

# ER Diagram explanation

This is where the entities, relationships, and attributes chosen for the ER Diagram to serve as a database for the problem given in the prompt will be explained.

## Entities

### - Customer / Singer

The Customer / Singer entity contains all the information that we need about the customer. The attributes were chosen as they are the necessities for us to know about the customer and things that would be beneficial to know. It contains information about:

- Customer Email (email of the customer)
- Customer ID (This Serves as the primary key in our database + as a surrogate key)
- Customer Phone number (Their phone number)
- Name (Name of the customer)

### - Song

Chosen as an entity as it relates to the songs that the Customers will be able to select from a list of songs. It is accessed by both Customers to select a song to put into a selected queue and by contributors to note their contributions to a song. It contains information about:

- Title (The Title of the selected song)
- Year (Year In which the song was released)
- Genre (Genre of the song)
- SongID (This in combination with Version serves as the primary key, This is assigned to each song to make it easy to locate)
- Version (Selected version of the song as a given song can have different variations, such as just instrumentals or a duet version. This in combination of SongID serves as the primary key)
- Duration (Length of the song)
- Current (This is a Boolean flag we added to determine what song is currently playing)
- Image Path (This is the Song picture, We added it so the UI looks a little nicer)

### - Contributor

Chosen as an entity as it will contain information about the contributors to each song as is required by the program. It Interacts with song to describe each person's role in a song.

- Name (The name of the Contributor)
- ContribID (Primary key + Surrogate key. Identifies each contributor with a unique ID)

## Relationships

### - Queues

A relationship that links the Customer / Singer and Song entities together. Chosen as a relationship to represent when a Singer selects a song to sing along to and what position they are in a queue. It has intersection data named: Time, which refers to the time when the singer selected their chosen song. This was chosen as intersection data as it is needed for the Queue entity to know how to sort the list of singers, and useful for the Singer entity as to show what songs they have sung in the past. This will be implemented and viewed on the website.

The cardinalities are as follows:

Customer / Singer – (1, m) – For each song, there can be many customers that chose to sing that song.

Song – (1, m) – For each Customer / Singer, there can be many songs that they have sung, which makes sense, since only the client that made the appointment at that time has the appointment reservation, and not any other client.

### - Priority Queues

A relationship that links the Customer / Singer and Song entities together. Chosen as a relationship to represent when a Singer selects a song to sing along to and what position they are in a queue. This is different from the queue above as the customer must pay money to be in this accelerated queue. It has intersection data named: Time, which refers to the time when the singer selected their chosen song, and Money, which represents if the customer chose to pay to be in the priority queue. These was chosen as intersection data as it is needed for the Queue entity to know how to sort the list of singers, and to know if they paid to be inserted into the priority queue. This will be viewed through the website and be ahead of the normal queue.

The cardinalities are as follows:

Customer / Singer – (1, m) – For each song, there can be many customers that chose to sing that song.

Song – (1, m) – For each Customer / Singer, there can be many songs that they have sung, which makes sense, since only the client that made the appointment at that time has the appointment reservation, and not any other client.

### - Contributors

A relationship that links Contributor with Song, in which it describes the role a contributor had in a song, with intersection data of Role. This intersection data is included to say what role a person had in a song. This will be useful to search for songs by a contributor.

The cardinalities are as follows:

Song – (1, m) – For each Contributor, there are many songs that they could've participated in. This makes sense since a particular artist can play multiple roles and/or play multiple songs.

Contributor – (1, m) – For each Song, there are many contributors that can participate in creating the song. This makes sense as it can either be a solo artist, a band, or a writer that contributes to a song.