

Assignment 2

Name: Kesav Adithya Venkidusamy

Course: DSC650 - Big Data

Instructor: Amirfarrokh Iranitalab

In [17]:

```
import json
from pathlib import Path
import os

import pandas as pd
import s3fs

def read_cluster_csv(file_path, endpoint_url='https://storage.budsc.midwest-datascience.com'):
    s3 = s3fs.S3FileSystem(
        anon=True,
        client_kwargs={
            'endpoint_url': endpoint_url
        }
    )
    return pd.read_csv(s3.open(file_path, mode='rb'))

current_dir = Path(os.getcwd()).absolute()
results_dir = current_dir.joinpath('results')
kv_data_dir = results_dir.joinpath('kvdb')
kv_data_dir.mkdir(parents=True, exist_ok=True)

people_json = kv_data_dir.joinpath('people.json')
visited_json = kv_data_dir.joinpath('visited.json')
sites_json = kv_data_dir.joinpath('sites.json')
measurements_json = kv_data_dir.joinpath('measurements.json')

## Setting up the directory name for source files

sites_dir = r'C:\Users\KesavAdithya\Documents\GitHub\dsc650\data\external\tidynomicon\site.csv'
person_dir = r'C:\Users\KesavAdithya\Documents\GitHub\dsc650\data\external\tidynomicon\person.csv'
visit_dir = r'C:\Users\KesavAdithya\Documents\GitHub\dsc650\data\external\tidynomicon\visited.csv'
measure_dir = r'C:\Users\KesavAdithya\Documents\GitHub\dsc650\data\external\tidynomicon\measurements.csv'
```

In [18]:

```
class KVDB(object):
    def __init__(self, db_path):
        self._db_path = Path(db_path)
        self._db = {}
        self._load_db()

    def _load_db(self):
        if self._db_path.exists():
            with open(self._db_path) as f:
                self._db = json.load(f)

    def get_value(self, key):
        return self._db.get(key)

    def set_value(self, key, value):
        self._db[key] = value

    def save(self):
        with open(self._db_path, 'w') as f:
            json.dump(self._db, f, indent=2)
```

In [19]:

```
def create_sites_kvdb():
    db = KVDB(sites_json)
    #df = read_cluster_csv('data/external/tidynomicon/site.csv')
    df = pd.read_csv(sites_dir)
    for site_id, group_df in df.groupby('site_id'):
        db.set_value(site_id, group_df.to_dict(orient='records')[0])
    db.save()
    print("The key-value database for sites file has been created successfully")

def create_people_kvdb():
    db = KVDB(people_json)
    ## TODO: Implement code
    df = pd.read_csv(person_dir)
    for person_id, group_df in df.groupby('person_id'):
        db.set_value(person_id, group_df.to_dict(orient='records')[0])
    db.save()
    print("The key-value database for person file has been created successfully")

def create_visits_kvdb():
    db = KVDB(visited_json)
```

```
## TODO: Implement code
## This dataset contains Visit id and site id; So, they will be used as composite key;
df = pd.read_csv(visit_dir)
for composite_id, group_df in df.groupby(['visit_id', 'site_id']):
    db.set_value(str(composite_id), group_df.to_dict(orient='records')[0])
db.save()
print("The key-value database for visit file has been created successfully")

def create_measurements_kvdb():
    db = KVDB(measurements_json)
    ## TODO: Implement code
    ## This dataset contains visit_id, person_id and quantity as unique key; So, they will be used as composite key
    df = pd.read_csv(measure_dir)
    for composite_id, group_df in df.groupby(['visit_id', 'person_id', 'quantity']):
        db.set_value(str(composite_id), group_df.to_dict(orient='records')[0])
    db.save()
    print("The key-value database for measurements file has been created successfully")
```

In [20]:

```
create_sites_kvdb()
create_people_kvdb()
create_visits_kvdb()
create_measurements_kvdb()
```

The key-value database for sites file has been created successfully
The key-value database for person file has been created successfully
The key-value database for visit file has been created successfully
The key-value database for measurements file has been created successfully

In []: