M22CS4.301: Data and Applications
Phase 1

Radhikesh Agrawal - 2021113013
Vansh Garg - 2021111006
Kyrylo Shyvam Kumar - 2021101080

IIIT Hyderabad — October 22, 2022

Introduction to the mini-world

Our mini-world is a music platform, μsic, that allows users to navigate through the world of notes and lyrics. The platform has the metadata of all songs, as well as current trends. We provide nearly Spotify like experience, with much more flexibilty in analysis. But unfortunately we can't play songs. :(

Purpose of the Database

We present users a way to add songs to playlist, track user's musical engagement and provide detailed analysis of what is popular in recent times. The platform is there to satisfy the needs of demanding users, that want to thave access to a music search engine on their fingetips and see music analytics, current trends and distribution of music tastes across the globe.

Users of the Database

Any person having interest in music, will find μsic to be amusing as it combines the Spotify experience, with detailed and exhaustive analysis touching different spheres of music world.

Application of the Database

This enables users of database to seemlessly explore different genres of songs, discover artists who have fascinating works but not necessary enjoy lots of publicity. Users can also get reports of listening habits of other people.

1 Entities

1.1 User

A user is any person who uses our platform μsic

1.1.1 Attributes

- User ID :: Int → 6 digit Int, Primary Key
- Username :: String → Not Null
- Name :: String → Not Null
- Age :: Int → Not Null
- Joining Date :: Not Null → String of type YYYY-MM-DD (Year, Month, Date)

1.2 Record Label Company

A record label company is responsible for distributing and for the branding of the song. Every company in our mini-world will has a unique name.

1.2.1 Attributes

- Company Name :: String → Not Null, Primary Key
- Songs labeled :: Int → Derived Attribute

1.3 Song

A song is any song which is in our platform μsic. μsic assigns each song with a unique song id. A song must be part of exactly one album.

1.3.1 Attributes

- Song Name :: String → Not Null
- Song ID :: Int → 8 digit Int, Primary Key
- Genre :: String → Not Null, Multivalued
- Duration :: Not Null → String of Type HH-MM-SS (Hours, Minutes, Seconds)
- Instruments used :: String → Multivalued
- Played :: Int → Number of times song is played, Derived Attribute

1.4 Album

An album is a collection of songs. It is a weak entity type and it is uniquely identified by it's name and the creator it was released by

1.4.1 Attributes

- Album Name :: String → Not Null, Partial Key
- Number of Songs :: Int → Not Null, Derived Attribute

1.5 Playlist

A playlist is a collection of songs which users can create and add songs to. It is a weak entity type and is uniquely identified by it's name and the user who created the playlist

1.5.1 Attributes

- Playlist Name :: String → Partial Key
- Total Duration :: Not Null → String of type HH-MM-SS (Hours, Minutes, Seconds), Derived Attribute
- Number of Songs :: Int → Not Null, Derived Attribute
- Status :: String from set $\{Public, Private\}$

1.6 Creator

A creator is any individual artist or group of artists who do anything related to music professionally. It is divided into two subclasses - Artist and Band. It is a strong entity type and all creators will have an account on our platform μsic.

1.6.1 Attributes

- Name :: String → Not Null
- User ID :: Int → 6 digit Int, Primary Key
- Creator ID :: Int → 6 digit Int, Not Null, Primary Key
- Creator Type :: Not Null → String from set $\{Individual, Group\}$
- Awards won :: String → Multivalued

Artist

An artist is an individual who works on music professionally.

1.6.2 Attributes

- Occupation :: String → Not Null, Multivalued
- Instruments played :: String → Multivalued

Band

A band is a collection of artists who release songs together. A band is uniquely identified by it's name.

1.6.3 Attributes

- Genre :: String → Not Null
- Creation Date :: Not Null → String of type YYYY-MM-DD (Year-Month-Date)
- Songs Released :: Int → Not Null, Derived Attribute

2 Relationship Types

2.1 Creator - Song

A creator works on a song or on a particular part of it. The relation is between the two strong entities - Creator and Song. This is a binary relation.

2.1.1 Attributes

- Participation :: String → Not Null, Multivalued (e.g. Composer, Vocalist, etc)

2.1.2 Participation Constraints

A creator can work on any number of songs so the (min, max) ratio for creator is $(1, N)$. A song must have atleast one creator who worked on it so the (min, max) ratio for song is $(1, N)$.

2.2 Artist - Band

An artist can belong to a band. This is a relation between the two subclasses of the strong entity type Creator.

2.2.1 Attributes

- Joining Year :: Not Null → String of type YYYY

2.2.2 Participation Constraints

An artist can at max be part of one band so the (min, max) ratio of an artist is $(0, 1)$. A band must have atleast one member so the (min, max) ratio of a band is $(1, N)$.

2.3 User listens to a Song

The relation is between the two strong entities - User and Song. This is a binary relation.

2.3.1 Attributes

- Timestamp :: Not Null → String of type YYYY-MM-DD-HH-MM-SS (Year, Month, Date, Hours, Minutes, Seconds)
- Duration :: Int → Not Null

2.3.2 Participation Constraints

A user can listen to arbritary number of songs so the (min, max) ratio for a user is $(0, N)$. A song can be played by any number of users so the (min, max) ratio is $(0, N)$. A user can also listen to the same song multiple times so a user can be in multiple relations with the same song each having a different value for timestamp.

2.4 User creates a Playlist

A user can create a playlist with arbritary number of songs. The relation is between the strong entity User and weak enttity Playlist. This is the identifying relation for the weak entity Playlist. This is binary relation.

2.4.1 Participation Constraints

A user can create arbritary number of playlists so the (mix, max) ratio for a user is $(0, N)$. A playlist will have exactly one user who created it so the (min, max) ratio will be $(1, 1)$.

2.5 User has edit access to a Playlist

Apart from the creator of the playlist. Other users can have edit access to a playlist so they can add and delete songs to a playlist. This is a binary relation.

2.5.1 Participation Constraints

A user can have access to any number of playlists so the (min, max) ratio of a user is $(0, N)$. A playlist can be accessed by arbritary number of users apart from the user who created the playlist so the (min, max) ratio of the playlist is $(0, N)$.

2.6 Song - Playlist - User - User

A Song is added to a playlist of a user by another or the same user. This is a relation between the entity types - Song, Playlist, User (creator), User (modifier). This is a ternary relation.

2.6.1 Participation Constraints

The (min, max) ratio for the entities are :

- Song → $(0, N)$.
- Playlist → $(1, N)$.
- User (Creator) → $(1, N)$
- User (Modifier) → $(0, N)$

2.7 Creator - Song - Album - Label Company

A creator releases a song as part of an album which is labeled by a company. This is a degree 4 relation.

2.7.1 Attributes

- Release Date :: Not Null → YYYY-MM-DD (Year, Month, Date)

2.7.2 Participation Constraints

The (min, max) ratio for the entities are :

- Creator → $(0, N)$
- Song → $(1, 1)$
- Album → $(1, N)$
- Label Company → $(0, N)$

3 Functional Requirements

The database is meant to enable easier querying regarding a song, performer/artist or record labelling companies. The database will act as a common platform where new information regarding a person/band/song can be accessed and updated. The functional requirements that the database will serve will be both retrieval and modification.

3.1 Modification

3.1.1 Insertion

Any new user who registers on μsic platform will be added databse with all the known and available information.

3.1.2 Release of album by a band

Band will release new albums, containing several titles. Other people, can include these albums/songs into their playlists as well as play them.

3.1.3 Change in structure of a band

Over the time, bands can take in new members or members can leave bands. Changes will be reflected in database.

3.1.4 Deleting songs in user history

Over the time, bands can take in new members or members can leave bands. Changes will be reflected in database.

3.2 Retrieval

3.2.1 Selection

- To select all songs,with details by a band (eg. All songs by band 'Queen').
- All the albums (with details) produced by labelling house.
- To select all songs,with details by a band (eg. All songs by band 'Queen').

3.2.2 Projection

- Select all band names which user did listen this month.
- Select all record label companiy names which produced albums of band.
- Users that have > 1 year presence on platform.
- Band names which have won at least 1 award.
- All song names which have > 100k plays.

3.2.3 Aggregate

- Total number of 'plays' an album has.
- The maximum number of awards won by band for any song.
- The average number of songs produced per band for a given record labelling company.

3.2.4 Search

- Search with partial text for name of record labelling company.
- Search for bands created last year (have matching year).
- Search of users whose name starts with 'X'.

3.3 Analysis

- Number award winning artists/bands which are not much popular (< 100k plays on the platform).
- Number of bands having atleast 2 guitarists.
- Number of very active users (more than 1000 songs played).
- Number of users which listen particular genre of music regularly (> x views during last month).