Falconine Falcon’s **“Movies Database ETL”**

Final Report

USC Viterbi Data Analytics Bootcamp – Project #2 ETL  
August 24, 2019

# Introduction

“Movies Database ETL” is a team project by the Falconine Falcons to demonstrate the functions of data extraction, transformation, and loading of data as part of a typical ETL project.

The project serves three main functions:

* *Extraction* of data from multiple sources for use in the final database.
* Manipulation of the data to the project’s requirements as part of data *transformation*.
* *Loading* of the transformed data into the final production database

The focus of this ETL project is 2019 theatrical movies. Data for movies can be found everywhere, but often important information can be missing from a single data source, requiring the aggregation of data from multiple sources.

In this project data is sourced from a CSV file and API call, manipulated in Python with Pandas, and stored in both MongoDB and PostgreSQL during the staging and production storage of the data.

# Extract

Two data sources contribute to the Movies Database ETL project. One is a superset of movie data stored in a CSV file and the other data source is provided by a credentialed API call to the Internet Movie Database (imdb.com).

all\_2019\_movies CSV: The CSV provides the archive, or historic data for the database. The superset of data contains over a dozen movie attributes of which only two, ‘Movie Title’ and ‘Release Year’, are required for the ETL project.

IMDB API: A tokenized request is made to the IMDB API to retrieve the additional data required for the Movie Database ETL. The request passes Movie Title and Release Year from the all\_2019\_movies CSV to perform a data call on the exact title required.

# Transform

Transformations needed to be performed on the data for both conducting the analysis and to meet the requirements of the target database.

all\_2019\_movies CSV: This dataset contains data not required in the final database. From this dataset ‘Movie Title’, ‘Release Year’, and the title ID, labeled ‘Const’ were ***filtered***. Further, the data in the ‘Const’ dataset was incorrectly formatted with extraneous characters. This data was ***cleaned*** in Python removing the extraneous characters.

IMDB API, MongoDB: The cleaned CSV data was used in making the API call to IMDB to ensure an exact tile match. The payload returned from the API call was stored in a MongoDB for staging.

# Load

The final Movies Database has been loaded and is stored as a relational database using PostgreSQL.

MongoDB, PostgreSQL: The data staged in MongoDB (named ‘movies\_db’) is read using MongoDB and then written into the final database in PostgreSQL (called ‘movies’).

# Conclusion

By executing the functions of a standard ETL project, data extraction, transformation and loading, we were able to present final production data in the “Movies Database ETL”.

We have learned how executing this ETL project, or variants of this project, could add a lot of value to a business, institution or other project related activity.