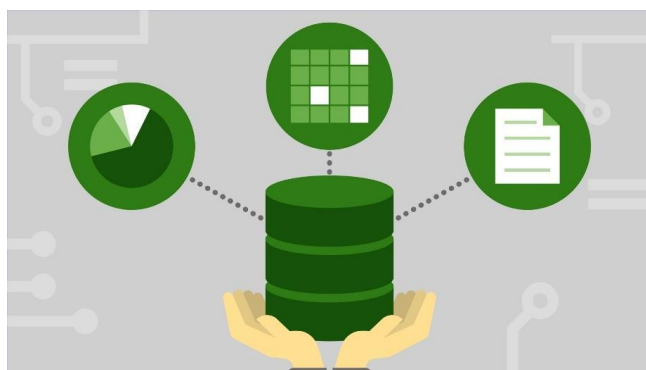


به نام خدا



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



آزمایشگاه پایگاه داده

دستور کار شماره پنج (دستورات پایه SQL)

امیرعلی رایگان

۸۱۰۱۹۷۶۲۳

بهار ۱۴۰۱

گزارش دستورکار انجام شده

آشنایی با دستورات پایه SQL

سوال ۱. با استفاده از یک inner join ساده به جواب رسیدم.

The screenshot shows the DBeaver 22.0.0 interface. The SQL Editor contains the following query:

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

The Results pane displays the following data:

| region_id | region_description | territory_id | 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 | |

سوال ۲.

در بین تمام کالا ها آنهایی که discounted برابر ۱ داشتند را انتخاب و تعداد کالا های هر دسته را نمایش دادم.

The screenshot shows the DBeaver 22.0.0 interface. The SQL Editor contains the following query:

```
-- Question 2
select c.category_name, count(*)
from categories c inner join products p on p.category_id = c.category_id
where p.discontinued = 1
group by c.category_name
order by count(*) desc
```

The Results pane displays the following data:

| category_name | count |
|----------------|-------|
| Meat/Poultry | 4 |
| Beverages | 3 |
| Produce | 1 |
| Condiments | 1 |
| Grains/Cereals | 1 |

سوال ۳.

ابتدا جدول `order_details` را با `order` ها ترکیب کردم و در نهایت قیمت کلی هر سفارش را نمایش دادم. در قسمت بعد این سوال هم ۱۰ سفارش گران مشخص شده است.

Database Navigator: postgres - localhost:5432

SQL Editor:

```
--Question 3
select o.order_id, sum( od.unit_price * od.quantity * (1-od.discount) ) as "cost"
from orders o, order_details od
where od.order_id = o.order_id
group by o.order_id
order by o.order_id
-----
select o.customer_id, sum( od.unit_price * od.quantity * (od.discount) )
from orders o, order_details od
where od.order_id = o.order_id
group by o.customer_id
order by sum( od.unit_price * od.quantity * (od.discount) ) desc
limit 10
```

Results (orders 1 x):

| order_id | cost |
|----------|------------------|
| 10.248 | 439.999980927 |
| 10.249 | 1.863.4000644684 |
| 10.250 | 1.552.6000234127 |
| 10.251 | 654.05985579 |
| 10.252 | 3.597.9001445159 |
| 10.253 | 1.444.799839783 |
| 10.254 | 556.6200096774 |
| 10.255 | 2.490.4999780655 |
| 10.256 | 517.8000068665 |
| 10.257 | 1.119.8999538422 |
| 10.258 | 1.614.8799881858 |
| 10.259 | 100.7999992371 |

Rows: 1200 row(s) fetched - 24ms (8ms fetch), on Apr 28, 14

Database Navigator: postgres - localhost:5432

SQL Editor:

```
--QUESTION 3
select o.order_id, sum( od.unit_price * od.quantity * (1-od.discount) ) as "cost"
from orders o, order_details od
where od.order_id = o.order_id
group by o.order_id
order by o.order_id
-----
select o.customer_id, sum( od.unit_price * od.quantity * (od.discount) )
from orders o, order_details od
where od.order_id = o.order_id
group by o.customer_id
order by sum( od.unit_price * od.quantity * (od.discount) ) desc
limit 10
```

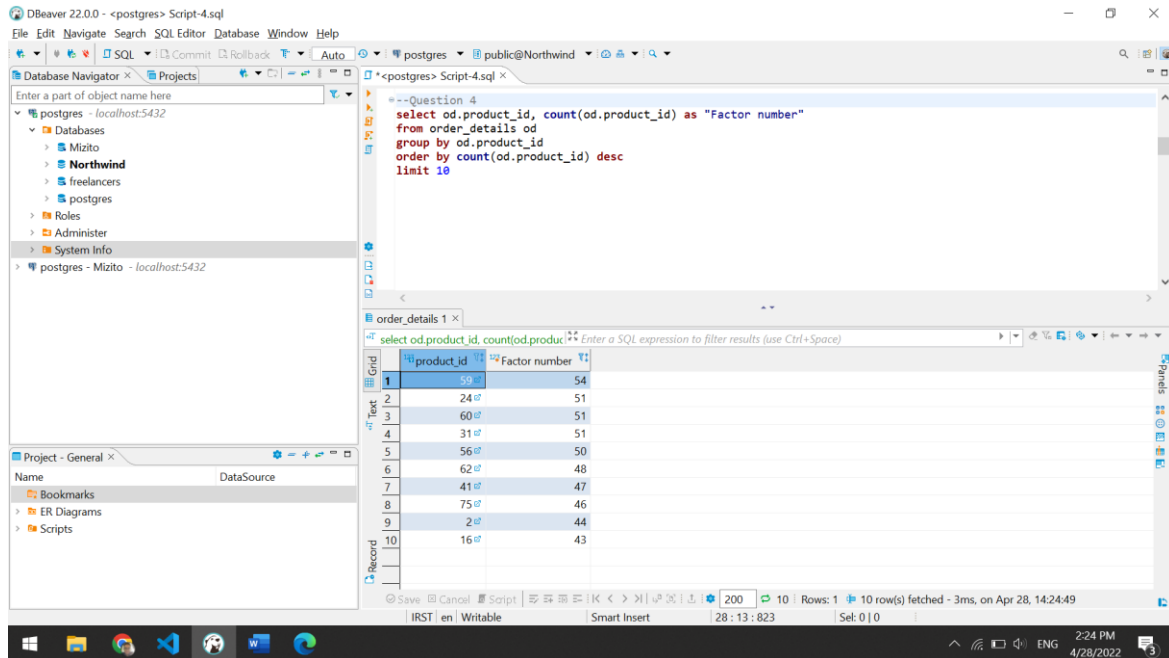
Results (orders 1 x):

| customer_id | sum |
|-------------|-------------------|
| SAVEFA | 11.311.4401023215 |
| ERNSH | 8.361.7016382363 |
| HUNGO | 7.337.4850809725 |
| QUICK | 7.206.0851172922 |
| QUEEN | 4.508.6025746987 |
| MEREP | 3.331.710078104 |
| PICCO | 3.131.0900697125 |
| FOLKO | 2.987.9875292376 |
| FRANK | 2.066.150535013 |
| BERGS | 2.040.5724995112 |

Rows: 10 row(s) fetched - 9ms, on Apr 28, 14:24:37

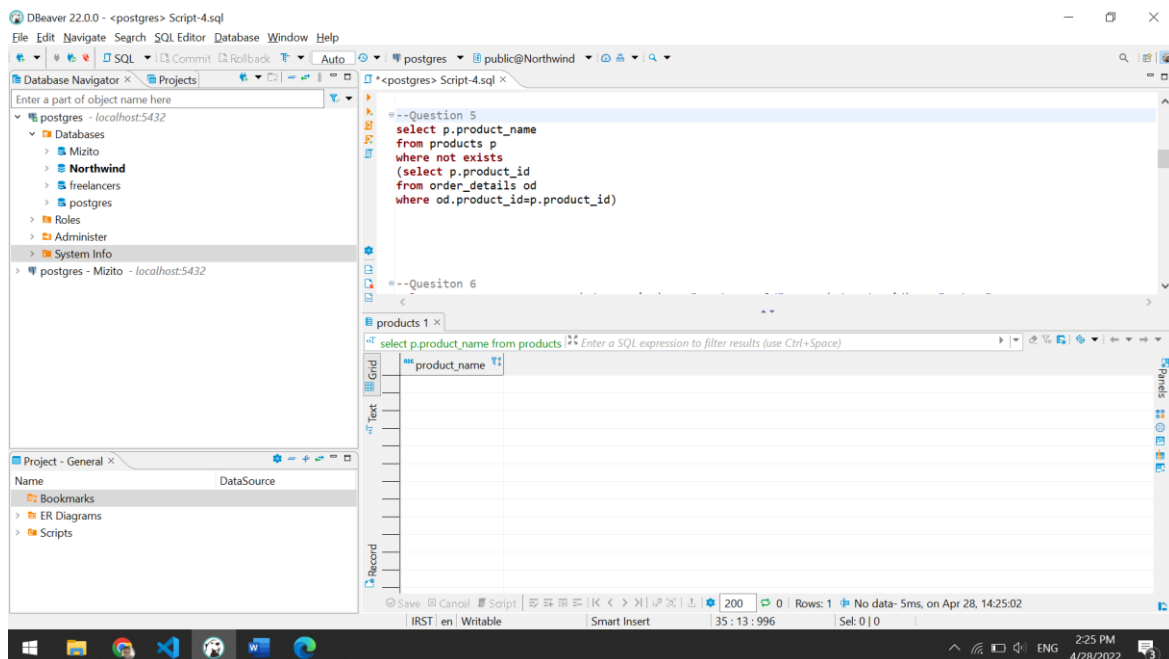
سوال ۴.

با استفاده از دستورات اولیه بیشترین فاکتورها نمایش داده شده اند.



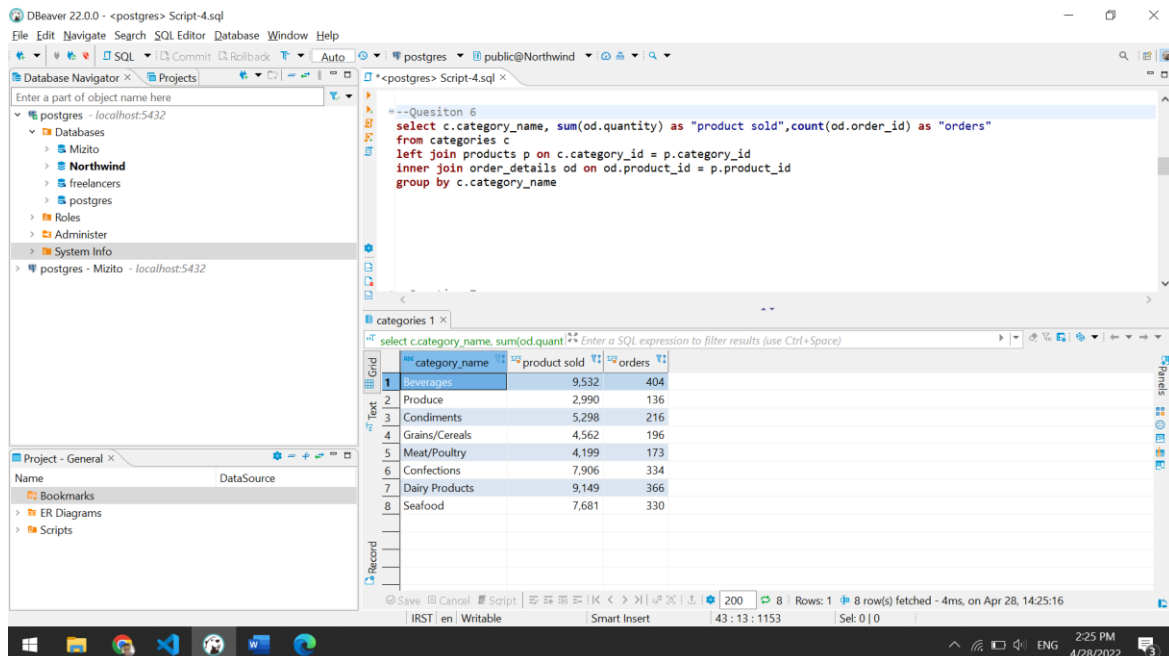
سوال ۵.

در بین محصولات تمام محصولاتی که با هیچ سفارشی در ارتباط نیستند مشخص شده است.



سوال ۶

ابتدا یک `left join` بین دسته ها و محصولات زدم (چون ممکن بود یکی یا چند تا از دسته ها دارای هیچ محصولی نباشند) و سپس با یک `inner join` با جدول `order_details` پاسخ سوال نمایش داده شد.



سوال ۷

در این سوال یک جدول موقت ساختم تا مجموع قیمت سفارش ها را محاسبه کند. سپس با ترکیب جدول کارکنان و سفارش ها و همچنین انتخاب سفارش هایی که در سال 1996 ثبت شده اند توانستم مجموع درآمدی که از فروشندگی هر کدام از کارکنان به دست آمده محاسبه کنم و سپس بالاترین آنها را با دستور `order by` و `limit` پیدا کنم.

The screenshot shows the DBeaver 22.0.0 interface. The left sidebar displays the database structure for 'postgres' on 'localhost:5432', including databases like 'Mizito' and 'Northwind'. The main editor shows a SQL script with a temporary table and a query to find the employee with the highest total income in 1996.

```

--Question 7
WITH temporaryTable as
(
SELECT od.order_id as id, sum( (od.unit_price*od.quantity)*(1-od.discount) ) as total_sale
from order_details od
group by od.order_id
)
select e.employee_id, e.first_name, e.last_name, sum(tt.total_sale) as total_income
FROM employees e inner join orders o on o.employee_id = e.employee_id
inner join temporaryTable tt on o.order_id = tt.id
where date_part('year', o.order_date) = 1996 and o.order_id in (select order_id from temporaryTable)
group by e.employee_id
order by sum(tt.total_sale) desc
limit 1

```

The results pane shows a single row for Margaret Peacock with a total income of 49,945.1153284936.

| employee_id | first_name | last_name | total_income |
|-------------|------------|-----------|-------------------|
| 1 | Margaret | Peacock | 49,945.1153284936 |

سوال ۸

در این سوال برای دادن table به هر زمان ارسال از دستور case استفاده کردیم.

The screenshot shows the DBeaver 22.0.0 interface. The left sidebar displays the database structure for 'postgres' on 'localhost:5432'. The main editor shows a SQL script that creates a table 'lable' based on the 'shipped_date' and 'order_date' from the 'orders' table.

```

--Question 8
SELECT o.order_id,
CASE
WHEN o.shipped_date-o.order_date = 0 THEN 'Fantastic'
WHEN o.shipped_date-o.order_date < 3 THEN 'Good'
ELSE 'Not good'
END AS lable
FROM orders o;

```

The results pane shows a list of orders with their corresponding 'lable' values.

| order_id | lable |
|----------|----------|
| 10,248 | Not good |
| 10,249 | Not good |
| 10,250 | Not good |
| 10,251 | Not good |
| 10,252 | Good |
| 10,253 | Not good |
| 10,254 | Not good |
| 10,255 | Not good |
| 10,256 | Good |
| 10,257 | Not good |
| 10,258 | Not good |
| 10,259 | Not good |

سوال ۹.

در این سوال با دستور **with recursive** به پیدا کردن سلسه مراتب کارکنان پرداختیم. به این شکل که ستون **reports_to** که در جدول کارکنان وجود دارد مشخص می کند هر کسی به کی باید گزارش دهد. به همین ترتیب اول کسی را انتخاب کردیم که نباید به کسی گزارش دهد. سپس افرادی که باید به او گزارش دهند و به همین منوال تمام افراد باید به چه کسی گزارش دهند. در انتها جدولی که مشخص کننده سلسه مراتب است قابل نمایش است.

The screenshot displays the DBeaver SQL Editor with a recursive query for PostgreSQL. The query is as follows:

```
--Question 9
with recursive company_hierarchy as (
    select ee.employee_id,
           ee.first_name,
           ee.last_name,
           ee.reports_to,
           0 as hierarchy_level
    from employees ee
    where ee.reports_to is null

    union all

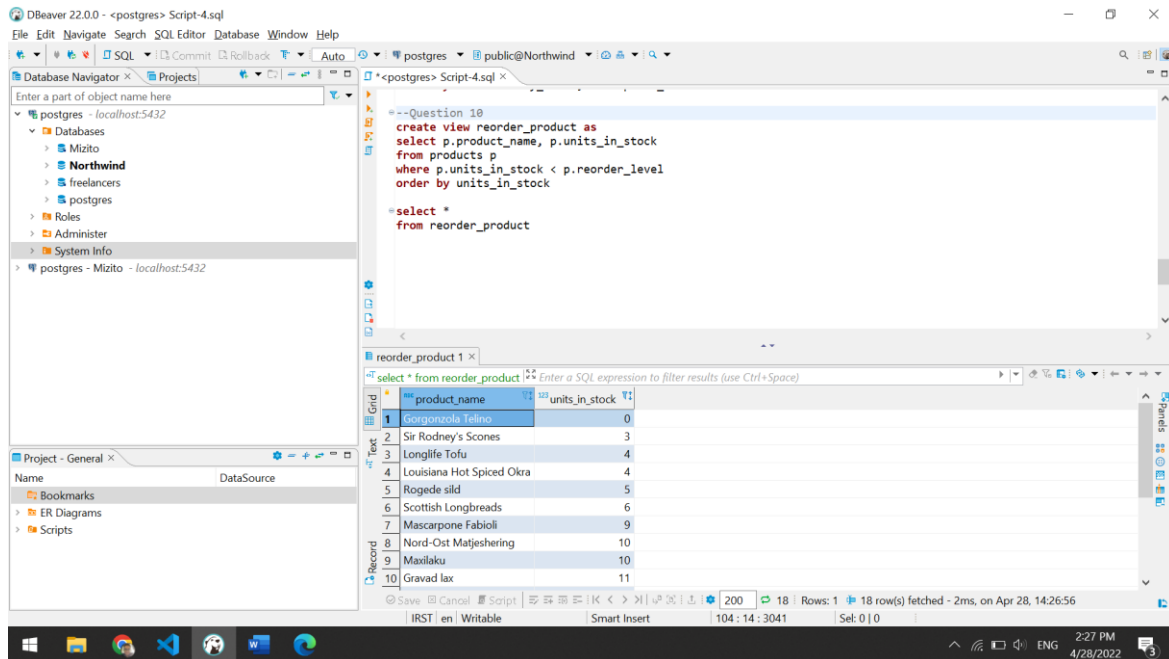
    select e.employee_id,
           e.first_name,
           e.last_name,
           e.reports_to,
           hierarchy_level + 1
    from employees e, company_hierarchy ch
    where e.reports_to = ch.employee_id
)
select ch.employee_id as employee_id,
       ch.first_name AS employee_first_name,
       ch.last_name AS employee_last_name,
       e.first_name AS boss_first_name,
       e.last_name AS boss_last_name,
       hierarchy_level
from company_hierarchy ch, employees e
where ch.employee_id = e.employee_id
order by ch.hierarchy_level, ch.reports_to
```

The results of the query are shown in a table with 9 rows:

| employee_id | employee_first_name | employee_last_name | boss_first_name | boss_last_name | hierarchy_level |
|-------------|---------------------|--------------------|-----------------|----------------|-----------------|
| 1 | Andrew | Fuller | Andrew | Fuller | 0 |
| 2 | Nancy | Davolio | Nancy | Davolio | 1 |
| 3 | Janet | Leverling | Janet | Leverling | 1 |
| 4 | Margaret | Peacock | Margaret | Peacock | 1 |
| 5 | Steven | Buchanan | Steven | Buchanan | 1 |
| 6 | Laura | Callahan | Laura | Callahan | 1 |
| 7 | Michael | Suyama | Michael | Suyama | 2 |
| 8 | Robert | King | Robert | King | 2 |
| 9 | Anne | Dodsworth | Anne | Dodsworth | 2 |

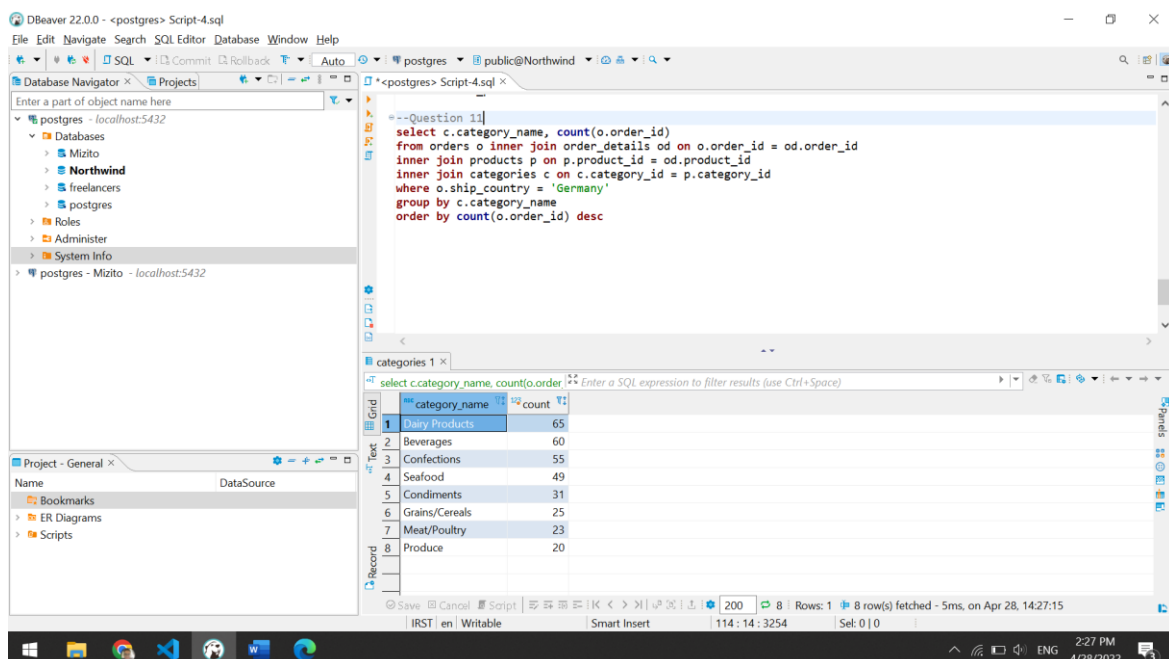
سوال ۱۰.

در این سوال ابتدا کوئری مورد نیاز را زدیم و سپس با دستور `create view` که قبل از است یک ویو برای خودمان ساختیم.



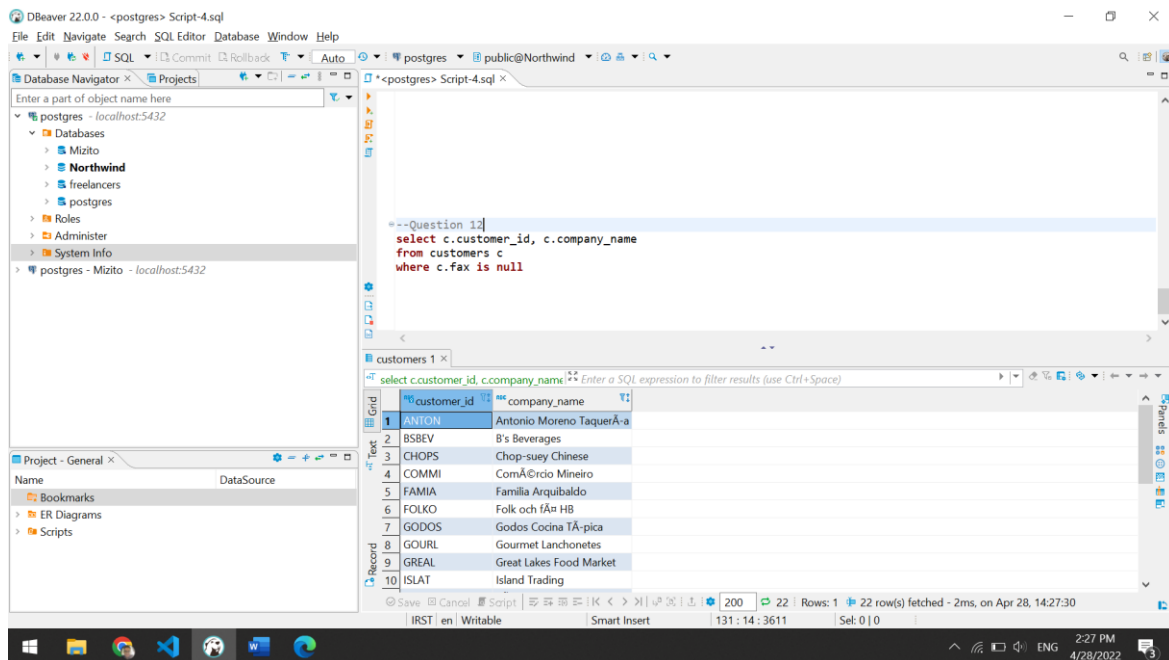
سوال ۱۱.

به وسیله دستور `join` چهار جدول `order`, `order_details`, `products`, `category` را به هم متصل کردیم و جواب را چاپ کردیم.



سوال ۱۲.

خیلی ساده.



مشکلات و توضیحات تکمیلی

مشکلی به وجود نیامد.

آنچه آموختم / پیشنهادات

با تست دستورات اولیه sql روی یک دیتابیس واقعی ، با این زبان بیشتر آشنا شدم.