ETL PROJECT REPORT

**Topic:**

**Contributors:**

Trending YouTube Videos

Firdaus Mehta

Natalie Manahan

Sreemanth Gurram

Table of Contents

Objective………………………………………………………………………………………………3

Problem Statement…………………………………………………………………………………….3

Extract Transform Load (ETL) Diagram……………………………………………………………...3

Data Clean-up and Analysis

Step 1 : Extraction……………………………………...……………………………………...3

Step 2 : Transforming………………………………………………………………………….4

Step 3: Loading………………………………………………………………………………..4

Resources……………………………………………………………………………………………...4

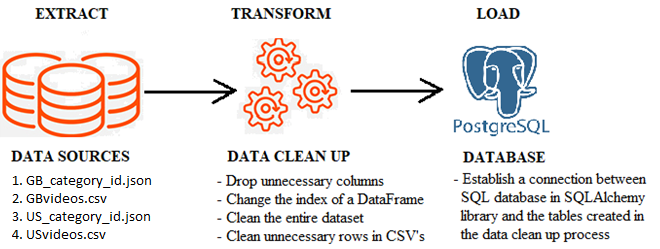
Objective

The objective of this report is to provide a detailed analysis of the ETL (Extract-Transform-Load) process by interpreting a dataset of Trending YouTube videos, obtained from Kaggle, cleaning the dataset in the desired form and loading it into a database for storage.

Problem Statement

Observe trending YouTube videos across the United States and Great Britain along with the dataset of various categories they belong to. Perform an ETL on the datasets that can be uploaded into production to determine which categories or videos are most or least popular, depending the on number of views, likes, dislikes, and comments.

ETL Diagram



Data Clean-up and Analysis

**Step 1: Extraction**

The dataset is a daily record of the top trending YouTube videos. Data is included for the CA, DE, FR, GB, IN, JP, KR, MX, RU, and US regions. The focus of this report will be on the US and GB regions. Each region’s data is in a separate file and includes the video title, channel title, publish time, tags, views, likes and dislikes, description, and comment count. The data also includes a category\_id field, which varies between regions and the categories for a specific video are in the associated JSON for that region. The data has been downloaded from the public platform Kaggle[5]. The files used for extraction were originally formatted as .csv and .json, which are then read into a Pandas data frame in Jupyter notebook.

**Step 2: Transforming**

* First, read the GBvideos.csv[2] and USvideos.csv[4] into Pandas data frames; videoGB\_df and videoUS\_df, using read\_csv.
* Drop the columns that are not relevant from the data frames by selecting the columns. The columns to keep and analyse are video\_id, title, category\_id, publish\_time, views, likes, dislikes, comment\_count into a new\_videoGB\_df and new\_videoUS\_df.
* In the new data frames, rename the columns publish\_time and comment\_count to ‘publish\_datetime’ and ‘comments’ respectively.
* Next, drop the duplicated video titles drop\_duplicates().
* Read the US\_category\_id.json[3] and GB\_category\_id.json[1] into a Pandas data frame, US\_ category\_df and GB\_category\_df, using read\_json().
* Extract the category\_id and category\_title from the item column of the category\_df, because of the way the JSON files are structured.
* From category\_df the item column is stored in a list format. Iterate over the list to extract and store the id and title in separate lists which are used in defining a new data frame, new\_category\_df.
* Merge the new\_videoGB\_df and new\_videoUS\_df separately with the new\_category\_df on category\_id and store them into a new data frame, mergedGB\_df and mergedUS\_df. A country column was added to each, storing “GB” and “US”.

**Step 3: Loading**

Use a relational d relational database, such as Postgres SQL to load the data. Then establish a connection between SQL database youtube\_video\_db using create\_engine() in the SQLAlchemy library. Load the data frames in the tables created; video\_GB, video\_US, category, video\_category\_GB, and video\_category\_US to be used for production.

Resources

1. <https://www.kaggle.com/datasnaek/youtube-new?select=GB_category_id.json>
2. <https://www.kaggle.com/datasnaek/youtube-new?select=GBvideos.csv>
3. <https://www.kaggle.com/datasnaek/youtube-new?select=US_category_id.json>
4. <https://www.kaggle.com/datasnaek/youtube-new?select=USvideos.csv>
5. <https://www.kaggle.com/datasnaek/youtube-new>