# Task:1. Database Design:

1. Create the database named "TechShop"

CREATE DATABASE TechShop;

USE TechShop;

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 VARCHAR(50),

LastName VARCHAR(50),

Email VARCHAR(100),

Phone VARCHAR(20),

Address VARCHAR(50))

CREATE TABLE Products(

ProductID INT PRIMARY KEY,

ProductName VARCHAR(100),

Description VARCHAR(500),

Price INT

)

CREATE TABLE Orders(

OrderID INT PRIMARY KEY,

CustomerID INT ,

OrderDate DATE ,

TotalAmount INT ,

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 ,

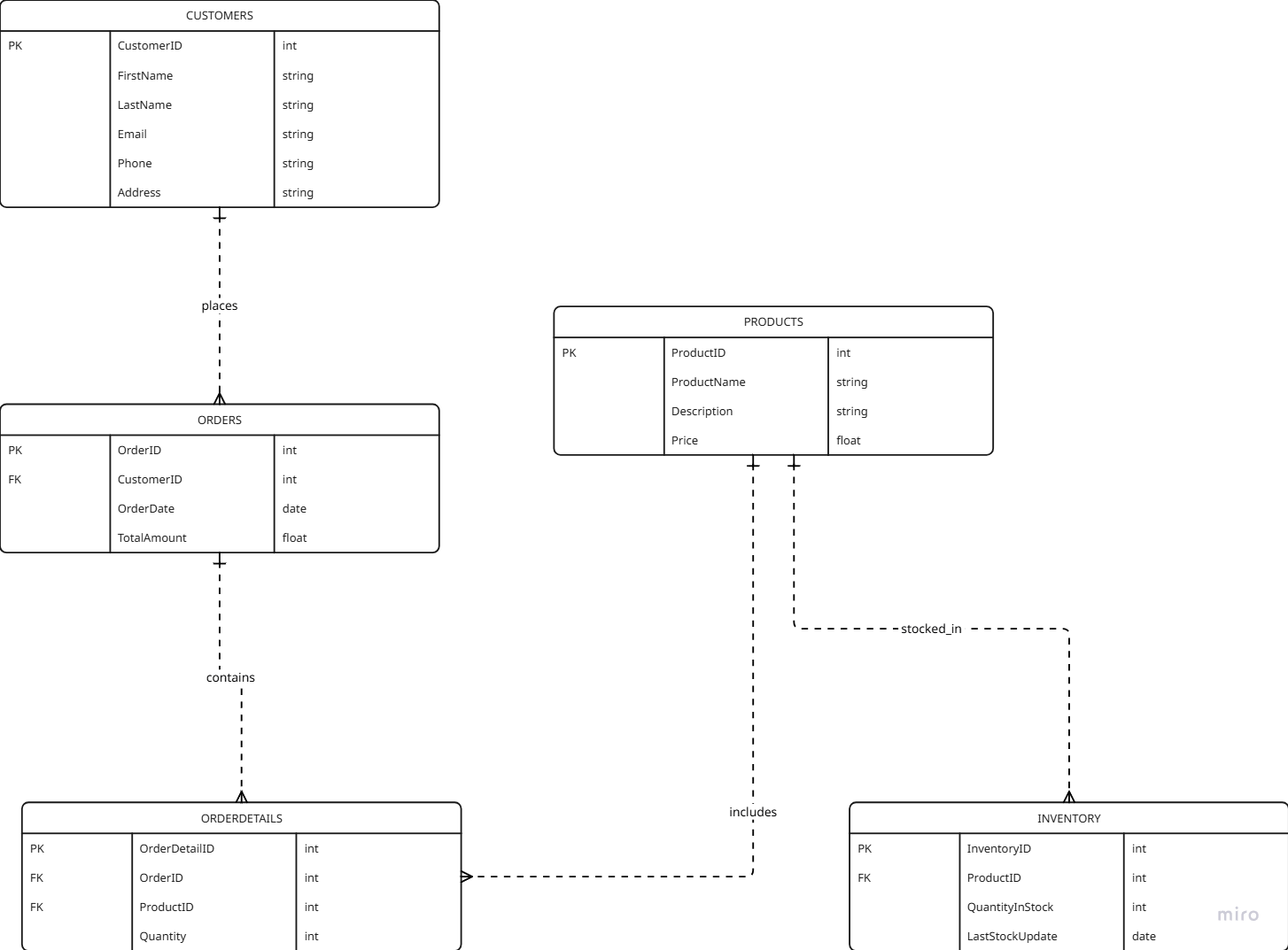
QuantityInStock INT,

LastStockUpdate INT,

FOREIGN KEY(productID) REFERENCES Products(productID)

)

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



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

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Email VARCHAR(100) UNIQUE,

Phone VARCHAR(15),

Address TEXT

);

CREATE TABLE Products (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(100),

Category VARCHAR(50),

Description TEXT,

Price DECIMAL(10,2)

);

CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

CustomerID INT,

OrderDate DATETIME DEFAULT GETDATE(),

TotalAmount DECIMAL(10,2),

Status VARCHAR(50) DEFAULT 'Pending',

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID) ON DELETE CASCADE

);

CREATE TABLE OrderDetails (

OrderDetailID INT PRIMARY KEY,

OrderID INT,

ProductID INT,

Quantity INT CHECK (Quantity > 0),

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) ON DELETE CASCADE,

FOREIGN KEY (ProductID) REFERENCES Products(ProductID) ON DELETE CASCADE

);

5. Insert at least 10 sample records into each of the following tables. a. Customers b. Products c. Orders d. OrderDetails

a:

INSERT INTO customers(CustomerID, FirstName, LastName, Email, Phone, Address) VALUES

(1, 'Amit', 'Sharma', 'amit.sharma@example.com', '9876543210', '123 MG Road, Mumbai'),

(2, 'Priya', 'Verma', 'priya.verma@example.com', '8765432109', '456 Park Street, Delhi'),

(3, 'Rahul', 'Gupta', 'rahul.gupta@example.com', '7654321098', '789 Brigade Road, Bangalore'),

(4, 'Sneha', 'Reddy', 'sneha.reddy@example.com', '6543210987', '321 Anna Nagar, Chennai'),

(5, 'Vikas', 'Yadav', 'vikas.yadav@example.com', '5432109876', '567 Banjara Hills, Hyderabad'),

(6, 'Pooja', 'Singh', 'pooja.singh@example.com', '4321098765', '890 Kothrud, Pune'),

(7, 'Arjun', 'Mehta', 'arjun.mehta@example.com', '3210987654', '234 Salt Lake, Kolkata'),

(8, 'Anjali', 'Chopra', 'anjali.chopra@example.com', '2109876543', '678 Gota, Ahmedabad'),

(9, 'Rohan', 'Deshmukh', 'rohan.deshmukh@example.com', '1098765432', '901 Indiranagar, Bangalore'),

(10, 'Kavita', 'Iyer', 'kavita.iyer@example.com', '0987654321', '345 Jayanagar, Bangalore');

b: Products

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

(1, 'Laptop', 'Dell Inspiron 15', 55000.00),

(2, 'Smartphone', 'Samsung Galaxy S21', 75000.00),

(3, 'Headphones', 'Sony WH-1000XM4', 15000.00),

(4, 'Tablet', 'Apple iPad 10.2-inch', 35000.00),

(5, 'Smartwatch', 'Apple Watch Series 7', 42000.00),

(6, 'Keyboard', 'Logitech Mechanical Keyboard', 7000.00),

(7, 'Mouse', 'HP Wireless Mouse', 1200.00),

(8, 'Monitor', 'LG 24-inch Full HD', 18000.00),

(9, 'Printer', 'Canon All-in-One Printer', 12000.00),

(10, 'External Hard Drive', 'WD 1TB Hard Drive', 5000.00);

C: Orders

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

(1, 1, '2024-03-01', 55000.00),

(2, 2, '2024-03-02', 75000.00),

(3, 3, '2024-03-03', 15000.00),

(4, 4, '2024-03-04', 35000.00),

(5, 5, '2024-03-05', 42000.00),

(6, 6, '2024-03-06', 7000.00),

(7, 7, '2024-03-07', 1200.00),

(8, 8, '2024-03-08', 18000.00),

(9, 9, '2024-03-09', 12000.00),

(10, 10, '2024-03-10', 5000.00);

D: OrderDetails

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

(1, 1, 1, 1),

(2, 2, 2, 1),

(3, 3, 3, 1),

(4, 4, 4, 1),

(5, 5, 5, 1),

(6, 6, 6, 1),

(7, 7, 7, 1),

(8, 8, 8, 1),

(9, 9, 9, 1),

(10, 10, 10, 1);

E: Inventory

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

VALUES

(1, 1, 50, '2025-03-01'),

(2, 2, 30, '2025-03-02'),

(3, 3, 100, '2025-03-03'),

(4, 4, 25, '2025-03-04'),

(5, 5, 60, '2025-03-05'),

(6, 6, 80, '2025-03-06'),

(7, 7, 15, '2025-03-07'),

(8, 8, 40, '2025-03-08'),

(9, 9, 70, '2025-03-09'),

(10, 10, 90, '2025-03-10');

TASK 2 -Select, Where, Between, AND, LIKE:

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

SELECT Firstname,Lastname,email FROM customers;

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

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

FROM Orders, Customers

WHERE Orders.CustomerID = customers.CustomerID ;

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

INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone, Address)

VALUES (11, 'Raj', 'Kapoor', 'raj.kapoor@example.com', '9876543211', '789 MG Road, Mumbai');

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

UPDATE Products

SET Price = Price \* 1.10

WHERE ProductName LIKE '%Laptop%'

OR ProductName LIKE '%Smartphone%'

OR ProductName LIKE '%Headphones%'

OR ProductName LIKE '%Tablet%'

OR ProductName LIKE '%Smartwatch%'

OR ProductName LIKE '%Keyboard%'

OR ProductName LIKE '%Mouse%'

OR ProductName LIKE '%Monitor%'

OR ProductName LIKE '%Printer%'

OR ProductName LIKE '%Hard Drive%';

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.

DECLARE @OrderID INT = 9

DELETE FROM OrderDetails

WHERE OrderID = @OrderID;

DELETE FROM Orders

WHERE OrderID = @OrderID;

SELECT \* FROM Orders

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(17,9,'2025-03-10',2000);

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.

DECLARE @new\_address VARCHAR(100)='SHIVAJI NAGAR YAWAL'

DECLARE @new\_email VARCHAR(100)='kavita@example.com'

DECLARE @customer\_id INT =10;

UPDATE customers

SET Address=@new\_address,email=@new\_email

WHERE CustomerID=@customer\_id;

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

SET TotalAmount = (

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

FROM OrderDetails OD

JOIN Products P ON OD.ProductID = P.ProductID

WHERE OD.OrderID = Orders.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.

DECLARE @CustomerID INT = 1; -- Set dynamically by user

DELETE FROM OrderDetails

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

DELETE FROM Orders

WHERE CustomerID = @CustomerID;

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, Category, Price, Description)

VALUES (101, 'Smartphone', 'Electronics', 29999, 'Latest 5G smartphone with 128GB storage.');

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.

DECLARE @OrderID INT = 5; -- User input

DECLARE @NewStatus VARCHAR(50) = 'Shipped'; -- User input

UPDATE Orders

SET Status = @NewStatus

WHERE OrderID = @OrderID;

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

SET NumberOfOrders = (

SELECT COUNT(\*)

FROM Orders

WHERE Orders.CustomerID = Customers.CustomerID

);