## **COSC 30603 - Lab 7: MongoDB Introduction**

Due Dec 4

## **Name: Hieu Nguyen**

**Installation:**

\*\*The following assumes the usage of the CentOS distribution of Linux. If you are using another distribution, use this link: <https://docs.mongodb.com/manual/administration/install-on-linux/>

Open Terminal and copy/paste the following:

sudo vi /etc/yum.repos.d/mongodb-org.repo

Add the following to the file above:

[mongodb-org-4.2]

name=MongoDB Repository

baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/4.2/x86\_64/

gpgcheck=1

enabled=1

gpgkey=https://www.mongodb.org/static/pgp/server-4.2.asc

Confirm that the MongoDB Repository is in place with:

yum repolist

Now install MongoDB with the yum utility:

sudo yum install mongodb-org

**Starting Mongo Daemon:**

sudo systemctl start mongod

Confirm the Daemon is listening with:

sudo tail /var/log/mongodb/mongod.log

Output should be:

[initandlisten] waiting for connections on port 27017

**Loading the Data:**

Before entering the Mongo Shell, load in the data by doing the following command in terminal (make sure script.js is in your current directory, or specify an absolute path to the file, I copied it to /root):

mongo < script.js

mongo

Now do the following in the Mongo Shell

show dbs

\*Ensure that the only dbs are admin, config, local, tcu

use tcu

show collections

\*Ensure that the only collections are courses, faculty, students

Get a feel of the data by looking through the collections by typing:

db.courses.find().pretty()

db.faculty.find().pretty()

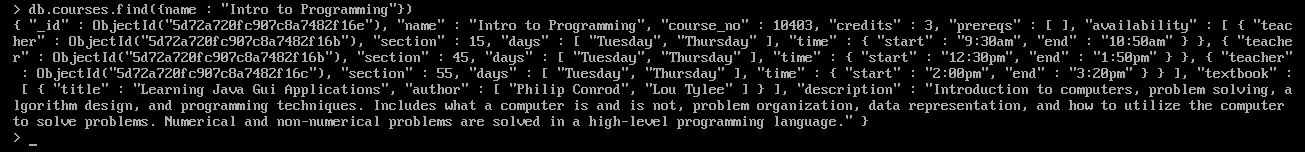
db.students.find().pretty()

Provide Mongo queries for each question. Please also take a screenshot of the result returned by Mongo and paste it after each question.

**Queries (50pts):**

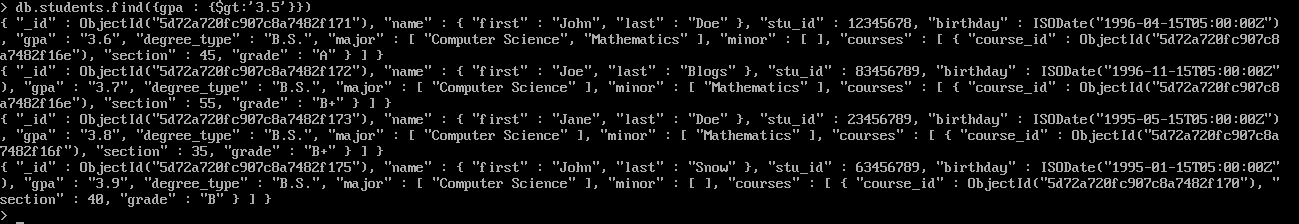
1. A prospective student is looking at TCU COSC, write a simple query to return the document in the courses collection that has a field of ‘name’, and a value of ‘Intro to Programming’.

db.courses.find({name : “Intro to Programming”})



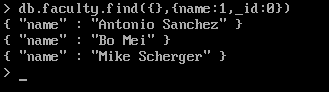
1. We want to find the students eligible for Honor Roll. Write a query for the students collection to find all of the students with a gpa higher than or equal to 3.5 (Hint: use the $gte operator, and make sure 3.5 is surrounded by quotes to match data type).

db.students.find({ gpa : { $gt: '3.5' }})



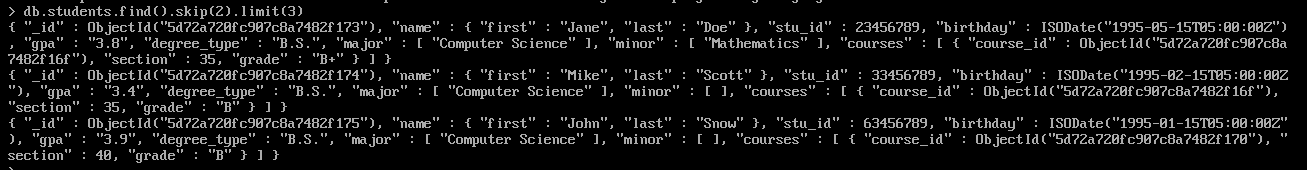
1. We just want a list of the teacher names, not including the \_id. Write a query to do so.

db.faculty.find({}, {name:1, \_id:0})



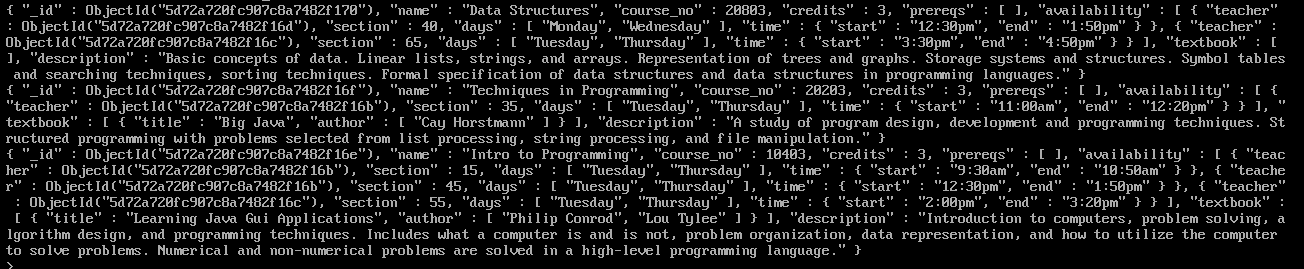
1. Our website wants to implement pagination, to test it out, write a simple query that returns 3 students from the students collection, skipping over the 2.

db.students.find().skip(2).limit(3)



1. Write a query to sort the courses in reverse order by ‘course\_no’.

db.courses.find().sort({course\_no : -1})



Provide Mongo write operations for each question. Please also take a screenshot of the result returned by Mongo and paste it after each question.

**Write Operations (50pts):**

1. Write the commands to insert two more faculty members in the faculty collection (info can be found on TCU COSC website).

db.faculty. insertOne(

{

"name" : "Bingyang Wei",

"site" : " http://personal.tcu.edu/bwei/"

}

)

db.faculty. insertOne(

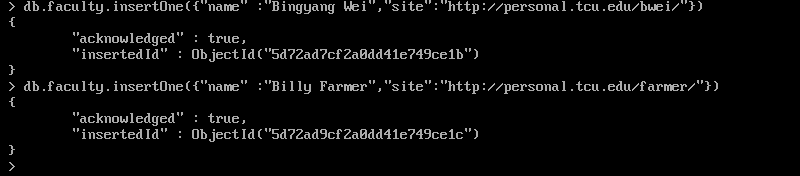
{

"name" : "Billy Farmer",

"site" : " http://csfaculty.tcu.edu/farmer/"

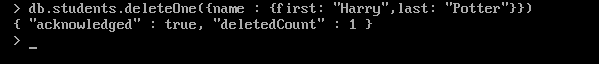
}

)



1. Looks like Harry Potter transferred. Write the command to delete his document from the collection.

db.students.deleteOne({name : {first: “Harry”, last : “Potter”}})



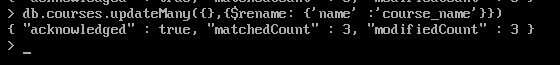
1. We want to add expected graduation date to the student records. Write a command to set a new field of ‘grad\_date’ to 2020 for every student.

db.students.updateMany( { } ,{ $set: { grad\_date: 2020} } )



1. Write a command to change the name of the field ‘name’ in the courses collection to ‘course\_name’ for every document.

db.courses.updateMany({ }, { $rename: { ‘name’ : ‘course\_name’ } })



1. Write the commands to update the ‘prereqs’ field of ‘Intro to Programming’ to [‘High School Algebra’], ‘Techniques in Programming’ to [‘Intro to Programming’], and ‘Data Structures’ to [‘Techniques in Programming’].

db.courses.updateOne( { name : “Intro to Programming” } ,{ $set: { prereqs: [‘High School Algebra’]} } )

db.courses.updateOne( { name : “Techniques in Programming” }, { $set: { prereqs: [“Intro to Programming”]} } )

db.courses.updateOne( { name : “Data Structures” }, { $set: { prereqs: [“Techniques in Programming”]} } )

