

## Actividad 11 - Procedimientos almacenados

### Calcular el total de alquileres por cliente:

Crea un procedimiento almacenado que devuelva el número total de alquileres realizados por cada cliente en la base de datos.

DELIMITER //

CREATE PROCEDURE alquilerClient()

-> BEGIN

-> SELECT customer.first\_name, COUNT(rental.rental\_id) AS cantAlquiler FROM customer JOIN rental ON rental.customer\_id = customer.customer\_id

-> GROUP BY customer.first\_name;

-> END //

DELIMITER ;

```
mysql> CREATE PROCEDURE alquilerClient()
-> BEGIN
-> SELECT customer.first_name, COUNT(rental.rental_id) AS cantAlquiler FROM customer JOIN rental ON rental.customer_id = customer.customer_id
-> GROUP BY customer.first_name;
-> END //
Query OK, 0 rows affected (0.01 sec)
```

### Obtener la lista de películas en stock:

Desarrolla un procedimiento almacenado que devuelva la lista de películas que actualmente están en stock (es decir, tienen al menos una copia disponible en el inventario).

DELIMITER //

CREATE PROCEDURE stockPeli()

-> BEGIN

-> SELECT film.title, COUNT(inventory.inventory\_id) as inventario FROM film JOIN inventory ON inventory.film\_id = film.film\_id

-> GROUP BY film.title

-> HAVING COUNT(inventory.inventory\_id) > 0;

-> END //

DELIMITER ;

```
mysql> CREATE PROCEDURE stockPeli()
-> BEGIN
-> SELECT film.title, COUNT(inventory.inventory_id) as inventario FROM film JOIN inventory ON i
nventory.film_id = film.film_id
-> GROUP BY film.title
-> HAVING COUNT(inventory.inventory_id) > 0;
-> END //
Query OK, 0 rows affected (0.01 sec)
```

### Calcular el total de ingresos de un cliente específico:

Crea un procedimiento almacenado que tome como parámetro el ID del cliente y devuelva el total de ingresos generados por ese cliente a través de sus alquileres de películas.

DELIMITER //

CREATE PROCEDURE ingresoClientes(customer\_ingreso smallint)

-> BEGIN

-> SELECT customer.customer\_id, customer.first\_name, SUM(payment.amount) as ingresos  
FROM customer JOIN rental ON rental.customer\_id = customer.customer\_id JOIN payment ON  
payment.rental\_id = rental.rental\_id

-> WHERE customer.customer\_id LIKE CONCAT(customer\_ingreso)

-> GROUP BY customer.customer\_id;

-> END //

DELIMITER ;

```
mysql> CREATE PROCEDURE ingresoClientes(customer_ingreso smallint)
-> BEGIN
-> SELECT customer.customer_id, customer.first_name, SUM(payment.amount) as ingresos FROM customer JOIN rental ON renta
l.customer_id = customer.customer_id JOIN payment ON payment.rental_id = rental.rental_id
-> WHERE customer.customer_id LIKE CONCAT(customer_ingreso)
-> GROUP BY customer.customer_id;
-> END //
Query OK, 0 rows affected (0.01 sec)
```

### Encontrar los actores principales de una película:

Crea un procedimiento almacenado que tome como parámetro el título de una película y devuelva los nombres de los actores principales que participaron en esa película.

DELIMITER //

CREATE PROCEDURE actoresPelículas(actor\_pelicula VARCHAR(255))

-> BEGIN

-> SELECT film.title, actor.first\_name, actor.last\_name FROM film JOIN film\_actor ON film.film\_id  
= film\_actor.film\_id JOIN actor ON actor.actor\_id = film\_actor.actor\_id

-> WHERE film.title LIKE CONCAT ("%",actor\_pelicula,"%");

-> END //

DELIMITER ;

```
mysql> CREATE PROCEDURE actoresPelículas(actor_pelicula VARCHAR(255))
-> BEGIN
-> SELECT film.title, actor.first_name, actor.last_name FROM film JOIN film_actor ON film.film_id =
film_actor.film_id JOIN actor ON actor.actor_id = film_actor.actor_id
-> WHERE film.title LIKE CONCAT ("%",actor_pelicula,"%");
-> END //
Query OK, 0 rows affected (0.05 sec)
```

### Calcular el total de películas disponibles de una categoría específica:

Crea un procedimiento almacenado que tome como parámetro el nombre de una categoría y devuelva el total de películas disponibles en esa categoría.

DELIMITER //

CREATE PROCEDURE peliculaCategoria(categoria\_peli VARCHAR(255))

-> BEGIN

-> SELECT category.name, COUNT(film.film\_id) as cantCategoria FROM category JOIN  
film\_category ON category.category\_id = film\_category.category\_id JOIN film ON film.film\_id =  
film\_category.film\_id

-> WHERE category.name LIKE CONCAT ("%",categoria\_peli,"%")

-> GROUP BY category.name;

-> END //

DELIMITER ;

```
mysql> CREATE PROCEDURE peliculaCategoria(categoria_peli VARCHAR(255))
-> BEGIN
-> SELECT category.name, COUNT(film.film_id) as cantCategoria FROM category JOIN film_c
ategory ON category.category_id = film_category.category_id JOIN film ON film.film_id = fil
m_category.film_id
-> WHERE category.name LIKE CONCAT ("%",categoria_peli,"%")
-> GROUP BY category.name;
-> END //
Query OK, 0 rows affected (0.01 sec)
```