Mongo DB – Lab (1)

Malak
El-Samra

Part 1

1. Open mongo shell and view the help

```
mongod> help
```

2. Identify your current working database and show list of available databases

```
mongod> db------------ mongod
mongod> 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).

```
mongod> use AlexITI
```

4. Show list of available databases. What did you notice?

```
AlexITI> use mongod
switched to db mongod
mongod> show dbs
AlexITI   40.00 KiB
ITI       72.00 KiB
Nursery  412.00 KiB
admin     40.00 KiB
config   108.00 KiB
local     72.00 KiB
```

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.students.insertMany([
   {name:"Rawia",age:26,department:{name:"PD"},skills:["JS","JQuery","C-Sharp"]},
   {name:"Malak",age:26,department:{name:"PD"},skills:["JS","JQuery","C-Sharp"]},
   {name:"Amr",age:26,department:{name:"PD"},skills:["JS","JQuery","C-Sharp"]},
   {name:"Omar",age:26,department:{name:"PD"},skills:["JS","JQuery","C-Sharp"]},
   {name:"Zeyad",age:26,department:{name:"PD"},skills:["JS","JQuery","C-Sharp"]},
   {name:"Nour",age:26,department:{name:"PD"},skills:["JS","JQuery","C-Sharp"]},
   {name:"Rougina",age:26,department:{name:"PD"},skills:["JS","JQuery","C-Sharp"]},
   {name:"Nawal",age:26,department:{name:"PD"},skills:["JS","JQuery","C-Sharp"]},
   {name:"Omnia",age:26,department:{name:"PD"},skills:["JS","JQuery","C-Sharp"]},
   {name:"Mahmoud",age:26,department:{name:"PD"},skills:["JS","JQuery","C-Sharp"]}
])
```

6. Search for your object by name.

```
AlexITI> db.students.find({name:"Malak"})
```

7. Search for your friend(s) by age.

```
AlexITI> db.students.find({age:26})
```

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

```
AlexITI> db.students.find({age:{$gt:25}})
```

9. Delete any of your friends by id.

```
AlexITI> db.students.deleteOne({_id: ObjectId("643802401dfa3f13c4f65182")})
```

10. View all documents in students' collection in a prettified format.

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

11. Count all documents in students' collection.

```
AlexITI> db.students.count()
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or
estimatedDocumentCount.
AlexITI> db.students.countDocuments()
```

Part 2

1. Create database with name ems

```
mongod> use ems
```

2. Insert the following data into "faculty" collection

```
db.faculty.insertMany([
        { "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"}
])
```

3. Get the details of all the faculty.

```
ems> db.faculty.find()
```

4. Get the count of all faculty members.

```
ems> db.faculty.countDocuments()
```

5. Get all the faculty members whose qualification is "Ph.D".

```
ems> db.faculty.find({qualification:"Ph.D"})
```

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

```
ems> db.faculty.find({exp:{$gt:8, $lt:12}})
```

7. Get all the faculty members who teach "MATHS" or "NETWORKING".

```
ems> db.faculty.find({subjects:{$in:["MATHS","NETWORKING"]}})
```

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

```
ems> db.faculty.find({subjects: "MATHS", age:{$gt:30}, qualification:"Ph.D"})
```

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

```
ems> db.faculty.find({$or:[{type:"Part Time"},{subjects: "JAVA"}]})
```

10. Add the following new faculty members:
{ "name":"Suresh Babu", "age":55,"gender":"M","exp":25,subjects:["MATHS","DE"],"type":"Full Time", "qualification":"Ph.D"}

```
ems> db.faculty.insertOne({ "name":"Suresh Babu",
"age":55,"gender":"M","exp":25,subjects:["MATHS","DE"],"type":"Full Time",
"qualification":"Ph.D"})
```

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

```
ems> db.faculty.updateMany({},{$inc:{age:1, exp:1}})
```

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

```
ems>db.faculty.updateOne({name:"Sivani"},{$set:{qualification:"Ph.D",
type:"Full Time"}})
```

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

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

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

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

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

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

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

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

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

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