CSCI 466

Karaoke Group Project

Team B

Relations, Entities, and Attributes

In our ER Diagram, we have a user that will have userID, timesSung, and name. Since multiple people can have the same name, we decide to use a userID so that we can make it the primary key and make it easier to differentiate between two different people. The user then has access to both the paidQueuedIn and the freeQueuedIn, and the relation to both is (0,m) since there can only be one user (0) but that user can have many songs in either queue (m). In the paidQueuedIn, we have four attributes, being date/time, amountPaid and two foreign keys being userID and versionID. The userID is the primary key that checks to see what user is in the queue. The versionID also is a primary key since the user will have to pick a song and the version of that songs to add to the queue. The date/time keeps track of the time each user joins the queue. Lastly, we have the amountPaid which is how we keep track of the queue. The higher the paid amount is, the higher up the queue the user will be. In our freeQueuedIn entity, we have three different attributes being are date/time, userID and versionID. userID is a foreign primary key, alongside versionID and they are the same as in paidQueuedIn. Date/time is important here, since the date/time attribute keeps track of the queue and the queue is from earliest to latest. Both the freeQueuedIn and the paidQueuedIn have a relation to KaraokeVersion that is (0,m) since both queues can have many songs inserted (m) but can only be inserted one at a time (0). The KaraokeVersion entity has two attributes to it, and they are the versionID and the versionName. The versionID is the primary key and it is how we identify between different variations for one single. If a song has a version that allows for a duet, then that version will have a separate ID than the same song but for solo performances. The versionName is the name given to the song variation, so it is either a solo song, duet song or trio song. The KaraokeVersion has a relation to the song entity, where “each karaoke version has a song” and that relation is (1,m). Each KaraokeVersion has only one song (1) but can have up to the three separate files (3). On the other hand, the relation from Song to Karaoke version is “each song has a karaoke version” and it is (1, 1) since each song has one karaoke version and vice versa (1,1). The Song entity has two attributes, and those are songID and Title. The songID is the primary key and it helps us know what song is actually being requested since multiple songs can share titles. The Title is an attribute that can help us narrow down the search on what song is being requested. The Song entity then has a relation to the Artist entity, and it is “song that was worked on by an artist”. The relation here is (1,m) as each unique song (1) can be worked on by multiple artists (m). On the other way around, the relation from Artist to Song is “each artist worked on a song” and the relation here is (0, m) since many artists can contribute to one song (m) but a song can technically not have an artist contribute to it (0). Our last entity is Artist, and it contains two attributes being artistID and name. The artistID is the primary key as multiple artists can share the same name, so this helps to narrow it down to one artist when searching for one.