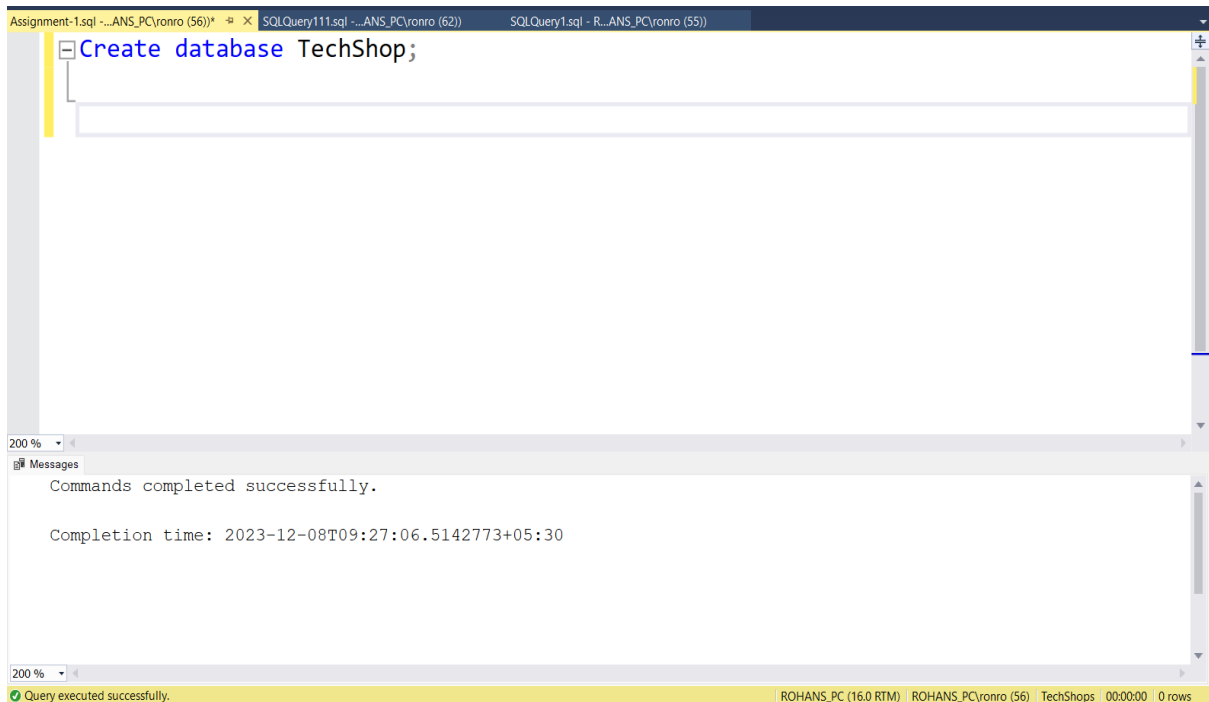


# Assignment 1

## Task 1: Database Design

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



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays the command `Create database TechShop;`. The bottom pane, labeled 'Messages', shows the output: 'Commands completed successfully.' and 'Completion time: 2023-12-08T09:27:06.5142773+05:30'. The status bar at the bottom indicates 'Query executed successfully.' and provides details about the server and database.

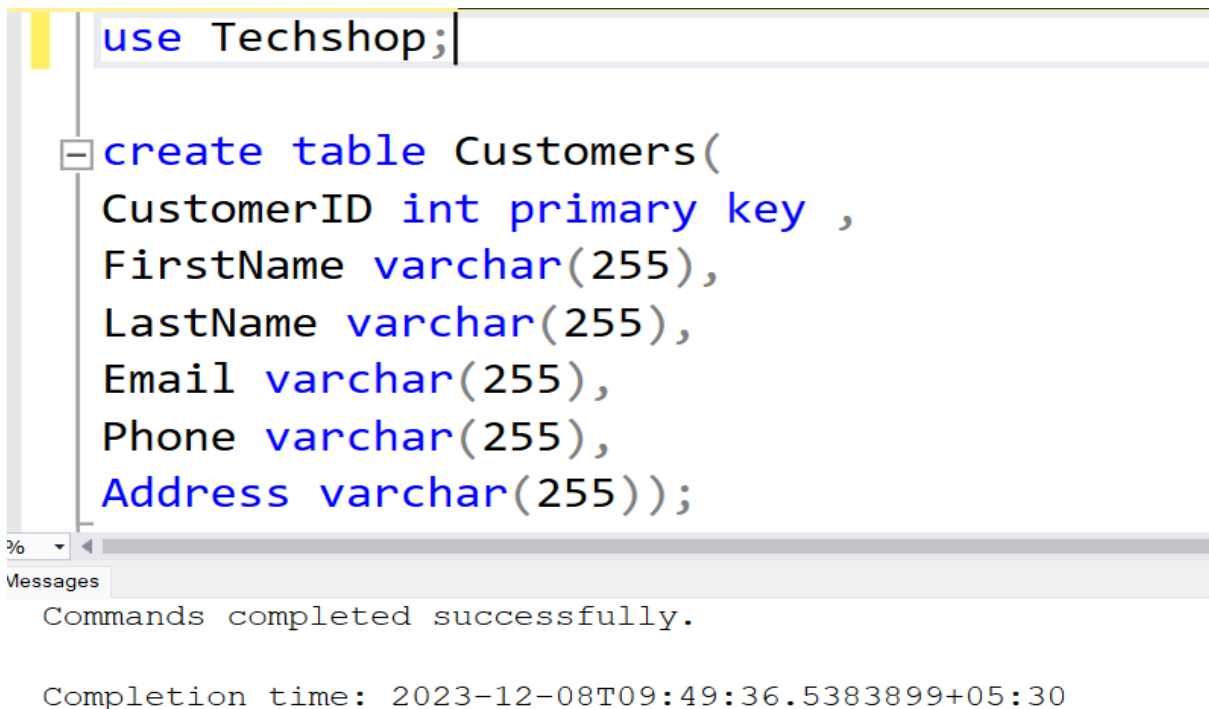
```
Create database TechShop;
```

Commands completed successfully.

Completion time: 2023-12-08T09:27:06.5142773+05:30

Query executed successfully. ROHANS\_PC (16.0 RTM) ROHANS\_PC\ronro (56) TechShops 00:00:00 0 rows

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



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays the command `use Techshop;`. The bottom pane, labeled 'Messages', shows the output: 'Commands completed successfully.' and 'Completion time: 2023-12-08T09:49:36.5383899+05:30'. The status bar at the bottom indicates 'Query executed successfully.' and provides details about the server and database.

```
use Techshop;
```

```
create table Customers(  
  CustomerID int primary key ,  
  FirstName varchar(255),  
  LastName varchar(255),  
  Email varchar(255),  
  Phone varchar(255),  
  Address varchar(255));
```

Commands completed successfully.

Completion time: 2023-12-08T09:49:36.5383899+05:30

```
create table Products(  
ProductID int primary key ,  
ProductName varchar(255),  
Description varchar(255),  
Price int);
```

Messages  
Commands completed successfully.  
Completion time: 2023-12-08T09:31:43.1377153+05:30

```
create table Orders(  
OrderID int primary key,  
CustomerID int ,  
FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),  
OrderDate Date,  
TotalAMount int);
```

% Messages  
Commands completed successfully.  
Completion time: 2023-12-08T09:32:57.1531153+05:30

```
create table Orders(  
OrderID int primary key,  
CustomerID int ,  
FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),  
OrderDate Date,  
TotalAMount int);
```

% Messages  
Commands completed successfully.  
Completion time: 2023-12-08T09:32:57.1531153+05:30

```
create table OrderDetails(  
  OrderDetailId int primary key ,  
  OrderId int ,  
  Foreign key (OrderId) references Orders(OrderID),  
  ProductID int ,  
  Foreign Key (ProductID) references Products(ProductID),  
  Quantity int );
```

Messages

Commands completed successfully.

Completion time: 2023-12-08T09:33:28.8146597+05:30

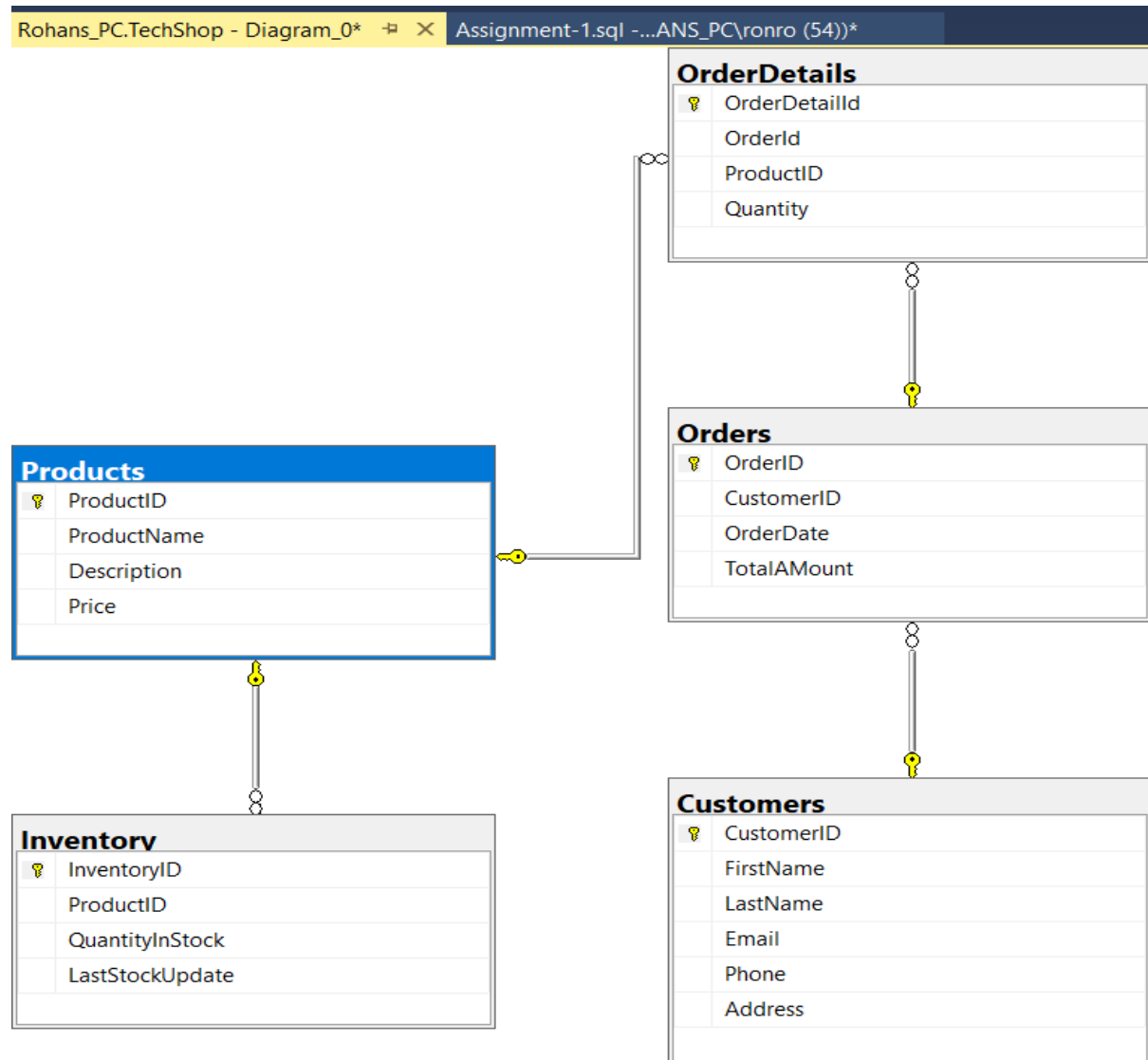
```
create table Inventory (  
  InventoryID INT PRIMARY KEY,  
  ProductID INT,  
  QuantityInStock INT,  
  LastStockUpdate DATE,  
  FOREIGN KEY (ProductID) REFERENCES Products(ProductID));
```

Messages

Commands completed successfully.

Completion time: 2023-12-08T09:34:54.5042291+05:30

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



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

Primary Key and Foreign Key constraints:

- Primary keys and foreign keys are already defined in the table creation statements.

### 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, 'John', 'Doe', 'john.doe@email.com', '1234567890', '123 Main St'),
(2, 'Jane', 'Smith', 'jane.smith@email.com', '9876543210', '456 Oak St'),
(3, 'Bob', 'Johnson', 'bob.johnson@email.com', '5551234567', '789 Pine St'),
(4, 'Alice', 'Williams', 'alice.williams@email.com', '1112223333', '456 Elm St'),
(5, 'Charlie', 'Brown', 'charlie.brown@email.com', '4445556666', '789 Oak St'),
(6, 'Eva', 'Davis', 'eva.davis@email.com', '7778889999', '123 Maple St'),
(7, 'Frank', 'Miller', 'frank.miller@email.com', '2223334444', '456 Birch St'),
(8, 'Grace', 'Jones', 'grace.jones@email.com', '9998887777', '789 Cedar St'),
(9, 'Henry', 'Garcia', 'henry.garcia@email.com', '3334445555', '123 Walnut St'),
(10, 'Ivy', 'Clark', 'ivy.clark@email.com', '6667778888', '456 Pine St');

(10 rows affected)

Completion time: 2023-12-08T09:53:12.6877296+05:30
```

```
INSERT INTO Products (ProductID, ProductName, Description, Price)
VALUES
(1, 'Laptop', 'Powerful laptop for professional use', 9999.99),
(2, 'Smartphone', 'High-performance smartphone', 6999.99),
(3, 'Water Bottle', 'To stored water', 299.99),
(4, 'Smartwatch', 'Fitness tracker and smartwatch combo', 1499.99),
(5, 'Washing Machine', 'Washes the clothes', 1299.99),
(6, 'Clocks', 'Tells the time', 599.99),
(7, 'Table Desk', 'For Doing Works', 799.99),
(8, 'Iron Box', 'To Iron the Clothes', 499.99),
(9, 'Mosquito Bat', 'To Kill Mosquitoes', 299.99),
(10, 'Bluetooth Speaker', 'Portable Bluetooth speaker with rich sound', 499.99);

(10 rows affected)

Completion time: 2023-12-08T09:54:05.4694554+05:30
```

```
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)
VALUES
(1, 1, '2023-01-01', 9999.99),
(2, 2, '2023-01-02', 6999.99),
(3, 3, '2023-01-03', 2999.99),
(4, 4, '2023-01-04', 1499.99),
(5, 5, '2023-01-05', 2599.98),
(6, 6, '2023-01-06', 1199.99),
(7, 7, '2023-01-07', 799.99),
(8, 8, '2023-01-08', 999.99),
(9, 9, '2023-01-09', 899.97),
(10, 10, '2023-01-10', 1499.98);

(10 rows affected)

Completion time: 2023-12-08T09:54:43.3572485+05:30
```

```
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity)
VALUES
    (1, 1, 1, 2),
    (2, 2, 3, 1),
    (3, 3, 5, 3),
    (4, 4, 7, 6),
    (5, 5, 2, 2),
    (6, 6, 6, 1),
    (7, 7, 10, 6),
    (8, 8, 4, 2),
    (9, 9, 8, 1),
    (10, 10, 9, 3);
```

6  
Messages

(10 rows affected)

Completion time: 2023-12-08T09:56:42.2068969+05:30

```
INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate)
VALUES
    (1, 1, 50, '2023-01-01'),
    (2, 2, 30, '2023-01-02'),
    (3, 3, 20, '2023-01-03'),
    (4, 4, 10, '2023-01-04'),
    (5, 5, 50, '2023-01-05'),
    (6, 6, 15, '2023-01-06'),
    (7, 7, 25, '2023-01-07'),
    (8, 8, 40, '2023-01-08'),
    (9, 9, 5, '2023-01-09'),
    (10, 10, 30, '2023-01-10');
```

%  
Messages

(10 rows affected)

Completion time: 2023-12-08T09:57:24.0020991+05:30

## 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;
```

200 %

Results Messages

	FirstName	LastName	Email
1	John	Doe	john.doe@email.com
2	Jane	Smith	jane.smith@email.com
3	Bob	Johnson	bob.johnson@email.com
4	Alice	Williams	alice.williams@email.com
5	Charlie	Brown	charlie.brown@email.com
6	Eva	Davis	eva.davis@email.com
7	Frank	Miller	frank.miller@email.com
8	Grace	Jones	grace.jones@email.com
9	Henry	Garcia	henry.garcia@email.com
10	Ivy	Clark	ivy.clark@email.com

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  
JOIN Customers c ON o.CustomerID = c.CustomerID;
```

200 %

Results Messages

	OrderID	OrderDate	FirstName	LastName
1	1	2023-01-01	John	Doe
2	2	2023-01-02	Jane	Smith
3	3	2023-01-03	Bob	Johnson
4	4	2023-01-04	Alice	Williams
5	5	2023-01-05	Charlie	Brown
6	6	2023-01-06	Eva	Davis
7	7	2023-01-07	Frank	Miller
8	8	2023-01-08	Grace	Jones
9	9	2023-01-09	Henry	Garcia
10	10	2023-01-10	Ivy	Clark

### 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 (11, 'Damon', 'Salvatore', 'damon.salvatore@email.com', '987-654-3210', '456 Oak St');
select * from Customers;
```

100 %

Results Messages

	CustomerID	FirstName	LastName	Email	Phone	Address
1	1	John	Doe	daniel.alexander@email.com	1234567890	789 Pine St
2	2	Jane	Smith	jane.smith@email.com	9876543210	456 Oak St
3	3	Bob	Johnson	bob.johnson@email.com	5551234567	789 Pine St
4	4	Alice	Williams	alice.williams@email.com	1112223333	456 Elm St
5	5	Charlie	Brown	charlie.brown@email.com	4445556666	789 Oak St
6	6	Eva	Davis	eva.davis@email.com	7778889999	123 Maple St
7	7	Frank	Miller	frank.miller@email.com	2223334444	456 Birch St
8	8	Grace	Jones	grace.jones@email.com	9998887777	789 Cedar St
9	9	Henry	Garcia	henry.garcia@email.com	3334445555	123 Walnut St
10	10	Ivy	Clark	ivy.clark@email.com	6667778888	456 Pine St
11	11	Damon	Salvatore	damon.salvatore@email.com	987-654-3210	456 Oak St

### 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 ProductName = 'Laptop' OR
ProductName = 'Smartphone' or
ProductName = 'Smartwatch' or
ProductName = 'Bluetooth Speaker';
select * from Products;
```

100 %

Results Messages

	ProductID	ProductName	Description	Price
1	1	Laptop	Powerful laptop for professional use	109998
2	2	Smartphone	High-performance smartphone	76998
3	3	Water Bottle	To stored water	299
4	4	Smartwatch	Fitness tracker and smartwatch combo	16498
5	5	Washing Machine	Washes the clothes	12999
6	6	Clocks	Tells the time	599
7	7	Table Desk	For Doing Works	799
8	8	Iron Box	To Iron the CLOthes	499
9	9	Mosquito Bat	To Kill Mosquitoes	299
10	10	Bluetooth Speaker	Portable Bluetooth speaker with rich sound	5498



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 = 6;  
DELETE FROM Orders WHERE OrderID = 6;
```

00 %  
Messages

(1 row affected)

Completion time: 2023-12-08T10:14:26.6429404+05:30

```
DELETE FROM Orders WHERE OrderID = 6;  
select * from Orders;  
select * from OrderDetails;
```

200 %

Results Messages

	OrderID	CustomerID	OrderDate	TotalAMount
1	2	2	2023-01-02	299
2	3	3	2023-01-03	38997
3	4	4	2023-01-04	4794
4	5	5	2023-01-05	153996
5	7	7	2023-01-07	32988
6	8	8	2023-01-08	32996
7	9	9	2023-01-09	499
8	10	10	2023-01-10	897
9	11	11	2023-02-01	NULL

```
select * from OrderDetails;
```

200 %

Results

Messages

	OrderDetailId	OrderId	ProductID	Quantity
1	2	2	3	1
2	3	3	5	3
3	4	4	7	6
4	5	5	2	2
5	7	7	10	6
6	8	8	4	2
7	9	9	8	1
8	10	10	9	3

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 (11,11, '2023-02-01', 1499.99);  
select * from Orders;
```

200 %

Results

Messages

	OrderID	CustomerID	OrderDate	TotalAMount
1	1	1	2023-01-01	9999
2	2	2	2023-01-02	69999
3	3	3	2023-01-03	29999
4	4	4	2023-01-04	14999
5	5	5	2023-01-05	2599
6	7	7	2023-01-07	7999
7	8	8	2023-01-08	9999
8	9	9	2023-01-09	8999
9	10	10	2023-01-10	14999
10	11	11	2023-02-01	1499

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 = 'daniel.alexander@email.com', Address = '789 Pine St'  
WHERE CustomerID = 1;  
select * from Customers;
```

Results		Messages				
CustomerID	FirstName	LastName	Email	Phone	Address	
1	John	Doe	daniel.alexander@email.com	1234567890	789 Pine St	
2	Jane	Smith	jane.smith@email.com	9876543210	456 Oak St	
3	Bob	Johnson	bob.johnson@email.com	5551234567	789 Pine St	
4	Alice	Williams	alice.williams@email.com	1112223333	456 Elm St	
5	Charlie	Brown	charlie.brown@email.com	4445556666	789 Oak St	
6	Eva	Davis	eva.davis@email.com	7778889999	123 Maple St	
7	Frank	Miller	frank.miller@email.com	2223334444	456 Birch St	
8	Grace	Jones	grace.jones@email.com	9998887777	789 Cedar St	
9	Henry	Garcia	henry.garcia@email.com	3334445555	123 Walnut St	
10	Ivy	Clark	ivy.clark@email.com	6667778888	456 Pine St	
11	Damon	Salvatore	damon.salvatore@email.com	987-654-3210	456 Oak St	

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(Quantity * Price)  
    FROM OrderDetails od  
    JOIN Products p ON od.ProductID = p.ProductID  
    WHERE od.OrderID = Orders.OrderID);  
select * from Orders;
```

Results		Messages			
OrderID	CustomerID	OrderDate	TotalAmount		
2	2	2023-01-02	299		
3	3	2023-01-03	38997		
4	4	2023-01-04	4794		
5	5	2023-01-05	153996		
7	7	2023-01-07	32988		
8	8	2023-01-08	32996		
9	9	2023-01-09	499		
10	10	2023-01-10	897		
11	11	2023-02-01	NULL		

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 OrderDetails WHERE OrderID IN
(SELECT OrderID FROM Orders WHERE CustomerID = 1);
DELETE FROM Orders WHERE CustomerID = 1;
select * from Orders;
select * from OrderDetails;

```

200 %

Results Messages

	OrderID	CustomerID	OrderDate	TotalAmount
1	2	2	2023-01-02	299
2	3	3	2023-01-03	38997
3	4	4	2023-01-04	4794
4	5	5	2023-01-05	153996
5	7	7	2023-01-07	32988
6	8	8	2023-01-08	32996
7	9	9	2023-01-09	499
8	10	10	2023-01-10	897
9	11	11	2023-02-01	NULL

```

DELETE FROM OrderDetails WHERE OrderID IN
(SELECT OrderID FROM Orders WHERE CustomerID = 1);
DELETE FROM Orders WHERE CustomerID = 1;
select * from Orders;
select * from OrderDetails;

```

200 %

Results Messages

	OrderDetailId	OrderID	ProductID	Quantity
1	2	2	3	1
2	3	3	5	3
3	4	4	7	6
4	5	5	2	2
5	7	7	10	6
6	8	8	4	2
7	9	9	8	1
8	10	10	9	3

**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.**

```

ALTER table products
add category varchar(255);
INSERT INTO Products (ProductID, ProductName, Description, Price, category)
VALUES (11, 'Tablet', 'High-performance Device', 69999.99, 'Electronic Gadgets');
select * from Products;

```

200 %

Results Messages

	ProductID	ProductName	Description	Price	category
1	1	Laptop	Powerful laptop for professional use	109998	NULL
2	2	Smartphone	High-performance smartphone	76998	NULL
3	3	Water Bottle	To stored water	299	NULL
4	4	Smartwatch	Fitness tracker and smartwatch combo	16498	NULL
5	5	Washing Machine	Washes the clothes	12999	NULL
6	6	Clocks	Tells the time	599	NULL
7	7	Table Desk	For Doing Works	799	NULL
8	8	Iron Box	To Iron the CLothes	499	NULL
9	9	Mosquito Bat	To Kill Mosquitoes	299	NULL
10	10	Bluetooth Speaker	Portable Bluetooth speaker with rich sound	5498	NULL
11	11	Tablet	High-performance Device	69999	Electronic Gadgets

**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.**

```

Alter table orders
add status varchar(255);
UPDATE Orders SET Status = 'Shipped' WHERE OrderID = 2;
select * from Orders;

```

200 %

Results Messages

	OrderID	CustomerID	OrderDate	TotalAMount	status
1	2	2	2023-01-02	299	Shipped
2	3	3	2023-01-03	38997	NULL
3	4	4	2023-01-04	4794	NULL
4	5	5	2023-01-05	153996	NULL
5	7	7	2023-01-07	32988	NULL
6	8	8	2023-01-08	32996	NULL
7	9	9	2023-01-09	499	NULL
8	10	10	2023-01-10	897	NULL
9	11	11	2023-02-01	NULL	NULL

```

Alter table orders
add status varchar(255);
UPDATE Orders
SET Status = 'Shipped'
WHERE OrderID = 2;
UPDATE Orders
Set status = 'Pending'
where OrderID = 3;
UPDATE Orders
Set status = 'Shipped'
where OrderID = 4;
UPDATE Orders
Set status = 'Pending'
where OrderID = 5;
UPDATE Orders
Set status = 'Pending'
where OrderID = 7;
UPDATE Orders
Set status = 'Shipped'
where OrderID = 8;
UPDATE Orders
Set status = 'Shipped'
where OrderID = 9;
UPDATE Orders
Set status = 'Pending'
where OrderID = 10;
UPDATE Orders
Set status = 'Shipped'
where OrderID = 11;

```

```
select * from Orders;
```

200 %

Results Messages

	OrderID	CustomerID	OrderDate	TotalAmount	status
1	2	2	2023-01-02	299	Shipped
2	3	3	2023-01-03	38997	Pending
3	4	4	2023-01-04	4794	Shipped
4	5	5	2023-01-05	153996	Pending
5	7	7	2023-01-07	32988	Pending
6	8	8	2023-01-08	32996	Shipped
7	9	9	2023-01-09	499	Shipped
8	10	10	2023-01-10	897	Pending
9	11	11	2023-02-01	NULL	Shipped

```
UPDATE Orders
Set status = 'Shipped'
where OrderID = 7;
select * from Orders;
```

200 %

Results Messages

	OrderID	CustomerID	OrderDate	TotalAmount	status
1	2	2	2023-01-02	299	Shipped
2	3	3	2023-01-03	38997	Pending
3	4	4	2023-01-04	4794	Shipped
4	5	5	2023-01-05	153996	Pending
5	7	7	2023-01-07	32988	Shipped
6	8	8	2023-01-08	32996	Shipped
7	9	9	2023-01-09	499	Shipped
8	10	10	2023-01-10	897	Pending
9	11	11	2023-02-01	NULL	Shipped

Query executed successfully

### Task 3. Aggregate functions, Having, Order By, Group by 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 Orders.OrderID, Customers.FirstName, Customers.LastName, Customers.Email, Customers.Phone, Customers.Address
FROM Orders
JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
```

125 %

Results Messages

	OrderID	FirstName	LastName	Email	Phone	Address
1	2	Jane	Smith	jane.smith@email.com	9876543210	456 Oak St
2	3	Bob	Johnson	bob.johnson@email.com	5551234567	789 Pine St
3	4	Alice	Williams	alice.williams@email.com	1112223333	456 Elm St
4	5	Charlie	Brown	charlie.brown@email.com	4445556666	789 Oak St
5	7	Frank	Miller	frank.miller@email.com	2223334444	456 Birch St
6	8	Grace	Jones	grace.jones@email.com	9998887777	789 Cedar St
7	9	Henry	Garcia	henry.garcia@email.com	3334445555	123 Walnut St
8	10	Ivy	Clark	ivy.clark@email.com	6667778888	456 Pine St
9	11	Damon	Salvatore	damon.salvatore@email.com	987-654-3210	456 Oak St

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 Products.ProductName, SUM(Orders.TotalAmount) AS TotalRevenue
FROM Products
JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID
JOIN Orders ON OrderDetails.OrderID = Orders.OrderID
GROUP BY Products.ProductName;

```

200 %

Results Messages

	ProductName	TotalRevenue
1	Bluetooth Speaker	32988
2	Iron Box	499
3	Mosquito Bat	897
4	Smartphone	153996
5	Smartwatch	32996
6	Table Desk	4794
7	Washing Machine	38997
8	Water Bottle	299

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

```

SELECT DISTINCT Customers.FirstName, Customers.LastName, Customers.Email, Customers.Phone, Customers.Address
FROM Customers
JOIN Orders ON Customers.CustomerID = Orders.CustomerID;

```

150 %

Results Messages

	FirstName	LastName	Email	Phone	Address
1	Alice	Williams	alice.williams@email.com	1112223333	456 Elm St
2	Bob	Johnson	bob.johnson@email.com	5551234567	789 Pine St
3	Charlie	Brown	charlie.brown@email.com	4445556666	789 Oak St
4	Damon	Salvatore	damon.salvatore@email.com	987-654-3210	456 Oak St
5	Frank	Miller	frank.miller@email.com	2223334444	456 Birch St
6	Grace	Jones	grace.jones@email.com	9998887777	789 Cedar St
7	Henry	Garcia	henry.garcia@email.com	3334445555	123 Walnut St
8	Ivy	Clark	ivy.clark@email.com	6667778888	456 Pine St
9	Jane	Smith	jane.smith@email.com	9876543210	456 Oak St

**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 Products.ProductName, SUM(OrderDetails.Quantity) AS TotalQuantityOrdered
FROM Products
JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID
GROUP BY Products.ProductName
ORDER BY TotalQuantityOrdered DESC;

```

190 %

Results		Messages	
	ProductName	TotalQuantityOrdered	
1	Bluetooth Speaker	6	

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

```

select productname, category from Products;

```

300 %

Results		Messages	
	productname	category	
1	Laptop	NULL	
2	Smartphone	NULL	
3	Water Bottle	NULL	
4	Smartwatch	NULL	
5	Washing Machine	NULL	
6	Clocks	NULL	
7	Table Desk	NULL	
8	Iron Box	NULL	
9	Mosquito Bat	NULL	
10	Bluetooth Speaker	NULL	
11	Tablet	Electronic Gadgets	

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

```

150 %

	FirstName	LastName	AverageOrderValue
1	Charlie	Brown	153996
2	Ivy	Clark	897
3	Henry	Garcia	499
4	Bob	Johnson	38997
5	Grace	Jones	32996
6	Frank	Miller	32988
7	Damon	Salvatore	NULL
8	Jane	Smith	299
9	Alice	Williams	4794

**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 Orders.OrderID, Customers.FirstName, Customers.LastName, SUM(Orders.TotalAmount) AS TotalRevenue
FROM Orders
JOIN Customers ON Orders.CustomerID = Customers.CustomerID
GROUP BY Orders.OrderID, Customers.FirstName, Customers.LastName
ORDER BY TotalRevenue DESC;

```

150 %

	OrderID	FirstName	LastName	TotalRevenue
1	5	Charlie	Brown	153996

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

```

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

```

200 %

	ProductName	OrderCount
1	Bluetooth Speaker	1
2	Iron Box	1
3	Mosquito Bat	1
4	Smartphone	1
5	Smartwatch	1
6	Table Desk	1
7	Washing Machine	1
8	Water Bottle	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.**

```
SELECT Customers.FirstName, Customers.LastName, Customers.Email, Customers.Phone, Customers.Address
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 = 'Smartphone';
```

150 %

Results Messages

	FirstName	LastName	Email	Phone	Address
1	Charlie	Brown	charlie.brown@email.com	4445556666	789 Oak St

**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 '2023-01-02' AND '2023-01-07';
```

200 %

Results Messages

	TotalRevenue
1	231074

## Task 4. Subquery and its type:

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

```
SELECT Customers.FirstName, Customers.LastName  
FROM Customers  
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID  
WHERE Orders.OrderID IS NULL;
```

200 %

Results Messages

	FirstName	LastName
1	John	Doe
2	Eva	Davis

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

```
SELECT COUNT(ProductID) AS TotalProducts  
FROM Products;
```

200 %

Results Messages

	TotalProducts
1	11

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

```
SELECT SUM(Orders.TotalAmount) AS TotalRevenue
FROM Orders
JOIN Customers ON Orders.CustomerID = Customers.CustomerID
WHERE Customers.FirstName = 'Ivy';
```

200 %

Results Messages

TotalRevenue	
1	897

**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 AVG(OrderDetails.Quantity) AS AvgQuantityOrdered
FROM OrderDetails
JOIN Products ON OrderDetails.ProductID = Products.ProductID
WHERE Products.ProductName = 'Smartphone';
```

200 %

Results Messages

AvgQuantityOrdered	
1	2

**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 SUM(TotalAmount) as TotalRevenue from Orders
where CustomerID = 7;
```

200 %

Results Messages

TotalRevenue	
1	32988

**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 TOP 1 Customers.FirstName, Customers.LastName, COUNT(Orders.OrderID) AS OrderCount
FROM Customers
JOIN Orders ON Customers.CustomerID = Orders.CustomerID
GROUP BY Customers.FirstName, Customers.LastName
ORDER BY OrderCount DESC;
```

	FirstName	LastName	OrderCount
1	Ivy	Clark	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.**

```
SELECT TOP 1 Products.ProductName, SUM(OrderDetails.Quantity) AS TotalQuantityOrdered
FROM Products
JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID
GROUP BY Products.ProductName
ORDER BY TotalQuantityOrdered DESC;
```

	ProductName	TotalQuantityOrdered
1	Bluetooth Speaker	6

**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 Customers.FirstName, Customers.LastName, SUM(Orders.TotalAmount) AS TotalSpending
FROM Customers
JOIN Orders ON Customers.CustomerID = Orders.CustomerID
JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID
JOIN Products ON OrderDetails.ProductID = Products.ProductID
WHERE Products.ProductName = 'Smartphone'
GROUP BY Customers.FirstName, Customers.LastName
ORDER BY TotalSpending DESC;
```

	FirstName	LastName	TotalSpending
1	Charlie	Brown	153996

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

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

150 %

Results Messages

	FirstName	LastName	AvgOrderValue
1	Charlie	Brown	153996
2	Ivy	Clark	897
3	Henry	Garcia	499
4	Bob	Johnson	38997
5	Grace	Jones	32996
6	Frank	Miller	32988
7	Damon	Salvatore	NULL
8	Jane	Smith	299
9	Alice	Williams	4794

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

150 %

Results Messages

	FirstName	LastName	OrderCount
1	Charlie	Brown	1
2	Ivy	Clark	1
3	Eva	Davis	0
4	John	Doe	0
5	Henry	Garcia	1
6	Bob	Johnson	1
7	Grace	Jones	1
8	Frank	Miller	1
9	Damon	Salvatore	1
10	Jane	Smith	1
11	Alice	Williams	1