Team 2:
Brian Douville, Nicolas Noonan, Jacob Swisher
3430 Database
Mohebbi Mohammad
6 May, 2020

Mohebbi Music Player Database Specification

Summary:

Music is the language that everyone understands and can talk along with, but there are not any programs that can share those interests. Registered users on this site will allow such users to not only share their music/podcast interests, but allow them to discuss amongst themselves which genre interests them. From this simple premise, we are creating our own music database. A user logs in and has access to all of our group's favorite music, artists, and podcasts.

Specifications:

- Artist: Artists are described by their name, first and last, their stage alias, the contract they have with a studio, their date of birth and death, time, date, and year, their biography and their primary genre or style of content. They are organized by their style and flow of their songs and who they collaborate with.
- Producer: Producers are described by their name, first and last, and their date of birth, time, date, and year. They are organized by which artists they promote as well as which studio they produce in.
- Studio: Studios are described by their name, location, address, which producers use their studio, and which artists have a contract with that studio. They are organized by which producer has worked in said studio.
- Podcast: Podcasts are described by their name, their genre and focus (science, religion, political, etc.), the participants of the podcasts, and who their target audience is. They are organized by which artist or artists partake in said podcasts.
- Song: Songs are described by which artist made the song, which album is the song in, which artist made the song, and the link for the song. They are organized by genre.
- Album: Albums are described by how many songs are in it, which artist created the album, the name of the album, and their target audience.
- Consumer: Consumers are described by their name, first and last, their email, their password, and the audience they associate themselves with.

Mohebbi Music Database Model Summarized:

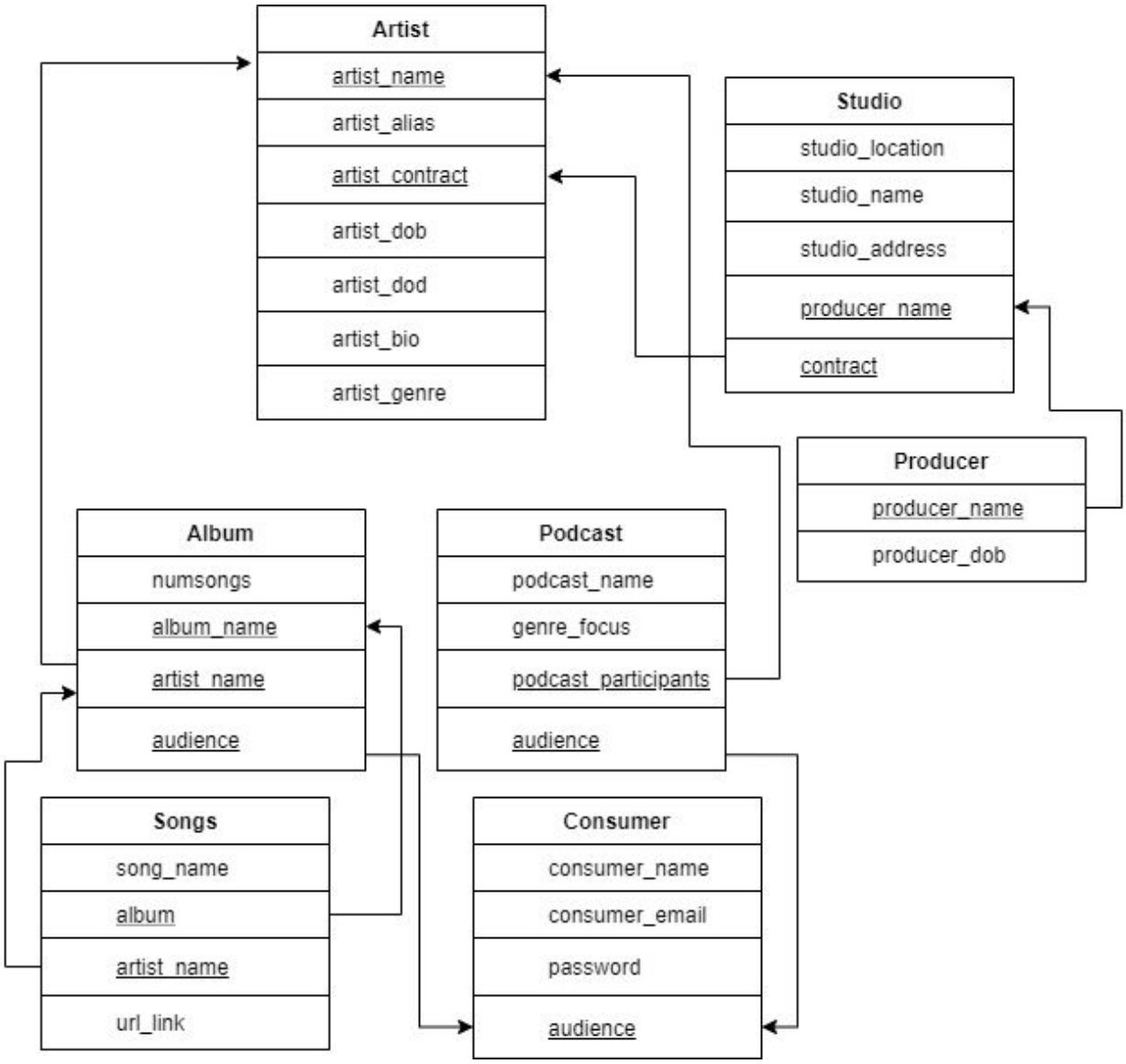
1. Artist
    a. <u>Artist Name</u>
        i. Fname
        ii. Lname
    b. Alias
    c. <u>Artist Contract</u>

    d. Artist DOB
        i. Time
        ii. Date
        iii. Year
    e. Artist DOD
        i. Time
        ii. Date
        iii. Year
    f. Artist bio
    g. Artist genre
2. Producer
    a. Producer Name
        i. Fname
        ii. Lname
    b. Producer DOB
        i. Time
        ii. Date
        iii. Year
3. Studio
    a. Location
    b. Name
    c. Address
    d. Producer Name
    e. Contract
4. Podcast
    a. Podcast Name
    b. Genre/Focus
5. Song
    a. Song Name
    b. Album Name
    c. Artist Name
    d. URL
6. Album
    a. NumSongs
    b. Album Name
    c. Artist Name
7. Consumer
    a. Name
        i. Fname
        ii. Lname
    b. Email
    c. Password

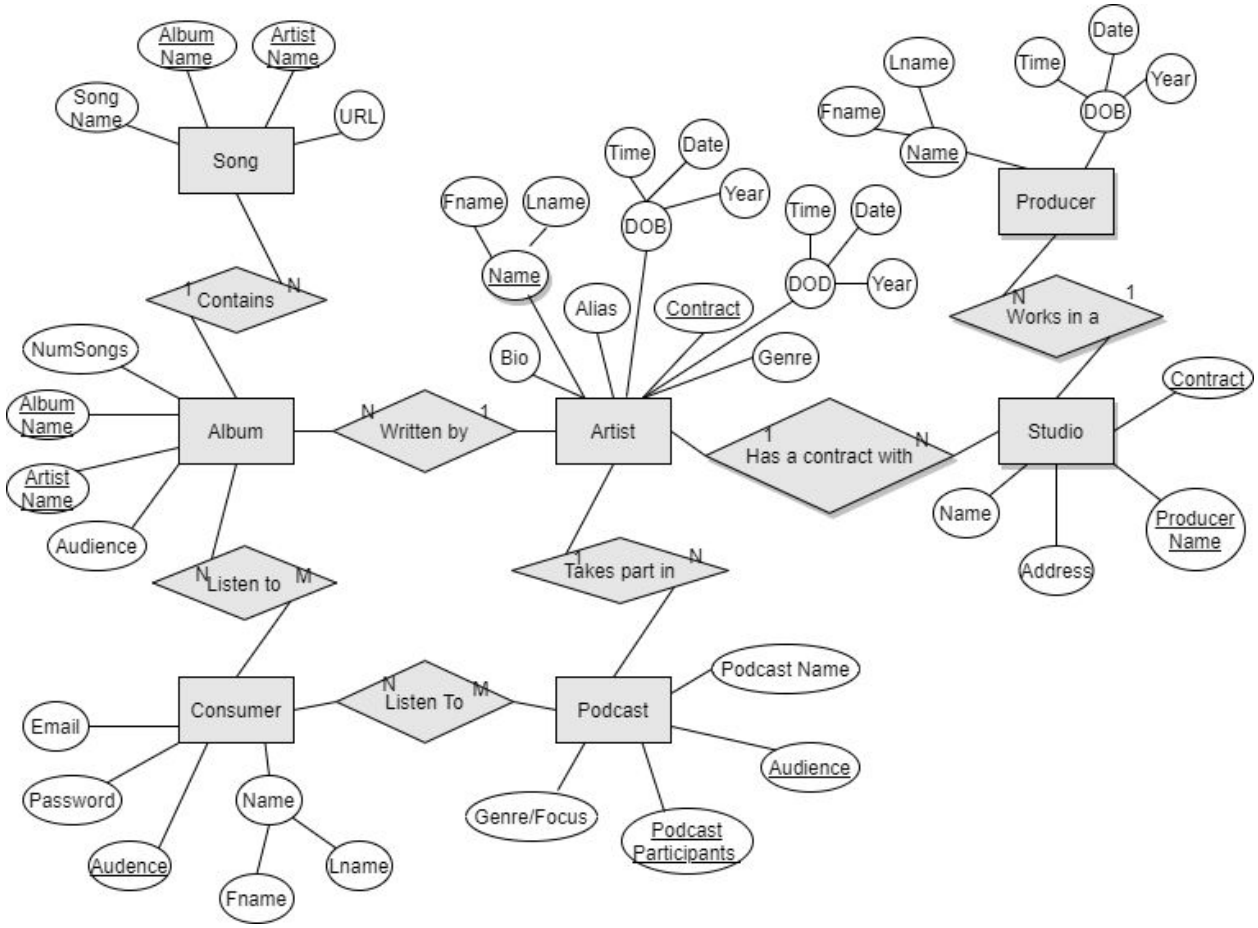
Relationships:

1. A Producer runs a Studio, N:1
2. A Studio has a contract with an Artist, N:1
3. An Artist can create a Podcast, N:M
4. An Artist can create an Album, N:1
5. An Album can have several Songs within it, 1:N
6. A Consumer listens to a Song, N:M
7. A Consumer listens to a podcast, N:M

ER Diagram:



RM Diagram:



Reflection on ER Diagram:

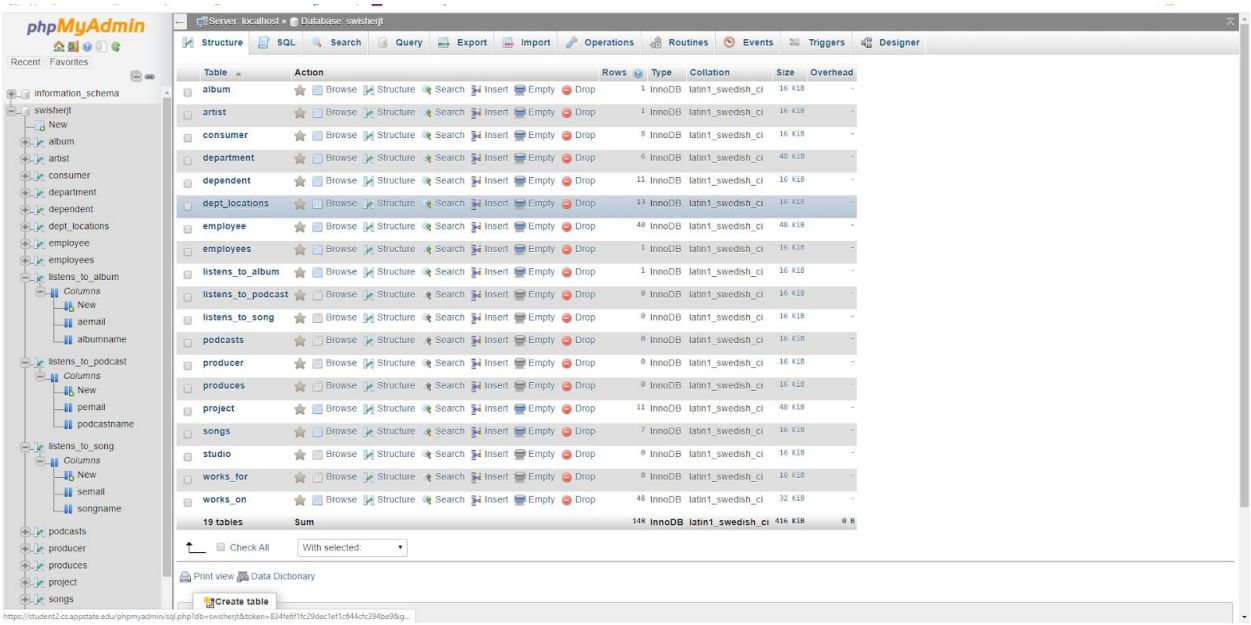
It was hard on deciding which factors to consider when creating a database not only centralized on who makes the music to who listens to it, but also how to combine each individual table together. The first model we transcribed didn't have a combination where it linked the producer and studio correctly. With this final ER diagram, we can now connect the producer with the studio, which connects to everything else.

Reflection on RM Diagram:

It came to the realization that the relationships between our entities were incorrect in our first RM diagram. For the Artist entity, we originally had the idea of having the name of the artist, their alias, and their date of birth, but in order to include as much information as we could we created more attributes that would allow for more preciseness in our artist. Also, The relationships between Artist with Podcasts and Albums needed to be changed as well, since we decided that a song would just be part of an album.

Database:

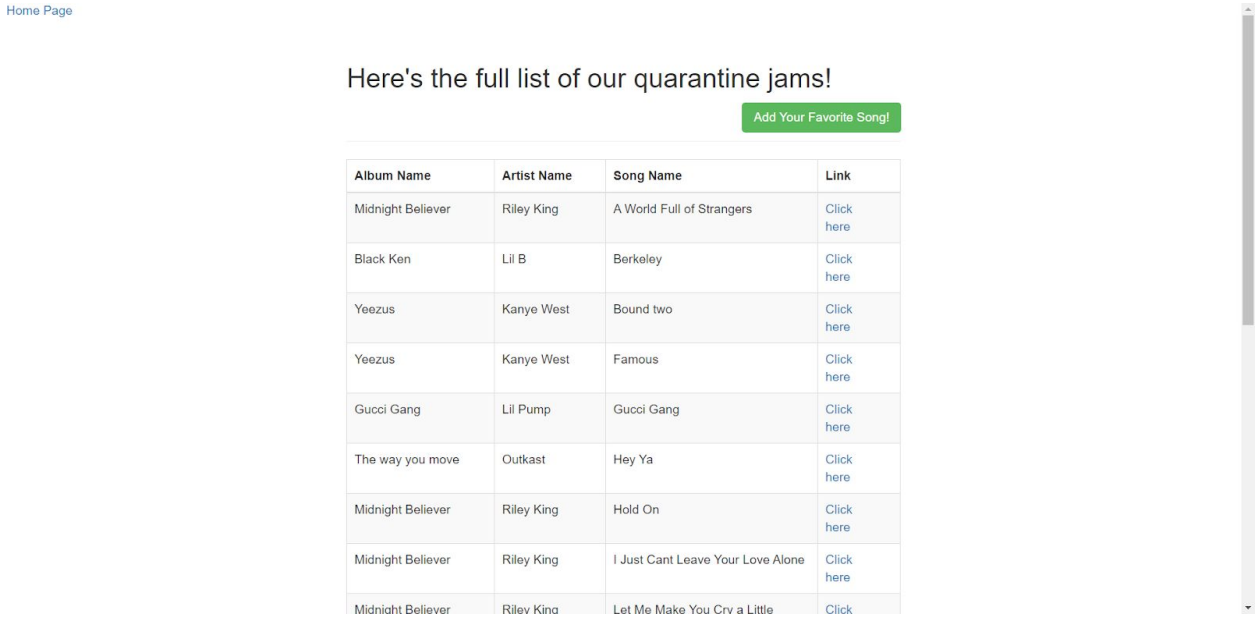
Given that all of the course labs used phpMyAdmin we decided it would be the best choice to use the platform for our database. We simply used one of the group members phpMyAdmin account for the entirety of the database. Also, we used the same group member's student2 account for php files loaded to the browser. Ultimately, the database allows for users to search and view a catalog of music that each member of our group has contributed too. Our main purpose of this database implementation is to collectively store and represent all of our favorite music, and podcasts that we have listened to during the lockdown. By sharing music together in this medium we could remain social, but still stay distant.



Web GUI:

Jacob, with considerable help from Nicholas, was able to create a simple web GUI that features a working CRUD. A CRUD is a system that interacts with the database to simply edit, insert, and remove columns. We spent many hours learning php, css, and html so that we could successfully implement the website.

https://student2.cs.appstate.edu/swisherjt/



Final Reflection:

Our team consisted of 4 people in the beginning, however due to the circumstances that happened in the middle of the semester, our 4th partner dropped and left our group, and with that the workload was left with 3 people. We collaboratively used what we learned from CS 3430 and utilized those teachings to create a hypothetical database and website where people can share and discuss with each other about music choices.

Brian Douville: Team Director

Nicholas Noonan: Database Dungeon Director

Jacob Swisher: Web Master