Databases - T1

Lab 10: SQL Sub Queries

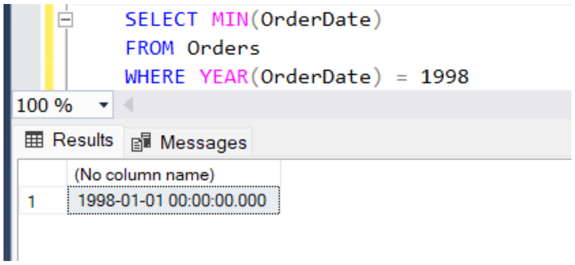
Syed Asghar Abbas Zaidi

07201, Fall 2025

1. Find the employee who processed the first order placed in the year 1998.

- **Output:** EmployeeID
- **Result:** 1 rows

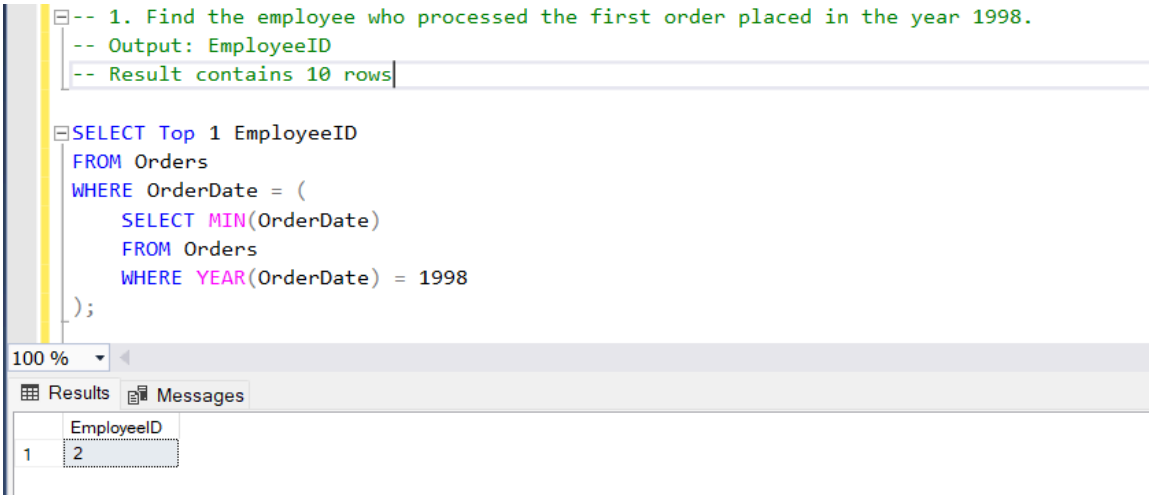
Subquery:



```sql
    SELECT MIN(OrderDate)
    FROM Orders
    WHERE YEAR(OrderDate) = 1998
```

	(No column name)
1	1998-01-01 00:00:00.000

Overall Query:



```sql
-- 1. Find the employee who processed the first order placed in the year 1998.
-- Output: EmployeeID
-- Result contains 10 rows

SELECT Top 1 EmployeeID
FROM Orders
WHERE OrderDate = (
    SELECT MIN(OrderDate)
    FROM Orders
    WHERE YEAR(OrderDate) = 1998
);
```

	EmployeeID
1	2

2. Select all employees who work directly under the top manager of the company.

- **Output:** EmployeeID
- **Result:** 5 rows

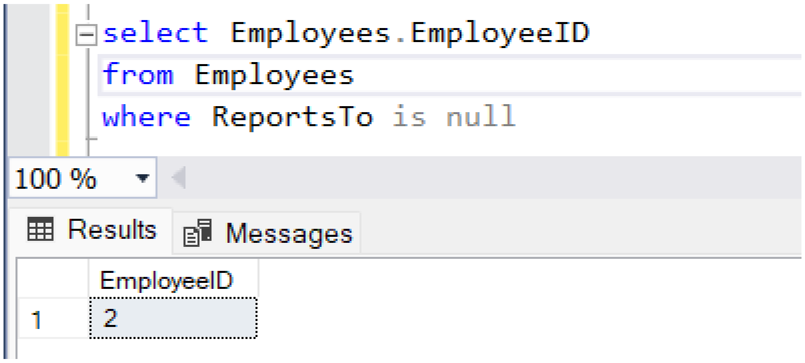
Subquery:

Databases - T1

Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi

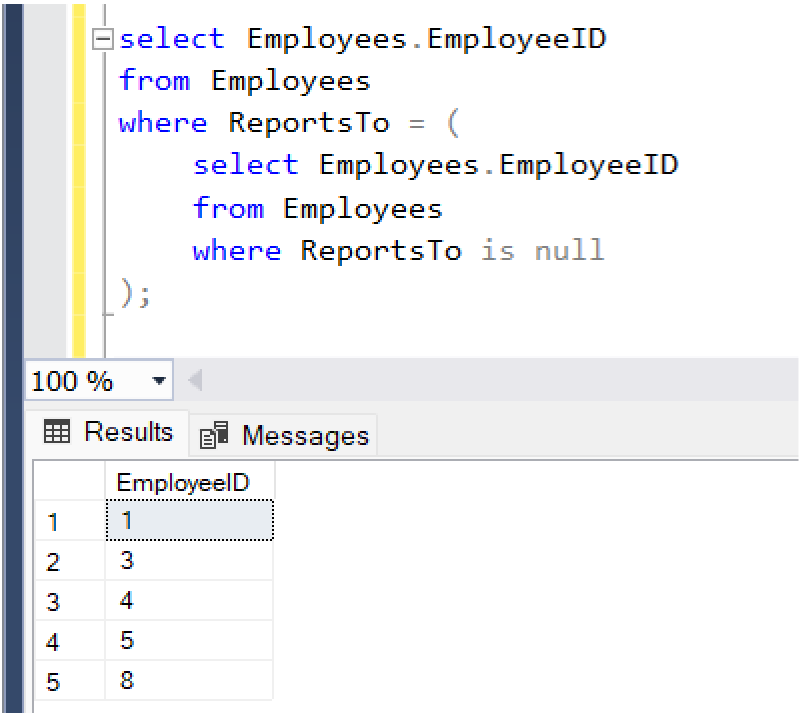
07201, Fall 2025



```
select Employees.EmployeeID
from Employees
where ReportsTo is null
```

	EmployeeID
1	2

Overall query:



```
select Employees.EmployeeID
from Employees
where ReportsTo = (
    select Employees.EmployeeID
    from Employees
    where ReportsTo is null
);
```

	EmployeeID
1	1
2	3
3	4
4	5
5	8

3. Select all employees who are assigned to territories in the 'Western' and 'Eastern' regions from the Region Table.

- **Output:** EmployeeID
- **Result:** 6 rows

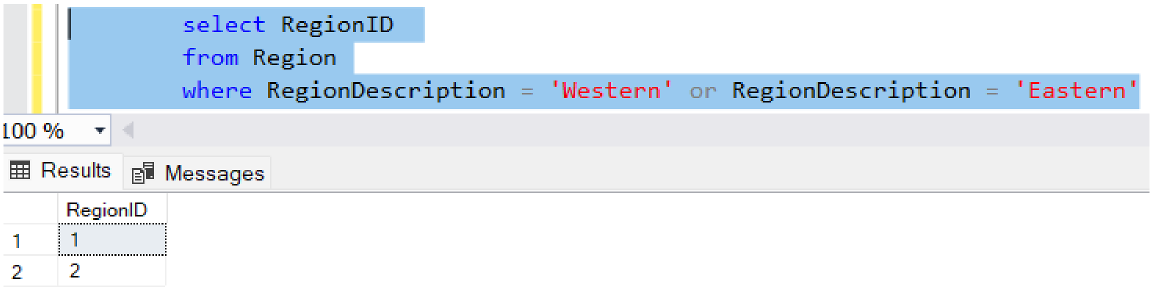
Innermost Subquery: Where I am trying to find what are the RegionID of Western and Eastern

Databases - T1

Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi

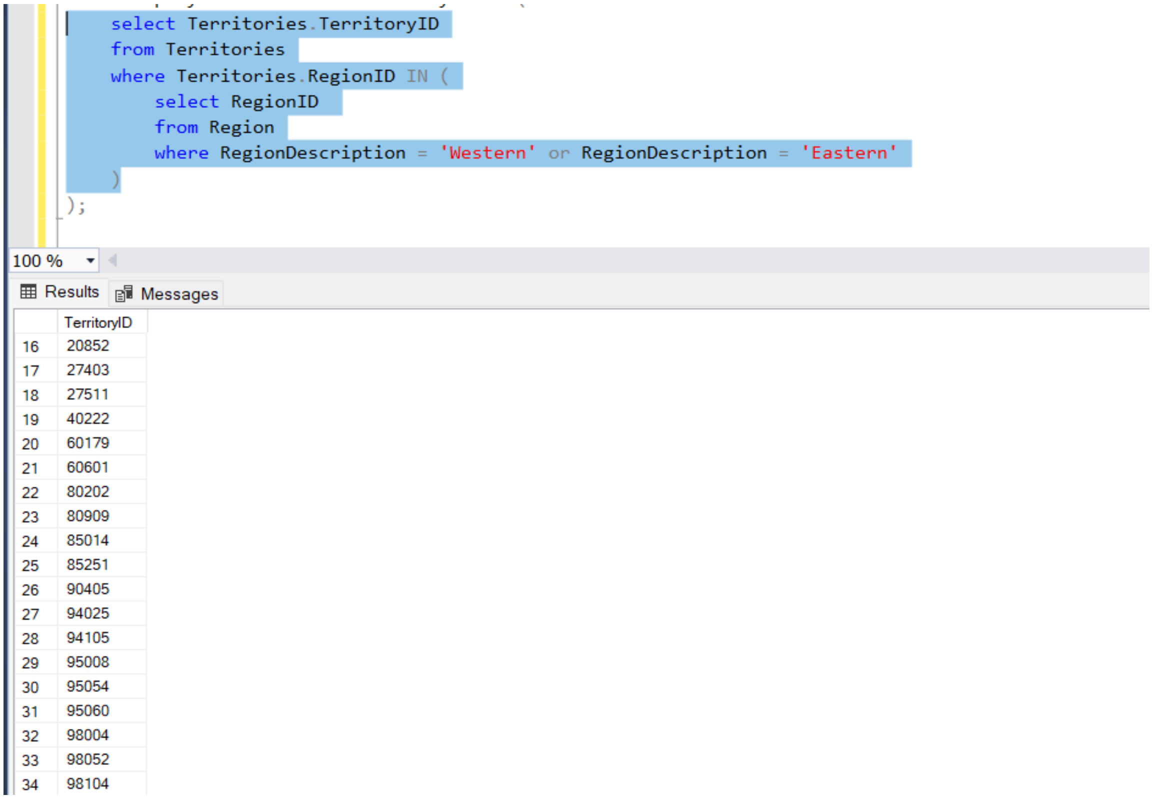
07201, Fall 2025



```
select RegionID
from Region
where RegionDescription = 'Western' or RegionDescription = 'Eastern'
```

	RegionID
1	1
2	2

Middle Subquery: Where I am trying to find all the territories that come under the RegionIDs of Eastern and Western



```
    select Territories.TerritoryID
    from Territories
    where Territories.RegionID IN (
        select RegionID
        from Region
        where RegionDescription = 'Western' or RegionDescription = 'Eastern'
    )
);
```

	TerritoryID
16	20852
17	27403
18	27511
19	40222
20	60179
21	60601
22	80202
23	80909
24	85014
25	85251
26	90405
27	94025
28	94105
29	95008
30	95054
31	95060
32	98004
33	98052
34	98104

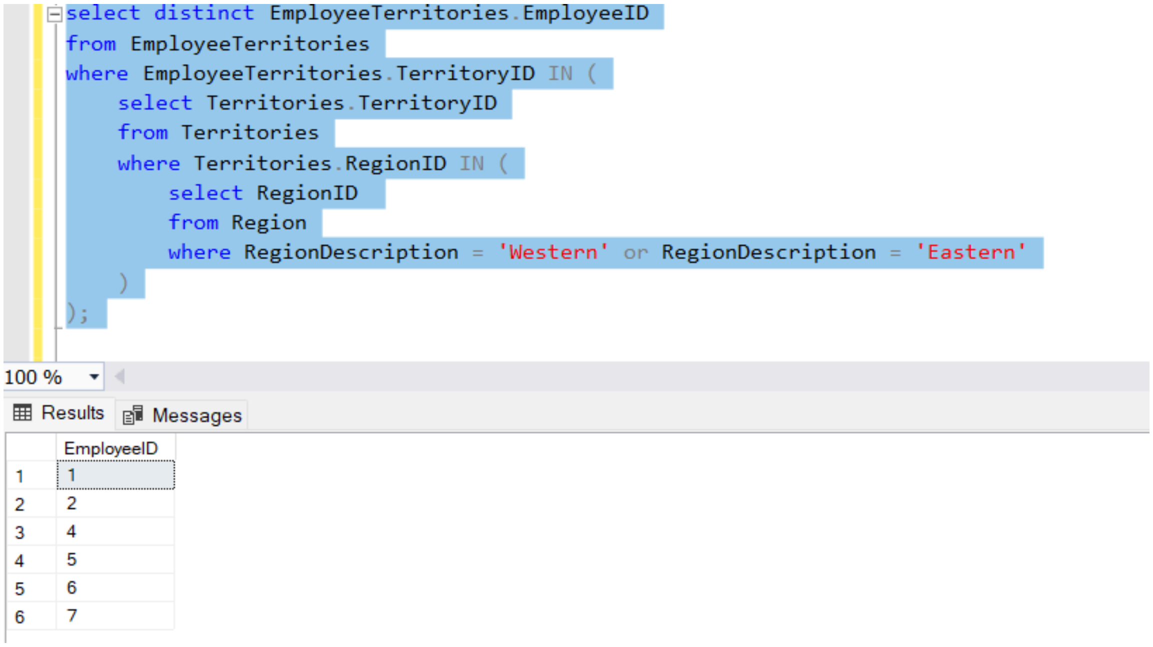
Overall Query: All the employees to be found in those territories that come the region of Western and Eastern

Databases - T1

Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi

07201, Fall 2025



```sql
select distinct EmployeeTerritories.EmployeeID
from EmployeeTerritories
where EmployeeTerritories.TerritoryID IN (
    select Territories.TerritoryID
    from Territories
    where Territories.RegionID IN (
        select RegionID
        from Region
        where RegionDescription = 'Western' or RegionDescription = 'Eastern'
    )
);
```

	EmployeeID
1	1
2	2
3	4
4	5
5	6
6	7

4. Select all Customers and Suppliers belonging to ‘Germany’.

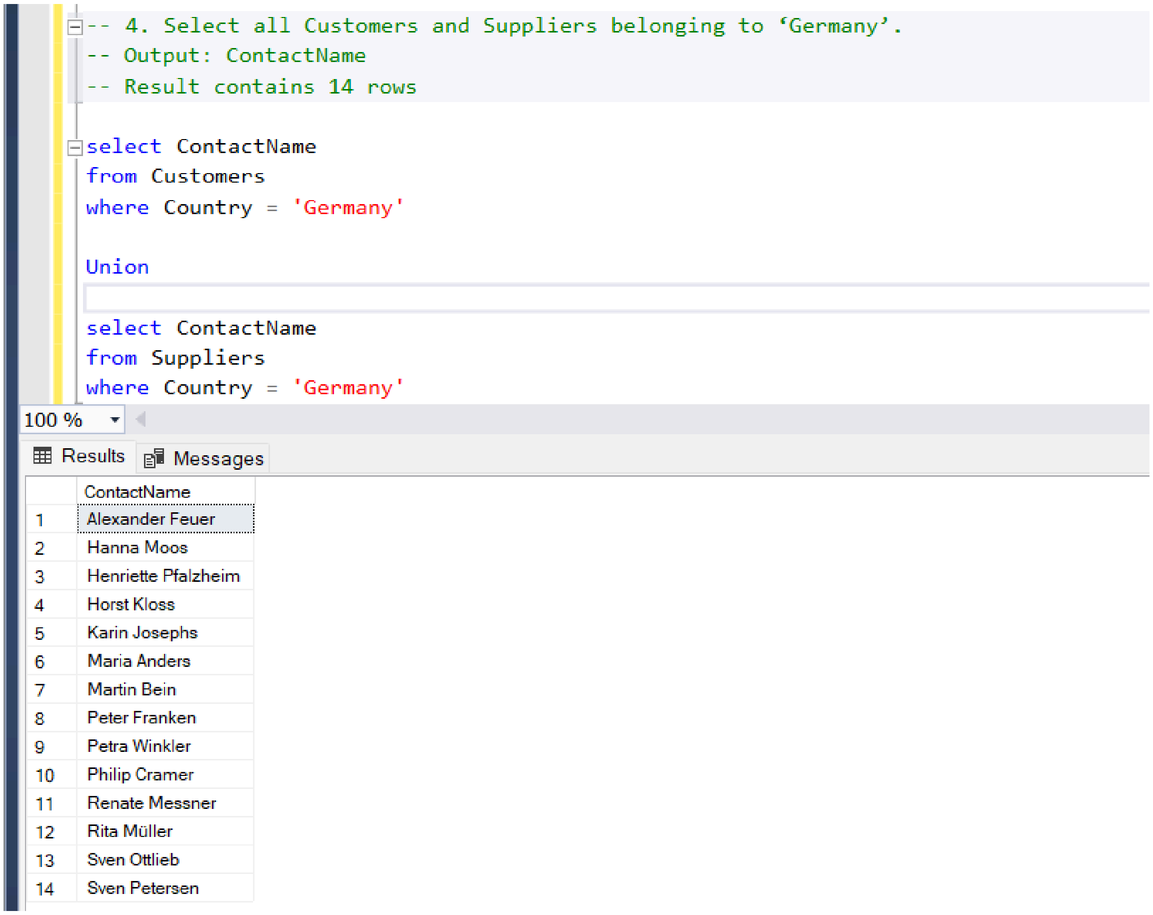
- **Output:** ContactName
- **Result:** 14 rows

Databases - T1

Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi

07201, Fall 2025



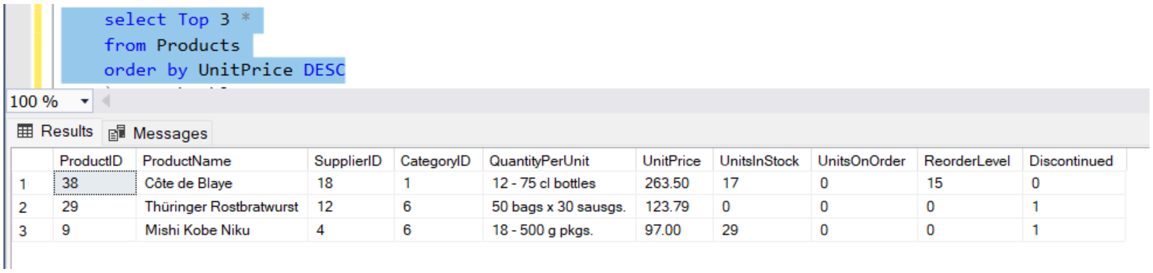
```
-- 4. Select all Customers and Suppliers belonging to 'Germany'.
-- Output: ContactName
-- Result contains 14 rows

select ContactName
from Customers
where Country = 'Germany'

Union

select ContactName
from Suppliers
where Country = 'Germany'
```

	ContactName
1	Alexander Feuer
2	Hanna Moos
3	Henriette Pfalzheim
4	Horst Kloss
5	Karin Josephs
6	Maria Anders
7	Martin Bein
8	Peter Franken
9	Petra Winkler
10	Philip Cramer
11	Renate Messner
12	Rita Müller
13	Sven Ottlieb
14	Sven Petersen

5. Find the 3rd most expensive product in the database.

- **Output:** ProductName
- **Result:** 1 row

Subquery:



```
select Top 3 *
from Products
order by UnitPrice DESC
```

	ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	ReorderLevel	Discontinued
1	38	Côte de Blaye	18	1	12 - 75 cl bottles	263.50	17	0	15	0
2	29	Thüringer Rostbratwurst	12	6	50 bags x 30 sausgs.	123.79	0	0	0	1
3	9	Mishi Kobe Niku	4	6	18 - 500 g pkgs.	97.00	29	0	0	1

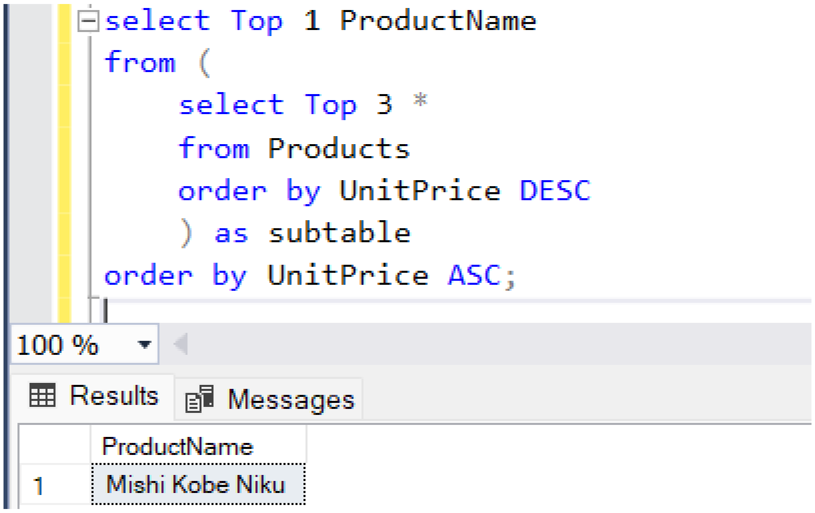
Overall Query:

Databases - T1

Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi

07201, Fall 2025



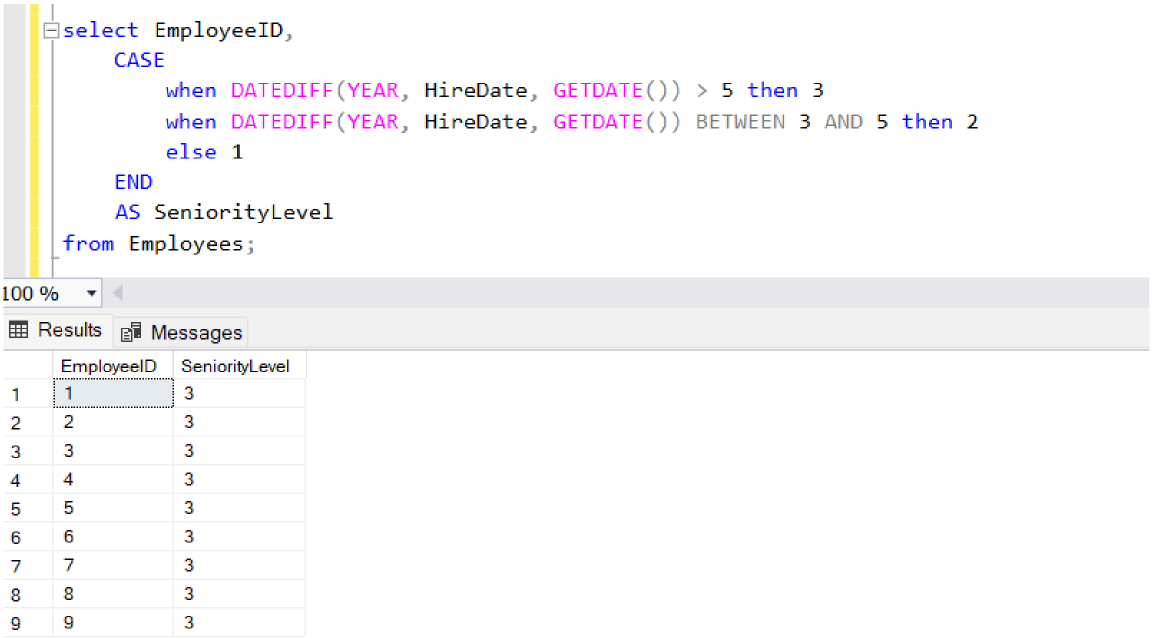
```sql
select Top 1 ProductName
from (
    select Top 3 *
    from Products
    order by UnitPrice DESC
    ) as subtable
order by UnitPrice ASC;
```

	ProductName
1	Mishi Kobe Niku

6. Select all employees and their Seniority level:

Seniority level = 3 *if employee has been with the company for more than 5 years.*
Seniority level = 2 *if employee has been with the company from 3-5 years.*
Seniority level = 1 *if employee has been with the company for less than 3 years.*

- **Output:** EmployeeID, SeniorityLevel
- **Result:** 9 rows



```sql
select EmployeeID,
    CASE
        when DATEDIFF(YEAR, HireDate, GETDATE()) > 5 then 3
        when DATEDIFF(YEAR, HireDate, GETDATE()) BETWEEN 3 AND 5 then 2
        else 1
    END
    AS SeniorityLevel
from Employees;
```

	EmployeeID	SeniorityLevel
1	1	3
2	2	3
3	3	3
4	4	3
5	5	3
6	6	3
7	7	3
8	8	3
9	9	3

Databases - T1

Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi
07201, Fall 2025

7. List all products and their types which indicate if they are:

Costly *(unit price > 80)*
Economical *(unit price between 30 and 80)*
Cheap *(unit price < 30)*

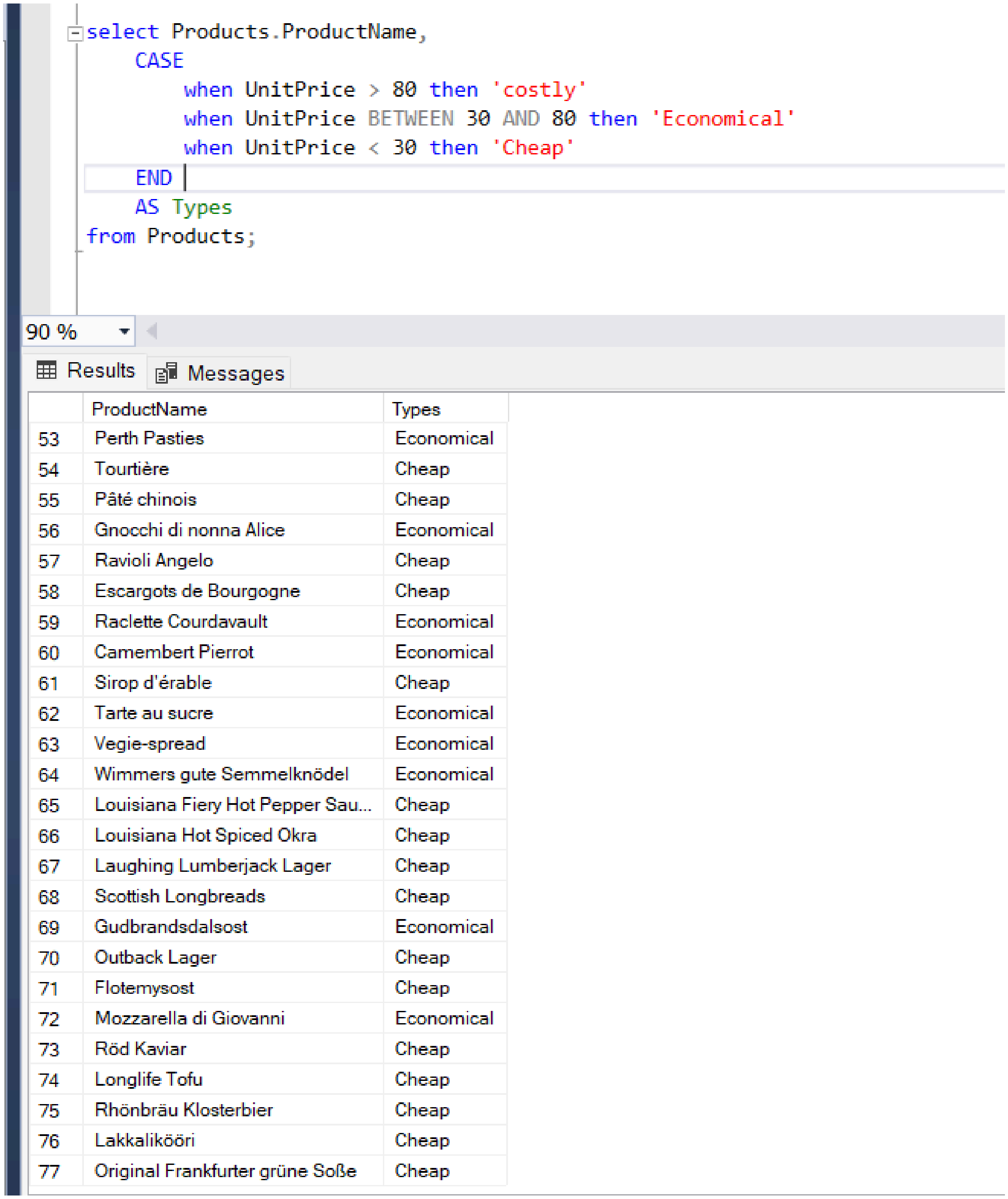
- **Output:** ProductName, Types
- **Result:** 77 rows

Databases - T1

Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi

07201, Fall 2025



```sql
select Products.ProductName,
    CASE
        when UnitPrice > 80 then 'costly'
        when UnitPrice BETWEEN 30 AND 80 then 'Economical'
        when UnitPrice < 30 then 'Cheap'
    END
    AS Types
from Products;
```

	ProductName	Types
53	Perth Pasties	Economical
54	Tourtière	Cheap
55	Pâté chinois	Cheap
56	Gnocchi di nonna Alice	Economical
57	Ravioli Angelo	Cheap
58	Escargots de Bourgogne	Cheap
59	Raclette Courdavault	Economical
60	Camembert Pierrot	Economical
61	Sirop d'érable	Cheap
62	Tarte au sucre	Economical
63	Vegie-spread	Economical
64	Wimmers gute Semmelknödel	Economical
65	Louisiana Fiery Hot Pepper Sau...	Cheap
66	Louisiana Hot Spiced Okra	Cheap
67	Laughing Lumberjack Lager	Cheap
68	Scottish Longbreads	Cheap
69	Gudbrandsdalsost	Economical
70	Outback Lager	Cheap
71	Flotemysost	Cheap
72	Mozzarella di Giovanni	Economical
73	Röd Kaviar	Cheap
74	Longlife Tofu	Cheap
75	Rhönbräu Klosterbier	Cheap
76	Lakkalikööri	Cheap
77	Original Frankfurter grüne Soße	Cheap

8. List all products and their trends based on the number of orders placed in the year 1997.

Databases - T1

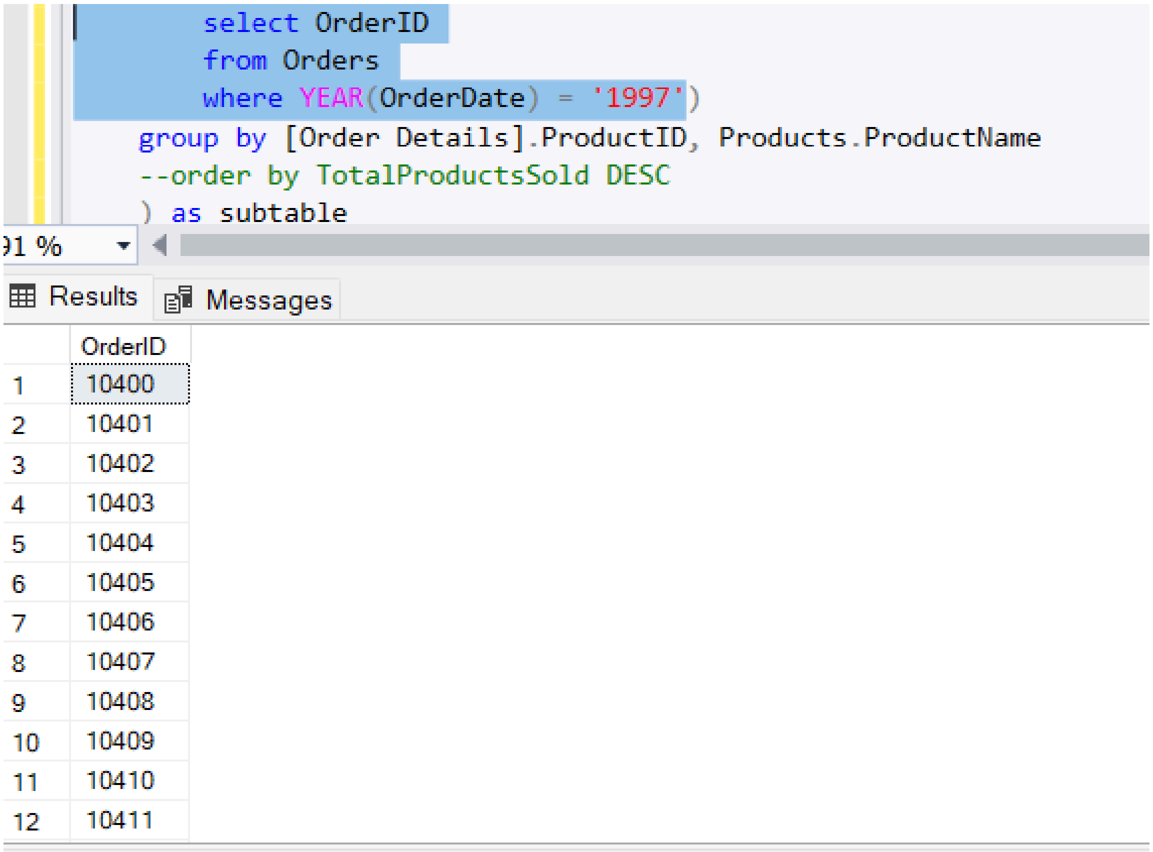
Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi

07201, Fall 2025

- If no. of orders **>= 50**, Trend = Customer favourite
- If 30 <= no. of orders <= 49, Trend = Trending
- If 10 <= no. of orders <= 29, Trend = On the rise
- Else, Trend = Not popular
- **Output:** ProductName, Trend
- **Result:** 77 rows

Subquery: tells me all the orderIDs that were placed in 1996



```
        select OrderID
        from Orders
        where YEAR(OrderDate) = '1997')
    group by [Order Details].ProductID, Products.ProductName
    --order by TotalProductsSold DESC
    ) as subtable
```

	OrderID
1	10400
2	10401
3	10402
4	10403
5	10404
6	10405
7	10406
8	10407
9	10408
10	10409
11	10410
12	10411

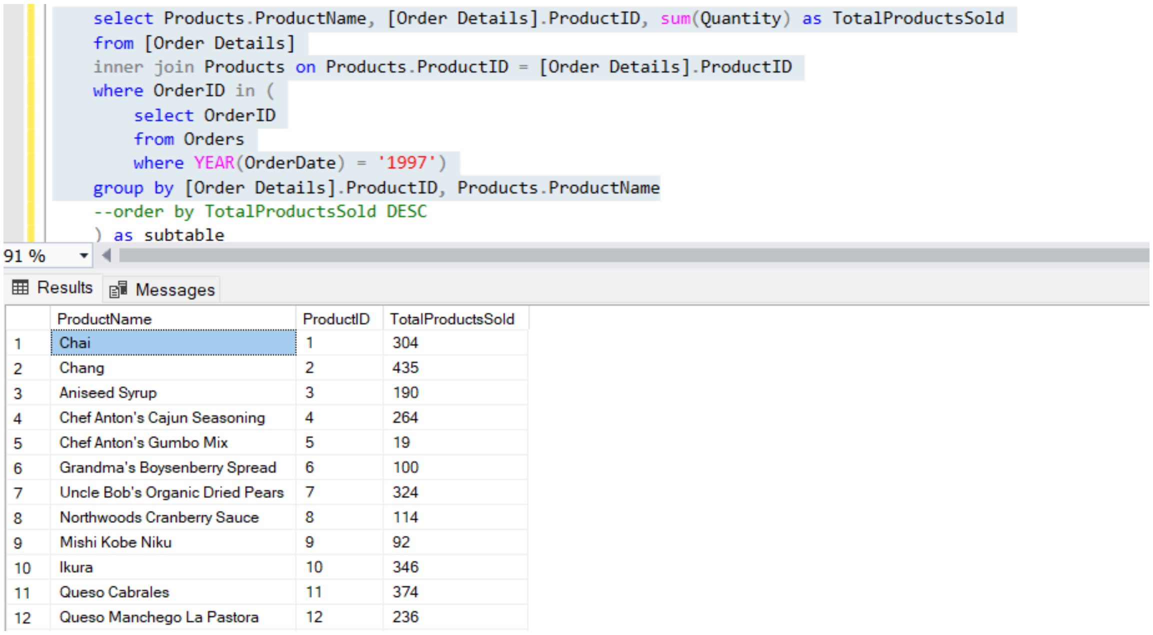
Upper query: This finds out how much each product was sold

Databases - T1

Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi

07201, Fall 2025



```sql
select Products.ProductName, [Order Details].ProductID, sum(Quantity) as TotalProductsSold
from [Order Details]
inner join Products on Products.ProductID = [Order Details].ProductID
where OrderID in (
    select OrderID
    from Orders
    where YEAR(OrderDate) = '1997')
group by [Order Details].ProductID, Products.ProductName
--order by TotalProductsSold DESC
) as subtable
```

	ProductName	ProductID	TotalProductsSold
1	Chai	1	304
2	Chang	2	435
3	Aniseed Syrup	3	190
4	Chef Anton's Cajun Seasoning	4	264
5	Chef Anton's Gumbo Mix	5	19
6	Grandma's Boysenberry Spread	6	100
7	Uncle Bob's Organic Dried Pears	7	324
8	Northwoods Cranberry Sauce	8	114
9	Mishi Kobe Niku	9	92
10	Ikura	10	346
11	Queso Cabrales	11	374
12	Queso Manchego La Pastora	12	236

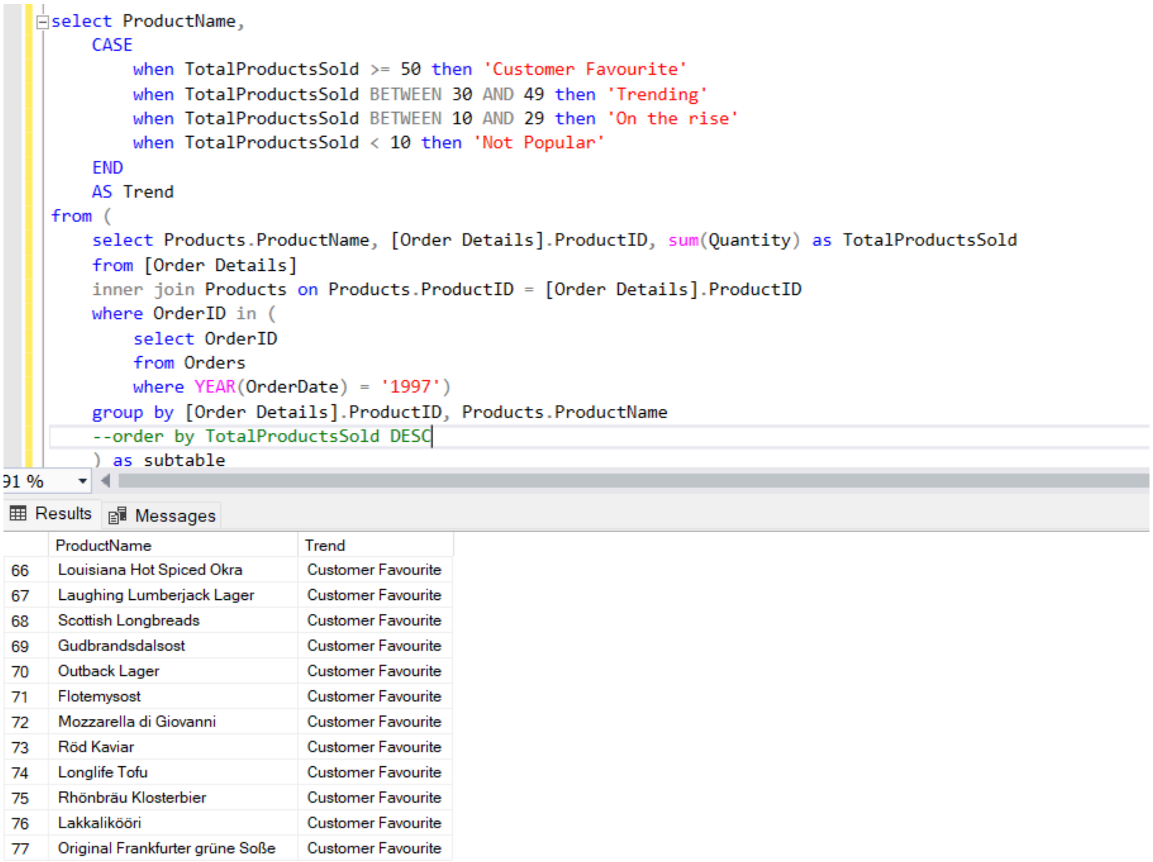
Final Query:

Databases - T1

Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi

07201, Fall 2025



```sql
select ProductName,
    CASE
        when TotalProductsSold >= 50 then 'Customer Favourite'
        when TotalProductsSold BETWEEN 30 AND 49 then 'Trending'
        when TotalProductsSold BETWEEN 10 AND 29 then 'On the rise'
        when TotalProductsSold < 10 then 'Not Popular'
    END
    AS Trend
from (
    select Products.ProductName, [Order Details].ProductID, sum(Quantity) as TotalProductsSold
    from [Order Details]
    inner join Products on Products.ProductID = [Order Details].ProductID
    where OrderID in (
        select OrderID
        from Orders
        where YEAR(OrderDate) = '1997')
    group by [Order Details].ProductID, Products.ProductName
    --order by TotalProductsSold DESC
    ) as subtable
```

	ProductName	Trend
66	Louisiana Hot Spiced Okra	Customer Favourite
67	Laughing Lumberjack Lager	Customer Favourite
68	Scottish Longbreads	Customer Favourite
69	Gudbrandsdalsost	Customer Favourite
70	Outback Lager	Customer Favourite
71	Flotemysost	Customer Favourite
72	Mozzarella di Giovanni	Customer Favourite
73	Röd Kaviar	Customer Favourite
74	Longlife Tofu	Customer Favourite
75	Rhönbräu Klosterbier	Customer Favourite
76	Lakkalikööri	Customer Favourite
77	Original Frankfurter grüne Soße	Customer Favourite

9. Find the total number of orders placed by each customer.

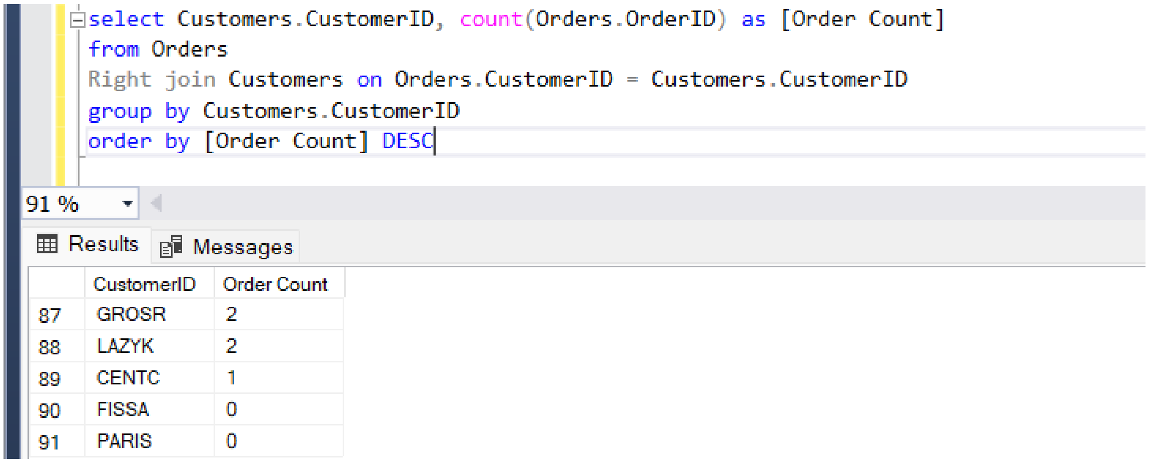
- **Output:** CustomerID
- **Result:** 91 rows

Databases - T1

Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi

07201, Fall 2025

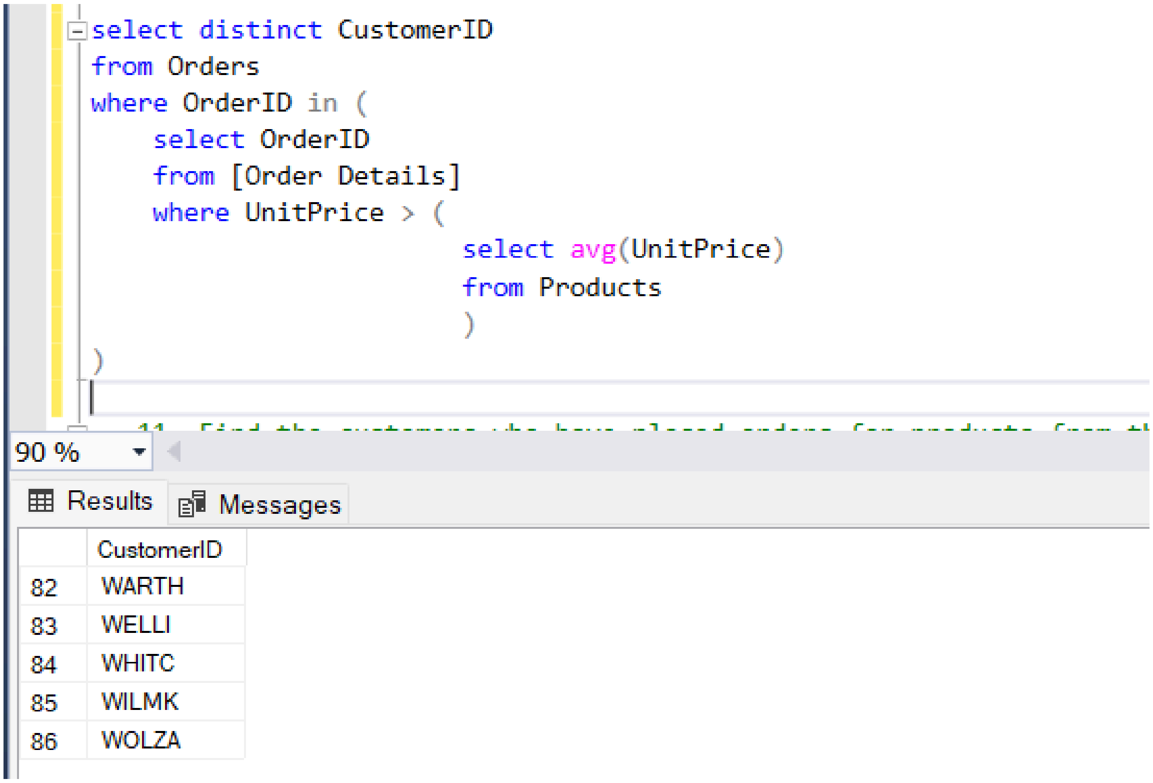


```sql
select Customers.CustomerID, count(Orders.OrderID) as [Order Count]
from Orders
Right join Customers on Orders.CustomerID = Customers.CustomerID
group by Customers.CustomerID
order by [Order Count] DESC
```

	CustomerID	Order Count
87	GROSR	2
88	LAZYK	2
89	CENTC	1
90	FISSA	0
91	PARIS	0

10. Retrieve customers who have placed orders for products with a price higher than the average price of all products.

- **Output:** CustomerID
- **Result:** 86 rows



```sql
select distinct CustomerID
from Orders
where OrderID in (
     select OrderID
     from [Order Details]
     where UnitPrice > (
                         select avg(UnitPrice)
                         from Products
                         )
)

  11. Find the customers who have placed orders for products from th
```

	CustomerID
82	WARTH
83	WELLI
84	WHITC
85	WILMK
86	WOLZA

Databases - T1

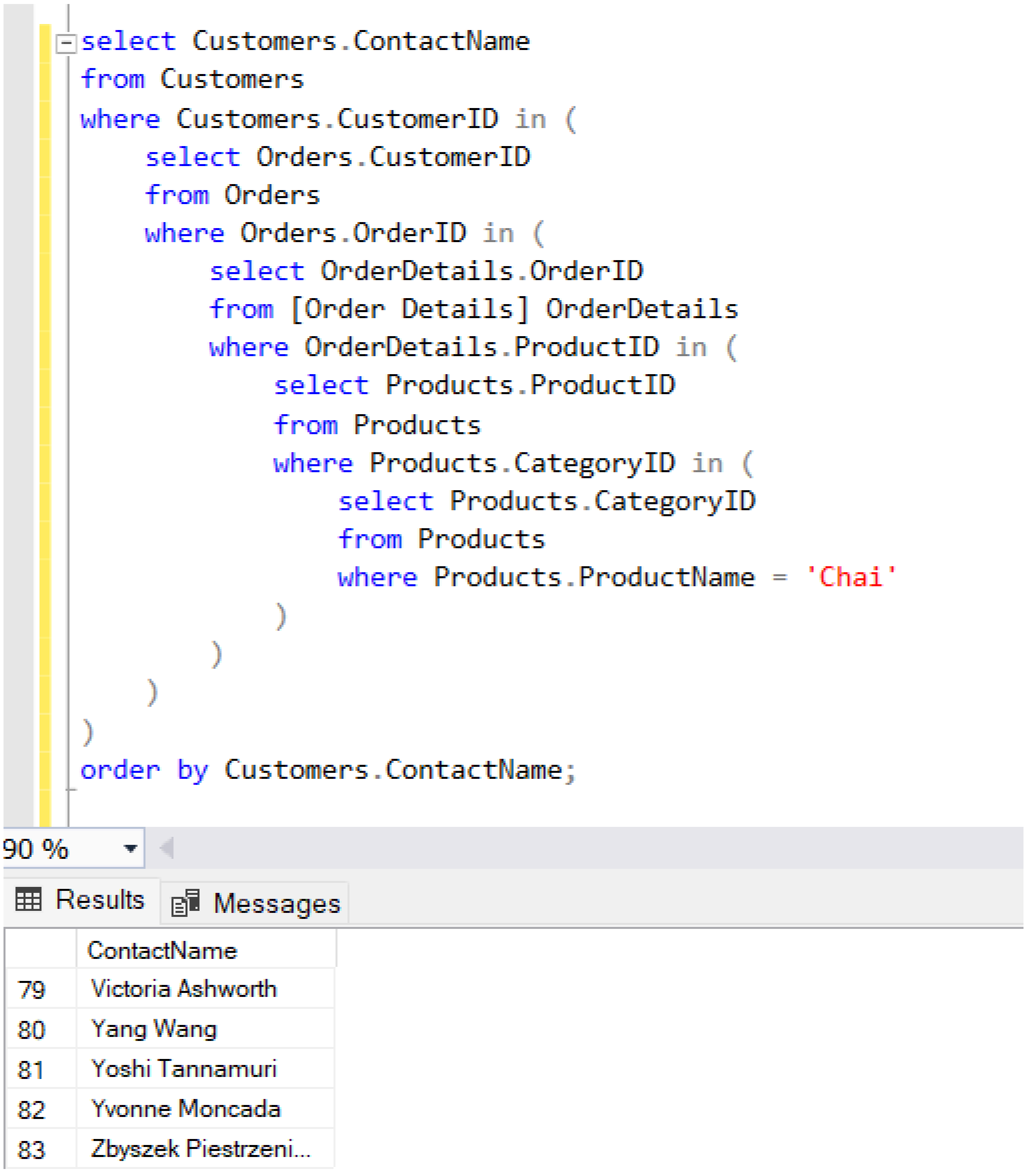
Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi

07201, Fall 2025

11. Find the customers who have placed orders for products from the same category as ‘Chai’.

- **Output:** Customers.ContactName
- **Result:** 83 rows

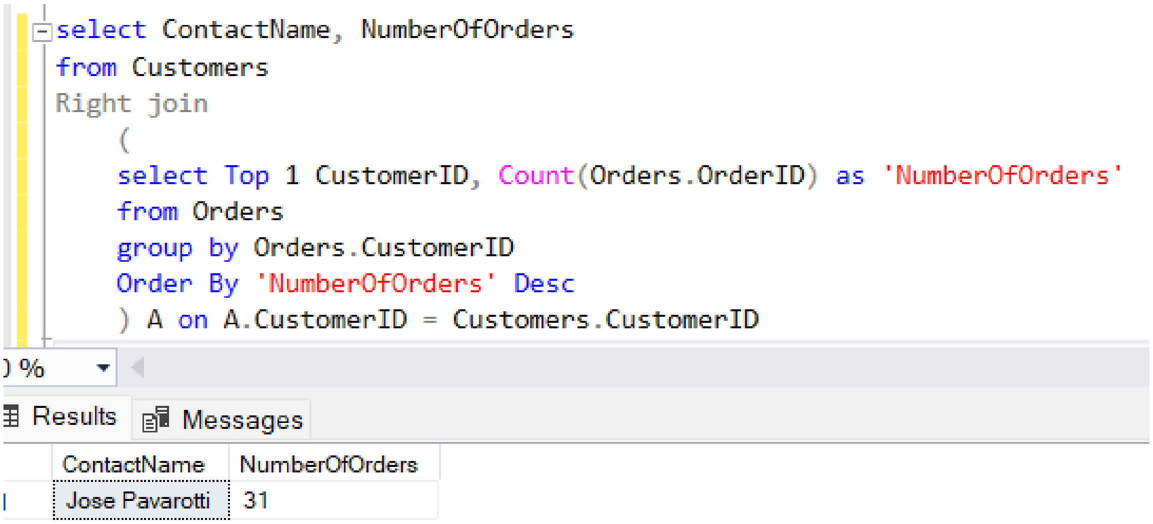


```sql
select Customers.ContactName
from Customers
where Customers.CustomerID in (
    select Orders.CustomerID
    from Orders
    where Orders.OrderID in (
        select OrderDetails.OrderID
        from [Order Details] OrderDetails
        where OrderDetails.ProductID in (
            select Products.ProductID
            from Products
            where Products.CategoryID in (
                select Products.CategoryID
                from Products
                where Products.ProductName = 'Chai'
            )
        )
    )
)
order by Customers.ContactName;
```

	ContactName
79	Victoria Ashworth
80	Yang Wang
81	Yoshi Tannamuri
82	Yvonne Moncada
83	Zbyszek Piestrzeni...

12. Find the customer who has placed the highest total number of orders.

- **Output:** ContactName, NumberOfOrders
- **Result:** 1 row



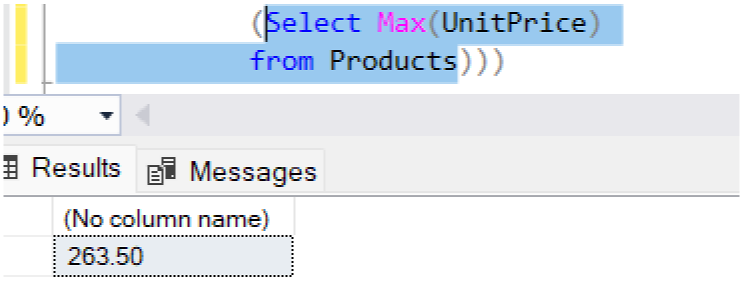
```
select ContactName, NumberOfOrders
from Customers
Right join
    (
    select Top 1 CustomerID, Count(Orders.OrderID) as 'NumberOfOrders'
    from Orders
    group by Orders.CustomerID
    Order By 'NumberOfOrders' Desc
    ) A on A.CustomerID = Customers.CustomerID
```

ContactName	NumberOfOrders
Jose Pavarotti	31

13 .List all the customers who have placed an order for the most expensive product.

- **Output:** ContactName
- **Result:** 12 rows

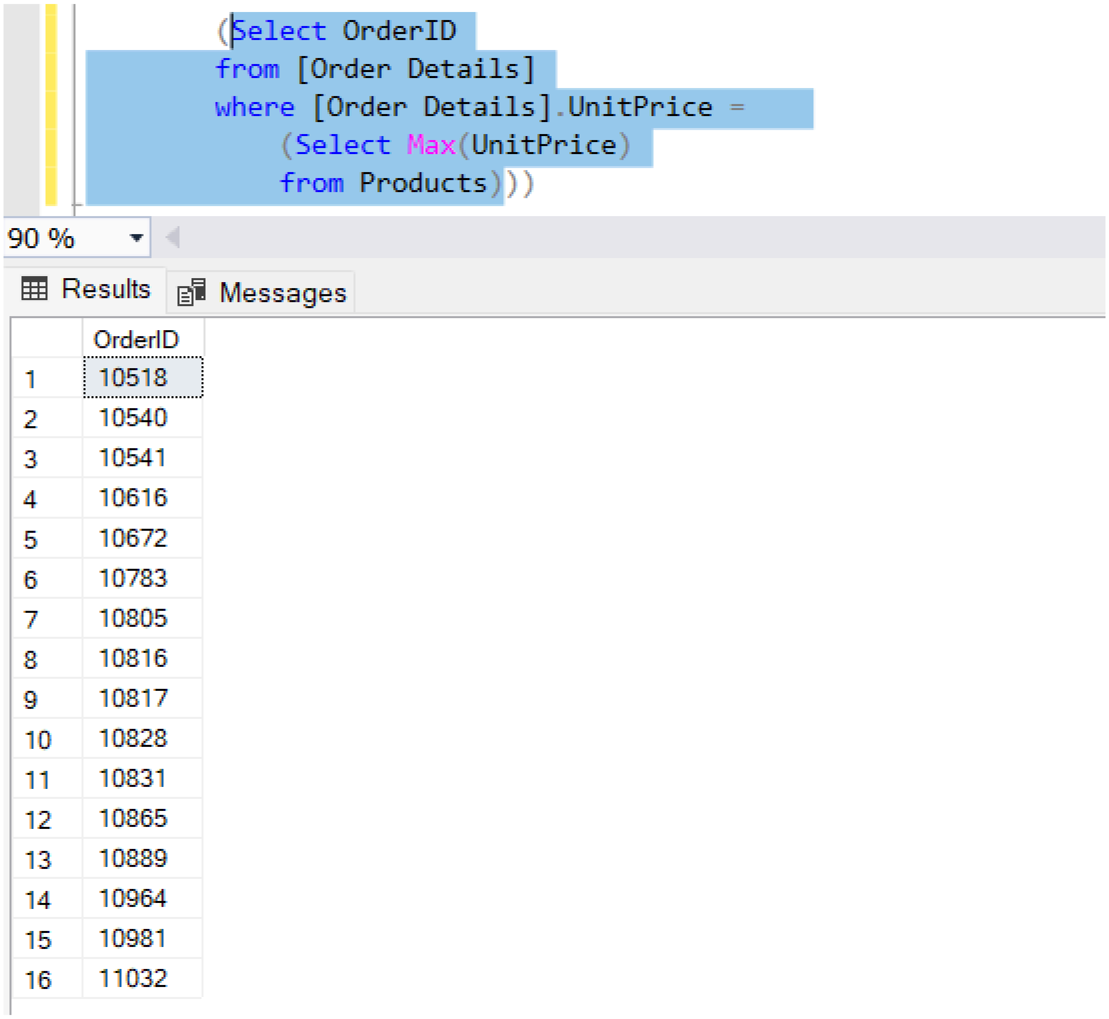
Subqueries:



```
            (Select Max(UnitPrice)
            from Products)))
```

(No column name)
263.50

Second level subquery:



```
         (Select OrderID
         from [Order Details]
         where [Order Details].UnitPrice =
             (Select Max(UnitPrice)
             from Products)))
```

	OrderID
1	10518
2	10540
3	10541
4	10616
5	10672
6	10783
7	10805
8	10816
9	10817
10	10828
11	10831
12	10865
13	10889
14	10964
15	10981
16	11032

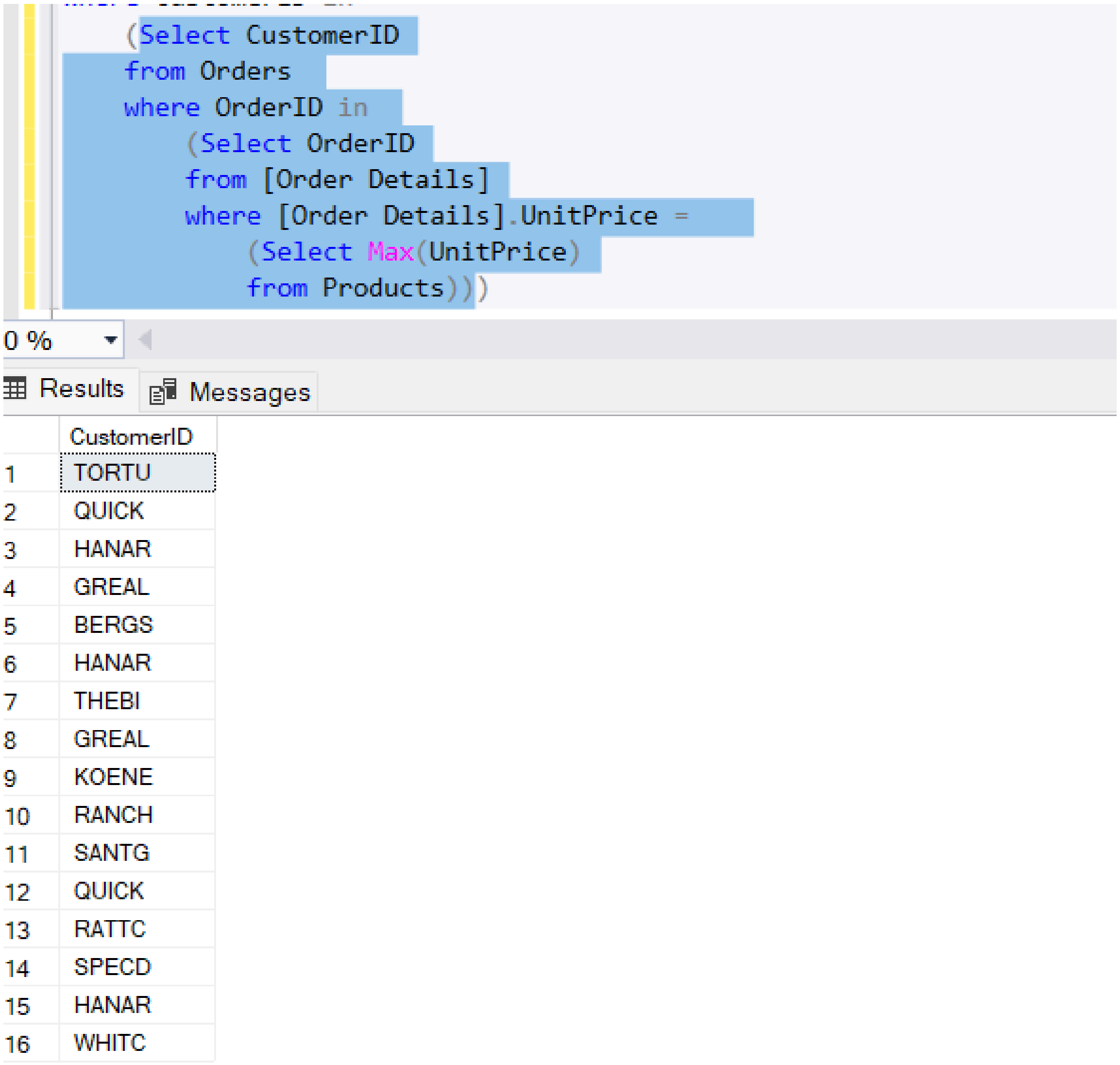
Third Level Subquery:

Databases - T1

Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi

07201, Fall 2025



```
    (Select CustomerID
    from Orders
    where OrderID in
        (Select OrderID
        from [Order Details]
        where [Order Details].UnitPrice =
            (Select Max(UnitPrice)
            from Products)))
```

	CustomerID
1	TORTU
2	QUICK
3	HANAR
4	GREAL
5	BERGS
6	HANAR
7	THEBI
8	GREAL
9	KOENE
10	RANCH
11	SANTG
12	QUICK
13	RATTC
14	SPECD
15	HANAR
16	WHITC

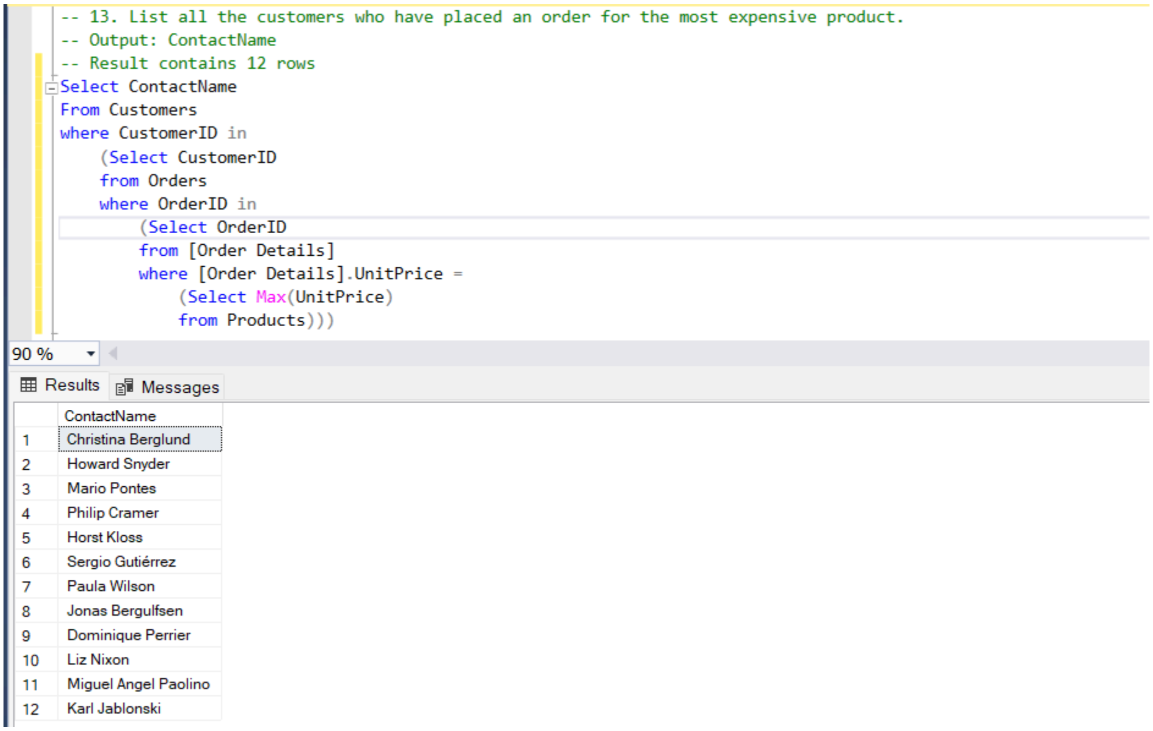
Overall Query:

Databases - T1

Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi

07201, Fall 2025



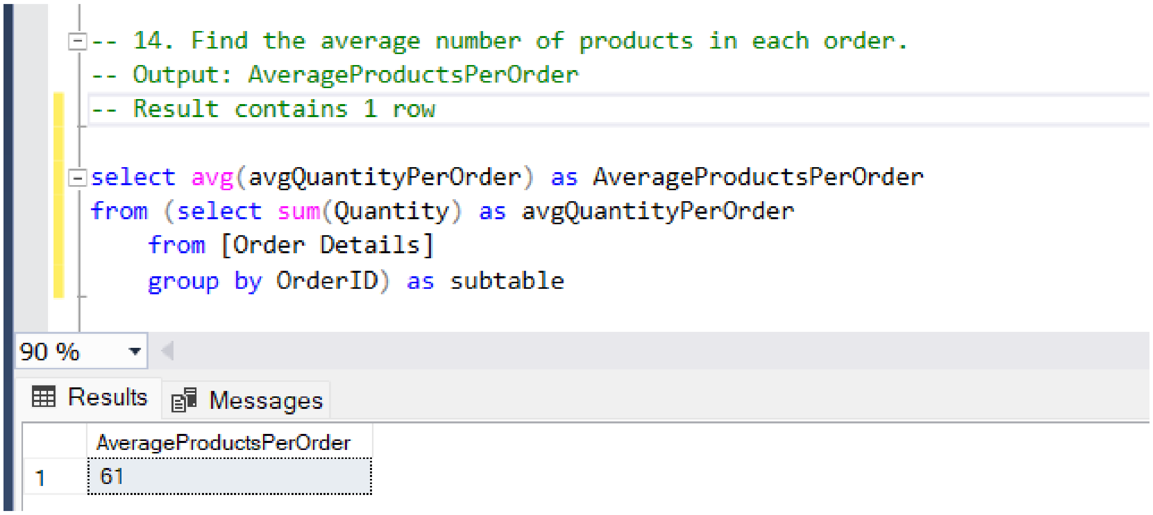
```sql
-- 13. List all the customers who have placed an order for the most expensive product.
-- Output: ContactName
-- Result contains 12 rows
Select ContactName
From Customers
where CustomerID in
    (Select CustomerID
    from Orders
    where OrderID in
        (Select OrderID
        from [Order Details]
        where [Order Details].UnitPrice =
            (Select Max(UnitPrice)
            from Products)))
```

	ContactName
1	Christina Berglund
2	Howard Snyder
3	Mario Pontes
4	Philip Cramer
5	Horst Kloss
6	Sergio Gutiérrez
7	Paula Wilson
8	Jonas Bergulfsen
9	Dominique Perrier
10	Liz Nixon
11	Miguel Angel Paolino
12	Karl Jablonski

14. Find the average number of products in each order.

- **Output:** AverageProductsPerOrder
- **Result:** 1 row

average quantity of all products ordered per order



```sql
-- 14. Find the average number of products in each order.
-- Output: AverageProductsPerOrder
-- Result contains 1 row

select avg(avgQuantityPerOrder) as AverageProductsPerOrder
from (select sum(Quantity) as avgQuantityPerOrder
    from [Order Details]
    group by OrderID) as subtable
```

	AverageProductsPerOrder
1	61

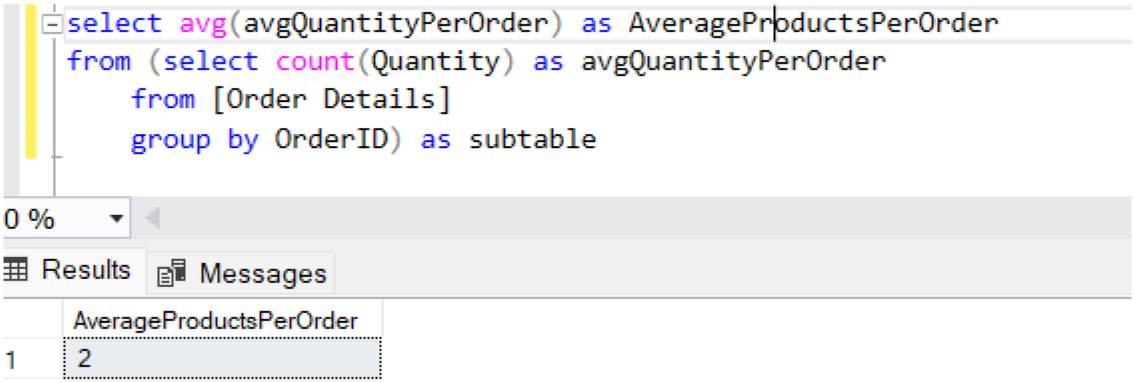
Databases - T1

Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi

07201, Fall 2025

There was abit of confusion, if question 14 wanted to ask me ***average number of distinct products per order*** then you will do the following

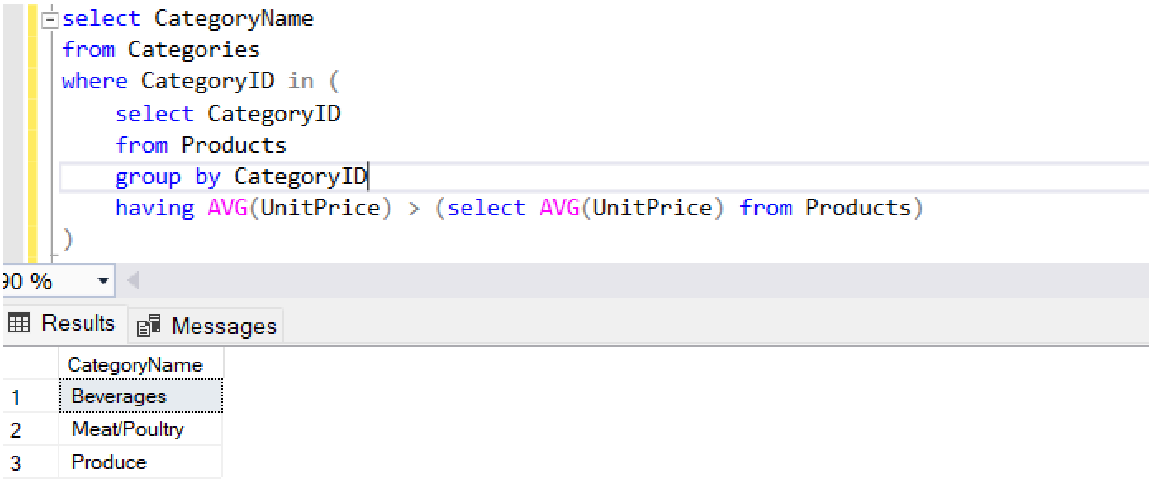


```sql
select avg(avgQuantityPerOrder) as AverageProductsPerOrder
from (select count(Quantity) as avgQuantityPerOrder
    from [Order Details]
    group by OrderID) as subtable
```

	AverageProductsPerOrder
1	2

15. Find the categories where the average product price is higher than the overall average product price.

- **Output:** CategoryName
- **Result:** 3 rows



```sql
select CategoryName
from Categories
where CategoryID in (
    select CategoryID
    from Products
    group by CategoryID
    having AVG(UnitPrice) > (select AVG(UnitPrice) from Products)
)
```

	CategoryName
1	Beverages
2	Meat/Poultry
3	Produce

16. Find the product which has the second highest price.

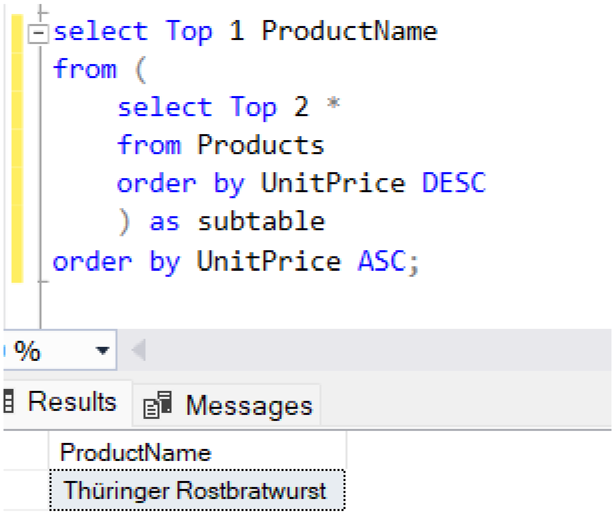
- **Output:** ProductName, UnitPrice
- **Result:** 1 row

Databases - T1

Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi

07201, Fall 2025

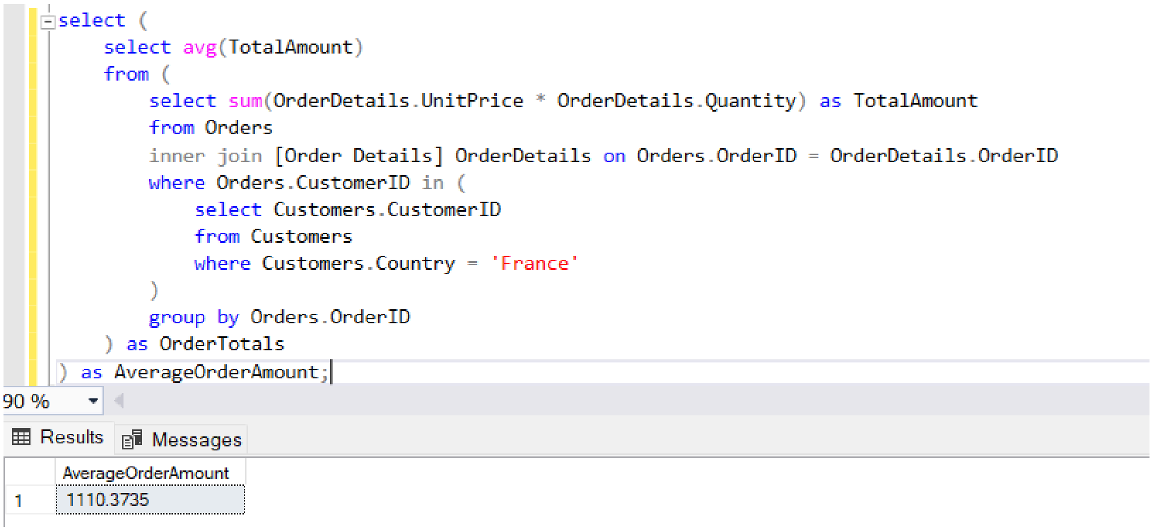


```
select Top 1 ProductName
from (
    select Top 2 *
    from Products
    order by UnitPrice DESC
    ) as subtable
order by UnitPrice ASC;
```

```
ProductName
Thüringer Rostbratwurst
```

17. Find the average order amount for customers from France.

- **Output:** AverageOrderAmount
- **Result:** 1 row



```
select (
    select avg(TotalAmount)
    from (
        select sum(OrderDetails.UnitPrice * OrderDetails.Quantity) as TotalAmount
        from Orders
        inner join [Order Details] OrderDetails on Orders.OrderID = OrderDetails.OrderID
        where Orders.CustomerID in (
            select Customers.CustomerID
            from Customers
            where Customers.Country = 'France'
        )
        group by Orders.OrderID
    ) as OrderTotals
) as AverageOrderAmount;
```

```
   AverageOrderAmount
1  1110.3735
```

Databases - T1

Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi

07201, Fall 2025

CODE

```sql
-- 1. Find the employee who processed the first order placed in the year 1998.
-- Output: EmployeeID
-- Result contains 10 rows

SELECT Top 1 EmployeeID
FROM Orders
WHERE OrderDate = (
      SELECT MIN(OrderDate)
      FROM Orders
      WHERE YEAR(OrderDate) = 1998
);

-- 2. Select all employees who work directly under the top manager of the company.
-- Output: EmployeeID
-- Result contains 5 rows

select Employees.EmployeeID
from Employees
where ReportsTo = (
      select Employees.EmployeeID
      from Employees
      where ReportsTo is null
);


-- 3. Select all employees who are assigned to territories in the 'Western' and 'Eastern' regions from the Region Table.
-- Output: EmployeeID
-- Result contains 6 rows

select distinct EmployeeTerritories.EmployeeID
from EmployeeTerritories
where EmployeeTerritories.TerritoryID IN (
      select Territories.TerritoryID
      from Territories
      where Territories.RegionID IN (
      select RegionID
      from Region
      where RegionDescription = 'Western' or RegionDescription =
'Eastern'
      )
);

-- 4. Select all Customers and Suppliers belonging to 'Germany'.
-- Output: ContactName
-- Result contains 14 rows

select ContactName
from Customers
where Country = 'Germany'

Union

select ContactName
from Suppliers
where Country = 'Germany'


-- 5. Find the 3rd most expensive product in the database.
-- Output: ProductName
-- Result contains 1 row

select Top 1 ProductName
from (
      select Top 3 *
      from Products
      order by UnitPrice DESC
      ) as subtable
order by UnitPrice ASC;

-- 6. Select all employees and their Seniority level:
-- • Seniority level = 3 if employee has been with the company for
more than 5 years.
-- • Seniority level = 2 if employee has been with the company from
3-5 years.
```

Databases - T1

Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi

07201, Fall 2025

```
-- • Seniority level = 1 if employee has been with the company for
less than 3 years.
-- Output: EmployeeID, SeniorityLevel
-- Result contains 9 rows

select EmployeeID,
      CASE
      when DATEDIFF(YEAR, HireDate, GETDATE()) > 5 then 3
      when DATEDIFF(YEAR, HireDate, GETDATE()) BETWEEN 3 AND 5 then 2
      else 1
      END
      AS SeniorityLevel
from Employees;


-- 7. List all products and their types which indicate if they are:
-- • Costly (unit price > 80)
-- • Economical (unit price between 30 and 80)
-- • Cheap (unit price < 30)
-- Output: ProductName, Types
-- Result contains 77 rows

select Products.ProductName,
      CASE
      when UnitPrice > 80 then 'costly'
      when UnitPrice BETWEEN 30 AND 80 then 'Economical'
      when UnitPrice < 30 then 'Cheap'
      END
      AS Types
from Products;




-- 8. List all products and their trends based on the number of
orders placed in the year 1997.
-- • Trend = Customer favourite if no. of orders >= 50
-- • Trend = Trending if 30 <= no. of orders <= 49
-- • Trend = On the rise if 10 <= no. of orders <= 29
-- • Trend = Not popular if no. of orders < 10
```

Databases - T1

Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi
07201, Fall 2025

```
-- Output: ProductName, Trend
-- Result contains 77 rows

select ProductName,
      CASE
      when TotalProductsSold >= 50 then 'Customer Favourite'
      when TotalProductsSold BETWEEN 30 AND 49 then 'Trending'
      when TotalProductsSold BETWEEN 10 AND 29 then 'On the rise'
      when TotalProductsSold < 10 then 'Not Popular'
      END
      AS Trend
from (
      select Products.ProductName, [Order Details].ProductID,
sum(Quantity) as TotalProductsSold
      from [Order Details]
      inner join Products on Products.ProductID = [Order
Details].ProductID
      where OrderID in (
      select OrderID
      from Orders
      where YEAR(OrderDate) = '1997')
      group by [Order Details].ProductID, Products.ProductName
      --order by TotalProductsSold DESC
      ) as subtable

-- 9. Find the total number of orders placed by each customer.
-- Output: CustomerID
-- Result contains 91 rows

select Customers.CustomerID, count(Orders.OrderID) as [Order Count]
from Orders
Right join Customers on Orders.CustomerID = Customers.CustomerID
group by Customers.CustomerID
order by [Order Count] DESC


-- 10. Retrieve customers who have placed orders for products with a
price higher than the average price of all products.
-- Output: CustomerID
```

Databases - T1

Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi

07201, Fall 2025

```sql
-- Result contains 86 rows

select distinct CustomerID
from Orders
where OrderID in (
      select OrderID
      from [Order Details]
      where UnitPrice > (
                        select avg(UnitPrice)
                        from Products
                        )
)

-- 11. Find the customers who have placed orders for products from the same category as 'Chai'.
-- Output: Customers.ContactName
-- Result contains 83 rows

select Customers.ContactName
from Customers
where Customers.CustomerID in (
      select Orders.CustomerID
      from Orders
      where Orders.OrderID in (
      select OrderDetails.OrderID
      from [Order Details] OrderDetails
      where OrderDetails.ProductID in (
            select Products.ProductID
            from Products
            where Products.CategoryID in (
                  select Products.CategoryID
                  from Products
                  where Products.ProductName = 'Chai'
            )
      )
      )
)
order by Customers.ContactName;
```

Databases - T1

Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi
07201, Fall 2025

```
-- 12. Find the customer who has placed the highest total number of
orders.
-- Output: ContactName, NumberOfOrders
-- Result contains 1 row

select ContactName, NumberOfOrders
from Customers
Right join
      (
      select Top 1 CustomerID, Count(Orders.OrderID) as
'NumberOfOrders'
      from Orders
      group by Orders.CustomerID
      Order By 'NumberOfOrders' Desc
      ) A on A.CustomerID = Customers.CustomerID


-- 13. List all the customers who have placed an order for the most
expensive product.
-- Output: ContactName
-- Result contains 12 rows
Select ContactName
From Customers
where CustomerID in
      (Select CustomerID
      from Orders
      where OrderID in
      (Select OrderID
      from [Order Details]
      where [Order Details].UnitPrice =
            (Select Max(UnitPrice)
            from Products)))

-- 14. Find the average number of products in each order.
-- Output: AverageProductsPerOrder
-- Result contains 1 row
```

Databases - T1

Lab 10: SQL Sub Queries

Syed Asghar Abbas Zaidi

07201, Fall 2025

```
select avg(avgQuantityPerOrder) as AverageProductsPerOrder
from (select count(Quantity) as avgQuantityPerOrder
      from [Order Details]
      group by OrderID) as subtable

-- 15. Find the categories where the average product price is higher
than the overall average product price.
-- Output: CategoryName
-- Result contains 3 rows
select CategoryName
from Categories
where CategoryID in (
      select CategoryID
      from Products
      group by CategoryID
      having AVG(UnitPrice) > (select AVG(UnitPrice) from Products)
)

-- 16. Find the product which has the second highest price.
-- Output: ProductName, UnitPrice

-- Result contains 1 row
select Top 1 ProductName
from (
      select Top 2 *
      from Products
      order by UnitPrice DESC
      ) as subtable
order by UnitPrice ASC;


-- 17. Find the average order amount for customers from France.
-- Output: AverageOrderAmount
-- Result contains 1 row

select (
      select avg(TotalAmount)
      from (
      select sum(OrderDetails.UnitPrice * OrderDetails.Quantity) as
```

```sql
TotalAmount
      from Orders
      inner join [Order Details] OrderDetails on Orders.OrderID =
OrderDetails.OrderID
      where Orders.CustomerID in (
            select Customers.CustomerID
            from Customers
            where Customers.Country = 'France'
      )
      group by Orders.OrderID
      ) as OrderTotals
) as AverageOrderAmount;
```