

CSCI 3287 Design and Analysis of Data Systems

Homework #6 -- MongoDB Lab

Overview

This Project is worth 120 points (out of 1000) toward your final grade. It is due on Sunday April 21st, at 11:59 p.m. **No extension will be allowed for this assignment.** Your assignment submission should be a document saved and submitted as a PDF file via the link found in the assignment section of “Week THIRTEEN” in Moodle which is the same place where you found this file.

This assignment will give you hands-on practice in working with the MongoDB “NoSQL” database software.

Objectives

1. Download and install MongoDB
2. Create a MongoDB database to store a collection of documents
3. Load a large amount of document-based data into the collection
4. Query the document collection to research a topic and answer questions

Introduction:

The following links are helpful for an introduction MongoDB concepts and learning the basics of the MongoDB query language.

1. https://www.tutorialspoint.com/mongodb/mongodb_overview.htm : This link provides a basic overview of MongoDB and the basic queries for inserting documents, updating documents, query documents etc.
2. <https://www.guru99.com/mongodb-tutorials.html#1> : This is a beginner tutorial and also has links for installing and running mongoDB with images which may be helpful for students.
3. <https://docs.mongodb.com/manual/introduction/> : The official MongoDB documentation page.

Installation:

The following links are helpful for completing the installation of mongoDB:

Mac System:

1. <https://docs.mongodb.com/manual/tutorial/install-mongodb-on-os-x/> : I have personally tried the installation and it works without issues. Hint: the **.bashrc** file in mac is `/Users/apple/.bash_profile`.

Windows System:

2. <https://docs.mongodb.com/tutorials/install-mongodb-on-windows/> : This is the official documentation for windows installation.

Linux System:

3. <https://docs.mongodb.com/manual/administration/install-on-linux/> : The official installation for linux installation.

<https://docs.mongodb.com/manual/installation/> : This link provides greater detail about the operating systems supported by mongoDB.

Students should install the community edition of mongoDB as it is freely available.

In the following 8 examples in Task 1, you will have the opportunity to play with different basic operations in MongoDB. You can then use these tools to learn with a real-world data set, and answer the questions below based on the data set in Task 2.

Task 1: Basic operations in MongoDB

1. Create a database: `use DATABASE_NAME`

Ex: `>use new_mongo_db`

The above command will create a database if it does not exist and/or use the database if it already exists.

Replace `DATABASE_NAME` with the name of the database you want to create.

2. Drop a database: `db.dropDatabase()`

Ex: `>use new_mongo_db`
`switched to db new_mongo_db`
`>db.dropDatabase()`

- i. First you need to switch to the database that has to be dropped. Then use the above command to drop that database.

3. Creating a collection: `db.createCollection(name, options)`

Ex:

```
>use new_mongo_db
switched to db new_mongo_db
>db.createCollection("test_collection")
{ "ok" : 1 }
```

- i. The above command creates the collection. But providing some initial additional options will be highly useful.

1.

```
>db.createCollection("mycol",{ capped : true, autoIndexId : true, size :
6142800, max : 10000 })
{ "ok" : 1 }
```

2. For more information on the options, please check the following link.

<https://docs.mongodb.com/manual/reference/method/db.createCollection/>

- ii. In mongoDB, it is not necessary to create a collection. When a new document is inserted, mongoDB creates a collection automatically.

4. Dropping a collection: `db.COLLECTION_NAME.drop()`

Ex:

```
>use new_mongo_db
switched to db new_mongo_db
>db.test_collection.drop()
True
```

- i. First go to the selected database and then use the above command to delete the collection
- ii.

5. Insert a document: `db.COLLECTION_NAME.insert(document)`

Ex:

```
> db.test_collection.insert({
  title: "Mongo Db practice",
  description: "this is my first MongoDB document"
})
```

Replace the `COLLECTION_NAME` with the name of the collection of your choice.

6. Query a document: `db.COLLECTION_NAME.find()`

- i.

Ex:

```
>db.test_collection.find().pretty()
```

- ii. The above query will display the documents present in the collection.

7. Update a document: `db.test_collection.update(SELECTION_CRITERIA, UPDATED_DATA)`

Ex:

```
>db.test_collection.update({'title':'MongoDB
practice'},{$set:{ 'title':'New MongoDB practice'}})
```

- i. The above example is used to update the documents that contain 'title' as 'MongoDB practice' to 'New MongoDB practice'

8. Delete a Document: `db.COLLECTION_NAME.remove(DELETION_CRITTERIA)`

- i. Remove only one record:
 1. `db.test_collection.remove({ status : "P" },1)`
 2. Here the first document which has this key value pair will be deleted.
- ii. Remove all records matching a condition:
 1. `db.test_collection.remove({ status : "P" })`
 2. Here all the documents which have this key value pair will be deleted.

Task 2: Use real-world data set to answer the following question

1. Download the JSON dataset from Moodle “primer-data.json”
2. From your terminal, type the following command:

```
mongoimport --db test --collection restaurants --drop --file  
~/Desktop/primer-data.json
```

The above command converts the json file and stores it as a set of documents with the collection name of “restaurants”. NOTE: this command is run in the OS Terminal Shell, not within the Mongo interface.

Answer the following questions by writing queries and displaying the results.

- (1) List the restaurants that have the string “Ice Cream” in their name. Return only the restaurant id and name. (HINT: use Regex.)
- (2) Find the names of all restaurants that serve either Italian or American cuisine and are located in the Brooklyn borough.
- (3) Return a list of boroughs ranked by the number of American restaurants in the borough. That is, for each borough, find how many restaurants serve American cuisine and print the borough and the number of such restaurants sorted descending by this number. (HINT: use the aggregate method, and use a \$group and a \$sum.)
- (4) Find the top 5 American restaurants in Manhattan that have the highest total score. Return for each restaurant the restaurant’s name and the total score. (HINT: use the aggregate method with \$unwind to parse out the scores array, followed by a \$group and a \$sum.)
- (5) Consider the area of the location field identified by the vertices [-74 , 40.5] , [-74 , 40.7] , [-73.5 , 40.5] and [-73.5 , 40.7]. Find the number of restaurants in this area that have received a grade score (at least one) more than 75. No need to sum scores. (Hint: count the restaurants

whose location coordinates mathematically fall within the bounds set by the coordinates in the question.)

Task 3: Interview grading

Interview grading sessions will be scheduled for Monday – Friday Apr 22nd – 26th

In the interview grading session, you must attend your grading meeting to qualify for points for Task 3. If you miss your meeting with graders/professor (without notifying graders/professor ahead of time with a suitable reason), this may result in a zero grade for Interview Grading part of the assignment. The graders/professor are under no obligation to reschedule your appointment if you miss your meeting, so don't miss this!

During the meeting, you will be asked about 2-3 questions related to this data set. You will need to write/run the query in MongoDB on your computer and show your output to graders/professors in order to get credit.

You should bring your laptop on which you have installed MongoDB to your interview grading session.

Submission Requirement:

Capture screen shots to show evidence of having completed Task 1 described above. Number each screen shot with the number of the assigned operation/task (1 – 8). Assemble (Copy & Paste) all screen shots into a document.

For Task 2, screenshot or write down your MongoDB command and output for each question.

Save your document as a PDF and upload to Moodle.

This is an **individual** assignment; each one must submit your own final deliverable for this assignment. Pair programming is not allowed.