**Assignment : TechShop, an electronic gadgets shop**

Scenario : You are working as a database administrator for a fictional company named "TechShop," which sells electronic gadgets. TechShop maintains data related to their products, customers, and orders. Your task is to design and implement a database for TechShop based on the following requirements:

**Task:1 Database Design**

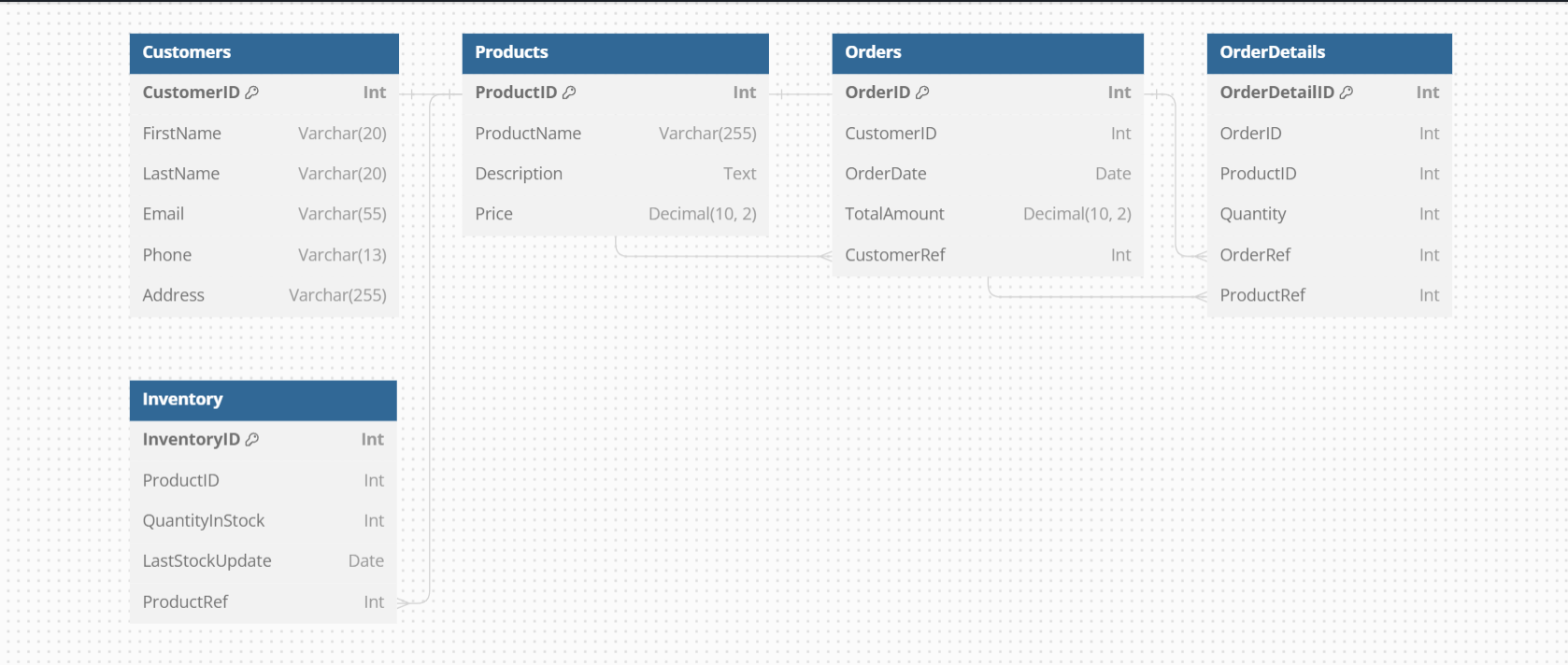
1. Create the database named "TechShop"

Query:-

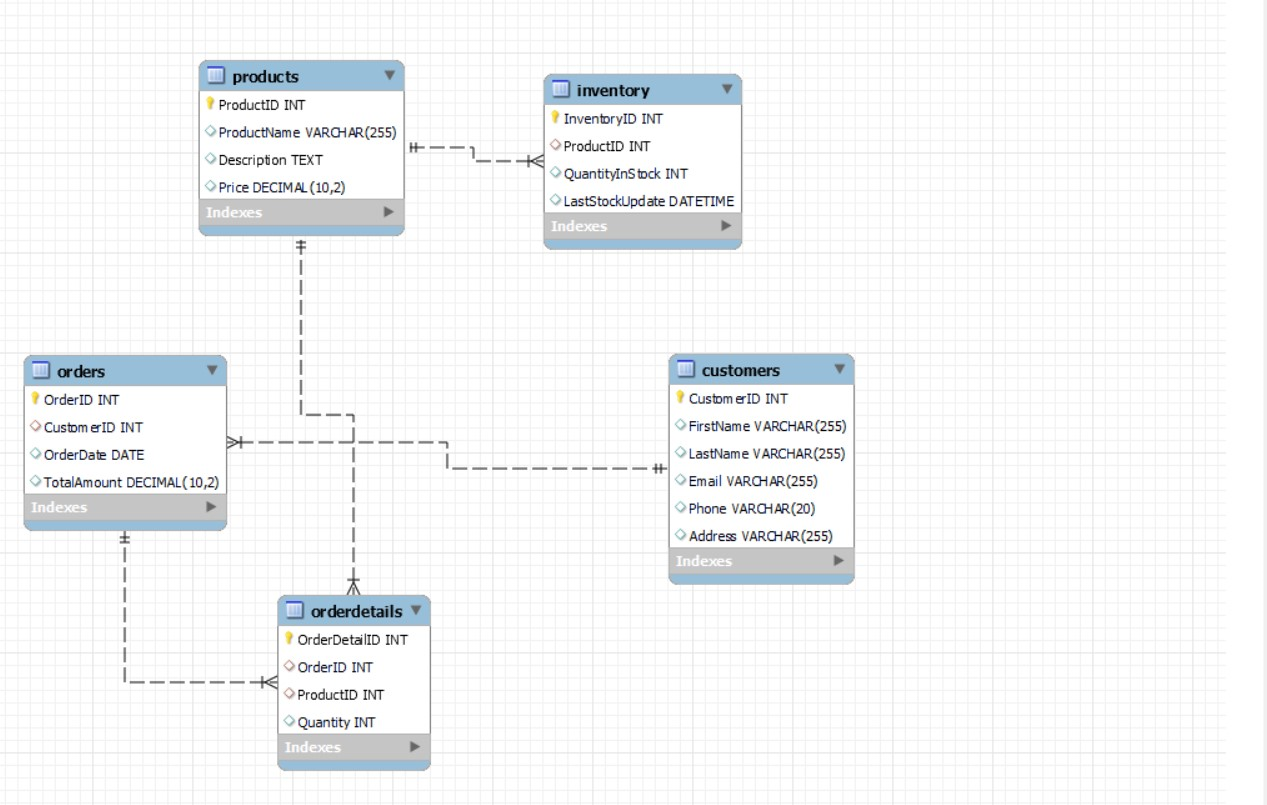
1.CREATE DATABASE TechShop;

2.USE TechShop;

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



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



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

Already implemented in above querys .

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

a. Customers

b. Products

c. Orders

d. OrderDetails

e. Inventory

Query:-

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

VALUES

(1, 'John', 'Doe', 'john.doe@email.com', '123-456-7890', '123 Main St'),

(2, 'Jane', 'Smith', 'jane.smith@email.com', '987-654-3210', '456 Oak St'),

(3, 'Bob', 'Johnson', 'bob.johnson@email.com', '555-123-4567', '789 Elm St'),

(4, 'Alice', 'Williams', 'alice.williams@email.com', '222-333-4444', '101 Pine Ave'),

(5, 'Charlie', 'Brown', 'charlie.brown@email.com', '777-888-9999', '202 Oak Ave'),

(6, 'Eva', 'Davis', 'eva.davis@email.com', '111-222-3333', '303 Maple St'),

(7, 'Frank', 'Moore', 'frank.moore@email.com', '444-555-6666', '404 Cedar Ave'),

(8, 'Grace', 'Taylor', 'grace.taylor@email.com', '666-777-8888', '505 Cedar St'),

(9, 'Henry', 'White', 'henry.white@email.com', '999-000-1111', '606 Elm Ave'),

(10, 'Ivy', 'Anderson', 'ivy.anderson@email.com', '123-789-4560', '707 Pine St');

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

VALUES

(1, 'Laptop', 'High-performance laptop with advanced features', 999.99),

(2, 'Smartphone', 'Latest smartphone with cutting-edge technology', 599.99),

(3, 'Tablet', 'Portable tablet for on-the-go computing', 299.99),

(4, 'Smartwatch', 'Wearable smartwatch with health tracking', 149.99),

(5, 'Headphones', 'Noise-canceling headphones for immersive audio', 129.99),

(6, 'Camera', 'Digital camera with high-resolution imaging', 799.99),

(7, 'Printer', 'All-in-one printer for home and office use', 199.99),

(8, 'Gaming Console', 'Latest gaming console for an immersive gaming experience', 499.99),

(9, 'Fitness Tracker', 'Health and fitness tracker for active lifestyles', 79.99),

(10, 'External Hard Drive', 'High-capacity external hard drive for data storage', 129.99);

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

VALUES

(1, 1, '2023-01-15', 999.99),

(2, 3, '2023-02-20', 149.99),

(3, 5, '2023-03-25', 299.99),

(4, 7, '2023-04-10', 799.99),

(5, 2, '2023-05-05', 599.99),

(6, 4, '2023-06-15', 199.99),

(7, 6, '2023-07-20', 499.99),

(8, 8, '2023-08-25', 129.99),

(9, 10, '2023-09-10', 79.99),

(10, 9, '2023-10-05', 129.99);

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

VALUES

(1, 1, 1, 1),

(2, 2, 5, 2),

(3, 3, 3, 1),

(4, 4, 6, 1),

(5, 5, 2, 1),

(6, 6, 7, 1),

(7, 7, 8, 1),

(8, 8, 10, 1),

(9, 9, 9, 3),

(10, 10, 4, 1);

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

VALUES

(1, 1, 20, '2023-01-01'),

(2, 2, 15, '2023-01-01'),

(3, 3, 25, '2023-01-01'),

(4, 4, 10, '2023-01-01'),

(5, 5, 30, '2023-01-01'),

(6, 6, 5, '2023-01-01'),

(7, 7, 12, '2023-01-01'),

(8, 8, 8, '2023-01-01'),

(9, 9, 18, '2023-01-01'),

(10, 10, 22, '2023-01-01');

**Tasks 2:**  Select, Where, Between, AND, LIKE:

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

Query:-

SELECT FirstName, LastName, Email FROM Customers;

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

Query:-

SELECT Orders.OrderID, OrderDate, FirstName, LastName

FROM Orders

JOIN Customers ON 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.

Query:-

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

VALUES (11, 'New', 'Customer', 'new.customer@email.com', '111-222-3333', '789 Pine St');

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

Query:-

UPDATE Products SET Price = Price \* 1.1 WHERE condtition ;

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.

Query:-

DELETE FROM OrderDetails WHERE OrderID = 2;

DELETE FROM Orders WHERE OrderID = 3;

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.

Query:-

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

VALUES (101,1, '2023-12-07', 3000);

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.

Query:-

SET Email = 'new.email@email.com', Address = xyz@gmail.com'

WHERE CustomerID = 3;

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.

Query:-

DELETE FROM OrderDetails WHERE OrderID IN (SELECT OrderID FROM Orders WHERE CustomerID = 3);

DELETE FROM Orders WHERE CustomerID = 5;

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.

Query:-

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

VALUES (101, 'New Gadget', 'Description of the new gadget', 199.99);

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.:

Query:-

UPDATE Orders SET Status = "Shipped" WHERE OrderID = 8;

**Task 3:** Aggregate functions, Having, Order By, Group By, Joins:

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

Query:-

SELECT Orders.OrderID, OrderDate, FirstName, LastName

FROM Orders

JOIN Customers ON Orders.CustomerID = Customers.CustomerID;

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

Query:-

SELECT ProductName, SUM(TotalAmount) AS TotalRevenue

FROM Products

JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID

JOIN Orders ON OrderDetails.OrderID = Orders.OrderID

GROUP BY Products.ProductID, ProductName;

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

Query:-

List all customers who have made at least one purchase:

SELECT DISTINCT FirstName, LastName, Email

FROM Customers

JOIN Orders ON Customers.CustomerID = Orders.CustomerID;

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.

Query:-

SELECT ProductName, SUM(Quantity) AS TotalQuantityOrdered

FROM Products

JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID

GROUP BY Products.ProductID, ProductName

ORDER BY TotalQuantityOrdered DESC

LIMIT 1;

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

Query:-

SELECT ProductName, Category

FROM Products;

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

Query:-

SELECT Orders.OrderID, OrderDate, FirstName, LastName, SUM(TotalAmount) AS TotalRevenue

FROM Orders

JOIN Customers ON Orders.CustomerID = Customers.CustomerID

JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

GROUP BY Orders.OrderID, OrderDate, FirstName, LastName

ORDER BY TotalRevenue DESC

LIMIT 1;

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

Query:-

SELECT ProductName, COUNT(OrderDetails.OrderDetailID) AS OrderCount

FROM Products

LEFT JOIN OrderDetails ON Products.ProductID =8;

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 .

Query:-

DECLARE @ProductName NVARCHAR(255);

SET @ProductName = 'Headphone';

SELECT

C.CustomerID,

CONCAT(C.FirstName, ' ', C.LastName) AS CustomerName,

O.ProductName

FROM

Customers C

JOIN

Orders O ON C.CustomerID = O.CustomerID

WHERE

O.ProductName = @ProductName;

10.write an sql query to calculate the total revenue generated by all order placed within a specific time period . allow users to input the start and dates as parameters .

Query:-

DECLARE @StartDate DATE;

DECLARE @EndDate DATE;

SET @StartDate = '2023-01-01';

SET @EndDate = '2023-12-31';

SELECT

SUM(Amount) AS TotalRevenue

FROM

Orders

WHERE

OrderDate BETWEEN @StartDate AND @EndDate

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

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

Query:-

SELECT CustomerID, FirstName, LastName

FROM Customers

WHERE CustomerID NOT IN (SELECT DISTINCT CustomerID FROM Orders);

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

Query:-

SELECT COUNT(ProductID) AS TotalProducts

FROM Products;

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

Query:-

SELECT SUM(TotalAmount) AS TotalRevenue

FROM Orders;

4. Write an SQL query to calculate the average quantity ordered for products in a specific category.

Query:-

SELECT AVG(Quantity) AS AvgQuantityOrdered

FROM OrderDetails

WHERE ProductID IN (SELECT ProductID FROM Products WHERE Category = 'Electronics');

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.

Query:-

SELECT SUM(TotalAmount) AS TotalRevenue

FROM Orders

WHERE CustomerID = 1;

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.

Query:-

SELECT CustomerID, FirstName, LastName, COUNT(OrderID) AS OrderCount

FROM Orders

JOIN Customers ON Orders.CustomerID = Customers.CustomerID

GROUP BY CustomerID

ORDER BY OrderCount DESC

LIMIT 1;

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.

Query:-

SELECT Category, SUM(Quantity) AS TotalQuantityOrdered

FROM Products

JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID

GROUP BY Category

ORDER BY TotalQuantityOrdered DESC

LIMIT 1;

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.

Query:-

SELECT CustomerID, FirstName, LastName, SUM(TotalAmount) AS TotalSpending

FROM Orders

JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

JOIN Products ON OrderDetails.ProductID = Products.ProductID

WHERE Products.Category = 'Electronics'

GROUP BY CustomerID

ORDER BY TotalSpending DESC

LIMIT 1;

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

Query:-

SELECT AVG(TotalAmount) AS AvgOrderValue

FROM Orders;

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 .

Query:-

SELECT

C.CustomerID,

CONCAT(C.FirstName, ' ', C.LastName) AS CustomerName,

COUNT(O.OrderID) AS OrderCount

FROM

Customers C

LEFT JOIN

Orders O ON C.CustomerID = O.CustomerID

GROUP BY

C.CustomerID, CustomerName

ORDER BY

OrderCount DESC;