Assignment - 1 :

Task-1:

1. Create the database named "TechShop".

A. create database TechShop;

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

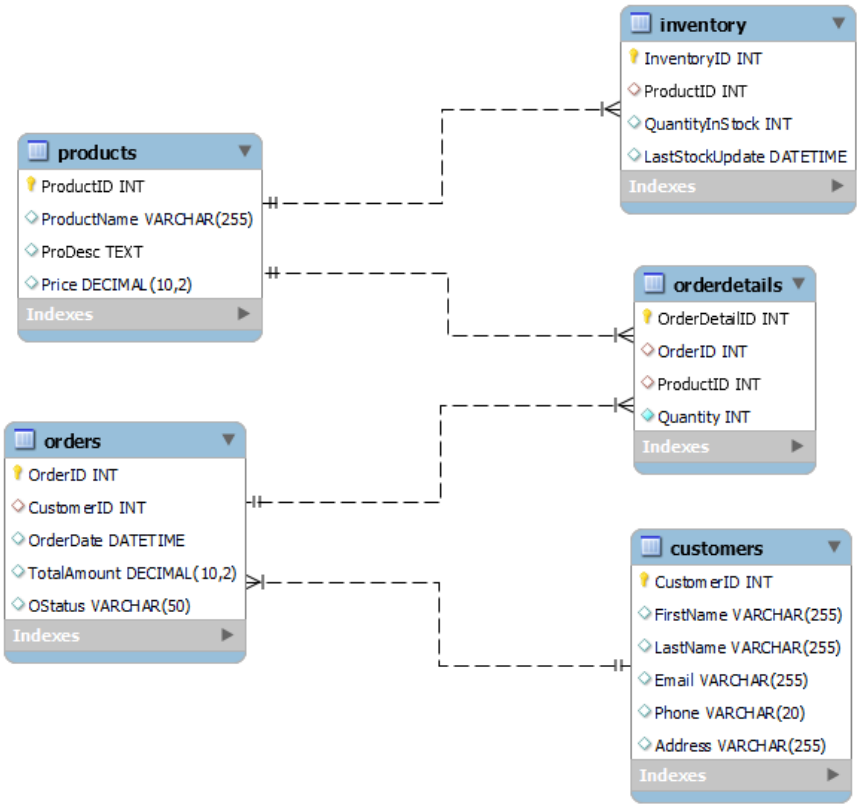
A.

```
create table Customers
(
    CustomerID varchar(6) primary key,
    FirstName varchar(50),
    LastName varchar(50),
    Email varchar(40),
    Phone bigint unique,
    Address text
);

create table Products
(
    ProductID varchar(6) primary key,
    ProductName varchar(50),
    ProDesc text,
    Price decimal(10, 2)
);

create table Orders
(
    OrderID varchar(6) primary key,
    CustomerID varchar(6),
    OrderDate date,
    TotalAmount Decimal(10, 2),
    Foreign key(CustomerID) references Customers(CustomerID)
);
```

```
create table OrderDetails
(
    OrderDetailID varchar(6) primary key,
    OrderID varchar(6),
    ProductID varchar(6),
    Quantity int,
    Foreign key(OrderID) references Orders(OrderID),
    Foreign key(ProductID) references Products(ProductID)
);

create table Inventory
(
    InventoryID varchar(6) primary key,
    ProductID varchar(6),
    QuantityInStock int,
    LastStockUpdate date,
    Foreign key(ProductID) references Products(ProductID)
);
```

52	15:23:39	create table Customers (CustomerID varchar(6) primary key, FirstName varchar(50), LastName varchar(50...	0 row(s) affected		0.031 sec
53	15:23:39	create table Products (ProductID varchar(6) primary key, ProductName varchar(50), ProDesc text, Pric...	0 row(s) affected		0.000 sec
54	15:23:39	create table Orders (OrderID varchar(6) primary key, CustomerID varchar(6), OrderDate date, TotalAmo...	0 row(s) affected		0.015 sec
55	15:23:39	create table OrderDetails (OrderDetailID varchar(6) primary key, OrderID varchar(6), ProductID varchar(6)...	0 row(s) affected		0.032 sec
56	15:23:39	create table Inventory (InventoryID varchar(6) primary key, ProductID varchar(6), QuantityInStock int, ...	0 row(s) affected		0.031 sec

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



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

```sql
-- Customers table
INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone, Address)
VALUES
('C001', 'John', 'Doe', 'john.doe@email.com', 1234567890, '123 Main St'),
('C002', 'Jane', 'Smith', 'jane.smith@email.com', 9876543210, '456 Oak St'),
('C003', 'Robert', 'Johnson', 'robert.johnson@email.com', 5551234567, '789 Pine St'),
('C004', 'Alice', 'Johnson', 'alice.johnson@email.com', 5559876543, '101 Elm St'),
('C005', 'Michael', 'Miller', 'michael.miller@email.com', 8885551234, '202 Maple St'),
('C006', 'Emily', 'Davis', 'emily.davis@email.com', 3338885555, '303 Birch St'),
('C007', 'David', 'Brown', 'david.brown@email.com', 1113335555, '404 Cedar St'),
('C008', 'Sophia', 'Smith', 'sophia.smith@email.com', 7779991111, '505 Oak St'),
('C009', 'William', 'Taylor', 'william.taylor@email.com', 2223334444, '606 Pine St'),
('C010', 'Olivia', 'Moore', 'olivia.moore@email.com', 9992223333, '707 Elm St');

select * from customers;
```

```sql
-- Products table
INSERT INTO Products (ProductID, ProductName, ProDesc, Price)
VALUES
('P001', 'Laptop', 'High-performance laptop', 899.99),
('P002', 'Smartphone', 'Latest smartphone model', 599.99),
('P003', 'Headphones', 'Noise-canceling headphones', 149.99),
('P004', 'Tablet', '10-inch tablet', 349.99),
('P005', 'Smartwatch', 'Fitness and health tracker', 129.99),
('P006', 'Desktop Computer', 'Powerful desktop computer', 1299.99),
('P007', 'Wireless Mouse', 'Ergonomic wireless mouse', 29.99),
('P008', 'External Hard Drive', '1TB external hard drive', 79.99),
('P009', 'Printer', 'Color laser printer', 249.99),
('P010', 'Gaming Console', 'Next-gen gaming console', 499.99);

select * from products;
```

```sql
-- Orders table
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)
VALUES
('O001', 'C001', '2023-01-12', 929.97),
('O002', 'C001', '2023-02-01', 729.96),
('O003', 'C002', '2023-02-15', 1279.74),
('O004', 'C002', '2023-03-05', 1599.83),
('O005', 'C003', '2023-03-20', 1399.50),
('O006', 'C003', '2023-04-02', 199.98),
('O007', 'C004', '2023-04-15', 949.75),
('O008', 'C004', '2023-05-01', 459.92),
('O009', 'C005', '2023-05-15', 469.87),
('O010', 'C005', '2023-06-01', 629.91),
('O011', 'C006', '2023-06-15', 99.98),
('O012', 'C006', '2023-07-01', 49.98),
('O013', 'C007', '2023-07-15', 159.89),
('O014', 'C007', '2023-08-01', 239.88),
('O015', 'C008', '2023-08-15', 649.86);
```

```sql
-- OrderDetails table
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity)
VALUES
('OD001', 'O001', 'P001', 2),
('OD002', 'O001', 'P002', 1),
('OD003', 'O002', 'P003', 3),
('OD004', 'O003', 'P004', 1),
('OD005', 'O004', 'P005', 2),
('OD006', 'O004', 'P006', 1),
('OD007', 'O005', 'P007', 4),
('OD008', 'O005', 'P008', 1),
('OD009', 'O006', 'P009', 2),
('OD010', 'O006', 'P010', 1),
('OD011', 'O007', 'P001', 1),
('OD012', 'O007', 'P002', 2),
('OD013', 'O008', 'P003', 3),
('OD014', 'O008', 'P004', 1),
('OD015', 'O009', 'P005', 2);
```

```sql
-- Inventory table
INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate)
VALUES
('I001', 'P001', 20, '2023-01-12'),
('I002', 'P002', 15, '2023-02-01'),
('I003', 'P003', 30, '2023-02-15'),
('I004', 'P004', 25, '2023-03-05'),
('I005', 'P005', 50, '2023-03-20'),
('I006', 'P006', 10, '2023-04-02'),
('I007', 'P007', 40, '2023-04-15'),
('I008', 'P008', 5, '2023-05-01'),
('I009', 'P009', 18, '2023-05-15'),
('I010', 'P010', 12, '2023-06-01'),
('I011', 'P001', 30, '2023-06-15'),
('I012', 'P002', 25, '2023-07-01'),
('I013', 'P003', 35, '2023-07-15'),
('I014', 'P004', 30, '2023-08-01'),
('I015', 'P005', 40, '2023-08-15');

select * from inventory;
```

7	15:36:09	INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone, Address) VALUES ('C001', 'John',...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
8	15:36:09	select * from customers LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
9	15:36:09	INSERT INTO Products (ProductID, ProductName, ProDesc, Price) VALUES ('P001', 'Laptop', 'High-performan...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
10	15:36:09	select * from products LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
11	15:36:09	INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES ('O001', 'C001', '2023-01-12', ...	15 row(s) affected Records: 15 Duplicates: 0 Warnings: 0	0.000 sec
12	15:36:09	select * from orders LIMIT 0, 1000	15 row(s) returned	0.000 sec / 0.000 sec
13	15:36:09	INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity) VALUES ('OD001', 'O001', 'P001', 2)...	15 row(s) affected Records: 15 Duplicates: 0 Warnings: 0	0.015 sec
14	15:36:09	select * from orderdetails LIMIT 0, 1000	15 row(s) returned	0.000 sec / 0.000 sec
15	15:36:09	INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate) VALUES ('I001', 'P001', ...	15 row(s) affected Records: 15 Duplicates: 0 Warnings: 0	0.000 sec
16	15:36:09	select * from inventory LIMIT 0, 1000	15 row(s) returned	0.000 sec / 0.000 sec

Task-2:

1.

```sql
--  Write an SQL query to retrieve the names and emails of all customers.
select concat(FirstName,' ',LastName) as `Full Name`, Email
from customers;
```

	Full Name	Email
▶	Rahul Kumar	rahul.kumar@example.com
	Priya Sharma	priya.sharma@example.com
	Vikram Singh	vikram.singh@example.com
	Deepika Patel	deepika.patel@example.com
	Amit Verma	amit.verma@example.com
	Ananya Nair	ananya.nair@example.com
	Raj Malhotra	raj.malhotra@example.com
	Neha Srivastava	neha.srivastava@example.com
	Sandeep Gupta	sandeep.gupta@example.com
	Shreya Rajput	shreya.rajput@example.com

2.

```sql
-- Write an SQL query to list all orders with their order dates and corresponding customer names.
 select c.FirstName, c.LastName, o.OrderID, date(o.OrderDate) as `Order Date`
 from Orders as o
 join Customers as c on o.CustomerId = c.CustomerID;
```

FirstName	LastName	OrderID	Order Date
Rahul	Kumar	1	2024-01-13
Priya	Sharma	2	2024-01-13
Vikram	Singh	3	2024-01-13
Deepika	Patel	4	2024-01-13
Amit	Verma	5	2024-01-13
Ananya	Nair	6	2024-01-13
Raj	Malhotra	7	2024-01-13
Neha	Srivastava	8	2024-01-13
Sandeep	Gupta	9	2024-01-13
Shreya	Rajput	10	2024-01-13

3.

```sql
-- 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('Avinash', 'Dubey', 'avi.dubey@example.com', '9832432687', '123, Mahi St, Ranchi');
```

```
5  12:57:43  Insert INTO Customers (FirstName, LastName, Email, Phone, Address) values('Avinash', 'Dubey', 'avi.dubey@exa...  1 row(s) affected  0.000 sec
```

4.

```sql
-- 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 + (0.1*Price);
```

```
3  13:00:38  Update Products Set Price = Price + (0.1*Price)  10 row(s) affected Rows matched: 10  Changed: 10  Warnings: 0  0.000 sec
```

5.

```sql
-- 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.
DELIMITER $$
CREATE PROCEDURE DeleteOrder(IN p_orderID VARCHAR(6))
BEGIN
    DELETE FROM OrderDetails WHERE OrderID = p_orderID;
    DELETE FROM Orders WHERE OrderID = p_orderID;
END $$
DELIMITER ;
```

```
6  13:02:16  CREATE PROCEDURE DeleteOrder(IN p_orderID VARCHAR(6)) BEGIN  DELETE FROM OrderDetails WH...  0 row(s) affected  0.016 sec
```

6.

```sql
-- 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, OStatus)
values(11, NOW(), 15000.00, 'Pending');
```

```
16  13:16:16  Insert into Orders (CustomerID, OrderDate, TotalAmount, OStatus)  values(11, NOW(), 15000.00, 'Pending')  1 row(s) affected  0.000 sec
```

7.

```sql
-- 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.
DELIMITER $$
Create Procedure updateCustomerInfo(
                IN p_customerID int,
                IN p_email varchar(255),
                IN p_phone varchar(20),
                IN p_address varchar(255)
                )
Begin
    Update customers
    Set email = p_email, phone = p_phone, address = p_address
    Where customerID = p_customerID;
END $$
DELIMITER ;
```

```
2  13:24:59  Create Procedure updateCustomerInfo(IN p_customerID int, IN p_email varchar(255), IN p_phone varchar(20), I...  0 row(s) affected  0.000 sec
```

```sql
call techshop.updateCustomerInfo(1, 'rk123@gmail.com', '8456123480', '123, Cross Road, Bangalore');
```

8.

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

9.

```
-- Write an SQL query to delete all orders and their associated order details
-- for a specific customer from the "Orders" and "OrderDetails" tables.
-- Allow users to input the customer ID as a parameter.
DELIMITER $$
CREATE PROCEDURE DeleteCustomerOrderDetails(IN p_customerID VARCHAR(6))
BEGIN
    DECLARE v_orderID INT;

    SELECT OrderID INTO v_orderID
    FROM Orders
    WHERE customerID = p_customerID;

    DELETE FROM Orders WHERE customerID = p_customerID;
    DELETE FROM OrderDetails WHERE orderID in (v_orderID);
END $$
DELIMITER ;
```

```
4  14:03:41  CREATE PROCEDURE DeleteCustomerOrderDetails(IN p_customerID VARCHAR(6)) BEGIN DECLARE v_ord...  0 row(s) affected  0.015 sec
```

10.

```
-- Write an SQL query to insert a new electronic gadget product
-- into the "Products" table, including product name, category, price, and any other relevant details.
 Insert into Products (ProductName, ProDesc, Price)
 values('VR-Set', 'Gaming and Entertainment', 12000.00);
```

```
13  14:08:41  Insert into Products (ProductName, ProDesc, Price)  values('VR-Set', 'Gaming and Entertainment', 12000.00)  1 row(s) affected  0.000 sec
```

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.
DELIMITER $$
CREATE PROCEDURE updateOrderStatus(IN p_orderID int, IN p_o_status varchar(50))
BEGIN
    update orders
    set ostatus = p_o_status
    where orderID = p_orderID;
END $$
DELIMITER ;
```

```
14  14:12:05  CREATE PROCEDURE updateOrderStatus(IN p_orderID int, IN p_o_status varchar(50)) BEGIN update orders...  0 row(s) affected  0.016 sec
```

12.

```sql
-- 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.
 alter table customers
 add column total_orders int;

 update customers c
 set total_orders = (
     select count(o.orderID)
     from orders o
     where c.customerID = o.customerID
     group by o.customerID
);
```

```
2  14:26:19  update customers c  set total_orders = ( select count(o.orderID) from orders o  where c.customerID = o.custo...  10 row(s) affected Rows matched: 11  Changed: 10  Warnings: 0  0.016 sec
```

Task-3:

1.

```sql
-- Write an SQL query to retrieve a list of all orders along with
-- customer information (e.g., customer name) for each order.

select o.*, concat(c.firstName,' ', c.lastName) as `Customer Name`
from orders o
join customers c on c.customerID = o.customerID;
```

OrderID	CustomerID	OrderDate	TotalAmount	OStatus	Customer Name
1	1	2024-01-13 12:43:58	110000.00	Shipped	Rahul Kumar
2	2	2024-01-13 12:43:58	49500.00	Shipped	Priya Sharma
3	3	2024-01-13 12:43:58	66000.00	Delivered	Vikram Singh
4	4	2024-01-13 12:43:58	51700.00	Pending	Deepika Patel
5	5	2024-01-13 12:43:58	46200.00	Shipped	Amit Verma
6	6	2024-01-13 12:43:58	71500.00	Delivered	Ananya Nair
7	7	2024-01-13 12:43:58	49500.00	Pending	Raj Malhotra
8	8	2024-01-13 12:43:58	66000.00	Shipped	Neha Srivastava
10	10	2024-01-13 12:43:58	46200.00	Pending	Shreya Rajput

2.

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

select p.productName, sum(p.price*od.quantity) as `Total Revenue`
from products p
join orderdetails od on p.productID = od.productID
group by p.productName
order by `Total Revenue` desc;
```

	productName	Total Revenue
▶	LED TV	115500.00
	Laptop	99000.00
	Air Conditioner	66000.00
	Washing Machine	55000.00
	Water Purifier	52800.00
	Refrigerator	44000.00
	Vacuum Cleaner	39600.00
	Camera	38500.00
	Mobile Phone	33000.00
	Microwave Oven	13200.00

3.

```sql
-- Write an SQL query to list all customers who have made at least one purchase.
-- Include their names and contact information.
select concat(firstName,' ', lastName) as `Customer Name`, email, phone
from customers
where total_orders >= 1;
-- OR --
select concat(firstName,' ', lastName) as `Customer Name`, c.Email, c.Phone
from customers c
join orders o on o.customerID = c.customerID
group by o.customerID
having count(o.customerID)>=1;
```

	Customer Name	Email	Phone
▶	Rahul Kumar	rk123@gmail.com	8456123480
	Priya Sharma	priya.sharma@example.com	8765432109
	Vikram Singh	vikram.singh@example.com	7654321098
	Deepika Patel	deepika.patel@example.com	6543210987
	Amit Verma	amit.verma@example.com	5432109876
	Ananya Nair	ananya.nair@example.com	4321098765
	Raj Malhotra	raj.malhotra@example.com	3210987654
	Neha Srivastava	neha.srivastava@example.com	2109876543
	Shreya Rajput	shreya.rajput@example.com	9216544215
	Avinash Dubey	avi.dubey@example.com	9832432687

4.

```sql
-- 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 od.productID, p.productName, sum(od.quantity) as total_orders
from orderdetails od
join products p on od.productID = p.productID
group by od.productID
order by total_orders desc
limit 1;
```

	productID	productName	total_orders
▶	9	Water Purifier	4

5.

```sql
-- Write an SQL query to retrieve a list of electronic gadgets along with their corresponding categories.
select productName, proDesc as Category
from products
order by proDesc;
```

productName	Category
LED TV	Entertainment
Mobile Phone	Entertainment
Laptop	Entertainment
VR-Set	Entertainment
Camera	Photography
Air Conditioner	Plant and Machinery
Washing Machine	Plant and Machinery
Refrigerator	Plant and Machinery
Microwave Oven	Plant and Machinery
Water Purifier	Plant and Machinery
Vacuum Cleaner	Plant and Machinery

6.

```sql
-- 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.customerID, concat(firstName,' ', lastName) as `Customer Name`, round(avg(o.totalAmount),2)
from customers c
join orders o on c.customerID = o.customerID
group by o.customerID;
```

customerID	Customer Name	Average_Spend
1	Rahul Kumar	110000.00
2	Priya Sharma	41250.00
3	Vikram Singh	66000.00
4	Deepika Patel	51700.00
5	Amit Verma	46200.00
6	Ananya Nair	71500.00
7	Raj Malhotra	49500.00
8	Neha Srivastava	66000.00
10	Shreya Rajput	46200.00
11	Avinash Dubey	38500.00

7.

```sql
-- Write an SQL query to find the order with the highest total revenue.
-- Include the order ID, customer information, and the total revenue.
select c.*, totalAmount
from orders o
join customers c on o.customerID = c.customerID
order by totalAmount desc
limit 1;
```

	CustomerID	FirstName	LastName	Email	Phone	Address	total_orders	totalAmount
▶	1	Rahul	Kumar	rk123@gmail.com	8456123480	123, Cross Road, Bangalore	1	110000.00

8.

```sql
-- Write an SQL query to list electronic gadgets and the number of times each product has been ordered.
select p.productname, count(od.productID) as `Count of Orders`
from orderdetails od
join products p on od.productId = p.productID
group by od.productID
order by count(od.productID) desc;
```

	productname	Count of Orders
▶	Air Conditioner	3
	LED TV	2
	Washing Machine	2
	Refrigerator	2
	Mobile Phone	2
	Laptop	2
	Camera	2
	Water Purifier	2
	Vacuum Cleaner	2
	Microwave Oven	1

9.

```sql
-- 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.
delimiter $$
create procedure FindCustomer(IN p_product_name varchar(255))
begin
    select c.*
    from customers c
    join orders o on c.customerID = o.customerID
    join orderdetails od on o.orderID = od.orderID
    join products p on od.productID = p.productID
    where p.ProductName = p_product_name;
end $$
delimiter ;
```

10.

```sql
call techshop.RevenueInBetween('2024-01-11', '2024-01-13');
```

	Total Revenue Generated
▶	121000.00

Task-4:

1.

```
-- Write an SQL query to find out which customers have not placed any orders.
 select c.*
 from customers c
 left join orders o on c.customerID = o.customerID
 where o.customerID is null;
```

	CustomerID	FirstName	LastName	Email	Phone	Address	total_orders
▶	9	Sandeep	Gupta	sandeep.gupta@example.com	1098765432	456, Aundh, Pune	NULL

2.

```
-- Write an SQL query to find the total number of products available for sale.
 select p.*
 from products p
 join inventory i on p.productID = i.productID
 where i.QuantityInStock > 0;
```

	ProductID	ProductName	ProDesc	Price
▶	1	LED TV	Entertainment	38500.00
	2	Air Conditioner	Plant and Machinery	33000.00
	3	Washing Machine	Plant and Machinery	27500.00
	4	Refrigerator	Plant and Machinery	22000.00
	5	Mobile Phone	Entertainment	16500.00
	6	Laptop	Entertainment	49500.00
	7	Microwave Oven	Plant and Machinery	13200.00
	8	Camera	Photography	38500.00
	9	Water Purifier	Plant and Machinery	13200.00
	10	Vacuum Cleaner	Plant and Machinery	19800.00

3.

```
-- Write an SQL query to find the total number of products available for sale.
 select p.*
 from products p
 join inventory i on p.productID = i.productID
 where i.QuantityInStock > 0;
```

	ProductID	ProductName	ProDesc	Price
▶	1	LED TV	Entertainment	38500.00
	2	Air Conditioner	Plant and Machinery	33000.00
	3	Washing Machine	Plant and Machinery	27500.00
	4	Refrigerator	Plant and Machinery	22000.00
	5	Mobile Phone	Entertainment	16500.00
	6	Laptop	Entertainment	49500.00
	7	Microwave Oven	Plant and Machinery	13200.00
	8	Camera	Photography	38500.00
	9	Water Purifier	Plant and Machinery	13200.00
	10	Vacuum Cleaner	Plant and Machinery	19800.00

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.
Delimiter $$
Create procedure AverageOrderQuantity(IN p_category_name varchar(255))
    Begin
        select proDesc as Category, avg(od.quantity) as `Average Orders`
        from products p
        join orderdetails od on p.productId = od.productID
        group by proDesc
        having proDesc = p_category_name;
    End $$
Delimiter ;
```

```
call techshop.AverageOrderQuantity('Entertainment');
```

	Category	Average Orders
▶	Entertainment	1.1667

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.
delimiter $$
create procedure RevenueByCustomer(IN p_customer_id int)
begin
    select sum(TotalAmount) as `Total Revenue`
    from orders
    where customerID = p_customer_id
    group by customerID;
end $$
delimiter ;

 call techshop.RevenueByCustomer(4);
```

	p_customer_id	Total Revenue
▶	4	51700.00

6.

```sql
-- 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 concat(firstName, ' ', lastname) as `Full Name`, total_orders
from customers
where total_orders = (select count(o.customerID) as total_orders
            from orders o
            group by o.customerID
            order by total_orders desc
            limit 1
);
```

	Full Name	total_orders
▶	Priya Sharma	2

7.

```sql
-- Write an SQL query to find the most popular product category,
-- which is the one with the highest total quantity ordered across all orders.
With cte as (
    select p.proDesc, sum(od.quantity) as ordered_quantity
    from orderdetails od
    join products p on od.productID = p.productID
    group by p.proDesc
)
select proDesc, ordered_quantity
from cte
where ordered_quantity =
            (select max(ordered_quantity)
            from cte);
```

	proDesc	ordered_quantity
▶	Plant and Machinery	14

8.

```sql
-- 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 o.customerID, p.proDesc, sum(od.quantity*p.price) as `Total Price`
from orders o
join orderdetails od on o.orderID = od.orderID
join products p on od.productID = p.productID
group by p.proDesc, o.customerID
having proDesc = 'Entertainment';
```

	customerID	proDesc	Total Price
▶	1	Entertainment	77000.00
	6	Entertainment	38500.00
	3	Entertainment	66000.00
	8	Entertainment	66000.00

9.

```sql
-- Write an SQL query to calculate the average order value
-- (total revenue divided by the number of orders) for all customers.
select CustomerID, round(avg(totalamount), 2) as `Average value`
from orders
group by customerID;
```

	CustomerID	Average value
▶	1	110000.00
	2	41250.00
	3	66000.00
	4	51700.00
	5	46200.00
	6	71500.00
	7	49500.00
	8	66000.00
	10	46200.00
	11	38500.00

10.

```sql
-- 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 o.customerID, concat(c.firstname, ' ', c.lastname) as `Full Name`, count(o.customerID) as `Total Orders`
from orders o
join customers c on o.customerID = c.customerID
group by o.customerID;
```

	customerID	Full Name	Total Orders
▶	1	Rahul Kumar	1
	2	Priya Sharma	2
	3	Vikram Singh	1
	4	Deepika Patel	1
	5	Amit Verma	1
	6	Ananya Nair	1
	7	Raj Malhotra	1
	8	Neha Srivastava	1
	10	Shreya Rajput	1
	11	Avinash Dubey	1