Online Shopping System
1. Customers (CustomerID, Name, Email, Phone, Address)
2. Products (ProductID, Name, Category, Price, StockQuantity)
3. Orders (OrderID, CustomerID, OrderDate, TotalAmount)
4. OrderDetails (OrderDetailID, OrderID, ProductID, Quantity, Subtotal)
Write queries for the following questions:
1. Create a Payments table with PaymentID, OrderID (FK), PaymentDate, AmountPaid, and
PaymentMethod.
2. Update the stock quantity of a product after an order is placed.
3. Retrieve Customer Name, Order Date, and TotalAmount for orders where the total amount
exceeds 5000.
4. Calculate the total sales per product category.
5. Find the top 5 customers who have spent the most on orders.

```
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(100),
    Email VARCHAR(100),
    Phone VARCHAR(15),
    Address VARCHAR(255)
);


CREATE TABLE Products (
    ProductID INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(100),
    Category VARCHAR(50),
    Price DECIMAL(10,2),
    StockQuantity INT
);

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


CREATE TABLE OrderDetails (
    OrderDetailID INT PRIMARY KEY AUTO_INCREMENT,
    OrderID INT,
    ProductID INT,
    Quantity INT,
```

```
    Subtotal DECIMAL(10,2),
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);

-- Customers
INSERT INTO Customers (Name, Email, Phone, Address) VALUES
('Amit', 'amit@example.com', '9876543210', 'Delhi'),
('Sneha', 'sneha@example.com', '9876543211', 'Mumbai'),
('Rahul', 'rahul@example.com', '9876543212', 'Bangalore'),
('Priya', 'priya@example.com', '9876543213', 'Chennai');

-- Products
INSERT INTO Products (Name, Category, Price, StockQuantity) VALUES
('iPhone 14', 'Electronics', 75000, 10),
('MacBook Air', 'Electronics', 95000, 5),
('T-Shirt', 'Clothing', 800, 50),
('Jeans', 'Clothing', 1500, 40),
('Coffee Mug', 'Kitchenware', 250, 100);

-- Orders
INSERT INTO Orders (CustomerID, OrderDate, TotalAmount) VALUES
(1, '2024-03-01', 150000),
(2, '2024-03-05', 6000),
(3, '2024-03-10', 1200),
(4, '2024-03-15', 300);

-- OrderDetails
INSERT INTO OrderDetails (OrderID, ProductID, Quantity, Subtotal) VALUES
(1, 1, 1, 75000),
(1, 2, 1, 95000),
(2, 3, 5, 4000),
(2, 4, 2, 3000),
(3, 5, 4, 1000),
(4, 5, 1, 250);

-- 1. Create a Payments table with PaymentID, OrderID (FK), PaymentDate,
AmountPaid, and PaymentMethod.
CREATE TABLE Payments (
    PaymentID INT PRIMARY KEY AUTO_INCREMENT,
    OrderID INT,
    PaymentDate DATE,
    AmountPaid DECIMAL(10,2),
    PaymentMethod VARCHAR(50),
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)
);

-- Payments
INSERT INTO Payments (OrderID, PaymentDate, AmountPaid, PaymentMethod) VALUES
(1, '2024-03-01', 150000, 'Card'),
```

```
(2, '2024-03-05', 6000, 'UPI'),
(3, '2024-03-10', 1200, 'Cash'),
(4, '2024-03-15', 300, 'Card');

-- 2. Update Stock Quantity of a Product After an Order is Placed
-- Reduce quantity for ProductID = 1 by 1 unit
UPDATE Products
SET StockQuantity = StockQuantity - 1
WHERE ProductID = 1;

-- 3. Retrieve Customer Name, Order Date, and TotalAmount where TotalAmount > 5000
SELECT c.Name, o.OrderDate, o.TotalAmount
FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
WHERE o.TotalAmount > 5000;

-- 4. Calculate Total Sales per Product Category
SELECT p.Category, SUM(od.SubTotal) AS TotalSales
FROM OrderDetails od
JOIN Products p ON p.ProductID = od.ProductID
GROUP BY p.category;

-- 5. Top 5 Customers Who Spent the Most on Orders
SELECT c.Name, SUM(o.TotalAmount) AS TotalSpent
FROM Orders o
JOIN Customers c ON o.CustomerID = o.CustomerID
GROUP BY c.CustomerID
ORDER BY TotalSpent DESC
LIMIT 5;
```