ETL Project Proposal (Group: ETL too hard)

# Question to be answered with data uploaded in PostgreSQL:

1. The correlation between unemployment rate and the amount of job postings
2. Find the most promising/growing industry in 2016

Ratio1= job postings / unemployment rate

Ratio2= job posting incremental rate per month

# Data Extraction -- all raw data saved in subfolder ‘Resources’:

1. Primary table 1: 2016 US job postings on Dice.com (Source from Kaggle, approximately 22000 rows) <https://www.kaggle.com/PromptCloudHQ/usbased-jobs-from-dicecom>
2. Primary table 2: US unemployment rate by county (1990-2016) (Source from Kaggle) <https://www.kaggle.com/jayrav13/unemployment-by-county-us/data#output.csv>
3. Mapping 1: US state acronym – with all US states’ full name and state codes
4. Mapping 2: Career industry mapping – with job titles and job industries

# Data Transformation – both processes are in main folder:

## Job postings transformation – file name is ‘cleaning job.ipynb’:

1. Remove all non-essential columns
2. Split the date column into month, date, year, and only keep the month column to align with the ‘Unemployment’ table
3. Split the location column and only keep the last two characters as state code
4. Upload to PostgreSQL database

## Unemployment rate table transformation – file name is ‘Unemployment.ipynb’:

1. Remove all years other than 2016 to align time range with ‘Job Posting’ table
2. Group by state and month and take average of the unemployment rate. Since the data is down to county level but we only need state level, we applied this grouping and cut the size of data
3. Change data type of ‘month’ into integer to align with ‘Job Posting’ table
4. Upload to PostgreSQL database

## US State Mapping transformation – file name is ‘Load Mappin.ipynb’:

1. Delete the 1st unknown character in the 1st column – ‘US\_state’ (data read in Python as ‘?’)

# Data Loading – process saved in main folder (python steps are in the 2 jupter files; and postgres process is in file ‘postgres\_tables’:

1. Databased chosen to upload data into: PostgreSQL
2. Created empty tables in PostgreSQL with the same column headers as final version of tables (after transformation)
3. Load all tables that are transformed from Python to PostgreSQL database