Database Advance Job sheet 2

From:

AL AZHAR RIZQI RIFA'I FIRDAUS

Class:

2 I

Absence:

01

Student Number Identity:

2241720263

Department:

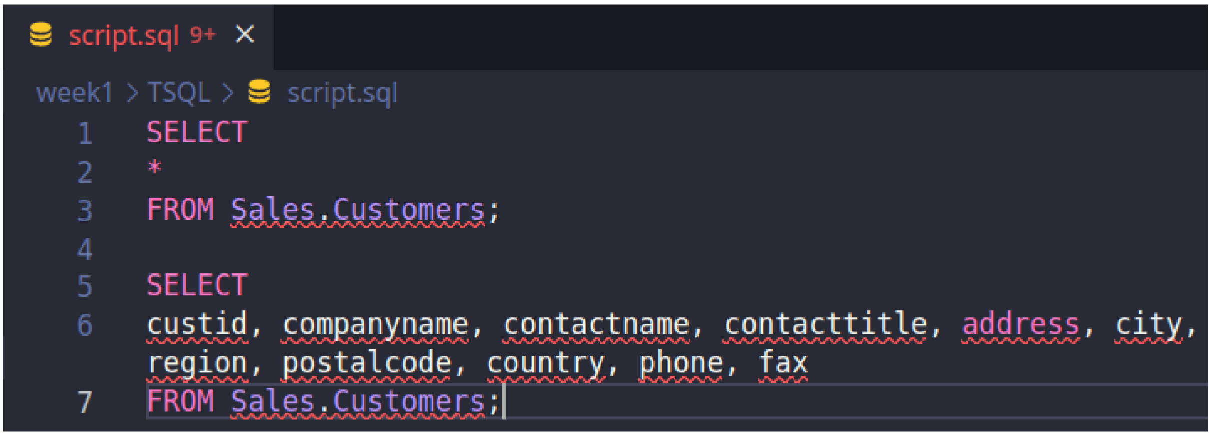
Information Technology

Study Program:

Informatics Engineering

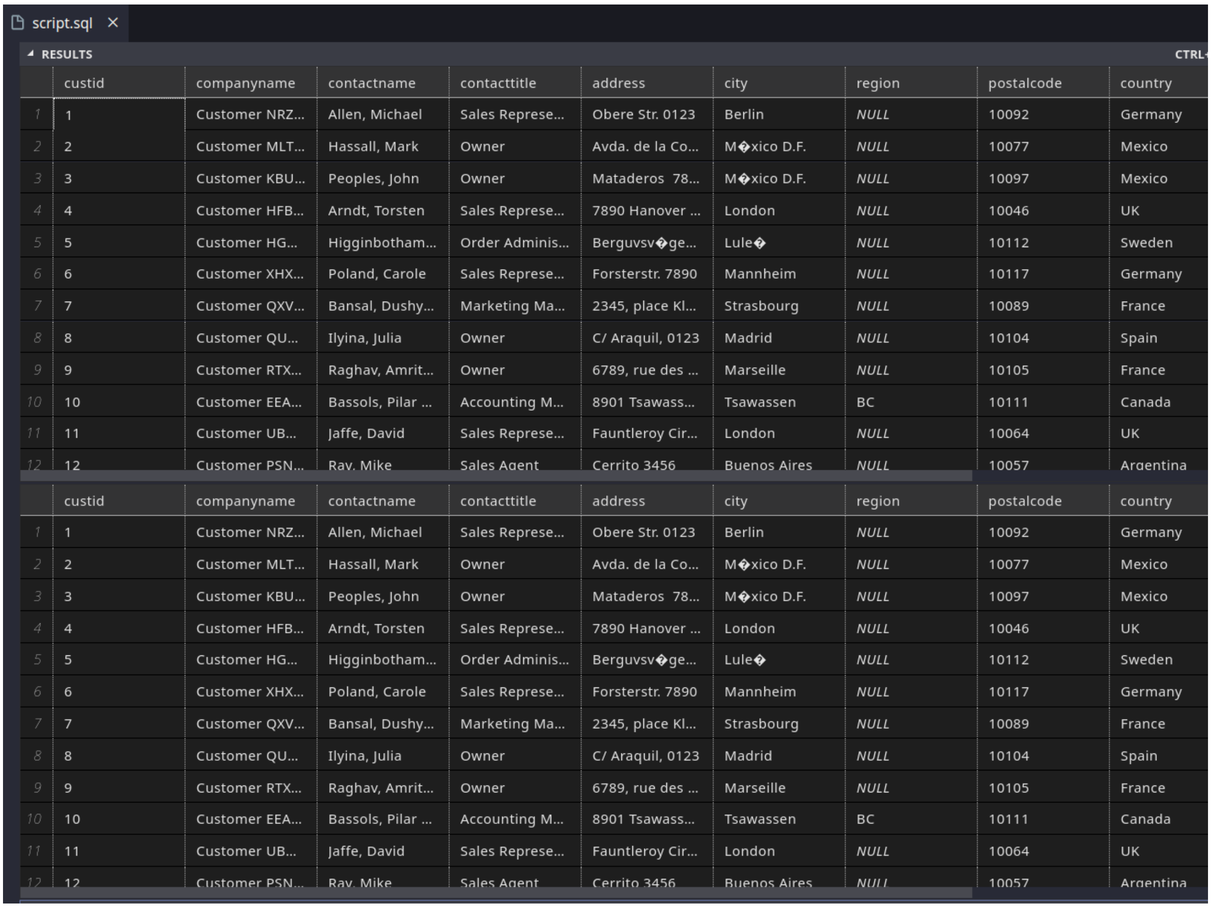
Practicum 1

Code:



```sql
SELECT
*
FROM Sales.Customers;

SELECT
custid, companyname, contactname, contacttitle, address, city,
region, postalcode, country, phone, fax
FROM Sales.Customers;
```

Result:



	custid	companyname	contactname	contacttitle	address	city	region	postalcode	country
1	1	Customer NRZ...	Allen, Michael	Sales Represe...	Obere Str. 0123	Berlin	NULL	10092	Germany
2	2	Customer MLT...	Hassall, Mark	Owner	Avda. de la Co...	M�xico D.F.	NULL	10077	Mexico
3	3	Customer KBU...	Peoples, John	Owner	Mataderos 78...	M�xico D.F.	NULL	10097	Mexico
4	4	Customer HFB...	Arndt, Torsten	Sales Represe...	7890 Hanover ...	London	NULL	10046	UK
5	5	Customer HG...	Higginbotham...	Order Adminis...	Berguvsv�ge...	Lule�	NULL	10112	Sweden
6	6	Customer XHX...	Poland, Carole	Sales Represe...	Forsterstr. 7890	Mannheim	NULL	10117	Germany
7	7	Customer QXV...	Bansal, Dushy...	Marketing Ma...	2345, place Kl...	Strasbourg	NULL	10089	France
8	8	Customer QU...	Ilyina, Julia	Owner	C/ Araquil, 0123	Madrid	NULL	10104	Spain
9	9	Customer RTX...	Raghav, Amrit...	Owner	6789, rue des ...	Marseille	NULL	10105	France
10	10	Customer EEA...	Bassols, Pilar ...	Accounting M...	8901 Tsawass...	Tsawassen	BC	10111	Canada
11	11	Customer UB...	Jaffe, David	Sales Represe...	Fauntleroy Cir...	London	NULL	10064	UK
12	12	Customer PSN...	Ray, Mike	Sales Agent	Cerrito 3456	Buenos Aires	NULL	10057	Argentina

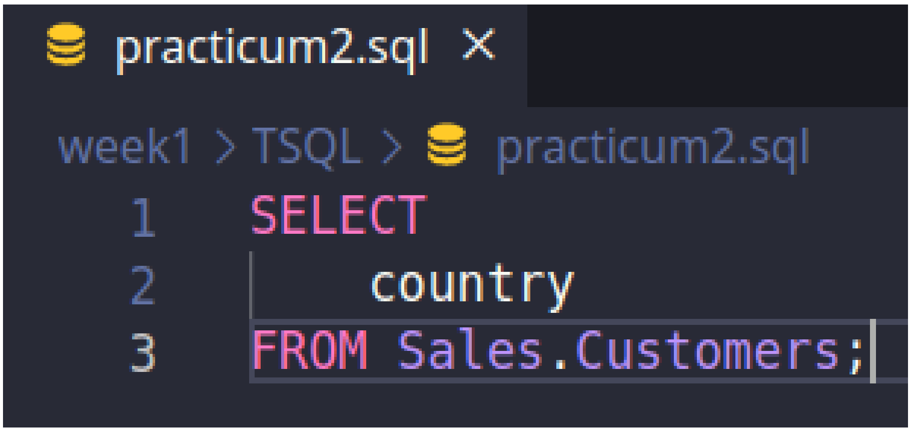
	custid	companyname	contactname	contacttitle	address	city	region	postalcode	country
1	1	Customer NRZ...	Allen, Michael	Sales Represe...	Obere Str. 0123	Berlin	NULL	10092	Germany
2	2	Customer MLT...	Hassall, Mark	Owner	Avda. de la Co...	M�xico D.F.	NULL	10077	Mexico
3	3	Customer KBU...	Peoples, John	Owner	Mataderos 78...	M�xico D.F.	NULL	10097	Mexico
4	4	Customer HFB...	Arndt, Torsten	Sales Represe...	7890 Hanover ...	London	NULL	10046	UK
5	5	Customer HG...	Higginbotham...	Order Adminis...	Berguvsv�ge...	Lule�	NULL	10112	Sweden
6	6	Customer XHX...	Poland, Carole	Sales Represe...	Forsterstr. 7890	Mannheim	NULL	10117	Germany
7	7	Customer QXV...	Bansal, Dushy...	Marketing Ma...	2345, place Kl...	Strasbourg	NULL	10089	France
8	8	Customer QU...	Ilyina, Julia	Owner	C/ Araquil, 0123	Madrid	NULL	10104	Spain
9	9	Customer RTX...	Raghav, Amrit...	Owner	6789, rue des ...	Marseille	NULL	10105	France
10	10	Customer EEA...	Bassols, Pilar ...	Accounting M...	8901 Tsawass...	Tsawassen	BC	10111	Canada
11	11	Customer UB...	Jaffe, David	Sales Represe...	Fauntleroy Cir...	London	NULL	10064	UK
12	12	Customer PSN...	Ray, Mike	Sales Agent	Cerrito 3456	Buenos Aires	NULL	10057	Argentina

1. How is this different from the result in the second step above?

- **The different is the first query use * which is wildcard that is used to select all table, then the second query use table name to select table that we want it.**

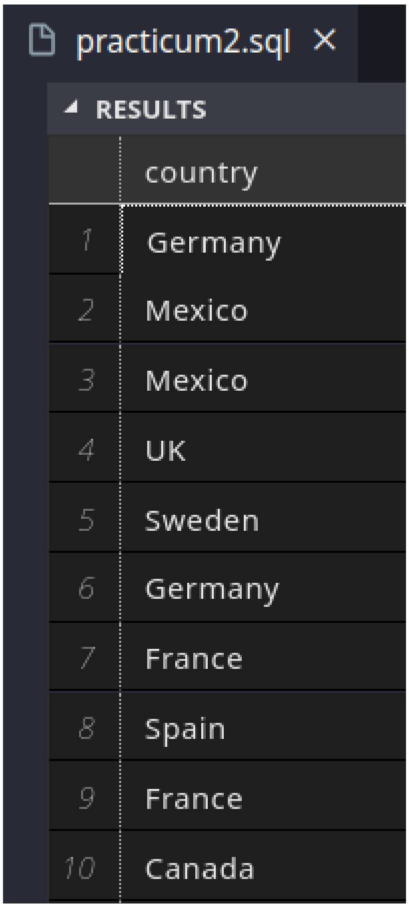
Practicum 2

Code:



```sql
SELECT
    country
FROM Sales.Customers;
```

Result:

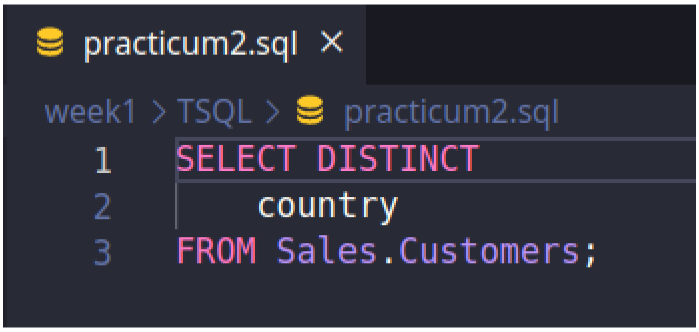


	country
1	Germany
2	Mexico
3	Mexico
4	UK
5	Sweden
6	Germany
7	France
8	Spain
9	France
10	Canada

2. Is there any duplicated data? If YES why? Capture the execution result of the SQL script above!

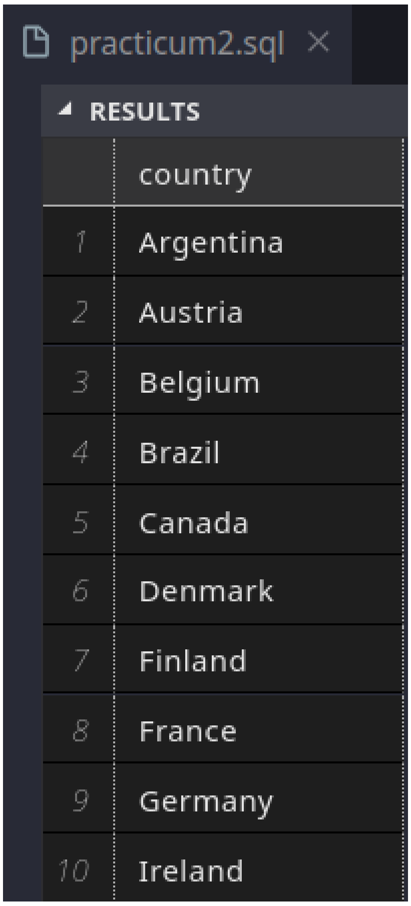
- **Yes, data country can be duplicated because at country, there is no unique identifier.**

Code:



```sql
SELECT DISTINCT
    country
FROM Sales.Customers;
```

Result:



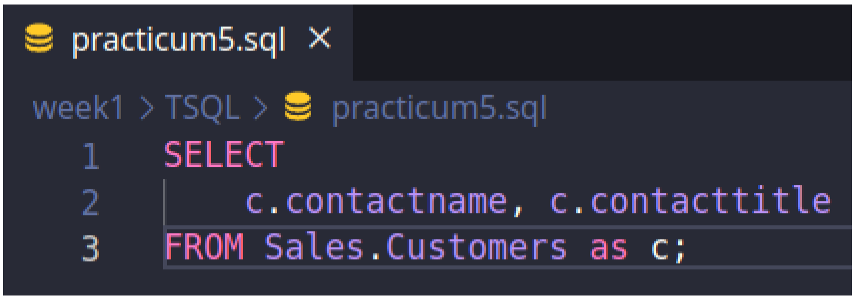
	country
1	Argentina
2	Austria
3	Belgium
4	Brazil
5	Canada
6	Denmark
7	Finland
8	France
9	Germany
10	Ireland

3. Is there any duplicated data? Explain the difference between the results of stage 4 and stage 3! What are the benefits of the DISTINCT command? Capture the execution result of the SQL script above!

- **No. The difference between both is at stage 3 there is duplicated data because it doesn't use distinct. Then, at stage 4 there is no duplicated data because it use distinct. The function of distinct is to return unique data. If there are duplicated data, when we use distinct, it'll return only one data that duplicated.**

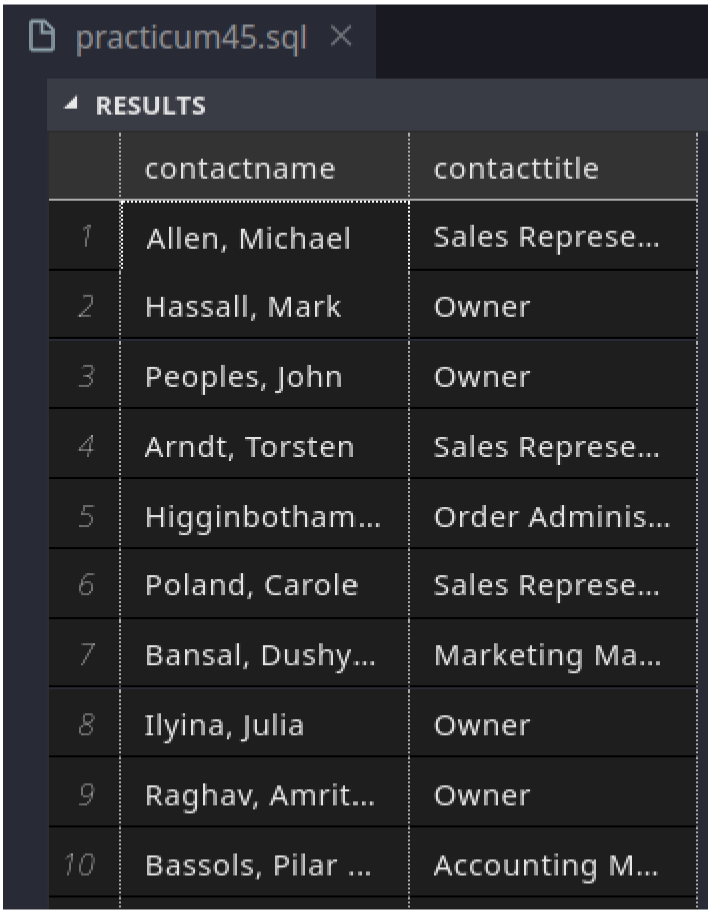
Practicum 3

Code:



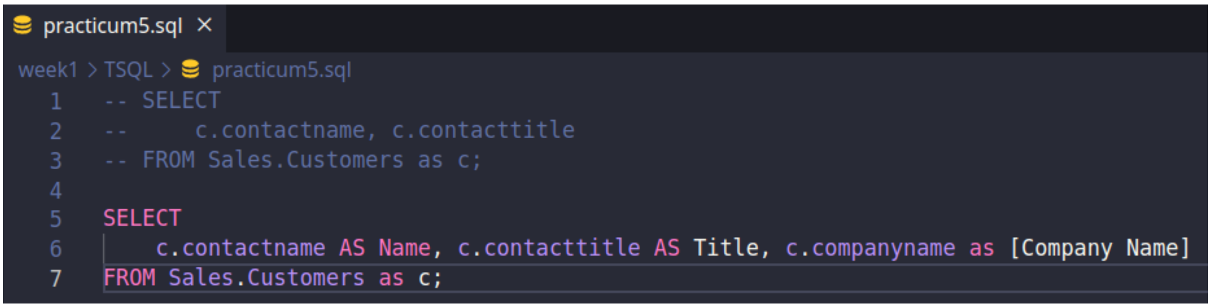
```sql
SELECT
    c.contactname, c.contacttitle
FROM Sales.Customers as c;
```

Result:



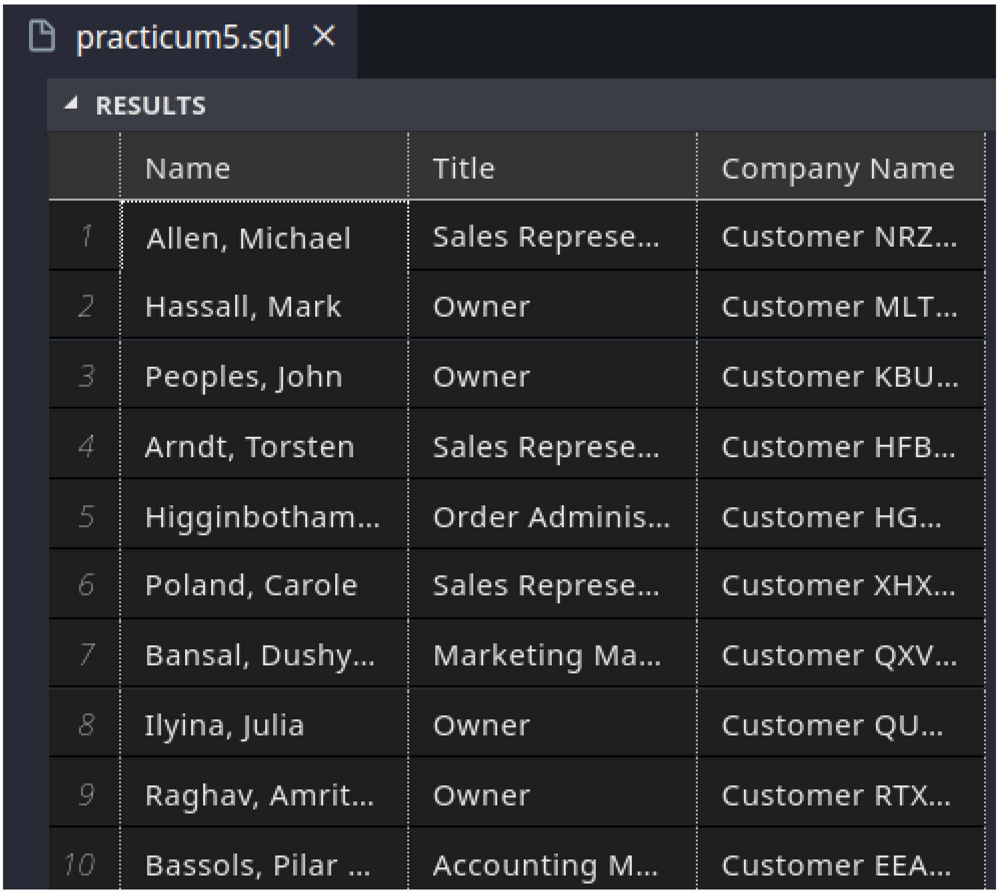
	contactname	contacttitle
1	Allen, Michael	Sales Represe...
2	Hassall, Mark	Owner
3	Peoples, John	Owner
4	Arndt, Torsten	Sales Represe...
5	Higginbotham...	Order Adminis...
6	Poland, Carole	Sales Represe...
7	Bansal, Dushy...	Marketing Ma...
8	Ilyina, Julia	Owner
9	Raghav, Amrit...	Owner
10	Bassols, Pilar ...	Accounting M...

Code:



```sql
-- SELECT
--     c.contactname, c.contacttitle
-- FROM Sales.Customers as c;

SELECT
    c.contactname AS Name, c.contacttitle AS Title, c.companyname as [Company Name]
FROM Sales.Customers as c;
```

Result:



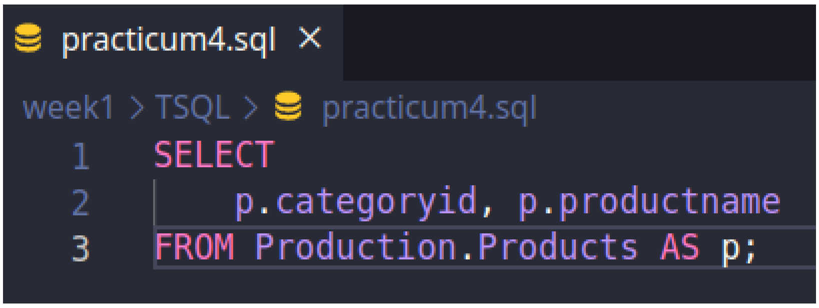
	Name	Title	Company Name
1	Allen, Michael	Sales Represe...	Customer NRZ...
2	Hassall, Mark	Owner	Customer MLT...
3	Peoples, John	Owner	Customer KBU...
4	Arndt, Torsten	Sales Represe...	Customer HFB...
5	Higginbotham...	Order Adminis...	Customer HG...
6	Poland, Carole	Sales Represe...	Customer XHX...
7	Bansal, Dushy...	Marketing Ma...	Customer QXV...
8	Ilyina, Julia	Owner	Customer QU...
9	Raghav, Amrit...	Owner	Customer RTX...
10	Bassols, Pilar ...	Accounting M...	Customer EEA...

4. What is the difference between the execution results of stage 1 and stage 3 queries above? What are the benefits of the AS command? Please explain! Capture the execution result of the SQL script above

- **The difference between both is the stage 1 use AS(ALIAS) at Sales.Customers table. It useful to modify the table name that we want it. Then the stage 3 use AS(ALIAS) at Sales.Customers table and column that we write it. It useful to modify the table name and the column name and display the column name as we want.**

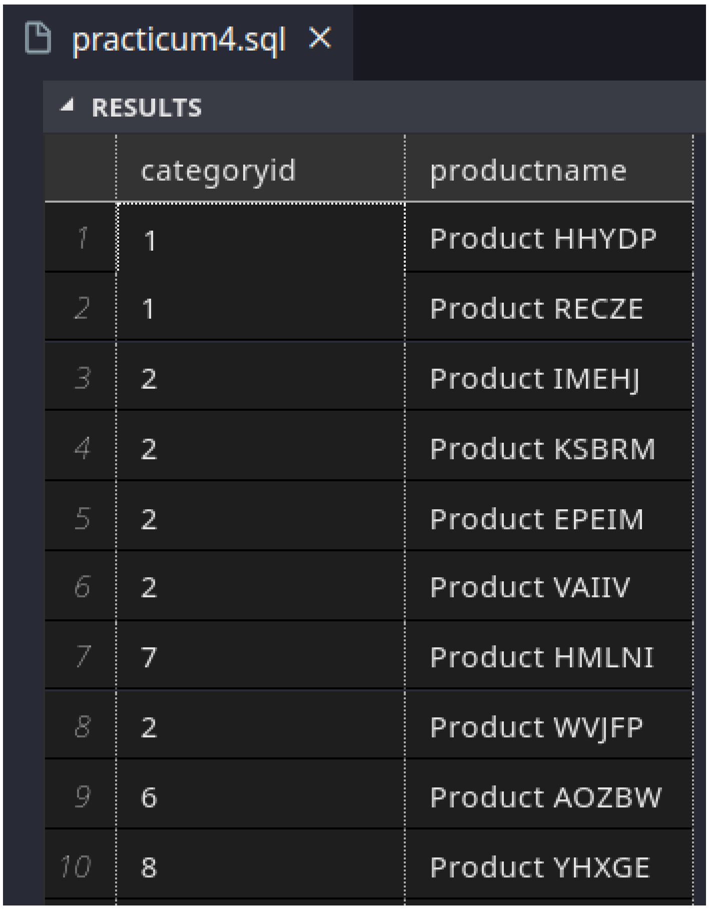
Practicum 4

Code:



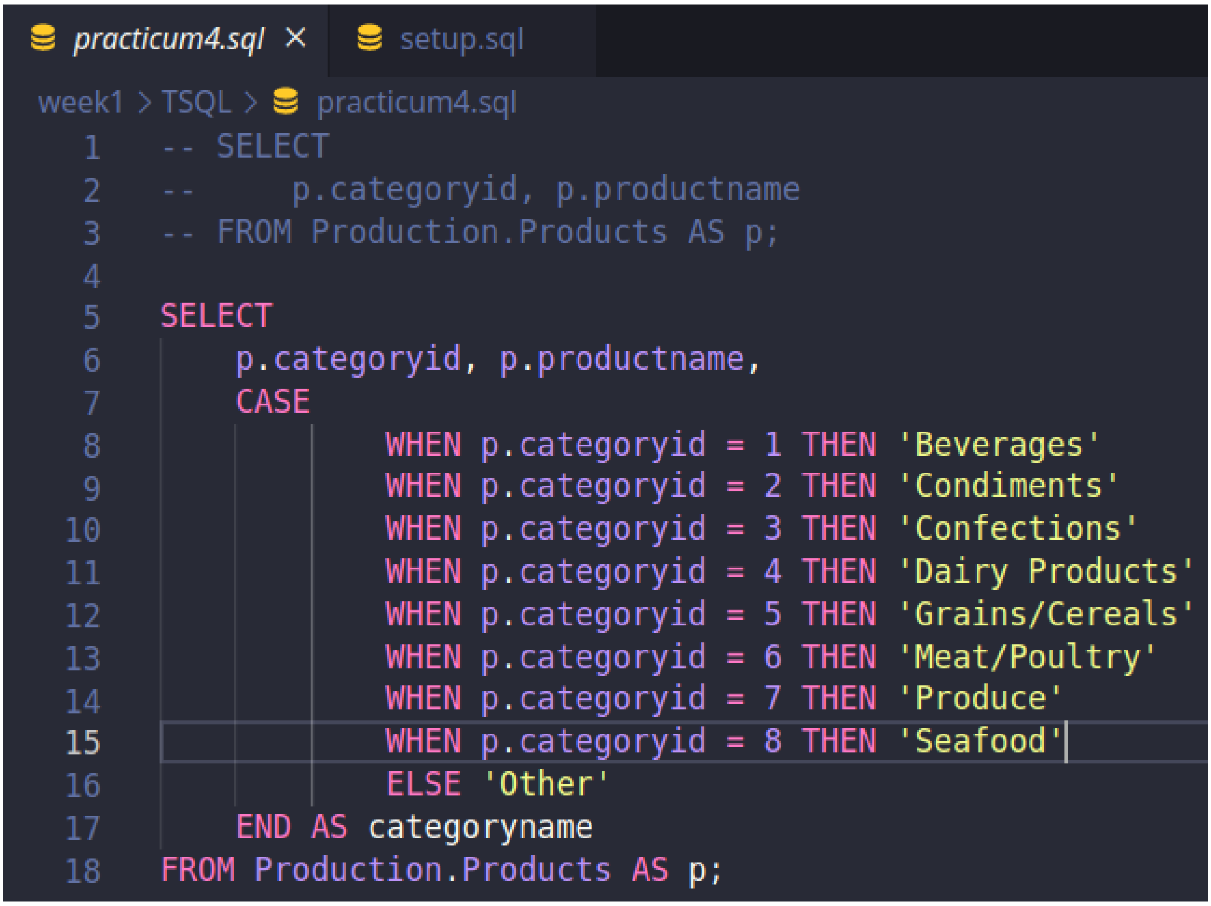
```
SELECT
    p.categoryid, p.productname
FROM Production.Products AS p;
```

Result:



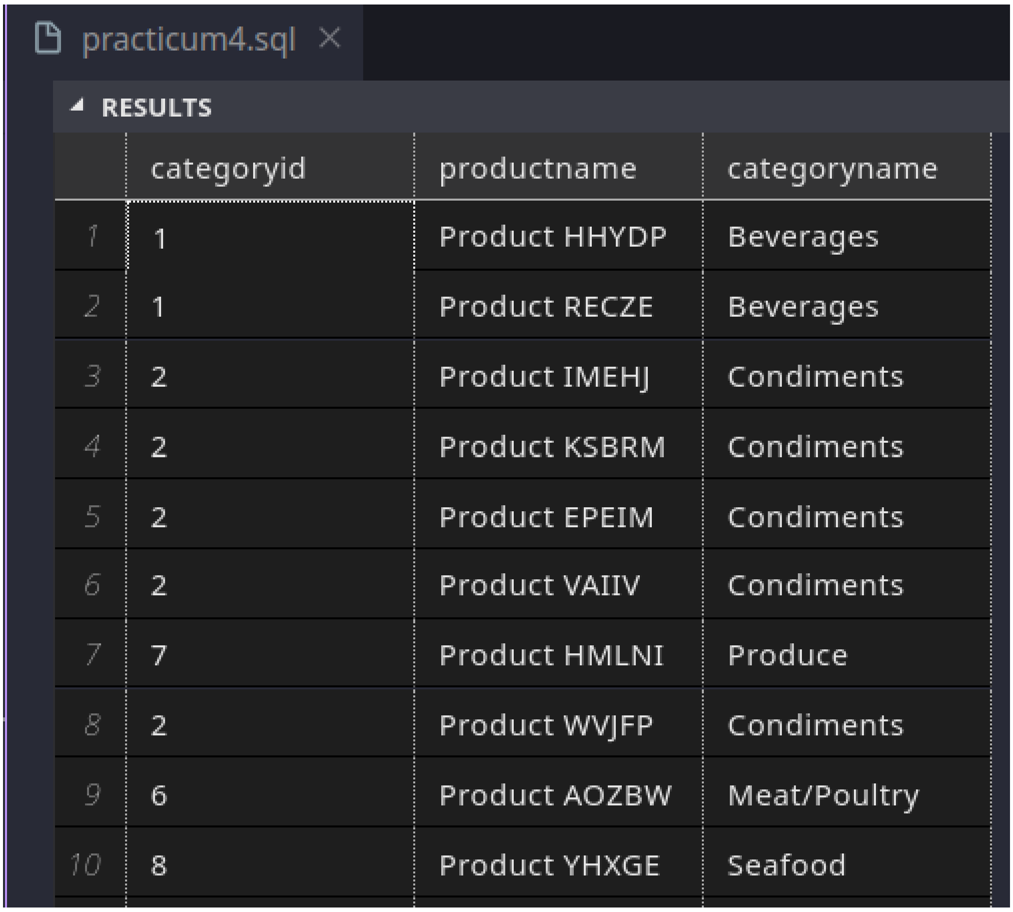
	categoryid	productname
1	1	Product HHYDP
2	1	Product RECZE
3	2	Product IMEHJ
4	2	Product KSBRM
5	2	Product EPEIM
6	2	Product VAIIV
7	7	Product HMLNI
8	2	Product WVJFP
9	6	Product AOZBW
10	8	Product YHXGE

Code:



```sql
-- SELECT
--     p.categoryid, p.productname
-- FROM Production.Products AS p;

SELECT
    p.categoryid, p.productname,
    CASE
            WHEN p.categoryid = 1 THEN 'Beverages'
            WHEN p.categoryid = 2 THEN 'Condiments'
            WHEN p.categoryid = 3 THEN 'Confections'
            WHEN p.categoryid = 4 THEN 'Dairy Products'
            WHEN p.categoryid = 5 THEN 'Grains/Cereals'
            WHEN p.categoryid = 6 THEN 'Meat/Poultry'
            WHEN p.categoryid = 7 THEN 'Produce'
            WHEN p.categoryid = 8 THEN 'Seafood'
            ELSE 'Other'
    END AS categoryname
FROM Production.Products AS p;
```

Result:

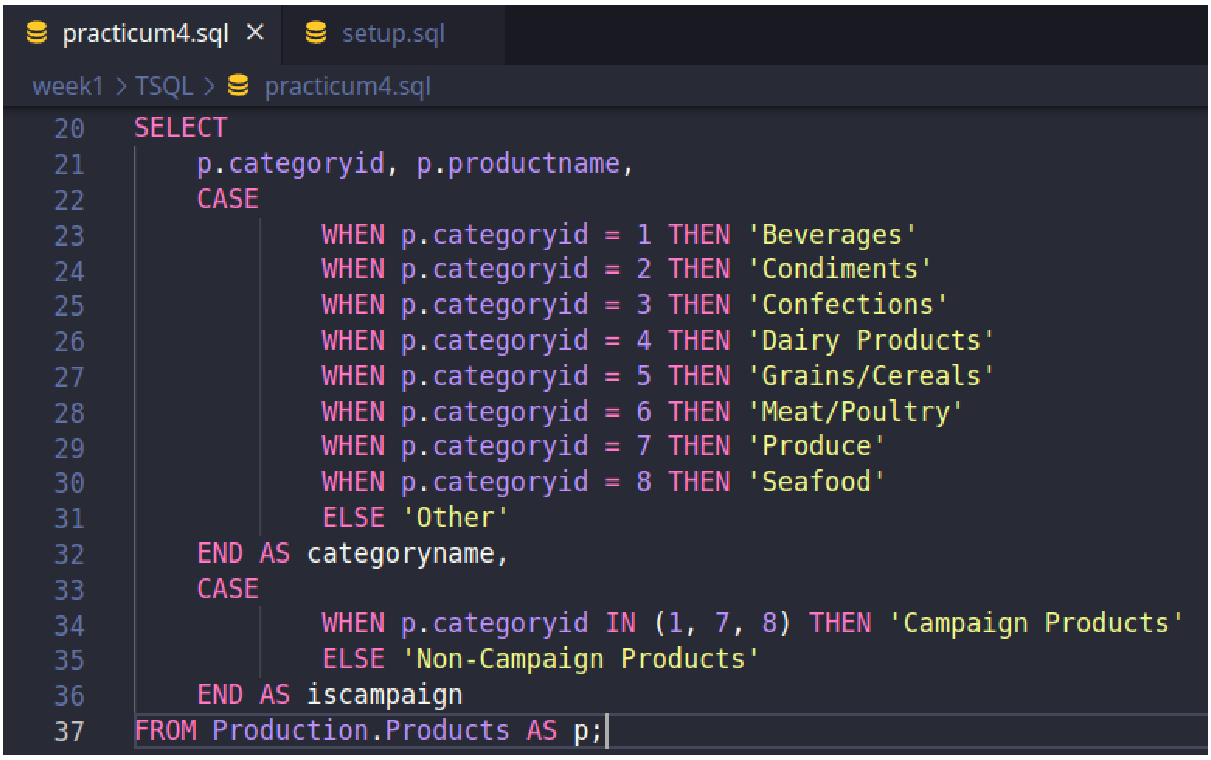


	categoryid	productname	categoryname
1	1	Product HHYDP	Beverages
2	1	Product RECZE	Beverages
3	2	Product IMEHJ	Condiments
4	2	Product KSBRM	Condiments
5	2	Product EPEIM	Condiments
6	2	Product VAIIV	Condiments
7	7	Product HMLNI	Produce
8	2	Product WVJFP	Condiments
9	6	Product AOZBW	Meat/Poultry
10	8	Product YHXGE	Seafood

5. What is the difference between the execution results of the stage 1 and stage 3 queries above? What are the benefits of the CASE command? Please explain! Capture the execution result of the SQL script above!

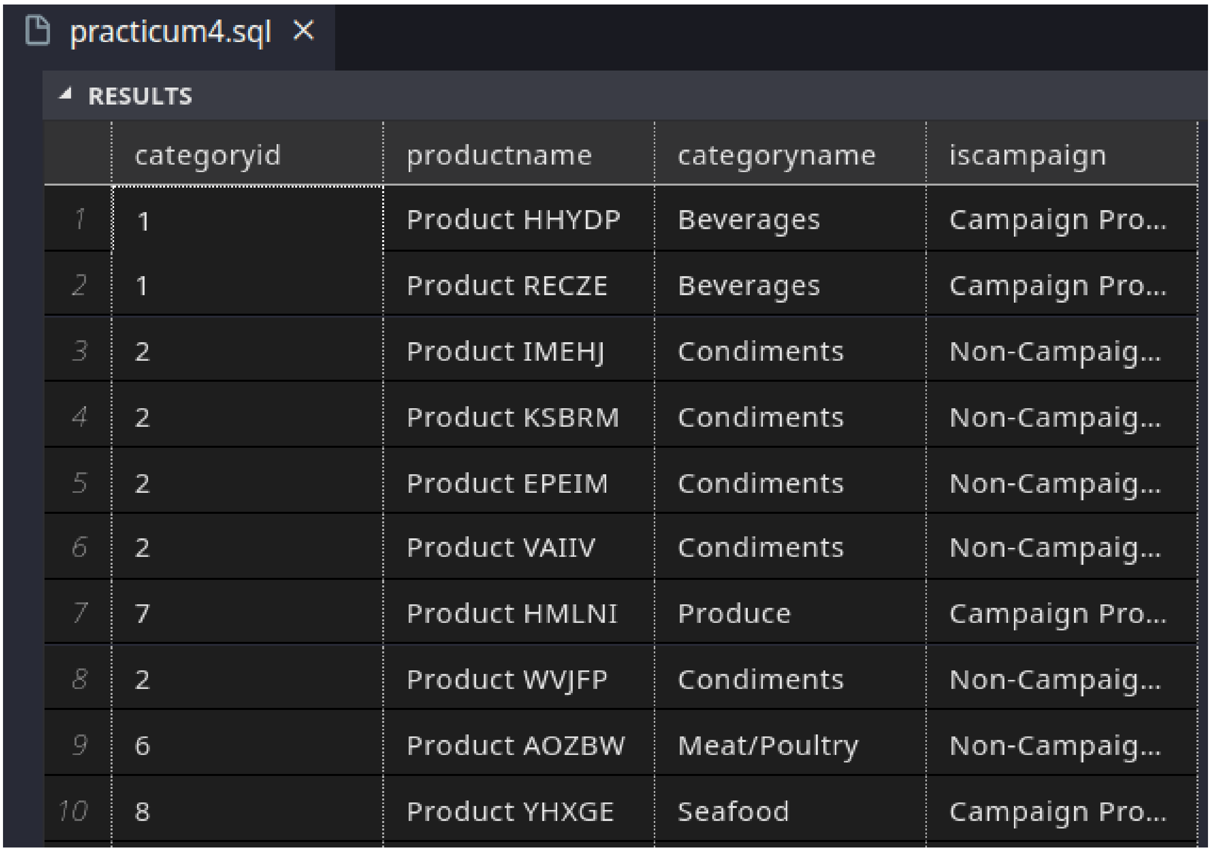
- **The difference between both is the stage 1 doing select query to get categoryid and productname column, then the stage 2 doing select uery to get categoryid and productname column, but there is addition that is categoryname column that is added by case condition that doing condition to add new value based on value categoryid.**

Code:



```sql
SELECT
    p.categoryid, p.productname,
    CASE
            WHEN p.categoryid = 1 THEN 'Beverages'
            WHEN p.categoryid = 2 THEN 'Condiments'
            WHEN p.categoryid = 3 THEN 'Confections'
            WHEN p.categoryid = 4 THEN 'Dairy Products'
            WHEN p.categoryid = 5 THEN 'Grains/Cereals'
            WHEN p.categoryid = 6 THEN 'Meat/Poultry'
            WHEN p.categoryid = 7 THEN 'Produce'
            WHEN p.categoryid = 8 THEN 'Seafood'
            ELSE 'Other'
    END AS categoryname,
    CASE
            WHEN p.categoryid IN (1, 7, 8) THEN 'Campaign Products'
            ELSE 'Non-Campaign Products'
    END AS iscampaign
FROM Production.Products AS p;
```

Result:



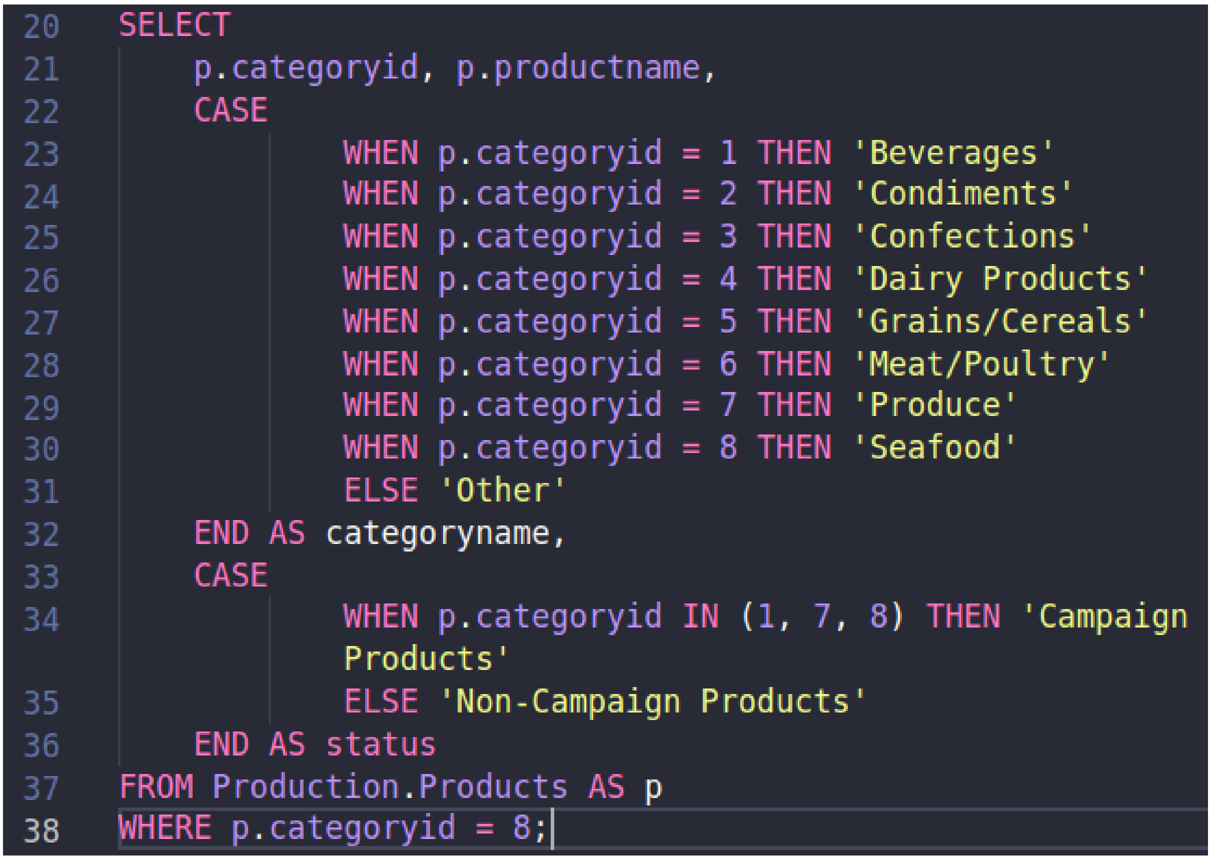
	categoryid	productname	categoryname	iscampaign
1	1	Product HHYDP	Beverages	Campaign Pro...
2	1	Product RECZE	Beverages	Campaign Pro...
3	2	Product IMEHJ	Condiments	Non-Campaig...
4	2	Product KSBRM	Condiments	Non-Campaig...
5	2	Product EPEIM	Condiments	Non-Campaig...
6	2	Product VAIIV	Condiments	Non-Campaig...
7	7	Product HMLNI	Produce	Campaign Pro...
8	2	Product WVJFP	Condiments	Non-Campaig...
9	6	Product AOZBW	Meat/Poultry	Non-Campaig...
10	8	Product YHXGE	Seafood	Campaign Pro...

6. Please capture the results, what data is obtained from the query command above? Explain!

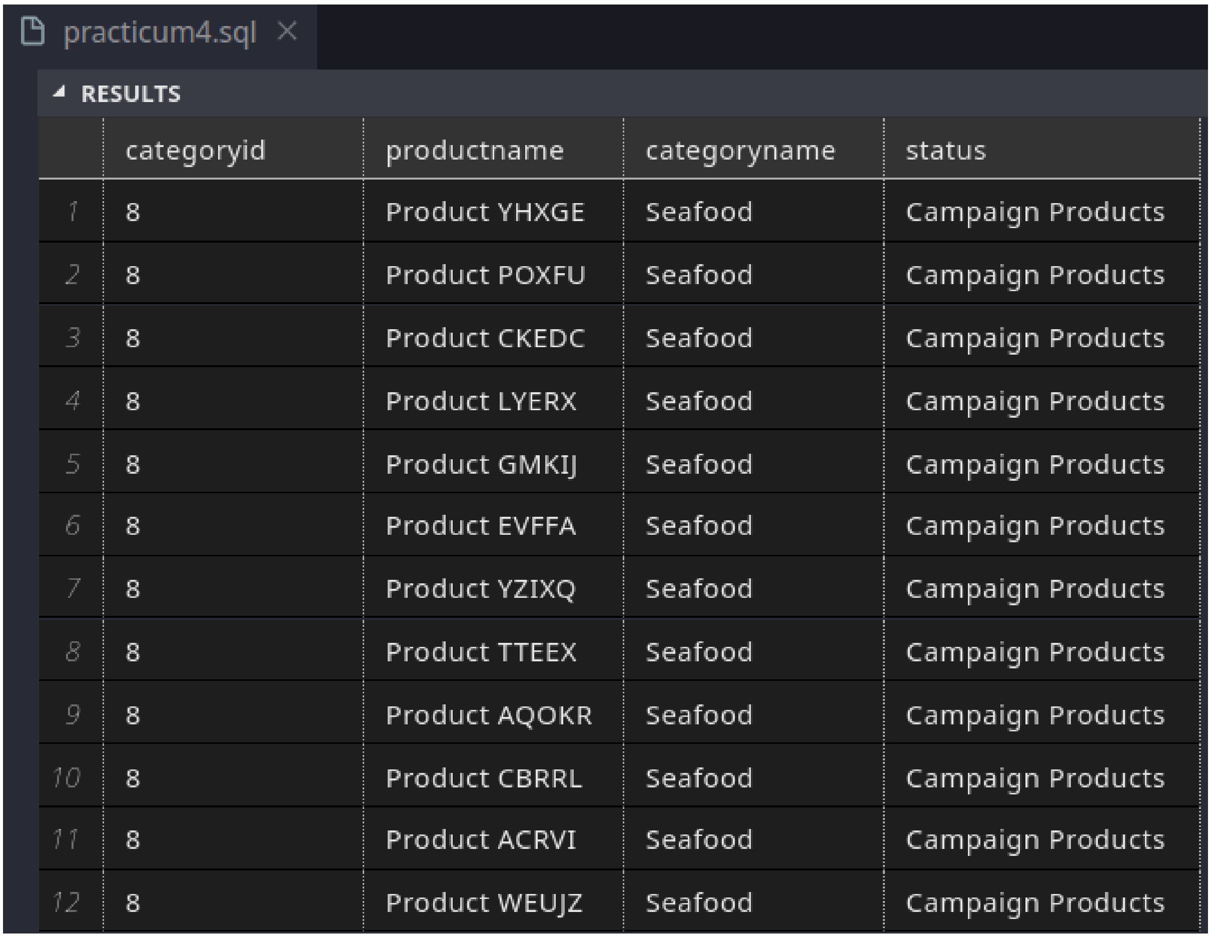
- **According query above, there is query that doing select column that same with before, but there is addition column that namely iscampaign that using CASE condition. The different**

with CASE before, this CASE use IN to combine categoryid value become one, so that we don’t repeat to write a query categoryid one by one. Then if the condition is true, iscampaign will be added ‘campaign products’, then else it’ll be added ‘non-campaign products’.

7. Based on question number 6, please display the data in the 'seafood' category only and use the ALIAS command to change the column name as shown below. Capture your SQL command and how many rows are generated!



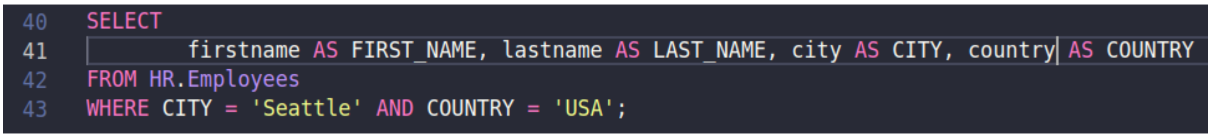
```sql
SELECT
    p.categoryid, p.productname,
    CASE
            WHEN p.categoryid = 1 THEN 'Beverages'
            WHEN p.categoryid = 2 THEN 'Condiments'
            WHEN p.categoryid = 3 THEN 'Confections'
            WHEN p.categoryid = 4 THEN 'Dairy Products'
            WHEN p.categoryid = 5 THEN 'Grains/Cereals'
            WHEN p.categoryid = 6 THEN 'Meat/Poultry'
            WHEN p.categoryid = 7 THEN 'Produce'
            WHEN p.categoryid = 8 THEN 'Seafood'
            ELSE 'Other'
    END AS categoryname,
    CASE
            WHEN p.categoryid IN (1, 7, 8) THEN 'Campaign Products'
            ELSE 'Non-Campaign Products'
    END AS status
FROM Production.Products AS p
WHERE p.categoryid = 8;
```



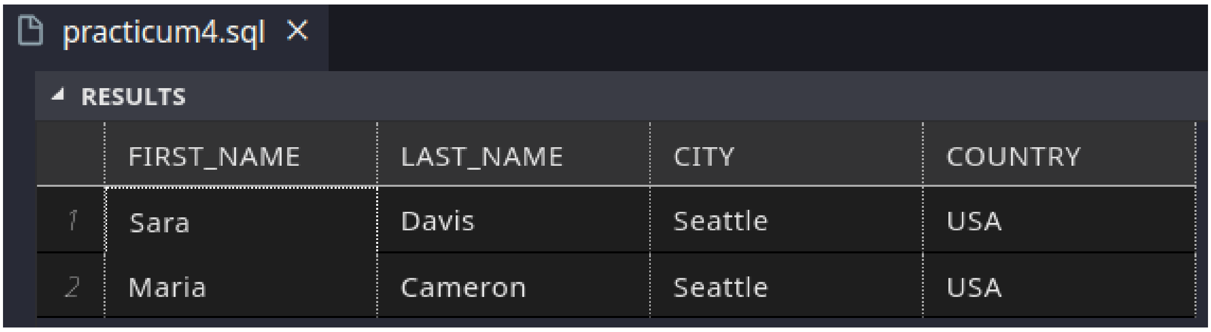
	categoryid	productname	categoryname	status
1	8	Product YHXGE	Seafood	Campaign Products
2	8	Product POXFU	Seafood	Campaign Products
3	8	Product CKEDC	Seafood	Campaign Products
4	8	Product LYERX	Seafood	Campaign Products
5	8	Product GMKIJ	Seafood	Campaign Products
6	8	Product EVFFA	Seafood	Campaign Products
7	8	Product YZIXQ	Seafood	Campaign Products
8	8	Product TTEEX	Seafood	Campaign Products
9	8	Product AQOKR	Seafood	Campaign Products
10	8	Product CBRRL	Seafood	Campaign Products
11	8	Product ACRVI	Seafood	Campaign Products
12	8	Product WEUJZ	Seafood	Campaign Products

- **There are 12 rows that generated.**

8. Display employees data from the HR.Employees table that are from the country 'USA' and the city 'Seattle', use the ALIAS command to rename the columns as shown below. Capture your SQL command!



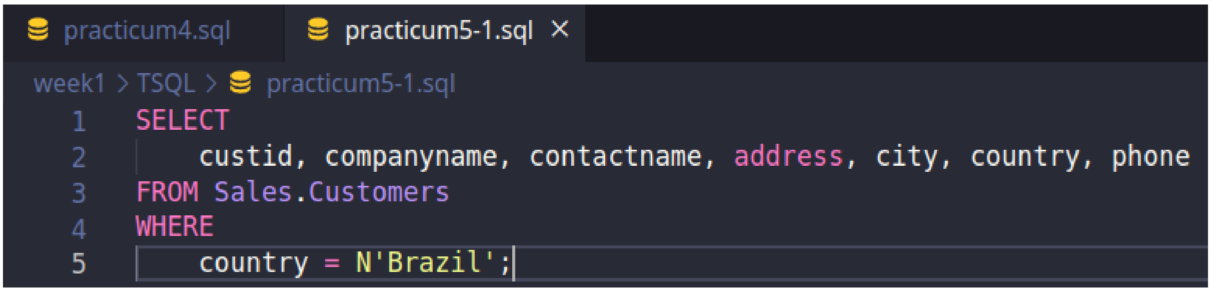
```
SELECT
        firstname AS FIRST_NAME, lastname AS LAST_NAME, city AS CITY, country AS COUNTRY
FROM HR.Employees
WHERE CITY = 'Seattle' AND COUNTRY = 'USA';
```



	FIRST_NAME	LAST_NAME	CITY	COUNTRY
1	Sara	Davis	Seattle	USA
2	Maria	Cameron	Seattle	USA

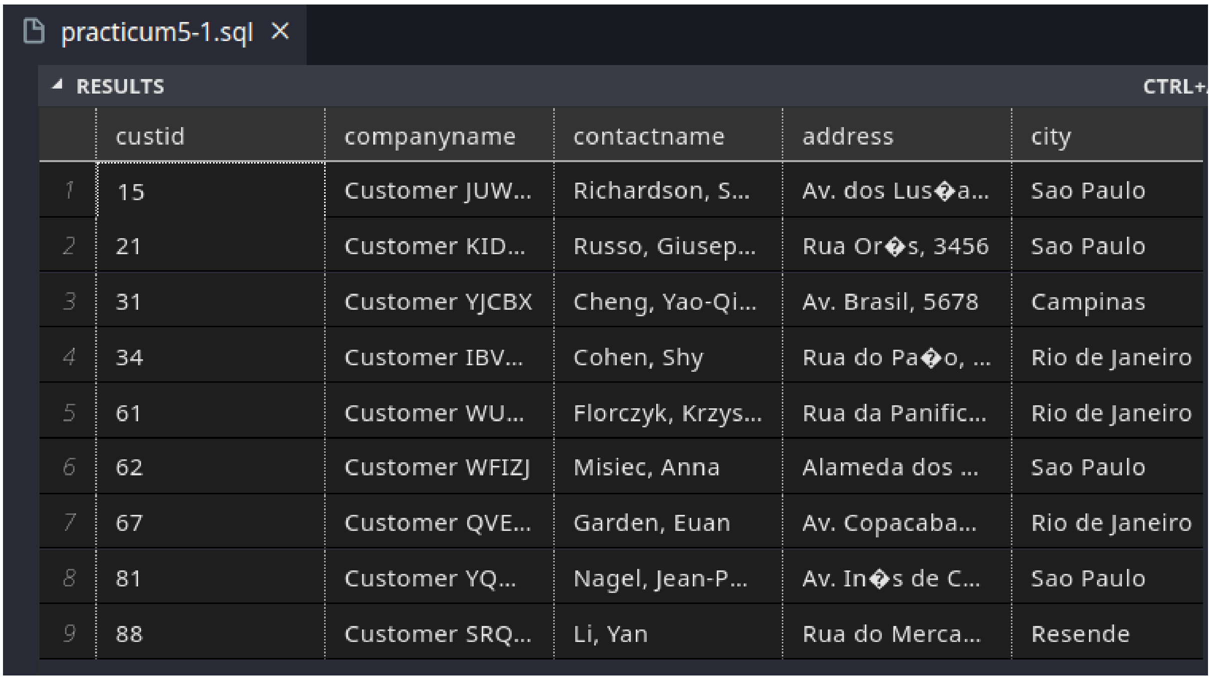
Practicum 5.1

Code:



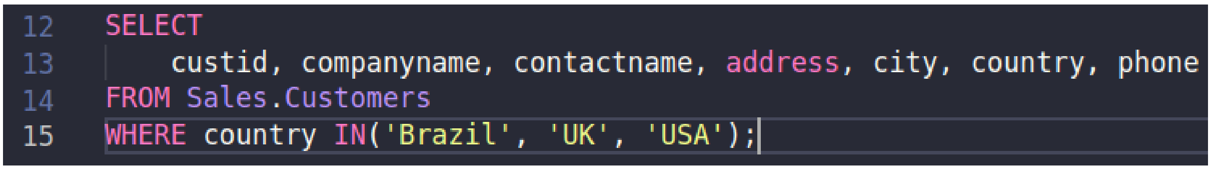
```sql
SELECT
    custid, companyname, contactname, address, city, country, phone
FROM Sales.Customers
WHERE
    country = N'Brazil';
```

Result:

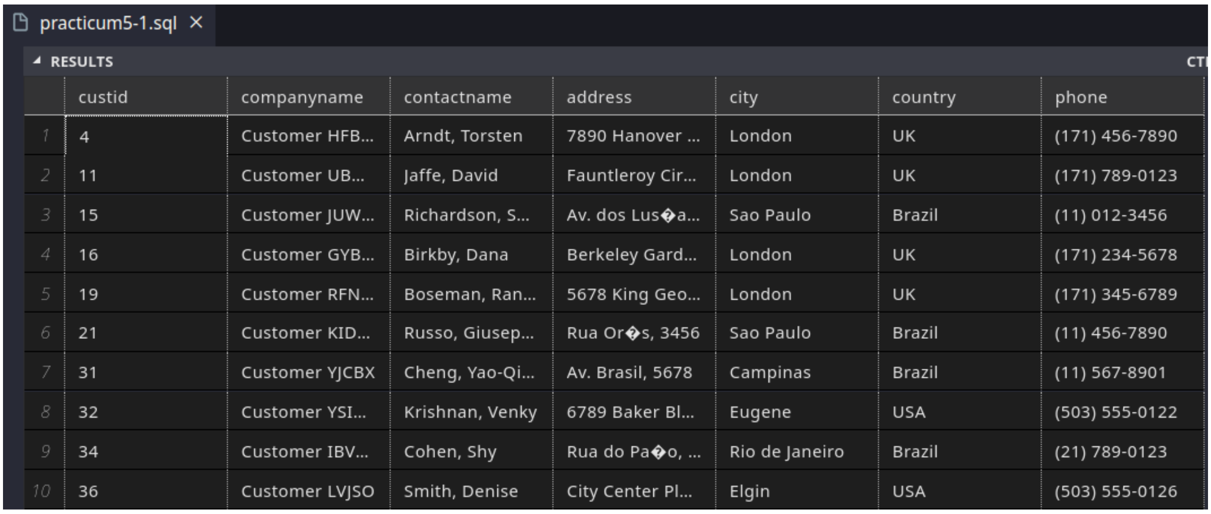


	custid	companyname	contactname	address	city
1	15	Customer JUW...	Richardson, S...	Av. dos Lus�a...	Sao Paulo
2	21	Customer KID...	Russo, Giusep...	Rua Or�s, 3456	Sao Paulo
3	31	Customer YJCBX	Cheng, Yao-Qi...	Av. Brasil, 5678	Campinas
4	34	Customer IBV...	Cohen, Shy	Rua do Pa�o, ...	Rio de Janeiro
5	61	Customer WU...	Florczyk, Krzys...	Rua da Panific...	Rio de Janeiro
6	62	Customer WFIZJ	Misiec, Anna	Alameda dos ...	Sao Paulo
7	67	Customer QVE...	Garden, Euan	Av. Copacaba...	Rio de Janeiro
8	81	Customer YQ...	Nagel, Jean-P...	Av. In�s de C...	Sao Paulo
9	88	Customer SRQ...	Li, Yan	Rua do Merca...	Resende

9. Write a SELECT command that will return values in the custid, companyname, contactname, address, city, country, and phone columns in the Sales.Customers table, then filter the results to only for "Brazil, UK and USA" (Use the IN predicate in the WHERE clause).

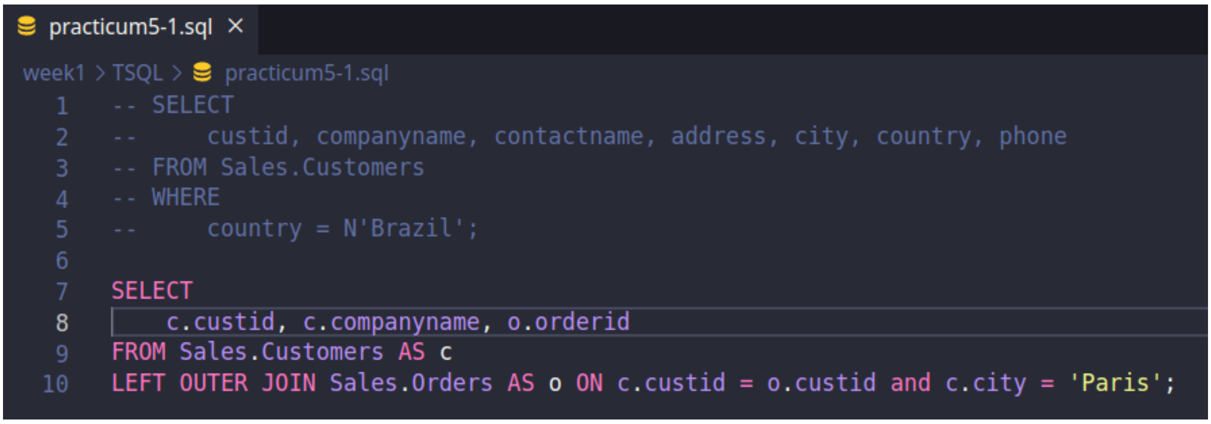


```sql
SELECT
    custid, companyname, contactname, address, city, country, phone
FROM Sales.Customers
WHERE country IN('Brazil', 'UK', 'USA');
```



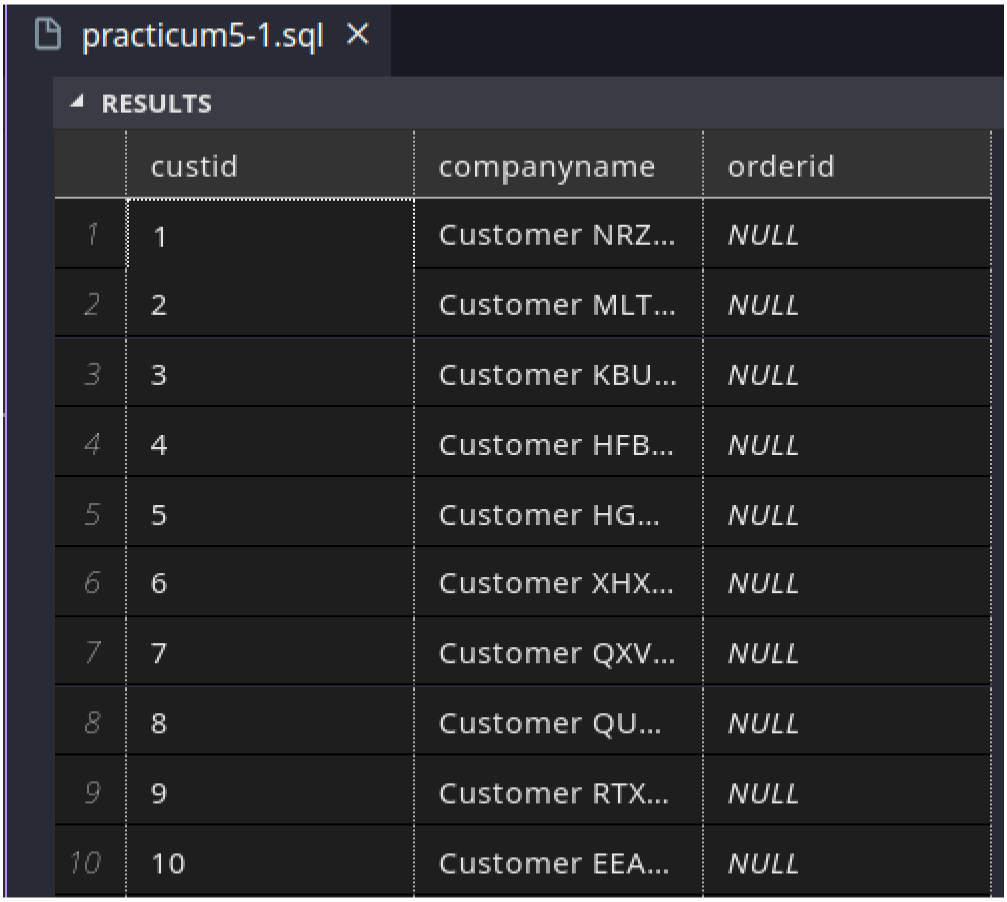
	custid	companyname	contactname	address	city	country	phone
1	4	Customer HFB...	Arndt, Torsten	7890 Hanover ...	London	UK	(171) 456-7890
2	11	Customer UB...	Jaffe, David	Fauntleroy Cir...	London	UK	(171) 789-0123
3	15	Customer JUW...	Richardson, S...	Av. dos Lus�a...	Sao Paulo	Brazil	(11) 012-3456
4	16	Customer GYB...	Birkby, Dana	Berkeley Gard...	London	UK	(171) 234-5678
5	19	Customer RFN...	Boseman, Ran...	5678 King Geo...	London	UK	(171) 345-6789
6	21	Customer KID...	Russo, Giusep...	Rua Or�s, 3456	Sao Paulo	Brazil	(11) 456-7890
7	31	Customer YJCBX	Cheng, Yao-Qi...	Av. Brasil, 5678	Campinas	Brazil	(11) 567-8901
8	32	Customer YSI...	Krishnan, Venky	6789 Baker Bl...	Eugene	USA	(503) 555-0122
9	34	Customer IBV...	Cohen, Shy	Rua do Pa�o, ...	Rio de Janeiro	Brazil	(21) 789-0123
10	36	Customer LVJSO	Smith, Denise	City Center Pl...	Elgin	USA	(503) 555-0126

Code:



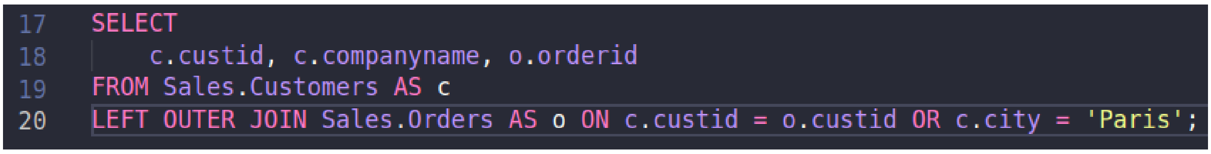
```sql
-- SELECT
--      custid, companyname, contactname, address, city, country, phone
-- FROM Sales.Customers
-- WHERE
--      country = N'Brazil';

SELECT
    c.custid, c.companyname, o.orderid
FROM Sales.Customers AS c
LEFT OUTER JOIN Sales.Orders AS o ON c.custid = o.custid and c.city = 'Paris';
```

Result:

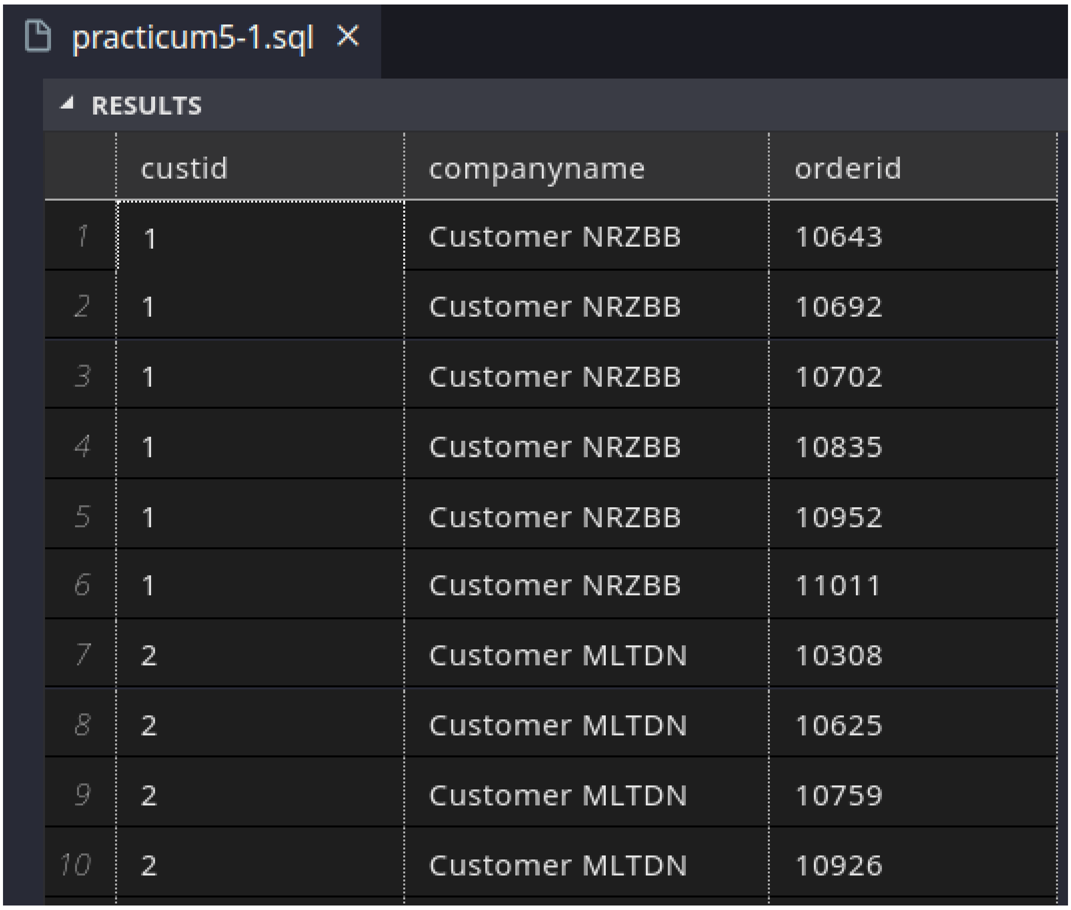


	custid	companyname	orderid
1	1	Customer NRZ...	NULL
2	2	Customer MLT...	NULL
3	3	Customer KBU...	NULL
4	4	Customer HFB...	NULL
5	5	Customer HG...	NULL
6	6	Customer XHX...	NULL
7	7	Customer QXV...	NULL
8	8	Customer QU...	NULL
9	9	Customer RTX...	NULL
10	10	Customer EEA...	NULL

10. Copy the T-SQL code in step 4 and modify it with the comparison operator for the city column in the WHERE clause with the OR operator. After that, execute the code, show the result!



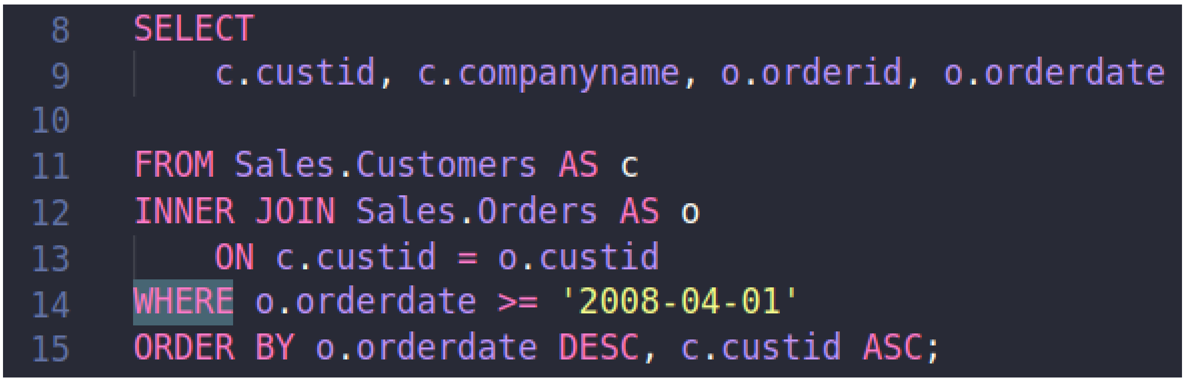
```sql
SELECT
    c.custid, c.companyname, o.orderid
FROM Sales.Customers AS c
LEFT OUTER JOIN Sales.Orders AS o ON c.custid = o.custid OR c.city = 'Paris';
```



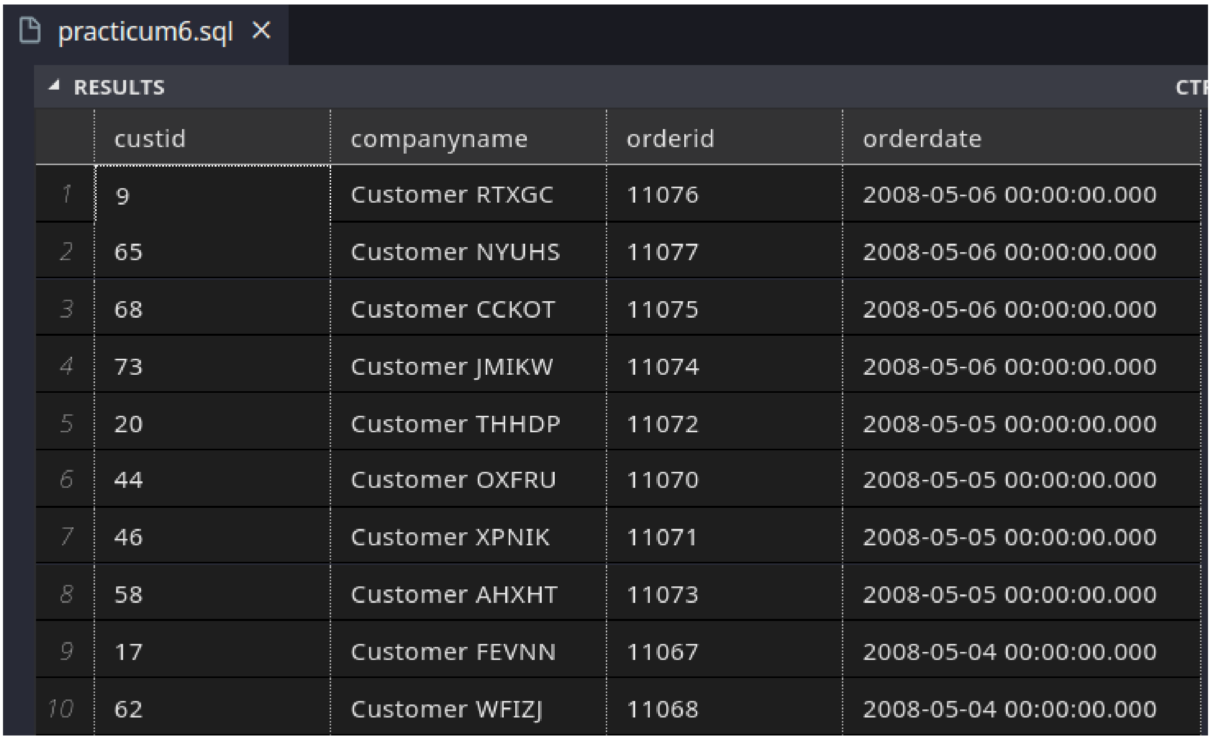
	custid	companyname	orderid
1	1	Customer NRZBB	10643
2	1	Customer NRZBB	10692
3	1	Customer NRZBB	10702
4	1	Customer NRZBB	10835
5	1	Customer NRZBB	10952
6	1	Customer NRZBB	11011
7	2	Customer MLTDN	10308
8	2	Customer MLTDN	10625
9	2	Customer MLTDN	10759
10	2	Customer MLTDN	10926

Practicum 6

11. Write a SELECT command to retrieve the custid, custname columns from the table Sales.Customers table and the orderid, orderdate columns from the Sales.Orders table! Filter the result only for orders on or after April 1, 2008. Then sort the results by orderdate descending and custid ascending!

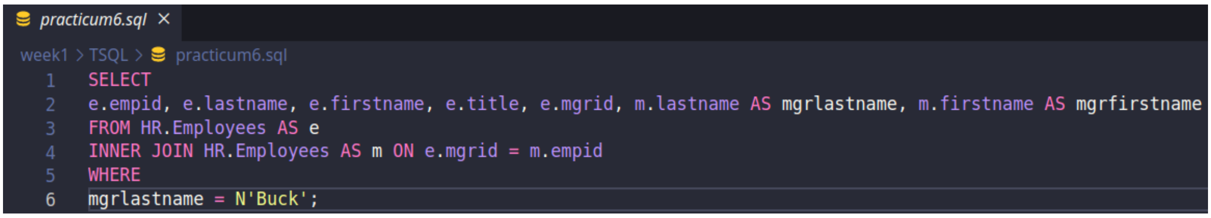


```sql
SELECT
    c.custid, c.companyname, o.orderid, o.orderdate

FROM Sales.Customers AS c
INNER JOIN Sales.Orders AS o
    ON c.custid = o.custid
WHERE o.orderdate >= '2008-04-01'
ORDER BY o.orderdate DESC, c.custid ASC;
```



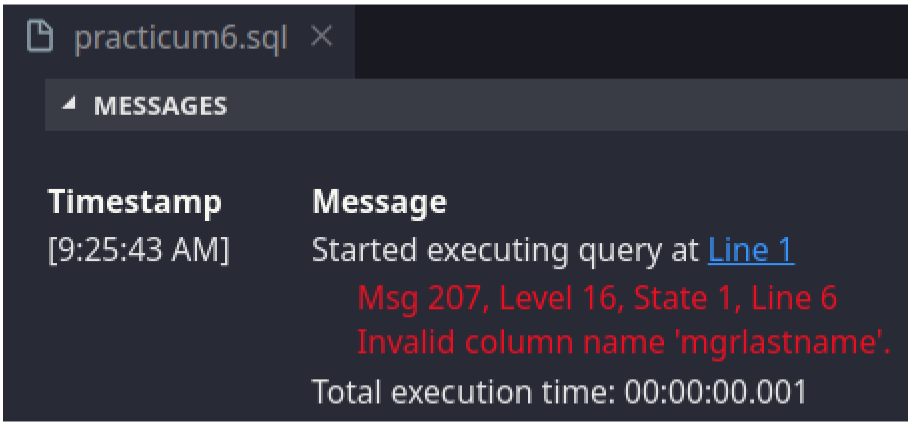
	custid	companyname	orderid	orderdate
1	9	Customer RTXGC	11076	2008-05-06 00:00:00.000
2	65	Customer NYUHS	11077	2008-05-06 00:00:00.000
3	68	Customer CCKOT	11075	2008-05-06 00:00:00.000
4	73	Customer JMIKW	11074	2008-05-06 00:00:00.000
5	20	Customer THHDP	11072	2008-05-05 00:00:00.000
6	44	Customer OXFRU	11070	2008-05-05 00:00:00.000
7	46	Customer XPNIK	11071	2008-05-05 00:00:00.000
8	58	Customer AHXHT	11073	2008-05-05 00:00:00.000
9	17	Customer FEVNN	11067	2008-05-04 00:00:00.000
10	62	Customer WFIZJ	11068	2008-05-04 00:00:00.000

Code:



```sql
SELECT
e.empid, e.lastname, e.firstname, e.title, e.mgrid, m.lastname AS mgrlastname, m.firstname AS mgrfirstname
FROM HR.Employees AS e
INNER JOIN HR.Employees AS m ON e.mgrid = m.empid
WHERE
mgrlastname = N'Buck';
```

Result:

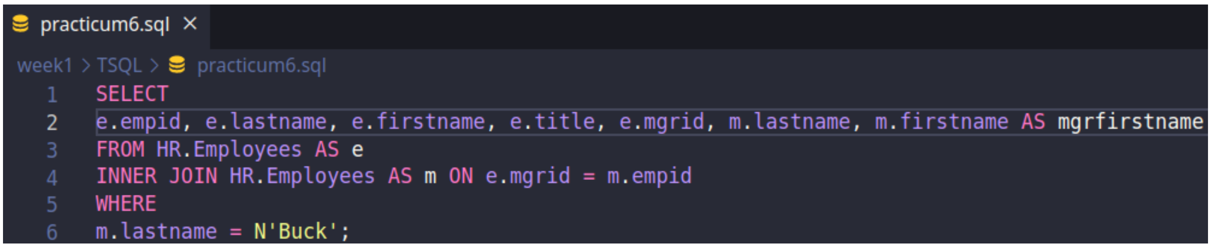


```
Timestamp      Message
[9:25:43 AM]   Started executing query at Line 1
                  Msg 207, Level 16, State 1, Line 6
                  Invalid column name 'mgrlastname'.
               Total execution time: 00:00:00.001
```

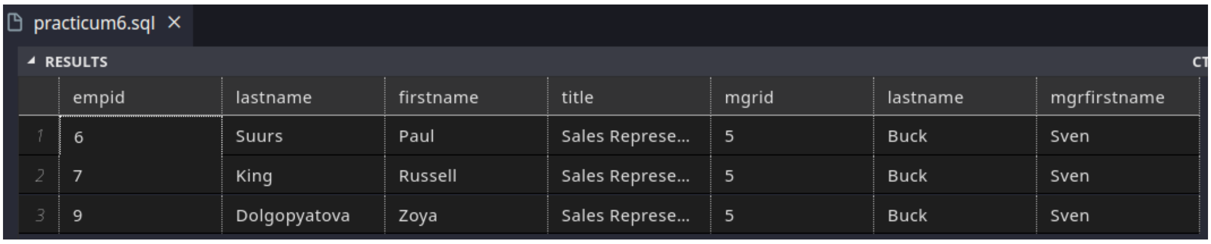
12. Execute the T-SQL command in step 3. Did an error occur? What was the error messages? What do you think is the cause?

- **According to the query above, the error message was appear because at WHERE can't write columns with AS.**

13. Make changes to the T-SQL commands to correct the error in the 3rd trial, then execute! Compare the execution result with the following result. If they are the same, then the test result is correct.

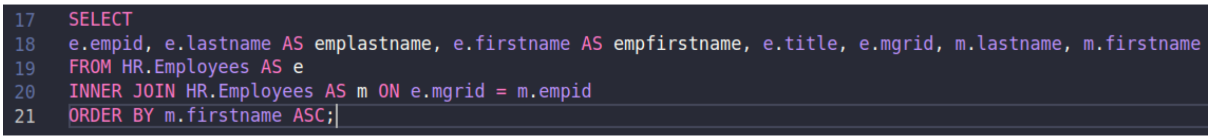


```sql
SELECT
e.empid, e.lastname, e.firstname, e.title, e.mgrid, m.lastname, m.firstname AS mgrfirstname
FROM HR.Employees AS e
INNER JOIN HR.Employees AS m ON e.mgrid = m.empid
WHERE
m.lastname = N'Buck';
```

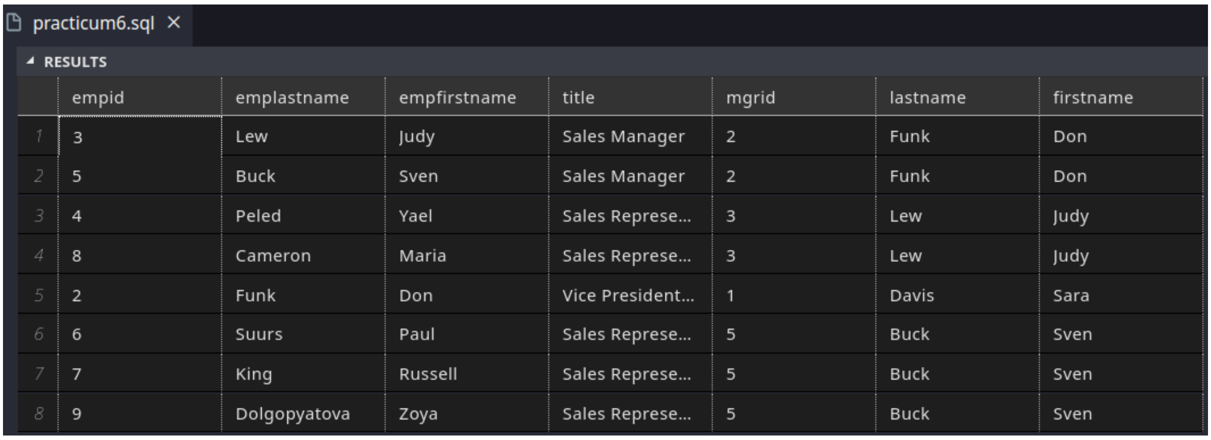


	empid	lastname	firstname	title	mgrid	lastname	mgrfirstname
1	6	Suurs	Paul	Sales Represe...	5	Buck	Sven
2	7	King	Russell	Sales Represe...	5	Buck	Sven
3	9	Dolgopyatova	Zoya	Sales Represe...	5	Buck	Sven

14. Copy the T-SQL command in trial 4, and modify it so that it lists all the employees ORDER BY manager's first name. Initially test using the table, then test using the table alias! Execute the T-SQL and compare it with the following result. If the results are the same, then the test is correct.



```sql
SELECT
e.empid, e.lastname AS emplastname, e.firstname AS empfirstname, e.title, e.mgrid, m.lastname, m.firstname
FROM HR.Employees AS e
INNER JOIN HR.Employees AS m ON e.mgrid = m.empid
ORDER BY m.firstname ASC;
```



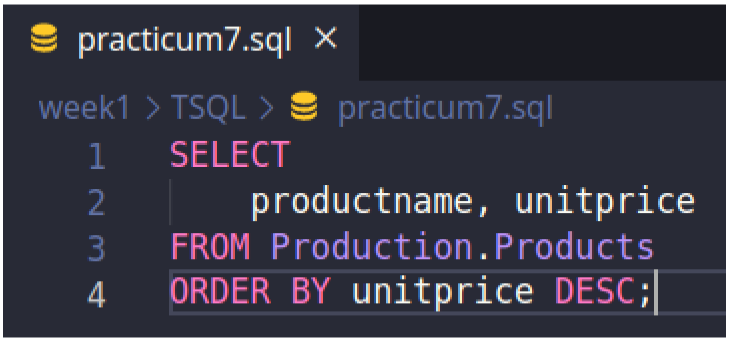
	empid	emplastname	empfirstname	title	mgrid	lastname	firstname
1	3	Lew	Judy	Sales Manager	2	Funk	Don
2	5	Buck	Sven	Sales Manager	2	Funk	Don
3	4	Peled	Yael	Sales Represe...	3	Lew	Judy
4	8	Cameron	Maria	Sales Represe...	3	Lew	Judy
5	2	Funk	Don	Vice President...	1	Davis	Sara
6	6	Suurs	Paul	Sales Represe...	5	Buck	Sven
7	7	King	Russell	Sales Represe...	5	Buck	Sven
8	9	Dolgopyatova	Zoya	Sales Represe...	5	Buck	Sven

15. Why can we use column names that match the table's original name or use a table alias?

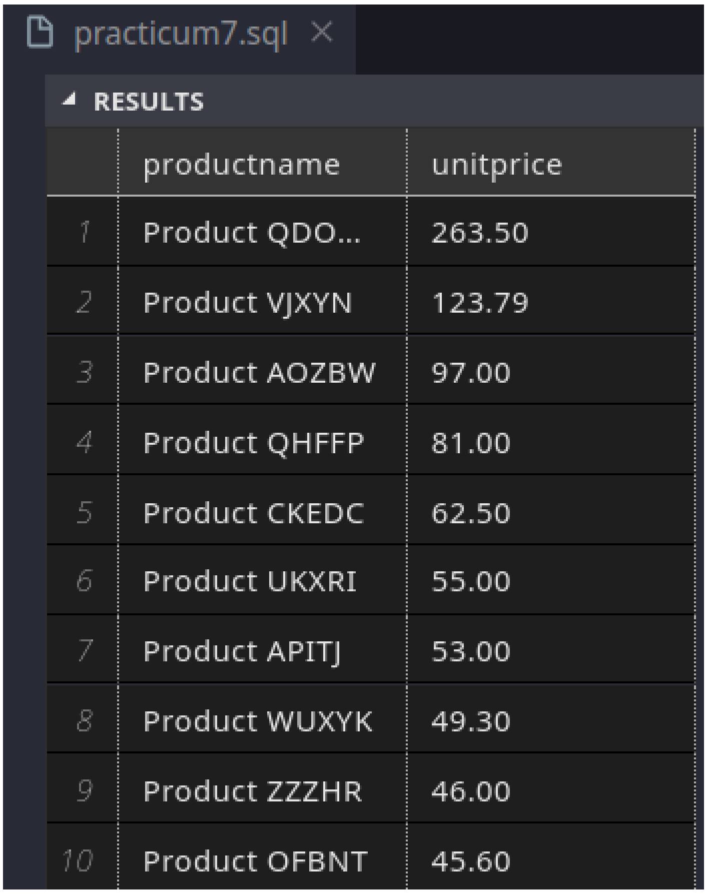
- **Because we can use ALIAS to change name column or table that we want.**

Practicum 7

16. Write a SELECT command to display the productname and unitprice columns in the Production.Products table sorted in descending order by unitprice! Display the execution result!

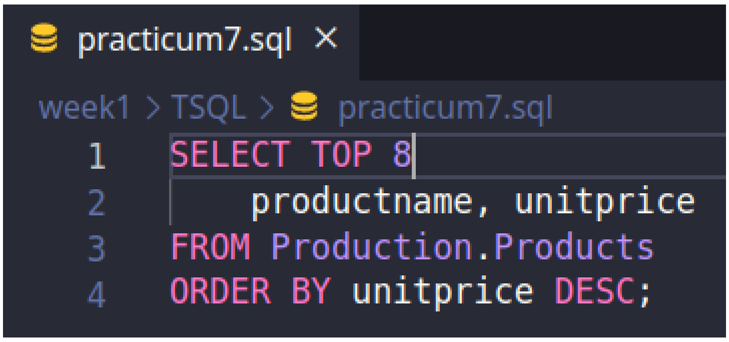


```sql
SELECT
    productname, unitprice
FROM Production.Products
ORDER BY unitprice DESC;
```

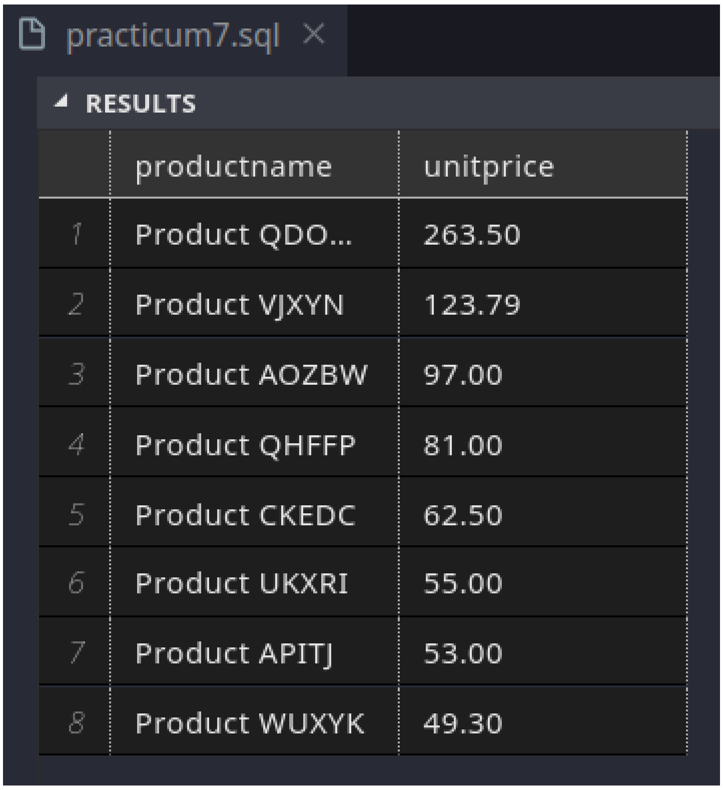


	productname	unitprice
1	Product QDO...	263.50
2	Product VJXYN	123.79
3	Product AOZBW	97.00
4	Product QHFFP	81.00
5	Product CKEDC	62.50
6	Product UKXRI	55.00
7	Product APITJ	53.00
8	Product WUXYK	49.30
9	Product ZZZHR	46.00
10	Product OFBNT	45.60

17. Copy and modify the T-SQL command in trial 2 with the restriction that only 8 products that children are displayed based on unitprice ordering!Execute the command, and compare it with the following results.



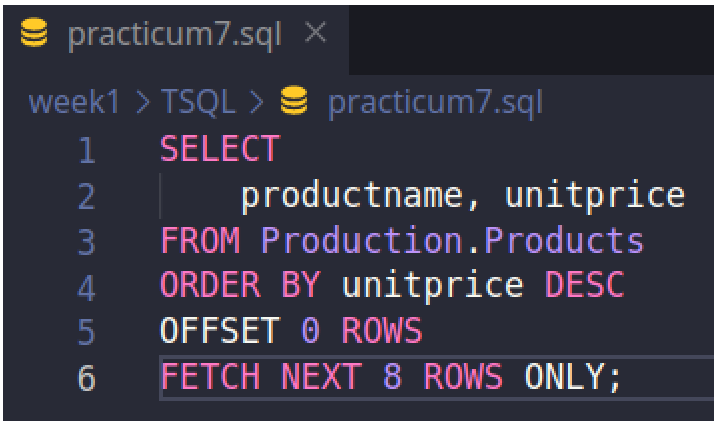
```sql
SELECT TOP 8
    productname, unitprice
FROM Production.Products
ORDER BY unitprice DESC;
```



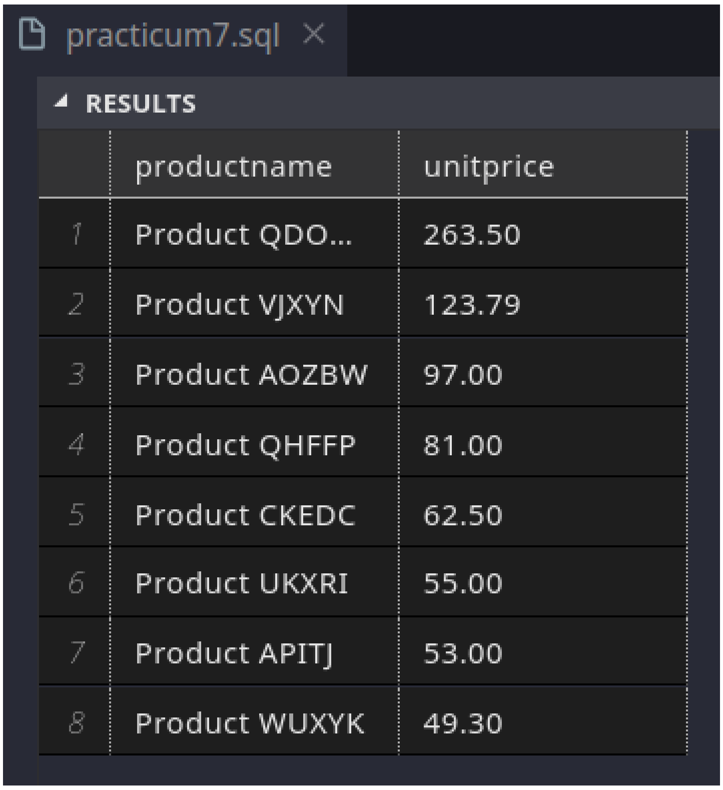
	productname	unitprice
1	Product QDO...	263.50
2	Product VJXYN	123.79
3	Product AOZBW	97.00
4	Product QHFFP	81.00
5	Product CKEDC	62.50
6	Product UKXRI	55.00
7	Product APITJ	53.00
8	Product WUXYK	49.30

18. Is it possible to implement the T-SQL command of trial 5 using the OFFSET-FETCH clause?

- **Yes it can. We write OFFSET to 0 ROWS, then set FETCH NEXT to 8 ROWS ONLY.**



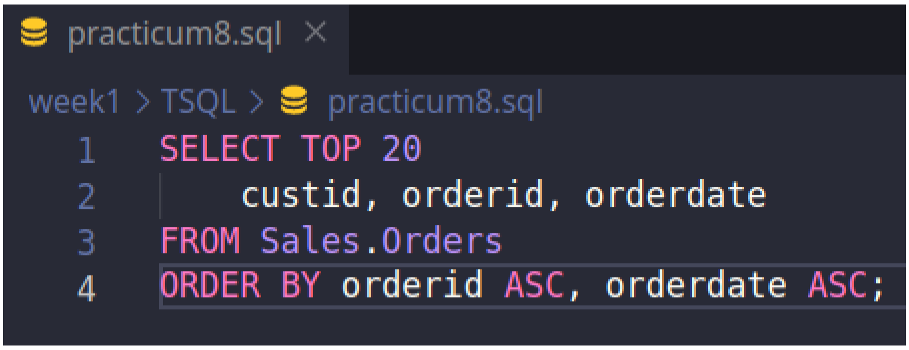
```
SELECT
    productname, unitprice
FROM Production.Products
ORDER BY unitprice DESC
OFFSET 0 ROWS
FETCH NEXT 8 ROWS ONLY;
```



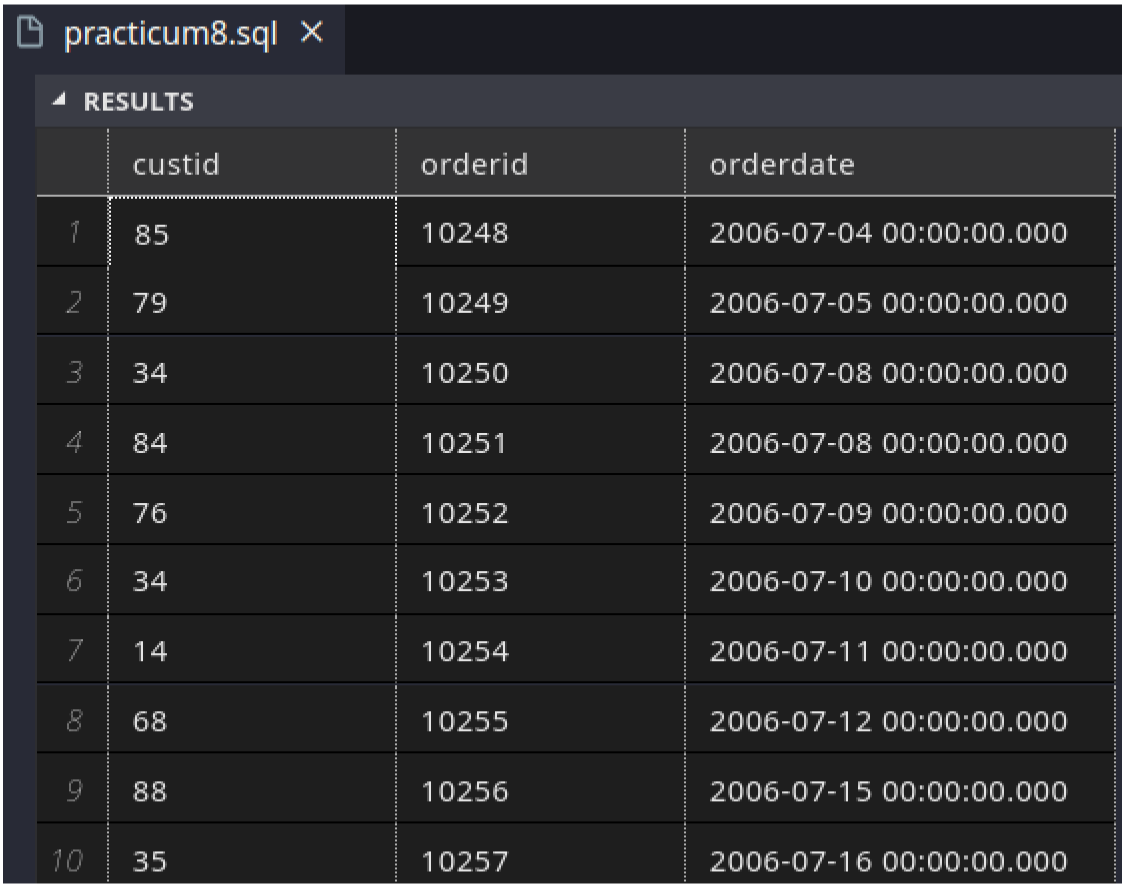
	productname	unitprice
1	Product QDO...	263.50
2	Product VJXYN	123.79
3	Product AOZBW	97.00
4	Product QHFFP	81.00
5	Product CKEDC	62.50
6	Product UKXRI	55.00
7	Product APITJ	53.00
8	Product WUXYK	49.30

Practicum 8

19. Write a SELECT command to display the custid, orderid, and orderdate columns in the Sales.Orders table. Sort the rows by orderdate and orderid. Retrieve the first 20 rows. Execute the command and compare the results below. If the results are the same, then your test is correct.

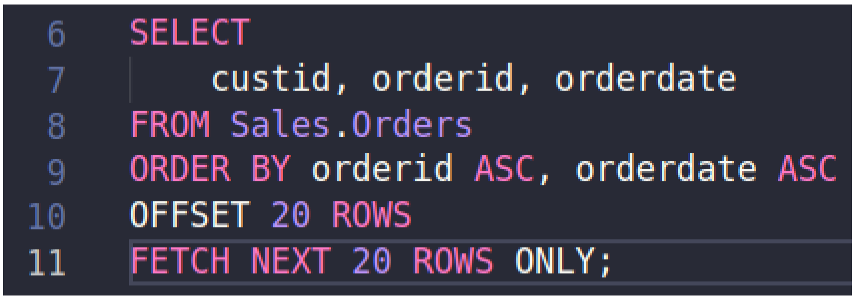


```sql
SELECT TOP 20
    custid, orderid, orderdate
FROM Sales.Orders
ORDER BY orderid ASC, orderdate ASC;
```

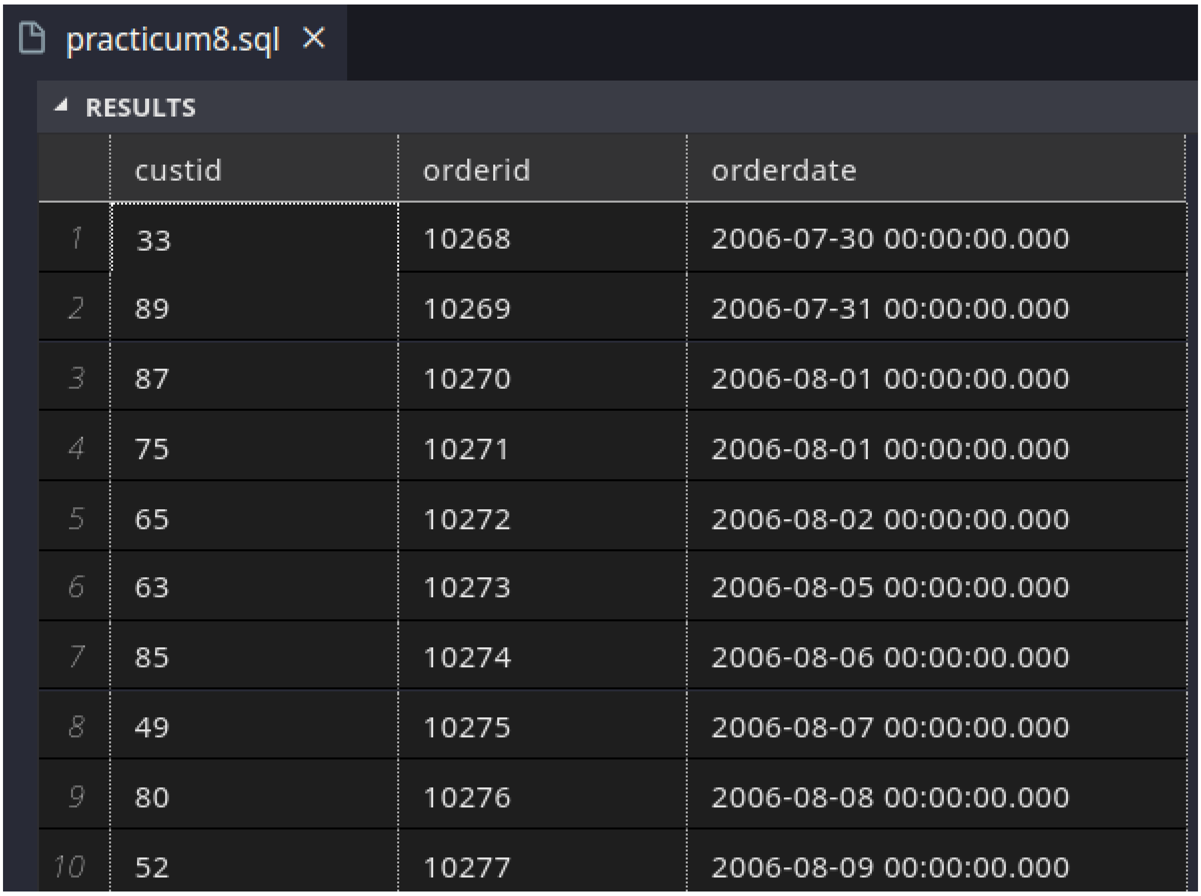


	custid	orderid	orderdate
1	85	10248	2006-07-04 00:00:00.000
2	79	10249	2006-07-05 00:00:00.000
3	34	10250	2006-07-08 00:00:00.000
4	84	10251	2006-07-08 00:00:00.000
5	76	10252	2006-07-09 00:00:00.000
6	34	10253	2006-07-10 00:00:00.000
7	14	10254	2006-07-11 00:00:00.000
8	68	10255	2006-07-12 00:00:00.000
9	88	10256	2006-07-15 00:00:00.000
10	35	10257	2006-07-16 00:00:00.000

20. Write a SELECT command to display the same result as question no. 19, skip the first 20 rows, and continue with the next 20 rows using the OFFSET- clause. FETCH CLAUSE! Execute the command and compare it with the following result. If the result is the same, then your test is correct.



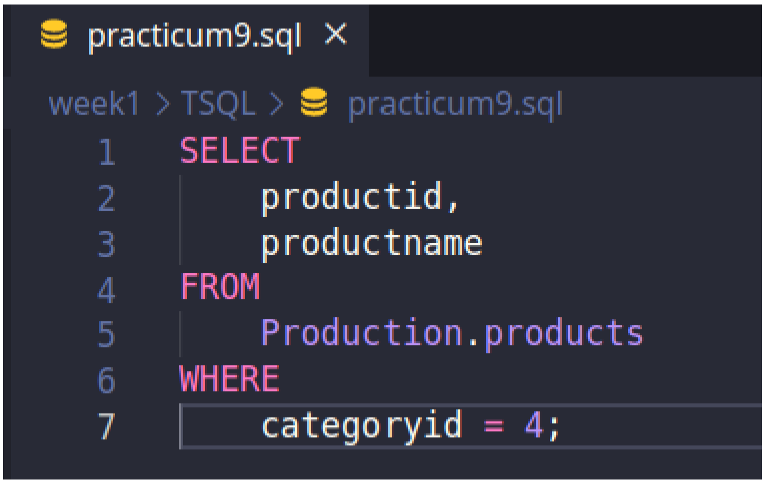
```sql
SELECT
    custid, orderid, orderdate
FROM Sales.Orders
ORDER BY orderid ASC, orderdate ASC
OFFSET 20 ROWS
FETCH NEXT 20 ROWS ONLY;
```



	custid	orderid	orderdate
1	33	10268	2006-07-30 00:00:00.000
2	89	10269	2006-07-31 00:00:00.000
3	87	10270	2006-08-01 00:00:00.000
4	75	10271	2006-08-01 00:00:00.000
5	65	10272	2006-08-02 00:00:00.000
6	63	10273	2006-08-05 00:00:00.000
7	85	10274	2006-08-06 00:00:00.000
8	49	10275	2006-08-07 00:00:00.000
9	80	10276	2006-08-08 00:00:00.000
10	52	10277	2006-08-09 00:00:00.000

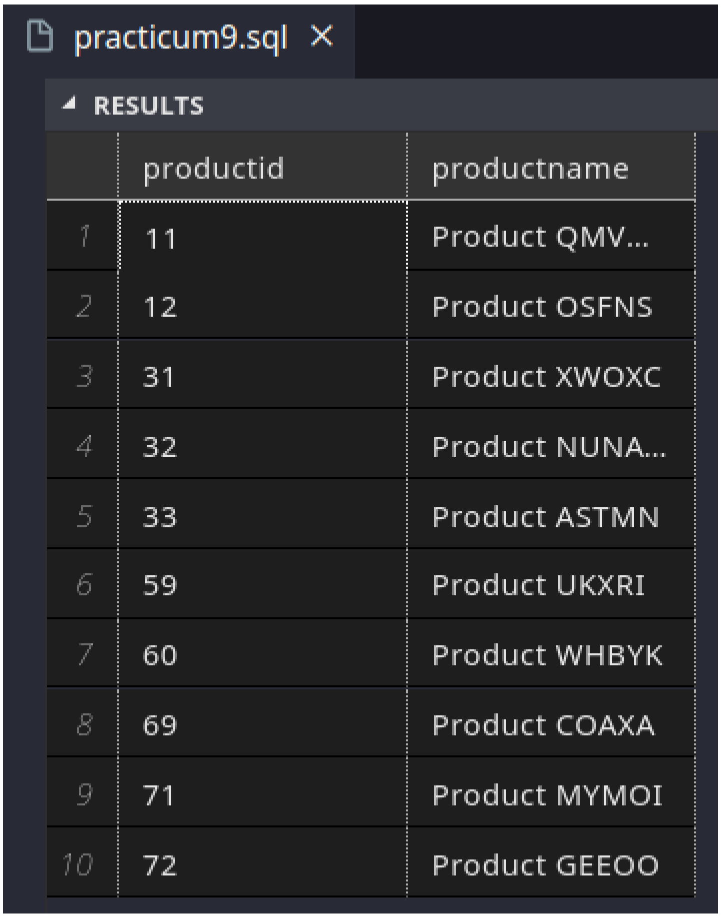
Practicum 9

Code:



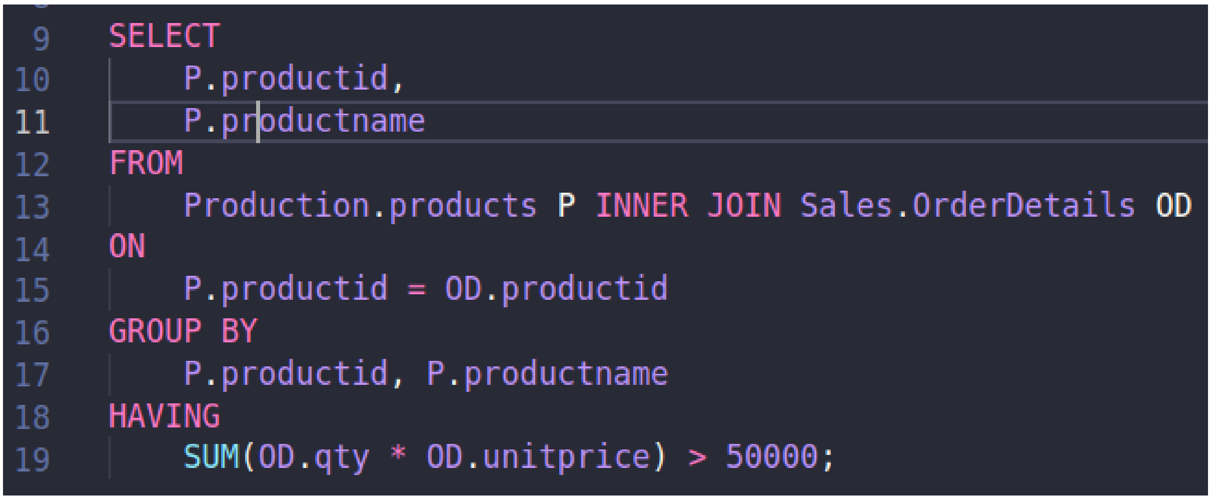
```sql
SELECT
    productid,
    productname
FROM
    Production.products
WHERE
    categoryid = 4;
```

Result:



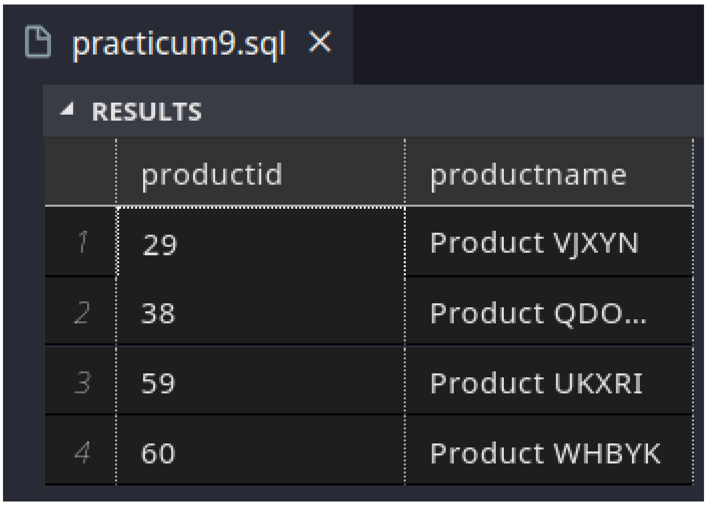
	productid	productname
1	11	Product QMV...
2	12	Product OSFNS
3	31	Product XWOXC
4	32	Product NUNA...
5	33	Product ASTMN
6	59	Product UKXRI
7	60	Product WHBYK
8	69	Product COAXA
9	71	Product MYMOI
10	72	Product GEEOO

Code:



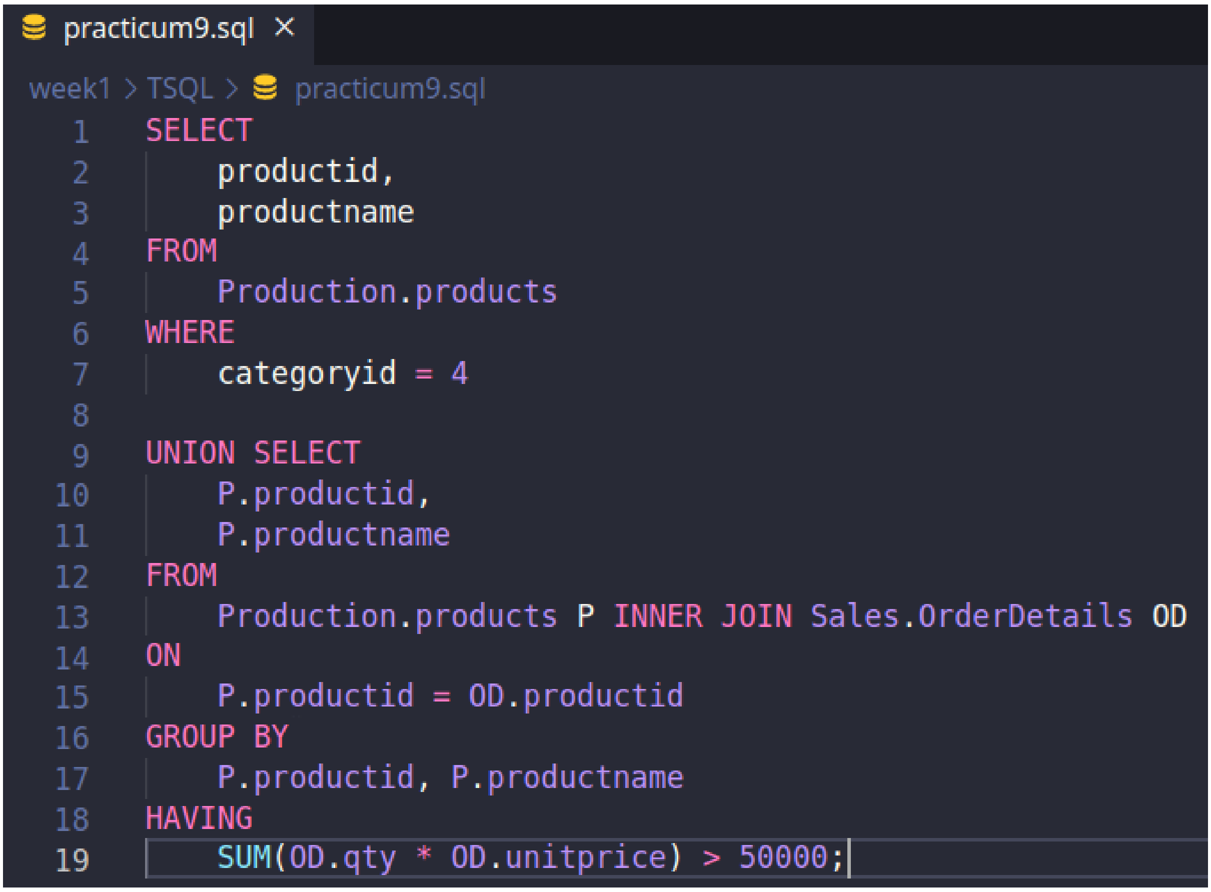
```sql
SELECT
    P.productid,
    P.productname
FROM
    Production.products P INNER JOIN Sales.OrderDetails OD
ON
    P.productid = OD.productid
GROUP BY
    P.productid, P.productname
HAVING
    SUM(OD.qty * OD.unitprice) > 50000;
```

Result:

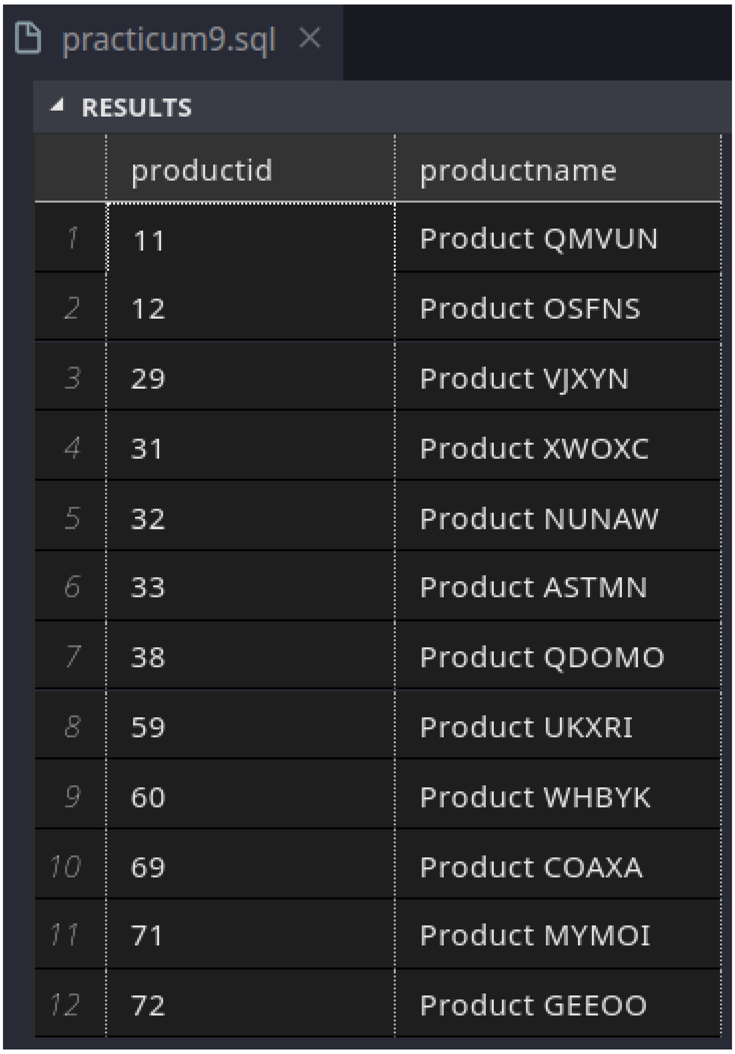


	productid	productname
1	29	Product VJXYN
2	38	Product QDO...
3	59	Product UKXRI
4	60	Product WHBYK

21. Write an SQL that displays the results in practicum-1 step-1 & 2 at once (combined) using UNION! at once (combined) by using UNION! Instructions: Put UNION between the two SQLs.

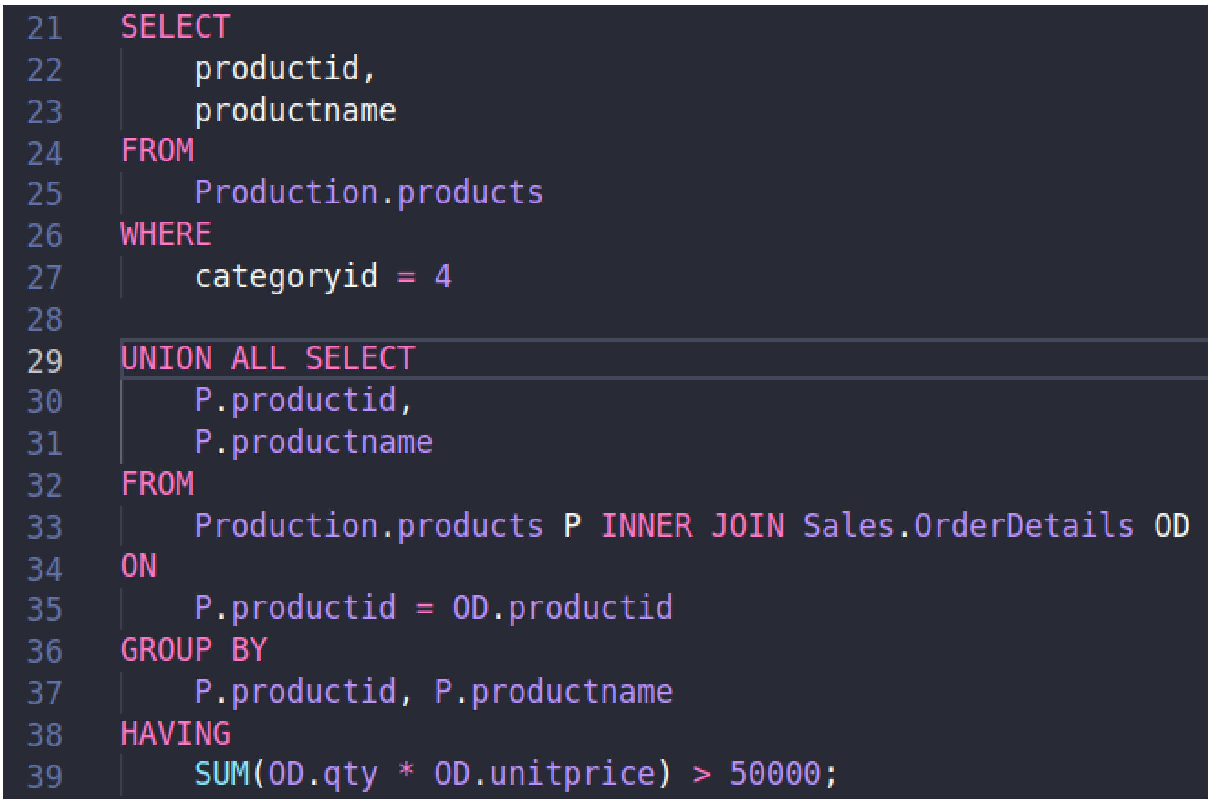


```sql
SELECT
    productid,
    productname
FROM
    Production.products
WHERE
    categoryid = 4

UNION SELECT
    P.productid,
    P.productname
FROM
    Production.products P INNER JOIN Sales.OrderDetails OD
ON
    P.productid = OD.productid
GROUP BY
    P.productid, P.productname
HAVING
    SUM(OD.qty * OD.unitprice) > 50000;
```

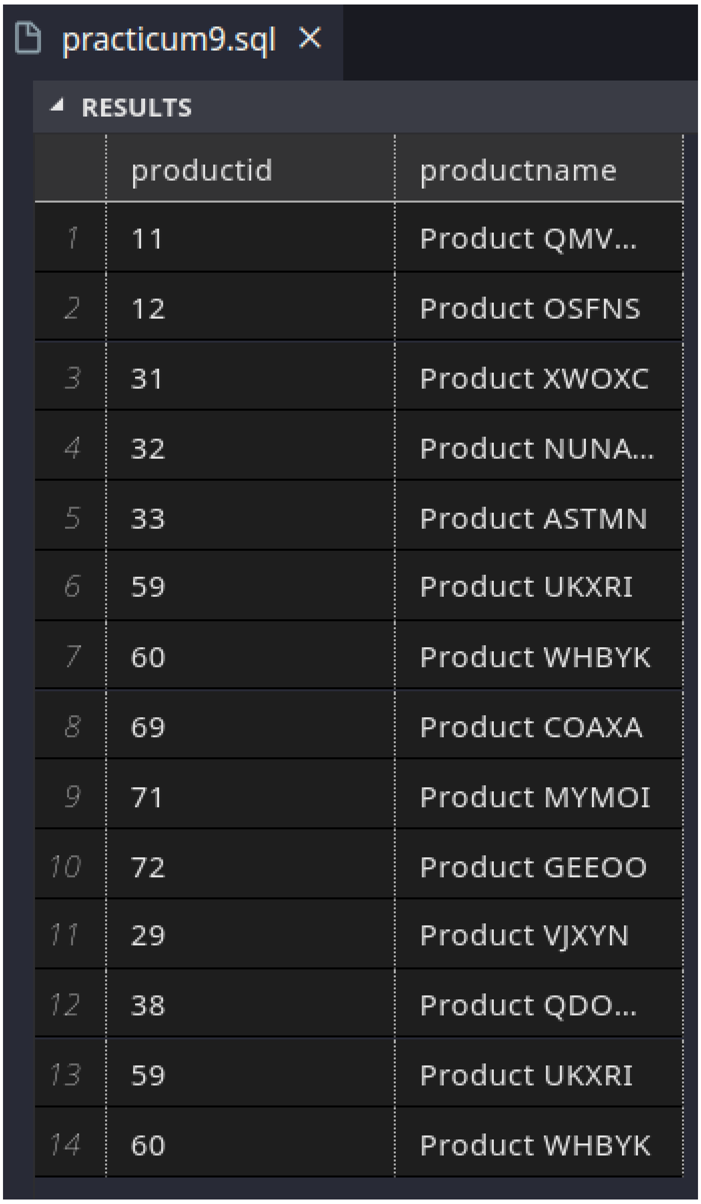


	productid	productname
1	11	Product QMVUN
2	12	Product OSFNS
3	29	Product VJXYN
4	31	Product XWOXC
5	32	Product NUNAW
6	33	Product ASTMN
7	38	Product QDOMO
8	59	Product UKXRI
9	60	Product WHBYK
10	69	Product COAXA
11	71	Product MYMOI
12	72	Product GEEOO

22. Similar to the previous step, this time write an SQL that displays results in practicum-1 step-1 & 2 at once (combined) by using UNION ALL!



```sql
SELECT
    productid,
    productname
FROM
    Production.products
WHERE
    categoryid = 4

UNION ALL SELECT
    P.productid,
    P.productname
FROM
    Production.products P INNER JOIN Sales.OrderDetails OD
ON
    P.productid = OD.productid
GROUP BY
    P.productid, P.productname
HAVING
    SUM(OD.qty * OD.unitprice) > 50000;
```



practicum9.sql

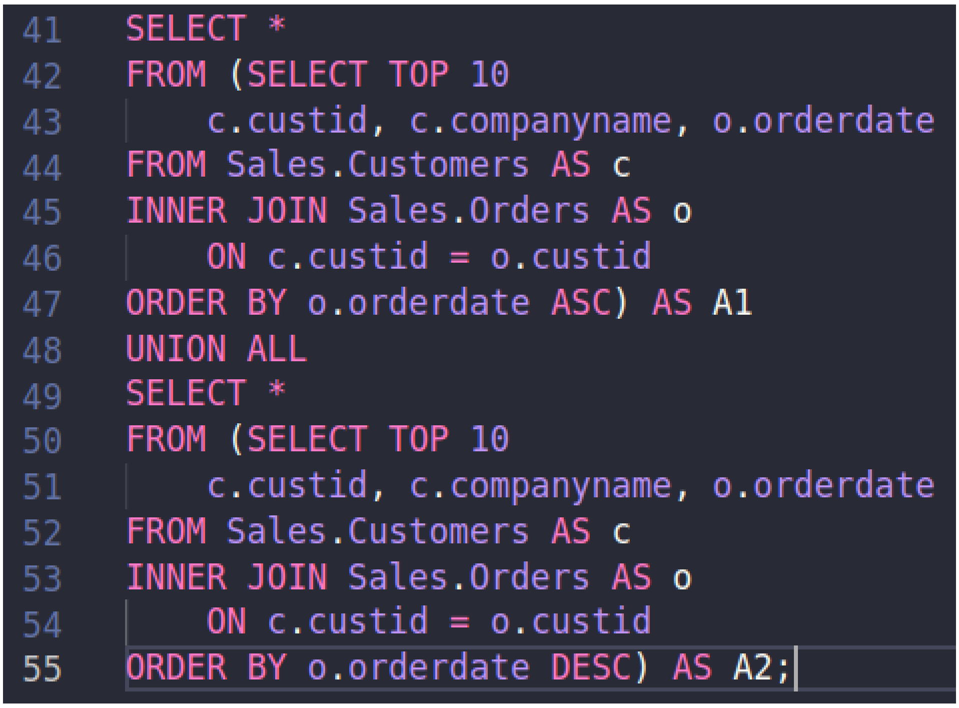
RESULTS

	productid	productname
1	11	Product QMV...
2	12	Product OSFNS
3	31	Product XWOXC
4	32	Product NUNA...
5	33	Product ASTMN
6	59	Product UKXRI
7	60	Product WHBYK
8	69	Product COAXA
9	71	Product MYMOI
10	72	Product GEEOO
11	29	Product VJXYN
12	38	Product QDO...
13	59	Product UKXRI
14	60	Product WHBYK

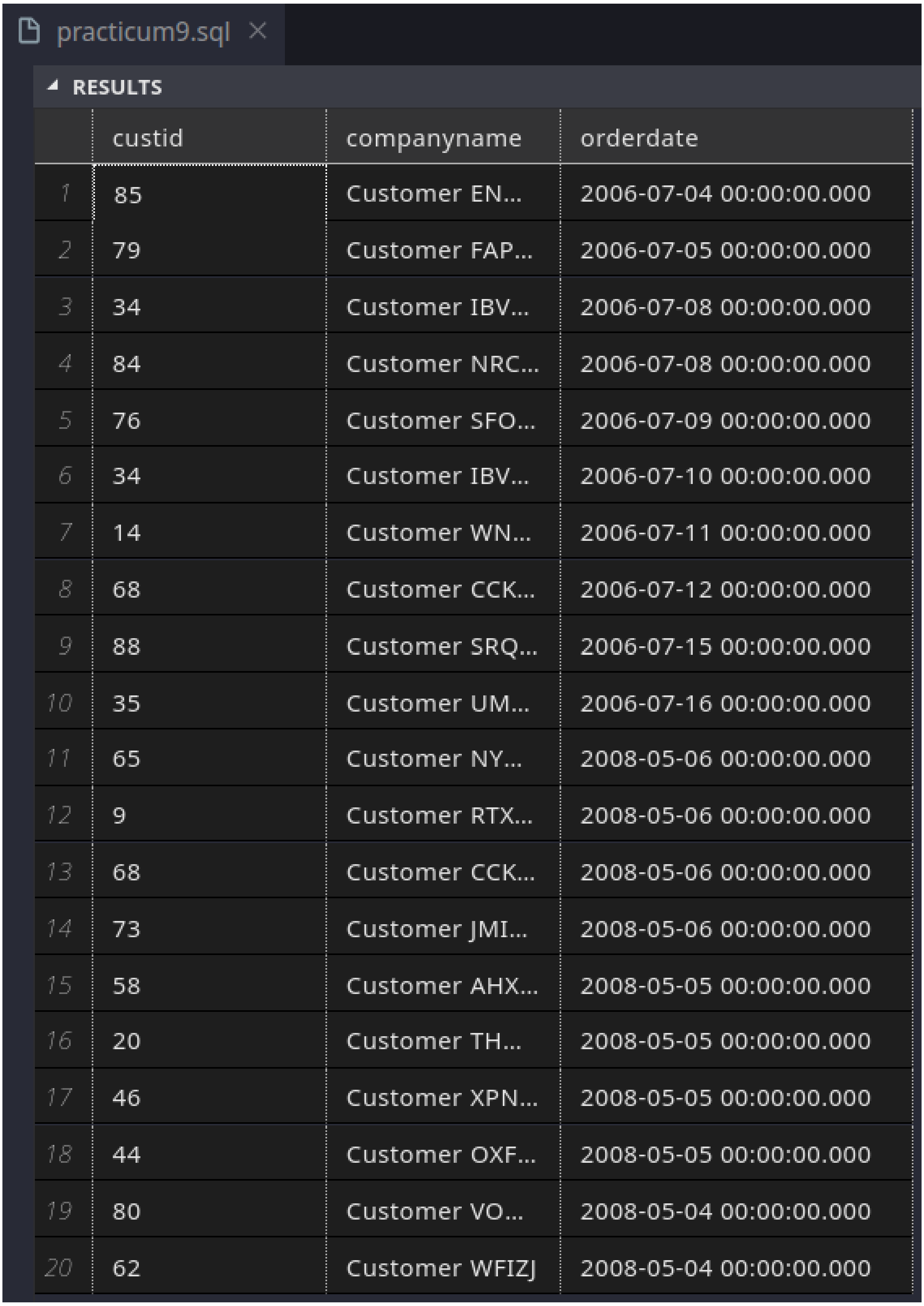
23. What is the difference between UNION and UNION ALL?

- **The difference between both is UNION ALL - Includes duplicates, then UNION - Excludes duplicates.**

24. Write SQL to display the 10 customers who bought the earliest as well as the 10 customers who bought last.



```sql
SELECT *
FROM (SELECT TOP 10
    c.custid, c.companyname, o.orderdate
FROM Sales.Customers AS c
INNER JOIN Sales.Orders AS o
    ON c.custid = o.custid
ORDER BY o.orderdate ASC) AS A1
UNION ALL
SELECT *
FROM (SELECT TOP 10
    c.custid, c.companyname, o.orderdate
FROM Sales.Customers AS c
INNER JOIN Sales.Orders AS o
    ON c.custid = o.custid
ORDER BY o.orderdate DESC) AS A2;
```



	custid	companyname	orderdate
1	85	Customer EN...	2006-07-04 00:00:00.000
2	79	Customer FAP...	2006-07-05 00:00:00.000
3	34	Customer IBV...	2006-07-08 00:00:00.000
4	84	Customer NRC...	2006-07-08 00:00:00.000
5	76	Customer SFO...	2006-07-09 00:00:00.000
6	34	Customer IBV...	2006-07-10 00:00:00.000
7	14	Customer WN...	2006-07-11 00:00:00.000
8	68	Customer CCK...	2006-07-12 00:00:00.000
9	88	Customer SRQ...	2006-07-15 00:00:00.000
10	35	Customer UM...	2006-07-16 00:00:00.000
11	65	Customer NY...	2008-05-06 00:00:00.000
12	9	Customer RTX...	2008-05-06 00:00:00.000
13	68	Customer CCK...	2008-05-06 00:00:00.000
14	73	Customer JMI...	2008-05-06 00:00:00.000
15	58	Customer AHX...	2008-05-05 00:00:00.000
16	20	Customer TH...	2008-05-05 00:00:00.000
17	46	Customer XPN...	2008-05-05 00:00:00.000
18	44	Customer OXF...	2008-05-05 00:00:00.000
19	80	Customer VO...	2008-05-04 00:00:00.000
20	62	Customer WFIZJ	2008-05-04 00:00:00.000