Query Processing

Submit Assignment

Due Tuesday by 11:59pm   **Points** 20
Submitting a text entry box or a file upload
Available Mar 24 at 12am - Mar 30 at 11:59pm 7 days

The required task is to build a simplified query processor that accesses data from the partitioned ratings table.

Input Data: - Same as in Assignment 1 i.e. ratings.dat file.
(http://initd.org/psycopg/docs/)

   (http://initd.org/psycopg/docs/) For more information on Psycopg, click here.
(http://initd.org/psycopg/docs/)   **(http://initd.org/psycopg/docs/)**

Required Task:

Below are the steps you need to follow to fulfill this assignment:

RangeQuery() -

o Implement a Python function RangeQuery that takes as input: (1) Ratings table stored in PostgreSQL, (2) RatingMinValue (3) RatingMaxValue (4) openconnection

o Please note that the RangeQuery would not use ratings table but it would use the range and round robin partitions of the ratings table.

o RangeQuery() then returns all tuples for which the rating value is larger than or equal to RatingMinValue and less than or equal to RatingMaxValue.

o The returned tuples should be stored in a text file, named RangeQueryOut.txt (in the same directory where Interface.py is present) such that each line represents a tuple that

has the following format such that PartitionName represents the full name of the partition i.e. RangeRatingsPart1 or RoundRobinRatingsPart4 etc. in which this tuple resides.

Example:

PartitionName, UserID, MovieID, Rating

RangeRatingsPart0,1,377,0.5

RoundRobinRatingsPart1,1,377,0.5

o Note: Please use ',' (COMMA, no space character) as delimiter between PartitionName, UserID, MovieID and Rating.

PointQuery() -

o Implement a Python function PointQuery that takes as input: (1) Ratings table stored in PostgreSQL, (2) RatingValue. (3) openconnection

o Please note that the PointQuery would not use ratings table but it would use the range and round robin partitions of the ratings table.

o PointQuery() then returns all tuples for which the rating value is equal to

RatingValue.

o The returned tuples should be stored in a text file, named PointQueryOut.txt (in the same directory where Interface.py is present) such that each line represents a tuple that has the following format such that PartitionName represents the full name of the partition i.e. RangeRatingsPart1 or RoundRobinRatingsPart4 etc. in which this tuple resides.

Example

PartitionName, UserID, MovieID, Rating

RangeRatingsPart3,23,459,3.5

RoundRobinRatingsPart4,31,221,0

o Note: Please use ',' (COMMA) as delimiter between PartitionName, UserID,

MovieID and Rating.

Please use the function signature exactly same as mentioned in Interface.py .

Naming Convention to be followed strictly:

Database name - dds_assignment

Name of Rating table - ratings

Postgres User name - postgres

Postgres password - 1234

Name of the Range Query Output file - RangeQueryOut.txt

Name of the Point Query Output file - PointQueryOut.txt

How to use tester.py:

Implement your functions in Interface.py and once done, use the tester again to generate the output.

DO NOT CHANGE tester.py and Assignment1.py. Changing anyone of these would cause problems and the system will stop working, and may lead to deduction of marks.

PLEASE KEEP IN MIND, this tester is just for your help. For grading purpose, an additional set of test cases would be used. It will try to break your code. So, please provide the functionalities accordingly, so that it handles all possible scenarios.

Instructions for Assignment: -

Please follow these instructions closely else Marks will be deducted.

1. Please follow the function signature as provided in the Interface.py.

2. Please use the same database name, table name, user name and password as provided in the assignment to keep it consistent.

3. Please make sure to run the provided tester and make sure there is no indentation error. In case of any compilation error, 0 marks will be given.

Submission Instructions: -

Only submit the Interface.py file. Do not change the file name. Do not put it into a folder or upload a zip file.

We provide a virtual machine that has the testing environment and an installed PostgreSQL.

OS username: user

OS password: user

Postgres username: postgres

Posgres password: 1234

You can download it and use any VM software such as VirtualBox to import it:

https://drive.google.com/file/d/1EBlmGZmqDQ8XGTuiHPoqP7XE2ZykAXkX/view?usp=sharing

You will use the following files within your assignment. These files are used to test your Interface.py

1. tester.py: Test your interface.py using this tester. Run it using "python tester.py".

- **tester.py** ↓ **(https://canvas.asu.edu/courses/75440/files/26129810/download?**
  download_frd=1)

  Minimize File Preview

  Page < **2** > of 2 | | | ZOOM | + | | ↗ |

```
        print "Delete tables"
        Assignment1.deleteTables('all', con);

        # Loading Ratings table
        print "Creating and Loading the ratings table"
        Assignment1.loadRatings('ratings', 'test_data.txt', con);

        # Doing Range Partition
        print "Doing the Range Partitions"
        Assignment1.rangePartition('ratings', 5, con);

        # Doing Round Robin Partition
        print "Doing the Round Robin Partitions"
        Assignment1.roundRobinPartition('ratings', 5, con);

        # Deleting Ratings Table because Point Query and Range Query should not use
ratings table instead they should use partitions.
        Assignment1.deleteTables('ratings', con);

        # Calling RangeQuery
        print "Performing Range Query"
        Assignment2.RangeQuery('ratings', 1.5, 3.5, con);
        #Assignment2.RangeQuery('ratings',1,4,con);

        # Calling PointQuery
        print "Performing Point Query"
        Assignment2.PointQuery('ratings', 4.5, con);
        #Assignment2.PointQuery('ratings',2,con);

        # Deleting All Tables
```

```
        Assignment1.deleteTables('all', con);

        if con:
            con.close()

    except Exception as detail:
        traceback.print_exc()
```

2. test_data.txt: some test data

- **test_data.txt** ↓ **(https://canvas.asu.edu/courses/75440/files/26129871/download?**
  download_frd=1)

3. Interface.py: Implement the interface in Interface.py

- **Interface.py** ↓ **(https://canvas.asu.edu/courses/75440/files/26129947/download?
  download_frd=1)**

  Minimize File Preview

  ZOOM

```
#!/usr/bin/python2.7
#
# Assignment2 Interface
#

import psycopg2
import os
import sys
# Donot close the connection inside this file i.e. do not perform
openconnection.close()
def RangeQuery(ratingsTableName, ratingMinValue, ratingMaxValue, openconnection):
    pass


def PointQuery(ratingsTableName, ratingValue, openconnection):
    pass


def writeToFile(filename, rows):
    f = open(filename, 'w')
    for line in rows:
        f.write(','.join(str(s) for s in line))
        f.write('\n')
    f.close()
```

4. Assignment1.py: the correct answer of Assignment 1 with slight modifications

- **Assignment1.py** ↓ **(https://canvas.asu.edu/courses/75440/files/26129873/download? download_frd=1)**

Sample output ↓ **(https://canvas.asu.edu/courses/75440/files/26129892/download?download_frd=1)**

- **Assignment1.py** ↓ **(https://canvas.asu.edu/courses/75440/files/26129873/download? download_frd=1)**