

CS585: Big Data Management

Assignment 4 **(MongoDB)**

Total Points: 65

Release Date: 11/06/2018

Due Date: 11/20/2018 (@ 11:59pm)

References: Lecture notes & MongoDB Manual

Note: The MongoDB version in the Virtual Machine you have is outdated. So some of the features (commands) that we cover in class may not work in the given version. One possibility is to deploy a free cloud-based instance from this link: <https://www.mongodb.com/download-center>

Question 1 [30 Points, 2 Points each sub-question]

Your task is to design a MongoDB database and apply some CRUD (Create/Read/Update/Delete) operations as follows.

Create a collection named “test”, and insert into this collection the documents found in this link (10 documents): <http://docs.mongodb.org/manual/reference/bios-example-collection/>

1) Write a CRUD operation(s) that inserts the following new records into the collection:

```
{
  "_id" : 20,
  "name" : {
    "first" : "Alex",
    "last" : "Chen"
  },
  "birth" : ISODate("1933-08-27T04:00:00Z"),
  "death" : ISODate("1984-11-07T04:00:00Z"),
  "contribs" : [
    "C++",
    "Simula"
  ],
  "awards" : [
    {
      "award" : "WPI Award",
      "year" : 1977,
      "by" : "WPI"
    }
  ]
}
```

```
{
  "_id" : 30,
  "name" : {
    "first" : "David",
    "last" : "Mark"
  },
  "birth" : ISODate("1911-04-12T04:00:00Z"),
  "death" : ISODate("2000-11-07T04:00:00Z"),
  "contribs" : [
    "C++",
    "FP",
    "Lisp"
  ],
  "awards" : [
    {
      "award" : "WPI Award",
      "year" : 1963,
      "by" : "WPI"
    },
    {
      "award" : "Turing Award",
      "year" : 1966,
      "by" : "ACM"
    }
  ]
}
```

- 2) Report all documents of people who got a “Turing Award” after 1976
- 3) Report all documents of people who got less than 3 awards or have contribution in “FP”
- 4) Report the contributions of “Dennis Ritchie” (only report the name and the contribution array)
- 5) Update the document of “Guido van Rossum” to add “OOP” to the contribution list.

- 6) Insert a new field of type array, called “comments”, into the document of “Alex Chen” storing the following comments: “He taught in 3 universities”, “died from cancer”, “lived in CA”
- 7) For each contribution by “Alex Chen”, say X, list the people’s names (first and last) who have contribution X. E.g., Alex Chen has two contributions in “C++” and “Simula”. Then, the output should be similar to:
 - a. {Contribution: “C++”,
 People: [{first: “Alex”, last: “Chen”}, {first: “David”, last: “Mark”}]},
 {Contribution: “Simula”,
}
- 8) Report the distinct organization that gave awards. This information can be found in the “by” field inside the “awards” array. The output should be an array of the distinct values, e.g., [“wpi”, “acm”, ...]
- 9) Delete from all documents the “death” field.
- 10) Delete from all documents any award given on 2011.
- 11) Update the award of document _id = 30, which is given by WPI, and set the year to 1965.
- 12) Add (copy) all the contributions of document _id = 3 to that of document _id = 30
- 13) Report only the names (first and last) of those individuals who won at least two awards in 2001.
- 14) Report the document with the largest id. First, you need to find the largest _id (using a CRUD statement), and then use that to report the corresponding document.
- 15) Report only one document where one of the awards is given by “ACM”.

Question 2 [15 Points, 5 Points each sub-question]

As a continuation over the dataset used in Question 1, answer the following aggregation queries:

- 1) Write an aggregation query that group by the award name, i.e., the “award” field inside the “awards” array, and reports the count of each award. (Use Map-Reduce mechanism)
- 2) Write an aggregation query that groups by the birth year, i.e., the year within the “birth” field, and report an array of _ids for each birth year. (Use Aggregate mechanism)
- 3) Report the document with the smallest and largest _ids. You first need to find the values of the smallest and largest, and then report their documents.

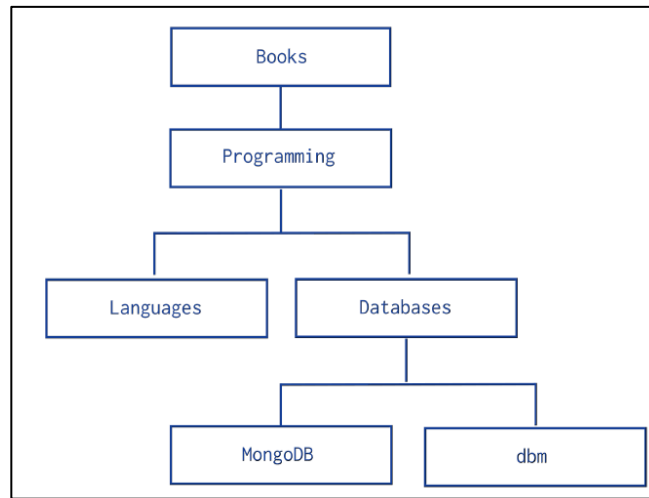


Figure 1: Tree Structure Relationships

Question 3 [20 Points, 5 Points each sub-question]

1) Assume we model the records and relationships in Figure 1 using the Parent-Referencing model (Slide 4 in MongoDB-3). Write a query to report the ancestors of “MongoDB”. The output should be an array containing values [{Name: “Databases”, Level: 1},
{Name: “Programming”, Level: 2},
{Name: “Books”, Level: 3}]

** Note: “Level” is the distance from “MongoDB” node to the other node. It should be computed in your code*

2) Assume we model the records and relationships in Figure 1 using the Parent-Referencing model (Slide 4 in MongoDB-3). You are given only the root node, i.e., `_id = “Books”`, write a query that reports the height of the tree. (It should be 4 in our case).

3) Assume we model the records and relationships in Figure 1 using the Child-Referencing model (Slide 9 in MongoDB-3). Write a query to report the parent of “dbm”.

4) Assume we model the records and relationships in Figure 1 using the Child-Referencing model (Slide 9 in MongoDB-3). Write a query to report the descendants of “Books”. The output should be an array containing values [“Programming”, “Languages”, “Databases”, “MongoDB”, “dbm”]

Submission Mechanism

Submit all your CRUD statements in a single file electronically using Canvas system.