Keaton Raymond
CS 4420-001

Music Chart Database Project

Overview

After examining the original table, I went through a couple different iterations for a design. Eventually I settled on which design I believed to be the best. This design includes 6 tables: label, people, ranking_details, ranking_overview, track, and track_writers. I believe that this was the best way to remove all dependencies and put the table into 3NF. Below I will describe in detail the contents of all 6 tables.

label

The label table includes an auto increment label_id key as well as the label name. Because the label has no other dependencies and nothing else associated with it, I believe this was the best way to handle that field.

people

The people table includes an auto increment person_id as well as a field for the person's name. Because a person can be either an artist or a writer, it makes sense to have a single table that holds all the distinct people entities and, depending on how they are referenced, determines whether they are the artist or one of the writers. Doing it this way eliminates as many repeat names as possible.

ranking_overview

The ranking_overview table is the first of two tables that deal with the rankings for each track; this table handles the overview for each track and has all the fields associated with the rank, aside from rankings for weeks 1-76. The fields of this table are: track_id (acts as both a foreign key to the track table and a primary key), weeks_ch, weeks_40, weeks_10, yearly_rank, weeks_peak, highest (the highest rank the track reached), date_entered, and date_peaked.

ranking_details

The ranking_details table is the second of two tables that deal with the rankings for each track. This table handles only the details needed to handle the data in the week1-week76 columns from the original table. The fields in this table are: track_id (acts as both a foreign key to the track table and part of the composite key), charting_week (the second part of the composite key), track_rank (the rank for that particular track on that particular week), and position week (the date where the charting_week occurred).

The reasoning for the composite key is that with knowing only the track_id, there are many different charting_weeks, track_ranks, and position_weeks, which would result in a nonunique key. However, there is only 1 unique combination of the track_id and the charting_week and, knowing the two of those, it is easy to discern the track_rank as well as the position_week.

The reasoning for the ranking details being split into two tables is to eliminate as much duplicate information as possible. Each track only references 1 ranking_overview entity but could reference as many as 76 ranking_details entities. Because of this, if the tables were combined, there would be up to 75 occurrences of duplicate data for the information stored in the ranking_overview table.

track

The track table holds all the fields that are directly related to the track. The fields in this table are: track_id (an auto increment primary key), track_year, track_title, track_time, label_id (a foreign key to the label table), bpm, track_comment, prefix, artist (a foreign key to the people table that references the person who is the artist for that song), and lab_num.

track_writers

The track_writers table is a linking table between the tracks and the writers. A linking table was necessary here because a single writer can write many songs and a single song can have many writers, thus causing a M:N relationship. The fields in this table are: track_id which (a foreign key to the track table), person_id (a foreign key to the track table and references the writers of the track as opposed to the artists referenced in the track table), and writer_num (the position the writer was listed as for each of the unique writer track columns). This difference in how the people table is referenced, whether in the track_writers table or in the artist field in the track table, is how the programmer can discern between whether a person was a writer or an artist on any given track.