# IST769 Homework 8 Submission

## Basic Information

Your Name: Srihari Busam   
Your SUID: sbusam  
Your Email: sbusam@syr.edu  
Date Due: 11/26/2021  
Homework #: 8

## QUESTIONS:

1. Write a MongoDB Query to retrieve Country name, population, and capital for all countries in the collection.
2. Write a MongoDB Query to retrieve Country name, population, and capital for all countries with a population under 500,000 sorted by population.
3. 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.
4. 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 |

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.

1. 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.

## ANSWERS:

### **ANSWER 1:**

|  |
| --- |
| Mongodb shell code |
| use demo  db.countries.find({},{"name":1, "capital":1, "population":1}) |

Screen shot to show the projection of name, capital and population from mongodb query

Text

Description automatically generated

### **ANSWER 2:**

|  |
| --- |
| Mongodb shell code |
| db.countries.find({"population": {$lt: 500000}},{"name":1, "capital":1, "population":1}).sort({"population": 1}) |

Screenshot:

Text

Description automatically generated

### **ANSWER3:**

|  |
| --- |
| Mongodb shell code |
| db.countries.find({"population": {$lt: 500000}},{"name":1, "capital":1, "population":1}).sort({"population": 1}).explain("executionStats")  db.countries.createIndex({population:1})  db.countries.find({"population": {$lt: 500000}},{"name":1, "capital":1, "population":1}).sort({"population": 1}).explain("executionStats") |

Before creation of index: (shows collection scan)

Screenshot1

Text

Description automatically generated

Screenshot:2

Text

Description automatically generated

Creation of index:

Text

Description automatically generated

After index creation the plan: ( Uses index scan IXSCAN):

Screenshot1

Text

Description automatically generated

Screenshot 2:

Text

Description automatically generated

### **ANSWER 4:**

A hash would be the right data structure to store the tablular data presented for the problem 3.

|  |
| --- |
| Redis commands |
| hmset product:1 Name Apple QtyOnHand 7 UnitPrice "2.49"  hmset product:2 Name Banana QtyOnHand 12 UnitPrice "1.99"  hmset product:3 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  hgetall product:1 |

Screenshot:

Text

Description automatically generated

### **ANSWER 5:**

A sorted set is the right redis data structure to store the leaderboard info.

|  |
| --- |
| Redis commands |
| zadd competition:golden-snow-ball:2018 97 Syracuse 68 Rochester 84 Buffalo  zrange competition:golden-snow-ball:2018 0 -1  zrange competition:golden-snow-ball:2018 0 -1 withscores  zadd competition:golden-snow-ball:2018 incr 10 Buffalo  zrange competition:golden-snow-ball:2018 0 -1 withscores |

Screenshot:

Text

Description automatically generated