SQL Code

```sql
create database ECommerceDB;
use ECommerceDB;



## Question 6 :  Create a database named ECommerceDB and perform the following  tasks:


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
values(1,'Electonics'),(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'),
```

```sql
(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 CustomerName , Email , count(Orders.CustomerID) as TotalNumberofOrders
from Customers left join Orders
on
Customers.CustomerID = Orders.CustomerID
group by Customers.CustomerID
ORDER BY Customers.CustomerName;

-- Question 8 :  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 ProductName , Price,StockQuantity ,Categories.CategoryName
from Products join Categories
on
Products.CategoryID = Categories.CategoryID
order by
Categories.CategoryName ASC,
Products.ProductName ASC;


-- Question 9 : 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 rn
    FROM Products p
    JOIN Categories c
```

```sql
    ON p.CategoryID = c.CategoryID
)

SELECT
    CategoryName,
    ProductName,
    Price
FROM RankedProducts
WHERE rn <= 2
ORDER BY CategoryName, Price DESC;



-- 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.

create database sakila;
use sakila;

select * from `sakila.actor`;
select * from `sakila.address`;
select * from `sakila.category`;
select * from `sakila.city`;
select * from `sakila.complete_add`;
select * from `sakila.country`;
select * from `sakila.customer`;
select * from `sakila.film (1)`;
select * from `sakila.payment`;
select * from `sakila.rental (1)`;

-- 1 . Identify the top 5 customers based on the total amount they’ve spent. Include customer
-- name, email, and total amount spent.

select first_name , last_name , email , round(sum(amount),2) as totalAmount
from `sakila.customer` join `sakila.payment`
on
`sakila.customer`.customer_id = `sakila.payment`.customer_id
group by `sakila.customer`.customer_id ,
`sakila.customer`.first_name,`sakila.customer`.last_name,`sakila.customer`.email
order by totalAmount desc
limit 5;

-- 2.Which 3 movie categories have the highest rental counts? Display the category name
-- and number of times movies from that category were rented.
# -> Since there is no relationship between rental and category tables, the question cannot be solved using the given schema.

```

```sql


-- 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.

select month(payment_date) as month,
round(sum(amount),2) as revenue
from `sakila.payment`
where year(payment_date) = 2005
group by month(payment_date);


-- 5. Identify customers who have rented more than 10 times in the last 6 months.

with new_table as
(select customer_id,count(customer_id) as count
from `sakila.rental (1)`
group by customer_id
having count(customer_id) > 10)

select first_name , last_name , count
from `sakila.customer` join new_table
on `sakila.customer`.customer_id = new_table.customer_id;



```