**Sivaprakas B M – Hexaware Python Batch 2 – Assignment 1**

**TASK 1:**

Create the database named "TechShop"

**CREATE DATABASE TechShop;**

****

-- Use the database

**USE TechShop;**

****

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

-- Create Table Customers

**CREATE TABLE Customers (**

**CustomerID INT PRIMARY KEY,**

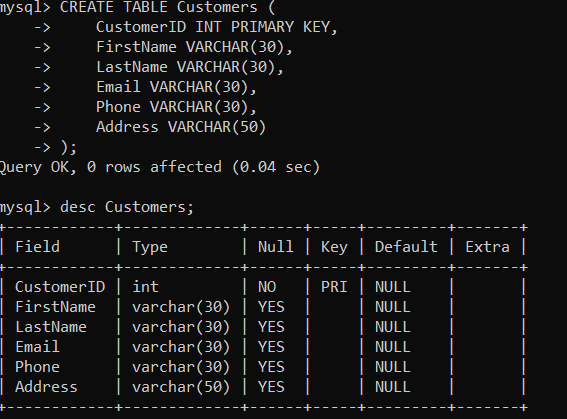
**FirstName VARCHAR(30),**

**LastName VARCHAR(30),**

**Email VARCHAR(30),**

**Phone VARCHAR(30),**

**Address VARCHAR(50) };**

****

-- Create Table Products

**CREATE TABLE Products (**

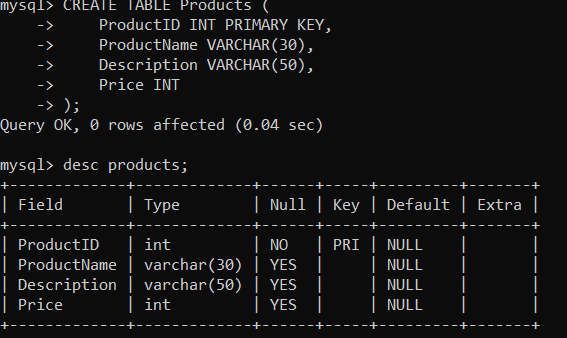
**ProductID INT PRIMARY KEY,**

**ProductName VARCHAR(30),**

**Description VARCHAR(50),**

**Price INT**

**);**

****

-- Create Table Orders

**CREATE TABLE Orders (**

**OrderID INT PRIMARY KEY,**

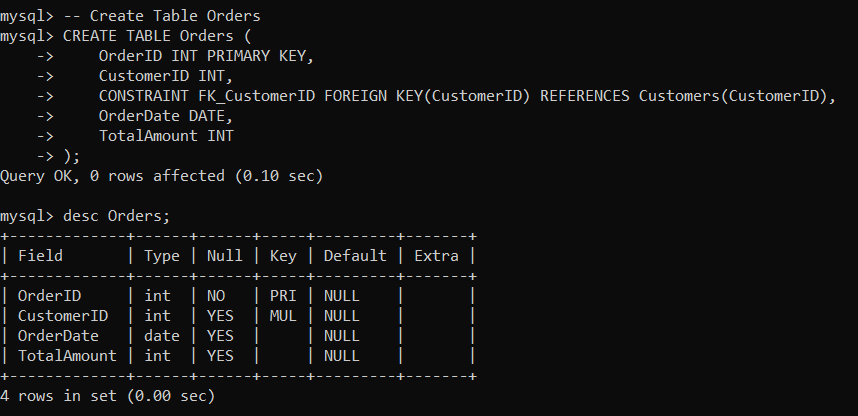
**CustomerID INT,**

**CONSTRAINT FK\_CustomerID FOREIGN KEY(CustomerID) REFERENCES Customers(CustomerID),**

**OrderDate DATE,**

**TotalAmount INT**

**);**

****

-- Create Table OrderDetails

**CREATE TABLE OrderDetails (**

**OrderDetailID INT PRIMARY KEY,**

**OrderID INT,**

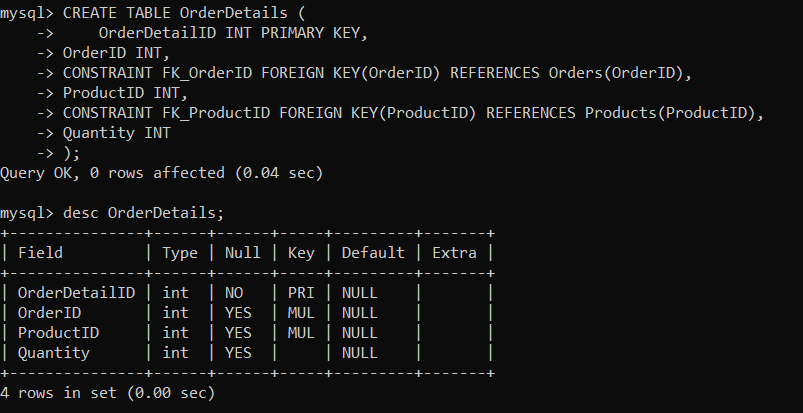
**CONSTRAINT FK\_OrderID FOREIGN KEY(OrderID) REFERENCES Orders(OrderID),**

**ProductID INT,**

**CONSTRAINT FK\_ProductID FOREIGN KEY(ProductID) REFERENCES Products(ProductID),**

**Quantity INT**

**);**

****

-- Create Table Inventory

**CREATE TABLE Inventory (**

**InventoryID INT PRIMARY KEY,**

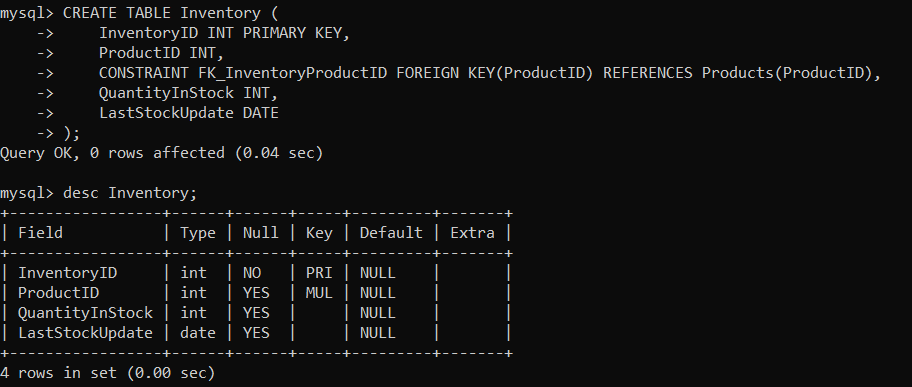
**ProductID INT,**

**CONSTRAINT FK\_InventoryProductID FOREIGN KEY(ProductID) REFERENCES Products(ProductID),**

**QuantityInStock INT,**

**LastStockUpdate DATE**

**);**

****

-- Insert Values into table Customers

**INSERT INTO Customers VALUES**

**(1, 'Rahul', 'Sharma', 'rahul.sharma@example.com', '5432109876', '5th North St, Kolkata'),**

**(2, 'Priya', 'Patel', 'priya.patel@example.com', '8765432109', '2nd West St, Mumbai'),**

**(3, 'Amit', 'Singh', 'amit.singh@example.com', '1098765432', '9th Garden St, Ahmedabad'),**

**(4, 'Neha', 'Kumar', 'neha.kumar@example.com', '3210987654', '7th Mount St, Pune'),**

**(5, 'Ananya', 'Gupta', 'ananya.gupta@example.com', '9876543210', '1st Main St, Bangalore'),**

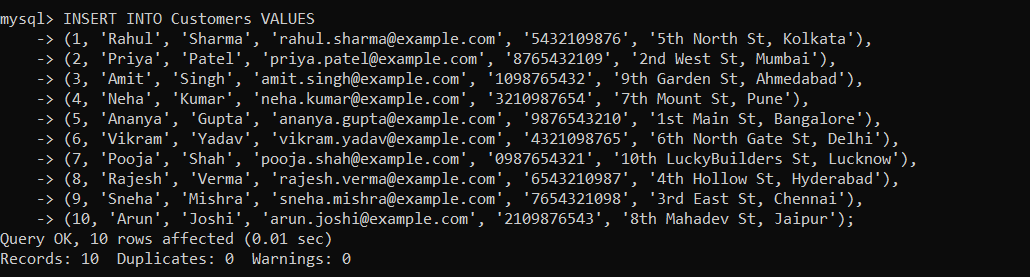
**(6, 'Vikram', 'Yadav', 'vikram.yadav@example.com', '4321098765', '6th North Gate St, Delhi'),**

**(7, 'Pooja', 'Shah', 'pooja.shah@example.com', '0987654321', '10th LuckyBuilders St, Lucknow'),**

**(8, 'Rajesh', 'Verma', 'rajesh.verma@example.com', '6543210987', '4th Hollow St, Hyderabad'),**

**(9, 'Sneha', 'Mishra', 'sneha.mishra@example.com', '7654321098', '3rd East St, Chennai'),**

**(10, 'Arun', 'Joshi', 'arun.joshi@example.com', '2109876543', '8th Mahadev St, Jaipur');**



-- Insert Values into table Products

**INSERT INTO Products (ProductID, ProductName, Description, Price) VALUES**

**(1, 'Smartphone', 'Latest smartphone model', 800),**

**(2, 'Smartwatch', 'Fitness and smart features', 300),**

**(3, 'Tablet', 'Compact tablet device', 500),**

**(4, 'Laptop', 'High-performance laptop', 1000),**

**(5, 'Camera', 'Digital camera with advanced features', 700),**

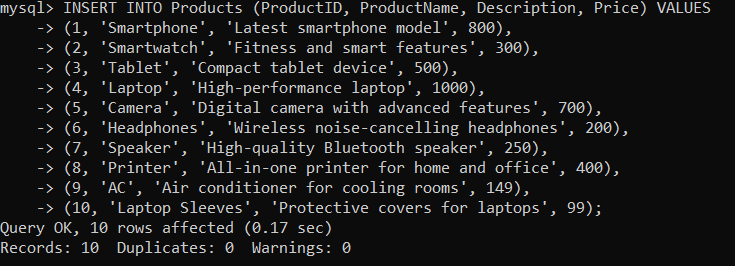
**(6, 'Headphones', 'Wireless noise-cancelling headphones', 200),**

**(7, 'Speaker', 'High-quality Bluetooth speaker', 250),**

**(8, 'Printer', 'All-in-one printer for home and office', 400),**

**(9, 'AC', 'Air conditioner for cooling rooms', 149),**

**(10, 'Laptop Sleeves', 'Protective covers for laptops', 99);**

****

-- Insert Values into table Orders

**INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount, Status) VALUES**

**(100, 1, '2024-04-01', 1200, 'Pending'),**

**(101, 2, '2024-04-02', 900, 'Shipped'),**

**(102, 3, '2024-04-03', 600, 'Pending'),**

**(103, 4, '2024-04-04', 1500, 'Shipped'),**

**(104, 5, '2024-04-05', 400, 'Pending'),**

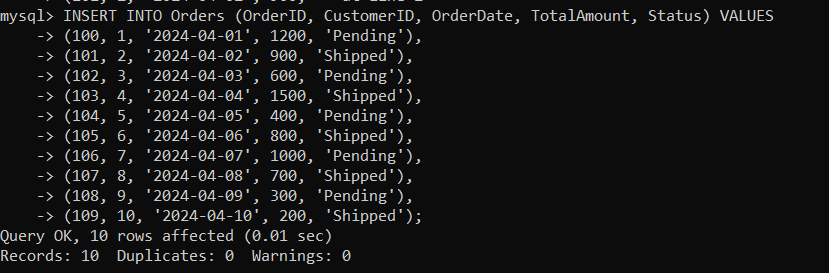
**(105, 6, '2024-04-06', 800, 'Shipped'),**

**(106, 7, '2024-04-07', 1000, 'Pending'),**

**(107, 8, '2024-04-08', 700, 'Shipped'),**

**(108, 9, '2024-04-09', 300, 'Pending'),**

**(109, 10, '2024-04-10', 200, 'Shipped');**

****

-- Insert Values into table OrderDetails

**INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity) VALUES**

**(200, 100, 1, 2),**

**(201, 100, 3, 1),**

**(202, 101, 2, 1),**

**(203, 102, 4, 3),**

**(204, 103, 1, 1),**

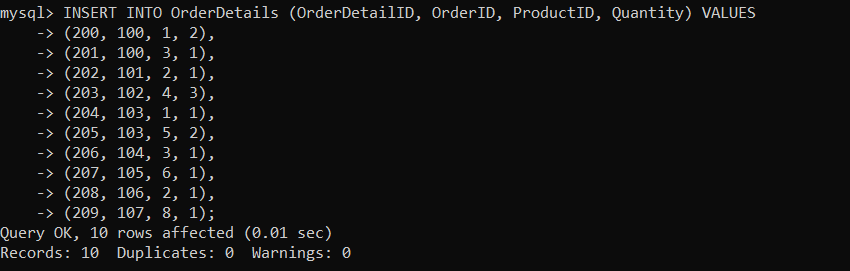
**(205, 103, 5, 2),**

**(206, 104, 3, 1),**

**(207, 105, 6, 1),**

**(208, 106, 2, 1),**

**(209, 107, 8, 1);**



-- Insert Values into table Inventory

**INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate) VALUES**

**(300, 1, 20, '2024-04-01'),**

**(301, 2, 15, '2024-04-02'),**

**(302, 3, 10, '2024-04-03'),**

**(303, 4, 5, '2024-04-04'),**

**(304, 5, 30, '2024-04-05'),**

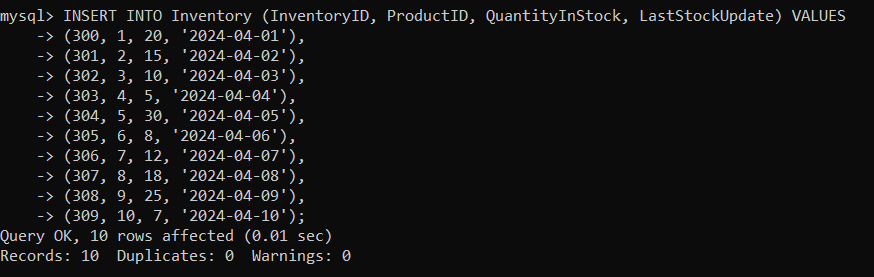
**(305, 6, 8, '2024-04-06'),**

**(306, 7, 12, '2024-04-07'),**

**(307, 8, 18, '2024-04-08'),**

**(308, 9, 25, '2024-04-09'),**

**(309, 10, 7, '2024-04-10');**



**Task 2:**

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

**SELECT CONCAT(FirstName,'' ,LastName) AS CustomerName , Email FROM Customers;**

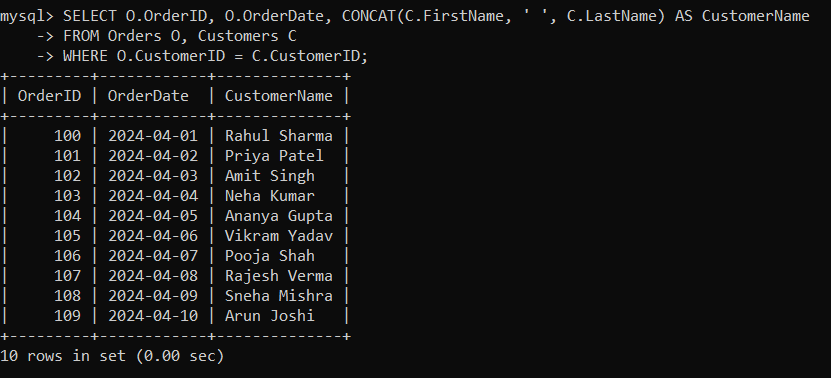


-- Write an SQL query to list all orders with their order dates and corresponding customer names.

**SELECT O.OrderID, O.OrderDate, CONCAT(C.FirstName, ' ', C.LastName) AS CustomerName**

**FROM Orders O, Customers C**

**WHERE O.CustomerID = C.CustomerID;**

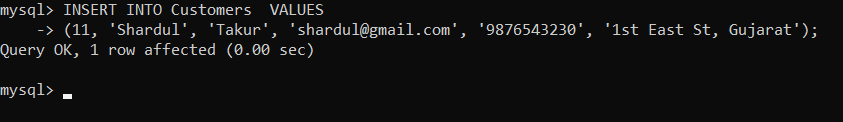


-- Write an SQL query to insert a new customer record into the "Customers" table. Include

customer information such as name, email, and address.

**INSERT INTO Customers VALUES**

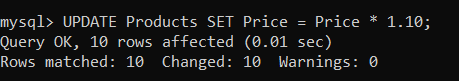
**(11, 'Shardul', 'Takur', 'shardul@gmail.com', '9876543230', '1st East St, Gujarat');**

****

-- 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;**

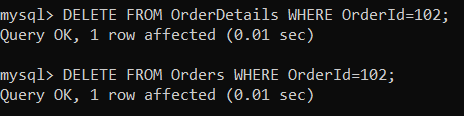
****

-- 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 FROM OrderDetails WHERE OrderId=102;**

**DELETE FROM Orders WHERE OrderId=102;**

****

-- Write an SQL query to insert a new order into the "Orders" table. Include the customer ID,

order date, and any other necessary information.

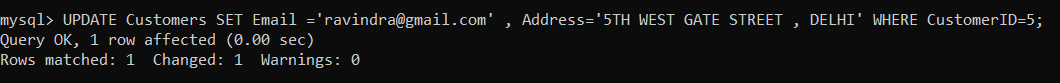
**INSERT INTO Orders VALUES(102, 3, '2024-04-03', 600, 'Pending');**



-- 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 ='ravindra@gmail.com' , Address='5TH WEST GATE STREET , DELHI' WHERE CustomerID=5;**



.

-- 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 Orders WHERE CustomerID = 3;**



-- 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.

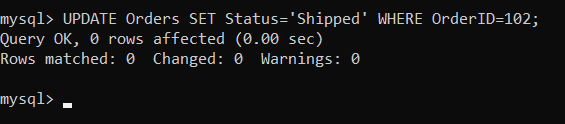
**INSERT INTO Products VALUES (11, 'Table Lamp', 'Modern table lamp with adjustable brightness', 50);**



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

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

**UPDATE Orders SET Status='Shipped' WHERE OrderID=102;**

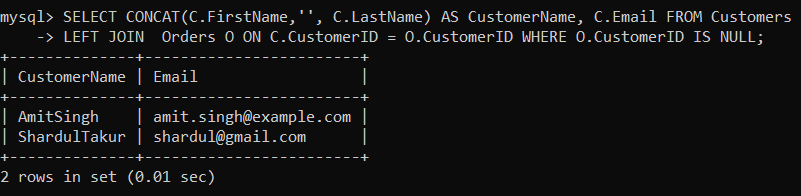


**TASK 3:**

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

**SELECT CONCAT(C.FirstName,'', C.LastName) AS CustomerName, C.Email FROM Customers C**

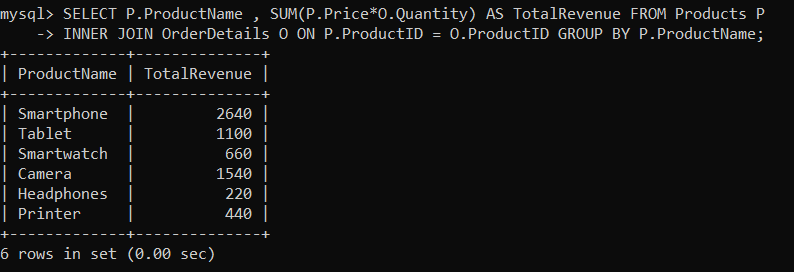
**LEFT JOIN Orders O ON C.CustomerID = O.CustomerID WHERE O.CustomerID IS NULL;**

****

-- 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(P.Price\*O.Quantity) AS TotalRevenue FROM Products P**

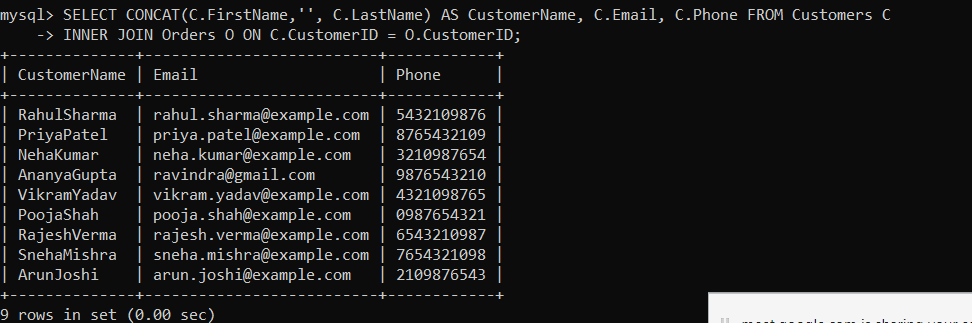
**INNER JOIN OrderDetails O ON P.ProductID = O.ProductID GROUP BY P.ProductName;**

****

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

**SELECT CONCAT(C.FirstName,'', C.LastName) AS CustomerName, C.Email, C.Phone FROM Customers C**

**INNER JOIN Orders O ON C.CustomerID = O.CustomerID;**

****

-- 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.

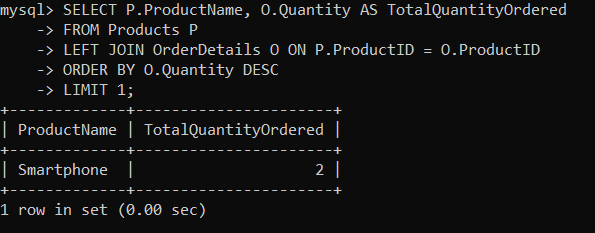
**SELECT P.ProductName, O.Quantity AS TotalQuantityOrdered**

**FROM Products P**

**LEFT JOIN OrderDetails O ON P.ProductID = O.ProductID**

**ORDER BY O.Quantity DESC**

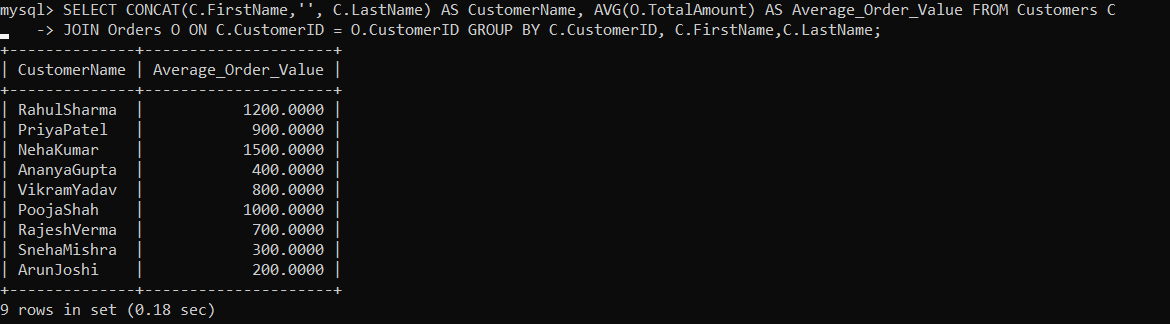
**LIMIT 1;**

****

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

**SELECT CONCAT(C.FirstName,'', C.LastName) AS CustomerName, AVG(O.TotalAmount) AS Average\_Order\_Value FROM Customers C**

**JOIN Orders O ON C.CustomerID = O.CustomerID GROUP BY C.CustomerID, C.FirstName,C.LastName;**

****

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

**SELECT O.OrderID, C.FirstName, C.LastName, SUM(P.Price \* OD.Quantity) AS TotalRevenue**

**FROM Orders O INNER JOIN OrderDetails OD ON O.OrderID = OD.OrderID**

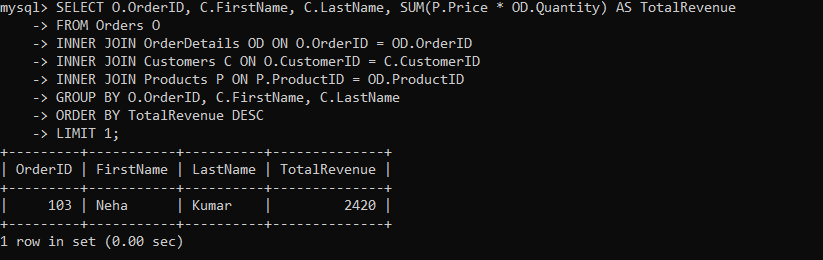
**INNER JOIN Customers C ON O.CustomerID = C.CustomerID**

**INNER JOIN Products P ON P.ProductID = OD.ProductID**

**GROUP BY O.OrderID, C.FirstName, C.LastName**

**ORDER BY TotalRevenue DESC**

**LIMIT 1;**

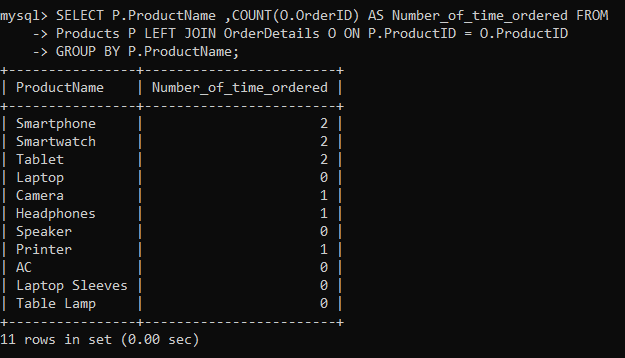
****

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

**SELECT P.ProductName ,COUNT(O.OrderID) AS Number\_of\_time\_ordered FROM**

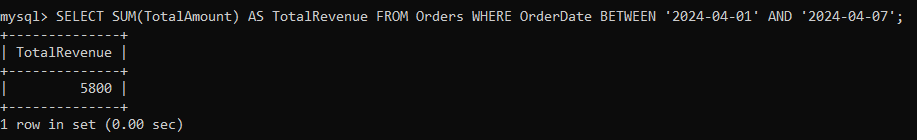
**Products P LEFT JOIN OrderDetails O ON P.ProductID = O.ProductID**

**GROUP BY P.ProductName;**

****

-- 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.

**SELECT SUM(TotalAmount) AS TotalRevenue FROM Orders WHERE OrderDate BETWEEN '2024-04-01' AND '2024-04-07';**

****

**TASK: 4**

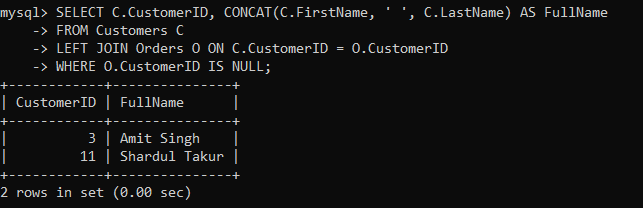
**--** Write an SQL query to find out which customers have not placed any orders

**SELECT C.CustomerID, CONCAT(C.FirstName, ' ', C.LastName) AS FullName**

**FROM Customers C**

**LEFT JOIN Orders O ON C.CustomerID = O.CustomerID**

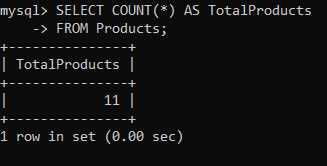
**WHERE O.CustomerID IS NULL;**

****

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

**SELECT COUNT(\*) AS TotalProducts**

**FROM Products;**



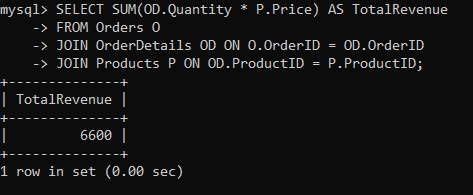
**--** Write an SQL query to calculate the total revenue generated by TechShop.

**SELECT SUM(OD.Quantity \* P.Price) AS TotalRevenue**

**FROM Orders O**

**JOIN OrderDetails OD ON O.OrderID = OD.OrderID**

**JOIN Products P ON OD.ProductID = P.ProductID;**



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

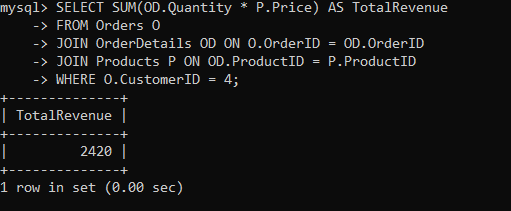
**SELECT SUM(OD.Quantity \* P.Price) AS TotalRevenue**

**FROM Orders O**

**JOIN OrderDetails OD ON O.OrderID = OD.OrderID**

**JOIN Products P ON OD.ProductID = P.ProductID**

**WHERE O.CustomerID = 4;**



**--** 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.

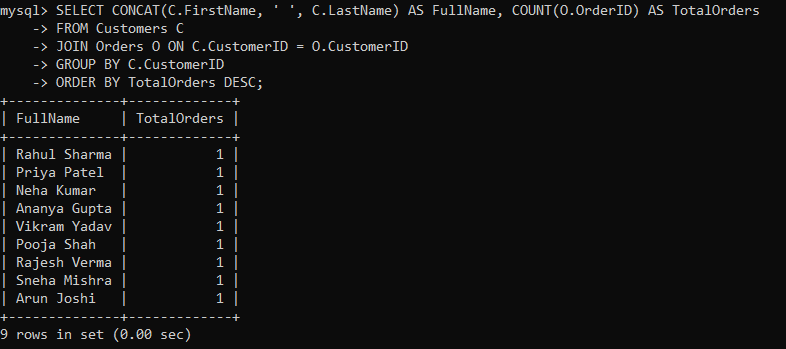
**SELECT CONCAT(C.FirstName, ' ', C.LastName) AS FullName, COUNT(O.OrderID) AS TotalOrders**

**FROM Customers C**

**JOIN Orders O ON C.CustomerID = O.CustomerID**

**GROUP BY C.CustomerID**

**ORDER BY TotalOrders DESC;**



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

**SELECT AVG(TotalRevenue / TotalOrders) AS AverageOrderValue**

**FROM (**

**SELECT SUM(OD.Quantity \* P.Price) AS TotalRevenue, COUNT(DISTINCT O.OrderID) AS TotalOrders**

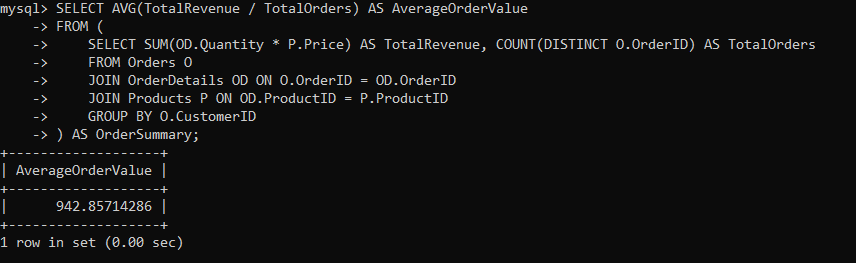
**FROM Orders O**

**JOIN OrderDetails OD ON O.OrderID = OD.OrderID**

**JOIN Products P ON OD.ProductID = P.ProductID**

**GROUP BY O.CustomerID**

**) AS OrderSummary;**



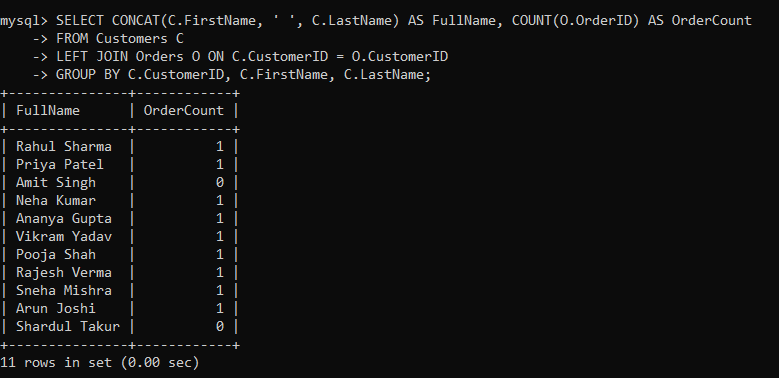
**--** Write an SQL query to find the total number of orders placed by each customer and list their names along with the order count.

**SELECT CONCAT(C.FirstName, ' ', C.LastName) AS FullName, COUNT(O.OrderID) AS OrderCount**

**FROM Customers C**

**LEFT JOIN Orders O ON C.CustomerID = O.CustomerID**

**GROUP BY C.CustomerID, C.FirstName, C.LastName;**

****