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

**Task:1. Database Design:**

**1. Create the database named "TechShop" ?**

Ans. create database TechShop;

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

Ans. CREATE TABLE Customers (

CustomerID int primary key auto\_increment,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Email VARCHAR(100),

Phone VARCHAR(20),

Address VARCHAR(255)

);

create table Products(

ProductId int primary key auto\_increment,

ProductName varchar(100),

Description text,

Price int

);

create table Orders(

OrderId int primary key auto\_increment,

CustomerId int,

OrderDate date,

TotalAmount int,

foreign key (CustomerId) references Customers(CustomerId) ON DELETE CASCADE

);

create table OrderDetails(

OrderDetailId int primary key auto\_increment,

OrderId int,

ProductId int,

Quantity int,

foreign key (OrderId) references Orders(OrderId) ON DELETE CASCADE ,

foreign key (ProductId) references Products(ProductId)ON DELETE CASCADE

);

create table Inventory(

InventoryId int primary key auto\_increment,

ProductId int,

QuantityInStock int,

LastStockUpdate date,

foreign key (ProductId) references Products(ProductId)ON DELETE CASCADE

);

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

Ans.



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

Ans. ALTER TABLE Customers

ADD CONSTRAINT PK\_Customers PRIMARY KEY (CustomerID);

ALTER TABLE Products

ADD CONSTRAINT PK\_Products PRIMARY KEY (ProductID);

ALTER TABLE Orders

ADD CONSTRAINT PK\_Orders PRIMARY KEY (OrderID);

ALTER TABLE Orders

ADD CONSTRAINT FK\_Orders\_Customers

FOREIGN KEY (CustomerID)

REFERENCES Customers(CustomerID);

ALTER TABLE OrderDetails

ADD CONSTRAINT PK\_OrderDetails PRIMARY KEY (OrderDetailID);

ALTER TABLE OrderDetails

ADD CONSTRAINT FK\_OrderDetails\_Orders

FOREIGN KEY (OrderID)

REFERENCES Orders(OrderID);

ALTER TABLE OrderDetails

ADD CONSTRAINT FK\_OrderDetails\_Products

FOREIGN KEY (ProductID)

REFERENCES Products(ProductID);

ALTER TABLE Inventory

ADD CONSTRAINT PK\_Inventory PRIMARY KEY (InventoryID);

ALTER TABLE Inventory

ADD CONSTRAINT FK\_Inventory\_Products

FOREIGN KEY (ProductID)

REFERENCES Products(ProductID);

**5.Create appropriate Primary Key and Foreign Key constraints for referential integrity.**

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

**a. Customers**

**b. Products**

**c. Orders**

**d. OrderDetails**

**e. Inventory**

Ans. insert into Customers(FirstName,LastName,Email,Phone,Address) values

('Raj','Mohan','raj@hexa.com','9154876238','Local Colony 1'),

('Aman','Panwar','aman@hexa.com','9154832588','Local Colony 2'),

('Ajay','Meena','ajay@hexa.com','9179436238','Local Colony 3'),

('Raghav','kandyal','raghav@hexa.com','9154346938','Local Colony 4'),

('Sohan','Kala','sohan@hexa.com','9175976238','Local Colony 5'),

('Ritik','Jain','ritik@hexa.com','9154873594','Local Colony 6'),

('Raju','Sharma','raju@hexa.com','9154876594','Local Colony 7'),

('Rajesh','Mohan','rajesh@hexa.com','9154876615','Local Colony 8'),

('Komal','rawat','komal@hexa.com','9154876279','Local Colony 9'),

('Ranjan','Mehra','ranjan@hexa.com','9154876244','Local Colony 10')

;

insert into Products(ProductName,Description,Price) values

('Bulb','Light\_Gadget',50),

('Headphones','Sound\_Gadget',3999),

('Speaker','Sound\_Gadget',12999),

('Television','Display\_Gadget',30000),

('SmartWatch','Watch\_Gadget',6000),

('Screen','Display\_Gadget',8000),

('Mouse','Pointing\_Gadget',2000),

('Keyboard','Typing\_Gadget',4000),

('Laptop','All\_in\_one\_Gadget',50000),

('SmartPhone','Communication\_Gadget',10000)

;

insert into Orders(CustomerId,OrderDate,TotalAmount) values

(1,'2023-12-01',12569),

(2,'2023-12-02',1469),

(3,'2023-12-03',10000),

(4,'2023-12-04',50000),

(5,'2023-12-05',2000),

(6,'2023-12-06',1000),

(7,'2023-12-07',79462),

(8,'2023-12-08',13578),

(9,'2023-12-09',45689),

(10,'2023-12-10',2567)

;

insert into OrderDetails(OrderId,ProductId,Quantity) values

(1,1,2),

(2,2,4),

(3,3,3),

(4,4,1),

(5,5,6),

(6,6,3),

(7,7,4),

(8,8,7),

(9,9,5),

(10,10,3)

;

insert into Inventory(ProductId,QuantityInStock,LastStockUpdate) values

(1,100,'2023-12-01'),

(2,200,'2023-12-02'),

(3,350,'2023-12-03'),

(4,480,'2023-12-04'),

(5,50,'2023-12-05'),

(6,65,'2023-12-06'),

(7,73,'2023-12-07'),

(8,89,'2023-12-08'),

(9,91,'2023-12-09'),

(10,18,'2023-12-10')

;

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

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

Ans. select FirstName,Email from Customers;

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

Ans. select Orders.OrderDate,Customers.FirstName 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.**

Ans.insert into Customers(FirstName,LastName,Email,Phone,Address) values

('Zaz','Jolly','zaz@hexa.com','9185647239','Local Colony 11');

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

Ans. update Products Set Price=Price\*1.1;

**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.**

Ans. Set @OrderId=10;

delete from Orders where OrderId=@OrderId;

**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.**

Ans. insert into Orders(CustomerId,OrderDate,TotalAmount) values

(11,'2023-12-11',15964);

**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.**

Ans. Set @CustomerId=5,@Email='tino@hexware.com',@address='Colony 12';

update Customers set Email=@Email,Address=@address where CustomerId=@CustomerId;

**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.**

Ans. update Orders set TotalAmount=(

select Price\*Quantity

from OrderDetails join Products On Products.ProductId=OrderDetails.ProductId

WHERE Orders.OrderID = OrderDetails.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.**

Ans. SET @OrderId=9;

delete from Orders where OrderId=@OrderId;

**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.**

Ans. insert into Products(ProductName,Description,Price) values

('Bluetooth','Connect smoothly',500);

**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.**

Ans. Set @Statuss='shipped',@OrderId=5;

alter table Orders add Status varchar(50);

update Orders set Status=@Statuss 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.**

Ans. alter table Customers add OrderCount int;

update Customers set OrderCount=(select count(OrderId) from Orders where

Customers.CustomerID = Orders.CustomerId);

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

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

Ans. select \* from Customers join Orders on Customers.CustomerID = Orders.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.**

Ans. SELECT ProductName, Quantity \*Price AS TotalRevenue

FROM OrderDetails

JOIN Products ON OrderDetails.ProductID = Products.ProductID;

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

Ans. select FirstName,LastName,Phone,Email from Customers join Orders on Customers.CustomerID = Orders.CustomerID

group by Customers.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.**

Ans. SELECT ProductName, Quantity AS TotalQuantityOrdered

FROM OrderDetails

JOIN Products ON OrderDetails.ProductID = Products.ProductID

ORDER BY TotalQuantityOrdered DESC

limit 1;

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

Ans. SELECT ProductName,Description As Catogories from Products;

**6. Write an SQL query to calculate the average order value for each customer. Include the customer's name and their average order value.**

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

FROM Customers

JOIN Orders ON Customers.CustomerID = Orders.CustomerID

GROUP BY Customers.CustomerID, Customers.FirstName, Customers.LastName;

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

Ans. SELECT OrderId,Customers.FirstName, Customers.LastName, TotalAmount AS TotalRevenue

FROM Orders

JOIN Customers ON Orders.CustomerId = Customers.CustomerID

ORDER BY TotalAmount DESC

limit 1;

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

Ans. SELECT Products.ProductID, Products.ProductName, count(OrderDetails.OrderID) AS OrderCount

FROM Products

LEFT JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID

GROUP BY Products.ProductID, Products.ProductName;

**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.**

Ans. Set @ProductName='Bulb';

SELECT DISTINCT Customers.CustomerID, Customers.FirstName, Customers.LastName 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 =@ProductName;

**10. 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.**

Ans. Set @StartDate='2023-12-01',@EndDate='2023-12-04';

SELECT SUM(TotalAmount) 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.**

Ans. SELECT CustomerID, FirstName, LastName

FROM Customers

WHERE CustomerID NOT IN (SELECT CustomerID FROM Orders);

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

Ans. SELECT COUNT(\*) AS TotalProducts

FROM Products;

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

Ans. SELECT SUM(TotalAmount) AS TotalRevenue

FROM Orders;

**4. 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.**

Ans. Set @CatogoryName='Typing\_Gadget';

SELECT AVG(Quantity) AS AverageQuantityOrdered

FROM OrderDetails

WHERE ProductID IN (SELECT ProductID FROM Products WHERE Products.Description= @CatogoryName);

**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.**

Ans. Set @CustomerId=4;

Select TotalAmount as TotalRevenue

from Orders where CustomerId=@CustomerId;

**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.**

Ans. SELECT (

SELECT FirstName FROM Customers Where orders.CustomerId=CustomerId

) AS "Customer Name",

COUNT(orderid) as "Total Orders"

FROM orders GROUP BY customerid ORDER BY count(\*) 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.**

Ans. SELECT Description AS Catagory FROM Products

WHERE Products.ProductId=(SELECT ProductId FROM OrderDetails

ORDER BY OrderDetails.Quantity 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.**

Ans. SELECT FirstName,LastName,(Select TotalAmount from Orders where Orders.CustomerId=Customers.CustomerId) as Total\_spending FROM

Customers

WHERE Customers.CustomerId=(SELECT CustomerId FROM Orders

ORDER BY Orders.TotalAmount 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.**

Ans. SELECT FirstName,LastName,(SELECT (Orders.TotalAmount/(Select Quantity from OrderDetails WHERE Orders.OrderId=OrderDetails.OrderId)

) FROM

Orders WHERE Orders.CustomerID = Customers.CustomerID)

AS Average\_Order\_Value

FROM Customers;

**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.**

Ans. SELECT FirstName,LastName,

(SELECT Quantity FROM OrderDetails

WHERE OrderDetails.OrderId=(SELECT OrderId from Orders

Where Orders.CustomerId=Customers.CustomerId

)

)AS OrderCount

FROM Customers;