


به نام خداوند بخشنده مهربان

|   |   |
|---|---|
| <p>تمرین اول درس اصول و طراحی پایگاه داده<br/>دکتر سعیده ممتازی<br/>ترم بهار 1400 – دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر<br/>زمان تحویل: 17 فروردین 1400</p> |  |
|---|---|

**لطفا قبل از شروع به حل کردن تمرین به نکات زیر توجه فرمایید:**

1. در صورت وجود هرگونه سوال یا ابهام می‌توانید با تدریسار از طریق ایمیل [farbod.rsi@gmail.com](mailto:farbod.rsi@gmail.com) در ارتباط باشید.
2. مهلت تحویل تمرین تا تاریخ 17 فروردین ماه می‌باشد. **بعد از این تاریخ تمرین به هیچ عنوان تحویل گرفته نمی‌شود.**
3. تمامی فایل‌های خواسته شده را در قالب یک فایل فشرده با نام HW1-StudentNumber در مدل بارگذاری کنید. به عنوان مثال: HW1-9631032.

۱- به سوالات زیر پاسخ دهید:

- A. وجه شباهت و تمایز محدودیت‌های Unique و Primary key را شرح دهید.
- B. برای هر یک از دستورات Data Manipulation Language (DML) و Data Definition Language (DDL) یک نمونه پرس‌وجو بنویسید.
- C. وجه شباهت و تمایز دستورات drop و truncate و delete را شرح دهید.
- D. سیستم مدیریت پایگاه داده (Database Management System) هنگام شاخص‌گذاری (indexing) چه تغییر یا تغییراتی بر روی داده ایجاد می‌کند و چه تاثیری دارد؟
- E. ۳ مورد از مزایای استفاده از دیتابیس نسبت به فایل سیستم را نام ببرید.

۲- جداول زیر را برای لیست انتخاب واحد دانشجویان در نظر بگیرید:

STUDENTS (id, name, entrance\_year, field, cellphone, gender)  
COURSES (id, name, unit)  
PROFESSORS (id, name, national\_id, gender)  
AVAILABLE\_COURSES (id, course\_id, professor\_id, time, year, semester)  
TAKEN\_COURSE (id, student\_id, available\_course\_id)

توضیحات:

field: رشته تحصیلی

unit: تعداد واحد درس مربوطه

available\_courses: دروس ارائه شده

year: سال ارائه واحد

semester: ترم ارائه واحد (برای ترم پاییز عدد ۱ و ترم زمستان عدد ۲)

با استفاده از جبر رابطه‌ای پرس و جوهای خواسته شده را بنویسید.

- A. لیست تمامی دروس ارائه شده در ترم های فرد بین سال های ۱۳۹۰ تا ۱۳۹۴.
- B. لیست دانشجویان ورودی ۱۳۹۸ که در انتخاب واحد ترم پاییز سال ۱۳۹۹ کمتر از ۱۲ واحد اخذ کرده‌اند.
- C. لیست دانشجویان پسر که هیچ درسی با استادی با نام Ali اخذ نکرده‌اند.

D. لیست تمامی دانشجویان دختر که تاکنون کمتر از ۸۰ واحد اخذ کرده اند.  
E. لیست نام تمامی اساتیدی که در ترم فرد درس 3 واحدی یا در ترم زوج درس 2 واحدی ارائه کرده اند.

۳- فرض کنید که مدیریت پایگاه داده یک شرکت ارسال بسته اینترنتی (مانند اسنپ باکس) را بر عهده دارید. این شرکت تعدادی کاربر و پیک موتوری دارد که هر سفارش توسط یک کاربر ایجاد می‌شود و یک پیک آن را قبول می‌کند.  
در ابتدا لازم است که پایگاه داده را با جداولی که در ادامه شرح داده می‌شوند بسازید و پرس‌وجوهای مربوطه را در فایلی با نام 3.sql بنویسید.

#### جدول کاربران (user)

- Id: شناسه یکتا عددی برای هر کاربر
- name: نام کاربری
- cellphone: شماره تلفن
- referred\_by: شناسه یکتا کاربری که ایشان را دعوت کرده (میتواند خالی باشد)
- credit: مقدار اعتبار کاربر به تومان (نمی‌تواند کمتر از ۰ باشد)

#### جدول پیک ها (driver)

- Id: شناسه یکتا عددی هر پیک
- name: نام کاربری
- cellphone: شماره تلفن
- address: آدرس محل سکونت
- is\_active: (۰ به معنای غیر فعال و ۱ به معنای فعال)
- score: میانگین امتیاز پیک

#### جدول سفارشات (order)

- Id: شناسه یکتا هر سفارش
- user\_id: شناسه کاربری که سفارش را ایجاد کرده
- driver\_id: شناسه پیکی که سفارش را قبول کرده (اگر هیچ پیکی سفارش را قبول نکرده باشد این فیلد خالی است)
- estimated\_value: ارزش تخمینی بسته ارسالی به تومان.

- price: هزینه سفارش
- is\_round\_trip: سفارش رفت و برگشتی
- start\_time: تاریخ و زمان آغاز سفارش
- finish\_time: تاریخ و زمان پایان سفارش
- score: امتیاز کاربر به پیک (بین ۱ تا ۵)

#### جدول افزایش اعتبار کاربران (payment)

- Id: شناسه یکتا هر پرداخت
- user\_id: شناسه کاربر
- amount: مبلغ افزایش اعتبار به تومان (باید مضربی از ۵۰۰ باشد)
- bank: نام بانک (باید یکی از بانک های [saderat, saman, sina, melli] باشد)
- time: تاریخ و زمان

#### وارد کردن اطلاعات

لازم است که برای هر یک از جداول ساخته شده ۴ یا ۵ سطر نمونه وارد کنید و پرس و جوهای مربوطه را در فایل مربوط به همین سوال بنویسید.

#### گزارش گیری

برای موارد خواسته شده، پرس و جو های SQL بنویسید (در فایل مربوط)

- نام کاربر در رکورد اول جدول کاربر را به **farbod** تغییر دهید.
- نام کاربری کاربرانی که هیچ سفارشی ثبت نکرده اند را برگردانید.
- کاربری که هیچ سفارشی نداشته است را از لیست جدول کاربران حذف کنید.
- آیدی تمامی کاربران با نام کاربری **farbod** را برگردانید.
- تمامی شماره تلفن هایی که هم به عنوان کاربر و هم به عنوان پیک ثبت شده اند را برگردانید.
- نام کاربری کاربرانی که حداقل ۲ سفارشی ثبت کرده اند که هیچ پیکی آن هارا قبول نکرده را برگردانید. (فیلد پیک در جدول سفارش خالی است).
- سفارش هایی که روز پایان سفارش فردای روز شروع سفارش بوده است را برگردانید.
- کاربرانی که پیک تکراری داشته اند.
- میانگین زمان سفارش هایی که امتیاز آن ها از ۳ بیشتر است را برگردانید.
- ۵ کاربری که بیشترین میانگین امتیازات را به سفارش ها داده اند را به ترتیب نزولی برگردانید.
- پیکی که در هفته اخیر بیشترین درآمد را داشته است را برگردانید.
- میانگین امتیازی که کاربرانی که حداقل یک بار از بانک سینا استفاده کرده اند به پیک ها داده اند را برگردانید.

M. مجموع قیمت سفارش هایی که توسط کاربرهایی که توسط کاربری با نام کاربری farbod معرفی (refer) شده اند را برگردانید.

۴- برای هر کدام از سوالات زیر پرس و جو مربوطه را در فایلی با نام 4.sql بنویسید.  
(توجه داشته باشید که سوالات مجزا هستند)

A.

جدولی از شماره دانشجویی دانشجویان و نمرات درس های آن ها به شکل زیر داریم.

GRADES (id, student\_id, course\_id, grade)

پرس و جویی بنویسید که میانگین نمرات هر درس موجود در جدول را برگرداند.

B.

جداولی از نمره های دروس پایگاه داده و سیستم عامل دانشجویان داریم. (نمرات از ۲۰ هستند)

DB\_GRADES (id, student\_id, grade)

OS\_GRADES (id, student\_id, grade)

توجه داشته باشید که لزوما دانشجویان دو کلاس یکسان نیستند و ممکن است دانشجویی فقط یکی از این دو درس را داشته باشد.

پرس و جویی بنویسید که شماره دانشجویی دانشجویانی که در درس پایگاه داده نمره قبولی کسب کرده اند درحالی که در درس سیستم عامل مردود نشده اند را برگرداند. (اگر درس سیستم عامل را اخذ نکرده باشند هم مشمول می شوند).

C.

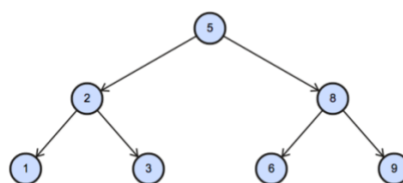
جدولی به نام BST داریم که نماینگر یک درخت جستجوی باینری می باشد و به ازای هر گره، والد آن گره را مشخص می کند.

برای نمونه داریم

| N | P    |
|---|------|
| 1 | 2    |
| 3 | 2    |
| 6 | 8    |
| 9 | 8    |
| 2 | 5    |
| 8 | 5    |
| 5 | null |

سطر اول نشان می‌دهد که گره ۲، والد گره ۱ است.

درخت متناظر جدول بالا به صورت زیر است.



پرس‌وجویی بنویسید که نقش هر گره را از بین نقش‌های Leaf، Inner و یا Root مشخص کند.

برای نمونه داده شده خروجی به صورت زیر خواهد بود.

```

1 Leaf
2 Inner
3 Leaf
5 Root
6 Leaf
8 Inner
9 Leaf
  
```

. D

جدولی شامل ۲ ستون به نام‌های X و Y داریم (x و y هر دو عدد صحیح هستند)


Table1 (x, y)

دو سطر  $(X_1, Y_1)$  و  $(X_2, Y_2)$  نسبت به هم متقارن هستند اگر و تنها اگر رابطه‌ی  $X_2 = Y_1$  و  $X_1 = Y_2$  بین آن‌ها برقرار باشد.

پرس‌وجویی بنویسید که تمامی سطرهایی که متقارن آن‌ها نیز در جدول موجود است را برگرداند.

موفق باشید

به نام خداوند بخشنده مهربان

|   |   |
|---|---|
| <p>تمرین اول درس اصول و طراحی پایگاه داده<br/>دکتر سعیده ممتازی<br/>ترم بهار 1400 – دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر<br/>زمان تحویل: 17 فروردین 1400</p> |  |
|---|---|

**لطفا قبل از شروع به حل کردن تمرین به نکات زیر توجه فرمایید:**

1. در صورت وجود هرگونه سوال یا ابهام می‌توانید با تدریسار از طریق ایمیل [farbod.rsi@gmail.com](mailto:farbod.rsi@gmail.com) در ارتباط باشید.
2. مهلت تحویل تمرین تا تاریخ 17 فروردین ماه می‌باشد. **بعد از این تاریخ تمرین به هیچ عنوان تحویل گرفته نمی‌شود.**
3. تمامی فایل‌های خواسته شده را در قالب یک فایل فشرده با نام HW1-StudentNumber در مدل بارگذاری کنید. به عنوان مثال: HW1-9631032.

۱- به سوالات زیر پاسخ دهید:

- A. وجه شباهت و تمایز محدودیت‌های Unique و Primary key را شرح دهید.
- B. برای هر یک از دستورات Data Manipulation Language (DML) و Data Definition Language (DDL) یک نمونه پرس‌وجو بنویسید.
- C. وجه شباهت و تمایز دستورات drop و truncate و delete را شرح دهید.
- D. سیستم مدیریت پایگاه داده (Database Management System) هنگام شاخص‌گذاری (indexing) چه تغییر یا تغییراتی بر روی داده ایجاد می‌کند و چه تاثیری دارد؟
- E. ۳ مورد از مزایای استفاده از دیتابیس نسبت به فایل سیستم را نام ببرید.

A

| Primary key                                       | Unique key   |
|---|--|
| مشخص کننده هر سطر از جدول                         | مشخص کننده هر سطر جدول در صورتی که کلید اصلی موجود نباشد |
| نمی‌تواند مقدار null داشته باشد                   | می‌تواند مقدار null داشته باشد                           |
| حداکثر یک کلید اصلی می‌تواند در هر جدول حاضر باشد | محدودیتی روی تعداد ندارد                                 |

B

DML:

```
INSERT INTO students (first_name, last_name, age) VALUES ('farbod', 'rasaei', 22);
```

DDL:

```
CREATE TABLE students (  
  id BIGINT PRIMARY KEY,  
  first_name VARCHAR (50) not null,  
  last_name VARCHAR (75) not null,  
  age INTEGER (50)  
);
```



C

هر ۳ دستور برای پاک کردن داده های جدول یا خود جدول هستند اما:

drop یک دستور DDL می باشد که باعث پاک شدن تمامی سطرهای ذخیره شده و همچنین دسترسی ها، ایندکس ها، محدودیت ها و تمامی مواردی که برای آن جدول ساخته شده اند می شود.

truncate نیز یک دستور DDL می باشد که تمامی سطرهای ذخیره شده در جدول را پاک می کند اما schema جدول باقی می ماند، در mysql اجازه truncate هنگامی که کلیدهای خارجی به سطرهایی از جدول اشاره می کنند وجود ندارد.

delete یک دستور DML می باشد که تنها سطرهای انتخاب شده در پرسمان (نه لزوماً تمامی سطرها) را پاک می کند. و برخلاف مورد بالا به راحتی قابل بازگردانی است.

D

شاخص گذاری یا ایندکسینگ برای افزایش سرعت دسترسی به داده های ذخیره شده صورت می گیرد، به این صورت که تمامی داده ها در ساختمان داده هایی که order پایین تری هنگام دسترسی (مانند b-trees) نیز ذخیره می شوند.

شاخص گذاری باعث افزایش هزینه عمل insert نیز می شود زیرا هر داده جدید باید هم به صورت عادی و هم در ساختمان داده هایی برای دسترسی سریع ذخیره شود.

E

قابلیت جستجو برای داده ای خاص با استفاده از query language (مانند sql).

قابلیت انجام چند عمل به صورت atomic.

رابطه ای بودن جداول و استفاده همزمان از چند جدول با توجه به نیاز.

## ۲- جداول زیر را برای لیست انتخاب واحد دانشجویان در نظر بگیرید:

STUDENTS (id, name, entrance\_year, field, cellphone, gender)

COURSES (id, name, unit)

PROFESSORS (id, name, national\_id, gender)

AVAILABLE\_COURSES (id, course\_id, professor\_id, time, year, semester)

TAKEN\_COURSE (id, student\_id, available\_course\_id)

توضیحات:

field: رشته تحصیلی

unit: تعداد واحد درس مربوطه

available\_courses: دروس ارائه شده

سال ارائه واحد: year:

ترم ارائه واحد (برای ترم پاییز عدد ۱ و ترم زمستان عدد ۲): semester:

با استفاده از جبر رابطه‌ای پرس و جوهای خواسته شده را بنویسید.

- A. لیست تمامی دروس ارائه شده در ترم های فرد بین سال های ۱۳۹۰ تا ۱۳۹۴.
- B. لیست دانشجویان ورودی ۱۳۹۸ که در انتخاب واحد ترم پاییز سال ۱۳۹۹ کمتر از ۱۲ واحد اخذ کرده‌اند.
- C. لیست دانشجویان پسر که هیچ درسی با استادی با نام Ali اخذ نکرده‌اند.
- D. لیست تمامی دانشجویان دختر که تاکنون کمتر از ۸۰ واحد اخذ کرده‌اند.
- E. لیست نام تمامی اساتیدی که در ترم فرد درس 3 واحدی یا در ترم زوج درس 2 واحدی ارائه کرده‌اند.

با توجه نبودن group by در مطالب این ترم جبر رابطه‌ای، موارد B و D امتیازی هستند.

A-

$\sigma_{\text{year} > 1390} \wedge \text{year} < 1394 \wedge \text{semester} = 1$  (AVAILABLE\_COURSES)

C-

$(\pi_{\text{STUDENT.gender}=1} (\text{STUDENTS} \bowtie \text{AVAILABLE\_COURSES} \bowtie \text{TAKEN\_COURSE} \bowtie \text{PROFESSORS})) -$

$(\pi_{\text{STUDENT.gender}=1 \wedge \text{PROFESSORS.name}='Ali'} (\text{STUDENTS} \bowtie \text{AVAILABLE\_COURSES} \bowtie \text{TAKEN\_COURSE} \bowtie \text{PROFESSORS}))$

E-

$\pi_{\text{Professors.name}} (\sigma_{\text{Unit}=3 \wedge \text{semester}=0} (\text{Courses} \bowtie \text{AVAILABLE\_COURSES} \bowtie \text{Professors}) \cup \sigma_{\text{Unit}=2 \wedge \text{semester}=1} (\text{Courses} \bowtie \text{AVAILABLE\_COURSES} \bowtie \text{Professors}))$

۳- فرض کنید که مدیریت پایگاه داده یک شرکت ارسال بسته اینترنتی (مانند اسنپ باکس) را بر عهده دارید. این شرکت تعدادی کاربر و پیک موتوری دارد که هر سفارش توسط یک کاربر ایجاد می‌شود و یک پیک آن را قبول می‌کند. در ابتدا لازم است که پایگاه داده را با جداولی که در ادامه شرح داده می‌شوند بسازید و پرس‌وجوهای مربوطه را در فایل‌ی با نام 3.sql بنویسید.

#### جدول کاربران (user)

- Id: شناسه یکتا عددی برای هر کاربر
- name: نام کاربری
- cellphone: شماره تلفن
- referred\_by: شناسه یکتا کاربری که ایشان را دعوت کرده (میتواند خالی باشد)
- credit: مقدار اعتبار کاربر به تومان (نمی‌تواند کمتر از ۰ باشد)

#### جدول پیک ها (driver)

- Id: شناسه یکتا عددی هر پیک
- name: نام کاربری
- cellphone: شماره تلفن
- address: آدرس محل سکونت
- is\_active: (۰ به معنای غیر فعال و ۱ به معنای فعال)
- score: میانگین امتیاز پیک

#### جدول سفارشات (order)

- Id: شناسه یکتا هر سفارش
- user\_id: شناسه کاربری که سفارش را ایجاد کرده
- driver\_id: شناسه پیکی که سفارش را قبول کرده (اگر هیچ پیکی سفارش را قبول نکرده باشد این فیلد خالی است)
- estimated\_value: ارزش تخمینی بسته ارسالی به تومان.
- price: هزینه سفارش
- is\_round\_trip: سفارش رفت و برگشتی
- start\_time: تاریخ و زمان آغاز سفارش
- finish\_time: تاریخ و زمان پایان سفارش
- score: امتیاز کاربر به پیک (بین ۱ تا ۵)

### جدول افزایش اعتبار کاربران (payment)

- Id: شناسه یکتا هر پرداخت
- user\_id: شناسه کاربر
- amount: مبلغ افزایش اعتبار به تومان (باید مضربی از ۵۰۰ باشد)
- bank: نام بانک (باید یکی از بانک های [saderat, saman, sina, melli] باشد)
- time: تاریخ و زمان

### وارد کردن اطلاعات

لازم است که برای هر یک از جداول ساخته شده ۴ یا ۵ سطر نمونه وارد کنید و پرس و جوهای مربوطه را در فایل مربوط به همین سوال بنویسید.

### گزارش گیری

برای موارد خواسته شده، پرس و جو های SQL بنویسید (در فایل مربوط)

- A. نام کاربر در رکورد اول جدول کاربر را به **farbod** تغییر دهید.
- B. نام کاربری کاربرانی که هیچ سفارشی ثبت نکرده اند را برگردانید.
- C. کاربری که هیچ سفارشی نداشته است را از لیست جدول کاربران حذف کنید.
- D. آیدی تمامی کاربران با نام کاربری **farbod** را برگردانید.
- E. تمامی شماره تلفن هایی که هم به عنوان کاربر و هم به عنوان پیک ثبت شده اند را برگردانید.
- F. نام کاربری کاربرانی که حداقل ۲ سفارشی ثبت کرده اند که هیچ پیکی آن هارا قبول نکرده را برگردانید. (فیلد پیک در جدول سفارش خالی است).
- G. سفارش هایی که روز پایان سفارش فردای روز شروع سفارش بوده است را برگردانید.
- H. کاربرانی که پیک تکراری داشته اند.
- I. میانگین زمان سفارش هایی که امتیاز آن ها از ۳ بیشتر است را برگردانید.
- J. ۵ کاربری که بیشترین میانگین امتیازات را به سفارش ها داده اند را به ترتیب نزولی برگردانید.
- K. پیکی که در هفته اخیر بیشترین درآمد را داشته است را برگردانید.
- L. میانگین امتیازی که کاربرانی که حداقل یک بار از بانک سینا استفاده کرده اند به پیک ها داده اند را برگردانید.
- M. مجموع قیمت سفارش هایی که توسط کاربرهایی که توسط کاربری با نام کاربری **farbod** معرفی شده اند (refer) برگردانید.

جواب ها در فایل 3.sql

۴- برای هر کدام از سوالات زیر پرسوجو مربوطه را در فایلی با نام 4.sql بنویسید.  
(توجه داشته باشید که سوالات مجزا هستند)

A.

جدولی از شماره دانشجویی دانشجویان و نمرات درس‌های آن‌ها به شکل زیر داریم.

`GRADES (id, student_id, course_id, grade)`

پرسوجویی بنویسید که میانگین نمرات هر درس موجود در جدول را برگرداند.

B.

جداولی از نمره‌های دروس پایگاه داده و سیستم‌عامل دانشجویان داریم. (نمرات از ۲۰ هستند)

`DB_GRADES (id, student_id, grade)`

`OS_GRADES (id, student_id, grade)`

توجه داشته باشید که لزوماً دانشجویان دو کلاس یکسان نیستند و ممکن است دانشجویی فقط یکی از این دو درس را داشته باشد.

پرسوجویی بنویسید که شماره دانشجویی دانشجویانی که در درس پایگاه داده نمره قبولی کسب کرده‌اند درحالی که در درس سیستم‌عامل مردود نشده‌اند را برگرداند. (اگر درس سیستم‌عامل را اخذ نکرده باشند هم مشمول می‌شوند).

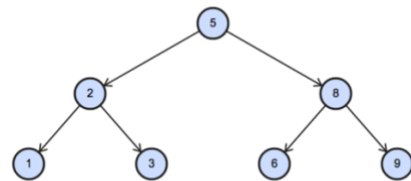
C.

جدولی به نام BST داریم که نماینگر یک درخت جستجوی باینری می‌باشد و به ازای هر گره، والد آن گره را مشخص می‌کند.

برای نمونه داریم

| N | P    |
|---|------|
| 1 | 2    |
| 3 | 2    |
| 6 | 8    |
| 9 | 8    |
| 2 | 5    |
| 8 | 5    |
| 5 | null |

سطر اول نشان می‌دهد که گره ۲، والد گره ۱ است.  
درخت متناظر جدول بالا به صورت زیر است.



پرس‌وجویی بنویسید که نقش هر گره را از بین نقش‌های Leaf، Inner و یا Root مشخص کند.  
برای نمونه داده شده خروجی به صورت زیر خواهد بود.

```
1 Leaf
2 Inner
3 Leaf
5 Root
6 Leaf
8 Inner
9 Leaf
```

. D

جدولی شامل ۲ ستون به نام‌های X و Y داریم (x و y هر دو عدد صحیح هستند)

Table1 (x, y)

دو سطر  $(X_1, Y_1)$  و  $(X_2, Y_2)$  نسبت به هم متقارن هستند اگر و تنها اگر رابطه‌ی  $X_1 = Y_2$  و  $X_2 = Y_1$  بین آن‌ها برقرار باشد.

پرس‌وجویی بنویسید که تمامی سطرهایی که متقارن آن‌ها نیز در جدول موجود است را برگرداند.

جواب‌ها در فایل 4.sql

موفق باشید



```
CREATE TABLE `users`
```

```
(
  `id`          int primary key auto_increment,
  `name`        varchar(255) NOT NULL,
  `cellphone`   varchar(11)  NOT NULL,
  `referred_by` int          null,
  `credit`      int DEFAULT 0 CHECK (`credit` >= 0),
  foreign key (referred_by) references users (id)
);
```

```
INSERT INTO `users` (`name`, `cellphone`, `referred_by`, `credit`)
VALUES ('Farbod', '09123960515', null, 15000),
      ('Alireza', '0919111111', 1, 0),
      ('Arian', '0912943121', null, 1000),
      ('Mohadeshe', '09353131313', 2, 0),
      ('Sara', '09367512596', null, 50000),
      ('Sosan', '09123579546', null, 5000);
```

```
#drivers
```

```
CREATE TABLE `drivers`
```

```
(
  `id`          int primary key auto_increment,
  `name`        varchar(255) NOT NULL,
  `cellphone`   varchar(11)  NOT NULL,
  `address`     varchar(255) NOT NULL,
  `is_active`   bool          NOT NULL,
  `score`       double CHECK (`score` >= 0 and `score` <= 5)
);
```

```
INSERT INTO `drivers` (`name`, `cellphone`, `address`, `is_active`, `score`)
VALUES ('Shayan', '09353654982', 'Tehran-Piroozi', 1, 4.1),
      ('Niusha', '09302548751', 'Tehran-punak', 0, 3),
      ('Abbas', '09015469129', 'Tehran-4rahvaliasr', 1, 1.2),
      ('Fati', '09362158519', 'Tehran-Molasadra', 0, 2.9),
      ('Hassan', '09333562131', 'Ahvaz-Mellat', 1, 0),
      ('Ahmad', '09172356491', 'Sanandaj', 1, 5);
```

```
#
```

```
CREATE TABLE `orders`
```

```
(
  `id`          int primary key auto_increment,
  `user_id`     int          not null,
  `driver_id`   int          default null,
  `estimated_value` int      not null,
  `price`       int          not null,
  `is_round_trip` bool      default 0,
  `start_time`  timestamp NOT NULL DEFAULT current_timestamp(),
  `finish_time` timestamp NOT NULL DEFAULT current_timestamp(),
  `score`       int CHECK (`score` >= 0 and `score` <= 5),
  foreign key (user_id) references users (id),
  foreign key (driver_id) references drivers (id)
);
```

```
INSERT INTO `orders` (`user_id`, `driver_id`, `estimated_value`, `price`, `is_round_trip`, `start_time`, `finish_time`, `score`)
VALUES (1, 2, 150000, 25000, 0, '2021-04-09 06:22:58', '2021-04-09 06:23:58', 5),
      (1, null, 2000000, 35000, 1, '2021-04-09 06:23:33', '2021-04-09 06:23:35', 4),
      (2, 1, 80000, 9500, 0, '2021-04-09 06:24:36', '2021-04-09 06:24:36', 1),
      (3, 4, 35000, 800000, 1, '2021-04-09 06:26:53', '2021-04-09 06:26:53', 2),
      (1, null, 50000, 12000, 0, '2021-04-09 06:27:26', '2021-04-09 06:27:26', 2);
```

```
# payment
```



```

CREATE TABLE `payment`
(
    `ID` int primary key auto_increment,
    `user_id` int not null,
    `amount` int CHECK (`amount` MOD 500 = 0),
    `bank` varchar(127) CHECK (`bank` in ('melli', 'sina', 'saman', 'saderat')),
    `time` timestamp NOT NULL DEFAULT current_timestamp(),
    foreign key (user_id) references users (id)
);

INSERT INTO `payment` (`user_id`, `amount`, `bank`, `time`)
VALUES (1, 1000, 'melli', '2021-04-09 06:54:34'),
(2, 20500, 'saderat', '2021-04-09 06:55:23'),
(3, 15000, 'sina', '2021-04-09 06:56:35'),
(4, 11500, 'saman', '2021-04-09 06:56:58'),
(2, 500, 'melli', '2021-04-09 06:57:22');

#A
update users
set name = 'Farbod'
where id = 1;

#B
SELECT name
FROM users
where id not in (SELECT DISTINCT user_id FROM orders);

#C
DELETE
from users
WHERE ID IN (SELECT id
             FROM users
             where id not in (SELECT DISTINCT user_id FROM orders));

#D
SELECT name
FROM users
WHERE name = 'farbod';

#E
(SELECT cellphone
 FROM users
  NATURAL JOIN drivers
 WHERE users.cellphone = drivers.cellphone);

#F

select name
from users
where id in (select user_id from (SELECT COUNT(id) as count, user_id FROM orders where driver_id is null GROUP BY user_id having count > 1))

#G
#1
SELECT *
FROM orders
where finish_time - start_time < 1000000;

#2
select *
from orders
where TIMESTAMPDIFF(DAY, start_time, finish_time) = 1;

#H

select user_id
from (select count(id) as cnt, user_id, driver_id
      from orders
      where driver_id is not null
      group by user_id, driver_id) as i
where cnt > 1;

```

```
#I
select avg(Finish_time - start_time)
from orders
where score > 3;

#J
select user_id
from (select avg(score) as sc, user_id
      from orders
      group by user_id
      order by sc desc
      limit 5) as s;

#K
select sum(price) as income, driver_id
from orders
where driver_id is not null
      and TIMESTAMPDIFF(DAY, start_time, now()) < 7
group by driver_id
order by income desc
limit 1;

#L
select avg(score)
from orders
where user_id in (select distinct (user_id) from payment where bank = 'sina');

#M
select sum(price)
from orders
where user_id in (select id from users where referred_by in (select id from users where name = 'farbod'))
```

```
-- a
select avg(grade),id from grades group by course_id;

-- b
select id from DB_grades where grade>10 except select id from OS_GRADES where grade <10;


--c

-- 1
select n,if (1=(select count(*) from bst as b where b.n = a.n and b.p is null),'Root',
  if (0=(select count(*) from bst as c where c.p = a.n),'Leaf','Inner')) from bst as a
group by n;

-- 2
-- for root
SELECT n FROM bst where p is null;
-- for leaf
select n from bst where n not in(select p from bst) and p is not null;
-- for inner
select distinct(p) from bst except SELECT n from bst where p is null;

--d
select * from Table1 as t1 join Table1 as t2 where (t1.x1 = t2.y2) and (t1.x2 = t2.y1)
;
```

بسمه تعالی

|  |   |
|--|---|
| <p>تمرین دوم درس اصول و طراحی پایگاه داده<br/>دکتر ممتازی<br/>ترم بهار ۱۴۰۰ - دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر<br/>زمان تحویل: ۱۱ اردیبهشت ۱۴۰۰</p> |  |
|--|---|

### لطفا قبل از شروع به حل کردن تمرین به نکات زیر توجه فرمایید:

۱. در صورت وجود هرگونه سوال یا ابهام می‌توانید با تدریسار از طریق ایمیل [aref78.m@gmail.com](mailto:aref78.m@gmail.com) در ارتباط باشید.
۲. مهلت تحویل تمرین تا تاریخ ۱۱ اردیبهشت ماه می‌باشد. بعد از این تاریخ تمرین به هیچ عنوان تحویل گرفته نمی‌شود.
۳. تمامی فایل‌های خواسته شده را در قالب یک فایل فشرده با نام HW2-StudentNumber در مودل بارگذاری کنید. به عنوان مثال: HW2-9631070.

## نمایشکده

به شما وظیفه توسعه پلتفرم نمایشکده سپرده شده است. این برنامه، یک سرویس پخش و استریم فیلم به صورت آنلاین می‌باشد. در پایین به ترتیب بخش‌های این پروژه را به طور کامل توضیح خواهیم داد.

در این سیستم، هر فردی می‌تواند با ارائه یک نام کاربری یکتا و رمز عبور مناسب، به رایگان عضویت "کاربر ساده" را کسب کند. همچنین برخی از مشخصات فردی هر کاربر مانند نام و نام خانوادگی، ایمیل، شماره همراه و کد ملی برای ثبت نام نیاز است. در ابتدای ثبت نام، حساب هر فرد خالی است. کاربران می‌توانند در پنل کاربری اقدام به افزایش موجودی کیف پول حساب خود کنند. هر کاربر در ازای پرداخت مبلغی مشخص، می‌تواند حساب خود را به مدت یک ماه، به "کاربر ویژه" ارتقا دهد.

این سیستم یک ادمین (یک نام کاربری و کلمه عبور یکتا و مشخص) دارد که این فرد می‌تواند هر مقداری فیلم و یا ویدیو به سرویس اضافه کند. برای هر فیلم به غیر از فایل<sup>۱</sup>، اسم آن، سازندگان، سال ساخت، توضیحات و تعداد تماشای آن فیلم نیاز است. هر فیلم می‌تواند به یک یا چند دسته بندی و یا تگ تعلق داشته باشد. به این صورت که به طور مثال کاربران بتوانند لیست فیلم‌های "کمدی" درون سرویس را مشاهده کنند.

هر کاربر می‌تواند در سیستم با گشتن فیلم‌های یک دسته بندی خاص و یا جستجو در نام فیلم‌ها و یا سازندگان آن‌ها، فیلم را پیدا کند و در صورت تمایل، آن را به رایگان مشاهده کند. هر کاربر باید بتواند به فیلم‌ها نظر و امتیاز دهد (از ۵) و در پایگاه داده این امتیاز ثبت شده و حین مشاهده مشخصات فیلم، میانگین امتیازات کاربران و پنلی برای مشاهده نظرات باید قابل مشاهده باشد.

موسس این پلتفرم با اضافه کردن ایده اکران‌های خصوصی، قصد اضافه کردن اکران فیلم‌های روز در این سیستم را دارد ولی برخلاف فیلم‌های عادی، این فیلم‌ها تنها برای کاربران ویژه قابل مشاهده هستند و کاربران عادی، دسترسی به این فیلم‌ها را ندارند. البته هر کاربر ویژه نیز برای مشاهده این فیلم‌های اکران خصوصی، باید جدای اشتراک، هزینه فیلم را نیز بپردازد. به عبارتی هر فیلم در اکران ویژه یک صفت قیمت دارد.

هر کاربر جدای کیف پول، یک صفت دیگر به اسم "امتیاز" نیز دارد. تمامی کاربران می‌توانند یک کاربر دیگر را "معرف" خود بنامند و با هر معرفی، برای کاربر معرف یک امتیاز ثبت می‌شود. در انتها کاربران می‌توانند با

---

<sup>۱</sup> برای این پروژه نیازی به ذخیره فایل و ویدیو در دیتابیس نیست. صرفاً یک متن کافیست.

مصرف ۳ امتیاز، به روش دیگری یک ماه اکانت ویژه خود را تمدید کنند. در صورت کاربر ویژه بودن، با مصرف ۳ امتیاز می‌توانند یک فیلم اکران خصوصی را به رایگان تماشا کنند. در انتها کاربران ویژه می‌توانند از مجموعه فیلم‌های محبوب خود لیست‌هایی درست کنند و این لیست‌ها را همه‌ی کاربران می‌توانند مشاهده کنند و از درون لیست، فیلمی را انتخاب کنند و به پل آن فیلم بروند.

سوالات:

۱. موجودیت‌های استخراج شده این مجموعه با صفات<sup>۲</sup> هرکدام را تعیین کنید. در صورت وجود موجودیت‌های ضعیف<sup>۴</sup>، آن‌ها را مشخص کنید و دلیل این انتخاب را شرح دهید. کلیدهای اصلی<sup>۵</sup> هر موجودیت را نیز تعیین کنید.
۲. تمام ارتباط بین موجودیت‌ها<sup>۶</sup> و چندی ارتباط<sup>۷</sup> آن‌ها را با ذکر دلیل مشخص کنید و همچنین در صورت وجود، کلیدهای هریک را نام ببرید.
۳. نمودار ارتباط موجودیت<sup>۸</sup> این مجموعه را ترسم کنید.
۴. طبق کاردینالیتی‌های کشف شده، تمامی جداول این پروژه را با نام صفات و کلیدهای اصلی و خارجی نام ببرید.

---

<sup>2</sup> Entity

<sup>3</sup> Attribute

<sup>4</sup> Weak entity


<sup>5</sup> Primary key

<sup>6</sup> Relations

<sup>7</sup> Cardinality

<sup>8</sup> ER Diagram

بسمه تعالی

|  |   |
|--|---|
| <p>پاسخ تمرین دوم درس اصول و طراحی پایگاه داده<br/>دکتر ممتازی<br/>ترم بهار ۱۴۰۰ - دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر</p> |  |
|--|---|

توجه: با توجه به پیاده سازی‌های متفاوت، جواب‌های بسیار متنوعی برای این تمرین می‌توان نوشت. تمام موجودیت‌ها و روابط و مخصوصاً انتخاب موجودیت‌های ضعیف می‌تواند سلیقه‌ای و به علت طراحی و پیاده سازی ذهنی فرد نویسنده باشد. پس الزامی بر درستی مطلق این پاسخ‌ها نیست.

۱. موجودیت‌های استخراج شده این مجموعه با صفات<sup>۲</sup> هر کدام را تعیین کنید. در صورت وجود موجودیت‌های ضعیف<sup>۳</sup>، آن‌ها را مشخص کنید و دلیل این انتخاب را شرح دهید. کلیدهای اصلی<sup>۴</sup> هر موجودیت را نیز تعیین کنید.

لیست موجودیت‌ها :

- کاربر
- کاربر ویژه ((موجودیت ضعیف)): زیرا به طور کلی این یک کاربر است و هنگام ویژه شدن هر کاربر یک ID کاربر Pro دریافت می‌کند. (زیرا اگر یک کاربر ویژه سطری مخصوص خود در جدول کاربران نداشته باشد، ویژه بودن این کاربر معنا ندارد زیرا عملاً کاربری وجود ندارد).
- فیلم
- فیلم‌های ویژه ((موجودیت ضعیف)): زیرا به طور کلی این دسته از فیلم‌ها همان فیلم عادی هستند ولی یک یا چند مشخصه متمایز کننده دارند.
- ادمین
- تگ (دسته بندی)
- لیست

\*\* لیست صفات هر کدام طبق صورت پروژه قابل استخراج است. کلیدهای اصلی نیز طبق طراحی شما می‌تواند متفاوت باشد. در ER Diagram همه این اطلاعات کامل کشیده شده است.

---

<sup>1</sup> Entity

<sup>2</sup> Attribute

<sup>3</sup> Weak entity

<sup>4</sup> Primary key



۲. تمام ارتباط بین موجودیت‌ها<sup>۵</sup> و چندی ارتباط<sup>۶</sup> آن‌ها را با ذکر دلیل مشخص کنید و همچنین در صورت وجود، کلیدهای هریک را نام ببرید.

لیست روابط:

- نظر (بین فیلم و کاربر)
  - رابطه چند به چند: زیرا هر فیلم می‌تواند توسط چند کاربر نقد شود و هر کاربر می‌تواند چند فیلم را نقد کند.
  - هر دو سمت partial: زیرا کاربران و فیلم‌ها می‌توانند بدون مشارکت در نظردهی باشند.
- مشاهده (بین فیلم و کاربر)
  - رابطه چند به چند: زیرا هر فیلم می‌تواند توسط چند کاربر تماشا شود و هر کاربر می‌تواند چند فیلم را تماشا کند.
  - هر دو سمت partial: زیرا کاربران و فیلم‌ها می‌توانند بدون مشارکت در مشاهده باشند.
- مشاهده فیلم‌های خاص (بین فیلم‌های مخصوص و کاربر ویژه)
  - رابطه چند به چند: زیرا هر فیلم می‌تواند توسط چند کاربر ویژه تماشا شود و هر کاربر ویژه می‌تواند چند فیلم را مشاهده کند.
  - هر دو سمت partial: زیرا کاربران و فیلم‌ها می‌توانند بدون مشارکت در تماشا فیلم‌های خاص باشند.
- فیلم-تگ (بین فیلم و دسته بندی)
  - رابطه چند به چند: زیرا هر فیلم می‌تواند توسط چند تگ دسته بندی شود و هر تگ می‌تواند به چند فیلم متصل شود.
  - هر دو سمت partial: زیرا تگ‌ها و فیلم‌ها می‌توانند بدون مشارکت در این رابطه باشند.
- IdentifyingProUser (بین کاربر و کاربر ویژه) ((Identifying relation))
  - رابطه یک به یک: زیرا هر کاربر ویژه تنها به یک کاربر متصل می‌شود و هر کاربر عادی حداکثر به یک کاربر ویژه مربوط می‌شود.
  - ارتباط به کاربر partial: زیرا کاربر می‌تواند یک کاربر ویژه باشد و یا نباشد.

---

<sup>5</sup> Relations

<sup>6</sup> Cardinality

○ ارتباط به کاربر ویژه **total** : زیرا کاربر ویژه قطعا یک به عنوان یک کاربر عادی نیز ثبت نام شده.

• **IdentifyingSpecMovie** (بین فیلم و فیلم های اکران ویژه) ((Identifying relation))

○ رابطه یک به یک: زیرا هر فیلم اکران ویژه تنها به یک فیلم متصل می شود و هر فیلم عادی حداکثر به یک فیلم اکران ویژه مربوط می شود.

○ ارتباط به فیلم **partial** : زیرا فیلم میتواند ویژه باشد و یا نباشد.

○ ارتباط به فیلم ویژه **total** : زیرا فیلم ویژه قطعا یک به عنوان یک فیلم عادی نیز ثبت نام شده و اطلاعات پایه ای را دارد.

• معرفی (بین کاربر و کاربر)

○ رابطه چند به یک: زیرا هر کاربر میتواند چندین کاربر را معرفی کند ولی یک کاربر حداکثر توسط یک کاربر معرفی شده است.

○ هر دو سمت **partial** : زیرا معرفی کردن و وجود معرف می تواند باشد و یا نباشد.

• **List\_relation** (رابطه ۳ گانه بین فیلم، کاربر ویژه و یک لیست)

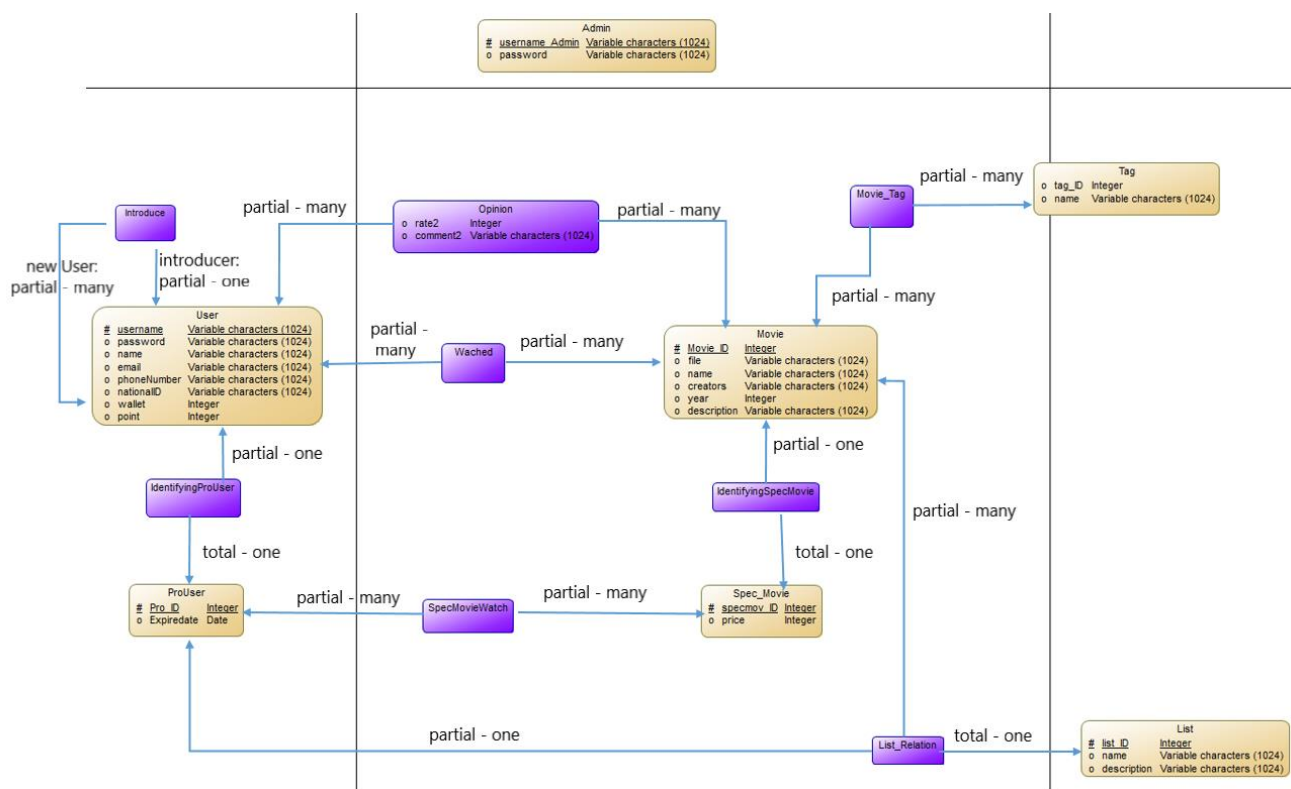
○ ارتباط با سمت فیلم **partial - many** : زیرا مجموعه {کاربر ویژه و لیست} یکتا با چند فیلم می تواند در این ارتباط قرار گیرد (در یک لیست بی شمار فیلم می تواند وجود داشته باشد)

○ ارتباط با سمت کاربر ویژه **partial - one** : زیرا مجموعه {فیلم و لیست} فقط به یک کاربر متصل می شود و هر کاربر ویژه می تواند لیست بسازد و یا نسازد.

○ ارتباط با سمت لیست **total - one** : زیرا هر رابطه فقط به یک لیست ارتباط دارد و هر لیست قطعا یک رابطه دارد.

\*\*\* برای این رابطه سه گانه گفته شده، می توان این مسئله را به روش های دیگر و یا حتی با دو رابطه دوگانه به صورت بهتر حل کرد. تمامی این موارد به دید طراح پایگاه داده متصل است و جواب یکتایی ندارد. این رابطه سه گانه برای آشنایی بیشتر و مثالی برای این روابط بود.

۳. نمودار ارتباط موجودیت<sup>۷</sup> این مجموعه را ترسم کنید.



۴. طبق کاردینالیتی‌های کشف شده، تمامی جداول این پروژه را با نام صفات و کلیدهای اصلی و خارجی نام

ببرید.

تمام موجودیت‌ها جدول مخصوص به خود را دارند.

روابط چند به چند نیز نیازمند جدول مختص خود هستند.


روابط چند به یک می‌توانند با موجودیت سمت "چند" خود ادغام شوند (به عنوان مثال در رابطه معرفی، هر کاربر می‌تواند چند کاربر را معرفی کند ولی یک کاربر تنها توسط یک کاربر معرفی شده. پس می‌توان برای هر کاربر یک صفت "معرف" در جدول خود قرار داد که می‌تواند خالی و یا null باشد و یا شامل id کاربر معرف باشد به صورت کلید خارجی)

<sup>7</sup> ER Diagram

روابط identifier برای موجودیت های ضعیف به موجودیت ضعیف ادغام می شوند. (به عنوان مثال هر کاربر ویژه با داشتن id کاربر عادی، به ریکورد اصلی خود متصل می شود).

روابط سه گانه نیز یک جدول جدا و مخصوص خود را می توانند داشته باشند. (البته نیاز به توضیح است که برای روابط بیشتر از ۲ طرفه شاید خیلی مناسب نباشد که بگوییم همیشه جدول جدا بنویسیم. اگر جدول one-one-one باشد می توان به یکی از سمت ها متصلش کرد. همین مثال رابطه سه گانه که در بالا گفته شده میتوان به سمت لیست متصلش کرد ولی به مشکل چند فیلم بر می خوریم. در کل برای حل این مسائل به بینش نورمالایز کردن جداول نیز نیاز داریم که در فصل بعدی یاد خواهید گرفت).

بسمه تعالی

|   |   |
|---|---|
| <p>پاسخ تمرین سوم درس اصول و طراحی پایگاه داده<br/>دکتر ممتازی<br/>ترم بهار 1400 – دانشکده کامپیوتر، دانشگاه صنعتی امیرکبیر</p> |  |
|---|---|

1. با استفاده از Armstrong's Axioms قانون pseudotransitivity را اثبات کنید.

If  $\alpha \rightarrow \beta$  holds and  $\gamma \beta \rightarrow \delta$  holds, then  $\alpha \gamma \rightarrow \delta$  holds (pseudotransitivity)

پاسخ)

$\alpha \rightarrow \beta$  and  $\gamma \beta \rightarrow \delta$

$\alpha \gamma \rightarrow \gamma \beta$  (augmentation)

$\gamma \beta \rightarrow \delta$  (given)

Therefore  $\alpha \gamma \rightarrow \delta$  (transitivity)

2. شمای رابطه‌ای  $R(A, B, C, D, E, F, G, H)$  با وابستگی های تابعی زیر را در نظر بگیرید.

$A \rightarrow B$

$B \rightarrow C$

$AB \rightarrow D$

$E \rightarrow AD$

$EC \rightarrow F$

$EFD \rightarrow GAC$

$AH \rightarrow E$

الف) Extraneous Attribute ها را مشخص کنید و وابستگی های تابعی بالا را به فرم Canonical Cover بنویسید.

ب) کلید های کاندید را بنویسید.

پاسخ)

الف)

B در سمت چپ  $AB \rightarrow D$  اضافه است. زیرا از وابستگی تابعی  $A \rightarrow B$  می توان از A، ویژگی B را نتیجه گرفت.

$F = \{A \rightarrow B, B \rightarrow C, A \rightarrow D, E \rightarrow AD, EC \rightarrow F, EFD \rightarrow GAC, AH \rightarrow E\}$

D در سمت راست  $E \rightarrow AD$  اضافه است. زیرا طبق رابطه  $A \rightarrow D$  می توان از A، ویژگی D را نتیجه گرفت.

$$F = \{A \rightarrow B, B \rightarrow C, A \rightarrow D, E \rightarrow A, EC \rightarrow F, EFD \rightarrow GAC, AH \rightarrow E\}$$

C در سمت چپ  $EC \rightarrow F$  اضافه است. زیرا از مجموعه وابستگی  $E \rightarrow A, A \rightarrow B, B \rightarrow C$  می‌توان از E، ویژگی C را نتیجه گرفت.

$$F = \{A \rightarrow B, B \rightarrow C, A \rightarrow D, E \rightarrow A, E \rightarrow F, EFD \rightarrow GAC, AH \rightarrow E\}$$

F در سمت چپ  $EFD \rightarrow GAC$  اضافه است. زیرا در وابستگی  $E \rightarrow F$  از E، به ویژگی F می‌رسیم. D نیز در سمت چپ اضافه است زیرا از مجموعه  $E \rightarrow A, A \rightarrow D$  با داشتن E می‌توان D را نتیجه گرفت. همینطور A و C در سمت راست اضافه هستند زیرا از وابستگی تابعی  $E \rightarrow A$  می‌توان A را نتیجه گرفت و با داشتن A از مجموعه وابستگی  $A \rightarrow B, B \rightarrow C$  می‌توان به C نیز رسید.

$$F = \{A \rightarrow B, B \rightarrow C, A \rightarrow D, E \rightarrow A, E \rightarrow F, E \rightarrow G, AH \rightarrow E\}$$

فرم Canonical Cover:

$$A \rightarrow B$$

$$B \rightarrow C$$

$$A \rightarrow D$$

$$E \rightarrow A$$

$$E \rightarrow F$$

$$E \rightarrow G$$

$$AH \rightarrow E$$

(ب)

با توجه به وابستگی‌های تابعی داریم  $(AH)^+ = ABCDEFGH$  و  $(EH)^+ = ABCDEFGH$ .

طبق شمای  $R(A, B, C, D, E, F, G, H)$ ، AH و EH کلیدهای کاندید می‌باشند.

3. درستی و یا نادرستی هر یک گزاره را با ذکر علت مشخص کنید.

a. اگر جدولی به فرم نرمال BCNF باشد، به فرم 3NF نیز خواهد بود.

b. اگر جدولی به فرم نرمال 3NF باشد، به فرم BCNF نیز خواهد بود.

c. اگر جدولی به فرم نرمال 4NF باشد، به فرم 3NF و BCNF نیز خواهد بود.

پاسخ)

- a. درست: در واقع شروط BCNF سخت تر از 3NF می باشد و به گونه ای نوع پیشرفته آن می باشد. طبق شروطی که برای BCNF داشتیم، به ازای هر وابستگی تابعی باید رابطه بدیهی باشد و یا سمت چپ super key باشد. که در اینصورت 3NF نیز خواهد بود.
- b. نادرست: طبق شروطی که 3NF دارد، اگر سمت راست وابستگی تابعی زیرمجموعه یکی از کلید های کاندید باشد، شرط را برآورده می کند. اما در اینصورت BCNF نخواهد بود.
- c. درست: در صورتی جدول به فرم نرمال 4NF است که به فرم نرمال BCNF باشد و همچنین multivalued نباشد. اگر بصورت BCNF باشد، 3NF نیز خواهد بود.

4. شمای رابطه ای  $R(A, B, C, D, E, F)$  با وابستگی های تابعی زیر را در نظر بگیرید.

$A \rightarrow B$

$BF \rightarrow D$

$ADF \rightarrow E$

$AE \rightarrow C$

$BC \rightarrow F$

الف) کلید های کاندید را مشخص کنید.

ب) ارضاء پذیری شروط BCNF و 3NF را برای هر یک از وابستگی های تابعی چک کنید.

ج) اگر تجزیه ای به فرم  $R_1(B, C, F)$  و  $R_2(A, B, C, D, E)$  داشته باشیم، نشان دهید این تجزیه lossless می باشد.

د) با ذکر مثال توضیح دهید که چرا تجزیه ای به فرم  $R_1(A, F, B)$  و  $R_2(B, C, D, E)$ ، lossy می باشد.

ه) شمای رابطه ای داده شده را به گونه ای تجزیه کنید که بصورت فرم نرمال BCNF باشد. کلید های هر جدول را نیز مشخص کنید.

پاسخ)

الف)



کلیدهای کاندید: AC, AE, AF

(ب)

BCNF:

طبق شرط BCNF، وابستگی یا باید بدیهی باشد و یا سمت چپ super key باشد.

$A \rightarrow B$  ❌

$BF \rightarrow D$  ❌

$ADF \rightarrow E$  OK

$AE \rightarrow C$  OK

$BC \rightarrow F$  ❌

این جدول بصورت نرمال BCNF نمی‌باشد.

3NF:

برای اینکه جدول در سطح نرمال 3NF باشد، باید تمامی وابستگی‌ها همان شروط BCNF را داشته باشند و یا سمت راست وابستگی، prime (زیر مجموعه یکی از کلیدهای کاندید) باشد.

$A \rightarrow B$  ❌

$BF \rightarrow D$  ❌

$ADF \rightarrow E$  OK

$AE \rightarrow C$  OK

$BC \rightarrow F$  OK

این جدول بصورت نرمال 3NF نمی‌باشد.

(ج)

رابطه  $R(A,B,C,D,E,F)$  به دو زیر رابطه  $R_1(B, C, F)$  و  $R_2(A, B, C, D, E)$  تجزیه شده است. که این دو رابطه  $R_1$  و  $R_2$  شامل تمامی attribute های  $R$  می‌باشند. از طرفی اگر  $\alpha$  اشتراک این دو رابطه باشد و ما

functional dependency به حالت  $\alpha \rightarrow R1$  و یا  $\alpha \rightarrow R2$  داشته باشیم، می توانیم ادعا کنیم که این تجزیه، Lossless است.

اشتراک این دو جدول : BC

$$(BC)^+ = (BCF)$$

می بینیم که با توجه به وابستگی های تابعی داده شده از BC به BCF که شامل تمامی ویژگی های  $R1$  می باشد، رسیدیم. پس این تجزیه Lossless می باشد.

(د)

جدول زیر را با مقادیر وارد شده درنظر بگیرید.

| A | F | B | C | D | E |
|---|---|---|---|---|---|
| 1 | x | a | b | d | f |
| 2 | y | a | c | e | g |

حال این جدول را به دو جدول زیر تجزیه می کنیم.

| A | F | B |
|---|---|---|
| 1 | x | a |
| 2 | y | a |

| B | C | D | E |
|---|---|---|---|
| a | b | d | f |
| a | c | e | g |

زمانی که این دو جدول را natural join کنیم خواهیم داشت:

| A | F | B | C | D | E |
|---|---|---|---|---|---|
| 1 | x | a | b | d | f |
| 1 | x | a | c | e | g |
| 2 | y | a | b | d | f |
| 2 | y | a | c | e | g |

می بینیم که با جدول اولیه ما متفاوت است و سطرهایی اضافه شده است که در دیتابیس ما موجود نبود.

بنابراین این تجزیه lossless نمی باشد.

طبق تعریف قسمت ج نیز می توان این را تشخیص داد. اشتراک این دو جدول ویژگی B می باشد و با توجه به وابستگی های رابطه ای داده شده نمی توان از B به تمامی ویژگی های حداقل یکی از دو جدول رسید. بنابراین lossy می باشد. به زبان ساده تر علت این امر این است که، زمانی که بتوان به تمامی ویژگی های حداقل یکی از جداول (از طریق functional dependency ها) رسید این معنی را می دهد که آن ویژگی های مشترک کلید های کاندید برای آن جدول هستند، در نتیجه نمی توان به ازای یک مقدار چند سطر متفاوت داشت. بنابراین به هنگام join کردن نیز مشکلی پیش نخواهد آمد.

نکته دیگر هم اینکه اگر اشتراک نداشته باشند، دیگر قابل join کردن نیستند. پس قطعاً lossy خواهد بود.

(۵)

1.  $A \rightarrow B$  ✗
2.  $BF \rightarrow D$  ✗
3.  $ADF \rightarrow E$  OK
4.  $AE \rightarrow C$  OK
5.  $BC \rightarrow F$  ✗

1:

$R1 = (A, B) \quad CK=\{A\}$

$R2 = (A, C, D, E, F) \quad CK=\{AC, AE, AF\}$

2:

$A \rightarrow B \quad BF \rightarrow D \quad AF \rightarrow D$

$R21 = (A, F, D) \quad CK=\{AF\}$

$R22 = (A, C, E, F) \quad CK=\{AC, AE, AF\}$

5:

$A \rightarrow B \quad BC \rightarrow F \quad AC \rightarrow F$

سمت چپ super key می باشد و تجزیه نمی کنیم.

بنابراین یک تجزیه در فرم BCNF بصورت زیر می‌باشد.

$$R1 = (A, B) \text{ CK}=\{A\}$$

$$R21 = (A, F, D) \text{ CK}=\{AF\}$$

$$R22 = (A, C, E, F) \text{ CK}=\{AC, AE, AF\}$$

5. رابطه  $R(A, B, C, D, E, F, G)$  را در نظر بگیرید. طبق وابستگی‌های تابعی زیر، به سوالات پاسخ دهید.

$$ABC \rightarrow D$$

$$AE \rightarrow F$$

$$AE \rightarrow C$$

$$C \rightarrow E$$

$$A \rightarrow B$$

$$B \rightarrow G$$

$$AF \rightarrow CG$$

$$FD \rightarrow G$$

الف) کلیدهای کاندید را بنویسید.

ب) برای هریک از وابستگی‌های تابعی داده شده در صورت سوال، ارضاپذیری شروط BCNF را چک کنید. سپس تجزیه‌ای به فرم BCNF ارائه دهید و کلیدهای هر رابطه را مشخص کنید. همینطور با ذکر دلیل بیان کنید آیا تجزیه حاصل Dependency preserving را رعایت می‌کند یا خیر؟

ج) برای هریک از وابستگی‌های تابعی داده شده، ارضاپذیری شروط 3NF را چک کنید. سپس آن‌ها را به فرم Canonical Cover در آورده و برای این مجموعه کاهش ناپذیر تجزیه‌ای به فرم 3NF ارائه دهید. کلیدهای هر رابطه را نیز مشخص کنید.

پاسخ)

الف)

کلیدهای کاندید: AC, AE, AF

(ب)

1.  $ABC \rightarrow D$  OK
2.  $AE \rightarrow F$  OK
3.  $AE \rightarrow C$  OK
4.  $C \rightarrow E$  ✗
5.  $A \rightarrow B$  ✗
6.  $B \rightarrow G$  ✗
7.  $AF \rightarrow CG$  OK
8.  $FD \rightarrow G$  ✗

4:

$R_1 = (C, E)$   $CK=\{C\}$

$R_2 = (A, B, C, D, F, G)$   $CK=\{AC, AF\}$

5:

$R_{21} = (A, B)$   $CK=\{A\}$

$R_{22} = (A, C, D, F, G)$   $CK=\{AC, AF\}$

6:

$A \rightarrow B, B \rightarrow G, A \rightarrow G$

$R_{221} = (A, G)$   $CK=\{A\}$

$R_{222} = (A, C, D, F)$   $CK=\{AC, AF\}$

با توجه به روابط حاصل، تجزیه را نمی‌توانیم ادامه دهیم. و تجزیه در فرم BCNF زیر را داریم:

$R_1 = (C, E)$   $CK=\{C\}$

$R_{21} = (A, B)$   $CK=\{A\}$

$R_{221} = (A, G)$   $CK=\{A\}$

$R_{222} = (A, C, D, F)$   $CK=\{AC, AF\}$

این تجزیه dependency preserving نمی‌باشد. بطور کلی تجزیه BCNF تضمینی برای حفظ وابستگی‌ها نمی‌دهد. در اینجا نیز وابستگی‌های تابعی  $AE \rightarrow F$ ,  $AE \rightarrow C$ ,  $B \rightarrow G$  و  $FD \rightarrow G$  را از دست داده‌ایم.

(ج)

1.  $ABC \rightarrow D$  OK
2.  $AE \rightarrow F$  OK
3.  $AE \rightarrow C$  OK
4.  $C \rightarrow E$  OK
5.  $A \rightarrow B$  ✗
6.  $B \rightarrow G$  ✗
7.  $AF \rightarrow CG$  OK
8.  $FD \rightarrow G$  ✗

فرم Canonical Cover:

$AC \rightarrow D$

$AE \rightarrow F$

$C \rightarrow E$

$A \rightarrow B$

$B \rightarrow G$

$AF \rightarrow C$

$FD \rightarrow G$

تجزیه‌ای به فرم 3NF:

$R1 = (\underline{A}, \underline{C}, D)$

$R2 = (\underline{A}, E, F)$

$R3 = (\underline{C}, E)$

$R4 = (\underline{A}, B)$

$R5 = (\underline{B}, G)$

$R6 = (\underline{A}, F, C)$

$R7 = (\underline{F}, \underline{D}, G)$

6. شمای رابطه‌ای مربوط به دانشگاهی بصورت زیر است. که شامل ویژگی های StudentID, FullName, Department (نام دانشکده دانشجو)، DepBudget (بودجه دانشکده)، Courses (درس های اخذ شده هر دانشجو) و Emails (یک یا چند) می باشد. کلید اصلی این جدول StudentID می باشد.

| StudentID | FullName | Department | DepBudget | Courses | Emails |
|-----------|----------|------------|-----------|---------|--------|
| X         | A        | CE         | 1000000   | DB      | C      |
| X         | A        | CE         | 1000000   | PL      | B      |
| ...       | ...      | ...        | ....      | ...     | ...    |
| Y         | B        | CS         | 2000000   | PL      | D      |
| Y         | B        | CS         | 2000000   | AP      | D      |
| ...       | ...      | ...        | ...       | ...     | ...    |

الف) تمامی وابستگی های این جدول را بنویسید و نوع هر یک را (functional یا multivalued) مشخص کنید.

ب) این طراحی چه مشکلاتی را به وجود می آورد؟ توضیح دهید.

ج) برای جلوگیری از رخ دادن این مشکلات، در صورت امکان یک و درغیراینصورت چندین تجزیه ارائه دهید که به صورت فرم های نرمال 3NF، BCNF و 4NF باشند. کلید های تمامی جداول را نیز مشخص کنید.

پاسخ)

الف)

StudentID → FullName, Department, DepBudget

Department → DepBudget

StudentID →→ Courses

StudentID →→ Emails

ب)

در این طراحی فیلد های multivalued که ارتباطی با هم ندارند و مستقل هستند، باعث ایجاد سطر های اضافی می شوند. اگر هر دانشجو n تعداد course و m تعداد ایمیل داشته باشد، در نتیجه به ازای آن  $m \times n$  سطر خواهیم داشت.

همینطور از ویژگی Department که زیرمجموعه کلید های کاندید نمی باشد، می توان DepBudget را بدست آورد. در صورتی که این ویژگی ها جدا نشوند باعث redundancy می شود؛ زیرا به ازای هر سطر باید تکرار شوند.

(ج)

تجزیه زیر بصورت فرم نرمال 4NF می باشد که همینطور شروط فرم های نرمال BCNF و 3NF را برآورده می کند.

کلید کاندید: StudentID

R1 = (Department, DepBudget)

R2 = (StudentID, Courses)

R3 = (StudentID, Emails)

R4 = (StudentID, FullName, Department)