**Project ETL**

Team 4: Joe Adams, Michelle Lucio, Ben Farniok, and Jessica Schmitz.

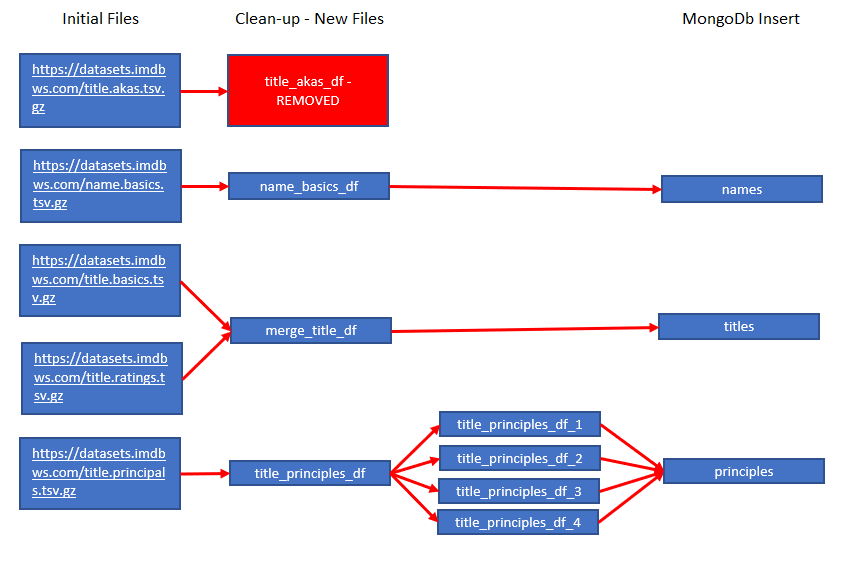
**SUMMARY:**

The project will deliver a database containing IMBD movie data. This data ranges from ratings, cast and crew information, year released, and much more. We began with six data sets, however we determined that one of the six data sets wasn’t useful due to duplicative data and lots of unnecessary details, so dropped it from our project (title.akas.tsv).

We then started to clean up the remaining 5 dataframes. When looking through our different dataframes, we found that the titles\_basics\_df included inappropriate content (adult films), so we removed it from our dataframe. In the remaining dataframes we began renaming column to be consistent across our dataframes, removing columns that held content we didn’t find useful, removing null values, and last merging two dataframe.

At first we ended with three dataframes from the original six that we would insert into MongoDB. However, after attempting to insert the “principle” dataframe, we found that it was too large of a dataframe, as it would error out and/or crash before it could be inserted fully. To solve for this issue, we split the data into 4 separate dataframes, ending with a total of 6 dataframes that we would insert into MongoDB. We decided on using MongoDB rather than PostgreSQL due to some of the columns containing lists of data that we believed were important to contain in our final database.

Here is a visual representation of how we completed the project:



**INITIAL DATA SETS**

Here are links to the original IMDB data sets. **Please note**: We did not include these in our repositories due to the large amount of data they contain. You will need to download them into a “Resources” folder at the same level as the repo folder, in order to get the script to run.

<https://datasets.imdbws.com/title.akas.tsv.gz>

<https://datasets.imdbws.com/name.basics.tsv.gz>

<https://datasets.imdbws.com/title.basics.tsv.gz>

<https://datasets.imdbws.com/title.ratings.tsv.gz>

<https://datasets.imdbws.com/title.principals.tsv.gz>

**CLEANING THE DATA**

* <https://datasets.imdbws.com/title.akas.tsv.gz>
  + Decided that this data wasn’t useful and ended up not using it
* <https://datasets.imdbws.com/title.basics.tsv.gz>
  + Removed inappropriate content (adult films “isAdult” column indicates this content)
    - Once adult films are removed from the database, we then removed the column “isAdult” as it wasn’t necessary since the remaining values are all not adult content
  + Renamed columns
    - ttconst to titleID
    - titleType to Type
  + Replace \N to N/A
  + Final data frame title: title\_basics\_clean
* <https://datasets.imdbws.com/title.principals.tsv.gz>
  + Removed columns
    - Ordering
    - job
  + Renamed columns
    - ttconst to titleID
    - nconst to nameID
  + Replace \N to N/A
  + After attempting to insert this database into MongoDB, it would error out due to the size of the data set. To cut this data down to a more manageable dataframe.
  + Final data frame title: title\_principals\_df\_1, title\_principals\_df\_2, title\_principals\_df\_3, andtitle\_principals\_df\_4
* <https://datasets.imdbws.com/title.ratings.tsv.gz>
  + Renamed columns
    - ttconst to titleID
  + Final data frame title: title\_ratings\_df
* <https://datasets.imdbws.com/name.basics.tsv.gz>
  + Removed columns:
    - birth year
    - death year
  + Replace
    - Replaced \N with N/A then dropped N/A values in “assocTitleID” column
  + Renamed columns
    - nconst to nameID
    - knowforTitles to assocTitleID
  + Final data frame title: name\_basics\_df
* Merged the separate data
  + Created new data set by merging title\_basics\_clean and title\_ratings\_df
  + Final data frame title: merge\_title\_df
* Final list of data frames:
  + merge\_title\_df
  + name\_basics\_df
  + title\_principals\_df\_1
  + title\_principals\_df\_2
  + title\_principals\_df\_3
  + title\_principals\_df\_4

**CREATING THE DATA BASE ON MONGODB**

* Set up MongoDB connection
* Used Pandas data frame to insert to MongoDB for each data frame:
  + merge\_title\_df as titles – took approximately 3 minutes
  + name\_basics\_df as names – took approximately 5 minutes
  + title\_principals\_df\_1 as principals – took approximately 5 minutes
  + title\_principals\_df\_2 as principals – took approximately 5 minutes
  + title\_principals\_df\_3 as principals – took approximately 5 minutes
  + title\_principals\_df\_4 as principals – took approximately 5 minutes