In [1]:
```
from pathlib import Path
import os
import sqlite3

import s3fs
import pandas as pd

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)


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

Create and Load Measurements Table

In [2]:
```
def create_measurements_table(conn):
    sql = """
    CREATE TABLE IF NOT EXISTS measurements (
        visit_id integer NOT NULL,
        person_id text NOT NULL,
        quantity text,
        reading real,
        FOREIGN KEY (visit_id) REFERENCES visits (visit_id),
        FOREIGN KEY (person_id) REFERENCES people (people_id)
        );
    """
    c = conn.cursor()
    c.execute(sql)

def load_measurements_table(conn):
    create_measurements_table(conn)
    df = read_cluster_csv('measurements.csv')
    measurements = df.values
    c = conn.cursor()
    c.execute('DELETE FROM measurements;') # Delete data if exists
    c.executemany('INSERT INTO measurements VALUES (?,?,?,?)', measurements)
```

Create and Load People Table

In [3]:
```
def create_people_table(conn):
    sql = """
    CREATE TABLE IF NOT EXISTS people (
        person_id text PRIMARY KEY,
        personal_name text NOT NULL,
        family_name text NOT NULL
        );
    """
    ## TODO: Complete SQL
    c = conn.cursor()
    c.execute(sql)
```

```
def load_people_table(conn):
    create_people_table(conn)
    ## TODO: Complete code
    df = read_cluster_csv('person.csv')
    people = df.values
    c = conn.cursor()
    c.execute('DELETE FROM people;')  # Delete data if exists
    c.executemany('INSERT INTO people VALUES (?,?,?)', people)
```

Create and Load Sites Table

In [4]:
```
def create_sites_table(conn):
    sql = """
    CREATE TABLE IF NOT EXISTS sites (
        site_id text PRIMARY KEY,
        latitude double NOT NULL,
        longitude double NOT NULL
        );
    """
    c = conn.cursor()
    c.execute(sql)

def load_sites_table(conn):
    create_sites_table(conn)
    ## TODO: Complete code
    df = read_cluster_csv('site.csv')
    sites = df.values
    c = conn.cursor()
    c.execute('DELETE FROM sites;')  # Delete data if exists
    c.executemany('INSERT INTO sites VALUES (?,?,?)', sites)
```

Create and Load Visits Table

In [5]:
```
def create_visits_table(conn):
    sql = """
    CREATE TABLE IF NOT EXISTS visits (
        visit_id integer PRIMARY KEY,
        site_id text NOT NULL,
        visit_date text,
        FOREIGN KEY (site_id) REFERENCES sites (site_id)
        );
    """

    c = conn.cursor()
    c.execute(sql)

def load_visits_table(conn):
    create_visits_table(conn)
    ## TODO: Complete code
    df = read_cluster_csv('visited.csv')
    visits = df.values
    c = conn.cursor()
    c.execute('DELETE FROM visits;')  # Delete data if exists
    c.executemany('INSERT INTO visits VALUES (?,?,?)', visits)
```

Create DB and Load Tables

In [6]:
```
db_path = results_dir.joinpath('patient-info.db')
```

```
conn = sqlite3.connect(str(db_path))
# TODO: Uncomment once functions completed
load_people_table(conn)
load_sites_table(conn)
load_visits_table(conn)
load_measurements_table(conn)

sql = '''SELECT * FROM visits;'''

c = conn.cursor()
c.execute(sql)

result = c.fetchall()
print(result)

conn.commit()
conn.close()
```

```
[(619, 'DR-1', '1927-02-08'), (622, 'DR-1', '1927-02-10'), (734, 'DR-3', '1930-01-07'),
(735, 'DR-3', '1930-01-12'), (751, 'DR-3', '1930-02-26'), (752, 'DR-3', None), (837, 'MS
K-4', '1932-01-14'), (844, 'DR-1', '1932-03-22')]
```