

MongoDB_Lab1

1 – open mongo shell and view the help

help

```
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])
```

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

db

test

show dbs

admin 40.00 KiB

config 108.00 KiB

local 40.00 KiB

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

switched to db iti

db.students.insertOne({ firstName: "Raneen",

... lastName: "Mahmoud",

... })

{

acknowledged: true,

insertedId: ObjectId("63f4c8cdc05fbd4f1109a50c")

}

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

show dbs

admin 40.00 KiB

config 108.00 KiB

iti 40.00 KiB

local 40.00 KiB

Notice: database iti created by me shows only when it has collection and at least one

document

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": "fayza", "age": 24 },
  { "name": "Mohamed", "courses": ["JS", "CSS"] },
  { "name": "Menna", "age": 22, "salary": 4000 },
  { name: "Zynab", age: 25 },
  { name: "Raneen", age: 23 }
])
```

6 – Search for your object by name.

```
db.students.findOne({name: "Mohamed"})
```

```
iti> db.students.findOne({name: "Mohamed"})
{
  _id: ObjectId("63f4ce7c61f16d96808c4495"),
  name: 'Mohamed',
  courses: [ 'JS', 'CSS' ]
}
iti>
```

7– Search for your friend(s) by age.

```
db.students.findOne({age: 22})
```

```
iti> db.students.findOne({age: 22})
{
  _id: ObjectId("63f4ce7c61f16d96808c4496"),
  name: 'Menna',
  age: 22,
  salary: 4000
}
```

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

```
iti> db.students.find({age: {$gt:23}})
```

```
2 |
iti> db.students.find({age: {$gt:23}})
[
  { _id: ObjectId("63f4ce7c61f16d96808c4494"), name: 'fayza', age: 24 },
  { _id: ObjectId("63f4d27a61f16d96808c4498"), name: 'Zynab', age: 25 }
]
iti> db.students.delete({_id:63f4d27a61f16d96808c4498})
```

9 – delete any of your friends by id.

```
db.students.deleteOne({_id:ObjectId("63f4d27a61f16d96808c4498")})
{ acknowledged: true, deletedCount: 1 }
```

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

```
C:\> mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
```

```
{ acknowledged: true, deletedCount: 0 }
iti> db.students.deleteOne({id:ObjectId("63f4d27a61f16d96808c4498")})
{ acknowledged: true, deletedCount: 0 }
iti> db.students.deleteOne({_id:ObjectId("63f4d27a61f16d96808c4498")})
{ acknowledged: true, deletedCount: 1 }
iti> db.students.fund()
TypeError: db.students.fund is not a function
iti> db.students.find()
[
  {
    _id: ObjectId("63f4c8cdc05fbd4f1109a50c"),
    firstName: 'Raneen',
    lastName: 'Mahmoud'
  },
  { _id: ObjectId("63f4ce7c61f16d96808c4494"), name: 'fayza', age: 24 },
  {
    _id: ObjectId("63f4ce7c61f16d96808c4495"),
    name: 'Mohamed',
    courses: [ 'JS', 'CSS' ]
  },
  {
    _id: ObjectId("63f4ce7c61f16d96808c4496"),
    name: 'Menna',
    age: 22,
    salary: 4000
  },
  {
    _id: ObjectId("63f4d25e61f16d96808c4497"),
    name: 'Raneen',
    age: 23
  }
]
```

11 – count all documents in students collection.

```
iti> db.students.count()
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
5
iti>
```

part 2

1- Create database with name ems

use 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": "Ful  
l 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.

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("63f4ee2e89c337b00bef6bf8"),
    '1': ObjectId("63f4ee2e89c337b00bef6bf9"),
    '2': ObjectId("63f4ee2e89c337b00bef6bfa"),
    '3': ObjectId("63f4ee2e89c337b00bef6bfb"),
    '4': ObjectId("63f4ee2e89c337b00bef6bfc"),
    '5': ObjectId("63f4ee2e89c337b00bef6bfd"),
    '6': ObjectId("63f4ee2e89c337b00bef6bfe"),
    '7': ObjectId("63f4ee2e89c337b00bef6bff"),
    '8': ObjectId("63f4ee2e89c337b00bef6c00"),
    '9': ObjectId("63f4ee2e89c337b00bef6c01")
  }
}
ems> db.faculty.find()
[
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bf8"),
    name: 'Krish',
    age: 35,
    gender: 'M',
    exp: 10,
    subjects: [ 'DS', 'C', 'OS' ],
    type: 'Full Time',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bf9"),
    name: 'Manoj',
    age: 38,
    gender: 'M',
    exp: 12,
    subjects: [ 'JAVA', 'DBMS' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bfa"),
    name: 'Ravi',
    age: 32,
    gender: 'M',
    exp: 8,
    subjects: [ 'OS', 'C', 'DBMS' ],
    type: 'Part Time',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bfb"),
    name: 'Anjali',
    age: 28,
    gender: 'F',
    exp: 5,
    subjects: [ 'C', 'OS', 'DBMS' ],
    type: 'Part Time',
    qualification: 'B.Tech'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bfc"),
    name: 'Vijay',
    age: 30,
    gender: 'M',
    exp: 7,
    subjects: [ 'C', 'OS', 'DBMS' ],
    type: 'Part Time',
    qualification: 'B.Tech'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bfd"),
    name: 'Priya',
    age: 25,
    gender: 'F',
    exp: 3,
    subjects: [ 'C', 'OS', 'DBMS' ],
    type: 'Part Time',
    qualification: 'B.Tech'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bfe"),
    name: 'Rohit',
    age: 33,
    gender: 'M',
    exp: 9,
    subjects: [ 'C', 'OS', 'DBMS' ],
    type: 'Part Time',
    qualification: 'B.Tech'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bff"),
    name: 'Sneha',
    age: 27,
    gender: 'F',
    exp: 4,
    subjects: [ 'C', 'OS', 'DBMS' ],
    type: 'Part Time',
    qualification: 'B.Tech'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6c00"),
    name: 'Amit',
    age: 31,
    gender: 'M',
    exp: 6,
    subjects: [ 'C', 'OS', 'DBMS' ],
    type: 'Part Time',
    qualification: 'B.Tech'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6c01"),
    name: 'Neha',
    age: 29,
    gender: 'F',
    exp: 5,
    subjects: [ 'C', 'OS', 'DBMS' ],
    type: 'Part Time',
    qualification: 'B.Tech'
  }
]
```

2. Get the count of all faculty members.

```
ems> db.faculty.count()
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
10
ems>
```

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

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

```
ems> db.faculty.find({qualification:'Ph.D'})
[
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bf9"),
    name: 'Manoj',
    age: 38,
    gender: 'M',
    exp: 12,
    subjects: [ 'JAVA', 'DBMS' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bfb"),
    name: 'Suresh',
    age: 40,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bfd"),
    name: 'Mani',
    age: 38,
    gender: 'F',
    exp: 10,
    subjects: [ 'JAVA', 'DBMS', 'OS' ],
    type: 'Part Time',
    qualification: 'Ph.D'
  },
  {


```

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

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

```
ems> db.faculty.find({exp:{$gt:8, $lt:12}})
[
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bf8"),
    name: 'Krish',
    age: 35,
    gender: 'M',
    exp: 10,
    subjects: [ 'DS', 'C', 'OS' ],
    type: 'Full Time',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bfb"),
    name: 'Suresh',
    age: 40,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bfd"),
    name: 'Mani',
    age: 38,
    gender: 'F',
    exp: 10,
    subjects: [ 'JAVA', 'DBMS', 'OS' ],
    type: 'Part Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bff"),
    name: 'Nagesh',
    age: 39,
    gender: 'M',
    exp: 11,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
  }
]
```

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

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

```
]
ems> db.faculty.find({subjects:{$in:["MATHS", "NETWORKING"]}})
[
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bfb"),
    name: 'Suresh',
    age: 40,
    gender: 'M',
    exp: 9,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bfe"),
    name: 'Sivani',
    age: 32,
    gender: 'F',
    exp: 8,
    subjects: [ 'C', 'CPP', 'MATHS' ],
    type: 'Part Time',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bff"),
    name: 'Nagesh',
    age: 39,
    gender: 'M',
    exp: 11,
    subjects: [ 'JAVA', 'DBMS', '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("63f4ee2e89c337b00bef6c01"),
    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:[{subject:"JAVA"}, {type:'Part Time'}]}))`

mongo: mongo> /usr/bin/mongo --host 127.0.0.1:27017 --directConnection=true&serverSelectionTimeoutMS=2000

```
ems> db.faculty.find({$or:[{subject:"JAVA"}, {type:'Part Time'}]}))
[
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bfa"),
    name: 'Anush',
    age: 32,
    gender: 'F',
    exp: 8,
    subjects: [ 'C', 'CPP' ],
    type: 'Part Time',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bfd"),
    name: 'Mani',
    age: 38,
    gender: 'F',
    exp: 10,
    subjects: [ 'JAVA', 'DBMS', 'OS' ],
    type: 'Part Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bfe"),
    name: 'Sivani',
    age: 32,
    gender: 'F',
    exp: 8,
    subjects: [ 'C', 'CPP', 'MATHS' ],
    type: 'Part Time',
    qualification: 'M.Tech'
  }
]
```

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"
})
```

9. Update the data of all faculty members by incrementing their age and exp by one year.

```
db.faculty.updateMany({}, { $inc: { exp: 1, age: 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.employees.updateOne({ name: 'Sivani' }, { $set: { qualification: 'Ph.D', type: 'Full Time' } })
```

Before Update

```
{
  _id: ObjectId("63f4ee2e89c337b00bef6bfe"),
  name: 'Sivani',
  age: 33,
  gender: 'F',
  exp: 9,
  subjects: [ 'C', 'CPP', 'MATHS' ],
  type: 'Part Time',
  qualification: 'M.Tech'
},
{
```

After Update

```
ems> db.faculty.findOne({name: 'Sivani'})
{
  _id: ObjectId("63f4ee2e89c337b00bef6bfe"),
  name: 'Sivani',
  age: 33,
  gender: 'F',
  exp: 9,
  subjects: [ 'C', 'CPP', 'MATHS' ],
  type: 'Part Time',
  qualification: 'M.Tech'
}
```

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.find({subjects:'MATHS'})
[
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bfe"),
    name: 'Sivani',
    age: 33,
    gender: 'F',
    exp: 9,
    subjects: [ 'C', 'CPP', 'MATHS', 'PSK' ],
    type: 'Part Time',
    qualification: 'M.Tech'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6c01"),
    name: 'Latha',
    age: 41,
    gender: 'F',
    exp: 14,
    subjects: [ 'MATHS', 'PSK' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4fe8e89c337b00bef6c02"),
    name: 'Suresh Babu',
    age: 56,
    gender: 'M',
    exp: 26,
    subjects: [ 'MATHS', 'DE', 'PSK' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  }
]
```

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

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

```
{ acknowledged: true, deletedCount: 1 }
```

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

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

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

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

```
]
ems> db.faculty.find().sort({age:-1}).limit(5)
[
  {
    _id: ObjectId("63f4ee2e89c337b00bef6c01"),
    name: 'Latha',
    age: 41,
    gender: 'F',
    exp: 14,
    subjects: [ 'MATHS', 'PSK' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bfb"),
    name: 'Suresh',
    age: 41,
    gender: 'M',
    exp: 10,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bff"),
    name: 'Nagesh',
    age: 40,
    gender: 'M',
    exp: 12,
    subjects: [ 'JAVA', 'DBMS', 'NETWORKING' ],
    type: 'Full Time',
    qualification: 'Ph.D'
  },
  {
    _id: ObjectId("63f4ee2e89c337b00bef6bfd"),
    name: 'Mani',
    age: 39,
    gender: 'F',
    exp: 11,
    subjects: [ 'JAVA', 'DBMS', 'OS' ],
```