بسمه تعالی

پاسخنامه تکلیف سوم درس پایگاه داده ها ۱

۱. جداول زیر را در نظر بگیرید.

- جدول کاربران شامل ۱۰۰ رکورد یکتا که با c_id یکتا میشود. (customers)
- جدول شماره تلفن ها شامل ۱۰۰ رکورد یکتا که هر سطر آن به شکل (c_id, telephone_number) است. (phones)
- فرضیات : هر کاربر می تواند چند شماره تلفن داشته باشد و واضح است که یک شماره تلفن نمی تواند مربوط به دو کاربر باشد. امکان این که کاربری بدون شماره تلفن ثبت شده باشد وجود دارد و واضح است که شماره تلفن بدون کاربر در جداول ذخیره نمی شود. پس با در نظر گرفتن این فرض ها هر سطر جدول شماره تلفن توسط (c_id, telephone_number) یکتا می شود.

مقادیر خواسته شده در جدول زیر کم ترین و بیشترین تعداد سطر خروجی حاصل از join کردن این دو جدول بر اساس c_id هستند. customers جدول سمت **چپ** و phones جدول سمت **راست** است.
مقادیر خواسته شده را به دست آورید.

	MIN	MAX
Inner join	۱۰۰	۱۰۰
Left outer join	۱۰۰	۱۹۹
Right outer join	۱۰۰	۱۰۰
Full outer join	۱۰۰	۱۹۹

۲. با توجه به پایگاه داده زیر :

جدول گره‌ها	**Node (NID, Name, Color, Description)**
	اطلاعات موجود در جدول گره‌ها شامل شماره، نام، رنگ و شرح مربوط به هر گره است.
جدول یال‌ها	**Edge (NID1, NID2, EdgeType)**
	هر سطر از جدول **Edge**، نشان دهنده وجود یک یال جهت‌دار از نوع **EdgeType** از گره با شماره **NID1** به گره با شماره **NID2** است.

پرس‌وجو اول	**select distinct** NID **from** node, edge **where** NID = edge.NID2 **and not exists** (**select** * **from** edge **where** edge.NID1 = NID)
پرس‌وجو دوم	**select** T1.NID **from** (**select count**(NID1) **as** cnt, NID **from** node **left outer join** edge **on** edge.NID1 – NID **group by** NID) T1, (**select count**(NID2) **as** cnt, NID **from** node **left outer join** edge **on** edge.NID2 – NID **group by** NID) T2 **where** T1.NID – T2.NID **and** T1.cnt<T2.cnt

نتیجه هر کدام از پرس و جو های اول و دوم را توضیح دهید و بیان کنید هر کدام چه خروجی ای برمی‌گردانند.

پرس و جوی اول لیست یکتا از شماره گره هایی را می دهد که نود انتهایی یالی جهت دار هستند اما نود ابتدایی هیچ یالی نیستند.
پرس و جوی دوم شماره گره هایی را می دهد که درجه خروجی آن ها کمتر از درجه ورودی آن هاست.

۳. با توجه به پایگاه داده دانشگاه :

a. یک view طراحی کنید که در آن لیست دانشجویان و اساتید با فیلد های شناسه دانشجو یا استاد ، نام دانشجو یا استاد و نوع دانشکده (اگر در اسم دانشکده Eng باشد نوع Engineer و در غیر این صورت Scientist) و نوع شخص(اگر استاد بود INS و اگر دانشجو STU) را نمایش دهد.

```
CREATE VIEW "view_stu_ins" AS (SELECT id,name,
(CASE WHEN dept_name LIKE '%Eng%' THEN 'Engineer' ELSE 'Scientist' END)
as dept_type,
'STU' as person_type FROM student )
UNION
(SELECT id,name,
(CASE WHEN dept_name LIKE '%Eng%' THEN 'Engineer' ELSE 'Scientist' END)
as dept_type,
'INS' as person_type FROM instructor);
```

۴.

```
CREATE VIEW "view_stu_ins" AS (SELECT id,name,
(CASE WHEN dept_name LIKE '%Eng%' THEN 'Engineer' ELSE 'Scientist' END) as dept_type,
'STU' as person_type FROM student )
UNION
(SELECT id,name,
(CASE WHEN dept_name LIKE '%Eng%' THEN 'Engineer' ELSE 'Scientist' END) as dept_type,
'INS' as person_type FROM instructor)
> OK
> Time: 0.014s
```

id	name	dept_type	person_type
96153	Sawah	Scientist	STU
37284	Benabd	Scientist	STU
82402	Grant	Scientist	STU
41596	Abeggl	Scientist	STU
58413	Xiong	Scientist	STU
19603	Colu	Scientist	STU
68554	Larsson	Engineer	STU
87193	Pinkus	Engineer	STU
48660	Emam	Scientist	STU
9933	Pircher	Scientist	STU
51416	Shan	Scientist	STU
94697	Pettersen	Scientist	STU
32119	Nagashima	Engineer	STU
38545	Fok	Engineer	STU
61356	Vulp	Scientist	STU

a. با کمک view که در سوال قبل ایجاد کردید پرس و جویی بنویسید که برای هر استاد بگوید چند درصد بودجه دانشکده اش به او تعلق می گیرد (نسبت حقوق به بودجه دانشکده) و برای هر دانشجو بگوید چه میزان از بودجه دانشکده اش به او تعلق می گیرد(نسبت بودجه دانشکده به تعداد دانشجویان دانشکده).

خروجی مورد انتظار: name, person_type, calc_number

```
SELECT v."name" ,v.person_type, (i.salary / d.budget * ١٠٠) as calc_number
from view_stu_ins as v
JOIN instructor as i ON i."id" = v."id"
JOIN department as d ON d.dept_name = i.dept_name
WHERE v.person_type = 'INS')
UNION

(SELECT v."name" , v.person_type,( d.budget / (SELECT COUNT(*) FROM student
WHERE dept_name = s.dept_name)) as calc_number
from view_stu_ins as v
JOIN student as s ON s."id" = v."id"
JOIN department as d ON d.dept_name = s.dept_name
WHERE v.person_type = 'STU')
```

name	person_type	calc_number
Kahs	STU	4647.6675294117647059
Shuming	INS	11.464241061703606286 00
Jordan	STU	8545.1110989010989011
Stylian	STU	5052.8033613445378151
Yoshimoto	STU	9238.8541860465116279
Gleit	STU	5975.5629059829059829
Papakir	STU	5975.5629059829059829
Benkov	STU	4463.0395959595959596
Noda	STU	4463.0395959595959596
Morton	STU	984.9878703703703704
So	STU	7984.2467391304347826
Chowdhury	STU	9814.1954166666666667
Crimm	STU	984.9878703703703704
Yüksel	STU	6432.0280000000000000
Kok	STU	4419.1079347826086957

۵. میخواهیم تغییرات زیر را در ساختار جداول پایگاه داده DVDRental اعمال کنیم:

a. مدیر مجموعه از ما میخواهد **از این به بعد** فیلم های زیر ۵۰ دقیقه را وارد مجموعه خود نکنیم.
برای انجام این کار constraint جدیدی برای جدول film بنویسید.

```
ALTER table film
ADD CHECK(length > ٥٠) NOT VALID
```

b. مدیر مجموعه میخواهد **از این به بعد** نوع پرداخت مشتریان را نیز در پایگاه خود ذخیره کنیم.
برای انجام خواسته مدیر یک ستون به نام PAY_TYPE به جدول payment اضافه کنید
که فقط میتواند یکی از مقادیر زیر را بپذیرد:

'credit_card' | 'cash' | 'online'

```
ALTER TABLE payment
ADD COLUMN PAY_TYPE VARCHAR(١٠) CHECK(PAY_TYPE IN ('online', 'cash',
'credit_card'))
```

٦. با توجه به پایگاه داده دانشگاه:

a. در قالب یک تراکنش دو دپارتمان به نامهای medical و dental و نام ساختمان Pasteur با بودجه به ترتیب ۷۰۰۰۰۰ و ۸۰۰۰۰۰ را به جدول دپارتمان اضافه کنید.

```
BEGIN;
```

```
INSERT INTO public.department(dept_name,building,budget)
VALUES('medical','pasture',7000000);

INSERT INTO public.department(dept_name,building,budget)
VALUES('dental','pasture',8000000);

COMMIT;
```

b. در یک تراکنش ۱۰ درصد از بودجه دپارتمان medicalرا به دپارتمان dentalاختصاص دهید.
(راهنمایی: ابتدا ۱۰ درصد از بودجه medicalرا به بودجه dentalاضافه کنید و سپس ۱۰ درصد از بودجه medicalکم کنید)

```
BEGIN;

UPDATE department
SET budget = budget+(select budget*0.1 from public.department where
dept_name='medical')
WHERE dept_name='dental';


UPDATE department
SET budget = budget*0.9
WHERE dept_name='medical';


COMMIT;
```

۷. در پایگاه داده DVDRental یک function بنویسید که شناسه یک actor را بگیرد و لیست تمام فیلم هایی که در آن نقش داشته به همراه تعداد دفعاتی که هر فیلم کرایه شده است را نمایش دهد. تصویر حاصل از خروجی function خود را به ازای ورودی دلخواه در پاسخ نامه قرار دهید.

```
create function get_actor_rental(a_id int)
returns table(rental_count bigint, fi_title varchar(255))
language plpgsql
as
$$
declare

begin
return query
   select count(rental.rental_id) as rental_count
   ,film.title as fi_title
   from actor join film_actor on actor.actor_id=film_actor.actor_id
       join inventory on inventory.film_id=film_actor.film_id
       join film on film.film_id=film_actor.film_id
       join rental on rental.inventory_id=inventory.inventory_id
   where actor.actor_id=a_id group by(film.film_id);

end;
$$;
```

۸. در پایگاه داده DVDRental یک procedure بنویسید که طی یک تراکنش نام دو فیلم را بگیرد و ۵ درصد از replacement_cost فیلم اول را از آن کم کرده و به فیلم دوم اضافه کند. این کار را با تعریف یک متغیر در داخل procedure انجام دهید. تصویر حاصل از خروجی procedure خود را به ازای ورودی دلخواه در پاسخ نامه قرار دهید.

```
create or replace procedure transfer_cost(
   f_name1 text,
   f_name2 text
)
language plpgsql
as $$
DECLARE

amount dec;

begin

       select replacement_cost*0.05 into amount
       from film where film.title = f_name1;

       update film
    set replacement_cost = replacement_cost - amount
    where film.title = f_name1;


    update film
    set replacement_cost = replacement_cost + amount
    where film.title = f_name2;


       commit;
end;$$


call transfer_cost('Academy Dinosaur','Ace Goldfinger');


select title,replacement_cost
from film where film.title = 'Academy Dinosaur' or film.title = 'Ace Goldfinger'
```

۹. در پایگاه داده DVDRental می خواهیم یک امکان جدید برای مشتریان به وجود آوریم:

a. یک trigger بنویسید که از این به بعد هر مشتری پس از این که ۳ فیلم اجاره کرد، به موعد تحویل آخرین فیلم اجاره شده توسط همان مشتری یک هفته اضافه کند.(پس از ارائه این خدمت به مشتری شمارش فیلم های اجاره شده او برای خدمت بعدی از سر گرفته شود. برای شمارش تعداد فیلم ها ستونی به نام count_check به جدول customer اضافه کنید.)

```
ALTER TABLE customer
       ADD count_check int DEFAULT 0;

       CREATE OR REPLACE FUNCTION Extension_func()
       RETURNS TRIGGER
       LANGUAGE PLPGSQL
       AS
       $$
       DECLARE temp_count int;
```

```
BEGIN

	select count_check into temp_count
	from customer
	where customer_id=NEW.customer_id;

	IF temp_count <= 2 THEN
		update customer
		set count_check=count_check+1
		where customer_id=NEW.customer_id;
	END IF;

	select count_check into temp_count
	from customer
	where customer_id=NEW.customer_id;

	IF temp_count = 3 THEN

		update rental
		set return_date = return_date + INTERVAL '1 WEEK'
		where rental_id=(select rental_id
						from rental
						where
customer_id=NEW.customer_id
						order by rental_date desc
						limit 1);

		update customer
		set count_check=0
		where customer_id=NEW.customer_id;

	END IF;

	RETURN NEW;
END;
$$

CREATE TRIGGER Conditional_Extension
AFTER INSERT
ON rental
FOR EACH ROW
EXECUTE PROCEDURE Extension_func();
```

b. کدی برای تست کردن trigger خود بنویسید و عکس مربوط به جدول rental که نشان دهنده آن است که تاریخ به درستی اضافه شده را در pdf پاسخ تحویلی خود قرار دهید.

```
insert into rental(rental_date,inventory_id,customer_id, return_date, staff_id)
	values( NOW(), 5, 1,NOW()+INTERVAl '1 WEEK', 1)

	insert into rental(rental_date,inventory_id,customer_id, return_date,
staff_id)
	values( NOW(), 6, 1,NOW()+INTERVAl '1 WEEK', 1)

	insert into rental(rental_date,inventory_id,customer_id, return_date,
staff_id)
	values( NOW(), 7, 1,NOW()+INTERVAl '1 WEEK', 1)
```

```
select * from rental where customer_id=1

select * from customer where customer_id=1
```

۱۰. در پایگاه داده DVDRental دستوری بنویسید که رتبه فیلم ها را بر اساس کل میزان مبلغ پرداختی کرایه آنها (نزولی)، یک ستون رتبه فیلم بین همه فیلم ها، یک ستون رتبه فیلم به تفکیک rating و مشخص کند که این فیلم از نظر میزان فروشش در چارک اول هست یا خیر (YES or NO) نتیجه را به ترتیب عنوان فیلم(صعودی) نمایش دهد.

خروجی مورد انتظار:

film_title,film_rating,rank_in_all,rank_in_rating,sum_amount,is_in_first_quartile

```
SELECT
    f.title as film_name ,
    f.rating as film_rating,
    rank() over (ORDER BY sum(amount) DESC) as rank_in_all,
    rank() over (PARTITION BY f.rating ORDER BY sum(amount) DESC) as
rank_in_rating,
    sum(amount) as sum_amount,

    (CASE
        WHEN (ntile(4) over (ORDER BY sum(amount) DESC)) = 1 THEN 'YES'
        ELSE 'NO'
        END) as is_in_first_quartile

  FROM payment as p
  JOIN rental as r ON r.rental_id = p.rental_id
  JOIN inventory as i ON i.inventory_id = r.inventory_id
  join film as f ON f.film_id = i.film_id
  GROUP BY f.film_id
  ORDER BY f.title
```

film_name	film_rating	rank_in_all	rank_in_rating	sum_amount	is_in_first_quartile
Academy Dinosaur	PG	718	147	33.79	NO
Ace Goldfinger	G	508	89	52.93	NO
Adaptation Holes	NC-17	699	137	34.89	NO
Affair Prejudice	G	261	47	83.79	NO
African Egg	G	557	99	47.89	NO
Agent Truman	PG	118	24	111.81	YES
Airplane Sierra	PG-13	263	64	82.85	NO
Airport Pollock	R	239	46	86.85	YES
Alabama Devil	PG-13	354	80	71.88	NO
Aladdin Calendar	NC-17	66	18	131.77	YES
Alamo Videotape	G	761	132	30.78	NO

۱۱. در پایگاه داده DVDRental می خواهیم گزارشی تحلیلی بنویسیم که در آن، جمع مبلغ پرداختی برای کرایه فیلم ها را در ماه به تفکیک درجه سنی (rating) فیلم ها مشاهده کنیم و در هر رکورد علاوه بر عدد ماه و درجه سنی، مقدار فروش ماه قبل و ماه بعد آن را نیز نیاز داریم نتیجه نهایی به ترتیب ماه صعودی باشد.

(اعداد ماه ها دقیقا پشت سر هم نیستند و لذا طبق رکورد ها ماه بعد از ۲ که رکوردی برای آن موجود می باشد ماه ۶ است)

(راهنمایی: در تهیه این گزارش تنها از توابع window استفاده نمایید)

```
SELECT
    EXTRACT(MONTH FROM payment_date) as month_number,
    f.rating,
    SUM(amount) as sum_amount,
    LAG(sum(amount),۱) OVER (PARTITION by f.rating ORDER BY
EXTRACT(MONTH FROM payment_date)) prev_month_sales,
    LEAD(sum(amount),۱) OVER (PARTITION by f.rating ORDER BY
EXTRACT(MONTH FROM payment_date)) next_month_sales


FROM payment as p
JOIN rental as r ON r.rental_id = p.rental_id
JOIN inventory as i ON i.inventory_id = r.inventory_id
JOIN film as f ON f.film_id = i.film_id
GROUP BY month_number,f.rating
ORDER BY month_number;
```

month_number	rating	sum_amount	prev_month_sales	next_month_sales
2	R	1745.78	(Null)	4782.76
2	PG	1658.99	(Null)	4757.52
2	NC-17	1667.89	(Null)	5085.03
2	PG-13	1856.58	(Null)	5316.63
2	G	1422.60	(Null)	3944.62
3	PG-13	5316.63	1856.58	6563.76
3	R	4782.76	1745.78	5461.78
3	NC-17	5085.03	1667.89	5768.44