Guía Completa de Consultas SQL para Sistema de Video Club

Introducción

En este documento, exploraremos las consultas SQL fundamentales para gestionar un sistema de Video Club utilizando PostgreSQL. Aprenderás cómo insertar, consultar, actualizar y eliminar datos de manera efectiva y segura.

1. Consultas de Inserción

1.1 Inserción de Películas

INSERT INTO Movies (title, nationality, production\_company, release\_year)

VALUES ('Pulp Fiction', 'Estados Unidos', 'Miramax', 1994);

Consideraciones

- Usar valores precisos

- Respetar tipos de datos

- Validar información antes de insertar

1.2 Inserción de Directores

INSERT INTO Directors (name, nationality)

VALUES ('Quentin Tarantino', 'Estados Unidos');

Características

- Nombre único

- Nacionalidad descriptiva

1.3 Vinculación Película-Director

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

VALUES (

(SELECT movie\_id FROM Movies WHERE title = 'Pulp Fiction'),

(SELECT director\_id FROM Directors WHERE name = 'Quentin Tarantino')

);

Técnica Avanzada

- Uso de subconsultas

- Vinculación dinámica de registros

1.4 Inserción de Actores

INSERT INTO Actors (name, nationality, gender)

VALUES ('John Travolta', 'Estados Unidos', 'Masculino');

1.5 Inserción de Copias de Película

INSERT INTO MovieCopies (movie\_id, conservation\_state)

VALUES (

(SELECT movie\_id FROM Movies WHERE title = 'Pulp Fiction'),

'Excelente'

);

1.6 Inserción de Socios

INSERT INTO Members (dni, name, address, phone, sponsor\_id)

VALUES ('12345678901', 'Juan Pérez', 'Calle 23 456', '52345678', NULL);

2. Consultas de Selección (Consultar Datos)

2.1 Consulta Básica de Películas

SELECT title, nationality, release\_year

FROM Movies;

2.2 Películas con Directores

SELECT m.title, d.name AS director

FROM Movies m

JOIN Movies\_Directors md ON m.movie\_id = md.movie\_id

JOIN Directors d ON md.director\_id = d.director\_id;

2.3 Películas de un Actor

SELECT m.title

FROM Movies m

JOIN Movies\_Actors ma ON m.movie\_id = ma.movie\_id

JOIN Actors a ON ma.actor\_id = a.actor\_id

WHERE a.name = 'John Travolta';

2.4 Socios con sus Avaladores

SELECT m.name AS socio, s.name AS avalador

FROM Members m

LEFT JOIN Members s ON m.sponsor\_id = s.member\_id;

3. Consultas de Actualización

3.1 Actualizar Información de Película

UPDATE Movies

SET production\_company = 'Universal Pictures'

WHERE title = 'Pulp Fiction';

3.2 Actualizar Estado de Copia

UPDATE MovieCopies

SET conservation\_state = 'Regular'

WHERE movie\_id = (SELECT movie\_id FROM Movies WHERE title = 'Pulp Fiction');

3.3 Cambiar Nacionalidad de Director

UPDATE Directors

SET nationality = 'Estados Unidos/Alemania'

WHERE name = 'Quentin Tarantino';

4. Consultas de Eliminación

4.1 Eliminar Película con Dependencias

-- Eliminar relaciones

DELETE FROM Movies\_Actors WHERE movie\_id = (SELECT movie\_id FROM Movies WHERE title = 'Pulp Fiction');

DELETE FROM Movies\_Directors WHERE movie\_id = (SELECT movie\_id FROM Movies WHERE title = 'Pulp Fiction');

DELETE FROM MovieCopies WHERE movie\_id = (SELECT movie\_id FROM Movies WHERE title = 'Pulp Fiction');

-- Eliminar película

DELETE FROM Movies WHERE title = 'Pulp Fiction';

4.2 Eliminar Actor

-- Primero eliminar referencias

DELETE FROM Movies\_Actors WHERE actor\_id = (SELECT actor\_id FROM Actors WHERE name = 'John Travolta');

DELETE FROM Actors WHERE name = 'John Travolta';

Consideraciones Importantes

Integridad Referencial

- Respetar relaciones entre tablas

- Usar transacciones

- Validar datos antes de insertarlos

Buenas Prácticas

- Usar parámetros preparados

- Validar entrada de datos

- Manejar excepciones

- Utilizar transacciones para operaciones complejas

Ejemplos de Transacción

BEGIN;

-- Operaciones

INSERT INTO Movies (...);

INSERT INTO Movies\_Directors (...);

INSERT INTO Actors (...);

COMMIT;

Consejos Adicionales

1. Usar `ON CONFLICT` para manejar inserciones duplicadas

2. Implementar validaciones a nivel de base de datos

3. Usar índices para mejorar rendimiento

4. Mantener consistencia en tipos de datos