**2/16/19**

**University of MN Data Visualization and Analytics Boot Camp**

**Cohort 2**

**Team FuzzyWuzzy Project 2: Final Report**

**Members: Anupa , Nandu, Eyamba, Nisha, Abe**

**Abstract:** For this project Team FuzzyWuzzy delved into further exploration of movie/film data from multiple csv files. The original data sets for these files came from a collection of over 45,000 movies aggregated from the International Movie Database and The Movie Database by way of Kaggle.com. From these sources we extracted and transformed data to create new tables that on their own, or when joined allowed further analysis of key film metrics. Each table was then added to a comprehensive database on our local machines that could be used for query and analysis.

**Extraction:**

**Data Sources:** We used the following csv files to create our database.

* Movie\_metadata.csv
* Keywords.csv
* Links.csv
* Credits.csv

**Tables Created:**

* Movies\_metadata: This table contains the following columns/characteristics:
  + Id, movie id, homepage, original title, overview, popularity rating, release date, runtime, tagline, and working title of each film.
* Movies\_finance: This table stored financial data such as budget and revenue numbers for each film. It contains the following:
  + Movie id, imdb id, and revenue for each film.
* Genre: Describes the genre categories each film falls within. Columns in this table include:
  + Id, movie id, genre id
* Genres\_master: Contains a master list of all genres and the specific genre id.
* Movies\_language: Displays the original language in which the movie was filmed. The table is comprised of the following:
  + Id, movie id, movie language, and movie language type.
* Production Company: Displays the production company or companies for each film. Table components consist of the following:
  + Production company id and production company name.
* Production Country/Location: Created to store the shoot location(s) for each movie. Components for this table include the following:
  + Id, movie id, country name, country abbreviation.
* Cast: Created to store actors associated with each film. It consists of the following:
  + Id, movie id, cast Id, movie character, actor id, actor name, gender.
* Crew: Created to store names of the crew members (e.g., director, producer) for each film
  + Contains: id, movie id, department, job title, crew name, gender.
* Keywords: This table was built to store keywords associated with each film.
  + Components include id, movie id, Keyword id, and keyword name.
  + For example, if one wanted to do a query to display movies associated with car chases.
* Links: houses the link id’s for both IMDB and TMDB, and consists of the movie id, imdb id, and tmdb id.

**Transformation**

Quite a bit of cleaning and transformation was required for these data sets. For example, the movies\_metdata.csv and credit.csv files had several columns that contained what seemed to be a list of dictionaries containing a vast amount of information, not all of which we needed.

**Example 1 from Credits.csv:** [{'credit\_id': '52fe4284c3a36847f8024f49', 'department': 'Directing', 'gender': 2, 'id': 7879, 'job': 'Director', 'name': 'John Lasseter', 'profile\_path': '/7EdqiNbr4FRjIhKHyPPdFfEEEFG.jpg'}, {'credit\_id': '52fe4284c3a36847f8024f4f', 'department': 'Writing', 'gender': 2, 'id': 12891, 'job': 'Screenplay', 'name': 'Joss Whedon', 'profile\_path': '/dTiVsuaTVTeGmvkhcyJvKp2A5kr.jpg'}, {'credit\_id': '52fe4284c3a36847f8024f55', 'department': 'Writing', 'gender': 2, 'id': 7, 'job': 'Screenplay', 'name': 'Andrew Stanton', 'profile\_path': '/pvQWsu0qc8JFQhMVJkTHuexUAa1.jpg'}, {'credit\_id': '52fe4284c3a36847f8024f5b', 'department': 'Writing', 'gender': 2, 'id': 12892, 'job': 'Screenplay', 'name': 'Joel Cohen', 'profile\_path': '/…

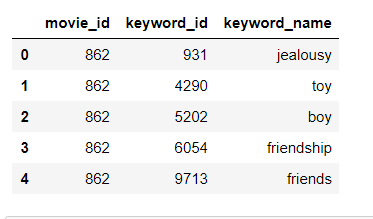
**Example 2 from Keywords.csv:** [{'id': 931, 'name': 'jealousy'}, {'id': 4290, 'name': 'toy'}, {'id': 5202, 'name': 'boy'}, {'id': 6054, 'name': 'friendship'}, {'id': 9713, 'name': 'friends'}, {'id': 9823, 'name': 'rivalry'}, {'id': 165503, 'name': 'boy next door'}, {'id': 170722, 'name': 'new toy'}, {'id': 187065, 'name': 'toy comes to life'}]..

To transforming this data necessitated the group writing code that would evaluate each row of the table and within this column extract and display only the information we wanted, such as character name, actor name, or job title. The result after transformation is below.

**Example Code:**



**Result**



**Load**

Once transformed, we were able to load our newly created data frames to the database by inserting them into their corresponding tables created in MySQL Workbench. We chose to do it this way so that we could run queries that would allow us to explore various trends, such as highest grossing films, directors and genres; most popular directors, films, and genres; most popular years for certain types of movies to be made, etc…

All files, tables, and SQL documents have been uploaded to the FuzzyWuzzy repository for review.