Assignment 1 – SQL

TechShop, an electronic gadgets shop

Name-Sarthak Londhey

Task:1. Database Design:

1. Create the database named "TechShop"

Query:- CREATE DATABASE TechShop;

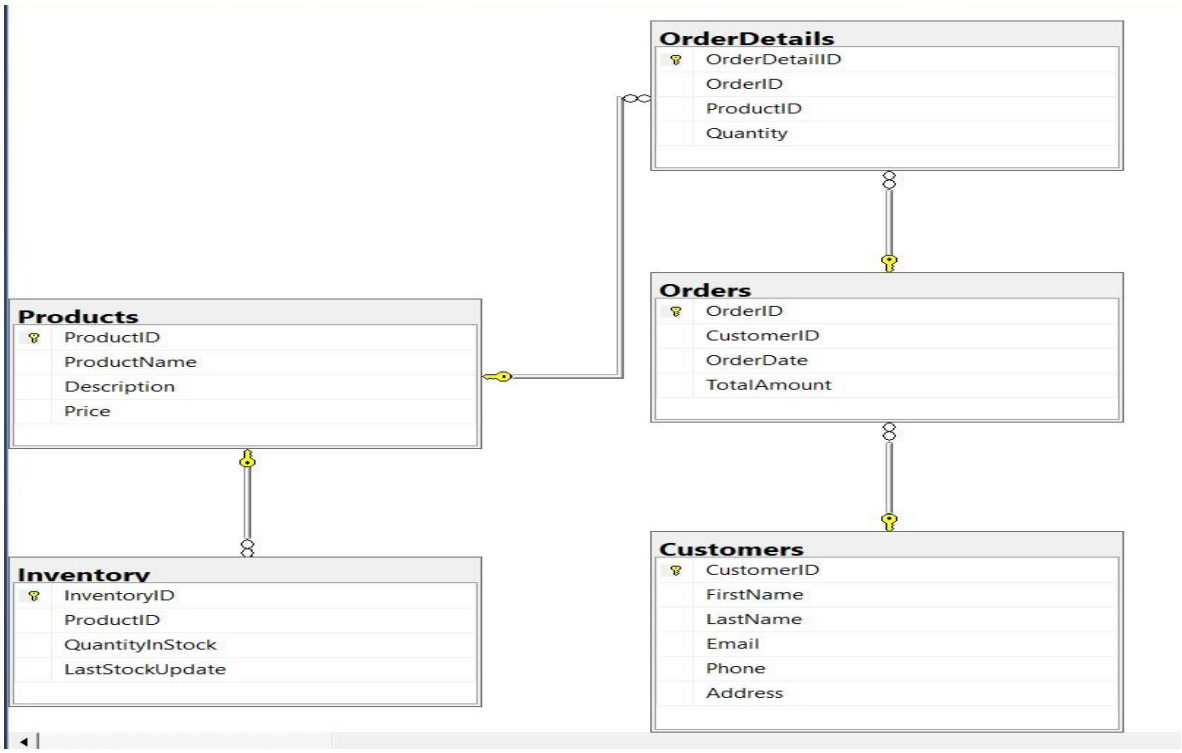
Use TechShop;

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

```
CREATE TABLE Customers (
CustomerID INT PRIMARY KEY IDENTITY(1,1),
   FirstName VARCHAR(50) NOT NULL,
   LastName VARCHAR(50) NOT NULL,
   Email VARCHAR(100) NOT NULL,
   Phone VARCHAR(20),
   Address VARCHAR(255)
);
CREATE TABLE Products (
   ProductID INT PRIMARY KEY IDENTITY(10,1),
   ProductName VARCHAR(100) NOT NULL,
   Description VARCHAR(MAX),
   Price DECIMAL(10, 2) NOT NULL
);
CREATE TABLE Orders (
   OrderID INT PRIMARY KEY IDENTITY(100,1),
   CustomerID INT FOREIGN KEY REFERENCES Customers(CustomerID),
   OrderDate DATE NOT NULL,
   TotalAmount DECIMAL(10, 2) NOT NULL
```

```
);
CREATE TABLE OrderDetails (
    OrderDetailID INT PRIMARY KEY IDENTITY(150,1),
    OrderID INT FOREIGN KEY REFERENCES Orders(OrderID),
    ProductID INT FOREIGN KEY REFERENCES Products(ProductID),
    Quantity INT
);
CREATE TABLE Inventory (
    InventoryID INT PRIMARY KEY IDENTITY(300,1),
    ProductID INT FOREIGN KEY REFERENCES Products(ProductID),
    QuantityInStock INT NOT NULL,
    LastStockUpdate DATE NOT NULL
);
```

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



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

All primary and foreign keys are inserted while creating the table.

5. Insert at least 10 sample records into each of the following tables. a. Customers b. Products c. Orders d. OrderDetails

Customers Table:

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

('Sarthak', 'Londhey', 'sarthak.londhey@gmail.com', '9826996450', 'A1-806, Avasa Township, Indore, Madhya Pradesh'),

('Ramesh', 'Gupta', 'rameshgupta@example.com', '8765432109', '456, Vijay Nagar, Bhopal, Madhya Pradesh'),

('Suresh', 'Patel', 'sureshpatel@example.com', '7654321098', '789, Old City, Jaipur, Rajasthan'),

('Dinesh', 'Verma', 'dineshverma@example.com', '6543210987', '1011, New Colony, Delhi, Delhi'),

('Rajesh', 'Singh', 'rajeshsingh@example.com', '5432109876', '1234, Model Town, Mumbai, Maharashtra'),

('Ganesh', 'Kumar', 'ganeshkumar@example.com', '4321098765', '5678, Banjara Hills, Hyderabad, Telangana'),

('Mahesh', 'Tiwari', 'maheshtiwari@example.com', '3210987654', '9012, Indiranagar, Bengaluru, Karnataka'),

('Narendra', 'Mishra', 'narendramishra@example.com', '2109876543', '1314, Salt Lake City, Kolkata, West Bengal'),

('Pradeep', 'Chauhan', 'pradeepchauhan@example.com', '1098765432', '1516, Beach Road, Chennai, Tamil Nadu'),

('Sanjeev', 'Bhatt', 'sanjeevbhatt@example.com', '9876543210', '1718, MG Road, Kochi, Kerala');

Products Table:

INSERT INTO Products (ProductName, Description, Price) VALUES

('iPhone 14 Pro', '6.1-inch Super Retina XDR display, A16 Bionic chip, 12MP dual camera system', 129999.00),

('Samsung Galaxy S23 Ultra', '6.8-inch Dynamic AMOLED 2X display, Snapdragon 8 Gen 2, 200MP camera', 119999.00),

('OnePlus 11', '6.7-inch Fluid AMOLED display, Snapdragon 8 Gen 2, 50MP camera', 59999.00),

('Xiaomi 13 Pro', '6.7-inch AMOLED display, Snapdragon 8 Gen 2, 50MP camera', 69999.00),

('Google Pixel 7 Pro', '6.7-inch LTPO OLED display, Google Tensor G2, 50MP camera', 84999.00),

('MacBook Pro M2', '13.3-inch Liquid Retina XDR display, M2 chip, 8GB RAM, 256GB SSD', 149999.00),

('Dell XPS 13', '13.4-inch InfinityEdge display, Intel Core i7-13700H, 16GB RAM, 512GB SSD', 129999.00),

('Lenovo ThinkPad X1 Carbon', '14-inch OLED display, Intel Core i7-13600H, 16GB RAM, 512GB SSD', 139999.00),

('HP Spectre x360', '13.3-inch AMOLED display, Intel Core i7-13600H, 16GB RAM, 512GB SSD', 129999.00),

('Acer Predator Helios 16', '16-inch IPS display, Intel Core i9-13900HX, 32GB RAM, 2TB SSD', 179999.00);

Orders Table:

INSERT INTO Orders (CustomerID, OrderDate, TotalAmount) VALUES

(1, '2023-01-25', 129999.00),

(2, '2023-04-01', 59999.00),

(3, '2023-08-10', 69999.00),

(4, '2023-11-15', 149999.00),

(5, '2023-12-20', 119999.00),

(6, '2024-01-25', 84999.00),

(7, '2024-05-01', 129999.00),

(8, '2024-06-05', 139999.00),

(9, '2024-07-10', 179999.00),

(10, '2024-08-24', 129999.00);

OrderDetails Table:

INSERT INTO OrderDetails (OrderID, ProductID, Quantity) VALUES

(100, 10, 1),

(101, 11, 1),

(102, 12, 1),

(103, 13, 1),

(104, 14, 1),

(105, 15, 1),

(106, 16, 1),

(107, 17, 1),

(108, 18, 1),

(109, 19, 1);

Inventory Table:

INSERT INTO Inventory (ProductID, QuantityInStock, LastStockUpdate) VALUES

(10, 10, '2023-12-31'),

(11, 15, '2023-12-31'),

(12, 20, '2023-12-31'),

(13, 5, '2023-12-31'),

(14, 8, '2023-12-31'),

(15, 12, '2023-12-31'),

(16, 7, '2023-12-31'),

(17, 11, '2023-12-31'),

(18, 6, '2023-12-31'),

(19, 9, '2023-12-31');

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;
```

100 %

Results | Messages

	FirstName	LastName	Email
1	Sarthak	Londhey	sarthak.londhey@gmail.com
2	Ramesh	Gupta	rameshgupta@example.com
3	Suresh	Patel	sureshpatel@example.com
4	Dinesh	Verma	dineshverma@example.com
5	Rajesh	Singh	rajeshsingh@example.com
6	Ganesh	Kumar	ganeshkumar@example.com
7	Mahesh	Tiwari	maheshtiwari@example.com
8	Narendra	Mishra	narendramishra@example.com
9	Pradeep	Chauhan	pradeepchauhan@example.com
10	Sanjeev	Bhatt	sanjeevbhatt@example.com

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

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

100 %

Results | Messages

	OrderID	OrderDate	CustomerName
1	100	2023-01-25	Sarthak Londhey
2	101	2023-04-01	Ramesh Gupta
3	102	2023-08-10	Suresh Patel
4	103	2023-11-15	Dinesh Verma
5	104	2023-12-20	Rajesh Singh
6	105	2024-01-25	Ganesh Kumar
7	106	2024-05-01	Mahesh Tiwari
8	107	2024-06-05	Narendra Mishra
9	108	2024-07-10	Pradeep Chauhan
10	109	2024-08-24	Sanjeev Bhatt

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 (FirstName, LastName, Email, Phone, Address)
VALUES ('Ayush', 'Agrawal', 'ayush.agrawal@example.com', '1234567890', '456, Sudama Nagar, Indore, Madhya Pradesh');

select * from Customers;
```

100 %

Results | Messages

	CustomerID	FirstName	LastName	Email	Phone	Address
1	1	Sarthak	Londhey	sarthak.londhey@gmail.com	9826996450	A1-806, Avasa Township, Indore, Madhya Pradesh
2	2	Ramesh	Gupta	rameshgupta@example.com	8765432109	456, Vijay Nagar, Bhopal, Madhya Pradesh
3	3	Suresh	Patel	sureshpatel@example.com	7654321098	789, Old City, Jaipur, Rajasthan
4	4	Dinesh	Verma	dineshverma@example.com	6543210987	1011, New Colony, Delhi, Delhi
5	5	Rajesh	Singh	rajeshsingh@example.com	5432109876	1234, Model Town, Mumbai, Maharashtra
6	6	Ganesh	Kumar	ganeshkumar@example.com	4321098765	5678, Banjara Hills, Hyderabad, Telangana
7	7	Mahesh	Tiwari	maheshtiwari@example.com	3210987654	9012, Indiranagar, Bengaluru, Karnataka
8	8	Narendra	Mishra	narendramishra@example.com	2109876543	1314, Salt Lake City, Kolkata, West Bengal
9	9	Pradeep	Chauhan	pradeepchauhan@example.com	1098765432	1516, Beach Road, Chennai, Tamil Nadu
10	10	Sanjeev	Bhatt	sanjeevbhatt@example.com	9876543210	1718, MG Road, Kochi, Kerala
11	11	Ayush	Agrawal	ayush.agrawal@example.com	1234567890	456, Sudama Nagar, Indore, Madhya Pradesh

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;

select * from Products;
```

100 %

Results | Messages

	ProductID	ProductName	Description	Price	Categories
1	10	iPhone 14 Pro	6.1-inch Super Retina XDR display, A16 Bionic chip, 1...	142998.90	Smartphones
2	11	Samsung Galaxy S23 Ultra	6.8-inch Dynamic AMOLED 2X display, Snapdragon 8 ...	131998.90	Smartphones
3	12	OnePlus 11	6.7-inch Fluid AMOLED display, Snapdragon 8 Gen 2, ...	65998.90	Smartphones
4	13	Xiaomi 13 Pro	6.7-inch AMOLED display, Snapdragon 8 Gen 2, 50M...	76998.90	Smartphones
5	14	Google Pixel 7 Pro	6.7-inch LTPO OLED display, Google Tensor G2, 50M...	93498.90	Smartphones
6	15	MacBook Pro M2	13.3-inch Liquid Retina XDR display, M2 chip, 8GB RA...	164998.90	Laptops
7	16	Dell XPS 13	13.4-inch InfinityEdge display, Intel Core i7-13700H, 1...	142998.90	Laptops
8	17	Lenovo ThinkPad X1 Carbon	14-inch OLED display, Intel Core i7-13600H, 16GB RA...	153998.90	Laptops
9	18	HP Spectre x360	13.3-inch AMOLED display, Intel Core i7-13600H, 16G...	142998.90	Laptops
10	19	Acer Predator Helios 16	16-inch IPS display, Intel Core i9-13900HX, 32GB RA...	197998.90	Laptops

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 OrderDetails
WHERE OrderID = 104;

select * from OrderDetails;
```

100 %

Results | Messages

	OrderDetailID	OrderID	ProductID	Quantity
1	150	100	10	1
2	151	101	11	1
3	152	102	12	1
4	153	103	13	1
5	155	105	15	1
6	156	106	16	1
7	157	107	17	1
8	158	108	18	1
9	159	109	19	1

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 (CustomerID, OrderDate, TotalAmount)
VALUES (1, '2024-09-20', 79999.00);

select * from Orders;
```

100 %

Results | Messages

	OrderID	CustomerID	OrderDate	TotalAmount
1	100	1	2023-01-25	129999.00
2	101	2	2023-04-01	59999.00
3	102	3	2023-08-10	69999.00
4	103	4	2023-11-15	149999.00
5	105	6	2024-01-25	84999.00
6	106	7	2024-05-01	129999.00
7	107	8	2024-06-05	139999.00
8	108	9	2024-07-10	179999.00
9	109	10	2024-08-24	129999.00
10	111	1	2024-09-20	79999.00

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 = 'newemail@example.com',
    Address = '1000, New Address, City, State'
WHERE CustomerID = 3;

select * from Customers;
```

100 %

Results | Messages

	CustomerID	FirstName	LastName	Email	Phone	Address
1	1	Sarthak	Londhey	sarthak.londhey@gmail.com	9826996450	A1-806, Avasa Township, Indore, Madhya Pradesh
2	2	Ramesh	Gupta	rameshgupta@example.com	8765432109	456, Vijay Nagar, Bhopal, Madhya Pradesh
3	3	Suresh	Patel	newemail@example.com	7654321098	1000, New Address, City, State
4	4	Dinesh	Verma	dineshverma@example.com	6543210987	1011, New Colony, Delhi, Delhi
5	5	Rajesh	Singh	rajeshsingh@example.com	5432109876	1234, Model Town, Mumbai, Maharashtra
6	6	Ganesh	Kumar	ganeshkumar@example.com	4321098765	5678, Banjara Hills, Hyderabad, Telangana
7	7	Mahesh	Tiwari	maheshtiwari@example.com	3210987654	9012, Indiranagar, Bengaluru, Karnataka
8	8	Narendra	Mishra	narendramishra@example.com	2109876543	1314, Salt Lake City, Kolkata, West Bengal
9	9	Pradeep	Chauhan	pradeepchauhan@example.com	1098765432	1516, Beach Road, Chennai, Tamil Nadu
10	10	Sanjeev	Bhatt	sanjeevbhatt@example.com	9876543210	1718, MG Road, Kochi, Kerala
11	11	Ayush	Agrawal	ayush.agrawal@example.com	1234567890	456, Sudama Nagar, Indore, Madhya Pradesh

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.

Sarthak Londhey_A...LUN5NR\acer (51))*

```
INSERT INTO OrderDetails (OrderID, ProductID, Quantity) VALUES
(109, 19, 1);

select * from OrderDetails;



UPDATE Orders
SET TotalAmount = (
    SELECT (od.Quantity * p.Price)
    FROM OrderDetails od
    JOIN Products p ON od.ProductID = p.ProductID
    WHERE od.Or
);

select * from Orders;
```

table TechShop.dbo.Products

100 %

Results | Messages

	OrderID	CustomerID	OrderDate	TotalAmount
1	100	1	2023-01-25	142998.90
2	101	2	2023-04-01	131998.90
3	102	3	2023-08-10	65998.90
4	103	4	2023-11-15	76998.90
5	105	6	2024-01-25	149999.00
6	106	7	2024-05-01	129999.00
7	107	8	2024-06-05	139999.00
8	108	9	2024-07-10	129999.00
9	109	10	2024-08-24	179999.00
10	112	1	2024-09-20	179999.00

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.

```
DECLARE @CustomerID INT = 6
DELETE FROM OrderDetails
WHERE OrderID IN (
    SELECT OrderID
    FROM Orders
    WHERE CustomerID = @CustomerID
);
DELETE FROM Orders
WHERE CustomerID = @CustomerID;

select * from OrderDetails;
```

100 %

Results | Messages

	OrderDetailID	OrderID	ProductID	Quantity
1	150	100	10	1
2	151	101	11	1
3	152	102	12	1
4	153	103	13	1
5	156	106	16	1
6	157	107	17	1
7	158	108	18	1
8	159	109	19	1
9	162	112	19	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.

Sarthak Londhey_A...LUN5NR\acer (51))*

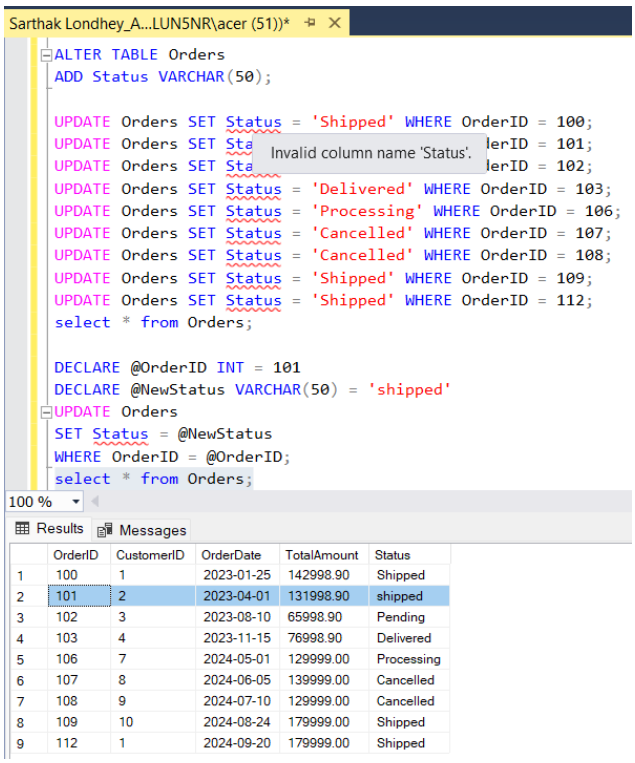
```
INSERT INTO Products (ProductName, Description, Price)
VALUES ('Smartwatch X200', 'A high-end smartwatch with various health tracking features and a sleek design.', 19999.00);

select * from Products;
```

100 %

Results | Messages

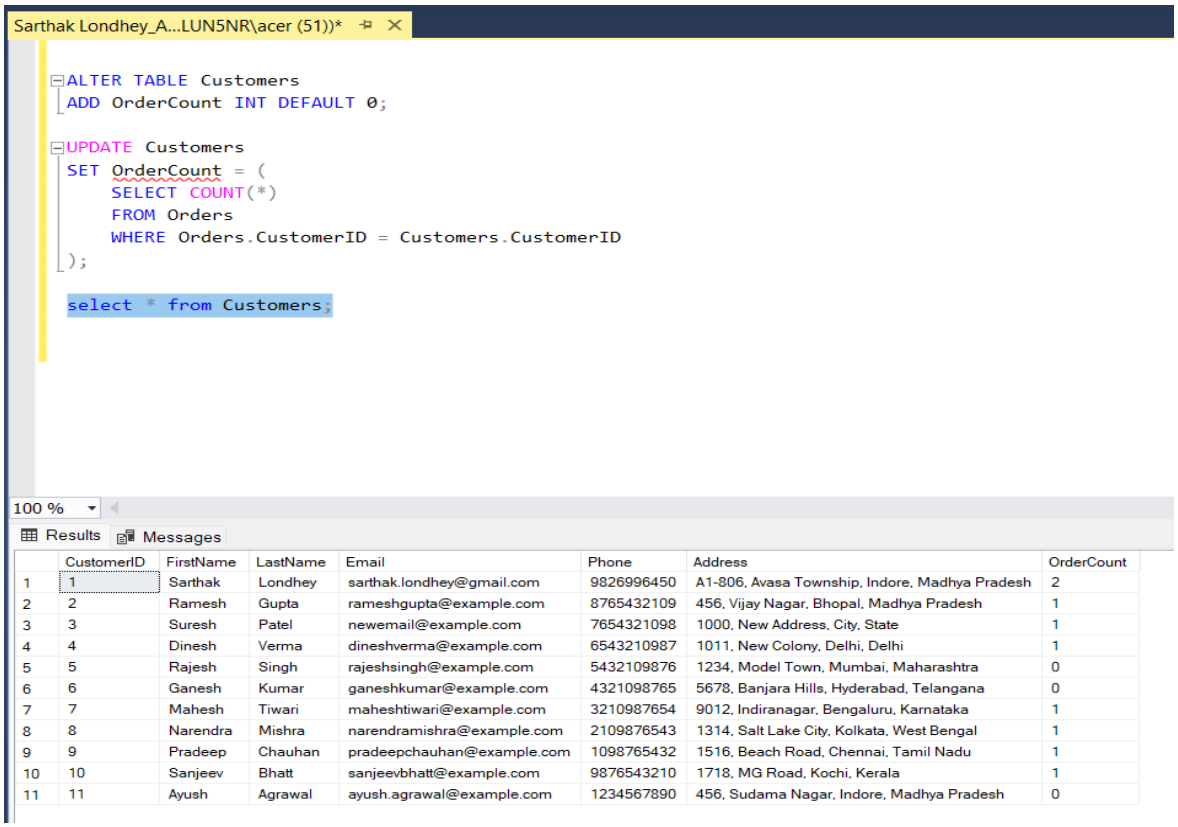
	ProductID	ProductName	Description	Price
1	10	iPhone 14 Pro	6.1-inch Super Retina XDR display, A16 Bionic chip, 1...	142998.90
2	11	Samsung Galaxy S23 Ultra	6.8-inch Dynamic AMOLED 2X display, Snapdragon 8 ...	131998.90
3	12	OnePlus 11	6.7-inch Fluid AMOLED display, Snapdragon 8 Gen 2, ...	65998.90
4	13	Xiaomi 13 Pro	6.7-inch AMOLED display, Snapdragon 8 Gen 2, 50M...	76998.90
5	14	Google Pixel 7 Pro	6.7-inch LTPO OLED display, Google Tensor G2, 50M...	93498.90
6	15	MacBook Pro M2	13.3-inch Liquid Retina XDR display, M2 chip, 8GB RA...	149999.00
7	16	Dell XPS 13	13.4-inch InfinityEdge display, Intel Core i7-13700H, 1...	129999.00
8	17	Lenovo ThinkPad X1 Carbon	14-inch OLED display, Intel Core i7-13600H, 16GB RA...	139999.00
9	18	HP Spectre x360	13.3-inch AMOLED display, Intel Core i7-13600H, 16G...	129999.00
10	19	Acer Predator Helios 16	16-inch IPS display, Intel Core i9-13900HX, 32GB RA...	179999.00
11	20	Smartwatch X200	A high-end smartwatch with various health tracking fea...	19999.00

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.



	OrderID	CustomerID	OrderDate	TotalAmount	Status
1	100	1	2023-01-25	142998.90	Shipped
2	101	2	2023-04-01	131998.90	shipped
3	102	3	2023-08-10	65998.90	Pending
4	103	4	2023-11-15	76998.90	Delivered
5	106	7	2024-05-01	129999.00	Processing
6	107	8	2024-06-05	139999.00	Cancelled
7	108	9	2024-07-10	129999.00	Cancelled
8	109	10	2024-08-24	179999.00	Shipped
9	112	1	2024-09-20	179999.00	Shipped

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.



	CustomerID	FirstName	LastName	Email	Phone	Address	OrderCount
1	1	Sarthak	Londhey	sarthak.londhey@gmail.com	9826996450	A1-806, Avasa Township, Indore, Madhya Pradesh	2
2	2	Ramesh	Gupta	rameshgupta@example.com	8765432109	456, Vijay Nagar, Bhopal, Madhya Pradesh	1
3	3	Suresh	Patel	newemail@example.com	7654321098	1000, New Address, City, State	1
4	4	Dinesh	Verma	dineshverma@example.com	6543210987	1011, New Colony, Delhi, Delhi	1
5	5	Rajesh	Singh	rajeshsingh@example.com	5432109876	1234, Model Town, Mumbai, Maharashtra	0
6	6	Ganesh	Kumar	ganeshkumar@example.com	4321098765	5678, Banjara Hills, Hyderabad, Telangana	0
7	7	Mahesh	Tiwari	maheshtiwari@example.com	3210987654	9012, Indiranagar, Bengaluru, Karnataka	1
8	8	Narendra	Mishra	narendramishra@example.com	2109876543	1314, Salt Lake City, Kolkata, West Bengal	1
9	9	Pradeep	Chauhan	pradeepchauhan@example.com	1098765432	1516, Beach Road, Chennai, Tamil Nadu	1
10	10	Sanjeev	Bhatt	sanjeevbhatt@example.com	9876543210	1718, MG Road, Kochi, Kerala	1
11	11	Ayush	Agrawal	ayush.agrawal@example.com	1234567890	456, Sudama Nagar, Indore, Madhya Pradesh	0

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.FirstName, C.LastName, C.Email, C.Phone
FROM Orders AS O
JOIN Customers AS C ON O.CustomerID = C.CustomerID;
```

100 %

Results | Messages

	OrderID	OrderDate	TotalAmount	FirstName	LastName	Email	Phone
1	100	2023-01-25	142998.90	Sarthak	Londhey	sarthak.londhey@gmail.com	9826996450
2	101	2023-04-01	131998.90	Ramesh	Gupta	rameshgupta@example.com	8765432109
3	102	2023-08-10	65998.90	Suresh	Patel	newemail@example.com	7654321098
4	103	2023-11-15	76998.90	Dinesh	Verma	dineshverma@example.com	6543210987
5	106	2024-05-01	129999.00	Mahesh	Tiwari	maheshtiwari@example.com	3210987654
6	107	2024-06-05	139999.00	Narendra	Mishra	narendramishra@example.com	2109876543
7	108	2024-07-10	129999.00	Pradeep	Chauhan	pradeepchauhan@example.com	1098765432
8	109	2024-08-24	179999.00	Sanjeev	Bhatt	sanjeevbhatt@example.com	9876543210
9	112	2024-09-20	179999.00	Sarthak	Londhey	sarthak.londhey@gmail.com	9826996450

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(OD.Quantity * P.Price) AS TotalRevenue
FROM OrderDetails AS OD
JOIN Products AS P ON OD.ProductID = P.ProductID
GROUP BY P.ProductName;
```

100 %

Results | Messages

	ProductName	TotalRevenue
1	Acer Predator Helios 16	359998.00
2	Dell XPS 13	129999.00
3	HP Spectre x360	129999.00
4	iPhone 14 Pro	142998.90
5	Lenovo ThinkPad X1 Carbon	139999.00
6	OnePlus 11	65998.90
7	Samsung Galaxy S23 Ultra	131998.90
8	Xiaomi 13 Pro	76998.90

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

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

100 %

Results | Messages

	FirstName	LastName	Email	Phone
1	Sarthak	Londhey	sarthak.londhey@gmail.com	9826996450
2	Ramesh	Gupta	rameshgupta@example.com	8765432109
3	Suresh	Patel	newemail@example.com	7654321098
4	Dinesh	Verma	dineshverma@example.com	6543210987
5	Mahesh	Tiwari	maheshtiwari@example.com	3210987654
6	Narendra	Mishra	narendramishra@example.com	2109876543
7	Pradeep	Chauhan	pradeepchauhan@example.com	1098765432
8	Sanjeev	Bhatt	sanjeevbhatt@example.com	9876543210

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.

```sql
209 SELECT top 1 p.ProductName,
210        SUM(od.Quantity) AS TotalQuantityOrdered
211 FROM OrderDetails od
212 JOIN Products p ON od.ProductID = p.ProductID
213 JOIN Inventory i ON p.ProductID = i.ProductID
214 WHERE i.QuantityInStock > 0
215 GROUP BY p.ProductName
216 ORDER BY TotalQuantityOrdered DESC;
217
218
```

100 %

Results | Messages

	ProductName	TotalQuantityOrdered
1	Acer Predator Helios 16	2

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

```
219 ALTER TABLE Products
220 ADD Categories VARCHAR(255);
221 UPDATE Products
222 SET Categories = 'Smartphones'
223 WHERE ProductID IN (1000, 1001, 1002, 1003, 1004);
224 UPDATE Products
225 SET Categories = 'Laptops'
226 WHERE ProductID IN (1005, 1006, 1007, 1008, 1009);
227 UPDATE Products
228 SET Categories = 'Smartwatches'
229 WHERE ProductID = 1010;
230
231
232 SELECT ProductName,
233        Categories AS CategoryName
234 FROM Products;
235
```

100 %

Results | Messages

	ProductName	CategoryName
1	iPhone 14 Pro	Smartphones
2	Samsung Galaxy S23 Ultra	Smartphones
3	OnePlus 11	Smartphones
4	Xiaomi 13 Pro	Smartphones
5	Google Pixel 7 Pro	Smartphones
6	MacBook Pro M2	Laptops
7	Dell XPS 13	Laptops
8	Lenovo ThinkPad X1 Carbon	Laptops
9	HP Spectre x360	Laptops
10	Acer Predator Helios 16	Laptops
11	Smartwatch X200	Smartwatches

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,
       AVG(o.TotalAmount) AS AverageOrderValue
FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
GROUP BY c.CustomerID, c.FirstName, c.LastName;
```

100 %

Results | Messages

	FirstName	LastName	AverageOrderValue
1	Sarthak	Londhey	161498.950000
2	Ramesh	Gupta	131998.900000
3	Suresh	Patel	65998.900000
4	Dinesh	Verma	76998.900000
5	Mahesh	Tiwari	129999.000000
6	Narendra	Mishra	139999.000000
7	Pradeep	Chauhan	129999.000000
8	Sanjeev	Bhatt	179999.000000

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

```sql
SELECT o.OrderID,
       c.FirstName,
       c.LastName,
       c.Email,
       c.Phone,
       o.TotalAmount AS TotalRevenue
FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
WHERE o.TotalAmount = (
    SELECT MAX(TotalAmount)
    FROM Orders
);
```

100 %

Results | Messages

	OrderID	FirstName	LastName	Email	Phone	TotalRevenue
1	109	Sanjeev	Bhatt	sanjeevbhatt@example.com	9876543210	179999.00
2	112	Sarthak	Londhey	sarthak.londhey@gmail.com	9826996450	179999.00

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

```sql
SELECT p.ProductName,
       COUNT(od.OrderID) AS NumberOfOrders
FROM Products p
LEFT JOIN OrderDetails od ON p.ProductID = od.ProductID
GROUP BY p.ProductID, p.ProductName;
```

100 %

Results | Messages

	ProductName	NumberOfOrders
1	iPhone 14 Pro	1
2	Samsung Galaxy S23 Ultra	1
3	OnePlus 11	1
4	Xiaomi 13 Pro	1
5	Google Pixel 7 Pro	0
6	MacBook Pro M2	0
7	Dell XPS 13	1
8	Lenovo ThinkPad X1 Carbon	1
9	HP Spectre x360	1
10	Acer Predator Helios 16	2
11	Smartwatch X200	0

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.

```
DECLARE @ProductName NVARCHAR(255) = 'iPhone 14 Pro';

SELECT DISTINCT c.CustomerID,
                c.FirstName,
                c.LastName,
                c.Email,
                c.Phone
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	Email	Phone
1	1	Sarthak	Londhey	sarthak.londhey@gmail.com	9826996450

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.

```
DECLARE @StartDate DATE = '2023-01-01';
DECLARE @EndDate DATE = '2024-06-30';

SELECT SUM(TotalAmount) AS TotalRevenue
FROM Orders
WHERE OrderDate BETWEEN @StartDate AND @EndDate;
```

100 %

Results | Messages

	TotalRevenue
1	687993.60

Task 4. Subquery and its type:

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

```
SELECT c.CustomerID, c.FirstName, c.LastName, c.Email, c.Phone, c.Address
FROM Customers c
LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
WHERE o.OrderID IS NULL;
```

100 %

Results | Messages

	CustomerID	FirstName	LastName	Email	Phone	Address
1	5	Rajesh	Singh	rajeshsingh@example.com	5432109876	1234, Model Town, Mumbai, Maharashtra
2	6	Ganesh	Kumar	ganeshkumar@example.com	4321098765	5678, Banjara Hills, Hyderabad, Telangana
3	11	Ayush	Agrawal	ayush.agrawal@example.com	1234567890	456, Sudama Nagar, Indore, Madhya Pradesh

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

```
SELECT SUM(QuantityInStock) AS TotalProductsAvailable
FROM Inventory;
```

100 %

Results | Messages

	TotalProductsAvailable
1	103

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

```
SELECT SUM(TotalAmount) AS TotalRevenue
FROM Orders
WHERE Status IN ('Delivered', 'Shipped');
```

100 %

Results | Messages

	TotalRevenue
1	711994.70

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.

```
DECLARE @Categories VARCHAR(50) = 'Laptops';
SELECT p.Categories, AVG(od.Quantity) AS AvgQuantityOrdered
FROM Products p
JOIN OrderDetails od ON p.ProductID = od.ProductID
WHERE p.Categories = @Categories
GROUP BY p.Categories;
```

100 %

Results | Messages

	Categories	AvgQuantityOrdered
1	Laptops	1

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.

```
DECLARE @customerID INT = 4;
SELECT c.CustomerID, CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName, SUM(o.TotalAmount) AS TotalRevenue
FROM Customers c
JOIN Orders o ON c.CustomerID = o.CustomerID
WHERE c.CustomerID = @customerID
GROUP BY c.CustomerID, c.FirstName, c.LastName;
```

100 %

Results | Messages

	CustomerID	CustomerName	TotalRevenue
1	4	Dinesh Verma	76998.90

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, CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName, 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;
```

100 %

Results | Messages

	CustomerID	CustomerName	OrderCount
1	1	Sarthak Londhey	2
2	2	Ramesh Gupta	1
3	3	Suresh Patel	1
4	4	Dinesh Verma	1
5	7	Mahesh Tiwari	1
6	8	Narendra Mishra	1
7	9	Pradeep Chauhan	1
8	10	Sanjeev Bhatt	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.

```sql
SELECT top 1 p.Categories, SUM(od.Quantity) AS TotalQuantityOrdered
FROM Products p
JOIN OrderDetails od ON p.ProductID = od.ProductID
GROUP BY p.Categories
ORDER BY TotalQuantityOrdered DESC;
```

100 %

Results | Messages

	Categories	TotalQuantityOrdered
1	Laptops	5

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.

```sql
SELECT TOP 1 C.FirstName, C.LastName, SUM(OD.Quantity * P.Price) AS TotalSpent
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.FirstName, C.LastName
ORDER BY TotalSpent DESC;
```

100 %

Results | Messages

	FirstName	LastName	TotalSpent
1	Sarthak	Londhey	322997.90

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, AVG(O.TotalAmount) AS AverageOrderValue
FROM Customers C
JOIN Orders O ON C.CustomerID = O.CustomerID
GROUP BY C.CustomerID, C.FirstName, C.LastName;
```

100 %

Results | Messages

	CustomerID	FirstName	LastName	AverageOrderValue
1	1	Sarthak	Londhey	161498.950000
2	2	Ramesh	Gupta	131998.900000
3	3	Suresh	Patel	65998.900000
4	4	Dinesh	Verma	76998.900000
5	7	Mahesh	Tiwari	129999.000000
6	8	Narendra	Mishra	139999.000000
7	9	Pradeep	Chauhan	129999.000000
8	10	Sanjeev	Bhatt	179999.000000

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 FirstName, LastName, OrderCount from Customers
order by ordercount desc;
```

100 %

Results | Messages

	FirstName	LastName	OrderCount
1	Sarthak	Londhey	2
2	Ramesh	Gupta	1
3	Suresh	Patel	1
4	Dinesh	Verma	1
5	Mahesh	Tiwari	1
6	Narendra	Mishra	1
7	Pradeep	Chauhan	1
8	Sanjeev	Bhatt	1
9	Ayush	Agrawal	0
10	Rajesh	Singh	0
11	Ganesh	Kumar	0