Basic Level

1. List all customers with their first and last names
2. Find all films with a rating of 'PG-13'
3. Show all staff members and their email addresses
4. Display all categories of films available
5. List the first 10 actors in alphabetical order by last name

Intermediate Level

1. Find all customers who live in the city 'London'
2. Show films that are longer than 120 minutes
3. List actors who have appeared in more than 30 films
4. Find the total number of films in each category
5. Display customers and the total amount they've spent on rentals
6. Show all films that have never been rented
7. Find the most popular film category by rental count

Advanced Level

1. List the top 5 customers by total rental payments
2. Find actors who have worked together in multiple films
3. Show monthly rental revenue for each store
4. Find customers who haven't rented anything in the last 30 days
5. Display the average rental duration for each film category
6. List stores with their total inventory count and revenue
7. Find films that are available in all stores
8. Show the correlation between film length and rental rate

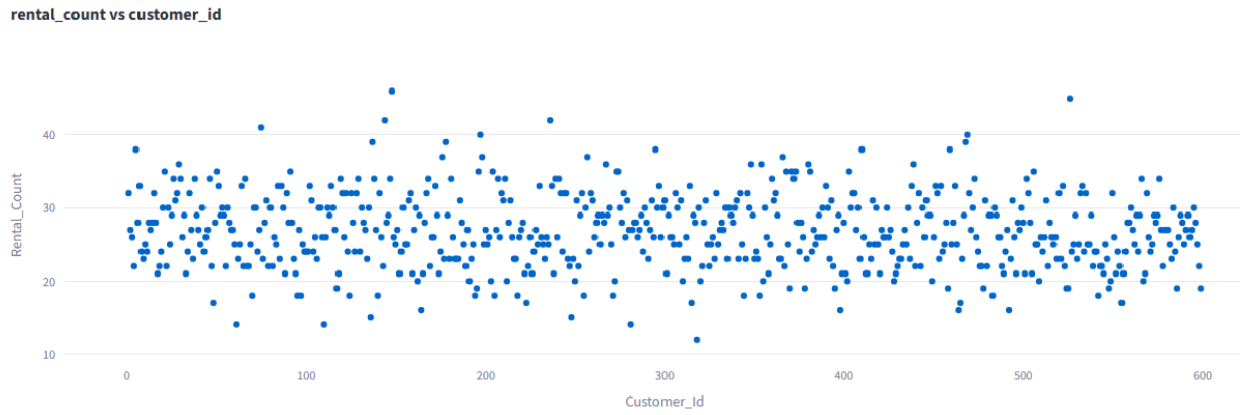
Expert Level

Question: "Analyze seasonal rental patterns by showing monthly rental trends for each film category, including the average customer age during peak months, most active customers per season, and correlation between film length and rental frequency across different time periods."

SQL:

```
SELECT YEAR(r.rental_date) AS rental_year, MONTH(r.rental_date) AS rental_month, c.name AS category_name, COUNT(r.rental_id) AS monthly_rental_count, AVG(f.length) AS average_film_length_rented 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 c ON fc.category_id = c.category_id GROUP BY rental_year, rental_month, c.name ORDER BY rental_year, rental_month, c.name;
```

Question: Create a customer segmentation based on rental frequency and spending



WITH CustomerActivity AS (SELECT c.customer_id, c.first_name, c.last_name, COUNT(r.rental_id) AS rental_count, SUM(p.amount) AS total_spending FROM customer c JOIN rental r ON c.customer_id = r.customer_id JOIN payment p ON r.rental_id = p.rental_id GROUP BY c.customer_id, c.first_name, c.last_name), CustomerSegments AS (SELECT customer_id, first_name, last_name, rental_count, total_spending, NTILE(4) OVER (ORDER BY rental_count) AS rental_segment_rank, NTILE(4) OVER (ORDER BY total_spending) AS spending_segment_rank FROM CustomerActivity) SELECT customer_id, first_name, last_name, rental_count, total_spending, rental_segment_rank, spending_segment_rank, CONCAT('R', rental_segment_rank, 'S', spending_segment_rank) AS customer_segment FROM CustomerSegments ORDER BY customer_id;

Question: Find the optimal inventory levels for each film per store

SQL:

WITH CurrentInventory AS (SELECT store_id, film_id, COUNT(inventory_id) AS current_stock FROM inventory GROUP BY store_id, film_id), RentalActivity AS (SELECT i.store_id, i.film_id, COUNT(r.rental_id) AS total_rentals FROM rental r JOIN inventory i ON r.inventory_id = i.inventory_id GROUP BY i.store_id, i.film_id) SELECT ci.store_id, f.title AS film_title, ci.current_stock, COALESCE(ra.total_rentals, 0) AS total_rentals FROM CurrentInventory ci JOIN film f ON ci.film_id = f.film_id LEFT JOIN RentalActivity ra ON ci.store_id = ra.store_id AND ci.film_id = ra.film_id ORDER BY ci.store_id, f.title;

Question: Analyze seasonal rental patterns by month and category

SQL:

SELECT MONTH(r.rental_date) AS rental_month, c.name AS category_name, COUNT(r.rental_id) AS rental_count FROM rental AS r JOIN inventory AS i ON r.inventory_id = i.inventory_id JOIN film AS f ON i.film_id = f.film_id JOIN film_category AS fc ON f.film_id = fc.film_id JOIN category AS c ON fc.category_id = c.category_id GROUP BY rental_month, category_name ORDER BY rental_month, category_name;

Question: Identify potential film recommendations for customers based on their rental history

SQL:

WITH CustomerRentedFilms AS (SELECT c.customer_id, f.film_id FROM customer c 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), CustomerRentedCategories AS (SELECT DISTINCT crf.customer_id, fc.category_id FROM CustomerRentedFilms crf JOIN film_category fc ON crf.film_id = fc.film_id), AllFilmsWithCategories AS (SELECT f.film_id, f.title AS film_title, fc.category_id FROM film f JOIN film_category fc ON f.film_id = fc.film_id) SELECT DISTINCT crc.customer_id, afc.film_id AS recommended_film_id, afc.film_title AS recommended_film_title FROM CustomerRentedCategories crc JOIN AllFilmsWithCategories afc ON crc.category_id = afc.category_id LEFT JOIN CustomerRentedFilms crf ON crc.customer_id = crf.customer_id AND afc.film_id = crf.film_id WHERE crf.film_id IS NULL -- Filter out films the customer has already rented ORDER BY crc.customer_id, afc.film_title;