Ira A. Fulton Schools of Engineering
Arizona State University

CSE 511: Data Processing at Scale

NoSQL Project

Purpose

You must carry out transformations and actions on a No-SQL database using python files for this project. Here, data is not saved in relations like it is in standard relational databases.

Objectives

Learners will be able to:

- Recognize fundamental concepts and features of NoSQL databases.
- Demonstrate retrieving data from NoSQL databases.
- Explain how to convert mathematical concepts into code.

Technology Requirements

- Python 3.10
- cython 3.0.0
- unqlite 0.9.6
- Jupyter Notebook

Project Description

Through this project, you will gain experience working with NoSQL databases and understand how to use Python to conduct several fundamental operations on them. Unlike conventional relational databases, NoSQL allows data to be stored more freely without the use of rigid schemas. Additionally, they are capable of handling massive amounts of data quickly.

Directions

Before getting started, download the following project files from the course

1. project1.ipynb

2. sample.db
3. CSE 511_NoSQL Project_Additional Test Cases.docx

Implement the functions provided in the Jupyter Notebook (**project1.ipynb**) to perform the operations as listed below. You may use the included **sample.db** to test your functions with the Python code provided.

A. **FindBusinessBasedOnCity(cityToSearch, saveLocation1, collection)** – This function searches the 'collection' given to find all the business present in the city provided in 'cityToSearch' and save it to 'saveLocation1'. For each business you find, you should store the name, full address, city, and state of the business in the following format. Each line of the saved file will contain: Name$FullAddress$City$State. ($ is the separator and must be present.)

B. **FindBusinessBasedOnLocation(categoriesToSearch, myLocation, maxDistance, saveLocation2, collection)** – This function searches the 'collection' given to find the name of **all** the businesses present in the 'maxDistance' from the given 'myLocation' (please use the distance algorithm attached in the Coursera Project Overview page titled "...sample") and save them to 'saveLocation2'. Each line of the output file will contain the name of the business only.

   Note: The function should search for **all** the categories given to find the business name based on location. For example, consider categories to search to be ["Food","Seafood"]. The function should return all the business names that have Food or Seafood.

Distance Algorithm

A distance algorithm will need to be used. Given two pairs of latitude and longitude as [lat2, lon2] and [lat1, lon1], you can calculate the distance between them using the formula given below:

DistanceFunction(lat2, lon2, lat1, lon1)

- var R = 3959; // miles
- var φ1 = lat1.toRadians();
- var φ2 = lat2.toRadians();
- var Δφ = (lat2-lat1).toRadians();
- var Δλ = (lon2-lon1).toRadians();

- var a = Math.sin(Δφ/2) * Math.sin(Δφ/2) + Math.cos(φ1) * Math.cos(φ2) * Math.sin(Δλ/2) * Math.sin(Δλ/2);
- var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
- var d = R * c;
- d is the distance between the given pair of latitude and longitude. The distance is in **miles.**

Reference: http://www.movable-type.co.uk/scripts/latlong.html

Test Cases

Additional test cases are provided in the file attached in the Course Project Overview page.

Important Note

While working on the project1.ipynb file,

1. Do not remove the comment provided in the cell where your function is defined for autograder to run your functions.

```
# Graded Cell: test_function
# DO NOT REMOVE THE ABOVE LINE


def FindBusinessBasedOnCity(cityToSearch,saveLocation1,collection):
    pass

def FindBusinessBasedOnLocation(categoriesToSearch, myLocation, maxDistance, saveLocation2, collection):
    pass
```

2. Do not split the functions into different cells else the grader will not run your functions.
3. Do not change the function names.

Report

In addition to your ipynb file, write a **2-3 page report detailing your work on the project**. Your report should be a PDF titled with “Last Name_First Name_CSE511_NoSQL Project Report.” Your report should include:

1. **Reflection**: How did you approach the project? What did you specifically do? (Write two function codes along with distance function and explain about them)
2. **Lessons Learned**: What did you learn by doing this project?

3. **Output**: Screenshot shows parts of your TXT output files.
4. **Result**: Result of your code by passing the test cases.

Submission Directions for Project Deliverables

You must submit each of your NoSQL Project deliverables through Gradescope. Carefully review submission directions outlined in this overview document in order to correctly earn credit for your work. Learners may not email or use other means to submit any assignment or project for review, including feedback, and grading.

The NoSQL Project includes two (2) deliverables:

- **Notebook File:** Submit your completed project1.ipynb file to Gradescope.
- **Report**: Submit your PDF report to Gradescope titled “Last Name_First Name_CSE511_NoSQL Project Report”.

Gradescope Submission

Your submission will be reviewed by the course team and then, after the due date has passed, your score will be populated from Gradescope into your grade. You will receive credit for the report if you covered all the mentioned parts.

1. Go to the Canvas Assignment, “**Submission: NoSQL Project**”.
2. Click the "**Load Submission…in new window**" button.
3. Once in Gradescope, select the project titled "**NoSQL Project**" and a pop-up window will appear.
4. In the pop-up window, submit your “**project1.ipynb**” file and your **PDF report** for grading.
5. If needed: to resubmit the project in Gradescope:
    a. Return to the Canvas submission and open Gradescope.
    b. You will be navigated to the "Autograder Results" page (if it is not your first submission).
    c. Click the "**Resubmit**" button on the bottom right corner of the page and repeat the process from Step 3.

Evaluation

There is one test case each for the two functions. If some part of your output data for either of the functions is incorrect, you will receive a partial score. If either of the two functions is correct, you will receive a full score. **If the submission fails, you will see the corresponding error logs that indicate *where* the error occurred.**

Common Errors:

1. Error: Submission was not a well-formed Jupyter Notebook file.
2. Save the file as IPYNB directly from Jupyter Notebook rather than saving it in another format and then replacing their properties as it makes the file unstable.
3. Runtime errors happen in your submission invalid syntax (<string>, line 22)
4. Error: Distance function, FindBusinessBasedOnCity, FindBusinessBasedOnLocation
5. Your submission did not define the FindBusinessBasedOnLocation function,
6. Runtime errors happen in your submission name 'data' is not defined.
7. The original function must remain untouched, and students must write code inside the function definition block.
8. Error: No such file in the directory: 'output_loc.txt'. Make sure your functions generate respective output files

Learner Checklist

Prior to submitting, read through the Learner Checklist to ensure you are ready to submit your best work.

- ☐ Did you title your file correctly and convert it into a single **.ipynb file**?
- ☐ Did you include your legal first and last name in the designated area on the **report**?
- ☐ Did you title your **report document** correctly and convert it into a single **pdf**?
    - **Last Name_First Name_CSE###_Name of Project**
- ☐ Did you answer all of the questions to the best of your ability?
- ☐ Did you make sure your answers directly address the prompt(s) in an organized manner that is easy to follow?

- ☐ Did you self-assess your open-ended responses using the rubric and make any necessary revisions?
- ☐ Did you proofread your work?