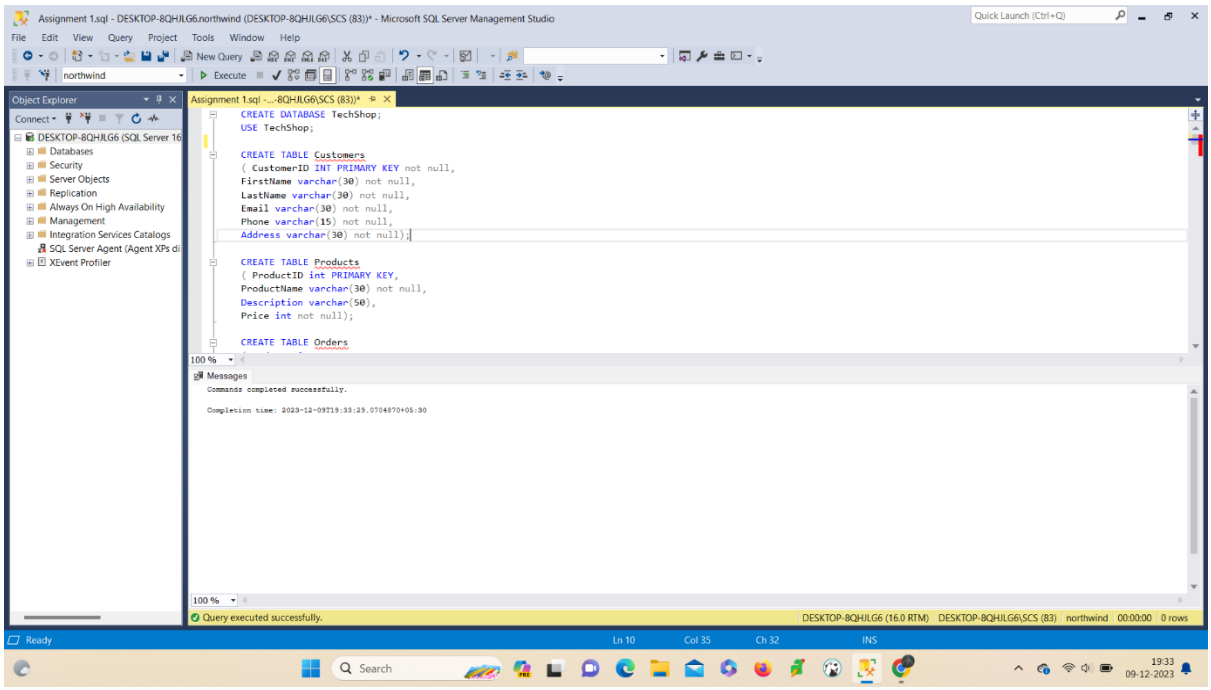
Assignment 1

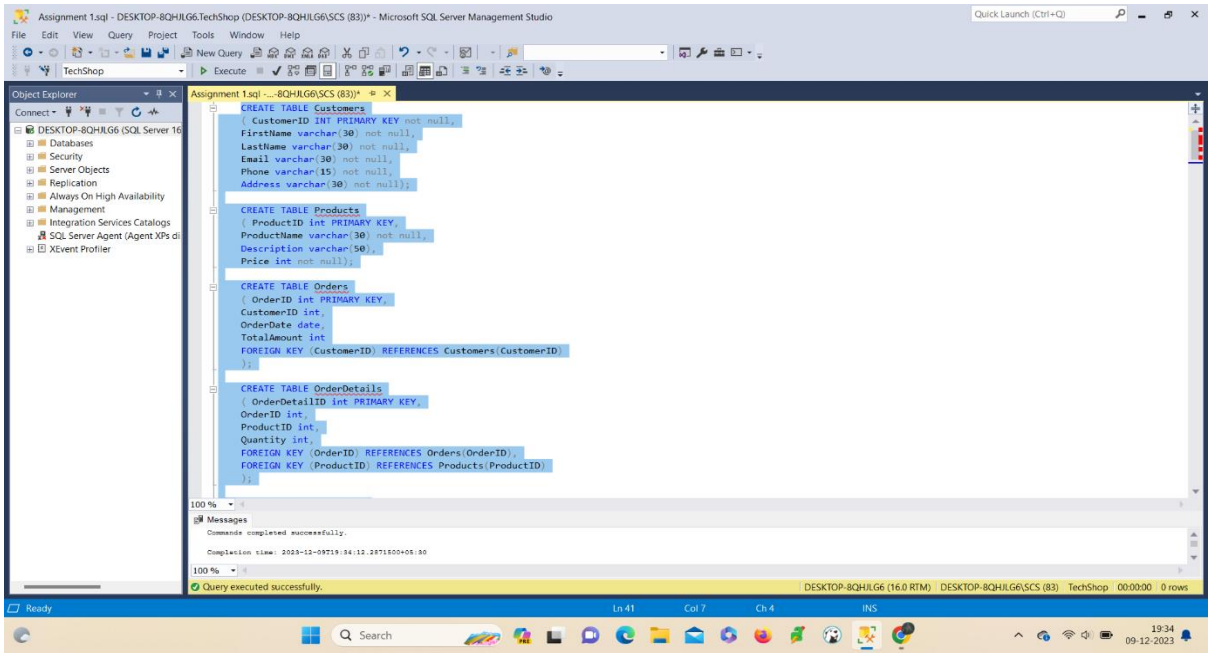
TechShop

Task 1: Database Design

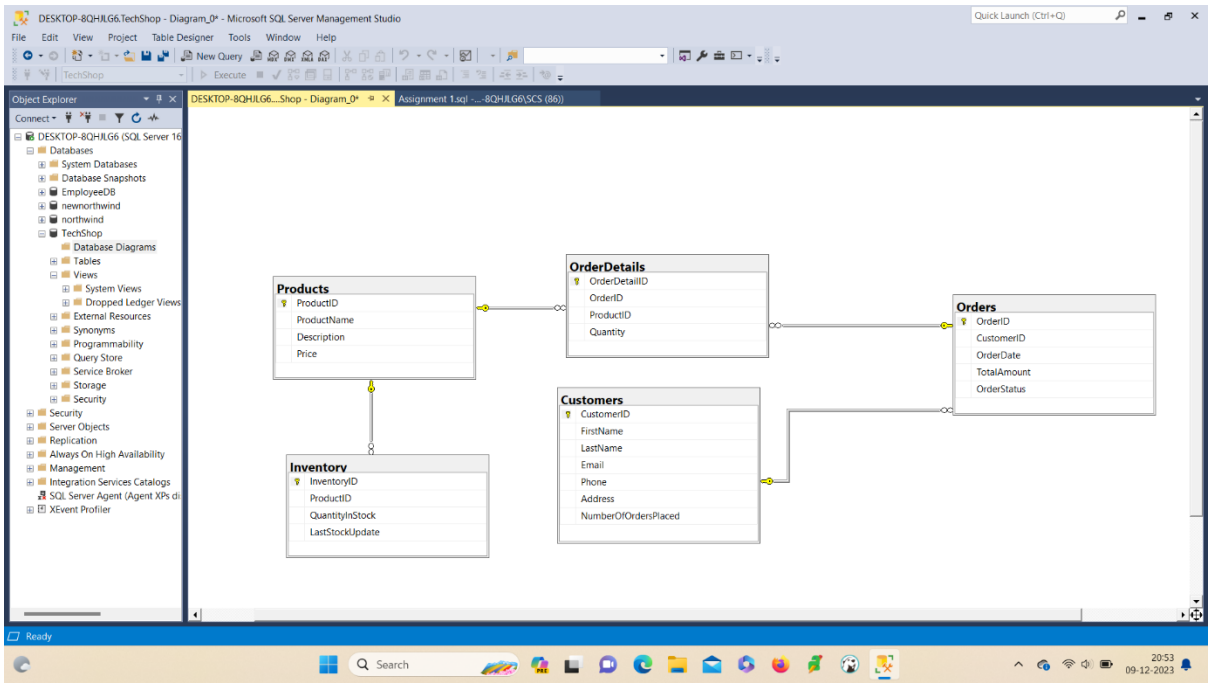
1.



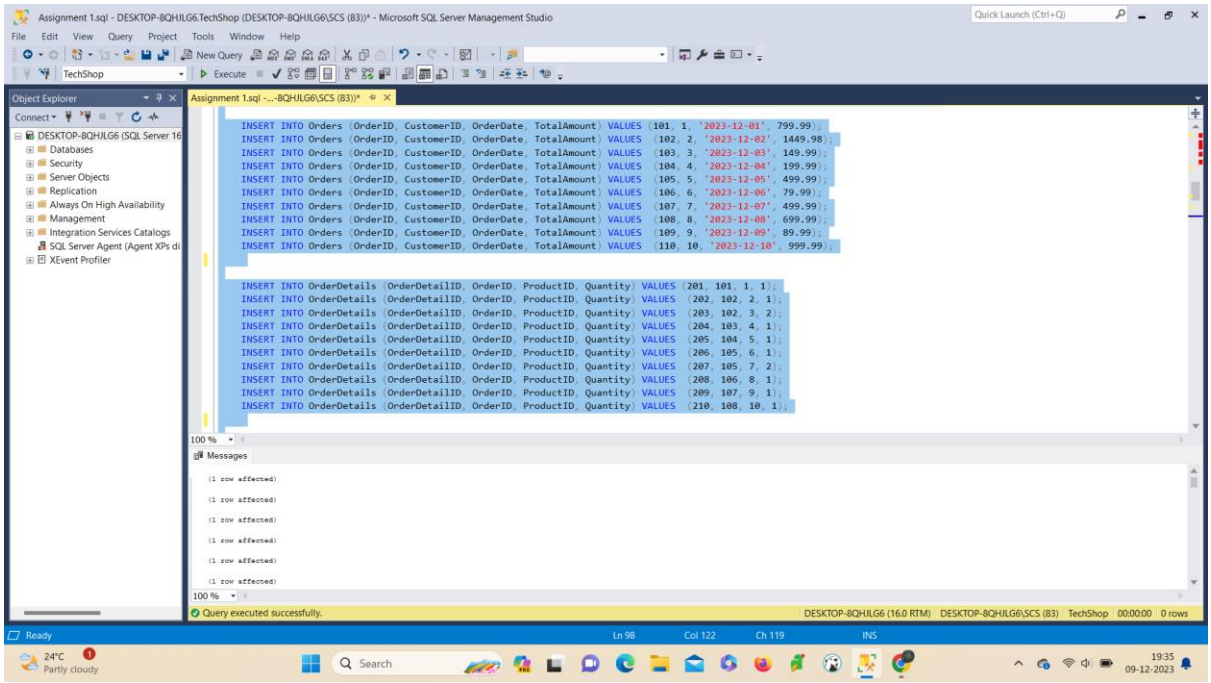
2.

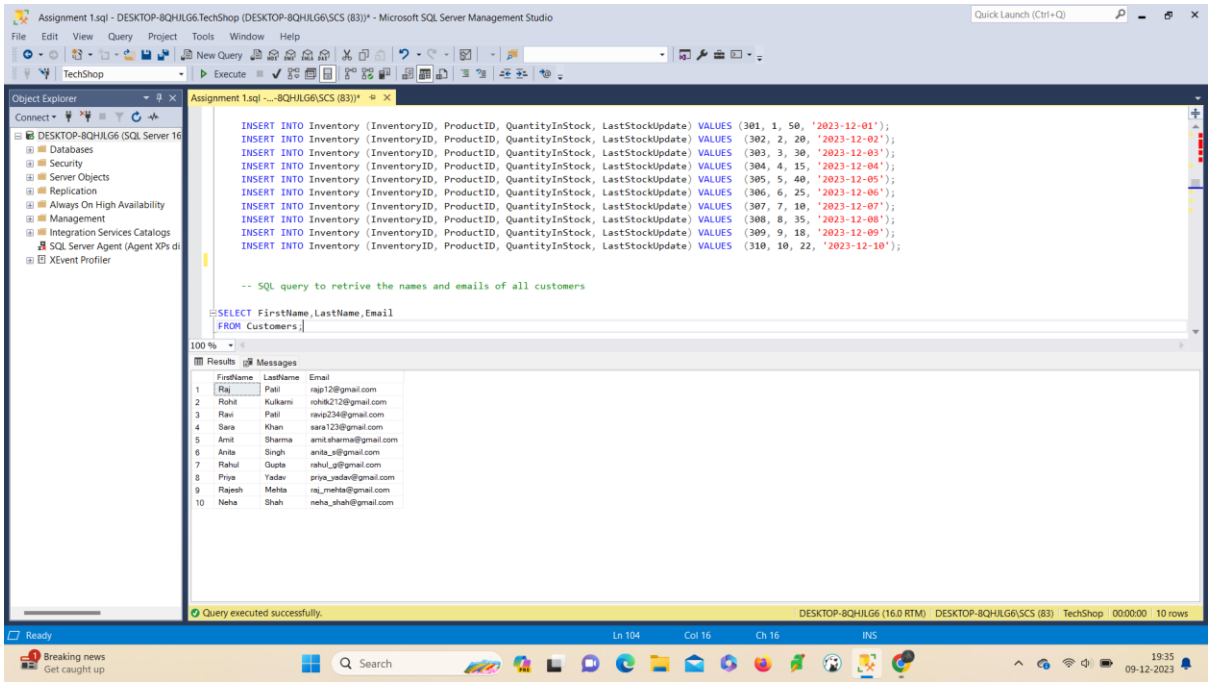


3.


DESKTOP-8QHJLG6.TechShop - Diagram_0* - Microsoft SQL Server Management Studio
Products
ProductID
ProductName
Description
Price
OrderDetails
OrderDetailID
OrderID
ProductID
Quantity
Orders
OrderID
CustomerID
OrderDate
TotalAmount
OrderStatus
Customers
CustomerID
FirstName
LastName
Email
Phone
Address
NumberOfOrdersPlaced
Inventory
InventoryID
ProductID
QuantityInStock
LastStockUpdate

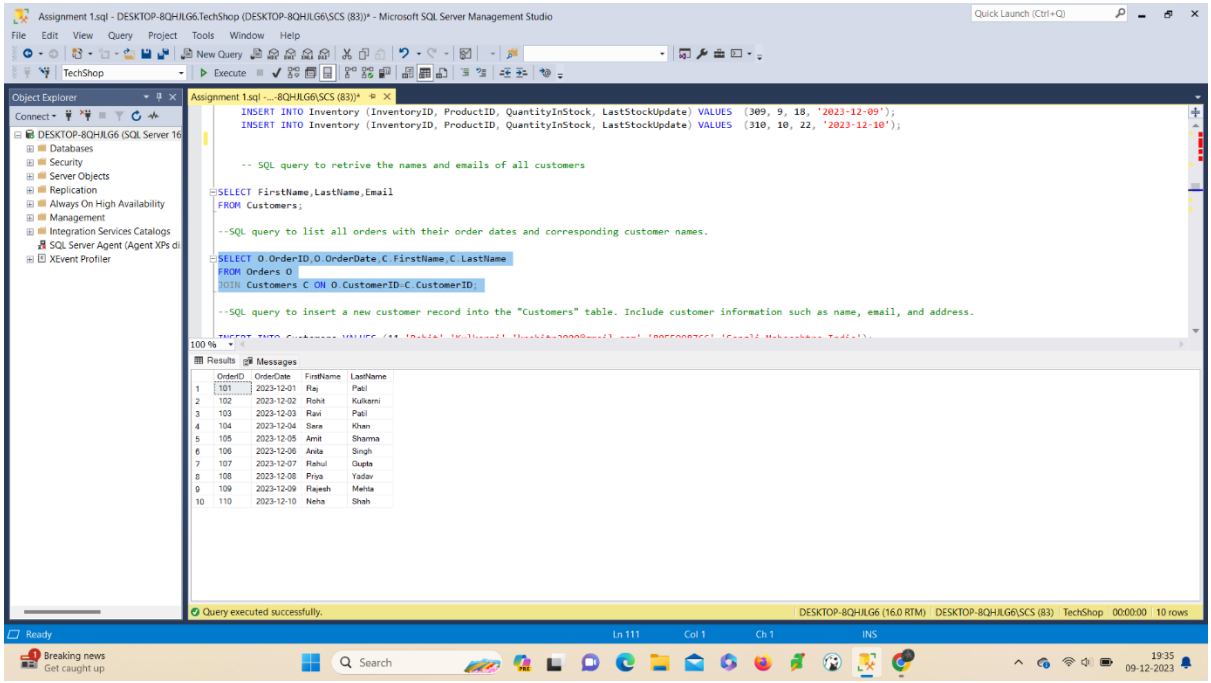
4.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES (101, 1, '2023-12-01', 799.99);
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES (102, 2, '2023-12-02', 1449.98);
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES (103, 3, '2023-12-03', 149.99);
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES (104, 4, '2023-12-04', 199.99);
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES (105, 5, '2023-12-05', 499.99);
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES (106, 6, '2023-12-06', 79.99);
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES (107, 7, '2023-12-07', 499.99);
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES (108, 8, '2023-12-08', 699.99);
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES (109, 9, '2023-12-09', 89.99);
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES (110, 10, '2023-12-10', 999.99);
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity) VALUES (201, 101, 1, 1);
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity) VALUES (202, 102, 2, 1);
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity) VALUES (203, 102, 3, 2);
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity) VALUES (204, 103, 4, 1);
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity) VALUES (205, 104, 5, 1);
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity) VALUES (206, 105, 6, 1);
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity) VALUES (207, 105, 7, 2);
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity) VALUES (208, 106, 8, 1);
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity) VALUES (209, 107, 9, 1);
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity) VALUES (210, 108, 10, 1);
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
Query executed successfully.

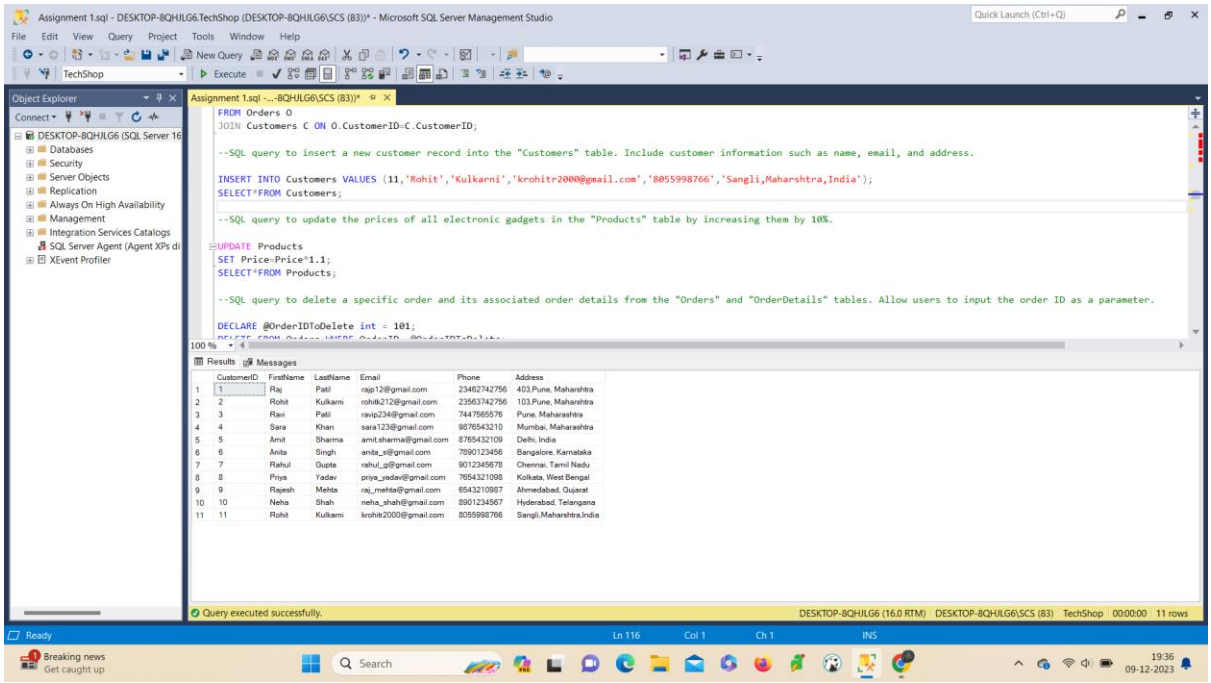

Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
TechShop
Object Explorer
Connect
DESKTOP-8QHJLG6 (SQL Server 16
Databases
Security
Server Objects
Replication
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs di
XEvent Profiler
Assignment 1.sql -...-8QHJLG6\SCS (83))*
INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate) VALUES (301, 1, 50, '2023-12-01');
INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate) VALUES (302, 2, 20, '2023-12-02');
INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate) VALUES (303, 3, 30, '2023-12-03');
INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate) VALUES (304, 4, 15, '2023-12-04');
INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate) VALUES (305, 5, 40, '2023-12-05');
INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate) VALUES (306, 6, 25, '2023-12-06');
INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate) VALUES (307, 7, 10, '2023-12-07');
INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate) VALUES (308, 8, 35, '2023-12-08');
INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate) VALUES (309, 9, 18, '2023-12-09');
INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate) VALUES (310, 10, 22, '2023-12-10');
-- SQL query to retrive the names and emails of all customers
SELECT FirstName,LastName,Email
FROM Customers;
100 %
Results
Messages
FirstName LastName Email
1 Raj Patil rajp12@gmail.com
2 Rohit Kulkarni rohit6212@gmail.com
3 Ravi Patil ravip234@gmail.com
4 Sara Khan sara123@gmail.com
5 Amit Sharma amit.sharma@gmail.com
6 Anita Singh anita_s@gmail.com
7 Rahul Gupta rahul_g@gmail.com
8 Priya Yadav priya_yadav@gmail.com
9 Rajesh Mehta raj_mehta@gmail.com
10 Neha Shah neha_shah@gmail.com
Query executed successfully.
DESKTOP-8QHJLG6 (16.0 RTM) DESKTOP-8QHJLG6\SCS (83) TechShop 00:00:00 10 rows
Ready
Ln 104 Col 16 Ch 16 INS
Breaking news
Get caught up
Search
19:35
09-12-2023

Task 2: Select, Where, Between, And, Like

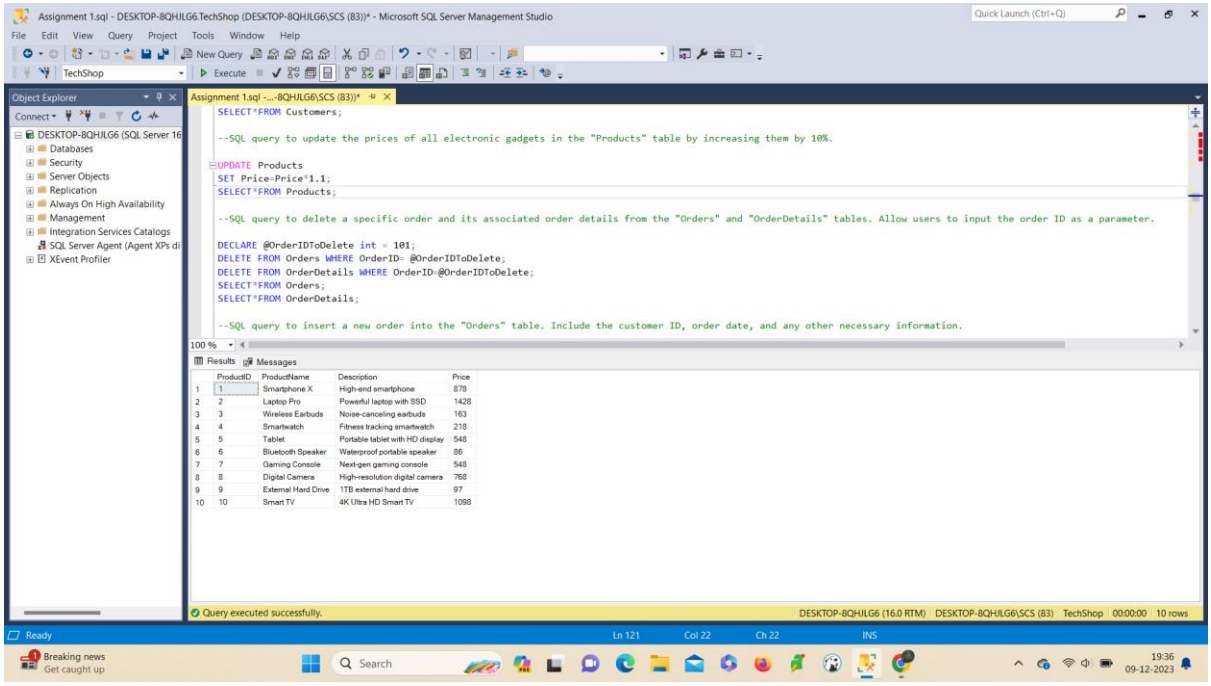
1.



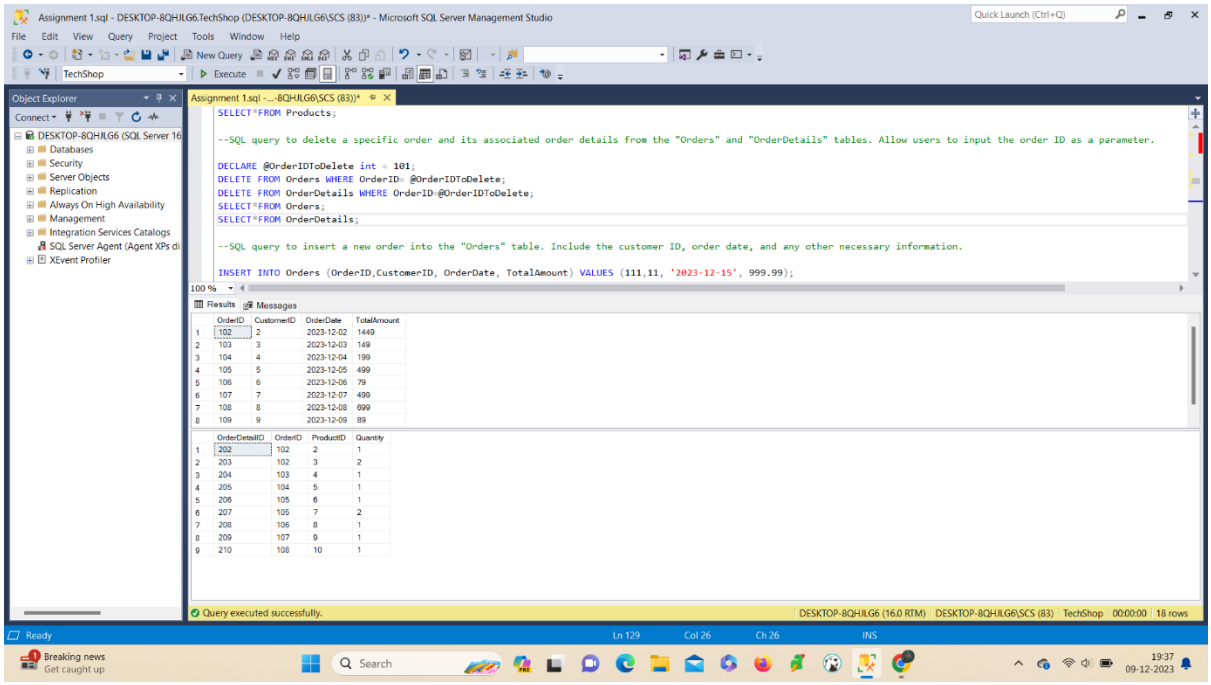
2.



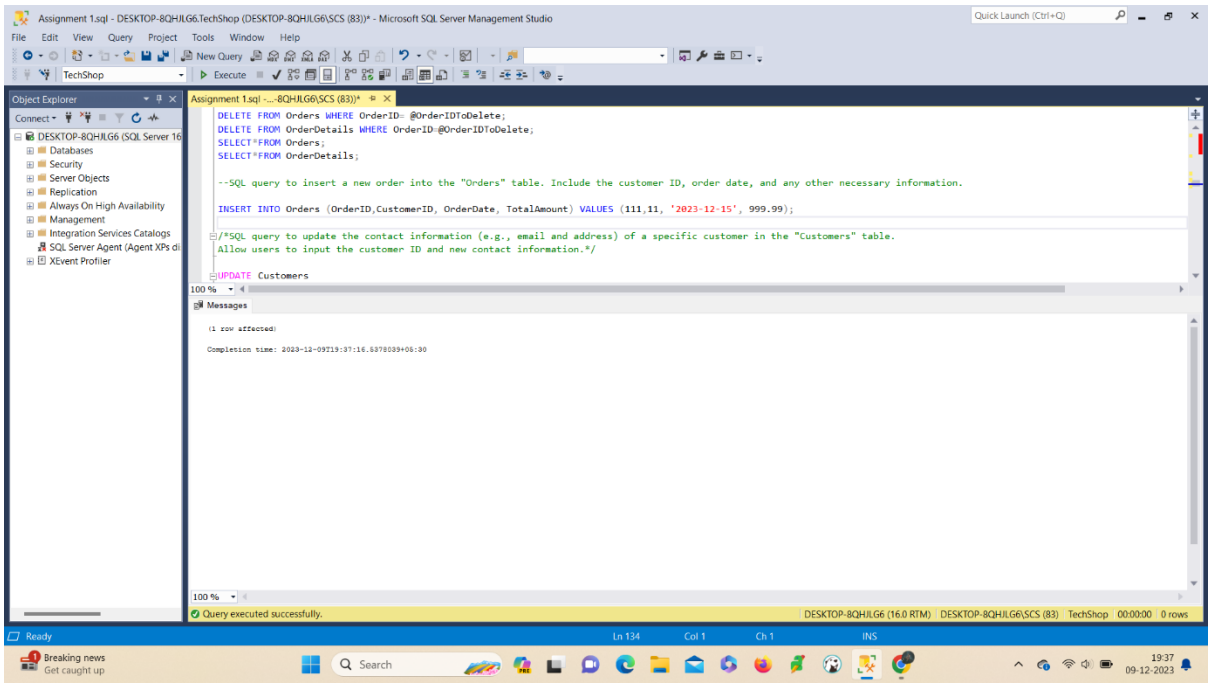
3.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
SELECT*FROM Customers;
--SQL query to update the prices of all electronic gadgets in the "Products" table by increasing them by 10%.
UPDATE Products
SET Price=Price*1.1;
SELECT*FROM Products;
--SQL query to delete a specific order and its associated order details from the "Orders" and "OrderDetails" tables. Allow users to input the order ID as a parameter.
DECLARE @OrderIDToDelete int = 101;
DELETE FROM Orders WHERE OrderID= @OrderIDToDelete;
DELETE FROM OrderDetails WHERE OrderID=@OrderIDToDelete;
SELECT*FROM Orders;
SELECT*FROM OrderDetails;
--SQL query to insert a new order into the "Orders" table. Include the customer ID, order date, and any other necessary information.
Query executed successfully.

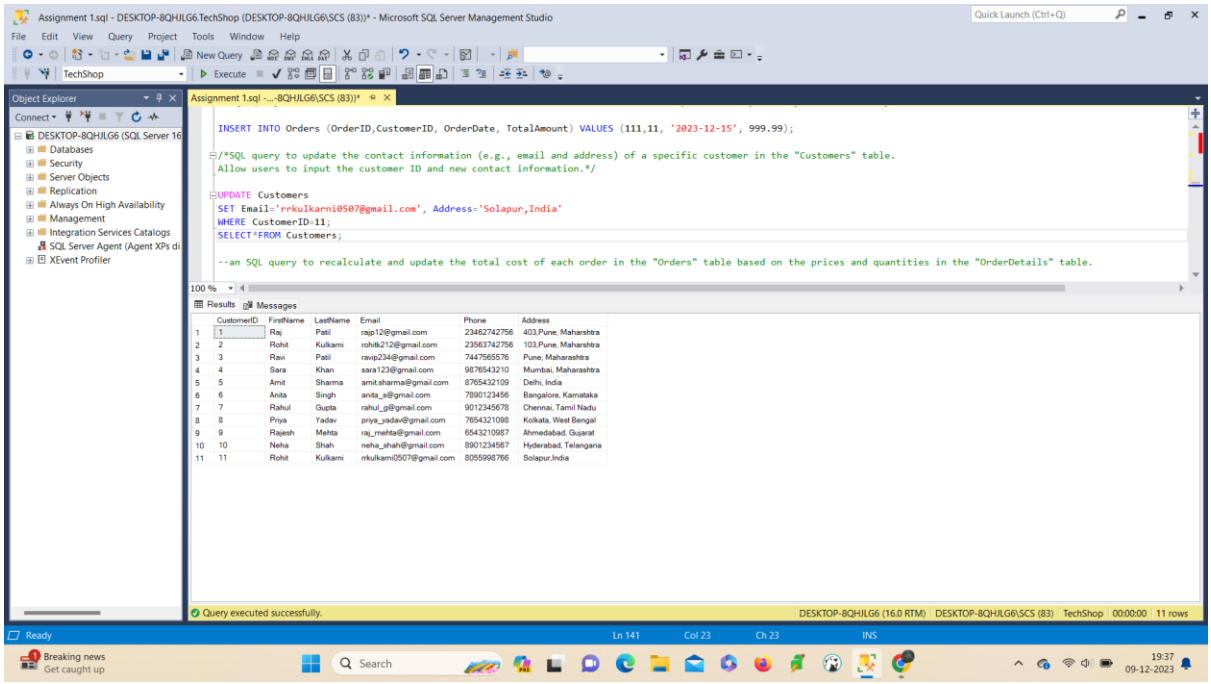
4.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
SELECT*FROM Products;
--SQL query to delete a specific order and its associated order details from the "Orders" and "OrderDetails" tables. Allow users to input the order ID as a parameter.
DECLARE @OrderIDToDelete int = 101;
DELETE FROM Orders WHERE OrderID= @OrderIDToDelete;
DELETE FROM OrderDetails WHERE OrderID=@OrderIDToDelete;
SELECT*FROM Orders;
SELECT*FROM OrderDetails;
--SQL query to insert a new order into the "Orders" table. Include the customer ID, order date, and any other necessary information.
INSERT INTO Orders (OrderID,CustomerID, OrderDate, TotalAmount) VALUES (111,11, '2023-12-15', 999.99);
Query executed successfully.

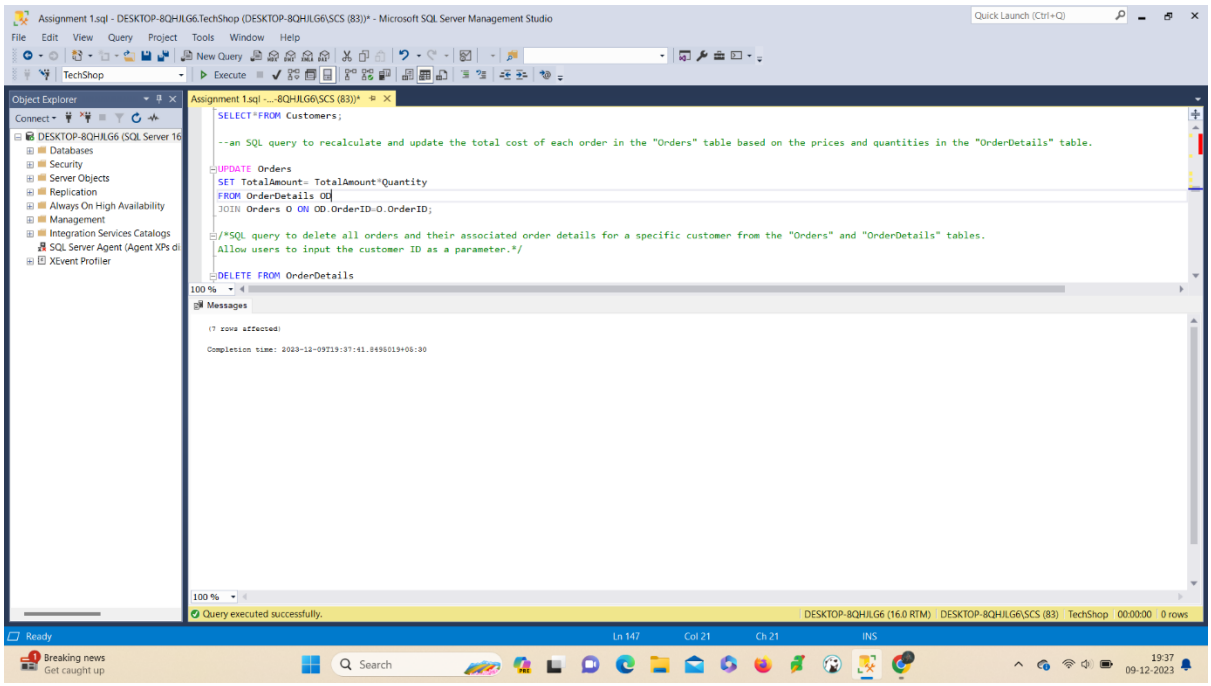
5.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
DELETE FROM Orders WHERE OrderID= @OrderIDToDelete;
DELETE FROM OrderDetails WHERE OrderID=@OrderIDToDelete;
SELECT*FROM Orders;
SELECT*FROM OrderDetails;
--SQL query to insert a new order into the "Orders" table. Include the customer ID, order date, and any other necessary information.
INSERT INTO Orders (OrderID,CustomerID, OrderDate, TotalAmount) VALUES (111,11, '2023-12-15', 999.99);
/*SQL query to update the contact information (e.g., email and address) of a specific customer in the "Customers" table.
Allow users to input the customer ID and new contact information.*/
UPDATE Customers
(1 row affected)
Completion time: 2023-12-09T19:37:16.8378039+05:30
Query executed successfully.

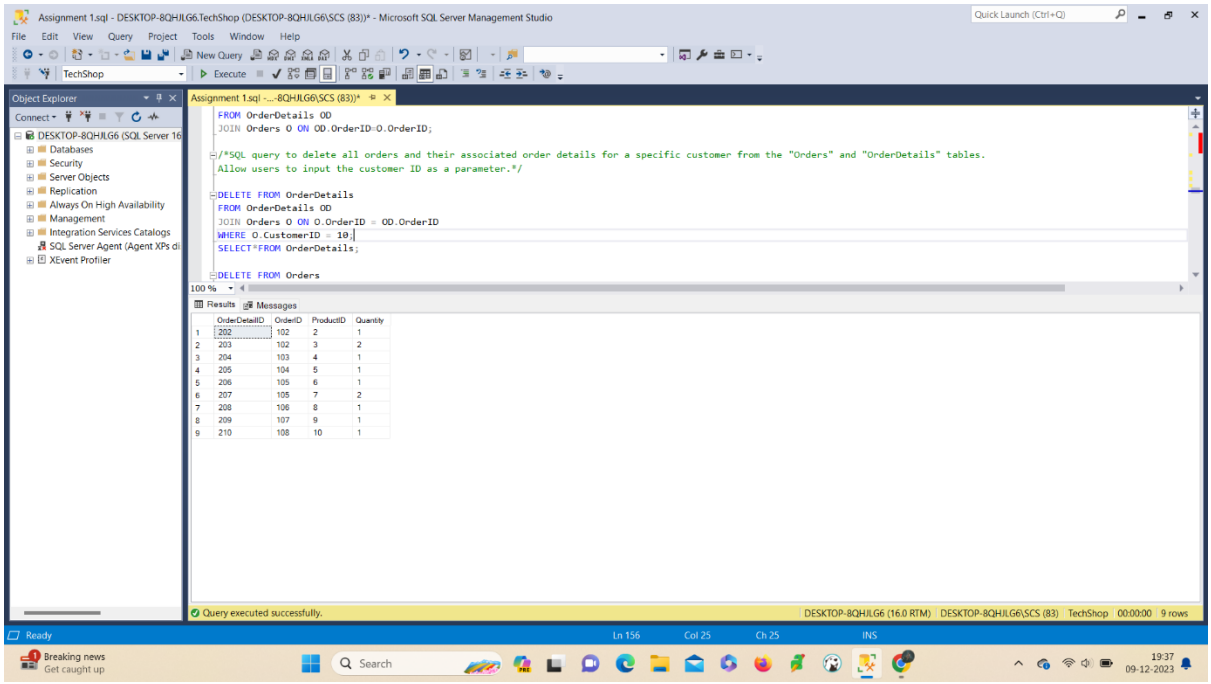
6.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
INSERT INTO Orders (OrderID,CustomerID, OrderDate, TotalAmount) VALUES (111,11, '2023-12-15', 999.99);
/*SQL query to update the contact information (e.g., email and address) of a specific customer in the "Customers" table.
Allow users to input the customer ID and new contact information.*/
UPDATE Customers
SET Email='rrkulkarni0507@gmail.com', Address='Solapur,India'
WHERE CustomerID=11;
SELECT*FROM Customers;
--an SQL query to recalculate and update the total cost of each order in the "Orders" table based on the prices and quantities in the "OrderDetails" table.
Query executed successfully.

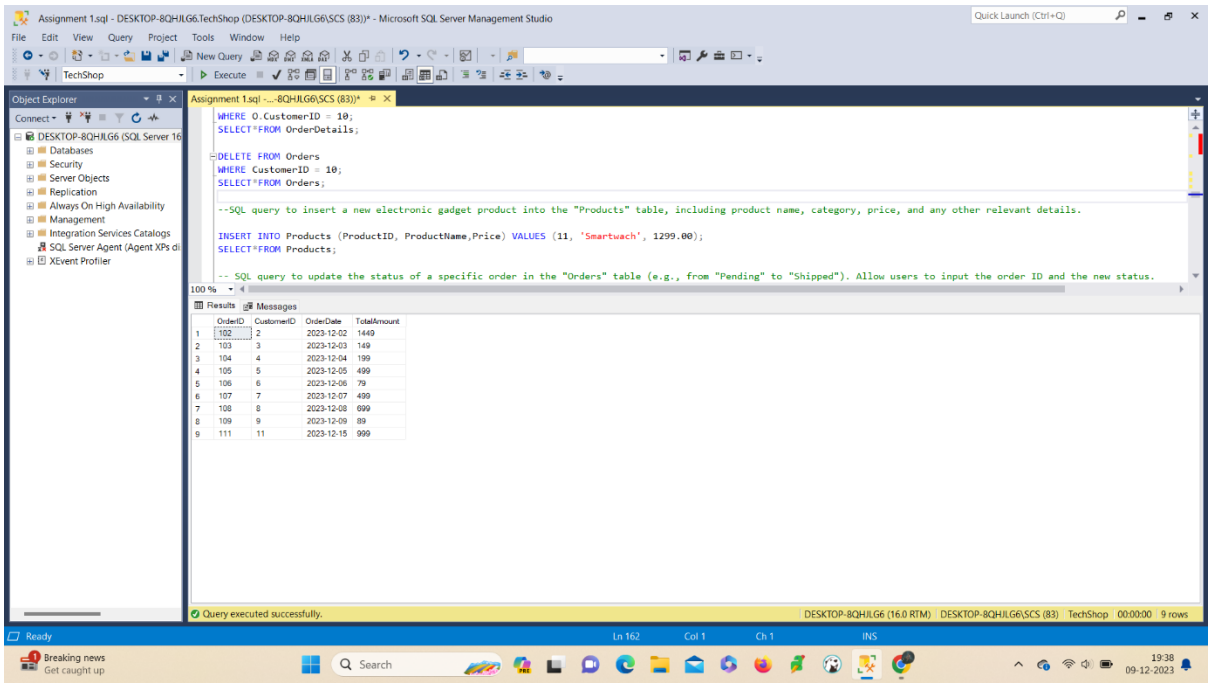
7.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
SELECT*FROM Customers;
--an SQL query to recalculate and update the total cost of each order in the "Orders" table based on the prices and quantities in the "OrderDetails" table.
UPDATE Orders
SET TotalAmount= TotalAmount*Quantity
FROM OrderDetails OD
JOIN Orders O ON OD.OrderID=O.OrderID;
/*SQL query to delete all orders and their associated order details for a specific customer from the "Orders" and "OrderDetails" tables.
Allow users to input the customer ID as a parameter.*/
DELETE FROM OrderDetails
(7 rows affected)
Completion time: 2023-12-09T19:37:41.8496019+05:30
Query executed successfully.

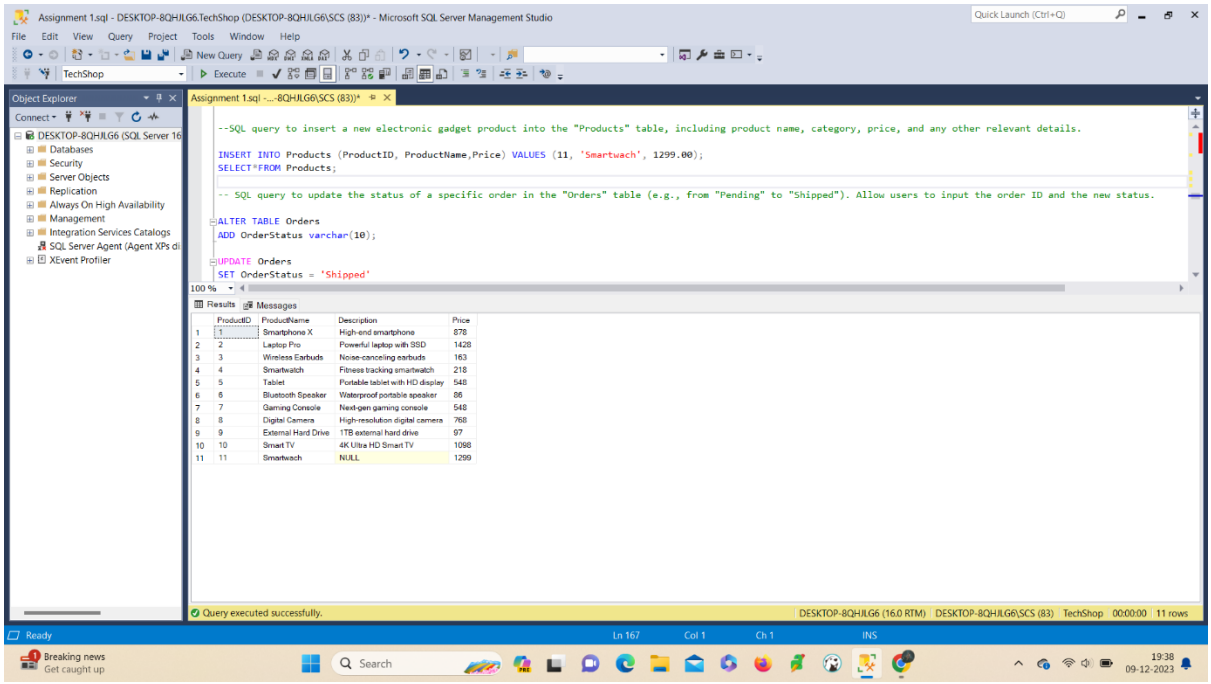
8.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
FROM OrderDetails OD
JOIN Orders O ON OD.OrderID=O.OrderID;
/*SQL query to delete all orders and their associated order details for a specific customer from the "Orders" and "OrderDetails" tables.
Allow users to input the customer ID as a parameter.*/
DELETE FROM OrderDetails
FROM OrderDetails OD
JOIN Orders O ON O.OrderID = OD.OrderID
WHERE O.CustomerID = 10;
SELECT*FROM OrderDetails;
DELETE FROM Orders
OrderDetailID OrderID ProductID Quantity
202 102 2 1
203 102 3 2
204 103 4 1
205 104 5 1
206 105 6 1
207 105 7 2
208 106 8 1
209 107 9 1
210 108 10 1
Query executed successfully.

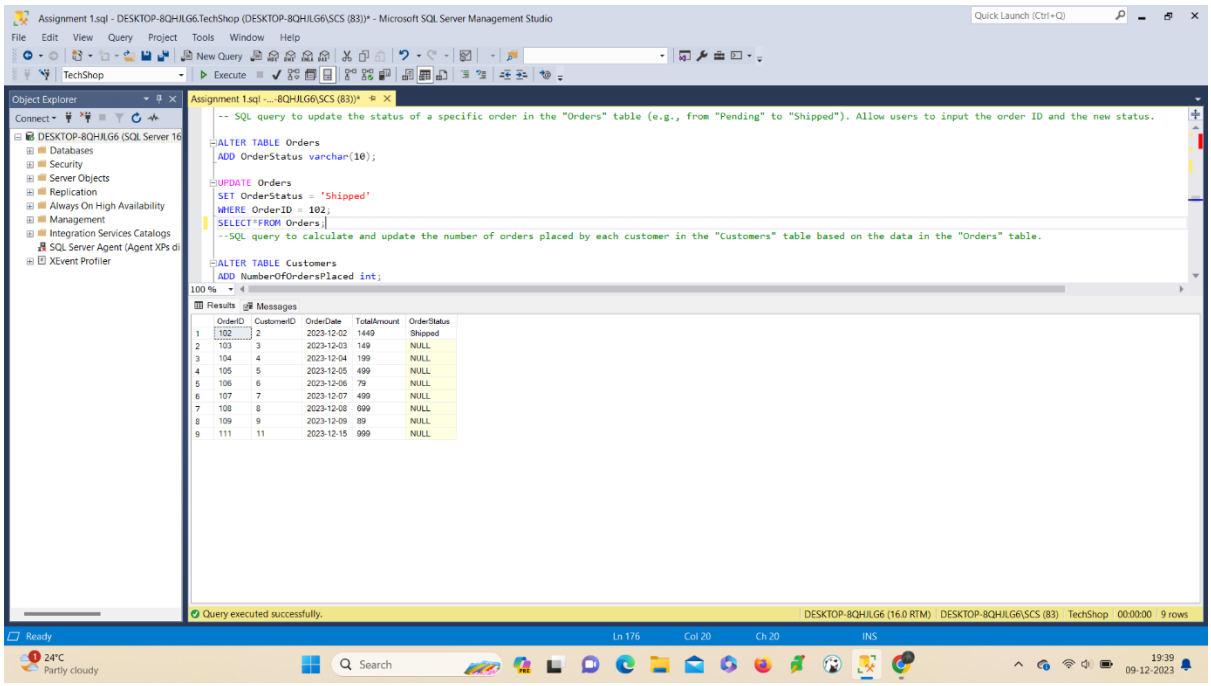
9.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
WHERE O.CustomerID = 10;
SELECT*FROM OrderDetails;
DELETE FROM Orders
WHERE CustomerID = 10;
SELECT*FROM Orders;
--SQL query to insert a new electronic gadget product into the "Products" table, including product name, category, price, and any other relevant details.
INSERT INTO Products (ProductID, ProductName,Price) VALUES (11, 'Smartwach', 1299.00);
SELECT*FROM Products;
-- SQL query to update the status of a specific order in the "Orders" table (e.g., from "Pending" to "Shipped"). Allow users to input the order ID and the new status.
Query executed successfully.

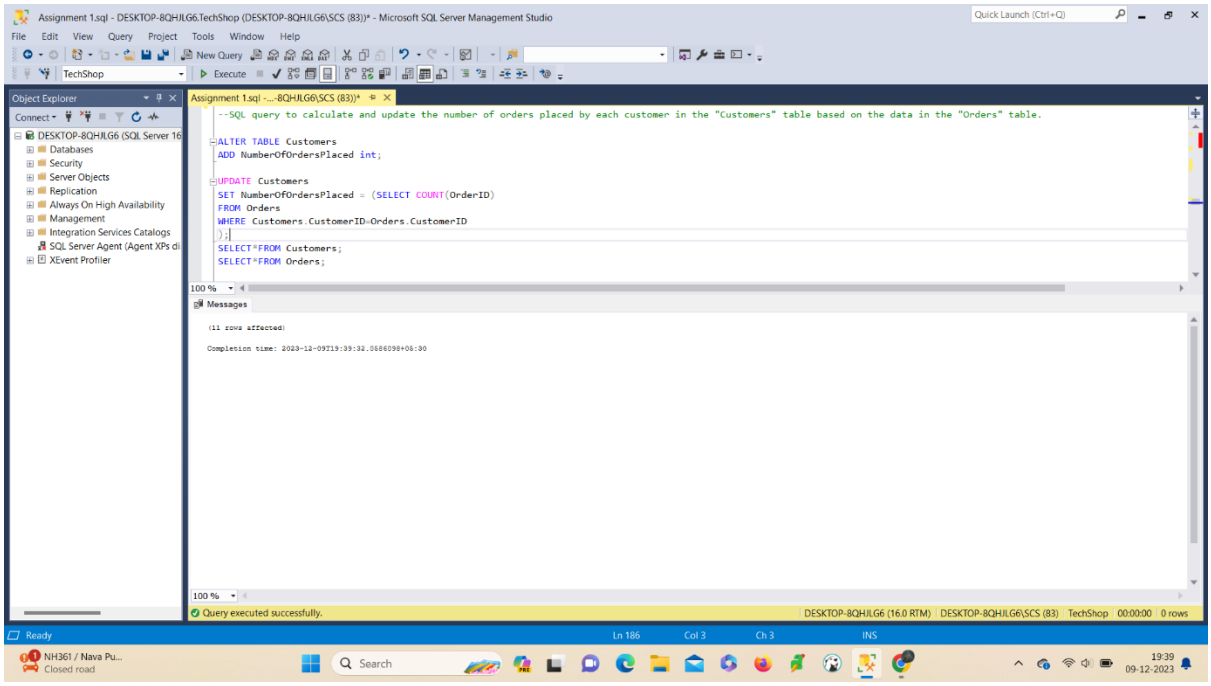
10.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
--SQL query to insert a new electronic gadget product into the "Products" table, including product name, category, price, and any other relevant details.
INSERT INTO Products (ProductID, ProductName,Price) VALUES (11, 'Smartwach', 1299.00);
SELECT*FROM Products;
-- SQL query to update the status of a specific order in the "Orders" table (e.g., from "Pending" to "Shipped"). Allow users to input the order ID and the new status.
ALTER TABLE Orders
ADD OrderStatus varchar(10);
UPDATE Orders
SET OrderStatus = 'Shipped'
Query executed successfully.

11.

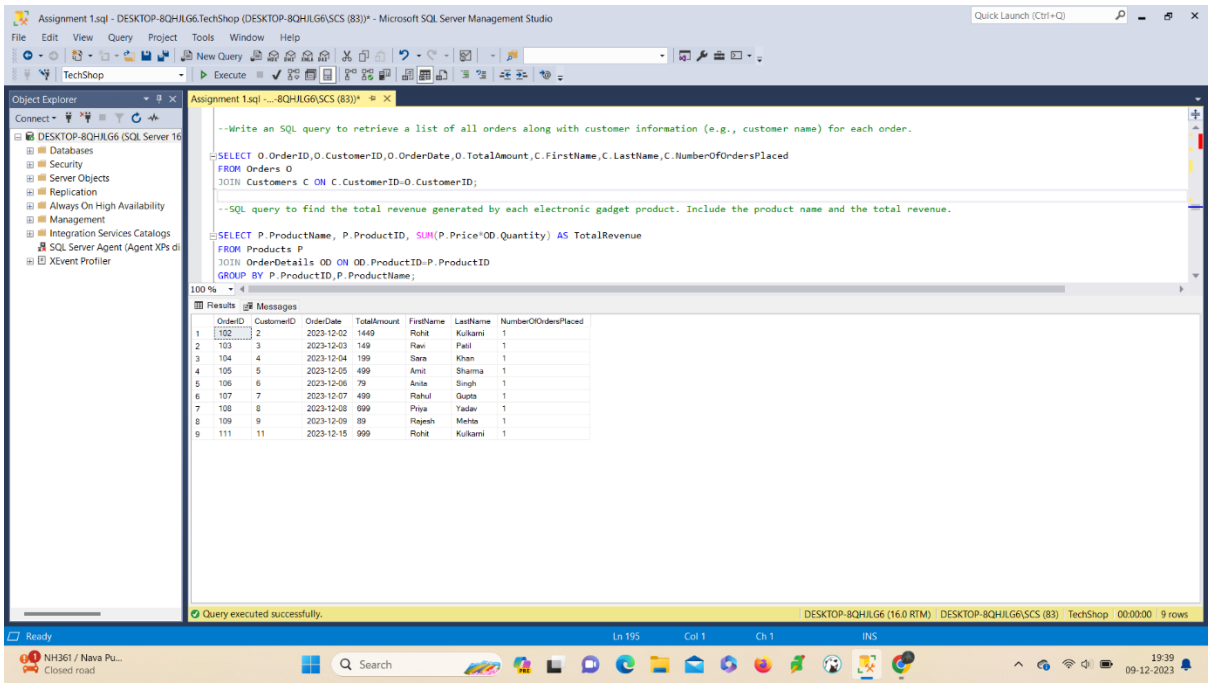

Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
-- SQL query to update the status of a specific order in the "Orders" table (e.g., from "Pending" to "Shipped"). Allow users to input the order ID and the new status.
ALTER TABLE Orders
ADD OrderStatus varchar(10);
UPDATE Orders
SET OrderStatus = 'Shipped'
WHERE OrderID = 102;
SELECT*FROM Orders;
--SQL query to calculate and update the number of orders placed by each customer in the "Customers" table based on the data in the "Orders" table.
ALTER TABLE Customers
ADD NumberOfOrdersPlaced int;
OrderID CustomerID OrderDate TotalAmount OrderStatus
1 102 2 2023-12-02 1449 Shipped
2 103 3 2023-12-03 149 NULL
3 104 4 2023-12-04 199 NULL
4 105 5 2023-12-05 499 NULL
5 106 6 2023-12-06 79 NULL
6 107 7 2023-12-07 499 NULL
7 108 8 2023-12-08 699 NULL
8 109 9 2023-12-09 89 NULL
9 111 11 2023-12-15 999 NULL
Query executed successfully.

12.

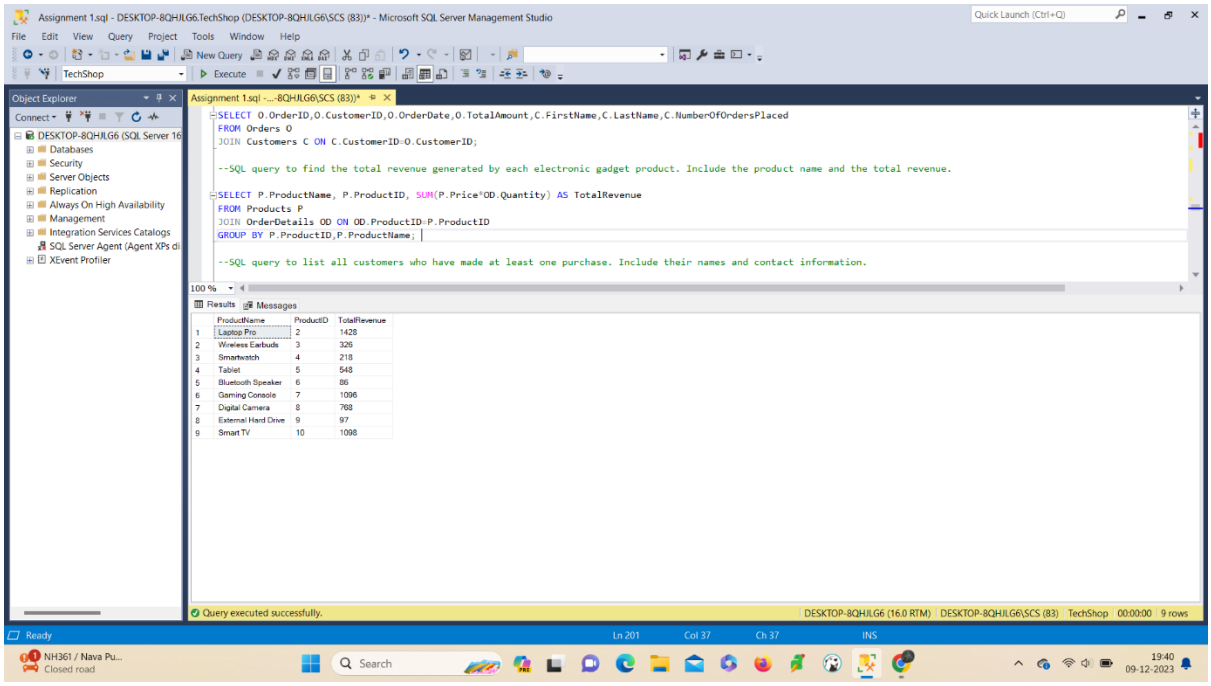

Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
--SQL query to calculate and update the number of orders placed by each customer in the "Customers" table based on the data in the "Orders" table.
ALTER TABLE Customers
ADD NumberOfOrdersPlaced int;
UPDATE Customers
SET NumberOfOrdersPlaced = (SELECT COUNT(OrderID)
FROM Orders
WHERE Customers.CustomerID=Orders.CustomerID
);
SELECT*FROM Customers;
SELECT*FROM Orders;
(11 rows affected)
Completion time: 2023-12-09T19:39:32.0586098+05:30
Query executed successfully.

Task 3: Aggregate functions, Having, Order By, Group By

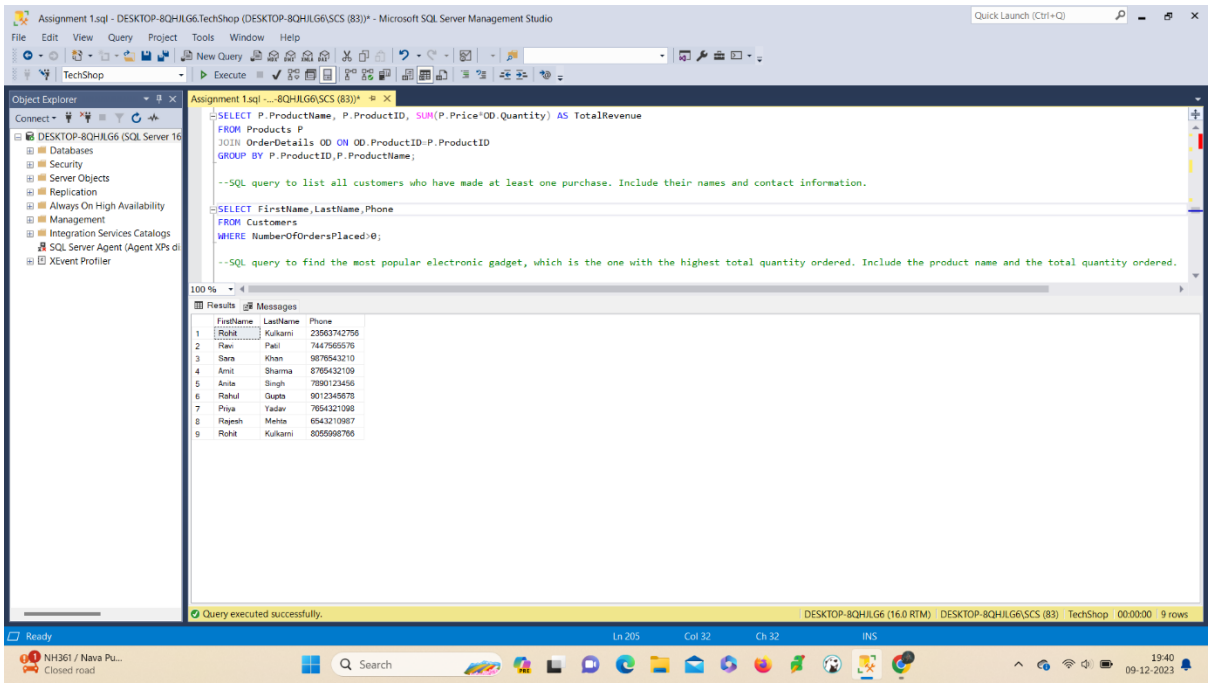
1.



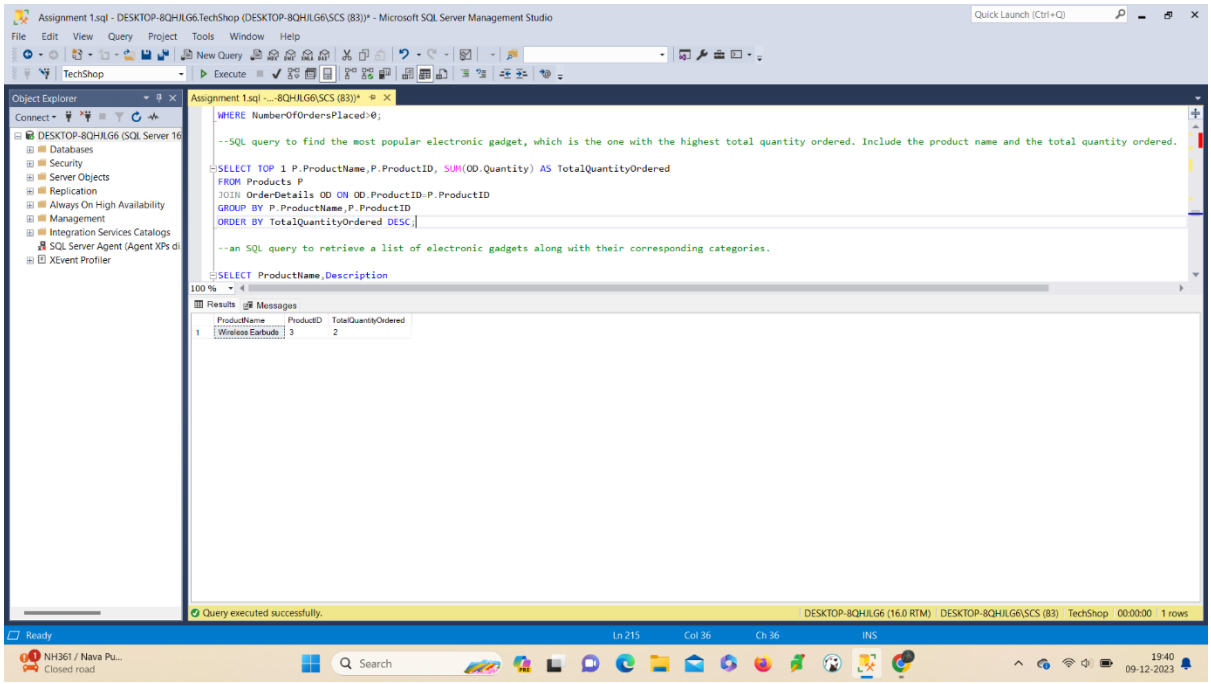
2.



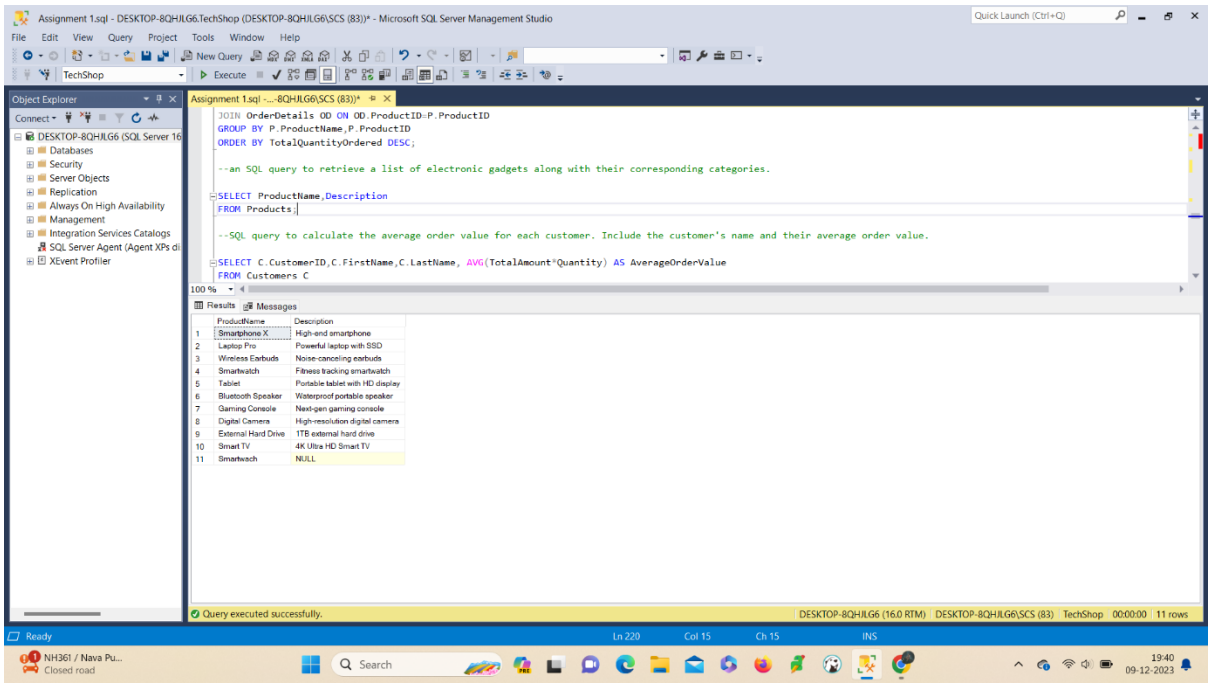
3.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
SELECT P.ProductName, P.ProductID, SUM(P.Price*OD.Quantity) AS TotalRevenue
FROM Products P
JOIN OrderDetails OD ON OD.ProductID=P.ProductID
GROUP BY P.ProductID,P.ProductName;
--SQL query to list all customers who have made at least one purchase. Include their names and contact information.
SELECT FirstName,LastName,Phone
FROM Customers
WHERE NumberOfOrdersPlaced>0;
--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.
FirstName LastName Phone
1 Rohit Kulkarni 23563742756
2 Ravi Patil 7447565576
3 Sara Khan 9876543210
4 Amit Sharma 8765432109
5 Anita Singh 7890123456
6 Rahul Gupta 9012345678
7 Priya Yadav 7654321098
8 Rajesh Mehta 6543210987
9 Rohit Kulkarni 8055998766
Query executed successfully.

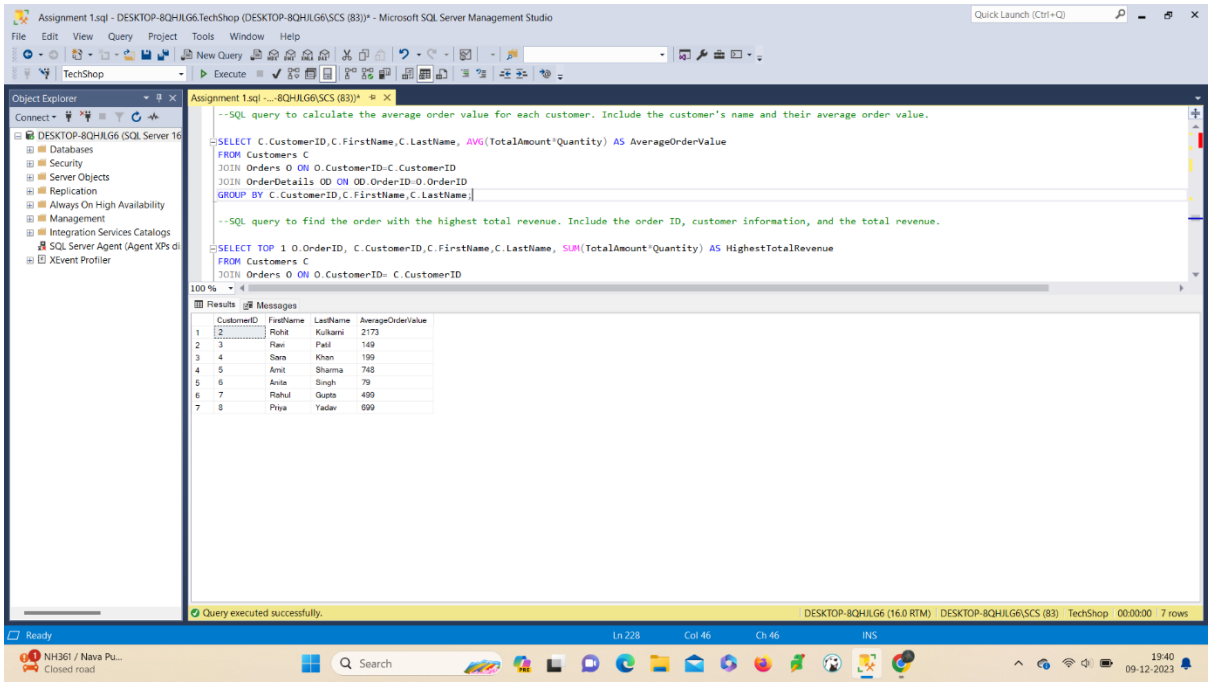
4.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
WHERE NumberOfOrdersPlaced>0;
--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 P.ProductName,P.ProductID, SUM(OD.Quantity) AS TotalQuantityOrdered
FROM Products P
JOIN OrderDetails OD ON OD.ProductID=P.ProductID
GROUP BY P.ProductName,P.ProductID
ORDER BY TotalQuantityOrdered DESC;
--an SQL query to retrieve a list of electronic gadgets along with their corresponding categories.
SELECT ProductName,Description
ProductName ProductID TotalQuantityOrdered
1 Wireless Earbuds 3 2
Query executed successfully.

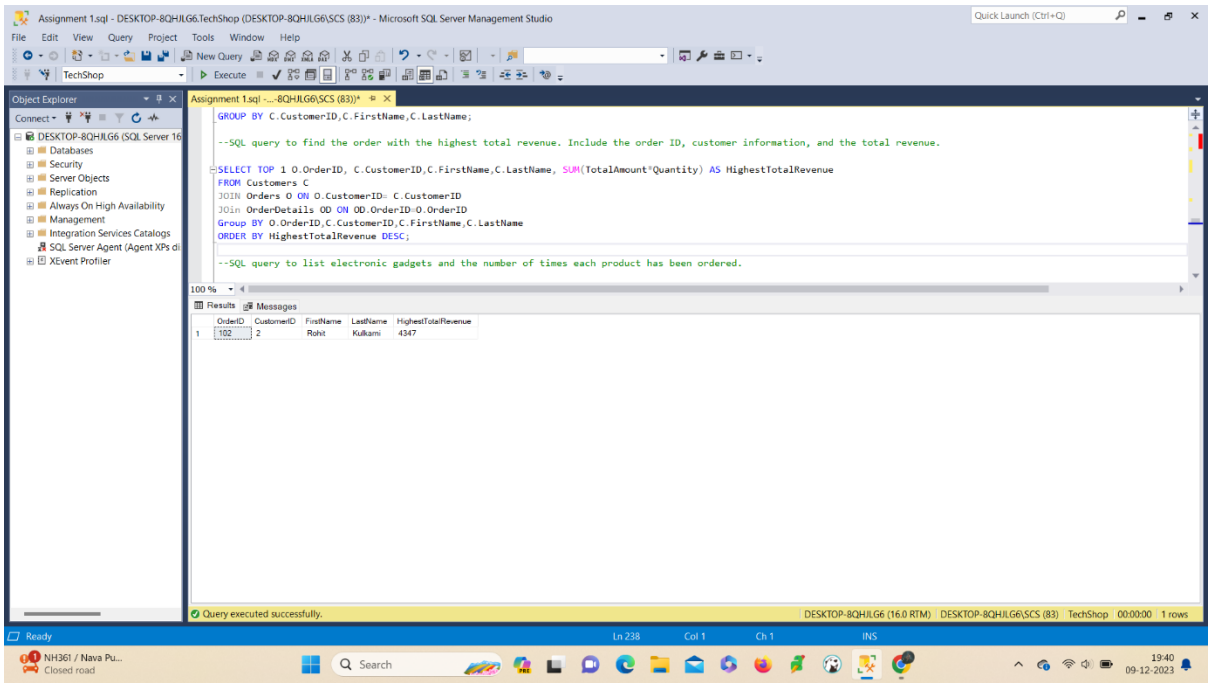
5.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
JOIN OrderDetails OD ON OD.ProductID=P.ProductID
GROUP BY P.ProductName,P.ProductID
ORDER BY TotalQuantityOrdered DESC;
--an SQL query to retrieve a list of electronic gadgets along with their corresponding categories.
SELECT ProductName,Description
FROM Products;
--SQL query to calculate the average order value for each customer. Include the customer's name and their average order value.
SELECT C.CustomerID,C.FirstName,C.LastName, AVG(TotalAmount*Quantity) AS AverageOrderValue
FROM Customers C
Query executed successfully.

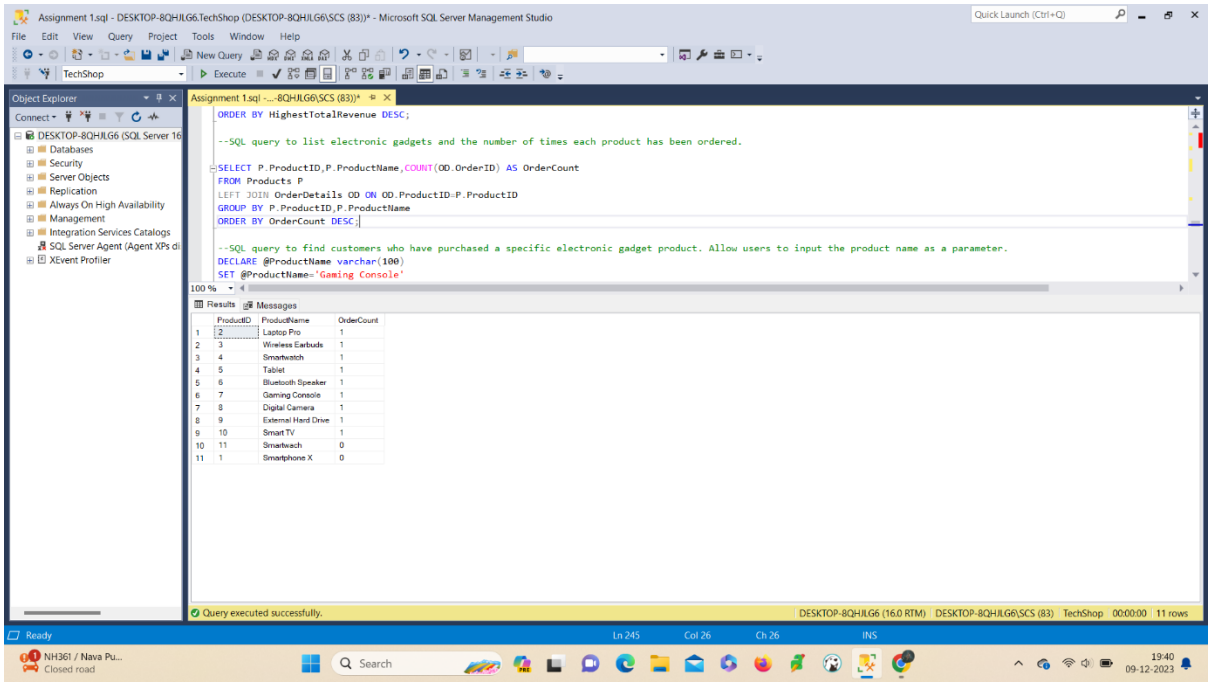
6.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
--SQL query to calculate the average order value for each customer. Include the customer's name and their average order value.
SELECT C.CustomerID,C.FirstName,C.LastName, AVG(TotalAmount*Quantity) AS AverageOrderValue
FROM Customers C
JOIN Orders O ON O.CustomerID=C.CustomerID
JOIN OrderDetails OD ON OD.OrderID=O.OrderID
GROUP BY C.CustomerID,C.FirstName,C.LastName;
--SQL query to find the order with the highest total revenue. Include the order ID, customer information, and the total revenue.
SELECT TOP 1 O.OrderID, C.CustomerID,C.FirstName,C.LastName, SUM(TotalAmount*Quantity) AS HighestTotalRevenue
FROM Customers C
JOIN Orders O ON O.CustomerID= C.CustomerID
Query executed successfully.

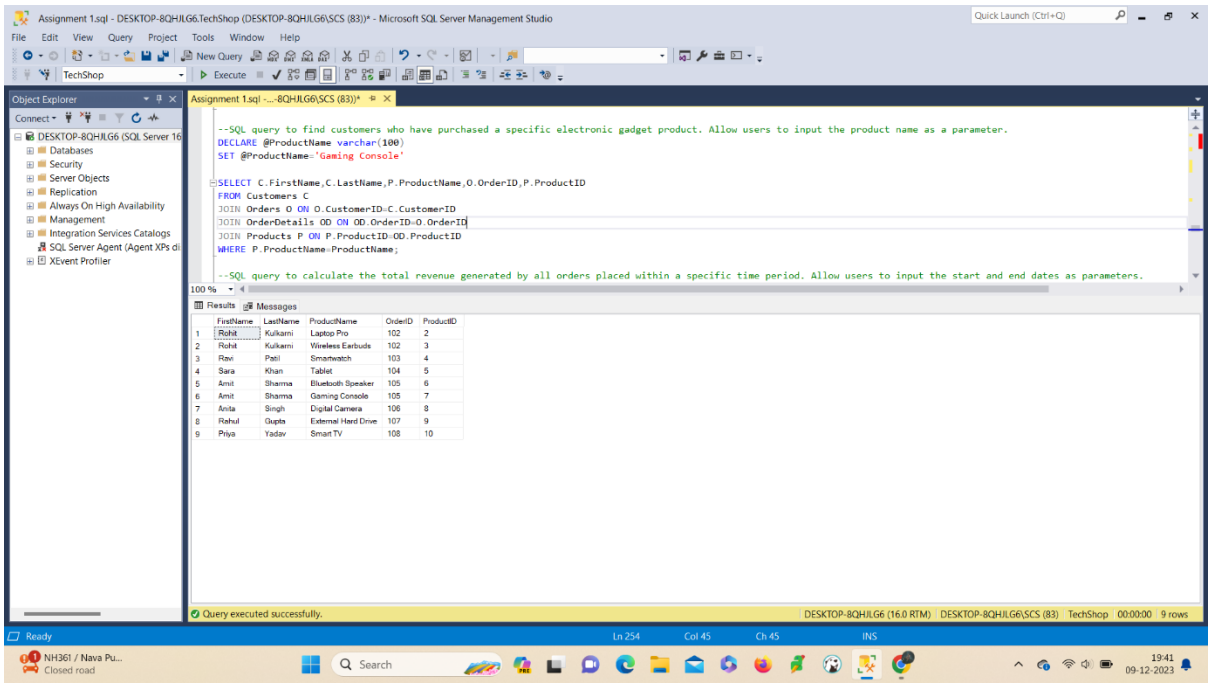
7.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
GROUP BY C.CustomerID,C.FirstName,C.LastName;
--SQL query to find the order with the highest total revenue. Include the order ID, customer information, and the total revenue.
SELECT TOP 1 O.OrderID, C.CustomerID,C.FirstName,C.LastName, SUM(TotalAmount*Quantity) AS HighestTotalRevenue
FROM Customers C
JOIN Orders O ON O.CustomerID= C.CustomerID
JOin OrderDetails OD ON OD.OrderID=O.OrderID
Group BY O.OrderID,C.CustomerID,C.FirstName,C.LastName
ORDER BY HighestTotalRevenue DESC;
--SQL query to list electronic gadgets and the number of times each product has been ordered.
OrderID CustomerID FirstName LastName HighestTotalRevenue
1 102 2 Rohit Kulkarni 4347
Query executed successfully.

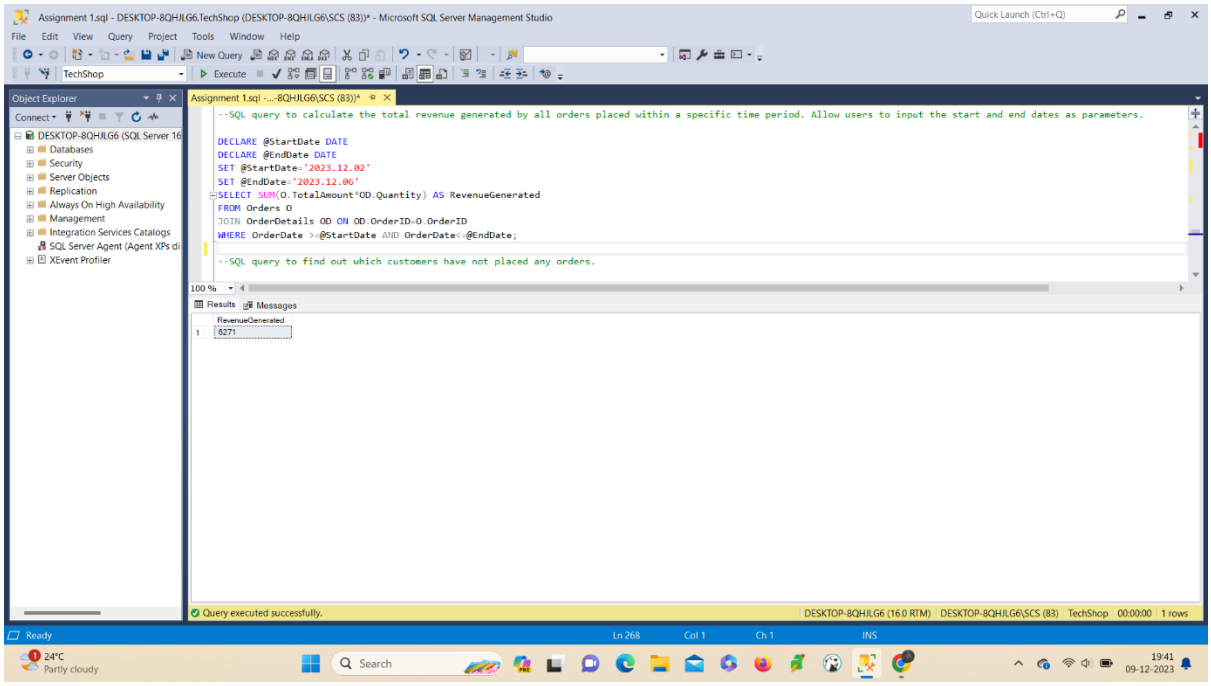
8.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
ORDER BY HighestTotalRevenue DESC;
--SQL query to list electronic gadgets and the number of times each product has been ordered.
SELECT P.ProductID,P.ProductName,COUNT(OD.OrderID) AS OrderCount
FROM Products P
LEFT JOIN OrderDetails OD ON OD.ProductID=P.ProductID
GROUP BY P.ProductID,P.ProductName
ORDER BY OrderCount DESC;
--SQL query to find customers who have purchased a specific electronic gadget product. Allow users to input the product name as a parameter.
DECLARE @ProductName varchar(100)
SET @ProductName='Gaming Console'
ProductID ProductName OrderCount
1 2 Laptop Pro 1
2 3 Wireless Earbuds 1
3 4 Smartwatch 1
4 5 Tablet 1
5 6 Bluetooth Speaker 1
6 7 Gaming Console 1
7 8 Digital Camera 1
8 9 External Hard Drive 1
9 10 Smart TV 1
10 11 Smartwach 0
11 1 Smartphone X 0
Query executed successfully.

9.

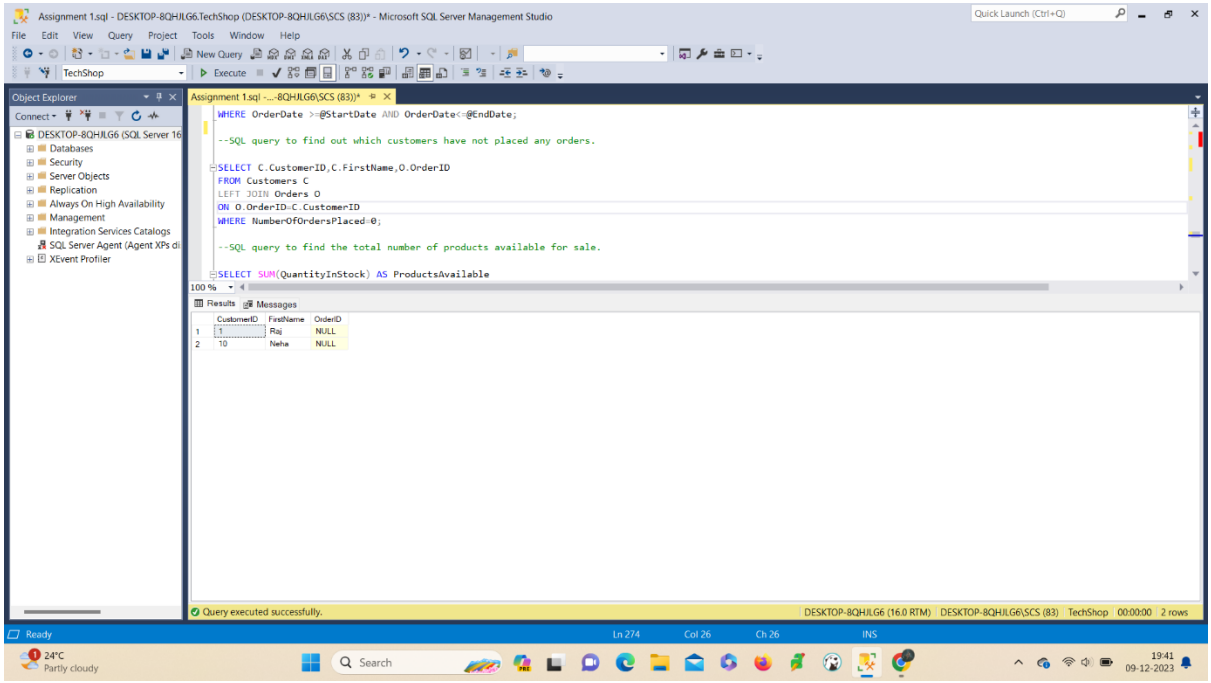

Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
--SQL query to find customers who have purchased a specific electronic gadget product. Allow users to input the product name as a parameter.
DECLARE @ProductName varchar(100)
SET @ProductName='Gaming Console'
SELECT C.FirstName,C.LastName,P.ProductName,O.OrderID,P.ProductID
FROM Customers C
JOIN Orders O ON O.CustomerID=C.CustomerID
JOIN OrderDetails OD ON OD.OrderID=O.OrderID
JOIN Products P ON P.ProductID=OD.ProductID
WHERE P.ProductName=ProductName;
--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.
FirstName LastName ProductName OrderID ProductID
1 Rohit Kulkarni Laptop Pro 102 2
2 Rohit Kulkarni Wireless Earbuds 102 3
3 Ravi Patil Smartwatch 103 4
4 Sara Khan Tablet 104 5
5 Amit Sharma Bluetooth Speaker 105 6
6 Amit Sharma Gaming Console 105 7
7 Anita Singh Digital Camera 106 8
8 Rahul Gupta External Hard Drive 107 9
9 Priya Yadav Smart TV 108 10
Query executed successfully.

10.

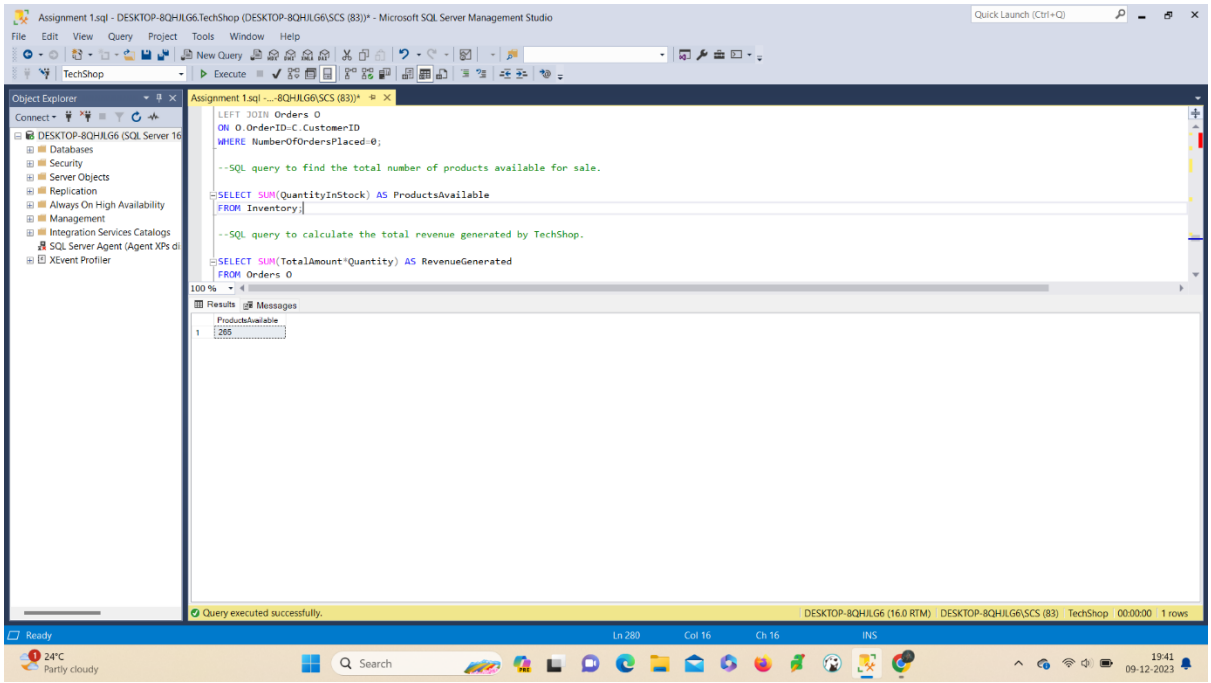

Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
--SQL query to calculate the total revenue generated by all orders placed within a specific time period. Allow users to input the start and end dates as parameters.
DECLARE @StartDate DATE
DECLARE @EndDate DATE
SET @StartDate='2023.12.02'
SET @EndDate='2023.12.06'
SELECT SUM(O.TotalAmount*OD.Quantity) AS RevenueGenerated
FROM Orders O
JOIN OrderDetails OD ON OD.OrderID=O.OrderID
WHERE OrderDate >=@StartDate AND OrderDate<=@EndDate;
--SQL query to find out which customers have not placed any orders.
RevenueGenerated
1 6271
Query executed successfully.

Task 4: Subquery and its Types

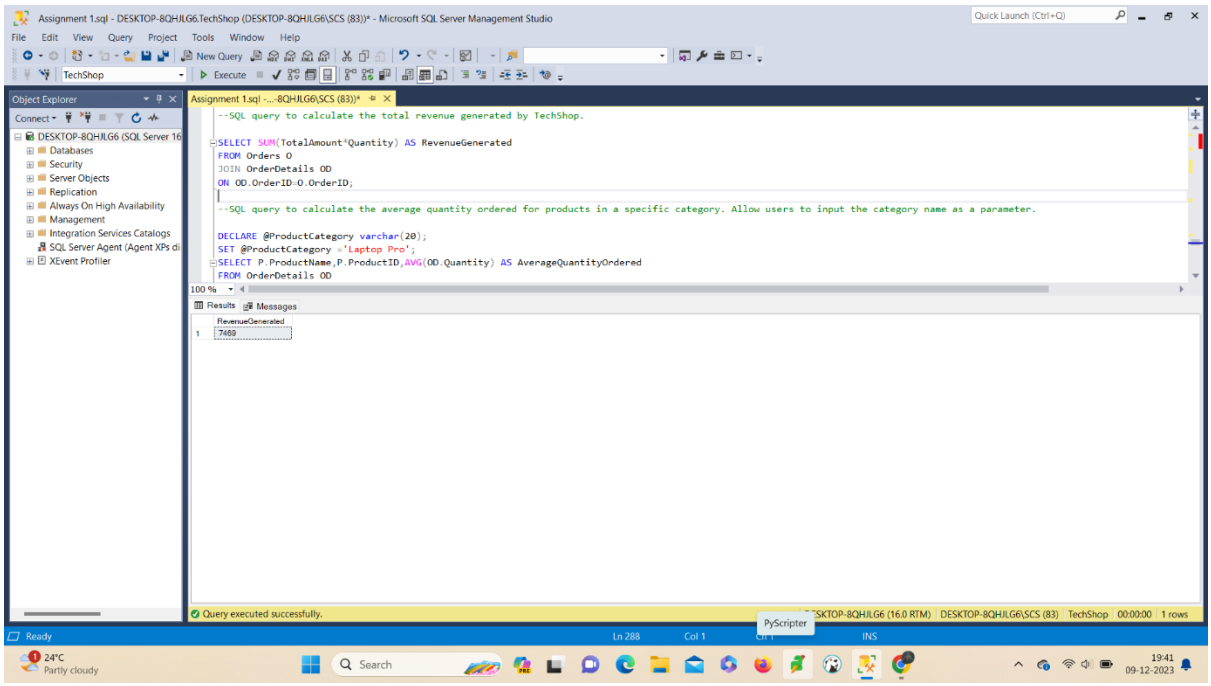
1.



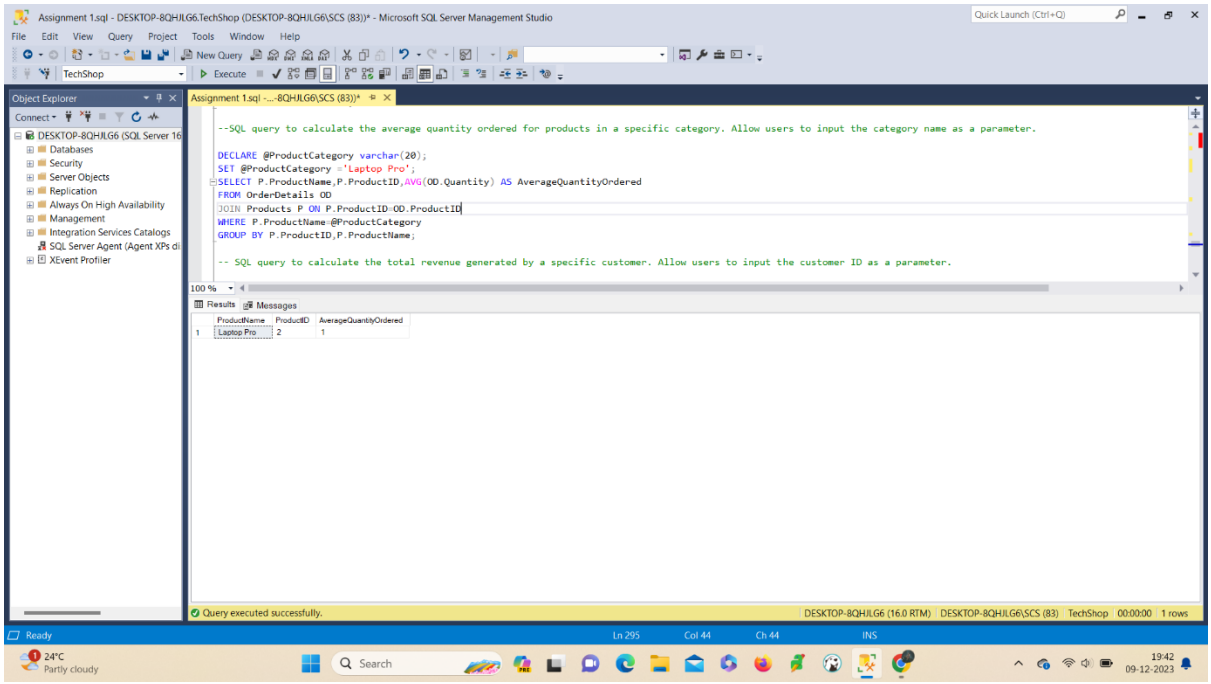
2.



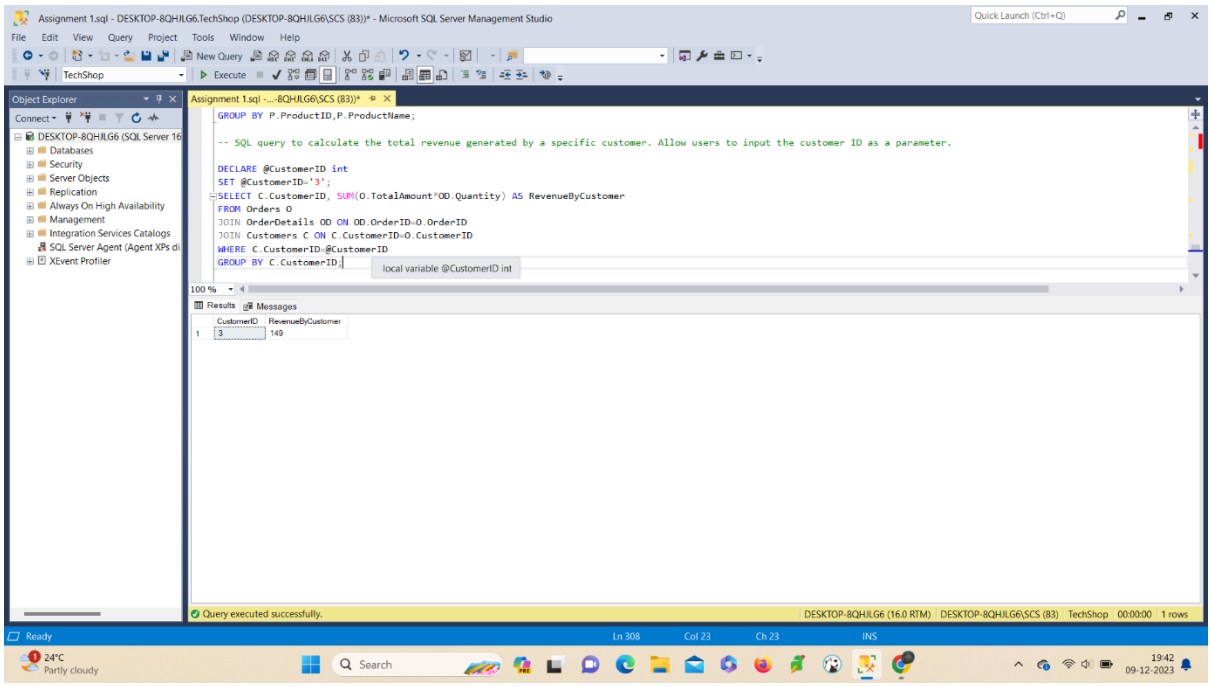
3.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
--SQL query to calculate the total revenue generated by TechShop.
SELECT SUM(TotalAmount*Quantity) AS RevenueGenerated
FROM Orders O
JOIN OrderDetails OD
ON OD.OrderID=O.OrderID;
--SQL query to calculate the average quantity ordered for products in a specific category. Allow users to input the category name as a parameter.
DECLARE @ProductCategory varchar(20);
SET @ProductCategory ='Laptop Pro';
SELECT P.ProductName,P.ProductID,AVG(OD.Quantity) AS AverageQuantityOrdered
FROM OrderDetails OD
RevenueGenerated
7469
Query executed successfully.

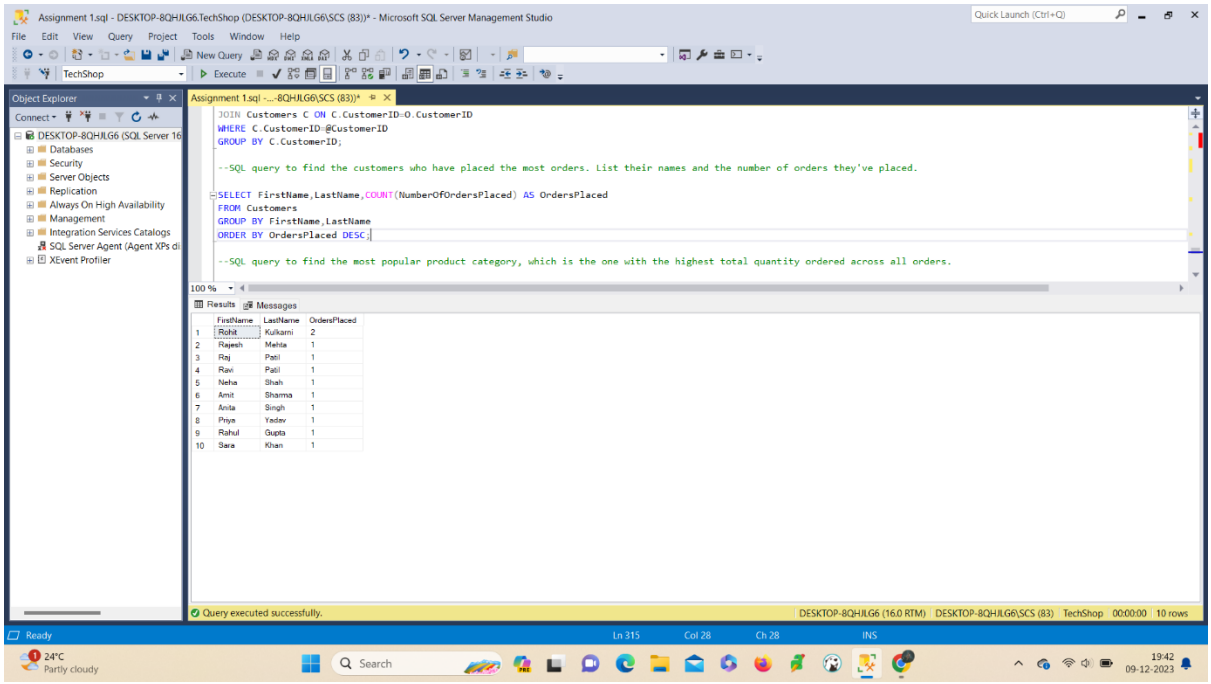
4.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
--SQL query to calculate the average quantity ordered for products in a specific category. Allow users to input the category name as a parameter.
DECLARE @ProductCategory varchar(20);
SET @ProductCategory ='Laptop Pro';
SELECT P.ProductName,P.ProductID,AVG(OD.Quantity) AS AverageQuantityOrdered
FROM OrderDetails OD
JOIN Products P ON P.ProductID=OD.ProductID
WHERE P.ProductName=@ProductCategory
GROUP BY P.ProductID,P.ProductName;
-- SQL query to calculate the total revenue generated by a specific customer. Allow users to input the customer ID as a parameter.
ProductName ProductID AverageQuantityOrdered
Laptop Pro 2 1
Query executed successfully.

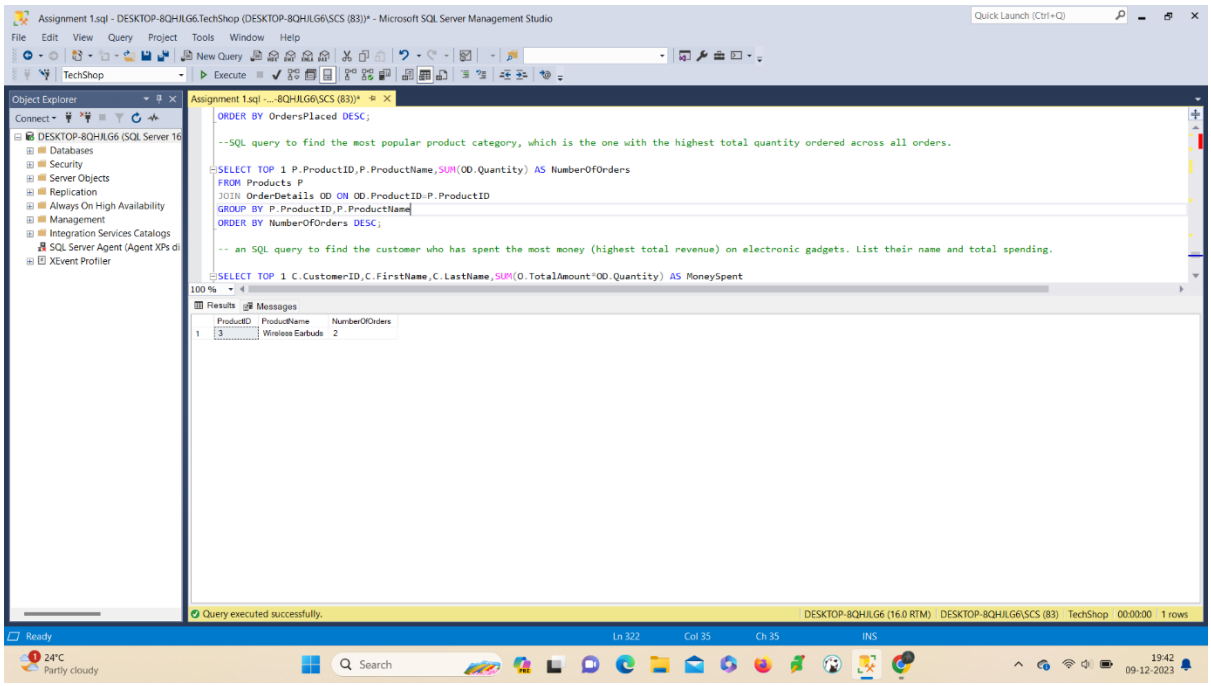
5.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
GROUP BY P.ProductID,P.ProductName;
-- SQL query to calculate the total revenue generated by a specific customer. Allow users to input the customer ID as a parameter.
DECLARE @CustomerID int
SET @CustomerID='3';
SELECT C.CustomerID, SUM(O.TotalAmount*OD.Quantity) AS RevenueByCustomer
FROM Orders O
JOIN OrderDetails OD ON OD.OrderID=O.OrderID
JOIN Customers C ON C.CustomerID=O.CustomerID
WHERE C.CustomerID=@CustomerID
GROUP BY C.CustomerID;
local variable @CustomerID int
CustomerID RevenueByCustomer
3 149
Query executed successfully.

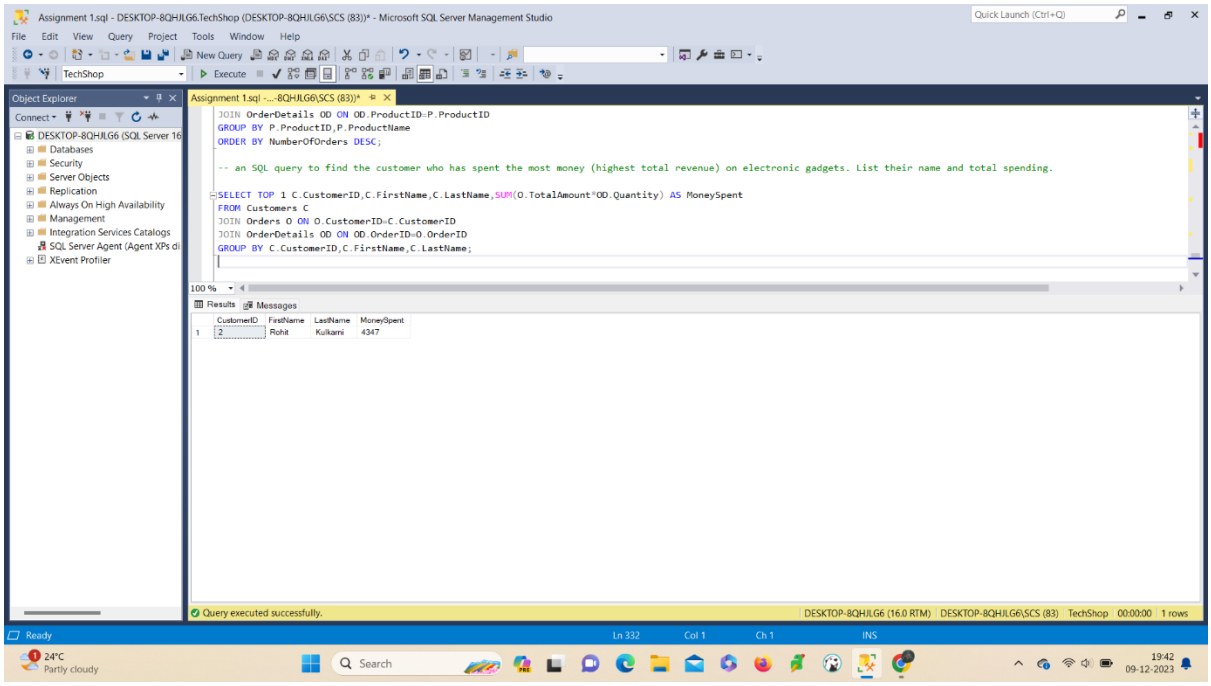
6.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
JOIN Customers C ON C.CustomerID=O.CustomerID
WHERE C.CustomerID=@CustomerID
GROUP BY C.CustomerID;
--SQL query to find the customers who have placed the most orders. List their names and the number of orders they've placed.
SELECT FirstName,LastName,COUNT(NumberOfOrdersPlaced) AS OrdersPlaced
FROM Customers
GROUP BY FirstName,LastName
ORDER BY OrdersPlaced DESC;
--SQL query to find the most popular product category, which is the one with the highest total quantity ordered across all orders.
FirstName LastName OrdersPlaced
1 Rohit Kulkarni 2
2 Rajesh Mehta 1
3 Raj Patil 1
4 Ravi Patil 1
5 Neha Shah 1
6 Amit Sharma 1
7 Anita Singh 1
8 Priya Yadav 1
9 Rahul Gupta 1
10 Sara Khan 1
Query executed successfully.

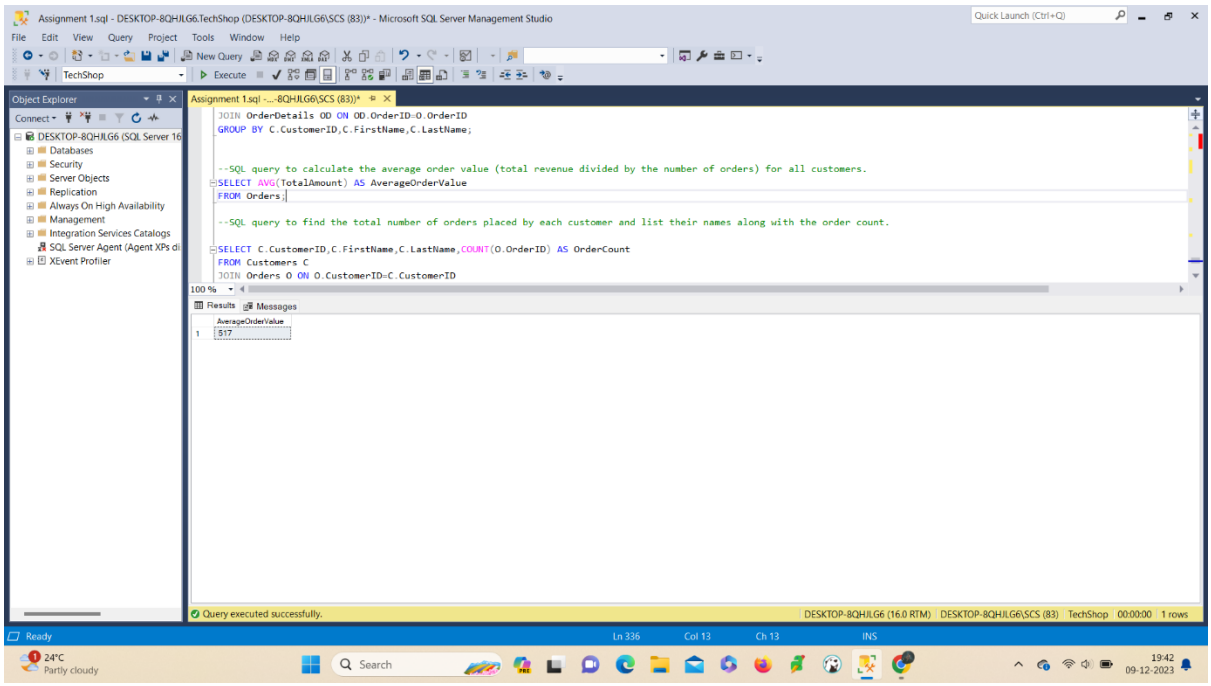
7.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
ORDER BY OrdersPlaced DESC;
--SQL query to find the most popular product category, which is the one with the highest total quantity ordered across all orders.
SELECT TOP 1 P.ProductID,P.ProductName,SUM(OD.Quantity) AS NumberOfOrders
FROM Products P
JOIN OrderDetails OD ON OD.ProductID=P.ProductID
GROUP BY P.ProductID,P.ProductName
ORDER BY NumberOfOrders DESC;
-- an SQL query to find the customer who has spent the most money (highest total revenue) on electronic gadgets. List their name and total spending.
SELECT TOP 1 C.CustomerID,C.FirstName,C.LastName,SUM(O.TotalAmount*OD.Quantity) AS MoneySpent
ProductID ProductName NumberOfOrders
3 Wireless Earbuds 2
Query executed successfully.

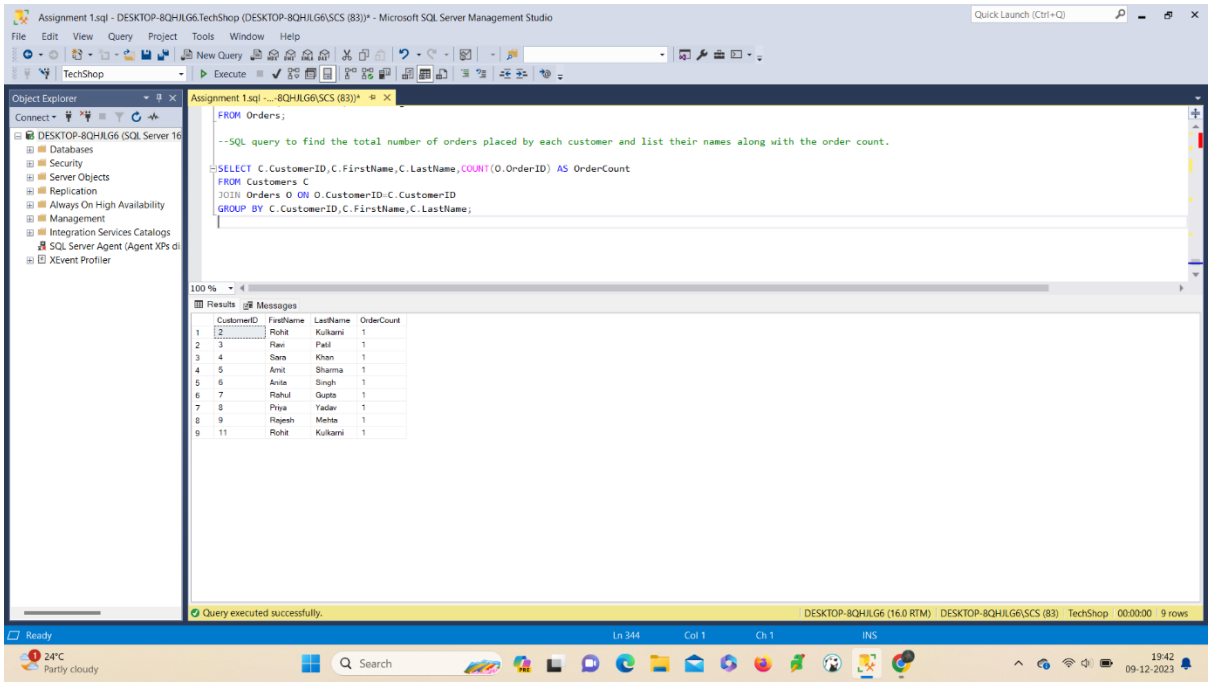
8.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
JOIN OrderDetails OD ON OD.ProductID=P.ProductID
GROUP BY P.ProductID,P.ProductName
ORDER BY NumberOfOrders DESC;
-- an SQL query to find the customer who has spent the most money (highest total revenue) on electronic gadgets. List their name and total spending.
SELECT TOP 1 C.CustomerID,C.FirstName,C.LastName,SUM(O.TotalAmount*OD.Quantity) AS MoneySpent
FROM Customers C
JOIN Orders O ON O.CustomerID=C.CustomerID
JOIN OrderDetails OD ON OD.OrderID=O.OrderID
GROUP BY C.CustomerID,C.FirstName,C.LastName;
CustomerID FirstName LastName MoneySpent
2 Rohit Kulkarni 4347
Query executed successfully.

9.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
JOIN OrderDetails OD ON OD.OrderID=O.OrderID
GROUP BY C.CustomerID,C.FirstName,C.LastName;
--SQL query to calculate the average order value (total revenue divided by the number of orders) for all customers.
SELECT AVG(TotalAmount) AS AverageOrderValue
FROM Orders;
--SQL query to find the total number of orders placed by each customer and list their names along with the order count.
SELECT C.CustomerID,C.FirstName,C.LastName,COUNT(O.OrderID) AS OrderCount
FROM Customers C
JOIN Orders O ON O.CustomerID=C.CustomerID
AverageOrderValue
517
Query executed successfully.

10.


Assignment 1.sql - DESKTOP-8QHJLG6.TechShop (DESKTOP-8QHJLG6\SCS (83))* - Microsoft SQL Server Management Studio
FROM Orders;
--SQL query to find the total number of orders placed by each customer and list their names along with the order count.
SELECT C.CustomerID,C.FirstName,C.LastName,COUNT(O.OrderID) AS OrderCount
FROM Customers C
JOIN Orders O ON O.CustomerID=C.CustomerID
GROUP BY C.CustomerID,C.FirstName,C.LastName;
CustomerID FirstName LastName OrderCount
2 Rohit Kulkarni 1
3 Ravi Patil 1
4 Sara Khan 1
5 Amit Sharma 1
6 Anita Singh 1
7 Rahul Gupta 1
8 Priya Yadav 1
9 Rajesh Mehta 1
11 Rohit Kulkarni 1
Query executed successfully.