

## **Questions for Sakila Database**

1. How many distinct last names of actors are there?
2. Which actors participated in the movie 'Academy Dinosaur'? Print their first and last names.
3. How many copies of the film 'Hunchback Impossible' exist in the inventory system?
4. What is the total amount paid by each customer for all their rentals? For each customer print their name and the total amount paid.
5. How many films from each category each store has? Print the store id, category name and number of films. Order the results by store id and category name.
6. Calculate the total revenue of each store.
7. Which actor participated in the most films? Print their full name and in how many movies they participated.
8. Find pairs of actors that participated together in the same movie and print their full names.
9. Each such pair should appear only once in the result. (You should have 10,385 rows in the result)
10. Display the top five most popular films, i.e., films that were rented the highest number of times.  
For each film print its title and the number of times it was rented.
11. Is the film 'Academy Dinosaur' available for rent from Store 1? You should check that the film exists as one of the items in the inventory of Store 1, and that there is no outstanding rental of that item with no return date.
12. Display the customer names and the total payments they've made, including customers with no payments, using FULL JOIN.
13. Count the number of films in each category, assigning a unique row number for each category based on the count using ROW\_NUMBER().
14. Retrieve the customer names and the total payments, ranking them based on payment amounts, and including the LEAD() value for the next customer's payment amount within the same rank.
15. Display the film titles and their replacement costs, ordering them by replacement costs in ascending order and incorporating the LAG() value for the previous film's replacement cost.
16. List the first and last names of customers, along with their rental counts, using DENSE\_RANK() to assign a dense rank based on the rental counts.
17. Retrieve the film titles and their rental rates, ordering them by rental rates in descending order and including the LEAD() value for the next rental rate.
18. Retrieve the customer names and the total payments, indicating whether the payment was made on a weekday or weekend.
19. Count the number of rentals made on each day of the week.
20. Retrieve the film titles and the average days between consecutive rentals for each film.
21. Display the customer names and the average time between their consecutive rentals.
22. List the film titles and their rental rates, showing the percentage change in rental rates compared to the average rental rate.