Task1

1)

CREATE DATABASE TechShop;



2)

CREATE TABLE Customers (

-> CustomerID INT AUTO\_INCREMENT PRIMARY KEY,

-> FirstName VARCHAR(100) NOT NULL,

-> LastName VARCHAR(100),

-> Email VARCHAR(100) UNIQUE NOT NULL,

-> Phone VARCHAR(20),

-> Address VARCHAR(255)

-> );

mysql> CREATE TABLE Products (

-> ProductID INT AUTO\_INCREMENT PRIMARY KEY,

-> ProductName VARCHAR(100) NOT NULL,

-> Description TEXT,

-> Price DECIMAL(10, 2) NOT NULL

-> );

mysql> CREATE TABLE Orders (

-> OrderID INT AUTO\_INCREMENT PRIMARY KEY,

-> CustomerID INT,

-> OrderDate DATE,

-> TotalAmount DECIMAL(10,2),

-> FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

-> );

mysql> CREATE TABLE OrderDetails (

-> OrderDetailID INT AUTO\_INCREMENT PRIMARY KEY,

-> OrderID INT,

-> ProductID INT,

-> Quantity INT,

-> FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),

-> FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

-> );

mysql> CREATE TABLE Inventory (

-> InventoryID INT AUTO\_INCREMENT PRIMARY KEY,

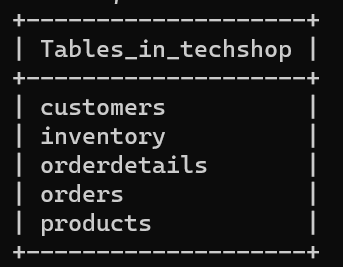
-> ProductID INT,

-> QuantityInStock INT,

-> LastStokeUpdate INT,

-> FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

-> );



3)

**A diagram of customer service

Description automatically generated with medium confidence**

5)

a) INSERT INTO Customers (FirstName, LastName, Email, Phone, Address) VALUES

-> ('Aaron', 'Sabu', 'aaron@gmail.com', '1234567890', '123 Main St'),

-> ('Aakash', 'Saji', 'aakash@gmail.com', '9876543210', '456 Elm St'),

-> ('Allen', 'Simon', 'allen@gmail.com', '9876559810', '457 Elm St'),

-> ('Akhil', 'Sam', 'akhil@gmail.com', '8976543210', '458 Elm St'),

-> ('Aravind', 'Sree', 'aravind@gmail.com', '7676543210', '459 Elm St'),

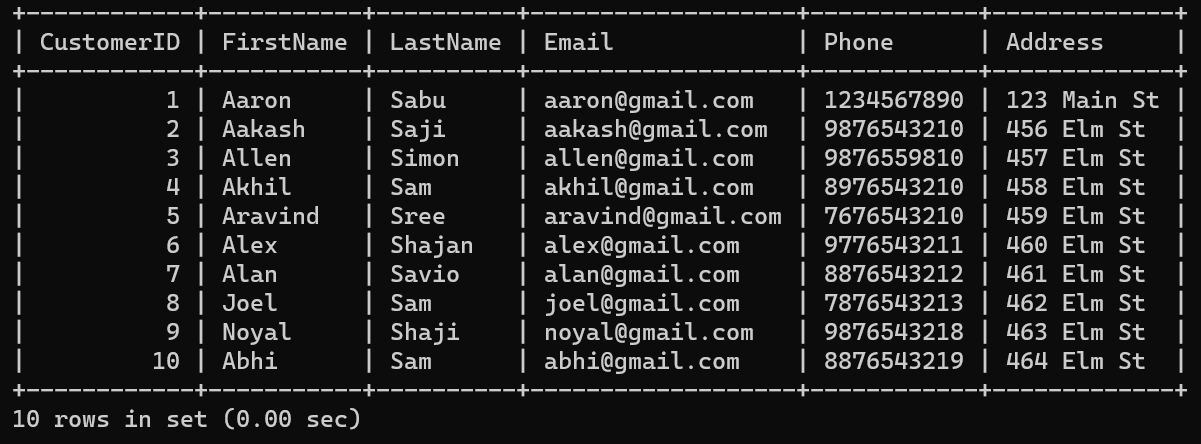
-> ('Alex', 'Shajan', 'alex@gmail.com', '9776543211', '460 Elm St'),

-> ('Alan', 'Savio', 'alan@gmail.com', '8876543212', '461 Elm St'),

-> ('Joel', 'Sam', 'joel@gmail.com', '7876543213', '462 Elm St'),

-> ('Noyal', 'Shaji', 'noyal@gmail.com', '9876543218', '463 Elm St'),

-> ('Abhi', 'Sam', 'abhi@gmail.com', '8876543219', '464 Elm St');



b) INTO Products (ProductName, Description, Price) VALUES

-> ('Laptop', 'Gaming laptop', 70000.00),

-> ('Smartphone', 'Latest smartphone model', 25000.00),

-> ('Laptop', 'Low-performance laptop', 70000.00),

-> ('Tablet', 'Gaming tablet', 70000.00),

-> ('Smartphone', 'Latest smartphone model', 20000.00),

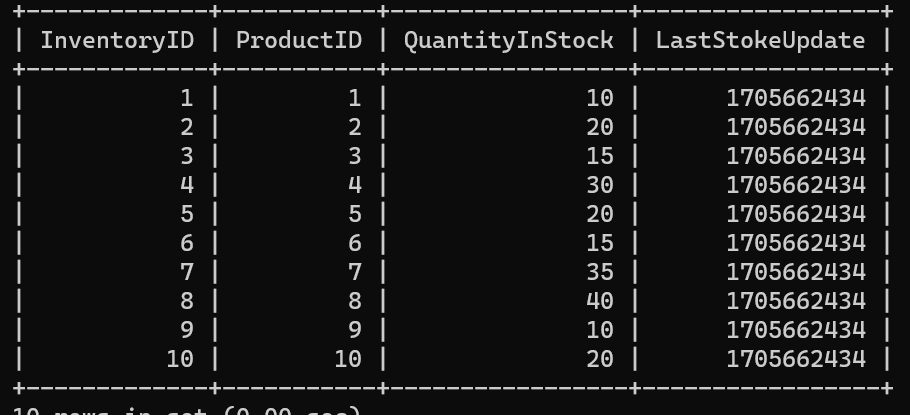
-> ('Laptop', 'Gaming laptop', 90000.00),

-> ('Laptop', 'Chrome OS', 25000.00),

-> ('Smartphone', 'Flip model', 90000.00),

-> ('Laptop', 'Performance laptop', 70000.00),

-> ('Laptop', 'Gaming laptop', 60000.00);



c) INSERT INTO OrderDetails (OrderID, ProductID, Quantity) VALUES

-> (11, 1, 2),

-> (21, 2, 1),

-> (12, 3, 2),

-> (22, 4, 1),

-> (13, 5, 1),

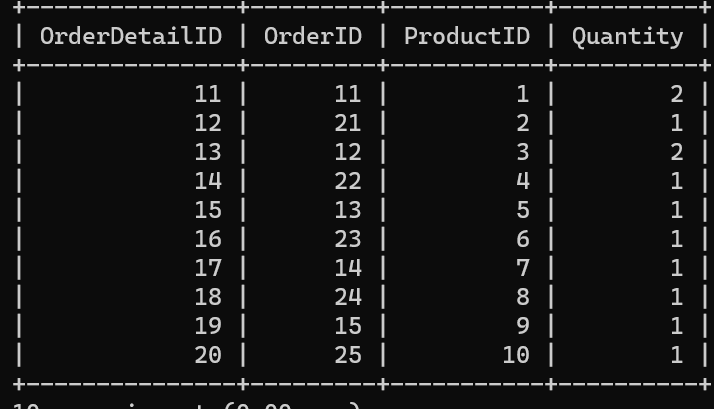
-> (23, 6, 1),

-> (14, 7, 1),

-> (24, 8, 1),

-> (15, 9, 1),

-> (25, 10, 1);



d) INSERT INTO Orders (CustomerID, OrderDate, TotalAmount) VALUES

-> (1, '2023-01-15', 80000.00),

-> (2, '2023-02-20', 90000.00),

-> (3, '2023-01-21', 50000.00),

-> (4, '2023-01-22', 85000.00),

-> (5, '2023-01-22', 90000.00),

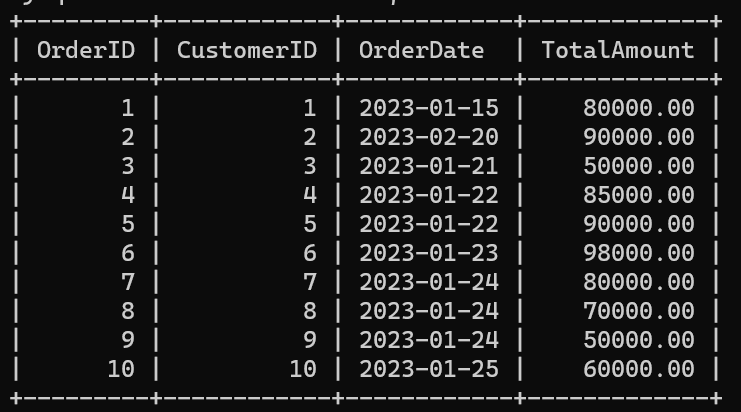
-> (6, '2023-01-23', 98000.00),

-> (7, '2023-01-24', 80000.00),

-> (8, '2023-01-24', 70000.00),

-> (9, '2023-01-24', 50000.00),

-> (10, '2023-01-25', 60000.00);



e) mysql> INSERT INTO Products (ProductName, Description, Price) VALUES

-> ('Laptop', 'Gaming laptop', 70000.00),

-> ('Smartphone', 'Latest smartphone model', 25000.00),

-> ('Laptop', 'Low-performance laptop', 70000.00),

-> ('Tablet', 'Gaming tablet', 70000.00),

-> ('Smartphone', 'Latest smartphone model', 20000.00),

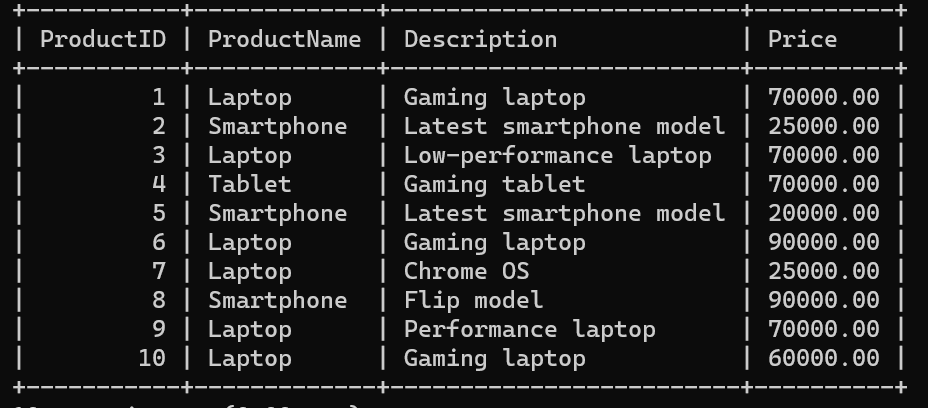
-> ('Laptop', 'Gaming laptop', 90000.00),

-> ('Laptop', 'Chrome OS', 25000.00),

-> ('Smartphone', 'Flip model', 90000.00),

-> ('Laptop', 'Performance laptop', 70000.00),

-> ('Laptop', 'Gaming laptop', 60000.00);

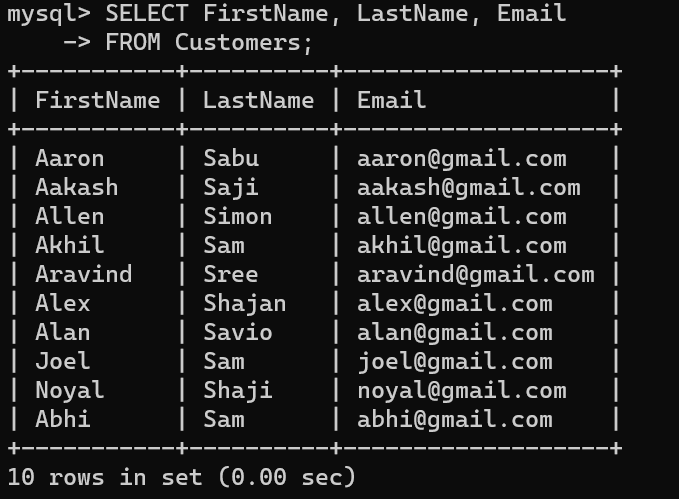


Task-2

1)

SELECT FirstName, LastName, Email

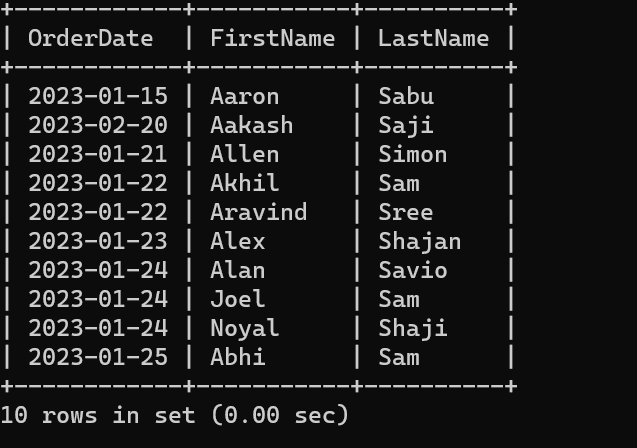
-> FROM Customers



1. SELECT Orders.OrderDate, Customers.FirstName, Customers.LastName

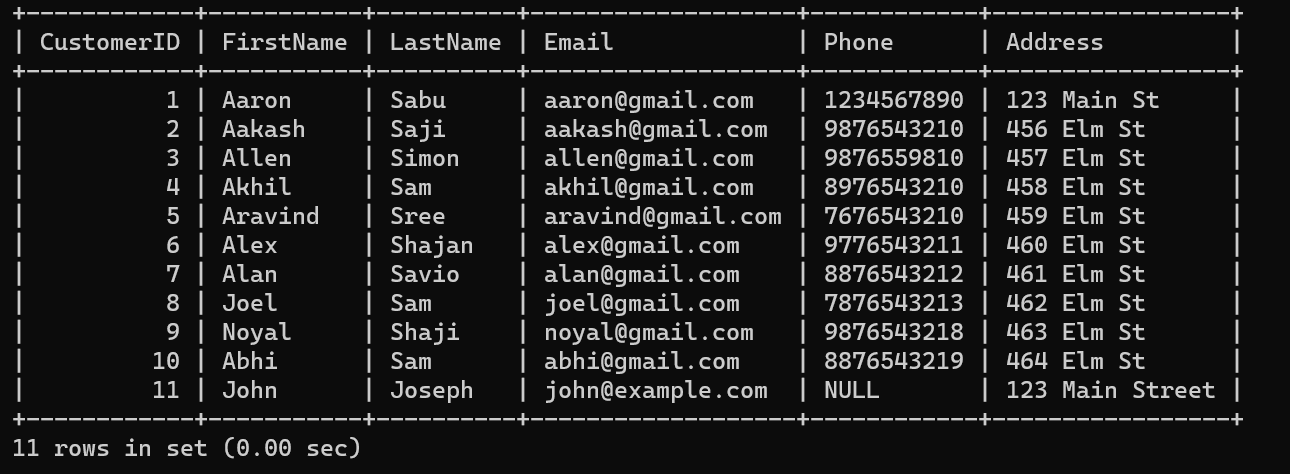
-> FROM Orders

-> INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID;



3) INSERT INTO Customers (FirstName, LastName, Email, Address)

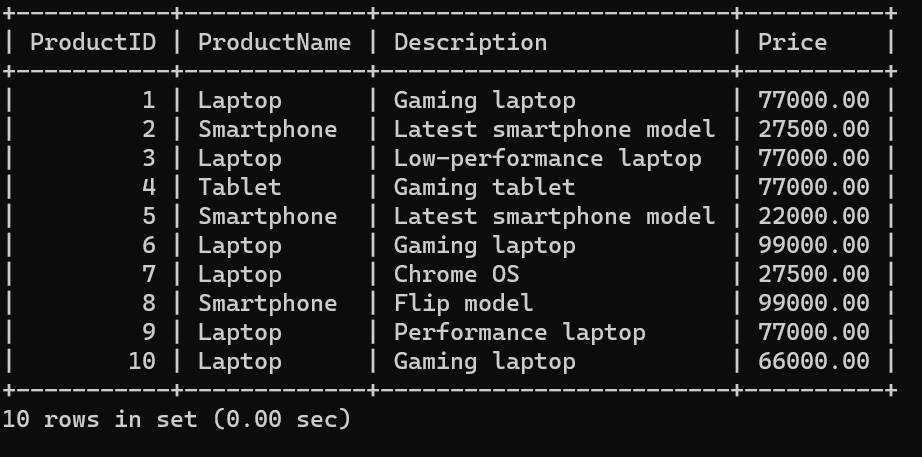
-> VALUES ('John', 'Joseph', 'john@example.com', '123 Main Street');



1. UPDATE Products

-> SET Price = Price + (Price\*0.1)

-> WHERE ProductID IN(1,2,3,4,5,6,7,8,9,10);

s

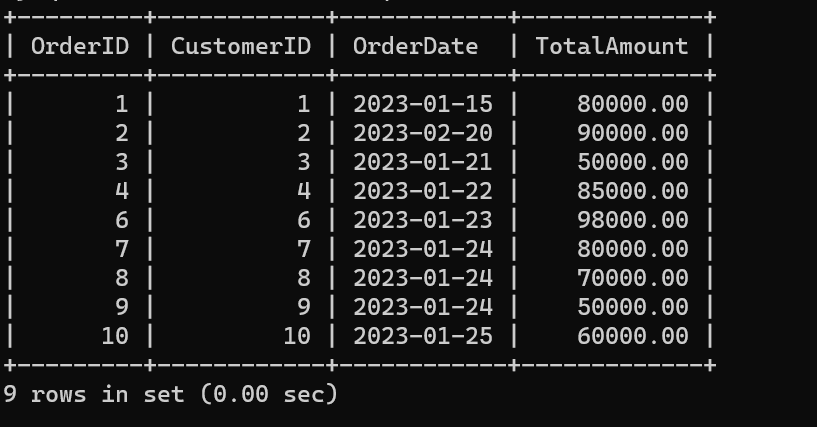
1. SET @OrderID = 5;

DELETE FROM Orders

-> WHERE OrderID = @OrderID;

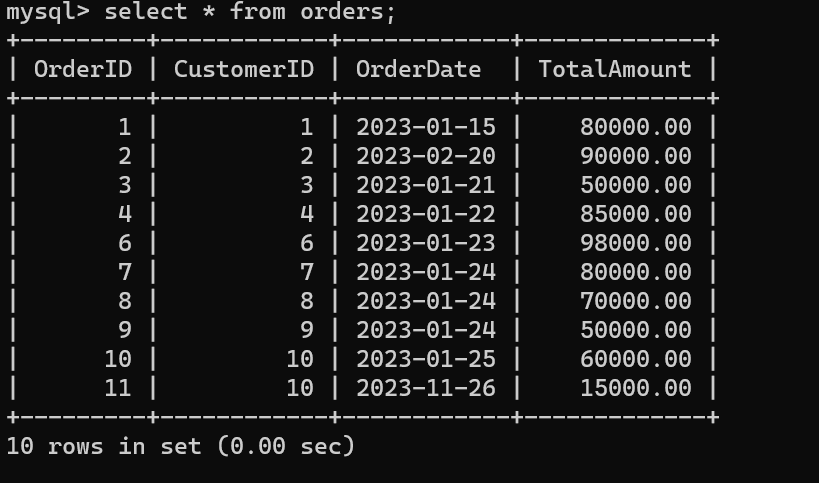
DELETE FROM OrderDetails

-> WHERE OrderID=@OrderID;



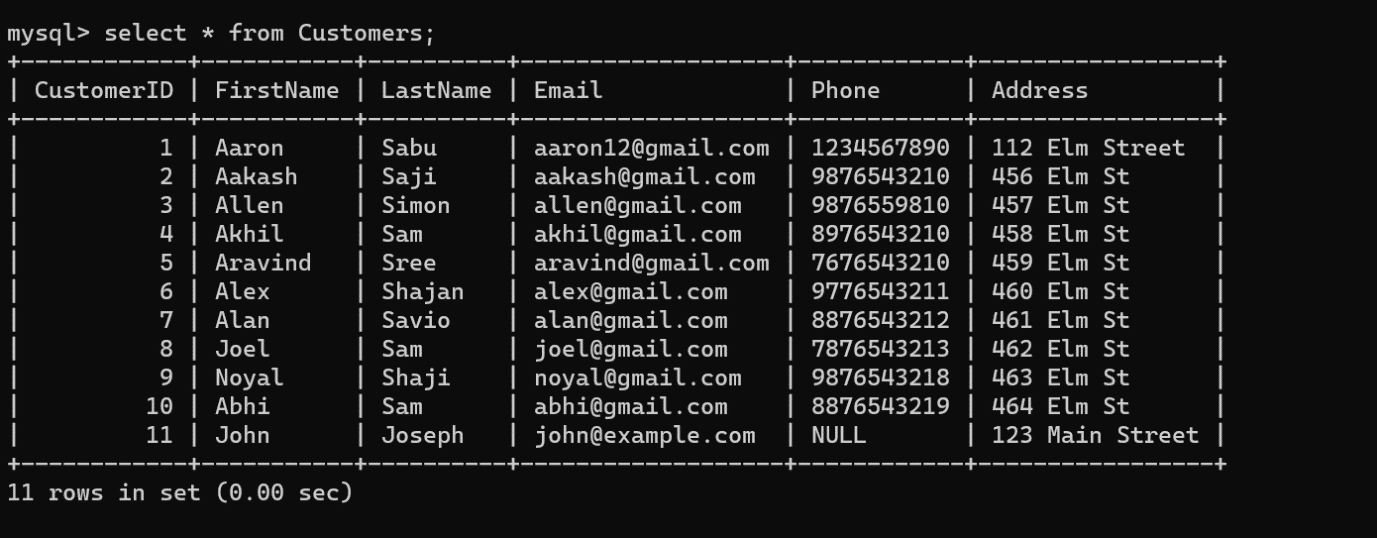
6)

INSERT INTO Orders (CustomerID, OrderDate, TotalAmount) VALUES (10, '2023-11-26', 15000);



7)

UPDATE Customers SET Email = 'aaron12@gmail.com', Address = '112 Elm Street' WHERE CustomerID = 1;



8)

SET TotalAmount = (

SELECT SUM(OrderDetails.Quantity \* Products.Price)

FROM OrderDetails, Products

WHERE OrderDetails.OrderID = Orders.OrderID AND

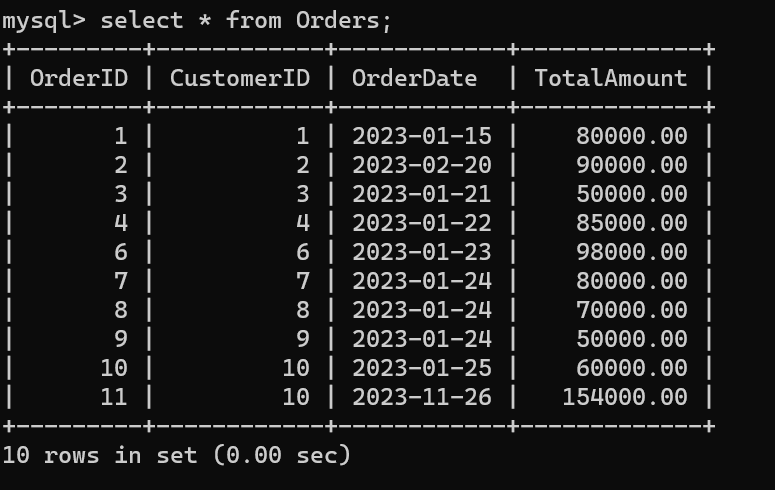
OrderDetails.ProductID = Products.ProductID )

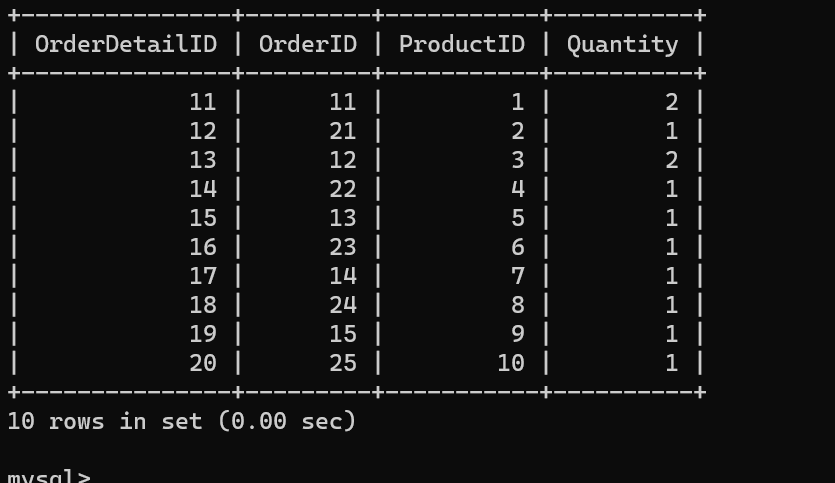
WHERE OrderID IN (

SELECT OrderID

FROM OrderDetails

);





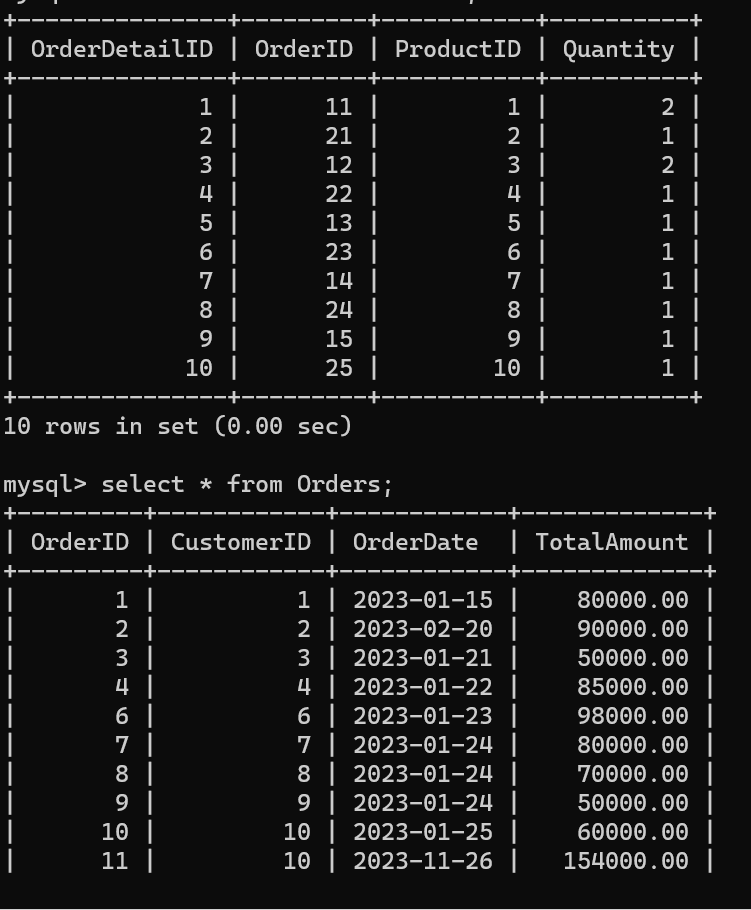
9)

DELETE FROM OrderDetails

WHERE OrderID IN (SELECT OrderID FROM Orders WHERE CustomerID = 5);

DELETE FROM Orders

WHERE CustomerID = 5;



10)

INSERT INTO Products (ProductName, Description, Price)

VALUES ('Macbook', 'Thin and light weight', 990000);



11)

ALTER TABLE Orders ADD COLUMN Status VARCHAR(50)

; ALTER TABLE Orders ALTER COLUMN Status SET DEFAULT 'pending';

UPDATE Orders SET Status = 'Shipped'

WHERE OrderID = 11;

12)

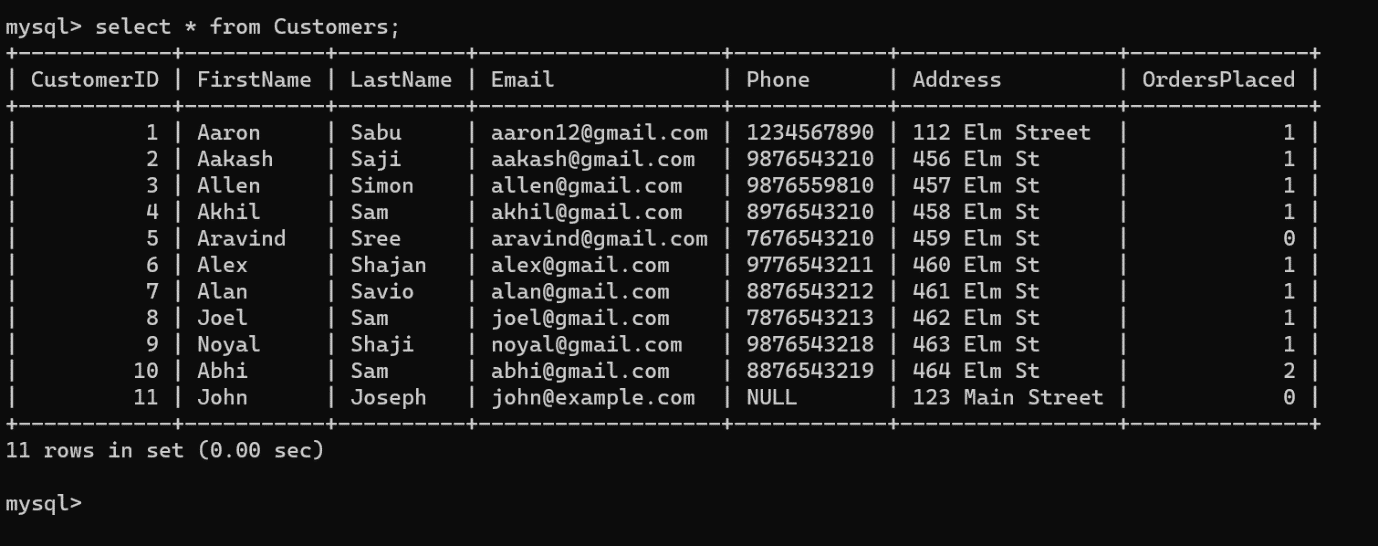
ALTER TABLE Customers

ADD COLUMN OrdersPlaced INT;

UPDATE Customers

SET OrdersPlaced = ( SELECT COUNT(OrderID) FROM Orders

WHERE Orders.CustomerID = Customers.CustomerID );



Task-3

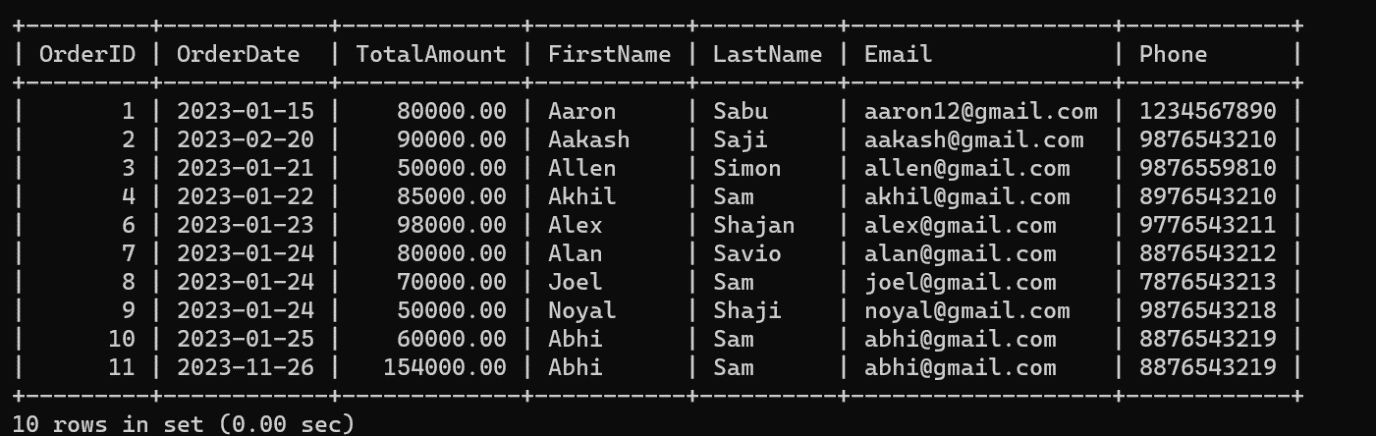
1)

SELECT Orders.OrderID, Orders. OrderDate, Orders. TotalAmount, Customers.FirstName,

-> Customers.LastName, Customers.Email, Customers.Phone

-> FROM Orders

-> JOIN Customers ON Orders.CustomerID = Customers.CustomerID;



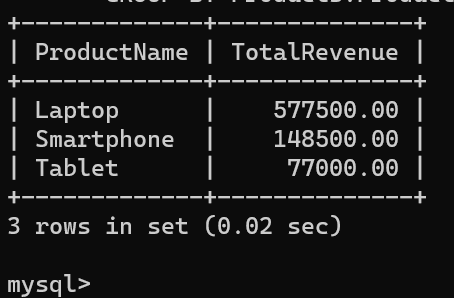
2)

SELECT Products.ProductName, SUM(OrderDetails.Quantity \* Products.Price) AS TotalRevenue

-> FROM OrderDetails

-> INNER JOIN Products ON OrderDetails.ProductID = Products.ProductID

-> GROUP BY Products.ProductName;



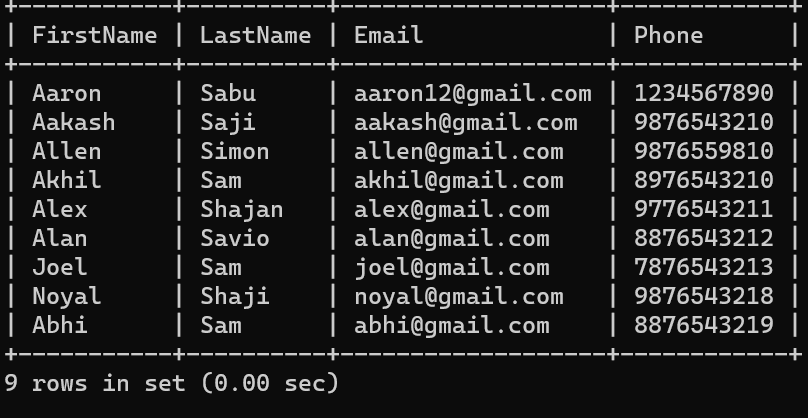
3)

SELECT DISTINCT Customers.FirstName, Customers.LastName, Customers.Email,

-> Customers.Phone

-> FROM Customers

-> INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID;



4)

mysql> SELECT Products.ProductName, SUM(OrderDetails.Quantity) AS TotalQuantityOrdered

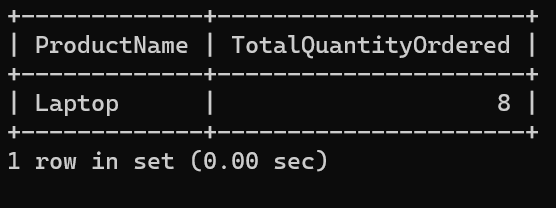
-> FROM OrderDetails

-> JOIN Products ON OrderDetails.ProductID = Products.ProductID

-> GROUP BY Products.ProductName

-> ORDER BY TotalQuantityOrdered DESC

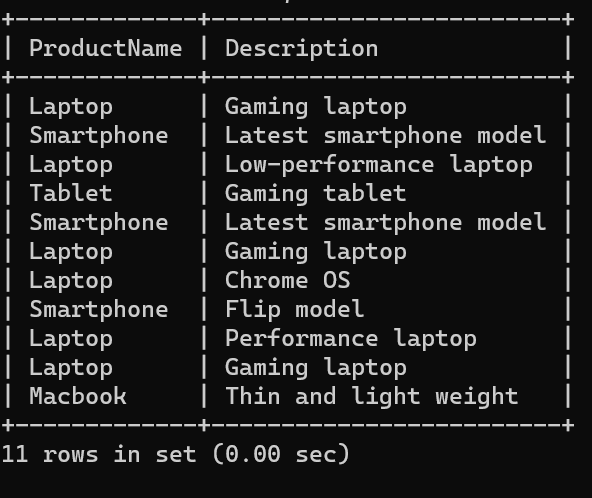
-> LIMIT 1;



5)

SELECT Products.ProductName, Products.Description

-> FROM Products;



6)

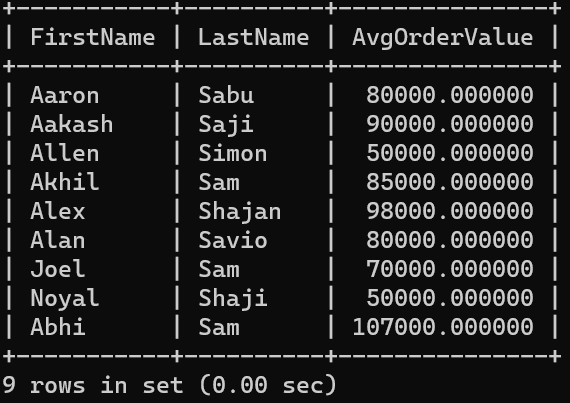
SELECT Customers.FirstName, Customers.LastName, AVG(Orders.TotalAmount) AS

-> AvgOrderValue

-> FROM Customers

-> INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID

-> GROUP BY Customers.FirstName, Customers.LastName;



7)

mysql> SELECT Orders.OrderID, Customers.FirstName, Customers.LastName,

-> SUM(OrderDetails.Quantity \* Products.Price) AS TotalRevenue

-> FROM Orders

-> INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID

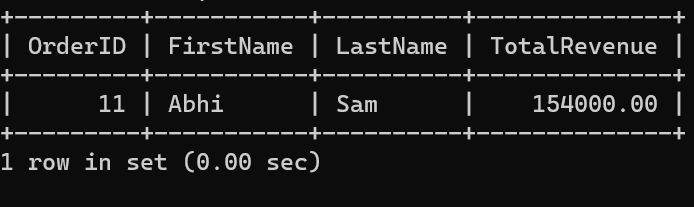
-> JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

-> JOIN Products ON OrderDetails.ProductID = Products.ProductID

-> GROUP BY Orders.OrderID, Customers.FirstName, Customers.LastName

-> ORDER BY TotalRevenue DESC

-> LIMIT 1;



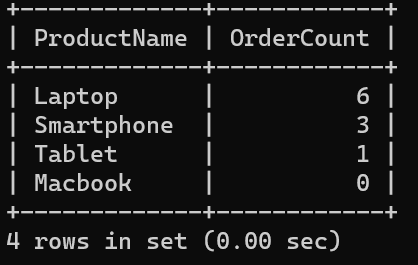
8)

SELECT Products.ProductName, COUNT(OrderDetails.OrderID) AS OrderCount

-> FROM Products

-> LEFT JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID

-> GROUP BY Products.ProductName;



9) DELIMITER $$

mysql> CREATE procedure get\_product(IN code varchar(100))

-> BEGIN

-> SELECT Customers.FirstName, Customers.LastName, Customers.Email, Customers.Phone

-> FROM Customers

-> JOIN Orders ON Customers.CustomerID = Orders.CustomerID

-> JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

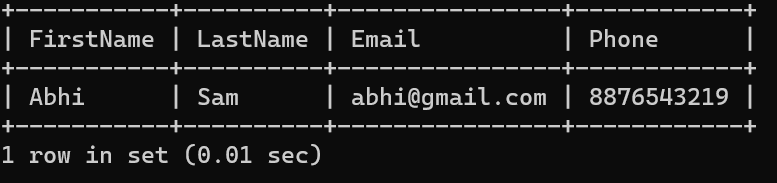
-> JOIN Products ON OrderDetails.ProductID = Products.ProductID

-> WHERE Products.ProductName = code;

-> END $$

DELIMITER ;

mysql> call get\_product("Laptop");



10)

mysql> DELIMITER $$

mysql> CREATE PROCEDURE get\_date(IN start\_date DATE, IN end\_date DATE)

-> BEGIN

-> SELECT SUM(Orders.TotalAmount) AS TotalRevenue

-> FROM Orders

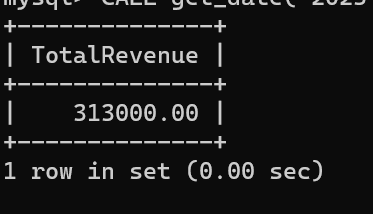
-> WHERE Orders.OrderDate BETWEEN start\_date AND end\_date;

-> END$$

Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;

mysql> CALL get\_date('2023-01-15', '2023-01-23');



Task 4

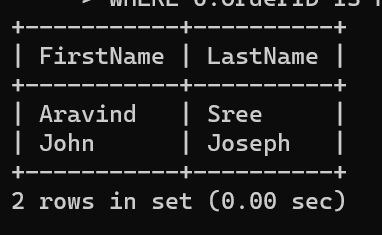
1)

SELECT c.FirstName, c.LastName

-> FROM Customers c

-> LEFT JOIN Orders o ON c.CustomerID = o.CustomerID

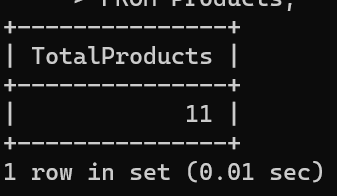
-> WHERE o.OrderID IS NULL;



2)

SELECT COUNT(\*) AS TotalProducts

-> FROM Products;



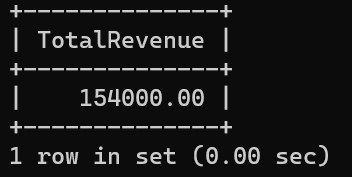
3)

SELECT SUM(od.Quantity \* p.Price) AS TotalRevenue

-> FROM Orders o

-> INNER JOIN OrderDetails od ON o.OrderID = od.OrderID

-> INNER JOIN Products p ON od.ProductID = p.ProductID;



4)

DELIMITER $$

mysql> CREATE PROCEDURE get\_name(IN name varchar(20))

-> BEGIN

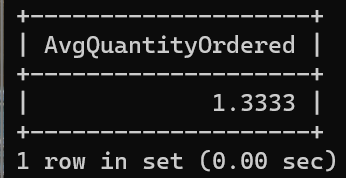
-> SELECT AVG(od.Quantity) AS AvgQuantityOrdered

-> FROM OrderDetails od

-> JOIN Products p ON od.ProductID = p.ProductID

-> WHERE p.ProductName = name;

-> END$$



5)

DELIMITER $$

mysql> CREATE PROCEDURE get\_id(IN id INT)

-> BEGIN

-> SELECT SUM(od.Quantity \* p.Price) AS TotalRevenue

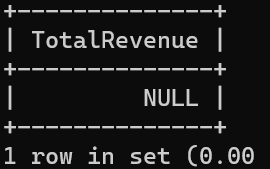
-> FROM Orders o

-> JOIN OrderDetails od ON o.OrderID = od.OrderID

-> JOIN Products p ON od.ProductID = p.ProductID

-> WHERE o.CustomerID = id;

-> END$$



6)

SELECT c.FirstName, c.LastName, COUNT(o.OrderID) AS OrderCount

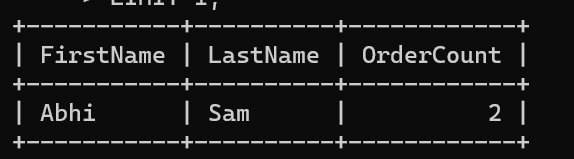
-> FROM Customers c

-> JOIN Orders o ON c.CustomerID = o.CustomerID

-> GROUP BY c.CustomerID

-> ORDER BY OrderCount DESC

-> LIMIT 1;



7)

SELECT p.ProductName, SUM(od.Quantity) AS TotalQuantityOrdered

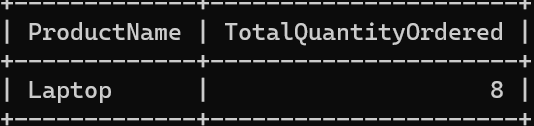
-> FROM OrderDetails od

-> JOIN Products p ON od.ProductID = p.ProductID

-> GROUP BY p.ProductName

-> ORDER BY TotalQuantityOrdered DESC

-> LIMIT 1;



8)

SELECT c.FirstName, c.LastName, SUM(od.Quantity \* p.Price) AS TotalSpending

-> FROM Customers c

-> JOIN Orders o ON c.CustomerID = o.CustomerID

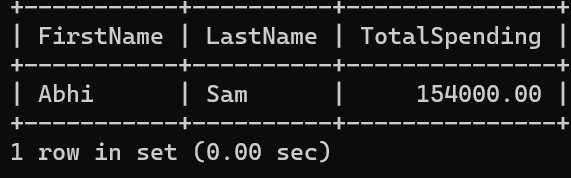
-> JOIN OrderDetails od ON o.OrderID = od.OrderID

-> JOIN Products p ON od.ProductID = p.ProductID

-> GROUP BY c.CustomerID

-> ORDER BY TotalSpending DESC

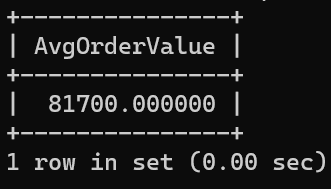
-> LIMIT 1;



9)

SELECT AVG(TotalAmount) AS AvgOrderValue

-> FROM Orders;



10)

SELECT c.FirstName, c.LastName, 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;

