INST327 – Section 0203 Team 8

Jason C. Bae

**Final Project Report**

Due: May 14th, 2021

**Reflections (Project Diary and Report)**

*Introduction*

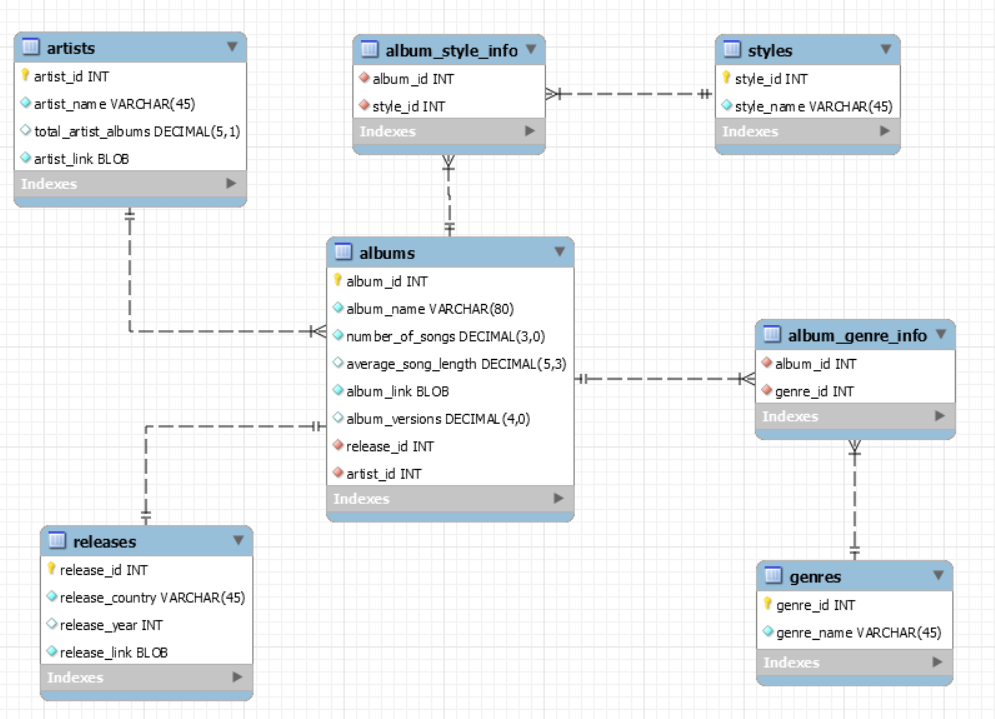
Many users encounter difficulties navigating through which type of music they are more inclined to listening as they browse through online music stores. Furthermore, their interest is easily distracted from thousands of available songs that serve as a hindrance to discovering their true musical taste. To reduce such difficulty, my team has developed a database that can provide a relational database structure understanding to enhance user experience with an interactive visual database to accurately assess user’s musical tastes and provide accurate analysis by displaying relevant songs of their interest.

From the wide range of music selections available in Discogs.com website, our database is focused on only a fraction of the available resources to better gauge user’s interests. By understanding user’s musical taste with more accuracy, our database can gather information about the user and conduct attentive analysis that is based on discovering about the user more than to just make a sale as promoted by the Discogs.com website with its huge selection of choices. Also, our database aims to provide an enhanced user experience with relational database structure that allows us to identify and access that are specifically related to their interests. Our database is separated by 5 tables and 2 linking tables that offer the users with many-to-many relationship understanding between tables. Lastly, our database has been simplified to liberate users from unnecessary information that distract the users from their original goal with their musical interest exploration, causing delayed.

*Database Description*

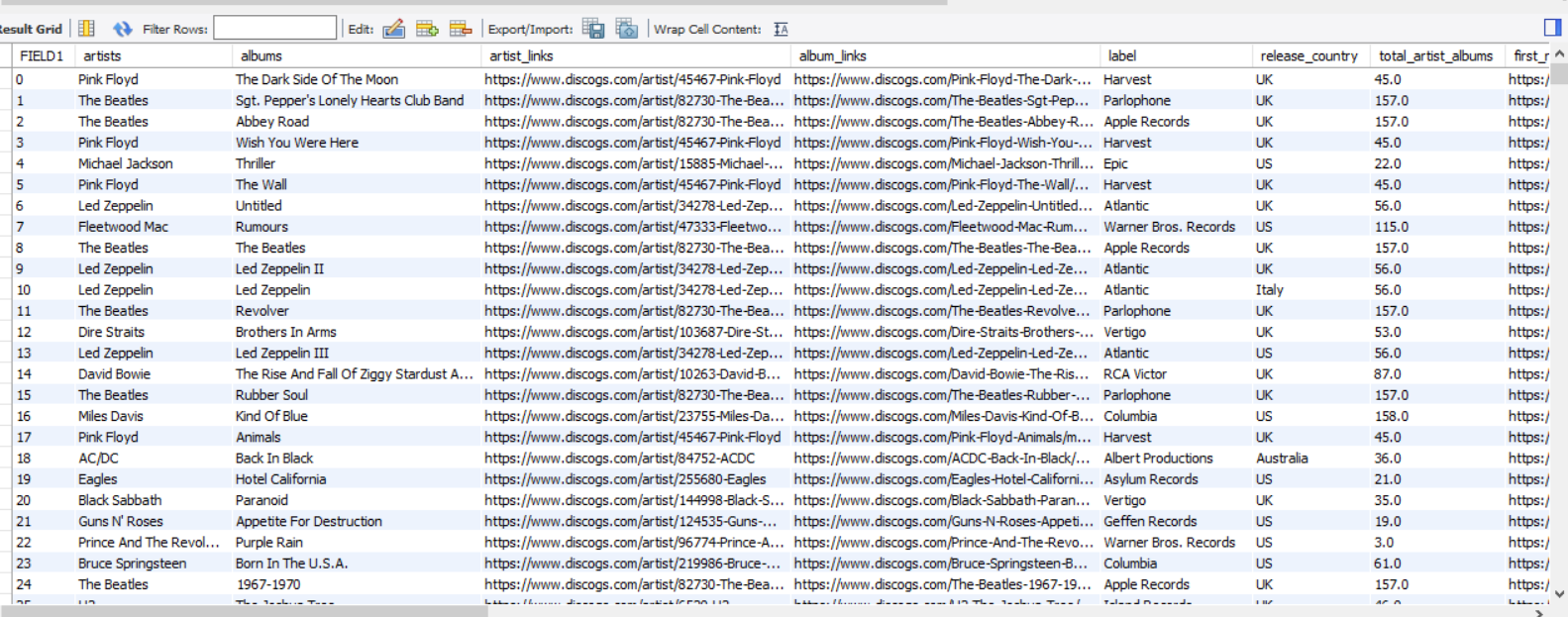
As mentioned, our database is comprised of 5 tables (artists, albums, styles, genres, and releases) and 2 linking tables (album\_genre\_info and album\_style\_info). Our database requires linking table to establish many-to-many relationship and to increase communications in the middle of promote database’s effect. Relational structures like linking tables allow users with an interactive experience that allows more flexibility in their experience with Discogs.com website, compared to the website that also includes unnecessary details.

Initial design of the database stemmed with the ERD diagram. As shown below at **Figure 1**, the database contains 7 tables with names: “artists”, “releases”, “styles”, “album\_style\_info”, genres, and “album\_genre\_info”. From the 7 tables, “album\_style\_info” and “album\_genre\_info” tables serve as a linking table for the many-to-many relationship that the “albums” table shares with “genres” and “styles” tables, because each album can have many genres and styles while each genre and each style can have multiple albums that fit the description.



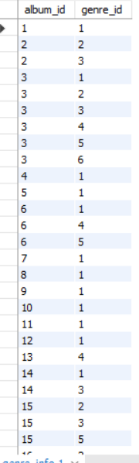
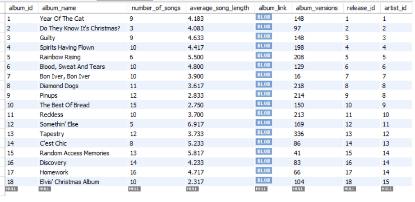
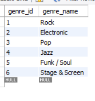
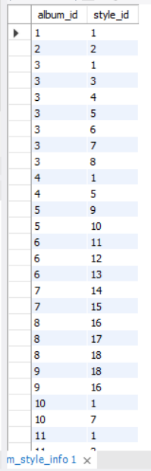
**Figure** 1. ERD Diagram.

After identifying the necessary attributes per entities, data must be first located and then imported into the database, so that it could display the logical flow shown by the ERD Diagram. Figure 2 displays the database that my team was able to locate online, which contains various information of Discogs.com website. However, since the database is displaying as only a table, it fails to fulfill the project requirements by not allowing relational database organization that would display more manipulation of the data to enhance user experience.



**Figure** 2. Source Database that displays Discogs.com website information.

Even though the source database failed to establish correct database organization in multiple tables, the source did contain various information that we could include into each entity for our database. Thus, our database imported data, translating from the source database, to be allocated into correct attributes in each entity, resulting the following entities shown below from Figure 3.

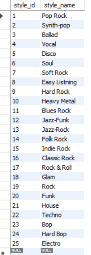
e

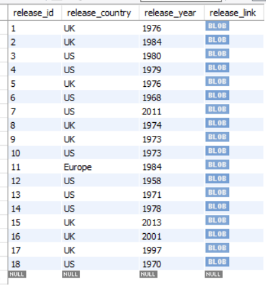
a

d

c

b





g

f

**Figure 3**(a-g). Each table used in the Database with each letter representing the tables.

Entities that contain the displayed elements from Figure 3 are as follows:

Figure 3(a) – “album\_style\_info” table

Figure 3(b) – “album\_genre\_info” table

Figure 3(c) – “albums” table

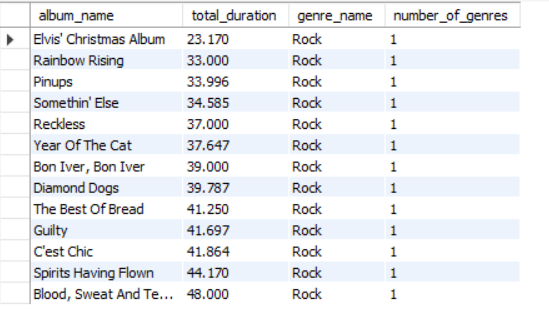
Figure 3(d) – “artists” table

Figure 3(e) – “genres” table

Figure 3(f) – “releases” table

Figure 3(g) – “styles” table

After allocating correct elements that fit the entity description as explained through the ERD diagram on Figure 2, our database had to serve its functionality that was to enhance user experience by identifying user’s musical interests and displaying only the relevant information. Thus, a view table was created to explain one of the possibilities that is allowed by our database. As shown below in Figure 4, our view table contains only the albums, out of the total 18 albums allowed, that has a genre as “Rock” and is the distinct genre, so that user can discover the music that has only “Rock” as its genre. So, the users with a strong “Rock” musical interests can expect their musical taste having been met from the albums shown in Figure 4.



**Figure** 4. View table result.

*Changes from original design*

During the final database development, our team faced numerous challenges that required changes from the original design. The main challenges that caused our database to deviate away from the original design was during information import, due to “release\_country” in “releases” entity containing more than one country, non-uniform date format in “release\_date” (Now “release\_year”), and some of attribute information was returning null for values that should not be null. Therefore, compared to the original design, our database had to select only the information that met the requirements set by the ERD diagram. Also, database attribute name for “release\_date” had to be switched into “release\_year”, due to various date formats provided by the source. Since some of date contained information down to an exact date while some did not, database had to be adjusted so that format can be uniform throughout the database.

Also, the database design had to be structured differently than to the original design, due to the database intentions. Prior to the final project, the main goal of the database was to display various information that are offered by Discog.com website, but with the development of its secondary goal to display correct music to suit user’s taste, our database undergone which type of information it collected and how it displayed them.

*Lessons Learned*

From this project, we were challenged to various concepts learned throughout the semester and translated our learning experience into that of a database that we had been working with throughout the database. During the project, students were tested for view table understanding, various query logics, normalization, and ERD logics. However, real values that were offered and perceived by us during this project came strongly in the real problem encounters that could not be translated from the theories and concepts discussed in class. For example, we were educated on normalization, but we did not get to realize various null value difficulties and requirements translated from how the attribute type was structured. Thus, this project served a great purpose in allowing students to practice the SQL concepts learned in class, inclusive of real-life challenges offered during the database development.

*Potential Future Work*

From the project requirements, various concepts and manipulations could be realized and represented by the database. However, if there was additional work that could be possible, we could consider about creating more view tables to display various functionalities that are commonly requested to be identified, so the users can locate the view table to narrow down on their interests.

Additionally, in the future, the database could contain more elements, so that it could represent the website with more accuracy as to allow more immersive user experience with the database. Since the database can only display few information, it can be difficult to be more specific to answer to user’s interests. Also, more attributes in each entity and more entities can offer to be more selective in which music should be displayed to offer the users with more selections and to enhance musical interest identification.