Collaborators: Kelvyn Guzman, Erik Osterlund, Francisco Zuluaga

UPenn Data Analytics & Visualization Bootcamp

ETL Project

**EXTRACT**

For this project the subject matter is movies and their respective awards. We are utilizing an IMDB dataset in a CSV format found on Kaggle.com. For the awards we are using an Oscar Awards dataset CSV which we also found on Kaggle.com. There were many movie datasets on this site but with most of the initial datasets we identified the lists were incomplete. We wanted to ensure that we could include awards for foreign movies and include as many motion pictures as well. By starting with the extensive IMdb dataset, we acquired the ratings from critics and users. The selected datasets create a unique relationship that can be expanded if needed in the future.

**TRANSFORM**

We are using Jupyter Notebook and pandas to transform the data as it was a seamless process loading the data from its CSV format. We cleaned up the data by removing unwanted columns and rows that had missing data. To better understand the data and the different technical aspects of it, we educated ourselves on the requirements of each award category and basic knowledge of the hierarchy of the awards.

We first cleaned up the data by searching through the column titles and deciding on which columns would match up between the two dataframes, and which columns were not as meaningful to our finalized dataframe. Utilizing a .unique() function, we were able to list out each award that had been won throughout the years. Categories and names changed throughout the years, so we wanted to make sure we included columns in the awards CSV that encompassed “film”, “picture”, and “feature”. We started by excluding the columns that were not relevant to the movies and the respective awards.

In the IMdb movie dataset, we excluded columns:

['imdb\_title\_id', ‘country’, 'date\_published', 'description',

'original\_title', 'duration', 'language', 'actors',

'avg\_vote', 'votes', 'budget', 'usa\_gross\_income',

'worlwide\_gross\_income', 'metascore', ‘writer’, ‘genre’,

‘reviews\_from\_users’,‘reviews\_from\_critics].

And included columns:

[‘imdb\_title\_id’, ‘title’, ‘year’, ‘genre’, ‘country’, ‘director’, ‘writer’,

‘Production\_company’]

In the Oscar Awards dataset, we utilized conditional statements to discover keywords related to a movie, such as “picture”, “film”, and “feature”. Our goal was to add the award listings to the extensive IMdb dataset to filter out which movies had won an Oscar.

We also deleted the values for movies that were only nominated for an Oscar but did not win one, to display a clean dataset of only the Oscar winners. We discovered that due to the large datasets, including all the awards throughout history was going to take a substantial amount of time to process and in an effort to reduce processing time we only included “Best Picture”, “Directing”, “Best Motion Picture”, “Cinematography (Black and White) and “Cinematography (Color)” categories.

During the analysis of our dataset we encountered that despite the large IMdb dataset we had, some of the awarded movies from the Oscars dataset were not listed and because no relationship could be established, the rows were dropped. In some instances there were movies that had in the directors column more than one name (tuple). To ensure that there was only one value in each cell, we split the values into separate columns to later append to our final dataframe.

In preparation for the Load step in this entire process called ETL, we generated multiple csv files to be utilized in a separate jupyter notebook file dedicated for that task alone. Some of the files created were based on the structure of the database:

* IMdb dataset cleaned and filtered (source: IMdb dataset).
* Oscars Awards with title ID extracted from IMdb (source: Both).
* Director's name (source: IMdb dataset).
* Oscars Awards categories with generated ID (source: Oscars Awards dataset).
* Awarded with movie title ID and oscar ID (source: Oscars Awards dataset).
* Awarded Directors with title ID and director ID (source: Oscars Awards dataset).

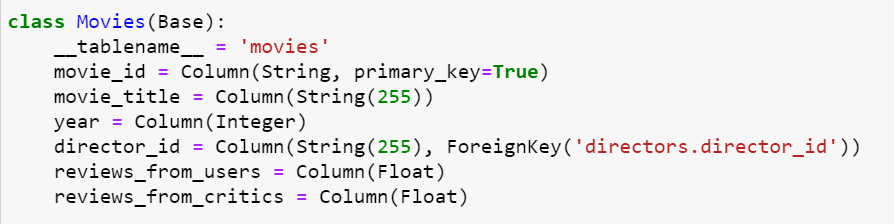
Following the three most common forms of Data Normalization (1NF, 2NF and 3NF), we created an ERD diagram to show the structure and relationship between our two data sources and what values we merged them on. We have included a visual of the ERD diagram shown below:



**LOAD**

We loaded our data into the relational database, PostGRES, after we finished cleaning it up. It was a seamless transition pivoting from Jupyter Notebook to PostGRES using SQLAlchemy. We chose a relational database because it provides the ability to generate meaningful information by joining tables. The ability to join tables gives us the ability to understand the relationships between the data.

Database was manually created and named “movies\_db” and by using classes we created our tables using the following format:



Once we added data into the first table called “Directors” to the database. It was a repetitive process by simply replacing the variable names and CSV files to complete with the addition of the data.