ETL Project Report

By: Alexander Jutila, Karthik Puttichanda, Lekshmi Nair

**Cinema World**

Many theaters worldwide have closed indefinitely due to the pandemic leaving the movie lovers missing the theater experience. Most see this as an opening for streaming services. But it has also brought an appreciation for the pleasures of going to the movies and their role in social life.

For our love of movies, we have compiled a list of Movie datasets based on ranking, movie name, budget, cost, revenue and rating by drawing on information from different sources. We have performed an ETL on Movie data to narrate the history and specifics of the world movie, list of movies on Netflix and the plot of the Cinema from Wikipedia.

**Extract**

We used four different datasets from the platform Kaggle to collect the Movie data. The data in the four files included the following information:

* Movie Data from IMDB and TMDB
* Movies on Netflix
* Plot of Movies from Wikipedia

The following sources for our datasets used:

<https://www.kaggle.com/rounakbanik/the-movies-dataset> (TMDB data)

<https://www.kaggle.com/shivamb/netflix-shows> (Netflix data)

<https://www.kaggle.com/jrobischon/wikipedia-movie-plots> (Wiki data)

<https://www.kaggle.com/danielgrijalvas/movies> (IMDB data)

**Jupyter Notebook:**

In order to extract the data we used Pandas functions in Jupyter Notebook to load all four CSV files.

A screenshot of a social media post

Description automatically generated

**Transform**

In order to transform the Movie data, we performed the following:

* Reviewed the files and transformed into data frames including the relevant columns.
* Renamed the column headers.
* Cleaned the data by dropping duplicates and null data, setting the index, and removing irrelevant data such as shows from the Netflix data.
* Used Pandas datetime function to reformat the date column and to remove records with invalid date.

The fields of interest include the following:

IMDB

* Movie Title
* Budget
* Country
* Company
* Director
* Star
* Rating
* IMDB Score and Vote count

TMDB

* Movie Title
* Original Language
* Runtime
* Release date
* TMDB Score and Vote count

Netflix

* Movie Title
* Date Added
* Listed in

Wiki

* Movie title
* Plot

**Jupyter Notebook:**

A screenshot of a social media post

Description automatically generated

**Load**

In order to load the Movie data, we performed the following:

* We used the quick database website to create the initial table schema that got loaded into the Postgres database that generated the set of tables.
* In order to normalize the data, we created sub tables for IMDB and TMDB Vote count and Scores. Which had foreign keys for Movie title to the respective IMDB and TMDB tables.
* After we pulled in the CSV files and loaded them into the data frames, we did an initial connection to the Postgres database using PG admin to store our original clean data sets.

**Jupyter Notebook:**

A screenshot of a social media post

Description automatically generated