MongoDB – Lab1

Part1

1. open mongo shell and view the help
    - mongosh
    - help

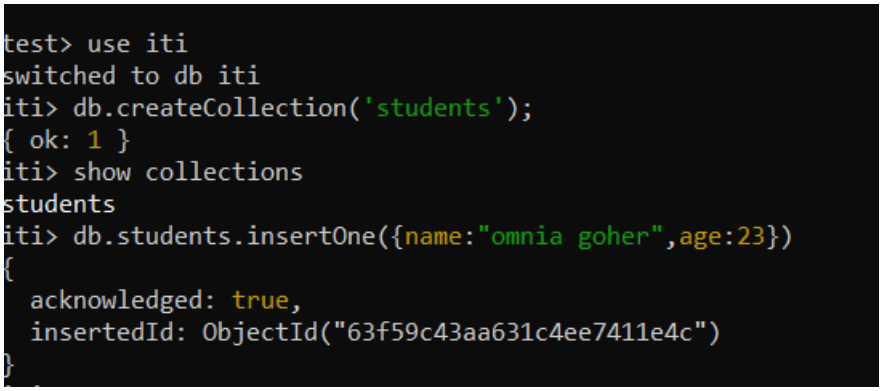
```
C:\Users\mera>mongosh
Current Mongosh Log ID: 63f5992564e4cb45aa7599d7
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.3.1
Using MongoDB:          6.0.4
Using Mongosh:          1.3.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting:
   2023-02-22T06:24:11.562+02:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

test> help

  Shell Help:

    use                                 Set current database
    show                                'show databases'/'show dbs': Print a list of all available databases.
                                        'show collections'/'show tables': Print a list of all collections for current database.
                                        'show profile': Prints system.profile information.
                                        'show users': Print a list of all users for current database.
                                        'show roles': Print a list of all roles for current database.
                                        'show log <type>': log for current connection, if type is not set uses 'global'
                                        'show logs': Print all logs.

    exit                                Quit the MongoDB shell with exit/exit()/.exit
    quit                                Quit the MongoDB shell with quit/quit()
    Mongo                               Create a new connection and return the Mongo object. Usage: new Mongo(URI, options [optional])
    connect                             Create a new connection and return the Database object. Usage: connect(URI, username [optional], password [optional])
    it                                  result of the last line evaluated; use to further iterate
    version                             Shell version
    load                                Loads and runs a JavaScript file into the current shell environment
    enableTelemetry                     Enables collection of anonymous usage data to improve the mongosh CLI
    disableTelemetry                    Disables collection of anonymous usage data to improve the mongosh CLI
    passwordPrompt                      Prompts the user for a password
    sleep                               Sleep for the specified number of milliseconds
    print                               Prints the contents of an object to the output
    printjson                           Alias for print()
    cls                                 Clears the screen like console.clear()
    isInteractive                       Returns whether the shell will enter or has entered interactive mode

  For more information on usage: https://docs.mongodb.com/manual/reference/method
test>
```

2. identify your current working database and show list of available databases
    - db
    - show dbs

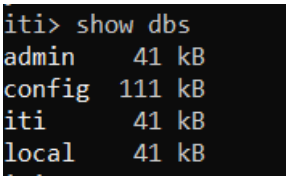
```
test> db
test
test> show dbs
admin     41 kB
config  12.3 kB
local     41 kB
test>
```

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');
    - show collections
    - db.students.insertOne({name:"omnia goher",age:23})



4. – show list of available databases. What did you notice?
    - Iti database created



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([
      {

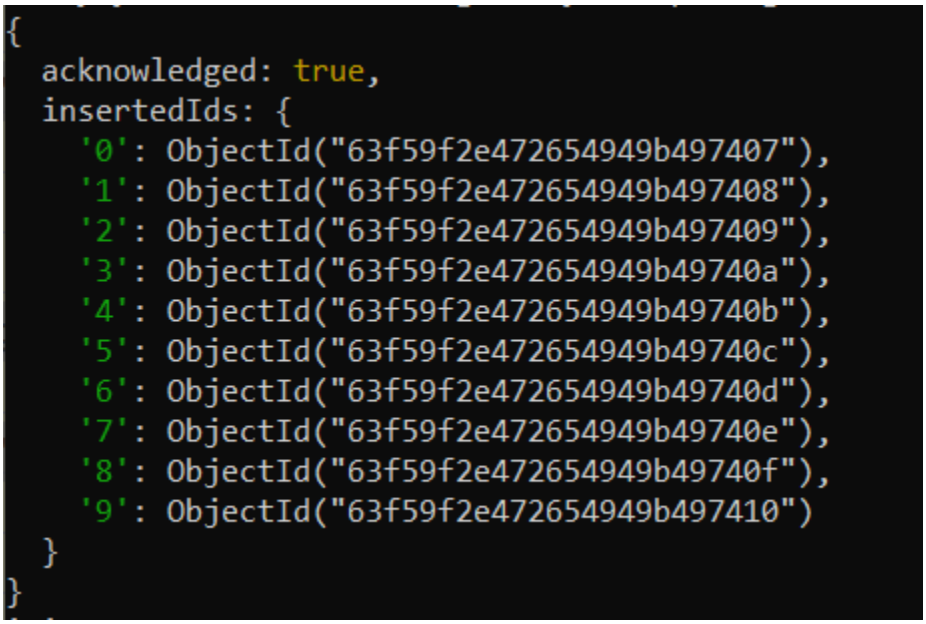
      name: "mariam reda",

      email: " mariam@gmail.com",

      age: 23,

      num: ["1", "2", "3"],

      },……..])



6. Search for your object by name.
    - db.students.find({name:"eman goher"});

```
iti> db.students.find({name:"eman goher"});
[
  {
    _id: ObjectId("63f59f2e472654949b497408"),
    name: 'eman goher',
    email: ' eman@gmail.com',
    age: 20,
    num: [ '4', '5', '6' ]
  }
]
```

7. Search for your friend(s) by age.
    - db.students.find({age:22});

```
iti> db.students.find({age:22});
[
  {
    _id: ObjectId("63f59f2e472654949b497409"),
    name: 'ahmed',
    email: 'ahmed@gmail.com',
    age: 22,
    num: [ '7', '8', '9' ]
  }
]
```

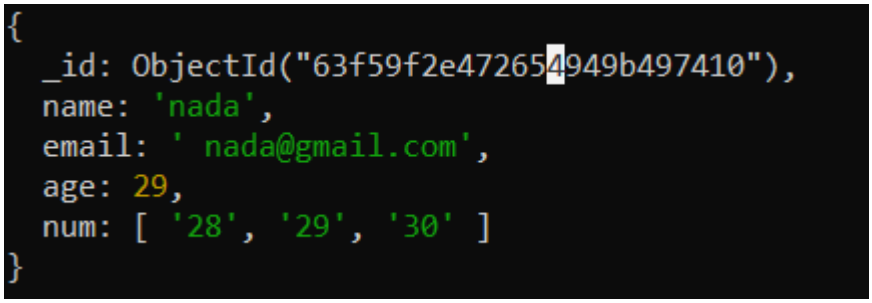
8. Search for all of your friends whose age is older than yours.
    - db.students.find({age:{$gt:23}});

```
iti> db.students.find({age:{$gt:23}});
[
  {
    _id: ObjectId("63f59f2e472654949b49740a"),
    name: 'ali',
    email: ' ali@gmail.com',
    age: 25,
    num: [ '10', '11', '12' ]
  },
  {
    _id: ObjectId("63f59f2e472654949b49740c"),
    name: 'mayar',
    email: ' mayar@gmail.com',
    age: 26,
    num: [ '16', '17', '18' ]
  },
  {
    _id: ObjectId("63f59f2e472654949b49740d"),
    name: 'rahma',
    email: ' rahma@gmail.com',
    age: 27,
    num: [ '19', '20', '21' ]
  },
  {
    _id: ObjectId("63f59f2e472654949b49740f"),
    name: 'wafaa',
    email: ' wafaa@gmail.com',
    age: 30,
    num: [ '25', '26', '27' ]
  },
  {
    _id: ObjectId("63f59f2e472654949b497410"),
    name: 'nada',
    email: ' nada@gmail.com',
    age: 29,
    num: [ '28', '29', '30' ]
  }
]
```

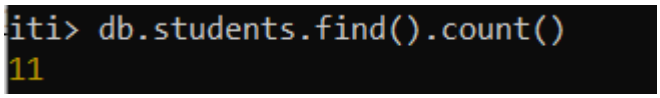
9. delete any of your friends by id.
    - db.students.deleteOne({_id:3})

```
iti> db.students.find({_id:3});
[ { _id: 3, name: 'hala ahmed', age: 22 } ]
iti> db.students.deleteOne({_id:3})
{ acknowledged: true, deletedCount: 1 }
```

10. view all documents in students collection in a prettified format.
    - db.students.find().pretty()



11. count all documents in students collection.
    - db.students.find().count()



Part2

1. Create database with name ems
    - use ems
    - db.createCollection("faculty");

```
iti> use ems
switched to db ems
ems> db.createCollection("faculty");
{ ok: 1 }
```

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"}

1) Get the details of all the faculty.
    - db.faculty.find()

```
{
  _id: ObjectId("63f5a7bcbaf8e206cb83c838"),
  name: 'Nagesh',
  age: 35,
  gender: 'M',
  exp: 9,
  subjects: [ 'JAVA', '.Net', 'NETWORKING' ],
  type: 'Full Time',
  qualification: 'Ph.D'
},
{
  _id: ObjectId("63f5a7bcbaf8e206cb83c839"),
  name: 'Latha',
  age: 40,
  gender: 'F',
  exp: 13,
  subjects: [ 'MATHS' ],
  type: 'Full Time',
  qualification: 'Ph.D'
}
```

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

```
ems> db.faculty.countDocuments()
10
```

3) Get all the faculty members whose qualification is “Ph.D”.
    - db.faculty.find({qualification:"Ph.D"})

```
ems> db.faculty.find({qualification:"Ph.D"})
[
  {
    _id: ObjectId("63f5a7bcbaf8e206cb83c831"),
    name: 'Manoj',
    age: 38,
    gender: 'M',
    exp: 12,
    subjects: [ 'JAVA', 'DBMS' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f5a7bcbaf8e206cb83c833"),
    name: 'Suresh',
    age: 40,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f5a7bcbaf8e206cb83c835"),
    name: 'Mani',
    age: 38,
    gender: 'F',
    exp: 10,
    subjects: [ 'JAVA', 'DBMS', 'OS' ],
    type: 'Part Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f5a7bcbaf8e206cb83c837"),
    name: 'Nagesh',
    age: 39,
    gender: 'M',
    exp: 11,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
```

4) Get all the faculty members whose experience is between 8 to 12 years.
    - db.faculty.find({exp:{$gt:8,$lt:12}})

```
ems> db.faculty.find({exp:{$gt:8,$lt:12}})
[   type: 'Full Time',
  { qualification: 'Ph.D'
    _id: ObjectId("63f5a7bcbaf8e206cb83c830"),
    name: 'Krish',
    age: 35,ectId("63f5a7bcbaf8e206cb83c838"),
    gender: 'M',h',
    exp: 10,
    subjects: [ 'DS', 'C', 'OS' ],
    type: 'Full Time',
    qualification: 'M.Tech'et', 'NETWORKING' ],
  },type: 'Full Time',
  { qualification: 'Ph.D'
    _id: ObjectId("63f5a7bcbaf8e206cb83c833"),
    name: 'Suresh',
    age: 40,ectId("63f5a7bcbaf8e206cb83c839"),
    gender: 'M',',
    exp: 9,,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D',
  },type: 'Full Time',
  { qualification: 'Ph.D'
    _id: ObjectId("63f5a7bcbaf8e206cb83c835"),
    name: 'Mani',
    age: 38,
    gender: 'F',
    exp: 10,
    subjects: [ 'JAVA', 'DBMS', 'OS' ],
    type: 'Part Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f5a7bcbaf8e206cb83c837"),
    name: 'Nagesh',
    age: 39,
    gender: 'M',
    exp: 11,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
```

5) Get all the faculty members who teach “MATHS” or “NETWORKING”
   - db.faculty.find({subjects:{$in:["MATHS","NETWORKING"]}})

```
ems> db.faculty.find({subjects:{$in:["MATHS","NETWORKING"]}})
[
  {
    _id: ObjectId("63f5a7bcbaf8e206cb83c833"),
    name: 'Suresh',
    age: 40,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f5a7bcbaf8e206cb83c836"),
    name: 'Sivani',
    age: 32,
    gender: 'F',
    exp: 8,
    subjects: [ 'C', 'CPP', 'MATHS' ],
    type: 'Part Time',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("63f5a7bcbaf8e206cb83c837"),
    name: 'Nagesh',
    age: 39,
    gender: 'M',
    exp: 11,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f5a7bcbaf8e206cb83c838"),
    name: 'Nagesh',
    age: 35,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', '.Net', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
```

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"})

```
ems>  db.faculty.find({subjects:"MATHS",age:{$gt:30},qualification:"Ph.D"})
[
  {
    _id: ObjectId("63f5a7bcbaf8e206cb83c839"),
    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"}]})

```
ems> db.faculty.find({$or:[{subjects:"JAVA"},{type:"Part Time"}]})
[
  {
    _id: ObjectId("63f5a7bcbaf8e206cb83c831"),
    name: 'Manoj',
    age: 38,
    gender: 'M',
    exp: 12,
    subjects: [ 'JAVA', 'DBMS' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f5a7bcbaf8e206cb83c832"),
    name: 'Anush',
    age: 32,
    gender: 'F',
    exp: 8,
    subjects: [ 'C', 'CPP' ],
    type: 'Part Time',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("63f5a7bcbaf8e206cb83c833"),
    name: 'Suresh',
    age: 40,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f5a7bcbaf8e206cb83c835"),
    name: 'Mani',
    age: 38,
    gender: 'F',
    exp: 10,
    subjects: [ 'JAVA', 'DBMS', 'OS' ],
    type: 'Part Time',
    qualification: 'Ph.D'
  },
```

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.insertOne({ "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"})
{
  acknowledged: true,
  insertedId: ObjectId("63f5ad26fc2467353316262d")
}
```

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}})

```
ems> db.faculty.updateMany({},{$inc:{age:1,exp:1}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 11,
  modifiedCount: 11,
  upsertedCount: 0
}
```

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",type:"Full Time"}})

```
ems> db.faculty.updateMany({name:"Sivani"},{$set:{qualification:"Ph.D",type:"Full Time"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

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"}})

```
ems> db.faculty.updateMany({subjects:"MATHS"},{$push:{subjects:"PSK"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0
}
```

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

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

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

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

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

```
ems> 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,_id:0}).sort({exp:1})

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

```
{
  _id: ObjectId("63f5a7bcbaf8e206cb83c839"),
  name: 'Latha',
  age: 41,
  gender: 'F',
  exp: 14,
  subjects: [ 'MATHS', 'PSK' ],
  type: 'Full Time',
  qualification: 'Ph.D'
},
{
  _id: ObjectId("63f5a7bcbaf8e206cb83c833"),
  name: 'Suresh',
  age: 41,
  gender: 'M',
  exp: 10,
  subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
  type: 'Full Time',
  qualification: 'Ph.D'
},
{
  _id: ObjectId("63f5a7bcbaf8e206cb83c837"),
  name: 'Nagesh',
  age: 40,
  gender: 'M',
  exp: 12,
  subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
  type: 'Full Time',
  qualification: 'Ph.D'
},
{
  _id: ObjectId("63f5a7bcbaf8e206cb83c835"),
  name: 'Mani',
  age: 39,
  gender: 'F',
  exp: 11,
  subjects: [ 'JAVA', 'DBMS', 'OS' ],
  type: 'Part Time',
  qualification: 'Ph.D'
},
{
  _id: ObjectId("63f5a7bcbaf8e206cb83c831"),
  name: 'Manoj',
  age: 39,
  gender: 'M',
  exp: 13,
  subjects: [ 'JAVA', 'DBMS' ],
  type: 'Full Time',
  qualification: 'Ph.D'
}
```