Project 5 CS585/DS 503: Big Data Management Spring 2019

"Working with NoSQL Data Engine"

Total Points: 100 points
Given Out: Monday, 8th April, 2019 (REMEMBER, NO CLASS on 15th!)
Due Date: Saturday, 20th April, 2019 (5:00PM)

Teams: Project is to be done in teams, called Project5-team.
Team members assigned by CS585/DS503 staff.

Project Scope and Submission

You will again work with the MongoDB NoSQL data engine. The task is to manage, query, and transform unstructured data using MongoDB. Please see the lecture notes as well as on-line resources such as the MongoDB manual. If you find a resource that you like, please also share it with all students on CANVAS. You have now used MongoDB for project 4, which has prepared you to get familiar with MongoDB. The same setup should work for you to complete this 5th and last project.

Submission Mechanism

You will submit below as one **single zip file** via CANVAS, namely:

1. This should contain a single text file containing your **MongoDB statements**.
2. This should also include a **report (.pdf or .doc)** that shows your results for each query, explains your solution strategy, and choices and/or issues with your solution using MongoDB.
3. In this report, you should describe **your team's methodology,** i.e., how often you meet, how you communicated, how you shared the work and finally how you created an agreed-up project deliverable. **Relative tasks accomplished** by each of your team members are to be specified if the work was not done in very close collaboration.
4. E**ach team member independently** submits peer team comments to CS585/DS503 staff via https://goo.gl/forms/ip0Yw7YbSDG5sXAe2. This is your personal assessment of the team's joint teamwork, your own contributions and effort as well as the contributions and effort of your team member to this project. *These comments will be treated confidentially. You will not be given a grade, until you submit your survey. You are also invited to talk to the CS585/DS503 about your team and/or project, as needed*

A Note on Expected Teamwork: *It is expected that each team member would first produce a draft of the full solutions for ALL problems to the best of their ability and in a timely fashion. That then as a team you discuss your solutions, agree on the overall best answer to each problem, and submit a final result that everyone in the team agrees to. Working with your team member is part of the assignment.*

Problem 1: Query Specification over Unstructured Data in MongoDB. [50 points]

Create a MongoDB database, a collection “famous-people”, and insert into this collection 10 documents from this link: https://docs.mongodb.com/manual/reference/bios-example-collection/ Then apply the following update operations to this collection.

1) Write an aggregation query that groups people by the award name, i.e., the “award” field inside the “awards” array, and reports the number of times each award has been given. (Hint: Use Map-Reduce mechanism)

2) Write an aggregation query that groups people by the award name, i.e., the “award” field inside the “awards” array, and for each award reports an array of all the years for when this award has been given (Hint: Use map-reduce or use aggregation mechanism)

3) Write an aggregation query that groups by the birth year, i.e., the year within the “birth” field, and for each year it reports both a total count of people with that same birth year as well as an array of _ids of people with that same birth year.

4) Report the document with the smallest and largest _ids. As strategy, you may first want to find the values of the smallest and largest _id first, and only then find and return their corresponding documents.

5) Search for and report all documents containing “Turing” as text substring. (Hint: You could use $text operator to represent the string search).

6) Search for and report all documents containing either “Turing” or “National Medal” or both as text substring. (Hint: Use $text operator to represent the string search).

7) Search for and report all documents containing both “Turing” and “National Medal” as text substring. (Hint: Use $text operator to represent the string search).

Problem 2. Querying Parent-Child Relationships in MongoDB. [50 pts]

1) Model the records and relationships in Figure 1 below using the Parent-Referencing model (See slide deck or MongoDB tutorial). You are given only the root node, i.e., _id = “Books”.; now write a query that reports the height of the tree. (Hint: Test your query on multiple nodes. Also, for Books, the result should be 4).

2)Model the records and relationships in Figure 1 using the Parent-Referencing model (See slide deck or MongoDB tutorial). Write a query to report the ancestors of “DBM”. The output should be an array containing values:

```
[ {Name: “Databases”, Level: 1},
  {Name: “Programming”, Level: 2},
  {Name: “Books”, Level: 3}]
```

(“Level” is the distance from “dbm” node to the respective ancestor node.)

3) Assume we model the records and relationships in Figure 1 using the Child-Referencing model. Write a query to report the direct parent of “MongoDB”.

4) Assume we model the records and relationships in Figure 1 using the Child-Referencing model. Write a query to report the full set of descendants of “Programming”. The output should be an array containing values [“Languages”, “Databases”, “MongoDB”, “dbm”]

5) Assume we model the records and relationships in Figure 1 using the Child-Referencing model. Write a query to report the siblings “Languages”.

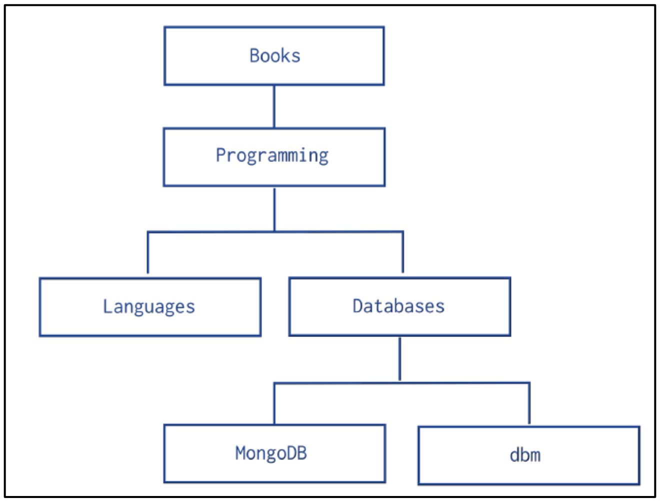

Figure: Tree Structure Relationships

-- The end --