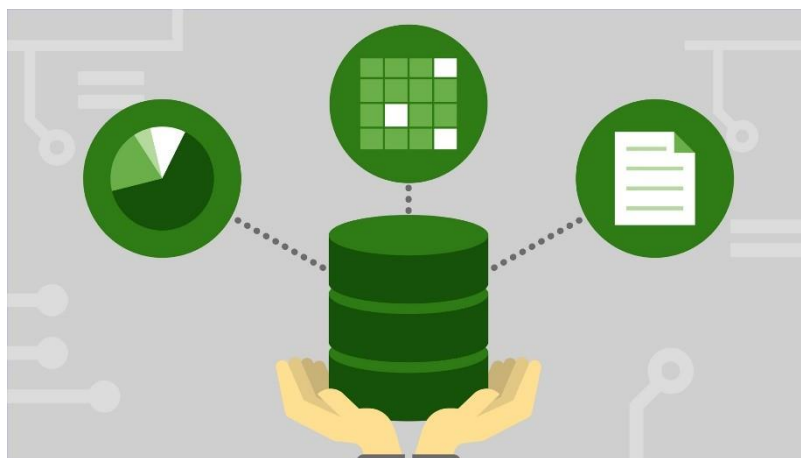


به نام خدا



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



آزمایشگاه پایگاه داده

دستور کار شماره ۴

سجاد علی‌زاده

۸۱۰۱۹۷۵۴۷

آبان ماه ۱۴۰۰

گزارش دستورکار انجام شده

سوال یک)

```

-- q1
select r.region_description , t.territory_description
from region r inner join territories t on r.region_id = t.region_id;

```

	ABC region_description	ABC territory_description
1	Eastern	Westboro
2	Eastern	Bedford
3	Eastern	Georgetow
4	Eastern	Boston
5	Eastern	Cambridge
6	Eastern	Braintree
7	Eastern	Providence

به ازای این کوئری ۵۳ خروجی به دست آمد.

سوال دو)

```

-- q2
select r.region_description , count(distinct et.employee_id)
from region r inner join territories t on r.region_id = t.region_id
inner join employee_territories et on et.territory_id = t.territory_id
group by r.region_id;

```

	ABC region_description	count
1	Eastern	4
2	Western	2
3	Northern	2
4	Southern	1

جداول region و territories و employee_territories را با یکدیگر جویین میکنیم تا هر کارمند را در کنار منطقه ای که در آن کار میکند داشته باشیم. حال بر اساس آیدی منطقه گروه بندی انجام میدهم و تعداد کارمندان در هر منطقه را میشماریم. توجه کنید برای count از عبارت distinct استفاده کردیم زیرا ممکن است یک نفر در چند ناحیه از یک منطقه فعالیت کند.

سوال سه

```

--- q3
select o.order_id , sum(od.quantity * od.unit_price * (1 - od.discount))
from orders o inner join order_details od on o.order_id = od.order_id
group by o.order_id;

```

orders 1 X

select o.order_id , sum(od.quantity * od.ur | Enter a SQL expression to filter results (use Ctrl+Space)

	order_id	sum
1	11,038	732.5999981999
2	10,782	12.5
3	10,725	287.7999954224
4	10,423	1,020

جداول orders و order_details را جوین میکنیم تا هر order_details را در کنار order آن داشته باشیم. سپس بر اساس شماره سفارش گروه بندی انجام میدهیم و به ازای هر شماره سفارش مجموع هزینه order_details آن را با استفاده از sum به دست می‌آوریم.

سوال چهار

```

--- q4
with order_count(product_id, quantity) as
(
    select od.product_id , sum(od.quantity)
    from order_details od
    group by od.product_id
)
select p.product_name , oc.quantity
from order_count oc inner join products p on oc.product_id = p.product_id
order by oc.quantity desc
limit 10;

```

products 1 X

with order_count(product_id, quantity) as | Enter a SQL expression to filter results (use Ctrl+Space)

	product_name	quantity
1	Camembert Pierrot	1,577
2	Raclette Courdavault	1,496
3	Gorgonzola Telino	1,397
4	Gnocchi di nonna Alice	1,263
5	Pavlova	1,158
6	Rhönbräu Klosterbier	1,155
7	Guaraná Fantástica	1,125
8	Boston Crab Meat	1,103
9	Tarte au sucre	1,083
10	Chang	1,057

ابتدا جدولی با نام order_count درست میکنیم که نشان میدهد هر محصول چند بار خریده شده است. سپس این جدول را با جدول محصولات جوین میکنیم تا بتوانیم نام هر محصول را نیز داشته باشیم. سپس بر اساس مقدار خریده شده به صورت نزولی مرتب میکنیم (order_by) و ۱۰ مقدار اول را نمایش میدهیم (limit). توجه کنید اگر محصولی هیچ خریدی نداشته باشد در نتیجه این کوئری نمی‌آید.

سوال پنج)

```

--- q5
select p.product_name
from products p left join order_details od on p.product_id = od.product_id
where od.order_id is null;

```

products 1 X

select p.product_name from products p le | Enter a SQL expression to filter results (use Ctrl+Space)

	ABC product_name

از left join استفاده میکنیم تا در هنگام جوین اگر محصولی وجود دارد که هیچ خریدی برای آن نیست، آن محصول در نتیجه جوین بیاید ولی مقدار order_id آن نال باشد. چون خروجی این کوئری خالی است پس هیچ محصولی وجود ندارد که فروش نرفته باشد.

سوال شش)

```

--- q6
select p.product_name , count(od.order_id)
from products p left join order_details od on p.product_id = od.product_id
group by p.product_id;

```

products 1 X

select p.product_name , count(od.order_id) | Enter a SQL expression to filter results (use Ctrl+Space)

	ABC product_name	count
1	Longlife Tofu	13
2	Tourtière	36
3	Thüringer Rostbratwurst	32
4	Flotemysost	42
5	Chef Anton's Cajun Seasoning	20
6	Scottish Longbreads	34

جداول products و order_details را جوین میکنیم تا هر کنار اطلاعات محصول آن داشته باشیم. توجه کنید از left join استفاده کردیم تا اگر برای محصولی هیچ سفارشی وجود نداشت آن محصول در خروجی بیاید. سپس بر اساس شناسه محصول گروه بندی انجام دادیم و تعداد order_detail هایی که مربوط به یک محصول بود را با استفاده از count شمردیم.

سوال هفت)

```

--- q7|
with order_price(order_id, employee_id, order_year , price) as
(
    select o.order_id , o.employee_id , extract (year from o.order_date)
        , sum(od.quantity * od.unit_price * (1 - od.discount))
    from orders o inner join order_details od on o.order_id = od.order_id
    group by o.order_id
), employees_sales(employee_id, last_name, total_sales) as
(
    select e.employee_id , e.last_name , sum(op.price) as total_sales
    from employees e left join order_price op on e.employee_id = op.employee_id
    where op.order_year = 1997
    group by e.employee_id
)
select last_name, total_sales
from employees_sales
order by total_sales desc
limit 1;

```

employees 1 X		
with order_price(order_id, employee_id, or Enter a SQL expression to filter results (use Ctrl+Space)		
	last_name	total_sales
1	Peacock	128,809.7907753933

ابتدا جدول موقتی با نام order_price ایجاد میکنیم که در آن چهار ستون وجود دارد: شناسه سفارش، شناسه کارمند، سالی که آن سفارش ثبت شده است و هزینه آن سفارش. برای به دست آوردن این جدول از کوئری سوال سه استفاده کرده‌ایم. همچنین همانطور که مشخص است از دستور extract برای به دست آوردن سال از تاریخ استفاده کردیم. پس از آن یک جدول موقت دیگر با نام employees_sales ایجاد میکنیم که در آن به ازای هر کارمند مجموع سفارشات آن کارمند در سال ۱۹۹۷ را محاسبه کردیم. برای محاسبه این جدول، جداول order_price و employees_sales را جوین کردیم تا هر سفارش را در کنار کارمند مربوطه داشته باشیم. (از left join استفاده کردیم تا اگر کارمندی هیچ سفارشی ندارد باز هم در این جدول بیاید) پس از آن فقط ردیف هایی که سال آنها برابر ۱۹۹۷ است مورد بررسی قرار دادیم و بر اساس شناسه کارمندان گروه بندی کردیم. پس از محاسبه این جدول آن را بر اساس فروش های کارمندان (total_sales) به صورت نزولی مرتب کردیم (order by) و عضو اول آن را نمایش دادیم (limit 1) که همان کارمند با بیشترین فروش است.

سوال هشت)

```

--- q8
select o.order_id ,
       case
         when o.shipped_date = o.order_date then 'Perfect'
         when o.shipped_date - o.order_date <= 3 then 'Good'
         when o.shipped_date - o.order_date > 3
              or o.shipped_date - o.order_date isnull then 'Bad'
         end order_label
from orders o;

```

orders 1 X

select o.order_id, case when o.shipped_da | Enter a SQL expression to filter results (use Ctrl+Space)

	order_id	order_label
1	10,248	Bad
2	10,249	Bad
3	10,250	Bad
4	10,251	Bad

با استفاده از case و when این کوئری را نوشتیم. اگر shipped_date برابر order_date باشد برچسب Perfect در خروجی نمایش داده میشود. اگر اختلاف این دو مقدار کمتر مساوی ۳ روز باشد برچسب Good و اگر اختلاف از ۳ روز بیشتر باشد یا این اختلاف null باشد (یعنی shipped_date هنوز مقدار ندارد) برچسب Bad نمایش داده میشود. توجه کنید اگر دو مقدار از جنس date را از یکدیگر کم کنیم اختلاف زمانی بر اساس روز به دست می‌آید.

سوال نه) فرض کنید می‌خواهیم employee‌هایی را که به صورت مستقیم یا غیرمستقیم به employee با شناسه ۲ گزارش می‌دهند (فیلد reports to) پیدا کنیم. (گزارش غیر مستقیم یعنی یک کارمند مستقیماً به کارمند شماره ۲ گزارش ندهد اما به کسی گزارش بدهد که او به کارمند شماره ۲ گزارش می‌دهد. البته گزارش دادن مدیر کارمند خاطی نیز می‌تواند به صورت غیر مستقیم باشد)

```

--- q9
with recursive reports_to as
(
    select e.employee_id , e.reports_to , e.last_name
    from employees e
    where e.employee_id = 2
    union
    select e.employee_id, e.reports_to , e.last_name
    from employees e inner join reports_to rt on e.reports_to = rt.employee_id
)
select *
from reports_to;

```

Results 1

with recursive reports_to as (select e.empl | Enter a SQL expression to filter results (use Ctrl+Space)

	employee_id	reports_to	last_name
1	2	[NULL]	Fuller
2	1	2	Davolio
3	3	2	Leverling
4	4	2	Peacock
5	5	2	Buchanan
6	8	2	Callahan
7	6	5	Suyama
8	7	5	King
9	9	5	Dodsworth

نتیجه شامل افرادی خواهد بود که مستقیماً به کارمند شماره ۲ گزارش می‌دهند (۱ و ۳ و ۴ و ۵ و ۸) یا به افرادی گزارش می‌دهند که آنها به کارمند شماره ۲ گزارش می‌دهند (۶ و ۷ و ۹). کوئری نوشته شده به صورت recursive در هر گام کارمندی را که به کارمندان پیدا شده در گام قبل گزارش می‌دهند پیدا میکند تا جایی که در یک گام نتواند چیزی پیدا کند. در نهایت نیز اجتماع نتیجه تمام گام‌ها را برمیگرداند.

(سوال ده)

به ازای تاریخ‌های بین ۱۹۹۶/۱۲/۲۴ و ۱۹۹۷/۹/۳۰ سفارشی که در آن تاریخ ارسال شده است در کنار مجموع هزینه سفارش و سال ارسال سفارش در خروجی آمده است. اگر در یک تاریخ بیش از یک سفارش ارسال شده باشد به ازای هر سفارش یک سطر در خروجی قرار خواهد گرفت. خروجی نیز بر اساس این تاریخ مرتب شده است.

```

--- q10
select extract(year from a.shipped_date) as year_, sum(b.Subtotal)
from orders a
inner join
(
  select distinct od.order_id , sum(od.unit_price * od.quantity) as Subtotal
  from order_details od
  group by od.order_id
) b on a.order_id = b.order_id
where a.shipped_date is not null
group by year_

```

Results 1 X

select extract(year from a.shipped_date) as | Enter a SQL expression to filter results (use Ctrl+Space)

	123 year_	123 sum
1	1,996	210,112.3011622429
2	1,997	649,038.8097105026
3	1,998	467,863.7197151184

قسمت order_by و شرط موجود در where را حذف میکنیم تا تمام زمان‌ها در نظر گرفته شوند. بر اساس سال گروه بندی میکنیم و با استفاده از اپراتور sum مجموع فروش‌ها در هر سال را به دست می‌آوریم.

(سوال یازده)

```

--- q11
create or replace view low_products as
(
  select p
  from products p where p.units_in_stock < p.reorder_level
  order by p.units_in_stock asc
);
select * from low_products;

```

low_products 1 X

select * from low_products | Enter a SQL expression to filter results (use Ctrl+Space)

	123 product_id	abc product_name	123 supplier_id	123 category_id	abc quantity_per_unit	123 unit_price	123 units_in_stock
1	31	Gorgonzola Telino	14	4	12 - 100 g pkgs	12.5	0
2	21	Sir Rodney's Scones	8	3	24 pkgs. x 4 pieces	10	3
3	74	Longlife Tofu	4	7	5 kg pkg.	10	4

یک ویو با نام low_products تعریف شده است که در آن محصولاتی که موجودی آنها از reorder_level کمتر است بازگردانده میشوند. همچنین خروجی بر اساس موجودی به صورت صعودی مرتب شده است. خروجی شامل ۱۸ سطر است.

سوال دوازده)

```

--- q12
with france_categories(category_name) as
(
    select distinct c.category_name
    from orders o
        inner join order_details od on od.order_id = o.order_id
        inner join products p on p.product_id = od.product_id
        inner join categories c on c.category_id = p.category_id
    where o.ship_country = 'France'
)
select *
from categories
where category_name not in (select fc.category_name from france_categories fc)

```

categories 1 ×

with france_categories(category_name) as | Enter a SQL expression to filter results (use Ctrl+Space)

category_id	category_name	description	picture

ابتدا در یک جدول موقت با نام france_categories نام دسته محصولات که به فرانسه ارسال داشته‌اند را ذخیره کردیم. این جدول را با جوین جداول order_details و products و categories ساختیم. سپس از جدول categories آنهایی را که در این جدول موقت نیستند (یعنی به فرانسه ارسال نداشته‌اند) به عنوان نتیجه نمایش می‌دهیم. همانطور که مشاهده می‌شود نتیجه خالی است.

سوال سیزده) همان سوال ۱۲ است.

سوال چهارده)

```

--- q14
select *
from customers c
where c.fax isnull;

```

customers 1 ×

select * from customers c where c.fax isnu | Enter a SQL expression to filter results (use Ctrl+Space)

customer_id	company_name	contact_name	contact_title	address	
1	ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312
2	BSBEV	B's Beverages	Victoria Ashworth	Sales Representative	Fauntleroy Circus
3	CHOPS	Chop-suey Chinese	Yang Wang	Owner	Hauptstr. 29
4	COMMI	Comércio Mineiro	Pedro Afonso	Sales Associate	Av. dos Lusíadas, 23
5	FAMIA	Familia Arquibaldo	Aria Cruz	Marketing Assistant	Rua Orós, 92

این کوئری ۲۲ ردیف جواب دارد.

سوال پانزده)

```

--- q15
create or replace view employee_with_age as
(
    select *, age(e.birth_date) as "employee_age"
    from employees e
);

select r.region_description , avg(ewa.employee_age)
from region r inner join territories t on r.region_id = t.region_id
    inner join employee_territories et on t.territory_id = et.territory_id
    inner join employee_with_age ewa on et.employee_id = ewa.employee_id
group by r.region_description;

```

region 1 x

select r.region_description , avg(ewa.employee_age) | Enter a SQL expression to filter results (use Ctrl+Space)

	ABC region_description	avg
1	Eastern	71 years 2 mons 25 days 20:12:37.885642
2	Western	60 years 4 mons 38 days 16:00:00
3	Southern	58 years 2 mons 19 days
4	Northern	58 years 8 mons 25 days 10:54:32.711564

ابتدا یک view به نام employee_with_age ساختیم که در کنار اطلاعات کارمندان، سن آنان را نیز نمایش دهد. سپس جداول region و employee_territories و employee_with_age را با یکدیگر جوین کردیم تا اطلاعات مناطق را در کنار کارمندان آن داشته باشیم. پس از آن بر اساس نام منطقه گروه بندی کردیم و به ازای هر منطقه میانگین سن کارمندان را با استفاده از اپراتور avg محاسبه کردیم.