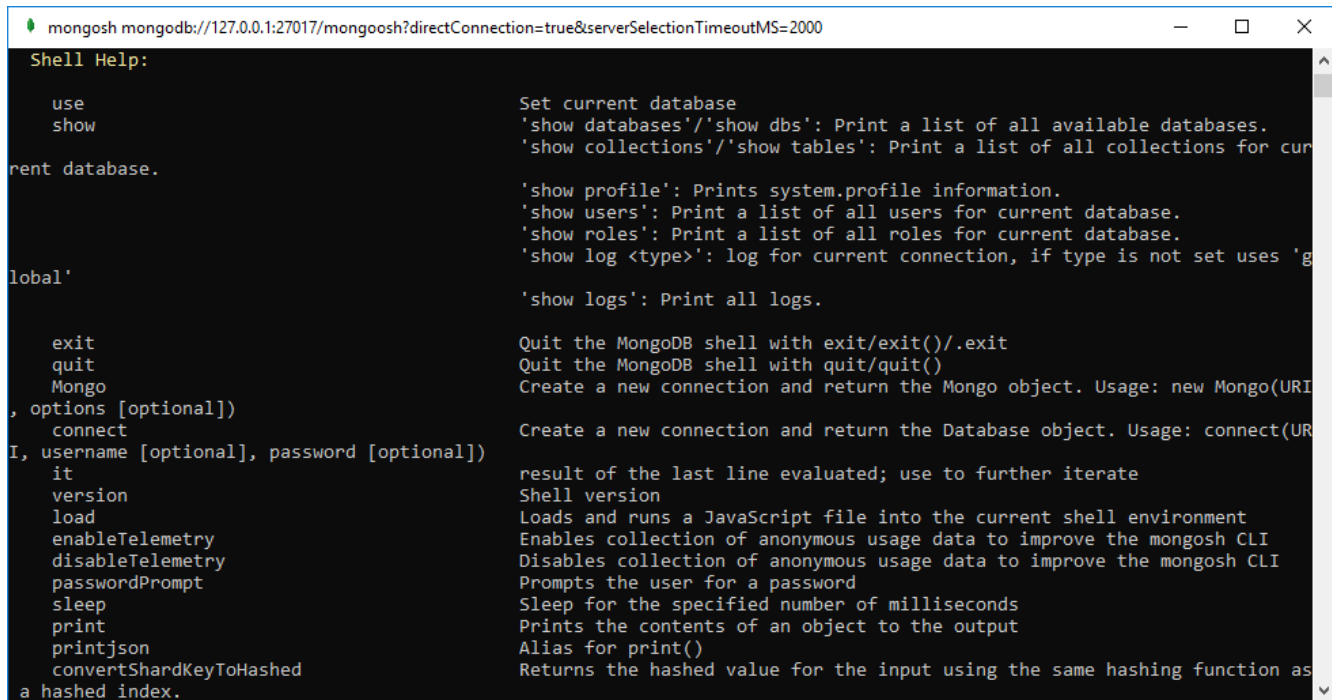


MongoDB_Lab1

1 – open mongo shell and view the help

Help

A screenshot of the MongoDB shell (mongosh) window. The title bar shows the connection string: 'mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000'. The main content area displays the 'Shell Help:' menu, which lists various commands and their descriptions in a two-column format. The commands listed include 'use', 'show', 'exit', 'quit', 'Mongo', 'connect', 'use', 'version', 'load', 'enableTelemetry', 'disableTelemetry', 'passwordPrompt', 'sleep', 'print', 'printjson', 'convertShardKeyToHashed', and 'a hashed index.'.

```
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 cur
               rent 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 'g
               lobal'
               '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(UR
I, 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.
```

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

db

show dbs

```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000

'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)
I, 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

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

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.insertOne({name:"Muhammad",age:26})`

```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000

I, 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

iti> db
iti
iti> show dbs
admin    40.00 KiB
config  108.00 KiB
local    40.00 KiB
iti> use iti
already on db iti
iti> db.students.insertOne({name:"Muhammad",age:26})
{
  acknowledged: true,
  insertedId: ObjectId("63f4dbe573b8a74a051a319a")
}
iti>
```

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

Show dbs

```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
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
iti> db
iti
iti> show dbs
admin 40.00 KiB
config 108.00 KiB
local 40.00 KiB
iti> use iti
already on db iti
iti> db.students.insertOne({name:"Muhammad",age:26})
{
  acknowledged: true,
  insertedId: ObjectId("63f4dbe573b8a74a051a319a")
}
iti> show dbs
admin 40.00 KiB
config 108.00 KiB
iti 8.00 KiB
local 40.00 KiB
iti>
```

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:"Muhammad", age:24, hobbies :
["running"] }, { name:"Ahmed", age:25, hobbies : ["swimming"] }, {
name:"Mahmoud", age:26, hobbies : ["football"] }, { name:"Youssef",
age:22, hobbies : ["basketball"] }, { name:"Saied", age:21, hobbies :
["vollyball"] }, { name:"Mazen", age:28, hobbies :
[{"writing":["poetry","jokes"]} ] }, { name:"Omar", age:27, hobbies :
["poetry"] }, { name:"Fahmy", age:24, hobbies : ["football","craft"] }, {
name:"Mostafa", age:24, hobbies : ["Music"] }, { name:"Samir",
age:28, hobbies : ["movies"] }, { name:"Waleed", age:24, hobbies :
["painting"] }, ] )
```

```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
2 |
iti> db.students.insertMany( [ { name:"Muhammad", age:24, hobbies : ["running"] }, { name:"Ahmed", age:25, hobbies : ["swimming"] }, { name:"Mahmoud", age:26, hobbies :
["football"] }, { name:"Youssef", age:22, hobbies : ["basketball"] }, { name:"Saied", age:21, hobbies : ["volleyball"] }, { name:"Mazen", age:28, hobbies : [{"writing":
"poetry","jokes"}] }, { name:"Omar", age:27, hobbies : ["poetry"] }, { name:"Fahmy", age:24, hobbies : ["football","craft"] }, { name:"Mostafa", age:24, hobbies : ["M
usic"] }, { name:"Samin", age:28, hobbies : ["movies"] }, { name:"Maieed", age:24, hobbies : ["painting"] }, ] )
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("63f4e68f73b8a74a051a319b"),
    '1': ObjectId("63f4e68f73b8a74a051a319c"),
    '2': ObjectId("63f4e68f73b8a74a051a319d"),
    '3': ObjectId("63f4e68f73b8a74a051a319e"),
    '4': ObjectId("63f4e68f73b8a74a051a319f"),
    '5': ObjectId("63f4e68f73b8a74a051a31a0"),
    '6': ObjectId("63f4e68f73b8a74a051a31a1"),
    '7': ObjectId("63f4e68f73b8a74a051a31a2"),
    '8': ObjectId("63f4e68f73b8a74a051a31a3"),
    '9': ObjectId("63f4e68f73b8a74a051a31a4"),
    '10': ObjectId("63f4e68f73b8a74a051a31a5")
  }
}
iti>
```

6 – Search for your object by name.

```
db.students.find({name:"Mazen"})
```

```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
]
iti> db.students.find({name:"mazen"})
iti> db.students.find({name:"Mazen"})
[
  {
    _id: ObjectId("63f4e68f73b8a74a051a31a0"),
    name: 'Mazen',
    age: 28,
    hobbies: [ { writing: [ 'poetry', 'jokes' ] } ]
  }
]
iti>
```

7– Search for your friend(s) by age.

```
db.students.find({age:26})
```

mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000

```
]
iti> db.students.find({name:"mazen"})
iti> db.students.find({name:"Mazen"})
[
  {
    _id: ObjectId("63f4e68f73b8a74a051a31a0"),
    name: 'Mazen',
    age: 28,
    hobbies: [ { writing: [ 'poetry', 'jokes' ] } ]
  }
]
iti> db.students.find({age:26})
[
  {
    _id: ObjectId("63f4dbe573b8a74a051a319a"),
    name: 'Muhammad',
    age: 26
  },
  {
    _id: ObjectId("63f4e68f73b8a74a051a319d"),
    name: 'Mahmoud',
    age: 26,
    hobbies: [ 'football' ]
  }
]
iti> _
```

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

`db.students.find({age:{$gt:25}})`

mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000

```
]
iti> db.students.find({age:{$gt:25}})
[
  {
    _id: ObjectId("63f4dbe573b8a74a051a319a"),
    name: 'Muhammad',
    age: 26
  },
  {
    _id: ObjectId("63f4e68f73b8a74a051a319d"),
    name: 'Mahmoud',
    age: 26,
    hobbies: [ 'football' ]
  }
]
iti> db.students.find({age:{$gt:25}})
[
  {
    _id: ObjectId("63f4dbe573b8a74a051a319a"),
    name: 'Muhammad',
    age: 26
  },
  {
    _id: ObjectId("63f4e68f73b8a74a051a319d"),
    name: 'Mahmoud',
    age: 26,
    hobbies: [ 'football' ]
  },
  {
    _id: ObjectId("63f4e68f73b8a74a051a31a0"),
    name: 'Mazen',
    age: 28,
    hobbies: [ { writing: [ 'poetry', 'jokes' ] } ]
  },
  {
    _id: ObjectId("63f4e68f73b8a74a051a31a1"),
    name: 'Omar',
    age: 27,
    hobbies: [ 'poetry' ]
  },
  {
    _id: ObjectId("63f4e68f73b8a74a051a31a4"),
    name: 'Samir',
    age: 28,
    hobbies: [ 'movies' ]
  }
]
iti>
```

9 – delete any of your friends by id.

`db.students.deleteOne({_id:ObjectId("63f4dbe573b8a74a051a319a")})`

```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
  name: 'Muhammad',
  age: 24,
  hobbies: [ 'running' ]
}
]
iti> db.students.deleteOne({_id:" 63f4dbe573b8a74a051a319a "})
{ acknowledged: true, deletedCount: 0 }
iti> db.students.deleteOne({_id:"63f4dbe573b8a74a051a319a"})
{ acknowledged: true, deletedCount: 0 }
iti> db.students.find({"name":"Muhammad"})
[
  {
    _id: ObjectId("63f4dbe573b8a74a051a319a"),
    name: 'Muhammad',
    age: 26
  },
  {
    _id: ObjectId("63f4e68f73b8a74a051a319b"),
    name: 'Muhammad',
    age: 24,
    hobbies: [ 'running' ]
  }
]
iti> db.students.deleteOne({_id:"63f4dbe573b8a74a051a319a"})
{ acknowledged: true, deletedCount: 0 }
iti> db.students.deleteOne({_id:"63f4e68f73b8a74a051a319b"})
{ acknowledged: true, deletedCount: 0 }
iti> db.students.deleteOne({_id:ObjectId("63f4dbe573b8a74a051a319a")})
{ acknowledged: true, deletedCount: 1 }
iti>
```

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

`db.students.find({}).pretty()`

```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
iti> db.students.find({}).pretty()
[
  {
    _id: ObjectId("63f4e68f73b8a74a051a319b"),
    name: 'Muhammad',
    age: 24,
    hobbies: [ 'running' ]
  },
  {
    _id: ObjectId("63f4e68f73b8a74a051a319c"),
    name: 'Ahmed',
    age: 25,
    hobbies: [ 'swimming' ]
  },
  {
    _id: ObjectId("63f4e68f73b8a74a051a319d"),
    name: 'Mahmoud',
    age: 26,
    hobbies: [ 'football' ]
  },
  {
    _id: ObjectId("63f4e68f73b8a74a051a319e"),
    name: 'Youssef',
    age: 22,
    hobbies: [ 'basketball' ]
  },
  {
    _id: ObjectId("63f4e68f73b8a74a051a319f"),
    name: 'Saied',
    age: 21,
  }
]
```

11 – count all documents in students collection.

`db.students.find({}).count()`

```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
age: 24,
hobbies: [ 'painting' ]
}
]
iti> db.students.find({}).count()
11
iti> _
```

part 2

- 1- Create database with name ems
use ems

```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
  age: 24,
  hobbies: [ 'painting' ]
}
]
iti> db.students.find({}).count()
11
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" }
```

db.createCollection("faculty")


```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
{"qualification":"Ph.D"}, {"name":"Latha","age":40,"gender":"F","exp":13,subjects:["MATHS"],"type":"FullTime","qualificati
on":"Ph.D"})
MongoInvalidArgumentError: Argument "docs" must be an array of documents
ems> db.faculty.insertMany([{"name":"Krish","age":35,"gender":"M","exp":10,subjects:["DS","C","OS"],"type":"FullTime","q
ualification":"M.Tech"}, {"name":"Manoj","age":38,"gender":"M","exp":12,subjects:["JAVA","DBMS"],"type":"FullTime","quali
fication":"Ph.D"}, {"name":"Anush","age":32,"gender":"F","exp":8,subjects:["C","CPP"],"type":"PartTime","qualification":"
M.Tech"}, {"name":"Suresh","age":40,"gender":"M","exp":9,subjects:["JAVA","DBMS","NETWORKING"],"type":"FullTime","qualifi
cation":"Ph.D"}, {"name":"Rajesh","age":35,"gender":"M","exp":7,subjects:["DS","C","OS"],"type":"FullTime","qualification
":"M.Tech"}, {"name":"Mani","age":38,"gender":"F","exp":10,subjects:["JAVA","DBMS","OS"],"type":"PartTime","qualification
":"Ph.D"}, {"name":"Sivani","age":32,"gender":"F","exp":8,subjects:["C","CPP","MATHS"],"type":"PartTime","qualification":
"M.Tech"}, {"name":"Nagesh","age":39,"gender":"M","exp":11,subjects:["JAVA","DBMS","NETWORKING"],"type":"FullTime","quali
fication":"Ph.D"}, {"name":"Nagesh","age":35,"gender":"M","exp":9,subjects:["JAVA",".Net","NETWORKING"],"type":"FullTime
","qualification":"Ph.D"}, {"name":"Latha","age":40,"gender":"F","exp":13,subjects:["MATHS"],"type":"FullTime","qualificat
ion":"Ph.D"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("63f4ec82603b706d50fecdd2"),
    '1': ObjectId("63f4ec82603b706d50fecdd3"),
    '2': ObjectId("63f4ec82603b706d50fecdd4"),
    '3': ObjectId("63f4ec82603b706d50fecdd5"),
    '4': ObjectId("63f4ec82603b706d50fecdd6"),
    '5': ObjectId("63f4ec82603b706d50fecdd7"),
    '6': ObjectId("63f4ec82603b706d50fecdd8"),
    '7': ObjectId("63f4ec82603b706d50fecdd9"),
    '8': ObjectId("63f4ec82603b706d50fecdda"),
    '9': ObjectId("63f4ec82603b706d50fecddb")
  }
}
ems>
```

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

```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
}
ems> db.faculty.find()
[
  {
    _id: ObjectId("63f4ec82603b706d50fecdd2"),
    name: 'Krish',
    age: 35,
    gender: 'M',
    exp: 10,
    subjects: [ 'DS', 'C', 'OS' ],
    type: 'FullTime',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("63f4ec82603b706d50fecdd3"),
    name: 'Manoj',
    age: 38,
    gender: 'M',
    exp: 12,
    subjects: [ 'JAVA', 'DBMS' ],
    type: 'FullTime',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ec82603b706d50fecdd4"),
    name: 'Anush',
    age: 32,
    gender: 'F',
    exp: 8,
    subjects: [ 'C', 'CPP' ],
    type: 'PartTime',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("63f4ec82603b706d50fecdd5"),
    name: 'Suresh',
    age: 40,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'FullTime',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ec82603b706d50fecdd6"),
    name: 'Rajesh',
    age: 35,
    gender: 'M',
    exp: 7,
    subjects: [ 'DS', 'C', 'OS' ],
    type: 'FullTime',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("63f4ec82603b706d50fecdd7"),
    name: 'Mani',
    age: 38,
    gender: 'F',
    exp: 10,
    subjects: [ 'JAVA', 'DBMS', 'OS' ],
    type: 'PartTime',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ec82603b706d50fecdd8"),
    name: 'Sivani',
    age: 32,
    gender: 'F',
    exp: 8,
    subjects: [ 'C', 'CPP', 'MATHS' ],
    type: 'PartTime',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("63f4ec82603b706d50fecdd9"),
    name: 'Nagesh',
    age: 39,
    gender: 'M',
    exp: 11,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'FullTime',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ec82603b706d50fecdda"),
    name: 'Nagesh',
    age: 35,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', '.Net', 'NETWORKING' ],
    type: 'FullTime',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ec82603b706d50fecddb"),
    name: 'Latha',
    age: 40,
    gender: 'F',
    exp: 13,
    subjects: [ 'MATHS' ],
    type: 'FullTime',
    qualification: 'Ph.D'
  }
]
```

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

```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
ems> db.faculty.find().count()
10
ems>
```

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

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

```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
10
ems> db.faculty.find({qualification:"Ph.D"})
[
  {
    _id: ObjectId("63f4ec82603b706d50fecdd3"),
    name: 'Manoj',
    age: 38,
    gender: 'M',
    exp: 12,
    subjects: [ 'JAVA', 'DBMS' ],
    type: 'FullTime',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ec82603b706d50fecdd5"),
    name: 'Suresh',
    age: 40,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'FullTime',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ec82603b706d50fecdd7"),
    name: 'Mani',
    age: 38,
    gender: 'F',
    exp: 10,
    subjects: [ 'JAVA', 'DBMS', 'OS' ],
    type: 'FullTime',
    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}})`

```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
]
ems> db.faculty.find({exp:{$gt:8,$lt:12}})
[
  {
    _id: ObjectId("63f4ec82603b706d50fecdd2"),
    name: 'Krish',
    age: 35,
    gender: 'M',
    exp: 10,
    subjects: [ 'DS', 'C', 'OS' ],
    type: 'FullTime',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("63f4ec82603b706d50fecdd5"),
    name: 'Suresh',
    age: 40,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'FullTime',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ec82603b706d50fecdd7"),
    name: 'Mani',
    age: 38,
    gender: 'F',
    exp: 10,
    subjects: [ 'JAVA', 'DBMS', 'OS' ],
    type: 'FullTime',
    qualification: 'Ph.D'
  }
]
```

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

```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
]
ems> db.faculty.find({$or:[{subjects:"MATH"},{subjects:"NETWORKING"}]})
[
  {
    _id: ObjectId("63f4ec82603b706d50fecdd5"),
    name: 'Suresh',
    age: 40,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'FullTime',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ec82603b706d50fecdd9"),
    name: 'Nagesh',
    age: 39,
    gender: 'M',
    exp: 11,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'FullTime',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ec82603b706d50fecdda"),
    name: 'Nagesh',
    age: 35,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', '.Net', 'NETWORKING' ],
    type: 'FullTime',
    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"})`

```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
type: 'FullTime',
qualification: 'Ph.D'
}
]
ems> db.faculty.find({subjects:"MATHS",age:{$gt:30},qualification:"Ph.D"})
[
  {
    _id: ObjectId("63f4ec82603b706d50fecddb"),
    name: 'Latha',
    age: 40,
    gender: 'F',
    exp: 13,
    subjects: [ 'MATHS' ],
    type: 'FullTime',
    qualification: 'Ph.D'
  }
]
ems>
```

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

`db.faculty.find({ $or: [{ subjects: "JAVA" }, { type: "PartTime" }] }, { name: 1, subjects: 1, type: 1, _id: 0 })`

```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
subjects: [ 'JAVA', '.Net', 'NETWORKING' ],
type: 'FullTime'
}
]
ems> db.faculty.find({ $or: [{ subjects: "JAVA" }, { type: "PartTime" } ] }, { name: 1, subjects: 1, type: 1, _id: 0 })
[
  { name: 'Manoj', subjects: [ 'JAVA', 'DBMS' ], type: 'FullTime' },
  { name: 'Anush', subjects: [ 'C', 'CPP' ], type: 'PartTime' },
  {
    name: 'Suresh',
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'FullTime'
  },
  {
    name: 'Mani',
    subjects: [ 'JAVA', 'DBMS', 'OS' ],
    type: 'PartTime'
  },
  {
    name: 'Sivani',
    subjects: [ 'C', 'CPP', 'MATHS' ],
    type: 'PartTime'
  },
  {
    name: 'Nagesh',
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'FullTime'
  },
  {
    name: 'Nagesh',

```

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":"FullTime","qualification":"Ph.D"})
```

A screenshot of a MongoDB shell window. The title bar shows the connection string: 'mongosh mongod://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000'. The shell prompt is 'ems>'. The user has entered a command to insert a new faculty member into the 'faculty' collection. The command is: 'db.faculty.insertOne({"name":"Suresh Babu","age":55,"gender":"M","exp":25,subjects:["MATHS","DE"],"type":"FullTime","qualification":"Ph.D"})'. The shell has responded with a confirmation object: '{ acknowledged: true, insertedId: ObjectId("63f4f066603b706d50fecddc") }'.

```
mongosh mongod://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
subjects: [ 'JAVA', '.Net', 'NETWORKING' ],
type: 'FullTime'
}
ems> db.faculty.insertOne({"name":"Suresh Babu","age":55,"gender":"M","exp":25,subjects:["MATHS","DE"],"type":"FullTime",
,"qualification":"Ph.D"})
{
  acknowledged: true,
  insertedId: ObjectId("63f4f066603b706d50fecddc")
}
ems>
```

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}})
```

```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
ems> db.faculty.updateMany({},{$inc:{age:1,exp:1}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 11,
  modifiedCount: 11,
  upsertedCount: 0
}
ems> _
```

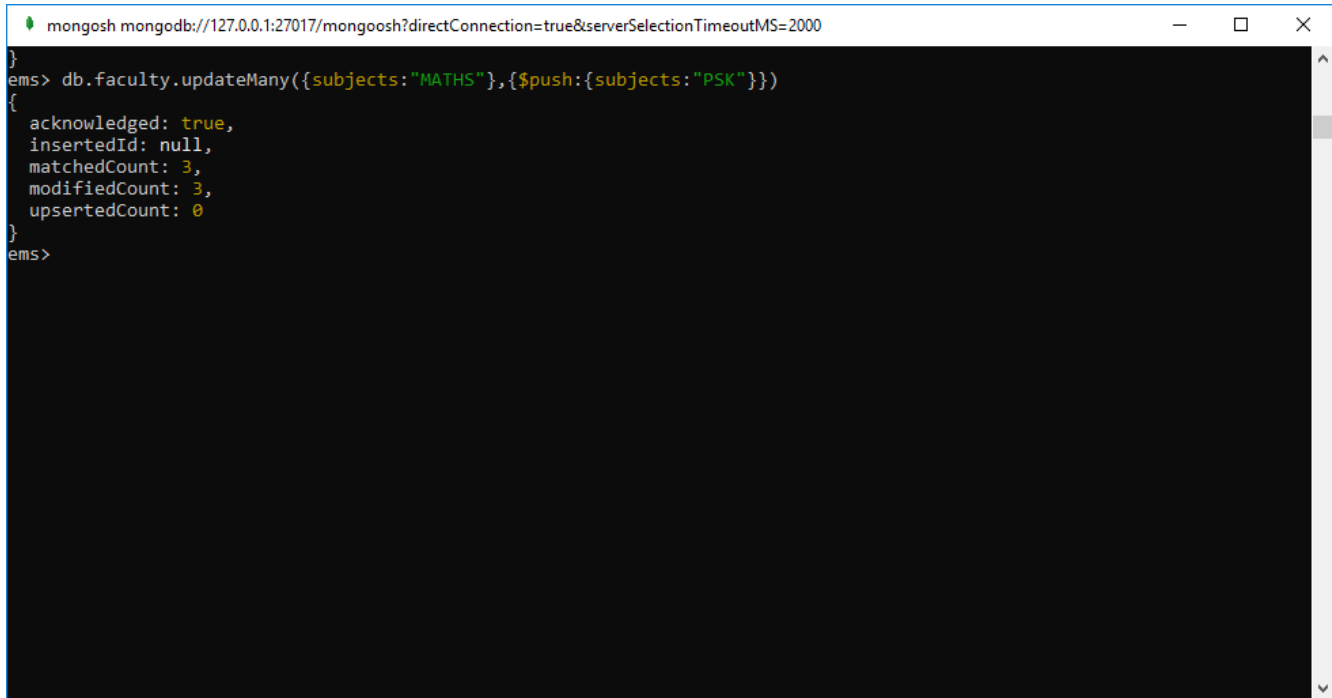
10. Update the faculty “Sivani” with the following data: update qualification to “Ph.D” and type to “Full Time”.

```
db.faculty.updateOne({name:"Sivani"},{$set:{qualification:"Ph.D",type:"FullTime"}})
```

```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
modifiedCount: 11,
upsertedCount: 0
}
ems> db.faculty.updateOne({name:"Sivani"},{$set:{qualification:"Ph.D",type:"FullTime"}})
{
  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”.

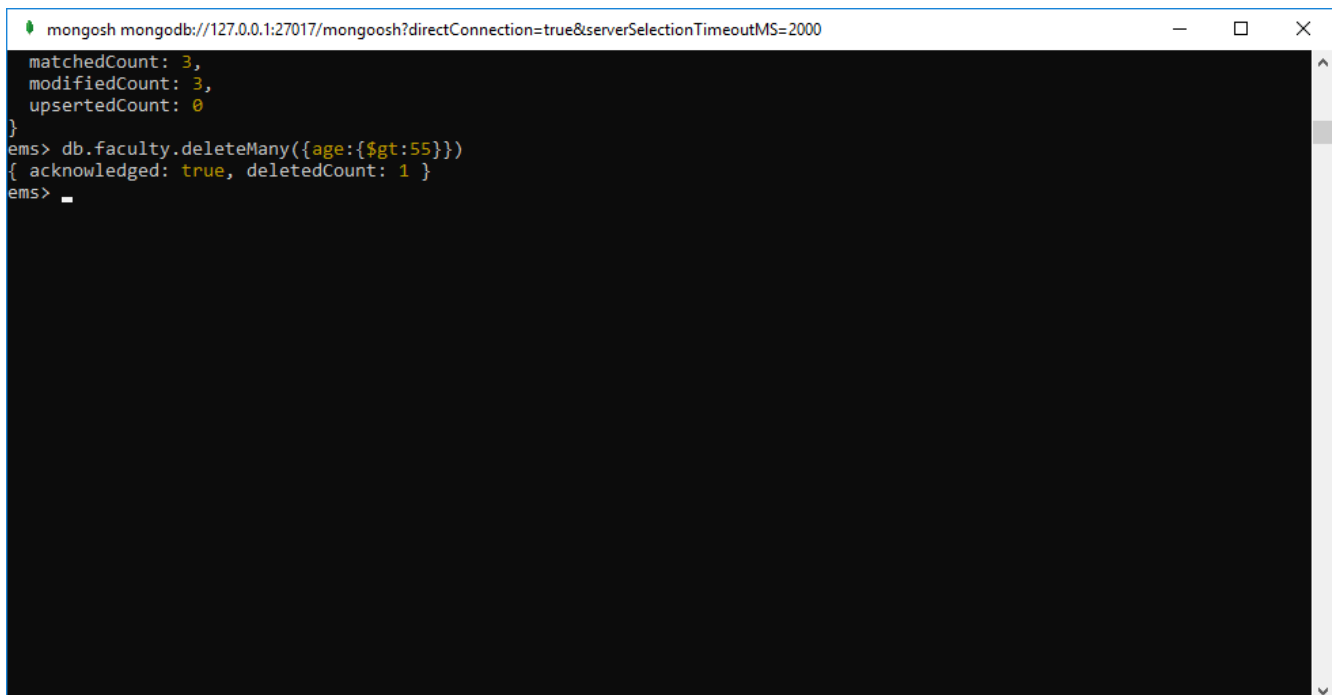
```
db.faculty.updateMany({subjects:"MATHS"},{$push:{subjects:"PSK"}})
```



```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
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}})
```



```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
matchedCount: 3,
modifiedCount: 3,
upsertedCount: 0
}
ems> db.faculty.deleteMany({age:{$gt:55}})
{ acknowledged: true, deletedCount: 1 }
ems>
```

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

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

```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
{ acknowledged: true, deletedCount: 1 }
ems> db.faculty.find({}, {_id:0,name:1,qualification:1})
[
  { 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' }
]
ems> _
```

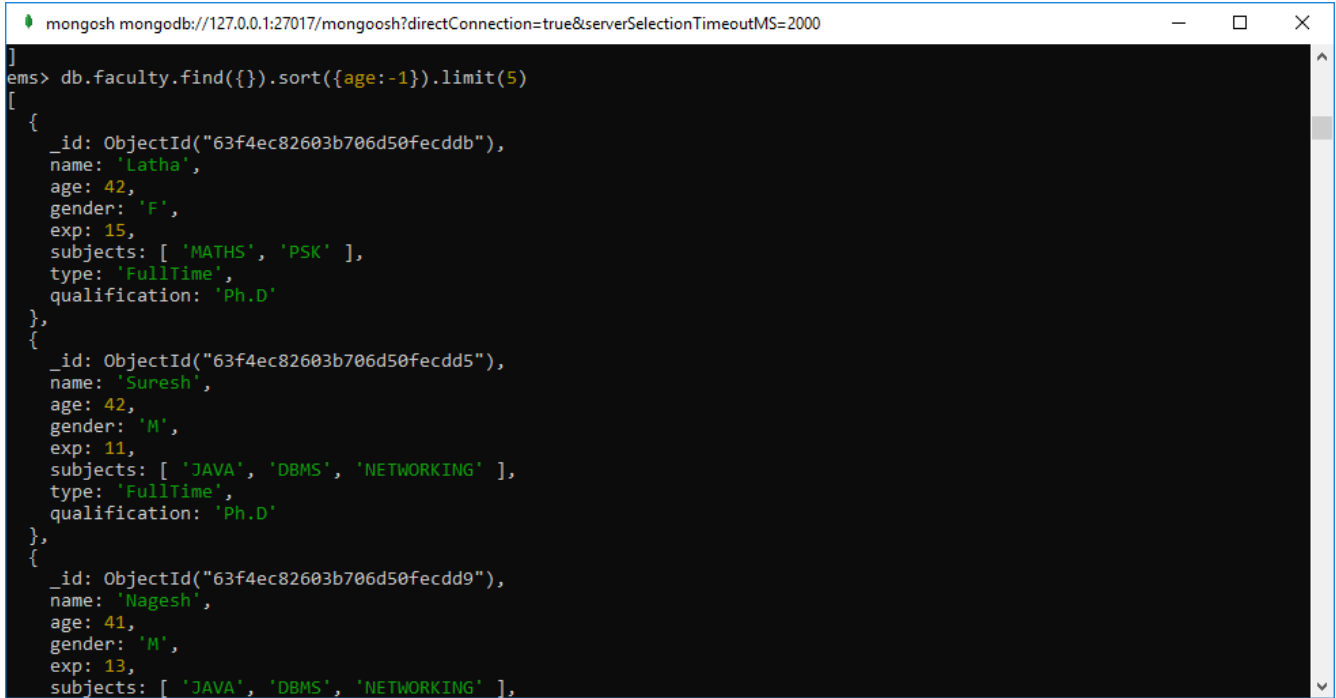
14. Get the name, qualification and exp of all faculty members and display the same in ascending order of exp.

```
db.faculty.find({}, {_id:0,name:1,qualification:1,exp:1}).sort({exp:1})
```

```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
{ name: 'Nagesh', qualification: 'Ph.D' },
{ name: 'Nagesh', qualification: 'Ph.D' },
{ name: 'Latha', qualification: 'Ph.D' }
]
ems> db.faculty.find({}, {_id:0,name:1,qualification:1,exp:1}).sort({exp:1})
[
  { name: 'Rajesh', exp: 9, qualification: 'M.Tech' },
  { name: 'Anush', exp: 10, qualification: 'M.Tech' },
  { name: 'Sivani', exp: 10, qualification: 'Ph.D' },
  { name: 'Suresh', exp: 11, qualification: 'Ph.D' },
  { name: 'Nagesh', exp: 11, qualification: 'Ph.D' },
  { name: 'Krish', exp: 12, qualification: 'M.Tech' },
  { name: 'Mani', exp: 12, qualification: 'Ph.D' },
  { name: 'Nagesh', exp: 13, qualification: 'Ph.D' },
  { name: 'Manoj', exp: 14, qualification: 'Ph.D' },
  { name: 'Latha', exp: 15, qualification: 'Ph.D' }
]
ems>
```


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)
```



The screenshot shows a MongoDB shell window with the following content:

```
mongosh mongodb://127.0.0.1:27017/mongoosh?directConnection=true&serverSelectionTimeoutMS=2000
ems> db.faculty.find({}).sort({age:-1}).limit(5)
[
  {
    _id: ObjectId("63f4ec82603b706d50fecddb"),
    name: 'Latha',
    age: 42,
    gender: 'F',
    exp: 15,
    subjects: [ 'MATHS', 'PSK' ],
    type: 'FullTime',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ec82603b706d50fecdd5"),
    name: 'Suresh',
    age: 42,
    gender: 'M',
    exp: 11,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'FullTime',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ec82603b706d50fecdd9"),
    name: 'Nagesh',
    age: 41,
    gender: 'M',
    exp: 13,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],

```