Foundation Technical Training

Assignment No. 1 - SQL - Electronic Gadgets - Task 3

Name: Niranjan Kolpe, Batch: C#-Batch 2

Tasks 3: Aggregate functions, Having, Order By, GroupBy and Joins:

1. Write an SQL query to retrieve a list of all orders along with customer information (e.g., customer name) for each order.
2. Write an SQL query to find the total revenue generated by each electronic gadget product. Include the product name and the total revenue.
3. Write an SQL query to list all customers who have made at least one purchase. Include their names and contact information.
4. Write an SQL query to find the most popular electronic gadget, which is the one with the highest total quantity ordered. Include the product name and the total quantity ordered.
5. Write an SQL query to retrieve a list of electronic gadgets along with their corresponding categories.
6. Write an SQL query to calculate the average order value for each customer. Include the customer's name and their average order value.
7. Write an SQL query to find the order with the highest total revenue. Include the order ID, customer information, and the total revenue.
8. Write an SQL query to list electronic gadgets and the number of times each product has been ordered.
9. Write an SQL query to find customers who have purchased a specific electronic gadget product. Allow users to input the product name as a parameter.
10. Write an SQL query to calculate the total revenue generated by all orders placed within a specific time period. Allow users to input the start and end dates as parameters.

Program Code in SQL:

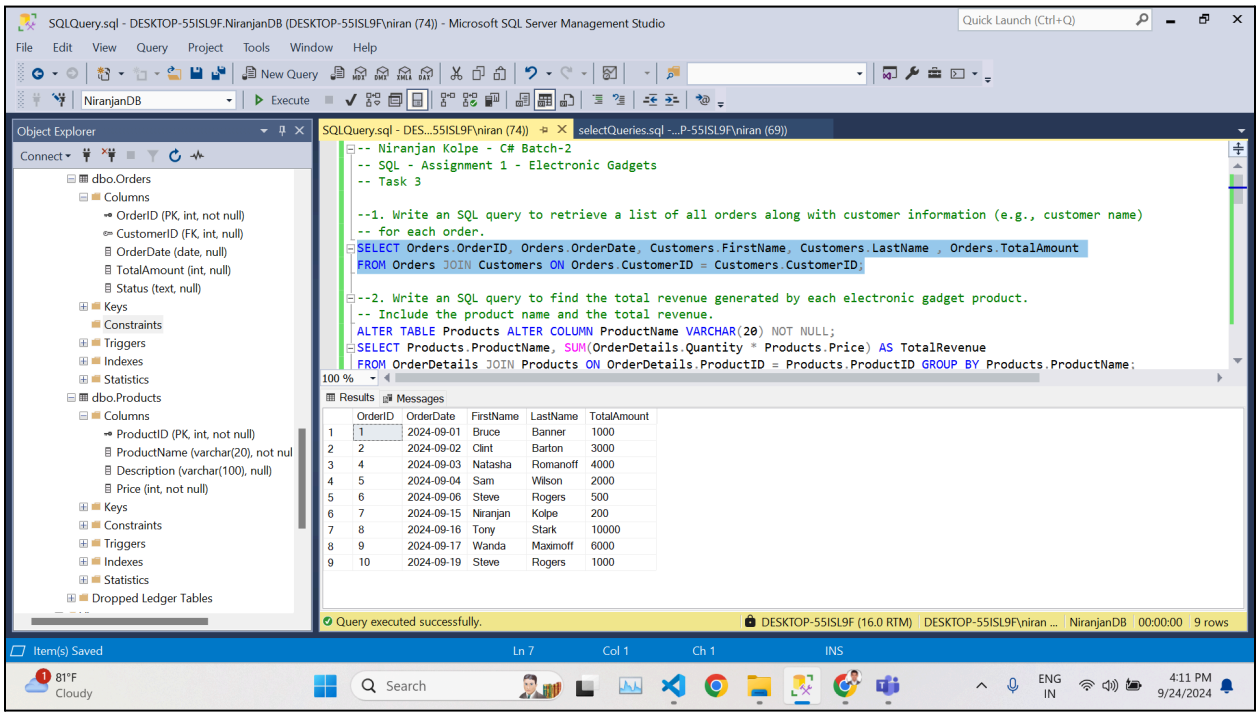
```
-- Niranjan Kolpe - C# Batch-2

-- SQL - Assignment 1 - Electronic Gadgets

-- Task 3


--1. Write an SQL query to retrieve a list of all orders along with customer
information (e.g., customer name)

-- for each order.

SELECT Orders.OrderID, Orders.OrderDate, Customers.FirstName, Customers.LastName
, Orders.TotalAmount
```

```sql
FROM Orders JOIN Customers ON Orders.CustomerID = Customers.CustomerID;


--2. Write an SQL query to find the total revenue generated by each electronic gadget product.

-- Include the product name and the total revenue.

ALTER TABLE Products ALTER COLUMN ProductName VARCHAR(20) NOT NULL;

SELECT Products.ProductName, SUM(OrderDetails.Quantity * Products.Price) AS TotalRevenue

FROM OrderDetails JOIN Products ON OrderDetails.ProductID = Products.ProductID GROUP BY Products.ProductName;


--3. Write an SQL query to list all customers who have made at least one purchase.

-- Include their names and contact information.

ALTER TABLE Customers ALTER COLUMN FirstName VARCHAR(20) NOT NULL;

ALTER TABLE Customers ALTER COLUMN LastName VARCHAR(20) NOT NULL;

SELECT DISTINCT Customers.FirstName, Customers.LastName, Customers.Email, Customers.Phone, Customers.Address

FROM Customers JOIN Orders ON Customers.CustomerID = Orders.CustomerID;


--4. Write an SQL query to find the most popular electronic gadget,

-- which is the one with the highest total quantity ordered.

-- Include the product name and the total quantity ordered.

SELECT TOP 1 Products.ProductName, SUM(OrderDetails.Quantity) AS TotalQuantityOrdered

FROM OrderDetails JOIN Products ON OrderDetails.ProductID = Products.ProductID

GROUP BY Products.ProductName ORDER BY TotalQuantityOrdered DESC;
```

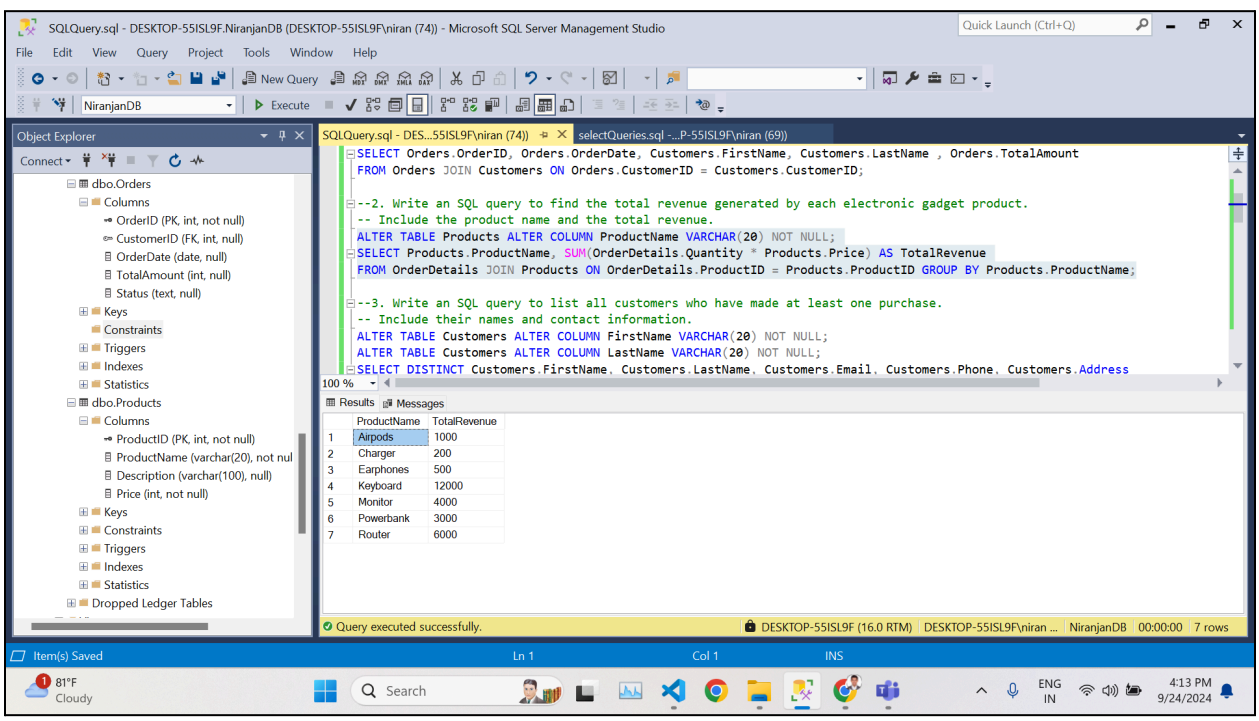
```sql
--5. Write an SQL query to retrieve a list of electronic gadgets along with their corresponding categories.

SELECT ProductName, Description FROM Products;


--6. Write an SQL query to calculate the average order value for each customer.

-- Include the customer's name and their average order value.

SELECT Customers.FirstName, Customers.LastName, AVG(Orders.TotalAmount) AS AverageOrderValue

FROM Orders JOIN Customers ON Orders.CustomerID = Customers.CustomerID

GROUP BY Customers.FirstName, Customers.LastName;


--7. Write an SQL query to find the order with the highest total revenue.

-- Include the order ID, customer information, and the total revenue.

SELECT TOP 1 Orders.OrderID, Customers.CustomerID, Customers.FirstName, Customers.LastName, Orders.TotalAmount AS TotalRevenue

FROM Orders JOIN Customers ON Orders.CustomerID = Customers.CustomerID ORDER BY TotalRevenue DESC;


--8. Write an SQL query to list electronic gadgets and the number of times each product has been ordered.

SELECT Products.ProductName, COUNT(OrderDetails.ProductID) AS NumberOfOrders

FROM OrderDetails JOIN Products ON OrderDetails.ProductID = Products.ProductID GROUP BY Products.ProductName;


--9. Write an SQL query to find customers who have purchased a specific electronic gadget product.

-- Allow users to input the product name as a parameter.

--ALTER TABLE Orders DROP COLUMN Status;

--ALTER TABLE Orders ADD OrdersStatus VARCHAR(20);
```

```sql
--UPDATE Orders SET OrdersStatus='Pending';

--UPDATE Orders SET OrdersStatus='Shipped' WHERE OrderID=8 AND CustomerID=3;


DECLARE @ProductName VARCHAR(20) = 'Keyboard';

SELECT * FROM Customers

JOIN Orders ON Customers.CustomerID = Orders.CustomerID

JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

JOIN Products ON OrderDetails.ProductID = Products.ProductID

WHERE Products.ProductName = @ProductName;


--10. Write an SQL query to calculate the total revenue generated by all orders placed within a specific time period.

-- Allow users to input the start and end dates as parameters.

DECLARE @StartDate DATE='2024-09-03', @EndDate DATE = '2024-09-16';

SELECT SUM(Orders.TotalAmount) AS TotalRevenue FROM Orders WHERE Orders.OrderDate BETWEEN @StartDate AND @EndDate;
```

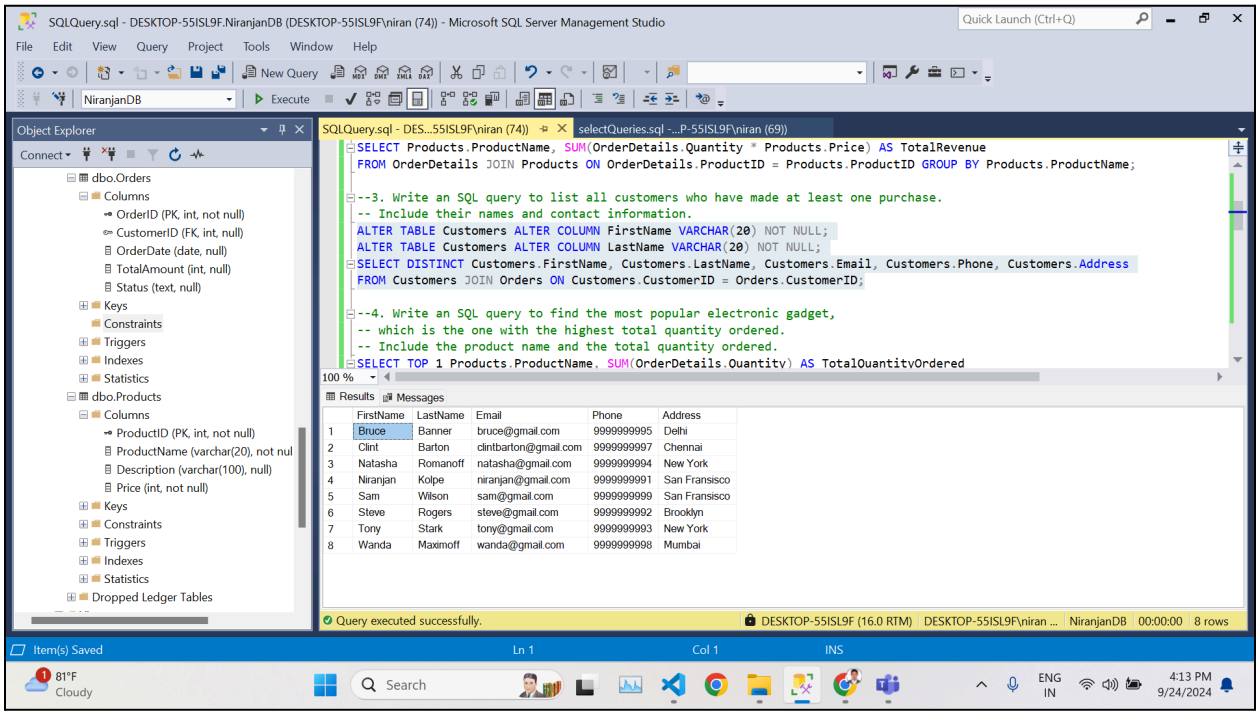
Output 1:



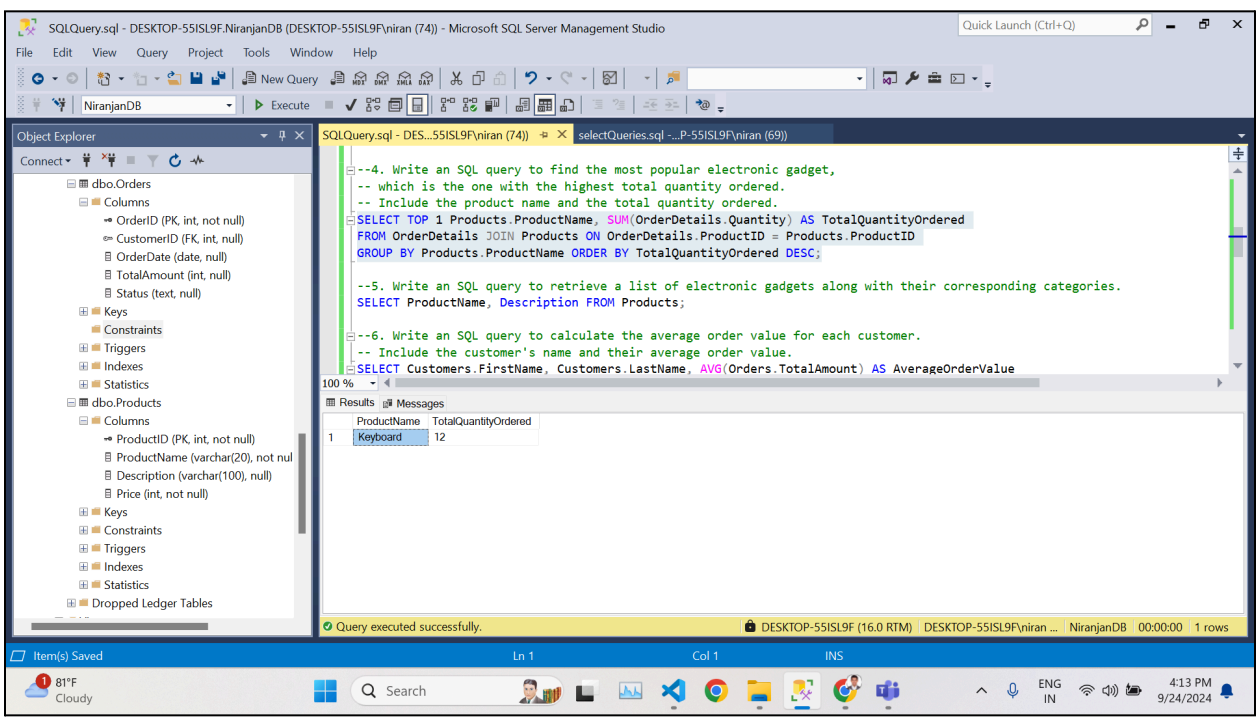
Output 2:



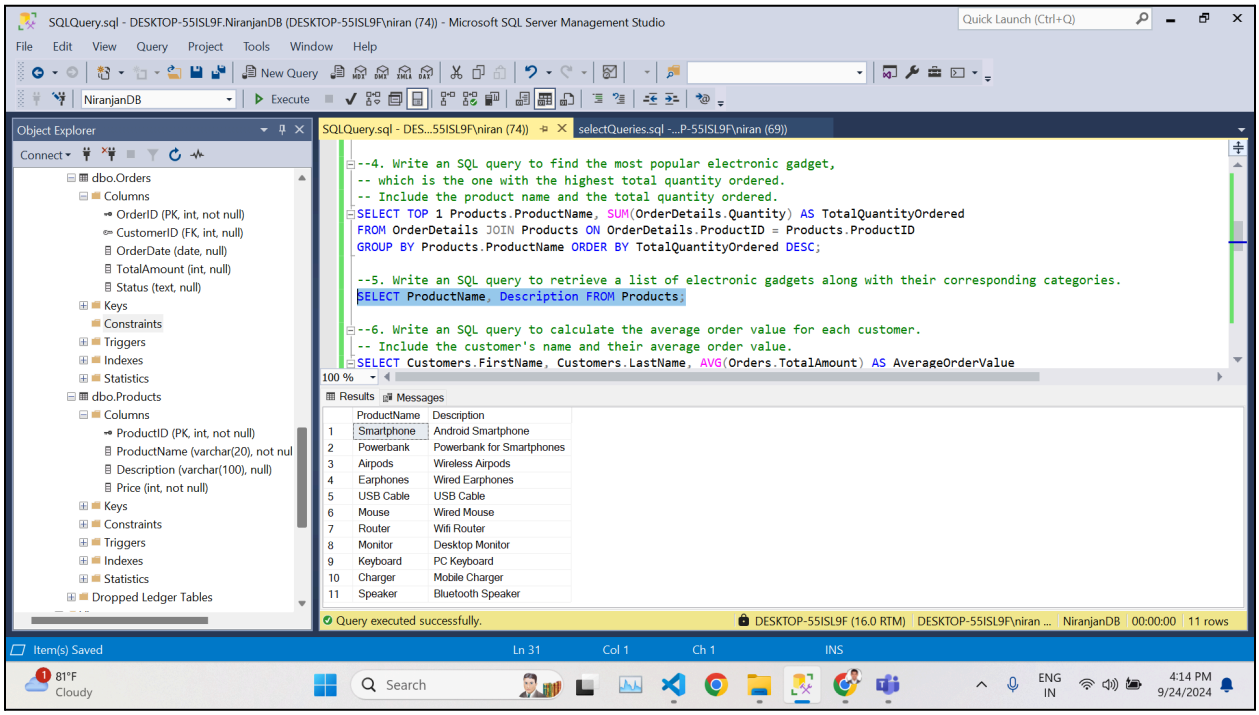
Output 3:



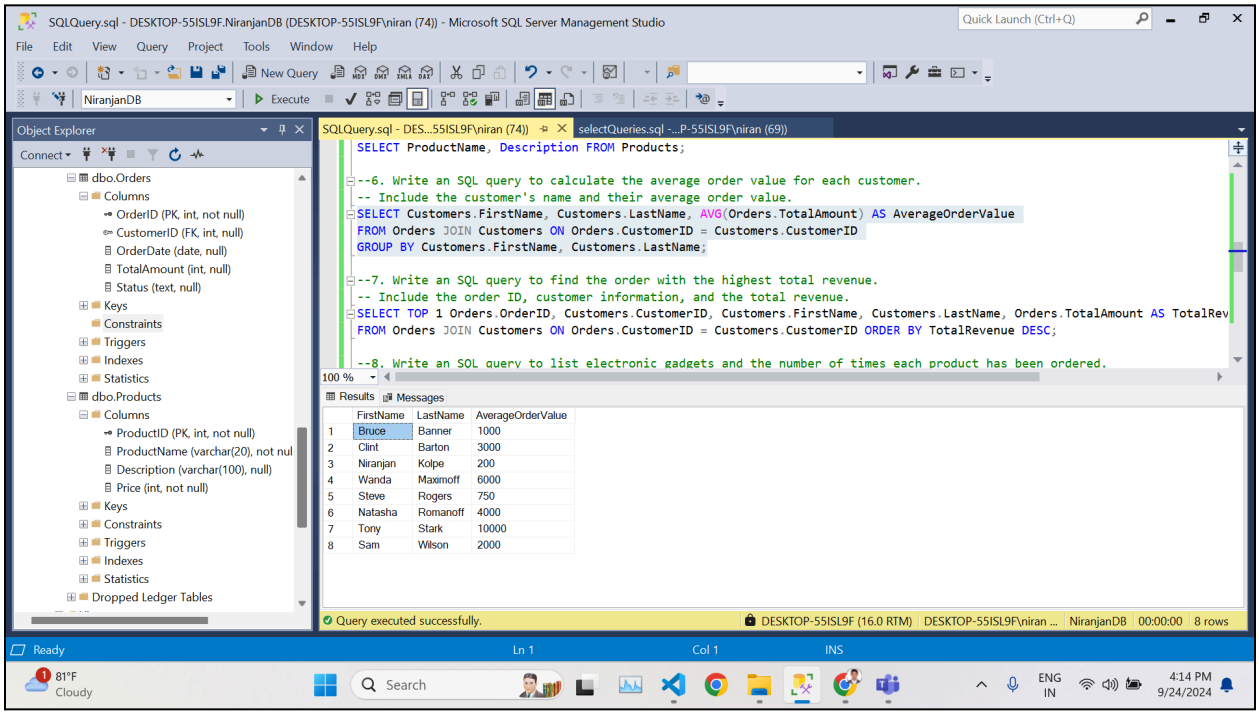
Output 4:



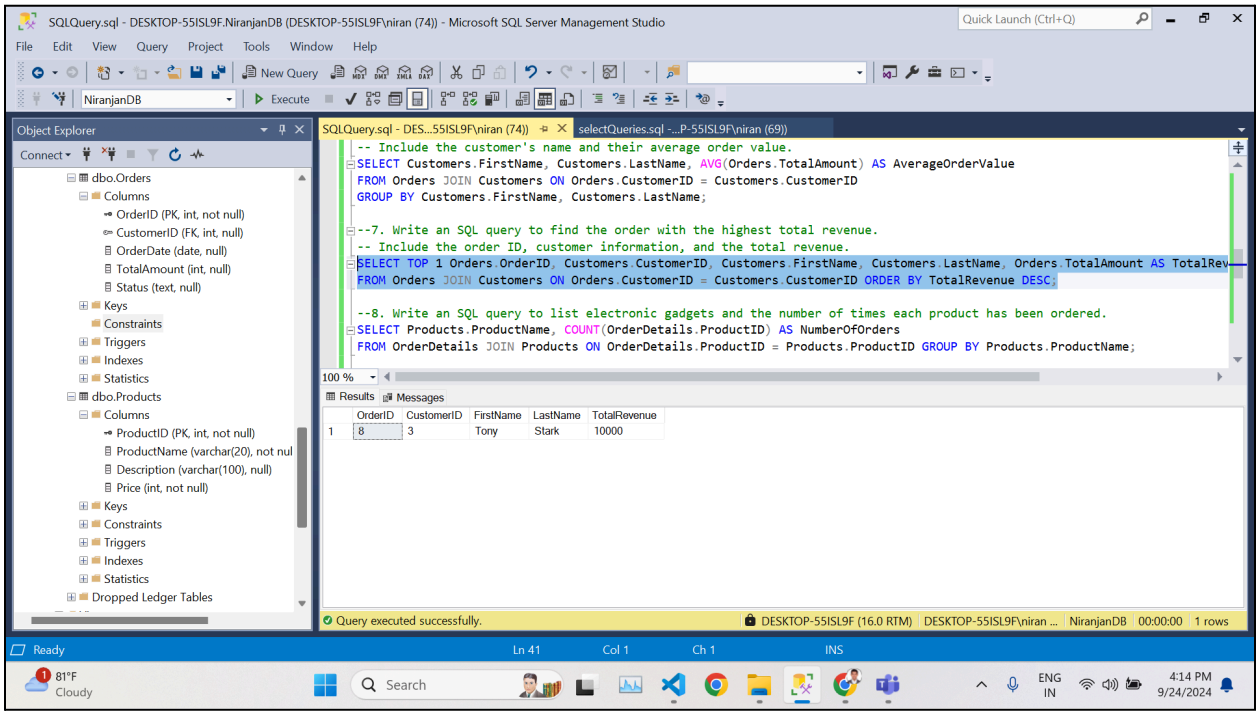
Output 5:



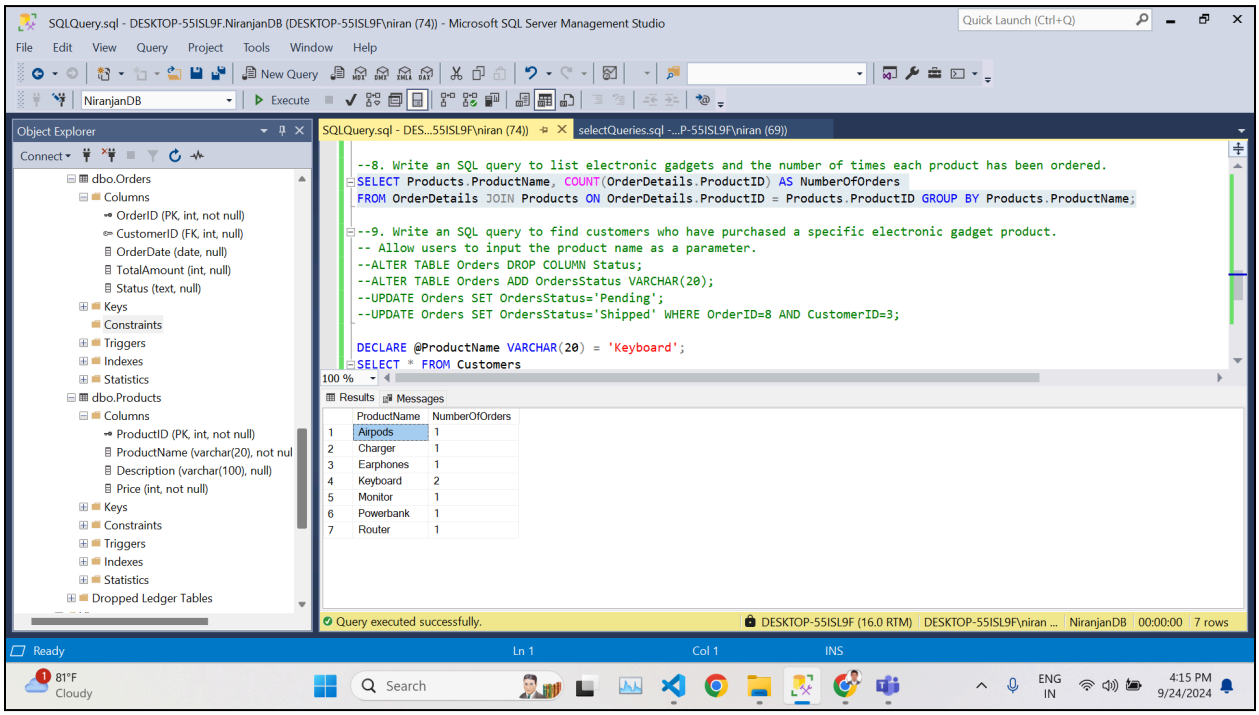
Output 6:



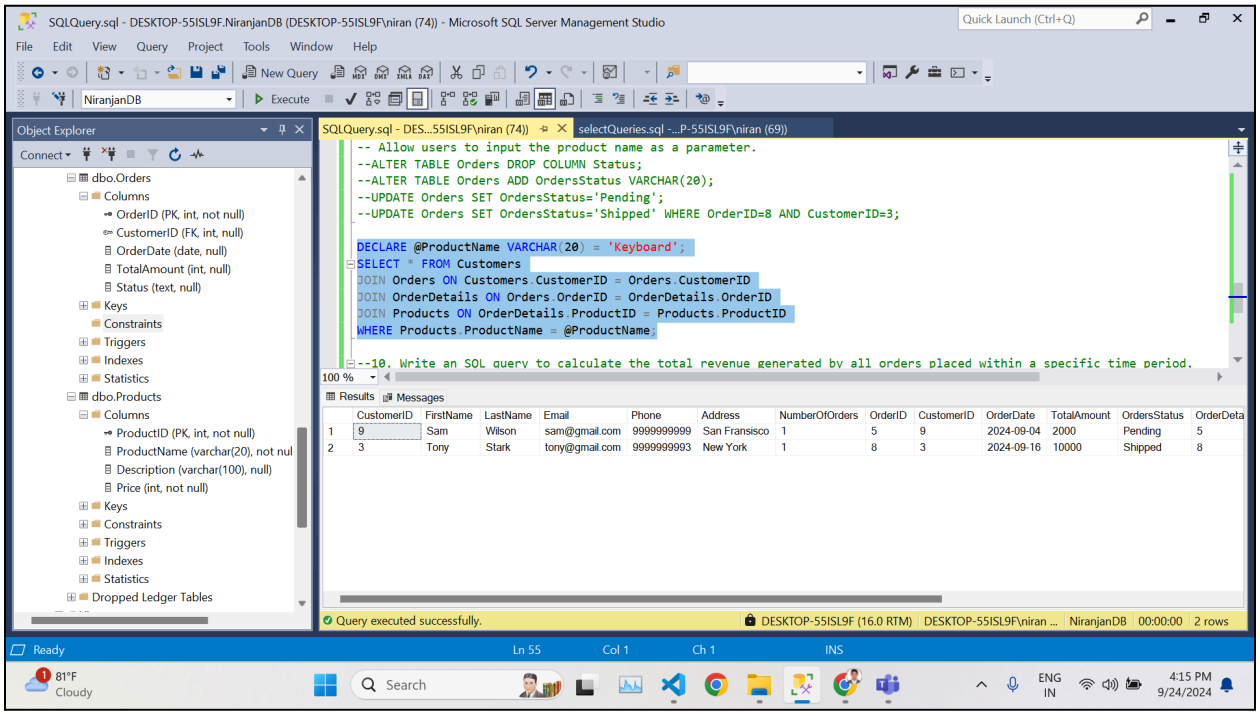
Output 7:



Output 8:



Output 9:



Output 10: