SQL PRACTICE QUESTIONS AND ANSWERS:

* 1a. Display the first and last names of all actors from the table actor.

SELECT first\_name,last\_name FROM sakila.actor;

* 1b. Display the first and last name of each actor in a single column in upper case letters. Name the column Actor Name.

SELECT upper(CONCAT(first\_name," ",last\_name)) AS Full\_name

FROM sakila.actor;

* 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?

SELECT actor\_id,

first\_name,

last\_name

FROM sakila.actor

WHERE first\_name = 'Joe';

* 2b. Find all actors whose last name contain the letters GEN:

SELECT actor\_id,

first\_name,

last\_name

FROM sakila.actor

WHERE last\_name LIKE '%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:

SELECT actor\_id,

first\_name,

last\_name

FROM sakila.actor

WHERE last\_name LIKE '%LI%'

ORDER BY 3,2;

* 2d. Using IN, display the country\_id and country columns of the following countries: Afghanistan, Bangladesh, and China:

SELECT country\_id,

country

FROM sakila.country

WHERE country IN ('Afghanistan', 'Bangladesh', 'China')

* 3a. Add a middle\_name column to the table actor. Position it between first\_name and last\_name. Hint: you will need to specify the data type.

|  |
| --- |
|  |
|  | ALTER TABLE actor  add column middle\_name varchar(20) |
|  | after first\_name; |

* 3b. You realize that some of these actors have tremendously long last names. Change the data type of the middle\_name column to blobs.

alter table actor

modify column middle\_name blob;;

* 3c. Now delete the middle\_name column.

alter table actor

drop column middle\_name;

* 4a. List the last names of actors, as well as how many actors have that last name.

SELECT last\_name, COUNT(\*)

FROM actor

GROUP BY 1;

* 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

SELECT last\_name, COUNT(\*)

FROM actor

GROUP BY 1

HAVING COUNT(\*) > 1;

* 4c. Oh, no! The actor HARPO WILLIAMS was accidentally entered in the actor table as GROUCHO WILLIAMS, the name of Harpo's second cousin's husband's yoga teacher. Write a query to fix the record.

UPDATE actor

SET first\_name = 'HARPO'

WHERE first\_name = 'GROUCHO' AND last\_name = 'WILLIAMS' AND actor\_id = 172;

* 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. Otherwise, change the first name to MUCHO GROUCHO, as that is exactly what the actor will be with the grievous error. BE CAREFUL NOT TO CHANGE THE FIRST NAME OF EVERY ACTOR TO MUCHO GROUCHO, HOWEVER! (Hint: update the record using a unique identifier.)

UPDATE actor

SET first\_name =

CASE

WHEN (actor\_id = 172 AND first\_name = 'HARPO') THEN 'GROUCHO'

WHEN (actor\_id = 172 AND first\_name <> 'HARPO') THEN 'MUCHO GROUCHO'

ELSE

first\_name

END

WHERE actor\_id = 172;

* 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>

show create table address;

* 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:

SELECT S.first\_name,

S.last\_name,

A.address

FROM staff AS S

JOIN address AS A

ON S.address\_id = A.address\_id;

* 6b. Use JOIN to display the total amount rung up by each staff member in August of 2005. Use tables staff and payment.

SELECT S.staff\_id,

S.first\_name,

S.last\_name,

SUM(P.amount) AS total\_amount

FROM staff AS S

JOIN payment AS P

ON S.staff\_id = P.staff\_id

WHERE DATE(P.payment\_date) BETWEEN '2005-08-01' AND '2005-08-31'

GROUP BY 1,2,3

* 6c. List each film and the number of actors who are listed for that film. Use tables film\_actor and film. Use inner join.

SELECT

F.title,

COUNT(\*) AS No\_of\_Actors

FROM film AS F

JOIN film\_actor AS A

ON F.film\_id = A.film\_id

GROUP BY 1;

* 6d. How many copies of the film Hunchback Impossible exist in the inventory system?

SELECT F.title,

COUNT(\*) AS copies

FROM film AS F

JOIN inventory AS I

ON F.film\_id = I.film\_id

WHERE F.title = 'Hunchback Impossible'

GROUP BY 1;

* 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:

SELECT

C.customer\_id,

C.first\_name,

C.last\_name,

SUM(P.amount) AS total\_amt

FROM customer AS C

JOIN payment AS P

ON C.customer\_id = P.customer\_id

GROUP BY 1,2,3

ORDER BY 3;

* 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.

SELECT title

FROM film

WHERE UPPER(title) LIKE 'K%' OR UPPER(title) LIKE 'Q%' AND language\_id = (

SELECT language\_id FROM language

WHERE name = 'English' )

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

SELECT F.film\_id,

A.actor\_id,

A.first\_name,

A.last\_name

FROM film\_actor AS F

JOIN actor AS A

ON F.actor\_id = A.actor\_id

WHERE F.film\_id =

(SELECT film\_id FROM film WHERE UPPER(title) = '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.

SELECT C.first\_name,

C.last\_name,

C.email

FROM CUSTOMER AS C

JOIN address AS A

ON C.address\_id = A.address\_id

JOIN city AS CI

ON A.city\_id = CI.city\_id

JOIN country AS CO

ON CI.country\_id = CO.country\_id

WHERE CO.country = 'Canada';

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

SELECT f.title

FROM film AS F

JOIN film\_category AS FC

ON F.film\_id = FC.film\_id

JOIN category AS C

ON FC.category\_id = C.category\_id

WHERE lower(C.name) = 'family'

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

SELECT F.title,

COUNT(\*)

FROM film AS F

JOIN inventory AS I

ON F.film\_id = I.film\_id

JOIN rental AS R

ON I.inventory\_id = R.inventory\_id

GROUP BY 1

ORDER BY 2 desc;

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

SELECT S.store\_id,

SUM(P.amount) As total\_amt

FROM store AS S

JOIN staff AS ST

ON S.store\_id = ST.store\_id

JOIN payment AS P

ON ST.staff\_id = P.staff\_id

GROUP BY 1;

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

SELECT S.store\_id,

CI.city AS city,

Co.country AS country

FROM store AS S

JOIN address AS A

ON S.address\_id = A.address\_id

JOIN city AS CI

ON A.city\_id = CI.city\_id

JOIN country AS CO

ON CI.country\_id = Co.country\_id;

* 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.)

SELECT C.name,

SUM(P.amount) AS gross\_Amount

FROM category AS C

JOIN film\_category AS FC

ON C.category\_id = FC.category\_id

JOIN inventory AS I

ON FC.film\_id = I.film\_id

JOIN rental AS R

ON I.inventory\_id = R.inventory\_id

JOIN payment AS P

ON R.rental\_id = P.rental\_id

GROUP BY 1

ORDER BY 2 DESC

LIMIT 5;

* 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.

CREATE VIEW Top\_5\_genres AS

SELECT C.name,

SUM(P.amount) AS gross\_Amount

FROM category AS C

JOIN film\_category AS FC

ON C.category\_id = FC.category\_id

JOIN inventory AS I

ON FC.film\_id = I.film\_id

JOIN rental AS R

ON I.inventory\_id = R.inventory\_id

JOIN payment AS P

ON R.rental\_id = P.rental\_id

GROUP BY 1

ORDER BY 2 DESC

LIMIT 5;

* 8b. How would you display the view that you created in 8a?
* SELECT \* FROM top\_5\_genres;
* 8c. You find that you no longer need the view top\_five\_genres. Write a query to delete it.

DROP VIEW top\_5\_genres;