In [1]:
```
import pandas as pd
import json as js
from functools import reduce
import datetime as dt
from sqlalchemy import create_engine
```

In [2]:
```
# open json file
with open("../Resources/kossal_copy_resources/all_electricity_supply.json") as supply_json:
    supply = js.load(supply_json)
```

In [3]:
```
# find number of columns from the data fram
num_of_col = len(supply)
num_of_col
```

Out[3]:
```
12
```

In [4]:
```
s='Net generation : coal : United States : all sectors : monthly'
start='Net generation : '
end=' : United States : all sectors : monthly'
text=s[s.find(start)+len(start):s.rfind(end)]
```

In [5]:
```
# find columns names string and get the power sources
# supply[0]['name'] result in string 'Net generation : coal : United States : all sectors : monthly'
col_name = []
for i in range(len(supply)):
    s=supply[i]['name']
    col_name.append(str(s[s.find(start)+len(start):s.rfind(end)]))
col_name
```

Out[5]:
```
['coal',
 'natural gas',
 'nuclear',
 'conventional hydroelectric',
 'wind',
 'petroleum liquids',
 'all utility-scale solar',
 'geothermal',
 'wood and wood-derived fuels',
 'other gases',
 'other biomass',
 'other']
```

In [6]:
```
# create list of dataFrame Name
df_name=['df_'+str(i) for i in range(len(supply))]
df_name
```

Out[6]:
```
['df_0',
 'df_1',
 'df_2',
 'df_3',
 'df_4',
 'df_5',
 'df_6',
 'df_7',
 'df_8',
 'df_9',
 'df_10',
 'df_11']
```

In [7]:
```
# Use for loop to create data frames
for index, col in enumerate(col_name):
       exec(f"{df_name[index]} = pd.DataFrame({supply[index]['data']}, columns=['date_id', '{col}'])")
```

In [8]:
```
# create a list of DataFrames for join loop
df_list = [eval(temp) for temp in df_name]
```

In [9]:
```
# join DataFrames
df_energy_source = reduce(lambda x, y: pd.merge(x,y, on = 'date_id'), df_list)
df_energy_source.head()
```

Out[9]:

	date_id	coal	natural gas	nuclear	conventional hydroelectric	wind	petroleum liquids	all utility-scale solar	geothermal	wood and wood-derived fuels	other gases	other biomass	otl
0	202009	68395.67702	141406.66981	65727.317	19182.11175	23176.03240	794.67219	7757.40868	1390.39750	2938.04250	947.17872	1512.22945	1006.847
1	202008	91164.99383	174129.84618	68982.187	24227.83917	22570.71305	925.96988	9369.91528	1426.23924	3302.49204	1002.10601	1589.15438	1096.055
2	202007	89845.33355	185433.35809	69385.440	27753.19342	22579.21085	963.15441	10379.27930	1430.94455	3111.53745	853.95303	1564.17641	1088.603
3	202006	65274.37394	143179.90065	67205.083	29137.98926	29871.72279	864.67926	9529.17251	1369.44148	2993.49186	756.49600	1458.91919	1016.003
4	202005	46529.48478	115854.51716	64337.970	30559.05329	28163.26916	707.14715	9734.37493	1448.82619	3099.05893	807.66622	1616.93726	1088.421

In [10]:
```
# rename column to for SQL Tables
df_energy_source=df_energy_source.rename(columns={'natural gas':'natural_gas',
                                  'conventional hydroelectric':'hydroelectric',
                                  'petroleum liquids':'petroleum_liquids',
                                  'all utility-scale solar': 'solar',
                                  'wood and wood-derived fuels':'wood',
                                  'other gases':'other_gases',
                                  'other biomass': 'other_biomass'
                                 })
df_energy_source.head()
```

Out[10]:

	date_id	coal	natural_gas	nuclear	hydroelectric	wind	petroleum_liquids	solar	geothermal	wood	other_gases	other_bioma
0	202009	68395.67702	141406.66981	65727.317	19182.11175	23176.03240	794.67219	7757.40868	1390.39750	2938.04250	947.17872	1512.229
1	202008	91164.99383	174129.84618	68982.187	24227.83917	22570.71305	925.96988	9369.91528	1426.23924	3302.49204	1002.10601	1589.154
2	202007	89845.33355	185433.35809	69385.440	27753.19342	22579.21085	963.15441	10379.27930	1430.94455	3111.53745	853.95303	1564.176
3	202006	65274.37394	143179.90065	67205.083	29137.98926	29871.72279	864.67926	9529.17251	1369.44148	2993.49186	756.49600	1458.919
4	202005	46529.48478	115854.51716	64337.970	30559.05329	28163.26916	707.14715	9734.37493	1448.82619	3099.05893	807.66622	1616.937

In [11]:
```
# columns of interest
col_int = ['date_id','natural_gas', 'nuclear', 'hydroelectric', 'wind', 'solar', 'wood', 'other_biomass', 'other']
```

In [12]:
```
# select column of interest
df_ener_sour = df_energy_source[col_int]
df_ener_sour = df_ener_sour.sort_values(by=['date_id'], ascending = True).reset_index(drop=True)
df_ener_sour.head()
```

Out[12]:

	date_id	natural_gas	nuclear	hydroelectric	wind	solar	wood	other_biomass	other
0	200101	42388.663	68707.077	18852.048	389.250	6.500	3191.212	1208.64715	991.66885
1	200102	37966.927	61272.407	17472.889	431.242	12.568	2697.151	1096.33055	871.94945
2	200103	44364.414	62140.712	20477.189	532.120	31.498	2852.540	1189.04028	931.15472
3	200104	45842.746	56003.026	18012.994	684.695	38.759	2821.007	1187.23919	957.45581
4	200105	50934.205	61512.445	19175.635	635.029	81.053	2739.933	1210.74093	986.26807

In [13]:
```
ener_sour_df = df_ener_sour.drop(columns='date_id')
ener_sour_df.head()
```

Out[13]:

	natural_gas	nuclear	hydroelectric	wind	solar	wood	other_biomass	other
0	42388.663	68707.077	18852.048	389.250	6.500	3191.212	1208.64715	991.66885
1	37966.927	61272.407	17472.889	431.242	12.568	2697.151	1096.33055	871.94945
2	44364.414	62140.712	20477.189	532.120	31.498	2852.540	1189.04028	931.15472
3	45842.746	56003.026	18012.994	684.695	38.759	2821.007	1187.23919	957.45581
4	50934.205	61512.445	19175.635	635.029	81.053	2739.933	1210.74093	986.26807

In [14]:
```
ener_sour_df.index.rename('date_id', inplace=True)
```

In [15]:
```
ener_sour_df.head()
```

Out[15]:

date_id	natural_gas	nuclear	hydroelectric	wind	solar	wood	other_biomass	other
0	42388.663	68707.077	18852.048	389.250	6.500	3191.212	1208.64715	991.66885
1	37966.927	61272.407	17472.889	431.242	12.568	2697.151	1096.33055	871.94945
2	44364.414	62140.712	20477.189	532.120	31.498	2852.540	1189.04028	931.15472
3	45842.746	56003.026	18012.994	684.695	38.759	2821.007	1187.23919	957.45581
4	50934.205	61512.445	19175.635	635.029	81.053	2739.933	1210.74093	986.26807

Load Data into Postgres

In [24]:
```
# Create database connection
connection_string = "postgres:postgres@localhost:5432/ETL_Project"
engine = create_engine(f'postgresql://{connection_string}')
```

In [25]:
```
# Confirm tables
engine.table_names()
```

Out[25]:
```
['Date', 'Demand_MillionKWH', 'Price_CentsPerKWH', 'Supply_ThousandMWH']
```

In [26]:
```
# Load DataFrame into database
ener_sour_df.to_sql(name='Supply_ThousandMWH', con=engine, if_exists='append', index=True)
```

In [ ]:
```

```