

به نام خدا

پاسخنامه‌ی تمرین سوم درس طراحی پایگاه داده

نیمسال اول ۱۴۰۱-۱۴۰۲

سوال اول

- INSERT INTO users VALUES ('John Smith', TRUE);

نادرست, در این پرس و جو به دلیل مشخص نکردن ترتیب ورودی مقادیر ستون ها, و اینکه مقادیر طبق ترتیب ستون های users نیستند, در نتیجه 'John Smith' داخل ستون id با نوع integer insert شده که موجب بروز ارور میشود.

فرم صحیح آن به این صورت می تواند باشد:

- ❖ INSERT INTO users (full_name, enabled) VALUES ('John Smith', false);

- INSERT INTO users (id, enabled) VALUES (3, FALSE);

نادرست, به دلیل نقض NOT NULL مربوط به ستون full_name

- INSERT INTO users (id, full_name, enabled) VALUES (1, 'Alissa Jackson', TRUE);

نادرست, ستون id به دلیل primary key بودن unique نیز باید باشد, در نتیجه در این پرس و جو unique بودن ستون id نقض میشود.

- UPDATE User SET User.id = null WHERE User.id = 2

نادرست, ستون id به دلیل primary key بودن NOT NULL نیز باید باشد, در نتیجه در این پرس و جو NOT NULL بودن ستون id نقض میشود.

- DELETE FROM users WHERE full_name='john smith'

درست, با حذف ردیفی که full_name = 'John Smith', مشکلی ایجاد نمیشود.

سوال دوم

a)

album_ID	artist_ID	album_name	artist_ID	artist_Name
1	1	Raised on Radio	1	Journey
2	1	Greatest Hits	1	Journey
3	2	Bat out of Hell	2	Meat Loaf
4	2	Dead Ringer	2	Meat Loaf
5	3	The Celts	3	Enya
6	4	Poet's Heart	4	Kate Wolf

b)

artist_ID	artist_Name	album_ID	artist_ID	album_NAME
1	Journey	1	1	Raised on Radio
1	Journey	2	1	Greatest Hits
2	Meat Loaf	3	2	Bat out of Hell
2	Meat Loaf	4	2	Dead Ringer
3	Enya	5	3	The Celts
4	Kate Wolf	6	4	Poet's Heart
5	Aerosmith	null	null	null

c)

artist_ID	artist_Name	album_ID	artist_ID	album_NAME
1	Journey	1	1	Raised on Radio
1	Journey	2	1	Greatest Hits
2	Meat Loaf	3	2	Bat out of Hell
2	Meat Loaf	4	2	Dead Ringer
3	Enya	5	3	The Celts
4	Kate Wolf	6	4	Poet's Heart

d)

این کوئری با توجه به شرطی که برای join خود دارد و باید primary key های null را بررسی کند، ترکیبات غیرممکن طبیعی دو جدول را خروجی میدهد.

(نکته : این کوئری در DBMS های مختلف خروجی یکسانی ندارد و در بعضی از آنها پشتیبانی نمیشود پس هرگونه جواب به این سوال درست در نظر گرفته میشود)

سوال سوم

A)

```
SELECT *  
FROM A  
LEFT JOIN B ON A.key = B.key
```

B)

```
SELECT *  
FROM A  
LEFT JOIN B ON A.key = B.key  
WHERE B.key IS NULL
```

C)

```
SELECT *  
FROM A  
FULL OUTER JOIN B ON A.key = B.key
```

D)

```
SELECT *  
FROM A  
FULL OUTER JOIN B ON A.key = B.key  
WHERE A.key IS NULL  
OR B.key IS NULL
```

E)

```
SELECT *  
FROM A  
INNER JOIN B ON A.key = B.key
```

F)

```
SELECT *  
FROM A  
RIGHT JOIN B ON A.key = B.key
```

G)

```
SELECT *  
FROM A  
RIGHT JOIN B ON A.key = B.key  
WHERE A.key IS NULL
```

سوال چهارم

```
CREATE TABLE Drivers(  
    ID INT AUTO_INCREMENT PRIMARY KEY,  
    first_name VARCHAR(30) NOT NULL,  
    last_name VARCHAR(30) NOT NULL,  
    phone_number CHAR(11) NOT NULL UNIQUE  
);
```

```
CREATE TABLE Cars(  
    ID INT AUTO_INCREMENT PRIMARY KEY,  
    model VARCHAR(10) NOT NULL,  
    plate_number CHAR(15) NOT NULL UNIQUE,  
    driver_ID INT,  
    FOREIGN KEY (driver_ID) REFERENCES Driver(ID)  
);
```

```
CREATE TABLE Passengers(  
    ID INT AUTO_INCREMENT PRIMARY KEY,  
    first_name VARCHAR(30) NOT NULL,  
    last_name VARCHAR(30) NOT NULL,  
    phone_number CHAR(11) NOT NULL UNIQUE,  
    created_at DATE  
);
```

```
CREATE TABLE Rides(  
    passenger_ID INT,  
    driver_ID INT,  
    car_ID INT,
```

```
origin_lat VARCHAR(15) NOT NULL,  
origin_lng VARCHAR(15) NOT NULL,  
destination_lat VARCHAR(15) NOT NULL,  
destination_lng VARCHAR(15) NOT NULL,  
ride_status VARCHAR(10) NOT NULL,  
created_at DATE,  
FOREIGN KEY (passenger_ID) REFERENCES Passengers(ID),  
FOREIGN KEY (driver_ID) REFERENCES Driver(ID),  
FOREIGN KEY (car_ID) REFERENCES cars(ID)  
);
```

توجه داشته باشید که: syntax به کار رفته در این نمونه پاسخ syntax مربوط به mysql است در صورتی که پاسخ شما syntax قابل قبول برای DBMS های دیگر را داشته باشد هم شما نمره‌ی کامل را دریافت خواهید کرد.

سوال پنجم

با توجه به اینکه برای پرس و جو های خواسته شده، یک پاسخ یکتا وجود ندارد، هر پاسخی غیر از پاسخ های زیر که نتیجه نهایی را برگرداند، مورد قبول می باشد.

a)

```
SELECT sum(salary) as sum, dept_name  
FROM instructor  
GROUP BY dept_name
```

b)

```
SELECT name  
FROM instructor  
WHERE dept_name = 'Physics' and salary between 45000  
and 100000;
```

c)

```
SELECT dept_name  
FROM instructor  
GROUP BY dept_name  
HAVING sum(salary) > 120000
```

d)

```
SELECT title  
FROM course
```

```
WHERE course_id LIKE 'CS-1%'
```