MongoDB\_Lab1

1 – open mongo shell and view the help

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

show databases

use os\_42

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.students.insert ({name:"rowan",age:23,work:”sw-engineer”})**

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

Before making insert into the database the iti db did not appear but it appears after insertion

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 users =[{

name:"rana",

age:25,

work:"A”

},

{

name:"reham",

age:22,

work:"B"

}

]

db.students.insertMany(users)

6 – Search for your object by name.

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

7– Search for your friend(s) by age.

db.students.find({age:24})

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

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

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

db.students.insert({id:1,name:"rown",age:25})

db.students.deleteOne({id:1})

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()

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

**part 2**

1- Create database with name 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"})

{ "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"}

1. Get the details of all the faculty.

db.faculty.find()2. Get the count of all faculty members.

db.faculty.count()->10

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

db.faculty.find({“qualifications’’:”PH.D’})

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

db.faculty.find({“exp”:{$in:[8,12]}}

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

db.faculty.find({$or:[{“subjects”:”Maths”},{“subjects”:”NETWORKING”}])

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({$or[{“subjects”:”Maths”},{{“qualifications”:”PH.D”},{“age”:30}}})

.

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

db.faculty.find({$or:[{“type”:”part time”},{“subjects”:”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":"Sures 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({$set:{age:{$age+1},{“exp”:$exp+1}}})

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”},{“type”:”fulll time”}})

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

db.faculty.updatMany({$set:{“subjects”:[“MATHS”,”PSK”]}})

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

db.faculty.deleteMany({“age”:{$gt:50}})

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

db.faculty.find({name},{qualification})

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

db.faculty.find({“name”},{“qualification”},{“exp”}).sort(“exp”:-1)

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)