**Integration Process**

At the beginning of our thought process, we wanted to find a way to retrieve a large amount of unbiased data from one of the API’s in a single query. We found that the Last.fm API enabled us to retrieve raw data for the top 1000 artists in their database. We retrieved the data using a python script and saved it in JSON format. The reason for saving the data in JSON format is to keep our API requests at a minimum.

Afterwards, we created a python script to filter the relevant data we were interested in from the JSON format, and saving it in CVS files.

In the process of converting our data from JSON format to CSV files we noticed that some artist names included undefined characters in the English language (for example – Beyoncé). We decided to exclude these records. We were left to work with 887 artist records.

Saving our data in CSV files enabled us to isolate problematic records that needed fixing using the filter and sorting features in EXCEL.

We wanted to add information to our Artists database, for example, their music’s genre, and we found this data from the MusicGraph API. The MusicGraph query was based on the artists name, but because the name was not a key, we were missing some artists. When we went back to check the reason for this, we noticed that the artist name was different in each API (for example, with or without the word “the” in the beginning). In order to fix this, we wrote a python script to overcome differences in the artist names and to receive all of the data we needed.

We were interested in adding information regarding our albums. We considered using the Last.fm API but unfortunately their queries returned many duplicated and inconsistent data (for example different release dates for the same albums), and so we decided against using this API. We decided to use the MusicGraph API instead to retrieve this data. The MusicGraph API returned some duplicate albums as well, but we solved this issue using the excel features.

Regarding the tracks, we decided to retrieve all of the songs in at most three albums per artist (we wanted to have at least 20,000 records but we didn’t want to have too much data). This data, too, contained duplicates which we eliminated with EXCEL features.

Afterwards, we retrieved the lyrics for each track in our database using the Lyrics.ovh API. To increase the success rate of the lyrics query for each song, we removed redundant characters from the track names in the track database (for example “(live)”, “[remix]”, etc.).

After retrieving the lyrics to our tracks, we noticed that many of the lyrics texts started with the same sequence of Spanish words (which are not a part of the actual lyrics song). Therefore we removed this sequence using a python script.

Lastly, we retrieved the data for our events database using BandsInTown API. After retrieving the data we noticed that many artists did not have any events, and the reason for that is that these artists stopped preforming.