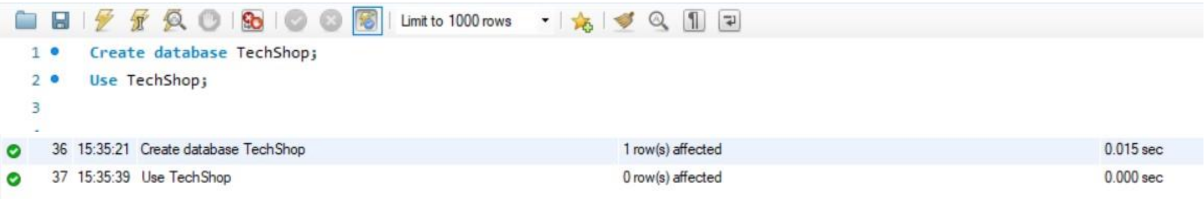
Assignment 1

Topic: TechShop, an electronic gadgets shop

Submitted By: Vrushali Tekchand Rahangdale **Email ID**: vrushutr08@gmail.com

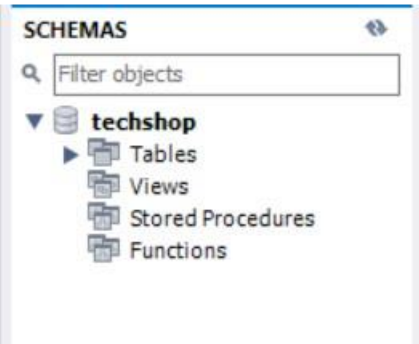
Task:1. Database Design:

1. Create the database named "TechShop"



```
Create database TechShop;
Use TechShop;
```

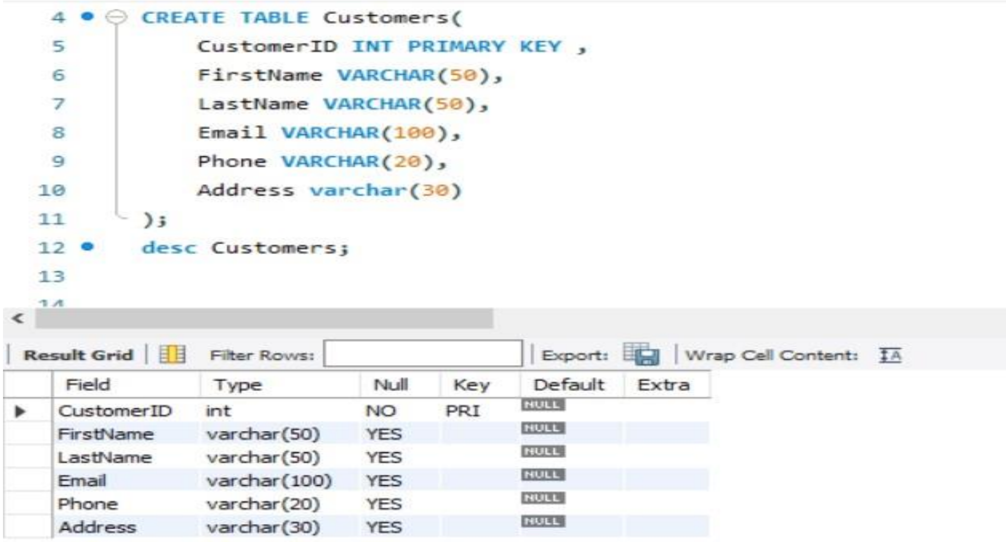
```
36  15:35:21  Create database TechShop    1 row(s) affected    0.015 sec
37  15:35:39  Use TechShop                0 row(s) affected    0.000 sec
```



2. Define the schema for the Customers, Products, Orders, OrderDetails and Inventory tables based on the provided schema.
3. Create appropriate Primary Key and Foreign Key constraints for referential integrity.

1. Customers:

- CustomerID (Primary Key)
- FirstName
- LastName
- Email
- Phone
- Address

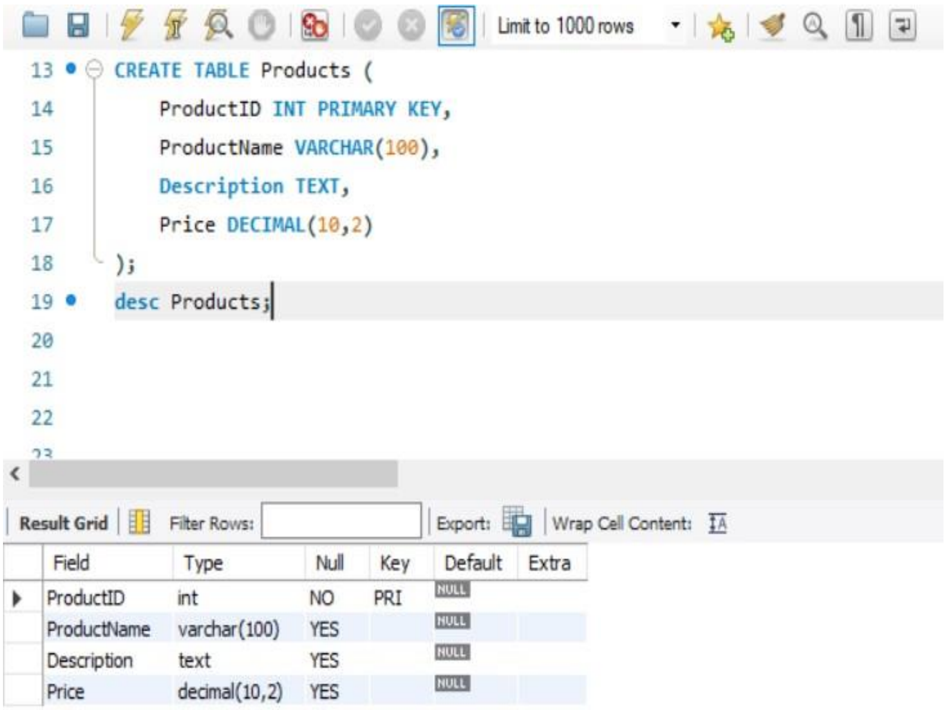


```
CREATE TABLE Customers(
        CustomerID INT PRIMARY KEY ,
        FirstName VARCHAR(50),
        LastName VARCHAR(50),
        Email VARCHAR(100),
        Phone VARCHAR(20),
        Address varchar(30)
);
desc Customers;
```

Field	Type	Null	Key	Default	Extra
CustomerID	int	NO	PRI	NULL	
FirstName	varchar(50)	YES		NULL	
LastName	varchar(50)	YES		NULL	
Email	varchar(100)	YES		NULL	
Phone	varchar(20)	YES		NULL	
Address	varchar(30)	YES		NULL	

2. Products:

- ProductID (Primary Key)
- ProductName
- Description
- Price

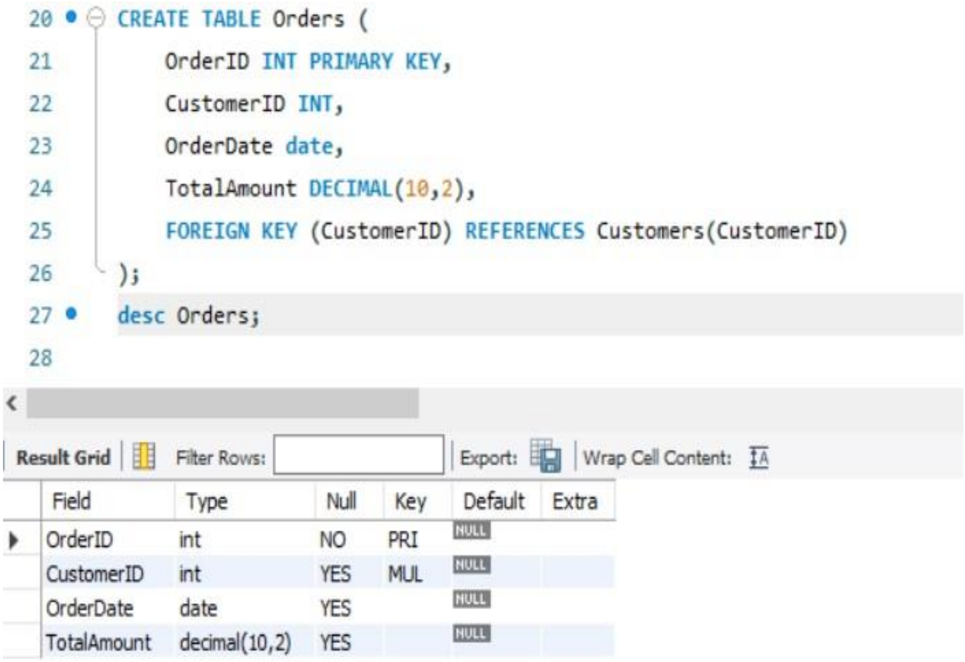


```sql
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(100),
    Description TEXT,
    Price DECIMAL(10,2)
);
desc Products;
```

Field	Type	Null	Key	Default	Extra
ProductID	int	NO	PRI	NULL	
ProductName	varchar(100)	YES		NULL	
Description	text	YES		NULL	
Price	decimal(10,2)	YES		NULL	

3. Orders:

- OrderID (Primary Key)
- CustomerID (Foreign Key referencing Customers)
- OrderDate
- TotalAmount

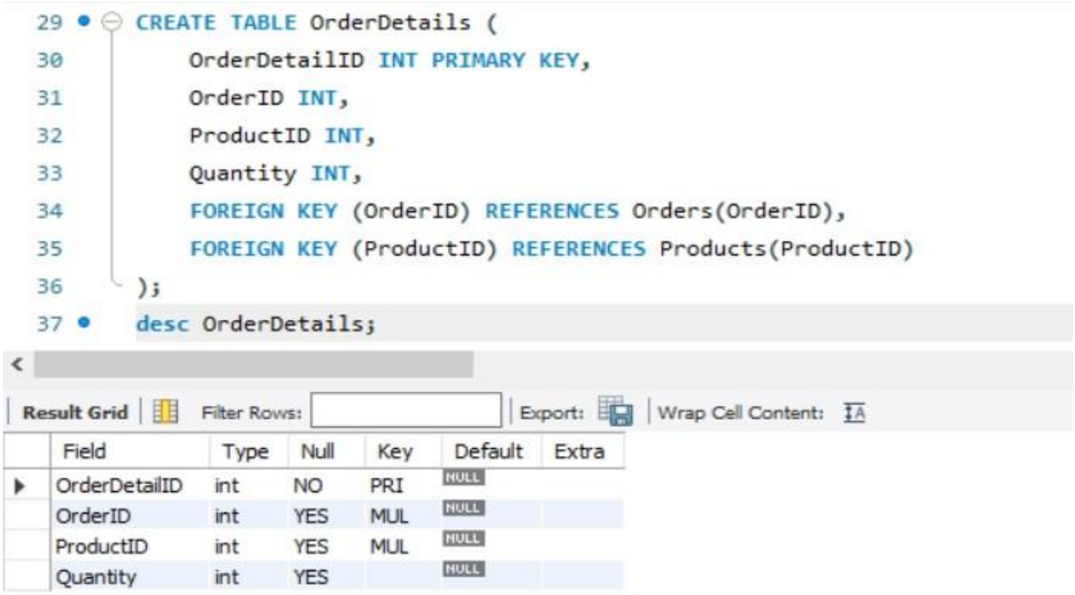


```sql
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate date,
    TotalAmount DECIMAL(10,2),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
desc Orders;
```

Field	Type	Null	Key	Default	Extra
OrderID	int	NO	PRI	NULL	
CustomerID	int	YES	MUL	NULL	
OrderDate	date	YES		NULL	
TotalAmount	decimal(10,2)	YES		NULL	

4. OrderDetails:

- OrderDetailID (Primary Key)
- OrderID (Foreign Key referencing Orders)
- ProductID (Foreign Key referencing Products)
- Quantity

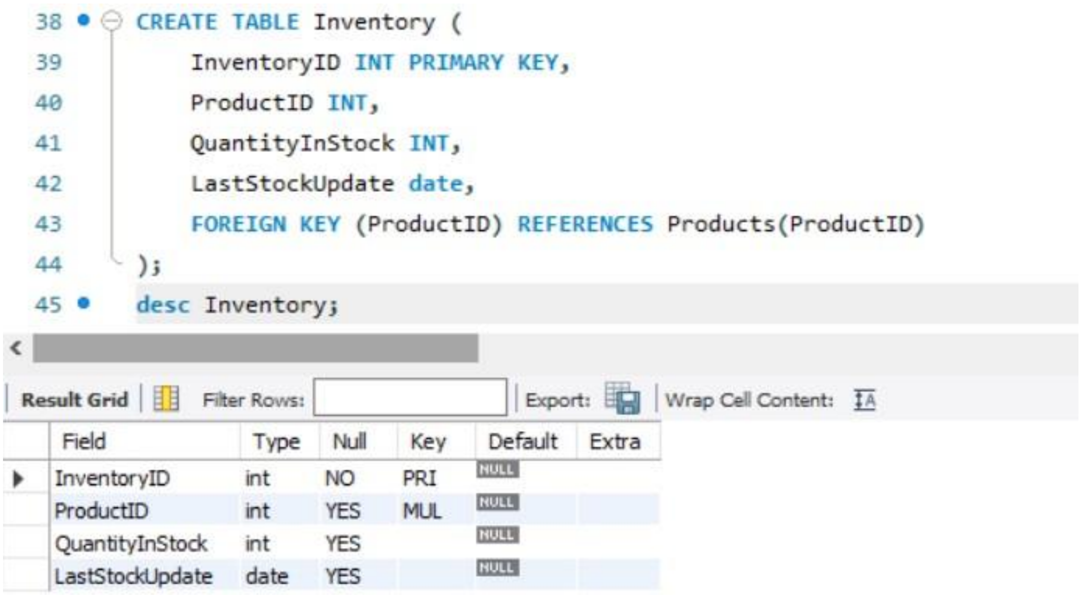


```sql
CREATE TABLE OrderDetails (
    OrderDetailID INT PRIMARY KEY,
    OrderID INT,
    ProductID INT,
    Quantity INT,
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
desc OrderDetails;
```

Field	Type	Null	Key	Default	Extra
OrderDetailID	int	NO	PRI	NULL	
OrderID	int	YES	MUL	NULL	
ProductID	int	YES	MUL	NULL	
Quantity	int	YES		NULL	

5. Inventory

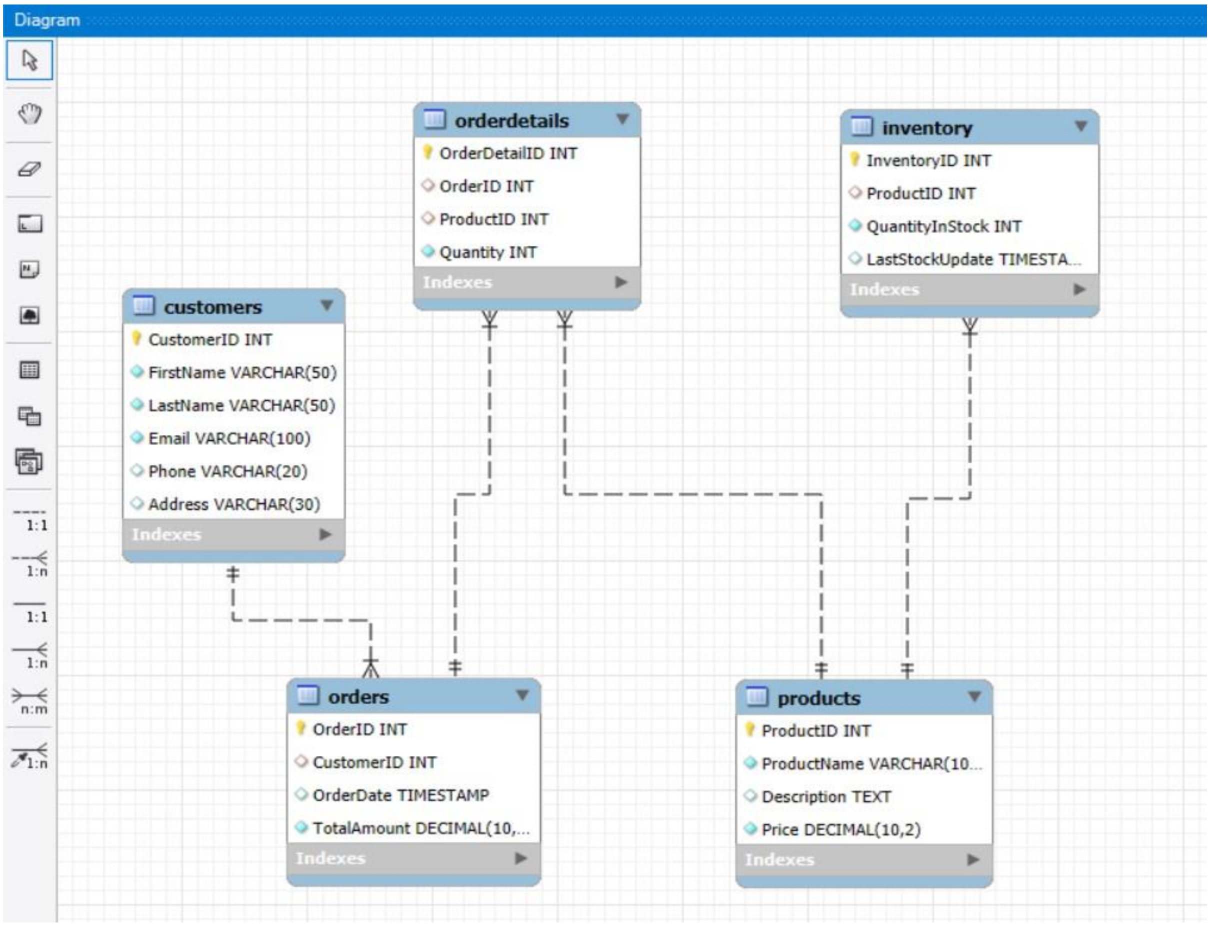
- InventoryID (Primary Key)
- ProductID (Foreign Key referencing Products)
- QuantityInStock
- LastStockUpdate



```sql
CREATE TABLE Inventory (
    InventoryID INT PRIMARY KEY,
    ProductID INT,
    QuantityInStock INT,
    LastStockUpdate date,
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
desc Inventory;
```

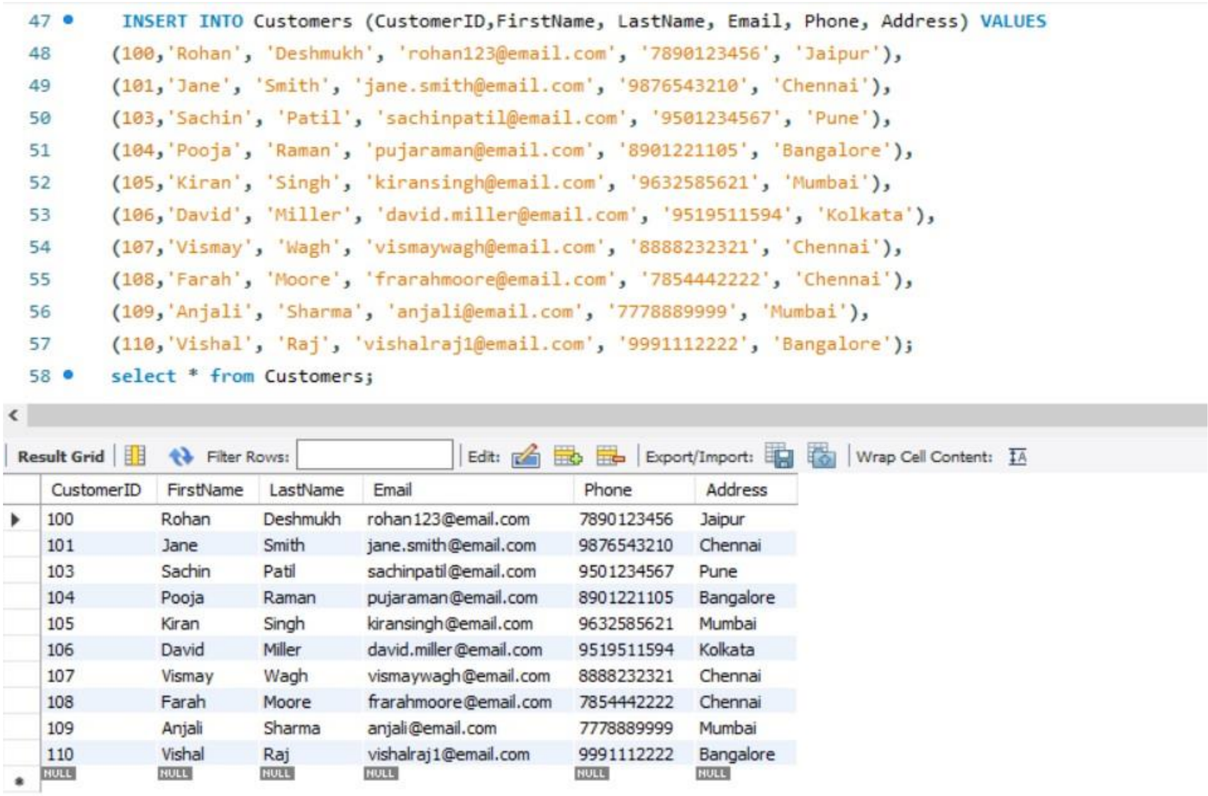
Field	Type	Null	Key	Default	Extra
InventoryID	int	NO	PRI	NULL	
ProductID	int	YES	MUL	NULL	
QuantityInStock	int	YES		NULL	
LastStockUpdate	date	YES		NULL	

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



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

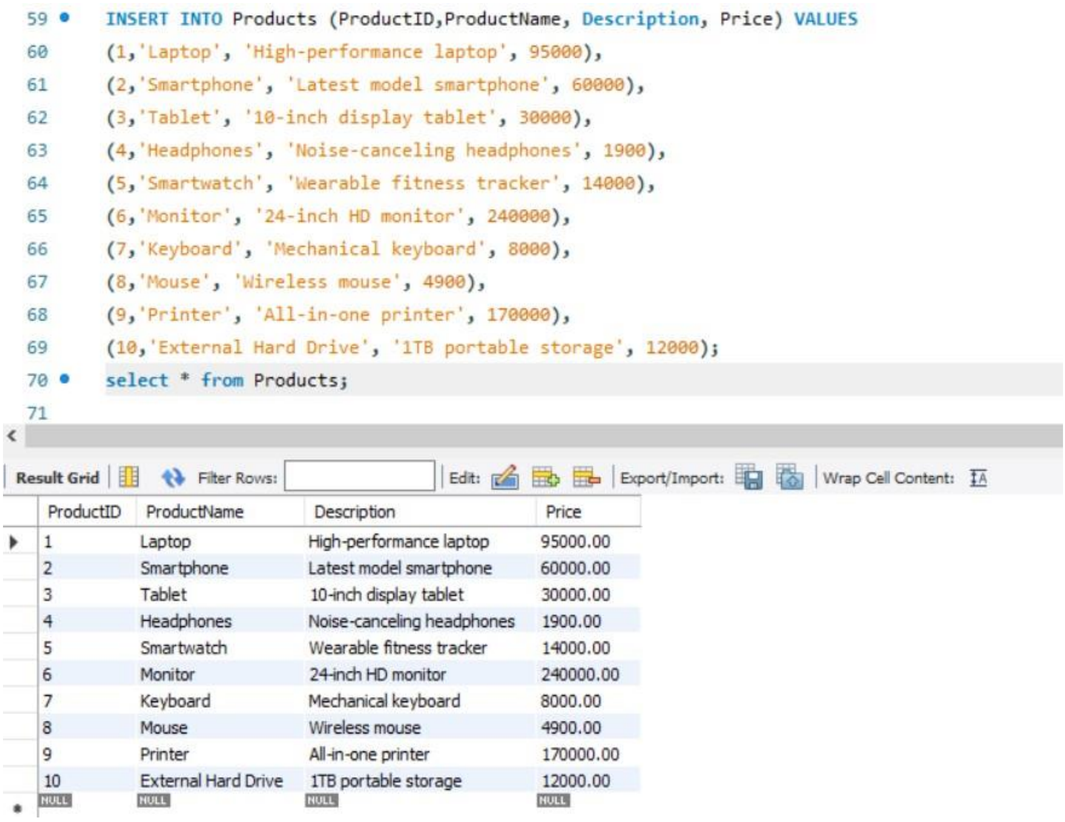
a. Customers Table



```
 INSERT INTO Customers (CustomerID,FirstName, LastName, Email, Phone, Address) VALUES
(100,'Rohan', 'Deshmukh', 'rohan123@email.com', '7890123456', 'Jaipur'),
(101,'Jane', 'Smith', 'jane.smith@email.com', '9876543210', 'Chennai'),
(103,'Sachin', 'Patil', 'sachinpatil@email.com', '9501234567', 'Pune'),
(104,'Pooja', 'Raman', 'pujaraman@email.com', '8901221105', 'Bangalore'),
(105,'Kiran', 'Singh', 'kiransingh@email.com', '9632585621', 'Mumbai'),
(106,'David', 'Miller', 'david.miller@email.com', '9519511594', 'Kolkata'),
(107,'Vismay', 'Wagh', 'vismaywagh@email.com', '8888232321', 'Chennai'),
(108,'Farah', 'Moore', 'frarahmoore@email.com', '7854442222', 'Chennai'),
(109,'Anjali', 'Sharma', 'anjali@email.com', '7778889999', 'Mumbai'),
(110,'Vishal', 'Raj', 'vishalraj1@email.com', '9991112222', 'Bangalore');
select * from Customers;
```

CustomerID	FirstName	LastName	Email	Phone	Address
100	Rohan	Deshmukh	rohan123@email.com	7890123456	Jaipur
101	Jane	Smith	jane.smith@email.com	9876543210	Chennai
103	Sachin	Patil	sachinpatil@email.com	9501234567	Pune
104	Pooja	Raman	pujaraman@email.com	8901221105	Bangalore
105	Kiran	Singh	kiransingh@email.com	9632585621	Mumbai
106	David	Miller	david.miller@email.com	9519511594	Kolkata
107	Vismay	Wagh	vismaywagh@email.com	8888232321	Chennai
108	Farah	Moore	frarahmoore@email.com	7854442222	Chennai
109	Anjali	Sharma	anjali@email.com	7778889999	Mumbai
110	Vishal	Raj	vishalraj1@email.com	9991112222	Bangalore
NULL	NULL	NULL	NULL	NULL	NULL

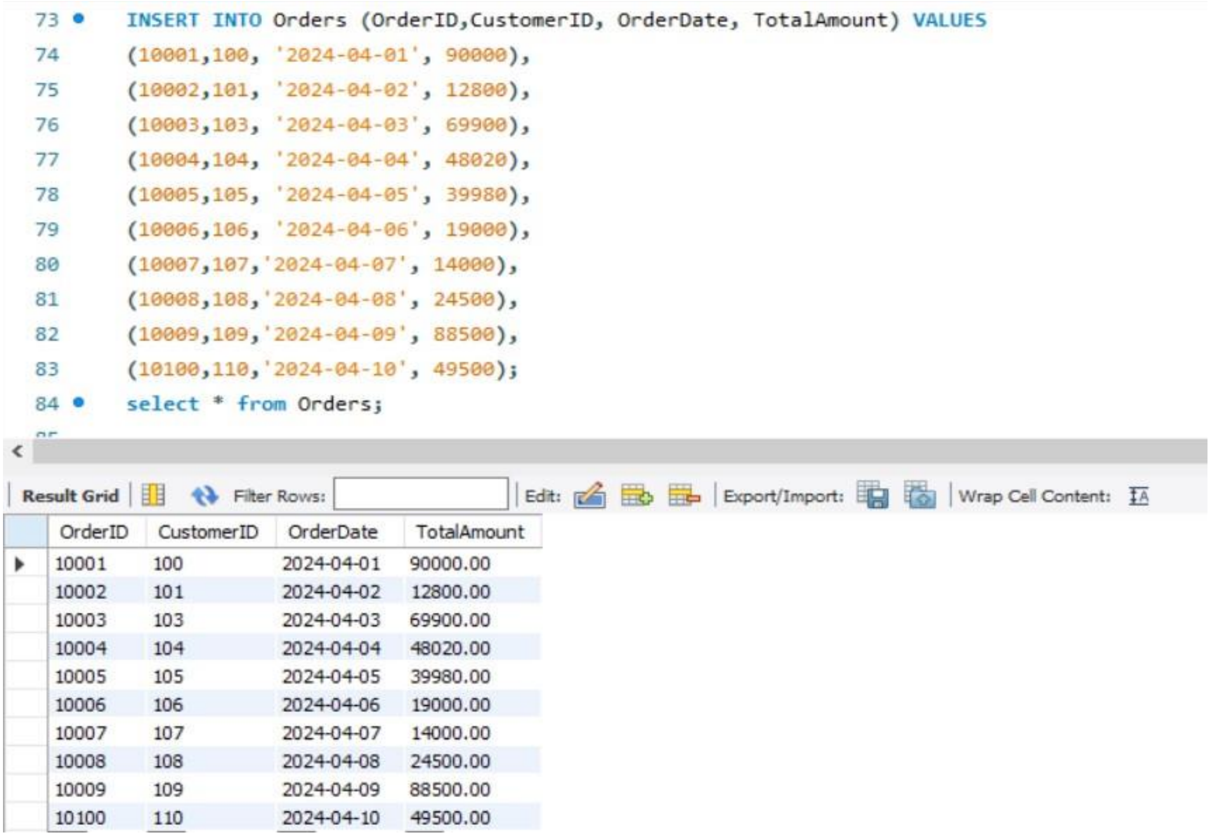
b.Product Table



```sql
INSERT INTO Products (ProductID,ProductName, Description, Price) VALUES
(1,'Laptop', 'High-performance laptop', 95000),
(2,'Smartphone', 'Latest model smartphone', 60000),
(3,'Tablet', '10-inch display tablet', 30000),
(4,'Headphones', 'Noise-canceling headphones', 1900),
(5,'Smartwatch', 'Wearable fitness tracker', 14000),
(6,'Monitor', '24-inch HD monitor', 240000),
(7,'Keyboard', 'Mechanical keyboard', 8000),
(8,'Mouse', 'Wireless mouse', 4900),
(9,'Printer', 'All-in-one printer', 170000),
(10,'External Hard Drive', '1TB portable storage', 12000);
select * from Products;
```

ProductID	ProductName	Description	Price
1	Laptop	High-performance laptop	95000.00
2	Smartphone	Latest model smartphone	60000.00
3	Tablet	10-inch display tablet	30000.00
4	Headphones	Noise-canceling headphones	1900.00
5	Smartwatch	Wearable fitness tracker	14000.00
6	Monitor	24-inch HD monitor	240000.00
7	Keyboard	Mechanical keyboard	8000.00
8	Mouse	Wireless mouse	4900.00
9	Printer	All-in-one printer	170000.00
10	External Hard Drive	1TB portable storage	12000.00
NULL	NULL	NULL	NULL

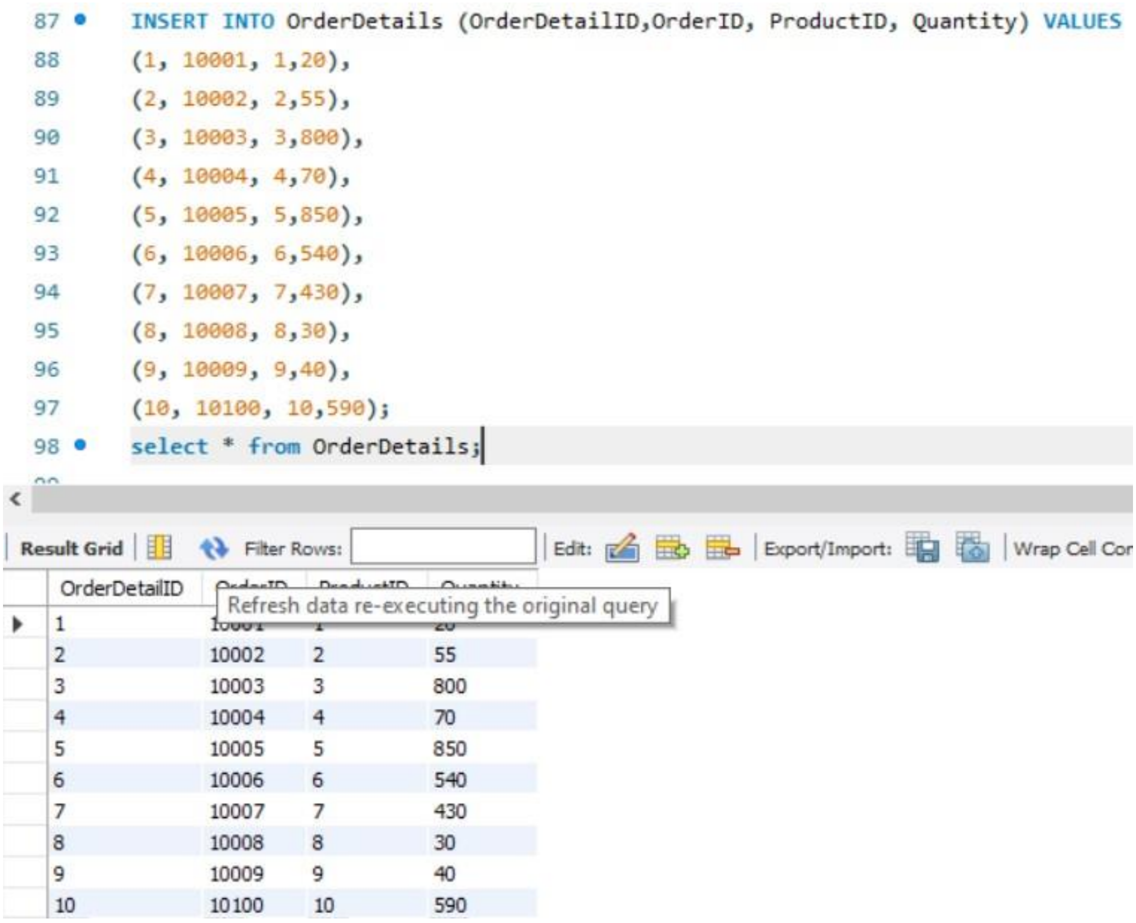
c.Orders Table



```sql
INSERT INTO Orders (OrderID,CustomerID, OrderDate, TotalAmount) VALUES
(10001,100, '2024-04-01', 90000),
(10002,101, '2024-04-02', 12800),
(10003,103, '2024-04-03', 69900),
(10004,104, '2024-04-04', 48020),
(10005,105, '2024-04-05', 39980),
(10006,106, '2024-04-06', 19000),
(10007,107,'2024-04-07', 14000),
(10008,108,'2024-04-08', 24500),
(10009,109,'2024-04-09', 88500),
(10100,110,'2024-04-10', 49500);
select * from Orders;
```

OrderID	CustomerID	OrderDate	TotalAmount
10001	100	2024-04-01	90000.00
10002	101	2024-04-02	12800.00
10003	103	2024-04-03	69900.00
10004	104	2024-04-04	48020.00
10005	105	2024-04-05	39980.00
10006	106	2024-04-06	19000.00
10007	107	2024-04-07	14000.00
10008	108	2024-04-08	24500.00
10009	109	2024-04-09	88500.00
10100	110	2024-04-10	49500.00

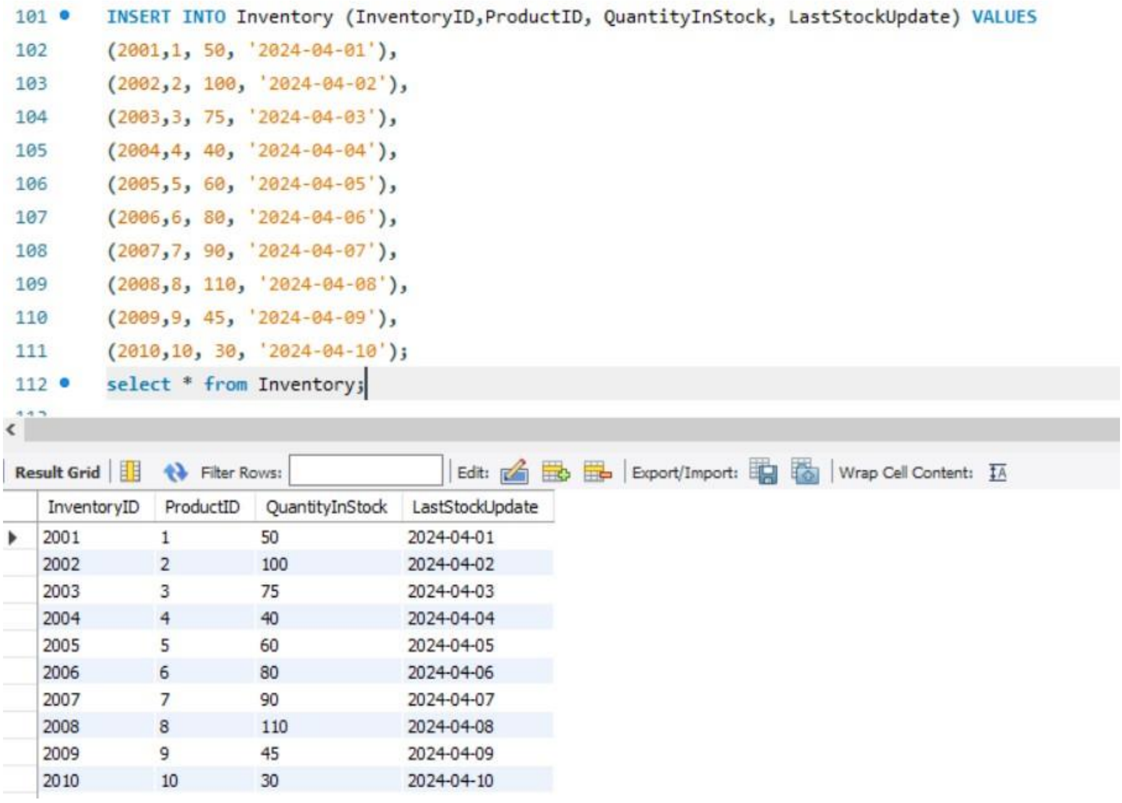
d. OrderDetails



```sql
INSERT INTO OrderDetails (OrderDetailID,OrderID, ProductID, Quantity) VALUES
(1, 10001, 1,20),
(2, 10002, 2,55),
(3, 10003, 3,800),
(4, 10004, 4,70),
(5, 10005, 5,850),
(6, 10006, 6,540),
(7, 10007, 7,430),
(8, 10008, 8,30),
(9, 10009, 9,40),
(10, 10100, 10,590);
select * from OrderDetails;
```

OrderDetailID	OrderID	ProductID	Quantity
1	10001	1	20
2	10002	2	55
3	10003	3	800
4	10004	4	70
5	10005	5	850
6	10006	6	540
7	10007	7	430
8	10008	8	30
9	10009	9	40
10	10100	10	590

e. Inventory

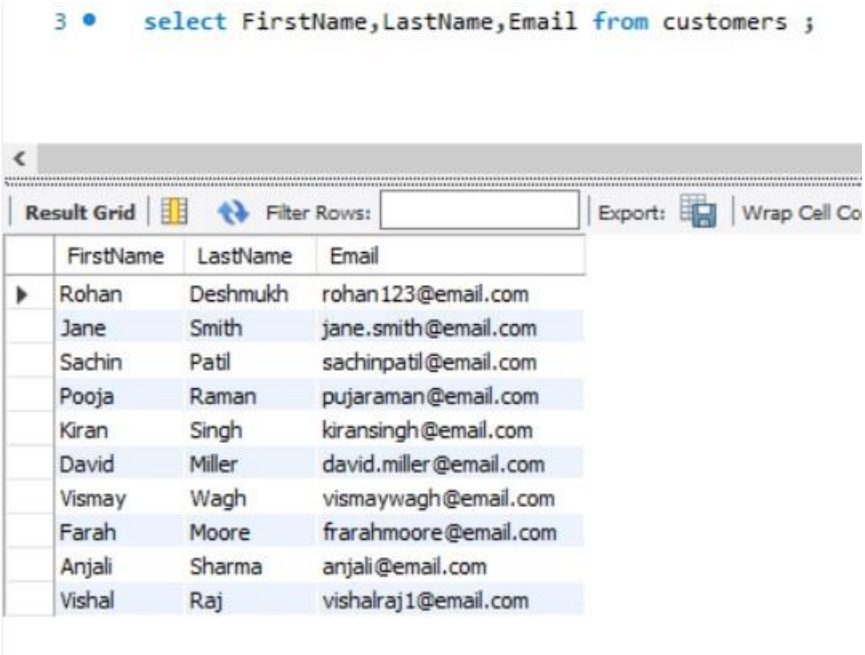


```sql
INSERT INTO Inventory (InventoryID,ProductID, QuantityInStock, LastStockUpdate) VALUES
(2001,1, 50, '2024-04-01'),
(2002,2, 100, '2024-04-02'),
(2003,3, 75, '2024-04-03'),
(2004,4, 40, '2024-04-04'),
(2005,5, 60, '2024-04-05'),
(2006,6, 80, '2024-04-06'),
(2007,7, 90, '2024-04-07'),
(2008,8, 110, '2024-04-08'),
(2009,9, 45, '2024-04-09'),
(2010,10, 30, '2024-04-10');
select * from Inventory;
```

InventoryID	ProductID	QuantityInStock	LastStockUpdate
2001	1	50	2024-04-01
2002	2	100	2024-04-02
2003	3	75	2024-04-03
2004	4	40	2024-04-04
2005	5	60	2024-04-05
2006	6	80	2024-04-06
2007	7	90	2024-04-07
2008	8	110	2024-04-08
2009	9	45	2024-04-09
2010	10	30	2024-04-10

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

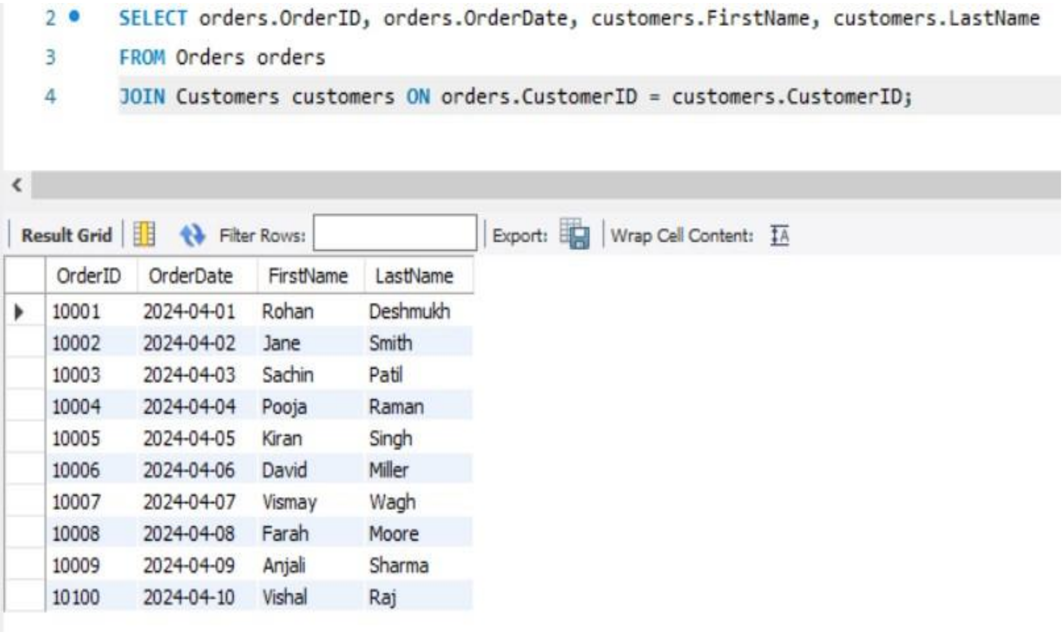
1. Write an SQL query to retrieve the names and emails of all customers.



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

FirstName	LastName	Email
Rohan	Deshmukh	rohan123@email.com
Jane	Smith	jane.smith@email.com
Sachin	Patil	sachinpatil@email.com
Pooja	Raman	pujaraman@email.com
Kiran	Singh	kiransingh@email.com
David	Miller	david.miller@email.com
Vismay	Wagh	vismaywagh@email.com
Farah	Moore	frarahmoore@email.com
Anjali	Sharma	anjali@email.com
Vishal	Raj	vishalraj1@email.com

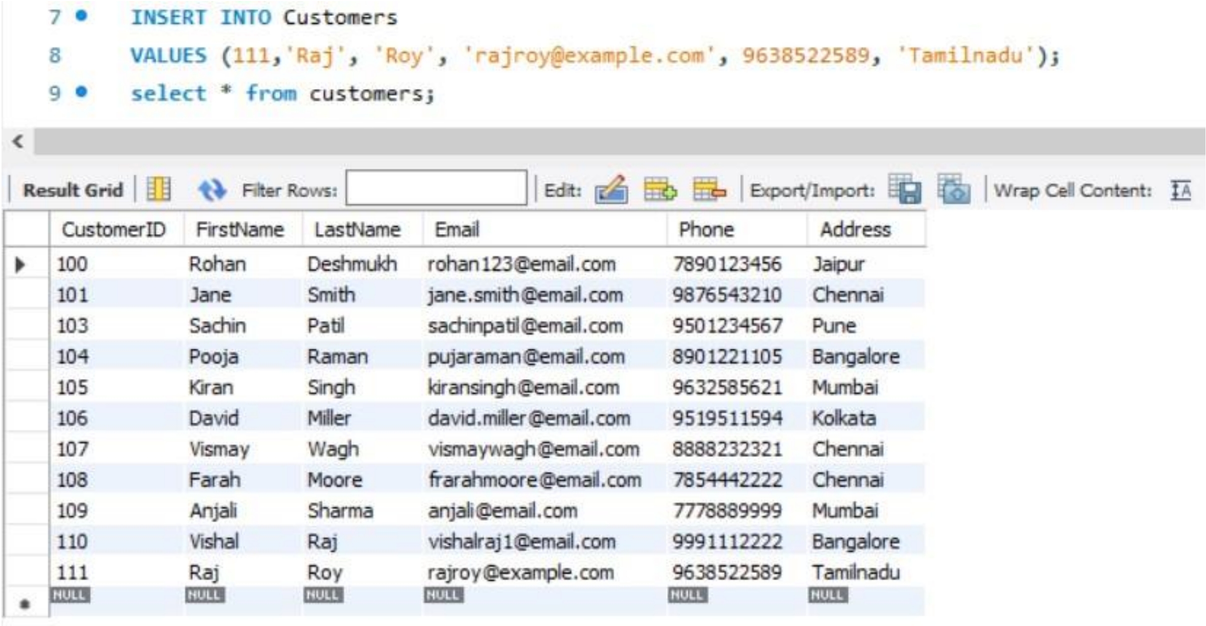
2. Write an SQL query to list all orders with their order dates and corresponding customer names.



```
SELECT orders.OrderID, orders.OrderDate, customers.FirstName, customers.LastName
FROM Orders orders
JOIN Customers customers ON orders.CustomerID = customers.CustomerID;
```

OrderID	OrderDate	FirstName	LastName
10001	2024-04-01	Rohan	Deshmukh
10002	2024-04-02	Jane	Smith
10003	2024-04-03	Sachin	Patil
10004	2024-04-04	Pooja	Raman
10005	2024-04-05	Kiran	Singh
10006	2024-04-06	David	Miller
10007	2024-04-07	Vismay	Wagh
10008	2024-04-08	Farah	Moore
10009	2024-04-09	Anjali	Sharma
10100	2024-04-10	Vishal	Raj

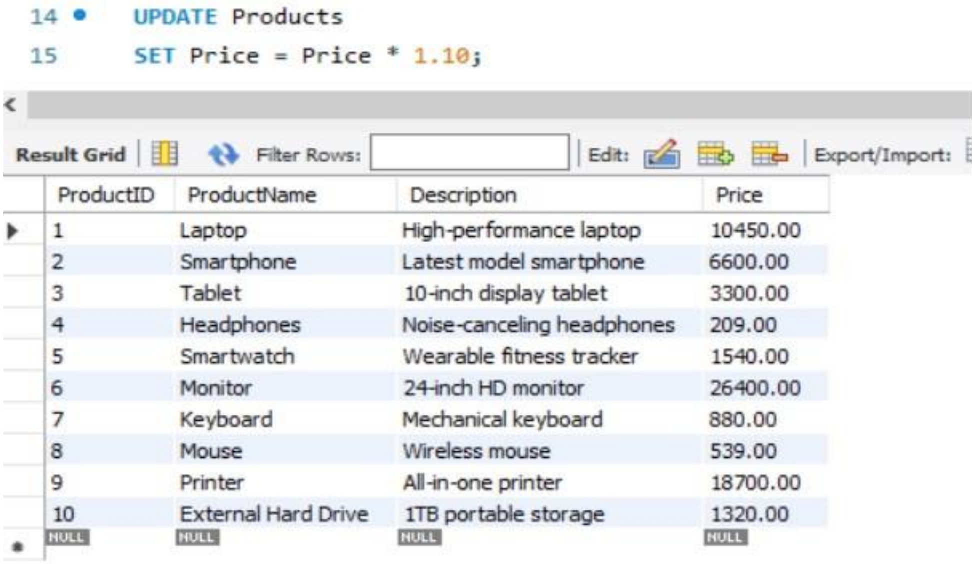
3. Write an SQL query to insert a new customer record into the "Customers" table. Include customer information such as name, email, and address.



```
INSERT INTO Customers
VALUES (111,'Raj', 'Roy', 'rajroy@example.com', 9638522589, 'Tamilnadu');
select * from customers;
```

CustomerID	FirstName	LastName	Email	Phone	Address
100	Rohan	Deshmukh	rohan123@email.com	7890123456	Jaipur
101	Jane	Smith	jane.smith@email.com	9876543210	Chennai
103	Sachin	Patil	sachinpatil@email.com	9501234567	Pune
104	Pooja	Raman	pujaraman@email.com	8901221105	Bangalore
105	Kiran	Singh	kiransingh@email.com	9632585621	Mumbai
106	David	Miller	david.miller@email.com	9519511594	Kolkata
107	Vismay	Wagh	vismaywagh@email.com	8888232321	Chennai
108	Farah	Moore	frarahmoore@email.com	7854442222	Chennai
109	Anjali	Sharma	anjali@email.com	7778889999	Mumbai
110	Vishal	Raj	vishalraj1@email.com	9991112222	Bangalore
111	Raj	Roy	rajroy@example.com	9638522589	Tamilnadu
NULL	NULL	NULL	NULL	NULL	NULL

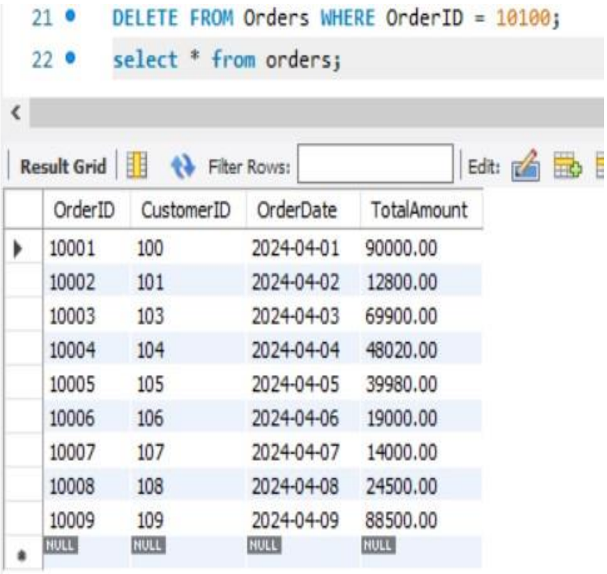
4. Write an SQL query to update the prices of all electronic gadgets in the "Products" table by increasing them by 10%.



```
UPDATE Products
SET Price = Price * 1.10;
```

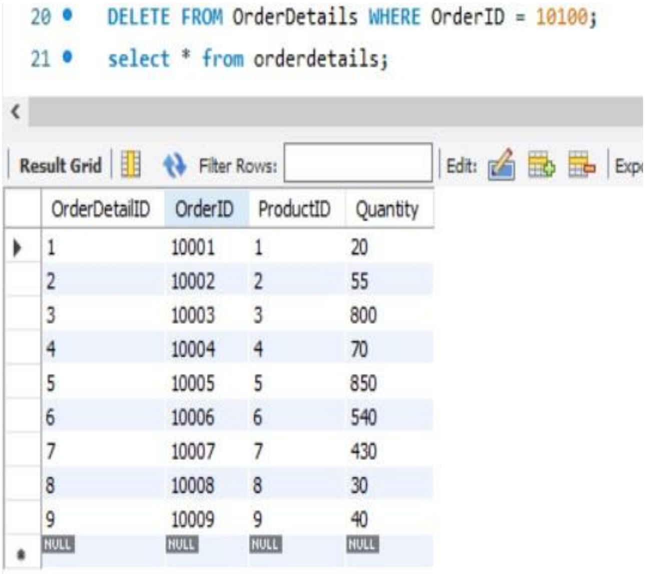
ProductID	ProductName	Description	Price
1	Laptop	High-performance laptop	10450.00
2	Smartphone	Latest model smartphone	6600.00
3	Tablet	10-inch display tablet	3300.00
4	Headphones	Noise-canceling headphones	209.00
5	Smartwatch	Wearable fitness tracker	1540.00
6	Monitor	24-inch HD monitor	26400.00
7	Keyboard	Mechanical keyboard	880.00
8	Mouse	Wireless mouse	539.00
9	Printer	All-in-one printer	18700.00
10	External Hard Drive	1TB portable storage	1320.00
NULL	NULL	NULL	NULL

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.



```sql
DELETE FROM Orders WHERE OrderID = 10100;
select * from orders;
```

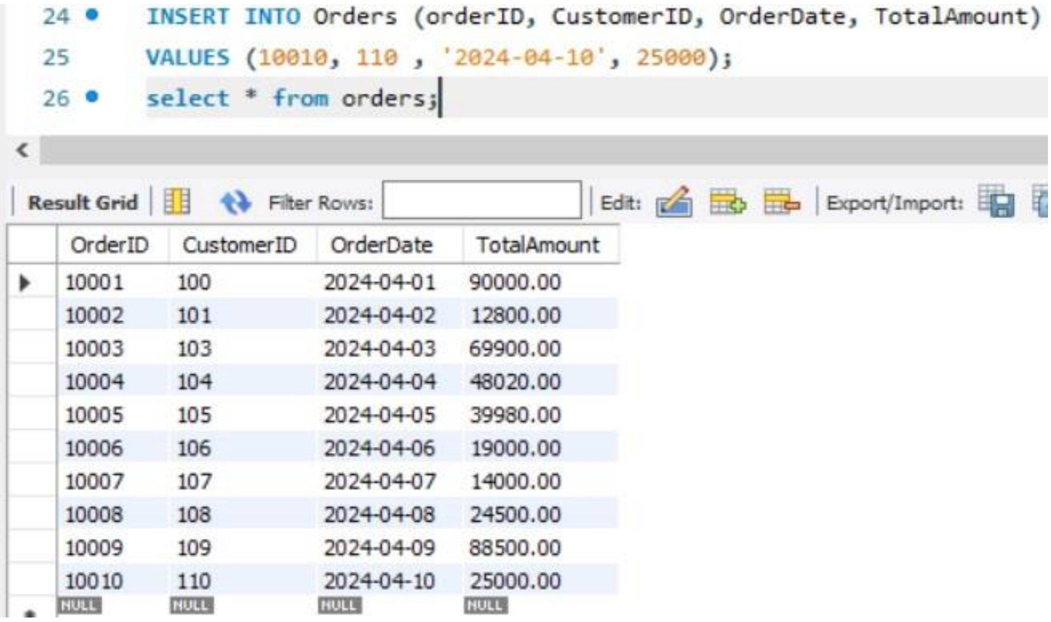
OrderID	CustomerID	OrderDate	TotalAmount
10001	100	2024-04-01	90000.00
10002	101	2024-04-02	12800.00
10003	103	2024-04-03	69900.00
10004	104	2024-04-04	48020.00
10005	105	2024-04-05	39980.00
10006	106	2024-04-06	19000.00
10007	107	2024-04-07	14000.00
10008	108	2024-04-08	24500.00
10009	109	2024-04-09	88500.00
NULL	NULL	NULL	NULL



```sql
DELETE FROM OrderDetails WHERE OrderID = 10100;
select * from orderdetails;
```

OrderDetailID	OrderID	ProductID	Quantity
1	10001	1	20
2	10002	2	55
3	10003	3	800
4	10004	4	70
5	10005	5	850
6	10006	6	540
7	10007	7	430
8	10008	8	30
9	10009	9	40
NULL	NULL	NULL	NULL

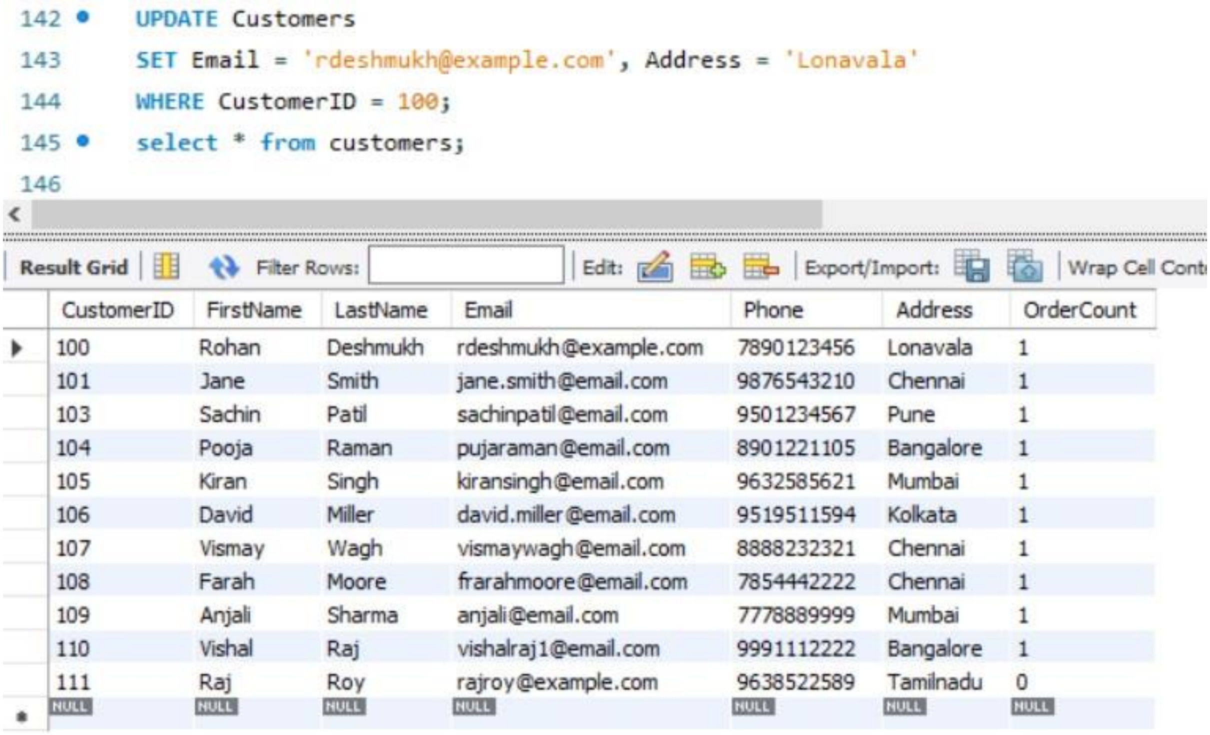
6. Write an SQL query to insert a new order into the "Orders" table. Include the customer ID, order date, and any other necessary information.



```sql
INSERT INTO Orders (orderID, CustomerID, OrderDate, TotalAmount)
VALUES (10010, 110 , '2024-04-10', 25000);
select * from orders;
```

OrderID	CustomerID	OrderDate	TotalAmount
10001	100	2024-04-01	90000.00
10002	101	2024-04-02	12800.00
10003	103	2024-04-03	69900.00
10004	104	2024-04-04	48020.00
10005	105	2024-04-05	39980.00
10006	106	2024-04-06	19000.00
10007	107	2024-04-07	14000.00
10008	108	2024-04-08	24500.00
10009	109	2024-04-09	88500.00
10010	110	2024-04-10	25000.00
NULL	NULL	NULL	NULL

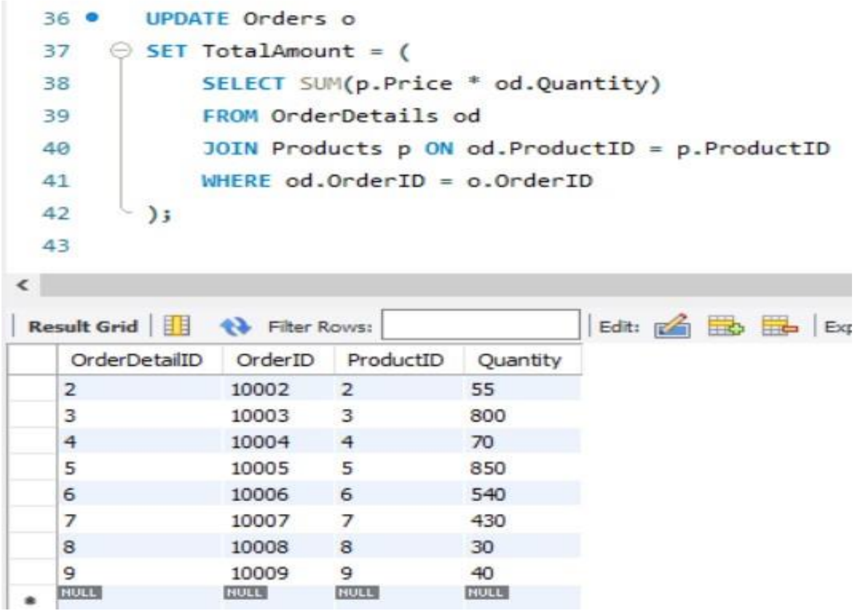
7. Write an SQL query to update the contact information (e.g., email and address) of a specific customer in the "Customers" table. Allow users to input the customer ID and new contact information.



```sql
UPDATE Customers
SET Email = 'rdeshmukh@example.com', Address = 'Lonavala'
WHERE CustomerID = 100;
select * from customers;
```

CustomerID	FirstName	LastName	Email	Phone	Address	OrderCount
100	Rohan	Deshmukh	rdeshmukh@example.com	7890123456	Lonavala	1
101	Jane	Smith	jane.smith@email.com	9876543210	Chennai	1
103	Sachin	Patil	sachinpatil@email.com	9501234567	Pune	1
104	Pooja	Raman	pujaraman@email.com	8901221105	Bangalore	1
105	Kiran	Singh	kiransingh@email.com	9632585621	Mumbai	1
106	David	Miller	david.miller@email.com	9519511594	Kolkata	1
107	Vismay	Wagh	vismaywagh@email.com	8888232321	Chennai	1
108	Farah	Moore	frarahmoore@email.com	7854442222	Chennai	1
109	Anjali	Sharma	anjali@email.com	7778889999	Mumbai	1
110	Vishal	Raj	vishalraj1@email.com	9991112222	Bangalore	1
111	Raj	Roy	rajroy@example.com	9638522589	Tamilnadu	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL

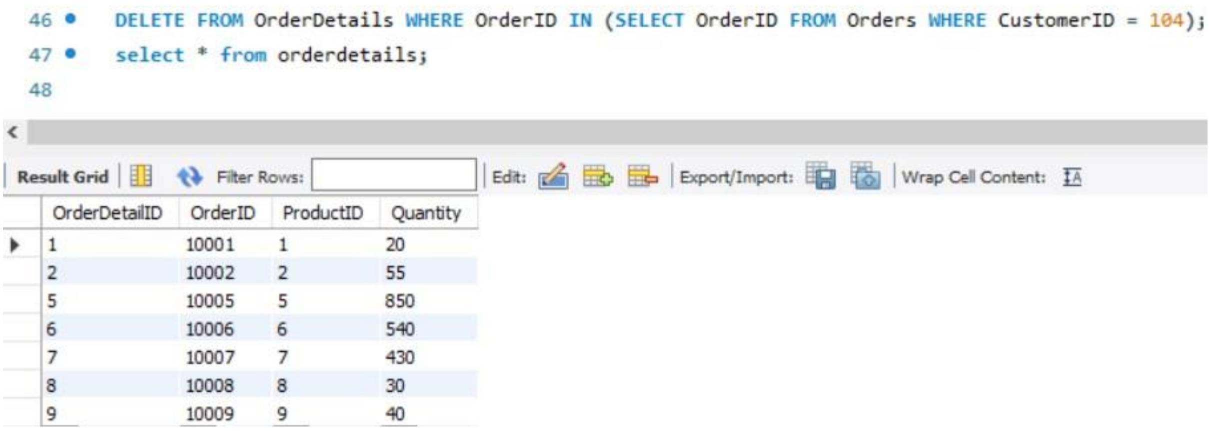
8. Write an SQL query to recalculate and update the total cost of each order in the "Orders" table based on the prices and quantities in the "OrderDetails" table.



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

OrderDetailID	OrderID	ProductID	Quantity
2	10002	2	55
3	10003	3	800
4	10004	4	70
5	10005	5	850
6	10006	6	540
7	10007	7	430
8	10008	8	30
9	10009	9	40
NULL	NULL	NULL	NULL

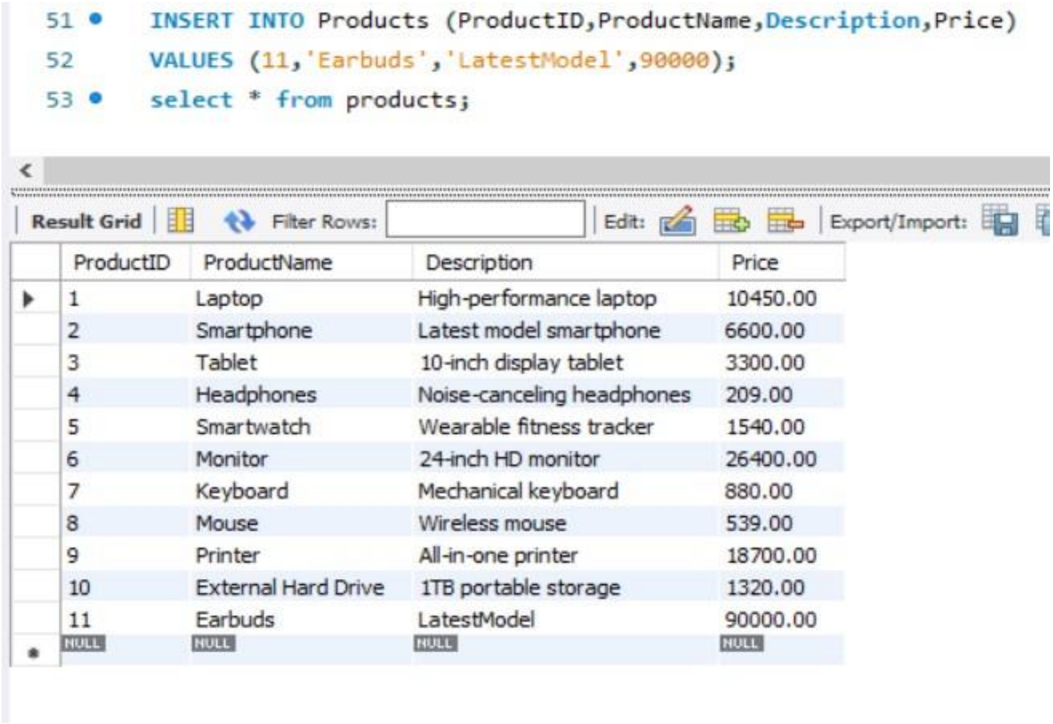
9. Write an SQL query to delete all orders and their associated order details for a specific customer from the "Orders" and "OrderDetails" tables. Allow users to input the customer ID as a parameter.



```
DELETE FROM OrderDetails WHERE OrderID IN (SELECT OrderID FROM Orders WHERE CustomerID = 104);
select * from orderdetails;
```

OrderDetailID	OrderID	ProductID	Quantity
1	10001	1	20
2	10002	2	55
5	10005	5	850
6	10006	6	540
7	10007	7	430
8	10008	8	30
9	10009	9	40

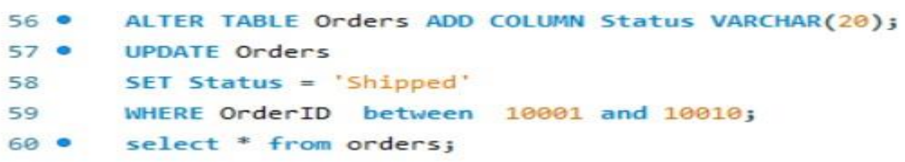
10. Write an SQL query to insert a new electronic gadget product into the "Products" table, including product name, category, price, and any other relevant details.



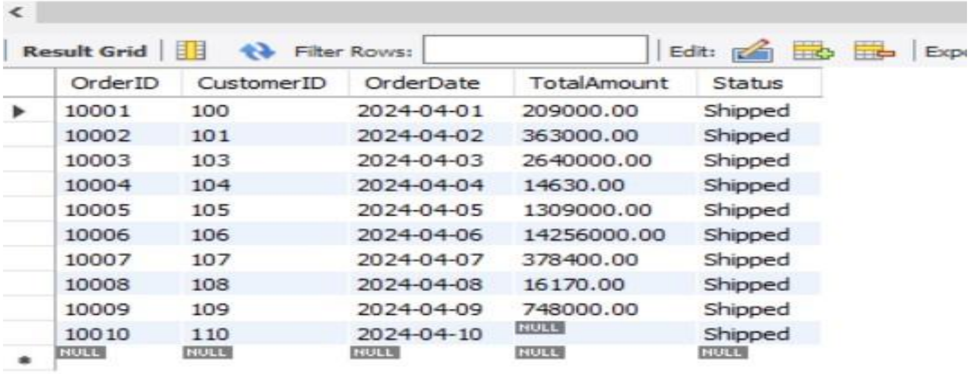
```
INSERT INTO Products (ProductID,ProductName,Description,Price)
VALUES (11,'Earbuds','LatestModel',90000);
select * from products;
```

ProductID	ProductName	Description	Price
1	Laptop	High-performance laptop	10450.00
2	Smartphone	Latest model smartphone	6600.00
3	Tablet	10-inch display tablet	3300.00
4	Headphones	Noise-canceling headphones	209.00
5	Smartwatch	Wearable fitness tracker	1540.00
6	Monitor	24-inch HD monitor	26400.00
7	Keyboard	Mechanical keyboard	880.00
8	Mouse	Wireless mouse	539.00
9	Printer	All-in-one printer	18700.00
10	External Hard Drive	1TB portable storage	1320.00
11	Earbuds	LatestModel	90000.00
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"). Allow users to input the order ID and the new status.

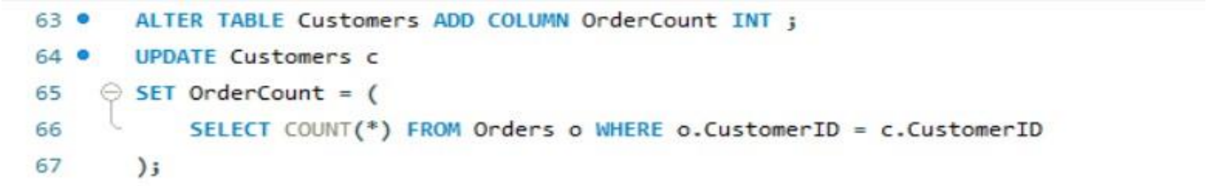


```
ALTER TABLE Orders ADD COLUMN Status VARCHAR(20);
UPDATE Orders
SET Status = 'Shipped'
WHERE OrderID  between  10001 and 10010;
select * from orders;
```

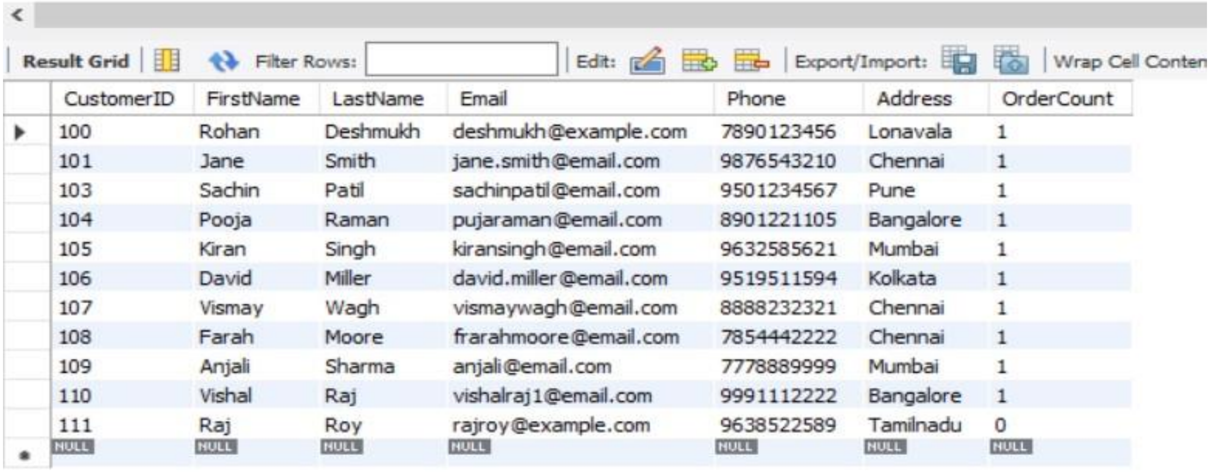


OrderID	CustomerID	OrderDate	TotalAmount	Status
10001	100	2024-04-01	209000.00	Shipped
10002	101	2024-04-02	363000.00	Shipped
10003	103	2024-04-03	2640000.00	Shipped
10004	104	2024-04-04	14630.00	Shipped
10005	105	2024-04-05	1309000.00	Shipped
10006	106	2024-04-06	14256000.00	Shipped
10007	107	2024-04-07	378400.00	Shipped
10008	108	2024-04-08	16170.00	Shipped
10009	109	2024-04-09	748000.00	Shipped
10010	110	2024-04-10	NULL	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.



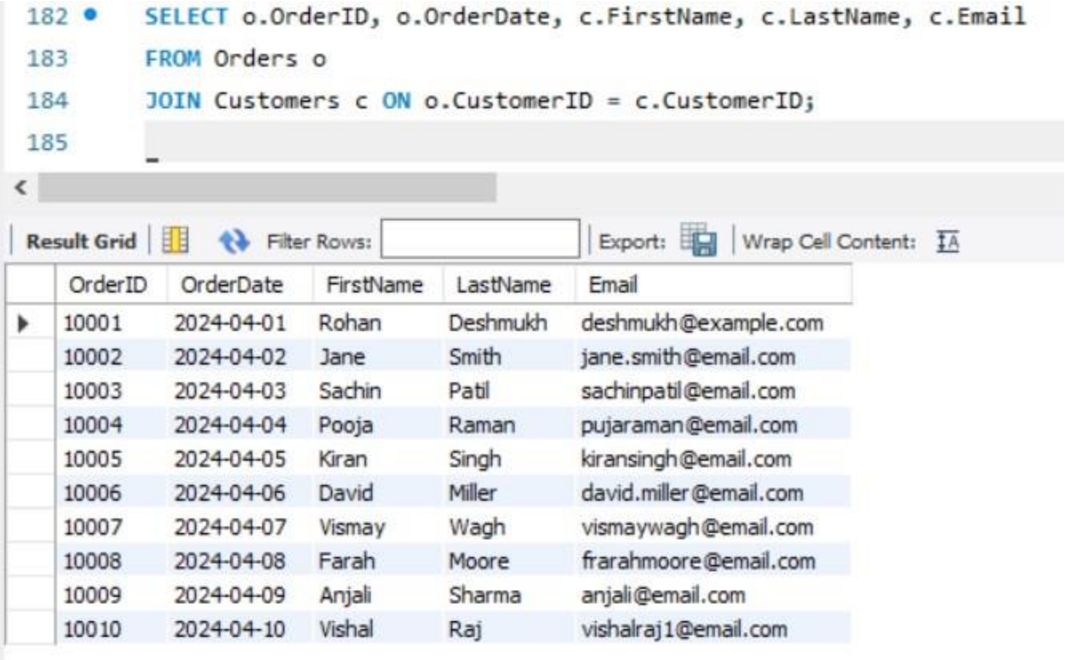
```
ALTER TABLE Customers ADD COLUMN OrderCount INT ;
UPDATE Customers c
SET OrderCount = (
    SELECT COUNT(*) FROM Orders o WHERE o.CustomerID = c.CustomerID
);
```



CustomerID	FirstName	LastName	Email	Phone	Address	OrderCount
100	Rohan	Deshmukh	deshmukh@example.com	7890123456	Lonavala	1
101	Jane	Smith	jane.smith@email.com	9876543210	Chennai	1
103	Sachin	Patil	sachinpatil@email.com	9501234567	Pune	1
104	Pooja	Raman	pujaraman@email.com	8901221105	Bangalore	1
105	Kiran	Singh	kiransingh@email.com	9632585621	Mumbai	1
106	David	Miller	david.miller@email.com	9519511594	Kolkata	1
107	Vismay	Wagh	vismaywagh@email.com	8888232321	Chennai	1
108	Farah	Moore	frarahmoore@email.com	7854442222	Chennai	1
109	Anjali	Sharma	anjali@email.com	7778889999	Mumbai	1
110	Vishal	Raj	vishalraj1@email.com	9991112222	Bangalore	1
111	Raj	Roy	rajroy@example.com	9638522589	Tamilnadu	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Task 3.
Aggregate functions, Having, Order By, GroupBy and Joins:

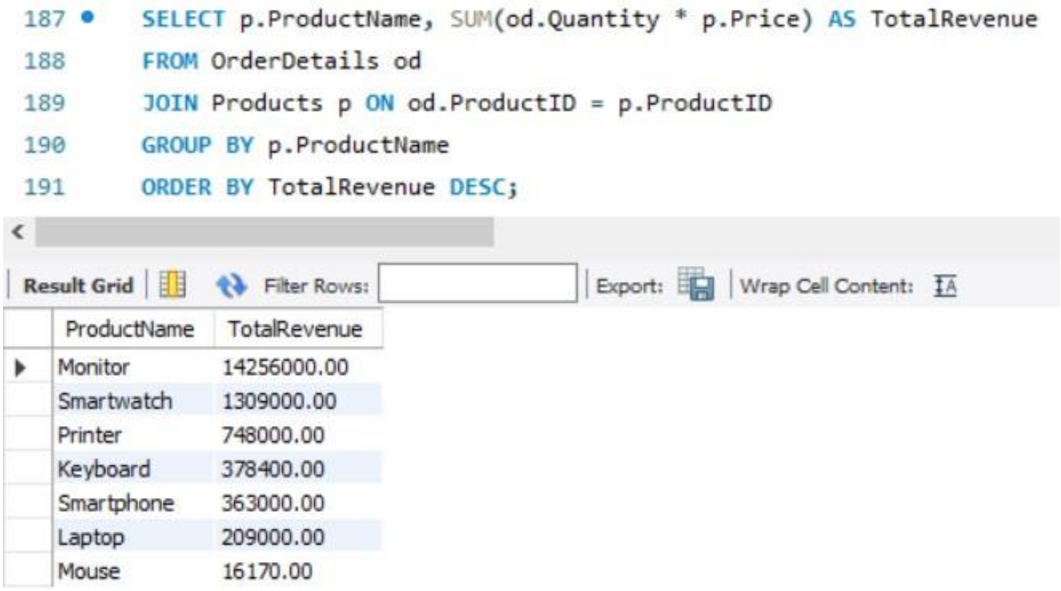
1. Write an SQL query to retrieve a list of all orders along with customer information (e.g., customer name) for each order.



```
SELECT o.OrderID, o.OrderDate, c.FirstName, c.LastName, c.Email
FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID;
```

OrderID	OrderDate	FirstName	LastName	Email
10001	2024-04-01	Rohan	Deshmukh	deshmukh@example.com
10002	2024-04-02	Jane	Smith	jane.smith@email.com
10003	2024-04-03	Sachin	Patil	sachinpatil@email.com
10004	2024-04-04	Pooja	Raman	pujaraman@email.com
10005	2024-04-05	Kiran	Singh	kiransingh@email.com
10006	2024-04-06	David	Miller	david.miller@email.com
10007	2024-04-07	Vismay	Wagh	vismaywagh@email.com
10008	2024-04-08	Farah	Moore	frarahmoore@email.com
10009	2024-04-09	Anjali	Sharma	anjali@email.com
10010	2024-04-10	Vishal	Raj	vishalraj1@email.com

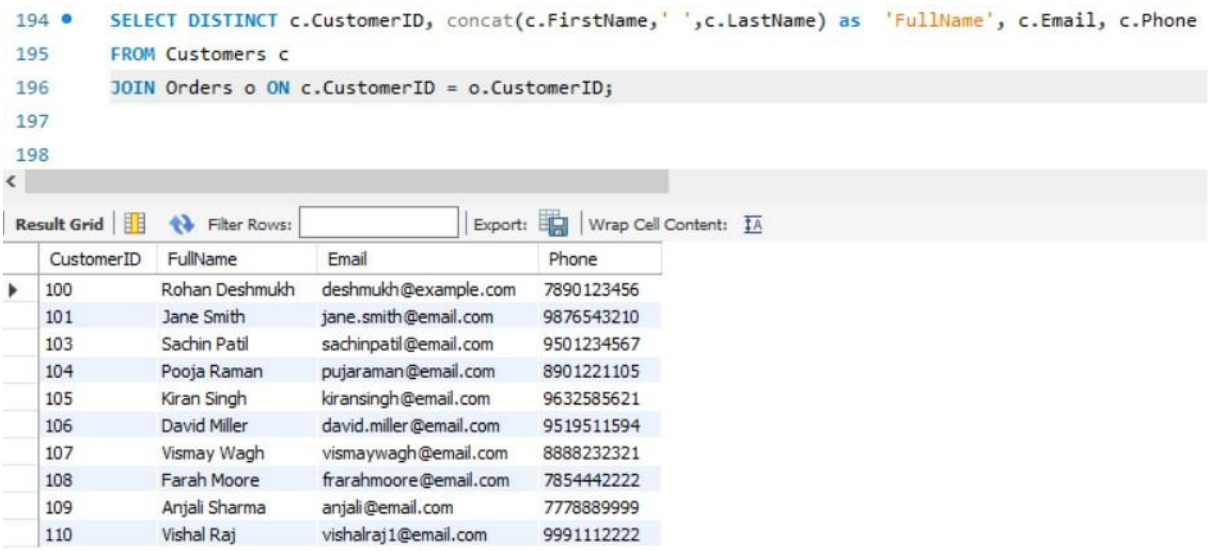
2. Write an SQL query to find the total revenue generated by each electronic gadget product. Include the product name and the total revenue.



```
SELECT p.ProductName, SUM(od.Quantity * p.Price) AS TotalRevenue
FROM OrderDetails od
JOIN Products p ON od.ProductID = p.ProductID
GROUP BY p.ProductName
ORDER BY TotalRevenue DESC;
```

ProductName	TotalRevenue
Monitor	14256000.00
Smartwatch	1309000.00
Printer	748000.00
Keyboard	378400.00
Smartphone	363000.00
Laptop	209000.00
Mouse	16170.00

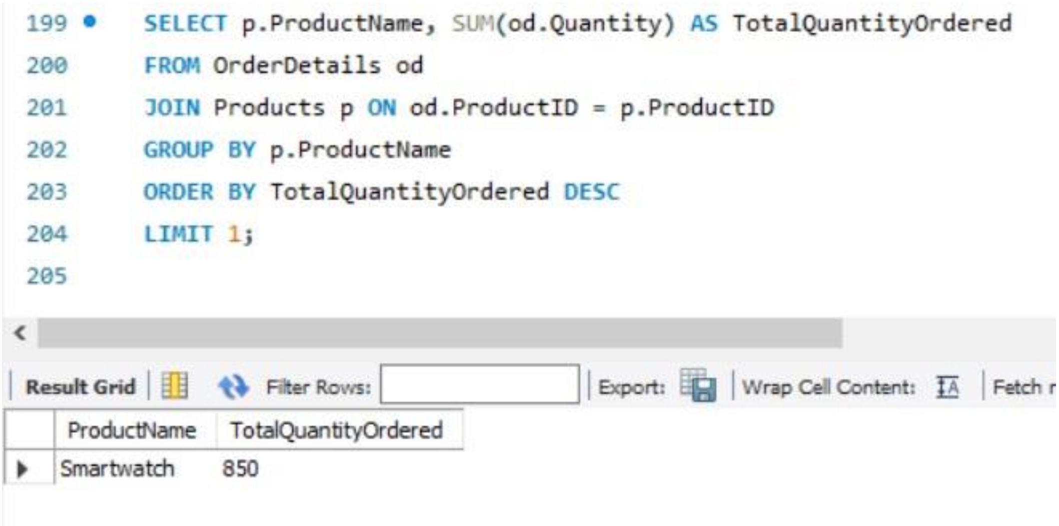
3. Write an SQL query to list all customers who have made at least one purchase. Include their names and contact information.



```
SELECT DISTINCT c.CustomerID, concat(c.FirstName,' ',c.LastName) as  'FullName', c.Email, c.Phone
FROM Customers c
JOIN Orders o ON c.CustomerID = o.CustomerID;
```

CustomerID	FullName	Email	Phone
100	Rohan Deshmukh	deshmukh@example.com	7890123456
101	Jane Smith	jane.smith@email.com	9876543210
103	Sachin Patil	sachinpatil@email.com	9501234567
104	Pooja Raman	pujaraman@email.com	8901221105
105	Kiran Singh	kiransingh@email.com	9632585621
106	David Miller	david.miller@email.com	9519511594
107	Vismay Wagh	vismaywagh@email.com	8888232321
108	Farah Moore	frarahmoore@email.com	7854442222
109	Anjali Sharma	anjali@email.com	7778889999
110	Vishal Raj	vishalraj1@email.com	9991112222

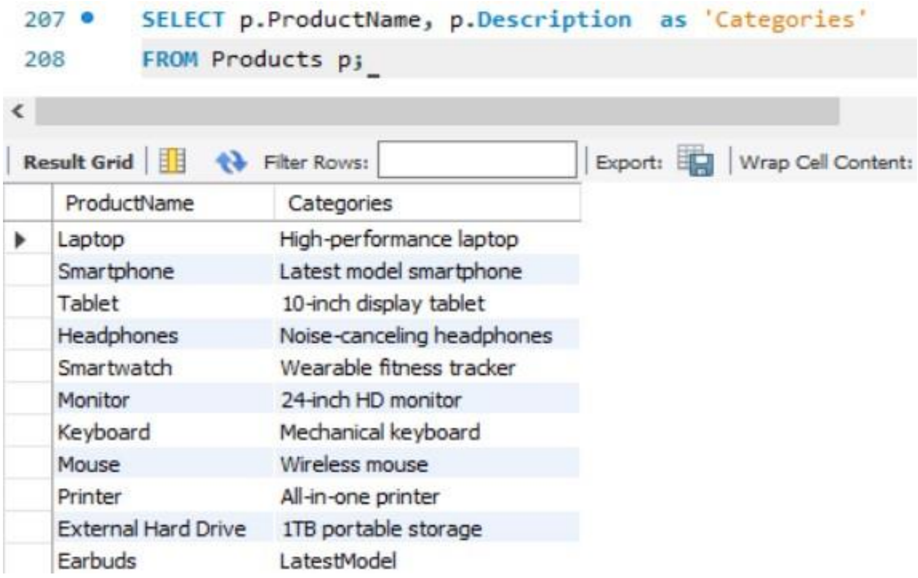
4. Write an SQL query to find the most popular electronic gadget, which is the one with the highest total quantity ordered. Include the product name and the total quantity ordered.



```
SELECT p.ProductName, SUM(od.Quantity) AS TotalQuantityOrdered
FROM OrderDetails od
JOIN Products p ON od.ProductID = p.ProductID
GROUP BY p.ProductName
ORDER BY TotalQuantityOrdered DESC
LIMIT 1;
```

ProductName	TotalQuantityOrdered
Smartwatch	850

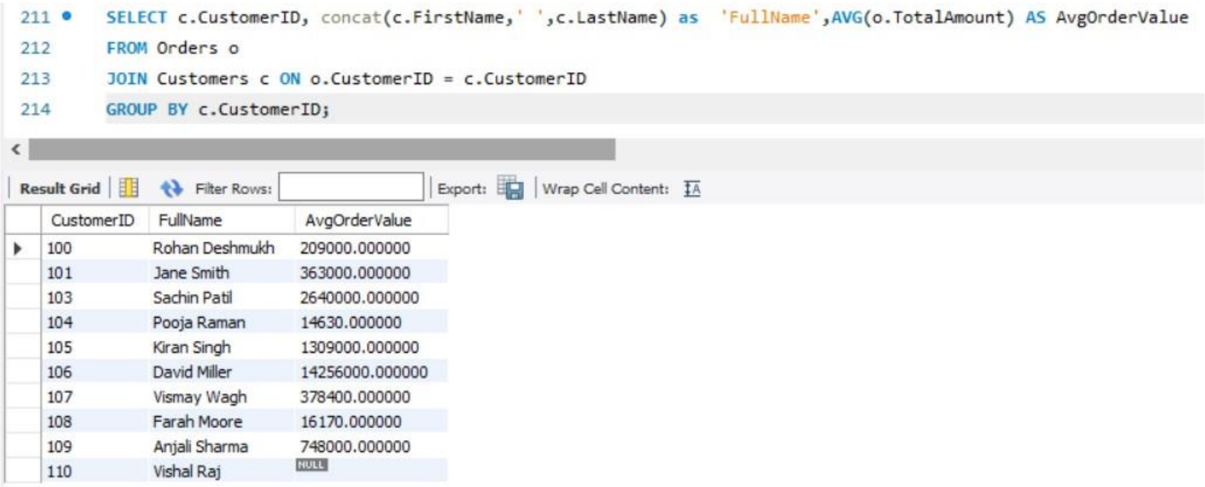
5. Write an SQL query to retrieve a list of electronic gadgets along with their corresponding categories.



```sql
SELECT p.ProductName, p.Description  as 'Categories'
FROM Products p;
```

ProductName	Categories
Laptop	High-performance laptop
Smartphone	Latest model smartphone
Tablet	10-inch display tablet
Headphones	Noise-canceling headphones
Smartwatch	Wearable fitness tracker
Monitor	24-inch HD monitor
Keyboard	Mechanical keyboard
Mouse	Wireless mouse
Printer	All-in-one printer
External Hard Drive	1TB portable storage
Earbuds	LatestModel

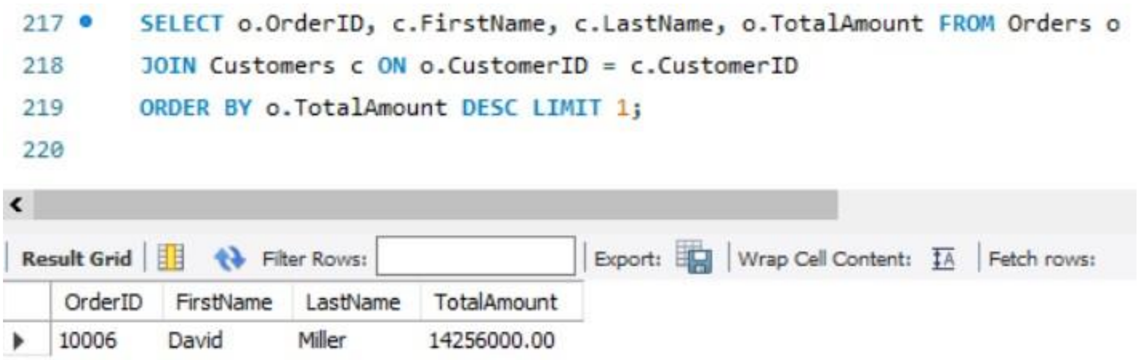
6. Write an SQL query to calculate the average order value for each customer. Include the customer's name and their average order value.



```sql
SELECT c.CustomerID, concat(c.FirstName,' ',c.LastName) as  'FullName',AVG(o.TotalAmount) AS AvgOrderValue
FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
GROUP BY c.CustomerID;
```

CustomerID	FullName	AvgOrderValue
100	Rohan Deshmukh	209000.000000
101	Jane Smith	363000.000000
103	Sachin Patil	2640000.000000
104	Pooja Raman	14630.000000
105	Kiran Singh	1309000.000000
106	David Miller	14256000.000000
107	Vismay Wagh	378400.000000
108	Farah Moore	16170.000000
109	Anjali Sharma	748000.000000
110	Vishal Raj	NULL

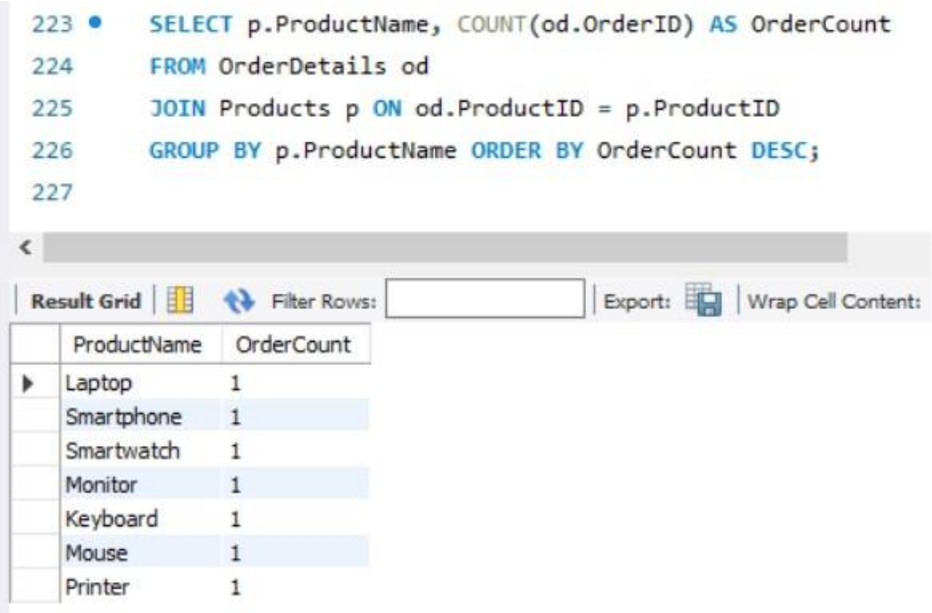
7. Write an SQL query to find the order with the highest total revenue. Include the order ID, customer information, and the total revenue.



```sql
SELECT o.OrderID, c.FirstName, c.LastName, o.TotalAmount FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
ORDER BY o.TotalAmount DESC LIMIT 1;
```

OrderID	FirstName	LastName	TotalAmount
10006	David	Miller	14256000.00

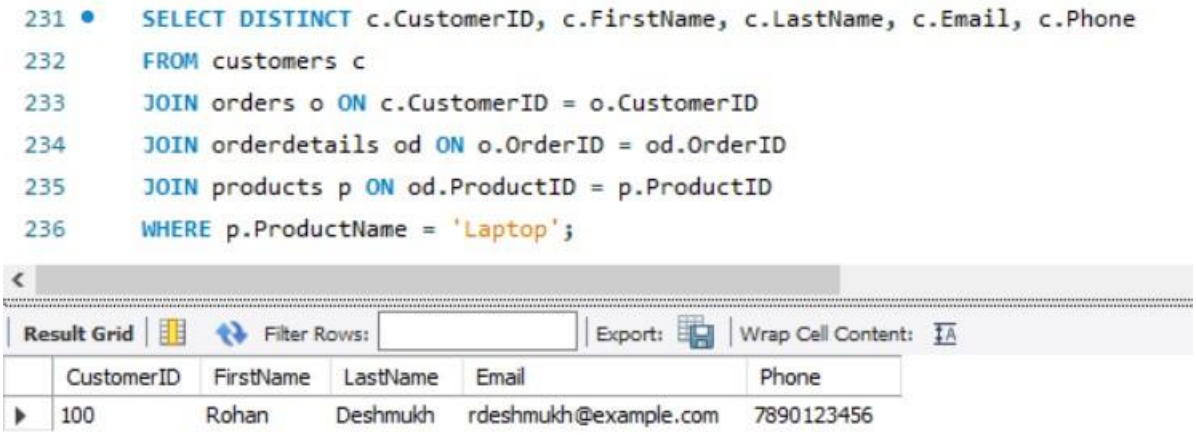
8. Write an SQL query to list electronic gadgets and the number of times each product has been ordered.



```sql
SELECT p.ProductName, COUNT(od.OrderID) AS OrderCount
FROM OrderDetails od
JOIN Products p ON od.ProductID = p.ProductID
GROUP BY p.ProductName ORDER BY OrderCount DESC;
```

ProductName	OrderCount
Laptop	1
Smartphone	1
Smartwatch	1
Monitor	1
Keyboard	1
Mouse	1
Printer	1

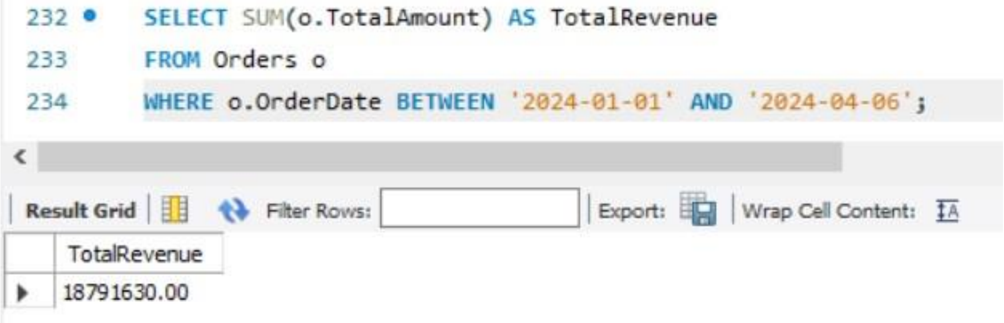
9. Write an SQL query to find customers who have purchased a specific electronic gadget product.
Allow users to input the product name as a parameter.



```sql
SELECT DISTINCT c.CustomerID, c.FirstName, c.LastName, c.Email, c.Phone
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
WHERE p.ProductName = 'Laptop';
```

CustomerID	FirstName	LastName	Email	Phone
100	Rohan	Deshmukh	rdeshmukh@example.com	7890123456

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.

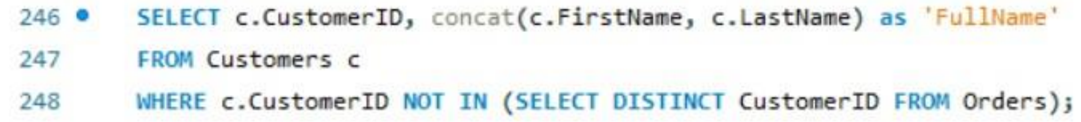


```sql
SELECT SUM(o.TotalAmount) AS TotalRevenue
FROM Orders o
WHERE o.OrderDate BETWEEN '2024-01-01' AND '2024-04-06';
```

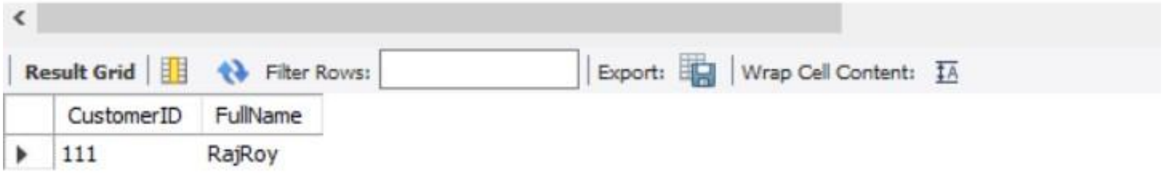
TotalRevenue
18791630.00

Task 4. Subquery and its type:

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

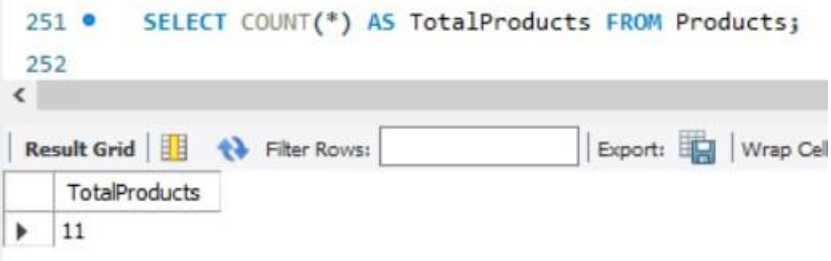


```
SELECT c.CustomerID, concat(c.FirstName, c.LastName) as 'FullName'
FROM Customers c
WHERE c.CustomerID NOT IN (SELECT DISTINCT CustomerID FROM Orders);
```



CustomerID	FullName
111	RajRoy

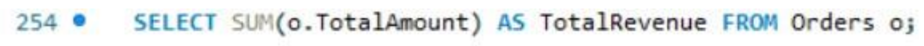
2. Write an SQL query to find the total number of products available for sale.



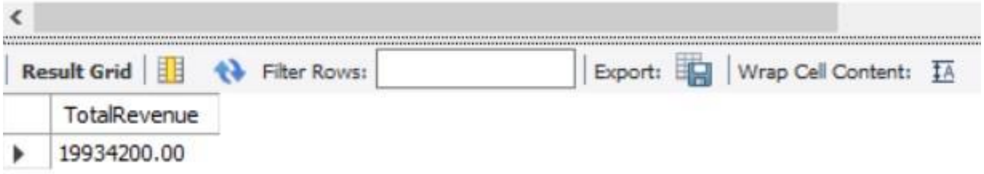
```
SELECT COUNT(*) AS TotalProducts FROM Products;
```

TotalProducts
11

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

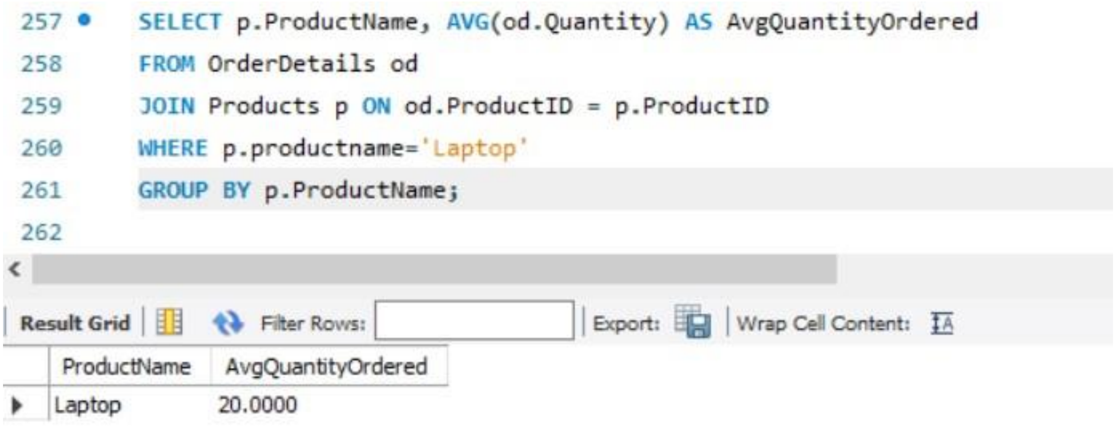


```
SELECT SUM(o.TotalAmount) AS TotalRevenue FROM Orders o;
```



TotalRevenue
19934200.00

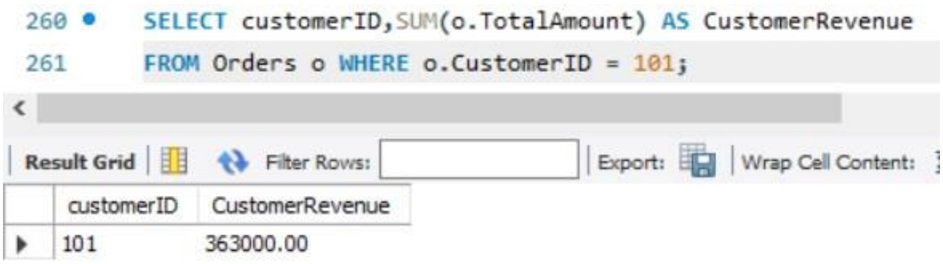
**4. Write an SQL query to calculate the average quantity ordered for products in a specific category.
Allow users to input the category name as a parameter.**



```
SELECT p.ProductName, AVG(od.Quantity) AS AvgQuantityOrdered
FROM OrderDetails od
JOIN Products p ON od.ProductID = p.ProductID
WHERE p.productname='Laptop'
GROUP BY p.ProductName;
```

ProductName	AvgQuantityOrdered
Laptop	20.0000

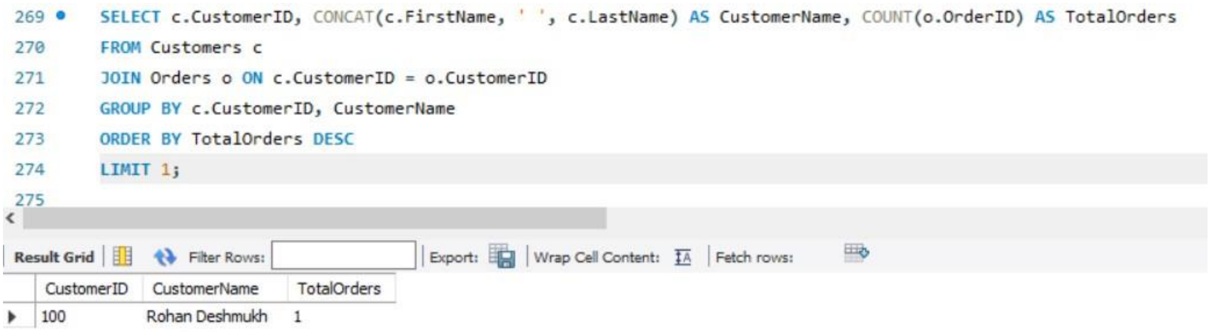
5. Write an SQL query to calculate the total revenue generated by a specific customer. Allow users to input the customer ID as a parameter.



```sql
SELECT customerID,SUM(o.TotalAmount) AS CustomerRevenue
FROM Orders o WHERE o.CustomerID = 101;
```

customerID	CustomerRevenue
101	363000.00

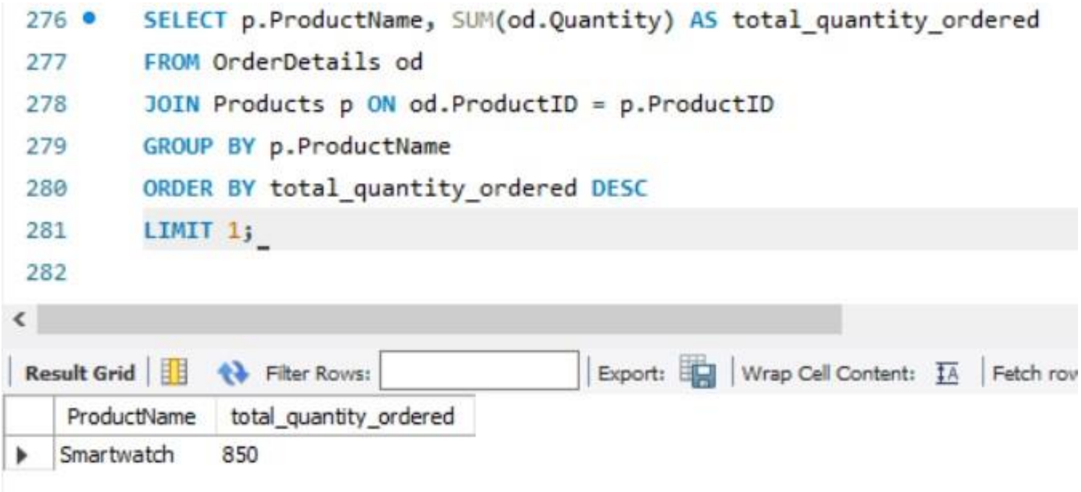
6. Write an SQL query to find the customers who have placed the most orders. List their names and the number of orders they've placed.



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

CustomerID	CustomerName	TotalOrders
100	Rohan Deshmukh	1

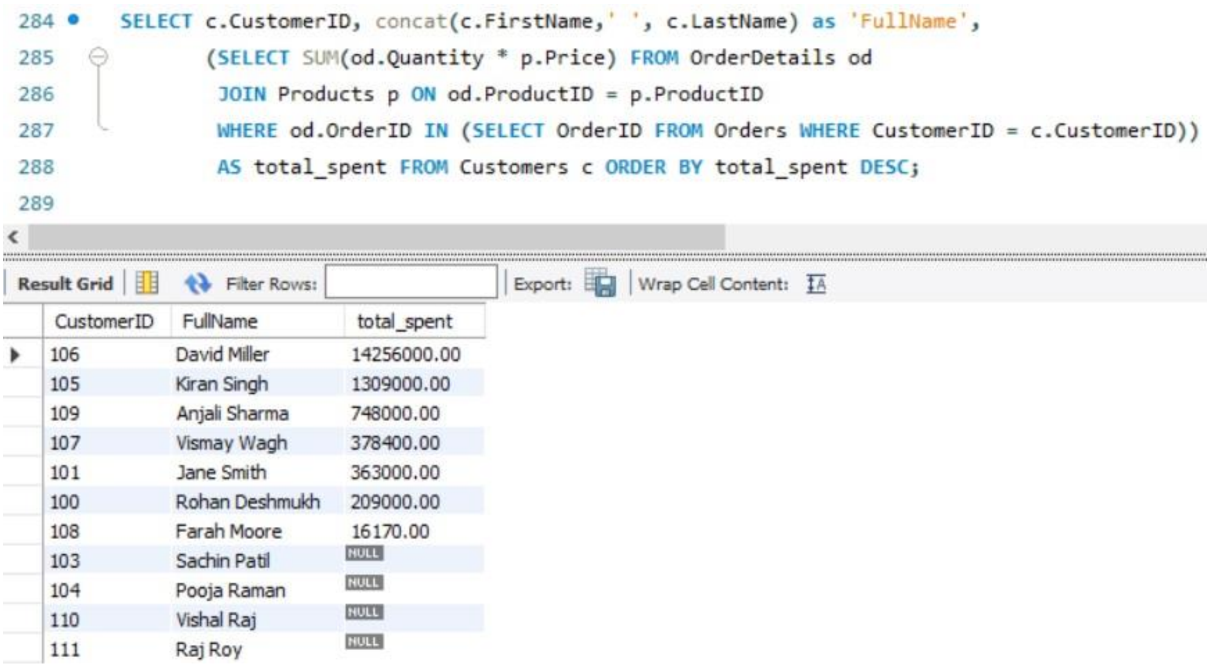
7. Write an SQL query to find the most popular product category, which is the one with the highest total quantity ordered across all orders.



```sql
SELECT p.ProductName, SUM(od.Quantity) AS total_quantity_ordered
FROM OrderDetails od
JOIN Products p ON od.ProductID = p.ProductID
GROUP BY p.ProductName
ORDER BY total_quantity_ordered DESC
LIMIT 1;
```

ProductName	total_quantity_ordered
Smartwatch	850

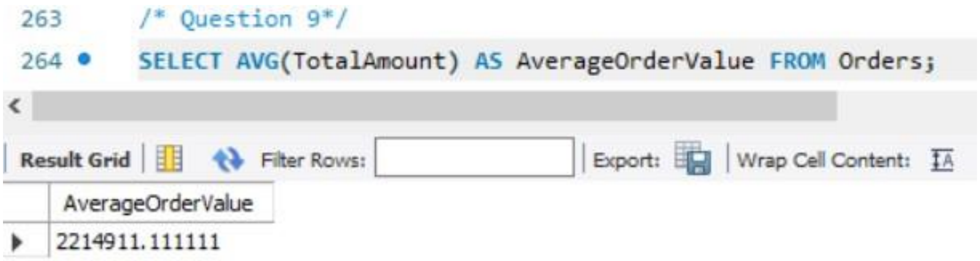
8. Write an SQL query to find the customer who has spent the most money (highest total revenue)
on electronic gadgets. List their name and total spending.



```
SELECT c.CustomerID, concat(c.FirstName,' ', c.LastName) as 'FullName',
        (SELECT SUM(od.Quantity * p.Price) FROM OrderDetails od
         JOIN Products p ON od.ProductID = p.ProductID
         WHERE od.OrderID IN (SELECT OrderID FROM Orders WHERE CustomerID = c.CustomerID))
         AS total_spent FROM Customers c ORDER BY total_spent DESC;
```

CustomerID	FullName	total_spent
106	David Miller	14256000.00
105	Kiran Singh	1309000.00
109	Anjali Sharma	748000.00
107	Vismay Wagh	378400.00
101	Jane Smith	363000.00
100	Rohan Deshmukh	209000.00
108	Farah Moore	16170.00
103	Sachin Patil	NULL
104	Pooja Raman	NULL
110	Vishal Raj	NULL
111	Raj Roy	NULL

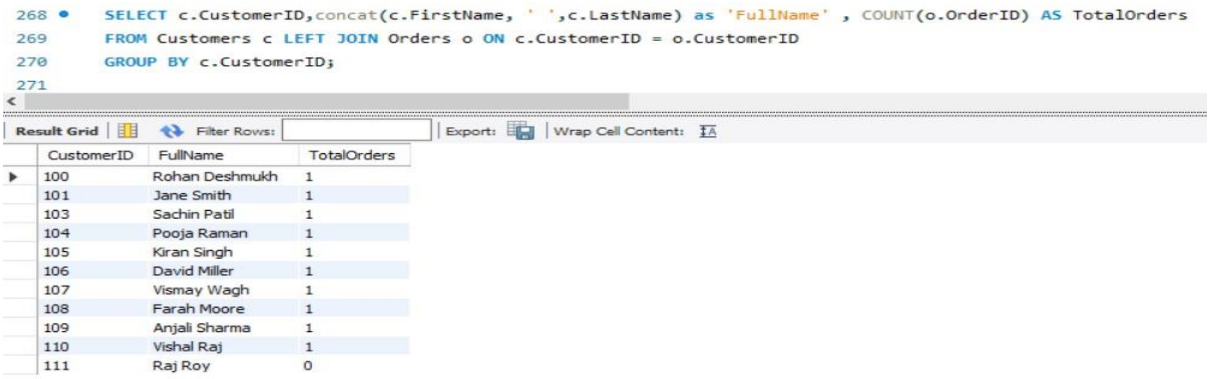
9. Write an SQL query to calculate the average order value (total revenue divided by the number of
orders) for all customers.



```
/* Question 9*/
SELECT AVG(TotalAmount) AS AverageOrderValue FROM Orders;
```

AverageOrderValue
2214911.111111

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,concat(c.FirstName, ' ',c.LastName) as 'FullName' , COUNT(o.OrderID) AS TotalOrders
FROM Customers c LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
GROUP BY c.CustomerID;
```

CustomerID	FullName	TotalOrders
100	Rohan Deshmukh	1
101	Jane Smith	1
103	Sachin Patil	1
104	Pooja Raman	1
105	Kiran Singh	1
106	David Miller	1
107	Vismay Wagh	1
108	Farah Moore	1
109	Anjali Sharma	1
110	Vishal Raj	1
111	Raj Roy	0