Sistema Gestor de Discografía - Documentación Técnica

Bibliotecas Utilizadas

iostream

Biblioteca estándar de C++ que proporciona funcionalidades básicas de entrada y salida. Se utiliza para:

- Interacción con el usuario a través de la consola

- Mostrar información mediante `cout`

- Recibir datos con `cin`

pqxx/pqxx

Biblioteca oficial de C++ para PostgreSQL que ofrece:

- Interfaz orientada a objetos para interactuar con bases de datos

- Conexión segura a bases de datos PostgreSQL

- Ejecución de consultas

- Manejo de transacciones

cstdlib

Biblioteca de utilidades generales del sistema que permite:

- Ejecutar comandos del sistema operativo

- Gestionar procesos del sistema

Funciones de Utilidad

Funciones de Interfaz

`limpiarPantalla()`

void limpiarPantalla() {

system("clear");

}

Limpia la pantalla de la consola utilizando el comando "clear". Mantiene una interfaz limpia y organizada durante la ejecución del programa.

`limpiarEntrada()`

void limpiarEntrada() {

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

Función crucial para la gestión del buffer de entrada:

- Limpia el estado del flujo de entrada

- Elimina caracteres residuales después de operaciones de lectura

- Previene problemas en lecturas secuenciales de datos

`esperarEntrada()`

void esperarEntrada() {

cout << "\nPresione Enter para continuar...";

cin.get();

}

Pausa la ejecución del programa hasta que el usuario presione Enter:

- Permite al usuario leer mensajes en pantalla

- Proporciona una experiencia de usuario más controlada

Estructura de Base de Datos

El sistema gestiona cinco tablas principales:

1. Companies: Información de compañías discográficas

- `company\_id`: Identificador único

- `name`: Nombre de la compañía

- `address`: Dirección de la compañía

2. Recordings: Detalles de grabaciones musicales

- `recording\_id`: Identificador único

- `title`: Título de la grabación

- `musical\_category`: Categoría musical

- `track\_count`: Número de temas

- `description`: Descripción opcional

- `company\_id`: Referencia a la compañía

3. Performers: Información de intérpretes

- `performer\_id`: Identificador único

- `name`: Nombre del intérprete

- `biography`: Biografía opcional

4. Performances: Relación entre grabaciones e intérpretes

- `recording\_id`: Grabación

- `performer\_id`: Intérprete

- `participation\_date`: Fecha de participación

5. PhysicalCopies: Copias físicas de grabaciones

- `copy\_id`: Identificador único

- `recording\_id`: Grabación relacionada

- `format\_id`: Formato de la copia

- `conservation\_state`: Estado de conservación

Funciones Principales

Conexión a Base de Datos

`conectar()`

Establece la conexión con la base de datos PostgreSQL:

- Configura parámetros de conexión

- Verifica conexión exitosa

- Maneja errores de conexión

Funciones de Visualización

`mostrarCatalogo()`

Muestra el catálogo completo de grabaciones:

- Utiliza joins para relacionar grabaciones con compañías

- Presenta información detallada de cada grabación

- Maneja valores nulos con `COALESCE()`

`mostrarInterpretes()`

Presenta información de intérpretes:

- Muestra nombre y biografía

- Lista grabaciones por intérprete

- Gestiona casos sin grabaciones o biografía

`mostrarCopiasFisicas()`

Exhibe el inventario de copias físicas:

- Detalla título, formato y estado de conservación

- Incluye información de la compañía

Gestión de Grabaciones

`registrarGrabacion()`

Proceso completo de registro de una nueva grabación:

1. Recopilación de información

- Validación de título

- Categoría musical

- Número de temas

2. Selección de compañía

3. Registro de intérpretes

4. Registro de copias físicas

5. Validaciones de entrada

6. Manejo de transacciones

`eliminarGrabacion()`

Proceso seguro de eliminación:

- Listado de grabaciones

- Confirmación del usuario

- Verificación de dependencias

- Eliminación en cascada

- Manejo de transacciones

Menú Principal

Características

- Interfaz clara y organizada

- Validación robusta de entrada

- Manejo de errores

- Opciones bien definidas

Opciones

1. Explorar Catálogo Completo

2. Directorio de Intérpretes

3. Inventario de Copias Físicas

4. Registrar Nueva Grabación

5. Gestionar Eliminaciones

0. Cerrar Sistema

Consideraciones de Diseño

Principios Implementados

- Modularidad: Funciones específicas con responsabilidades claras

- Manejo de Errores:

- Uso consistente de try-catch

- Mensajes descriptivos

- Seguridad de Datos:

- Transacciones para operaciones críticas

- Validación de entradas

- Usabilidad:

- Interfaz intuitiva

- Retroalimentación constante al usuario

Análisis Detallado de Consultas SQL en el Sistema de Gestión Discográfica

Consultas de Creación de Tablas

1. Tabla de Compañías

CREATE TABLE IF NOT EXISTS Companies (

company\_id SERIAL PRIMARY KEY,

name VARCHAR(100) NOT NULL,

address TEXT NOT NULL,

UNIQUE(name)

);

Propósito: Almacenar información de compañías discográficas

- `SERIAL`: Genera automáticamente un ID único incremental

- `VARCHAR(100)`: Limita el nombre a 100 caracteres

- `NOT NULL`: Campos obligatorios

- `UNIQUE(name)`: Previene nombres de compañía duplicados

- Ejemplo: Registra compañías como "Sony Music" o "Universal Records"

2. Tabla de Grabaciones

CREATE TABLE IF NOT EXISTS Recordings (

recording\_id SERIAL PRIMARY KEY,

title VARCHAR(100) NOT NULL,

musical\_category VARCHAR(50) NOT NULL,

track\_count INTEGER NOT NULL CHECK (track\_count > 0),

description TEXT,

company\_id INTEGER REFERENCES Companies(company\_id),

UNIQUE(title)

);

Detalles Importantes:

- `CHECK (track\_count > 0)`: Garantiza que toda grabación tenga al menos un tema

- `REFERENCES Companies(company\_id)`: Establece relación con tabla de compañías

- `UNIQUE(title)`: Previene grabaciones con títulos idénticos

3. Tabla de Intérpretes

CREATE TABLE IF NOT EXISTS Performers (

performer\_id SERIAL PRIMARY KEY,

name VARCHAR(100) NOT NULL,

biography TEXT,

UNIQUE(name)

);

Características:

- Permite almacenar información detallada de intérpretes

- Biografía opcional

- Nombre único para evitar duplicados

Consultas de Visualización Detalladas

Consulta de Catálogo de Grabaciones

SELECT

r.recording\_id,

r.title,

r.musical\_category,

r.track\_count,

COALESCE(r.description, 'Sin descripción') AS description,

COALESCE(c.name, 'Sin compañía') AS company\_name,

COALESCE(string\_agg(DISTINCT p.name, ', '), 'Sin intérpretes') AS performers,

COALESCE(string\_agg(DISTINCT f.name, ', '), 'Sin formatos') AS available\_formats

FROM Recordings r

LEFT JOIN Companies c ON r.company\_id = c.company\_id

LEFT JOIN Performances perf ON r.recording\_id = perf.recording\_id

LEFT JOIN Performers p ON perf.performer\_id = p.performer\_id

LEFT JOIN PhysicalCopies pc ON r.recording\_id = pc.recording\_id

LEFT JOIN Formats f ON pc.format\_id = f.format\_id

GROUP BY r.recording\_id, r.title, r.musical\_category,

r.track\_count, r.description, c.name

ORDER BY r.title;

Análisis Detallado de la Consulta

Propósito: Recuperar información completa de grabaciones con sus relaciones

Funciones SQL Avanzadas:

- `COALESCE()`: Proporciona valores por defecto para campos nulos

- `'Sin descripción'` si no hay descripción

- `'Sin compañía'` si no hay compañía asociada

- `string\_agg()`: Concatena valores de múltiples filas en una cadena

- Agrupa intérpretes y formatos en listas separadas por coma

Joins Utilizados:

- `LEFT JOIN`: Mantiene todos los registros de la tabla principal (Recordings)

- Conecta grabaciones con:

1. Compañías

2. Interpretaciones

3. Intérpretes

4. Copias físicas

5. Formatos

Agrupamiento y Ordenamiento:

- `GROUP BY`: Agrupa resultados para evitar duplicados

- `ORDER BY r.title`: Ordena alfabéticamente por título de grabación

Consulta de Intérpretes

SELECT

p.performer\_id,

p.name,

COALESCE(p.biography, 'Sin biografía') AS biography,

COUNT(DISTINCT perf.recording\_id) as recording\_count,

COALESCE(string\_agg(DISTINCT r.title, ', '), 'Sin grabaciones') AS recordings

FROM Performers p

LEFT JOIN Performances perf ON p.performer\_id = perf.performer\_id

LEFT JOIN Recordings r ON perf.recording\_id = r.recording\_id

GROUP BY p.performer\_id, p.name, p.biography

ORDER BY p.name;

Características Destacadas:

- Cuenta grabaciones por intérprete

- Genera lista de grabaciones

- Maneja casos sin grabaciones

- Ordena alfabéticamente por nombre

Consulta de Copias Físicas

SELECT

pc.copy\_id,

r.title,

f.name as format,

pc.conservation\_state,

COALESCE(c.name, 'Sin compañía') as company

FROM PhysicalCopies pc

JOIN Recordings r ON pc.recording\_id = r.recording\_id

JOIN Formats f ON pc.format\_id = f.format\_id

LEFT JOIN Companies c ON r.company\_id = c.company\_id

ORDER BY r.title, f.name;

Funcionalidades:

- Detalla copias físicas con información de grabación

- Incluye formato y estado de conservación

- Maneja casos sin compañía asociada

Patrones de Consulta Recurrentes

Estrategias de Manejo de Datos Nulos

1. `COALESCE()`: Proporciona valores por defecto

2. `LEFT JOIN`: Preserva registros sin relaciones completas

3. `string\_agg()`: Concatena valores múltiples

Validaciones Integradas

- Restricciones de integridad referencial

- Prevención de duplicados

- Validaciones de datos al nivel de base de datos

Consideraciones Técnicas

Rendimiento

- Uso de índices implícitos (PRIMARY KEY, UNIQUE)

- Joins eficientes

- Agrupamiento para reducir duplicados

Flexibilidad

- Esquema que permite registros con información parcial

- Relaciones flexibles entre entidades

Seguridad

- Restricciones de tipo

- Prevención de datos inconsistentes

- Manejo graciosos de escenarios sin datos completos