Music Streaming Content Management System

CMPE 351 – Database Systems Project
Student: YAĞMUR DOĞAN – 121200126

---

1. Introduction

This project focuses on designing and implementing a Music Streaming Content Management System using fundamental database concepts and a fully functional web based user interface.

The system is inspired by real world platforms such as Spotify and Apple Music and enables full management of:

- Artists and their social media links
- Albums and tracks
- Track moods (multivalued attribute)
- Users
- Premium and Free subscription types (subclasses)
- Playlists and playlist-track relationships

The main objectives of this project are:

- Designing an accurate ER Diagram
- Implementing relational schema
- Applying normalization
- Performing full CRUD operations
- Integrating SQLite database with a Streamlit user interface

---

2. System Overview

1. Artist Management

- Add artist
- Update artist
- Delete artist
- View all artists

2. Artist Social Links (Multivalued Attribute)

- Add social media accounts per artist
- Update social links
- Delete social links

3. Album Management

- Add albums
- Update albums
- Delete albums
- Assign artists to albums

4. Track & Mood Management

- Add tracks
- Update tracks
- Delete tracks
- Assign moods to tracks (multivalued)

5. User Management

- Add base users
- Update users
- Delete users

6. Subscription Management

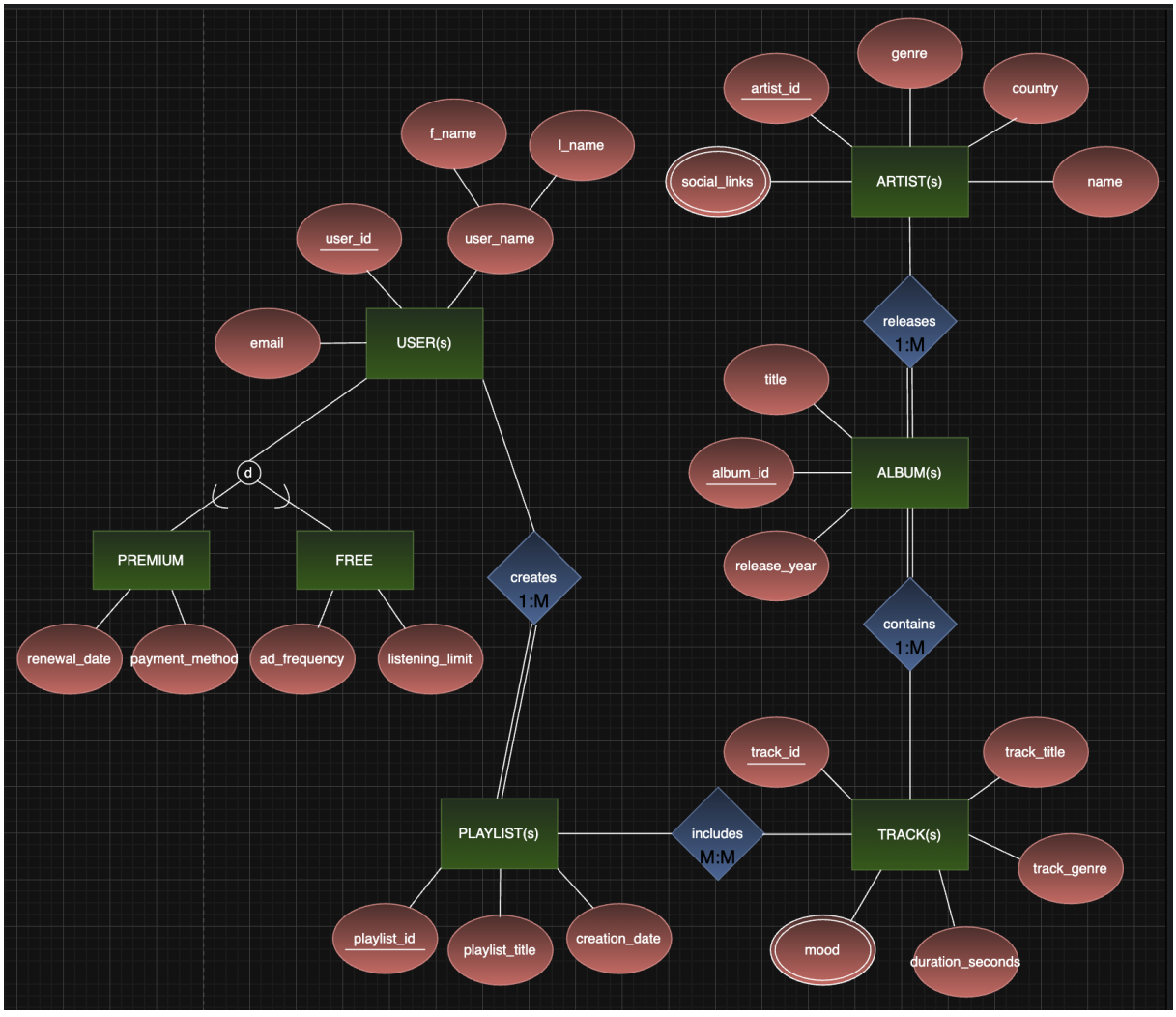
- Premium users with:
  - renewal_date
  - payment_method
- Free users with:
  - ad_frequency
  - listening_limit

7. Playlist Management

- Create playlists
- Assign users to playlists
- Add and remove tracks from playlists

---

3. ER Diagram Design



4. ER Schema

ARTIST=(artist_id, name, genre, country, {social_links})

ALBUM=(album_id, title, release_year)

TRACK=(track_id, track_title, duration_seconds, track_genre, {mood})

USER=(user_id, user_name(f_name, l_name), email)

PREMIUM=(user_id, renewal_date, payment_method)

FREE=(user_id, ad_frequency, listening_limit)

PLAYLIST=(playlist_id, playlist_title, creation_date)

releases=(artist_id (FK), album_id (FK))

contains=(album_id (FK), track_id (FK))

creates=(user_id (FK), playlist_id (FK))

includes=(play_id (FK), track_id (FK))

Relationships:

- ARTIST releases ALBUM
- ALBUM contains TRACK
- USER creates PLAYLIST
- PLAYLIST includes TRACK

5. Relational Schema

ARTIST=(artist_id, name, genre, country)

ArtistSocialLinks=(social_id, artist_id (FK), platform, social_link)

USER=(user_id, f_name, l_name, email)

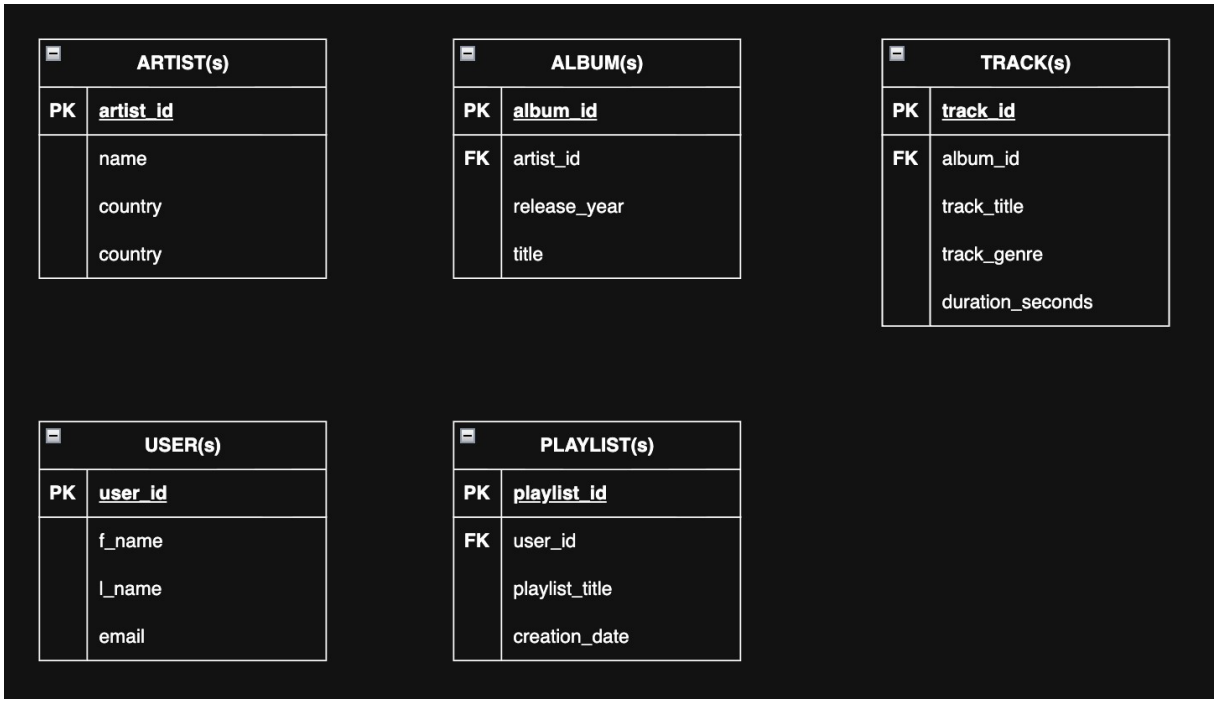
TRACK=(track_id, track_title, duration_seconds, track_genre)

TrackMoods=(mood_id, track_id (FK), mood)

PlaylistTrakcs=(playlist_id (FK/PK), track_id (FK/PK), position)

---

6. Tables



7. User Interface Implementation

The user interface is built using Streamlit, allowing real time interaction with the database.

Main Features:

- Sidebar navigation menu
- All tables support:
  - Insert
  - Update
  - Delete
  - View
- Dynamic foreign key selection
- Auto refresh after every operation using:

  st.rerun(). -> This ensures the interface always displays **live and consistent data**.

8. CRUD Operations

The system fully supports CRUD (Create, Read, Update, Delete) operations for all main entities in the database through the Streamlit user interface. Each operation ensures real time interaction with the SQLite database.

Artist Management

- Users can add new artists to the system.
- Existing artist information such as name, country, and genre can be updated.
- Artists can be removed from the database.
- All artists can be viewed in a tabular format.

Artist Social Links (Multivalued Attribute)

- Multiple social media accounts can be added for each artist.
- Existing social links can be updated.
- Social links can be deleted individually.

- All social links are displayed with their associated artists.

Album Management

- New albums can be created and linked to specific artists.
- Album titles, artist associations and release years can be updated.
- Albums can be deleted.
- The full album list can be displayed.

Track and Mood Management

- New tracks can be added to the system and linked to albums.
- Track information such as title, duration and genre can be updated.
- Tracks can be deleted.
- Each track can have multiple moods assigned as a multivalued attribute.
- Track moods can also be added and removed dynamically.

User Management

- New users can be created with first name, last name and email information.
- Existing user details can be updated.
- Users can be deleted from the system.
- All registered users can be viewed.

Premium User Management

- Users can be assigned a Premium subscription.
- Premium users have a renewal date and payment method.
- Premium subscription details can be updated.
- Premium users can be removed from the Premium category without deleting the base user.

Free User Management

- Users can be assigned a Free subscription.
- Free users have ad frequency and listening limits.
- Free subscription details can be updated.
- Free users can be removed from the Free category without deleting the base user.

Playlist Management

- Users can create new playlists.
- Playlists can be assigned to different users.
- Playlist information can be updated.
- Playlists can be deleted.

Playlist–Track Relationship Management

- Tracks can be added to playlists with a specific position order.
- Tracks can be removed from playlists.
- Playlist contents can be viewed using join queries.

All CRUD operations are executed through a dynamic web interface and are immediately reflected in the database without requiring a system restart.