IST 659 Database Administration Concepts and Database Management

# FANTASY MIXTAPE LEAGUE

## A DATABASE EXPERIMENT

Kendra Danforth Osburn

December 2018

# Introduction

## Business Case

Thanks to streaming services like Spotify and Apple Music, finding and discovering new music is easier than ever. Meanwhile, self-publishing platforms like Soundcloud have given lesser-known artists a massive distribution network for their songs. Suddenly, music previously only heard in coffee shops or open-mic nights could find its way into any car, home, or crowded subway train via a simple internet connection. As a result, today's listeners have instant access to tens or even hundreds of millions of songs.

## Marketing Pitch

This revolution of access has resulted in a new dilemma: With such a glut of ear-candy, how can listeners hope to discover "good" music? For that matter, given the decline of radio stations and music labels as gatekeepers, who decides which music is "good"? Fantasy Mixtape League attempts to answer this question.

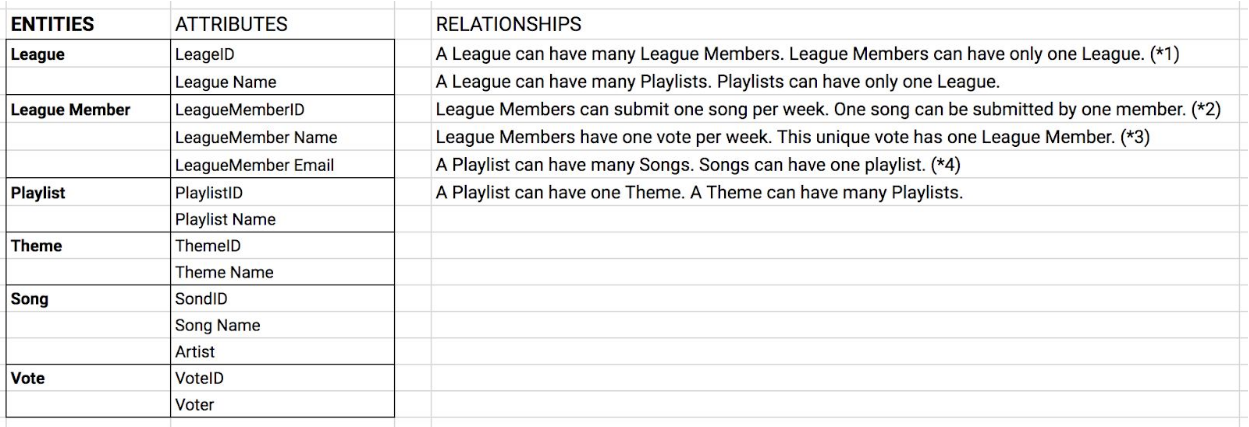
## Business Narrative

Fantasy Mixtape League is a group of friends (the League) who collaborate to build weekly playlists. Each week, the league leader (Commissioner) announces the playlist theme and gives members seven days to submit a song that fits that theme. After all songs have been submitted, they're compiled into one master playlist, which is then distributed to League members. Members have one week to listen to and vote for their favorite song for that playlist. At the end of the week, votes are counted, songs ranked, and a "best song" is crowned. This repeats every week, with a new theme each week.

# The Data Models

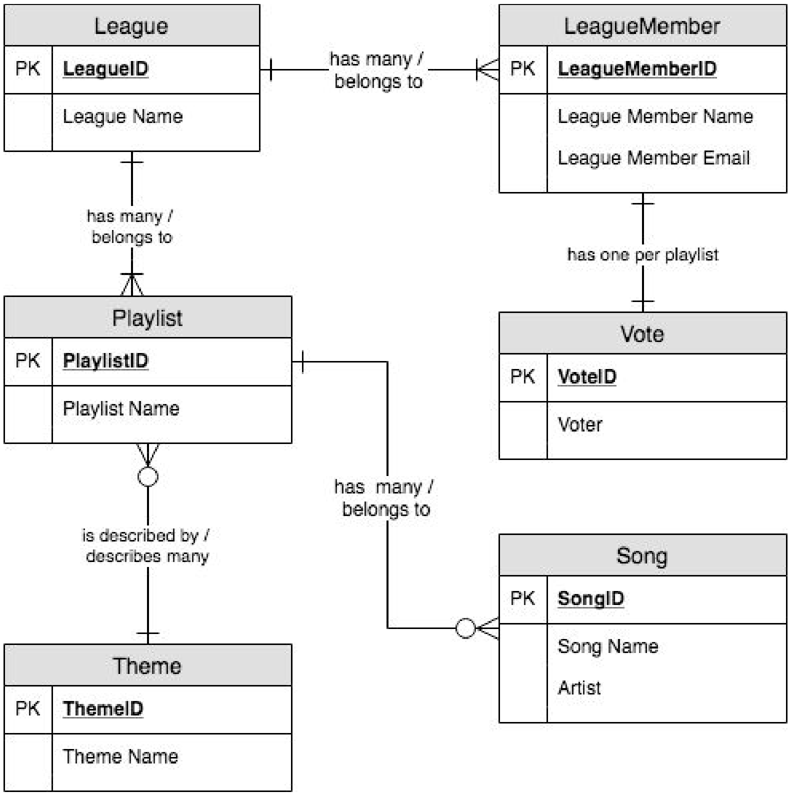
## Conceptual Model

Building off of our business narrative, we can identify our entities and relationships as:



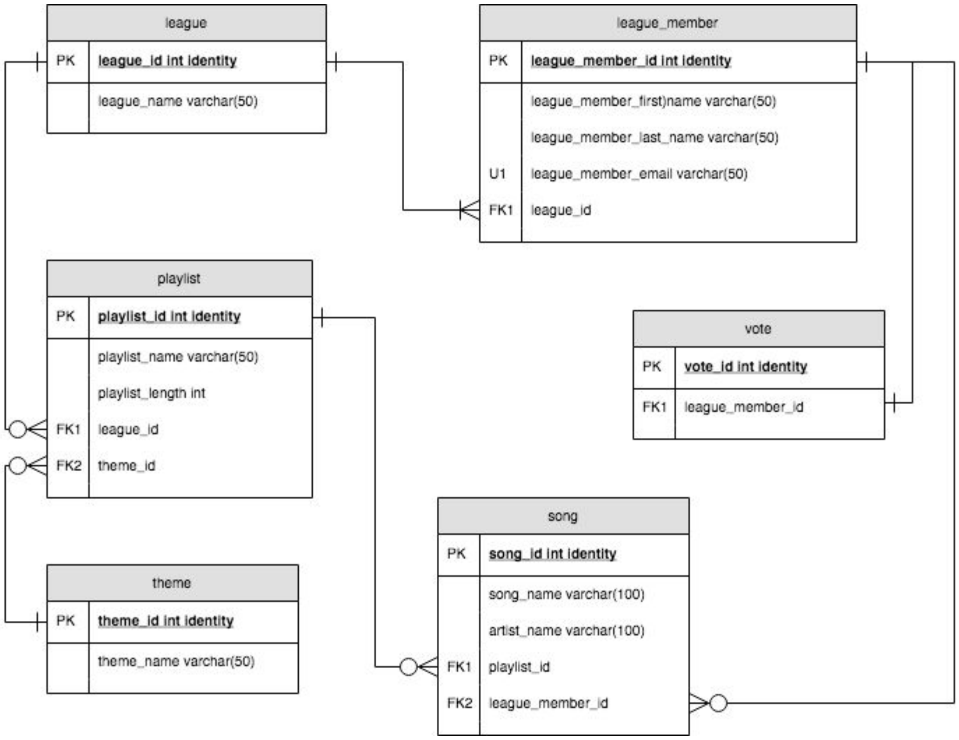
| ENTITIES | ATTRIBUTES | RELATIONSHIPS |
|---|---|---|
| **League** | LeageID | A League can have many League Members. League Members can have only one League. (*1) |
| | League Name | A League can have many Playlists. Playlists can have only one League. |
| **League Member** | LeagueMemberID | League Members can submit one song per week. One song can be submitted by one member. (*2) |
| | LeagueMember Name | League Members have one vote per week. This unique vote has one League Member. (*3) |
| | LeagueMember Email | A Playlist can have many Songs. Songs can have one playlist. (*4) |
| **Playlist** | PlaylistID | A Playlist can have one Theme. A Theme can have many Playlists. |
| | Playlist Name | |
| **Theme** | ThemeID | |
| | Theme Name | |
| **Song** | SondID | |
| | Song Name | |
| | Artist | |
| **Vote** | VoteID | |
| | Voter | |

## Conceptual Model Diagram



# Logical Model

## Logical Model Diagram



## Stakeholders

The concerned parties in this experiment are the “groupies,” or participants in Fantasy Mixtape League. The commissioner will want to know stats about his/her league. The moguls behind Fantasy Mixtape League will want to understand how best to serve their community of leagues.

## Business Questions

1. Who are the most popular music tastemakers (which user submits the most popular songs)?
2. Which songs, artists, playlists, and tastemakers are the most “controversial” (i.e., receive the most votes)?
3. Are there any correlations between the characteristics of songs (e.g. beats per minute, length, use of vocals) and the amount of votes they receive?
4. Are there any correlations between the characteristics of playlists (e.g. degree of musical variation as measured by BPM, artists featured) and the number of votes they receive?
5. Which tags are the most popular (most used)?

## Concerns & Considerations

Considered adding Album entity (an Album has many songs. A song can have/often doesn’t have many albums. Examples include “Best Of” albums.) However, decided against it as it wasn’t much use to the scope of this particular project as the playlists are essentially becoming user-created albums.

Considered adding a more complex “Artist” entity before realizing that even Spotify doesn’t differentiate between single musicians and bands with multiple musicians, preferring to lump them all into an artist category and only alluding to the “band members” within a (guessing here) varchar(200+) “artist description” field.

Considered separating “follower” into its own category, then realized that each user is actually a follower, by definition, of their league and their league playlists. So follower becomes redundant. This made me consider “listener/voter” and “user/curator” differences.

(*1) For now -- in the future they can potentially have multiple leagues, one for work, family etc.
(*2) Theoretically, a song can be submitted by multiple members. This has happened. However, for this model, we will constrain this.
(*3) Not sure the best way to handle this. In the real world, I’d consult with my superiors.
(*4) Not sure the best way to handle this. In the real world, I’d consult with my superiors. Use-case that is causing consternation: If there are many leagues and the theme is “Best Halloween Song” Monster Mash is going to be on multiple playlists across multiple leagues. Additionally, it could be on multiple playlists in that league -- “Best Halloween Song” and “Most Overplayed Song.” But it will not be on a specific playlist more than once. I have a feeling this is like a student having many classrooms. I wouldn't ever make this connection, because a student has many classes and those classes each have one classroom, but there is a good possibility that

(especially in smaller schools) this classroom could house multiple classes for this student. But again, I wouldn’t ever connect student and classroom.