Hexaware Technologies LTD :- Foundation Training Program

External Assignment MySQL

Trainer:- Mr. Veerababu

Date:- 30/11/2023

TEAM Members:

1. Suraj Kumar:- 7-10(Aggregate functions and subqueries), (and creation of tables(All DDL Commands))
2. ANOOP KUMAR: 1-7(SQL queries)
3. Abhishek Negi:- 2-9(Joins)
4. Riju Antony Josdas: 10(Joins), 1-6(Aggregate functions and subqueries)
5. Gourav Singh: 8-12(SQL queries), 1-2(Joins)

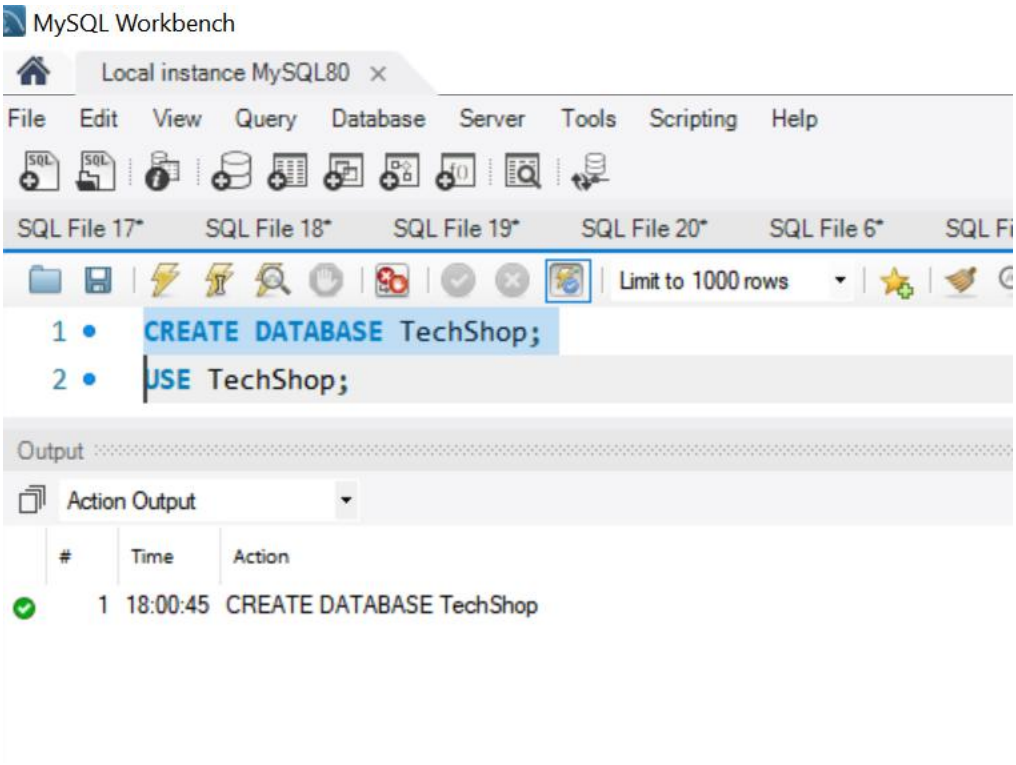
RDBMS Assignment: TechShop, an electronic gadgets shop

Scenario: You are working as a database administrator for a fictional company named "TechShop," which sells electronic gadgets. TechShop maintains data related to their products, customers, and orders. Your task is to design and implement a database for TechShop based on the following requirements:

Tasks:

1. Database Design (Normalization):

1. Create the database named "TechShop"



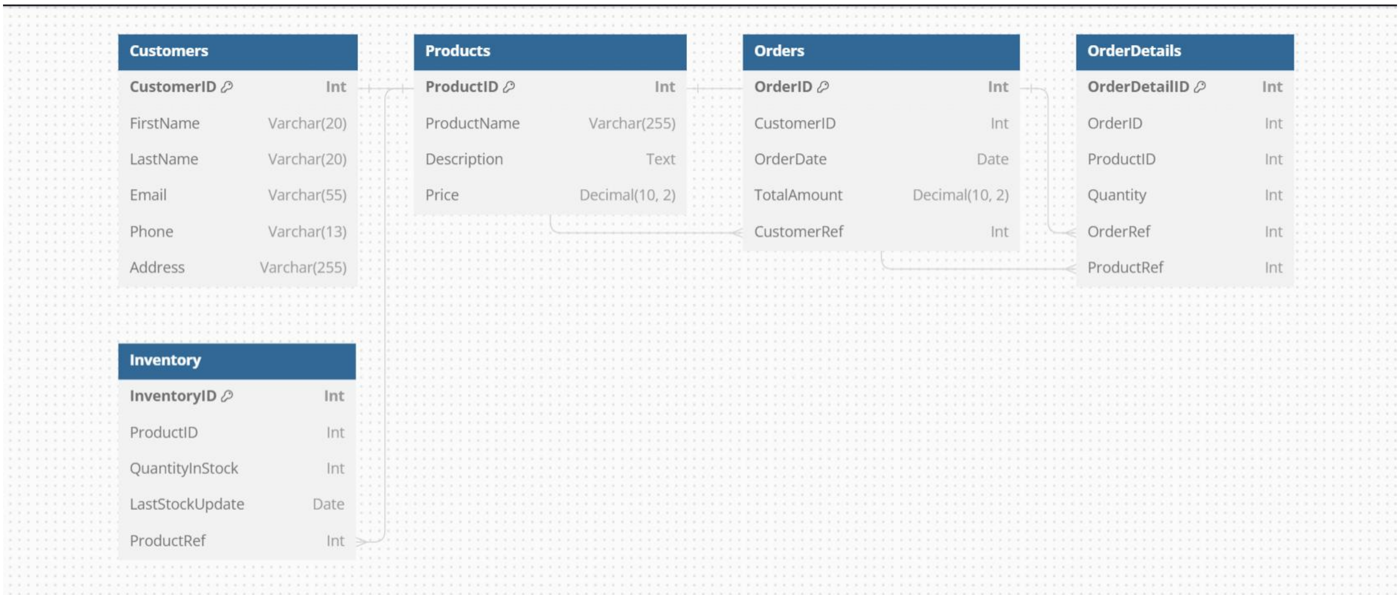
```
CREATE DATABASE TechShop;
USE TechShop;
```

```
#   Time       Action
1   18:00:45   CREATE DATABASE TechShop
```

2. Define the schema for the Customers, Products, Orders, OrderDetails and Inventory tables
based on the provided schema.

3. Perform the first three normal forms (1NF, 2NF, 3NF) analysis on the above tables.

4. Create an ERD (Entity Relationship Diagram) for the database.



Customers	
CustomerID	Int
FirstName	Varchar(20)
LastName	Varchar(20)
Email	Varchar(55)
Phone	Varchar(13)
Address	Varchar(255)

Products	
ProductID	Int
ProductName	Varchar(255)
Description	Text
Price	Decimal(10, 2)

Orders	
OrderID	Int
CustomerID	Int
OrderDate	Date
TotalAmount	Decimal(10, 2)
CustomerRef	Int

OrderDetails	
OrderDetailID	Int
OrderID	Int
ProductID	Int
Quantity	Int
OrderRef	Int
ProductRef	Int

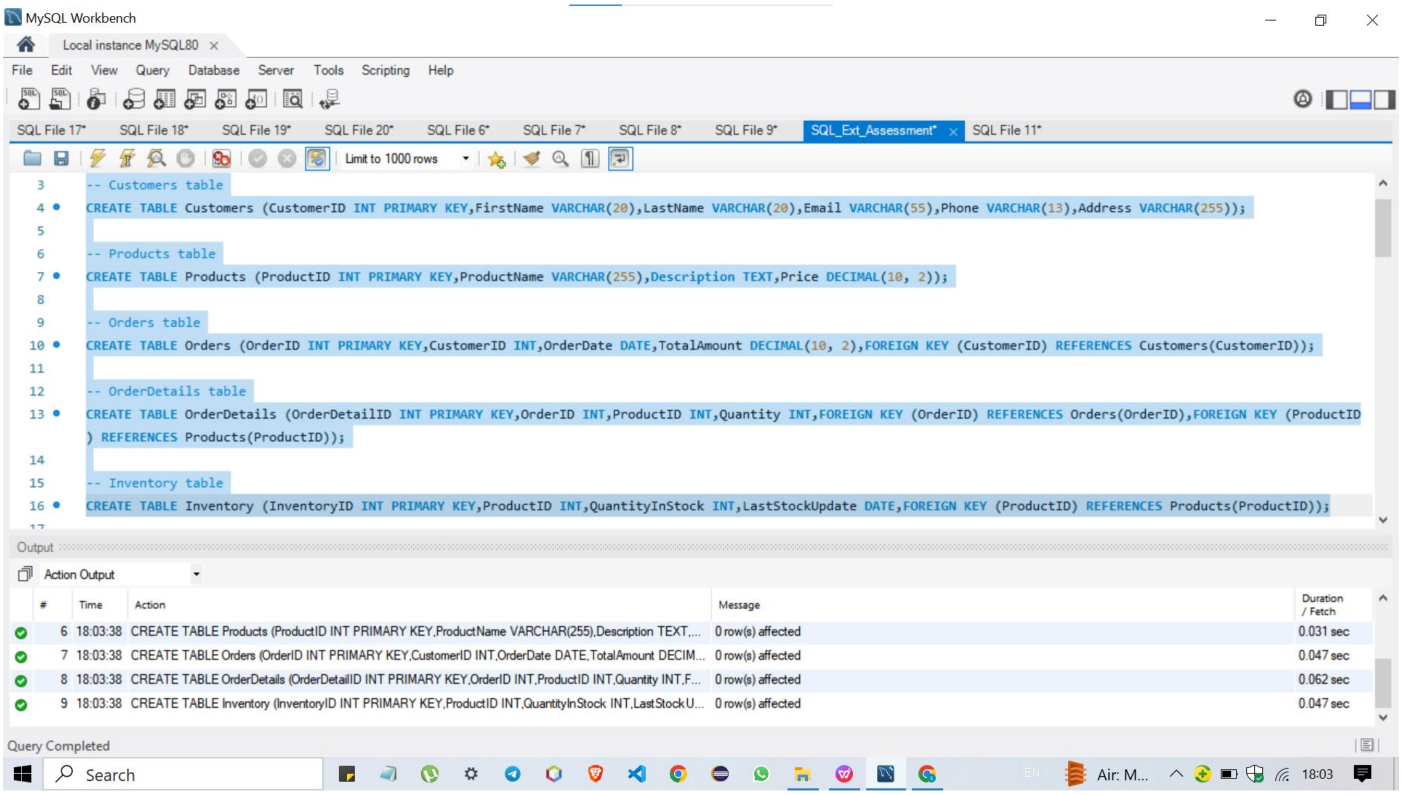
Inventory	
InventoryID	Int
ProductID	Int
QuantityInStock	Int
LastStockUpdate	Date
ProductRef	Int

5. Create appropriate Primary Key and Foreign Key constraints for referential integrity.

2. *Data Definition Language (DDL):*

1. Write SQL scripts to create the mentioned tables with appropriate data types, constraints, and
relationships.

- Customers
- Products
- Orders
- OrderDetails
- Inventory



MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 17* SQL File 18* SQL File 19* SQL File 20* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL_Ext_Assessment* SQL File 11*
Limit to 1000 rows
Output
Action Output

```sql
-- Customers table
CREATE TABLE Customers (CustomerID INT PRIMARY KEY,FirstName VARCHAR(20),LastName VARCHAR(20),Email VARCHAR(55),Phone VARCHAR(13),Address VARCHAR(255));

-- Products table
CREATE TABLE Products (ProductID INT PRIMARY KEY,ProductName VARCHAR(255),Description TEXT,Price DECIMAL(10, 2));

-- Orders table
CREATE TABLE Orders (OrderID INT PRIMARY KEY,CustomerID INT,OrderDate DATE,TotalAmount DECIMAL(10, 2),FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID));

-- OrderDetails table
CREATE TABLE OrderDetails (OrderDetailID INT PRIMARY KEY,OrderID INT,ProductID INT,Quantity INT,FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),FOREIGN KEY (ProductID
) REFERENCES Products(ProductID));

-- Inventory table
CREATE TABLE Inventory (InventoryID INT PRIMARY KEY,ProductID INT,QuantityInStock INT,LastStockUpdate DATE,FOREIGN KEY (ProductID) REFERENCES Products(ProductID));
```

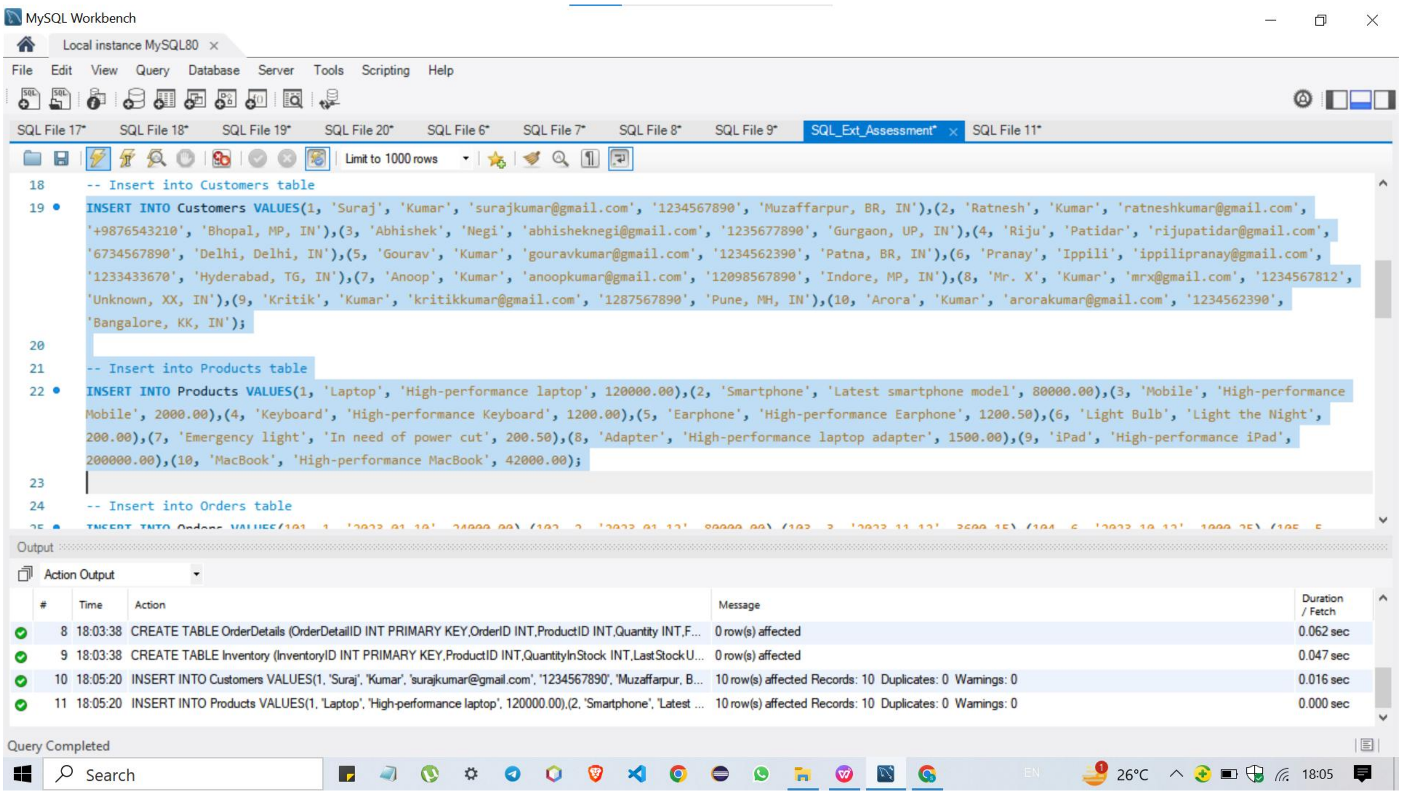
	#	Time	Action	Message	Duration / Fetch
✓	6	18:03:38	CREATE TABLE Products (ProductID INT PRIMARY KEY,ProductName VARCHAR(255),Description TEXT,...	0 row(s) affected	0.031 sec
✓	7	18:03:38	CREATE TABLE Orders (OrderID INT PRIMARY KEY,CustomerID INT,OrderDate DATE,TotalAmount DECIM...	0 row(s) affected	0.047 sec
✓	8	18:03:38	CREATE TABLE OrderDetails (OrderDetailID INT PRIMARY KEY,OrderID INT,ProductID INT,Quantity INT,F...	0 row(s) affected	0.062 sec
✓	9	18:03:38	CREATE TABLE Inventory (InventoryID INT PRIMARY KEY,ProductID INT,QuantityInStock INT,LastStockU...	0 row(s) affected	0.047 sec

Query Completed

3. Data Manipulation Language (DML):

a. Insert at least 10 sample records into each of the following tables.

- Customers
- Products
- Orders
- OrderDetails
- Inventory


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 17* SQL File 18* SQL File 19* SQL File 20* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL_Ext_Assessment* SQL File 11*
Limit to 1000 rows

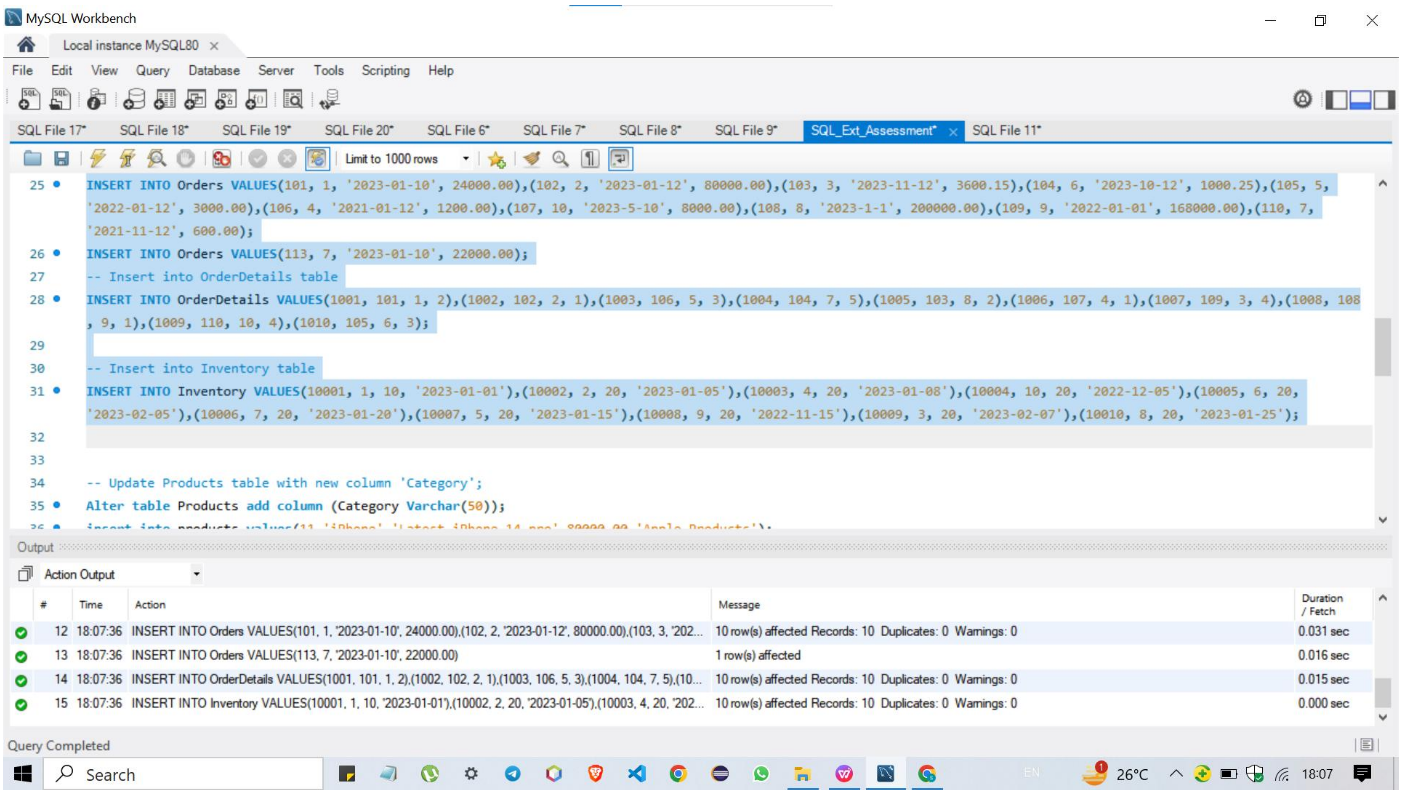
```sql
-- Insert into Customers table
INSERT INTO Customers VALUES(1, 'Suraj', 'Kumar', 'surajkumar@gmail.com', '1234567890', 'Muzaffarpur, BR, IN'),(2, 'Ratnesh', 'Kumar', 'ratneshkumar@gmail.com', '+9876543210', 'Bhopal, MP, IN'),(3, 'Abhishek', 'Negi', 'abhisheknegi@gmail.com', '1235677890', 'Gurgaon, UP, IN'),(4, 'Riju', 'Patidar', 'rijupatidar@gmail.com', '6734567890', 'Delhi, Delhi, IN'),(5, 'Gourav', 'Kumar', 'gouravkumar@gmail.com', '1234562390', 'Patna, BR, IN'),(6, 'Pranay', 'Ippili', 'ippilipranay@gmail.com', '1233433670', 'Hyderabad, TG, IN'),(7, 'Anoop', 'Kumar', 'anoopkumar@gmail.com', '12098567890', 'Indore, MP, IN'),(8, 'Mr. X', 'Kumar', 'mrx@gmail.com', '1234567812', 'Unknown, XX, IN'),(9, 'Kritik', 'Kumar', 'kritikkumar@gmail.com', '1287567890', 'Pune, MH, IN'),(10, 'Arora', 'Kumar', 'arorakumar@gmail.com', '1234562390', 'Bangalore, KK, IN');

-- Insert into Products table
INSERT INTO Products VALUES(1, 'Laptop', 'High-performance laptop', 120000.00),(2, 'Smartphone', 'Latest smartphone model', 80000.00),(3, 'Mobile', 'High-performance Mobile', 2000.00),(4, 'Keyboard', 'High-performance Keyboard', 1200.00),(5, 'Earphone', 'High-performance Earphone', 1200.50),(6, 'Light Bulb', 'Light the Night', 200.00),(7, 'Emergency light', 'In need of power cut', 200.50),(8, 'Adapter', 'High-performance laptop adapter', 1500.00),(9, 'iPad', 'High-performance iPad', 200000.00),(10, 'MacBook', 'High-performance MacBook', 42000.00);

-- Insert into Orders table
INSERT INTO Orders VALUES(101, 1, '2023-01-10', 24000.00),(102, 2, '2023-01-12', 80000.00),(103, 3, '2023-11-12', 3600.15),(104, 6, '2023-10-12', 1000.25),(105, 5
```

Output

Action Output

```
#   Time      Action                                                                                    Message                                                      Duration / Fetch
8   18:03:38  CREATE TABLE OrderDetails (OrderDetailID INT PRIMARY KEY,OrderID INT,ProductID INT,Quantity INT,F...  0 row(s) affected                                            0.062 sec
9   18:03:38  CREATE TABLE Inventory (InventoryID INT PRIMARY KEY,ProductID INT,QuantityInStock INT,LastStockU...  0 row(s) affected                                            0.047 sec
10  18:05:20  INSERT INTO Customers VALUES(1, 'Suraj', 'Kumar', 'surajkumar@gmail.com', '1234567890', 'Muzaffarpur, B...  10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0  0.016 sec
11  18:05:20  INSERT INTO Products VALUES(1, 'Laptop', 'High-performance laptop', 120000.00),(2, 'Smartphone', 'Latest ...  10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0  0.000 sec
```

Query Completed



```
INSERT INTO Orders VALUES(101, 1, '2023-01-10', 24000.00),(102, 2, '2023-01-12', 80000.00),(103, 3, '2023-11-12', 3600.15),(104, 6, '2023-10-12', 1000.25),(105, 5,
'2022-01-12', 3000.00),(106, 4, '2021-01-12', 1200.00),(107, 10, '2023-5-10', 8000.00),(108, 8, '2023-1-1', 200000.00),(109, 9, '2022-01-01', 168000.00),(110, 7,
'2021-11-12', 600.00);
INSERT INTO Orders VALUES(113, 7, '2023-01-10', 22000.00);
-- Insert into OrderDetails table
INSERT INTO OrderDetails VALUES(1001, 101, 1, 2),(1002, 102, 2, 1),(1003, 106, 5, 3),(1004, 104, 7, 5),(1005, 103, 8, 2),(1006, 107, 4, 1),(1007, 109, 3, 4),(1008, 108
, 9, 1),(1009, 110, 10, 4),(1010, 105, 6, 3);

-- Insert into Inventory table
INSERT INTO Inventory VALUES(10001, 1, 10, '2023-01-01'),(10002, 2, 20, '2023-01-05'),(10003, 4, 20, '2023-01-08'),(10004, 10, 20, '2022-12-05'),(10005, 6, 20,
'2023-02-05'),(10006, 7, 20, '2023-01-20'),(10007, 5, 20, '2023-01-15'),(10008, 9, 20, '2022-11-15'),(10009, 3, 20, '2023-02-07'),(10010, 8, 20, '2023-01-25');



-- Update Products table with new column 'Category';
Alter table Products add column (Category Varchar(50));
insert into products values(11,'iPhone','Latest iPhone 14 pro',80000.00,'Apple Products');
```

```
#   Time      Action                                                                                         Message                                                      Duration / Fetch
12  18:07:36  INSERT INTO Orders VALUES(101, 1, '2023-01-10', 24000.00),(102, 2, '2023-01-12', 80000.00),(103, 3, '202...   10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0   0.031 sec
13  18:07:36  INSERT INTO Orders VALUES(113, 7, '2023-01-10', 22000.00)                                           1 row(s) affected                                           0.016 sec
14  18:07:36  INSERT INTO OrderDetails VALUES(1001, 101, 1, 2),(1002, 102, 2, 1),(1003, 106, 5, 3),(1004, 104, 7, 5),(10...   10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0   0.015 sec
15  18:07:36  INSERT INTO Inventory VALUES(10001, 1, 10, '2023-01-01'),(10002, 2, 20, '2023-01-05'),(10003, 4, 20, '202...   10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0   0.000 sec
```

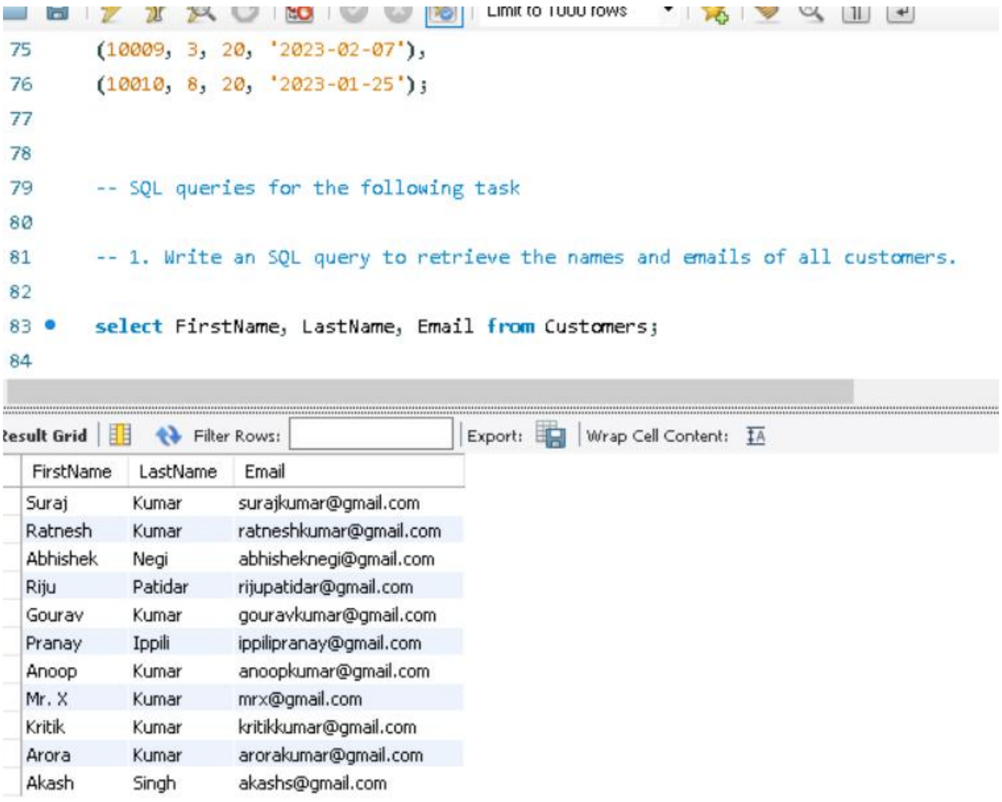
Query Completed

b. Write SQL queries for the following tasks:

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

Query

select FirstName, LastName, Email from Customers;



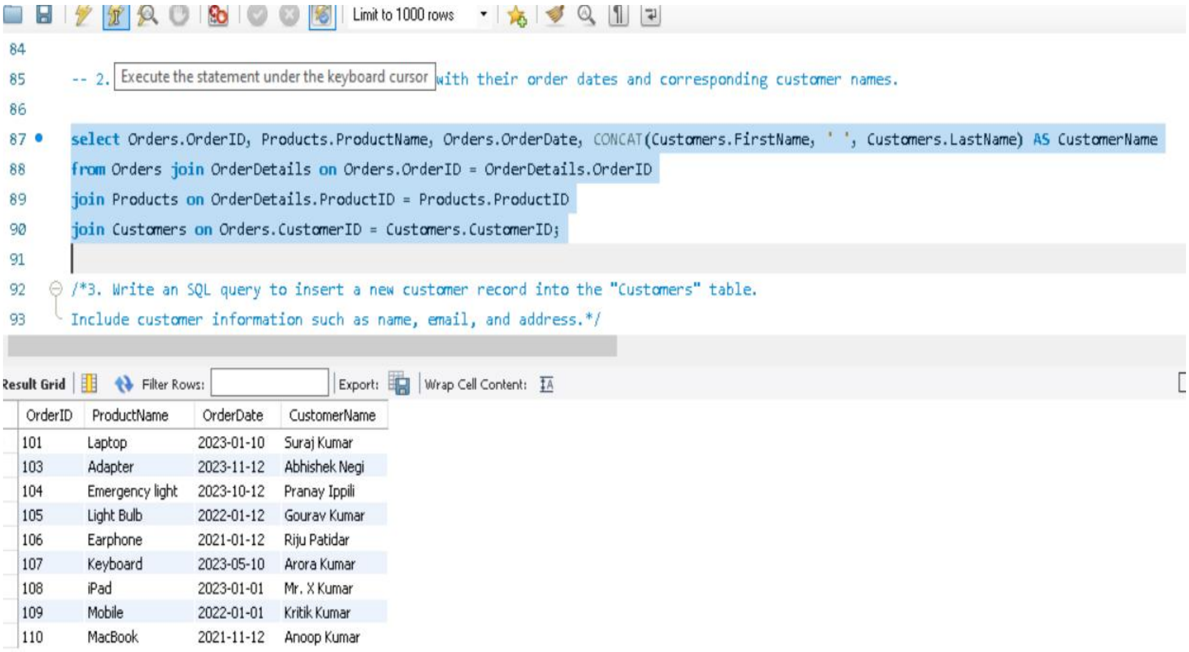
```
      (10009, 3, 20, '2023-02-07'),
      (10010, 8, 20, '2023-01-25');


-- SQL queries for the following task

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

select FirstName, LastName, Email from Customers;
```

FirstName	LastName	Email
Suraj	Kumar	surajkumar@gmail.com
Ratnesh	Kumar	ratneshkumar@gmail.com
Abhishek	Negi	abhisheknegi@gmail.com
Riju	Patidar	rijupatidar@gmail.com
Gourav	Kumar	gouravkumar@gmail.com
Pranay	Ippili	ippilipranay@gmail.com
Anoop	Kumar	anoopkumar@gmail.com
Mr. X	Kumar	mrx@gmail.com
Kritik	Kumar	kritikkumar@gmail.com
Arora	Kumar	arorakumar@gmail.com
Akash	Singh	akashs@gmail.com

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

Query:

```
select Orders.OrderID, Products.ProductName, Orders.OrderDate, CONCAT(Customers.FirstName, ' ', Customers.LastName) AS CustomerName
from Orders join OrderDetails on Orders.OrderID = OrderDetails.OrderID
join Products on OrderDetails.ProductID = Products.ProductID
join Customers on Orders.CustomerID = Customers.CustomerID;
```



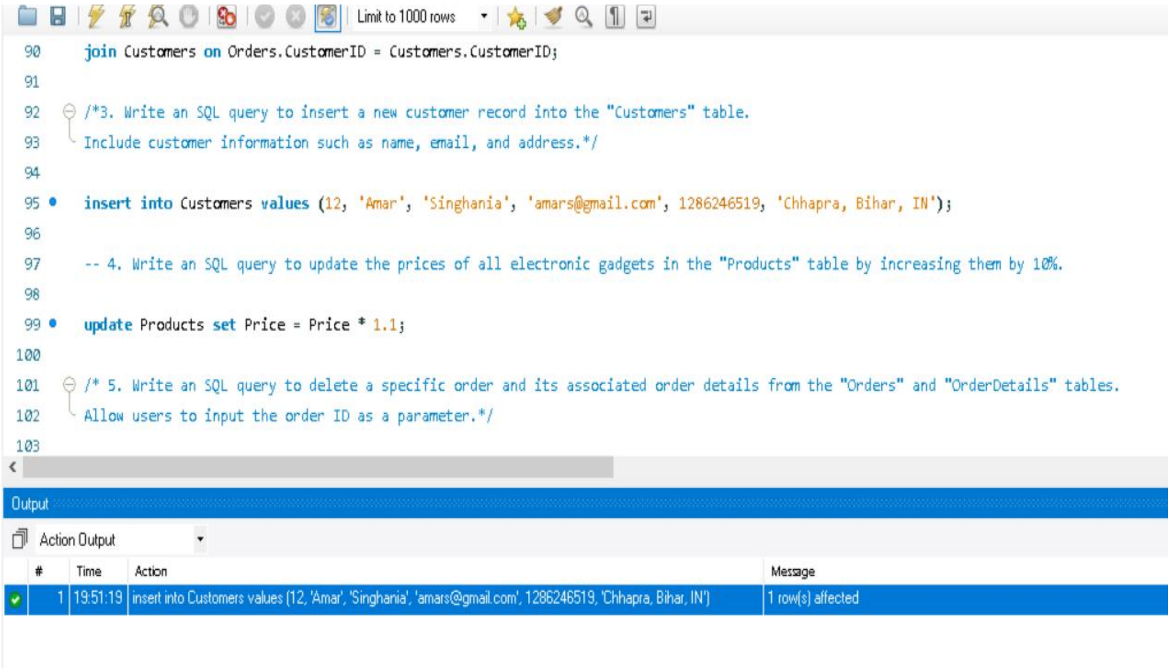
```
-- 2. with their order dates and corresponding customer names.

select Orders.OrderID, Products.ProductName, Orders.OrderDate, CONCAT(Customers.FirstName, ' ', Customers.LastName) AS CustomerName
from Orders join OrderDetails on Orders.OrderID = OrderDetails.OrderID
join Products on OrderDetails.ProductID = Products.ProductID
join Customers on 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.*/
```

OrderID	ProductName	OrderDate	CustomerName
101	Laptop	2023-01-10	Suraj Kumar
103	Adapter	2023-11-12	Abhishek Negi
104	Emergency light	2023-10-12	Pranay Ippili
105	Light Bulb	2022-01-12	Gourav Kumar
106	Earphone	2021-01-12	Riju Patidar
107	Keyboard	2023-05-10	Arora Kumar
108	iPad	2023-01-01	Mr. X Kumar
109	Mobile	2022-01-01	Kritik Kumar
110	MacBook	2021-11-12	Anoop Kumar

3. Write an SQL query to insert a new customer record into the "Customers" table. Include customer information such as name, email, and address.

Query:

insert into Customers values (11, 'Akash', 'Singh', 'akashs@gmail.com', 1235246519, 'Banaras, UP, IN');



```
join Customers on 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 (12, 'Amar', 'Singhania', 'amars@gmail.com', 1286246519, 'Chhapra, Bihar, IN');

-- 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 * 1.1;

/* 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.*/

```

Output

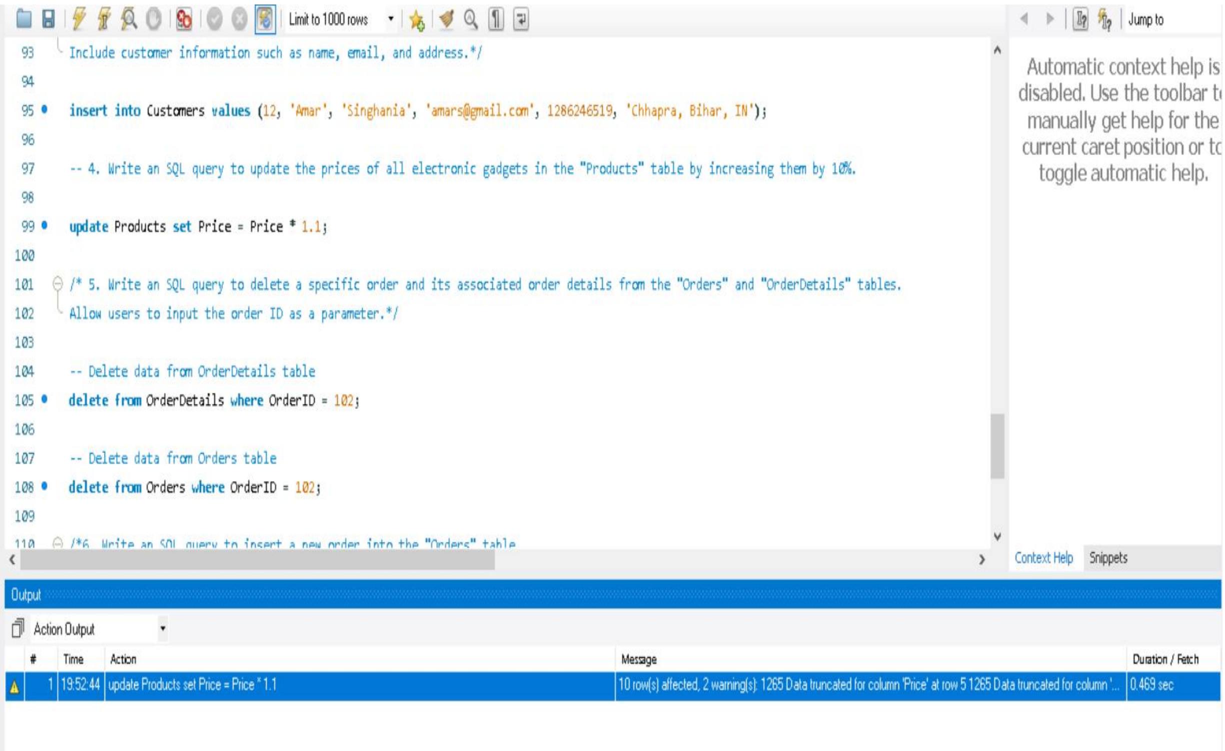
Action Output

	#	Time	Action	Message
✓	1	19:51:19	insert into Customers values (12, 'Amar', 'Singhania', 'amars@gmail.com', 1286246519, 'Chhapra, Bihar, IN')	1 row(s) affected

4. Write an SQL query to update the prices of all electronic gadgets in the "Products" table by increasing them by 10%.

Query:

update Products set Price = Price * 1.1;



```sql
Include customer information such as name, email, and address.*/

insert into Customers values (12, 'Amar', 'Singhania', 'amars@gmail.com', 1286246519, 'Chhapra, Bihar, IN');

-- 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 * 1.1;

/* 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.*/

-- Delete data from OrderDetails table
delete from OrderDetails where OrderID = 102;

-- Delete data from Orders table
delete from Orders where OrderID = 102;

/*6. Write an SQL query to insert a new order into the "Orders" table
```

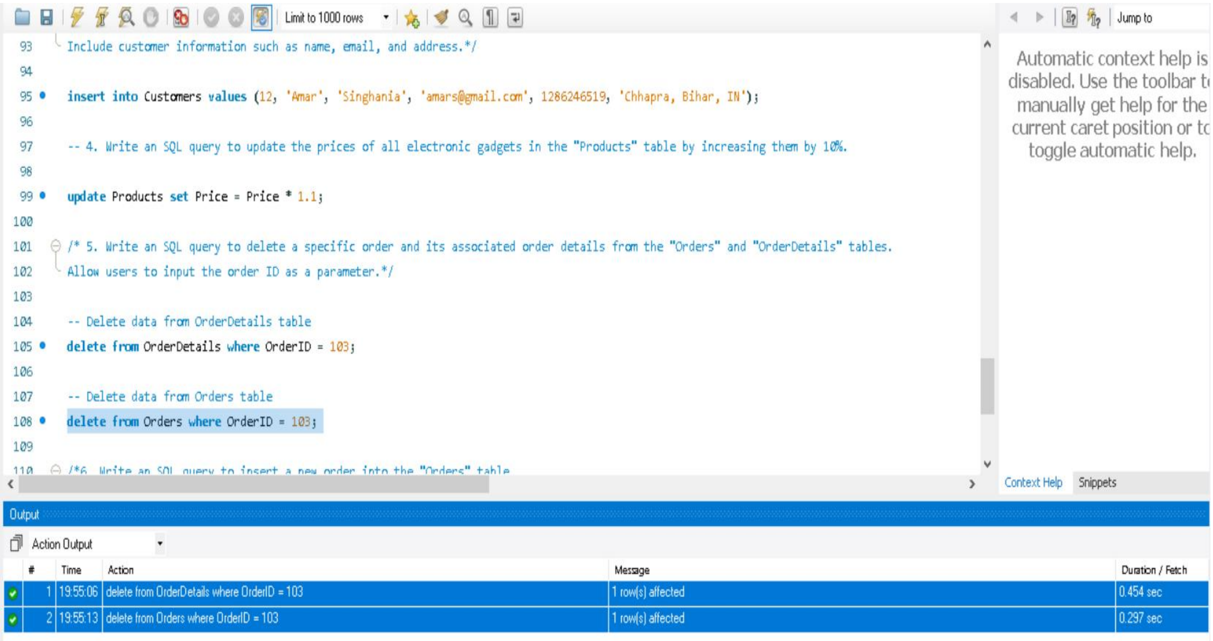
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or tc toggle automatic help.

	#	Time	Action	Message	Duration / Fetch
⚠	1	19:52:44	update Products set Price = Price * 1.1	10 row(s) affected, 2 warning(s): 1265 Data truncated for column 'Price' at row 5 1265 Data truncated for column '...	0.469 sec

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.

Query:

delete from OrderDetails where OrderID = 102;
delete from Orders where OrderID = 102;



```sql
Include customer information such as name, email, and address.*/

insert into Customers values (12, 'Amar', 'Singhania', 'amars@gmail.com', 1286246519, 'Chhapra, Bihar, IN');

-- 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 * 1.1;

/* 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.*/

-- Delete data from OrderDetails table
delete from OrderDetails where OrderID = 103;

-- Delete data from Orders table
delete from Orders where OrderID = 103;

/*6. Write an SQL query to insert a new order into the "Orders" table
```

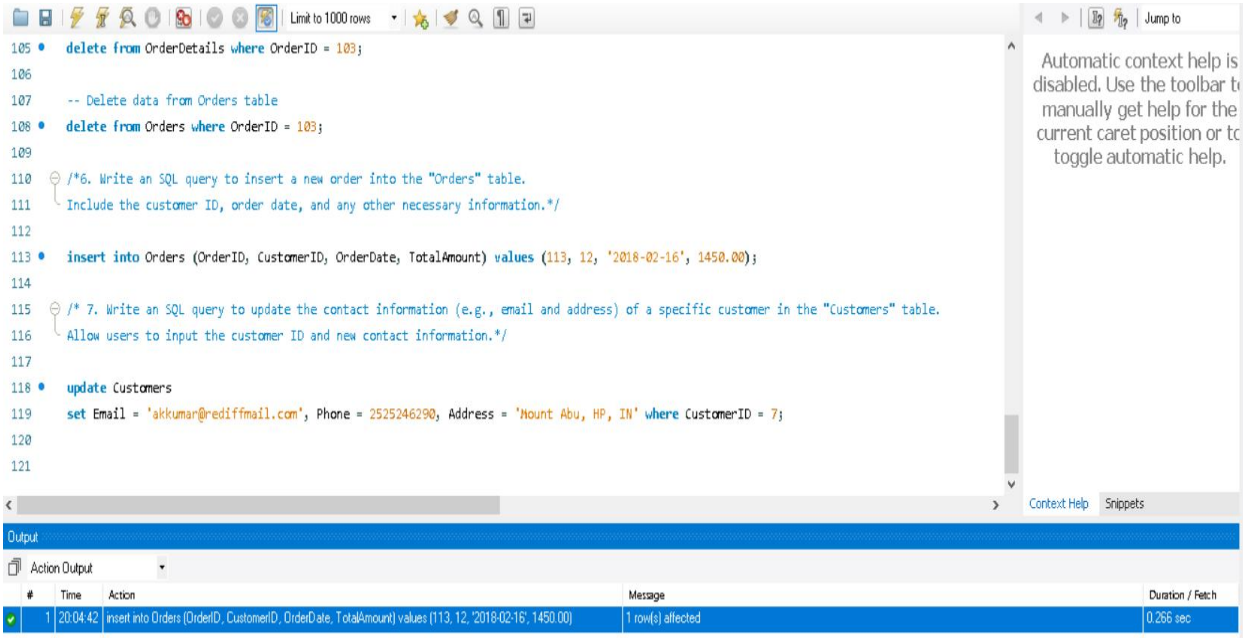
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or tc toggle automatic help.

	#	Time	Action	Message	Duration / Fetch
✓	1	19:55:06	delete from OrderDetails where OrderID = 103	1 row(s) affected	0.454 sec
✓	2	19:55:13	delete from Orders where OrderID = 103	1 row(s) affected	0.297 sec

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.

Query:

insert into Orders (OrderID, CustomerID, OrderDate, TotalAmount) values (113, 12, '2023-06-16', 1500.00);



```
delete from OrderDetails where OrderID = 103;

-- Delete data from Orders table
delete from Orders where OrderID = 103;

/*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 (OrderID, CustomerID, OrderDate, TotalAmount) values (113, 12, '2018-02-16', 1450.00);

/* 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.*/

update Customers
set Email = 'akkumar@rediffmail.com', Phone = 2525246290, Address = 'Mount Abu, HP, IN' where CustomerID = 7;
```

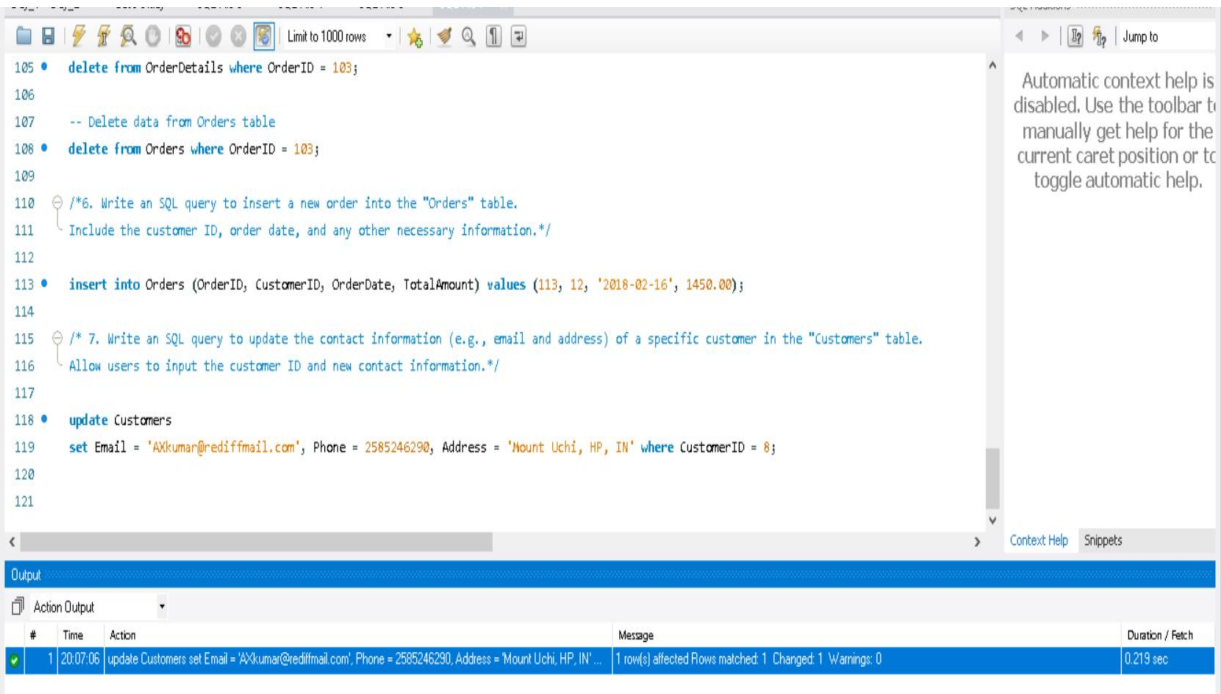
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or tc toggle automatic help.

	#	Time	Action	Message	Duration / Fetch
✓	1	20:04:42	insert into Orders (OrderID, CustomerID, OrderDate, TotalAmount) values (113, 12, '2018-02-16', 1450.00)	1 row(s) affected	0.266 sec

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.

Query:

```
update Customers
set Email = 'akkumar@rediffmail.com', Phone = 2525246290, Address = 'Mount Abu, HP, IN' where
CustomerID = 7;
```



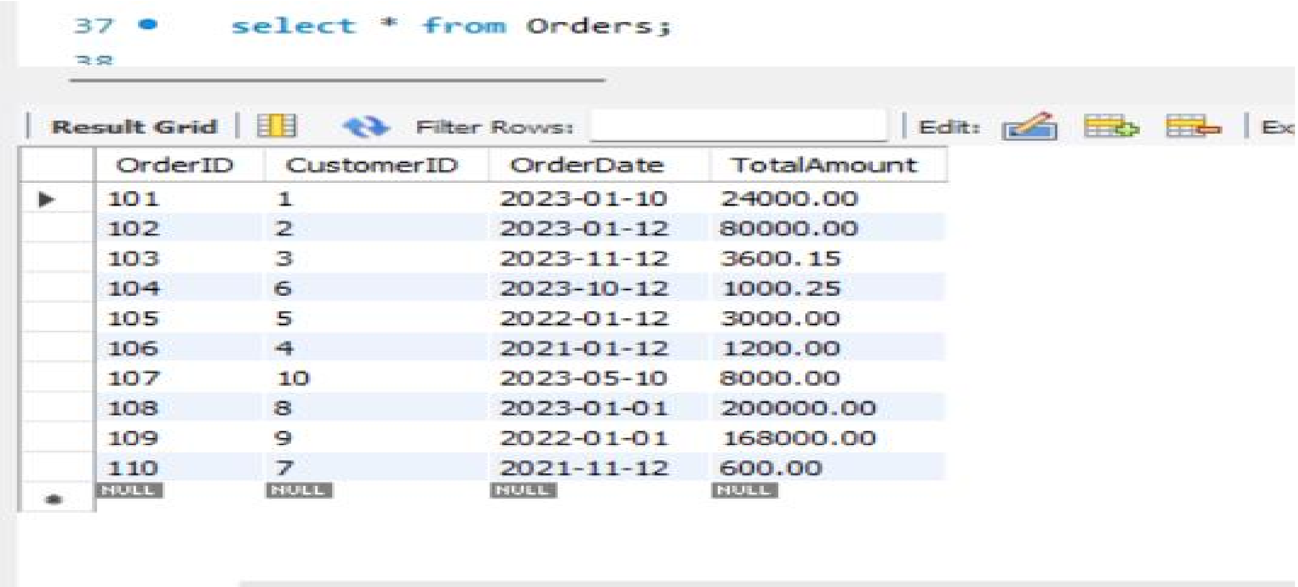
Limit to 1000 rows
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or tc toggle automatic help.
Context Help
Snippets
Output
Action Output

```
delete from OrderDetails where OrderID = 103;

-- Delete data from Orders table
delete from Orders where OrderID = 103;

/*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 (OrderID, CustomerID, OrderDate, TotalAmount) values (113, 12, '2018-02-16', 1450.00);

/* 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.*/

update Customers
set Email = 'AXkumar@rediffmail.com', Phone = 2585246290, Address = 'Mount Uchi, HP, IN' where CustomerID = 8;
```

	#	Time	Action	Message	Duration / Fetch
✓	1	20:07:06	update Customers set Email = 'AXkumar@rediffmail.com', Phone = 2585246290, Address = 'Mount Uchi, HP, IN' ...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.219 sec

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.

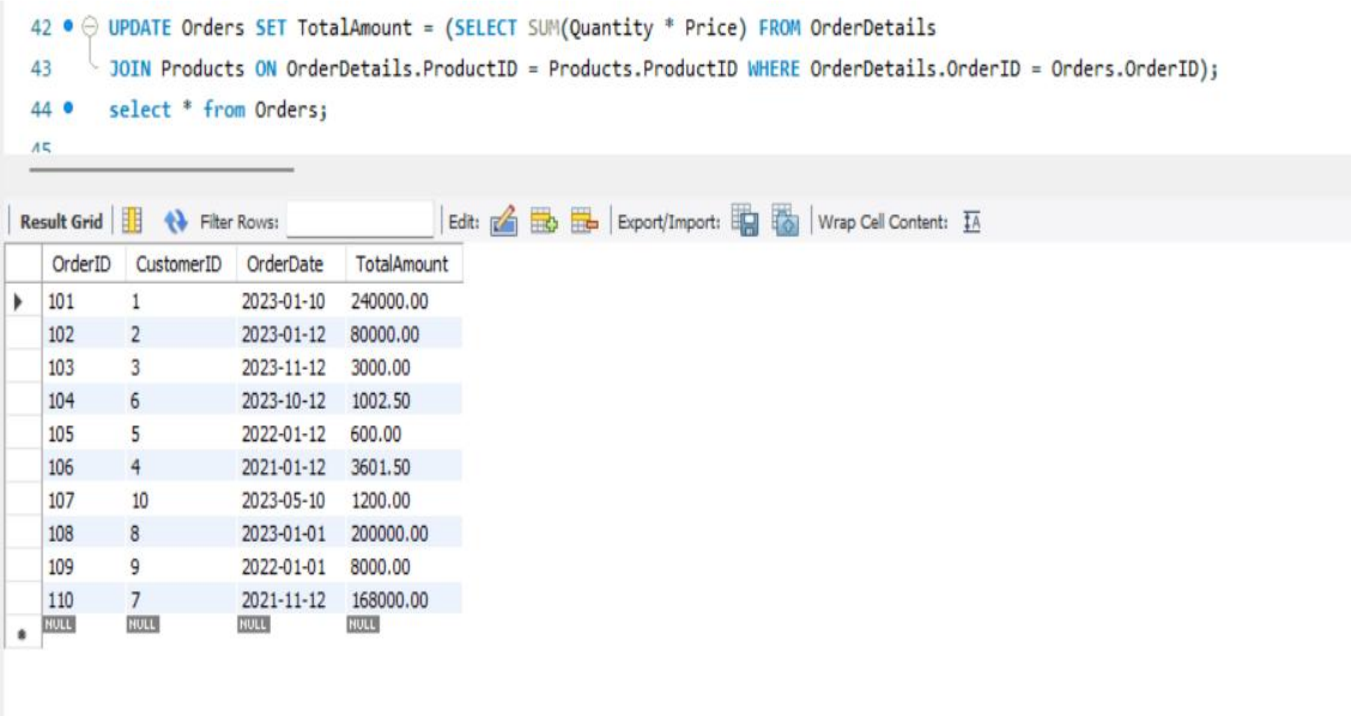
Order Table-



```sql
select * from Orders;
```

OrderID	CustomerID	OrderDate	TotalAmount
101	1	2023-01-10	24000.00
102	2	2023-01-12	80000.00
103	3	2023-11-12	3600.15
104	6	2023-10-12	1000.25
105	5	2022-01-12	3000.00
106	4	2021-01-12	1200.00
107	10	2023-05-10	8000.00
108	8	2023-01-01	200000.00
109	9	2022-01-01	168000.00
110	7	2021-11-12	600.00
NULL	NULL	NULL	NULL

*Query-update Orders set TotalAmount = (select sum(Quantity * Price) from OrderDetails join Products on OrderDetails.ProductID = Products.ProductID whereOrderDetails.OrderID = Orders.OrderID);*

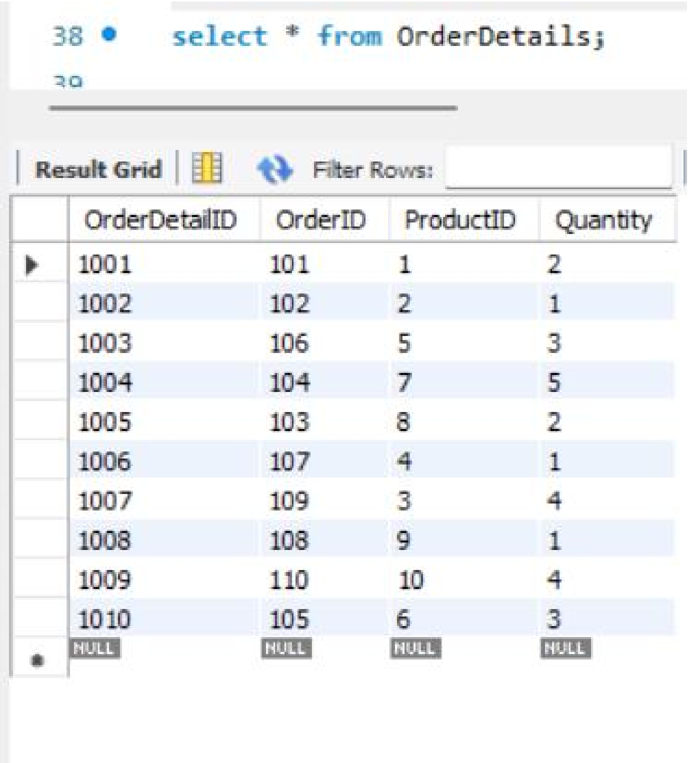


```sql
UPDATE Orders SET TotalAmount = (SELECT SUM(Quantity * Price) FROM OrderDetails
JOIN Products ON OrderDetails.ProductID = Products.ProductID WHERE OrderDetails.OrderID = Orders.OrderID);
select * from Orders;
```

OrderID	CustomerID	OrderDate	TotalAmount
101	1	2023-01-10	240000.00
102	2	2023-01-12	80000.00
103	3	2023-11-12	3000.00
104	6	2023-10-12	1002.50
105	5	2022-01-12	600.00
106	4	2021-01-12	3601.50
107	10	2023-05-10	1200.00
108	8	2023-01-01	200000.00
109	9	2022-01-01	8000.00
110	7	2021-11-12	168000.00
NULL	NULL	NULL	NULL

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.

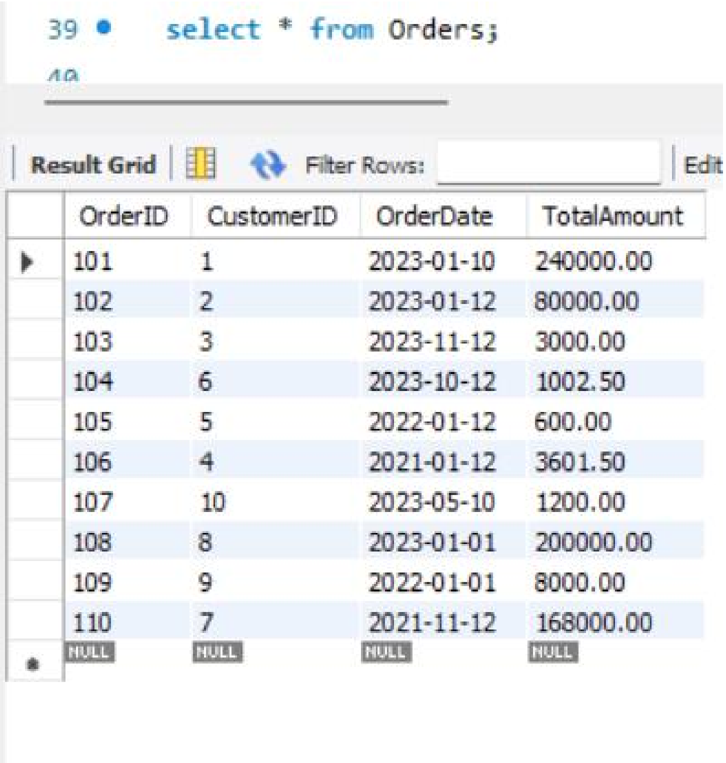
Order Details Table-



```
select * from OrderDetails;
```

OrderDetailID	OrderID	ProductID	Quantity
1001	101	1	2
1002	102	2	1
1003	106	5	3
1004	104	7	5
1005	103	8	2
1006	107	4	1
1007	109	3	4
1008	108	9	1
1009	110	10	4
1010	105	6	3
NULL	NULL	NULL	NULL

Order Table-



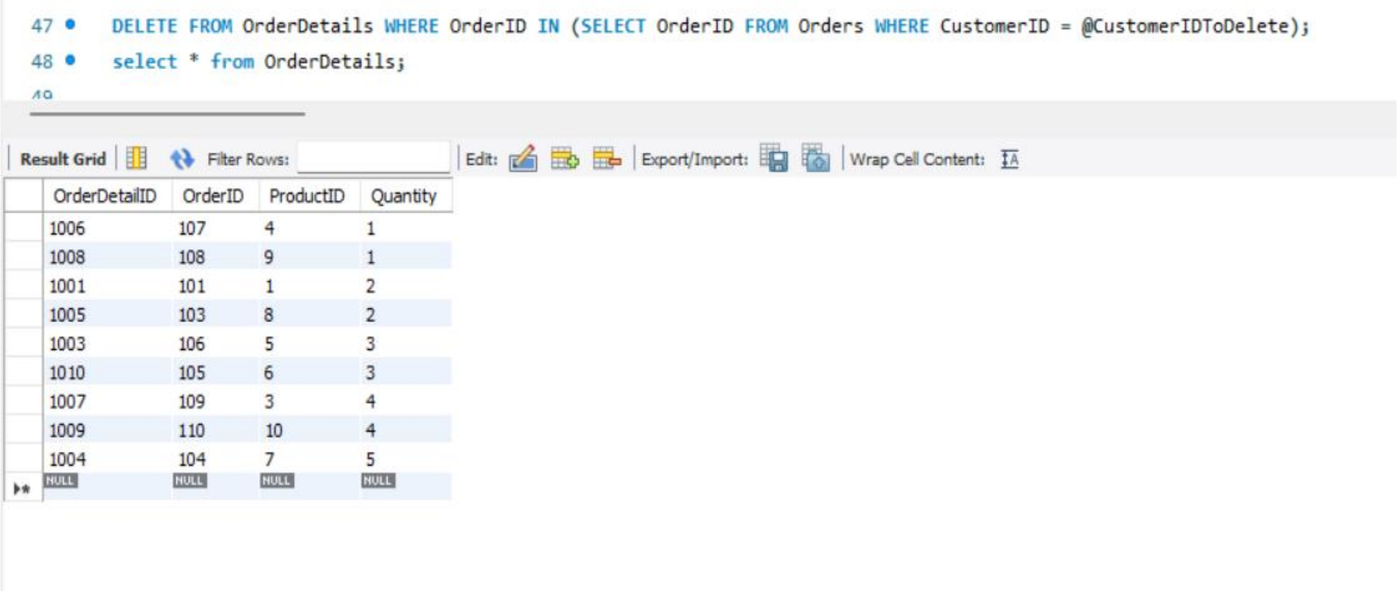
```
select * from Orders;
```

OrderID	CustomerID	OrderDate	TotalAmount
101	1	2023-01-10	240000.00
102	2	2023-01-12	80000.00
103	3	2023-11-12	3000.00
104	6	2023-10-12	1002.50
105	5	2022-01-12	600.00
106	4	2021-01-12	3601.50
107	10	2023-05-10	1200.00
108	8	2023-01-01	200000.00
109	9	2022-01-01	8000.00
110	7	2021-11-12	168000.00
NULL	NULL	NULL	NULL

Query-

SET @CustomerIDToDelete := 2;

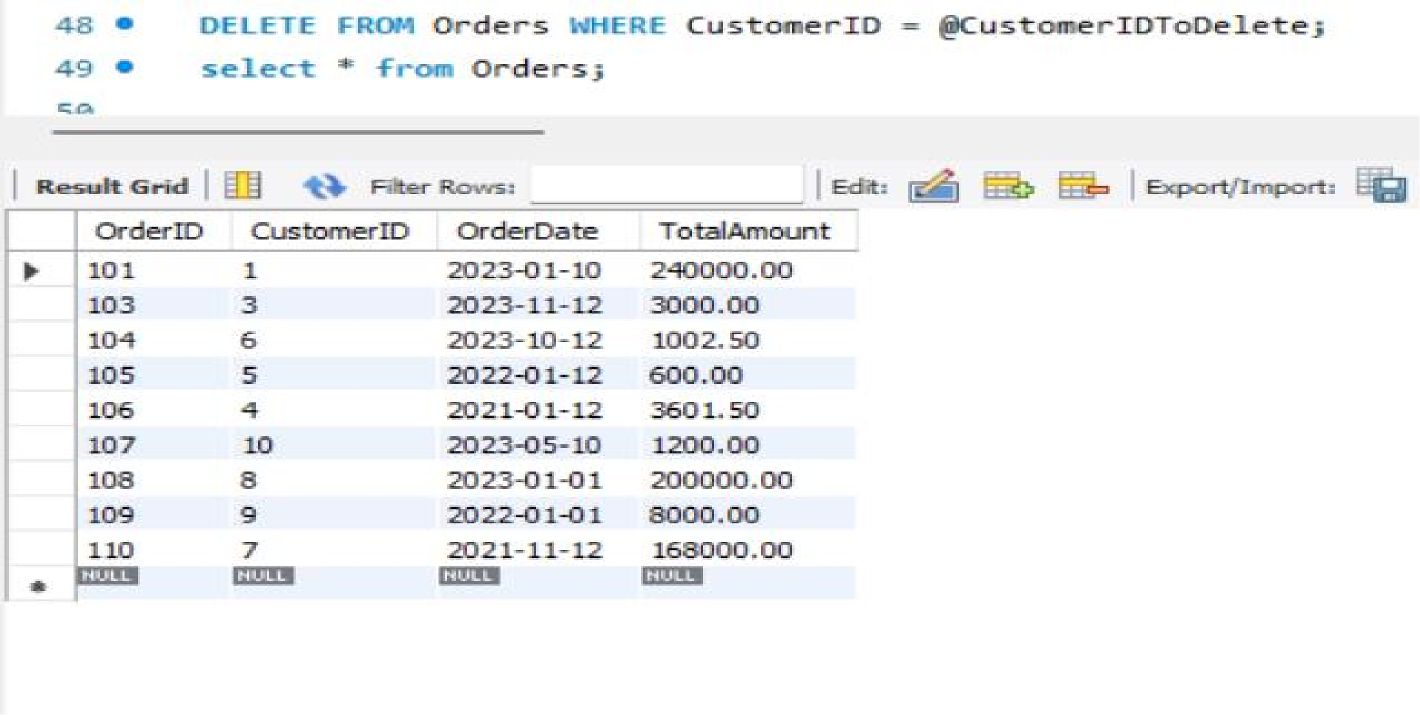
DELETE FROM OrderDetails WHERE OrderID IN (SELECT OrderID FROM Orders WHERE CustomerID = @CustomerIDToDelete);



```
DELETE FROM OrderDetails WHERE OrderID IN (SELECT OrderID FROM Orders WHERE CustomerID = @CustomerIDToDelete);
select * from OrderDetails;
```

OrderDetailID	OrderID	ProductID	Quantity
1006	107	4	1
1008	108	9	1
1001	101	1	2
1005	103	8	2
1003	106	5	3
1010	105	6	3
1007	109	3	4
1009	110	10	4
1004	104	7	5
NULL	NULL	NULL	NULL

DELETE FROM Orders WHERE CustomerID = @CustomerIDToDelete;

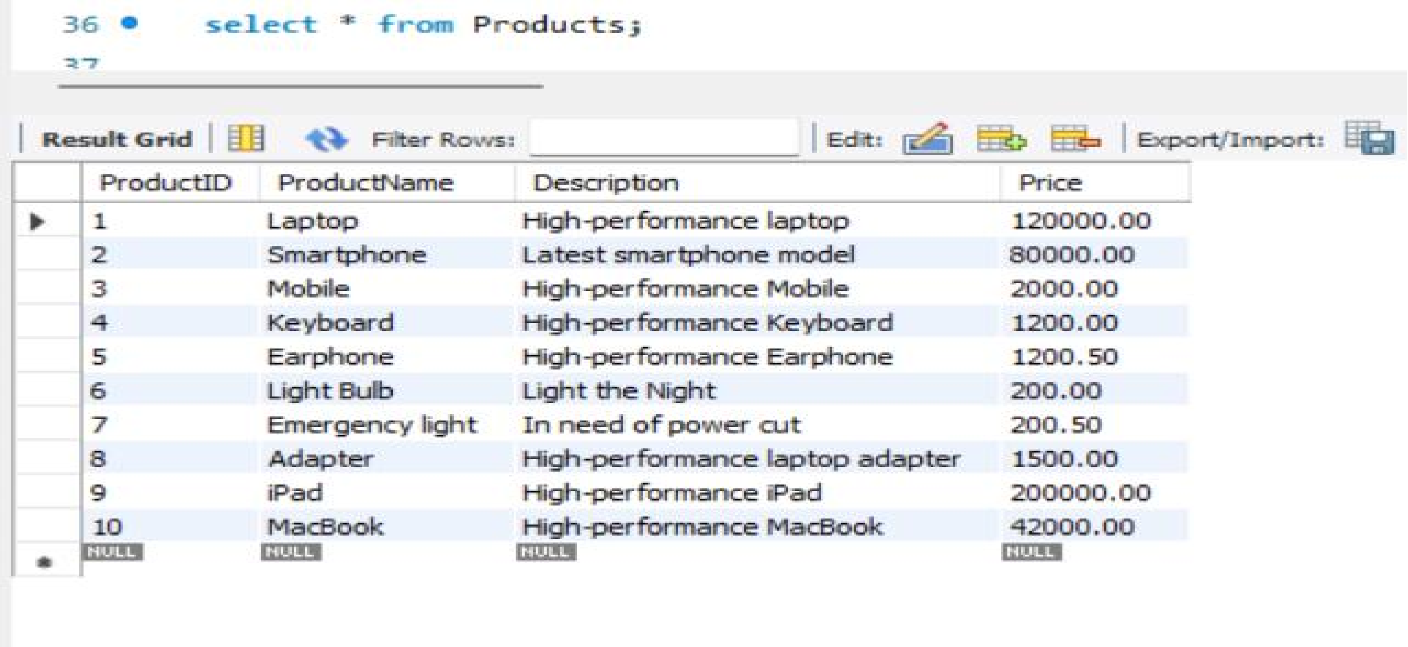


```sql
DELETE FROM Orders WHERE CustomerID = @CustomerIDToDelete;
select * from Orders;
```

OrderID	CustomerID	OrderDate	TotalAmount
101	1	2023-01-10	240000.00
103	3	2023-11-12	3000.00
104	6	2023-10-12	1002.50
105	5	2022-01-12	600.00
106	4	2021-01-12	3601.50
107	10	2023-05-10	1200.00
108	8	2023-01-01	200000.00
109	9	2022-01-01	8000.00
110	7	2021-11-12	168000.00
NULL	NULL	NULL	NULL

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.

Product Table-

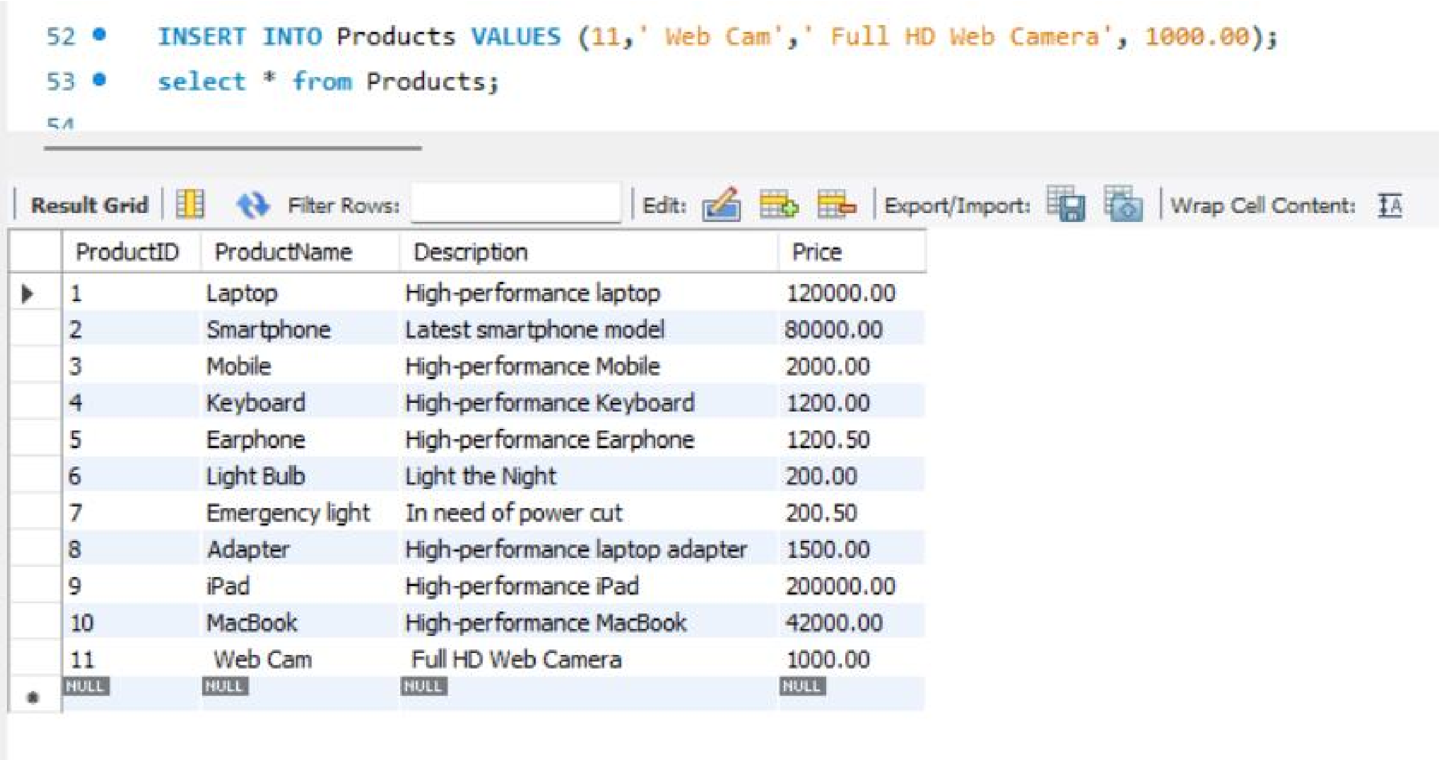


```sql
select * from Products;
```

ProductID	ProductName	Description	Price
1	Laptop	High-performance laptop	120000.00
2	Smartphone	Latest smartphone model	80000.00
3	Mobile	High-performance Mobile	2000.00
4	Keyboard	High-performance Keyboard	1200.00
5	Earphone	High-performance Earphone	1200.50
6	Light Bulb	Light the Night	200.00
7	Emergency light	In need of power cut	200.50
8	Adapter	High-performance laptop adapter	1500.00
9	iPad	High-performance iPad	200000.00
10	MacBook	High-performance MacBook	42000.00
NULL	NULL	NULL	NULL

Query-

INSERT INTO Products VALUES (11,' Web Cam',' Full HD Web Camera', 1000.00);

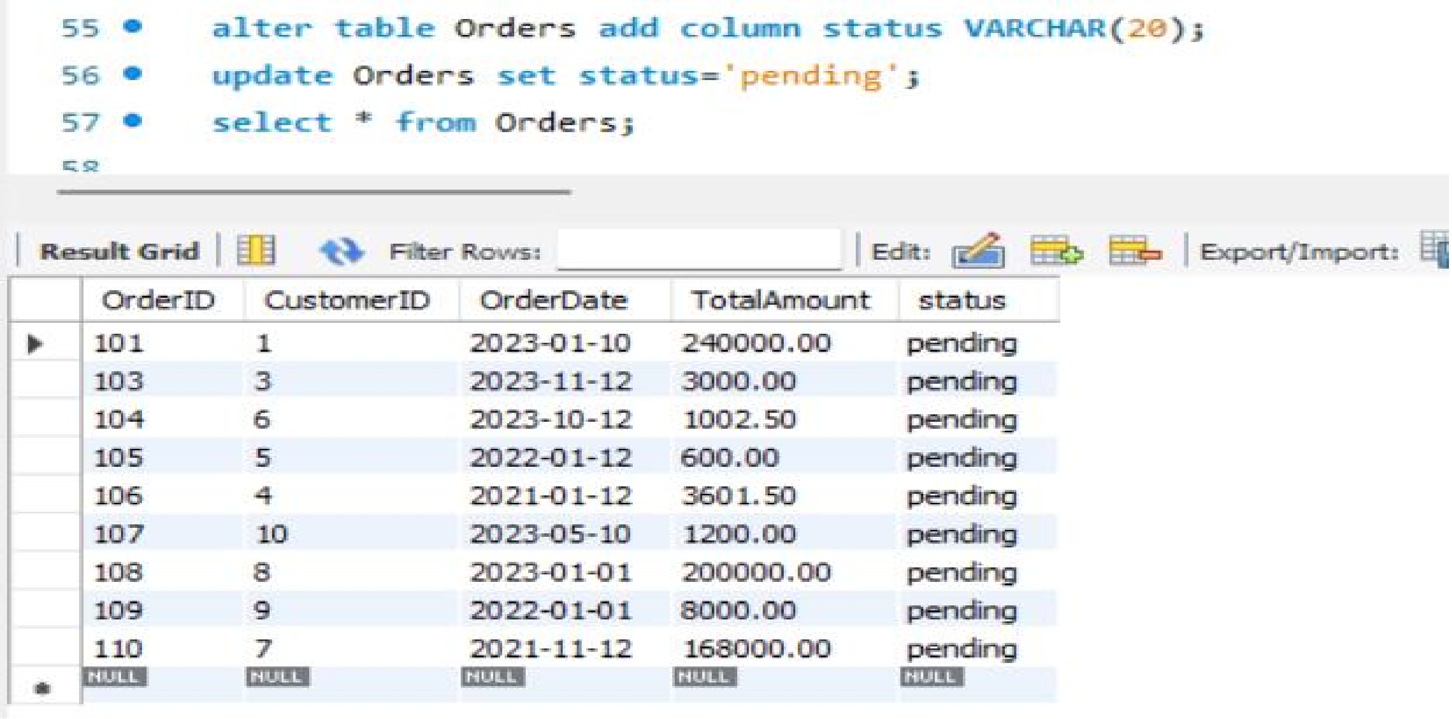


```sql
INSERT INTO Products VALUES (11,' Web Cam',' Full HD Web Camera', 1000.00);
select * from Products;
```

	ProductID	ProductName	Description	Price
▶	1	Laptop	High-performance laptop	120000.00
	2	Smartphone	Latest smartphone model	80000.00
	3	Mobile	High-performance Mobile	2000.00
	4	Keyboard	High-performance Keyboard	1200.00
	5	Earphone	High-performance Earphone	1200.50
	6	Light Bulb	Light the Night	200.00
	7	Emergency light	In need of power cut	200.50
	8	Adapter	High-performance laptop adapter	1500.00
	9	iPad	High-performance iPad	200000.00
	10	MacBook	High-performance MacBook	42000.00
	11	Web Cam	Full HD Web Camera	1000.00
*	NULL	NULL	NULL	NULL

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

New Order Table-

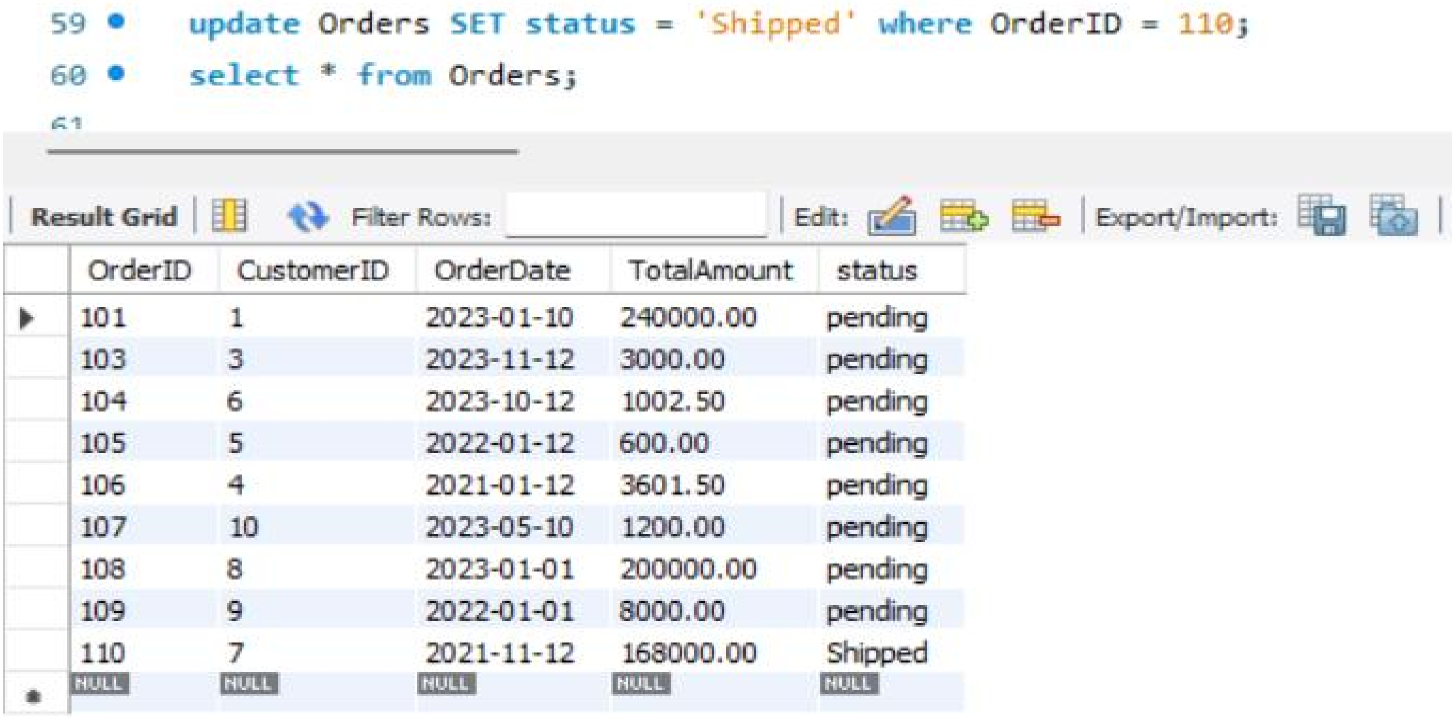


```sql
alter table Orders add column status VARCHAR(20);
update Orders set status='pending';
select * from Orders;
```

	OrderID	CustomerID	OrderDate	TotalAmount	status
▶	101	1	2023-01-10	240000.00	pending
	103	3	2023-11-12	3000.00	pending
	104	6	2023-10-12	1002.50	pending
	105	5	2022-01-12	600.00	pending
	106	4	2021-01-12	3601.50	pending
	107	10	2023-05-10	1200.00	pending
	108	8	2023-01-01	200000.00	pending
	109	9	2022-01-01	8000.00	pending
	110	7	2021-11-12	168000.00	pending
*	NULL	NULL	NULL	NULL	NULL

Query-

update Orders SET status = 'Shipped' where OrderID = 110;

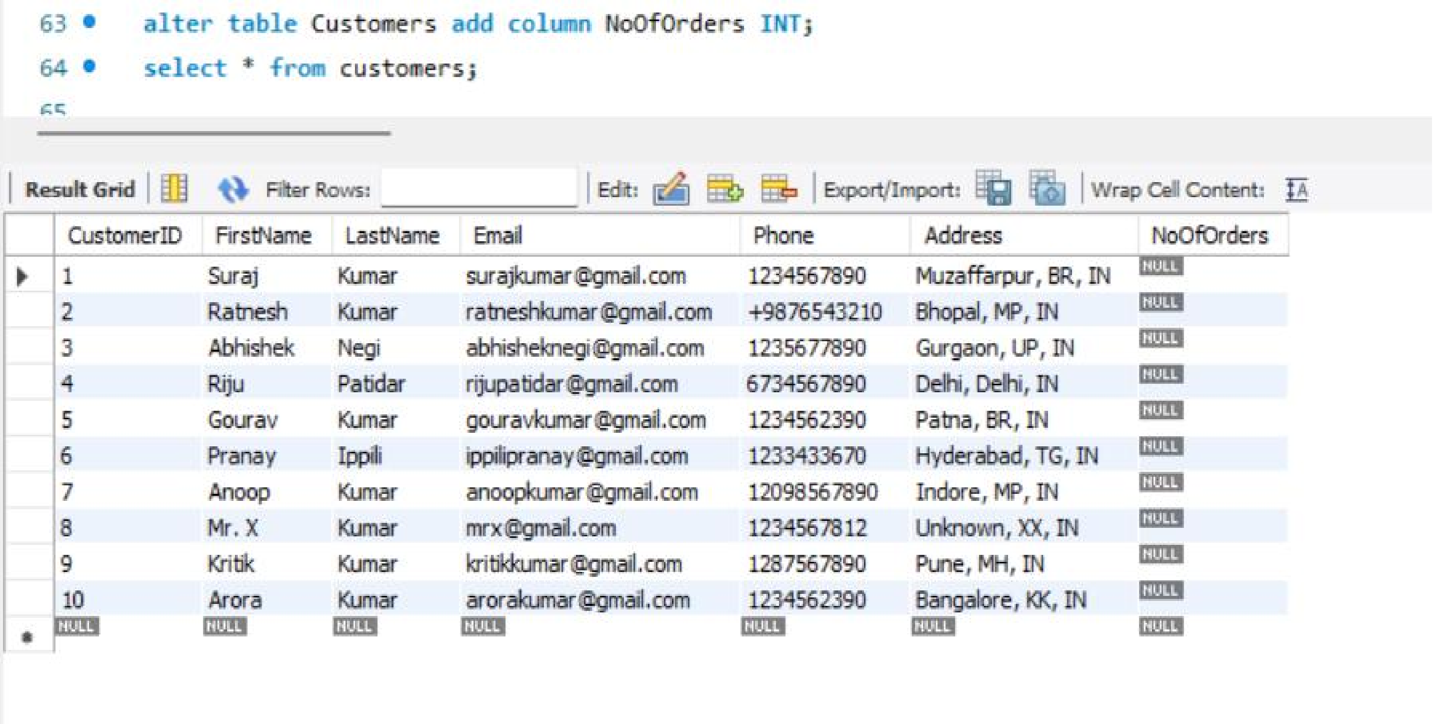


```
update Orders SET status = 'Shipped' where OrderID = 110;
select * from Orders;
```

OrderID	CustomerID	OrderDate	TotalAmount	status
101	1	2023-01-10	240000.00	pending
103	3	2023-11-12	3000.00	pending
104	6	2023-10-12	1002.50	pending
105	5	2022-01-12	600.00	pending
106	4	2021-01-12	3601.50	pending
107	10	2023-05-10	1200.00	pending
108	8	2023-01-01	200000.00	pending
109	9	2022-01-01	8000.00	pending
110	7	2021-11-12	168000.00	Shipped
NULL	NULL	NULL	NULL	NULL

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.

New Customer Table-

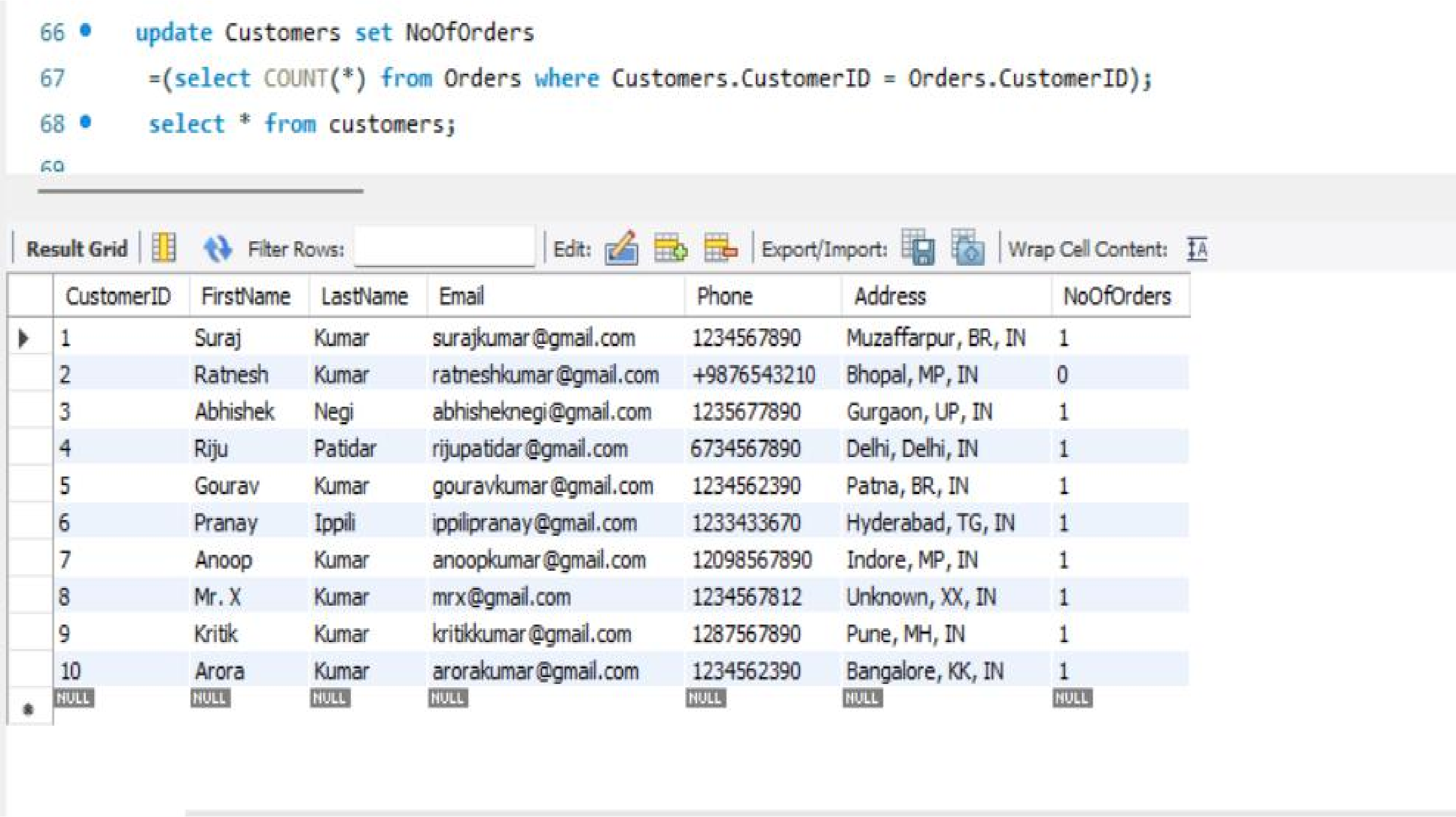


```
alter table Customers add column NoOfOrders INT;
select * from customers;
```

CustomerID	FirstName	LastName	Email	Phone	Address	NoOfOrders
1	Suraj	Kumar	surajkumar@gmail.com	1234567890	Muzaffarpur, BR, IN	NULL
2	Ratnesh	Kumar	ratneshkumar@gmail.com	+9876543210	Bhopal, MP, IN	NULL
3	Abhishek	Negi	abhisheknegi@gmail.com	1235677890	Gurgaon, UP, IN	NULL
4	Riju	Patidar	rijupatidar@gmail.com	6734567890	Delhi, Delhi, IN	NULL
5	Gourav	Kumar	gouravkumar@gmail.com	1234562390	Patna, BR, IN	NULL
6	Pranay	Ippili	ippilipranay@gmail.com	1233433670	Hyderabad, TG, IN	NULL
7	Anoop	Kumar	anoopkumar@gmail.com	12098567890	Indore, MP, IN	NULL
8	Mr. X	Kumar	mrx@gmail.com	1234567812	Unknown, XX, IN	NULL
9	Kritik	Kumar	kritikkumar@gmail.com	1287567890	Pune, MH, IN	NULL
10	Arora	Kumar	arorakumar@gmail.com	1234562390	Bangalore, KK, IN	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Query-

update Customers set NoOfOrders =(select COUNT(*) from Orders where Customers.CustomerID = Orders.CustomerID);



```
update Customers set NoOfOrders
 =(select COUNT(*) from Orders where Customers.CustomerID = Orders.CustomerID);
 select * from customers;
```

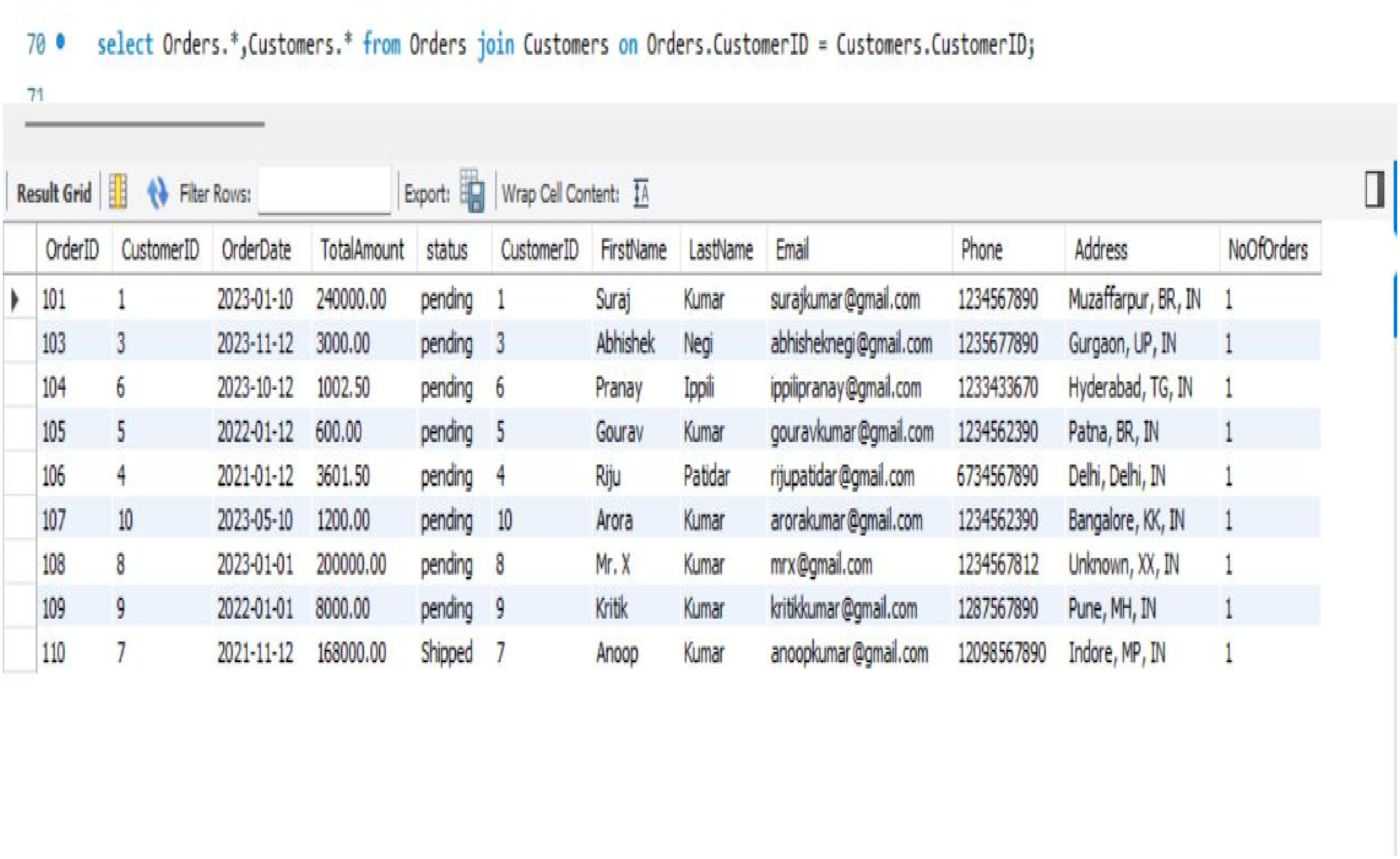
CustomerID	FirstName	LastName	Email	Phone	Address	NoOfOrders
1	Suraj	Kumar	surajkumar@gmail.com	1234567890	Muzaffarpur, BR, IN	1
2	Ratnesh	Kumar	ratneshkumar@gmail.com	+9876543210	Bhopal, MP, IN	0
3	Abhishek	Negi	abhisheknegi@gmail.com	1235677890	Gurgaon, UP, IN	1
4	Riju	Patidar	rijupatidar@gmail.com	6734567890	Delhi, Delhi, IN	1
5	Gourav	Kumar	gouravkumar@gmail.com	1234562390	Patna, BR, IN	1
6	Pranay	Ippili	ippilipranay@gmail.com	1233433670	Hyderabad, TG, IN	1
7	Anoop	Kumar	anoopkumar@gmail.com	12098567890	Indore, MP, IN	1
8	Mr. X	Kumar	mrx@gmail.com	1234567812	Unknown, XX, IN	1
9	Kritik	Kumar	kritikkumar@gmail.com	1287567890	Pune, MH, IN	1
10	Arora	Kumar	arorakumar@gmail.com	1234562390	Bangalore, KK, IN	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL

4. JOINS

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

Query-

select Orders.,Customers.* from Orders join Customers on Orders.CustomerID = Customers.CustomerID;*



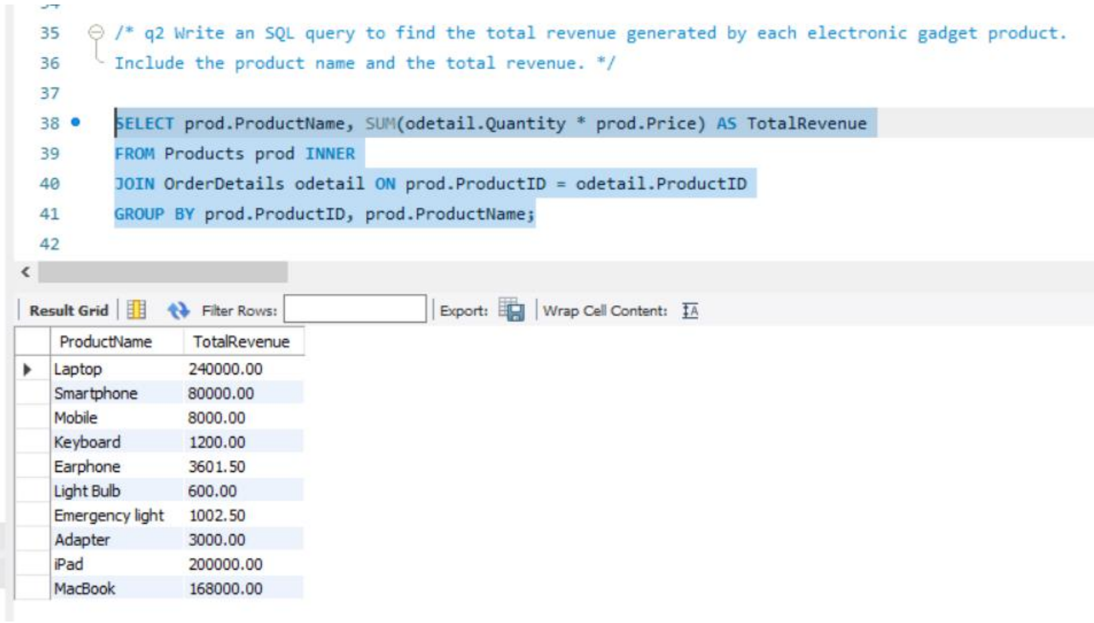
```
select Orders.*,Customers.* from Orders join Customers on Orders.CustomerID = Customers.CustomerID;
```

OrderID	CustomerID	OrderDate	TotalAmount	status	CustomerID	FirstName	LastName	Email	Phone	Address	NoOfOrders
101	1	2023-01-10	240000.00	pending	1	Suraj	Kumar	surajkumar@gmail.com	1234567890	Muzaffarpur, BR, IN	1
103	3	2023-11-12	3000.00	pending	3	Abhishek	Negi	abhisheknegi@gmail.com	1235677890	Gurgaon, UP, IN	1
104	6	2023-10-12	1002.50	pending	6	Pranay	Ippili	ippilipranay@gmail.com	1233433670	Hyderabad, TG, IN	1
105	5	2022-01-12	600.00	pending	5	Gourav	Kumar	gouravkumar@gmail.com	1234562390	Patna, BR, IN	1
106	4	2021-01-12	3601.50	pending	4	Riju	Patidar	rijupatidar@gmail.com	6734567890	Delhi, Delhi, IN	1
107	10	2023-05-10	1200.00	pending	10	Arora	Kumar	arorakumar@gmail.com	1234562390	Bangalore, KK, IN	1
108	8	2023-01-01	200000.00	pending	8	Mr. X	Kumar	mrx@gmail.com	1234567812	Unknown, XX, IN	1
109	9	2022-01-01	8000.00	pending	9	Kritik	Kumar	kritikkumar@gmail.com	1287567890	Pune, MH, IN	1
110	7	2021-11-12	168000.00	Shipped	7	Anoop	Kumar	anoopkumar@gmail.com	12098567890	Indore, MP, IN	1

Q2 Write an SQL query to find the total revenue generated by each electronic gadget product. Include the product name and the total revenue.

Query

```
SELECT prod.ProductName, SUM(odetail.Quantity * prod.Price) AS TotalRevenue
FROM Products prod INNER
JOIN OrderDetails odetail ON prod.ProductID = odetail.ProductID
GROUP BY prod.ProductID, prod.ProductName;
```



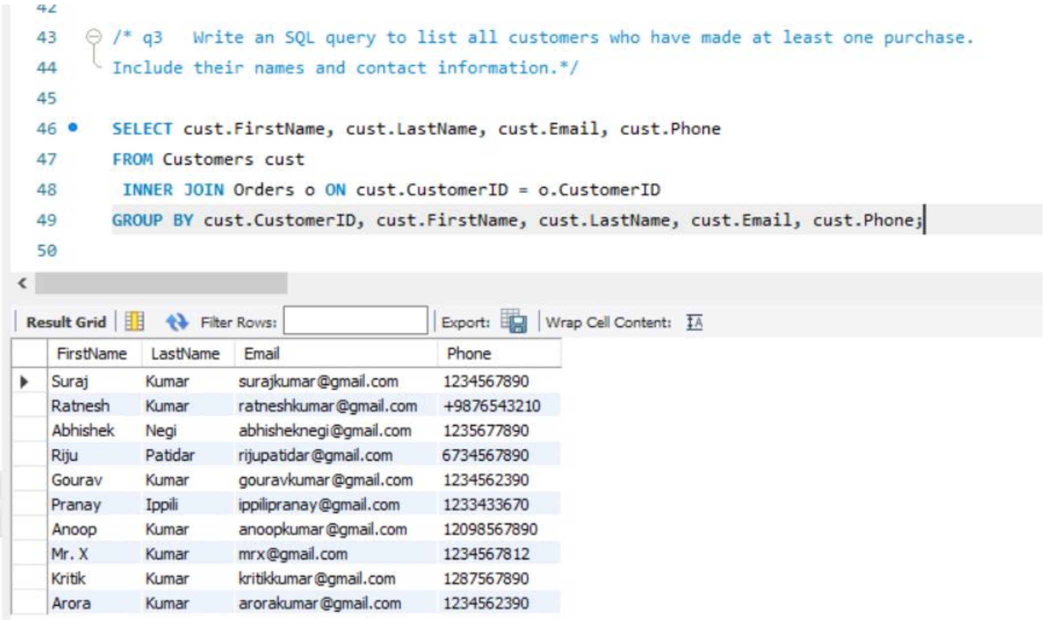
```
/* q2 Write an SQL query to find the total revenue generated by each electronic gadget product.
Include the product name and the total revenue. */

SELECT prod.ProductName, SUM(odetail.Quantity * prod.Price) AS TotalRevenue
FROM Products prod INNER
JOIN OrderDetails odetail ON prod.ProductID = odetail.ProductID
GROUP BY prod.ProductID, prod.ProductName;
```

ProductName	TotalRevenue
Laptop	240000.00
Smartphone	80000.00
Mobile	8000.00
Keyboard	1200.00
Earphone	3601.50
Light Bulb	600.00
Emergency light	1002.50
Adapter	3000.00
iPad	200000.00
MacBook	168000.00

Q3 Write an SQL query to list all customers who have made at least one purchase. Include their names and contact information.

Query

```
SELECT cust.FirstName, cust.LastName, cust.Email, cust.Phone
FROM Customers cust
INNER JOIN Orders o ON cust.CustomerID = o.CustomerID
GROUP BY cust.CustomerID, cust.FirstName, cust.LastName, cust.Email, cust.Phone;
```



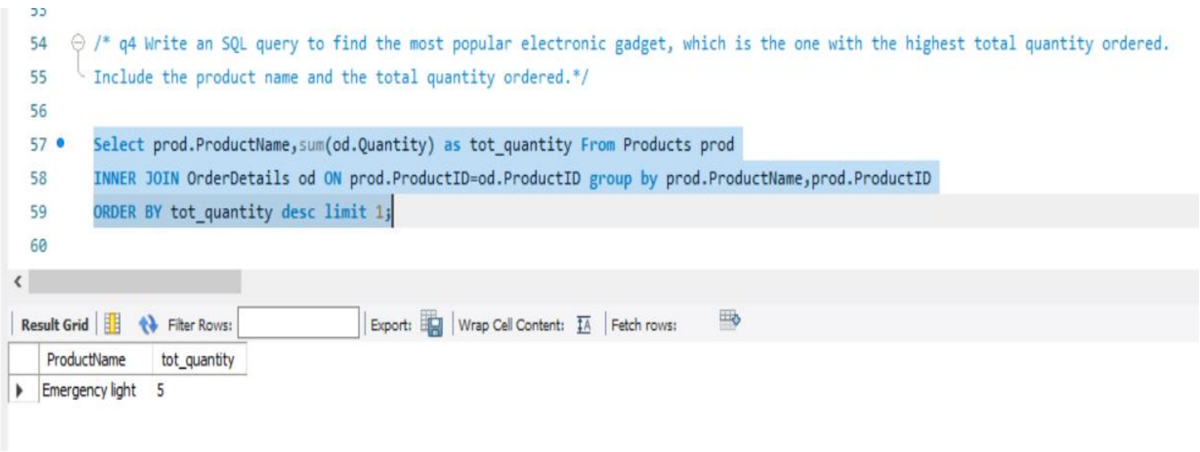
```
/* q3  Write an SQL query to list all customers who have made at least one purchase.
Include their names and contact information.*/

SELECT cust.FirstName, cust.LastName, cust.Email, cust.Phone
FROM Customers cust
 INNER JOIN Orders o ON cust.CustomerID = o.CustomerID
GROUP BY cust.CustomerID, cust.FirstName, cust.LastName, cust.Email, cust.Phone;
```

FirstName	LastName	Email	Phone
Suraj	Kumar	surajkumar@gmail.com	1234567890
Ratnesh	Kumar	ratneshkumar@gmail.com	+9876543210
Abhishek	Negi	abhisheknegi@gmail.com	1235677890
Riju	Patidar	rijupatidar@gmail.com	6734567890
Gourav	Kumar	gouravkumar@gmail.com	1234562390
Pranay	Ippili	ippilipranay@gmail.com	1233433670
Anoop	Kumar	anoopkumar@gmail.com	12098567890
Mr. X	Kumar	mrx@gmail.com	1234567812
Kritik	Kumar	kritikkumar@gmail.com	1287567890
Arora	Kumar	arorakumar@gmail.com	1234562390

Q4 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.

Query

```
Select prod.ProductName,sum(od.Quantity) as tot_quantity From Products prod
INNER JOIN OrderDetails od ON prod.ProductID=od.ProductID group by prod.ProductName,prod.ProductID
ORDER BY tot_quantity desc limit 1;
```



```
/* q4 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 prod.ProductName,sum(od.Quantity) as tot_quantity From Products prod
INNER JOIN OrderDetails od ON prod.ProductID=od.ProductID group by prod.ProductName,prod.ProductID
ORDER BY tot_quantity desc limit 1;
```

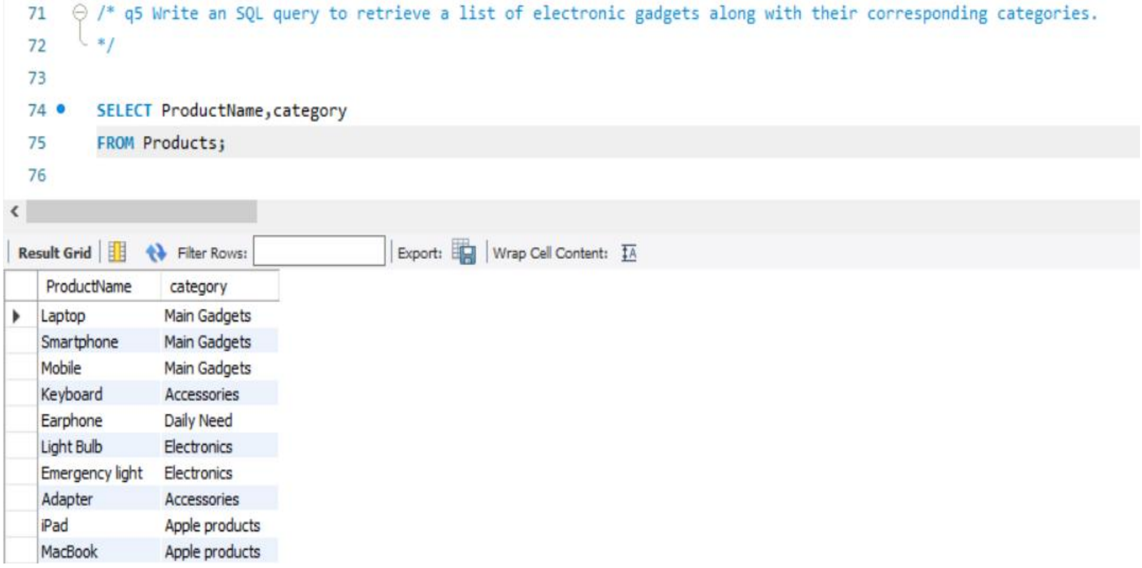
ProductName	tot_quantity
Emergency light	5

Q5 Write an SQL query to retrieve a list of electronic gadgets along with their corresponding categories.

```
ALTER Table Products
ADD Column Category varchar(50);
UPDATE Products SET Category='Main Gadgets' WHERE Productname IN ('Laptop','Smartphone','Mobile');
UPDATE Products SET Category='Accessories' WHERE Productname IN ('keyboard','Adapter');
UPDATE Products SET Category='Apple products' WHERE Productname IN ('IPad','Macbook');
UPDATE Products SET Category='Electronics' WHERE Productname IN ('Light Bulb','Emergency Light');
UPDATE Products SET Category='Daily Need' WHERE Productname IN ('Earphone');
```

Query

```
SELECT ProductName,category
FROM Products;
```



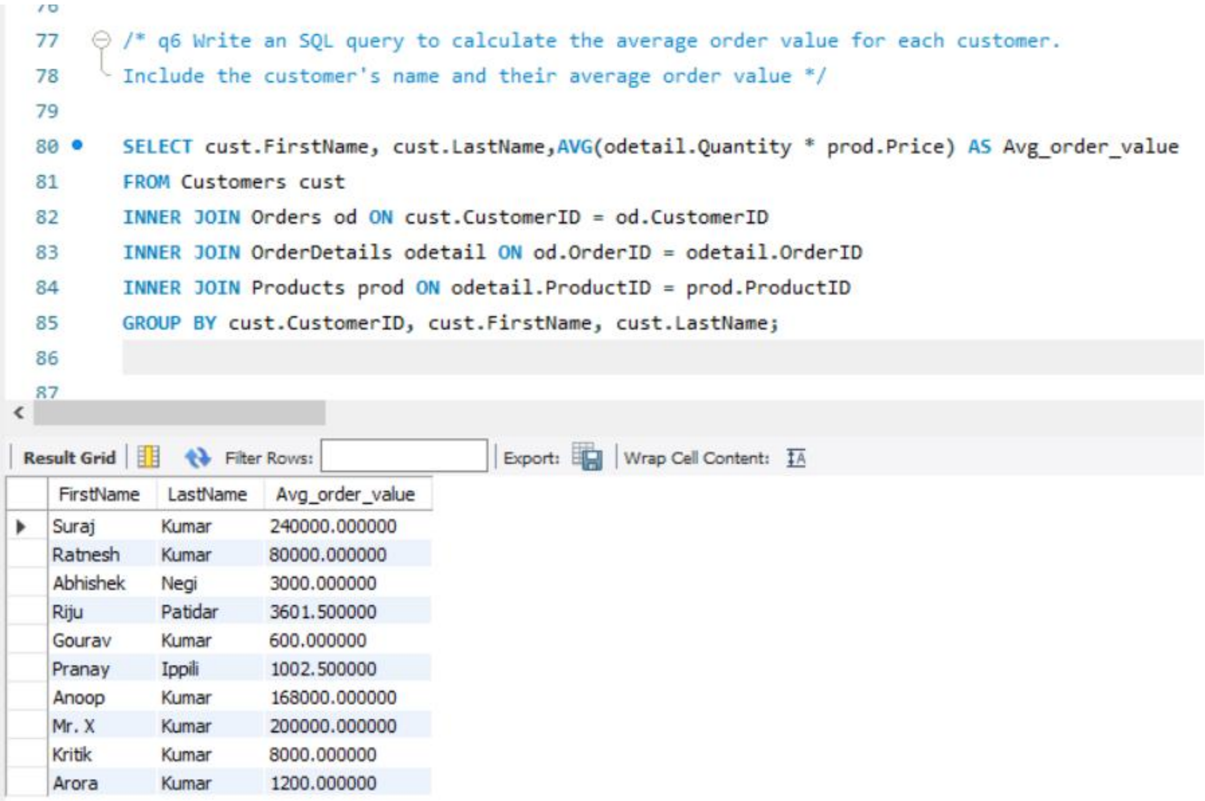
```
/* q5 Write an SQL query to retrieve a list of electronic gadgets along with their corresponding categories.
*/

SELECT ProductName,category
FROM Products;
```

ProductName	category
Laptop	Main Gadgets
Smartphone	Main Gadgets
Mobile	Main Gadgets
Keyboard	Accessories
Earphone	Daily Need
Light Bulb	Electronics
Emergency light	Electronics
Adapter	Accessories
iPad	Apple products
MacBook	Apple products

Q6 Write an SQL query to calculate the average order value for each customer. Include the customer's name and their average order value

Query

```
SELECT cust.FirstName, cust.LastName,AVG(odetail.Quantity * prod.Price) AS Avg_order_value
FROM Customers cust
INNER JOIN Orders od ON cust.CustomerID = od.CustomerID
INNER JOIN OrderDetails odetail ON od.OrderID = odetail.OrderID
INNER JOIN Products prod ON odetail.ProductID = prod.ProductID
GROUP BY cust.CustomerID, cust.FirstName, cust.LastName;
```



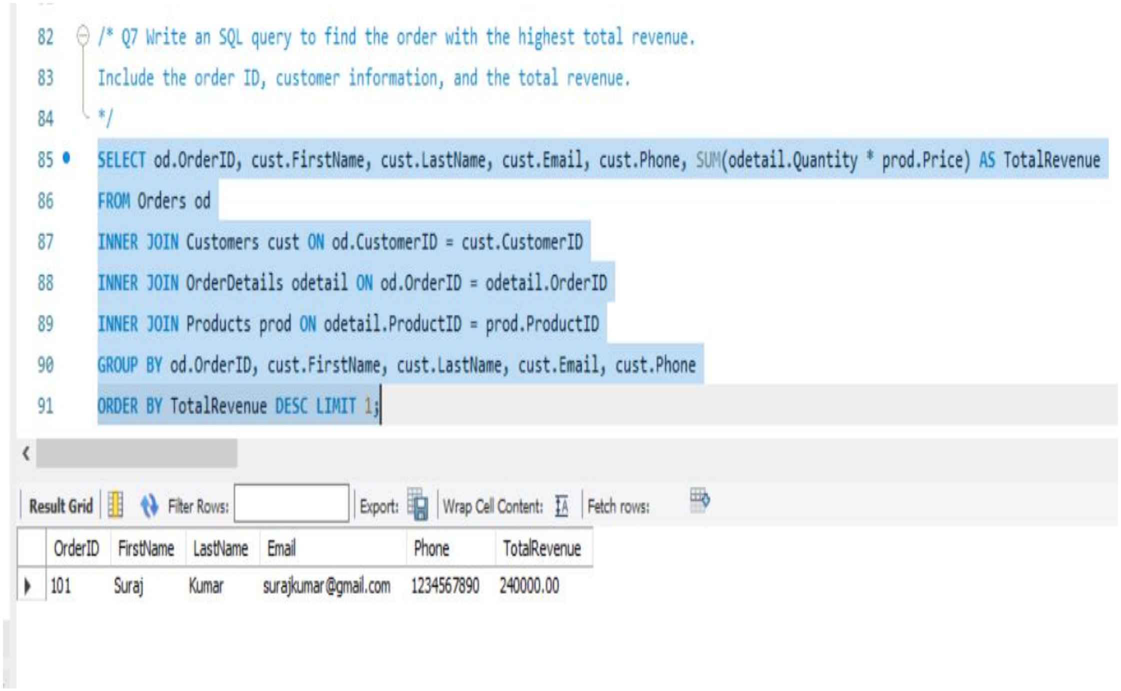
```
/* q6 Write an SQL query to calculate the average order value for each customer.
Include the customer's name and their average order value */

SELECT cust.FirstName, cust.LastName,AVG(odetail.Quantity * prod.Price) AS Avg_order_value
FROM Customers cust
INNER JOIN Orders od ON cust.CustomerID = od.CustomerID
INNER JOIN OrderDetails odetail ON od.OrderID = odetail.OrderID
INNER JOIN Products prod ON odetail.ProductID = prod.ProductID
GROUP BY cust.CustomerID, cust.FirstName, cust.LastName;
```

FirstName	LastName	Avg_order_value
Suraj	Kumar	240000.000000
Ratnesh	Kumar	80000.000000
Abhishek	Negi	3000.000000
Riju	Patidar	3601.500000
Gourav	Kumar	600.000000
Pranay	Ippili	1002.500000
Anoop	Kumar	168000.000000
Mr. X	Kumar	200000.000000
Kritik	Kumar	8000.000000
Arora	Kumar	1200.000000

Q7 Write an SQL query to find the order with the highest total revenue. Include the order ID, customer information, and the total revenue.

Query

```
SELECT od.OrderID, cust.FirstName, cust.LastName, cust.Email, cust.Phone, SUM(odetail.Quantity
* prod.Price) AS TotalRevenue
FROM Orders od
INNER JOIN Customers cust ON od.CustomerID = cust.CustomerID
INNER JOIN OrderDetails odetail ON od.OrderID = odetail.OrderID
INNER JOIN Products prod ON odetail.ProductID = prod.ProductID
GROUP BY od.OrderID, cust.FirstName, cust.LastName, cust.phone
ORDER BY TotalRevenue DESC LIMIT 1;
```



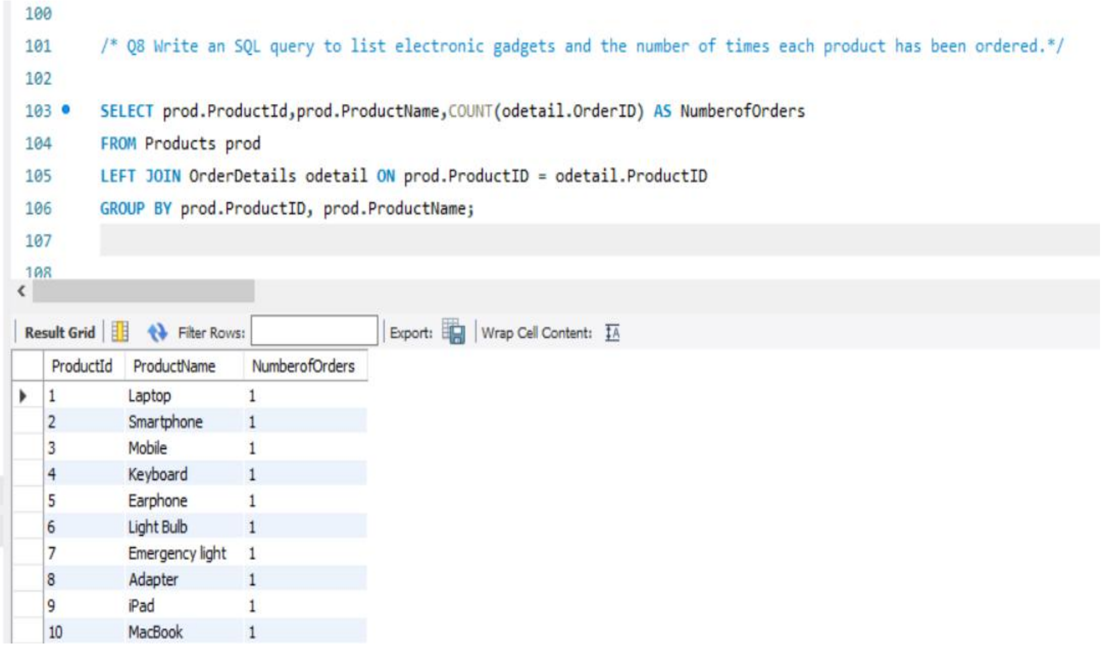
```
/* Q7 Write an SQL query to find the order with the highest total revenue.
Include the order ID, customer information, and the total revenue.
*/
SELECT od.OrderID, cust.FirstName, cust.LastName, cust.Email, cust.Phone, SUM(odetail.Quantity * prod.Price) AS TotalRevenue
FROM Orders od
INNER JOIN Customers cust ON od.CustomerID = cust.CustomerID
INNER JOIN OrderDetails odetail ON od.OrderID = odetail.OrderID
INNER JOIN Products prod ON odetail.ProductID = prod.ProductID
GROUP BY od.OrderID, cust.FirstName, cust.LastName, cust.Email, cust.Phone
ORDER BY TotalRevenue DESC LIMIT 1;
```

OrderID	FirstName	LastName	Email	Phone	TotalRevenue
101	Suraj	Kumar	surajkumar@gmail.com	1234567890	240000.00

Q8 Write an SQL query to list electronic gadgets and the number of times each product has been ordered

Query

SELECT prod.ProductId,prod.ProductName,COUNT(odetail.OrderID) AS NumberofOrders
FROM Products prod
LEFT JOIN OrderDetails odetail ON prod.ProductID = odetail.ProductID
GROUP BY prod.ProductID, prod.ProductName;



```
/* Q8 Write an SQL query to list electronic gadgets and the number of times each product has been ordered.*/

SELECT prod.ProductId,prod.ProductName,COUNT(odetail.OrderID) AS NumberofOrders
FROM Products prod
LEFT JOIN OrderDetails odetail ON prod.ProductID = odetail.ProductID
GROUP BY prod.ProductID, prod.ProductName;
```

ProductId	ProductName	NumberofOrders
1	Laptop	1
2	Smartphone	1
3	Mobile	1
4	Keyboard	1
5	Earphone	1
6	Light Bulb	1
7	Emergency light	1
8	Adapter	1
9	iPad	1
10	MacBook	1

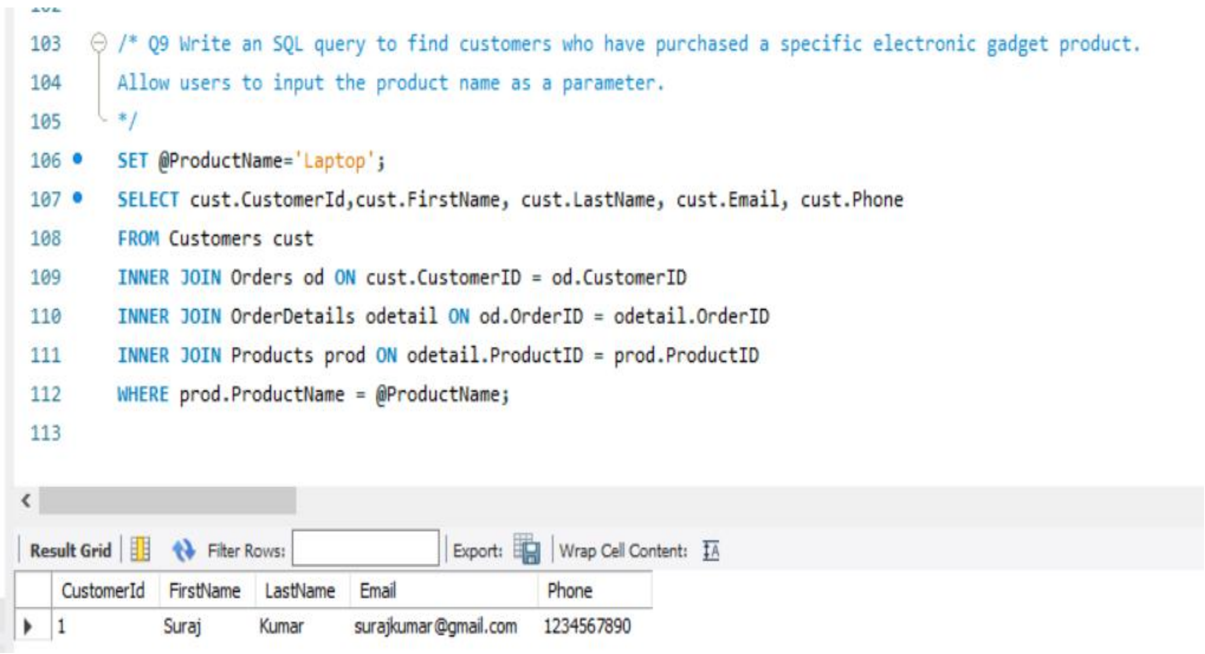
Q9 Write an SQL query to find customers who have purchased a specific electronic gadget product. Allow users to input the product name as a parameter.

Query

SET @ProductName=’Laptop’

SELECT cust.CustomerId, cust.FirstName, cust.LastName, cust.Email, cust.Phone

```
FROM Customers cust
INNER JOIN Orders od ON cust.CustomerID = od.CustomerID
INNER JOIN OrderDetails odetail ON od.OrderID = odetail.OrderID
INNER JOIN Products prod ON odetail.ProductID = prod.ProductID
WHERE prod.ProductName = @ProductName;
```

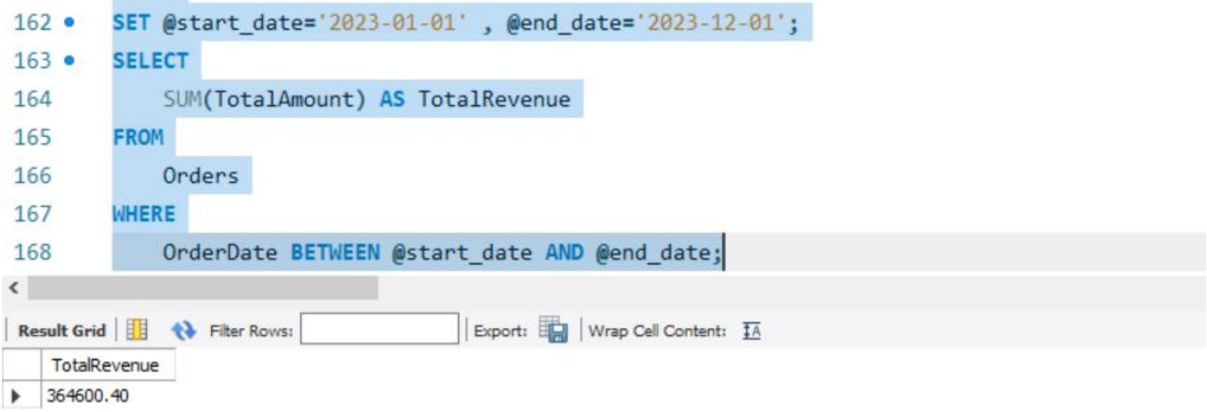


```
/* Q9 Write an SQL query to find customers who have purchased a specific electronic gadget product.
Allow users to input the product name as a parameter.
*/
SET @ProductName='Laptop';
SELECT cust.CustomerId,cust.FirstName, cust.LastName, cust.Email, cust.Phone
FROM Customers cust
INNER JOIN Orders od ON cust.CustomerID = od.CustomerID
INNER JOIN OrderDetails odetail ON od.OrderID = odetail.OrderID
INNER JOIN Products prod ON odetail.ProductID = prod.ProductID
WHERE prod.ProductName = @ProductName;
```

CustomerId	FirstName	LastName	Email	Phone
1	Suraj	Kumar	surajkumar@gmail.com	1234567890

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.

```
SET @start_date='2023-01-01' , @end_date='2023-12-01';
SELECT
   SUM(TotalAmount) AS TotalRevenue
FROM
   Orders
WHERE
OrderDate BETWEEN @start_date AND
@end_date;
```



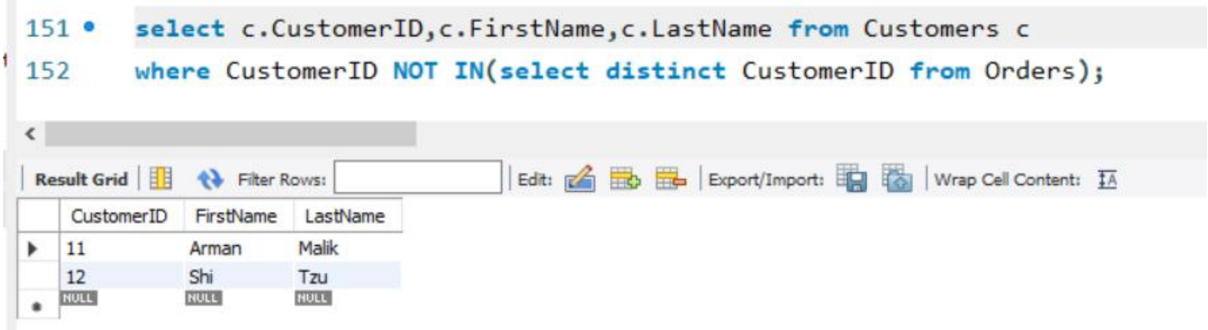
```
SET @start_date='2023-01-01' , @end_date='2023-12-01';
SELECT
    SUM(TotalAmount) AS TotalRevenue
FROM
    Orders
WHERE
    OrderDate BETWEEN @start_date AND @end_date;
```

TotalRevenue
364600.40

#Subquery

1. Write an SQL query to find out which customers have not placed any orders.

```
select c.CustomerID,c.FirstName,c.LastName from Customers c
where CustomerID NOT IN(select distinct CustomerID from Orders);
```

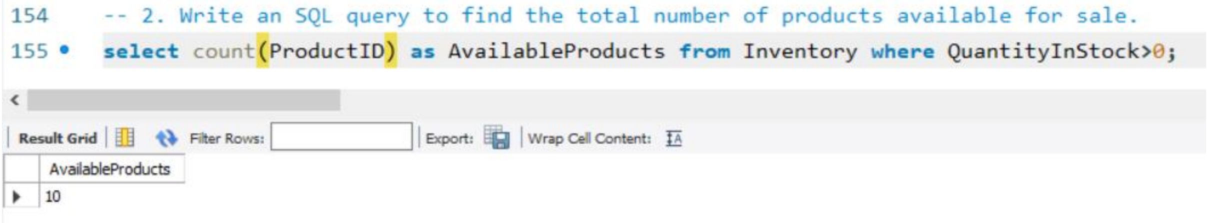


```
select c.CustomerID,c.FirstName,c.LastName from Customers c
where CustomerID NOT IN(select distinct CustomerID from Orders);
```

CustomerID	FirstName	LastName
11	Arman	Malik
12	Shi	Tzu
NULL	NULL	NULL

2. Write an SQL query to find the total number of products available for sale.

select count(ProductID) as AvailableProducts from Inventory where QuantityInStock>0;

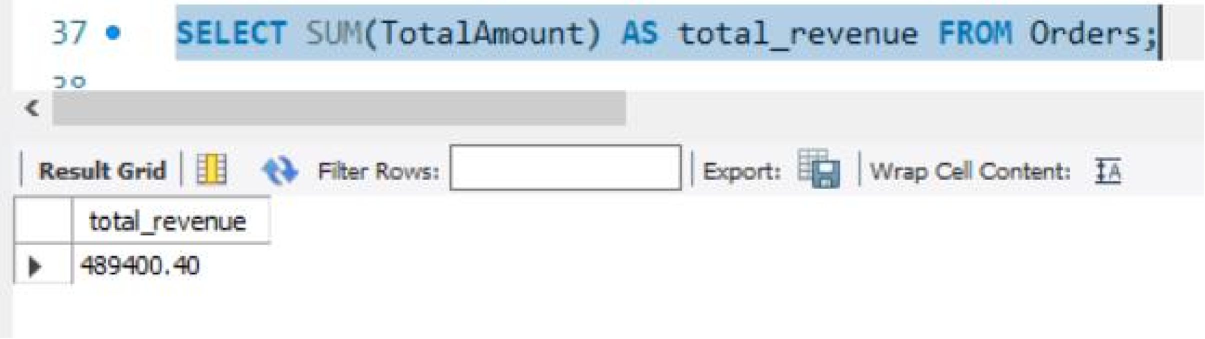


```
-- 2. Write an SQL query to find the total number of products available for sale.
select count(ProductID) as AvailableProducts from Inventory where QuantityInStock>0;
```

AvailableProducts
10

3. Write an SQL query to calculate the total revenue generated by TechShop.

SELECT SUM(TotalAmount) AS total_revenue FROM Orders;

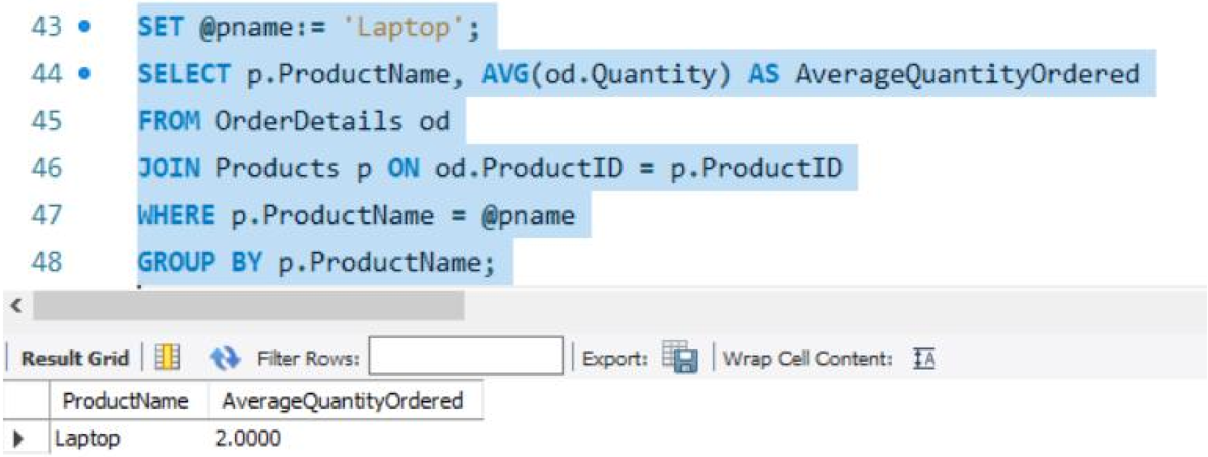


```
SELECT SUM(TotalAmount) AS total_revenue FROM Orders;
```

total_revenue
489400.40

4. Write an SQL query to calculate the average quantity ordered for products in a specific category. Allow users to input the category name as a parameter.

SET @pname:= 'Laptop';
SELECT p.ProductName, AVG(od.Quantity) AS AverageQuantityOrdered
FROM OrderDetails od
JOIN Products p ON od.ProductID = p.ProductID
WHERE p.ProductName = @pname
GROUP BY p.ProductName;

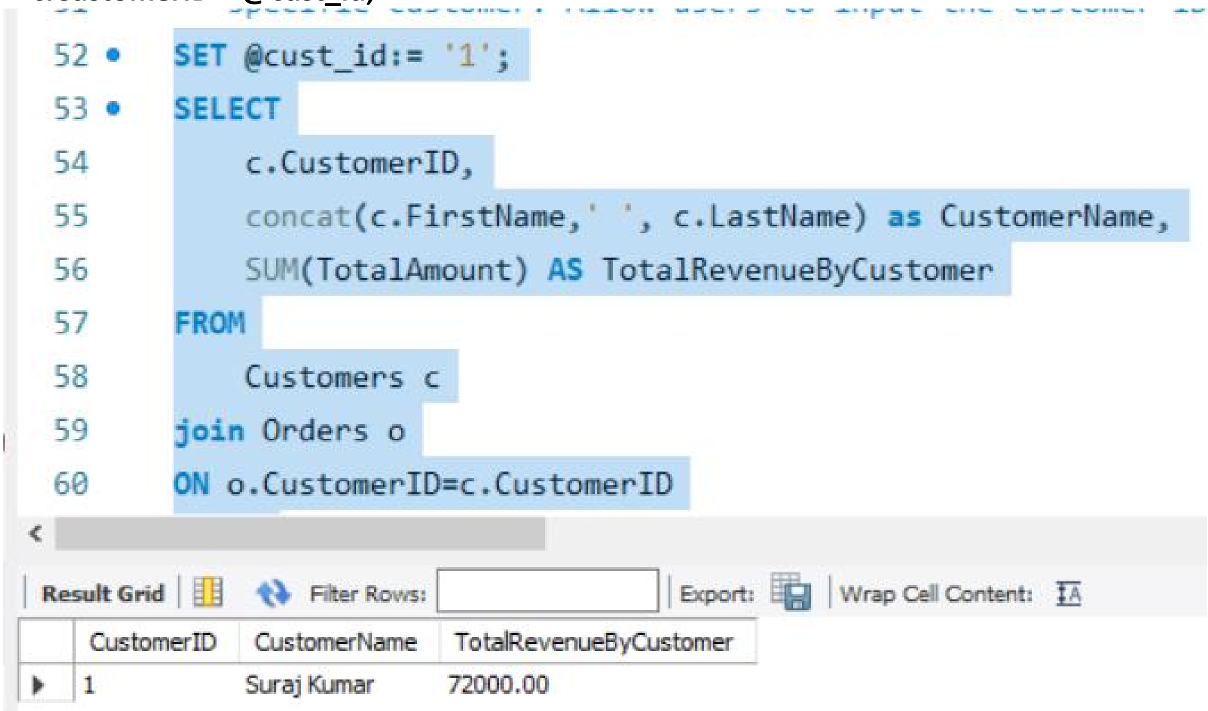


```
SET @pname:= 'Laptop';
SELECT p.ProductName, AVG(od.Quantity) AS AverageQuantityOrdered
FROM OrderDetails od
JOIN Products p ON od.ProductID = p.ProductID
WHERE p.ProductName = @pname
GROUP BY p.ProductName;
```

ProductName	AverageQuantityOrdered
Laptop	2.0000

5. Write an SQL query to calculate the total revenue generated by a specific customer. Allow users to input the customer ID as a parameter.

```
SET @cust_id:= '1';
SELECT
   c.CustomerID,
   concat(c.FirstName,' ', c.LastName) as CustomerName,
   SUM(TotalAmount) AS TotalRevenueByCustomer
FROM
   Customers c
join Orders o
ON o.CustomerID=c.CustomerID
WHERE
   c.CustomerID = @cust_id;
```

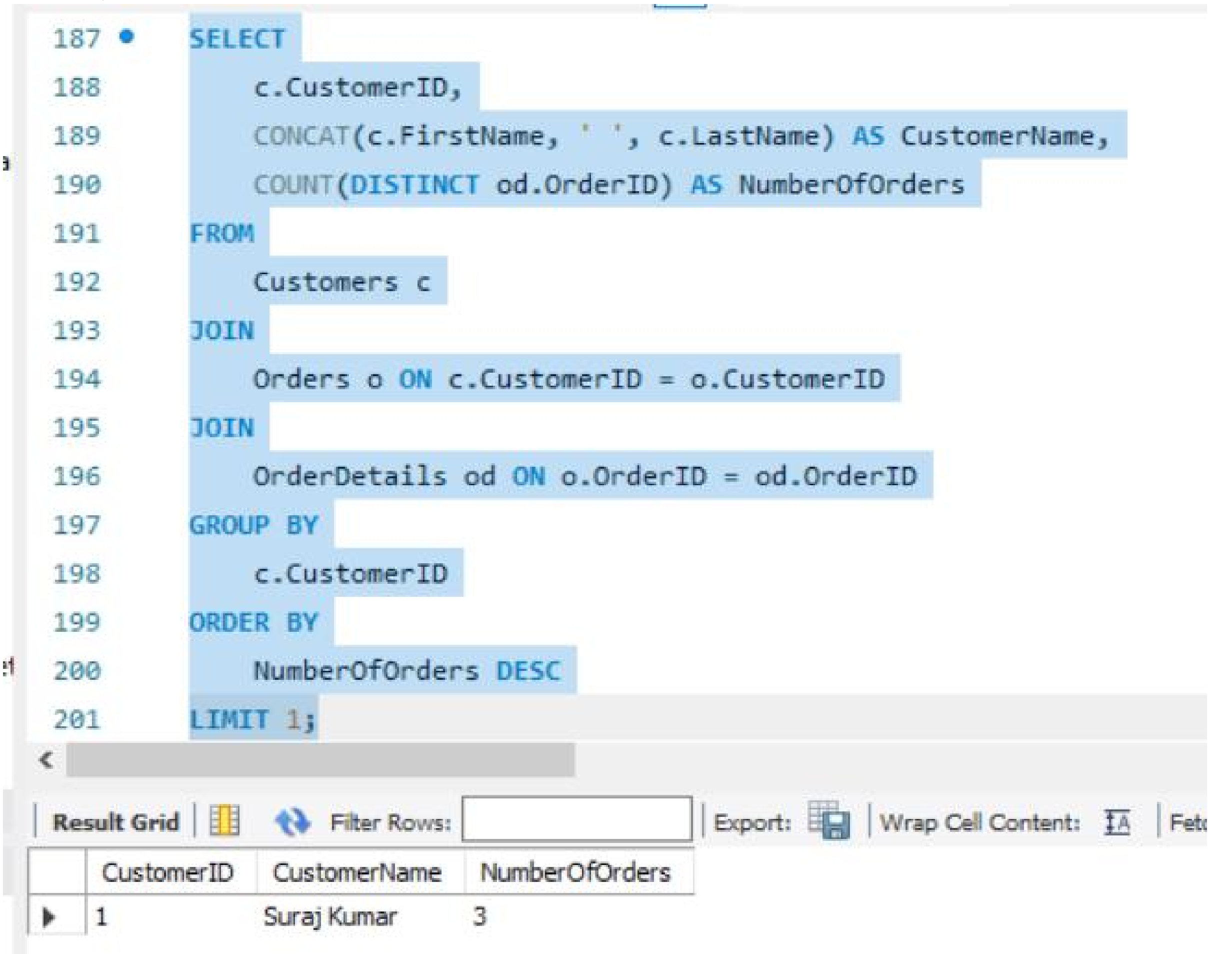


```
SET @cust_id:= '1';
SELECT
    c.CustomerID,
    concat(c.FirstName,' ', c.LastName) as CustomerName,
    SUM(TotalAmount) AS TotalRevenueByCustomer
FROM
    Customers c
join Orders o
ON o.CustomerID=c.CustomerID
```

CustomerID	CustomerName	TotalRevenueByCustomer
1	Suraj Kumar	72000.00

6. Write an SQL query to find the customers who have placed the most orders. List their names and the number of orders they've placed.

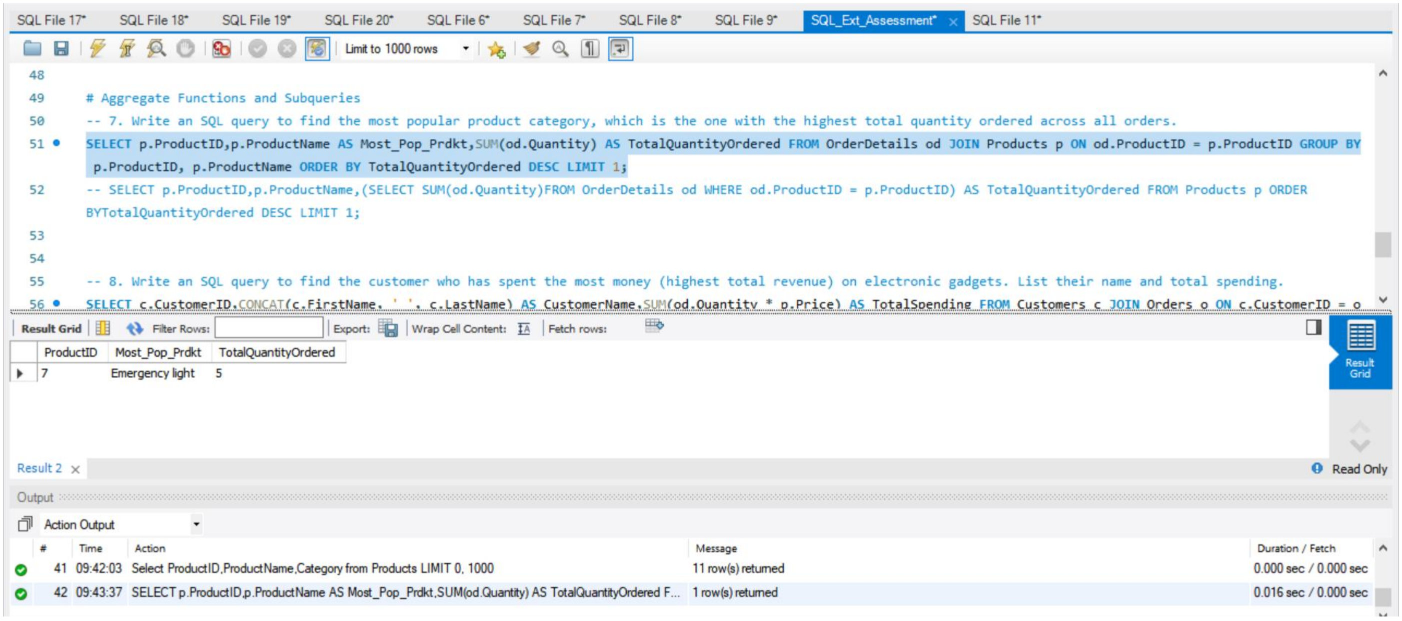
```
SELECT
   c.CustomerID,
   CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName,
   COUNT(DISTINCT od.OrderID) AS NumberOfOrders
FROM
   Customers c
JOIN
   Orders o ON c.CustomerID = o.CustomerID
JOIN
   OrderDetails od ON o.OrderID = od.OrderID
GROUP BY
   c.CustomerID
ORDER BY
   NumberOfOrders DESC
LIMIT 1;
```



```
SELECT
    c.CustomerID,
    CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName,
    COUNT(DISTINCT od.OrderID) AS NumberOfOrders
FROM
    Customers c
JOIN
    Orders o ON c.CustomerID = o.CustomerID
JOIN
    OrderDetails od ON o.OrderID = od.OrderID
GROUP BY
    c.CustomerID
ORDER BY
    NumberOfOrders DESC
LIMIT 1;
```

	CustomerID	CustomerName	NumberOfOrders
▶	1	Suraj Kumar	3

7. Write an SQL query to find the most popular product category, which is the one with the highest total quantity ordered across all orders.

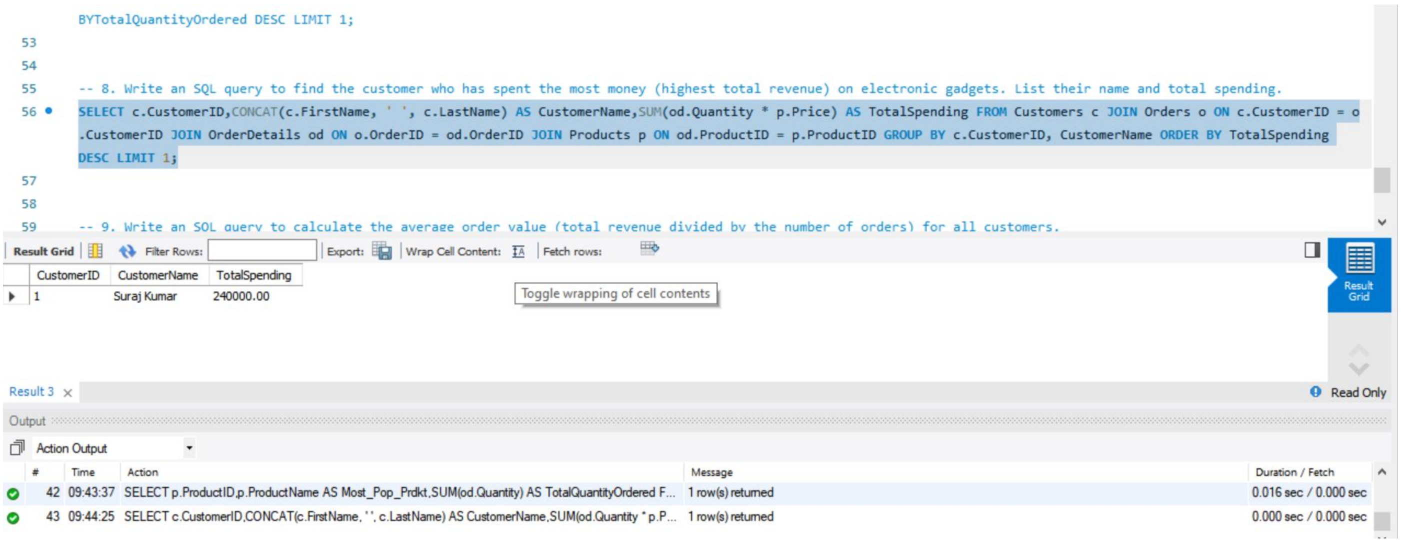


```
# Aggregate Functions and Subqueries
-- 7. Write an SQL query to find the most popular product category, which is the one with the highest total quantity ordered across all orders.
SELECT p.ProductID,p.ProductName AS Most_Pop_Prdkt,SUM(od.Quantity) AS TotalQuantityOrdered FROM OrderDetails od JOIN Products p ON od.ProductID = p.ProductID GROUP BY
 p.ProductID, p.ProductName ORDER BY TotalQuantityOrdered DESC LIMIT 1;
-- SELECT p.ProductID,p.ProductName,(SELECT SUM(od.Quantity)FROM OrderDetails od WHERE od.ProductID = p.ProductID) AS TotalQuantityOrdered FROM Products p ORDER
BYTotalQuantityOrdered DESC LIMIT 1;


-- 8. Write 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 c.CustomerID,CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName,SUM(od.Quantity * p.Price) AS TotalSpending FROM Customers c JOIN Orders o ON c.CustomerID = o
```

ProductID	Most_Pop_Prdkt	TotalQuantityOrdered
7	Emergency light	5

```
#   Time      Action                                                                     Message             Duration / Fetch
41  09:42:03  Select ProductID,ProductName,Category from Products LIMIT 0, 1000          11 row(s) returned  0.000 sec / 0.000 sec
42  09:43:37  SELECT p.ProductID,p.ProductName AS Most_Pop_Prdkt,SUM(od.Quantity) AS TotalQuantityOrdered F...  1 row(s) returned  0.016 sec / 0.000 sec
```

8. Write 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.

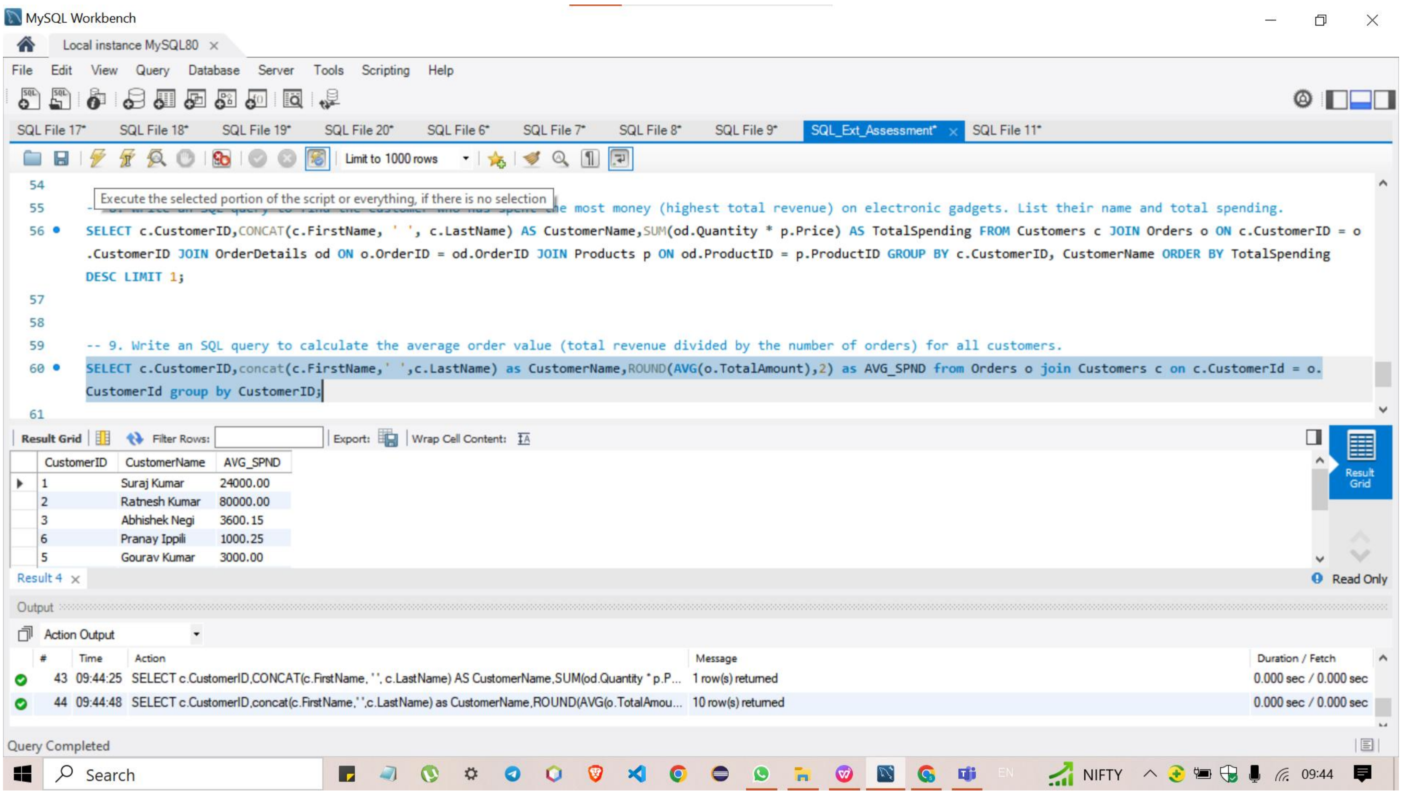


```
BYTotalQuantityOrdered DESC LIMIT 1;


-- 8. Write 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 c.CustomerID,CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName,SUM(od.Quantity * p.Price) AS TotalSpending FROM Customers c JOIN Orders o ON c.CustomerID = o
.CustomerID JOIN OrderDetails od ON o.OrderID = od.OrderID JOIN Products p ON od.ProductID = p.ProductID GROUP BY c.CustomerID, CustomerName ORDER BY TotalSpending
DESC LIMIT 1;


-- 9. Write an SQL query to calculate the average order value (total revenue divided by the number of orders) for all customers.
```

CustomerID	CustomerName	TotalSpending
1	Suraj Kumar	240000.00

```
#   Time      Action                                                                     Message             Duration / Fetch
42  09:43:37  SELECT p.ProductID,p.ProductName AS Most_Pop_Prdkt,SUM(od.Quantity) AS TotalQuantityOrdered F...  1 row(s) returned  0.016 sec / 0.000 sec
43  09:44:25  SELECT c.CustomerID,CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName,SUM(od.Quantity * p.P...  1 row(s) returned  0.000 sec / 0.000 sec
```

9. Write an SQL query to calculate the average order value (total revenue divided by the number of orders) for all customers.



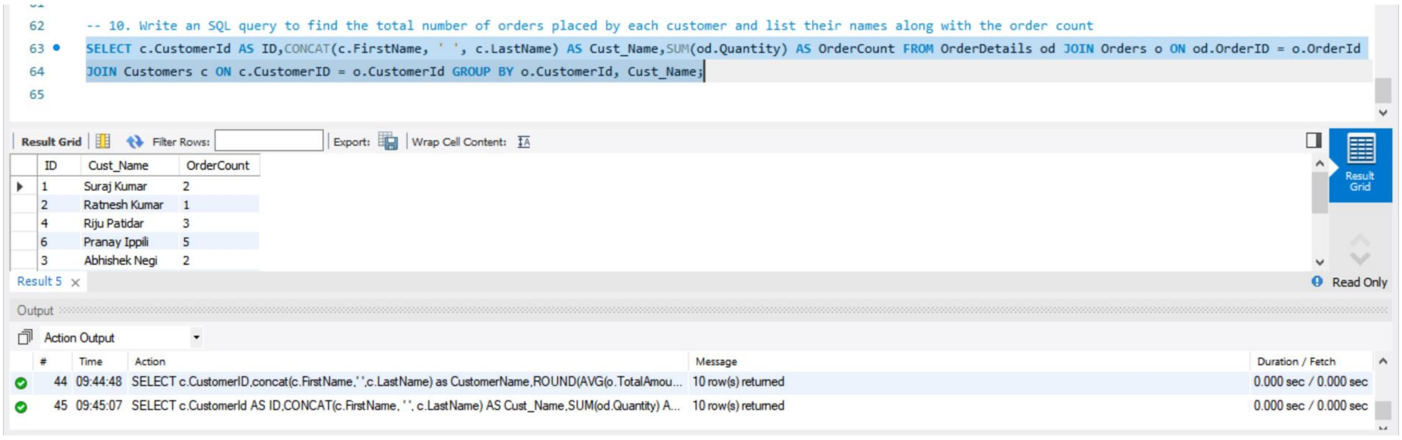
MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 17* SQL File 18* SQL File 19* SQL File 20* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL_Ext_Assessment* SQL File 11*
Limit to 1000 rows
Execute the selected portion of the script or everything, if there is no selection

```sql
-- 8. Write 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 c.CustomerID,CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName,SUM(od.Quantity * p.Price) AS TotalSpending FROM Customers c JOIN Orders o ON c.CustomerID = o
.CustomerID JOIN OrderDetails od ON o.OrderID = od.OrderID JOIN Products p ON od.ProductID = p.ProductID GROUP BY c.CustomerID, CustomerName ORDER BY TotalSpending
DESC LIMIT 1;


-- 9. Write an SQL query to calculate the average order value (total revenue divided by the number of orders) for all customers.
SELECT c.CustomerID,concat(c.FirstName,' ',c.LastName) as CustomerName,ROUND(AVG(o.TotalAmount),2) as AVG_SPND from Orders o join Customers c on c.CustomerId = o.
CustomerId group by CustomerID;
```

CustomerID	CustomerName	AVG_SPND
1	Suraj Kumar	24000.00
2	Ratnesh Kumar	80000.00
3	Abhishek Negi	3600.15
6	Pranay Ippili	1000.25
5	Gourav Kumar	3000.00

#	Time	Action	Message	Duration / Fetch
43	09:44:25	SELECT c.CustomerID,CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName,SUM(od.Quantity * p.P...	1 row(s) returned	0.000 sec / 0.000 sec
44	09:44:48	SELECT c.CustomerID,concat(c.FirstName,' ',c.LastName) as CustomerName,ROUND(AVG(o.TotalAmou...	10 row(s) returned	0.000 sec / 0.000 sec

10. Write an SQL query to find the total number of orders placed by each customer and list their names along with the order count.



```sql
-- 10. Write an 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 AS ID,CONCAT(c.FirstName, ' ', c.LastName) AS Cust_Name,SUM(od.Quantity) AS OrderCount FROM OrderDetails od JOIN Orders o ON od.OrderID = o.OrderId
JOIN Customers c ON c.CustomerID = o.CustomerId GROUP BY o.CustomerId, Cust_Name;
```

ID	Cust_Name	OrderCount
1	Suraj Kumar	2
2	Ratnesh Kumar	1
4	Riju Patidar	3
6	Pranay Ippili	5
3	Abhishek Negi	2

#	Time	Action	Message	Duration / Fetch
44	09:44:48	SELECT c.CustomerID,concat(c.FirstName,' ',c.LastName) as CustomerName,ROUND(AVG(o.TotalAmou...	10 row(s) returned	0.000 sec / 0.000 sec
45	09:45:07	SELECT c.CustomerId AS ID,CONCAT(c.FirstName, ' ', c.LastName) AS Cust_Name,SUM(od.Quantity) A...	10 row(s) returned	0.000 sec / 0.000 sec

Thank You