INST 327 - Database Design and Modeling

Group 6 Sec 203

Chris Summers, Arthur Wu, Andrew Cunniff, Ucheoma Okonkwo, Desmond Lam

5/8/2020

Team Project Final Report

Introduction

The United States music industry is one of the largest entertainment markets in the entire world. With a revenue of $11.1 billion in 2019, one can clearly see that the music industry is extremely important to the United States' economy and influence across the globe (Rys, 1). To document the most celebrated songs and artists within this industry, weekly and yearly leaderboards of the top 100 songs are compiled by Billboard, an established company within the music industry that tracks the top trending records sold and streamed in the United States. Although Billboard documents trending songs on a weekly and yearly basis, Billboard does not offer any lists or collections that allow efficient comparisons of top songs across decades or longer periods of time.

Therefore, our team developed a database of songs ranked in the Billboard Hot 100 over multiple years. We decided to make this dataset because all members of the group have interest in music and would like to see how the United States music industry has developed and changed over the years.

For the database, we specifically want to analyze the top songs on both a weekly and yearly basis, so our sample data focuses on data from Billboard's Year-End Hot 100 ranking charts and Billboard's weekly Hot 100 ranking charts. For the years between 1964 and 2020, There are 100 songs listed in each year's Year-End Hot 100 chart and 100 songs for each week within each year. Therefore, we estimate that the completely populated database would contain approximately 269,500 song entries. In our database, we included attributes such as the song name, artist, album, and lyrics for each song. We also included attributes that relate to the weekly popularity of a song such as weekly ranking, and the number of consecutive weeks a song stays on the Hot 100 chart which will be achieved by drawing data from both Billboard Year-End and Weekly Hot 100 lists.

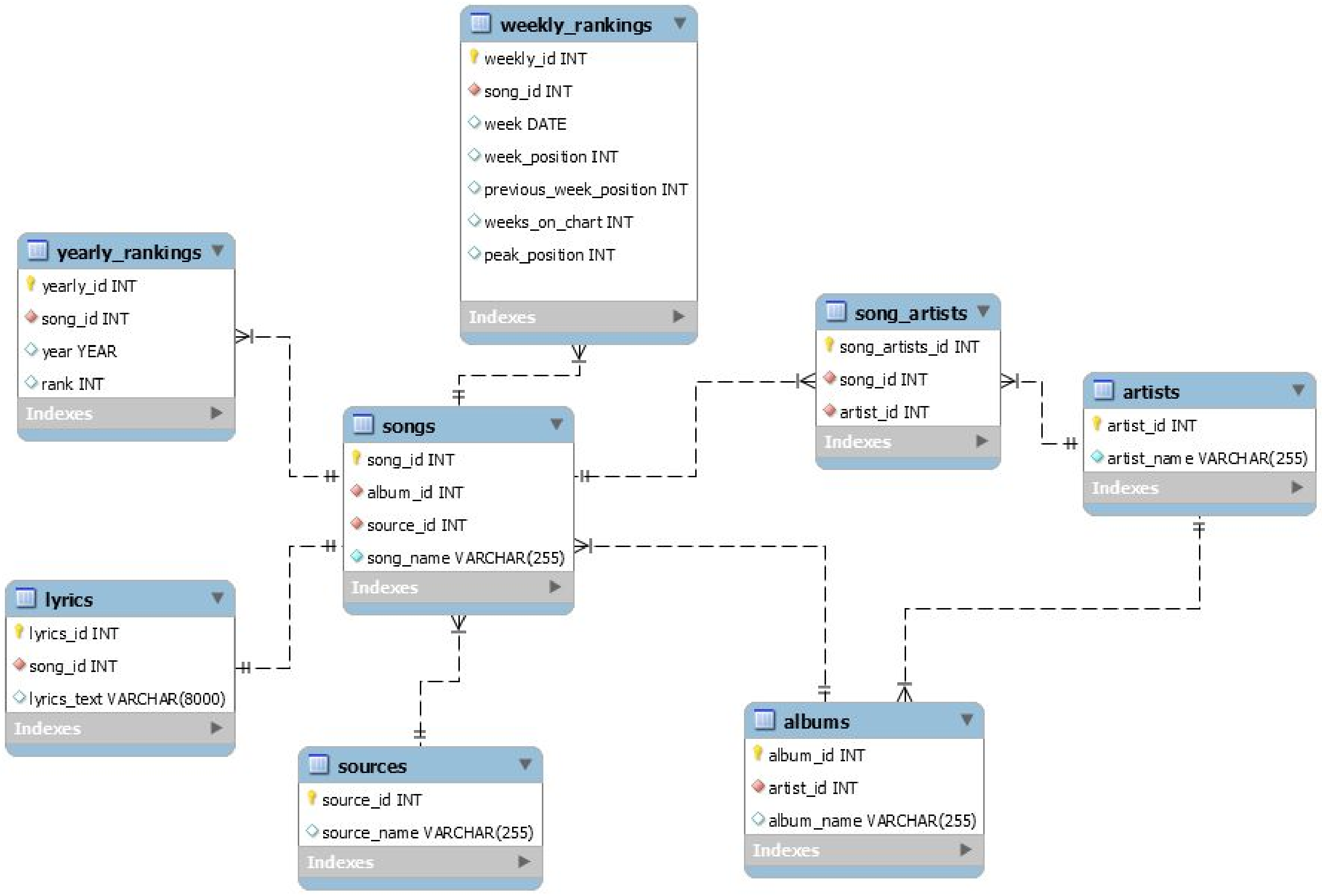
The target audience for our data set is aimed at those curious to survey the iconography of the most popular songs over multiple years. Our team hopes that through our database, our audience would be given the ability to implement particular queries about the variables included in our dataset. This will be particularly useful for those studying the history of United States popular music and recording companies who want to predict future trends in music popularity based on past trends.

Database Description

Logical Design

In designing the structure of our database, there were several key decisions made in order to maintain the relationships between each entity. To begin with, our song table is the main focus of our database, and therefore serves as the central entity that is linked to all other tables in some fashion. The relationship between songs, albums, and artists forms a sort of loop with one another. An artist can have many songs snad many albums. An album can have many songs, but only one artist. A song can only have one artist and one album. This determines which table receives the foreign key of the other. The relationship between lyrics and songs is very much one to one. One song can only have one (officially) associated lyrics. The relationship between songs and sources is many to one, as one source can have many songs, but not the other way around (due to the nature of our data). For rankings, a song can have many rankings, but because rankings are based on a year or week, one ranking can not have many songs. This is the logical conclusion.

When it comes to the design of our tables, for the artist table specifically, it was decided that artists would not receive an artist_first_name and artist_last_name column respectively due to a large number of artists going by one named aliases, such as "Beyonce", "Drake", or "Adele". These aliases are typically not linked to the artist's real name, therefore it would feel illogical to use first name and last name columns. Many artists listed are also classified as bands, therefore using a first name, last name system would have to include each member's name. Since we don't have that information, let alone the majority of artist first names and last names to begin with, using a simple artist_name column was the best choice. As for the issue of songs that feature multiple artists, we created a linking table to accomodate for an instance of multiple artists being attributed to the same song. Here is the diagram of our ERD:



Physical Database

Our MySQL database had been shared amongst teammates using a single “dump” sql file accordingly. Teammates who had downloaded the backup of our database reported no issues executing views and queries. Using basic SELECT queries displays all of the integrated sample data that has been imported into their respective tables. Despite initial confusion and issues with our original database, the exchange of files themselves was not a problem and each time the database was backed up and downloaded it ran as it was designed to. Overall, our database and backup databases load correctly in MySQL workbench.

Sample Data

The main source used for sample data in our database is RakenNimer’s “Billboard 1964-2015 Songs + Lyrics” dataset retrieved from his Kaggle account[1]. This dataset includes basic descriptive information such as song name, artist name and year for all songs featured on Billboard Year-End Hot 100 from 1964 to 2015. We also used the ‘Hot Stuff’ and ‘Audio Features’ datasets from Data.World by Sean Miller[2] to obtain more information on the songs listed on RakenNimer’s dataset. Some additional columns that Sean Miller’s datasets will supply are song genre and song rank position on the chart. By combining the data from these two datasets, we will be able to obtain a complete set of data for all songs featured on the Billboard Hot 100 from 1964 to 2015. These two datasets are also formatted as csv files, which made importing data into our database a simple process.

To account for the limited time we have to work on the database, our group used sample data from two yearly hot 100 lists and two weekly hot 100 lists. The years that we drew sample data from were 1970 and 1988. The weeks that we chose to draw data from were 8/2/1958 and 1/25/1986. To organize and successfully import our datasets as sample data for the database, we first downloaded both datasets and sorted the data into separate csv files by table; The artist name data is placed in the csv file for the artist table, the

[1] https://www.kaggle.com/rakannimer/billboard-lyrics/data
[2] https://data.world/kcmillersean/billboard-hot-100-1958-2017

album name data is placed in the csv file for the album name and so forth. After sorting all data from the two datasets into separate csv files by table, we imported each table csv file into the tables ontlined and developed in our ERD, carefully following the foriegn key structure of our database.

Views / Queries

View Name	Req. A	Req. B	Req. C	Req. D	Req. E
MostLyrics	X			X	
Top10	X	X		X	X
FirstWord		X	X		
BetterThanAvgPeak Pos	X	X	X	X	X
ArtistsWith2+Songs	X	X	X	X	

Changes From Original Design

We had experienced many changes from our original design, both in the sample data we used and the structure of our database itself. For starters, we had to remove a few tables from our original database concept, including the databases, song rankings, and (unintentionally created) songs_has_song_rankings tables. While importing our sample data, one of our original samples had been removed from the internet. This led to the removal of the database table. The removal of the other two was a result of confusion in determining the relationship between songs and song rankings. The data types had to be changed as well to accommodate larger values than originally expected (eg varchar(45) -> varchar(250) or even as high as varchar(8000)). We had also decided to change our range of years from the past 50 years to a much smaller time period due to limitations and issues with data cleaning.

In importing our sample data, many changes had to be made to the data itself. For the billboard lyrics table, any value that wasn't an integer in the 'source' column was changed to 404 and any rows that had a value that contained an irregular character (any character not on a regular keyboard) were removed. Rows containing irregular characters were also removed from the audio features dataset.

Lessons Learned

While developing our final complete database, we not only learned the process of structuring and populating a database, but we also experienced issues that hurt our progress and learned many lessons in regards to troubleshooting and fixing errors in data and in code. One major issue that we came across was that the dataset we drew data from did not provide the data in the format that would allow our database to be normalized. An example of this is the artist name column, where some names included all featured artists within the same name entry. To fix this, we were required to decrease our sample data size and manually clean the data within the dataset which caused stress for our team and took a great deal of time. From addressing this issue, our team learned that it is important to develop solid and complete sample data before importing into a database. If we skip the data cleaning process, many more errors will appear in our database which will cause greater stress in the future. Another lesson we learned is that more data leads to more errors. Initially, our group planned to import 50 years of yearly and weekly hot 100 lists into our database which resulted in approximately 30,000 entries. As we attempted to clean and import the large amount of data, we realized that there were many underlying errors related to syntax and data matching that prevented us from successfully importing the complete dataset. From this, we learned that it is better to import small amounts of data at a time to maximize the perfection and to minimize errors. Overall, our group learned many lessons regarding data and database structure, and by resolving the major issues within our database, we have a better understanding of data science and database development.

Potential Future Work

We could have the opportunity of creating a similar ERD database like this again. As we were struggling through how to create a proper database like this one, we could be able to find ways to improve and expand upon it. Possible future work that we can perform to further develop and improve the database include addressing current issues within our database and updating the database with new data and tables. One particular issue that exists within the current version of our database is the incomplete and faulty datasets that we draw sample data from for our database. As mentioned in our sample data section, our team performed data cleaning on both datasets to remove song entries that had strange unicode characters which caused errors in the importing process. A potential solution to address this issue in the future is to find a more complete dataset that does not contain data errors or to develop our own dataset through web scraping. Another improvement that we could add to our dataset in the future is to update tables to include missing columns such as song genre and song release date. This will make our dataset more useful and provide more information to users. We can also consider maintaining the database by inserting data from future Billboard Hot 100 charts. Also, we can look into developing databases for other music charts in the future as well. Besides the popular Billboard Hot 100 charts, we can look into developing similar databases for other Billboard leaderboards such as the Billboard International chart or Billboard charts for the top songs by genre. Overall, there are many more improvements we can make to expand upon our current database, and many other potential projects that we can initiate to organize data within the music industry.

References

Rys, Dan. “US Recorded Music Revenue Reaches $11.1 Billion in 2019, 79% From Streaming: RIAA Year-End Report.” *Billboard*, 25 Feb. 2020, www.billboard.com/articles/business/8551881/riaa-music-industry-2019-revenue-streaming-vinyl-digital-physical.