```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
C:\Users\scs>mongosh
Current Mongosh Log ID: 63f6423147b6314107762fff
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.7.1
Using MongoDB:          6.0.4
Using Mongosh:          1.7.1

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


To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

------
   The server generated these startup warnings when booting
   2023-02-22T18:21:10.833+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
    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> show dbs
admin      40.00 KiB
config    108.00 KiB
local      40.00 KiB
test> db
test
test> use iti
switched to db iti
```

```
iti> db.createCollection('students');
{ ok: 1 }
iti> show collections
students
iti> db.students.insertOne({firstName:"asmaa",lastNam
.com"})
{
  acknowledged: true,
  insertedId: ObjectId("63f64c81fa6eb91b208be3a2")
}
```

```
iti> show dbs
admin    40.00 KiB
config   72.00 KiB
iti      40.00 KiB
local    72.00 KiB
```

```
iti> db.students.insertMany([{firstName:"mayar",lastName:"hamed",email:"mayar@gmail.com",age:25,numbers:
["0123456789","0147852369"],},{firstName:"ahmed",lastName:"gamal",email:"ahmed@gmail.com",age:35,numbers
:["0326598741","02589631447"],},{firstName:"nesma",lastName:"gamal",email:"nesma@gmail.com",age:38,numbe
rs:["0369741258","0147896523","0236458791"],},{firstName:"samia",lastName:"raslan",email:"samia@gmail.co
m",age:62,numbers:["0125689745"],},{firstName:"ahmed",lastName:"saeed",email:"ahmeds@gmail.com",age:25,n
numbers:["0235874196","0258749631"],}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("63f6aab8b7dac05cada9c366"),
    '1': ObjectId("63f6aab8b7dac05cada9c367"),
    '2': ObjectId("63f6aab8b7dac05cada9c368"),
    '3': ObjectId("63f6aab8b7dac05cada9c369"),
    '4': ObjectId("63f6aab8b7dac05cada9c36a")
  }
}
```

```
iti> db.students.find({firstName:"asmaa"})
[
  {
    _id: ObjectId("63f64c81fa6eb91b208be3a2"),
    firstName: 'asmaa',
    lastName: 'gamal',
    age: 24,
    email: 'asmaagamal99192@gmail.com'
  }
]
```

```
iti> db.students.find({age:35})
[
  {
    _id: ObjectId("63f6aab8b7dac05cada9c367"),
    firstName: 'ahmed',
    lastName: 'gamal',
    email: 'ahmed@gmail.com',
    age: 35,
    numbers: [ '0326598741', '02589631447' ]
  }
]
```

```
iti> db.students.find({age:{$gt:24}})
[
  {
    _id: ObjectId("63f6aab8b7dac05cada9c366"),
    firstName: 'mayar',
    lastName: 'hamed',
    email: 'mayar@gmail.com',
    age: 25,
    numbers: [ '0123456789', '0147852369' ]
  },
  {
    _id: ObjectId("63f6aab8b7dac05cada9c367"),
    firstName: 'ahmed',
    lastName: 'gamal',
    email: 'ahmed@gmail.com',
    age: 35,
    numbers: [ '0326598741', '02589631447' ]
  },
  {
    _id: ObjectId("63f6aab8b7dac05cada9c368"),
    firstName: 'nesma',
    lastName: 'gamal',
    email: 'nesma@gmail.com',
    age: 38,
    numbers: [ '0369741258', '0147896523', '0236458791' ]
  },
  {
    _id: ObjectId("63f6aab8b7dac05cada9c369"),
    firstName: 'samia',
    lastName: 'raslan',
    email: 'samia@gmail.com',
    age: 62,
    numbers: [ '0125689745' ]
  },
  {
    _id: ObjectId("63f6aab8b7dac05cada9c36a"),
    firstName: 'ahmed',
    lastName: 'saeed',
    email: 'ahmeds@gmail.com',
    age: 25,
    numbers: [ '0235874196', '0258749631' ]
  }
]
```

```
iti> db.students.find().pretty()
[
  {
    _id: ObjectId("63f64c81fa6eb91b208be3a2"),
    firstName: 'asmaa',
    lastName: 'gamal',
    age: 24,
    email: 'asmaagamal99192@gmail.com'
  },
  {
    _id: ObjectId("63f6aab8b7dac05cada9c366"),
    firstName: 'mayar',
    lastName: 'hamed',
    email: 'mayar@gmail.com',
    age: 25,
    numbers: [ '0123456789', '0147852369' ]
  },
  {
    _id: ObjectId("63f6aab8b7dac05cada9c367"),
    firstName: 'ahmed',
    lastName: 'gamal',
    email: 'ahmed@gmail.com',
    age: 35,
    numbers: [ '0326598741', '02589631447' ]
  },
  {
    _id: ObjectId("63f6aab8b7dac05cada9c368"),
    firstName: 'nesma',
    lastName: 'gamal',
    email: 'nesma@gmail.com',
    age: 38,
    numbers: [ '0369741258', '0147896523', '0236458791' ]
  },
  {
    _id: ObjectId("63f6aab8b7dac05cada9c369"),
    firstName: 'samia',
    lastName: 'raslan',
    email: 'samia@gmail.com',
    age: 62,
    numbers: [ '0125689745' ]
  },
  {
    _id: ObjectId("63f6aab8b7dac05cada9c36a"),
    firstName: 'ahmed',
    lastName: 'saeed',
    email: 'ahmeds@gmail.com',
    age: 25,
    numbers: [ '0235874196', '0258749631' ]
```

```
iti> db.students.find().count();
6
```