# **ETL Project REPORT**

# Meng Chen, Yi Xiao, Mfon Udo-Imeh

# **Project Title** – WORLDWIDE Oil Prices Vs Consumer Inflation Rates

# **Objective –** Create a database of Worldwide country oil and gas volumes and prices data with the inflation rates for the year 2000.

**# EXTRACT - Data Source for ETL project**

2 CSV files were downloaded from Kaggle

File 1 - inflation\_consumer\_prices.csv - Contains worldwide consumer inflation data for countries from 1960 - 2017

File 2 - Oil and Gas 1932-2014.csv - Contains world wide oil and gas production volumes and prices from 1932 - 2014

**# TRANSFORM - Data Cleanup and Analysis**

In Jupyter Notebook the following was performed

Put each CSV into a pandas DataFrame

The files were filtered to show only data for the year 2000.

Copy Only the columns needed into a new dataFrame from the 2 data files

For file 1 - Inflation Data - columns - "Country Name", County Code", "2000: Inflation",

For file 2 - Oil and Gas Data - "cty\_name","id","year","oil\_price\_2000","oil\_value\_2000","gas\_price\_2000","population

Rename as required to cleanup data set

Perform transformation on the "oil gas price" dataframe and group on country id to get data rolled up to country level since the inflation data file information is given at country level.

Drop the null data from the inflation data file

Use aggregation averages to get the right values for oil price, gas price, oil volumes and population columns from the "oil and gas Data"

Cleaned data consist of Countries showing inflation rates and the oil and gas prices for the year 2000.

**# LOAD Data into Database**

The Extracted data is in rows and columns so it will be loaded into the Postgress relational database

Create 2 tables in Postgress Database to hold the 2 data sets

1. Country Oil and Gas Volumes and Prices data
2. Country inflation data

Create a connection to database.

Check for a successful connection to the Postgres database and confirm that the tables have been created.

Append DataFrames to tables. Use indexes as required.

Confirm successful \*\*Load\*\* by querying database.

Join the two tables and select the data to review and perform further analysis

Query showing loaded joined table data is shown in the jupyter notebote.