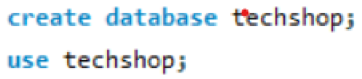
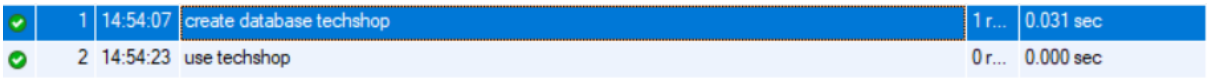
Assignment – 1 TechShop (SQL)

Task 1:

1. Create the database named "TechShop"



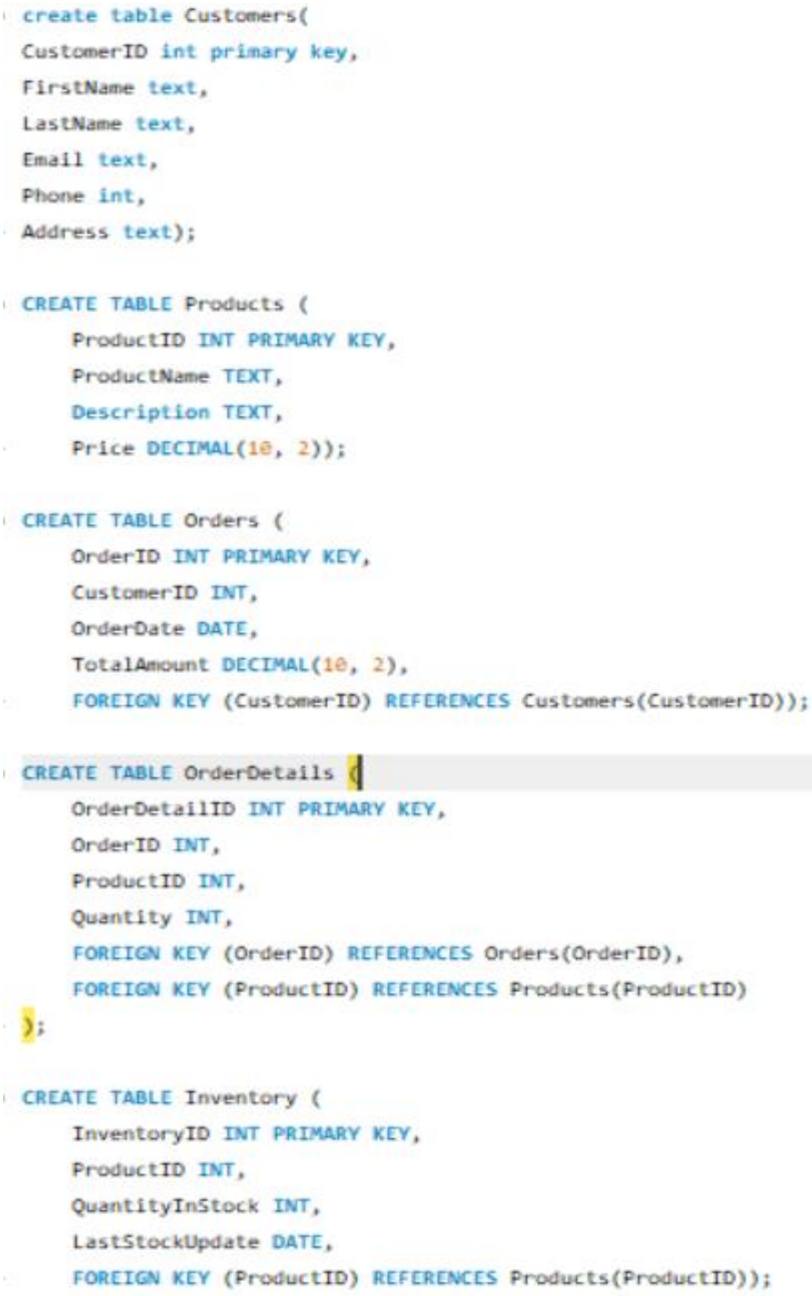
```
create database techshop;
use techshop;
```



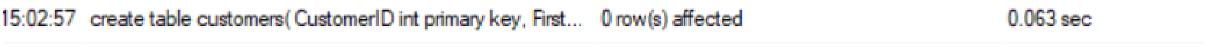
```
1   14:54:07   create database techshop   1 r...   0.031 sec
2   14:54:23   use techshop               0 r...   0.000 sec
```

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

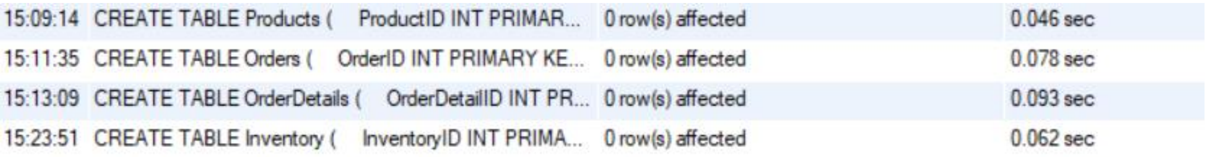
based on the provided schema.



```
create table Customers(
CustomerID int primary key,
FirstName text,
LastName text,
Email text,
Phone int,
Address text);

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

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

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)
);

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

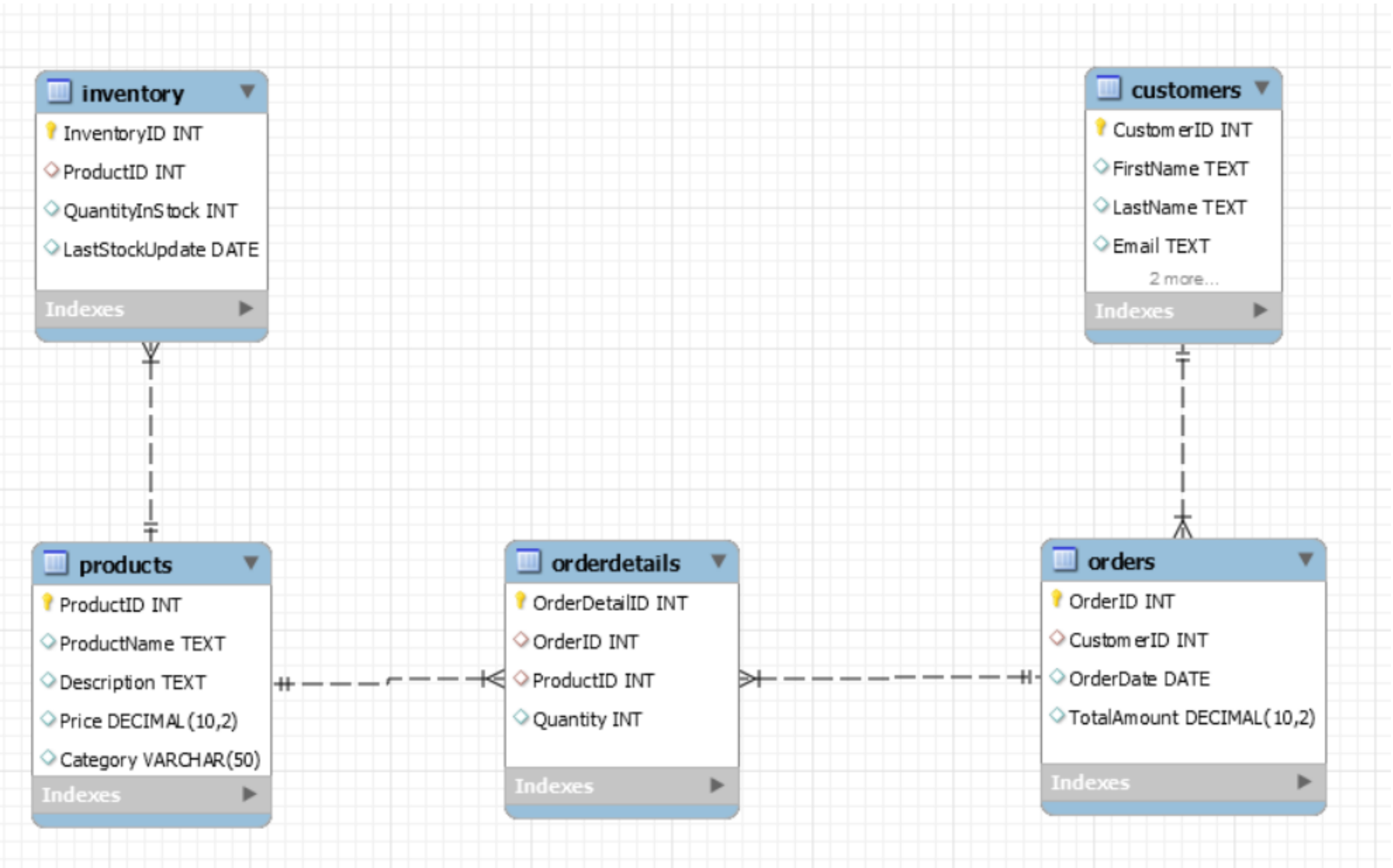


```
15:02:57   create table customers( CustomerID int primary key, First...   0 row(s) affected   0.063 sec
```



```
15:09:14  CREATE TABLE Products (    ProductID INT PRIMAR...   0 row(s) affected    0.046 sec
15:11:35  CREATE TABLE Orders (    OrderID INT PRIMARY KE...    0 row(s) affected    0.078 sec
15:13:09  CREATE TABLE OrderDetails (    OrderDetailID INT PR...  0 row(s) affected    0.093 sec
15:23:51  CREATE TABLE Inventory (    InventoryID INT PRIMA...  0 row(s) affected    0.062 sec
```

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



inventory
- InventoryID INT
- ProductID INT
- QuantityInStock INT
- LastStockUpdate DATE

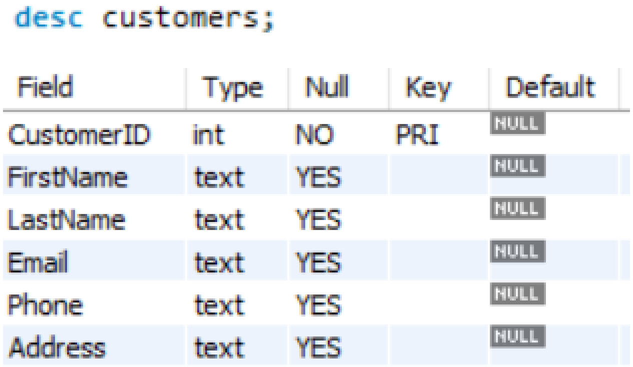
customers
- CustomerID INT
- FirstName TEXT
- LastName TEXT
- Email TEXT
- 2 more...

products
- ProductID INT
- ProductName TEXT
- Description TEXT
- Price DECIMAL(10,2)
- Category VARCHAR(50)

orderdetails
- OrderDetailID INT
- OrderID INT
- ProductID INT
- Quantity INT

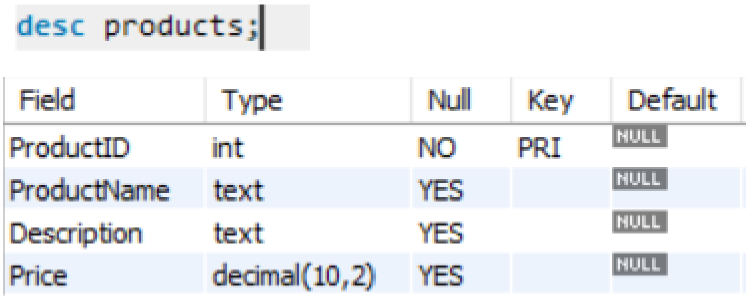
orders
- OrderID INT
- CustomerID INT
- OrderDate DATE
- TotalAmount DECIMAL(10,2)

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



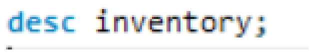
```
desc customers;
```

Field	Type	Null	Key	Default
CustomerID	int	NO	PRI	NULL
FirstName	text	YES		NULL
LastName	text	YES		NULL
Email	text	YES		NULL
Phone	text	YES		NULL
Address	text	YES		NULL

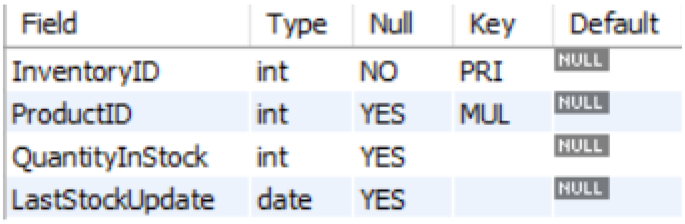


```
desc products;
```

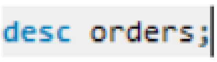
Field	Type	Null	Key	Default
ProductID	int	NO	PRI	NULL
ProductName	text	YES		NULL
Description	text	YES		NULL
Price	decimal(10,2)	YES		NULL



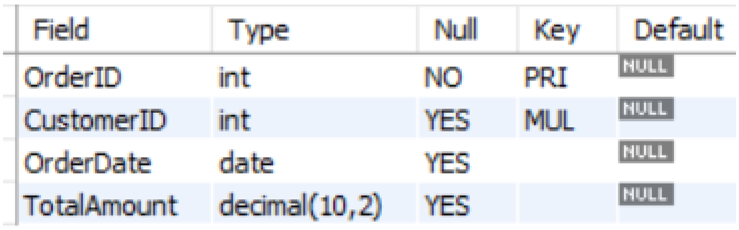
```
desc inventory;
```



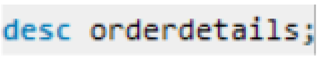
Field	Type	Null	Key	Default
InventoryID	int	NO	PRI	NULL
ProductID	int	YES	MUL	NULL
QuantityInStock	int	YES		NULL
LastStockUpdate	date	YES		NULL



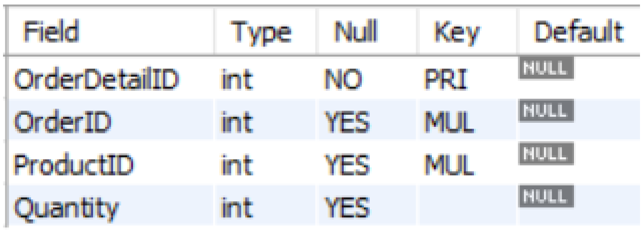
```
desc orders;
```



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



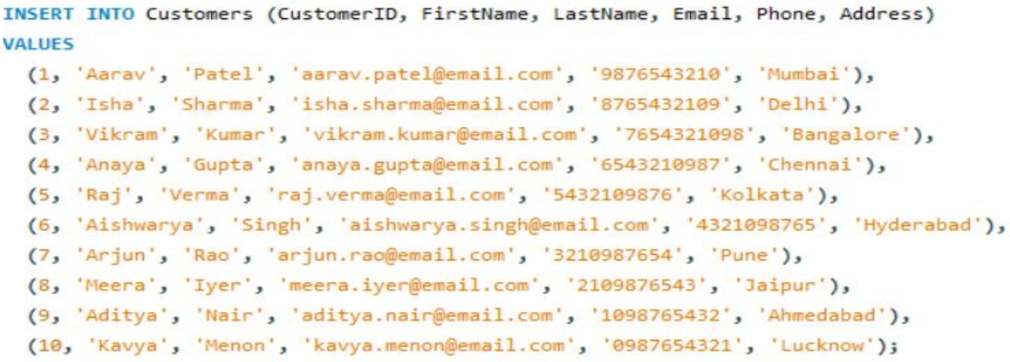
```
desc orderdetails;
```



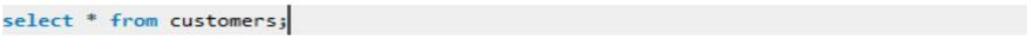
Field	Type	Null	Key	Default
OrderDetailID	int	NO	PRI	NULL
OrderID	int	YES	MUL	NULL
ProductID	int	YES	MUL	NULL
Quantity	int	YES		NULL

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

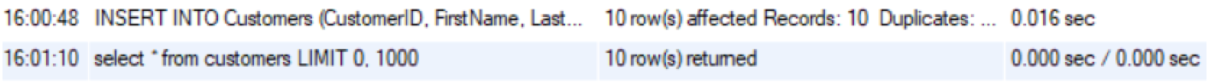
a. Customers



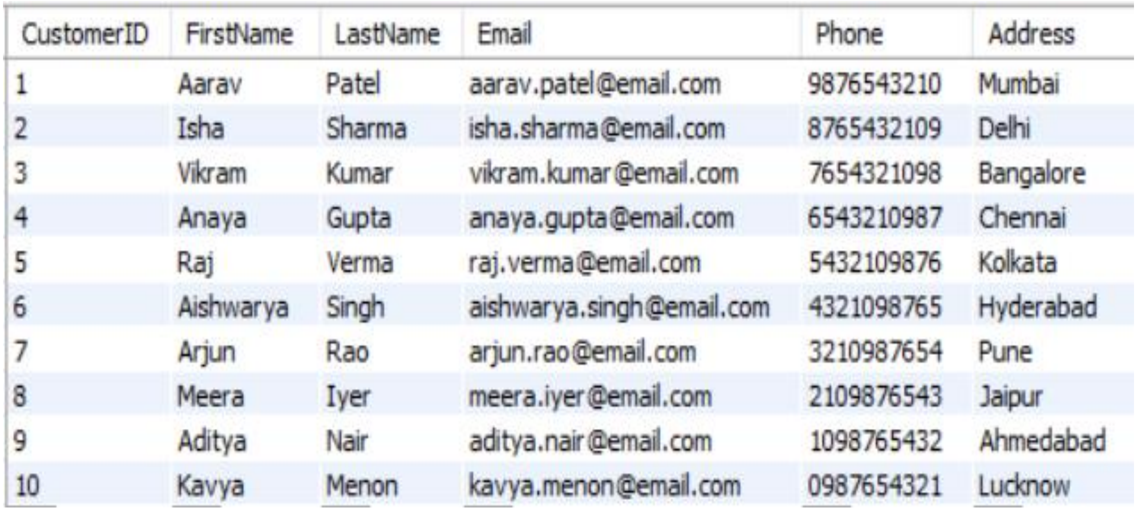
```
INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone, Address)
VALUES
  (1, 'Aarav', 'Patel', 'aarav.patel@email.com', '9876543210', 'Mumbai'),
  (2, 'Isha', 'Sharma', 'isha.sharma@email.com', '8765432109', 'Delhi'),
  (3, 'Vikram', 'Kumar', 'vikram.kumar@email.com', '7654321098', 'Bangalore'),
  (4, 'Anaya', 'Gupta', 'anaya.gupta@email.com', '6543210987', 'Chennai'),
  (5, 'Raj', 'Verma', 'raj.verma@email.com', '5432109876', 'Kolkata'),
  (6, 'Aishwarya', 'Singh', 'aishwarya.singh@email.com', '4321098765', 'Hyderabad'),
  (7, 'Arjun', 'Rao', 'arjun.rao@email.com', '3210987654', 'Pune'),
  (8, 'Meera', 'Iyer', 'meera.iyer@email.com', '2109876543', 'Jaipur'),
  (9, 'Aditya', 'Nair', 'aditya.nair@email.com', '1098765432', 'Ahmedabad'),
  (10, 'Kavya', 'Menon', 'kavya.menon@email.com', '0987654321', 'Lucknow');
```



```
select * from customers;
```

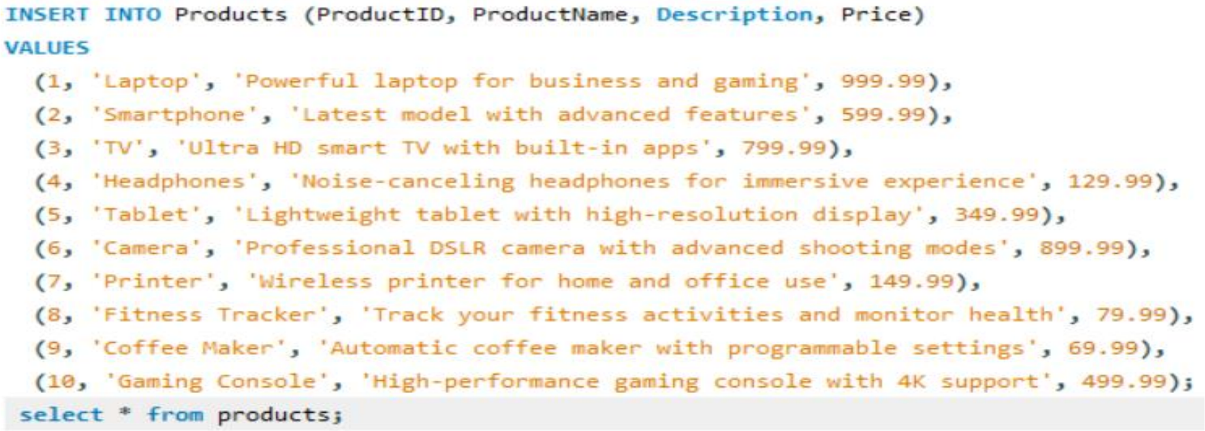


```
16:00:48  INSERT INTO Customers (CustomerID, FirstName, Last...  10 row(s) affected Records: 10  Duplicates: ...  0.016 sec
16:01:10  select * from customers LIMIT 0, 1000                   10 row(s) returned                                0.000 sec / 0.000 sec
```

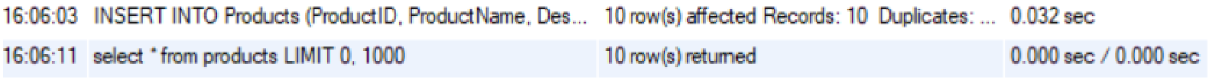


CustomerID	FirstName	LastName	Email	Phone	Address
1	Aarav	Patel	aarav.patel@email.com	9876543210	Mumbai
2	Isha	Sharma	isha.sharma@email.com	8765432109	Delhi
3	Vikram	Kumar	vikram.kumar@email.com	7654321098	Bangalore
4	Anaya	Gupta	anaya.gupta@email.com	6543210987	Chennai
5	Raj	Verma	raj.verma@email.com	5432109876	Kolkata
6	Aishwarya	Singh	aishwarya.singh@email.com	4321098765	Hyderabad
7	Arjun	Rao	arjun.rao@email.com	3210987654	Pune
8	Meera	Iyer	meera.iyer@email.com	2109876543	Jaipur
9	Aditya	Nair	aditya.nair@email.com	1098765432	Ahmedabad
10	Kavya	Menon	kavya.menon@email.com	0987654321	Lucknow

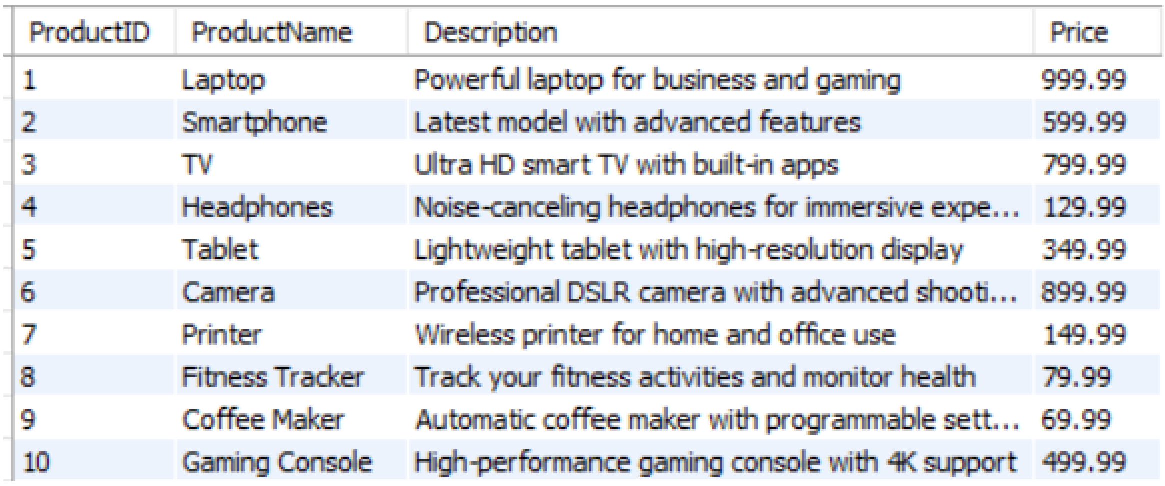
b. Products



```
INSERT INTO Products (ProductID, ProductName, Description, Price)
VALUES
  (1, 'Laptop', 'Powerful laptop for business and gaming', 999.99),
  (2, 'Smartphone', 'Latest model with advanced features', 599.99),
  (3, 'TV', 'Ultra HD smart TV with built-in apps', 799.99),
  (4, 'Headphones', 'Noise-canceling headphones for immersive experience', 129.99),
  (5, 'Tablet', 'Lightweight tablet with high-resolution display', 349.99),
  (6, 'Camera', 'Professional DSLR camera with advanced shooting modes', 899.99),
  (7, 'Printer', 'Wireless printer for home and office use', 149.99),
  (8, 'Fitness Tracker', 'Track your fitness activities and monitor health', 79.99),
  (9, 'Coffee Maker', 'Automatic coffee maker with programmable settings', 69.99),
  (10, 'Gaming Console', 'High-performance gaming console with 4K support', 499.99);
 select * from products;
```

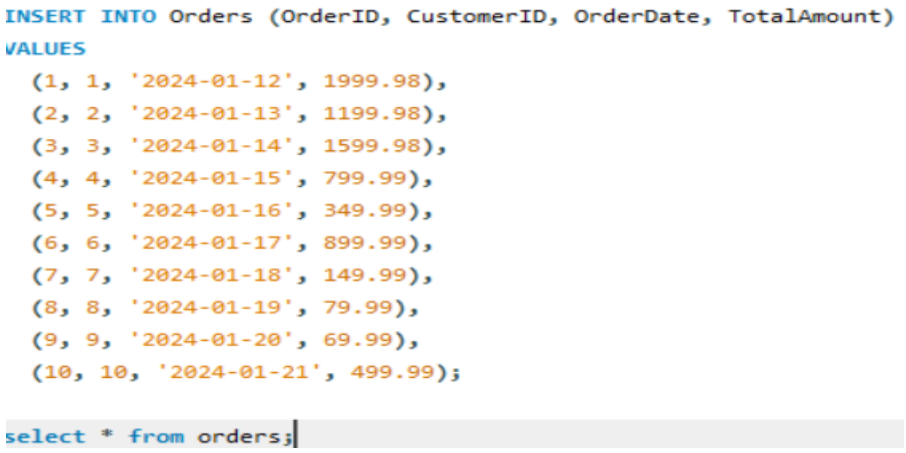


```
16:06:03  INSERT INTO Products (ProductID, ProductName, Des...  10 row(s) affected Records: 10  Duplicates: ...  0.032 sec
16:06:11  select * from products LIMIT 0, 1000                  10 row(s) returned                               0.000 sec / 0.000 sec
```

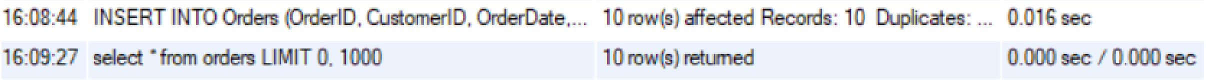


ProductID	ProductName	Description	Price
1	Laptop	Powerful laptop for business and gaming	999.99
2	Smartphone	Latest model with advanced features	599.99
3	TV	Ultra HD smart TV with built-in apps	799.99
4	Headphones	Noise-canceling headphones for immersive expe...	129.99
5	Tablet	Lightweight tablet with high-resolution display	349.99
6	Camera	Professional DSLR camera with advanced shooti...	899.99
7	Printer	Wireless printer for home and office use	149.99
8	Fitness Tracker	Track your fitness activities and monitor health	79.99
9	Coffee Maker	Automatic coffee maker with programmable sett...	69.99
10	Gaming Console	High-performance gaming console with 4K support	499.99

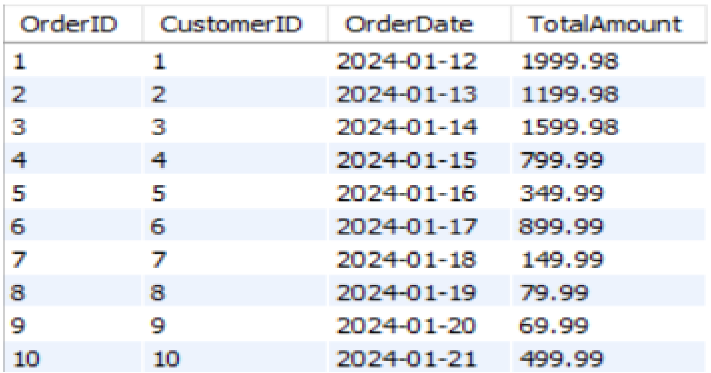
c. Orders



```
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)
VALUES
  (1, 1, '2024-01-12', 1999.98),
  (2, 2, '2024-01-13', 1199.98),
  (3, 3, '2024-01-14', 1599.98),
  (4, 4, '2024-01-15', 799.99),
  (5, 5, '2024-01-16', 349.99),
  (6, 6, '2024-01-17', 899.99),
  (7, 7, '2024-01-18', 149.99),
  (8, 8, '2024-01-19', 79.99),
  (9, 9, '2024-01-20', 69.99),
  (10, 10, '2024-01-21', 499.99);

select * from orders;
```

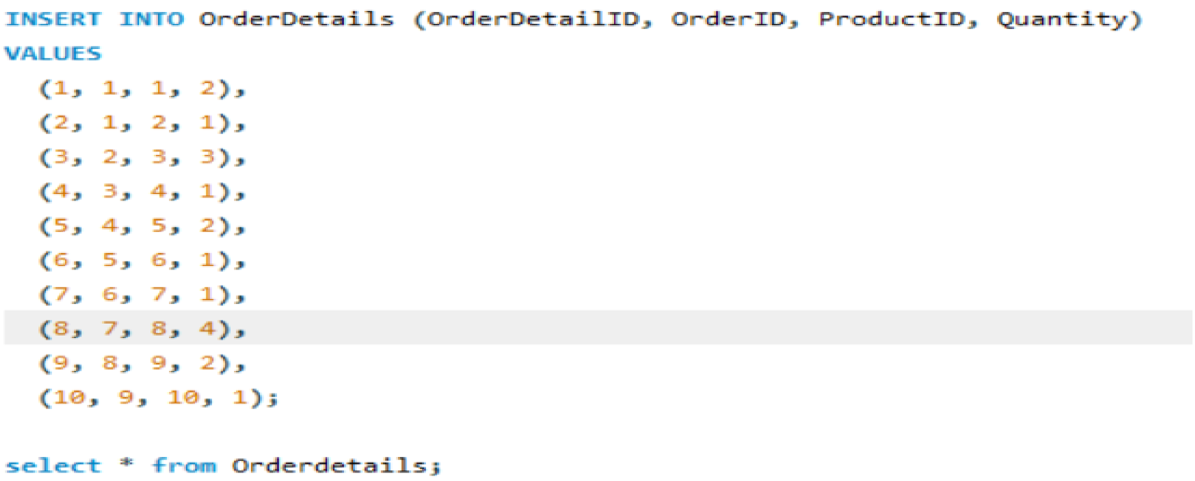


```
16:08:44  INSERT INTO Orders (OrderID, CustomerID, OrderDate,...  10 row(s) affected Records: 10  Duplicates: ...  0.016 sec
16:09:27  select * from orders LIMIT 0, 1000                      10 row(s) returned                               0.000 sec / 0.000 sec
```

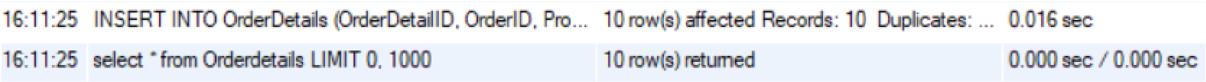


OrderID	CustomerID	OrderDate	TotalAmount
1	1	2024-01-12	1999.98
2	2	2024-01-13	1199.98
3	3	2024-01-14	1599.98
4	4	2024-01-15	799.99
5	5	2024-01-16	349.99
6	6	2024-01-17	899.99
7	7	2024-01-18	149.99
8	8	2024-01-19	79.99
9	9	2024-01-20	69.99
10	10	2024-01-21	499.99

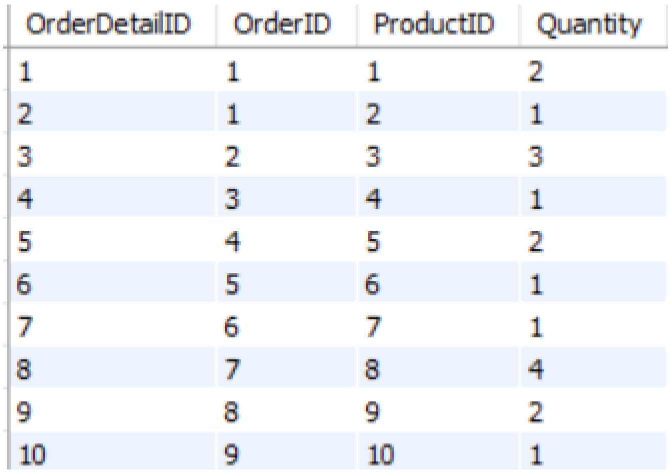
d. OrderDetails



```
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity)
VALUES
  (1, 1, 1, 2),
  (2, 1, 2, 1),
  (3, 2, 3, 3),
  (4, 3, 4, 1),
  (5, 4, 5, 2),
  (6, 5, 6, 1),
  (7, 6, 7, 1),
  (8, 7, 8, 4),
  (9, 8, 9, 2),
  (10, 9, 10, 1);

select * from Orderdetails;
```

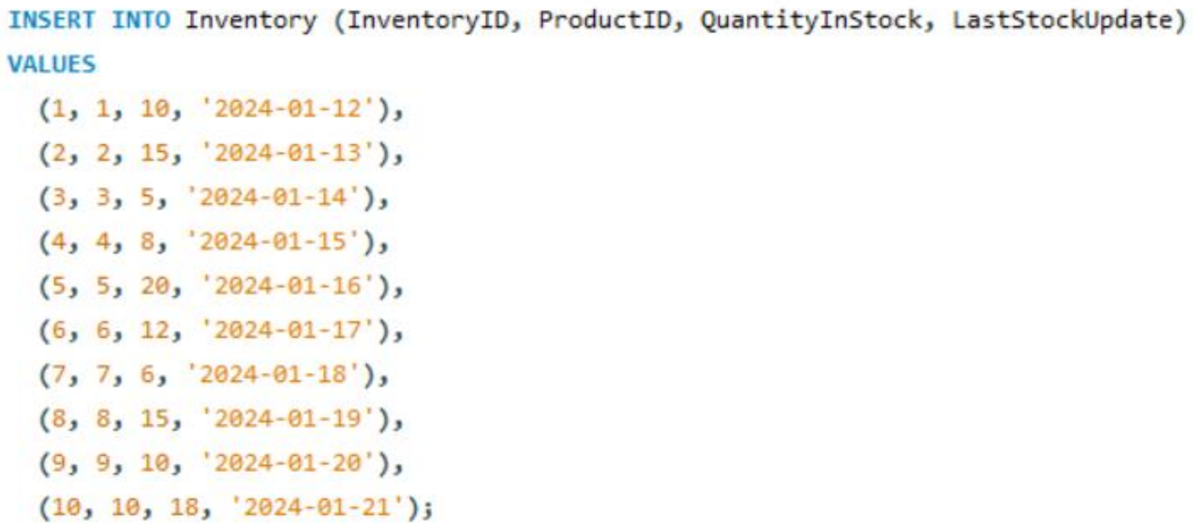


```
16:11:25  INSERT INTO OrderDetails (OrderDetailID, OrderID, Pro...  10 row(s) affected Records: 10  Duplicates: ...  0.016 sec
16:11:25  select * from Orderdetails LIMIT 0, 1000                  10 row(s) returned                               0.000 sec / 0.000 sec
```

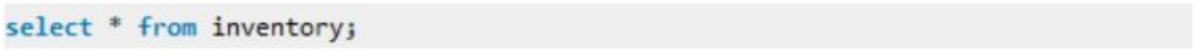


OrderDetailID	OrderID	ProductID	Quantity
1	1	1	2
2	1	2	1
3	2	3	3
4	3	4	1
5	4	5	2
6	5	6	1
7	6	7	1
8	7	8	4
9	8	9	2
10	9	10	1

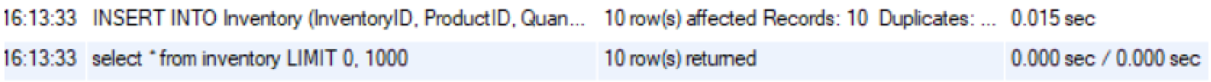
e. Inventory



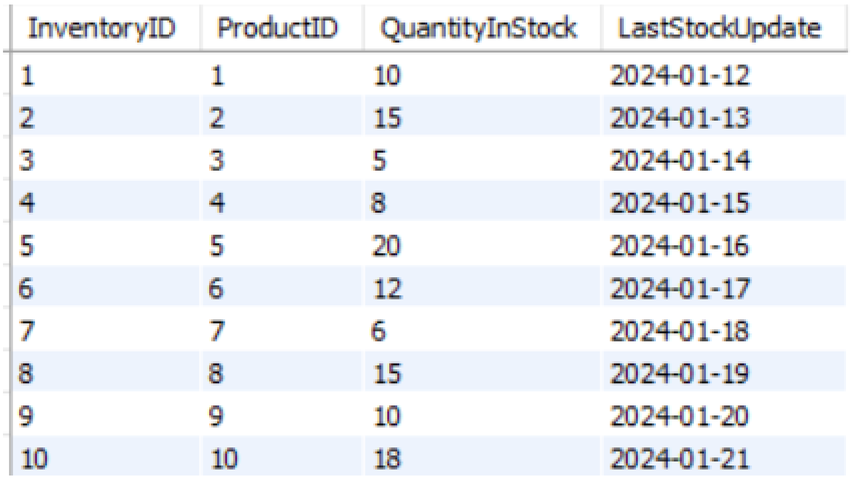
```
INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate)
VALUES
  (1, 1, 10, '2024-01-12'),
  (2, 2, 15, '2024-01-13'),
  (3, 3, 5, '2024-01-14'),
  (4, 4, 8, '2024-01-15'),
  (5, 5, 20, '2024-01-16'),
  (6, 6, 12, '2024-01-17'),
  (7, 7, 6, '2024-01-18'),
  (8, 8, 15, '2024-01-19'),
  (9, 9, 10, '2024-01-20'),
  (10, 10, 18, '2024-01-21');
```



```
select * from inventory;
```



```
16:13:33  INSERT INTO Inventory (InventoryID, ProductID, Quan...  10 row(s) affected Records: 10  Duplicates: ...  0.015 sec
16:13:33  select * from inventory LIMIT 0, 1000                  10 row(s) returned                            0.000 sec / 0.000 sec
```

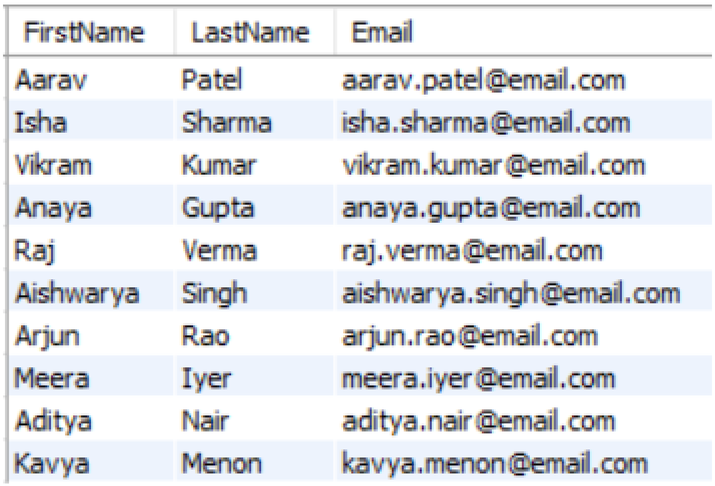


InventoryID	ProductID	QuantityInStock	LastStockUpdate
1	1	10	2024-01-12
2	2	15	2024-01-13
3	3	5	2024-01-14
4	4	8	2024-01-15
5	5	20	2024-01-16
6	6	12	2024-01-17
7	7	6	2024-01-18
8	8	15	2024-01-19
9	9	10	2024-01-20
10	10	18	2024-01-21

Task 2:

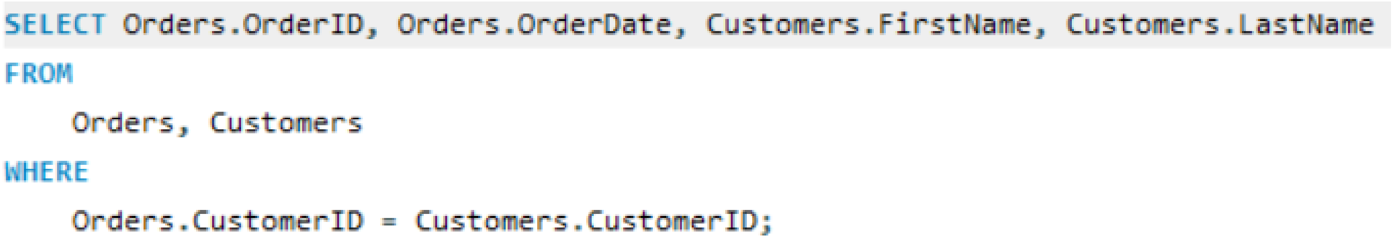
1. Write an SQL query to retrieve the names and emails of all customers.

SELECT FirstName, LastName, Email FROM Customers;

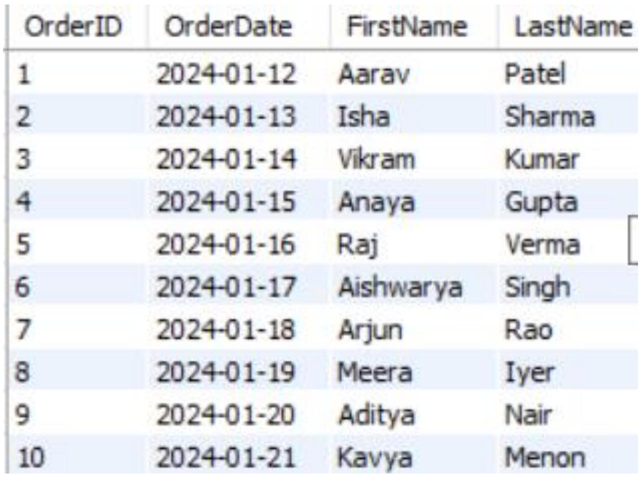


FirstName	LastName	Email
Aarav	Patel	aarav.patel@email.com
Isha	Sharma	isha.sharma@email.com
Vikram	Kumar	vikram.kumar@email.com
Anaya	Gupta	anaya.gupta@email.com
Raj	Verma	raj.verma@email.com
Aishwarya	Singh	aishwarya.singh@email.com
Arjun	Rao	arjun.rao@email.com
Meera	Iyer	meera.iyer@email.com
Aditya	Nair	aditya.nair@email.com
Kavya	Menon	kavya.menon@email.com

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

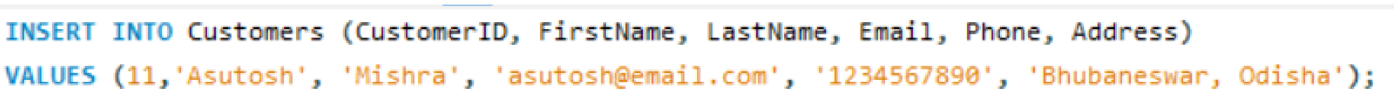


```sql
SELECT Orders.OrderID, Orders.OrderDate, Customers.FirstName, Customers.LastName
FROM
    Orders, Customers
WHERE
    Orders.CustomerID = Customers.CustomerID;
```

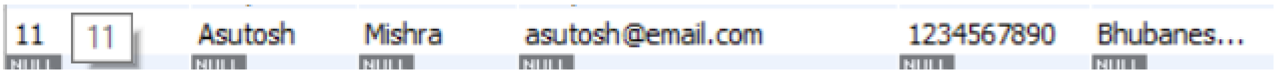


OrderID	OrderDate	FirstName	LastName
1	2024-01-12	Aarav	Patel
2	2024-01-13	Isha	Sharma
3	2024-01-14	Vikram	Kumar
4	2024-01-15	Anaya	Gupta
5	2024-01-16	Raj	Verma
6	2024-01-17	Aishwarya	Singh
7	2024-01-18	Arjun	Rao
8	2024-01-19	Meera	Iyer
9	2024-01-20	Aditya	Nair
10	2024-01-21	Kavya	Menon

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

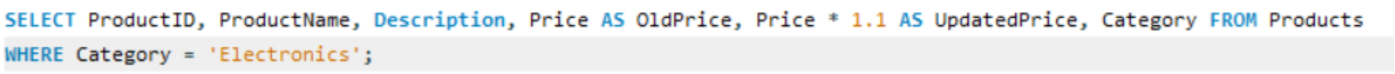


```sql
INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone, Address)
VALUES (11,'Asutosh', 'Mishra', 'asutosh@email.com', '1234567890', 'Bhubaneswar, Odisha');
```

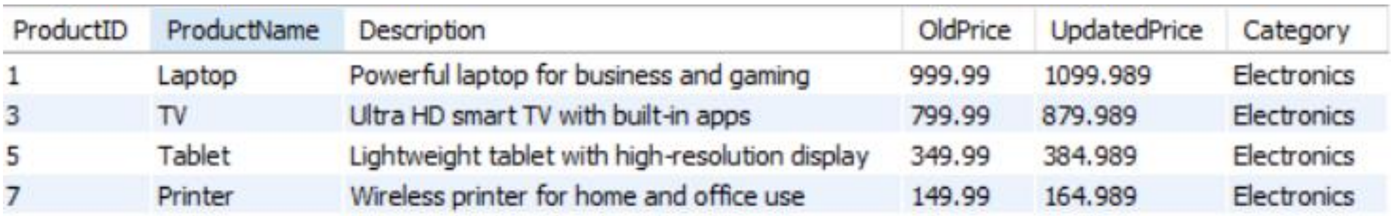


11	11	Asutosh	Mishra	asutosh@email.com	1234567890	Bhubanes...
NULL		NULL	NULL	NULL	NULL	NULL

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

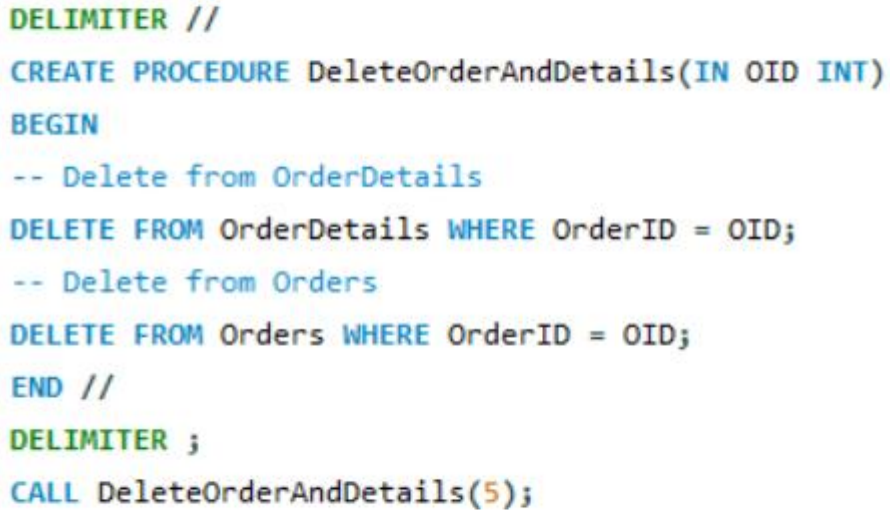


```sql
SELECT ProductID, ProductName, Description, Price AS OldPrice, Price * 1.1 AS UpdatedPrice, Category FROM Products
WHERE Category = 'Electronics';
```



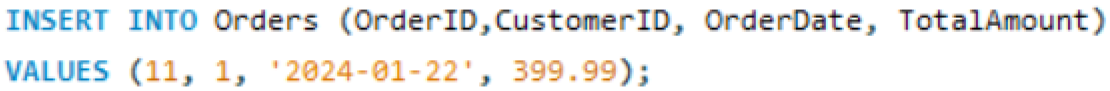
ProductID	ProductName	Description	OldPrice	UpdatedPrice	Category
1	Laptop	Powerful laptop for business and gaming	999.99	1099.989	Electronics
3	TV	Ultra HD smart TV with built-in apps	799.99	879.989	Electronics
5	Tablet	Lightweight tablet with high-resolution display	349.99	384.989	Electronics
7	Printer	Wireless printer for home and office use	149.99	164.989	Electronics

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.

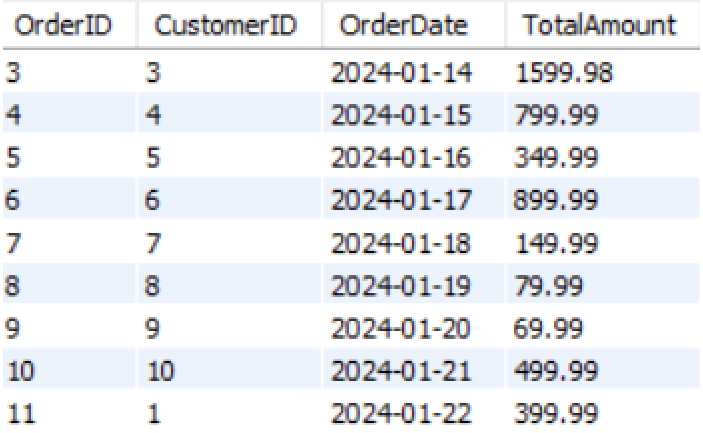


```
DELIMITER //
CREATE PROCEDURE DeleteOrderAndDetails(IN OID INT)
BEGIN
-- Delete from OrderDetails
DELETE FROM OrderDetails WHERE OrderID = OID;
-- Delete from Orders
DELETE FROM Orders WHERE OrderID = OID;
END //
DELIMITER ;
CALL DeleteOrderAndDetails(5);
```

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.

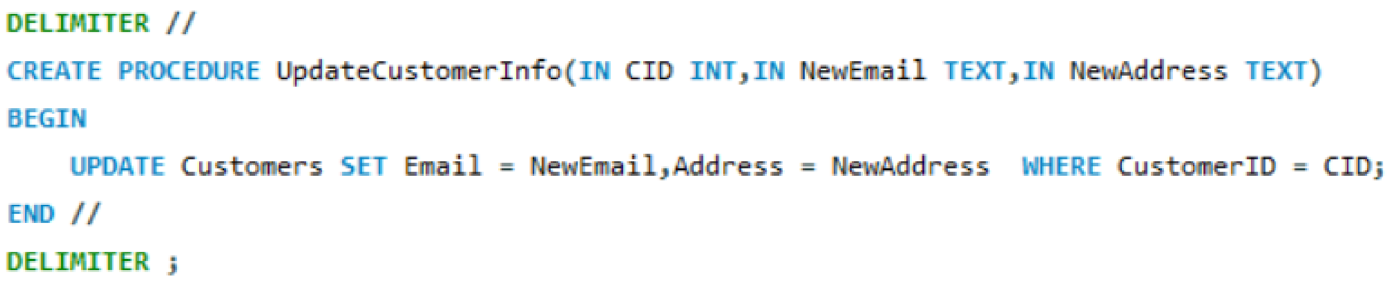


```
INSERT INTO Orders (OrderID,CustomerID, OrderDate, TotalAmount)
VALUES (11, 1, '2024-01-22', 399.99);
```

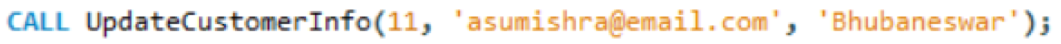


OrderID	CustomerID	OrderDate	TotalAmount
3	3	2024-01-14	1599.98
4	4	2024-01-15	799.99
5	5	2024-01-16	349.99
6	6	2024-01-17	899.99
7	7	2024-01-18	149.99
8	8	2024-01-19	79.99
9	9	2024-01-20	69.99
10	10	2024-01-21	499.99
11	1	2024-01-22	399.99

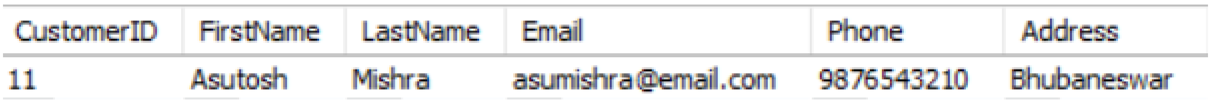
7. Write an SQL query to update the contact information (e.g., email and address) of a specific customer in the "Customers" table. Allow users to input the customer ID and new contact information.



```
DELIMITER //
CREATE PROCEDURE UpdateCustomerInfo(IN CID INT,IN NewEmail TEXT,IN NewAddress TEXT)
BEGIN
    UPDATE Customers SET Email = NewEmail,Address = NewAddress  WHERE CustomerID = CID;
END //
DELIMITER ;
```

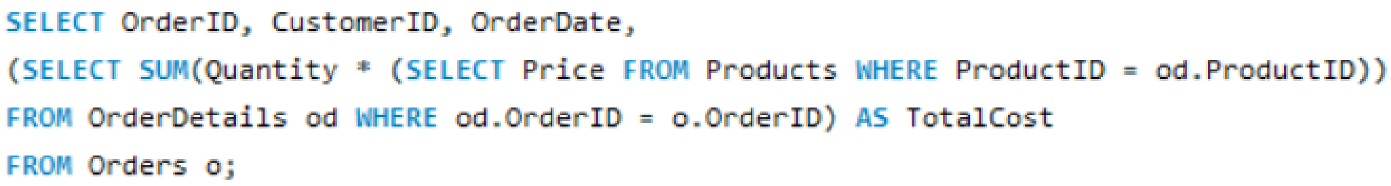


```
CALL UpdateCustomerInfo(11, 'asumishra@email.com', 'Bhubaneswar');
```

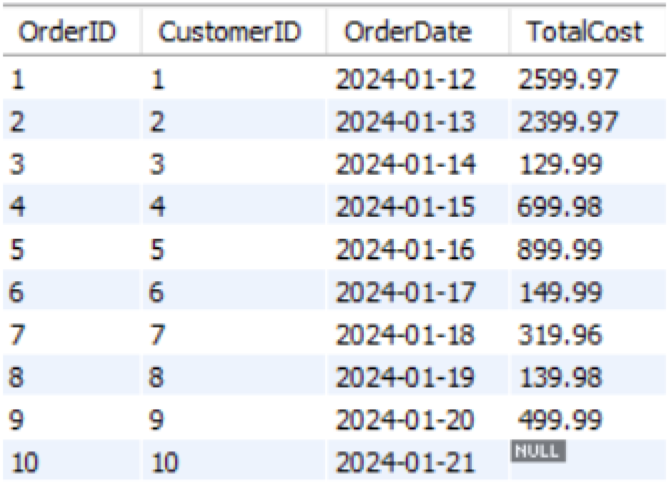


CustomerID	FirstName	LastName	Email	Phone	Address
11	Asutosh	Mishra	asumishra@email.com	9876543210	Bhubaneswar

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.

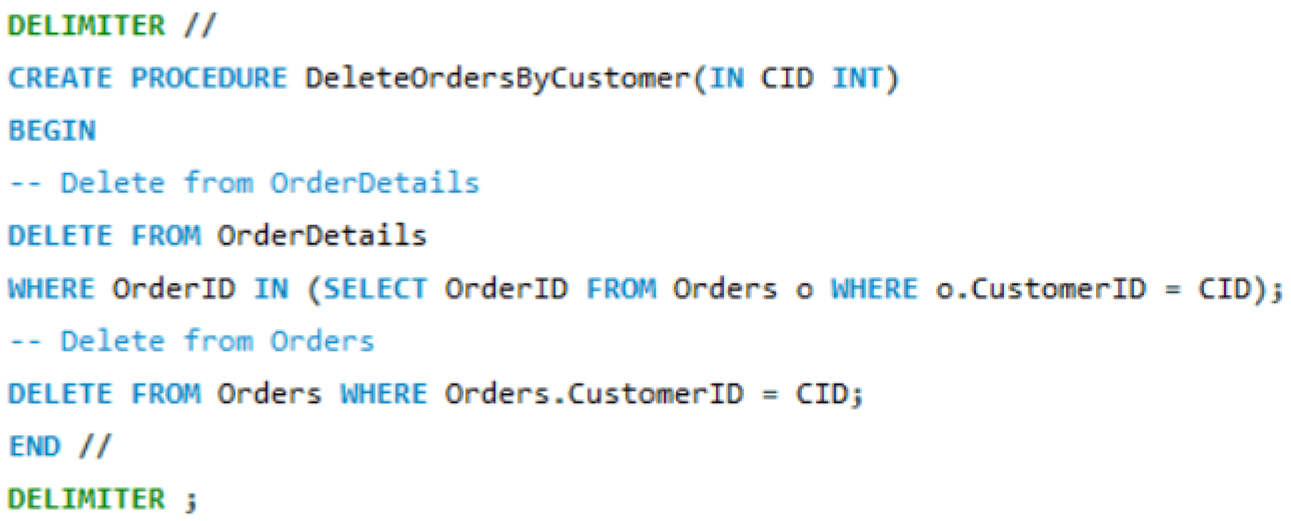


```
SELECT OrderID, CustomerID, OrderDate,
(SELECT SUM(Quantity * (SELECT Price FROM Products WHERE ProductID = od.ProductID))
FROM OrderDetails od WHERE od.OrderID = o.OrderID) AS TotalCost
FROM Orders o;
```

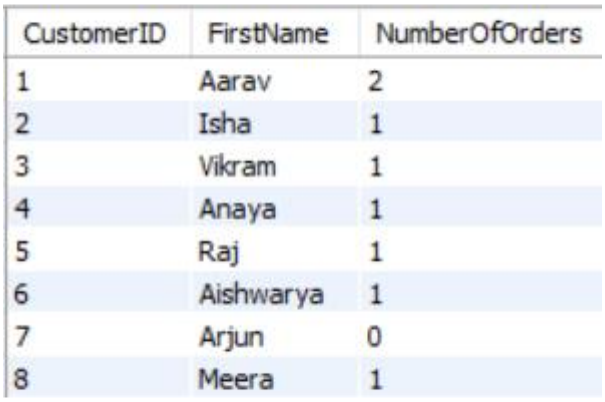


OrderID	CustomerID	OrderDate	TotalCost
1	1	2024-01-12	2599.97
2	2	2024-01-13	2399.97
3	3	2024-01-14	129.99
4	4	2024-01-15	699.98
5	5	2024-01-16	899.99
6	6	2024-01-17	149.99
7	7	2024-01-18	319.96
8	8	2024-01-19	139.98
9	9	2024-01-20	499.99
10	10	2024-01-21	NULL

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.

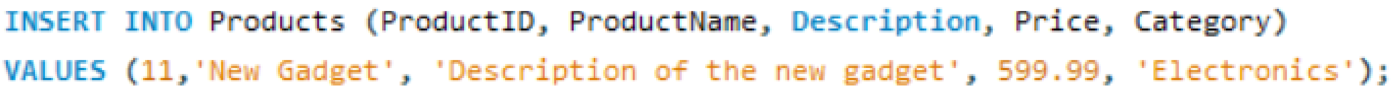


```
DELIMITER //
CREATE PROCEDURE DeleteOrdersByCustomer(IN CID INT)
BEGIN
-- Delete from OrderDetails
DELETE FROM OrderDetails
WHERE OrderID IN (SELECT OrderID FROM Orders o WHERE o.CustomerID = CID);
-- Delete from Orders
DELETE FROM Orders WHERE Orders.CustomerID = CID;
END //
DELIMITER ;
```

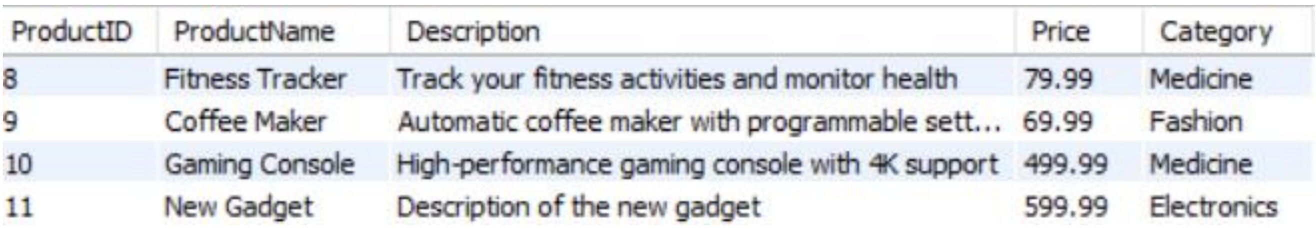


CustomerID	FirstName	NumberOfOrders
1	Aarav	2
2	Isha	1
3	Vikram	1
4	Anaya	1
5	Raj	1
6	Aishwarya	1
7	Arjun	0
8	Meera	1

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.

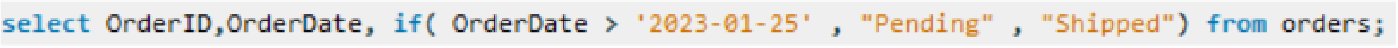


```
INSERT INTO Products (ProductID, ProductName, Description, Price, Category)
VALUES (11,'New Gadget', 'Description of the new gadget', 599.99, 'Electronics');
```

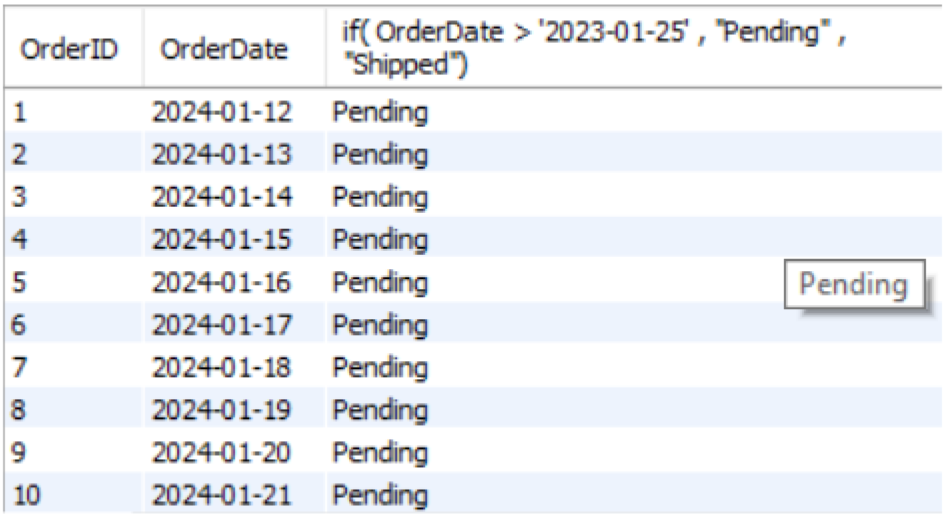


ProductID	ProductName	Description	Price	Category
8	Fitness Tracker	Track your fitness activities and monitor health	79.99	Medicine
9	Coffee Maker	Automatic coffee maker with programmable sett...	69.99	Fashion
10	Gaming Console	High-performance gaming console with 4K support	499.99	Medicine
11	New Gadget	Description of the new gadget	599.99	Electronics

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.

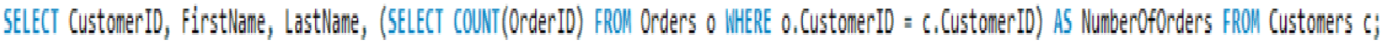


```
select OrderID,OrderDate, if( OrderDate > '2023-01-25' , "Pending" , "Shipped") from orders;
```

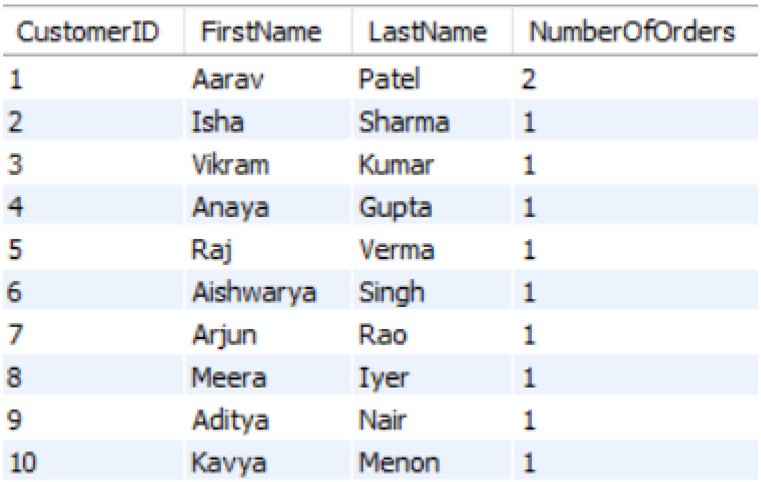


OrderID	OrderDate	if(OrderDate > '2023-01-25' , "Pending" , "Shipped")
1	2024-01-12	Pending
2	2024-01-13	Pending
3	2024-01-14	Pending
4	2024-01-15	Pending
5	2024-01-16	Pending
6	2024-01-17	Pending
7	2024-01-18	Pending
8	2024-01-19	Pending
9	2024-01-20	Pending
10	2024-01-21	Pending

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.



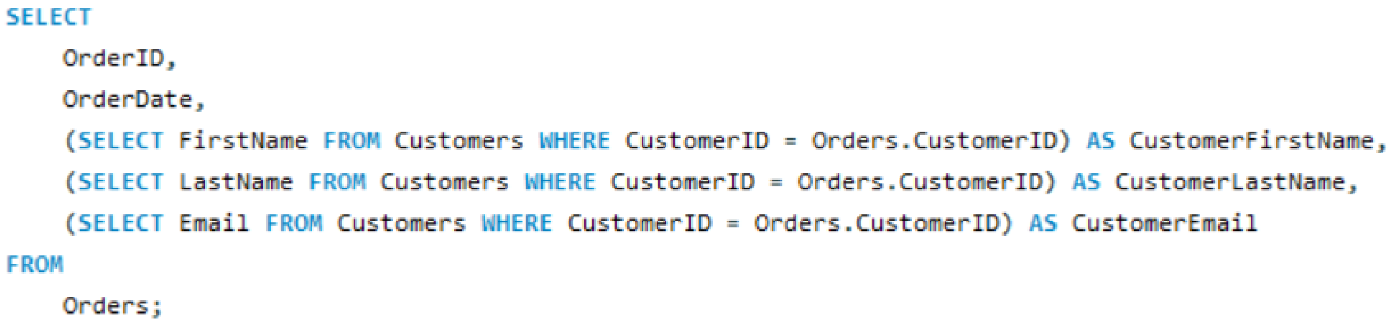
```
SELECT CustomerID, FirstName, LastName, (SELECT COUNT(OrderID) FROM Orders o WHERE o.CustomerID = c.CustomerID) AS NumberOfOrders FROM Customers c;
```



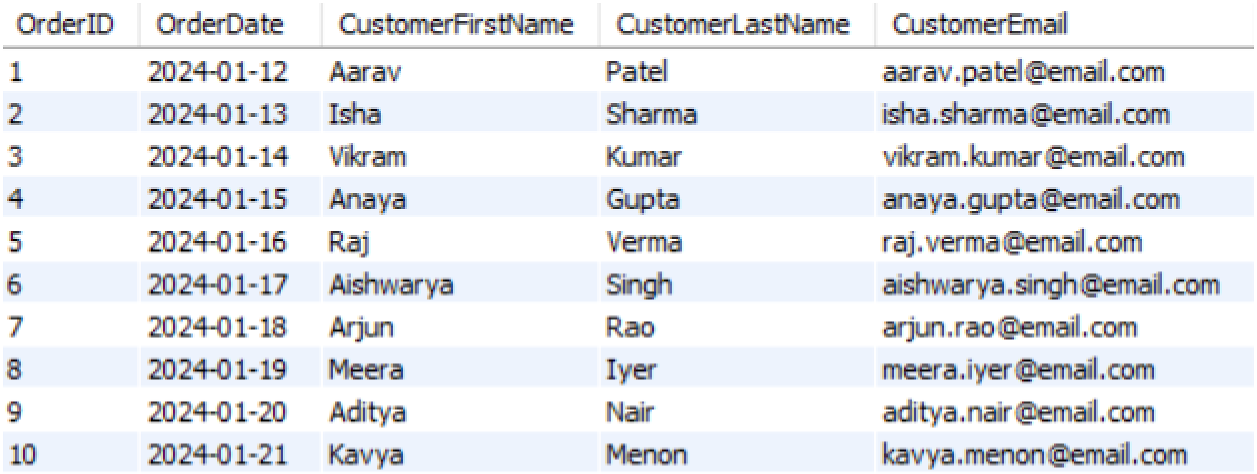
CustomerID	FirstName	LastName	NumberOfOrders
1	Aarav	Patel	2
2	Isha	Sharma	1
3	Vikram	Kumar	1
4	Anaya	Gupta	1
5	Raj	Verma	1
6	Aishwarya	Singh	1
7	Arjun	Rao	1
8	Meera	Iyer	1
9	Aditya	Nair	1
10	Kavya	Menon	1

Task 3:

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



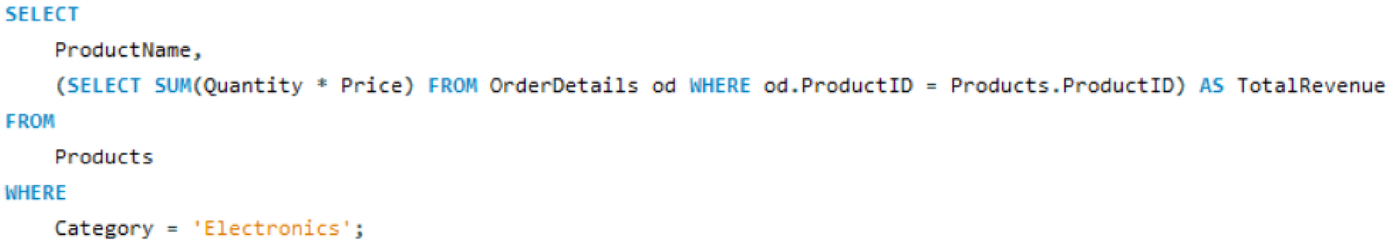
```
SELECT
    OrderID,
    OrderDate,
    (SELECT FirstName FROM Customers WHERE CustomerID = Orders.CustomerID) AS CustomerFirstName,
    (SELECT LastName FROM Customers WHERE CustomerID = Orders.CustomerID) AS CustomerLastName,
    (SELECT Email FROM Customers WHERE CustomerID = Orders.CustomerID) AS CustomerEmail
FROM
    Orders;
```



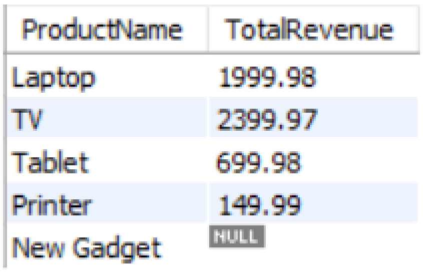
OrderID	OrderDate	CustomerFirstName	CustomerLastName	CustomerEmail
1	2024-01-12	Aarav	Patel	aarav.patel@email.com
2	2024-01-13	Isha	Sharma	isha.sharma@email.com
3	2024-01-14	Vikram	Kumar	vikram.kumar@email.com
4	2024-01-15	Anaya	Gupta	anaya.gupta@email.com
5	2024-01-16	Raj	Verma	raj.verma@email.com
6	2024-01-17	Aishwarya	Singh	aishwarya.singh@email.com
7	2024-01-18	Arjun	Rao	arjun.rao@email.com
8	2024-01-19	Meera	Iyer	meera.iyer@email.com
9	2024-01-20	Aditya	Nair	aditya.nair@email.com
10	2024-01-21	Kavya	Menon	kavya.menon@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.



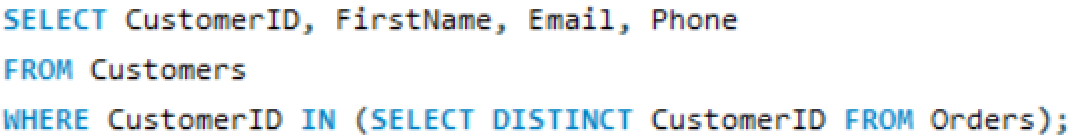
```
SELECT
    ProductName,
    (SELECT SUM(Quantity * Price) FROM OrderDetails od WHERE od.ProductID = Products.ProductID) AS TotalRevenue
FROM
    Products
WHERE
    Category = 'Electronics';
```



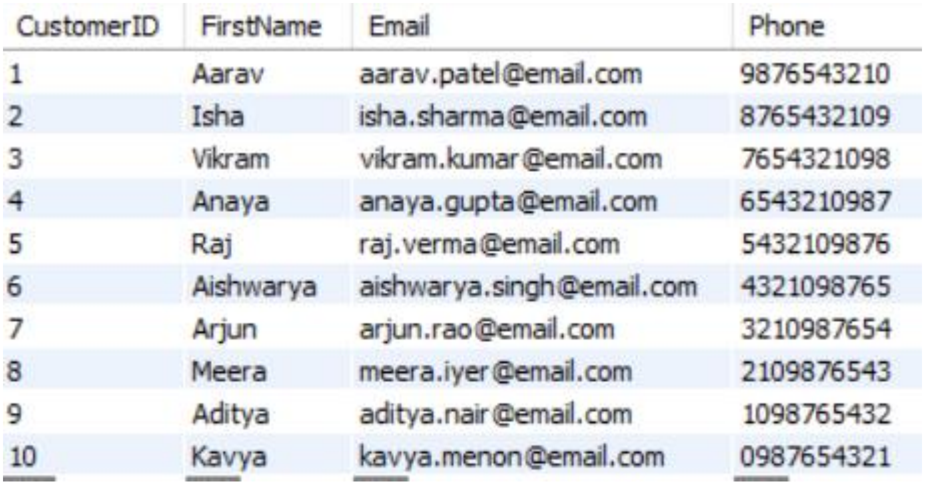
ProductName	TotalRevenue
Laptop	1999.98
TV	2399.97
Tablet	699.98
Printer	149.99
New Gadget	NULL

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

names and contact information.

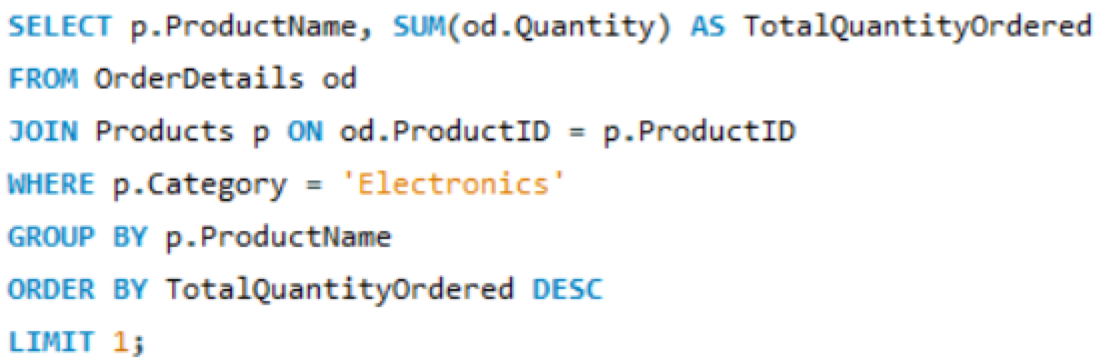


```
SELECT CustomerID, FirstName, Email, Phone
FROM Customers
WHERE CustomerID IN (SELECT DISTINCT CustomerID FROM Orders);
```

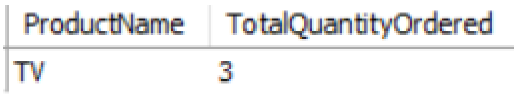


CustomerID	FirstName	Email	Phone
1	Aarav	aarav.patel@email.com	9876543210
2	Isha	isha.sharma@email.com	8765432109
3	Vikram	vikram.kumar@email.com	7654321098
4	Anaya	anaya.gupta@email.com	6543210987
5	Raj	raj.verma@email.com	5432109876
6	Aishwarya	aishwarya.singh@email.com	4321098765
7	Arjun	arjun.rao@email.com	3210987654
8	Meera	meera.iyer@email.com	2109876543
9	Aditya	aditya.nair@email.com	1098765432
10	Kavya	kavya.menon@email.com	0987654321

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.

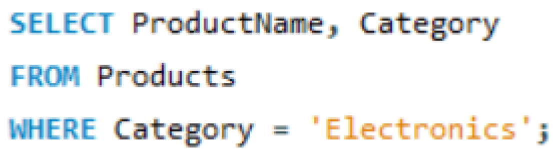


```
SELECT p.ProductName, SUM(od.Quantity) AS TotalQuantityOrdered
FROM OrderDetails od
JOIN Products p ON od.ProductID = p.ProductID
WHERE p.Category = 'Electronics'
GROUP BY p.ProductName
ORDER BY TotalQuantityOrdered DESC
LIMIT 1;
```

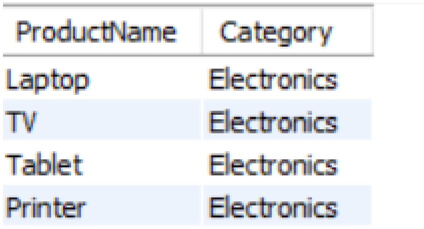


ProductName	TotalQuantityOrdered
TV	3

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

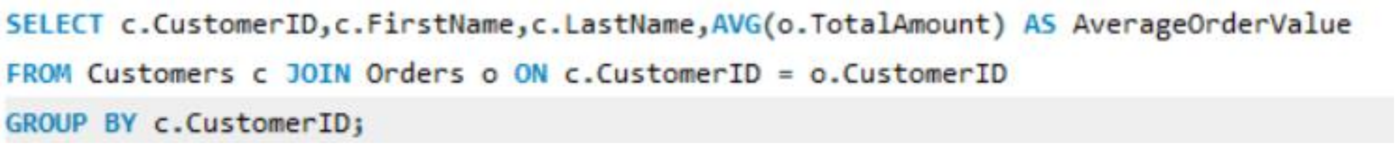


```
SELECT ProductName, Category
FROM Products
WHERE Category = 'Electronics';
```

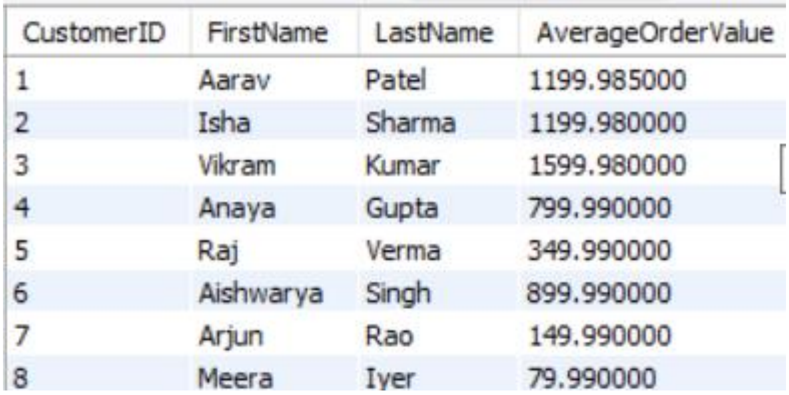


ProductName	Category
Laptop	Electronics
TV	Electronics
Tablet	Electronics
Printer	Electronics

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

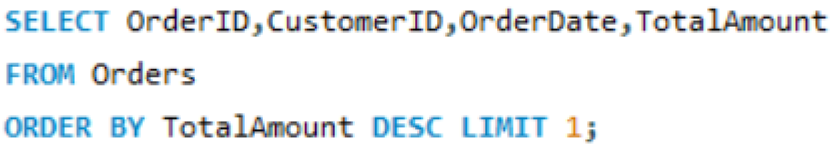


```
SELECT c.CustomerID,c.FirstName,c.LastName,AVG(o.TotalAmount) AS AverageOrderValue
FROM Customers c JOIN Orders o ON c.CustomerID = o.CustomerID
GROUP BY c.CustomerID;
```

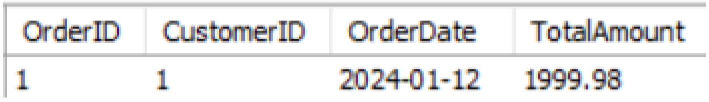


CustomerID	FirstName	LastName	AverageOrderValue
1	Aarav	Patel	1199.985000
2	Isha	Sharma	1199.980000
3	Vikram	Kumar	1599.980000
4	Anaya	Gupta	799.990000
5	Raj	Verma	349.990000
6	Aishwarya	Singh	899.990000
7	Arjun	Rao	149.990000
8	Meera	Iyer	79.990000

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

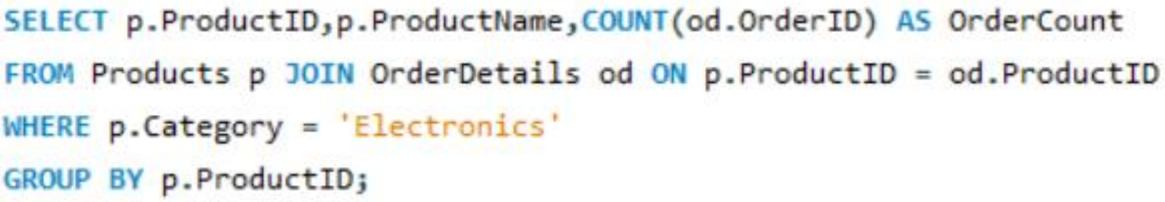


```
SELECT OrderID,CustomerID,OrderDate,TotalAmount
FROM Orders
ORDER BY TotalAmount DESC LIMIT 1;
```

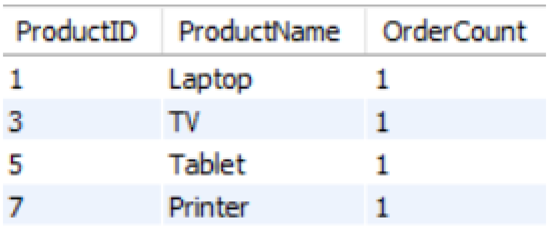


OrderID	CustomerID	OrderDate	TotalAmount
1	1	2024-01-12	1999.98

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

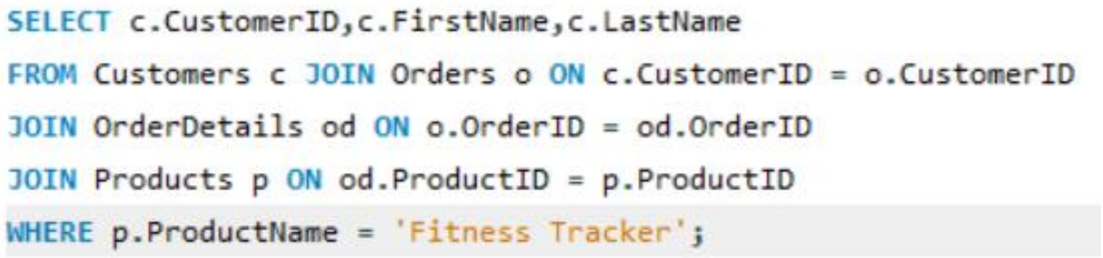


```
SELECT p.ProductID,p.ProductName,COUNT(od.OrderID) AS OrderCount
FROM Products p JOIN OrderDetails od ON p.ProductID = od.ProductID
WHERE p.Category = 'Electronics'
GROUP BY p.ProductID;
```

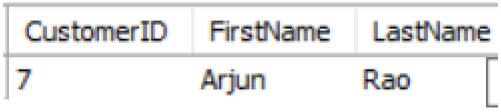


ProductID	ProductName	OrderCount
1	Laptop	1
3	TV	1
5	Tablet	1
7	Printer	1

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.

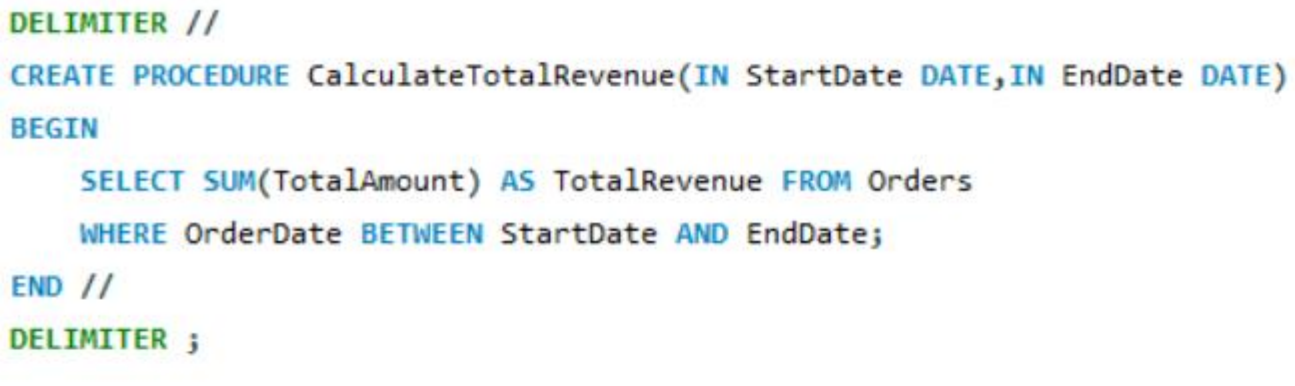


```
SELECT c.CustomerID,c.FirstName,c.LastName
FROM Customers c JOIN Orders o ON c.CustomerID = o.CustomerID
JOIN OrderDetails od ON o.OrderID = od.OrderID
JOIN Products p ON od.ProductID = p.ProductID
WHERE p.ProductName = 'Fitness Tracker';
```

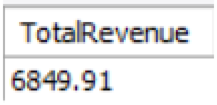


CustomerID	FirstName	LastName
7	Arjun	Rao

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.



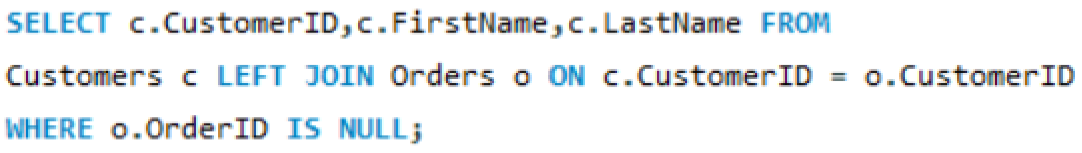
```
DELIMITER //
CREATE PROCEDURE CalculateTotalRevenue(IN StartDate DATE,IN EndDate DATE)
BEGIN
    SELECT SUM(TotalAmount) AS TotalRevenue FROM Orders
    WHERE OrderDate BETWEEN StartDate AND EndDate;
END //
DELIMITER ;
```



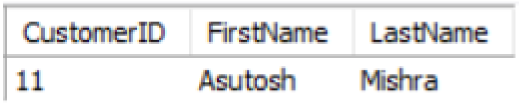
TotalRevenue
6849.91

Task 4:

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

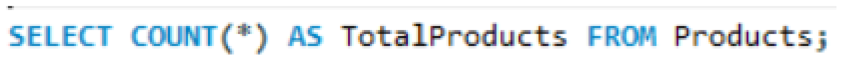


```
SELECT c.CustomerID,c.FirstName,c.LastName FROM
Customers c LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
WHERE o.OrderID IS NULL;
```

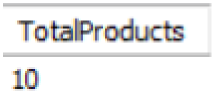


CustomerID	FirstName	LastName
11	Asutosh	Mishra

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

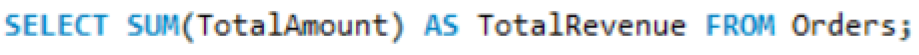


```
SELECT COUNT(*) AS TotalProducts FROM Products;
```

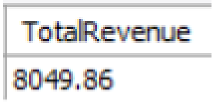


TotalProducts
10

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

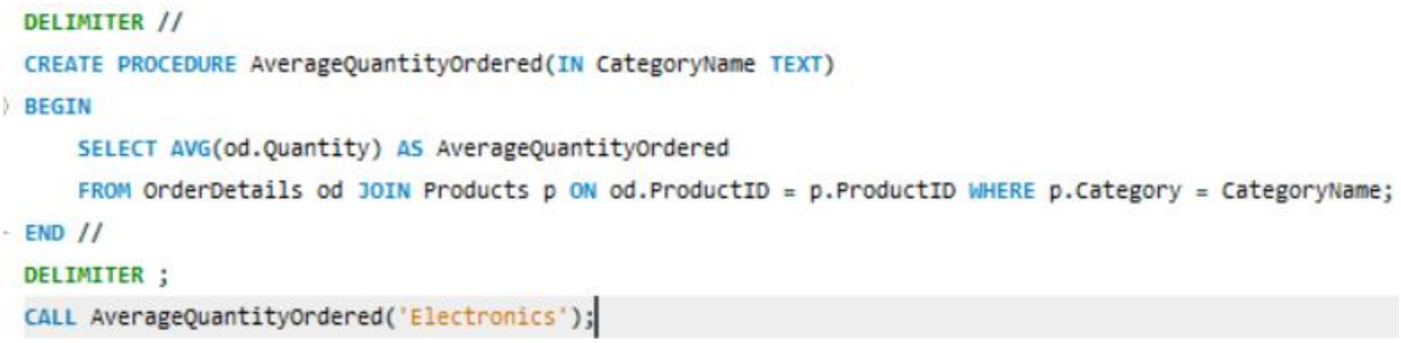


```
SELECT SUM(TotalAmount) AS TotalRevenue FROM Orders;
```

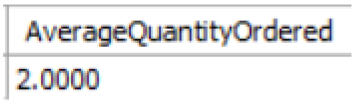


TotalRevenue
8049.86

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.

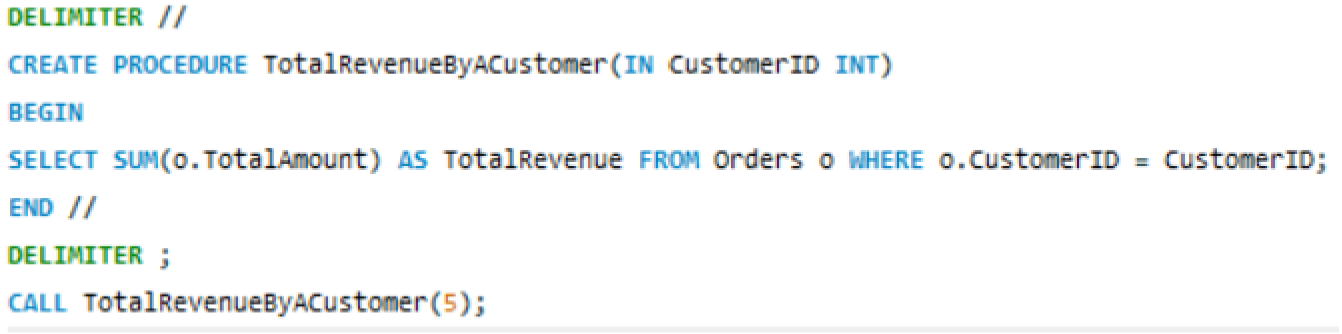


```
DELIMITER //
CREATE PROCEDURE AverageQuantityOrdered(IN CategoryName TEXT)
BEGIN
    SELECT AVG(od.Quantity) AS AverageQuantityOrdered
    FROM OrderDetails od JOIN Products p ON od.ProductID = p.ProductID WHERE p.Category = CategoryName;
END //
DELIMITER ;
CALL AverageQuantityOrdered('Electronics');
```

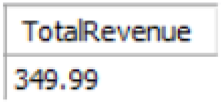


AverageQuantityOrdered
2.0000

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.

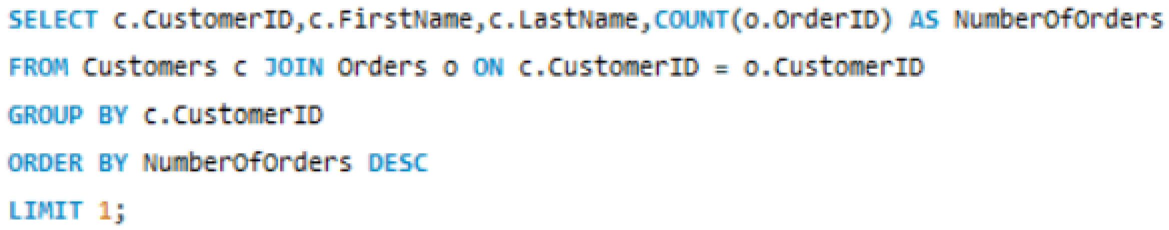


```
DELIMITER //
CREATE PROCEDURE TotalRevenueByACustomer(IN CustomerID INT)
BEGIN
SELECT SUM(o.TotalAmount) AS TotalRevenue FROM Orders o WHERE o.CustomerID = CustomerID;
END //
DELIMITER ;
CALL TotalRevenueByACustomer(5);
```

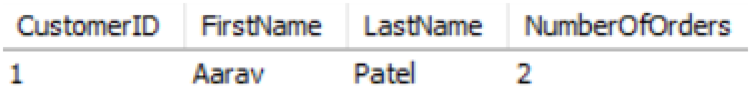


TotalRevenue
349.99

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.

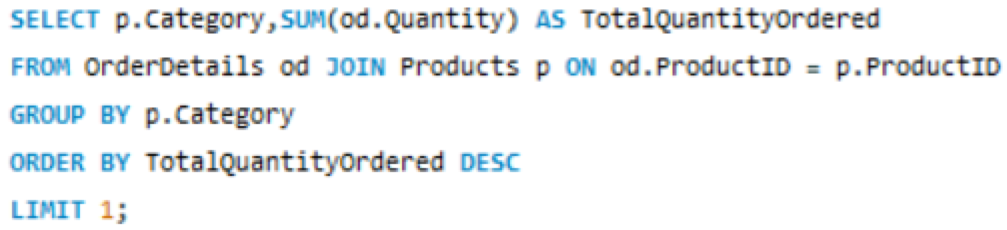


```
SELECT c.CustomerID,c.FirstName,c.LastName,COUNT(o.OrderID) AS NumberOfOrders
FROM Customers c JOIN Orders o ON c.CustomerID = o.CustomerID
GROUP BY c.CustomerID
ORDER BY NumberOfOrders DESC
LIMIT 1;
```

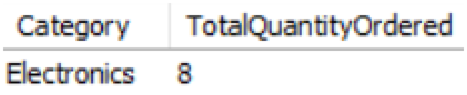


CustomerID	FirstName	LastName	NumberOfOrders
1	Aarav	Patel	2

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.

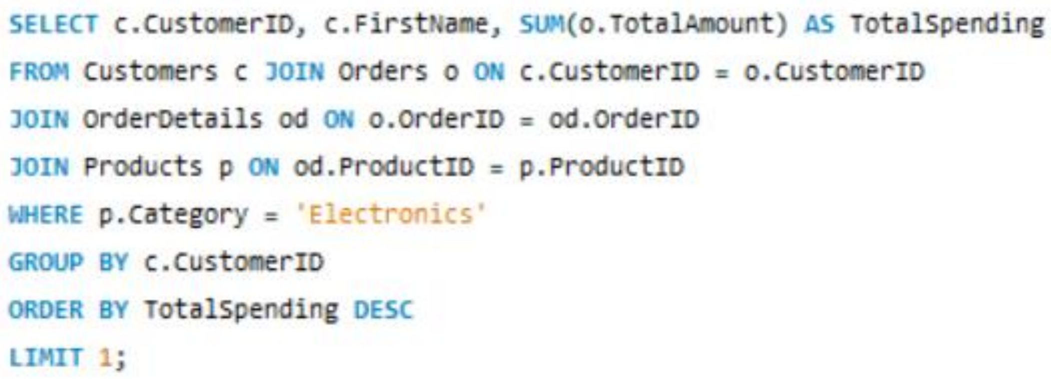


```
SELECT p.Category,SUM(od.Quantity) AS TotalQuantityOrdered
FROM OrderDetails od JOIN Products p ON od.ProductID = p.ProductID
GROUP BY p.Category
ORDER BY TotalQuantityOrdered DESC
LIMIT 1;
```

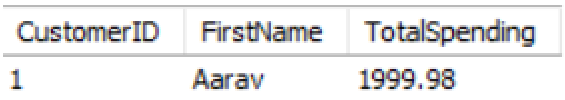


Category	TotalQuantityOrdered
Electronics	8

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.

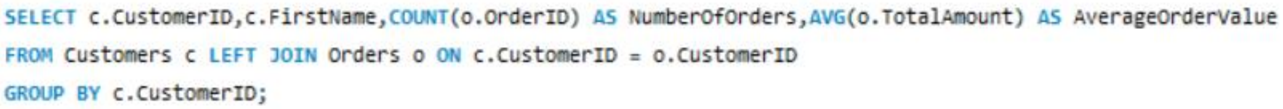


```
SELECT c.CustomerID, c.FirstName, SUM(o.TotalAmount) AS TotalSpending
FROM Customers c JOIN Orders o ON c.CustomerID = o.CustomerID
JOIN OrderDetails od ON o.OrderID = od.OrderID
JOIN Products p ON od.ProductID = p.ProductID
WHERE p.Category = 'Electronics'
GROUP BY c.CustomerID
ORDER BY TotalSpending DESC
LIMIT 1;
```

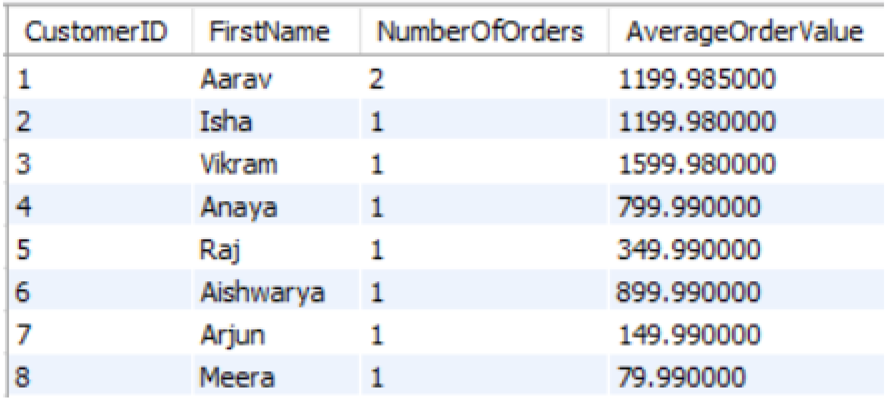


CustomerID	FirstName	TotalSpending
1	Aarav	1999.98

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

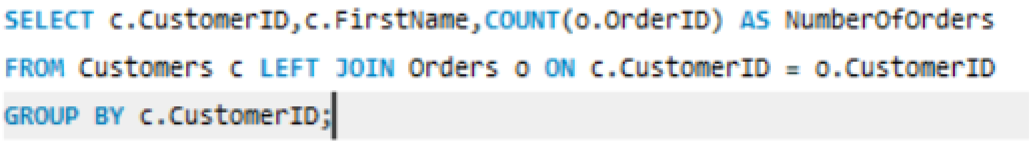


```
SELECT c.CustomerID,c.FirstName,COUNT(o.OrderID) AS NumberOfOrders,AVG(o.TotalAmount) AS AverageOrderValue
FROM Customers c LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
GROUP BY c.CustomerID;
```

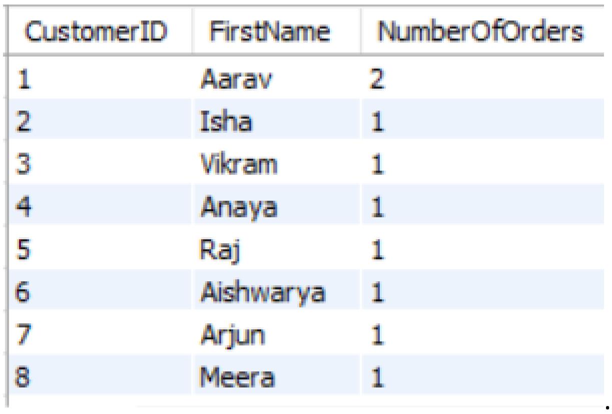


CustomerID	FirstName	NumberOfOrders	AverageOrderValue
1	Aarav	2	1199.985000
2	Isha	1	1199.980000
3	Vikram	1	1599.980000
4	Anaya	1	799.990000
5	Raj	1	349.990000
6	Aishwarya	1	899.990000
7	Arjun	1	149.990000
8	Meera	1	79.990000

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,c.FirstName,COUNT(o.OrderID) AS NumberOfOrders
FROM Customers c LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
GROUP BY c.CustomerID;
```



CustomerID	FirstName	NumberOfOrders
1	Aarav	2
2	Isha	1
3	Vikram	1
4	Anaya	1
5	Raj	1
6	Aishwarya	1
7	Arjun	1
8	Meera	1