Peer Review Assignment - Data Engineer - ETL

Estimated time needed: **20** minutes

Objectives

In this final part you will:

- Run the ETL process
- Extract bank and market cap data from the JSON file `bank_market_cap.json`
- Transform the market cap currency using the exchange rate data
- Load the transformed data into a seperate CSV

For this lab, we are going to be using Python and several Python libraries. Some of these libraries might be installed in your lab environment or in SN Labs. Others may need to be installed by you. The cells below will install these libraries when executed.

In [1]:
```
!pip install glob
!pip install pandas
!pip install requests
!pip install datetime
```

```
ERROR: Could not find a version that satisfies the requirement glob (from versions: none)
ERROR: No matching distribution found for glob
Requirement already satisfied: pandas in /opt/conda/envs/Python-3.8-main/lib/python3.8/site-packages (1.2.4)
Requirement already satisfied: python-dateutil>=2.7.3 in /opt/conda/envs/Python-3.8-main/lib/python3.8/site-packages (from pandas) (2.8.1)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.8-main/lib/python3.8/site-packages (from pandas) (2021.1)
Requirement already satisfied: numpy>=1.16.5 in /opt/conda/envs/Python-3.8-main/lib/python3.8/site-packages (from pandas) (1.19.2)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.8-main/lib/python3.8/site-packages (from python-dateutil>=2.7.3->pandas) (1.15.0)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.8-main/lib/python3.8/site-packages (2.25.1)
Requirement already satisfied: idna<3,>=2.5 in /opt/conda/envs/Python-3.8-main/lib/python3.8/site-packages (from requests) (2.8)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /opt/conda/envs/Python-3.8-main/lib/python3.8/site-packages (from requests) (1.26.6)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/envs/Python-3.8-main/lib/python3.8/site-packages (from requests) (2021.5.30)
Requirement already satisfied: chardet<5,>=3.0.2 in /opt/conda/envs/Python-3.8-main/lib/python3.8/site-packages (from requests) (3.0.4)
Collecting datetime
  Downloading DateTime-4.3-py2.py3-none-any.whl (60 kB)
     |████████████████████████████████| 60 kB 11.8 MB/s eta 0:00:01
Requirement already satisfied: pytz in /opt/conda/envs/Python-3.8-main/lib/python3.8/site-packages (from datetime) (2021.1)
Collecting zope.interface
  Downloading zope.interface-5.4.0-cp38-cp38-manylinux2010_x86_64.whl (259 kB)
     |████████████████████████████████| 259 kB 15.9 MB/s eta 0:00:01
Requirement already satisfied: setuptools in /opt/conda/envs/Python-3.8-main/lib/python3.8/site-packages (from zope.interface->datetime) (52.0.0.post20210125)
Installing collected packages: zope.interface, datetime
Successfully installed datetime-4.3 zope.interface-5.4.0
```

Imports

Import any additional libraries you may need here.

In [2]:
```
import glob
import pandas as pd
from datetime import datetime
```

As the exchange rate fluctuates, we will download the same dataset to make marking simpler. This will be in the same format as the dataset you used in the last section

In [3]:
```
!wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0221EN-SkillsNetwork/labs/module%206/Lab%20-%20Extract%20Transform%20Load/data/
!wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0221EN-SkillsNetwork/labs/module%206/Lab%20-%20Extract%20Transform%20Load/data/
!wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0221EN-SkillsNetwork/labs/module%206/Final%20Assignment/exchange_rates.csv
```

```
--2021-08-15 23:14:00--  https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0221EN-SkillsNetwork/labs/module%206/Lab%20-%20Extract%20Transform%20Load/data/bank_market_cap_1.json
Resolving cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud)... 198.23.119.245
Connecting to cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud)|198.23.119.245|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2815 (2.7K) [application/json]
Saving to: ‘bank_market_cap_1.json’

bank_market_cap_1.j 100%[===================>]   2.75K  --.-KB/s    in 0s

2021-08-15 23:14:00 (69.0 MB/s) - ‘bank_market_cap_1.json’ saved [2815/2815]

--2021-08-15 23:14:01--  https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0221EN-SkillsNetwork/labs/module%206/Lab%20-%20Extract%20Transform%20Load/data/bank_market_cap_2.json
Resolving cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud)... 198.23.119.245
Connecting to cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud)|198.23.119.245|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1429 (1.4K) [application/json]
Saving to: ‘bank_market_cap_2.json’

bank_market_cap_2.j 100%[===================>]   1.40K  --.-KB/s    in 0s

2021-08-15 23:14:01 (35.4 MB/s) - ‘bank_market_cap_2.json’ saved [1429/1429]
```

```
--2021-08-15 23:14:02--  https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0221EN-SkillsNetwork/labs/module%206/Final%20Assignment/exchange_rates.csv
Resolving cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud)... 198.23.119.245
Connecting to cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud)|198.23.119.245|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 590 [text/csv]
Saving to: ‘exchange_rates.csv’

exchange_rates.csv  100%[===================>]     590  --.-KB/s    in 0s

2021-08-15 23:14:02 (16.1 MB/s) - ‘exchange_rates.csv’ saved [590/590]
```

Extract

JSON Extract Function

This function will extract JSON files.

In [4]:
```
def extract_from_json(file_to_process):
    dataframe = pd.read_json(file_to_process)
    return dataframe
```

Extract Function

Define the extract function that finds JSON file `bank_market_cap_1.json` and calls the function created above to extract data from them. Store the data in a `pandas` dataframe. Use the following list for the columns.

In [5]:
```
columns=['Name','Market Cap (US$ Billion)']
```

In [43]:
```
def extract():
    # Write your code here
    extracted_data= pd.DataFrame(columns=['Name','Market Cap (US$ Billion)'])
    #extracted_data
    for jsonfile in glob.glob('bank_market_cap_1.json'):
        extracted_data = extracted_data.append(extract_from_json(jsonfile), ignore_index=True)
    return extracted_data
```

Question 1 Load the file `exchange_rates.csv` as a dataframe and find the exchange rate for British pounds with the symbol `GBP`, store it in the variable `exchange_rate`, you will be asked for the number. Hint: set the parameter `index_col` to 0.

In [35]:
```
# Write your code here
csv_dataframe = pd.read_csv('exchange_rates.csv',index_col=0)
exchange_rate = csv_dataframe.loc['GBP','Rates']
exchange_rate
```

Out[35]: 0.7323984208000001

Transform

Using `exchange_rate` and the `exchange_rates.csv` file find the exchange rate of USD to GBP. Write a transform function that

1. Changes the `Market Cap (US$ Billion)` column from USD to GBP
2. Rounds the Market Cap (US$ Billion)` column to 3 decimal places
3. Rename `Market Cap (US$ Billion)` to `Market Cap (GBP$ Billion)`

In [38]:
```
def transform(data):
    # Write your code here
    data['Market Cap (US$ Billion)']= round(data['Market Cap (US$ Billion)']*exchange_rate,3)
    data.rename(columns={'Market Cap (US$ Billion)':'Market Cap (GBP$ Billion)'}, inplace=True)
    return data
```

Load

Create a function that takes a dataframe and load it to a csv named `bank_market_cap_gbp.csv`. Make sure to set `index` to `False`.

In [39]:
```
def load(data_to_load):
    # Write your code here
    data_to_load.to_csv("bank_market_cap_gbp.csv", index=False)
```

Logging Function

Write the logging function `log` to log your data:

In [40]:
```
def log(message):
    # Write your code here
    timestamp_format= '%h-%d-%Y:%H:%M:%S'
    now= datetime.now()
    timestamp= now.strftime(timestamp_format)
    #with open("Logfile.txt",'a') as data:
    print(timestamp+', '+message+'\n')
```

Running the ETL Process

Log the process accordingly using the following `"ETL Job Started"` and `"Extract phase Started"`

In [41]:
```
# Write your code here
log("ETL Job Started")
log("Extract phase Started")
```

```
Aug-15-2021:23:44:55, ETL Job Started

Aug-15-2021:23:44:55, Extract phase Started
```

Extract

`Question 2` Use the function `extract`, and print the first 5 rows, take a screen shot:

In [42]:
```
# Call the function here
extracted_data=extract()
# Print the rows here
```

```
extracted_data.head()
```

Out[42]:

	Name	Market Cap (US$ Billion)
0	JPMorgan Chase	390.934
1	Industrial and Commercial Bank of China	345.214
2	Bank of America	325.331
3	Wells Fargo	308.013
4	China Construction Bank	257.399

Log the data as `"Extract phase Ended"`

In [15]:
```
# Write your code here
log("Extract phase Ended")
```

```
Aug-15-2021:23:22:56, Extract phase Ended
```

Transform

Log the following `"Transform phase Started"`

In [16]:
```
# Write your code here
log("Transform phase Started")
```

```
Aug-15-2021:23:23:14, Transform phase Started
```

`Question 3` Use the function `transform` and print the first 5 rows of the output, take a screen shot:

In [17]:
```
# Call the function here
transformed_data=transform(extracted_data)
# Print the first 5 rows here
transformed_data.head()
```

Out[17]:

	Name	Market Cap (GBP$ Billion)
0	JPMorgan Chase	286.319
1	Industrial and Commercial Bank of China	252.834
2	Bank of America	238.272
3	Wells Fargo	225.588
4	China Construction Bank	188.519

Log your data `"Transform phase Ended"`

In [18]:
```
# Write your code here
# Write your code here
log("Transform phase Ended")
```

```
Aug-15-2021:23:23:50, Transform phase Ended
```

Load

Log the following `"Load phase Started"`.

In [19]:
```
# Write your code here
log("Load phase Started")
```

```
Aug-15-2021:23:24:04, Load phase Started
```

Call the load function

In [20]:
```
# Write your code here
load(transformed_data)
```

Log the following `"Load phase Ended"`.

In [21]:
```
# Write your code here
log("Load phase Ended")
```

```
Aug-15-2021:23:24:32, Load phase Ended
```

Authors

Ramesh Sannareddy, Joseph Santrcangelo and Azim Hirjani

Other Contributors

Rav Ahuja

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2020-11-25	0.1	Ramesh Sannareddy	Created initial version of the lab