Team: Iteration Station

Project: SongDB

Team names: Nico Espina, Fernando Cuevas, Ricardo Carrillo

# Motivation:

We want to revolutionize the way people choose to listen to music by helping people make informed decisions on what songs they should listen to. Our database was created for music aficionados and common listeners alike, and is intended to be easy to use and easily expanded. In the future we plan to add user created critiques and we will like to potentially make into an app.

# Application:

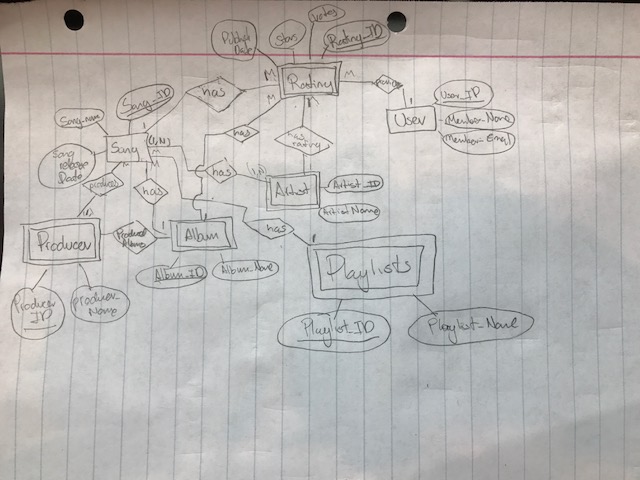
The SongDB’s purpose to allow for users and artists to rate their favorite songs. The application works very similar to imbd or Netflix databases. The application area of the database is to store information related to music releases. For each song, album and artists there will be a ratings system out of 5 stars. The database tracks ratings, songs, artists, discography, albums, release dates, playlists, etc. The SongDB is intended to be used form any computer of mobile device.

# Methodology:

### **Database management system**

mySQL was employed for its ease of use and compatibility with many database tools. mySQL workbench was used to create the Song DB database and ER Diagram.

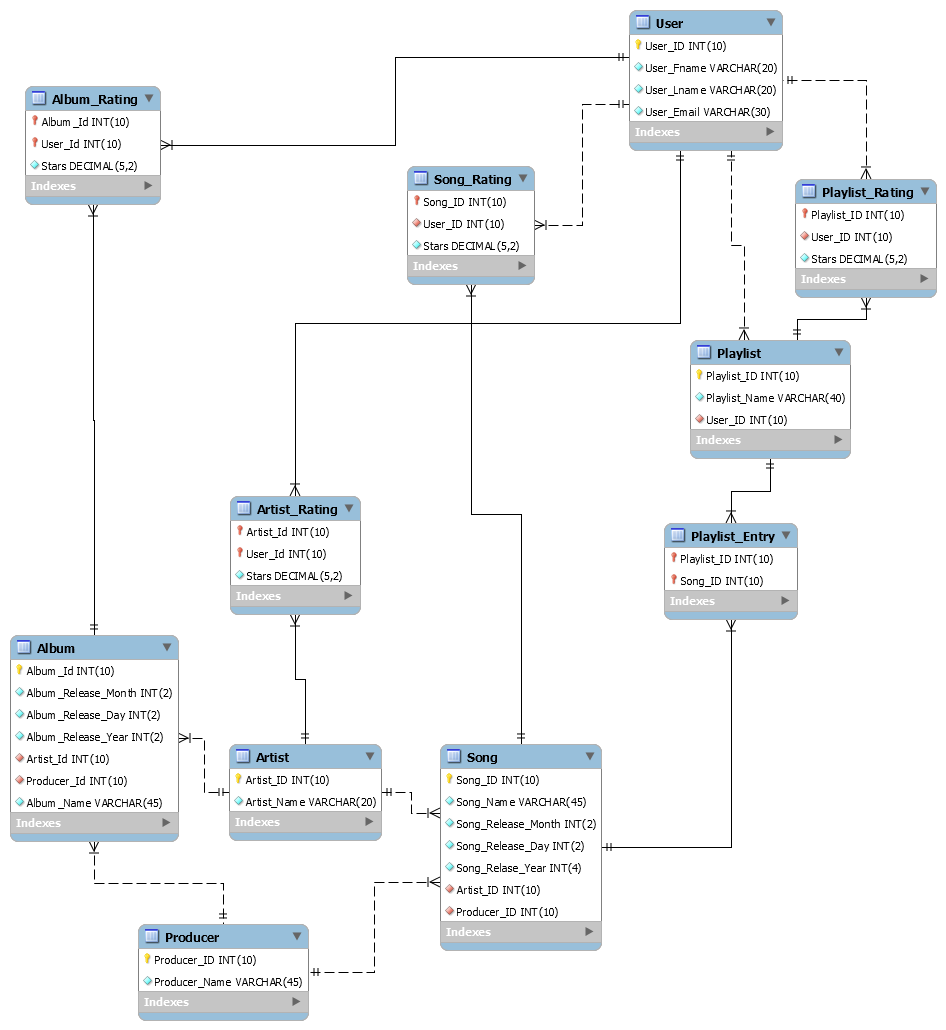
### Original ER Diagram:



### Normalization

|  |  |  |  |
| --- | --- | --- | --- |
| **Table** | **General Attributes and Data Types** | **Explanation** | **Form** |
| **Album** | Album\_ID ->PK  Album Release Month  Album Release Day  Album Release Year  Artist\_ID: FK  Producer\_ID: FK  Album Name | Album\_ID -> Album Release Month  Album\_ID -> Album Release Day  Album\_ID -> Album Release Year  Album\_ID -> Producer\_ID  Album\_ID -> Album Name | 3NF,  all of the attribute are fully dependent on the Album\_ID and there are no transient dependencies. |
| **Album\_Rating** | Album\_ID: PK/FK  User\_ID: PK/FK  Stars | Album\_ID -> Stars  User\_ID -> Stars | 3NF, all of the attribute are fully dependent on the Album\_ID and there are no transient dependencies. |
| **Artist** | Artist\_ID: PK  Artist\_Name | Artist -> Artist Name | 3NF, all of the attribute are fully dependent on the Artist\_ID and there are no transient dependencies. |
| **Artist\_Rating** | Artist\_ID: PK/FK  User\_ID: PK/FK  Stars | Artist\_ID -> Stars  User\_ID -> Stars | 3NF, all of the attribute are fully dependent on the  and there are no transient dependencies. |
| **Playlist** | Playlist\_ID: PK  Playlist\_Name  User\_ID: FK | Playlist ID -> Playlist\_Name  Playlist ID ->User\_ID | 3NF, all of the attribute are fully dependent on the Playlist\_ID  and there are no transient dependencies. |
| **Playlist\_Entry** | Playlist\_ID: PK/ FK  Song\_ID: PK/FK | Playlist\_ID -> Song\_ID  Song\_ID -> Playlist\_ID | 3NF, all of the attribute are fully dependent on the Playlist\_ID and there are no transient dependencies. |
| **Playlist\_Rating** | Playlist\_ID: PK/FK  User\_ID: PK/FK  Stars: DECIMAL (5,2) | Playlist\_ID -> Stars  User\_ID -> Stars | 3NF, all of the attribute are fully dependent on the Playlist\_ID and there are no transient dependencies. |
| **Producer** | Producer\_ID: PK  Producer\_Name | Producer\_ID ->Producer\_Name | 3NF, all of the attribute are fully dependent on the Producer\_ID and there are no transient dependencies. |
| **Song** | Song\_ID: PK  Song\_Name  Song Release Month  Song Release Day  Song Release Year  Artist\_ID : FK  Producer\_ID: FK | Song\_ID -> Song\_Name  Song\_ID -> Song Release Month  Song\_ID -> Song Release Day  Song\_ID -> Song Release Year  Song\_ID -> Artist\_ID  Song\_ID -> Producer\_ID | 3NF,all of the attribute are fully dependent on the Song\_ID and there are no transient dependencies. |
| **Song\_Rating** | Song\_ID: PK/FK  User\_ID: PK/FK  Stars | Song\_ID -> Stars  User\_ID -> Stars | 3NF, all of the attribute are fully dependent on the Song\_ID and there are no transient dependencies. |
| **User** | User\_ID: PK  User\_Name  User\_Email | User\_ID -> User\_Name  User\_ID -> User\_Email | 3NF, all of the attribute are fully dependent on the User\_ID and there are no transient dependencies. |

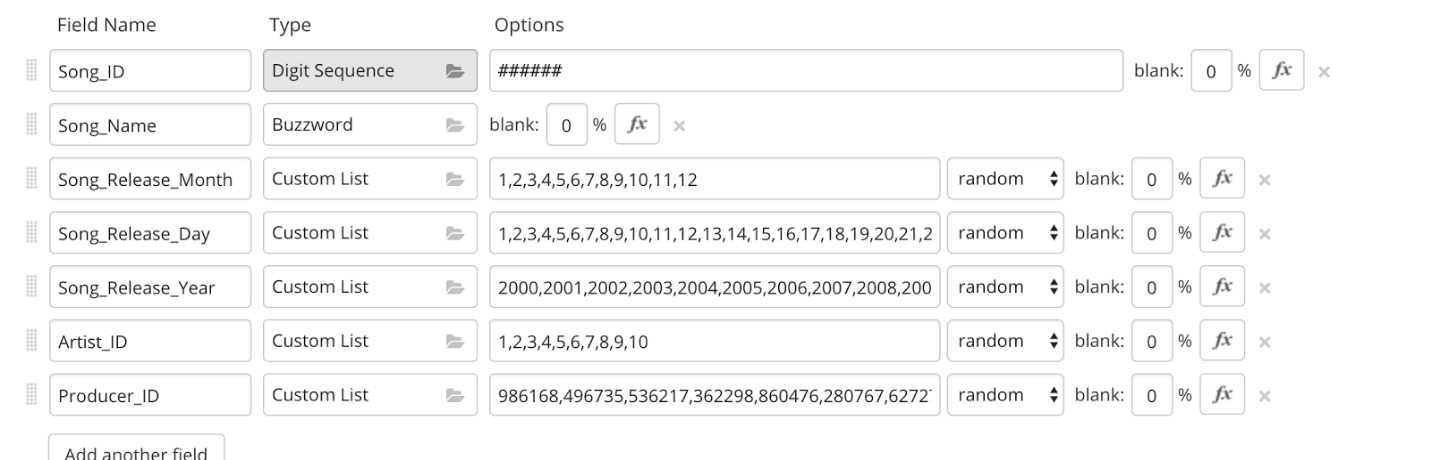
### Final ER Diagram after Normalization:



### Final Entities

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity** | **General Attributes and Data Types** | **DataType** | **Constraints** |
| **Album** | Album\_ID: PK  Album Release Month  Album Release Day  Album Release Year  Artist\_ID: FK  Producer\_ID: FK  Album Name | Int(10)  Int(2)  Int(2)  Int(2)  Int(10)  Int(10)  varchar(45) | Album\_Artist\_ID ->int(10)  Album\_Producer\_ID -> FK int(10) |
| **Album\_Rating** | Album\_ID: PK/FK  User\_ID: PK/FK  Stars | Int(10)  Int(10)  DECIMAL(5,2) | Album\_RateID  Album\_UserID |
| **Artist** | Artist\_ID: PK  Artist\_Name | Int(10)  varchar(20)  varchar(20) |  |
| **Artist\_Rating** | Artist\_ID: PK/FK  User\_ID: PK/FK  Stars | Int(10)  Int(10)  DECIMAL(5,2) | Artist\_RateID  Artist\_UserID |
| **Playlist** | Playlist\_ID: PK  Playlist\_Name  User\_ID: FK | Int(10)  varchar(40)  Int(10) | Playlist\_User\_ID |
| **Playlist\_Entry** | Playlist\_ID: PK/ FK  Song\_ID: PK/FK | Int(10)  Int(10) | Entry\_PlaylistID -> FK  Entry\_SongID -> FK |
| **Playlist\_Rating** | Playlist\_ID: PK/FK  User\_ID: PK/FK  Stars: DECIMAL (5,2) | Int(10)  Int(10)  DECIMAL(5,2) | Playlist\_RateID ->FK  Playlist\_UserID ->FK |
| **Producer** | Producer\_ID: PK  Producer\_Fname  Producer\_Lname | Int(10)  varchar(20)  varchar(20) |  |
| **Song** | Song\_ID: PK  Song\_Name  Song Release Month  Song Release Day  Song Release Year  Artist\_ID - Int: FK  Producer\_ID: FK | Int(10)  varchar(45)  Int(2)  Int(2)  Int(4)  Int(10)  Int(10) | Song\_Artist\_ID  Song\_Producer\_ID |
| **Song\_Rating** | Song\_ID: PK/FK  User\_ID: PK/FK  Stars | Int(10)  Int(10)  DECIMAL(5,2) | Song\_RateID  Song\_UserID |
| **User** | User\_ID: PK  User\_Fname  User\_Lname  User\_Email | Int(10)  varchar(20)  varchar(20)  varchar(20) |  |

### Database Populated using Mackaroo.com



### **Database Hosting**



The SQL server and webpage was hosted on Amazon webservices including:

* AWS relational Database Service:

Hosted SQL server

* AWS elastic cloud computing:

Running on Linux, installed Apache web server and connected to AWS relational database service. Server also help the php GUI

Host name: testiterationstation.cz5jcu0pis3s.us-west-2.rds.amazonaws.com

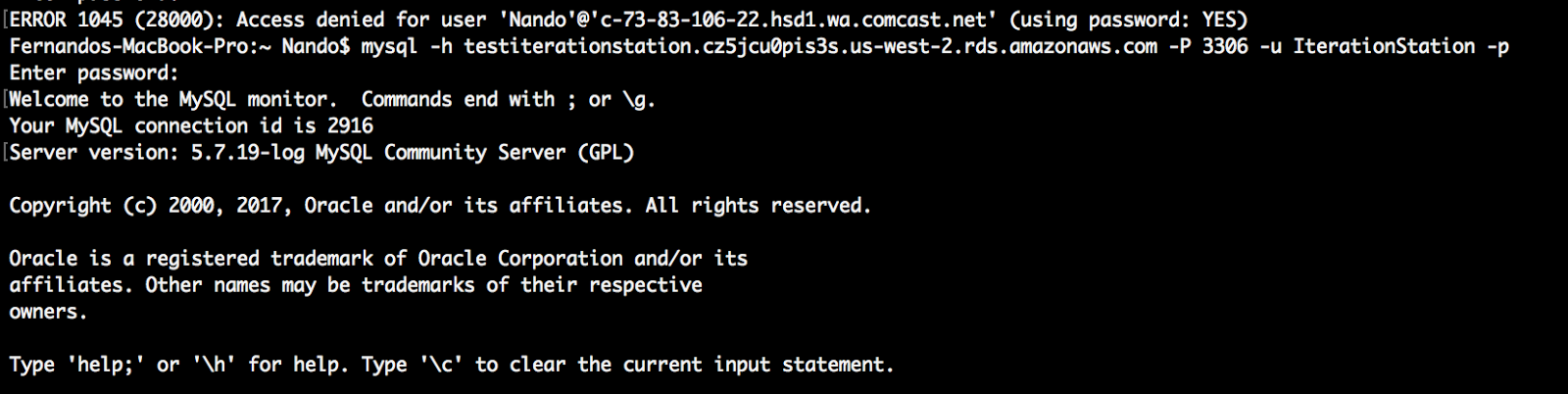
Port - 3306                    [reference Figure 1.1]

Username

IterationStation                           [Capital I and S]

Password

nicorichardfernando                  [ALL lowercase]



### **UI Tools**

HTML and php were used to display content.

Link to final web page: [ec2-54-218-50-196.us-west-2.compute.amazonaws.com/WebPage.php](http://ec2-54-218-50-196.us-west-2.compute.amazonaws.com/WebPage.php)

### UI Examples

