

# **DataBase System**

## **Lab-6**



Session 2022 - 2026

Abdul Rehman 2022-CS-79

**Submitted To:**

Mr. Nazeef Ul Haq

Department of Computer Science  
**University of Engineering and Technology Lahore Pakistan**

# Contents

<b>1 Lab-6 Tasks:</b>	<b>3</b>
1.1 Return customers and their orders, including customers who placed no orders (CustomerID, OrderID, OrderDate) . . . . .	3
1.2 Report only those customer IDs who never placed any order. (CustomerID, OrderID, OrderDate) . . . . .	4
1.3 Report those customers who placed orders on July,1997. (CustomerID, OrderID, OrderDate) . . . . .	5
1.4 Report the total orders of each customer. (customerID, totalorders) . . . . .	6
1.5 Write a query to generate a five copies of each employee. (EmployeeID, FirstName, LastName) . . . . .	7
1.6 List all the products whose price is more than average price. . . . .	8
1.7 Find the second highest price of product. . . . .	9
1.8 Write a query that returns a row for each employee and day in the range 04-07-1996 through 04-08-1997. (EmployeeID, Date) . . . . .	10
1.9 Return US customers, and for each customer return the total number of orders and total quantities. (CustomerID, Totalorders, totalquantity) . . . . .	11
1.10 Write a query that returns all customers in the output, but matches them with their respective orders only if they were placed on July 04,1997. (CustomerID, CompanyName, OrderID, Orderdate) . . . . .	12
1.11 Are there any employees who are older than their managers? . . . . .	13
1.12 List that names of those employees and their ages. (EmployeeName, Age, Manager Age) . . . . .	14
1.13 List the names of products which were ordered on 8th August 1997. (ProductName, OrderDate) . . . . .	15
1.14 List the addresses, cities, countries of all orders which were serviced by Anne and were shipped late. (Address, City, Country) . . . . .	16
1.15 List all countries to which beverages have been shipped. (Country) . . . . .	17

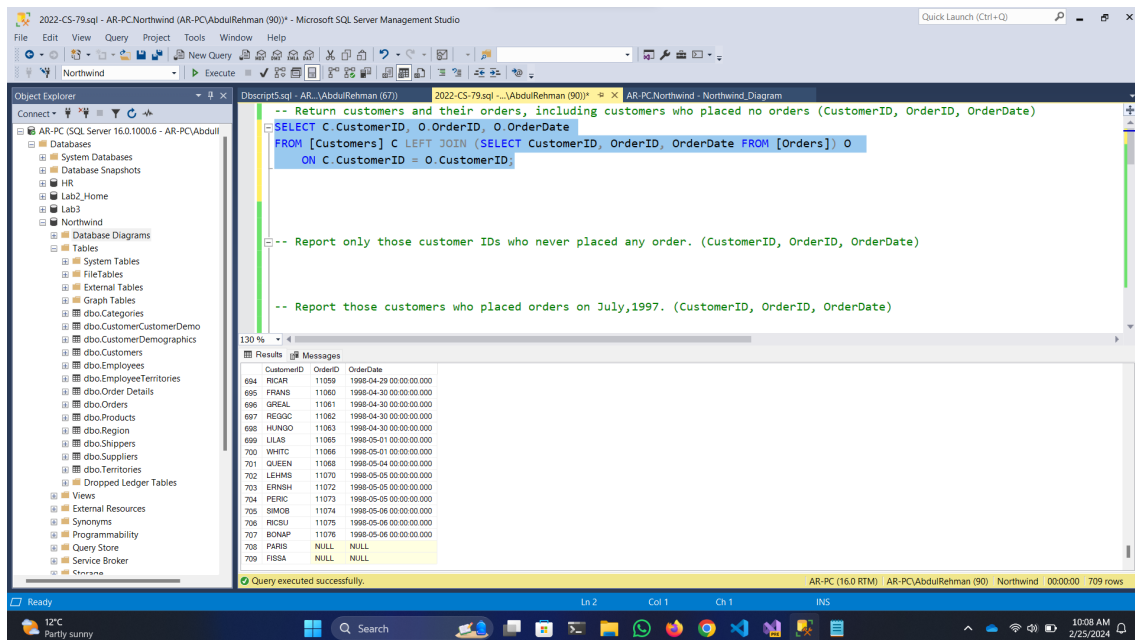
# 1 Lab-6 Tasks:

## 1.1 Return customers and their orders, including customers who placed no orders (CustomerID, OrderID, OrderDate)

### SQL Query:

```
SELECT C.CustomerID, O.OrderID, O.OrderDate
FROM [Customers] C
LEFT JOIN (SELECT CustomerID, OrderID, OrderDate FROM [Orders]) O
ON C.CustomerID = O.CustomerID;
```

### Result:



The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The left pane shows the 'Object Explorer' with the 'Northwind' database selected. The right pane shows a query window with the following SQL code:

```
-- Return customers and their orders, including customers who placed no orders (CustomerID, OrderID, OrderDate)
SELECT C.CustomerID, O.OrderID, O.OrderDate
FROM [Customers] C LEFT JOIN (SELECT CustomerID, OrderID, OrderDate FROM [Orders]) O
ON C.CustomerID = O.CustomerID;

-- Report only those customer IDs who never placed any order. (CustomerID, OrderID, OrderDate)

-- Report those customers who placed orders on July,1997. (CustomerID, OrderID, OrderDate)
```

The 'Results' pane shows the output of the query, displaying columns 'CustomerID', 'OrderID', and 'OrderDate'. The data is as follows:

CustomerID	OrderID	OrderDate
694	RICAR	11059 1998-04-29 00:00:00.000
695	FRANS	11060 1998-04-30 00:00:00.000
696	GREAL	11061 1998-04-30 00:00:00.000
697	RESIDC	11062 1998-04-30 00:00:00.000
698	HUNDO	11063 1998-04-30 00:00:00.000
699	LILAS	11065 1998-05-01 00:00:00.000
700	WHETC	11066 1998-05-01 00:00:00.000
701	QUEEN	11068 1998-05-04 00:00:00.000
702	LEHMS	11070 1998-05-05 00:00:00.000
703	ERNSH	11072 1998-05-05 00:00:00.000
704	PERIC	11073 1998-05-05 00:00:00.000
705	SHMOB	11074 1998-05-05 00:00:00.000
706	RICSU	11075 1998-05-06 00:00:00.000
707	BONAP	11076 1998-05-06 00:00:00.000
708	PARIS	NULL NULL
709	FOSIA	NULL NULL

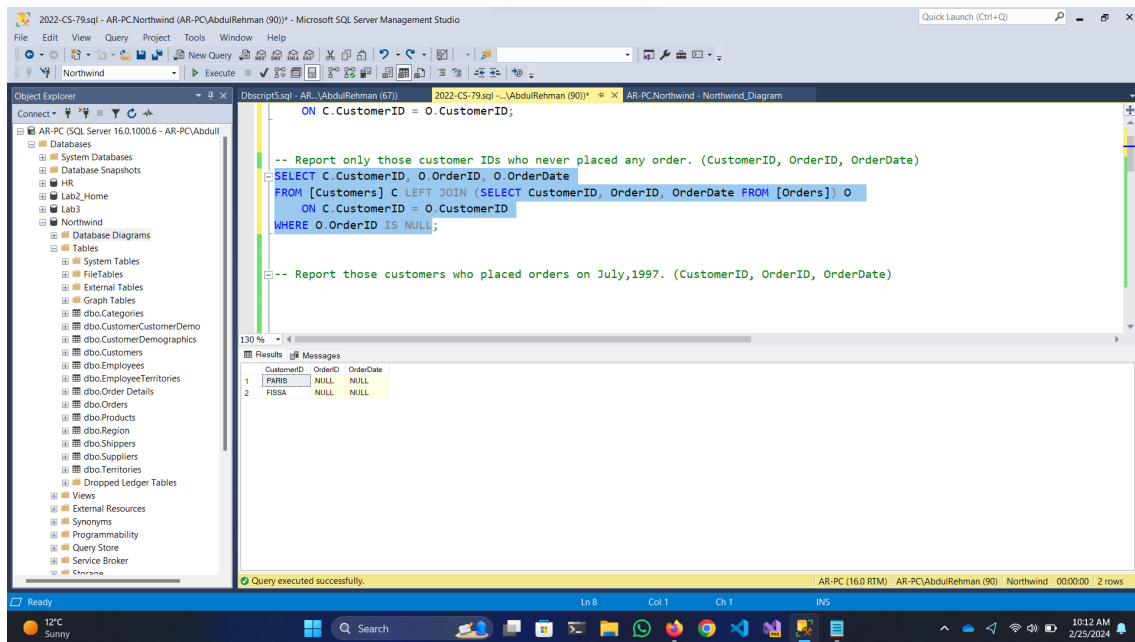
The status bar at the bottom indicates 'Query executed successfully.' and '709 rows'.

## 1.2 Report only those customer IDs who never placed any order. (CustomerID, OrderID, OrderDate)

### SQL Query:

```
SELECT C.CustomerID, O.OrderID, O.OrderDate
FROM [Customers] C
LEFT JOIN (SELECT CustomerID, OrderID, OrderDate FROM [Orders]) O
ON C.CustomerID = O.CustomerID
WHERE O.OrderID IS NULL;
```

### Result:



The screenshot displays the Microsoft SQL Server Management Studio interface. The query editor shows the following SQL query:

```
-- Report only those customer IDs who never placed any order. (CustomerID, OrderID, OrderDate)
SELECT C.CustomerID, O.OrderID, O.OrderDate
FROM [Customers] C LEFT JOIN (SELECT CustomerID, OrderID, OrderDate FROM [Orders]) O
ON C.CustomerID = O.CustomerID
WHERE O.OrderID IS NULL;

-- Report those customers who placed orders on July,1997. (CustomerID, OrderID, OrderDate)
```

The Results pane shows the output of the query:

CustomerID	OrderID	OrderDate
1	PAWS	NULL
2	FISSA	NULL

The status bar at the bottom indicates "Query executed successfully." and "2 rows".

### 1.3 Report those customers who placed orders on July,1997. (CustomerID, OrderID, OrderDate)

#### SQL Query:

```
SELECT C.CustomerID, O.OrderID, O.OrderDate
FROM [Customers] C , (
    SELECT CustomerID, OrderID, OrderDate FROM [Orders]) O
WHERE C.CustomerID = O.CustomerID
    AND (MONTH(O.OrderDate) = '7' AND YEAR(O.OrderDate) = '1997')
```

#### Result:

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'Northwind' database selected. The right pane shows a query window with the following SQL code:

```
-- Report only those customer IDs who never placed any order. (CustomerID, OrderID, OrderDate)
SELECT C.CustomerID, O.OrderID, O.OrderDate
FROM [Customers] C LEFT JOIN (SELECT CustomerID, OrderID, OrderDate FROM [Orders]) O
ON C.CustomerID = O.CustomerID
WHERE O.OrderID IS NULL;

-- Report those customers who placed orders on July,1997. (CustomerID, OrderID, OrderDate)
SELECT C.CustomerID, O.OrderID, O.OrderDate
FROM [Customers] C ,(SELECT CustomerID, OrderID, OrderDate FROM [Orders]) O
WHERE C.CustomerID = O.CustomerID AND (MONTH(O.OrderDate) = '7' AND YEAR(O.OrderDate) = '1997');

-- Report the total orders of each customer. (customerID, totalorders)
```

The 'Results' pane shows the output of the query, displaying a list of customers who placed orders in July 1997. The results are as follows:

CustomerID	OrderID	OrderDate
1	WELLI	1997-07-01 00:00:00.000
2	REGIOC	1997-07-02 00:00:00.000
3	QUICK	1997-07-03 00:00:00.000
4	GREAL	1997-07-04 00:00:00.000
5	MCTECP	1997-07-07 00:00:00.000
6	LEHMAS	1997-07-08 00:00:00.000
7	LEHMAS	1997-07-09 00:00:00.000
8	OLDWO	1997-07-09 00:00:00.000
9	ERNSHA	1997-07-10 00:00:00.000
10	WHITC	1997-07-11 00:00:00.000
11	PICOO	1997-07-11 00:00:00.000
12	BIBREY	1997-07-15 00:00:00.000
13	HUNGC	1997-07-18 00:00:00.000
14	HILAA	1997-07-18 00:00:00.000
15	VANFE	1997-07-17 00:00:00.000
16	SAIEA	1997-07-18 00:00:00.000
17	TBDATM	1997-07-15 00:00:00.000

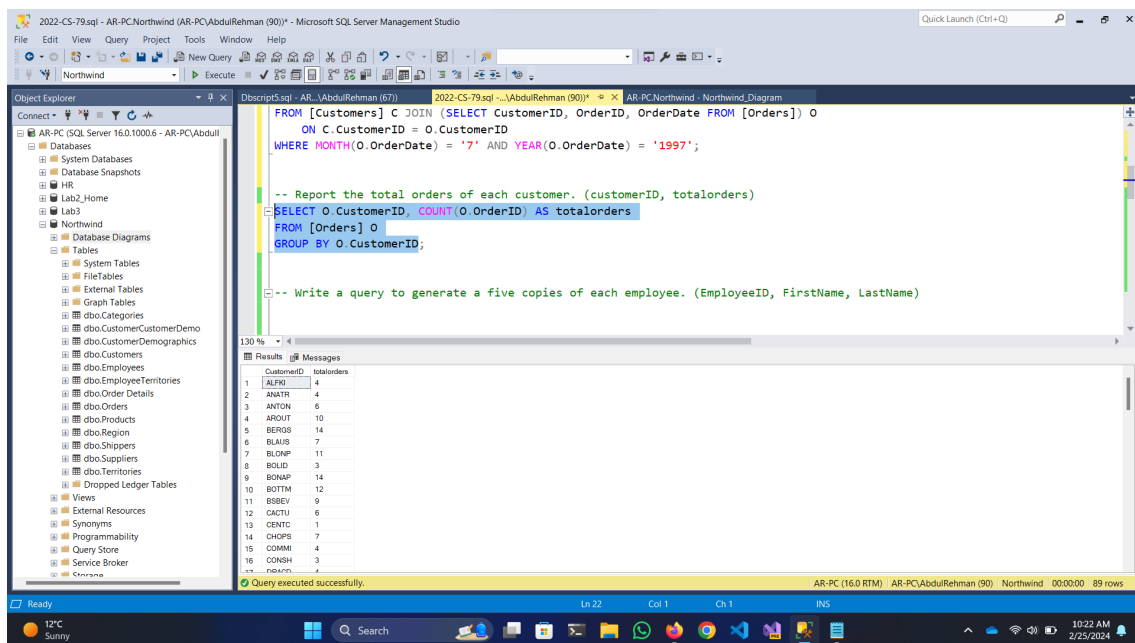
The status bar at the bottom indicates that the query was executed successfully, returning 26 rows.

## 1.4 Report the total orders of each customer. (customerID, totalorders)

### SQL Query:

```
SELECT O.CustomerID, COUNT(O.OrderID) AS totalorders
FROM [Orders] O
GROUP BY O.CustomerID;
```

### Result:



The screenshot displays the Microsoft SQL Server Management Studio interface. The query editor shows a SQL query to report the total orders for each customer. The query is as follows:

```
FROM [Customers] C JOIN (SELECT CustomerID, OrderID, OrderDate FROM [Orders]) O
ON C.CustomerID = O.CustomerID
WHERE MONTH(O.OrderDate) = '7' AND YEAR(O.OrderDate) = '1997';

-- Report the total orders of each customer. (customerID, totalorders)
SELECT O.CustomerID, COUNT(O.OrderID) AS totalorders
FROM [Orders] O
GROUP BY O.CustomerID;

-- Write a query to generate a five copies of each employee. (EmployeeID, FirstName, LastName)
```

The Results pane shows the output of the query, displaying a table with two columns: CustomerID and totalorders. The data is as follows:

CustomerID	totalorders
1 ALFKI	4
2 ANATL	4
3 ANTON	6
4 AROUT	10
5 BERGS	14
6 BLAUS	7
7 BOLNP	11
8 BOLID	3
9 BONAP	14
10 BOTTM	12
11 BSBEV	9
12 CACTU	6
13 CENTC	1
14 CHOPS	7
15 COMMI	4
16 CONSH	3

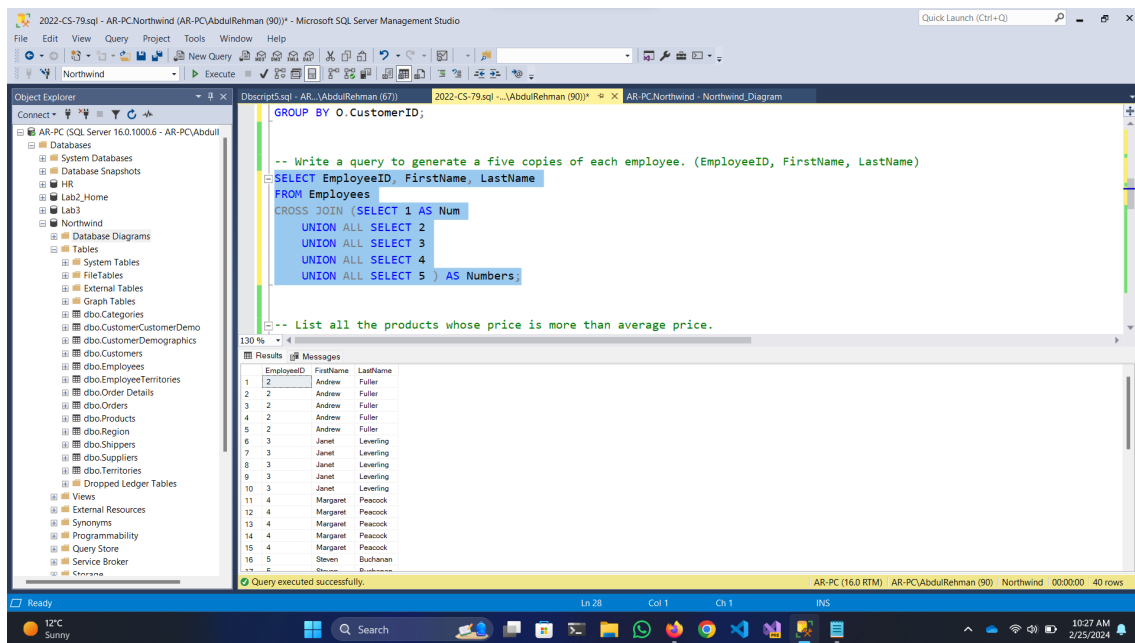
The status bar at the bottom indicates that the query was executed successfully, returning 89 rows.

## 1.5 Write a query to generate a five copies of each employee. (EmployeeID, FirstName, LastName)

### SQL Query:

```
SELECT EmployeeID, FirstName, LastName
FROM Employees
CROSS JOIN (SELECT 1 AS Num
            UNION ALL SELECT 2
            UNION ALL SELECT 3
            UNION ALL SELECT 4
            UNION ALL SELECT 5 ) AS Numbers;
```

### Result:



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL query:

```
-- Write a query to generate a five copies of each employee. (EmployeeID, FirstName, LastName)
SELECT EmployeeID, FirstName, LastName
FROM Employees
CROSS JOIN (SELECT 1 AS Num
            UNION ALL SELECT 2
            UNION ALL SELECT 3
            UNION ALL SELECT 4
            UNION ALL SELECT 5 ) AS Numbers;
```

The query results are displayed in the Results pane, showing 40 rows. The data is as follows:

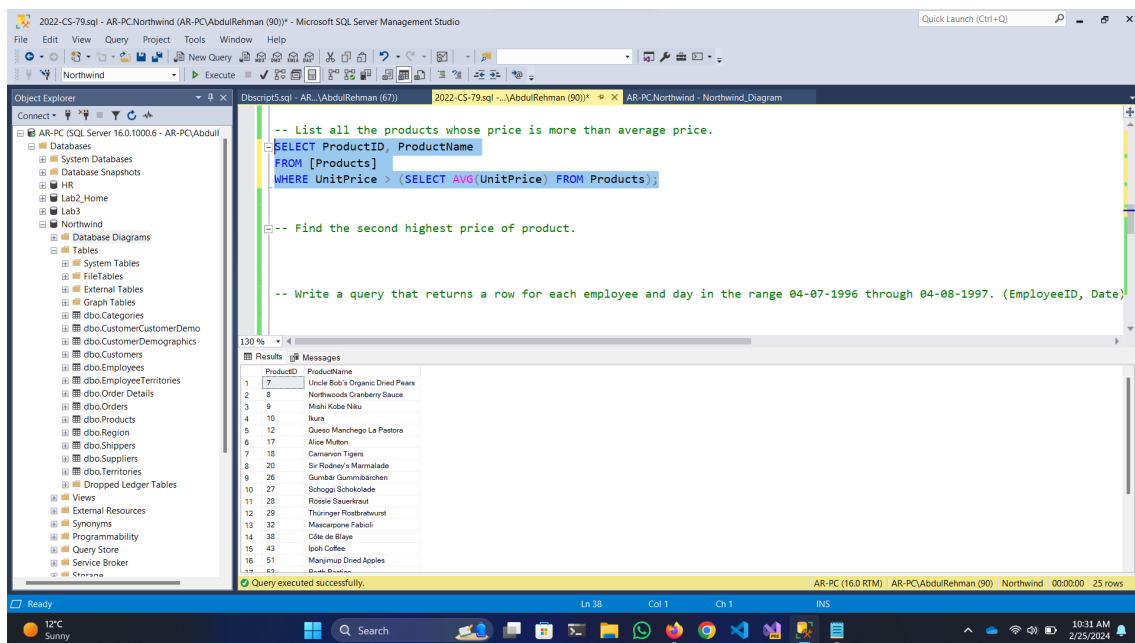
EmployeeID	FirstName	LastName
1	Andrew	Fuller
2	Andrew	Fuller
3	Andrew	Fuller
4	Andrew	Fuller
5	Andrew	Fuller
6	Janet	Leverling
7	Janet	Leverling
8	Janet	Leverling
9	Janet	Leverling
10	Janet	Leverling
11	Margaret	Peacock
12	Margaret	Peacock
13	Margaret	Peacock
14	Margaret	Peacock
15	Margaret	Peacock
16	Steven	Buchanan
17	Steven	Buchanan
18	Steven	Buchanan
19	Steven	Buchanan
20	Steven	Buchanan
21	Andrew	Fuller
22	Andrew	Fuller
23	Andrew	Fuller
24	Andrew	Fuller
25	Andrew	Fuller
26	Janet	Leverling
27	Janet	Leverling
28	Janet	Leverling
29	Janet	Leverling
30	Janet	Leverling
31	Margaret	Peacock
32	Margaret	Peacock
33	Margaret	Peacock
34	Margaret	Peacock
35	Margaret	Peacock
36	Steven	Buchanan
37	Steven	Buchanan
38	Steven	Buchanan
39	Steven	Buchanan
40	Steven	Buchanan

## 1.6 List all the products whose price is more than average price.

### SQL Query:

```
SELECT ProductID, ProductName
FROM [Products]
WHERE UnitPrice > (SELECT AVG(UnitPrice) FROM Products);
```

### Result:



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL query:

```
-- List all the products whose price is more than average price.
SELECT ProductID, ProductName
FROM [Products]
WHERE UnitPrice > (SELECT AVG(UnitPrice) FROM Products);

-- Find the second highest price of product.

-- Write a query that returns a row for each employee and day in the range 04-07-1996 through 04-08-1997. (EmployeeID, Date)
```

The query has been executed successfully, and the results are displayed in the Results pane. The results show a list of products with their ProductID and ProductName.

ProductID	ProductName
7	Uncle Bob's Organic Dried Pears
8	Northwind Cranberry Sauce
9	Mishi Kobe Niku
10	Ikura
12	Quepao Menchego Le Pastora
17	Aloa Mutton
18	Camaron Tiger
20	Sir Rodney's Marmalade
26	Gumbier Gummibärchen
27	Schoggi Schokolade
28	Rössle Sauerkraut
29	Thüringer Rostbratwurst
32	Mascarpone Fabioli
38	Côte de Boeuf
43	Iso Coffee
51	Mangup Dried Apples

The status bar at the bottom indicates that the query was executed successfully, returning 25 rows.

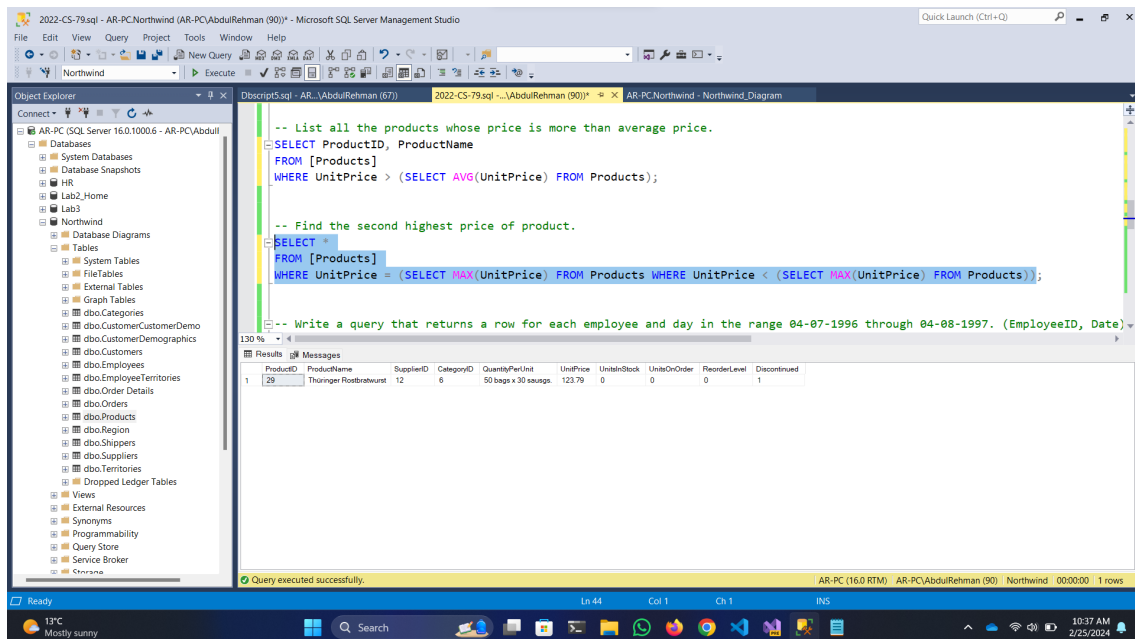


## 1.7 Find the second highest price of product.

### SQL Query:

```
SELECT *
FROM [Products]
WHERE UnitPrice = (
    SELECT MAX(UnitPrice) FROM Products WHERE UnitPrice < (
        SELECT MAX(UnitPrice) FROM Products));
```

### Result:



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL code:

```
-- List all the products whose price is more than average price.
SELECT ProductID, ProductName
FROM [Products]
WHERE UnitPrice > (SELECT AVG(UnitPrice) FROM Products);

-- Find the second highest price of product.
SELECT *
FROM [Products]
WHERE UnitPrice = (SELECT MAX(UnitPrice) FROM Products WHERE UnitPrice < (SELECT MAX(UnitPrice) FROM Products));

-- Write a query that returns a row for each employee and day in the range 04-07-1996 through 04-08-1997. (EmployeeID, Date)
```

The Results pane shows the output of the second query, displaying a single row for the product 'Thoringer Rostbratwurst'.

ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	ReorderLevel	Discontinued
29	Thoringer Rostbratwurst	12	6	50 bags x 30 sausgs.	123.79	0	0	0	1

The status bar at the bottom indicates that the query was executed successfully, returning 1 row.

## 1.8 Write a query that returns a row for each employee and day in the range 04-07-1996 through 04-08-1997. (EmployeeID, Date)

### SQL Query:

```
SELECT E.EmployeeID, CAST(O.OrderDate AS Date) AS Date
FROM [Employees] E, (
    SELECT EmployeeID, OrderDate FROM [Orders]) AS O
WHERE E.EmployeeID = O.EmployeeID
    AND (O.OrderDate >= '1996-07-04'
        AND O.OrderDate <= '1997-08-04')
ORDER BY Date;
```

### Result:

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL query:

```
-- Write a query that returns a row for each employee and day in the range 04-07-1996 through 04-08-1997. (EmployeeID, Date)
SELECT E.EmployeeID, CAST(O.OrderDate AS Date) AS Date
FROM [Employees] E, (SELECT EmployeeID, OrderDate FROM [Orders]) AS O
WHERE E.EmployeeID = O.EmployeeID AND (O.OrderDate >= '1996-07-04' AND O.OrderDate <= '1997-08-04')
ORDER BY Date;
```

The query was executed successfully, and the results are shown in the Results pane. The results table has two columns: EmployeeID and Date. The data is as follows:

EmployeeID	Date
1	1996-07-04
2	1996-07-05
3	1996-07-06
4	1996-07-07
5	1996-07-08
6	1996-07-09
7	1996-07-10
8	1996-07-11
9	1996-07-12
10	1996-07-13
11	1996-07-14
12	1996-07-15
13	1996-07-16
14	1996-07-17
15	1996-07-18
16	1996-07-19

The status bar at the bottom indicates that the query was executed successfully and returned 318 rows.

## 1.9 Return US customers, and for each customer return the total number of orders and total quantities. (CustomerID, Totalorders, totalquantity)

### SQL Query:

```
SELECT C.CustomerID,
       (SELECT COUNT(O.OrderID) FROM [Orders] O
        WHERE O.CustomerID = C.CustomerID) AS TotalOrders,
       (SELECT SUM(OD.Quantity) FROM [Order Details] OD
        WHERE OD.OrderID IN
          (SELECT O.OrderID FROM [Orders] O
           WHERE O.CustomerID = C.CustomerID)) AS TotalQuantity
FROM [Customers] C
WHERE C.Country = 'USA';
```

### Result:

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the SQL query for returning US customers with their total orders and total quantities. The query is executed successfully, and the results are shown in the Results pane.

CustomerID	TotalOrders	TotalQuantity
1. GREAL	10	286
2. HUNGO	4	102
3. LAZYK	1	10
4. LETTS	3	150
5. LONEP	7	103
6. OLDWO	9	493
7. RATTC	13	909
8. SAKKA	25	3181
9. SPLUR	7	283
10. THEBI	3	44
11. THECR	3	56
12. TRASH	3	88
13. WH4TC	13	986

## 1.10 Write a query that returns all customers in the output, but matches them with their respective orders only if they were placed on July 04,1997. (CustomerID, CompanyName, OrderID, Orderdate)

### SQL Query:

```
SELECT C.CustomerID, C.CompanyName, OrderID, Orderdate
FROM [Customers] C
LEFT JOIN (SELECT CustomerID, OrderID, OrderDate FROM [Orders]) O
ON C.CustomerID = O.CustomerID AND O.OrderDate = '1997-07-04';
```

### Result:

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'Northwind' database selected. The right pane shows a query window with the following SQL code:

```
WHERE C.Country = 'USA';

-- Write a query that returns all customers in the output, but matches them with their respective orders only if they were p
=SELECT C.CustomerID, C.CompanyName, OrderID, Orderdate
FROM [Customers] C
LEFT JOIN (SELECT CustomerID, OrderID, OrderDate FROM [Orders]) O
ON C.CustomerID = O.CustomerID AND O.OrderDate = '1997-07-04';

-- Are there any employees who are older than their managers?
```

The 'Results' pane at the bottom shows the output of the query, which is a table with the following columns: CustomerID, CompanyName, OrderID, and Orderdate. The table contains 11 rows of data, all with NULL values for OrderID and Orderdate.

CustomerID	CompanyName	OrderID	Orderdate
76	SUPRID	Suprines delices	NULL
77	THEBI	The Big Cheese	NULL
78	THECR	The Cracker Box	NULL
79	TOMSP	Toni's Spezialiten	NULL
80	TORTU	Tortuga Restaurante	NULL
81	TRADH	Tradigo Hipermercados	NULL
82	TRAFH	Trail's Head Gourmet Provisioners	NULL
83	VAFFE	Vaffeljernet	NULL
84	VICTE	Victualies en stock	NULL
85	VINET	Vins et alcoole Chevalier	NULL
86	WANOK	Die Wandende Kuh	NULL
87	WARTH	Wartens Herku	NULL
88	WELLI	Wellington Importadora	NULL
89	WHITC	White Clover Markets	NULL
90	WILMK	Wilman Kala	NULL
91	WOLZA	Wolski Zajazd	NULL

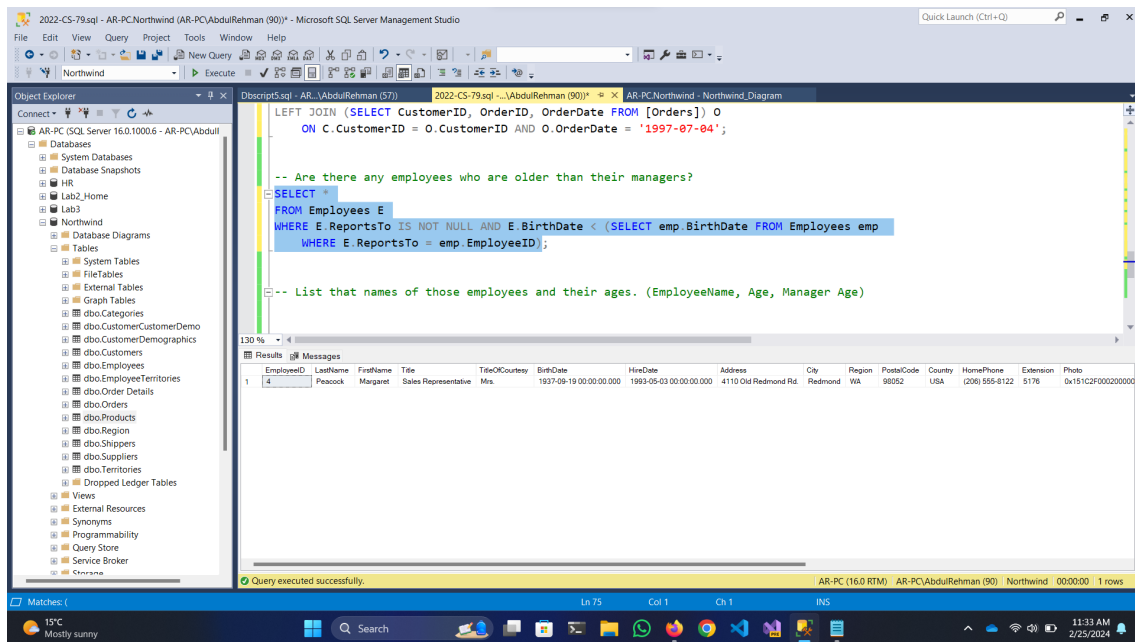
The status bar at the bottom indicates that the query was executed successfully, returning 91 rows.

## 1.11 Are there any employees who are older than their managers?

### SQL Query:

```
SELECT *
FROM Employees E
WHERE E.ReportsTo IS NOT NULL AND E.BirthDate <
      (SELECT emp.BirthDate FROM Employees emp
       WHERE E.ReportsTo = emp.EmployeeID);
```

### Result:



The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The left pane shows the 'Object Explorer' with the 'Northwind' database selected. The right pane shows the 'Query Editor' with the following SQL query:

```
-- Are there any employees who are older than their managers?
SELECT *
FROM Employees E
WHERE E.ReportsTo IS NOT NULL AND E.BirthDate < (SELECT emp.BirthDate FROM Employees emp
WHERE E.ReportsTo = emp.EmployeeID);

-- List that names of those employees and their ages. (EmployeeName, Age, Manager Age)
```

The 'Results' pane shows the output of the query, which is a single row of data for the employee 'Margaret' (EmployeeID 4). The columns displayed are: EmployeeID, LastName, FirstName, Title, Title/Country, BirthDate, HireDate, Address, City, Region, PostalCode, Country, HomePhone, Extension, and Photo.

EmployeeID	LastName	FirstName	Title	Title/Country	BirthDate	HireDate	Address	City	Region	PostalCode	Country	HomePhone	Extension	Photo
4	Peacock	Margaret	Sales Representative	Mrs.	1957-09-19 00:00:00.000	1993-09-03 00:00:00.000	4110 Old Redmond Rd.	Redmond	WA	98052	USA	(206) 555-8122	5176	0x151C2F00020000

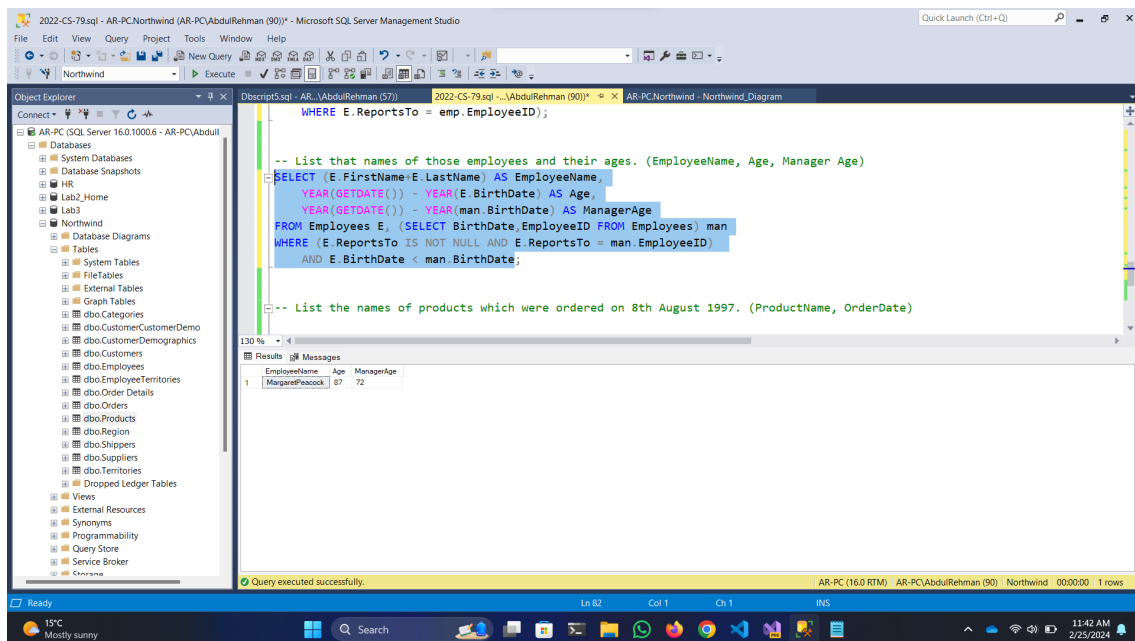
The status bar at the bottom indicates that the query was executed successfully, returning 1 row.

## 1.12 List that names of those employees and their ages. (EmployeeName, Age, Manager Age)

### SQL Query:

```
SELECT (E.FirstName+E.LastName) AS EmployeeName,  
       YEAR(GETDATE()) - YEAR(E.BirthDate) AS Age,  
       YEAR(GETDATE()) - YEAR(man.BirthDate) AS ManagerAge  
FROM Employees E, (SELECT BirthDate,EmployeeID FROM Employees) man  
WHERE (E.ReportsTo IS NOT NULL AND E.ReportsTo = man.EmployeeID)  
      AND E.BirthDate < man.BirthDate;
```

### Result:



The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The left pane shows the 'Object Explorer' with the 'Northwind' database selected. The right pane shows the 'Query Editor' with the following SQL query:

```
-- List that names of those employees and their ages. (EmployeeName, Age, Manager Age)  
SELECT (E.FirstName+E.LastName) AS EmployeeName,  
       YEAR(GETDATE()) - YEAR(E.BirthDate) AS Age,  
       YEAR(GETDATE()) - YEAR(man.BirthDate) AS ManagerAge  
FROM Employees E, (SELECT BirthDate,EmployeeID FROM Employees) man  
WHERE (E.ReportsTo IS NOT NULL AND E.ReportsTo = man.EmployeeID)  
      AND E.BirthDate < man.BirthDate;
```

The 'Results' pane at the bottom shows the output of the query, which is a single row:

EmployeeName	Age	ManagerAge
Margaret Peacock	67	72

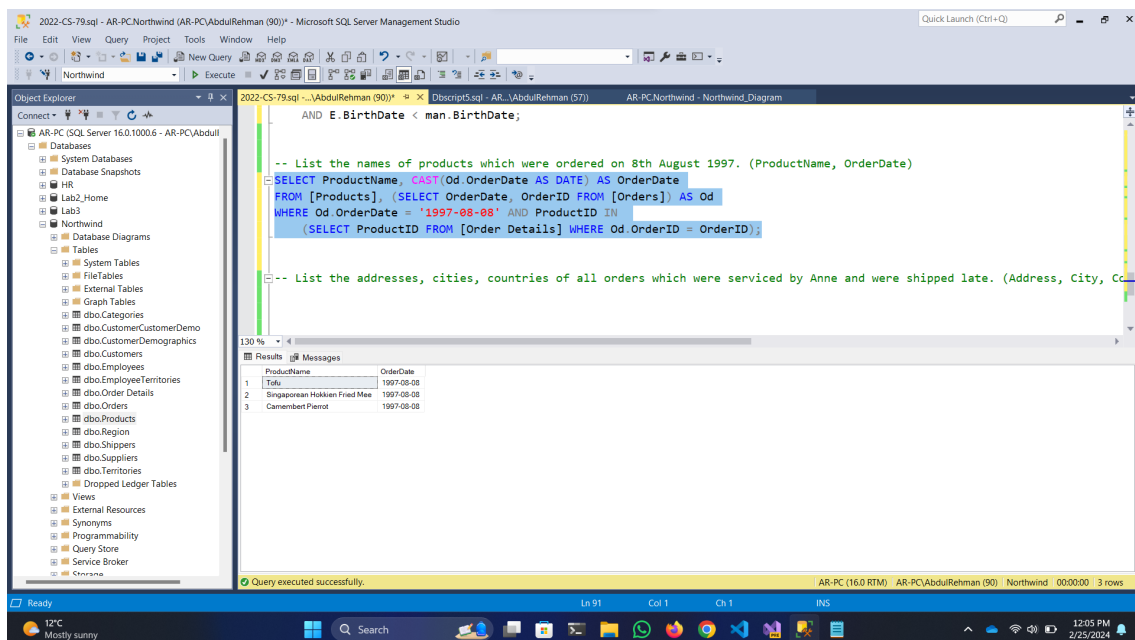
The status bar at the bottom indicates that the query was executed successfully, returning 1 row.

## 1.13 List the names of products which were ordered on 8th August 1997. (ProductName, OrderDate)

### SQL Query:

```
SELECT ProductName, CAST(Od.OrderDate AS DATE) AS OrderDate
FROM [Products], (SELECT OrderDate, OrderID FROM [Orders]) AS Od
WHERE Od.OrderDate = '1997-08-08' AND ProductID IN
      (SELECT ProductID FROM [Order Details] WHERE Od.OrderID = OrderID);
```

### Result:



The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the Object Explorer with the 'Northwind' database selected. The right pane shows a query window with the following SQL query:

```
-- List the names of products which were ordered on 8th August 1997. (ProductName, OrderDate)
SELECT ProductName, CAST(Od.OrderDate AS DATE) AS OrderDate
FROM [Products], (SELECT OrderDate, OrderID FROM [Orders]) AS Od
WHERE Od.OrderDate = '1997-08-08' AND ProductID IN
      (SELECT ProductID FROM [Order Details] WHERE Od.OrderID = OrderID);
```

The query results are displayed in the bottom pane, showing a table with two columns: ProductName and OrderDate. The results are as follows:

ProductName	OrderDate
Tofu	1997-08-08
Singaporean Hokkien Fried Mee	1997-08-08
Camembert Pierrot	1997-08-08

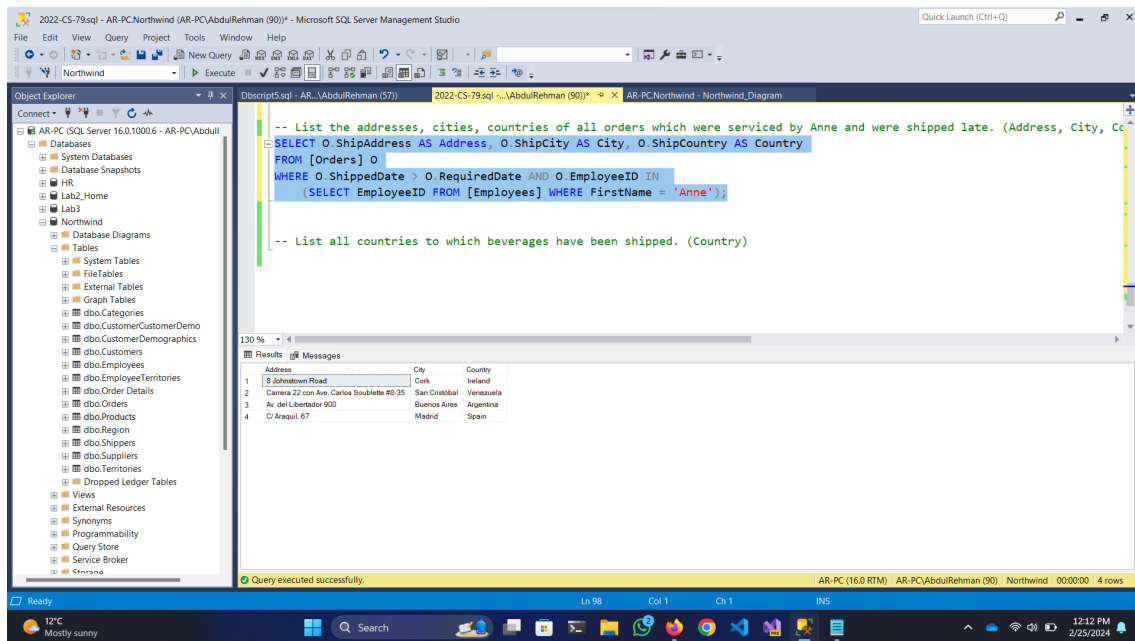
The status bar at the bottom indicates that the query was executed successfully.

## 1.14 List the addresses, cities, countries of all orders which were serviced by Anne and were shipped late. (Address, City, Country)

### SQL Query:

```
SELECT O.ShipAddress AS Address, O.ShipCity AS City, O.ShipCountry AS Country
FROM [Orders] O
WHERE O.ShippedDate > O.RequiredDate AND O.EmployeeID IN
(SELECT EmployeeID FROM [Employees] WHERE FirstName = 'Anne');
```

### Result:



The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the Object Explorer with the Northwind database selected. The right pane shows a query window with the following SQL query:

```
-- List the addresses, cities, countries of all orders which were serviced by Anne and were shipped late. (Address, City, Country)
SELECT O.ShipAddress AS Address, O.ShipCity AS City, O.ShipCountry AS Country
FROM [Orders] O
WHERE O.ShippedDate > O.RequiredDate AND O.EmployeeID IN
(SELECT EmployeeID FROM [Employees] WHERE FirstName = 'Anne');
```

Below the query window, the Results pane displays the following data:

Address	City	Country
8 Johnstown Road	Cork	Ireland
Carretera 22 con Ave. Carlos Soublette #8-35	San Cristobal	Venezuela
Av. del Libertador 900	Buenos Aires	Argentina
C/ Araquil, 67	Madrid	Spain

The status bar at the bottom indicates that the query was executed successfully, returning 4 rows.



## 1.15 List all countries to which beverages have been shipped. (Country)

### SQL Query:

```
SELECT O.ShipCountry AS Country
FROM [Orders] O
WHERE O.OrderID IN
    (SELECT OD.OrderID FROM [Order Details] OD WHERE OD.ProductID IN
        (SELECT P.ProductID FROM [Products] P WHERE P.CategoryID =
            (SELECT CategoryID FROM [Categories] WHERE LOWER(CategoryName) = 'beve
```

### Result:

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL query:

```
WHERE O.ShippedDate > O.RequiredDate AND O.EmployeeID IN
    (SELECT EmployeeID FROM [Employees] WHERE FirstName = 'Anne');

-- List all countries to which beverages have been shipped. (Country)
SELECT O.ShipCountry AS Country
FROM [Orders] O
WHERE O.OrderID IN
    (SELECT OD.OrderID FROM [Order Details] OD WHERE OD.ProductID IN
        (SELECT P.ProductID FROM [Products] P WHERE P.CategoryID =
            (SELECT CategoryID FROM [Categories] WHERE LOWER(CategoryName) = 'beverages')));
```

The Results pane shows the output of the query, which is a list of countries:

Country
59 Denmark
60 Germany
61 Brazil
62 Canada
63 France
64 Sweden
65 UK
66 France
67 Germany
68 USA
69 Sweden
70 Sweden
71 Germany
72 Finland
73 Belgium
74 Sweden

The status bar at the bottom indicates that the query was executed successfully, returning 301 rows.