	C2
S2	C3

آنچه واضح است، این است که ساختار دید GSTCO از نظر تئوری پذیرای ذخیره‌سازی است، یعنی قوانین مدل رابطه‌ای همچون داشتن حداقل یک کلید کاندید و داشتن ستون‌های واقعی و نه مجازی حفظ می‌گردد

و همچنین تناظر سطر به سطر مابین دید و جدول پایه حفظ می‌گردد.

فرض کنید کاربر با اجرای دستور زیر بخواهد یک سطر جدید را در دید GSTCO درج نماید:

Insert into GSTCO

Value ('S3', 'C3')

از آنجا که دید GSTCO استقلال وجودی ندارد، عملیات درج فوق در واقع روی جدول پایه STCO انجام می‌گردد. اما جدول STCO دارای سه ستون است، درحالیکه، فقط دو ستون در دستور فوق مربوط به جدول STCO مقداردهی شده‌است. بنابراین برای ستون GRADE مقدار NULL در نظر گرفته می‌شود. فرض کنید در هنگام تعریف جدول STCO برای ستون GRADE خاصیت NOT NULL در نظر گرفته شده باشد، یعنی حتماً می‌بایست ستون GRADE دارای مقدار باشد و مقدار NULL برای آن قابل پذیرش نیست. بنابراین مطابق قانون جامعیت بانک نباید ستون GRADE با NULL مقداردهی شود.

در واقع دستور فوق به هنگام اجرا، به دستور زیر تبدیل می‌گردد:

Insert into STCO

Value ('S3', 'C3', NULL)

در هنگام ذخیره‌سازی عملی بر روی دید GSTCO مطابق قانون جامعیت بانک در جداول نمی‌توان در ستون GRADE از جدول STCO که خاصیت NOT NULL برای آن در نظر گرفته شده‌است، مقدار NULL را درج کرد، پس اجرای دستور ذخیره‌سازی بر روی دید GSTCO به صورت عملی از سوی DBMS پذیرفته نمی‌شود و رد می‌شود. بنابراین ذخیره‌سازی بر روی دید GSTCO به صورت عملی قابل انجام نیست به دلیل عدم رعایت قوانین جامعیت بانک در مدل رابطه‌ای.

همچنین فرض کنید کاربر با اجرای دستور زیر بخواهد یک سطر جدید را در دید GSTCO درج نماید:

Insert into GSTCO

Value ('S10', 'C10', 20)

از آنجا که دید GSTCO استقلال وجودی ندارد، عملیات درج فوق در واقع روی جدول پایه STCO انجام می‌گردد.

در واقع دستور فوق به هنگام اجرا، به دستور زیر تبدیل می‌گردد:

Insert into STCO

Value ('S10', 'C10', 20)

در هنگام ذخیره‌سازی عملی بر روی دید GSTCO هر چند هر سه ستون جدول STCO به واسطه دستور درج، مقداردهی شده‌اند، اما این بار قانون جامعیت ارجاعی نقض شده است، زیرا ستون‌های COID و STID به عنوان کلیدهای خارجی در جدول STCO تعریف شده‌اند و مطابق تعریف محتوایی کلید خارجی و قانون جامعیت ارجاعی، مجموعه مقادیر کلید خارجی باید همواره زیر مجموعه، مجموعه مقادیر کلید کاندید معادل خود باشد تا ارجاع NULL ایجاد نگردد، یعنی مجموعه مقادیر کلید خارجی STID در جدول STCO باید همواره زیر مجموعه، مجموعه مقادیر کلید کاندید معادل خود یعنی STID در جدول ST باشد و مجموعه مقادیر کلید خارجی COID در جدول STCO باید همواره زیر مجموعه، مجموعه مقادیر کلید کاندید معادل خود یعنی COID در جدول CO باشد، که درج مقادیر S10 و C10 در جدول STCO این قاعده را نقض می‌کند، پس اجرای دستور ذخیره‌سازی بر روی دید GSTCO به صورت عملی از سوی DBMS پذیرفته نمی‌شود و رد می‌شود. بنابراین ذخیره‌سازی بر روی دید GSTCO به صورت عملی قابل انجام نیست به دلیل عدم رعایت قوانین جامعیت بانک در مدل رابطه‌ای.

همچنین فرض کنید کاربر با اجرای دستور زیر بخواهد یک سطر جدید را در دید GSTCO درج نماید:

Insert into GSTCO

Value ('S1', 'C1', 20)

از آنجا که دید GSTCO استقلال وجودی ندارد، عملیات درج فوق در واقع روی جدول پایه STCO انجام می‌گردد.

در واقع دستور فوق به هنگام اجرا، به دستور زیر تبدیل می‌گردد:

Insert into STCO

Value ('S1', 'C1', 20)

در هنگام ذخیرهسازی عملی بر روی دید GSTCO هر چند هر سه ستون جدول STCO به واسطه دستور درج، مقداردهی شده‌اند، اما این‌بار قانون جامعیت درون رابطه‌ای نقض شده است، زیرا ستون‌های STID و COID باهم به عنوان کلید کاندید در جدول STCO تعریف شده‌اند و مطابق تعریف محتوایی کلید کاندید، مقادیر کلید کاندید باید یکتا باشد، که درج مجدد مقادیر S1 و C1 به طور همزمان در قالب یک سطر در جدول STCO این قاعده را نقض می‌کند، پس اجرای دستور ذخیرهسازی بر روی دید GSTCO به صورت عملی از سوی DBMS پذیرفته نمی‌شود و رد می‌شود. بنابراین ذخیرهسازی بر روی دید GSTCO به صورت عملی قابل انجام نیست به دلیل عدم رعایت قوانین جامعیت بانک در مدل رابطه‌ای.

دید V1 مطرح شده در صورت سوال به صورت زیر است:

CREATE VIEW V1

AS SELECT ST.STID, STNAME, COID

FROM ST JOIN STCO

در تعریف دید V1 از الحاق طبیعی استفاده شده است، بدنه دید V1 به صورت زیر است:

ST.STID	STNAME	COID
S1	Sn1	C1
S1	Sn1	C3
S2	Sn2	C2
S2	Sn2	C3

آنچه واضح است، این است که ساختار دید V1 از نظر تئوری پذیرای ذخیرهسازی است، یعنی قوانین مدل رابطه‌ای همچون داشتن حداقل یک کلید کاندید و داشتن ستون‌های واقعی و نه مجازی حفظ می‌گردد و همچنین تناظر سطر به سطر مابین دید و جدول پایه حفظ می‌گردد.

فرض کنید کاربر با اجرای دستور زیر بخواهد یک سطر جدید را در دید V1 درج نماید:

Insert into V1

Value ('S4', 'Sn4', 'C4')

از آنجا که دید V1 استقلال وجودی ندارد، عملیات درج فوق در واقع روی جداول پایه ST و STCO انجام می‌گردد.

در واقع دستور فوق به هنگام اجرا، به دستورات زیر می‌باشد تبدیل گردد:

Insert into ST

Value ('S4', 'Sn4', NULL, NULL)

Insert into STCO

Value ('S4', 'C4', NULL)

Insert into CO

Value ('C4', NUUL, NULL)

در هنگام ذخیره‌سازی عملی بر روی دید V1، برای مثال برای حفظ قوانین جامعیت بانک می‌بایست در چند جای مختلف از بانک درجه‌هایی براساس سه دستور فوق انجام گردد، که SQL در عمل توانایی انجام چنین کاری را ندارد، شناخت و اجرای دستورات فوق برای SQL امکان‌پذیر نیست، از آنجاکه عدم اجرای هر یک از دستورات فوق جامعیت بانک را برهم می‌زند، و SQL هم توانایی شناخت و اجرای همه دستورات فوق را ندارد، پس اجرای دستور فوق یعنی ذخیره‌سازی بر روی دید V1 به صورت عملی از سوی DBMS پذیرفته نمی‌شود و رد می‌شود. بنابراین ذخیره‌سازی بر روی دید V1 به صورت عملی قابل انجام نیست به دلیل عدم رعایت قوانین جامعیت بانک در مدل رابطه‌ای.

غیر پذیرا از نظر تئوری و عملی

در این حالت دید از نظر تئوری پذیرای ذخیره‌سازی نیست، یعنی قوانین مدل رابطه‌ای همچون داشتن حداقل یک کلید کاندید و داشتن ستون‌های واقعی و نه مجازی حفظ نمی‌گردد و همچنین تناظر سطر به سطر مابین دید و جدول پایه حفظ نمی‌گردد. همچنین از نظر عملی هم پذیرای ذخیره‌سازی نیست به دلیل عدم پذیرای بودن ذخیره‌سازی در دید از نظر تئوری.

موارد زیر پذیرای بودن ذخیره‌سازی بر روی دید را از نظر تئوری تضمین می‌کند:

- در تعریف دید فقط از یک جدول پایه با حداقل یک کلید کاندید استفاده شده باشد.

اگر رعایت نشود حفظ قوانین مدل رابطه‌ای همچون داشتن حداقل یک کلید کاندید و حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.

- در تعریف دید از دستور select distinct استفاده نشده باشد.

اگر رعایت نشود حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.

- در تعریف دید از دستور group by استفاده نشده باشد.

اگر رعایت نشود حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.

- در تعریف دید از توابع آماری استفاده نشده باشد.

اگر رعایت نشود حفظ قوانین مدل رابطه‌ای همچون داشتن ستون‌های واقعی و نه مجازی نقض می‌شود.

- در تعریف دید از زیرپرس و جوهای تو در تو استفاده نشده باشد.

اگر رعایت نشود حفظ تناظر سطر به سطر مابین دید و جدول پایه نقض می‌شود.

مورود زیر پذیرای بودن ذخیره‌سازی بر روی دید را از نظر عملی تضمین می‌کند:

- در هنگام اجرای ذخیره‌سازی بر روی دید، قوانین جامعیت بانک نقض نگردد.

سه جدول ST، CO و STCO با مقادیر زیر را در نظر بگیرید:

STID	STNAME	STLEVEL	STMAJOR	COID	CONAME	CREDIT
S1	Sn1			C1	Cn1	
S2	Sn2			C2	Cn2	
S3	Sn3			C3	Cn3	

جدول ST

جدول CO

STID	COID	GRADE
S1	C1	18
S1	C3	20
S2	C2	16
S2	C3	14

جدول STCO

دید V2 مطرح شده در صورت سوال به صورت زیر است:

```
CREATE VIEW V2
    AS SELECT STID, AVG(GRADE)
    FROM STCO
    GROUP BY STID
```

در تعریف دید V2 از یک جدول پایه، دستور group by وتابع آماری avg استفاده شده است، بدنه دید V2 به صورت زیر است:

STID	no column name
S1	19
S2	15

آنچه واضح است، این است که دید V2 از نظر تئوری پذیرای ذخیره‌سازی نیست، یعنی قوانین مدل رابطه‌ای همچون داشتن حداقل یک کلید کاندید و داشتن ستون‌های واقعی و نه مجازی حفظ نمی‌گردد و همچنین تناظر سطر به سطر مابین دید و جدول پایه حفظ نمی‌گردد.

فرض کنید کاربر با اجرای دستور زیر بخواهد یک سطر جدید را در دید V2 درج نماید:

Insert into V2

Value ('S3', 12)

در هنگام ذخیره‌سازی عملی بر روی دید V2، برای مثال، مقصد مقدار 12 مشخص نیست، زیرا مقصد آن یک ستون مجازی است و نه یک ستون واقعی، اگر به فرض محال هم درج شود میانگین نمرات دانشجو S3 برابر 12 است حال آنکه دروس آن مشخص نیست، پس اجرای دستور ذخیره‌سازی بر روی دید V2 به صورت عملی از سوی DBMS پذیرفته نمی‌شود و رد می‌شود. بنابراین ذخیره‌سازی بر روی دید V2 به صورت عملی قابل انجام نیست به دلیل عدم پذیرا بودن ذخیره‌سازی در دید V2 از نظر تئوری.

۲۳- گزینه (۳) صحیح است.

جداوی مجازی (دیدها) تصویری از جداول حقیقی (پایه) هستند، یعنی توسط سیستم به جداول پایه متصل می‌شوند و وجود خارجی ندارند. دسترسی به آنها از دید کاربر مستقیم ولی از دید سیستم غیرمستقیم است، یعنی سیستم، هرگونه استخراج اطلاعات را از روی جداول پایه انجام می‌دهد. محتوای دید در لحظه‌ی اجرای دید، تولید می‌شود. این محتوای تولید شده از سوی دید، در جایی ذخیره نمی‌گردد، بلکه در هر لحظه بر اساس ساختار دید از روی جداول پایه، استخراج می‌گردد.

مطرح شده در صورت سوال به صورت زیر است:

```
CREATE VIEW V1
AS SELECT SID AS ID, SUM(QTY) AS SQTY
    FROM SP JOIN S
    WHERE LEVEL>10
    GROUP BY SID
```

مطابق تعریف فوق، محتویات جدول مجازی $V1$ کاملاً متکی به جداول S و SP است. در واقع می‌توان گفت در هر زمان محتویات $V1$ عبارتند از نتیجه‌ی اجرای پرس و جوی زیر:

```
SELECT SID AS ID, SUM(QTY) AS SQTY
FROM SP JOIN S
WHERE LEVEL>10
GROUP BY SID
```

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

```
SELECT S.SID AS ID, SUM(QTY) AS SQTY
FROM SP JOIN S
on S.SID=SP.SID
WHERE LEVEL>10
GROUP BY S.SID
```

جداول زیر را در نظر بگیرید:

<u>SID</u>	SName	Level	City	<u>SID</u>	PID	QTY	<u>PID</u>	...
s1	sn1	10	c1	s1	p1	500	p1	
s2	sn2	20	c2	s1	p2	600	p2	
s3	sn3	30	c3	s2	p1	700		
				s2	p2	800		
				s3	p1	300		
				s3	p2	400		

تولید کنندگان S

تولید SP

قطعات P

با توجه به جداول فوق، خروجی پرس و جو پس از انجام عملگر $JOIN$ به صورت زیر است:

SID	SName	Level	City	SID	PID	QTY
s1	sn1	10	c1	s1	p1	500
s1	sn1	10	c1	s1	p2	600
s2	sn2	20	c2	s2	p1	700
s2	sn2	20	c2	s2	p2	800
s3	sn3	30	c3	s3	p1	300
s3	sn3	30	c3	s3	p2	400

در ادامه پس از انجام دستور $WHERE LEVEL>10$ خروجی پرس و جو به صورت زیر است:

SID	SName	Level	City	SID	PID	QTY
s2	sn2	20	c2	s2	p1	700
s2	sn2	20	c2	s2	p2	800
s3	sn3	30	c3	s3	p1	300
s3	sn3	30	c3	s3	p2	400

همچنین در ادامه، پس از انجام دستور $GROUP BY S.SID$ براساس $S.SID$ خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:

S2	S3
P1 700 ...	P1 300 ...
P2 800 ...	P2 400 ...
گروه اول	گروه دوم
SUM (QTY)=1500	SUM (QTY)=700

و در نهایت دستور SUM(QTY) داخل دستور SELECT برای هر گروه به طور مستقل محاسبه می‌گردد و در خروجی پرس و جو قرار می‌گیرد، بنابراین خروجی نهایی جدول مجازی V1 به صورت زیر است:

SID	SQTY
s2	1500
s3	700

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

در صورت سوال مطرح شده است که دستور زیر در سطح خارجی با نگاشت E/C تبدیل به چه دستوری در سطح ادراکی می‌شود؟

```
SELECT ID, SQTY
FROM V1
WHERE SQTY>1000
```

خروجی پرس و جوی فوق پس از انجام دستور WHERE SQTY>1000 بر روی جدول مجازی V1 به صورت زیر است:

SID	SQTY
s2	1500

هنگامی که در یک پرس و جو از جدول مجازی استفاده می‌شود، ابتدا تعریف جدول مجازی مربوطه، در آن پرس و جو جایگذاری می‌شود و سپس پرس و جو اجرا می‌گردد، در واقع می‌توان گفت پرس و جوی فوق، عبارتند از نتیجه‌ی اجرای پرس و جوی زیر:

```
SELECT S.SID AS ID, SUM(QTY) AS SQTY
FROM SP JOIN S
on S.SID=SP.SID
WHERE LEVEL>10
GROUP BY S.SID
HAVING SUM(QTY)> 1000
```

دقت کنید که دستور WHERE SQTY>1000 در پرس و جوی زیر بر روی سطرهای جدول مجازی V1 انجام می‌شود، به عبارت دقیق‌تر دستور WHERE SQTY>1000 در پرس و جوی زیر بر روی حاصل جمع تعداد قطعات در گروه‌های حاصل از دستور GROUP BY S.SID موجود در جدول مجازی V1 انجام می‌شود.

توجه: پرس و جوی زیر از نظر نحوی صحیح است، زیرا خروجی جدول مجازی V1 تعدادی سطر است و دستور WHERE SQTY>1000 می‌تواند بر روی سطرهای اعمال گردد.

```
SELECT ID, SQTY
FROM V1
WHERE SQTY>1000
```

و خروجی زیر نیز حاصل می شود:

SID	SQTY
s2	1500

اما هنگامی که قرار است، ساختار کلی پرس و جو بدون جدول مجازی V1 مشاهده شود، از دستور WHERE SQTY > 1000 HAVING SUM(QTY) > 1000 به جای برای انتخاب گروه‌ها استفاده می‌شود.

```
SELECT S.SID AS ID, SUM (QTY) AS SQTY  
FROM SP JOIN S  
on S.SID=SP.SID  
WHERE LEVEL>10  
GROUP BY S.SID  
HAVING SUM (QTY) > 1000
```

همانطور که گفتیم، پس از انجام دستور GROUP BY S.SID براساس S.SID خروجی پرس و جو به صورت زیر گروهبندی می شود.

S2	S3
P1 700 ...	P1 300 ...
<u>P2 800 ...</u>	<u>P2 400 ...</u>

گروه اول گروه دوم

و در نهایت دستور HAVING SUM(QTY)> 1000 برای هر گروه به طور مستقل اعمال می‌گردد.
توجه: دستور HAVING بر روی گروه‌ها، اعمال می‌گردد.

S2	S3
P1 700 ...	P1 300 ...
<u>P2 800 ...</u>	<u>P2 400 ...</u>
گروه اول	گروه دوم
SUM (QTY) = 1500	SUM (QTY) = 700

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

در واقع پرس و جوی فوق، شماره تولیدکنندگانی را در خروجی قرار می‌دهد، که هم سطح آنها بالای 10 باشد و هم مجموع شماره قطعات تولیدی آنها بالای 1000 باشد، خروجی نهایی پرس و جو به صورت زیر است:

SID	SQTY
s2	1500

۲۴ - گزینه (۱) صحیح است.

جداول زیر را در نظر بگیرید:

<u>SID</u>	SName	Level	..	<u>SID</u>	<u>PID</u>	QTY	<u>PID</u>	...
s1	sn1	6		s1	p1	10	p1	
s2	sn2	2		s1	p2	20	p2	
s3	sn3	3		s2	p2	30		

مطرح شده در صورت سوال به صورت زیر است:

```
CREATE ASSERTION MYCONST
CHECK (NOT EXISTS (SELECT *
                   FROM S
                  WHERE S.LEVEL<5 AND EXISTS (SELECT *
                                                FROM SP
                                               WHERE SP.SID = S.SID
                                                 AND SP.PID = 'p1')))
```

الگوهای چهارگانه (همه، هیچ، بعضی و حداقل) زیر را در نظر بگیرید:
الگوی اول (00):

```
SELECT SID
      FROM S
     WHERE NOT EXISTS (SELECT *
                           FROM P
                          WHERE NOT EXISTS (SELECT *
                                            FROM SP
                                           WHERE S.SID = SP.SID
                                             AND P.PID = SP.PID))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«شماره تولیدکنندگانی که، وجود نداشته باشد (NOT EXISTS) قطعه‌ای که، تولید نشده باشد (NOT EXISTS). یعنی شماره تولیدکنندگانی که همه قطعات را تولید کرده‌اند.»
با توجه به جداول فوق خروجی پرس و جوی الگوی اول به صورت زیر است:

SID	
s1	

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول P بررسی می‌گردد، تا مشخص گردد آیا هیچ سطری در جدول P وجود ندارد که شرط اتصال S.SID = SP.SID AND P.PID = SP.PID موجود در select باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

الگوی دوم (01):

```
SELECT SID
      FROM S
     WHERE NOT EXISTS (SELECT *
                           FROM P
                          WHERE EXISTS (SELECT *
                                         FROM SP
                                           WHERE S.SID = SP.SID
                                             AND P.PID = SP.PID))
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«شماره تولیدکنندگانی که، وجود نداشته باشد (NOT EXISTS) قطعه‌ای که، تولید شده باشد (EXISTS). یعنی شماره تولیدکنندگانی که هیچ قطعه‌ای را تولید نکرده‌اند.»

با توجه به جداول فوق خروجی پرس و جوی **الگوی دوم** به صورت زیر است:

SID

s3

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول P بررسی می‌گردد، تا مشخص گردد آیا هیچ سطری در جدول P وجود ندارد که شرط اتصال $S.SID = SP.SID \text{ AND } P.PID = SP.PID$ موجود در select داخلی را برقرار کند، اگر وجود نداشته باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

الگوی سوم (10):

```
SELECT SID  
FROM S  
WHERE EXISTS (SELECT *  
               FROM P  
               WHERE NOT EXISTS (SE  
                               FR  
                               WI  
                               AI
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:
شماره تولیدکننده کانی که وجود داشته باشد (STS)
(NOT EXISTS). یعنی شماره تولیدکننده کانی که بعضی قطعه
با توجه به جداول فوق خروجی پرس و جوی الگوی سوم با

SID

s2

s3

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول P بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول P وجود دارد که شرط اتصال select S.SID = SP.SID AND P.PID = SP.PID موجود در داخلی را برقرار نکند، اگر وجود داشته باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

الگوی چهارم (11)

```
SELECT SID  
FROM S  
WHERE EXISTS (SELECT *  
               FROM P  
               WHERE E
```

پرس و جوی فوق را همانند سورها به صورت زیر بخوانید:

«شماره تولیدکنندگانی که، وجود داشته باشد (EXISTS) قطعه‌ای که، تولید شده باشد». یعنی

شماره تولیدکنندگانی که حداقل یک قطعه را تولیدکرده‌اند».

با توجه به جداول فوق خروجی پرس و جوی الگوی چهارم به صورت زیر است:

SID

s1

s2

در پرس و جوی فوق به ازای حرکت در هر سطر از جدول S، یک بار به طور کامل از ابتدا تا انتهای جدول P بررسی می‌گردد، تا مشخص گردد آیا حداقل یک سطر در جدول P وجود دارد که شرط اتصال موجود در $S.SID = SP.SID \text{ AND } P.PID = SP.PID$ select داخلي را برقرار کند، اگر وجود داشته باشد، سطر مورد نظر از جدول S در خروجی نمایش داده می‌شود.

ASSERTION مطرح شده در صورت سوال را مجدداً در نظر بگیرید:

```
CREATE ASSERTION MYCONST
CHECK (NOT EXISTS (SELECT *
                     FROM S
                  WHERE S.LEVEL<5 AND EXISTS (SELECT *
                                              FROM SP
                                              WHERE SP.SID = S.SID
                                                AND SP.PID = 'p1')))
```

توجه: ASSERTION مطرح شده در صورت سوال، مطابق الگوی دوم می‌باشد.

توجه: دقیقاً دقت کنید که دلیل این ASSERTION این است که خواننده می‌شوند عمل می‌کنند، و دقیقاً به همان شکل که خواننده می‌شوند اجازه ذخیره‌سازی داده‌ها را به جداول می‌دهند.

ASSERTION فرق را همانند سورها به صورت زیر بخوانید:

«تولیدکنندگانی با سطح کمتر از 5 وجود نداشته باشد (NOT EXISTS) که، قطعه‌ای با شناسه p1 را تولید کرده باشد (EXISTS). یعنی تولیدکنندگانی با سطح کمتر از 5 که هیچ قطعه‌ای با شناسه p1 را تولید نکرده‌اند». برای مثال تولیدکننده s2 با سطح 2 و تولیدکننده s3 با سطح 3 حق تولید قطعه p1 را ندارند.

۲۵- گزینه (۲) صحیح است.

جدول زیر را در نظر بگیرید:

STID	Name	Major	Department
s1	sn1	m1	d1
s2	sn2	m1	d1
s3	sn3	m2	d2
s4	sn4	m2	d2
s5	sn5	m3	d3
s6	sn6	m3	d3

Student جدول

ASSERTION مطرح شده در صورت سوال به صورت زیر است:

```

CREATE ASSERTION MYASSERT
CHECK (NOT EXISTS (SELECT DEPARTMENT
                     FROM STUDENT
                     GROUP BY DEPARTMENT
                     HAVING COUNT (DISTINCT (MAJOR))>1))

```

توجه: دقت کنید که ASSERTION ها دقیقا همانطور که خوانده می‌شوند عمل می‌کنند، و دقیقا به همان شکل که خوانده می‌شوند اجازه ذخیره‌سازی داده‌ها را به جداول می‌دهند.

در هر درج و بروزرسانی ASSERTION فوق بر روی ورود داده‌ها در جدول STUDENT نظارت می‌کند تا جامعیت داخلی تعریف شده توسط طراح بانک اطلاعات برقرار گردد. به این نحو که پس از ورود داده‌ها ابتدا پرس و جوی زیر اجرا می‌شود:

```

SELECT DEPARTMENT
FROM STUDENT
GROUP BY DEPARTMENT
HAVING COUNT (DISTINCT (MAJOR))>1

```

پس از انجام دستور GROUP BY DEPARTMENT براساس خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:

d1	d2	d3
s1 sn1 m1	s3 sn3 m2	s5 sn5 m3
<u>s2 sn2 m1</u>	<u>s4 sn4 m2</u>	<u>s6 sn6 m3</u>
گروه اول	گروه دوم	گروه سوم

در نهایت دستور HAVING COUNT (DISTINCT (MAJOR))>1 برای هر گروه به طور مستقل اعمال می‌گردد.

توجه: دستور HAVING بر روی گروه‌ها، اعمال می‌گردد.

d1	d2	d3
s1 sn1 m1	s3 sn3 m2	s5 sn5 m3
<u>s2 sn2 m1</u>	<u>s4 sn4 m2</u>	<u>s6 sn6 m3</u>
گروه اول	گروه دوم	گروه سوم

COUNT(DISTINCT(MAJOR)) = 1 COUNT(DISTINCT(MAJOR)) = 1 COUNT(DISTINCT(MAJOR)) = 1

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

توجه: دستور Distinct، سطرهای تکراری در ستون MAJOR را حذف می‌کند.

در واقع پرس و جوی فوق، شماره دانشکده‌هایی (DEPARTMENT) را در خروجی قرار می‌دهد که بیشتر از یک رشته (MAJOR) ارائه می‌کنند، که مطابق آنچه واضح است در حال حاضر هیچ یک از دانشکده‌ها بیشتر از یک رشته ارائه نمی‌کردند. بنابراین خروجی نهایی پرس و جو یک جدول تهی به صورت زیر است:

DEPARTMENT

حال اگر در ادامه دستور NOT EXISTS انجام گردد، از آنجا که جلوی NOT تهی است، در

نتیجه مقدار NOT EXISTS برابر TRUE می‌شود و این بدین معنی است که ASSERTION در حالت امن قرار دارد، یعنی جامعیت داخلی تعریف شده توسط طراح بانک اطلاعات برقرار است. حال اگر دستور درج رکورد ('s7', 'sn7', 'm4', 'd3') از سوی کاربر بر روی جدول STUDENT به صورت زیر انجام شود:

```
INSERT INTO STUDENT (STID,NAME,MAJOR,DEPARTMENT)
VALUES ('s7','sn7','m4','d3')
```

آنگاه ASSERTION تعریف شده توسط طراح بانک اطلاعات جلوی درج رکورد ('s7', 'sn7', 'm4', 'd3') بر روی جدول STUDENT را می‌گیرد، زیرا در این شرایط دانشکده d3 دو رشته m3 و m4 را دارد که این موضوع خلاف جامعیت داخلی تعریف شده توسط طراح بانک اطلاعات است و بانک اطلاعات را نامن کرده است، بنابراین اجرای دستور درج مذکور، از سوی DBMS پذیرفته نمی‌شود و رد می‌شود. جدول زیر را پس از دستور درج مذکور در نظر بگیرید:

STID	Name	Major	Department
s1	sn1	m1	d1
s2	sn2	m1	d1
s3	sn3	m2	d2
s4	sn4	m2	d2
s5	sn5	m3	d3
s6	sn6	m3	d3
s7	sn7	m4	d3

Student جدول

مطرح شده در صورت سوال به صورت زیر است:

```
CREATE ASSERTION MYASSERT
CHECK (NOT EXISTS (SELECT DEPARTMENT
                     FROM STUDENT
                     GROUP BY DEPARTMENT
                     HAVING COUNT (DISTINCT (MAJOR))>1))
```

در هر درج و بروزرسانی ASSERTION فوق بر روی ورود داده‌ها در جدول STUDENT نظارت می‌کند تا جامعیت داخلی تعریف شده توسط طراح بانک اطلاعات برقرار گردد. به این نحو که پس از ورود داده‌ها ابتدا پرس و جوی زیر اجرا می‌شود:

```
SELECT DEPARTMENT
FROM STUDENT
GROUP BY DEPARTMENT
HAVING COUNT (DISTINCT (MAJOR))>1
```

پس از انجام دستور GROUP BY DEPARTMENT براساس DEPARTMENT خروجی پرس و جو به صورت زیر گروه‌بندی می‌شود:

d1	d2	d3
s1 sn1 m1	s3 sn3 m2	s5 sn5 m3
s2 sn2 m1	s4 sn4 m2	s6 sn6 m3
		<u>s7 sn7 m4</u>
گروه اول	گروه دوم	گروه سوم

در نهایت دستور HAVING COUNT (DISTINCT (MAJOR))>1 برای هر گروه به طور مستقل اعمال می‌گردد.
توجه: دستور HAVING بر روی گروه‌ها، اعمال می‌گردد.

d1	d2	d3
s1 sn1 m1	s3 sn3 m2	s5 sn5 m3
s2 sn2 m1	s4 sn4 m2	s6 sn6 m3
		<u>s7 sn7 m4</u>
گروه اول	گروه دوم	گروه سوم

COUNT(DISTINCT(MAJOR)) = 1 COUNT(DISTINCT(MAJOR)) = 1 COUNT(DISTINCT(MAJOR)) = 2

توجه: دستور GROUP BY، سرگروه‌ها را، راهی خروجی می‌کند.
توجه: دستور Distinct، سطرهای تکراری در ستون MAJOR را حذف می‌کند.

در واقع پرس و جوی فوق، شماره دانشکده‌هایی (DEPARTMENT) را در خروجی قرار می‌دهد که بیشتر از یک رشته (MAJOR) ارائه می‌کنند، که مطابق آنچه واضح است در حال حاضر دانشکده‌ی d3 بیشتر از یک رشته ارائه کرده است. بنابراین خروجی نهایی پرس و جو به صورت زیر است:

DEPARTMENT
d3

حال اگر در ادامه دستور NOT EXISTS انجام گردد، از آنجا که جلوی NOT EXISTS غیرتهی است، در نتیجه مقدار NOT EXISTS برابر FALSE می‌شود و این بدین معنی است که ASSERTION در حالت نامن قرار دارد، یعنی جامعیت داخلی تعریف شده توسط طراح بانک اطلاعات برقرار نیست. بنابراین اجرای دستور درج مذکور، از سوی DBMS پذیرفته نمی‌شود و رد می‌شود. و بانک اطلاعات به حالت امن باز می‌گردد. به همین سادگی ...

- ۲۶- گزینه (۳) صحیح است.
جدول زیر را در نظر بگیرید.

کلید خارجی		
PersonID	ManagerID	Level
<i>Person</i>		

کلید خارجی برای ارتباط میان جداول یا ارتباط درون یک جدول مورد استفاده قرار می‌گیرد. به ازای هر مقدار موجود در یک کلید خارجی، باید دقیقاً یک مقدار متناظر در کلید کاندید متناظر آن وجود داشته باشد، در غیر این صورت می‌گوییم، کلید خارجی دارای ارجاع NULL است. به بیان دیگر، مقادیر کلید خارجی همواره باید زیرمجموعه مقادیر کلید کاندید باشد.

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

دستور زیر بر روی جدول Person تعریف شده است:

Foreign key (managerID) References person (personID)

on delete CASCADE

یعنی کلید خارجی جدول Person یعنی ستون ManagerID به تغییرات (حذف) کلید کاندید جدول Person یعنی ستون ID به فرم cascade حساس باشد و واکنش نشان دهد، وقت 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 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، وقت 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 جدول Person به فرم restrict 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 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.

Person		
PersonID	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

Card(Person) = 8

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

Person		
PersonID	ManagerID	Level
14	NULL	1

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 انتهای می‌ماند به صورت زیر است:

Person		
PersonID	ManagerID	Level
14	NULL	1

- ۲۷- گزینه (۱) صحیح است.

صورت سوال به این شکل است:

فرض کنید اسکیماتی جدول‌های E1، E2 و E3 مطابق با دستورات زیر بوده و تعداد سطرهای هر کدام به

ترتیب، M و K باشد. پس از اجرای دستور زیر، تعداد سطرهای جدولهای E1، E2 و E3 به ترتیب N، N' و K' است. کدام مورد نمی‌تواند رخ دهد؟

Delete from E1 where id1=5

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

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

کلید خارجی	کلید کاندید	کلید خارجی	کلید کاندید
			↓
	a id2	b id3	
	5 8	8 2	
	3 6		
5	3	8	2
3	6		
1			
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 مسئله 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 است، یعنی اگر رکوردهای 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 رکوردهای در جدول E2 حذف شود که منجر به حذف در جدول E3 شود، آن حذف در جدول E2 در جهت حفظ قانون جامعیت ارجاعی پذیرفته نمی‌شود و رد می‌شود.

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

Delete from E1 where id1 = 5

کلید کاندید		کلید خارجی		کلید خارجی	
<u>id1</u>		a	<u>id2</u>	b	<u>id3</u>
5		5	8	8	2
3		3	6		
1					
E1		E2		E3	

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

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

کلید کاندید		کلید خارجی		کلید کاندید	
<u>id1</u>		a	<u>id2</u>	b	<u>id3</u>
5		5	8	8	2
3		3	6		
1					
E1		E2		E3	

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

پس از اجرای دستور Delete هیچ رکوردهای 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 جداول 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.

$$M' = M, N' = N, K' = K \quad (2)$$

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

کلید کاندید	کلید خارجی	کلید خارجی
\downarrow	\downarrow	\downarrow
<u>id1</u>	a id2	b id3
1 3 7	7 3 8 6	8 2

E1 E2 E3

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

Delete from E1 where id1=5

کلید کاندید	کلید خارجی	کلید خارجی
\downarrow	\downarrow	\downarrow
<u>id1</u>	a id2	b id3
1 3 7	7 3 8 6	8 2

E1 E2 E3

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

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

کلید کاندید	کلید خارجی	کلید خارجی
\downarrow	\downarrow	\downarrow
<u>id1</u>	a id2	b id3
1 3 7	7 3 8 6	8 2

E1 E2 E3

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

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

$$M' < M, N' < N, K' = K \quad (3)$$

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

کلید کاندید	کلید خارجی	کلید کاندید	کلید خارجی
\downarrow	\downarrow	\downarrow	\downarrow
<u>id1</u>	a id2	b id3	
1 5 7	7 5 8 6	8 2	

E1 E2 E3

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

Delete from E1 where id1=5

کلید کاندید	کلید خارجی	کلید خارجی
\downarrow	\downarrow	\downarrow
<u>id1</u>	a <u>id2</u>	b <u>id3</u>
$\frac{1}{5}$	$\frac{7}{5}$	$\frac{8}{2}$

*E1**E2**E3*

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

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

کلید کاندید	کلید خارجی	کلید کاندید	کلید خارجی	کلید خارجی
\downarrow	\downarrow	\downarrow	\downarrow	\downarrow
<u>id1</u>	a <u>id2</u>	$\frac{1}{7}$	$\frac{7}{8}$	$\frac{8}{2}$ <u>id3</u>
$\frac{1}{7}$				

*E1**E2**E3*

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

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

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

$$M' < M, N' = N, K' = K \quad (4)$$

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

کلید کاندید	کلید کاندید	کلید خارجی	کلید کاندید	کلید خارجی
\downarrow	\downarrow	\downarrow	\downarrow	\downarrow
<u>id1</u>	a <u>id2</u>	$\frac{1}{5}$	$\frac{7}{8}$	$\frac{8}{2}$ <u>id3</u>
$\frac{5}{3}$	$\frac{3}{6}$			
$\frac{7}{}$				

*E1**E2**E3*

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

Delete from E1 where id1=5

کلید کاندید	کلید کاندید	کلید خارجی	کلید کاندید	کلید خارجی
\downarrow	\downarrow	\downarrow	\downarrow	\downarrow
<u>id1</u>	a <u>id2</u>	$\frac{1}{5}$	$\frac{7}{8}$	$\frac{8}{2}$ <u>d3</u>
$\frac{5}{3}$	$\frac{3}{6}$			
$\frac{7}{}$				

*E1**E2**E3*

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

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

		کلید خارجی			
		کلید کاندید	کلید خارجی		
↓	↓				
<u>id1</u>		a	id2	b	id3
3	7	7	8	8	2
7		3	6		
E1		E2		E3	

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

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

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

- ۲۸- گزینه (۲) صحیح است.

صورت سوال به این شکل است:

چه تعداد از گزاره‌های داده شده درست است؟

(الف) تعداد کلیدهای کاندید یک رابطه از تعداد ابرکلیدهای آن رابطه همواره کمتر یا مساوی است.
 گزاره اول درست است، زیرا در حالت کلی، اگر رابطه R، دارای n خصیصه باشد، آنگاه تعداد ابرکلیدهای آن حداقل یک و حداکثر $1 - 2^n$ 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 $1 - 2^n$ ابرکلید، حالت خاص جدول تمام کلید هم پوشش می‌دهد چون بیان حداکثر $1 - 2^n$ مقادیر کوچکتر و برابر خودش را پوشش می‌دهد. اگر در یک رابطه با n خصیصه، تک تک خصیصه‌ها به تهایی 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 $1 - 2^n$ است که بیشترین مقدار ممکن در تعداد ابرکلیدهای یک رابطه با n خصیصه است. ابرکلید بدون صفت نداریم، بنابراین حالت $\binom{n}{0}$ اضافه است. همانطور که گفتیم در حالت کلی، یک رابطه دارای n خصیصه، شرایط مختلفی را در تعداد ابرکلید می‌تواند تجربه کند، که حداکثر تعداد ابرکلیدهایی که می‌تواند تجربه کند برابر $\binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n}$ یا $2^n - 1$ است. در واقع عبارت زیر برقرار است:

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

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

مثال:

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

S جدول *SP* جدول *P* جدول

(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 تهایی 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 تهایی خاصیت کلیدی دارد، بنابراین صفت نام خانوادگی، عضو زائد است.

توجه: یک جدول می‌تواند چندین کلید کاندید داشته باشد.

مثال:

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

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

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

مثال: یک جدول تمام کلید.

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

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

ب) اگر رابطه R دارای n خصیصه باشد، آنگاه تعداد ابرکلیدهای این رابطه حداقل 2^n است. گزاره دوم نادرست است، زیرا در حالت کلی، اگر رابطه R ، دارای n خصیصه باشد، آنگاه تعداد ابرکلیدهای آن حداقل یک و حداقل $1 - 2^n$ است.

ج) دستور ALTER Table در SQL، کاتالوگ سیستم را به روز می‌کند.

گزاره سوم درست است، زیرا در یک سیستم بانک اطلاعات، اسمی زیادی مورد استفاده قرار می‌گیرند، از آنجایی که افراد زیاد و متفاوتی در یک مجموعه بانک اطلاعات در گیر هستند، مرجعی برای ایجاد یکنواختی و هماهنگی در نام داده‌ها و معنای آنها ضروری است. این مرجع، دیکشنری داده‌ها نام دارد. در اختصار کاتالوگ سیستم یا دیکشنری داده‌ها شامل تمامی اطلاعات سیستمی مربوط به پایگاه داده‌ها همچون مشخصات سیستمی جداول و سطوح دسترسی کاربران می‌باشد که به طور خودکار توسط DBMS بروزرسانی می‌گردد. به بیان کامل تر هرگونه تغییرات حاصل از دستورات DDL در پایگاه داده همچون ایجاد جداول، تغییر جداول، حذف جداول، ایجاد شاخص، حذف شاخص، ایجاد View، حذف View و یا هرگونه تغییرات حاصل از دستورات DML در پایگاه داده همچون درج، حذف و بروزرسانی رکوردها که منجر به تغییر تعداد رکوردها و به تبع آن تغییر اندازه جداول پایگاه داده‌ها می‌شود و یا هر گونه تغییرات حاصل از ایجاد و تغییر سطوح دسترسی توسط دستورات DCL در پایگاه داده همچون تخصیص سطوح دسترسی به کاربران و هر آنچه که مربوط به مشخصات سیستمی پایگاه داده باشد در کاتالوگ سیستم نگهداری می‌شود. در واقع شناسنامه بانک اطلاعات، دیکشنری داده‌ها یا کاتالوگ سیستم است.

دیکشنری داده در جایگاه خود، پایگاه داده‌ای سیستمی شامل اطلاعاتی سیستمی در مورد پایگاه داده یک محیط عملیاتی می‌باشد که حاوی «داده‌هایی درباره داده» است که گاهی اوقات به نام «فراداده» یا «دادگان» به معنی داده در مورد داده معرفی می‌گردد.

کاتالوگ سیستم به واسطه اجرای تمامی دستورات DDL و برخی دستورات DML مانند Insert و Delete و نه Select و Update دستخوش تغییر می‌گردد. دستور ALTER Table یکی از دستورات DDL برای اعمال تغییرات روی جداول پایگاه داده است، دستورات DDL در SQL دستورات ظرف ساز و ساختارساز در پایگاه داده هستند. از آنجا که تمامی دستورات DDL منجر به تغییرات در پایگاه داده می‌شود، بنابراین کاتالوگ سیستم به واسطه اجرای تمامی دستورات DDL دستخوش تغییر می‌گردد، نتیجه اینکه دستور ALTER Table در SQL، کاتالوگ سیستم را به روز می‌کند.

د) استفاده از View و Index می‌تواند استقلال داده‌ای را افزایش دهد.

گزاره چهارم نادرست است، زیرا معماری ANSI برای پایگاه داده‌ها شامل سه لایه زیر است:

۱- لایه خارجی.

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

۳- لایه داخلی (فیزیکی).

یک محصول نرم‌افزاری به واسطه فرآیند تولید نرم‌افزار که شامل فعالیت‌های مدل تحلیل، مدل طراحی،

پیاده‌سازی و تست می‌باشد، ایجاد می‌گردد. کاربران نهایی در لایه خارجی، مدل تحلیل و مدل طراحی در لایه ادراکی و فعالیت پیاده‌سازی در لایه داخلی قرار دارند.

زبان‌های پیاده‌سازی

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

زبان پیاده‌سازی برنامه کاربردی (وجه عملکرد)

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

زبان پیاده‌سازی بانک اطلاعات (وجه داده)

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

استقلال داده‌ای

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

۱- استقلال فیزیکی داده‌ها

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

۲- استقلال منطقی داده‌ها

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

برای مثال مدل رابطه‌ای از تجربیدی به نام جدول استفاده می‌کند و داده‌ها هر چه باشند در قالب چند جدول ریخته می‌شوند و نحوه ذخیره‌سازی داده‌ها روی رسانه‌ها از دید برنامه کاربردی مخفی است. در حالی که در روش فایلینگ تعاریف مربوط به فایل‌های داده‌ای، در فایل برنامه کاربردی می‌آمد. 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 صد در صد) تأمین شده است.

أنواع تغيير در مدل طراحى (طراحى منطقى يا ادراكي خاص)

۱- رشد پایگاه داده ها به دليل مطرح شدن نيازهای جديد مشترى: مانند درج جدول جديد، ترکيب جداول، تجزيه جداول.

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

مثال: اگر جدولی داراي چهار ستون باشد و ستون پنجمي نيز به آن اضافه گردد، در صورتى كه برنامه کاربردي سابق نياز به دستكارى و تغيير نداشته باشد، استقلال منطقى دادهها براساس تغييرات نيز لحظه شده است.

توجه: از آن جا كه با حذف جداول، دادهها هم از بين می روند، بنابراین برنامههای کاربردی نسبت به حذف جداول **هيچگاه استقلال منطقى** نخواهند داشت.

همانطور كه گفتيم يك محصول نرمافزاری از دو وجهه **عملکرد** (برنامه کاربردی) و **داده** (بانک اطلاعات) تشکيل شده است، بخش داده (بانک اطلاعات) كه با SQL پياده‌سازی می‌شود به مفاهيم استقلال دادهای مرتبط است. ساختار وجهه داده توسط دستورات DDL نظير Create Table، Create View و Create Index و دیگر دستورات آن ايجاد و مدیريت می‌گردد. و مقادير وجهه داده توسط دستورات Insert DML نظير Insert و Update و دیگر دستورات آن ايجاد و مدیريت می‌گردد.

دستور Create Table با ساخت مفهوم جدول، كمك به برقراری استقلال دادهای از نوع **استقلال فизيکي** دادهها ميان يك برنامه کاربردی و دادهها می‌كند، به معنى وابسته نبودن برنامههای کاربردی به دادههای ذخیره شده، يعني همانطور كه گفتيم، مدل رابطهای از تجربیدي به نام جدول استفاده می‌كند و دادهها هر چه باشند در قالب چند جدول ريخته می‌شوند و نحوه ذخیره‌سازی دادهها روی رسانه‌ها از ديد برنامه کاربردی مخفی است. وقت كنيد كه Table بخشی از وجهه داده است. در واقع بخش داده از بخش‌های مختلف View و Index تشکيل شده است.

دستور Create View با ساخت مفهوم ديد، تا حدی كمك به برقراری استقلال دادهای از نوع **استقلال منطقى** دادهها ميان يك برنامه کاربردی و دادهها می‌كند، به معنى وابسته نبودن برنامههای کاربردی به دادههای ذخیره شده، يعني همانطور كه گفتيم، اگر جدولی داراي چهار ستون باشد و ستون پنجمي نيز به آن اضافه گردد، در صورتى كه برنامه کاربردی سابق نياز به دستكارى و تغيير نداشته باشد، استقلال منطقى دادهها براساس تغييرات نيز لحظه شده است. از آنجاكه View روی ساختار قدیم شامل نام جدول قدیم و ستون‌های قدیم می‌شود، اگر جدولی داراي چهار ستون باشد و ستون پنجمي نيز به آن اضافه گردد، آنگاه بدون تغييرات در ساختار View بخش داده و به تبع تغييرات در ساختار بخش عملکرد (برنامه کاربردی)، امكان حيات برنامه کاربردی بدون اشكال همچنان وجود دارد و اين يعني View حافظ استقلال دادهای از نوع استقلال منطقى دادهها است. وقت كنيد كه View بخشی از وجهه داده است. در واقع بخش داده از بخش‌های مختلف Table، View و Index تشکيل شده است.

دستور Create Index با ساخت مفهوم شاخص، هيچ كمكی به برقراری استقلال دادهای از نوع **استقلال فيزيکي** دادهها و استقلال منطقى دادهها ميان يك برنامه کاربردی و دادهها نمي‌كند، به معنى وابسته نبودن برنامههای کاربردی به دادههای ذخیره شده. مهمترین کاربرد شاخص، افرايش سرعت جستجو و بازيابي اطلاعات است و اين مفهوم و کاربرد هيچ ارتباطی به مفهوم استقلال دادهای ندارد. شاخص فقط باعث بالا رفتن سرعت دستيابي به اطلاعات می‌گردد. شاخص‌ها برای بهبود فرآيند جستجو و بازيابي اطلاعات در جداول ايجاد می‌شوند. به عبارت دیگر شاخص‌ها به فرآيند جستجو و بازيابي

اطلاعات، سرعت می‌بخشند. شاخص‌ها باعث می‌شوند موتور جستجوی پایگاه داده کل یک جدول را برای پیداکردن رکورد یا رکوردهای مورد نظر به طور کامل نگردد. اساساً شاخص‌ها بر روی ستون‌هایی باید تنظیم شود که بیشتر مورد جستجو قرار می‌گیرند.

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

توجه: تعریف و نگهداری شاخص موجب تحمیل سربار حافظه‌ای به سیستم می‌شود. شاخص بر روی هارد دیسک نگهداری می‌شود و هنگام استفاده به حافظه اصلی آورده می‌شود، بنابراین اعمال سیاست شاخص‌گذاری، بر افزایش حجم اطلاعات ذخیره شده بر روی حافظه اصلی و هارد دیسک تاثیر دارد.

توجه: عملیات درج، حذف و بروزرسانی در جدولی که شاخص دارد یعنی خود جدول پایه، نسبت به جدولی که شاخص ندارد زمان بیشتری مصرف می‌کند و به تبع این عملیات کنترل خواهد بود. زیرا شاخص‌ها نیز همگام با جداول پایه خود نیاز به بروزرسانی دارند. بنابراین تنها روی ستون‌هایی شاخص ایجاد می‌گردد، که به تناوب روی آنها جستجو انجام می‌شود. بنابراین هرچه تعداد شاخص‌های یک جدول بیشتر باشد، سرعت Insert، Update و Delete در آن جدول کمتر می‌شود. اما خود شاخص عامل جستجو و بازیابی سریع اطلاعات از یک جدول پایه است.

توجه: شاخص‌گذاری، افزونگی تکنیکی دارد و مقداری از فضای حافظه اصلی و هارد دیسک را اشغال می‌کند. به عبارت دیگر تعریف هر شاخص روی یک جدول هزینه زیادی را به پایگاه داده تحمیل می‌کند و اگر نرخ تغییرات محتوایی پایگاه داده زیاد باشد این هزینه به صورت قابل توجهی افزایش می‌یابد. بنابراین طراحان پایگاه داده ترجیح می‌دهند که روی هر جدول بیش از یک شاخص تعریف نکنند.

توجه: شاخص‌گذاری، منجر به افزایش سرعت جستجو و بازیابی اطلاعات و به تبع کاهش زمان جستجو و بازیابی اطلاعات می‌شود، نبود شاخص باعث کنندی سرعت جستجو و زیادی آن باعث کنندی فرآیندهای درج، حذف بروزرسانی رکوردها در جداول پایه می‌شود. بنابراین راز نه در افراد است و نه در تفريط بلکه راز در تعادل است، گاهی هم استفاده‌ی مکرر از گزینه‌های خوب، نتیجه‌ی بد هم می‌تواند به همراه داشته باشد! و به قول معروف در حوزه‌ی سلامتی نیاکان ما گفته‌اند که کم بخور، همیشه بخور...

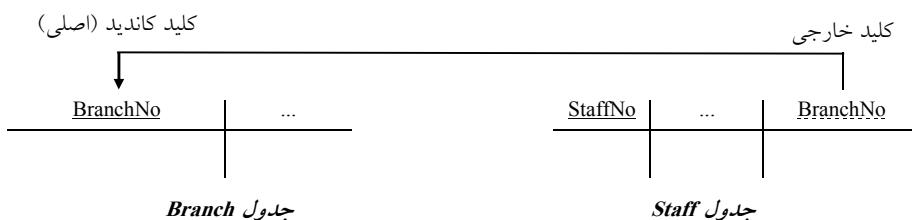
توجه: شاخص (Index) با هدف افزایش سرعت جستجو و بازیابی اطلاعات و به تبع کاهش زمان جستجو و بازیابی اطلاعات ایجاد می‌گردد. و به عنوان نمونه‌ی دیگری از فراداده‌ها، مشخصات سیستمی شاخص در کاتالوگ سیستم نگهداری می‌شود.

دقیق کنید که Index بخشی از وجه داده است. در واقع بخش داده از بخش‌های مختلف Table و View و Index تشکیل شده است.

همانطور که گفتیم استفاده از View می‌تواند استقلال داده‌ای را افزایش دهد. اما استفاده از Index به دلیل نامرتب بودن آن با مفهوم استقلال داده‌ای نمی‌تواند استقلال داده‌ای را افزایش یا کاهش دهد و اثری بر استقلال داده‌ای ندارد.

۲۹- گزینه (۱) صحیح است.

دو جدول Branch و Staff را به صورت زیر در نظر بگیرید:



مطابق دستورات Create Table برای دو جدول Branch و Staff ستون BranchNo توسط دستور Primary key(BranchNo) به عنوان کلید اصلی برای جدول Branch انتخاب می‌گردد. همچنین ستون StaffNo توسط دستور Primary key(StaffNo) به عنوان کلید اصلی برای جدول Staff انتخاب می‌گردد. همچنین ستون BranchNo توسط دستور زیر:

Foreign key (BranchNo) References Branch (BranchNo)

On delete CASCADE

On update CASCADE

به عنوان کلید خارجی برای جدول Staff انتخاب می‌گردد. ستون BranchNo از جدول Staff به عنوان کلید خارجی به ستون BranchNo از جدول Branch مطابق ساختار تعريف جدول Staff به شكل On Update Cascade و On Delete Cascade ارجاع می‌کند.

برای رفتار ستون کلید خارجی در یک جدول مقصد، در قبال تغییرات کلید کاندید از یک جدول مبداء گزینه‌های زیر وجود دارد:

```
Create Table    نام جدول
(
    نام ستونها
    :
    foreign key ... references ...
    on delete option
    on update option
    :
)
```

فیلد option می‌تواند یکی از موارد زیر باشد:

(restrict) no action

گزینه پیش فرض است و هیچ عملی انجام نمی‌شود. یعنی اگر بر اثر عملیات حذف، درج و بروزرسانی، قوانین جامعیت نقض گردند، این اعمال انجام نمی‌گردند. در واقع زمانی عملیات حذف، درج و بروزرسانی انجام می‌گردند که قوانین جامعیت نقض نگردند.

Cascade

اگر سطرهای جداول مرجع حذف یا بروزرسانی شود، کلید خارجی جدولی که به آن ارجاع کرده است نیز حذف یا بروزرسانی خواهد شد.

Set NULL

همان روش NULLIFY است که اگر سطرهای جدول مرجع حذف یا بروزرسانی شود، کلید خارجی

جدولی که به آن ارجاع کرده است با مقدار NULL پر می‌شود.

توجه: کلید خارجی برای ارتباط میان جداول مورد استفاده قرار می‌گیرد.

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

یک کلید خارجی در یک رابطه هیچگاه نباید ارجاع NULL داشته باشد، این مسأله را به عنوان یک قانون جامعیتی داخلی در مدل رابطه‌ای، می‌شناسیم و آن را **قانون جامعیت ارجاعی** می‌نامیم.

هر مقداری که در کلید خارجی وجود دارد، باید دارای مقدار متناظر در کلید کاندید مقصد باشد ولی عکس آن صادق نیست.

دستور زیر بر روی جدول Staff تعریف شده است:

Foreign key (BranchNo) References Branch (BranchNo)

On delete CASCADE

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

همچنین دستور زیر بر روی جدول Staff تعریف شده است:

Foreign key (BranchNo) References Branch (BranchNo)

On update CASCADE

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

اگر دستورات مطرح شده در صورت سوال به ترتیب زیر اجرا شوند، آنگاه کارمندان فعلی شعب 3 و 4 به شعبه 5 منتقل می‌گردند:

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

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 مقصود باشد ولی عکس آن صادق نیست.

توجه: جهت جلوگیری از ایجاد ارجاع NULL از جدول Staff به جدول Branch ابتدا در قدم اول می‌باشد مقدار شعبه 5 در جدول Branch درج گردد. سپس در قدم دوم کلیه سطرهایی که در جدول Staff به شعبه‌های 3 و 4 ارجاع می‌کنند توسط دستور بروزرسانی به شعبه 5 ارجاع کنند. و در نهایت در قدم سوم کلیه سطرهایی که در جدول Branch شامل شعبه‌های 3 و 4 هستند از جدول Branch حذف شوند.

توجه: دقت کنید که اگر در ابتدای دستور Insert انجام شود و سپس دستور Delete انجام شود، آنگاه از آنچهایی که ستون BranchNo از جدول Staff به عنوان کلید خارجی به ستون BranchNo از جدول Branch مطابق ساختار تعريف جدول Staff، به شکل On Delete Cascade ارجاع می‌کند. پس از حذف سطرهای شعبه‌های 3 و 4 از جدول Branch تمامی سطرهای شعبه‌های 3 و 4 به دلیل خاصیت On Delete Cascade از جدول Staff نیز حذف می‌شوند که دیگر امکان بروزرسانی و اجرای دستور Update برای سطرهای شعبه‌های 3 و 4 در جدول Staff وجود ندارد.

فرم جداول و تعداد رکوردهای جداول Staff Branch قبل از انجام دستور Insert به صورت زیر است:
کلید کاندید (اصلی) کلید خارج

<u>BranchNo</u>	...		
3	...		
4	...		
6	...		
8	...		

Branch جدول

<u>StaffNo</u>	...		<u>BranchNo</u>
S1	...		3
S2	...		4
S3	...		6
S4	...		8

Staff جدول

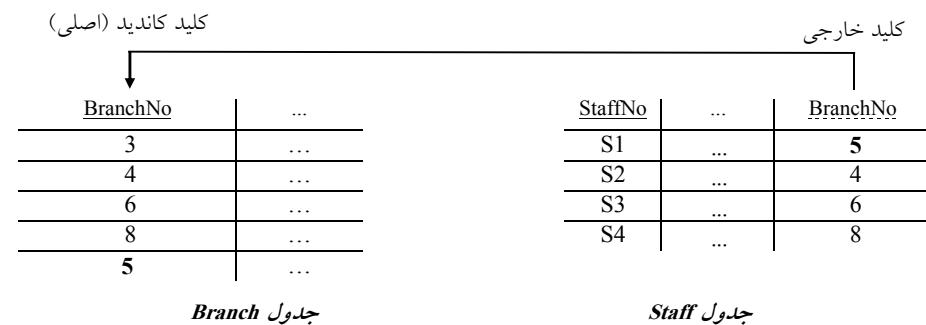
فرم جداول و تعداد رکوردهای جداول Branch و Staff پس از انجام دستور Insert به صورت زیر است:

نام دستور	SQL دستور
a	Insert into Branch values (5, 'تهران', 'شادی', 021222324)
کلید کاندید (اصلی)	کلید خارجی
	
<u>BranchNo</u>	<u>StaffNo</u>
3	S1
4	S2
6	S3
8	S4
5	...

کنکور کارشناسی ارشد (پایگاه داده)

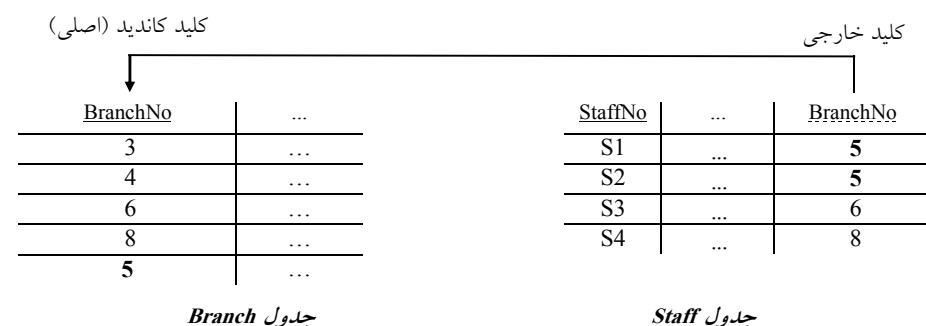
فرم جداول و تعداد رکوردهای جداول Branch و Staff پس از انجام دستور Update به صورت زیر است:

نام دستور	SQL دستور
b	Update Staff set BranchNo=5 where BranchNo=3



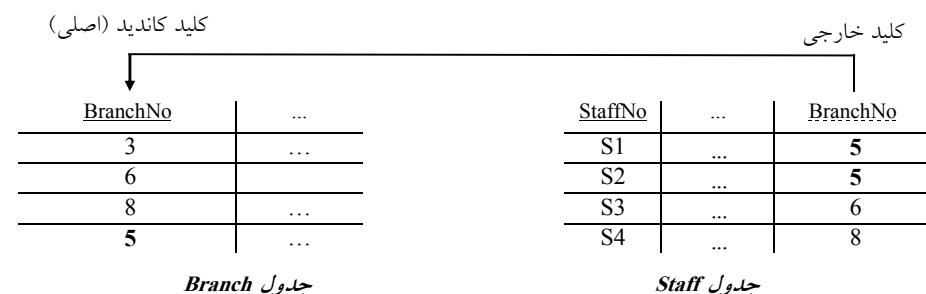
فرم جداول و تعداد رکوردهای جداول Branch و Staff پس از انجام دستور Update به صورت زیر است:

نام دستور	SQL دستور
d	Update Staff set BranchNo=5 where BranchNo=4



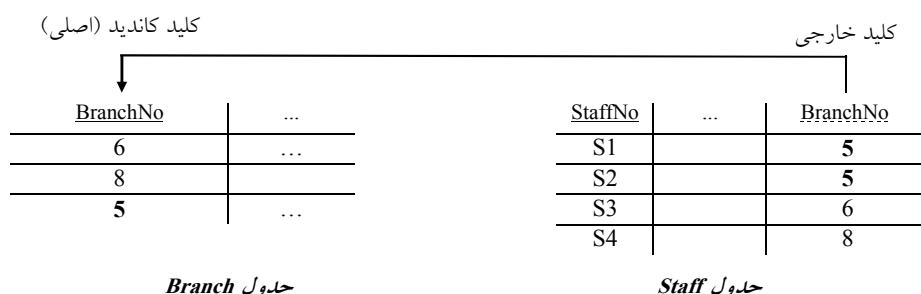
فرم جداول و تعداد رکوردهای جداول Branch و Staff پس از انجام دستور Delete به صورت زیر است:

نام دستور	SQL دستور
c	Delete From Branch where BranchNo=4



فرم جداول و تعداد رکوردهای جداول Staff Branch پس از انجام دستور Delete به صورت زیر است:

نام دستور	دستور SQL
e	Delete From Branch where BranchNo=3



توجه: علاوه بر ترتیب $abdce$ که در گزینه اول آمده است و پاسخ سوال نیز می‌باشد سایر ترتیب‌های $abdec$ و $abdec$ و $adbce$ و $adbee$ نیز درست و امکان پذیر است. بنابراین گزینه‌های دوم، سوم و چهارم را به طور کامل کتاب می‌گذاریم، پس پُر واضح است که گزینه اول پاسخ سوال است.

تست‌های فصل هشتم: وابستگی تابعی

۱ - کدام گزینه، معادل مجموعه وابستگی‌های تابعی مقابله است؟ (مهندسی کامپیوuter- دولتی ۸۳)

$$\{(X, Y, Z) \rightarrow ABCD, A \rightarrow B, X \rightarrow Y\}$$

$$\{(X, Y) \rightarrow ACD, A \rightarrow B, X \rightarrow Y\} \quad (1)$$

$$\{(X, Y, Z) \rightarrow ABCD, XA \rightarrow BY\} \quad (2)$$

$$\{(X, Z) \rightarrow ACD, A \rightarrow B, X \rightarrow Y\} \quad (3)$$

$$\{(Y, Z) \rightarrow ACD, A \rightarrow B, X \rightarrow Y\} \quad (4)$$

۲ - جدولی با پنج ستون A, B, C, D, E دارای وابستگی‌های زیر است. کدام گزینه، کلید کاندید این جدول است؟ (مهندسی IT - دولتی ۸۳)

$$\{A \rightarrow BC, B \rightarrow D, D \rightarrow A\}$$

$$(A, D, E) \quad (4)$$

$$(A, D) \quad (3)$$

$$(A, E) \quad (2)$$

$$(D, E) \quad (1)$$

۳ - در بانک زیر کلید کاندید کدام است؟ (مهندسی کامپیوuter- دولتی ۸۴)

$$R = \{A, B, C, D, E, H\}$$

$$F = \{A \rightarrow B, B \rightarrow C, D \rightarrow E, H \rightarrow A\}$$

$$ADH \quad (4)$$

$$ABD \quad (3)$$

$$HD \quad (2)$$

$$EH \quad (1)$$

۴ - رابطه R(A,B,C,D,E,F,G,H) را با وابستگی‌های کاربردی زیر در نظر بگیرید:

کدام یک از وابستگی‌های کاربردی زیر می‌تواند حذف شود، بدون آن‌که کلید اصلی R تغییری بکند؟

(مهندسی IT - دولتی ۸۴)

$$A \rightarrow BCD$$

$$AD \rightarrow E \quad (1)$$

$$AD \rightarrow E$$

$$F \rightarrow GH \quad (2)$$

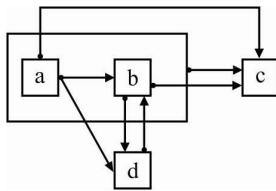
$$EFG \rightarrow H$$

$$EFG \rightarrow H \quad (3)$$

$$F \rightarrow GH$$

$$A \rightarrow BCD \quad (4)$$

۵ - نمودار FD زیر را در نظر بگیرید. کدام FD متعلق به مجموعه حداقل FD ها است؟ (مهندسی IT - دولتی ۸۵)



$$b \rightarrow c \quad (1)$$

$$a \rightarrow c \quad (2)$$

$$a \rightarrow d \quad (3)$$

$$ab \rightarrow c \quad (4)$$

۶ - حداقل مجموعه وابستگی‌های تابعی S' که معادل مجموعه وابستگی‌های تابعی S می‌باشد، کدام است؟

(مهندسی IT - دولتی ۸۶)

$$S = \{A \rightarrow B, A \rightarrow C, A \rightarrow E, B \rightarrow C, B \rightarrow E, C \rightarrow E\}$$

$$S' = \{A \rightarrow B, A \rightarrow C, C \rightarrow E, B \rightarrow E\} \quad (1)$$

$$S' = \{A \rightarrow B, B \rightarrow C, C \rightarrow E\} \quad (2)$$

$$S' = \{A \rightarrow B, A \rightarrow E, B \rightarrow E\} \quad (3)$$

$$S' = \{A \rightarrow B, A \rightarrow C, A \rightarrow E, C \rightarrow E\} \quad (4)$$

- ۷ - رابطه $R(A,B,C,D,E,F)$ با مجموعه وابستگی‌های FD مفروض است. کلید کاندید در رابطه R کدام است؟ (مهندسی IT-آزاد ۸۵)

$$FD = \{AB \rightarrow C, DB \rightarrow E, C \rightarrow F, B \rightarrow CE, A \rightarrow BE, D \rightarrow C\}$$

$$(A, B, D) \quad (4)$$

$$(A, D) \quad (3)$$

$$(B, D) \quad (2)$$

$$(A, B) \quad (1)$$

- ۸ - مجموعه غیرقابل کاهش وابستگی‌ها برای مجموعه وابستگی FD شامل چند وابستگی است؟ (مهندسی IT-آزاد ۸۵)

$$FD = \{AC \rightarrow BD, B \rightarrow DE, A \rightarrow D, A \rightarrow C, BC \rightarrow E, D \rightarrow E, A \rightarrow E\}$$

$$6 \quad (4)$$

$$5 \quad (3)$$

$$4 \quad (2)$$

$$3 \quad (1)$$

- ۹ - اگر مجموعه وابستگی زیر در رابطه $R(A,B,C,D,E,F,G)$ مفروض باشد، آنگاه کدام وابستگی زیر را می‌توان نتیجه گرفت: (مهندسی IT-آزاد ۸۶)

$$\{A \rightarrow B, CD \rightarrow F, B \rightarrow D, B \rightarrow C, BE \rightarrow G, G \rightarrow F, AG \rightarrow B\}$$

$$A \rightarrow F \quad (4)$$

$$G \rightarrow D \quad (3)$$

$$G \rightarrow B \quad (2)$$

$$B \rightarrow G \quad (1)$$

- ۱۰ - در رابطه با (x,y,z,w) اگر $x \rightarrow z$ و $y \rightarrow w$ برقرار باشد، نامزد کلیدی رابطه R کدام است؟ (مهندسی کامپیوuter-دولتی ۸۷)

$$(1) \text{ نامزد کلیدی } (x,y) \text{ است.}$$

$$(2) \text{ نامزد کلیدی فقط } x \text{ یا فقط } y \text{ است.}$$

$$(3) \text{ نامزد کلیدی } (x,y,z,w) \text{ است.}$$

- ۱۱ - فرض کنید AB کلید اصلی شمای $R(A,B,C,D,E,H,K)$ باشد، کدام یک از وابستگی‌های تابعی زیر وابستگی جزئی است؟ (مهندسی کامپیوuter-آزاد ۸۷)

$$DEH \rightarrow K \quad (4)$$

$$D \rightarrow EH \quad (3)$$

$$A \rightarrow CD \quad (2)$$

$$AB \rightarrow E \quad (1)$$

- ۱۲ - اگر A یک مجموعه از صفات در شمای R باشد و داشته باشیم $A \xrightarrow{\text{FFD}} R$ وابستگی تابعی کامل به A دارد) در این صورت A شمای R است. (مهندسی IT-آزاد ۸۷)

$$(1) \text{ کلید کاندید}$$

$$(2) \text{ کلید اصلی}$$

$$(3) \text{ کلید فرعی}$$

- ۱۳ - رابطه $R(A,B,C,D,E,H,K,L)$ با وابستگی‌های تابعی زیر داده شده است. کلید کاندید R کدام است؟ (مهندسی IT-آزاد ۸۷)

$$F = \{AB \rightarrow L, C \rightarrow H, CD \rightarrow E, EH \rightarrow B, K \rightarrow C\}$$

$$ADK \quad (4)$$

$$ABDHK \quad (3)$$

$$K \quad (2)$$

$$ABK \quad (1)$$

- ۱۴ - اگر در جدول ST داشته باشیم $T \# \rightarrow \text{Code}$ ، آنگاه این جدول چند نامزد کلیدی (Candidate Key) دارد؟ (مهندسی کامپیوuter-دولتی ۸۸)

$ST(S\#, T\#, Date, Time, Code)$

(۴) چهار

(۳) سه

(۲) دو

(۱) یک

۱۵ - کدام مورد، درباره وابستگی‌های تابعی (Functional Dependencies) غلط می‌باشد؟ (مهندسی IT- دولتی ۸۸)

$$AB \rightarrow C, D \rightarrow AE, AF \rightarrow C, EF \rightarrow GA \Rightarrow AD \rightarrow C \quad (1)$$

$$AB \rightarrow C, D \rightarrow AE, AE \rightarrow C, EF \rightarrow GA \Rightarrow DF \rightarrow G \quad (2)$$

$$X \rightarrow Y \wedge YZ \rightarrow W \Rightarrow XZ \rightarrow W \quad (3)$$

$$X \rightarrow Y \wedge Y \rightarrow Z \wedge X \rightarrow W \Rightarrow XZ \rightarrow W \quad (4)$$

۱۶ - رابطه R با وابستگی‌های تابعی F داده شده است. کلید اصلی R کدام است؟ (مهندسی IT- آزاد ۸۸)

R(A,B,C,D,E,F,G,H)

$$F = \{ABC \rightarrow DE, G \rightarrow DE, E \rightarrow F, H \rightarrow E, F \rightarrow G\}$$

ABC (۴)

ABE (۳)

ABCH (۲)

ABH (۱)

۱۷ - فرض کنید AB کلید اصلی شمای R(A,B,C,D,E,H) باشد. کدام یک از وابستگی‌های تابعی زیر، وابستگی انقلالی است؟ (مهندسی IT- آزاد ۸۷)

$$B \rightarrow D \quad (4)$$

$$D \rightarrow EH \quad (3)$$

$$A \rightarrow DEH \quad (2)$$

$$AB \rightarrow C \quad (1)$$

۱۸ - فرض کنید (U) R = عنوان یک رابطه باشد که روی مجموعه صفات U تعریف شده است و وابستگی‌های تابعی موجود در مجموعه F در مورد آن اعمال شده است. با فرض این که R دارای یک کلید منحصر به فرد K است. کدام یک از این نتیجه‌ها در مورد گزاره زیر صادق است؟ (مهندسی IT- دولتی ۸۸)

$$\forall X \subseteq U \forall A \in U : X \rightarrow A \in F^+ \wedge A \notin X$$

$$X \not\subseteq K \quad (2)$$

$$X \subset K \quad (1)$$

(A ∈ K) (۴) جزئی از کلید اصلی است. (A ∉ K) (۳) جزئی از کلید اصلی نیست.

۱۹ - رابطه R(A,B,C,D,E) و وابستگی‌های تابعی زیر را در نظر بگیرید.

$$A \rightarrow B$$

$$AB \rightarrow CD$$

$$D \rightarrow ABC$$

(مهندسی کامپیوuter- دولتی ۸۹)

ABD (۴)

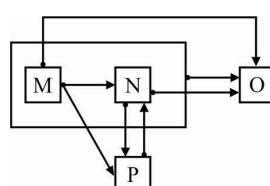
AB (۳)

AE (۲)

AD (۱)

کدام گزینه کلید رابطه است؟

۲۰ - نمودار وابستگی‌های تابعی (FD) زیر را در نظر بگیرید:



(مهندسی کامپیووتر-آزاد ۸۹ گروه ب)

$$M \rightarrow O \quad (4)$$

$$M \rightarrow P \quad (3)$$

$$(MN) \rightarrow O \quad (2)$$

$$N \rightarrow O \quad (1)$$

- ۲۱ - رابطه $R(A,B,C,D,E)$ با وابستگی‌های تابعی زیر را در نظر بگیرید:

$$A \rightarrow B, B \rightarrow C, C \rightarrow A, D \rightarrow E, E \rightarrow D$$

تعداد ابرکلید (فوق کلید) ها و کلیدهای کاندید رابطه (به ترتیب از راست به چپ) چندتا است؟

(مهندسی کامپیووتر-آزاد ۸۹ گروه ب)

$$(1) \text{ ابرکلید} = 8 \text{ و کلید کاندید} = 21$$

$$(2) \text{ ابرکلید} = 18 \text{ و کلید کاندید} = 7$$

$$(3) \text{ ابرکلید} = 21 \text{ و کلید کاندید} = 8$$

- ۲۲ - رابطه $R(A,B,C,D,E)$ و وابستگی‌های تابعی (Functional Dependency) زیر را در نظر بگیرید. فرضکنید نمی‌دانیم \square چیست و \square می‌تواند هر زیر مجموعه غیرتهی از صفات R باشد.

$$A \rightarrow BC$$

$$CD \rightarrow E$$

$$\square \rightarrow D$$

(مهندسی کامپیووتر-دولتی ۹۰)

کدام یک از گزاره‌های زیر مستقل از \square صحیح است؟(I) هر کلید کاندید R شامل A است.(II) هیچ کلید کاندیدی از R شامل C نیست.(III) بعضی از کلیدهای کاندید R شامل C هستند و بعضی دیگر شامل C نمی‌باشند.

$$(1) I \text{ و } II \quad (2) \text{ فقط } III \quad (3) I \text{ و } III$$

- ۲۳ - فرض کنید رابطه‌ای به صورت (V, R) که V مجموعه‌ای از صفات است و R با وابستگی‌های تابعیداده شده باشد که در آن $x_1 \rightarrow y_1, x_2 \rightarrow y_2, \dots, x_n \rightarrow y_n$ زیر مجموعه V هستند، می‌باشند. کدام یک از گزینه‌های زیر صحیح است؟ (U) اجتماع مجموعه‌ای است).

(مهندسی IT-دولتی ۹۰)

$$(1) \text{ اگر } \bigcup_{i=1}^n (x_i - y_i) \text{ یک ابرکلید باشد، آن‌گاه } R \text{ تنها یک کلید کاندید خواهد داشت.}$$

$$(2) \text{ اگر } V - \bigcup_{i=1}^n (x_i - y_i) \text{ یک ابرکلید باشد، آن‌گاه } R \text{ تنها یک کلید کاندید خواهد داشت.}$$

$$(3) \text{ اگر } V - \bigcup_{i=1}^n (y_i - x_i) \text{ یک ابرکلید باشد، آن‌گاه } R \text{ حداقل یک کلید کاندید خواهد داشت.}$$

$$(4) \text{ اگر } V - \bigcup_{i=1}^n (y_i - x_i) \text{ یک ابرکلید باشد، آن‌گاه } R \text{ تنها یک کلید کاندید خواهد داشت.}$$

- ۲۴ - رابطه R با وابستگی‌های تابعی F به صورت زیر مفروض است. کلید اصلی R کدام است؟

(مهندسی کامپیووتر-آزاد ۹۰ گروه الف)

$$R(A,B,C,D,E,F,G,H,I,J)$$

$$F\{ABD \rightarrow EG, C \rightarrow DG, E \rightarrow FG, I \rightarrow H, H \rightarrow J, AB \rightarrow C, G \rightarrow F\}$$

$$ABDI \quad (4)$$

$$ABD \quad (3)$$

$$AB \quad (2)$$

$$ABI \quad (1)$$

-۳۲ - رابطه $R(A,B,C,D,E)$ و مجموعه وابستگی های تابعی F را در نظر بگیرید.

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

کدام یک از گزینه های نمی تواند یک کلید کاندید برای R باشد؟ (مهندنس کامپیووتر-دولتی ۹۵)

- | | | | |
|--------|--------|-------|-------|
| CD (۴) | BC (۳) | E (۲) | B (۱) |
|--------|--------|-------|-------|

-۳۳ - رابطه $R(A,B,C,D,E)$ و مجموعه وابستگی های تابعی زیر را در نظر بگیرید. کلید کاندید این رابطه

$$FD = \{(A, B) \rightarrow C, B \rightarrow D, (D, E) \rightarrow C\}$$

کدام است؟ (مهندنس کامپیووتر-دولتی ۹۶)

- | | | | |
|--------------------|--------------------|-----------------|-----------------|
| {(A, B, D, E)} (۴) | {(A, B, C, E)} (۳) | {(A, B, E)} (۲) | {(A, B, D)} (۱) |
|--------------------|--------------------|-----------------|-----------------|

-۳۴ - در رابطه R با مجموعه صفات $\{A, B, C, D\}$, اگر وابستگی های تابعی (FD) به صورت

$$F = \{(A, B) \rightarrow (C, D), D \rightarrow B\}$$

کاندید این رابطه کدام است؟ (مهندنس IT- دولتی ۹۶)

- | | | | |
|------------|----------------|----------------|-----------|
| (A, D) (۴) | (A, D) فقط (۳) | (A, B) فقط (۲) | D فقط (۱) |
|------------|----------------|----------------|-----------|

-۳۵ - با توجه به رابطه $R(A, B, C, D, E, F)$ و مجموعه وابستگی های تابعی زیر، رابطه R چند کلید کاندید

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

- | | | | |
|-------|-------|-------|-------|
| 1 (۴) | 2 (۳) | 3 (۲) | 4 (۱) |
|-------|-------|-------|-------|

-۳۶ - با توجه به رابطه $R(A, B, C, D, E)$ و مجموعه وابستگی های تابعی زیر، کدام مورد نادرست است؟

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

(مهندنس IT- دولتی ۹۷) ۱) و $\{C, B\}$ و $\{B\}$ هر دو کلید کاندید هستند.

۲) و $\{E\}$ و $\{A\}$ و $\{C, D\}$ هر دو کلید کاندید هستند.

۳) و $\{E\}$ و $\{C, D\}$ هر دو کلید کاندید هستند.

-۳۷ - رابطه $R(A, B, C, D, E)$ و وابستگی های تابعی زیر را در نظر بگیرید:

$$f = \{A \rightarrow B, AB \rightarrow CD, D \rightarrow ABC\}$$

(مهندنس کامپیووتر- دولتی ۹۸) کلید کاندید رابطه کدام است؟

- | | | | |
|--------|--------|--------|--------|
| BE (۴) | AE (۳) | AD (۲) | AB (۱) |
|--------|--------|--------|--------|

-۳۸ - با در نظر گرفتن رابطه $R(A, B, C, D, E)$ و مجموعه وابستگی های تابعی

$$F = \{AB \rightarrow C, CD \rightarrow E, DE \rightarrow B\}$$

(مهندنس IT- دولتی ۹۸) این رابطه چند ابر کلید دارد؟

- | | | | |
|--------|--------|-------|-------|
| 12 (۴) | 10 (۳) | 7 (۲) | 3 (۱) |
|--------|--------|-------|-------|

-۳۹ - با در نظر گرفتن رابطه $R(A, B, C, D, E)$ و مجموعه وابستگی های تابعی

$$F = \{AB \rightarrow C, CD \rightarrow E\}$$

(مهندنس کامپیووتر- دولتی ۹۹) این رابطه چند ابر کلید دارد؟

- | | | | |
|-------|-------|-------|-------|
| 7 (۴) | 4 (۳) | 3 (۲) | 1 (۱) |
|-------|-------|-------|-------|

-۴۰ - رابطه $R(A, B, C, D, E, F)$ را در نظر بگیرید. فرض کنیم A و (B, C) دو کلید کاندید این رابطه و تنها

کلیدهای کاندید باشند. این رابطه چند ابر کلید دارد؟ (مهندنس کامپیووتر- دولتی ۹۹)

- | | | | |
|--------|--------|--------|-------|
| 40 (۴) | 32 (۳) | 24 (۲) | 8 (۱) |
|--------|--------|--------|-------|

پاسخ تست‌های فصل هشتم: وابستگی تابعی

۱- گزینه (۳) صحیح است.

به طور کلی دو مجموعه وابستگی تابعی F_1 و F_2 معادل هستند اگر و فقط اگر $F_1^+ = F_2^+$. ابتدا باید مجموعه وابستگی‌های مطرح شده مورد بررسی قرار گیرد.

با کمی دقت می‌توان موارد زائد را حذف کرد، از آنجا که $X \rightarrow Y$ ، پس می‌توان Y را در سمت چپ از $(X, Z) \rightarrow ABCD$ نوشت $(X, Y, Z) \rightarrow ABCD$.

$$\left. \begin{array}{l} XYZ \rightarrow ABCD \\ X \rightarrow Y \end{array} \right\} XZ \rightarrow ABCD$$

هم‌چنین چون $B \rightarrow A$ پس می‌توان B را در سمت راست وابستگی از کنار A حذف کرد و به جای $(X, Z) \rightarrow ACD$ نوشت $(X, Y, Z) \rightarrow ABCD$.

$$\left. \begin{array}{l} XZ \rightarrow ABCD \\ A \rightarrow B \end{array} \right\} XZ \rightarrow ACD$$

بنابراین داریم:

$$\{(X, Z) \rightarrow ACD, A \rightarrow B, X \rightarrow Y\}$$

پُرواضح است که مجموعه فوق، حاصل از حذف موارد زائد در مجموعه وابستگی‌های مطرح شده دقیقاً مطابق گزینه سوم است.

سایر گزینه‌ها شرایط لازم برای معادل بودن با مجموعه داده شده را دارا نیستند.

۲- گزینه‌های (۱) و (۲) صحیح است.

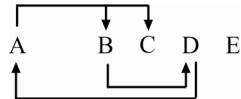
وابستگی‌های زیر را برای جدول $R(A, B, C, D, E)$ داریم:

$$A \rightarrow BC$$

$$B \rightarrow D$$

$$D \rightarrow A$$

دیاگرام وابستگی‌های فوق به صورت زیر است:



گزینه سوم نادرست است. زیرا، ستون E در سمت راست هیچ یک از وابستگی‌های تابعی نیامده است، بنابراین ستون E حتماً عضو کلید کاندید باید باشد.

با کمی بررسی در نمودار وابستگی‌ها، پُرواضح است که ستون A ، ستون‌های B و C و به واسطه ستون B و رابطه تعدی، ستون D را هم می‌دهد. بنابراین ستون A ، صفت‌های B ، C و D را می‌دهد.

پس از آنجا که A ، ستون‌های B ، C و D را می‌دهد و ستون E هم حتماً باید عضو کلید کاندید باشد، پس AE کلید کاندید این جدول تلقی می‌شود. از طرف دیگر چون وابستگی $D \rightarrow A$ برقرار است، و به تبع ستون A هم ستون‌های B و C را می‌دهد، می‌توان نتیجه گرفت که DE هم، کلید کاندید دیگری برای این جدول است.

از طرفی دیگر هم، چون وابستگی $D \rightarrow B$ برقرار است و وابستگی $D \rightarrow A$ هم برقرار است و به تبع ستون A هم ستون‌های B و C را می‌دهد، می‌توان نتیجه گرفت که BE کلید کاندید دیگری برای این جدول است. بنابراین جدول $R(A,B,C,D,E)$ ، دارای سه کلید کاندید AE، DE و BE می‌باشد و کلید کاندید دیگری هم ندارد.

با توجه به وابستگی‌های مطرح شده:

$$A \rightarrow BC$$

$$B \rightarrow D$$

$$D \rightarrow A$$

بستار کلیدهای کاندید AE، DE و BE به صورت زیر است:

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

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

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

بنابراین گزینه‌های اول و دوم درست هستند.

گزینه چهارم نادرست است. زیرا ADE ابرکلید است، اما کلید کاندید نیست، زیرا ADE، عضو زائدی مثل A یا D دارد، چون وجود دو ستون A و D به صورت با هم در ساخت یک کلید کاندید با توجه به وابستگی‌های مطرح شده، ضروری نیست و وجود هر یک به تنها به همراه ستون E، برای ساخت کلید کاندید، لازم و کافی است.

۳- گزینه (۲) صحیح است.

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

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

۲- عضو زائد نداشته باشد.

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

توجه: ما نام این روابط را به عنوان مبدع آن «قوانين ارسسطو» نام‌گذاری کردیم، این قوانین به «قوانين چهارگانه ارسسطو» نیز موسوم است. این قوانین می‌تواند جایگزین مناسبی برای قوانین آرمسترانگ باشد.

قانون اول ارسسطو

روش اول:

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

روش دوم:

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

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

A → BC ⇒ BC - A = BC

مثال:

توجه: استفاده از روش اول مستلزم گام ابتدایی حذف وابستگی‌های بدیهی است، اگر در حذف

وابستگی‌های بدیهی دچار خطأ می‌شوید، از روش دوم استفاده نمایید.

توجه: وابستگی بدیهی: اگر در یک وابستگی، بخش سمت راست، زیر مجموعه بخش سمت چپ باشد آن وابستگی را، وابستگی بدیهی می‌نامند.

مثال:

$$AB \rightarrow A$$

با توجه به وابستگی‌های مطرح شده داریم:

$$A \rightarrow B$$

$$B \rightarrow C$$

$$D \rightarrow E$$

$$H \rightarrow A$$

$$ABCDEH - ABCE = DH$$

بنابر رابطه فوق صفات DH حتماً باید عضو کلید کاندید باشد.

بستان صفات DH به صورت زیر است:

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

براساس بستان فوق، صفات DH، همه ستون‌ها را تولید می‌کند، پس صفات DH کلید کاندید می‌باشد.

قانون دوم ارسسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روشن اول یا دوم) همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

گزینه اول نادرست است. زیرا، عضو کلید کاندید DH را ندارد.

گزینه سوم نادرست است. زیرا، عضو کلید کاندید DH را ندارد.

گزینه چهارم نادرست است. زیرا، عضو زائد A را دارد، هر چند عضو کلید کاندید DH را دارد. صفات DH همه ستون‌ها را به تنهایی با استفاده از وابستگی‌های مطرح شده تولید می‌کنند، بنابراین صفت A، عضو زائد است. مجموعه صفات ADH در واقع ابرکلید است.

۴- گزینه (۳) صحیح است.

طرح محترم: در مباحث پایگاه داده، مفهومی، تحت عنوان وابستگی کاربردی وجود ندارد، بنابراین، عبارت وابستگی کاربردی به وابستگی تابعی باید اصلاح گردد.

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

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

۲- عضو زائد نداشته باشد.

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

قانون اول ارسسطو

روشن اول:

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

روشن دوم:

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

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

$$A \rightarrow BCD$$

$$AD \rightarrow E$$

$$EFG \rightarrow H$$

$$F \rightarrow GH$$

$$ABCDEFGH - BCDEGH = AF$$

بنابر رابطه فوق صفات AF حتماً باید عضو کلید کاندید باشد.

بستار صفات AF به صورت زیر است:

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

براساس بستار فوق، صفات AF، همه ستون‌ها را تولید می‌کند، پس صفات AF کلید کاندید می‌باشد.

قانون دوم ارسسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم) همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

مجدداً وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E, F, G, H)$ R در نظر بگیرید:

$$A \rightarrow BCD$$

$$AD \rightarrow E$$

$$EFG \rightarrow H$$

$$F \rightarrow GH$$

حال اگر وابستگی $\rightarrow H$ EFG، از مجموعه وابستگی‌ها حذف گردد، تأثیری در کلید کاندید رابطه R نخواهد داشت. زیرا صفت H در سمت راست وابستگی $\rightarrow H$ EFG توسط وابستگی $\rightarrow GH$ F تولید می‌گردد. در واقع بود و نبود وابستگی $\rightarrow H$ EFG در یافتن کلید کاندید رابطه R تأثیری نخواهد داشت. بنابراین بدون وابستگی $\rightarrow H$ EFG داریم:

$$A \rightarrow BCD$$

$$AD \rightarrow E$$

$$F \rightarrow GH$$

$$ABCDEFGH - BCDEGH = AF$$

بنابراین با حذف وابستگی $\rightarrow H$ EFG، صفات AF هم چنان عضو کلید کاندید رابطه R خواهد بود. بستار صفات AF، براساس وابستگی‌های فوق و بدون استفاده از وابستگی $\rightarrow H$ EFG، به صورت زیر است:

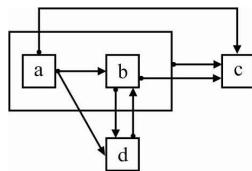
$$\{AF\}^+ = \{A, F, B, C, D, E, G, H\}$$

براساس بستار فوق، صفات AF، همه ستون‌ها را تولید می‌کند. پس صفات AF هم‌چنان کلید کاندید می‌باشد.

- گزینه (۱) صحیح است.

وابستگی‌های تابعی متناظر با نمودار مطرح شده به صورت زیر است:

$a \rightarrow b$
 $a \rightarrow c$
 $b \rightarrow c$
 $a \rightarrow d$
 $b \rightarrow d$
 $d \rightarrow b$
 $ab \rightarrow c$



راه حل اول:

به دست آوردن مجموعه حداقل FDها براساس قوانین زیر:

- ۱- سمت راست همه وابستگی‌ها توسط عمل تجزیه، تک خصیصه‌ای باشد.
- ۲- سمت چپ همه وابستگی‌ها در صورت امکان کمینه باشد.
- ۳- همه وابستگی‌های زائد حذف گردد.

وابستگی $c \rightarrow ab$ به دلیل وجود وابستگی $b \rightarrow a$ قابل تغییر به وابستگی $c \rightarrow a$ است.

در ادامه پس از حذف وابستگی‌های زائد داریم:

حذف وابستگی $c \rightarrow a$ به دلیل زیر:

$$\begin{cases} a \rightarrow b \\ b \rightarrow c \end{cases} \Rightarrow a \rightarrow c$$

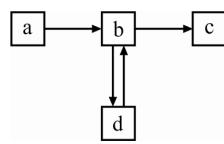
حذف وابستگی $d \rightarrow a$ به دلیل زیر:

$$\begin{cases} a \rightarrow b \\ b \rightarrow d \end{cases} \Rightarrow a \rightarrow d$$

بنابراین مجموعه حداقل FD‌ها به صورت زیر است:

$a \rightarrow b$
 $b \rightarrow c$
 $b \rightarrow d$
 $d \rightarrow b$

هم‌چنان نمودار وابستگی‌های مجموعه حداقل FD‌ها به صورت زیر است:



بنابراین وابستگی $c \rightarrow b$ مطرح شده در گزینه اول، متعلق به مجموعه حداقل FD‌ها است.

راه حل دوم: (حذف گزینه)

اگر وابستگی متعلق به مجموعه حداقل FD‌ها باشد، باید توسط وابستگی‌های دیگر تولید گردد. گزینه اول درست است. زیرا، وابستگی $c \rightarrow b$ متعلق به مجموعه حداقل FD‌ها است، زیرا توسط وابستگی‌های دیگر قابل تولید نیست.

یعنی اگر صفت b را داشته باشیم، نمی‌توان از صفت b بدون استفاده از وابستگی $c \rightarrow b$ و با استفاده از وابستگی‌های دیگر به صفت c رسید. پُر واضح است که نمی‌شود. گزینه دوم نادرست است. زیرا وابستگی $c \rightarrow a$ قابل حذف است، چون اگر وابستگی $c \rightarrow a$ حذف گردد،

از صفت a توسط وابستگی های دیگر می توان به صفت c رسید، بنابراین وابستگی $c \rightarrow a$ زائد است.

$$\begin{cases} a \rightarrow b \\ b \rightarrow c \end{cases} \Rightarrow a \rightarrow c$$

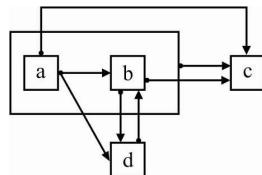
گزینه سوم نادرست است. زیرا، وابستگی $b \rightarrow a$ قابل حذف است، چون اگر وابستگی $a \rightarrow d$ حذف گردد، از صفت a توسط وابستگی های دیگر می توان به صفت d رسید. بنابراین وابستگی $a \rightarrow d$ زائد است.

$$\begin{cases} a \rightarrow b \\ b \rightarrow d \end{cases} \Rightarrow a \rightarrow d$$

گزینه چهارم نادرست است. زیرا، وابستگی $c \rightarrow ab$ به دلیل وجود وابستگی $b \rightarrow a$ قابل تغییر به $a \rightarrow c$ می باشد و مطابق آنچه پیش از این گفتیم وابستگی $c \rightarrow a$ زائد است.

راه حل سوم: (با استفاده از نمودار)

نمودار مطرح شده را در نظر بگیرید:



گزینه اول درست است. زیرا، وابستگی $c \rightarrow b$ متعلق به مجموعه حداقل FD ها است، زیرا توسط مسیرهای دیگر قابل دسترسی نیست.

یعنی اگر مطابق نمودار فوق، مربع صفت b را داشته باشیم، هیچ مسیر دیگری به مربع صفت c وجود ندارد. گزینه دوم نادرست است. زیرا، وابستگی $c \rightarrow a$ قابل حذف است، چون اگر وابستگی $a \rightarrow c$ حذف گردد، از مربع صفت a توسط مسیر $a \rightarrow b \rightarrow c$ می توان به مربع صفت c دسترسی پیدا کرد. بنابراین وابستگی $a \rightarrow c$ زائد است.

گزینه سوم نادرست است. زیرا، وابستگی $a \rightarrow d$ قابل حذف است، چون اگر وابستگی $a \rightarrow d$ حذف گردد، از مربع صفت a توسط مسیر $a \rightarrow b \rightarrow d$ می توان به مربع صفت d دسترسی پیدا کرد. بنابراین وابستگی $a \rightarrow d$ زائد است.

گزینه چهارم نادرست است، زیرا، وابستگی $ab \rightarrow c$ قابل حذف است، چون اگر وابستگی $ab \rightarrow c$ حذف گردد، از مربع صفت a توسط مسیر $a \rightarrow b \rightarrow c$ می توان به مربع صفت c دسترسی پیدا کرد. بنابراین وابستگی $ab \rightarrow c$ زائد است.

۶- گزینه (۲) صحیح است.

وابستگی های مطرح شده به صورت زیر است:

- A → B
- A → C
- A → E
- B → C
- B → E
- C → E

مجموعه حداقل FD ها براساس قوانین زیر ایجاد می‌گردد:

- ۱- سمت راست همه وابستگی‌ها توسط عمل تجزیه تک خصیصه‌ای باشد.
 - ۲- سمت چپ همه وابستگی‌ها در صورت امکان کمینه باشد.
 - ۳- همه وابستگی‌های زائد حذف گردد.
- حذف وابستگی C → A به دلیل زیر:

$$\begin{cases} A \rightarrow B \\ B \rightarrow C \end{cases} \Rightarrow A \rightarrow C$$

حذف وابستگی E → A به دلیل زیر:

$$\begin{cases} A \rightarrow B \\ B \rightarrow E \end{cases} \Rightarrow A \rightarrow E$$

حذف وابستگی E → B به دلیل زیر:

$$\begin{cases} B \rightarrow C \\ C \rightarrow E \end{cases} \Rightarrow B \rightarrow E$$

بنابراین مجموعه حداقل FD ها به صورت زیر است:

$$A \rightarrow B$$

$$B \rightarrow C$$

$$C \rightarrow E$$

۷- گزینه (۳) صحیح است.

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

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

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

قانون اول ارسسطو

روش اول:

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

روش دوم:

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

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

$$AB \rightarrow C$$

$$DB \rightarrow E$$

$$C \rightarrow F$$

$$B \rightarrow CE$$

$$A \rightarrow BE$$

$$D \rightarrow C$$

$$ABCDEF - BCEF = AD$$

بنابر رابطه فوق صفات AD حتماً باید عضو کلید کاندید باشد. بستار صفات AD به صورت زیر است:
 $\{AD\}^+ = \{A, D, B, C, E, F\}$

براساس بستار فوق، صفات AD، همه ستون‌ها را تولید می‌کند، پس صفات AD کلید کاندید می‌باشد.

قانون دوم ارسسطو

هرگاه عضو کلید کاندید حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.
 گزینه اول نادرست است. زیرا، عضو کلید کاندید AD را ندارد.
 گزینه دوم نادرست است. زیرا، عضو کلید کاندید AD را ندارد.
 گزینه سوم درست است. زیرا، عضو کلید کاندید AD، به تنها یک کلید کاندید است.
 گزینه چهارم نادرست است. زیرا، عضو زائد B را دارد، هر چند عضو کلید کاندید AD را دارد. صفات AD همه ستون‌ها را به تنها یک با استفاده از وابستگی‌های مطرح شده تولید می‌کند، بنابراین صفت B، عضو زائد است. مجموعه صفات ABD در واقع ابرکلید است.

- گزینه (۲) صحیح است.

وابستگی‌های مطرح شده به صورت زیر است:

$$AC \rightarrow BD$$

$$B \rightarrow DE$$

$$A \rightarrow D$$

$$A \rightarrow C$$

$$BC \rightarrow E$$

$$D \rightarrow E$$

$$A \rightarrow E$$

مجموعه حداقل FD ها براساس قوانین زیر ایجاد می‌گردد:

۱- سمت راست همه وابستگی‌ها توسط عمل تجزیه، تک خصیصه‌ای باشد.

۲- سمت چپ همه وابستگی‌ها در صورت امکان کمینه باشد.

۳- همه وابستگی‌های زائد حذف گردد.

بنابراین، براساس قوانین فوق داریم:

$$(1) AC \rightarrow B$$

$$(2) AC \rightarrow D$$

$$(3) B \rightarrow D$$

$$(4) B \rightarrow E$$

$$(5) A \rightarrow D$$

$$(6) A \rightarrow C$$

$$(7) BC \rightarrow E$$

$$(8) D \rightarrow E$$

$$(9) A \rightarrow E$$

وابستگی $B \rightarrow AC$ به دلیل وجود وابستگی $C \rightarrow A$ قابل تغییر به وابستگی $A \rightarrow B$ است.

وابستگی $D \rightarrow AC$ به دلیل وجود وابستگی $C \rightarrow A$ قابل تغییر به وابستگی $A \rightarrow D$ است و چون وابستگی حاصل یعنی $A \rightarrow D$ تکراری است، این وابستگی حذف می‌گردد.

حذف وابستگی $E \rightarrow BC$ به دلیل زیر:

هدف رسیدن به صفت E است، از آن جا که صفت B در سمت راست وابستگی $BC \rightarrow E$ ، توسط وابستگی $E \rightarrow B$ ، به تنهایی قادر به تولید صفت E است، پس صفت C در سمت چپ وابستگی $BC \rightarrow E$ ، زائد است و وابستگی $E \rightarrow BC$ باقی می‌ماند و چون وابستگی حاصل یعنی $E \rightarrow B$ تکراری است، این وابستگی حذف می‌گردد.
حذف وابستگی $E \rightarrow A$ به دلیل زیر:

$$\begin{cases} A \rightarrow D \\ D \rightarrow E \end{cases} \Rightarrow A \rightarrow E$$

حذف وابستگی $D \rightarrow A$ به دلیل زیر:

$$\begin{cases} A \rightarrow B \\ B \rightarrow D \end{cases} \Rightarrow A \rightarrow D$$

حذف وابستگی $E \rightarrow B$ به دلیل زیر:

$$\begin{cases} B \rightarrow D \\ D \rightarrow E \end{cases} \Rightarrow B \rightarrow E$$

بنابراین مجموعه حداقل FD ها به صورت زیر است:

$$\begin{aligned} A &\rightarrow B \\ B &\rightarrow D \\ A &\rightarrow C \\ D &\rightarrow E \end{aligned}$$

۹- گزینه (۴) صحیح است.

وابستگی های مطرح شده به صورت زیر است:

$$\begin{aligned} A &\rightarrow B \\ CD &\rightarrow F \\ B &\rightarrow D \\ B &\rightarrow C \\ BE &\rightarrow G \\ G &\rightarrow F \\ AG &\rightarrow B \end{aligned}$$

سؤال: آیا می‌توان از صفت B بدون استفاده از وابستگی $G \rightarrow B$ و با استفاده از وابستگی های دیگر به

$$\text{صفت } G \text{ رسید؟} \quad \{B \rightarrow G\}^+ =$$

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

پاسخ: خیر، نمی‌توان رسید، زیرا خصیصه B با همراهی خصیصه E می‌تواند خصیصه G را تولید کند و به تنهایی قادر به این کار نیست. بنابراین گزینه اول نادرست است.

سؤال: آیا می‌توان از صفت G بدون استفاده از وابستگی $G \rightarrow B$ و با استفاده از وابستگی های دیگر به

$$\text{صفت } B \text{ رسید؟} \{G \rightarrow B\}^+ = \{G, F\}$$

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

پاسخ: خیر، نمی‌توان رسید، زیرا خصیصه G می‌تواند خصیصه F را تولید کند، اما خصیصه F در سمت چپ هیچ یک از وابستگی‌های تابعی نیامده است. بنابراین هیچ خصیصه دیگری را نمی‌تواند تولید کند.

.

.

.

سؤال: آیا می‌توان از صفت G بدون استفاده از وابستگی $D \rightarrow G$ و با استفاده از وابستگی‌های دیگر به

$$\text{صفت } D \text{ رسید؟} \{G \rightarrow D\}^+ = \{G, F\}$$

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

پاسخ: خیر، نمی‌توان رسید، زیرا خصیصه G می‌تواند خصیصه F را تولید کند، اما خصیصه F در سمت چپ هیچ یک از وابستگی‌های تابعی نیامده است. بنابراین هیچ خصیصه دیگری را نمی‌تواند تولید کند.

.

.

.

سؤال: آیا می‌توان از صفت A بدون استفاده از وابستگی $F \rightarrow A$ و با استفاده از وابستگی‌های دیگر به

$$\text{صفت } F \text{ رسید؟} \{A \rightarrow F\}^+ = \{A, B, C, D, F\}$$

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

پاسخ: بله، می‌توان رسید، روال زیر را در نظر بگیرید:

$$A \rightarrow B$$

$$\begin{cases} B \rightarrow C \\ B \rightarrow D \end{cases} \Rightarrow B \rightarrow CD$$

$$\begin{cases} B \rightarrow CD \\ CD \rightarrow F \end{cases} \Rightarrow B \rightarrow F$$

$$\begin{cases} A \rightarrow B \\ B \rightarrow F \end{cases} \Rightarrow A \rightarrow F$$

توجه: نیازی نیست، روال فوق را دنبال کنید، کافی است، با چشم، بر روی وابستگی‌های اصلی مطرح شده حرکت کیم. A خود را می‌دهد، A می‌دهد B را، B می‌دهد C و D را، C و D را داریم، پس می‌دهد F را، بنابراین از A به F بدون استفاده از وابستگی $A \rightarrow F$ رسیدیم، به همین سادگی، همین.

۱۰- گزینه (۱) صحیح است.

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

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

۲- عضو زائد نداشته باشد.

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

قانون اول ارسسطو

روش اول:

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

روش دوم:

$$\text{با توجه به وابستگی‌های مطرح شده برای رابطه } R(x,y,z,w) \text{ داریم:}$$

$$\begin{aligned} & R - \bigcup_{i=1}^n (x_i y_i) - (z_i w_i) = \text{عضو کلید کاندید} \\ & x \rightarrow z \\ & y \rightarrow w \\ & xyzw - zw = xy \end{aligned}$$

بنابر رابطه فوق صفات xy حتماً باید عضو کلید کاندید باشد.

بستانار صفات xy به صورت رویه‌رو است:

براساس بستانار فوق، صفات xy همه ستون‌ها را تولید می‌کند، پس صفات xy کلید کاندید می‌باشد.

قانون دوم ارسسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

گزینه اول درست است. زیرا، عضو کلید کاندید xy ، به تنهایی کلید کاندید است.

گزینه دوم و چهارم نادرست است، زیرا، عضو کلید کاندید xy را ندارد. صفت x یا y به تنهایی خاصیت کلیدی ندارد. xy هر دو با هم عضو کلید کاندید هستند.

گزینه سوم نادرست است. زیرا، اعضاء زائد z و w را دارد، هر چند عضو کلید کاندید xy را دارد. صفات xy همه ستون‌ها را به تنهایی و با استفاده از وابستگی‌های مطرح شده تولید می‌کنند، بنابراین صفات z و w اعضا زائد هستند. مجموعه صفات $xyzw$ در واقع ابرکلید است.

۱۱- گزینه (۲) صحیح است.

طبق تعریف وابستگی بخشی (جزئی)، عبارت است از وابستگی یک عنصر غیرکلیدی به بخشی از کلید کاندید. از آنجا که مطابق صورت سؤال AB کلید اصلی (کاندید) است. پس $A \rightarrow CD$ وابستگی دو عنصر غیرکلیدی C و D به A که بخشی از کلید است، را نشان می‌دهد. بنابراین می‌توان دریافت که $A \rightarrow CD$ یک وابستگی بخشی (جزئی) تلقی می‌شود. سایر گزینه‌ها با تعریف وابستگی بخشی انطباق ندارد.

۱۲- گزینه (۱) صحیح است.

مطابق مفروضات مطرح شده در صورت سؤال، چون کلیه خصیصه‌های R وابستگی تابع کامل به A دارند، پس A حتماً کلید کاندید این رابطه است، اما این که حتماً کلید اصلی آن باشد مشخص نیست، زیرا انتخاب کلید اصلی بر عهده طراح بانک و براساس مورد کاربرد، از یکی از کلیدهای کاندید انتخاب می‌گردد.

۱۳- گزینه (۴) صحیح است.

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

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

۲- عضو زائد نداشته باشد.

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

قانون اول ارسسطو

روش اول:

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

روش دوم:

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

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

$AB \rightarrow L$

$C \rightarrow H$

$CD \rightarrow E$

$EH \rightarrow B$

$K \rightarrow C$

$ABCDEFHKL - BCEHL = ADK$

بنابر رابطه فوق صفات ADK حتماً باید عضو کلید کاندید باشد.

$\{ADK\}^+ = \{A, D, K, C, E, H, B, L\}$ بستار صفات ADK به صورت مقابل است:

براساس بستار فوق، صفات ADK ، همه ستون‌ها را تولید می‌کند، پس صفات ADK کلید کاندید می‌باشد.

قانون دوم ارسسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

گزینه اول نادرست است. زیرا، عضو کلید کاندید ADK را ندارد.

گزینه دوم نادرست است. زیرا، عضو کلید کاندید ADK را ندارد.

گزینه سوم نادرست است. زیرا، اعضا زائد B و H را دارد، هر چند عضو کلید کاندید ADK را دارد، صفات ADK همه ستون‌ها را به تنهایی و با استفاده از وابستگی‌های مطرح شده تولید می‌کنند، بنابراین صفات B و H اعضا زائد هستند. مجموعه صفات $ABDHK$ در واقع ابرکلید است

گزینه چهارم درست است. زیرا، عضو کلید کاندید ADK ، به تنهایی کلید کاندید است.

۱۴- گزینه (۲) صحیح است.

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

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

۲- عضو زائد نداشته باشد.

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

قانون اول ارسسطو

روش اول:

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

روش دوم:

$$R - \bigcup_{i=1}^n [x_i, y_i] \in \text{اعضو کلید کاندید}$$

با توجه به وابستگی‌های مطرح شده برای رابطه $(S\#, T\#, Date, Time, Code)$ و اعلام $(S\#, T\#)$ به عنوان کلید کاندید از طرف طراح سؤال، داریم:

$S\#, T\# \rightarrow Date$

$S\#, T\# \rightarrow Time$

$S\#, T\# \rightarrow Code$

$Code \rightarrow T\#$

$$S\# \not\rightarrow Date \not\rightarrow Time \not\rightarrow Code \rightarrow T\# Date Time Code = S\#$$

بنابر رابطه فوق صفت $S\#$ حتماً باید عضو کلید کاندید باشد. بستار صفت $S\#$ به صورت زیر است:

$$\{S\#\}^+ = \{S\#\}$$

براساس بستار فوق، صفت $S\#$ فقط ستون $S\#$ را تولید می‌کند، پس صفت $S\#$ فقط عضو کلید کاندید می‌باشد و کلید کاندید نمی‌باشد.

قانون سوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول، برخی از ستون‌ها را تولید کند، بدین معنی است که، جدول موردنظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت.

با کمی دقت و بررسی، پُر واضح است که اگر صفت $T\#$ در کنار عضو کلید $S\#$ قرار بگیرد، همکاری صفات $(S\#, T\#)$ می‌تواند، همه ستون‌ها را تولید کند، بنابراین مطابق آنچه در صورت سؤال هم آمده است، $(S\#, T\#)$ یک کلید کاندید این جدول است.

بستار صفات $(S\#, T\#)$ به صورت زیر است:

$$\{S\#, T\#\}^+ = \{S\#, T\#, Date, Time, Code\}$$

براساس بستار فوق، صفات $(S\#, T\#)$ ، همه ستون‌ها را تولید می‌کنند، پس صفات $(S\#, T\#)$ کلید کاندید است.

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

هم‌چنین از آن‌جا که $T\# \rightarrow Code$. پس می‌توان ترکیب دو خصیصه $(S\#, Code)$ را هم کلید کاندید دیگری برای این جدول تلقی کرد. چون وقتی $(S\#, T\#)$ کلید کاندید است و همه ستون‌ها را تولید می‌کند، پس $(S\#, Code)$ هم کلید کاندید است و همه ستون‌ها را تولید می‌کند، زیرا در نهایت طبق وابستگی $Code \rightarrow T\#$ ، صفت $T\#$ را می‌دهد و $(S\#, Code)$ به $(S\#, T\#)$ تبدیل می‌گردد.

بستار صفات $(S\#, Code)$ به صورت زیر است:

$$\{S\#, Code\}^+ = \{S\#, Code, T\#, Date, Time\}$$

براساس بستار فوق، صفات $(S \neq, Code)$ ، همه ستون‌ها را تولید می‌کنند، پس صفات $(S \neq, Code)$ کلید کاندید است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده در مجموع دارای دو کلید کاندید است، که مطابق قانون سوم ارسسطو عضو کلید کاندید $S\#$ در بین هر دو کلید کاندید به طور مشترک قرار دارد.

توجه: طراح بانک، کلید کاندید $(S\#, T\#)$ را با رسم خط، به عنوان کلید اصلی انتخاب کرده است، بنابراین کلید کاندید $(S\#, Code)$ به عنوان کلید فرعی تلقی می‌گردد. این نتیجه مهم بود.

۱۵- گزینه (۱) صحیح است.

گزینه اول نادرست است. زیرا، وابستگی‌های مطرح شده به صورت زیر است:

$$AB \rightarrow C$$

$$D \rightarrow AE$$

$$AF \rightarrow C$$

$$EF \rightarrow GA$$

{سؤال: آیا می‌توان از صفات AD با استفاده از وابستگی‌های فوق به صفت C رسید؟}

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

پاسخ: خیر، نمی‌توان رسید. زیرا A، D، E را می‌دهد و A و E را می‌دهد و کار تمام می‌شود. به بیان دیگر، وابستگی تابعی \rightarrow AD قابل استنتاج از وابستگی‌های تابعی داده شده در گزینه اول نیست.

گزینه دوم درست است. زیرا، وابستگی‌های مطرح شده به صورت زیر است:

$$AB \rightarrow C$$

$$D \rightarrow AE$$

$$AE \rightarrow C$$

$$EF \rightarrow GA$$

{سؤال: آیا می‌توان از صفات DF با استفاده از وابستگی‌های فوق به صفت G رسید؟}

$$\{DF \rightarrow G\}^+ = \{DF\}^+ = \{D, F, A, E, \boxed{G}\}$$

پاسخ: بله، می‌توان رسید، زیرا D، F، A را می‌دهد، E را می‌دهد و E و F و G هم و A را می‌دهد.

راه حل کلاسیک:

$$D \rightarrow AE \Rightarrow D \rightarrow E \\ EF \rightarrow GA \Rightarrow DF \rightarrow GA \Rightarrow DF \rightarrow G$$

گزینه سوم درست است. زیرا، وابستگی‌های مطرح شده به صورت زیر است:

$$X \rightarrow Y$$

$$YZ \rightarrow W$$

{سؤال: آیا می‌توان از صفات XZ با استفاده از وابستگی‌های فوق به صفت W رسید؟}

$$\{XZ \rightarrow W\}^+ = \{XZ\}^+ = \{X, Z, Y, \boxed{W}\}$$

پاسخ: بله، می‌توان رسید، زیرا X, X را می‌دهد و Z, Z را می‌دهد و X, Y را می‌دهد و Z, Y را می‌دهد و W هم W را می‌دهد.

راه حل کلاسیک:

$$\left. \begin{array}{l} X \rightarrow Y \\ YZ \rightarrow W \end{array} \right\} \Rightarrow XZ \rightarrow W$$

گزینه چهارم درست است. زیرا، وابستگی‌های مطرح شده به صورت زیر است:

$$\begin{array}{l} X \rightarrow Y \\ Y \rightarrow Z \\ X \rightarrow W \end{array}$$

{سؤال: آیا می‌توان از صفات XZ با استفاده از وابستگی‌های فوق به صفت W رسید؟}

$$\{XZ \rightarrow W\}^+ = \{XZ\}^+ = \{X, Z, Y, \overline{W}\}$$

پاسخ: بله، می‌توان رسید، زیرا X, X را می‌دهد و Z, Z را می‌دهد و X, Y را می‌دهد و W را می‌دهد.

راه حل کلاسیک:

$$\left. \begin{array}{l} X \rightarrow Y \\ Y \rightarrow Z \end{array} \right\} \Rightarrow X \rightarrow Z$$

$$\left. \begin{array}{l} X \rightarrow Z \\ X \rightarrow W \end{array} \right\} \Rightarrow XZ \rightarrow W$$

۱۶- گزینه (۲) صحیح است.

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

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

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

قانون اول ارسسطو

روش اول:

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

روش دوم:

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

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

$$ABC \rightarrow DE$$

$$G \rightarrow DE$$

$$E \rightarrow F$$

$$H \rightarrow E$$

$$F \rightarrow G$$

$$ABCDEF GH - DEF GH = ABC$$

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

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

براساس بستار فوق، صفات ABCH، همه ستون‌ها را تولید می‌کند، پس صفات ABCH کلید کاندید می‌باشد.

قانون دوم ارسسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

گزینه اول نادرست است. زیرا، عضو کلید کاندید ABCH را ندارد.

گزینه دوم درست است. زیرا، عضو کلید کاندید ABCH، به تهایی کلید کاندید است.

گزینه سوم نادرست است. زیرا، عضو کلید کاندید ABCH را ندارد.

گزینه چهارم نادرست است. زیرا، عضو کلید کاندید ABCH را ندارد.

۱۷- گزینه (۳) صحیح است.

طبق تعریف، وابستگی یک عنصر غیرکلیدی به یک عنصر غیرکلیدی دیگر وابستگی انتقالی، وابستگی یک عنصر غیرکلیدی به کلید کاندید، وابستگی تابعی کامل و وابستگی یک عنصر غیرکلیدی به بخشی از کلید کاندید، وابستگی بخشی خوانده می‌شود.

گزینه اول نادرست است. زیرا، $C \rightarrow AB$ ، وابستگی تابعی کامل است.

گزینه دوم نادرست است. زیرا، $A \rightarrow DEH$ ، وابستگی بخشی است.

گزینه سوم درست است. زیرا $EH \rightarrow D$ ، وابستگی انتقالی است.

گزینه چهارم نادرست است. زیرا، $D \rightarrow B$ ، وابستگی بخشی است.

۱۸- گزینه (۴) صحیح است.

مثال: جدول R(a,b,c,d) را با وابستگی‌های زیر در نظر بگیرید:

$$a \rightarrow b$$

$$b \rightarrow c$$

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

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

۲- عضو زائد نداشته باشد.

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

قانون اول ارسسطو

روشن اول:

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

روشن دوم:

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

با توجه به وابستگی‌های مطرح شده برای رابطه $R(a, b, c, d)$ داریم:

$$a \mathrel{b} \mathrel{c} d - b \mathrel{c} a \mathrel{d}$$

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

$$\{ad\}^+ = \{a, d, b, c\}$$

براساس بستار فوق، صفات ad ، همه ستون‌ها را تولید می‌کند، پس صفات ad کلید کاندید می‌باشد.

قانون دوم ارسسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روشن اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود. یک کلید کاندید وجود دارد که کلید اصلی هم خواهد بود و بدون کلید فرعی. در یک قاعده کلی، صفات سمت راست وابستگی‌های غیربدیهی، جزء کلید کاندید (اصلی) نخواهند بود.

حال گزاره مطرح شده را در نظر بگیرید:

$$\forall X \subseteq U \quad \forall A \in U : \underline{X \rightarrow A \in F^+ \wedge A \notin X}$$

شرط گزاره

گزاره فوق بدین معنی است که X زیر مجموعه‌ای از صفات U است و A صفتی از صفات U است.

حال اگر مطابق شرط $\rightarrow A \in F^+$ بدین معنی که زیر مجموعه صفاتی، صفت A را بدهد، بدین معنی است که A به X وابستگی تابعی دارد. شرط بعدی یعنی $A \notin X$ بیان می‌کند که A متعلق به X نباشد، چون در این صورت وابستگی بدیهی ایجاد می‌کند. مانند $b \rightarrow ab$ که صفت b را می‌دهد و b زیر مجموعه ab هست. بنابراین منظور از عبارت $\rightarrow A \in F^+ \wedge A \notin X$ وابستگی‌هایی است که بدیهی نباشد. مانند وابستگی‌های غیربدیهی زیر برای $R(a, b, c, d)$ به عبارت دیگر غیربدیهی باشند.

$$\begin{array}{c} X \\ \hat{a} \rightarrow \hat{b} \end{array}$$

$$\begin{array}{c} X \\ \hat{b} \rightarrow \hat{c} \end{array}$$

همان‌طور که پیش از این بیان شد، کلید کاندید جدول فوق ad می‌باشد، که صورت سوال، کلید کاندید را K نامگذاری کرده است.

$$ad \equiv k$$

پُر واضح است که صفات b و c جزء کلید کاندید (اصلی) نیستند. زیرا در یک قاعده کلی صفات سمت راست وابستگی‌های غیربدیهی، جزء کلید کاندید (اصلی) نخواهند بود.

گزینه اول نادرست است. زیرا، $X \subset K$ (به معنی هر زیر مجموعه‌ای از صفات جدول R ، با نام X زیر مجموعه کلید کاندید است) همواره برقرار نیست.

مثال: $\begin{array}{c} b \quad ad \\ \hat{X} \subset K \end{array}$ ، یعنی b زیر مجموعه ad نیست.

گزینه دوم نادرست است، زیرا $X \not\subset K$ (به معنی هر زیر مجموعه‌ای از صفات جدول R با نام X ، زیر مجموعه کلید کاندید نیست) همواره برقرار نیست.

مثال: $\frac{a}{X} \not\in \bar{K}$, یعنی a زیر مجموعه ad نیست.

گزینه سوم نادرست است. زیرا، اگر جدولی مطابق آنچه گفته شده، فقط یک کلید کاندید داشته باشد، امکان ندارد، صفات سمت راست وابستگی‌ها جزء کلید کاندید (اصلی) جدول باشد. بنابراین گزینه چهارم درست است.

- گزینه (۲) صحیح است.

مجموعه‌ی پوششی A (یعنی مجموعه‌ی تمام صفت‌هایی که با A قابل دسترسی‌اند) عبارت است از:

$$\{A\}^* = \{A, B, C, D\}$$

مالحظه می‌شود که فقط صفت E در مجموعه‌ی فوق نیست پس A و E به همراه هم کلیدند.

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

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

۲- عضو زائد نداشته باشد.

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

قانون اول ارسسطو

روش اول:

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

روش دوم:

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

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

$$A \rightarrow B$$

$$AB \rightarrow CD$$

$$D \rightarrow ABC$$

$$ABC \not\subseteq DE - ABCD = E$$

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

$$\{E\}^+ = \{E\}$$

براساس بستار فوق، صفت E فقط ستون E را تولید می‌کند، پس صفت E فقط عضو کلید کاندید می‌باشد و کلید کاندید نمی‌باشد.

قانون سوم ارسسطو

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 دقیق و برسی، پُر واضح است که اگر صفت A در کثار عضو کلید کاندید E قرار بگیرد، همکاری صفات (E و A) می‌تواند، همه ستون‌ها را تولید کند، بنابراین صفات AE کلید کاندید جدول R خواهد بود.
بستار صفات AE به صورت زیر است:

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

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

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

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

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

بستار صفات (D, E) به صورت زیر است:

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

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

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

گزینه اول نادرست است زیرا، عضو کلید کاندید E را ندارد.

گزینه دوم درست است. زیرا، صفات AE کلید کاندید جدول R است.

گزینه سوم نادرست است. زیرا، عضو کلید کاندید E را ندارد.

گزینه چهارم نادرست است. زیرا، عضو کلید کاندید E را ندارد.

۲۰- گزینه (۱) صحیح است.

با توجه به نمودار FD داده شده، وابستگی‌های زیر تعریف شده‌اند:

- ✗ (1) $M \rightarrow N$
- ✗ (2) $M \rightarrow O$
- ✗ (3) $M \rightarrow P$
- ✓ (4) $N \rightarrow O$
- ✓ (5) $N \rightarrow P$
- ✓ (6) $P \rightarrow N$
- ✗ (7) $MN \rightarrow O$

می‌توان دید که وابستگی (7) از روی وابستگی‌های (2) و (4) قابل نتیجه‌گیری است، پس می‌توان وابستگی (7) را کنار گذاشت. بنابراین گزینه دوم نادرست است. هم‌چنین وابستگی (2) از روی وابستگی‌های (1) و (4) به دست می‌آید، پس وابستگی (2) را نیز کنار می‌گذاریم. بنابراین گزینه چهارم نادرست است.

وابستگی (3) نیز از روی وابستگی‌های (1) و (5) به دست می‌آید، پس وابستگی (3) را نیز کنار می‌گذاریم. بنابراین گزینه سوم نادرست است.

وابستگی (1) نیز از روی وابستگی‌های (3) و (6) به دست می‌آید. پس وابستگی (1) را نیز کنار می‌گذاریم. **توجه:** وقتی یک وابستگی، کار گذاشته می‌شود، به معنی عدم استفاده از آن نیست، زیرا توسط وابستگی‌های دیگر مجدداً قابل تولید است. بنابراین می‌توان از آن استفاده نمود.

وابستگی‌های باقیمانده (یعنی مجموعه‌ی حداقل FD‌ها) به صورت زیر خواهد بود.

$$(4) \quad N \rightarrow O$$

$$(5) \quad N \rightarrow P$$

$$(6) \quad P \rightarrow N$$

۲۱ - گزینه (۲) صحیح است.

با توجه به سه وابستگی تابعی اول (از سمت چپ)، مشاهده می‌شود که از بین صفت‌های A و B و C با داشتن هر کدام می‌توان مقدار دوتای دیگر را تعیین کرد. هم‌چنین با توجه به دو وابستگی تابعی آخر، مشاهده می‌شود که با داشتن هر یک از صفات D و E می‌توان دیگری را به دست آورد. برای ساختن یک ابرکلید باید از بین سه صفت A و B و C حداقل یکی و از بین دو صفت D و E نیز حداقل یکی را انتخاب کرد. برای انتخاب حداقل یک صفت از بین سه صفت A و B و C یا باید یکی از آن‌ها را انتخاب کنیم یا دو تا یا سه تا را. تعداد حالات ممکن برای انتخاب حداقل یک صفت از بین سه صفت A و B و C برابر است با:

$$\binom{3}{1} + \binom{3}{2} + \binom{3}{3} = 7$$

به طور مشابه، تعداد حالات ممکن برای انتخاب حداقل یک صفت از بین دو صفت D و E برابر است با:

$$\binom{2}{1} + \binom{2}{2} = 3$$

بنابراین:

برای ساختن یک کلید کاندید، باید از بین سه صفت A و B و C دقیقاً یک صفت و از بین دو صفت D و E نیز دقیقاً یک صفت انتخاب کنیم. تعداد حالات ممکن برای خواهد بود با:

$$\binom{3}{1} + \binom{2}{1} = 3 \times 2 = 6$$

۲۲ - گزینه (۱) صحیح است.

از آنجا که حرف A در سمت راست هیچ یک از وابستگی‌های تابعی وجود ندارد. بنابراین حرف A حتماً جزئی از کلید کاندید است. بنابراین {A,...} کلید کاندید است.

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

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

۲- عضو زائد نداشته باشد.

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

قانون اول ارسسطو

روش اول:

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

روش دوم:

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

$$\begin{aligned} A \rightarrow B \quad C \\ CD \rightarrow E \\ \square \rightarrow D \\ ABCDE - BCDE = A \end{aligned}$$

بنابر رابطه فوق صفت A حتماً باید عضو کلید کاندید باشد. بنابراین گزاره I درست است. بستار صفت A به صورت زیر است:

$$\{A\}^+ = \{A, B, C, \dots\}$$

توجه: چند نقطه (...) برای این است که نمی‌دانیم سمت چپ وابستگی \rightarrow \square چیست. اما آن‌چه که پر واضح است، این است که کلید کاندید تحت هیچ شرایطی شامل صفات B و C نخواهد بود، زیرا توسط صفت A، به عنوان عضو کلید کاندید قابل تولید است. بنابراین گزاره II درست و گزاره III نادرست است. در نهایت مشخص می‌شود که گزینه اول درست است.

۲۳- گزینه (۴) صحیح است.

توجه: گزینه چهارم، همان قانون اول و دوم ارسسطو می‌باشد.

جدول (R = {A → B, B → C, D → E, H → A}) با وابستگی‌های {A → B, B → C, D → E, H → A} را در نظر بگیرید:
مطابق رابطه گزینه چهارم داریم:

$$V - \bigcup_{i=1}^n (y_i - x_i)$$

$$y_1 - x_1 = A \rightarrow B \Rightarrow B - A = B$$

$$y_2 - x_2 = B \rightarrow C \Rightarrow C - B = C$$

$$y_3 - x_3 = D \rightarrow E \Rightarrow E - D = E$$

$$y_4 - x_4 = H \rightarrow A \Rightarrow A - H = A$$

بنابراین عبارت $\bigcup_{i=1}^n (y_i - x_i)$ در یک مجموعه وابستگی تابعی، صفاتی که سمت راست وابستگی‌های

تابعی هستند و جزء دترمینانها (سمت چپ رابطه) نیستند را می‌دهد. سپس عبارت $(\sum_{i=1}^n y_i - V) -$ این صفات را از مجموعه صفات حذف می‌کند. صفاتی که از مجموعه V باقی می‌مانند، عضو کلید کاندید رابطه خواهند بود.

$$(A, B, \emptyset, D, E, H) - (ABCE) = DH$$

بنابر رابطه فوق صفات DH باید عضو کلید کاندید باشد.

بستار صفات DH به صورت زیر است:

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

براساس بستار فوق، صفات DH ، همه ستون‌ها را تولید می‌کند، پس صفات DH کلید کاندید می‌باشد.

قانون دوم ارسسطو

هرگاه عضو کلید کاندید حاصل از تفاضل قانون اول (روش اول یا دوم) همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

۲۴- گزینه (۱) صحیح است.

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

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

۲- عضو زائد نداشته باشد.

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

قانون اول ارسسطو

روش اول:

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

روش دوم:

$$R - \left(\sum_{i=1}^n y_i - (\text{راست}) \right) = \text{عضو کلید کاندید}$$

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

$ABD \rightarrow EG$

$C \rightarrow DG$

$E \rightarrow FG$

$I \rightarrow H$

$H \rightarrow J$

$AB \rightarrow C$

$G \rightarrow F$

$$ABCDEF \not\rightarrow HIJ - CDEFGHJ = ABI$$

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

$$\{ABI\}^+ = \{A, B, I, C, H, D, G, E, F, J\}$$

براساس بستار فوق، صفات ABI، همه ستون‌ها را تولید می‌کند، پس صفات ABI کلید کاندید می‌باشد.

قانون دوم ارسسطو

هرگاه عضو کلید کاندید حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کاندید جدول خواهد بود.

گزینه اول درست است. زیرا، عضو کلید کاندید ABI، به تنها یک کلید کاندید است.

گزینه دوم نادرست است. زیرا، عضو کلید کاندید ABI را ندارد.

گزینه سوم نادرست است. زیرا، عضو کلید کاندید ABI را ندارد.

گزینه چهارم نادرست است. زیرا، عضو زائد D را دارد، هر چند عضو کلید کاندید ABI را دارد.

صفات ABI، همه ستون‌ها را به تنها یک با استفاده از وابستگی‌های مطرح شده تولید می‌کنند، بنابراین صفت D، عضو زائد است. مجموعه صفات ABDI در واقع ابرکلید است.

۲۵- گزینه (۳) صحیح است.

با کمی بررسی می‌توان دید که رابطه‌ی داده شده، با توجه به وابستگی‌های تابعی آن، دو کلید کاندید دارد: ABC و DC (توجه کنید که کلید کاندید باید کمینه باشد یعنی زیر مجموعه‌ی محضی از صفات آن، کلید نباشد) برای به دست آوردن ابرکلیدها، باید به هر یک از دو کلید کاندید فوق، صفر یا بیشتر از صفت‌های دیگر را اضافه کرد. بنابراین باید به کلید کاندید DC، صفر، یک، دو یا سه مورد از صفت‌های دیگر (E,B,A) اضافه کرد. برای این کار 8 حالت وجود دارد:

$$\binom{3}{0} + \binom{3}{1} + \binom{3}{2} + \binom{3}{3} = 2^3 = 8$$

به طور مشابه، باید به کلید کاندید ABC، صفر، یک یا دو صفت از بین صفت‌های دیگر (E,D) اضافه کرد. برای این کار 4 حالت وجود دارد:

$$\binom{2}{0} + \binom{2}{1} + \binom{2}{2} = 2^2 = 4$$

ترکیب اول: هیچ یا ترکیبی از صفات باقی مانده + کلید کاندید DC = ابرکلید

DC +	A B E	ترکیبات صفات باقی مانده	ابرکلیدها
	0 0 0	تهی	DC
	0 0 1	E	D C E
	0 1 0	B	D C B
	0 1 1	B E	D C B E
	1 0 0	A	D C A
	1 0 1	A E	D C A E
	1 1 0	A B	D C A B
	1 1 1	A B E	D C A B E

توجه: واضح است که ۸ ابرکلید، ایجاد می‌گردد. (2^3)

ترکیب دوم: هیچ یا ترکیبی از صفات باقیمانده + کلید کاندید $= ABC$ = ابرکلید

$A B C +$

E D	ترکیبات صفات باقیمانده	ابرکلیدها
0 0	نهی	A B C
0 1	D	A B C D
1 0	E	A B C E
1 1	E D	A B C E D

توجه: واضح است که ۴ ابرکلید، ایجاد می‌گردد. (2^2)

بنابراین حاصل جمع ابرکلیدها ۱۲ عدد خواهد بود، که از این مجموعه حاصل، ابرکلیدهای ABCDE و ABCD در دو مجموعه تکراری هستند، بنابراین با کنار گذاشتن ابرکلیدهای تکراری، در نهایت ۱۰ ابرکلید خواهیم داشت. به همین سادگی.

۲۶- گزینه (۱) صحیح است.

راه حل اول: کلاسیک

گزینه اول: برقرار نیست، چون از روی E و C نمی‌توان به D رسید.

گزینه دوم درست است، چون $C \rightarrow E$ ، پس واضح است که $AE \rightarrow C$.

گزینه سوم درست است.

چون صفت سمت راست \rightarrow زیر مجموعه صفات سمت چپ \rightarrow است. (وابستگی بدیهی)

گزینه چهارم درست است، چون از دو وابستگی اول و دوم به دست می‌آید. $D \rightarrow AC$.

راه حل دوم: وابستگی‌های مطرح شده به صورت زیر است:

$$A \rightarrow B$$

$$BC \rightarrow D$$

$$E \rightarrow C$$

گزینه اول درست است، زیرا:

{سؤال: آیا می‌توان از صفات CE با استفاده از وابستگی‌های فوق به صفت D رسید؟}

$$\{CE \rightarrow D\}^+ = \{CE\}^+ = \{C, E\}$$

پاسخ: خیر نمی‌توان رسید، زیرا، صفات CE، C و E را می‌دهد و کار تمام می‌شود.

گزینه دوم نادرست است، زیرا:

{سؤال: آیا می‌توان از صفات AE با استفاده از وابستگی‌های فوق به صفت C رسید؟}

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

پاسخ: بله می‌توان رسید، زیرا، صفات AE، A و E را می‌دهد. A، B را می‌دهد و E، C را می‌دهد و BC را

داریم پس D را هم می‌دهد.

گزینه سوم نادرست است. زیرا، وابستگی بدیهی است و واضح است که حتماً BC را تولید می‌کند.

گزینه چهارم نادرست است. زیرا:

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

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

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

- گزینه (۲) صحیح است.

رابطه $C \rightarrow A$ را می‌توان از روی روابط $B \rightarrow A$ و $C \rightarrow B$ به دست آورد، پس افزونه است.

- گزینه (۴) صحیح است.

از روی وابستگی تابعی $\rightarrow AB$ می‌توان نتیجه گرفت، $C \rightarrow A$ و $C \rightarrow B$.

از $C \rightarrow B$ و $C \rightarrow D$ می‌توان نتیجه گرفت: $C \rightarrow D$.

از $C \rightarrow A$ و $C \rightarrow D$ نتیجه می‌گیریم: $C \rightarrow AD$.

از $C \rightarrow A$ و $AD \rightarrow E$ نتیجه می‌گیریم: $C \rightarrow E$.

- گزینه (۱) صحیح است.

ابتدا وابستگی‌های تابعی را بازنویسی می‌کنیم. طوری که سمت راست هر وابستگی فقط یک صفت باشد:

- | | | | |
|-----------------------|-----------------------|------------------------|-----------------------|
| (1) $A \rightarrow B$ | (2) $A \rightarrow C$ | (3) $A \rightarrow D$ | (4) $A \rightarrow K$ |
| (5) $K \rightarrow C$ | (6) $B \rightarrow D$ | (7) $BC \rightarrow D$ | |

دیده می‌شود که با داشتن وابستگی (6) می‌توان وابستگی (7) را نتیجه گرفت پس وابستگی (7) اضافی است و باید حذف شود.

هم‌چنین از وابستگی‌های (1) و (6) می‌توان وابستگی (3) را نتیجه گرفت پس وابستگی (3) اضافی است و حذف می‌شود. به طور مشابه، از وابستگی‌های (4) و (5) می‌توان وابستگی (2) را نتیجه گرفت پس وابستگی (2) اضافی است و باید حذف شود. بنابراین، وابستگی‌های زیر باقی می‌مانند:

- | | | | |
|-----------------------|-----------------------|-----------------------|-----------------------|
| (1) $A \rightarrow B$ | (4) $A \rightarrow K$ | (5) $K \rightarrow C$ | (6) $B \rightarrow D$ |
|-----------------------|-----------------------|-----------------------|-----------------------|

- گزینه (۴) صحیح است.

{سؤال: آیا می‌توان از صفت A با استفاده از وابستگی‌های زیر به صفت C رسید؟}

$$\{A \rightarrow C\}^+ =$$

$$A \rightarrow B$$

$$AB \rightarrow C$$

$$\{A\}^+ = \{A, B, \boxed{C}\}$$

پاسخ: بله، می‌توان رسید، A خود را می‌دهد، A ، B را می‌دهد و AB را داریم، پس C را می‌دهد. بنابراین گزینه اول درست است.

{سؤال: آیا می‌توان از صفات AC با استفاده از وابستگی زیر به صفت D رسید؟}

$$\{AC \rightarrow D\}^+ =$$

$$A \rightarrow B$$

$$BC \rightarrow D$$

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

پاسخ: بله، می‌توان رسید، A و C خود را می‌دهند، A، B را می‌دهد و BC را داریم، پس D را می‌دهد.
بنابراین گزینه دوم درست است.

{سؤال: آیا می‌توان از صفات AC با استفاده از وابستگی‌های زیر به صفات BD رسید؟}

$$\{AC \rightarrow BD\}^+ =$$

$$A \rightarrow B$$

$$AC \rightarrow D$$

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

پاسخ: بله، می‌توان رسید، A و C خود را می‌دهند، A، B را می‌دهد و AC را داریم، پس D را می‌دهد.
بنابراین گزینه سوم درست است.

{سؤال: آیا می‌توان از صفت A با استفاده از وابستگی‌های زیر به صفت D رسید؟}

$$\{A \rightarrow D\}^+ =$$

$$AB \rightarrow BC$$

$$C \rightarrow D$$

$$\{A\}^+ = \{A\}$$

پاسخ: خیر، نمی‌توان رسید، A خودش را می‌دهد و کار تمام می‌شود.

۳۱- گزینه (۲) صحیح است.

به راحتی می‌توان مجموعه پوششی هر یک از مجموعه صفات داده شده را محاسبه کرد.

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

$$\{AD\}^* = \{A, D, E\}, \{AB\}^* = \{A, B, C, D, E\}, \{CD\}^* = \{A, B, C, D, E\}$$

مشاهده می‌شود که تمام صفت‌ها از C، AB و AC قابل دسترسی‌اند. پس این مجموعه از صفت‌ها ابرکلید هستند. CD ابرکلید است، اما کلید کاندید نیست زیرا صفت D عضو زائد است، چون صفت C به تنهایی همه ستون‌ها را تولید می‌کند. هم‌چنین AC ابرکلید است، اما کلید کاندید نیست زیرا صفت A عضو زائد است چون صفت C به تنهایی همه ستون‌ها را تولید می‌کند. C و AB ابرکلید هستند و کلید کاندید هم هستند چون عضو زائد ندارند.

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

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

۲- عضو زائد نداشته باشد.

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

قانون اول ارسسطو

روش اول:

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

روش دوم:

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

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

$AB \rightarrow C$

$C \rightarrow A$

$C \rightarrow BD$

$D \rightarrow E$

$ABCDE - ABCDE = \text{تهی}$

قانون چهارم ارسسطو

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

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

براساس بستار فوق، صفت C ، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفت C ، کلید کاندید می‌باشد.

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

براساس بستار فوق، صفات AB ، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفات AB ، کلید کاندید می‌باشد.

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

- ۳۲- گزینه (۱) صحیح است.

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

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

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

قانون اول ارسسطو

روش اول:

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

روش دوم:

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

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

$$A \rightarrow BC$$

$$CD \rightarrow E$$

$$B \rightarrow D$$

$$E \rightarrow A$$

$$ABC\bar{C}DE - ABCDE = \text{تهی}$$

قانون چهارم ارسسطو

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 ستون های B و D را تولید می کند، پس صفت B ، کلید کاندید نمی باشد. بنابراین گزینه اول پاسخ سوال خواهد بود.

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

براساس بستار فوق، صفت A ، همه ستون ها را بدون عضو زائد تولید می کند، پس صفت A ، کلید کاندید می باشد.

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

براساس بستار فوق، صفت E ، همه ستون ها را بدون عضو زائد تولید می کند، پس صفت E ، کلید کاندید می باشد.

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

براساس بستار فوق، صفات BC ، همه ستون ها را بدون عضو زائد تولید می کند، پس صفات BC ، کلید کاندید می باشد.

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

براساس بستار فوق، صفات CD ، همه ستون ها را بدون عضو زائد تولید می کند، پس صفات CD ، کلید کاندید می باشد.

توجه: همان طور که مشاهده می شود، مطابق قانون چهارم ارسسطو، هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید فرق وجود ندارد.

- ۳۳- گزینه (۲) صحیح است.

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

۱- ابر کلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه ها را تولید کند.

۲- عضو زائد نداشته باشد.

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

قانون اول ارسسطو

روش اول:

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

روش دوم:

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

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

$$(A, B) \rightarrow C$$

$$B \rightarrow D$$

$$(D, E) \rightarrow C$$

$$ABC\bar{C}DE - CD = ABE$$

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

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

براساس بستار فوق، صفات ABE، همه ستون‌ها را تولید می‌کند، پس صفات ABE کلید کاندید می‌باشد.

قانون دوم ارسسطو

هرگاه عضو کلید کاندید حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

گزینه اول نادرست است. زیرا، عضو کلید کاندید ABE را ندارد.

گزینه دوم درست است. زیرا، عضو کلید کاندید ABE، به تنها یک کلید کاندید است.

گزینه سوم نادرست است. زیرا، عضو زائد C را دارد، هر چند عضو کلید کاندید ABE را دارد. صفات ABE همه ستون‌ها را به تنها یک با استفاده از وابستگی‌های مطرح شده تولید می‌کند، بنابراین صفت C، عضو زائد است. مجموعه صفات ABCE در واقع ابرکلید است.

گزینه چهارم نادرست است. زیرا، عضو زائد D را دارد، هر چند عضو کلید کاندید ABE را دارد. صفات ABE همه ستون‌ها را به تنها یک با استفاده از وابستگی‌های مطرح شده تولید می‌کند، بنابراین صفت D، عضو زائد است. مجموعه صفات ABDE در واقع ابرکلید است.

۳۴- گزینه (۴) صحیح است.

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

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

۲- عضو زائد نداشته باشد.

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

قانون اول ارسسطو

روش اول:

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

روش دوم:

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

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

$$AB \rightarrow CD$$

$$D \rightarrow B$$

$$ABC\bar{D} - BCD = A$$

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

$$\{A\}^+ = \{A\}$$

براساس بستار فوق، صفت A فقط ستون A را تولید می کند، پس صفت A فقط عضو کلید کاندید می باشد و کلید کاندید نمی باشد.

قانون سوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، برخی از ستون ها را تولید کند، بدین معنی است که، جدول مورد نظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهد داشت.

با کمی دقت و بررسی، پُر واضح است که اگر صفت B در کنار عضو کلید کاندید A قرار بگیرد، همکاری صفات (B و A) می تواند، همه ستون ها را تولید کند، بنابراین صفات AB کلید کاندید جدول R خواهد بود. بستار صفات AB به صورت زیر است:

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

براساس بستار فوق، صفات AB ، همه ستون ها را تولید می کنند، پس صفات AB ، کلید کاندید است.

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

همچنین از آن جا که $\rightarrow B$ ، پس می توان ترکیب دو خصیصه (A, D) را هم کلید کاندید دیگری برای این جدول تلقی کرد.

چون وقتی (A, B) کلید کاندید است و همه ستون ها را تولید می کند، پس (A, D) هم کلید کاندید است و همه ستون ها را تولید می کند، زیرا در نهایت طبق وابستگی $D \rightarrow B$ ، صفت D ، صفت B را می دهد و (A, B) به (A, D) تبدیل می گردد.

بستار صفات AD به صورت زیر است:

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

براساس بستار فوق، صفات (A, D) ، همه ستون ها را تولید می کنند، پس صفات (A, D) کلید کاندید است.

توجه: دقت کنید که هیچ گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده، در مجموع،

دارای دو کلید کاندید است که مطابق قانون سوم ارسسطو، عضو کلید کاندید A در بین هر دو کلید کاندید به طور مشترک قرار دارد.

گزینه اول نادرست است زیرا، عضو کلید کاندید A را ندارد.

گزینه دوم نادرست است. زیرا، صفات AD هم کلید کاندید جدول R است.

گزینه سوم نادرست است. زیرا، صفات AB هم کلید کاندید جدول R است.

گزینه چهارم درست است. زیرا، صفات AB و AD هر دو کلیدهای کاندید جدول R هستند.

۳۵- گزینه (۲) صحیح است.

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

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

۲- عضو زائد نداشته باشد.

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

قانون اول ارسسطو

روش اول:

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

روش دوم:

$$\bigcup_{i=1}^n [x_i, y_i] \cap (p_j) = \text{عضو کلید کاندید}$$

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:

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

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

قانون سوم ارسسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روشن 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 نخواهد داشت.

با کمی دقیق و بررسی، پُر واضح است که اگر صفت A در کنار عضو کلید کاندید F قرار بگیرد، همکاری صفات (F و A) می‌تواند، همه ستون‌ها را تولید کند، بنابراین صفات AF کلید کاندید جدول R نخواهد 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: دقیق 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 $A \rightarrow D$ ، 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: دقیق کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد.

همچنین از آن‌جا که $D \rightarrow B$ ، 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 $D \rightarrow B$ ، صفت B، صفت D را می‌دهد و (D, F) به (B, 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: دقیق کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده، در مجموع، دارای سه کلید کاندید است که مطابق قانون سوم ارسطو، عضو کلید کاندید F در بین هر سه کلید کاندید به طور مشترک قرار دارد.

۳۶- گزینه (۲) صحیح است.

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

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

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدینهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\bigcup_{i=1}^n [x_i \cdot (چپ) - y_i \cdot (راست)] = \text{عضو کلید کاندید}$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E)$ داریم:

$A \rightarrow BC$

$CD \rightarrow E$

$B \rightarrow D$

$E \rightarrow A$

$ABC\bar{D}\bar{E} - ABCDE = \text{نهی}$

قانون چهارم ارسسطو

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 ماقبی 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.

۳۷- گزینه (۳) صحیح است.

مجموعه‌ی پوششی A (یعنی مجموعه‌ی تمام صفت‌هایی که با A قابل دسترسی‌اند) عبارت است از:

$$\{A\}^* = \{A, B, C, D\}$$

مالحظه می‌شود که فقط صفت E در مجموعه‌ی فوق نیست پس A و E به همراه هم کلیدند.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\bigcup_{i=1}^n [y_i - (راست) x_i] = \text{عضو کلید کاندید}$$

با توجه به وابستگی‌های مطرح شده برای رابطه R(A,B,C,D,E) داریم:

$$A \rightarrow B$$

$$AB \rightarrow CD$$

$$D \rightarrow ABC$$

$$ABC \not\rightarrow E - ABCD = E$$

بنابر رابطه فوق صفت E حتماً باید عضو کلید کاندید باشد. بستار صفت E به صورت زیر است:

$$\{E\}^+ = \{E\}$$

براساس بستار فوق، صفت E، فقط ستون E را تولید می‌کند، پس صفت E فقط عضو کلید کاندید می‌باشد و کلید کاندید نمی‌باشد.

قانون سوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، برخی از ستون‌ها را تولید کند، بدین معنی است که، جدول موردنظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت.

براساس بستار فوق، صفت E به عنوان عضو کلید کاندید همه ستون‌ها را تولید نمی‌کند، بنابراین مطابق قانون سوم ارسطو باید صفاتی در کنار صفت E قرار گیرد تا کلید کاندید تولید گردد. این صفات کناری از صفات باقی مانده به غیر از صفات E انتخاب می‌گردند، صفات باقی مانده عبارتنداز D و C و B و A به از این مجموعه صفات C و B هم کنار می‌گذاریم، زیرا کمکی در تولید صفات دیگر نمی‌کنند. چون در سمت چپ هیچ یک از وابستگی‌های تابعی به تنها بی نیامده‌اند. بنابراین با ترکیب صفات A یا D با صفت E کلیدهای کاندید تولید می‌گردد.

$$\{AE\}^+ = \{A, E, B, C, D\}$$

$$\{DE\}^+ = \{D, E, A, B, C\}$$

پس ترکیبات صفات AE و DE کلیدهای کاندید جدول R هستند.

به بیان دیگر با کمی دقیق و برسی، پُر واضح است که اگر صفت A در کنار عضو کلید کاندید E قرار بگیرد، همکاری صفات (E و A) می‌تواند، همه ستون‌ها را تولید کند، بنابراین صفات AE کلید کاندید جدول R نخواهد بود.

بستار صفات AE به صورت زیر است:

$$\{AE\}^+ = \{A, E, B, C, D\}$$

براساس بستار فوق، صفات AE، همه ستون‌ها را تولید می‌کنند، پس صفات AE، کلید کاندید است.

توجه: دقیق کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد.

همچنین از آنجا که $A \rightarrow D$ ، پس می‌توان ترکیب دو خصیصه (D, E) را هم کلید کاندید دیگری برای این جدول تلقی کرد.

چون وقتی (A, E) کلید کاندید است و همه ستون‌ها را تولید می‌کند، پس (D, E) هم کلید کاندید است و همه ستون‌ها را تولید می‌کند، زیرا در نهایت طبق وابستگی $D \rightarrow A$ ، صفت D، صفت A را می‌دهد و (D, E) به (A, E) تبدیل می‌گردد.

بستار صفات (D, E) به صورت زیر است:

$$\{DE\}^+ = \{D, E, A, B, C\}$$

براساس بستار فوق، صفات (D, E)، همه ستون‌ها را تولید می‌کنند، پس صفات (D, E) کلید کاندید است.

توجه: دقیق کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده، در مجموع، دارای دو کلید کاندید است که مطابق قانون سوم ارسطو، عضو کلید کاندید E در بین هر دو کلید کاندید به طور مشترک قرار دارد.

گزینه اول نادرست است زیرا، عضو کلید کاندید E را ندارد.

گزینه دوم نادرست است. زیرا، عضو کلید کاندید E را ندارد.

گزینه سوم درست است. زیرا، صفات AE کلید کاندید جدول R است.

گزینه چهارم نادرست است. زیرا، کلیدهای کاندید فقط و فقط AE و DE هستند. به بیان دیگر بستار BE $\{B, E\}^+ = \{B, E\}$ همه ستون‌ها را تولید نمی‌کند:

۳۸- گزینه (۲) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = \bigcup_{i=1}^n [y_i - (چپ) x_i] \quad (\text{راست})$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E)$ داریم:

$$AB \rightarrow C$$

$$CD \rightarrow E$$

$$DE \rightarrow B$$

$$ABCDE - BCE = AD$$

بنابر رابطه فوق صفات AD حتماً باید عضو کلید کاندید باشد. بستار صفات AD به صورت زیر است:

$$\{AD\}^+ = \{A, D\}$$

براساس بستار فوق، صفات AD، فقط ستون‌های A و D را تولید می‌کند، پس صفات AD فقط عضو کلید کاندید می‌باشد و کلید کاندید نمی‌باشد.

قانون سوم ارسسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، برخی از ستون‌ها را تولید کند، بدین معنی است که، جدول موردنظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت.

براساس بستار فوق، صفات AD به عنوان عضو کلید کاندید همه ستون‌ها را تولید نمی‌کند، بنابراین مطابق قانون سوم ارسسطو باید صفاتی در کنار صفات AD قرار گیرد تا کلید کاندید تولید گردد. این صفات کناری از صفات باقی مانده به غیر از صفات AD انتخاب می‌گردند، صفات باقی مانده عبارتنداز E و C و B که از این مجموعه هیچ صفت دیگری را کنار نمی‌گذاریم، زیرا در تولید صفات دیگر کمک می‌کنند. بنابراین با ترکیب صفات E و C و B با صفات AD کلیدهای کاندید تولید می‌گردد.

$$\{ADE\}^+ = \{A, D, E, B, C\}$$

$$\{ADC\}^+ = \{A, D, C, E, B\}$$

$$\{ADB\}^+ = \{A, D, B, C, E\}$$

پس ترکیبات صفات ADE و ADC و ADB کلیدهای کاندید جدول R هستند.
به بیان دیگر با کمی دقت و بررسی، پُر واضح است که اگر صفت E در کنار عضو کلید کاندید AD قرار بگیرد، همکاری صفات (E و D) می‌تواند، همه ستون‌ها را تولید کند، بنابراین صفات ADE کلید کاندید جدول R خواهد بود.
بستار صفات ADE به صورت زیر است:

$$\{ADE\}^+ = \{A, D, E, B, C\}$$

براساس بستار فوق، صفات ADE، همه ستون‌ها را تولید می‌کنند، پس صفات ADE، کلید کاندید است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد.

در ادامه به شکل بازگشتی جهت کشف ماتقی کلیدهای کاندید داریم:

همچنین از آن‌جا که \rightarrow CD، پس می‌توان ترکیب سه خصیصه (A,D,C) را هم کلید کاندید دیگری برای این جدول تلقی کرد.

چون وقتی (A,D,E) کلید کاندید است و همه ستون‌ها را تولید می‌کند، پس (A,D,C) هم کلید کاندید است و همه ستون‌ها را تولید می‌کند، زیرا در نهایت طبق وابستگی CD \rightarrow E، صفات CD، صفت E را می‌دهد و (A,D,C) به (A,D,E) تبدیل می‌گردد.

بستار صفات (A,D,C) به صورت زیر است:

$$\{ADC\}^+ = \{A, D, C, E, B\}$$

براساس بستار فوق، صفات (A,D,C)، همه ستون‌ها را تولید می‌کنند، پس صفات (A,D,C) کلید کاندید است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد.

همچنین از آن‌جا که \rightarrow AB، پس می‌توان ترکیب سه خصیصه (A,D,B) را هم کلید کاندید دیگری برای این جدول تلقی کرد.

چون وقتی (A,D,C) کلید کاندید است و همه ستون‌ها را تولید می‌کند، پس (A,D,B) هم کلید کاندید است و همه ستون‌ها را تولید می‌کند، زیرا در نهایت طبق وابستگی C \rightarrow AB، صفات AB، صفت C را می‌دهد و (A,D,C) به (A,D,B) تبدیل می‌گردد.

بستار صفات (A,D,B) به صورت زیر است:

$$\{ADB\}^+ = \{A, D, B, C, E\}$$

براساس بستار فوق، صفات (A,D,B)، همه ستون‌ها را تولید می‌کنند، پس صفات (A,D,B) کلید کاندید است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده، در مجموع، دارای سه کلید کاندید است که مطابق قانون سوم ارسطو، عضو کلید کاندید AD در بین هر سه کلید کاندید به طور مشترک قرار دارد.

ترکیب اول: هیچ یا ترکیبی از صفات باقی مانده + کلید کاندید $= \text{ابرکلید}$

$A D E +$

B C	ترکیبات صفات باقی مانده	ابرکلیدها
0 0	تهی	A D E
0 1	C	A D E C
1 0	B	A D E B
1 1	B C	A D E B C

توجه: واضح است که 4 ابرکلید، ایجاد می‌گردد. (2^2)

ترکیب دوم: هیچ یا ترکیبی از صفات باقیمانده + کلید کاندید $= \text{ابرکلید}$

$A D C +$

B E	ترکیبات صفات باقی مانده	ابرکلیدها
0 0	تهی	A D C
0 1	D	A D C E
1 0	E	A D C B
1 1	E D	A D C B E

توجه: واضح است که 4 ابرکلید، ایجاد می‌گردد. (2^2)

ترکیب سوم: هیچ یا ترکیبی از صفات باقیمانده + کلید کاندید $= \text{ابرکلید}$

$A D B +$

C E	ترکیبات صفات باقی مانده	ابرکلیدها
0 0	تهی	A D B
0 1	E	A D B E
1 0	C	A D B C
1 1	C E	A D B C E

توجه: واضح است که 4 ابرکلید، ایجاد می‌گردد. (2^2)

بنابراین حاصل جمع ابرکلیدها 12 عدد خواهد بود، که از این مجموعه حاصل، ابرکلیدهای ADCE و ADCBE و ADBE و ADBC و DBCE در سه مجموعه تکراری هستند، بنابراین با کنار گذاشتن ابرکلیدهای تکراری، در نهایت 7 ابرکلید خواهیم داشت. به همین سادگی.

۳۹- گزینه (۳) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسسطو

روشن اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\bigcup_{i=1}^n [x_i - y_i \text{ (راست)} - R \text{ (چپ)}] = \text{عضو کلید کاندید}$$

توجه: عبارت $[x_i - y_i \text{ (راست)} - R \text{ (چپ)}]$ به طور مستقل بر روی تک تک وابستگی ها انجام می گردد.

$$A \rightarrow BC \Rightarrow BC - A = BC$$

مثال:

توجه: استفاده از روش اول مستلزم گام ابتدایی حذف وابستگی های بدیهی است، اگر در حذف وابستگی های بدیهی دچار خطأ می شویم، از روش دوم استفاده نمایید.

با توجه به وابستگی های مطرح شده برای رابطه $R(A,B,C,D,E)$ داریم:

$$AB \rightarrow C$$

$$CD \rightarrow E$$

$$ABCDE - CE = ABD$$

بنابر رابطه فوق صفات ABD حتماً باید عضو کلید کاندید باشد. بستار صفات ABD به صورت زیر است:

$$\{ABD\}^+ = \{A, B, D, C, E\}$$

براساس بستار فوق، صفات ABD، همه ستون ها را تولید می کند، پس صفات ABD کلید کاندید می باشد.

قانون دوم ارسسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم) همه ستون ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

ساخت ابر کلید: هیچ یا ترکیبی از صفات باقیمانده + کلید کاندید $=$ ابر کلید

A B D +	C E	ترکیبات صفات باقیمانده	ابر کلیدها
0 0		تهی	A B D
0 1	E		A B D E
1 0	C		A B D C
1 1	C E		A B D C E

توجه: واضح است که 4 ابر کلید، ایجاد می گردد. (2^2)

بنابراین حاصل جمع ابر کلیدها 4 عدد خواهد بود، ابر کلیدهای **ABDC**، **ABDE**، **ABD** و **ABDCE**، به همین سادگی.

۴۰- گزینه (۴) صحیح است.

برای درک سوال و راه حل آن به مثال زیر دقت نمایید:

مثال: با توجه به رابطه $R(A,B,C,D,E)$ و مجموعه وابستگی های 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 $\{A\}$ 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 ارسسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\bigcup_{i=1}^n [x_i \cdot y_i] - R(\text{راست}) = \text{عضو کلید کاندید}$$

توجه: عبارت $[x_i \cdot y_i] - R(\text{راست})$ 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 خطأ می‌شوید، از روش دوم استفاده نمایید.

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E)$ داریم:

$A \rightarrow BC$

$CD \rightarrow E$

$B \rightarrow D$

$E \rightarrow A$

$ABC \cdot DE - ABCDE =$ تهی

قانون چهارم ارسسطو

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 ماقعی 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 +	B C D E F	ترکیبات صفات باقی مانده	ابرکلیدها
	0 0 0 0 0	نهی	A
	0 0 0 0 1	F	A F
	0 0 0 1 0	E	A E
	0 0 0 1 1	E F	A E F

	1 1 1 1 1	B C D E F	A B C D E F

توجه: واضح است که 32 ابرکلید، ایجاد می‌گردد. (2^5)

ترکیب دوم: هیچ یا ترکیبی از صفات باقیمانده + کلید کاندید BC = ابرکلید

B C +	A D E F	ترکیبات صفات باقی مانده	ابرکلیدها
	0 0 0 0	نهی	B C
	0 0 0 1	F	B C F
	0 0 1 0	E	B C E
	0 0 1 1	E F	B C E F

	1 1 1 1	A D E F	B C A D E F

توجه: واضح است که ۱۶ ابرکلید، ایجاد می‌گردد. (2^4)
بنابراین حاصل جمع ابرکلیدها $= 48 = 32 + 16$ 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.

A B C +	D E F	ترکیبات صفات باقی مانده	ابرکلیدها
0 0 0		نهی	A B C
0 0 1	F		A B C F
0 1 0	E		A B C E
0 1 1	E F		A B C E F
1 0 0	D		A B C D
1 0 1	D F		A B C D F
1 1 0	D E		A B C D E
1 1 1	D E F		A B C D E F

توجه: واضح است که ۸ ابرکلید تکراری، میان دو مجموعه ترکیب اول و دوم وجود دارد. (2^3)

توجه: بنابراین با کنار گذاشتن ابرکلیدهای تکراری، در نهایت ۴۰ ابرکلید خواهیم داشت. به همین سادگی

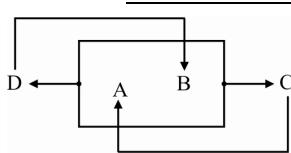
تست‌های فصل نهم: نرمال‌سازی

- ۱- کدام یک از گزاره‌های زیر در رابطه با «عمده‌ترین اهداف نرمال‌سازی رابطه‌ها» نادرست است؟
(مهندسی کامپیوترا- دولتی ۷۶)

- ۱) کاهش بعضی انواع افزونگی (Redundancy).
- ۲) کاهش سربار (Overhead) سیستم در پاسخگویی در عمل بازیابی.
- ۳) تسهیل در اعمال برخی محدودیت‌های جامعیتی (Integrity Constraints).
- ۴) اجتناب از بعضی از انواع آنومالی در عملیات روی پایگاه.

- ۲- کدام گزینه از اهداف کلی فرآیند نرمال‌سازی (normalization) نیست؟
(مهندسی کامپیوترا- دولتی ۷۵)

- ۱) تسهیل اعمال بعضی از قواعد جامعیتی (integrity rules).
- ۲) تسهیل پیاده‌سازی دید کاربر (user view).
- ۳) حذف بعضی از انواع قواعد وابستگی تابعی (functional dependency).
- ۴) کاهش بعضی از انواع افزونگی (redundancy).



- ۳- اگر رابطه $R(A,B,C,D)$ با توابع وابستگی شکل مقابل مفروض باشد، کدام یک از گزینه‌ها نادرست است؟
(مهندسی کامپیوترا- دولتی ۸۱۳)

- ۱) (A,B) و (C,D) کلید کاندید رابطه R می‌باشند.
- ۲) رابطه R نرمال فرم BCNF می‌باشد، اما نرمال فرم 3NF نمی‌باشد.
- ۳) رابطه R نرمال فرم 2NF و نیز نرمال فرم 3NF می‌باشد.
- ۴) رابطه R نرمال فرم 3NF می‌باشد اما نرمال فرم BCNF نمی‌باشد.

(مهندسی IT- دولتی ۸۱۳)

- ۴- کدام گزینه صحیح است؟

- ۱) جدولی که BCNF باشد، حتماً 3NF است.
- ۲) اگر جدولی 3NF باشد، وابستگی انتقالی ندارد.
- ۳) اگر جدولی وابستگی انتقالی نداشته باشد، 3NF است.
- ۴) ۱ و ۲.

- ۵- جدول $R(X,Y,Z)$ به دو جدول $R_1(X,Y)$ و $R_2(X,Z)$ تجزیه و نرمال شده است. کدام گزینه از شرط‌های صحت این عمل می‌باشد؟ ($X \rightarrow R_1$ یعنی X کلید R_1 است).
(مهندسی کامپیوترا- دولتی ۸۱۴)

$$\begin{array}{ll} X \rightarrow R_2 & X \rightarrow R_1 \\ X \rightarrow R_2 \text{ و } X \rightarrow R_1 & X \rightarrow R_2 \text{ و } X \rightarrow R_1 \end{array}$$

- ۶- رابطه XYZ به شکل رویه‌رو را در نظر بگیرید. این رابطه در چه سطحی از نرمال بودن است؟
(مهندسی IT- دولتی ۸۵)

- | X | Y | Z |
|----|----|----|
| x1 | y1 | z2 |
| x1 | y2 | z1 |
| x2 | y1 | z1 |
| x1 | y1 | z1 |
- ۱) 1NF
 - ۲) 2NF
 - ۳) 3NF
 - ۴) BCNF

-۷ با نرمال سازی رابطه R تا سطح 3NF تعداد جداول حاصل چند عدد می شود؟ مجموعه وابستگی ها در FD (مهندسی IT-آزاد ۱۸۵)

نمایش داده شده است.

$$R(A, \underline{B, C, D}, E, F(G, H), I)$$

$$FD = \{BC \rightarrow A, B \rightarrow G, BCD \rightarrow I, G \rightarrow H, E \rightarrow I, BCD \rightarrow E, BCD \rightarrow A\}$$

5 (۴)

3 (۳)

6 (۲)

4 (۱)

-۸ اگر در رابطه R کلید اصلی BCD باشد، آن گاه کدام وابستگی تابعی زیر شرط 2NF (مهندسی IT-آزاد ۱۸۶)

بودن رابطه R را نقض می کند؟

$$A \rightarrow EF$$

$$E \rightarrow F$$

$$DCB \rightarrow A$$

$$BC \rightarrow E$$

-۹ رابطه $R(A, B, C, D, E, F, G)$ با کلید اصلی BCD و با مجموعه وابستگی تابعی زیر مفروض است:

$$FD = \{BCD \rightarrow E, BCD \rightarrow B, BCD \rightarrow F, BD \rightarrow A, F \rightarrow G, E \rightarrow F, C \rightarrow E\}$$

در صورت تبدیل آن به 3NF چند رابطه ایجاد می شود؟

5 (۴)

3 (۳)

4 (۲)

6 (۱)

-۱۰ اگر وابستگی چند مقداری به شکل $Y \rightarrow X$ بقرار باشد، و همچنین داشته باشیم $V \subseteq W \subseteq Y$ آن گاه طبق قانون افزایشی (augmentation) می توان درستی کدام گزینه زیر را نتیجه گرفت؟ (مهندسی IT-دولتی ۱۸۷)

$$WX \rightarrow \rightarrow WY \quad (4) \quad WX \rightarrow \rightarrow VY \quad (3) \quad VX \rightarrow VY \quad (2) \quad VX \rightarrow \rightarrow WY \quad (1)$$

-۱۱ در یک پایگاه داده محیط آموزشی دانشگاه برای جدول دانشجو سه صفت شماره دانشجویی، کد پستی و شهر محل سکونت با وابستگی های تابعی زیر وجود دارند:

{شهر محل سکونت \rightarrow کد پستی محل سکونت \rightarrow شماره دانشجویی}

با توجه به این که رابطه شهرها و کد پستی ثابت است، سطح نرمال مناسب برای تجزیه این جدول چیست؟

BCNF (۴)

3NF (۳)

2NF (۲)

1NF (۱)

-۱۲ مجموعه وابستگی های تابعی (FD) رویه را در نظر بگیرید (هر حرف یک صفت است). برای این مجموعه پایگاه داده رابطه ای با سه جدول AIE , EFG و $ABCDH$ طراحی شده است. این پایگاه داده در چه سطح نرمالی قرار دارد؟ (مهندسی IT-دولتی ۱۸۷)

$$A \rightarrow IEG$$

1NF (۱)

$$AB \rightarrow IECD$$

2NF (۲)

$$ABE \rightarrow CDFGH$$

3NF (۳)

$$AC \rightarrow IEC$$

BCNF (۴)

$$CDE \rightarrow DE$$

$$E \rightarrow FG$$

$$F \rightarrow E$$

-۱۳ در طراحی بانک اطلاعاتی رابطه ای، کدام یک از معیارها مهم تر است؟ (مهندسی IT-دولتی ۱۸۸)

(۱) رعایت شرط Nonloss Join

(۲) داشتن کلید اصلی در رابطه ها

(۳) حفظ وابستگی های تابعی

(مهندسی IT- دولتی ۸۸)

۱۴- کدام مورد از خصوصیات شکل نرمال اول نیست؟

۱) هیچ رابطه تو در تو (Nested) وجود ندارد.

۲) هیچ صفت مرکب (Composite Attribute) وجود ندارد.

۳) هیچ صفت چند مقداری (Multivalued Attribute) وجود ندارد.

۴) هیچ وابستگی تابعی تراگذاری (Transitive Functional Dependency) وجود ندارد.

(مهندسی کامپیووتر- آزاد ۸۸)

۱۵- کدام گزینه در خصوص نرمال سازی نادرست است؟

۱) نرمال سازی باعث افزایش سرعت پاسخ‌گویی به پرس‌جو می‌شود.

۲) نرمال سازی تا سطح ۳NF اجباری است.

۳) نرمال سازی ممکن است باعث افزایش افزونگی شود.

۴) حل مشکل مقادیر پوچ (NULL) یکی از اهداف نرمال سازی است.

(مهندسی IT- آزاد ۸۸)

۱۶- با توجه به عبارات زیر، کدام گزینه صحیح است؟

۱) هر رابطه‌ای ۲NF باشد، حتماً ۱NF نیز هست.

۲) هر رابطه‌ای ۱NF باشد، حتماً ۲NF نیز هست.

۳) هر رابطه‌ای BCNF باشد، حتماً ۳NF نیز هست.

۴) هر رابطه‌ای ۳NF باشد، حتماً BCNF نیز هست.

۴) ۳ و ۱ و ۲ و ۳

۱) ۲ و ۳

۱۷- در رابطه $R(A,B,C,D,E)$ با مجموعه وابستگی‌های S کدام یک از گزینه‌های زیر درست است؟

(مهندسی کامپیووتر- آزاد ۸۸)

 $S = \{A \rightarrow C, B \rightarrow D, AB \rightarrow E\}$ ۱) رابطه R نرمال ۲NF می‌باشد.۲) رابطه R نرمال ۳NF می‌باشد.۳) رابطه R نرمال ۳NF و BCNF می‌باشد.۴) اگر رابطه R به روابط $R_1(A,B,E)$ و $R_2(B,D)$ و $R_3(A,C)$ تفکیک شود، هر سه رابطه حاصل نرمال ۳NF و BCNF می‌باشند.

۱۸- اگر جدولی در فرم نرمال BCNF باشد، ممکن است در کدام فرم نرمال دیگر نباشد؟

(مهندسی کامپیووتر- آزاد ۸۸)

۴) ۴NF

۳) ۳NF

۲) ۲NF

۱) ۱NF

(مهندسی IT- آزاد ۸۸)

۱۹- کدام گزینه از اهداف کلی فرآیند نرمال سازی نیست؟

۱) تسهیل در پیاده‌سازی دید کاربر

۲) تسهیل اعمال برخی از قوانین جامعیتی

۴) کاهش برخی از انواع افزونگی

۳) حذف برخی از مقادیر NULL

۲۰- اگر در جدول ST وابستگی تابعی $\rightarrow S \# \rightarrow Code$ را داشته باشیم، آنگاه این جدول در کدام فرم نرمال

(مهندسی IT- دولتی ۸۹)

۱) نمی‌باشد؟ $ST(S\#, T\#, Date, Time, Code)$

۴) BCNF

۳) CCNF

۲) ۲NF

۱) ۳NF

۲۱- اگر تمامی اطلاعات جداول را در یک جدول بریزیم با چه مشکلاتی روبرو می‌شویم؟
 (مهندسی کامپیوتر- آزاد ۸۹ گروه الف)

- | | |
|-----------------------|---------------------------------------|
| (Anomaly) (۲) بی‌نظمی | (Data Redundancy) (۱) افزونگی داده‌ها |
| (۴) همه موارد | (NULL Values) (۳) مقادیر نهی |

۲۲- کدام گزینه از عمدۀ ترین اهداف نرم‌السازی نیست?
 (مهندسی IT- آزاد ۸۹)
 ۱) کاهش بعضی از انواع افزونگی‌ها
 ۲) حذف بعضی از انواع وابستگی تابعی
 ۳) تسهیل اعمال بعضی از قواعد جامعیتی
 ۴) کاهش سربار (Over head) سیستم در پاسخگویی در عمل بازیابی

۲۳- رابطه A(X,Y,Z,P,Q,R) با وابستگی‌های تابعی (Functional Dependency) شکل زیر را در نظر بگیرید.
 (مهندسی کامپیوتر- دولتی ۹۰)
 این رابطه در کدام سطح نرم‌الاست؟



۲۴- کدام یک از گزینه‌های زیر نادرست است?
 (۱) در SQL، عملگرهای \neq all و not in هم ارز هستند.

(۲) رابطه R با وابستگی تابعی F در BCNF است اگر و فقط اگر رابطه R با وابستگی‌های تابعی F^+ (بسنار F) در BCNF باشد.

(۳) در رابطه‌ای با اسکیمای R(A,B,C,D) به طور منطقی می‌توان از وجود وابستگی چند مقداری $A \rightarrow\rightarrow BC$ وابستگی‌های منطقی $B \rightarrow\rightarrow A$ و $C \rightarrow\rightarrow A$ را نتیجه گیری کرد.

(۴) دو مجموعه وابستگی‌های زیر هم ارز (معادل) نیستند:

$$F_1 = \{AB \rightarrow E, A \rightarrow CD, A \rightarrow E, E \rightarrow D, D \rightarrow A\}$$

$$F_2 = \{AB \rightarrow D, E \rightarrow AD, A \rightarrow CD, D \rightarrow AE, EC \rightarrow B\}$$

۲۵- فرض کنید رابطه R با مجموعه وابستگی‌های تابعی F داده شده است. صفت $A \in R$ را ناهنجار (abnormal) می‌گوییم اگر و فقط اگر $x \rightarrow R \notin F^+ \wedge A \in (x^+ - x)$ ($\exists x \subseteq R : x \rightarrow R \notin F^+ \wedge A \in (x^+ - x)$) که نمایانگر بسنار (closure) مجموعه صفات x تحت وابستگی‌های تابعی F می‌باشد. کدام یک از گزینه‌های زیر نادرست است؟
 (مهندسی IT- دولتی ۹۰)

(۱) اگر رابطه R در 3NF باشد، آن‌گاه هیچ صفت غیرکلیدی در رابطه R ناخواهد بود.

(۲) اگر رابطه R در BCNF باشد، آن‌گاه هیچ صفت ناهنجاری در R وجود نخواهد داشت.

(۳) اگر هیچ صفت ناهنجاری در R وجود نداشته باشد، آن‌گاه رابطه R در BCNF خواهد بود.

(۴) هیچ کدام

-۲۶- رابطه $R(A,B,C,D,E)$ و وابستگی های تابعی زیر را در نظر بگیرید:

$$A \rightarrow B, BC \rightarrow D, E \rightarrow C$$

اگر نگاشت R بر روی $S(B,C,D,E)$ را در نظر بگیریم، کدام یک از وابستگی های تابعی در S برقرار است
(مهندسی IT- دولتی ۹۰)

$$B \rightarrow E \quad (4)$$

$$BC \rightarrow D \quad (3)$$

و شرط $BCNF$ را برای S نقض نمی کند؟

$$E \rightarrow C \quad (2)$$

$$BE \rightarrow D \quad (1)$$

-۲۷- جدول $R(A,B,C,D,E,F,H,I)$ در طی فرآیند نرمال سازی به جدول $R_1(A,B)$ ، $R_2(B,C,F)$ ، $R_3(D,E)$ ، $R_4(C,F,J,I)$ شکسته شده است.

(مهندسی کامپیوuter-آزاد ۹۰ گروه الف)

(۱) این شکستن قطعاً غلط است.

(۲) این کار به شرطی درست است که A کلید اصلی جدول R باشد.

(۳) این کار به شرطی درست است که AB کلید اصلی جدول R باشد.

(۴) بدون داشتن وابستگی های تابعی نمی توان اظهار نظری کرد.

-۲۸- کدام گزینه در خصوص نرمال سازی غلط است؟ (مهندسی کامپیوuter-آزاد ۹۰ گروه الف)

(۱) نرمال سازی عموماً به افزایش سرعت پرس و جوها منجر می شود.

(۲) وجود مقادیر پوج (NULL Values) می تواند فرآیند نرمال سازی را با مشکل مواجه نماید.

(۳) نرمال سازی می تواند باعث کاهش بی نظمی (Anomaly) در پایگاه داده شود.

(۴) نرمال سازی تا سطح 3NF عموماً حجم پایگاه داده را کاهش می دهد.

-۲۹- جدول $R(A,B,C,D,E,F)$ با وابستگی های تابعی $F = \{AB \rightarrow CF, C \rightarrow DE, E \rightarrow F\}$ در طی فرآیند نرمال سازی به سه جدول R_1 و R_2 و R_3 شکسته شده است.

(مهندسی کامپیوuter-آزاد ۹۰ گروه الف)

(۱) در صورتی که وابستگی های تابعی جداول به ترتیب $\{AB \rightarrow C\}$ ، $F_1 = \{C \rightarrow DE\}$ و $F_2 = \{E \rightarrow F\}$ باشد، این شکستن درست است.

(۲) در صورتی که وابستگی های تابعی جداول به ترتیب $\{AB \rightarrow CF\}$ ، $F_1 = \{C \rightarrow D\}$ و $F_2 = \{E \rightarrow F\}$ باشد، این شکستن درست است.

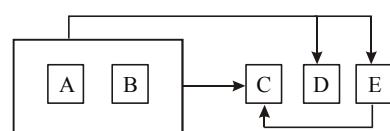
(۳) این شکستن قطعاً غلط است و جدول R باید به چهار جدول شکسته شد.

(۴) این شکستن قطعاً غلط است و جدول R باید به دو جدول شکسته شود.

-۳۰- با توجه به نمودار وابستگی های تابعی داده شده رابطه $R(A,B,C,D,E)$ در چه سطحی از نرمال قرار

(مهندسی کامپیوuter-آزاد ۹۰ گروه ب)

دارد؟



(۲) سطح دوم نرمال (2NF)

(۴) سطح نرمال (1NF)

(۱) سطح سوم نرمال (3NF)

(۳) سطح نرمال BCNF

-۳۱- رابطه $R(A,B,C,D)$ و وابستگی های تابعی زیر مفروضند:

(مهندسی کامپیوuter-آزاد ۹۰ گروه ب)

$$A \rightarrow B, B \rightarrow C, BC \rightarrow A$$

این رابطه:

- (۱) ۳NF هست ولی BCNF نیست.
 (۲) ۳NF نیست.
 (۳) BCNF هست ولی ۴NF نیست.
 (۴) ۴NF هست.

۳۲- یک طراحی منطقی خوب و مناسب یک پایگاه داده باید کدام یک از خصوصیات زیر را داشته باشد؟
 (مهندسی کامپیووتر- آزاد ۹۰ گروه ب)

- (۱) نمایش جامع و واضحی از محیط دنیای واقعی داشته باشد و در برگیرنده تمام خصوصیات مورد نیاز کاربران، به گونه‌ای که تولید برنامه‌های کاربردی به راحتی امکان‌پذیر باشد.
 (۲) تمام محدودیت‌های (قواعد) جامعیتی، که قابل اعمال در هر مرحله از طراحی جامعیتی باشند، در طراحی منظور شده است.
 (۳) کمترین میزان اختلاط اطلاعاتی را داشته باشد و کمترین مقدار هیچ مقدار (NULL) در پایگاه داده ایجاد گردد.
 (۴) همه موارد

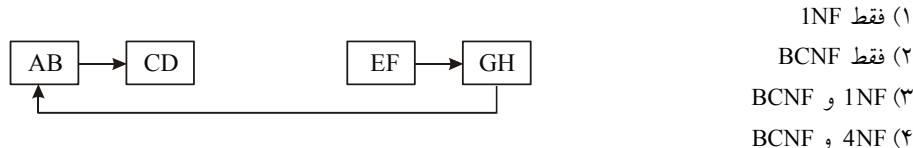
۳۳- کدام گزینه در خصوص نرمال‌سازی صحیح است؟
 (مهندسی IT- آزاد ۹۰)
 (۱) نرمال‌سازی نکردن می‌تواند به افزایش سرعت برخی پرس‌وجوها منجر شود.
 (۲) نرمال‌سازی نکردن می‌تواند به کاهش سرعت برخی پرس‌وجوها منجر شود.
 (۳) نرمال‌سازی تا سطح ۳NF عموماً باعث کاهش افزونگی می‌شود.
 (۴) هر سه مورد

۳۴- جدول R با وابستگی‌های تابعی $F = \{AB \rightarrow CD, C \rightarrow DE, E \rightarrow F\}$ در طی فرآیند نرمال‌سازی به سه جدول R_1 و R_2 و R_3 شکسته شده است.
 (مهندسی IT- آزاد ۹۰)
 (۱) در صورتی که وابستگی‌های تابعی جداول به ترتیب $F_1 = \{AB \rightarrow C\}$ و $F_2 = \{C \rightarrow DE\}$ باشد، این شکستن درست است.
 (۲) در صورتی که وابستگی‌های تابعی جداول به ترتیب $F_1 = \{AB \rightarrow CD\}$ و $F_2 = \{C \rightarrow E\}$ باشد این شکستن درست است.
 (۳) این شکستن قطعاً غلط است و جدول R باید به دو جدول شکسته شود.
 (۴) این شکستن قطعاً غلط است و جدول R باید به چهار جدول شکسته شود.

۳۵- جدول R در طی فرآیند نرمال‌سازی به جداول $R_1(A, B, C)$ و $R_2(B, F)$ و $R_3(D, E)$ شکسته شده است.
 (مهندسی IT- آزاد ۹۰)

- (۱) این کار به شرطی درست است که A کلید اصلی جدول R باشد.
 (۲) این کار به شرطی درست است که AB کلید اصلی جدول R باشد.
 (۳) این شکستن قطعاً غلط است.
 (۴) بدون داشتن وابستگی‌های تابعی نمی‌توان اظهارنظر کرد.

۳۶- اگر کوتاه‌ترین کلید را کلید اصلی بگیریم آن‌گاه بانک اطلاعات زیر در کدام فرم نرمال است?
 (مهندسی کامپیووتر- دولتی ۹۱)



۳۷- رابطه $R(A, B, C, D, E, G)$ با وابستگی های تابعی F را در نظر بگیرید:

$$F = \{A \rightarrow BC, BE \rightarrow G, G \rightarrow CD, AD \rightarrow BG, AE \rightarrow G\}$$

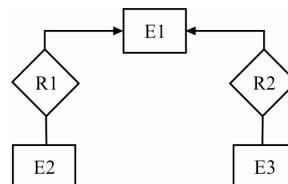
(مهندسی IT- دولتی ۹۱)

کدام یک از گزینه های زیر صحیح نیست؟

- (۱) صفت B در وابستگی تابعی $AD \rightarrow BG$ اضافی (extraneous) است.
- (۲) تنها کلید کاندید رابطه R , $\{AE\}$ است.
- (۳) وابستگی تابعی $AE \rightarrow G$ تکراری (redundant) است.
- (۴) رابطه R در نرمال فرم دوم است.

۳۸- فرض کنید که بین سه مجموعه موجودیت $E1, E2$ و $E3$ مجموعه رابطه هایی به شکل زیر برقرار باشد»

(مهندسی IT- دولتی ۹۱)



ولی در زمان طراحی دیاگرام ER، طراح به اشتباه یک رابطه سه گانه میان این سه مجموعه موجودیت در نظر گرفته است. حال پس از تبدیل دیاگرام ER به مدل رابطه ای، رابطه ای (جدولی) برای رابطه سه گانه در نظر گرفته شده است. آیا می توان با اعمال نرمال فرم ها به طراحی درست رسید؟

(۱) بله، با استفاده از 4NF می توان رسید، البته به شرط داشتن وابستگی های تابعی و چند مقداری موجود.

(۲) بله، با استفاده از BCNF می توان رسید، البته به شرط داشتن وابستگی های تابعی موجود.

(۳) خیر، به علت اشتباه اولیه در طراحی دیاگرام ER، بازیابی رابطه های درست دو تابی میسر نیست.

(۴) بله، با استفاده از 3NF می توان رسید، البته به شرط داشتن وابستگی های تابعی موجود.

۳۹- فرض کنید رابطه $R(A1, A2, A3, \dots, An)$ را داشته باشیم. فرض کنید X, Y, Z و W زیر مجموعه هایی از R باشند. دو گزاره زیر را در نظر بگیرید:

گزاره اول: اگر داشته باشیم $Y \rightarrow X$ و $Z \subseteq W$ آنگاه می توان به طور منطقی نتیجه گرفت $. WX \rightarrow YZ$

گزاره دوم: اگر $Y \rightarrow X$ وجود داشته باشد و زیر مجموعه W دارای خواص زیر باشد $W \cap Y = \emptyset$ و $Z \subseteq Y$ و $W \rightarrow Z$ آنگاه می توان به طور منطقی نتیجه گرفت $. X \rightarrow Z$

(۱) فقط گزاره اول صحیح است.

(۲) فقط گزاره دوم صحیح است.

(۳) هر دو گزاره صحیح هستند.

-۴۰- رابطه $R = (A, B, C, D, E)$ و وابستگی‌های تابعی (FD) را به صورت ذیل در نظر بگیرید:

$$FD = \{A \rightarrow D, A \rightarrow E, D \rightarrow E, (A, B) \rightarrow D, (A, B) \rightarrow E, (A, B) \rightarrow C\}$$

(مهندسی کامپیووتر-آزاد ۹۱)

کدام تجزیه بهتر و حداقل سطح نرمالیتی آن تجزیه کدام است؟

$$2NF, R''(A, D, E), R'(A, B, C) \quad (۱)$$

$$3NF, R'''(D, E), R''(A, D), R'(A, B, C) \quad (۲)$$

$$3NF, R'''(A, E), R''(A, D), R'(A, B, C) \quad (۳)$$

$$3NF, R''(A, B, E), R''(A, B, D), R'(A, B, C) \quad (۴)$$

-۴۱- رابطه $R(A, B, C, D)$ و مجموعه وابستگی‌های تابعی (F) را در نظر بگیرید.

$$F = \{A \rightarrow B, B \rightarrow C, C \rightarrow D\}$$

(مهندسی کامپیووتر-دولتی ۹۱)

کدام یک از این تجزیه‌ها بدون گمشدگی (Lossless) نیست؟

یادآوری: رابطه R و مجموعه وابستگی‌های تابعی F روی R را در نظر بگیرید. تجزیه R به دو رابطه با

مجموعه صفات X و Y یک تجزیه «بدون گمشدگی» با توجه به F است. اگر به ازای هر نمونه r از R که F

مطابقت دارد، $\Pi_X(r) \bowtie \Pi_Y(r) = r$ ، به عبارت دیگر رابطه اولیه را می‌توان از روی رابطه‌های جدید

به دست آورد.

$$R_1(A, B), R_2(B, C), R_3(C, D) \quad (۱)$$

$$R_1(A, B), R_2(A, C), R_3(A, D) \quad (۲)$$

$$R_1(A, D), R_2(B, D), R_3(C, D) \quad (۳)$$

(۴) هیچ کدام- به عبارت دیگر همه این تجزیه‌ها بدون گمشدگی هستند.

-۴۲- کدام یک از گزینه‌های زیر برای رابطه Booking و وابستگی‌های تابعی داده شده آن، که اطلاعات رزرو

اتاق در یک هتل را نگهداری می‌کنند، غلط است؟

(مهندسی IT- دولتی ۹۱)

Booking (guestID, guestName, creditCard, roomNo, roomCat, from, to)

$F = \{$

- 1) $guestID \rightarrow guestName, creditCard$
- 2) $roomNo \rightarrow roomCat$
- 3) $roomNo, from \rightarrow guestID, to$
- 4) $roomNo, to \rightarrow guestID, from$

$\}$

(۱) این رابطه در هیچ یک از نرمال فرم‌های 3NF, 2NF و BCNF نیست.

(۲) دو وابستگی تابعی سوم و چهارم باعث نقض شدن 3NF در رابطه نمی‌شود.

(۳) اگر این رابطه به سه رابطه $R1$ (guestID, roomNo, from, to) و $R2$ (roomNo, roomCat) و $R3$ (creditCard, roomNo, from

اتلاف (lossless decomposition) است.

(۴) اگر این رابطه به سه رابطه $R1$ (guestID, guestName, creditCard) و $R2$ (roomNo, roomCat) و $R3$ (guestID, roomNo, form, to)

تجزیه شود، هر سه رابطه در 3NF هستند ولی در BCNF نیستند.

(مهندسی IT- دولتی ۹۱)

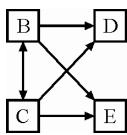
-۴۳- رابطه زیر و مجموعه وابستگی‌های تابعی آن را در نظر بگیرید:

$$R(A, B, C, D, E, F, G, H) \quad F = \{D \rightarrow EF, F \rightarrow C, DG \rightarrow AB\}$$

کدام یک از گزینه‌های زیر غلط است؟

- ۲) تنها کلید کاندید DGH است.
 ۴) رابطه فوق در فرم نرمال دوم است.

۴۴- نمودار وابستگی تابعی رابطه‌ای به صورت ذیل است. B و C کلید کاندید هستند. این رابطه در کدام سطح نرمالیتی است؟
 (مهندسی کامپیووتر-آزاد ۹۲)



- ۱) 3NF
 ۲) BCNF
 ۳) 1NF
 ۴) 2NF

۴۵- رابطه دانشجو (کد دانشجو، نام، نام خانوادگی، درس، استاد) در کدام سطح نرمال قرار دارد؟ با فرض آن که یک استاد می‌تواند چندین درس را ارائه نماید و یک درس می‌تواند توسط چندین استاد ارائه شود؟
 (مهندسی کامپیووتر-آزاد ۹۲)

- ۱) 4NF
 ۲) 1NF
 ۳) 3NF
 ۴) 2NF

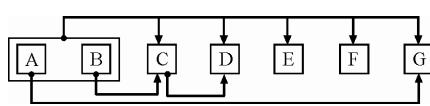
۴۶- با داشتن رابطه R(A,B,C,D,E,F,G,H) و مجموعه وابستگی‌های تابعی F = {BE → GH, G → FA, D → C, F → B} کدام گزینه نادرست است؟
 (مهندسی کامپیووتر-دولتی ۹۳)
 (۱) رابطه R نرمال 3NF نیست.
 (۲) کلیدی وجود ندارد که شامل A نباشد.
 (۳) کلیدی وجود ندارد که شامل D نباشد.

۴۷- رابطه (R(A, B, C, D, E) و مجموعه وابستگی‌های تابعی F را در نظر بگیرید.

$$F = \{BC \rightarrow A, A \rightarrow D, D \rightarrow C, D \rightarrow E\}$$

کدام یک از وابستگی‌های تابعی زیر BCNF را نقض نمی‌کند؟
 (مهندسی کامپیووتر-دولتی ۹۴)
 (۱) D → E
 (۲) A → D
 (۳) D → C
 (۴) BC → A

۴۸- رابطه (R(A, B, C, D, E, F, G) که نمودار وابستگی‌های تابعی آن به صورت زیر می‌باشد مفروض است. اگر بخواهد تا سطح 2NF نرمال سازی شود، جداول حاصل کدامند؟
 (مهندسی IT - دولتی ۹۴)



- ۱) BC, AG, ABDEF
 ۲) BC, CD, AG, ABEF
 ۳) BCD, AG, ABEF
 ۴) ABCDG, ABEF

۴۹- کدام یک از موارد زیر صحیح است؟
 (مهندسی کامپیووتر-دولتی ۹۵)
 (۱) رابطه‌ای نرمال است که هیچ یک از صفات ساده‌اش چندمقداری نباشند.
 (۲) کلید کاندید رابطه می‌تواند کاهش پذیر باشد.
 (۳) تاپل‌های یک رابطه نظم دارند.
 (۴) رابطه تاپل تکراری ندارد.

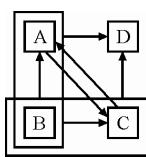
۵۰- فرض کنید در رابطه (AB) کلید اصلی و (BC) کلید کاندید و روابط وابستگی تابعی به صورت زیر باشد. کدام گزینه در خصوص این رابطه صحیح است؟
 (مهندسی IT-دولتی ۹۵)

۱) رابطه R در فرم نرمال ۳NF است ولی BCNF نیست.

۲) رابطه R در فرم نرمال BCNF است.

۳) رابطه R در فرم نرمال ۳NF نیست.

۴) رابطه R در فرم نرمال ۲NF نیست.



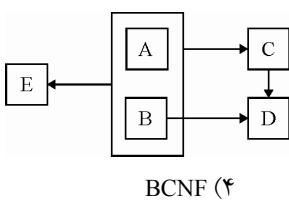
۵۱- اگر $R(A,B,C,D,E)$ یک رابطه در فرم نرمال BCNF و ABC تنها کلید R باشد، کدام یک از وابستگی‌های تابعی زیر حتماً برای R برقرار است؟
 (مهندسی IT-دولتی ۹۵)

$ABCD \rightarrow E$ (۴)

$ABD \rightarrow E$ (۳)

$ACE \rightarrow D$ (۲)

$BCE \rightarrow D$ (۱)



۵۲- اگر مجموعه وابستگی‌های تابعی (FDهای) یک رابطه به صورت شکل مقابل و کلید اصلی آن (A,B) باشد، این رابطه در چه سطح نرمالی است؟
 (مهندسی IT-دولتی ۹۶)

BCNF (۴)

3NF (۳)

2NF (۲)

1NF (۱)

۵۳- اگر در رابطه R ، تعداد ابرکلیدها با تعداد کلیدهای کاندید برابر باشد، آنگاه کدام مورد در خصوص رابطه R نادرست است؟
 (مهندسی IT-دولتی ۹۷)

۱) تنها یک خصیصه دارد.

۲) در فرم نرمال ۳NF است.

۳) همه خصیصه‌های R NOT NULL هستند.

۴) تعداد وابستگی‌های تابعی نابدیهی R ، صفر هستند.

۵۴- رابطه و وابستگی‌های تابعی زیر را در نظر بگیرید.

$R(X,Y,Z)$

۱) $Y \rightarrow Z$ ۲) $XZ \rightarrow Y$ ۳) $X \rightarrow Z$

(مهندسی کامپیووت-دولتی ۹۸)

با توجه به رابطه فوق کدام عبارت نادرست است؟

۱) کلید کاندید است.

۲) این رابطه در فرم نرمال BCNF نیست.

۳) وابستگی سوم، اضافه و قابل حذف است.

۴) X در وابستگی دوم، مشخصه اضافه و قابل حذف است.

پاسخ تست‌های فصل نهم: نرم‌السازی

۱- گزینه (۲) صحیح است.

جدول studclg با مقادیر آن به شکل زیر را در نظر بگیرید:

S#	Sname	Clg#	Clgname
8621	Ali	12	Computer
8442	Reza	12	Computer
8731	Abbas	NULL	NULL

افزونگی

در جدول studclg

- این اطلاعات که «کد دانشکده‌ی کامپیوتر برابر 12 است.» در سطرهای اول و دوم تکرار شده است، پس افزونگی داده وجود دارد.
- اگر شخصی بخواهد که دانشکده‌ی کامپیوتر را به 22 تغییر دهد و به اشتباه، فقط مقدار clg# در سطر اول جدول studclg را عوض کند، از این به بعد دانشکده کامپیوتر دو کد خواهد داشت: 12 و 22. این موضوع یعنی بی‌نظمی یا آنومالی به معنی نقض جامعیت (صحت و سازگاری) در باک اطلاعات است.
- در سطر آخر از این جدول، مقادیر NULL وجود دارد. به طور کلی، به ازای هر سطر از جدول studclg که clg# آن برابر NULL باشد یک سطر در جدول studclg خواهیم داشت که مقادیر دو ستون آخر آن NULL خواهد بود. پس مشکل مقادیر NULL نیز وجود دارد. حال اگر جدول studclg را به دو جدول clg و stud به شکل زیر تجزیه و نرم‌ال کنیم:

S#	Sname	Clg#	Clg#	Clgname
8621	Ali	12	12	Computer
8442	Reza	12		
8731	Abbas	NULL		

جدول Stud

مشکلات فوق مرتفع و نتایج زیر حاصل خواهد شد.

- کاهش افزونگی طبیعی (محتوایی) مانند تکرار بی‌دلیل سطر (12, computer)
- افزایش افزونگی تکنیکی به دلیل تعریف کلید خارجی (در قالب ستون‌های مشترک) بنابراین کاهش بعضی از انواع افزونگی، طبیعی کاهش و تکنیکی افزایش می‌یابد. بنابراین گزینه اول درست است.
- جداول تجزیه شده را مجدداً در نظر بگیرید:

S#	Sname	Clg#	Clg#	Clgname
8621	Ali	12	12	Computer
8442	Reza	12		
8731	Abbas	NULL		

جدول Stud

حال پرس‌وجوی زیر را در نظر بگیرید:

شماره دانشجویی همه دانشجویان را ارائه دهید.

```
select S#
from stud
```

پُر واضح است که این پرس‌وجو نیازی به الحاق دو جدول stud و clg ندارد. بنابراین این پرس‌وجو سربار حاصل از الحاق دو جدول را ندارد.

حال اگر پرس‌وجوی فوق را بر روی جدول studclg در حالت غیرنرم‌الاجرا کنیم. از آنجا که طول رکورد در جدول studclg نسبت به جدول stud بزرگ‌تر است، واضح است که سرعت اجرای پرس‌وجوی فوق بر روی جدول stud بیش‌تر است. به بیان دیگر زمان اجرای پرس‌وجوی فوق بر روی جدول stud کاهش می‌یابد. مطابق رابطه‌ی زیر:

$$\text{طول رکورد} = \frac{(\text{L})}{(\text{R})} \text{ زمان انتقال رکورد}$$

حال پرس‌وجوی زیر را در نظر بگیرید:

مشخصات کامل همه دانشجویان دانشکده کامپیوتر را ارائه دهید:

```
select *
from stud, clg
where stud.clg# = clg.clg# AND clgname = 'computer'
```

واضح است که این پرس‌وجو نیازی به الحاق دو جدول stud و clg دارد. بنابراین این پرس‌وجو سربار حاصل از الحاق دو جدول را دارد.

در حالی که اگر پرس‌وجوی فوق بر روی جدول studclg در حالت غیرنرم‌الاجرا می‌گردید.

```
select *
from studclg
where clgname = 'computer'
```

نیاز به عمل الحاق و سربار حاصل از آن نبود.

توجه: طول رکورد در پرس‌وجوی اخیر، در هر دو حالت جداول نرم‌الاجرا نشده و جداول نرم‌الاجرا شده برابر است، اما در حالت جداول نرم‌الاجرا شده، برای داشتن تمام ستون‌ها نیاز به یک عمل اضافه و دارای سربار به نام الحاق وجود دارد.

بنابراین از آنجا که نرم‌السازی باعث تجزیه جداول می‌شود گاهی اوقات برای بازیابی نیاز به پیوند مجدد این جداول داریم که باعث افزایش سربار سیستم می‌گردد، البته نه همیشه مانند پرس‌وجوی قبل که نیاز به الحاق نداشت.

بنابراین گزینه دوم نادرست است، زیرا نرم‌السازی رابطه‌ها گاهی موجب افزایش سربار سیستم در عمل بازیابی به دلیل نیاز به الحاق جداول می‌شود و در نتیجه سرعت پاسخگویی به پرس‌وجوهای را کاهش می‌دهد. بنابراین افزایش یا کاهش سرعت پرس‌وجوهای بر روی جداول نرم‌الاجرا شده، بستگی به نوع پرس‌وجوهای دارد.

گزینه سوم و چهارم درست است. زیرا برقراری جامعیت (صحبت و سازگاری) در جداول نرم‌الاجرا شده، ساده‌تر و قابل کنترل‌تر می‌باشد. مانند استفاده از دستور cascade on update برای حفظ سازگاری ستون clg در دو جدول stud و clg در برابر تغییرات. در حالی که در جدول غیرنرم‌الاجرا studclg شده studclg# تغییر 12 به

22 به صورت دستی و توسط انسان انجام می‌شود و خطای انسانی در بر دارد و ممکن است، تغییر 12 به 22 در سطر دوم فراموش گردد! انسان است و جایز الخطأ بودنش. همچنین حل مشکل مقادیر پوچ (NULL)، یکی دیگر از اهداف نرمال‌سازی است و این که حل پدیده آنومالی (بی‌نظمی و ناسازگاری)، یکی دیگر از اهداف نرمال‌سازی است.

۲- گزینه (۲) صحیح است.

گزینه‌های اول و چهارم به صورت مشخص جزء اهداف فرآیند نرمال‌سازی محسوب می‌شوند. گزینه دوم ارتباطی به بحث نرمال‌سازی ندارد و به عنوان یکی از اهداف اصلی نرمال‌سازی محسوب نمی‌شود، زیرا اهداف نرمال‌سازی کاهش افزونگی داده‌ها (کاهش افزونگی طبیعی یا محتوایی) مقابله با بی‌نظمی، آنومالی و ناسازگاری و سهولت در برقراری جامعیت (صحت و سازگاری) و کاهش مقادیر تهی (NULL) است. اینها اهداف نرمال‌سازی هستند، اما در این مسیر و رسیدن به این اهداف لازم است برخی وابستگی‌های افزونگی ساز، مثل وابستگی بخشی (وابستگی صفت غیرکلید به جزئی از کلید کاندید)، وابستگی انتقالی (وابستگی صفت غیرکلید به صفت غیرکلید) حذف گردد، که این مسیر است و نه هدف. که طراح محترم گزینه سوم را به عنوان نرمال‌سازی در نظر گرفته‌اند. البته مفهومی تحت عنوان انواع قواعد وابستگی تابعی در مباحث پایگاه داده وجود ندارد که حال نرمال‌سازی موجب حذف برخی از این قوانین گردد، بهتر است گزینه سوم به «حذف بعضی از وابستگی‌های تابعی» اصلاح گردد که حتی در این صورت نیز به عنوان یکی از اهداف کلی فرآیند نرمال‌سازی محسوب نمی‌شود، بلکه حذف بعضی از وابستگی‌های تابعی، مسیر رسیدن به اهداف نرمال‌سازی است. پس از آن‌که نرمال‌سازی با اهداف مذکور انجام شد، پیاده‌سازی دید کاربر هم در فاز پیاده‌سازی انجام می‌گردد، نرمال‌سازی و کاهش افزونگی‌ها فدای راحتی برای پیاده‌سازی دید کاربر نمی‌گردد. البته متذکر می‌شویم که تسهیل پیاده‌سازی دید کاربر هیچ جایگاهی در اهداف نرمال‌سازی ندارد، نرمال‌سازی که انجام شد، پیاده‌سازی دید کاربر هم براساس نیازها و سلیقه‌ی کاربر انجام خواهد شد.

۳- گزینه (۲) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابر-کلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

توجه: ما نام این روابط را به عنوان مبدع آن «قوانین ارسسطو» نام‌گذاری کردیم، این قوانین به «قوانین چهارگانه ارسسطو» نیز موسوم است.

قانون اول ارسسطو

روشن اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیربدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روشن دوم:

$$\sum_{i=1}^n (x_i - y_i) = \text{عضو کلید کاندید}$$

توجه: عبارت $[x_i - y_i] = \text{رسانید}$ به طور مستقل بر روی تک تک وابستگی‌ها انجام می‌گردد.

$$A \rightarrow BC \Rightarrow BC - A = BC$$

مثال:

توجه: استفاده از روش اول مستلزم گام ابتدایی حذف وابستگی های بدیهی است، اگر در حذف وابستگی های بدیهی دچار خطا می شویم، از روش دوم استفاده نماییم.
با توجه به وابستگی های مطرح شده برای رابطه $R(A,B,C,D)$ داریم:

$$AB \rightarrow C$$

$$AB \rightarrow D$$

$$C \rightarrow A$$

$$D \rightarrow B$$

$$ABC\bar{D} - ABCD =$$

$$\text{تهی}$$

قانون چهارم ارسسطو

هرگاه عضو کلید کاندید، حاصل از تفاصل قانون اول (روش اول یا دوم)، تهی گردد، بدین معنی است که، جدول فوق چندین کلید کاندید دارد، که هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید آن وجود ندارد. بنابراین باید کلید کاندید با بررسی دقیق بر روی مجموعه وابستگی کشف گردد.

$$\{AB\}^+ = \{A, B, C, D\}$$

براساس بستار فوق، صفات AB، همه ستون ها را بدون عضو زائد تولید می کند، پس صفات AB، کلید کاندید می باشد.

حال کلید کاندید AB را در نظر بگیرید، از آن جا که مطابق وابستگی $C \rightarrow A$ ، صفت C را می دهد. می توان به جای صفت A در کلید کاندید AB، صفت C را قرار داد. چون مجدداً مطابق وابستگی $C \rightarrow A$ ، صفت A را می دهد که منجر به ایجاد کلید کاندید CB می گردد.
مطابق الگوی زیر:

$$AB$$

$$\uparrow \Rightarrow CB$$

$$C$$

$$\{CB\}^+ = \{C, B, A, D\}$$

براساس بستار فوق، صفات CB، همه ستون ها را بدون عضو زائد تولید می کند، پس صفات CB کلید کاندید می باشد.

حال مجدداً کاندید AB را در نظر بگیرید، از آن جا که مطابق وابستگی $B \rightarrow D$ صفت D، صفت B را می دهد. می توان به جای صفت B در کلید کاندید AB، صفت D را قرار داد. چون مجدداً مطابق وابستگی $D \rightarrow B$ ، صفت D را می دهد که منجر به ایجاد کلید کاندید AD می گردد.
مطابق الگوی زیر:

$$AB$$

$$\uparrow \Rightarrow AD$$

$$D$$

$$\{AD\}^+ = \{A, D, B, C\}$$

براساس بستار فوق، صفات AD، همه ستون ها را بدون عضو زائد تولید می کند، پس صفات AD کلید کاندید است.

حال کلید کاندید CB یا AD را در نظر بگیرید، براساس وابستگی های $C \rightarrow A$ و $D \rightarrow B$ و مطابق الگوی زیر داریم:

$$\begin{array}{l}
 \text{CB} \Rightarrow \text{CD} \\
 \uparrow \\
 \text{D} \\
 \text{AD} \Rightarrow \text{CD} \\
 \uparrow \\
 \text{C} \\
 \{\text{CD}\}^+ = \{\text{C}, \text{D}, \text{A}, \text{B}\}
 \end{array}$$

براساس بستار فوق، صفات CD، همه ستون‌ها را بدون عضو زائد تولید می‌کند، پس صفات CD کلید کاندید است.

بررسی مجموعه وابستگی‌های مطرح شده نشان داد که (A,B) و (C,B) و (A,D) و (C,D) به عنوان کلیدهای کاندید برای رابطه R محسوب می‌شوند. بنابراین گزینه اول گزاره‌ای درست است. توجه: همان‌طور که مشاهده می‌شود، مطابق قانون چهارم ارسطو، هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید فوق وجود ندارد.

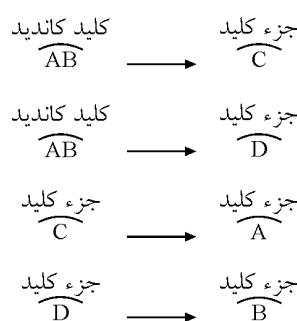
مستقل از متن سؤال گزینه دوم گزاره‌ای نادرست را بیان می‌کند، زیرا طبق تعریف هر رابطه‌ای در BCNF باشد در 3NF هم هست.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم اول را به صورت زیر بیان کرد:

- دارای حداقل یک کلید کاندید باشد.
 - همه خصیصه‌های آن غیرقابل تجزیه باشند (جدول باید فاقد خصیصه‌های مرکب باشد)
 - همه خصیصه‌های آن تک مقداری باشند (جدول باید فاقد خصیصه‌های چند مقداری باشد)
- واضح است که جدول مطرح شده در فرم اول نرمال قرار دارد.
- به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:
- جدول باید در نرمال فرم اول باشد.
 - جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مؤلفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می‌نامند. مؤلفه غیرکلید: هر صفتی که عضو هیچ کلید کاندیدی نباشد، به عنوان مؤلفه غیرکلیدی نامیده می‌شود. مؤلفه جزء کلید کاندید: هر صفتی که عضو حداقل یک کلید کاندید باشد، به عنوان مؤلفه جزء کلید نامیده می‌شود.

وابستگی‌های مطرح شده را در نظر بگیرید:



در وابستگی های فوق، وابستگی بخشی وجود ندارد. بنابراین جدول مربوطه در نرمال فرم دوم هم قرار دارد.

- به طور کلی می توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:
- جدول باید در نرمال فرم دوم باشد.

- جدول باید فاقد وابستگی انتقالی باشد.

وابستگی انتقالی: وابستگی یک مؤلفه غیرکلیدی به یک مؤلفه غیرکلیدی دیگر را وابستگی انتقالی می نامند.

در وابستگی های فوق، وابستگی انتقالی وجود ندارد. بنابراین جدول مربوطه در نرمال فرم سوم هم قرار دارد.

به طور کلی می توان شرط قرار داشتن یک جدول در نرمال فرم BCNF را به صورت زیر بیان کرد:

- سمت چپ همه مجموعه وابستگی ها، ابرکلید باشد.

در وابستگی های فوق، سمت چپ وابستگی اول و دوم در مجموعه داده ابرکلید است. اما سمت چپ وابستگی های سوم و چهارم ابرکلید نیست.

بنابراین این جدول به دلیل نقض شرایط مربوطه، در BCNF واقع نیست. در نتیجه گزینه های سوم و چهارم نیز گزاره های درستی را بیان می کنند.

۴- گزینه (۴) صحیح است.

جدولی که BCNF باشد، حتماً 3NF است، بنابراین گزینه اول درست است به طور کلی می توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم دوم باشد.

- جدول باید فاقد وابستگی انتقالی باشد.

وابستگی انتقالی: وابستگی یک مؤلفه غیرکلیدی به یک مؤلفه غیرکلیدی دیگر را وابستگی انتقالی می نامند.

بنابراین گزینه دوم درست است و گزینه سوم نادرست است. زیرا عدم وابستگی انتقالی برای یک جدول نشانه 3NF بودن یک جدول بلکه جدول باید در نرمال فرم دوم هم باشد.

مثال: جدول زیر را در نظر بگیرید:

a	b	c
1	2	3
3	6	5
7	2	6
1	4	8

در جدول فوق ab کلید کاندید است و صفت غیرکلید c با عضو کلید کاندید یعنی b، وابستگی بخشی دارد. اما نمی توان ادعا کرد که چون جدول فوق فاقد وابستگی انتقالی است، پس 3NF است. زیرا همان طور که گفته ایم جدولی در 3NF است که علاوه بر عدم وابستگی انتقالی، 2NF هم باشد، که جدول فوق به دلیل وجود وابستگی بخشی، 2NF نیست، پس جدول فوق 3NF هم نیست.

۵- گزینه (۴) صحیح است.

جدول (Z,Y,X) R تنها یک کلید کاندید دارد، این جدول فاقد وابستگی بخشی (وابستگی غیرکلید به عضو

کلید کاندید) و وابستگی انتقالی (وابستگی غیرکلید به غیرکلید) می‌باشد. و تنها دارای وابستگی تابعی کامل است.

به صورت زیر:

$R(\underline{X}, Y, Z)$
وابستگی تابعی کامل $X \rightarrow Y$
وابستگی تابعی کامل $X \rightarrow Z$

در جدول R، صفت X، کلید کاندید است.

جدول R به دلیل عدم وجود وابستگی بخشی و وابستگی انتقالی در نرمال فرم سوم (3NF) قرار دارد. هم‌چنین از آنجا که سمت چپ تمام وابستگی‌ها، ابرکلید است، بنابراین جدول فوق در سطح BCNF نیز قرار دارد. هم‌چنین اگر جدولی در BCNF باشد و حداقل یک کلید کاندید تک خصیصه‌ای داشته باشد، حتماً در نرمال فرم چهارم (4NF) هم واقع است. بنابراین جدول فوق در سطح 4NF نیز قرار دارد. هم‌چنین اگر جدولی در نرمال فرم چهارم باشد و همه کلیدهای کاندید آن تک خصیصه‌ای باشند آن جدول در نرمال فرم پنجم واقع است، بنابراین جدول فوق در سطح 5NF نیز قرار دارد. بنابراین جدول R نیاز به نرمال سازی ندارد. اما نظر طراح محترم، اصرار بر نرمال سازی است!

مطابق ضوابط ریسانن تجزیه R به دو رابطه R1 و R2 مطلوب است. اگر و فقط اگر R1 و R2 مستقل باشند، یعنی:

۱- شرط لازم: دو جدول الحق‌پذیر باشند (صفت مشترک در دو جدول، حداقل در یکی کلید کاندید باشد)

۲- شرط کافی: تمام وابستگی‌های تابعی R قابل استنتاج از روی وابستگی‌های تابعی R1 و R2 باشد یا در آنها موجود باشد.

در صورت سؤال مطرح شده است که «کدام گزینه از شرط‌های صحت این عمل می‌باشد؟» بنابراین گزینه چهارم با گزاره اول (شرط لازم) ضوابط ریسانن تطابق دارد.

بنابراین گزینه چهارم، پاسخ این تست خواهد بود.

اما برای رسیدن به تجزیه مطلوب باید هر دو شرط از ضوابط ریسانن برقرار باشد. با توجه به این که صفت X در جدول R کلید کاندید است، و به تبع وابستگی‌های $Y \rightarrow X$ و $Z \rightarrow X$ در این جدول وجود دارد.

با نرمال سازی جدول R به R1 و R2 نیز، وابستگی‌های $Y \rightarrow X$ و $Z \rightarrow X$ باید حفظ گردد.

بنابراین برای برقراری گزاره دوم (شرط کافی) ضوابط ریسانن باید صفت X در دو جدول R1 و R2 کلید کاندید باشد.

$R(\underline{X}, Y, Z)$

$X \rightarrow Y$

$X \rightarrow Z$

پس از تجزیه داریم:

$R1(\underline{X}, Y) \quad R2(\underline{X}, Z)$
 $X \rightarrow Y \quad , \quad X \rightarrow Z$

۶- گزینه (۴) صحیح است.

جدول XYZ , $7 = 1^3$ حالت مختلف، می‌تواند به صورت زیر کلید کاندید داشته باشد.

X Y Z	صفات	تکرار
0 0 0	تهی	
0 0 1	Z	سطر تکراری دارد
0 1 0	Y	سطر تکراری دارد
0 1 1	Y Z	سطر تکراری دارد
1 0 0	X	سطر تکراری دارد
1 0 1	X Z	سطر تکراری دارد
1 1 0	X Y	سطر تکراری دارد
1 1 1	X Y Z	سطر تکراری ندارد

با توجه به جدول فوق و بررسی دقیق محتویات جدول داده شده، واضح است که جدول داده شده تمام کلید است، یعنی کلید کاندید آن XYZ است. طبق تعریف، جدولی که تمام کلید است، در نرمال فرم BCNF قرار دارد. توجه کنید که جدولی که در BCNF است، حتماً در نرمال فرم‌های اول، دوم و سوم هم قرار دارد، بنابراین گزینه‌های اول تا سوم هم درست هستند. بهتر بود بالاترین سطح نرمال این جدول مورد سؤال قرار می‌گرفت.

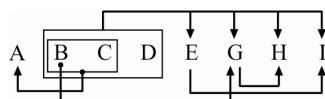
۷- گزینه (۴) صحیح است.

جدولی در سطح نرمال فرم اول قرار دارد که صفت چند مقداری و مرکب نداشته باشد. از آنجا که در رابطه R ، صفت F یک صفت مرکب است، بنابراین این جدول در نرمال فرم اول قرار ندارد. برای نرمال‌سازی این جدول ابتدا صفت مرکب F باید حذف و اجزای آن در جدول R جایگزین گردد.

به این ترتیب جدول R به شکل زیر در نرمال فرم اول قرار می‌گیرد:

$R(A, \underline{B, C}, D, E, G, H, I)$

نمودار وابستگی‌های جدول به صورت زیر است:



به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم اول باشد.

- جدول باید فاقد وابستگی بخشی باشد.

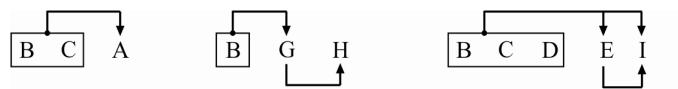
وابستگی بخشی: وابستگی یک مولفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می‌نامند.

با توجه به این که ترکیب صفات BCD به عنوان کلید کاندید رابطه R معرفی شده است، وابستگی‌های تابعی زیر:

$B \rightarrow G$

$BC \rightarrow A$

به عنوان وابستگی بخشی محسوب می‌گردد. بنابراین جداول R در نرمال فرم دوم قرار ندارد. بنابراین برای قرار دادن جداول R، در نرمال فرم دوم، باید وابستگی‌های بخشی از جداول R خارج گردد.



وابستگی انتقالی

بنابراین جدول R به سه جدول کوچک‌تر تجزیه می‌گردد:

$$R3(\underline{B, C, D, E, I}) \quad R2(\underline{B, G, H}) \quad R1(\underline{B, C, A})$$

با توجه به کلید کاندید جدول R1 می‌توان دریافت که این جدول فاقد وابستگی بخشی و انتقالی است، بنابراین این جدول در نرمال فرم سوم قرار دارد. همچنین از آنجا که سمت چپ تمام وابستگی‌های جدول R1 ابرکلید است، بنابراین جدول R1 در سطح BCNF قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم دوم باشد.
- جدول باید فاقد وابستگی انتقالی باشد.

وابستگی انتقالی: وابستگی یک مؤلفه غیرکلیدی به یک مؤلفه غیرکلیدی دیگر را وابستگی انتقالی می‌نامند.
در جدول R2 و R3 وابستگی‌های تابعی زیر:

$$G \rightarrow H$$

$$E \rightarrow I$$

به عنوان وابستگی انتقالی محسوب می‌گردد. بنابراین جداول R2 و R3 در نرمال فرم سوم قرار ندارند. بنابراین برای قرار دادن R2 و R3، در نرمال فرم سوم، باید وابستگی‌های انتقالی از جداول R2 و R3 خارج گردد.



بنابراین جداول R2 و R3 به جداول کوچک‌تر تجزیه می‌گردد:

$$R21(\underline{B, G}) \quad R22(\underline{G, H}) \quad R31(\underline{B, C, D, E}) \quad R32(\underline{E, I})$$

با توجه به کلیدهای کاندید در جداول R21، R22، R31 و R32 می‌توان دریافت که این جداول فاقد وابستگی بخشی و انتقالی هستند و بنابراین در نرمال فرم سوم قرار دارند. همچنین از آنجا که سمت چپ تمام وابستگی‌های جداول R21، R22، R31 و R32 ابرکلید است، بنابراین جداول R21، R22، R31 و R32 در سطح BCNF نیز قرار دارند. به این ترتیب با تجزیه جدول R به پنج جدول R1، R21، R22، R31 و R32 می‌توان جدول R را در نرمال فرم سوم و سطح BCNF قرار داد.

- گزینه (۱) صحیح است.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم اول باشد.
- جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مؤلفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می‌نامند.

با توجه به این که ترکیب صفات BCD به عنوان کلید کاندید رابطه R معرفی شده است، وابستگی‌های زیر را داریم:

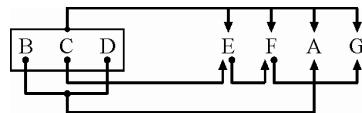
B	C	D	وابستگی‌ها
0	0	0	ϕ
0	0	1	$D \rightarrow X$
0	1	0	$C \rightarrow X$
0	1	1	$CD \rightarrow X$
1	0	0	$B \rightarrow X$
1	0	1	$BD \rightarrow X$
1	1	0	$BC \rightarrow X$
1	1	1	$BCD \rightarrow R$

وابستگی بخشش
X : مؤلفه غیر کلیدی
گزینه اول
وابستگی تابعی کامل }

بنابراین فرم نرمال دوم جدول R به واسطه وابستگی بخشی $E \rightarrow BC$ موجود در گزینه اول، نقض می‌گردد.

- ۹- گزینه (۴) صحیح است.

جدولی در سطح نرمال فرم اول قرار دارد که صفت چند مقداری و مرکب نداشته باشد. از آنجا که در جدول R ، صفت چند مقداری و مرکب وجود ندارد، بنابراین این جدول در نرمال فرم اول قرار دارد. نمودار وابستگی‌های جدول R به صورت زیر است:



به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

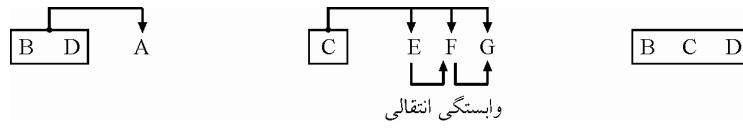
- جدول باید در نرمال فرم اول باشد.
- جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مؤلفه غیر کلیدی، به جزئی از کلید کاندید را وابستگی بخشی می‌نامند. با توجه به این که ترکیب صفات BCD به عنوان کلید رابطه R معرفی شده است، وابستگی‌های تابعی زیر:

$BD \rightarrow A$

$C \rightarrow E$

به عنوان وابستگی بخشی محسوب می‌گردد. بنابراین جدول R در نرمال فرم دوم قرار ندارد. بنابراین برای قرار دادن جدول R در نرمال فرم دوم، باید وابستگی‌های بخشی از جدول R خارج گردد.



بنابراین جدول R به سه جدول کوچکتر تجزیه می‌گردد:

$R1(B, D, A) \quad R2(C, E, F, G) \quad R3(B, C, D)$

با توجه به کلید کاندید در دو جدول $R1$ و $R3$ می‌توان دریافت که این دو جدول فاقد وابستگی بخشی و انتقالی هستند، بنابراین در نرمال فرم سوم قرار دارند. همچنین از آنجا که سمت چپ تمام وابستگی‌های دو جدول $R1$ و $R3$ ، ابرکلید است، بنابراین دو جدول $R1$ و $R3$ در سطح BCNF نیز قرار دارند.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

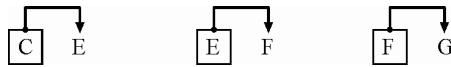
- جدول باید در نرمال فرم دوم باشد.
- جدول باید فاقد وابستگی انتقالی باشد.

وابستگی انتقالی: وابستگی یک مؤلفه غیرکلیدی به یک مؤلفه غیرکلیدی دیگر را وابستگی انتقالی می‌نامند. در جدول R2 وابستگی‌های تابعی زیر:

$$E \rightarrow F$$

$$F \rightarrow G$$

به عنوان وابستگی انتقالی محسوب می‌گردد. بنابراین جدول R2 در نرمال فرم سوم قرار ندارد. بنابراین برای قرار دادن جدول R2، در نرمال فرم سوم، باید وابستگی‌های انتقالی از جدول R2 خارج گردد.



بنابراین جدول R2 به سه جدول کوچک‌تر تجزیه می‌گردد.

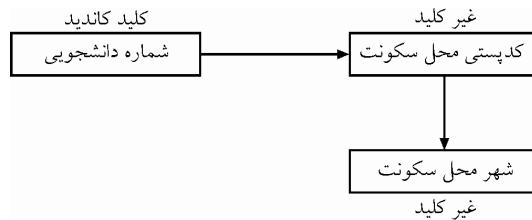
$$R21(\underline{C}, E) \quad R22(\underline{E}, F) \quad R23(\underline{F}, G)$$

با توجه به کلیدهای کاندید در دو جدول R21، R22 و R23 می‌توان دریافت که این سه جدول فاقد وابستگی بخشی و انتقالی هستند و بنابراین در نرمال فرم سوم قرار دارند. هم‌چنین از آن‌جا که سمت چپ تمام وابستگی‌های سه جدول R21، R22 و R23 ابرکلید است، بنابراین سه جدول R21، R22 و R23 در سطح BCNF نیز قرار دارند. به این ترتیب با تجزیه جدول R به پنج جدول R1، R21، R22 و R3 و R4 می‌توان جدول R را در نرمال فرم سوم و سطح BCNF قرار داد.

۱۰- گزینه (۳) صحیح است.

قانون افزایشی به این صورت بیان می‌شود که اگر وابستگی چند مقداری $Y \rightarrow\rightarrow X$ موجود باشد و $V \subseteq W$ آن‌گاه می‌توان نتیجه گرفت $WY \rightarrow\rightarrow VY$.
به بیان دیگر با توجه به این که $V \subseteq W$ است، می‌توان نتیجه گرفت که $V \rightarrow\rightarrow W$ است و از طرفی دیگر با توجه به فرض مسئله $Y \rightarrow\rightarrow X$ می‌توان نتیجه گرفت $VY \rightarrow\rightarrow WY$.

۱۱- گزینه (۴) صحیح است.



از آن‌جا که در جدول دانشجو، صفت چند مقداری و مرکب وجود ندارد، بنابراین این جدول در نرمال فرم اول قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم اول باشد.
- جدول باید فاقد وابستگی بخشی باشد.

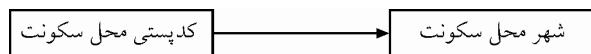
جدول دانشجو در حال حاضر $2NF$ است، زیرا کلید کاندید آن شماره دانشجویی است و وابستگی بخشی وجود ندارد.

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 مولفه غیرکلیدی به یک مولفه غیرکلیدی دیگر را، وابستگی انتقالی می‌نامند.

در جدول دانشجو، وابستگی تابعی زیر:



به عنوان وابستگی انتقالی محسوب می‌گردد. بنابراین جدول دانشجو در نرمال فرم سوم قرار ندارند. بنابراین برای قرار دادن جدول دانشجو، در نرمال فرم سوم، باید وابستگی‌های انتقالی از جدول دانشجو خارج کردن.



بنابراین جدول دانشجو به دو جدول کوچکتر تجزیه می‌گردد.

(شهر محل سکونت و کد پستی محل سکونت) $stu2$ (کد پستی محل سکونت و شماره دانشجویی) $stu1$ با توجه به کلیدهای کاندید در دو جدول $stu1$ و $stu2$ می‌توان دریافت که این دو جدول فاقد وابستگی بخشی و انتقالی هستند، بنابراین در نرمال فرم سوم قرار دارند. هم‌چنین از آنجا که سمت چپ تمام وابستگی‌های دو جدول $stu1$ و $stu2$ برکلید است، بنابراین این دو جدول در سطح BCNF نیز قرار دارند. به این ترتیب با تجزیه جدول دانشجو به دو جدول $stu1$ و $stu2$ می‌توان جدول دانشجو را در نرمال فرم سوم و سطح BCNF قرار داد.

البته این دو جدول در نرمال فرم‌های چهارم و پنجم نیز قرار دارند، اما مطابق آنچه در گزینه‌ها آمده است، گزینه چهارم به عنوان پاسخ خواهد بود.

۱۲- گزینه (۴) صحیح است.

راه حل اول:

فرم کمینه وابستگی‌های مطرح شده به صورت زیر است:

۱- گام اول: تجزیه همه صفت‌های سمت راست.

$A \rightarrow I$

$A \rightarrow E$

$A \rightarrow G$

$AB \rightarrow I$ $AB \rightarrow E$ $AB \rightarrow C$ $AB \rightarrow D$ $ABE \rightarrow C$ $ABE \rightarrow D$ $ABE \rightarrow F$ $ABE \rightarrow G$ $ABE \rightarrow H$ $AC \rightarrow I$ $AC \rightarrow E$

C زیر مجموعه AC است، بنابراین این وابستگی بدیهی بوده و حذف می‌گردد.

 $AC \rightarrow C \times$

DE زیر مجموعه CDE است، بنابراین این وابستگی بدیهی بوده و حذف می‌گردد.

 $CDE \rightarrow DE \times$ $E \rightarrow F$ $E \rightarrow G$ $F \rightarrow E$

۲- گام دوم: خلاصه‌سازی صفت‌های سمت چپ تا حد امکان

 $A \rightarrow I$ $A \rightarrow E$ $A \rightarrow G$ $AB \rightarrow I$ $AB \rightarrow E$ $AB \rightarrow C$ $AB \rightarrow D$

زیرا توسط وابستگی $E \rightarrow AB$ ، صفات E را می‌دهد.

 $ABE \rightarrow C \Rightarrow AB \rightarrow C$ $ABE \rightarrow D \Rightarrow AB \rightarrow D$ $ABE \rightarrow F \Rightarrow AB \rightarrow F$ $ABE \rightarrow G \Rightarrow AB \rightarrow G$ $ABE \rightarrow H \Rightarrow AB \rightarrow H$ $AC \rightarrow I$ $AC \rightarrow E$ $E \rightarrow F$ $E \rightarrow G$ $F \rightarrow E$

۳- گام سوم: حذف وابستگی‌های زائدی که از روی سایر وابستگی‌ها قابل تولید است.

$$A \rightarrow I \checkmark$$

$$A \rightarrow E \checkmark$$

$$A \rightarrow G \times$$

با داشتن صفت A بدون وابستگی $G \rightarrow A$ ، توسط وابستگی‌های $A \rightarrow E$ و $E \rightarrow G$ می‌توان به صفت G رسید.

$$\{A\}^+ = \{A, E, \boxed{G}\}$$

$$AB \rightarrow I \times$$

با داشتن صفات AB بدون وابستگی $AB \rightarrow I$ ، توسط وابستگی $I \rightarrow A$ می‌توان به صفت I رسید. بنابراین وابستگی $I \rightarrow AB$ حذف می‌گردد.

$$AB \rightarrow E \times$$

با داشتن صفات AB بدون وابستگی $AB \rightarrow E$ ، توسط وابستگی $E \rightarrow A$ می‌توان به صفت E رسید. بنابراین وابستگی $E \rightarrow AB$ حذف می‌گردد.

$$AB \rightarrow C \checkmark$$

$$AB \rightarrow D \checkmark$$

$$AB \rightarrow C \times$$

$$AB \rightarrow D \times$$

$$AB \rightarrow F \times$$

با داشتن صفات AB بدون وابستگی $AB \rightarrow F$ ، توسط وابستگی $F \rightarrow A$ و $A \rightarrow E$ می‌توان به صفت F رسید. بنابراین وابستگی $F \rightarrow AB$ حذف می‌گردد.

$$\{AB\}^+ = \{A, B, E, \boxed{F}\}$$

$$AB \rightarrow G \times$$

با داشتن صفات AC بدون وابستگی $AC \rightarrow AB$ ، توسط وابستگی $AB \rightarrow G$ می‌توان به صفت G رسید. بنابراین وابستگی $G \rightarrow AB$ حذف می‌گردد.

$$AB \rightarrow H \checkmark$$

$$AC \rightarrow I \times$$

با داشتن صفات AC بدون وابستگی $AC \rightarrow I$ ، توسط وابستگی $I \rightarrow A$ ، می‌توان به صفت I رسید. بنابراین وابستگی $I \rightarrow AC$ حذف می‌گردد.

$$AC \rightarrow E \times$$

با داشتن صفات AC بدون وابستگی $AC \rightarrow E$ ، توسط وابستگی $E \rightarrow A$ ، می‌توان به صفت E رسید. بنابراین وابستگی $E \rightarrow AC$ حذف می‌گردد.

$$E \rightarrow F \checkmark$$

$$E \rightarrow G \checkmark$$

$$F \rightarrow E \checkmark$$

بنابراین وابستگی های کمینه، مجموعه وابستگی های مطرح شده به صورت زیر خواهد بود:

$$A \rightarrow I$$

$$A \rightarrow E$$

$$AB \rightarrow C$$

$$AB \rightarrow D$$

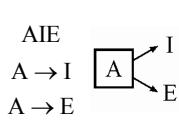
$$AB \rightarrow H$$

$$E \rightarrow F$$

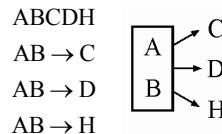
$$E \rightarrow G$$

$$F \rightarrow E$$

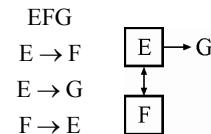
حال با توجه به وابستگی های مربوط به هر جدول، سطح نرمال مربوط به هر جدول را می توان مشخص نمود:



کلید کاندید A می باشد.



کلید کاندید AB می باشد.



کلید کاندید E یا F می باشد.

همان طور که مشاهده می شود سمت چپ تمام وابستگی های تابعی در هر سه جدول ابرکلید است، بنابراین هر سه جدول در BCNF قرار دارند. بنابراین پایگاه داده در BCNF قرار دارد.

راه حل دوم: پیدا کردن کلید کاندید هر جدول (بدون نیاز به کمینه سازی وابستگی ها)

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می آید:

قانون اول ارسسطو

روش اول:

اجتماع تمام خصیصه های سمت راست وابستگی های غیر بدیهی - تمام خصیصه های جدول = عضو کلید کاندید

روش دوم:

$$\text{عضو کلید کاندید} = R - \bigcup_{i=1}^n [y_i(x_i) - \text{(راست)}]$$

با توجه به وابستگی های مطرح شده برای رابطه $R1(A, I, E)$ داریم:

$$A \rightarrow IEG$$

$$AB \rightarrow IECD$$

$$ABE \rightarrow CDFGH$$

$$AC \rightarrow IEC$$

* $CDE \rightarrow DE$ ⇒ وابستگی بدیهی، کنار گذاشته می‌شود.

$E \rightarrow FG$

$F \rightarrow E$

$AIE - IEGCDFH = A$

بنابر رابطه فوق صفت A حتماً باید عضو کلید کاندید باشد.

بستار صفت A فقط برای جدول $R1(A,I,E)$ به صورت زیر است:

$$\{A\}^+ = \{A, I, E\}$$

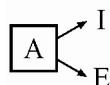
براساس بستار فوق، صفت A ، همه ستون‌های جدول $R1(A,I,E)$ را تولید می‌کند، پس صفت A کلید کاندید می‌باشد.

قانون دوم ارسسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

براساس وابستگی‌های مطرح شده جدول $R1(A,I,E)$ با کلید کاندید A ، فاقد هر گونه وابستگی بخشی (وابستگی غیرکلید به عضو کلید کاندید) و انتقالی (وابستگی غیرکلید به غیرکلید) است. بنابراین این جدول در فرم نرمال 3NF قرار دارد.

بنابراین نمودار وابستگی تابعی جدول $R1(A,I,E)$ به صورت زیر خواهد بود:



همان‌طور که مشاهده می‌شود سمت چپ تمام وابستگی‌های تابعی جدول $R1(A,I,E)$ ابرکلید است، بنابراین جدول $R1(A,I,E)$ در BCNF قرار دارد.

هم‌چنین با توجه به وابستگی‌های مطرح شده برای جدول $R2(A,B,C,D,H)$ داریم:

$A \rightarrow IEG$

$AB \rightarrow IECD$

$ABE \rightarrow CDFGH$

$AC \rightarrow IEC$

* $CDE \rightarrow DE$ ⇒ وابستگی بدیهی، کنار گذاشته می‌شود

$E \rightarrow FG$

$F \rightarrow E$

$ABCDFH - IEGCDFH = AB$

بنابر رابطه فوق صفات AB حتماً باید عضو کلید کاندید باشد.

بستار صفات AB فقط برای جدول $R2(A,B,C,D,H)$ به صورت زیر است:

$$\{AB\}^+ = \{A, B, C, D, H\}$$

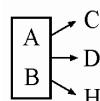
براساس بستار فوق، صفات AB ، همه ستون‌های جدول $R2(A,B,C,D,H)$ را تولید می‌کند، پس صفات AB کلید کاندید می‌باشد.

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.

براساس وابستگی‌های مطرح شده جدول $R2(A, B, C, D, H)$ با کلید کاندید AB ، فاقد هرگونه وابستگی بخشی (وابستگی غیرکلید به عضو کلید کاندید) و انتقالی (وابستگی غیرکلید به غیرکلید) است. بنابراین جدول در فرم نرمال 3NF قرار دارد.

بنابراین نمودار وابستگی تابعی جدول $R2(A, B, C, D, H)$ به صورت زیر خواهد بود:



همان‌طور که مشاهده می‌شود سمت چپ تمام وابستگی‌های تابعی جدول $R2(A, B, C, D, H)$ ابرکلید است، بنابراین جدول $R2(A, B, C, D, H)$ در BCNF قرار دارد.

هم‌چنین با توجه به وابستگی‌های مطرح شده برای جدول $R3(E, F, G)$ داریم:

$$A \rightarrow IEG$$

$$AB \rightarrow IECD$$

$$ABE \rightarrow CDFGH$$

$$AC \rightarrow IEC$$

* $CDE \rightarrow DE$ \Rightarrow کنار گذاشته می‌شود وابستگی بدیهی،

$$E \rightarrow FG$$

$$F \rightarrow E$$

$$EFG - IEGCDFH = \text{تهی}$$

قانون چهارم ارسطو

هرگاه عضو کلید کاندید حاصل از تفاضل قانون اول (روش اول یا دوم)، تهی گردد، بدین معنی است که، جدول فوق چندین کلید کاندید دارد، که هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید آن وجود ندارد. بنابراین باید کلید کاندید با بررسی دقیق بر روی مجموعه وابستگی کشف گردد.

با کمی دقت و بررسی پُر واضح است که براساس وابستگی‌های مطرح شده، در جدول $R3(E, F, G)$ صفت E و F کلید کاندید هستند. در واقع جدول $R3$ دو کلید کاندید دارد.

$$\{E\}^+ = \{E, F, G\}$$

براساس بستار فوق صفت E ، همه ستون‌های جدول $R3$ را تولید می‌کند، پس صفت E کلید کاندید است.

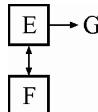
$$\{F\}^+ = \{F, E, G\}$$

براساس بستار فوق صفت F ، همه ستون‌های جدول $R3$ را تولید می‌کند، پس صفت F کلید کاندید است.

توجه: همان‌طور که مشاهده می‌شود، مطابق قانون چهارم ارسطو، هیچ عضو کلید کاندید مشترکی، بین تمامی کلیدهای کاندید فوق وجود ندارد.

براساس وابستگی های مطرح شده جدول $R3(E,F,G)$ با دو کلید کاندید E و F ، فاقد هر گونه وابستگی بخشی (وابستگی غیرکلید به عضو کلید کاندید) و انتقالی (وابستگی غیرکلید به غیرکلید) است. بنابراین این جدول در فرم نرمال 3NF قرار دارد.

بنابراین نمودار وابستگی های تابعی جدول $R3(E,F,G)$ به صورت زیر خواهد بود.



همانطور که مشاهده می شود سمت چپ تمام وابستگی های تابعی جدول $R3(E,F,G)$ ابرکلید است، بنابراین جدول $R3(E,F,G)$ در BCNF هم قرار دارد.

۱۳- گزینه (۱) صحیح است.

زمانی برای جداول یک بانک اطلاعاتی کلید اصلی تعیین می گردد که جداول یک بانک اطلاعاتی با هدف کاهش افروندگی های طبیعی تا حد BCNF، مقابله با بی نظمی ها و ناسازگاری ها و مقادیر تهی نرمال سازی شده باشند.

پس در طراحی بانک اطلاعاتی رابطه ای، پدیده نرمال سازی (روال تجزیه یا کاهش) در اولویت اول قرار دارد.

شرط تجزیه مطلوب به صورت زیر است:

۱- شرط لازم (**Nonloss join**): دو جدول الحقیقتی باشند (صفت مشترک در دو جدول، حداقل در یکی کلید کاندید باشد) این شرط، مهم ترین شرط نرمال سازی است، زیرا شرط لازم است.

۲- شرط کافی (**حفظ وابستگی های تابعی**): وابستگی های تابعی جدول پایه قابل استنتاج از روی وابستگی های تابعی جداول حاصل از تجزیه باشد یا در آنها موجود باشد.

در ادامه پس از نرمال سازی جداول یک بانک اطلاعاتی، برای هر یک از جداول، یک کلید اصلی توسط طراح بانک اطلاعات، مشخص می گردد.

۱۴- گزینه (۲) صحیح است.

عملیات نرمال سازی یک عملیات چند مرحله ای است که با طی آنها اهداف نرمال سازی به تدریج حاصل می شود. نخستین مرحله نرمال سازی تبدیل جداول به نرمال فرم اول است. برای قرار گرفتن یک جدول در نرمال فرم اول، ان جدول باید در شروط خاصی صدق کند. یک جدول (رابطه) برای برقرار گرفتن در نرمال فرم اول باید دارای شروط زیر باشد: (حفظ قوانین جامعیت درون رابطه ای)

۱- دارای حداقل یک کاندید باشد.

۲- جدول تو جدول یا جدول تو در تو نباشد (یعنی جدول فاقد صفت مرکب و چند مقداری باشد)

۱-۲- همه صفات آن غیرقابل تجزیه (Atomic) باشند. (جدول باید فاقد صفت مرکب باشد)

۲-۲- همه صفات جدول تک مقداری باشند. (جدول باید فاقد صفت چند مقداری باشد).

بنابراین گزینه های اول تا سوم درست هستند.

به طور کلی می توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم دوم باشد.
- جدول باید فاقد وابستگی انتقالی (تراگذری) باشد.

وابستگی انتقالی (تراگذری): وابستگی یک مؤلفه غیر کلیدی به یک مؤلفه غیر کلیدی دیگر را وابستگی انتقالی می‌نامند.
بنابراین گزینه چهارم ارتباطی به نرمال فرم اول ندارد و پاسخ این سؤال خواهد بود.

۱۵- گزینه (۱) صحیح است.

جدول studclg با مقادیر آن به شکل زیر را در نظر بگیرید:

S#	Sname	Clg#	Clgname
8621	Ali	12	Computer
8442	Reza	12	Computer
8731	Abbas	NULL	NULL

در جدول studclg

- این اطلاعات که «کد دانشکده کامپیوتر برابر ۱۲ است» در سطرهای اول و دوم تکرار شده است، پس افزونگی داده وجود دارد.
- اگر شخصی بخواهد کد دانشکده کامپیوتر را به ۲۲ تغییر دهد و به اشتباہ، فقط مقدار clg# در سطر اول جدول studclg را عوض کند، از این به بعد دانشکده کامپیوتر دو کد خواهد داشت: ۱۲ و ۲۲. این موضوع یعنی بی‌نظمی یا آنومالی به معنی نقض جامعیت (صحت و سازگاری) در بانک اطلاعات است.
- در سطر آخر از این جدول، مقادیر NULL وجود دارد. به طور کلی، به ازای هر سطر از جدول clg# آن برابر NULL باشد یک سطر در جدول studclg خواهیم داشت که مقادیر دو ستون آخر آن NULL خواهد بود. پس مشکل مقادیر NULL نیز وجود دارد. حالا گر جدول studclg را به دو جدول clg و stud به شکل زیر تجزیه و نرمال کنیم:

S#	Sname	Clg#	Clg#	Clgname
8621	Ali	12	12	Computer
8442	Reza	12		
8731	Abbas	NULL		

جدول Stud

مشکلات فوق مرتفع و نتایج زیر حاصل خواهد شد.

- کاهش افزونگی طبیعی (محتوایی) مانند تکرار بی‌دلیل سطر (12, computer)
- افزایش افزونگی تکنیکی به دلیل تعریف کلید خارجی (در قالب ستون‌های مشترک) بنابراین کاهش بعضی از انواع افزونگی، طبیعی کاهش و تکنیکی افزایش می‌یابد. بنابراین گزینه سوم درست است.

جداول تجزیه شده را مجدداً در نظر بگیرید:

S#	Sname	Clg#	Clg#	Clgname
8621	Ali	12	12	Computer
8442	Reza	12		
8731	Abbas	NULL		

جدول Clg

حال پرس‌وجوی زیر را در نظر بگیرید:

شماره دانشجویی همه دانشجویان را ارائه دهید:

```
select S#
from stud
```

پُر واضح است که این پرس‌وجو نیازی به الحاق دو جدول stud و clg ندارد.
بنابراین این پرس‌وجو سربار حاصل از الحاق دو جدول را ندارد.

حال اگر پرس‌وجوی فوق را بر روی جدول studclg در حالت غیرنرمال اجرا کنیم. از آنجا که طول رکورد در جدول studclg نسبت به جدول stud بزرگتر است، واضح است که سرعت اجرای پرس‌وجوی فوق بر روی جدول stud افزایش می‌یابد، به بیان دیگر زمان اجرای پرس‌وجوی فوق بر روی جدول stud کاهش می‌یابد. مطابق رابطه‌ی زیر:

$$\text{طول رکورد} = \frac{\text{زمان انتقال رکورد}}{\text{نرخ انتقال}}$$

حال پرس‌وجوی صفحه بعد را در نظر بگیرید:

مشخصات کامل همه دانشجویان دانشکده کامپیوتر را ارائه دهید:

```
select *
from stud, clg
where stud.clg# = clg.clg# AND clgname= 'computer'
```

واضح است که این پرس‌وجو و جو نیاز به الحاق دو جدول stud و clg دارد.
بنابراین این پرس‌وجو سربار حاصل از الحاق دو جدول را دارد.

در حالی که اگر پرس‌وجوی فوق بر روی جدول studclg در حالت غیرنرمال به شکل زیر اجرا می‌گردید.

```
select *
from studclg
where clgname= 'computer'
```

نیاز به عمل الحاق و سربار حاصل از آن نبود.

توجه: طول رکورد در پرس‌وجوی اخیر، در هر دو حالت جدول نرمال نشده و جداول نرمال شده برابر است، اما در حالت جداول نرمال شده، برای داشتن تمام ستون‌ها، نیاز به یک عمل اضافه و دارای سربار به نام الحاق وجود دارد.

بنابراین از آنجا که نرمال‌سازی باعث تجزیه جداول می‌شود، گاهی اوقات برای بازیابی نیاز به پیوند مجدد این جداول داریم که باعث افزایش سربار سیستم می‌گردد، البته نه همیشه مانند پرس‌وجوی قبل که نیاز به الحاق نداشت.

بنابراین گزینه اول نادرست است، زیرا نرمال‌سازی رابطه‌ها گاهی موجب افزایش سربار سیستم در عمل بازیابی به دلیل نیاز به الحاق جداول می‌شود و در نتیجه سرعت پاسخگویی به پرس‌وجوها را کاهش می‌دهد. بنابراین افزایش یا کاهش سرعت پرس‌وجوها بر روی جداول نرمال شده، بستگی به نوع پرس‌وجوها دارد.

نرمال‌سازی تا سطح 3NF و در صورت نیاز تا سطح BCNF اجباری است بنابراین گزینه دوم هم درست است.

هم چنین حل مشکل مقادیر پوچ (NULL)، یکی دیگر از اهداف نرمال سازی است. بنابراین گزینه چهارم هم درست است.

و این که حل پدیده آنومالی (بی نظمی و ناسازگاری) یکی دیگر از اهداف نرمال سازی است.

۱۶- گزینه (۲) صحیح است.

طبق تعریف هر جدولی که در نرمال فرم دوم باشد، حتماً در نرمال فرم اول نیز قرار دارد (گزاره اول درست است)، اما عکس این مطلب همواره برقرار نیست. (گزاره دوم نادرست است)، همچنین طبق تعریف هر جدولی که در BCNF باشد. حتماً در نرمال فرم سوم نیز قرار دارد (گزاره سوم درست است)، اما عکس این مطلب همواره برقرار نیست (گزاره چهارم نادرست است)

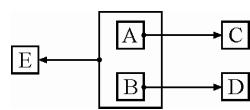
نرمال سازی مانند درجات علمی است.

دیپلم، فوق دیپلم، لیسانس، فوق لیسانس، دکتری، فوق دکتری

5NF، 4NF، BCNF، 3NF، 2NF، 1NF

۱۷- گزینه (۴) صحیح است.

شكل زیر، نمودار وابستگی های تابعی صفات را نشان می دهد. پُر واضح است که کلید این جدول صفات AB است. بنابراین بدیهی است که جدول 2NF نیست به دلیل وجود وابستگی های بخشی $C \rightarrow A$ و $D \rightarrow B$ (وابستگی غیرکلید به عضو کلید کاندید) پس جدول مذکور 3NF نیز نیست. پس سه گزینه اول نادرست است. بنابراین گزینه چهارم درست است.



به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می آید:

قانون اول ارسسطو

روشن اول:

اجتماع تمام خصیصه های سمت راست وابستگی های غیربدیهی - تمام خصیصه های جدول = عضو کلید کاندید

روشن دوم:

$$\text{عضو کلید کاندید} = R - \left(\bigcup_{i=1}^n [x_i (چپ) - y_i (راست)] \right)$$

با توجه به وابستگی های مطرح شده برای رابطه $R(A,B,C,D,E)$ داریم:

$$A \rightarrow C$$

$$B \rightarrow D$$

$$AB \rightarrow E$$

$$ABC\overline{D}\overline{E} - CDE = AB$$

بنابر رابطه فوق صفت AB حتماً باید عضو کلید کاندید باشد.
بستار صفت AB به صورت زیر است:

$$\{AB\}^+ = \{A, B, C, D, E\}$$

براساس بستار فوق، صفات AB ، همه ستونها را تولید می‌کند، پس صفات AB ، کلید کاندید می‌باشد.

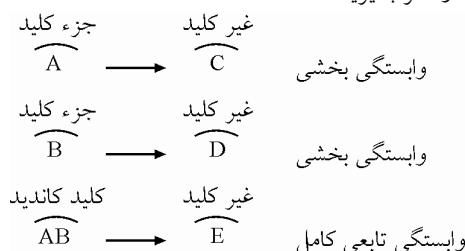
قانون دوم ارسسطو

هرگاه عضو کلید کاندید حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستونها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم اول باشد. (که هست، زیرا صفت چند مقداری و مرکب ندارد.)
- جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مؤلفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می‌نامند.
وابستگی‌های مطرح شده را در نظر بگیرید:



در وابستگی‌های فوق، وابستگی بخشی $C \rightarrow A$ و $D \rightarrow B$ وجود دارد، بنابراین جدول مربوطه در نرمال فرم دوم قرار ندارد، بنابراین رویه نرمال‌سازی (تجزیه مطلوب) جهت قرارگیری جدول R در نرمال فرم دوم توسط عمل تجزیه جدول R می‌بایست انجام گردد.

شرطیت تجزیه مطلوب به صورت زیر است:

۱- شرط لازم: دو جدول الحقیقی‌بذری باشد (صفت مشترک در دو جدول، حداقل در یکی کلید کاندید باشد)

شرط الحقیقی‌بذری به دو شکل زیر انجام می‌گردد:

۱-۱- برای حذف وابستگی بخشی از تئوری هیث استفاده می‌گردد.

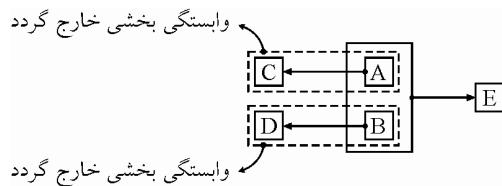
تئوری هیث: این تئوری می‌گوید، وابستگی بخشی (وابستگی غیرکلید به جز کلید کاندید) را از جدول پایه خارج کنید.

۱-۲- برای حذف وابستگی انتقالی از تئوری ریسانن استفاده می‌گردد.

تئوری ریسانن: این تئوری می‌گوید، وابستگی انتقالی (وابستگی غیرکلید به غیرکلید) را از جدول پایه خارج کنید.

۲- شرط کافی: وابستگی‌های تابعی جدول پایه قابل استنتاج از روی وابستگی‌های تابعی جداول حاصل از تجزیه باشد یا در آن‌ها موجود باشد.

با توجه به نمودار وابستگی‌های جدول $R(A, B, C, D, E)$ داریم:



بنابراین، پس از انجام مرحله نرمال سازی فرم دوم داریم:

$$\begin{array}{lll} R1(A,B,E) & R2(B,D) & R3(A,C) \\ AB \rightarrow E & B \rightarrow D & A \rightarrow C \end{array}$$

توجه: در این تجزیه شرط کافی هم برقرار است، زیرا وابستگی های تابعی جدول پایه، قابل استنتاج از روی وابستگی های تابعی جداول حاصل از تجزیه می باشد.

به طور کلی می توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم دوم باشد (که هست، زیرا وابستگی های بخشی از جدول پایه خارج گردید).
- جدول باید فاقد وابستگی انتقالی باشد.

وابستگی انتقالی: وابستگی یک مولفه غیرکلیدی به یک مولفه غیرکلیدی دیگر را وابستگی انتقالی می نامند. همان‌طور که مشاهده می شود در هیچ یک از وابستگی های تابعی جداول R2, R1 و R3 وابستگی انتقالی وجود ندارد. بنابراین این پایگاه داده در نرمال فرم سوم هم قرار دارد. هم‌چنین همان‌طور که واضح است، سمت چپ تمام وابستگی های تابعی در هر سه جدول ابرکلید است و بنابراین هر سه جدول در BCNF قرار دارند، بنابراین این پایگاه داده در BCNF قرار دارد.

۱۸- گزینه (۴) صحیح است.

طبق تعریف رابطه‌ای که در BCNF است، حتماً در 3NF است و رابطه‌ای که در 3NF است حتماً در 2NF نیز واقع است. هم‌چنین رابطه‌ای که در 2NF است حتماً در 1NF هم قرار دارد. در نتیجه می‌توان گفت که رابطه‌ای که در BCNF است حتماً در نرمال فرم‌های اول، دوم و سوم هم قرار دارد. اما رابطه‌ای که در BNCF است ممکن است در 4NF باشد و البته ممکن است در 4NF نباشد.

۱۹- گزینه (۱) صحیح است.

گزینه‌های دوم تا چهارم به صورت مشخص جزو اهداف فرآیند نرمال سازی محسوب می‌شوند. گزینه اول ارتباطی به بحث نرمال سازی ندارد و به عنوان یکی از اهداف اصلی نرمال سازی محسوب نمی‌شود. زیرا اهداف نرمال سازی کاهش افزونگی داده‌ها (کاهش افزونگی طبیعی یا محتوایی)، مقابله با بی‌نظمی، آنومالی و ناسازگاری و سهوالت در برقراری قوانین جامعیت (صحبت و سازگاری) و کاهش مقادیر تهی (NULL) است. اینها اهداف نرمال سازی هستند، اما در این مسیر و رسیدن به این اهداف لازم است برخی وابستگی‌های افزونگی ساز، مثل وابستگی بخشی (وابستگی صفت غیرکلید به جزئی از کلید کاندید) وابستگی انتقالی (وابستگی صفت غیرکلید به صفت غیرکلید) حذف گردد، که این مسیر است و نه هدف. پس از آن که نرمال سازی با اهداف مذکور انجام شد، پیاده‌سازی دید کاربر هم در فاز پیاده‌سازی انجام می‌گردد. نرمال سازی و کاهش افزونگی‌ها فدای راحتی برای پیاده‌سازی دید کاربر نمی‌گردد. البته متذکر

می‌شویم که تسهیل پیاده‌سازی دید کاربر هیچ جایگاهی در اهداف نرم‌السازی ندارد، نرم‌السازی که انجام شد، پیاده‌سازی دید کاربر هم براساس نیازها و سلیقه‌ی کاربر انجام خواهد شد.

۲۰- گزینه (۴) صحیح است.

بر طبق تعریف، جدولی در BCNF است که ستون‌های آن فقط به کلیدهای کاندیدش وابستگی تابعی داشته باشند، اما در این جدول، وابستگی $S\# \rightarrow \text{Code}$ یک وابستگی به ستون‌های غیرکلیدی است.

۲۱- گزینه (۴) صحیح است.

برای مشاهده مشکلات، به مثال ساده‌ی زیر توجه کنید:
فرض کنید بانک اطلاعات شامل دو جدول stud و clg به صورت زیر باشد:

stud			clg	
S#	Sname	clg#	clg#	clgName
8621	Ali	12	12	Computer
8442	Reza	12		
8731	Abbas	NULL		

اگر اطلاعات دو جدول فوق را در یک جدول بریزیم جدول زیر به دست می‌آید:

studclg			
S#	Sname	clg#	Clgname
8621	Ali	12	Computer
8442	Reza	12	Computer
8731	Abbas	NULL	NULL

در جدول : studclg

- این اطلاعات که «کد دانشکده‌ی کامپیوتر برابر 12 است» در سطرهای اول و دوم تکرار شده است، پس افزونگی داده وجود دارد.
- اگر شخصی بخواهد کد دانشکده‌ی کامپیوتر را به 22 تغییر دهد و به اشتباه، فقط مقدار # clg در سطر اول جدول studclg را عوض کند، از این به بعد دانشکده‌ی کامپیوتر دو کد خواهد داشت: 12 و 22. این موضوع یعنی بی‌نظمی در بانک اطلاعات.
- در سطر آخر از این جدول، مقادیر NULL وجود دارد. به طور کلی، به ازای هر سطر از جدول studclg آن برابر NULL باشد یک سطر در جدول studclg خواهیم داشت که مقادیر دو ستون آخر آن NULL خواهد بود. پس مشکل مقادیر NULL نیز وجود دارد.

۲۲- گزینه (۴) صحیح است.

اگر نرم‌السازی صورت نگیرد و تمام اطلاعات در یک جدول ذخیره شود مشکلات افزونگی داده‌ها، وجود وابستگی‌های تابعی و دشوار بودن کنترل صحت و جامعیت اطلاعات وجود دارد اما مزیت آن این است که در هنگام بازیابی اطلاعات، نیازی به پیوند زدن چند جدول نیست. در طرف مقابل، با انجام نرم‌السازی، اطلاعات در چند جدول تقسیم می‌شوند و مشکلات فوق برطرف می‌شوند. اما برای بازیابی اطلاعات،

معمولًاً لازم خواهد بود چند جدول را با هم پیوند بزنیم، از آن جایی که پیوند زدن جداول عمل زمانگیری است، این عمل باعث ایجاد سربار و در نتیجه کند شدن بازیابی می شود.

۲۳- گزینه (۳) صحیح است.

با کمی بررسی می توان دید که PY کلید اصلی است. رابطه ای در فرم $2NF$ است که اگر و فقط اگر INF باشد و هر صفت غیرکلید با جزوی از کلید اصلی وابستگی تابعی نداشته باشد. از آن جا که وابستگی $\rightarrow Z$ یک وابستگی به زیر مجموعه ای از کلید اصلی (یعنی Y) است. پس این رابطه در $2NF$ نیست، بنابراین در $3NF$ و $BCNF$ هم نیست.

۲۴- گزینه (۴) صحیح است.

گزینه اول عبارت درستی است: دو عبارت زیر را در نظر بگیرید: (x یک مقدار و S یک مجموعه است):

$$x \text{ not in } S , \quad x \neq \text{all}(S)$$

عبارت سمت راست می گوید مقدار x با هیچ یک از اعضای مجموعه S برابر نیست و عبارت سمت چپ می گوید مقدار x در مجموعه S نیست. این دو عبارت با هم یکسانند.

گزینه دوم عبارت درستی است: اگر رابطه ای در $BCNF$ باشد صفت های آن فقط به کلیدهای کاندیدش وابستگی تابعی دارند، یعنی به صفت های غیرکلید وابستگی ندارند و این یعنی وابستگی انتقالی وجود ندارد، مطمئناً با به دست آوردن مجموعه پوششی مربوطه (F^+) یا با به دست آوردن هر مجموعه وابستگی دیگری معادل با مجموعه وابستگی اولیه، هیچ وابستگی انتقالی به وجود نمی آید. پس رابطه کماکان در $BCNF$ خواهد بود.

گزینه سوم عبارت نادرستی است: اگر $B \rightarrow A$ و $C \rightarrow A$ آنگاه $BC \rightarrow A$.

گزینه چهارم نیز عبارت درستی است: دو مجموعه وابستگی هم ارزند اگر مجموعه پوششی آنها برابر باشد. چون به دست آوردن مجموعه پوششی زمانبر است عدم هم ارزی F_1 و F_2 را با یک مثال نشان می دهیم:

وابستگی $B \rightarrow EC$ در F_2 به هیچ صورتی از وابستگی های موجود در F_1 قابل استنتاج نیست، چون صفت B در هیچ یک از وابستگی های F_1 در سمت راست " \rightarrow " نیامده است.

۲۵- گزینه (۴) صحیح است.

شرط $x \rightarrow R \notin F^+ : (\exists x \subseteq R)$ به این معنی است که اگر، x ای (متعلق به مجموعه صفات رابطه R موجود باشد که نتوان از آن توسط تمام وابستگی های تابعی رابطه R ، تمام صفات رابطه R را به دست آورد، این بدین معنی خواهد بود که x کلید نیست.

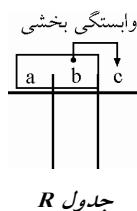
حال اگر این x کلید نباشد ولی با x بتوان به برخی صفات دیگر دسترسی پیدا کرد، این چه معنی خواهد داشت، واضح است که این معنی استنباط می شود که در دل این x صفاتی قرار دارند که وابستگی بخشی یا وابستگی انتقالی و یا وابستگی معکوس، ایجاد کرده اند.

عبارت $(x \rightarrow A \in A)$ ، بدین معنی است که، صفتی مانند A وجود داشته باشد، که از مجموعه صفات غیر کلیدی x قابل دستیابی باشد، یعنی با صفاتی از x بتوان به صفت A رسید. این بدین معنی است که صفت A به مجموعه صفات غیر کلید x وابسته است و این یعنی وجود وابستگی بخشی یا وابستگی انتقالی و یا وابستگی معکوس در رابطه R .

بنابراین اگر حاصل تفاضل $(x^+ - x^-)$ ، تهی گردد، نتیجه این است که در رابطه R وابستگی بخشی، انتقالی و معکوس وجود ندارد و اگر حاصل تفاضل $(x^+ - x^-)$ غیرتهی گردد، نتیجه این است که در رابطه R وابستگی بخشی یا انتقالی و یا معکوس وجود دارد.

بنابراین صفت ناهنجار A ، زمانی در رابطه R قرار دارد که حاصل تفاضل $(x^+ - x^-)$ غیر تهی گردد. یعنی در رابطه R وابستگی بخشی یا انتقالی یا معکوس وجود داشته باشد.

مثال: وابستگی بخشی.



وابستگی بخشی: وابستگی یک مؤلفه غیر کلیدی به جزئی از کلید کاندید را وابستگی بخشی می‌نامند.
توجه: در جدول R ، صفات ab کلید کاندید است و صفت c با صفت b وابستگی بخشی دارد. بنابراین اگر مقدار x را برابر b در نظر بگیریم (صفت b کلید نیست، پس شرط $x \rightarrow R \notin F^+$ برقرار است)، با توجه به وابستگی بخشی $c \rightarrow b$ ، آنگاه خواهیم داشت:

$$x = b$$

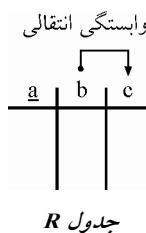
$$\{x\}^+ = \{b\}^+ = \{b, c\}$$

$$A \in (x^+ - x^-) \rightarrow (\{b, c\} - \{b\}) = \{c\}$$

حاصل تفاضل $(x^+ - x^-)$ غیرتهی شد. چون در رابطه R وابستگی بخشی وجود دارد. پس صفت c یک صفت ناهنجار است، زیرا از صفت b عضو کلید کاندید می‌توان به آن رسید.

توجه: صفت c یک صفت غیر کلید ناهنجار است.

مثال: وابستگی انتقالی.



وابستگی انتقالی: وابستگی یک مؤلفه غیر کلیدی، به یک مؤلفه غیر کلیدی دیگر را وابستگی انتقالی می‌نامند.

توجه: در جدول R ، صفت a کلید کاندید است و صفت c با صفت b وابستگی انتقالی دارد. بنابراین اگر مقدار x را برابر b در نظر بگیریم (صفت b کلید نیست، پس شرط $x \rightarrow R \notin F^+$ برقرار است)، با توجه به وابستگی انتقالی $c \rightarrow b$ ، آنگاه خواهیم داشت:

$$x = b$$

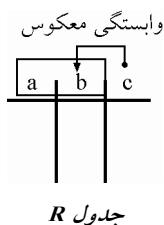
$$\{x\}^+ = \{b\}^+ = \{b, c\}$$

$$A \in (x^+ - x) \rightarrow (\{b, c\} - \{b\}) = \{c\}$$

حاصل تفاضل $(x^+ - x)$ غیر تهی شد. چون در رابطه R وابستگی انتقالی وجود دارد. پس صفت c یک صفت ناهنجار است، زیرا از صفت b غیر کلید می‌توان به آن رسید.

توجه: صفت c یک صفت غیر کلید ناهنجار است.

مثال: وابستگی معکوس.



وابستگی معکوس: وابستگی یک عضو کلید کاندید به عضو یک کلید کاندید دیگر یا مؤلفه غیر کلیدی را، وابستگی معکوس می‌نامند.

عضو کلید کاندید \rightarrow عضو کلید کاندید

عضو کلید کاندید \rightarrow غیر کلید

توجه: در جدول R , صفات ab کلید کاندید است و صفت b با صفت c وابستگی معکوس دارد.

بنابراین اگر مقدار x را برابر c در نظر بگیریم (صفت c کلید نیست، پس شرط $R \notin F^+$ $\rightarrow x$ برقرار است)، با توجه به وابستگی معکوس $b \rightarrow c$, آنگاه خواهیم داشت:

$$x = b$$

$$\{x\}^+ = \{c\}^+ = \{c, b\}$$

$$A \in (x^+ - x) \rightarrow (\{c, b\} - \{c\}) = \{b\}$$

حاصل تفاضل فوق $(x^+ - x)$ غیر تهی شد. چون در رابطه R وابستگی معکوس وجود دارد. پس صفت b یک صفت ناهنجار است، زیرا از صفت c غیر کلید می‌توان به آن رسید.

توجه مهم: صفت b یک صفت عضو کلید کاندید ناهنجار است.

از آنجا که در جدول R , صفت چند مقداری و مرکب وجود ندارد، بنابراین این جدول در نرمال فرم اول قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم اول باشد.

- جدول باید فاقد وابستگی بخشی باشد.

گزینه اول درست است.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم دوم باشد.

- جدول باید فاقد وابستگی انتقالی باشد.

مطابق گزینه اول، اگر رابطه R در 3NF باشد، آنگاه وابستگی بخشی و وابستگی انتقالی نخواهد داشت، پس هیچ صفت غیرکلیدی ناهنجاری را هم نخواهد داشت. هر چند رابطه R وابستگی بخشی و انتقالی ندارد، اما ممکن است وابستگی معکوس را داشته باشد، یعنی رابطه در نرمال فرم سوم باشد اما در BCNF نباشد. که در این حالت رابطه R دارای صفت عضو کلید کاندید ناهنجار است.

گزینه اول را مجدداً بخوانید «اگر رابطه R در 3NF باشد، آنگاه هیچ صفت غیرکلیدی در رابطه R ناهنجار نخواهد بود»، این گزینه درست است. زیرا امکان ندارد صفات غیرکلیدی در ناهنجار باشد، به دلیل عدم وابستگی بخشی و انتقالی، هر چند ممکن است صفت عضو کلید کاندید ناهنجار داشته باشد، به دلیل وجود وابستگی معکوس، اما گزینه گفته است هیچ صفت غیر کلیدی در رابطه R ناهنجار نخواهد بود، که درست است.

گزینه دوم درست است.

به طور کلی می‌توان شروط قرار داشتن یک جدول در BCNF را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم سوم باشد.

- جدول باید فاقد وابستگی معکوس باشد.

توجه: در یک نگاه دیگر برای BCNF می‌توان گفت، جدولی در BCNF قرار دارد که همه شروع‌های وابستگی‌ها، ابرکلید باشد. به بیان دیگر هر گاه سمت چپ همه وابستگی‌ها، ابرکلید باشد، آن‌گاه آن جدول در BCNF قرار دارد.

مطابق گزینه دوم، اگر رابطه R در BCNF باشد، آنگاه وابستگی بخشی، وابستگی انتقالی و وابستگی معکوس نخواهد داشت، پس هیچ صفت ناهنجاری (اعم از صفت غیرکلید ناهنجار حاصل از وابستگی بخشی و وابستگی انتقالی به دلیل عدم وجود وابستگی بخشی و انتقالی و اعم از صفت عضو کلید کاندید ناهنجار حاصل از وابستگی معکوس به دلیل عدم وجود وابستگی معکوس) نخواهد داشت. گزینه سوم درست است.

مطابق گزینه سوم اگر هیچ صفت ناهنجاری در R وجود نداشته باشد، (اعم از صفت غیر کلید ناهنجار حاصل از وابستگی بخشی و وابستگی انتقالی به دلیل عدم وجود وابستگی بخشی و انتقالی و اعم از صفت عضو کلید کاندید ناهنجار حاصل از وابستگی معکوس به دلیل عدم وجود وابستگی معکوس) این نتیجه حاصل خواهد شد که رابطه R در BCNF قرار دارد.

۲۶- گزینه (۱) صحیح است.

با کمی بررسی می‌توان دید که کلید کاندید در S برابر است با BE، برای این که جدول S در BCNF باشد باید تمام صفت‌های آن فقط به کلید کاندید وابستگی تابعی داشته باشند، گزینه‌های دوم و سوم وابستگی تابعی به غیرکلید کاندید را نشان می‌دهند. وابستگی تابعی گزینه‌ی چهارم نیز اصلاً وجود ندارد اما وابستگی تابعی گزینه‌ی اول یک وابستگی تابعی به کلید کاندید است.

۲۷- گزینه (۱) صحیح است.

در هنگام نرمال‌سازی جداول، برای این که جدول R در فرم باشد باید ابتدا برای آن، کلیدهای کاندید را به دست آورد و یکی از کلیدهای کاندید را به عنوان کلید اصلی در نظر گرفت. دو نکته در مورد کلید اصلی وجود دارد:

- ۱) تمام فیلدهای دیگر به کلید اصلی وابستگی تابعی دارند.
- ۲) حتی پس از انجام مراحل نرمال سازی (NF,...,3NF,2NF) می‌توان از روی کلید اصلی (که در یکی از جداول حاصل از نرمال سازی قرار دارد) به فیلدهای دیگر (که در جداول مختلف حاصل از نرمال سازی هستند) دسترسی پیدا کرد، این دسترسی از طریق کلیدهای خارجی صورت می‌گیرد.
- حال به جدول R_3 توجه کنید. فیلدهای آن (یعنی D و E) در هیچ جدول دیگری وجود ندارد. بنابراین بین جدول R_3 و جداول دیگر هیچ ارتباطی وجود ندارد. پس نه از صفت‌های DE می‌توان صفات دیگر را پیدا کرد و نه از صفت‌های دیگر می‌توان صفت‌های DE را یافت. پس نرمال سازی انجام شده قطعاً اشتباه است. چون هیچ کلید اصلی‌ای نمی‌توان برای رابطه‌ی اولیه‌ی R متصور شد.

۲۸- گزینه (۲) صحیح است.

گزینه‌ی اول صحیح است، با انجام نرمال سازی، افزونگی‌ها عموماً حذف می‌شوند و عموماً حجم جداول کوچک‌تر می‌شود، بنابراین جستجوها با سرعت بیشتری صورت می‌پذیرد.

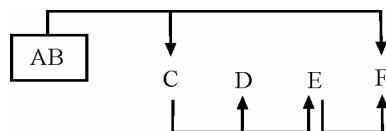
گزینه‌ی دوم غلط است، فرآیند نرمال سازی به اطلاعات درون جداول کاری ندارد و فقط براساس وابستگی‌های تعریف شده روی ستون‌های جداول عمل می‌کند پس NULL بودن مقادیر تأثیری بر فرآیند نرمال سازی ندارد.

گزینه‌ی سوم صحیح است، یکی از اهداف نرمال سازی کاهش بی‌نظمی در پایگاه داده است.

گزینه‌ی چهارم صحیح است، با انجام نرمال سازی، عموماً افزونگی حذف و در نتیجه حجم پایگاه داده کم می‌شود.

۲۹- گزینه (۱) صحیح است.

وابستگی‌های ذکر شده در شکل زیر آمده‌اند:



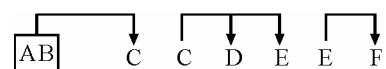
مراحل نرمال سازی به شرح زیر است:

تنها کلید کاندید، AB است:

$$\{AB\}^+ = \{A, B, C, D, E, F\}$$

پس آن را به عنوان کلید اصلی در نظر می‌گیریم. تا اینجا جدول R در 1NF هست. چون وابستگی به زیر مجموعه‌ای از کلید اصلی وجود ندارد. پس جدول R در 2NF هم هست.

برای این که جدول به 3NF برود، باید وابستگی‌های انتقالی را حذف کنیم. با انجام این کار جداول زیر به دست می‌آیند.



جداول فوق نرمال هستند و به نرمال سازی بیشتر نیاز ندارند. با توجه به جداول به دست آمده و وابستگی‌های آن‌ها گزینه اول صحیح است.

۳۰- گزینه (۲) صحیح است.

با توجه به شکل می‌توان دید که AB کلید اصلی است:

$$\{AB\}^+ = \{A, B, C, D, E\}$$

کلید کاندید دیگری نیز وجود ندارد. پس جدول در 1NF هست.

از آن جایی که وابستگی تابعی به زیر مجموعه‌ای از کلید اصلی وجود ندارد پس جدول در 2NF نیز هست.

اما چون وابستگی انتقالی زیر وجود دارد جدول در 3NF نیست:

$$AB \rightarrow E, \quad E \rightarrow C$$

۳۱- گزینه (۲) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیربدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم

$$R - \bigcup_{i=1}^n [x_i (چپ) - y_i (راست)] = \text{عضو کلید کاندید}$$

با توجه به وابستگی‌های مطرح شده برای رابطه R(A,B,C,D) داریم:

$$A \rightarrow B$$

$$B \rightarrow C$$

$$BC \rightarrow A$$

$$ABC \not\rightarrow D$$

بنابر رابطه فوق صفت D حتماً باید عضو کلید کاندید باشد. بستار صفات D به صورت زیر است:

$$\{D\}^+ = \{D\}$$

براساس بستار فوق، صفت D، فقط ستون D را تولید می‌کند، پس صفت D فقط عضو کلید کاندید می‌باشد و کلید کاندید نمی‌باشد.

قانون سوم ارسسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول، برخی از ستون‌ها را تولید کند، بدین معنی است که، جدول مورد نظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهد داشت.

با کمی دقیق و برسی، پُر واضح است که اگر صفت A در کنار عضو کلید کاندید D قرار بگیرد، همکاری صفات (D و A) می‌تواند، همه ستون‌ها را تولید کند، بنابراین صفات AD کلید کاندید جدول R نخواهد بود.

$$\{AD\}^+ = \{A, B, C, D\}$$

براساس بستار فوق، صفات AD، همه ستون‌ها را تولید می‌کنند، پس صفات AD، کلید کاندید است.

توجه: دقیق کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد.

همچنین از آن جا که $A \rightarrow BC$ و $C \rightarrow B$ ، پس می‌توان ترکیب دو خصیصه (B,D) را هم کلید کاندید دیگری برای این جدول تلقی کرد، دقیق کنید که صفت C جزء کلید کاندید نیست، زیرا توسط صفت B قابل دسترسی است.

چون وقتی (A,D) کلید کاندید است و همه ستون‌ها را تولید می‌کند، پس (B,D) هم، کلید کاندید است و همه ستون‌ها را تولید می‌کند، زیرا در نهایت طبق وابستگی $A \rightarrow BC$ و $B \rightarrow C$ ، صفت B، صفت A را می‌دهد و (B,D) به (A,D) تبدیل می‌گردد.

$$\{B,D\}^+ = \{B, D, C, A\}$$

براساس بستار فوق، صفات (B,D)، همه ستون‌ها را تولید می‌کنند، پس صفات (B,D) کلید کاندید است.

توجه: دقیق کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده، در مجموع، دارای دو کلید کاندید است که مطابق قانون سوم ارسطو، عضو کلید کاندید D در بین هر دو کلید کاندید به طور مشترک قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم اول را به صورت زیر بیان کرد:

- دارای حداقل یک کلید کاندید باشد.
- همه خصیصه‌های آن غیرقابل تجزیه باشند (جدول باید فاقد خصیصه‌های مرکب باشد)
- همه خصیصه‌های آن تک مقداری باشند (جدول باید فاقد خصیصه‌های چند مقداری باشد)

واضح است که جدول مطرح شده در فرم اول نرمال قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم اول باشد.
- جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مولفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می‌نامند.

مولفه غیرکلیدی: هر صفتی که عضو هیچ کلید کاندیدی نباشد، به عنوان مولفه غیرکلیدی نامیده می‌شود.

مولفه جزء کلید کاندید: هر صفتی که عضو حداقل یک کلید کاندید باشد، به عنوان مولفه جزء کلید کاندید نامیده می‌شود.

وابستگی‌های مطرح شده را در نظر بگیرید:

جزء کلید $A \rightarrow B$ جزء کلید $B \rightarrow C$

وابستگی بخشی: وابستگی بخشی: جزء کلید $C \rightarrow A$

در وابستگی‌های فوق، وابستگی بخشی وجود دارد. بنابراین جدول مربوطه در نرمال فرم دوم قرار ندارد، پس در نرمال فرم سوم، BCNF و نرمال فرم چهارم هم قرار ندارد.

۳۲- گزینه (۴) صحیح است.

۳۳- گزینه (۴) صحیح است.

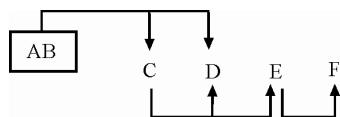
با انجام نرمال‌سازی، چند جدول به دست می‌آید که با کلیدهای خارجی به هم وصل‌اند. برای استخراج اطلاعات از روی چند جدول نیاز است جداول با هم پیوند بخورند. پیوند زدن جداول زمان‌بر است در نتیجه اگر جداول نرمال نشوند در برخی موارد سرعت پرس‌وجوها بیشتر خواهد بود. پس گزینه اول صحیح است.

از سوی دیگر، اگر بانک اطلاعاتی نرمال‌سازی نشود و تمام اطلاعات در یک جدول ذخیره شود، تعداد سطرهای این جدول بسیار زیاد خواهد بود. در نتیجه اگر به دنبال پرس‌وجوه اطلاعات از یک موجودیت (که در حالت نرمال شده یک جدول می‌بود) باشیم، باید در تعداد سطرهای زیادی به دنبال آن بگردیم در صورتی که در حالت نرمال شده، تعداد سطرهای جدول مربوط به آن موجودیت بسیار کم‌تر می‌بود، پس گزینه دوم نیز صحیح است.

گزینه سوم نیز صحیح است. ممکن است جداولی که در 3NF هستند باز هم افزونگی داشته باشند که در آن صورت از نرمال‌سازی سطح 4NF یا 5NF استفاده می‌شود.

۳۴- گزینه (۱) صحیح است.

وابستگی‌های ذکر شده در سؤال به صورت زیرند:

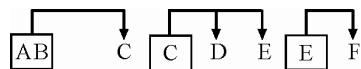


با کمی بررسی می‌توان دید که AB تنها کلید کاندید و در نتیجه کلید اصلی است:

$$\{A, B\}^+ = \{A, B, C, D, E, F\}$$

پس جدول فوق در 1NF هست.

جدول فوق در 2NF هم هست چون هیچ صفتی به زیر مجموعه‌ای از کلید اصلی وابسته نیست. ولی جدول فوق در 3NF نیست چون دو وابستگی انتقالی وجود دارد: یکی $\{AB \rightarrow C, C \rightarrow DE\}$ و دیگری $\{C \rightarrow E, E \rightarrow F\}$ با تبدیل جدول فوق به 3NF، جداول زیر به دست می‌آیند.



جدوال به دست آمده دیگر افزونگی ندارند و بنابراین به نرمال‌سازی بیشتر احتیاج نیست. با توجه به جداول به دست آمده، گزینه اول صحیح است.

۳۵- گزینه (۳) صحیح است.

در هنگام نرمال‌سازی جداول، برای این که جدول R در نرمال 1NF باشد باید ابتدا برای آن، کلیدهای کاندید را به دست آورد و یکی از کلیدهای کاندید را به عنوان کلید اصلی در نظر گرفت. دو نکته در مورد کلید اصلی وجود دارد:

- ۱) تمام فیلدهای دیگر به کلید اصلی وابستگی تابعی دارند.
- ۲) حتی پس از انجام مراحل نرمال سازی ($3NF$, $2NF$, ...) می‌توان از روی کلید اصلی (که در یکی از جداول حاصل از نرمال سازی قرار دارد) به فیلدهای دیگر (که در جداول مختلف حاصل از نرمال سازی هستند) دسترسی پیدا کرد. این دسترسی از طریق کلیدهای خارجی صورت می‌گیرد.
- حال به جداول $R2$ توجه کنید. فیلدهای آن (یعنی B و F) در هیچ جدول دیگری وجود ندارد. بنابراین بین جدول $R2$ و جداول دیگر هیچ ارتباطی وجود ندارد. پس نه از صفت‌های BF می‌توان به صفات دیگر دسترسی داشت و نه از صفت‌های دیگر می‌توان به BF دسترسی یافت. پس نرمال سازی انجام شده اشتباه است. چون هیچ کلید اصلی‌ای نمی‌توان برای رابطه اولیه R متصور شد.

۳۶- گزینه (۱) صحیح است.

این بانک اطلاعاتی در BCNF نیست چون صفت‌های آن به صفت‌هایی غیر از کلیدهای کاندید وابستگی تابعی دارند (وابستگی‌های $CD \rightarrow AB$ و $H \rightarrow AB$)

این بانک اطلاعاتی همچنین در $3NF$ نیست چون وابستگی انتقالی (یعنی وابستگی به صفت‌های غیرکلیدی) وجود دارد، (مثال همان وابستگی‌های $CD \rightarrow AB$, $AB \rightarrow H$). اما این بانک اطلاعاتی در $2NF$ هست چون صفت‌های آن به زیر مجموعه‌های کلید اصلی وابستگی تابعی ندارند (کلید اصلی در اینجا EF است چون طبق صورت سؤال، کوتاهترین کلید ممکن را کلید اصلی فرض کرده است). این بانک در $1NF$ هم هست (چون کلیدهای آن قابل تعیین‌اند و صفت‌ها از دامنه تو در تو نیستند)

توجه نمایید که طراح پاسخ مناسبی را تعیین نکرده است اگر گزینه اول فقط $2NF$ می‌بود پاسخ بهتری بود.

۳۷- گزینه (۴) صحیح است.

گزینه اول صحیح است، چون با توجه به وابستگی $BC \rightarrow A$ با داشتن A می‌توان B را به دست آورد. پس با حذف B از وابستگی $AD \rightarrow BG$ اطلاعاتی از بین نمی‌رود.

گزینه دوم صحیح است، توجه کنید که صفت‌های A و E در سمت راست هیچ یک از وابستگی‌های تابعی نیستند پس از روی صفات دیگر قابل دستیابی نیستند و بنابراین حتماً باید در کلید کاندید حضور داشته باشند. بعلاوه، با داشتن E و A می‌توان تمام صفت‌های دیگر را به دست آورد، پس $\{AE\}$ تنها کلید کمینه است و بنابراین تنها کلید کاندید است.

گزینه سوم نیز صحیح است، چون با ترکیب وابستگی‌های $BC \rightarrow A$ و $G \rightarrow BE$ می‌توان $G \rightarrow AE$ را نتیجه گرفت. اما گزینه چهارم صحیح نیست چون در رابطه داده شده، وابستگی انتقالی وجود دارد، مثلاً $G \rightarrow CD$ و $CD \rightarrow AE$ ، یا $G \rightarrow G$ و $G \rightarrow BE$ بنابراین این رابطه در $2NF$ نیست.

۳۸- گزینه (۱) صحیح است.

با توجه به شکل رسم شده، رابطه $R1$ یک رابطه‌ی یک به چند بین $E1$ و $E2$ را نشان می‌دهد به طوری که هر تاپل از $E2$ با یک تاپل از $E1$ مرتبط است و به عبارت مهم‌تر، «هر تاپل $E1$ مرتبط با چند تاپل از $E2$ است.» این عبارت بیان‌کننده یک وابستگی چند مقداری بین $E1$ و $E2$ است. ما این وابستگی را به صورت $e2 \rightarrow e1 \rightarrow e1$ نشان می‌دهیم که در آن $e1$ و $e2$ به ترتیب کلیدهای $E1$ و $E2$ هستند.

به طور مشابه، یک وابستگی چند مقداری بین $E1$ و $E3$ به صورت $e3 \rightarrow e1 \rightarrow e1$ وجود دارد. وقتی به جای ارتباط‌های $R1$ و $R2$ یک ارتباط سه گانه فرار داده شود و اگر این رابطه سه گانه را R در نظر بگیریم.

بنابر تئوری فاگین (Fagin)، تجزیه‌ای مطلوب است که:

$$\left. \begin{array}{l} R(A,B,C) \\ A \rightarrow\rightarrow B \\ A \rightarrow\rightarrow C \end{array} \right\} \Rightarrow \left. \begin{array}{l} R1(A,B) \\ R2(A,C) \end{array} \right\} \text{فرم نرمال 4NF}$$

بنابراین با توجه به دو وابستگی چندمقداری $e2 \rightarrow\rightarrow e1 \rightarrow\rightarrow e3$ و $e1 \rightarrow\rightarrow e2 \rightarrow\rightarrow e3$ تجزیه مطلوب در فرم نرمال 4NF به صورت زیر خواهد بود:

$$R1(e1,e2), \quad R2(e1,e3)$$

باید کلیدهای سه موجودیت $E1, E2$ و $E3$ به جدول متناظر با این رابطه سه گانه اضافه شود. بنابراین در این جدول، دو وابستگی چند مقداری $e2 \rightarrow\rightarrow e1 \rightarrow\rightarrow e3$ و $e1 \rightarrow\rightarrow e2 \rightarrow\rightarrow e3$ وجود خواهد داشت. برای رفع این وابستگی‌های چند مقداری باید جدول را به 4NF برد که البته برای این کار باید وابستگی‌های تابعی و چند مقداری را دانست.

۳۹- گزینه (۳) صحیح است.

قبل از هر چیز توجه کنید که W, X, Y و Z مجموعه‌هایی از صفات $A1$ تا An هستند، بنابراین از شرط $Z \subseteq W$ در گزاره اول می‌توان نتیجه گرفت که $W \rightarrow Z$ ، یعنی از روی مجموعه صفات W می‌توان صفات موجود در مجموعه Z را نتیجه گرفت. برای مثال اگر $W = \{A1, A2, A3\}$ و $Z = \{A1, A2\}$ باشند (و در نتیجه $Z \subseteq W$ باشد) بدیهی است که داریم:

$$\{A1, A2, A3\} \rightarrow \{A1, A2\}$$

یعنی از روی صفات $\{A1, A2, A3\}$ می‌توان صفات $\{A1, A2\}$ را به دست آورد.

با توجه به توضیحات فوق، گزاره اول اگر $Y \rightarrow X$ می‌بود و همچنین داریم $Z \subseteq W$ که از روی آن می‌فهمیم که $Z \rightarrow W$. بنابراین طبق قاعده ترکیب داریم:

$$\left. \begin{array}{l} X \rightarrow Y \\ W \rightarrow Z \end{array} \right\} \Rightarrow WX \rightarrow YZ$$

گزاره دوم اگر $Y \rightarrow X$ می‌بود، از سوی دیگر از $Z \subseteq Y$ در می‌یابیم که $Y \rightarrow Z$ بنابراین طبق قاعده انتقال داریم:

$$\left. \begin{array}{l} X \rightarrow Y \\ Y \rightarrow Z \end{array} \right\} \Rightarrow X \rightarrow Z$$

اما در گزاره اول داریم:

$$X \rightarrow\rightarrow Y, \quad Z \subseteq W \Rightarrow WX \rightarrow YZ$$

مطابق قاعده افزایشی داریم:

$$\left. \begin{array}{l} X \rightarrow\rightarrow Y \\ Z \subseteq W \end{array} \right\} \quad \left. \begin{array}{l} X \rightarrow\rightarrow Y \\ W \rightarrow Z \end{array} \right\} \Rightarrow XW \rightarrow YZ$$

بنابراین گزاره اول عبارت درستی است.

همچنین در گزاره دوم داریم:

$$X \rightarrow\rightarrow Y, \quad W \cap Y, \quad W \rightarrow Z, \quad Z \subseteq Y \Rightarrow X \rightarrow Z$$

مطلوب قاعده انعقاد داریم:

$$\left. \begin{array}{l} X \rightarrow\rightarrow Y \\ W \cap Y = \emptyset \\ W \rightarrow Z \\ Z \subseteq Y \end{array} \right\} X \rightarrow Z$$

بنابراین گزاره دوم هم عبارت درستی است.

۴۰- گزینه (۲) صحیح است.

برای پاسخگویی به این سؤال باید ابتدا مجموعه وابستگی پوششی FD^+ و سپس مجموعه وابستگی بهینه FD_{opt} را به دست آوریم و بعد از آن مراحل نرمال‌سازی را انجام دهیم. با کمی بررسی دیده می‌شود که مجموعه داده شده همان FD^+ است. پس با انجام قواعد لازم، FD_{opt} به صورت زیر به دست می‌آید:

$$FD_{opt} = \{A \rightarrow D, D \rightarrow E, AB \rightarrow C\}$$

با توجه به FD_{opt} مشاهده می‌شود که تنها کلید کاندید AB است. تا اینجا، جدول مربوطه در 1NF است. با بردن جداول به 2NF (یعنی با حذف وابستگی‌های تابعی به زیر مجموعه‌های کلید اصلی) دو جدول زیر، با وابستگی‌های تابعی نشان داده شده به دست می‌آیند.



جدول 1 را نمی‌توان بیش از این نرمال کرد اما جدول 2 وابستگی انتقالی دارد و باید به 3NF برود:



بنابراین سه جدول (1) و (3) و (4) به دست می‌آیند که در 3NF هستند و بیش از این قابل نرمال‌سازی نیستند. البته جداول در سطح BCNF نیز می‌باشند.

۴۱- گزینه (۳) صحیح است.

در مورد گزینه اول، با توجه به وابستگی‌های تابعی داده شده، با کمی بررسی می‌توان دید که از پیوند طبیعی سه جدول R_3, R_2, R_1 جدول R به دست می‌آید.

در مورد گزینه دوم، چون کلید اصلی جدول R (یعنی A) در هر سه جدول R_3, R_2, R_1 آمده است از پیوند طبیعی سه جدول مذکور جدول R به دست می‌آید.

اما در مورد گزینه سوم، نمی‌توان با پیوند طبیعی R_3, R_2, R_1 به جدول R رسید.

یکی از شرایط در تجزیه بدون گمشدگی (LossLess) اینست که صفت مشترک در یکی از روابط کلید باشد که گزینه سوم این شرط را دارا نیست.

توجه: اگر سؤال، تجزیه مطلوب را در نظر داشت، گزینه اول، پاسخ می‌بود، زیرا هم الحق پذیر است و هم وابستگی‌های تابعی جدول پایه را حفظ کرده است.

۴۲- گزینه (۴) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می آید:

قانون اول ارسطو

روش اول

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیر بدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم

$$R - \sum_{i=1}^n [x_i(y_i - \text{راست})] = \text{عضو کلید کاندید}$$

با توجه به وابستگی‌های مطرح شده برای رابطه:

Booking (guestID, guestName, creditCard, roomNo, roomCat, from, to)

داریم:

guestID → questName, creditCard

roomNo → roomCat

roomNo, from → guestID, to

roomNo, to → guestID, from

guestID, guestName, creditCard, roomNo, roomCat, from, to –

guestID, guestName, creditCard, roomCat, from, to = roomNo

بنابر رابطه فوق صفت roomNo حتماً باید عضو کلید کاندید باشد.

بستار صفت roomNo به صورت روبرو است:

قانون سوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول، برخی از ستون‌ها را تولید کند، بدین معنی است که جدول موردنظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهد داشت. بنابراین صفت roomCat که توسط عضو کلید کاندید یعنی roomNo تولید مم‌گ دد، در کلید جایگاهی نخواهد داشت.

براساس بستار فوق، صفت roomNo به عنوان عضو کلید کاندید همه ستون‌ها را تولید نمی‌کند، بنابراین مطابق قانون سوم ارسطو باید صفاتی در کنار صفت roomNo قرار گیرد تا کلید کاندید تولید گردد. این صفات کناری از صفات باقی مانده انتخاب می‌گردند، صفات باقی مانده عبارتند از guestID, guestName, creditCard, from, to کنار می‌گذاریم، زیرا کمکی در تولید صفات دیگر نمی‌کنند. چون در سمت چپ هیچ یک از وابستگی‌های تابعی نیامده است، بنابراین با ترکیب صفات from یا to با صفت roomNo کلیدهای کاندید تولید می‌گردد.

$$\{roomNo, from\}^+ = \{roomNo, from, guestID, to, guestName, creditCard, roomCat\}$$

$$\{roomNo, to\}^+ = \{roomNo, to, guestID, from, guestName, creditCard, roomCat\}$$

پس ترکیب صفات $\{roomNo, to\}$ و $\{roomNo, from\}$ کلیدهای کاندید جدول Booking هستند.

توجه: دقت کنید که ترکیب صفات $\{roomNo, guestID\}$ نمی‌تواند به عنوان کلید کاندید محسوب گردد، چون همه صفات را تولید نمی‌کند. بستار صفات $\{roomNo, guestID\}$ به صورت زیر است:

$$\{roomNo, guestID\}^+ = \{roomNo, guestID, guestName, creditCard, roomCat\}$$

توجه: کلید اصلی، یکی از کلیدهای کاندید است که بر اساس سلیقه طراح پایگاه داده، انتخاب می‌شود. مطابق قوانین مدل رابطه‌ای، هر جدول حتماً باید یک کلید اصلی داشته باشد. همچنین هر جدول فقط می‌تواند یک کلید اصلی داشته باشد. هر کلید کاندیدی که به عنوان کلید اصلی انتخاب نشده است، یک کلید فرعی است. فرض کنید که ترکیب صفات $\{roomNo, from\}$ کلید اصلی و ترکیب صفات $\{roomNo, to\}$ کلید فرعی جدول Booking باشد.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده در مجموع دارای دو کلید کاندید است. که مطابق قانون سوم ارسطو، عضو کلید کاندید roomNo، در بین هر دو کلید کاندید به طور مشترک قرار دارد.

حال یکبار دیگر وابستگی‌های مطرح شده برای رابطه:

Booking (guestID, guestName, creditCard, roomNo, roomCat, from, to)

را در نظر بگیرید:

غیرکلید guestID → questName غیرکلید وابستگی انتقالی

غیرکلید guestID → creditCard غیرکلید وابستگی انتقالی

غیرکلید roomNo → roomCat عضو کلید کاندید وابستگی بخشی

غیرکلید roomNo, from → guestID کلید کاندید وابستگی کامل

عضو کلید کاندید roomNo, from → to کلید کاندید وابستگی کامل

غیرکلید roomNo, to → guestID کلید کاندید وابستگی کامل

عضو کلید کاندید roomNo, to → from کلید کاندید وابستگی کامل

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم اول را به صورت زیر بیان کرد:

- دارای حداقل یک کلید کاندید باشد.

- همه خصیصه‌های آن غیرقابل تجزیه باشند (جدول باید فاقد خصیصه‌های مرکب باشد)

- همه خصیصه‌های آن تک مقداری باشند (جدول باید فاقد خصیصه‌های چند مقداری باشد)

واضح است که جدول مطرح شده در فرم اول نرمال قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم اول باشد.

- جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مولفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می‌نامند.

مولفه غیر کلید: هر صفتی که عضو هیچ کلید کاندیدی نباشد، به عنوان مؤلفه غیر کلیدی نامیده می‌شود.
مؤلفه جزء کلید کاندید: هر صفتی که عضو حداقل یک کلید کاندید باشد، به عنوان مؤلفه جزء کلید نامیده می‌شود.

در واستگی‌های فوق، واستگی بخشی وجود دارد. بنابراین جدول مربوطه در نرمال فرم دوم قرار ندارد.

وابستگی بخشی roomCat——roomNo غیر کلید عضو کلید کاندید

به طور کلی می‌توان شرط قرار داشتن یک جدول در نرم‌افزار 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 غیر کلیدی به یک مؤلفه غیر کلیدی دیگر را وابستگی انتقالی می‌نامند. در وابستگی‌های فوق، وابستگی بخشی و انتقالی وجود دارد. بنابراین جدول مربوطه در نرم‌ال فرم سوم هم قرار ندارد.

وابستگی بخشی roomNo → roomCat عضو کلید کاندید غیرکلید

وابستگی انتقالی guestID——questName غیر کلید غیر کلید

وابستگی انتقالی guestID → creditCard → غير كليد

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم BCNF را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم سوم باشد.
 - جدول باید فاقد وایستگی معکوس باشد

وابستگی معکوس: ابستگی یک عضو کلید کاندید به عضو یک کلید کاندید دیگر یا مؤلفه غیر کلیدی را،
وابستگی معکوس می نامند.

عضو کلید کاندید → عضو کلید کاندید

عضو کلید کاندید → غیرکلید

در وابستگی های فوق، وابستگی معکوس وجود ندارد. اما وابستگی های بخشی و انتقالی وجود دارد، یعنی همانطور که گفته شد جدول مربوطه در نرمال فرم دوم و سوم قرار ندارد، بنابراین جدول مربوطه در نرمال فرم هم CBNF قرار ندارد.

در یک نگاه دیگر برای نرم‌ال فرم BCNF می‌توان گفت، جدولی در نرم‌ال فرم BCNF قرار دارد که همه شروع‌های وابستگی‌ها، ابرکلید باشد. به بیان دیگر هرگاه سمت چپ همه وابستگی‌ها، ابرکلید باشد، آن‌گاه

ان جدول در BCNF قرار دارد که در وابستگی های فوق این چنین نیست. پس BCNF هم نیست.
در وابستگی های فوق، سمت چپ وابستگی های کامل، ابر کلید است. اما سمت چپ وابستگی های بخشی و
انتقالی، ابر کلید نیست.

بنابراین این جدول به دلیل نقض شرایط مربوطه، در نرمال فرم BCNF قرار ندارد. در نتیجه گزینه اول عبارت درستی است و به تبع پاسخ سوال هم نیست.

همچنین دو وابستگی تابعی سوم و چهارم مطرح شده در صورت سوال یعنی، وابستگی‌های زیر:

وابستگی کامل roomNo,from guestID کلید کاندید غیر کلید

عضو کلید کاندید to → roomNo,from کلید کاندید و ایستگی کاما

غیر کلید roomNo,to → guestID کلید کاندید وابستگی کامل
 عضو کلید کاندید roomNo,to → from کلید کاندید باعث نقض شدن نرمال فرم سوم یعنی 3NF نمی شود. زیرا وابستگی های بخشی و انتقالی باعث نقض شدن نرمال فرم سوم یعنی 3NF می شود. یعنی وابستگی های اول و دوم مطرح شده در صورت سوال یعنی، وابستگی های زیر:
 غیر کلید questName → guestID غیر کلید وابستگی انتقالی
 غیر کلید guestID → creditCard غیر کلید وابستگی انتقالی
 غیر کلید roomCat → roomNo عضو کلید کاندید وابستگی بخشی در نتیجه گزینه دوم عبارت درستی است و به تبع پاسخ سوال هم نیست.
 در وابستگی های فوق، وابستگی بخشی roomCat → roomNo وجود دارد، بنابراین جدول مربوطه در نرمال فرم دوم قرار ندارد، بنابراین رویه نرمال سازی (تجزیه مطلوب) جهت قرار گیری جدول Booking در نرمال فرم دوم توسط عمل تجزیه جدول Booking می باشد انجام گردد.

شرط تجزیه مطلوب به صورت زیر است:

۱- شرط لازم (Nonloss Join): دو جدول الحاق پذیر باشند (صفت مشترک در دو جدول، حداقل در یکی کلید کاندید باشد)

شرط الحاق پذیری به دو شکل زیر انجام می گردد:

۱-۱- برای حذف وابستگی بخشی از تنوری هیث استفاده می گردد.

تنوری هیث: این تنوری می گوید، وابستگی بخشی (وابستگی غیر کلید به جز کلید کاندید) را از جدول پایه خارج کنید.

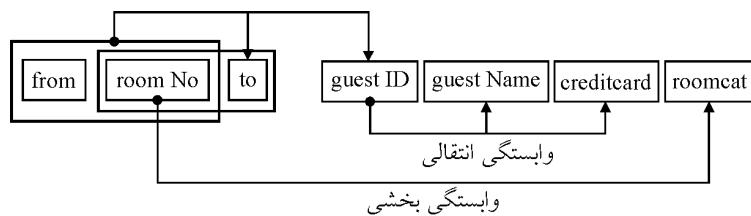
۱-۲- برای حذف وابستگی انتقالی از تنوری ریسانن استفاده می گردد.

تنوری ریسانن: این تنوری می گوید، وابستگی انتقالی (وابستگی غیر کلید به غیر کلید) را از جدول پایه خارج کنید.

۲- شرط کافی (حفظ وابستگی های تابعی): وابستگی های تابعی جدول پایه قابل استنتاج از روی وابستگی های تابعی جداول حاصل از تجزیه باشد یا در آنها موجود باشد.
 با توجه به نمودار وابستگی های جدول:

Booking (guestID, guestName, creditCard, roomNo, roomCat, from, to)

داریم:



توجه: کلید اصلی، یکی از کلیدهای کاندید است که بر اساس سلیقه طراح پایگاه داده، انتخاب می شود. مطابق قوانین مدل رابطه‌ای، هر جدول حتماً باید یک کلید اصلی داشته باشد. همچنین هر جدول فقط می‌تواند یک کلید اصلی داشته باشد. هر کلید کاندیدی که به عنوان کلید اصلی انتخاب نشده است، یک کلید فرعی است. فرض کنید که ترکیب صفات $\{roomNo, from\}$ کلید اصلی و ترکیب صفات $\{roomNo, to\}$ کلید فرعی جدول Booking باشد.

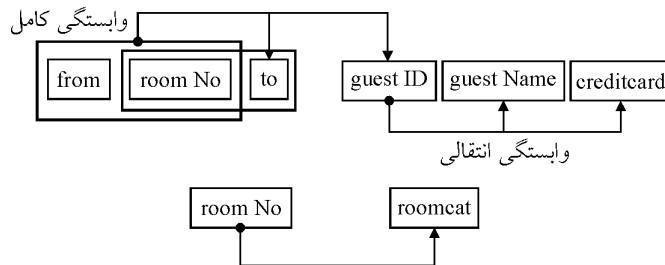
به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم اول باشد.

- جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مؤلفه غیرکلیدی، به جزیی از کلید کاندید را وابستگی بخشی می‌نامند. با توجه به این که ترکیب صفات $\{roomNo, to\}$ و $\{roomNo, from\}$ به عنوان کلیدهای کاندید رابطه Booking معروفی شده است، وابستگی تابعی زیر:

وابستگی بخشی غیرکلید roomNo \longrightarrow roomCat عضو کلید کاندید
به عنوان وابستگی بخشی محسوب می‌گردد. بنابراین جداول Booking در نرمال فرم دوم قرار ندارد. بنابراین برای قرار دادن جداول Booking، در نرمال فرم دوم، باید وابستگی بخشی از جداول Booking خارج گردد.



بنابراین جدول Booking به دو جدول کوچک‌تر تجزیه می‌گردد:

R1 (from, roomNo, to, guestID, guestName, creditCard)

R2 (roomNo, roomCat)

با توجه به کلید کاندید جدول R1 می‌توان دریافت که این جدول فاقد وابستگی بخشی است، بنابراین این جدول در نرمال فرم دوم قرار دارد. همچنین با توجه به کلید کاندید جدول R2 می‌توان دریافت که این جدول فاقد وابستگی بخشی و انتقالی است، بنابراین این جدول در نرمال فرم دوم و سوم قرار دارد. همچنین از آنجا که سمت چپ تمام وابستگی‌های جدول R2 ابرکلید است، بنابراین جدول R2 در سطح BCNF هم قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم دوم باشد.

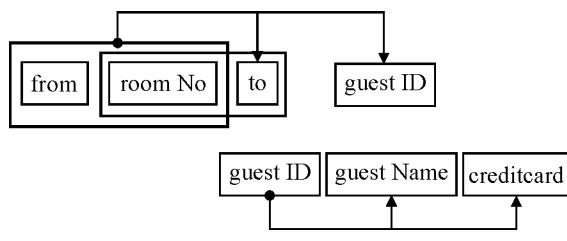
- جدول باید فاقد وابستگی انتقالی باشد.

وابستگی انتقالی: وابستگی یک مؤلفه غیرکلیدی به یک مؤلفه غیرکلیدی دیگر را وابستگی انتقالی می‌نامند.

در جدول R1 وابستگی تابعی زیر:

غیر کلید guestID \longrightarrow questName, creditCard وابستگی انتقالی

به عنوان وابستگی انتقالی محسوب می‌گردد. بنابراین جدول R1 در نرمال فرم سوم قرار ندارد.
بنابراین برای قرار دادن R1، در نرمال فرم سوم، باید وابستگی انتقالی از جدول R1 خارج گردد.



بنابراین جدول R1 به جداول کوچک‌تر تجزیه می‌گردد:

R11(from, roomNo, to, guestID)

R12(guestID, guestName, creditCard)

با توجه به کلیدهای کاندید در جداول R11 و R12 می‌توان دریافت که این جداول فاقد وابستگی بخشی و انتقالی هستند و بنابراین در نرمال فرم دوم و سوم قرار دارند. هم‌چنین از آن‌جا که سمت چپ تمام وابستگی‌های جداول R11 و R12 ابرکلید است، بنابراین جداول R11 و R12 در سطح BCNF نیز قرار دارند. به این ترتیب با تجزیه جدول Booking به سه جدول R11، R12 و R2 می‌توان جدول Booking را در نرمال فرم دوم، سوم و سطح BCNF قرار داد.

توجه: آنچه تا به اینجا مطرح شد، مربوط به تجزیه مطلوب جدول Booking بود.

توجه: در گزینه‌ی سوم مطرح شده است که، اگر رابطه Booking به سه رابطه:

R1(guestID, roomNo, from, to) -۱

R2(roomNo, roomCat) -۲

R3(guestName, creditCard, roomNo, from) -۳

تجزیه شود، هر سه رابطه در 3NF و BCNF هستند و تجزیه بدون اتلاف (lossless decomposition) است. در واقع گزینه‌ی سوم به دنبال تجزیه‌ی مطلوب و برقراری شرط لازم و شرط کافی نیست. بلکه فقط برقراری شرط لازم یعنی تجزیه بدون اتلاف (lossless decomposition) و به تبع تجزیه نامطلوب را مدنظر قرار داده است. دقت کنید که تجزیه مطلوب پیش‌تر بررسی شد. شرط لازم به صورت زیر است:

شرط لازم (Nonloss Join): دو جدول الحاق‌پذیر باشند (صفت مشترک در دو جدول، حداقل در یکی کلید کاندید باشد)

با توجه به وابستگی‌های مرتبط برای رابطه R1(guestID, roomNo, from, to) R1 داریم:

roomNo, from \longrightarrow guestID

roomNo, from \longrightarrow to

roomNo, to \longrightarrow guestID

roomNo, to \longrightarrow from

guestID, roomNo, from, to = roomNo

بنابر رابطه فوق صفت roomNo حتماً باید عضو کلید کاندید باشد.

بستار صفت roomNo برای جدول R1(guestID,roomNo,from,to) به صورت زیر است:

$$\{roomNo\}^+ = \{roomNo\}$$

قانون سوم ارسسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول، برخی از ستون‌ها را تولید کند، بدین معنی است که، جدول موردنظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت.

براساس بستار فوق، صفت roomNo به عنوان عضو کلید کاندید همه ستون‌ها را تولید نمی‌کند، بنابراین مطابق قانون سوم ارسسطو باید صفاتی در کتاب صفات roomNo قرار گیرد تا کلید کاندید تولید گردد. این صفات کناری از صفات باقی مانده انتخاب می‌گردد، صفات باقی مانده عبارتند از guestID,from,to البته از این مجموعه صفات guestID را هم کنار می‌گذاریم، زیرا کمکی در تولید صفات دیگر نمی‌کند. چون در سمت چپ هیچ یک از وابستگی‌های تابعی نیامده است. بنابراین با ترکیب صفات from یا to با صفت roomNo کلیدهای کاندید تولید می‌گردد.

$$\{roomNo,from\}^+ = \{roomNo,from,guestID,to\}$$

$$\{roomNo,to\}^+ = \{roomNo,to,guestID,from\}$$

پس ترکیب صفات $\{roomNo,to\}$ و $\{roomNo,from\}$ کلیدهای کاندید جدول R1 هستند.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه R1 در مجموع دارای دو کلید کاندید است. که مطابق قانون سوم ارسسطو، عضو کلید کاندید roomNo، در بین هر دو کلید کاندید به طور مشترک قرار دارد.

حال یکبار دیگر وابستگی‌های مطرح شده برای رابطه R1(guestID,roomNo,from,to) را در نظر بگیرید:

وابستگی کامل roomNo,from————>guestID غیرکلید roomNo,from————>guestID کلید کاندید

وابستگی کامل roomNo,from————>to عضو کلید کاندید roomNo,from————>to کلید کاندید

وابستگی کامل roomNo,to————>guestID غیرکلید roomNo,to————>guestID کلید کاندید

وابستگی کامل roomNo,to————>from عضو کلید کاندید roomNo,to————>from کلید کاندید

در وابستگی‌های فوق، وابستگی بخشی، انتقالی و معکوس وجود ندارد. بنابراین جدول R1 در نرم‌ال فرم

دوم، سوم و BCNF قرار دارد.

با توجه به وابستگی‌های مرتبط برای رابطه R2(roomNo,roomCat) R داریم:

roomNo————>roomCat

roomNo,roomCat – roomCat = roomNo

بنابر رابطه فوق صفت roomNo حتماً باید عضو کلید کاندید باشد.

بستار صفت roomNo برای جدول R2(roomNo,roomCat) به صورت زیر است:

$$\{roomNo\}^+ = \{roomNo, roomCat\}$$

قانون دوم ارسسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روشن اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

حال یکبار دیگر وابستگی‌های مطرح شده برای رابطه R2(roomNo,roomCat) را در نظر بگیرید:

وابستگی کامل roomNo → roomCat غیرکلید در وابستگی‌های فوق، وابستگی بخشی، انتقالی و معکوس وجود ندارد. بنابراین جدول R2 در نرمال فرم دوم، سوم و BCNF قرار دارد.

با توجه به وابستگی‌های مرتبط برای رابطه R3(guestName,creditCard,roomNo,from) داریم:

$$\{\}$$

در وابستگی‌های فوق، وابستگی بخشی، انتقالی و معکوس وجود ندارد. بنابراین جدول R3 در نرمال فرم دوم، سوم و BCNF قرار دارد. در واقع جدول R3 یک جدول تمام کلید است. و یک جدول تمام کلید همواره در سطح BCNF نرمال قرار دارد.

توجه: همانطور که گفته‌یم، در گزینه‌ی سوم مطرح شده است که، اگر رابطه Bookin به سه رابطه:

$$R1(\text{guestID}, \text{roomNo}, \text{from}, \text{to}) \quad -1$$

$$R2(\text{roomNo}, \text{roomCat}) \quad -2$$

$$R3(\text{guestName}, \text{creditCard}, \text{roomNo}, \text{from}) \quad -3$$

تجزیه شود، هر سه رابطه در 3NF و BCNF هستند و تجزیه بدون اتلاف (lossless decomposition) است. در واقع گزینه‌ی سوم به دنبال تجزیه‌ی مطلوب و برقراری شرط لازم و شرط کافی نیست. بلکه فقط برقراری شرط لازم یعنی تجزیه بدون اتلاف (lossless decomposition) و به تبع تجزیه نامطلوب را مدنظر قرار داده است. وقت کنید که تجزیه مطلوب پیش‌تر بررسی شد. شرط لازم به صورت زیر است:
شرط لازم (Nonloss Join): دو جدول الحقیقی باشند (صفت مشترک در دو جدول، حداقل در یکی کلید کاندید باشد)

در گزینه‌ی سوم صفت مشترک در دو جدول، حداقل در یکی کلید کاندید است، بنابراین شرط لازم برقرار است و تجزیه بدون اتلاف (lossless decomposition) است. به صورت زیر:

$$R1(\text{guestID}, \text{roomNo}, \text{from}, \text{to})$$

$$R2(\text{roomNo}, \text{roomCat})$$

$$R3(\text{guestName}, \text{creditCard}, \text{roomNo}, \text{from})$$

همچنین همانطور که گفته‌یم هر سه جدول R1، R2 و R3 مطرح شده در گزینه‌ی سوم، در نرمال فرم دوم، سوم و BCNF قرار دارند. در نتیجه گزینه سوم عبارت درستی است و به تبع پاسخ سوال هم نیست.

توجه: در گزینه‌ی چهارم مطرح شده است که، اگر رابطه Booking به سه رابطه:

$$R1(\text{guestID}, \text{guestName}, \text{creditCard}) \quad -1$$

و $R2(\text{roomNo}, \text{roomCat})$ -۲

$R3(\text{guestID}, \text{roomNo}, \text{from,to})$ -۳

تجزیه شود، هر سه رابطه در 3NF هستند ولی در BCNF نیستند. در واقع گزینه‌ی چهارم به دنبال تجزیه‌ی مطلوب و برقراری شرط لازم و شرط کافی نیست. بلکه فقط برقراری سطوح نرمال را مدد نظر قرار داده است.

با توجه به وابستگی‌های مرتبط برای رابطه $R1(\text{guestID}, \text{guestName}, \text{creditCard})$ داریم:

$\text{guestID} \longrightarrow \text{guestName}, \text{creditCard}$

$\text{guestID}, \text{guestName}, \text{creditCard} - \text{guestName}, \text{creditCard} = \text{guestID}$

بنابر رابطه فوق صفت guestID حتماً باید عضو کلید کاندید باشد.

بستار صفت guestID برای جدول $R1(\text{guestID}, \text{guestName}, \text{creditCard})$ به صورت زیر است:

$$\{\text{guestID}\}^+ = \{\text{guestID}, \text{guestName}, \text{creditCard}\}$$

قانون دوم ارسسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.
حال یکبار دیگر وابستگی‌های مطرح شده برای رابطه $R1(\text{guestID}, \text{guestName}, \text{creditCard})$ را در نظر بگیرید:

وابستگی کامل $\text{guestID} \longrightarrow \text{guestName}, \text{creditCard}$ کلید کاندید
در وابستگی فوق، وابستگی بخشی، انتقالی و معکوس وجود ندارد. بنابراین جدول $R1$ در نرمال فرم دوم، سوم و BCNF قرار دارد.

با توجه به وابستگی‌های مرتبط برای رابطه $R2(\text{roomNo}, \text{roomCat})$ داریم:

$\text{roomNo} \longrightarrow \text{roomCat}$

$\text{roomNo}, \text{roomCat} - \text{roomCat} = \text{roomNo}$

بنابر رابطه فوق صفت roomNo حتماً باید عضو کلید کاندید باشد.

بستار صفت roomNo برای جدول $R2(\text{roomNo}, \text{roomCat})$ به صورت زیر است:

$$\{\text{roomNo}\}^+ = \{\text{roomNo}, \text{roomCat}\}$$

قانون دوم ارسسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

حال یکبار دیگر وابستگی‌های مطرح شده برای رابطه $R2(\text{roomNo}, \text{roomCat})$ را در نظر بگیرید:

وابستگی کامل $\text{roomNo} \longrightarrow \text{roomCat}$ کلید کاندید
در وابستگی‌های فوق، وابستگی بخشی، انتقالی و معکوس وجود ندارد. بنابراین جدول $R2$ در نرمال فرم دوم، سوم و BCNF قرار دارد.

با توجه به وابستگی های مرتبط برای رابطه $R3(guestID, roomNo, from, to)$ داریم:

$roomNo, from \longrightarrow guestID$

$roomNo, from \longrightarrow to$

$roomNo, to \longrightarrow guestID$

$roomNo, to \longrightarrow from$

$guestID, roomNo, from, to - guestID, from, to = roomNo$

بنابر رابطه فوق صفت $roomNo$ حتماً باید عضو کلید کاندید باشد.

بستار صفت $roomNo$ برای جدول $R3(guestID, roomNo, from, to)$ به صورت زیر است:

$$\{roomNo\}^+ = \{roomNo\}$$

قانون سوم ارسسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول، برخی از ستون ها را تولید کند، بدین معنی است که، جدول موردنظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلید های کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت.

براساس بستار فوق، صفت $roomNo$ به عنوان عضو کلید کاندید همه ستون ها را تولید نمی کند، بنابراین مطابق قانون سوم ارسسطو باید صفاتی در کنار صفت $roomNo$ قرار گیرد تا کلید کاندید تولید گردد. این صفات کناری از صفات باقی مانده انتخاب می گردند، صفات باقی مانده عبارتند از $guestID, from, to$ البته از این مجموعه صفات $guestID$ را هم کنار می گذاریم، زیرا کمکی در تولید صفات دیگر نمی کند. چون در سمت چپ هیچ یک از وابستگی های تابعی نیامده است. بنابراین با ترکیب صفات $from$ یا to با صفت $roomNo$ کلید های کاندید تولید می گردد.

$$\{roomNo, from\}^+ = \{roomNo, from, guestID, to\}$$

$$\{roomNo, to\}^+ = \{roomNo, to, guestID, from\}$$

پس ترکیب صفات $\{roomNo, to\}$ و $\{roomNo, from\}$ کلید های کاندید جدول $R3$ هستند.

توجه: دقت کنید که هیچ گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه $R3$ در مجموع دارای دو کلید کاندید است. که مطابق قانون سوم ارسسطو، عضو کلید کاندید $roomNo$ ، در بین هر دو کلید کاندید به طور مشترک قرار دارد.

حال یکبار دیگر وابستگی های مطرح شده برای رابطه $R3(guestID, roomNo, from, to)$ را در نظر بگیرید:

$roomNo, from \longrightarrow guestID$ غیر کلید

وابستگی کامل

$roomNo, from \longrightarrow to$ عضو کلید کاندید

وابستگی کامل

$roomNo, to \longrightarrow guestID$ غیر کلید

وابستگی کامل

$roomNo, to \longrightarrow from$ عضو کلید کاندید

وابستگی کامل

در وابستگی‌های فوق، وابستگی بخشی، انتقالی و معکوس وجود ندارد. بنابراین جدول R3 در نرمال فرم دوم، سوم و BCNF قرار دارد.

توجه: همانطور که گفتیم، در گزینه‌ی چهارم مطرح شده است که، اگر رابطه Booking به سه رابطه:

R1 (guestID, guestName, creditCard) -۱

R2 (roomNo, roomCat) -۲

R3 (guestID, roomNo, from, to) -۳

تجزیه شود، هر سه رابطه در 3NF هستند. اما همانطور که گفتیم هر سه جدول R1 و R2 و R3 مطرح شده در گزینه‌ی چهارم، در نرمال فرم دوم، سوم و BCNF قرار دارند. در نتیجه گزینه چهارم عبارت نادرستی است و به تبع پاسخ سوال است.

۴۳- گزینه (۳) صحیح است.

با کمی بررسی می‌توان دید تنها کاندید DGH است. پس گزینه دوم صحیح است. برای به دست آوردن یک ابرکلید باید از بین پنج صفتی که در کلید کاندید نیستند تعداد صفر، یک، دو، ... یا پنج صفت را به دلخواه انتخاب و به صفات DGH اضافه کنیم. تعداد انتخاب‌های ممکن به صورت زیر به دست می‌آید:

$$\binom{5}{0} + \binom{5}{1} + \binom{5}{2} + \binom{5}{3} + \binom{5}{4} + \binom{5}{5} = 2^5 = 32$$

پس این رابطه 32 ابرکلید دارد. پس گزینه اول صحیح است.

رابطه داده شده در 2NF نیست چون وابستگی تابعی \rightarrow D یک وابستگی به زیر مجموعه‌ای از صفات کلید اصلی است. پس گزینه سوم غلط است.

چون رابطه داده شده در 2NF نیست پس در 3NF هم نیست (البته وابستگی انتقالی $F \rightarrow C$ و $C \rightarrow F$ هم وجود دارد). پس گزینه چهارم صحیح است.

۴۴- گزینه (۲) صحیح است.

این رابطه در 1NF هست چون همه کلیدهای آن معلوم‌اند، تمام صفت‌ها به کلید اصلی وابستگی تابعی دارند (هر یک از دو کلید کاندید را به دلخواه کلید اصلی فرض کنید)، و صفت‌های آن از دامنه‌های تو در تو (چند مقداری و مرکب) نیستند.

این رابطه هم‌چنین در 2NF هست چون هیچ یک از صفت‌های غیرکلیدی آن به زیر مجموعه‌ای از کلید کاندید وابسته نیستند. (وابستگی بخشی ندارد).

این رابطه هم‌چنین در 3NF هست چون صفت‌های غیرکلیدی آن (یعنی D و E) به هم وابستگی ندارند. (وابستگی انتقالی ندارد).

هم‌چنین این جدول در BCNF هم هست چون ستون‌های آن فقط به کلیدهای کاندید وابستگی تابعی دارند. هرگاه همه شروع‌های مجموعه وابستگی‌ها، ابرکلید بودند، آن جدول BCNF است. به بیان دیگر هرگاه سمت چپ همه وابستگی‌ها، ابرکلید باشد، آن‌گاه آن جدول در BCNF قرار دارد.

۴۵- گزینه (۲) صحیح است.

این رابطه در 1NF هست چون کلیدهای آن معلوم‌اند (کد دانشجو)، صفت‌های آن به کلید اصلی (کد

دانشجو) وابسته‌اند، و صفت‌ها از دامنه‌های تو در تو نیستند.

این رابطه در 2NF هست چون هیچ صفتی به زیر مجموعه‌ای از کلید کاندید اصلی وابستگی ندارد.

این رابطه در 3NF هم هست چون وابستگی به صفت‌های غیرکلیدی نداریم (توجه کنید که از روی نام درس نمی‌توان نام استاد آن را به صورت یکتا تعیین کرد).

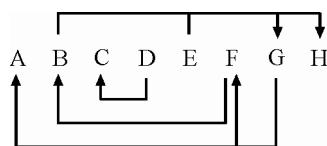
بنابراین وابستگی استاد → درس وجود ندارد. به دلیل مشابه، وابستگی درس → استاد هم وجود ندارد)

این جدول در BCNF هم هست چون ستون‌های آن فقط به کلیدهای کاندیدش وابسته‌اند.

اما این رابطه در 4NF نیست چون وابستگی‌های چند مقادیری غیربدیهی درس → → استاد و استاد → درس وجود دارند که در آن‌ها «درس» و «استاد» ابرکلید نیستند.

۴۶- گزینه (۳) صحیح است.

نمودار وابستگی تابعی رابطه R به صورت زیر است:



به سادگی دیده می‌شود که صفت A توسط برخی از صفات دیگر قابل دستیابی است (توسط G به طور مستقیم و توسط B و F به طور با واسطه). بنابراین هر کلید که شامل این صفات باشد نیازی نیست شامل صفت A باشد. پس گزینه سوم عبارت نادرستی است و پاسخ این سؤال است.

همچنین مشاهده می‌شود که صفت D توسط هیچ یک از صفات دیگر قابل دستیابی نیست. بنابراین هر کلیدی باید حتماً شامل D باشد. پس گزینه چهارم عبارت درستی است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسسطو

روشن اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیربدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روشن دوم:

$$\text{عضو کلید کاندید} = \bigcup_{i=1}^n [x_i(y_i) - (x_i)]$$

با توجه به وابستگی‌های مطرح شده برای رابطه R(A,B,C,D,E,F,G,H) داریم:

$$BE \rightarrow GH$$

$$G \rightarrow FA$$

$$D \rightarrow C$$

$$F \rightarrow B$$

$$\text{A B C D E F G H} - \text{A B C F G H} = \text{D E}$$

بنابر رابطه فوق صفت DE حتماً باید عضو کلید کاندید باشد. بنابراین گرینه چهارم درست است و گرینه سوم نادرست است.

$$\{DE\}^+ = \{D, E, C\}$$

بستار صفات DE به صورت رو به رو است:

قانون سوم ارسسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول، برخی از ستون‌ها را تولید کند، بدین معنی است که، جدول موردنظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد.

همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت.

براساس بستار فوق، صفات DE به عنوان عضو کلید کاندید همه ستون‌ها را تولید نمی‌کند، بنابراین مطابق قانون سوم ارسسطو باید صفاتی در کنار صفات DE قرار گیرد تا کلید کاندید تولید گردد. این صفات کناری از صفات باقی مانده به غیر از صفات CDE انتخاب می‌گردند، صفات باقی مانده عبارتنداز H و G و F و B و A البته از این مجموعه صفات A و H را هم کنار می‌گذاریم، زیرا کمکی در تولید صفات دیگر نمی‌کنند. چون در سمت چپ هیچ یک از وابستگی‌های تابعی نیامده‌اند. بنابراین با ترکیب صفات B یا F یا G یا با صفات DE کلیدهای کاندید تولید می‌گردد.

$$\{BDE\}^+ = \{B, D, E, G, H, C, F, A\}$$

$$\{FDE\}^+ = \{F, D, E, B, C, G, H, A\}$$

$$\{GDE\}^+ = \{G, D, E, C, F, A, B, H\}$$

پس ترکیبات صفات BDE و FDE و GDE کلیدهای کاندید جدول R هستند.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده در مجموع دارای سه کلید کاندید است. که مطابق قانون سوم ارسسطو، عضو کلید کاندید DE، در بین هر سه کلید کاندید به طور مشترک قرار دارد.

حال یک بار دیگر وابستگی‌های مطرح شده برای رابطه R(A, B, C, D, E, F, G, H) را در نظر بگیرید:

عضو کلید کاندید G → BE عضو کلید کاندید BE → عضو کلید کاندید وابستگی معکوس

غیر کلید H → BE عضو کلید BE → عضو کلید کاندید وابستگی بخشی

عضو کلید کاندید F → G عضو کلید کاندید G → عضو کلید کاندید وابستگی معکوس

غیر کلید A → G عضو کلید کاندید G → عضو کلید کاندید وابستگی بخشی

غیر کلید C → D عضو کلید کاندید D → عضو کلید کاندید وابستگی بخشی

عضو کلید کاندید B → F عضو کلید کاندید F → عضو کلید کاندید وابستگی معکوس

وابستگی معکوس: وابستگی یک عضو کلید کاندید به عضو یک کلید کاندید دیگر یا مؤلفه غیر کلیدی را، وابستگی معکوس می‌نامند.

عضو کلید کاندید \rightarrow عضو کلید کاندید

عضو کلید کاندید \rightarrow غیر کلید

در وابستگی های فوق، وابستگی بخشی (غیر کلید به عضو کلید کاندید) وجود دارد، بنابراین جدول در نرمال فرم دوم قرار ندارد، و در نتیجه در فرم نرمال سطح سوم و BCNF هم قرار ندارد. بنابراین گزینه اول و دوم درست هستند.

در یک نگاه دیگر برای BCNF می توان گفت، جدولی در BCNF قرار دارد که همه شروع های وابستگی ها، ابر کلید باشد. به بیان دیگر هرگاه سمت چپ همه وابستگی ها، ابر کلید باشد، آنگاه آن جدول در BCNF قرار دارد که در وابستگی های فوق این چنین نیست. پس BCNF هم نیست.

۴۷- گزینه (۱) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابر کلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می آید:

قانون اول ارسسطو

روش اول:

اجتماع تمام خصیصه های سمت راست وابستگی های غیر بدیهی - تمام خصیصه های جدول = عضو کلید کاندید

روش دوم:

$$\bigcup_{i=1}^n (x_i - y_i) \cap R(\text{راست}) = \text{عضو کلید کاندید}$$

توجه: وابستگی بدیهی: اگر در یک وابستگی، بخش سمت راست، زیرمجموعه بخش سمت چپ باشد، آن وابستگی را، وابستگی بدیهی می نامند.

مثال:

با توجه به وابستگی های مطرح شده برای رابطه (E و D و C و B و A) داریم:

$BC \rightarrow A$

$A \rightarrow D$

$D \rightarrow C$

$D \rightarrow E$

$$A \not\rightarrow B \not\rightarrow C \not\rightarrow D \not\rightarrow E = B$$

بنابر رابطه فوق صفت B حتماً باید عضو کلید کاندید باشد.

بستار صفت B به صورت زیر است:

$$\{B\}^+ = \{B\}$$

براساس بستار فوق، صفت B، فقط ستون B را تولید می کند، پس صفت B فقط عضو کلید کاندید می باشد و کلید کاندید نمی باشد.

قانون سوم ارسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، برخی از ستون‌ها را تولید کند، بدین معنی است که، جدول مورد نظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کنند تا کلید کاندید ایجاد گردد. همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت.

صفات همراه عضو کلید کاندید، از صفات باقی‌مانده به غیر از صفت B انتخاب می‌گردد، صفات باقی‌مانده عبارتند از E و D و C و A. البته از این مجموعه، صفت E را هم کنار می‌گذاریم، زیرا کمکی در تولید صفات دیگر نمی‌کند. چون در سمت چپ هیچ یک از وابستگی‌های تابعی نیامده است. بنابراین با ترکیب صفات A یا C یا D با صفت B کلیدهای کاندید تولید می‌گردد.

$$\{AB\}^+ = \{A, B, D, C, E\}$$

$$\{CB\}^+ = \{C, B, A, D, E\}$$

$$\{DB\}^+ = \{D, B, C, A, E\}$$

براساس بستارهای فوق، صفات AB، CB و DB، هر یک به تنها یی همه ستون‌ها را تولید می‌کنند، پس صفات AB، CB و DB، کلیدهای کاندید جدول R هستند.

حال یکبار دیگر وابستگی‌های مطرح شده برای رابطه R(A,B,C,D,E) R(A, B, C, D, E) را در نظر بگیرید:

عضو کلید کاندید → A وابستگی کامل

عضو کلید کاندید → D وابستگی معکوس

عضو کلید کاندید → C وابستگی معکوس

غیرکلید → E وابستگی بخشی

وابستگی معکوس: وابستگی یک عضو کلید کاندید به یک عضو کلید کاندید دیگر یا مؤلفه غیرکلیدی را، وابستگی معکوس می‌نامند.

عضو کلید کاندید → عضو کلید کاندید

عضو کلید کاندید → غیرکلید

در وابستگی‌های فوق، وابستگی بخشی (غیرکلید به عضو کلید کاندید) وجود دارد، بنابراین جدول در نرمال فرم دوم قرار ندارد، و در نتیجه در فرم نرمال سطح سوم و BCNF هم قرار ندارد.

در یک نگاه دیگر، برای BCNF می‌توان گفت، جدولی در BCNF قرار دارد که همه شروع‌های وابستگی‌ها، ابرکلید باشد. به بیان دیگر هرگاه سمت چپ همه وابستگی‌ها، ابرکلید باشد، آنگاه آن جدول در BCNF قرار دارد که در وابستگی‌های فوق این چنین نیست. پس BCNF هم نیست.

۴۸- گزینه (۳) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر بدست می‌آید:

قانون اول ارسسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیربدیهی – تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\bigcup_{i=1}^n [x_i \cdot (چپ) - y_i \cdot (راست)] = R \text{ عضو کلید کاندید}$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D, E, F, G)$ داریم:

$AB \rightarrow CDEFG$

$A \rightarrow G$

$B \rightarrow C$

$C \rightarrow D$

$AB \not\rightarrow DEF \not\rightarrow CDEFG = AB$

بنابر رابطه فوق صفات AB حتماً باید عضو کلید کاندید باشد. بستار صفات AB به صورت زیر است:

$$\{AB\}^+ = \{A, B, C, D, E, F, G\}$$

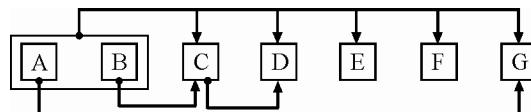
براساس بستار فوق، صفات AB ، همه ستون‌ها را تولید می‌کند، پس صفات AB کلید کاندید می‌باشد.

قانون دوم ارسسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روش اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

جدولی در سطح نرمال فرم اول قرار دارد که صفت چند مقداری و مرکب نداشته باشد. از آنجا که در جدول R ، صفت چند مقداری و مرکب وجود ندارد، بنابراین این جدول در نرمال فرم اول قرار دارد.

نمودار وابستگی‌های جدول R به صورت زیر است:



به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم اول باشد.
- جدول باید فاقد وابستگی بخشی باشد.

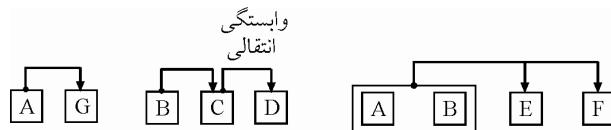
وابستگی بخشی: وابستگی یک مؤلفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می‌نامند. با توجه به این که ترکیب صفات AB کلید کاندید رابطه R می‌باشد، وابستگی‌های تابعی زیر:

$A \rightarrow G$

$B \rightarrow C$

به عنوان وابستگی بخشی محسوب می‌گردند. بنابراین جدول R در نرمال فرم دوم قرار ندارد.

بنابراین برای قرار دادن جدول R ، در نرمال فرم دوم، باید وابستگی‌های بخشی از جدول R خارج گردد.



بنابراین جدول R به سه جدول کوچکتر تجزیه می‌گردد:

$R_1(A, G)$

$R_2(B, C, D)$

$R_3(A, B, E, F)$

با توجه به کلید کاندید در دو جدول R_1 و R_3 می‌توان دریافت که این دو جدول فاقد وابستگی بخشی و انتقالی هستند، بنابراین در نرمال فرم سوم قرار دارند. همچنین از آن جا که سمت چپ تمام وابستگی‌های دو جدول R_1 و R_3 ، ابرکلید است، بنابراین دو جدول R_1 و R_3 در سطح BCNF نیز قرار دارند. بنابراین گزینه سوم درست است. اما راه حل را تا سطح BCNF برای جدول R_2 نیز ادامه می‌دهیم.

- جدول باید در نرمال فرم دوم باشد.
- جدول باید فاقد وابستگی انتقالی باشد.

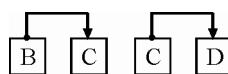
• وابستگی انتقالی: وابستگی یک مؤلفه غیرکلیدی به یک مؤلفه غیرکلیدی دیگر را وابستگی انتقالی می‌نامند.

در جدول R_2 وابستگی تابعی زیر:

$C \rightarrow D$

به عنوان وابستگی انتقالی محسوب می‌گردد. بنابراین جدول R_2 در نرمال فرم سوم قرار ندارد.

بنابراین برای قرار دادن جدول R_2 ، در نرمال فرم سوم، باید وابستگی‌های انتقالی از جدول R_2 خارج گردد.



بنابراین جدول R_2 به دو جدول کوچکتر تجزیه می‌گردد.

$R_{21}(B, C)$

$R_{22}(C, D)$

با توجه به کلیدهای کاندید در دو جدول R_{21} و R_{22} می‌توان دریافت که این دو جدول فاقد وابستگی بخشی و انتقالی هستند و بنابراین در نرمال فرم سوم قرار دارند. همچنین از آن جا که سمت چپ تمام وابستگی‌های سه جدول R_{21} و R_{22} ، ابرکلید است، بنابراین دو جدول R_{21} و R_{22} در سطح BCNF نیز قرار دارند. به این ترتیب با تجزیه جدول R به چهار جدول R_1 ، R_{21} ، R_{22} و R_3 می‌توان جدول R را در نرمال فرم سوم و سطح BCNF قرار داد.

۴۹- گزینه (۴) صحیح است.

گزینه اول نادرست است. عملیات نرمال‌سازی یک عملیات چند مرحله‌ای است که با طی آن‌ها اهداف نرمال‌سازی به تدریج حاصل می‌شود. نخستین مرحله نرمال‌سازی تبدیل جداول پایه به نرمال فرم اول است. برای قرار گرفتن یک جدول در نرمال فرم اول، شروط خاصی باید برقرار باشد. یک جدول (رابطه)

برای قرار گرفتن در نرمال فرم اول باید دارای شروط زیر باشد: (حفظ قوانین جامعیت درون رابطه‌ای)

۱- دارای حداقل یک کاندید باشد.

۲- جدول تو جدول یا جدول تو در تو نباشد (یعنی جدول فاقد صفت مرکب و چند مقداری باشد)

۲-۱- همه صفات آن غیرقابل تجزیه (Atomic) باشند. (جدول باید فاقد صفت مرکب باشد)

۲-۲- همه صفات جدول تک مقداری باشند. (جدول باید فاقد صفت چند مقداری باشد).

گزینه اول برای تحقق سطح اول نرمال‌سازی، یعنی نرمال فرم اول، فقط به یکی از سه شرط فوق اشاره کرده است.

گزینه دوم نادرست است. زیرا، ابرکلیدی که عضو زائد نداشته باشد، کلید کاندید است. به عبارت دیگر ابرکلیدی که کاهش ناپذیر باشد، کلید کاندید است.

ابرکلید (Super key)

هر ترکیبی از صفات (ستون ها) که خاصیت کلیدی داشته باشد، یک ابرکلید است. دقت کنید که خاصیت کلیدی بر اساس محیط خارج مشخص می‌گردد و نه محتوی جدول.

تعريف خاصیت کلیدی: اگر بر اساس آن مورد، جستجو انجام شود، فقط و فقط یک مورد را به صورت یکتا برگرداند. در واقع قدرت تفکیک داشته باشد.

مثال: شماره دانشجویی یا شماره ملی

توجه: هر ترکیبی از ابرکلید با سایر ستون های جدول مورد نظر نیز یک ابرکلید است. بنابراین ابرکلید می‌تواند عضو زائد داشته باشد و کمینه (minimal) نباشد، پس گزینه دوم پاسخ سؤال است.

مثال:

S#	Sname	city
S1	Sn1	C1
S2	Sn2	C2
S3	Sn3	C2
S4	Sn4	C2

ابرکلید است.

ابرکلید نیست.

(Sname,City) : ابرکلید نیست.

(S#,Sname) : ابرکلید است.

(S#,City) : ابرکلید است.

مثال: تعداد ابرکلیدهای مثال فوق چند عدد است؟

پاسخ: برابر 4 عدد می‌باشد.

S#	+	Sname	City	ابر کلید
خاصیت	0	0 →	(S#)	
کلیدی دارد	0	1 →	(S#,City)	
بنابراین	1	0 →	(S#,Sname)	
ابرکلید است.	1	1 →	(S#,Sname,City)	

کلید کاندید (Candidate key)

ابرکلیدی که عضو زائد نداشته باشد، کلید کاندید است.

مثال:

S#	Sname	City
S1	Sn1	C1
S2	Sn2	C2
S3	Sn3	C2

جدول *S*

S#	P#	QTY
S1	P1	10
S1	P2	20
S2	P1	30

جدول *SP*

P#	Pname	Color
P1	Pn1	Red
P2	Pn2	Blue
P3	Pn3	Blue

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 تهابی 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 تهابی خاصیت کلیدی دارد، بنابراین صفت نام خانوادگی، عضو زائد است.

توجه: یک جدول می‌تواند چندین کلید کاندید داشته باشد.

مثال:

شماره دانشجویی	شماره ملی	نام خانوادگی	نام
کلید کاندید	کلید کاندید		
(کلید فرعی)	(کلید فرعی)		

توجه: در مدل رابطه‌ای، هر رابطه حتماً حداقل یک کلید کاندید دارد، زیرا در بدترین شرایط، همه صفات با هم کلید کاندید می‌شوند، که به این رابطه تمام کلید (All key) گفته می‌شود.

توجه: یک رابطه، تحت هیچ شرایطی نمی‌تواند به دلیل استفاده از خاصیت مجموعه‌ای بودن، سطر تکراری داشته باشد. بنابراین یک رابطه، حداقل یک کلید کاندید دارد.

مثال: یک جدول تمام کلید.

a	b	c
1	2	3
1	6	3
1	2	7
8	2	3

ویژگی‌های رابطه در مدل رابطه‌ای به صورت زیر است:

۱- در رابطه، تاپل تکراری وجود ندارد. زیرا رابطه یک مجموعه است و مجموعه در ریاضیات طبق تعریف تاپل تکراری ندارد. بنابراین در جدول هم رکوردهای تکراری نداریم. بنابراین گزینه چهارم درست است.

۲- تاپل‌ها در رابطه نظم مکانی ندارند. زیرا مجموعه نظم ندارد. بنابراین ترتیب رکوردها در جدول اهمیت ندارد. بنابراین گزینه سوم نادرست است.

۳- ترتیب صفات خاصه در یک تاپل مهم نیست. البته به شرطی که ترتیب صفات خاصه در تمام تاپل‌های موجود در یک رابطه با یکدیگر سازگار باشند. بنابراین ترتیب فیلدها در یک جدول نیز مهم نیست، نظم ندارد و می‌توان آنرا جایه‌جا کرد. البته به شرطی که ترتیب فیلدها در تمام رکوردهای موجود در یک جدول با یکدیگر سازگار باشند.

۴- همهی مقادیر صفات خاصه در تاپل‌ها تجزیه‌ناپذیرند. به عبارت دیگر در مدل رابطه‌ای نمی‌توان فیلد مرکبی که از فیلد‌های ساده تشکیل شده‌است، تعریف نمود. یعنی فیلد‌های موجود در جداول مدل رابطه‌ای نباید مرکب مثل تاریخ تولد و آدرس باشد و نباید چند مقداری مثل مدرک تحصیلی باشد.

۵- گزینه (۱) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابر کلید 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 ارسسطو

روشن اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیربدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روشن دوم:

$$\bigcup_{i=1}^n [x_i - (y_i \text{ چپ})] = \text{عضو کلید کاندید}$$

با توجه به وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D)$ داریم:

$$AB \rightarrow C$$

$$AB \rightarrow D$$

$$BC \rightarrow A$$

$$BC \rightarrow D$$

$$A \rightarrow C$$

$$C \rightarrow A$$

$$ABC \not\rightarrow ACD = B$$

بنابر رابطه فوق صفت B حتماً باید عضو کلید کاندید باشد.

بستار صفات B به صورت رو به رو است:

قانون سوم ارسسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول، برخی از ستون‌ها را تولید کند، بدین معنی است که، جدول موردنظر، چندین کلید کاندید دارد، که این عضو کلید کاندید، در بین تمامی کلیدهای کاندید، به طور مشترک قرار دارد، بنابراین صفات دیگری نیز، باید عضو کلید کاندید را همراهی کند تا کلید کاندید ایجاد گردد. همچنین مطابق این قانون، صفاتی که توسط عضو کلید کاندید، قابل دسترسی هستند، در کلید کاندید جایگاهی نخواهند داشت.

براساس بستار فوق، صفت B به عنوان عضو کلید کاندید همه ستون‌ها را تولید نمی‌کند، بنابراین مطابق قانون سوم ارسسطو باید صفاتی در کنار صفت B قرار گیرد تا کلید کاندید تولید گردد. این صفات کناری از صفات باقی مانده انتخاب می‌گردند، صفات باقی مانده عبارتند از D و C و A البته از این مجموعه صفت D را هم کنار می‌گذاریم، زیرا کمکی در تولید صفات دیگر نمی‌کند. چون در سمت چپ هیچ یک از وابستگی‌های تابعی نیامده است. بنابراین با ترکیب صفات A یا C با صفت B کلیدهای کاندید تولید می‌گردد.

$$\{AB\}^+ = \{A, B, C, D\}$$

$$\{BC\}^+ = \{B, C, A, D\}$$

پس ترکیبات صفات AB و BC کلیدهای کاندید جدول R هستند.

توجه: کلید اصلی، یکی از کلیدهای کاندید است که بر اساس سلیقه طراح پایگاه داده، انتخاب می شود. مطابق قوانین مدل رابطه‌ای، هر جدول حتماً باید یک کلید اصلی داشته باشد. همچنین هر جدول فقط می‌تواند یک کلید اصلی داشته باشد. هر کلید کاندیدی که به عنوان کلید اصلی انتخاب نشده است، یک کلید فرعی است. مطابق فرض سوال AB کلید اصلی و کلید BC کلید کاندید(کلید فرعی) در نظر گرفته شده است.

توجه: دقت کنید که هیچ‌گاه، کلید کاندید نباید عضو زائد داشته باشد. بنابراین رابطه داده شده در مجموع دارای دو کلید کاندید است. که مطابق قانون سوم ارسطو، عضو کلید کاندید B، در بین هر دو کلید کاندید به طور مشترک قرار دارد.

حال یکبار دیگر وابستگی‌های مطرح شده برای رابطه $R(A, B, C, D)$ را در نظر بگیرید:

عضو کلید کاندید $AB \longrightarrow C$ کلید کاندید وابستگی کامل

غیرکلید $AB \longrightarrow D$ کلید کاندید وابستگی کامل

عضو کلید کاندید $BC \longrightarrow A$ کلید کاندید وابستگی کامل

غیرکلید $BC \longrightarrow D$ کلید کاندید وابستگی کامل

عضو کلید کاندید $C \longrightarrow A$ عضو کلید کاندید وابستگی معکوس

عضو کلید کاندید $A \longrightarrow C$ عضو کلید کاندید وابستگی معکوس

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم اول را به صورت زیر بیان کرد:

- دارای حداقل یک کلید کاندید باشد.

• همه خصیصه‌های آن غیرقابل تجزیه باشند (جدول باید فاقد خصیصه‌های مرکب باشد)

• همه خصیصه‌های آن تک مقداری باشند (جدول باید فاقد خصیصه‌های چند مقداری باشد)

واضح است که جدول مطرح شده در فرم اول نرمال قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم اول باشد.

- جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مؤلفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می‌نامند.

مؤلفه غیرکلید: هر صفتی که عضو هیچ کلید کاندیدی نباشد، به عنوان مؤلفه غیرکلیدی نامیده می‌شود.

مؤلفه جزء کلید کاندید: هر صفتی که عضو حداقل یک کلید کاندید باشد، به عنوان مؤلفه جزء کلید نامیده می‌شود.

در وابستگی‌های فوق، وابستگی بخشی وجود ندارد. بنابراین جدول مربوطه در نرمال فرم دوم هم قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم دوم باشد.

- جدول باید فاقد وابستگی انتقالی باشد.

وابستگی انتقالی: وابستگی یک مؤلفه غیرکلیدی به یک مؤلفه غیرکلیدی دیگر را وابستگی انتقالی می‌نامند.

در وابستگی‌های فوق، وابستگی انتقالی وجود ندارد. بنابراین جدول مربوطه در نرمال فرم سوم هم قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم BCNF را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم سوم باشد.
- جدول باید فاقد وابستگی معکوس باشد.

وابستگی معکوس: وابستگی یک عضو کلید کاندید به عضو یک کلید کاندید دیگر یا مؤلفه غیرکلیدی را، عضو کلید کاندید \rightarrow عضو کلید کاندید \rightarrow غیرکلید کاندید \rightarrow غیرکلید

در وابستگی‌های فوق، وابستگی معکوس وجود دارد. بنابراین جدول مربوطه در نرمال فرم BCNF قرار ندارد. در یک نگاه دیگر برای نرمال فرم BCNF می‌توان گفت، جدولی در نرمال فرم BCNF قرار دارد که همه شروع‌های وابستگی‌ها، ابرکلید باشد. به بیان دیگر هرگاه سمت چپ همه وابستگی‌ها، ابرکلید باشد، آن‌گاه آن جدول در BCNF قرار دارد که در وابستگی‌های فوق این چنین نیست. پس BCNF هم نیست. در وابستگی‌های فوق، سمت چپ وابستگی‌های اول، دوم، سوم و چهارم ابرکلید است. اما سمت چپ وابستگی‌های پنجم و ششم ابرکلید نیست.

بنابراین این جدول به دلیل نقض شرایط مربوطه، در نرمال فرم BCNF قرار ندارد. در نتیجه گزینه اول پاسخ سوال خواهد بود.

۵۱- گزینه (۴) صحیح است.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم BCNF را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم سوم باشد.
- جدول باید فاقد وابستگی معکوس باشد.

وابستگی معکوس: وابستگی یک عضو کلید کاندید به عضو یک کلید کاندید دیگر یا مؤلفه غیرکلیدی را، عضو کلید کاندید \rightarrow عضو کلید کاندید \rightarrow غیرکلید کاندید \rightarrow غیرکلید

در یک نگاه دیگر برای نرمال فرم BCNF می‌توان گفت، جدولی در نرمال فرم BCNF قرار دارد که همه شروع‌های وابستگی‌ها، ابرکلید باشد. به بیان دیگر هرگاه سمت چپ همه وابستگی‌ها، ابرکلید باشد، آن‌گاه آن جدول در BCNF قرار دارد.

طبق فرض سوال جدول $R(A,B,C,D,E)$ در نرمال فرم BCNF قرار دارد و داری تنها یک کلید کاندید به صورت ABC می‌باشد، بنابراین هر ترکیبی از کلید کاندید ABC با صفات باقی‌مانده ابرکلید می‌سازد. برای به دست آوردن ابرکلیدها، باید به کلید کاندید ABC، صفر یا بیشتر، از صفت‌های باقی‌مانده را اضافه کرد. بنابراین باید به کلید کاندید ABC، صفر، یک، یا دو مورد از صفت‌های دیگر (D, E) اضافه کرد. برای این کار 4 حالت، وجود دارد:

$$\binom{2}{0} + \binom{2}{1} + \binom{2}{2} = 2^2 = 4$$

ساخت ابرکلید: هیچ یا ترکیبی از صفات باقی‌مانده + کلید کاندید ABC = ابرکلید

A B C +	D E	ترکیبات صفات باقی‌مانده	ابرکلیدها
	0 0	تهی	A B C
	0 1	E	A B C E
	1 0	D	A B C D
	1 1	D E	A B C D E

توجه: واضح است که ۴ ابرکلید، ایجاد می‌گردد. (2^2)
حال وابستگی‌های مطرح شده در گزینه‌های صورت سوال را در نظر بگیرید:

$$BCE \rightarrow D$$

$$ACE \rightarrow D$$

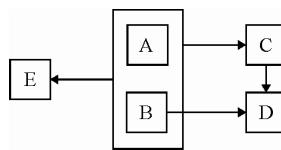
$$ABD \rightarrow E$$

$$ABCD \rightarrow E$$

براساس وابستگی‌های فوق، سمت چپ وابستگی‌های گزینه‌های اول، دوم و سوم ابرکلید نیست. اما سمت چپ وابستگی گزینه چهارم ابرکلید است، موجود در سطر سوم از جدول فوق. بنابراین از وابستگی‌های فوق، تنها سمت چپ وابستگی گزینه چهارم ابرکلید قرار دارد که پاسخ سوال نیز خواهد بود، به همین سادگی...

۵۲- گزینه (۱) صحیح است.

شکل زیر، نمودار وابستگی‌های تابعی صفات را نشان می‌دهد. پر واضح است که کلید این جدول صفات AB است. بنابراین بدیهی است که جدول 2NF نیست به دلیل وجود وابستگی‌های بخشی $B \rightarrow D$ (وابستگی غیرکلید به عضو کلید کاندید) پس جدول مذکور 3NF و BCNF نیز نیست. پس گزینه‌های دوم، سوم و چهارم پاسخ سوال نیستند. بنابراین گزینه اول پاسخ سوال است.



به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیربدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\text{با توجه به وابستگی‌های مطرح شده برای رابطه } R(A,B,C,D,E) \text{ داریم:}$$

$$\bigcup_{i=1}^n (x_i - R(y_i - (x_i - \text{عضو کلید کاندید})) = \text{عضو کلید کاندید}$$

$$C \rightarrow D$$

$$B \rightarrow D$$

$$AB \rightarrow CE$$

$$ABC\bar{D}\bar{E} - CDE = AB$$

بنابر رابطه فوق صفت AB حتماً باید عضو کلید کاندید باشد.

بستار صفت AB به صورت زیر است:

$$\{AB\}^+ = \{A, B, C, D, E\}$$

براساس بستار فوق، صفات AB، همه ستون‌ها را تولید می‌کند، پس صفات AB، کلید کاندید می‌باشد.

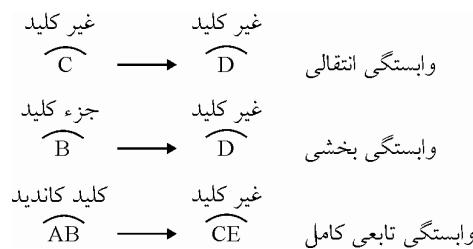
قانون دوم ارسسطو

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 وجود ندارد).
- جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مؤلفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می‌نامند.
وابستگی‌های مطرح شده را در نظر بگیرید:



در وابستگی‌های فوق، وابستگی بخشی $D \rightarrow B$ وجود دارد، بنابراین جدول مربوطه در نرمال فرم دوم قرار ندارد و به تبع در نرمال فرم سوم و BCNF هم قرار ندارد.

۵۳- گزینه (۱) صحیح است.

صورت سوال به این شکل است:

اگر در رابطه R، تعداد ابرکلیدها با تعداد کلیدهای کاندید برابر باشد، آنگاه کدام مورد در خصوص رابطه R نادرست است؟

(۱) تنها یک خصیصه دارد.

گزینه اول نادرست است، زیرا در جدول تمام کلید، یک جدول فقط و فقط یک ابرکلید دارد و فقط و فقط هم یک کلید کاندید دارد. رابطه تمام کلید مثلاً ممکن است سه ستون داشته باشد، در این حالت یک جدول فقط و فقط یک ابرکلید دارد و فقط و فقط هم یک کلید کاندید دارد. یعنی حداقل یک ابرکلید و یک کلید کاندید دارد و حداقل 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 حداقل 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 ۳NF 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 ۳NF 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

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.

- ۵۴- گزینه (۳) صحیح است.

به طور کلی کلید کاندید باید دو شرط زیر را داشته باشد:

۱- ابرکلید باشد (خاصیت کلیدی داشته باشد) یعنی همه خصیصه‌ها را تولید کند.

۲- عضو زائد نداشته باشد.

به طور کلی عضو کلید کاندید از روابط زیر به دست می‌آید:

قانون اول ارسسطو

روش اول:

اجتماع تمام خصیصه‌های سمت راست وابستگی‌های غیربدیهی - تمام خصیصه‌های جدول = عضو کلید کاندید

روش دوم:

$$\bigcup_{i=1}^n (x_i, y_i) = \text{عضو کلید کاندید}$$

با توجه به وابستگی‌های مطرح شده برای رابطه R(X, Y, Z) داریم:

$$\begin{aligned} Y &\rightarrow Z \\ XZ &\rightarrow Y \\ X &\rightarrow Z \\ XYZ - YZ &= X \end{aligned}$$

بنابر رابطه فوق صفت X حتماً باید عضو کلید کاندید باشد. بستار صفت X به صورت زیر است:

$$\{X\}^+ = \{X, Y, Z\}$$

براساس بستار فوق، صفت X ، همه ستون‌ها را تولید می‌کند، پس صفت X ، کلید کاندید می‌باشد.

قانون دوم ارسسطو

هرگاه عضو کلید کاندید، حاصل از تفاضل قانون اول (روشن اول یا دوم)، همه ستون‌ها را تولید کند، آن عضو کلید کاندید، تنها کلید کاندید جدول خواهد بود.

حال یکبار دیگر وابستگی‌های مطرح شده برای رابطه $R(X, Y, Z)$ را در نظر بگیرید:

وابستگی انتقالی $\text{غیرکلید } Z \longrightarrow Y \text{ غیرکلید}$

وابستگی تابعی $\text{غیرکلید } Y \longrightarrow XZ \text{ ابرکلید}$

وابستگی کامل $\text{غیرکلید } Z \longrightarrow X \text{ کلید کاندید}$

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم اول را به صورت زیر بیان کرد:

- دارای حداقل یک کلید کاندید باشد.

• همه خصیصه‌های آن غیرقابل تجزیه باشند (جدول باید فاقد خصیصه‌های مرکب باشد)

• همه خصیصه‌های آن تک مقداری باشند (جدول باید فاقد خصیصه‌های چند مقداری باشد)

واضح است که جدول مطرح شده در فرم اول نرمال قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم دوم را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم اول باشد.

• جدول باید فاقد وابستگی بخشی باشد.

وابستگی بخشی: وابستگی یک مؤلفه غیرکلیدی، به جزئی از کلید کاندید را وابستگی بخشی می‌نامند.

مؤلفه غیرکلید: هر صفتی که عضو هیچ کلید کاندیدی نباشد، به عنوان مؤلفه غیرکلیدی نامیده می‌شود.

مؤلفه جزء کلید کاندید: هر صفتی که عضو حداقل یک کلید کاندید باشد، به عنوان مؤلفه جزء کلید نامیده می‌شود.

در وابستگی‌های فوق، وابستگی بخشی وجود ندارد. بنابراین جدول مربوطه در نرمال فرم دوم هم قرار دارد.

به طور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم سوم را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم دوم باشد.

• جدول باید فاقد وابستگی انتقالی باشد.

وابستگی انتقالی: وابستگی یک مؤلفه غیرکلیدی به یک مؤلفه غیرکلیدی دیگر را وابستگی انتقالی می‌نامند.

در وابستگی‌های فوق، وابستگی انتقالی وجود دارد. بنابراین جدول مربوطه در نرمال فرم سوم قرار ندارد. و

به تبع در نرمال فرم BCNF هم قرار ندارد.

بطور کلی می‌توان شروط قرار داشتن یک جدول در نرمال فرم BCNF را به صورت زیر بیان کرد:

- جدول باید در نرمال فرم سوم باشد.

• جدول باید فاقد وابستگی معکوس باشد.

وابستگی معکوس: وابستگی یک عضو کلید کاندید به عضو یک کلید کاندید دیگر یا مؤلفه غیرکلیدی را، عضو کلید کاندید \rightarrow عضو کلید کاندید.

عضو کلید کاندید \rightarrow غیرکلید

در وابستگی های فوق، وابستگی معکوس وجود ندارد. اما جدول فوق در نرمال فرم سوم قرار ندارد، بنابراین جدول مربوطه در نرمال فرم BCNF هم قرار ندارد.

در یک نگاه دیگر برای نرمال فرم BCNF می توان گفت، جدولی در نرمال فرم BCNF قرار دارد که همه شروع های وابستگی ها، ابرکلید باشد. به بیان دیگر هرگاه سمت چپ همه وابستگی ها، ابرکلید باشد، آن گاه آن جدول در BCNF قرار دارد که در وابستگی های فوق این چنین نیست. پس BCNF هم نیست.

در وابستگی های فوق، سمت چپ وابستگی های دوم و سوم ابرکلید است. اما سمت چپ وابستگی اول ابرکلید نیست.

بنابراین این جدول به دلیل نقض شرایط مربوطه، در نرمال فرم BCNF قرار ندارد. در نتیجه گزینه دوم گزاره درستی است.

گزینه اول گزاره درستی است، زیرا صفت X در سمت راست هیچ یک از وابستگی های تابعی مطرح شده در صورت سوال نیست، پس از روی صفات دیگر قابل دستیابی نیست و بنابراین حتماً باید در کلید کاندید حضور داشته باشد. بعلاوه، با داشتن X می توان تمام صفت های دیگر را به دست آورد، پس {X} تنها کلید کمینه است و بنابراین ستون X تنها کلید کاندید رابطه R است.

گزینه دوم نیز گزاره درستی است، زیرا با توجه به وابستگی Z \rightarrow Y وابستگی انتقالی یعنی وابستگی غیرکلید به غیرکلید وجود دارد، بنابراین رابطه R در سطح نرمال فرم سوم قرار ندارد و به تبع همین موضوع در سطح نرمال فرم BCNF هم قرار ندارد.

گزینه سوم گزاره نادرستی است، زیرا وابستگی سوم یعنی وابستگی Z \rightarrow X اضافه و قابل حذف نیست، چون با داشتن X نمی توان از روی وابستگی های تابعی باقیمانده به Z رسید. در واقع وابستگی X \rightarrow Z یک وابستگی اصلی از مجموعه وابستگی های بهینه است و یک وابستگی فرعی نیست.

گزینه چهارم نیز گزاره نادرستی است، زیرا ستون X در وابستگی دوم، مشخصه اضافه و قابل حذف نیست. چون ستون X توسط ستون دیگری تولید نمی شود یعنی در سمت راست هیچکدام از وابستگی های مطرح شده در صورت سوال قرار ندارد که بتوانیم ستون X را حذف کنیم. به عبارت دیگر اگر ستون X توسط ستون Z تولید می شد، آنگاه ستون X در وابستگی دوم، مشخصه اضافه و قابل حذف می بود که اینطور نیست.

توجه: البته اگر طراح محترم در گزینه چهارم به جای ستون X از ستون Z استفاده می کرد که ماهم فکر می کنیم همین بوده است اما در مرحله حروفچینی سوال، ستون Z به صورت X حروفچینی شده است. آنگاه گزینه چهارم هم گزاره درستی می بود و همان گزینه سوم در بین چهار گزینه فقط گزاره ای نادرست می شد و پاسخ سوال هم همان گزینه سوم و مد نظر طراح محترم می شد.

توجه: دقت کنید که با توجه به وابستگی Z \rightarrow X با داشتن X می توان Z را به دست آورد. پس با حذف Z از وابستگی Y \rightarrow XZ اطلاعاتی از بین نمی رود و حالت بهینه و کمینه وابستگی دوم به فرم Y \rightarrow X می باشد. دقت کنید که عامل حذف ستون X در وابستگی دوم، وجود و حضور وابستگی اصلی سوم است.

وابستگی اصلی سوم تحت هیچ شرایطی قابل حذف نیست.

توجه: سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه سوم را به عنوان پاسخ اعلام کرده بود.

تست‌های فصل دهم: شاخص‌گذاری اطلاعات

۱- جدول T را در نظر بگیرید که روی ستون a **Clustered Index** شده است. کدام مورد درست است؟

(مهندسی کامپیوتر- دولتی ۹۷)

- ۱) اعمال سیاست شاخص‌گذاری، تاثیری بر حجم اطلاعات ذخیره شده بر روی دیسک ندارد.
- ۲) با اعمال سیاست شاخص‌گذاری، پاسخ به Range Query های مرتبط به a با سرعت بیشتری انجام می‌شود.
- ۳) با اعمال سیاست شاخص‌گذاری، پاسخ به Equality Query های مرتبط به a با سرعت کمتری انجام می‌شود.
- ۴) همه موارد درست هستند.

۲- در مورد شاخص‌گذاری (**Indexing**)، کدام مورد نادرست است؟

(مهندسی IT- دولتی ۹۷)

- ۱) B^+ Tree index برای پاسخ به Range Query مفید است.
- ۲) هر جدول می‌تواند حداقل یک شاخص کلاستر شده داشته باشد.
- ۳) شاخص‌های از نوع Hash برای پاسخ به Range Query مفید هستند.
- ۴) هرچه تعداد شاخص‌های یک جدول بیشتر باشد، سرعت Insert در آن جدول کمتر می‌شود.

پاسخ تست‌های فصل دهم: شاخص‌گذاری اطلاعات

۱- گزینه (۲) صحیح است.

دو هدف اصلی سیستم ذخیره و بازیابی اطلاعات در پایگاه داده‌ها، اول سرعت عملیات در ذخیره و بازیابی اطلاعات و دوم صرفه‌جویی در مصرف حافظه است. برای مثال کاهش افزونگی محتوایی (طبیعی) توسط نرمال‌سازی جداول منجر به کاهش میزان حافظه مصرفی می‌شود. عمل واکنشی تک تک رکوردها وقت‌گیر است، برای رفع این عیب شاخص‌یا Index ابداع شد. برای اینکه بازیابی داده‌ها با سرعت و کارایی بیشتر صورت گیرد، از شاخص استفاده می‌شود. شاخص ساختمان داده‌ای است که سیستم مدیریت پایگاه داده‌ها به کمک آن رکوردهای خاص را در یک فایل با سرعت زیاد پیدا می‌کند و به این ترتیب سرعت پاسخ به پرس و جوها افزایش می‌یابد. هر ساختار شاخص، حاوی رکوردهایی است که در هر رکورد یک مقدار کلیدی (کلید جستجو) و آدرس منطقی رکوردهای فایل داده‌ای نگهداری می‌شود.

اغلب سیستم‌های مدیریت پایگاه داده‌ها، از ساختار درخت برای ایجاد شاخص‌ها استفاده می‌کنند. عمق درخت، بیشترین تعداد سطوح از ریشه به برگ است. عمق ممکن است در مسیرهای مختلف از ریشه تا برگ متفاوت باشد. و همچنین عمق ممکن است در مسیرهای مختلف از ریشه تا برگ یکسان باشد، که در این شرایط با درخت متوازن و B^- -Tree و B^+ -Tree مواجه هستیم. هرچه درجه‌ی گره‌های درخت بیشتر شود، درخت پهن‌تر و کم‌عمق‌تری ایجاد می‌شود. از آنجاییکه زمان دسترسی در یک درخت، بیشتر وابسته به عمق درخت است تا پهنای آن، پس ساخت درخت پهن و کم‌عمق در ایجاد شاخص باعث افزایش سرعت جستجو می‌شود، ساختارهای B^- -Tree و B^+ -Tree درخت‌هایی با عمق کم و پهنای زیاد هستند. ساختار B^+ -Tree index برای پاسخ به Range Query و Equality Query مفید است.

صورت سوال به این شکل است:

جدول T را در نظر بگیرید که روی ستون a Clustering Index شده است. کدام مورد درست است؟

۱) اعمال سیاست شاخص‌گذاری، تاثیری بر حجم اطلاعات ذخیره شده بر روی دیسک ندارد.

گزینه اول نادرست است، زیرا تعریف و نگهداری شاخص موجب تحمیل سربار حافظه‌ای به سیستم می‌شود. شاخص بر روی دیسک نگهداری می‌شود و هنگام استفاده به حافظه اصلی آورده می‌شود، بنابراین اعمال سیاست شاخص‌گذاری، بر افزایش حجم اطلاعات ذخیره شده بر روی حافظه اصلی و هارد دیسک تاثیر دارد.

۲) با اعمال سیاست شاخص‌گذاری، پاسخ به Range Query‌های مرتبط به a، با سرعت بیشتری انجام می‌شود.

گزینه دوم درست است، زیرا Clustering Index توسط ساختار Tree⁺ پیاده‌سازی می‌شود. در

ساختمار B^+ Tree جستجو و بازیابی نقطه‌ای (Equality Query) از مرتبه $O(\log_t n)$ و جستجو و بازیابی بازه‌ای (Range Query) از مرتبه $O(\log_t n + k)$ است. بنابراین اعمال سیاست شاخص‌گذاری توسط ساختار B^+ Tree باعث می‌شود Range Query های مرتب، با سرعت بیشتری انجام شود.

(۳) با اعمال سیاست شاخص‌گذاری، پاسخ به a ، با سرعت کمتری انجام می‌شود.

گزینه سوم نادرست است، زیرا Clustered Index توسط ساختار B^+ Tree پیاده‌سازی می‌شود. در ساختار B^+ Tree جستجو و بازیابی نقطه‌ای (Equality Query) از مرتبه $O(\log_t n)$ و جستجو و بازیابی بازه‌ای (Range Query) از مرتبه $O(\log_t n + k)$ است. بنابراین اعمال سیاست شاخص‌گذاری توسط ساختار B^+ Tree باعث می‌شود Range Query های مرتب، با سرعت بیشتری انجام شود.

(۴) همه موارد درست هستند.

گزینه چهارم نادرست است، زیرا فقط گزینه دوم درست است.

- ۲- گزینه (۳) صحیح است.

دو هدف اصلی سیستم ذخیره و بازیابی اطلاعات در پایگاه داده‌ها، اول سرعت عملیات در ذخیره و بازیابی اطلاعات و دوم صرفه‌جویی در مصرف حافظه است. برای مثال کاهش افزونگی محتوایی (طبیعی) توسط نرم‌افزار منجر به کاهش میزان حافظه مصرفی می‌شود. عمل واکشی 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 کم عمق‌تری ایجاد می‌شود. از آنجاییکه زمان دسترسی در یک درخت، بیشتر وابسته به عمق درخت است تا پهنه‌ای آن، پس ساخت درخت پهن و کم عمق در ایجاد شاخص

باعث افزایش سرعت جستجو می‌شود، ساختارهای B^- -Tree و B^+ -Tree درخت‌های با عمق کم و پهنای زیاد هستند. ساختار B^+ -Tree index برای پاسخ به Range Query و Equality Query مفید است.

صورت سوال به این شکل است:

در مورد شاخص‌گذاری (Indexing)، کدام مورد نادرست است؟

(۱) B^+ -Tree index برای پاسخ به Range Query مفید است.

گزینه اول درست است، زیرا Clustered Index توسط ساختار B^+ -Tree پیاده‌سازی می‌شود. در ساختار B^+ -Tree جستجو و بازیابی نقطه‌ای Equality Query از مرتبه $O(\log t n)$ بازجو و بازیابی نقطه‌ای Range Query از مرتبه $O(\log t n + k)$ است. بنابراین اعمال سیاست شاخص‌گذاری توسط ساختار B^+ -Tree باعث می‌شود Range Query‌ها و Equality Query‌ها مرتبط با سرعت بیشتری انجام شود.

(۲) هر جدول می‌تواند حداکثر یک شاخص کلاستر شده داشته باشد.

گزینه دوم درست است، زیرا هر جدول می‌تواند حداکثر یک clustered Index داشته باشد.

(۳) شاخص‌های از نوع Hash برای پاسخ به Range Query مفید هستند.

گزینه سوم نادرست است، زیرا در ساختار فایل مستقیم (تکنیک درهم‌سازی) جستجو و بازیابی نقطه‌ای Equality Query از مرتبه $O(1)$ است. دستیابی از مرتبه $O(1)$ به جدول پایه، به این معناست که بدون توجه به اندازه جدول پایه، جهت دسترسی به یک رکورد دلخواه، همواره تعداد اندکی پیگرد نیاز است. در حالت ایده‌آل با یک حرکت می‌توان به رکورد مورد نظر دسترسی پیدا کرد. همچنین در ساختار فایل مستقیم (تکنیک درهم‌سازی) جستجو و بازیابی بازه‌ای Range Query می‌باشد برای تک اعضای بازه به طور مستقل تابع hash اجرا شود که این امر مستلزم صرف وقت و هزینه زیادی است. فایل مستقیم بی‌نظم است و امکان پردازش سریالی و ترتیبی را ندارد، زیرا فایل مستقیم بر حسب کلید مرتب نشده است. به عبارت دیگر به علت درهم بودن فایل، واکنشی رکورد بعدی مشابه واکنشی یک رکورد جدید است، پس امکان پردازش سریالی و ترتیبی در آن وجود ندارد، ساختار فایل مستقیم مناسب محیط‌هایی است که دستیابی سریع به رکوردها مورد نیاز است و پردازش‌ها ترتیبی و سریالی نباشد. بنابراین برای جستجو و بازیابی بازه‌ای Range Query، شاخص‌گذاری مرتب شده با ساختار B^+ -Tree با مرتبه $O(\log t n + k)$ مناسب‌تر است. و شاخص‌های از نوع Hash برای پاسخ به Range Query مفید نیست.

(۴) هرچه تعداد شاخص‌های یک جدول بیشتر باشد، سرعت Insert در آن جدول کمتر می‌شود.

گزینه چهارم درست است، زیرا عملیات درج، حذف و بروزرسانی در جدولی که شاخص دارد یعنی خود

جدول پایه، نسبت به جدولی که شاخص ندارد زمان بیشتری مصرف می‌کند و به تبع این عملیات کندتر خواهد بود. زیرا شاخص‌ها نیز همگام با جداول پایه خود نیاز به بروزرسانی دارند. بنابراین تنها روی ستون‌هایی شاخص ایجاد می‌گردد، که به تناوب روی آنها جستجو انجام می‌شود. بنابراین هرچه تعداد شاخص‌های یک جدول بیشتر باشد، سرعت Insert، Update و Delete در آن جدول کمتر می‌شود. اما خود شاخص عامل جستجو و بازیابی سریع اطلاعات از یک جدول پایه است.

مشاوره تخصصی رشته کامپیوتر و IT

در راستای رسالت مؤسسه فرهنگی و انتشاراتی بابان مبنی بر ارتقای سطح علم و دانش کشور و کمک همه جانبه به دانشجویان و داوطلبان گرامی، در جهت قبولی در کنکور کارشناسی ارشد و دکتری مهندسی کامپیوتر و IT دو طرح زیر را پایه‌ریزی کرده‌ایم:

(۱) ارائه مشاوره تخصصی حضوری و غیرحضوری (تلفنی و آنلاین)

(۲) برگزاری کلاس‌های حضوری و غیرحضوری (فیلم آموزشی و کلاس آنلاین)
بدین منظور برای تماس با شما و تعیین وقت قبلی عدد ۲ را به شماره ۰۰۰۰۲۰۶۰۴۵ پیامک نمائید.

برای آشنایی بیشتر با خدمات ارائه شده توسط موسسه بابان به وب سایت [khalilifar.ir](#) یا کanal تلگرام [@arastookhalilifar](#) مراجعه فرمایید.

تلفن دفتر مرکزی موسسه بابان: ۰۲۱-۷۷۹۷۷۸۶۸

پایگاه اطلاع رسانی موسسه بابان: [www.baban.ir](#)

تلفن دفتر فروشگاه انتشارات بابان: ۰۲۱-۷۷۹۷۳۳۸۶

فروشگاه اینترنتی انتشارات بابان: [shop.baban.ir](#)