**Netflix ETL - Final Report**

**Project #2**

Date: October 12, 2021

Team Sociotracker:

Angela Angulo

Bonnie Bailey

Joseph March

Marisa Kiger

**Extraction:**

We used four datasets from Kaggle and one dataset from Yahoo Finance (cited below). All of the data sources utilized were in CSV format. Timeframes included in the datasets varied, ranging from 2002-2021 overall.

**Kaggle**

Netflix TV Series Dataset:

<https://www.kaggle.com/harshitshankhdhar/netflix-and-amazon-prime-tv-series-dataset>

* *tv\_shows\_data.csv* (columns: Title, Genre, Premiere, No\_of\_Seasons, No\_of\_Episodes)

Netflix Subscribers and Revenue by Country :

<https://www.kaggle.com/pariaagharabi/netflix2020>

* *DataNetflixRevenue2020\_V2.csv* (columns: Area, Years, Revenue)
* *DataNetflixSubscriber2020\_V2.csv* (columns: Area, Years, Revenue)

Netflix Original Films & IMDB Scores:

<https://www.kaggle.com/luiscorter/netflix-original-films-imdb-scores>

* *NetflixOriginals.csv* (columns: Title, Genre, Premier, Runtime, IMDB Score, Language)

**Yahoo Finance**

Netflix Inc. Stock Historical Prices & Data:

<https://finance.yahoo.com/quote/NFLX/history?period1=1033948800&period2=1633564800&interval=1d&filter=history&frequency=1d&includeAdjustedClose=true>

* *Yahoo Finance NFLX.csv* (columns: Date, Open, High, Low, Close, Adj Close, Volume)

**Transform:**

**Data Normalization**

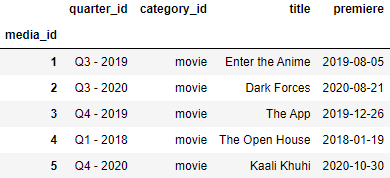
Once we gathered our datasets, our first step was to decide how the data should be organized in the database. We decided to create a relational database in pgAdmin, and built an Entity Relationship Diagram in second normalization form using QuickDBD. We reorganized the original data in order to eliminate fields with multiple values and reduce redundancies within each of the tables. We also established appropriate primary and foreign keys for each of the entities in the database.

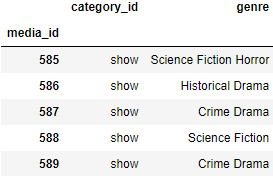
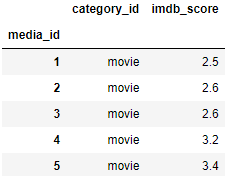
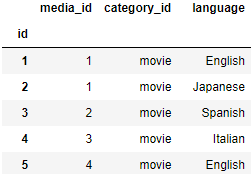
Text

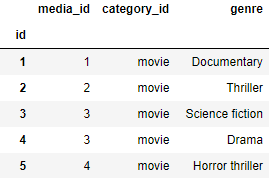
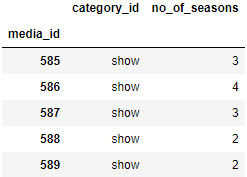
Description automatically generated with low confidence

**Data Cleaning**

After importing all of the CSV files into their own tables, our first steps in cleaning the tables were to determine which data was not relevant and what needed to be added. We removed the Runtime column from the Netflix Original Films & IMDB Scores data table and the No\_of\_Seasons column from the Netflix TV Series Dataset. Then we merged the two tables together so we could give all of the movies and TV series their own designated Media\_ID. We also calculated a Quarter table from the premiere date so the Quarter a movie or TV series was released in could be compared to the other tables. Then we broke the table up into several tables to conform to 2nd normalization. This became the following tables: Netflix\_Movie\_Show\_List, Netflix\_Movies\_IMDB, Netflix\_Movies\_Lang, Netflix\_Movies\_Genre, Netflix\_Shows\_Seasons, and Netflix\_Shows\_Genre tables.

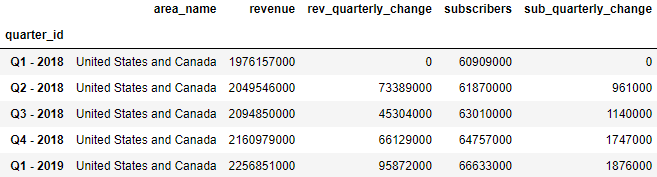


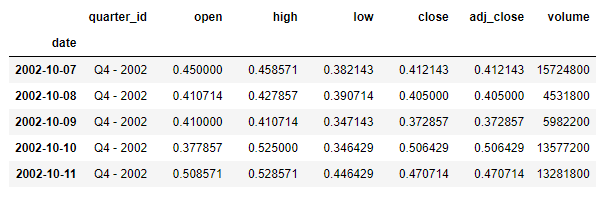
 

We then created a Quarters table which included a Quarter\_ID for every quarter and year. This serves as a table that can be used to combine our other tables which rely on Quarter\_ID to show the time periods represented.

Next, we combined the Netflix Subscribers and Revenue by Country tables and calculated quarterly change columns by revenue and number of subscribers. These were further split into four tables, by area, to create the Netflix\_Rev\_Sub\_USC, Netflix\_Rev\_Sub\_EMEA, Netflix\_Rev\_Sub\_LA, and Netflix\_Rev\_Sub\_AP tables.



Last, we modified the Netflix Stock Price (All Time) Dataset to create the Netflix\_Stock\_Prices table by adding a Quarter column and setting the date column as the index.



After cleaning the data, we used SQLAlchemy to import all of the tables into those created by our schema in PostgreSQL. As part of this process we had to change all of our columns to lowercase first so that they matched SQLalchemy’s table conversion.

Some additional cleanup we performed included the following:

* Changed order of columns to be consistent across all tables.
* Converted format of Quarter\_IDs to be consistent across all tables.
* Formatted date/premiere rows to datetime and the same date format.
* Split rows with combined data cells into single data rows.
* Exported a series of CSV files of our cleaned dataframes for storage purposes.

**Load:**

We wrote a table schemata based upon the sql code exported from QuickDBD. Once the tables were created in pgAdmin, we imported our cleaned data using SQLAlchemy. With our final database, we performed three queries to display joined tables.

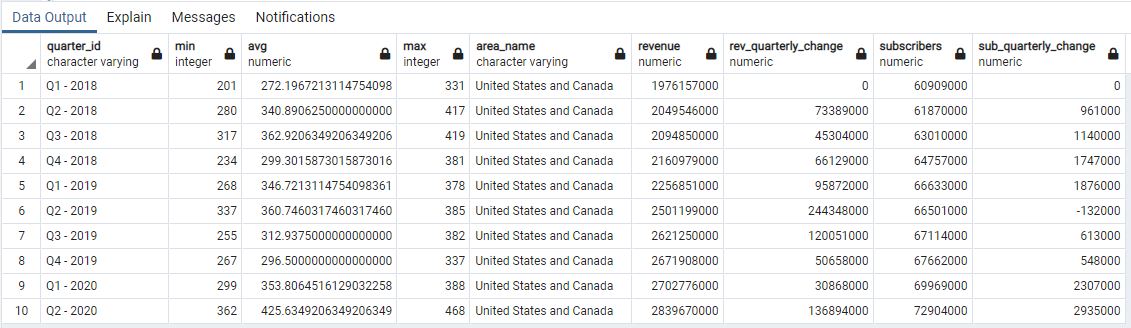
**Query 1** - In order to display a summary of all the information for each Netflix original movie and show, we used Join and Union statements to present the following information within the same table:

* The Netflix\_Movie\_Show\_List columns “Media\_ID”, “Quarter\_ID”, “Category\_ID”, “Title”, and “Premier”
* The Netflix\_Movies\_Lang column “Language”
* The Netflix\_Movies\_IMDB column “IMDB\_Score”
* The Netflix\_Shows\_Seasons column “No\_of\_Seasons”
* The “Genre” columns from Netflix\_Movies\_Genre and Netflix\_Shows\_Genre



**Query 2** - To display a financial comparison between stock prices, revenue, and number of subscribers for the United States and Canada per quarter, we used Join statements, as well as a subquery to calculate the minimum, maximum, and average closing stock prices grouped by quarter, to present the following information within the same table:

* The Quarter column “Quarter\_ID”
* The calculated Min, Avg, and Max of the Netflix\_Stock\_Prices “Close” column
* The Neflix\_Rev\_Sub\_USC columns “Area\_Name”, “Revenue”, “Rev\_Quarterly\_Change”, “Subscribers”, “Sub\_Quarterly\_Change”



**Query 3** - Our final query brought together the data from both of the previous queries to display the media and financial information in one table.



**Summary:**

The datasets we created can be used to compare financial outcomes to media released on a quarterly basis for Netflix. Data can be used to compare subscriber and revenue increases by geographical area, as well as stock prices by date or quarter. We also maintained the IMBD, genre, and language data so that comparisons can be formed based on these unique areas. Our SQL joined tables allow for a user to recognize relationships between differing elements of financial outcomes and media releases.