

به نام خدا
تمرین شماره 2

جبر رابطه ای

1.

(الف)

Select s_number from section where p_name = x ;

$$\Pi_{s_number}(\sigma_{p_name=x}(section))$$

(ب)

Select p_name from professor where college_number=aut_number;

$$\Pi_{p_name}(\sigma_{college_number=aut_number}(professor))$$

(ج)

خروجی این دستور در واقع لیستی از دانشجویان به همراه دانشگاه آن ها و شهر آن دانشگاه است که با
عل natural join بدست آمده است

2.

(الف)

Select s_id from borrow where date = 94;

$$\Pi_{s_id}(\sigma_{date=94}(student))$$

(ب)

Select s_id, major from student natural join (

Select s_id from (Select s_id , b_id from Borrow natural join (

Select b_id from book where title = database and library_name = UT_library))));

$$\Pi_{sid,major}(student) \bowtie (\Pi_{sid}(\Pi_{sid,bid}(borrow) \\ \bowtie \left(\Pi_{bid} \left(\sigma_{title=database \text{ and } library_name=UTlibrary}(book) \right) \right)))$$

(ج)

(Select library_name from Library where city = Esfahan) natural join Book ;

$$\Pi_{library_name} \left(\sigma_{city=esfahan}(Library) \right) \bowtie book$$

(د)

؟؟؟؟ به نظر این دستور برای جبر رابطه ای را ما خوانده ایم زیرا شبیه aggregate function می باشد!!!

3.

(الف)

نام کالاهایی که قبل از سال 2014 خریداری شده اند

(ب)

نام مشتریانی که کالایی را در سال 2015 خریداری کرده باشند.

(ج) نام مشتریانی که هیچ کالایی نخریده اند

4.

(الف)

$$\Pi_{viewers}(\sigma_{viewers < 3m}(Episode))$$

(ب)

$$\Pi_{C.house} \left(\sigma_{E.num=A.num, E.season=A.season, A.name=c.name, A.num \leq 4} \left(\rho_C(characters) \right. \right. \\ \left. \left. \times \rho_E(Episodes) \times \rho_A(Appearances) \right) \right)$$

(الف)

Create view record as select name , salary , year_of_work where year_of_work > 5 and salary > 1m ;

(ب)

Select sum(salary) from ((Select employee_id from Work where company_id = 2) natural join (select id , salary from Employee)) ;

(ج)

select DISTINCT company_id from (select * from (select employee_id as id , company_id from Work) NATURAL JOIN (select id , salary from Employee) where salary > 2m) ;

(د)

select company_id , max(number_of_employee) from (select company_id , count(company_id) as number_of_employee from Work GROUP BY company_id) ;

(هـ)

select id , salary from Employee where salary > (

select max(salary) from ((

select employee_id as id from (

(select id , name from Company) NATURAL join

(select employee_id , company_id as id from Work)

) where name = "samsung") NATURAL join

(select id , salary from Employee)

)

);

3.

در دستور select سطر های تکراری می تواند وجود داشته باشد ولی می توان با دستور distinct این تکرار را از بین برد. دستورات مربوط به مجموعه ها همگی duplicate را حذف می کنند و برای نگه داری آن ها می توان از all استفاده کرد.

4. در مواقعی که داده ای برای آن ستون نداشته باشیم. استفاده از null می تواند باعث این شود که در محاسبات منطقی و ریاضی نیز داده های null ایجاد شود. همچنین داده های null در محاسبات تعداد داده نیز شرکت می کنند و خود این نیز ممکن است برای ما مشکل آفرین باشد

5.

(الف)

اسم فیلمی که بیشترین زمان اجاره شده است .

(ب)

اسامی فیلم هایی که تعداد درخواست سفارششان بیشتر از 4 باشد

(ج)

(د)

تهیه کنندگانی که فیلم تولید کرده اند و اسم فیلمشان شامل عبارت pride and prejudice باشد و قیمتشان کمینه باشد قیمت فیلم هایی که تهیه کننده دارند و نامشان شامل عبارت pride and prejudice است

نام تهیه کننده ای که فیلمی تولید کرده که نامش شامل pride and prejudice و قیمتش کمینه است در بین تمامی فیلم هایی که شامل عبارت pride and prejudice است .

6.

برای پیاده سازی راحت تر کوئری های تو در تو می توان از جداول موقت استفاده کرد.

Create table table_name as [query expression] with data

Create table temporary_table_name like original_table_name

7.

اگر r natural join n را در نظر بگیریم :

تعداد سطر های left outer join می شود : n_r

تعداد سطر های right outer join می شود : n_s

تعداد سطر های full outer join میشود : $n_r + n_s - n$

تمرین عملی :

1.

Create function is_prime(num integer) returns bool

Begin

Declare flag bool ;

Set flag = true ;

Declare temp integer ;

Set temp = num – 1;

While temp >1 do

 If num % temp = 0 then

 Flag = false ;

 End if ;

 Set temp = temp – 1 ;

End while ;

Return flag ;

End

2.

3.

create procedure insert_employee(in name varchar(50), in family varchar(50))

begin

 insert into employees values (name , family , NOW()) ;

end//

.4

```
create procedure delete_employee(in name varchar(50), in family varchar(50))  
begin
```

```
    delete from employees where employees.name = name and  
    employees.family = family ;
```

```
end//
```

```
create procedure update_employee(in name varchar(50), in family varchar(50))  
begin
```

```
    update employees set hired_date = NOW() where employees.name = name  
    and employees.family = family;
```

```
end//
```

```
create procedure update_employee(in name varchar(50), in family varchar(50))  
begin
```

```
    update employees set hired_date = NOW() where employees.name = name  
    and employees.family = family;
```

```
end//
```

```
create procedure update_employee(in name varchar(50), in family varchar(50))  
begin
```

```
    select * from employees where employees.name = name and  
    employees.family = family ;
```

```
end//
```