

بسمه تعالی

تمرین اول درس اصول و طراحی پایگاه داده
دکتر ممتازی

ترم پائیز ۱۳۹۹ - دانشکده کامپیوتر، دانشگاه صنعتی امیرکبیر
زمان تحويل: ۱۲ آبان ماه



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

۱. در صورت وجود هرگونه سوال یا ابهام می‌توانید با تدریسیار از طریق ایمیل a.varaste.n@gmail.com در ارتباط باشید.
۲. مهلت تحويل تمرین تا تاریخ دوازده آبان می‌باشد. بعد از این تاریخ تمرین به هیچ عنوان تحويل گرفته نمی‌شود.
۳. تمامی فایل‌های خواسته شده را در قالب یک فایل فشرده با نام HW1-StudentNumber در مدل بارگذاری کنید. به عنوان مثال: HW1-9531888

- (۱) تفاوت File System و پایگاه داده در چیست؟ با وجود File System چه نیازی به استفاده از پایگاه داده وجود دارد؟ مزایای استفاده از پایگاه داده چیست؟
- (۲) تفاوت ^۱DDL و ^۲DML چیست؟
- (۳) درستی و نادرستی عبارت‌های زیر را مشخص کنید. علت را توضیح دهید و یا مثال بزنید.
- عملگر انتخاب^۳ در جبر رابطه‌ای نسبت به خودش دارای خاصیت جابجایی^۴ نیست.
 - در جبر رابطه‌ای هنگام استفاده از عملگر پرتو^۵، تاپل‌های تکراری فقط یک بار در خروجی ظاهر می‌شوند.
 - دو عملگر انتخاب و پرتو نسبت به یکدیگر مطلقاً جابجایی پذیر نیستند.
 - عملگر انتخاب دارای خاصیت توزیع پذیری بر اجتماع است.
 - عملگر انتخاب دارای خاصیت توزیع پذیری بر اشتراک است.
 - عملگر الحق طبیعی^۶ دارای خاصیت‌های جابجایی و شرکت‌پذیری^۷ است.
- (۴) روابط زیر درخصوص کتابخانه دانشگاه صنعتی امیرکبیر را در نظر بگیرید:

BOOKS(DocId, Title, Publisher, Year)

STUDENTS(StId, StName, Field, Age) (Field: رشته تحصیل دانشجو)

AUTHORS(AName, City)

BORROWS(DocId, StId)

HAS-WRITTEN(DocId, AName)

DESCRIBES(DocId, Keyword)

با استفاده از جبر رابطه‌ای پرس‌و‌جوهای خواسته شده را بنویسید.

- لیست سال و عنوان همه کتاب‌ها.
- همه اطلاعات دانشجویانی که در رشته Computer Engineering تحصیل می‌کنند.
- لیست کتاب‌هایی که تا سال ۱۳۹۸ توسط انتشارات دانشگاه صنعتی امیرکبیر به چاپ رسیده.
- اسامی نویسنده‌گانی که در شهر تهران زندگی می‌کنند.
- اسامی دانشجویانی که بالای ۲۵ سال سن دارند و رشته آنها Computer Engineering نیست.

^۱ Data Definition Language

^۲ Data Manipulation Language

^۳ Select

^۴ Commutative Property

^۵ Project

^۶ Natural Join

^۷ Associative Property

- f. لیست تمام کتاب‌هایی که دانشجویان مهندسی کامپیوتر به امانت گرفته اند.
- g. عنوان تمام کتاب‌هایی که به وسیله هاریسون نوشته شده.
- h. نام و سن جوان ترین دانشجو رشته مهندسی کامپیوتر.
- i. نام نویسنده‌گان کتاب‌هایی که دانشجویی با نام Carlos آنها را قرض گرفته.

(۵) رابطه **People (NationalID, Name, BirthDate, BirthCity, FatherID, MotherID)** را در نظر

بگیرید. با استفاده از جبر رابطه‌ای پرس‌وجوهای خواسته شده را بنویسید.

- a. شماره ملی افرادی که متولد شهرهای تهران و یا شیراز هستند.
- b. نام افرادی که متولد سال ۱۳۷۷ در تبریز هستند.
- c. شماره ملی افرادی که در سال ۱۴۰۰ بیشتر از ۲۳ سال سن دارند.
- d. نام و تاریخ تولد فرزندان فردی با شماره ملی ۱۰۶۱۴۲۶۲۸۶.
- e. اسمی برادران یا خواهران فردی با شماره ملی ۶۸۲۶۲۴۱۶۰۱
- f. شماره ملی افرادی که محل تولدشان با محل تولد پدر و مادرشان یکسان است.
- g. نام و شماره ملی کسانی که برادر یا خواهر بزرگتر از خود دارند.
- h. نام و شماره ملی عموها و عمه‌های فردی با شماره ملی ۱۰۶۱۴۲۶۲۸۶
- i. نام کسانی که برادر یا خواهر دوقلو دارند.

(۶) روابط زیر را در نظر بگیرید:

Customer (CustomerID, Name)

Stuff (ItemName, Company)

Buy (CustomerID, ItemName, Year)

توضیح دهید هر کدام از این عبارت‌ها چه چیزی را نتیجه می‌دهد.

- a. $\pi_{\text{ItemName}}(\sigma_{\text{Year} < 2020}(\text{Buy}))$
- b. $\pi_{\text{Name}}(\delta_{\text{Year} = 2020}(\text{Buy} \bowtie \text{Customer}))$
- c. $\pi_{\text{Name}}(\text{Customer} \bowtie (\pi_{\text{CustomerID}}(\text{Customer}) - \pi_{\text{CustomerID}}(\text{Buy})))$

موفق باشد

بسمه تعالیٰ

پاسخ تمرین اول درس اصول و طراحی پایگاه داده

دکتر ممتازی

ترم پائیز ۱۳۹۹ - دانشکده کامپیوتر، دانشگاه صنعتی امیرکبیر



ANSWER (d)

ANSWER (e)

ANSWER (f)

ANSWER (a) ①

ANSWER (b)

ANSWER (c)

a) $\Pi_{Year, Title}(\text{Books})$ ④

b) $\sigma_{Field = "Computer Engineering"}(\text{STUDENTS})$

c) $\sigma_{Publisher = "AUT" \wedge Year < 1398}(\text{Books})$

d) $\Pi_{AName}(\sigma_{City = "Tehran"}(\text{AUTHORS}))$

e) $\Pi_{stName}(\sigma_{Age > 25}(\text{STUDENTS})) - \Pi_{stName}(\sigma_{Field = "Computer Engineering"}(\text{STUDENTS}))$

f) $\Pi_{Title}(\sigma_{Field = "CE"}(\text{BORROWS} \bowtie \text{STUDENTS} \bowtie \text{Books}))$

g) $\Pi_{Title}(\sigma_{AName = "Harrison"}(\text{HAS-WRITTEN} \bowtie \text{Books}))$

h) $\Pi_{stName, Age}(\sigma_{Field = "CE"}(\text{STUDENTS})) -$

$\Pi_{s_1.stName, s_1.Age} \left(\sigma_{s_1.Field = "CE" \wedge s_2.Field = "CE"} \left(P_{s_1}(\text{STUDENTS}) \times P_{s_2}(\text{STUDENTS}) \right) \right)$
 $\wedge s_1.Age > s_2.Age$

i) $\Pi_{AName}(\sigma_{stName = "Carlos"}(\text{STUDENTS} \bowtie \text{BORROWS} \bowtie \text{HASWRITTEN}))$

(W)

$$a) \pi_{\text{NationalID}} \left(\sigma_{\text{birth city} = "Tehran"}(\text{People}) \right) \cup \pi_{\text{NationalID}} \left(\sigma_{\text{birth city} = "shiraz"}(\text{People}) \right)$$

$$b) \pi_{\text{Name}} \left(\sigma_{\text{birth city} = "Tabriz"} \wedge \text{Birth Date} = 1377(\text{People}) \right)$$

$$c) \pi_{\text{NationalID}} \left(\sigma_{\text{Birth Date} < 1377}(\text{People}) \right)$$

$$d) \pi_{\text{name}, \text{Birth Date}} \left(\sigma_{\text{Father ID} = 1061426286}(\text{People}) \right) \cup \pi_{\text{name}, \text{Birth Date}} \left(\sigma_{\text{Mother ID} = 1061426286}(\text{People}) \right)$$

$$e) \pi_{\text{Name}} \left(\sigma_{\text{NationalID} = 0926241601} \left(\sigma_{P_1.\text{Father ID} = P_2.\text{Father ID}} \wedge P_1.\text{Mother ID} = P_2.\text{Mother ID} \right. \right. \\ \left. \left. \wedge (P_1(\text{People}) \times P_2(\text{People})) \right) \right)$$

$$f) \pi_{P_1.\text{NationalID}} \left(\sigma_{P_1.\text{Father ID} = P_2.\text{NationalID}} \wedge P_1.\text{city} = P_2.\text{city} \wedge P_1.\text{Mother ID} = P_3.\text{NationalID} \wedge P_1.\text{city} = P_3.\text{city} \right) \\ \left(P_{P_1}(\text{People}) \times P_{P_2}(\text{People}) \times P_{P_3}(\text{People}) \right)$$

$$g) \pi_{P_1.\text{Name}, P_1.\text{NationalID}} \left(\sigma_{P_1.\text{Father ID} = P_2.\text{Father ID}} \wedge P_1.\text{Mother ID} = P_2.\text{Mother ID} \wedge P_1.\text{Birth Date} > P_2.\text{Birth Date} \right) \\ \left(P_{P_1}(\text{People}) \times P_{P_2}(\text{People}) \right)$$

$$h) P_{\text{father}} : (\pi_{\text{Father ID}} \left(\sigma_{\text{NationalID} = 1061426286}(\text{People}) \right)) \\ P_{\text{grandfather}} : (\pi_{P_1.\text{Father ID}} \left(\sigma_{P_1.\text{NationalID} = \text{father}.\text{NationalID}} \left(P_{P_1}(\text{People}) \times \text{father} \right) \right))$$

$$\pi_{P_2.\text{Name}, P_2.\text{NationalID}} \left(\sigma_{P_2.\text{Father ID} = \text{grandfather}.\text{NationalID}} \wedge P_2.\text{NationalID} \neq \text{father}.\text{NationalID} \right)$$

$$i) \pi_{P_1.\text{Name}} \left(\sigma_{P_1.\text{MotherID} = P_2.\text{MotherID}} \wedge_{P_1.\text{FatherID} = P_2.\text{MotherID}} \wedge_{P_1.\text{BirthDate} = P_2.\text{BirthDate}} \wedge_{P_1.\text{NationalID} \neq P_2.\text{NationalID}} \left(P_{P_1}(\text{People}) \times P_{P_2}(\text{People}) \right) \right)$$

- ١٥
- . العنوان الاسم الجنس المواليد المواليد الجنس العنوان (a)
 - . العنوان العنوان الجنس المواليد المواليد الجنس (b)
 - . العنوان العنوان الجنس المواليد المواليد (c)

بسمه تعالیٰ

<p>تمرین اول درس اصول و طراحی پایگاه داده دکتر ممتازی ترم پائیز ۱۳۹۹ – دانشکده کامپیوتر، دانشگاه صنعتی امیرکبیر زمان تحويل: ۱ آذر ۱۳۹۹</p>	
--	---

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

۱. در صورت وجود هرگونه سوال یا ابهام می‌توانید با تدریسیار از طریق ایمیل aref78.m@gmail.com در ارتباط باشید.
۲. مهلت تحويل تمرین تا تاریخ ۱ آذر می‌باشد. بعد از این تاریخ تمرین به هیچ عنوان تحويل گرفته نمی‌شود.
۳. تمامی فایل‌های خواسته شده را در قالب یک فایل فشرده با نام HW2-StudentNumber در مدل بارگذاری کنید. به عنوان مثال: HW2-9531888

فراگ کوییز

یک شرکت توسعه نرم افزار برای برنامه جدید خود به دنبال یک نفر برای کار با پایگاه داده است. این برنامه که "فراگ کوییز" نام دارد، یک مسابقه روزانه در در زمینه اطلاعات عمومی است. این برنامه اینگونه است که کاربران پس از ایجاد اکانت کاربری، می توانند روزانه تعداد حداقل ۵ آزمون ۵ سواله بگیرند و پاسخ دهند.

همه سوال ها به صورت ۴ گزینه ای مطرح می شوند. سوال ها در موضوعات "علمی"، "فرهنگی و ادبی"، "جغرافیا و تاریخ" و "ورزشی" می توانند طرح شوند و هر آزمون تماماً متعلق به یکی از این زمینه ها می باشد. در واقع برای هر آزمون، تعدادی از سوالات زمینه مربوط به صورت تصافی انتخاب می شوند و به کاربر داده می شوند و کاربر پس از اتمام آزمون، به ازای هر پاسخ درست، ۱ امتیاز مثبت دریافت میکنند. در انتهای هر روز، به کاربرانی که بیشترین امتیاز را گرفته باشند، یک جام نقره ای تعلق میگیرد.

در پایگاه داده خواسته شده نیاز به جدول های زیر با داده های گفته شده داریم:

جدول کاربران:

- نام کاربری (UserId)
- نام و نام خانوادگی
- شماره همراه
- تعداد جام ها

سوالات:

- شماره سوال (QId)
- موضوع
- صورت سوال
- گزینه ۱
- گزینه ۲
- گزینه ۳
- گزینه ۴
- پاسخ صحیح

آزمون ها:

- شماره آزمون (QuizId)
- کاربر آزمون دهنده (UserId)
- تاریخ و زمان ثبت آزمون
- زمینه آزمون

سوالات آزمون:

- شماره آزمون (QuizId)
- شماره سوال (QId)
- پاسخ ثبت شده

Query های خواسته شده:

- (۱) نمایش کاربرانی که نام آنها "رضا" است.
- (۲) نمایش کاربران به ترتیب نزولی تعداد جام.
- (۳) کاربرانی که در یک روز مشخص آزمون داده اند.
- (۴) لیست کاربرانی که حداقل یک آزمون در موضوع "ورزشی" داده اند.
- (۵) تعداد آزمون هایی که در هر موضوع انجام شده.
- (۶) شماره گزینه انتخاب شده غالب توسط کاربران در سوالات موضوع "ورزشی"
- (۷) شماره گزینه صحیح غالب در کل سوالات
- (۸) لیستی از موضوعات و تعداد سوالات ثبت شده برای هر کدام.
- (۹) لیستی از کاربران و تعداد آزمون های داده شده هر کاربر.
- (۱۰) نمایش کاربران به ترتیب بیشترین پاسخ درست.
- (۱۱) لیست آزمون های کاربر "رضا ایرانی" و امتیاز هر کدام.
- (۱۲) نمایش کاربران به ترتیب میانگین درصد پاسخ درست (اگر کامل به آزمون پاسخ داده شود، درصد پاسخ ۱۰۰ درصد می شود).
- (۱۳) نمایش کاربران و امتیازات کسب شده به ترتیب نزولی.
- (۱۴) نمایش کاربرانی که حداقل یک آزمون با امتیاز کامل برای آنها ثبت شده باشد.

(۱۵) محبوب ترین موضوع در بین موضوعات (بر اساس تعداد آزمون ثبت شده)

(۱۶) لیستی از کاربران و تاریخ اولین آزمون ثبت شده برای آنها.

(۱۷) کاربرانی که در یک روز مشخص، ۳ یا بیشتر آزمون داده اند.

موارد تحويلی:

- در فایلی به نام report.pdf در مورد کلید های اصلی و خارجی هر جدول توضیح دهید. و دیاگرام شما (schema diagram) این پایگاه داده را رسم کنید.
- دستورات ساخت جداول گفته شده را در فایلی به نام tables.sql ارسال کنید.
- تعدادی رکورد به جداول اضافه کنید (به اندازه کافی تا بتوان درستی query های قسمت بعد را چک کرد) و در فایل inserts.sql ارسال کنید.
- Query های خواسته شده را پاسخ دهید و به ترتیب، تمامی را در فایلی به نام queries.sql ارسال کنید.

توضیحات کلی و راهنمایی:

برای دید کلی پایگاه داده گفته شده، به مثال زیر توجه کنید.

در شکل زیر داده های لازم برای یک آزمون برای یک کاربر نشان داده شده:

UserID	Name	PhoneNumber	Trophies
u01	Aref Motamed	09121111111	1

شکل ۱ - جدول کاربران

QuizId	UserID	Date	Topic
1	u01	"2020-11-1 10:00:00"	Sports

شکل ۲ - جدول آزمون ها

Qid	Topic	Question	A	B	C	D	Right Answer
1	Sports	hello?	:)	:))	:))))	:()	B
2	Sports	hello2?	:)	:))	:))))	:()	A
3	Sports	hello3?	:)	:))	:))))	:()	C
4	Sports	hello4?	:)	:))	:))))	:()	D
5	Sports	hello5?	:)	:))	:))))	:()	D

شکل ۳ - جدول سوالات

QuizId	Qid	Answer
1	1 A	
1	2 B	
1	3 C	
1	4 D	
1	5 A	

شکل ۴ جدول سوالات آزمون ها

همچنین لازم به ذکر است که تصاویر بالا صرفا برای مفهوم کلی شمای دیتابیس و نحوه ذخیره داده ها بوده و نیازی نیست لزوماً تایپ داده ها و نام attribute ها همانند شکل بالا باشد. برای محدودیت های نرم افزاری گفته شده در صورت پروژه مانند سقف تعداد آزمون ها و یا محاسبه جام ها نیازی نیست در خود دیتابیس توسط ابزار های کنترلی پیاده سازی شوند و اگر فرض شود که داده ها به صورت دستی درست وارد شده اند برای ما کافیست. (نیازی به پیاده سازی trigger ها و دیگر ابزار های کنترلی نیست.)

موفق باشید

بسمه تعالیٰ

پاسخ تمرین دوم درس اصول و طراحی پایگاه داده

دکتر ممتازی

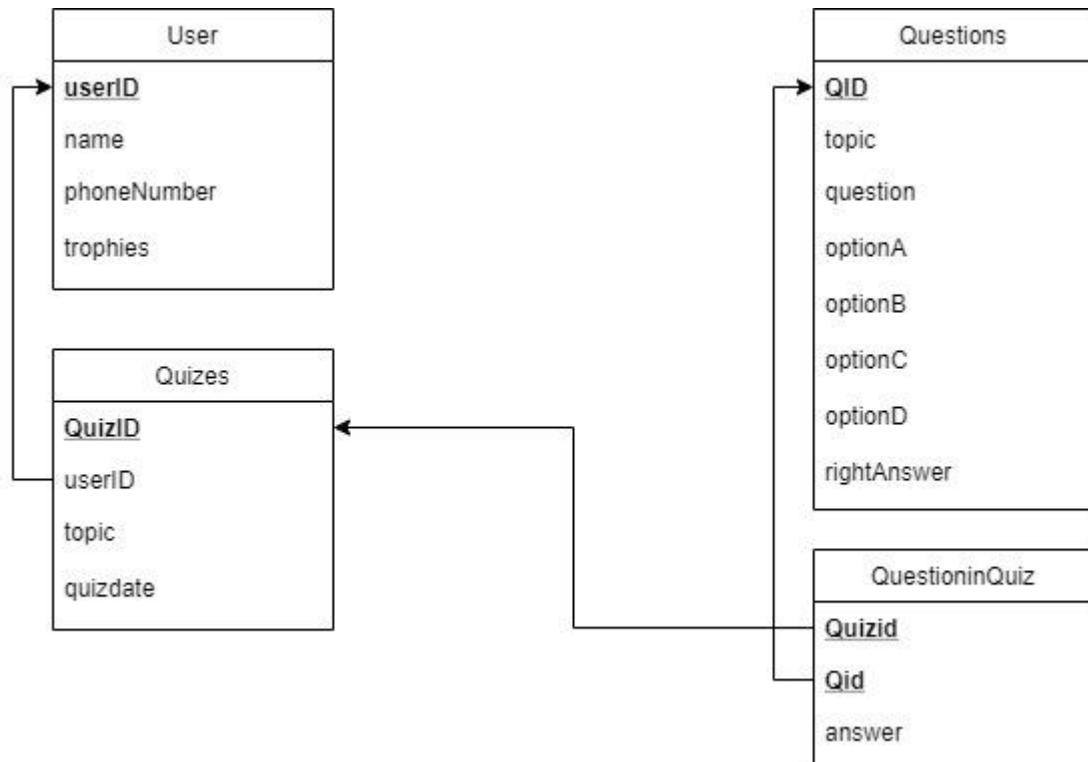
ترم پائیز ۱۳۹۹ - دانشکده کامپیوتر، دانشگاه صنعتی امیرکبیر



توجه شود که تمام این پاسخ‌ها تنها پاسخ درست این تمرین نیست و با توجه به پیاده‌سازی‌های مختلف و یا روش‌های متفاوت می‌توان به پاسخ‌های بسیار متفاوت رسید.

پاسخ:

یک راه پیاده‌سازی این دیتابیس در شما دیاگرام زیر توضیح داده شده است.



دستورات ساخت جداول:

```

CREATE TABLE users (
    userid int AUTO_INCREMENT PRIMARY KEY,
    name varchar(255) NOT null,
    phoneNumber varchar(15),
    trophies int (10) not null DEFAULT 0
);

CREATE TABLE questions (
    qid int AUTO_INCREMENT PRIMARY KEY,
    topic varchar(255) NOT null,
    question varchar(255) NOT NULL,
    optionA varchar(255) not null,
    optionB varchar(255) not null,
    optionC varchar(255) not null,
    optionD varchar(255) not null,
    rightanswer varchar(10) not null,
    CONSTRAINT rightanswer CHECK (rightanswer in ('A','B','C','D')),
    CONSTRAINT topic CHECK (topic in ('sports','history&geometry','litrature','science'))
);

CREATE TABLE quizes (
    quizid int AUTO_INCREMENT PRIMARY KEY,
    userid int NOT NULL,
    topic varchar(255) NOT null,
    quizdate datetime,
    foreign key (userid) references users(userid)
    CONSTRAINT topic CHECK (topic in ('sports','history&geometry','litrature','science'))
);

CREATE TABLE questionsinquiz (
    quizid int not null,
    qid int NOT NULL,
    answer varchar(10) not null,
    CONSTRAINT answer CHECK (answer in ('A','B','C','D')),
    foreign key (qid) references questions(qid),
    foreign key (quizid) references quizes(quizid),
    CONSTRAINT topic CHECK (topic in ('sports','history&geometry','litrature','science'))
);

```

همانطور که در صورت تمرین گفته شد، دستورات کنترلی مانند constraint check ها اجباری نبوده.

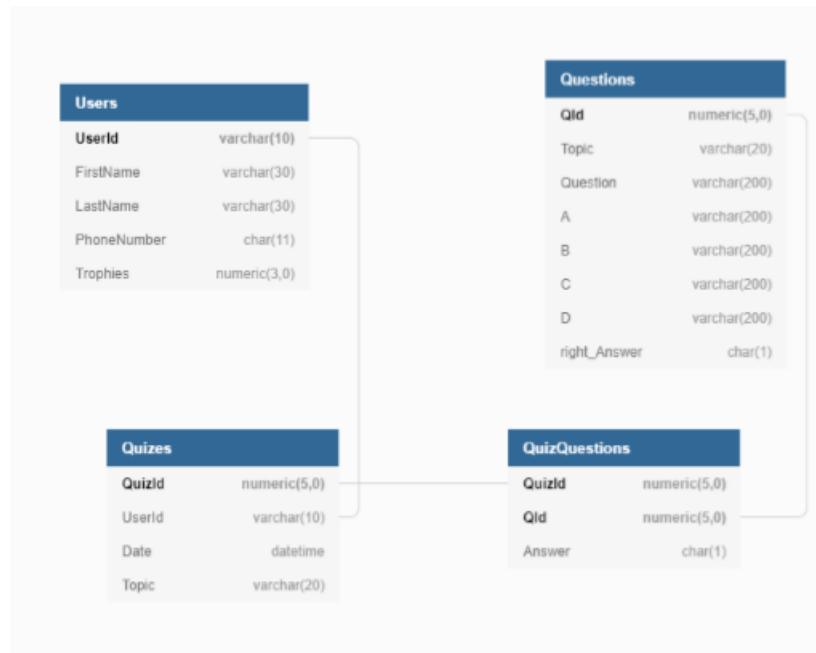
دستورات insert نیز طبق دستورات تدریس شده باید به صورت زیر باشند:

```
INSERT INTO Users (name, phoneNumber, trophies) VALUES ('Aref Motamed',  
'09121234567', 3);
```

تنها نکته مهم در این قسمت این است که باید داده های جداولی که دارای کلید خارجی هستند، بعد از داده های جداول مرجع اضافه شوند. به طور مثال ابتدا داده های جدول کاربران پر شوند و سپس داده های جدول آزمون اضافه شوند. در غیر این صورت به طور مثال ممکن است در جدول کوییز ما آزمونی برای کاربر شماره ۳ میسازیم و اگر این کاربر وجود نداشته باشد، این داده نمیتواند اضافه شود.

های خواسته شده: Query

- همانطور که گفته شد در مبحث نوشتن کوئری میتوان با روش های متفاوت، پاسخ های متفاوتی به دست آورد. به همین علت سعی بر این شد تا در کوئری هایی که کمی پیچیده تر بودند، یک پاسخ دیگر نیز داده شوند. برای پاسخ دوم یک جدول دیگر استفاده می کنیم تا یک مدل دیگر از این دیتابیس را مشاهده کنیم:



(۱) نمایش کاربرانی که نام آنها "رضا" است.

```
#1 SELECT * from users where name LIKE 'Reza%'
```

(۲) نمایش کاربران به ترتیب نزولی تعداد جام.

```
#2 SELECT * FROM users order by trophies DESC
```

(۳) کاربرانی که در یک روز مشخص آزمون داده اند.

```
#3 SELECT * from users where userid in
    (select userid from quizzes where quizdate between '2020-11-07 00:00:00' and '2020-11-07 23:59:59')
```

(۴) لیست کاربرانی که حداقل یک آزمون در موضوع "ورزشی" داده اند.

```
#4 SELECT * from users where userid in
    (select userid from quizzes where topic = 'sports')
```

(۵) تعداد آزمون هایی که در هر موضوع انجام شده.

```
#5 select topic, count(*) from quizes GROUP by topic
```

(۶) شماره گزینه انتخاب شده غالب توسط کاربران در سوالات موضوع "ورزشی"

```
#6 SELECT questionsinquiz.answer AS Favourite_Option , COUNT(*) AS quantity
from questions , quizes, questionsinquiz
where (questions.qid = questionsinquiz.qid) and (quizes.quizid = questionsinquiz.quizid) and (questions.topic='sports')
group by questionsinquiz.answer
ORDER BY quantity DESC
LIMIT 1
```

```
SELECT *
FROM (
    SELECT count(Answer) AS Numbers , Answer
    FROM quizquestions
    WHERE QuizId IN (
        SELECT QuizId
        FROM Quizes
        WHERE Topic = "Sports"
    )
    GROUP BY Answer
) AS answer_numbers
WHERE Numbers IN (
    SELECT max(Numbers)
    FROM (
        SELECT count(Answer) AS Numbers , Answer
        FROM quizquestions
        WHERE QuizId IN (
            SELECT QuizId
            FROM Quizes
            WHERE Topic = "Sports"
        )
        GROUP BY Answer
    ) AS answer_numbers
);
```

یک پاسخ دیگر

(۷) شماره گزینه صحیح غالب در کل سوالات

```
#7 SELECT rightanswer , COUNT(*) AS Quantity
from questions
group by rightanswer
ORDER BY quantity DESC
LIMIT 1
```

یک پاسخ دیگر

```
SELECT *
FROM (
    SELECT right_answer, count(right_answer) AS Numbers
    FROM Questions
    GROUP BY right_answer
) AS right_answer_numbers
WHERE Numbers IN (
    SELECT max(Numbers)
    FROM (
        SELECT count(right_answer) AS Numbers
        FROM Questions
        GROUP BY right_answer
    ) AS maximum_number
);
```

(۸) لیستی از موضوعات و تعداد سوالات ثبت شده برای هر کدام.

```
#8   SELECT topic , count(*) AS Quantity
      from questions
      group by topic
```

(۹) لیستی از کاربران و تعداد آزمون های داده شده هر کاربر.

```
#9   select userid , COUNT(*) AS Quantity
      from quizzes
      GROUP by userid
```

(۱۰) نمایش کاربران به ترتیب بیشترین پاسخ درست.

```
#10 SELECT quizzes.userid , COUNT(*) AS CorrectAnswers
       FROM questions, questionsinquiz, quizzes
      WHERE questions.rightanswer = questionsinquiz.answer AND questions.gid=questionsinquiz.gid AND quizzes.quizid= questionsinquiz.quizid
        GROUP BY quizzes.userid
        ORDER BY CorrectAnswers DESC
```

یک پاسخ دیگر

```
SELECT *
FROM (
    SELECT UserId, count(QuizId) AS RightAnswerNumbers
    FROM (
        SELECT quizquestions.QuizId
        FROM quizquestions NATURAL JOIN questions
        WHERE quizquestions.Answer = questions.right_answer
    ) AS right_answer_quizzes NATURAL JOIN quizzes
    GROUP BY UserId
) AS user_right_answer_numbers NATURAL JOIN Users
ORDER BY RightAnswerNumbers Desc;
```

(۱۱) لیست آزمون های کاربر "رضا ایرانی" و امتیاز هر کدام.

```
#11 SELECT quizes.quizid, COUNT(*)
      FROM questions , quizes, questionsinquiz
     WHERE (questions.qid = questionsinquiz.qid)
       and (quizes.quizid = questionsinquiz.quizid)
       and (questions.rightanswer = questionsinquiz.answer)
       and quizes.userid = (SELECT userid FROM users WHERE name = "Reza Irani")
    GROUP BY quizes.quizid
```

یک پاسخ دیگر

```
SELECT QuizId, count(Answer)
FROM Users NATURAL JOIN Quizes NATURAL JOIN quizquestions natural join questions
WHERE FirstName = 'Reza' AND LastName = 'Irani' AND Answer = right_answer
GROUP BY QuizID;
```

(۱۲) نمایش کاربران به ترتیب میانگین درصد پاسخ درست (اگر کامل به آزمون پاسخ داده شود، درصد پاسخ ۱۰۰ درصد می شود).

```
SELECT QuizId, IFNULL(NumberofRightAnswers, 0)/TotalAnswers * 100 as PointPercent
FROM (
    SELECT QuizId, count(QuizId) AS NumberofRightAnswers
    FROM Users NATURAL JOIN Quizes NATURAL JOIN quizquestions NATURAL JOIN Questions
    WHERE Answer = right_answer
    GROUP BY QuizId
) AS QuizWithRightAnswers
NATURAL RIGHT OUTER JOIN
(
    SELECT QuizId, count(QuizId) AS TotalAnswers
    FROM Users NATURAL JOIN Quizes NATURAL JOIN quizquestions
    GROUP BY QuizId
) AS TotalQuizes;
```

(۱۳) نمایش کاربران و امتیازات کسب شده به ترتیب نزولی.

```
#13 SELECT users.name, COUNT(*) AS Marks
      FROM users, quizes, questionsinquiz, questions
     WHERE (questions.qid = questionsinquiz.qid)
       and (quizes.quizid = questionsinquiz.quizid)
       and (questions.rightanswer = questionsinquiz.answer)
       AND (quizes.userid = users.userid)
    GROUP BY users.userid
    ORDER BY Marks DESC
```

یک پاسخ دیگر

```
SELECT *
FROM (
    SELECT UserId, count(QuizId) AS GainedPoints
    FROM (
        SELECT quizquestions.QuizId
        FROM quizquestions NATURAL JOIN questions
        WHERE quizquestions.Answer = questions.right_answer
    ) AS right_answer_quizes NATURAL JOIN quizes
    GROUP BY UserId
) AS user_right_answer_numbers NATURAL JOIN Users
ORDER BY GainedPoints Desc;
```

(۱۴) نمایش کاربرانی که حداقل یک آزمون با امتیاز کامل برای آنها ثبت شده باشد.

```
#14 SELECT U.name
FROM users U, quizzes Q
WHERE U.userid=Q.userid and
(SELECT COUNT(*)
FROM quizzes S, questionsinquiz P, questions T
WHERE S.quizid=p.quizid and p.qid=t.qid and t.rightanswer=p.answer and S.quizid=Q.quizid
GROUP BY S.quizid
)=5
```

یک پاسخ دیگر

```
SELECT *
FROM USERS
WHERE UserId in (
    SELECT UserId
    FROM (
        SELECT Users.*, QuizId, count(QuizId) AS NumberOfRightAnswers
        FROM Users NATURAL JOIN Quizes NATURAL JOIN quizquestions NATURAL JOIN Questions
        WHERE Answer = right_answer
        GROUP BY QuizId
    ) AS QuizWithRightAnswers
    NATURAL RIGHT OUTER JOIN
    (
        SELECT QuizId, count(QuizId) AS TotalAnswers
        FROM Users NATURAL JOIN Quizes NATURAL JOIN quizquestions
        GROUP BY QuizId
    ) AS TotalQuizes
    WHERE NumberOfRightAnswers /TotalAnswers = 1
);
```

(۱۵) محبوب ترین موضوع در بین موضوعات (بر اساس تعداد آزمون ثبت شده)

```
#15 SELECT topic , COUNT(quizes.quizid) as numberOfQuizes
FROM quizes
GROUP BY topic
ORDER BY numberOfQuizes DESC
LIMIT 1
```

یک پاسخ دیگر

```
SELECT *
FROM (
    SELECT Topic, count(QuizId) AS NumberOfQuizes
    FROM quizes
    GROUP BY Topic
) AS quiz_with_numbers
WHERE NumberOfQuizes in (
    SELECT max(NumberOfQuizes)
    FROM (
        SELECT Topic, count(QuizId) AS NumberOfQuizes
        FROM quizes
        GROUP BY Topic
    ) AS quiz_with_numbers
);
```

۱۶) لیستی از کاربران و تاریخ اولین آزمون ثبت شده برای آنها.

```
#16 SELECT users.name, S.quizdate
      FROM users, quizes S
     WHERE S.userid=users.userid and not EXISTS (
          SELECT *
          FROM quizes Q
         WHERE Q.userid= users.userid and Q.quizdate<S.quizdate
      )
```

یک پاسخ دیگر

```
SELECT Users.* , min(date)
      FROM Quizes NATURAL JOIN Users
     GROUP BY UserId;
```

۱۷) کاربرانی که در یک روز مشخص، ۳ و یا بیشتر آزمون داده اند.

```
#17 SELECT DISTINCT users.name
      FROM users,quizes
     WHERE users.userid =quizes.userid and (SELECT COUNT(*)
       FROM quizes S
      WHERE S.userid =users.userid and S.quizdate BETWEEN '2020-11-05 00:00:00' and '2020-11-06 00:00:00'
    ) >= 3
```

موفق باشد

بسمه تعالی

تمرین سوم درس اصول و طراحی پایگاه داده
دکتر ممتازی

ترم پائیز ۱۳۹۹ – دانشکده کامپیوتر، دانشگاه صنعتی امیرکبیر
زمان تحويل: ۱۳۹۹/۱۰/۱



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

۱. در صورت وجود هرگونه سوال یا ابهام می‌توانید با تدریسیار از طریق ایمیل tavasolireza10@gmail.com در ارتباط باشید.
۲. مهلت تحويل تمرین تا تاریخ ۱۳۹۹/۱۰/۱ می‌باشد. بعد از این تاریخ تمرین به هیچ عنوان تحويل گرفته نمی‌شود.
۳. تمامی فایل‌های خواسته شده را در قالب یک فایل فشرده با نام HW3_StudentNumber در مدل بارگذاری کنید. به عنوان مثال: HW3_9731000

در این پروژه قصد داریم پایگاه داده مربوط به یک مجموعه کتابخانه را طراحی نماییم. در ادامه این فایل ابتدا تعریف پروژه بطور مسحی ذکر می‌شود. در بخش بعدی موارد مطلوب در این تمرین ذکر می‌شود.

تعریف پروژه

این مجموعه کتابخانه دارای شعبه‌های مختلفی است. هر شعبه با شناسه یکتا، آدرس و تلفن آن مشخص می‌شود. برای هر کتاب در این مجموعه کتابخانه، مشخصاتی مانند عنوان کتاب، دسته‌بندی موضوعی کتاب، تعداد صفحات، قیمت، نام انتشارات و نویسنده کتاب نگهداری می‌شود. هر کتاب ممکن است یک یا تعداد بیشتری نویسنده داشته باشد. همچنین برای هر کتاب یک شناسه یکتا وجود دارد، اما اگر کتابی چند جلدی باشد، نیاز است که در کنار شناسه آن کتاب، شماره جلد آن کتاب نیز مشخص شود تا بتوان بصورت مجزا هر جلد از کتاب را تفکیک نمود. چنانچه کتابی تک جلدی باشد می‌توان شناسه‌ی صفر را بصورت پیش‌فرض برای شماره‌ی جلد آن در نظر گرفت. برای هر یک از انتشارات، نام آن، آدرس و وبسایت فروش آن نگهداری می‌شود. در این مجموعه کتابخانه، در هر یک از شعبه‌ها چندین انبار برای نگهداری کتاب‌ها وجود دارد که در هر انبار برای هر کتاب، یک شناسه اصلی کتاب که مرتبط به ماهیت آن کتاب است، وجود دارد و یک شناسه نیز مخصوص هر نسخه از تیراژ آن کتاب می‌باشد. برای مثال یک شناسه اصلی برای تمام کتاب‌های پایگاه داده سیلبرشاتس وجود دارد ولی هر تعداد کتاب پایگاه‌داده‌ای که چاپ شده است و در انبار وجود دارد، یک شناسه یکتا که مشخص‌کننده‌ی آن نسخه است را نیز دارا می‌باشد.

مشتریان این کتابخانه به سه دسته تقسیم می‌شوند (افراد عادی، دانشجویان و اساتید دانشگاه). مشخصات هر کدام از این افراد شامل کد ملی، نام، نام خانوادگی، تلفن و آدرس نگهداری می‌شود. هر یک از افراد می‌توانند بیشتر از یک تلفن و آدرس در سیستم داشته باشند. برای افراد عادی، شغل آن‌ها، برای دانشجویان، شماره دانشجویی و نام دانشگاه و برای اساتید، شماره شناسایی و نام دانشگاه نیز مشخص می‌شود.

هر یک از افراد برای قرض گرفتن کتاب باید یک حساب در سامانه داشته باشند. در این حساب، مقدار موجودی که آن حساب دارد و تاریخ ساخت حساب مشخص می‌شود. هر حساب دارای یک تاریخچه قرض گرفتن است. در این تاریخچه، هر بار قرض گرفتن کتاب با یک شناسه یکتا مشخص می‌شود و در آن شناسه کتابی که قرض گرفته شده، تاریخ دریافت کتاب، تعداد روزهای مجاز تا بازگشت کتاب، تاریخی که کتاب بازگردانده شده و مقداری که از موجودی حساب کاربر برداشت شده است، نگهداری می‌شود.

مشخصات مشتریان و حساب‌های آن‌ها در تمام شعبه‌ها یکسان است و هر مشتری با یک حساب می‌تواند از تمام شعبه‌های این کتابخانه، کتابی قرض بگیرد.

مطلوبات پروژه

۱. لیست تمام Entity‌های سیستم را اخراج و ذکر نمایید. تمامی Weak Entity‌ها را بطور دقیق مشخص نمایید و Strong Entity مربوط به هر یک را مشخص کنید.
۲. برای هر Entity لیست تمام Attribute‌ها را مشخص کنید. توجه کنید Primary Key‌ها و Foreign Key‌ها بطور دقیق مشخص شده باشند و در Entity‌ها مولفه و Foreign Key که به آن ارجاع داده می‌شود، مشخص باشد.
۳. لیست تمام Relation‌ها را تهیه کنید. همینطور برای هر رابطه، تمام Attribute‌ها و کلیدهای مربوطه را مشخص نمایید.
۴. ERD نهایی را بطور کامل رسم نمایید. روابط تمام Entity‌ها و Cardinality تمام روابط باید بطور دقیق در این نمودار مشخص شده باشد.
۵. تمام دستورات لازم برای ایجاد پایگاه داده مطابق با طراحی انجام شده را در یک فایل متون بنویسید بطوریکه اجرای فایل متون بطور کامل و بدون اشکال، پایگاه داده طراحی شده را ایجاد نماید.
۶. اکسپورت پایگاه داده ایجاد شده را ضمیمه فایل‌های ارسالی نمایید.

توضیحات:

۱. پاسخ سوالات ۱ الی ۴ را در قالب یک فایل PDF با نام StudentNumber.pdf در پوشه تمرين خود قرار دهید.
۲. در فایل به نام table.sql ساخت جداول مربوط به سوال ۵ را بیاورید.
۳. تمامی فایل‌های خواسته شده را در قالب یک فایل فشرده با نام HW3_StudentNumber در مدل بارگذاری کنید.

موفق باشید

بسمه تعالیٰ

پاسخ تمرین سوم درس اصول و طراحی پایگاه داده

دکتر ممتازی

ترم پائیز ۱۳۹۹ - دانشکده کامپیوتر، دانشگاه صنعتی امیرکبیر



١. لیست تمام Entity‌های سیستم را اخراج و ذکر نمایید. تمامی Weak Entity‌ها را بطور دقیق مشخص نمایید و Strong Entity مربوط به هر یک را مشخص کنید.
٢. برای هر Entity لیست تمام Attribute‌ها را مشخص کنید. توجه کنید Foreign Key‌ها و Primary Key‌ها را مشخص کنید.
٣. لیست تمام Relation‌ها را تهیه کنید. همینطور برای هر رابطه، تمام Attribute‌ها و کلیدهای مربوطه را مشخص نمایید.
٤. ERD نهایی را بطور کامل رسم نمایید. روابط تمام Entity‌ها و Cardinality را مشخص کنید. در این نمودار مشخص شده باشد.
٥. تمام دستورات لازم برای ایجاد پایگاه داده مطابق با طراحی انجام شده را در یک فایل متنی بنویسید. بطوریکه اجرای فایل متنی بطور کامل و بدون اشکال، پایگاه داده طراحی شده را ایجاد نماید.
٦. اکسپورت پایگاه داده ایجاد شده را ضمیمه فایل‌های ارسالی نمایید.

پاسخ:

Entities:

Entity Library (**library_id**, address, phone)

Entity Book (**book_id**, title, category, pages, publisher, price, edition)

publisher foreign key to Publisher(publisher_id)

Entity Publisher(**publisher_id**, name, address, URL)

Entity Author(**author_id**, book_id, name)

book_id foreign key to Book(book_id)

Entity Account(**account_id**, account_balance, created_date)

Entity Customer(**customer_id**, first_name, last_name)

Entity address(**customer_id, address**)

customer_id foreign key to Customer(customer_id)

Entity phone(**customer_id, phone_number**)

customer_id foreign key to Customer(customer_id)

Entity Faculty(**customer_id**, faculty_id, university_name)

customer_id foreign key to Customer(customer_id)

Entity Student(**customer_id**, student_id, university_name)

customer_id foreign key to Customer(customer_id)

Entity Regular(**customer_id**, job)

customer_id foreign key to Customer(customer_id)

Weak Entity loan_history(**loan_id**, book_id, start_date, return_date, changed_amount, legal_loan_duration)

Strong Entity: Account

book_id foreign key to Book(book_id)

Entity Inventory (**inventory_id**, **book_id**, **book_issue**)

book_id foreign key to Book(book_id)

Relationships:

customer_address(**customer_id**, address)

customer_phone(**customer_id**, phone)

owns_account(**customer_id**, account_id)

library_inventory(library_id, **inventory_id**, **book_id**, **book_issue**)

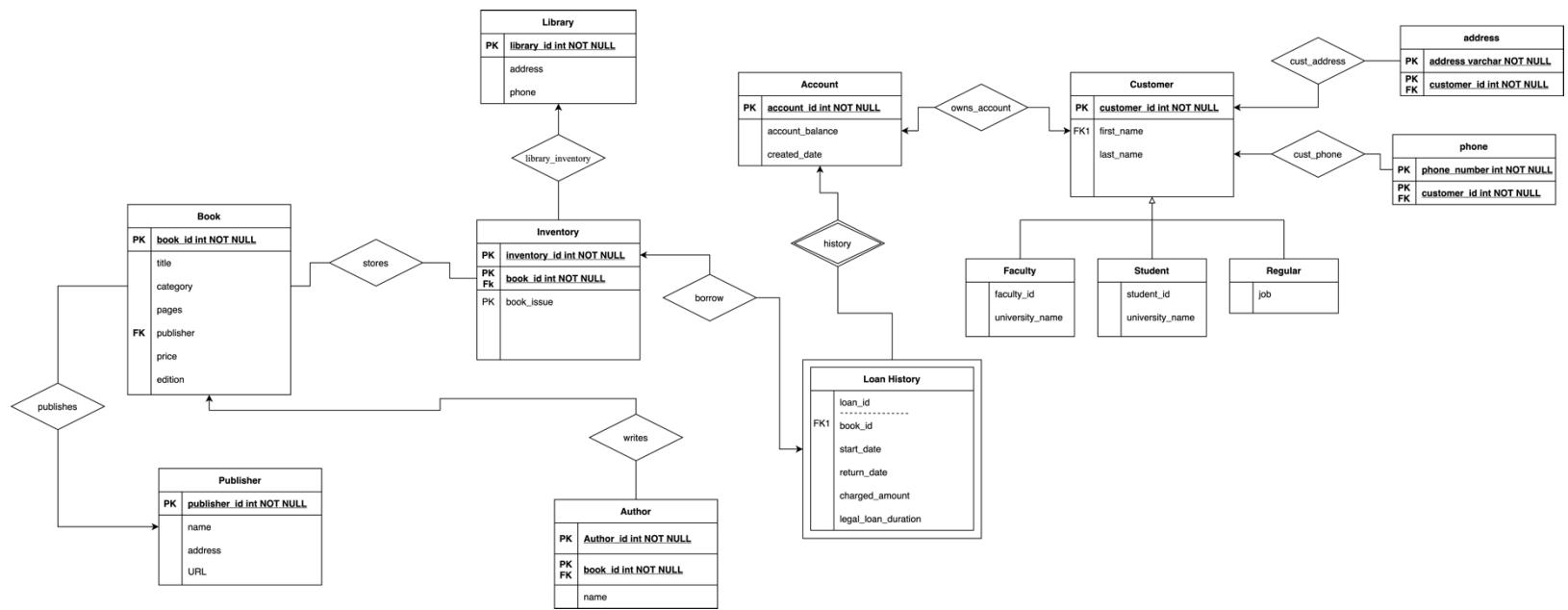
stores(**book_id**, **inventory_id**)

publishes(**book_id**, publisher_id)

writes(**book_id**, author_id)

borrow(**loan_id**, **inventory_id**, **book_id**)

history(account_id, loan_id)



بسمه تعالی

<p>تمرین چهارم درس اصول و طراحی پایگاه داده دکتر ممتازی ترم پائیز 1399 – دانشکده کامپیوتر، دانشگاه صنعتی امیرکبیر زمان تحويل: 17 دی ماه</p>	
---	---

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

1. در صورت وجود هرگونه سوال یا ابهام می‌توانید با تدریسیار از طریق ایمیل hosein.btf@gmail.com در ارتباط باشید.
2. مهلت تحويل تمرین تا تاریخ هفده دی می‌باشد. بعد از این تاریخ تمرین به هیچ عنوان تحويل گرفته نمی‌شود.
3. تمامی فایل‌های خواسته شده را در قالب یک فایل فشرده با نام HW4-StudentNumber در مدل .HW4-9531888 بارگذاری کنید. به عنوان مثال:

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

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

2. رابطه $R(A, B, C, D, E, F, G)$ با وابستگی های تابعی زیر را درنظر بگیرید.

$A \rightarrow B$

$B \rightarrow C$

$AD \rightarrow C$

$AC \rightarrow D$

$A \rightarrow F$

$AFD \rightarrow G$

$D \rightarrow FA$

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

ب) کلید های کاندید را بنویسید.

3. رابطه $R(A, B, C, D, E, F, G)$ با وابستگی های تابعی زیر را درنظر بگیرید.

$A \rightarrow C$

$BC \rightarrow G$

$CF \rightarrow E$

$AF \rightarrow D$

الف) ارضاعیتی شرط BCNF را برای هر یک از وابستگی ها چک کنید. سپس رابطه را به فرم BCNF تبدیل کنید و کلیدهای هر رابطه را مشخص کنید.

ب) اگر تجزیه‌ای به شکل $R2(A, C, G, B, F)$ و $R1(A, D, E, F)$ داشته باشیم، نشان دهید این تجزیه lossless می‌باشد.

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

$$A \rightarrow BG$$

$$A \rightarrow C$$

$$BC \rightarrow F$$

$$AF \rightarrow E$$

$$DF \rightarrow A$$

$$ACD \rightarrow B$$

$$F \rightarrow C$$

الف) کلید های کاندید را بنویسید.

ب) فرم Canonical Cover وابستگی های تابعی بالا را بنویسید.

ج) برای هریک از وابستگی های تابعی داده شده درصورت سوال، ارضاعیتی شرط BCNF را چک کنید. سپس تجزیه‌ای به فرم BCNF ارائه دهید و کلیدهای هر رابطه را مشخص کنید. با ذکر دلیل بیان کنید آیا تجزیه حاصل Dependency preserving را رعایت می‌کند یا خیر؟

د) برای هریک از وابستگی های تابعی به فرم Canonical Cover، ارضاعیتی شرط 3NF را چک کنید و کلیدهای هر رابطه را مشخص کنید. سپس تجزیه‌ای به فرم 3NF ارائه دهید.

5. جدول Profile(UserName, FullName, Email, Skill) را درنظر بگیرید. این جدول شامل چهار فیلد UserName(نام کاربری)، FullName(نام و نام خانوادگی)، Email(مجموعه ایمیل های فرد) و Skill(مجموعه مهارت ها) می‌باشد. با توجه به این طراحی به سوالات زیر پاسخ دهید.

(توجه: نام کاربری یکتا است. هر کاربر یک نام و نام خانوادگی دارد، اما می‌تواند بیش از یک ایمیل و چندین مهارت داشته باشد.)

الف) وابستگی‌ها را بنویسید و نوع هر یک را مشخص کنید.

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

ج) تجزیه‌ای ارائه دهید که از به وجود آمدن این مشکلات جلوگیری کند.

موفق باشید

بسمه تعالیٰ

پاسخ تمرین چهارم درس اصول و طراحی پایگاه داده

دکتر ممتازی

ترم پائیز ۱۳۹۹ – دانشکده کامپیوتر، دانشگاه صنعتی امیرکبیر



۱. با استفاده از Armstrong's Axioms قانون decomposition را اثبات کنید.

(پاسخ)

$X \rightarrow YZ$ then $X \rightarrow Y$ and $X \rightarrow Z$.

1. $X \rightarrow YZ$, Given

2. $YZ \rightarrow Y$, Reflexivity

3. $X \rightarrow Y$, Transitivity on 1 and 2.

Similar proof for $X \rightarrow Z$.

1. $X \rightarrow YZ$, Given

2. $YZ \rightarrow Z$, Reflexivity

3. $X \rightarrow Z$, Transitivity on 1 and 2.

۲. رابطه $R(A, B, C, D, E, F, G)$ با وابستگی های تابعی زیر را درنظر بگیرید.

$A \rightarrow B$

$B \rightarrow C$

$AD \rightarrow C$

$AC \rightarrow D$

$A \rightarrow F$

$AFD \rightarrow G$

$D \rightarrow FA$

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

ب) کلید های کاندید را بنویسید.

(پاسخ)

(الف)

D در سمت چپ $AD \rightarrow C$ اضافه است. زیرا در مجموعه وابستگی $A \rightarrow B$ و $B \rightarrow C$ می‌توان از صفت A ، ویژگی C را نتیجه گرفت.

$$F = \{A \rightarrow B, B \rightarrow C, A \rightarrow C, AC \rightarrow D, A \rightarrow F, AFD \rightarrow G, D \rightarrow FA\}$$

C در سمت چپ $AC \rightarrow D$ اضافه است. زیرا در مجموعه وابستگی $A \rightarrow B$ و $B \rightarrow C$ می‌توان از صفت A ، ویژگی C را نتیجه گرفت.

$$F = \{A \rightarrow B, B \rightarrow C, A \rightarrow C, A \rightarrow D, A \rightarrow F, AFD \rightarrow G, D \rightarrow FA\}$$

F و D در سمت چپ $AFD \rightarrow G$ اضافه است. زیرا در وابستگی $A \rightarrow D$ می‌توان از صفت A ، ویژگی D را نتیجه گرفت. همینطور در وابستگی $A \rightarrow F$ از A ، به ویژگی F می‌رسیم.

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

F در سمت راست $D \rightarrow FA$ اضافه است. زیرا در وابستگی $A \rightarrow F$ از A ، به ویژگی F می‌رسیم.

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

همینطور در مجموعه وابستگی $A \rightarrow C$ را نتیجه گرفت. پس $A \rightarrow B$ ، $B \rightarrow C$ اضافه است.

:Canonical Cover فرم

$$A \rightarrow B$$

$$B \rightarrow C$$

$$A \rightarrow D$$

$$A \rightarrow F$$

$$A \rightarrow G$$

$$D \rightarrow A$$

(ب)

با توجه به وابستگی‌های تابعی داریم $(A)^+ = ABCDFG$ و $(D)^+ = ABCDFG$. طبق رابطه $R(A, B, C, D, E, F, G)$ کلید های AE و DE می‌باشند.

۳. رابطه $R(A, B, C, D, E, F, G)$ با وابستگی های تابعی زیر را درنظر بگیرید.

$A \rightarrow C$

$BC \rightarrow G$

$CF \rightarrow E$

$AF \rightarrow D$

الف) ارضاعیت شرط BCNF را برای هر یک از وابستگی ها چک کنید. سپس رابطه را به فرم BCNF تبدیل کنید و کلیدهای هر رابطه را مشخص کنید.

ب) اگر تجزیه ای به شکل $R2(A, C, G, B, F)$ و $R1(A, D, E, F)$ داشته باشیم، نشان دهید این تجزیه lossless می باشد.

(پاسخ)

(الف)

با توجه به وابستگی های تابعی داده شده، ABF کلید کاندید می باشد.

1. $A \rightarrow C$ ✗
2. $BC \rightarrow G$ ✗
3. $CF \rightarrow E$ ✗
4. $AF \rightarrow D$ ✗

1:

$R1 = (A, C)$ CK={A}

$R2 = (A, B, D, E, F, G)$ CK={ABF}

2:

$BC \rightarrow G$ $A \rightarrow C$ $BA \rightarrow G$

$R21 = (A, B, G)$ CK={AB}

$R22 = (A, B, D, E, F)$ CK={ABF}

3:

$CF \rightarrow E$ $A \rightarrow C$ $AF \rightarrow E$

$R221 = (A, F, E)$ CK={AF}

$R222 = (A, B, D, F)$ CK={ABF}

4:

$R2221 = (A, F, D)$ CK={AF}

$R2222 = (A, B, F)$ CK={ABF}

بنابراین یک تجزیه در فرم BCNF بصورت زیر می‌باشد.

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

$R21 = (\underline{A}, \underline{B}, G)$

$R221 = (\underline{A}, \underline{F}, E)$

$R2221 = (\underline{A}, \underline{F}, D)$

$R2222 = (\underline{A}, \underline{B}, F)$

(ب)

رابطه $R(A,B,C,D,E,F,G)$ به دو زیر رابطه $R1(A,D,E,F)$ و $R2(A,C,G,B,F)$ تجزیه شده است. که این دو رابطه $R1$ و $R2$ شامل تمامی attribute های R می‌باشند. از طرفی اگر α اشتراک این دو رابطه باشد و ما $\alpha \rightarrow R1$ و $\alpha \rightarrow R2$ داشته باشیم، می‌توانیم ادعا کنیم که این تجزیه Lossless است.

اشتراک این دو جدول : AF

$(AF)^+ = (AFD)^+ = (AFDC)^+ = AFDCE$

می‌بینیم که با توجه به وابستگی‌های تابعی داده شده از AF به $AFCDE$ که شامل تمامی ویژگی‌های $R1$ می‌باشد، رسیدیم. پس این تجزیه Lossless می‌باشد.

می باشد.

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

$A \rightarrow BG$

$A \rightarrow C$

$BC \rightarrow F$

$AF \rightarrow E$

$DF \rightarrow A$

$ACD \rightarrow B$

$F \rightarrow C$

الف) کلید های کاندید را بنویسید.

ب) فرم Canonical Cover وابستگی های تابعی بالا را بنویسید.

ج) برای هریک از وابستگی های تابعی داده شده درصورت سوال، ارضاعیتی شرط BCNF را چک کنید. سپس تجزیهایی به فرم BCNF ارائه دهید و کلیدهای هر رابطه را مشخص کنید. با ذکر دلیل بیان کنید آیا تجزیه حاصل Dependency preserving را رعایت می کند یا خیر؟

د) برای هریک از وابستگی های تابعی به فرم Canonical Cover، ارضاعیتی شرط 3NF را چک کنید و کلیدهای هر رابطه را مشخص کنید. سپس تجزیهایی به فرم 3NF ارائه دهید.

(پاسخ)

الف) AD و DF کلیدهای کاندید می باشند.

(ب)

:Canonical Cover فرم

$A \rightarrow BG$

$A \rightarrow C$

$BC \rightarrow F$

$A \rightarrow E$

$DF \rightarrow A$

$F \rightarrow C$

(ج)

1. $A \rightarrow BG$ ✗
2. $A \rightarrow C$ ✗
3. $BC \rightarrow F$ ✗
4. $AF \rightarrow E$ ✗
5. $DF \rightarrow A$ OK
6. $ACD \rightarrow B$ OK
7. $F \rightarrow C$ ✗

1:

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

$R2 = (A, C, D, E, F) CK=\{AD, DF\}$

2:

$R21 = (A, C) CK=\{A\}$

$R22 = (A, D, E, F) CK=\{AD, DF\}$

4:

$R221 = (A, F, E) CK=\{AF\}$

$R222 = (A, D, F) CK=\{AD, DF\}$

3:

$BC \rightarrow F$ $A \rightarrow BC$ $A \rightarrow F$

$R2221 = (A, F) CK=\{A\}$

$R2221 = (A, D) CK=\{AD\}$

R2212 = (A, E) CK={A}

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

R1 = (A, B, G)

R21 = (A, C)

R2221 = (A, F)

R2221 = (A, D)

R2212 = (A, E)

(د)

1. $A \rightarrow BGCE$
2. $BC \rightarrow F$
3. $DF \rightarrow A$
4. $F \rightarrow C$

R1 = (A, B, G, C, E)

R2 = (B, C, F)

R3 = (D, F, A)

R4 = (E, C)

۵. جدول Profile(UserName, FullName, Email, Skill) را درنظر بگیرید. این جدول شامل چهار فیلد UserName(نام کاربری)، FullName(نام و نام خانوادگی)، Email(مجموعه ایمیل های فرد) و Skill(مجموعه مهارت ها) می باشد. با توجه به این طراحی به سوالات زیر پاسخ دهید.

(توجه: نام کاربری یکتا است. هر کاربر یک نام و نام خانوادگی دارد، اما می‌تواند بیش از یک ایمیل و چندین مهارت داشته باشد).

الف) وابستگی‌ها را بنویسید و نوع هر یک را مشخص کنید.

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

ج) تجزیه‌ای ارائه دهید که از به وجود آمدن این مشکلات جلوگیری کند.

(پاسخ)

(الف)

UserName → FullName

UserName →→ Email

UserName →→ Skill

(ب)

UserName	FullName	Email	Skill
Hosein_b	Hosein Beheshti	hosein.btf@gmail.com	Database design
Hosein_b	Hosein Beheshti	hosein.btf@gmail.com	php
Hosein_b	Hosein Beheshti	hosein@gmail.com	Database design
Hosein_b	Hosein Beheshti	hosein@gmail.com	php

همانطور که می‌بینیم در این طراحی فیلد های multivalue که ارتباطی با هم ندارند باعث ایجاد سطر های اضافی می‌شوند. و برای جلوگیری از این پیشامد می‌توانیم با استفاده از الگوریتم 4nf جدول را تجزیه کنیم.

(ج)

R1(UserName, FullName)

R2(UserName, Email)

R3(UserName, Skill)

<p>پروژه درس اصول و طراحی پایگاه داده</p> <p>دکتر ممتازی</p> <p>ترم پائیز ۱۳۹۹ – دانشکده کامپیوتر، دانشگاه صنعتی امیرکبیر</p> <p>زمان تحويل: ۹۹/۱۱/۲۰</p>	
---	---

لطفاً پیش از شروع به حل کردن تمرین به نکات زیر توجه فرمایید:

۱. در صورت وجود هرگونه سوال یا ابهام می‌توانید با تدریسیار از طریق ایمیل alirezasalehy@aut.ac.ir در ارتباط باشید.
۲. مهلت تحويل پروژه تا تاریخ ۹۹/۱۱/۲۰ می‌باشد.
۳. همه کدهای SQL خود را در یک فایل .sql قرار دهید.
۴. فایل .sql را به همراه فایل‌های برنامه رابط کاربری به همراه تمامی کتابخانه‌های استفاده شده (به صورتی که در روی دستگاه‌های دیگر قابل اجرا باشد)، در قالب یک فایل zip با فرمت زیر در سایت بارگذاری نمایید. "StudentNumber>.zip>"

در این پروژه می بایست پایگاه داده ای را که برای کتابخانه زنجیره ای در تمرین 3 طراحی کردید را پیاده سازی کنید. در صورت نیاز به بازخوانی تعریف پروژه می توانید به تمرین 3 مراجعه کنید. توجه داشته باشید که در این پروژه برای سادگی فرض می شود که تنها یک شعبه و تنها یک انبار داشته باشیم. در نتیجه نیازی به جدول برای انبار و شعبه نیست.

هدف از این صورت پروژه، پیاده سازی مدل پایگاه داده، کار با SQL به صورت مستقیم و پیاده سازی روابط و مدیریت داده ها و کار با مدل پیاده سازی شده از یک رابط کاربری جداگانه می باشد.

توجه داشته باشید که ساختار هایی که برای موجودیت ها گفته می شود ممکن است کامل نباشد و شما باید بر اساس صورت پروژه که در تمرین سه بود و صلاح دید خود ویژگی هایی را به آن بیافزایید. ولی در معماری خود اصول نرمال سازی شماتیک پایگاه داده برای پیشگیری از افروندگی و... را به کار ببرید.

در این پروژه باید برای همگی توانمندی های خواسته شده یک تابعی مجزا با ورودی های مناسب تعریف کنید و که این تابع ها بدست رابط کاربری فراخوانی می شوند.

توجه داشته باشید که می توانید نرم افزار های DBMS دلخواه مانند؛ MySQL یا MariaDB را برای پیاده سازی پایگاه داده و هر یک از زبان های برنامه نویسی دلخواه برای پیاده سازی رابط کاربری، به کار ببرید.

نکات کلی:

- همه مورد های خواسته شده باید کاملا در SQL پیاده سازی شوند، به این صورت که رابط کاربری تنها خروجی ها گرفته شده از پایگاه داده را سامان و نمایش می دهد.
- برای نگه داری رشته های طولانی می توانید طول 512 را در نظر بگیرید.
- برای نگه داری شماره تلفن و یا شماره های دیگر از ویژگی های عددی استفاده نکنید. زیرا صفر آغازین آن ها از دست می روند.
- مواردی که با "به صورت خودکار" توصیف شده اند، باید با trigger پیاده سازی شوند.
- **موارد امتیازی با سبز نمایش داده شده اند.**

حساب کاربری

هر مشتری دارای یک حساب کاربری می باشد، اطلاعات هر فرد به دو بخش اطلاعات سیستمی مانند "نام کاربری و گذرواژه" و اطلاعات شخصی مانند "نام آدرس شماره تلفن و..." دسته بندی می شود.

اطلاعات سیستمی :

این بخش داده های زیر را در بر دارد:

- نام کاربری:

یکتا می باشد و باید دست کم 6 کاراکتر از میان حروف الفبا و رقم باشد، نام کاربری به هنگام نمایش داده شدن حساس به کوچک و بزرگ بودن حرف می باشد و به هنگام احزار هویت یا ساخت حساب کاربری بزرگ یا کوچک بودن حرف ها مهم می باشد. برای نمونه دو کاربر با نام کاربری farhad Farhad مجاز نمی باشد. و همچنین اگر کاربری که با نام Farhad ثبت نام کرده باشد به هنگام نمایش به کاربر باید به همین صورت نمایش داده شود.

- گذرواژه :

گذرواژه نیز باید دست کم 8 کاراکتر دربردازندۀ حروف الفبا و رقم باشد (هر دو)، گذرواژه به هنگام دریافت باید در پایگاه داده به صورت درهم ریخته (Hashed) نگه داری شود. و در هنگام ورود هر فرد به حساب خود باید گذرواژه وارد شده درهم ریخته شود و با آنچه در پایگاه داده ذخیره شده است سنجیده شود، برابر بودن این دو مقدار درهم ریخته نشان دهنده درستی گذرواژه می باشد.

- زمان ساخت حساب:

تاریخ ساخت حساب هر فرد در هنگام ثبت نام باید در پایگاه داده نگه داری شود.

اطلاعات شخصی:

این بخش داده های زیر را در بر دارد:

- آدرس مکان زندگی

- نام

- نام خانوادگی

- نوع کاربری (دانشجو، استاد، عمومی)

- و....

ثبت نام:

مشتری ها برای اینکه بتوانند از خدمات سامانه بهره مند شوند نخست باید حساب کاربری بسازند. برای این کار کاربر باید همه داده های اطلاعات شخصی و سیستمی به جز تاریخ ثبت نام و شماره آیدی خود را وارد کنند. عملیات ثبت نام باید در یک تراکنش انجام شود (Transaction) که چنانچه مشکلی پیش آمد و مثلاً ورودی های "Farhad" و "farhad" یا اینکه گفته شده برای اطلاعات حساب کاربری را نداشت، برای نمونه؛ نام کاربری تکراری باشد ("Farhad") یا اینکه گذروازه هم رقم و هم حرف را در بر نداشته باشد، اگر عملیاتی بر روی جدول ها انجام شده بود بازگردانی شود.

توجه شود که همان گونه که بالا تر گفته شد همه پیاده سازی ها و حتی پیام های خطاب نیز باید در SQL پیاده سازی شود و رابط کاربری تنها خروجی را در صورت موفقیت آمیز بودن یا روبه رو شدن با خطاب به کاربر نشان می دهد.

پیام های خروجی باید متناسب با خطاب باشند.

ورود به حساب:

کاربران برای ورود باید نام کاربری و گذروازه خود را در رابط کاربری وارد کنند. در SQL بررسی می شود که آیا در هم ریخته گذروازه وارد شده برابر با مقدار در هم ریخته نگه داری شده برای حساب کاربر می باشد یا نه. و اگر برابر نبودند پیام مناسب را بر می گرداند. توجه داشته باشید که احراز هویت (وارد کردن نام کاربری و گذروازه) تنها باید یک بار و در ورود کاربر به حساب خود انجام شود. و انجام عملیات های دیگر نام و گذروازه کاربر نباید دوباره به پایگاه داده فرستاده شود.

برای این کار می توانید شناسه ای را به عنوان خروجی عملیات ورود باز گردانید که این تگ می تواند در هم ریخته "نام کاربری + گذروازه + برچسب زمانی درخواست ورود" باشد که با عملیات ورود کاربر به حساب کاربری به وی اختصاص داده می شود و با عملیات خروج از حساب شناسه منقضی خواهد شد.

در عملیات های بعدی تگ گرفته شده در عملیات ورود، به همراه دیگر مقدار های لازم برای آن عملیات به پایگاه داده فرستاده می شود. و به این صورت مشخص می شود کدام کاربر عملیات مورد نظر را انجام میدهد.

یا به جای استفاده از تگ میتوان عملیات های درخواست شده از پایگاه داده مانند افزایش موجودی را برای آخرین فردی که به سامانه وارد شده است انجام دهید. برای اینکار می توانید از یک جدول استفاده کنید زمان ورود هر

کاربر به حساب کاربری رو خود را ثبت میکند و برای انجام عملیات افزایش موجودی، عملیات را برای آخرین کاربر وارد شده به حساب انجام می دهد. (بدیهی است این روش نسبت به استفاده از تگ محدودیت دارد.)

گرفتن اطلاعات حساب:

هر کاربر می بایست بتواند اطلاعات شخصی و سیستمی حساب خود را ببیند.

عملیات های کاربر

کاربر می تواند عملیات های گرفتن کتاب، پس دادن کتاب، جستجو کتاب، افزایش موجودی را انجام دهد. نوع کاربری کاربران سطح دسترسی آن ها را مشخص می کند.

جستجو کتاب:

کاربران می توانند با نام کتاب، نام نویسنده، نسخه و زمان چاپ کتاب ها را جستجو کنند. توجه داشته باشید که هر ترکیبی از 4 ویژگی ممکن است در جستجو به کار برود. برای نمونه "نام نویسنده + نام کتاب" یا "نام نویسنده + زمان چاپ". نتیجه‌ی جستجو باید بر اساس نام کتاب مرتب شده باشد.

گرفتن کتاب:

کاربران بر حسب نوع حساب کاربری خود می توانند بر مجموعه ای از کتاب ها دسترسی داشته باشند به این صورت که کاربر عمومی می تواند همه کتاب ها مگر کتاب های آموزشی دانشگاهی و کتاب های مرجع، دانشجو همه کتاب ها مگر کتاب های مرجع و استاد نیز همه کتاب ها را درخواست دهد. (با اینکه این دسته بندی ممکن است غیر واقعی باشد ولی برای بالاتر بردن کیفیت آموزشی این پروژه در نظر گرفته شده است)

همچنین در گرفتن کتاب موجودی حساب کاربر و تاریخچه وی و موجودی انبار نیز بررسی می شود به این صورت که کاربر با 4 بار دیرکرد در پس دادن کتاب در بازه دوماهه اخیر، به اندازه یک ماه محروم می شود.

هزینه گرفتن هر کتاب نیز به اندازه 5 درصد بهای کتاب می باشد و هزینه گرفتن کتاب از موجود حساب وی کاسته می شود. اگر موجودی حساب کاربر کمتر از این مقدار داشت نمی تواند کتاب را بگیرد.

همان گونه که پیش از این گفته شد همه این موارد به همراه خروجی مناسب در صورت موفقیت یا شکست عملیات باید در SQL پیاده سازی شود.

با موفقیت آمیز بودن درخواست باید یکی از کتاب های انبار کاهش یابد. چه عملیات موفقیت آمیز باشد و چه نباشد باید در تاریخچه ثبت گردد. و نتیجه نیز می تواند موفقیت آمیز بودن، محرومیت یا نبود موجودی مالی یا نبود کتاب باشد باید در تاریخچه ثبت گردد. (برای سادگی فرض کنید نتیجه تنها یکی از این ها خواهد بود. یعنی به ترتیب سه مشکل محرومیت، نبود موجودی مالی یا نبود کتاب بررسی می شود و در صورت مشاهده اولین مشکل همان ثبت می شود. مثلا اگر هم موجودی حساب کافی نبود و هم کتاب نبود عدم موجودی حساب ثبت شود).

همچنین به صورت خودکار برای مسئولین کتابخانه پیام رویداد درخواست موفقیت آمیز کتاب با جزئیات آن ثبت می شود. (به صورت متنی، مانند در تاریخ A کاربر B کتاب C را به صورت موفقیت آمیز درخواست داده است).

تحویل کتاب:

با تحویل کتاب به کتابخانه یک واحد به موجودی انبار افزوده می شود. و این عملیات در تاریخچه ثبت میگردد. همچنین به صورت خودکار پیامی برای توصیف رویداد در صندوق پیام مسئولین ثبت می شود. که در این پیام با تاخیر بودن یا بدون تاخیر بودن این تحویل نیز نمایش داده می شود.

افزایش موجودی:

کاربر می تواند موجودی خود را افزایش دهد، به بازه ای که به عنوان ورودی به تابع مربوط به این عملیات داده می شود نیز باید دقت شود و مثلا اگر ورودی تابع برابر با 5 باشد پیام خطای مناسبی بازگردانده شود.

کتاب

کتاب ها دارای ویژگی های گفته شده در صورت پروژه می باشند.

مسئول کتابخانه

کتاب خانه می تواند یک یا چندین مسئول کتابخانه داشته باشد که همانند کاربران دارایی حساب کاربری می باشند و نوع کاربری برای حساب آن ها نیز مدیر و کتابدار می باشد.

کتابدار و مدیر این توانایی را دارند که کتاب هایی را به انبار بافزايد، همچنین می توانند کاربران کتابخانه را نیز جستجو کنند و تاریخچه و اطلاعات شخصی آن ها را ببینند. مسئولین کتابخانه یک صندوق پیام مشترک دارند که رویداد های درخواست های پذیرفته شده برای گرفتن کتاب و پس دادن کتاب در آن ثبت می شود.

ولی تنها مدیر می تواند حساب کاربری یک کاربر را حذف کند. پس از حذف حساب کاربری یک کاربر باید همه ردیف های مربوط به آن کاربر در جدول های مختلف پاک گردد. (می توانید از کلید خارجی برای این کار بهره ببرید)

افزودن کتاب:

مسئول کتابخانه می تواند چند نسخه از یک کتاب را به موجودی انبار بیافزاید. یا در صورت نبود کتاب یک ردیف تازه برای آن در نظر بگیرد.

لیست درخواست های گرفتن کتاب پذیرفته شده:

مسئول کتابخانه می بایست بتواند درخواست هایی را که موفقیت آمیز بوده اند ببیند. لیست درخواست ها باید به صورت صفحه بندی شده و بر اساس تاریخ از زودترین به دورترین مرتب شده باشد. و اگر 10 درخواست وجود داشته باشد با درخواست صفحه 1، 5 نتیجه نخست و با درخواست صفحه 2، 5 نتیجه دیگر نمایش داده شود.

لیست کتاب هایی که زمان تحويل آن ها گذشته:

مسئول کتابخانه باید بتواند کتاب هایی را که زمان تحويل آن ها گذشته است ببیند. کتاب ها بر اساس بیشترین تاخیر به کمترین تاخیر باید فهرست شوند.

تاریخچه افرادی که یک کتاب را گرفته اند:

مسئول باید این توانایی را داشته باشد که تاریخچه گرفتن و پس داده شدن یک کتاب را مشاهده کند. تاریخچه باید به ترتیب زمانی از نزدیک ترین به دور ترین مرتب شده باشد.

جستجوی کاربران:

کاربر مسئول کتابخانه باید بتواند کاربران را بر اساس نام کاربری یا نام خانوادگی جستجو کند. در صورت وجود بیش از یک نتیجه (جستجو با نام خانوادگی) باید به ترتیب نام خانوادگی مرتب شده باشند. و نتیجه ها صفحه بندی شده باشند به این صورت که اگر 10 حساب در نتیجه جستجو بود، با درخواست صفحه یک 5 نتیجه نخست و با درخواست صفحه دوم 5 نتیجه دیگر خروجی داده شود.

توجه داشته باشید که تنها کاربران مسئول باید بتوانند دیگر کاربران را جستجو کنند یا اطلاعات آن ها را ببینند و این مورد باید در کد SQL شما بررسی گردد.

دیدن اطلاعات شخصی و تاریخچه کاربر:

کاربر مسئول می توانند اطلاعات شخصی و تاریخچه عملیات های کاربران را ببینند. تاریخچه عملیات ها باید بر اساس تاریخ از نزدیک ترین به دور ترین مرتب شده باشد.

کار با رابط کاربری

- برای پیاده سازی رابط کاربری می توانید از زبان برنامه نویسی یا اسکریپت دلخواه خود بهره ببرید.
- تنها کار رابط کاربری تفسیر ورودی هار کاربر و فراخوانی تابع های SQL با ورودی های مناسب و ساماندهی (در صورت نیاز) و نمایش خروجی این تابع ها می باشد.
- برای رابط کاربری پیاده سازی برنامه که با کنسول ورودی و خروجی بگیرد، بسنده می کند.
- خروجی می باید به صورت جدول باشد که گونه ای که نام ستون ها (ویژگی ها) و مقدار آن ها به طور مناسبی نمایان باشد. (چه در GUI و چه در CLI)

نکات پایانی

- افرادی که رابط کاربری را پیاده سازی نمی کنند و تنها بخش SQL را انجام می دهند. 80 درصد نمره را خواهند گرفت.
- افرادی که تنها رابط کاربری را پیاده سازی کنند، نمره ای دریافت نخواهند کرد.
- منطق سامانه نباید در رابط کاربری پیاده سازی شود.