Assignment-5

Redis: 35 points

Submit a PDF with code listing, and screenshots showing outputs of insert(), delete(), and the queries. Screenshots should be uniquely distinguishable for each submission. Be careful of plagiarism from online sources/peers.

Read about redis-py and redis-cli to connect to your cloud redis database.

1. Write the method connect() to create a connection to Redis. [5]
2. Write the methods load_users() and load_scores() to load the data into the redis db. Use appropriate data structures. Provide details of the Redis data structures that you are using. [5]
3. Write the method query1() that returns all the attributes of the user by usr. [5]
4. Write the method query2() that the coordinate (longitude and latitude) of the user by the usr. [5]
5. Write the method query3() that get the keys and last names of the users whose ids do not start with an odd number. [5]
6. Write the method query4() that returns the female in China or Russia with a latitude between 40 and 46. [5]
7. Write the method query5() that gets the email ids of the top 10 players(in terms of score) in leaderboard:2. [5]

```
import csv
import re
from traceback import print_stack
from pyparsing import Regex
import redis
from redis.commands.search.field import TextField, NumericField, TagField
from redis.commands.search.indexDefinition import IndexDefinition
from redis.commands.search.query import Query
# https://redis.readthedocs.io/en/stable/examples.html
import sys



class Redis_Client():
    redis = None
    def __init__(self):
        self.redis = self.redis

    """
        Connect to redis with "host", "port", "db", "username" and "password".
    """
    def connect(self):
        try:
            # TODO:

            print("Connect to Redis.")
```

```
    except:
        print_stack()

"""
    Load the users dataset into Redis DB.
"""
def load_users(self, file):
    result = 0
    # TODO:

    print("Load data for user")
    print(result)
    return result

"""
    Load the scores dataset into Redis DB.
"""
def load_scores(self): #leaderboards for users

    pipe = self.redis.pipeline()

    #open and read from file
    # TODO:

    result = pipe.execute()
    print("load data for scores")
    return result

# """
#     Delete all users in the DB.
# """
# def delete_users(self, hashes):
#     pipe = self.redis.pipeline()
#     for hash in hashes:

#         pipe.delete(hash)

#     result = pipe.execute()
#     return result

# """
#     Erase everything in the DB.
# """
# def delete_all(self):
#     self.redis.flushdb()

"""
Return all the attribute of the user by usr
"""
def query1(self, usr):
```

```
        print("Executing query 1.")
        # TODO:

        # print(result)
        # return result

    """
    Return the coordinate (longitude and latitude) of the user by the usr.
    """
    def query2(self, usr):
        print("Executing query 2.")
        #TODO:

        # print(coordinates)
        # return coordinates

    """
    Get the keys and last names of the users whose ids do not start with an odd number.
    We want to search for a subset of keyspace with the cursor at 1280.

    To avoid the searching of the entire keyspace, we only want to go through only a small number of elements per call.
    That is, we expect to only search through the subset of the keyspace, and then incrementally iterate the next keyspace only if needed.
    (https://redis.io/commands/scan/). You can test the scan query in the redis-cli.
    """
    def query3(self):
        print("Executing query 3.")
        # TODO:Get the keys and last names of the users whose ids do not start with an odd number. Searching for the keyspace start at cursor 1280.

        #print(userids, result_lastnames)
        #return userids, result_lastnames

    """
    Return the female in China or Russia with the latitude between 40 and 46.
    """
    def query4(self):
        print("Executing query 4.")
        # TODO: In order to query attributes other than the primary key, you need to first create a secondary index in Redis with the following specification:
        # gender(text), country(tag), latitude(Numeric), first_name(text).


        # for doc in result.docs: # result is a returned object from redis search()
        #     print(doc)
        # return result #returns a list of document objects

    """
    Get the email ids of the top 10 players(in terms of score) in leaderboard:2
```

```
    """
    def query5(self):
        print("Executing query 5.")
        #TODO:

        # print(result)
        # return result
# git@github.com:redis-developer/redis-datasets.git

rs = Redis_Client()
rs.connect()
#rs.load_users("../datasets/users.txt")
#rs.load_scores()
rs.query1(299)
rs.query2(2836)
rs.query3()
rs.query4()
rs.query5()
```