**Project 2, Group 17’s Report**

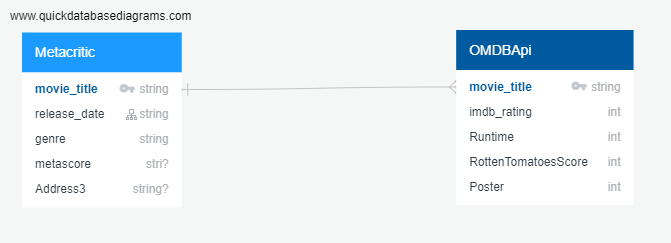
Jonathan Michel, Haslett Fernandes, Justin Ho

The Trilogy

**December 8th, 2022**

**Extract Transform and Load**

The data sources we gathered to represent our ETL project were the OMDb API and a Metacritic csv file utilized from Kaggle. The Metacritic Database(MetaDb) served as the base dataset for future additions and manipulations; MetaDb gave baseline information on movie information such as release date, genre, Metacritic score, and most importantly, the 9120 movie titles that would serve as a reference and the primary key for the dataset. The information gathered from the OMDB API functioned mainly to append onto the Metacritic Dataset by including Rotten Tomatoes rating, IMDB rating, and other pertinent data dependent on the film basis.



To make things easier, some data cleaning was needed so only the data we needed was displayed. After loading the MetaDb csv into a pandas dataframe, we dropped the following columns: “meta\_mixed”, “meta\_negative”, “meta\_positive”, “user\_mixed”, user\_negative,” “user\_positive”, and ”user\_score”. We then created new columns such as “Imdb\_rating”, ”Runtime”, ”Rotten Tomatoes Score”, and ”Poster” to fill them with data from the OMDb API. This gave us a clean dataframe that could be easily manipulated. However, the first issue we encountered is the limit of the API key for the OMDb api. The key had a query limit of 1000 per day and so we decided to split our large dataframe of over 9000 films into ten separate dataframes that would each have their own individual API key. Once each data frame was created we carefully loaded in the data from the OMDB api with success.

To clean our data, we wanted to format the three review scores into a percentage to make them more uniform. There were some inconsistencies in the formatting of the movie scores. The “imdb\_rating” was formatted as a float, the “metascore” was formatted as an integer, and the rotten tomatoes score was formatted as a percentage. There were also a few issues with some data not being present or not being formatted correctly with the rest of the column. However, once our data was cleaned and properly formatted, we utilized sqlalchemy in order to load our data. From here, we were able to combine them all together to create one final movie database with about 9000 rows of data!

The limitations of our data that can be improved upon starts with the api key for OMDb. The query limit of 1000 per day did slow us down but we still managed. Another limitation we noticed was that the OMDb api only contained the data of critic reviews and not the reviews of the general public. It would be an improvement to see the audience review along with the critic reviews for Metacritic and Rotten Tomatoes.