Team 11:

Jose Martinez Carvajal

Muhammad Habib

We plan on using YouTube Statistical data from Kaggle and also ---. We will be extracting data from CSVs as well as JSON files. We will clean it up, link tables using primary keys and load it into a PG Admin database

Our goal is that this resulting database could be used to swiftly find and visualize the top trending YouTube videos for the week, month or year based on the number of comments, likes and views.

Visualize what determines the top trending videos of the year based on views, comments and likes.

**Report:**

Category ID can be joined and see what categories are the most popular and which ones get the most views

The sources of data that you will extract from. YouTube statistics from Kaggle. We extracted multiple datasets from Kaggle. USA channel statistics which gave us the subscriber count and view count for the specific video; USA video interactions, USA trending video interactions gave data about the trending date, publish date and time and all the data about likes, dislikes and comments which would help determine what are the top trending videos and also when they were published to see if there was any correlation between times of publishing to determine its viralness

*\* The type of transformation needed for this data (cleaning, joining, filtering, aggregating, etc).*

We brought in all the columns which were in the CSV into our Data Frame after parsing out the Snippet data and extracting the channel\_id, assignable\_id and title from it. We had to rename a few columns to make it easer to load in Postgres. We brought in the columns which we needed to make our determination of top trending videos

We imported pandas libraries to help run the code in Jupyter notebook. After converting into a data frame and cleaning up the data by in the csv files. There is a category\_id. In the json file you can find what category name the category\_id refers to make it easier to extract it.

*with open('./youtube\_files/US\_category\_id.json', 'r') as json\_data:*

*data = json.load(json\_data)*

*for category in data['items']:*

*category\_name[category['id']] = category['snippet']['title']*

*\* The type of final production database to load the data into (relational or non-relational).*

Relational since we will be linking the category IDS in 2 separate tables which will determine which category of videos become top trending.

*\* The final tables or collections that will be used in the production database.*

We created the database and tables in Postgres SQL

**Uses for this table:**

Find the top trending video, not necessarily the most viewed video

What video is gaining traction and going viral?

What video is being talked about the most on the internet?