**ETL Project Overview**

Extract:

Our ETL project purpose was to extract different kinds of data related to oil (production, reserves, prices etc.) from different sources and data types (web scraping, and excel files). Web scraped tables were scraped from wikepedia, the brent and crude oil prices were retrieved from the eia.gov website, and all other excel data was retrieved from the OPEC website. We made use of a module called dataflows which enables you to access and use any data from html over json to excel and more.

The benefit of this module is that you do not have to download any resources. One also stays flexible in regards of adding new data if it stays a small to medium sized database. It is faster tehn pandas compiling and extremely generic which is beyond wanted for an ETL Pipeline project.

Transform

The wikepedia tables were loaded into a pandas data frame using the .read\_html function. After retrieving these tables they were cleaned of all extra tags and formatted to be cleanly inputted into our SQL database.

The dataflows data was saved as raw csv files, which after preliminary formatting with dataflows needed to be cleaned and transformed a bit more for readability and usability. This was being done by using pandas and jupyter notebook. Simple commands for slicing as iloc , reset index and drop\_na were used.

From there we exported all data to csv files in the data folder ready to be loaded.

Load

To load our database we first connected to postgres via psycopg2, and from here we created a new database named “oil”. If you run the etl\_script, you will be asked for your username and password. The entire process is controlled via a python script and needs no access of PGadmin. This connection was closed and another connection was created to connect to the newly created oil database using sqlalchemy. From here we used the os.listdir() method to check our folder containing our csv files and make sure each of these files was a csv file and that they were not corrupt. A list of the non-corrupt csv files was added to a list. A for loop was run over these csv files which reads the csvs into a dataframe and inserts this dataframe into a new table in our oil SQL database. We added if and else statements on this for loop to handle new csvs and csvs which may interfere with already existing tables.

We also added a security method via sql query to keep the integrity of historic data alive and guarantee no data loss. We basically add the new dataframe if a table with the same name already exists and remove any duplicates in postgres.

Scripting

The extract, transform, and load processes were executed using a bash script to ensure integrity and dataflow. To execute the script just run etl\_script.sh