```sql
-- Basic Aggregate Functions:

-- Question 1: Retrieve the total number of rentals made in the Sakila database. Hint: Use the
COUNT() function.

use sakila;

select count(rental_id) AS total_no_of_rentals  from rental;


-- Question 2: Find the average rental duration (in days) of movies rented from the Sakila database.
Hint: Utilize the AVG() function.

SELECT AVG(DATEDIFF(return_date, rental_date)) AS average_duration_in_days

FROM rental;


-- String Functions:

-- Question 3: Display the first name and last name of customers in uppercase. Hint: Use the UPPER ()
function.

use mavenmovies;

SELECT UPPER(first_name) AS upper_first_name, UPPER(last_name) AS upper_last_name

FROM actor;


-- Question 4:Extract the month from the rental date and display it alongside the rental ID. Hint:
Employ the MONTH() function.

SELECT rental_id, MONTH(rental_date) AS rental_month

FROM rental;


-- GROUP BY:

-- Question 5 Retrieve the count of rentals for each customer (display customer ID and the count of
rentals).

-- Hint: Use COUNT () in conjunction with GROUP BY.

SELECT c.customer_id, COUNT(r.rental_id) AS rental_count

FROM customer c

LEFT JOIN rental r ON c.customer_id = r.customer_id

GROUP BY c.customer_id;


-- Question 6:
```

```
-- Find the total revenue generated by each store.
-- Hint: Combine SUM() and GROUP BY.

SELECT s.store_id, SUM(p.amount) AS total_revenue
FROM store s
JOIN staff st ON s.store_id = st.store_id
JOIN payment p ON st.staff_id = p.staff_id
GROUP BY s.store_id;

-- Question 7
-- Joins
-- Display the title of the movie, customer s first name, and last name who rented it.
-- Hint: Use JOIN between the film, inventory, rental, and customer tables.

SELECT f.title AS movie_title, c.first_name, c.last_name
FROM film f
JOIN inventory i ON f.film_id = i.film_id
JOIN rental r ON i.inventory_id = r.inventory_id
JOIN customer c ON r.customer_id = c.customer_id;

-- Question 8:
-- Retrieve the names of all actors who have appeared in the film "Gone with the Wind."
-- Hint: Use JOIN between the film actor, film, and actor tables.
SELECT a.first_name, a.last_name
FROM actor a
JOIN film_actor fa ON a.actor_id = fa.actor_id
JOIN film f ON fa.film_id = f.film_id
WHERE f.title = 'Gone with the Wind';

-- Here I am ghetting the empty output so to cross verify I have given like this so no movie was
named like that
```

```
SELECT title
FROM film
WHERE title = 'Gone with the Wind';

-- GROUP BY:
-- Question 1:
-- Determine the total number of rentals for each category of movies.
-- Hint: JOIN film_category, film, and rental tables, then use cOUNT () and GROUP BY.
SELECT fc.category_id, COUNT(r.rental_id) AS rental_count
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
GROUP BY fc.category_id;

-- Question 2:
-- Find the average rental rate of movies in each language.
-- Hint: JOIN film and language tables, then use AVG () and GROUP BY.
SELECT l.name AS language, AVG(f.rental_rate) AS avg_rental_rate
FROM film f
JOIN language l ON f.language_id = l.language_id
GROUP BY l.name;

SELECT *
FROM language;
SELECT l.name AS language, AVG(f.rental_rate) AS avg_rental_rate
FROM film f
JOIN language l ON f.language_id = l.language_id
GROUP BY l.name;

SELECT l.name AS language, AVG(f.rental_rate) AS avg_rental_rate
```

```
FROM film f
RIGHT JOIN (
   SELECT language_id, AVG(rental_rate) AS avg_rate
   FROM film
   GROUP BY language_id
) AS avg_table ON f.language_id = avg_table.language_id
JOIN language l ON f.language_id = l.language_id
GROUP BY l.name;



-- Joins
-- Retrieve the customer names along with the total amount they've spent on rentals.
-- Hint: JOIN customer, payment, and rental tables, then use SUM() and GROUP BY.
SELECT c.first_name, c.last_name, SUM(p.amount) AS total_amount_spent
FROM customer c
JOIN payment p ON c.customer_id = p.customer_id
JOIN rental r ON c.customer_id = r.customer_id
GROUP BY c.customer_id;

-- Question 4:
-- List the titles of movies rented by each customer in a particular city (e.g., 'London').
-- Hint: JOIN customer, address, city, rental, inventory, and film tables, then use GROUP BY.

SELECT c.first_name, c.last_name, f.title AS rented_movie_title
FROM customer c
JOIN address a ON c.address_id = a.address_id
JOIN city ci ON a.city_id = ci.city_id
JOIN rental r ON c.customer_id = r.customer_id
JOIN inventory i ON r.inventory_id = i.inventory_id
JOIN film f ON i.film_id = f.film_id
```

```sql
WHERE ci.city = 'London'

ORDER BY c.first_name, c.last_name, f.title;


-- Advanced Joins and GROUP BY:

-- Question 5:

-- Display the top 5 rented movies along with the number of times they've been rented.

-- Hint: JOIN film, inventory, and rental tables, then use cOUNT() and GROUP BY, and limit the results.

SELECT f.title AS movie_title, COUNT(*) AS rental_count

FROM film f

JOIN inventory i ON f.film_id = i.film_id

JOIN rental r ON i.inventory_id = r.inventory_id

GROUP BY f.title

ORDER BY rental_count DESC

LIMIT 5;


-- Question 6:

-- Determine the customers who have rented movies from both stores (store ID 1 and store ID 2).

-- Hint: Use JOINS with rental, inventory, and customer tables and consider COUNT() and GROUP BY.


SELECT c.customer_id, c.first_name, c.last_name

FROM customer c

JOIN rental r ON c.customer_id = r.customer_id

JOIN inventory i ON r.inventory_id = i.inventory_id

JOIN store s ON i.store_id = s.store_id

WHERE s.store_id IN (1, 2)

GROUP BY c.customer_id, c.first_name, c.last_name

HAVING COUNT(DISTINCT s.store_id) = 2;
```