

MongoDB_Lab1

Name: Karim Ali Ramadan Ali

Track : Professional Web Development & BI (Intake 43) – Alex Branch - ITI

1 – open mongo shell and view the help

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Microsoft Windows [Version 10.0.19044.2728]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Arabtech>mongosh
Current Mongosh Log ID: 643a05a3641906574f936533
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.8.0
Using MongoDB:      6.0.5
Using Mongosh:      1.8.0

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

-----
  The server generated these startup warnings when booting
  2023-04-15T01:35:36.102+02:00: Access control is not enabled for the database. Read and write access to data and configuration is possible even on databases that require authentication. This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip 127.0.0.1 to disable this warning
  2023-04-15T01:35:36.103+02:00: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip 127.0.0.1 to disable this warning
-----

-----
  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>
```

```
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()
convertShardKeyToHashed Returns the hashed value for the input using the same hashing function as a hashed index.
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

```
❏ mongosh mongodb://127.0.0.1:27017/?directConnection=true&ss
```

```
test> show dbs
ITI                144.00 KiB
NurserySystem      380.00 KiB
admin              40.00 KiB
config             72.00 KiB
local              72.00 KiB
test> db
test
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).

```
iti> use ITI
switched to db ITI
```

```
ITI> db.createCollection("students")
{ ok: 1 }
```

```
ITI> db.students.insertOne( { _id:1,name:"Cristiano Ronaldo",age:37 } )
{ acknowledged: true, insertedId: 1 }
```

```
ITI> db.students.insertMany( [ { _id:2,name:"Sergio Ramos",age:38 } , { _id:3,name:"Vinisious Junior",age:24 } ] )
{ acknowledged: true, insertedIds: { '0': 2, '1': 3 } }
```

```
ITI> db.students.find({})
[
  { _id: 1, name: 'Cristiano Ronaldo', age: 37 },
  { _id: 2, name: 'Sergio Ramos', age: 38 },
  { _id: 3, name: 'Vinisious Junior', age: 24 }
]
```

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

```
ITI> show dbs
ITI                120.00 KiB
NurserySystem      380.00 KiB
admin              40.00 KiB
config             108.00 KiB
local              72.00 KiB
```

Database ITI 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).

```
db.students.insertMany( [
    { _id:4,name:"Karim Ali",age:26},
    { _id:5,name:"Mohamed Ahmed",age:26},
    { _id:6,name:"Hesham Abden",age:25},
    { _id:7,name:"Ahmed Hassanin",age:23},
    { _id:8,name:"Mohamed Moustafa",age:24}
])
```

```
ITI> db.students.insertMany( [{_id:4,name:"Karim Ali",age:26},{_id:5,name:"Mohamed Ahmed",age:26},{_id:6,name:"Hesham Abden",age:25},{_id:7,name:"Ahmed Hassanin",age:23},{_id:8,name:"Mohamed Moustafa",age:24}] )
{ acknowledged: true,
  insertedIds: { '0': 4, '1': 5, '2': 6, '3': 7, '4': 8 } }

ITI> db.students.insertMany( [
...   { _id:9,
...     name:"Ashraf Mohamed",
...     age:30,
...     Subjects:[{name:"Javascript",date:new Date(2022,11, 25, 3, 0)},
...               {name:"ASP.Net MVC",date:new Date(2021,10, 13, 4, 0)}
...     ]
...   },
...   { _id:10,
...     name:"Ahmed Hamza",
...     age:24,
...     Subjects:[{name:"MongoDB",date:new Date(2023,4, 7, 1, 0)},
...               {name:"XML",date:new Date(2022,12, 11, 4, 0)}
...     ]
...   },
... ] )
{ acknowledged: true, insertedIds: { '0': 9, '1': 10 } }
ITI>
```

```
ITI> db.students.find()
[
  { _id: 1, name: 'Cristiano Ronaldo', age: 37 },
  { _id: 2, name: 'Sergio Ramos', age: 38 },
  { _id: 3, name: 'Vinisious Junior', age: 24 },
  { _id: 4, name: 'Karim Ali', age: 26 },
  { _id: 5, name: 'Mohamed Ahmed', age: 26 },
  { _id: 6, name: 'Hesham Abden', age: 25 },
  { _id: 7, name: 'Ahmed Hassanin', age: 23 },
  { _id: 8, name: 'Mohamed Moustafa', age: 24 },
  {
    _id: 9,
    name: 'Ashraf Mohamed',
    age: 30,
    Subjects: [
      { name: 'Javascript', date: ISODate("2022-12-25T01:00:00.000Z") },
      {
        name: 'Asp.Net MVC',
        date: ISODate("2021-11-13T02:00:00.000Z")
      }
    ]
  },
  {
    _id: 10,
    name: 'Ahmed Hamza',
    age: 24,
    Subjects: [
      { name: 'MongoDB', date: ISODate("2023-05-06T23:00:00.000Z") },
      { name: 'XML', date: ISODate("2023-01-11T02:00:00.000Z") }
    ]
  }
]
ITI> _
```

6 – Search for your object by name.

```
ITI> db.students.find({name:"Ahmed Hamza"})
[
  {
    _id: 10,
    name: 'Ahmed Hamza',
    age: 24,
    Subjects: [
      { name: 'MongoDB', date: ISODate("2023-05-06T23:00:00.000Z") },
      { name: 'XML', date: ISODate("2023-01-11T02:00:00.000Z") }
    ]
  }
]
ITI>
```

7– Search for your friend(s) by age.

```
ITI> db.students.find({age:24})
[
  { _id: 3, name: 'Vinisious Junior', age: 24 },
  { _id: 8, name: 'Mohamed Moustafa', age: 24 },
  {
    _id: 10,
    name: 'Ahmed Hamza',
    age: 24,
    Subjects: [
      { name: 'MongoDB', date: ISODate("2023-05-06T23:00:00.000Z") },
      { name: 'XML', date: ISODate("2023-01-11T02:00:00.000Z") }
    ]
  }
]
ITI>
```

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

```
ITI> db.students.find({age:{$gt:26}})
[
  { _id: 1, name: 'Cristiano Ronaldo', age: 37 },
  { _id: 2, name: 'Sergio Ramos', age: 38 },
  {
    _id: 9,
    name: 'Ashraf Mohamed',
    age: 30,
    Subjects: [
      { name: 'Javascript', date: ISODate("2022-12-25T01:00:00.000Z") },
      {
        name: 'Asp.Net MVC',
        date: ISODate("2021-11-13T02:00:00.000Z")
      }
    ]
  }
]
```

9 – delete any of your friends by id.

```
ITI> db.students.deleteOne({_id:3})
{ acknowledged: true, deletedCount: 1 }
ITI> _
```

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

```
ITI> db.students.find({}).pretty()
[
  { _id: 1, name: 'Cristiano Ronaldo', age: 37 },
  { _id: 2, name: 'Sergio Ramos', age: 38 },
  { _id: 4, name: 'Karim Ali', age: 26 },
  { _id: 5, name: 'Mohamed Ahmed', age: 26 },
  { _id: 6, name: 'Hesham Abden', age: 25 },
  { _id: 7, name: 'Ahmed Hassanin', age: 23 },
  { _id: 8, name: 'Mohamed Moustafa', age: 24 },
  {
    _id: 9,
    name: 'Ashraf Mohamed',
    age: 30,
    Subjects: [
      { name: 'Javascript', date: ISODate("2022-12-25T01:00:00.000Z") },
      {
        name: 'Asp.Net MVC',
        date: ISODate("2021-11-13T02:00:00.000Z")
      }
    ]
  },
  {
    _id: 10,
    name: 'Ahmed Hamza',
    age: 24,
    Subjects: [
      { name: 'MongoDB', date: ISODate("2023-05-06T23:00:00.000Z") },
      { name: 'XML', date: ISODate("2023-01-11T02:00:00.000Z") }
    ]
  }
]
```

11 – count all documents in students' collection.

```
ITI> db.students.count({})
DeprecationWarning: Collection.count
9
ITI> db.students.countDocuments({})
9
ITI>
ITI> db.students.countDocuments({age:{$gt:26}})
3
ITI> _
```

part 2

- 1- Create database with name ems

```
ITI> use ems
switched to db ems
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> let facultyArray = [
...   { "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>
```



```
ems> db.faculty.insertMany(facultyArray)
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("643a198c630c4797534e5820"),
    '1': ObjectId("643a198c630c4797534e5821"),
    '2': ObjectId("643a198c630c4797534e5822"),
    '3': ObjectId("643a198c630c4797534e5823"),
    '4': ObjectId("643a198c630c4797534e5824"),
    '5': ObjectId("643a198c630c4797534e5825"),
    '6': ObjectId("643a198c630c4797534e5826"),
    '7': ObjectId("643a198c630c4797534e5827"),
    '8': ObjectId("643a198c630c4797534e5828"),
    '9': ObjectId("643a198c630c4797534e5829")
  }
}
ems>
```

```

ms> db.faculty.find({})

{
  _id: ObjectId("643a198c630c4797534e5820"),
  name: 'Krish',
  age: 35,
  gender: 'M',
  exp: 10,
  subjects: [ 'DS', 'C', 'OS' ],
  type: 'Full Time',
  qualification: 'M.Tech'
},
{
  _id: ObjectId("643a198c630c4797534e5821"),
  name: 'Manoj',
  age: 38,
  gender: 'M',
  exp: 12,
  subjects: [ 'JAVA', 'DBMS' ],
  type: 'Full Time',
  qualification: 'Ph.D'
},
{
  _id: ObjectId("643a198c630c4797534e5822"),
  name: 'Anush',
  age: 32,
  gender: 'F',
  exp: 8,
  subjects: [ 'C', 'CPP' ],
  type: 'Part Time',
  qualification: 'M.Tech'
},
{
  _id: ObjectId("643a198c630c4797534e5823"),
  name: 'Suresh',
  age: 40,
  gender: 'M',
  exp: 9,
  subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
  type: 'Full Time',
  qualification: 'Ph.D'
},
{
  _id: ObjectId("643a198c630c4797534e5824"),
  name: 'Rajesh',
  age: 35,
  gender: 'M',
  exp: 7,
  subjects: [ 'DS', 'C', 'OS' ],
  type: 'Full Time',
  qualification: 'M.Tech'
},
{
  _id: ObjectId("643a198c630c4797534e5825"),
  name: 'Mani',
  age: 38,
  gender: 'F',
  exp: 10,
  subjects: [ 'JAVA', 'DBMS', 'OS' ],
  type: 'Part Time',
  qualification: 'Ph.D'
},
{
  _id: ObjectId("643a198c630c4797534e5826"),
  name: 'Sivani',
  age: 32,
  gender: 'F',
  exp: 8,
  subjects: [ 'C', 'CPP' ],
  type: 'Part Time',
  qualification: 'M.Tech'
}

```

1. Get the details of all the faculty.

2. Get the count of all faculty members.

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

```

ems> db.faculty.find({qualification:"Ph.D"})
[
  {
    _id: ObjectId("643a198c630c4797534e5821"),
    name: 'Manoj',
    age: 38,
    gender: 'M',
    exp: 12,
    subjects: [ 'JAVA', 'DBMS' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5823"),
    name: 'Suresh',
    age: 40,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5825"),
    name: 'Mani',
    age: 38,
    gender: 'F',
    exp: 10,
    subjects: [ 'JAVA', 'DBMS', 'OS' ],
    type: 'Part Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5827"),
    name: 'Nagesh',
    age: 39,
    gender: 'M',
    exp: 11,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5828"),
    name: 'Nagesh',
    age: 35,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', '.Net', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5829"),
    name: 'Latha',
    age: 40,
    gender: 'F',
    exp: 13,
    subjects: [ 'MATHS' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  }
]

```

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

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

```
ems> db.faculty.find( { $and:[ {exp:{$gte:8}} , {exp:{$lte:12}} ] } )
[
  {
    _id: ObjectId("643a198c630c4797534e5820"),
    name: 'Krish',
    age: 35,
    gender: 'M',
    exp: 10,
    subjects: [ 'DS', 'C', 'OS' ],
    type: 'Full Time',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("643a198c630c4797534e5821"),
    name: 'Manoj',
    age: 38,
    gender: 'M',
    exp: 12,
    subjects: [ 'JAVA', 'DBMS' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5822"),
    name: 'Anush',
    age: 32,
    gender: 'F',
    exp: 8,
    subjects: [ 'C', 'CPP' ],
    type: 'Part Time',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("643a198c630c4797534e5823"),
    name: 'Suresh',
    age: 40,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
]
```

```
ems> db.faculty.find( { exp:{$gte:8} , exp:{$lte:12} } )
[
  {
    _id: ObjectId("643a198c630c4797534e5820"),
    name: 'Krish',
    age: 35,
    gender: 'M',
    exp: 10,
    subjects: [ 'DS', 'C', 'OS' ],
    type: 'Full Time',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("643a198c630c4797534e5821"),
    name: 'Manoj',
    age: 38,
    gender: 'M',
    exp: 12,
    subjects: [ 'JAVA', 'DBMS' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5822"),
    name: 'Anush',
    age: 32,
    gender: 'F',
    exp: 8,
    subjects: [ 'C', 'CPP' ],
    type: 'Part Time',
    qualification: 'M.Tech'
  },
  {
```

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

```
ems> db.faculty.find({ $or: [ {subjects:'MATHS'}, {subjects:'NETWORKING'} ] })
[
  {
    _id: ObjectId("643a198c630c4797534e5823"),
    name: 'Suresh',
    age: 40,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5826"),
    name: 'Sivani',
    age: 32,
    gender: 'F',
    exp: 8,
    subjects: [ 'C', 'CPP', 'MATHS' ],
    type: 'Part Time',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("643a198c630c4797534e5827"),
    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("643a198c630c4797534e5829"),
    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 : [{type: 'Part Time'}, {subjects: 'JAVA'}]})
[
  {
    _id: ObjectId("643a198c630c4797534e5821"),
    name: 'Manoj',
    age: 38,
    gender: 'M',
    exp: 12,
    subjects: [ 'JAVA', 'DBMS' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5822"),
    name: 'Anush',
    age: 32,
    gender: 'F',
    exp: 8,
    subjects: [ 'C', 'CPP' ],
    type: 'Part Time',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("643a198c630c4797534e5823"),
    name: 'Suresh',
    age: 40,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
  }
]
```


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"
})
```

```
Shell Output (Documents) x
→ → | 50 | Documents 1 to 1 |
{
  "acknowledged" : true,
  "insertedId" : ObjectId("644102a33ddd66ebed4e8aab")
}
```

```
db.faculty.findOne({name:"Suresh Babu"})
```

```
{
  "_id" : ObjectId("644102a33ddd66ebed4e8aab"),
  "name" : "Suresh Babu",
  "age" : 55.0,
  "gender" : "M",
  "exp" : 25.0,
  "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.

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

10. 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'}}  
... )  
{  
  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”.

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

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

```
ems> db.faculty.find(
...   {}, // criteria
...   { name:1, qualification:1 } // projection
... )
[
  {
    _id: ObjectId("643a198c630c4797534e5820"),
    name: 'Krish',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("643a198c630c4797534e5821"),
    name: 'Manoj',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5822"),
    name: 'Anush',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("643a198c630c4797534e5823"),
    name: 'Suresh',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5824"),
    name: 'Rajesh',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("643a198c630c4797534e5825"),
    name: 'Mani',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5826"),
    name: 'Sivani',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5827"),
    name: 'Nagesh',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5828"),
    name: 'Nagesh',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5829"),
    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(
...     {}, // criteria
...     { name:1 , qualification:1 , exp:1 },
... ).sort({exp:1})
[
  {
    _id: ObjectId("643a198c630c4797534e5824"),
    name: 'Rajesh',
    exp: 8,
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("643a198c630c4797534e5822"),
    name: 'Anush',
    exp: 9,
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("643a198c630c4797534e5826"),
    name: 'Sivani',
    exp: 9,
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5823"),
    name: 'Suresh',
    exp: 10,
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5828"),
    name: 'Nagesh',
    exp: 10,
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5820"),
    name: 'Krish',
    exp: 11,
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("643a198c630c4797534e5825"),
    name: 'Mani',
    exp: 11,
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5827"),
    name: 'Nagesh',
    exp: 12,
    qualification: 'Ph.D'
  },
]
```

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("643a198c630c4797534e5829"),
    name: 'Latha',
    age: 41,
    gender: 'F',
    exp: 14,
    subjects: [ 'MATHS', 'PSK' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5823"),
    name: 'Suresh',
    age: 41,
    gender: 'M',
    exp: 10,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5827"),
    name: 'Nagesh',
    age: 40,
    gender: 'M',
    exp: 12,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5825"),
    name: 'Mani',
    age: 39,
    gender: 'F',
    exp: 11,
    subjects: [ 'JAVA', 'DBMS', 'OS' ],
    type: 'Part Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("643a198c630c4797534e5821"),
    name: 'Manoj',
    age: 39,
    gender: 'M',
    exp: 13,
    subjects: [ 'JAVA', 'DBMS' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  }
]
```