Data Management and Database Design
Spring 2024
P2: Database Design, Conceptual Model

Group 13:

- Anirudha Joshi
- Samarjeet Ganesh Chavan
- Preeti Kulkarni
- Taranjot Dang
- Dandan Zhu

MUSIC ANALYTICS MANAGEMENT SYSTEM

BUSINESS PROBLEM

A music-related database management project like the one we are working on could address several business problems and opportunities within the music industry. Some potential business problems that this project could help solve include:

1. **Content Management:** Efficiently managing a vast catalog of music tracks, artists, albums, and playlists is a challenge for music streaming platforms and record labels. A well-organized database can streamline content management processes, making it easier to add, update, and remove music content.

2. **Personalization and Recommendation:** Music platforms often strive to deliver personalized recommendations to users based on their preferences, listening history, and behavior. A comprehensive database that captures user interactions, track features, and playlist data can power advanced recommendation algorithms, enhancing user satisfaction and engagement.

3. **Royalty Management:** Tracking and managing royalties for artists, songwriters, and rights holders is a complex task in the music industry. A database that records track streams, chart performances, and playlist inclusions can facilitate accurate royalty calculations and payments, ensuring fair compensation for all stakeholders.

4. **Analytics and Insights:** Music platforms and labels can leverage data analytics to gain insights into music trends, user behaviors, and market performance. By analyzing data from the database, businesses can make informed decisions regarding content curation, marketing strategies, and artist development.

5. **User Engagement and Retention:** Understanding user preferences and behaviors is crucial for increasing user engagement and retention on music platforms. By analyzing user interactions, playlist data, and listening habits stored in the database, businesses can implement strategies to enhance the user experience, increase retention rates, and reduce churn.

6. **Content Licensing and Rights Management:** Managing licensing agreements and rights for music content is essential for ensuring legal compliance and avoiding copyright infringement. A centralized database that tracks licensing agreements, rights ownership, and usage permissions can help businesses stay compliant and mitigate legal risks.

7. **Marketing and Promotion:** Data from the database can be used to identify emerging artists, trending tracks, and popular genres, guiding marketing, and promotional campaigns. By targeting specific user segments based on their preferences and behaviors, businesses can maximize the effectiveness of their marketing efforts and drive engagement.

Overall, a music-related database management project can address various business challenges and opportunities in the music industry, ranging from content management and royalty tracking to personalized recommendations and market insights.

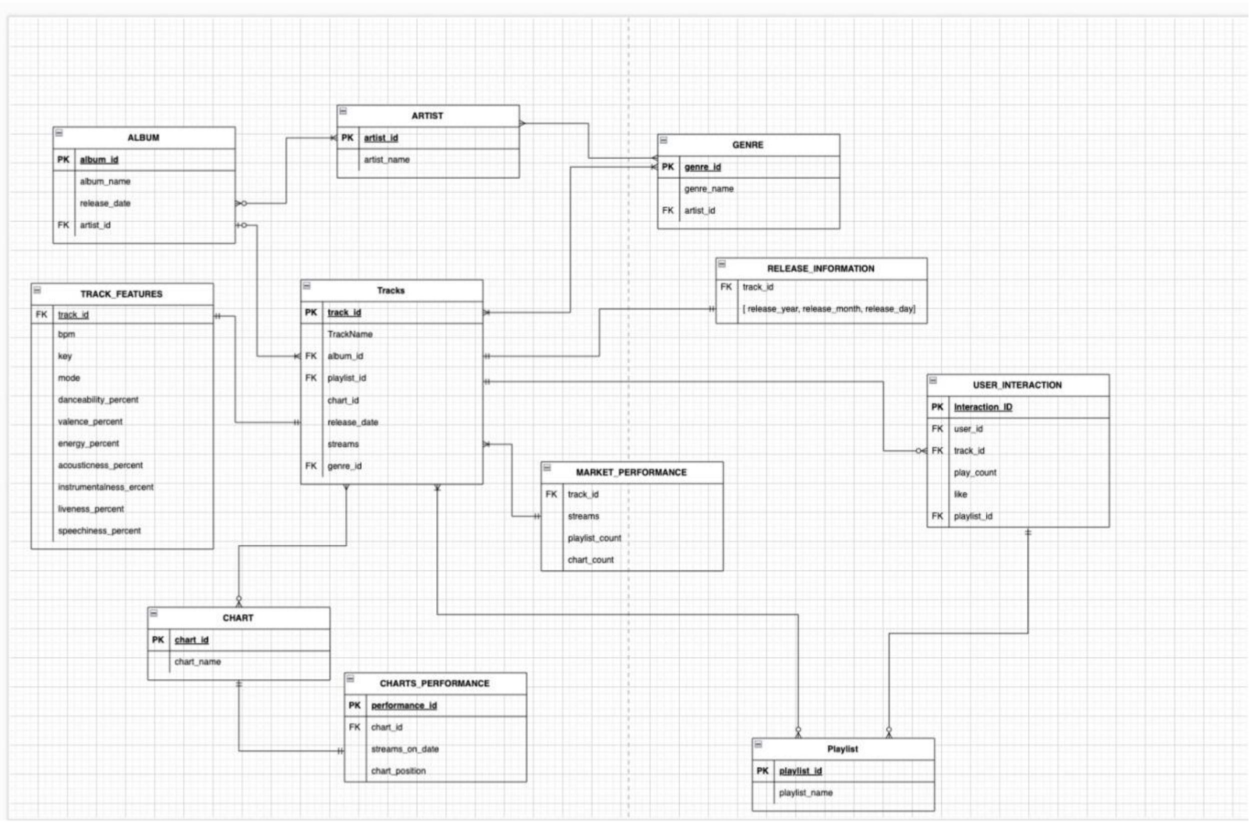
ENTITIES:

1. **Tracks:** Contains information about each track, such as track name, release date (year, month, day), BPM (beats per minute), key, mode, danceability, valence, energy, acousticness, instrumentalness, liveness, and speechiness percentages.
2. **Artists:** Stores artist-related information. Each artist has a unique identifier and name. Tracks may be linked to one or more artists.
3. **Albums:** While the dataset does not explicitly include album names, if available, this entity could store album-related information and link to the tracks and artists involved.
4. **Playlists:** Information about Spotify playlists that include the track, such as playlist count and possibly the playlist names or IDs if available.
5. **Charts**: Details tracks' presence on Spotify charts, including the number of times it appeared on charts and stream counts.
6. **Genres:** If genre data is available for each track or artist, this entity can store genre names. The relationship between tracks/artists and genres would likely be many-to-many, requiring additional junction tables.
7. **Charts_Performance:** This entity could detail the performance of tracks on Spotify charts over time, including date-specific streams, chart positions, and duration on charts if this data can be captured or inferred.
8. **User_Interaction:** This hypothetical entity could log user interactions with tracks, such as plays, likes, and adds to playlists, if such data were available, supporting detailed user behavior analytics.
9. **Release_Information:** Specifically focuses on the release details of tracks, separating out the release date components for more complex temporal analyses.
10. **Track_Features:** Focuses on the musical and audio features of tracks, such as BPM, key, mode, and various percentages (danceability, valence, energy, etc.), enabling analyses on music trends and preferences.
11. **Market_Performance**: An entity to capture the market performance of tracks, including streams, chart appearances, and presence in playlists, possibly segmented by regions if such data is available or can be integrated.

RELATIONSHIP:

- **Tracks to Artists** (Many-to-Many): A track can have multiple artists, and an artist can have multiple tracks.
- **Tracks to Albums** (Many-to-One): Each track belongs to one album, but an album can contain multiple tracks. This assumes album data is available and included in your schema.
- **Artists to Albums** (Many-to-Many): An artist can release multiple albums, and an album can feature multiple artists. If albums are tracked, a junction table might be necessary unless albums are uniquely linked to single artists.
- **Tracks to Playlists** (Many-to-Many): A track can be included in multiple playlists, and a playlist can contain multiple tracks. This relationship is managed through the
- **Tracks to Genres** (Many-to-Many): A track can belong to multiple genres, and a genre can include many tracks. This relationship would require a junction table if genre data is available and included.
- **Artists to Genres** (Many-to-Many): An artist can produce works in multiple genres, and a genre can encompass works by many artists. This relationship also would likely require a junction table for proper management.
- **Tracks to Charts_Performance** (One-to-Many): Each track can have multiple chart performance entries (reflecting its performance over time), but each chart performance entry is linked to one track.
- **Tracks to Market_Performance** (One-to-One): Each track has one market performance entry that summarizes its overall market performance, including streams and presence in playlists and charts.
- **Tracks to Track_Features** (One-to-One): Each track has a set of features (like BPM, key, mode, etc.) that are unique to it.
- **Tracks to Release_Information** (One-to-One): Each track has one release information entry detailing its release date.
- **User_Interaction to Tracks** (Many-to Optional One): Assuming user interaction data is available, each interaction (like play, like, add to playlist) is associated with one track, but a track can have many interactions.
- **User_Interaction to Playlist** (One-to Optional Many): Assuming user interaction data is available, each interaction (like play, like, add to playlist) is associated with optionally many playlist, but a playlist can have one interaction.

ERD:


ARTIST
PK artist_id
artist_name
ALBUM
PK album_id
album_name
release_date
FK artist_id
GENRE
PK genre_id
genre_name
FK artist_id
RELEASE_INFORMATION
FK track_id
[release_year, release_month, release_day]
TRACK_FEATURES
FK track_id
bpm
key
mode
danceability_percent
valence_percent
energy_percent
acousticness_percent
instrumentalness_ercent
liveness_percent
speechiness_percent
Tracks
PK track_id
TrackName
FK album_id
FK playlist_id
chart_id
release_date
streams
FK genre_id
USER_INTERACTION
PK Interaction_ID
FK user_id
FK track_id
play_count
like
FK playlist_id
MARKET_PERFORMANCE
FK track_id
streams
playlist_count
chart_count
CHART
PK chart_id
chart_name
CHARTS_PERFORMANCE
PK performance_id
FK chart_id
streams_on_date
chart_position
Playlist
PK playlist_id
playlist_name