A blue and white logo

AI-generated content may be incorrect.

**PGDM- Big Data Analytics**

**Big Data Management & Analytics**

**Project Report**

**ERD – Music Streaming Platform Database**

**Submitted to: Prof. Amarnath Mitra**

**Submitted By: Group 10**

**Ashish Michael Chauhan – 055007**

**Itighne Sinha – 055015**

# **INTRODUCTION**

The purpose of this report is to document the database schema and entity relationship diagram (ERD) for a music streaming platform. This system is designed to efficiently manage various components, including users, playlists, songs, albums, artists, subscriptions, and payments, ensuring seamless data organization, retrieval, and management.

With the increasing demand for digital music platforms, a well-structured and normalized database is essential to maintain data integrity, minimize redundancy, and optimize query performance.

This report provides a detailed overview of the database schema, outlining the structure and attributes of each entity. Additionally, it explains the relationships between tables, ensuring that data dependencies are properly managed.

To uphold data consistency and eliminate anomalies, the database is designed to comply with the First Normal Form (1NF) and Second Normal Form (2NF). This ensures that all tables have a unique identifier (primary key), no repeating groups, and that partial dependencies are eliminated. The report also highlights primary and foreign key constraints, which define how entities are linked and maintain referential integrity.

# **DATA DESCRIPTION AND SCHEMA**

This database is designed to support a music streaming platform. It consists of multiple tables that store information about users, artists, albums, songs, playlists, listening history, likes, subscriptions, and payments. The relationships among these tables ensure an efficient and structured data flow within the platform.

1. The Users table stores details such as user\_id (primary key), username, email, password\_hash, date\_of\_birth, country, subscription\_type, and created\_at. Users can have different subscription plans that grant varying levels of access to content.
2. The Artists table captures information about musicians, including artist\_id (primary key), name, genre, country, bio, and created\_at. Artists produce albums and songs that are available for streaming.
3. The Albums table maintains details of musical collections, uniquely identified by album\_id, with attributes like title, release\_date, genre, and artist\_id (foreign key linking to Artists).
4. Each album contains multiple songs stored in the Songs table, which includes song\_id, title, artist\_id, album\_id, duration, genre, release\_date, popularity, and playlist\_id.
5. Users can create and manage Playlists, stored in the Playlists table with playlist\_id (primary key), user\_id (foreign key), name, and created\_at. The many-to-many relationship between playlists and songs is maintained through the playlist\_id attribute in the Songs table.
6. The Listening History table tracks user interactions with songs, recording history\_id (primary key), user\_id (foreign key), song\_id (foreign key), and played\_at.
7. The Likes table functions similarly, storing like\_id (primary key), user\_id (foreign key), song\_id (foreign key), and liked\_at to track user preferences.
8. The Subscriptions table records user plans, with subscription\_id (primary key), user\_id (foreign key), plan\_type, start\_date, end\_date, and payment\_status.
9. Payments made by users are recorded in the Payments table, which includes payment\_id (primary key), user\_id (foreign key), amount, payment\_date, payment\_method, and status.

This structured dataset ensures data consistency, referential integrity, and efficient query execution, supporting a well-optimized music streaming platform with a focus on personalization and seamless content organization.

**TABLE DESCRIPTIONS:**

1. **USERS**
   * **Primary Key:** user\_id (STRING, NOT NULL)
   * **Attributes:**
     + username (STRING, NOT NULL)
     + email (STRING, NOT NULL)
     + password\_hash (STRING, NOT NULL)
     + date\_of\_birth (DATE, NOT NULL)
     + country (STRING, NOT NULL)
     + subscription\_type (STRING, NOT NULL)
     + created\_at (DATETIME, NOT NULL)
2. **ARTISTS**
   * **Primary Key:** artist\_id (STRING, NOT NULL)
   * **Attributes:**
     + name (STRING, NOT NULL)
     + genre (STRING, NOT NULL)
     + country (STRING, NOT NULL)
     + bio (STRING, NULL)
     + created\_at (DATETIME, NOT NULL)
3. **ALBUMS**
   * **Primary Key:** album\_id (STRING, NOT NULL)
   * **Attributes:**
     + title (STRING, NOT NULL)
     + artist\_id (STRING, FOREIGN KEY)
     + release\_date (DATE, NOT NULL)
     + genre (STRING, NOT NULL)
4. **SONGS**
   * **Primary Key:** song\_id (STRING, NOT NULL)
   * **Attributes:**
     + title (STRING, NOT NULL)
     + artist\_id (STRING, FOREIGN KEY)
     + album\_id (STRING, FOREIGN KEY)
     + duration (DURATION, NOT NULL)
     + genre (STRING, NOT NULL)
     + release\_date (DATE, NOT NULL)
     + popularity (NUMBER, NOT NULL)
     + playlist\_id (STRING, FOREIGN KEY)
5. **PLAYLISTS**
   * **Primary Key:** playlist\_id (STRING, NOT NULL)
   * **Attributes:**
     + user\_id (STRING, FOREIGN KEY)
     + name (STRING, NOT NULL)
     + created\_at (DATETIME, NOT NULL)
6. **LISTENING HISTORY**
   * **Primary Key:** history\_id (STRING, NOT NULL)
   * **Attributes:**
     + user\_id (STRING, FOREIGN KEY)
     + song\_id (STRING, FOREIGN KEY)
     + played\_at (DATETIME, NOT NULL)
7. **LIKES**
   * **Primary Key:** like\_id (STRING, NOT NULL)
   * **Attributes:**
     + user\_id (STRING, FOREIGN KEY)
     + song\_id (STRING, FOREIGN KEY)
     + liked\_at (DATETIME, NOT NULL)
8. **SUBSCRIPTIONS**
   * **Primary Key:** subscription\_id (STRING, NOT NULL)
   * **Attributes:**
     + user\_id (STRING, FOREIGN KEY)
     + plan\_type (STRING, NOT NULL)
     + start\_date (DATE, NOT NULL)
     + end\_date (DATE, NOT NULL)
     + payment\_status (STRING, NOT NULL)
9. **PAYMENTS**
   * **Primary Key:** payment\_id (STRING, NOT NULL)
   * **Attributes:**
     + user\_id (STRING, FOREIGN KEY)
     + amount (DECIMAL, NOT NULL)
     + payment\_date (DATETIME, NOT NULL)
     + payment\_method (STRING, NOT NULL)
     + status (STRING, NOT NULL)

**RELATIONSHIP DESCRIPTIONS:**

1. **Users and Playlists:**
   * **Type:** One-to-Many
   * **Cardinality:**
     + A user can create multiple playlists (0..N)
     + A playlist belongs to one user (1)
2. **Users and Listening History:**
   * **Type:** One-to-Many
   * **Cardinality:**
     + A user can have multiple listening history records (0..N)
     + A history record belongs to one user (1)
3. **Users and Likes:**
   * **Type:** Many-to-Many
   * **Cardinality:**
     + A user can like multiple songs (0..N)
     + A song can be liked by multiple users (0..N)
4. **Users and Subscriptions:**
   * **Type:** One-to-One
   * **Cardinality:**
     + A user can have one active subscription at a time (0..1)
     + A subscription belongs to one user (1)
5. **Users and Payments:**
   * **Type:** One-to-Many
   * **Cardinality:**
     + A user can make multiple payments (0..N)
     + A payment belongs to one user (1)
6. **Artists and Albums:**
   * **Type:** One-to-Many
   * **Cardinality:**
     + An artist can have multiple albums (0..N)
     + An album belongs to one artist (1)
7. **Albums and Songs:**
   * **Type:** One-to-Many
   * **Cardinality:**
     + An album can contain multiple songs (0..N)
     + A song belongs to one album (1)
8. **Artists and Songs:**
   * **Type:** Many-to-Many
   * **Cardinality:**
     + An artist can produce multiple songs (0..N)
     + A song can be produced by multiple artists (M)
9. **Playlists and Songs:**
   * **Type:** Many-to-Many
   * **Cardinality:**
     + A playlist can contain multiple songs (0..N)
     + A song can be part of multiple playlists (0..N)
10. **Songs and Listening History:**

* **Type:** Many-to-Many
* **Cardinality:**
  + A song can appear in multiple listening history records (0..N)
  + A history record is associated with many song (1)

**PRIMARY AND FOREIGN KEY TABLE**

|  |  |  |
| --- | --- | --- |
| Table Name | Primary Key | Foreign Keys |
| Users | user\_id | None |
| Artists | artist\_id | None |
| Albums | album\_id | artist\_id |
| Songs | song\_id | artist\_id, album\_id, playlist\_id |
| Playlists | playlist\_id | user\_id |
| Listening History | history\_id | user\_id, song\_id |
| Likes | like\_id | user\_id, song\_id |
| Subscriptions | subscription\_id | user\_id |
| Payments | payment\_id | user\_id |

**ENTITY RELATIONSHIP DIAGRAM**

The following Entity Relationship Diagram represents the relationships and attributes of the system, ensuring normalization and eliminating redundancy

# 

# **CONCLUSION**

This database schema is designed to efficiently manage a music streaming platform by implementing a well-structured relational database with normalized tables and well-defined relationships. By ensuring data integrity, scalability, and optimal performance, the schema effectively organizes and maintains critical information related to users, playlists, songs, albums, artists, subscriptions, and payments while eliminating redundancy.

A key aspect of this design is the structured relationship between users and their interactions with the platform, such as playlist creation, listening history, and song preferences. The schema ensures compliance with the First and Second Normal Forms (1NF and 2NF), eliminating partial dependencies and maintaining referential integrity across all entities. The use of primary keys uniquely identifies each record, while foreign keys establish and maintain logical connections between tables, preventing anomalies and duplication.

Additionally, the schema is highly scalable, allowing the system to handle an increasing number of users, songs, and transactions without compromising efficiency. Its structured approach supports seamless content management, efficient data retrieval, and system reliability, making it a strong foundation for a modern music streaming platform.

Furthermore, this design is adaptable for future enhancements, including personalized recommendations, advanced analytics, and AI-driven music discovery, ensuring long-term viability and continuous improvement of the system.