

Explicación del desarrollo del dbdiagram

Contenedor corriendo

The screenshot shows the Docker Hub interface with the following details:

- Container CPU usage:** 0.42% / 800% (8 CPUs available)
- Container memory usage:** 374.9MB / 3.74GB
- Show charts:** Link to view charts.
- Search bar:** Search for containers.
- Filter:** Only show running containers.
- Table of running containers:**

Name	Container ID	Image	Port(s)	CPU (%)	Last st:	Actions
logistoredb	-	-	-	0.42%	3 hours	[Edit]
mysql_logistic	6ccb53eef47c	mysql:8.0	3306:3306	0.42%	3 hours	[Edit]
- Walkthroughs:**
 - Multi-container applications (8 mins)
 - \$ docker init (3 mins)
- [View more in the Learning center](#)

La problemática nos indica que se debe buscar una solución para la plataforma de streaming Musical -SoundStream la cual presenta fallas en la composición de su estructura a continuación se pensara en la solución mediante diagramas el cual se utilizara para resolver la estructura y poder continuar con el desarrollo de dicha solución

Como primera instancia creamos una tabla de usuarios en las cual contendrá los campos necesarios para mantener un control así mismo poder saber en qué parte o instancia se presenta un problema. Se pensó en una estructura que pueda cumplir con las normativas que cumplan la 1N, 2N Y 3N las cuales fueron necesarias para el desarrollo de la problemática. De acuerdo con la información proporcionada en el documento (los errores señalados en la documentación). Se pudo realizar las tablas así como sus parámetros necesarios y la relación entre ellos para poder tener un control y registro de todos los procesos. Cabe recalcar que mínimo son 7 tablas de acuerdo ala problemática presentada

```
Table usuarios {
    id_usuario int [pk, increment]
    nombre varchar(100)
    correo varchar(100) [unique]
    pais varchar(50)
```

```
fecha_nacimiento date
fecha_registro datetime
created_at datetime
created_by varchar(50)
updated_at datetime
updated_by varchar(50)
active boolean
}
```

```
Table roles {
    id_rol int [pk, increment]
    nombre varchar(50) [unique]
    descripcion varchar(100)
    created_at datetime
    created_by varchar(50)
    updated_at datetime
    updated_by varchar(50)
    active boolean
}
```

```
Table usuarios_roles {
    id_usuario int [ref: > usuarios.id_usuario]
    id_rol int [ref: > roles.id_rol]
    created_at datetime
    created_by varchar(50)
    updated_at datetime
    updated_by varchar(50)
    active boolean
    primary key (id_usuario, id_rol)
}
```

```
Table artistas {
    id_artista int [pk, increment]
    nombre varchar(100)
    genero_id int [ref: > generos.id_genero]
    created_at datetime
    created_by varchar(50)
    updated_at datetime
    updated_by varchar(50)
    active boolean
}
```

```
Table generos {
```

```
id_genero int [pk, increment]
nombre varchar(50) [unique]
created_at datetime
created_by varchar(50)
updated_at datetime
updated_by varchar(50)
active boolean
}
```

```
Table albums {
    id_album int [pk, increment]
    titulo varchar(100)
    id_artista int [ref: > artistas.id_artista]
    id_genero int [ref: > generos.id_genero]
    portada varchar(255)
    fecha_lanzamiento date
    created_at datetime
    created_by varchar(50)
    updated_at datetime
    updated_by varchar(50)
    active boolean
}
```

```
Table canciones {
    id_cancion int [pk, increment]
    titulo varchar(100)
    id_album int [ref: > albums.id_album]
    id_artista int [ref: > artistas.id_artista]
    duracion int
    es_publica boolean
    created_at datetime
    created_by varchar(50)
    updated_at datetime
    updated_by varchar(50)
    active boolean
}
```

```
Table playlists {
    id_playlist int [pk, increment]
    nombre varchar(100)
    id_usuario int [ref: > usuarios.id_usuario]
    created_at datetime
    created_by varchar(50)
    updated_at datetime
```

```
updated_by varchar(50)
active boolean
}

Table playlists_canciones {
    id_playlist int [ref: > playlists.id_playlist]
    id_cancion int [ref: > canciones.id_cancion]
    orden int
    created_at datetime
    created_by varchar(50)
    updated_at datetime
    updated_by varchar(50)
    active boolean
    primary key (id_playlist, id_cancion)
}

Table escuchas {
    id_escucha int [pk, increment]
    id_usuario int [ref: > usuarios.id_usuario]
    id_cancion int [ref: > canciones.id_cancion]
    timestamp datetime
    duracion int
    created_at datetime
    created_by varchar(50)
    updated_at datetime
    updated_by varchar(50)
    active boolean
}

Table suscripciones {
    id_suscripcion int [pk, increment]
    id_usuario int [ref: > usuarios.id_usuario]
    tipo varchar(20)
    fecha_inicio date
    fecha_fin date
    estado varchar(20)
    created_at datetime
    created_by varchar(50)
    updated_at datetime
    updated_by varchar(50)
    active boolean
}
```

Evidencia de dbDiagram

Untitled Diagram

Save Import Export Publish to dbdocs

dbdiagram AI Help Sign in

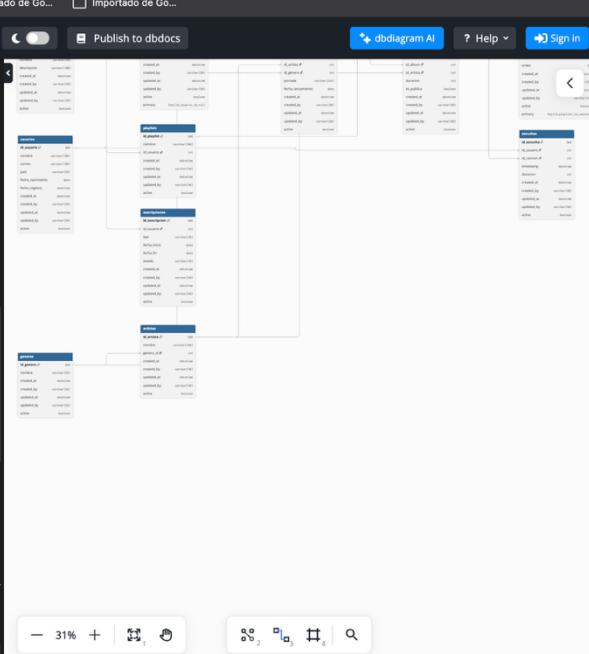
```

1 // Cantooooooo es guapo poderoso, vigoroso
2 // Usuarios (1)
3 // Se realizan validaciones para tener mayor control
4 Table usuarios {
5     id_usuario int [pk, increment]
6     nombre varchar(100)
7     correo varchar(100) [unique]
8     pais varchar(50)
9     fecha_nacimiento date
10    fecha_registro datetime
11    created_at datetime
12    created_by varchar(50)
13    updated_at datetime
14    updated_by varchar(50)
15    active boolean
16 }
17 }
18 // Roles (2)
19 //De acuerdo a las caracteristicas(Privilegios de suscripciones que ofrece)
20 //son las cosas que usuario podra o no podra hacer
21
22 Table roles {
23     id_rol int [pk, increment]
24     nombre varchar(50) [unique]
25     descripcion varchar(100)
26     created_at datetime
27     created_by varchar(50)
28     updated_at datetime
29     updated_by varchar(50)
30     active boolean
31 }
32
33 Table usuarios_roles {
34     id_usuario int [ref: > usuarios.id_usuario]
35     id_rol int [ref: > roles.id_rol]
36     created_at datetime
37     created_by varchar(50)
38     updated_at datetime
39     updated_by varchar(50)
40     active boolean
41     primary key (id_usuario, id_rol)
42 }
43 // Artistas (3)
44 // Para tener el registro de los artistas como la musica
45 Table artistas {
46     id_artista int [pk, increment]
47     nombre varchar(100)
48     genero_id int [ref: > generos.id_genero]
49     created_at datetime
50     created_by varchar(50)
51     updated_at datetime
52     updated_by varchar(50)
53     active boolean
54 }
55 // Generos (4)
56 // Tipos de generos de acuerdo a los generos existentes en la app
57 Table generos {
58     id_genero int [pk, increment]
59     nombre varchar(50) [unique]
60     created_at datetime
61     created_by varchar(50)
62     updated_at datetime
63     updated_by varchar(50)
64     active boolean
65 }
66 // Albums (5)
67 // Lo que contenga album asi como genero,artista, entre otros parametros
68 //para tener un registro
69 Table albums {
70     id_album int [pk, increment]
71     titulo varchar(100)
72     id_artista int [ref: > artistas.id_artista]
73 }
```

The database diagram illustrates the relationships between the following tables:

- usuarios**: Primary key is id_usuario.
- roles**: Primary key is id_rol.
- generos**: Primary key is id_genero.
- artistas**: Primary key is id_artista.
- albums**: Primary key is id_album.
- usuarios_roles**: Primary key is (id_usuario, id_rol). It has foreign keys referencing id_usuario from usuarios and id_rol from roles.
- generos_artistas**: Primary key is (id_genero, id_artista). It has foreign keys referencing id_genero from generos and id_artista from artistas.
- albums_genres**: Primary key is (id_album, id_genero). It has foreign keys referencing id_album from albums and id_genero from generos.

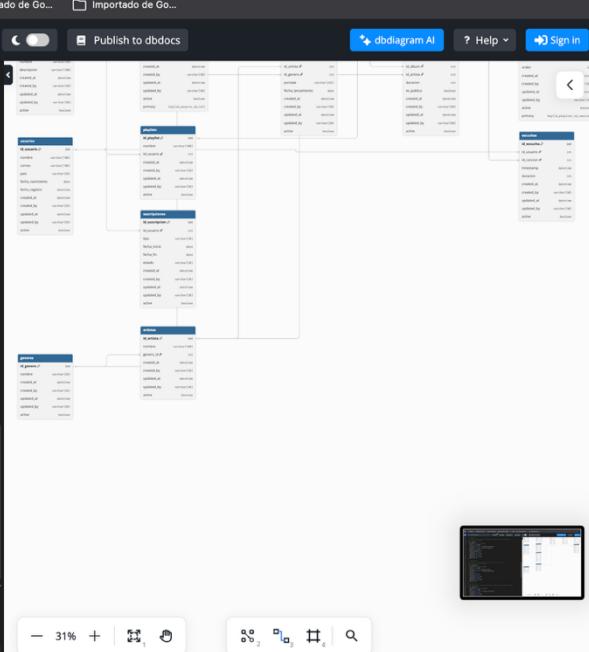
Relationships are indicated by lines connecting the tables, such as a line from the **usuarios** table to the **usuarios_roles** table, and another from the **artistas** table to the **generos_artistas** table.



```

66 // Albums (5)
67 // Lo que contenga album asi como genero,artista, entre otros parametros
68 //para tener un registro
69 Table albums {
70     id_album int [pk, increment]
71     titulo varchar(100)
72     id_artista int [ref: > artistas.id_artista]
73     id_genero int [ref: > generos.id_genero]
74     portada varchar(255)
75     fecha_lanzamiento date
76     created_at datetime
77     created_by varchar(50)
78     updated_at datetime
79     updated_by varchar(50)
80     active boolean
81 }
82 // Canciones (6)
83 // Canciones se crea registros de ls canciones para tener control
84 //así mismo su genero y apartado
85 Table canciones {
86     id_cancion int [pk, increment]
87     titulo varchar(100)
88     id_album int [ref: > albums.id_album]
89     id_artista int [ref: > artistas.id_artista]
90     duracion int
91     es_publica boolean
92     created_at datetime
93     created_by varchar(50)
94     updated_at datetime
95     updated_by varchar(50)
96     active boolean
97 }
98 // Playlist (7)
99 // Se tiene registro de estas para evitar errores o duplicaciones de la misma
100 Table playlists {
101     id_playlist int [pk, increment]
102     nombre varchar(100)
103     id_usuario int [ref: > usuarios.id_usuario]
104     created_at datetime
105     created_by varchar(50)
106     updated_at datetime
107     updated_by varchar(50)
108     active boolean
109 }
110 Table playlists_canciones {
111     id_playlist int [ref: > playlists.id_playlist]
112     id_cancion int [ref: > canciones.id_cancion]
113     orden int
114     created_at datetime
115     created_by varchar(50)
116     updated_at datetime
117     updated_by varchar(50)
118     active boolean
119     primary key (id_playlist, id_cancion)
120 }
121 // Escuchas (8)
122 // se tiene un control de lo que el usuario esta escuchando (registro)
123 //para tener un control de igual manera (se pueda visualizar )
124 Table escuchas {
125     id_escucha int [pk, increment]
126     id_usuario int [ref: > usuarios.id_usuario]
127     id_cancion int [ref: > canciones.id_cancion]
128     timestamp datetime
129     duracion int
130     created_at datetime
131     created_by varchar(50)
132     updated_at datetime
133     updated_by varchar(50)
134     active boolean
135 }
136

```



Evidencia de los scripts

Creacion de las Tablas mediante DbDiagram la cual se utilizara para subirlo en scripts
 Lo siguiente es el codigo mediante las tablas

```
CREATE TABLE IF NOT EXISTS generos (
    id_genero INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(50) UNIQUE,
    created_at DATETIME,
    created_by VARCHAR(50),
    updated_at DATETIME,
    updated_by VARCHAR(50),
    active BOOLEAN
);
```

```
CREATE TABLE IF NOT EXISTS artistas (
    id_artista INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100),
    genero_id INT,
    created_at DATETIME,
    created_by VARCHAR(50),
    updated_at DATETIME,
    updated_by VARCHAR(50),
    active BOOLEAN,
    FOREIGN KEY (genero_id) REFERENCES generos(id_genero)
);
```

```
CREATE TABLE IF NOT EXISTS albums (
    id_album INT AUTO_INCREMENT PRIMARY KEY,
    titulo VARCHAR(100),
    id_artista INT,
    id_genero INT,
    portada VARCHAR(255),
    fecha_lanzamiento DATE,
    created_at DATETIME,
    created_by VARCHAR(50),
    updated_at DATETIME,
    updated_by VARCHAR(50),
    active BOOLEAN,
    FOREIGN KEY (id_artista) REFERENCES artistas(id_artista),
    FOREIGN KEY (id_genero) REFERENCES generos(id_genero)
);
```

```
CREATE TABLE IF NOT EXISTS canciones (
    id_cancion INT AUTO_INCREMENT PRIMARY KEY,
    titulo VARCHAR(100),
    id_album INT,
    id_artista INT,
    duracion INT,
```

```
    es_publica BOOLEAN,  
    created_at DATETIME,  
    created_by VARCHAR(50),  
    updated_at DATETIME,  
    updated_by VARCHAR(50),  
    active BOOLEAN,  
    FOREIGN KEY (id_album) REFERENCES albums(id_album),  
    FOREIGN KEY (id_artista) REFERENCES artistas(id_artista)  
);
```

```
CREATE TABLE IF NOT EXISTS usuarios (  
    id_usuario INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(100),  
    correo VARCHAR(100) UNIQUE,  
    pais VARCHAR(50),  
    fecha_nacimiento DATE,  
    fecha_registro DATETIME,  
    created_at DATETIME,  
    created_by VARCHAR(50),  
    updated_at DATETIME,  
    updated_by VARCHAR(50),  
    active BOOLEAN  
);
```

```
CREATE TABLE IF NOT EXISTS roles (  
    id_rol INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(50) UNIQUE,  
    descripcion VARCHAR(100),  
    created_at DATETIME,  
    created_by VARCHAR(50),  
    updated_at DATETIME,  
    updated_by VARCHAR(50),  
    active BOOLEAN  
);
```

```
CREATE TABLE IF NOT EXISTS usuarios_roles (  
    id_usuario INT,  
    id_rol INT,  
    created_at DATETIME,  
    created_by VARCHAR(50),  
    updated_at DATETIME,  
    updated_by VARCHAR(50),  
    active BOOLEAN,  
    PRIMARY KEY (id_usuario, id_rol),
```

```
FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario),
FOREIGN KEY (id_rol) REFERENCES roles(id_rol)
);
```

```
CREATE TABLE IF NOT EXISTS playlists (
    id_playlist INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100),
    id_usuario INT,
    created_at DATETIME,
    created_by VARCHAR(50),
    updated_at DATETIME,
    updated_by VARCHAR(50),
    active BOOLEAN,
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)
);
```

```
CREATE TABLE IF NOT EXISTS playlists_canciones (
    id_playlist INT,
    id_cancion INT,
    orden INT,
    created_at DATETIME,
    created_by VARCHAR(50),
    updated_at DATETIME,
    updated_by VARCHAR(50),
    active BOOLEAN,
    PRIMARY KEY (id_playlist, id_cancion),
    FOREIGN KEY (id_playlist) REFERENCES playlists(id_playlist),
    FOREIGN KEY (id_cancion) REFERENCES canciones(id_cancion)
);
```

```
CREATE TABLE IF NOT EXISTS escuchas (
    id_escucha INT AUTO_INCREMENT PRIMARY KEY,
    id_usuario INT,
    id_cancion INT,
    timestamp DATETIME,
    duracion INT,
    created_at DATETIME,
    created_by VARCHAR(50),
    updated_at DATETIME,
    updated_by VARCHAR(50),
    active BOOLEAN,
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario),
    FOREIGN KEY (id_cancion) REFERENCES canciones(id_cancion)
);
```

```
CREATE TABLE IF NOT EXISTS suscripciones (
    id_suscripcion INT AUTO_INCREMENT PRIMARY KEY,
    id_usuario INT,
    tipo VARCHAR(20),
    fecha_inicio DATE,
    fecha_fin DATE,
    estado VARCHAR(20),
    created_at DATETIME,
    created_by VARCHAR(50),
    updated_at DATETIME,
    updated_by VARCHAR(50),
    active BOOLEAN,
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)
);
```

Evidencia de los scripts

Scripts > Tablas.sql

```
1  CREATE TABLE IF NOT EXISTS generos (
2    id_genero INT AUTO_INCREMENT PRIMARY KEY,
3    nombre VARCHAR(50) UNIQUE,
4    created_at DATETIME,
5    created_by VARCHAR(50),
6    updated_at DATETIME,
7    updated_by VARCHAR(50),
8    active BOOLEAN
9  );
10 ;
11
12 CREATE TABLE IF NOT EXISTS artistas (
13    id_artista INT AUTO_INCREMENT PRIMARY KEY,
14    nombre VARCHAR(100),
15    genero_id INT,
16    created_at DATETIME,
17    created_by VARCHAR(50),
18    updated_at DATETIME,
19    updated_by VARCHAR(50),
20    active BOOLEAN,
21    FOREIGN KEY (genero_id) REFERENCES generos(id_genero)
22 );
23
24 CREATE TABLE IF NOT EXISTS albums (
25    id_album INT AUTO_INCREMENT PRIMARY KEY,
26    titulo VARCHAR(100),
27    id_artista INT,
28    id_genero INT,
29    portada VARCHAR(255),
30    fecha_lanzamiento DATE,
31    created_at DATETIME,
32    created_by VARCHAR(50),
33    updated_at DATETIME,
34    updated_by VARCHAR(50),
35    active BOOLEAN,
36    FOREIGN KEY (id_artista) REFERENCES artistas(id_artista),
37    FOREIGN KEY (id_genero) REFERENCES generos(id_genero)
38 );
```

Lín. 19, col. 5 Espacios: 4 UTF-8 CRLF {} MS SQL

Scripts > Tablas.sql

```
40 CREATE TABLE IF NOT EXISTS canciones (
41     id_cancion INT AUTO_INCREMENT PRIMARY KEY,
42     titulo VARCHAR(100),
43     id_album INT,
44     id_artista INT,
45     duracion INT,
46     es_publica BOOLEAN,
47     created_at DATETIME,
48     created_by VARCHAR(50),
49     updated_at DATETIME,
50     updated_by VARCHAR(50),
51     active BOOLEAN,
52     FOREIGN KEY (id_album) REFERENCES albums(id_album),
53     FOREIGN KEY (id_artista) REFERENCES artistas(id_artista)
54 );
55
56 CREATE TABLE IF NOT EXISTS usuarios (
57     id_usuario INT AUTO_INCREMENT PRIMARY KEY,
58     nombre VARCHAR(100),
59     correo VARCHAR(100) UNIQUE,
60     pais VARCHAR(50),
61     fecha_nacimiento DATE,
62     fecha_registro DATETIME,
63     created_at DATETIME,
64     created_by VARCHAR(50),
65     updated_at DATETIME,
66     updated_by VARCHAR(50),
67     active BOOLEAN
68 );
69
70 CREATE TABLE IF NOT EXISTS roles (
71     id_rol INT AUTO_INCREMENT PRIMARY KEY,
72     nombre VARCHAR(50) UNIQUE,
73     descripcion VARCHAR(100),
74     created_at DATETIME,
75     created_by VARCHAR(50),
76     updated_at DATETIME,
77     updated_by VARCHAR(50),
```

Lín. 19, col. 5 Espacios: 4 UTF-8 CRLF {} MS SQL

```
Scripts > Tablas.sql
,,,
69
70 CREATE TABLE IF NOT EXISTS roles (
71     id_rol INT AUTO_INCREMENT PRIMARY KEY,
72     nombre VARCHAR(50) UNIQUE,
73     descripcion VARCHAR(100),
74     created_at DATETIME,
75     created_by VARCHAR(50),
76     updated_at DATETIME,
77     updated_by VARCHAR(50),
78     active BOOLEAN
79 );
80
81 CREATE TABLE IF NOT EXISTS usuarios_roles (
82     id_usuario INT,
83     id_rol INT,
84     created_at DATETIME,
85     created_by VARCHAR(50),
86     updated_at DATETIME,
87     updated_by VARCHAR(50),
88     active BOOLEAN,
89     PRIMARY KEY (id_usuario, id_rol),
90     FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario),
91     FOREIGN KEY (id_rol) REFERENCES roles(id_rol)
92 );
93
94 CREATE TABLE IF NOT EXISTS playlists (
95     id_playlist INT AUTO_INCREMENT PRIMARY KEY,
96     nombre VARCHAR(100),
97     id_usuario INT,
98     created_at DATETIME,
99     created_by VARCHAR(50),
100    updated_at DATETIME,
101    updated_by VARCHAR(50),
102    active BOOLEAN,
103    FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)
104 );
105
106 CREATE TABLE IF NOT EXISTS playlists_canciones (
Lín. 19, col. 5 Espacios: 4 UTF-8 CRLF {} MS SQL 🔍 🔍
```

```
Scripts > Tablas.sql
106 CREATE TABLE IF NOT EXISTS playlists_canciones (
107     id_playlist INT,
108     id_cancion INT,
109     orden INT,
110     created_at DATETIME,
111     created_by VARCHAR(50),
112     updated_at DATETIME,
113     updated_by VARCHAR(50),
114     active BOOLEAN,
115     PRIMARY KEY (id_playlist, id_cancion),
116     FOREIGN KEY (id_playlist) REFERENCES playlists(id_playlist),
117     FOREIGN KEY (id_cancion) REFERENCES canciones(id_cancion)
118 );
119
120 CREATE TABLE IF NOT EXISTS escuchas (
121     id_escucha INT AUTO_INCREMENT PRIMARY KEY,
122     id_usuario INT,
123     id_cancion INT,
124     timestamp DATETIME,
125     duracion INT,
126     created_at DATETIME,
127     created_by VARCHAR(50),
128     updated_at DATETIME,
129     updated_by VARCHAR(50),
130     active BOOLEAN,
131     FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario),
132     FOREIGN KEY (id_cancion) REFERENCES canciones(id_cancion)
133 );
134
135 CREATE TABLE IF NOT EXISTS suscripciones (
136     id_suscripcion INT AUTO_INCREMENT PRIMARY KEY,
137     id_usuario INT,
138     tipo VARCHAR(20),
139     fecha_inicio DATE,
140     fecha_fin DATE,
141     estado VARCHAR(20),
142     created_at DATETIME,
143     created_by VARCHAR(50),
```

Lín. 19, col. 5 Espacios: 4 UTF-8 CRLF {} MS SQL 

```
Scripts > Tablas.sql
120 CREATE TABLE IF NOT EXISTS escuchas (
121     FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario),
122     FOREIGN KEY (id_cancion) REFERENCES canciones(id_cancion)
123 );
124
125 CREATE TABLE IF NOT EXISTS suscripciones (
126     id_suscripcion INT AUTO_INCREMENT PRIMARY KEY,
127     id_usuario INT,
128     tipo VARCHAR(20),
129     fecha_inicio DATE,
130     fecha_fin DATE,
131     estado VARCHAR(20),
132     created_at DATETIME,
133     created_by VARCHAR(50),
134     updated_at DATETIME,
135     updated_by VARCHAR(50),
136     active BOOLEAN,
137     FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)
138 );
139
140
141
142
143
144
145
146
147
148 
```

Lín. 19, col. 5 Espacios: 4 UTF-8 CRLF {} MS SQL

Parte de las inserciones

02-inserciones-datas

```
INSERT IGNORE INTO generos (nombre, created_at, created_by, updated_at,
updated_by, active) VALUES
('Rock', NOW(), 'admin', NOW(), 'admin', 1),
('Pop', NOW(), 'admin', NOW(), 'admin', 1),
('Reggaeton', NOW(), 'admin', NOW(), 'admin', 1),
('Jazz', NOW(), 'admin', NOW(), 'admin', 1),
('Electronica', NOW(), 'admin', NOW(), 'admin', 1);
```

```
INSERT IGNORE INTO artistas (nombre, genero_id, created_at, created_by,
updated_at, updated_by, active) VALUES
('Queen', 1, NOW(), 'admin', NOW(), 'admin', 1),
```

```
('Shakira', 2, NOW(), 'admin', NOW(), 'admin', 1),
('Bad Bunny', 3, NOW(), 'admin', NOW(), 'admin', 1),
('Miles Davis', 4, NOW(), 'admin', NOW(), 'admin', 1),
('Daft Punk', 5, NOW(), 'admin', NOW(), 'admin', 1);

INSERT IGNORE INTO albums (titulo, id_artista, id_genero, portada,
fecha_lanzamiento, created_at, created_by, updated_at, updated_by, active) VALUES
('A Night at the Opera', 1, 1, 'queen_operajpg', '1975-11-21', NOW(), 'admin', NOW(),
'admin', 1),
('Laundry Service', 2, 2, 'shakira_laundry.jpg', '2001-11-13', NOW(), 'admin', NOW(),
'admin', 1),
('YHLQMDLG', 3, 3, 'badbunny_yhlqmdlg.jpg', '2020-02-29', NOW(), 'admin', NOW(),
'admin', 1),
('Kind of Blue', 4, 4, 'milesdavis_kindofblue.jpg', '1959-08-17', NOW(), 'admin', NOW(),
'admin', 1),
('Discovery', 5, 5, 'daftpunk_discovery.jpg', '2001-03-12', NOW(), 'admin', NOW(),
'admin', 1);

INSERT IGNORE INTO canciones (titulo, id_album, id_artista, duracion, es_publica,
created_at, created_by, updated_at, updated_by, active) VALUES
('Bohemian Rhapsody', 1, 1, 354, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Somebody to Love', 1, 1, 295, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Whenever, Wherever', 2, 2, 210, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Objection', 2, 2, 220, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Safaera', 3, 3, 300, 1, NOW(), 'admin', NOW(), 'admin', 1),
('La Difícil', 3, 3, 180, 1, NOW(), 'admin', NOW(), 'admin', 1),
('So What', 4, 4, 545, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Blue in Green', 4, 4, 329, 1, NOW(), 'admin', NOW(), 'admin', 1),
('One More Time', 5, 5, 320, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Digital Love', 5, 5, 300, 1, NOW(), 'admin', NOW(), 'admin', 1);

INSERT IGNORE INTO usuarios (nombre, correo, pais, fecha_nacimiento,
fecha_registro, created_at, created_by, updated_at, updated_by, active) VALUES
('Juan Perez', 'juanperez@mail.com', 'México', '1990-05-10', NOW(), NOW(), 'admin',
NOW(), 'admin', 1),
('Ana Gómez', 'anagomez@mail.com', 'Colombia', '1985-08-22', NOW(), NOW(),
'admin', NOW(), 'admin', 1),
('Carlos Ruiz', 'carlosruiz@mail.com', 'Argentina', '2000-01-15', NOW(), NOW(), 'admin',
NOW(), 'admin', 1),
('Lucía Torres', 'luciatorres@mail.com', 'Chile', '1995-12-30', NOW(), NOW(), 'admin',
NOW(), 'admin', 1),
('Pedro Silva', 'pedrosilva@mail.com', 'Perú', '1988-03-05', NOW(), NOW(), 'admin',
NOW(), 'admin', 1);
```

```
INSERT IGNORE INTO roles (nombre, descripcion, created_at, created_by, updated_at, updated_by, active) VALUES
('admin', 'Administrador total', NOW(), 'admin', NOW(), 'admin', 1),
('auditor', 'Auditoría y escritura', NOW(), 'admin', NOW(), 'admin', 1),
('programador', 'Solo lectura', NOW(), 'admin', NOW(), 'admin', 1);
```

```
INSERT IGNORE INTO usuarios_roles (id_usuario, id_rol, created_at, created_by, updated_at, updated_by, active) VALUES
(1, 1, NOW(), 'admin', NOW(), 'admin', 1),
(2, 2, NOW(), 'admin', NOW(), 'admin', 1),
(3, 3, NOW(), 'admin', NOW(), 'admin', 1),
(4, 2, NOW(), 'admin', NOW(), 'admin', 1),
(5, 3, NOW(), 'admin', NOW(), 'admin', 1);
```

```
INSERT IGNORE INTO playlists (nombre, id_usuario, created_at, created_by, updated_at, updated_by, active) VALUES
('Rock Clásico', 1, NOW(), 'admin', NOW(), 'admin', 1),
('Pop Hits', 2, NOW(), 'admin', NOW(), 'admin', 1),
('Reggaeton Party', 3, NOW(), 'admin', NOW(), 'admin', 1),
('Jazz Relax', 4, NOW(), 'admin', NOW(), 'admin', 1),
('Electro Vibes', 5, NOW(), 'admin', NOW(), 'admin', 1);
```

```
INSERT IGNORE INTO playlists_canciones (id_playlist, id_cancion, orden, created_at, created_by, updated_at, updated_by, active) VALUES
(1, 1, 1, NOW(), 'admin', NOW(), 'admin', 1),
(1, 2, 2, NOW(), 'admin', NOW(), 'admin', 1),
(2, 3, 1, NOW(), 'admin', NOW(), 'admin', 1),
(2, 4, 2, NOW(), 'admin', NOW(), 'admin', 1),
(3, 5, 1, NOW(), 'admin', NOW(), 'admin', 1),
(3, 6, 2, NOW(), 'admin', NOW(), 'admin', 1),
(4, 7, 1, NOW(), 'admin', NOW(), 'admin', 1),
(4, 8, 2, NOW(), 'admin', NOW(), 'admin', 1),
(5, 9, 1, NOW(), 'admin', NOW(), 'admin', 1),
(5, 10, 2, NOW(), 'admin', NOW(), 'admin', 1);
```

```
INSERT IGNORE INTO escuchas (id_usuario, id_cancion, timestamp, duracion, created_at, created_by, updated_at, updated_by, active) VALUES
(1, 1, '2025-01-10 10:00:00', 354, NOW(), 'admin', NOW(), 'admin', 1),
(2, 3, '2025-02-15 12:30:00', 210, NOW(), 'admin', NOW(), 'admin', 1),
(3, 5, '2025-03-20 15:45:00', 300, NOW(), 'admin', NOW(), 'admin', 1),
(4, 7, '2025-04-25 18:00:00', 545, NOW(), 'admin', NOW(), 'admin', 1),
(5, 9, '2025-05-30 20:15:00', 320, NOW(), 'admin', NOW(), 'admin', 1);
```

```
INSERT IGNORE INTO suscripciones (id_usuario, tipo, fecha_inicio, fecha_fin, estado, created_at, created_by, updated_at, updated_by, active) VALUES  
(1, 'premium', '2025-01-01', '2025-12-31', 'activa', NOW(), 'admin', NOW(), 'admin', 1),  
(2, 'free', '2025-01-01', NULL, 'activa', NOW(), 'admin', NOW(), 'admin', 1),  
(3, 'familiar', '2025-01-01', '2025-06-30', 'cancelada', NOW(), 'admin', NOW(), 'admin', 1),  
(4, 'estudiantil', '2025-01-01', '2025-12-31', 'activa', NOW(), 'admin', NOW(), 'admin', 1),  
(5, 'premium', '2025-01-01', '2025-03-31', 'cancelada', NOW(), 'admin', NOW(), 'admin', 1);
```

Evidencia

Scripts > 02-insert.sql

```
1
2  INSERT IGNORE INTO generos (nombre, created_at, created_by, updated_at,
3      updated_by, active) VALUES
4      ('Rock', NOW(), 'admin', NOW(), 'admin', 1),
5      ('Pop', NOW(), 'admin', NOW(), 'admin', 1),
6      ('Reggaeton', NOW(), 'admin', NOW(), 'admin', 1),
7      ('Jazz', NOW(), 'admin', NOW(), 'admin', 1),
8      ('Electronica', NOW(), 'admin', NOW(), 'admin', 1);
9
10
11  INSERT IGNORE INTO artistas (nombre, genero_id, created_at, created_by,
12      updated_at, updated_by, active) VALUES
13      ('Queen', 1, NOW(), 'admin', NOW(), 'admin', 1),
14      ('Shakira', 2, NOW(), 'admin', NOW(), 'admin', 1),
15      ('Bad Bunny', 3, NOW(), 'admin', NOW(), 'admin', 1),
16      ('Miles Davis', 4, NOW(), 'admin', NOW(), 'admin', 1),
17      ('Daft Punk', 5, NOW(), 'admin', NOW(), 'admin', 1);
18
19  INSERT IGNORE INTO albums (titulo, id_artista, id_genero, portada,
20      fecha_lanzamiento, created_at, created_by, updated_at, updated_by, active)
21      VALUES
22      ('A Night at the Opera', 1, 1, 'queen_opera.jpg', '1975-11-21', NOW(), 'admin',
23          NOW(), 'admin', 1),
24      ('Laundry Service', 2, 2, 'shakira_laundry.jpg', '2001-11-13', NOW(), 'admin',
25          NOW(), 'admin', 1),
26      ('YHLQMDLG', 3, 3, 'badbunny_yhlqmdlg.jpg', '2020-02-29', NOW(), 'admin', NOW
27          (), 'admin', 1),
28      ('Kind of Blue', 4, 4, 'milesdavis_kindofblue.jpg', '1959-08-17', NOW(),
29          'admin', NOW(), 'admin', 1),
30      ('Discovery', 5, 5, 'daftpunk_discovery.jpg', '2001-03-12', NOW(), 'admin', NOW
31          (), 'admin', 1);
32
33  INSERT IGNORE INTO canciones (titulo, id_album, id_artista, duracion,
34      es_publica, created_at, created_by, updated_at, updated_by, active) VALUES
35      ('Bohemian Rhapsody', 1, 1, 354, 1, NOW(), 'admin', NOW(), 'admin', 1),
36      ('Somebody to Love', 1, 1, 295, 1, NOW(), 'admin', NOW(), 'admin', 1),
37      ('Whenever, Wherever', 2, 2, 210, 1, NOW(), 'admin', NOW(), 'admin', 1),
38      ('Objection', 2, 2, 220, 1, NOW(), 'admin', NOW(), 'admin', 1),
39      ('Safaera', 3, 3, 300, 1, NOW(), 'admin', NOW(), 'admin', 1),
40      ('La Difícil', 3, 3, 180, 1, NOW(), 'admin', NOW(), 'admin', 1),
41      ('So What', 4, 4, 545, 1, NOW(), 'admin', NOW(), 'admin', 1),
42      ('Blue in Green', 4, 4, 329, 1, NOW(), 'admin', NOW(), 'admin', 1),
43      ('One More Time', 5, 5, 320, 1, NOW(), 'admin', NOW(), 'admin', 1),
```

```
Scripts > 02-insert.sql
-- INSERT IGNORE INTO usuarios (nombre, correo, pais, telefono_mvil,
--     fecha_registro, created_at, created_by, updated_at, updated_by, active)
-- VALUES
40 ('Juan Pérez', 'juanperez@mail.com', 'México', '1990-05-10', NOW(), NOW(),
    'admin', NOW(), 'admin', 1),
41 ('Ana Gómez', 'anagomez@mail.com', 'Colombia', '1985-08-22', NOW(), NOW(),
    'admin', NOW(), 'admin', 1),
42 ('Carlos Ruiz', 'carlosruiz@mail.com', 'Argentina', '2000-01-15', NOW(), NOW(),
    'admin', NOW(), 'admin', 1),
43 ('Lucía Torres', 'luciatorres@mail.com', 'Chile', '1995-12-30', NOW(), NOW(),
    'admin', NOW(), 'admin', 1),
44 ('Pedro Silva', 'pedrosilva@mail.com', 'Perú', '1988-03-05', NOW(), NOW(),
    'admin', NOW(), 'admin', 1);
45
46
47 INSERT IGNORE INTO roles (nombre, descripcion, created_at, created_by,
    updated_at, updated_by, active) VALUES
48 ('admin', 'Administrador total', NOW(), 'admin', NOW(), 'admin', 1),
49 ('auditor', 'Auditoría y escritura', NOW(), 'admin', NOW(), 'admin', 1),
50 ('programador', 'Solo lectura', NOW(), 'admin', NOW(), 'admin', 1);
51
52
53 INSERT IGNORE INTO usuarios_roles (id_usuario, id_rol, created_at, created_by,
    updated_at, updated_by, active) VALUES
54 (1, 1, NOW(), 'admin', NOW(), 'admin', 1),
55 (2, 2, NOW(), 'admin', NOW(), 'admin', 1),
56 (3, 3, NOW(), 'admin', NOW(), 'admin', 1),
57 (4, 2, NOW(), 'admin', NOW(), 'admin', 1),
58 (5, 3, NOW(), 'admin', NOW(), 'admin', 1);
59
60
61 INSERT IGNORE INTO playlists (nombre, id_usuario, created_at, created_by,
    updated_at, updated_by, active) VALUES
62 ('Rock Clásico', 1, NOW(), 'admin', NOW(), 'admin', 1),
63 ('Pop Hits', 2, NOW(), 'admin', NOW(), 'admin', 1),
64 ('Reggaeton Party', 3, NOW(), 'admin', NOW(), 'admin', 1),
65 ('Jazz Relax', 4, NOW(), 'admin', NOW(), 'admin', 1),
66 ('Electro Vibes', 5, NOW(), 'admin', NOW(), 'admin', 1);
67
68
69 INSERT IGNORE INTO playlists_canciones (id_playlist, id_cancion, orden,
    created_at, created_by, updated_at, updated_by, active) VALUES
70 (1, 1, 1, NOW(), 'admin', NOW(), 'admin', 1),
71 (1, 2, 2, NOW(), 'admin', NOW(), 'admin', 1),
72 (2, 3, 1, NOW(), 'admin', NOW(), 'admin', 1),
73 (2, 4, 2, NOW(), 'admin', NOW(), 'admin', 1);
```

```

Scripts > 02-insert.sql
67
68
69  INSERT IGNORE INTO playlists_canciones (id_playlist, id_cancion, orden,
       created_at, created_by, updated_at, updated_by, active) VALUES
70  (1, 1, 1, NOW(), 'admin', NOW(), 'admin', 1),
71  (1, 2, 2, NOW(), 'admin', NOW(), 'admin', 1),
72  (2, 3, 1, NOW(), 'admin', NOW(), 'admin', 1),
73  (2, 4, 2, NOW(), 'admin', NOW(), 'admin', 1),
74  (3, 5, 1, NOW(), 'admin', NOW(), 'admin', 1),
75  (3, 6, 2, NOW(), 'admin', NOW(), 'admin', 1),
76  (4, 7, 1, NOW(), 'admin', NOW(), 'admin', 1),
77  (4, 8, 2, NOW(), 'admin', NOW(), 'admin', 1),
78  (5, 9, 1, NOW(), 'admin', NOW(), 'admin', 1),
79  (5, 10, 2, NOW(), 'admin', NOW(), 'admin', 1);
80
81
82  INSERT IGNORE INTO escuchas (id_usuario, id_cancion, timestamp, duracion,
       created_at, created_by, updated_at, updated_by, active) VALUES
83  (1, 1, '2025-01-10 10:00:00', 354, NOW(), 'admin', NOW(), 'admin', 1),
84  (2, 3, '2025-02-15 12:30:00', 210, NOW(), 'admin', NOW(), 'admin', 1),
85  (3, 5, '2025-03-20 15:45:00', 300, NOW(), 'admin', NOW(), 'admin', 1),
86  (4, 7, '2025-04-25 18:00:00', 545, NOW(), 'admin', NOW(), 'admin', 1),
87  (5, 9, '2025-05-30 20:15:00', 320, NOW(), 'admin', NOW(), 'admin', 1);
88
89
90  INSERT IGNORE INTO suscripciones (id_usuario, tipo, fecha_inicio, fecha_fin,
       estado, created_at, created_by, updated_at, updated_by, active) VALUES
91  (1, 'premium', '2025-01-01', '2025-12-31', 'activa', NOW(), 'admin', NOW(),
       'admin', 1),
92  (2, 'free', '2025-01-01', NULL, 'activa', NOW(), 'admin', NOW(), 'admin', 1),
93  (3, 'familiar', '2025-01-01', '2025-06-30', 'cancelada', NOW(), 'admin', NOW(),
       'admin', 1),
94  (4, 'estudiantil', '2025-01-01', '2025-12-31', 'activa', NOW(), 'admin', NOW(),
       'admin', 1),
95  (5, 'premium', '2025-01-01', '2025-03-31', 'cancelada', NOW(), 'admin', NOW(),
       'admin', 1);

```

Scripts

03-Privilegios

CREATE USER IF NOT EXISTS 'admin_db'@'%' IDENTIFIED BY 'Admin123!';
GRANT ALL PRIVILEGES ON *.* TO 'admin_db'@'%' WITH GRANT OPTION;

CREATE USER IF NOT EXISTS 'auditor_db'@'%' IDENTIFIED BY 'Auditor123!';
GRANT SELECT, INSERT, UPDATE ON tu_base_de_datos.* TO 'auditor_db'@'%';

CREATE USER IF NOT EXISTS 'programador_db'@'%' IDENTIFIED BY
'Programador123!';
GRANT SELECT ON tu_base_de_datos.* TO 'programador_db'@'%';

FLUSH PRIVILEGES;

Evidencia

```
1 CREATE USER IF NOT EXISTS 'admin_db'@'%' IDENTIFIED BY 'Admin123!';
2 GRANT ALL PRIVILEGES ON *.* TO 'admin_db'@'%' WITH GRANT OPTION;
3
4 CREATE USER IF NOT EXISTS 'auditor_db'@'%' IDENTIFIED BY 'Auditor123!';
5 GRANT SELECT, INSERT, UPDATE ON tu_base_de_datos.* TO 'auditor_db'@'%';
6
7 CREATE USER IF NOT EXISTS 'programador_db'@'%' IDENTIFIED BY 'Programador123!';
8 GRANT SELECT ON tu_base_de_datos.* TO 'programador_db'@'%';
9
10
11 FLUSH PRIVILEGES;
```

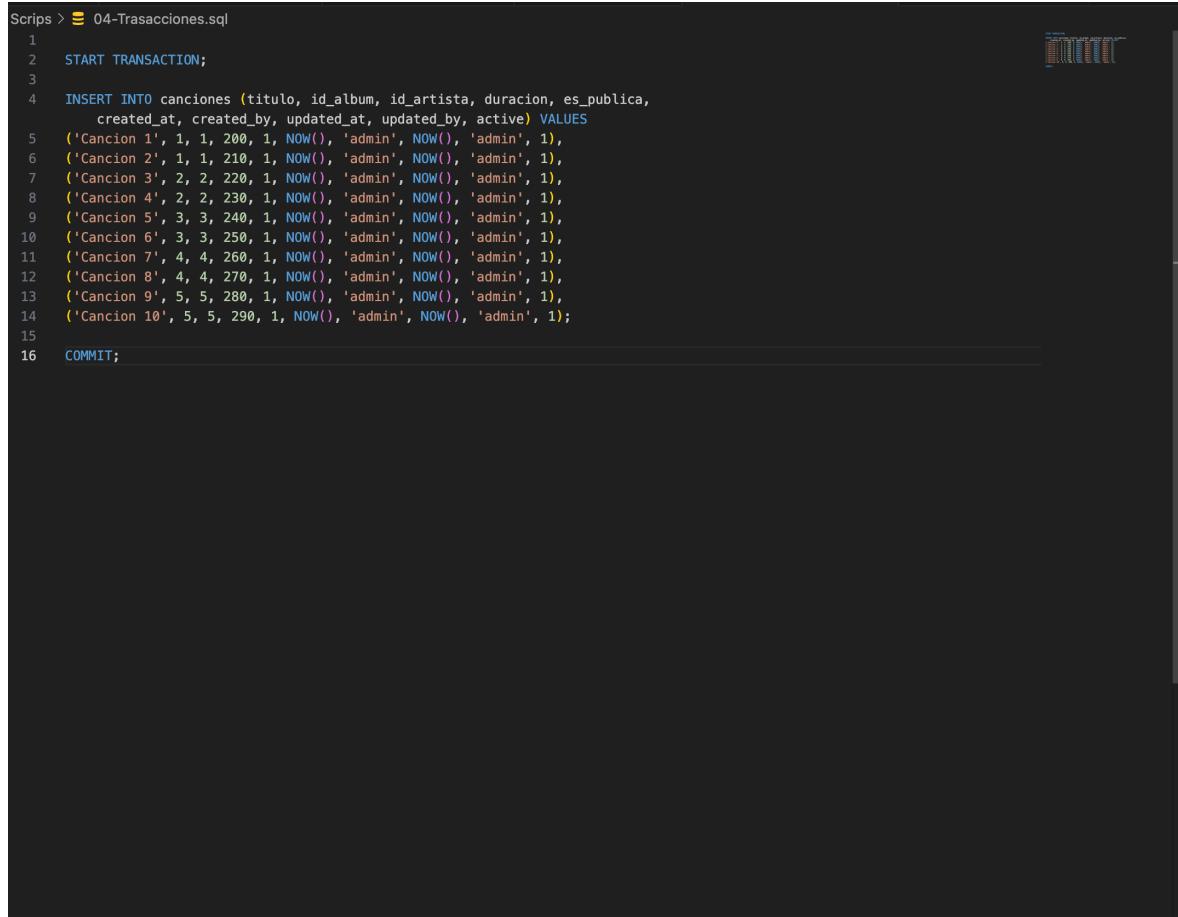
Scripts

04-comints-insert-canciones

START TRANSACTION;

```
INSERT INTO canciones (titulo, id_album, id_artista, duracion, es_publica, created_at,
created_by, updated_at, updated_by, active) VALUES
('Cancion 1', 1, 1, 200, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Cancion 2', 1, 1, 210, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Cancion 3', 2, 2, 220, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Cancion 4', 2, 2, 230, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Cancion 5', 3, 3, 240, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Cancion 6', 3, 3, 250, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Cancion 7', 4, 4, 260, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Cancion 8', 4, 4, 270, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Cancion 9', 5, 5, 280, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Cancion 10', 5, 5, 290, 1, NOW(), 'admin', NOW(), 'admin', 1);
Commit;
```

Evidencia



```
Scripts > 04-Trasacciones.sql
1 START TRANSACTION;
2
3
4 INSERT INTO canciones (titulo, id_album, id_artista, duracion, es_publica,
5 created_at, created_by, updated_at, updated_by, active) VALUES
6 ('Cancion 1', 1, 1, 200, 1, NOW(), 'admin', NOW(), 'admin', 1),
7 ('Cancion 2', 1, 1, 210, 1, NOW(), 'admin', NOW(), 'admin', 1),
8 ('Cancion 3', 2, 2, 220, 1, NOW(), 'admin', NOW(), 'admin', 1),
9 ('Cancion 4', 2, 2, 230, 1, NOW(), 'admin', NOW(), 'admin', 1),
10 ('Cancion 5', 3, 3, 240, 1, NOW(), 'admin', NOW(), 'admin', 1),
11 ('Cancion 6', 3, 3, 250, 1, NOW(), 'admin', NOW(), 'admin', 1),
12 ('Cancion 7', 4, 4, 260, 1, NOW(), 'admin', NOW(), 'admin', 1),
13 ('Cancion 8', 4, 4, 270, 1, NOW(), 'admin', NOW(), 'admin', 1),
14 ('Cancion 9', 5, 5, 280, 1, NOW(), 'admin', NOW(), 'admin', 1),
15 ('Cancion 10', 5, 5, 290, 1, NOW(), 'admin', NOW(), 'admin', 1);
16 COMMIT;
```

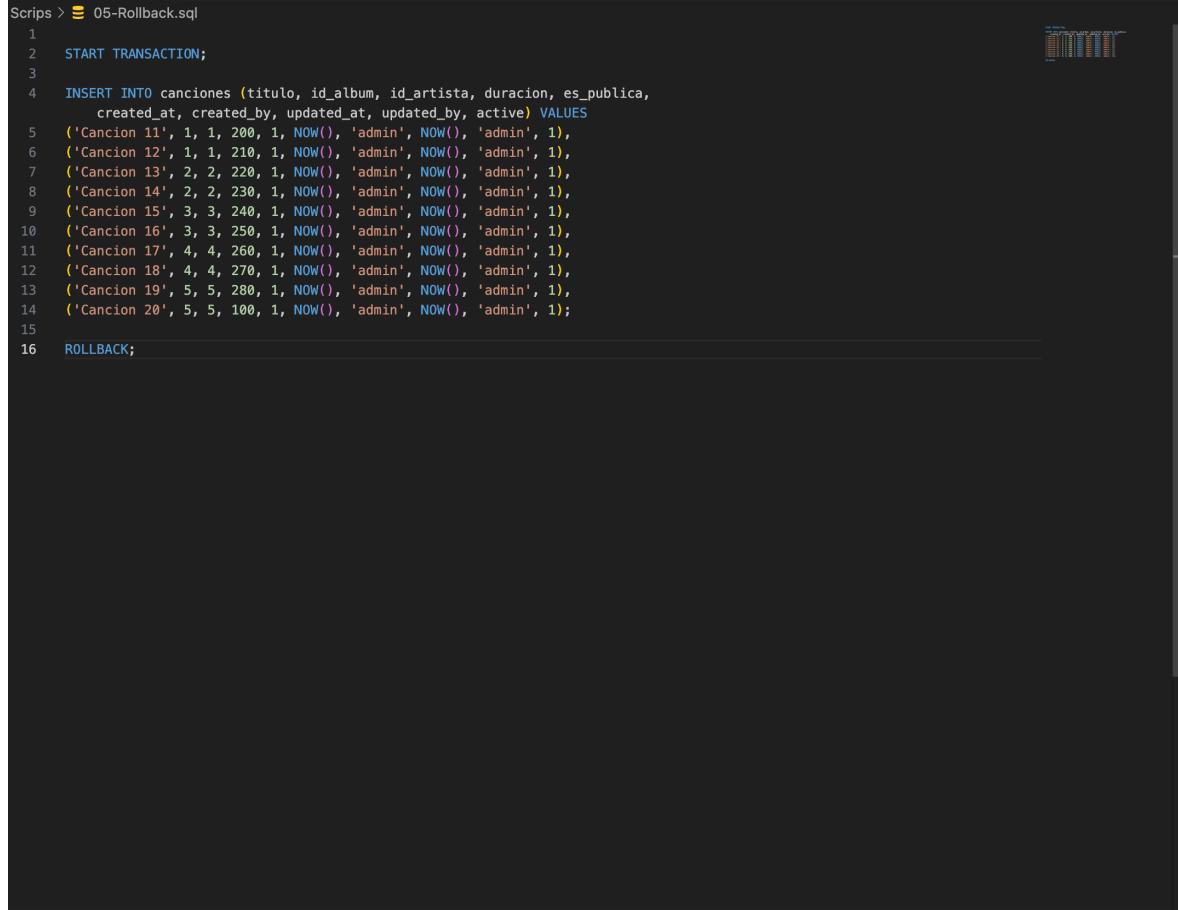
Rollback

```
START TRANSACTION;
```

```
INSERT INTO canciones (titulo, id_album, id_artista, duracion, es_publica, created_at,
created_by, updated_at, updated_by, active) VALUES
('Cancion 11', 1, 1, 200, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Cancion 12', 1, 1, 210, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Cancion 13', 2, 2, 220, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Cancion 14', 2, 2, 230, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Cancion 15', 3, 3, 240, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Cancion 16', 3, 3, 250, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Cancion 17', 4, 4, 260, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Cancion 18', 4, 4, 270, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Cancion 19', 5, 5, 280, 1, NOW(), 'admin', NOW(), 'admin', 1),
('Cancion 20', 5, 5, 100, 1, NOW(), 'admin', NOW(), 'admin', 1);
```

ROLLBACK;

Evidencia



The screenshot shows a MySQL Workbench interface. On the left, there's a 'Scripts' pane with a file named '05-Rollback.sql'. The code in the script is:

```
1
2 START TRANSACTION;
3
4 INSERT INTO canciones (titulo, id_album, id_artista, duracion, es_publica,
5     created_at, created_by, updated_at, updated_by, active) VALUES
6     ('Cancion 11', 1, 1, 200, 1, NOW(), 'admin', NOW(), 'admin', 1),
7     ('Cancion 12', 1, 1, 210, 1, NOW(), 'admin', NOW(), 'admin', 1),
8     ('Cancion 13', 2, 2, 220, 1, NOW(), 'admin', NOW(), 'admin', 1),
9     ('Cancion 14', 2, 2, 230, 1, NOW(), 'admin', NOW(), 'admin', 1),
10    ('Cancion 15', 3, 3, 240, 1, NOW(), 'admin', NOW(), 'admin', 1),
11    ('Cancion 16', 3, 3, 250, 1, NOW(), 'admin', NOW(), 'admin', 1),
12    ('Cancion 17', 4, 4, 260, 1, NOW(), 'admin', NOW(), 'admin', 1),
13    ('Cancion 18', 4, 4, 270, 1, NOW(), 'admin', NOW(), 'admin', 1),
14    ('Cancion 19', 5, 5, 280, 1, NOW(), 'admin', NOW(), 'admin', 1),
15    ('Cancion 20', 5, 5, 100, 1, NOW(), 'admin', NOW(), 'admin', 1);
16 ROLLBACK;
```

Registros

```
SELECT u.nombre, SUM(e.duracion) AS minutos
FROM usuarios u
JOIN escuchas e ON u.id_usuario = e.id_usuario
WHERE YEAR(e.timestamp) = 2025
GROUP BY u.id_usuario
ORDER BY minutos DESC
LIMIT 1;
```

```
SELECT u.nombre, COUNT(*) AS veces, GROUP_CONCAT(MONTH(s.fecha_fin)) AS
meses_cancelados
FROM usuarios u
JOIN suscripciones s ON u.id_usuario = s.id_usuario
WHERE YEAR(s.fecha_fin) = 2025 AND s.estado = 'cancelada'
GROUP BY u.id_usuario
```

```
ORDER BY veces DESC
```

```
LIMIT 1;
```

```
SELECT MONTH(fecha_inicio) AS mes, COUNT(*) AS cantidad
FROM suscripciones
WHERE YEAR(fecha_inicio) = 2025 AND tipo IN ('premium', 'familiar', 'estudiantil')
GROUP BY mes
ORDER BY mes;
```

Evidencia

```
Scripts > 06-Registros.sql
```

```
1  SELECT u.nombre, SUM(e.duracion) AS minutos
2  FROM usuarios u
3  JOIN escuchas e ON u.id_usuario = e.id_usuario
4  WHERE YEAR(e.timestamp) = 2025
5  GROUP BY u.id_usuario
6  ORDER BY minutos DESC
7  LIMIT 1;
8
9
10 SELECT u.nombre, COUNT(*) AS veces, GROUP_CONCAT(MONTH(s.fecha_fin)) AS
11     meses_cancelados
12 FROM usuarios u
13 JOIN suscripciones s ON u.id_usuario = s.id_usuario
14 WHERE YEAR(s.fecha_fin) = 2025 AND s.estado = 'cancelada'
15 GROUP BY u.id_usuario
16 ORDER BY veces DESC
17 LIMIT 1;
18
19 SELECT MONTH(fecha_inicio) AS mes, COUNT(*) AS cantidad
20 FROM suscripciones
21 WHERE YEAR(fecha_inicio) = 2025 AND tipo IN ('premium', 'familiar',
22     'estudiantil')
23 GROUP BY mes
24 ORDER BY mes;
```