MongoDB_Lab1

1 – open mongo shell and view the help

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
C:\Users\K I N G\Downloads\mongosh-1.7.1-win32-x64\bin>mongosh
Current Mongosh Log ID: 63f4c88f386cf0dc9f1dac4c
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.7.1
Using MongoDB:          5.0.14
Using Mongosh:          1.7.1

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

------
   The server generated these startup warnings when booting
   2023-02-21T13:10:52.544+02:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

------
   Enable MongoDB's free cloud-based monitoring service, which will then receive and display
   metrics about your deployment (disk utilization, CPU, operation statistics, etc).

   The monitoring data will be available on a MongoDB website with a unique URL accessible to you
   and anyone you share the URL with. MongoDB may use this information to make product
   improvements and to suggest MongoDB products and deployment options to you.

   To enable free monitoring, run the following command: db.enableFreeMonitoring()
   To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
------

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
```

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

```
  For more information on usage: https://docs.mongodb.com/manual/reference/method
test> db.getName()
test
test> show dbs
admin    40.00 KiB
config  108.00 KiB
local    40.00 KiB
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).

```
test> use iti
switched to db iti
iti> db.students.insert({
... "Id":1,
... "Name":"sara"
... "Age":24})
Uncaught:
SyntaxError: Unexpected token, expected "," (4:0)

  2 | "Id":1,
  3 | "Name":"sara"
> 4 | "Age":24})
    | ^
  5 |

iti> db.students.insert({ "Id":1, "Name":"sara", "Age":24})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63f4ce5c45992a2a360af253") }
}
iti>
```

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

```
iti> show dbs
admin    40.00 KiB
config  108.00 KiB
iti      40.00 KiB
local    40.00 KiB
```

I notice that the new db iti added to the available databases as I insert collection in it

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).

```
iti> db.students.insert([{ "Name": "wafaa", "Age": 22, "Hobbies": ["Reading", "music"] }, { "Name": "omar", "Age": 24, "Hobbies": ["football", "reading"] }, { "Name": "ahmed", "Age": 26, "Hobbies": ["swimming", "music"] }, { "Name": "salma", "Age": 28, "Hobbies": ["gym", "reading"] }, { "Name": "mohamed", "Age": 22, "Hobbies": ["swimming", "gym"] }, { "Name": "sayed", "Age": 23 }, { "Name": "esraa", "Age": 23 }, { "Name": "nehad", "Age": 24 }, { "Name": "radwa", "Age": 25 }])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("63f4d6d845992a2a360af255"),
    '1': ObjectId("63f4d6d845992a2a360af256"),
    '2': ObjectId("63f4d6d845992a2a360af257"),
    '3': ObjectId("63f4d6d845992a2a360af258"),
    '4': ObjectId("63f4d6d845992a2a360af259"),
    '5': ObjectId("63f4d6d845992a2a360af25a"),
    '6': ObjectId("63f4d6d845992a2a360af25b"),
    '7': ObjectId("63f4d6d845992a2a360af25c"),
    '8': ObjectId("63f4d6d845992a2a360af25d")
  }
}
```

6 – Search for your object by name.

```
iti> db.students.find({"Name":"salma"})
[
  {
    _id: ObjectId("63f4d6d845992a2a360af258"),
    Name: 'salma',
    Age: 28,
    Hobbies: [ 'gym', 'reading' ]
  }
]
```

7– Search for your friend(s) by age.

```
iti> db.students.find({Age:24})
[
  {
    _id: ObjectId("63f4ce5c45992a2a360af253"),
    Id: 1,
    Name: 'sara',
    Age: 24
  },
  {
    _id: ObjectId("63f4d4b645992a2a360af254"),
    Name: 'sondos',
    Age: 24,
    Hobbies: [ 'Reading', 'swimming' ]
  },
  {
    _id: ObjectId("63f4d6d845992a2a360af256"),
    Name: 'omar',
    Age: 24,
    Hobbies: [ 'football', 'reading' ]
  },
  { _id: ObjectId("63f4d6d845992a2a360af25c"), Name: 'nehad', Age: 24 }
```

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

```
iti> db.students.find({Age:{$gt:24}})
[
  {
    _id: ObjectId("63f4d6d845992a2a360af257"),
    Name: 'ahmed',
    Age: 26,
    Hobbies: [ 'swimming', 'music' ]
  },
  {
    _id: ObjectId("63f4d6d845992a2a360af258"),
    Name: 'salma',
    Age: 28,
    Hobbies: [ 'gym', 'reading' ]
  },
  { _id: ObjectId("63f4d6d845992a2a360af25d"), Name: 'radwa', Age: 25 }
]
```

9 – delete any of your friends by id.

```
  },
  { _id: ObjectId("63f4d6d845992a2a360af25d"), Name: 'radwa', Age: 25 }
]
iti> db.students.remove({_id:ObjectId("63f4d6d845992a2a360af25d")})
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.
{ acknowledged: true, deletedCount: 1 }
iti> db.students.find({Age:{$gt:24}})
[
  {
    _id: ObjectId("63f4d6d845992a2a360af257"),
    Name: 'ahmed',
    Age: 26,
    Hobbies: [ 'swimming', 'music' ]
  },
  {
    _id: ObjectId("63f4d6d845992a2a360af258"),
    Name: 'salma',
    Age: 28,
    Hobbies: [ 'gym', 'reading' ]
```

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

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
]
iti> db.students.find().pretty()
[
  {
    _id: ObjectId("63f4ce5c45992a2a360af253"),
    Id: 1,
    Name: 'sara',
    Age: 24
  },
  {
    _id: ObjectId("63f4d4b645992a2a360af254"),
    Name: 'sondos',
    Age: 24,
    Hobbies: [ 'Reading', 'swimming' ]
  },
  {
    _id: ObjectId("63f4d6d845992a2a360af255"),
    Name: 'wafaa',
    Age: 22,
    Hobbies: [ 'Reading', 'music' ]
  },
  {
    _id: ObjectId("63f4d6d845992a2a360af256"),
    Name: 'omar',
    Age: 24,
    Hobbies: [ 'football', 'reading' ]
  },
  {
    _id: ObjectId("63f4d6d845992a2a360af257"),
    Name: 'ahmed',
    Age: 26,
    Hobbies: [ 'swimming', 'music' ]
  },
  {
    _id: ObjectId("63f4d6d845992a2a360af258"),
    Name: 'salma',
    Age: 28,
    Hobbies: [ 'gym', 'reading' ]
  },
  {
    _id: ObjectId("63f4d6d845992a2a360af259"),
    Name: 'mohamed',
    Age: 22,
    Hobbies: [ 'swimming', 'gym' ]
  },
  { _id: ObjectId("63f4d6d845992a2a360af25a"), Name: 'sayed', Age: 23 },
  { _id: ObjectId("63f4d6d845992a2a360af25b"), Name: 'esraa', Age: 23 },
  { _id: ObjectId("63f4d6d845992a2a360af25c"), Name: 'nehad', Age: 24 }
]
```

11 – count all documents in students collection.

```
]
iti> db.students.countDocuments()
10
iti>
```

---

part 2

1- Create database with name ems

```
iti> use ems
switched to db 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"}

```
ems> 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"}
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("63f4e09045992a2a360af25f"),
    '1': ObjectId("63f4e09045992a2a360af260"),
    '2': ObjectId("63f4e09045992a2a360af261"),
    '3': ObjectId("63f4e09045992a2a360af262"),
    '4': ObjectId("63f4e09045992a2a360af263"),
    '5': ObjectId("63f4e09045992a2a360af264"),
    '6': ObjectId("63f4e09045992a2a360af265"),
    '7': ObjectId("63f4e09045992a2a360af266"),
    '8': ObjectId("63f4e09045992a2a360af267"),
    '9': ObjectId("63f4e09045992a2a360af268")
  }
}
```

1. Get the details of all the faculty.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
ems> db.faculty.find()
[
  {
    _id: ObjectId("63f4e09045992a2a360af25f"),
    name: 'Krish',
    age: 35,
    gender: 'M',
    exp: 10,
    subjects: [ 'DS', 'C', 'OS' ],
    type: 'Full Time',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("63f4e09045992a2a360af260"),
    name: 'Manoj',
    age: 38,
    gender: 'M',
    exp: 12,
    subjects: [ 'JAVA', 'DBMS' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4e09045992a2a360af261"),
    name: 'Anush',
    age: 32,
    gender: 'F',
    exp: 8,
    subjects: [ 'C', 'CPP' ],
    type: 'Part Time',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("63f4e09045992a2a360af262"),
    name: 'Suresh',
    age: 40,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4e09045992a2a360af263"),
    name: 'Rajesh',
    age: 35,
    gender: 'M',
    exp: 7,
    subjects: [ 'DS', 'C', 'OS' ],
    type: 'Full Time',
```

2. Get the count of all faculty members.

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

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

```
ems> db.faculty.find({qualification:'Ph.D'})
[
  {
    _id: ObjectId("63f4e09045992a2a360af260"),
    name: 'Manoj',
    age: 38,
    gender: 'M',
    exp: 12,
    subjects: [ 'JAVA', 'DBMS' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4e09045992a2a360af262"),
    name: 'Suresh',
    age: 40,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4e09045992a2a360af264"),
    name: 'Mani',
    age: 38,
    gender: 'F',
    exp: 10,
    subjects: [ 'JAVA', 'DBMS', 'OS' ],
    type: 'Part Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4e09045992a2a360af266"),
    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.

```
  },
ems> db.faculty.find({exp:{$in:[8,12]}})
[  _id: ObjectId("63f4e09045992a2a360af267"),
  { name: 'Nagesh',
    _id: ObjectId("63f4e09045992a2a360af260"),
    name: 'Manoj',
    age: 38,
    gender: 'M','JAVA', '.Net', 'NETWORKING' ],
    exp: 12,ull Time',
    subjects: [ 'JAVA', 'DBMS' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },_id: ObjectId("63f4e09045992a2a360af268"),
  { name: 'Latha',
    _id: ObjectId("63f4e09045992a2a360af261"),
    name: 'Anush',
    age: 32,
    gender: 'F','MATHS' ],
    exp: 8,Full Time',
    subjects: [ 'C', 'CPP' ],
    type: 'Part Time',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("63f4e09045992a2a360af265"),
    name: 'Sivani',
    age: 32,
    gender: 'F',
```

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

```
ems> db.faculty.find({$or:[{subjects:"MATHS"},{subjects:"NETWORKING"}]})
[
  {
    _id: ObjectId("63f4e09045992a2a360af262"),
    name: 'Suresh',
    age: 40,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4e09045992a2a360af265"),
    name: 'Sivani',
    age: 32,
    gender: 'F',
    exp: 8,
    subjects: [ 'C', 'CPP', 'MATHS' ],
    type: 'Part Time',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("63f4e09045992a2a360af266"),
    name: 'Nagesh',
    age: 39,
    gender: 'M',
    exp: 11,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
```

6. 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'})
[
  {
    _id: ObjectId("63f4e09045992a2a360af268"),
    name: 'Latha',
    age: 40,
    gender: 'F',
    exp: 13,
    subjects: [ 'MATHS' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  }
]
ems>
```

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

```
ems> db.faculty.find({$or:[{subjects:"JAVA"},{type:"Part Time"}]})
[
  {
    _id: ObjectId("63f4e09045992a2a360af260"),
    name: 'Manoj',
    age: 38,
    gender: 'M',
    exp: 12,
    subjects: [ 'JAVA', 'DBMS' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4e09045992a2a360af261"),
    name: 'Anush',
    age: 32,
    gender: 'F',
    exp: 8,
    subjects: [ 'C', 'CPP' ],
    type: 'Part Time',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("63f4e09045992a2a360af262"),
    name: 'Suresh',
    age: 40,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full 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"}

```
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("63f4e6ff45992a2a360af269")
}
```

9. 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}})
{
  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”.

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

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

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

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

```
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.

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

15. 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)
[
  {
    _id: ObjectId("63f4e09045992a2a360af268"),
    name: 'Latha',
    age: 41,
    gender: 'F',
    exp: 14,
    subjects: [ 'MATHS', 'PSK' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4e09045992a2a360af262"),
    name: 'Suresh',
    age: 41,
    gender: 'M',
    exp: 10,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4e09045992a2a360af266"),
    name: 'Nagesh',
    age: 40,
    gender: 'M',
    exp: 12,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4e09045992a2a360af264"),
```