

به نام خدا

## پاسخنامه تکلیف دوم درس پایگاه داده ها ۱

ترم اول ۱۳۹۹ – ۱۴۰۰

---

۱- توضیح دهید که

الف) با استفاده از DDL چه چیزهایی را می توان برای یک جدول تعریف کرد؟

پاسخ :

- CREATE TABLE ایجاد یک table جدید
- ALTER TABLE اعمال تغییرات در table
- DROP TABLE پاک کردن یک table
- CREATE INDEX ایجاد یک شاخصه
- DROP INDEX حذف یک شاخص

ب) به چه منظور از index استفاده میکنیم؟

پاسخ :

عمل ایندکس گذاری یک ابزار بسیار مناسب برای بالا بردن سرعت استخراج اطلاعات است البته اگر به درستی تعریف شود!

---

۲- مزایا و معایب استفاده از varchar را نام ببرید.

پاسخ :

معایب :

اگر اندازه داده ها خیلی بزرگ باشد ، استفاده از varchar حافظه زیادی مصرف می کند و داده ها در پشت صحنه به عنوان یک مقدار TEXT اداره می شوند. بنابراین هنگام کار با مقدار VARCHAR به پردازش اضافی نیاز است و performance را تحت تأثیر قرار می دهد.

مزایا :

به دلیل اینکه varchar به همان اندازه که داده داخل آن نیاز دارد حافظه می گیرد پس نسبت به نوع داده های fixed-length حافظه کمتری نیاز دارد.

---

۳- فایده ی استفاده از foreign key در ارتباط با جامعیت داده ها را توضیح دهید.

پاسخ :

Data integrity از صحت کیفیت داده ها اطمینان حاصل می نماید و همچنین کمک می کند تا داده ها بدون تغییر و منحصر بفرد حفظ شوند. یکی از راه های اطمینان از یکپارچگی اطلاعات (Data Integrity) Referential Integrity است که این قابلیت به حفظ ارتباط بین جداول در هنگام اضافه شدن یا پاک شدن رکوردهای جدید یا قدیمی کمک می کند و به همین منظور FK ها استفاده می شوند.

۴- در چه مواقعی ممکن است حاصل avg با sum /count یکی نباشد؟

پاسخ :

اگر برای برخی از ستونهایی که از آنها استفاده می کنید مقادیر NULL داشته باشید میانگین با  $\frac{\text{مجموع}}{\text{تعداد}}$  ممکن است متفاوت باشد ، زیرا برای محاسبه تعداد ، همه ردیف ها حتی اگر NULL باشند ، محاسبه می شوند. بنابراین میانگین با  $\frac{\text{مجموع}}{\text{تعداد}}$  متفاوت می شود.

به عنوان مثال :

Query Editor

Query History

1

select \* from test;

2

3

select avg(grades) as Using\_Avg\_Function, sum(grades)/count(\*) as Using\_Sum\_Count\_Function from test;

Data Output

Explain

Messages

Notifications

grades

double precision

1

12

2




10

3

20

4

[null]

Query Editor		Query History	
<pre>1 select * from test; 2 3 select avg(grades) as Using_Avg_Function, sum(grades)/count(*) as Using_Sum_Count_Function from test;</pre>			
Data Output			
Explain			
Messages			
Notifications			
	using_avg_function double precision	 using_sum_count_function double precision	
1	14		10.5

۵- recursive join چیست؟ با یک مثال توضیح دهید.

پاسخ :

recursive join یک عمل مرکب است که شامل تکرار عملیات join است. به عنوان مثال در پایگاه داده‌ی روابط خانوادگی که برای هر فرد یک فیلد پدر و یک فیلد مادر وجود دارد ، استفاده از recursive join یک راه مناسب برای بازیابی تمام اجداد شناخته می‌شود.

۶- تفاوت دو دستور زیر را توضیح دهید :

```
SELECT DISTINCT COUNT (V_CODE) FROM PRODUCT;  
SELECT COUNT (DISTINCT V_CODE) FROM PRODUCT;
```

پاسخ :

کوئری SELECT COUNT (DISTINCT V\_CODE) تعداد مقادیر یونیک را در ستون V\_CODE می‌دهد. در حالی که کوئری SELECT DISTINCT COUNT (V\_CODE) خروجی unique counts of values می‌باشد که بدون grouping خروجی آن تنها یک آیتم است که شامل total count است. به مثال زیر دقت کنید :

Query Editor		Query History	
1	create table t(a int);		
2	insert into t values (1),(2),(3),(3);		
3	select * from t;		
4	select count (distinct a) from t;		
5	select distinct count (a) from t group by a;		
Data Output			
Explain			
Messages			
Notifications			
	a	integer	
1		1	
2		2	
3		3	
4		3	

Query Editor		Query History	
1	create table t(a int);		
2	insert into t values (1),(2),(3),(3);		
3	select * from t;		
4	select count (distinct a) from t;		
5	select distinct count (a) from t group by a;		
Data Output			
Explain			
Messages			
Notifications			
	count	bigint	
1		2	
2		1	

Query Editor		Query History	
1	create table t(a int);		
2	insert into t values (1),(2),(3),(3);		
3	select * from t;		
4	select count (distinct a) from t;		
5	select distinct count (a) from t group by a;		
Data Output			
Explain			
Messages			
Notifications			
	count	bigint	
1		3	

۷- correlated subquery چیست؟ با یک مثال توضیح دهید.

پاسخ :

به subquery هایی که حاوی correlation name از کوئری بیرونی باشند ، correlated subquery گفته می‌شود . مثلاً در اسکرپت زیر ، یک correlation name که در اینجا S است (که در کوئری بیرونی تعریف شده است) در قسمت where از subquery استفاده شده است.

```
select course id
from section as S
where semester = 'Fall' and year= 2017
and exists (select * from section as T
            where semester = 'Spring'
            and year= 2018
            and S.course id= T.course id);
```

---

۸ دستوری بنویسید که از جدول Customer (ID, Name, LastUpdate) که به اشتباه برای یک مشتری ممکن است حاوی چند رکورد باشد ،آخرین رکورد براساس تاریخ و ساعت درج شده در LastUpdate حفظ گردد و رکوردهای اضافه حذف شوند.

پاسخ :

```
Delete from TableA(SELECT *
                    FROM TableA
                    WHERE ID NOT IN (SELECT MAX(ID)
                                     FROM TableA
                                     GROUP BY Value)
                    order by LastUpdate)
```

۹- با توجه به پایگاه داده University که در کلاس درس مورد بررسی قرار گرفت، برای هر یک از موارد زیر اسکرپت های SQL مناسب بنویسید.

( A ) نام تمام دپارتمان هایی که بودجه شان از بودجه ی دپارتمان فلسفه بیشتر است را به صورت مرتب شده براساس حروف الفبا.

پاسخ :

```
select dept_name
from department
where budget > (select budget
                 from department
                 where dept_name = 'Psychology')
```

	dept_name [PK] character varying (20)
1	Physics
2	Finance

( B ) برای تمام دانشجویانی که یک درس را حداقل سه بار اخذ کرده اند، لیستی شامل شماره دانشجویی و course\_id درس مربوطه را نشان دهید.

پاسخ :

```
select student.id, course_id, count(*) as Count_Course
from student, takes
where student.id = takes.id
group by student.id, course_id
having count(*) >= 3
```

	id character varying (5)	course_id character varying (8)	count_course bigint
1	39925	362	3
2	27236	362	3
3	5414	362	3
4	16480	362	3
5	49611	362	3
6	69581	362	3
7	16969	362	3
8	39978	362	3
9	44881	362	3
10	9993	362	3

( C ) نام و id مربوط به اساتیدی که کلیه درسهای ارائه شده در دانشکده خودشان را تدریس کرده اند.

پاسخ :


```
select *
from instructor as INS
where not exists ((select course_id
                    from course
                    where course.dept_name = INS.dept_name)
except
(select course_id
 from teaches
 where teaches.id = INS.id))
```

خروجی تهی است !

( D ) نام همه دانشجویان دانشکده تاریخ را که اسمشان تنها سه حرف دارد، لیست کنید.

پاسخ :




```
select name
from student
where dept_name = 'History' and name like '___'
```

	name character varying (20)	
1	Yap	
2	Sud	
3	Maw	
4	Usi	
5	Ssu	

۱۰- برای پایگاه داده dvd rental برای هریک از موارد زیر اسکریپت های SQL مناسب بنویسید. ممکن است برای هر مورد بیش از یک پاسخ درست موجود باشد.



( A ) نام، نام خانوادگی و اسم شهر مشتریان ایرانی که نام آنها ۵ حرفی است .  
پاسخ :

```
Select first_name,last_name,city
from customer,address,city,country
where customer.address_id = address.address_id
and city.city_id = address.city_id
and city.country_id = country.country_id
and country = 'Iran'
and first_name like '_____'
```

	 first_name character varying (45)	 last_name character varying (45)	 city character varying (50)
1	Harry	Arce	Najafabad
2	Tommy	Collazo	Qomsheh
3	Oscar	Aquino	Sirjan

( B ) نام تمام فیلم هایی که کمتر از ۱۰۰ دقیقه اند و امتیاز آنها کمتر از ۲ بوده است و در فروشگاه ۲ فروخته شده اند و مشتری آنها را در مدت کمتر از یک روز به فروشگاه برگردانده، لیست کنید.  
پاسخ :

```
Select title
from film ,inventory,rental
where film.film_id = inventory.film_id
and rental.inventory_id = inventory.inventory_id
and rental_rate < 2
and return_date::date - rental_date::date < 1
and store_id = 2
and length < 100
```

	 title character varying (255)	
1	Caddyshack Jedi	
2	Cheaper Clyde	
3	Tomatoes Hellfighters	

(C) به دو روش، لیست همه بازیگرانی که در فیلم های با امتیاز بیشتر از ۴ بازی کرده اند اما بازیگر در هیچ فیلمی که بیشتر از ۱۸۰ دقیقه است بازی نکرده باشد.

پاسخ :

```
-- Method One
Select title
from film ,inventory,rental
where film.film_id = inventory.film_id
      and rental.inventory_id = inventory.inventory_id
      and rental_rate < 2
      and return_date::date - rental_date::date < 1
      and store_id = 2
      and length < 100

-- Method Two
(select distinct actor.actor_id,actor.first_name
 from film,film_actor,actor
 where (film.film_id = film_actor.film_id)
        and (film_actor.actor_id = actor.actor_id)
        and rental_rate > 4)
except
(select actor.actor_id,actor.first_name
 from film,film_actor,actor
 where (film.film_id = film_actor.film_id)
        and (film_actor.actor_id = actor.actor_id)
        and length > 180)
order by actor_id
```

	actor_id [PK] integer	first_name character varying (45)
1	2	Nick
2	6	Bette
3	7	Grace
4	8	Matthew
5	10	Christian

\* خروجی ۷۱ سطر است که ۵ سطر اول آن

عبارت است از :



( D ) لیست نام همه بازیگرانی که در فیلم های با امتیاز بیشتر از ۴ بازی کرده اند و لیست نام تمام مشتریانی که فیلم هایی با امتیاز کمتر از ۱ کرایه کردند و در مدت کمتر از یک روز به فروشگاه برگردانده اند.

پاسخ :

```
(select customer.first_name
  from film, inventory, rental, customer
 where (film.film_id = inventory.film_id)
        and (rental.inventory_id = inventory.inventory_id)
        and (customer.customer_id = rental.customer_id)
        and rental_rate < 1
        and return_date::date - rental_date::date < 1)
union
(select actor.first_name
  from film , film_actor, actor
 where (film.film_id = film_actor.film_id)
        and (film_actor.actor_id = actor.actor_id)
        and rental_rate > 4)
```

	first_name character varying (45)
1	Sylvester
2	Humphrey
3	Rita
4	Renee
5	Henry

\* خروجی ۱۶۱ سطر است که ۵ سطر اول آن

عبارت است از :

( E ) لیست نام تمامی بازیگرانی که در فیلم هایی بازی کرده اند که امتیاز آنها از امتیاز تمامی فیلم هایی که مدت زمانشان بیشتر از ۱۸۴ دقیقه است کمتر است را تهیه کنید.

پاسخ :

```
select distinct actor.first_name
from film,film_actor,actor
where (film.film_id = film_actor.film_id)
      and (film_actor.actor_id = actor.actor_id)
      and rental_rate < all (select rental_rate
                             from film
                             where length > 184)
```

\* خروجی ۱۲۸ سطر است که ۵ سطر اول آن

عبارت است از :

	first_name character varying (45)
1	Adam
2	Al
3	Alan
4	Albert
5	Alec

( F ) لیست id، جمع مبلغ پرداختی و تعداد سفارش مشتری هایی که تعداد سفارشات آنها کمتر از ۱۵ تاست را تهیه کنید.

پاسخ :

```
select customer_id,sum(amount),count(rental_id)
from payment
group by customer_id
having count(rental_id) < 15
```

	customer_id smallint	sum numeric	count bigint
1	272	65.87	13
2	315	67.86	14
3	110	49.88	12
4	281	32.90	10
5	464	67.86	14
6	318	27.93	7
7	136	59.86	14
8	48	67.86	14
9	61	57.87	13
10	124	57.86	14
11	310	68.87	13
12	248	37.87	13

G) لیست مشتری هایی که تعداد سفارشهای آنها از میانگین تعداد سفارشهای کل مشتریان بیشتر است.  
پاسخ :

```
with S as (select payment.customer_id, count (rental_id) as CT
            from payment
            group by payment.customer_id)
select customer_id, CT
from S
where CT > (select avg (CT)
            from S)
```

	customer_id smallint	ct bigint
1	87	28
2	273	28
3	550	31
4	51	30
5	190	25

\* خروجی ۲۷۹ سطر است که ۵ سطر اول آن

عبارت است از :

( H ) لیست نام پر امتیاز ترین و طولانی ترین فیلم هر ژانر را تهیه کنید.

پاسخ :

```
with S as(select distinct category_id,max(length)as ML
          from film,film_category
          where film.film_id=film_category.film_id
          group by category_id),
      T as(select distinct category_id,max(rental_rate)as MS
          from film,film_category
          where film.film_id=film_category.film_id
          group by category_id),
      A as(select T.category_id,title as longest
          from T,S,film_category,film
          where T.category_id = S.category_id
               and film_category.category_id = S.category_id
               and film_category.film_id=film.film_id
               and film.length = S.ML),
      B as(select T.category_id,title as max_score
          from T,S,film_category,film
          where T.category_id = S.category_id
               and film_category.category_id = S.category_id
               and film_category.film_id=film.film_id
               and film.rental_rate = T.MS)

select A.category_id,max_score,longest
from A,B
where A.category_id = B.category_id
```

	category_id smallint	max_score character varying (255)	longest character varying (255)
1	12	Chamber Italian	Home Pity
2	12	Grosse Wonderful	Home Pity
3	11	Airport Pollock	Analyze Hoosiers
4	11	Airport Pollock	Love Suicides
5	7	Bright Encounters	Jacket Frisco

\* خروجی ۴۷۵ سطر است که ۵ سطر اول

آن عبارت است از :

\* توجه : با توجه به اینکه این قسمت ابهام داشت ، لذا تمام فرض های معقول بررسی و سپس نمره دهی خواهد شد.

I) تعداد فیلم های اجاره شده در هر ژانر به صورت مجزا به چه میزان است؟ (برای مثال در ژانر ورزش، چند فیلم مجزا اجاره داده شده است؟)

پاسخ :

```
SELECT c.name AS Genre,
       count(DISTINCT cu.customer_id) AS Total_rent_demand
FROM category c, film_category fc, film f, inventory i,
     rental r, customer cu
where c.category_id = fc.category_id
     and f.film_id = fc.film_id
     and f.film_id = i.film_id
     and i.inventory_id = r.inventory_id
     and r.customer_id = cu.customer_id
GROUP BY 1
ORDER BY 2 DESC;
```

	genre character varying (25)	total_rent_demand bigint
1	Sports	519
2	Action	510
3	Sci-Fi	507
4	Family	501
5	Drama	501
6	Animation	500
7	Comedy	495
8	Foreign	493
9	Documentary	483
10	Children	482
11	Games	474
12	New	468
13	Classics	468
14	Horror	451
15	Music	447
16	Travel	442

J) دستوری بنویسید که جدولی با دو ستون با نام های rating و favorite\_genre را برگرداند.

Rating بیانگر رده سنی است و مقادیر آن به شکل زیر است:

G: General audiences

PG: Parental guidance suggested

R: Restricted

PG-13: Parents strongly cautioned

NC-17: No children under 17 admitted

favorite\_genre بیانگر ژانری است که بیشترین تعداد فیلم در آن رده سنی را داشته باشد.

پاسخ :

```
select distinct
case
  when rating = 'G' then 'General audiences'
  when rating = 'R' then 'Restricted'
  when rating = 'PG' then 'Parental guidance suggested'
  when rating = 'PG-13' then 'Parents strongly cautioned'
  when rating = 'NC-17' then 'No children under 17 admitted'
end as rating,
case
  when rating = 'G' then (with T as (select
category_id,count(film.film_id) as CT from film,film_category where
film_category.film_id = film.film_id and film.rating = 'G'
group by category_id)
select category_id from T where CT = (
select max(CT) from T))

  when rating = 'R' then (with T as (select
category_id,count(film.film_id) as CT from film,film_category where
film_category.film_id = film.film_id and film.rating = 'R'
group by category_id)
select category_id from T where CT = (
select max(CT) from T))

  when rating = 'PG' then (with T as (select
category_id,count(film.film_id) as CT from film,film_category where
film_category.film_id = film.film_id and film.rating = 'PG'
group by category_id)
select category_id from T where CT = (
select max(CT) from T))

  when rating = 'PG-13' then (with T as (select
category_id,count(film.film_id) as CT from film,film_category where
film_category.film_id = film.film_id and film.rating = 'PG-13'
group by category_id)
select category_id from T where CT = (
select max(CT) from T))

  when rating = 'NC-17' then (with T as (select
category_id,count(film.film_id) as CT from film,film_category where
film_category.film_id = film.film_id and film.rating = 'NC-17'
group by category_id)
select category_id from T where CT = (
select max(CT) from T))
end as favorite_genre
from film,film_category where film_category.film_id = film.film_id
```

	rating text	favorite_genre smallint
1	General audiences	1
2	Parental guidance suggested	8
3	No children under 17 admitted	12
4	Restricted	14
5	Parents strongly cautioned	7

K) دستوری بنویسید در هر ژانر تعداد فیلم هایی که دیرتر، زودتر و یا به موقع تحویل داده شده اند را بیابید.  
پاسخ :

```
with T1 as (select *,DATE_PART('day', return_date -
rental_date) as date_diff from rental),
T2 as (select film.film_id,rental_duration, date_diff,
case
when rental_duration > date_diff then 'Returned
early'
when rental_duration = date_diff then 'Returned
on Time'
else 'Returned late'
end as return_status
from film, inventory,T1
where film.film_id = inventory.film_id
and T1.inventory_id = inventory.inventory_id
),
C1 as (select category_id,count(T2.film_id) as early from
T2,film_category where
T2.film_id=film_category.film_id and T2.return_status
= 'Returned early'
group by category_id),
C2 as (select category_id,count(T2.film_id) as late from
T2,film_category where
T2.film_id=film_category.film_id and T2.return_status
= 'Returned late'
group by category_id),
C3 as (select category_id,count(T2.film_id) as onTime from
T2,film_category where
T2.film_id=film_category.film_id and T2.return_status
= 'Returned on Time'
group by category_id)

select C1.category_id,early,late,onTime from C1,C2,C3 where
C1.category_id = C2.category_id and C1.category_id =
C3.category_id
```

	category_id smallint	early bigint	late bigint	ontime bigint
1	1	541	466	105
2	2	571	473	122
3	3	482	363	100
4	4	472	372	95
5	5	420	405	116
6	6	472	456	122
7	7	547	410	103
8	8	540	424	132
9	9	518	394	121
10	10	443	410	116
11	11	396	355	95
12	12	419	316	95
13	13	418	437	85
14	14	510	492	99
15	15	516	544	119
16	16	473	269	95

موفق باشید.