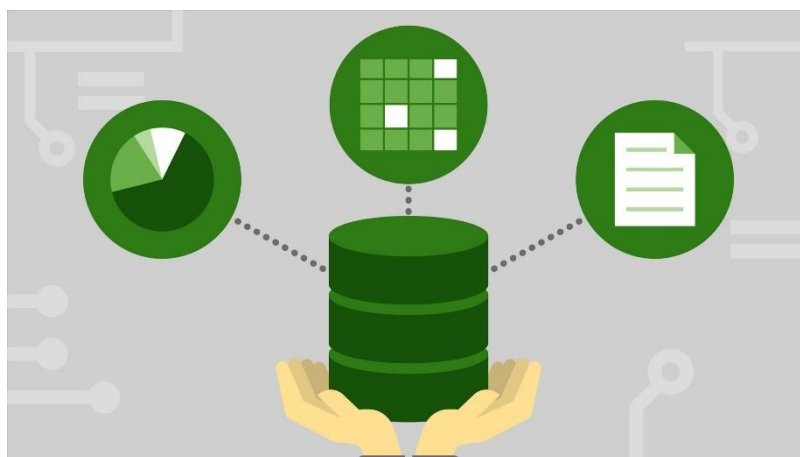


به نام خدا



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



آزمایشگاه پایگاه داده

دستور کار شماره 4

نگار مرادی

810198543

مهر 1401

گزارش دستورکار انجام شده

دیتابیس داده شده را ایجاد کردیم. در جدول زیر جداول مربوط به این دیتابیس رو میبیند.

Table Name	Size
categories	32K
customer_customer_demo	
customer_demographics	16K
customers	56K
employee_territories	24K
employees	32K
order_details	192K
orders	176K
products	24K
region	24K
shippers	24K
suppliers	32K
territories	24K
us_states	24K
Views	
Materialized Views	

(1) از جویین داخلی territories, region و سپس select کردن نام territories, region را نمایش میدهم و میبینیم که به region چه territory وجود دارد. این جدول دارای 53 ردیف است.

```
-- 1
select r.region_description , t.territory_description
from region r inner join territories t on r.region_id = t.region_id;
```

region_description	territory_description
Eastern	Westboro
Eastern	Bedford
Eastern	Georgetown
Eastern	Boston
Eastern	Cambridge
Eastern	Braintree
Eastern	Providence
Northern	Hollis
Northern	Portsmouth
Eastern	Wilton
Eastern	Morristown
Eastern	Edison

53 row(s) fetched - 16ms (1ms fetch), on 2023-01-15 at 16:48:06

(2) برای به دست آوردن محصولات ابتدا یک جویین داخلی بر روی categories, product انجام میدهیم و آن هایی که از رده خارج شده هستند را نگه میداریم و بعد از آن از توابع پنجره ای استفاده میکنیم. با استفاده از partition بر روی categories.id گروه بندی را انجام میدهیم. در انتها بر اساس تعداد محصولات از رده خارج شده جدول را مرتب میکنیم.

```
-- 2
select distinct c.category_name , count(p.product_id) over (partition by c.category_name) as discount_amount
from products p inner join categories c on p.category_id = c.category_id
where p.discontinued = 1
order by discount_amount desc;
```

	asc category_name	123 discount_amount
1	Meat/Poultry	4
2	Beverages	3
3	Condiments	1
4	Grains/Cereals	1
5	Produce	1

(3) ابتدا قیمت کل پرداختی مشتری را به ازای هر سفارش محاسبه میکنیم. برای اینکار از جویین داخلی order, order_detail استفاده میکنیم. یک partition بر روی id هر order میسازیم و قیمت کل را برمیگردانیم.

```
-- 3
select distinct od.order_id , sum(od.quantity * od.unit_price * (1 - od.discount)) over (partition by od.order_id) as total_paid
from order_details od;
```

	123 order_id	123 total_paid
1	10,384	2,222.4000968933
2	11,072	5,217.9999847412
3	10,525	818.3999944389
4	10,458	3,891.0000514984
5	10,668	625.2749989647
6	10,273	2,037.2799582076
7	10,514	8,623.4499168396
8	10,457	1,584
9	11,054	305
10	10,608	1,064
11	10,554	1,728.5250058923
12	10,643	814.4999828339

برای پیدا کردن مشتری هایی که بیشترین تخفیف را گرفته اند همه تخفیف های هر کدام را باهم جمع میزنیم و سپس به صورت نزولی مرتب میکنیم و با استفاده از district مشتری با ایدی تکراری را حذف کرده و 10 نفر اول را نمایش میدهیم. در اینجا گروه بندی بر اساس id هر customer است.

```
select distinct c.customer_id , sum(od.quantity * od.unit_price * od.discount) over (partition by c.customer_id) as total_discount
from customers c inner join orders o on o.customer_id = c.customer_id inner join order_details od on od.order_id = o.order_id
order by total_discount desc
limit 10;
```

	customer_id	total_discount
1	SAVEA	11,311.4401023215
2	ERNSH	8,361.7016382363
3	HUNGO	7,337.4850809725
4	QUICK	7,206.0851172922
5	QUEEN	4,508.6025746987
6	MEREP	3,331.710078104
7	PICCO	3,131.0900697125
8	FOLKO	2,987.9875292376
9	FRANK	2,066.150535013
10	BERGS	2,040.5724995112

4) برای به دست آوردن بیشترین تعداد خرید ابتدا جویین داخلی product, order_details را انجام میدهیم و سپس با استفاده از توابع پنجره ای پارتیشن بندی را بر روی نام محصول انجام میدهیم تا تعداد خرید را به دست آوریم. محصولات با نام تکراری را حذف میکنیم و بر اساس order_counts مرتب کرده و 10 تای اول را نشان میدهیم.

```
select distinct p.product_name , count(od.order_id) over (partition by p.product_name) as order_counts
from products p inner join order_details od on p.product_id = od.product_id
order by order_counts desc
limit 10;
```

	product_name	order_counts
1	Raclette Courdavault	54
2	Gorgonzola Telino	51
3	Camembert Pierrot	51
4	Guaraná Fantástica	51
5	Gnocchi di nonna Alice	50
6	Tarte au sucre	48
7	Jack's New England Clam Chowder	47
8	Rhönbräu Klosterbier	46
9	Chang	44
10	Pavlova	43

5) از `product, order_details, left join` استفاده میکنیم و چک میکنیم که کدام `id order` برابر `null` دارد و فروش نرفته است. همانطور که در تصویر زیر میبینید همه ی محصولات فروش رفته اند.

[illegible]

6) برای به دست آوردن کتگوری ها ابتدا بر روی categories , products left outer join انجام میدهم و سپس برای به دست آوردن تعداد سفارش ها این قسمت را با order_detail جویین داخلی میکنیم و partition را بر روی category_name انجام میدهم. در آخر کتگوری های با نام تکراری را حذف میکنیم.

	ABC category_name	123 order_counts
1	Grains/Cereals	196
2	Meat/Poultry	173
3	Dairy Products	366
4	Condiments	216
5	Confections	334
6	Produce	136
7	Beverages	404
8	Seafood	330

(7) برای به دست آوردن فروش ها در یک سال خاص ابتدا یک جدول میسازیم تا سفارشات در یک سال را نگه دارد (به عنوان مثال سال 1996) در این جدول با استفاده از select و extract سفارشات سال 1996 را نگه میداریم (ایدی سفارش و employee_id). در جدول بعدی مجموع فروش هر سفارش را در آن سال به دست می آوریم با استفاده از جوین داخلی جدول قبل که سفارش در آن سال را داشت و order_detail . سپس از توابع پنجره ای استفاده کرده و گروه بندی را بر اساس order_detail id.

در نهایت جوین خارجی جدول قبل را با employee به دست می آوریم و مجموع فروش برای هر کارمند را محاسبه میکنیم و بر اساس مقدار فروش مرتب کرده و اولین نفر را نمایش میدهیم.

```
with orders_in_year as
(
  select distinct o.order_id , o.employee_id from orders o where extract(year from o.order_date) = 1996
), orders_amount_in_year as
(
  select distinct od.order_id , o2.employee_id , sum(od.unit_price * od.quantity * (1 - od.discount))
  over (partition by od.order_id) as total_paid
  from orders_in_year o2 inner join order_details od on o2.order_id = od.order_id
)
select distinct e.employee_id , e.last_name, sum(oy.total_paid)
over (partition by e.employee_id) as solds_amount
from employees e left join orders_amount_in_year oy on e.employee_id = oy.employee_id

order by solds_amount desc
limit 1;
```

	127 employee_id	ABC last_name	127 solds_amount
1	4	Peacock	49,945.1153284936

8) با استفاده از case , when هنگامیکه order_date برابر shipping_date باشد Perfect label را می نویسیم و هنگامیکه اختلاف روز تحویل و سفارش کم تر از 3 روز باشد Good label و در غیر این صورت Bad label را میزنیم.

```
-- 8
select o.order_id ,
case
  when o.shipped_date = o.order_date then 'Perfect'
  when o.shipped_date - o.order_date <= 3 then 'Good'
  else 'Bad'
end order_label
from orders o;
```

	127 order_id	ABC order_label
1	10,248	Bad
2	10,249	Bad
3	10,250	Bad
4	10,251	Bad
5	10,252	Good
6	10,253	Bad
7	10,254	Bad
8	10,255	Good

(9) با مشاهده جدول employee در میابیم که برای فیلد reports_to می توان دستور recursive نوشت. در دستور زیر توالی کسانی را که به شخص دیگری گزارش کار میدهند را برای کارمندی با id 8 محاسبه میکنیم.

```
with recursive reporters as
(
  (select e.employee_id, e.first_name , e.last_name , e.reports_to as manager_id
   from employees e
   where e.employee_id = 8)
  union
  (select e2.employee_id, e2.first_name , e2.last_name , e2.reports_to
   from employees e2
   inner join reporters r on e2.employee_id = r.manager_id)
)
select *
from reporters r;
```

	123 employee_id	ABC first_name	ABC last_name	123 manager_id
1	8	Laura	Callahan	2
2	2	Andrew	Fuller	[NULL]

در خروجی میبینیم که کارمند با ایدی 8 به کارمند با ایدی 2 گزارش میدهد.

(10) برای ساختن این ویو باید تمام product هایی که تعداد آن ها در انبار از recorder_level کمتر باشد را نشان دهیم.

```
create view products_that_need_reorder as
(
  select p.product_id , p.product_name , p.units_in_stock
   from products p
   where p.units_in_stock < p.reorder_level
   order by p.units_in_stock asc
);
select * from products_that_need_reorder;
```

	123 product_id	ABC product_name	123 units_in_stock
1	31	Gorgonzola Telino	0
2	21	Sir Rodney's Scones	3
3	74	Longlife Tofu	4
4	66	Louisiana Hot Spiced Okra	4
5	45	Rogede sild	5
6	68	Scottish Longbreads	6
7	32	Mascarpone Fabioli	9
8	30	Nord-Ost Matjeshering	10

(11) تعداد order ها را با جوین داخلی product, categories, order, order_details محاسبه میکنیم و چک میکنیم که کشور گیرنده آلمان باشد.

```
select distinct c.category_name , count(o.order_id)
over (partition by c.category_name) as order_counts
from orders o
inner join order_details od on o.order_id = od.order_id
inner join products p on od.product_id = p.product_id
inner join categories c on c.category_id = p.category_id
where o.ship_country = 'Germany'

order by order_counts desc
```

	ABC category_name	123 order_counts
1	Dairy Products	65
2	Beverages	60
3	Confections	55
4	Seafood	49
5	Condiments	31
6	Grains/Cereals	25
7	Meat/Poultry	23
8	Produce	20

(12) برای این دستور کافیت مشتریان را بیابیم که فیلد fax آن ها null نباشد.

```
select c.customer_id , c.contact_name
from customers c
where c.fax is null;
```

	ABC customer_id	ABC contact_name
1	ANTON	Antonio Moreno
2	BSBEV	Victoria Ashworth
3	CHOPS	Yang Wang
4	COMMI	Pedro Afonso
5	FAMIA	Aria Cruz
6	FOLKO	Maria Larsson
7	GODOS	José Pedro Freyre
8	GOURL	André Fonseca
9	GREAL	Howard Snyder
10	ISLAT	Helen Bennett

آنچه آموختم / پیشنهادات

در این دستور کار با کار با توابع پنجره ای آشنا شدم که بنظرم کوثری نوشتن را ساده تر و خوانا تر میکرد و مشکل خاصی برام پیش نیومد.