Documentación de Consultas SQL - Sistema Gestor de Cinema

Consultas de Inserción

1. Tabla Movies (Películas)

INSERT INTO Movies (

distribution\_title,

original\_title,

genre,

original\_language,

spanish\_subtitles,

origin\_countries,

production\_year,

website\_url,

duration,

age\_rating,

release\_date\_santiago,

synopsis

) VALUES (

'Título de Distribución',

'Título Original',

'Género',

'Idioma Original',

true,

'{"País1","País2"}',

2024,

'http://ejemplo.com',

'2 hours 30 minutes',

'Apta todo público',

CURRENT\_DATE,

'Sinopsis de la película'

);

Esta consulta representa la inserción de una nueva película en el sistema. Observemos varios aspectos importantes:

- El campo `origin\_countries` utiliza el formato de array de PostgreSQL

- La duración se almacena como un intervalo de tiempo

- La fecha de estreno utiliza la función CURRENT\_DATE para la fecha actual

- Los booleanos se representan como true/false

2. Tabla Directors (Directores)

INSERT INTO Directors (name, nationality)

VALUES ('Nombre del Director', 'Nacionalidad');

Una consulta simple pero fundamental que registra los directores en el sistema. La unicidad del nombre está garantizada por la restricción UNIQUE en la columna name.

3. Tabla Movies\_Directors (Relación Películas-Directores)

INSERT INTO Movies\_Directors (movie\_id, director\_id)

VALUES (1, 1);

Esta consulta establece la relación entre películas y directores. Es importante notar que:

- Ambas columnas son claves foráneas

- La combinación de ambas forma una clave primaria compuesta

- Se deben proporcionar IDs válidos existentes en las tablas respectivas

4. Tabla Actors (Actores)

INSERT INTO Actors (name, nationality)

VALUES ('Nombre del Actor', 'Nacionalidad');

Similar a la tabla Directors, mantiene la información básica de los actores con la restricción de nombre único.

5. Tabla Characters (Personajes)

INSERT INTO Characters (character\_name, actor\_id)

VALUES ('Nombre del Personaje', 1);

Esta tabla almacena los personajes y su relación con los actores que los interpretan. La consulta vincula automáticamente el personaje con un actor existente mediante actor\_id.

6. Tabla Movies\_Actors (Relación Películas-Actores-Personajes)

INSERT INTO Movies\_Actors (movie\_id, actor\_id, character\_id)

VALUES (1, 1, 1);

Esta consulta es particularmente interesante porque establece una relación triple entre:

- La película (movie\_id)

- El actor (actor\_id)

- El personaje específico (character\_id)

7. Tabla Theaters (Complejos de Cine)

INSERT INTO Theaters (name, address, phone)

VALUES ('Nombre del Cine', 'Dirección', '+56912345678');

Almacena la información básica de los complejos de cine. El formato telefónico sigue el estándar chileno.

8. Tabla Rooms (Salas)

INSERT INTO Rooms (name, number, seat\_capacity, theater\_id)

VALUES ('Nombre de la Sala', 1, 150, 1);

La consulta incluye:

- Nombre descriptivo de la sala

- Número de identificación dentro del complejo

- Capacidad de asientos

- Referencia al complejo (theater\_id)

9. Tabla Shows (Funciones)

INSERT INTO Shows (weekday, start\_time, room\_id, movie\_id)

VALUES ('Lunes', '14:30', 1, 1);

Esta consulta programa una función estableciendo:

- Día de la semana

- Hora de inicio en formato 24 horas

- Sala específica

- Película a proyectar

10. Tabla Promotions (Promociones)

INSERT INTO Promotions (description, discount)

VALUES ('Descripción de la Promoción', 25.00);

Registra promociones donde el descuento se almacena como un valor numérico que representa el porcentaje.

11. Tabla Shows\_Promotions (Relación Funciones-Promociones)

INSERT INTO Shows\_Promotions (show\_id, promo\_id)

VALUES (1, 1);

Establece qué promociones aplican a qué funciones, permitiendo múltiples promociones por función.

12. Tabla Reviews (Reseñas)

INSERT INTO Reviews (

movie\_id,

reviewer\_name,

reviewer\_age,

review\_date,

rating,

comment

) VALUES (

1,

'Nombre del Revisor',

25,

CURRENT\_TIMESTAMP,

'5 estrellas',

'Comentario sobre la película'

);

Esta consulta registra una reseña completa, incluyendo:

- Referencia a la película

- Información del revisor

- Fecha y hora exacta del registro

- Calificación y comentario detallado

Consultas de Eliminación

Eliminación de Película con Dependencias

-- Eliminar promociones de funciones relacionadas

DELETE FROM Shows\_Promotions

WHERE show\_id IN (

SELECT show\_id

FROM Shows

WHERE movie\_id = 1

);

-- Eliminar funciones

DELETE FROM Shows

WHERE movie\_id = 1;

-- Eliminar reseñas

DELETE FROM Reviews

WHERE movie\_id = 1;

-- Eliminar relaciones con actores

DELETE FROM Movies\_Actors

WHERE movie\_id = 1;

-- Eliminar relaciones con directores

DELETE FROM Movies\_Directors

WHERE movie\_id = 1;

-- Finalmente, eliminar la película

DELETE FROM Movies

WHERE movie\_id = 1;

Esta secuencia de eliminación es crucial porque:

1. Respeta las restricciones de clave foránea

2. Elimina registros relacionados en orden específico

3. Mantiene la integridad referencial de la base de datos

Consultas de Modificación

Actualizar Información de Película

UPDATE Movies

SET

distribution\_title = 'Nuevo Título',

synopsis = 'Nueva Sinopsis',

age\_rating = '+15 años'

WHERE movie\_id = 1;

Actualizar Capacidad de Sala

UPDATE Rooms

SET seat\_capacity = 200

WHERE room\_id = 1;

Actualizar Descuento de Promoción

UPDATE Promotions

SET discount = 30.00

WHERE promo\_id = 1;

Consideraciones Importantes

1. Integridad Referencial

- Todas las operaciones respetan las relaciones entre tablas

- Las eliminaciones siguen un orden específico para evitar violaciones de clave foránea

- Las actualizaciones mantienen la consistencia de los datos

2. Validación de Datos

- Las cadenas de texto respetan las longitudes máximas definidas

- Los valores numéricos están dentro de rangos aceptables

- Las fechas y horas siguen formatos específicos

3. Transacciones

- Las operaciones complejas se realizan dentro de transacciones

- Se garantiza la atomicidad de las operaciones

- Se incluye manejo de errores y rollback cuando es necesario