```sql
CREATE DATABASE ECommerceDB;

USE ECommerceDB;

CREATE TABLE Categories (
    CategoryID INT PRIMARY KEY,
    CategoryName VARCHAR(50) NOT NULL UNIQUE
);

CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(100) NOT NULL UNIQUE,
    CategoryID INT,
    Price DECIMAL(10,2) NOT NULL,
    StockQuantity INT,
    FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID)
);

CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY,
    CustomerName VARCHAR(100) NOT NULL,
    Email VARCHAR(100) UNIQUE,
    JoinDate DATE
);

CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE NOT NULL,
    TotalAmount DECIMAL(10,2),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);

INSERT INTO Categories (CategoryID, CategoryName) VALUES
(1, 'Electronics'),
(2, 'Books'),
(3, 'Home Goods'),
(4, 'Apparel');

INSERT INTO Products (ProductID, ProductName, CategoryID, Price,
StockQuantity) VALUES
(101, 'Laptop Pro', 1, 1200.00, 50),
(102, 'SQL Handbook', 2, 45.50, 200),
(103, 'Smart Speaker', 1, 99.99, 150),
(104, 'Coffee Maker', 3, 75.00, 80),
(105, 'Novel : The Great SQL', 2, 25.00, 120),
(106, 'Wireless Earbuds', 1, 150.00, 100),
(107, 'Blender X', 3, 120.00, 60),
(108, 'T-Shirt Casual', 4, 20.00, 300);

INSERT INTO Customers (CustomerID, CustomerName, Email, JoinDate)
VALUES
(1, 'Alice Wonderland', 'alice@example.com', '2023-01-10'),
(2, 'Bob the Builder', 'bob@example.com', '2022-11-25'),
(3, 'Charlie Chaplin', 'charlie@example.com', '2023-03-01'),
(4, 'Diana Prince', 'diana@example.com', '2021-04-26');

INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES
```

```
(1001, 1, '2023-04-26', 1245.50),
(1002, 2, '2023-10-12', 99.99),
(1003, 1, '2023-07-01', 145.00),
(1004, 3, '2023-01-14', 150.00),
(1005, 2, '2023-09-24', 120.00),
(1006, 1, '2023-06-19', 20.00);

-- Question 7 : Generate a report showing CustomerName, Email, and the
-- TotalNumberofOrders for each customer. Include customers who have
not placed
-- any orders, in which case their TotalNumberofOrders should be 0.
Order the results
-- by CustomerName.

SELECT
    c.CustomerName,
    c.Email,
    COUNT(o.OrderID) AS TotalNumberOfOrders
FROM Customers c
LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
GROUP BY c.CustomerID, c.CustomerName, c.Email
ORDER BY c.CustomerName;

-- Retrieve Product Information with Category: Write a SQL query to
-- display the ProductName, Price, StockQuantity, and CategoryName for
all
-- products. Order the results by CategoryName and then ProductName
alphabetically

SELECT
    p.ProductName,
    p.Price,
    p.StockQuantity,
    c.CategoryName
FROM Products p
JOIN Categories c ON p.CategoryID = c.CategoryID
ORDER BY c.CategoryName, p.ProductName;

-- Write a SQL query that uses a Common Table Expression (CTE) and a
-- Window Function (specifically ROW_NUMBER() or RANK()) to display the
-- CategoryName, ProductName, and Price for the top 2 most expensive
products in
-- each CategoryName

WITH RankedProducts AS (
    SELECT
        c.CategoryName,
        p.ProductName,
        p.Price,
        ROW_NUMBER() OVER (PARTITION BY c.CategoryName ORDER BY p.Price
DESC) AS RowNum
    FROM Products p
    JOIN Categories c ON p.CategoryID = c.CategoryID
)
SELECT
    CategoryName,
    ProductName,
    Price
```

```
FROM RankedProducts
WHERE RowNum <= 2;

-- Question 10 : You are hired as a data analyst by Sakila Video
Rentals, a global movie
-- rental company. The management team is looking to improve decision-
making by
-- analyzing existing customer, rental, and inventory data.
-- Using the Sakila database, answer the following business questions
to support key strategic
-- initiatives.
-- Tasks & Questions:
-- 1. Identify the top 5 customers based on the total amount theyâ□□ve
spent. Include customer
-- name, email, and total amount spent.
-- 2. Which 3 movie categories have the highest rental counts? Display
the category name
-- and number of times movies from that category were rented.
-- 3. Calculate how many films are available at each store and how many
of those have
-- never been rented.
-- 4. Show the total revenue per month for the year 2023 to analyze
business seasonality.
-- 5. Identify customers who have rented more than 10 times in the last
6 months


-- 1. Top 5 customers based on total amount spent

SELECT
    c.first_name AS CustomerName,
    c.email AS Email,
    SUM(p.amount) AS TotalAmountSpent
FROM payment p
JOIN customer c ON p.customer_id = c.customer_id
GROUP BY c.customer_id, c.first_name, c.email
ORDER BY TotalAmountSpent DESC
LIMIT 5;


-- 2. Top 3 movie categories with highest rental counts

SELECT
    cat.name AS CategoryName,
    COUNT(r.rental_id) AS RentalCount
FROM rental r
JOIN inventory i ON r.inventory_id = i.inventory_id
JOIN film f ON i.film_id = f.film_id
JOIN film_category fc ON f.film_id = fc.film_id
JOIN category cat ON fc.category_id = cat.category_id
GROUP BY cat.category_id, cat.name
ORDER BY RentalCount DESC
LIMIT 3;


-- 3. Films available at each store and how many have never been rented

SELECT
```

```
    s.store_id,
    COUNT(DISTINCT i.film_id) AS TotalFilms,
    COUNT(DISTINCT i.film_id) - COUNT(DISTINCT r.inventory_id) AS
NeverRentedFilms
FROM store s
JOIN inventory i ON s.store_id = i.store_id
LEFT JOIN rental r ON i.inventory_id = r.inventory_id
GROUP BY s.store_id;


-- 4. Total revenue per month for the year 2023

SELECT
    DATE_FORMAT(p.payment_date, '%Y-%m') AS Month,
    SUM(p.amount) AS TotalRevenue
FROM payment p
WHERE YEAR(p.payment_date) = 2023
GROUP BY DATE_FORMAT(p.payment_date, '%Y-%m')
ORDER BY Month;


-- 5. Customers who rented more than 10 times in the last 6 months

SELECT
    c.first_name AS CustomerName,
    c.email AS Email,
    COUNT(r.rental_id) AS TotalRentals
FROM rental r
JOIN customer c ON r.customer_id = c.customer_id
WHERE r.rental_date >= DATE_SUB(CURDATE(), INTERVAL 6 MONTH)
GROUP BY c.customer_id, c.first_name, c.email
HAVING COUNT(r.rental_id) > 10
ORDER BY TotalRentals DESC;
```