



SUPERVISED BY:

Prof. Nazeef Ul Haq

SUBMITTED BY:

Waleed Ahmed 2022-CS-41

SUBJECT:

Lab 05 Hometask

SECTION:

A

1. Return customers and their orders, including customers who placed no orders. (CustomerID, OrderID, OrderDate)

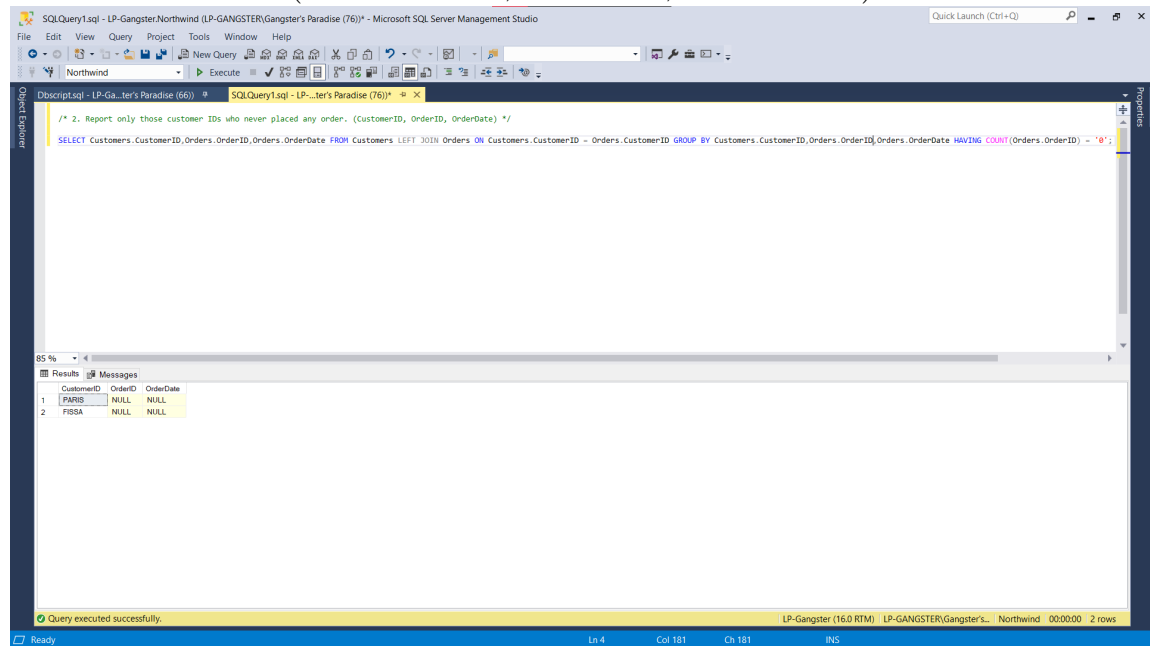
The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The top pane displays a query in the 'Query Editor' window. The query is a SQL statement that uses a LEFT JOIN to retrieve all customers and their orders, including customers who have not placed any orders. The bottom pane shows the 'Results' window, which displays the output of the query as a table with three columns: CustomerID, OrderID, and OrderDate. The results are sorted by CustomerID and OrderID.

```
/* 1. Return customers and their orders, including customers who placed no orders. (CustomerID, OrderID, OrderDate) */  
SELECT Customers.CustomerID, Orders.OrderID, Orders.OrderDate FROM Customers LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
```

CustomerID	OrderID	OrderDate
1	10341	1996-07-04 00:00:00.000
2	10249	1996-07-05 00:00:00.000
3	10250	1996-07-08 00:00:00.000
4	10251	1996-07-08 00:00:00.000
5	10252	1996-07-09 00:00:00.000
6	10253	1996-07-10 00:00:00.000
7	10254	1996-07-11 00:00:00.000
8	10255	1996-07-12 00:00:00.000
9	10256	1996-07-15 00:00:00.000
10	10257	1996-07-16 00:00:00.000
11	10259	1996-07-18 00:00:00.000
12	10260	1996-07-19 00:00:00.000
13	10261	1996-07-19 00:00:00.000
14	10262	1996-07-22 00:00:00.000
15	10263	1996-07-23 00:00:00.000
16	10264	1996-07-24 00:00:00.000
17	10265	1996-07-25 00:00:00.000
18	10266	1996-07-26 00:00:00.000

Query executed successfully.

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



The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The main window displays a query in the 'SQLQuery1.sql' file. The query is designed to find customer IDs who have never placed an order by using a LEFT JOIN and the HAVING clause to filter out rows where an order exists.

```
/* 2. Report only those customer IDs who never placed any order. (CustomerID, OrderID, OrderDate) */  
SELECT Customers.CustomerID, Orders.OrderID, Orders.OrderDate FROM Customers LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID GROUP BY Customers.CustomerID, Orders.OrderID, Orders.OrderDate HAVING COUNT(Orders.OrderID) = '0';
```

Below the query editor, the 'Results' pane shows the output of the query. It contains two rows of data, each with three columns: CustomerID, OrderID, and OrderDate.

	CustomerID	OrderID	OrderDate
1	PARIS	NULL	NULL
2	FBISSA	NULL	NULL

The status bar at the bottom indicates that the query was executed successfully and returned 2 rows.

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

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The top pane displays a query in the Query Editor:

```
/* 3. Report those customers who placed orders on July,1997. (CustomerID, OrderID, OrderDate) */  
SELECT CustomerID,OrderID,OrderDate FROM Orders GROUP BY CustomerID,OrderID,OrderDate HAVING MONTH(OrderDate) = '7' AND YEAR(OrderDate) = '1997';
```

The bottom pane shows the results of the query, displaying a table with three columns: CustomerID, OrderID, and OrderDate. The results are as follows:

CustomerID	OrderID	OrderDate
WELLI	10585	1997-07-01 00:00:00.000
REGUC	10586	1997-07-02 00:00:00.000
QUICK	10588	1997-07-03 00:00:00.000
GREAL	10589	1997-07-04 00:00:00.000
MEREP	10590	1997-07-07 00:00:00.000
LEHMS	10592	1997-07-08 00:00:00.000
LEHMS	10593	1997-07-09 00:00:00.000
OLDWO	10594	1997-07-09 00:00:00.000
ERNSH	10595	1997-07-10 00:00:00.000
WHITC	10596	1997-07-11 00:00:00.000
PHOOD	10597	1997-07-11 00:00:00.000
BOBEV	10599	1997-07-15 00:00:00.000
HUNGIC	10600	1997-07-16 00:00:00.000
HILAA	10601	1997-07-16 00:00:00.000
VAFEE	10602	1997-07-17 00:00:00.000
SAVEA	10603	1997-07-18 00:00:00.000
TRADH	10606	1997-07-22 00:00:00.000
SAVEA	10607	1997-07-22 00:00:00.000

The status bar at the bottom indicates that the query was executed successfully, showing the server name, database, and the number of rows returned (26 rows).

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

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

```
/* 4. Report the total orders of each customer. (customerID, totalorders) */  
SELECT Customers.CustomerID, COUNT(Orders.OrderID) AS [Total Orders] FROM Customers LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID GROUP BY Customers.CustomerID;
```

The Results pane displays the following data:

CustomerID	Total Orders
ALFKI	4
ANATR	4
ANTON	6
AROUT	10
BERGS	14
BLAUS	7
BLOMP	11
BOLID	3
BONAP	14
BOTTM	12
BSBEV	9
CACTU	6
CELTIC	1
CHOPS	7
COMMI	4
CONSH	3
DELCO	4
DUMON	3

The status bar at the bottom indicates: "Query executed successfully. LP-Gangster (16.0 RTM) LP-GANGSTER;Gangster's... Northwind 00:00:00 91 rows".

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

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The query editor displays the following SQL query:

```
/* 5. Write a query to generate a five copies of each employee. (EmployeeID, FirstName, LastName) */  
SELECT E1.EmployeeID,E1.FirstName,E1.LastName FROM Employees AS E1,Employees AS E2 WHERE E2.EmployeeID % 2 = 0 OR E2.EmployeeID = 3 ORDER BY EmployeeID;
```

The Results pane shows the output of the query, which is a list of employee records. The status bar at the bottom indicates that the query was executed successfully and returned 40 rows.

EmployeeID	FirstName	LastName
2	Andrew	Fuller
2	Andrew	Fuller
2	Andrew	Fuller
2	Andrew	Fuller
2	Andrew	Fuller
3	Janet	Leverling
3	Janet	Leverling
3	Janet	Leverling
3	Janet	Leverling
3	Janet	Leverling
4	Margaret	Peacock
4	Margaret	Peacock
4	Margaret	Peacock
4	Margaret	Peacock
4	Margaret	Peacock
5	Steven	Buchanan
5	Steven	Buchanan
5	Steven	Buchanan
5	Steven	Buchanan
5	Steven	Buchanan

6. 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 screenshot shows the Microsoft SQL Server Enterprise Manager interface. The top pane displays a query in the 'Query Editor' window. The query is as follows:

```
/* 6. 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 EmployeeID, OrderDate AS Date FROM ORDERS WHERE OrderDate >= '1996-07-04 00:00:00.000' AND OrderDate <= '1997-08-04 00:00:00.000';
```

The bottom pane shows the 'Results' window with the following data:

EmployeeID	Date
5	1996-07-04 00:00:00.000
6	1996-07-05 00:00:00.000
4	1996-07-06 00:00:00.000
3	1996-07-08 00:00:00.000
4	1996-07-09 00:00:00.000
5	1996-07-09 00:00:00.000
6	1996-07-10 00:00:00.000
3	1996-07-11 00:00:00.000
5	1996-07-11 00:00:00.000
9	1996-07-12 00:00:00.000
3	1996-07-15 00:00:00.000
4	1996-07-16 00:00:00.000
4	1996-07-18 00:00:00.000
4	1996-07-19 00:00:00.000
4	1996-07-19 00:00:00.000
8	1996-07-22 00:00:00.000
9	1996-07-23 00:00:00.000
6	1996-07-24 00:00:00.000
2	1996-07-25 00:00:00.000
3	1996-07-26 00:00:00.000

The status bar at the bottom indicates 'Query executed successfully.' and '318 rows'.

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

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

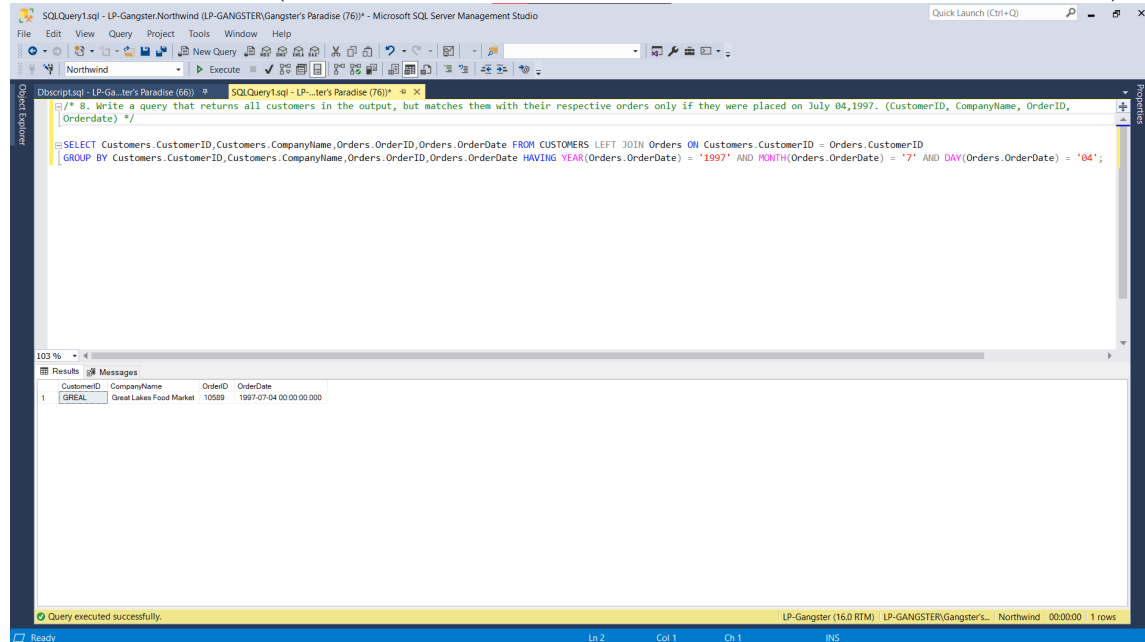
```
/* 7. Return US customers, and for each customer return the total number of orders and total quantities. (CustomerID, Totalorders, totalquantity) */
SELECT Customers.CustomerID, COUNT(DISTINCT Orders.OrderID) AS Totalorders, SUM(Quantity) AS totalquantity FROM Customers LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID CROSS JOIN [Order Details]
WHERE Orders.OrderID = [Order Details].OrderID GROUP BY Customers.CustomerID, Orders.ShipCountry HAVING Orders.ShipCountry = 'USA';
```

The Results pane displays the following data:

CustomerID	Totalorders	totalquantity
10	286	
4	102	
1	10	
3	150	
7	103	
9	493	
13	909	
25	3851	
7	263	
3	44	
3	59	
3	89	
13	986	

The status bar at the bottom indicates: "Query executed successfully. LP-Gangster (16.0 RTM) LP-GANGSTER\Gangster's... Northwind 00:00:00 13 rows".

8. 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)



9. Are there any employees who are older than their managers?

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The main window shows a query executed in the 'Northwind' database. The query is designed to find employees who are older than their managers. The results pane shows a single row of data.

Query Text:

```
/* 9. Are there any employees who are older than their managers? */
SELECT E1.FirstName + ' ' + E1.LastName + ' is older than his manager ' + E2.FirstName + ' ' + E2.LastName AS [Older Employees]
FROM Employees AS E1, Employees AS E2
GROUP BY E2.FirstName, E2.LastName, E1.FirstName, E1.LastName, E1.BirthDate, E2.BirthDate, E2.EmployeeID, E1.ReportsTo
HAVING (2024 - YEAR(E1.BirthDate) > 2024 - YEAR(E2.BirthDate)) AND E2.EmployeeID = E1.ReportsTo;
```

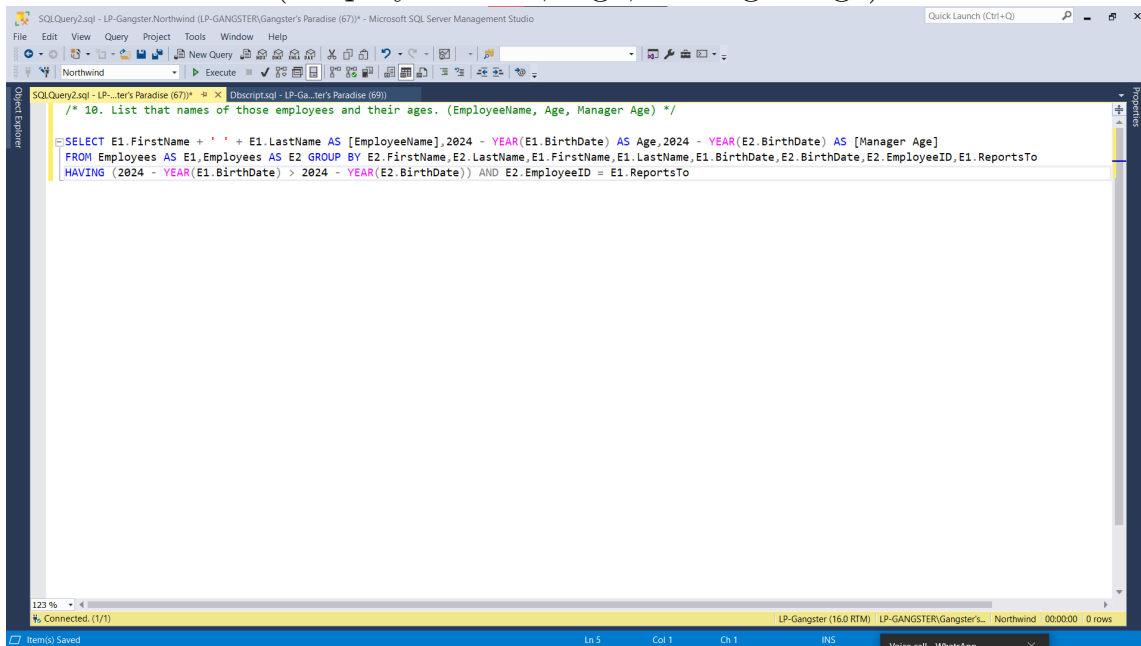
Results:

Older Employees
1 Margaret Peacock is older than his manager Andrew Fuller

Query executed successfully.

LP-Gangster (16.0 RTM) LP-GANGSTER/Gangster's... Northwind 00:00:00 1 rows

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



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

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The main window displays a query in the 'Query Editor' pane. The query is as follows:

```
/* 11. List the names of products which were ordered on 8th August 1997. (ProductName, OrderDate) */  
  
SELECT Products.ProductName, Orders.OrderDate FROM Products, Orders, [Order Details] WHERE Orders.OrderID = [Order Details].OrderID AND Products.ProductID = [Order Details].ProductID  
GROUP BY Products.ProductName, Orders.OrderDate HAVING YEAR(Orders.OrderDate) = '1997' AND MONTH(Orders.OrderDate) = '8' AND DAY(Orders.OrderDate) = '8';
```

The 'Results' pane at the bottom shows the output of the query, which is a table with two columns: ProductName and OrderDate. The table contains three rows of data:

ProductName	OrderDate
Camembert Pierrot	1997-08-08 00:00:00.000
Singaporean Hokkien Fried Mee	1997-08-08 00:00:00.000
Tofu	1997-08-08 00:00:00.000

The status bar at the bottom indicates that the query was executed successfully and shows the server name 'LP-Gangster (16.0 RTM)' and the database 'Northwind'.

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

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The query window displays the following SQL query:

```
/* 12. List the addresses, cities, countries of all orders which were serviced by Anne and were shipped late. (Address, city, country) */
SELECT Orders.ShipAddress,Orders.ShipCity,Orders.ShipCountry FROM Employees,Orders WHERE Employees.EmployeeID = Orders.EmployeeID AND Employees.FirstName = 'Anne';
```

The Results pane shows the following data:

ShipAddress	ShipCity	ShipCountry
187 Suffolk Ln.	Boise	USA
12 rue des Bouchers	Marseille	France
Rua Orla 92	Sao Paulo	Brazil
23 Tsawassen Blvd.	Tsawassen	Canada
Boulevard Tirou, 255	Charleroi	Belgium
Fauntleroy Circus	London	UK
Magazinweg 7	Frankfurt a.M.	Germany
24 place Kléber	Strasbourg	France
722 Delaware Blvd.	Kirkland	USA
Strada Provinciale 124	Reggio Emilia	Italy
8 Johnstown Road	Cork	Ireland
Berguvägen 8	Luleå	Sweden
8 Johnstown Road	Cork	Ireland
Carerra 22 con Ave. Carlos Soublette #8-35	San Cristóbal	Venezuela

The status bar at the bottom indicates: LP-Gangster (16.0 RTM) LP-GANGSTER\Gangster's... Northwind 00:00:00 43 rows. A message at the bottom states: Query executed successfully.

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

