**Chosen Data Sets:**

For our project, we will be performing an analysis of Netflix content. We found our two data sets from Kaggle, listed below:

Netflix Shows- <https://www.kaggle.com/chasewillden/netflix-shows?select=Netflix+Shows.csv>

Netflix Titles- <https://www.kaggle.com/shivamb/netflix-shows>

**Introduction:**

Both datasets are CSV files. The general transformation for both of these datasets will require dropping unnecessary columns, removing NULL values, sorting tables, checking and removing duplicates, and fixing the layout of some rows. The final result includes two separate SQL tables but the two tables do not have a Foreign Key. We chose SQL because we thought it would provide an easy way of viewing and analyzing the data for an analyst.

**For Netflix Shows:**

The goal for Netflix Shows is to create a database that is prepared to perform analysis on the relationship between Age Rating and User Rating.

We had to begin by removing the unneeded columns. Since we wanted to provide a dataset ready to analyze the relationship between Age Rating and User Rating, we removed 'Rating Description', 'Release Year', and 'User Rating Size'.

That left us with columns 'Title', 'Rating', 'Rating\_Level', 'User Rating Score'. These columns were intentionally left in the dataset because they give the analyst the ability to perform an analysis of Age Rating and User Rating. However, the column 'Rating\_Level' gives the analyst the flexibility to go into more detail, such as seeing what specific kinds of content Netflix was adding for each 'Rating'.

After we dropped unnecessary columns, it was important to drop the NULL values from both the 'Title' and 'User Rating Score'. There should not be any NULLs in these two columns because that will be the relationship between the title and its popularity.

After the NULLs were dropped, we needed to rename the columns to better reflect the type of data inside each column. Columns 'Title' and 'Rating\_Level' could remain the same since they best represented the information inside each column. However, 'Rating' was better represented by 'Age\_Rating' to avoid confusion from the title's popularity rating. 'User Rating Score' was better represented by 'User\_Rating' to represent the score users gave to the title. After that was completed, we replaced any empty NULLs with empty string.

After the empty NULLs were replaced with empty string, we needed to make sure that the data type for each column is correct. So we labeled "title" as string, "age\_rating" as string, "rating\_level" as string, and "user\_rating" as integer.

We needed to check for duplicates once again for this data set. For this data set, we chose to check for duplicates by 'Title'. We got many duplicates under 'Title'. We decided to double-check this result by using two of the listed duplicates, "Grey's Anatomy", and "Prison Break". Once we confirmed that these were indeed duplicates, we dropped them and reset the index according to "Title". Then we established a database connection to SQL.

**For Netflix Titles:**

The focus of Netflix Titles will be to create a database that is ready to analyze the distribution of the different Age Ratings across Netflix.

We had to begin by removing the unneeded columns. Since we wanted to provide a dataset ready to analyze the distribution of Age Ratings across Netflix, we removed 'Duration', 'Cast', 'Director', 'Date\_added', 'Release\_year', 'Description', 'Country'. That left us with columns 'Show\_id', 'Type', 'Title', 'Rating', and 'Listed\_in'. These columns were intentionally left in the dataset because they give the analyst the ability to dig a little deeper into the distribution of age ratings across Netflix. For example, an analyst would be able to compare the distribution of age ratings against type, which could determine if tv shows or movies have more of one kind of age rating.

After we dropped unnecessary columns, it was important to check if the remaining columns had any NULL values. The only NULL values present were in the 'Rating' column. We dropped the NULL from the 'Rating' category because the whole point of this dataset is to measure the distribution of age ratings across Netflix. If the 'Rating' is NULL, that data can not be used in the analysis. Once the NULLs were dropped, we did a check to make sure there were no remaining NULL values. For Netflix Titles, there were no remaining NULLs.

Now that the data has been somewhat cleaned of issues, the columns needed to be renamed to be more representative of the data. 'Show\_id', 'Type' and 'Title' could remain as they were since they properly reflected the information they contained. However 'Rating' needed to be changed to 'Age\_rating' to better represent that this column contained age ratings, not user ratings. 'Listed\_in' needed to be changed to 'Category', because this column listed all the genres the title was listed under. After that was completed, we replaced any empty NULLs with empty string.

After the empty NULLs were replaced with empty string, we needed to make sure that the data type for each column is correct. So we labeled 'Show\_id' as an integer, 'Type' as string, 'Title' as string, 'Age\_rating' as string, and 'Category' as string.

We needed to check for duplicates once again for this data set. We first tried to do it by 'Title'. We got many duplicates under 'Title'. However, the only part that was repeated was the title. The type, category, and show\_id were all different. For example, the title 'Lavender' was marked as a duplicate, however there was a tv show and a movie called 'Lavender' and they were unrelated from one another. Therefore, we decided to check duplicates by 'show\_id'. Once we learned there were no duplicates left in this dataset, we reset the index by 'Show\_id'. Then we established a database connection to SQL.