Sql Assignment

Question 1: Explain the fundamental differences between DDL, DML, and DQL commands in SQL. Provide one example for each type of command.

Answer:
DDL (Data Definition Language) commands define and modify the structure of database objects, such as tables and schemas.
Example: CREATE TABLE Students (ID INT PRIMARY KEY, Name VARCHAR(50));
DML (Data Manipulation Language) commands manipulate data within tables. Example: INSERT INTO Students (ID, Name) VALUES (1, 'Alice');
DQL (Data Query Language) commands query information from the database. Example: SELECT * FROM Students;

Question 2 : What is the purpose of SQL constraints? Name and describe three common types of constraints, providing a simple scenario where each would be useful.

Answer:
The purpose of SQL constraints is to enforce rules at the table level, ensuring data integrity and consistency.

- PRIMARY KEY: Ensures each record in a table is unique. Scenario: Student ID in a Students table.
- UNIQUE: Ensures all values in a column are different. Scenario: Email addresses in a user table.
- FOREIGN KEY: Ensures the referential integrity of data between tables. Scenario: CustomerID in Orders referencing Customers.

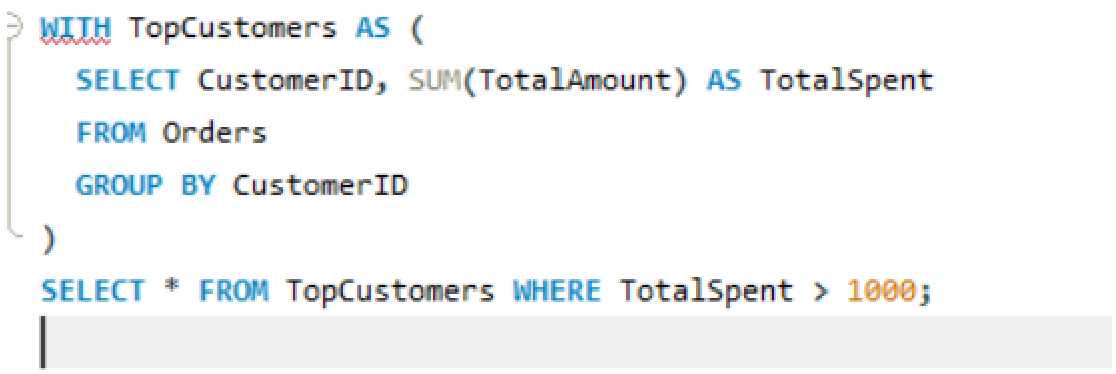
Question 3 : Explain the difference between LIMIT and OFFSET clauses in SQL. How would you use them together to retrieve the third page of results, assuming each page has 10 records?

Answer:
LIMIT specifies the maximum number of records to return, while OFFSET skips a specified number of records before returning results. For the third page: SELECT * FROM table_name LIMIT 10 OFFSET 20; (This skips the first 20 records, showing the next 10)

Question 4 : What is a Common Table Expression (CTE) in SQL, and what are its main benefits? Provide a simple SQL example demonstrating its usage.

Answer:
A CTE is a temporary result set defined within the execution scope of a single SELECT, INSERT, UPDATE, or DELETE statement, improving readability and organization for complex queries.
Example:

sql



```
WITH TopCustomers AS (
  SELECT CustomerID, SUM(TotalAmount) AS TotalSpent
  FROM Orders
  GROUP BY CustomerID
)
SELECT * FROM TopCustomers WHERE TotalSpent > 1000;
```

Question 5 : Describe the concept of SQL Normalization and its primary goals. Briefly explain the first three normal forms (1NF, 2NF, 3NF).

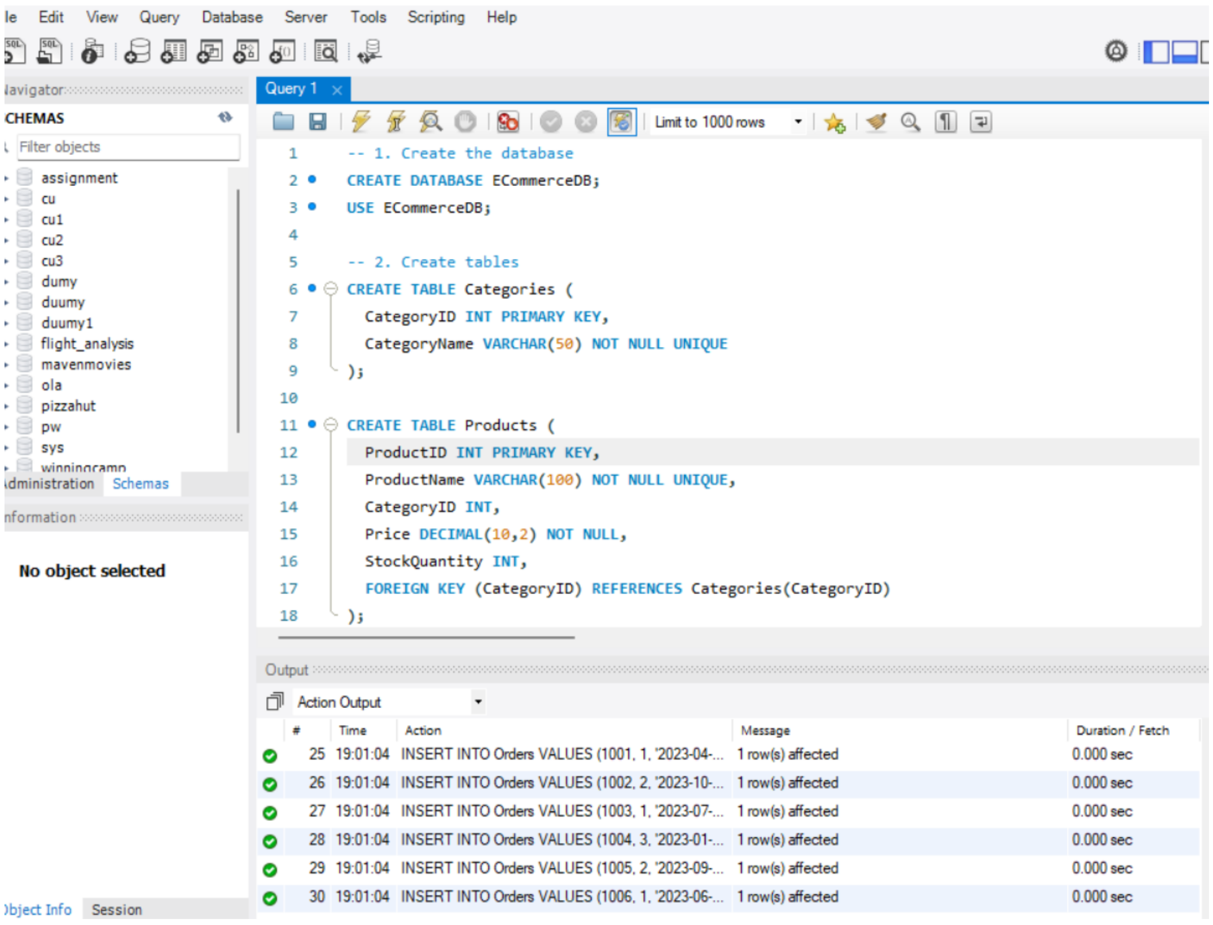
Answer:

Normalization organizes data to reduce redundancy and improve integrity.

- 1NF: Eliminate repeating groups by ensuring each field contains only atomic values.
- 2NF: Ensure all non-key attributes are fully functional on the primary key.
- 3NF: Remove transitive dependencies to ensure non-key attributes depend only on the primary key.

Question 6: Create a database named ECommerceDB and perform the following: (1) Create tables as per specification, (2) Insert provided records.
Answer:

U can find the query practical sheet

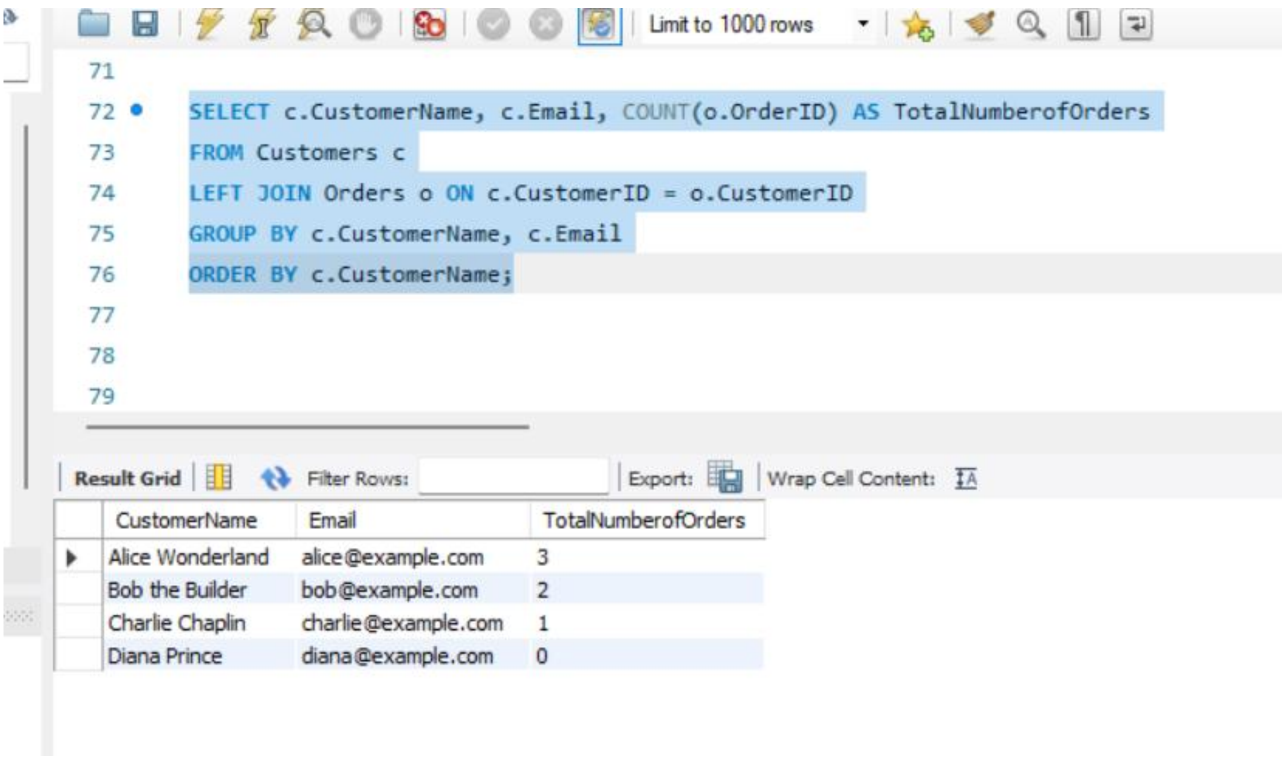


```sql
-- 1. Create the database
CREATE DATABASE ECommerceDB;
USE ECommerceDB;

-- 2. Create tables
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)
);
```

```
    #   Time      Action                                          Message             Duration / Fetch
✓  25  19:01:04  INSERT INTO Orders VALUES (1001, 1, '2023-04-...  1 row(s) affected   0.000 sec
✓  26  19:01:04  INSERT INTO Orders VALUES (1002, 2, '2023-10-...  1 row(s) affected   0.000 sec
✓  27  19:01:04  INSERT INTO Orders VALUES (1003, 1, '2023-07-...  1 row(s) affected   0.000 sec
✓  28  19:01:04  INSERT INTO Orders VALUES (1004, 3, '2023-01-...  1 row(s) affected   0.000 sec
✓  29  19:01:04  INSERT INTO Orders VALUES (1005, 2, '2023-09-...  1 row(s) affected   0.000 sec
✓  30  19:01:04  INSERT INTO Orders VALUES (1006, 1, '2023-06-...  1 row(s) affected   0.000 sec
```

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 by CustomerName.

Answer:

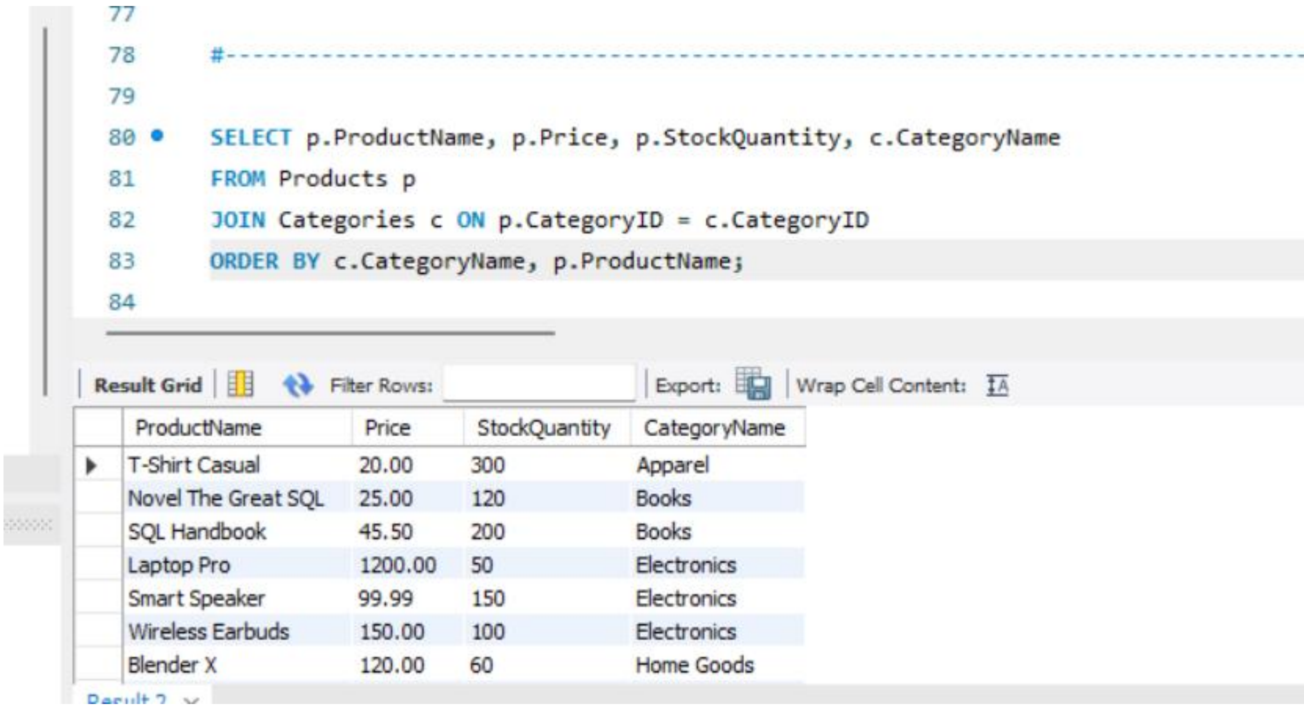


```sql
SELECT c.CustomerName, c.Email, COUNT(o.OrderID) AS TotalNumberofOrders
FROM Customers c
LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
GROUP BY c.CustomerName, c.Email
ORDER BY c.CustomerName;
```

CustomerName	Email	TotalNumberofOrders
Alice Wonderland	alice@example.com	3
Bob the Builder	bob@example.com	2
Charlie Chaplin	charlie@example.com	1
Diana Prince	diana@example.com	0

Question 8: Retrieve Product Information with Category. Display ProductName, Price, StockQuantity, and CategoryName for all products. Order by CategoryName then ProductName.

Answer:



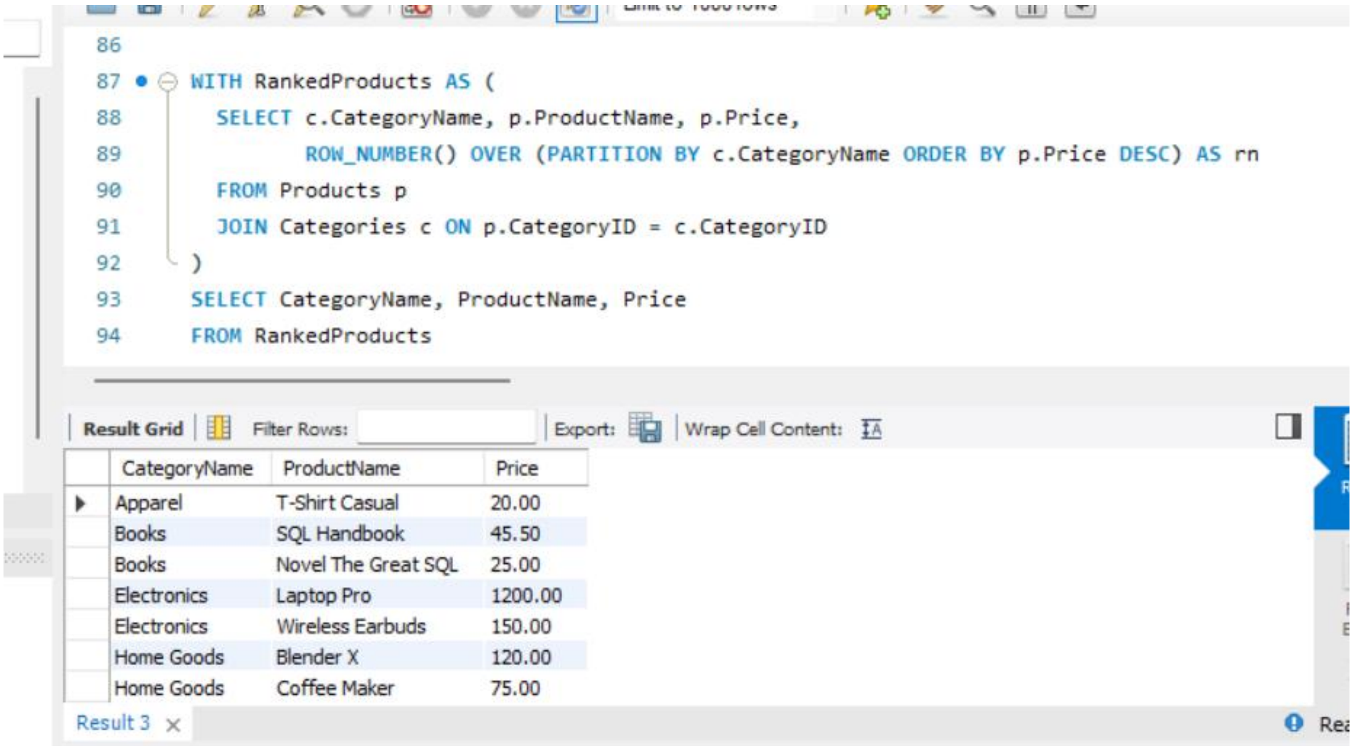
```
#--------------------------------------------------------------------------------

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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ProductName	Price	StockQuantity	CategoryName
T-Shirt Casual	20.00	300	Apparel
Novel The Great SQL	25.00	120	Books
SQL Handbook	45.50	200	Books
Laptop Pro	1200.00	50	Electronics
Smart Speaker	99.99	150	Electronics
Wireless Earbuds	150.00	100	Electronics
Blender X	120.00	60	Home Goods

Question 9: Use a CTE and a window function (ROW_NUMBER or RANK) to display CategoryName, ProductName, and Price for the top 2 most expensive products in each CategoryName.

Answer:



```
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 ON p.CategoryID = c.CategoryID
)
SELECT CategoryName, ProductName, Price
FROM RankedProducts
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

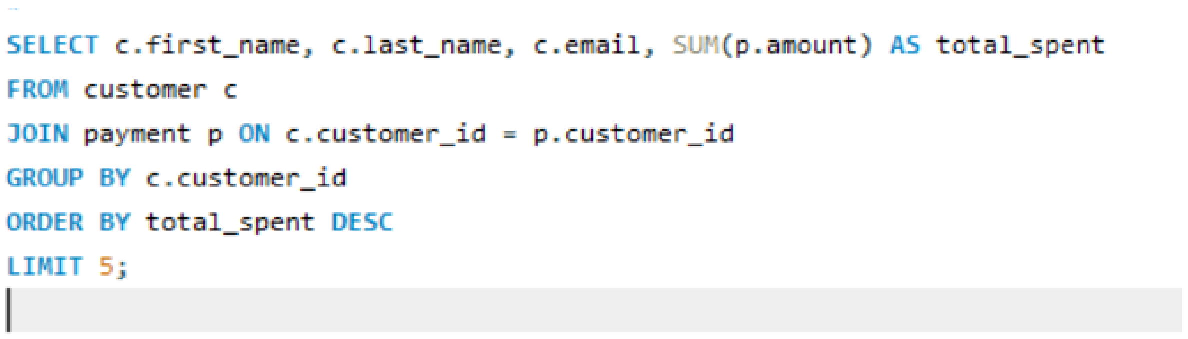
CategoryName	ProductName	Price
Apparel	T-Shirt Casual	20.00
Books	SQL Handbook	45.50
Books	Novel The Great SQL	25.00
Electronics	Laptop Pro	1200.00
Electronics	Wireless Earbuds	150.00
Home Goods	Blender X	120.00
Home Goods	Coffee Maker	75.00

Result 3

Question 10: Data analyst business SQL questions for the Sakila database.

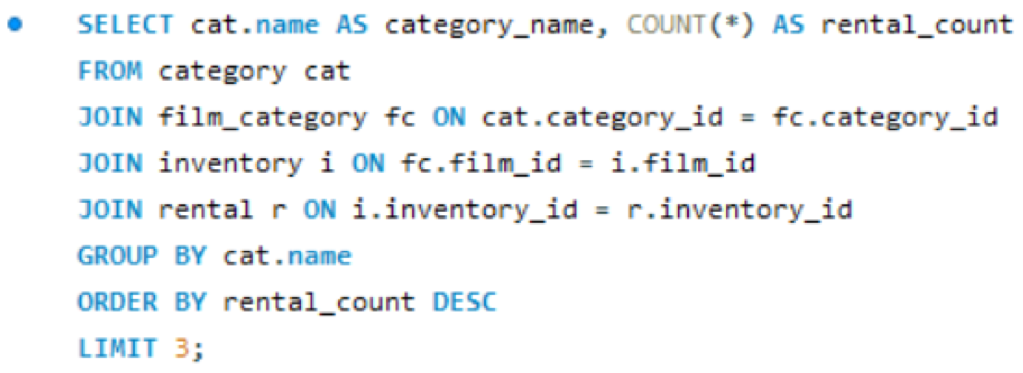
Answer

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



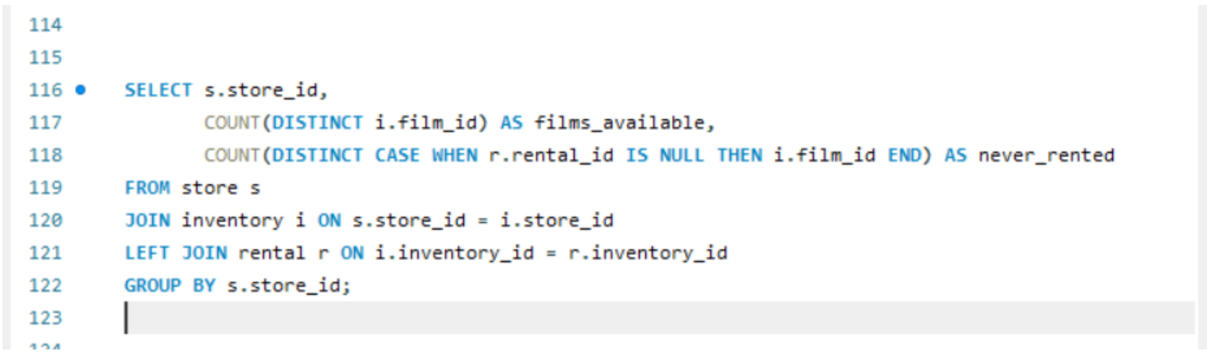
```sql
SELECT c.first_name, c.last_name, c.email, SUM(p.amount) AS total_spent
FROM customer c
JOIN payment p ON c.customer_id = p.customer_id
GROUP BY c.customer_id
ORDER BY total_spent DESC
LIMIT 5;
```

2. Which 3 movie categories have the highest rental counts? Display the category name and number of times movies from that category were rented.



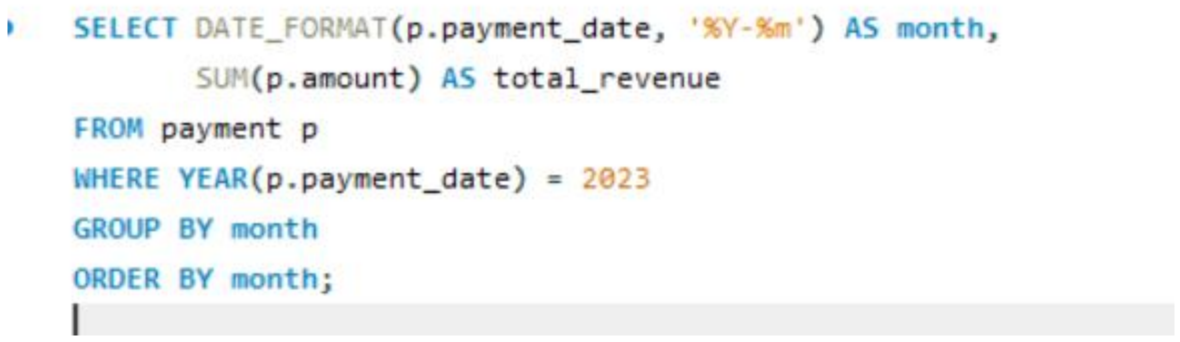
```sql
SELECT cat.name AS category_name, COUNT(*) AS rental_count
FROM category cat
JOIN film_category fc ON cat.category_id = fc.category_id
JOIN inventory i ON fc.film_id = i.film_id
JOIN rental r ON i.inventory_id = r.inventory_id
GROUP BY cat.name
ORDER BY rental_count DESC
LIMIT 3;
```

3. Calculate how many films are available at each store and how many of those have never been rented.



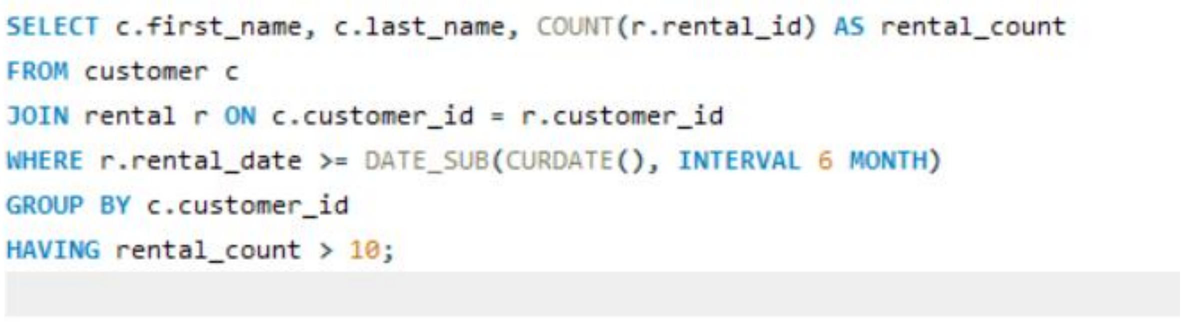
```sql
SELECT s.store_id,
       COUNT(DISTINCT i.film_id) AS films_available,
       COUNT(DISTINCT CASE WHEN r.rental_id IS NULL THEN i.film_id END) AS never_rented
FROM store s
JOIN inventory i ON s.store_id = i.store_id
LEFT JOIN rental r ON i.inventory_id = r.inventory_id
GROUP BY s.store_id;
```

4. Show the total revenue per month for the year 2023 to analyze business seasonality.



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

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



```sql
SELECT c.first_name, c.last_name, COUNT(r.rental_id) AS rental_count
FROM customer c
JOIN rental r ON c.customer_id = r.customer_id
WHERE r.rental_date >= DATE_SUB(CURDATE(), INTERVAL 6 MONTH)
GROUP BY c.customer_id
HAVING rental_count > 10;
```