Foundation Technical Training

Assignment No. 1 - SQL - Electronic Gadgets - Task 2

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

Tasks 2: Select, Where, Between, AND, LIKE:

1. Write an SQL query to retrieve the names and emails of all customers.
2. Write an SQL query to list all orders with their order dates and corresponding customer names.
3. Write an SQL query to insert a new customer record into the "Customers" table. Include customer information such as name, email, and address.
4. Write an SQL query to update the prices of all electronic gadgets in the "Products" table by increasing them by 10%.
5. Write an 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.
6. Write an SQL query to insert a new order into the "Orders" table. Include the customer ID, order date, and any other necessary information.
7. Write an 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.
8. Write 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.
9. Write an 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.
10. Write an SQL query to insert a new electronic gadget product into the "Products" table, including product name, category, price, and any other relevant details.
11. Write an 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.
12. Write an 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.

Program Code in SQL:

1.
```sql
-- Write an SQL query to retrieve the names and emails of all customers.

USE NiranjanDB;

SELECT FirstName, LastName, Email FROM Customers;
```

2.
```sql
-- Write an SQL query to list all orders with their order dates and corresponding customer names.

SELECT OrderID, FirstName, LastName, OrderDate FROM Orders, Customers WHERE Orders.CustomerID=Customers.CustomerID;
```

3. -- Write an SQL query to insert a new customer record into the "Customers" table.

-- Include customer information such as name, email, and address.

```
INSERT INTO Customers VALUES (11, 'Benedict', 'Cumberbatch', 'benedict@gmail.com', 9999999982, 'Chennai');

SELECT * FROM Customers;
```

4. -- Write an SQL query to update the prices of all electronic gadgets in the "Products" table by

-- increasing them by 10%.

```
UPDATE Products SET Price=Price+(Price*10/100);

SELECT * FROM Products;
```

5. -- Write an 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 @OrderID int = 10;


--before deletion

select * from Orders;

select * from OrderDetails;


--after deletion

delete from OrderDetails

where OrderID=@OrderID;


delete from Orders
```

```sql
where OrderID = @OrderID;


Select * from Orders;

select * from OrderDetails;


6. -- Write an 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 VALUES (10, 2, '2024-09-19', 1000);

SELECT * FROM Orders;


7. -- Write an 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.

DECLARE @CustomerID int = 7;

DECLARE @Email varchar(50) = 'clintbarton@gmail.com';


SELECT * FROM Customers WHERE CustomerID = @CustomerID;

UPDATE Customers SET Email = @Email WHERE CustomerID = @CustomerID;

SELECT * from Customers WHERE CustomerID = @CustomerID;


8. -- Write 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.


SELECT * FROM Orders;
```

```
-- The "TotalAmount" column has already been set according to the "Price" and
"Quantity" columns,

-- when the order was placed and the record for it was inserted.
```

9.
```
-- Write an 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.


SELECT * FROM Orders;

SELECT * FROM OrderDetails;

BEGIN TRANSACTION;

DECLARE @CustomerID INT = 6;

DELETE FROM OrderDetails WHERE OrderID IN (SELECT OrderID FROM Orders WHERE
CustomerID = @CustomerID);

DELETE FROM Orders WHERE CustomerID = @CustomerID;

COMMIT TRANSACTION;

SELECT * FROM Orders;

SELECT * FROM OrderDetails;
```

10.
```
-- Write an SQL query to insert a new electronic gadget product into the
"Products" table,

-- including product name, category, price, and any other relevant details.


BEGIN TRANSACTION;

SELECT * FROM Products;
```

```
INSERT INTO Products (ProductID, ProductName, Description, Price) VALUES (11,
'Speaker', 'Bluetooth Speaker', 2000);

SELECT * FROM Products;

COMMIT TRANSACTION;
```

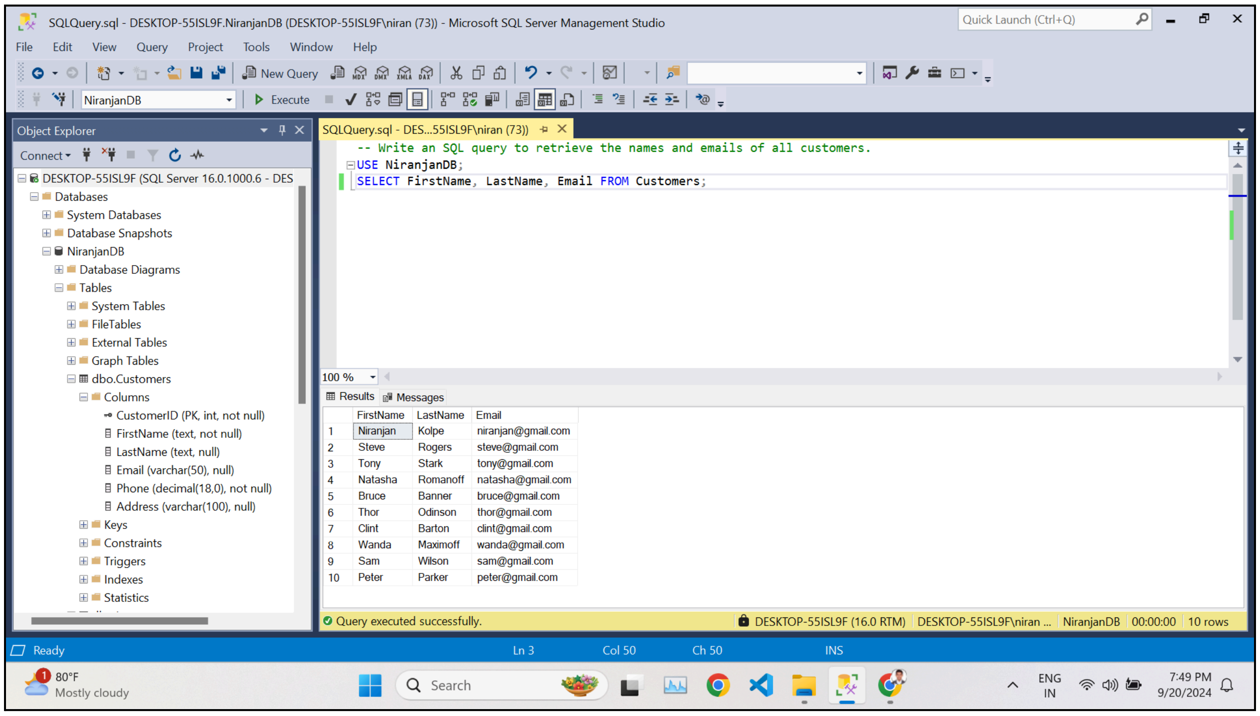
```
11. -- Write an SQL query to update the status of a specific order in the
"Orders" table (e.g., from

-- "Pending" to "Shipped"). Allow users to input the order ID and the new status.
```

```
UPDATE Orders SET Status='Pending';

SELECT * FROM Orders;

BEGIN TRANSACTION;

DECLARE @OrderID INT = 8;

UPDATE Orders SET Status='Shipped' WHERE OrderID=@OrderID;

SELECT * FROM Orders;

COMMIT TRANSACTION;
```

```
12. -- Write an 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.
```

```
BEGIN TRANSACTION;

UPDATE Customers SET NumberOfOrders = (SELECT COUNT(*) FROM Orders WHERE
Orders.CustomerID = Customers.CustomerID);

SELECT * FROM Customers;

COMMIT TRANSACTION;
```

Output 1:

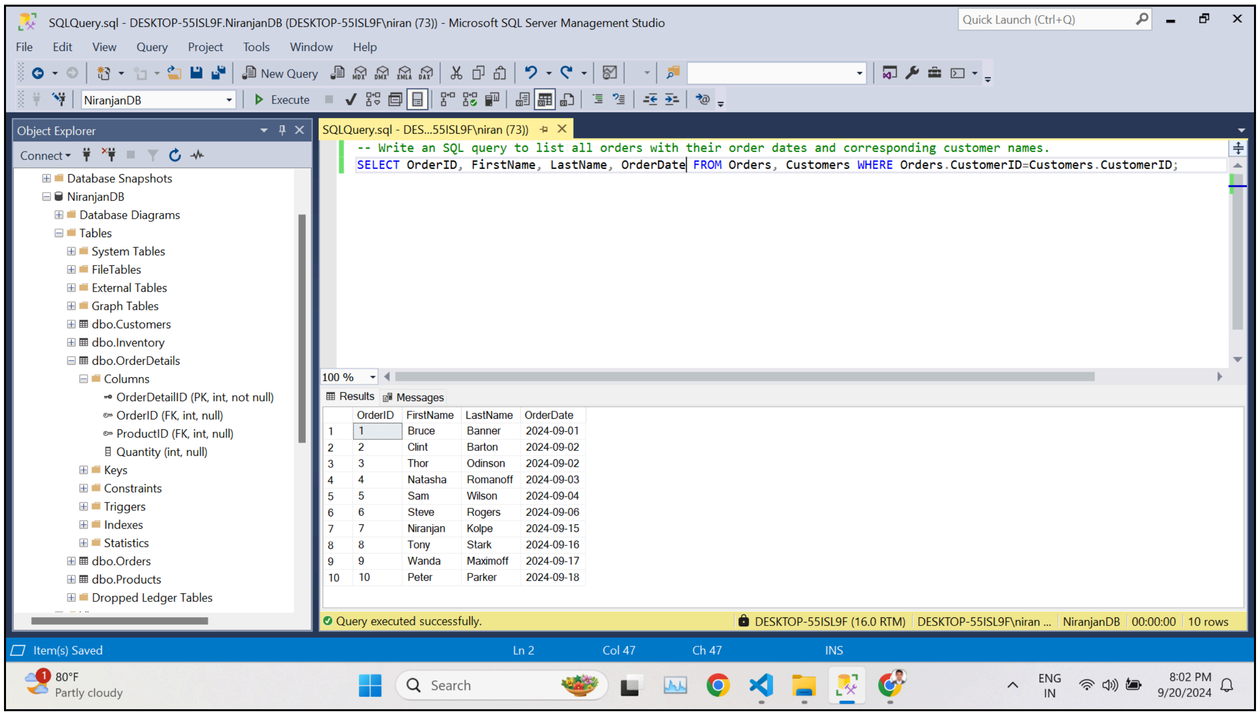

SQLQuery.sql - DESKTOP-55ISL9F.NiranjanDB (DESKTOP-55ISL9F\niran (73)) - Microsoft SQL Server Management Studio
Object Explorer
DESKTOP-55ISL9F (SQL Server 16.0.1000.6 - DES
Databases
NiranjanDB
dbo.Customers
CustomerID (PK, int, not null)
FirstName (text, not null)
LastName (text, null)
Email (varchar(50), null)
Phone (decimal(18,0), not null)
Address (varchar(100), null)
Results
Messages
Ready
Ln 3
Col 50
Ch 50
INS

```sql
-- Write an SQL query to retrieve the names and emails of all customers.
USE NiranjanDB;
SELECT FirstName, LastName, Email FROM Customers;
```

	FirstName	LastName	Email
1	Niranjan	Kolpe	niranjan@gmail.com
2	Steve	Rogers	steve@gmail.com
3	Tony	Stark	tony@gmail.com
4	Natasha	Romanoff	natasha@gmail.com
5	Bruce	Banner	bruce@gmail.com
6	Thor	Odinson	thor@gmail.com
7	Clint	Barton	clint@gmail.com
8	Wanda	Maximoff	wanda@gmail.com
9	Sam	Wilson	sam@gmail.com
10	Peter	Parker	peter@gmail.com

Query executed successfully. DESKTOP-55ISL9F (16.0 RTM) | DESKTOP-55ISL9F\niran ... | NiranjanDB | 00:00:00 | 10 rows

Output 2:

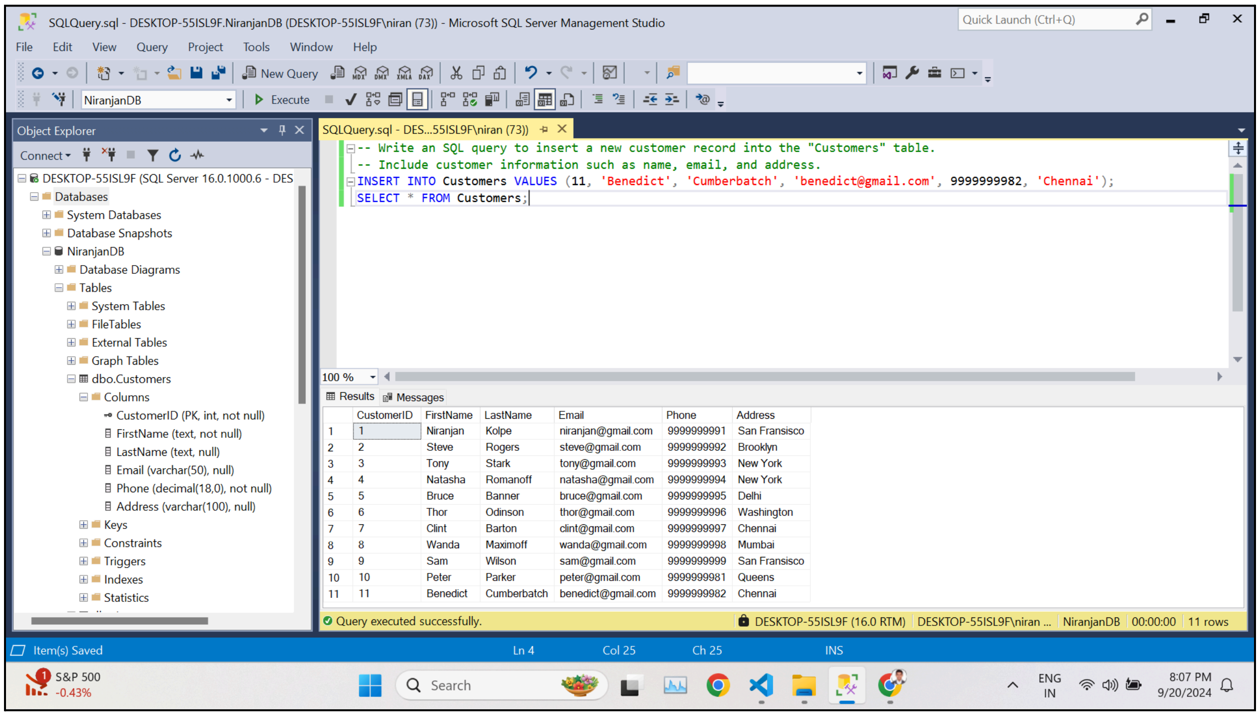


```sql
-- Write an SQL query to list all orders with their order dates and corresponding customer names.
SELECT OrderID, FirstName, LastName, OrderDate FROM Orders, Customers WHERE Orders.CustomerID=Customers.CustomerID;
```

	OrderID	FirstName	LastName	OrderDate
1	1	Bruce	Banner	2024-09-01
2	2	Clint	Barton	2024-09-02
3	3	Thor	Odinson	2024-09-02
4	4	Natasha	Romanoff	2024-09-03
5	5	Sam	Wilson	2024-09-04
6	6	Steve	Rogers	2024-09-06
7	7	Niranjan	Kolpe	2024-09-15
8	8	Tony	Stark	2024-09-16
9	9	Wanda	Maximoff	2024-09-17
10	10	Peter	Parker	2024-09-18

Query executed successfully. DESKTOP-55ISL9F (16.0 RTM) | DESKTOP-55ISL9F\niran ... | NiranjanDB | 00:00:00 | 10 rows

Output 3:

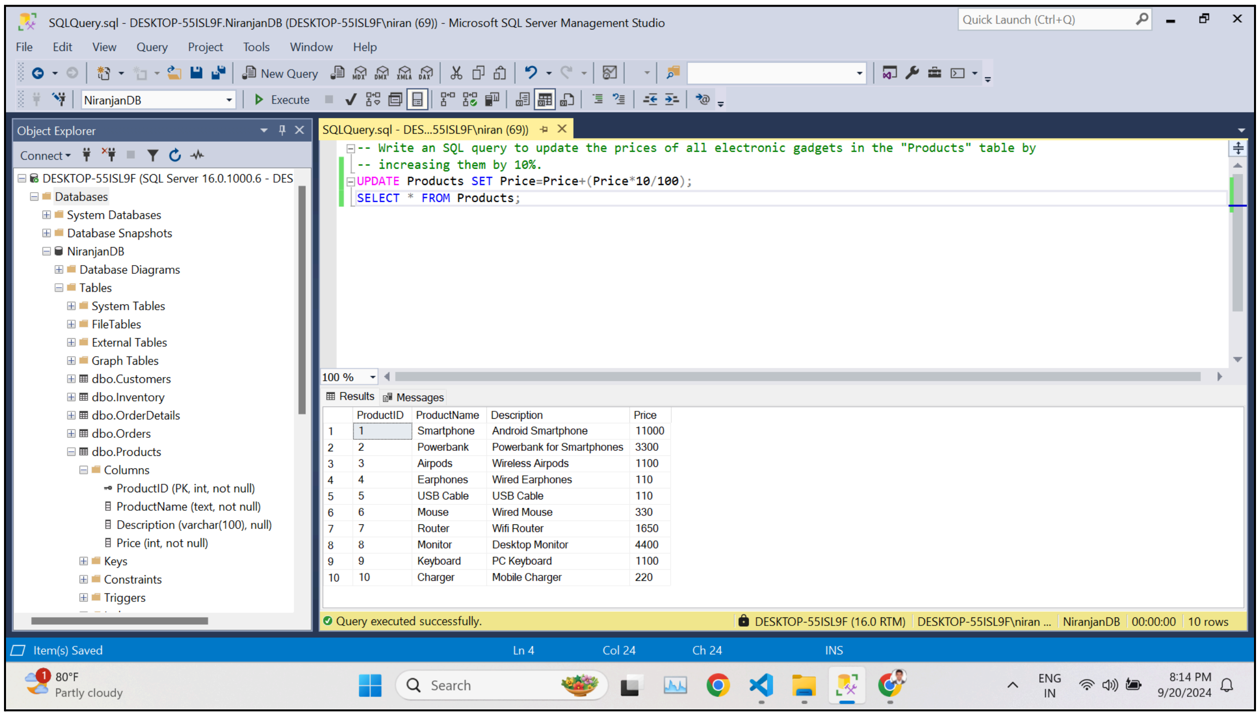


```
-- Write an SQL query to insert a new customer record into the "Customers" table.
-- Include customer information such as name, email, and address.
INSERT INTO Customers VALUES (11, 'Benedict', 'Cumberbatch', 'benedict@gmail.com', 9999999982, 'Chennai');
SELECT * FROM Customers;
```

	CustomerID	FirstName	LastName	Email	Phone	Address
1	1	Niranjan	Kolpe	niranjan@gmail.com	9999999991	San Fransisco
2	2	Steve	Rogers	steve@gmail.com	9999999992	Brooklyn
3	3	Tony	Stark	tony@gmail.com	9999999993	New York
4	4	Natasha	Romanoff	natasha@gmail.com	9999999994	New York
5	5	Bruce	Banner	bruce@gmail.com	9999999995	Delhi
6	6	Thor	Odinson	thor@gmail.com	9999999996	Washington
7	7	Clint	Barton	clint@gmail.com	9999999997	Chennai
8	8	Wanda	Maximoff	wanda@gmail.com	9999999998	Mumbai
9	9	Sam	Wilson	sam@gmail.com	9999999999	San Fransisco
10	10	Peter	Parker	peter@gmail.com	9999999981	Queens
11	11	Benedict	Cumberbatch	benedict@gmail.com	9999999982	Chennai

```
Query executed successfully.    DESKTOP-55ISL9F (16.0 RTM)   DESKTOP-55ISL9F\niran ...   NiranjanDB   00:00:00   11 rows
```

Output 4:

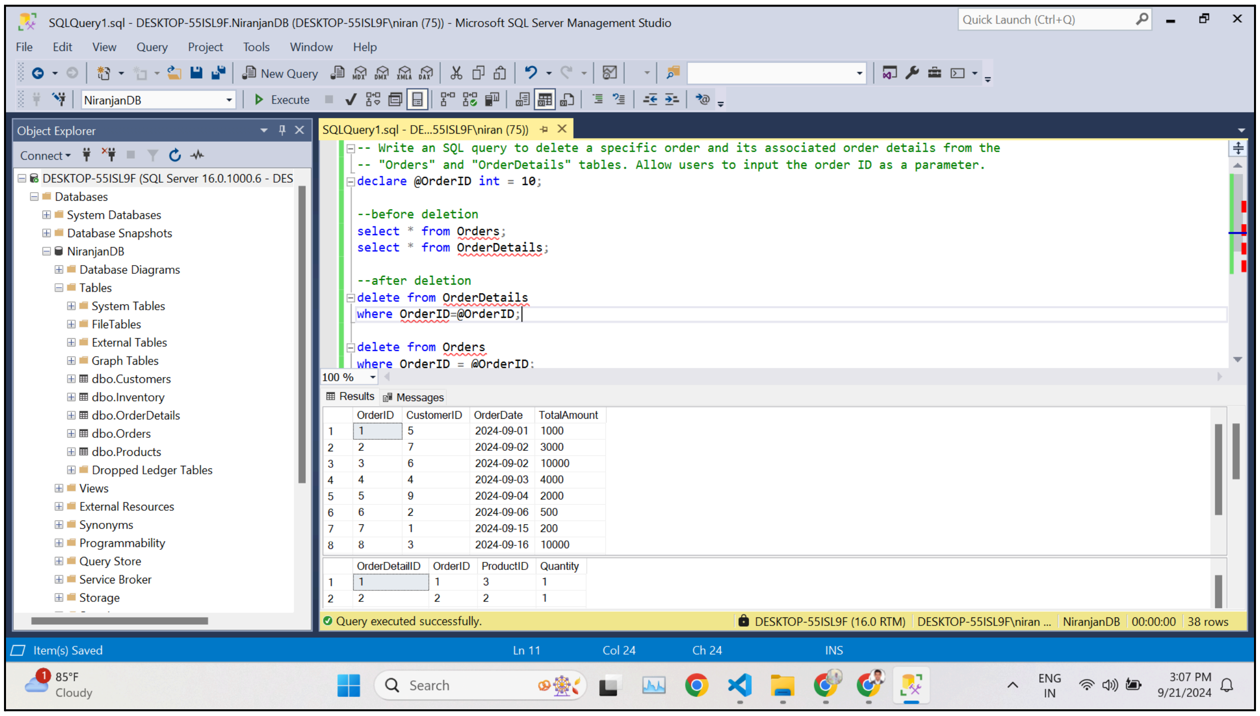


```
-- Write an SQL query to update the prices of all electronic gadgets in the "Products" table by
-- increasing them by 10%.
UPDATE Products SET Price=Price+(Price*10/100);
SELECT * FROM Products;
```

	ProductID	ProductName	Description	Price
1	1	Smartphone	Android Smartphone	11000
2	2	Powerbank	Powerbank for Smartphones	3300
3	3	Airpods	Wireless Airpods	1100
4	4	Earphones	Wired Earphones	110
5	5	USB Cable	USB Cable	110
6	6	Mouse	Wired Mouse	330
7	7	Router	Wifi Router	1650
8	8	Monitor	Desktop Monitor	4400
9	9	Keyboard	PC Keyboard	1100
10	10	Charger	Mobile Charger	220

```
Query executed successfully.    DESKTOP-55ISL9F (16.0 RTM)   DESKTOP-55ISL9F\niran ...   NiranjanDB   00:00:00   10 rows
```

Output 5:



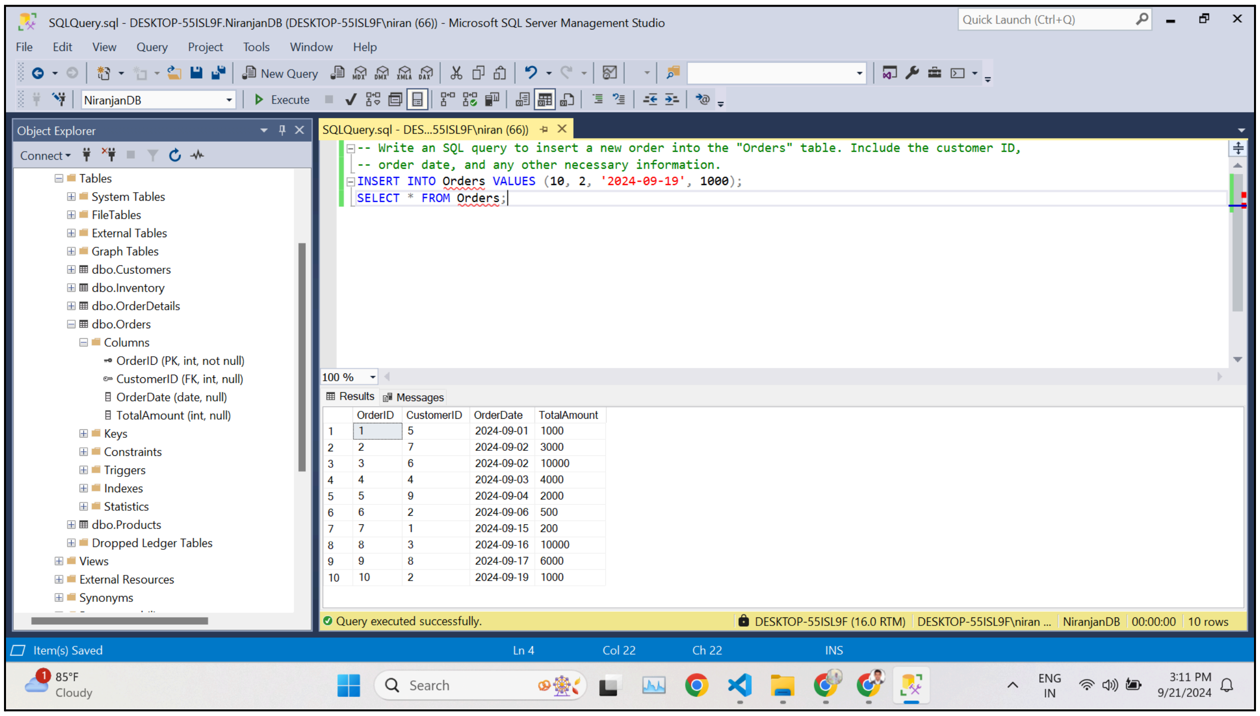
```
-- Write an SQL query to delete a specific order and its associated order details from the
-- "Orders" and "OrderDetails" tables. Allow users to input the order ID as a parameter.
declare @OrderID int = 10;

--before deletion
select * from Orders;
select * from OrderDetails;

--after deletion
delete from OrderDetails
where OrderID=@OrderID;

delete from Orders
where OrderID = @OrderID;
```

	OrderID	CustomerID	OrderDate	TotalAmount
1	1	5	2024-09-01	1000
2	2	7	2024-09-02	3000
3	3	6	2024-09-02	10000
4	4	4	2024-09-03	4000
5	5	9	2024-09-04	2000
6	6	2	2024-09-06	500
7	7	1	2024-09-15	200
8	8	3	2024-09-16	10000

	OrderDetailID	OrderID	ProductID	Quantity
1	1	1	3	1
2	2	2	2	1

```
Query executed successfully.    DESKTOP-55ISL9F (16.0 RTM) | DESKTOP-55ISL9F\niran ... | NiranjanDB | 00:00:00 | 38 rows
```

Output 6:

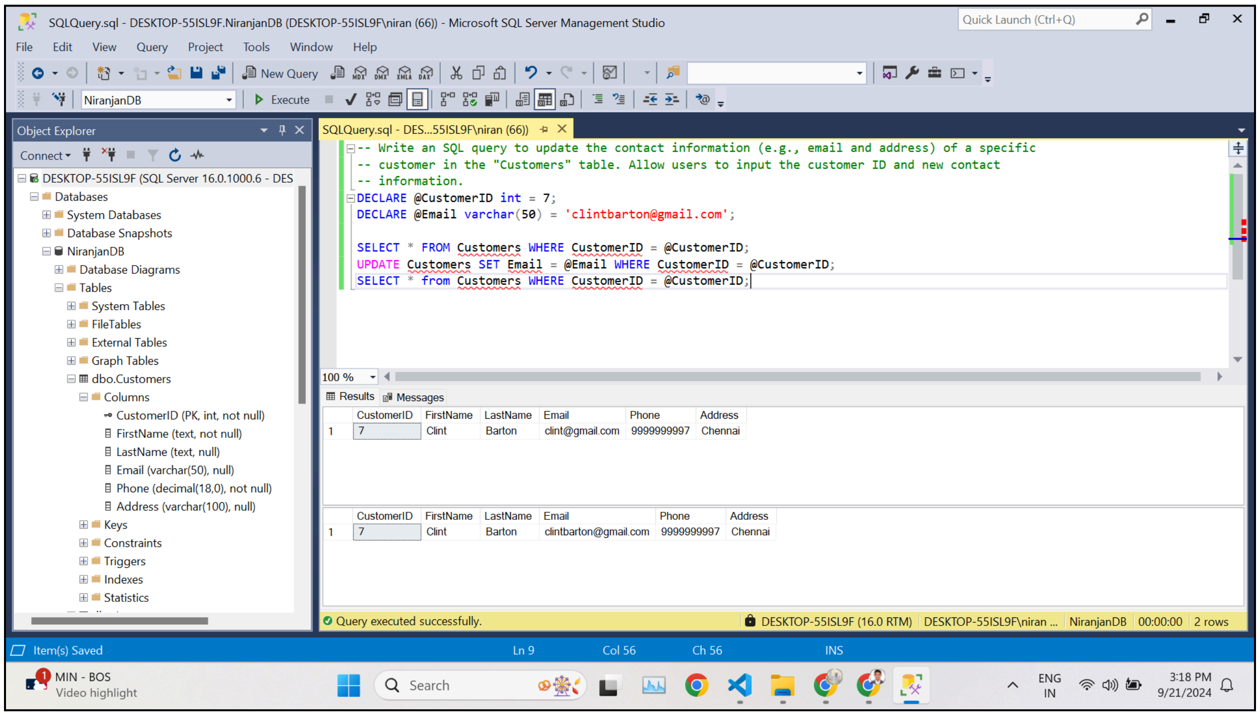


```
-- Write an SQL query to insert a new order into the "Orders" table. Include the customer ID,
-- order date, and any other necessary information.
INSERT INTO Orders VALUES (10, 2, '2024-09-19', 1000);
SELECT * FROM Orders;
```

	OrderID	CustomerID	OrderDate	TotalAmount
1	1	5	2024-09-01	1000
2	2	7	2024-09-02	3000
3	3	6	2024-09-02	10000
4	4	4	2024-09-03	4000
5	5	9	2024-09-04	2000
6	6	2	2024-09-06	500
7	7	1	2024-09-15	200
8	8	3	2024-09-16	10000
9	9	8	2024-09-17	6000
10	10	2	2024-09-19	1000

```
Query executed successfully.    DESKTOP-55ISL9F (16.0 RTM) | DESKTOP-55ISL9F\niran ... | NiranjanDB | 00:00:00 | 10 rows
```

Output 7:



SQLQuery.sql - DESKTOP-55ISL9F.NiranjanDB (DESKTOP-55ISL9F\niran (66)) - Microsoft SQL Server Management Studio

```
DESKTOP-55ISL9F (SQL Server 16.0.1000.6 - DES
  Databases
    System Databases
    Database Snapshots
    NiranjanDB
      Database Diagrams
      Tables
        System Tables
        FileTables
        External Tables
        Graph Tables
        dbo.Customers
          Columns
            CustomerID (PK, int, not null)
            FirstName (text, not null)
            LastName (text, null)
            Email (varchar(50), null)
            Phone (decimal(18,0), not null)
            Address (varchar(100), null)
          Keys
          Constraints
          Triggers
          Indexes
          Statistics
```

```sql
-- Write an 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.
DECLARE @CustomerID int = 7;
DECLARE @Email varchar(50) = 'clintbarton@gmail.com';

SELECT * FROM Customers WHERE CustomerID = @CustomerID;
UPDATE Customers SET Email = @Email WHERE CustomerID = @CustomerID;
SELECT * from Customers WHERE CustomerID = @CustomerID;
```

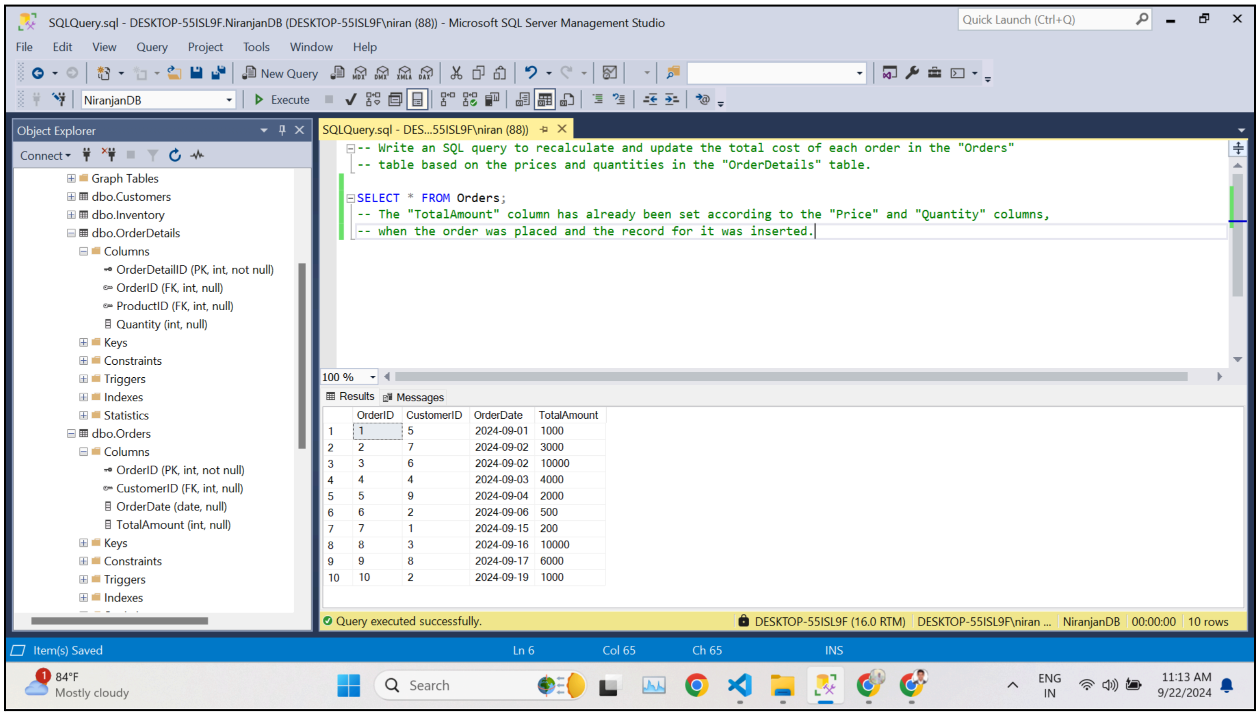
	CustomerID	FirstName	LastName	Email	Phone	Address
1	7	Clint	Barton	clint@gmail.com	9999999997	Chennai

	CustomerID	FirstName	LastName	Email	Phone	Address
1	7	Clint	Barton	clintbarton@gmail.com	9999999997	Chennai

Query executed successfully. | DESKTOP-55ISL9F (16.0 RTM) | DESKTOP-55ISL9F\niran ... | NiranjanDB | 00:00:00 | 2 rows

Item(s) Saved Ln 9 Col 56 Ch 56 INS

Output 8:



SQLQuery.sql - DESKTOP-55ISL9F.NiranjanDB (DESKTOP-55ISL9F\niran (88)) - Microsoft SQL Server Management Studio

```
        Graph Tables
        dbo.Customers
        dbo.Inventory
        dbo.OrderDetails
          Columns
            OrderDetailID (PK, int, not null)
            OrderID (FK, int, null)
            ProductID (FK, int, null)
            Quantity (int, null)
          Keys
          Constraints
          Triggers
          Indexes
          Statistics
        dbo.Orders
          Columns
            OrderID (PK, int, not null)
            CustomerID (FK, int, null)
            OrderDate (date, null)
            TotalAmount (int, null)
          Keys
          Constraints
          Triggers
          Indexes
```

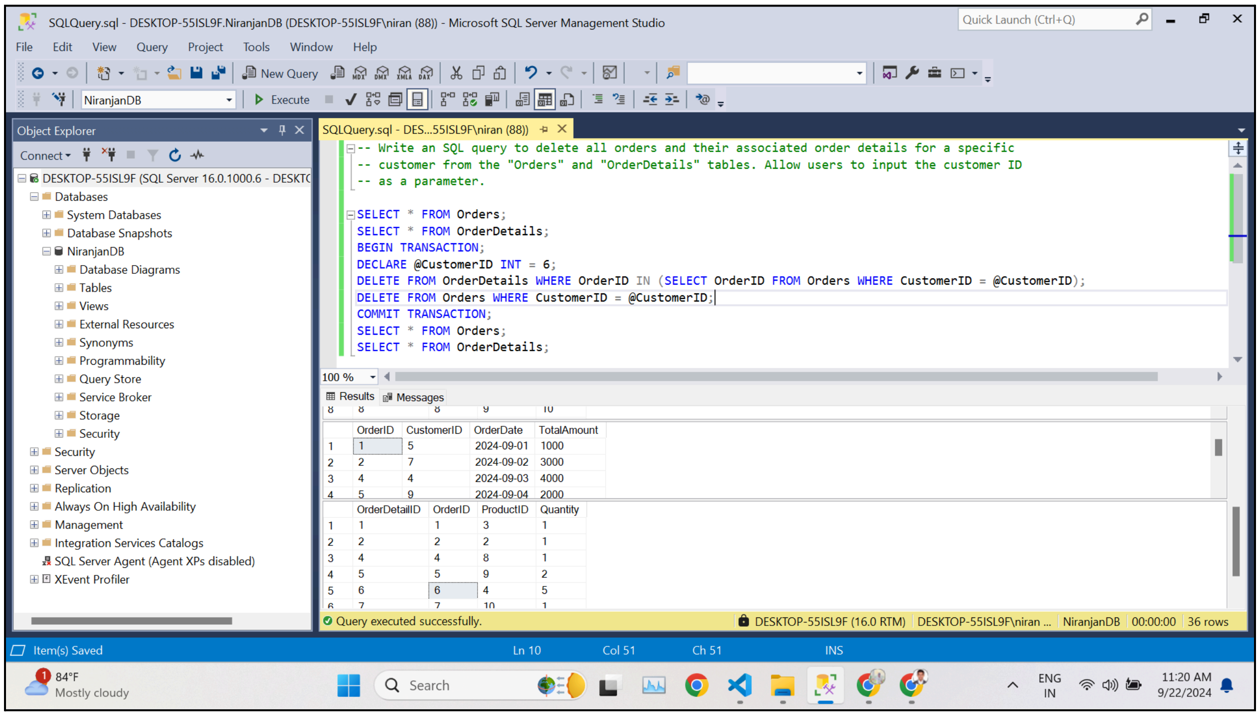
```sql
-- Write an SQL query to recalculate and update the total cost of each order in the "Orders"
-- table based on the prices and quantities in the "OrderDetails" table.

SELECT * FROM Orders;
-- The "TotalAmount" column has already been set according to the "Price" and "Quantity" columns,
-- when the order was placed and the record for it was inserted.
```

	OrderID	CustomerID	OrderDate	TotalAmount
1	1	5	2024-09-01	1000
2	2	7	2024-09-02	3000
3	3	6	2024-09-02	10000
4	4	4	2024-09-03	4000
5	5	9	2024-09-04	2000
6	6	2	2024-09-06	500
7	7	1	2024-09-15	200
8	8	3	2024-09-16	10000
9	9	8	2024-09-17	6000
10	10	2	2024-09-19	1000

Query executed successfully. | DESKTOP-55ISL9F (16.0 RTM) | DESKTOP-55ISL9F\niran ... | NiranjanDB | 00:00:00 | 10 rows

Item(s) Saved Ln 6 Col 65 Ch 65 INS

Output 9:



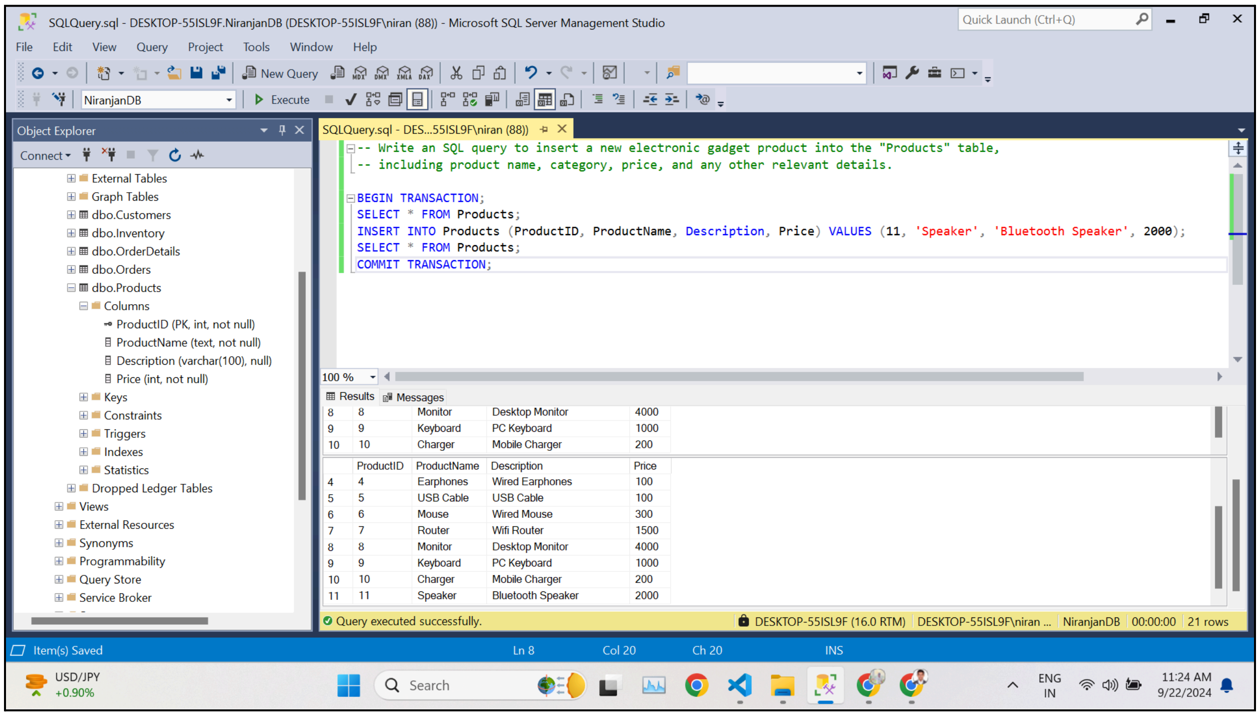
SQLQuery.sql - DESKTOP-55ISL9F.NiranjanDB (DESKTOP-55ISL9F\niran (88)) - Microsoft SQL Server Management Studio

```sql
-- Write an 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.

SELECT * FROM Orders;
SELECT * FROM OrderDetails;
BEGIN TRANSACTION;
DECLARE @CustomerID INT = 6;
DELETE FROM OrderDetails WHERE OrderID IN (SELECT OrderID FROM Orders WHERE CustomerID = @CustomerID);
DELETE FROM Orders WHERE CustomerID = @CustomerID;
COMMIT TRANSACTION;
SELECT * FROM Orders;
SELECT * FROM OrderDetails;
```

```
8    8    8    9    10

     OrderID  CustomerID  OrderDate   TotalAmount
1    1        5           2024-09-01  1000
2    2        7           2024-09-02  3000
3    4        4           2024-09-03  4000
4    5        9           2024-09-04  2000

     OrderDetailID  OrderID  ProductID  Quantity
1    1              1        3          1
2    2              2        2          1
3    4              4        8          1
4    5              5        9          2
5    6              6        4          5
6    7              7        10         1
```

Query executed successfully. | DESKTOP-55ISL9F (16.0 RTM) | DESKTOP-55ISL9F\niran ... | NiranjanDB | 00:00:00 | 36 rows

Output 10:



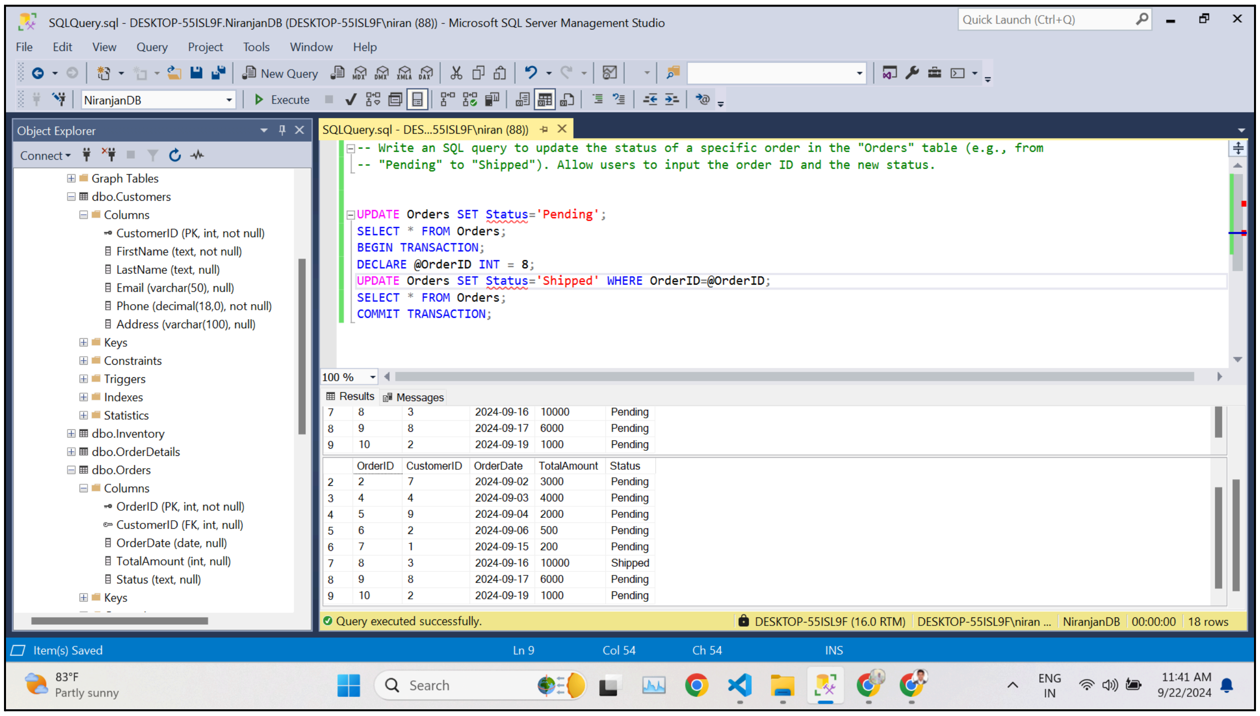
SQLQuery.sql - DESKTOP-55ISL9F.NiranjanDB (DESKTOP-55ISL9F\niran (88)) - Microsoft SQL Server Management Studio

```sql
-- Write an SQL query to insert a new electronic gadget product into the "Products" table,
-- including product name, category, price, and any other relevant details.

BEGIN TRANSACTION;
SELECT * FROM Products;
INSERT INTO Products (ProductID, ProductName, Description, Price) VALUES (11, 'Speaker', 'Bluetooth Speaker', 2000);
SELECT * FROM Products;
COMMIT TRANSACTION;
```

```
8    8    Monitor    Desktop Monitor    4000
9    9    Keyboard   PC Keyboard        1000
10   10   Charger    Mobile Charger     200

     ProductID  ProductName  Description        Price
4    4          Earphones    Wired Earphones    100
5    5          USB Cable    USB Cable          100
6    6          Mouse        Wired Mouse        300
7    7          Router       Wifi Router        1500
8    8          Monitor      Desktop Monitor    4000
9    9          Keyboard     PC Keyboard        1000
10   10         Charger      Mobile Charger     200
11   11         Speaker      Bluetooth Speaker  2000
```

Query executed successfully. | DESKTOP-55ISL9F (16.0 RTM) | DESKTOP-55ISL9F\niran ... | NiranjanDB | 00:00:00 | 21 rows

Output 11:



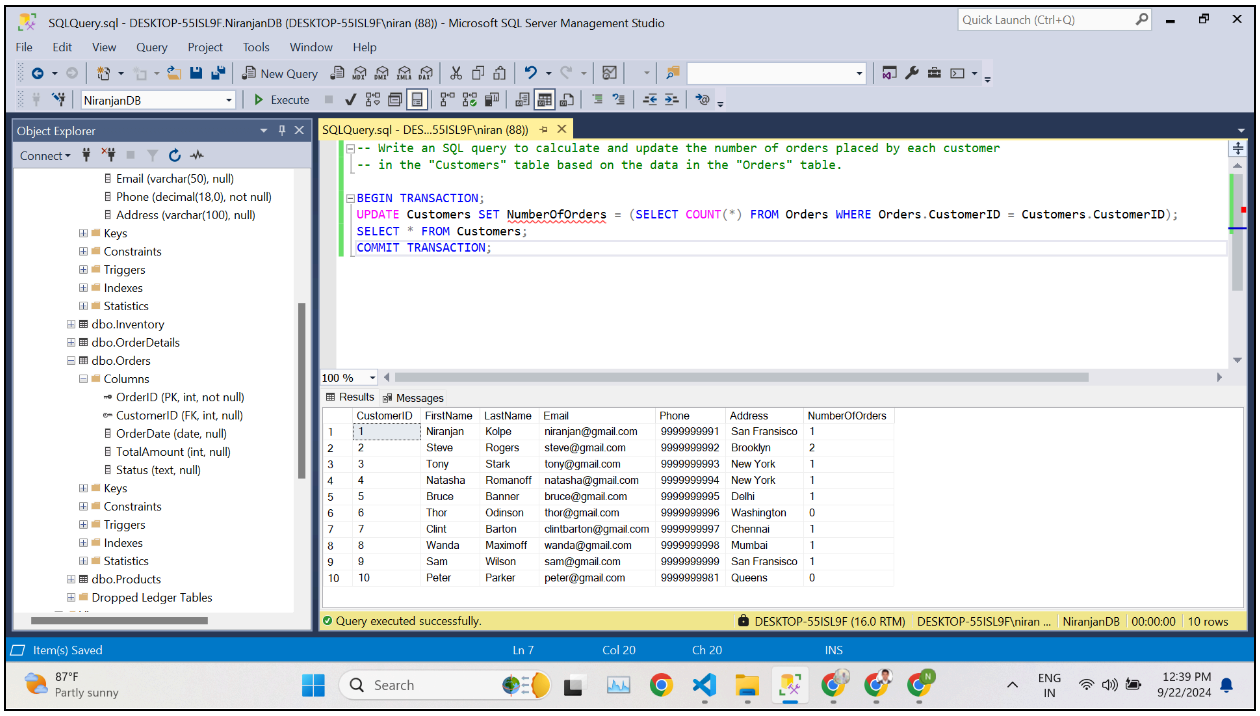
```sql
-- Write an SQL query to update the status of a specific order in the "Orders" table (e.g., from
-- "Pending" to "Shipped"). Allow users to input the order ID and the new status.

UPDATE Orders SET Status='Pending';
SELECT * FROM Orders;
BEGIN TRANSACTION;
DECLARE @OrderID INT = 8;
UPDATE Orders SET Status='Shipped' WHERE OrderID=@OrderID;
SELECT * FROM Orders;
COMMIT TRANSACTION;
```

7	8	3	2024-09-16	10000	Pending
8	9	8	2024-09-17	6000	Pending
9	10	2	2024-09-19	1000	Pending

	OrderID	CustomerID	OrderDate	TotalAmount	Status
2	2	7	2024-09-02	3000	Pending
3	4	4	2024-09-03	4000	Pending
4	5	9	2024-09-04	2000	Pending
5	6	2	2024-09-06	500	Pending
6	7	1	2024-09-15	200	Pending
7	8	3	2024-09-16	10000	Shipped
8	9	8	2024-09-17	6000	Pending
9	10	2	2024-09-19	1000	Pending

Query executed successfully. DESKTOP-55ISL9F (16.0 RTM) DESKTOP-55ISL9F\niran ... NiranjanDB 00:00:00 18 rows

Output 12:



```sql
-- Write an 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.

BEGIN TRANSACTION;
UPDATE Customers SET NumberOfOrders = (SELECT COUNT(*) FROM Orders WHERE Orders.CustomerID = Customers.CustomerID);
SELECT * FROM Customers;
COMMIT TRANSACTION;
```

	CustomerID	FirstName	LastName	Email	Phone	Address	NumberOfOrders
1	1	Niranjan	Kolpe	niranjan@gmail.com	9999999991	San Fransisco	1
2	2	Steve	Rogers	steve@gmail.com	9999999992	Brooklyn	2
3	3	Tony	Stark	tony@gmail.com	9999999993	New York	1
4	4	Natasha	Romanoff	natasha@gmail.com	9999999994	New York	1
5	5	Bruce	Banner	bruce@gmail.com	9999999995	Delhi	1
6	6	Thor	Odinson	thor@gmail.com	9999999996	Washington	0
7	7	Clint	Barton	clintbarton@gmail.com	9999999997	Chennai	1
8	8	Wanda	Maximoff	wanda@gmail.com	9999999998	Mumbai	1
9	9	Sam	Wilson	sam@gmail.com	9999999999	San Fransisco	1
10	10	Peter	Parker	peter@gmail.com	9999999981	Queens	0

Query executed successfully. DESKTOP-55ISL9F (16.0 RTM) DESKTOP-55ISL9F\niran ... NiranjanDB 00:00:00 10 rows