MongoDB_Lab1

1 – open mongo shell and view the help

Installation and configuration

Windows: https://fastdl.mongodb.org/windows/mongodb-windows-x86_64-5.0.6-signed.msi

UBUNTU:

Reload local package database

$ sudo apt-get update

Install the MongoDB packages

$ sudo apt-get install mongodb

MongoDB service

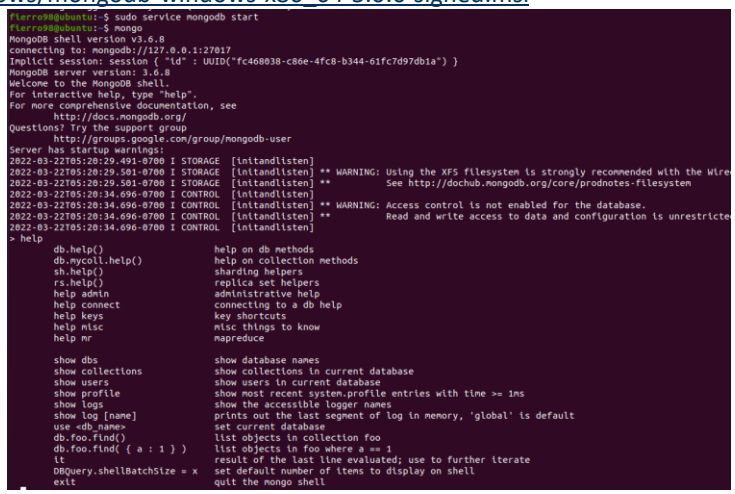
$ sudo service mongodb start

Start MongoDB Shell

$ mongo

> help

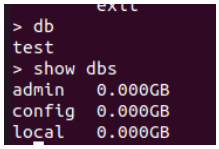
(or) $ mongo --help



2 – identify your current working database and show list of available databases

> db

> show dbs



3 – create a new database called Iti and create a collection named “students”. Insert whatever data you want about yourself (include name and age in your details).

> use iti

> db.createCollection('students')

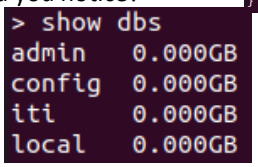
> db.students.insert ({"Name": "Mahmoud Kamal", "Age": 24})

> db.students.find().pretty()

4– show a list of available databases. What did you notice?

> show dbs

My database is added and all dbs have 0GB



5 – Insert un-structured or semi-structured data for 10 of your friends (include name and age in your details. The documents should have different types of data i.e., arrays, strings, documents, integers).

> db.createCollection('friends')

> db.friends.insert ([{"Name": "Mahmoud Kamal", "Age": 24, "Hobbies":["Football", "Swimming"]}, {"Name": "Amr", "Age": 24, "Hobbies":["Football", "Games"]}, {"Name": "Ahmed", "Age": 24}, {"Name": "Nada", "Age": 23}, {"Name":

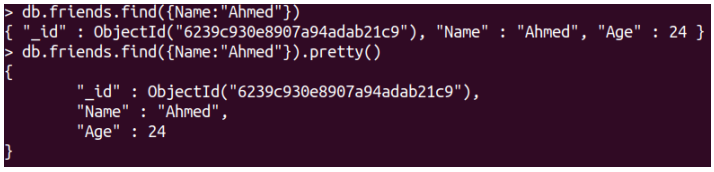
"Donia", "Age": 23}, {"Name": "Mohamed", "Age": 26}, {"Name": "Hassan", "Age": 24}, {"Name": "Youssef", "Age": 23}, {"Name": "Salma", "Age": 24}, {"Name": "Rana", "Age": 24}])

> db.friends.find().pretty()

6 – Search for your object by name.

> db.friends.find({Name:"Ahmed"})

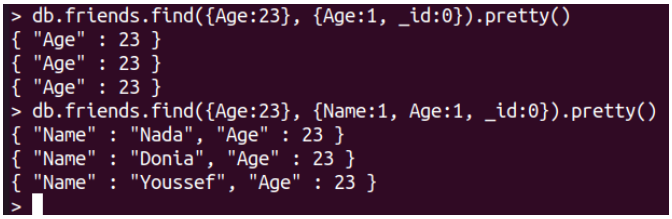
> db.friends.find({Name:"Ahmed"}).pretty()



7–Search for your friend(s) by age.

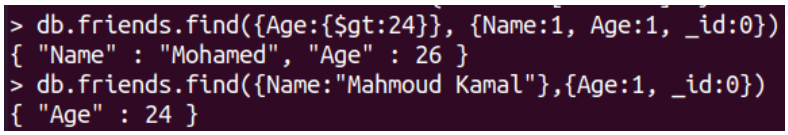
> db.friends.find({Age:23}, {Name:1, Age:1, _id:0})

> db.friends.find({Age:23}, { Name:1, Age:1, _id:0}).pretty()



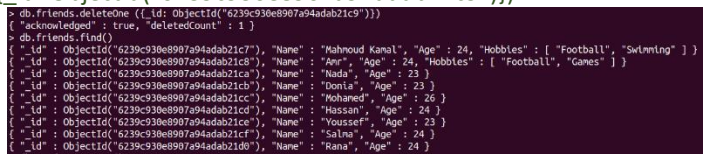
8 – Search for all of your friends whose age is older than yours.

> db.friends.find({Age:{$gt:24}}, {Name:1, Age:1, _id:0})



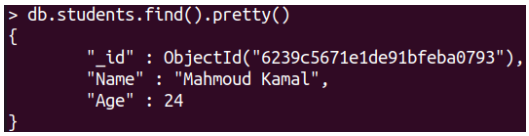
9 – delete any of your friends by id.

> db.friends.deleteOne ({_id: ObjectId("6239c930e8907a94adab21c9")})



10 – view all documents in students' collection in a prettified format.

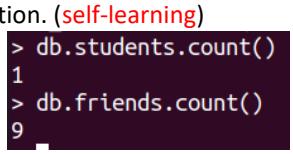
> db.students.find().pretty()



11 – count all documents in students' collection. (self-learning)

> db.students.count()

> db.friends.count()



---

part 2

1- Create database with name ems

> use ems

> db

```
> use ems
switched to db ems
> db
ems
```

2- Insert the following data into "faculty" collection

> db.faculty.insert([

```
{ "name":"Krish", "age":35,"gender":"M","exp":10,subjects:["DS","C","OS"],"type":"Full Time","qualification":"M.Tech" },
    { "name":"Manoj", "age":38,"gender":"M","exp":12,subjects:["JAVA","DBMS"],"type":"Full Time",
     "qualification":"Ph.D"},
     { "name":"Anush", "age":32,"gender":"F","exp":8,subjects:["C","CPP"],"type":"Part Time","qualification":"M.Tech" },
     { "name":"Suresh", "age":40,"gender":"M","exp":9,subjects:["JAVA","DBMS","NETWORKING"],"type":"Full Time",
     "qualification":"Ph.D"},
     { "name":"Rajesh", "age":35,"gender":"M","exp":7,subjects:["DS","C","OS"],"type":"Full Time","qualification":"M.Tech"
     },
     { "name":"Mani", "age":38,"gender":"F","exp":10,subjects:["JAVA","DBMS","OS"],"type":"Part Time",
     "qualification":"Ph.D"},
    { "name":"Sivani", "age":32,"gender":"F","exp":8,subjects:["C","CPP","MATHS"],"type":"Part
     Time","qualification":"M.Tech" },
    { "name":"Nagesh", "age":39,"gender":"M","exp":11,subjects:["JAVA","DBMS","NETWORKING"],"type":"Full Time",
     "qualification":"Ph.D"},
    { "name":"Nagesh", "age":35,"gender":"M","exp":9,subjects:["JAVA",".Net","NETWORKING"],"type":"Full Time",
     "qualification":"Ph.D"},
    { "name":"Latha", "age":40,"gender":"F","exp":13,subjects:["MATHS"],"type":"Full Time", "qualification":"Ph.D"}])
```

```
> db.faculty.insert([ { "na
e":"Manoj", "age":38,"gende
"F","exp":8,subjects:["C","
MS","NETWORKING"],"type":"F
 Time","qualification":"M.T
"Ph.D"}, { "name"
ame":"Nagesh", "age":39,"ge
"age":35,"gender":"M","exp"
:"F","exp":13,subjects:["MA
BulkWriteResult({
        "writeErrors" : [ ]
        "writeConcernErrors
        "nInserted" : 10,
        "nUpserted" : 0,
        "nMatched" : 0,
        "nModified" : 0,
        "nRemoved" : 0,
        "upserted" : [ ]
})
> db.faculty.count()
10
```

1. Get the details of all the faculty.

> db.faculty.find()

2. Get the count of all faculty members.

> db.faculty.count()

```
> db.faculty.count()
10
```

3. Get all the faculty members whose qualification is “Ph.D”.

> db.faculty.find({qualification: "Ph.D"}, {name:1, qualification:1, _id:0})

```
> db.faculty.find({qualification: "Ph.D"}, {name:1, qualification:1, _id:0})
{ "name" : "Manoj", "qualification" : "Ph.D" }
{ "name" : "Suresh", "qualification" : "Ph.D" }
{ "name" : "Mani", "qualification" : "Ph.D" }
{ "name" : "Nagesh", "qualification" : "Ph.D" }
{ "name" : "Nagesh", "qualification" : "Ph.D" }
{ "name" : "Latha", "qualification" : "Ph.D" }
```

4. Get all the faculty members whose experience is between 8 to 12 years.

> db.faculty.find({exp:{ $in:[8, 9, 10, 11, 12]}}, {name:1, exp:1, _id:0})

(or)

> db.faculty.find({exp:{$gt: 8}, exp:{$lt: }}, {name:1, exp:1, _id:0})

```
> db.faculty.find({exp:{ $in:[8, 9, 10, 11, 12]}}, {name:1, exp:1, _id:0})
{ "name" : "Krish", "exp" : 10 }
{ "name" : "Manoj", "exp" : 12 }
{ "name" : "Anush", "exp" : 8 }
{ "name" : "Suresh", "exp" : 9 }
{ "name" : "Mani", "exp" : 10 }
{ "name" : "Sivani", "exp" : 8 }
{ "name" : "Nagesh", "exp" : 11 }
{ "name" : "Nagesh", "exp" : 9 }
> db.faculty.find({exp:{$gt: 8}, exp:{$lt: 12}}, {name:1, exp:1, _id:0})
{ "name" : "Krish", "exp" : 10 }
{ "name" : "Anush", "exp" : 8 }
{ "name" : "Suresh", "exp" : 9 }
{ "name" : "Rajesh", "exp" : 7 }
{ "name" : "Mani", "exp" : 10 }
{ "name" : "Sivani", "exp" : 8 }
{ "name" : "Nagesh", "exp" : 11 }
{ "name" : "Nagesh", "exp" : 9 }
```

5. Get all the faculty members who teach “MATHS” or “NETWORKING”.

```
> db.faculty.find({$or:[{subjects: "MATHS"}, {subjects: "NETWORKING"}]}, {name:1, subjects:1, _id:0})
{ "name" : "Suresh", "subjects" : [ "JAVA", "DBMS", "NETWORKING" ] }
{ "name" : "Sivani", "subjects" : [ "C", "CPP", "MATHS" ] }
{ "name" : "Nagesh", "subjects" : [ "JAVA", "DBMS", "NETWORKING" ] }
{ "name" : "Nagesh", "subjects" : [ "JAVA", ".Net", "NETWORKING" ] }
{ "name" : "Latha", "subjects" : [ "MATHS" ] }
```

6. Get all the faculty members who teach “MATHS” and whose age is more than 30 years and qualification must be “Ph.D”.

```
> db.faculty.find({subjects: "MATHS", age:{$gt: 30}, qualification: "Ph.D"},{ _id:0})
{ "name" : "Latha", "age" : 40, "gender" : "F", "exp" : 13, "subjects" : [ "MATHS" ], "type" : "Full Time", "qualification" : "Ph.D" }
```

7. Get all the faculty members who are working part-time or who teach “JAVA”.

```
> db.faculty.find({$or:[{subjects: "JAVA"}, { type: "Part Time"}]}, {name:1, subjects:1, type:1, _id:0})
{ "name" : "Manoj", "subjects" : [ "JAVA", "DBMS" ], "type" : "Full Time" }
{ "name" : "Anush", "subjects" : [ "C", "CPP" ], "type" : "Part Time" }
{ "name" : "Suresh", "subjects" : [ "JAVA", "DBMS", "NETWORKING" ], "type" : "Full Time" }
{ "name" : "Mani", "subjects" : [ "JAVA", "DBMS", "OS" ], "type" : "Part Time" }
{ "name" : "Sivani", "subjects" : [ "C", "CPP", "MATHS" ], "type" : "Part Time" }
{ "name" : "Nagesh", "subjects" : [ "JAVA", "DBMS", "NETWORKING" ], "type" : "Full Time" }
{ "name" : "Nagesh", "subjects" : [ "JAVA", ".Net", "NETWORKING" ], "type" : "Full Time" }
```

8. Add the following new faculty members:

{ "name":"Suresh Babu", "age":55, "gender":"M", "exp":25, subjects: ["MATHS","DE"], "type":"Full Time", "qualification":"Ph.D"}

```
> db.faculty.insert({ "name":"Suresh Babu", "age":55, "gender":"M", "exp":25, subjects:     ["MATHS","DE"],
    "type":"Full Time", "qualification":"Ph.D"})
> db.faculty.count()
10
> db.faculty.insert({ "name":"Sures
WriteResult({ "nInserted" : 1 })
> db.faculty.count()
11
```

9. Update the data of all faculty members by incrementing their age and exp by one year.

```
> db.faculty.updateMany({}, {$inc: { "age": 1, "exp": 1}})
{ "acknowledged" : true, "matchedCount" : 11, "modifiedCount" : 11 }
```

10. Update the faculty “Sivani” with the following data: update qualification to “Ph.D” and type to “Full Time”.

```
> db.faculty.update(
{"name": "Sivani"},
{$set: {"qualification": "Ph.D"}})
> db.faculty.update( {name: "Sivani"}, {$set: {qualification: "Ph.D"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.faculty.find( {name: "Sivani"})
{ "_id" : ObjectId("6239d52f04c3acd2aa100c0e"), "name" : "Sivani", "age" : 33, "gender" : "F", "exp" : 9,
 Time", "qualification" : "Ph.D" }
```

11. Update all faculty members who are teaching “MATHS” such that they should now also teach “PSK”.

> db.faculty.updateMany(
{"subjects": "MATHS"},
{$push: {"subjects": "PSK"}})

```
> db.faculty.updateMany( {"subjects": "MATHS"}, {$push: {"subjects": "PSK"}})
{ "acknowledged" : true, "matchedCount" : 3, "modifiedCount" : 3 }
> db.faculty.find( {"subjects": "MATHS"})
{ "_id" : ObjectId("6239d52f04c3acd2aa100c0e"), "name" : "Sivani", "age" : 33, "gender" : "F", "exp" : 9, "subjects" : [ "C", "CPP", "MATHS", "PSK" ], "type"
: "Part Time", "qualification" : "Ph.D" }
{ "_id" : ObjectId("6239d52f04c3acd2aa100c11"), "name" : "Latha", "age" : 41, "gender" : "F", "exp" : 14, "subjects" : [ "MATHS", "PSK" ], "type" : "Full Time
", "qualification" : "Ph.D" }
{ "_id" : ObjectId("6239dde204c3acd2aa100c12"), "name" : "Suresh Babu", "age" : 56, "gender" : "M", "exp" : 26, "subjects" : [ "MATHS", "DE", "PSK" ], "type"
: "Full Time", "qualification" : "Ph.D" }
```

12. Delete all faculty members whose age is more than 55 years.

> db.faculty.deleteMany({age:{$gt:55}})

```
> db.faculty.count()
11
> db.faculty.deleteMany({age:{$gt:55}})
{ "acknowledged" : true, "deletedCount" : 1 }
> db.faculty.count()
10
```

13. Get only the name and qualification of all faculty members.

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

```
> db.faculty.find({}, {name:1, qualification:1, _id:0})
{ "name" : "Krish", "qualification" : "M.Tech" }
{ "name" : "Manoj", "qualification" : "Ph.D" }
{ "name" : "Anush", "qualification" : "M.Tech" }
{ "name" : "Suresh", "qualification" : "Ph.D" }
{ "name" : "Rajesh", "qualification" : "M.Tech" }
{ "name" : "Mani", "qualification" : "Ph.D" }
{ "name" : "Sivani", "qualification" : "Ph.D" }
{ "name" : "Nagesh", "qualification" : "Ph.D" }
{ "name" : "Nagesh", "qualification" : "Ph.D" }
{ "name" : "Latha", "qualification" : "Ph.D" }
```

14. Get the name, qualification and exp of all faculty members and display the same in ascending order of exp.

> db.faculty.find({}, {name:1, qualification:1, exp:1, _id:0}).sort({exp:1})

```
> db.faculty.find({}, {name:1, qualification:1, exp:1, _id:0}).sort({exp:1})
{ "name" : "Rajesh", "exp" : 8, "qualification" : "M.Tech" }
{ "name" : "Anush", "exp" : 9, "qualification" : "M.Tech" }
{ "name" : "Sivani", "exp" : 9, "qualification" : "Ph.D" }
{ "name" : "Suresh", "exp" : 10, "qualification" : "Ph.D" }
{ "name" : "Nagesh", "exp" : 10, "qualification" : "Ph.D" }
{ "name" : "Krish", "exp" : 11, "qualification" : "M.Tech" }
{ "name" : "Mani", "exp" : 11, "qualification" : "Ph.D" }
{ "name" : "Nagesh", "exp" : 12, "qualification" : "Ph.D" }
{ "name" : "Manoj", "exp" : 13, "qualification" : "Ph.D" }
{ "name" : "Latha", "exp" : 14, "qualification" : "Ph.D" }
> db.faculty.find({}, {name:1, qualification:1, exp:1, _id:0}).sort({exp:1}).forEach(printjson)
{ "name" : "Rajesh", "exp" : 8, "qualification" : "M.Tech" }
{ "name" : "Anush", "exp" : 9, "qualification" : "M.Tech" }
{ "name" : "Sivani", "exp" : 9, "qualification" : "Ph.D" }
{ "name" : "Suresh", "exp" : 10, "qualification" : "Ph.D" }
{ "name" : "Nagesh", "exp" : 10, "qualification" : "Ph.D" }
{ "name" : "Krish", "exp" : 11, "qualification" : "M.Tech" }
{ "name" : "Mani", "exp" : 11, "qualification" : "Ph.D" }
{ "name" : "Nagesh", "exp" : 12, "qualification" : "Ph.D" }
{ "name" : "Manoj", "exp" : 13, "qualification" : "Ph.D" }
{ "name" : "Latha", "exp" : 14, "qualification" : "Ph.D" }
```

15. Sort the faculty details by their age (descending order) and get the details of the first five faculty members only.

> db.faculty.find().sort({age:-1}).limit(5)

```
> db.faculty.find().sort({age:-1}).limit(5)
{ "_id" : ObjectId("6239d52f04c3acd2aa100c0b"), "name" : "Suresh", "age" : 41, "gender" : "M", "exp" : 10, "subjects" : [ "JAVA", "DBMS", "NETWORKING" ], "typ
e" : "Full Time", "qualification" : "Ph.D" }
{ "_id" : ObjectId("6239d52f04c3acd2aa100c11"), "name" : "Latha", "age" : 41, "gender" : "F", "exp" : 14, "subjects" : [ "MATHS", "PSK" ], "type" : "Full Time
", "qualification" : "Ph.D" }
{ "_id" : ObjectId("6239d52f04c3acd2aa100c0f"), "name" : "Nagesh", "age" : 40, "gender" : "M", "exp" : 12, "subjects" : [ "JAVA", "DBMS", "NETWORKING" ], "typ
e" : "Full Time", "qualification" : "Ph.D" }
{ "_id" : ObjectId("6239d52f04c3acd2aa100c09"), "name" : "Manoj", "age" : 39, "gender" : "M", "exp" : 13, "subjects" : [ "JAVA", "DBMS" ], "type" : "Full Time
", "qualification" : "Ph.D" }
{ "_id" : ObjectId("6239d52f04c3acd2aa100c0d"), "name" : "Mani", "age" : 39, "gender" : "F", "exp" : 11, "subjects" : [ "JAVA", "DBMS", "OS" ], "type" : "Part
 Time", "qualification" : "Ph.D" }
```