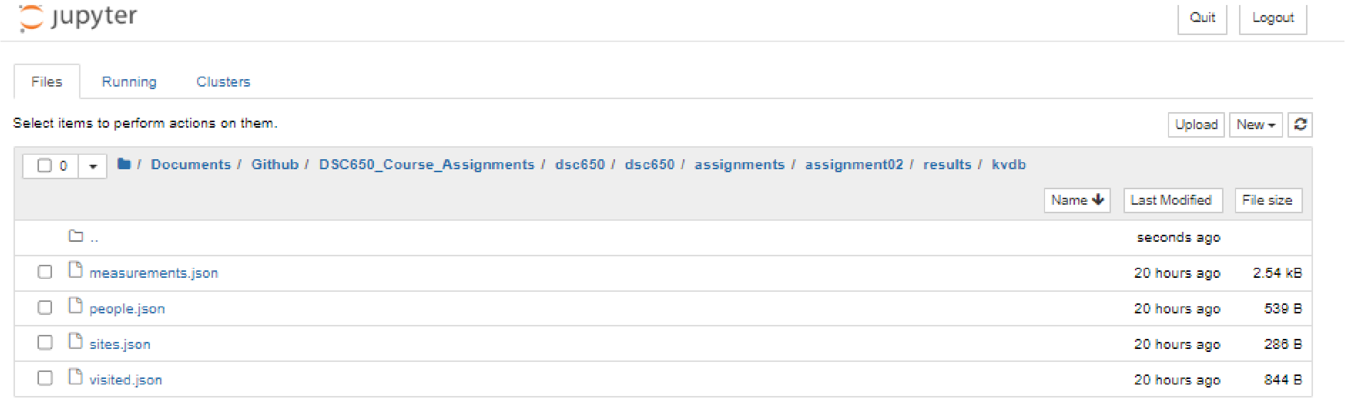
DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 02 Code and Outputs
Jake Meyer
03/27/2023

Assignment 02 Outputs and Code

Assignment 2.1 Code Output:



Select items to perform actions on them.

/ Documents / Github / DSC650_Course_Assignments / dsc650 / dsc650 / assignments / assignment02 / results / kvdb

Name	Last Modified	File size
..	seconds ago	
measurements.json	20 hours ago	2.54 kB
people.json	20 hours ago	539 B
sites.json	20 hours ago	286 B
visited.json	20 hours ago	844 B

Measurements.json Output:

```
{
 "(619, 'dyer', 'rad')": {
  "visit_id": 619,
  "person_id": "dyer",
  "quantity": "rad",
  "reading": 9.82
 },
 "(619, 'dyer', 'sal')": {
  "visit_id": 619,
  "person_id": "dyer",
  "quantity": "sal",
  "reading": 0.13
 },
 "(622, 'dyer', 'rad')": {
  "visit_id": 622,
  "person_id": "dyer",
  "quantity": "rad",
  "reading": 7.8
 },
 "(622, 'dyer', 'sal')": {
  "visit_id": 622,
  "person_id": "dyer",
  "quantity": "sal",
  "reading": 0.09
 },
 "(734, 'lake', 'sal')": {
  "visit_id": 734,
    "person_id": "lake",
    "quantity": "sal",
    "reading": 0.05
  },
  "(734, 'pb', 'rad')": {
    "visit_id": 734,
    "person_id": "pb",
    "quantity": "rad",
    "reading": 8.41
  },
  "(734, 'pb', 'temp')": {
    "visit_id": 734,
    "person_id": "pb",
    "quantity": "temp",
    "reading": -21.5
  },
  "(735, 'pb', 'rad')": {
    "visit_id": 735,
    "person_id": "pb",
    "quantity": "rad",
    "reading": 7.22
  },
  "(735, 'pb', 'sal')": {
    "visit_id": 735,
    "person_id": "pb",
    "quantity": "sal",
    "reading": 0.06
  },
  "(735, 'pb', 'temp')": {
    "visit_id": 735,
    "person_id": "pb",
    "quantity": "temp",
    "reading": -26.0
  },
  "(751, 'pb', 'rad')": {
    "visit_id": 751,
    "person_id": "pb",
    "quantity": "rad",
    "reading": 4.35
  },
  "(751, 'pb', 'temp')": {
    "visit_id": 751,
    "person_id": "pb",
    "quantity": "temp",
    "reading": -18.5
  },
  "(752, 'lake', 'rad')": {
    "visit_id": 752,
    "person_id": "lake",
    "quantity": "rad",
    "reading": 2.19
  },
  "(752, 'lake', 'sal')": {
    "visit_id": 752,
    "person_id": "lake",
    "quantity": "sal",
    "reading": 0.09
  },
  "(752, 'lake', 'temp')": {
    "visit_id": 752,
    "person_id": "lake",
    "quantity": "temp",
    "reading": -16.0
  },
  "(752, 'roe', 'sal')": {
    "visit_id": 752,
    "person_id": "roe",
    "quantity": "sal",
    "reading": 41.6
  },
  "(837, 'lake', 'rad')": {
    "visit_id": 837,
    "person_id": "lake",
    "quantity": "rad",
    "reading": 1.46
  },
  "(837, 'lake', 'sal')": {
    "visit_id": 837,
    "person_id": "lake",
    "quantity": "sal",
    "reading": 0.21
  },
  "(837, 'roe', 'sal')": {
    "visit_id": 837,
    "person_id": "roe",
    "quantity": "sal",
    "reading": 22.5
  },
  "(844, 'roe', 'rad')": {
    "visit_id": 844,
    "person_id": "roe",
    "quantity": "rad",
    "reading": 11.25
  }
}
```

People.json Output:

```
{
  "danforth": {
    "person_id": "danforth",
    "personal_name": "Frank",
    "family_name": "Danforth"
  },
  "dyer": {
    "person_id": "dyer",
    "personal_name": "William",
    "family_name": "Dyer"
  },
  "lake": {
    "person_id": "lake",
    "personal_name": "Anderson",
    "family_name": "Lake"
  },
  "pb": {
    "person_id": "pb",
    "personal_name": "Frank",
    "family_name": "Pabodie"
  },
  "roe": {
    "person_id": "roe",
    "personal_name": "Valentina",
    "family_name": "Roerich"
  }
}
```

Sites.json Output:

```
{
  "DR-1": {
    "site_id": "DR-1",
    "latitude": -49.85,
    "longitude": -128.57
  },
  "DR-3": {
    "site_id": "DR-3",
    "latitude": -47.15,
    "longitude": -126.72
  },
  "MSK-4": {
    "site_id": "MSK-4",
    "latitude": -48.87,
    "longitude": -123.4
  }
}
```

Visited.json Output:

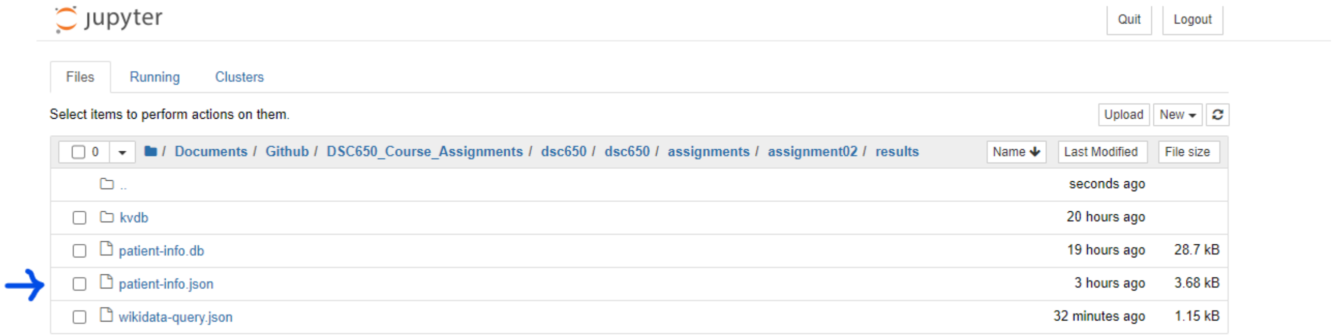
```
{
  "(619, 'DR-1')": {
    "visit_id": 619,
    "site_id": "DR-1",
    "visit_date": "1927-02-08"
  },
  "(622, 'DR-1')": {
    "visit_id": 622,
    "site_id": "DR-1",
    "visit_date": "1927-02-10"
  },
  "(734, 'DR-3')": {
    "visit_id": 734,
    "site_id": "DR-3",
    "visit_date": "1930-01-07"
  },
  "(735, 'DR-3')": {
    "visit_id": 735,
    "site_id": "DR-3",
    "visit_date": "1930-01-12"
  },
  "(751, 'DR-3')": {
    "visit_id": 751,
    "site_id": "DR-3",
    "visit_date": "1930-02-26"
  },
  "(752, 'DR-3')": {
    "visit_id": 752,
    "site_id": "DR-3",
    "visit_date": NaN
  },
```

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 02 Code and Outputs
Jake Meyer
03/27/2023

```
  "(837, 'MSK-4')": {
    "visit_id": 837,
    "site_id": "MSK-4",
    "visit_date": "1932-01-14"
  },
  "(844, 'DR-1')": {
    "visit_id": 844,
    "site_id": "DR-1",
    "visit_date": "1932-03-22"
  }
}
```

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 02 Code and Outputs
Jake Meyer
03/27/2023

Assignment 2.2 Code Output:



Select items to perform actions on them.

/ Documents / Github / DSC650_Course_Assignments / dsc650 / dsc650 / assignments / assignment02 / results

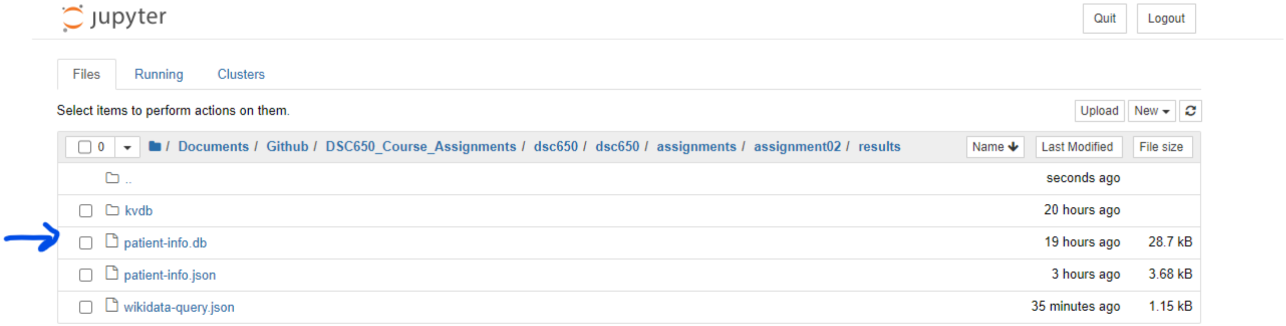
```
..                        seconds ago
kvdb                      20 hours ago
patient-info.db           19 hours ago   28.7 kB
patient-info.json         3 hours ago    3.68 kB
wikidata-query.json       32 minutes ago 1.15 kB
```

Patient-info.json Output:

{"_default": {"1": {"person_id": "danforth", "personal_name": "Frank", "family_name": "Danforth", "visits": []}, "2": {"person_id": "dyer", "personal_name": "William", "family_name": "Dyer", "visits": [{"visit_id": 619, "site_id": "DR-1", "visit_date": "1927-02-08", "site": {"site_id": "DR-1", "latitude": -49.85, "longitude": -128.57}, "measurements": [{"visit_id": 619, "person_id": "dyer", "quantity": "rad", "reading": 9.82}, {"visit_id": 619, "person_id": "dyer", "quantity": "sal", "reading": 0.13}]}, {"visit_id": 622, "site_id": "DR-1", "visit_date": "1927-02-10", "site": {"site_id": "DR-1", "latitude": -49.85, "longitude": -128.57}, "measurements": [{"visit_id": 622, "person_id": "dyer", "quantity": "rad", "reading": 7.8}, {"visit_id": 622, "person_id": "dyer", "quantity": "sal", "reading": 0.09}]}]}, "3": {"person_id": "lake", "personal_name": "Anderson", "family_name": "Lake", "visits": [{"visit_id": 752, "site_id": "DR-3", "visit_date": NaN, "site": {"site_id": "DR-3", "latitude": -47.15, "longitude": -126.72}, "measurements": [{"visit_id": 752, "person_id": "lake", "quantity": "rad", "reading": 2.19}, {"visit_id": 752, "person_id": "lake", "quantity": "sal", "reading": 0.09}, {"visit_id": 752, "person_id": "lake", "quantity": "temp", "reading": -16.0}]}, {"visit_id": 837, "site_id": "MSK-4", "visit_date": "1932-01-14", "site": {"site_id": "MSK-4", "latitude": -48.87, "longitude": -123.4}, "measurements": [{"visit_id": 837, "person_id": "lake", "quantity": "rad", "reading": 1.46}, {"visit_id": 837, "person_id": "lake", "quantity": "sal", "reading": 0.21}]}, {"visit_id": 734, "site_id": "DR-3", "visit_date": "1930-01-07", "site": {"site_id": "DR-3", "latitude": -47.15, "longitude": -126.72}, "measurements": [{"visit_id": 734, "person_id": "lake", "quantity": "sal", "reading": 0.05}]}]}, "4": {"person_id": "pb", "personal_name": "Frank", "family_name": "Pabodie", "visits": [{"visit_id": 751, "site_id": "DR-3", "visit_date": "1930-02-26", "site": {"site_id": "DR-3", "latitude": -47.15, "longitude": -126.72}, "measurements": [{"visit_id": 751, "person_id": "pb", "quantity": "rad", "reading": 4.35}, {"visit_id": 751, "person_id": "pb", "quantity": "temp", "reading": -18.5}]}, {"visit_id": 734, "site_id": "DR-3", "visit_date": "1930-01-07", "site": {"site_id": "DR-3", "latitude": -47.15, "longitude": -126.72}, "measurements": [{"visit_id": 734, "person_id": "pb", "quantity": "rad", "reading": 8.41}, {"visit_id": 734, "person_id": "pb", "quantity": "temp", "reading": -21.5}]}, {"visit_id": 735, "site_id": "DR-3", "visit_date": "1930-01-12", "site": {"site_id": "DR-3", "latitude": -47.15, "longitude": -126.72}, "measurements": [{"visit_id": 735, "person_id": "pb", "quantity": "rad", "reading": 7.22}, {"visit_id": 735, "person_id": "pb", "quantity": "sal", "reading": 0.06}, {"visit_id": 735, "person_id": "pb", "quantity": "temp", "reading": -26.0}]}]}, "5": {"person_id": "roe", "personal_name": "Valentina", "family_name": "Roerich", "visits": [{"visit_id": 752, "site_id": "DR-3", "visit_date": NaN, "site": {"site_id": "DR-3", "latitude": -47.15, "longitude": -126.72}, "measurements": [{"visit_id": 752, "person_id": "roe", "quantity": "sal", "reading": 41.6}]}, {"visit_id": 844, "site_id": "DR-1", "visit_date": "1932-03-22", "site": {"site_id": "DR-1", "latitude": -49.85, "longitude": -128.57}, "measurements": [{"visit_id": 844, "person_id": "roe", "quantity": "rad", "reading": 11.25}]}, {"visit_id": 837, "site_id": "MSK-4", "visit_date": "1932-01-14", "site": {"site_id": "MSK-4", "latitude": -48.87, "longitude": -123.4}, "measurements": [{"visit_id": 837, "person_id": "roe", "quantity": "sal", "reading": 22.5}]}]}}}

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 02 Code and Outputs
Jake Meyer
03/27/2023

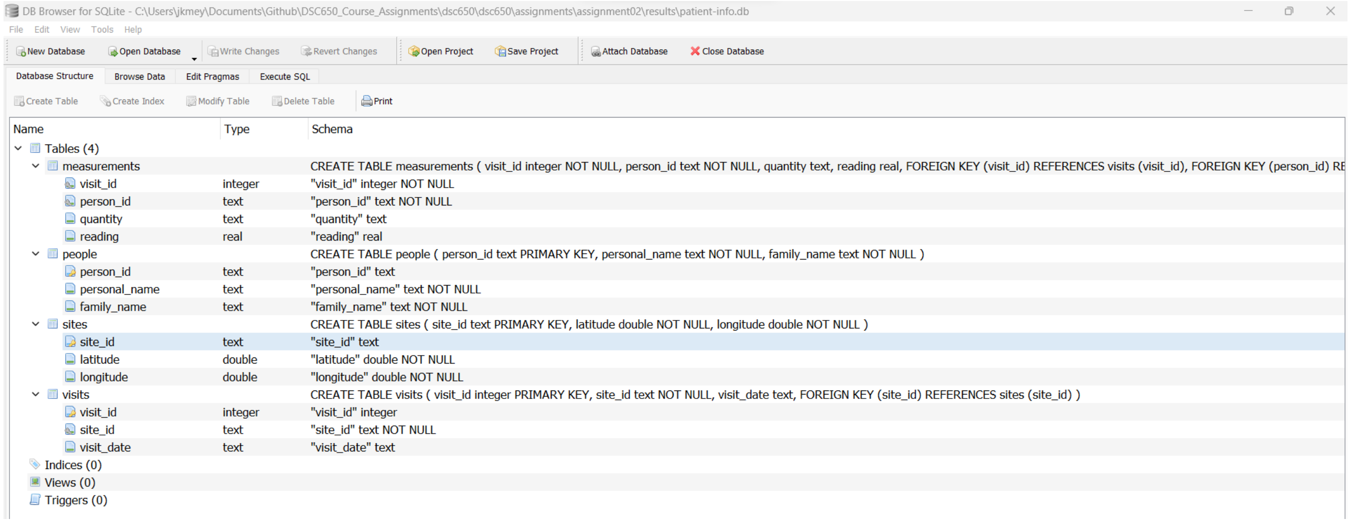
Assignment 2.3 Code Output:



Select items to perform actions on them.

/ Documents / Github / DSC650_Course_Assignments / dsc650 / dsc650 / assignments / assignment02 / results

Name	Last Modified	File size
..	seconds ago	
kvdb	20 hours ago	
patient-info.db	19 hours ago	28.7 kB
patient-info.json	3 hours ago	3.68 kB
wikidata-query.json	35 minutes ago	1.15 kB

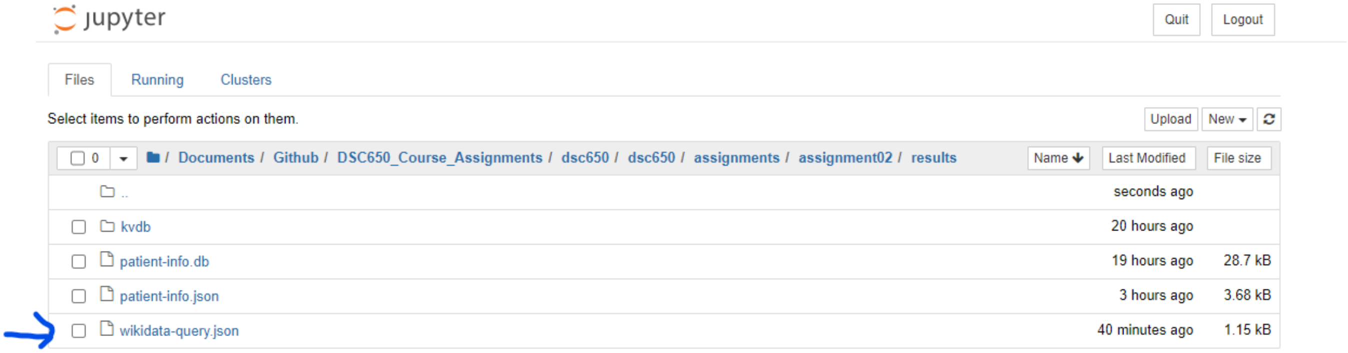


DB Browser for SQLite - C:\Users\jkmey\Documents\Github\DSC650_Course_Assignments\dsc650\dsc650\assignments\assignment02\results\patient-info.db

Name	Type	Schema
Tables (4)		
measurements		CREATE TABLE 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) RE
visit_id	integer	"visit_id" integer NOT NULL
person_id	text	"person_id" text NOT NULL
quantity	text	"quantity" text
reading	real	"reading" real
people		CREATE TABLE people (person_id text PRIMARY KEY, personal_name text NOT NULL, family_name text NOT NULL)
person_id	text	"person_id" text
personal_name	text	"personal_name" text NOT NULL
family_name	text	"family_name" text NOT NULL
sites		CREATE TABLE sites (site_id text PRIMARY KEY, latitude double NOT NULL, longitude double NOT NULL)
site_id	text	"site_id" text
latitude	double	"latitude" double NOT NULL
longitude	double	"longitude" double NOT NULL
visits		CREATE TABLE visits (visit_id integer PRIMARY KEY, site_id text NOT NULL, visit_date text, FOREIGN KEY (site_id) REFERENCES sites (site_id))
visit_id	integer	"visit_id" integer
site_id	text	"site_id" text NOT NULL
visit_date	text	"visit_date" text
Indices (0)		
Views (0)		
Triggers (0)		

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 02 Code and Outputs
Jake Meyer
03/27/2023

Assignment 2.4 Code Output:



Select items to perform actions on them.

□ 0 / Documents / Github / DSC650_Course_Assignments / dsc650 / dsc650 / assignments / assignment02 / results

Name	Last Modified	File size
..	seconds ago	
□ kvdb	20 hours ago	
□ patient-info.db	19 hours ago	28.7 kB
□ patient-info.json	3 hours ago	3.68 kB
□ wikidata-query.json	40 minutes ago	1.15 kB

[{"date":"2023-03-09T00:00:00Z","event":"http://www.wikidata.org/entity/Q111458258"},{"date":"2023-03-02T00:00:00Z","event":"http://www.wikidata.org/entity/Q111458314","eventLabel":"2022–23 Biathlon World Cup – Stage 7"},{"date":"2023-03-16T00:00:00Z","event":"http://www.wikidata.org/entity/Q111458340"},{"date":"2023-03-05T00:00:00Z","event":"http://www.wikidata.org/entity/Q111460810","eventLabel":"2023 Vasaloppet"},{"date":"2023-03-12T00:00:00Z","event":"http://www.wikidata.org/entity/Q115801843","eventLabel":"2023 Women's Hockey Junior Africa Cup"},{"date":"2023-03-12T00:00:00Z","event":"http://www.wikidata.org/entity/Q115802035","eventLabel":"2023 Men's Hockey Junior Africa Cup"},{"date":"2023-03-18T00:00:00Z","event":"http://www.wikidata.org/entity/Q115803958","eventLabel":"UFC 286"},{"date":"2023-02-25T00:00:00Z","event":"http://www.wikidata.org/entity/Q115807057","eventLabel":"UFC Fight Night 220"},{"date":"2023-03-04T00:00:00Z","event":"http://www.wikidata.org/entity/Q115857639","eventLabel":"UFC 285"},{"date":"2023-03-11T00:00:00Z","event":"http://www.wikidata.org/entity/Q115857750","eventLabel":"UFC Fight Night 221"}]

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 02 Code and Outputs
Jake Meyer
03/27/2023

Assignment 2.1 Code (kvdb File Code):

```
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')
```

In [2]:

```
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 [3]:

```
def create_sites_kvdb():
    db = KVDB(sites_json)
    df_sites = pd.read_csv('site.csv')
    for site_id, group_df in df_sites.groupby('site_id'):
        db.set_value(site_id, group_df.to_dict(orient='records')[0])
    db.save()


def create_people_kvdb():
    db = KVDB(people_json)
    df_people = pd.read_csv('person.csv')
    for person_id, group_df in df_people.groupby('person_id'):
        db.set_value(person_id, group_df.to_dict(orient='records')[0])
    db.save()


def create_visits_kvdb():
    db = KVDB(visited_json)
    df_visits = pd.read_csv('visited.csv')
    for composite_id, group_df in df_visits.groupby(["visit_id", "site_id"]):
        key=str(composite_id)
        db.set_value(key, group_df.to_dict(orient='records')[0])
    db.save()


def create_measurements_kvdb():
    db = KVDB(measurements_json)
    df_measurements = pd.read_csv('measurements.csv')
    for composite_id, group_df in df_measurements.groupby(['visit_id', 'person_id', 'quantity']):
        key=str(composite_id)
        db.set_value(key, group_df.to_dict(orient='records')[0])
    db.save()
```

In [4]:

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

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 02 Code and Outputs
Jake Meyer
03/27/2023

Assignment 2.2 Code (documentdb.ipynb):

```
from pathlib import Path
import json
import os

from tinydb import TinyDB

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 _load_json(json_path):
    with open(json_path) as f:
        return json.load(f)


class DocumentDB(object):
    ## You can use the code from the previous example if you would like
    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')

    # use with open command for all of the json files
    with open(sites_json) as f:
        _sites_Data = json.load(f)
    with open(measurements_json) as f:
        _measurements_Data = json.load(f)
    with open(people_json) as f:
        _people_Data = json.load(f)
    with open(visited_json) as f:
        _visit_Data = json.load(f)


    def __init__(self, db_path):
        self._db_path = Path(db_path)
        self._db = None
        self._load_db()


    def _get_sites(self, site_id):
        '''
        Function: Get site data
        arguments: site_id (str)
        returns: site (json)
        '''
        site = self._sites_Data[str(site_id)]
        return site
```

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 02 Code and Outputs
Jake Meyer
03/27/2023

```
    def _get_measurements(self, person_id):
        '''
        Function: Get measurements data
        arguments: person_id (str)
        returns: measurements (json)
        '''
        measurements = []
        # Use for loop to get measurements data added into array
        for measurement in self._measurements_Data.values():
            if str(measurement['person_id']) == str(person_id):
                measurements.extend([measurement])
        return measurements


    def _get_visits(self, visit_id):
        '''
        Function: Get visits and sites data
        arguments: visit_id (str)
        returns: visit (array)
        '''
        visit = [visit for key, visit in self._visit_Data.items() if visit['visit_id'] == visit_id][0]
        site_id = visit['site_id']
        site = self._get_sites(site_id)
        visit['site'] = site
        return visit


    def _load_db(self):
        self._db = TinyDB(self._db_path)
        people = self._people_Data.items()
        for person_id, person_data in people:
            measurements = self._get_measurements(person_id)
            visit_ids = set([measurement['visit_id'] for measurement in measurements])
            visits = []
            for visit_id in visit_ids:
                visit = self._get_visits(visit_id)
                visit['measurements'] = [measurement for measurement in measurements if visit_id ==
measurement['visit_id']]
                visits.append(visit)
            person_data['visits'] = visits
            #print(json.dumps(person_data, indent = 4))
            self._db.insert(person_data)
```

In [2]:

```
db_path = results_dir.joinpath('patient-info.json')
if db_path.exists():
    os.remove(db_path)

db = DocumentDB(db_path)
```

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 02 Code and Outputs
Jake Meyer
03/27/2023

Assignment 2.3 Code (rdbms.ipynb):

```
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.com'):
    s3 = s3fs.S3FileSystem(
        anon=True,
        client_kwargs={
            'endpoint_url': endpoint_url
        }
    )
    return pd.read_csv(s3.open(file_path, mode='rb'))
```

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 = pd.read_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]:

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 02 Code and Outputs
Jake Meyer
03/27/2023

```
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
        );
    """

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

def load_people_table(conn):
    create_people_table(conn)
    df = pd.read_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)
    df = pd.read_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)
    df = pd.read_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]:

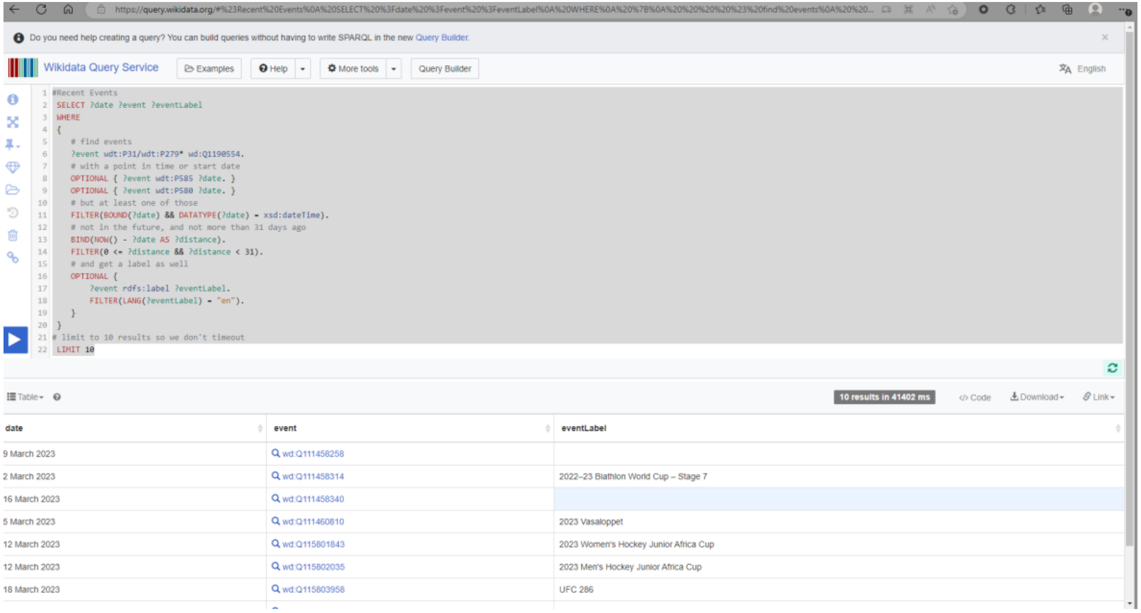
```
db_path = results_dir.joinpath('patient-info.db')
conn = sqlite3.connect(str(db_path))
# TODO: Uncomment once functions completed
load_people_table(conn)
load_sites_table(conn)
load_visits_table(conn)
load_measurements_table(conn)

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

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 02 Code and Outputs
Jake Meyer
03/27/2023

Assignment 2.4 Code:

Wikidata Query Service used to generate the .json file.


https://query.wikidata.org/#%23Recent%20Events%0A%20SELECT%20%3Fdate%20%3Fevent%20%3FeventLabel%0A%20WHERE%0A%20%7B%0A%20%20%20%20%23%20find%20events%0A%20%20...
Wikidata Query Service
Examples
Help
More tools
Query Builder
English
Table
Code
Download
Link

Do you need help creating a query? You can build queries without having to write SPARQL in the new Query Builder.

```
#Recent Events
SELECT ?date ?event ?eventLabel
WHERE
{
   # find events
   ?event wdt:P31/wdt:P279* wd:Q1190554.
   # with a point in time or start date
   OPTIONAL { ?event wdt:P585 ?date. }
   OPTIONAL { ?event wdt:P580 ?date. }
   # but at least one of those
   FILTER(BOUND(?date) && DATATYPE(?date) = xsd:dateTime).
   # not in the future, and not more than 31 days ago
   BIND(NOW() - ?date AS ?distance).
   FILTER(0 <= ?distance && ?distance < 31).
   # and get a label as well
   OPTIONAL {
       ?event rdfs:label ?eventLabel.
       FILTER(LANG(?eventLabel) = "en").
   }
}
# limit to 10 results so we don't timeout
LIMIT 10
```

10 results in 41402 ms

date	event	eventLabel
9 March 2023	wd:Q111458258	
2 March 2023	wd:Q111458314	2022–23 Biathlon World Cup – Stage 7
16 March 2023	wd:Q111458340	
5 March 2023	wd:Q111460810	2023 Vasaloppet
12 March 2023	wd:Q115801843	2023 Women's Hockey Junior Africa Cup
12 March 2023	wd:Q115802035	2023 Men's Hockey Junior Africa Cup
18 March 2023	wd:Q115803958	UFC 286

```
#Recent Events
 SELECT ?date ?event ?eventLabel
 WHERE
 {
    # find events
    ?event wdt:P31/wdt:P279* wd:Q1190554.
    # with a point in time or start date
    OPTIONAL { ?event wdt:P585 ?date. }
    OPTIONAL { ?event wdt:P580 ?date. }
    # but at least one of those
    FILTER(BOUND(?date) && DATATYPE(?date) = xsd:dateTime).
    # not in the future, and not more than 31 days ago
    BIND(NOW() - ?date AS ?distance).
    FILTER(0 <= ?distance && ?distance < 31).
    # and get a label as well
    OPTIONAL {
        ?event rdfs:label ?eventLabel.
        FILTER(LANG(?eventLabel) = "en").
    }
 }
# limit to 10 results so we don't timeout
 LIMIT 10
```