

## DECISIÓN DE LA ESTRUCTURA DE LA BASE DE DATOS

La elección de crear cuatro tablas en la base de datos proporciona una estructura organizada y eficiente para gestionar los datos recolectados de las APIs, asegurando la integridad, escalabilidad y rendimiento del sistema.

Se optó por dividir la información en cuatro tablas distintas para mantener la claridad y la organización. Esto ayuda a evitar la redundancia y facilita el mantenimiento de los datos.

Para ello tuvimos que modificar los csv's iniciales que recogían los datos de las cuatro tablas y crear:

- Para el csv que suministra los datos a la tabla *artistas\_spotify*, creamos un identificador único, asociado al nombre de artista.
- Una herencia de este id único a los csv's de las otras tres tablas, para que estuviesen relacionadas, ya que también contenían el campo que contiene el nombre de artista. Lo que hacemos es que sustituimos este campo por este id.

### Estructura de las Tablas:

#### ▪ ***Tabla 1: artistas\_spotify***

Esta tabla almacena el nombre de los artistas y un id único para cada uno de ellos que servirá de unión con el resto de tablas de la base de datos.

Sus columnas son:

(id\_artista): que almacena un identificador. Será la PK

(artista): que almacena el nombre del artista

incluyen un identificador único (id) y el nombre del artista (nombre\_artista).

#### ▪ ***Tabla 2: canciones\_spotify***

Esta tabla contiene detalles sobre las canciones suministradas por Spotify.

Sus columnas incluyen:

(id\_cancion): identificador único por cada canción extraída de la API y los datos relacionados con ésta. Será la PK

(id\_artista\_sp:) identificador único, asociado al artista y que servirá de unión con la tabla *artistas\_spotify*, como FK

(tipo): que almacenará el dato 'track' – canción-. En futuras revisiones podrá almacenar también 'album', que nos permitirá tener acceso a esta información

(título\_canción): que contiene los títulos de las canciones

(nombre\_album): que contiene el nombre del álbum donde se incluye la canción

(año): contiene el año de lanzamiento de la canción

(género): contiene el género al que pertenece la canción

### ***Tabla 3: info\_artistas\_last\_fm***

Aquí se encuentra información adicional que suministra la API Last FM, sobre los artistas que ya habíamos recopilado en la API de Spotify y que están recogidos en la tabla artistas\_spotify.

Las columnas incluyen:

(id\_info\_lfm): contiene un identificador único para cada grupo de datos que alinea PK

(id\_artista\_lfm): contiene un identificador heredado de la tabla artistas\_spotify, y por el que se relaciona FK

(biografia): contiene una breve biografía del artista

(listeners): contiene la cantidad de oyentes que tiene este artista en el rango suministrado

(playcount): contiene la cantidad de reproducciones que ha tenido el artista en el rango suministrado

(artistas\_similares): contiene una lista de artistas que son similares al éste.

### ***Tabla 4: info\_artistas\_musicbrainz***

Esta tabla proporciona datos que suministra la API de MusicBrainz sobre los artistas que nosotros le hemos proporcionado basándonos en los datos extraídos en la API de Spotify.

Incluye las siguientes columnas:

(id\_info\_mb): contiene un identificador único para cada grupo de datos que alinea PK

(id\_artista\_mb): contiene un identificador heredado de la tabla artistas\_spotify, y por el que se relaciona FK

(tipo): contiene el dato 'Person' si es un artista en solitario o el dato 'Group' si corresponde a un grupo de música

(pais\_origen): contiene el país de origen del artista

(area\_origen): contiene el área de origen del artista

(inicio\_actividad): contendrá la fecha de inicio de actividad, en el caso de que el tipo de artista sea un 'Group' y este dato haya sido suministrado por la API. En cualquier otro caso devolverá un valor nulo.

(fin\_actividad): contendrá la fecha de fin de actividad, en el caso de que el tipo de artista sea un 'Group' y este dato haya sido suministrado por la API. En cualquier otro caso devolverá un valor nulo.

(fecha\_nacimiento): contendrá la fecha de nacimiento del artista, en el caso de que el tipo de artista sea 'Person' y este dato haya sido suministrado por la API, En cualquier otro caso devolverá un valor nulo.

(fecha\_fallecimiento): contendrá la fecha de fallecimiento del artista, en el caso de que el tipo de artista sea 'Person' y este dato haya sido suministrado por la API. En cualquier otro caso devolverá un valor nulo.

### **Beneficios:**

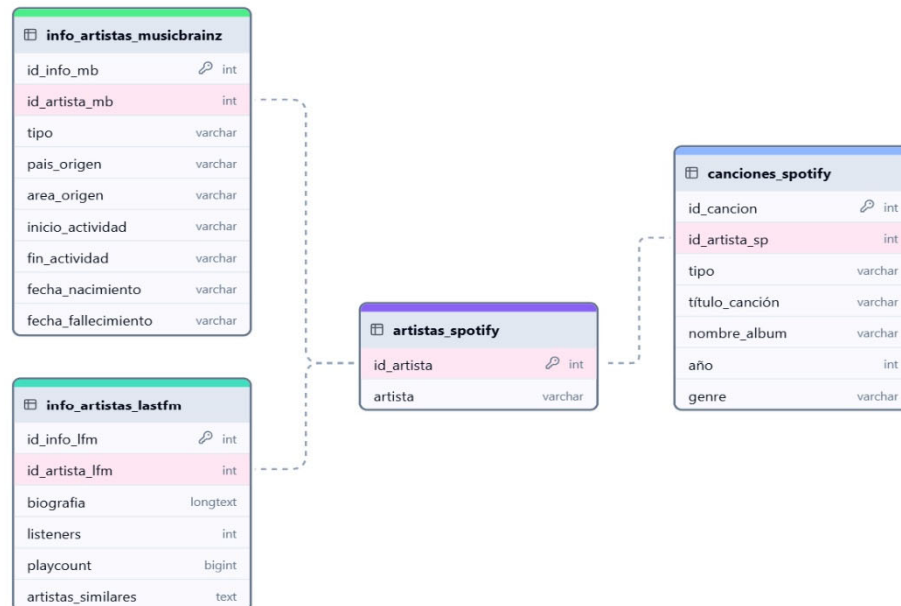
- **Escalabilidad:** Facilita la adición de nuevas características o tablas en el futuro sin complicar la estructura existente.
- **Consultas Eficientes:** Permite realizar consultas más rápidas y específicas, optimizando el rendimiento de la base de datos.
- **Facilidad de Mantenimiento:** Aislando diferentes tipos de datos, es más sencillo actualizar o modificar información sin afectar otras tablas.

### **Integración con APIs:**

- Cada tabla está diseñada para recibir datos específicos de las APIs, lo que permite una fácil integración y actualización de la información.

Representación gráfica que muestra la estructura de la una base de datos

## ER DIAGRAM



Como next step, se nos propuso la creación de tablas separadas de Music Brainz, según fuesen artistas o solistas. Y así lo hicimos.

Creamos una tabla, llamada `info_mb_artistas` que contiene los mismos datos que la original de Music Brainz, pero que solo tiene los datos de fecha de nacimiento y de fallecimiento, en vez de las cuatro fechas que almacenaba.

Y una más, llamada `info_mb_grupos` que contiene los mismos datos que la original de Music Brainz, pero que solo va a contener los datos de fecha de inicio de actividad y fecha fin de actividad, en vez de las cuatro fechas que almacenaba.

Hemos dejado para próximas actualizaciones la realización de las consultas con estas tablas, en vez de utilizar la original.