1. \*\*Rank the customers based on the total amount they've spent on rentals:\*\*

SELECT customer\_id, CONCAT(first\_name, ' ', last\_name) AS customer\_name,

       amount, RANK() OVER (ORDER BY amount DESC) AS rank\_by\_amount

FROM (

    SELECT c.customer\_id, c.first\_name, c.last\_name, SUM(p.amount) AS amount

    FROM customer c

    JOIN payment p ON c.customer\_id = p.customer\_id

    GROUP BY c.customer\_id

) AS total\_amount\_per\_customer;

2. \*\*Calculate the cumulative revenue generated by each film over time.\*\*

SELECT film\_id, title, rental\_date, amount,

       SUM(amount) OVER (PARTITION BY film\_id ORDER BY rental\_date) AS cumulative\_revenue

FROM (

    SELECT f.film\_id, f.title, r.rental\_date, p.amount

    FROM film f

    JOIN inventory i ON f.film\_id = i.film\_id

    JOIN rental r ON i.inventory\_id = r.inventory\_id

    JOIN payment p ON r.rental\_id = p.rental\_id

) AS film\_revenue;

3. \*\*Determine the average rental duration for each film, considering films with similar lengths:\*\*

SELECT film\_id, title, rental\_duration,

       AVG(rental\_duration) OVER (PARTITION BY rental\_duration) AS avg\_rental\_duration

FROM film;

4. \*\*Identify the top 3 films in each category based on their rental counts:\*\*

SELECT category\_id, film\_id, title, rental\_count,

       RANK() OVER (PARTITION BY category\_id ORDER BY rental\_count DESC) AS rank\_in\_category

FROM (

    SELECT fc.category\_id, f.film\_id, f.title, COUNT(r.rental\_id) AS rental\_count

    FROM film f

    JOIN film\_category fc ON f.film\_id = fc.film\_id

    JOIN inventory i ON f.film\_id = i.film\_id

    JOIN rental r ON i.inventory\_id = r.inventory\_id

    GROUP BY fc.category\_id, f.film\_id, f.title

) AS category\_film\_rentals

WHERE rank\_in\_category <= 3;

5. \*\*Calculate the difference in rental counts between each customer's total rentals and the average rentals across all customers:\*\*

SELECT customer\_id, CONCAT(first\_name, ' ', last\_name) AS customer\_name,

       rental\_count,

       rental\_count - AVG(rental\_count) OVER () AS diff\_from\_avg\_rentals

FROM (

    SELECT c.customer\_id, c.first\_name, c.last\_name, 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

) AS customer\_rentals;

6. \*\*Find the monthly revenue trend for the entire rental store over time:\*\*

SELECT DATE\_FORMAT(payment\_date, '%Y-%m') AS payment\_month,

       SUM(amount) OVER (ORDER BY DATE\_FORMAT(payment\_date, '%Y-%m')) AS monthly\_revenue

FROM payment

GROUP BY DATE\_FORMAT(payment\_date, '%Y-%m');

7. \*\*Identify the customers whose total spending on rentals falls within the top 20% of all customers:\*\*

SELECT customer\_id, CONCAT(first\_name, ' ', last\_name) AS customer\_name, total\_amount,

       NTILE(5) OVER (ORDER BY total\_amount DESC) AS percentile

FROM (

    SELECT c.customer\_id, c.first\_name, c.last\_name, SUM(p.amount) AS total\_amount

    FROM customer c

    JOIN payment p ON c.customer\_id = p.customer\_id

    GROUP BY c.customer\_id

) AS customer\_amounts

WHERE percentile = 1;

8. \*\*Calculate the running total of rentals per category, ordered by rental count:\*\*

\SELECT category\_id, film\_id, title, rental\_count,

       SUM(rental\_count) OVER (PARTITION BY category\_id ORDER BY film\_id) AS running\_total

FROM (

    SELECT fc.category\_id, f.film\_id, f.title, COUNT(r.rental\_id) AS rental\_count

    FROM film f

    JOIN film\_category fc ON f.film\_id = fc.film\_id

    JOIN inventory i ON f.film\_id = i.film\_id

    JOIN rental r ON i.inventory\_id = r.inventory\_id

    GROUP BY fc.category\_id, f.film\_id, f.title

) AS category\_film\_rentals;

9. \*\*Find the films that have been rented less than the average rental count for their respective categories:\*\*

SELECT category\_id, film\_id, title, rental\_count,

       AVG(rental\_count) OVER (PARTITION BY category\_id) AS avg\_rental\_count

FROM (

    SELECT fc.category\_id, f.film\_id, f.title, COUNT(r.rental\_id) AS rental\_count

    FROM film f

    JOIN film\_category fc ON f.film\_id = fc.film\_id

    JOIN inventory i ON f.film\_id = i.film\_id

    JOIN rental r ON i.inventory\_id = r.inventory\_id

    GROUP BY fc.category\_id, f.film\_id, f.title

) AS category\_film\_rentals

WHERE rental\_count < AVG(rental\_count) OVER (PARTITION BY category\_id);

10. \*\*Identify the top 5 months with the highest revenue and display the revenue generated in each month:\*\*

SELECT payment\_month, monthly\_revenue

FROM (

    SELECT DATE\_FORMAT(payment\_date, '%Y-%m') AS payment\_month,

           SUM(amount) AS monthly\_revenue,

           RANK() OVER (ORDER BY SUM(amount) DESC) AS revenue\_rank

    FROM payment

    GROUP BY DATE\_FORMAT(payment\_date, '%Y-%m')

) AS monthly\_revenues

WHERE revenue\_rank <= 5;