

Instructions

- 1a. Display the first and last names of all actors from the table actor.
- 1b. Display the first and last name of each actor in a single column in upper case letters. Name the column Actor Name.
- 2a. You need to find the ID number, first name, and last name of an actor, of whom you know only the first name, "Joe." What is one query would you use to obtain this information?
- 2b. Find all actors whose last name contain the letters GEN:
- 2c. Find all actors whose last names contain the letters LI. This time, order the rows by last name and first name, in that order:
- 2d. Using IN, display the country_id and country columns of the following countries: Afghanistan, Bangladesh, and China:
- 3a. You want to keep a description of each actor. You don't think you will be performing queries on a description, so create a column in the table actor named description and use the data type BLOB (Make sure to research the type BLOB, as the difference between it and VARCHAR are significant).
- 3b. Very quickly you realize that entering descriptions for each actor is too much effort. Delete the description column.
- 4a. List the last names of actors, as well as how many actors have that last name.
- 4b. List last names of actors and the number of actors who have that last name, but only for names that are shared by at least two actors
- 4c. The actor HARPO WILLIAMS was accidentally entered in the actor table as GROUCHO WILLIAMS. Write a query to fix the record.
- 4d. Perhaps we were too hasty in changing GROUCHO to HARPO. It turns out that GROUCHO was the correct name after all! In a single query, if the first name of the actor is currently HARPO, change it to GROUCHO.
- 5a. You cannot locate the schema of the address table. Which query would you use to re-create it?

Hint: <https://dev.mysql.com/doc/refman/5.7/en/show-create-table.html>

- 6a. Use JOIN to display the first and last names, as well as the address, of each staff member. Use the tables staff and address:
- 6b. Use JOIN to display the total amount rung up by each staff member in August of 2005. Use tables staff and payment.
- 6c. List each film and the number of actors who are listed for that film. Use tables film_actor and film. Use inner join.
- 6d. How many copies of the film Hunchback Impossible exist in the inventory system?

6e. Using the tables payment and customer and the JOIN command, list the total paid by each customer. List the customers alphabetically by last name:

![[Total amount paid]](Images/total_payment.png)

7a. The music of Queen and Kris Kristofferson have seen an unlikely resurgence. As an unintended consequence, films starting with the letters K and Q have also soared in popularity. Use subqueries to display the titles of movies starting with the letters K and Q whose language is English.

7b. Use subqueries to display all actors who appear in the film Alone Trip.

7c. You want to run an email marketing campaign in Canada, for which you will need the names and email addresses of all Canadian customers. Use joins to retrieve this information.

7d. Sales have been lagging among young families, and you wish to target all family movies for a promotion. Identify all movies categorized as family films.

7e. Display the most frequently rented movies in descending order.

7f. Write a query to display how much business, in dollars, each store brought in.

7g. Write a query to display for each store its store ID, city, and country.

7h. List the top five genres in gross revenue in descending order. (Hint: you may need to use the following tables: category, film_category, inventory, payment, and rental.)

8a. In your new role as an executive, you would like to have an easy way of viewing the Top five genres by gross revenue. Use the solution from the problem above to create a view. If you haven't solved 7h, you can substitute another query to create a view.

8b. How would you display the view that you created in 8a?

8c. You find that you no longer need the view top_five_genres. Write a query to delete it.

Appendix: List of Tables in the Sakila DB

A schema is also available as sakila_schema.svg. Open it with a browser to view.

'actor'
'actor_info'
'address'
'category'
'city'
'country'
'customer'
'customer_list'
'film'

'film_actor'
'film_category'
'film_list'
'film_text'
'inventory'
'language'
'nicer_but_slower_film_list'
'payment'
'rental'
'sales_by_film_category'
'sales_by_store'
'staff'
'staff_list'
'store'