VIT UNIVERSITY, ANDHRA PRADESH

Lab Sheet 2 : Mongo DB

Academic year: 2019-2020 | Branch/ Class: B.Tech/M.Tech
Semester: Winter | Date:
Faculty Name: Prof.S.Karthikeyan | School: SCOPE
Student name: | Reg. no.:

MongoDB Queries #2 – NoSQL Databases

Individual Assignment

20 Points

1]

Install MongoDB

1] Find a YouTube video which enables you to successfully install, and start, MongoDB on your windows server. Set up your databases at C:\Data\db. Set up your application in C:\Program Files\MongoDB.

2] _____ The URL for the tutorial I used was _?_.

3] _____ {Sign/Pledge} I have successfully installed MongoDB on my Ubuntu/windows server.

4] Command used to install MOngoDB_____

5] How to start the Mongo DB .write the command_____

Starting MongoDB

Review ➔ Learn MongoDB
http://www.tutorialspoint.com/mongodb

Review ➔ Learn MongoDB
https://docs.mongodb.com/manual

1] _____ MongoDB is a high performance D_?_ O_?_ Database. This does not mean that the only thing you can store in the

2] _____ Let us assume that you have opened a command/terminal window whose current directory is the bin directory in MongoDB. Write the line of code that you could use to launch the Mongo database server application.

3] _____ {T/F} Unless you install MongoDB as a service, you will have to go back to the bin directory and launch the database server application whenever you want to use Mongo.

4] _____ Let us assume that you have opened a second command/terminal window whose current directory is the bin directory in MongoDB. Write the line of code that you could use to start the Mongo.

BSON ➔ Binay JSON

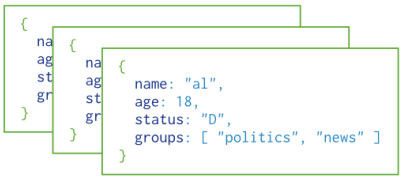
MongoDB stores data in the form of JSON-like value pair documents; these documents are analogous to Structures/Classes in programming languages that associate keys with values (e.g. dictionaries, hashes, maps, and associative arrays).

Formally, MongoDB documents are BSON (binary representation of JSON) documents with additional type information. In BSON documents, the value of a field can be any of the BSON data types, including other documents, arrays, and arrays of documents.

The graphic above, represents the BSON document for Sue.

1] _____ MongoDB data is stored using JSON-like value pairs; this format is called B_?_ format.

Collections



MongoDB stores all documents in Collections. Each collection contains one, or more related documents that have a set of shared common indexes. Collections are analogous to a table in relational databases. If the database were Trinity University, then the collections might be students, faculty, classes, departments, etc. Although the documents within a collection do not have to have exactly the same fields, they should be related for optimal performance. ➔ it does not make sense to combine the students and the departments into the same collection.

1] _____ Within each database, MongoDB stores all documents in C_?_.

2] _____ Each mongo C_?_ contains one, or more related documents that have a set of shared common indexes.

CRUD

In computer programming, create, read, update and delete (as an acronym CRUD) are the four basic functions of persistent storage, It is also sometimes used to describe user interface conventions that facilitate viewing, searching, and changing information.. The term was likely first popularized by James Martin in his 1983 book Managing the Data-base Environment. It is important to note that with data protection concepts → it is legally not allowed to delete data directly.

1] _____ CRUD is an acronym for _?_.

2] _____ BREAD is an acronym for _?_. [Hint: "Browse, Read, Edit, Add, Delete" → Not On Exam or Quiz]

3] _____ MADS is an acronym for _?_. [Hint: "Modify, All, Delete, Show" → Not On Exam or Quiz]

EXECUTE ALL OF THE FOLLOWING QUERY REQUESTS

Be sure to execute All of the query requests in this lab.

Show The Databases

1] _____ Write the line of code that will list all of the Databases.

2] _____ The database that is automatically installed with each and every new Mongo install is called l_?_.

Create A New Database ➔ use

1] _____ Write the line of code that will prepare to create a new database, called **Trinity**, and make it the default database. The database is actually created with the first insert.

2] _____ It is sometimes the case that the database is not actually created until something is written to that database. Write the line of code to add → **name: "Tom"** → to a collection called **students**;

3] _____ {T/F} When I execute the command "**show dbs**", I now see **Trinity** and **local**.

4] _____ {T/F} Suppose I return to MongoDB next week. I start the server. I open the command window. When I execute the command "**use Trinity**", this makes Trinity my default database.

Add A New Record ➔ db.insert

1] _____ It is sometimes the case that the database is not actually created until something is written to that database. Write the

line of code to add → **name: "Dick"** → to a collection called **students**;

2] _____ It is sometimes the case that the database is not actually created until something is written to that database. Write the line of code to add → **name: "Harry"** → to a collection called **students**.

3] _____ It is sometimes the case that the database is not actually created until something is written to that database. Write the line of code to add → **name: "Harry"** → to a collection called **students**;

4] _____ It is sometimes the case that the database is not actually created until something is written to that database. Write the line of code to add → **name: "Harry"** → to a collection called **students**;

5] _____ It is sometimes the case that the database is not actually created until something is written to that database. Write the line of code to add → **name: "Harry"** → to a collection called **students**;

Show Collections

1] _____ Write the line of code that will list all of the collections in the default database.

2] _____ Write the line of code that will make **local** the default database.

3] _____ List at least one of the collections in the **local** the default database.

Configuring The Mongo Prompt

1] _____ The Mongo prompt is _?_

2] _____ Write the line of code that you could use to make the mongo prompt = **Database Name ->[** Hint: Select Mongo Shell ➔ select Configure the mongo Shell on the Mongo Documentation Tutorial] Not on Exam/Quiz!

Display Documents → find()

1] _____ Write the line of code that you could use to display all of the documents in the **student** collection.

2] _____ If you have entered only the documents specified in this lab, you will have _?_ documents in the student listing from student.find();

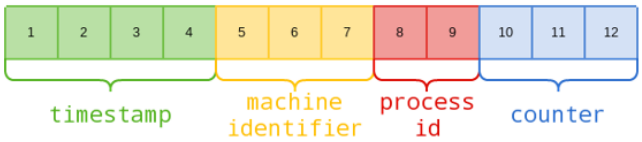
3] _____ If you have entered only the documents specified in this lab, each of the documents will have _?_ fields.;

4] _____ In MongoDB, if you do not have an object of ObjectID type, the database will automatically create one called _?_ ; it is the equivalent of a primary key in relational database. This makes each and every record unique.

To avoid these bottlenecks MongoDB uses ***ObjectId*** as a default , which uses 12 bytes of storage. Now this 12 bytes are very interestingly divided into 4 sub parts to ensure that you will always get unique _id value in any case → even though items may be on hundreds of computers in many data clusters.

- _id generated on two different machines.
- _id generated on the same machine but by two different processes.
- _id generated on the same machine ,by the same process but in same second.

The following graphic shows the Bytes Distribution for ObjectID:



Display Documents In Order → find().sort

1] _____ Write the line of code that you could use to display all of the documents in the **student** collection → in order by **name**.

2] _____ Write the line of code that you could use to display all of the documents in the **student** collection → in order by **_id**.

Display A Partial Set Of The Documents → find().limit

1] _____ Write the line of code that you could use to display the first three of the documents in the **student** collection → in order by **name**.

2] _____ Write the line of code that you could use to display the first of the documents in the **student** collection → in order by **_id**.

Delete A Collection → drop() → Collection

1] _____ Write the line of code that will prepare to create a new database, called **Test**, and make it the default database. The database is actually created with the first insert.

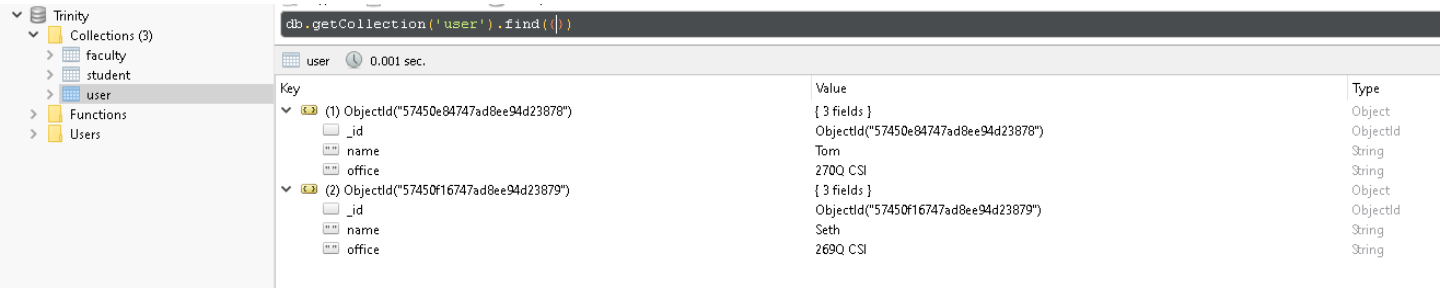
2] _____ Write the line of code to add → **name: "Dick"** → to a collection called **user**.

3] _____ Write the line of code to delete the collection called **User**.

Single or Double Quotes?

```
db.user.insert({name: "Tom", office: "270Q CSI"})
db.user.insert({name: 'Seth', office: '269Q CSI'})
```

1] _____ {Y/N} Execute the two lines above. Does it appear that programmers have the choice of using either single or double quotes to bound string data? Note that the ObjectID's will be different than mine.



Delete A Document → remove() → Document

1] _____ Write the line of code that will prepare to create a new database, called **Trinity**, and make it the default database. The database is actually created with the first insert.

2] _____ Write the line of code to delete the **User** whose name is **Tom**.

3] _____ Write the line of code to delete all of the **students**.

Field Names Are Case Sensitive

```
db.user.remove({'Name': 'Seth'})
```

1] _____ Execute the query above to remove Seth. Can you tell me why it did not remove Seth?

2] _____ {T/F} Case sensitivity does not matter when I am using a field name in a query.

String Matches Are Case Sensitive

```
db.user.remove({'name': 'seth'})
```

1] _____ Execute the query above to remove Seth. Can you tell me why it did not remove Seth?

2] _____ {T/F} Case sensitivity does matter when I am using a string in a query.

Create File Faculty.txt

1] _____ {Sign/Pledge} I have created a file, called **Faculty.txt,** that contains the 8 data records seen below.

*C:\Users\thicks\Desktop\faculty.txt - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

faculty.txt

```
db.faculty.insert({name: "Dr. Myers", specialty: "Theory", tuid: 100000, tenured: true})
db.faculty.insert({name: "Dr. Hicks", specialty: "Database", tuid: 100001, tenured: true})
db.faculty.insert({name: "Dr. Lewis", specialty: "Theory", tuid: 100002, tenured: true})
db.faculty.insert({name: "Dr. Fogarty", specialty: "Functional Programming", tuid: 100003, tenured: false})
db.faculty.insert({name: "Dr. Hibbs", specialty: "Bioinformatics", tuid: 100004, tenured: false})
db.faculty.insert({name: "Dr. Zhang", specialty: "Artificial Intelligence", tuid: 100005, tenured: true})
db.faculty.insert({name: "Dr. Jiang", specialty: "Gane Theory", tuid: 100006, tenured: false})
db.faculty.insert({name: "Dr. Massingil", specialty: "Operating Systems", tuid: 100007, tenured: true})
```

Normal text file | length : 788 lines : 9 | Ln : 9 Col : 1 Sel : 0 | 0 | Dos\Windows | UTF-8 | INS

Create File Student.txt

1] _____ {Sign/Pledge} I have created a file, called **Student.txt**, that contains the 10 data records seen below.

*C:\Users\thicks\Desktop\Student.txt - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

Student.txt | faculty.txt

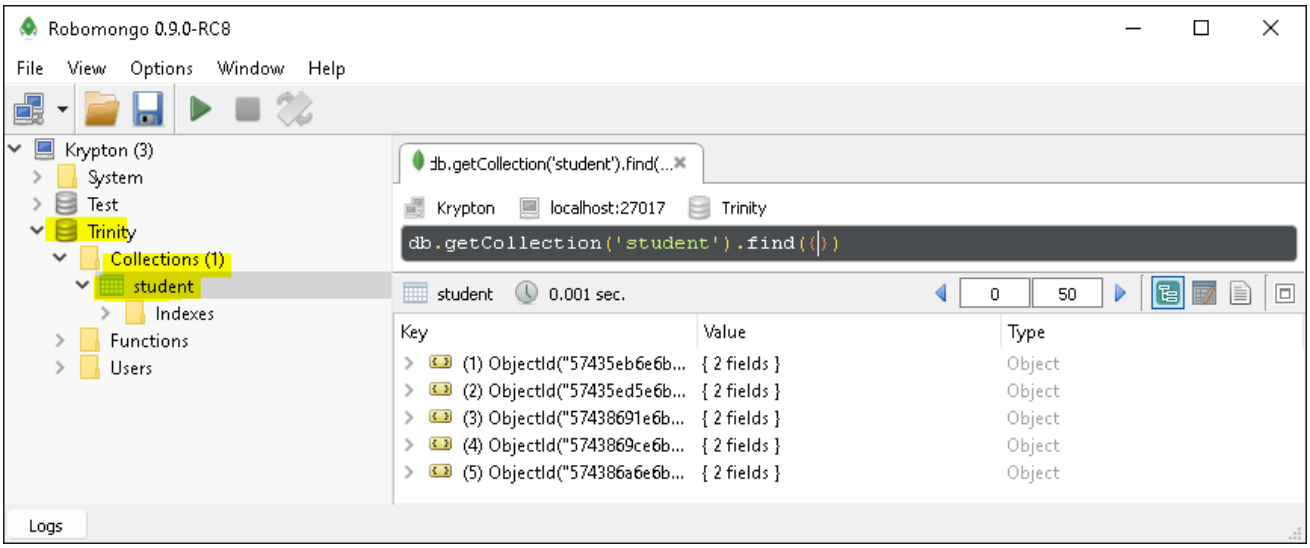
```
db.student.insert({name: "Matt", major: "Computer Science", tuid: 990000, gender: 'Male'})
db.student.insert({name: "Judy", major: "Physics", tuid: 990001, gender: 'Female'})
db.student.insert({name: "Peter", major: "Math", tuid: 990005, gender: 'Male'})
db.student.insert({name: "Doug", major: "Computer Science", tuid: 990006, gender: 'Male'})
db.student.insert({name: "Ann", major: "Computer Science", tuid: 990007, gender: 'Female'})
db.student.insert({name: "Zach", major: "Computer Science", tuid: 990008, gender: 'Male'})
db.student.insert({name: "Allison", major: "Computer Science", tuid: 990009, gender: 'Female'})
db.student.insert({name: "Nathan", major: "Math", tuid: 990002, gender: 'Male'})
db.student.insert({name: "Susan", major: "Computer Science", tuid: 990003, gender: 'Female'})
db.student.insert({name: "Betty", major: "Computer Science", tuid: 990004, gender: 'Female'})
```

Normal text file | length : 906 lines : 12 | Ln : 11 Col : 1 Sel : 0 | 0 | Dos\Windows | UTF-8 | INS

Robomongo

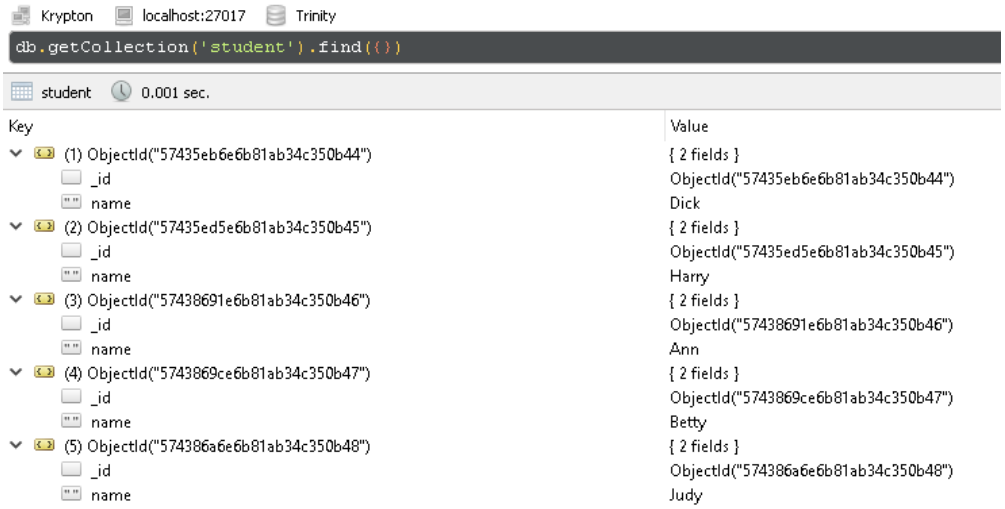
1] _____ {Y/N} Copy the first of the insertions from Faculty.txt to the clipboard. Try to paste it into the command window running mongo. Were you successful?

2] _____ {Y/N} Start Robomongo. Connect to your server. Were you successful creating the connection? If not, do a YouTube search on Robomongo. Robomongo provides a free GUI entry into your Mongo database.

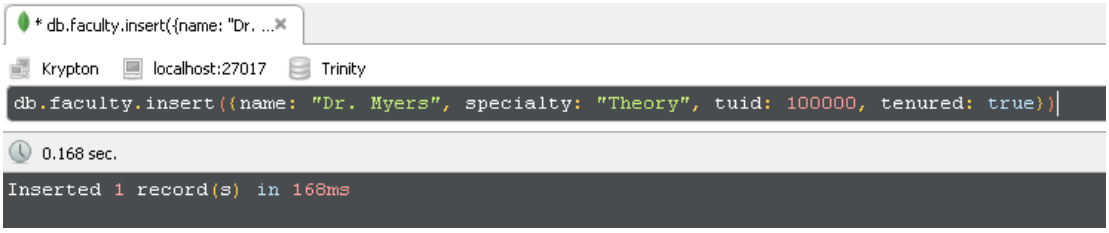


3] _____ {Y/N} Using the mouse, open double click on the Trinity Database. Using the mouse, double click on Collections. Using the mouse, double click on the student collection. Note that the ObjectID's will be different than mine. Do you see some entries similar to that above?

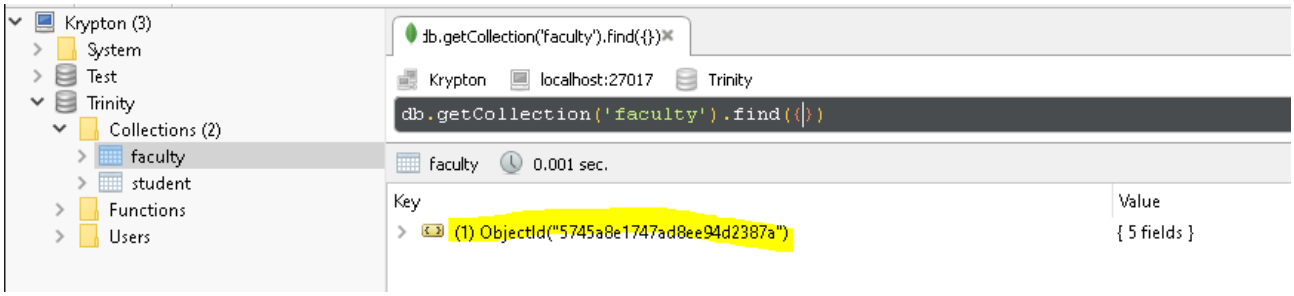
The Green Triangle at the top of Robomongo is used to execute the queries in the black query box. When you double clicked on the student collection above, it added the command, db.getCollection('student').find({}), to the query box and executed the query. See Above.



4] _____ {Y/N} Using the mouse, click on the beside each of the entries of the student collection. Do you see some entries expanded in a format similar to that above?



5] _____ {Y/N} Copy the first of the insertions from Faculty.txt to the clipboard. Try to paste it into the black query window of Robomongo & Execute. Were you successful? (See Above)



6] _____ {Y/N} When I show the items in the faculty collection, I see something similar to that above.

7] _____ {Y/N} Do other query commands that you have been using, like "Show Collections" work within the GUI? See Above.

pretty()

```
{ "_id" : ObjectId("5745d0e6747ad8ee94d238ae"), "name" : "Dr. Myers", "specialty" : "Theory", "tu
id" : 100000, "tenured" : true }
```

1] _____ Write the line of code that you could use to display all of the documents in the **faculty** collection – as illustrated above.

```
{
        "_id" : ObjectId("5745d0e6747ad8ee94d238ae"),
        "name" : "Dr. Myers",
        "specialty" : "Theory",
        "tuid" : 100000,
        "tenured" : true
}
```

2] _____ Write the line of code that you could use to display all of the documents in the **faculty** collection in the pretty format shown above. This is quite useful as the number of records increase.

Faculty Collection

1] _____ Write the line of code to delete the collection called **Faculty**.

Krypton localhost:27017 Trinity

```
db.faculty.insert({name: "Dr. Myers", specialty: "Theory", tuid: 100000, tenured: true})
db.faculty.insert({name: "Dr. Hicks", specialty: "Database", tuid: 100001, tenured: true})
db.faculty.insert({name: "Dr. Lewis", specialty: "Theory", tuid: 100002, tenured: true})
db.faculty.insert({name: "Dr. Fogarty", specialty: "Functional Programming", tuid: 100003, tenured: false})
db.faculty.insert({name: "Dr. Hibbs", specialty: "Bioinformatics", tuid: 100004, tenured: false})
db.faculty.insert({name: "Dr. Zhang", specialty: "Artificial Intelligence", tuid: 100005, tenured: true})
db.faculty.insert({name: "Dr. Jiang", specialty: "Gane Theory", tuid: 100006, tenured: false})
db.faculty.insert({name: "Dr. Massingil", specialty: "Operating Systems", tuid: 100007, tenured: true})
```

2] _____ {Y/N} I have pasted the 8 insertions, from my Faculty.txt file, to the clipboard. I have pasted them into the query box at the top of Robomongo and Executed them all of them at once.

Krypton localhost:27017 Trinity

```
db.getCollection('faculty').find({})
```

faculty 0.001 sec.

Key	Value
(1) ObjectId("5745d0e6747ad8ee94d238ae")	{ 5 fields }
_id	ObjectId("5745d0e6747ad8ee94d238ae")
name	Dr. Myers
specialty	Theory
tuid	100000.0
tenured	true
(2) ObjectId("5745d0e6747ad8ee94d238af")	{ 5 fields }
(3) ObjectId("5745d0e6747ad8ee94d238b0")	{ 5 fields }
(4) ObjectId("5745d0e6747ad8ee94d238b1")	{ 5 fields }
(5) ObjectId("5745d0e6747ad8ee94d238b2")	{ 5 fields }
(6) ObjectId("5745d0e6747ad8ee94d238b3")	{ 5 fields }
(7) ObjectId("5745d0e6747ad8ee94d238b4")	{ 5 fields }
(8) ObjectId("5745d0e6747ad8ee94d238b5")	{ 5 fields }

3] _____ {Y/N} Using the mouse, double-click on each of the Faculty collection. Note that the ObjectID's will be different than mine. My output looks something like that above.

Different Student Collection

1] _____ Write the line of code to delete the collection called **Student**.

* db.getCollection('faculty').fin... | db.getCollection('student').find... | * db.getCollection('student').fin...

Krypton localhost:27017 Trinity

```
db.student.insert({name: "Matt", major: "Computer Science", tuid: 990000, gender: 'Male'})
db.student.insert({name: "Judy", major: "Physics", tuid: 990001, gender: 'Female'})
db.student.insert({name: "Peter", major: "Math", tuid: 990005, gender: 'Male'})
db.student.insert({name: "Doug", major: "Computer Science", tuid: 990006, gender: 'Male'})
db.student.insert({name: "Ann", major: "Computer Science", tuid: 990007, gender: 'Female'})
db.student.insert({name: "Zach", major: "Computer Science", tuid: 990008, gender: 'Male'})
db.student.insert({name: "Allison", major: "Computer Science", tuid: 990009, gender: 'Female'})
db.student.insert({name: "Nathan", major: "Math", tuid: 990002, gender: 'Male'})
db.student.insert({name: "Susan", major: "Computer Science", tuid: 990003, gender: 'Female'})
db.student.insert({name: "Betty", major: "Computer Science", tuid: 990004, gender: 'Female'})
```

2] _____ {Y/N} I have pasted the 10 insertions, from my Student.txt to the clipboard. I have pasted them into the query box at the top of Robomongo and Executed them all of them at once.

Krypton localhost:27017 Trinity

```
db.getCollection('student').find({})
```

student 0.001 sec.

Key	Value
(1) ObjectId("5745c8b8747ad8ee94d23894")	{ 5 fields }
_id	ObjectId("5745c8b8747ad8ee94d23894")
name	Matt
major	Computer Science
tuid	990000.0
gender	Male
(2) ObjectId("5745c8b8747ad8ee94d23895")	{ 5 fields }
(3) ObjectId("5745c8b8747ad8ee94d23896")	{ 5 fields }
(4) ObjectId("5745c8b8747ad8ee94d23897")	{ 5 fields }
(5) ObjectId("5745c8b8747ad8ee94d23898")	{ 5 fields }
(6) ObjectId("5745c8b8747ad8ee94d23899")	{ 5 fields }
(7) ObjectId("5745c8b8747ad8ee94d2389a")	{ 5 fields }
(8) ObjectId("5745c8b8747ad8ee94d2389b")	{ 5 fields }
(9) ObjectId("5745c8b8747ad8ee94d2389c")	{ 5 fields }
(10) ObjectId("5745c8b8747ad8ee94d2389d")	{ 5 fields }

3] _____ {Y/N} Using the mouse, double-click on each of the Student collection. Note that the ObjectID's will be different than mine. My output looks something like that above.

limit()

```
{ "_id" : ObjectId("5745c135747ad8ee94d23882"), "name" : "Matt", "major" : "Computer Science", "t
uid" : 990000, "gender" : "Male" }
{ "_id" : ObjectId("5745c135747ad8ee94d23883"), "name" : "Judy", "major" : "Physics", "tuid" : 99
0001, "gender" : "Female" }
{ "_id" : ObjectId("5745c135747ad8ee94d23884"), "name" : "Nathan", "major" : "Math", "tuid" : 990
002, "gender" : "Male" }
{ "_id" : ObjectId("5745c135747ad8ee94d23885"), "name" : "Susan", "major" : "Computer Science", "
tuid" : 990003, "gender" : "Female" }
{ "_id" : ObjectId("5745c135747ad8ee94d23886"), "name" : "Betty", "major" : "Computer Science", "
tuid" : 990004, "gender" : "Female" }
{ "_id" : ObjectId("5745c135747ad8ee94d23887"), "name" : "Peter", "major" : "Math", "tuid" : 9900
05, "gender" : "Male" }
```

1] _____ Write the line of code that you could use to display the first 6 **faculty** documents in the pretty format shown above.

```
{
        "_id" : ObjectId("5744c06cd4db8ebd30de3750"),
        "name" : "Dr. Myers",
        "specialty" : "Theory",
        "TUID" : 100000,
        "Tenured" : true
}
{
        "_id" : ObjectId("5744c3bfd4db8ebd30de3751"),
        "name" : "Dr. Hicks",
        "specialty" : "Database",
        "TUID" : 100001,
        "Tenured" : true
}
{
        "_id" : ObjectId("5744c3bfd4db8ebd30de3752"),
        "name" : "Dr. Lewis",
        "specialty" : "Theory",
        "TUID" : 100002,
        "Tenured" : true
}
```

2] _____ Write the line of code that you could use to display the first 3 **faculty** documents in the pretty format shown above.

3] _____ Write the line of code that you could use to display the first 4 **student** documents.

4] _____ Write the line of code that you could use to display the first 2 **student** documents in the pretty format

sort()

```
{ "_id" : ObjectId("5745d0e6747ad8ee94d238b1"), "name" : "Dr. Fogarty", "specialty" : "Functional
 Programming", "tuid" : 100003, "tenured" : false }
{ "_id" : ObjectId("5745d0e6747ad8ee94d238b2"), "name" : "Dr. Hibbs", "specialty" : "Bioinformati
cs", "tuid" : 100004, "tenured" : false }
{ "_id" : ObjectId("5745d0e6747ad8ee94d238af"), "name" : "Dr. Hicks", "specialty" : "Database", "
tuid" : 100001, "tenured" : true }
{ "_id" : ObjectId("5745d0e6747ad8ee94d238b4"), "name" : "Dr. Jiang", "specialty" : "Gane Theory"
, "tuid" : 100006, "tenured" : false }
{ "_id" : ObjectId("5745d0e6747ad8ee94d238b0"), "name" : "Dr. Lewis", "specialty" : "Theory", "tu
id" : 100002, "tenured" : true }
{ "_id" : ObjectId("5745d0e6747ad8ee94d238b5"), "name" : "Dr. Massingil", "specialty" : "Operatin
g Systems", "tuid" : 100007, "tenured" : true }
{ "_id" : ObjectId("5745d0e6747ad8ee94d238ae"), "name" : "Dr. Myers", "specialty" : "Theory", "tu
id" : 100000, "tenured" : true }
{ "_id" : ObjectId("5745d0e6747ad8ee94d238b3"), "name" : "Dr. Zhang", "specialty" : "Artificial I
ntelligence", "tuid" : 100005, "tenured" : true }
```

1] _____ Write the line of code that you could use to display all of the **faculty** documents in order by name as illustrated above.

2] _____ Write the line of code that you could use to display the first six of the **faculty** documents in order by name.

3] _____ Write the line of code that you could use to display all of the **faculty** documents in order by tuid.

4] _____ Write the line of code that you could use to display all of the **faculty** documents in pretty format order by tuid.

5] _____ Write the line of code that you could use to display the first six of the **faculty** documents in the pretty format in order by name.

6] _____ Write the line of code that you could use to display the first six of the **student** documents in the pretty format in order by name.

7] _____ Write the line of code that you could use to display the first 10 of the **student** documents in the pretty format in order by major.

8] _____ Write the line of code that you could use to display all of the **faculty** documents in order by tenured → but when the tenure is the same → order the data by name. Your results should resemble those below.

```
{ "_id" : ObjectId("5745d0e6747ad8ee94d238b1"), "name" : "Dr. Fogarty", "specialty" : "Functional
 Programming", "tuid" : 100003, "tenured" : false }
{ "_id" : ObjectId("5745d0e6747ad8ee94d238b2"), "name" : "Dr. Hibbs", "specialty" : "Bioinformati
cs", "tuid" : 100004, "tenured" : false }
{ "_id" : ObjectId("5745d0e6747ad8ee94d238b4"), "name" : "Dr. Jiang", "specialty" : "Gane Theory"
, "tuid" : 100006, "tenured" : false }
{ "_id" : ObjectId("5745d0e6747ad8ee94d238af"), "name" : "Dr. Hicks", "specialty" : "Database", "
tuid" : 100001, "tenured" : true }
{ "_id" : ObjectId("5745d0e6747ad8ee94d238b0"), "name" : "Dr. Lewis", "specialty" : "Theory", "tu
id" : 100002, "tenured" : true }
{ "_id" : ObjectId("5745d0e6747ad8ee94d238b5"), "name" : "Dr. Massingil", "specialty" : "Operatin
g Systems", "tuid" : 100007, "tenured" : true }
{ "_id" : ObjectId("5745d0e6747ad8ee94d238ae"), "name" : "Dr. Myers", "specialty" : "Theory", "tu
id" : 100000, "tenured" : true }
{ "_id" : ObjectId("5745d0e6747ad8ee94d238b3"), "name" : "Dr. Zhang", "specialty" : "Artificial I
ntelligence", "tuid" : 100005, "tenured" : true }
```

9] _____ Write the line of code that you could use to display all of the **student** documents in order by major → but when the major is the same → order the data by name. Your results should resemble those below.

```
{ "_id" : ObjectId("5745de9c747ad8ee94d238be"), "name" : "Allison", "major" : "Computer Science",
 "tuid" : 990009, "gender" : "Female" }
{ "_id" : ObjectId("5745de9c747ad8ee94d238bc"), "name" : "Ann", "major" : "Computer Science", "tu
id" : 990007, "gender" : "Female" }
{ "_id" : ObjectId("5745de9c747ad8ee94d238c1"), "name" : "Betty", "major" : "Computer Science", "
tuid" : 990004, "gender" : "Female" }
{ "_id" : ObjectId("5745de9c747ad8ee94d238bb"), "name" : "Doug", "major" : "Computer Science", "t
uid" : 990006, "gender" : "Male" }
{ "_id" : ObjectId("5745de9c747ad8ee94d238b8"), "name" : "Matt", "major" : "Computer Science", "t
uid" : 990000, "gender" : "Male" }
{ "_id" : ObjectId("5745de9c747ad8ee94d238c0"), "name" : "Susan", "major" : "Computer Science", "
tuid" : 990003, "gender" : "Female" }
{ "_id" : ObjectId("5745de9c747ad8ee94d238bd"), "name" : "Zach", "major" : "Computer Science", "t
uid" : 990008, "gender" : "Male" }
{ "_id" : ObjectId("5745de9c747ad8ee94d238bf"), "name" : "Nathan", "major" : "Math", "tuid" : 990
002, "gender" : "Male" }
{ "_id" : ObjectId("5745de9c747ad8ee94d238ba"), "name" : "Peter", "major" : "Math", "tuid" : 9900
05, "gender" : "Male" }
{ "_id" : ObjectId("5745de9c747ad8ee94d238b9"), "name" : "Judy", "major" : "Physics", "tuid" : 99
0001, "gender" : "Female" }
```

find() → Match A Single Field Equality

1] _____ Write the line of code that you could use to display all of the **faculty** documents whose name is "Dr. Myers".

```
db.faculty.find({name: 'Dr. Lewis'})
```

2] _____ {T/F} The query above could be used to display all of the faculty documents whose name is 'Dr. Lewis'.

```
db.faculty.find({'name': 'Dr. Lewis'})
```

3] _____ {T/F} The query above could be used to display all of the faculty documents whose name is 'Dr. Lewis'.
[T]

```
db.faculty.find({name: "Dr. Lewis"})
```

4] _____ {T/F} The query above could be used to display all of the faculty documents whose name is 'Dr. Lewis'.
[T]

```
db.faculty.find({"name": "Dr. Lewis"})
```

5] _____ {T/F} The query above could be used to display all of the faculty documents whose name is 'Dr. Lewis'.
[T]

```
db.faculty.find({'name': "Dr. Lewis"})
```

6] _____ {T/F} The query above could be used to display all of the faculty documents whose name is 'Dr. Lewis'.
[T]

```
db.faculty.find({name: "Dr. Lewis'})
```

7] _____ {T/F} The query above could be used to display all of the faculty documents whose name is 'Dr. Lewis'.

8] _____ Write the line of code that you could use to display all of the **faculty** documents whose tuid is 100001.

```
db.faculty.find({tuid: '100002'})
```

9] _____ {T/F} The query above could be used to display all of the faculty documents whose trinity id is 100002.

10] _____ Write the line of code that you could use to display all of the tenured **faculty** documents in order by name.

11] _____ Write the line of code that you could use to display all of the tenured **faculty** documents in order by name.

12] _____ Write the line of code that you could use to display all of the Computer Science **student** documents in order by name.

What To Turn In

- - - - - - - - - - - - - No Lab Is Complete Until Both Are Complete - - - - - - - - - - - - -

1] You sign & submit the Pledge form.
   a) Review the Pledge statement
   b) Record the amount of time you think you spent on this lab
   c) Staple all pages of this lab. Fold in half length-wise (like a hot-dog). Put your name on the outside. Place it on the Vtop before the beginning of lecture on the day it is due. The penalty for late homework will not exceed 25% off per day.

2] Place all programming code associated with this program, if any, in the Professor's Code Drop Box
   a) I do not accept programs by mail; do not submit labs via email!

- - - - - - - - - - - - - - - - - Comments - - - - - - - - - - - - - - - - -

A] Programs that do not compile are worth little, if anything.
B] If a print statement format is off, the penalties will often be less than the 25% per day late penalty; turn in the lab. You would not be happy if you went to Best Buy and purchased a large screen TV that did everything except show the picture; you would consider it pretty worthless. Most users consider software that does not work properly pretty useless as well. If the lab is not working correctly, credit will be small (if any); you might be better to accept a 25% (1 day) late penalty and turn in the lab working correctly!
C] Start all programs early so that you can get in contact with the professor if you have problems.
D] If you are turning in this lab late, you may
   - hand it to me if I am in the office
   - put it in the mail box outside my office door
   - slide it under the outer door to our suite {if locked}
   - slide it under my office door. The sooner I get late labs, the sooner the late penalty meter quits clicking.

E] Backup your programs in at least three places. Put a copy on your Y drive. Put a copy on your flash drive. Put a copy on your personal computer. Send yourself a copy in your e-mail.