TASK-1

Date: 27/11/23

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:

Database Tables:

1. Customers:
- CustomerID (Primary Key)
- FirstName
- LastName
- Email
- Phone
- Address

2. Products:
- ProductID (Primary Key)
- ProductName
- Description
- Price

3. Orders:
- OrderID (Primary Key)
- CustomerID (Foreign Key referencing Customers)
- OrderDate
- TotalAmount

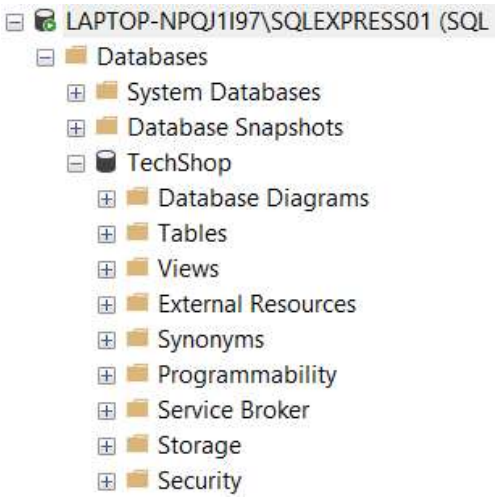
4. OrderDetails:
- OrderDetailID (Primary Key)
- OrderID (Foreign Key referencing Orders)
- ProductID (Foreign Key referencing Products)
- Quantity

5. Inventory
- InventoryID (Primary Key)
- ProductID (Foreign Key referencing Products)
- QuantityInStock
- LastStockUpdate

Questions:

Database Design (Normalization):

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:

Customers Table:
Customers

| CustomerID (PK) | FirstName | LastName | Email | Phone | Address |

Products Table:

Products

| ProductID (PK) | ProductName | Description | Price |

Orders Table:

Orders

| OrderID (PK) | CustomerID (FK) | OrderDate | TotalAmount |

OrderDetails Table:

OrderDetails

| OrderDetailID (PK) | OrderID (FK) | ProductID (FK) | Quantity |

Inventory Table:

Inventory

| InventoryID (PK) | ProductID (FK) | QuantityInStock | LastStockUpdate |
---

3. Perform the first three normal forms (1NF, 2NF, 3NF) analysis on the above tables.
Ans:

1NF
A relation R is said to be in 1NF, iff
- Repeated groups are not allowed in 1NF. Multivalued and composite attribute are not allowed in 1NF

2NF
A relation is said to be in 2NF iff,
- It is in 1NF
- No partial FDs are present in it

3NF
A relation is said to be in 3NF iff,
- It is in 2NF
- No Transitive FDs are present in it.
- A non-prime attribute must not be transitively dependent on the key attribute.

Customers Table:
1NF, 2NF, 3NF: Satisfied (No repeating groups, no partial dependencies, and no transitive dependencies)
Products Table:
1NF, 2NF, 3NF: Satisfied (No repeating groups, no partial dependencies, and no transitive dependencies)
Orders Table:
1NF, 2NF, 3NF: Satisfied (No repeating groups, no partial dependencies, and no transitive dependencies)
OrderDetails Table:
1NF, 2NF, 3NF: Satisfied (No repeating groups, composite primary key, and no transitive dependencies)

Inventory Table:
1NF, 2NF, 3NF: Satisfied (No repeating groups, composite primary key, and no transitive dependencies)

EXAMPLE:

CustID	FName	LName	Email	Phone	Address
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
2	David	Will	david.w@email.com	444-555-6666	789 Pine St
4	Emily	Taylor	emily.t@email.com	888-333-4444	303 Maple Ave

In 1NF:
Break down the repeating groups:

CustID	FName	LName	Email	Phone	
1	John	Doe	john.doe@email.com	123-456-7890	
2	Jane	Smith	jane.smith@email.com	987-654-3210	
3	David	Will	david.w@email.com	444-555-6666	
4	Emily	Taylor	emily.t@email.com	888-333-4444	

Now, each row has a unique CustomerID.

In 2NF:
Identify the partial dependency:

The partial dependency is on the {CustomerID} for the Address attribute.

Addresses Table (New):

CustID	**Address**
1	123 Main St
2	456 Oak St
3	789 Pine St
4	303 Maple Ave

Now, the original data is in 2NF.

In 3NF:

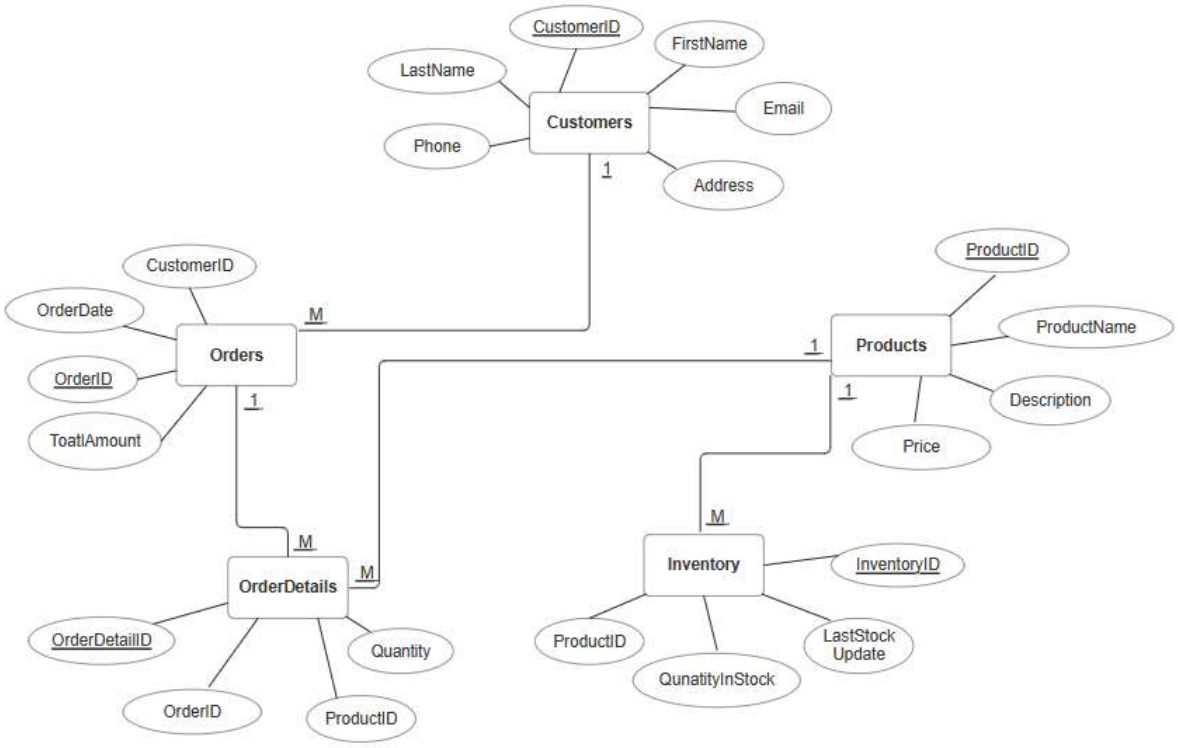
Identify transitive dependencies:

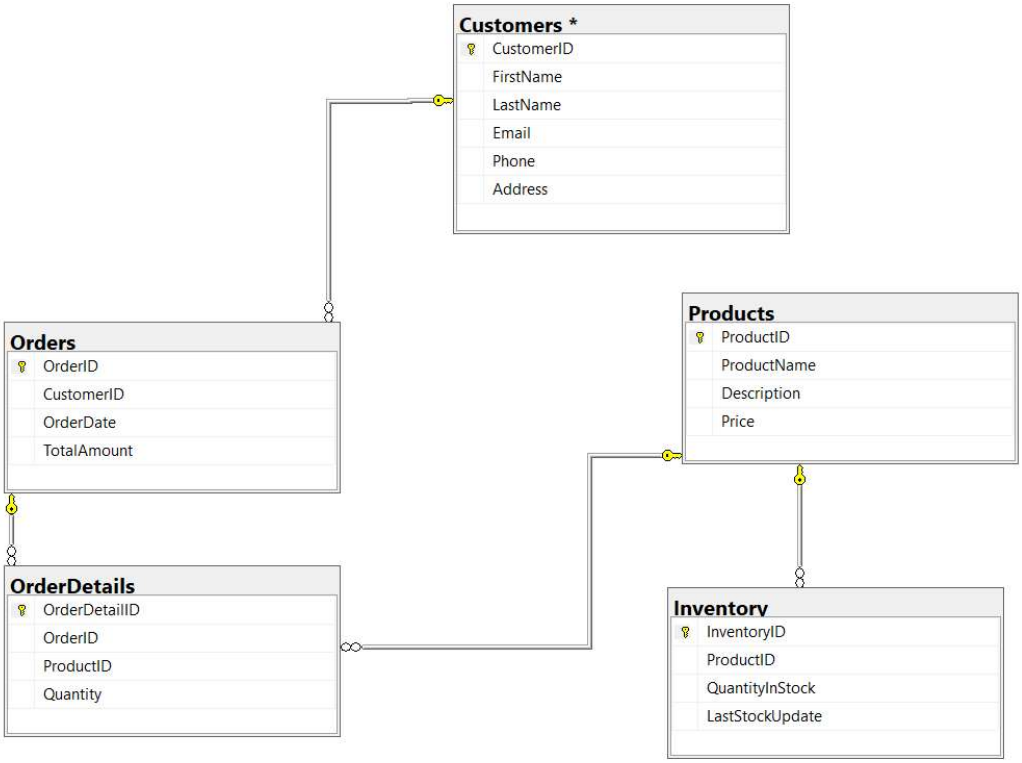
Phone is dependent on Address.

CustID	**Phone**
1	123-456-7890
2	987-654-3210
3	444-555-6666
4	888-333-4444

Now, the original data is in 3NF.

4. Create an ERD (Entity Relationship Diagram) for the database.
Ans:


CustomerID
FirstName
LastName
Email
Customers
Phone
Address
1
CustomerID
OrderDate
M
Orders
OrderID
1
ToatlAmount
ProductID
ProductName
1
Products
1
Description
Price
M
M
OrderDetails
M
Inventory
InventoryID
OrderDetailID
Quantity
ProductID
LastStock Update
QunatityInStock
OrderID
ProductID


Customers *
CustomerID
FirstName
LastName
Email
Phone
Address
Orders
OrderID
CustomerID
OrderDate
TotalAmount
Products
ProductID
ProductName
Description
Price
OrderDetails
OrderDetailID
OrderID
ProductID
Quantity
Inventory
InventoryID
ProductID
QuantityInStock
LastStockUpdate

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

Customers Table:

CustomerID (Primary Key)
FirstName
LastName
Email
Phone
Address

Products Table:

ProductID (Primary Key)
ProductName
Description
Price

Orders Table:

OrderID (Primary Key)
CustomerID (Foreign Key referencing Customers)
OrderDate
TotalAmount
Foreign Key Constraint:FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

OrderDetails Table:

OrderDetailID (Primary Key)
OrderID (Foreign Key referencing Orders)
ProductID (Foreign Key referencing Products)
Quantity

Foreign Key Constraints:

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)
FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

Inventory Table:
InventoryID (Primary Key)
ProductID (Foreign Key referencing Products)
QuantityInStock
LastStockUpdate
Foreign Key Constraint: FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

Data Definition Language (DDL):

1. Write SQL scripts to create the mentioned tables with appropriate data types, constraints, and relationships.
Ans:

```sql
-- Customers Table
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Email VARCHAR(100),
    Phone VARCHAR(20),
    Address VARCHAR(255)
);

-- Products Table
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(100),
    Description TEXT,
    Price DECIMAL(10, 2)
);

-- Orders Table
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    TotalAmount DECIMAL(10, 2),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);

-- OrderDetails Table
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)
);

-- Inventory Table
CREATE TABLE Inventory (
    InventoryID INT PRIMARY KEY,
```

```
    ProductID INT,
    QuantityInStock INT,
    LastStockUpdate DATETIME,
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
```

Data Manipulation Language (DML):

a. Insert at least 10 sample records into each of the following tables.
• Customers
• Products
• Orders
• OrderDetails
• Inventory
Ans:

```
-- Sample records for Customers table
INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone, Address)
VALUES
    (1, 'Amit', 'Sharma', 'amit.sharma@email.com', '9876543210', '123 Main St, Mumbai'),
    (2, 'Priya', 'Patel', 'priya.patel@email.com', '9876543211', '456 Oak St, Ahmedabad'),
    (3, 'Rahul', 'Singh', 'rahul.singh@email.com', '9876543212', '789 Pine St, Delhi'),
    (4, 'Sneha', 'Gupta', 'sneha.gupta@email.com', '9876543213', '101 Willow St,
Kolkata'),
    (5, 'Vikram', 'Verma', 'vikram.verma@email.com', '9876543214', '202 Cedar St,
Bangalore'),
    (6, 'Neha', 'Kumar', 'neha.kumar@email.com', '9876543215', '303 Maple St,
Chennai'),
    (7, 'Ankit', 'Malhotra', 'ankit.malhotra@email.com', '9876543216', '404 Birch St,
Hyderabad'),
    (8, 'Ritu', 'Saxena', 'ritu.saxena@email.com', '9876543217', '505 Elm St, Pune'),
    (9, 'Ravi', 'Choudhary', 'ravi.choudhary@email.com', '9876543218', '606 Pine St,
Jaipur'),
    (10, 'Pooja', 'Shukla', 'pooja.shukla@email.com', '9876543219', '707 Cedar St,
Lucknow');
```

```sql
-- Sample records for Products table
INSERT INTO Products (ProductID, ProductName, Description, Price)
VALUES
    (1, 'Smartphone X', 'High-performance smartphone', 799.99),
    (2, 'Laptop Pro', 'Powerful laptop for professionals', 1299.99),
    (3, 'Wireless Earbuds', 'Premium wireless earbuds', 149.99),
    (4, 'Tablet Z', 'Portable and lightweight tablet', 499.99),
    (5, 'Smartwatch 3', 'Water-resistant smartwatch', 199.99),
    (6, 'Bluetooth Speaker', 'High-quality portable speaker', 79.99),
    (7, 'Gaming Console', 'Next-gen gaming console', 399.99),
    (8, 'Camera Kit', 'Professional photography kit', 899.99),
    (9, 'Fitness Tracker', 'Track your fitness activities', 59.99),
    (10, 'Drones Unlimited', 'Explore the skies with our drones', 599.99);

-- Sample records for Orders table
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)
VALUES
    (1, 3, '2023-11-28', 149.99),
    (2, 7, '2023-11-28', 799.99),
    (3, 1, '2023-11-27', 1299.99),
    (4, 5, '2023-11-27', 499.99),
    (5, 2, '2023-11-26', 199.99),
    (6, 8, '2023-11-26', 79.99),
    (7, 4, '2023-11-25', 399.99),
    (8, 6, '2023-11-25', 899.99),
    (9, 9, '2023-11-24', 59.99),
    (10, 10, '2023-11-24', 599.99);

-- Sample records for OrderDetails table
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity)
VALUES
    (1, 1, 3, 2),
    (2, 2, 1, 1),
    (3, 3, 2, 1),
    (4, 4, 4, 3),
    (5, 5, 5, 2),
    (6, 6, 6, 1),
    (7, 7, 7, 1),
    (8, 8, 8, 1),
    (9, 9, 9, 5),
```

```
    (10, 10, 10, 1);

-- Sample records for Inventory table
INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate)
VALUES
    (1, 1, 50, '2023-11-28'),
    (2, 2, 30, '2023-11-28'),
    (3, 3, 100, '2023-11-28'),
    (4, 4, 20, '2023-11-28'),
    (5, 5, 40, '2023-11-28'),
    (6, 6, 50, '2023-11-28'),
    (7, 7, 15, '2023-11-28'),
    (8, 8, 10, '2023-11-28'),
    (9, 9, 80, '2023-11-28'),
    (10, 10, 25, '2023-11-28');
```

b. Write SQL queries for the following tasks:

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

Ans: SELECT FirstName, LastName, Email FROM Customers;

```
SELECT FirstName, LastName, Email FROM Customers;
```

100 %

Results | Messages

	FirstName	LastName	Email
1	Amit	Sharma	amit.sharma@email.com
2	Priya	Patel	priya.patel@email.com
3	Rahul	Singh	rahul.singh@email.com
4	Sneha	Gupta	sneha.gupta@email.com
5	Vikram	Verma	vikram.verma@email.com
6	Neha	Kumar	neha.kumar@email.com
7	Ankit	Malhotra	ankit.malhotra@email.com
8	Ritu	Saxena	ritu.saxena@email.com
9	Ravi	Choudhary	ravi.choudhary@email.com
10	Pooja	Shukla	pooja.shukla@email.com

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

Ans:

```sql
SELECT Orders.OrderID, Orders.OrderDate,
Customers.FirstName, Customers.LastName
FROM Orders
JOIN Customers ON
Orders.CustomerID = Customers.CustomerID;
```

100 %

Results | Messages

	OrderID	OrderDate	FirstName	LastName
1	1	2023-11-28	Rahul	Singh
2	2	2023-11-28	Ankit	Malhotra
3	3	2023-11-27	Amit	Sharma
4	4	2023-11-27	Vikram	Verma
5	5	2023-11-26	Priya	Patel
6	6	2023-11-26	Ritu	Saxena
7	7	2023-11-25	Sneha	Gupta
8	8	2023-11-25	Neha	Kumar
9	9	2023-11-24	Ravi	Choudhary
10	10	2023-11-24	Pooja	Shukla

Or

```sql
SELECT OrderID, OrderDate,
    (SELECT CONCAT(FirstName, ' ', LastName)
    FROM Customers WHERE
    Customers.CustomerID = Orders.CustomerID)
    AS CustomerName
FROM Orders;
```

100 %

Results | Messages

	OrderID	OrderDate	CustomerName
1	1	2023-11-28	Rahul Singh
2	2	2023-11-28	Ankit Malhotra
3	3	2023-11-27	Amit Sharma
4	4	2023-11-27	Vikram Verma
5	5	2023-11-26	Priya Patel
6	6	2023-11-26	Ritu Saxena
7	7	2023-11-25	Sneha Gupta
8	8	2023-11-25	Neha Kumar
9	9	2023-11-24	Ravi Choudhary
10	10	2023-11-24	Pooja Shukla

SELECT Orders.OrderID, Orders.OrderDate, Customers.FirstName, Customers.LastName
FROM Orders
JOIN Customers ON Orders.CustomerID = Customers.CustomerID;

SELECT OrderID, OrderDate,
(SELECT CONCAT(FirstName, ' ', LastName) FROM Customers WHERE Customers.CustomerID = Orders.CustomerID) AS CustomerName
FROM Orders;

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 (CustomerID, FirstName, LastName, Email, Phone, Address)
VALUES (11, 'Swanam', 'Krishna', 'swanam.krishna@email.com', '7736780630', '110 Thanniyam, Thrissur');
Select* from Customers;
```

100 %

Results | Messages

	CustomerID	FirstName	LastName	Email	Phone	Address
1	1	Amit	Sharma	amit.sharma@email.com	9876543210	123 Main St, Mumbai
2	2	Priya	Patel	priya.patel@email.com	9876543211	456 Oak St, Ahmedabad
3	3	Rahul	Singh	rahul.singh@email.com	9876543212	789 Pine St, Delhi
4	4	Sneha	Gupta	sneha.gupta@email.com	9876543213	101 Willow St, Kolkata
5	5	Vikram	Verma	vikram.verma@email.com	9876543214	202 Cedar St, Bangalore
6	6	Neha	Kumar	neha.kumar@email.com	9876543215	303 Maple St, Chennai
7	7	Ankit	Malhotra	ankit.malhotra@email.com	9876543216	404 Birch St, Hyderabad
8	8	Ritu	Saxena	ritu.saxena@email.com	9876543217	505 Elm St, Pune
9	9	Ravi	Choudhary	ravi.choudhary@email.com	9876543218	606 Pine St, Jaipur
10	10	Pooja	Shukla	pooja.shukla@email.com	9876543219	707 Cedar St, Lucknow
11	11	Swanam	Krishna	swanam.krishna@email.com	7736780630	110 Thanniyam, Thrissur

INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone, Address)
VALUES (11, 'Swanam', 'Krishna', 'swanam.krishna@email.com', '7736780630', '110 Thanniyam, Thrissur');
Select* from Customers;

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

Ans:

select * from Products;
UPDATE Products
SET Price = Price * 1.10;
select * from Products;

```sql
select * from Products;
UPDATE Products
SET Price = Price * 1.10;

select * from Products;
```

100 %

Results | Messages

	ProductID	ProductName	Description	Price
1	1	Smartphone X	High-performance smartphone	879.99
2	2	Laptop Pro	Powerful laptop for professionals	1429.99
3	3	Wireless Earbuds	Premium wireless earbuds	164.99
4	4	Tablet Z	Portable and lightweight tablet	549.99
5	5	Smartwatch 3	Water-resistant smartwatch	219.99
6	6	Bluetooth Speaker	High-quality portable speaker	87.99
7	7	Gaming Console	Next-gen gaming console	439.99
8	8	Camera Kit	Professional photography kit	989.99
9	9	Fitness Tracker	Track your fitness activities	65.99
10	10	Drones Unlimited	Explore the skies with our drones	659.99

	ProductID	ProductName	Description	Price
1	1	Smartphone X	High-performance smartphone	967.99
2	2	Laptop Pro	Powerful laptop for professionals	1572.99
3	3	Wireless Earbuds	Premium wireless earbuds	181.49
4	4	Tablet Z	Portable and lightweight tablet	604.99
5	5	Smartwatch 3	Water-resistant smartwatch	241.99
6	6	Bluetooth Spea...	High-quality portable speaker	96.79
7	7	Gaming Console	Next-gen gaming console	483.99
8	8	Camera Kit	Professional photography kit	1088.99
9	9	Fitness Tracker	Track your fitness activities	72.59
10	10	Drones Unlimited	Explore the skies with our dron...	725.99

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: DECLARE @OrderIDToDelete INT;
SET @OrderIDToDelete =6;

-- Delete from OrderDetails table
DELETE FROM OrderDetails
WHERE OrderID = @OrderIDToDelete;

-- Delete from Orders table
DELETE FROM Orders
WHERE OrderID = @OrderIDToDelete;

select * from Orders;

```sql
DECLARE @OrderIDToDelete INT;
SET @OrderIDToDelete =6; /* User input: specify the order ID to delete */

-- Delete from OrderDetails table
DELETE FROM OrderDetails
WHERE OrderID = @OrderIDToDelete;

-- Delete from Orders table
DELETE FROM Orders
WHERE OrderID = @OrderIDToDelete;

select * from Orders;
```

100 %

Results | Messages

	OrderID	CustomerID	OrderDate	TotalAmount
1	1	3	2023-11-28	149.99
2	2	7	2023-11-28	799.99
3	3	1	2023-11-27	1299.99
4	4	5	2023-11-27	499.99
5	5	2	2023-11-26	199.99
6	7	4	2023-11-25	399.99
7	8	6	2023-11-25	899.99
8	9	9	2023-11-24	59.99
9	10	10	2023-11-24	599.99

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:

```sql
select * from Orders;
DECLARE @OrderID INT;
DECLARE @CustomerID INT;
DECLARE @OrderDate DATE;
DECLARE @TotalAmount DECIMAL(10, 2);

-- Set values for the new order
SET @OrderID = 6;
SET @CustomerID = 5;
SET @OrderDate = GETDATE();
SET @TotalAmount = 280.99;

-- Insert into the Orders table
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)
VALUES (@OrderID, @CustomerID, @OrderDate, @TotalAmount);

select * from Orders;
```

```sql
select * from Orders;
DECLARE @OrderID INT;
DECLARE @CustomerID INT;
DECLARE @OrderDate DATE;
DECLARE @TotalAmount DECIMAL(10, 2);

-- Set values for the new order
SET @OrderID = 6/* User input: specify the order ID */;
SET @CustomerID = 5/* User input: specify the customer ID */;
SET @OrderDate = GETDATE();  -- You can replace this with the actual order date
SET @TotalAmount = 280.99/* User input: specify the total amount */;

-- Insert into the Orders table
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)
VALUES (@OrderID, @CustomerID, @OrderDate, @TotalAmount);

select * from Orders;
```

80 %

Results | Messages

	OrderID	CustomerID	OrderDate	TotalAmount
1	1	3	2023-11-28	149.99
2	2	7	2023-11-28	799.99
3	3	1	2023-11-27	1299.99
4	4	5	2023-11-27	499.99
5	5	2	2023-11-26	199.99
6	7	4	2023-11-25	399.99
7	8	6	2023-11-25	899.99
8	9	9	2023-11-24	59.99
9	10	10	2023-11-24	599.99

	OrderID	CustomerID	OrderDate	TotalAmount
1	1	3	2023-11-28	149.99
2	2	7	2023-11-28	799.99
3	3	1	2023-11-27	1299.99
4	4	5	2023-11-27	499.99
5	5	2	2023-11-26	199.99
6	6	5	2023-11-28	280.99
7	7	4	2023-11-25	399.99
8	8	6	2023-11-25	899.99
9	9	9	2023-11-24	59.99
10	10	10	2023-11-24	599.99

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: select * from Customers;

```sql
DECLARE @CustomerIDToUpdate INT;
DECLARE @NewEmail NVARCHAR(255);
DECLARE @NewAddress NVARCHAR(255);

-- User input: specify the customer ID, new email, and new address
SET @CustomerIDToUpdate = 11;
SET @NewEmail = 'swanam.k@email.com';
SET @NewAddress = '110 Thrissur, Kerala';

-- Update the contact information in the Customers table
UPDATE Customers
SET Email = @NewEmail, Address = @NewAddress
WHERE CustomerID = @CustomerIDToUpdate;
```

```sql
select * from Customers;
DECLARE @CustomerIDToUpdate INT;
DECLARE @NewEmail NVARCHAR(255);
DECLARE @NewAddress NVARCHAR(255);

-- User input: specify the customer ID, new email, and new address
SET @CustomerIDToUpdate = 11/* User input: specify the customer ID */;
SET @NewEmail = 'swanam.k@email.com'/* User input: specify the new email */;
SET @NewAddress = '110 Thrissur, Kerala' /* User input: specify the new address */;

-- Update the contact information in the Customers table
UPDATE Customers
SET Email = @NewEmail, Address = @NewAddress
WHERE CustomerID = @CustomerIDToUpdate;
```

80 %

Results | Messages

	CustomerID	FirstName	LastName	Email	Phone	Address
1	1	Amit	Sharma	amit.sharma@email.com	9876543210	123 Main St, Mumbai
2	2	Priya	Patel	priya.patel@email.com	9876543211	456 Oak St, Ahmedabad
3	3	Rahul	Singh	rahul.singh@email.com	9876543212	789 Pine St, Delhi
4	4	Sneha	Gupta	sneha.gupta@email.com	9876543213	101 Willow St, Kolkata
5	5	Vikram	Verma	vikram.verma@email.com	9876543214	202 Cedar St, Bangalore
6	6	Neha	Kumar	neha.kumar@email.com	9876543215	303 Maple St, Chennai
7	7	Ankit	Malhotra	ankit.malhotra@email.com	9876543216	404 Birch St, Hyderabad
8	8	Ritu	Saxena	ritu.saxena@email.com	9876543217	505 Elm St, Pune
9	9	Ravi	Choudhary	ravi.choudhary@email.com	9876543218	606 Pine St, Jaipur
10	10	Pooja	Shukla	pooja.shukla@email.com	9876543219	707 Cedar St, Lucknow
11	11	Swanam	Krishna	swanam.krishna@email.com	7736780630	110 Thanniyam, Thrissur

	CustomerID	FirstName	LastName	Email	Phone	Address
1	1	Amit	Sharma	amit.sharma@email.com	9876543210	123 Main St, Mumbai
2	2	Priya	Patel	priya.patel@email.com	9876543211	456 Oak St, Ahmedabad
3	3	Rahul	Singh	rahul.singh@email.com	9876543212	789 Pine St, Delhi
4	4	Sneha	Gupta	sneha.gupta@email.com	9876543213	101 Willow St, Kolkata
5	5	Vikram	Verma	vikram.verma@email.com	9876543214	202 Cedar St, Bangalore
6	6	Neha	Kumar	neha.kumar@email.com	9876543215	303 Maple St, Chennai
7	7	Ankit	Malhotra	ankit.malhotra@email.com	9876543216	404 Birch St, Hyderabad
8	8	Ritu	Saxena	ritu.saxena@email.com	9876543217	505 Elm St, Pune
9	9	Ravi	Choudh...	ravi.choudhary@email.com	9876543218	606 Pine St, Jaipur
10	10	Pooja	Shukla	pooja.shukla@email.com	9876543219	707 Cedar St, Lucknow
11	11	Swanam	Krishna	swanam.k@email.com	7736780630	110 Thrissur, Kerala

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:

```sql
UPDATE Orders
SET TotalAmount = (
   SELECT SUM(Quantity * (SELECT Price FROM Products WHERE ProductID =
OrderDetails.ProductID))
   FROM OrderDetails
   WHERE OrderDetails.OrderID = Orders.OrderID
);
SELECT * FROM Orders;
```

Or

```
select * from Orders;
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
);
SELECT * FROM Orders;
```

```
select * from Orders;
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
);
SELECT * FROM Orders;
```

100 %

Results | Messages

	OrderID	CustomerID	OrderDate	TotalAmount
1	1	3	2023-11-28	149.99
2	2	7	2023-11-28	799.99
3	3	1	2023-11-27	1299.99
4	4	5	2023-11-27	499.99
5	5	2	2023-11-26	199.99
6	6	5	2023-11-28	280.99
7	7	4	2023-11-25	399.99
8	8	6	2023-11-25	899.99
9	9	9	2023-11-24	59.99
10	10	10	2023-11-24	599.99

	OrderID	CustomerID	OrderDate	TotalAmount
1	1	3	2023-11-28	362.98
2	2	7	2023-11-28	967.99
3	3	1	2023-11-27	1572.99
4	4	5	2023-11-27	1814.97
5	5	2	2023-11-26	483.98
6	6	5	2023-11-28	NULL
7	7	4	2023-11-25	483.99
8	8	6	2023-11-25	1088.99
9	9	9	2023-11-24	362.95
10	10	10	2023-11-24	725.99

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:

```
select *  from Orders, OrderDetails;
DECLARE @CustomerIDToDelete INT;
SET @CustomerIDToDelete = 5;
```

-- Delete from OrderDetails table
DELETE FROM OrderDetails
WHERE OrderID IN (SELECT OrderID FROM Orders WHERE CustomerID = @CustomerIDToDelete);
-- Delete from Orders table
DELETE FROM Orders
WHERE CustomerID = @CustomerIDToDelete;
select * from Orders, OrderDetails;

```
select *  from Orders, OrderDetails;
DECLARE @CustomerIDToDelete INT;
-- User input: specify the customer ID
SET @CustomerIDToDelete = 5/* User input: specify the customer ID */;
-- Delete from OrderDetails table
DELETE FROM OrderDetails
WHERE OrderID IN (SELECT OrderID FROM Orders WHERE CustomerID = @CustomerIDToDelete);
-- Delete from Orders table
DELETE FROM Orders
WHERE CustomerID = @CustomerIDToDelete;
select *  from Orders, OrderDetails;
```

100 %

Results | Messages

	OrderID	CustomerID	OrderDate	TotalAmount	OrderDetailID	OrderID	ProductID	Quantity
1	1	3	2023-11-28	362.98	1	1	3	2
2	2	7	2023-11-28	967.99	1	1	3	2
3	3	1	2023-11-27	1572.99	1	1	3	2
4	4	5	2023-11-27	1814.97	1	1	3	2
5	5	2	2023-11-26	483.98	1	1	3	2
6	6	5	2023-11-28	NULL	1	1	3	2
7	7	4	2023-11-25	483.99	1	1	3	2
8	8	6	2023-11-25	1088.99	1	1	3	2
9	9	9	2023-11-24	362.95	1	1	3	2
10	10	10	2023-11-24	725.99	1	1	3	2

	OrderID	CustomerID	OrderDate	TotalAmount	OrderDetailID	OrderID	ProductID	Quantity
1	1	3	2023-11-28	362.98	1	1	3	2
2	2	7	2023-11-28	967.99	1	1	3	2
3	3	1	2023-11-27	1572.99	1	1	3	2
4	5	2	2023-11-26	483.98	1	1	3	2
5	7	4	2023-11-25	483.99	1	1	3	2
6	8	6	2023-11-25	1088.99	1	1	3	2
7	9	9	2023-11-24	362.95	1	1	3	2
8	10	10	2023-11-24	725.99	1	1	3	2
9	1	3	2023-11-28	362.98	2	2	1	1
10	2	7	2023-11-28	967.99	2	2	1	1

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:

select * from Products;
-- Add the Category column to the Products table
ALTER TABLE Products
ADD Category NVARCHAR(255);

-- Insert a new electronic gadget with the Category information
INSERT INTO Products (ProductID, ProductName, Description, Price, Category)
VALUES (11, 'New Gadget', 'Cutting-edge features and technology', 699.99,
'Electronic Gadget');
select * from Products;

```sql
select *  from Products;
-- Add the Category column to the Products table
ALTER TABLE Products
ADD Category NVARCHAR(255);
-- Insert a new electronic gadget with the Category information
INSERT INTO Products (ProductID, ProductName, Description, Price, Category)
VALUES (11, 'New Gadget', 'Cutting-edge features and technology', 699.99,
'Electronic Gadget');
select *  from Products;
```

100 %

Results | Messages

	ProductID	ProductName	Description	Price	Category
1	1	Smartphone X	High-performance smartphone	967.99	NULL
2	2	Laptop Pro	Powerful laptop for professionals	1572.99	NULL
3	3	Wireless Earbuds	Premium wireless earbuds	181.49	NULL
4	4	Tablet Z	Portable and lightweight tablet	604.99	NULL
5	5	Smartwatch 3	Water-resistant smartwatch	241.99	NULL
6	6	Bluetooth Speaker	High-quality portable speaker	96.79	NULL
7	7	Gaming Console	Next-gen gaming console	483.99	NULL
8	8	Camera Kit	Professional photography kit	1088.99	NULL
9	9	Fitness Tracker	Track your fitness activities	72.59	NULL
10	10	Drones Unlimited	Explore the skies with our drones	725.99	NULL

	ProductID	ProductName	Description	Price	Category
1	1	Smartphone X	High-performance smartphone	967.99	NULL
2	2	Laptop Pro	Powerful laptop for professionals	1572.99	NULL
3	3	Wireless Earbuds	Premium wireless earbuds	181.49	NULL
4	4	Tablet Z	Portable and lightweight tablet	604.99	NULL
5	5	Smartwatch 3	Water-resistant smartwatch	241.99	NULL
6	6	Bluetooth Speaker	High-quality portable speaker	96.79	NULL
7	7	Gaming Console	Next-gen gaming console	483.99	NULL
8	8	Camera Kit	Professional photography kit	1088.99	NULL
9	9	Fitness Tracker	Track your fitness activities	72.59	NULL
10	10	Drones Unlimited	Explore the skies with our dron...	725.99	NULL
11	11	New Gadget	Cutting-edge features and tech...	699.99	Electro...

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:

```sql
select * from Orders;
-- Add the Status column to the Orders table with a default value of 'Pending'
ALTER TABLE Orders
ADD Status NVARCHAR(50) DEFAULT 'Pending';

-- UPDATE Orders SET Status = 'Pending';

DECLARE @OrderIDToUpdate INT;
DECLARE @NewStatus NVARCHAR(50);

-- User input: specify the order ID and the new status

SET @OrderIDToUpdate = 5
SET @NewStatus = 'Shipped'

-- Update the status in the Orders table
```

UPDATE Orders
SET Status = @NewStatus
WHERE OrderID = @OrderIDToUpdate;

select * from Orders;

```
select * from Orders;
DECLARE @OrderIDToUpdate INT;
DECLARE @NewStatus NVARCHAR(50);
-- User input: specify the order ID and the new status
SET @OrderIDToUpdate = 5/* User input: specify the order ID */;
SET @NewStatus = 'Shipped'/* User input: specify the new status */;
-- Update the status in the Orders table
UPDATE Orders
SET Status = @NewStatus
WHERE OrderID = @OrderIDToUpdate;

select * from Orders;
```

100 %

Results | Messages

	OrderID	CustomerID	OrderDate	TotalAmount	Status
1	1	3	2023-11-28	362.98	NULL
2	2	7	2023-11-28	967.99	NULL
3	3	1	2023-11-27	1572.99	NULL
4	5	2	2023-11-26	483.98	NULL
5	7	4	2023-11-25	483.99	NULL
6	8	6	2023-11-25	1088.99	NULL
7	9	9	2023-11-24	362.95	NULL
8	10	10	2023-11-24	725.99	NULL

	OrderID	CustomerID	OrderDate	TotalAmount	Status
1	1	3	2023-11-28	362.98	NULL
2	2	7	2023-11-28	967.99	NULL
3	3	1	2023-11-27	1572.99	NULL
4	5	2	2023-11-26	483.98	Shipped
5	7	4	2023-11-25	483.99	NULL
6	8	6	2023-11-25	1088.99	NULL
7	9	9	2023-11-24	362.95	NULL
8	10	10	2023-11-24	725.99	NULL

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:

select * from Customers;
/*ALTER TABLE Customers
ADD OrdersCount INT;*/
UPDATE Customers
SET OrdersCount = (
  SELECT COUNT(*)
  FROM Orders
  WHERE Orders.CustomerID = Customers.CustomerID
)
select * from Customers;

```sql
UPDATE Customers
SET OrdersCount = (
    SELECT COUNT(*)
    FROM Orders
    WHERE Orders.CustomerID = Customers.CustomerID
)
select * from Customers;
```

100 %

Results | Messages

	CustomerID	FirstName	LastName	Email	Phone	Address	OrdersCount
1	1	Amit	Sharma	amit.sharma@email.com	9876543210	123 Main St, Mumbai	NULL
2	2	Priya	Patel	priya.patel@email.com	9876543211	456 Oak St, Ahmedabad	NULL
3	3	Rahul	Singh	rahul.singh@email.com	9876543212	789 Pine St, Delhi	NULL
4	4	Sneha	Gupta	sneha.gupta@email.com	9876543213	101 Willow St, Kolkata	NULL
5	5	Vikram	Verma	vikram.verma@email.com	9876543214	202 Cedar St, Bangalore	NULL
6	6	Neha	Kumar	neha.kumar@email.com	9876543215	303 Maple St, Chennai	NULL
7	7	Ankit	Malhotra	ankit.malhotra@email.com	9876543216	404 Birch St, Hyderabad	NULL
8	8	Ritu	Saxena	ritu.saxena@email.com	9876543217	505 Elm St, Pune	NULL
9	9	Ravi	Choudhary	ravi.choudhary@email.com	9876543218	606 Pine St, Jaipur	NULL
10	10	Pooja	Shukla	pooja.shukla@email.com	9876543219	707 Cedar St, Lucknow	NULL
11	11	Swanam	Krishna	swanam.k@email.com	7736780630	110 Thrissur, Kerala	NULL

	CustomerID	FirstName	LastName	Email	Phone	Address	OrdersCount
1	1	Amit	Sharma	amit.sharma@email.com	9876543210	123 Main St, Mumbai	1
2	2	Priya	Patel	priya.patel@email.com	9876543211	456 Oak St, Ahmedabad	1
3	3	Rahul	Singh	rahul.singh@email.com	9876543212	789 Pine St, Delhi	1
4	4	Sneha	Gupta	sneha.gupta@email.com	9876543213	101 Willow St, Kolkata	1
5	5	Vikram	Verma	vikram.verma@email.com	9876543214	202 Cedar St, Bangalore	0
6	6	Neha	Kumar	neha.kumar@email.com	9876543215	303 Maple St, Chennai	1
7	7	Ankit	Malhotra	ankit.malhotra@email.com	9876543216	404 Birch St, Hyderabad	1
8	8	Ritu	Saxena	ritu.saxena@email.com	9876543217	505 Elm St, Pune	0
9	9	Ravi	Choudh...	ravi.choudhary@email.com	9876543218	606 Pine St, Jaipur	1
10	10	Pooja	Shukla	pooja.shukla@email.com	9876543219	707 Cedar St, Lucknow	1
11	11	Swanam	Krishna	swanam.k@email.com	7736780630	110 Thrissur, Kerala	0

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:

```sql
SELECT
   Orders.OrderID,
   Orders.OrderDate,
   Customers.CustomerID,
   CONCAT(Customers.FirstName, ' ', Customers.LastName) AS CustomerName
FROM
   Orders
JOIN
   Customers ON Orders.CustomerID = Customers.CustomerID;
```

```sql
--1
SELECT
    Orders.OrderID,
    Orders.OrderDate,
    Customers.CustomerID,
    CONCAT(Customers.FirstName, ' ', Customers.LastName) AS CustomerName
FROM
    Orders
JOIN
    Customers ON Orders.CustomerID = Customers.CustomerID;
```

100 %

Results | Messages

	OrderID	OrderDate	CustomerID	CustomerName
1	1	2023-11-28	3	Rahul Singh
2	2	2023-11-28	7	Ankit Malhotra
3	3	2023-11-27	1	Amit Sharma
4	5	2023-11-26	2	Priya Patel
5	7	2023-11-25	4	Sneha Gupta
6	8	2023-11-25	6	Neha Kumar
7	9	2023-11-24	9	Ravi Choudhary
8	10	2023-11-24	10	Pooja Shukla

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:

```sql
SELECT
   ProductName,
   SUM(Quantity * Price) AS TotalRevenue
FROM
   Products
JOIN
   OrderDetails ON Products.ProductID = OrderDetails.ProductID
JOIN
   Orders ON OrderDetails.OrderID = Orders.OrderID
GROUP BY
   ProductName;
```

```sql
SELECT
    ProductName,
    SUM(Quantity * Price) AS TotalRevenue
FROM
    Products
JOIN
    OrderDetails ON Products.ProductID = OrderDetails.ProductID
JOIN
    Orders ON OrderDetails.OrderID = Orders.OrderID
GROUP BY
    ProductName;
```

100 %

Results | Messages

	ProductName	TotalRevenue
1	Camera Kit	1088.99
2	Drones Unlimited	725.99
3	Fitness Tracker	362.95
4	Gaming Console	483.99
5	Laptop Pro	1572.99
6	Smartphone X	967.99
7	Smartwatch 3	483.98
8	Wireless Earbuds	362.98

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

Ans:

```sql
SELECT
   Customers.CustomerID,Customers.FirstName,Customers.LastName,Customers.Phone
FROM
   Customers
JOIN
   Orders ON Customers.CustomerID = Orders.CustomerID
GROUP BY
   Customers.CustomerID, Customers.FirstName, Customers.LastName, Customers.Phone;
```

```sql
--3
SELECT
    Customers.CustomerID,
    Customers.FirstName,
    Customers.LastName,
    Customers.Phone
FROM
    Customers
JOIN
    Orders ON Customers.CustomerID = Orders.CustomerID
GROUP BY
    Customers.CustomerID, Customers.FirstName,
    Customers.LastName, Customers.Phone;
```

100 %

Results | Messages

	CustomerID	FirstName	LastName	Phone
1	1	Amit	Sharma	9876543210
2	2	Priya	Patel	9876543211
3	3	Rahul	Singh	9876543212
4	4	Sneha	Gupta	9876543213
5	6	Neha	Kumar	9876543215
6	7	Ankit	Malhotra	9876543216
7	9	Ravi	Choudhary	9876543218
8	10	Pooja	Shukla	9876543219

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:

```sql
SELECT TOP 1 P.ProductName, SUM(OD.Quantity) AS TotalQuantityOrdered
FROM Products P
JOIN OrderDetails OD ON P.ProductID = OD.ProductID
JOIN Orders O ON OD.OrderID = O.OrderID
WHERE P.Category = 'Electronic Gadget'
GROUP BY P.ProductID, P.ProductName
ORDER BY TotalQuantityOrdered DESC;
```

```sql
SELECT TOP 1
    P.ProductName,
    SUM(OD.Quantity) AS TotalQuantityOrdered
FROM
    Products P
JOIN
    OrderDetails OD ON P.ProductID = OD.ProductID
JOIN
    Orders O ON OD.OrderID = O.OrderID
WHERE
    P.Category = 'Electronic Gadget'
GROUP BY
    P.ProductID, P.ProductName
ORDER BY
    TotalQuantityOrdered DESC;
```

100 %

Results | Messages

	ProductName	TotalQuantityOrdered
1	Fitness Tracker	5

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

Ans:

SELECT P.ProductName, P.Category
FROM Products P
WHERE P.Category = 'Electronic Gadget';

```sql
SELECT
    P.ProductName,
    P.Category
FROM
    Products P
WHERE
    P.Category = 'Electronic Gadget';
```

100 %

Results | Messages

	ProductName	Category
1	Smartphone X	Electronic Gadget
2	Laptop Pro	Electronic Gadget
3	Wireless Earbuds	Electronic Gadget
4	Tablet Z	Electronic Gadget
5	Smartwatch 3	Electronic Gadget
6	Bluetooth Speaker	Electronic Gadget
7	Gaming Console	Electronic Gadget
8	Camera Kit	Electronic Gadget
9	Fitness Tracker	Electronic Gadget
10	Drones Unlimited	Electronic Gadget
11	GPS	Electronic Gadget

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 C.CustomerID, C.FirstName, C.LastName, AVG(OD.Quantity * P.Price) AS AvgOrderValue
FROM Customers C
JOIN Orders O ON C.CustomerID = O.CustomerID
JOIN OrderDetails OD ON O.OrderID = OD.OrderID
JOIN Products P ON OD.ProductID = P.ProductID
GROUP BY C.CustomerID, C.FirstName, C.LastName;

```sql
--6
SELECT C.CustomerID, C.FirstName, C.LastName, AVG(OD.Quantity * P.Price) AS AvgOrderValue
FROM Customers C
JOIN Orders O ON C.CustomerID = O.CustomerID
JOIN OrderDetails OD ON O.OrderID = OD.OrderID
JOIN Products P ON OD.ProductID = P.ProductID
GROUP BY C.CustomerID, C.FirstName, C.LastName;
```

100 %

Results | Messages

	CustomerID	FirstName	LastName	AvgOrderValue
1	1	Amit	Sharma	1549.360000
2	2	Priya	Patel	0.000000
3	3	Rahul	Singh	357.840000
4	4	Sneha	Gupta	0.000000
5	6	Neha	Kumar	1072.730000
6	7	Ankit	Malhotra	953.450000
7	9	Ravi	Choudhary	358.000000
8	10	Pooja	Shukla	715.110000

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 TOP 1 O.OrderID, C.CustomerID, C.FirstName, C.LastName, SUM(OD.Quantity * P.Price) AS TotalRevenue
FROM Orders O
JOIN Customers C ON O.CustomerID = C.CustomerID
JOIN OrderDetails OD ON O.OrderID = OD.OrderID
JOIN Products P ON OD.ProductID = P.ProductID
GROUP BY O.OrderID, C.CustomerID, C.FirstName, C.LastName
ORDER BY TotalRevenue DESC;

```sql
--7
SELECT TOP 1 O.OrderID, C.CustomerID, C.FirstName, C.LastName, SUM(OD.Quantity * P.Price) AS TotalRevenue
FROM Orders O
JOIN Customers C ON O.CustomerID = C.CustomerID
JOIN OrderDetails OD ON O.OrderID = OD.OrderID
JOIN Products P ON OD.ProductID = P.ProductID
GROUP BY O.OrderID, C.CustomerID, C.FirstName, C.LastName
ORDER BY TotalRevenue DESC;
```

100 %

Results | Messages

	OrderID	CustomerID	FirstName	LastName	TotalRevenue
1	3	1	Amit	Sharma	1549.36

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

Ans:

SELECT P.ProductID, P.ProductName, COUNT(OD.OrderID) AS OrderCount
FROM Products P
JOIN OrderDetails OD ON P.ProductID = OD.ProductID
GROUP BY P.ProductID, P.ProductName;

```sql
SELECT P.ProductID, P.ProductName, COUNT(OD.OrderID) AS OrderCount
FROM Products P
JOIN OrderDetails OD ON P.ProductID = OD.ProductID
GROUP BY P.ProductID, P.ProductName;
```

100 %

Results | Messages

	ProductID	ProductName	OrderCount
1	1	Smartphone X	1
2	2	Laptop Pro	1
3	3	Wireless Earbuds	1
4	5	Smartwatch 3	1
5	7	Gaming Console	1
6	8	Camera Kit	1
7	9	Fitness Tracker	1
8	10	Drones Unlimited	1

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:

DECLARE @ProductName NVARCHAR(255) = 'Laptop Pro';
SELECT C.CustomerID, C.FirstName, C.LastName
FROM Customers C
JOIN Orders O ON C.CustomerID = O.CustomerID
JOIN OrderDetails OD ON O.OrderID = OD.OrderID

JOIN Products P ON OD.ProductID = P.ProductID
WHERE P.ProductName = @ProductName;

```
DECLARE @ProductName NVARCHAR(255) = 'Laptop Pro';
SELECT C.CustomerID, C.FirstName, C.LastName
FROM Customers C
JOIN Orders O ON C.CustomerID = O.CustomerID
JOIN OrderDetails OD ON O.OrderID = OD.OrderID
JOIN Products P ON OD.ProductID = P.ProductID
WHERE P.ProductName = @ProductName;
```

100 %

Results | Messages

	CustomerID	FirstName	LastName
1	1	Amit	Sharma

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:

SELECT * FROM Orders;
DECLARE @StartDate DATE = '2023-11-25', @EndDate DATE = '2023-11-27';
SELECT SUM(OD.Quantity * P.Price) AS TotalRevenue
FROM Orders O
JOIN OrderDetails OD ON O.OrderID = OD.OrderID
JOIN Products P ON OD.ProductID = P.ProductID
WHERE O.OrderDate BETWEEN @StartDate AND @EndDate;

```sql
--10
select * from Orders;
DECLARE @StartDate DATE = '2023-11-25', @EndDate DATE = '2023-11-27';
SELECT SUM(OD.Quantity * P.Price) AS TotalRevenue
FROM Orders O
JOIN OrderDetails OD ON O.OrderID = OD.OrderID
JOIN Products P ON OD.ProductID = P.ProductID
WHERE O.OrderDate BETWEEN @StartDate AND @EndDate;
```

100 %

Results | Messages

	TotalRevenue
1	2622.09

 # Aggregate Functions

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 DISTINCT 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(OD.Quantity * P.Price) AS TotalRevenue FROM OrderDetails OD JOIN Products P ON OD.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.

Ans:

DECLARE @CategoryName NVARCHAR(255) = 'Electronic Gadget';
SELECT AVG(OD.Quantity) AS AvgQuantityOrdered FROM OrderDetails OD
JOIN Products P ON OD.ProductID = P.ProductID WHERE P.Category = @CategoryName;

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:

```
DECLARE @CustomerID INT = 6;
SELECT SUM(OD.Quantity * P.Price) AS TotalRevenue FROM OrderDetails OD
JOIN Products P ON OD.ProductID = P.ProductID
WHERE OD.OrderID IN (SELECT OrderID 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 C.CustomerID, C.FirstName, C.LastName, COUNT(O.OrderID) AS OrderCount
FROM Customers C
JOIN Orders O ON C.CustomerID = O.CustomerID
GROUP BY C.CustomerID, C.FirstName, C.LastName
ORDER BY OrderCount DESC;
```

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 P.Category
FROM Products P
JOIN OrderDetails OD ON P.ProductID = OD.ProductID
GROUP BY P.Category
ORDER BY SUM(OD.Quantity) DESC;
```

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 AVG(TotalRevenue) AS AverageOrderValue
FROM (SELECT SUM(OD.Quantity * P.Price) AS TotalRevenue FROM OrderDetails OD
JOIN Products P ON OD.ProductID = P.ProductID GROUP BY OD.OrderID) AS OrderTotals;
```

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

Ans:

```
SELECT AVG(TotalRevenue) AS AverageOrderValue
FROM (SELECT SUM(OD.Quantity * P.Price) AS TotalRevenue
FROM OrderDetails OD
JOIN Products P ON OD.ProductID = P.ProductID
GROUP BY OD.OrderID) AS OrderTotals;
```

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 C.CustomerID, C.FirstName, C.LastName,
COUNT(O.OrderID) AS OrderCount FROM Customers C  JOIN Orders O ON C.CustomerID =
O.CustomerID GROUP BY C.CustomerID, C.FirstName, C.LastName;
```