IST769 Homework Submission Template

Basic Information

Your Name: Mark Roberts  
Your SUID: 598273961  
Your Email: mrober04@syr.edu  
Date Due: August 31, 2021   
Homework #: 8

Your Answers:

1. From Impala, use the two external tables **weblogs** created from **clickstream/logs\_noheader** and **iplookup** created from **clickstream/iplookup\_noheader** you created in the previous assignment to complete this question. Use the impala shell to answer the following questions, making sure to include the SELECT query you used to answer it.
   1. How many GET and POST requests are there in the weblogs?
   2. How many requests have Mac in the user agent?

How many hosts (ip addresses) have Mac in the user agent?

1. Write a MongoDB Query to retrieve Country name, population, and capital for all countries in the collection.

|  |
| --- |
| CODE |
| docker-compose up -d  docker-compose exec mongo mongoimport -u admin -p pass --authenticationDatabase=admin -d demo -c countries--file=Europe.json --jsonarray  docker-compose exec mongo mongo -u admin -p pass --authenticationDatabase=admin  db.adminCommand({listDatabases:1, nameOnly:true})  use demo  db.runCommand({listCollections: 1.0, nameOnly:true});  db.countries.findOne();  db.countries.findOne({},{name:1,population:1,capital:1});  db.countries.find({),{name:1,population:1,capital:1}); |
| SCREENSHOT/OUTPUT |
|  |

1. Write a MongoDB Query to retrieve Country name, population, and capital for all countries with a population under 500,000 sorted by population.

|  |
| --- |
| CODE |
| db.countries.find({population: {$lt:500000}},{name:1,population:1,capital:1}).sort({population:1}) |
| SCREENSHOT/OUTPUT |
|  |

1. Use the**. explain(“executionStats”)** method to analyze the query you wrote in the previous step. Write an index to improve the performance of the query, then perform another explain to demonstrate it worked. Include the code of the index you wrote, the and the relevant output of the execution stats which demonstrate the index is being used.

|  |
| --- |
| CODE |
| db.countries.find({population: {$lt:500000}},{name:1,population:1,capital:1}).sort({population:-1}).explain("executionStats")  db.countries.createIndes({population:1}) |
| SCREENSHOT/OUTPUT |
|  |

1. Select the most appropriate Redis data structure to store the following information:

|  |  |  |  |
| --- | --- | --- | --- |
| Product ID | Name | Qty On Hand | Unit Price |
| 1 | Apple | 7 | 2.49 |
| 2 | Banana | 12 | 1.99 |
| 3 | Cherry | 9 | 4.99 |

1. Execute the commands to store this information in Redis. Make sure to namespace your key and each of the fields should be retrievable under the key used.

|  |
| --- |
| CODE |
| cd redis  docker-compose up -d  docker-compose exec redis redis-cli  hmset product:1 Name "Apple" QtyOnHand 7 UnitPrice "2.49"  hmset product:1 Name "Banana" QtyOnHand 12 UnitPrice "1.99"  hmset product:1 Name "Cherry" QtyOnHand 9 UnitPrice "4.99"  hmget product:1 Name QtyOnHand UnitPrice  hmget product:2 Name QtyOnHand UnitPrice  hmget product:3 Name QtyOnHand UnitPrice  hexists product:3 Name  hgetall product:2 |
| SCREENSHOT/OUTPUT |
|  |

Select the most appropriate Redis data structure to store the following information:

The 2018 Golden Snowball Competition for the Upstate NY City with the Highest Snowfall. Scores updated hourly.

|  |  |  |  |
| --- | --- | --- | --- |
| City | Syracuse | Rochester | Buffalo |
| Snowfall Inches | 97 | 68 | 84 |

Execute the commands to store this information in Redis. Make sure to namespace your key and each of the snowfall values should be updatable. For example, you should be able to add 10 inches to Buffalo to make it 94. You should be able to display the information upon request.

|  |
| --- |
| CODE |
| zadd competition:goldensnowball:2018 97 Syracuse 68 Rochester 84 Buffalo  zrange competition:goldensnowball:2018 0 -1  zrange competition:goldensnowball:2018 0 -1 withscores  zadd competition:goldensnowball:2018 incr 10 Buffalo  zrange competition:goldensnowball:2019 0 -1 withscores |
| SCREENSHOT/OUTPUT |
|  |