Technical Report

1. Why we selected our data sets?
   1. We found a dataset of Oscars Best Picture Winners/Nominees
   2. To see where we could watch these films, we found a dataset of all films available across the major streaming platforms
   3. Manually double checked to make sure there were overlapping years across data sets
   4. Since these were finite and relatable datasets that could be connected, we chose to work in Postgres SQL
2. Why/How we chose to clean the data sets?
   1. We thought about what a viewer would want to see about a movie they were about to watch (Title, IMDB Rating, Genre, Year of Release)
   2. Based on that, we isolated columns that viewers would want from both data sets
   3. To allow for comparison of datasets in PG Admin, we renamed several columns to match across both datasets
   4. We noticed that the Streaming Platforms columns (Netflix, Hulu, Prime Video, Disney+) were in a certain format (1=Yes, 0=No)
   5. To make reading the data in these columns easier, we changed 1 to Yes and 0 to No
3. Why/How we linked the data sets?
   1. We realized that it would be important to have a Unique ID for each movie name for PG Admin
   2. We looked at each data set and first tried to think about how to use indexes to create Unique ID
   3. The indexes across both datasets could not correspond because multiple films were present in only 1 dataset
   4. We decided to create a new data frame of film names compiled from both datasets
   5. We did this and used the index from the new data frame to serve as the Unique ID across both original data frame
   6. However, when we investigated the new data frame there were duplicate movie names present
   7. Upon further investigation, we found that some duplicates were true duplicates and others were cause by a film having multiple years of release
   8. Went back and revised the new data frame to include Name and Year of Release for each movie from each dataset
   9. We then inserted this Unique Id back into the original 2 datasets as a new column
4. What did we do in Postgres SQL
   1. Since we created the tables in Jupyter Notebook, we checked that they were correctly pulled over
   2. We tried to set the primary key as Unique ID from the master films lists with it being a Foreign Key in both Oscars and Streaming tables
   3. Found an error that said that one Unique ID was duplicated
   4. Returned to Jupyter Notebook, reset the index on the master list to update the Unique ID to no longer have a duplicate
   5. We reloaded the data in Postgres SQL and were able to set the Primary Key and Foreign Key as planned