**Assignment on sql**

**TechShop, an electronic gadgets shop**

**Name:Jeffry Joshuva Amalan J**

**Task:1. Database Design:**

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

create database TechShop

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

create table Customers(CustomerID int not null primary key,FirstName varchar(50),LastName varchar(50),Email varchar(50), Phone varchar(50),Address varchar(50))

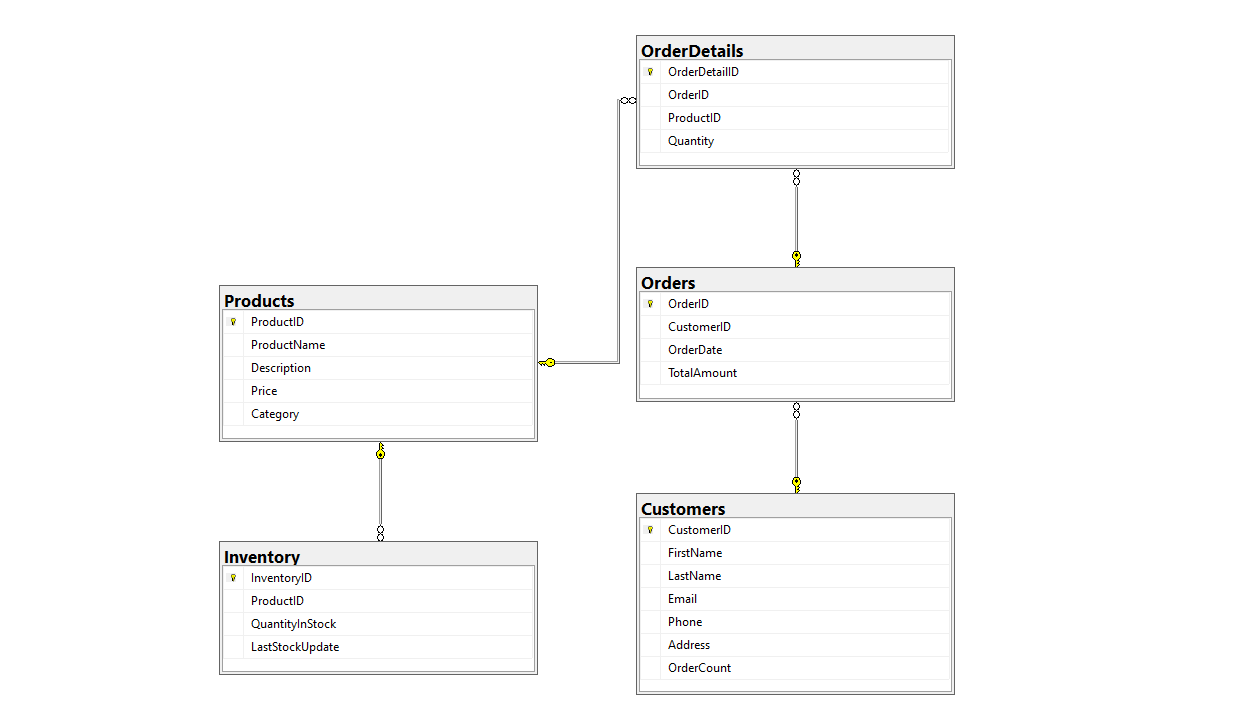
create table Products(ProductID int not null primary key,ProductName varchar(50),Description varchar(255),Price decimal(10,4))

create table Orders(OrderID int not null primary key,CustomerID int,OrderDate varchar(50),TotalAmount decimal(10,4),foreign key (CustomerID) references Customers(CustomerID))

create table OrderDetails(OrderDetailID int not null 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 not null primary key,ProductID int,QuantityInStock int,LastStockUpdate varchar(50), 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 Orders(OrderID int not null primary key,CustomerID int,OrderDate varchar(50),TotalAmount decimal(10,4),foreign key (CustomerID) references Customers(CustomerID))

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

**a. Customers**

**b. Products**

**c. Orders**

**d. OrderDetails**

**e. Inventory**

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

(1, 'Jeffry', 'Amalan', 'jeffryamalan@gmail.com', '7653678765', '12/B North street,chennai'),

(2, 'Saran', 'Raj', 'saran@gmail.com', '7543565456', '45-A East Road,chennai'),

(3, 'Kishore', 'Kumar', 'kishore@gmail.com', '8987566634', '7/A North street,bangalore'),

(4, 'Ram', 'Kumar', 'ram@gmail.com', '9098734554', '1st cross street,kovilpatti'),

(5, 'Jeyanth', 'Raja', 'raja@gmail.com', '8788765456', '12C 2nd street,Erode'),

(6, 'Jeya', 'Prince', 'prince@gmail.com', '7689465476', '5-c,South street, Nazareth'),

(7, 'Yoga', 'Raj', 'yoga@gmail.com', '7890123456', '159 church street, bangalore'),

(8, 'Joshua', 'Samuel', 'joshuasamuel@gmail.com', '8901234567', '8 north street,chennai'),

(9, 'Prem', 'kumar', 'prem@gmail.com', '9012345678', '3rd street,Nagarkovil'),

(10, 'Samuel', 'Rajkumar', 'samuel@gmail.com', '9123490789', '33/C 2nd cross street, erode');

insert into Products(ProductID, ProductName, Description, Price) values

(201, 'Laptop', '15-inch screen, 8GB RAM', 750.9999),

(202, 'Mouse', 'Wireless optical mouse', 15.4999),

(203, 'Keyboard', 'Mechanical keyboard', 45.2500),

(204, 'Monitor', '24-inch LED monitor', 120.0000),

(205, 'Smartphone', '64GB storage, Android', 299.9900),

(206, 'Headphones', 'Noise-canceling headphones', 60.0000),

(207, 'Webcam', 'HD webcam', 35.7599),

(208, 'USB Drive', '32GB USB 3.0', 9.9999),

(209, 'Charger', 'Laptop charger 65W', 25.4999),

(210, 'Speaker', 'Bluetooth portable speaker', 40.0000);

insert into Orders (OrderID, CustomerID, OrderDate, TotalAmount) values

(101, 1, '2025-06-01', 850.25),

(102, 2, '2025-06-02', 45.25),

(103, 1, '2025-06-03', 299.99),

(104, 4, '2025-06-03', 180.00),

(105, 5, '2025-06-05', 120.00),

(106, 3, '2025-06-06', 105.75),

(107, 7, '2025-06-06', 60.00),

(108, 4, '2025-06-06', 149.75),

(109, 9, '2025-06-09', 59.99),

(110, 10, '2025-06-13', 25.49);

insert into OrderDetails (OrderDetailID, OrderID, ProductID, Quantity) values

(301, 101, 201, 1),

(302, 101, 202, 2),

(303, 102, 203, 2),

(304, 103, 205, 5),

(305, 104, 204, 1),

(306, 105, 204, 4),

(307, 106, 206, 1),

(308, 107, 206, 3),

(309, 108, 201, 3),

(310, 108, 209, 7),

(1011, 109, 210, 1),

(1012, 110, 208, 6)

insert into Inventory(InventoryID, ProductID, QuantityInStock, LastStockUpdate) values

(401, 201, 20, '2025-05-27'),

(402, 202, 100, '2025-06-01'),

(403, 203, 50, '2025-06-02'),

(404, 204, 30, '2025-06-03'),

(405, 205, 25, '2025-06-03'),

(406, 206, 40, '2025-06-04'),

(407, 207, 35, '2025-06-05'),

(408, 208, 80, '2025-06-08'),

(409, 209, 60, '2025-06-010'),

(410, 210, 45, '2025-06-013')

**Tasks 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 o.OrderID,o.OrderDate,c.FirstName+' '+c.LastName from Orders o inner join Customers c on c.CustomerID = o.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(1, 'Muthu', 'Kumar', 'kumar@gmail.com', '7653688769', '13-C,NGK street,chennai')

**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 Description = 'Electronics'

**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.**

delete from Orders where OrderID = 110

delete from OrderDetails where OrderID = 110

**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(OrderID,CustomerID,OrderDate,TotalAmount)values(110,5,'2025-06-13',40.0000)

**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.**

update Customers set Email = 'saranraj@gmail.com',Address = '45-A East Road,Tirunelveli' where CustomerID = 2

**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 inner 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.**

delete from Orders where CustomerID = 9

delete from OrderDetails where CustomerID= 9

**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,Description,Price)values(211,'Smartwatch','Gadget',30.0000)

**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**

update Orders set Status = 'Shipped' where OrderID = 109;

**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 OrderCount = (select COUNT(\*) from Orders where Orders.CustomerID = Customers.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.**

select o.OrderID,o.OrderDate,o.TotalAmount,c.CustomerID,c.FirstName + ' ' + c.LastName AS CustomerName, c.Email, c.Phone, c.Address from Orders o inner join Customers c on o.CustomerID = c.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.**

select p.ProductName , sum(p.Price\*o.Quantity) as TotalRevenue from Products p inner join OrderDetails o on p.ProductID = o.ProductID group by p.ProductName

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

select distinct c.CustomerID, c.FirstName+' '+c.LastName as Name, c.Email, c.Phone, c.Address from Customers c inner join Orders o on c.CustomerID = o.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.**

select top 1 p.ProductName,sum(o.Quantity) as TotalQuantityOrdered from Products p inner join OrderDetails o on p.ProductID = o.ProductID group by ProductName order by sum(o.Quantity) desc

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

**Categories**

Select ProductName, Category from Product

**6. Write an SQL query to calculate the average order value for each customer. Include the**

**customer's name and their average order value**

select c.FirstName +' '+ c.LastName as Name , avg(TotalAmount) as Average from Customers c inner join Orders o on c.CustomerID = o.CustomerID group by c.CustomerID,c.FirstName,c.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.**

select top 1 o.OrderID,c.CustomerID, c.FirstName+' '+c.LastName as Name, c.Email, c.Phone, c.Address,o.TotalAmount from Customers c inner join Orders o on o.CustomerID = c.CustomerID order by o.TotalAmount desc

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

select p.ProductName,count(o.ProductID) as TotalTimes from Products p inner join OrderDetails o on p.ProductID = o.ProductID group by p.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.**

select FirstName+' '+LastName as CustomerName,ProductName from Customers c inner join Orders o on c.CustomerID=o.CustomerID inner join OrderDetails o1 on o.OrderID=o1.OrderID inner join Products p on p.ProductID=o1.ProductID where ProductName = 'Laptop'

**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**

select sum(TotalAmount) as TotalRevenue from Orders where OrderDate between '2025-06-01' and '2025-06-05'

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

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

select FirstName+' '+LastName as CustomerName from Customers c left join Orders o on c.CustomerID=o.CustomerID where o.OrderID is NULL

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

Select count(distinct ProductID) as NumberOfProducts from Inventory;

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

select sum(o.Quantity \* p.Price) as TotalRevenue from OrderDetails o join Products p on o.ProductID = p.ProductID

**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.**

select Category,avg(od.Quantity) as AverageQuantityOrdered from Products p inner join OrderDetails od ON p.ProductID = od.ProductID where p.Category = 'Gadgets' group by p.Category

**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.**

select c.FirstName+' '+c.LastName as CustomerName,sum(o.TotalAmount) from Customers c inner join Orders o on c.CustomerID = o.CustomerID where c.CustomerID = 1 group by c.FirstName,c.LastName

**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.**

select c.CustomerID,c.FirstName+ ' '+ c.LastName as CustomerName,count(o.OrderID) as NumberOfOrders from Customers c inner join Orders o on c.CustomerID = o.CustomerID group by c.CustomerID, c.FirstName, c.LastName having count(o.OrderID) = (select max(OrderCount)from (select count(OrderID) as OrderCount from Orders group by CustomerID) as OrderCounts)

**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**

select p.Category, sum(od.Quantity) as TotalQuantityOrdered from OrderDetails od inner join Products p on od.ProductID = p.ProductID group by p.Category having sum(od.Quantity) = (select max(totalqty) from (select sum(Quantity) as totalqty from OrderDetails od2 inner join Products p2 on od2.ProductID = p2.ProductID group by p2.Category) as CategoryTotals)

**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**

select top 1 c.Firstname + ' ' + c.Lastname as Customername, sum(od.Quantity \* p.Price) as TotalSpending from Customers c inner join orders o on c.CustomerID = o.CustomerID join OrderDetails od on o.OrderID = od.OrderID inner join Products p on od.ProductID = p.ProductID where p.Category = 'Electronic' group by c.CustomerID, c.Firstname, c.Lastname order by TotalSpending desc

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

select c.CustomerID,c.FirstName + ' ' + c.LastName as CustomerName, count(o.OrderID) as NumberOfOrders,sum(o.TotalAmount) as TotalRevenue, round(sum(o.TotalAmount) \* 1.0 / count(o.OrderID), 2) as AverageOrderValue from Customers c inner join Orders o on c.CustomerID = o.CustomerID group by c.CustomerID, c.FirstName, c.LastName

**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.**

select c.CustomerID, c.FirstName + ' ' + c.LastName as CustomerName, count(o.OrderID) as TotalOrders from Customers c left join Orders o on c.CustomerID = o.CustomerID group by c.CustomerID, c.FirstName, c.LastName;