**Project 2 - ETL - Group 6**

**Members Ben, Brendan, Kinlay, Sam**

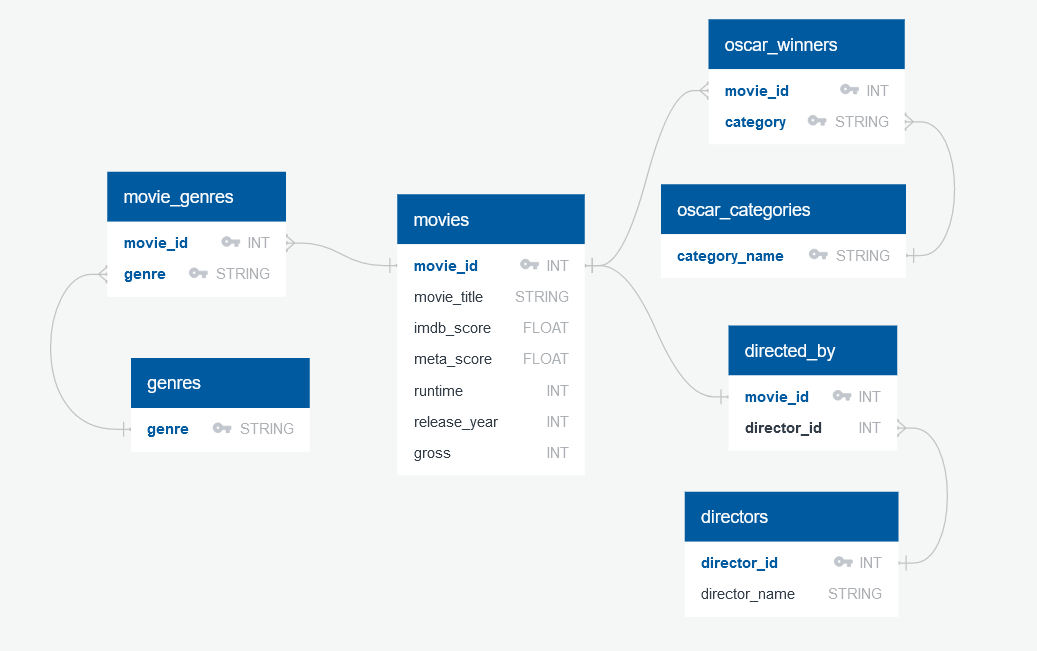
**Objective -** To perform ETL on pulled data of the “IMDB top 1000 rated movies”, combined with data on the “academy awards winners” displayed in a relational database. This will include a table showing which movies from the IMDB top 1000 rated have won an Academy award and in which Category.

**Data Sets** - The data sets used were from the following web addresses

<https://www.kaggle.com/datasets/harshitshankhdhar/imdb-dataset-of-top-1000-movies-and-tv-shows>

<https://www.kaggle.com/datasets/theacademy/academy-awards>

**Process –** Once the data was sourced, we created an Entity Relationship Diagram for the data.

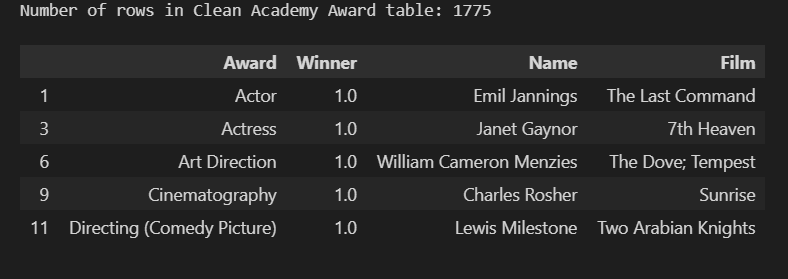
****

**ETL**

**Extract -** The files for the “IMDB top 1000 rated movies”, and the “academy awards winners” were downloaded as csv files and then read into pandas’ data frames.

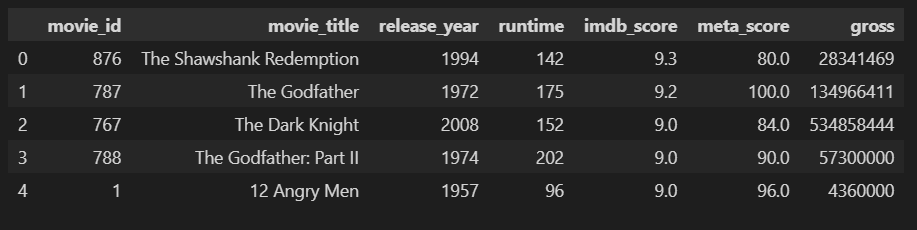
**Transform -** The IMDBdata frame was first tested to see what data types were associated with each column, and were altered to locate errors in the data along with updating the data type to suit the column type. In cases of minimal errors, we adjusted the data as this was not a large data set.

The academy awards data frame was cleaned by first selecting only the useful columns before dropping all NAN value rows to remove missing data and all nominees(non-winners), as we only wanted winners. Further, duplicates were dropped to remove multiple winners for the same award as we only want 1 winner to show per award per movie.



The IMDB table was reduced, so only the columns required from our ‘movie’s table’ in our ERD plan will show, and Unique Identifiers were also added for ‘Movie Titles’

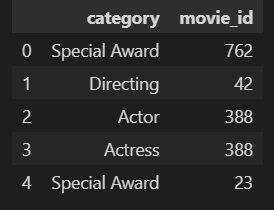
The ‘Movies’ Table was created by only using the appropriate columns from the cleaned and updated IMDB data to match our ERD.



By finding the unique values in the director’s column we were able to create our director’s table which was then appropriately renamed to reflect this and only the correct columns selected.

To create the ‘directed by’ table the movies table was again used and reduced to the required columns and were renamed.

In order to create the ‘Oscar winners’ table, we used 2 left merges, as when inspecting the data, we realized that in the Academy awards data movie titles were sometimes put in the ‘name’ column instead of the film column where they should have been. Once the 2 left merges were completed an outer merge was used along with column filtering to only display data with the Movies that won awards along with their category that they won in.



The function “.unique” was used to reduce the column to only show the Oscar category once each to create a data frame purely for Oscar categories.

The ‘Movie Genres’ table required each film title to correspond only to one genre in the corresponding row. If a film title had more than one genre, duplications of the film title were created for each unique genre per film. The Drama genre was also cleaned up as it picked up an empty space before Drama e.g.‘ Drama’ instead of ‘Drama’ which was stripped to avoid double-ups in of genre column.

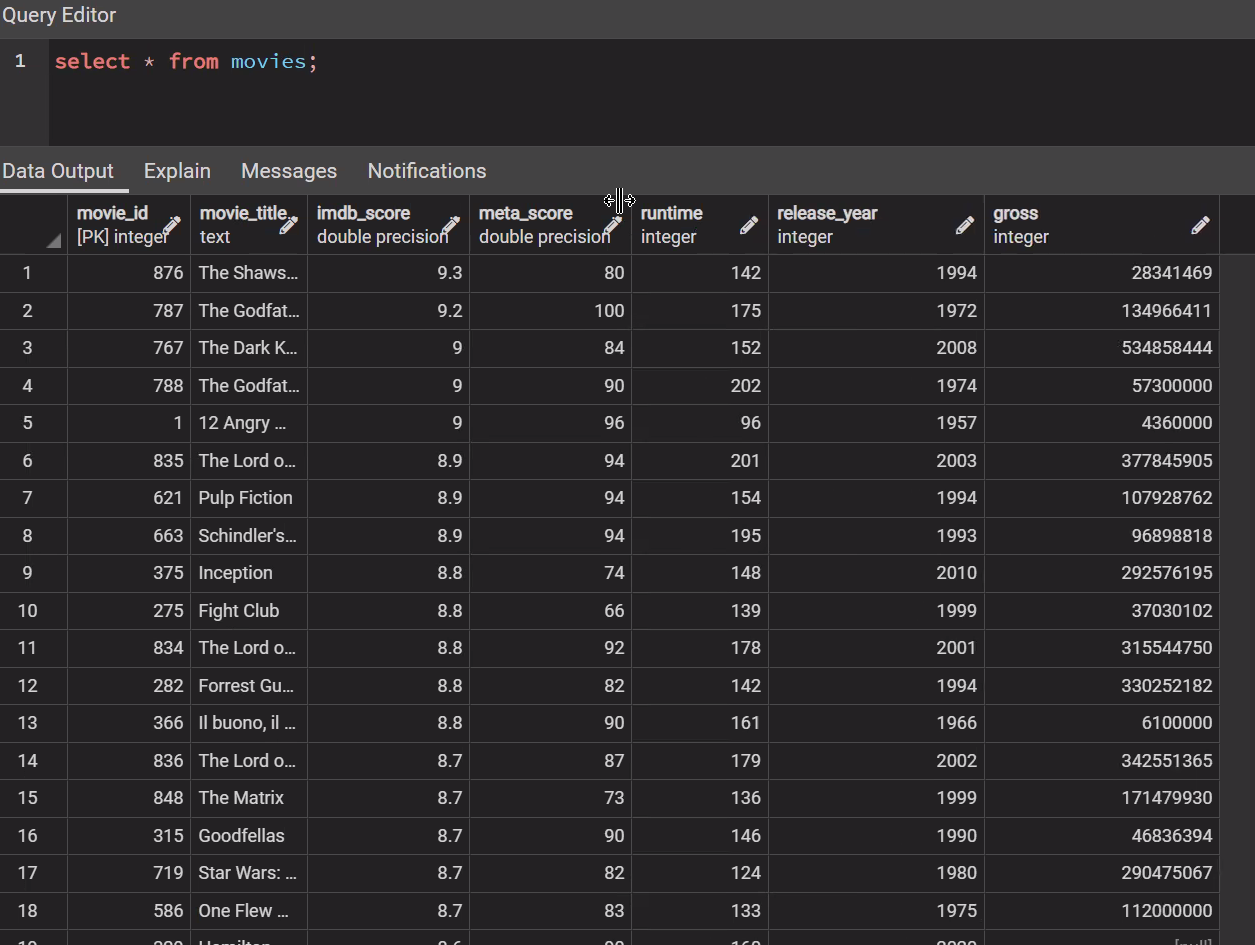


The ‘Genres’ Table was created by using unique to create a single column table.

All Tables were saved as variables ready to be loaded.

**Load -** InPostgres Tables were created to reflect the tables from our ERD. Once the tables were created, we added the code into our python file in order to load the files into our Postgres Database.





The above screenshot is an example of our Postgres database for the movies table.

**Reason for Choice -** We chose these data sets in order to create a structured relational database for the IMDB movies combined with Academy Award data, so that we can breakdown the data into multiple levels to create a normalized database. As this data is separated into multiple tables based on categories it makes it easier to update/alter the data as opposed to having it all in one big table. Setting up the database in this format also allows for greater Flexibility for adjustments to its structure going forward.