

1. Muestra el nombre y apellido de todos los actores.

```
SELECT first_name, last_name  
FROM actor;
```

2. Muestra el nombre y apellido de cada actor en una sola columna, en mayúscula. Nombra la columna "Nombre del actor".

```
SELECT UPPER(CONCAT(first_name, ' ', last_name)) AS "Nombre del actor"  
  
FROM actor;
```

3. Muestra el ID, nombre y apellido de un actor, de quien solo tienes el nombre "Joe".

```
SELECT actor_id, first_name, last_name  
  
FROM actor  
  
WHERE first_name = 'Joe';
```

4. Encuentra los actores cuyo apellido contenga "GEN".

```
SELECT actor_id, first_name, last_name  
  
FROM actor  
  
WHERE last_name LIKE '%GEN%'
```

5. Encuentra los actores cuyo apellido contenga "LI". Ordena las filas por apellido y nombre (en ese orden).

```
SELECT actor_id, first_name, last_name
FROM actor
WHERE last_name LIKE '%LI%'
ORDER BY last_name, first_name;
```

6. Usando la función IN, muestra el nombre y apellido de todos los clientes llamados "Terry", "Jessie" o "Alice".

```
SELECT first_name, last_name
FROM customer
WHERE first_name IN ('Terry', 'Jessie', 'Alice');
```

7. Muestra el apellido de cada actor y la cantidad de actores que tienen ese apellido.

```
SELECT last_name, COUNT(*) AS cantidad
FROM actor
```

GROUP BY last_name;8. Muestra el apellido y la cantidad de actores que tienen ese apellido, pero solo los apellidos compartidos por dos o más actores.

```
SELECT last_name, COUNT(*) AS cantidad
FROM actor
GROUP BY last_name
HAVING COUNT(*) >= 2;
```

9. Usando join, muestra el nombre, apellido y dirección de cada miembro del staff.

```
SELECT S.first_name, S.last_name, A.address  
FROM staff S  
JOIN address A  
ON S.address_id = A.address_id;
```

10. Muestra el total de dinero recaudado por cada empleado durante agosto del 2005.

```
SELECT s.staff_id,  
       s.first_name,  
       s.last_name,  
       SUM(p.amount) AS total_recaudado  
FROM staff s  
JOIN payment p  
ON s.staff_id = p.staff_id  
WHERE p.payment_date BETWEEN '2005-08-01' AND '2005-08-31'  
GROUP BY s.staff_id, s.first_name, s.last_name;
```

11. Despliega la cantidad de actores por cada película.

```
SELECT f.film_id,  
       f.title,  
       COUNT(fa.actor_id) AS cantidad_actores  
FROM film f  
JOIN film_actor fa  
  ON f.film_id = fa.film_id  
GROUP BY f.film_id, f.title  
ORDER BY cantidad_actores DESC;
```

12. ¿Cuántas copias hay inventariadas en el sistema de la película "Hunchback Impossible"?

```
SELECT f.title, COUNT(i.inventory_id) AS cantidad_copias  
FROM film f  
JOIN inventory i  
  ON f.film_id = i.film_id  
WHERE f.title = 'Hunchback Impossible'  
GROUP BY f.title;
```

13. Muestra el total de dinero pagado por cada cliente, solo si ha realizado compras. Ordena los clientes por apellido de forma ascendente.

```
SELECT c.customer_id,  
       c.first_name,  
       c.last_name,  
       SUM(p.amount) AS total_pagado  
FROM customer c  
JOIN payment p  
  ON c.customer_id = p.customer_id  
GROUP BY c.customer_id, c.first_name, c.last_name  
ORDER BY c.last_name ASC;
```

14. Se debe realizar una campaña de marketing en Canada. Para esto necesitas el nombre y correo electrónico de todos los clientes canadienses. Despliega esta información.

```
SELECT c.first_name,  
       c.last_name,  
       c.email  
FROM customer c  
JOIN address a  
  ON c.address_id = a.address_id  
JOIN city ci  
  ON a.city_id = ci.city_id  
JOIN country co  
  ON ci.country_id = co.country_id  
WHERE co.country = 'Canada';
```

15. Identifica todas las películas categorizadas como familiares (categoría "family").

```
SELECT f.film_id,  
       f.title  
FROM film f  
JOIN film_category fc  
  ON f.film_id = fc.film_id  
JOIN category c  
  ON fc.category_id = c.category_id  
WHERE c.name = 'Family';
```

16. Muestra las películas más arrendadas en orden descendente.

```
SELECT f.film_id,  
       f.title,  
       COUNT(r.rental_id) AS total_rentas  
FROM film f  
JOIN inventory i  
  ON f.film_id = i.film_id  
JOIN rental r  
  ON i.inventory_id = r.inventory_id  
GROUP BY f.film_id, f.title  
ORDER BY total_rentas DESC;
```

17. Despliega el dinero recaudado por cada tienda.

```
SELECT s.store_id,  
       SUM(p.amount) AS total_recaudado  
FROM store s  
JOIN staff st  
  ON s.store_id = st.store_id  
JOIN payment p  
  ON st.staff_id = p.staff_id  
GROUP BY s.store_id;
```