

## ASSIGNMENT 1

### Task:1. Database Design:

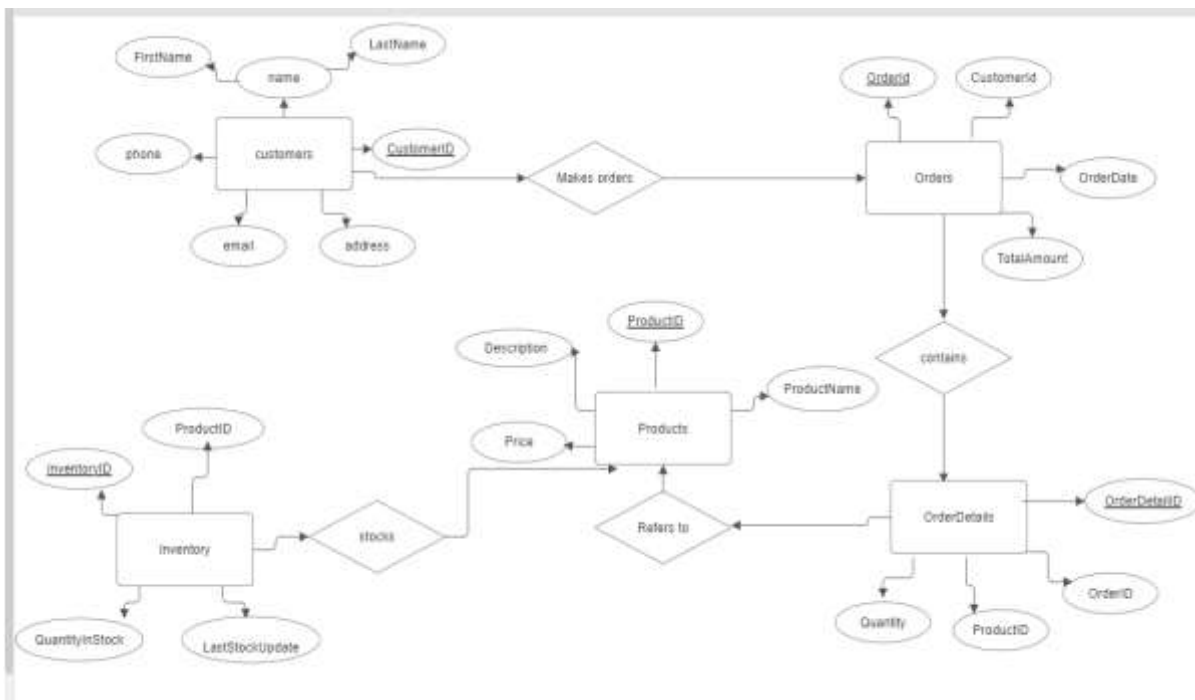
1) create database techshop;

use techshop;

2)

- I. create table customers (customerid int primary key, Firstname varchar(15), Lastname varchar(15), email varchar(25), phone varchar (15), address varchar(50));
- II. create table orders (orderid int primary key, customerid int, orderdate date, totalamount int, foreign key(customerid) references customers(customerid));
- III. create table products( productid int primary key, productname varchar(50), description varchar(200), price decimal(10,2));
- IV. 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));
- V. create table inventory ( inventoryid int primary key, productid int, quantityInStock int, laastStockUpdate date, foreign key (productid) references products(productid));

3) ER Diagram



5)

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

(1, 'John', 'Doe', 'john.doe@example.com', '123-456-7890', '123 Main St'),  
(2, 'Alice', 'Smith', 'alice.smith@example.com', '987-654-3210', '456 Oak Ave'),  
(3, 'Bob', 'Johnson', 'bob.johnson@example.com', '555-123-4567', '789 Elm St'),  
(4, 'Emily', 'Brown', 'emily.brown@example.com', '444-222-3333', '101 Pine St'),  
(5, 'Michael', 'Jones', 'michael.jones@example.com', '999-888-7777', '202 Maple Ave'),  
(6, 'Sara', 'Taylor', 'sara.taylor@example.com', '777-666-5555', '303 Cedar St'),  
(7, 'David', 'Wilson', 'david.wilson@example.com', '222-333-4444', '404 Birch St'),  
(8, 'Emma', 'Martinez', 'emma.martinez@example.com', '111-222-3333', '505 Walnut Ave'),  
(9, 'James', 'Garcia', 'james.garcia@example.com', '666-555-4444', '606 Oak St'),  
(10, 'Olivia', 'Lopez', 'olivia.lopez@example.com', '333-444-5555', '707 Elm Ave');

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

(1, 1, '2024-04-25', 1499.98),  
(2, 2, '2024-04-26', 899.99),  
(3, 3, '2024-04-27', 799.99),  
(4, 4, '2024-04-28', 249.99),  
(5, 5, '2024-04-29', 149.99),  
(6, 6, '2024-04-30', 79.99),  
(7, 7, '2024-05-01', 89.99),  
(8, 8, '2024-05-02', 499.99),  
(9, 9, '2024-05-03', 699.99),  
(10, 10, '2024-05-04', 29.99);

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

(1, 'Laptop', '15" Intel Core i7, 16GB RAM, 512GB SSD', 1299.99),  
(2, 'Smartphone', '6.5" AMOLED, 128GB Storage, Snapdragon 888', 899.99),  
(3, 'Tablet', '10.5" iPad Pro, 256GB Storage, Wi-Fi + Cellular', 799.99),  
(4, 'Headphones', 'Over-Ear Noise-Canceling Bluetooth Headphones', 249.99),  
(5, 'Smartwatch', 'Fitness Tracker with Heart Rate Monitor', 149.99),  
(6, 'Bluetooth Speaker', 'Portable Waterproof Speaker with 20W Output', 79.99),  
(7, 'External Hard Drive', '2TB USB 3.0 Portable Hard Drive', 89.99),  
(8, 'Gaming Console', 'Next-Gen Gaming Console with 4K HDR', 499.99),

(9, 'Digital Camera', '20MP DSLR Camera with 4K Video Recording', 699.99),

(10, 'Wireless Mouse', 'Ergonomic Wireless Mouse with Silent Clicks', 29.99);

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

(1, 1, 1, 1),

(2, 1, 2, 2),

(3, 2, 3, 1),

(4, 2, 4, 1),

(5, 3, 5, 1),

(6, 3, 6, 2),

(7, 4, 7, 1),

(8, 5, 8, 1),

(9, 6, 9, 1),

(10, 7, 10, 1);

INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LaastStockUpdate) VALUES

(1, 1, 10, '2024-04-26'),

(2, 2, 20, '2024-04-26'),

(3, 3, 15, '2024-04-26'),

(4, 4, 30, '2024-04-26'),

(5, 5, 25, '2024-04-26'),

(6, 6, 40, '2024-04-26'),

(7, 7, 12, '2024-04-26'),

(8, 8, 5, '2024-04-26'),

(9, 9, 8, '2024-04-26'),

(10, 10, 20, '2024-04-26');

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

1)select concat(firstname,lastname) as name,email from customers;

2) select concat(firstname,lastname) as name,orderdate from customers c,orders o where o.customerid=c.customerid;

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

VALUES (11,'John', 'Doe', 'john.doe@example.com', '123 Main St');

4)update products set price= price+(price\*0.1);

5) delete from orderDetails where orderid=2;

delete from orders where orderid=2;

6) insert into orders values (2, 2, '2024-04-26', 899.99);

7)UPDATE Customers SET Email = 'newemail@example.com', Address = '456 Oak St' WHERE CustomerID = 2;

8)UPDATE Orders

SET TotalAmount = (

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

FROM OrderDetails

INNER JOIN Products ON OrderDetails.ProductID = Products.ProductID

WHERE OrderDetails.OrderID = Orders.OrderId);

9) DELETE FROM OrderDetails WHERE OrderID IN (SELECT OrderID FROM Orders WHERE CustomerID=4);

DELETE FROM Orders WHERE CustomerID = 4;

10)INSERT INTO Products (ProductName,productid, Price, Description)

VALUES ('oven', 11, 499.99, 'Description of the new gadget');

11) alter table orders add status varchar(23);

UPDATE Orders

SET Status = 'shipped'

WHERE OrderID IN (7,8,9,10);

UPDATE Orders

SET Status = 'shipped'

WHERE OrderID 1;

12) alter table customers add noOfOrder int;

UPDATE Customers AS c

SET noOfOrder = (SELECT COUNT(\*) FROM Orders AS o WHERE o.CustomerID = c.CustomerID);

### Task 3. Aggregate functions, Having, Order By, GroupBy and Joins:

1) select \* from customers c,orders o where o.customerid=c.customerid;

2) SELECT Products.ProductName, SUM(Orders.TotalAmount) AS TotalRevenue

FROM OrderDetails

JOIN Products ON OrderDetails.ProductID = Products.ProductID

JOIN Orders ON OrderDetails.OrderID = Orders.OrderID

GROUP BY Products.ProductName;

3) select firstname,lastname,email,phone from customers where nooforder>0;

(or)

SELECT FirstName, LastName, Email, Phone

FROM Customers

WHERE CustomerID IN (SELECT DISTINCT CustomerID FROM Orders);

4)select productname,sum(quantity) as highestquantity from products p,orderdetails o where p.productid=o.productid group by p

.productname order by highestquantity desc limit 1;

5) select productname,description from products where description ="electronic gadget";

6) select concat(c.firstname ,c.lastname ) as name,avg(o.totalamount) from customers c, orders o where o.customerid=c.customerid group by name;

7)select orderid,concat(c.firstname,c.lastname) as name, phone,email max(totalamount) from orders o,customers c where c.customerid=o.customerid group by name;

8) select productname,count(orderid) from orderdetails o,products p where o.productid=p.productid group by productname;

9) SELECT CONCAT(c.FirstName, ' ', c.LastName) AS Name, p.ProductName

-> FROM OrderDetails o

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

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

-> JOIN Customers c ON od.CustomerID = c.CustomerID

-> WHERE p.ProductName = 'Laptop';

10)select sum(totalamount) from orders between 2024-04-25 and 2024-04-28;

## Task 4. Subquery and its type:

1) select concat(firstname,lastname) as name from customers where customerid in(select customerid from customers);

2) select productname from products where productid in(select productidfrom inventory);

3) SELECT SUM(TotalAmount) AS TotalRevenue FROM Orders;

4)SELECT p.productname, avg(od.quantity) FROM orderdetails od ,products p where od.productid = p.productid GROUP BY p.productname;

5)SELECT c.customerid, SUM(o.totalamount)  
FROM customers c , orders o where c.customerid = o.customerid  
GROUP BY c.customerid  
HAVING c.customerid IN (SELECT customerid FROM orders);

6) SELECT CONCAT(c.firstname, c.lastname) AS name, COUNT(o.orderid) AS order\_count, c.customerid  
FROM customers c  
JOIN orders o ON o.customerid = c.customerid  
GROUP BY c.customerid  
ORDER BY order\_count DESC  
LIMIT 1;

7)SELECT Category  
FROM Products  
GROUP BY Category  
ORDER BY SUM(Quantity) DESC  
LIMIT 1;

8)SELECT CONCAT(FirstName, LastName) AS CustomerName, MAX(TotalAmount) AS TotalSpending  
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.description = 'Electronic Gadget'
```

```
GROUP BY Customers.CustomerID
```

```
ORDER BY TotalSpending DESC
```

```
LIMIT 1;
```

```
9)SELECT AVG(TotalAmount) AS AvgOrderValue FROM Orders;
```

```
10)SELECT CONCAT(FirstName, LastName) AS CustomerName, COUNT(*) AS OrderCount
```

```
FROM Customers c, Orders o
```

```
where c.CustomerID = o.CustomerID
```

```
GROUP BY C.CustomerID;
```