

Actividad 09 – Consultas

1. Listar todos los clientes y su información de contacto.

```
mysql> SELECT customer_id, first_name, email FROM customer;
```

customer_id	first_name	email
1	MARY	MARY.SMITH@sakilacustomer.org
2	PATRICIA	PATRICIA.JOHNSON@sakilacustomer.org
3	LINDA	LINDA.WILLIAMS@sakilacustomer.org
4	BARBARA	BARBARA.JONES@sakilacustomer.org
5	ELIZABETH	ELIZABETH.BROWN@sakilacustomer.org
6	JENNIFER	JENNIFER.DAVIS@sakilacustomer.org
7	MARIA	MARIA.MILLER@sakilacustomer.org
8	SUSAN	SUSAN.WILSON@sakilacustomer.org
9	MARGARET	MARGARET.MOORE@sakilacustomer.org
10	DOROTHY	DOROTHY.TAYLOR@sakilacustomer.org
11	LISA	LISA.ANDERSON@sakilacustomer.org
12	NANCY	NANCY.THOMAS@sakilacustomer.org
13	KAREN	KAREN.JACKSON@sakilacustomer.org
14	BETTY	BETTY.WHITE@sakilacustomer.org
15	HELEN	HELEN.HARRIS@sakilacustomer.org
16	SANDRA	SANDRA.MARTIN@sakilacustomer.org
17	DONNA	DONNA.THOMPSON@sakilacustomer.org
18	CAROL	CAROL.GARCIA@sakilacustomer.org
19	RUTH	RUTH.MARTINEZ@sakilacustomer.org

2. Mostrar todas las películas con una duración mayor a 120 minutos.

```
mysql> SELECT title, length FROM film WHERE length > 120;
```

title	length
AFRICAN EGG	130
AGENT TRUMAN	169
ALAMO VIDEOTAPE	126
ALASKA PHANTOM	136
ALI FOREVER	150
ALLEY EVOLUTION	180
AMERICAN CIRCUS	129
ANALYZE HOOSIERS	181
ANONYMOUS HUMAN	179
ANTITRUST TOMATOES	168
APOLLO TEEN	153
ARACHNOPHOBIA ROLLERCOASTER	147
ARGONAUTS TOWN	127
ARIZONA BANG	121
ARMY FLINTSTONES	148
ARSENIC INDEPENDENCE	137
ARTIST COLDBLOODED	170
ATLANTIS CAUSE	170
BABY HALL	153
BADMAN DAWN	162

3. Encontrar todos los empleados que son gerentes de tienda.

```
mysql> SELECT first_name FROM staff INNER JOIN store ON staff.store_id = store.manager_staff_id;
```

first_name
Mike
Jon

2 rows in set (0.00 sec)

4. Listar los clientes que han alquilado más de 10 películas.

```
mysql> SELECT rental.customer_id, customer.first_name, customer.last_name, COUNT(rental_id) AS rental_film FROM rental JOIN customer
ON rental.customer_id = customer.customer_id GROUP BY rental.customer_id HAVING COUNT(rental_id) >10;
```

customer_id	first_name	last_name	rental_film
1	MARY	SMITH	32
2	PATRICIA	JOHNSON	27
3	LINDA	WILLIAMS	26
4	BARBARA	JONES	22
5	ELIZABETH	BROWN	38
6	JENNIFER	DAVIS	28
7	MARIA	MILLER	33
8	SUSAN	WILSON	24

5. Mostrar las 5 películas más populares basadas en el número de alquileres.

```
mysql> SELECT film.film_id, film.title, COUNT(rental.rental_id) as num_film FROM film JOIN inventory ON film.film_id = inventory.film_id JOIN rental ON inventory.in
ventory_id = rental.inventory_id GROUP BY film.film_id, film.title ORDER BY num_film LIMIT 5;
```

film_id	title	num_film
904	TRAIN BUNCH	4
400	HARDLY ROBBERS	4
584	MIXED DOORS	4
107	BUNCH MINDS	5
94	BRAVEHEART HUMAN	5

5 rows in set (0.02 sec)

6. Encontrar todas las tiendas y el número de empleados en cada una.

```
mysql> SELECT store.store_id, COUNT(staff.staff_id) as num_staff FROM store left JOIN staff ON store.store_id = staff.staff_id GROUP
BY store.store_id;
```

store_id	num_staff
1	1
2	1

2 rows in set (0.00 sec)

7. Listar todas las películas que no han sido alquiladas nunca.

```
mysql> SELECT film.film_id, film.title FROM film JOIN inventory ON inventory.film_id = film.film_id JOIN rental ON rental.inventory_
id = inventory.inventory_id WHERE rental.rental_id = 0;
```

Empty set (0.00 sec)

8. Mostrar los 10 clientes que más gastaron en alquileres.

```
mysql> SELECT customer.customer_id, customer.first_name, SUM(payment.amount) as gasto FROM customer JOIN payment ON payment.customer_
id = customer.customer_id GROUP BY customer.customer_id, customer.first_name LIMIT 10;
```

customer_id	first_name	gasto
1	MARY	118.68
2	PATRICIA	128.73
3	LINDA	135.74
4	BARBARA	81.78
5	ELIZABETH	144.62
6	JENNIFER	93.72
7	MARIA	151.67
8	SUSAN	92.76
9	MARGARET	89.77
10	DOROTHY	99.75

10 rows in set (0.02 sec)

9. Encontrar las películas alquiladas en un rango de fechas específico.

```
mysql> SELECT film.film_id, film.title, COUNT(rental.rental_id) as cant_renta FROM film JOIN inventory ON inventory.inventory_id = film.film
_id JOIN rental ON rental.inventory_id = inventory.inventory_id GROUP BY film.film_id, film.title HAVING COUNT(rental.rental_id) = 0;
```

Empty set (0.01 sec)

10. Listar los actores que no han participado en ninguna película.

```
mysql> SELECT actor.actor_id, actor.first_name, COUNT(film.film_id) as cant_peli
FROM actor JOIN film_actor ON film_actor.actor_id = actor.actor_id JOIN film ON f
ilm.film_id = film_actor.film_id GROUP BY actor.actor_id, actor.first_name HAVING
COUNT(film.film_id) = 0;
```

Empty set (0.01 sec)

11. Mostrar los géneros de películas con el menor número de títulos en el inventario.

```
mysql> SELECT category.category_id, category.name, COUNT(film.film_id) as cant_peliculas FROM category JOIN film_category ON film_category.category_id = category.category_id JOIN film ON film.film_id = film_category.film_id GROUP BY category.category_id, category.name ORDER BY COUNT(film.film_id) ASC LIMIT 6;
```

category_id	name	cant_peliculas
12	Music	51
11	Horror	56
4	Classics	57
16	Travel	57
5	Comedy	58
3	Children	60

6 rows in set (0.00 sec)

12. Encontrar los clientes que tienen alquileres atrasados actualmente.

```
mysql> SELECT customer.customer_id, customer.first_name, rental.return_date FROM customer JOIN rental ON rental.customer_id = customer.customer_id WHERE rental.return_date IS NULL;
```

customer_id	first_name	return_date
155	GAIL	NULL
335	GREGORY	NULL
83	LOUISE	NULL
219	WILLIE	NULL
99	EMILY	NULL
192	LAURIE	NULL
11	LISA	NULL
597	FREDDIE	NULL
53	HEATHER	NULL
521	ROLAND	NULL
216	NATALIE	NULL