


بسمه تعالی

<p>تمرین اول درس اصول و طراحی پایگاه داده دکتر ممتازی ترم پاییز 1400 – دانشکده کامپیوتر، دانشگاه صنعتی امیرکبیر زمان تحویل: 10 آبان 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 در ارتباط باشید.
2. مهلت تحویل تمرین تا تاریخ 10 آبان ماه می‌باشد. **بعد از این تاریخ تمرین به هیچ عنوان تحویل گرفته نمی‌شود.**
3. تمامی فایل‌های خواسته شده را در قالب یک فایل zip با نام HW1-StudentNumber در مودل بارگذاری کنید. به عنوان مثال: HW1-9631032.

-۱

- A. مفهوم data integrity در پایگاه‌های داده را توضیح دهید و این ویژگی را با فایل سیستم‌ها مقایسه کنید.
- B. برای هریک از داده‌های بدون ساختار، نیمه ساختارمند و ساختارمند مثالی بیاورید.
- C. کلید واژه on delete cascade در چه مواردی استفاده می‌شود و چه تاثیری دارد.
- D. پنج قسمت اصلی یک DBMS را نام برده و یکی را به دلخواه توضیح دهید.
- E. سه جدولی که به نظر شما در پایگاه داده اینستاگرام وجود دارند را نام ببرید و ستون‌ها و رابطه‌هایی که دارند را ذکر کنید.

-۲

جداول زیر را برای یک فروشگاه آنلاین در نظر بگیرید.

user (user_id, name, address, gender)

product (product_id, name, category, price, created_at)

purchases (user_id, product_id, is_paid_online, date)

توضیحات:

category: مشخص کننده دسته‌بندی محصولات

is_paid_online: مشخص کننده پرداخت آنلاین یا پرداخت نقدی

توصیفات خواسته شده را به صورت جبر رابطه‌ای بنویسید.

- A. نام کاربرانی که تا به حال خریدی نداشته‌اند.
- B. نام کاربرانی که: آقا هستند و از محصولات دسته "suit" خرید داشته‌اند یا خانم هستند و از محصولات دسته "shoe" خرید داشته‌اند.
- C. میانگین قیمت محصولاتی که به صورت آنلاین پرداخت شده‌اند.
- D. محصولاتی که از سال ۱۳۹۰ به بعد تولید شده‌اند و تا به حال به صورت پرداخت نقدی خریداری شده‌اند.

۳-

فرض کنید که مدیریت پایگاه داده سایت آموزش آشپزی را بر عهده دارید و جدول های آن به صورت زیر هستند.

پاسخ های مربوط به این سوال را در فایل با نام 3.sql همراه بقیه پاسخ های خود ارسال کنید.

جدول کاربران (user)

- name: شناسه کاربری (یکتا می باشد)
- cellphone: شماره تلفن (نمی تواند تکراری باشد)
- referred_by: شناسه یکتا کاربری که ایشان را دعوت کرده (میتواند خالی باشد)

جدول غذاها (foods)

- id: شناسه یکتا عددی غذا
- name: نام غذا
- author: شناسه کاربری که دستور این غذا را نوشته
- recipe: متن دستور پخت غذا
- time: زمان لازم برای پخت غذا به دقیقه

جدول مواد اولیه (ingredients)

- id: شناسه ماده اولیه
- name: نام ماده اولیه
- price: هزینه ماده (باید بر ۵۰۰ بخش پذیر باشد)

جدول مواد اولیهی غذاها (food_ingredients)

- food_id: شناسه غذا
- ingredient_id: شناسه ماده اولیه
- amount: مقدار لازم از ماده اولیه

جدول مواد اولیه‌ی که کاربران در دسترس دارند (user_ingredients)

- user_name: شناسه کاربر
- ingredient_id: شناسه ماده اولیه
- amount: مقدار ماده اولیه که کاربر دارد

جدول نظرات کاربران درباره غذا (comments)

- Id: شناسه نظر
- user_name: شناسه کاربر نظر دهنده
- food_id: شناسه غذا (هر کاربر به هر غذا فقط ۱ بار میتواند امتیاز دهد)
- rate: امتیاز (باید بین ۱ تا ۵ باشد)
- comment: نظر (نظر نوشته شده نباید شامل کلمه "bimaze" باشد)

ساخت جدول‌ها:

ابتدا لازم است که دستورات sql که جداولی را مطابق توضیحات بالا ایجاد می کنند را بنویسید.

وارد کردن اطلاعات:

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

گزارش گیری:

برای موارد خواسته شده، کوئری های sql بنویسید (در فایل مربوط).

- A. نام تمامی غذاهایی که در اسم آن‌ها کلمه polo وجود دارد را برگردانید
- B. میانگین امتیازاتی که هر کاربر داده است را برگردانید
- C. میانگین امتیازاتی که تمامی غذاهای هر کاربر دریافت کرده است را برگردانید. (اگر غذایی ندارد در لیست حضور نداشته باشد)
- D. نظری که کاربران به غذاهای خودشان داده اند را پاک کنید.
- E. کاربری که بیشترین اعضا را به سایت دعوت کرده را برگردانید.
- F. غذاهایی که هیچ مواد اولیه ای برای آن‌ها ثبت نشده است را پاک کنید.
- G. محتویات جدول غذاها را به همراه ستون جدیدی با نام total_price برگردانید. مقدار این ستون باید برابر با جمع حاصل ضرب قیمت واحد مواد اولیه‌ی هر غذا در تعدادشان باشد. اگر ماده‌ی اولیه‌ای برای غذا ثبت نشده بود، مقدار این ستون باید صفر باشد.
- H. کاربری که مجموع ارزش مواد اولیه در دسترس‌اش بیشترین مقدار را دارد را برگردانید.
- I. شناسه‌ی غذاهایی که کاربر با نام کاربری farbod با استفاده از مواد اولیه‌ای که در اختیار دارد می‌تواند بپزد را برگردانید.
- J. شناسه، عنوان، میانگین امتیازات و تعداد آرای ۱۰ غذایی که بالاترین میانگین امتیاز را کسب کرده‌اند و حداقل ۵ نظر برایشان ثبت شده را برگردانید.

- K. قیمت ۳ ماده غذایی که کمترین قیمت را دارند را ۳ برابر کنید.
- L. میانگین زمان پخت غذاهایی که حداقل یک ماده‌ای با قیمت بیشتر از ۱۰۰۰ دارند را برگردانید.

۴- برای هرکدام از موارد زیر، دستور sql مناسب بنویسید.
پاسخ‌های مربوط به این سوال را در فایل با نام 4.sql همراه بقیه پاسخ‌های خود ارسال کنید.

جدولی از دانشجویان شامل شماره دانشجویی و اسم دانشجویان داریم.

students (student_id, name)

جدول دیگری از دوست بودن دانشجویان با هم داریم (به ازای هر دو دانشجوی دوست، دو سطر در این جدول به صورت (x,y) و (y,x) داریم).

friends (student_id, friend_id)

همچنین جدول دیگری از نمرات درس پایگاه داده دانشجویان داریم.


db_grades

(student_id, grade)

- A. نام دانشجویان را مرتب شده اساس تعداد دوست‌هایشان برگردانید (نزولی)
- B. دانشجویانی که خودشان نمره مردودی کسب کرده اند و دوستی دارند که نمره قبولی گرفته را برگردانید.
- C. دانشجویانی که تمامی دوستانشان مردود شده‌اند را برگردانید.
- D. دانشجویانی که دوستی دارند که از آن‌ها نمره بیشتر و همچنین دوستی دارند که از آن‌ها نمره کمتر گرفته را برگردانید.
- E. (امتیازی) واریانس نمره دانشجویان مردود شده را برگردانید.

موفق باشید

بسمه تعالی

<p>پاسخ تمرین اول درس اصول و طراحی پایگاه داده دکتر ممتازی ترم پاییز 1400 – دانشکده کامپیوتر، دانشگاه صنعتی امیرکبیر</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 در ارتباط باشید.
2. مهلت تحویل تمرین تا تاریخ 10 آبان ماه می‌باشد. **بعد از این تاریخ تمرین به هیچ عنوان تحویل گرفته نمی‌شود.**
3. تمامی فایل‌های خواسته شده را در قالب یک فایل zip با نام HW1-StudentNumber در مودل بارگذاری کنید. به عنوان مثال: HW1-9631032.

-۱

- A. مفهوم data integrity در پایگاه‌های داده را توضیح دهید و این ویژگی را با فایل سیستم‌ها مقایسه کنید.
- B. برای هریک از داده‌های بدون ساختار، نیمه ساختارمند و ساختارمند مثالی بیاورید.
- C. کلید واژه on delete cascade در چه مواردی استفاده می‌شود و چه تاثیری دارد.
- D. پنج قسمت اصلی یک DBMS را نام برده و یکی را به دلخواه توضیح دهید.
- E. سه جدولی که به نظر شما در پایگاه داده اینستاگرام وجود دارند را نام ببرید و ستون‌ها و رابطه‌هایی که دارند را ذکر کنید.

- A. یک پارچگی داده به اطمینان از صحت و کامل بودن داده گفته می‌شود. در dbms های امروزی، شروطی که داده‌ها می‌بایست داشته باشند سنجیده می‌شود و سپس اجازه عملیات داده می‌شود، این درحالی است که در فایل سیستم، این اطمینان از یک‌پارچگی وجود ندارد.
- B. بدون ساختار: یک قطعه موسیقی
نیمه ساختارمند: یک فایل json
ساختارمند: یک جدول در دیتابیس رابطه‌ای
- C. در یک دیتابیس رابطه‌ای، برخی از سطون‌های یک جدول اشاره‌گری به سطون دیگری از جدول دیگری هستند و درواقع موجودیت آن‌ها با اشاره به سطر والد تعریف می‌شود، حال اگر بخواهیم وقتی که سطر والد حذف می‌شود، سطر فرزند نیز حذف شود از این دستور استفاده می‌کنیم.
- D. ۱. پردازش گر کوئری (Query Processor): رابطی بین کاربران و dbms engine که کوئری‌های ورودی کاربر را به مدلی قابل فهم برای engine تبدیل می‌کند.
۲. Database engine
۳. File manager
۴. Transaction manager
۵. Authorization

E.

Users (user_id, name, joining_date)

Follows (follower_id, following_id)

Photos (id, user_id, caption, date, location)

لینک‌های مرتبط برای مطالعه بیشتر ([database schema](#), [system design](#))

-۲

جداول زیر را برای یک فروشگاه آنلاین در نظر بگیرید.

user (user_id, name, address, gender)

product (product_id, name, category, price, created_at)

purchases (user_id, product_id, is_paid_online, date)

توضیحات:

category: مشخص کننده دسته‌بندی محصولات

is_paid_online: مشخص کننده پرداخت آنلاین یا پرداخت نقدی

توصیفات خواسته شده را به صورت جبر رابطه‌ای بنویسید.

- A. نام کاربرانی که تا به حال خریدی نداشته‌اند.
- B. نام کاربرانی که: آقا هستند و از محصولات دسته "suit" خرید داشته‌اند یا خانم هستند و از محصولات دسته "shoe" خرید داشته‌اند.
- C. میانگین قیمت محصولاتی که به صورت آنلاین پرداخت شده‌اند.
- D. محصولاتی که از سال ۱۳۹۰ به بعد تولید شده‌اند و تا به حال به صورت پرداخت نقدی خریداری شده‌اند.

- A. $\pi_{user.name}(user) - \pi_{user.name}(user \bowtie purchases)$
- B. $\pi_{user.name}(\sigma_{user.gender = 'male' \wedge product.category = 'suit'}(user \bowtie purchases \bowtie product) \cup \sigma_{user.gender = 'female' \wedge product.category = 'shoe'}(user \bowtie purchases \bowtie product))$
- C. $avg(\pi_{product.price}(\sigma_{purchases.is_paid_online = 1}(purchases \bowtie product)))$
- D. $\sigma_{product.created_at \geq 1390 \wedge purchases.is_paid_online = 0}(product \bowtie purchases)$

-۳

فرض کنید که مدیریت پایگاه داده سایت آموزش آشپزی را بر عهده دارید و جدول های آن به صورت زیر هستند.

پاسخ‌های مربوط به این سوال را در فایل با نام 3.sql همراه بقیه پاسخ‌های خود ارسال کنید.

پاسخ در فایل 3.sql

جدول کاربران (user)

- name: شناسه کاربری (یکتا می‌باشد)
- cellphone: شماره تلفن (نمی‌تواند تکراری باشد)
- referred_by: شناسه یکتا کاربری که ایشان را دعوت کرده (میتواند خالی باشد)

جدول غذاها (foods)

- Id: شناسه یکتا عددی غذا
- name: نام غذا
- author: شناسه کاربری که دستور این غذا را نوشته
- recipe: متن دستور پخت غذا
- time: زمان لازم برای پخت غذا به دقیقه

جدول مواد اولیه (ingredients)

- Id: شناسه ماده اولیه
- name: نام ماده اولیه
- price: هزینه ماده (باید بر ۵۰۰ بخش پذیر باشد)

جدول مواد اولیه‌ی غذاها (food_ingredients)

- food_id: شناسه غذا
- ingredient_id: شناسه ماده اولیه
- amount: مقدار لازم از ماده اولیه

جدول مواد اولیه‌ی که کاربران در دسترس دارند (user_ingredients)

- user_name: شناسه کاربر
- ingredient_id: شناسه ماده اولیه
- amount: مقدار ماده اولیه که کاربر دارد

جدول نظرات کاربران درباره غذا (comments)

- id: شناسه نظر
- user_name: شناسه کاربر نظر دهنده
- food_id: شناسه غذا (هر کاربر به هر غذا فقط ۱ بار میتواند امتیاز دهد)
- rate: امتیاز (باید بین ۱ تا ۵ باشد)
- comment: نظر (نظر نوشته شده نباید شامل کلمه "bimaze" باشد)

ساخت جدول‌ها:

ابتدا لازم است که دستورات sql که جداولی را مطابق توضیحات بالا ایجاد می کنند را بنویسید.

وارد کردن اطلاعات:

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

گزارش گیری:

برای موارد خواسته شده، کوئری های sql بنویسید (در فایل مربوط).

- نام تمامی غذاهایی که در اسم آن‌ها کلمه polo وجود دارد را برگردانید
- میانگین امتیازاتی که هر کاربر داده است را برگردانید
- میانگین امتیازاتی که تمامی غذاهای هر کاربر دریافت کرده است را برگردانید. (اگر غذایی ندارد در لیست حضور نداشته باشد)
- نظری که کاربران به غذاهای خودشان داده اند را پاک کنید.
- کاربری که بیشترین اعضا را به سایت دعوت کرده را برگردانید.
- غذاهایی که هیچ مواد اولیه ای برای آن‌ها ثبت نشده است را پاک کنید.
- محتویات جدول غذاها را به همراه ستون جدیدی با نام total_price برگردانید. مقدار این ستون باید برابر با جمع حاصل ضرب قیمت واحد مواد اولیه‌ی هر غذا در تعدادشان باشد. اگر ماده‌ی اولیه‌ای برای غذا ثبت نشده بود، مقدار این ستون باید صفر باشد.
- کاربری که مجموع ارزش مواد اولیه در دسترس‌اش بیشترین مقدار را دارد را برگردانید.
- شناسه‌ی غذاهایی که کاربر با نام کاربری farbod با استفاده از مواد اولیه‌ای که در اختیار دارد می‌تواند بپزد را برگردانید.
- شناسه، عنوان، میانگین امتیازات و تعداد آرای ۱۰ غذایی که بالاترین میانگین امتیاز را کسب کرده‌اند و حداقل ۵ نظر برایشان ثبت شده را برگردانید.
- قیمت ۳ ماده غذایی که کمترین قیمت را دارند را ۳ برابر کنید.
- میانگین زمان پخت غذاهایی که حداقل یک ماده‌ای با قیمت بیشتر از ۱۰۰۰ دارند را برگردانید.

۴- برای هر کدام از موارد زیر، دستورات sql مناسب بنویسید.

پاسخ‌های مربوط به این سوال را در فایلی با نام 4.sql همراه بقیه پاسخ‌های خود ارسال کنید.

پاسخ در فایل 4.sql

جدولی از دانشجویان شامل شماره دانشجویی و اسم دانشجویان داریم.

students (student_id, name)

جدول دیگری از دوست بودن دانشجویان با هم داریم (به ازای هر دو دانشجوی دوست، دو سطر در این جدول به صورت (x,y) و (y,x) داریم).

friends (student_id, friend_id)

همچنین جدول دیگری از نمرات درس پایگاه داده دانشجویان داریم.

db_grades

(student_id, grade)

- A. نام دانشجویان را مرتب شده اساس تعداد دوست‌هایشان برگردانید (نزولی)
- B. دانشجویانی که خودشان نمره مردودی کسب کرده اند و دوستی دارند که نمره قبولی گرفته را برگردانید.
- C. دانشجویانی که تمامی دوستانشان مردود شده‌اند را برگردانید.
- D. دانشجویانی که دوستی دارند که از آن‌ها نمره بیشتر و همچنین دوستی دارند که از آن‌ها نمره کمتر گرفته را برگردانید.
- E. (امتیازی) واریانس نمره دانشجویان مردود شده را برگردانید.

خسته نباشید

```

--CREATE TABLES
create table user
(
    name          varchar(255) primary key,
    cellphone     varchar(20) unique,
    referred_by   varchar(255) null,
    FOREIGN KEY (referred_by) REFERENCES user (name)
);

create table foods
(
    id           int primary key auto_increment,
    name         varchar(255),
    author       varchar(255),
    recipe_text  text null,
    time        int not null,
    FOREIGN KEY (author) REFERENCES user (name)
);

create table ingredients
(
    id           int primary key auto_increment,
    name         varchar(255),
    price        int not null,
    constraint valid_price check ( price % 500 = 0 )
);

create table food_ingredients
(
    food_id      int,
    ingredient_id int,
    amount       int not null,
    foreign key (food_id) references foods (id),
    foreign key (ingredient_id) references ingredients (id)
);

create table user_ingredients
(
    user_name     varchar(255),
    ingredient_id int,
    amount        int not null,
    foreign key (user_name) references user (name),
    foreign key (ingredient_id) references ingredients (id)
);

create table comments
(
    id           int primary key auto_increment,
    user_name     varchar(255),
    food_id      int,
    rate         int,
    comment       text null,
    constraint limit_rate check ( rate >= 1 and rate <= 5 ),
    constraint prevent_comment check ( comment not like '%bimaze%' ),
    FOREIGN KEY (user_name) REFERENCES user (name),
    FOREIGN KEY (food_id) REFERENCES foods (id),
    UNIQUE (user_name, food_id)
);

--INSERT DATA

--users
insert into user values ('farbod', '0912121212', null);
insert into user values ('amirali', '0919314132', null);
insert into user values ('aylin', '0913141212', 'amirali');
insert into user values ('nastaran', '0912894322', 'farbod');

--foods
insert into foods (name, author, recipe, time)
VALUES ('qorme sabzi', 'nastaran', 'this is qorme sabazi recipe', 300);

insert into foods (name, author, recipe, time)

```

```

VALUES ('pizza', 'farbod', 'this is pizza recipe', 45);

insert into foods (name, author, recipe, time)
VALUES ('nimroo', 'aylin', 'this is nimroo recipe', 5);

insert into foods (name, author, recipe, time)
VALUES ('adas polo', 'farbod', 'this is adas polo recipe', 180);

--ingredients
insert into ingredients (name, price)
VALUES ('sabzi', 500);

insert into ingredients (name, price)
VALUES ('adas', 2000);

insert into ingredients (name, price)
VALUES ('polo', 1000);

insert into ingredients (name, price)
VALUES ('panir', 3000);

insert into ingredients (name, price)
VALUES ('sabzi', 500);

insert into ingredients (name, price)
VALUES ('egg', 1500);

--food_ingredients
insert into food_ingredients values (1, 1, 2);
insert into food_ingredients values (4, 6, 1);
insert into food_ingredients values (2, 3, 10);
insert into food_ingredients values (3, 4, 6);

--user_ingredients
insert into user_ingredients values ('farbod', 6, 20);
insert into user_ingredients values ('farbod', 4, 200);
insert into user_ingredients values ('aylin', 1, 100);
insert into user_ingredients values ('aylin', 2, 2);

--comments
insert into comments (user_name, food_id, rate, comment)
values ('farbod', 1, 5, 'awli bood');

insert into comments (user_name, food_id, rate, comment)
values ('aylin', 1, 1, 'eftezah bood');

insert into comments (user_name, food_id, rate, comment)
values ('farbod', 2, 3, 'average');

insert into comments (user_name, food_id, rate, comment)
values ('amirali', 1, 5, 'aaali booddd');

--REPORT QUERIES

--A
select name from foods where name like '%polo%';

--B
select avg(rate), user_name from comments group by user_name;

--C
select author, avg(c.rate) from foods join user on user.name = foods.author right join
comments c on foods.id = c.food_id group by author;

--D
--1 (works on postgres)
delete from comments where id in(select c.id from comments c join foods f on c.food_id
= f.id where c.user_name = f.author);
--2 works everywhere

```

```
create temporary table cidstodelete as select c.id from comments c join foods f on c.fo
od_id = f.id where c.user_name = f.author;
delete from comments where id in (select id from idstodelete);
```

```
--E
select * from (select referred_by, count(referred_by) c from user where referred_by is
not null group by referred_by order by c desc) c limit 1;
```

```
--F
delete from foods where id not in (select food_id from food_ingredients);
```

```
--G
--Function coalesce is used to convert null value into 0 in a select statement which h
as extra point in this question.(not necessary)
select name, coalesce(sum(cost), 0) as total_price
from (
    select fo.name, amount * price as cost
    from foods as fo
        left join food_ingredients fi on fo.id = fi.food_id
        left join ingredients i on fi.ingredient_id = i.id) f
group by name;
```

```
--H
select user_name, sum(p) as sum
from (
    select user_name, amount * price as p
    from user_ingredients
        left join ingredients i on user_ingredients.ingredient_id = i.id) a
group by user_name order by sum desc limit 1;
```

```
--I
select distinct (food_id)
from (
    select *
    from food_ingredients as fi
        join (select user_name, ingredient_id as ii, amount as am
            from user_ingredients
            where user_name = 'farbod') ui
        on fi.ingredient_id = ui.ii) c
where amount <= am
and food_id not in (
    select food_id
    from food_ingredients
    where ingredient_id not in (select user_ingredients.ingredient_id from user_ingred
ients where user_name = 'farbod')
);
```

```
--J
select food_id, count(rate) cn, avg(rate) av, name
from (
    select food_id, name, rate
    from comments
        join foods f on comments.food_id = f.id
    ) x
group by food_id
having cn >= 5 order by av desc limit 10;
```

```
--K
--1 (postgres)
update ingredients set price = price * 3 where id in (select id from ingredients order
by price asc limit 3);
--2 (everywhere)
create temporary table lowest as select * from ingredients order by price asc limit 3;
update ingredients set price = price * 3 where id in (select id from lowest);
```

```
--L
select avg(time)
from foods
```

```
where id in (  
  select distinct (food_id)  
  from ingredients  
    join food_ingredients fi on ingredients.id = fi.ingredient_id  
  where price > 1000);
```

```
--A
select s.name, count(*) cnt
from friends
      join students s on friends.student_id = s.id
group by student_id
order by cnt desc;

--B
select distinct (dg1.student_id)
from db_grades dg1,
     friends,
     db_grades dg2
where dg1.student_id = friends.student_id
     and dg2.student_id = friends.friend_id
     and dg1.grade < 10
     and dg2.grade > 10;


--create view for next 2 questions
create view friends_grades as
select dg1.student_id as studnet_id1, dg1.grade as grade1, dg2.student_id as studnet_i
d2, dg2.grade as grade2
from db_grades dg1,
     friends,
     db_grades dg2
where dg1.student_id = friends.student_id
     and dg2.student_id = friends.friend_id;

--C
select distinct(studnet_id1)
from friends_grades
where grade2 < 10
     and studnet_id1 not in (select studnet_id1 from friends_grades where grade2 >= 10);

--D
select distinct(studnet_id1)
from friends_grades
where grade2 > grade1
     and studnet_id1 in (select studnet_id1 from friends_grades where grade2 < grade1);

--E
select var_pop(grade)
from db_grades
where grade < 10;
```


بسمه تعالی

<p>تمرین دوم درس اصول طراحی پایگاه داده</p> <p>دکتر ممتازی</p> <p>ترم پاییز ۱۴۰۰ - دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر</p> <p>آخرین زمان تحویل: دوشنبه، ۱ آذر ماه ۱۴۰۰</p>	
--	---

لطفاً قبل از شروع به حل تمرین به نکات زیر توجه فرمایید:

۱. در صورت وجود هرگونه سوال یا ابهام می‌توانید با تدریس‌یار از طریق ایمیل ali_faraji@aut.ac.ir و یا آی‌دی تلگرامی [@ali_fa98](https://t.me/ali_fa98) در ارتباط باشید. (لطفاً در عنوان ایمیل کلمه «دیتابیس» را بنویسید).
۲. تمامی فایل‌ها را در قالب یک فایل فشرده به صورت **HW2_stdNo.zip** ارسال فرمایید. مانند **HW2_9831001.zip**
۳. قبل از شروع به طراحی ابتدا یکبار از اول تا انتهای فایل تمرین را بخوانید.

معرفی

ما در این تمرین می‌خواهیم مدل فرضی و ساده شده‌ای از سیستم واکسیناسیون یک کشور را داشته باشیم. مثال‌ها سعی شده است که نزدیک واقعیت در نظر گرفته شود ولی در حالت کلی این یک سیستم فرضی است.

شرح سامانه

در این سامانه اطلاعات افراد وجود دارد که شامل نام، نام خانوادگی، کد ملی و تاریخ تولد، جنسیت آن‌ها است. همچنین برای هر فرد به صورت بله/خیر مشخص می‌شود که این فرد بیماری خاص دارد یا نه. می‌دانیم که ویال‌های واکسن در «مراکز بهداشت» نگهداری می‌شوند که اسم، آدرس و کد مرکز بهداشت را داریم. ویال‌های واکسن در حالت کلی برای یک برند خاص هستند و هر ویال ممکن است در هر کشوری ساخته شده باشد. دقت کنید که هر ویال واکسن ممکن است چندین دوز در خود داشته باشد و مثلاً یک ویال ۵ دوز واکسن در خود دارد یا هر عدد دیگر که کارخانه تولید کننده بر اساس برند مشخص می‌کند. نکات مهم برای هر ویال واکسن محل تولید، تاریخ تولید، شماره سریال و تعداد دوز می‌باشد که در یک «مرکز بهداشت» نگهداری می‌شود.

هر برند واکسن برای یک شرکت/مؤسسه به خصوص است که نام و ملیت و نوع آن مؤسسه (دولتی/نیمه خصوصی/خصوصی) مشخص می‌باشد و ممکن است یک مؤسسه، دو برند واکسن داشته باشد. همچنین هر برند یک تعداد دوز واکسینه شدن دارد که هر شخص برای واکسینه شدن باید به آن تعداد دوز تزریق کند تا به آن فرد «شخص واکسینه» بگوییم، لازم به ذکر است که هر برند یک نام یکتا هم دارد.

تزریق واکسن هم به این صورت است که در یک زمان مشخص یک واکسن در یک «مرکز واکسیناسیون» توسط یک واکسیناتور به فرد تزریق شده است و ما می‌خواهیم این اطلاعات را به عنوان تاریخچه نگهداری کنیم. بعد از تزریق هم هر شخص می‌تواند یک امتیاز به «مرکز واکسیناسیون» که واکسنش را زده است بدهد به این صورت که امتیاز عددی از ۱ تا ۵ است. مراکز واکسیناسیون مانند مراکز بهداشت هستند ولی کد مرکز ندارند و می‌توانند هرجایی باشند مانند مسجد، مدرسه و ... (هر شخص برای یک مرکز، یک امتیاز می‌تواند بدهد) (توجه کنید که هر مرکز بهداشتی یک مرکز واکسیناسیون است ولی عکس این جمله صادق نیست)

مبحث دیگر واکسیناتورها هستند که دو دسته کلی دارند: پرستارها و پزشکان.

پرستارها علاوه بر اطلاعات به مانند یک فرد معمولی که قبل گفته شده است، یک سطح مدرک و یک کد پرستاری دارد. پزشکان هم به مانند افراد جامعه اطلاعات کلی دارند و علاوه بر آن یک شماره نظام پزشکی هم دارند که باید نگهداری شود.


مطلوبات تمرین

۱. لیست موجودیت‌ها (Entity) را مشخص کنید.
 - * نوع موجودیت از نظر weak entity و strong entity را مشخص کنید.
۲. برای هر entity باید attribute‌ها مشخص شوند.
 - * Primary Key‌ها را مشخص کنید.
۳. روابط بین entity‌ها را مشخص کنید.
 - * در صورت لزوم attribute این روابط را هم مشخص کنید.
۴. ER Diagram نهایی را رسم کنید. توجه کنید طبق یک notation استاندارد تمام cardinality‌ها و entity‌ها را مشخص کنید.
۵. دستورات لازم برای ایجاد چنین پایگاه داده‌ای را بنویسید.
۶. اکسپورت پایگاه داده خودتان را در قالب یک فایل sql ارسال کنید.

مواردی که باید در قالب یک فایل zip ارسال کنید

1. فایل PDF مربوط به سوال‌های ۱ تا ۴
 2. فایل‌های SQL مربوط به سوال‌های ۵ و ۶
- نکته: پروژه پایانی شما مربوط به همین سامانه خواهد بود پس سعی کنید اولاً حتماً این تمرین را انجام دهید و دوماً تا حد امکان سعی کنید به صورت استاندارد و اصولی سیستم خود را طراحی کنید.
- این خط جز این تکلیف نیست ولی می‌تواند مثالی از سؤال پروژه باشد: کوثری درصد «افراد واکسینه شده» را برای هر برند بنویسید.

با تشکر - تیم تدریس‌یاری

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

لیست موجودیت‌ها:

موجودیت	Strong/Weak	Attributes
شرکت (یا موسسه)	Strong	<ul style="list-style-type: none"> نام ملیت نوع
برند	Strong	<ul style="list-style-type: none"> نام تعداد دوز
واکسن	Strong	<ul style="list-style-type: none"> شماره سریال تاریخ تولید مکان تولید تعداد دوز
مرکز واکسیناسیون	Strong	<ul style="list-style-type: none"> نام آدرس <ul style="list-style-type: none"> خیابان شهر استان کد پستی
مرکز بهداشت	Strong	<ul style="list-style-type: none"> کد مرکز
شخص	Strong	<ul style="list-style-type: none"> نام نام خانوادگی کد ملی جنسیت تاریخ تولد بیماری خاص

پزشک	Strong	• کد نظام پزشکی
پرستار	Strong	• کد پرستاری • سطح

تکمیلی:

تزریق*	-	• تاریخ و زمان
امتیاز**	Weak	• مقدار

* تزریق یک موجودیت مجازی است که از تجزیه رابطه‌ی «تزریق» که یک رابطه‌ی چهار طرفه است به دست آمده است.

** اگر امتیاز با به عنوان یک موجودیت در نظر گرفته‌اید پس Weak Entity می‌باشد. ما در این تمرین به صورت یک رابطه در نظر گرفته‌ایم.

لیست روابط:

belongs: رابطه‌ی بین شرکت و برندهایش را مشخص می‌کند.

vac_brand: رابطه‌ی بین واکسن و برند آن را مشخص می‌کند.

Storage: رابطه‌ی بین مرکز بهداشت و واکسن‌های ذخیره شده در آن‌جا

Score: رابطه‌ی بین مرکز واکسیناسیون و یک شخص است که ویژگی به نام value دارد و امتیاز را مشخص می‌کند.

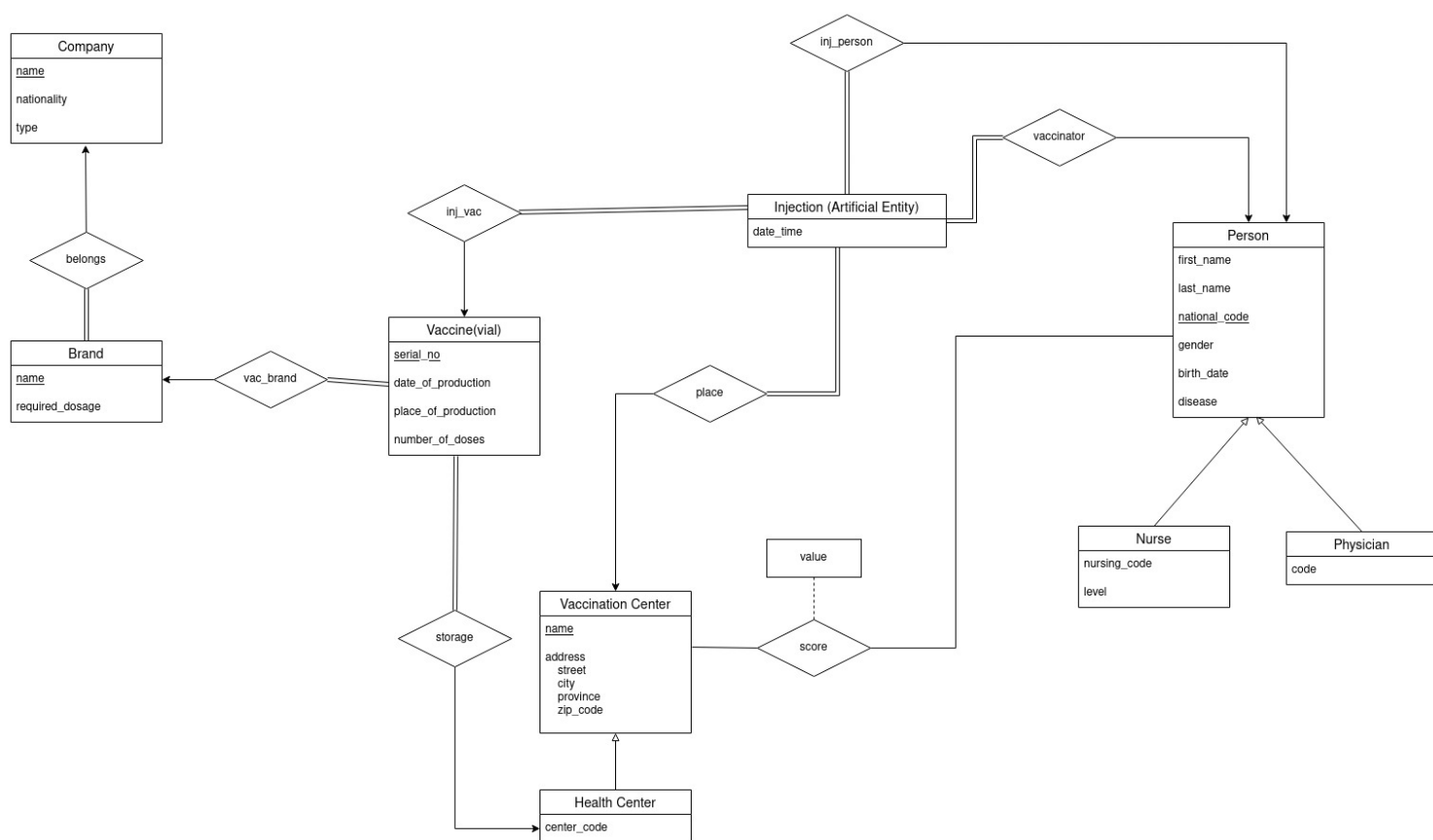
inj_vac: رابطه‌ی بین تزریق و واکسن تزریق شده را مشخص می‌کند.

inj_person: رابطه‌ی بین تزریق و شخص را مشخص می‌کند.


Vaccinator: رابطه‌ی بین تزریق و شخص واکسیناتور را مشخص می‌کند.

Place: رابطه‌ی بین تزریق و مرکز واکسیناسیون را مشخص می‌کند.

دیاگرام:



بسمه تعالی

<p>تمرین سوم درس اصول و طراحی پایگاه داده</p> <p>دکتر ممتازی</p> <p>ترم پاییز ۱۴۰۰ - دانشکده کامپیوتر، دانشگاه صنعتی امیرکبیر</p> <p>زمان تحویل: ۴ دی ۱۴۰۰</p>	
--	---

لطفاً 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 esrafiliyanm@gmail.com در ارتباط باشید. لطفاً در عنوان ایمیل کلمه «دیتابیس» را ذکر کنید.
- ۲- مهلت تحویل تمرین تا تاریخ ۴ دی ماه می‌باشد. **بعد از این تاریخ تمرین به هیچ عنوان تحویل گرفته نمی‌شود.**
- ۳- تمامی فایل‌های خواسته شده را در قالب یک فایل pdf با نام HW3-StudentNumber در سامانه کورسز بارگذاری کنید. به عنوان مثال: HW3-9731007.

۱- در رابطه

$R(A, B, C)$

سه چندتایی $(1,2,3)$, $(2,2,3)$, $(3,4,5)$ وجود دارند. کدامیک از وابستگی‌های تابعی زیر بر روی R برقرار هستند؟ توضیح دهید.

$$A \longrightarrow BC$$

$$B \longrightarrow A$$

$$C \longrightarrow AB$$

۲- دو ادعای زیر درباره وابستگی‌های تابعی را در نظر بگیرید:

a) If $\alpha \longrightarrow \beta\gamma$ holds then $\alpha \longrightarrow \beta$ holds and $\alpha \longrightarrow \gamma$ holds

b) If $\alpha\beta \longrightarrow \gamma$ holds then $\alpha \longrightarrow \gamma$ holds and $\beta \longrightarrow \gamma$ holds

با استفاده از Armstrong's Axioms درستی هرکدام را اثبات کنید یا برای نادرست بودن آن مثال نقض بیاورید.

۳- رابطه و وابستگی‌های تابعی زیر را در نظر بگیرید:

$R(A, B, C, D, E)$

$$A \longrightarrow B$$

$$AC \longrightarrow BD$$

$$AB \longrightarrow C$$

$$BCD \longrightarrow A$$

الف) A^+ را بدست آورید.

ب) Extraneous Attribute ها را بیابید و ادعای خود را بیازمایید.

ج) کلیدهای کاندید رابطه R را بدست آورید.

۴- رابطه و وابستگی‌های تابعی زیر را در نظر بگیرید:

$R(A, B, C, D, E, F, G, H)$

$A \rightarrow E$

$BE \rightarrow D$

$AD \rightarrow BE$

$BDH \rightarrow E$

$AC \rightarrow E$

$F \rightarrow A$

$E \rightarrow B$

$D \rightarrow H$

$BG \rightarrow F$

$CD \rightarrow A$

الف) وابستگی‌های تابعی بالا را به فرم Canonical Cover بنویسید و کلیدهای کاندید را مشخص کنید.

ب) رابطه بالا را به فرم 3NF تجزیه کنید.

ج) آیا پاسخ شما در قسمت (ب) به فرم BCNF نیز هست؟ توضیح دهید.

د) نشان دهید اگر رابطه R را به دو رابطه زیر تجزیه کنیم، این تجزیه lossless است.

$R_1(A, B, D, E)$

$R_2(A, C, F, G, H)$

۵- رابطه و وابستگی‌های تابعی زیر را در نظر بگیرید:

$R(A, B, C, D, E, F, G)$

$CD \rightarrow B$

$FD \rightarrow C$

$AF \rightarrow D$

$BG \rightarrow E$

$ACE \rightarrow FG$

الف) برای رابطه بالا تجزیه‌ای به فرم BCNF ارائه دهید. سپس با ذکر دلیل بیان کنید آیا تجزیه شما Dependency Preserving هست یا خیر.

ب) تجزیه زیر را در نظر بگیرید. با ذکر دلیل نشان دهید آیا روابط این تجزیه به فرم 3NF هستند یا خیر.

$R_1(B, C, D) \quad \{CD \rightarrow B\}$

$R_2(C, D, F) \quad \{DF \rightarrow C\}$

$R_3(A, B, D, E, F, G) \quad \{AF \rightarrow D, BG \rightarrow E\}$

$R_4(A, C, E, F, G) \quad \{ACE \rightarrow FG\}$

۶- رابطه زیر را که شامل اطلاعات افرادی است که واکسن تزریق کرده‌اند، در نظر بگیرید:

$Vaccinated_info(full_name, ID, vaccine_brand, diseases, emergency_phone_number)$

full_name: نام کامل هر فرد (یکتاست)

ID: کد ملی هر فرد (یکتاست)

vaccine_brand: نام برند واکسنی که تزریق شده است (هر فرد می‌تواند تنها از یک برند واکسن تزریق کند)

disease: مجموعه بیماری‌های خاص هر فرد (هر فرد می‌تواند چند بیماری خاص داشته باشد)


emergency_phone_number: مجموعه شماره‌های تماس اضطراری برای هر فرد (هر فرد می‌تواند چند شماره تماس اضطراری داشته باشد)

الف) وابستگی‌های رابطه بالا را بنویسید.

ب) با ذکر مثال مشکلات این طراحی را توضیح دهید.

ج) تجزیه‌ای برای جلوگیری از مشکلات ذکر شده در بخش (ب) ارائه دهید.

بسمه تعالی

<p>پاسخنامه تمرین سوم درس اصول و طراحی پایگاه داده</p> <p>دکتر ممتازی</p> <p>ترم پاییز ۱۴۰۰ - دانشکده کامپیوتر، دانشگاه صنعتی امیرکبیر</p>	
--	---

۱- در رابطه

$R(A, B, C)$

سه چندتایی $(1,2,3)$, $(2,2,3)$, $(3,4,5)$ وجود دارند. کدامیک از وابستگی‌های تابعی زیر بر روی R برقرار هستند؟ توضیح دهید.

$$A \longrightarrow BC$$

$$B \longrightarrow A$$

$$C \longrightarrow AB$$

پاسخ:

می‌دانیم وابستگی تابعی $\alpha \longrightarrow \beta$ بر روی رابطه R برقرار است اگر و تنها اگر دو چندتایی t_1 , t_2 که در α مشترک‌اند شرط زیر را داشته باشند:

$$t_1[\alpha] = t_2[\alpha] \implies t_1[\beta] = t_2[\beta]$$

با توجه به تعریف بیان شده و چندتایی‌های موجود هر یک از وابستگی‌ها را می‌آزماییم:

در وابستگی اول شرط اولیه اشتراک بین چندتایی‌ها وجود ندارد. پس می‌توان گفت این وابستگی برقرار است.

در وابستگی دوم در چندتایی‌های $(1,2,3)$, $(2,2,3)$ برای B مقدار مشترک 2 را داریم اما برای A دو مقدار متفاوت 1 و 2 پس این وابستگی برقرار نیست.

در وابستگی سوم در چندتایی‌های $(1,2,3)$, $(2,2,3)$ برای C مقدار مشترک 3 را داریم اما برای AB دو مقدار متفاوت $(1,2)$ و $(2,2)$ پس این وابستگی برقرار نیست.

۲- دو ادعای زیر درباره وابستگی‌های تابعی را در نظر بگیرید:

- a) If $\alpha \longrightarrow \beta\gamma$ holds then $\alpha \longrightarrow \beta$ holds and $\alpha \longrightarrow \gamma$ holds
b) If $\alpha\beta \longrightarrow \gamma$ holds then $\alpha \longrightarrow \gamma$ holds and $\beta \longrightarrow \gamma$ holds

با استفاده از Armstrong's Axioms درستی هرکدام را اثبات کنید یا برای نادرست بودن آن مثال نقض بیاورید.

پاسخ:

a)

$\alpha \longrightarrow \beta\gamma$ given

$\beta\gamma \longrightarrow \beta$ reflexivity rule

$\alpha \longrightarrow \beta$ transitivity rule

$\beta\gamma \longrightarrow \gamma$ reflexive rule

$\alpha \longrightarrow \gamma$ transitive rule

b)

این ادعا غلط است. به عنوان مثال نقض رابطه زیر را در نظر بگیرید:

$R(A, B, C)$

دو چندتایی $(1,3,3)$, $(1,2,4)$ را در نظر بگیرید. همانطور که در سوال ۱ بیان شد در اینجا وابستگی زیر برقرار است:

$AB \longrightarrow C$

اما رابطه $A \longrightarrow C$ برقرار نیست.

۳- رابطه و وابستگی‌های تابعی زیر را در نظر بگیرید:

$R(A, B, C, D, E)$

$A \longrightarrow B$

$AC \longrightarrow BD$

$AB \longrightarrow C$

$BCD \longrightarrow A$

الف) A^+ را بدست آورید.

ب) Extraneous Attribute ها را بیابید و ادعای خود را بیازمایید.

ج) کلیدهای کاندید رابطه R را بدست آورید.

پاسخ:

الف)

A^+

1- result = A

2- result = AB (1st functional dependency)

3- result = ABC (3rd functional dependency)

4- result = ABCD (2nd functional dependency) $\Rightarrow A^+ = ABCD$

ب) با استفاده از موارد زیر ادعای خود را در هر مورد می‌آزماییم:

$\alpha \longrightarrow \beta$

■ To test if attribute $A \in \alpha$ is extraneous in α

1. compute $(\{\alpha\} - A)^+$ using the dependencies in F
2. check that $(\{\alpha\} - A)^+$ contains β ; if it does, A is extraneous in α

■ To test if attribute $A \in \beta$ is extraneous in β

1. compute α^+ using only the dependencies in $F' = (F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - A)\}$,
2. check that α^+ contains A ; if it does, A is extraneous in β

$$AC \longrightarrow BD$$

ابتدا اضافه بودن B را می‌آزماییم:

F':

$$A \longrightarrow B$$

$$AC \longrightarrow D$$

$$AB \longrightarrow C$$

$$BCD \longrightarrow A$$

$$(AC)^+ = ACBD$$

با توجه به اینکه $(AC)^+$ شامل B نیز هست پس ادعای ما درست است.

سپس اضافه بودن C را با استفاده از وابستگی‌های اصلی می‌آزماییم:

$$A^+ = ABCD$$

با توجه به اینکه A^+ شامل C نیز هست پس ادعای ما درست است.

پس این وابستگی به وابستگی زیر تبدیل می‌شود:

$$A \longrightarrow D$$

حال برای وابستگی زیر اضافه بودن B را می‌آزماییم:

$$AB \longrightarrow C$$

$$(A)^+ = ABCD$$

با توجه به اینکه A^+ شامل B نیز هست پس ادعای ما درست است.

پس این وابستگی به وابستگی زیر تبدیل می‌شود:

$$A \longrightarrow C$$

ج)

با توجه به بخش الف می‌توان A را به عنوان کلید کاندید محسوب کرد. همچنین:

$$(BCD)^+ = ABCD$$

و هیچ زیر مجموعه‌ای از BCD موجود نیست که بتوانیم از آن به R برسیم. پس BCD نیز یک کلید کاندید است.

۴- رابطه و وابستگی‌های تابعی زیر را در نظر بگیرید:

$R(A, B, C, D, E, F, G, H)$

$$A \rightarrow E$$

$$BE \rightarrow D$$

$$AD \rightarrow BE$$

$$BDH \rightarrow E$$

$$AC \rightarrow E$$

$$F \rightarrow A$$

$$E \rightarrow B$$

$$D \rightarrow H$$

$$BG \rightarrow F$$

$$CD \rightarrow A$$

الف) وابستگی‌های تابعی بالا را به فرم Canonical Cover بنویسید و کلیدهای کاندید را مشخص کنید.

ب) رابطه بالا را به فرم 3NF تجزیه کنید.

ج) آیا پاسخ شما در قسمت (ب) به فرم BCNF نیز هست؟ توضیح دهید.

د) نشان دهید اگر رابطه R را به دو رابطه زیر تجزیه کنیم، این تجزیه lossless است.

$R_1(A, B, D, E)$

$R_2(A, C, F, G, H)$

پاسخ:

الف) B در سمت چپ وابستگی $BE \rightarrow D$ اضافه است زیرا در $E \rightarrow B$ می‌توان از صفت E به ویژگی B رسید.

$$F = A \rightarrow E, E \rightarrow D, AD \rightarrow BE, BDH \rightarrow E, AC \rightarrow E, F \rightarrow A, E \rightarrow B, D \rightarrow H, BG \rightarrow F, CD \rightarrow A$$

D در سمت چپ وابستگی $AD \rightarrow BE$ اضافه است زیرا در $A \rightarrow E, E \rightarrow B$ می‌توان از A به EB رسید. E نیز در سمت راست اضافه است زیرا در $A \rightarrow E$ می‌توان از A به E رسید.

$$F = A \rightarrow E, E \rightarrow D, A \rightarrow B, BDH \rightarrow E, AC \rightarrow E, F \rightarrow A, E \rightarrow B, D \rightarrow H, BG \rightarrow F, CD \rightarrow A$$

H در سمت چپ $BDH \rightarrow E$ اضافه است زیرا در $D \rightarrow H$ می‌توان از D به H رسید:

$$F = A \rightarrow E, E \rightarrow D, A \rightarrow B, BD \rightarrow E, AC \rightarrow E, F \rightarrow A, E \rightarrow B, D \rightarrow H, BG \rightarrow F, CD \rightarrow A$$

C در سمت چپ $E \rightarrow AC$ اضافه است زیرا در $A \rightarrow E$ از A می‌توان به E رسید پس این وابستگی اضافه است:

$$F = A \rightarrow E, E \rightarrow D, A \rightarrow B, BD \rightarrow E, F \rightarrow A, E \rightarrow B, D \rightarrow H, BG \rightarrow F, CD \rightarrow A$$

وابستگی $A \rightarrow B$ نیز اضافه است زیرا می‌توان از $A \rightarrow E, E \rightarrow B$ آن را نتیجه گرفت:

$$F = A \rightarrow E, E \rightarrow BD, BD \rightarrow E, F \rightarrow A, D \rightarrow H, BG \rightarrow F, CD \rightarrow A$$

مجموعه وابستگی بالا فرم canonical cover است.

ب) یک تجزیه ممکن:

$$R1(A, E) \quad A \rightarrow E, Ck = A$$

$$R2(B, D, E) \quad E \rightarrow BD, BD \rightarrow E, Ck = E, BD$$

$$R3(A, F) \quad F \rightarrow A, Ck = F$$

$$R4(D, H) \quad D \rightarrow H, Ck = D$$

$$R5(B, F, G) \quad BG \rightarrow F, Ck = BG$$

$$R6(A, C, D) \quad CD \rightarrow A, Ck = CD$$

کلیدهای کاندید رابطه R به شرح زیر است:

$$ACG, BCG, CDG, CEG, CFG$$

هیچ کدام از روابط بالا شامل یکی از کلیدهای کاندید رابطه R نیستند پس یکی از کلیدها را انتخاب می‌کنیم و یک رابطه که آن را شامل شود اضافه می‌کنیم:

$$R7(B, C, G) \quad \text{no functional dependency} \quad Ck = BCG$$

ج) تجزیه بالا به فرم BCNF نیز هست. زیرا همانطور که مشخص شده در سمت چپ وابستگی‌های هر رابطه کلید کاندید آن رابطه قرار دارد.

(د) با توجه به تعریف زیر داریم:

A decomposition of R into R_1 and R_2 is lossless join if at least one of the following dependencies is in F^+ :

- $R_1 \cap R_2 \rightarrow R_1$
- $R_1 \cap R_2 \rightarrow R_2$

$$R1 \cap R2 = A$$

$$A^+ = (AE)^+ = (AEB)^+ = (AEBD)$$

همانطور که مشاهده می‌کنیم توانستیم با توجه به وابستگی‌های داده شده از A به AEBD برسیم که شامل تمام ویژگی‌های R1 است. پس می‌توان نتیجه گرفت این تجزیه lossless است.

۵- رابطه و وابستگی‌های تابعی زیر را در نظر بگیرید:

$R(A, B, C, D, E, F, G)$

$CD \rightarrow B$

$FD \rightarrow C$

$AF \rightarrow D$

$BG \rightarrow E$

$ACE \rightarrow FG$

الف) برای رابطه بالا تجزیه‌ای به فرم BCNF ارائه دهید. سپس با ذکر دلیل بیان کنید آیا تجزیه شما Dependency Preserving هست یا خیر.

ب) تجزیه زیر را در نظر بگیرید. با ذکر دلیل نشان دهید آیا روابط این تجزیه به فرم 3NF هستند یا خیر.

$R1(B, C, D) \quad \{CD \rightarrow B\}$

$R2(C, D, F) \quad \{DF \rightarrow C\}$

$R3(A, B, D, E, F, G) \quad \{AF \rightarrow D, BG \rightarrow E\}$

$R4(A, C, E, F, G) \quad \{ACE \rightarrow FG\}$

پاسخ:

الف) ابتدا کلیدهای کاندید را بدست می‌آوریم. کلیدهای این رابطه عبارتند از:

ACE, AEF, AFG, ABCG, ACDG

همانطور که مشخص است تنها وابستگی آخر BCNF را نقض نمی‌کند.

$$CD \longrightarrow B$$

R1(B, C, D), Ck = CD

R2(A, C, D, E, F, G), Ck = ACE, AEF, AFG, ACDG

$$CD \longrightarrow B, BG \longrightarrow E \implies CDG \longrightarrow E$$

$$FD \longrightarrow C$$

R21(C, D, F), Ck = FD

R22(A, D, E, F, G), Ck = AEF, AFG

$$FD \longrightarrow C, ACE \longrightarrow FG \implies AFDE \longrightarrow FG, AF \longrightarrow D \implies AFE \longrightarrow G$$

$$FD \longrightarrow C, CDG \longrightarrow E \implies FDG \longrightarrow E$$

$$AF \longrightarrow D$$

R221(A, D, F), Ck = AF

R222(A, E, F, G), Ck = AEF, AFG

$$AF \longrightarrow D, FDG \longrightarrow E \implies AFG \longrightarrow E$$

در نتیجه تجزیه بالا به فرم BCNF است.

ب) در همه روابط به جز رابطه R3 در سمت چپ وابستگی‌ها کلید کاندید آن رابطه قرار دارد.

در رابطه R3 کلید کاندید AFBG است و همانطور که مشخص است ویژگی سمت راست وابستگی‌ها نیز در کلید کاندید وجود ندارد. پس این رابطه به فرم 3NF نیست.

پاسخنامه تمرین سوم درس اصول و طراحی پایگاه داده

۶- رابطه زیر را که شامل اطلاعات افرادی است که واکسن تزریق کرده‌اند، در نظر بگیرید:

Vaccinated_info(full_name, ID, vaccine_brand, diseases, emergency_phone_number)

full_name: نام کامل هر فرد (یکتااست)

ID: کد ملی هر فرد (یکتااست)

vaccine_brand: نام برند واکسنی که تزریق شده است (هر فرد می‌تواند تنها از یک برند واکسن تزریق کند)

disease: مجموعه بیماری‌های خاص هر فرد (هر فرد می‌تواند چند بیماری خاص داشته باشد)

emergency_phone_number: مجموعه شماره‌های تماس اضطراری برای هر فرد (هر فرد می‌تواند چند شماره تماس اضطراری داشته باشد)

الف) وابستگی‌های رابطه بالا را بنویسید.

ب) با ذکر مثال مشکلات این طراحی را توضیح دهید.

ج) تجزیه‌ای برای جلوگیری از مشکلات ذکر شده در بخش (ب) ارائه دهید.

پاسخ:

الف) وابستگی‌ها را می‌توان به شکل زیر نشان داد:

ID → full_name, vaccine_brand

ID →→ disease

ID →→ emergency_phone_number

ب)

ID	full_name	vaccine_brand	disease	emergency_phone_number
1	Jane Doe	Sinopharm	Asthma	09121111111
1	Jane Doe	Sinopharm	Asthma	09131111111
1	Jane Doe	Sinopharm	Diabetes	09121111111
1	Jane Doe	Sinopharm	Diabetes	09131111111

پاسخنامه تمرین سوم درس اصول و طراحی پایگاه داده

همانطور که در جدول بالا میبینیم در این طراحی فیلدهای multivalued که ارتباطی باهم ندارند باعث ایجاد سطرهای اضافه می شود. برای جلوگیری از این مشکل می توان با استفاده از 4NF جدول را تجزیه کنیم.

ج)

R1(full_name, ID, vaccine_brand)

R2(ID, disease)

R3(ID, emergency_phone_number)



پروژه درس اصول و طراحی پایگاه داده
دکتر ممتازی
ترم پائیز ۱۴۰۰ - دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر
زمان تحویل: ۸ صبح ۱۴۰۰/۱۱/۰۷

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

۱. در صورت وجود هرگونه سوال یا ابهام میتوانید با تدریسار از طریق آیدی تلگرام @mohmdRad یا ایمیل mhmohammadirad@gmail.com در ارتباط باشید.
۲. مهلت تحویل پروژه تا ۸ صبح ۱۴۰۰/۱۱/۰۷ می باشد و این تاریخ قابل تمدید نیست.
۳. همه کدهای SQL خود را در یک فایل sql قرار دهید و این فایل را به همراه فایل های برنامه رابط کاربری به همراه تمامی کتابخانه های استفاده شده (به صورتی که روی دستگاه های دیگر قابل اجرا باشد)، در قالب یک فایل zip با فرمت StudentNumber.zip در سایت بارگذاری نمایید.

در این پروژه می‌بایست پایگاه داده‌ای مشابه سیستم واکسیناسیون که در تمرین دوم طراحی کردید را پیاده‌سازی کنید. در این پروژه برای سادگی فقط مراکز واکسیناسیون در نظر گرفته می‌شوند و فرض می‌شود که واکسن‌ها در مراکز واکسیناسیون ذخیره و ارائه می‌شوند در نتیجه نیازی به جدول برای مراکز بهداشت نیست. همچنین فرض می‌شود که هر موسسه تولید واکسن فقط یک برند تولید می‌کند و نیازی به ایجاد جدولی برای موسسات نیست. هدف از این صورت پروژه، پیاده‌سازی مدل پایگاه داده، کار با SQL به صورت مستقیم و پیاده‌سازی روابط و مدیریت داده‌ها و کار با مدل پیاده‌سازی شده از یک رابط کاربری جداگانه می‌باشد.

توجه داشته باشید که ساختارهای لازم برای موجودیت‌ها در این پروژه ذکر شده است ولی می‌توانید بر اساس صلاح دید خود ویژگی‌هایی را به آن بیافزایید. ولی در معماری خود اصول نرمال‌سازی شماتیک پایگاه داده برای پیشگیری از افزونگی و ... را به کار ببرید.

در این پروژه باید برای تمام توانمندی‌های خواسته شده تابع یا روالی مجزا با ورودی‌های مناسب تعریف کنید که این توابع یا روال‌ها توسط رابط کاربری فراخوانی می‌شوند.

توجه داشته باشید که می‌توانید از نرم افزارهای DBMS مانند MariaDB یا MySQL برای پیاده‌سازی پایگاه داده، و هر یک از زبان‌های برنامه‌نویسی برای پیاده‌سازی رابط کاربری، استفاده کنید.

نکات کلی :

- همه‌ی موارد خواسته شده باید کاملاً در SQL پیاده‌سازی شوند و رابط کاربری فقط نتیجه گرفته شده از پایگاه داده را به صورت دلخواه شما نمایش دهد.
- برای نگهداری رشته‌های طولانی می‌توانید طول ۵۱۲ را در نظر بگیرید.
- برای نگهداری شماره تلفن، کدملی یا شماره‌های دیگر از ویژگی‌های عددی استفاده نکنید زیرا صفر ابتدای آن از دست می‌رود.
- مواردی که با قید "به صورت خودکار" توصیف شده اند، باید با trigger یا با استفاده از کلید خارجی پیاده‌سازی شوند.
- موارد امتیازی با سبز نمایش داده شده‌اند.

حساب کاربری

هر کاربر یک حساب دارد که اطلاعاتش به دو بخش اطلاعات سیستمی مانند " نام کاربری و گذرواژه " و اطلاعات شخصی مانند " نام، شماره تلفن، تاریخ تولد و ... " دسته‌بندی می‌شود.

اطلاعات سیستمی:

این بخش داده‌های زیر را در بر دارد :

- نام کاربری (کد ملی): کد ملی هر شخص یک رشته عددی ۱۰ رقمی یکتا می‌باشد.
 - نباید یک کد ملی بیش از یک بار در جدول وجود داشته باشد.
 - رشته‌های غیر عددی و رشته‌هایی با طول کمتر یا بیشتر از ۱۰ رقم نباید پذیرفته شود.
- این مورد باید به صورت خودکار بررسی شود. تشخیص این خطاها در SQL پیاده سازی شود و در صورت هرگونه مغایرتی پیغام مناسب نمایش داده شود.

- گذرواژه: گذرواژه باید حداقل ۸ کاراکتر دربردارنده حروف الفبا و عدد (هر دو) باشد.
 - هر گذرواژه باید هم شامل حروف الفبا و هم اعداد باشد و طول گذرواژه نیز کمتر از ۸ کاراکتر نباشد.
 - گذرواژه به هنگام دریافت باید در پایگاه داده به صورت درهم ریخته (Hashed) نگهداری شود و در هنگام ورود فرد به حساب خود، باید گذرواژه وارد شده ابتدا درهم ریخته شود و سپس با آنچه در پایگاه داده ذخیره شده است مقایسه شود. برای این کار می‌توانید از تابع [MD5](#) استفاده کنید.
- زمان ساخت حساب: زمان ساخت حساب هر فرد (تاریخ و ساعت) در هنگام ثبت نام باید در پایگاه داده نگهداری شود.
 - مشخص کردن این زمان باید در SQL پیاده‌سازی شود و زمان به وقت ایران مشخص و ذخیره شود.

اطلاعات شخصی:

این بخش داده‌های زیر را در بر دارد:

- نام
- نام خانوادگی
- جنسیت
- تاریخ تولد (تاریخ شمسی)
- بیماری خاص (فرد بیماری خاص دارد یا خیر)

اطلاعات کادر درمان:

برای کادر درمان (پرستاران و پزشکان) علاوه بر دو دسته اطلاعات ذکر شده، باید اطلاعات مورد نیاز دیگری نیز ذخیره شود.

برای پزشکان:

- کد نظام پزشکی: کد نظام پزشکی یک رشته‌ی عددی ۵ رقمی یکتا می‌باشد یعنی کد وارد شده باید دقیقا ۵ کاراکتر عددی باشد و یک کد نظام پزشکی بیش از یک‌بار در جدول وجود نداشته باشد.

برای پرستاران:

- سطح: سطح پرستار می‌تواند یکی از موارد «مترون»، «سوپروایزر»، «پرستار» و «بهبیار» باشد و سطوح خارج از این موارد باید به صورت خودکار بررسی و جلوگیری شود.
- کد پرستاری: کد پرستاری یک رشته‌ی عددی ۸ رقمی یکتا می‌باشد یعنی کد وارد شده باید دقیقا ۸ کاراکتر عددی باشد و یک کد پرستاری بیش از یک‌بار در جدول وجود نداشته باشد.

ثبت نام:

اشخاص برای استفاده از خدمات سامانه ابتدا باید حساب کاربری ایجاد کنند. برای این کار، کاربران معمولی باید همه‌ی اطلاعات شخصی و سیستمی خود (به جز زمان ثبت نام که به صورت خودکار اضافه می‌شود) را وارد کنند. عملیات ثبت نام باید در یک تراکنش (Transaction) انجام شود تا چنانچه مشکلی پیش آمد و ورودی‌های کاربر قابل قبول نبود عملیات روی جداول اعمال نشود و بازگردانی شود. مثلا ممکن است کد ملی کاراکترهای غیر عددی داشته باشد یا تکراری باشد و یا گذرواژه هم رقم و هم حرف را در بر نداشته باشد که باید پیام‌های خروجی متناسب با خطا نمایش داده شوند.

همانطور که گفته شد کادر درمان (پزشکان و پرستاران) غیر از این اطلاعات باید اطلاعات دیگری نیز وارد کنند. در صورت درخواست کاربر برای ثبت نام به عنوان کادر درمان باید این اطلاعات نیز دریافت و صحت‌سنجی شوند.

*** توجه شود که همان‌گونه که بالاتر گفته شد همه پیاده‌سازی‌ها و حتی پیام‌های خطا نیز باید در SQL پیاده‌سازی شود و رابط کاربری فقط خروجی دریافت شده را در صورت موفقیت‌آمیز بودن یا روبه‌رو شدن با خطا به کاربر نشان می‌دهد.

ورود به حساب:

کاربران برای ورود باید کد ملی و گذرواژه خود را در رابط کاربری وارد کنند. در SQL بررسی می‌شود که آیا درهم‌ریخته‌ی گذرواژه‌ی وارد شده با مقدار درهم‌ریخته‌ی نگه‌داری شده برای حساب کاربر برابر است یا خیر. اگر برابر نبودند پیغام خطای مناسب باز می‌گردد و اگر برابر بودند ورود کاربر با موفقیت انجام می‌شود. برای ثبت ورود کاربر می‌توانید جدولی در نظر بگیرید که در صورت وارد کردن صحیح کد ملی و گذرواژه، اطلاعات مورد نیاز از کاربر وارد شده و زمان ورود را ذخیره کند. به این صورت برای انجام بقیه‌ی عملیات‌ها نیازی به دوباره فرستادن اطلاعات ورود نیست و آخرین کاربر وارد شده به سیستم به عنوان درخواست دهنده‌ی آن عملیات‌ها در نظر گرفته می‌شود؛ پس وارد کردن نام کاربری و گذرواژه تنها باید یکبار و در هنگام ورود کاربر به حساب خود انجام شود.

می‌توانید هنگام ورود موفق کاربر، یک تگ (مثلاً یک رشته کاراکتری تصادفی) در جدول ثبت ورودها اضافه کنید و از طرف دیگر این تگ را برای کاربر نیز ارسال کنید. کاربران برای دریافت هر خدمتی باید این تگ را نیز در درخواست‌شان ارسال کنند و با مقایسه کردن تگ دریافت شده از کاربران مشخص شود که این عملیات مربوط به کدام کاربر بوده است. این تگ با عملیات ورود کاربر به حساب کاربری‌اش به وی اختصاص داده می‌شود و با عملیات خروج از حساب منقضی خواهد شد.

عملیات‌های کاربران

کاربران در سه سطح کاربران معمولی، پرستاران و پزشکان هستند. نوع کاربری، سطح دسترسی کاربر را مشخص می‌کند.

عملیات‌های پزشکان:

پزشکان باید بتوانند عملیات‌های ایجاد برند جدید، ایجاد مرکز درمانی جدید و حذف حساب‌های کاربری را انجام دهند.

ایجاد برند جدید:

برای ایجاد برند جدید باید موارد زیر وارد شود.

- نام برند: نام برند باید نامی یکتا باشد.
- دوز تزریق: تعداد دوزی که باید هر فرد تزریق کند تا یک فرد واکسینه باشد.
- تعداد روزهای مناسب برای فاصله بین دوزها

- هنگامی که پزشکی یک برند جدید ایجاد کرد باید شماره نظام پزشکی او نیز به عنوان ایجاد کننده برند ذخیره شود. این شماره نظام پزشکی نباید توسط خود پزشک وارد شود و باید با توجه به فرد وارد شده به صورت خودکار مشخص شود.

ایجاد مرکز درمانی جدید:

برای ایجاد مرکز درمانی جدید باید موارد زیر وارد شود.

- نام مرکز درمانی: باید نامی یکتا باشد
- آدرس

حذف حساب‌های کاربری:

پزشکان می‌توانند حساب کاربری هر فرد را حذف کنند. پس از حذف حساب کاربری یک کاربر، باید همه ردیف‌های مربوط به آن کاربر در جدول‌های مختلف حذف گردد. (می‌توانید از کلید خارجی برای این کار استفاده کنید)

عملیات های پرستاران:

همان‌طور که گفته شد پرستاران سطوح مختلفی دارند و بر این اساس وظایف مختلفی نیز خواهند داشت.

ایجاد واکسن جدید (ویال):

فقط پرستارانی که سطح «مترون» دارند می‌توانند ویال‌های جدید را ثبت کنند. برای ثبت یک ویال باید موارد زیر وارد شود.

- شماره سریال: این شماره باید یک رشته عددی یکتا باشد.
- برند: برند وارد شده باید جزو برندهای ثبت شده باشد. بررسی این مورد باید به صورت خودکار انجام شود و چنان‌چه برند وارد شده در جدول برندها وجود نداشته نباید ویال اضافه شود و خطا نمایش داده شود.
- تاریخ تولید
- تعداد دوز

تزریق:

همه پرستاران می‌توانند تزریقات را ثبت کنند. برای ثبت تزریق جدید باید موارد زیر وارد شود.

- کد ملی شخص (تزریق شونده): هنگام ثبت تزریق باید کد ملی شخص در جدول کاربران ثبت شده وجود داشته باشد. این شخص می‌تواند کاربر معمولی، پرستار و یا پزشک باشد و همه افراد به شرط وجود داشتن در جداول می‌توانند واکسن دریافت کنند.
- محل تزریق: نام محل تزریق باید در جدول مراکز واکسیناسیون موجود باشد.
- سریال واکسن: سریال واکسن وارد شده باید در جدول واکسن‌ها وجود داشته باشد.
- تاریخ تزریق: تاریخ ثبت تزریق باید به صورت خودکار و به وقت ایران ذخیره شود.
- کد پرستاری تزریق‌کننده: شخص تزریق‌کننده باید نوع کاربری پرستار داشته باشد. کد پرستاری نباید توسط خود کاربر وارد شود بلکه باید به صورت خودکار و با تشخیص کاربر وارد شده به سیستم ذخیره شود.

ثبت تزریق باید در یک تراکنش انجام شود و در صورت هر مغایرتی با شرایط ذکر شده، خطای مناسب نمایش داده شود.

همان‌طور که می‌دانید برای هر ویال تعداد دوز قابل مصرف مشخص شده است. چنان‌چه تعداد تزریق‌هایی که برای یک شماره سریال ثبت شده، به این تعداد دوز مشخص برسد این ویال کاملاً مصرف شده است و نباید تزریق جدیدی با این شماره سریال ایجاد شود.

اگر شخصی دوز اول واکسنش را از برندی تزریق کرده باشد، باید دوز دومش را نیز از همان برند تزریق کند. در غیر این صورت باید از ثبت تزریق جلوگیری شود. مشخص کردن برند برای هر دوز واکسن نیز با استفاده از شماره سریال واکسن و اطلاعات ثبت شده برای آن شماره سریال در جدول ویال‌ها به راحتی قابل انجام است.

عملیات‌های کلی کاربران:

علاوه بر موارد ذکر شده، همه کاربران (کاربران معمولی، پزشکان و پرستاران) باید بتوانند اطلاعات حساب خود را مشاهده کنند، گذرواژه خود را تغییر دهند، امتیازدهی کنند و امتیاز مراکز را مشاهده کنند.

گرفتن و تغییر اطلاعات حساب:

هر کاربر باید بتواند اطلاعات شخصی و سیستمی حساب خود را مشاهده کند. همچنین باید کاربران بتوانند گذرواژه خود را تغییر دهند.

امتیازدهی:

هر کاربر باید بتواند امتیازی بین ۱ تا ۵ به مرکز واکسیناسیونی که واکسنش را در آن زده است بدهد. برای امتیازدهی، باید کاربر نام مرکز واکسیناسیون و امتیاز مورد نظرش را وارد کند. باید بررسی شود کاربری که در حال امتیاز دادن به مرکز واکسیناسیون است قبلاً در آن مرکز تزریق کرده باشد و همچنین به آن تزریق امتیاز نداده باشد. در غیر این صورت نباید امتیاز ثبت شود و پیغام مناسبی نمایش داده شود.

مشاهده امتیاز مراکز واکسیناسیون:

هر یک از کاربران باید بتوانند امتیاز همه مراکز واکسیناسیونی که امتیازی دریافت کرده‌اند را مشاهده کنند. امتیاز مرکز واکسیناسیون با میانگین گرفتن از امتیازهایی که کاربران به آن مرکز داده‌اند به دست می‌آید. لیست مراکز باید به صورت صفحه‌بندی شده و بر اساس امتیاز به صورت نزولی (مراکز با امتیاز بیشتر در صفحات اول) مرتب شده باشد و در هر صفحه ۵ مرکز را نمایش دهید یعنی اگر ۱۰ مرکز امتیاز گرفته باشند، با درخواست صفحه اول ۵ نتیجه نخست و با درخواست صفحه دوم ۵ نتیجه بعدی نمایش داده شود. (پس از میانگین گرفتن فقط تا یک رقم اعشار را نشان دهید)

مراکزی که هنوز امتیازی دریافت نکرده‌اند را در انتهای لیست و با مقدار امتیاز «بدون امتیاز» نمایش دهید.

مشاهده تعداد تزریق‌ها در هر روز:

هر یک از کاربران باید بتوانند تعداد افرادی که در هر روز واکسن دریافت می‌کنند را مشاهده کنند. روزها باید به ترتیب باشند و روزهای اخیر در نتایج بالاتر نمایش داده شوند.

مشاهده تعداد افراد واکسینه برای هر برند:

هر یک از کاربران باید بتوانند تعداد افراد واکسینه برای هر برند را مشاهده کنند. برای هر برند مشخص شده است که هر شخص باید چند دوز از این برند را تزریق کند تا فرد واکسینه باشد. برای هر برند تعداد افرادی که تعداد دوز کافی را تزریق کرده‌اند و واکسینه شده‌اند را نمایش دهید و تعداد کل افراد واکسینه را نیز مشخص کنید.

مشاهده امتیاز مراکز واکسیناسیون برای هر برند:

در هر مرکز واکسیناسیون یک یا چند برند واکسن عرضه می‌شود. کاربران باید بتوانند برای هر برند، سه مرکز واکسیناسیونی که بیشترین امتیاز را کسب کرده‌اند را به ترتیب مشاهده کنند. (برای این که مشخص کنید امتیازی که کاربر به مرکز واکسیناسیون داده مربوط به کدام برند است، باید به مشخصات تزریق کاربر در جدول تزریقات مراجعه کنید تا مشخص شود کاربر به ازای تزریق کدام برند به مرکز امتیاز داده است.)

شخصی سازی نمایش امتیازهای مراکز واکسیناسیون:

پس از تزریق دوز اول، مشخص است که کاربر برای دوزهای بعدی باید کدام برند را تزریق کند. پس می‌توان برای نمایش امتیاز مراکز واکسیناسیون فقط آن مراکزی را به کاربر نمایش داد که واکسن مورد نیاز کاربر را عرضه می‌کنند.

کار با رابط کاربری

- برای پیاده‌سازی رابط کاربری می‌توانید از زبان برنامه‌نویسی یا اسکریپت دلخواه خود استفاده کنید.
- تنها کار رابط کاربری تفسیر ورودی‌های کاربر و فراخوانی تابع‌های SQL با ورودی‌های مناسب و ساماندهی (در صورت نیاز) و نمایش خروجی این تابع‌ها می‌باشد.
- برای رابط کاربری، پیاده‌سازی برنامه‌ای که با کنسول ورودی و خروجی را بگیرد کافی‌ست.
- خروجی باید به صورت جدول باشد به صورتی که نام ستون‌ها (ویژگی‌ها) و مقدار آن‌ها مشخص باشد. (چه در GUI و چه در CLI)

نکات پایانی

- افرادی که رابط کاربری را پیاده‌سازی نمی‌کنند و تنها بخش SQL را انجام می‌دهند 80 درصد نمره را خواهند گرفت.
- پیاده‌سازی رابط کاربری به صورت GUI حداکثر ۵ درصد نمره امتیازی خواهد داشت.
- افرادی که فقط رابط کاربری را پیاده‌سازی کنند، نمره‌ای دریافت نخواهند کرد.
- منطق سامانه نباید در رابط کاربری پیاده‌سازی شود.

موفق باشید