

# TECHSHOP ASSIGNMENT DOCUMENTATION

## 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

## Task 1- Database Design:

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

```
techshopquery.sql...ANTHINI\nishash (53)  ↗ X [REDACTED]
DROP DATABASE IF EXISTS TechShop;
CREATE DATABASE TechShop;
USE TechShop;

100 %  ↵
Messages
Commands completed successfully.

Completion time: 2025-03-31T15:15:08.6190419+05:30
```

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

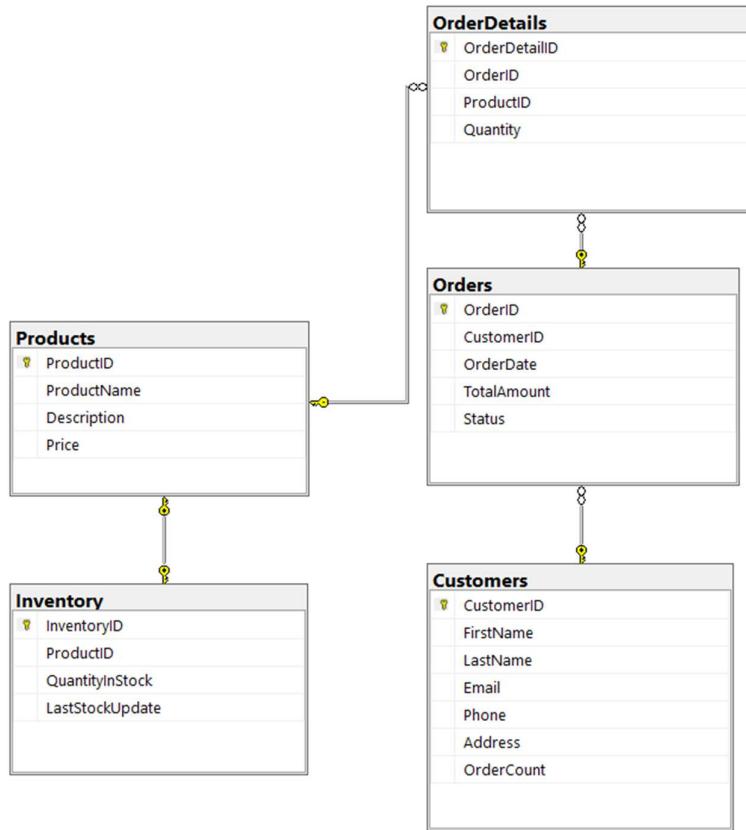
```
techshopquery.sql...ANTHINI\nishash (53)*  ↗ X [REDACTED]
-- Create Customers table
CREATE TABLE Customers (
    CustomerID INT IDENTITY(1,1) PRIMARY KEY,
    FirstName NVARCHAR(50) NOT NULL,
    LastName NVARCHAR(50) NOT NULL,
    Email NVARCHAR(100) UNIQUE NOT NULL,
    Phone NVARCHAR(15) UNIQUE NOT NULL,
    Address NVARCHAR(255) NOT NULL
);
GO
```

```
techshopquery.sql...ANTHINI\nisha (53)*  ↗ X
-- Create Products table
CREATE TABLE Products (
    ProductID INT IDENTITY(1,1) PRIMARY KEY,
    ProductName NVARCHAR(100) NOT NULL,
    Description NVARCHAR(255),
    Price DECIMAL(10,2) NOT NULL CHECK (Price > 0)
);
GO
-- Create Orders table
CREATE TABLE Orders (
    OrderID INT IDENTITY(1,1) PRIMARY KEY,
    CustomerID INT NOT NULL,
    OrderDate DATETIME DEFAULT GETDATE(),
    TotalAmount DECIMAL(10,2) DEFAULT 0,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID) ON DELETE CASCADE
);
GO
-- Create OrderDetails table
CREATE TABLE OrderDetails (
    OrderDetailID INT IDENTITY(1,1) PRIMARY KEY,
    OrderID INT NOT NULL,
    ProductID INT NOT NULL,
    Quantity INT NOT NULL CHECK (Quantity > 0),
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) ON DELETE CASCADE,
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID) ON DELETE CASCADE
);
GO
-- Create Inventory table
CREATE TABLE Inventory (
    InventoryID INT IDENTITY(1,1) PRIMARY KEY,
    ProductID INT NOT NULL UNIQUE,
    QuantityInStock INT NOT NULL CHECK (QuantityInStock >= 0),
    LastStockUpdate DATETIME DEFAULT GETDATE(),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID) ON DELETE CASCADE
);
100 %  ▾
Messages
Commands completed successfully.

Completion time: 2025-03-31T15:15:08.6190419+05:30
```

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

Nishanthini\NISHA...hShop - Diagram\_0\* techshopquery.sql...ANTHINI\nisha (53)\*



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

This question is already done in qn:2

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

- Customers
- Products
- Orders
- OrderDetails
- Inventory

```

techshopquery.sql...ANTHINI\nishash (53)* ↵ X
-- Insert sample Customers
INSERT INTO Customers (FirstName, LastName, Email, Phone, Address) VALUES
('Rohith', 'Kumar', 'rohit.kumar@email.com', '9876543210', 'Hyderabad, Telangana'),
('Sandeep', 'Reddy', 'sandeep.reddy@email.com', '9876543211', 'Bangalore, Karnataka'),
('Gani', 'Sheikh', 'gani.sheikh@email.com', '9876543212', 'Chennai, Tamil Nadu'),
('Nikila', 'M', 'nikila.m@email.com', '9876543213', 'Mumbai, Maharashtra'),
('Vicky', 'Raj', 'vicky.raj@email.com', '9876543214', 'Pune, Maharashtra'),
('Nivedha', 'S', 'nivedha.s@email.com', '9876543215', 'Coimbatore, Tamil Nadu'),
('Yogesh', 'Naidu', 'yogesh.naidu@email.com', '9876543216', 'Delhi'),
('Harini', 'Rao', 'harini.rao@email.com', '9876543217', 'Kolkata, West Bengal'),
('Harish', 'Patel', 'harish.patel@email.com', '9876543218', 'Ahmedabad, Gujarat'),
('Prabha', 'Devi', 'prabha.devi@email.com', '9876543219', 'Jaipur, Rajasthan');

GO
-- Insert sample Products
INSERT INTO Products (ProductName, Description, Price) VALUES
('Smartphone', 'Latest Android smartphone', 19999.00),
('Laptop', 'High-performance laptop', 59999.00),
('Smartwatch', 'Fitness and health tracking smartwatch', 4999.00),
('Headphones', 'Noise-cancelling over-ear headphones', 2999.00),
('Tablet', '10-inch Android tablet', 14999.00),
('Bluetooth Speaker', 'Portable wireless speaker', 1999.00),
('Gaming Console', 'Next-gen gaming console', 39999.00),
('Power Bank', '10000mAh power bank', 999.00),
('Wireless Mouse', 'Ergonomic wireless mouse', 799.00),
('Keyboard', 'Mechanical gaming keyboard', 2499.00);

GO

```

```

techshopquery.sql...ANTHINI\nishash (53)* ↵ X
-- Insert sample Orders
INSERT INTO Orders (CustomerID, OrderDate, TotalAmount) VALUES
(1, '2025-03-20', 22998.00),
(2, '2025-03-21', 999.00),
(3, '2025-03-19', 19999.00),
(4, '2025-03-18', 8498.00),
(5, '2025-03-22', 59999.00),
(6, '2025-03-17', 4999.00),
(7, '2025-03-16', 14999.00),
(8, '2025-03-23', 2999.00),
(9, '2025-03-15', 39999.00),
(10, '2025-03-14', 799.00);

GO
-- Insert sample OrderDetails
INSERT INTO OrderDetails (OrderID, ProductID, Quantity) VALUES
(1, 1, 1), -- Rohith bought a Smartphone
(2, 8, 1), -- Sandeep bought a Power Bank
(3, 2, 1), -- Gani bought a Laptop
(4, 4, 2), -- Nikila bought 2 Headphones
(5, 3, 1), -- Vicky bought a Smartwatch
(6, 5, 1), -- Nivedha bought a Tablet
(7, 6, 1), -- Yogesh bought a Bluetooth Speaker
(8, 7, 1), -- Harini bought a Gaming Console
(9, 9, 1), -- Harish bought a Wireless Mouse
(10, 10, 1); -- Prabha bought a Keyboard
GO

-- Insert sample Inventory Data
INSERT INTO Inventory (ProductID, QuantityInStock, LastStockUpdate) VALUES
(1, 50, '2025-03-10'),
(2, 30, '2025-03-11'),
(3, 20, '2025-03-12'),
(4, 100, '2025-03-13'),
(5, 25, '2025-03-14'),
(6, 40, '2025-03-15'),
(7, 15, '2025-03-16'),
(8, 80, '2025-03-17'),
(9, 60, '2025-03-18'),
(10, 35, '2025-03-19');

GO

```

```

(10 rows affected)

Completion time: 2025-03-23T07:52:18.4704820+05:30

100 % ▾
Query executed successfully. NISHANTHINI\NISHANTHINI (16... NISHANTHINI\Nisha (57) TechShop 00:00:00 0 rows
techshopquery.sql...ANTHINI\nish (53)* ↗ X

-- Display all Customers
SELECT * FROM Customers;

-- Display all Products
SELECT * FROM Products;

-- Display all Orders
SELECT * FROM Orders;

-- Display all OrderDetails
SELECT * FROM OrderDetails;

-- Display all Inventory
SELECT * FROM Inventory;

```

	CustomerID	FirstName	LastName	Email	Phone	Address
1	1	Rohith	Kumar	rohit.kumar@email.com	9876543210	Hyderabad, Telangana
2	2	Sandeep	Reddy	sandeep.reddy@email.com	9876543211	Bangalore, Karnataka
3	3	Gani	Sheikh	gani.sheikh@email.com	9876543212	Chennai, Tamil Nadu
4	4	Nikila	M	nikila.m@email.com	9876543213	Mumbai, Maharashtra
5	5	Vicky	Raj	vicky.raj@email.com	9876543214	Pune, Maharashtra
6	6	Nivedha	S	nivedha.s@email.com	9876543215	Coimbatore, Tamil Nadu
7	7	Yogesh	Naidu	yogesh.naidu@email.com	9876543216	Delhi
8	8	Harini	Rao	harini.rao@email.com	9876543217	Kolkata, West Bengal
9	9	Harish	Patel	harish.patel@email.com	9876543218	Ahmedabad, Gujarat
10	10	Prabha	Devi	prabha.devi@email.com	9876543219	Jaipur, Rajasthan

100 % ▾

Results Messages

	ProductID	ProductName	Description	Price
1	1	Smartphone	Latest Android smartphone	19999.00
2	2	Laptop	High-performance laptop	59999.00
3	3	Smartwatch	Fitness and health tracking smartwatch	4999.00
4	4	Headphones	Noise-cancelling over-ear headphones	2999.00
5	5	Tablet	10-inch Android tablet	14999.00
6	6	Bluetooth Speaker	Portable wireless speaker	1999.00
7	7	Gaming Console	Next-gen gaming console	39999.00
8	8	Power Bank	10000mAh power bank	999.00
9	9	Wireless Mouse	Ergonomic wireless mouse	799.00
10	10	Keyboard	Mechanical gaming keyboard	2499.00

100 %

Results Messages

	OrderID	CustomerID	OrderDate	TotalAmount
1	1	1	2025-03-20 00:00:00.000	22998.00
2	2	2	2025-03-21 00:00:00.000	999.00
3	3	3	2025-03-19 00:00:00.000	19999.00
4	4	4	2025-03-18 00:00:00.000	8498.00
5	5	5	2025-03-22 00:00:00.000	59999.00
6	6	6	2025-03-17 00:00:00.000	4999.00
7	7	7	2025-03-16 00:00:00.000	14999.00
8	8	8	2025-03-23 00:00:00.000	2999.00
9	9	9	2025-03-15 00:00:00.000	39999.00
10	10	10	2025-03-14 00:00:00.000	799.00

✓ Query executed successfully.

100 %

Results Messages

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

✓ Query executed successfully.

100 %

Results Messages

	InventoryID	ProductID	QuantityInStock	LastStockUpdate
1	1	1	50	2025-03-10 00:00:00.000
2	2	2	30	2025-03-11 00:00:00.000
3	3	3	20	2025-03-12 00:00:00.000
4	4	4	100	2025-03-13 00:00:00.000
5	5	5	25	2025-03-14 00:00:00.000
6	6	6	40	2025-03-15 00:00:00.000
7	7	7	15	2025-03-16 00:00:00.000
8	8	8	80	2025-03-17 00:00:00.000
9	9	9	60	2025-03-18 00:00:00.000
10	10	10	35	2025-03-19 00:00:00.000

✓ Query executed successfully.

## Task 2- Select, Where, Between, AND, LIKE:

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

```
techshopquery.sql...ANTHINI\nisha (53)* ↵ X
-- 1 Retrieve the names and emails of all customers
PRINT 'Displaying all customer names and emails:';
SELECT FirstName, LastName, Email FROM Customers;
GO
```

100 %

Results Messages

	FirstName	LastName	Email
1	Rohith	Kumar	rohit.kumar@email.com
2	Sandeep	Reddy	sandeep.reddy@email.com
3	Gani	Sheikh	gani.sheikh@email.com
4	Nikila	M	nikila.m@email.com
5	Vicky	Raj	vicky.raj@email.com
6	Nivedha	S	nivedha.s@email.com
7	Yogesh	Naidu	yogesh.naidu@email.com
8	Harini	Rao	harini.rao@email.com
9	Harish	Patel	harish.patel@email.com
10	Prabha	Devi	prabha.devi@email.com

Query executed successfully.

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

```
techshopquery.sql...ANTHINI\nisha (53)* ↵ X
-- 2 List all orders with their order dates and corresponding customer names
PRINT 'Listing all orders with customer names:';
SELECT O.OrderID, O.OrderDate, C.FirstName, C.LastName
FROM Orders O
JOIN Customers C ON O.CustomerID = C.CustomerID;
GO
```

100 %

Results Messages

	OrderID	OrderDate	FirstName	LastName
1	1	2025-03-20 00:00:00.000	Rohith	Kumar
2	2	2025-03-21 00:00:00.000	Sandeep	Reddy
3	3	2025-03-19 00:00:00.000	Gani	Sheikh
4	4	2025-03-18 00:00:00.000	Nikila	M
5	5	2025-03-22 00:00:00.000	Vicky	Raj
6	6	2025-03-17 00:00:00.000	Nivedha	S
7	7	2025-03-16 00:00:00.000	Yogesh	Naidu
8	8	2025-03-23 00:00:00.000	Harini	Rao
9	9	2025-03-15 00:00:00.000	Harish	Patel
10	10	2025-03-14 00:00:00.000	Prabha	Devi

Query executed successfully.

3. Write an SQL query to insert a new customer record into the "Customers" table. Include customer information such as name, email, and address.

The screenshot shows an SQL query being run in SSMS. The query inserts a new customer record into the 'Customers' table and then displays the updated list of customers. The results grid shows 11 rows of customer data, including the newly inserted row at the bottom.

```
-- 3 Insert a new customer record
PRINT 'Inserting a new customer...'
INSERT INTO Customers (FirstName, LastName, Email, Phone, Address)
VALUES ('Raj', 'Sharma', 'raj.sharma@email.com', '9876543220', 'Chennai, Tamil Nadu');

-- Verify the insert
PRINT 'New customer inserted. Displaying updated customer list:';
SELECT * FROM Customers;
GO
```

	CustomerID	FirstName	LastName	Email	Phone	Address
1	1	Rohith	Kumar	rohit.kumar@email.com	9876543210	Hyderabad, Telangana
2	2	Sandeep	Reddy	sandeep.reddy@email.com	9876543211	Bangalore, Karnataka
3	3	Gani	Sheikh	gani.sheikh@email.com	9876543212	Chennai, Tamil Nadu
4	4	Nikila	M	nikila.m@email.com	9876543213	Mumbai, Maharashtra
5	5	Vicky	Raj	vicky.raj@email.com	9876543214	Pune, Maharashtra
6	6	Nivedha	S	nivedha.s@email.com	9876543215	Coimbatore, Tamil Nadu
7	7	Yogesh	Naidu	yogesh.naidu@email.com	9876543216	Delhi
8	8	Harini	Rao	harini.rao@email.com	9876543217	Kolkata, West Bengal
9	9	Harish	Patel	harish.patel@email.com	9876543218	Ahmedabad, Gujarat
10	10	Prabha	Devi	prabha.devi@email.com	9876543219	Jaipur, Rajasthan
11	11	Raj	Sharma	raj.sharma@email.com	9876543220	Chennai, Tamil Nadu

Query executed successfully.

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

The screenshot shows an SQL query that updates the 'Price' column in the 'Products' table by multiplying it by 1.10. It also includes a verification step to display the updated product list.

```
-- 4 Update prices of all electronic gadgets by increasing them by 10%
PRINT 'Updating product prices by 10%...';
UPDATE Products
SET Price = Price * 1.10;

-- Verify price update
PRINT 'Prices updated successfully. Displaying updated products list:';
SELECT * FROM Products;
GO
```

100 %

Results Messages

ProductID	ProductName	Description	Price
1	Smartphone	Latest Android smartphone	21998.90
2	Laptop	High-performance laptop	65998.90
3	Smartwatch	Fitness and health tracking smartwatch	5498.90
4	Headphones	Noise-cancelling over-ear headphones	3298.90
5	Tablet	10-inch Android tablet	16498.90
6	Bluetooth Speaker	Portable wireless speaker	2198.90
7	Gaming Console	Next-gen gaming console	43998.90
8	Power Bank	10000mAh power bank	1098.90
9	Wireless Mouse	Ergonomic wireless mouse	878.90
10	Keyboard	Mechanical gaming keyboard	2748.90

Query executed successfully.

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.

```
techshopquery.sql...ANTHINI\nishaa (53)*  ↗ X
-- 5 Delete a specific order and its associated order details (Using Stored Procedure)
PRINT 'Creating stored procedure for deleting an order...';
GO
CREATE OR ALTER PROCEDURE DeleteOrderAndDetails
    @OrderID INT
AS
BEGIN
    -- Check if the OrderID exists before attempting to delete
    IF NOT EXISTS (SELECT 1 FROM Orders WHERE OrderID = @OrderID)
    BEGIN
        PRINT 'Error: Order ID does not exist.';
        RETURN;
    END

    -- Step 1: Delete associated order details first
    DELETE FROM OrderDetails WHERE OrderID = @OrderID;

    -- Step 2: Delete the order
    DELETE FROM Orders WHERE OrderID = @OrderID;

    PRINT 'Order and associated details deleted successfully.';
END;
GO

-- EXECUTE: Run this command to delete an order
EXEC DeleteOrderAndDetails @OrderID = 3;
GO
SELECT * FROM OrderDetails;
```

100 %

Messages

Creating stored procedure for deleting an order...

(1 row affected)

(1 row affected)

Order and associated details deleted successfully.

Completion time: 2025-03-23T10:26:12.1339634+05:30

100 %

Query executed successfully.

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.

The screenshot shows a SQL query window titled 'techshopquery.sql...ANTHINI\nisha (53)\*'. The query inserts a new row into the 'Orders' table with CustomerID 2, OrderDate as GETDATE(), and TotalAmount 0. It then selects all rows from the 'Orders' table. The results grid shows 11 rows of order data. A message at the bottom indicates the query was executed successfully.

```
-- 6 Insert a new order into the "Orders" table
PRINT 'Inserting a new order...';
INSERT INTO Orders (CustomerID, OrderDate, TotalAmount)
VALUES (2, GETDATE(), 0);

-- Verify the insert
PRINT 'New order inserted. Displaying updated orders list:';
SELECT * FROM Orders;
GO
```

OrderID	CustomerID	OrderDate	TotalAmount
1	1	2025-03-20 00:00:00.000	22998.00
2	2	2025-03-21 00:00:00.000	999.00
3	4	2025-03-18 00:00:00.000	8498.00
4	5	2025-03-22 00:00:00.000	59999.00
5	6	2025-03-17 00:00:00.000	4999.00
6	7	2025-03-16 00:00:00.000	14999.00
7	8	2025-03-23 00:00:00.000	2999.00
8	9	2025-03-15 00:00:00.000	39999.00
9	10	2025-03-14 00:00:00.000	799.00
10	11	2025-03-23 10:27:52.230	0.00

Query executed successfully.

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.

The screenshot shows the creation of a stored procedure named 'UpdateCustomerInfo'. The procedure takes three parameters: @CustomerID (INT), @NewEmail (NVARCHAR(100)), and @NewAddress (NVARCHAR(255)). It first checks if the customer exists. If not, it prints an error and returns. Otherwise, it updates the 'Customers' table with the new email and address where CustomerID matches the input parameter. Finally, it prints a success message. An EXECUTE command is shown at the bottom to run the procedure with sample data.

```
-- 7 Update the contact information of a specific customer (Using Stored Procedure)
PRINT 'Creating stored procedure for updating customer contact info...';
GO
CREATE OR ALTER PROCEDURE UpdateCustomerInfo
    @CustomerID INT,
    @NewEmail NVARCHAR(100),
    @NewAddress NVARCHAR(255)
AS
BEGIN
    -- Check if Customer exists before updating
    IF NOT EXISTS (SELECT 1 FROM Customers WHERE CustomerID = @CustomerID)
    BEGIN
        PRINT 'Error: Customer ID does not exist.';
        RETURN;
    END

    UPDATE Customers
    SET Email = @NewEmail, Address = @NewAddress
    WHERE CustomerID = @CustomerID;

    PRINT 'Customer information updated successfully.';
END;
GO

-- EXECUTE: Run this command to update a customer's contact details
EXEC UpdateCustomerInfo @CustomerID = 5, @NewEmail = 'stjoseph@email.com', @NewAddress = 'Chennai, TamilNadu';
GO
```

```
100 % 
Messages
Creating stored procedure for updating customer contact info...
(1 row affected)
Customer information updated successfully.

Completion time: 2025-03-23T10:29:26.7448570+05:30
```

```
100 % 
Query executed successfully.
```

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.

```
techshopquery.sql...ANTHINI\nishaa (53)* ↵ ×

-- 8 Recalculate and update the total cost of each order
PRINT 'Recalculating and updating total order costs...';
UPDATE Orders
SET TotalAmount = (
    SELECT SUM(P.Price * OD.Quantity)
    FROM OrderDetails OD
    JOIN Products P ON OD.ProductID = P.ProductID
    WHERE OD.OrderID = Orders.OrderID
);
-- Verify update
PRINT 'Order totals updated successfully. Displaying updated orders:';
SELECT * FROM Orders;
GO
```

	OrderID	CustomerID	OrderDate	TotalAmount
1	1	1	2025-03-20 00:00:00.000	21998.90
2	2	2	2025-03-21 00:00:00.000	1098.90
3	4	4	2025-03-18 00:00:00.000	6597.80
4	5	5	2025-03-22 00:00:00.000	5498.90
5	6	6	2025-03-17 00:00:00.000	16498.90
6	7	7	2025-03-16 00:00:00.000	2198.90
7	8	8	2025-03-23 00:00:00.000	43998.90
8	9	9	2025-03-15 00:00:00.000	878.90
9	10	10	2025-03-14 00:00:00.000	2748.90
10	11	2	2025-03-23 10:27:52.230	NULL

```
100 % 
Query executed successfully.
```

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.

```
techshopquery.sql...ANTHINI\nisha (53)* ↻ X

-- 9 Delete all orders and associated order details for a specific customer (Using Stored Procedure)
PRINT 'Creating stored procedure for deleting all orders for a customer...';
GO
CREATE OR ALTER PROCEDURE DeleteCustomerOrders
    @CustomerID INT
AS
BEGIN
    -- Check if Customer exists before deleting orders
    IF NOT EXISTS (SELECT 1 FROM Orders WHERE CustomerID = @CustomerID)
    BEGIN
        PRINT 'Error: Customer has no orders.';
        RETURN;
    END

    DELETE FROM OrderDetails WHERE OrderID IN (SELECT OrderID FROM Orders WHERE CustomerID = @CustomerID);
    DELETE FROM Orders WHERE CustomerID = @CustomerID;

    PRINT 'Orders and order details for the customer deleted successfully.';
END;
GO

-- EXECUTE: Run this command to delete all orders for a customer
EXEC DeleteCustomerOrders @CustomerID = 4;
GO
```

```
techshopquery.sql...ANTHINI\nisha (53)* ↻ X

-- EXECUTE: Run this command to delete all orders for a customer
EXEC DeleteCustomerOrders @CustomerID = 4;
GO
-- Display orders along with order details using a JOIN
PRINT 'Displaying all remaining orders with their order details:';
SELECT
    O.OrderID,
    O.CustomerID,
    O.OrderDate,
    O.TotalAmount,
    OD.OrderDetailID,
    OD.ProductID,
    OD.Quantity
FROM Orders O
LEFT JOIN OrderDetails OD ON O.OrderID = OD.OrderID
ORDER BY O.OrderID;
GO
```

100 %

OrderID	CustomerID	OrderDate	TotalAmount	OrderDetailID	ProductID	Quantity
1	1	2025-03-20 00:00:00.000	21998.90	1	1	1
2	2	2025-03-21 00:00:00.000	1098.90	2	8	1
3	5	2025-03-22 00:00:00.000	5498.90	5	3	1
4	6	2025-03-17 00:00:00.000	16498.90	6	5	1
5	7	2025-03-16 00:00:00.000	2198.90	7	6	1
6	8	2025-03-23 00:00:00.000	43998.90	8	7	1
7	9	2025-03-15 00:00:00.000	878.90	9	9	1
8	10	2025-03-14 00:00:00.000	2748.90	10	10	1
9	11	2025-03-23 10:27:52.230	NULL	NULL	NULL	NULL

Query executed successfully.

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.

```
-- 10 Insert a new electronic gadget into the "Products" table
PRINT 'Inserting a new product...';
INSERT INTO Products (ProductName, Description, Price)
VALUES ('Wireless Earbuds', 'Bluetooth 5.0 wireless earbuds with noise cancellation', 3499.00);

-- Verify insert
PRINT 'New product added. Displaying updated products list:';
SELECT * FROM Products;
GO
```

The screenshot shows the results of the query execution. A table titled 'Results' displays 11 rows of product data. The last row, which was inserted, has 'ProductID' 11, 'ProductName' 'Wireless Earbuds', 'Description' 'Bluetooth 5.0 wireless earbuds with noise cancellation', and 'Price' 3499.00.

	ProductID	ProductName	Description	Price
1	1	Smartphone	Latest Android smartphone	21998.90
2	2	Laptop	High-performance laptop	65998.90
3	3	Smartwatch	Fitness and health tracking smartwatch	5498.90
4	4	Headphones	Noise-cancelling over-ear headphones	3298.90
5	5	Tablet	10-inch Android tablet	16498.90
6	6	Bluetooth Speaker	Portable wireless speaker	2198.90
7	7	Gaming Console	Next-gen gaming console	43998.90
8	8	Power Bank	10000mAh power bank	1098.90
9	9	Wireless Mouse	Ergonomic wireless mouse	878.90
10	10	Keyboard	Mechanical gaming keyboard	2748.90
11	11	Wireless Earbuds	Bluetooth 5.0 wireless earbuds with noise cancellation	3499.00

Query executed successfully.

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.

```
-- 11 Update the status of a specific order
-- Ensure 'Status' column exists
IF NOT EXISTS (SELECT * FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'Orders' AND COLUMN_NAME = 'Status')
BEGIN
    ALTER TABLE Orders ADD Status NVARCHAR(20) DEFAULT 'Pending';
END
GO

PRINT 'Creating stored procedure for updating order status...';
GO

CREATE OR ALTER PROCEDURE UpdateOrderStatus
    @OrderID INT,
    @NewStatus NVARCHAR(20)
AS
BEGIN
    UPDATE Orders
    SET Status = @NewStatus
    WHERE OrderID = @OrderID;

    PRINT 'Order status updated successfully.';
END;
GO

-- EXECUTE: Run this command to update order status
EXEC UpdateOrderStatus @OrderID = 6, @NewStatus = 'Shipped';
GO

SELECT * FROM Orders;
```

100 %

Results Messages

	OrderID	CustomerID	OrderDate	TotalAmount	Status
1	1	1	2025-03-20 00:00:00.000	21998.90	NULL
2	2	2	2025-03-21 00:00:00.000	1098.90	NULL
3	5	5	2025-03-22 00:00:00.000	5498.90	NULL
4	6	6	2025-03-17 00:00:00.000	16498.90	Shipped
5	7	7	2025-03-16 00:00:00.000	2198.90	NULL
6	8	8	2025-03-23 00:00:00.000	43998.90	NULL
7	9	9	2025-03-15 00:00:00.000	878.90	NULL
8	10	10	2025-03-14 00:00:00.000	2748.90	NULL
9	11	2	2025-03-23 10:27:52.230	NULL	NULL

✓ Query executed successfully.

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.

```
techshopquery.sql...ANTHINI\nisha (53)* ↵ X
-- 12 Calculate and update the number of orders placed by each customer
-- Ensure 'OrderCount' column exists
IF NOT EXISTS (SELECT * FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'Customers' AND COLUMN_NAME = 'OrderCount')
BEGIN
    ALTER TABLE Customers ADD OrderCount INT DEFAULT 0;
END
GO

PRINT 'Creating stored procedure for updating order counts...';
GO
CREATE OR ALTER PROCEDURE UpdateCustomerOrderCount
AS
BEGIN
    UPDATE Customers
    SET OrderCount = (
        SELECT COUNT(*) FROM Orders WHERE Orders.CustomerID = Customers.CustomerID
    );
    PRINT 'Order counts updated successfully.';
END;
GO

-- EXECUTE: Run this command to update order counts for all customers
EXEC UpdateCustomerOrderCount;
GO
-- Display updated Customers table with OrderCount
PRINT 'Displaying updated Customers table with OrderCount:';
SELECT CustomerID, FirstName, LastName, Email, Phone, Address, OrderCount FROM Customers;
GO
```

```
techshopquery.sql...ANTHINI\nisha (53)* ↵ X
-- Display Customers and their orders using a JOIN
PRINT 'Displaying Customers and their respective orders:';
SELECT
    C.CustomerID,
    C.FirstName,
    C.LastName,
    C.Email,
    C.OrderCount,
    O.OrderID,
    O.OrderDate,
    O.TotalAmount
FROM Customers C
LEFT JOIN Orders O ON C.CustomerID = O.CustomerID
ORDER BY C.CustomerID;
GO
```

100 %

Results Messages

	CustomerID	FirstName	LastName	Email	Phone	Address	OrderCount
1	1	Rohith	Kumar	rohit.kumar@email.com	9876543210	Hyderabad, Telangana	1
2	2	Sandeep	Reddy	sandeep.reddy@email.com	9876543211	Bangalore, Karnataka	2
3	3	Gani	Sheikh	gani.sheikh@email.com	9876543212	Chennai, Tamil Nadu	0
4	4	Nikila	M	nikila.m@email.com	9876543213	Mumbai, Maharashtra	0
5	5	Vicky	Raj	stjoseph@email.com	9876543214	Chennai, TamilNadu	1
6	6	Nivedha	S	nivedha.s@email.com	9876543215	Coimbatore, Tamil Nadu	1
7	7	Yogesh	Naidu	yogesh.naidu@email.com	9876543216	Delhi	1
8	8	Harini	Rao	harini.rao@email.com	9876543217	Kolkata, West Bengal	1
9	9	Harish	Patel	harish.patel@email.com	9876543218	Ahmedabad, Gujarat	1
10	10	Prabha	Devi	prabha.devi@email.com	9876543219	Jaipur, Rajasthan	1
11	11	Raj	Sharma	raj.sharma@email.com	9876543220	Chennai, Tamil Nadu	0

	CustomerID	FirstName	LastName	Email	OrderCount	OrderID	OrderDate	TotalAmount
1	1	Rohith	Kumar	rohit.kumar@email.com	1	1	2025-03-20 00:00:00.000	21998.90
2	2	Sandeep	Reddy	sandeep.reddy@email.com	2	2	2025-03-21 00:00:00.000	1098.90
3	2	Sandeep	Reddy	sandeep.reddy@email.com	2	11	2025-03-23 10:27:52.230	NULL
4	3	Gani	Sheikh	gani.sheikh@email.com	0	NULL	NULL	NULL
5	4	Nikila	M	nikila.m@email.com	0	NULL	NULL	NULL
6	5	Vicky	Raj	stjoseph@email.com	1	5	2025-03-22 00:00:00.000	5498.90
7	6	Nivedha	S	nivedha.s@email.com	1	6	2025-03-17 00:00:00.000	16498.90
8	7	Yogesh	Naidu	yogesh.naidu@email.com	1	7	2025-03-16 00:00:00.000	2198.90
9	8	Harini	Rao	harini.rao@email.com	1	8	2025-03-23 00:00:00.000	43998.90
10	9	Harish	Patel	harish.patel@email.com	1	9	2025-03-15 00:00:00.000	878.90
11	10	Prabha	Devi	prabha.devi@email.com	1	10	2025-03-14 00:00:00.000	2748.90
12	11	Raj	Sharma	raj.sharma@email.com	0	NULL	NULL	NULL

Query executed successfully.

NISHANTHINI

### 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.

techshopquery.sql...ANTHINI\Nisha (53)\* ↗ X

```
-- 1 Retrieve a list of all orders along with customer information (e.g., customer name) for each order
PRINT 'Displaying all orders with customer information:';
SELECT
    O.OrderID,
    O.OrderDate,
    O.TotalAmount,
    C.CustomerID,
    C.FirstName + ' ' + C.LastName AS CustomerName,
    C.Email,
    C.Phone
FROM Orders O
JOIN Customers C ON O.CustomerID = C.CustomerID
ORDER BY O.OrderDate DESC;
GO
```

100 %

Results Messages

	OrderID	OrderDate	TotalAmount	CustomerID	CustomerName	Email	Phone
1	11	2025-03-23 10:27:52.230	NULL	2	Sandeep Reddy	sandeep.reddy@email.com	9876543211
2	8	2025-03-23 00:00:00.000	43998.90	8	Harini Rao	harini.rao@email.com	9876543217
3	5	2025-03-22 00:00:00.000	5498.90	5	Vicky Raj	stjoseph@email.com	9876543214
4	2	2025-03-21 00:00:00.000	1098.90	2	Sandeep Reddy	sandeep.reddy@email.com	9876543211
5	1	2025-03-20 00:00:00.000	21998.90	1	Rohith Kumar	rohit.kumar@email.com	9876543210
6	6	2025-03-17 00:00:00.000	16498.90	6	Nivedha S	nivedha.s@email.com	9876543215
7	7	2025-03-16 00:00:00.000	2198.90	7	Yogesh Naidu	yogesh.naidu@email.com	9876543216
8	9	2025-03-15 00:00:00.000	878.90	9	Harish Patel	harish.patel@email.com	9876543218
9	10	2025-03-14 00:00:00.000	2748.90	10	Prabha Devi	prabha.devi@email.com	9876543219

2. Write an SQL query to find the total revenue generated by each electronic gadget product. Include the product name and the total revenue.

The screenshot shows an SQL query being run in SSMS. The query selects ProductID, ProductName, and TotalRevenue from the OrderDetails and Products tables, grouped by ProductID and ProductName, and ordered by TotalRevenue in descending order. The results show 8 rows of data, and a message at the bottom indicates the query was executed successfully.

```
-- 2 Find the total revenue generated by each electronic gadget product
PRINT 'Displaying total revenue generated by each product:';
SELECT
    P.ProductID,
    P.ProductName,
    SUM(P.Price * OD.Quantity) AS TotalRevenue
FROM OrderDetails OD
JOIN Products P ON OD.ProductID = P.ProductID
GROUP BY P.ProductID, P.ProductName
ORDER BY TotalRevenue DESC;
GO
```

	ProductID	ProductName	TotalRevenue
1	7	Gaming Console	43998.90
2	1	Smartphone	21998.90
3	5	Tablet	16498.90
4	3	Smartwatch	5498.90
5	10	Keyboard	2748.90
6	6	Bluetooth Speaker	2198.90
7	8	Power Bank	1098.90
8	9	Wireless Mouse	878.90

Query executed successfully.

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

The screenshot shows an SQL query being run in SSMS. The query selects CustomerID, CustomerName, Email, and Phone from the Customers and Orders tables, grouped by CustomerID, FirstName, LastName, Email, and Phone, and having a count of orders greater than 0. The results show 4 rows of data, and a message at the bottom indicates the query was executed successfully.

```
-- 3 List all customers who have made at least one purchase
PRINT 'Displaying customers who have made at least one purchase:';
SELECT
    C.CustomerID,
    C.FirstName + ' ' + C.LastName AS CustomerName,
    C.Email,
    C.Phone,
    COUNT(O.OrderID) AS NumberOfOrders
FROM Customers C
JOIN Orders O ON C.CustomerID = O.CustomerID
GROUP BY C.CustomerID, C.FirstName, C.LastName, C.Email, C.Phone
HAVING COUNT(O.OrderID) > 0
ORDER BY NumberOfOrders DESC;
GO
```

100 %

Results Messages

	CustomerID	CustomerName	Email	Phone	NumberOfOrders
1	2	Sandeep Reddy	sandeep.reddy@email.com	9876543211	2
2	5	Vicky Raj	stjoseph@email.com	9876543214	1
3	6	Nivedha S	nivedha.s@email.com	9876543215	1
4	7	Yogesh Naidu	yogesh.naidu@email.com	9876543216	1
5	8	Harini Rao	harini.rao@email.com	9876543217	1
6	9	Harish Patel	harish.patel@email.com	9876543218	1
7	10	Prabha Devi	prabha.devi@email.com	9876543219	1
8	1	Rohith Kumar	rohit.kumar@email.com	9876543210	1

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.

```
techshopquery.sql...ANTHINI\nishash (53)* ↵ X
-- 4 Find the most popular electronic gadget (highest total quantity ordered)
PRINT 'Displaying the most popular electronic gadget based on total quantity ordered:';
SELECT TOP 1
    P.ProductID,
    P.ProductName,
    SUM(OD.Quantity) AS TotalQuantityOrdered
FROM OrderDetails OD
JOIN Products P ON OD.ProductID = P.ProductID
GROUP BY P.ProductID, P.ProductName
ORDER BY TotalQuantityOrdered DESC;
GO
```

100 %

PRINT Displaying most popular gadgets

Results Messages

	ProductID	ProductName	TotalQuantityOrdered
1	3	Smartwatch	1

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

```
techshopquery.sql...ANTHINI\nishash (53)* ↵ X
-- 5 Retrieve a list of electronic gadgets along with their corresponding categories
PRINT 'Displaying electronic gadgets along with their categories:';
SELECT
    P.ProductID,
    P.ProductName,
    P.Description
FROM Products P
ORDER BY P.ProductName;
GO
```

100 %

	ProductID	ProductName	Description
1	6	Bluetooth Speaker	Portable wireless speaker
2	7	Gaming Console	Next-gen gaming console
3	4	Headphones	Noise-cancelling over-ear headphones
4	10	Keyboard	Mechanical gaming keyboard
5	2	Laptop	High-performance laptop
6	8	Power Bank	10000mAh power bank
7	1	Smartphone	Latest Android smartphone
8	3	Smartwatch	Fitness and health tracking smartwatch
9	5	Tablet	10-inch Android tablet
10	11	Wireless Earbuds	Bluetooth 5.0 wireless earbuds with noise cancel...
11	9	Wireless Mouse	Ergonomic wireless mouse

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

```
-- 6 Calculate the average order value for each customer
PRINT 'Displaying average order value per customer:';
SELECT
    C.CustomerID,
    C.FirstName + ' ' + C.LastName AS CustomerName,
    AVG(O.TotalAmount) AS AvgOrderValue
FROM Customers C
JOIN Orders O ON C.CustomerID = O.CustomerID
GROUP BY C.CustomerID, C.FirstName, C.LastName
ORDER BY AvgOrderValue DESC;
GO
```

100 %

	CustomerID	CustomerName	AvgOrderValue
1	8	Harini Rao	43998.900000
2	1	Rohith Kumar	21998.900000
3	6	Nivedha S	16498.900000
4	5	Vicky Raj	5498.900000
5	10	Prabha Devi	2748.900000
6	7	Yogesh Naidu	2198.900000
7	2	Sandeep Reddy	1098.900000
8	9	Harish Patel	878.900000

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

The screenshot shows an open query window titled "techshopquery.sql...ANTHINI\nisha (53)\*". The code is as follows:

```
-- 7 Find the order with the highest total revenue
PRINT 'Displaying the order with the highest total revenue:';
SELECT TOP 1
    O.OrderID,
    O.OrderDate,
    O.TotalAmount,
    C.CustomerID,
    C.FirstName + ' ' + C.LastName AS CustomerName,
    C.Email
FROM Orders O
JOIN Customers C ON O.CustomerID = C.CustomerID
ORDER BY O.TotalAmount DESC;
GO
```

The results pane shows a single row of data:

	OrderID	OrderDate	TotalAmount	CustomerID	CustomerName	Email
1	8	2025-03-23 00:00:00.000	43998.90	8	Harini Rao	harini.rao@email.com

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

The screenshot shows an open query window titled "techshopquery.sql...ANTHINI\nisha (53)\*". The code is as follows:

```
-- 8 List electronic gadgets and the number of times each product has been ordered
PRINT 'Displaying electronic gadgets and the number of times they have been ordered:';
SELECT
    P.ProductID,
    P.ProductName,
    COUNT(OD.OrderID) AS NumberofTimesOrdered
FROM OrderDetails OD
JOIN Products P ON OD.ProductID = P.ProductID
GROUP BY P.ProductID, P.ProductName
ORDER BY NumberofTimesOrdered DESC;
GO
```

The results pane shows the following data:

	ProductID	ProductName	NumberofTimesOrdered
1	1	Smartphone	1
2	3	Smartwatch	1
3	5	Tablet	1
4	6	Bluetooth Speaker	1
5	7	Gaming Console	1
6	8	Power Bank	1
7	9	Wireless Mouse	1
8	10	Keyboard	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.

```
techshopquery.sql...ANTHINI\nisha (53)* ✎ X
-- 9 Find customers who have purchased a specific electronic gadget product (Dynamic Input)
PRINT 'Creating stored procedure to find customers who purchased a specific product...';
GO
CREATE OR ALTER PROCEDURE GetCustomersByProduct
    @ProductName NVARCHAR(100)
AS
BEGIN
    PRINT 'Retrieving customers who have purchased: ' + @ProductName;

    SELECT DISTINCT
        C.CustomerID,
        C.FirstName + ' ' + C.LastName AS CustomerName,
        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;
END;
GO

-- EXECUTE: Run this command to get customers who bought a specific product
EXEC GetCustomersByProduct @ProductName = 'Smartphone';
GO
```

100 %

Results Messages

	CustomerID	CustomerName	Email	Phone
1	1	Rohith Kumar	rohit.kumar@email.com	9876543210

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.

```
techshopquery.sql...ANTHINI\nisha (53)* ✎ X
-- 10 Calculate total revenue generated by orders placed within a specific time period (Dynamic Input)
PRINT 'Creating stored procedure to calculate total revenue in a given time period...';
GO
CREATE OR ALTER PROCEDURE GetTotalRevenueByDateRange
    @StartDate DATE,
    @EndDate DATE
AS
BEGIN
    PRINT 'Calculating total revenue from ' + CAST(@StartDate AS NVARCHAR) + ' to ' + CAST(@EndDate AS NVARCHAR);

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

-- EXECUTE: Run this command to get revenue for a date range
EXEC GetTotalRevenueByDateRange @StartDate = '2025-03-01', @EndDate = '2025-03-31';
GO
```

100 %

Results Messages

	TotalRevenue
1	94921.20

#### Task 4- Subquery and its type:

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

```
-- 1 Find customers who have NOT placed any orders
PRINT 'Displaying customers who have not placed any orders:';
SELECT
    C.CustomerID,
    C.FirstName + ' ' + C.LastName AS CustomerName,
    C.Email,
    C.Phone
FROM Customers C
LEFT JOIN Orders O ON C.CustomerID = O.CustomerID
WHERE O.OrderID IS NULL;
GO
```

CustomerID	CustomerName	Email	Phone
3	Gani Sheikh	gani.sheikh@gmail.com	9876543212
4	Nikila M	nikila.m@gmail.com	9876543213
11	Raj Sharma	raj.sharma@gmail.com	9876543220

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

```
-- 2 Find the total number of products available for sale
PRINT 'Displaying total number of products available for sale:';
SELECT COUNT(ProductID) AS TotalProducts
FROM Products;
GO
```

TotalProducts
11

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

```
-- 3 Calculate the total revenue generated by TechShop
PRINT 'Displaying total revenue generated by TechShop:';
SELECT SUM(TotalAmount) AS TotalRevenue
FROM Orders;
GO
```

TotalRevenue
94921.20

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.

```
-- 4 Calculate the average quantity ordered for products in a specific category (Dynamic Input)
PRINT 'Creating stored procedure to get average quantity ordered for a specific category...';
GO
CREATE OR ALTER PROCEDURE GetAverageQuantityByCategory
    @CategoryName NVARCHAR(100)
AS
BEGIN
    PRINT 'Retrieving average quantity ordered for category: ' + @CategoryName;

    SELECT
        COALESCE(AVG(OD.Quantity), 0) AS AvgQuantityOrdered
    FROM OrderDetails OD
    JOIN Products P ON OD.ProductID = P.ProductID
    WHERE P.Description = @CategoryName;
END;
GO

-- EXECUTE: Run this command to get average quantity for a category
EXEC GetAverageQuantityByCategory @CategoryName = 'Smartphone';
GO
```

AvgQuantityOrdered
0

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.

```
techshopquery.sql...ANTHINI\nishash (53)* ↵ X
-- 5 Calculate the total revenue generated by a specific customer (Dynamic Input)
PRINT 'Creating stored procedure to calculate total revenue for a specific customer...';
GO
CREATE OR ALTER PROCEDURE GetTotalRevenueByCustomer
    @CustomerID INT
AS
BEGIN
    PRINT 'Calculating total revenue for Customer ID: ' + CAST(@CustomerID AS NVARCHAR);
    SELECT
        C.CustomerID,
        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;
END;
GO

-- EXECUTE: Run this command to get revenue for a customer
EXEC GetTotalRevenueByCustomer @CustomerID = 5;
GO
```

100 %

Results Messages

	CustomerID	CustomerName	TotalRevenue
1	5	Vicky Raj	5498.90

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.

```
techshopquery.sql...ANTHINI\nishash (53)* ↵ X
-- 6 Find the customers who have placed the most orders
PRINT 'Displaying customers who have placed the most orders:';
SELECT TOP 5
    C.CustomerID,
    C.FirstName + ' ' + C.LastName AS CustomerName,
    COUNT(O.OrderID) AS NumberOfOrders
FROM Customers C
JOIN Orders O ON C.CustomerID = O.CustomerID
GROUP BY C.CustomerID, C.FirstName, C.LastName
ORDER BY NumberOfOrders DESC;
GO
```

100 %

Results Messages

	CustomerID	CustomerName	NumberOfOrders
1	2	Sandeep Reddy	2
2	1	Rohith Kumar	1
3	8	Harini Rao	1
4	7	Yogesh Naidu	1
5	6	Nivedha S	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.

```
techshopquery.sql...ANTHINI\nisha (53)* # X
-- 7 Find the most popular product category (highest total quantity ordered)
PRINT 'Displaying the most popular product category based on total quantity ordered:';
SELECT TOP 1
    P.Description AS Category,
    SUM(OD.Quantity) AS TotalQuantityOrdered
FROM OrderDetails OD
JOIN Products P ON OD.ProductID = P.ProductID
GROUP BY P.Description
ORDER BY TotalQuantityOrdered DESC;
GO
```

Category	TotalQuantityOrdered
10-inch Android tablet	1

8. Write an SQL query to find the customer who has spent the most money (highest total revenue) on electronic gadgets. List their name and total spending.

```
techshopquery.sql...ANTHINI\nisha (53)* # X
-- 8 Find the customer who has spent the most money (highest total revenue)
PRINT 'Displaying the customer who has spent the most money:';
SELECT TOP 1
    C.CustomerID,
    C.FirstName + ' ' + C.LastName AS CustomerName,
    SUM(O.TotalAmount) AS TotalSpending
FROM Customers C
JOIN Orders O ON C.CustomerID = O.CustomerID
GROUP BY C.CustomerID, C.FirstName, C.LastName
ORDER BY TotalSpending DESC;
GO
```

CustomerID	CustomerName	TotalSpending
8	Harini Rao	43998.90

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

```
techshopquery.sql...ANTHINI\nisha (53)* # X
-- 9 Calculate the average order value for all customers
PRINT 'Displaying the average order value for all customers:';
SELECT
    SUM(TotalAmount) / NULLIF(COUNT(OrderID), 0) AS AverageOrderValue
FROM Orders;
GO
```

AverageOrderValue
10546.800000

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.

```
techshopquery.sql...ANTHINI\nisha (53)* # X
-- 10 Find the total number of orders placed by each customer
PRINT 'Displaying the total number of orders placed by each customer:';
SELECT
    C.CustomerID,
    C.FirstName + ' ' + C.LastName AS CustomerName,
    COUNT(O.OrderID) AS TotalOrders
FROM Customers C
LEFT JOIN Orders O ON C.CustomerID = O.CustomerID
GROUP BY C.CustomerID, C.FirstName, C.LastName
ORDER BY TotalOrders DESC;
GO
```

CustomerID	CustomerName	TotalOrders
2	Sandeep Reddy	2
1	Rohith Kumar	1
5	Vicky Raj	1
6	Nivedha S	1
7	Yogesh Naidu	1
8	Harini Rao	1
9	Harish Patel	1
10	Prabha Devi	1
11	Raj Sharma	0
3	Gani Sheikh	0
4	Nikila M	0

## OOPS IMPLEMENTATION

```
C:\Users\nisha\OneDrive\Documents\TechShop>python main.py

➊ Welcome to TechShop CLI ➋
1⠁ Manage Customers
2⠁ Manage Products
3⠁ Manage Orders
4⠁ Manage Inventory
5⠁ Exit
Enter your choice: 1

👤 Customer Management
1⠁ Register New Customer
2⠁ View Customer Details
3⠁ Update Customer Info
4⠁ Back to Main Menu
Enter your choice: 1
Enter First Name: Dharshan
Enter Last Name: Aruna
Enter Email: appu@gmail.com
Enter Phone: 8825598237
Enter Address: Chennai ,India
Connected to TechShop database successfully!
✓ Customer registered successfully!

C:\Users\nisha\OneDrive\Documents\TechShop>python main.py

➊ Welcome to TechShop CLI ➋
1⠁ Manage Customers
2⠁ Manage Products
3⠁ Manage Orders
4⠁ Manage Inventory
5⠁ Exit
Enter your choice: 1

👤 Customer Management
1⠁ Register New Customer
2⠁ View Customer Details
3⠁ Update Customer Info
4⠁ Back to Main Menu
Enter your choice: 2
Enter Customer ID: 1
Connected to TechShop database successfully!
█ Customer Details: [(1, 'Rohith', 'Kumar', 'rohith.kumar@email.com', '9876543210', 'Hyderabad, Telangana', 1)]

👤 Customer Management
1⠁ Register New Customer
2⠁ View Customer Details
3⠁ Update Customer Info
4⠁ Back to Main Menu
Enter your choice: 1
Enter Customer ID: 1
Enter New Email: rohithanbu@gmail.com
Enter New Address: Minjur,Tamilnadu
Connected to TechShop database successfully!
✓ Customer info updated successfully!

👤 Customer Management
1⠁ Register New Customer
2⠁ View Customer Details
3⠁ Update Customer Info
4⠁ Back to Main Menu
Enter your choice: |
```

```
👤 Customer Management
1□ Register New Customer
2□ View Customer Details
3□ Update Customer Info
4□ Back to Main Menu
Enter your choice: 4
```

```
💻 Welcome to TechShop CLI 💻
1□ Manage Customers
2□ Manage Products
3□ Manage Orders
4□ Manage Inventory
5□ Exit
Enter your choice: |
```

```
💻 Welcome to TechShop CLI 💻
1□ Manage Customers
2□ Manage Products
3□ Manage Orders
4□ Manage Inventory
5□ Exit
Enter your choice: 2
```

#### 📦 Product Management

```
1□ Add New Product
2□ View Product Details
3□ Back to Main Menu
```

```
Enter your choice: 1
```

```
Enter Product Name: Clock
```

```
Enter Description: Timer
```

```
Enter Price: 300
```

```
Connected to TechShop database successfully!
```

```
✓ Product 'Clock' added successfully!
```

#### 📦 Product Management

```
1□ Add New Product
2□ View Product Details
3□ Back to Main Menu
```

```
Enter your choice: |
```

```
C:\Users\nisha\OneDrive\Documents\TechShop>python main.py

[ Welcome to TechShop CLI ]
1□ Manage Customers
2□ Manage Products
3□ Manage Orders
4□ Manage Inventory
5□ Exit
Enter your choice: 2

[ Product Management ]
1□ Add New Product
2□ View Product Details
3□ Back to Main Menu
Enter your choice: 2
Enter Product ID: 1
Connected to TechShop database successfully!
[ Product Details: [(1, 'Smartphone', 'Latest Android smartphone', Decimal('21998.90'))]

[ Product Management ]
1□ Add New Product
2□ View Product Details
3□ Back to Main Menu
Enter your choice: |

[ Welcome to TechShop CLI ]
1□ Manage Customers
2□ Manage Products
3□ Manage Orders
4□ Manage Inventory
5□ Exit
Enter your choice: 3

[ Order Management ]
1□ Place New Order
2□ View Order Details
3□ Update Order Status
4□ Back to Main Menu
Enter your choice: 1
Enter Customer ID: 1
Enter Order Date (YYYY-MM-DD): 2025-03-12
Enter Total Amount: 120
Connected to TechShop database successfully!
[ Order placed successfully for CustomerID 1!

[ Order Management ]
1□ Place New Order
2□ View Order Details
3□ Update Order Status
4□ Back to Main Menu
Enter your choice: |

[ Order Management ]
1□ Place New Order
2□ View Order Details
3□ Update Order Status
4□ Back to Main Menu
Enter your choice: 2
Enter Order ID: 1
Connected to TechShop database successfully!
[ Order Details: [(1, datetime.datetime(2025, 3, 20, 0, 0), Decimal('21998.90'), None, 'Rohith', 'Kumar')]

[ Order Management ]
1□ Place New Order
2□ View Order Details
3□ Update Order Status
4□ Back to Main Menu
Enter your choice: |
```

```
📄 Order Management
1□ Place New Order
2□ View Order Details
3□ Update Order Status
4□ Back to Main Menu
Enter your choice: 3
Enter Order ID: 1
Enter New Order Status: Shipped
Connected to TechShop database successfully!
✓ Order 1 updated to status: Shipped
```

```
📄 Order Management
1□ Place New Order
2□ View Order Details
3□ Update Order Status
4□ Back to Main Menu
Enter your choice: |
```

```
💻 Welcome to TechShop CLI 💻
1□ Manage Customers
2□ Manage Products
3□ Manage Orders
4□ Manage Inventory
5□ Exit
Enter your choice: 4

⌚ Inventory Management
1□ View Product Prices
2□ List All Products
3□ Back to Main Menu
Enter your choice: 1
Enter Product ID: 1
Connected to TechShop database successfully!
⌚ Product Price: 21998.90

⌚ Inventory Management
1□ View Product Prices
2□ List All Products
3□ Back to Main Menu
Enter your choice: 2
Connected to TechShop database successfully!
⌚ Product List: [(1, 'Smartphone', 'Latest Android smartphone', Decimal('21998.90')), (2, 'Laptop', 'High-performance laptop', Decimal('65998.90')), (3, 'Smartwatch', 'Fitness and health tracking smartwatch', Decimal('5498.90')), (4, 'Headphones', 'Noise-cancelling over-ear headphones', Decimal('3298.90')), (5, 'Tablet', '10-inch Android tablet', Decimal('16498.90')), (6, 'Bluetooth Speaker', 'Portable wireless speaker', Decimal('2198.90')), (7, 'Gaming Console', 'Next-gen gaming console', Decimal('43998.90')), (8, 'Power Bank', '10000mAh power bank', Decimal('1098.90')), (9, 'Wireless Mouse', 'Ergonomic wireless mouse', Decimal('878.90')), (10, 'Keyboard', 'Mechanical gaming keyboard', Decimal('2748.90')), (11, 'Wireless Earbuds', 'Bluetooth 5.0 wireless earbuds with noise cancellation', Decimal('3499.00')), (12, 'Laptop', 'Gaming Laptop with 16GB RAM', Decimal('75000.00')), (13, 'Laptop', 'Gaming Laptop with 16GB RAM', Decimal('75000.00')), (14, 'Clock', 'Timer', Decimal('300.00'))]

⌚ Inventory Management
1□ View Product Prices
2□ List All Products
3□ Back to Main Menu
Enter your choice: 3
```

```
📦 Inventory Management
1□ View Product Prices
2□ List All Products
3□ Back to Main Menu
Enter your choice: 3

💻 Welcome to TechShop CLI 💻
1□ Manage Customers
2□ Manage Products
3□ Manage Orders
4□ Manage Inventory
5□ Exit
Enter your choice: 5
👋 Exiting TechShop CLI. Have a great day!
```

```
C:\Users\nisha\OneDrive\Documents\TechShop>
```