# پاسخنامه

## تمرینات و کنجکاوی‌های کتاب

# مفاهیم بنیادی پایگاه داده‌ها

**(جلد ۱)**

**سیّد محمّد تقی روحانی رانکوهی**

# تاریخچه تغییرات

| شماره نسخه | تاریخ | توضیحات |
|---|---|---|
| ۱,۰,۰ | ۱۳۹۸/۰۲/۰۱ | انتشار نسخه اولیه |

منبع طرح: اینترنت

# پیشگفتار

اینک ویراست نخست **پاسخنامه** تمرینات و کنجکاوی‌های کتاب " مفاهیم بنیادی پایگاه داده‌ها " . این کتاب حاوی پاسخ بیشتر تمرینات و کنجکاوی‌های کتاب مزبور است .

توجّه استفاده کنندگان محترم را به نکات زیر جلب می‌کنم :

۱. منابع مهمتر این پاسخنامه عبارتند از :

I. مفاهیم بنیادی پایگاه داده‌ها، سیّد محمّد تقی روحانی رانکوهی ، ویراست چهارم

II. An Introduction to Database Systems, C. J. Data, 8th edition (2003)

III. Database System Concepts, A. Silberschatz, H. F. Korth, S. Sudarsham, 6th Edition (2010)

IV. Fundamentals of Database Systems, R. Elmasri, S. B. Navathe, 6th Edition (2010)

V. Database Systems. A Practical Approach to Design, Implementation and Management, T. M. Connolly, C. E. Begg, 5th Edition (2009)

VI. Database Management Systems, R. Ramakrishnan, J. Gehrke, 3rd Edition (2003)

۲. استفاده از این پاسخنامه به منظور یادگیری ، برای خواهندگان آزاد است . اما اقتباس و برداشت از آن به هر میزان با هدف آموزش ، به شرطی مجاز است که مشخصات کامل منبع قید شود .

۳. تولید هر گونه نسخه دیگر از این اثر ، بطور جزیی یا کلّی و به هر شکل و صورت به قصد عرضه برای فروش و سودجویی ، کاری است کاملاً خلاف اخلاق تعلیم و تعلّم و فعلی است ضدّ وجدان و قانون ؛ همچون کارهایی

که چندی است متأسفانه در مورد سایر آثار دانشگاهی منتشرشده این مؤلف انجام شده است ، و به نظر می‌رسد بیشتر آنها مصداق همین امر خلاف باشند .

۴. نسخه دست‌نوشته این پاسخنامه را ابتدا در اختیار دستیارم در چند ترم تدریس در دانشکده مهندسی کامپیوتر دانشگاه صنعتی شریف ، آقای مجتبی ورمزیار ، دانش‌آموخته بسیار صالح و صادق ، اینک خود دانشور و فنّوَر در زمینه علم و فن رایانگری (انفورماتیک : دانش و فن کامپیوتر) قرار دادم . ایشان با همکاری چند دانش‌دوست دیگر از همان دانشکده ، ( به ترتیب حروف ) خانم‌ها شقایق اسماعیلی و پردیس پاشاخانلو ، و آقایان محمد امین صبّاغیان ، ایمان جامی مقدم ، امیرعلی معین‌فر ، و سیاوش نظری ، نسخه کنونی را آماده کرده‌اند . اگر نبود همّت این دانش آموختگان صمیمی ، صورت کنونی این اثر پدید نمی‌آمد . از همه این یاری‌دهندگان با خلوص و بویژه از آقای ورمزیار صمیمانه تشکر می‌کنم و آرزومندم در کسب و نشر دانش و فن بیش از پیش موفق و مؤید باشند .

از استفاده‌کنندگان محترم خواهشمندم نظرات بهتر ساز خود را به آقای ورمزیار اعلان کنند ، از پیش سپاسگزارم .

امید دارم که جوانان دانش‌دوست در **امر خطیر دانش جویی و دانش پژوهی** ، آن سان که در شأن تاریخی و کنونی این حیطه است ، بیش از پیش صادقانه کوشا و پیوسته قرین موفقیّت باشند .

سیّد محمّد تقی روحانی رانکوهی

دانشیار دانشگاه شهید بهشتی

**تهران، بهمن ماه یکهزار و سیصد و نود و هفت**

# یادداشت

این کتاب حاوی پاسخ بیشتر کنجکاوی‌ها و تمرینات کتاب " مفاهیم بنیادی پایگاه داده‌ها " ( ویراست چهارم کتاب ) است ، به جز تمرینات درون‌گفتار گفتار ۱۲ ( که در دست آماده‌سازی است ) و نیز بعض تمرینات و کنجکاوی‌هایی که مشابه دارند یا یافتن پاسخ آنها آسان است .

در برخی موارد که پاسخ یک تمرین یا کنجکاوی در متن کتاب " مفاهیم بنیادی پایگاه داده‌ها " یافت می‌شود ، با ذکر نشانی به کتاب مزبور ارجاع داده شده است . برای درک بهتر پاسخ‌ها ، اکیداً توصیه می‌شود کتاب مزبور با دقت مطالعه شود .

این پاسخنامه ، بنا به خواست جناب استاد روحانی رانکوهی (مؤلف) به صورت رایگان و آزاد در فضای مجازی عرضه می شود . با این همه ، اگر استفاده‌کننده‌ای تمایل داشت ، می‌تواند مبلغی را به دلخواه به مؤسسه خیریه زیر یا هر مورد خیر دیگر به تشخیص خویش ، کمک کند :

- موسسه خیریه محک به نشانی https://mahak-charity.org/main/index.php/fa

پاسخنامه حاضر بارها بازنگری شده است تا حتی‌الامکان عاری از هر گونه غلط باشد . با این همه ادعایی بر بی‌عیب بودن آن نیست . از این رو ، با احترام و سپاس از استفاده‌کنندگان محترم خواهشمندم هر گونه نقد ، نظر و پیشنهاد بهترساز را به یکی از دو نشانی زیر ارسال کنند :

db1.answersbook@gmail.com

mahmoud1468@gmail.com

ضمناً به دلیل امکان تکمیل و یا اصلاح برخی موارد ، این پاسخنامه شماره نسخه و تاریخچه تغییرات دارد که در ابتدای آن آمده است . همواره آخرین نسخه در هر یک از دو نشانی زیر در دسترس خواهد بود :

https://db1answersbook.github.io

http://ce.sharif.edu/~varmazyar/db/index.html

در پایان ، امیدوارم این پاسخنامه کمکی باشد به یادگیری بهتر و بیشتر درس پایگاه داده‌ها در دوره کارشناسی .

از جناب استاد روحانی رانکوهی نیز بسیار سپاسگزارم ، به خاطر آنچه به من آموختند ، از علم و فن و نیز فراتر از این حیطه‌ها ، در دو دوره تحصیلی ، در چند ترمی که دستیار ایشان بودم ، و طی آماده‌سازی این نسخه .

مجتبی ورمزیار

دانش‌آموخته ارشد مهندسی کامپیوتر (نرم‌افزار)

دانشگاه صنعتی شریف

**تهران، بهمن ماه یکهزار و سیصد و نود و هفت**

# فهرست مطالب


تاریخچه تغییرات ..... ۲
پیشگفتار ..... ۴
یادداشت ..... ۶
گفتار ۱ ..... ۱۱
کنجکاوی‌ها ..... ۱۱
تمرینات پایان گفتار ..... ۱۱
گفتار ۲ ..... ۱۲
کنجکاوی‌ها ..... ۱۲
تمرینات درون گفتار ..... ۱۳
تمرینات پایان گفتار ..... ۱۴
گفتار ۳ ..... ۱۷
کنجکاوی‌ها ..... ۱۷
تمرینات درون گفتار ..... ۲۱
تمرینات پایان گفتار ..... ۲۷
گفتار ۴ ..... ۴۶
کنجکاوی‌ها ..... ۴۶
تمرینات درون گفتار ..... ۴۸
تمرینات پایان گفتار ..... ۴۹
گفتار ۵ ..... ۵۱
کنجکاوی‌ها ..... ۵۱
تمرینات درون گفتار ..... ۵۲
تمرینات پایان گفتار ..... ۵۲

گفتار ۶ ..... ۵۴
کنجکاوی‌ها ..... ۵۴
تمرینات درون گفتار ..... ۵۴
تمرینات پایان گفتار ..... ۵۴
گفتار ۷ ..... ۵۵
کنجکاوی‌ها ..... ۵۵
تمرینات درون گفتار ..... ۵۵
تمرینات پایان گفتار ..... ۵۵
گفتار ۸ ..... ۵۶
کنجکاوی‌ها ..... ۵۶
تمرینات درون گفتار ..... ۵۶
تمرینات پایان گفتار ..... ۵۶
گفتار ۹ ..... ۵۸
کنجکاوی‌ها ..... ۵۸
تمرینات درون گفتار ..... ۵۸
تمرینات پایان گفتار ..... ۵۹
گفتار ۱۰ ..... ۶۰
کنجکاوی‌ها ..... ۶۰
تمرینات درون گفتار ..... ۶۱
تمرینات پایان گفتار ..... ۶۴
گفتار ۱۱ ..... ۷۳
کنجکاوی‌ها ..... ۷۳
تمرینات درون گفتار ..... ۷۳
تمرینات پایان گفتار ..... ۷۷
گفتار ۱۲ ..... ۷۹

کنجکاوی‌ها ۷۹
تمرینات پایان گفتار ۸۰
گفتار ۱۳ ۱۰۹
کنجکاوی‌ها ۱۰۹
تمرینات درون گفتار ۱۱۰
تمرینات پایان گفتار ۱۱۵
گفتار ۱۴ ۱۲۵
کنجکاوی‌ها ۱۲۵
تمرینات درون گفتار ۱۲۸
تمرینات پایان گفتار ۱۳۳
گفتار ۱۵ ۱۴۶
کنجکاوی‌ها ۱۴۶
تمرینات درون گفتار ۱۴۸
تمرینات پایان گفتار ۱۵۶
گفتار ۱۶ ۲۰۵
کنجکاوی‌ها ۲۰۵
تمرینات درون گفتار ۲۰۷
تمرینات پایان گفتار ۲۱۰

# گفتار ۱

## کنجکاوی‌ها

۱-۱* : به کوتاهی ، می‌توان گفت که داده مبنای تولید اطلاع و اطلاع مبنای تولید دانش (شناخت) است . دانش پایه‌ی اصلی و اساسی تصمیم‌سازی در تمام فعالیت‌ها در جوامع بشری بطور عام و در سازمان‌ها بطور خاص است .

۱- ۲ : سیستمی است که برپایه‌ی مجموعه‌هایی از داده‌های ذخیره‌شده و نرم‌افزارهای لازم ، نیازهای اطلاعاتی کاربران را برآورده می‌سازد .

۱- ۳ : هر سیستمی که عملکردش برپایه‌ی مجموعه‌ای از دانش باشد . دانش لازم در چنین سیستمی ممکن است توسط عامل انسانی تأمین شود یا عامل ماشینی . در حالت دوم وجود یک پایگاه دانش الزامی است . توجه داشته باشیم که سیستم پایگاه دانش با سیستم دانش-پایه تفاوت دارد .

## تمرینات پایان گفتار

۱-۴ :

- داده نیم‌ساختمند : متن HTML ، XML و . . .
- داده ناساختمند : تصویر یک صفحه‌ی روزنامه که در آن متن ، عکس و . . . وجود دارد ، نقشه ، صدا و . . .

۱- ۹ : در این باره می‌توان از جمله [روحا ۷۸- الف] معرفی شده در فهرست منابع کتاب مراجعه کرد .

* توصیه می‌شود در موضوع این کنجکاوی به کتاب‌های ذیربط مراجعه شود .

# گفتار ۲

## کنجکاوی‌ها

۲-۱ : یعنی در برنامه‌ی کاربردی جنبه‌هایی از فایلینگ باید بطور مستقیم یا غیرمستقیم استفاده شوند ، مثلاً برنامه‌ی فایل‌پرداز به فرمت رکورد فایل وابسته است . این وابستگی به این دلیل وجود دارد که اساساً در مشی فایلینگ ، مفهوم اصلی همان فایل است و همه‌ی آنچه در فایلینگ منطقی و / یا فیزیکی مطرح‌اند ، به نحوی در برنامه‌ی فایل‌پرداز مورد ارجاع و استفاده قرار می‌گیرند .

۲-۲ :

- برخی از مزایای مشی فایلینگ :

  - پیچیدگی نرم‌افزار واسط کمتر است .
  - دستیابی به فایل‌ها ، نسبت به وقتی که DBMS در میان باشد ، ساده‌تر و سریعتر است .
  - برنامه‌ساز به مجموعه‌ی فایل‌های موردنیازش دستیابی دارد و می‌تواند کنترل‌های لازم را خود اِعمال کند .
  - سربار ( فزونکاری ) موجود در سیستم پایگاهی برای اجرای درخواست‌ها ، در مشی فایلینگ کمتر است .
  - نیازی به زبان برنامه‌سازی خاصی نیست ، بلکه از یک ( بیش از یک ) زبان برنامه‌سازی متعارف و رایج استفاده می‌شود .

- در شرایطی می‌توان از مشی فایلینگ استفاده کرد ، از جمله :

  - سیستم ترجیحاً تک کاربری باشد .
  - اشتراک داده‌ها مطرح نباشد .
  - امنیت بالا برای داده‌ها مطرح نباشد و کنترل دقیق لازم نباشد .

- حجم داده‌ها پائین باشد و مرتباً رشد نکند .
- کاربرد ساده باشد .
- ارتباطات بین داده‌ها پیچیده نباشد .
- هزینه‌ی کافی برای استفاده از مشی پایگاهی موجود نباشد .
- تخصص در دانش و فن پایگاهی وجود نداشته باشد .
- ایجاد سیستم یکپارچه مورد نظر نباشد .
- تغییرات در داده‌ها کم باشد .
- بسامد درخواست‌های موردی خیلی پائین باشد .

۲-۳ : اصطلاح بانک داده‌ها که در متون دهه‌های ۱۹۷۰ و ۱۹۸۰ مطرح بوده و هنوز هم جای جای در متون دیده می‌شود ، را می‌توان کم و بیش معادل اصطلاح پایگاه داده‌ها دانست . اما برخی بر این نظرند که بانک داده‌ها مجموعه‌ای است از چند پایگاه داده‌ها در یک سازمان که می‌توانند با یکدیگر ، در محدوده‌ی فعالیت‌های سازمان ، ارتباط معنایی داشته باشند . از این دیدگاه ، سیستم بانک داده‌ها مجموعه‌ای است از سیستم‌های پایگاهی درون یک سازمان ، هر چند ممکن است بطور فیزیکی پراکنده باشند . توجه داشته باشیم که این برداشت از مفهوم بانک داده‌ها ربطی به سیستم با معماری چندپایگاهی ندارد .

۲-۴ : به گفتار چهارم مراجعه شود .

۲-۵ : به پاسخ تمرین ۲ و همچنین به گفتار چهارم مراجعه شود .

۲-۶ : یعنی تعریف داده‌ها به گونه‌ای که دربرگیرنده‌ی همه‌ی نیازهای داده‌ای کاربران باشد . به بیان دیگر تعریف داده‌هایی که طراح-پیاده‌ساز می‌بیند و با کمی تسامح می‌توان گفت اجتماع داده‌هایی است که همه‌ی کاربران نیاز دارند .

# تمرینات درون گفتار

۲-۲ : بیشتر جنبه‌های فایلینگ منطقی و فایلینگ فیزیکی می‌توانند تغییر کنند ، از جمله :

- قالب رکورد
- نوع فیلدها
- طول رکورد
- ساختار فایل
- استراتژی دستیابی
- لوکالیتی رکوردها

- رسانه‌ی ذخیره‌سازی و ...

۲-۳ : آنچه به عنوان معایب مشی ناپایگاهی برشمردیم ، عکس آن‌ها جزء مزایای مشی پایگاهی است (برای فهرست مزایا رجوع شود به گفتار هشتم) .

۲-۴ : به منابع مهندسی فایل ها مراجعه شود .

# تمرینات پایان گفتار

۲-۲ : در این شکل نکات مهمتر عبارت‌اند از :

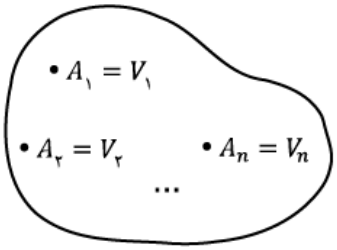
- شکل نمایش کلی تعریف پایگاه داده‌ها است .
- تعدد و تنوع کاربران
- تعدد و تنوع دیدهای کاربران نسبت به داده‌های دیده شده
- تفاوت در UFIها
- لزوم وجود یک تعریف جامع از داده‌ها
- لزوم وجود حداقل یک DBMS
- کاربران در محیط فرا فایلی کار می‌کنند .
- وحدت ذخیره سازی و کاهش حتی الامکان افزونگی
- مجتمع بودن داده‌ها
- DBMS در خود یا با خود یک FS دارد .
- سه لایه تعریف و کنترل داده‌ها وجود دارد : دو لایه در محیط فرافایلی و یک لایه در سطح فایلینگ منطقی .
- سه لایه عملیات در داده‌ها وجود دارد .

۲-۱۴ : مفهوم مطرح در سطح انتزاعی یعنی مطرح در سطحی فراتر از سطح نمایش منطقی و سطح پیاده سازی

مثال : مفهوم نمونه رکورد در سه سطح :

در سطح انتزاعی :

مجموعه‌ای است از جفت‌های : نام صفت و مقدار صفت .

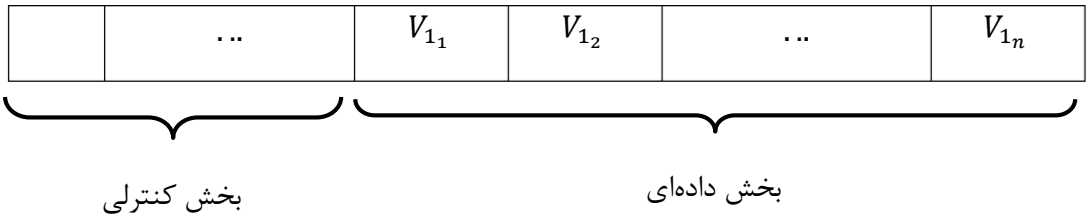



$A_i$ : نام صفت $V_i$ : مقدار صفت

در سطح نمایش منطقی ، با فرض قالب (فرمت) ثابت-مکان :

| $V_{1_1}$ | $V_{1_2}$ | .. | $V_{1_n}$ |
|---|---|---|---|

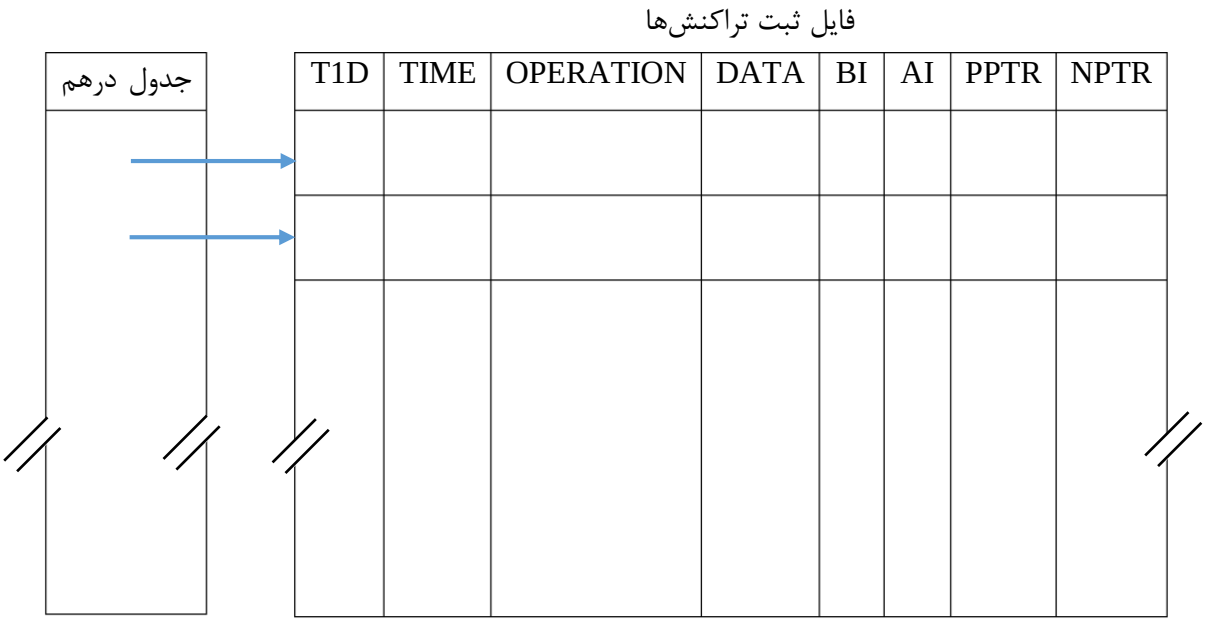
در سطح پیاده‌سازی :




کنجکاوی : آیا مفهوم رکود در سطح نمایش منطقی ، قالب دیگری هم می‌تواند داشته باشد ؟

کنجکاوی : در بخش کنترلی چه فیلدهایی می‌تواند وجود داشته باشد ؟

۲-۱۶ : یک ساختار ممکن برای قالب رکورد فایل ثبت می‌تواند دارای فیلدهای زیر باشد[۱] :




T1D : شناسه تراکنش

[۱] برگرفته از : مفاهیم بنیادی پایگاه داده ، جلد دوم ، از همین قلم

TIME : زمان انجام عمل

OPERATION : نوع عمل

DATA : داده تحت عمل

BI : پیش‌تصویر ( تصویر داده پیش از عمل )

AI : پس‌تصویر ( تصویر داده بعد از عمل )

PPTR : اشاره گر به مدخل قبلی مربوط به یک تراکنش

NPTR : اشاره گر به مدخل بعدی مربوط به یک تراکنش

۲-۱۸ : به قسمت ۶ از گفتار ۱۶ مراجعه شود .

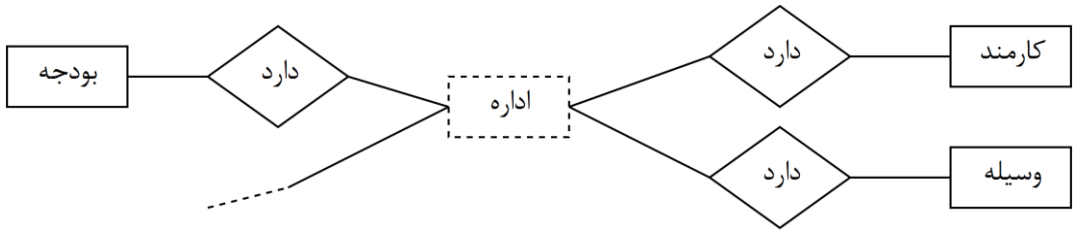
# گفتار ۳

## کنجکاوی‌ها

۳-۱ : به متون درس‌های تجزیه و تحلیل سیستم‌ها و مهندسی نرم‌افزار مراجعه شود .

۳-۲ : نخست توجه داشته باشیم که « مفهوم ذهنی » ( در مقابل « مفهوم عینی » ) با « مفهوم مجازی » فرق دارد . منظور از نوع موجودیت مجازی ( شاید بهتر باشد بگوئیم : نوع موجودیت ضمنی ) ، نوع موجودیتی است که در مرحله‌ی مدلسازی معنایی دیده می‌شود ، امّا در مرحله‌ی طراحی منطقی متناظر ندارد .

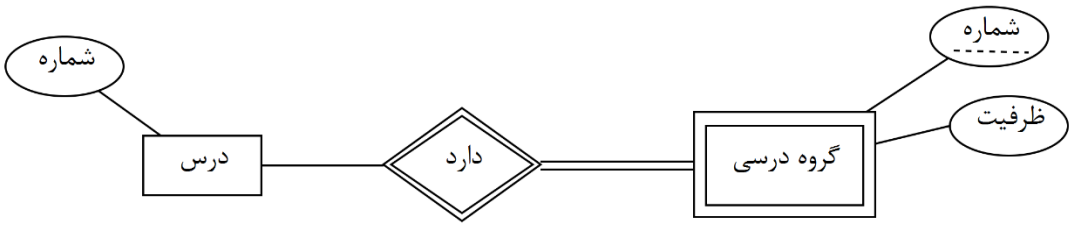
مثال : فرض کنیم می‌خواهیم برای فعالیت‌های درون یک اداره ، یک سیستم پایگاهی ایجاد کنیم . مفاهیمی مانند کارمند ، بودجه ، اتاق کار ، مدیر ، وسیله‌ی کار و . . . بطور عینی مطرح‌اند ، امّا خود مفهوم اداره ، هرچند مطرح است ، امّا نه به عنوان یک مفهوم عینی به گونه‌ای که نوع موجودیت عینی متناظر داشته باشد . این نوع موجودیت را در مستطیل با اضلاع خط‌چین نشان می‌دهیم :




توجه داریم که با این برداشت نسبت به مفهوم اداره ، این مفهوم در مرحله‌ی طراحی عنصر متناظر خاص خود را ندارد .

۳-۳ : مفهوم پیشنیاز درس می‌تواند صفت نوع موجودیت درس باشد ، البته چند مقداری است .

۳-۴ : مفهوم شماره‌ی گروه درسی می‌تواند صفت نوع موجودیت درس باشد ، البته چند مقداری است . این مفهوم را می‌توان با استفاده از نوع موجودیت ضعیف هم مدلسازی کرد :



البته اگر تنها صفت شماره‌ی گروه درسی در محیط مورد نظر باشد ، می‌توان آنرا صفت نوع موجودیت درس در نظر گرفت .

۳-۵ : این صفت در ظاهر ساده است امّا در معنا مرکب است ، یعنی هر جزء آن در محیط ، معنای خاص و کاربرد خود را دارد . در اینجا این سوال مهم ( بویژه در عمل ) مطرح می‌شود که در مرحله‌ی طراحی چگونه باید چنین صفتی را نمایش داد : آیا باید مثلاً با یک ستون جدول نمایش داد یا با چند ستون ؟

کنجکاوی : پاسخ چیست ؟

۳-۶ : اگر مقادیر مجموعه صفات تشکیل دهنده‌ی صفت مرکّب چنان باشند که بتوان آنها را به نمونه‌های متمایز یک نوع موجودیت منتسب کرد ، و حداقل یک نوع ارتباط بین آن مجموعه صفات و یک نوع موجودیت دیگر تشخیص داد ، در این صورت پاسخ مثبت است .

۳-۷ : صفتی است که جزء مجموعه صفت‌های طبیعی نوع موجودیت در محیط نیست و خود مدلساز آنرا به یک نوع موجودیت منتسب می‌کند . مانند شماره‌هایی که به نمونه موجودیت‌ها به عنوان « شناسه » می‌دهیم ، یا شماره‌ی سطر که گاه در عمل به سطرهای جدول داده می‌شود .

۳-۸ : از دیدگاه تئوریک پاسخ مثبت است . البته در عمل هم مطرح است ، مثلاً صفت میزان افزایش حقوق سالانه نوع موجودیت کارمند ، مشتق است و اگر طی یک سال ، به هر دلیلی ، بیش از یک بار افزایش حقوق داشته باشیم ، چند مقداری می‌شود .

برای صفت مرکّب مشتق می‌توان مدت زمان سپری‌شده از آخرین مراجعه‌ی بیمار به بیمارستان ( یا پزشک خانواده و یا موارد مشابه ) را مثال زد . نخست آنکه اگر مدت زمان را برحسب « روز – ماه – سال » در نظر بگیریم ، صفت مرکّب خواهد بود . همچنین با توجه به اینکه این صفت از روی تاریخ آخرین مراجعه‌ی بیمار قابل‌محاسبه می‌باشد ، یک صفت مشتق محسوب می‌شود . در نتیجه یک صفت مرکّب و مشتق خواهد بود .

۳-۹ : پاسخ به این سؤال بویژه با مطالعه‌ی گفتار دهم بدست می‌آید . در اینجا به برخی از محدودیت‌ها اشاره می‌کنیم :

- محدودیت میدانی (دامنه‌ای)
- محدودیت نمایشی
- محدودیت پردازشی

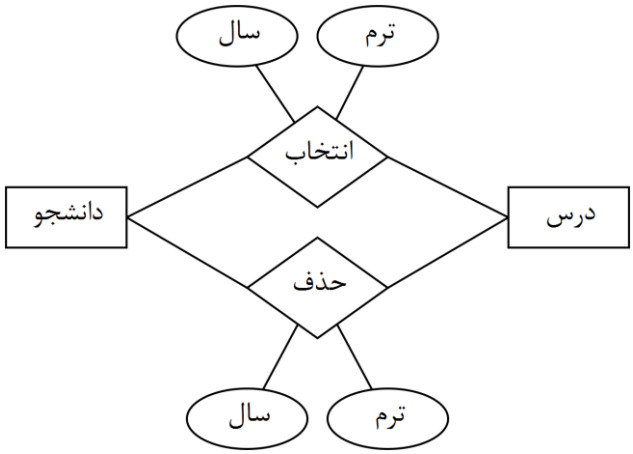
- محدودیت وابستگی به صفت ( های ) دیگر ، از جمله وابستگی تابعی ( گفتار پانزدهم ) ، وابستگی شمول ( $\{ B_{values} \} \subseteq \{ A_{values} \}$ ) .
- محدودیت مدیریتی

ذکر مثال در هر مورد را به خواننده وا می‌گذاریم .

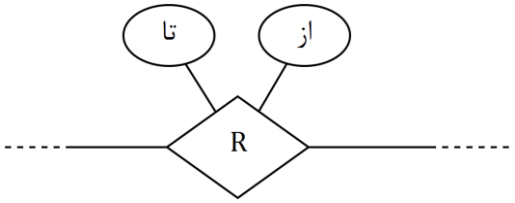
۳-۱۰ : اگر در برنامه‌ی کاربردی به هر یک از اجزاء $YY$ و $T$ بطور مستقل نیاز نباشد ، یا بسامد پاسخگویی به این نیاز خیلی پائین باشد ، پاسخ مثبت است .

۳-۱۱ : بله

- مثال :




- مثال : صفات زمانی که معمولاً به نوع ارتباط‌ها داده می‌شود :




۳-۱۲ : بله ، بویژه وقتی که چندی نوع ارتباط $M : N$ باشد ، هرچند در نمودار $ER$ به صورت تک مقداری نمایش داده می‌شود .

۳-۱۳ : اساساً مفهوم تاریخ و مفاهیم مشابه آن ، مانند ترم ، سال و . . . بطور معمول صفت در نظر گرفته می‌شوند ، امّا در کاربردهای خاص می‌توان آنها را نوع موجودیت دید ، به شرط آنکه بین این نوع موجودیت و نوع موجودیت ( های ) دیگر حداقل یک نوع ارتباط قابل شناسایی باشد .

مفهوم گروه درسی را می‌توان به صورت نوع موجودیت ضعیف مدلسازی کرد .

۳-۱۴ : برای پاسخ این سؤال به گفتار دهم ، بحث قواعد ( محدودیت‌های ) جامعیتی مراجعه شود . امّا از هم اینک باید توجه داشته باشیم که مجموعه‌ی محدودیت‌های معنایی محیط در سه مرحله قابل نمایش یا اِعمال هستند :

- برخی از محدودیت‌ها در مرحله‌ی مدلسازی معنایی قابل نمایش‌اند .
- برخی دیگر در مرحله‌ی طراحی منطقی پایگاه داده‌ها اِعمال می‌شوند .
- بالاخره برخی هم در مرحله‌ی پیاده‌سازی ( در برنامه‌های کاربردی ) اعلان یا اِعمال می‌شوند .

۳-۱۵ : معمولاً خیر ، بلکه مدلسازی باید به گونه‌ای باشد که این مفاهیم با نوشتن برنامه‌های مناسب ، توسط سیستم پایگاهی « تولید » شوند .

۳-۱۶ : خیر ، ممکن است $M : N$ و در حالت خاص $1 : 1$ هم باشد .

۳-۱۷ : بله ، به پاسخ تمرین ۲۸ از گفتار ۳ مراجعه شود .

۳-۱۸ : پاسخ مثبت است به شرط آنکه بسامد مراجعه به نوع ارتباط دوگانی پائین و بسامد مراجعه به نوع ارتباط سه‌گانی ( یا بیشتر ) بالا باشد و ملاحظات کارایی دیگری مطرح نباشد .

۳-۱۹ : به تعریف دو تکنیک دقت شود .

۳-۲۰ : خیر ، نوع موجودیت « مجتمع » را نوع ارتباط بین نوع موجودیت ( های ) شرکت‌کننده در آن نمایش می‌دهد ( که با مستطیل نشان می‌دهیم ) . صفات نوع ارتباط را نمی‌توان به نوع موجودیت ( ها ) منتسب کرد .

۳-۲۱ : خیر ، به پاسخ کنجکاوی ۱۴ مراجعه شود .

۳-۲۲ : در بستگی یکسویه تنها یک رده ، رده‌ی دیگر را « می‌بیند » و از آن استفاده می‌کند . در بستگی دو سویه هر دو رده یکدیگر را « می‌بینند » و می‌توانند از یکدیگر استفاده کنند .

۳-۲۳ : در بیان غیرصوری مجموعه‌ای است از امکانات ساختاری ، برزشی (دستکاری، عملیاتی ) و کنترلی ، به گونه‌ای که کاربر بتواند داده‌هایش را به کمک عنصر ( عناصر ) ساختاری نمایش دهد ، به کمک دستورهایی در داده‌هایش عملیات انجام دهد و روی آنها کنترل اِعمال کند . ( به گفتار دهم مراجعه شود ) .

# تمرینات درون گفتار

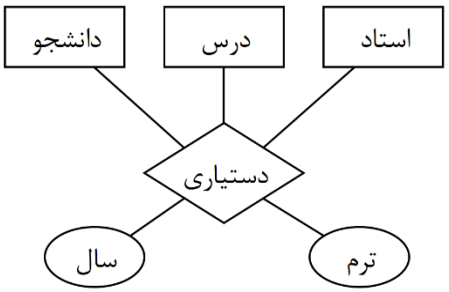
۳-۱ : سابقه کار کارمند ، دوره‌های تحصیلی کارمند ، سابقه پرداخت قسط‌های وام .

۳-۲ : صفت ساده چندمقداری : شماره تلفن استاد ، شماره‌ی پیش نیازهای درس ، عنوان مدارک تحصیلی استاد .

صفت مرکب چندمقداری : اثر منتشره استاد ، سابقه تحصیل کارمند ، منابع درس .

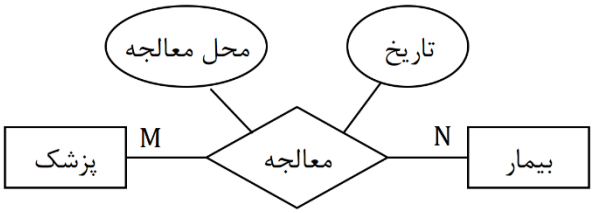
۳-۴ : نوع ارتباط‌های : همکار بودن کارمند ، تشکیل شدن قطعه از قطعات ، مرجع بودن کتاب برای کتاب .

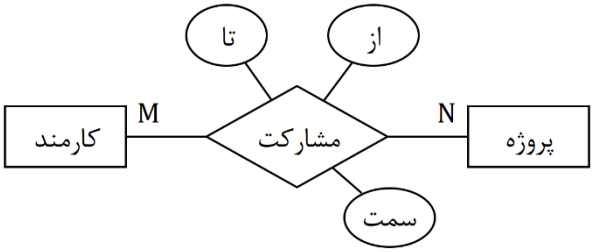
۳-۵ :



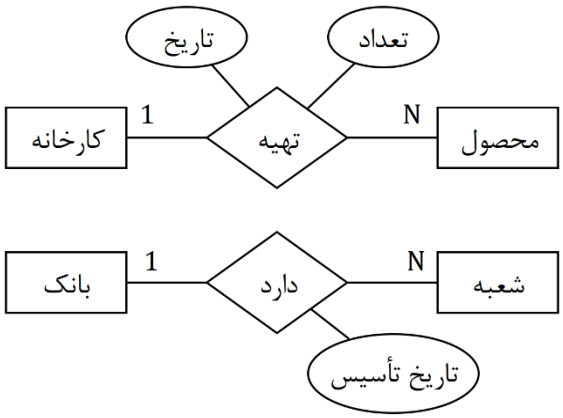

۳-۷ :

- چندی $M:N$

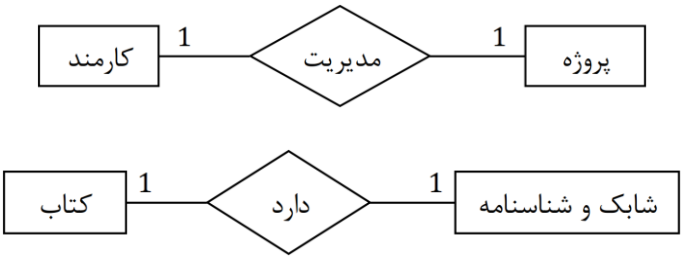



- چندی $1:N$

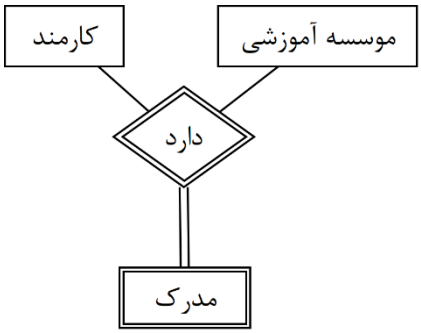



- چندی 1:1

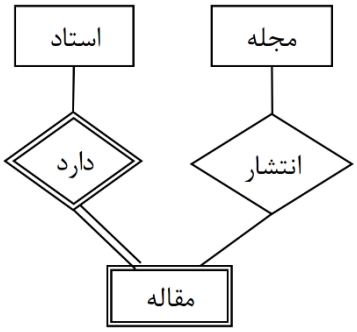



۳-۹ :

مثال برای نمودار شکل ۳-۱۰ :




مثال برای نمودار شکل ۳-۱۲ :

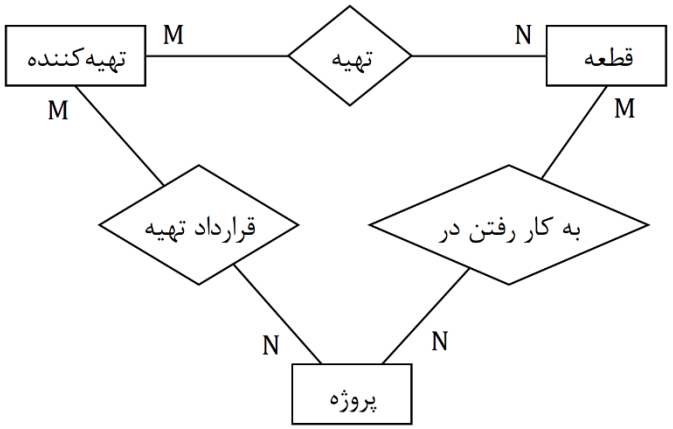


۳-۱۰ : صفات نوع موجودیت ضعیف «انتخاب» :

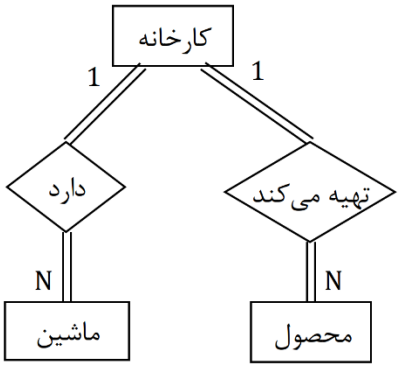
شماره‌ی گروه درسی ، زمان ارائه ، شماره‌ی کلاس ، ظرفیت گروه درسی.

۳-۱۱ : برای هرمورد یک مثال ذکر می‌کنیم. ذکر مثال دیگر را به خواننده وامی‌گذاریم.

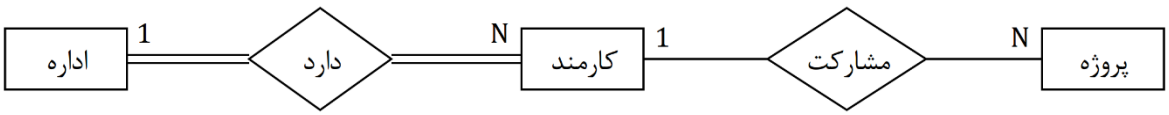
مثال دام پیوندی حلقه‌ای :




مثال دام چندشاخه :




مثال دام گسل :



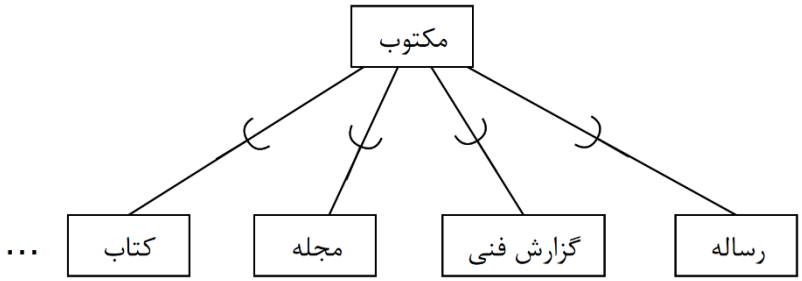

۳-۱۲ :

- ماشین و اجزایش

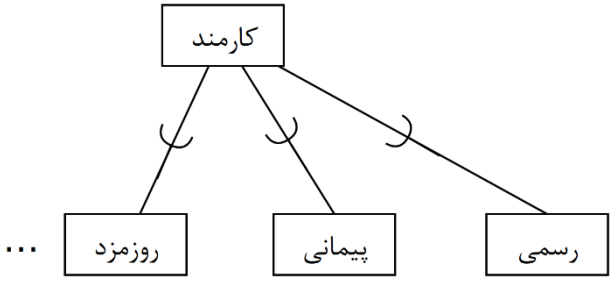
- تیم ورزشی دارد : ورزشکار ، داور ، مربّی ، کارمند و ... .

۳-۱۳ : یک مثال ذکر می‌کنیم. ذکر مثال دیگر را به خواننده وامی‌گذاریم :

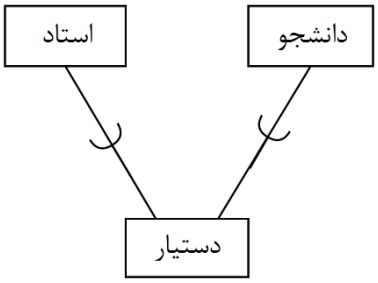
- تخصیص




- تعمیم : کارمند رسمی ، کارمند روزمزد ، کارمند پیمانی ، کارمند رسمی. نوع موجودیت کارمند را در سطح بالا در نظر می‌گیریم به عنوان نوع موجودیت عام.

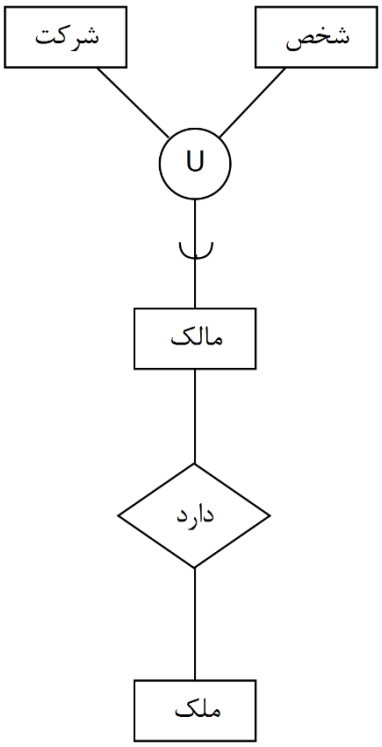



۳-۱۴ : یک مثال ذکر می‌کنیم :

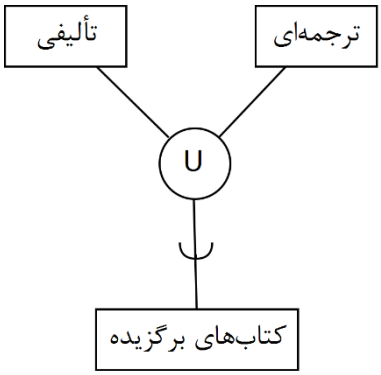



۳-۱۵ :

- مثال ۱ :

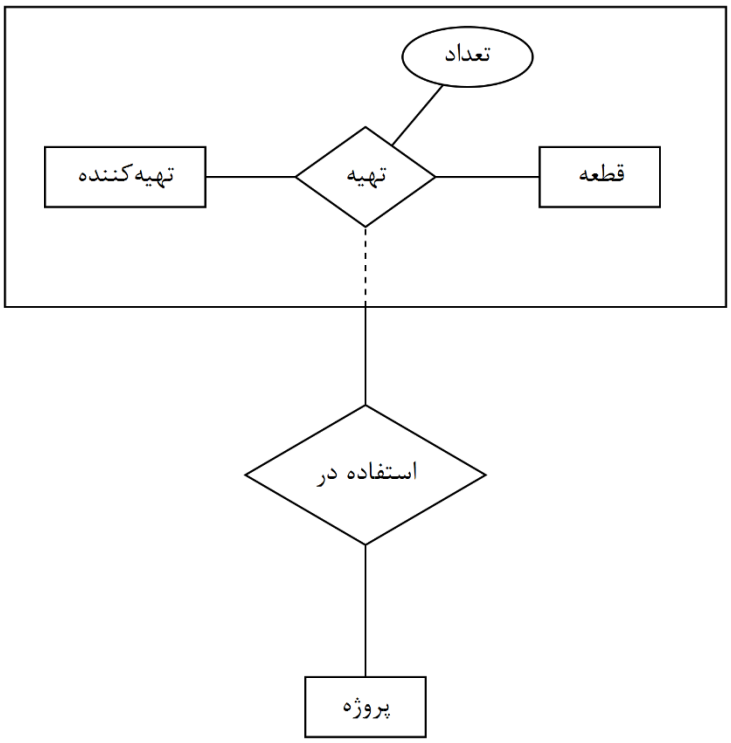



- مثال ۲ :

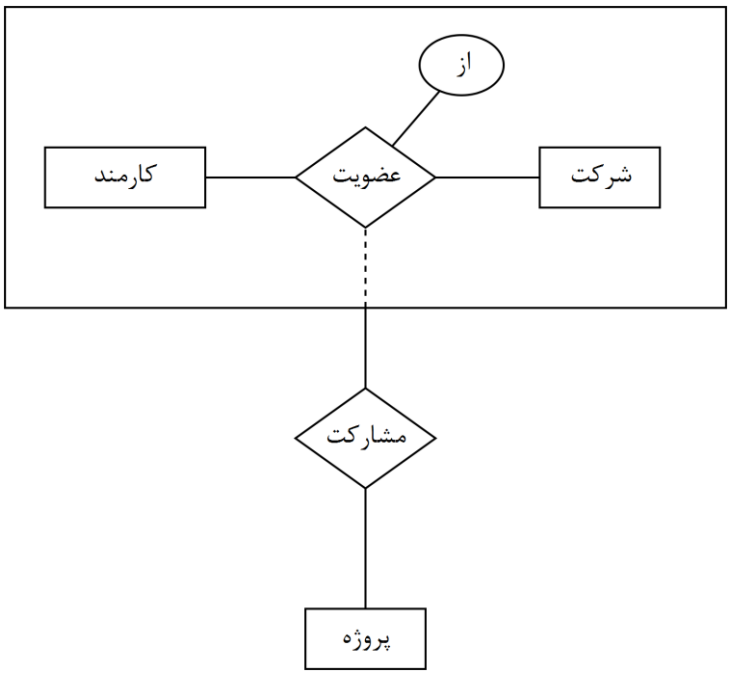



۳-۱۶:

- مثال ۱ :


تعداد
تهیه‌کننده
تهیه
قطعه
استفاده در
پروژه

- مثال ۲ :

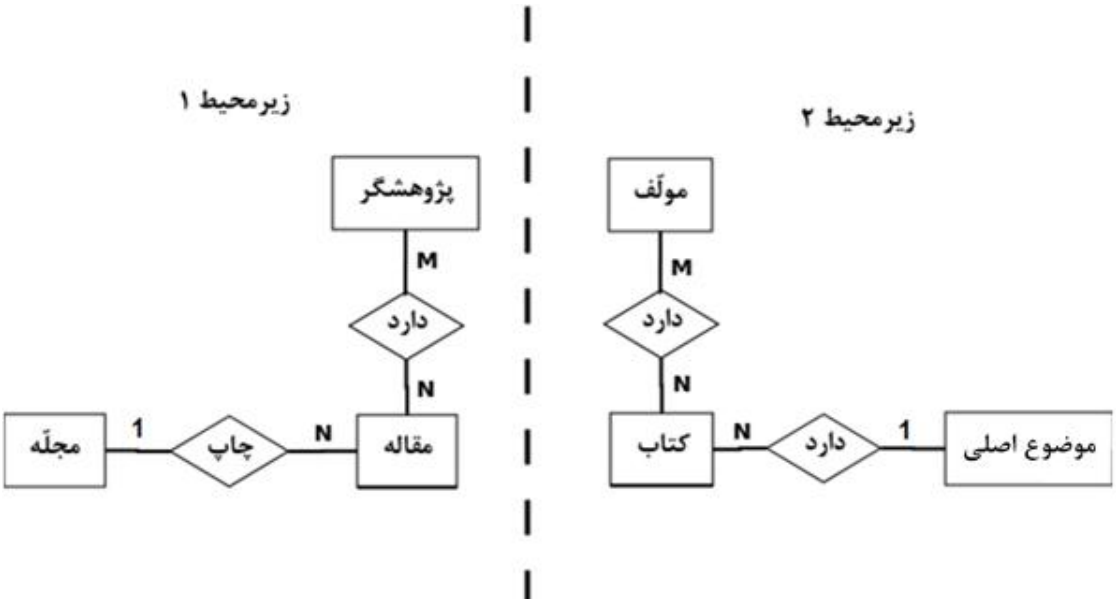

از
کارمند
عضویت
شرکت
مشارکت
پروژه

# تمرینات پایان گفتار [۲]

۳-۱ : با استفاده از ارتباط IS-A-PART-OF می‌توان مدلسازی کرد .

مثلاً :
- کتاب «دارد» بخش
- بخش «دارد» فصل
- فصل «دارد» عنوان داخلی اصلی
- عنوان داخلی «دارد» عنوان داخلی درونی
- عنوان داخلی درونی «دارد» موضوع

۳-۲ : ابتدا هر یک از زیرمحیط های ۱ و ۲ را به صورت زیر و جداگانه مدل می کنیم :

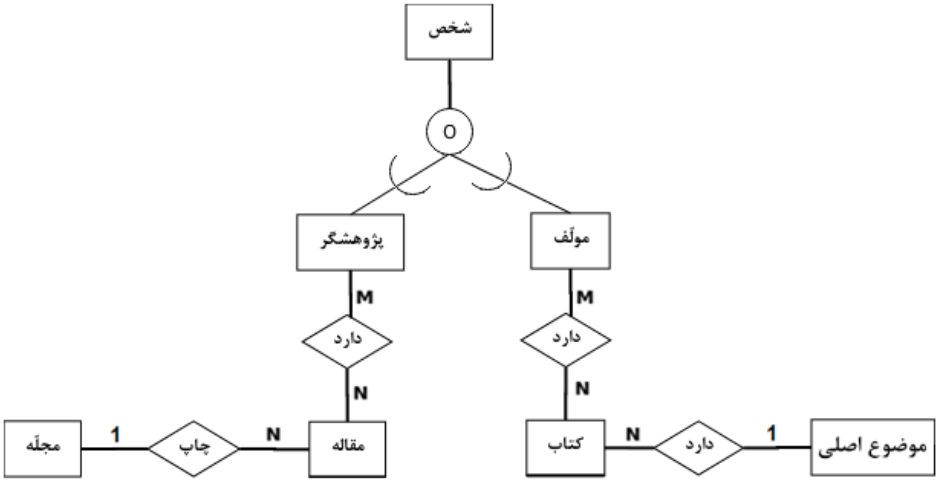



حال با ادغام دو زیر محیط ، مدلسازی زیر را برای دو محیط ارائه می دهیم :

[۲] توجه :

۱- مدلسازی‌های انجام شده در تمرین‌های این گفتار نه کامل اند و نه تنها مدلسازی ممکن .

۲- در پاسخ‌ها ، جزئیات مدلسازی مانند چندی‌ها ، نوع مشارکت ، مجموعه صفات نوع‌موجودیت‌ها ، و نوع ارتباط‌ها نشان داده نشده‌اند . به خواننده توصیه می‌شود این جزئیات را مشخص کند .

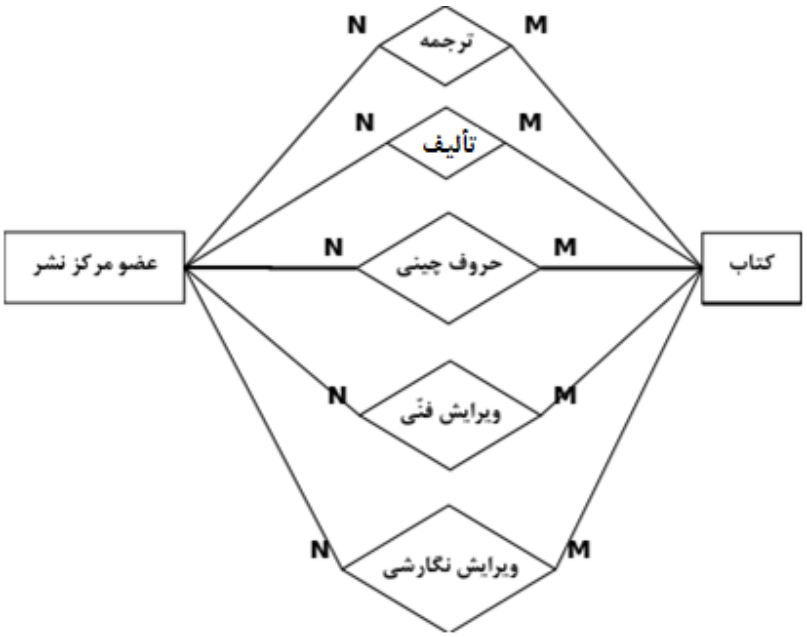



۳-۳ : با ایده گرفتن از روش UML یا روش‌های دیگری که احیاناً در درس(های) دیگر می‌بینید ، به این سوال پاسخ دهید .

۳-۴ : به متن گفتار مراجعه شود .

۳-۵ : نوع ارتباط زیر را می توان بین نوع موجودیت های کتاب و عضو مرکز نشر در نظر گرفت :

- حروف چینی
- ویرایش نگارشی
- ویرایش فنّی
- تایپ
- ترجمه

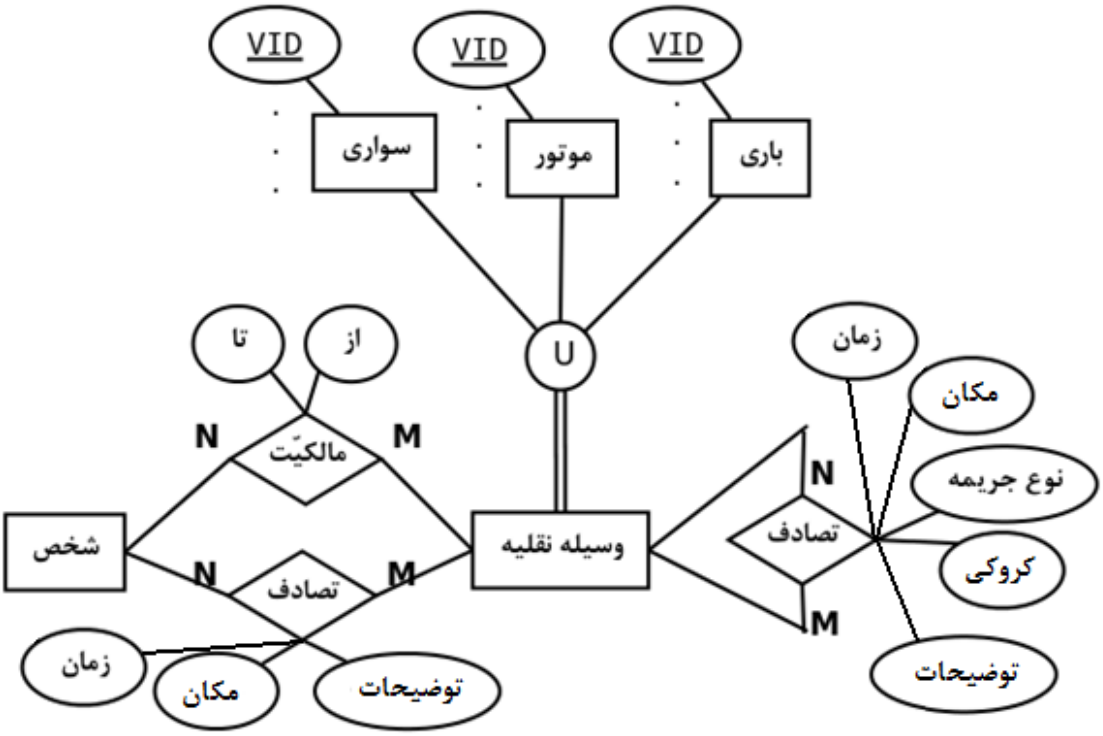



۳-۶ : این مدلسازی با توجه به تکنیک تجزیه دیده شده است .

نخست باید متذکر شویم که کاربر در این محیط ، به هر یک از مفاهیم استان ، شهر ، خیابان ، پلاک ، و کد پستی به صورت جداگانه نیاز داشته است (در غیر این صورت این گونه مدلسازی اشتباه است ) .

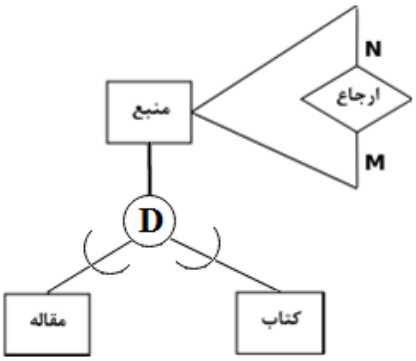
امّا برای بررسی میزان توجیه پذیری این مدلسازی ، باید به بررسی نوع موجودیت های استان ، شهر ، خیابان ، پلاک ، و کدپستی بپردازیم . در صورتی که این نوع موجودیت ها تک صفتی باشند و همچنین نوع ارتباط ( های ) دیگری با نوع موجودیت های محیط نداشته باشند ، می توان هر یک را به صورت یک صفت ساده برای نوع موجودیت نشانی در نظر گرفت و دلیل قابل توجه برای استفاده از تکنیک تجزیه وجود ندارد .

۷- در این محیط ، فرض را بر این می نهیم که وسیله نقلیه عبوری در شهر ، سواری ، موتور ، یا باری هستند که بین آنها ممکن است تصادفاتی بروز کند . همچنین ممکن است بین یک شخص و یک وسیله نقلیه نیز تصادفی رخ دهد . در این محیط همواره جرائم رانندگی ناشی از تصادفات بین وسیله نقلیه و شخص بر عهده مالک وسیله نقلیه است . یک مدلسازی ممکن چنین است :

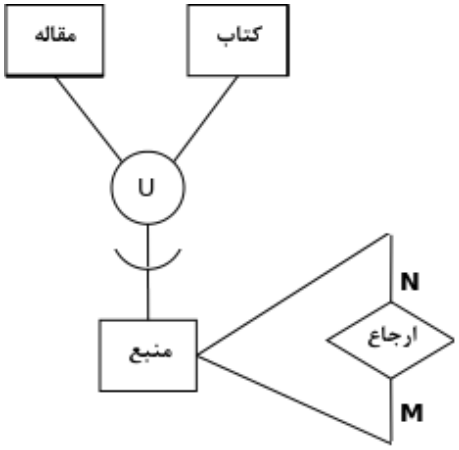



کنجکاوی : مفهوم «مقصّر» را چگونه باید در مدلسازی دخالت داد ؟

۳-۹ : می‌توان با استفاده از تکنیک تخصیص ، چنین مدل کرد :

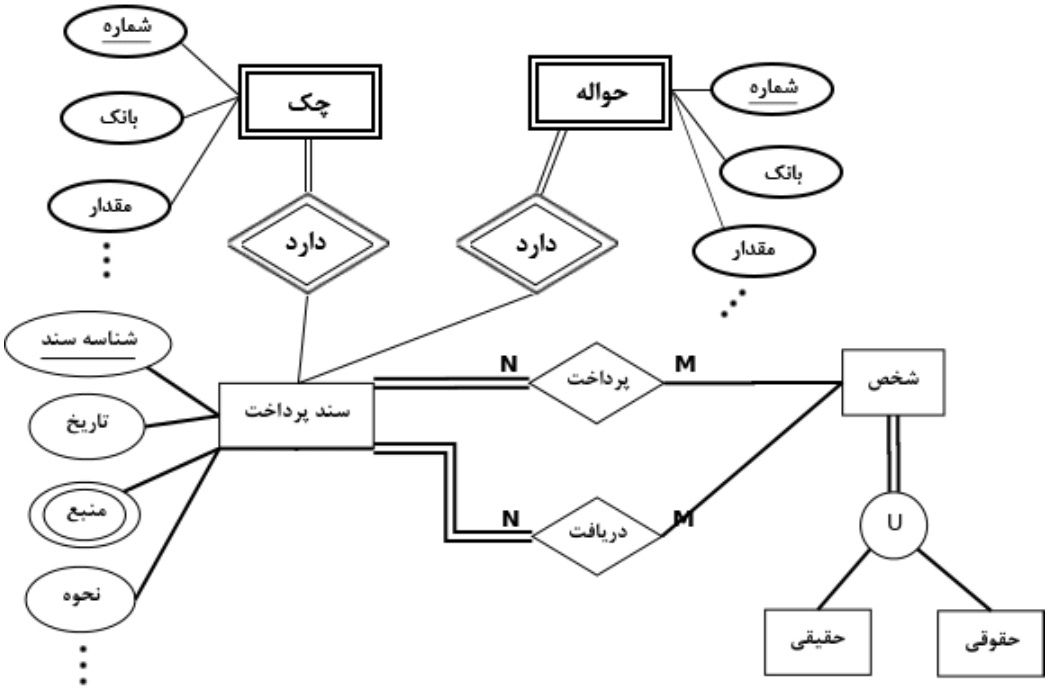



در اینجا این پرسش مطرح است که آیا می‌توان این محیط را با مفهوم زیر نوع اجتماع به صورت زیر مدل کرد ؟ اگر بله ، کدام مدلسازی توجیه پذیرتر است ؟

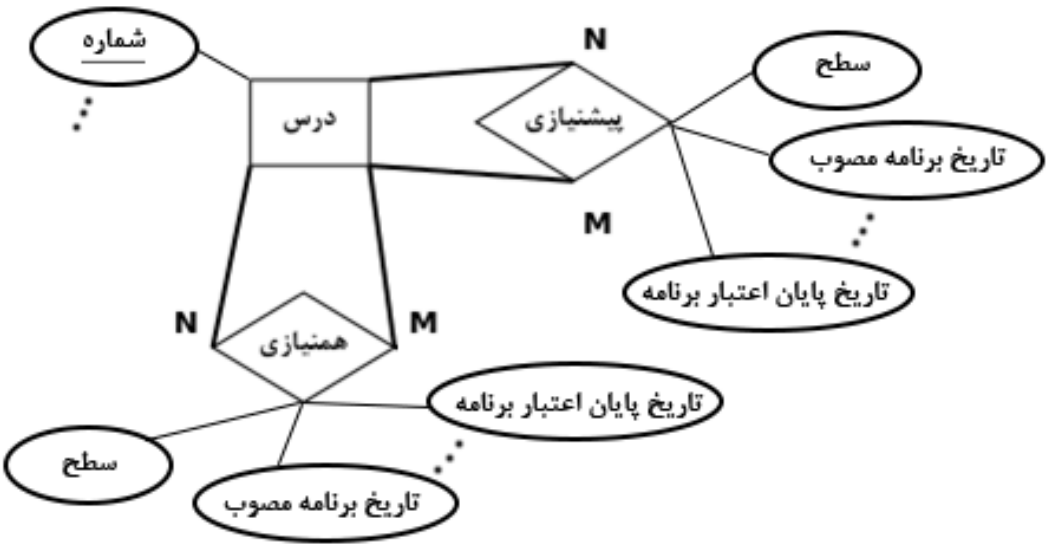



کنجکاوی : صفات نوع ارتباط ارجاع چه می‌تواند باشد ؟

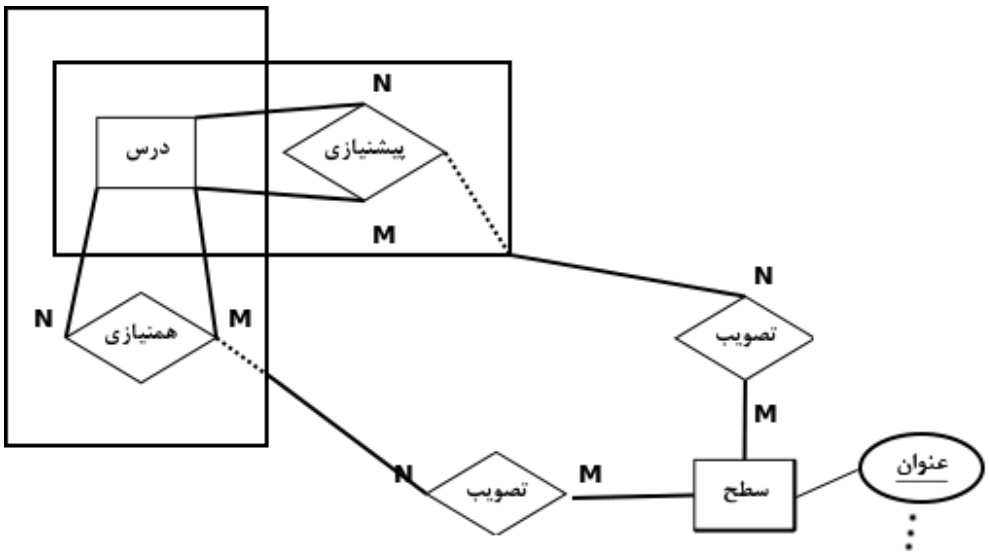
۳-۱۰ : یک مدلسازی ممکن چنین است :



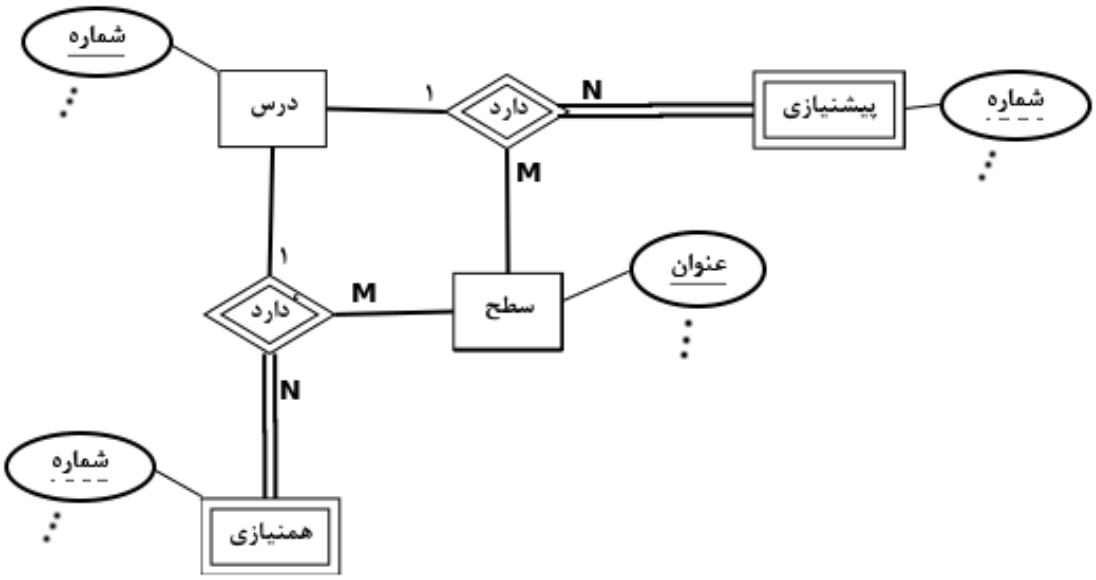

۳-۱۱ : الف ) مدلسازی شماره یک



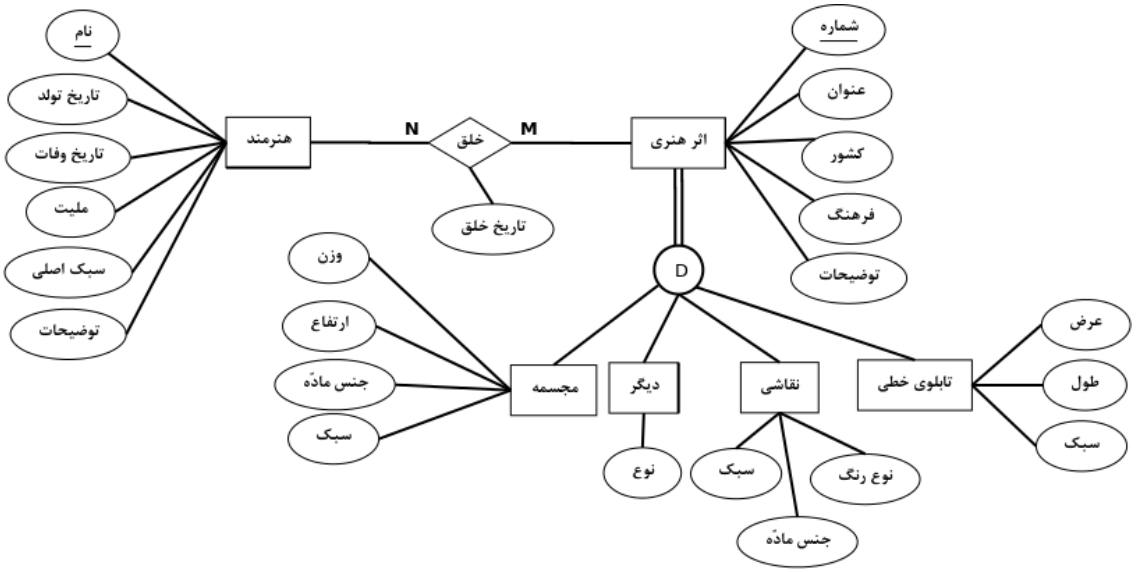

ب ) مدلسازی شماره دو

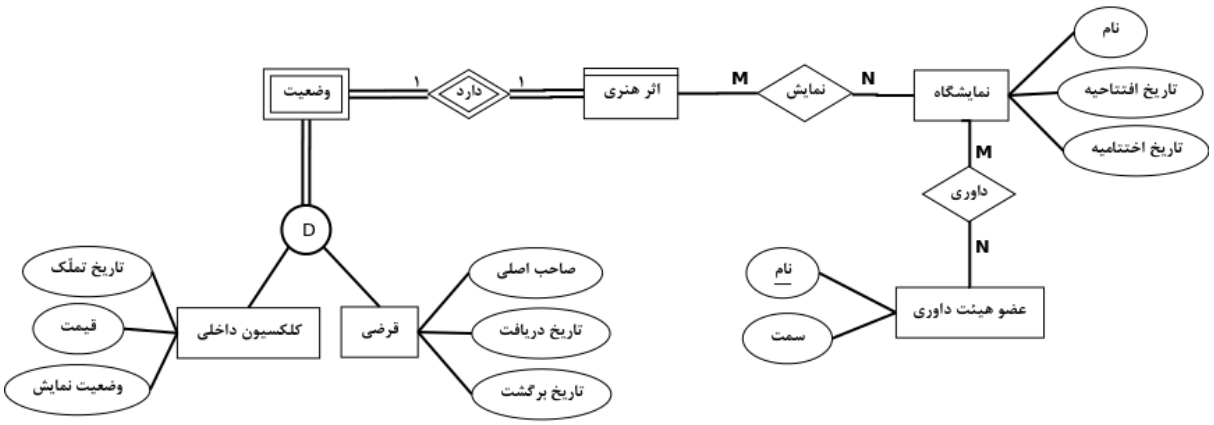


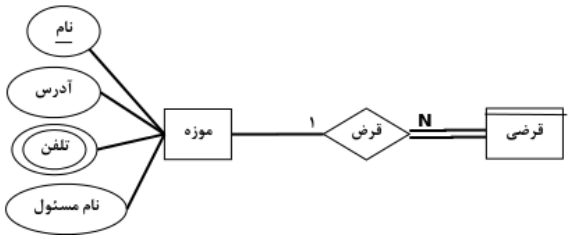

پ ) مدلسازی شماره ۳




۳-۱۲ : یک مدلسازی ممکن چنین است :




۳-۱۳ : در صورتی که معنای نوع ارتباط R2 متفاوت با معنای نوع ارتباط R1 باشد ، نوع ارتباط R2 باید در نظر گرفته شود . زیرا نوع موجودیت های E1 و E3 می توانند مستقل از نوع ارتباط R1 به واسطه نوع ارتباط R2 با یکدیگر در ارتباط باشند . درنتیجه امکان مدلسازی نوع ارتباط R2 با نوع ارتباط R1 وجود ندارد . همچنین تمام نمونه ارتباط های R2 دونوع موجودیتی هستند ، در صورتی که نمونه ارتباط های R3 حتما باید سه نوع موجودیتی باشند . برای

مثال ممکن است نمونه e از نوع موجودیت E1 با نمونه f از نوع موجودیت E3 با یکدیگر در ارتباط باشند ( ارتباط R2 ) و این دو نمونه هیچ گاه در ارتباط R1 مشارکت نداشته باشند . در این صورت اگر بخواهیم R2 را با R1 مدل کنیم ، می بایست یک اطلاع دو موجودیتی را به اطلاع سه موجودیتی تبدیل کنیم که این تبدیل ناممکن و اشتباه است .

۳-۱۴ : اولاً : شرط استفاده از تعمیم عبارتند از :

الف ) داشتن شناسه‌ی مشترک

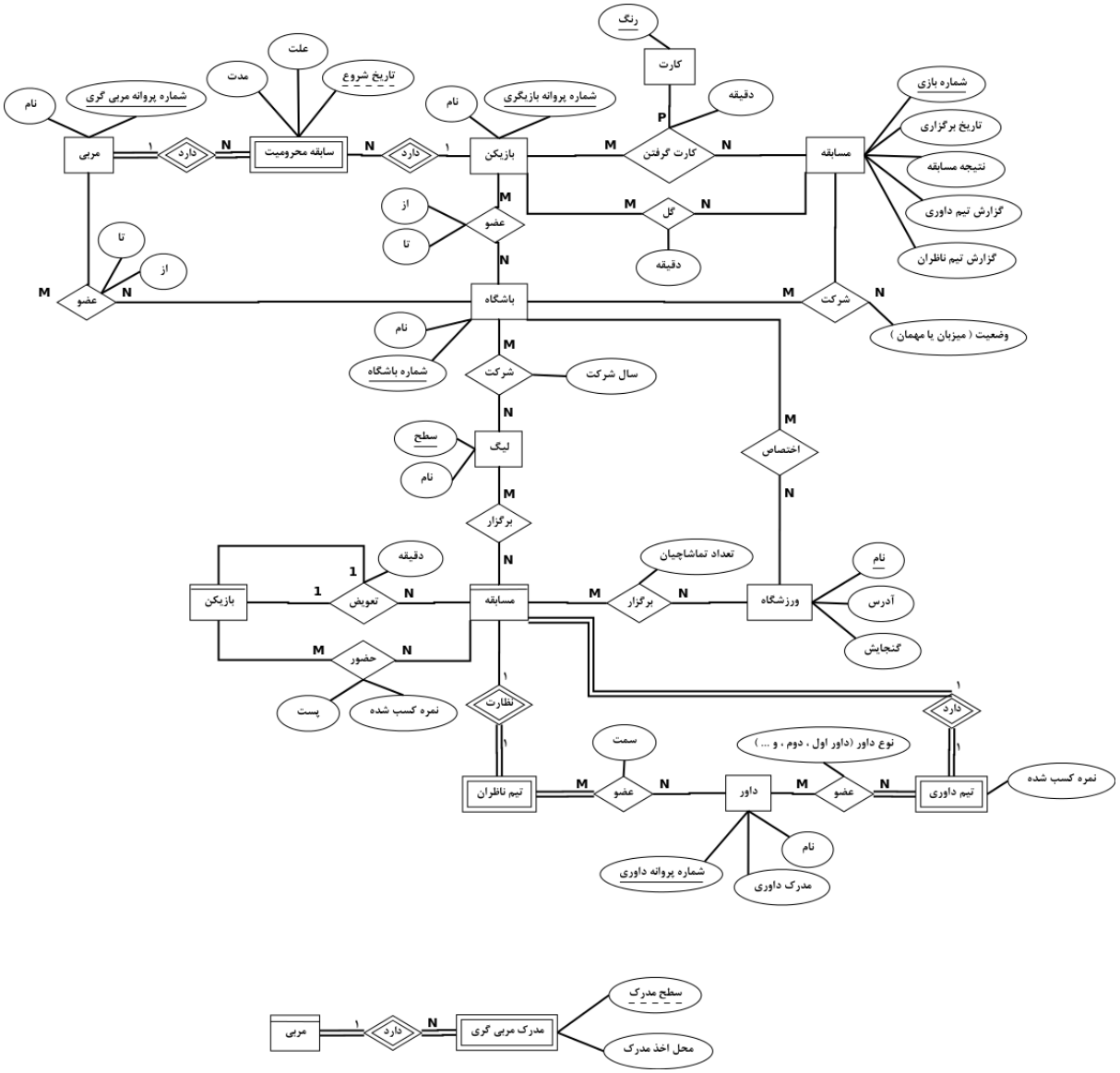
ب ) وجود حداقل دو زیرنوع

پ ) [ شرط ارجحیت ] هرچه صفات مشترک بیشتر ، تعمیم توجیه پذیرتر .

ت ) [ شرط ارجحیت ] وجود نوع ارتباط ( های ) هم معنا بین تمام زیرنوع ها و نوع موجودیت ( های ) دیگر .

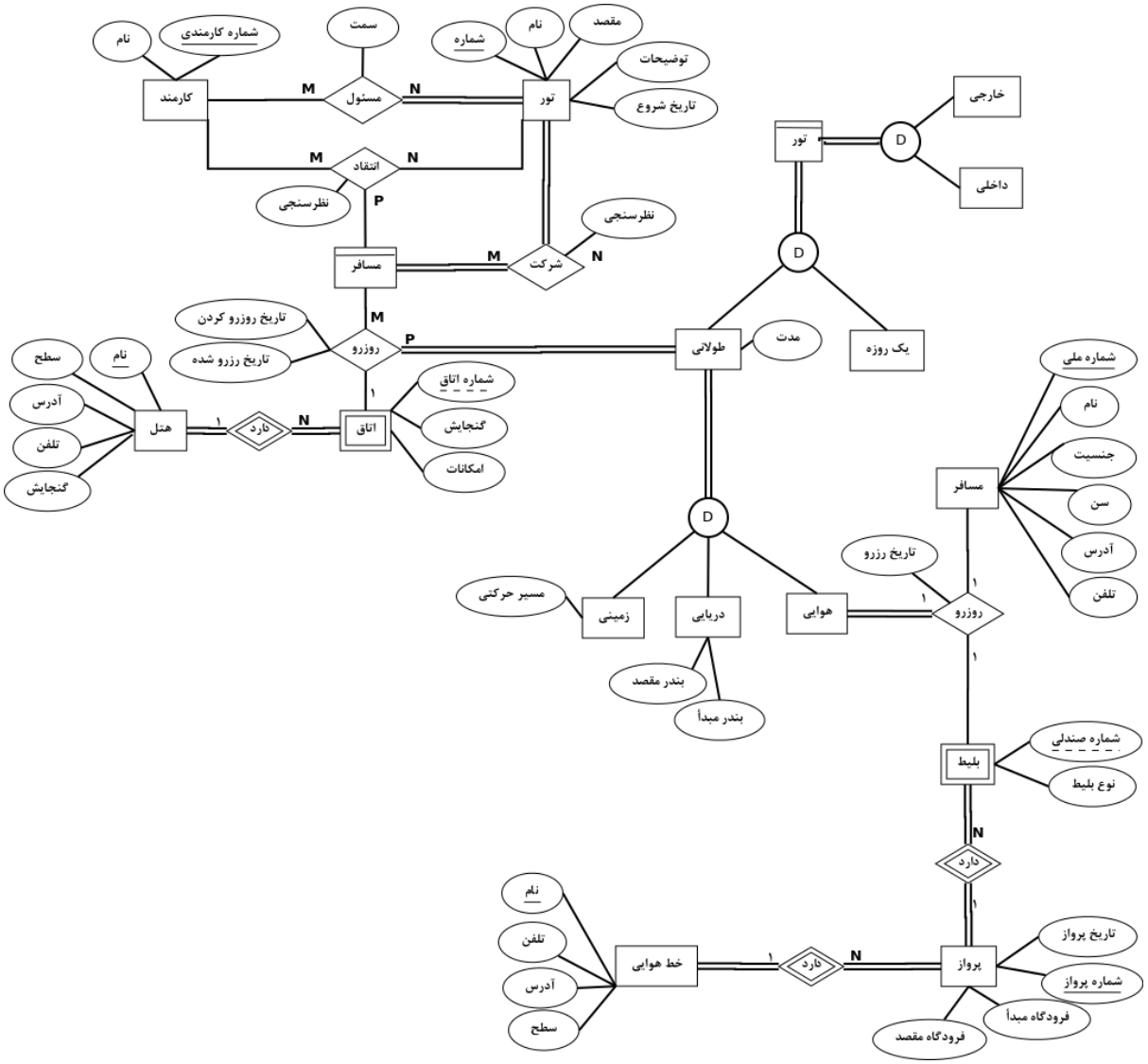
ثانیاً : خیر . در حالت کلّی این کار امکان پذیر نیست . مگر آنکه از یک شناسه ساختگی ( مانند کاری که در زیرنوع اجتماع در حالتی که شناسه زبرنوع ها از یک میدان نباشند ، می کنیم ) کمک گرفته و تعمیم را انجام داد .

۳-۱۵ : الف ) یک مدلسازی ممکن چنین است :



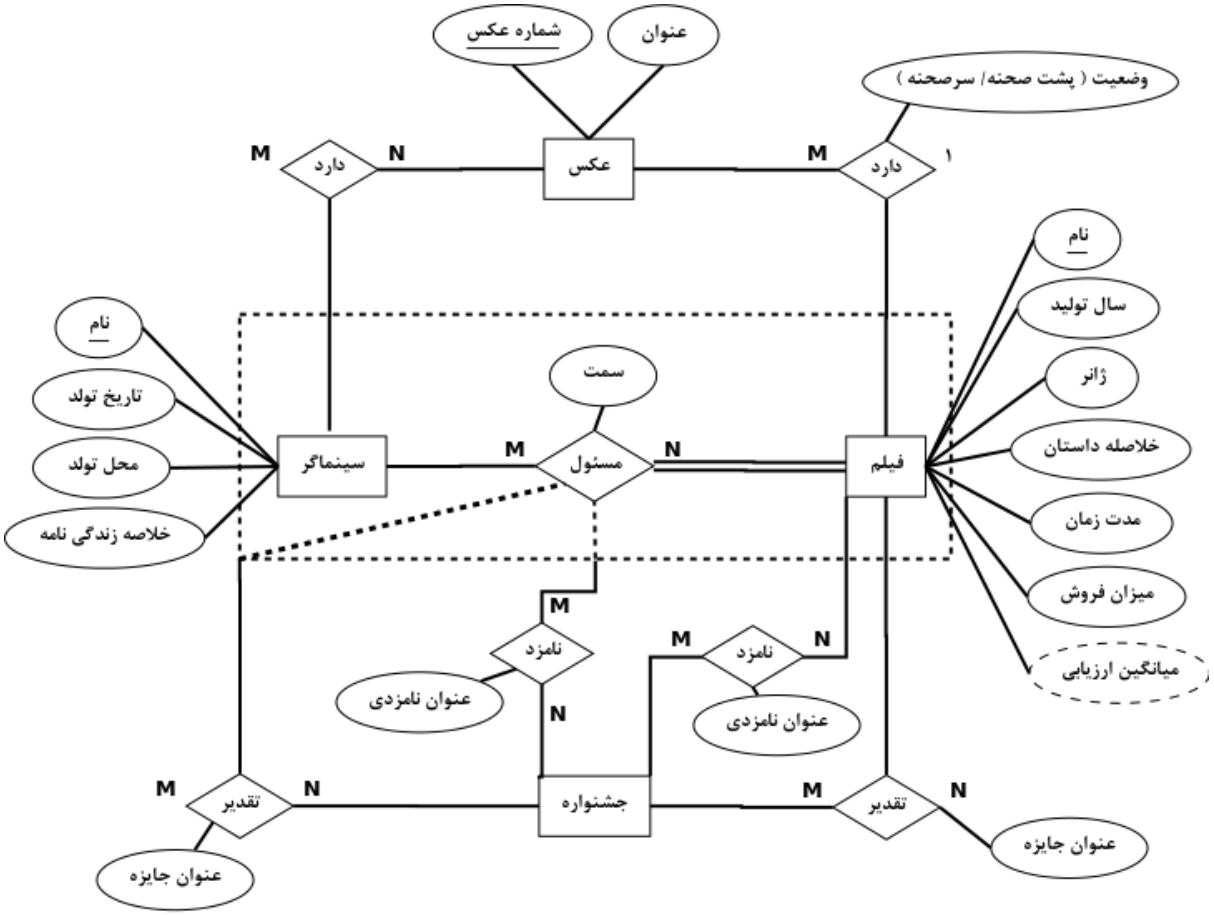

ب ) به خواننده واگذار می‌شود .

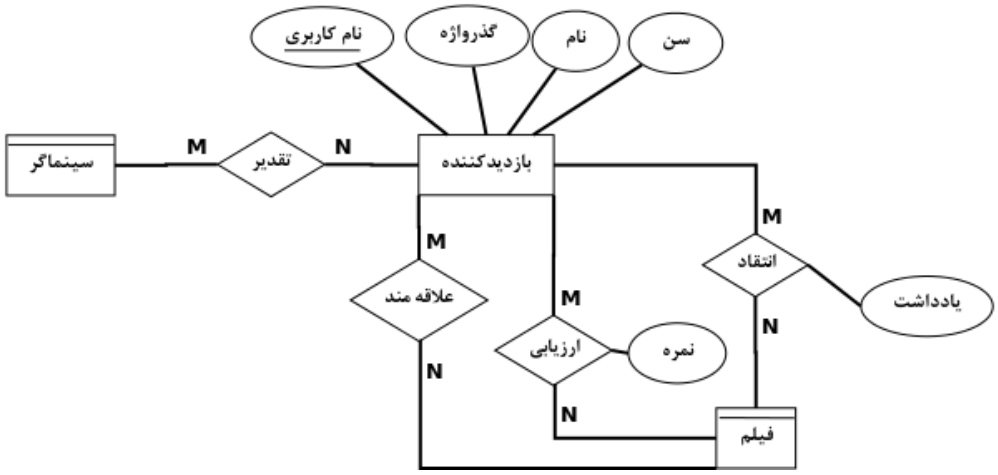
۳-۱۶ : الف ) یک مدلسازی ممکن چنین است :



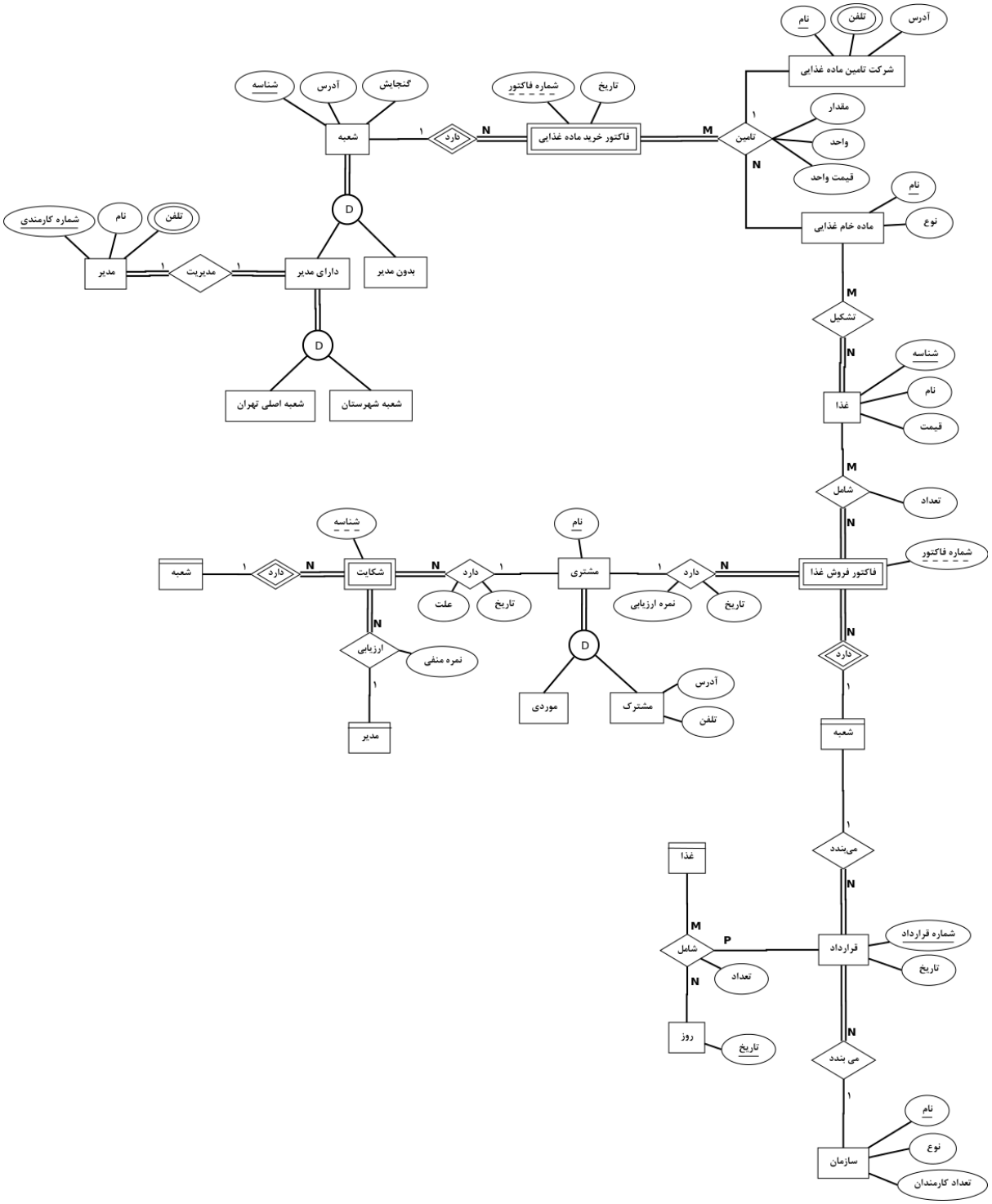

ب ) به خواننده واگذار می‌شود .

۳-۱۷ : یک مدلسازی ممکن چنین است :



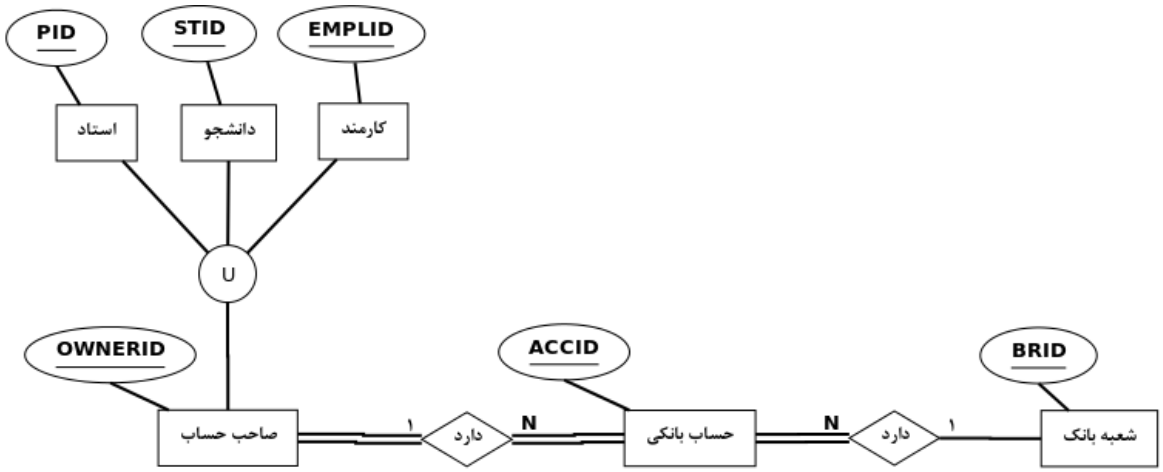
۳-۱۸ : الف ) یک مدلسازی ممکن چنین است :



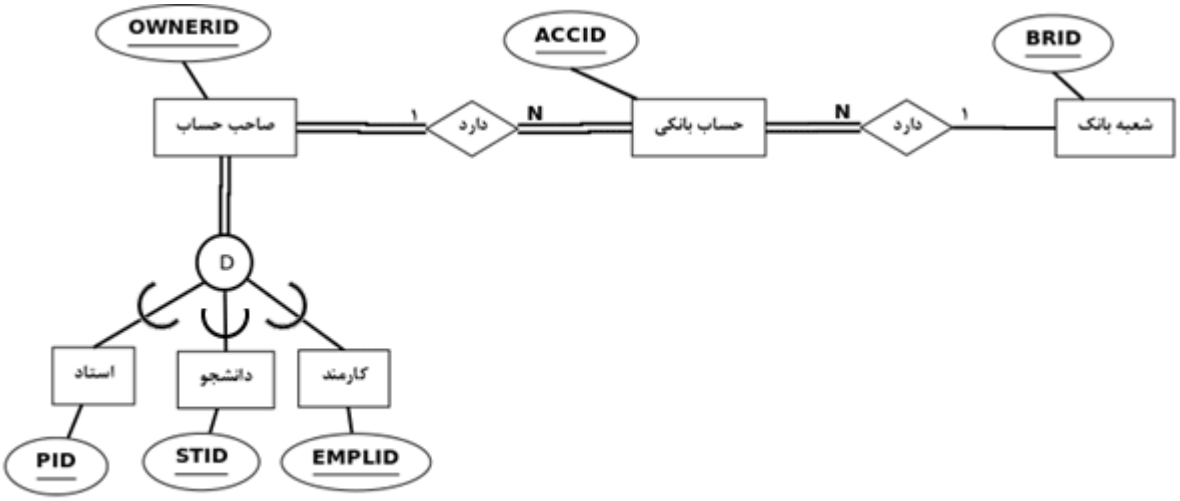

ب ) به خواننده واگذار می‌شود .

۳-۲۱ :

روش اول : با استفاده از مفهوم U-Type :




روش دوم : با استفاده از تکنیک تخصیص :



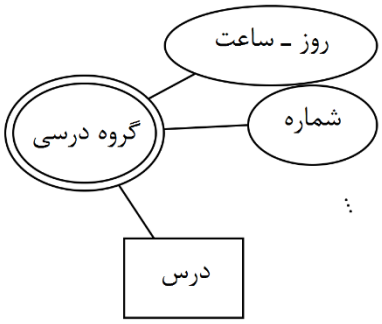

۳-۲۲ : برای بررسی انواع مدلسازی های ممکن ، به نکات زیر دقت کنید :

الف ) گروه درسی

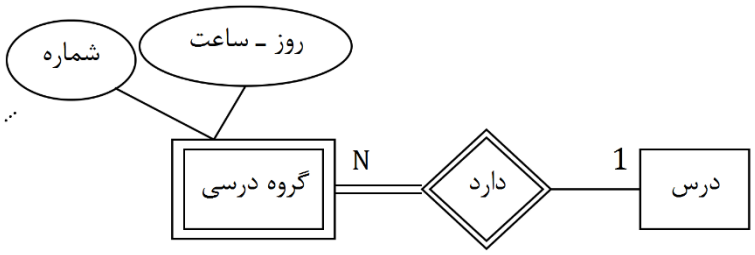
مفهوم گروه درسی را به دو صورت می توان مدلسازی کرد : ۱- صفت چند مقداری درس ۲- نوع موجودیت ضعیف برای نوع موجودیت درس

در صورتی که گروه درسی دارای بیش از یک صفت باشد ، در حالت اول تبدیل به یک صفت چند مقداری مرکّب برای درس خواهد شد . در چنین شرایطی شاید برتری با حالت دوم ، یعنی در نظر گرفتن یک نوع موجودیت ضعیف برای

گروه درسی باشد . به خصوص اگر در دیگر بخش های مدلسازی ظاهر می شود . شکل زیر دو حالت فوق از مدلسازی را نمایش می دهد :




مدلسازی با صفت مرکّب چندمقداری

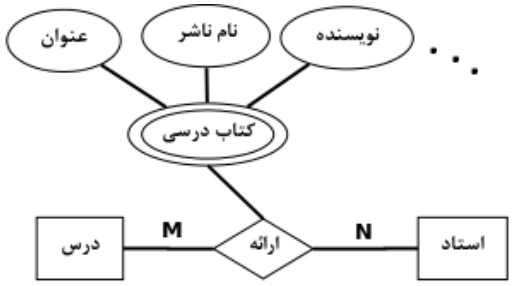



مدلسازی با نوع موجودیت ضعیف

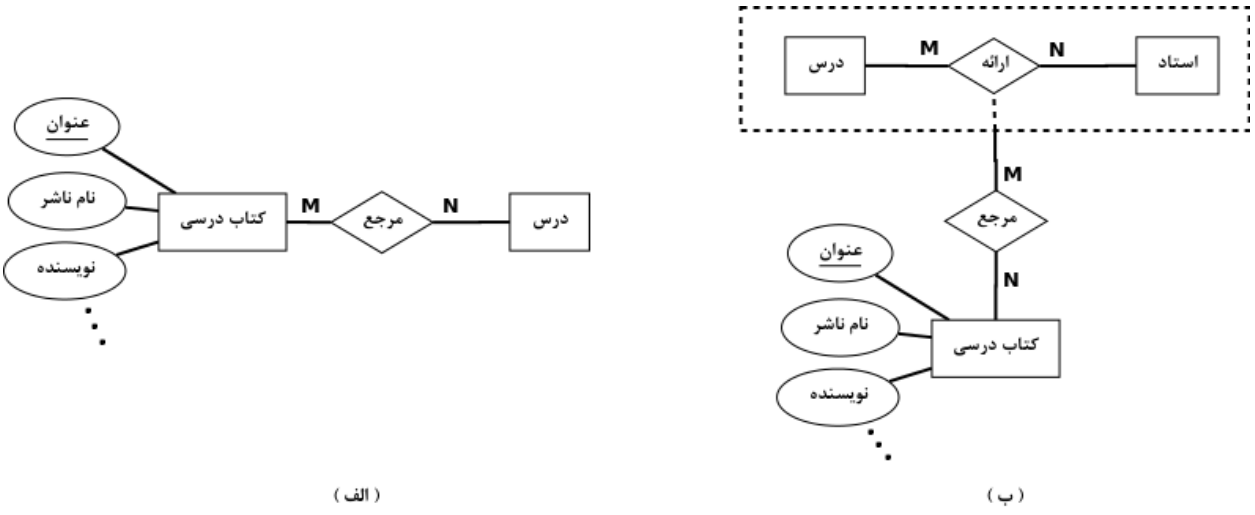
ب ) کتاب درسی

مفهوم کتاب درسی را نیز به دو صورت می توان در نظر گرفت : ۱- صفت مرکّب ۲- نوع موجودیت مجزّا . همچنین در هر یک از حالات ذکر شده ( چه صفت و چه نوع موجودیت مجزّا ) دو حالت را می توان متصور بود : ۱- کتاب درسی مستقل از استاد برای درس تعیین می شود ۲- بسته به استادی که درس را ارائه می کند ، کتاب درسی می تواند متفاوت باشد . لذا چهار حالت متمایز از هم می تواند وجود داشته باشد . در شکل های زیر مدلسازی به چهار روش فوق را مشاهده می کنید :

**مدلسازی با صفت مرکب چند مقداری برای نوع موجودیت درس**

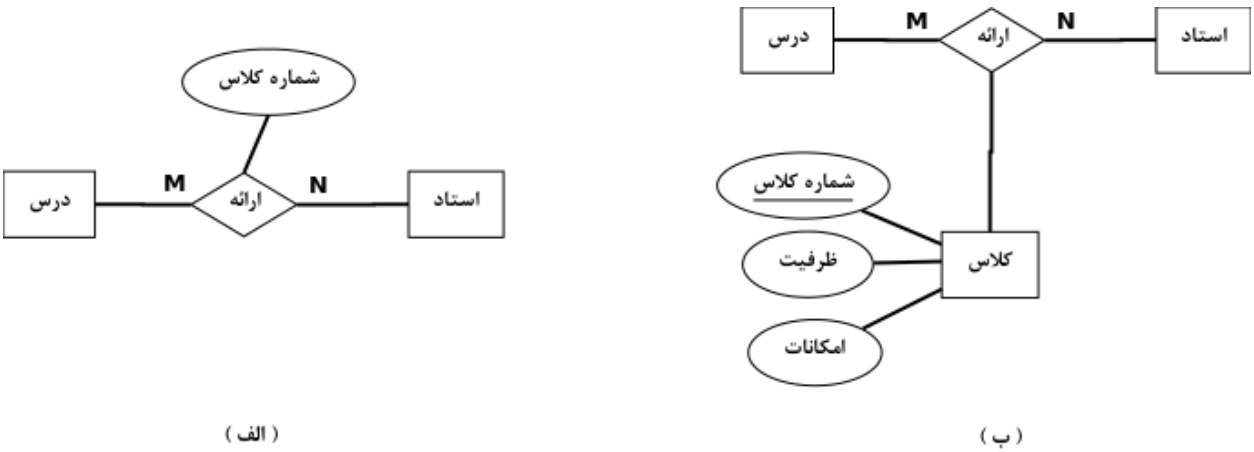



**مدلسازی با صفت مرکب چند مقداری برای نوع ارتباط ارائه شدن درس**



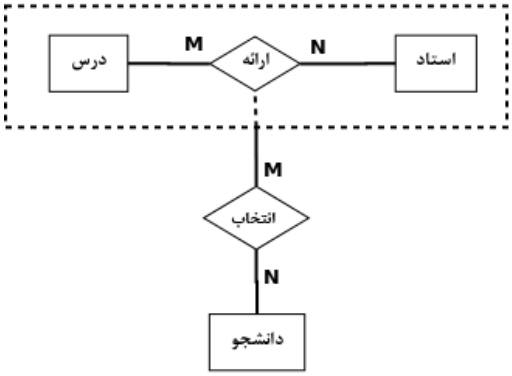

پ ) کلاس درس

مفهوم کلاس درس را نیز می توان به دو صورت مدلسازی کرد : ۱- صفت مرکّب ۲- نوع موجودیت مجزّا . در شکل زیر دو گونه مدلسازی دیده می‌شود :

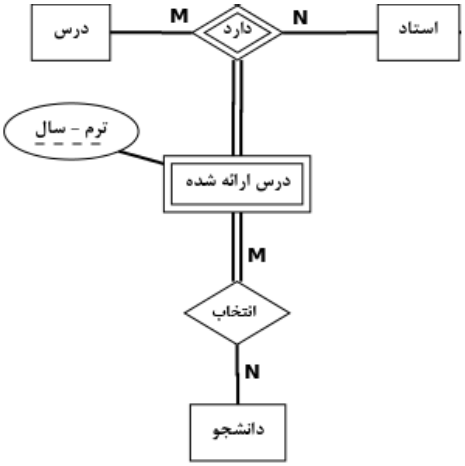



ت ) استاد – درس – دانشجو

در کتاب درسی چهار گونه مدلسازی این سه نوع موجودیت آورده شده است . اما با توجه به نوع موجودیت های دیگر و روش های مدلسازی بیان شده در مراحل قبل ، دو گونه از مدلسازی این سه نوع موجودیت که با دیگر نوع موجودیت های مطرح ، نظیر کتاب درسی و گروه درسی و کلاس درس ، و نوع مدلسازی های مطرح شده ، سازگاری بیشتری دارد ، در زیر آورده شده است . اولین مدلسازی به کمک تکنیک تجمیع است ( که مشابه آن در کتاب موجود است ) و گونه دیگر به کمک نوع موجودیت ضعیف است .

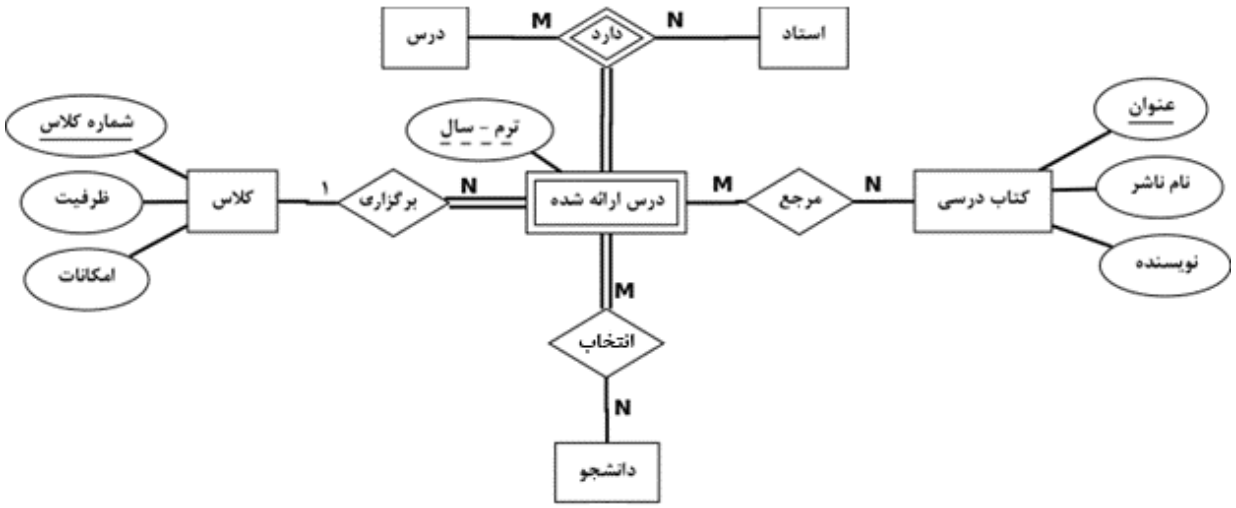



( الف )




( ب )

با توجه به آنچه گفته شد ، این محیط را می‌توان به گونه های مختلفی ( با ترکیب حالات مختلف مدلسازی مطرح شده که با یکدیگر سازگارند ) مدل کرد . یک گونه از این مدلسازی‌ها در شکل زیر آورده شده است :




۳-۲۳ : طبق شکل ، نوع موجودیت‌های E و M و G قوی هستند و نوع موجودیت‌های F و H و K ضعیف هستند . امّا نوع موجودیت‌های H و K شرکت کننده در نوع ارتباط R3 نسبت به هم قوی هستند . بعلاوه نوع موجودیت‌های K و F شرکت کننده در نوع ارتباط R2 نسبت به نوع موجودیت G قوی هستند . پس در مجموع شش نوع موجودیت قوی و سه نوع موجودیت ضعیف وجود دارد .

۳-۲۴ : از آنجایی که زیرنوع G هم صفات E و هم صفات F را به ارث می برد ، و همچنین خود می تواند صفاتی افزون بر صفات E و F داشته باشد ، داریم ( $A_E$ و $A_F$ و $A_G$ به ترتیب مجموعه صفات E و F و G هستند ) :

$$A_G \supseteq A_E \cup A_F \Rightarrow |A_G| \geq |A_E \cup A_F|$$

اما لزوما نمی توان گفت :

$$|A_G| \geq |A_E| + |A_F|$$

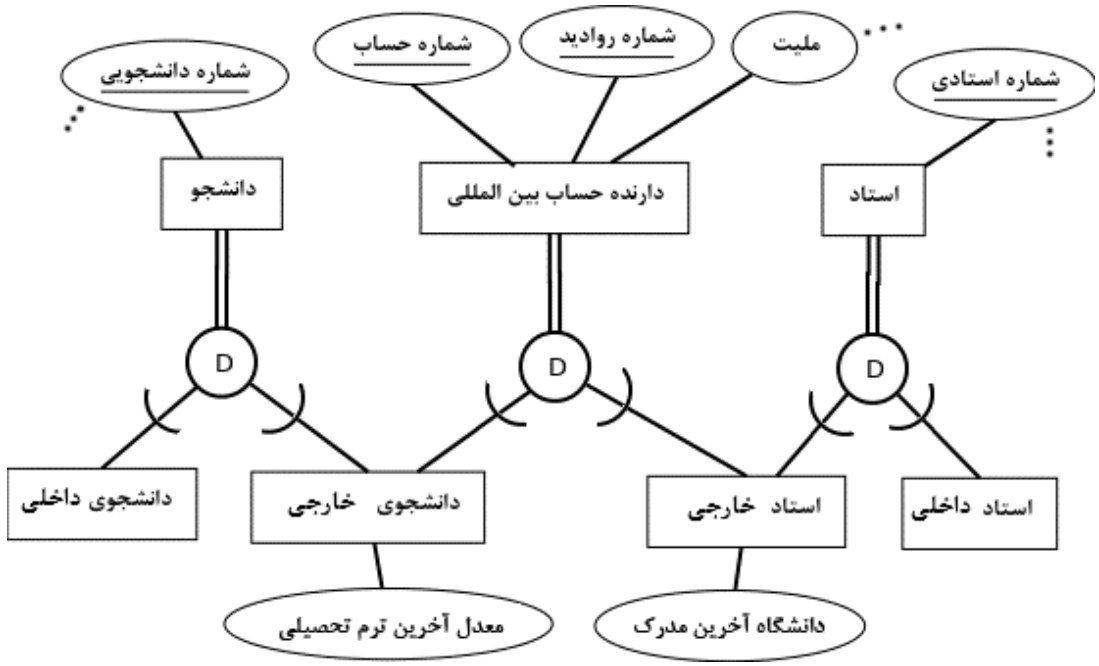
۳-۲۶ : تفاوت های زیر را برای این دو نوع می توان برشمرد :

زیرنوع با بیش از یک زبرنوع ، تمام صفات زبرنوع ها را به ارث می برد . در صورتی که یک نمونه از زیرنوع U-Type تنها صفات آن زبرنوعی را که از آن دسته است ، به ارث می برد .

در زیرنوع عادی ، یک نمونه از زیرنوع می تواند در تمام نوع ارتباط هایی که زبرنوع هایش شرکت دارند ، شرکت کند ، اما یک نمونه از زیرنوع دسته لزوما نمی تواند در نوع ارتباط های زبرنوع هایش شرکت داشته باشد . به عبارتی در زیرنوع U-Type ارتباط Is-A برقرار نیست و در نتیجه نوع ارتباط های زبرنوع به ارث برده نمی شود .

در صورتی که شناسه های زبرنوع ها از میدان های متفاوتی باشند ، زیرنوع با بیش از یک زبرنوع دارای چندین شناسه خواهد بود ، در صورتی که در این حالت برای زیرنوع U-Type یک شناسه جداگانه تعریف می شود و درنتیجه یک شناسه بیشتر ندارد .

۳-۲۸ : فرض کنید در یک دانشگاه بین المللی ، دو نوع دانشجوی داخلی و دانشجوی خارجی شناسایی شده اند . همچنین در این محیط ، اساتید نیز به دو دسته استاد خارجی و استاد داخلی تقسیم می شوند . حال فرض کنید در این دانشگاه ، تمام استادان و دانشجویان خارجی ، باید یک حساب بین المللی داشته باشند . در صورتی که برای دانشجو ، استاد ، و دارنده حساب بین المللی از تکنیک تخصیص استفاده کنیم ، نیاز به ارث بری چندگانه داریم . مدلسازی زیر برای این محیط ارائه شده است :



۳-۳۱ : در پاسخ به این سؤال بررسی می‌کنیم در چه حالتی می‌توان هر کدام از نوع ارتباط‌ها را حذف کرد ( بدیهی است که امکان حذف دو نوع ارتباط به صورت همزمان وجود ندارد و حداکثر یکی از نوع ارتباط‌ها می تواند افزونه باشد . زیرا در صورت حذف دو نوع ارتباط با هم ، یکی از نوع ارتباط‌ها ایزوله می شود و و مدلسازی خرد جهان واقع را به درستی صورت نمی‌گیرد )

الف ) نوع ارتباط « مادر است » :

این ارتباط نوع را در صورتی می‌توان حذف کرد که از طریق نوع ارتباط پدر بودن و نوع ارتباط ازدواج بتوان مادر یک فرزند را پیدا کرد . چون هر فرزند تنها یک پدر دارد ، در نتیجه تنها در صورتی می توانیم مادر فرزند را پیدا کنیم که بتوانیم همسر پدرش را ( در حین تولد آن فرزند ) پیدا کنیم . حال اگر تاریخ تولد فرزند در مدلسازی دیده شده باشد ، در صورتی که هر مرد در یک بازه زمانی تنها یک همسر داشته باشد ، می توانیم مادر آن فرزند را به صورت یکتا پیدا کنیم ، در غیر این صورت ، ممکن است نتوانیم مادر یک فرزند را در این مدلسازی به درستی تشخیص دهیم .

اما اگر تاریخ تولد فرزند را نداشته باشیم ، تنها درصورتی می توانیم مادر فرزند را به درستی تشخیص دهیم که یک مرد تنها یک همسر داشته باشد ( در تمام طول حیات ) .

ب ) نوع ارتباط « پدر است » :

نوع ارتباط «پدر است » نیز کاملا مشابه ارتباط « مادر است » است و نیاز به بررسی ندارد .

پ ) نوع ارتباط ازدواج :

در این حالت ، این نوع ارتباط را تنها در صورتی می توانیم حذف کنیم که : هر زوجی حداقل یک فرزند را داشته باشند . در غیر این صورت مدلسازی خرد جهان واقع را به درستی انجام نشده است . اما در صورتی که هر زوج حداقل یک فرزند را داشته باشند ، چون هر فرزند دقیقا یک مادر و دقیقا یک پدر دارد ، می توان نوع ارتباط ازدواج یک نمونه مرد و زن را به درستی تشخیص داد و نتیجه گرفت . همچنین اینکه یک مرد با یک زن ازدواج نکرده باشد ، از طریق بررسی تمام فرزندان آن مرد ( و یا تمام فرزندان آن زن ) قابل بررسی است .

# گفتار ۴

## کنجکاوی‌ها*

۴-۱ : جدول نمایش مفهوم ریاضی **رابطه** است . رابطه عنصر ساختاری اساسی در مدل رابطه‌ای است . در مدل رابطه‌ای هیچیک از مفاهیم فایلینگ منطقی و فیزیکی مطرح نیستند . اشاره‌گر از مفاهیم مدل رابطه‌ای نیست . ستون‌های $STID$ و $COID$ در جدول $STCOT$ امکان ارجاع مقداری هستند به سطر ( هایی ) از جدول‌های $STT$ و $COT$ .

۴-۲ : یکی از دلایل این جداسازی این است که تغییرات در برنامه‌های تعریف داده‌ها تأثیری روی برنامه‌های برزش داده‌ها ( عملیات در داده‌ها ) نداشته باشد . توجه داشته باشیم که پایگاه داده‌ها می‌تواند رشد کند و یا طراحی منطقی آن تغییر کند ( به گفتار پنجم مراجعه شود ) . در هر دو حالت ، برنامه‌های تعریف داده‌ها تغییر می‌کند . این تغییر نباید منجر به تغییر در برنامه‌های کاربردی موجود شود . برای توضیح بیشتر به گفتار سیزدهم ، قسمت ۳-۱-۱ مراجعه شود .

کنجکاوی : دلایل دیگر چیست ؟

۴-۳ : باید از تکنیک درج گروهی استفاده شود .

- مثال : فرض می‌کنیم جدول موقت $ITST$ را تعریف کرده باشیم . در این صورت چنین می‌نویسیم :

```
INSERT INTO ITST
        SELECT *
                FROM STT
                WHERE STMJR = 'IT';
```

۴-۴ : جدول $STCOT$ از طریق ستون $STID$ به جدول $STT$ و ستون $COID$ به جدول $COT$ ارجاع مقداری دارد . با حذف درس شماره‌ی $COM777$ ، باید که جامعیت داده‌ها حفظ شود . برای این منظور راه‌هایی وجود دارد که در گفتار دهم ، قسمت ۸ خواهیم دید ، در اینجا تنها به یک روش اشاره می‌کنیم : تمام سطرهای جدول $STCOT$ دارای همان شماره‌ی درس حذف می‌شوند .

کنجکاوی : راه‌های دیگر کدامند ؟

* در این قسمت تنها به کنجکاوی‌های مربوط به ساختار داده‌ی جدولی پاسخ می‌دهیم .

۴-۵ : پاسخ کنجکاوی ۴ دیده شود .

۴-۶ : سطرهای با شماره‌ی سطر $COM202$ هم باید بهنگام درآیند .

۴-۷ : استانده‌سازی در زمینه‌های فنی مزایا و البته معایبی هم دارد . برای پاسخگویی به این سؤال باید به متون ذیربط مراجعه کرد .

۴-۸ : به پاسخ تمرین ۱۶ از گفتار دهم مراجعه شود .

۴-۹ : نام ندارد ، مگر این که بنویسیم :

```
SELECT BKID , BKTITLE , BKPRICE * f AS BKPR
```

در این صورت نام ستون ، $BKPR$ است .

۴-۱۰ : می‌توان نوشت :

```
SELECT COUNT ( COID ) AS CN
     FROM COT.
```

**توجه** : در اینجا نیازی به $DISTINCT$ نیست .

۴-۱۱ : کنش « ثبت نام » با جدول $STCOT$ نمایش داده شده است ، پس باید از همین جدول استفاده شود .

۴-۱۲ : کلمه‌ی « منطقاً » یعنی سیستم فایل متناظر با جدول $STCOT$ را بطور فیزیکی سازماندهی مجدد نمی‌کند . البته این پاسخ جنبه‌ی تئوریک دارد . باید دید DBMS های موجود در اجرای این دستور چگونه رفتار می‌کنند .

۴-۱۳ : لازم نیست ، زیرا در صورت وجود حداقل دو دانشجوی همنام ، کاربر باید در جدول جواب ببیند که مثلاً هر دو دانشجوی با نام « دانشدوست » همرشته با دانشجوی با شماره‌ی داده شده هستند .

۴-۱۴ : خیر ، زیرا ارجاع مقداری از $STCOT$ به $STT$ و $COT$ است و نه برعکس .

۴-۱۵ : این ویژگی می‌گوید که ساختار داده‌ی جدولی قواعد جامعیت ذاتی ندارد ، برای پاسخ به این کنجکاوی باید به بحث جامعیت در مدل رابطه‌ای مراجعه کرد . در اینجا به کوتاهی می‌گوئیم که در مقایسه با مدل سلسله مراتبی یا شبکه‌ای که در آنها برخی قواعد جامعیت ذاتی وجود دارد ( مثلاً « نمونه فرزند بدون نمونه پدر نمی‌تواند باشد » ، یا «نمونه عضو بدون نمونه مالک نمی‌تواند باشد » ) ، در مدل رابطه‌ای ، حداقل از لحاظ تئوریک ، چنین محدودیت‌هایی ناشی از ماهّیت خود مدل ، وجود ندارد . ( به گفتار دهم ، قسمت ۸ مراجعه شود ) .

# تمرینات درون گفتار

۴-۱ : چندی نوع ارتباط «مدیریت» تغییر می‌کند. ممکن است $1:N$ بشود یا $M:N$ ، بستگی به قواعد معنایی محیط دارد.

۴-۲ : به گفتار چهاردهم مراجعه شود.

۴-۳ : به مباحث مهندسی فایل‌ها مراجعه شود. در ساده‌ترین حالت ، برای هرجدول یک فایل ترتیبی زمانی یا ترتیبی کلیدی می‌توان در نظر گرفت. اما بر پایه‌ی یکی از این دو ساختار می‌توان ساختار شاخص دار (متراکم یا نامتراکم) ایجاد کرد. بسته به امکانات DBMS ممکن است ساختار فایل درهم (با استفاده از تابع درهمسازی) را هم در نظر داشت.

۴-۴ تا ۴-۱۸ : مانند این تمرین‌ها در متن گفتار دوازدهم و در پایان آن گفتار وجود دارد. می‌توان به تمرین‌های آن گفتار مراجعه کرد. در اینجا به چهار تمرین پاسخ می‌دهیم.

۴-۷ :

```
SELECT STNAME
     FROM    STT, STCOT
          WHERE   COID = 'com222'
               AND (GRADE = (SELECT MAX(GRADE
                              FROM  STCOT) AS MAXG
                    OR
                    GRADE = (SELECT MIN(GRADE)
                              FROM STCOT) AS MING);
```

۴-۱۰ :

```
SELECT  COID, TR, YRYR, AVG(GRADE)
     FROM  STCOT
     GROUP  BY (COID , TR , YRYR);
```

: ۱۲-۴

```
SELECT  COUNT(*) , GRADE
     FROM  STCOT
          WHERE  COID = ‘com333’
          GROUP  BY GRADE;
```

: ۱۷-۴

```
DELETE  FROM  STCOT  WHERE TR = ’1’ AND YRYR = ’91-92’
     AND  STID IN (SELECT STID
                       FROM  STT
                       WHERE  STLEV = ‘bs’
                            AND
                                 STMJR = ‘comp’);
```

# تمرینات پایان گفتار[۳]

۴-۶ : دستورها از جمله می‌توانند چنین باشند :

- تعریف مجموعه
- کنترل عدم وجود عنصر تکراری در مجموعه
- درج عنصر به مجموعه
- حذف عنصر از مجموعه
- تغییر تعداد عناصر (کاردینالیتی) مجموعه
- تغییر نوع عناصر مجموعه
- تعریف محدودیت‌های جامعیتی ناظر به عناصر مجموعه از جمله محدودیت مقداری
- تعریف محدودیت‌های معنایی بین مجموعه‌ها از جمله محدودیت‌های عملیاتی
- عملیات ریاضی در مجموعه‌ها
- تعریف محدودیت‌های پردازشی عناصر مجموعه

[۳] بیشتر تمرین‌های این گفتار درباره مدل‌های داده‌ای سلسله مراتبی و شبکه‌ای است که عملاً از موضوع‌های تدریس درس پایگاه داده‌ها خارج شده‌اند . از این رو به این تمرین‌ها پاسخ نمی‌دهیم .

- تغییر محدودیت‌های جامعیتی
- تعریف زیر مجموعه روی یک یا چند مجموعه
- تعریف محدودیت‌های عملیاتی روی زیر مجموعه

۴-۱۱ : چون نمونه فرزند بدون نمونه پدر نمی‌تواند باشد و از طریق شناسه نمونه پدرش شناسایی می‌شود ، از این دیدگاه پاسخ مثبت است .

۴-۲۱ : به طور کلی سه راه است :

۱- طراحی سیستم‌های جدید و بی استفاده گذاشتن سیستم‌های بازمانده پس از حصول اطمینان از عملیاتی شدن سیستم‌های جدید به طور کامل ، و رضایت‌بخش .

۲- حفظ سیستم‌های بازمانده و طراحی و تولید یک سیستم مبدّل به نحوی که هرگاه لازم باشد ، با انجام تبدیلات لازم ، نیازهای کاربران را به صورت یک سیستم جدید پاسخ دهد .

۳- طراحی و ایجاد یک سیستم چند پایگاهی

شرح این راه حل‌ها از حیطه این درس خارج است .

۴-۱ : داده نیم‌ساختمند : متن HTML ، XML و . . .

داده ناساختمند : تصویر یک صفحه‌ی روزنامه که در آن متن ، عکس و . . . وجود دارد ، نقشه ، صدا و . . .

۱- ۹ : در این باره می‌توان از جمله [روحا ۷۸- الف] معرفی شده در فهرست منابع کتاب مراجعه کرد .

# گفتار ۵

## کنجکاوی‌ها

۵-۱ : به گفتار سیزدهم قسمت‌های ۳-۱ ، ۳-۲ و ۳-۳ مراجعه شود .

۵-۲ : برخی اطلاعات دیگر :

- اندازه‌ی صفحه
- تعداد صفحه‌های هر جدول
- میزان گسترش ممکن برای هر فایل
- چگونگی تناظر بین جدول‌ها و فایل‌ها

۵-۳ : اساساً کارایی یک سیستم پایگاهی مستقیماً و کاملاً بستگی دارد به امکانات سیستم در مدیریت فایلینگ پایگاه و وضع فایلینگ فیزیکی . هر چه لایه‌ی فایلینگ پایگاه و واحدهای مربوطه کاراتر عمل کنند ، کارایی سیستم پایگاهی بیشتر است . کارایی یک سیستم پایگاهی معمولاً با توان عملیاتی آن و زمان پاسخدهی به تراکنش‌ها ارزیابی می‌شود ، یعنی به تعداد تراکنش قابل اجرا در واحد زمان و مدت اجرای کامل تراکنش . در این ارزیابی ، میزان کارایی فایلینگ پایگاه نقش عمده دارد و بعلاوه عوامل دیگری هم دخالت دارند از جمله چگونگی بهینه‌سازی پرسش‌ها و پروتوکل‌های کنترل همروندی و . . . . اما تا آنجا که به ارزیابی میزان کارایی لایه‌ی فایلینگ مربوط می‌شود ، باید با استفاده از پارامترهای مطرح در مهندسی فایل‌ها و روش‌های تنظیم پایگاه‌داده بویژه تنظیم لایه فایلینگ ، انجام شود .

۵-۴ : سربار ( فزونکاری ) در سیستم پایگاهی چند منشأ دارد از جمله :

- نگاشت بین سطوح
- اعمال قواعد جامعیت
- اعمال ضوابط ایمنی ( در سطح DBMS )
- بهینه‌سازی پرسش

۵-۵ : به مباحث مربوطه در تئوری زبان‌های برنامه‌سازی مراجعه شود .

۵-۶ : به قسمت ۸-۵ از گفتار ششم مراجعه شود .

# تمرینات درون گفتار

۵-۱ :

```
CREATE  VIEW  U2V1 (STNUM, STDEPT)
     AS  SELECT  STT.STID, STT.STDEID
           FROM  STT;


CREATE  VIEW  U2V2 (STID, STPROG, COTITLE, TR, YRYR)
     AS  SELECT  STT.STID, STT.STMJR, COT.COTITLE,
           STCOT.TR, STCOT.YRYR
                 FROM  STT, COT, STCOT;


CREATE  VIEW  U3V1 (STID, STNAM, TR, YRYR, AVG)
     AS  SELECT  STT.STID, STT.STNAME, STCOT.TR,
                 STCOT.YRYR, AVG(GRADE) AS AVG
           FROM  STT, COT, STCOT
     GROUP BY  STID, STNAME, TR, YRYR;
```

# تمرینات پایان گفتار

۵-۶ : با استفاده از مدل داده‌ای که ترجیحاً باید مجموعه‌ای از مفاهیم ریاضی داشته باشد (به گفتار دهم مراجعه شود)

۵-۷ : دلیل چند سطحی بودن معماری پایگاه داده این است که کاربر بتواند در یک محیط فرا فایلی داده‌های خود را تعریف کند؛ در آن‌ها عملیات انجام دهد و روی آن‌ها تا حدی کنترل داشته باشد . به علاوه برای تأمین استقلال داده‌ای فیزیکی ، وجود حداقل سه سطح الزامی است .

۵-۸ : باید تمام ساخت‌هایی را که یک زبان برنامه سازی متعارف دارد ، برای DBL در نظر گرفته شود . (به تئوری زبان‌های برنامه سازی مراجعه شود) .

۵-۱۴ : باید موارد زیر را مشخص کنیم :

- فرمت رکود فایل

- ثابت یا متغیر بودن طول رکورد
- ساختار فایل
- استراتژی دستیابی به رکوردها
- فیلد کلید
- حداکثر اندازه فایل
- نوع هر یک از فیلدها
- اندازه بلاک
- میزان فضای آزاد در بلاک در برگزاری اولیه در فایل
- ثابت یا متغیر بودن طول فیلدها
- ارتباط منطقی فایل با فایل‌های دیگر
- مورد استفاده فایل : خواندن و یا نوشتن
- کاربردهای مجاز
- عملیات مجاز هر کاربر
- فشرده سازی بشود یا نه
- چگونگی ذخیره سازی فایل روی رسانه
- بسامد دستیابی به رکوردهای فایل
- نرخ عملیات در فایل : بازیابی ، درج ، حذف ، همگام سازی
- ضوابط ایمنی ( امنیت ) فایل
- لوکالیتی درون فایلی
- لوکالیتی بین فایل‌ها

# گفتار ۶

## کنجکاوی‌ها

۶-۱ : زیرا تعداد شیوه‌ی دستیابی و تنوع ساختار فایل‌ها نشان دهنده‌ی امکانات DBMS است در مدیریت فایلینگ منطقی پایگاه داده و نیز امکانات سیستم عامل در مدیریت فایلینگ فیزیکی پایگاه داده است . هر چه این امکانات بیشتر باشد ، سیستم کاراتر و با انعطافپذیری بیشتر می‌تواند عمل کند .

۶-۲ : برای دیدگاه‌های دیگر ، باید از پارامترهای شناخت DBMS ( DBMS-شناسی ) آگاه باشیم تا بتوانیم بر اساس برخی از آن پارامترها ، رده‌بندی‌های دیگری از این نرم‌افزار ارائه کنیم ( به قسمت ۸ از همین گفتار مراجعه شود ) .

۶-۳ : برای پاسخ به این کنجکاوی ، باید مستندات فنّی DBMS ها را بررسی کرد که معمولاً به دشواری در دسترس هستند . البته در سیستم‌های پایگاهی نامتمرکز ، طرح‌های دیگری هم متصور هستند .

۶-۴ : تا آنجا که به عملیات در یک سیستم پایگاهی مربوط می‌شود ، اسلوب‌های عملیاتی متعارف همان‌ها هستند که در متن گفتار آمده است .

## تمرینات درون گفتار

تمرین‌های درون این گفتار در اساس برای مطالعه بیشتر خواننده پیشنهاد شده اند.

## تمرینات پایان گفتار

تمرین‌های پایان این گفتار در اساس برای توصیه به خواننده به منظور مطالعه در متون تکنیکی پایگاه داده است . پاسخ برخی از تمرین‌ها در متن گفتارها وجود دارند هرچند نه به طور کامل .

# گفتار ۷

## کنجکاوی‌ها

۷-۱ : ممکن است در بعضی از محیط‌ها ، در شرایط خاص ، وظایف دیگری هم مطرح شوند .

۷-۲ : همه‌ی وظایف برشمرده به هر حال به نحوی مطرح‌اند ، البته میزان فعالیت لازم در انجام هر وظیفه می‌تواند خیلی کمتر باشد . از لحاظ کیفی هم برخی از وظایف ( مثلاً وظایف ۳۲ ، ۳۷ ، ۳۹ ، ۴۱ و . . . ) تخفیف پیدا کنند .

۷-۳ : زمینه‌های مهمتر :

- سیستم عامل
- شبکه
- سخت‌افزار
- مهندسی نرم‌افزار
- معماری سیستم‌های پایگاهی
- مدیریّت فعالیت‌های تیمی
- مهندسی ارتباطات
- رفتارشناسی کاربران در سازمان

۷-۴ : مدیر داده‌ها ، مدیر امور پژوهش-توسعه ، مسئول تنظیم مستندات

وظایف این اعضاء بطور مستقیم با DBMS مرتبط نیست ( هر چند گاه ممکن است بطور غیرمستقیم مرتبط باشد ).

## تمرینات درون گفتار

تمرین‌های درون این گفتار در اساس برای مطالعه بیشتر خواننده پیشنهاد شده اند.

## تمرینات پایان گفتار

تمرین‌های پایان این گفتار در اساس برای توصیه به خواننده به منظور مطالعه در متون تکنیکی پایگاه داده است . پاسخ برخی از تمرین‌ها در متن گفتارها وجود دارند هرچند نه به طور کامل .

# گفتار ۸

## کنجکاوی‌ها

۸-۱ : هر یک از مزایای برشمرده اهمیت خاص خود را دارد . تقسیم این مزایا به دو دسته‌ی « اصلی » و « فرعی » کار سرراستی نیست . شاید بتوان مزایای شماره‌ی ۱۹ ، ۲۰ ، ۲۲ ، ۲۷ ، ۲۸ و ۲۹ را ، نسبت به سایر مزایا « فرعی » دانست .

۸-۲ : برخی معایب مستقیم یا غیرمستقیم دیگر :

- لزوم بهینه‌سازی پرسش‌ها که گاه سربار در پی دارد .
- امکانات ناکافی DBMS های موجود و انحراف آنها از معیارها و ضوابط تئوریک مطرح در دانش و فن پایگاه داده‌ها
- وابسته‌شدن برنامه‌های کاربردی به جنبه‌های یک DBMS مشخص
- دشواری در اندرکنش بین دو یا بیش از دو DBMS . . .
- دشواری در تصمیم‌گیری درباره‌ی معماری سیستم پایگاهی

۸-۳ : از جمله شرایط دیگر :

- تنوع کاربردها
- ساختمندبودن بیشتر داده‌ها ( هر چند ایجاد سیستم پایگاهی ( به معنای دقیق آن ) برای داده‌های نیم‌ساختمند و ناساختمند هم امکان‌پذیر است ، اما دشواری‌هایی دارد ) .

## تمرینات درون گفتار

تمرین‌های درون این گفتار در اساس برای مطالعه بیشتر خواننده پیشنهاد شده اند.

## تمرینات پایان گفتار

تمرین‌های پایان این گفتار در اساس برای توصیه به خواننده به منظور مطالعه در متون تکنیکی پایگاه داده است . پاسخ برخی از تمرین‌ها در متن گفتارها وجود دارند هرچند نه به طور کامل .

# گفتار ۹

## کنجکاوی‌ها*

۹-۱ : برخی عوامل دیگر :

- ماهیت تراکنش‌ها از نظر نوع عملیات ، اندازه ( تراکنش کوتاه ، تراکنش بلند )
- مدل تراکنش‌ها ( مسطح ، تودرتو ، زنجیره‌ای ، . . . )
- توان عملیاتی و کارایی مورد انتظار از سیستم
- ماهیت سازمان ( از نظر متمرکز بودن یا نامتمرکز بودن )
- امکانات مالی سازمان
- نرخ دسترس‌پذیری داده‌ها
- الگوی مراجعه‌ی کاربران به سیستم

۹-۲ : پاسخ را به خواننده وامی‌گذاریم .

۹-۳ : پاسخ مثبت است .

۹-۴ : مزایای سیستم پایگاهی با معماری مشتری/ خدمتگزار بستگی به گونه‌ی آن دارد . اما می‌توان هماهنگی بهتر با نیاز داده‌ای-پردازشی کاربر ( مشتری ) را هم از مزایا این معماری دانست .

## تمرینات درون گفتار

تمرین‌های درون این گفتار در اساس برای مطالعه بیشتر خواننده پیشنهاد شده اند.

* در این قسمت تنها به کنجکاوی‌های مربوط به معماری مشتری/ خدمتگزار پاسخ می‌دهیم .

# تمرینات پایان گفتار

تمرین‌های پایان این گفتار در اساس برای توصیه به خواننده به منظور مطالعه در متون تکنیکی پایگاه داده است . پاسخ برخی از تمرین‌ها در متن گفتارها وجود دارند هرچند نه به طور کامل .

# گفتار ۱۰

## کنجکاوی‌ها

۱۰-۱ : خیر ، اگر بیش از یک صفت رابطه از یک میدان مقدار بگیرند ، تعداد میدان‌های رابطه کمتر از درجه‌ی آن می‌شود . بطور کلی داریم : $m \leq n$ که در آن m تعداد میدان‌ها و n درجه‌ی رابطه است .

۱۰-۲ : اگر مقادیر صفات COID و STID از اجزاء با معنا تشکیل شده‌باشند ، مانند صفت شماره‌ی دانشجو که معمولاً چنین است ، می‌توان گفت که از لحاظ معنایی تجزیه‌شدنی هستند ، هر چند در عمل به صورت یک واحد تجزیه‌نشدنی نمایش داده می‌شوند و پیاده‌سازی می‌شوند ( پاسخ کنجکاوی ۵ از گفتار سوم دیده‌شود ) .

۱۰-۳ : هر رابطه حداقل یک معنا دارد ، اما ممکن است بیش از یک معنا هم داشته باشد ، اگرچه پدیده‌ی اختلاط اطلاعات بروز می‌کند که می‌تواند مشکلاتی ایجاد کند ( به گفتار پانزدهم مراجعه شود ) .

۱۰-۴ : چون یک درس ممکن است بیش از یک پیشنیاز داشته‌باشد و یا اصلاً پیشنیاز نداشته باشد ، هم افزونگی پدید می‌آید و هم هیچمقدار ؛ بنابراین پاسخ منفی است ( زیرا طراحی بدی است ) .

۱۰-۵ : می‌توان آنرا یک صفت در همان رابطه‌ی نمایشگر نوع موجودیت درس در نظر گرفت . البته با پذیرش بروز میزانی هیچمقدار و چنانچه ملاحظات خاص دیگری ناظر به کارایی مطرح نباشد ( به گفتار چهاردهم مراجعه شود ) .

۱۰-۶ : روش‌های معرفی مقادیر میدان از لحاظ تئوریک :

۱. دادن مقادیر مستقیماً در دستور تعریف میدان

۲. دادن مقادیر بکمک یک الگوی مولّد مقادیر ، مثلاً تصاعد حسابی یا هندسی

۳. قرار دادن مقادیر در یک جدول ( احیاناً تک ستونی ) و دادن نام آن در دستور تعریف میدان

کنجکاوی : آیا روش‌های دیگری هم متصور است ؟

۱۰-۷ : از جمله ملاحظات پیاده‌سازی میدان این است که باید برای عملیات روی میدان‌ها قواعد مشخص وجود داشته باشد . مقایسه‌ی صفت CREDIT با صفت GRADE بی‌معناست و چنین است عمل تفریق این دو صفت از یکدیگر . اما عمل ضرب CREDIT در GRADE بامعناست و باید در تعریف قواعد عملیاتی منظور شود .

۱۰-۸ : معایب رابطه‌ی تمام کلید :

- طولانی بودن کلید
- عدم امکان درج تاپل ناقص در رابطه

برای مشکل طولانی بودن ، در عمل به تاپل‌ها ( سطرهای جدول ) شماره داده می‌شود تا نقش کلید را ایفا کند .

کنجکاوی : این کار چه مشکلی پدید می‌آورد ؟

۱۰-۹ : دلیل کاهش‌ناپذیر بودن کلید کاندید این است که اگر کاهش‌پذیر باشد ، دیگر لزومی ندارد آن ترکیب صفات کاهش‌پذیر کلید کاندید باشد ، زیرا حداقل یک صفت آن افزونه است و سبب طولانی شدن کلید کاندید می‌شود .

از نظر تمییز تاپل‌های رابطه از یکدیگر ، معرفی یک کلید کاندید کفایت می‌کند ، اما اگر امکان معرفی بیش از یک کلید کاندید در سیستم وجود داشته باشد ، سیستم می‌تواند با ایجاد شاخص روی کلیدهای کاندید ، استراتژی دستیابی را گسترش دهد و در پاسخگویی به برخی از پرسش‌ها ، از شاخص‌های از پیش ایجاد شده استفاده کند . البته ایجاد شاخص سربار خاص خود را دارد ( به مباحث مربوطه در مهندسی فایل‌ها مراجعه شود ) .

۱۰-۱۰ : کلید کاندید عامل تمییز تاپل‌های رابطه از یکدیگر است . تاپل نمایشگر یک نمونه از یک نوع موجودیت است در معنای عام . اگر مفهوم کلید کاندید وجود نداشته باشد ، نمی‌توان مشخص کرد کدام تاپل نمایشگر کدام نمونه موجودیت در خُرد جهان واقع است .

۱۰-۱۱ : در عمل یا باید از یکی از سه روش دیگر برحسب مورد ، استفاده کرد ، یا اساساً طراحی بدون چرخه‌ی ارجاع داشت .

کنجکاوی : چگونه ؟

۱۰-۱۲ : وقتی که یک مدل داده‌ای تنها یک عنصر ساختاری اساسی داشته باشد ، در این صورت هم نوع موجودیت و هم نوع ارتباط را الزاماً باید با همان عنصر نمایش داد . این کار می‌تواند هم سبب بروز افزونگی شود و هم در کنترل جامعیت پایگاه داده‌ها دشواری و سربار ایجاد کند . همیشه وجود میزانی تنوع در عناصر ساختاری مدل داده‌ای می‌تواند سودمند باشد به شرط آنکه نقش هر عنصر در طراحی کاملاً روشن باشد و نقش‌ها با یکدیگر تداخل نداشته باشند .

# تمرینات درون گفتار

۱۰-۱ :

الف : تاریخ اخذ آخرین مدرک استادان رشته 'کامپیوتر' را بدهید.

ب : مشخصات معلم‌های متولد ۱۲ اردیبهشت‌ماه از سال ۱۳۵۷ تا ۱۳۷۰ را بدهید.

: ۲-۱۰

نام تاریخ فوت تاریخ تولد کد ملّی

- $PERSON(NC\ ,\ BDATE\ ,\ DDATE\ ,\ PNAME\ ,\ldots)$

- $PERS(NC\ ,\ FNAME\ ,\ LNAME\ ,\ \ FNC\ ,\ \ MNC\ ,\ldots)$

کدملّی کدملّی نام خانوادگی نام کد ملی

مادر پدر

: ۴-۱۰

```
CREATE     DOMAIN     T     CHAR(1) …
CREATE     DOMAIN     YY    CHAR(5) …
CREATE     DOMAIN     TRYRYR
      TR   DOMAIN     T
      YRYR DOMAIN     YY          ;
```

پاسخ تمرین ۱۷ پایان این گفتار هم دیده شود.

۵-۱۰ : به پاسخ تمرین ۶ پایان این گفتار مراجعه شود.

: ۶-۱۰

```
CREATE  RELATION  NNSTCOT
     (STID  CHAR(8)  NOTNULL ,
          CTYG  RELATION
               (COID …
                TR  …
                YRYR …
                GRADE …)
     PRIMARY KEY  STID;
```

نتیجه : دستورهای DDL (و نیز DML و DCL) پیچیده می‌شوند.

: ۷-۱۰

- مثال

شماره تلفن کد ملی

$PERTEL(\underline{PNC\ ,PHONO})$

تاریخ افزایش حقوق شماره

$EMPLOYEE(\underline{EMID\ ,\ SALAUG\ ,\ DATE})$

قاعده معنایی : یک کارمند ممکن است در بیش از یک تاریخ ، به مبلغ یکسان افزایش حقوق بگیرد.

۸-۱۰ :

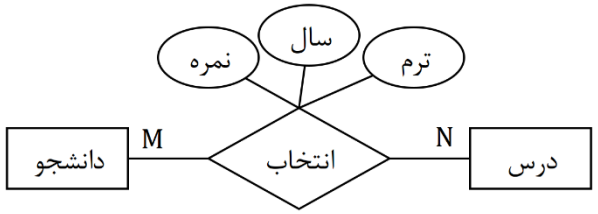
شماره دانشجویی کدملی

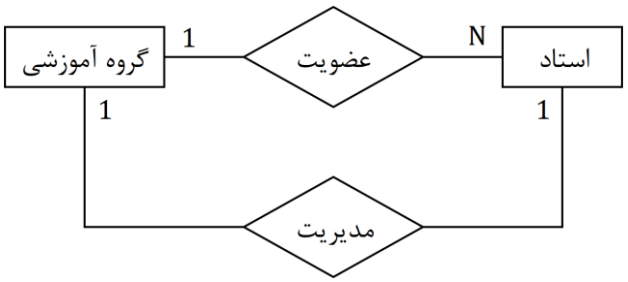
$$STUDENT(\underline{STID}\ ,STNAME\ ,\dots,\underline{STNC}\ ,\dots)$$

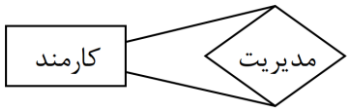
$$DOCTOR(\underline{DOID}\ ,DNAME\ ,\dots,\underline{DNC}\ ,\dots)$$

شماره نظام پزشکی کدملی

۹-۱۰ :




به پاسخ تمرین ۱۳ پایان این گفتار مراجعه شود.

۱۲-۱۰ : رابطه‌ی مثال ۲۲ دیده شود.

۱۳-۱۰ :

شماره کارمند شماره همکار شماره مدیر

$$EMP(\underline{EID\ ,CEID}\ ,\dots\dots,\underline{EMGRID}\ ,\dots)$$

- اگر درجه‌ی این رابطه $m$ باشد، تعداد میدان‌هایش $n = m - 2$ است.
- این رابطه چرخه‌ی ارجاع به خود دارد.
- دگرنامی دو صفت شماره‌ی همکار و شماره‌ی مدیر الزامی است.
- این رابطه در 1NF است (رجوع شود به گفتار ۱۵) .
- هر کارمند حداقل یک همکار باید داشته باشد.

کنجکاوی : اگر بخواهیم سابقه «تحت مدیریّت بودن» کارمند را داشته باشیم، چه تغییری در همین رابطه باید ایجاد کرد؟ چه پیش می‌آید؟

۱۰-۱۶ :

اولاً :

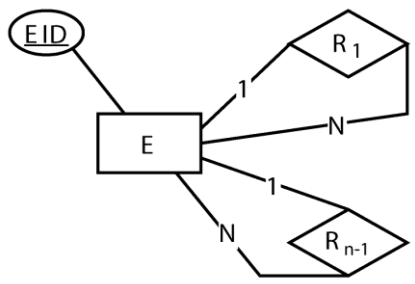
- استاد نمی‌تواند دو طرح پژوهشی به طور همزمان داشته باشد.
- استاد پژوهشگر نمی‌تواند بیش از دو درس در ترم تدریس کند.
- استاد پژوهشگر باید هرسال دو مقاله در مجله معتبر بین‌المللی به نشر برساند.

ثانیاً : چون «پژوهشگر» گونه‌ای است از «استاد» (ارتباط IS-A)، قاعده‌ی C2 باید به صورت زیر اِعمال شود :

- حذف تاپل از رابطه نمایشگر استاد، در صورتی منجر به حذف تاپل از رابطه نمایشگر پژوهشگر می‌شود که استاد یک پژوهشگر باشد.
- حذف تاپل از رابطه‌ی نمایشگر پژوهشگر منجر به حذف تاپل از رابطه‌ی نمایشگر استاد نمی‌شود.
- درج تاپل در رابطه‌ی پژوهشگر منوط است به درج همان تاپل، ابتدا در رابطه‌ی نمایشگر استاد یا بررسی وجود آن در این رابطه.
- بهنگام سازی صفات ناکلید در تاپلی از رابطه پژوهشگر منجر می‌شود به بهنگام سازی همان صفات در تاپل متناظر در رابطه‌ی استاد.
- بهنگام سازی صفات در تاپلی از رابطه نمایشگر استاد به شرطی منجر به بهنگام سازی آن صفات در تاپلی از رابطه‌ی پژوهشگر می‌شود که آن تاپل مربوط به یک نمونه پژوهشگر باشد.

# تمرینات پایان گفتار

۲-۱۰ : وقتی که نوع موجودیت E تنها صفت شناسه داشته باشد و یا خودش $n-1$ نوع ارتباط با چندی $1:N$ یا $1:1$ داشته باشد و نوع ارتباط ها صفت نداشته باشند و این محیط را تنها با یک رابطه نمایش دهیم ، رابطه‌ای از درجه‌ی $n$ خواهیم داشت که در آن تمام صفات از یک میدان مقدار می‌گیرند .




$RE\ (\underline{EID}, \underline{E1ID}, \underline{E2ID}, \dots, \underline{E_{n-1}ID})$

حالت خاص : $n = 0$ ، رابطه ای داریم که عنوان آن تهی است و تنها صفت آن روی میدان تهی تعریف می‌شود .

۱۰-۳ : از مزیت های دیگر میدان این است که امکانی است برای تشخیص ارجاعات بین رابطه ها ( و پاسخ قست دوم این سوال بدین ترتیب مثبت است ) . در واقع می توان با استفاده از مفهوم میدان ، مفهوم کلید خارجی را تعمیم داد : اگر $A \in H_{R1}$ و $B \in H_{R2}$ و A و B از یک میدان باشند ، می توان گفت نوعی ارتباط معنایی بین $R1$ و $R2$ وجود دارد به نحوی که ارجاع از طریق کلید خارجی حالت خاصی است از این ارتباط معنایی . بحث بیشتر در این باره خارج از حد این کتاب است .

۱۰-۴ : کد ملّی مادر ، کد ملّی پدر ، کد ملّی شخص

مثال ۱ : $PERSON\ (\ \underline{PNC}, \dots, PFNC, PMNC\ )$

مثال ۲ : $PERSON\ (\ \underline{PNC}, \dots\dots, \underbrace{\underline{BDATE}}_{\text{تاریخ تولید}}, \underbrace{\underline{DDATE}}_{\text{تاریخ فوت}}, \underbrace{\underline{FWDATE}}_{\text{تاریخ اولین ازدواج}}\ )$

۱۰-۵ : موارد زیر می تواند تغییر کنند :

- نوع
- نام
- طیف مقادیر

توجّه : در صورت تغییر نام ، شِمای پایگاه داده یعنی تعریف رابطه ها هم متناسباً باید تغییر کند .

۱۰-۶ : رابطه غیر نرمال زیر را در نظر می گیریم :

$\text{ERTELMAC}\ (\ \underbrace{\underline{\text{PNC}}}_{\text{کد ملّی}}, \underbrace{\underline{\text{PHONO}}}_{\text{شماره تلفن}}, \underbrace{\underline{\text{MACNO}}}_{\text{شماره ماشین}}\ )$

دستورات لازم پیچده می شوند :

درج تاپل : درج کن $< c_7 , p_7 , m_7 >$را .

با سینتاکس دلخواه :

```
APPEND      `p7` TO PHONO SET
            `m7` TO PHONE SET
            WHERE PNC = C7 ;
```

یا :

$INSERT\ INTO\ SUPREL\ PERTELMAC\ \ SUBTUPLE < c_7 >$
$\quad THEN$
$INSERT\ INTO\ SUBREL\ PERTELMAC\ .PHONO = `p_7`$
$\quad AND$
$INSERT\ INTO\ SUBREL\ PERTELMAC\ .MACNO = `m_7`;$

عمل حذف :

DELETE FROM PERTELMAC
$\quad$WHERE FEW = `C$_7$`
$\quad AND\ FROM\ PERTELMAC\ SUBRELS$
$\quad\quad WHERE\ PHONE\ = `M_7`$
$\quad\quad\quad AND$
$\quad\quad\quad MACNO\ = `M_7`;$

عمل بهنگام سازی :

$UPDATE\ PERTELMAC$
$\quad SET\ PNC = `C_8`$
$\quad WHERE\ PNC = `C_7`;$

اما برای بهنگام سازی صفت های دیگر :

$UDATE\ PERTEL\ MAC$
$\quad WHERE\ PNC = `C_7`$
$UPDATE\ SUBREL$
$\quad SET\ PHONO = `M_8`$
$\quad WHERE\ PHONE = `M_7`$
$UPDATE\ SUBREL$
$\quad SET\ MACNO = `M_8`$
$\quad WHERE\ MACNO = `M_7`$

ذکر مثال دوم را به خواننده وا می گذاریم .

۷-۱۰ :

اولاً : رابطهٔ $EMP$ غیر نرمال است زیرا یکی از صفات آن ، خود رابطه است رابطهٔ $PROJS$ .

توجّه : صرف اینکه صفت $PROJS$ در رابطهٔ $EMP$ مرکّب است دلیل غیر نرمال بودن این رابطه نیست . باید به ازاء حداقل یک مقدار مشخص از صفت $E\#$ ، مقداری برای صفت $PROJS$ در $EMP$ وجود داشته باشد که یک نمونه از بدنهٔ رابطهٔ $PROJS$ با کار دنیا لیتی حداقل دو ، باشد . هرگاه به ازاء هر مقدار از صفت $E\#$ ، یک بدنهٔ حداکثر تک تاپلی ( با کاردینا لیتی یک ) برای رابطهٔ $PROJS$ وجود داشته باشد ، رابطهٔ $EMP$ غیر نرمال نیست .

ثانیاً :

کلید کاندید رابطهٔ : $EMP : (E\#, PROJ\#)$

کلید کاندید رابطهٔ : $PROJS : (PROJ\#, HOURS)$

۱۰-۸ : رابطه مفهومی است ریاضی و ماهیتاً انتزاعی و عنصر ساختاری اساسی در مدل رابطهٔ است . فایل ترتیبی یک امکان ( و نه تنها امکان ) برای پیاده سازی مفهوم ریاضی رابطه در محیط فایلینگ منطقی و فیزیکی است .

توجّه : اساساً هیچیک از مفاهیم مربوط به فایلینگ در مدل رابطه‌ای مطرح نیستند .

۱۰-۹ : مثال :

$$EMP\ (\underbrace{ENC}_{\text{کد ملی}}, \underbrace{NAME}_{\text{نام}}, \underbrace{BDATE}_{\text{تاریخ تولد}}, \underbrace{LODATE}_{\text{مدرک اخذ آخرین مدرک}}, \underbrace{EEDATE}_{\text{خدمت شروع خدمت}}, \underbrace{EJOB}_{\text{سمت}})$$

خاصیت : درجهٔ رابطه شش ، اما تعداد میدان هایش چهار است .

۱۰-۱۰ : در اِعمال قاعدهٔ $C2$ با روش $CASCADE$ مشکل ایجاد می شود .

روش $RESTRICTED$ هم قابل اِعمال نیست ( نکتهٔ ۸ از همین گفتار را ببینید )

۱۰-۱۱ : مسند رابطه تفسیر یا معنای رابطه از نظر کاربر است و معمولاً به طور غیر صوری و به زبان طبیعی بیان می شود ، حال آنکه محدودیت رابطه ای یک عبارت بولی است بیان شده در یک زبان صوری و معمولاً شامل ارجاعاتی است به رابطه های پایگاه داده ، که باید همیشه به $TRUE$ ارزیابی شود . برای اطلاع بیشتر به آثار سی . جی . دیت به ویژه $[DATE\ 12\text{-}b]$ مراجعه شود .

۱۰-۱۲ : پاسخ این سوال را به عنوان کنجکاوی به خواننده وا می گذاریم . فهرست ارائه شده در متن برای مزایا و معایب مدل رابطه ای نسبتاً کامل است .

۱۰-۱۳ :

اولاً : الف ، ب و ج : مانند مثال های ۲۱ ، ۲۲ . ۲۳ متن گفتار می توان پاسخ داد .

د : چرخه ارجاع چهار رابطه‌ای :

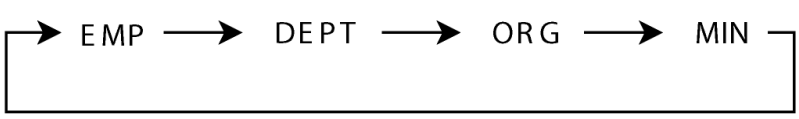
$$EMPL\ (\ \overbrace{\underline{E\#}}^{\text{شماره کارمند}}\ , \ldots , \overbrace{\underline{D\#}}^{\text{شماره اداره}}\ )$$

$$DEPT\ (\ \underline{D\#}, \ldots , \overbrace{\underline{ORG\#}}^{\text{شماره سازمان}}\ )$$

$$DRG\ (\ \underline{ORG\#}, \ldots , \overbrace{\underline{MZN\#}}^{\text{شماره وزارت}}\ )$$

شماره کارمندی وزیر

$MZN\ (\ \underline{MZN\#}, \ldots , \overbrace{\underline{\underline{\underline{\underline{\underline{\underline{\underline{MINZST\#}}}}}}}}\ )$




ثانیاً : حالاتی که در آن چرخه ارجاع پدید می‌آید . در پاسخ تمرین ۳۷ از گفتار چهاردهم بیان شده است .

ثالثاً : به خواننده واگذار می شود .

۱۰-۱۴ : بله ، وقتی که بیش از یک رابطه ، کلید کاندید یکسان داشته باشند .

مثال : فرض کنیم رابطه $STT$ به رابطه هایی به صورت زیر تجزیهٔ افقی شده باشد :

$STCOMP = \delta_{STMJR=`comp`}\ (\ STT\ )$

$STMAST = \ \delta_{STMJR=`IT`}\ (\ STT\ )$

و همین طور رشته های دیگر . . . . .

رابطه $STCOT$ از طریق $STID$ که کلید خارجی‌اش است ، به این رابطه‌ها ارجاع می دهد .

۱۰-۱۵ : چون بدنهٔ رابطه مجموعه است و مجموعه عنصر تکراری ندارد ، بنابراین تاپل‌های رابطه باید به نحوی از یکدیگر متمایز باشند . عامل تمایز همان مفهوم کلید اصلی است که متناظر شناسهٔ اصلی نوع موجودیت در معنای عام است در سطح طراحی منطقی در مدل رابطه ای . قاعده $C1$ هم ناظر به کلید اصلی است . پس می توان گفت این قاعده به تعبیری ذاتی مدل رابطه ای است .

۱۰-۱۶ : اگر منظور از رابطه در این سوال رابطهٔ مبنا باشد ، آنگاه :

- با حذف رابطهٔ مبنا از لحاظ تئوریک :

تعریف رابطه از کاتالوگ حذف می شود .

فایل متناظر با رابطه حذف می شود .

شاخص ( های ) ایجاد شده روی رابطه نامعتبر می شوند .

دیدهای تعریف شده روی رابطه نامعتبر می شوند .

ارجاعات به رابطهٔ نامعتبر می شوند .

فضای اختصاص داده شده به رابطه و ساختارهای کمکی آن آزاد می شوند . [ امّا در عمل در دستور DROP TABLE دو گزینه RESTRICT و CASCADE وجود دارد . ]

کنجکاوی : این گزینه های به چه کار می آیند ؟

- با حذف یک صفت از رابطه :

تعریف صفت از کاتالوگ حذف می شود .

دیدهای دارای آن صفت نامعتبر می شوند .

فایل متناظر با رابطه بازسازی می شود ( کنجکاوی : چگونه ؟ )

شاخص روی آن صفت نامعتبر می شود .

- با حذف یک محدودیت جامعیتی :

اگر بطور اعلانی به سیستم داده شده باشد ، تعریف آن از کاتالوگ خارج می شود .

اگر دارای رهانا در سیستم باشد ، آن رهانا نامعتبر می شود .

۱۰-۱۷ : ابتدا باید هر یک از میدان‌های میدان مرکّب را تعریف ، سپس میدان مرکّب را روی آن‌ها تعریف کنیم .

```
CREATE DOMAIN NAM ...
CREATE DOMAIN CIT ...
⋮
CREATE DOMAIN ZIPC ...
CREATE DOMAIN ADDRESS
      NAME DOMAIN NAM
      CITY DOMAIN CIT
      ⋮
      ZIPCODE DOMAIN ZIPC
CREATE DOMAIN CN CHAR ( 6 ) . . .
```

۱۰-۱۸ :

```
CREATE RELATION COPRECO
      COID DOMAIN CN
      PRECOID DOMAIN CN
      PRIMARY KEY ( COID, PRECOID )
      FOREIGN KEY COID REFERENCES COT
            ON DELETE CASCADE
            ON UPDATE CASCADE
      FOREIGN KEY PRECOID REFERENCES COT
            ON DELETE CASCADE
            ON UPDATE CASCADE
```

۱۰-۱۹ : تعریف رابطه‌ها به طرز معمول نوشته می‌شود . اما چون در این رابطه‌ها چرخهٔ ارجاع وجود دارد ، در نوشتن قاعده $C2$ باید به مشکلاتی که چرخه ارجاع ایجاد می کند ، توجه داشت . ( پاسخ تمرین ۱۰ را ببینید ) .

۱۰-۲۰ : در این تکنیک محدودیت جامعیتی داده شده توسط کاربر در درخواست کاربر اِعمال می‌شود و سپس درخواست اجرا می شود .

- مزیّت : محدودیت یکبار اعلان و در کاتولوگ سیستم نگهداری می شود .
- عیب : اگر در خواست کاربر با بسامد بالا اجرا شود ، تغییر درخواست باید تکرار شود و این کار سربار دارد . البته اگر اِعمال محدودیت فقط یکبار انجام شود ، طبعاً سربار تکرار تغییر وجود ندارد ، اما اگر کاربر بخواهد خود محدودیت را تغییر دهد و یا محدودیت های دیگری را هم اِعمال کند ، برنامه کاربر ( شامل درخواست کاربر ) باید متناسباً تغییر کند .

۱۰- ۲۱ :

۱۰-۲۱-۱ : خیر ، زیرا رابطه تاپل تکراری ندارد . وجود هیچمقدار در صفت کلید مانع تمییز تاپل‌ها می‌شود .

۱۰-۲۱-۲ : خیر ، گزاره‌ی درست باید مربوط به یک نمونه‌ی مشخص از یک نوع موجودیت باشد .

۱۰-۲۱-۳ : خیر ، به دلیل گفته شده در قسمت ۲۱-۱ .

۱۰-۲۱-۴ : خیر

۱۰-۲۱- ۵ : درج تاپل آخر ناممکن است .

۱۰-۲۲ : فرض می کنیم که $r$ بدنهٔ رابطهٔ R ، و $t_1$ و $t_2$ دو تاپل از $r$ باشند :

$t_1 \in \mathrm{r}, t_2 \in \mathrm{r}$

- تعریف سوپر کلید : صفت $S \subseteq H_R$ سوپر کلید R است هر گاه :

$\forall (t_1 \in \mathrm{r}, t_2 \in \mathrm{r})\ t_1[S] \neq t_2[S]$

- تعریف کلید کاندید : صفت $C$ کلید کاندید رابطه R است هرگاه :

$1)\ \forall (t_1 \in \mathrm{r}, t_2 \in \mathrm{r})\ t_1[C] \neq t_2[C]$

$2)\ C \nrightarrow X \mid X \subset C$

- تعریف کلید اصلی و کلید بدیل همانست که تعریف کلید کاندید .

- تعریف کلید خارجی :

فرض :

۱- $R_2 . F$ کلید خارجی در $R2$

۲- CS مجموعهٔ کلیدهای کاندید رابطه $R1$

۳- می توان $F$ را چنین تعریف کرد :

$1) F \in CS$

$2) \{F_{VALUS}\} \subseteq \{\square_{\square\square\square\square\square}\} \mid \square \in \square\square \; and \; C \neq F$

۱۰-۲۳ :

الف : از طریق طراحی مناسب رابطه ها

ب : با اِعلان NOT NULL برای کلید خارجی نمایشگر نوع ارتباط .

۱۰- ۲۴ : به عنوان تمرین اختیاری در نظر گرفته شود .

مسئله ماهیتاً یک مسئلهی ریاضی است که صورت آن چنین است :

فرض می‌کنیم B مجموعه ای از X تایی‌ها به صورت $\langle d_{1_i}, d_{2_i}, \dots, d_{n_i} \rangle$ باشد ، $i = ۱۰۰۰$ . به نحوی که $d_{1_i}$ مقادیر $A_1$ ، $d_{2_i}$ مقادیر $A_2$ ، . . . ، $d_{n_i}$ مقادیر $A_n$ ، باشند . پیدا کنید زیر مجموعه ای از $k$ تایی‌ها ، $k \leq n - ۱$ از مجموعهی $\langle A_1, A_2, \dots, A_n \rangle$ را به نحوی که مقدار هیچ دو $k$ تایی در آن یکسان نباشد .

۱۰- ۲۵ : با توجه به نظر دیت ، رابطه درجه صفر می تواند چنین نقشی داشته باشد . ( به آثار سی . جی . دیت رجوع شود ) . ضمناً این تمرین اختیاری منظور شود .

۱۰- ۲۶ : ‘‘تنها یک مدل رابطهای وجود دارد’’ . این بیان از دیت است . یعنی مدل رابطهای مجموعهای از مفاهیم ریاضی دارد که هریک تعریف واحد و دقیق دارند ، بنابراین خود مدل هم واحد است . توجه داشته باشیم که پیاده سازی یک $DBMS$ بر پایهی مدل رابطهای با خود مدل تفاوت دارد . پیاده سازی ها می‌توانند متعدد باشند . باید دید به چه میزان به مفاهیم تئوریک مدل رابطهای واحد وفادار هستند .

۱۰- ۲۷ : هیچیک از مفاهیم فایلینگ منطقی و فیزیکی ، از جمله اشاره گر ، در مدل رابطه ای مطرح نیستند .

۱۰- ۲۹ : کلید ناهوشمند کلیدی است که اجزاء تشکیل دهنده‌ی آن ، در یک حیطه‌ی معنایی مشخص ، بی معنا هستند .

کنکجاوی : این نوع کلید چه مزیّت و چه عیبی دارد ؟

۱۰-۳۰ : بهتر است کلید حتی الامکان از نوع عدد صحیح باشد .

کنجکاوی : چرا ؟

۱۰- ۳۱ : به خواننده توصیه می شود شماره دهی به سطرهای جدول را روی یک سیستم انجام دهد و بررسی کند آیا مشکلی بروز می کند یا نه ؟ ( به ویژه به عملیات درج و حذف دقت شود ) .

# گفتار ۱۱

## کنجکاوی‌ها

۱۱-۱ : گروه‌بندی انجام نمی‌شود .

۱۱-۲ : نتیجه اینکه رابطه‌ی جواب ، از دیدگاه تئوریک ، رابطه به معنای دقیق نیست ، زیرا رابطه حداقل یک کلید کاندید دارد .

۱۱-۳ : رابطه‌ی عملوند سمت چپ در عمل JOIN بزرگ می‌شود و در نتیجه عملیات زمانگیرتر می‌شود .

## تمرینات درون گفتار

۱۱-۱ :

$$\sigma_{<STID=\text{'}7712011\text{'}>}(STCOT) \cap \sigma_{<TR=\text{'}1\text{'}>}(STCOT) \cap \sigma_{<YRYR=\text{'}77-78\text{'}>}(STCOT)$$

۱۱-۲ :

- مثال :

$$R_3 = \sigma_{<STLEV='ms' \wedge STMJR=\text{'}phys\text{'}>}(STT)$$

$$R_4 = \sigma_{<STMJ=\text{'}phys\text{'}>}(STT) \cup \sigma_{<STLEV='ms' \wedge STMJR=\text{'}phys\text{'}>}(STT)$$

اگر: $R_4\ MINUS\ R_3 \neq \emptyset$

یعنی وجود دارد حداقل یک دانشجوی رشته‌ی فیزیک که مقطع تحصیلی‌اش «کارشناسی ارشد» نیست.

- مثال :

$$R_1 = \pi_{<STID>}\sigma_{<COID='PRO222' \wedge TR='2' \wedge YRYR='92',GRADE>}(STCOT)$$

$$R_2 = \pi_{<GRADE>}(R_1)$$

اگر : $card[R_1] = card[R_2]$

یعنی : هیچ دو دانشجو در درس مزبور در ترم- سال مزبور نمره یکسان نگرفته‌اند!

- ذکر مثال دیگر به خواننده واگذار می‌شود.

۱۱-۵ :

- عملگر برون پیوند چپ : رابطه S ار درجه m و $T = \{(null , \dots \dots , null)\}$ از درجه m را در نظر می‌گیریم :

- $R \ltimes\!\!\!\!\bowtie S = (R \bowtie S) \cup \left(R - \pi_{<H_R>}(R \bowtie S)\right) \times T$

- عملگر برون پیوند کامل :

رابطه‌های R و S به ترتیب از درجه‌ی m و n و $T_1 = \{(null , \dots \dots , null)\}$ و $T_2 = \{(null , \dots \dots , null)\}$ و به ترتیب با کاردینالیتی n و m مفروض‌اند

- $R \ltimes\!\!\!\!\bowtie\!\!\!\!\rtimes S = \left((R \bowtie S) \cup \left(R - \pi_{<H_R>}(R \bowtie S)\right) \times T_2\right) \cup \left((R \bowtie S) \cup \left(S - \pi_{<H_S>}(R \bowtie S)\right) \times T_1\right)$

- شبیه‌سازی عملگر برون پیوند راست به خواننده واگذار می‌شود.

- شبیه‌سازی عملگر نیم‌پیوند :

- $R \ltimes S = \pi_{<H_{R1}>}(R \bowtie S)$

۱۱-۶ :

PROF(PRID, PRNAME, PRRANK)

RESEARCHER(PRID, RESAREA, CURRESTIT)

R = PROF  OUTER-UNION  RESEARCHER

| R (PRID, | PRNAME, | PRANK, | RESAREA, | CURRESTIT) |
|---|---|---|---|---|
| 200 | pr2 | r2 | ? | ? |
| 100 | pr2 | r2 | ? | ? |
| 300 | ? | ? | s3 | c3 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

۱۱-۸ : کلید کاندید $R_3 = R_1 \bowtie R_2$ ؟

- اگر صفت پیوند کلید کاندید در رابطه‌ی $R_1$ باشد ، هر کلید کاندید رابطه‌ی $R_2$، کلید کاندید $R_3$ می‌شود.
- اگر صفت پیوند کلید کاندید در رابطه‌ی $R_2$ باشد ، هر کلید کاندید رابطه‌ی $R_1$ کلید کاندید $R_3$ می‌شود.
- اگر صفت پیوند در هردو رابطه کلید کاندید باشد ، همان صفت کلید کاندید $R_3$ می‌شود.
- در حالت کلی ، تنها می‌توان گفت که رابطه تمام کلید است. البته توجه داریم که $R_3$ (حاصل پیوند طبیعی $R_1$ و $R_2$) از دیدگاه تئوریک ، یک رابطه به معنای ریاضی نیست.

۱۱-۱۰ : COPRECO(COID, PRECOID)

- اگر بخواهیم شماره‌ی تمام درس‌های پیش‌نیاز بلافصل درس 'MAT777' را بیابیم، پرسش چنین است :

$\pi_{<PRECOID>}\sigma_{COID='MAT777'}(COPRECO)$

- اما اگر بخواهیم شماره‌ی تمام درس‌های پیش‌نیاز درس 'MAT777' در درخت پیش‌نیازی را بیابیم، باید پرسش بازگشتی بنویسیم :

برای آسانی در نوشتن نام‌ها فرض می‌کنیم نام رابطه‌ی COPRECO، CP و نام دو صفت آن به ترتیب C و P باشند.

فرض می‌کنیم بدنه‌ی رابطه‌ی CP در یک لحظه چنین باشد :

CP

| C | P |
|---|---|
| C1 | C2 |
| C1 | C3 |
| C2 | C4 |
| C3 | C4 |
| C4 | C5 |

اگر جدول جواب را CAP بنامیم، بدنه‌ی آن چنین خواهد بود :

| CAP | C | P |
|---|---|---|
| | C1 | C2 |
| | C1 | C3 |
| | C2 | C4 |
| | C3 | C4 |
| | C4 | C5 |
| | C1 | C4 |
| | C2 | C5 |
| | C3 | C5 |
| | C1 | C5 |

پرسش بازگشتی چنین است[۴] :

```
CAP := CP
DO until CAP reaches a “fixpoint” :
     WITH (R1 := CP RENAME {P AS PZ},
           R2 := CAP RENAME {C AS CZ},
          R3 := (R1 JOIN R2){C , P} ) :
     CAP := CAP UNION R3;
END;
```

منظور از fixpoint در اینجا یعنی تا لحظه‌ای که تاپل دیگری در CAP درج نشود.

۱۱-۱۱ : مانند تمرین ۱۱-۱۰ نوشته می‌شود.

۱۱-۱۲ :

```
CO.COTITLE  WHERE  NOTEXISTS  ST(
           NOT EXISTS  STCO (STCO.STID = ST.STID
                       AND STCO.COID = CO.COID
                       AND YRYR = ’78-79’
                       AND TR = ‘2’
                       AND GRADE  >= 10) );
```

۱۱-۱۳ :

[۴] منطق پرسش برگرفته از : DATE. C. J. , SQL and Relational Theory , D , REILLY , 2012;

$$\pi_{<SN1\ ,SN2>}\left(\sigma_{(SN1<SN2)}\left(\pi_{<SN1\ ,STNAME>}(STT \quad RENAME\ STID\ AS\ SN1)\right.\right.$$
$$\left.\left.\bowtie \pi_{<SN2\ ,STNAME>}(STT \quad RENAME\ STID\ AS\ SN2)\right)\right)$$

# تمرینات پایان گفتار

۱-۱۱ : تعداد پرتوهای ناقص : $m!$

۲-۱۲ : در جبر رابطه‌ای دو عملکرد ضرب و تقسیم لزوماً عکس یکدیگر نیستند . ( توصیه می شود با مثال بررسی شود ) .

۳-۱۱ : به مثال های متن درس مراجعه شود . مانند پرسش های خواسته شده ، کم و بیش ، در متن گفتار وجود دارد .

۶-۱۱ : این عبارات جبری معادل اند با : $R_1 \div R_2$ .

۸-۱۱ : از لحاظ تئوریک متصور است و حاصل ، رابطه‌ای است از درجه ی صفر ( دارای عنوان تهی ) .

۹-۱۱ : وقتی که $H_{R_1} = H_{R_2}$ باشد .

کنجکاوی : چرا ؟

۱۰-۱۱ :

- اثبات : $R_1 \bowtie \left(R_2 \cap R_3\right) = (R_1 \bowtie R_2) \cap (R_1 \bowtie R_3)$

چون عملکرد پیوند حالت خاصی است از عملکرد ضرب کارتزین ، بنابراین با استفاده از خواص ضرب کارترین به آسانی ثابت می شود .

- اثبات : $R_1 \bowtie \left(R_2 \cup R_3\right) = (R_1 \bowtie R_2) \cup (R_1 \bowtie R_3)$

همان توضیح مورد قبل در اینجا هم صادق است .

- اثبات : $\left(R_1 \bowtie R_2\right) DIVIDE\ R_2 = R_2$

اگر $H_{R_1} \cap H_{R_2} = \phi$

داریم : $H_{(R_1 \bowtie R_2)} = H_{R_1} \cup H_{R_2}$

بر می‌نهیم : $\left(R_1 \bowtie R_2\right) = R_3$

داریم : $H\left(R_3\ DEVIDE\ R_2\right) = H_{R_3} - H_{R_2} = \left(H_{R_1} \cup H_{R_2}\right) - H_{R_2}$

و چون : $H_{R_1} \cap H_{R_2} = \phi$

پس : $\left(H_{R_1} \cup H_{R_2}\right) - H_{R_2} = H_{R_1}$

در نتیجه : $\left(R_1 \bowtie R_2\right) DIVIDE\ R_2 = R_2$

۱۲-۱۱ : همیان مجموعهٔ ریاضی نیست زیرا عضو تکراری دارد . با استفاده از خواص مجموعه‌ی ریاضی ، می توان نشان داد که کدام مورد بر قرار نیست .

۱۳-۱۱ :

۱- برای $\Pi_{\langle L\rangle}\left(R\right)$ داریم : $\Pi_{\langle L\rangle}\left(\Pi_{\langle L\rangle}\left(R\right)\right) = \Pi_{\langle L\rangle}\left(R\right)$

۲- برای $\sigma_{cond}\left(R\right)$ داریم : $\sigma_{cond}\left(\sigma_{cond}\left(R\right)\right) = \sigma_{cond}\left(R\right)$

۳- $\gamma$عملکرد گروه بندی است و داریم : $\gamma_{\langle L\rangle}\left(\gamma_{\langle L\rangle}\left(R\right)\right) = \gamma_{\langle L\rangle}\left(R\right)$

۱۴-۱۱ : نه با جبر رابطه‌ای و نه با حساب رابطه‌ای .

کنجکاوی : چرا ؟

۱۵-۱۱ :

$$\Pi_{\langle STID,STNAME,TR,YRYR\rangle}\left(STTX\ \left(\gamma_{\langle STID,TR,YRYR\rangle} STCOT\ ADD\ AVG(GRADE)\right) AS\ R_1\right)$$

# گفتار ۱۲

## کنجکاوی‌ها

۱۲-۱: با نوشتن RESTRICT : در صورت وجود ارجاع به میدان ( در تعریف ستون ، یا در تعریف دید یا در تعریف محدودیت جامعیتی ) ، دستور DROP اجرا نمی‌شود . با نوشتن CASCADE ، دستور DROP اجرا می‌شود . با اجرای این دستور ، ارجاعات در تعریف دید یا در محدودیت جامعیتی بطور ضمنی نامعتبر می‌شوند . اما ستون‌های رجوع‌کننده به میدان حذف‌شده ، حذف نمی‌شوند ، بلکه به صورت زیر تغییر می‌کنند :

- به جای اینکه روی میدان حذف‌شده تعریف شده باشند ، مستقیماً روی همان نوع داده میدان تعریف می‌شوند .
- اگر از قبل مقدار پیش‌نهاده صریح نداشته باشند ، همان مقدار پیش‌نهاده صریح میدان ، اینک دیگر حذف‌شده را ، خواهند داشت .
- محدودیت‌های جامعیتی تعریف‌شده در تعریف میدان ، اینک دیگر حذف‌شده ، را همچنان به ارث می‌برند .

۱۲-۲:

- می‌توان گزینه‌ای در نظر گرفت تا در شرایط خاصی که باید تعریف شوند ، این دستور اجرا نشود .
- دگرنامی ستون هم می‌تواند در نظر گرفته شود .

۱۲-۳:

- اگر RESTRICT نوشته شود : چنانچه به جدول مبنا در تعریف دید یا در تعریف محدودیت جامعیتی ارجاع داده شده باشد ، دستور DROP اجرا نمی‌شود .
- اگر CASCADE نوشته شود ، دستور DROP اجرا می‌شود .

۱۲-۴:

- نامعتبر شدن هر پرسشی که در آن به جدول حذف‌شده ارجاع داده شده باشد .

- نامعتبر شدن تعریف دید ذخیره شده و رابطه لحظه‌ای و هر ساختار دیگر که در تعریف آن‌ها به نحوی به جدول حذف‌شده ارجاع داده شده باشد .

۱۲-۵: اگر از گزینه UNIQUE استفاده نشود ، MATCH معادل است با IN .

۱۲-۶: این عملگر همان کاربردی را دارد که عملگر OUTER UNION . در واقع اگر $T_1$ و $T_2$ دو جدول با ستون‌هایی نایکسان باشند ( در تئوری : دو رابطه که نوع سازگار نیستند ) ، این عملگر امکان می‌دهد تا « اجتماع » آن دو جدول محاسبه شود .

۱۲-۷: با این دو گزینه شرایط سلب ( بازپس‌گیری ) امتیاز داده‌شده تعیین می‌شود . به مثال زیر توجه کنیم :

$U_1$ : GRANT SELECT ON TABLE STT TO $U_2$ WITH GRANT OPTION ;

$U_2$ : GRANT SELECT ON TABLE STT TO $U_3$ WITH GRANT OPTION ;

$U_3$ : GRANT SELECT ON TABLE STT TO $U_4$ WITH GRANT OPTION ;

فرض می‌کنیم $U_1$ امتیاز داده‌شده را از $U_2$ بازپس بگیرد :

$U_1$ : REVOKE SELECT ON TABLE STT TO $U_2$ …

گذشته از جزئیات ، در اینجا اگر گزینه RESTRICT نوشته‌شود ، دستور REVOKE اجرا نمی‌شود . اما اگر CASCADE نوشته شود ، امتیاز داده‌شده به $U_2$ و توسط $U_2$ به $U_3$ و توسط $U_3$ به $U_4$ بازپس گرفته می‌شود .

# تمرینات پایان گفتار

۱۲-۲: در این سؤال تنها پاسخ برخی موارد مهم‌تر آورده و پاسخ دیگر موارد بر عهده‌ی خواننده گذاشته شده است.

مثال ۱۸:

```
SELECT STID, COID, TR, YRYR, GRADE AS GR
FROM STCOT
```

مثال ۲۹:

```
SELECT STNAME
FROM STT, STCOT, COT
WHERE STT.STID=STCOT.STID
      AND STCOT.COID=COT.COID
      AND COTYPE='p'
      AND TR='1' AND YRYR='81-82'
```

مثال ۳۰:

```
SELECT STNAME
FROM STT
WHERE NOT EXISTS
     (SELECT COID FROM STCOT WHERE STID=’76010222’ AND TR=’1’ AND
YRYR=’82-83’
     EXCEPT
     SELECT COID FROM STCOT WHERE STCOT.STID=STT.STID AND TR=’1’ AND
YRYR=’82-83’)
```

مثال ۳۵:

```
SELECT COTITLE
FROM COT
WHERE NOT EXISTS
(SELECT * FROM STT WHERE STMJR=’Math’
     AND EXISTS (SELECT *
 FROM STCOT
                      WHERE STCOT.STID=STT.STID
                           AND STCOT.COID=COT.COID
                           AND TR=’1 AND YRYR=’78-79’
AND GRADE<10)
                           OR
                           NOT EXISTS
(SELECT * FROM STCOT
                                WHERE STCOT.STID=STT.STID
                                     AND STCOT.COID=COT.COID
AND TR=’1 AND YRYR=’78-79’)
```

مثال ۳۸:

```
SELECT PRID, PRNAME, RANK
FROM PROF
WHERE PRID NOT IN (SELECT DPMGRID FROM DEPT)
```

مثال ۵۱:

```
SELECT COTITLE
FROM COT
WHERE NOT EXISTS
     (SELECT * FROM STT
          WHERE EXISTS
               (SELECT * FROM STCOT
                WHERE STCOT.STID=STT.STID
                AND STCOT.COID=COT.COID
                AND TR=’2’ AND YRYR=’78-79’
 AND GRADE<10)
               OR NOT EXISTS
                    (SELECT * FROM STCOT
                     WHERE STCOT.STID=STT.STID
                          AND STCOT.COID=COT.COID
                          AND TR=’2’ AND YRYR=’78-79’)
```

مثال ۵۸:

```
SELECT COTITLE
FROM COT
WHERE NOT EXISTS
      (SELECT STID FROM STCOT WHERE STCOT.COID=’LIT111’
      EXCEPT
      SELECT STID FROM STCOT WHERE STCOT.COID=COT.COID)
```

مثال ۵۹:

```
SELECT STNAME
FROM STT, STCOT, COT
WHERE STT.STID=STCOT.STID AND STCOT.COID=COT.COID AND CREDIT=’2’
```

۱۲-۳: در این سؤال تنها پاسخ برخی موارد مهم‌تر آورده و پاسخ دیگر موارد بر عهده‌ی خواننده گذاشته شده است.

Q2:

```
SELECT SNAME
FROM STT
WHERE STID NOT IN
(SELECT STID
 FROM STCOT
 WHERE TR=1 AND YRYR=’79-80’ AND GRADE<=15)
           AND STID IN
(SELECT STID
 FROM STCOT
 WHERE TR=1 AND YRYR=’79-80’ AND GRADE>15);
```

توجه: اگر دانشجویی در این ترم و سال هیچ درسی نگرفته باشد، در جواب این پرسش SQL نمی‌آید، مگر آنکه شرط دوم کلاز WHERE برداشته شود.

کنجکاوی: اگر نمره‌ی یک دانشجو در این ترم و سال هیچمقدار باشد، آیا این پرسش درست کار می‌کند؟

Q3:

روش اول: پرسش بهم‌بسته

```
SELECT COTITLE
FROM COT
WHERE NOT EXISTS
(SELECT *
 FROM STT
```

```
      WHERE STDEID='d333'
AND STLEV='bs'
          AND NOT EXISTS
(SELECT *
 FROM STCOT
 WHERE STCOT.COID=COT.COID
AND STCOT.STID=STT.STID));
```

Q4: فرض کنیم جدول PUBT (PUBID, PUBTITLE, PRID, …) نمایشگر نوع موجودیت اثر منتشره به همراه ارتباط انتشار بین استاد و اثر منتشره باشد (با این فرض که هر اثر منتشره تنها توسط یک استاد منتشر می‌شود.) در این جدول PUBID شناسه اثر، PUBTITLE عنوان اثر و PRID شناسه استاد صاحب اثر است.

```
SELECT *
FROM PROFT
WHERE PRID IN
(SELECT PRID
 FROM PUBT
 GROUP BY PRID
 HAVING COUNT(*)>3);
```

Q5:

```
SELECT STID, COID
FROM STT, COT
WHERE STDEID='d222' AND CODEID='d222'
AND NOT EXISTS
          (SELECT *
 FROM STCOT
 WHERE STCOT.STID=STT.STID
AND STCOT.COID=COT.COID);
```

Q8:

```
WITH RECURSIVE
      PRECOURST(COID, PRECOID) AS
      SELECT COID, PRECOID
            FROM COPRECO
            UNION
            SELECT COPRECO.COID, PRECOURST.PRECOID
            FROM COPRECO, PRECOURST
            WHERE COPRECO.PRECOID=PRECOURST.COID
      SELECT C1.COTITLE, C2.COTITLE
      FROM PRECOURST, COT C1, COT C2
      WHERE PRECOURST.COID=C1.COID AND
            PRECOURST.PRECOID=C2.COID;
```

۱۲-۶:

به مثال ۴۲ از گفتار ۱۲ مراجعه شود.

۷-۱۲:

با استفاده از NOT و NOT EXISTS می‌توان آن را شبیه‌سازی کرد. علت را می‌توان در هم‌ارزی زیر جست:

$$\forall x\, P(x) \equiv \neg\exists x\, \neg P(x)$$

۹-۱۲:

در پاسخ به این سؤال، فرض شده است کارنامه شامل نام و شماره دانشجویی، نام درس‌ها، نمره و واحد هر درس، نام استاد درس، ترم-سال و همچنین جمع واحدها، جمع واحدهای گذرانده و جمع واحدهای نامشخص است. معدل ترم به این صورت محاسبه شده است که در صورتی که نمره هیچ درسی مشخص نباشد، معدل نامشخص است؛ در غیر اینصورت، با نمرات مشخص معدل محاسبه می‌شود.

چهار جدول STT، COT، PRT و STCOT را در نظر بگیرید.

STT (STID, STNAME, …)

COT (COID, COTITLE, CREDIT, …)

PRT (PRID, PRNAME, …)

STCOT (STID, COID, TR, YRYR, PRID, GRADE, …)

ابتدا چند دید بر روی این جداول تعریف می‌کنیم:

```
CREATE VIEW VIEW_OFFER AS
SELECT STID, STNAME, COID, COTITLE, GRADE, CREDIT, PRNAME,
       TR, YRYR
FROM STCOT, COT, PRT, STT
WHERE STCOT.PRID = PRT.PRID
AND STCOT.COID = COT.COID
AND STCOT.STID = STT.STID;

CREATE VIEW VIEW_GPA AS
SELECT STID, TR, YRYR, SUM (GRADE*CREDIT)/SUM(CREDIT) AS GPA
FROM VIEW_OFFER
WHERE GRADE NOT NULL
     GROUP BY STID, TR, YRYR;

CREATE VIEW VIEW_PASSED_CREDITS AS
SELECT STID, sum(CREDIT) AS SUM, TR, YRYR
FROM VIEW_OFFER
WHERE GRADE > 10
GROUP BY STID, TR, YRYR;

CREATE VIEW VIEW_ALL_CREDITS AS
SELECT STID, sum(CREDIT) AS SUM, TR, YRYR
```

```
FROM VIEW_OFFER
GROUP BY STID, TR, YRYR;

CREATE VIEW VIEW_NA_CREDITS AS
SELECT STID, sum(CREDIT) AS sum, TR, YRYR
FROM VIEW_OFFER
WHERE GRADE IS NULL
GROUP BY STID, TR, YRYR;
```

حال رویه موردنظر را می‌نویسیم:

```
CREATE PROC TRANSCRIPT
@ID CHAR(8), @YEAR CHAR(5), @TERM CHAR(1)
AS
SELECT *
FROM VIEW_OFFER
FULL OUTER JOIN VIEW_GPA
FULL OUTER JOIN VIEW_PASSED_CREDITS
FULL OUTER JOIN VIEW_NA_CREDITS
FULL OUTER JOIN VIEW_ALL_CREDITS
ON VIEW_OFFER.STID = @ID
AND VIEW_OFFER.TR = @TERM
AND VIEW_OFFER.YRYR = @YEAR;
```

سپس مانند مثال ۳۴ یا ۳۵ گفتار ۱۲ عمل کرده و پس از فراخوانی رویه‌ی نوشته شده، اطلاعات را به گونه‌ی موردنظر نمایش می‌دهیم.

۱۲-۱۰:

جداول پاسخ سؤال ۹ را در نظر بگیرید. ابتدا یک دید روی این جداول تعریف می‌کنیم.

```
CREATE VIEW VIEW_OFFERED_COURS AS
SELECT PRID, PRNAME, COID, COTITLE, TR, YRYR
FROM STCOT, COT, PRT
WHERE STCOT.PRID = PRT.PRID
AND STCOT.COID = COT.COID;
```

حال رویه موردنظر را می‌نویسیم:

```
CREATE PROC TEACHING
@ID CHAR(8), @YEAR CHAR(5), @TERM CHAR(1)
AS
SELECT *
FROM VIEW_OFFERED_COURS
ON VIEW_OFFERED_COURS.PRID = @ID
AND VIEW_OFFERED_COURS.TR = @TERM
AND VIEW_OFFERED_COURS.YRYR = @YEAR;
```

سپس مانند مثال ۳۴ یا ۳۵ گفتار ۱۲ عمل کرده و پس از فراخوانی رویه‌ی نوشته شده، اطلاعات را به گونه‌ی موردنظر نمایش می‌دهیم.

:۱۲-۱۲

I: x <> ALL R

II: x NOT IN R

اگر یک x مشخص وجود داشته باشد که برای آن I برقرار باشد، یعنی به ازاء همه‌ی عناصر r موجود درR، r با آن x مشخص برابر نیست. این جمله به این معنا است که برای آن x مشخص، x∉R و در نتیجه x NOT IN R.

از طرف دیگر فرض می‌کنیم x مشخصی وجود دارد که برای آن x NOT IN R. یعنی به ازاء همه‌ی عناصر r درونR، هیچ عنصری برابر آن x مشخص نیست. این جمله معادل است با اینکه برای همه‌ی عناصر r داریم r≠x. در نتیجه x<>ALL R.

پس ALL <> همان NOT IN است.

:۱۵-۱۲

در اینجا به نحوه‌ی نوشتن رهانا در سه سیستم PostgreSQL، DB2 و Sysbase می‌پردازیم. گفتنی است با توجه به شباهت نحوه‌ی نوشتن رهانا در این سیستم‌ها با آنچه در بخش ۳-۲-۱ گفتار ۱۲ آمده است، از توضیحات تکراری می‌پرهیزیم و تنها توضیحات تکمیلی هر یک می‌آوریم.

**رهانا در PostgreSQL[۵]**

نحوه‌ی نوشتن رهانا در این سیستم به صورت زیر است:

```
CREATE [ CONSTRAINT ] TRIGGER name
{ BEFORE | AFTER | INSTEAD OF } { event [ OR ... ] }
ON table    [ FROM referenced_table_name ]
[ NOT DEFERRABLE
 | [ DEFERRABLE ] { INITIALLY IMMEDIATE | INITIALLY DEFERRED } ]
[ FOR [ EACH ] { ROW | STATEMENT } ]
[ WHEN ( condition ) ]
    EXECUTE PROCEDURE function_name ( arguments )
```

توضیحات:

[۵] برگرفته از https://www.postgresql.org/docs/9.1/static/sql-createtrigger.html

- با گذاشتن گزینه‌ی CONSTRAINT در ابتدای دستور، یک رهانا از نوع «محدودیت-رهانا»[۶]تعریف می‌شود. این نوع رهانا همان رهانای عادی است با این تفاوت که زمانبندی اجرای آن را می‌توان با دستور SET CONSTRAINTS مشخص کرد (رجوع کنید به پاسخ سؤال ۵۸ همین گفتار.) گفتنی است این نوع رهانا تنها در حالت AFTER و به ازاء هر سطر، یعنی برای حالت FOR EACH ROW قابل تعریف است.
- Event همان رویداد است، اما افزون بر INSERT، DELETE و UPDATE می‌تواند TRUNCATE نیز باشد.[۷]
- Table نام جدول یا دیدی است که رهانا برای آن تعریف می‌شود. اما امکان‌پذیری تعریف رهانا برای جدول و یا دید بسته به دیگر گزینه‌ها تعیین می‌شود، از جمله زمان اجرای رهانا، رویداد و اینکه رهانا در سطح دستور اجرا می‌شود یا سطر. جدول زیر امکانپذیری تعریف رهانا برای دید یا جدول را در حالات گوناگون نمایش می‌دهد.

| زمان اجرا | رخداد | در سطح سطر | در سطح دستور |
|---|---|---|---|
| **BEFORE** | INSERT/UPDATE/DELETE | جدول | جدول و دید |
| | TRUNCATE | - | جدول |
| **AFTER** | INSERT/UPDATE/DELETE | جدول | جدول و دید |
| | TRUNCATE | - | جدول |
| **INSTEAD OF** | INSERT/UPDATE/DELETE | دید | - |
| | TRUNCATE | - | - |

- referenced_table_name نام جدول دیگر ارجاع داده شده در محدودیت است.[۸]
- زمانبندی پیش‌نهاده‌ی اجرای *محدودیت-رهانا* را می‌توان با گزینه‌هایی همچون NOT DEFERABLE یا DEFERABLE INITIALY IMMEDIATE تعیین کرد. توضیحات بیشتر درباره‌ی این گزینه‌ها را می‌توانید در پاسخ سؤال ۵۸ بخوانید.

---

[۶] CONSTRAINT TRIGGER

[۷] به کمک دستور TRUNCATE می‌توان تمام سطرهای یک جدول یا یک مجموعه از جدول‌ها را پاک کرد و کارکرد آن همانند اجرای دستور DELETE با شرط همیشه درست است.

[۸] استفاده از این گزینه در حالت کلی توصیه نمی‌شود و برای محدودیت کلید خارجی است.

- condition یک عبارت بولی است که در صورت درست بودن، رهانا اجرا می‌شود. امکان ارجاع به مقادیر ستون‌های یک سطر از جدول قدیم و جدید به ترتیب با پیشوند OLD و NEW در این عبارت وجود دارد (فقط در حالت FOR EACH ROW.) همچنین در این عبارت نمی‌توان زیرپرسش نوشت. افزون بر این، در حالت INSTEAD OF نمی‌توان از گزینه‌ی WHEN condition استفاده کرد.
- function-name نام تابعی است که در صورت برقراری شرایط رهانا، اجرا می‌شود. این تابع آرگومان ورودی ندارد و خروجی آن از نوع trigger است. البته امکان دسترسی به ستون‌های جدول قدیم و جدید در بدنه‌ی تابع با همان پیشوندهای OLD و NEW وجود دارد.

**رهانا در DB2[۹]**

نحوه‌ی نوشتن رهانا در این سیستم به صورت زیر است:

```
CREATE  TRIGGER  trigger-name
{NO CASCADE BEFORE | AFTER | INSTEAD OF}
     {INSERT | DELETE | UPDATE [OF column-name, …]}
ON  {table-name | view-name}
[ REFERENCING    [OLD [AS] correlation-name]
                 [NEW [AS] correlation-name]
                 [OLD-TABLE [AS] table-identifiers]
                 [NEW-TABLE [AS] table-identifiers]]
[  FOR EACH ROW  MODE DB2SQL [NOT SECURED | SECURED]
 | FOR EACH STATEMENT  MODE DB2SQL]
[WHEN  search-condition]
{ trigger-SQL-statement |
  BEGIN ATOMIC  trigger-SQL-statement, …  END }
```

توضیحات:

- در حالت NO CASCADE BEFORE رهانا هیچ رهانای دیگری را فعال نمی‌کند.
- در حالت NO CASCADE BEFORE و AFTER رهانا تنها برای جدول تعریف می‌شود و در حالت INSTEAD OF تنها برای دید.
- در تعریف رهانا برای دید امکان استفاده از کلاز WHEN و FOR EACH STATEMENT وجود ندارد.
- امکان تعریف رهانا برای هر نوع دیدی وجود **ندارد**، از جمله دیدی که با گزینه‌ی WITH CASCADE CHECK تعریف شده باشد. شرایط کامل دیدهای مجاز برای تعریف رهانا در منبع یاد شده آمده است.
- در حالت BEFORE و INSTEAD OF امکان استفاده از FOR EACH STATEMENT وجود ندارد.

[۹] برگرفته از https://www.ibm.com/support/knowledgecenter/SSEPEK_10.0.0/sqlref/src/tpc/db2z_sql_createtrigger.html

- بسته به زمان فعال‌سازی رهانا (BEFORE، AFTER و INSTEAD OF) مجاز بودن استفاده از انواع دستورات SQL در بدنه‌ی رهانا (trigger-SQL-statement) مشخص می‌شود. برای نمونه در حالت BEFORE امکان استفاده از دستورات DELETE، TRUNCATE و UPDATE وجود ندارد. محدودیت‌های کامل را می‌توانید در منبع یاد شده ببینید.

**رهانا در Sysbase[۱۰]**

نحوه‌ی نوشتن رهانا در این سیستم به شکل زیر است:

```
create trigger [owner.]trigger_name
     on [owner.]table_name
     {for {insert , update} | instead of {insert, update, delete}}
     [as
          [if update (column_name)
               [{and | or} update (column_name)]...]
               SQL_statements
          [if update (column_name)
               [{and | or} update (column_name)]...
               SQL_statements]...]
```

توضیحات:

- زمان فعال‌سازی BEFORE وجود ندارد.
- امکان گذاشتن ترکیبی از رویدادها برای یک رهانا وجود دارد، یعنی یک رهانا می‌تواند هم برای درج باشد و هم حذف.
- برای مشخص کردن ستون‌های جدول در رویداد بهنگام‌سازی باید از کلاز if update (column_name) استفاده کرد. همچنین به نظر می‌توان برای ستون‌ها و حالات گوناگون، بیش از یک بار از این کلاز استفاده کرد.
- به نظر سطح اجرای رهانا تنها در حالت دستوری است (هم‌ارز FOR EACH STATEMENT).
- برای ارجاع به جدول‌های قدیم و جدید به ترتیب از کلیدواژه‌های deleted و inserted استفاده می‌شود. (هم‌ارز OLD-TABLE و NEW-TABLE)

[۱۰] برگرفته از http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.dc36272.1570/html/commands/X19955.htm

- در اینجا نیز محدودیت‌هایی برای دستورات SQL استفاده‌شده در بدنه‌ی رهانا وجود دارد که در منبع یادشده آمده است.

۱۲-۱۶:

BLOB کوته‌نوشت Binary Large OBject است و CLOB نیز کوته‌نوشت Character Large OBject. این دو نوع (البته به همراه DBLOB) برای ذخیره‌سازی داده‌های بسیار بزرگ هستند. برای نمونه در سیستم DB2 بیشینه حجم یک داده از این انواع ۲ گیگابایت است. BLOB مخصوص رشته‌های دودویی (باینری) است و می‌تواند در ذخیره‌سازی فیلم یا صدا به کار رود. CLOB نیز برای رشته‌های متنی با کاراکترهای تک‌بایتی است و DBLOB نیز برای رشته‌ها با کاراکترهای دوبایتی.

از آنجایی که جابجایی این داده‌ها از پایگاه‌داده به سمت مشتری هزینه‌بردار و زمانگیر است، برنامه‌های کاربردی برای ارجاع به اینگونه داده‌ها از LOB Loc استفاده می‌کنند. در واقع LOB Loc یک متغیر در برنامه‌ی کاربردی به زبان میزبان است که به یک مقدار از این انواع اشاره می‌کند.

۱۲-۱۷:

ما در اینجا فهرست حضور و غیاب را به این صورت در نظر گرفته‌ایم که شامل نام و شماره و تعداد واحد درس، نیمسال و سال تحصیلی، نام استاد، شماره دانشجویی به صورت مرتب شده و نام دانشجویان است. همچنین، تنها هسته‌ی اصلی مربوط به SQL نوشته شده است. برای اینکه بتوانید پاسخ را با جزییات کامل بنویسید، به مثال‌های ۳۴ و ۳۵ گفتار ۱۲ مراجعه کنید.

با توجه به جداول تعریف شده در پاسخ سؤال ۹، به این صورت یک رویه تعریف می‌کنیم:

```
CREATE PROC PRESABS
@ID COID(5), @YEAR CHAR(5), @TERM CHAR(1)
AS
SELECT COTITLE, COID, TR, YRYR, CREDIT, PRNAME, STNAME, STID
FROM STCOT JOIN PRT JOIN COT JOIN STT
WHERE STCOT.COID = @ID
AND STCOT.TR = @TERM
AND STCOT.YRYR = @YEAR
ORDER BY STID;
```

سپس مانند مثال ۳۴ یا ۳۵ گفتار ۱۲ عمل کرده و پس از فراخوانی رویه‌ی نوشته شده، اطلاعات را به گونه‌ی موردنظر نمایش می‌دهیم.

۱۲-۱۸:

توجه داشته باشید که در پاسخ این سؤال، تنها هسته‌ی اصلی مربوط به SQL که یک رویه است نوشته شده است. برای اینکه بتوانید پاسخ را با جزییات کامل بنویسید، به مثال ۳۴ گفتار ۱۲ مراجعه کنید.

با توجه به دیدها و جداول تعریف شده در پاسخ سؤال ۹، به این صورت یک رویه تعریف می‌کنیم:

```
CREATE PROC CONDSTUD
@YEAR CHAR(5), @TERM CHAR(1)
AS
SELECT STID, STNAME
FROM VIEW_GPA
WHERE VIEW_GPA.GPA < 12
AND VIEW_GPA.TR = @TERM
AND VIEW_GPA.YRYR = @YEAR;
```

توجه داشته باشید که دید VIEW_GPA شامل STID، TR، YRYR و GPA است.

۱۲-۲۲:

برای این‌کار می‌توان گام‌های زیر را پیمود:

۱. ساخت جدولی جدید با نامی دیگر ، شامل تمام ستون‌های جدول موردنظر (ترجیحاً با همان نام‌ها) به جز ستون(های) موردنظر برای حذف.

۲. انتقال تمام تاپل‌های جدول موردنظر به جدول جدید به کمک درج گروهی.

۳. حذف جدول موردنظر از کاتالوگ.

۴. ساخت جدولی جدید هم‌نام با جدول موردنظر ، شامل تمام ستون‌های آن جدول به جز ستون(های) موردنظر برای حذف.

۵. انتقال تمام تاپل‌ها از جدول ساخته شده در گام ۱ به جدول ساخته شده در گام ۴.

۶. حذف جدول ساخته شده در گام ۱.

برای نمونه فرض کنید می‌خواهیم ستون مربوط به شماره شناسنامه (BIRTH_CERTIFICATE_ID) را از جدول PERSON حذف کنیم. جدول PERSON به صورت زیر است :

PERSON (NATIONAL_ID, NAME, BIRTH_CERTIFICATE_ID, … )

برای این کار می‌توان چنین عمل کرد :

```
1. CREATE TABLE PERSON_TEMP
(NATIONAL_ID CHAR(12) NOT NULL,
NAME CHAR(255) NOT NULL,
…)
PRIMARY KEY(NATIONAL_ID)…;

2. INSERT INTO PERSON_TEMP
SELECT NATIONAL_ID, NAME, …
FROM PERSON;

3. DROP TABLE PERSON;

4. CREATE TABLE PERSON
(NATIONAL_ID CHAR(12) NOT NULL,
NAME CHAR(255) NOT NULL,
…)
PRIMARY KEY(NATIONAL_ID))…;

5. INSERT INTO PERSON
SELECT NATIONAL_ID, NAME, …
FROM PERSON_TEMP;

6. DROP TABLE PERSON_TEMP;
```

۱۲-۲۴:

با فرض هیچمقدارناپذیری کلیدهای خارجی در هر دو جدول، به خاطر قاعده‌ی C2 در حالی که جدول T2 هنوز خالی است، امکان درج سطر در هیچکدام از دو جدول نیست. زیرا در صورتی که بخواهیم نخست در جدول T1 سطری را درج کنیم، هر مقداری که برای کلید خارجی در این سطر بگذاریم در جدول T2 مقدار مرجع ندارد (جدول T2 خالی است) و از طرفی به دلیل هیچمقدارناپذیری، حتماً باید مقداری برای آن بگذاریم. بنابراین درخواست درج سطر رد می‌شود. همچنین اگر بخواهیم نخست در جدول T2 سطری را درج کنیم، مشکل مشابه برای کلید خارجی این جدول پیش خواهد آمد.

یک راهکار برای این مشکل، به تعویق انداختن اجرای قاعده‌ی C2 در سیستم تا پایان اجرای تراکنش است به گونه‌ای که در لحظه‌ی درج سطرها در دو جدول، این قاعده بررسی نشود و تنها پس از اجرای کامل تراکنش و درج سطرها در هر دو جدول، این قاعده بررسی شود. برای این کار می‌توان از دستور SET CONSTRAINT استفاده کرد. توضیحات مربوط به این دستور در پاسخ سؤال ۵۸ آمده است.

۱۲-۲۵:

عملگرهای UNION و EXCEPT را در SQL می‌توان در عملیات بازیابی و بین عبارات بازیابی (… SELECT) به کار برد. بنابراین به صورت عادی نمی‌توان از آن‌ها برای عملیات ذخیره‌سازی استفاده کرد. اما می‌توان نتیجه‌ی عبارات بازیابی دارای این عملگرها را در یک جدول موقت یا جدول پاسخ یک پرسش ذخیره کرد. یا به کمک عملگر INSERT

نتیجه‌ی آن را در جدولی مبنا ذخیره کرد. بنابراین این عملگرها به تنهایی امکان ذخیره‌سازی داده را به ما نمی‌دهند. همچنین استفاده از عملگر INSERT نیز لزوماً عملکرد مورد نظر ما را به دست نمی‌آورد. برای نمونه جدول T(A, B, C) را در نظر بگیرید. اگر بخواهیم سطر با مقدار کلید a1 را از این جدول حذف کنیم، چنین دستوری این کار را **نمی‌کند**:

```
INSERT INTO T VALUES (
      SELECT * FROM T
            EXCEPT
      SELECT * FROM T WHERE A='a1')
```

حتی این دستور ممکن است با خطا مواجه شود، زیرا می‌خواهد تمام سطرها به جز سطر با کلید a1 را دوباره در T درج کند و به دلیل تکراری بودن مقدار کلید اجازه‌ی درج این سطرها را نمی‌دهد. راهکار درست آن است که نتیجه‌ی عبارت بازیابی را در یک جدول موقت ذخیره کرد. سپس تمام سطرهای جدول T را حذف کرد. سپس سطرهای جدول موقت را در جدول اصلی ذخیره کرد. در این صورت هم از دستور DELETE استفاده کرده‌ایم و هم دستور INSERT. حال آنکه می‌توانستیم در یک گام و با دستور DELETE این کار را انجام دهیم. به صورت زیر:

```
DELETE * FROM T WHERE A='a1'
```

بنابراین تنها با دستورات EXCEPT و UNION نمی‌توان عملیات ذخیره‌سازی را انجام داد.

۱۲-۲۶:

$R$

| $\underline{A}$ | $B$ | $C$ |
|---|---|---|
| $a1$ | $b1$ | $c1$ |
| $a2$ | $b1$ | $c1$ |
| $a3$ | $b2$ | $c2$ |

نتیجه پرسمان I خواهد شد:

| $B$ | $C$ |
|---|---|
| $b1$ | $c1$ |
| $b1$ | $c1$ |
| $b2$ | $c2$ |

نتیجه پرسمان II خواهد شد:

| $B$ | $C$ |
|---|---|
| $b1$ | $c1$ |
| $b2$ | $c2$ |

۱۲-۲۷: نتیجه‌ی حاصل از ارزیابی «درست» خواهد بود.

۱۲-۲۸: حاصل عملگر پرتو شامل مقادیر تکراری نیست اما در پیاده‌سازی SQL آن، موارد تکراری حذف نمی‌شوند مگر اینکه صریحاً اعلام شود. بنابراین، هنگامی دو عبارت داده شده نتیجه یکسان دارند که ستون NAME فاقد مقادیر تکراری باشد.

۱۲-۲۹: نتیجه اجرای دستور در سه بسته‌ی MySQL 5.6، PostgreSQL 9.3 و MS SQL Server 2014 برابر صفر است.

۱۲-۳۰: یک محدودیت پایگاهی؛ مجموع تعداد تاپل‌های R1 و R2 باید کمتر از ۲۰۰ باشد.

۱۲-۳۱:

برای حل این سؤال، ابتدا باید از NX آغاز کنیم. به ازای هر سطر، اگر X بیشتر از ۳۰۰ باشد ۱۰۰ واحد از آن می‌کاهد؛ در غیر این صورت، ۱۰۰ واحد به آن می‌افزاید. در نتیجه NX به ترتیب چنین خواهد شد: ۲۰۰، ۲۵۰، ۳۰۰ و ۳۰۰.

سپس با توجه به شرط NX BETWEEN(250, 300) سه سطر آخر انتخاب خواهند شد و خواهیم داشت:

| $NX$ | $Y$ | $Z$ |
|---|---|---|
| 250 | $y1$ | $z2$ |
| 300 | $y2$ | $z3$ |
| 300 | $y2$ | $z4$ |

۱۲-۳۲: دستور UNION JOIN دستوری است مشابه عملگر OUTER UNION که در SQL:1992 اضافه شده و قرار بوده که در SQL:2003 حذف شود. رابطه‌ی R چنین خواهد شد:

| $A$ | $B$ | C | $D$ |
|---|---|---|---|
| 1 | 2 | 3 | $NULL$ |

| | | | |
|---|---|---|---|
| 4 | 5 | 6 | $NULL$ |
| 7 | 8 | 9 | NULL |
| 11 | 12 | NULL | 13 |
| 14 | 15 | NULL | 16 |

در نتیجه درجه رابطه ۴ و کاردینالیتی آن ۵ خواهد بود.

۱۲-۳۳:

ممکن است این طور به نظر آید که این پرسش در پاسخ به خواسته‌ی زیر نوشته شده است:

«سطرهایی از جدول R را بدهید که مقدار ستون A از آن سطر حداقل با مقدار در ستون A در یکی از دو جدول S یا T برابر باشد.»

در این صورت هم‌ارز عبارت $(R \cap S) \cup (R \cap T)$ است. حال آنکه این استدلال غلط است. زیرا در صورت تهی بودن هر یک از جداول S و T نتیجه‌ی پرسش تهی است. اما نتیجه‌ی این عبارت جبر رابطه‌ای لزوماً تهی نیست.

اما پیش از آنکه بگوییم این عبارت هم‌ارز کدام عبارت جبر رابطه‌ای است باید یادآور شویم از آنجایی که SELECT سطرهای تکراری را به خودی خود حذف نمی‌کند، این عبارت **نمی‌تواند** هم‌ارز هیچ عبارت جبر رابطه‌ای باشد. اما اگر از این مساله با تسامح بگذریم، می‌توان آن را هم‌ارز عبارت زیر دانست:

$$\pi_{<A>}(\sigma_{<A=SA\ OR\ A=\square A>}(A \times (S\ RENAME\ A\ AS\ SA) \times (T\ RENAME\ A\ AS\ TA)))$$

۱۲-۳۴: جدول جواب شامل دو ستون A و AVG(B) خواهد بود. ذیل ستون A مقدار آن در نخستین سطر جدول Rو ذیل ستون AVG(B) میانگین مقادیر ستون B وجود خواهد داشت. در صورتی که یکی از مقادیر ستون B برابر NULLباشد، در محاسبه‌ی AVG(B) نخواهد داشت.

۱۲-۳۵: به جدول حاصل از ارزیابی SELECT R2.A FROM R2 جدول T و به سطر حاصل از ارزیابی R1.A، R1می‌گوییم. در این صورت حاصل ارزیابی MATCH UNIQUE هنگامی «درست» است که T دقیقا یک سطر مساوی R1 داشته باشد.

۱۲-۳۶: با اجرای هر یک از دستورات ... ALTER TABLE و ... DROP TABLE کاتالوگ تغییر خواهد کرد، زیرا:

- با اجرای دستور نخست، اطلاعات مربوط به ستون‌های یک جدول تغییر خواهد کرد (مانند اضافه شدن یک ستون جدید و یا حذف یک ستون از جدول) و از آنجایی که کاتالوگ اطلاعات مربوط به ستون‌های هر جدول را در خود نگه می‌دارد، کاتالوگ در نتیجه‌ی اجرای این دستور تغییر خواهد کرد.
- با اجرای دستور دوم، اطلاعات مربوط به یک جدول از کاتالوگ حذف خواهد شد.

اما با اجرای دستور DELETE FROM TABLE ... کاتالوگ لزوماً تغییر نخواهد کرد، بلکه می‌تواند تحت شرایطی تغییر کند. برای نمونه، اگر کاتالوگ تعداد سطرهای هر جدول را در خود نگه دارد، با اجرای این دستور و حذف حداقل یک سطر کاتالوگ تغییر خواهد کرد. اما در حالت کلی‌تر، در صورت نگهداری اطلاعاتی که ناظر به و یا متأثّر از اندازه و تعداد سطرهای جدول است (چه مستقیم و چه غیرمستقیم)، با اجرای این دستور ممکن است کاتالوگ تغییر کند. برای نمونه، اگر با حذف یک سطر، یک صفحه از صفحات یک جدول نیز حذف شود، در صورتی که کاتالوگ شمار صفحات هر جدول را داشته باشد، کاتالوگ همچنان تغییر خواهد کرد.

۱۲-۳۸:

یک راه ساده و الگوریتمی برای شبیه‌سازی عملگر *EXCEPT* در SQL، به‌کارگیری عملگر NOT IN در کلاز WHERE است. در واقع کافی است یک شرط را با شرط‌های کلاز WHERE عبارت SQL اول (عملوند اول عملگر EXCEPT) عطف[۱۱] کنیم، و یا در صورت نبود کلاز WHERE، این شرط را به همراه کلاز WHERE به این عبارت بیافزاییم. این شرط عبارت است از «نبودن سطر مورد نظر در عبارت دوم (عملوند دوم عملگرEXCEPT )، به کمک عملگر NOT IN.»

برای نمونه، به پرسش روبرو در کتاب پایگاه داده DATE دقت کنید: «نام و شماره‌ی تهیه‌کنندگانی را بدهید که قطعه با شماره‌ی p2 را تهیه نکرده باشند». این پرسش را می‌توان به کمک عملگر EXCEPT به صورت زیر نوشت:

```
SELECT  S#, SNAME
FROM  S

EXCEPT
SELECT  S.S#,  S.SNAME
FROM  S, SP
WHERE  S.S#=SP.S#  AND  SP.P#='p2'
```

عبارت فوق را می‌توان با کمک عملگر NOT IN به صورت زیر بازنویسی کرد:

```
SELECT  S#, SNAME
FROM  S
WHERE  (S#, SNAME)  NOT IN (
SELECT  S.S#,  S.SNAME
FROM  S, SP
```

[۱۱] AND

```
WHERE  S.S#=SP.S#  AND  SP.P#='p2')
```

البته شبیه‌سازی عملگر EXCEPT به کمک عملگر NOT EXISTS نیز امکان‌پذیر است.

کنجکاوی: پرسش فوق را به کمک عملگر NOT EXISTS بازنویسی کنید.

کنجکاوی: به غیر از NOT IN و NOT EXISTS، به کمک چه عملگرهایی می‌توان عملگر EXCEPT را شبیه‌سازی کرد؟

کنجکاوی: الگوریتم تبدیل عبارت SQL شامل عملگر EXCEPT به عبارت SQL هم‌ارز بدون این عملگر و با استفاده از عملگر NOT IN را به یک زبان برنامه‌سازی دلخواه بنویسید.

۱۲-۳۹:

```
SELECT A1, …, An FROM R1
WHERE B1, …, Bm IN (SELECT B1, …, Bm FROM R2)
GROUP BY A1, …, An
HAVING COUNT(*) = (SELECT COUNT(*) FROM R2);
```

۱۲-۴۰:

این دو جدول از طریق دو کلید خارجی متقابلاً به یکدیگر ارجاع دارند و مشکل یاد شده در سؤال ۲۴ در اینجا نیز پدید می‌آید. (راهکار مشکل نیز در پاسخ سؤال ۲۴ اشاره شده است)

۱۲-۴۱:

رجوع شود به بخش ۱-۱۱ و ۲-۱۴ در گفتار ۱۲.

۱۲-۴۲:

ارتباط تشکیل شدن قطعه از قطعات بسیار همانند ارتباط پیشنیازهای یک درس است و برنامه‌ی خواسته شده در این سؤال همانند برنامه‌ی نوشته شده در بخش ۲-۱۴ است که پیشنیازهای درس با شماره‌ی ’COM222‘ را برمی‌گرداند.

۱۲-۴۳:

- درج تاپل در STCOT: به عهده‌ی خواننده
- حذف تاپل از COT: با توجه به قاعده‌ی C2 (ارجاع ستون COID از جدول STCOT به ستون COID در جدول COT)، در صورت وجود سطرهایی ارجاع‌کننده از جدول STCOT به سطر در حال حذف در COT، به دو صورت می‌توان عمل کرد:

- جلوگیری از عمل حذف
- حذف تمام سطرهای ارجاع‌کننده

در حالت نخست، رهانای زیر قاعده‌ی جامعیت C2 را در این عمل کنترل می‌کند:

```
CREATE TRIGGER TDC1
BEFORE DELETE ON COT
REFERENCING OLD ROW AS OCOT
WHEN (EXISTS (SELECT * FROM STCOT WHERE STCOT.COID=OCOT.COID))
ABORT TRANSACTION
FOR EACH ROW
```

در حالت دوم نیز می‌توان از رهانای زیر بهره برد:

```
CREATE TRIGGER TDC2
BEFORE DELETE ON COT
REFERENCING OLD ROW AS OCOT
DELETE FROM STCOT WHERE STCOT.COID=OCOT.COID
FOR EACH ROW
```

- بهنگام‌سازی شماره دانشجو و شماره گروه آموزشی او: به عهده‌ی خواننده (راهنمایی: همانند حذف تاپل از COT)
- جلوگیری از اخذ بیش از n واحد در یک ترم توسط دانشجویانی که در ترم قبل معدل کمتر از a دارند: با فرض وجود رویه‌های AverageSTTR، SumCreditsSTTR، PrevTRTR، و PrevTRYR که به ترتیب معدل یک دانشجو در یک ترم-سال، جمع تعداد واحدهای اخذ شده توسط یک دانشجو در یک ترم-سال، شماره ترم قبل یک ترم-سال، و شماره سال ترم-سال قبل یک ترم-سال را می‌دهند، و با این فرض که محدودیت به ازاء درج هر سطر در جدول STCOT بررسی می‌شود، رهانای مورد نظر به صورت زیر می‌تواند باشد:

```
CREATE TRIGGER TAE
AFTER INSERT ON STCOT
REFERENCING NEW ROW AS NSTCOT
WHEN ( SumCreditsSTTR(NSTCOT.STID, NSTCOT.TR, NSTCOT,YR)>n
AND AverageSTTR(NSTCOT.STID, PrevTRTR(NSTCOT.TR), PrevTRYR(NSTCOT.TR,
NSTCOT.YR))<a )
DELETE FROM STCOT WHERE STCOT.STID=NSTCOT.STID AND
                        STCOT.COID=NSTCOT.COID AND
                        STCOT.TR=NSTCOT.TRY AND
                        STCOT.YR=NSTCOT.YR
```

- درج تاپل در دید V( COID, COTITLE, COCRED, CODEID): در این رهانا به جای درج سطر در این دید باید سطری را در جدول مبنای COT درج کرد (با فرض تکراری نبودن مقدار کلید). تنها باید به

این نکته توجه داشت که حین درج سطر در جدول COT برای ستون COTYPE که بیانگر نوع درس است و در این دید وجود ندارد، مقدار 'pr' (درس عملی) گذاشته می‌شود.

```
CREATE TRIGGER TIV
      INSTEAD OF INSERT ON V
      REFERENCING NEW ROW AS NV
INSERT INTO COT VALUES (NV.COID, NV.COTITLE, NV.COCRED, 'pr', NV.CODEID)
```

- جلوگیری از درج تاپل در STT در صورت مشخص نبودن مقدار صفت «نمره» در آن پاپل: به عهده‌ی خواننده
- اِعمال این محدودیت که «مقدار نمره وارد شده هیچگاه تغییر نمی‌کند» : به عهده‌ی خواننده

۱۲-۴۵:

انواع جدول به شرح زیر است:

- **جدول مبنا:** تعریف و مثال در بخش ۱-۴
- **جدول موقت:** تعریف در بخش ۱-۷

مثال: از جدول موقت زیر می‌توان برای نگهداری نمرات یک دانشجو در دروس ثبت‌نامی وی استفاده کرد. به کمک این جدول می‌توان مواردی همچون تولید کارنامه‌ی ترمی یا تعیین وضعیت تحصیلی دانشجو استفاده کرد:

```
CREATE TEMPORARY TABLE STGrades
      (COID       CHAR(6),
      COTITLE     CHAR(16),
      CREDIT      SMALLINT,
      TR          CHAR(1),
      YRYR        CHAR(5),
      GRADE       DEC(2,2))
```

- **جدول همانند:** تعریف و مثال در بخش ۱-۸
- **جدول جواب پرسش:** تعریف و مثال در بخش ۱-۹
- **جدول مجازی (دید خارجی):** تعریف و مثال در بخش ۱-۱۰
- **دید موقت:** تعریف و مثال در بخش ۱-۱۱ (با فرض اینکه دید بازگشتی نیز نوعی دید موقت است. برای دید بازگشتی به ۲-۱۴ رجوع کنید)

- **دید ذخیره‌شده:** دیدی است که نتیجه‌ی آن همانند یک جدول در پایگاه داده ذخیره شده است. بنابراین مجازی نیست و در بازیابی محاسبه نمی‌شود. در نتیجه می‌تواند به تسریع در بازیابی اطلاعات کمک کند.
- در SQL SERVER 2016[۱۲] با ساخت نمایه‌ی خوشه‌ساز[۱۳] بر روی یک دید، می‌توان یک دید ذخیره‌شده ساخت که به *Indexed view*[۱۴] خوانده می‌شود. این دید همانند جدول با نمایه‌ی خوشه‌ساز ذخیره می‌شود. البته امکان تعریف ساخت نمایه‌ی خوشه‌ساز برای هر دیدی وجود ندارد. شرایط و نکات مربوط به ساخت اینگونه دید را می‌توانید در منبع آن ببینید. همچنین بهینه‌ساز این سیستم بدون ارجاع به اینگونه دید در پرسش و به طور خودکار از آن استفاده می‌کند. برای این امر به مثال آورده شده در منبع دقت کنید.

:۴۷-۱۲

کلاز **LATERAL** را می‌توان پیش از یک زیرپرسش در کلاز **FROM** آورد. در این صورت می‌توان در زیرپرسش به ستون‌های موارد پیشین آن، یعنی جدول‌ها یا زیرپرسش‌های از پیش‌آمده در کلاز **FROM** ارجاع داد. برای نمونه به پرسش زیر در پایگاه‌داده‌ی تهیه‌کننده-قطعه از DATE دقت کنید:

```
SELECT  *
FROM   SP, LATERAL (SELECT SNAME FROM S WHERE S.S# = SP.S#) A
```

این پرسش نسبتاً ساده شماره و نام تهیه‌کنندگان را به همراه اطلاعات قطعات تهیه‌شده توسط آن‌ها می‌دهد. چنانچه می‌بینید در زیرپسشی که پس از کلاز **LATERAL** آمده است به جدول SP که پیش از آن در کلاز **FROM** آمده است ارجاع داده شده است.

بنابراین، زیرپرسش پس از کلاز **LATERAL** را می‌توان یک زیرپرسش بهم‌بسته دانست که به ازاء هر سطر از موارد مرجع از پیش آمده در کلاز **FROM** اجرا می‌شود و نتیجه‌ی حاصل با آن سطر ضرب دکارتی می‌شود. این بدان معناست که در صورت تهی بودن نتیجه‌ی اجرای زیرپرسش به ازاء یک سطر از جدول(های) مرجع، آن سطر نادیده گرفته می‌شود، زیرا نتیجه‌ی ضرب دکارتی با جدول (رابطه‌ی) تهی، تهی است. به خاطر همین امر گاه برای پرهیز از حذف اینگونه سطرها کلاز LATERAL را با عملگر پیوند بیرونی چپ (LEFT OUTER JOIN) و به عنوان عملوند راست آن به کار می‌برند. در این صورت امکان ارجاع به عملوند سمت چپ نیز وجود دارد. برای نمونه به پرسش زیر دقت کنید:

```
SELECT  sname, A.*
     FROM  S  LEFT OUTER JOIN
LATERAL (SELECT * FROM SP WHERE SP.S# = S.S#) A ON TRUE
```

[۱۲] https://msdn.microsoft.com/en-us/library/ms191432.aspx

[۱۳] Clustered Index

[۱۴] دید نمایه‌دار

این پرسش همانند پرسش پیشین است با این تفاوت که نام تهیه‌کنندگانی که هیچ قطعه‌ای تهیه نکرده باشند را نیز برمی‌گرداند.

هرچند کلاز LATERAL نیز همانند برخی دیگر از کلازها در SQL، افزونه است، اما کاربرد اصلی آن در فراخوانی توابع است. در این حالت می‌توان برخی از ستون‌های جدول و/ یا زیرپرسش (های) مرجع را به عنوان پارامتر ورودی به تابع داد. برای مثال پرسش زیر شماره‌ی زوج تهیه‌کنندگان همشهری را می‌دهد که حداقل یک قطعه‌ی مشترک تهیه کرده باشند. در این پرسش تابع CoCitySuppliers به کار رفته است که شماره‌ی تهیه‌کنندگان همشهر یک تهیه‌کننده را برمی‌گرداند:

```
SELECT  SP1.S#, A.S#
     FROM   SP  SP1,
          (SP AS SP2
JOIN
LATERAL (CoCitySuppliers(SP2.S#)) CS
ON S#) A
```

۱۲-۴۹:

این دو کلاز در تعریف دید و در کلاز WITH CHECK OPTION به کار می‌روند، به صورت زیر:

- WITH LOCAL CHECK OPTION
- WITH CASCADED CHECK OPTION

کاربرد این دو کلاز مشخص کردن شرط‌های مورد بررسی در عملیات درج و بهنگام‌سازی در دیدها است. برای روشن‌تر شدن این موضوع، با فرض وجود جدول Person (Name) به سه دید زیر توجه کنید[۱۵]:

```
CREATE VIEW V1
     AS SELECT * FROM Person WHERE Name LIKE ‘A%’
CREATE VIEW V2
     AS SELECT * FROM V1 WHERE Name LIKE ‘%Z’
     WITH LOCAL CHECK OPTION
CREATE VIEW V3
     AS SELECT * FROM V1 WHERE Name LIKE ‘%Y’
     WITH CASCADED CHECK OPTION
```

سوالی که در اینجا مطرح آن است که در عملیات درج یا بهنگام‌سازی در هر یک از دیدهای فوق، کدام یک از شرط‌های موجود در کلازهای WHERE در تعریف سه دید بررسی می‌شوند. در پاسخ به این سوال:

[۱۵] برگرفته با تغییرات از مثال http://www.ibm.com/support/knowledgecenter/ssw_i5_54/sqlp/rbafyexcascade.htm

۱. در عملیات درج بر روی V1، هیچ شرطی بررسی نمی‌شود. زیرا تعریف این دید فاقد کلاز WITH CHECK OPTION است. بنابراین، حتی امکان درج نامی که با A شروع نمی‌شود نیز وجود دارد، هرچند این خلاف معنای دید V1 است.

۲. در عملیات درج بر روی V2، تنها شرط موجود در تعریف همان دید بررسی می‌شود. بنابراین، تنها امکان درج نام‌هایی که با Z پایان می‌یابند وجود دارد. در غیر این صورت، درخواست درج رد می‌شود.

۳. در عملیات درج بر روی دید V3، هم شرط موجود در تعریف دید V3 بررسی می‌شود و هم شرط تعریف دید V1. بدین ترتیب، تنها امکان درج نام‌هایی وجود دارد که هم با A شروع می‌شوند و هم با Y تمام می‌شوند.

با توجه به این پاسخ، می‌توان دریافت که:

- آوردن کلاز LOCAL بدین معناست که شرط موجود در تعریف دید بررسی می‌شوند، و نه **لزوماً** شرط(های) دیدهای زیرین (دیدهایی که این دید، مستقیم یا غیرمستقیم، بر روی آن‌ها تعریف شده است).
- آوردن کلاز CASCADE به معنای بررسی تمام شرط‌های موجود در تعریف دید و دیدهای زیرین آن است.

حال به دو دید زیر که بر روی دیدهای V2 و V3 تعریف شده‌اند توجه کنید:

```
CREATE VIEW V4
      AS SELECT * FROM V2 WHERE Name LIKE 'AB%'
CREATE VIEW V5
      AS SELECT * FROM V4 WHERE Name LIKE '%XZ'
      WITH LOCAL CHECK OPTION
```

در عملیات درج بر روی هر یک از این دو دید، سیستم به صورت زیر عمل می‌کند:

- V4: بررسی شرط موجود در تعریف دید V2
- V5: بررسی شرط موجود در تعریف دید V2 و V5

در اینجا باید توجه کنیم که هرچند دید V4 دارای کلاز WITH CHECK OPTION نیست، اما عملیات درج بر روی این دید، منطقاً تبدیل به عملیات درج بر روی دو دید V2 می‌شود. و با توجه به آنچه پیشتر گفته شد، شرط تعریف دید V2 در مورد آن‌ها بررسی می‌شود، هرچند شرط موجود در تعریف دید V4 و V5 بررسی نمی‌شود. همچنین برای دید V5 شرط تعریف دید V2 بررسی می‌شود. اما افزون بر آن، شرط تعریف دید V5 نیز به دلیل کلاز WITH LOCAL CHECK OPTION در نظر گرفته می‌شود.

۱۲-۵۰:

رجوع شود به بخش ۱-۱۰ گفتار ۱۲.

۵۱-۱۲:

در صورتی که هیچکدام از جداول T1 و T2 تهی نباشند، این مقادیر بازیابی می‌شوند. اما حتی اگر یکی از این دو جدول تهی باشد، از آنجایی که حاصل ضرب دکارتی سه جدول تهی می‌شود، هیچ‌یک از مقادیر T.a بازیابی نمی‌شوند.

۵۲-۱۲:

الف:

```
SELECT STID,
      (CASE
WHEN SCORE<40 THEN 'F'
            WHEN SCORE<60 AND SCORE>=40 THEN 'C'
            WHEN SCORE<80 AND SCORE>=60 THEN 'B'
WHEN SCORE>=80 THEN 'A'
ELSE 'WRONG SCORE'
      END)
FROM MARKS;
```

ب:

```
SELECT L_SCORE, COUNT(L_SCORE)
FROM (SELECT STID,
            (CASE
                  WHEN SCORE<40 THEN 'F'
                  WHEN SCORE<60 AND SCORE>=40 THEN 'C'
                  WHEN SCORE<80 AND SCORE>=60 THEN 'B'
                  WHEN SCORE>=80 THEN 'A'
                  ELSE 'WRONG SCORE'
            END) AS L_SCORE
      FROM MARKS) AS STID_LSCORE
GROUP BY L_SCORE;
```

۵۳-۱۲:

۱ نادرست است، زیرا رابطه‌های STT و STCOT دارای ستون‌های هم‌نام هستند (STID) و باید ستون‌های هم‌نام را حداقل در یک رابطه دگرنامید.

۲ درست است.

۳ درست است.

۴ نادرست است، به همان دلیل گزینه‌ی ۱. همچنین، خروجی این عبارت، دارای مقادیر تکراری است و بنابراین رابطه نیست، حال آنکه در مدل رابطه‌ای، حاصل یک عبارت (جبر یا حساب رابطه‌ای) همواره یک رابطه است و رابطه نیز عنصر تکراری ندارد.

۵ نادرست است، زیرا در مدل رابطه‌ای، حاصل یک عبارت همواره یک رابطه است و رابطه نیز یک مجموعه است و عناصر مجموعه ترتیب ندارند.

۱۲-۵۴:

روش اول:

```
SELECT  *  FROM  PROF
     WHERE PRID IN (SELECT PRID FROM PRCO
                    GROUP BY PRID
                    HAVING COUNT(*)=1)
```

روش دوم:

```
SELECT * FROM PROF
     WHERE PRID IN (SELECT PRID FROM PRCO) AND
PRID NOT IN (SELECT PRCO1.PRID FROM PRCO PRCO1, PRCO PRCO2
                          WHERE PRCO1.PRID=PRCO2.PRID
                               AND PRCO1.COID<>PRCO2.COID)
```

روش سوم:

```
SELECT * FROM PROF
     WHERE EXISTS (SELECT * FROM PRCO
WHERE PRCO.PRID=PROF.PRID AND
                          NOT EXISTS (SELECT * FROM PRCO PRCO1
                                         WHERE
PRCO1.PRID=PRCO.PRID
                                         AND
PRCO1.COID<>PRCO.COID))
```

۱۲-۵۵:

- سور رابطه‌ای این عبارت در SQL به صورت زیر است:

```
NOT  EXISTS (
     ((SELECT  A  FROM  R1) EXCEPT (SELECT  A FROM R2))
     UNION
     ((SELECT  A  FROM  R2) EXCEPT (SELECT  A FROM R1))
)
```

البته این عبارت را می‌توان با شبیه‌سازی EXCEPT نیز به روش‌های دیگر نوشت. (رجوع کنید به پاسخ سؤال ۳۸)

یک کاربرد این سور برای جدولی است که تجزیه‌ی عمودی شده است و کلید اصلی هر دو جدول یکی است. در این حالت اگر در هر یک از دو جدول حداقل یک ستون هیچ‌قدم‌ناپذیر به جز کلید اصلی وجود داشته باشد، ستون کلید اصلی (ساده یا مرکب) در هر دو جدول باید یکی باشند.

- سور رابطه‌ای این عبارت در SQL به صورت زیر است:

```
NOT EXISTS ((SELECT A FROM R2) EXCEPT (SELECT A FROM R1))
AND
EXISTS ((SELECT A FROM R1) EXCEPT (SELECT A FROM R2))
```

البته برای زیرمجموعه (و نه زیرمجموعه‌ی محض) نیازی به سور دوم نیست.

در صورتی که منظور زیرمجموعه باشد، و نه زیرمجموعه‌ی محض، کاربرد این عبارت در بررسی قاعده‌ی C2 (کلید خارجی) است. اما در صورتی که منظور براستی زیرمجموعه‌ی محض باشد، یک کاربرد آن در یک نوع طراحی جدولی (رابطه‌ای) برای مدلسازی زیرنوع-زبرنوع است. اگر جدول زیرنوع و زبرنوع جدا از یکدیگر باشند و تخصیص ناقص باشد، ستون کلید اصلی از جدول زیرنوع باید زیرمجموعه‌ی محض ستون کلید اصلی از جدول زبرنوع باشد.

۱۲-۵۶:

استفاده از رهانا در موارد زیر توصیه نمی‌شود:

- در مواردی که جایگزین سازوکارهای دیگری در SQL وجود دارد و کارآیی آن از رهانا بیشتر است؛ برای نمونه محدودیت قاعده‌ی C2 و یا هیچمقدارپذیری یک ستون.
- در مواردی که اجرای راهانا مستقل از اجرای دستورات اصلی تراکنش است و در واقع می‌توان آن (ها) را در تراکنشی دیگر اجرا کرد، بویژه اگر احتمال اجرا نشدن رهانا وجود دارد و یا رهانا بسیار زمانگیر است (در مقایسه با تراکنش اصلی.) برای نمونه، اگر بخواهیم پس از بهنگام‌سازی یک ستون از یک جدول یک اِ-نامه[۱۶] (ایمیل) بفرستیم، در صورت استفاده از رهانا و همچنین در دسترس نبودن خدمتگزار اِ-نامه، تراکنش ناموفق خواهد بود و دستور اصلی نیز اجرا نخواهد شد (چنانکه می‌دانیم، رهانا درون تراکنشی اجرا می‌شود که دستور مربوطه اجرا می‌شود.) اینگونه موارد را بهتر است در سطح برنامه‌ی کاربردی کنترل کنیم و یا انجام دهیم.
- در مواردی که کارآیی سیستم به شدت افت می‌کند. (؟!)

۱۲-۵۷:

کلید اصلی می‌تواند متشکل از چند ستون باشد، حال آنکه کلید کلید جایگزین یا ساختگی نوعاً تک‌ستونی است. بنابراین، سربار ساخت نمایه و بررسی یکتایی مقدار آن (در درج سطر) نسبت به کلید اصلی می‌تواند کمتر باشد. ضمن اینکه مقدار دهی کلید جایگزین معمولاً با خود سیستم است و نه کاربرد.

[۱۶] E-Mail

اما کلید جایگزین، برخلاف کلید اصلی، فاقد معناست. از این رو، ارتباط‌هایی معنایی که قابل نمایش با کلید اصلی هستند را **لزوماً** نمی‌توان با کلید جایگزین نمایش داد.

۱۲-۵۸:

این دو گزینه در دستورSET CONSTRAINT، «حالت یک محدودیت»[17] و بنابراین زمان بررسی آن محدودیت در تراکنش جاری را مشخص می‌کنند. در واقع، اگر حالت محدودیت IMMEDIATE باشد، محدودیت در انتهای هر عبارت SQL بررسی می‌شود. اما اگر حالت محدودیت DEFFERED باشد، محدودیت در انتهای تراکنش بررسی می‌شود. گفتنی است هر محدودیت به صورت پیش‌نهاده یک حالت دارد که در تعریف محدودیت می‌توان آن را مشخص کرد. همچنین، یک محدودیت می‌توان تعویق‌پذیر باشد یا نباشد که آن نیز در تعریف محدودیت مشخص می‌شود. برای نمونه، در تعریف محدودیت در سیستم PostgreSQL سه گزینه وجود دارد:

- DEFERABLE INITIALLY DEFERED
- DEFERABLE INITIALLY IMMEDIATE
- NOT DEFERABLE

گزینه‌های اول و دوم، محدودیت‌های تعویق‌پذیرند که حالت پیش‌نهاده‌ی آن‌ها نیز مشخص شده است. گزینه‌ی سوم نیز محدودیت تعویق‌ناپذیر را تعریف می‌کند. بدیهی است حالت محدودیت‌های دسته‌ی سوم همواره IMMEDIATE است و دستور SET CONSTRAINT نیز امکان به تعویق انداختن این گونه محدودیت‌ها را ندارد.

۱۲-۶۰:

عنوان درس با معدل بیشینه در نیمسال دوم سال تحصیلی ۸۸-۸۹

۱۲-۶۱:

با فرض وجود جداول STT، COT، و STCOT، یک روش پاسخ به این پرسش در SQL به صورت زیر است:

```
SELECT STID, STNAME
FROM STT
WHERE STID NOT IN
(SELECT STID
       FROM (STT JOIN STCOT) JOIN COT
       WHERE TR='2' AND YRYR='88-89'
       GROUP BY STID
       HAVING SUM(CREDIT*GRADE)/SUM(CREDIT)
< ANY (SELECT SUM(CREDIT*GRADE)/SUM(CREDIT)
                  FROM STT JOIN STCOT JOIN COT
                  WHERE TR='2' AND YRYR='88-89'
```

[17] Constraint Mode

```
GROUP BY STID))
```

۶۲-۱۲:

اولاً: نام کارمندانی را بدهید که حقوق آن‌ها از حقوق تمام مدیران بیشتر باشد.

ثانیاً: بسته به برخود سیستم با NULL دارد:

- اگر حاصل ارزیابی عدد $\geq NULL$ **نادرست** باشد، آنگاه حداقل نام تمام کارمندان غیرمدیر و دارای حقوق مشخص برگردانده می‌شود. اگر نه، نام هیچ‌یک از اینگونه کارمندان برگردانده نمی‌شود.
- اگر حاصل ارزیابی $NULL \geq NULL$ **نادرست** باشد، آنگاه حداقل نام تمام کارمندان دارای حقوق با مقدار $NULL$، از جمله تمام مدیران، برگردانده می‌شود. اگر نه، نام هیچ‌یک از اینگونه کارمندان برگردانده نمی‌شود.

ثالثاً: بسته به برخود سیستم با NULL دارد:

- اگر حاصل ارزیابی عدد $\geq NULL$ **درست** باشد، آنگاه نام هیچ کارمند غیرمدیری برگردانده نمی‌شود. اگر نه، بستگی به حقوق دیگر مدیران دارد. در این حالت اگر حقوق کارمندی بیشتر از حقوق تمام مدیران دارای حقوق مشخص باشد، نام او برگردانده می‌شود.
- اگر حاصل ارزیابی هر دو عبارت $NULL \geq NULL$ و $NULL \geq$ عدد **نادرست** باشد آنگاه نام تمام مدیران با حقوق نامشخص (هیچمقدار) هم برگردانده می‌شود، اگر نه، نام هیچ مدیری برگردانده نمی‌شود.

رابعاً: بسته به برخورد سیستم در مقایسه‌ی NULL با عدد و یا NULL، پاسخ دلخواه از پرسش مطرح شده در بخش اولاً لزوماً برگردانده **نمی‌شود**. برای نمونه هیچگاه انتظار دیدن نام یک مدیر در پاسخ این پرسش وجود ندارد، اما چنانکه می‌بینیم ممکن است در شرایطی نام یک مدیر نیز برگردانده شود. یا در حالتی که حقوق یک کارمند از تمام مدیران بیشتر باشد به جز یک مدیر با مقدار حقوق NULL، ممکن است نام کارمند برگردانده نشود، حال آنکه ممکن است انتظار ما این باشد که نام کارمند در این حالت برگردانده شود.

۶۳-۱۲:

الف : نام تمام کارمندان و تمام مدیران ایشان (مدیران مستقیم و غیرمستقیم)

ب : تمام زوج‌های (مبدأ، مقصد) که می‌توان به نحوی از مبدأ به مقصد رسید (با یک یا چند پرواز مستقیم یا غیرمستقیم)

۶۴-۱۲:

کلید بدیل را می‌توان با به کمک کلازهای NOT NULL و UNIQUE به سیستم معرفی کرد. (همانند پاسخ سؤال ۶۵)

۱۲-۶۵:

با اِعمال محدودیت‌های NOT NULL و UNIQUE روی یک ستون می‌توان این‌کار را انجام داد[۱۸]. اگر کلید اصلی متشکل از چند ستون باشد، غیر از اِعمال محدودیت NOT NULL روی تک‌تک آن‌ها، محدودیت UNIQUE را با چنین دستوری می‌توان روی چند ستون اِعمال کرد:

```
ALTER TABLE tableName
ADD CONSTRAINT uq_tableName UNIQUE(col1, col2);
```

۱۲-۶۶:

الف : تمام دانشجویان دپارتمان D11 باید درس C11 را گرفته باشند.

ب : تمام دانشجویان رشته‌ی «کامپیوتر» باید درس C12 را گرفته باشند.

[۱۸] باید دقت کرد در صورتی که ستون از پیش وجود داشته باشد، نباید دارای مقدار تکراری و یا هیچمقدار باشد.

# گفتار ۱۳

## کنجکاوی‌ها

۱-۱۳ : به راهنمای SQL استانده مراجعه شود .

۲-۱۳ : به پاسخ تمرین ۱۹ از همین گفتار مراجعه شود .

۳-۱۳ : معمولاً خیر ، مگر اینکه تناظر بین رابطه‌های مبنا و فایل‌ها چند به یک باشد .

۴-۱۳ : باید دید مدلسازی معنایی چگونه انجام شده‌است . اگر به جای نوع ارتباط « حذف » در مدلسازی ، مفهوم « حذف درس » را تنها یک صفت از نوع ارتباط « انتخاب درس » بگیریم ، در این صورت می‌توان آن صفت را در عنوان رابطه‌ی STCOT وارد کرد .

کنجکاوی : آیا این طرز مدلسازی و طراحی خوب است ؟

۵-۱۳ :

- مورد ۳ را نمی‌توان عیب مهمی برای مفهوم دید دانست ، زیرا ماهّیت مفهوم دید محدودیت ساختاری را ایجاب می‌کند . اگر قرار باشد مثلاً کاربری تمامی یک یا چند رابطه‌ی مبنا را ببیند ، مدیر پایگاه داده‌ها می‌تواند چنین اجازه‌ای را به او بدهد .
- در SQL استانده ، عملیات بازیابی از دید هم در بعض انواع دید محدودیت دارد که می‌تواند یک عیب برای دید باشد ( نه از دیدگاه تئوریک ) .

۶-۱۳ : به پاسخ تمرین ۶ از همین گفتار مراجعه شود .

۷-۱۳ : به پاسخ تمرین ۶ از همین گفتار مراجعه شود .

۸-۱۳ : از دیدگاه عملی باید رفتار سیستم مورد استفاده را بررسی کرد . اما از دیدگاه تئوریک به نظر می‌رسد نظرخواهی از کاربر یک راه برای تصمیم‌گیری باشد . راه دیگر این است که سیستم محدودیت معنایی هر یک از دو رابطه‌ی $R_1$ و $R_2$ را بشناسد . برای مثال اگر $R_1$ حاوی تاپل‌های دانشجویان رشته‌ی کامپیوتر باشد و $R_2$ حاوی تاپل‌های دانشجویان رشته‌ی « تکنولوژی اطلاعات » ، در این صورت سیستم با بررسی مقدار صفت رشته‌ی تحصیلی در تاپل درج‌شدنی ، می‌تواند تصمیم بگیرد . توجه داشته باشیم که $R_1$ و $R_2$ در عمل اجتماع باید نوع-سازگار باشند .

کنجکاوی : آیا راه دیگری متصور است ؟

۹-۱۳ : می‌توان از گزینه‌ی DISTINCT یا UNIQUE استفاده کرد ، اما این کار کمکی به حل مشکل این دید در عملیات ذخیره‌سازی نمی‌کند .

۱۰-۱۳ : هر قاعده‌ای ذاتاً نوعی محدودیت است . پاسخ سؤال مثبت است .

۱۱-۱۳ : اگر به تعارض معنایی بین این دید و رابطه‌های مبنایی زیرین توجه نکنیم ، عملیات ذخیره‌سازی بطور « مکانیکی » قابل انجام است ، که البته بی‌معنا است .

# تمرینات درون گفتار

۱-۱۳ : باید بنویسیم : $V_3 = \pi_{<STNAME,\ TY,\ YRYR>} \sigma_{GRADE<10\ AND\ COID="com111">}$

چون عملگر پرتو تکراری‌ها را حذف می‌کند و ممکن است حداقل دو دانشجو نام یکسان داشته باشند ، معلوم نخواهد بود که یک نام مشخص مربوط به کدام دانشجو است .

۲-۱۳ :

مثال :

```
UPDATE     U1STV
     SET        STDEPT = ‘D003’
     WHERE STNUM = ‘80011122’ ;
```

این دستور تبدیل می‌شود :

```
UPDATE STT
     SET        STDEID = ‘D003’
     WHERE STID = ‘80011122’ ;
```

و سپس تبدیل می‌شود به :

```
UPDATE     STT2
     SET        STDEID = ‘D003’
     WHERE STID = ‘80011122’ ;
```

- مثال :

```
DELETE     FROM    U1STV
     WHERE  STNUM = ‘80011122’ ;
```

این دستور تبدیل می‌شود :

```
DELETE     FROM    STT
```

```
      WHERE STID = ‘80011122’ ;
```

و سپس تبدیل می‌شود به :

```
DELETE      FROM     STT1
      WHERE STID = ‘80011122’ ;
DELETE      FROM     STT2
      WHERE STID = ‘80011122’ ;
```

۱۳-۴ :

```
CREATE      VIEW  V41(SNAM)
      AS    SELECT      STNAME
            FROM        V4 ;

SELECT      DISTINCT    SNAM
            FROM  V41 ;
```

تبدیل می‌شود :

```
SELECT      DISTINCT    STNAME
            FROM        V4 ;
```

و سپس :

```
SELECT      DISTINCT    STNAME
            FROM        STT , STCOT
                  WHERE STT.STID = STCOT.STID ;
```

۱۳-۵ :

```
CREATE      VIEW  SCT ( SN , TERM , AV )
      AS    SELECT      STCOT.STID , STCOT.TR , AVG ( GRADE )
            FROM  STCOT
            GROUP BY ( STID , TR ) ;
SELECT      SCT.*
            FROM  SCT ;
```

تبدیل می‌شود به :

```
SELECT      STCOT.STID , STCOT.TR , AVG ( GRADE )
            FROM        STCOT
            GROUP BY ( STID , TR ) ;
SELECT      SN , MAX ( AV )
            FROM        SCT
            WHERE TERM = 2 ;
```

کنجکاوی : آیا این دستور اجرا می‌شود؟

کنجکاوی : در DBMS مورد استفاده‌تان رفتار سیستم با دیدهای آماری چگونه است؟

۹-۱۳ :

$R_1$ ( A, B ) ,  $R_2$ ( B, C )

AJPB = $\pi_{<A,\ C>}\ (R_1 \bowtie R_2)$

عملیات :

- درج :

```
INSERT   INTO   AJPB
     VALUES   <a1, c>;
```

تبدیل E/C ناممکن است زیرا رابطه‌های $R_1$ و $R_2$ ، تمام-کلید هستند.

- حذف :

```
DELETE   FROM   AJPB
     WHERE   B = 'b';
```

تبدیل E/C :

```
DELETE FROM R1
     WHERE   B = 'b';
DELETE FROM R2
     WHERE   B = 'b';
```

- بهنگام سازی :

```
UPDATE   AJPB
     SET   B = 'd'
     WHERE   A = 'a2';
```

تبدیل E/C :

```
UPDATE   R1
     SET   B = 'd'
     WHERE   A = 'a2';
```

کنجکاوی : آیا اجرای این عملیات عارضه‌ی جانبی دارد؟

۱۱-۱۳ : رابطه‌های زیر را در نظر می‌گیریم :

DEPT (DEID, DTIT, CITY, DPHON)

شماره اداره  عنوان اداره  شهر اداره  شماره تلفن

PROJ (PJID, PJTIT, DEID, CITY)

شهر محل اجرا اداره مجری عنوان پروژه شماره پروژه

- دید جاصل پیوند NK-NK :

DPV = DEPT JOIN PROJ

این دید مشخصات اداره‌ها و پروژه‌های «همشهر» را می‌دهد.

فرض می‌کنیم بدنه دو رابطه در یک لحظه چنین باشد :

| PROJ | PJID, | ..., | ..., | CITY |
|---|---|---|---|---|
| | j1 | | | c2 |
| | j2 | | | c1 |
| | j3 | | | c3 |
| | j4 | | | c4 |
| | j5 | | | c1 |

| DEPT | DEID, | ..., | CITY, | ... |
|---|---|---|---|---|
| | d1 | | c1 | |
| | d2 | | c2 | |
| | d3 | | c1 | |
| | d4 | | c4 | |

بدنه‌ی دید DPV :

| DPV | DEID, | ..., | CITY, | ... | PJID | ... |
|---|---|---|---|---|---|---|
| | d1 | | c1 | | j2 | |
| | d1 | | c1 | | j5 | |
| | d2 | | c2 | | j1 | |
| | d3 | | c1 | | j2 | |
| | d3 | | c1 | | j5 | |
| | d4 | | c4 | | j4 | |

- عوارض جانبی در عملیات :
  - در درج :

```
INSERT   INTO   DPV
     VALUES (d6 ... c1 ... j6 ...);
```

تبدیل E/C :

```
INSERT   INTO   DEPT
     VALUES (d6 ... c1 ...);
INSERT   INTO   PROJ
     VALUES (j6 ... c1);
```

حال اگر درخواست کنیم :

```
SELECT *
     FROM   DPV;
```

تاپل‌های زیر هم در دید ظاهر می‌شوند :

<d6, . . . , c1, . . . , j2, . . . >

<d6, . . . , c1, . . . , j5, . . . >

که کاربر درخواست نکرده است که در دیدش درج شود.

  - در حذف :

```
DELETE   FROM   DPV
     WHERE   DEID = ‘d1’  AND  PJID = ‘j2’;
```

تبدیل E/C :

```
DELETE   FROM   DEPT
     WHERE   DEID = ‘d1’;
DELETE   FROM   PROJ
     WHERE   PJID = ‘j2’;
```

حال اگر درخواست کنیم :

```
SELECT *
     FROM DPV;
```

تاپل‌های دیگری هم از دید کاربر حذف می‌شوند که مورد نظر کاربر نبوده است :

<d1, . . . , c1, . . . , j5, . . . >

<d3, . . . , c1, . . . , j2, . . . >

- در بهنگام سازی :

```
UPDATE   DPV
      SET   CITY = ‘c8’
      WHERE   DEID = ‘d3’ AND PJID = ‘j2’;
```

تبدیل E/C :

```
UPDATE   DEPT
      SET   CITY = ‘c8’
      WHERE   DEID = ‘d3’;
UPDATE   PROJ
      SET   CITY = ‘c8’
      WHERE   PJID = ‘j2’;
```

حال اگر درخواست کنیم :

```
SELECT *
      FROM DPV;
```

تاپل زیر از دید کاربر ناپدید می‌شود :

<d3, . . . , c1, . . . , j5, . . . >

نتیجه اینکه کاربر دارای این گونه دیدها باید از امکان بروز عوارض جانبی در عملیات ذخیره‌سازی آگاه باشد.

# تمرینات پایان گفتار

۱۳-۶ :

۱. دید حاصل پیوند CK-CK :

فرض می‌کنیم رابطه‌ی STT به دو رابطه‌ی زیر تجزیه عمودی شده باشد :

$STT1\ (\underline{STID}, STNAME, STDEID)$
$STT2\ (\underline{STID}, STLEV, STMJR)$

دید زیر را در نظر می‌گیریم :

$STT = STT1 \bowtie STT2$

چون این پیوند از نوع CK-CK است ، بنابراین بین تاپل‌های STT و تاپل‌های STT1 و STT2 تناظر یک به یک تاپلی وجود دارد .

- عمل درج : یک تاپل در STT1 و یک تاپل در STT2 درج می‌شود . به شرط آنکه تاپل کامل را بخواهیم از STT در پایگاه داده درج کنیم . اگر تاپل کامل نباشد و صفاتی که مقدارشان داده نشده ، محدودیّت هیچمقدار ناپذیری نداشته باشند ، دو تاپل ناقص ، هریک در یکی از دو رابطه‌ی مبنای زیرین ( STT1 , STT2 ) درج می‌شوند .
- عمل حذف : تاپل حذف شدنی از هر دو رابطه مبنای زیرین حذف می‌شود .
- عمل بهنگام سازی : بسته به اینکه چه صفاتی را بخواهیم بهنگام درآوریم ، عمل ممکن در یکی از دو رابطه‌ی مبنای زیرین ، یا در هردو ، انجام شود .

۲. دید حاصل پیوند CK-FK :

مثال زیر را در نظر می‌گیریم :

$$MYV = STT \bowtie STCOT$$

باتوجّه به اینکه کلید خارجی در رابطه‌ی STCOT یکتایی مقدار ندارد ، تناظر یک به یک تاپلی بین تاپل‌های MYV و رابطه‌های STT و STCOT برقرار نیست .

- عمل درج : ممکن است یک تاپل در STT درج شود ( و ممکن است هم نشود ) و در STCOT هم چنین است .

توجه : فرض بر این است که تاپل با کلید تکراری در MYV درج نمی‌کنیم .

به مثال زیر دقت شود :

| $STT$ | $STID$ | ... | ... |
|---|---|---|---|
| | $s_1$ | ... | ... |
| | $s_2$ | ... | ... |
| | $s_3$ | ... | ... |
| | $s_4$ | ... | ... |

| $STCOT$ | $STID$ | $COID$ | $TR$ | $YRYR$ | $GRADE$ |
|---|---|---|---|---|---|
| | $s_1$ | $c_1$ | ... | ... | ... |
| | $s_1$ | $c_2$ | ... | ... | ... |
| | $s_2$ | $c_1$ | ... | ... | ... |
| | $s_3$ | $c_3$ | ... | ... | ... |
| | $s_3$ | $c_4$ | ... | ... | ... |

```
INSERT INTO MYV
     TUPLE < s5 , … , c7 , … > ;
```

تبدیل E/C چنین است :

```
INSERT INTO STT
```

```
      TUPLE < S4 , … > ;
INSERT INTO STCOT
      TUPLE < c7 , … > ;
```

اگر بخواهیم تاپل <… , c5 , … , s1> را در MYV درج کنیم ، در تبدیل E/C خواهیم داشت :

```
INSERT INTO STCOT
      TUPLE < c5 , … > ;
```

اگر بخواهیم تاپل <… , c2 , … , s6> را در MYV درج کنیم ، در تبدیل E/C خواهیم داشت :

```
INSERT INTO STT
      TUPLE < s6 , … > ;
```

کنجکاوی : در چه وضعی ممکن است درخواست درج در MYV رد شود؟

- عمل حذف : یک تاپل از رابطه‌ی STT و یک تاپل از رابطه‌ی STCOT حذف می‌شوند اما با حذف این تاپل‌ها ممکن است تاپل‌هایی از دید MYV حذف شوند که مورد نظر نبوده‌است .

مثال :

| *MYV* | *STID* | … | *COID* | *TR* | *YRYR* | *GRA□E* |
|---|---|---|---|---|---|---|
| | $s_1$ | … | $c_1$ | … | … | … |
| | $s_1$ | … | $c_2$ | … | … | … |
| | $s_2$ | … | $c_1$ | … | … | … |
| | $s_3$ | … | $c_3$ | … | … | … |
| | $s_3$ | … | $c_4$ | … | … | … |

```
DELETE FROM MYV
      WHERE STID = ‘s1’ AND COID = ’c1’;
```

در تبدیل E/C خواهیم داشت :

```
DELETE FROM STT
      WHERE STID = ‘s1’ ;
DELETE FROM STCOT
      WHERE STID = ‘s1’ AND COID = ‘c1’ ;
```

با حذف تاپل <…, s1> از رابطه‌ی STT ، دو تاپل از MYV حذف می‌شوند ، تاپل , c1 , … , s1> <… که موردنظر بود و حذف <… , c2 , … , s1> که موردنظر نبود .

بدیهی است کاربر دارای دید MyV ، باید از امکان بروز این عارضه‌ی جانبی آگاه باشد .

- عمل بهنگام سازی :

بسته به اینکه چه صفت یا صفاتی را بخواهیم بهنگام درآوریم ، عمل بهنگام سازی در MYV ممکن است تبدیل به عمل بهنگام سازی در رابطه‌ی STT و/یا STCOT شود .

ذکر مثالی در این باره را به خواننده وا می‌گذاریم .

کنجکاوی : آیا ممکن است عارضه‌ی جانبی بروز کند؟

۳. دید حاصل پیوند FK-FK :

در این نوع دید ، در حالت کلّی تناظر یک‌به‌یک بین تاپل‌های دید و تاپل‌های رابطه‌های زیرین وجود ندارد .

مثال : $STCODEP = STT \bowtie_{STT.STDEID = COT.CODEID} COT$

این دید حاصل پیوند روی کلید خارجی STDED ( شماره‌ی گروه آموزشی دانشجو ) از رابطه‌ی STT و کلید خارجی CODEID ( شماره‌ی گروه آموزشی درس ) از رابطه‌ی COT ، است .

$STT$

| $\underline{STID}$ | ... | $STDEID$ |
|---|---|---|
| $s_1$ | ... | $d_1$ |
| $s_2$ | ... | $d_1$ |
| $s_3$ | ... | $d_3$ |
| $s_4$ | ... | $d_2$ |

$COT$

| $\underline{COID}$ | ... | $CODEID$ |
|---|---|---|
| $c_1$ | ... | $d_1$ |
| $c_2$ | ... | $d_2$ |
| $c_3$ | ... | $d_1$ |
| $c_4$ | ... | $d_4$ |
| $c_5$ | ... | $d_5$ |

$STCODEP$

| $STID$ | ... | $STDEID$ | $COID$ | ... | $CODEID$ |
|---|---|---|---|---|---|
| $s_1$ | ... | $d_1$ | $c_1$ | ... | $d_1$ |
| $s_1$ | ... | $d_1$ | $c_3$ | ... | $d_1$ |
| $s_2$ | ... | $d_1$ | $c_1$ | ... | $d_1$ |
| $s_2$ | ... | $d_1$ | $c_3$ | ... | $d_1$ |
| $s_4$ | ... | $d_2$ | $c_2$ | ... | $d_2$ |

عدم وجود تناظر یک‌به‌یک در این مثال به خوبی دیده می‌شود .

- عمل درج :

ممکن است تاپلی در رابطه‌ی STT درج شود یا درج نشود . در مورد رابطه‌ی COT هم چنین است .

مثال :

```
INSERT INTO STCODEP
      TUPLE <s5 , … , d1 , c1 , … ,d1> ;
```

در تبدیل E/C :

```
INSERT INTO STT
      TUPLE <s5 , … , d1> ;
```

مثال :

```
INSERT INTO STCODEP
      TUPLE <s6 , … , d6 , c6 , … ,d6> ;
```

در تبدیل E/C :

```
INSERT INTO STT
      TUPLE <s6 , … , d6> ;
INSERT INTO COT
      TUPLE <c6 , … , d6> ;
```

- عمل حذف : منجر می‌شود به حذف یک تاپل از هردو رابطه STT و COT و البته عارضه جانبی هم دارد . ذکر مثال را به خواننده وا می‌گذاریم .
- عمل بهنگام سازی : منجر می‌شود به بهنگام سازی در یکی از دو رابطه یا در هردو رابطه .

مثال :

```
UPDATE STCODEP
      SET STMJR = ‘Phys’
      WHERE STID = ‘s2’ ;
```

در تبدیل E/C :

```
UPDATE STT
      SET STMJR = ‘Phys’
      WHERE STID = ‘s2’ ;
```

ذکر مثال‌های دیگر را به خواننده وا می‌گذاریم .

کنجکاوی : اگر صفت STDEID یا CODEID را در دید STCODEP بهنگام درآوریم ، چه پیش می‌آید؟

۱۳ – ۷ : دید زیر را در نظر می‌گیریم :

$$STCOV = STT \bowtie STCOT$$

این دید حاصل برون پیوند چپ دو رابطه است ( از نوع CK-FK ) .

با توجّه به بدنه‌ی دو رابطه در تمرین ۶ داریم :

$STCOV$

| $STID$ | ... | $COID$ | $TR$ | ... |
|---|---|---|---|---|
| $s_1$ | ... | $c_1$ | ... | ... |
| $s_1$ | ... | $c_2$ | ... | ... |
| $s_2$ | ... | $c_1$ | ... | ... |
| $s_3$ | ... | $c_3$ | ... | ... |
| $s_3$ | ... | $c_4$ | ... | ... |
| $s_4$ | ... | ? | ? | ... |

صرف‌نظر از اینکه STCOV از دیدگاه مدل رابطه‌ای ، رابطه نیست ، عملیات درج ، حذف و بهنگام‌سازی از این دید در پایگاه داده ، کم و بیش به همان طرز که در تمرین ۶ قسمت دید حاصل پیوند CK-FK گفته شد ، انجام می‌شوند .

کنجکاوی : آیا تفاوتی بین عملیات از این دید و دید پیوند CK-FK ، دیده شده در تمرین ۶ ، وجود دارد؟

۱۳ – ۸ :

اولاً :

```
CREATE VIEW SOFTHARD ( STID , STDEG , COID , GRADE ) AS
      SELECT STID , STLEV , COID , GRADE
      FROM STT , STCOT
      WHERE STMJR = ‘Software’ OR STMJR = ‘Hardware’
```

ثانیاً : این دید یک دید گزیشی-پرتوی و در عین حال پیوندی CK-FK است . نکته مهم در این دید این است که در فهرست صفات دیده ، صفت داده شده در شرط تعریف دید یعنی STMJT وجود ندارد . این وضع در عمل درج مشکل خاصی ایجاد می‌کند که در ادامه می‌بینیم .

- عمل درج : چگونگی تبدیل E/C همانست که در تمرین ۶ دیدیم . اما دو مشکل بروز می‌کند . یک مشکل این است که برای صفات خارج از محدوده‌ی دید ، هیچمقدار پدید می‌آید یا اینکه سیستم مقدار پیش‌نهاده را می‌گذارد . بنابر اگر این صفت ( صفاتی ) محدودیت هیچمقدار ناپذیری داشته باشند ، درخواست درج رد

می‌شود . مشکل مهم‌تر این است که چون صفت STMJR در محدوده‌ی دید نیست ، اگر بخواهیم پس از درج تاپل در SOFTHARD ، آن‌را بازیابی کنیم ، تاپل درج شده در رابطه‌ی جواب وجود ندارد ، زیرا دید دارای مسند گزینش با شرط رشته‌ی نرم‌افزار یا سخت‌افزار است و سیستم در صفت STMJR در رابطه‌ی STT یا هیچمقدار می‌گذارد یا مقدار پیش‌نهاده . برای حل مشکل باید یک رهانا نوشت ( مثال ۲۸ از گفتار ۱۲ دیده شود ) .

- عمل حذف : به همان طرز است که در تمرین ۶ گفته شد .
- عمل بهنگام‌سازی : به همان طرز است که در تمرین ۶ گفته شد .

کنجکاوی : آیا ممکن است درخواست بهنگام‌سازی رد شود؟

ثالثاً : چون دید SOFTHARD از جمله روی رابطه‌ی مبنای STT تعریف می‌شود ، با حذف این رابطه ، دید تعریف شده روی آن نامعتبر می‌شود و در نتیجه استقلال داده‌ای منطقی تأمین نخواهد بود . باید پیش از حذف رابطه‌ی STT ، آن را به صورت یک رابطه‌ی مجازی ( دید ) روی دو رابطه‌ی STT1 و STT2 تعریف کرد ( قسمت ۳-۱-۱ از گفتار ۱۲ دیده شود ) .

۱۳ – ۱۴ : قسمت ۳ از گفتار دهم دیده شود .

۱۳ – ۱۵ : قسمت ۴-۴-۶ از گفتار شانزدهم دیده شود .

۱۳ – ۱۷ : اگر روی رابطه‌ی R دیدی تعریف نشده باشد ، مشکلی برای استقلال داده‌ای منطقی بروز نمی‌کند . امّا اگر مثلاً دید V روی رابطه‌ی R تعریف شده باشد ، باید که پیش از حذف R ، این رابطه را به صورت یک رابطه‌ی مجازی به صورت $R = R_1\ UNION\ R_2$ تعریف کرد تا تعریف V روی رابطه‌ی R معتبر بماند . بدیهی است عمل روی V اینک ابتدا تبدیل به عمل روی دید R می‌شود و سپس به عمل روی R1 یا R2 ( به فرض $R_1\ INTERSECT\ R_2$ دقت شود ) . قسمت ۴-۲-۱-۵ از گفتار سیزدهم دیده شود .

۱۳ – ۱۸ : مانند مثال ۲۸ از گفتار دوازدهم انجام می‌شود .

۱۳ – ۱۹ : این دید ، تمام رابطه‌ی R است . حال اگر طرّاح رابطه‌ی R را با افزودن صفت ( صفاتی ) گسترش دهد ، در بازیابی از دید V این صفات «دیده» نمی‌شوند . مگر اینکه دید V را دوباره تعریف کنیم ( البته پس از حذف دید قبلی ) . نتیجه اینکه تعریف دید حالت پویا ندارد ( حداقل در سیستم‌های موجود ) ، یعنی گسترش رابطه‌ی زیرین بطور پویا روی دید اِعمال نمی‌شود . البته از دیدگاه تئوریک این کار متصور است .

۱۳ – ۲۱ : داریم :

$$V = \pi_{<STID,STNAME,STMJR>}\ (\ STT \ltimes STCOT\ )$$

این دید پرتوی است از رابطه‌ی حاصل از نیم‌پیوند رابطه‌های STT و STCOT از نوع CK-FK . تاپل‌های V ، تاپل‌های پیوندشدنی از رابطه‌ی STT است ، البته دارای سه صفت ( و نه تمام صفات ) این رابطه .

- عمل درج : درج تاپل در V تبدیل می‌شود به درج تاپلی ناقص در STT به شرط آنکه تاپل پیوند شدنی با این تاپل در رابطه‌ی STCOT وجود داشته باشد . در غیر این‌صورت در بازیابی از V ، این تاپل درج شده «دیده» نمی‌شود ، زیرا با هیچ تاپلی از STCOT پیوند نمی‌شود . ضمناً برای صفاتSTLEV و STDEID در رابطه‌ی STT یا هیچمقدار درج می‌شود یا مقدار پیش‌نهاده . چنانچه این دو صفت محدودیت هیچمقدار ناپذیری داشته باشند ، درخواست درج تاپل در V رد می‌شود .
- عمل حذف : منجر می‌شود به حذف یک تاپل از رابطه‌ی STT .

کنجکاوی : آیا این حذف عارضه‌ی جانبی دارد؟

- عمل بهنگام سازی :

درخواست بهنگام سازی صفات STNAME و/یا STMJR تبدیل می‌شود به بهنگام سازی همین صفات در رابطه‌ی STT و مشکلی ندارد .

کنجکاوی : درخواست بهنگام سازی صفت STID چطور؟

۱۳ – ۲۲ : داریم : $S(S\#, SNAME, STATUS, CITY)$

دید V1 شامل شماره و نام تهیه‌کنندگانی است که مقدار وضعیت آنها بیشینه نباشد . مثلاً اگر بدنه‌ی رابطه‌ی S چنین باشد :

$S$

| $S\#$ | $SNAME$ | $STATUS$ | $CITY$ |
|---|---|---|---|
| $s_1$ | $sn_1$ | 10 | … |
| $s_2$ | $sn_2$ | 15 | … |
| $s_3$ | $sn_3$ | 17 | … |
| $s_4$ | $sn_4$ | 8 | … |
| $s_5$ | $sn_5$ | 17 | … |

در بدنه‌ی V1 داریم :

| $V_1$ | $S\#$ | $SNAME$ |
|---|---|---|
| | $s_1$ | $sn_1$ |
| | $s_2$ | $sn_2$ |
| | $s_4$ | $sn_4$ |

اولاً : دید V1 روی یک رابطه‌ی مبنا تعریف شده و حافظ کلید است . پس دیدی است پذیرا ، اما ممکن است درخواست درج یا بهنگام سازی رد شود .

- عمل درج : منجر می‌شود به درج یک تاپل در رابطه‌ی S . اما اگر صفات STATUS و/یا CITY هیچمقدار ناپذیر باشند ، درخواست درج رد می‌شود .

توجّه : اگر صفت STATUS محدودیت هیچمقدار ناپذیری نداشته باشد ، تاپل درج شدنی در V1 ، در رابطه S درج می‌شود اما در بازیابی از V1 ، «دیده» نخواهد شد .

کنجکاوی : چرا؟

- عمل بهنگام سازی : منجر می‌شود به بهنگام سازی صفت SNAME و/یا صفت #S در S ( البته در عمل ممکن است درخواست بهنگام سازی صفت #S رد شود ) .

ثانیاً :

```
CREATE VIEW V1 ( S# , SNAME ) AS
      SELECT S.S# , S.SNAME
      FROM  S
      WHERE STATUS < ( SELECT MAX ( STATUS ) FROM S ) ;
```

بررسی پذیرا یا ناپذیرا بودن این دید در SQL را به خواننده وامی‌گذاریم ( این دید را روی سیستمی که می‌شناسید ، تعریف و وضع آن‌را بررسی کنید ) .

۱۳ – ۲۳ : این دید ، دیدی است گزینشی-پرتوی روی رابطه‌ی حاصل از پیوند سه رابطه ، از نوع CK-FK-CK .

تبدیل E/C در عملیات از این دید روی رابطه‌های زیرین STT ، COT و STCOT به همان طرزی است که در تمرین ۶ دیدیم ( البته در اینجا به جای دو رابطه‌ی مبنای زیرین ، سه رابطه‌ی مبنای زیرین داریم ) .

۱۳ – ۲۴ : در متن گفتار مثال مشابه وجود دارد .

۱۳ – ۲۵ : کاربردهای تکنیک دید روی دید را یادآوری می‌کنیم :

- تأمین استقلال داده‌ای منطقی در قبال سازماندهی مجدد سطح ادراکی ( تغییر طرّاحی منطقی )
- تقویت بیشتر ایمنی ( امنیت ) پایگاه داده ( نکته‌ی ۲ از متن گفتار دیده شود ) .

# گفتار ۱۴

## کنجکاوی‌ها

۱۴-۱ : برخی خصوصیات دیگر طراحی خوب :

- حتی‌الامکان با کمترین تعداد رابطه‌ها انجام شود .
- حتی‌الامکان رابطه‌ها کمترین اشتراک معنا را داشته باشند .
- درجه‌ی رابطه‌ها بزرگ نباشد .
- حتی‌الامکان هر صفت استقلال معنایی داشته باشد ( به بیان دیگر ، حتی الامکان صفات مرکب نباشند ) .
- صفات فاقد معنا و ساختگی نداشته باشد .

۱۴-۲ : هر یک از سه طرز مزیت و عیب خود را دارد :

- طراحی ۱ :
  - درجه‌ی رابطه کوچکتر ( نسبت به طراحی ۲ )
  - یافتن مشخصات کامل دانشجوی همپروژه به زمان بیشتری نیاز دارد ( نسبت به طراحی ۲ ) .
- طراحی ۲ :
  - درجه‌ی رابطه : بزرگتر ( نسبت به طراحی ۱ و ۳ )
  - یافتن مشخصات کامل جفت دانشجوی همپروژه : سریعتر ( نسبت به طراحی ۱ و ۳ )
  - اگر فقط شماره‌ی همپروژه‌ها مورد نظر باشد : کارایی کمتر (نسبت به طراحی ۱ و ۳ )
  - کاردینالیتی رابطه نصف کاردینالیتی رابطه‌ها در طراحی ۱ و ۳
- طراحی ۳ :
  - درجه‌ی رابطه‌ی اصلی : کمتر

- بازیابی جفت شماره‌ی همپروژه‌ها : سریعتر
- بدست آوردن مشخصات کامل همپروژه‌ها نیاز به عمل پیوند دارد .

نتیجه اینکه در انتخاب یکی از این سه طراحی باید بویژه به نیاز داده‌ای کاربران توجه داشت .

- طرز طراحی دیگری وجود ندارد .

۱۴-۳ : طرز طراحی دیگری وجود ندارد ، مگر در حالت خیلی خاص ، یعنی وقتی که نوع موجودیت قوی تنها صفت شناسه داشته باشد . در این صورت می‌توان با یک رابطه طراحی کرد . البته اگر تعداد صفات نوع موجودیت قوی خیلی کم ( در حد چند صفت ) باشد و بسامد دستیابی به اطلاعات مربوط به « اثر منشرشده » بالا باشد و تعداد نمونه‌های نوع موجودیت قوی زیاد نباشد ، می‌توان با پذیرش میزانی افزونگی ، طراحی را با یک رابطه انجام داد .

۱۴-۴ : به پاسخ تمرین ۳۳ از همین گفتار مراجعه شود .

۱۴-۵ : از دیدگاه تئوریک پاسخ مثبت است ، اما در عمل باید بویژه به بسامد مراجعه به نوع موجودیت کارمند – دانشجو توجه داشت . چنانچه این بسامد بالا باشد ، طراحی یک رابطه‌ی جدا بهتر است . همچنین اگر زیرنوع از خود صفات دیگری هم داشته باشد ، بهتر است یک رابطه‌ی جدا طراحی شود .

کنجکاوی : اگر رابطه‌ی جدا طراحی نشود ، چگونه باید زیرنوع موجودیت را نمایش داد ؟

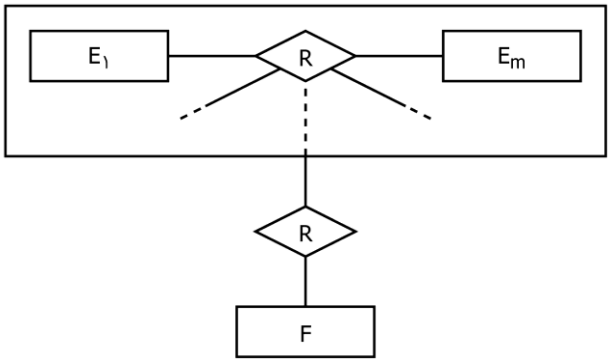
۱۴-۶ : اگر بسامد مراجعه به صفت NO-OF-CHAP بالا باشد ، بهتر است این صفت را برای نوع موجودیت کتاب در نظر بگیریم و در عنوان رابطه‌ی BOOK وارد کنیم .

کنجکاوی : اگر این صفت در نظر گرفته نشود ، چگونه باید به درخواست کاربر در این مورد پاسخ داد ؟

۱۴-۷ : در حالت خاص می‌تواند $1 : 1$ یا $M : N$ هم باشد . اگر چندی $1 : 1$ باشد ، شناسه‌ی نوع موجودیت جزء می‌تواند کلید باشد .

۱۴-۸ : بله ، اگر نخواهیم همه‌ی رابطه‌ها نرمال باشند .

۹-۱۴ : معمولاً درجه‌ی ارتباط با ارتباط ، دو است :




درجه‌ی $R_1$ : $n$

درجه‌ی $R_2$ : $2$

تعداد رابطه‌های لازم : $n + 1 + 1 + 1 = n + 3$

حال اگر درجه‌ی ارتباط $R_2$ ( حداقل از دیدگاه تئوریک ) $n' > 2$ باشد ، در این صورت ، تعداد رابطه‌های لازم برابر است با : $n + 1 + n' - 1 + 1 = n + n' + 1$

( توجه : در متن کنجکاوی آمده است : $n + n' + 2$ که اشتباه چاپی است ) .

۱۰-۱۴ : به پاسخ تمرین ۳۶ از همین گفتار مراجعه شود .

۱۱-۱۴ : به پاسخ تمرین ۵۷ از همین گفتار مراجعه شود .

۱۲-۱۴ : در موارد زیر :

- در طراحی ارتباط " IS-A " رابطه‌های نمایشگر زبرنوع و زیرنوع ( ها ) کلید کاندید یکسان دارند .
- وقتی که رابطه‌ای را تجزیه‌ی افقی می‌کنیم ، رابطه‌های حاصله کلید کاندید یکسان دارند به شرطی که در مسند گزینش صفت دیگری ، غیر از کلید کاندید رابطه‌ی اصلی نباشد .
- در تجزیه‌ی عمودی رابطه‌ی R به رابطه‌های $R_1$ ، $R_2$ و ... $R_n$ به نحوی : $H_{R_1} \cap H_{R_2} \cap ... \cap H_{R_n} = CK_R$
- وقتی که یک صفت یا ترکیبی کاهش‌ناپذیر از صفات در مجموعه عنوان مثلاً $R_1$ و $R_2$ باشند و کلید کاندید دوم در هر دو رابطه باشند . مثلاً :

EMPL ( EID , … . , ENC , … . )
CK

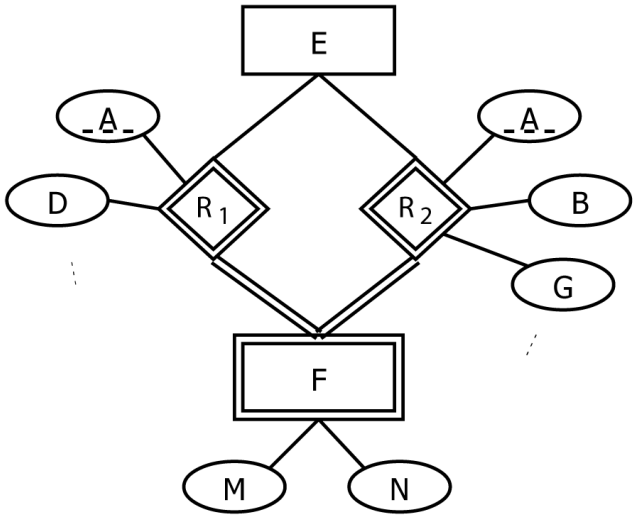
STUD ( STID , … , ENC , … )

البته اگر در این حالت ، مدلسازی با ارتباط $IS - A'$ را انجام ندهیم .

- وقتی که بین دو نوع موجودیت E و F بیش از یک نوع ارتباط با چندی $1:1$ یا $N:1$ داشته باشیم و برای هر نوع ارتباط ، رابطه‌ی جدا طراحی کنیم .

- وقتی که یک نوع موجودیت بیش از یک صفت چندمقداری داشته باشد و برای هر صفت چندمقداری یک رابطه‌ی جدا با استفاده از تکنیک در نظر گرفتن صفات به تعداد حداکثر مقدار صفت چندمقداری ، طراحی کنیم .

- وقتی که نوع موجودیت ضعیف F با نوع موجودیت قوی E بیش از یک نوع ارتباط شناسا داشته باشد ، هر یک با صفات خاص خود ، اما صفت ممیزه نوع ضعیف برای هر دو نوع ارتباط یکسان باشد به صورت زیر :



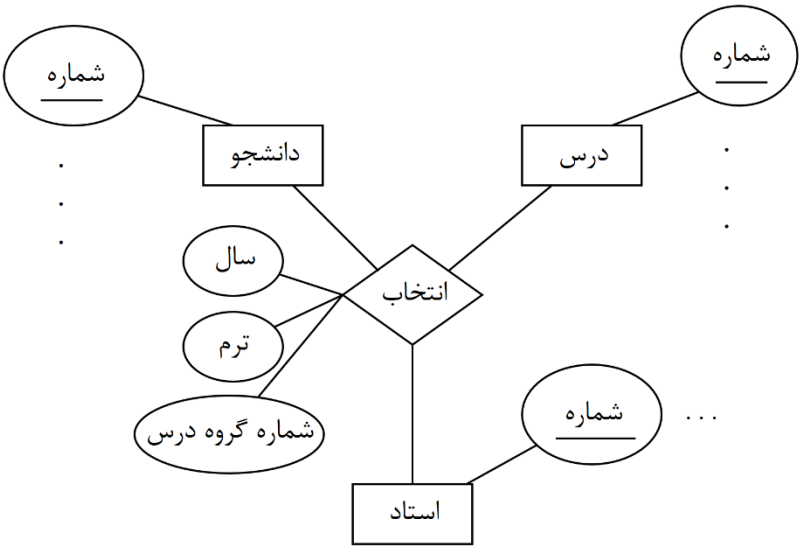

این مدل حداقل از لحاظ تئوریک متصور است .

کنجکاوی : آیا وضع دیگری هم وجود دارد ؟

۱۴-۱۳ : پاسخ به خواننده واگذار می‌شود .

# تمرینات درون گفتار

۱۴-۱ : یک مدلسازی ممکن چنین است :




طراحی :

$STUD(\underline{STID}\ , \ldots)$

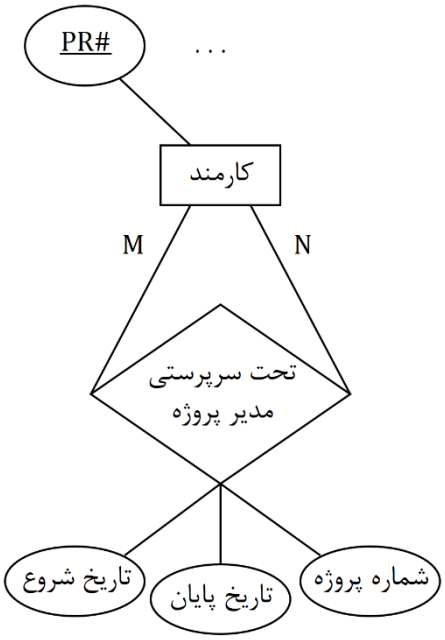
$COUR(\underline{COID}\ , \ldots)$

$PROF(\underline{PRID}\ , \ldots)$

$SCP(STID, COID\ , PRID, TR\ , YRYR\ , SECNO)$

کلید کاندید این رابطه ، ترکیب $(STID\ , COID\ , TR\ , YRYR)$ است .

۳-۱۴ : مدلسازی چنین است :



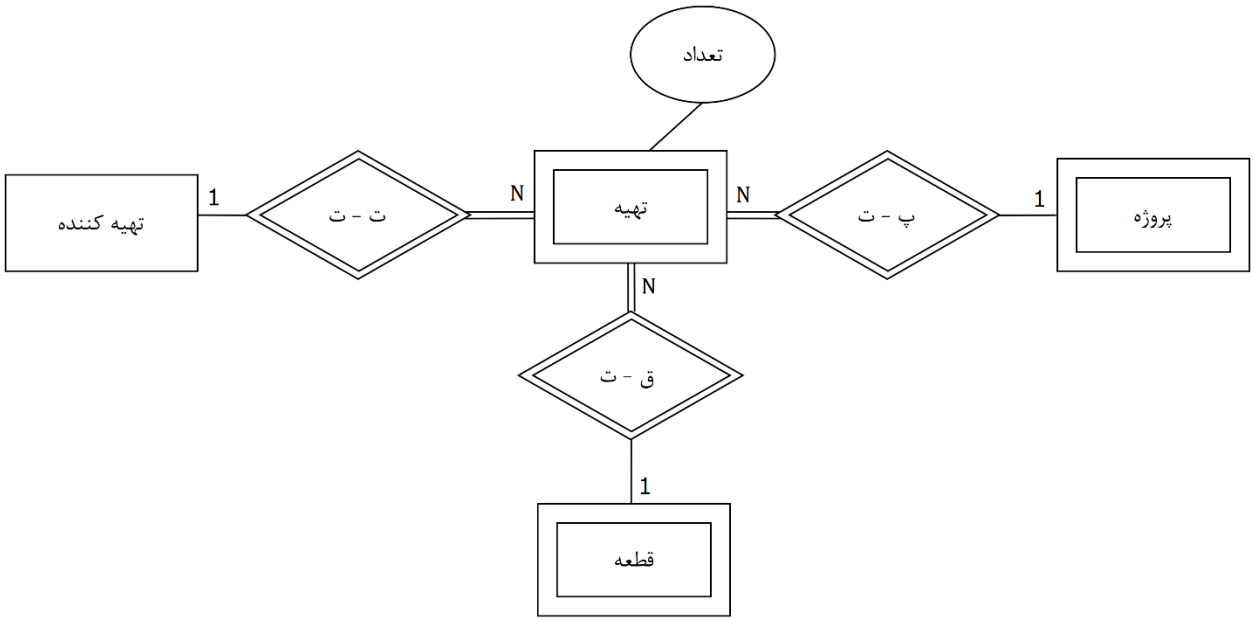

توجه : چندی نوع ارتباط تغییر می‌کند (مگر اینکه محدودیت خاصی وجود داشته باشد) .

طراحی : $PROF(PR\#\ ,\ldots\ldots)$

$PRJMGR(PR\#\ ,PRMGRID\ ,PJID\ ,BDATE\ ,FDATE\ )$

کنجکاوی : کلید کاندید این رابطه چیست؟

۴-۱۴ :




یک طراحی ممکن چنین است :

$S(\underline{S\#}, \dots\dots)$

$J(\underline{J\#}, \dots\dots)$

$P(\underline{P\#}, \dots\dots)$

$SUPPLY(\underline{S\#}, \underline{QTY}, \underline{P\#}, \underline{J\#})$

طراحی دیگر :

$SSUP(\underline{S\#}, QTY)$

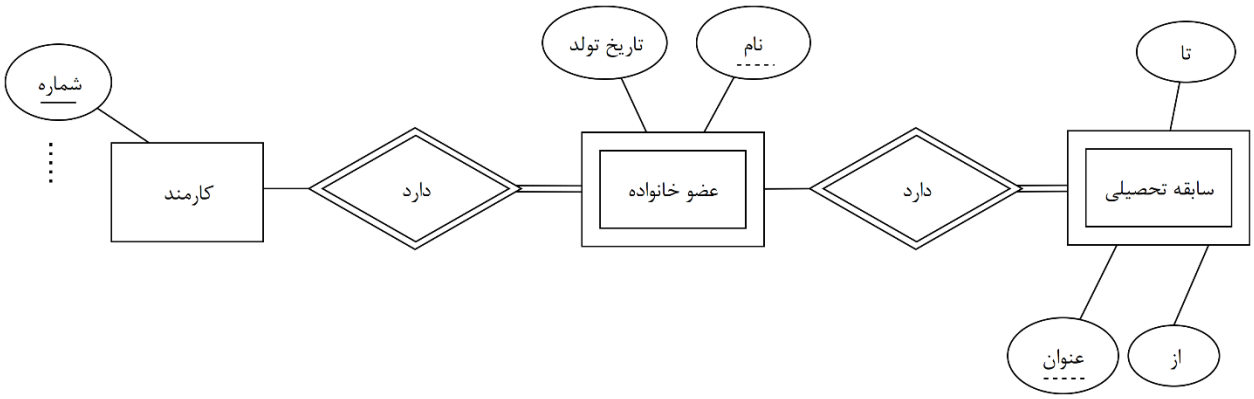
$JSUP(\underline{J\#}, QTY)$

$PSUP(\underline{P\#}, QTY)$

کنجکاوی : آیا این طرز طراحی توجیه شدنی است؟ آیا در پیوند این سه رابطه $(SSUP \bowtie JSUP \bowtie PSUP)$ ممکن است مشکلی بروز کند ؟

۱۴-۵ : طرز طراحی همان است که در حالت هفتم گفته شد .

مثال :




طراحی :

$EMPL(\underline{EMID}\ , \ldots\ldots)$

$EMEM(\underline{EMID}\ , MNAME\ , BDATE)$

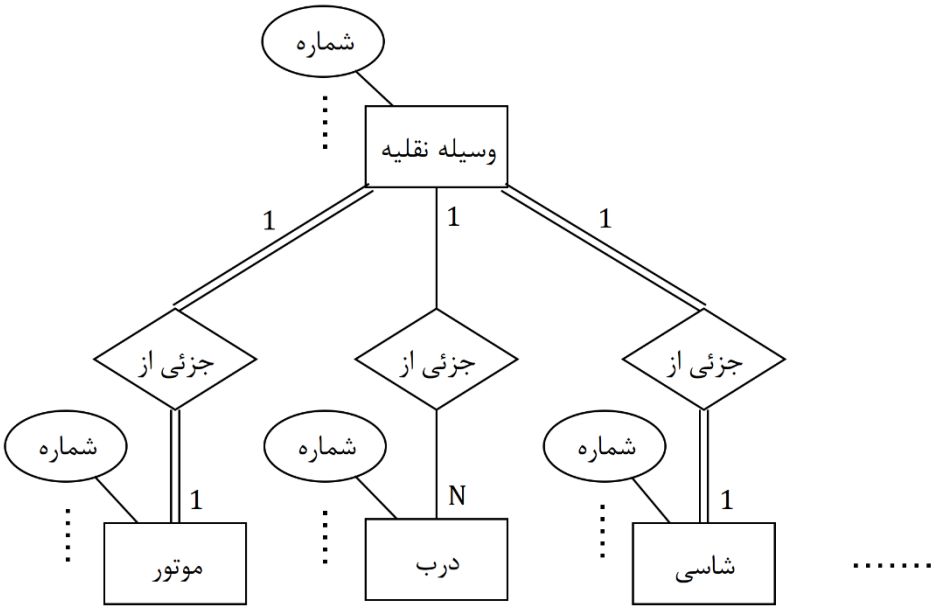
$EDUCHIS(\underline{EMID}\ , MNAME\ , TITLE\ , FTOM\ , TO)$

نتیجه : هرچه تعداد سطوح نوع ارتباط شناسا بیشتر باشد ، کاردینالیتی کلید کاندید رابطه‌ها بیشتر ، تعداد کلید خارجی بیشتر و بعلاوه کلید(های) خارجی مرکب هم وجود خواهند داشت .

۱۴-۶ : به پاسخ تمرین ۳۳ از پایان همین گفتار مراجعه شود .

۱۴-۷ :

مثال :




طراحی :

$VEIC(\underline{VID}\ , \dots\dots)$

$VEIC(\underline{VID}\ , \underline{MOID})$

$DOOR(\underline{VID}\ , \underline{DOID})$

$SHAS(\underline{VID}\ , \underline{SHID})$

ذکر مثال دیگر به خواننده واگذار می‌شود .

# تمرینات پایان گفتار

در این قسمت ، حلّ برخی از تمرین‌ها را می‌آوریم . تمرین‌های مشابه به‌طور مشابه حل می‌شوند .

۱۴-۱ : در پیوستار ۱ کتاب مفاهیم بنیادی پایگاه‌داده‌ها نمودار کامل‌تر محیط آموزشی ارائه و طرّاحی منطقی آن انجام شده است .

۱۴-۲ : مثال ۲۵ از گفتار سوّم :

رابطه‌ها :

$EMP\ (\underline{ENO}, \ldots)$

$KAR\ (\underline{ENO, JOB, FROM}, TO)$

$MOR\ (\underline{ENO, TYPE, FROM}, TO)$

$MAM\ (\underline{ENO}, LOC, \underline{SUB, FROM}, TO)$

$TASH\ (\underline{ENO, DATE}, TYPE, CAUSE)$

$AMO\ (\underline{ENO, TIT, FROM}, TO)$

کنجکاوی : آیا کلید کاندید رابطه‌های طرّاحی شده می‌تواند جز آنهایی باشد که مشخص شده اند؟

۱۴-۳ : مثال‌هایی از گفتار سوم :

مثال ۲۸ :

مثال ۲۹ :

$DEPT\ (\underline{DEID}, \ldots)$

$PROF\ (\underline{PRID}, \ldots, DEID)$

$COUR\ (\underline{COID}, \ldots, PRID)$

$FACU\ (\underline{FAID}, \ldots)$

$PROF\ (\underline{PROF}, \ldots, FAID)$

~~$DEPT\ (\underline{DEID}, \ldots)$~~

$DEPT\ (\underline{DEID}, \ldots, FAID)$

~~$PROF\ (\underline{PRID}, \ldots, DEID)$~~

پس از رفع دام پیوندی :

$COUR\ (\underline{COID}, \ldots, PRID, DEID)$

مثال ۳۰ :

$COMP\ (\underline{COM\#}, \ldots)$

$BORD\ (\underline{COM\#, BORD\#}, \ldots)$

$MANIT\ (\underline{COM\#, MANIT}, \ldots)$

$MAIN\ (\underline{COM\#, MAIN\#}, \ldots)$

$DISK\ (\underline{COM\#, DISK\#}, \ldots)$

مثال ۳۱ : طراحی با تکنیک عمومی :

$STUD\ (\underline{STID}, FNAME, LNAME, \ldots)$

$DOCT\ (\underline{STID}, A, \ldots)$

$UNGR\ (\underline{STID}, X, \ldots)$

$GRDA\ (\underline{STID}, B, \ldots)$

توجه : X و . . . و A و . . . و B و . . . ، به ترتیب صفات خاص نوع موجودیت‌های دانشجوی دوره‌ی دکترا ، دانشجوی دوره‌ی کارشناسی و دانشجوی دوره‌ی کارشناسی ارشد فرض شده‌اند .

مثال ۳۲ :

در هر دو حالت الف و ب ، تخصیص کامل است ، می‌توان از تکنیک دوم استفاده کرد :

$MEMP\ (\underline{EMP\#},\ \overbrace{\ldots\ldots\ldots}^{\text{صفات مشترک}},\ \overbrace{\ldots\ldots\ldots}^{\text{صفات خاص کارمند مرد}})$

$FEMP\ (\underline{EMP\#},\ \overbrace{\ldots\ldots\ldots}^{\text{صفات مشترک}},\ \overbrace{\ldots\ldots\ldots}^{\text{صفات خاص کارمند زن}})$

۴-۱۴ : به حلّ تمرین ۵ مراجعه شود .

۵-۱۴ :

$PROJECT\ (\underline{PJ\#}, PJNAME, PJBUDGET\ , PJMGR\#)$

$EMPLOYEE\ (\underline{PJ\#, E\#}\ , ENAME, ESALARY)$

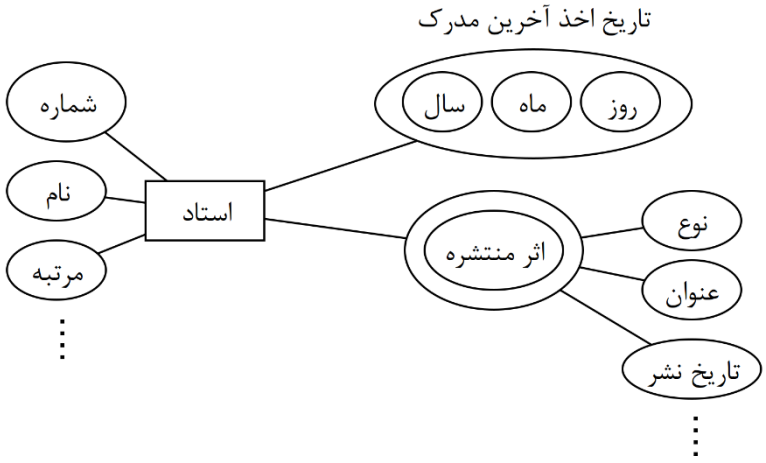
$SKILL\ (\underline{PJ\#, E\#}\ , SKTITLE, SKLEVEL)$

$SPEND\ (\underline{PJ\#, SPD\#}, DATE, AMOUNT)$

۶-۱۴ :

$PROF\ (\underline{PRID}, NAME, RANK, YYYY, MM, DD)$

$PRPUB\ (\underline{PRID, PUBTIT}, DATE, TYPE)$




نتیجه اینکه در مدل رابطه‌ای بهتر است صفت مرکّب به صورت تعدادی صفت ساده نمایش داده شود .

کنجکاوی : چرا؟

۷-۱۴ :

- کارخانه : $KAR$ ( KID, ... )
- ماشین : $MAC$ ( MID, ... )
- سرپرست + سرپرستی : $MGR$ (MGID, ... , KID)
- کارگر : $WOR$ (WID, ... , KID, MID )
- همکاری : $COL$ ( WID, WWID, ... )

کنجکاوی : آیا با تعداد کمتر یا تعداد بیشتر رابطه می‌توان طراحی کرد؟

۸-۱۴ : می‌توان از تمرین ۷ ایده گرفت .

۱۰-۱۴ :

- کتاب + تصحیح + ویرایش : $BOOK$ ( BK#, ... , MEM#, MMEM#)
- عضو : $MEMB$ ( MEM#, ... )
- ترجمه : $TARJ$ (BK#, MEM#, ... )
- تألیف : $TALIF$ (BK#, MEM#, ... )

کنجکاوی : در چه وضعی طرّاحی با بیش از چهار رابطه قابل توجیه است؟

۱۱-۱۴ :

- مجله + وابسته بودن : $MAJ$ (TIT, NUM, ... , ATIT )
- انجمن علمی : $ANJ$ (ATIT, ... )
- عضو تحریریه : $OZV$ (TIT, NAM, FROM, TO, ADR )
- مقاله + انتشار : $PAP$ (CODE, MTIT, ABS, FNUM, PNUM )
- واژه کلیدی : $PWAJ$ ($CODE$, $WAJ$)
- مؤلف مقاله : $PAUT$ (CODE, AUT )
- منابع مقاله : $PMAN$ (CODE, RES )
- ترجمه بودن : $PTAR$ (CODE, MOT, CCODE )
- مرجع بودن : $PREF$ (CODE, RCODE )

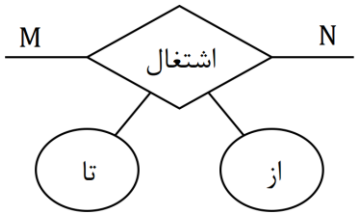
توجّه : CCODE و RCODE نام‌های دیگر صفت CODE هستند .

کنجکاوی : در چه وضعی با کمتر از این تعداد هم می‌توان طرّاحی را انجام داد به نحوی که قابل توجیه باشد؟

۱۲-۱۴ : ابتدا با استفاده از ارتباط "IS-A" مدلسازی و سپس طراحی می‌کنیم .

۱۳-۱۴ :

- اداره + مشاوره + مدیریت : $DEPT$( D#, ... , DMGR#, E#)
- کارمند + اشتغال : $EMPL$ ( E# ... D#)
- همکاری : $COL$ ( ... V)




کنجکاوی : آیا می‌توان نوع ارتباط مشاوره را به طرز دیگر نمایش داد؟

۱۴-۱۴ : باید برای توع ارتباط‌های اشتغال و مدیریت صفات زمانی در نظر گرفت . در این صورت چندی نوع ارتباط‌ها تغییر می‌کند . چندی ۱ : ۱ به چندی 1 : N یا M : N تبدیل می‌شود . چندی 1 : N به چندی M : N تبدیل می‌شود ( البته با توجه به قواعد معنایی محیط ) . بدیهی است که طرّاحی هم تغییر می‌کند .

۱۵-۱۴ :

1. $AIRPORT\ (\underline{AIRPORTCODE}, NAME, CITY, STATE)$
2. $AIRPLANETYPE\ (\underline{TYPENAME}, max-seats)$
3. $CANLAND\ (\underline{AIRPORTCODE, AIRPLANETYPE})$
4. $AIRPLANE\ (\underline{AIRPLANEID}, Total-no-of-seats, TYPENAME)$
5. $FLIGHTLEG\ (\underline{AIRPORTCODE, LegNo}, scheduled\ AriTime, scheduled\ DepTime, NUMBER)$
6. $FLIGHT\ (\underline{NUMBER}, \ldots\ldots\ldots, Airline)$
7. $FARE\ (\underline{NUMBER, CODE}, AMOUNT)$
8. $LEGINSTANCE-ASSIGNED(\underline{AIRPLANEID, Date}, No-of-avail-seats)$
9. $LEGINSTANCE-ARRIVE-DEPARTS\ (\underline{AIRPORTCODE, Date}, DepTime, AriTime)$
10. $LEGINSTANCE-OF\ (\underline{AIRPORTCODE, LegNo, Date})$
11. $LEGINSTANCE-SEAT(\underline{AIRPLANEID, Date, SeatNo})$

توجه :

در مدلسازی انجام شده [ELMA03] ( و نیز به همان صورت در [ELMA11] ) ، نوع موجودیت LEGINSTANCE نوع ضعیف چهار نوع موجودیت قوی در نظر گرفته شده است . می‌توان رابطه شماره ۱۰ را به دو رابطه‌ی جدا تقسیم کرد .

۱۷-۱۴ :

اولاً : بین استاد و پژوهشگر ارتباط "IS-A" وجود دارد .

ثانیاً : نه ، زیرا این کار به این معناست که به جای دو رابطه ، یک رابطه طراحی کنیم . از نظر تبدیل نمودار ارتباط "IS-A" به طراحی رابطه‌ای ، چنین کاری نادرست نیست ، اما اگر درصد کمی از استادان پژوهشگر باشند ، حجم هیچمقدار در پایگاه داده افزایش می‌یابد . بعلاوه درجه‌ی رابطه بزرگ می‌شود و این وضع می‌تواند سبب اُفت کارایی در اجرای پرسش‌هایی شود که به تعداد کمی از صفات نیاز دارند . اما توجّه داشته باشیم که در طراحی با دو رابطه ،

چنانچه به اطلاعات کامل پژوهشگر(ها) نیاز باشد ، باید عمل پیوند انجام شود . پس اگر بسامد این درخواست بالا باشد ، باید با پذیرش میزانی هیچمقدار ، دو رابطه را در هم ادغام کرد .

ثالثاً : نوشتن شِما به خواننده واگذار می‌شود .

کنجکاوی : آیا صفت PRID در رابطه‌ی RESEARCHER به نوعی کلید خارجی هم هست؟

۱۸-۱۴ :

اولاً :

$STUD\ (\underline{STID}\ , \ldots)$

$COUR\ (\underline{COID}\ , \ldots)$

$STCO\ (\underline{STID, COID}, SECNO, \underline{PRID}, TR, YR)$

$PROF\ (\underline{PRID}\ , \ldots)$

ثانیاً :

$OFFERING\ (\underline{PRID}\ , \underline{STID, COID}, TR, YR)$

رابطه‌ی STCO هم تغییر می‌کند :

$STCO\ (\underline{STID, COID}\ , SECNO)$

کنجکاوی : آیا صفت «گروه درسی» باید در عنوان رابطه OFFERING وارد شود؟

۲۲-۱۴ :

- وسیله نقلیه : $MACH$ ( M#, ... )
- قطعه : $PART$ ( P#, ... )
- تشکیل شدن : $TASH$ (P#, PP#, Quantity)
- جزیی است از : $JOZE$ (M#, P#, Quantity )

$SAVARI$ ( M#, $\overbrace{\text{... ... ...}}^{\text{صفات سواری}}$ )

$VANET$ (M#, $\overbrace{\text{... ... ...}}^{\text{صفات وانت}}$)

$CAMION$ ( M#, $\overbrace{\text{... ... ...}}^{\text{صفات کامیون}}$ )

۲۵-۱۴ :

الف : ارتباط "IS-A" : فرض : E زبرنوع و F زیرنوع باشد .

باید که : $\{F - key\ values\} \subset \{E - key\ values\}$

بنابراین در عملیات درج ، حذف و بهنگام سازی باید این محدودیت ( محدودیت شمول ) رعایت شود .

ب : ارتباط "IS-A-PART-OF" : فرض : E : PART و F : SUB PART

در رابطه‌ی نمایشگر نوع موجودیت جزء ( SUB PART ) ، کلید کاندید رابطه‌ی نمایشگر نوع موجودیت کل ( PART ) ، کلید خارجی است ، پس قاعده $C_2$ طبق معمول باید رعایت شود .

پ : ارتباط سلسله مراتبی : در رابطه نمایشگر نوع رکورد فرزند ، کلید کاندید رابطه‌ی نمایشگر نوع رکورد پدر ، کلید خارجی است ، پس قاعده $C_2$ طبق معمول باید رعایت شود .

توجه : در حالت ب و پ هم کلید خارجی رابطه‌ی رجوع کننده با کلید کاندید رابطه‌ی مرجع ارتباط شمول دارد .

کنجکاوی : آیا در حالت الف ، مفهوم کلید خارجی در معنای متعارف مطرح است؟

۳۰-۱۴ :

$STUD\ (\underline{STID}, SNAME, \ldots, \underline{\ldots\ldots OID})$

$EMPL\ (\underline{EMID}, ENAME, COID, \ldots, \underline{\ldots\ldots OID})$

$OWNER\ (\underline{OID}, ZAMEN, MOAREF, \ldots)$

$ACCOUNT\ (\underline{ACNO}, TYPE, ODATE, \ldots, \underline{\ldots\ldots OID})$

۳۳-۱۴ :

روش ۱ : طرّاحی با دو رابطه :

$PROF\ (\underline{PRID}, \ldots)$

$PRPUB\ (\underline{PRID}, PTIT, TYPE, DATE)$

روش ۲ : طرّاحی با یک رابطه ، با فرض معلوم بودن حداکثر تعداد مقداری که صفت چند مقداری می‌گیرد :

$PROFPUB\ (\underline{PRID}, \ldots PUB1, PUB2, \ldots, PUBi, TYPE1, TYPE2, \ldots, TYPEi, DATE1, DATE2, \ldots, DATEi)$

روش ۳ : طرّاحی با یک رابطه ، به شرط اینکه هر استاد حداقل یک اثر منتشره داشته باشد :

$PROFPUB\ (\underline{PRID}, \ldots\ldots, \underline{PTIT}, TYPE, DATE)$

کنجکاوی : آیا در رابطه PROFPUBL ، ممکن است صفات TYPE و/یا DATE هم جزء کلید بشوند؟

بحث :

روش ۱ : روش عمومی است ، محدودیّت خاصی ( که در روش ۲ و ۳ مطرح است ) ندارد . اگر بسامد مراجعه به استاد و اثر منتشره ، بطور مستقل از یکدیگر بالا باشد ، روش ۱ مناسب‌تر است ، بویژه اگر تعداد صفات استاد زیاد باشد .

روش ۲ : وقتی مناسب است که :

- تعداد صفات استاد زیاد نباشد .
- حداکثر تعداد مقداری که صفت چندمقداری می‌گیرد ، عدد کوچک باشد .

- صفت چندمقداری ترجیحاً ساده باشد .
- درصد بالایی از استادان به تعداد نزدیک به حداکثر تعداد مقدار ، اثر منتشره داشته باشند .
- بسامد مراجعه به اطلاعات استاد و اثر منتشره با هم بالا باشد .

روش ۳ : وقتی مناسب است که :

- تعداد صفات استاد زیاد نباشد .
- هر نمونه استاد حداقل یک اثر منتشره داشته باشد .
- حداکثر تعداد مقداری که صفت چندمقداری می‌گیرد ، کم باشد . درصد بالایی از استادان به تعداد نزدیک به حداقل تعداد مقداری که صفت چندمقداری می‌گیرد ، اثر منتشره داشته باشند .

کنجکاوی : آیا برای هریک از سه روش ، شرایط دیگری هم مطرح است؟

۱۴-۳۵ :

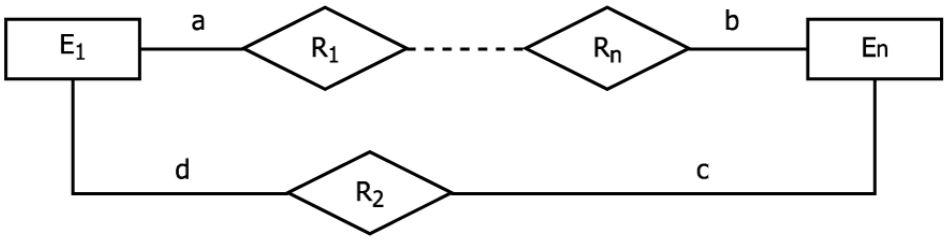
۱. اگر کلید خارجی جزء تشکیل دهنده‌ی کلید اصلی ( و نه هر کلید کاندید ) باشد .
۲. اگر کلید خارجی نمایشگر نوع ارتباط با چندی 1 : N باشد با محدودیت مشارکت الزامی نوع موجودیت سمت N در نوع ارتباط .
۳. اگر کلید خارجی نمایشگر نوع ارتباط 1 : 1 باشد و مشارکت یکی از دو نوع موجودیت شرکت کننده در نوع ارتباط ، الزامی باشد .

۱۴-۳۶ : در حالات زیر :

۱. در رابطه‌ی نمایشگر نوع ارتباط 1 : N
۲. در رابطه‌ی نمایشگر نوع ارتباط 1 : 1
۳. در نمایش زبرنوع وقتی که زیرنوع U-type دارد و شناسه‌ی زبرنوع‌ها از میدان‌های متفاوت باشد .

کنجکاوی : آیا حالت دیگری هم وجود دارد؟

۱۴-۳۷ : برخی از حالات تشکیل چرخه‌ی ارجاع :




فرض : $n \geq 1$

- $d = N , c = 1 , b = N , a = 1$
- $d = 1 , c = 1 , b = 1 , a = 1$ و مشارکت الزامی $E_1$ در $R_1$ و $E_n$ در $R_n$.
- $d = 1 , c = 1 , b = N , a = 1$ و مشارکت الزامی $E_1$ در $R_1$ .
- $d = N , c = 1 , b = 1 , a = 1$ و مشارکت الزامی $E_n$ در $R_n$.
- $d = N , c = 1 , b = 1 , a = N$

کنجکاوی : حالات دیگر کدامند؟ آیا ممکن است در حالت ( حالاتی ) ، چندی M : N هم باشد؟

۳۸-۱۴ :

باید معنای هررابطه را مشخص کرد . فرض می‌کنیم N=2 باشد :

- اگر رابطه $R_1$ نمایشگر زبرنوع و $R_2$ نمایشگر زیرنوع باشد ، مفهوم کلید خارجی بطور ضمنی مطرح است و $\{R_2 - CK - values\} \subset \{R_1 - CK - values\}$
- اگر $R_1$ و $R_2$ حاصل تجزیه افقی دو رابطه باشند به نحوی که این دو رابطه تاپل(های) مشترک نداشته باشند ، مفهوم کلید خارجی مطرح نیست . اما اگر تاپل(های) مشترک داشته باشند ، باید در عملیات ذخیره‌سازی روی تاپل(های) مشترک ، سازگاری داده‌ها تأمین شود . نمی‌توان گفت که مفهوم کلید خارجی مطرح است .
- اگر $R_1$ و $R_2$ حاصل تجزیه عمودی رابطه R باشند به نحوی که $H_{R_1} \cap H_{R_2} = CK$ ، ممکن است مسئله تأمین سازگاری داده‌ها مطرح شود ، اما مفهوم کلید خارجی به طور صریح مطرح نیست .
- اگر رابطه‌های $R_1$ و $R_2$ با طرّاحی بااشتراک طراحی شده باشند ، چنانچه کلید کاندید در آن‌ها مشترک باشد ، باید با دقت در معنای دو رابطه و ارتباط معنایی آن‌ها با هم ، به سازگاری داده‌ها توجّه داشت ، ممکن است مفهوم کلید خارجی به طور ضمنی مطرح باشد .

۳۹-۱۴ :

الف :

- وقتی که کارمندی برنامه‌ساز شود یا دیگر سمت برنامه‌سازی نداشته باشد ، در هردو حالت باید یک عمل درج و یک عمل حذف در/از هر دو رابطه انجام شود .
- باید محدودیت : $EMP \quad JOIN \quad PROG = \emptyset$ را هم اِعلان کنیم .
- این محدودیت را به صورت $EMP \cap \left(\Pi_{EID}(PROG)\right) = \emptyset$ هم می‌توان اعلان کرد .
- اگر بخواهیم رابطه(های) دیگری به کارمندان ارجاع داشته باشد ، اعلان ارجاع به سادگی اعلان یک کلید خارجی نیست .

ب :

- در این حالت افزونگی پدید می‌آبد . در واقع اگر مثلاً E برنامه‌ساز است ، قطعاً کارمند هم هست ، پس دیگر نیازی به تصریح ندارد . البته این افزونگی از طراحی ناشی می‌شود . اگر تنها یک رابطه به صورت $EMPL\ (\underline{ENO}, LANG)$ طراحی کنیم و داده‌های همه کارمندان را در همین رابطه وارد کنیم ، مشکل افزونگی و نیز مشکلات قسمت الف بروز نمی‌کند ، در عوض «هیچمقدار» پدید می‌آید . توجه داریم که در این حالت نیازی به در نظر گرفتن صفت نوع نیست ، زیرا تخصیص تک شاخه است ( تنها یک نوع موجودیت زیرنوع در مدلسازی دیده شده است ) .

  کنجکاوی : اگر در عنوان رابطه EMP صفت JOB را هم اضافه کنیم ، چه پیش می‌آید؟

# گفتار ۱۵

## کنجکاوی‌ها

۱۵-۱ : با در نظر گرفتن رابطه‌ی $R\ (A_1, \dots, A_n)$ ، به دو روش می‌توان پاسخ داد :

روش ۱ ) برای اینکه تعداد وابستگی‌های تابعی نامهم را بدست آوریم ، ابتدا باید نحوه‌ی ساخته شدن چنین وابستگی‌هایی را شرح دهیم . برای ایجاد چنین وابستگی‌هایی ، کافی است که i صفت از n صفت عنوان رابطه‌ی R را انتخاب کرده ( $0 \leq i \leq n$ ) ، در سمت راست وابستگی تابعی قرار دهیم و سمت چپ وابستگی تابعی نیز از اجتماع همان i صفات انتخاب‌شده ( برای سمت راست ) و زیرمجموعه‌ای از صفات انتخاب‌نشده ( که تعدادش $n - i$ است ) بدست می‌آید ( تا با حضور صفات انتخاب‌شده برای سمت راست وابستگی در سمت چپ ، چنین وابستگی تابعی نامهم شود ) . به عنوان مثالی برای $i = 2$ ، می‌توان وابستگی‌های تابعی به فرم $T \cup (A_i, A_j) \rightarrow (A_i, A_j)$ را در نظر گرفت ( که در آن $i \neq j$ و $1 \leq i, j \leq n$ و $T \subseteq H_R - \{A_i, A_j\}$ است ) .

حال با توجه به توضیحات ، تعداد چنین وابستگی‌هایی به راحتی قابل بدست آوردن است . چون انتخاب i صفت از n صفت ( برای حضور در سمت راست و چپ وابستگی تابعی ) با $\binom{n}{i}$ روش ممکن است ( $0 \leq i \leq n$ ) و تعداد زیرمجموعه‌های یک مجموعه‌ی $n - i$ عضوی برابر $2^{n-i}$ است ، طبق اصل ضرب تعداد وابستگی‌های نامهم برابر است با : $\sum_{i=0}^{n} 2^{n-i} \binom{n}{i}$ .

حال باید اثبات کرد که این عبارت برابر است با $3^n$ . اثبات چنین ادعایی با توجه به تساوی دوجمله‌ای $(a + b)^n = \sum_{i=0}^{n} a^i b^{n-i} \binom{n}{i}$ و جایگذاری $a = 1$ و $b = 2$ حاصل می‌شود .

روش ۲ ) با توجه به روش ایجاد چنین وابستگی‌هایی در روش اول ، می‌توان گفت که برای تشکیل چنین وابستگی‌هایی برای هر صفت از عنوان رابطه ، سه حالت ممکن است :

a. صفت در وابستگی تابعی حضور ندارد .

b. صفت در وابستگی تابعی در هر دو سمت وابستگی حضور دارد .

c. صفت در وابستگی تابعی تنها در سمت چپ وابستگی حضور دارد .

پس با در نظر گرفتن اصل ضرب ، در کل تعداد حالات ممکن $3^n$ خواهد بود .

۱۵-۲ : در هر دو طرف وابستگی تابعی می‌توان هر زیر مجموعه‌ای را قرار داد ، یعنی :

$2^n \times 2^n = 2^{2n}$ ( در صورت کنجکاوی $2^{n+1}$ آمده است که اشتباه چاپی است ) .

<u>توجه</u> : در این محاسبه زیر مجموعه‌ی تهی هم منظور شده است . در غیر این صورت داریم :

$(2^n - 1) \times (2^n - 1) = (2^n - 1)^2$ .

۱۵-۳ : بله ، فرض می‌کنیم در رابطه‌ی $R(A, B, C, D)$ داریم :

$$F = \{A \rightarrow C, B \rightarrow D\}$$

در این صورت دو زیر مجموعه از F ، یعنی $\{B \rightarrow D\}$ و $\{A \rightarrow C\}$ داریم که صفت مشترک ندارند .

۱۵-۴ : آزمون وجود $F_c$ هزینه کمتری دارد ، زیرا دو شرط باید بررسی شود به جای سه شرط که در آزمون وجود $F$ لازم است .

۱۵-۵ : خیر ، خیر .

۱۵-۶ : برای آنکه بتوان با حداقل تعداد وابستگی‌های تابعی ، بهترین طراحی ممکن را انجام داد و تمام محدودیت‌های معنایی محیط را هم از این طریق اعمال کرد .

۱۵-۷ : تجزیه‌ی دیگر چنین است :

$$\begin{cases} STCO(\underline{STID, COID}) \\ PRCO(\underline{PRID, COID}) \end{cases}$$

<u>کنجکاوی</u> : آیا این تجزیه خوب است ؟

۱۵-۸ : به دلیل وجود دو وابستگی تابعی : $\begin{cases} (STID, COID) \rightarrow PRID \\ PRID \rightarrow COID \end{cases}$

که بیانگر دو قاعده‌ی معنایی در این محیط هستند .

۱۵-۹ : صِرف رعایت یکتایی مقادیر کلید کاندید ایجاب نمی‌کند که وابستگی تابعی بین دو صفت عمده ( هر یک جزء یک کلید کاندید متمایز از کلید کاندید دیگر ) رعایت شود .

<u>کنجکاوی</u> : آیا این رابطه باید به سه رابطه تجزیه شود ؟

۱۵-۱۰ : با اضافه کردن صفت TR ، قواعد معنایی محیط می‌تواند تغییر کند و چنین است هرگاه صفت YRYR را اضافه کنیم . در واقع در هر دو حالت کلیدهای کاندید رابطه می‌توانند تغییر کنند .

۱۵-۱۱ : ضوابط قضیه ریسانن در تجزیه‌ی رابطه‌ی $3NF$ به رابطه‌های $BCNF$ مطرح‌اند و اساساً در مورد تبدیل رابطه‌ی $BCNF$ به رابطه‌های $4NF$ اِعمال نمی‌شوند .

۱۵-۱۲ : اگر مجموعه‌ی تهی صفات را در نظر نگیریم ، پاسخ مثبت است ، وگرنه پاسخ منفی است زیرا در این صورت داریم :

$A \twoheadrightarrow B \mid \{\}$

$B \twoheadrightarrow A \mid \{\}$

$\{\} \twoheadrightarrow A \mid B$

۱۵-۱۳ : دلایل در اساس همان‌ها هستند که در متن برشمرده‌ایم ، اما می‌توان موارد زیر را هم اضافه کرد :

- طراحی با اشتراک معنا
- تجزیه عمودی رابطه به نحوی که در پرتوها غیر از کلید کاندید رابطه اصلی ، صفت ( صفات ) مشترک دیگری هم ، بنا به ملاحظاتی ، باشند .
- استفاده از تکنیک جدول جستجو ( به قسمت ۴-۴-۴ از گفتار ۱۶ مراجعه شود ) .
- استفاده از تکنیک رابطه ( های ) اضافی ( به قسمت ۴-۴-۵ مراجعه شود ) .

# تمرینات درون گفتار

۱۵-۲ :

وابستگی‌های تابعی رابطه‌ی R :

$$F = \{ A \to D, A \to C, (C\ , D) \to B, (A\ , D) \to C\}$$

$$\begin{cases} A \to D \\ (A\ , D) \to C \end{cases} \Rightarrow (A\ , A) \to C \Rightarrow A \to C$$

پس $A \rightarrow C$ افزونه است . وابستگی تابعی دیگر افزونه نیستند .

۱۵-۳ : در رابطه R با بدنه داده شده داریم :

$B \rightarrow C$

| R | (A, | B, | C, | D) |
|---|---|---|---|---|
| | a1 | b1 | c1 | d1 |
| | a2 | b1 | c1 | d2 |
| | a3 | b2 | c2 | d3 |
| | a4 | b3 | c3 | d4 |

بعد از اجرای GROUP BY داریم :

| R | (A, | B, | C, | D) |
|---|---|---|---|---|
| | a1 | b1 | c1 | d1 |
| | a2 | b1 | c1 | d2 |
| | a3 | b2 | c2 | d3 |
| | a4 | b3 | c3 | d4 |

می‌بینیم که در هر گره ، وجود ندارد مقداری برای B به نحوی که :

$MAX(C) \neq MIN(C)$

یعنی به ازای یک مقدار مشخص از B ، تنها یک مقدار C وجود دارد ، یعنی : $B \rightarrow C$

۱۵-۴ :

.۴

$$F_1 = \begin{Bmatrix} B \rightarrow D \\ E \rightarrow C \\ (A, C) \rightarrow D \\ (C\ , D) \rightarrow A \\ (B\ , E) \rightarrow E \end{Bmatrix}$$

$$\begin{cases} B \rightarrow D \\ E \rightarrow C \end{cases} \Rightarrow (B, E) \rightarrow (C\ , D)$$

$$\begin{cases} (B\ , E) \rightarrow (C\ , D) \\ (C\ , D) \rightarrow A \end{cases} \Rightarrow (B\ , E) \rightarrow A$$

پس $(B\ , E) \rightarrow A$ افزونه است . وابستگی‌های دیگر افزونه نیستند .

$$F_2 = \left\{ \begin{array}{c} A \to (C,D,E) \\ B \to (C,E) \\ (A,D) \to E \\ (C\ ,D) \to F \\ (B\ ,D) \to A \\ (C,E,D) \to (A,B,D) \end{array} \right\}$$

$$\begin{cases} A \to (C,D,E) \\ (C,D,E) \to (A,B,D) \end{cases} \Rightarrow A \to (A,B\ ,D)$$

$$A \to (A,B,D) \Longrightarrow \begin{cases} A \longrightarrow B \\ A \longrightarrow D \end{cases}$$

$$A \to (C,D,E) \Longrightarrow \begin{cases} A \longrightarrow C \\ A \longrightarrow D \\ A \longrightarrow E \end{cases} \Rightarrow (A,D) \longrightarrow E$$

پس $(A,D) \to E$ افزونه است .

$$B \to (C,E) \Longrightarrow \begin{cases} B \longrightarrow C \\ B \longrightarrow E \end{cases}$$

$$\begin{cases} A \to B \\ B \to C \end{cases} \Rightarrow A \to C$$

پس $A \to C$ افزونه است .

$$(C,D,E) \to (A,B,D) \Longrightarrow \begin{cases} (C,D,E) \longrightarrow A \\ (C,D,E) \longrightarrow B \\ (C,D,E) \longrightarrow D \end{cases}$$

$$\begin{cases} (C,D,E) \to A \\ A \to B \end{cases} \Rightarrow (C,D,E) \to B$$

پس $(C,D,E) \to B$ افزونه است .

$$\begin{cases} (C,D,E) \to A \\ A \to D \end{cases} \Rightarrow (C,D,E) \to D$$

پس $(C,D,E) \to D$ افزونه است .

$$\left\{\begin{array}{c} A \to B \\ A \to E \\ B \to C \\ B \to E \\ (C\ , D) \to F \\ (B, D) \to A \\ (C, D, E) \to A \end{array}\right\}$$

۱۵ – ۴ : به خواننده واگذار می‌شود .

۱۵ – ۵ : در رابطه $R(A, B, C, D)$ داریم :

$$A \to D, (C, D) \to B, (A, D) \to C$$

$$\begin{cases} A \to D \\ (A, D) \to C \end{cases} \Rightarrow A \to C$$

$$\begin{cases} A \to D \\ (C, D) \to B \end{cases} \Rightarrow (A, C) \to B$$

$$\begin{cases} A \to C \\ (A, C) \to B \end{cases} \Rightarrow A \to B$$

بنابراین : $A^+ = \{A, B, C, D\}$ پس A کلید کاندید است . این رابطه تنها یک کلید کاندید دارد که صفت A است ، پس (C,D) در کلید کاندید R وجود ندارد .

۱۵ – ۶ :

- مثال ۱ :

$R1(\underline{JID}, JBUD, JCITY, STATE)$

$JID \to JBUD$

داریم :

$JID \longrightarrow JCITY \longrightarrow STATE$

- مثال ۲ :

$R2(\underline{PRID}, \underline{PRNAME}, PRROOM, RPHON)$

$PRID \to PRNAME$

$PRID \longrightarrow PRROOM \longrightarrow RPHON$

۱۵ – ۷ :

- مثال ۱ :

داریم : $PROF(\underline{PRID}, PRNAME, PRLDIP, DIPUNIV)$

شماره استاد — نام — عنوان آخرین مدرک — دانشگاه محل اخذ

$PRID \rightarrow PRLDIP \rightarrow DIPUNIV$

رابطه‌ی PROF در 2NF است .

- مثال ۲ :

$PROJ(\underline{JID}, JTITLE, JCITY, JZIP)$

$JID \rightarrow JCITY \rightarrow JZIP$

رابطه‌ی PROJ در 2NF است .

۱۵ – ۹ :

$$\begin{cases} SD\ (\underline{STID}, STNAME, STDEID) \\ JD\ (\underline{STDEID, STMJR}) \end{cases}$$

این تجزیه خوب نیست ، زیرا شرایط قضیه‌ی ریسانن را ندارد .

۱۵ – ۱۱ : تجزیه خوب نیست ، زیرا حافظ وابستگی تابعی نیستند .

در اوّلی $(STID, COTITLE) \longrightarrow GRADE$ و در دوّمی $(STID, COID) \longrightarrow GRADE$ از دست می‌روند .

۱۵ – ۱۲ : کلیدهای کاندید تغییر می‌کنند :

$STCOGRRYR\ (\ \underline{COID, STID}\ , COTITLE\ , TR\ , YEAR\ , GRADE\ )$

CK CK

رابطه کماکان در 3NF است .

۱۵ – ۱۴ : رابطه $R(A, B)$ مفروض است . اگر مجموعه صفات تهی را در نظر بگیریم ، داریم : $A \longrightarrow \emptyset$ و $B \longrightarrow \emptyset$ و رابطه در BCNF نیست . اگر مجموعه تهی را در نظر نگیریم ، رابطه در BCNF است ، زیرا در رابطه دوگانی دترمینانی غیر از کلید کاندید وجود ندارد (یا A کلید کاندید است یا B کلید کاندید است یا هردو یا ترکیب (A,B) .)

۱۵ – ۱۵ :

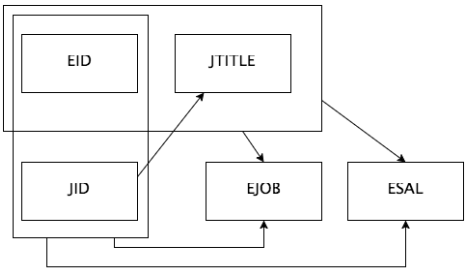
شماره‌ی کارمند — عنوان پروژه — حقوق کارمند در پروژه

EMPPROJ ( EID , JID , JTITLE , EJOB , ESAL )

شماره‌ی پروژه — سمت کارمند در پروژه

قاعده‌ی معنایی : یک کارمند در بیش از یک پروژه مشارکت دارد .

نمودار وابستگی‌های تابعی :




در این رابطه JID دترمینان است ، اما کلید کاندید نیست ، پس در BCNF نیست . تجزیه‌ی این رابطه به خواننده واگذار می‌شود .

۱۵ – ۱۶ :

$$F: \{X \to Y, Y \to\to (Z,W), (U,V) \to Z\}$$

$$(U,V) \to Z \Rightarrow (U,V) \to\to Z$$

$$Y \to\to (Z,W) \Rightarrow \begin{cases} Y \to\to (W-Z) \\ Y \to\to (Z-W) \end{cases}$$

پس :

$$F^+ = \{X \to\to Y,\ Y \to\to (Z,W),\ (U,V) \to Z, (U,V) \to\to Z,\ Y \to\to (W-Z),\ Y \to (Z-W)\}$$

۱۵ – ۱۷ :

فرض : F : مجموعه‌ی وابستگی‌های تابعی و وابستگی‌های چندمقداری رابطه‌ی R .

- الگوریتم[۱۹]

1. $Set\ D := \{ R \}$;
2. While there is a relation schema Q in D that is not in 4NF, do

   {choose a relation schema Q in D that is not in 4NF;

   Find a nontrivial $x \to\to y$ in Q D that validate 4NF;

   ??? Q in D by two relation schemas $(Q - y)$ and $(x \cup y)$;

   };

توجه : این الگوریتم برپایه‌ی قضیه فاگین نوشته شده است .

۱۵ – ۱۸ : باید در رابطه‌ی BCNF وابستگی چندمقداری وجود داشته باشد . تا در 4NF نباشد :

EJL (EID, JID, LOC)

[۱۹] برگرفته از : EMASRI R. , NAVATHE S. B. , Database Systems , 6th ed. Pearson , 2011

رابطه EJL تمام کلید است ، پس در BCNF است ، اما در 4NF نیست .

| | | |
|---|---|---|
| $e1$ | $\begin{Bmatrix} j1 \\ j2 \end{Bmatrix}$ | $\{e1\}$ |
| $e2$ | $\begin{Bmatrix} j1 \\ j3 \\ j4 \end{Bmatrix}$ | $\begin{Bmatrix} e1 \\ e2 \end{Bmatrix}$ |

۱۵ – ۱۹ :

- تجزیه‌ی ۱ :

$$\begin{cases} STT1(\underline{STID}, STNAME) \\ STT2(\underline{STID}, STLEV) \\ STT1(\underline{STID}, STMJR) \\ STT1(\underline{STID}, STDEID) \end{cases}$$

- تجزیه‌ی ۲ :

$$\begin{cases} STT1(\underline{STID}, STMJR) \\ STT2(\underline{STID}, STNAME, STDEID) \end{cases}$$

۱۵ – ۲۰ : بله ، زیرا در این صورت تمام وابستگی‌های پیوندی آن ناشی از کلید(های) کاندید ساده خواهد بود .

توجه : در روش تشخیص سریع 5NF دیدیم که : اگر رابطه‌ای در 3NF باشد و همه کلیدهای کاندید آن صفات ساده باشند ، رابطه در 5NF است . چون رابطه‌ی BCNF ، در 3NF هم هست ، پس به طریق اولی ، گزاره‌ی داده شده در تمرین درست است .

۱۵ – ۲۱ :

- مثال ۱ :

(این رابطه محدودیّت با ماهیت چرخشی دارد .)

| ECJ | (EID, | CID, | JID) |
|---|---|---|---|
| | شماره کارمند | شماره قرارداد | شماره پروژه |
| | e1 | c1 | j2 |
| | e1 | c2 | j1 |
| | e2 | c1 | j1 |
| | e1 | c1 | j1 |

- مثال ۲ :

▪ قاعده : یک بیمار ممکن است در بیش از یک مدت ، تحت درمان یک پزشک باشد .

▪ این رابطه محدودیت با ماهیت چرخشی دارد .

| PDH (PID, | شماره بیمار | DID, | شماره پزشک | TDU) | مدت درمان |
|---|---|---|---|---|---|
| | p7 | | d8 | | t2 |
| | p7 | | d9 | | t1 |
| | p9 | | d8 | | t1 |
| | p7 | | d8 | | y1 |

۱۵ – ۲۲ :

$$PRCODE = JD * (PC,\ PD,\ CD)$$

به نحوی که :

$$PD(PRID, DEID),\ \ PC(PRID, COID),\ \ CD(CID, DEID)$$

۱۵ – ۲۳ : تجزی‌ی افقی رابطه‌ی

$$STT(STID, STNAME, STLEV, STMJR, STDEID)$$

در روزهای ثبت نام به صورت زیر :

$$STD1 = \sigma_{<STDEID='D1'>}(STT)$$

$$STD2 = \sigma_{<STDEID='D2'>}(STT)$$

...

…

$$STD3 = \sigma_{<STDEID='D3'>}(STT)$$

۱۵ – ۲۴ : فهرست ارائه شده از مزایا و معایب به نظر کامل می‌آید . در اینجا توصیه می‌کنیم در مورد دو عیب شماره‌ی ۱۰ و شماره‌ی ۱۱ بیشتر فکر شود و مثال قید شود .

# تمرینات پایان گفتار

۱۵-۱ : رابطه‌های قسمت ۱-۶ از گفتار ۱۱ :

- رابطهٔ دانشجو : STT (STID, STNAME , STLEV , STMJR , STDEID)

این رابطه یک کلید کاندید دارد که می‌شود همان کلید اصلی . این رابطه در $3NF$ است و چون تنها کلید کاندید آن ، صفت ساده است ، پس در $5NF$ است .

- رابطهٔ درس : COT (COID , COTITLE , CREDIT , COTYPE , CODEID)

این رابطه هم در $5NF$ است (به دلیل گفته شده در بالا) .

- رابطهٔ انتخاب : STCOT (STID , COID , TR , YRYR , GRADE)

این رابطه غیر از تنها کلید کاندیدش ، یک صفت دیگر دارد ، پس در $6NF$ است .

- رابطه گروه آموزشی : DEPT (DEID , DETITLE , DEPHON , … , PRID)

این رابطه در $SNF$ است (به دلیل گفته شده در مورد رابطهٔ $STT$)

- رابطهٔ پیشنیازی : COPRECO ($COID , PRECOID$)

این رابطه تمام-کلید است ، پس حداقل در $BCNF$ است . محدودیّت وابستگی چند مقداری ندارد ، پس در $4NF$ است . محدودیّت با ماهیّت چرخشی ندارد ، پس در $5NF$ است . و در واقع در $6NF$ هم هست .

- رابطهٔ کتاب : BOOK (BKID , BKTITLE , BKSTATE , BKPRICE , COID)

این رابطه در $5NF$ است (به دلیل گفته شده در مورد رابطهٔ $STT$) .

- رابطهٔ استاد :

PROF (PRID , PRNAME , SPECIAL , RANK , REASARA , TEACHAREA , … , DEID)

این رابطه در $5NF$ است (به دلیل گفته شده در مورد رابطهٔ $STT$) .

- رابطهٔ استاد راهنما- دانشجو : PRST (PRID , STID , TR , YRYR)

این رابطه یک دترمینان دارد که همان کلید کاندید آن است . پس در $BCNF$ است . محدودیّت وابستگی چند تعدادی ندارد ، پس در $4NF$ است . محدودیّت با ماهیّت چرخشی ندارد ، پس در $5NF$ است .

کنجکاوی : آیا این رابطه در $6NF$ است؟

۱۵-۲ : رابطهٔ خواسته شده : EMPPOJ (EN , JN , EM , JT , EF , JC , JS)

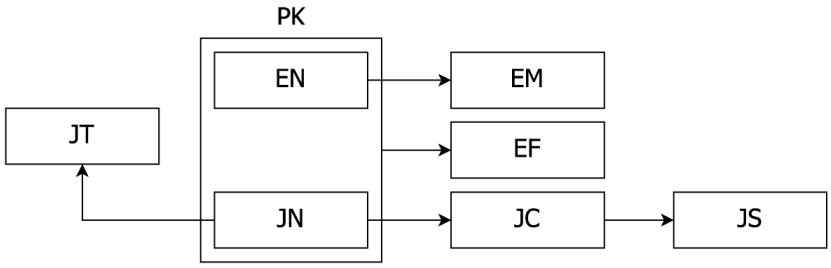
معنای صفات :

| نام صفت | معنا |
|---|---|
| EN | شمارهٔ کارمند |
| JN | شمارهٔ پروژه |
| EM | نام کارمند |
| JT | عنوان پروژه |
| EF | سمت کارمند در پروژه |
| JC | شهر محل اجرای پروژه |
| JS | استان محل اجرای پروژه |

قواعد معنایی عبارتنداز :

- یک کارمند در بیش از یک پروژه فعالیّت دارد .
- یک پروژه در یک مکان اجرا می‌شود .
- یک پروژه یک مبلغ بودجه دارد .
- سایر قواعد معنایی مشخّص‌اند .

نمودار وابستگی‌های تابعی :




رابطه‌ها :

$EMP(\underline{EN}, EM)$
$EJF(\underline{EN, JN}, EF)$
$JT(\underline{JN}, JT, JC)$
$JC(\underline{JC}, JS)$

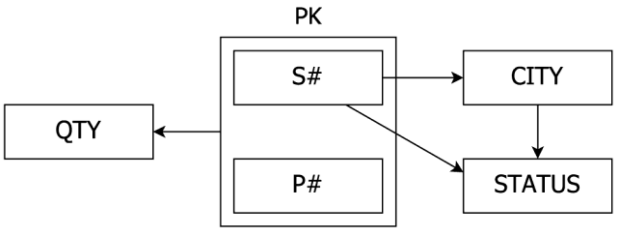
تمام این رابطه‌ها در $5NF$ هستند .

کنجکاوی : چند رابطه در $6NF$ هستند؟

۱۵-۳ :

$S\ (\underline{S\#\ ,P\#}\ ,CITY\ ,STATUS\ ,QTY)$

نمودار وابستگی‌های تابعی :



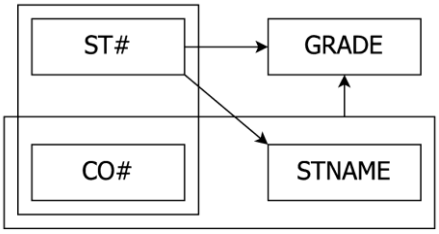

رابطه‌ها :

$SC\ (\underline{S\#}\ ,CITY)$ , $CS\ (\underline{CITY}\ ,STATUS)$ , $SP\ (\underline{S\#\ ,P\#}\ ,QTY)$

هر سه رابطهٔ بالا در $5NF$ (و نیز در $6NF$) هستند .

۱۵-۴ : SCPG (ST# , CO# , STNAME , GRADE)

اولاً : این رابطه دو کلید همپوشا دارد . این رابطه طبق تعریف زانیولو ، در $3NF$ است .




اثبات : در وابستگی‌های تابعی

$(ST\#\ ,CO\#)\ \rightarrow\ GRADE$
$(CO\#\ ,STNAME)\ \rightarrow\ GRAME$

سمت چپ کلید کاندید و در نتیجه سوپر کلید است .

اما در وابستگی $S\#\ \rightarrow\ SNAME$ ، سمت چپ سوپر کلید نیست . اما سمت راست جزیی از کلید کاندید است . پس رابطه در $3NF$ است .

کنجکاوی : چرا در $BCNF$ نیست؟

ثانیاً : این رابطه به دو طرز تجزیه می‌شود :

$$I: \begin{cases} SCG\ (\underline{ST\#, CO\#}, GRADE) \\ SS(\underline{ST\#}, STNAME) \end{cases}$$

$$II: \begin{cases} CSG\ (\underline{CO\#, STNAME}, GRADE) \\ SS\ (\underline{ST\#}, STNAME) \end{cases}$$

تجزیهٔ I ، شرط اوّل تجزیهٔ خوب دارد : صفت مشترک در یکی از دو پرتو ، کلید کاندید است ، امّا این تجزیه شرط دوّم را ندارد ، یعنی وابستگی تابعی $(CO\#, STNAME) \rightarrow GRADE$ از دست می‌رود .

تجزیه II هم تجزیهٔ خوبی نیست .

کنجکاوی : چرا؟

ثالثاً : با افزودن صفت $TERM$ به رابطهٔ $SCPG$ ، کلیدهای کاندید رابطه تغییر می‌کنند :

$$SCPGT\ (\underline{ST\#, CO\#, TERM}, STNAME, GRADE)$$

کلیدهای کاندید رابطه در دو صفت اشتراک دارند .

کنجکاوی : در این حالت رابطه در کدام فرم نرمال است؟

رابعاً : در این صورت رابطه در $1NF$ است . زیرا داریم :

$$\begin{cases} (ST\#, CO\#) \rightarrow STNAME \\ \text{ST\#} \rightarrow STNAME \end{cases}$$

وابستگی تابعی $(ST\#, CO\#) \rightarrow STNAME$ ناتمام است .

تجزیه رابطه چنین است :

$$SCG\ (\underline{ST\#, CO\#}, GRADE)$$
$$SS\ (\underline{ST\#}, STNAME)$$

هر دو پرتو در $5NF$ هستند .

خامساً : این دو رابطه در موارد زیر با هم تفاوت دارند :

۵. در فرم نرمال : در حالت اوّل رابطه در $3NF$ و در حالت دوم در $1NF$ است .

۶. در تعداد طرز تجزیه

۷. در وضع تجزیه : تجزیه در حالت اوّل خوب نیست ، حال آنکه در حالت دوم ، تجزیه خوب است .

۸. در میزان افزونگی

کنجکاوی : چرا میزان افزونگی در دو حالت متفاوت است؟

۱۵-۵ : نامگذاری

ماشین : $M$ کارگر : $W$ جنس : $P$

شناسهٔ ماشین : $M\#$ شناسه کارگر : $W\#$ شناسه جنس : $P\#$

$$\begin{cases}(W\#, P\#) \rightarrow M\# \\ M\# \rightarrow P\#\end{cases}$$

برای هر یک از سه نوع موجودیت ، یک رابطه طرّاحی می‌کنیم :

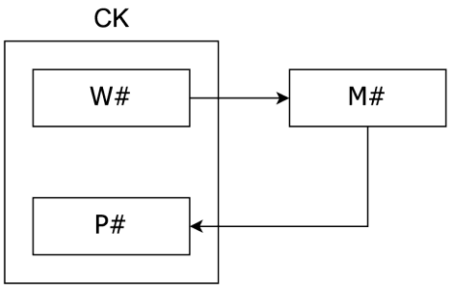
$$M\ (\underline{M\#}, \dots) \quad , \quad W\ (\underline{W\#}, \dots) \quad , \quad P\ (\underline{P\#}, \dots)$$

فرض می‌کنیم که در این سه رابطه ، وابستگی‌های تابعی ناشی از صفات غیر کلید کاندید وجود ندارد . بنابراین این سه رابطه در $5NF$ هستند .

حال برای ارتباط بین سه نوع موجودیّت رابطه زیر را طرّاحی می‌کنیم .

$$WPM\ (\underline{W\#, P\#}, M\#)$$

نمودار تابعی این رابطه چنین است :




این رابطه در $3NF$ است ، اما در $BCNF$ نیست ، زیرا $M\#$ دترمینان است امّا کلید کاندید نیست .

رابطه $MWP$ به صورت زیر تجزیه می‌شود :

$$WP(\underline{W\#, P\#}) \text{ و } MP\ (M\#, \underline{P\#})$$

هر دو رابطه در $5NF$ هستند . امّا تصمیم طرّاح به طرّاحی نرمالترین رابطه‌ها برای این محیط ، تصمیم درستی نیست ، زیرا با تجزیهٔ رابطهٔ $WPM$ به دو رابطهٔ $WP$ و $MP$ وابستگی تابعی $(W\#, P\#) \rightarrow M\#$ از دست می‌رود . در واقع این تجزیه خوب نیست . پس برای این محیط ، طرّاحی همان چهار رابطه $M$ ، $W$ و $P$ و $WPN$ کفایت می‌کند .

۶-۱۵ :

| D# | J# | P# |
|---|---|---|
| $D_1$ | $\begin{Bmatrix} J_1 \\ J_2 \end{Bmatrix}$ | $\begin{Bmatrix} P_1 \\ P_2 \end{Bmatrix}$ |
| $D_2$ | $\begin{Bmatrix} J_3 \\ J_4 \end{Bmatrix}$ | $\{P_2\}$ |
| $D_3$ | $\begin{Bmatrix} J_3 \\ J_4 \end{Bmatrix}$ | $\{P_5\}$ |

اولاً : رابطه $DJP$ تمام کلید است ، پس حداقل در $BCNF$ است .

ثانیاً : در این رابطه دو وابستگی چند مقداری وجود دارد که عبارتند : $\begin{cases} D\# \twoheadrightarrow J\# \\ D\# \twoheadrightarrow P\# \end{cases}$

طبق قضیه فاگین ، این رابطه به صورت زیر تجزیه می‌شود :

$DJ\ (\underline{D\#, J\#})$ و $DP\ (\underline{D\#, P\#})$

این تجزیه خوب است ، زیرا داریم :

$$DJP = DJ \bowtie DP$$

نتیجه اینکه شرایط تجزیه خوب (شرایط ریسانن) برای فرم‌های قوی‌تر از $BCNF$ ، مطرح نیست ، بلکه تجزیهٔ رابطهٔ $BCNF$ که در $4NF$ نباشد ، باید طبق قضیهٔ فاگین انجام شود .

ثالثاً : با اضافه کردن صفت $DATE$ داریم :

$$DJPD(D\#, J\#, P\#, DATE)$$

در این رابطه وابستگی چند مقداری ادغام شده وجود دارد ، زیرا در یک پرتو درجهٔ سه از آن ، وابستگی چند مقداری وجود دارد :

$$DJP = \prod_{<D\#,J\#,P\#>} DJPD$$

رابطه‌ای که وابستگی چند مقداری ادغام شده دارد ، می‌تواند $5NF$ (و حتی $6NF$ باشد) . در اینجا رابطهٔ $DJPD$ در $5NF$ و نیز در $6NF$ است . توجّه داشته باشیم که کلید کاندید این رابطه $(D\#, J\#, P\#)$ است و غیر از کلید کاندید ، تنها یک صفت دیگر دارد ، پس در $6NF$ است .

۷-۱۵ :

اولاً :

$R_1$

| $A$ | $B$ |
|---|---|
| $a_1$ | $b_1$ |
| $a_3$ | $b_1$ |
| $a_3$ | $b_2$ |
| $a_4$ | $b_2$ |

$R_2$

| $B$ | $C$ |
|---|---|
| $b_1$ | $c_1$ |
| $b_1$ | $c_2$ |
| $b_2$ | $c_3$ |
| $b_2$ | $c_4$ |

ثانیاً : می‌بینیم که $R' \neq R$ دلیلش این است که صفت مشترک در $R_1$ و $R_2$ ، در هیچیک از دو رابطهٔ کلید کاندید نیست.

R'

| $A$ | $B$ | $C$ |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_1$ | $b_1$ | $c_2$ |
| $a_3$ | $b_1$ | $c_1$ |
| $a_3$ | $b_1$ | $c_2$ |
| $a_3$ | $b_2$ | $c_3$ |
| $a_3$ | $b_2$ | $c_4$ |
| $a_4$ | $b_2$ | $c_3$ |
| $a_4$ | $b_2$ | $c_4$ |

در واقع این تجزیه شرایط قضیهٔ ریسانن را ندارد . در اثر پیوند دو پرتو $R_1$ و $R_2$ ، تاپل‌های حشو بروز می‌کند .

کنجکاوی : آیا در این تجزیه وابستگی‌های تابعی رابطهٔ $R$ حفظ می‌شوند؟

۸-۱۵ :

$R(A, B, C, D, E)$

داریم : $\{ (A, B) \to C, (C, D) \to E , (D, E) \to B \}$

ابتدا باید کلیدهای کاندید را پیدا کنیم :

از قاعدهٔ شبه تعدّی : $\begin{cases} (A,B) \to C \\ (C,D) \to E \end{cases} \Rightarrow (A, B, D) \to E$

از قاعدهٔ چپ‌افزایی : $(A, B) \to C \Rightarrow (A, B, D) \to C$

پس $(A, B, D)$ یک کلید کاندید است .

از قاعدهٔ چپ‌افزایی : $(C, D) \to (C, D) \Rightarrow (A, C, D) \to (C, D)$

از قاعدهٔ تعدّی : $\begin{cases} (A, C, D) \to (C, D) \\ (C, D) \to E \end{cases} \Rightarrow (A, C, D) \to E$

از قاعدهٔ افزایش : $(A, C, D) \to E \Rightarrow (A, C, D) \to (D, E)$

از قاعدهٔ تعدّی : $\begin{matrix} (A, C, D) \to (D, E) \\ (D, E) \to B \end{matrix} \Rightarrow (A, C, D) \to B$

پس $(A, C, D)$ هم کلید کاندید است .

از قاعدهٔ چپ‌افزایی : $(D, E) \to (D, E) \Rightarrow (A, D, E) \to (D, E)$

از قاعدهٔ تعدّی : $\begin{cases} (A, D, E) \to (D, E) \\ (D, E) \to B \end{cases} \Rightarrow (A, D, E) \to B$

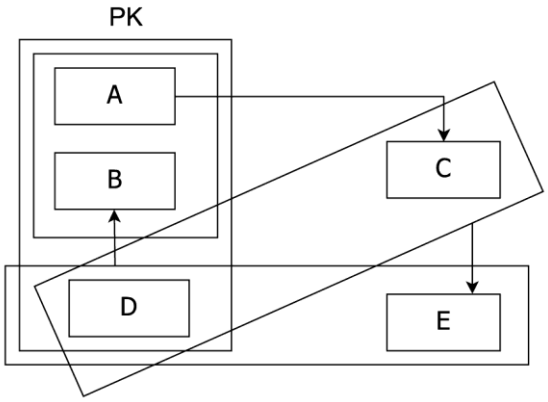
از قاعدهٔ شبه تعدّی : $\begin{cases} (A, D, E) \to B \\ (A, B) \to C \end{cases} \Rightarrow (A, D, E) \to C$

پس $(A, D, E)$ هم کلید کاندید است .

اگر کلید کاندید (مثلاً $(A, B, D)$) را کلید اصلی درنظر بگیریم ، دو کلید کاندید دیگر ، کلید بدیل می‌شوند .

آیا این رابطه تجزیه خوب دارد؟ این رابطه در $BCNF$ نیست (چرا؟) . فرض می‌کنیم $(A, B, D)$ کلید اصلی رابطه باشد .

نمودار وابستگی‌های تابعی رسم می‌کنیم :




یک تجزیهٔ ممکن چنین است :

$$R_1(\underline{A,B},C) \quad , \quad R_2(\underline{D,E},B) \quad , \quad R_3(\underline{C,D},E)$$

این تجزیه خوب نیست زیرا شرایط قضیّهٔ ریسانن را ندارد .

کنجکاوی : رابطه‌های حاصله در کدام فرم نرمال هستند؟

مجموعهٔ کاهش ناپذیر : $\{ (A,B) \to C, (C,D) \to E, (D,E) \to B \}$

زیرا همین مجموعه شرایط مجموعهٔ کاهش ناپذیر را دارد .

کنجکاوی : چرا؟

۹-۱۵ : داریم : $R(A,B,C,D,E,F,G,H,I,J)$

$(A,B,C) \to E$ , $(A,B) \to G, B \to F, C \to J, (C,J) \to I$ و $G \to H$

این مجموعه کاهش پذیر است . زیرا مثلاً از $C \to J$ و $(C,J) \to I$ نتیجه می‌شود : $C \to I$ ، یعنی سمت چپ $(C,J) \to I$کاهش می‌یابد به $C \to I$ .

کنجکاوی : یک کلید کاندید این رابطه چیست؟

کنجکاوی : مجموعهٔ کاهش ناپذیر کدام است؟

پاسخ : روشن است که اجتماع سمت چپ تمام وابستگی‌های تابعی داده شده ، سوپر کلید است : $(A,B,C,G,J)$

می‌توان از این مجموعه $J$ را حذف کرد ، زیرا داریم $C \to J$ . صفت $G$ را هم می‌توان حذف کرد . زیرا : $(A,B) \to G$ . پس $(A,B,C)$ یک کلید کاندید است .

کنجکاوی : اگر به جای $(A,B,C) \to E$ داشته باشیم : $(A,B,D) \to E$ ، کلید کاندید رابطه چیست؟

۱۵-۱۰ : داریم : $R(A,B,C,D,E)$

1) $\{ A \to B, (A,B) \to C, D \to (A,C), D \to E \}$
2) $\{ A \to (B,C), D \to (A,E)\}$

این دو مجموعه معادلند . یک راه برای اثبات این ادعا این است که نشان دهیم از مجموعهٔ ۱ می‌توان مجموعهٔ ۲ را بدست آورد و بر عکس . اما راه دیگر این است که نشان دهیم بستار دو مجموعه یکسان است . مجموعهٔ ۱ را $F$ و مجموعهٔ ۲ را $G$ می‌نامیم . با روش اول نشان می‌دهیم که $F$ و $G$ معادلند .

داریم : $G = \{ A \to (B,C), D \to (A,E)\}$

$$A \to (B,C) \Rightarrow \begin{cases} A \to B \\ A \to C \end{cases} \Rightarrow (A,B) \to C$$
$$\begin{cases} D \to (A,E) \to D \to A \\ A \to C \end{cases} \Rightarrow D \to C \Rightarrow D \to (A,C)$$
$$D \to (A,E) \to \begin{cases} D \to A \\ D \to E \end{cases}$$

پس از $G$ می‌توان $F$ را بدست آورد . اینک نشان می‌دهیم که از $F$ هم می‌توان $G$ را بدست آورد .

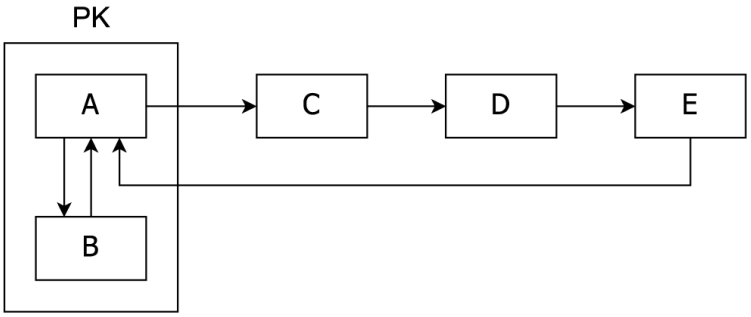
داریم : $F = \{ A \to B, (A,B) \to C, D \to (A,C), D \to E \}$

$$D \to (A,C) \Rightarrow \begin{cases} D \to A \\ D \to C \end{cases}$$
$$\begin{cases} D \to A \\ D \to E \end{cases} \Rightarrow D \to (A,E)$$
$$A \to B \Rightarrow A \to (A,B)$$
$$\begin{cases} A \to (A,B) \\ (A,B) \to C \end{cases} \Rightarrow A \to C$$
$$\begin{cases} A \to C \\ A \to B \end{cases} \Rightarrow A \to (B,C)$$

کنجکاوی : آیا این روش همان روش اثبات $F^+ = G^+$ است؟

۱۵-۱۱ : داریم : $R(\underline{A,B},C,D,E,F)$ و $A \to E \,.\, D \to E \,.\, C \to D \,.\, A \rightleftarrows B$

نمودار وابستگی‌های تابعی چنین است :




این رابطه در $2NF$ است . زیرا وابستگی تابعی ناتام ندارد . امّا وابستگی تابعی با واسطه دارد ، پس در $3NF$ نیست . تجزیهٔ رابطه چنین است :

$R_1(\underline{A},\underline{B})$ , $R_2(\underline{A},C)$ یا $(\underline{B},C)$ , $R_3(\underline{C},D)$ , $R_4(\underline{D},E)$ , $R_5(\underline{E},D)$

تمام رابطه‌های حاصله در $5NF$ ( و نیز در $6NF$) هستند .

۱۵-۱۲ : رابطهٔ $R_3(\underline{EMP\#},DEPT\#)$ افزونه است . می‌توان صفت $DEPT\#$ را در رابطهٔ $R_1$ قرار دارد زیرا در هر دو رابطه صفت $EMP\#$ تنها کلید کاندید است .

۱۵-۱۳ : با توجه به کلید کاندید رابطه‌ها و این نکته که تمام صفات رابطه با کلید کاندید آن وابستگی تابعی دارند ، نتیجه می‌شود که گزارهٔ **الف** نادرست است . گزاره‌های دیگر درست هستند .

۱۵-۱۴ : $INVENTORY(INV-NO\ ,UNIT-PRICE,COLOR,SIZE)$

برای تشخیص وجود وابستگی‌های چند مقداری ، باید بتوان مجموعه‌ای به خوبی تعریف شده برای هر یک از دو صفت (ساده یا مرکّب) متناظر با یک مقدار مشخص از جفت صفات دیگر (هر یک ساده یا مرکّب) را در بدنهٔ داده شده باز شناخت . (البته از لحاظ تئوریک باید طبق تعریف تئوریک وابستگی چند مقداری عمل کرد) .

در بدنهٔ داده شده داریم :

INVEMTURY

| INVUNIT | | COLOR | SIZE |
|---|---|---|---|
| INV-NO | UNIT-PRICE | | |
| {11 , 2} | | $\begin{Bmatrix}C_1\\C_2\end{Bmatrix}$ | $\begin{Bmatrix}1\\2\\3\\4\end{Bmatrix}$ |
| {12 , 4} | | $\begin{Bmatrix}C_3\\C_4\\C_2\end{Bmatrix}$ | $\begin{Bmatrix}1\\2\end{Bmatrix}$ |
| | | | |

بنابراین داریم : $\begin{cases} INVUNIT \twoheadrightarrow COLOR \\ INVUNIT \twoheadrightarrow SIZE \end{cases}$

توجّه : سمت چپ این دو وابستگی چند مقداری صفت مرکّب (INV-NO , UNIT-PRICE) است .

این رابطه در $BCNF$ است زیرا تمام-کلید است . باید طبق قضیهٔ فاگین تجزیه شود :

$INVUNITCOLOR$ (INV-NO , UNIT-PRICE , $COLOR$)

$INVUNITSIZE$ (INV-NO , UNIT-PRICE , $SIZE$)

کنجکاوی : این دو رابطه در کدام فرم نرمال هستند؟

۱۵-۱۵ : رابطهٔ داده شده : $REL(C, T, H, R, S, G)$

اولاً : با توجّه به بدنهٔ داده شده ، کلید این رابطه $(C, T, H, R, S)$ است . در این رابطه وابستگی چند مقداری وجود ندارد (به کنجکاوی‌های داده شده دقّت شود) . این رابطه غیر از کلید کاندیدیش ، یک صفت دیگر ، $G$ ، دارد ، پس در $6NF$ است .

کنجکاوی ۱ : آیا این رابطه کلید کاندید دیگر دارد؟

کنجکاوی ۲ : آیا در این رابطه وابستگی چند مقداری ادغام شده وجود دارد؟

کنجکاوی ۳ : اگر مقادیر صفت $R$ در شش تاپل ، به ترتیب $R_1$ ، $R_1$ ، $R_1$ ، $R_2$ ، $R_2$ و $R_2$ باشند ، چه پیش می‌آید؟

پاسخ کنجکاوی ۱ : بله ، ترکیب $(C, T, H, R, G)$ هم کلید کاندید است .

پاسخ کنجکاوی ۲ : بله ، زیرا در یک پرتو روی سه گروه از صفات ، وابستگی چند مقداری وجود دارد :

$$R = \prod_{<C,T,H,S,G>} (R \in L)$$

بدنهٔ $R$ چنین است :

R ( C , T , H , S , G )

| | | |
|---|---|---|
| $\{c_1 \ , \ t_1\}$ | $\begin{Bmatrix} H_1 \\ H_2 \\ H_3 \end{Bmatrix}$ | $\{s_1 \ , \ b\}$ |
| $\{c_1 \ , \ t_1\}$ | $\begin{Bmatrix} H_1 \\ H_2 \\ H_3 \end{Bmatrix}$ | $\{s_2 \ , \ c\}$ |

پس داریم :

$$\begin{cases} (C,T) \twoheadrightarrow H \\ (C,T) \twoheadrightarrow (S,G) \end{cases}$$

توجّه : وجود وابستگی چند مقداری ادغام شده مانع $5NF$ (و حتی $6NF$) بودن رابطه نیست .

پاسخ کنجکاوی ۳ : در این صورت ، دو وابستگی چند مقداری در رابطه پدید می‌آید و رابطه در $BCNF$ است . بدنهٔ رابطه چنین است :

R ( C , T , H , R , S , G )

| | | |
|---|---|---|
| $\{c_1 \ , \ t_1\}$ | $\begin{Bmatrix} H_1 \\ H_2 \\ H_3 \end{Bmatrix}$ | $\{R_1 \ , \ s_1 \ , \ b\}$ |
| $\{c_1 \ , \ t_1\}$ | $\begin{Bmatrix} H_1 \\ H_2 \\ H_3 \end{Bmatrix}$ | $\{R_2 \ , \ s_2 \ , \ b\}$ |

و داریم : $(C,T) \twoheadrightarrow (R,S,G)$ و $(C,T) \twoheadrightarrow H$

۱۵-۱۶ : رابطهٔ داده شده : ADR (<u>NAME</u> , STREET , CITY, STATE , ZIP)

داریم : $(STREET, CITY, STATE) \rightarrow ZIP$

$ZIP \rightarrow (CITY, STATE)$

و چون $NAME$ کلید کاندید است ، داریم : $NAME \rightarrow ZIP$

پس در این رابطه وابستگی تابعی با واسطه وجود دارد ، در نتیجه رابطه در $2NF$ است .

این رابطه به صورت زیر تجزیه می‌شود :

$NSZ\ (\underline{NAME}, STREET, ZIP)$
$ZCS\ (\underline{ZIP}, CITY, STATE)$

این هر دو رابطه در $5NF$ هستند .

کنجکاوی : آیا در این تجزیه همهٔ وابستگی‌های تابعی حفظ می‌شوند؟

اگر قرار باشد نشانی کامل افراد بطور هفتگی تولید شود ، برنامهٔ کاربردی باید دو رابطه $NSZ$ و $ZCS$ را پیوند کند و این عمل سربار خاص خود را دارد . بنابراین چنین تجزیه‌ای توصیه نمی‌شود . نتیجه این که گاه لازم می‌شود که رابطه‌ای را در سطح نرمال پائین‌تر نگاه داریم تا سرعت اجرای برنامه کاربردی تأمین شود .

۱۵-۱۷ : رابطهٔ داده شده :

$$R(\underline{X,Y},Z,V,W)$$

داریم : $X \to Z \, , X \to U \, , U \to V \, , Y \to W$

در این رابطه وابستگی تابعی ناتام وجود دارد ، پس رابطه در $1NF$ است . این رابطه به صورت زیر تجزیه می‌شود :

$$R_1(\underline{X},Z,U) \quad , \quad R_2(\underline{U},V) \quad , \quad R_3(\underline{Y},W) \quad , \quad R_4(\underline{X,Y})$$

همه این رابطه‌ها در $5NF$ هستند (سه رابطهٔ $R_2$ ، $R_3$ و $R_4$ در $6NF$ هم هستند) .

در این تجزیه تمام وابستگی‌های تابعی حفظ می‌شوند ، زیرا داریم :

$$X \to Z \, , X \to U \, , U \to V \, , Y \to W$$

و از قاعدهٔ چپ‌افزایی : $(X,Y) \to W$ و $(X,Y) \to V$ و $(X,Y) \to U$ و $(X,Y) \to Z$

۱۵-۱۸ : داریم :

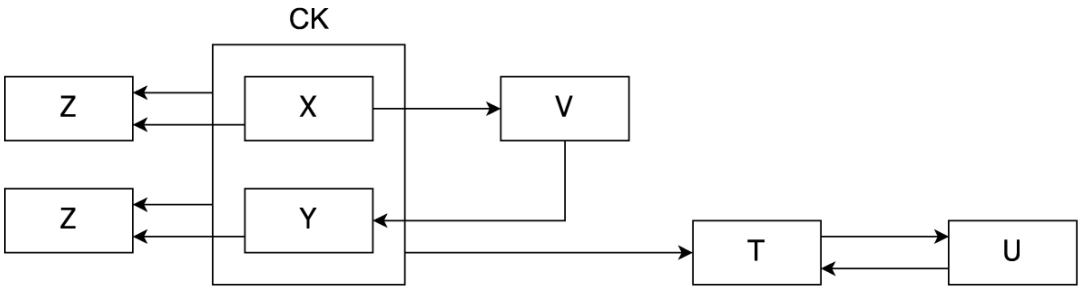
$$Y \to W \, , (X,Y) \to Z , X \to U \, , (X,Y) \to V$$

$$V \to Y \, , (X,Y) \to T , T \rightleftarrows U$$

رابطهٔ عمومی $R(X,Y,x,w,u,V,T)$ را در نظر می‌گیریم . در این رابطه $(X,Y)$ کلید کاندید است ، زیرا داریم :

$$\begin{cases} Y \to W \Rightarrow (X,Y) \to W \\ (X,Y) \to Z \\ (X,Y) \to V \\ \begin{cases} (X,Y) \to T \\ T \to U \end{cases} \Rightarrow (X,Y) \to U \end{cases}$$

برای وضوح بیشتر نمودار وابستگی‌های تابعی را رسم می‌کنیم :




نرمالترین رابطه‌ها :

$R_1(\underline{X,Y},V,T)$ $R_2(\underline{X},Z)$ $R_3(\underline{Y},W)$ $R_4(\underline{V},Y)$ $R_5(\underline{T},\underline{U})$

همهٔ رابطه‌ها در $5NF$ هستند (و غیر از $R_1$ ، بقیه در $6NF$ هستند) . به آسانی می‌توان نشان داد که همهٔ وابستگی‌های تابعی حفظ شده‌اند .

۱۹-۱۵ : رابطهٔ داده شده :

| $R$ | $X$ | $Y$ | $Z$ |
|---|---|---|---|
| | $x_1$ | $y_2$ | $z_3$ |
| | $x_3$ | $y_2$ | $z_6$ |
| | $x_3$ | $y_4$ | $z_2$ |

اولاً :

$R_1 = \prod_{<x,y>}(R) =$

| $X$ | $Y$ |
|---|---|
| $x_1$ | $y_2$ |
| $x_3$ | $y_2$ |
| $x_3$ | $y_4$ |

$R_۲ = \prod_{<غوظ>}(R) =$

| $Y$ | $Z$ |
|---|---|
| $y_2$ | $z_3$ |
| $y_2$ | $z_6$ |
| $y_4$ | $z_2$ |

ثانیاً : با پیوند دو پرتو ، تاپل‌های حشو بروز می‌کند .

کنجکاوی : چرا تاپل‌های حشو بروز کرده است؟

ثالثاً : هر چه باشد مقادیر $X$ یا $Z$ ، باز هم پدیدهٔ مزبور بروز می‌کند زیرا صفت مشترک در دو پرتو ، $Y$ ، در هیچ‌یک از دو رابطه $R_1$ و $R_2$ ، کلید کاندید نیست (شرط اول قضیه ریسانن برقرار نیست) .

$R_1 \bowtie R_2 = R'$

| $X$ | $Y$ | $Z$ | |
|---|---|---|---|
| $x_1$ | $y_2$ | $z_3$ | |
| $x_1$ | $y_2$ | $z_6$ | ← تاپل حشو |
| $x_3$ | $y_2$ | $z_3$ | |
| $x_3$ | $y_2$ | $z_6$ | ← تاپل حشو |
| $x_3$ | $y_4$ | $z_2$ | |

توجّه : دلیل ارائه شده در قسمت ثالثاً ، پاسخ کنجکاوی قسمت ثانیاً است .

۱۵-۲۰ : رابطهٔ داده شده :

$$BOOK(\underline{BKID}\ ,\underline{BKTITLE}\ ,BKPUB\ ,PKPRICE\ ,BKPAGENO)$$

اولاً :

$$BK1\ (\underline{BKID}\ ,BKPUB\ ,BKPR2CE)$$
$$BK2\ (\underline{BKID}\ ,\underline{BKTITLE})$$
$$BK3\ (\underline{BKID}\ ,BKPAGENO)$$

این تجزیه خوب است (شرایط قضیه ریسانن را دارد) .

توجّه : در عنوان رابطه‌های $BK1$ و / یا $BK3$ می‌توان به جای صفت $BKID$ صفت $BKTITLE$ را وارد کرد .

امّا اگر این تجزیه به دلیل نرمالترسازی باشد ، لازم نیست ، زیرا رابطهٔ $BOOK$ در $5NF$ است .

ثانیاً : رابطهٔ $BOOK$ دو کلید کاندید ساده دارد ، وابستگی‌های تابعی ناتام و با واسطه ندارد ، پس در $5NF$ است .

توجّه : اگر صورت نرمال $6NF$ مورد نظر باشد ، رابطهٔ $BOOK$ را می‌توان به چهار رابطهٔ زیر تجزیه کرد :

$BK1\ (\underline{BKID}, BKPUB)$
$BK2\ (\underline{BKID}, BKPRICE)$
$BK3\ (\underline{BKID}, \underline{BKTITLE})$
$BK4\ (\underline{BKID}, BKPAGENO$

توجّه : در عنوان رابطه‌های $BK1$ و $BK2$ و $BK4$ به جای صفت $BKID$ می‌توان صفت $BKTITLE$ را وارد کرد .

۱۵-۲۱ : سیستم‌های پایگاهی سلسله مراتبی و شبکه‌ای ، اینک چندان مطرح نیستند . انجام طرّاحی‌های خواسته شده را به خواننده علاقمند به اینگونه سیستم‌ها وامی‌گذاریم .

(راهنمایی : این طرّاحی مربوط است به یک نوع ارتباط سه‌گانی و با چندی $M : N : P$ که در آن $Q$ صفت نوع ارتباط است) .

۱۵-۲۲ : رابطهٔ داده شده :

$R$

| $A$ | $B$ | $C$ |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_1$ | $b_2$ | $c_1$ |
| $a_1$ | $b_1$ | $c_2$ |
| $a_1$ | $b_2$ | $c_2$ |
| $a_2$ | $b_3$ | $c_3$ |
| $a_3$ | $b_4$ | $c_1$ |
| $a_3$ | $b_4$ | $c_4$ |

اولاً : چون رابطه تمام کلید است ، پس حداقل در $BCNF$ است .

ثانیاً : در این رابطه وابستگی‌های چند مقداری زیر وجود دارد : $A \twoheadrightarrow B\ , A \twoheadrightarrow C$

کنجکاوی ۱ : چرا؟

بنابراین می‌تواند طبق قضّیه فاگین به این صورت تجزیه شود : $R_1(A, B)$ ، $R_2(A, C)$

این تجزیه خوب است . زیرا $R_1 \bowtie R_2 = R$ و چون رابطه تمام-کلید است ، شرط محفوظ ماندن وابستگی اصلاً مطرح نیست . البّته در اینجا این شرط هم برقرار است .

کنجکاوی۲ : چرا؟

پاسخ کنجکاوی۲ : وابستگی‌های تابعی R عبارتند از : $H_R \to X | X \subseteq H_R$

و این وابستگی‌ها نامهّم هستند . همین وابستگی‌ها را می‌توان از اجتماع وابستگی‌های تابعی موجود در $R_1$ و $R_2$ بدست آورد : $H_{R_1} \to X | X \subseteq H_{R_1}$ و $H_{R_2} \to Y | Y \subseteq H_{R_2}$

$$(A,B) \to A \, , (A,B) \to B \Rightarrow \begin{cases} (A,B,C) \to A \\ (A,B,C) \to B \end{cases}$$
$$(A,C) \to A, (A,C) \to C \Rightarrow \begin{cases} (A,B,C) \to A \\ (A,B,C) \to C \end{cases} \Rightarrow (A,B,C) \to (A,B,C)$$

۱۵-۲۳ : رابطهٔ داده شده :

$$R(A,B,C,D,E,F)$$

مفروضات : $A \to B \, , A \to C \, , D \to F \, , D \to A$

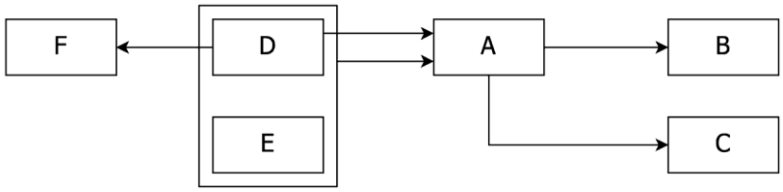
باتوجه به وابستگی‌های تابعی داده شده ، داریم :

$$(A,D) \to A \, , (A,D) \to B \, , (A,D) \to C \, , (D,A) \to F$$

$$\begin{cases} D \to A \\ (A,D) \to B \end{cases} \Rightarrow D \to B \quad \begin{cases} D \to A \\ (A,D) \to C \end{cases} \Rightarrow D \to C$$

درنتیجه $(D,E) \to A,B,C,D,E,F$ ، پس $(D,E)$ کلید کاندید است .

نمودار وابستگی‌های تابعی چنین است :




این رابطه در $1NF$ است و به صورت زیر تجزیه می‌شود :

$$R_1(\underline{D}, F, A) \quad , \quad R_2(\underline{A}, B, C) \quad , \quad R_3(\underline{D,E})$$

اثبات :

وابستگی‌های تابعی رابطهٔ R حفظ می‌شوند ، زیرا :

$$\begin{cases} D \to F \Rightarrow (D,E) \to F \\ D \to A \Rightarrow \begin{cases} (D,E) \to A \\ A \to B \end{cases} \Rightarrow (D,E) \to B \end{cases}$$

$$\begin{cases} (D,E) \to A \\ A \to C \end{cases} \Rightarrow (D,E) \to C$$

$$(D,E) \to (D,E) \Rightarrow \begin{cases} (D,E) \to D \\ (D,E) \to F \end{cases}$$

با پیوند پرتوها تاپل حشو بروز نمی‌کند ، زیرا :

$$R_{13} = R_1 \bowtie R_3 \Rightarrow R_{13}(D,F,A,E)$$

چون صفت مشترک در دو $R_1$ و $R_3$ ، در $R_1$ کلید کاندید است ، پس این پیوند از نوع CK-FK است .

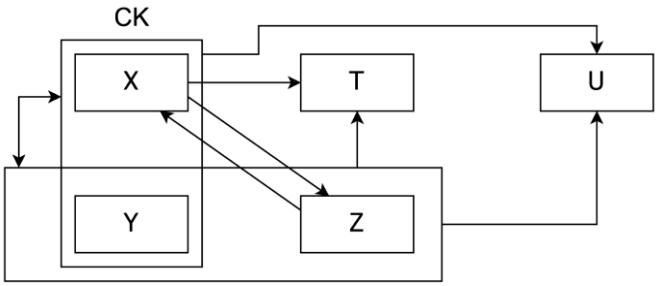
و داریم : $R = R_{13} \bowtie R_2$

زیرا صفت مشترک در $R_2$ و $R_{12}$ ، در $R_2$ کلید کاندید است ، پس این پیوند از نوع CK-FK است ، که بازسازنده است .

۲۴-۱۵ : رابطهٔ داده شده : $REL(X,Y,Z,T,U)$

مفروضات : $(X,Y)$ و $(Y,Z)$ کلیدهای کاندید هستند و $X \rightleftarrows Z$

اولاً : نمودار وابستگی‌های تابعی :




این رابطه در $3NF$ است .

اثبات : باتوجّه به تعریف زانیولو از $3NF$ ، در وابستگی‌های تابعی $X \to Z$ و $Z \to X$ ، سمت راست هر دو وابستگی صفت جزء کلید هستند . یعنی شرط سوّم $3NF$ بودن برقرار است .

توجّه : این رابطه در $BCNF$ نیست زیرا $X$ و $Z$ هر دو دترمینان هستند امّا کلید کاندید نیستند .

کنجکاوی : آیا این رابطه با تعریف کاد هم در $3NF$ است؟

راهنمایی : وابستگی تابعی $X \to Z$ یا $Z \to X$ مانع $2NF$ بودن رابطه نیست .

ثانیاً : اگر داشته باشیم : $T \to U$ ، در این صورت رابطه دیگر در $3NF$ نیست ، زیرا :

$$\begin{cases}(X,Z) \to T \to U \\ (Y,Z) \to T \to U\end{cases}$$

یعنی وابستگی تابعی با واسطه پدید می‌آید .

این رابطه در $2NF$ است و برای نرمالتر شدن باید تجزیه شود .

تجزیه‌های ممکن :

$$I \begin{cases}R_1\ (\underline{X,Y},T) \\ R_2\ (\underline{X},\underline{Z}) \\ R_3\ (\underline{T},U)\end{cases} \quad II \begin{cases}R_1\ (\underline{Y,Z},T) \\ R_2\ (\underline{X},\underline{Z}) \\ R_3\ (\underline{T},U)\end{cases}$$

کنجکاوی : آیا این دو تجزیه خوب هستند؟

راهنمایی : از قضیهٔ ریسانن استفاده شود .

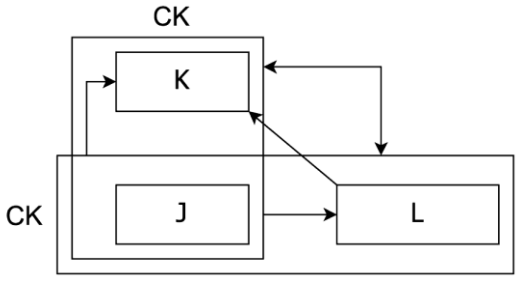
۱۵-۲۵ : رابطهٔ داده شده : $S(J,K,L)$

مفروضات : $L \to K$ و $(J,K) \to L$

این رابطه دو کلید کاندید دارد : $(J,K)$ و $(L,J)$

توجّه : از $L \to K$ داریم $(L,J) \to K$

نمودار وابستگی‌های تابعی :




این رابطه در $BCNF$ نیست زیرا $L$ دترمینان است اما کلید کاندید نیست ، با توجه به تعریف زانیولو از $3NF$ ، این رابطه در $3NF$ است زیرا در $L \to K$ صفت $K$ جزء کلید است . وابستگی‌های دیگر ناشی از کلید کاندید هستند و کلید کاندید هم خود سوپر کلید است .

این رابطه به دو طرز تجزیه دارد :

$$I \begin{cases} R_1(\underline{L}, K) \\ R_2(\underline{L, J}) \end{cases} \qquad II \begin{cases} R_1(\underline{L}, K) \\ R_2(\underline{K, J}) \end{cases}$$

همهٔ پرتوها در $BCNF$ (بلکه در $6NF$) هستند . این تجزیه‌ها خوب نیستند ، زیرا شرایط قضیهٔ ریسانن را ندارند .

کنجکاوی : کدام تجزیه ، کدام شرط را ندارد؟

نتیجه اینکه در تجزیهٔ رابطه $3NF$ به رابطه‌های $BCNF$ ممکن است تجزیه بی‌حذف و / یا بی حشو نباشد . یعنی رابطه تجزیه شدنی تجزیهٔ خوب نداشته باشد ، در این صورت نباید آن را تجزیه کرد .

اگر $(K, L)$ هم کلید کاندید باشد ، در این صورت رابطه سه کلید کاندید دارد که دو به دو همپوشا هستند و دترمینان ناکلید ندارد ، در نتیجه رابطه در $BCNF$ است .

توجّه : صِرف همپوشا بودن کلیدهای کاندید مانع $BCNF$ بودن (و حتّی در فرم قویتر بودن) رابطه نیست .

۱۵-۲۶ : با توجه به نمودار داده شده ، رابطه‌ها چنین‌اند :

$R_1(\underline{\text{SERIAL-N0 , ITEM-IN-SALE}}\text{ , QTY-SOLD , ITEM-PRICE})$
$R_2(\underline{\text{SERIAL-N0}}\text{ , }\underline{SELLER})$
$R_3(\underline{SELLER}\text{ , }\underline{REGION})$

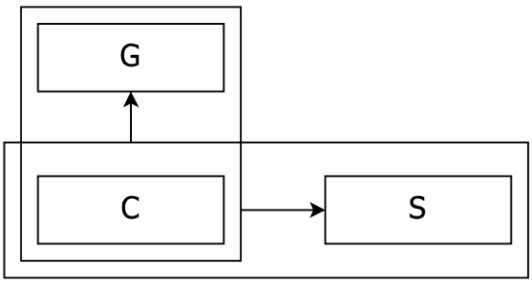
هر سه رابطه در $5NF$ هستند (رابطه‌های $R_2$ و $R_3$ در $6NF$ هم هستند) .

۱۵-۲۷ : رابطهٔ داده شده : $EXAM(S, C, G)$

قاعدهٔ معنایی : دو دانشجو در یک درس نمرهٔ یکسان نمی‌گیرند . با توجه به قاعده معنایی داریم : $(C, G) \to S$ بعلاوه ، یک دانشجو در یک درس ، یک نمره می‌گیرد . پس : $(S, C) \to G$

این رابطه دو کلید کاندید همپوشا دارد : $(C, G)$ و $(S, C)$ .

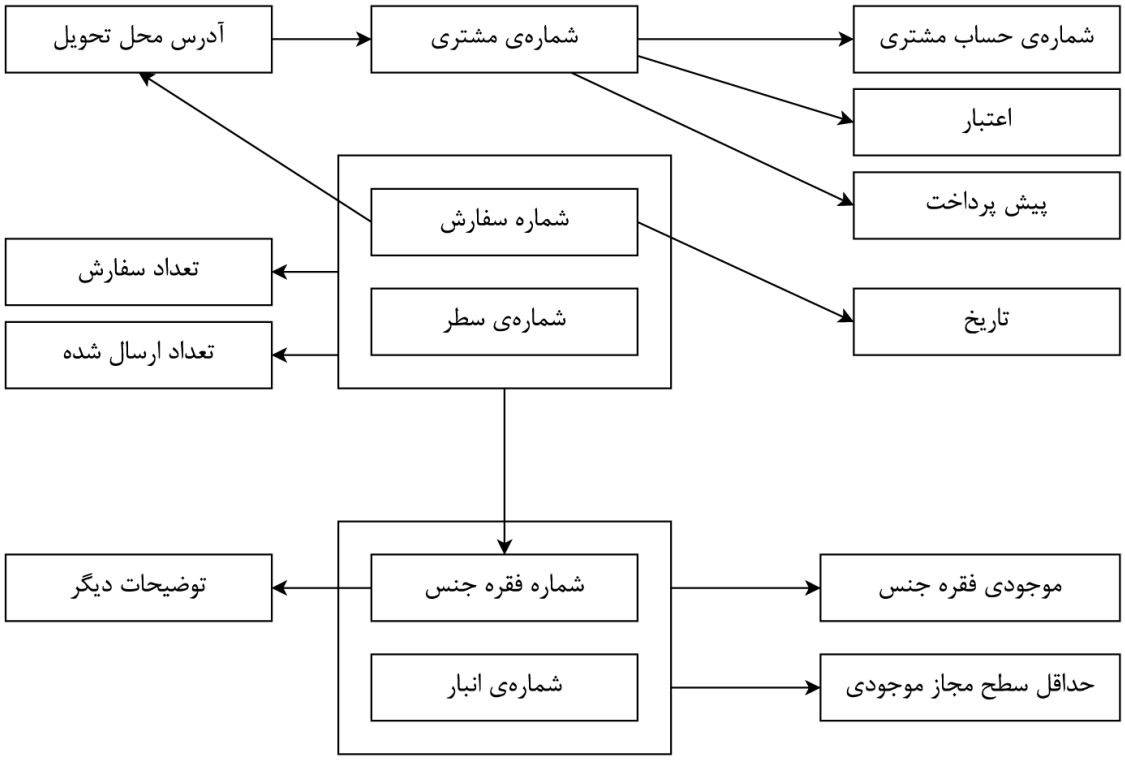
نمودار وابستگی‌های تابعی چنین است :




این رابطه در $BCNF$ است ، دو دترمینان دارد و هر دو کلید کاندید هستند . این رابطه نیازی به تجزیه ندارد .

توجّه : از این مسئله هم این نتیجه حاصل می‌شود که صِرف همپوشا بودن کلیدهای کاندید ، مانع $BCNF$ بودن رابطه (و حتی در فرم قویتر بودن) نیست .

۱۵-۲۸ : با توجه به قواعد معنایی محیط ، نمودار وابستگی‌های تابعی چنین است :




رابطه‌های لازم برای این محیط :

$CUST\ (\underline{CUST\#}\ ,BAL\ ,CREDLIM\ ,DISCOUMT)$
$SHIPTO\ (\underline{ADDRESS}\ ,CUST\#)$
$ORDHEAD\ (\underline{ORD\#}\ ,ADDRESS\ ,DATE)$
$ORDLINE\ (\underline{ORD\#\ ,LINE\#}\ ,ITEM\#\ ,QTYORD\ ,QTYOUT)$
$ITEM\ (\underline{ITEM\#}\ ,DESCN)$
$IP\ (\underline{ITEM\#\ ,PLANT\#}\ ,QTYOH\ ,DANGER)$

این طراحی از $[DATEO3]$ است .

۱۵-۲۹ : رابطهٔ داده شده :

$STCOPR\ (STID\ ,COID\ ,PRID\ ,COSEC\#\ ,TERM)$

با توجّه به قاعدهٔ : "یک دانشجو یک درس را در یک ترم با یک استاد انتخاب می‌کند" ، داریم :

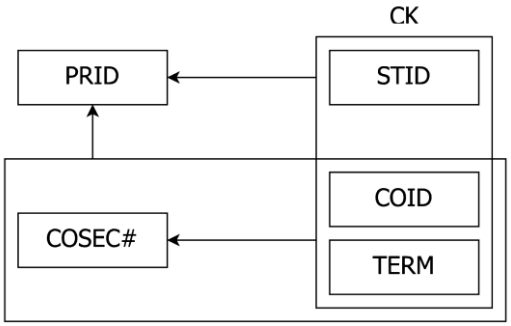
$(STID\ ,COID\ ,TERM) \rightarrow PRID$
$(STID\ ,COID\ ,TERM) \rightarrow COSEC\#$

وابستگی تابعی دوم از قاعدهٔ : "یک دانشجو در یک ترم ، در یک گروه درسی از یک درس ثبت‌نام می‌کند" ، بدست می‌آید.

امّا در این محیط معمولاً این قاعده را هم داریم که : "یک گروه درسی از یک درس در یک ترم توسط یک استاد تدریس می‌شود" .

بر پایهٔ این قاعده داریم : $(COID\ , COSEC\#\ , TERM) \rightarrow PROF$

نمودار وابستگی‌های تابعی رابطه چنین است :




توجّه : در این رابطه $(COID\ , COSEC\#\ , TERM)$ کلید کاندید نیست ، زیرا نداریم :

$(CIOD\ , COSEC\#\ , TERM) \rightarrow STID$

پس در این رابطه یکی از دترمینان‌ها کلید کاندید نیست ، پس رابطه در $BCNF$ نیست . این رابطه در $3NF$ است زیرا نه وابستگی تابعی ناتمام دارد و نه وابستگی تابعی با واسطه .

این رابطه را می‌توان به سه رابطه زیر تجزیه کرد :

$R_1(\underline{STID\ , COID\ , TERM\ }, COSEC\#)$
$R_2(\underline{COID\ , COSEC\#\ , TERM\ }, PRID)$
$R_3\big(\underline{STID\ , COID\ , TERM\ }, PRID\big)$

کنجکاوی : اگر به جای دو رابطهٔ $R_1$ و $R_3$ (که کلید کاندید یکسان دارند) ، همان رابطهٔ $STCOPR$ را داشته باشیم ، چه پیش می‌آید؟

۱۵-۳۰ : رابطه‌های لازم چنین‌اند :

CLINIC (C-NAME, ADDRESS, PHON-NO)

DOCTOR (C-NAME, D-NAME, SPECIALITY, YRSEXPER)

EMPLOYEE (C-NAME, EMPNO, E-NAME, ADDRESS)

INSURCO (C-NAME, EMPNO, I-NAME, ADDRESS, PHON-NO)

CHILD (C-NAME, EMPNO, CL-NAME, CHILD-AGE, SCH-YEAR)

نتیجه اینکه در تبدیل مدل سلسله مراتبی به طرّاحی پایگاه‌داده رابطه‌ای ، به تدریج که ژرفای مسیرها افزایش می‌یابد ، کلیدهای کاندید طولانی‌تر می‌شوند . بعلاوه تعداد و طول کلیدهای خارجی به تدریج زیادتر می‌شود . یادآوری می‌کنیم که یکی از معایت رابطهٔ نرمال ، دشواری در نمایش ارتباط‌های سلسله مراتبی است .

توجه : هنوز تعداد زیادی از سیستم‌های پایگاهی موجود در دنیا ، مدل سلسله مراتبی یا شبکه‌ای دارند . این سیستم‌ها پیش رابطه‌ای را اصطلاحاً سیستم‌های بازمانده[۲۰](میراثی) از گذشته می‌نامند . تبدیل این سیستم‌های رابطه‌ای یا شیء-رابطه‌ای از مشکلات محیط‌های عملیاتی است . از این‌رو آگاهی از طرز تبدیل طرّاحی سلسله مراتبی و / یا شبکه‌ای به طرّاحی رابطه‌ای بسیار سودمند است .

۳۱-۱۵ :

(a $\{W \to Y , X \to Z\} \to \{(W,X) \to Y\}$

$\{(W,X) \to (Y,Z)\} \Rightarrow \{(W,X) \to Y, (W,X) \to Z\}$

a : درست است .

(b $X \to Y \; and \; Z \subseteq Y \to \{X \to Z\}$

$Z \subseteq Y \;\Rightarrow Y \to Z$

$\begin{cases} X \to Y \\ Y \to Z \end{cases} \Rightarrow \begin{cases} Y \to Z \\ X \to Z \end{cases}$

b : درست است .

(c $\{X \to Y , X \to W , (W,Y) \to Z\} \to \{X \to Z\}$

$\begin{cases} X \to Y \\ X \to W \end{cases} \Rightarrow \{X \to (Y,W)\}$

$\begin{cases} X \to (Y,W) \\ (W,Y) \to Z \end{cases} \Rightarrow X \to Z$

c : درست است .

(d $\{(X,Y) \to Z, Y \to W\} \to \{(X,W) \to Z\}$

---

[۲۰] LEGACY SYSTEM

$$\begin{cases}(X,Y)\to Z\\ Y\to W\end{cases}\Rightarrow (X,Y)\to W$$

d : درست است .

e) $\{X\to Y\,,Y\to Z\}\to\{X\to Y\}$

e : به نظر می‌رسد در این سؤال اشتباه تایپی وجود داشته باشد .

f) $\{X\to Y\,,(X,Y)\to Z\}\to\{X\to Z\}$

$$\begin{cases}X\to Y\\ (X,Y)\to Z\end{cases}\Rightarrow (X,X)\to Z\Rightarrow X\to Z$$

f : درست است .

g) $\{X\to Y\,,Z\to W\}\to\{(X,Z)\to(Y,W)\}$

$$\begin{cases}X\to Y\\ Z\to W\end{cases}\Rightarrow (X,Z)\to(Y,W)$$

g : درست است .

h) $\{(X,Y)\to Z\,,Z\to X\}\to\{X\to Y\}$

$$\begin{cases}Z\to X\\ (X,Y)\to Z\end{cases}\Rightarrow (Z,Y)\to Z$$ بدیهی است

$$\begin{cases}(X,Y)\to X\\ Z\to X\end{cases}\Rightarrow (X,Y)\to X$$ بدیهی است

h : درست نیست .

i) $\{X\to Y\,,Y\to Z\}\to\{X\to(Y,X)\}$

$$\begin{cases}X\to Y\\ Y\to X\end{cases}\Rightarrow X\to Z$$

$$\begin{cases}X\to Y\\ X\to Z\end{cases}\Rightarrow X\to(Y,Z)$$

i : درست است .

j) $\{(X,Y)\to Z\,,Z\to W\}\to\{X\to W\}$

$$\begin{cases}(X,Y)\to Z\\ Z\to W\end{cases}\Rightarrow (X,Y)\to W$$

j : درست نیست .

۱۵-۳۲ :

رابطهٔ داده شده : $R\ (A, B, C, D, E, F, G, H, I, J)$

مفروضات :

$$(A, B) \to C, A \to (D, E), B \to F, F \to (G, H), D \to (I, J)$$

$$A \to (D, E) \Rightarrow (A, B) \to (D, E) \Rightarrow \begin{cases} (A, B) \to D \\ (A, B) \to E \end{cases}$$

$$\begin{cases} B \to F \\ (A, B) \to B \end{cases} \Rightarrow (A, B) \to F$$

$$\begin{cases} (A, B) \to F \\ F \to (G, H) \end{cases} \Rightarrow (A, B) \to (G, H) \Rightarrow \begin{cases} (A, B) \to G \\ (A, B) \to H \end{cases}$$

$$\begin{cases} (A, B) \to D \\ D \to (I, J) \end{cases} \Rightarrow (A, B) \to (I, J) \Rightarrow \begin{cases} (A, B) \to I \\ (A, B) \to J \end{cases}$$

پس : $(A, B)^+ = \{A, B, C, D, E, F, G, H, I, J\}$ و چون $(A, B)$ کاهش ناپذیر است ، پس $(A, B)$ کلید کاندید رابطهٔ $R$ است .

کنجکاوی : آیا این رابطه کلید کاندید دیگری هم دارد؟

۱۵-۳۳ :

رابطهٔ $PCS\ (PRID, COID, SECID)$ را در نظر می‌گیریم . معنای رابطه چنین است : «استاد $h$ درس $c$ را در گروه درسی $n$ تدریس می‌کند» . قواعد معنایی چنین است :

۱- یک استاد بیش از یک درس تدریس می‌کند .

۲- یک درس در بیش از یک گروه ارائه می‌شود .

۳- یک استاد در بیش از یک گروه درسی از یک درس تدریس نمی‌کند .

با توجّه به این قواعد کلیدهای کاندید عبارتند از : $(COID, SPECID)$ و $(PRID, COID)$ .

داریم: $(COID, SPEC) \to PRIFD\ , (PRID, COID) \to SPECID$

این رابطه دو دترمینان دارد که هر دو کلید کاندید هستند ، پس رابطه در $BCNF$ است .

: ۳۴-۱۵

در گفتار سوم مسائل مشابه برای مدلسازی معنایی وجود دارد . همچنین در گفتار چهاردهم مسائل مشابه در موضوع طرّاحی بالا به پایین وجود دارد . اگر طراحی با روش بالا به پایین به درستی انجام شود ، رابطه‌های حاصله معمولاً باید در 3NF باشند . پس در صورت لزوم رابطه‌ها را نرمالتر می‌کنیم.

: ۳۵-۱۵

رابطهٔ داده شده : $R(A, B, C)$

مفروضات : $C \to B, A \to B$

ابتدا رابطه را به دو پرتو $R_1(\underline{A}, B)$ و $R_2(\underline{C}, B)$ تجزیه می‌کنیم . این تجزیه خوب نیست زیرا : $R_1 \bowtie R_2 \neq R$

توجه : صفت مشترک در هیچ یک از دو رابطه کلید نیست و در نتیجه با پیوند $R_1$ و $R_2$ تاپل حشو بروز می‌کند . هر چند که در این تجزیه وابستگی‌های تابعی رابطهٔ $R_1$ حفظ می‌شود . بنابراین یک پرتو دیگر هم لازم است به صورت : $R_3(\underline{A, C})$ و در نتیجه $R = R_1 \bowtie R_2 \bowtie R_3$ .

کنجکاوی : آیا در پیوند $R_1 \bowtie R_2 \bowtie R_3$ تاپل حشو بروز نمی‌کند؟

: ۳۶-۱۵

اولاً : رابطهٔ خواسته شده : $ECM\ (EN, ECN, EMN)$

$EN$ : شمارهٔ کارمند ، $ECN$ شمارهٔ همکار ، $EMN$ : شمارهٔ مدیر بلافصل قواعد معنایی چنین اند :

۱- یک کارمند بیش از یک همکار دارد .

۲- یک کارمند یک مدیر بلافصل دارد .

۳- یک کارمند بیش از یک کارمند تحت مدیریت دارد .

ثانیا : سه ویژگی مهمتر این رابطه :

۱- هر سه صفت از یک میدان مقدار می‌گیرند .

۲- رابطه نمایشگر دو نوع ارتباط با معنای متفاوت است :

نوع رابطهٔ «همکار بودن» و نوع ارتباط «مدیریت» .

۳- در گراف ارجاع رابطه ، چرخه ارجاع وجود دارد .

کنجکاوی : آیا این رابطه ویژگی‌های دیگری دارد؟

ثالثاً : کلید کاندید رابطه $(EN, ECN)$ است و داریم :

$$\begin{cases}(EN, ECN) \rightarrow EMN \\ EN \rightarrow EMN\end{cases}$$

در این رابطه وابستگی تابعی ناتام وجود دارد ، پس در $1NF$ است . این رابطه به دو رابطهٔ زیر تجزیه می‌شود :

$$EC\ (\underline{EN, ECN})$$
$$EM\ (\underline{EN}, EMN)$$

هر دو رابطه در $5NF$ (و نیز در $6NF$) هستند .

نتیجه اینکه در نمایش دو یا بیشتر از دو نوع ارتباط با معانی متفاوت در یک رابطه ، پدیدهٔ نامطلوب اختلاط اطلاعات بروز می‌کند که معمولاً توجیه ناپذیر است و سبب می‌شود تا رابطه در سطح نرمال ضعیفی باشد (حتی در $1NF$ باشد) .

۱۵-۳۷ :

الف :

- حل مسئله با رسم نمودار FDها :

با توجه به قواعد معنایی :

$$\begin{cases}(BKID, AUTID) \rightarrow PUBID \\ BKID \rightarrow AUTID\end{cases} \Rightarrow BKID \rightarrow PUBID$$

نمودار وابستگی‌های تابعی :

رابطه‌های لازم :

$$BOOK\ (\underline{BKID}, \dots)$$
$$AUTHOR\ (AUTID, \dots)$$
$$PUBLISHER\ (PUBID, \dots)$$

فرض می‌کنیم که در این سه رابطه وابستگی تابعی با واسطه وجود ندارد . با این فرض هر سه رابطه در 5NF هستند .

برای نوع ارتباط بین سه نوع موجودیت ، می‌توان یک رابطه سه‌گانی به صورت زیر طراحی کرد :

BAP (BKID, AUTID, PUBID)

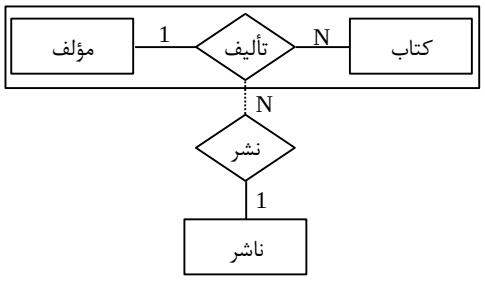
این رابطه هم در 5NF است .

اگر ملاحظات خاصی در ارتباط با کارایی اجرای پرسش‌های خاص مطرح نباشد ، می‌توان رابطهٔ BAP را در رابطه BOOK ادغام کرد :

BOOKAP (BKID, … , AUTID, PUBID)

این رابطه هم در 5NF است .

- حل مسئله با رسم نمودار ER :




رابطه‌های حاصل از تبدیل این نمودار چنین اند :

BOOKAP (BKID, … AUTID, PUBID)
AUTHOR (AUTID, …)
PUBLISHER (PUBID, …)

نتیجه : در این مسئله در طراحی با روش بالا به پایین هم مستقیماً همان رابطه‌های 5NF بدست می‌آیند و نیازی به نرمالترسازی رابطه‌های حاصل نیست . البته در صورت لزوم می‌توان ارتباط بین سه نوع موجودیت را با یک رابطه جدا هم نمایش داد ( با همان رابطهٔ BAP که دیدیم) .

ب : رابطه‌ها باید در نرمالترین فرم باشند زیرا چون یک مؤلف چند کتاب تألیف می‌کند و همچنین یک ناشر چند کتاب منتشر می‌کند ، ادغام رابطه‌های AUTHOR و/یا PUBLISHER با رابطهٔ BOOK طراحی خوبی نیست .

۳۸-۱۵ :

رابطه داده شده : $R\ (A, B, C, D, E)$

$$F = \left\{A \xrightarrow{1} (B, C), B \xrightarrow{2} (C, D), E \xrightarrow{3} (A, D)\right\}$$

$$A \twoheadrightarrow (B, C) \Rightarrow A \xrightarrow{4} (D, E)$$

$$B \twoheadrightarrow (C, D) \Rightarrow B \xrightarrow{5} (A, E)$$

$$E \twoheadrightarrow (A,D) \Rightarrow E \xrightarrow{6} (B,C)$$
$$\begin{cases} A \twoheadrightarrow (B,C) \\ A \twoheadrightarrow (D,E) \end{cases} \Rightarrow A \xrightarrow{7} (B,C,D,E)$$
$$\begin{cases} B \twoheadrightarrow (C,D) \\ B \twoheadrightarrow (A,E) \end{cases} \Rightarrow B \xrightarrow{8} (A,C,D,E)$$
$$\begin{cases} E \twoheadrightarrow (A,D) \\ E \twoheadrightarrow (B,C) \end{cases} \Rightarrow E \xrightarrow{9} (A,B,C,D)$$

پس : $F^+ = \{1,2,3,4,5,6,7,8,9\}$

۱۵-۳۹ :

رابطهٔ داده شده : $R\ (A,B,C,D)$

الف) مفروضات : $D \to A, B \to C$

داریم : $(B,D) \to (A,C) \Rightarrow \begin{cases} (B,D) \to A \\ (B,D) \to C \end{cases}$

پس در این حالت کلید کاندید $(B,D)$ است . تجزیهٔ $R$ به دو رابطهٔ $R_1(B,C)$ و $R_2(A,D)$ خوب نیست زیرا صفت مشترک در هیچیک از دو پرتو کلید کاندید نیست .

ب) مفروضات : $(A,B) \to C, C \to A, C \to D$

$$\begin{cases} C \to A \\ C \to D \end{cases} \Rightarrow C \to (A,D)$$
$$\Rightarrow (A,B) \to (A,D) \Rightarrow (A,B) \to D$$

پس $(A,B)$ کلید کاندید است . تجزیهٔ $R$ به دو پرتو $R_1(A,C,D)$ و $R_2(B,C)$ خوب نیست .

پ) داریم : $C \to (A,D)\ , A \to (B,C)$

$$A \to (B,C) \Rightarrow \begin{cases} A \to B \\ A \to C \end{cases}$$
$$C \to (A,D) \Rightarrow \begin{cases} C \to A \\ C \to D \end{cases} \Rightarrow \begin{cases} A \to B \\ A \to C \\ A \to D \\ C \to A \\ C \to B \\ C \to D \end{cases}$$

پس هر یک از دو صفت $A$ و $D$ کلید کاندید هستند . در نتیجه تجزیه $\begin{cases} R_1(A,B,C) \\ R_2(A,D) \end{cases}$ خوب است . (صفت $A$ در هر دو رابطه کلید کاندید است و وابستگی‌های تابعی نیز حفظ می‌شوند) .

ت) مفروضات : $A \to B, B \to C, C \to D$

$A \to B, B \to C \Rightarrow A \to C$

$$\begin{cases} A \to C \\ C \to D \end{cases} \Rightarrow A \to D$$

پس $A$ کلید کاندید است . تجزیهٔ $R$ به دو پرتو $R_1(A,B)$ و $R_2(A,C,D)$ تجزیهٔ خوبی است .

ث) مفروضات : $A \to B, B \to C, C \to D$

کلید کاندید رابطه صفت $A$ است . تجزیهٔ $R$ به سه پرتو $R_1, R_2, R_3$ خوب نیست .

۱۵-۴۰ :

رابطهٔ داده شده : $R(A,B,C,D,E)$

فرض : $F = \{A \to (B,C), (C,D) \to E, B \to D, E \to A\}$

باید نشان دهیم که تجزیهٔ $R$ به دو پرتو $R_1(A,B,C)$ و $R_2(A,D,E)$ تجزیه خوبی است .

اثبات :

$$A \to (B,C) \Rightarrow \begin{cases} A \to B \\ A \to C \end{cases}$$
$$\begin{cases} A \to B \\ B \to D \end{cases} \Rightarrow A \to D$$
$$\begin{cases} A \to C \\ (C,D) \to E \end{cases} \Rightarrow (A,D) \to E$$
$$\begin{cases} A \to D \\ (A,D) \to E \end{cases} \Rightarrow A \to E$$

یعنی : $A^+ = \{A,B,C,D,E\}$ ، پس صفت $A$ کلید کاندید رابطهٔ $R$ است (البته چون $E \to A$ پس صفت $E$ هم کلید کاندید است) و تمام FDها هم در این تجزیه حفظ می‌شوند . بنابراین تجزیهٔ $R$ به دو پرتو $R_1, R_2$ ، تجزیهٔ خوبی است .

کنجکاوی : آیا این رابطه تجزیهٔ خوب دیگری هم دارد؟

۱۵-۴۱ :

رابطهٔ داده شده : $R(A,B,C,D,E,F,G)$

مفروضات : $F = \{(A,B) \to C, (A,C) \to B, (A,D) \to E, B \to D, (B,C) \to A, E \to G\}$

کلید کاندید این رابطه $(A,B)$ است .

اثبات :

$$\begin{cases}(A,B)\to C\\ B\to D\end{cases}\Rightarrow (A,B)\to (C,D)\Rightarrow (A,B)\to D$$
$$\begin{cases}(A,D)\to E\\ B\to D\end{cases}\Rightarrow (A,B)\to E$$
$$\begin{cases}(A,B)\to E\\ E\to G\end{cases}\Rightarrow (A,B)\to G$$

پس : $(A,B)^+=\{A,B,C,D,E,G\}$

توجّه : به روش دیگر هم می‌توان کلید کاندید را بدست آورد . اجتماع سمت چپ تمام وابستگی‌های تابعی ، سوپرکلید است : $(A,B,C,D,E)$

چون داریم
- $(A,B)\to C$ ، پس $C$ را می‌توان از سوپر کلید حذف کرد .
- $(A,D)\to E$ ، پس $E$ را هم می‌توان از سوپر کلید حذف کرد .
- $B\to D$ ، پس $D$ را هم می‌توان از سوپر کلید حذف کرد .

بنابراین $(A,B)$ کلید کاندید است .

- تجزیهٔ $a$ :

کلید کاندید پرتوها را مشخص می‌کنیم :

$$R_1(\underline{A,B}) \quad , \quad R_2(\underline{B},C) \quad , \quad R_3(\underline{A,B},D,E) \quad , \quad R_4(\underline{E},G)$$

این تجزیه حافظ وابستگی‌های تابعی نیست ، اما بی حشو است .

کنجکاوی : کدامیک از وابستگی‌ها از دست می‌رود؟

- تجزیهٔ $b$ :

کلیدهای کاندید پرتوها را مشخص می‌کنیم :

$$R_1(\underline{A,B},C) \quad , \quad R_2(\underline{A,C,D},E) \quad , \quad R_3(\underline{A,D,G})$$

این تجزیه بی‌حشو نیست و وابستگی‌های تابعی هم حفظ نمی‌شوند .

۴۲-۱۵ :

رابطهای داده شده : $R(A,B,C,D,E)\ ,S(A,B,C,D,E)$

مفروضات : در رابطهٔ $R$ :

$$A \to (B,C), (B,C) \to E, E \to (D,A)$$

در رابطهٔ $S$ :

$$A \to (B,C), B \to E, E \to (D,A)$$

بررسی رابطهٔ $R$ :

$$\begin{cases} A \to (B,C) \\ (B,C) \to E \end{cases} \Rightarrow A \to E$$

$$\begin{cases} A \to E \\ E \to (D,A) \end{cases} \Rightarrow A \to (D,A) \Rightarrow A \to D$$

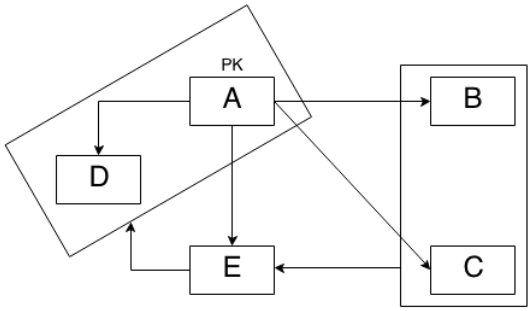
$$A \to (B,C) \Rightarrow \begin{cases} A \to B \\ A \to C \end{cases}$$

پس : $A^+ = \{A,B,C,D,E\}$

$$E \to (D,A) \Rightarrow \begin{cases} E \to D \\ E \to A \end{cases}$$

پس : $E^+ = \{A,B,C,D,E\}$

بنابر این رابطهٔ $R$ دو کلید کاندید ساده دارد : $A$ و $E$




نمودار وابستگی‌های تابعی $R$ با فرض اینکه صفت $A$ را کلید اصلی بگیریم :

چون کلید اصلی رابطه $R$ ساده است ، پس وابستگی تابعی ناتمام ندارد . اما وابستگی تابعی با واسطه دارد :

$$A \to (B,C) \to E \to (D,A)$$

بنابر این رابطهٔ $R$ در $2NF$ است .

بررسی رابطهٔ $S$ :

$$A \to (B,C) \Rightarrow \begin{cases} A \to B \\ A \to C \end{cases}$$

$$\begin{cases} A \to B \\ B \to E \end{cases} \Rightarrow A \to E$$

$$\begin{cases} A \to E \\ E \to (D,A) \Rightarrow \begin{cases} E \to D \\ E \to A \end{cases} \end{cases} \Rightarrow A \to D$$

پس : $A^+ = \{A,B,C,D,E\}$

و داریم : $E^+ = \{A, B, C, D, E\}$

رابطهٔ $S$ هم دو کلید کاندید ساده دارد . پس در $2NF$ هست ، اما چون وابستگی تابعی با واسطه دارد در $3NF$ نیست .

۴۳-۱۵ :

در رابطهٔ $R$ داریم :

$$\{A \to B, (A, B, C, D) \to E, (E, F) \to G, (E, F) \to H, (A, C, D, E) \to (E, G)\}$$

محاسبهٔ مجموعهٔ کاهش ناپذیر :

$$\begin{cases} A \to B \\ (A, B, C, D) \to E \\ (E, F) \to G \\ (E, F) \to H \\ (A, C, D, F) \to E \\ (A, C, D, F) \to G \end{cases}$$

این مجموعه کاهش ناپذیر است .

۴۴-۱۵ :

رابطهٔ داده شده : $EDUC(B\#, C\#, T, Y, J, D\#, P\#, G, F)$

با توجه به قواعد معنایی : $J \to D\#, D\# \to F, P\# \to D\#, S\# \to J$

کلید کاندید این رابطه $(S\#, C\#, T, Y)$ است .

رابطه‌های لازم :

$$SC(\underline{S\#, C\#, T, Y}, P\#, G)$$
$$SJ(\underline{S\#}, J)$$
$$JD(\underline{J}, D\#) \ , \quad DF(\underline{D\#}, F)$$
$$PD(\underline{P\#}, D\#)$$

تمام رابطه‌ها در $5NF$ هستند (چهار رابطهٔ دوگانی در $6NF$ هم هستند) . با توجه به کلید کاندید رابطه‌ها به آسانی می‌توان دریافت که تمام وابستگی‌های تابعی حفظ می‌شوند .

۴۵-۱۵ :

رابطهٔ داده شده : $R(A,B,C)$

باید نشان دهیم : با داشتن $\{A \to B, B \to A, A \to C\}$ لزوما نداریم : $R = JD * (AC, BC)$

با توجه به وابستگی‌های تابعی معلوم می‌شود که $A$ و $B$ ، کلید کاندید هستند . چون صفت مشترک در $R_1(A,C)$ و $R_2(B,C)$ صفت $C$ است و این صفت در هیچیک از دو پرتو کلید کاندید نیست ، بنابراین از پیوند این دو پرتو لزوماً رابطهٔ $R$ به دست نمی‌آید (ممکن است به ازاءِ مقدار خاصّی از بدنهٔ رابطه $R = JD * (AC, BC)$ برقرار باشد) .

: ۴۶-۱۵

رابطهٔ $R(A,B,C)$ با صفات ساده یا مرکّب را در نظر می‌گیریم . فرض می‌کنیم که $R$ در $5NF$ باشد اما در $4NF$ نباشد . در این صورت داریم : $A \to\to, A \to\to C$ یعنی در رابطهٔ $R$ دو وابستگی چندمقداری وجود دارد . پس به ازاءِ یک مقدار مشخص از $(A,C)$ ، مثلا $(a_1, c_1)$ حداقل دومقدار متمایز از $B$ مثلاً $\{b_1, b_2\}$ وجود دارد . همچنین به ازاء یک مقدار مشخص از $(A,B)$ ، حداقل دو مقدار متمایز از $C$ ، مثلاً $(c_1, c_2)$ وجود دارد . پس در بدنهٔ $R$ داریم :

در این صورت در رابطهٔ $R$ محدودیت با ماهیت چرخشی وجود ندارد (رجوع شود به تعریف محدودیت با ماهیت چرخشی) . پس $R$ در $5NF$ نیست و این خلاف فرض است .

R ( A , B , C )

| A | B | C |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_1$ | $b_1$ | $c_2$ |
| $a_1$ | $b_2$ | $c_1$ |
| $a_1$ | $b_2$ | $c_2$ |
| ⋮ | ⋮ | ⋮ |

: ۴۷-۱۵

فرض می‌کنیم که رابطهٔ $R$ در $4NF$ باشد ولی در $BCNF$ نباشد . اگر $R$ در $4NF$ باشد ، یعنی وابستگی چندمقداری در آن وجود ندارد . در این صورت :

- یا $R$ تمام کلید است ، پس در $BCNF$ است .
- یا $X \subset H_R$ کلید کاندید $R$ است ، در این صورت : $X \to H_R - X$ .

  حال می‌گوییم چنین رابطه‌ای نمی‌تواند دترمینان دیگری داشته باشد ، زیرا :

  - یا $Y \to X$ پس : $Y \to H_R - X$ ، پس $Y$ هم باید کلید کاندید باشد .
  - یا $Y \to H_R - X$ ، که در این صورت $(Y, X)$ باید کلید کاندید باشد و این با فرض اینکه $X$ کلید کاندید است ، تعارض دارد .

پس رابطه $4NF$ در $BCNF$ هم هست .

: ۴۸-۱۵

رابطهٔ داده شده : $R(C,T,X,A)$

مفروضات :

$$R = JD * (R_1, R_2, R_3) \begin{vmatrix} R_1(C,T) \\ R_2(T,X,A) \\ R_3(C,X,A) \end{vmatrix}$$

بر می‌نهیم : $(X,A) = Z$ ، داریم : $R = R_1(C,T) \bowtie R_2(T,Z) \bowtie R_3(C,Z)$

در این رابطه محدودیت با ماهیت چرخشی وجود دارد .

یعنی اگر در $R$ داشته باشیم : $\langle c_1, \dots, \overbrace{(x_1, a_1)}^{z_1} \rangle, \langle c_1, t_1, \dots \rangle, \langle \dots, t_1, \overbrace{(x_1, a_1)}^{z_1} \rangle$ آنگاه باید داشته باشیم : $\langle c_1, t_1, \overbrace{(x_1, a_1)}^{z_1} \rangle$

R

| | | Z | |
|---|---|---|---|
| C | T | X | A |
| $c_1$ | $t_1$ | $x_2$ | $a_2$ |
| $c_1$ | $t_2$ | $x_1$ | $a_1$ |
| $c_2$ | $t_1$ | $x_1$ | $a_1$ |
| $c_1$ | $t_1$ | $x_1$ | $a_1$ |

یعنی با داشتن سه فقره اطلاع دو موجودیّتی $\langle c_1, t_1 \rangle$ و$\langle t_1, z_1 \rangle$ و $\langle c_1, z_1 \rangle$ باید استنتاج کنیم : $\langle c_1, t_1, z_1 \rangle$ . به بیان دیگر پدیده دام پیوندی حلقه‌ای در اینجا مطرح نیست . برای درک بهتر مطلب به مثالی از بدنهٔ رابطهٔ $R$ توجّه شود .

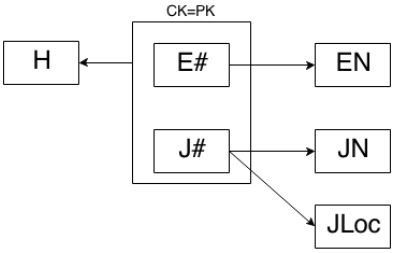
این رابطه به دلیل داشتن محدودیّت با ماهیّت چرخشی در $5NF$ نیست ، بلکه در $4NF$ است .

۴۹-۱۵ :

رابطهٔ داده شده : $EMP(\underline{E\#, J\#}, H, EN, JN, JLoc)$ — $CK$

قواعد معنایی : یک پروژه در یک مکان اجرا می‌شود . در یک مکان تعدادی پروژه اجرا می‌شود .

نمودار وابستگی‌های تابعی :




این رابطه در $1NF$ است .

نرمالترین رابطه‌ها :

$$EJH\left(\underline{E\#, J\#}, H\right) \quad , \quad ENAME\left(\underline{E\#}, JN\right) \quad , \quad JNJ(\underline{J\#}, JN, JLoc)$$

این تجزیه خوب است زیرا شرایط قضّیه ریسانن را دارد . همهٔ رابطه‌ها در $5NF$ هستند .

کنجکاوی : آیا این رابطه‌ها در 6NF هستند؟

۱۵-۵۰ :

رابطهٔ داده شده : $CSCG(\underline{COID, STID}, COTIT, GR)$

کلیدهای کاندید : $(STID, COID)$ و $(STID, COTIT)$

پاسخ قسمت‌های اولاً ، ثانیاً و ثالثاً در این مسأله همانست که در مسئلهٔ ۴ آمده است .

رابعاً : رابطهٔ جدید : $CSCGTY(COID, STID, COTIT, TR, YR, GR)$

الف) با دستور ALTER TABE انجام می‌شود .

ب) این گسترش به دو صورت ایستا و پویا می‌تواند انجام شود . در حالت ایستا فایل متناظر با رابطه (جدول) دوباره تعریف می‌شود و داده‌های ذخیره شده به فایل متناظر با جدول جدید منتقل می‌شوند و فایل متناظر با جدول قبلی حذف می‌شود . در حالت پویا رکوردهای فایل موجود متناظر گسترش داده می‌شوند به این ترتیب که هر بار سطری به منظور انجام عمل بهنگام سازی بازیابی می‌شود ، سطر جدید با دو ستون اضافه شده درج می‌شود ، حتی اگر مقدار دو ستون در سطر هیچمقدار باشد .

پ) چون ممکن است دانشجو در یک ترم در یک درس مردود شود ، بنابراین کلید رابطه $(STID, COID, TR, YR)$ است .

$CSCGTY(\underline{COID, STID, TR, YR}, COTIT, GR)$

این رابطه کلید کاندید دیگری هم دارد : $(STID, COTIT, TR, YR)$

و داریم : $COID \rightarrow COTIT$

پس این رابطه در $BCNF$ هم نیست ، چون $COID$ دترمینان است اما کلید کاندید نیست .

با توجه به تعریف زانیولو از $3NF$ ، این رابطه در $3NF$ است .

این رابطه دو طرز تجزیه دارد :

$$I\begin{cases} SCTYG(\underline{STID, COID, TR, YR}, GR) \\ CT(\underline{COID}, COTIT) \end{cases}$$

$$II\begin{cases} STTYG(\underline{STID, COTIT, TR, YR}, GR) \\ CT(\underline{COID}, COTIT) \end{cases}$$

کنجکاوی : آیا این تجزیه‌ها خوب هستند؟

۱۵-۵۱ :

رابطهٔ داده شده : $R(A,B,C)$ فرض : $A \twoheadrightarrow B$

تاپل‌های داده شده : $\langle 1,2,3 \rangle$ ، $\langle 1,4,5 \rangle$ و $\langle 1,6,7 \rangle$

اولاً : چون وابستگی چندمقداری در رابطه‌های با سه صفت همیشه جفت است ، پس داریم : $A \twoheadrightarrow C$

بنابراین باید مجموعه‌های خوش تعریف مطرح شده در تعریف مفهوم وابستگی‌های تابعی ، در این رابطه وجود داشته باشند . پس تاپل‌های دیگر چنین اند :

| R | A | B | C |
|---|---|---|---|
| | 1 | $\begin{Bmatrix}2\\4\\6\end{Bmatrix}$ | $\begin{Bmatrix}3\\5\\7\end{Bmatrix}$ |

$\langle 1,2,5 \rangle, \langle 1,2,7 \rangle$
$\langle 1,4,3 \rangle, \langle 1,4,7 \rangle$
$\langle 1,6,3 \rangle, \langle 1,6,5 \rangle$

ثانیاً : طبق قضیه فاگین این رابطه به صورت زیر تجزیه می‌شود :

$$R_1(A,B)\ , R_2(A,C)$$

پس یک $JD$ دارد : $R = JD * (R_1, R_2)$

ثالثاً : رابطه‌ی $R$ تمام کلید است و در $BCNF$ است . چنین رابطه‌ای یک وابستگی پیوندی به دو پرتوش دارد که از کلید کاندید رابطه ناشی نمی‌شود . یعنی رابطه $BCNF$ که $5NF$ نباشد ، را می‌توان با مفهوم وابستگی پیوندی به دو پرتو (و نه سه پرتو) تجزیه کرد به نحوی که پرتوها سوپر کلید نیستند . رابطه‌های حاصله در 5NF هستند .

۱۵-۵۳ :

شماره استاد — نام — مرتبه — شمارهٔ گروه آموزشی

$$PROF(\underline{\overbrace{PRID}}, \overbrace{PRNAME}, \overbrace{PRRANK}, \overbrace{PRDEID})$$

این رابطه یک کلید کاندید ساده دارد و وابستگی تابعی با واسطه هم ندارد ، پس حداقل در 3NF است .

$$PRID \rightarrow PRNAME \quad , \quad PRID \rightarrow PRRANK \quad , \quad PRID \rightarrow PRDEID$$

با توجه به روش سریع تشخیص 5NF ، این رابطه در $5NF$ است . اما افزونگی دارد .

توجّه : این رابطه افزونگی دارد ، ولی این افزونگی از نوع افزونگی طبیعی است : تعدادی استاد عضو یک گروه آموزشی هستند ؛ مرتبهٔ تعدادی از آنها می‌تواند یکسان باشد و ممکن است نام مشابه هم داشته باشند .

۵۴-۱۵ :

رابطه‌های داده شده : $R_1(X,Y)\ ,R_2(X,Z)$

معنای $R_1$ : پدر $X$ ، $Y$ است .

معنای $R_2$ : مادر $X$ ، $Z$ است .

در هر دو رابطه $X$ کلید کاندید است .

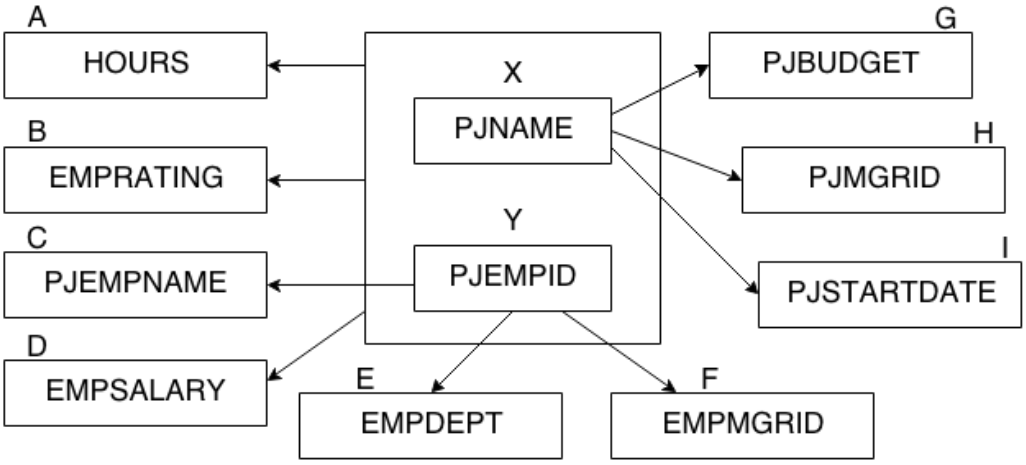
بحث در مورد این طراحی :

در هر دو رابطه داریم : $X \to Y$ و $X \to Z$ ، هر دو رابطه در $6NF$ هستند .

- اگر بخواهیم پدر و مادر $X$ را بشناسیم ، باید عمل پیوند انجام شود . بنابراین اگر بسامد این درخواست بالا باشد ، باید دو رابطه را ادغام کنیم : $R(\underline{X},Y,Z)$ . البته اگر برای نمونه‌هایی از $X$ مشخصات پدر یا مادر معلوم نباشد (پدیده‌ای که حداقّل از نظر رویّه‌های سازمان ثبت احوال ، در پاره‌ای از کشورها ممکن است وجود داشته باشد) ، در این صورت در رابطهٔ $R$ هیچمقدار بروز می‌کند .
- اگر اساساً به جای دو رابطهٔ $R_1$ و $R_2$ یک رابطه به صورت $R_3(X,P)$ طراحی کنیم با کلید کاندید $(X,P)$ که در آن $P$ مثلاً کد ملّی پدر یا مادر باشد ، در این صورت چون $(X,P)$ تنها کلید کاندید رابطه است ، پس می‌شود کلید اصلی . حال اگر کد ملّی پدر یا مادر نامعلوم باشد ، رابطه بدون کلید اصلی می‌شود که حداقل از لحاظ تئوری اصلاً رابطه نیست .

۵۵-۱۵ :

در رابطهٔ داده شده ، تنها کلید کاندید ، $(PJNAME,PJEMPID)$ است . با توجه به قواعد معنایی داده شده ، نمودار وابستگی‌های تابعی چنین است :




برای آسانی در نوشتن عنوان رابطه ، صفات بالا را با یک کاراکتر مشخص کرده‌ایم .

رابطه‌های $3NF$ :

$R_1(\underline{X,Y},A,B,D)$ , $R_2(\underline{Y},C,E,F)$ , $R_3(\underline{X},G,H,I)$

کنجکاوی : آیا با این طراحی می‌توان به پرسش زیر پاسخ داد :

«نمره '۴' را به کارمند 'e' در پروژه 'y' ، کدام مدیر داده است؟ »

اگر بلی ، چگونه؟ اگر نه ، چه باید کرد؟

۵۶-۱۵ :

رابطهٔ داده شده :

$$PROF(PRNAME, PRDEID, ROOMNUM, RANK, DATBEG)$$

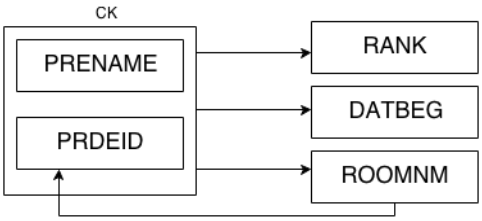
با توجه به قواعد معنایی ، کلید کاندید رابطه : $(PRNAME, PRDDEID)$ است .

و داریم : $ROOMNUM \rightarrow PREDEID$

چون نام استاد یکتا نیست ، نداریم :

$$PRNAME \begin{cases} \rightarrow ROOMNUM \\ \rightarrow RANK \\ \rightarrow DATBEG \end{cases}$$

نمودار وابستگی‌های تابعی چنین است :




این رابطه در $BCNF$ نیست ، زیرا صفت $ROOMNUM$ دترمینان است اما کلید نیست . در این رابطه وابستگی تابعی ناتمام وجود دارد ، پس در $2NF$ است . وابستگی تابعی با واسطه ندارد ، پس در $3NF$ است .

کنجکاوی : آیا با تعریف زانیولو از 3NF هم این رابطه در $3NF$ است؟

۱۵-۵۷ :

رابطهٔ داده شده : $R(A, B, C, D, E)$

مفروضات : $\{A \to B, (B, C) \to D, D \to (B, C), C \to A)$

اولاً :

$$\begin{cases} A \to B \\ (B, C) \to D \end{cases} \Rightarrow (A, C) \to D$$

$$\begin{cases} (A, C) \to D \\ C \to A \end{cases} \Rightarrow C \to D$$

$$\begin{cases} C \to D \\ D \to (B, C) \end{cases} \Rightarrow C \to B$$

پس $(C, E)$ کلید کاندید است .

$$D \to (B, C) \Rightarrow \begin{cases} D \to B \\ D \to C \end{cases}$$

پس $(D, E)$ هم کلید کاندید است .

ثانیاً : پرتوهای تجزیه : $R_1(A, B, E)$ و $R_2(B, C, D)$ هستند . این تجزیه خوب نیست ، زیرا شرایط قضیهٔ ریسانن را ندارد .

کنجکاوی : کدامیک از وابستگی‌های تابعی رابطهٔ $R$ در این تجزیه از دست می‌روند؟

۱۵-۵۸ :

رابطهٔ $R(A, B)$ را در نظر می‌گیریم . چهار حالت متصور است :

$A$ کلید کاندید باشد ، پس داریم $A \to B$ و وابستگی تابعی با واسطه نداریم ، پس $R$ در $3NF$ است .

$B$ کلید کاندید باشد ، پس داریم : $B \to A$ ، پس $R$ در $3NF$ است .

هم $A$ و هم $B$ کلید کاندید باشند ، پس داریم : $A \rightleftarrows B$ ، باز هم رابطه در $3NF$ است .

$(A, B)$ کلید کاندید باشند . در این صورت رابطه $R$ تمام کلید است و حداقل در $BCNF$ است ، پس در $3NF$ هم هست .

اگر مجموعه صفات : $\{\}$ را در نظر نگیریم ، با توجه به استدلال بالا ، رابطهٔ دوگانی همیشه در $BCNF$ هست (می‌تواند در $4NF$ و $5NF$ و $6NF$ هم باشد . )

۱۵-۵۹ :

شرط $BCNF$ بودن این است که رابطه دترمینان غیرکلید کاندید نداشته باشد . امّا اگر رابطه $BCNF$ در $4NF$ نباشد ، معنایش این است که پدیدهٔ وابستگی چندمقداری در آن وجود دارد (بین سه صفت یا سه گروه از صفات به نحوی که اجتماع سه گروه صفات همان عنوان رابطه باشد) . وجود وابستگی چندمقداری ، طبق تعریف ، ایجاب می‌کند که رابطه تمام-کلید باشد .

کنجکاوی : آیا می‌توان با برهان خلف هم ثابت کرد؟

۱۵-۶۰ :

رابطهٔ $R$ در $2NF$ است هر گاه به ازاءِ هر صفت $A$ از $R$ یکی از دو شرط زیر برقرار باشد :

۱- $A$ صفت عمده باشد .

۲- اگر $X$ تنها کلید کاندید $R$ باشد ، وابستگی تابعی $Y \to A | Y \subset X$ برقرار نباشد .

۱۵-۶۱ :

رابطه‌های $R_1(A, B, C, D, E)$ و $R_2(F, G, H, I, J, K)$ را در نظر می‌گیریم . اگر قرار باشد که معنای یک پرتو از دو رابطه یکسان باشد ، باید که صفات آن دو پرتو ، مثلاً دو به دو ، از یک میدان (دامنه) باشند ، مثلا :

میدان صفات $D_2 : H, C$ میدان صفات $D_1 : A, F$

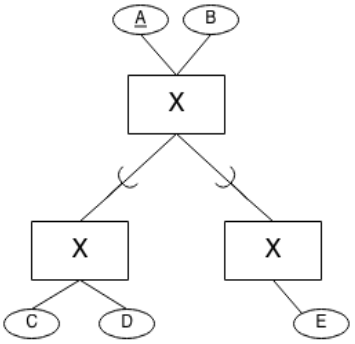
در این صورت معنای پرتوهای $R_{1_1}(A, F)$ و $R_{2_1}(H, C)$ یکسان است . توجه داریم که صفت از میدانش می‌گیرد : معنا را ، نوع را و طیف مقادیر را و نه لزوماً نام را .

به این طرز طراحی اصطلاحاً طراحی نامستقل یا با اشتراک معنا گویند .

توجه داشته باشیم هم میدان بودن صفات $A, F$ و نیز صفات $C, H$ لزوماً منجر به بروز افزونگی نمی‌شوند .

این طرز طراحی در شرایطی می‌تواند خوب باشد .

- مثال : نمودار زیر را در نظر بگیرید :




اگر این تخصیص کامل باشد ، می‌توان با دو رابطه طراحی را انجام داد :

$$YR(\underline{A}, B, C, D) \quad , \quad ZR(\underline{A}, B, E)$$

در اینجا معنای دو پرتو $\Pi_{\langle A,B \rangle}(YR)$ و $\Pi_{\langle A,B \rangle}(ZR)$ یکسان است . توجه داشته باشیم که در اینجا صفات هم میدان از دو رابطه ، همنام هم هستند .

- مثال دیگر از وضعی که این طرز طراحی می‌تواند پذیرفتنی باشد : رابطهٔ زیر را در نظر می‌گیریم :

$$R(\underline{W, X}, Y, Z, T, U, V)$$

فرض می‌کنیم کلید کاندید این رابطه $(W, X)$ باشد (یا فقط $W$) و بسامد مراجعه به صفات $W, X, Z$ بسیار بالا باشد . در این صورت $R$ را به دو رابطه به صورت زیر تجزیه عمودی می‌کنیم :

$$R_1(\underline{W, X}, Z) \quad , \quad R_2(\underline{W, X}, Y, T, U, V)$$

در اینجا هم دو پرتو $\Pi_{\langle A,B \rangle}(YR)$ و $\Pi_{\langle A,B \rangle}(ZR)$ هم معنا هستند .

- مثال دیگر ، وضعی است که در آن یک رابطه را تجزیه افقی می‌کنیم . در این حالت $N \geq 2$ رابطه داریم که نوع سازگارند (عنوان همهٔ آنها یکسان است) یعنی هم‌معنا هستند .

کنجکاوی : آیا وضعیت‌های دیگری هم وجود دارد که در آن طراحی نامستقل پذیرفتنی باشد؟

۶۲-۱۵ :

فرض : $R_i \ (i = 1,2, \dots n)$ پرتوهای دلخواه از رابطهٔ $R$ هستند . می‌خواهیم ثابت کنیم :

$$R \subseteq R_1 \bowtie R_2 \bowtie R_3 \bowtie \cdots \bowtie R_n$$

ابتدا برای $n = 2$ ثابت می‌کنیم : $R \subseteq R_1 \bowtie R_2$

چند حالت متصور است :

- اگر $H_{R_1} \cap H_{R_2} = CK_R$ ، آنگاه طبق تعریف $CK$ داریم : $R = R_1 \bowtie R_2$
- اگر $H_{R_1} \cap H_{R_2} = A$ در $R_1$ کلید کاندید باشد ، در این صورت در $R_2$ کلید خارجی است . پس به ازاءِ یک تاپل از $R_1$ با مقدار مشخص $A$ ، حداقل دو تاپل در $R_2$ با همان مقدار $A$ وجود دارد . پس $R \subset R_1 \bowtie R_2$
- اگر $H_{R_1} \cap H_{R_2} = A$ و $A$ نه در $R_1$ کلید کاندید باشد و نه در $R_2$ ، در این صورت به طریق اولی داریم : $R \subset R_1 \bowtie R_2$

حال می‌گوئیم : فرض می‌کنیم که حکم برای $n-1$ ثابت باشد یعنی :

$$R' \subseteq R_1 \bowtie R_2 \bowtie R_3 \bowtie \cdots \bowtie R_{n-1}$$

پس در پیوند $R' \bowtie R_n$ هم یکی از حالات بالا پیش می‌آید ، بنابراین با همان استدلال ، حکم ثابت است . یعنی :

$$R \subseteq R_1 \bowtie R_2 \bowtie R_3 \bowtie \cdots \bowtie R_n$$

۱۵-۶۳ :

فرض : $X$ مجموعه‌ای است از صفات رابطهٔ $R$ .

می‌خواهیم ثابت کنیم : $(X^+)^+ = X^+$

فرض کنیم :

$$X \to \{A_1, A_2, \dots, A_i, A_j, A_k, \dots, A_m\}$$
$$X^+ = \{A_1, A_2, \dots, A_i, A_j, A_k, \dots, A_m\}$$

در این صورت وجود دارد $m$ وابستگی تابعی که در آنها $A_i$ ، $(i = 1, \dots, m)$ در سمت راست است . حال اگر $(X^+)^+ \neq X^+$باشد ، در این صورت یا $(X^+)^+ \subset X^+$ است یا $X^+ \subset (X^+)^+$ .

در حالت اوّل ، وجود دارد حداقل یک صفت ، مثلاً $Z$ ، به نحوی که :

$$X^+ \to Z \Rightarrow X^+ \to \{A_1, A_2, \dots, A_i, A_j, A_k, \dots, A_m, Z\}$$

و این خلاف فرض است .

در حالت دوم ، وجود دارد حداقل یک صفت ، مثلاً $A_j \in X^+$ به نحوی که :

$$(X^+)^+ = X^+ - A_j = \{A_1, A_2, \dots, A_i, A_k, \dots, A_m, Z\}$$

و این هم خلاف فرض است .

$X^+ \to \{A_1, A_2, \dots, A_i, A_k, \dots, A_m, Z\}$
$X^+ \to X^+ - A_j$

و این هم خلاف فرض است . پس حکم ثابت است .

۱۵-۶۴ :

اثبات قواعد آرمسترانگ

۱ : قاعدهٔ انعکاس : اگر $B \subseteq A$ آنگاه : $A \to B$

فرض کنیم در رابطهٔ $R$ دو تاپل $t_1$ و $t_2$ وجود داشته باشند به نحوی که : $t_1(A) = t_2(A)$

در این صورت با توجه به فرض : $t_1(B) = t_2(B)$

پس : $A \to B$ باید در $R$ برقرار باشد .

۲ : قاعدهٔ تعدّی : اگر $A \to B$ و $B \to C$ آنگاه $A \to C$ .

فرض کنیم در رابطهٔ $R$ داشته باشیم : $A \to B$ و $B \to C$ ، در این صورت به ازاءِ هر دو تاپل $t_1$ و $t_2$ در $R$ به نحوی که : $t_1(A) = t_2(A)$ ، باید داشته باشیم : $t_1(B) = t_2(B)$ و چون باید داشته باشیم : $t_1(C) = t_2(C)$ پس $A \to C$ باید در $R$ برقرار باشد .

توجه : اثبات از طریق برهات خلف در متن گفتار ۱۵ آمده است .

۳ : قاعدهٔ افزایش : اگر $A \to B$ آنگاه $(A, C) \to (B, C)$

از برهان خلف استفاده می‌کنیم .

فرض کنیم در رابطهٔ $R$ داریم : $A \to B$ ولی $(A, C) \to (B, C)$ برقرار نباشد . در این صورت باید دو تاپل $t_1$ و $t_2$ در $R$ وجود داشته باشند به نحوی که :

1) $t_1(A) = t_2(A)$
2) $t_1(B) = t_2(B)$
3) $t_1(A, C) = t_2(A, C)$
4) $t_1(B, C) \neq t_2(B, C)$

این وضع ناممکن است زیرا از (1) و (3) داریم :

1) $t_1(C) = t_2(C)$

که از (2) و (5) نتیجه می‌شود :

$$4)\ t_1(B,C) = t_2(B,C)$$

که با (4) تناقض دارد .

۴ : قاعدهٔ تجزیه : اگر $A \to (B,C)$ آنگاه $A \to B$ و $A \to C$

با استفاده از قاعدهٔ ۱ و ۲ داریم :

فرض : $A \to (B,C)$

و چون $B \subseteq (B,C)$ ، داریم : $(B,C) \to B$

پس : $A \to B$

و داریم : $(B,C) \to C$

$$\begin{cases} A \to (B,C) \\ (B,C) \to C \end{cases} \Rightarrow A \to C$$

۵ : قاعدهٔ اجتماع : اگر $A \to B$ و $A \to C$ آنگاه $A \to (B,C)$

داریم : $A \to B$ و $C \to D$ ، پس : $(A,A) \to (A,B) \Rightarrow A \to (A,B)$

چون $A \to C$ ، پس داریم : $A \to (A,B,C)$

و از آنجا : $\begin{cases} A \to A \\ A \to (B,C) \end{cases}$

راه دیگر : $\begin{matrix} A \to C \Rightarrow (A,B) \to (B,C) \\ A \to (A,B) \end{matrix} \Rightarrow A \to (B,C)$

۶ : قاعدهٔ ترکیب : اگر $A \to B$ و $C \to D$ آنگاه $(A,C) \to (B,D)$

$$\left.\begin{matrix} A \to B \Rightarrow (A,C) \to (B,C) \\ C \to D \Rightarrow (A,C) \to (C,D) \end{matrix}\right\} \Rightarrow (A,C) \to (B,C,D) \Rightarrow \begin{cases} (A,C) \to C \\ (A,C) \to (B,D) \end{cases}$$

۷ : قاعدهٔ شبه تعدّی : اگر $A \to B$ و $(B,C) \to D$ آنگاه $(A,C) \to D$

داریم :

$A \to B, (B, C) \to D$
$A \to B \Rightarrow (A, C) \to (B, C)$
$\begin{cases}(A, C) \to (B, C) \\ (B, C) \to D\end{cases} \Rightarrow (A, C) \to D$

۸ : قاعدهٔ یگانگی عمومی : در متن گفتار ۱۵ اثبات شده‌است .

توجه : تمرین‌های ۶۵ تا ۸۰ اختیاری در نظر گرفته شوند .

۷۱-۱۵ :

این موضوع وقتی ممکن است پیش آید که بخواهیم رابطه‌ای را به صورت حافظ وابستگی‌های تابعی تجزیه کنیم .

- مثال : در رابطهٔ $R = (\underline{A, B}, C)$ داریم : $(A, B) \to C$ و فرض کنیم : $C \to A$ ، این رابطه در $BCNF$ نیست . اگر حتماً بخواهیم آنرا نرمالتر کنیم ، باید چنان تجزیه کنیم که تجزیه حافظ وابستگی‌ها باشد ، یعنی چنین تجزیه کنیم : $R_1(\underline{A, B}), R_2(\underline{C}, A), R_3(\underline{A, B}, C)$ می‌بینیم که تعداد پرتوها بیش از حدّ لازم است . در این وضعیت بهتر است اصلاً رابطه تجزیه نشود .
- به پاسخ تمرین ۲۵ هم توجّه شود .

۷۲-۱۵ :

در حالت خاص ممکن است . یعنی رابطهٔ $R$ حاصلضرب کارتزین مثلاً دو رابطهٔ $R_1$ و $R_2$ باشد و $H_{R_1} \cap H_{R_2} = \phi$ .
در این صورت داریم : $R_1 \bowtie R_2 = R_1 \times R_2$

پس اگر رابطهٔ $R$ را به دو رابطهٔ $R_1$ و $R_2$ تجزیه کنیم ، این دو رابطه هیچ صفت مشترک نخواهند داشت . البته چنین تجزیه‌ای می‌تواند با هدف نرمالترسازی انجام شود .

۷۳-۱۵ :

وابستگی پیوندی ادغام شده (EJD) :

رابطهٔ $R$ دارای وابستگی پیوندی ادغام شده است هرگاه :

در یک پرتو درجهٔ $n > 2$ از $R$ وابستگی پیوندی وجود داشته باشد .

در یک زیرمجموعهٔ افقی از $R$ وابستگی پیوندی وجود داشته باشد .

مثال :

R

| A | B | C | D |
|---|---|---|---|
| $a_1$ | $b_1$ | $c_2$ | $d_1$ |
| $a_1$ | $b_2$ | $c_1$ | $d_7$ |
| $a_2$ | $b_1$ | $c_1$ | $d_1$ |
| $a_1$ | $b_1$ | $c_1$ | $d_2$ |

$R_1 = \Pi_{\langle A,B,C\rangle}(\mathrm{R}) =$

| A | B | C |
|---|---|---|
| $a_1$ | $b_1$ | $c_2$ |
| $a_1$ | $b_2$ | $c_1$ |
| $a_2$ | $b_1$ | $c_1$ |
| $a_1$ | $b_1$ | $c_1$ |

رابطهٔ $R_1$ وابستگی پیوندی به سه پرتوش دارد :

$$R_1 = JD * \left(R_{1_1}(A,B), R_{1_2}(B,C), R_{1_3}(A,C)\right)$$

مثال :

$R_1 = \Theta_{(A=a_1)\vee(A_{a_2})}(R) =$

| A | B | C |
|---|---|---|
| $a_1$ | $b_1$ | $c_2$ |
| $a_1$ | $b_2$ | $c_1$ |
| $a_2$ | $b_1$ | $c_1$ |
| $a_1$ | $b_1$ | $c_1$ |

$R =$

| A | B | C |
|---|---|---|
| $a_1$ | $b_1$ | $c_2$ |
| $a_1$ | $b_2$ | $c_1$ |
| $a_2$ | $b_1$ | $c_1$ |
| $a_1$ | $b_1$ | $c_1$ |
| $a_3$ | $b_2$ | $c_2$ |
| $a_7$ | $b_6$ | $c_5$ |

رابطهٔ $R$ وابستگی پیوندی به سه پرتوش دارد .

۷۴-۱۵ :

این "اِظهار" محدودیَت با ماهیت چرخشی را اِعلان می‌کند ، یعنی :

$$R = JD * (R_1(A,b), R_2(B,C), R_3(A,C))$$

کنجکاوی : چگونه؟

: ۷۵-۱۵

این "اظهار" محدودیت زیر را اعلان می‌کند :

$$(A, B) \rightarrow (C, D)$$

پس : $(A, B) \rightarrow C$ , $(A, B) \rightarrow D$ . بنابراین $(A, B)$ کلید کاندید رابطهٔ $R$ در $5NF$ است .

(می‌توان رابطهٔ $R$ را به دو رابطهٔ $R_1(\underline{A, B}, C)$ و $R_2(\underline{A, B}, D)$ تجزیه کرد ، رابطه‌های $R_1$ و $R_2$ در $6NF$ هستند . )

: ۷۶-۱۵

دلیل این است که بیشتر قواعد آرمسترانگ را می‌توان حالت خاص این قواعد دانست . بررسی را به خواننده وا می‌گذاریم .

: ۷۷-۱۵

رابطهٔ زیر را در نظر می‌گیریم :

$$R(A, B, C, D, \dots)$$

اگر $A$ کلید اصلی رابطهٔ $R$ باشد ، برای اینکه در $2NF$ نباشد ، صفت $A$ باید مرکّب باشد . حال می‌خواهیم ببینیم در چه وضعی این رابطهٔ $1NF$ به دو رابطه تجزیه می‌شود :

هرگاه داشته باشیم :

$$A \rightarrow B \; and \; X \rightarrow B \mid X \subset A \qquad\qquad A \rightarrow C \; and \; Y \rightarrow C \mid Y \subset A$$

در این صورت $R$ به سه رابطه تجزیه می‌شود . به بیان دیگر اگر حداقل دو وابستگی تابعی ناتمام در $R$ وجود داشته باشد به نحوی که سمت راست وابستگی تابعی ایجاد کنندهٔ وابستگی نام در آن وابستگی‌های تابعی ناتمام ، متمایز باشند و نیز سمت چپ آنها هم متمایز باشند ، رابطهٔ $1NF$ به حداقل سه رابطه تجزیه می‌شود .

: ۷۸-۱۵

می‌توان چنین طرّاحی را انجام داد ، امّا لازمه‌اش اینست که در رابطه(های) غیرنرمال عملیات ذخیره‌سازی (درج ، حذف و بهنگام‌سازی) انجام نشود ، بلکه فقط عملیات بازیابی انجام شود .

# گفتار ۱۶

## کنجکاوی‌ها

۱۶-۱ : می‌توان نکات زیر را هم در نظر داشت :

- نوع پرسش از نظر نوع سیستم پردازش : پرسش BATCH[۲۱] ، پرسش OLAP[۲۲] ، پرسش DSS [۲۳]
- عملگرهای استفاده‌شده در پرسش
- کاردینالیتی رابطه‌های دخیل در پرسش
- درجه‌ی رابطه‌های دخیل در پرسش

و برخی نکات دیگر . . .

۱۶-۲ : روی این گونه صفات نباید شاخص ایجاد کرد ، زیرا با تغییر مقادیر آنها ، ساختار شاخص باید متناسباً بازسازی شود ( بطور ایستا یا پویا ) و این کار سربار (فزونکاری) دارد .

۱۶-۳ :

- تعیین فایل‌هایی که باید شاخص داشته باشند .
- تعیین صفات شاخص
- نوع شاخص

۱۶-۴ : اگر درجه رابطه کوچک باشد ، می‌توان تاپل‌ها رابطه را در گره‌های درخت شاخص ذخیره کرد ، در واقع جدول را با ساختار شاخص ایجاد کرد .

۱۶-۵ : وقتی که بخواهیم تاپل‌هایی را بازیابی کنیم که مقدار صفت جستجو در آنها یکسان باشد .

۱۶-۶ : می‌توان هدف‌ها و مزایای زیر را هم برشمرد :

---

[۲۱] BATCH QUERY
[۲۲] ONLINE ANALYTIC QUERY
[۲۳] DECISION SUPPORT QUERY

- طولانی‌تر کردن زمان بین دو خرابی در سیستم
- استفاده بهینه از امکانات سیستم مدیریت
- تسهیل و تسریع تشخیص اشکال‌ها در سیستم پایگاهی

۱۶-۷ : هیچمقدار معایبی دارد از جمله :

- افزایش حافظه‌ی هرز
- دشواری در انجام عملیات روی داده‌ها ( یادآوری : هیچمقدار ، یک « مقدار ناشناخته » است . )
- ایجاد تاپل ( های ) آونگان ( معلق ) در پیوند پرتوهای رابطه .

از این میان ، بویژه عیب اول مستقیماً در کارایی سیستم تأثیر دارد .

کنجکاوی : آیا عیب (های) دیگری هم مطرح است ؟

۱۶-۸ : افزایش همروندی از طریق بخش‌بندی داده‌ی مشترک ( مثلاً تجزیه‌ی عمودی و / یا افقی یک رابطه بزرگ به تعدادی رابطه‌ی کوچکتر ، وقتی که سیستم تنها قفل در سطح رابطه را می‌پذیرد . البته این تکنیک وقتی مؤثر است که هر یک از تراکنش‌ها به یک بخش جدا از بخش‌های دیگر ، نیاز داشته باشد .

۱۶-۹ :

- میانای (واسط) کاربر پسند داشته باشد .
- امکان اندرکنش با کاربر را فراهم سازد .
- اطلاعات جامع از کاربر دریافت کند .
- فراروند مدلسازی و طراحی را مرحله به مرحله به پیش ببرد .
- قادر به اِعمال اصلاحات مدلساز - طراح باشد .
- ترجیحاً « هوشمند » باشد .
- تعارض‌های مرحله‌ی ادغام نمودارهای جریی به منظور تهیه نمودار جامع را تشخیص دهد .
- همه مفاهیم موجود در روش مدلسازی معنایی را پشتیبانی کند .

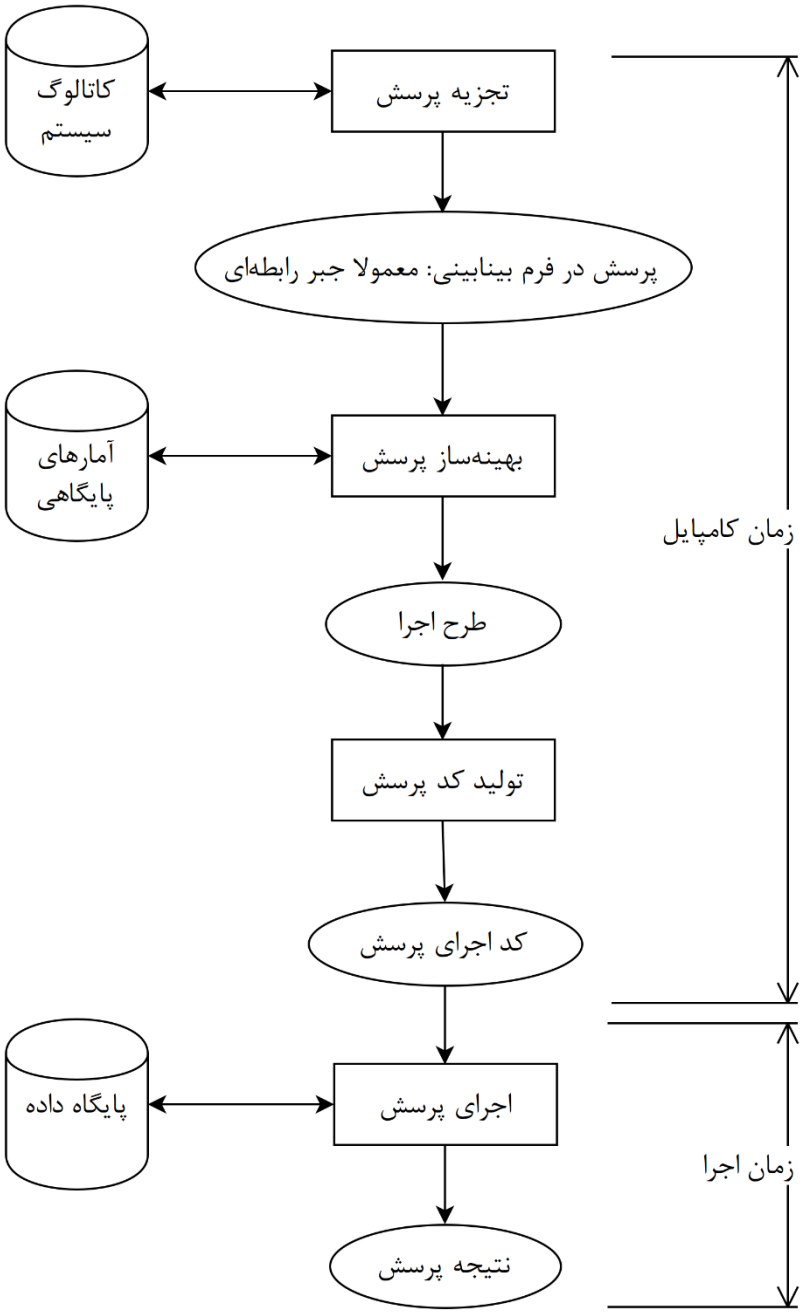
- افزونگی‌های معنایی موجود در اطلاعات ارائه‌شده توسط کاربر را تشخیص دهد .
- . . .

ضمناً به پاسخ تمرین‌های پایان همین گفتار توجه شود .

۱۶-۱۰ : بیان دلایل دیگر به خواننده واگذار می‌شود .

# تمرینات درون گفتار

۱۶-۱ : در نمودار زیر فازهای پردازش و بهینه‌سازی پرسش دیده می‌شوند :



۲-۱۶ :

۱- تامین محیط دخیره‌سازی کاراتر :

رابطه‌ی $R(\underline{A}, B, C, D, E)$ را در نظر می‌گیریم. اگر بسامد دستیابی به صفات A و D و E بالا باشد و صفات B و C با بسامد پائین مورد دستیابی قرار بگیرند، بهتر است رابطه‌ی R به دو رابطه‌ی زیر تجزیه عمودی شود.

$R_1(\underline{A}, D, E)$

$R_2(\underline{A}, B, C)$

توجّه : فرض بر این است که تناظر بین رابطه‌ها و فایل یک به یک باشد (که در عمل معمولاً چنین است).

با این تجزیه سیستم برای رابطه‌ی $R_1$ فایل جداگانه با رکوردهای کوتاه‌تر ایجاد و آن را با لوکالیتی قوی‌تر ذخیره می‌کند و در نتیجه کارایی سیستم در پردازش آن بیشتر می‌شود.

البته ممکن است به همین دلیل، یعنی افزایش کارایی فایلینگ پایگاه داده، در مواردی تجزیه افقی هم انجام بشود.

۲- معمولاً در سیستم‌های موجود، تعداد ستون‌های جدول (صفات رابطه) یک حداکثر مشخصی دارد که باید در طراحی جدول‌ها در نظر داشت. اگر تعداد ستون‌های جدول از حداکثر مجاز درگذرد، باید جدول را تجزیه‌ی عمودی کرد.

۳- اگر یک رابطه مبنا R را به دو یا بیش از دو رابطه‌ی مبنای جدید $R_1$ و $R_2$ تجزیه عمودی کنیم، برای معتبر ماندن تعریف، دیدهای قبلاً تعریف شده روی R، باید رابطه‌ی R را به صورت یک رابطه‌ی مجازی روی $R_1$ و $R_2$ تعریف کنیم. (به گفتار سیزدهم، قسمت ۳-۱-۱ مراجعه شود). در این وضعیت جدید کاربران اساساً از وجود رابطه‌های مبنای جدید $R_1$ و $R_2$ بی‌اطلاع‌اند. همین بی‌اطلاعی از ساختار داده‌ها در لایه زیرین، سبب افزایش بیش‌تر ایمنی پایگاه داده‌ها می‌شود (علاوه بر مفهوم دید که خود نوعی مکانیسم خودکار ایمنی است) .

۴- برای نرمال‌ترسازی رابطه‌ها و ارائه طراحی بهتر، رابطه‌ها را به‌طور عمودی تجزیه می‌کنیم.

۵- در سیستم‌های پایگاهی توزیع شده نیاز به طرح توزی-تخصیص داده‌ها داریم. در طرح توزیع باید چگونگی تجزیه عمودی و/یا افقی و گاه ترکیبی و هم‌چنین تولید نسخه(های) تکراری را مشخص کنیم.

۶- در کنترل همروندی، اگر واحد قفل شدنی (دانگی قفل) تاپل (سطر) باشد، چنان‌چه درجه رابطه (تعداد صفات) زیاد باشد، و تعدادی تراکنش بخواهند در سطح تاپل (سطر) قفل داشته باشند و هریک به بعضی از صفات نیاز داشته باشند، بهتر است رابطه را تجزیه عمودی کنیم تا بتوان به بیش از یک تراکنش قفل داد و بدین‌ترتیب همروندی را افزایش داد.

حال اگر واحد قفل شدنی رابطه باشد، باز هم با تجزیه عمودی یا افقی آن، در شرایطی می‌توان میزان همروندی را افزایش داد.

۷- کاهش حجم هیچ‌مقدار

تجزیه عمودی رابطه می‌تواند سبب کاهش هیچ‌مقدار شود. برای مثال رابطه‌ی $R$ را در نظر می‌گیریم :

| R | (A, | B, | C, | D) |
|---|---|---|---|---|
| | a1 | b1 | c1 | d1 |
| | a2 | b1 | ? | d2 |
| | a3 | b3 | ? | ? |
| | a4 | b2 | ? | ? |
| | a5 | b5 | ? | ? |

اگر این رابطه را به دو رابطه‌ی $R_1$ و $R_2$ تجزیه‌ی عمودی کنیم به صورت زیر :

| R1 | (A, | B) |
|---|---|---|
| | a1 | b1 |
| | a2 | b1 |
| | a3 | b3 |
| | a4 | b2 |
| | a5 | b5 |

| R2 | (C, | D) |
|---|---|---|
| | c1 | d1 |
| | ? | d2 |

میزان هیچ‌مقدار کاهش می‌یابد.

کنجکاوی : با این طرز تجزیه چگونه باید رابطه‌ی □ را بازسازی کرد ؟

# تمرینات پایان گفتار

۱۶–۱ : کارهای مرحله‌ی ۱ ، مرحله‌ی ۲ و مرحله‌ی ۳ . البته در مرحله‌ی طرّاحی منطقی باید به محدودیت‌ها و امکانات سیستم مدیریّت پایگاه داده‌ها هم توجّه داشت .

۱۶–۲ : نیاز(های) داده‌ای و پردازشی جدید کاربران ، محدودیت(های) عملیاتی در دیدهای موجود ، اینکه دید کاربر که رابطه‌ای مجازی است ، آیا باید تبدیل به دید ذخیره شده شود ... ؛ در هر صورت کارایی اجرای برنامه‌های کاربردی مهم‌ترین عاملی است که باید همیشه در نظر داشت .

۱۶–۴ :

- ارائه پیشنهادهای مدلسازی معنایی داده‌ها
- ارائه پیشنهادهای طراحی منطقی
- تولید شمای پایگاه داده‌ها
- اِعمال اصلاحات و تغییرات پیشنهادی کاربر
- توصیه‌ی پیشنهاد برتر

۱۶–۶ ، ۷ و ۸ : به متون درس «مهندسی نرم افزار» مراجعه شود .

۱۶–۹ : برخی از عوامل عبارتند از :

- بسامد اجرای پرسش‌ها
- بسامد درخواست صفات
- وضع مسند(های) داده شده در پرسش‌ها
- میزان همروندی مورد انتظار در اجرای تراکنش‌ها
- ماهیت عملیات (بازیابی، درج، حذف، بهنگام‌سازی) و نرخ این عملیات
- ماهیت عملیات از نظر حجم داده‌ی دخیل : تک تاپلی ، تعدادی تاپل ، تمام رابطه ، تعدادی رابطه ...
- سرعت اجرای مورد انتظار
- عملگرهای استفاده شده در پرسش
- ماهیّت پرسش از نظر سیستم پردازش : سیستم برون خط ، سیستم برخط ، سیستم پشتیبانی تصمیم

- حجم داده مورد تظر در پاسخ پرسش‌ها : تک تاپل ، گروهی از تاپل‌ها ...
- نوع ستون‌ها : عدد صحیح ، کاراکتری ...
- امکانات سیستم در تأمین ساختار فایل‌ها
- امکانات سیستم در تأمین استراتژی‌های دستیابی
- ...

۱۶–۱۰ : به مباحث مربوط در مهندسی فایل‌ها مراجعه شود .

۱۶–۱۱ :

۹. Q : مشخصات کامل همه دانشجویان مهمان را بدهید .

۱۰. Q : مشخصات استادان با رتبه «دانشیار» را بدهید .

۱۱. Q : شماره درس‌هایی که دانشجوی شماره «۸۸۸۸۰۰۷۷» در آن‌ها مردود شده است، را بدهید .

۱۲. Q : شماره و نام دانشجویان دوره «کارشناسی» رشته‌ی فیزیک را بدهید .

نوشتن این پرسش‌ها در SQL منطقاً نباید دشوار باشد .

۱۳. Q :

```
SELECT STNAME
     FROM   STT
          WHERE STID IN
               (SELECT STID
                    FROM   COT
                         WHERE TR = 1 AND YRYR = ’91-92’
                              AND GRADE < 10
                              AND COID   IN
                                   (SELECT COID
                                   FROM COT
                                   WHERE COTYPE = ’h’));
```

این پرسش را می‌توان به پرسشهای زیر تبدیل کرد :

```
INSERT INTO T1
     SELECT COID
          FROM  COT
               WHERE COTYPE = ’h’ ;
INSERT INTO T2
     SELECT STID
          FROM   STCOT JOIN T1 ;
INSERT INTO T3
```

```
SELECT STNAME
      FROM   STT JOIN T2 ;
```

توجه : در اینجا پرسش سه سطحی به سه پرسش تک سطحی تبدیل شده‌است ، اما کاراتر بودن اجرای این سه پرسش نسبت به پرسش سه سطحی بستگی دارد به الگوریتمهایی که برپایه‌ی آنها سیستم عملگرها ، از جمله عملگر JOIN ، را اجرا می‌کند .

۱۴. Q :

```
SELECT EMPID
      FROM PROF
            WHERE RANK ≠ ’e1’
                  OR
                  RANK ≠ ’e2’ ;
```

چون میدان مقادیر صفت RANK چهارمقدار دارد : $\{e1, e2, e3, e4\}$ ، می‌توان این پرسش را تبدیل به پرسش زیر کرد :

```
SELECT EMPID
      FROM PROF
            WHERE RANK = ’e3’ OR RANK = ’e4’ ;
```

۱۵. Q : به پاسخ تمرین ۱۶ از گفتار پانزدهم مراجعه شود .

۱۶–۱۴ :

Q : شماره و عنوان درس‌هایی را بدهید که در ترم ۲ سال تحصیلی ۹۲ توسط استادان با مرتبه مربّی تدریس می‌شوند .

باید از رابطه‌های COUR ، COSEC و PROF استفاده شود . (به پیوستار ۱ مراجعه شود) .

```
SELECT COID, COTITLE
      FROM COUR JOIN COSEC JOIN PROF
            WHERE TR = ’2’ AND YR = ’92’
                  AND RANK = ‘instrcuture’ ;
```

نوشتن این پاسخ با عملگر نیم‌پیوند و با سور وجودی آسان است . توصیه می‌شود که روی سیستمی که می‌شناسید ، اجرا کنید.

۱۶– ۱۶ :

الف : ساختار شاخص‌دار با شاخص اصلی روی DID و شاخص روّم روی DEID .

ب : باید دید چه استراتژی‌های دیگری در سیستم وجود دارد . اگر امکان ذخیره‌سازی بیش از یک جدول مبنا در یک فایل دجود داشته باشد ، می‌توان دو جدول را در یک فایل ذخیره کرد و از سیستم درخواست کرد تا آن فایل را با لوکالیتی قوی ذخیره کند .

طرح دیگر این است که دو فایل متناظر با دو جدول را با لوکالیتی قوی بین فایل‌ها ذخیره کنیم .

البته امکان تغییر طراحی هم وجود دارد ، امّا در سوال تأکید روی طراحی فیزیکی است .

۱۶–۱۷ : اساساً ایجاد و مدیریت فایل با رکوردهای با طول متغیر دشوار است و سربار زیادی دارد . لازمه‌ی متغیر بودن طول تاپل این است که طول مقادیر حداقل یک صفت (اندازه‌ی ستون) متغیر باشد .

از جمله دلایل این ترجیح :

- سهولت پیاده‌سازی جدول
- سهولت اِعمال تغییرات در جدول
- ایجاد شاخص روی ستون با طول متغیر دشوار است و فزونکاری دارد .
- مصرف بهتر حافظه

برای اطلاع در این باره می‌توان به منابع مهندسی فایل ، از جمله [روحا ۸۶-الف] مراجعه کرد .