MongoDB\_Lab1

1 – open mongo shell and view the 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

b.students.insert({name:"mohamed",age:23,track:"os"})

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

Show dbs;

Iti data base is added

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).

Var list\_students=[{name:"mohamed3",age:23,track:"os",Hobbies:["footb

all","chess"]},{name:"mohamed4",age:23,track:"os",Hobbies:["football","ch

ess"]},{name:"mohamed5",age:23,track:"os",Hobbies:["football","chess"]},{

name:"mohamed6",age:23,track:"os",Hobbies:["football","chess"]},{name:"

mohamed7",age:23,track:"os",Hobbies:["football","chess"]},{name:"moham

ed8",age:23,track:"os",Hobbies:["football","chess"]},{name:"mohamed9",ag

e:23,track:"os",Hobbies:["football","chess"]}];

db.students.insertMany(list\_students);

6 – Search for your object by name.

db.students.find({name:"mohamed6"})

7– Search for your friend(s) by age.

db.students.find({age:23})

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

db.students.find({age:{$lt:23}})

**9** – delete any of your friends by id.

db.students.deleteOne({\_id:ObjectId("6239c1a6164e9c88c176b9fa")})

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.countDocuments({})

**---------------------------------------------------------**

**part 2**

1. Create database with name ems

Use ems

2- Insert the following data into "faculty" collection

{ "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"}

Var students =[all objects ]

db.faculty.insertMany(students)

1. Get the details of all the faculty.

show collections

1. Get the count of all faculty members.

db.faculty.countDocuments({})

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

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

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

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

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

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

1. 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({$and:[{subjects:{$elemMatch:{$in:["MATHS"]}}},{age:{$gt:30}},{qualification:"Ph.D"} ]})

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

db.faculty.find({$or:[{type:"PartTime"},{subjects:{$elemMatch:{$in:["JAVA"]}}}]})

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"} )

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

db.faculty.updateMany({},{$inc:{age:1}})

db.faculty.updateMany({},{$inc:{exp:1}})

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

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

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

db.faculty.updateMany({subjects:{$elemMatch:{$in:["MATHS"]}}},{$push:{subjects:"PSK"}} )

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

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

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

db.faculty.find( { }, { name: 1, qualification: 1 } ).pretty()

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} ).sort({exp:1}).pretty()

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)