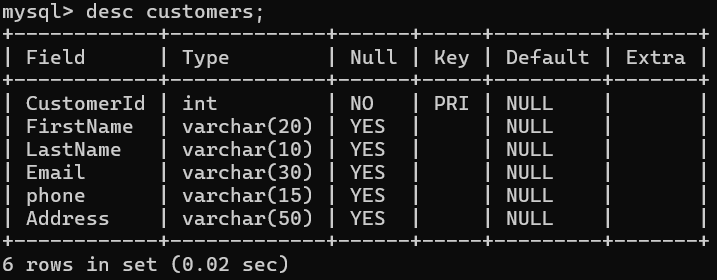
**J523 SHIVAM KUMAR MySQL ASSIGNMENT**

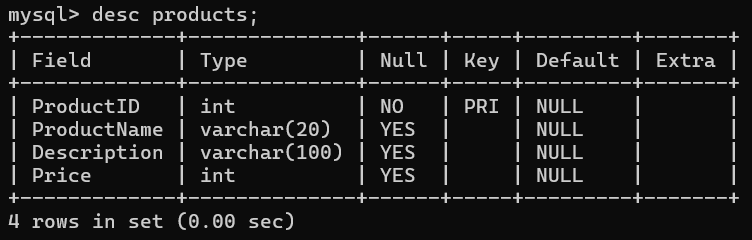
 **TechShop, an electronic gadgets shop**

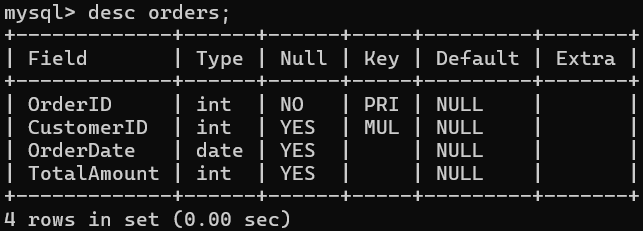
**Task:1. Database Design:**

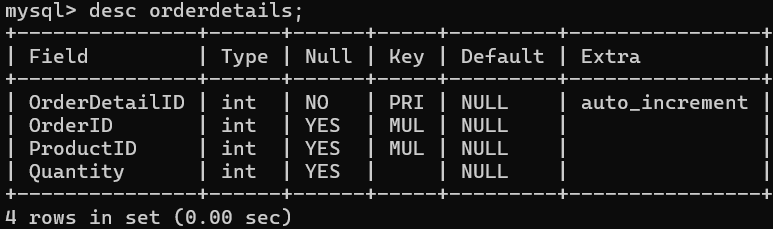
1. Create the database named "TechShop"  
-> **Create Database TechShop;**

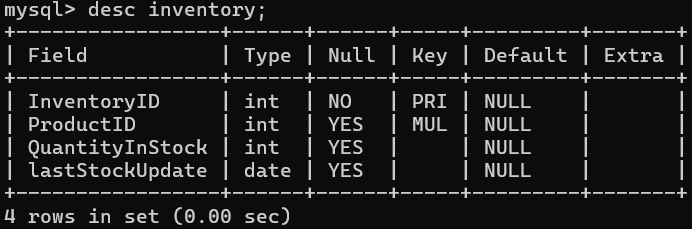
2. Define the schema for the Customers, Products, Orders, OrderDetails, and Inventory tables based on the provided schema.



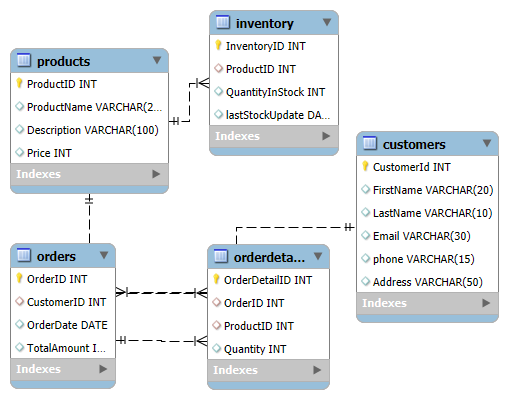




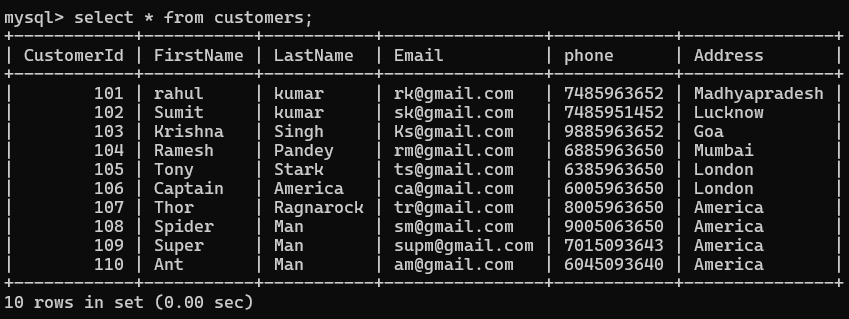


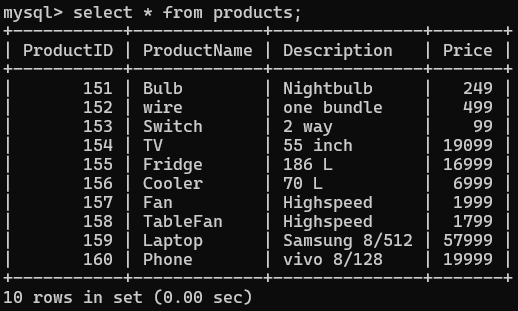


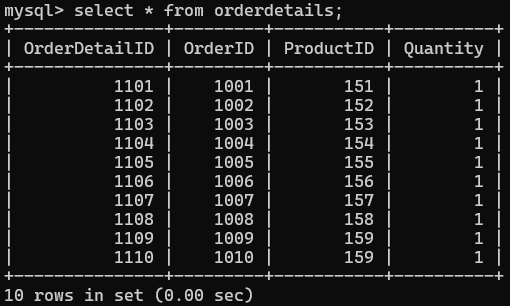
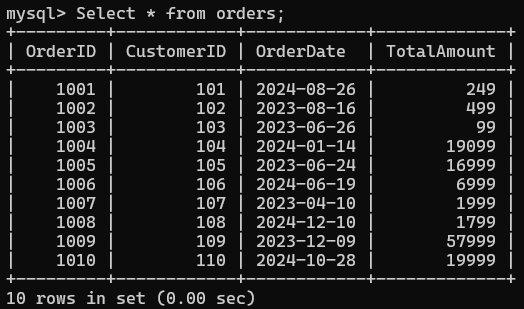
3. Create an ERD (Entity Relationship Diagram) for the database.

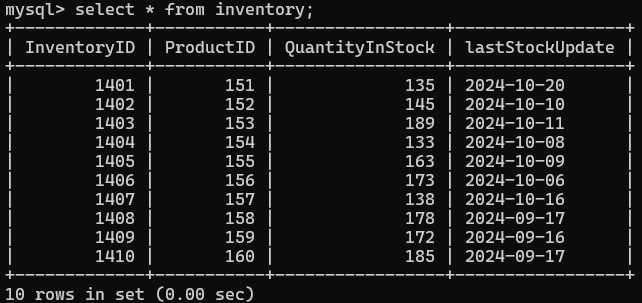


5. Insert at least 10 sample records into each of the following tables.





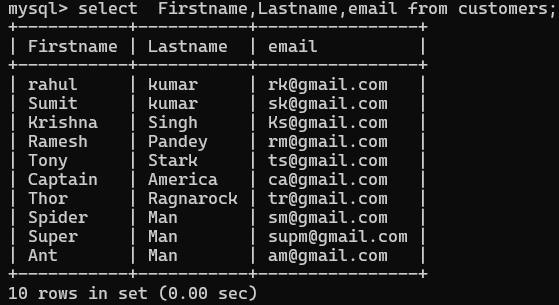




**Task 2.**

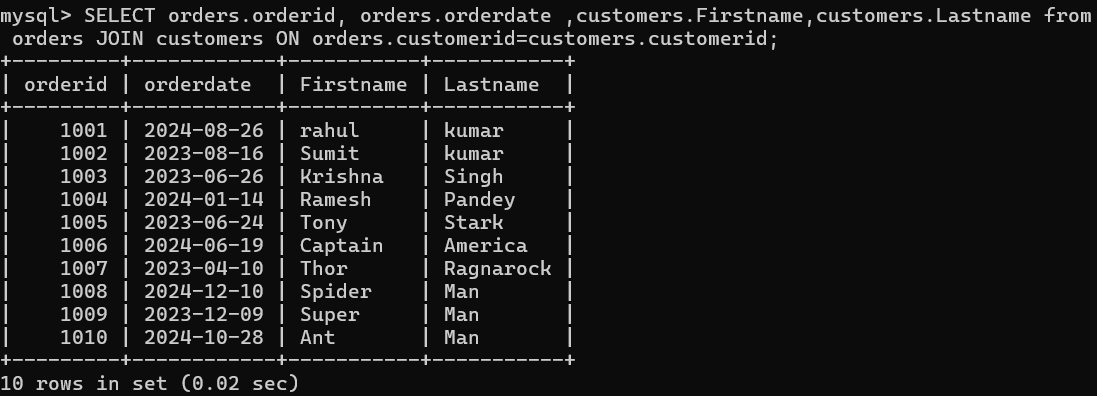
1. Write an SQL query to retrieve the names and emails of all customers.

->Select Firstname,Lastname,email from customers;



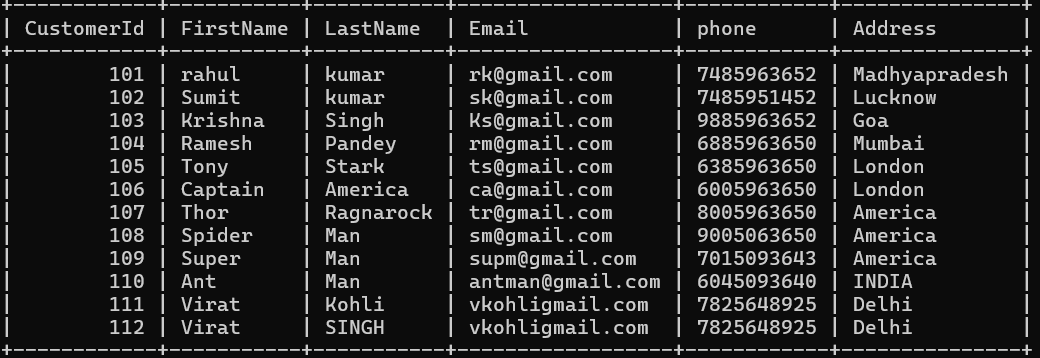
1. Write an SQL query to list all orders with their order dates and corresponding customer names.

* SELECT orders.orderid, orders.orderdate ,customers.Firstname, customers.Lastname from orders JOIN customers ON orders.customerid=customers.customerid;



1. 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,'Virat','Kohli','vkohligmail.com',7825648925,'Delhi');**



1. 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;**

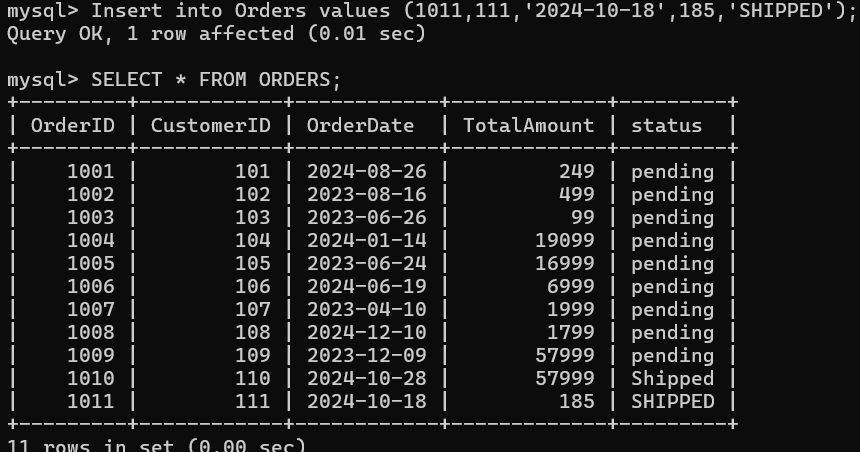
5.Write an SQL query to delete a specific order and its associated order details from the "Orders" and "OrderDetails" tables. Allow users to input the order ID as a parameter.

-> **Delete FROM orderdetails where orderid=1010;**

**-> Delete FROM orders where orderid=1010;**

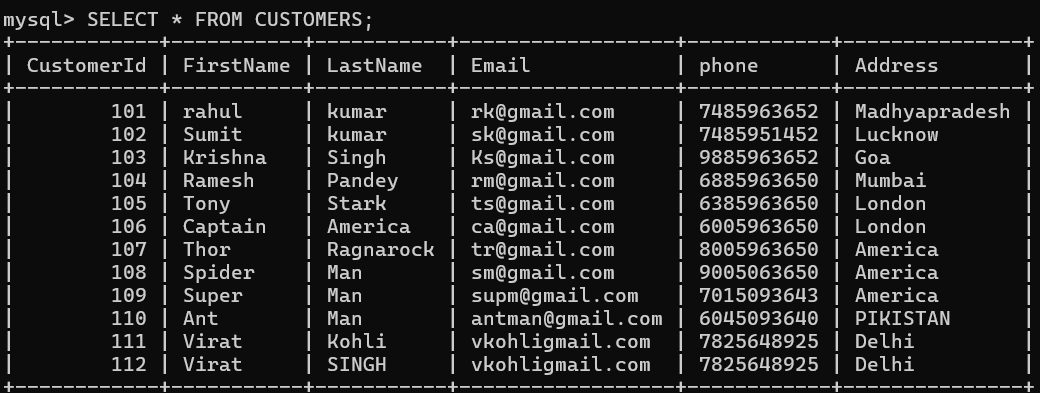
6. Write an SQL query to insert a new order into the "Orders" table. Include the customer ID, order date, and any other necessary information.

**-> Insert into Orders values (1010,110,'2024-10-18',185);**

****

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

**->Update Customers SET email='antman@gmail.com', address='INDIA' WHERE Customerid=110;**

****

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.

**-> UPDATE Orders O SET TotalAmount = (**

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

**FROM OrderDetails OD**

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

**WHERE OD.OrderID = O.OrderID );**

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 = 110);**

**-> DELETE FROM Orders**

**WHERE CustomerID = 110;**

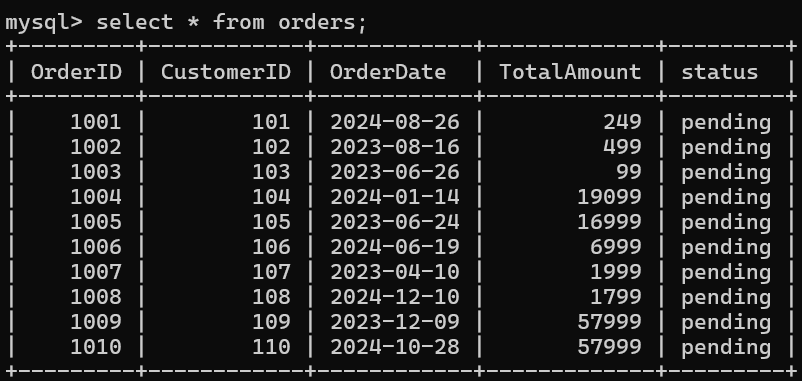
10. Write an SQL query to insert a new electronic gadget product into the "Products" table, including product name, category, price, and any other relevant details.

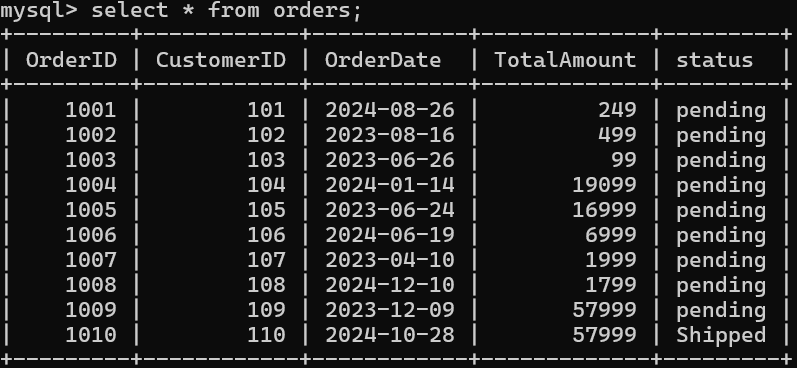
**-> INSERT INTO Products values(161,'MOTOR','2 KB',3200);**

****

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

**-> update orders set status='Shipped' where orderid =1010;**

****

****

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.

* **UPDATE Customers C**

**SET NumberOfOrders = ( SELECT COUNT(\*)**

**FROM Orders O WHERE O.CustomerID = C.CustomerID);**

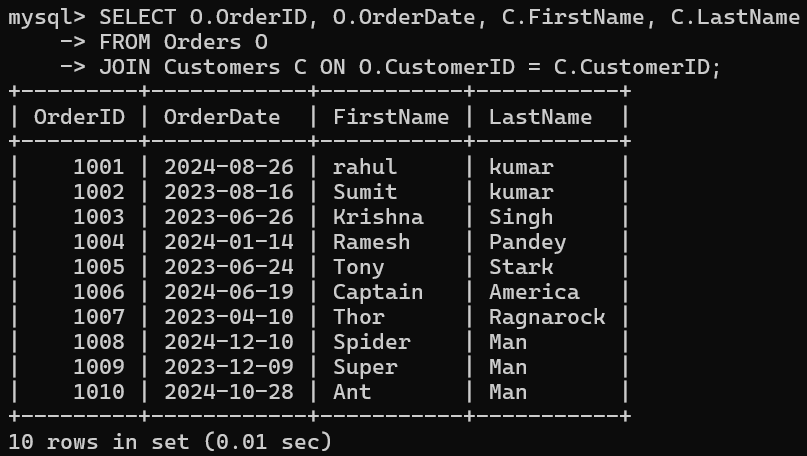
**Task 3**

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

* **SELECT O.OrderID, O.OrderDate, C.FirstName, C.LastName**

**FROM Orders O**

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



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

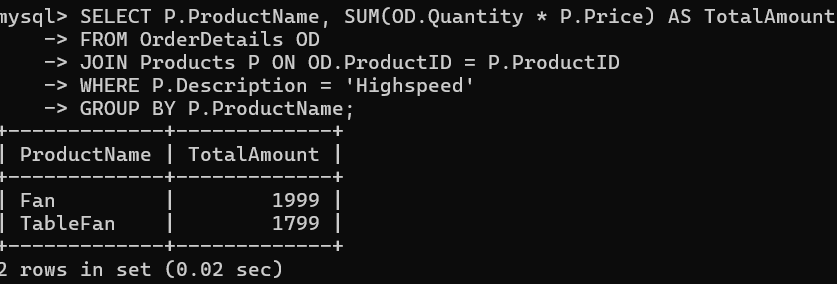
* SELECT P.ProductName, SUM(OD.Quantity \* P.Price) AS TotalAmount

FROM OrderDetails OD

JOIN Products P ON OD.ProductID = P.ProductID

WHERE P.Description = 'Highspeed'

GROUP BY P.ProductName;



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

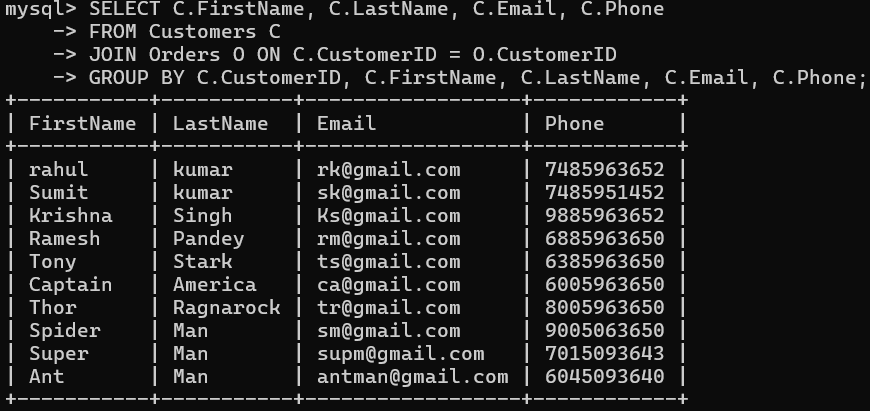
* SELECT P.ProductName, SUM(OD.Quantity \* P.Price) AS TotalAmount

FROM OrderDetails OD

JOIN Products P ON OD.ProductID = P.ProductID

WHERE P.Description = 'Highspeed'

GROUP BY P.ProductName;



1. 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 TotalQuantity

FROM OrderDetails OD

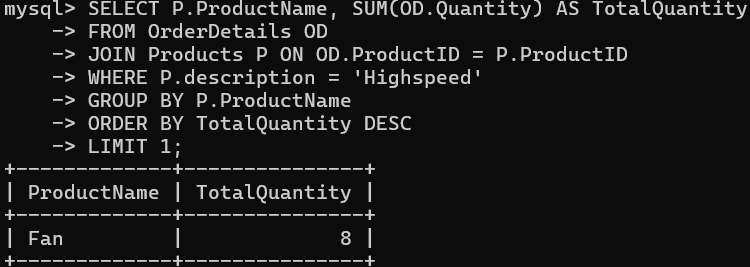
JOIN Products P ON OD.ProductID = P.ProductID

WHERE P.description = 'Highspeed'

GROUP BY P.ProductName

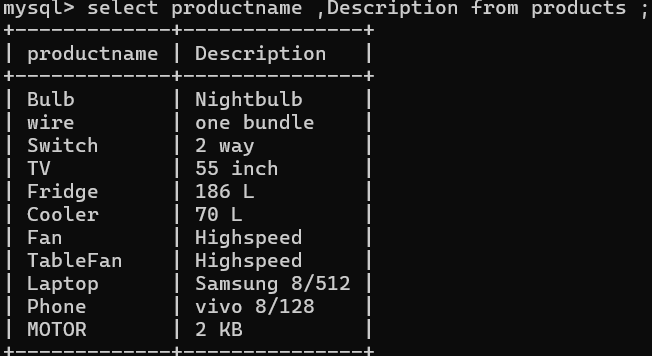
ORDER BY TotalQuantity DESC

LIMIT 1;



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

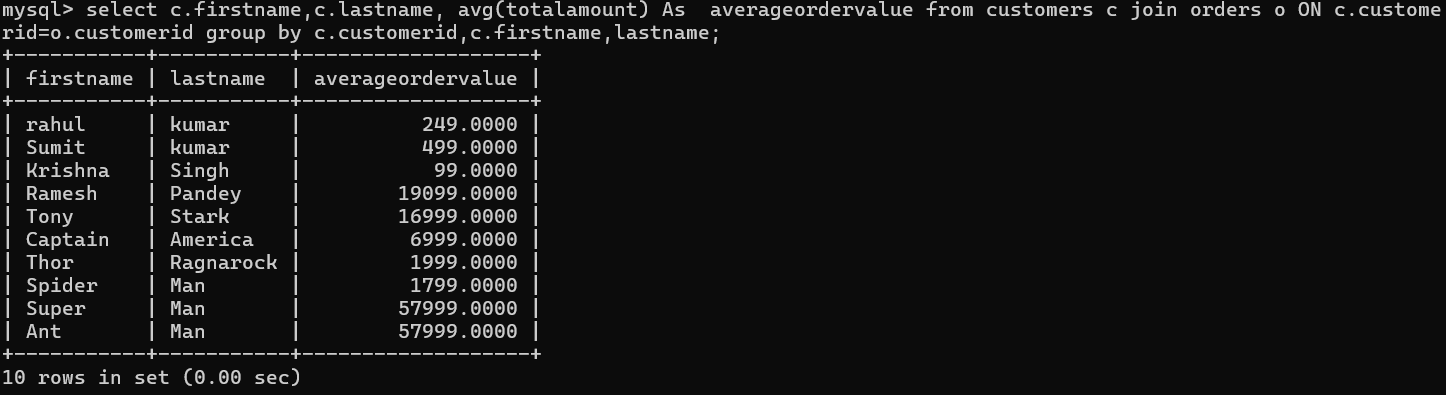
* select productname ,Description from products ;



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 c.firstname,c.lastname, avg(totalamount) As averageordervalue from customers c join orders o ON c.custome

rid=o.customerid group by c.customerid,c.firstname,lastname;



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

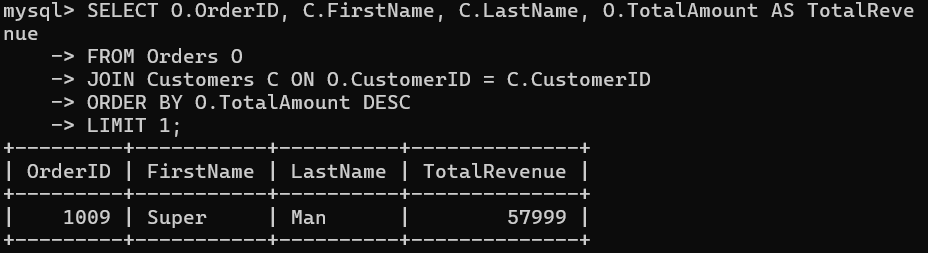
* SELECT O.OrderID, C.FirstName, C.LastName, O.TotalAmount AS TotalRevenue

FROM Orders O

JOIN Customers C ON O.CustomerID = C.CustomerID

ORDER BY O.TotalAmount DESC

LIMIT 1;



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

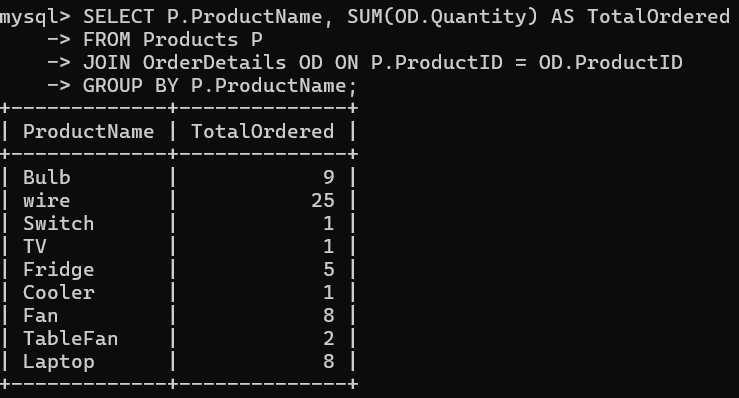
* SELECT P.ProductName, SUM(OD.Quantity) AS TotalOrdered

FROM Products P

JOIN OrderDetails OD ON P.ProductID = OD.ProductID

WHERE P.Category = 'Electronic Gadgets'

GROUP BY P.ProductName;



1. 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.FirstName, C.LastName, C.Email, C.Phone

FROM Customers C

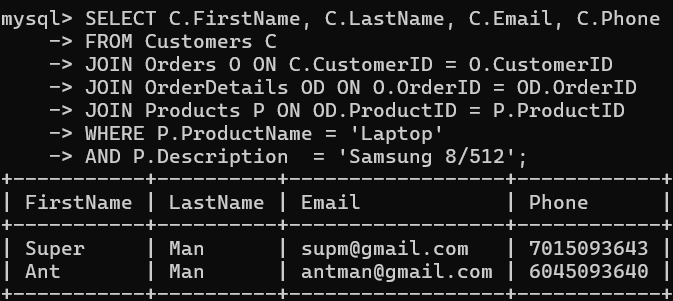
JOIN Orders O ON C.CustomerID = O.CustomerID

JOIN OrderDetails OD ON O.OrderID = OD.OrderID

JOIN Products P ON OD.ProductID = P.ProductID

WHERE P.ProductName = 'Laptop'

AND P.Description = 'Samsung 8/512';

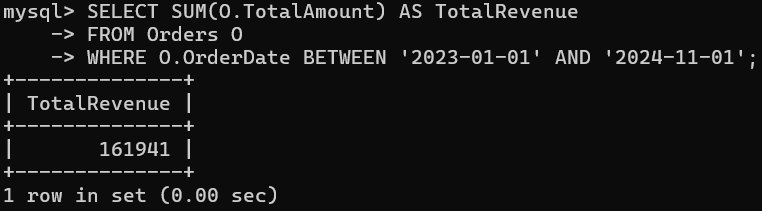


1. Write an SQL query to calculate the total revenue generated by all orders placed within a specific time period. Allow users to input the start and end dates as parameters.

* SELECT SUM(O.TotalAmount) AS TotalRevenue

FROM Orders O

WHERE O.OrderDate BETWEEN '2023-01-01' AND '2024-11-01';



**Task 4. Subquery and its type:**

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

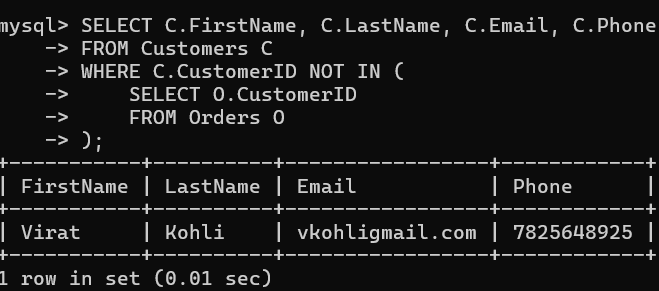
* SELECT C.FirstName, C.LastName, C.Email, C.Phone

FROM Customers C

WHERE C.CustomerID NOT IN (

SELECT O.CustomerID

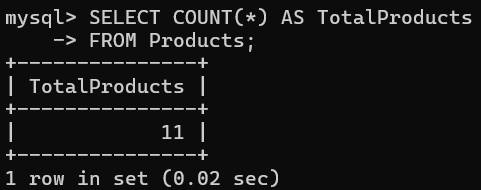
FROM Orders O );



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

* SELECT COUNT(\*) AS TotalProducts

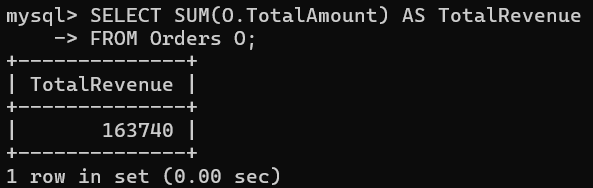
FROM Products;



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

* SELECT SUM(O.TotalAmount) AS TotalRevenue

FROM Orders O;



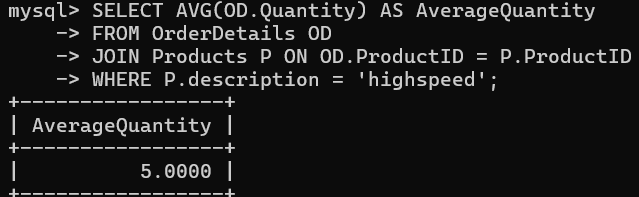
1. Write an SQL query to calculate the average quantity ordered for products in a specific category. Allow users to input the category name as a parameter.

* SELECT AVG(OD.Quantity) AS AverageQuantity

FROM OrderDetails OD

JOIN Products P ON OD.ProductID = P.ProductID

WHERE P.description = 'highspeed';

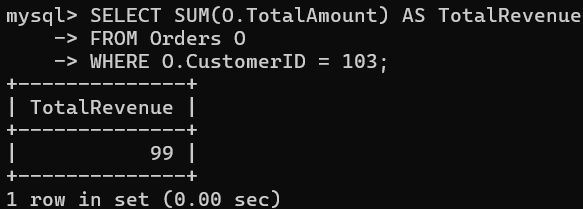


1. Write an SQL query to calculate the total revenue generated by a specific customer. Allow users to input the customer ID as a parameter.

* SELECT SUM(O.TotalAmount) AS TotalRevenue

FROM Orders O

WHERE O.CustomerID = 103;



1. 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 FirstName, LastName, NumberOfOrders

FROM (

SELECT C.FirstName, C.LastName, COUNT(O.OrderID) AS NumberOfOrders

FROM Customers C

JOIN Orders O ON C.CustomerID = O.CustomerID

GROUP BY C.CustomerID, C.FirstName, C.LastName) AS CustomerOrders

WHERE NumberOfOrders = (

SELECT MAX(NumberOfOrders)

FROM (

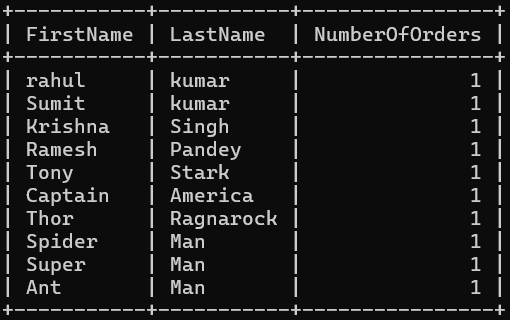
SELECT COUNT(O.OrderID) AS NumberOfOrders

FROM Customers C

JOIN Orders O ON C.CustomerID = O.CustomerID

GROUP BY C.CustomerID

) AS OrderCounts);



1. 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.description, SUM(OD.Quantity) AS TotalQuantityOrdered

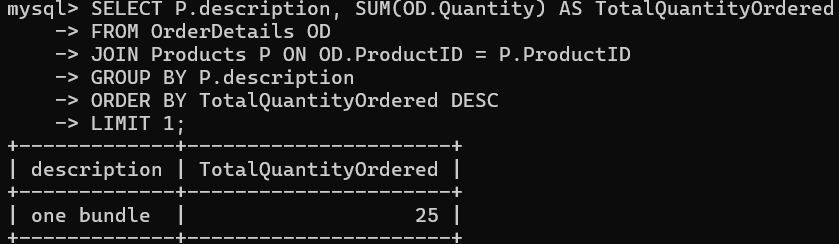
FROM OrderDetails OD

JOIN Products P ON OD.ProductID = P.ProductID

GROUP BY P.description

ORDER BY TotalQuantityOrdered DESC

LIMIT 1;



1. 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 FirstName, LastName, TotalSpending

FROM (

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

FROM Customers C

JOIN Orders O ON C.CustomerID = O.CustomerID

JOIN OrderDetails OD ON O.OrderID = OD.OrderID

JOIN Products P ON OD.ProductID = P.ProductID

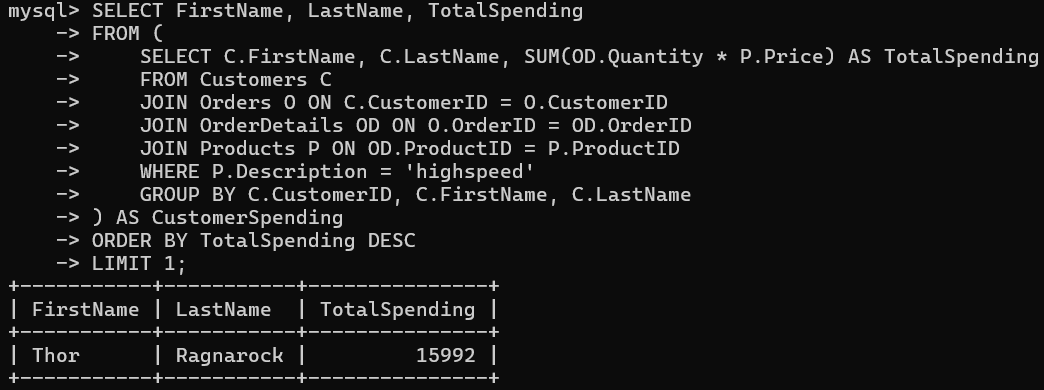
WHERE P.Description = 'highspeed'

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

) AS CustomerSpending

ORDER BY TotalSpending DESC

LIMIT 1;



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

SELECT FirstName, LastName, AverageOrderValue

FROM (

SELECT C.FirstName, C.LastName, AVG(O.TotalAmount) AS AverageOrderValue

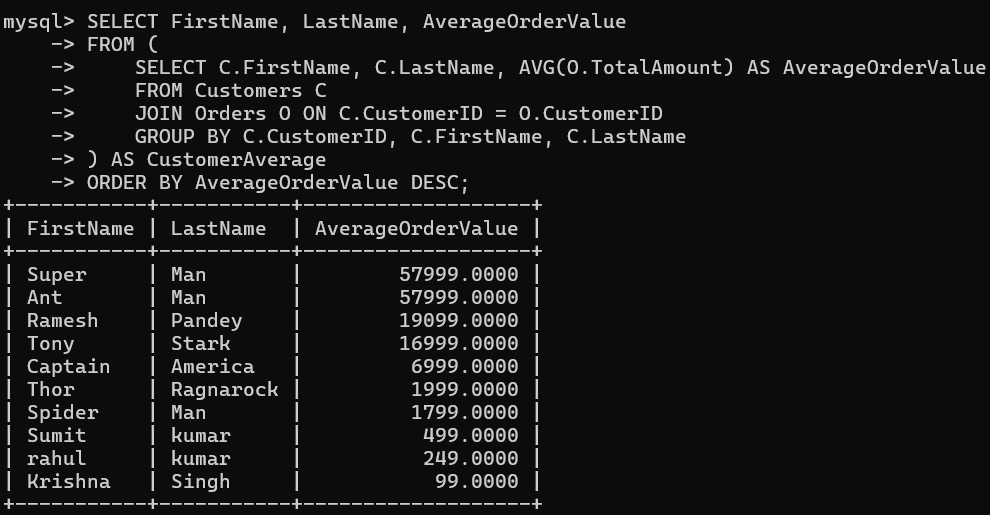
FROM Customers C

JOIN Orders O ON C.CustomerID = O.CustomerID

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

) AS CustomerAverage

ORDER BY AverageOrderValue DESC;



1. 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 FirstName, LastName, OrderCount

FROM (

SELECT C.FirstName, C.LastName, COUNT(O.OrderID) AS OrderCount

FROM Customers C

JOIN Orders O ON C.CustomerID = O.CustomerID

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

) AS CustomerOrders

ORDER BY OrderCount DESC;

