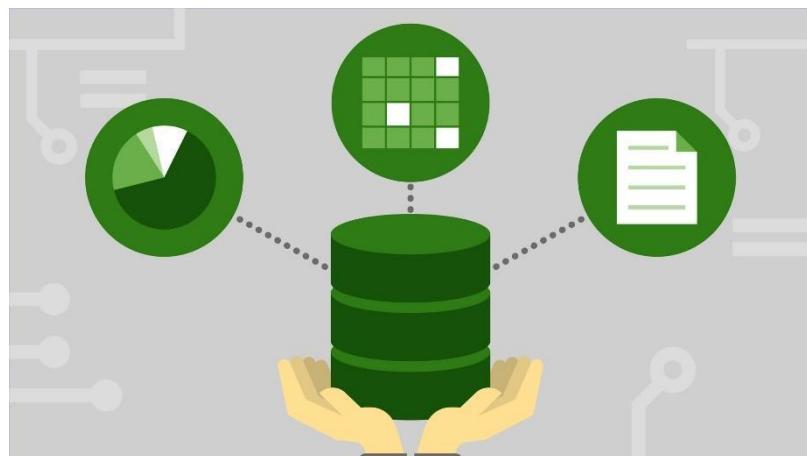


به نام خدا



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



آزمایشگاه پایگاه داده

دستور کار شماره ۴

آتیه آرمین

۸۱۰۱۹۷۶۴۸

۱۴۰۰ آذر ماه

گزارش دستور کار انجام شده

1) برای این کار همانطور که گفته شده است از inner join استفاده کرده‌ام.

```

Database Navigator
localhost - localhost:3006
postres - localhost:5432
Jobinja
atharmin
jobportal
postgres
Northwind
Schemas
public
Tables
categories
customer.customer_demo
customer_demographics
customers
employee_territories
employees
order_details
orders
products
region
shippers
suppliers
territories
us_states
Views
Materialized Views
Indexes
Functions
Sequences
Data types
Aggregate functions
Roles
Administrator
Extensions
Storage
System Info
Jobinja - localhost:5432

<postres> Script-3
select r.region_description, t.territory_description
from region r inner join territories t on r.region_id = t.region_id;

```

Grid	Region Description	Territory Description
1	Eastern	Westboro
2	Eastern	Bedford
3	Eastern	Georgetown
4	Eastern	Boston
5	Eastern	Cambridge
6	Eastern	Braintree
7	Eastern	Providence
8	Northern	Hollis
9	Northern	Portsmouth
10	Eastern	Wilton
11	Eastern	Morristown
12	Eastern	Edison
13	Eastern	New York
14	Eastern	New York

Record 53 : 53 row(s) fetched - 4ms

IRST en_IR Writable Smart Insert 3 : 1 : 122 Sel: 0 | 0

2) همانطور که گفته شده است، رابطه مستقیمی بین ناحیه و کارمند وجود ندارد و باید از جدول واسطه دو که استفاده کنیم. برای این کار نیز از inner join استفاده می‌کنیم.

```

<postres> Script-3
select r.region_description , count(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;

```

Grid	Region Description	Count
1	Eastern	19
2	Northern	11
3	Western	15
4	Southern	4

Record 4 : 4 row(s) fetched - 12ms (+1ms)

IRST en_IR Writable Smart Insert 3 : 22 : 213 Sel: 0 | 0

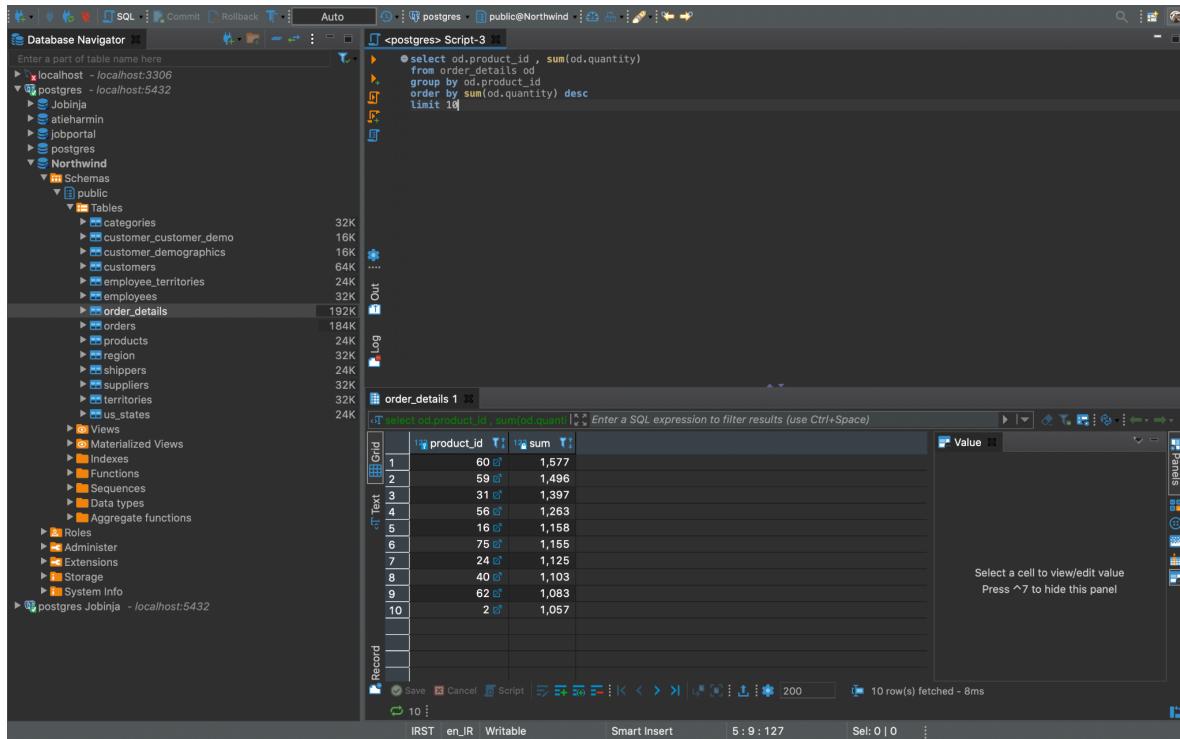
(3) برای این کار باید روی group by order_id ها بزنیم تا اطلاعات مورد نیاز را برای هر سفارش بدست آوریم. همچنین باید توجه داشته باشیم که قیمت کل پرداختی مشتری قیمت هر کلا ضرب در تعداد سفارش ضرب در ۱ منهای تخفیف کالا است.

```

SELECT od.order_id, sum(od.unit_price * od.quantity * (1 - od.discount))
FROM order_details od
GROUP BY od.order_id;
  
```

	order_id	sum
1	11,038	732.5999981999
2	10,792	12.5
3	10,725	287.7999954224
4	10,423	1,020
5	10,518	4,150.0500068665
6	10,356	1,106.3999938965
7	10,963	57.7999995947
8	10,596	1,180.8800248978
9	10,282	155.4000053406
10	10,658	4,464.5999716014
11	10,283	1,414.8000202179
12	10,579	317.75
13	10,693	2,071.1999798298
14	10,896	750.5

۴) برای آنکه ترتیب را رعایت کنیم از دستور `order by` استفاده میکنیم و سپس از دستور `limit` برای آنکه تنها ۱۰ تای آن ها را نشان دهد استفاده میکنیم.



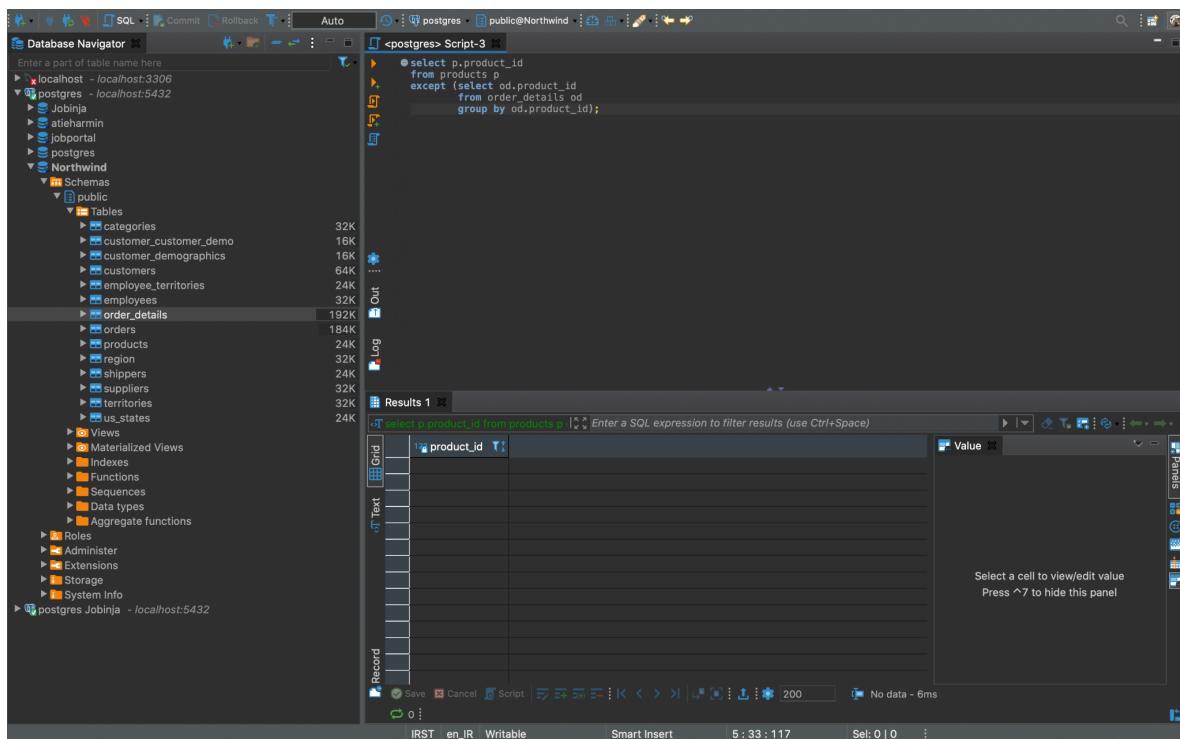
```

<postgres> Script-3
● select od.product_id , sum(od.quantity)
  from order_details od
  group by od.product_id
  order by sum(od.quantity) desc
  limit 10

```

product_id	sum
1	1,577
2	1,496
3	1,397
4	1,263
5	1,158
6	1,155
7	1,125
8	1,103
9	1,083
10	1,057

۵) خیر با انجام دستور زیر میتوانیم بینیم که هیچ محصولی نشان داده نمیشود.



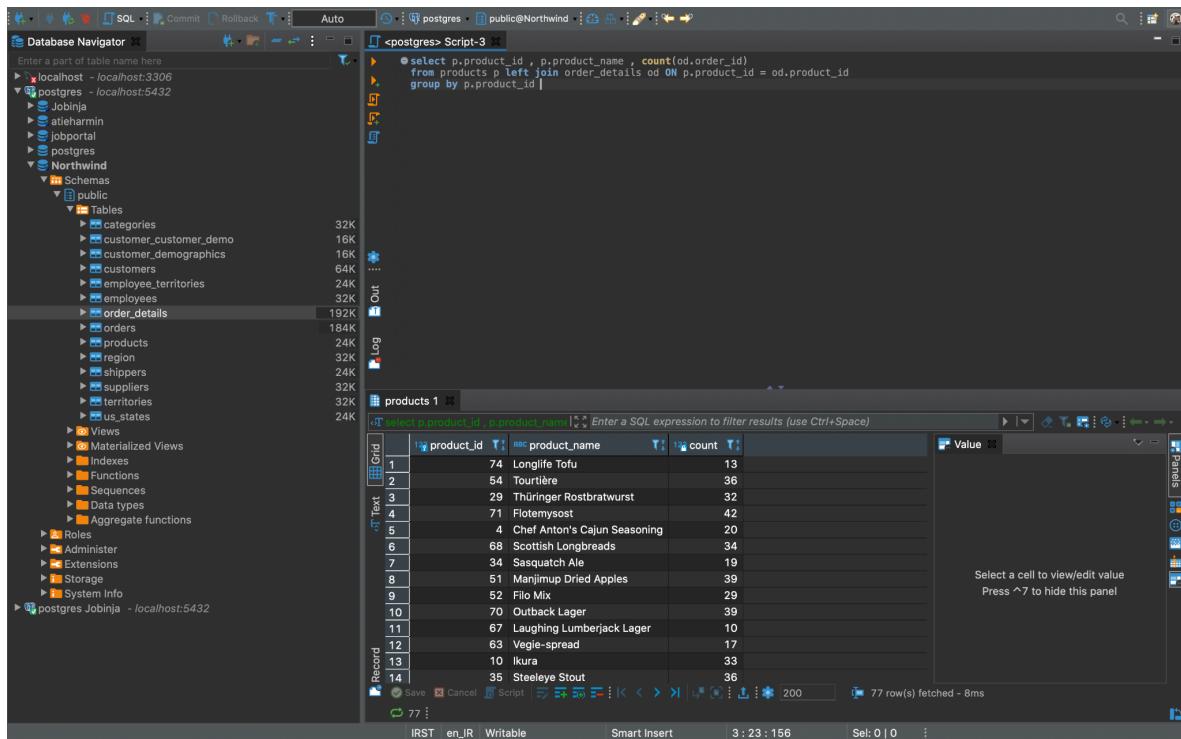
```

<postgres> Script-3
● select p.product_id
  from products p
 except (select od.product_id
          from order_details od
          group by od.product_id);

```

product_id

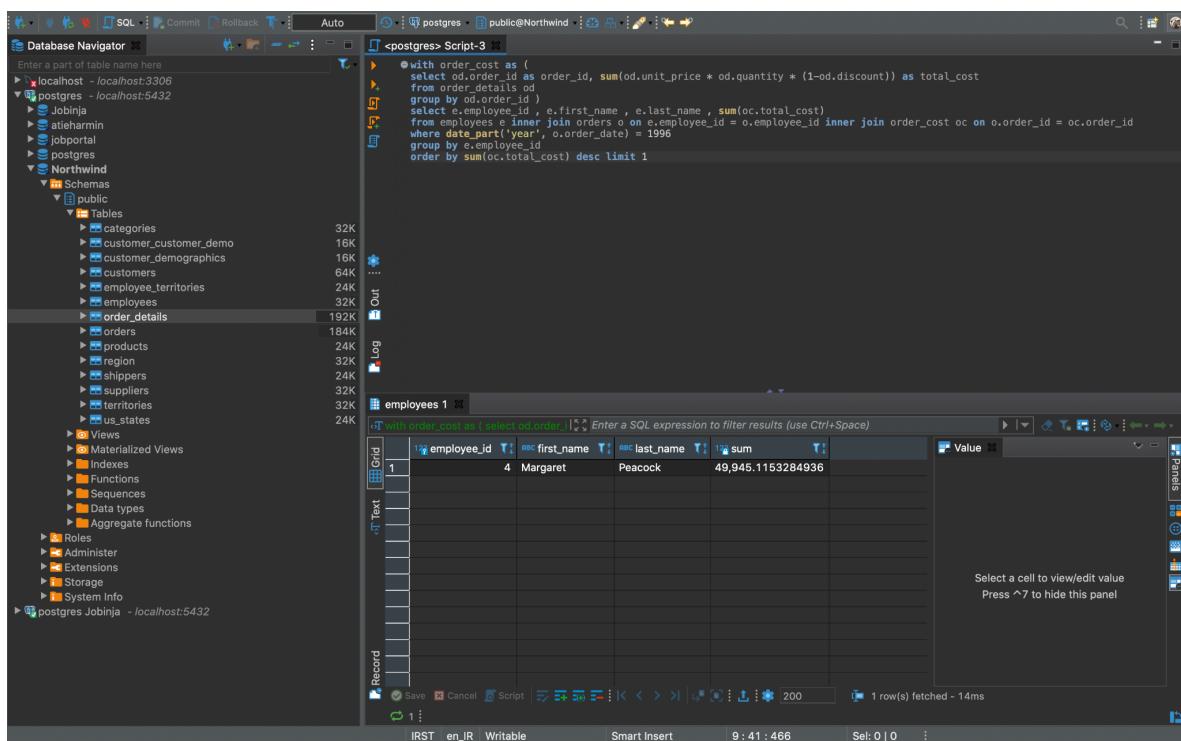
(6) برای این کار از left join استفاده میکنیم تا محصولاتی که در هیچ سفارشی نبوده‌اند هم نشان داده شوند.



```
select p.product_id , 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 |
```

product_id	product_name	count
1	Longlife Tofu	13
2	Tourtiere	36
3	Thüringer Rostbratwurst	32
4	Flötenschinken	42
5	Chef Anton's Cajun Seasoning	20
6	Scottish Longbreads	34
7	Sasquatch Ale	19
8	Manjimup Dried Apples	39
9	Filo Mix	29
10	Outback Lager	39
11	Laughing Lumberjack Lager	10
12	Vegie-spread	17
13	Ikura	33
14	Steeleye Stout	36

(7) این دستور را با توجه به راهنمایی هایی که در صورت پروژه آمده است زده‌ام.



```
with order_cost as (
    select od.order_id as order_id , sum(od.unit_price * od.quantity * (1-od.discount)) as total_cost
    from order_details od
    group by od.order_id )
select e.employee_id , e.first_name , e.last_name , sum(oc.total_cost)
from employees e inner join orders o on e.employee_id = o.employee_id inner join order_cost oc on o.order_id = oc.order_id
where date_part('year', o.order_date) = 1996
group by e.employee_id
order by sum(oc.total_cost) desc limit 1
```

employee_id	first_name	last_name	sum
1	Margaret	Peacock	49,945.1153284936

(8) برای برچسب گذاری از دستورات case - when - then استفاده میکنیم.

```


select o.order_id,
       case when DATE_PART('day', o.shipped_date::timestamp - o.order_date::timestamp) = 0 then 'نامناسب'
            when DATE_PART('day', o.shipped_date::timestamp - o.order_date::timestamp) <= 3 then 'خوب'
            else 'ناتناسب'
       end as status
  from orders o


```

order_id	status
1	نامناسب
2	ناتناسب
3	نامناسب
4	ناتناسب
5	خوب
6	نامناسب
7	ناتناسب
8	خوب
9	خوب
10	نامناسب
11	ناتناسب
12	نامناسب
13	نامناسب
14	ناتناسب

(10) این کوئری مجموع هزینه یک سفارش را در یک بازه‌ی زمانی مشخص برای ارسال، بدون در نظر گرفتن تخفیف‌ها محاسبه میکند.

```


select date_part('year', o.shipped_date), sum(b.Subtotal)
  from orders o 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 o.order_id = b.order_id
 where o.shipped_date is not null
 group by date_part('year', o.shipped_date)
 order by date_part('year', o.shipped_date)


```

date_part	sum
1,996	210,112.3011622429
1,997	649,038.8097105026
1,998	467,863.7197151184

۱۱) برای ساخت view از دستور create view استفاده میکنیم.

```

<postgres> Script-3 > new_reorders
create view new_reorders 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
  
```

The screenshot shows the pgAdmin III interface. On the left, the Database Navigator pane displays the schema structure of the Northwind database, including tables like categories, customers, and products. In the center, the main window shows a SQL script titled 'Script-3' with the command to create a view named 'new_reorders'. Below the script, the 'Statistics 1' panel shows the query and its execution details. At the bottom, the status bar indicates '0 row(s) updated - 13ms'.

The screenshot shows the pgAdmin III interface with the 'Properties' tab selected for the 'new_reorders' view. The 'Columns' section lists three columns: product_id (int2), product_name (varchar), and units_in_stock (int2). The 'Dependencies' section shows a dependency on the 'products' table. Other tabs include Data, ER Diagram, Triggers, Rules, Statistics, Permissions, Source, and Virtual. The status bar at the bottom shows '3 items'.

12) برای این دستور از `not in` استفاده میکنیم که میخواهیم `product` هایی را بیابیم که به فرانسه ارسال نشده اند.

The screenshot shows the pgAdmin 4 interface. On the left, the Database Navigator pane displays the connection to 'postgres' on 'localhost:5432' and the schema 'public'. It lists various tables such as categories, customers, products, and order_details. In the center, a script editor window titled 'Script-3' contains the following SQL query:

```

select c.category_id , c.category_name
  from products p inner join categories c on p.category_id = c.category_id
 where p.product_id not in (
    select od.product_id
      from order_details od inner join orders o on od.order_id = o.order_id
     where o.ship_country = 'France')
 group by c.category_id

```

Below the script, a results grid titled 'categories 1' shows the following data:

category_id	category_name
1	Dairy Products
2	Condiments
3	Seafood
4	Meat/Poultry

The bottom status bar indicates '4 row(s) fetched - 11ms'.

13) این بخش مانند بخش ۱۲ است.

14) برای این کار از `null` استفاده میکنیم.

The screenshot shows the pgAdmin 4 interface. On the left, the Database Navigator pane displays the connection to 'postgres' on 'localhost:5432' and the schema 'public'. It lists various tables such as categories, customers, products, and order_details. In the center, a script editor window titled 'Script-3' contains the following SQL query:

```

select c.contact_name
  from customers c
 where c.fax is null

```

Below the script, a results grid titled 'customers 1' shows the following data:

contact_name
Antonio Moreno
Victoria Ashworth
Yang Wang
Pedro Afonso
Aria Cruz
Maria Larsson
José Pedro Freyre
André Fonseca
Howard Snyder
Helen Bennett
Philip Cramer
Jaime Yorres
Alexander Feuer
Isabel de Castro

The bottom status bar indicates '22 row(s) fetched - 2ms'.

(15)

The screenshot shows the pgAdmin 4 interface. On the left, the Database Navigator displays the Northwind database schema, including tables like categories, customers, and employees. In the center, a script editor window titled 'Script-3' contains a SQL query to create a view named 'employee_age'. Below the script is a results grid titled 'region 1' showing the average age by region:

region_id	region_description	avg
1	Eastern	71 years 2 mons 34 days 05:03:09.491874
2	Northern	58 years 8 mons 33 days 10:54:32.711564
3	Western	60 years 4 mons 46 days 16:00:00
4	Southern	58 years 2 mons 27 days

The screenshot shows the pgAdmin 4 interface. On the left, the Database Navigator displays the Northwind database schema. In the center, a properties dialog for the 'employee_age' view is open. The 'Data' tab is selected, showing the view's name, object ID, owner, and extra options. The 'Columns' tab lists the columns: 'employee_id' (int2) and 'age' (interval). The 'Dependencies' tab lists triggers, rules, statistics, permissions, source, and virtual dependencies.