

Ejercicio 1: Mostrar los títulos de las películas disponibles

Objetivo: Usar SELECT y DISTINCT.

```
SELECT DISTINCT title FROM film;
```

Muestra una lista única de todos los títulos de películas en la tabla film.

Ejercicio 2: Listar las películas que duran más de 120 minutos

Objetivo: Usar WHERE.

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

Devuelve los títulos de las películas junto con su duración si exceden los 120 minutos.

Ejercicio 3: Ordenar las películas por duración en orden descendente

Objetivo: Usar ORDER BY.

```
SELECT title FROM film ORDER BY length DESC;
```

Solución: Lista las películas ordenadas de mayor a menor duración.

Ejercicio 4: Contar las películas por clasificación

Objetivo: Usar GROUP BY y una función agregada.

```
SELECT rating COUNT (*) AS total_films FROM film GROUP BY rating;
```

Muestra el número de películas agrupadas por clasificación.

Ejercicio 5: Buscar películas cuyo título contiene la palabra "Love"

Objetivo: Usar funciones de caracteres.

```
SELECT title FROM film WHERE title LIKE '%Love%';
```

Devuelve los títulos de películas que contienen la palabra "Love".

Ejercicio 6: Calcular la duración promedio de las películas por clasificación

Objetivo: Usar funciones agregadas y GROUP BY.

```
SELECT rating AVG(length) AS duración_promedio FROM film GROUP BY rating;
```

Muestra la duración promedio de las películas agrupadas por clasificación.

Ejercicio 7: Obtener el cliente más reciente registrado

Objetivo: Usar ORDER BY y LIMIT.

```
SELECT first_name, last_name, create_date FROM customer ORDER BY create_date  
DESC LIMIT 1;
```

Devuelve el nombre, apellido y fecha de creación del cliente más reciente.

Ejercicio 8: Contar cuántos actores tienen un apellido que empieza con 'A'

Objetivo: Usar funciones de caracteres y COUNT.

```
SELECT COUNT(*) AS actores_con_a FROM actor WHERE last_name LIKE 'A%';
```

Cuenta cuántos actores tienen un apellido que comienza con la letra "A".

Ejercicio 9: Mostrar el ingreso total generado por cada tienda

Objetivo: Usar funciones agregadas y GROUP BY.

```
SELECT stored_id SUM(amount) AS total_income FROM payment GROUP BY stored_id;
```

Muestra el ingreso total generado por cada tienda.

Ejercicio 10: Obtener los 5 actores más frecuentes en películas

Objetivo: Usar funciones agregadas, GROUP BY y ORDER BY.

```
SELECT actor_id COUNT (*) AS film_cont FROM film_actor GROUP BY actor_id ORDER BY film_count DESC LIMIT 5;
```

Devuelve los IDs de los 5 actores que aparecen en más películas, junto con la cantidad de películas en las que han actuado.

Ejercicio 11: Contar el número total de clientes

Objetivo: Usar COUNT.

```
SELECT COUNT(*) AS total_customers FROM customer;
```

Devuelve el total de registros en la tabla customer.

Ejercicio 12: Encontrar la duración máxima de una película

Objetivo: Usar MAX.

```
SELECT MAX (length) AS max_length FROM film;
```

Muestra la duración máxima de las películas en la tabla film.

Ejercicio 13: Encontrar la duración mínima de una película

Objetivo: Usar MIN.

```
SELECT MIN(length) AS min_length FROM film;
```

Muestra la duración mínima de las películas en la tabla film.

Ejercicio 14: Calcular el ingreso promedio generado por cada pago

Objetivo: Usar AVG.

```
SELECT AVG(amount) AS avg_payment FROM payment;
```

Devuelve el monto promedio de los pagos realizados en la tabla payment.

Ejercicio 15: Calcular el ingreso total generado por todas las transacciones

Objetivo: Usar SUM.

```
SELECT SUM(amount) AS total_incore FROM payment;
```

