**2017-2019 US Movies Database**

Our objective was to create a database of movies, including main cast and crew info, key box office figures and movie details for films released from 2017-2019 in the US. Our main source of data were three CSV files from [<https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset>](https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset). This included CSV files with data scrapped from IMDb. Any movie that has a 100 user votes was on it and was recent until 11/17/2019

We decided to keep it to three tables. One table was a movie information table, another was cast and crew, and another was a junction table which contained the IMDb movie id and name ids from both, linking both tables.

**Transformation**

**Movies DataFrame**

Some columns were removed to trim down the data. Original\_title', 'year', 'description', 'votes', 'metascore' , 'production\_company', 'reviews\_from\_users', 'reviews\_from\_critics', 'writers', 'actors' columns were all removed for several reasons: 1)We wanted to have the database contain mostly main objective information about the movies, so the review and score columns were removed, leaving only the IMDB user score, 2) To remove redundancy, as the year info is already in the 'date\_published' field, 3) We wanted to have the cast and crew info separate, so ‘writers’ and ‘actors’ were removed, 4) Lastly, some information like production company we didn't think it was important, so that was removed to trim down the table.

Since we were taking movies from 2017 onwards, we did a date filter on date published using >2016-12-31. Now, the date\_published column had only the published year listed for some of the movies, and these were removed in the filtering process as they didn't have complete publishing info. There were not too many of these movies to make an impact.

Other transformations done:

-Did a str.contains script on the country field for "USA" to include moves that had some level of

US production

-The financial columns had the '$' removed to avoid symbols in SQL and also budgets with non-US figures like CAD, CNY, EUR, and GBP replaced with NaN as we wented to keep US figures only.

-renamed column 'worlwide\_gross\_income' to 'worldwide\_gross-income' for proper spelling

**Names DataFrame**

Names file contained list of details of more than 175,000 people but only a portion of them were needed. Those with dates of death of 2016 onwards and with null figures(ie still alive) were kept as these people were likely to have worked on the films in the database.

The birth\_country column was cleaned up to replace countries such as West Germany and USSR with proper names like Germany and Russia.

**Title Principles (Junction) DataFrame**

This contained the primary keys from the two other DataFrames, the name\_ids and title\_ids. These are unique identifiers for the movie titles and the cast and crew from the Names table and links both tables.

Only the ordering and job columns were removed. Ordering was subjective, as it ranked each person's importance to the film (actors, then behend the scenes jobs like directors, writers listed next) didn't serve any meaning and was subjective, as director could be just as important as actor. The job column was a repeat of sorts to the profession column.

The square brackets and apostrophes were removed from characters columns to keep the column clean.

**Challenges and New Knowledge**

-it was difficult on how to go about replaced the foreign currencies from the budget column with nAn if they contained CAD, EUR, CNY, or GBP.

-str.replace and str.rstrip were two items learned in cleaning up columns that came in useful in cleaning up date

-.str.contains was useful in finding substrings.