Requirements Checklist

**SQLite back end:** We created our GUI using Java swing. This interacted with an sqlite backend via use of the JDBC library.

**Coded using SQL, not higher-level primitives**: Whenever interacting with the database, we used sql code in conjunction with the JDBC library.

**Mass load feature**: This feature is implemented via the import button. When pressing the import button of the application, the program will look for a file named “input.csv”. It will delete the current database and import all the data it parses from the input.csv file.

**Report feature**: This feature is implemented via the export button. When the user presses the export button of the application, the program will export the current database library to the file “output.csv”. This output file will be in the same format as the input csv file.

* + Features that exhibit the following abilities in, or in conjunction with, SQL:
    - Represent the input data using tables
  + We have several tabs in our application, five of the tabs are used to show the data within our five tables. Specifically the data from the songs, artists, songArtist, songFile, and files tables are all represented through tables.
    - Build relations across input data tables
  + We have three tables: songs table, artists table, files, table
  + Then we have 2 additional tables that act like maps between the three tables mentioned above. We have a table mapping songs to files and another table mapping songs to artists.
    - Ask questions answerable from a single table
  + In our main tab of the application, users are able to query for which artists originated from a given country or state.
    - Ask questions answerable using more than one table (joins)
  + Once again, in the main tab of our application, users are able to query which songs were created by a given artist. This looks across the songs table and the artists table.
    - Use views
  + Under our main tab, we have a textbox area that is supposed to be representative of a view. Rather than creating a dynamically generated table to display that information, the textbox area simply shows the same data as strings in a .csv format. There are 4 views that are shown.
    - one shows the songs created by a given artist
    - one shows artists who had a given debut year
    - one shows artists from a given country
    - one shows artists from a given state
    - Use foreign keys
  + In our songArtist table and songFile table, there exists the use of foreign keys. The billboardYear and rank fields from the song table act as primary keys in the song table. However, they act as foreign keys within the songArtist and songFile table in order to reference the rows in the song table
    - Summarizing data (Average, mean, median, histogram, etc)
  + In order to summarize data, we displaying the following statistics:
    - Percent of songs featuring multiple artists
    - Percent of artists that is a group
    - Average size of file in bytes
    - Average length of files in soncds
    - Mutate the database by deleting data that meets some non-trivial criteria, specifically in at least one case where the delete affects more than one table and is likely to affect multiple rows of at least one table.
  + The user has the ability to delete a song, an artist, or a file.
  + Cases:
    - Delete a song
      * When a song is deleted, the corresponding file to the song is deleted. In addition, the mapping from the song to its artist as well as the mapping from the song to its file is deleted. If the song was the last song the artist has, then the artist is also deleted.
      * All tables are potentially affected.
    - Delete an artist
      * All songs corresponding to the artist are deleted as well as the artist. The mappings contained in the songFile and songArtist table are also deleted.
      * All tables are affected.
    - Delete a file
      * The corresponding song to the file is deleted. If the artist only had this one song, then the artist is also deleted. The mappings contained in the songFile and songArtist table are also deleted.
      * All tables are potentially affected.
    - Allow the user to add additional data with the same fields as the original input
  + Within the main tab of the application, an user can add a new song by filling out the appropriate fields. If the fields are filled with valid information (not null; no commas, quotes, or apostrophes are contained; numerical values are logical), then the data is added.