VRANJAN_rdbms

March 26, 2021

[18]:

```
# Course - DSC 650 - Data Mining
# Name - Vikas Ranjan
# Assignment - Week 2 - rdbms
```

[19]:

```
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)

#file_path = 'data/external/tidynomicon'

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'))
```

[20]:

```
# Create and Load

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_m = read_cluster_csv('data/external/tidynomicon/measurements.csv')
    measurements = df_m.values
    c = conn.cursor()
    c.execute('DELETE FROM measurements;') # Delete data if exists
    c.executemany('INSERT INTO measurements VALUES (?,?,?,?)', measurements)
```

[21]:

```
# Create and load people

def create_people_table(conn):
    sql = """
    CREATE TABLE IF NOT EXISTS people (
        people_id text NOT NULL,
        personal_name text,
        family_name text
        );
    """

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

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

[22]:

```
# Create and load sites table

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_s = read_cluster_csv('data/external/tidynomicon/site.csv')
    sites = df_s.values
    c = conn.cursor()
    c.execute('DELETE FROM sites;') # Delete data if exists
    c.executemany('INSERT INTO sites VALUES (?,?,?)', sites)
```

[23]:

```
# Create and load visits

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_v = read_cluster_csv('data/external/tidynomicon/visited.csv')
    visits = df_v.values
    c = conn.cursor()
    c.execute('DELETE FROM visits;') # Delete data if exists
    c.executemany('INSERT INTO visits VALUES (?,?,?)', visits)
```

[24]:

```
# Create DB and Load

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()
```

```
<built-in method fetchall of sqlite3.Cursor object at 0x7f1dce47eb90>
```

[ ]:

```

```