G.Divya Sri

2200030179

DAY-1

```
Microsoft Windows [Version 10.0.22000.2538]

(c) Microsoft Corporation. All rights reserved.

C:\Users\Bhuvaneswari>mongosh

Current Mongosh Log ID: 65cae5f6b7a89c6f40457bdd

Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.4

Using MongoDB:        7.0.5

Using Mongosh:        2.1.4

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

------

 The server generated these startup warnings when booting 2024-02-13T09:12:36.356+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

------

test> show dbs  //shows all the dbs

admin   40.00 KiB

config  72.00 KiB

local   72.00 KiB

test> use facultydb //switched to db facultydb

switched to db facultydb

facultydb> show dbs

admin   40.00 KiB

config  72.00 KiB

local   72.00 KiB

facultydb> db.createCollection("faculty")  //used to create a new Collection
```

```
{ ok: 1 }

facultydb> show dbs

admin      40.00 KiB

config     72.00 KiB

facultydb   8.00 KiB

local      72.00 KiB

facultydb> show collections  // to retrieve the collections

faculty

facultydb> db

facultydb

facultydb> db.faculty.drop() // to drop the collection

true

facultydb> show collections


facultydb> db.dropDatabase() // to drop the database

{ ok: 1, dropped: 'facultydb' }

facultydb> show dbs

admin   40.00 KiB

config  72.00 KiB

local   72.00 KiB

facultydb> use admin  //switched to the admin

switched to db admin

admin> db

admin

admin> use facultydb

switched to db facultydb

facultydb> db.createCollection("faculty")
```

```
{ ok: 1 }

facultydb> show collections

faculty

facultydb> //insert document

facultydb> db.faculty.insert({fid:4654,fname:"SURYA"})

DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.

{

 acknowledged: true,

  insertedIds: { '0': ObjectId('65cae705b7a89c6f40457bde') }

}

facultydb> db.faculty.find() // to display the documents

[

  {

    _id: ObjectId('65cae705b7a89c6f40457bde'),

    fid: 4654,

    fname: 'SURYA'

  }

]

facultydb> db.faculty.find().pretty()

[

  {

    _id: ObjectId('65cae705b7a89c6f40457bde'),

    fid: 4654,

    fname: 'SURYA'

  }

]

facultydb> db.faculty.find()
```

```
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  }
]
facultydb> db.faculty.insertOne({fid:4414,fname:"HARI"}) //inserts one document into the collection
{
  acknowledged: true,
  insertedId: ObjectId('65cae787b7a89c6f40457bdf')
}
facultydb> db.faculty.insertMany([{fid:5000},{fid:1234,fname:"KLU"}]) //inserts many documents into the collection
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('65cae7f8b7a89c6f40457be0'),
    '1': ObjectId('65cae7f8b7a89c6f40457be1')
  }
}
facultydb> db.faculty.find()
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
  _id: ObjectId('65cae787b7a89c6f40457bdf'),
   fid: 4414,
  fname: 'HARI'
  },
  { _id: ObjectId('65cae7f8b7a89c6f40457be0'), fid: 5000 },
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be1'),
    fid: 1234,
    fname: 'KLU'
  }
]
facultydb> db.faculty.find().size() //it is used to the no.of documents in the faculty collection
4
facultydb> db.faculty.countDocuments() //used to count the documents in the faculty collection
4
facultydb> db.faculty.find().limit(2)  // used to retrieve the first two documents in the faculty collection
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI'
  }
]
facultydb> db.faculty.find().limit(140) // if it exceeds the limit it display all the documents in the faculty collection
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI'
  },
  { _id: ObjectId('65cae7f8b7a89c6f40457be0'), fid: 5000 },
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be1'),
    fid: 1234,
    fname: 'KLU'
  }
]
facultydb> db.faculty.find().sort({fid:1}) //used to retrieve the documents in the ascending order based on the fid field in faculty collection
[
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be1'),
    fid: 1234,
    fname: 'KLU'
   },
   {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI'
   },
   {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
   },
   { _id: ObjectId('65cae7f8b7a89c6f40457be0'), fid: 5000 }
  ]
  facultydb> db.faculty.find().sort({fid:-1})  //used to retrieve the documents in the descending order based on the fid field in faculty collection

  [
   { _id: ObjectId('65cae7f8b7a89c6f40457be0'), fid: 5000 },
   {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
   },
   {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI'
  },
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be1'),
    fid: 1234,
    fname: 'KLU'
  }
]
facultydb> db.faculty.find().sort() // used to retrieve the documents in faculty collection without condition
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI'
  },
  { _id: ObjectId('65cae7f8b7a89c6f40457be0'), fid: 5000 },
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be1'),
    fid: 1234,
    fname: 'KLU'
  }
]
```

```
facultydb> db.faculty.find({fid:1234}) //used to retrieve the document based on the fid field in faculty collection

[

  {

    _id: ObjectId('65cae7f8b7a89c6f40457be1'),

    fid: 1234,

    fname: 'KLU'

  }

]

facultydb> db.faculty.updateOne({fid:1234},{$set:{fname:"KLEF"}}) //updating the name in single document based on the fid field in faculty collection

{

  acknowledged: true,

  insertedId: null,

  matchedCount: 1,

  modifiedCount: 1,

  upsertedCount: 0

}

facultydb> db.faculty.find().limit(1) //retrieving the first document in the faculty collection

[

  {

    _id: ObjectId('65cae705b7a89c6f40457bde'),

    fid: 4654,

    fname: 'SURYA'

  }

]

facultydb> db.faculty.find().limit(2) // retrieving the first two documents in the faculty collection

[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI'
  }
]
facultydb> db.faculty.find().limit(3) //retrieving the first three documents in the faculty collection
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI'
  },
  { _id: ObjectId('65cae7f8b7a89c6f40457be0'), fid: 5000 }
]
facultydb> db.faculty.find().limit(4)  // retrieving the first four documents in the faculty collection
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI'
  },
  { _id: ObjectId('65cae7f8b7a89c6f40457be0'), fid: 5000 },
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be1'),
    fid: 1234,
    fname: 'KLEF'
  }
]
facultydb> db.faculty.find({fid:5000}) //to display the document based on the fid field
[ { _id: ObjectId('65cae7f8b7a89c6f40457be0'), fid: 5000 } ]
facultydb> db.faculty.updateOne({fid:5000},{$set:{fname:"Harshitha",dept:"CSE"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
facultydb> db.faculty.find({fid:5000})
[
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be0'),
    fid: 5000,
    dept: 'CSE',
    fname: 'Harshitha'
  }
]
facultydb> db.fcaulty.countDocuments()
0
facultydb> db.faculty.countDocuments()
4
facultydb> db.faculty.find().size()
4
facultydb> db.faculty.insertOne({fid:1764,fname:"Sanjana",dept:"CSE"})
{
  acknowledged: true,
  insertedId: ObjectId('65caeb45b7a89c6f40457be2')
}
facultydb> db.faculty.findOne({fid:1234})
{ _id: ObjectId('65cae7f8b7a89c6f40457be1'), fid: 1234, fname: 'KLEF' }
facultydb> db.faculty.findOne({fid:1234,fname:"KLEF"})
{ _id: ObjectId('65cae7f8b7a89c6f40457be1'), fid: 1234, fname: 'KLEF' }
facultydb> db.faculty.findOne({fid:1234,fname:"VIT"})
null
facultydb> db.faculty.find({dept:"CSE"}) //used to display the documents based on dept field
```

```
[
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be0'),
    fid: 5000,
    dept: 'CSE',
    fname: 'Harshitha'
  },
  {
    _id: ObjectId('65caeb45b7a89c6f40457be2'),
    fid: 1764,
    fname: 'Sanjana',
    dept: 'CSE'
  }
]
facultydb> db.faculty.findOne({dept:"CSE"}) //used to find only one document based on the dept field in faculty collection
{
  _id: ObjectId('65cae7f8b7a89c6f40457be0'),
  fid: 5000,
  dept: 'CSE',
  fname: 'Harshitha'
}
facultydb> db.faculty.find({fid:1234})
[
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be1'),
    fid: 1234,
    fname: 'KLEF'
  }
]
facultydb> db.faculty.deleteOne({fid:1234})  //used to delete one document baed on the fid field in faculty collection
{ acknowledged: true, deletedCount: 1 }
facultydb> db.faculty.find().size()
4
facultydb> db.faculty.countDocuments()
4
facultydb> db.faculty.find({dept:"CSE"})
[
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be0'),
    fid: 5000,
    dept: 'CSE',
    fname: 'Harshitha'
  },
  {
    _id: ObjectId('65caeb45b7a89c6f40457be2'),
    fid: 1764,
    fname: 'Sanjana',
    dept: 'CSE'
  }
]
facultydb> db.faculty.deleteMany({dept:"CSE"}) //used to delete the documents based on the dept field in the faculty collection
{ acknowledged: true, deletedCount: 2 }
facultydb> db.faculty.find().size()
```

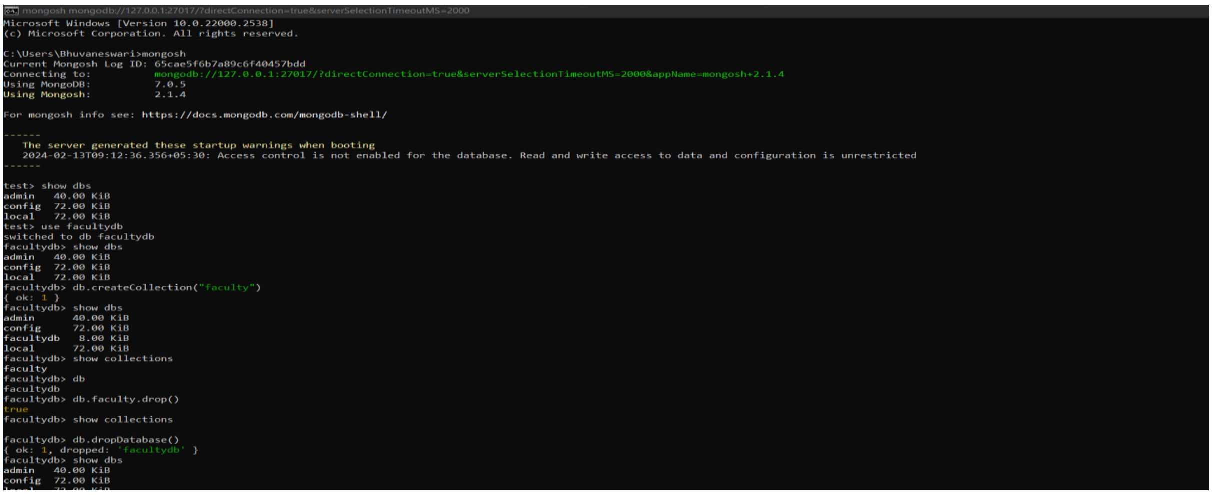
2

facultydb> db.faculty.countDocuments()

2

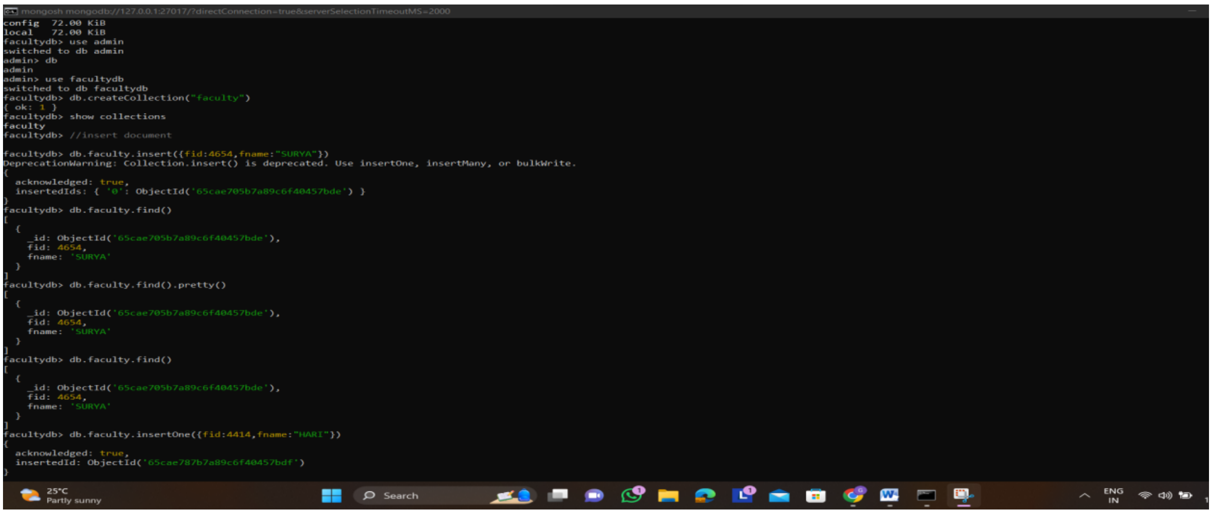
facultydb>

Screenshots of the above data:-



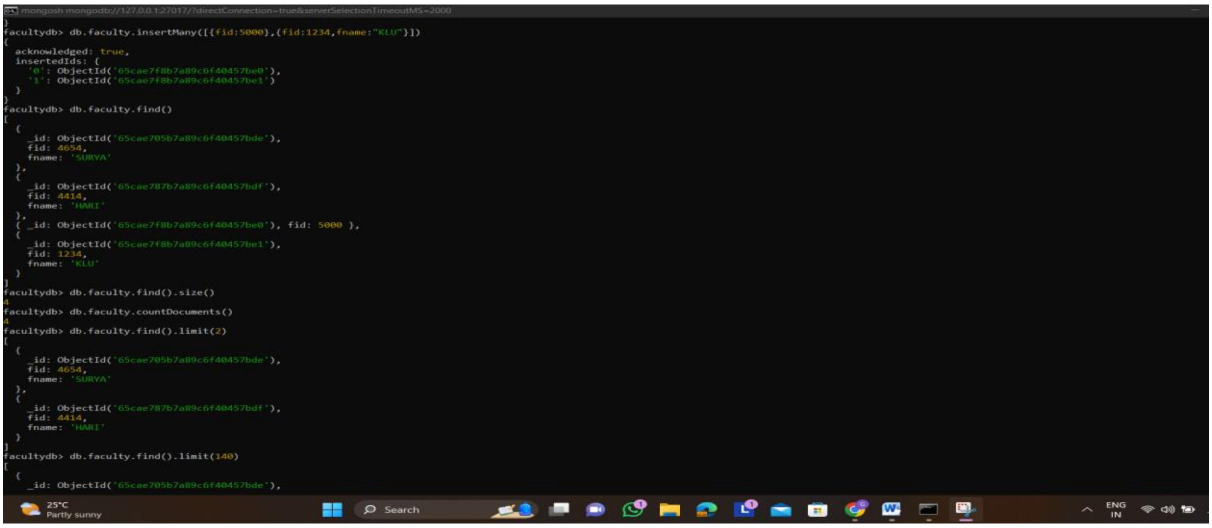
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Bhuvaneswari>mongosh
Current Mongosh Log ID: 65cae5f6b7a89c6f40457bdd
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.4
Using MongoDB:          7.0.5
Using Mongosh:          2.1.4

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

------
   The server generated these startup warnings when booting
   2024-02-13T09:12:36.356+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

test> show dbs
admin   40.00 KiB
config  72.00 KiB
local   72.00 KiB
test> use facultydb
switched to db facultydb
facultydb> show dbs
admin   40.00 KiB
config  72.00 KiB
local   72.00 KiB
facultydb> db.createCollection("faculty")
{ ok: 1 }
facultydb> show dbs
admin       40.00 KiB
config      72.00 KiB
facultydb    8.00 KiB
local       72.00 KiB
facultydb> show collections
faculty
facultydb> db
facultydb
facultydb> db.faculty.drop()
true
facultydb> show collections

facultydb> db.dropDatabase()
{ ok: 1, dropped: 'facultydb' }
facultydb> show dbs
admin   40.00 KiB
config  72.00 KiB
local   72.00 KiB
```

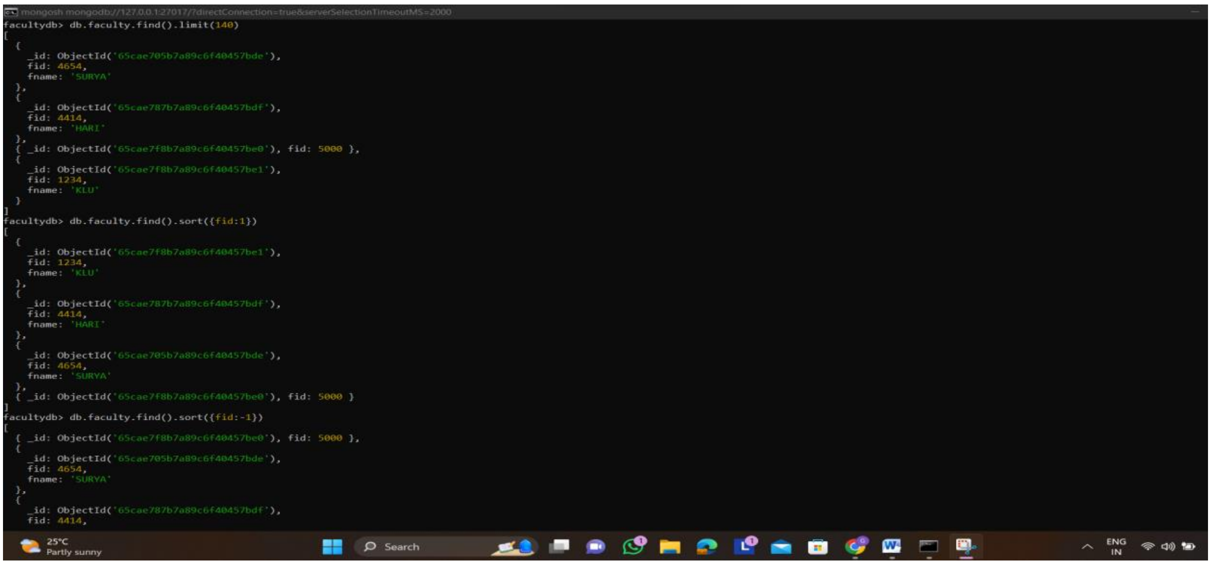


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
25°C
Partly sunny
Search
ENG
IN

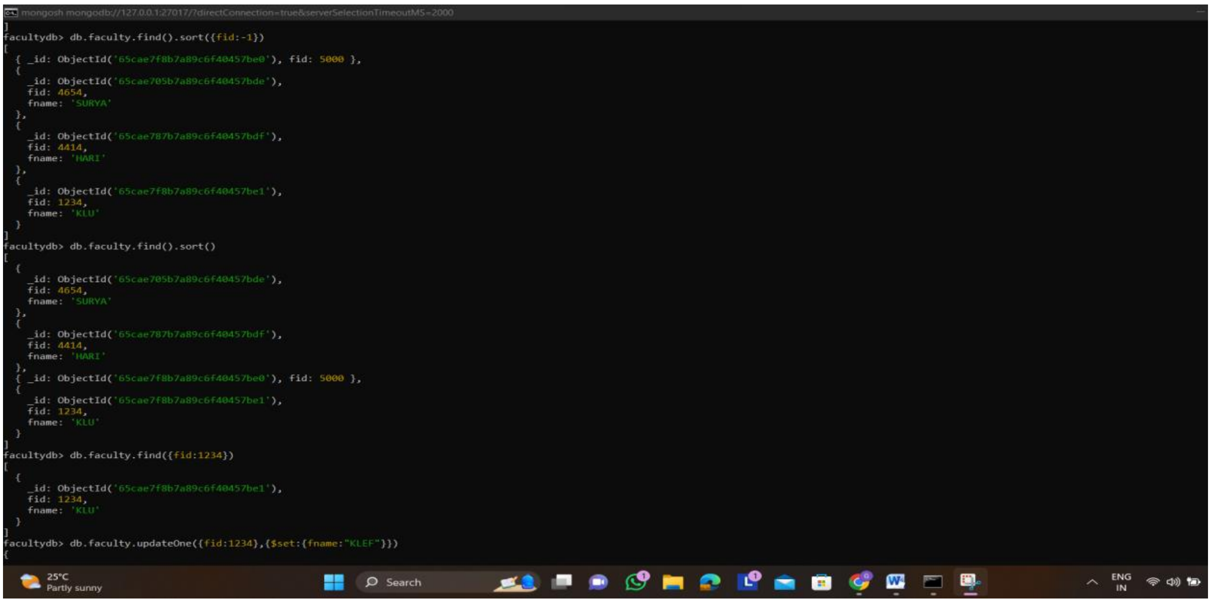
```
config  72.00 KiB
local   72.00 KiB
facultydb> use admin
switched to db admin
admin> db
admin
admin> use facultydb
switched to db facultydb
facultydb> db.createCollection("faculty")
{ ok: 1 }
facultydb> show collections
faculty
facultydb> //insert document

facultydb> db.faculty.insert({fid:4654,fname:"SURYA"})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('65cae705b7a89c6f40457bde') }
}
facultydb> db.faculty.find()
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  }
]
facultydb> db.faculty.find().pretty()
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  }
]
facultydb> db.faculty.find()
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  }
]
facultydb> db.faculty.insertOne({fid:4414,fname:"HARI"})
{
  acknowledged: true,
  insertedId: ObjectId('65cae787b7a89c6f40457bdf')
}
```


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
25°C
Partly sunny
Search
ENG
IN

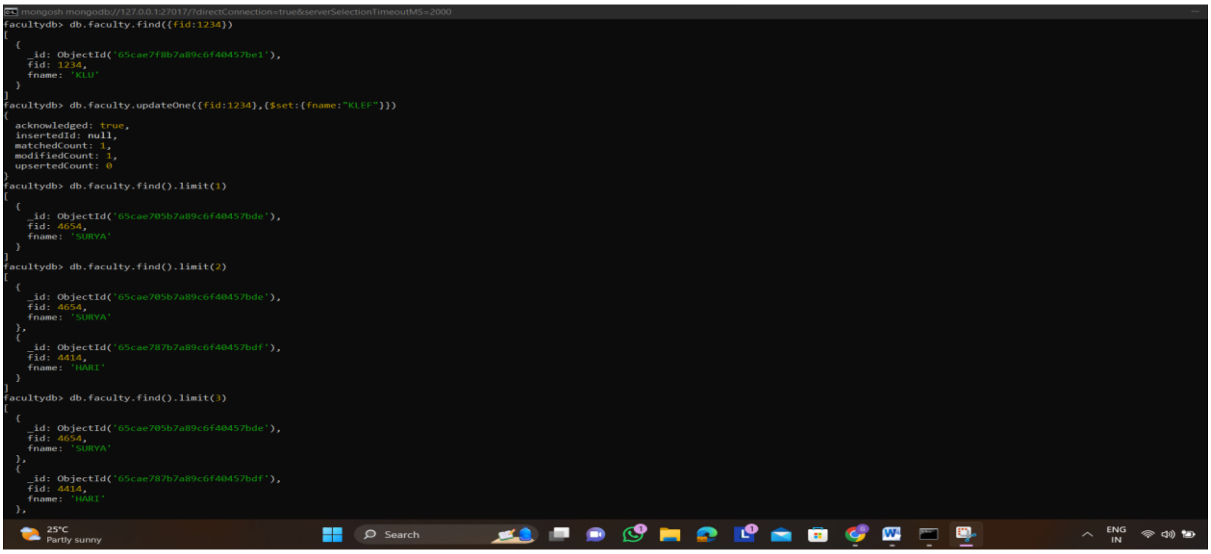
```
}
facultydb> db.faculty.insertMany([{fid:5000},{fid:1234,fname:"KLU"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('65cae7f8b7a89c6f40457be0'),
    '1': ObjectId('65cae7f8b7a89c6f40457be1')
  }
}
facultydb> db.faculty.find()
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI'
  },
  { _id: ObjectId('65cae7f8b7a89c6f40457be0'), fid: 5000 },
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be1'),
    fid: 1234,
    fname: 'KLU'
  }
]
facultydb> db.faculty.find().size()
4
facultydb> db.faculty.countDocuments()
4
facultydb> db.faculty.find().limit(2)
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI'
  }
]
facultydb> db.faculty.find().limit(140)
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
```


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
25°C
Partly sunny
Search
ENG
IN

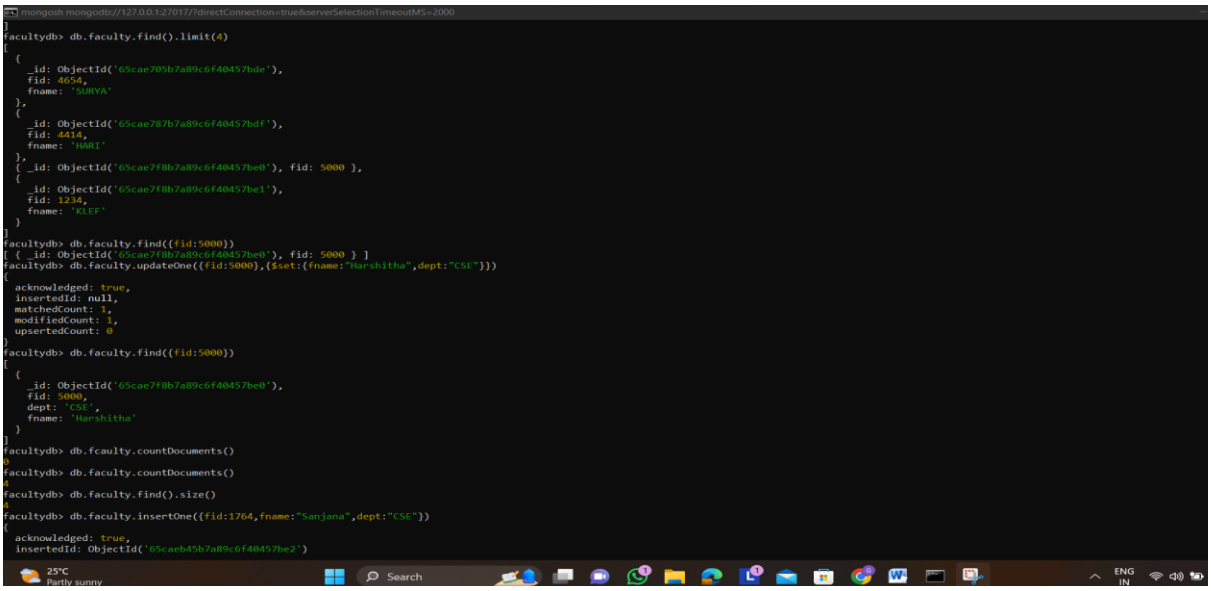
```
facultydb> db.faculty.find().limit(140)
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI'
  },
  { _id: ObjectId('65cae7f8b7a89c6f40457be0'), fid: 5000 },
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be1'),
    fid: 1234,
    fname: 'KLU'
  }
]
facultydb> db.faculty.find().sort({fid:1})
[
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be1'),
    fid: 1234,
    fname: 'KLU'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI'
  },
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  },
  { _id: ObjectId('65cae7f8b7a89c6f40457be0'), fid: 5000 }
]
facultydb> db.faculty.find().sort({fid:-1})
[
  { _id: ObjectId('65cae7f8b7a89c6f40457be0'), fid: 5000 },
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
```


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
25°C
Partly sunny
Search
ENG
IN

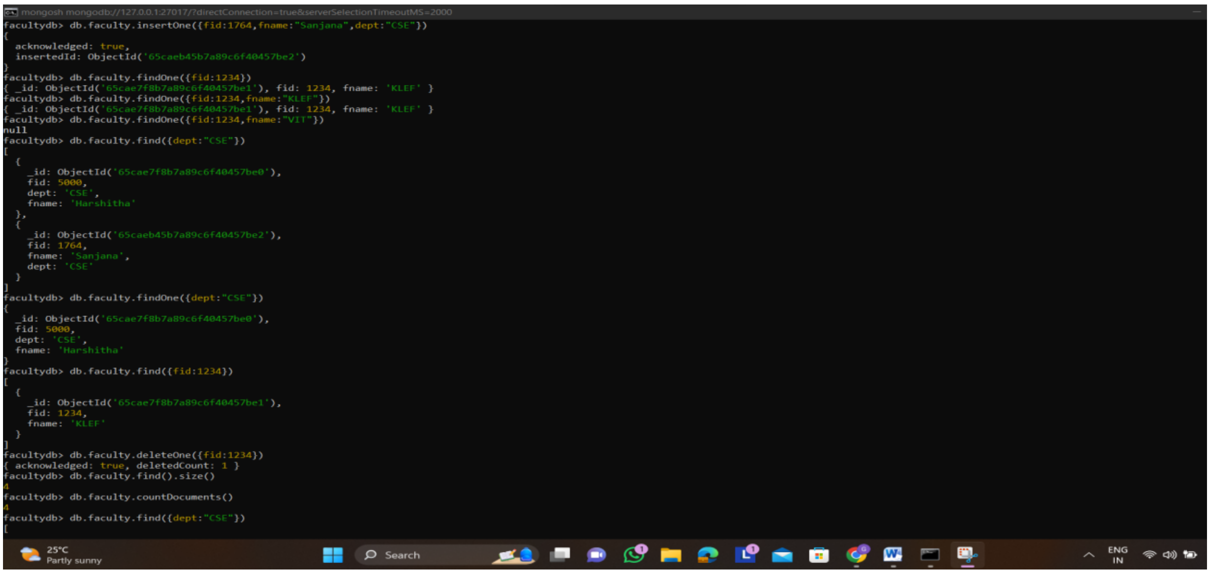
```
]
facultydb> db.faculty.find().sort({fid:-1})
[
  { _id: ObjectId('65cae7f8b7a89c6f40457be0'), fid: 5000 },
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI'
  },
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be1'),
    fid: 1234,
    fname: 'KLU'
  }
]
facultydb> db.faculty.find().sort()
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI'
  },
  { _id: ObjectId('65cae7f8b7a89c6f40457be0'), fid: 5000 },
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be1'),
    fid: 1234,
    fname: 'KLU'
  }
]
facultydb> db.faculty.find({fid:1234})
[
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be1'),
    fid: 1234,
    fname: 'KLU'
  }
]
facultydb> db.faculty.updateOne({fid:1234},{$set:{fname:"KLEF"}})
{
```


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
25°C
Partly sunny
Search
ENG
IN

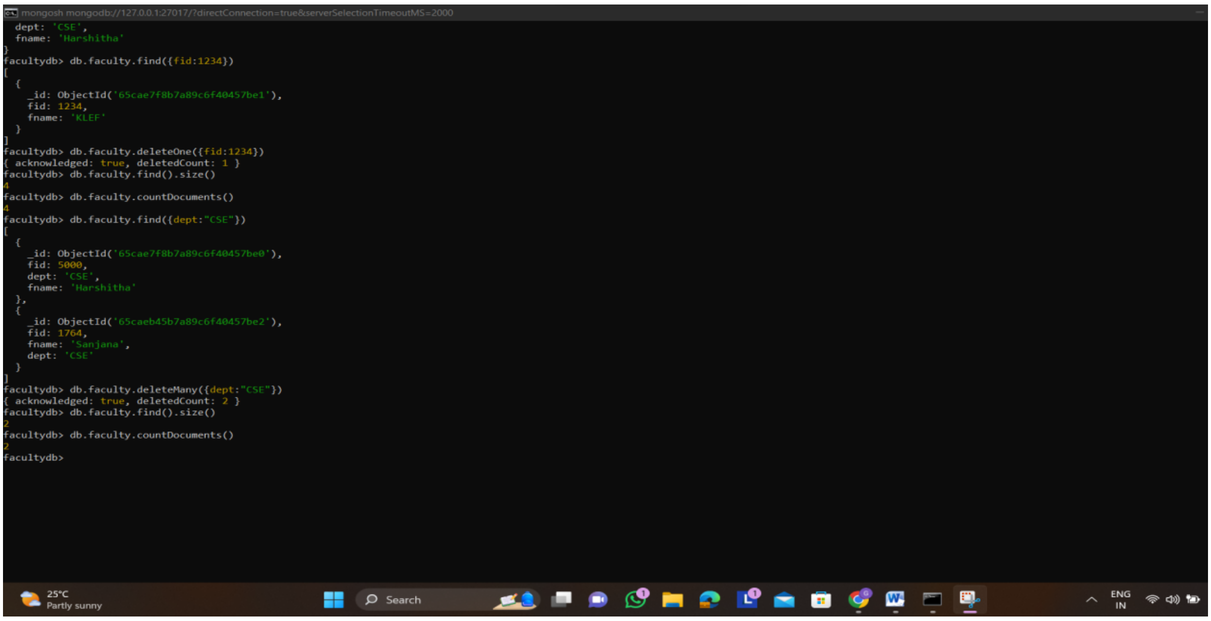
```
facultydb> db.faculty.find({fid:1234})
[
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be1'),
    fid: 1234,
    fname: 'KLU'
  }
]
facultydb> db.faculty.updateOne({fid:1234},{$set:{fname:"KLEF"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
facultydb> db.faculty.find().limit(1)
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  }
]
facultydb> db.faculty.find().limit(2)
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI'
  }
]
facultydb> db.faculty.find().limit(3)
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI'
  },
```


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
25°C
Partly sunny
Search
ENG
IN

```
]
facultydb> db.faculty.find().limit(4)
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI'
  },
  { _id: ObjectId('65cae7f8b7a89c6f40457be0'), fid: 5000 },
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be1'),
    fid: 1234,
    fname: 'KLEF'
  }
]
facultydb> db.faculty.find({fid:5000})
[ { _id: ObjectId('65cae7f8b7a89c6f40457be0'), fid: 5000 } ]
facultydb> db.faculty.updateOne({fid:5000},{$set:{fname:"Harshitha",dept:"CSE"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
facultydb> db.faculty.find({fid:5000})
[
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be0'),
    fid: 5000,
    dept: 'CSE',
    fname: 'Harshitha'
  }
]
facultydb> db.fcaulty.countDocuments()
0
facultydb> db.faculty.countDocuments()
4
facultydb> db.faculty.find().size()
4
facultydb> db.faculty.insertOne({fid:1764,fname:"Sanjana",dept:"CSE"})
{
  acknowledged: true,
  insertedId: ObjectId('65caeb45b7a89c6f40457be2')
```


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
25°C
Partly sunny
Search
ENG
IN

```
facultydb> db.faculty.insertOne({fid:1764,fname:"Sanjana",dept:"CSE"})
{
  acknowledged: true,
  insertedId: ObjectId('65caeb45b7a89c6f40457be2')
}
facultydb> db.faculty.findOne({fid:1234})
{ _id: ObjectId('65cae7f8b7a89c6f40457be1'), fid: 1234, fname: 'KLEF' }
facultydb> db.faculty.findOne({fid:1234,fname:"KLEF"})
{ _id: ObjectId('65cae7f8b7a89c6f40457be1'), fid: 1234, fname: 'KLEF' }
facultydb> db.faculty.findOne({fid:1234,fname:"VIT"})
null
facultydb> db.faculty.find({dept:"CSE"})
[
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be0'),
    fid: 5000,
    dept: 'CSE',
    fname: 'Harshitha'
  },
  {
    _id: ObjectId('65caeb45b7a89c6f40457be2'),
    fid: 1764,
    fname: 'Sanjana',
    dept: 'CSE'
  }
]
facultydb> db.faculty.findOne({dept:"CSE"})
{
  _id: ObjectId('65cae7f8b7a89c6f40457be0'),
  fid: 5000,
  dept: 'CSE',
  fname: 'Harshitha'
}
facultydb> db.faculty.find({fid:1234})
[
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be1'),
    fid: 1234,
    fname: 'KLEF'
  }
]
facultydb> db.faculty.deleteOne({fid:1234})
{ acknowledged: true, deletedCount: 1 }
facultydb> db.faculty.find().size()
4
facultydb> db.faculty.countDocuments()
4
facultydb> db.faculty.find({dept:"CSE"})
[
```



mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
25°C
Partly sunny
Search
ENG
IN

```
  dept: 'CSE',
  fname: 'Harshitha'
}
facultydb> db.faculty.find({fid:1234})
[
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be1'),
    fid: 1234,
    fname: 'KLEF'
  }
]
facultydb> db.faculty.deleteOne({fid:1234})
{ acknowledged: true, deletedCount: 1 }
facultydb> db.faculty.find().size()
4
facultydb> db.faculty.countDocuments()
4
facultydb> db.faculty.find({dept:"CSE"})
[
  {
    _id: ObjectId('65cae7f8b7a89c6f40457be0'),
    fid: 5000,
    dept: 'CSE',
    fname: 'Harshitha'
  },
  {
    _id: ObjectId('65caeb45b7a89c6f40457be2'),
    fid: 1764,
    fname: 'Sanjana',
    dept: 'CSE'
  }
]
facultydb> db.faculty.deleteMany({dept:"CSE"})
{ acknowledged: true, deletedCount: 2 }
facultydb> db.faculty.find().size()
2
facultydb> db.faculty.countDocuments()
2
facultydb>
```

DAY-2

```
Microsoft Windows [Version 10.0.22000.2538]

(c) Microsoft Corporation. All rights reserved.


C:\Users\Bhuvaneswari>mongosh

Current Mongosh Log ID: 65caf10f9f550e50ce889e6e

Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.4

Using MongoDB:          7.0.5

Using Mongosh:          2.1.4


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


------

   The server generated these startup warnings when booting

   2024-02-13T09:12:36.356+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

------
```

```
test> show dbs

admin      40.00 KiB

config    108.00 KiB

facultydb  72.00 KiB

local      72.00 KiB

test> use facultydb

switched to db facultydb

facultydb> db

facultydb

facultydb> show collections

faculty

facultydb> db.faculty.find()

[

  {

    _id: ObjectId('65cae705b7a89c6f40457bde'),

    fid: 4654,

    fname: 'SURYA'

  },

  {

    _id: ObjectId('65cae787b7a89c6f40457bdf'),

    fid: 4414,

    fname: 'HARI'

  }

]

facultydb> db.faculty.find({}) //to display the documents in faculty with no condition

[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI'
  }
]
facultydb> db.faculty.updateMany({},{$set:{gender:"MALE"}}) //setting(or) updating the gender as male to documents in faculty collection
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
facultydb> db.faculty.find({})
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA',
    gender: 'MALE'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI',
    gender: 'MALE'
  }
]
facultydb> db.faculty.find({fid:4414})
[
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI',
    gender: 'MALE'
  }
]
facultydb> db.faculty.updateOne({fid:4414},{$set:{fname:"SEETHA",gender:"FEMALE"}}) //updating the fname as SEETHA and gender as FEMALE in single document based on the fid field
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
facultydb> db.faculty.updateOne({fid:1234},{$set:{fid:1870,fname:"SRI"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
facultydb> db.faculty.updateOne({fid:1234},{$set:{fid:1870,fname:"SRI"}},{upsert:true})  //update a single document that matched the fid field .if no matches it will insert the fid as 1870 and fname as SRI
{
  acknowledged: true,
  insertedId: ObjectId('65caf3ad94fed2fa6843db53'),
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1
}
facultydb> db.faculty.find()
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA',
    gender: 'MALE'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'SEETHA',
    gender: 'FEMALE'
  },
  {
    _id: ObjectId('65caf3ad94fed2fa6843db53'),
    fid: 1870,
    fname: 'SRI'
  }
]
facultydb> db.faculty.countDocuments()
3
facultydb> db.faculty.countDocuments({gender:"FEMALE"}) //count the documents based on the condition that gender as FEMALE
1
facultydb> db.faculty.find({fid:1870})
[
  {
    _id: ObjectId('65caf3ad94fed2fa6843db53'),
    fid: 1870,
    fname: 'SRI'
  }
]
facultydb> db.faculty.updateMany({fid:1870},{$set:{gender:"FEMALE"}}) //updating the documents matches the fid field as 1870 gender as FEMALE in faculty collection
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
```

```
}
facultydb> db.faculty.findOne({fid:1870})
{
  _id: ObjectId('65caf3ad94fed2fa6843db53'),
  fid: 1870,
  fname: 'SRI',
  gender: 'FEMALE'
}
facultydb> db.faculty.countDocuments({gender:"FEMALE"})
2
facultydb> db.faculty.distinct("gender") //used to retrieve the distinct values of a specific field
[ 'FEMALE', 'MALE' ]
facultydb> db.createCollection("emp")  //creates a new collection as emp
{ ok: 1 }
facultydb> show collections
emp
faculty
facultydb> db.emp.insertMany([{_id:1,name:"A"},{_id:2,name:"B"}]) //inserting the documents in emp collection
{ acknowledged: true, insertedIds: { '0': 1, '1': 2 } }
facultydb> db.emp.find({})
[ { _id: 1, name: 'A' }, { _id: 2, name: 'B' } ]
facultydb> db.emp.insertMany([{_id:3,name:"C"},{_id:4,name:"D"}])
{ acknowledged: true, insertedIds: { '0': 3, '1': 4 } }
facultydb> db.emp.find()
[
  { _id: 1, name: 'A' },
  { _id: 2, name: 'B' },
  { _id: 3, name: 'C' },
  { _id: 4, name: 'D' }
]
facultydb> db.emp.find().size()
4
facultydb> db.emp.find().skip(1)  //used to retrieving the documents while skipping first document in emp collection
[ { _id: 2, name: 'B' }, { _id: 3, name: 'C' }, { _id: 4, name: 'D' } ]
facultydb> db.emp.find().skip(0)
[
  { _id: 1, name: 'A' },
  { _id: 2, name: 'B' },
  { _id: 3, name: 'C' },
  { _id: 4, name: 'D' }
]
facultydb> db.emp.find().skip(3) //used to retrieving the documents while skipping first three documents in emp collection

[ { _id: 4, name: 'D' } ]
facultydb> db.emp.find().sort()
[
  { _id: 1, name: 'A' },
  { _id: 2, name: 'B' },
  { _id: 3, name: 'C' },
  { _id: 4, name: 'D' }
]
facultydb> db.emp.find().sort({name:1}) //used to sort in ascending order based on the name field in emp collection
[
  { _id: 1, name: 'A' },

  { _id: 2, name: 'B' },

  { _id: 3, name: 'C' },

  { _id: 4, name: 'D' }

]

facultydb> db.emp.find().sort({name:-1})  //used to sort in descending order based on the name field in emp collection


[

  { _id: 4, name: 'D' },

  { _id: 3, name: 'C' },

  { _id: 2, name: 'B' },

  { _id: 1, name: 'A' }

]

facultydb> db.emp.find().sort({name:-1}).limit(1) //used to display the first document after sorting in descending based on name field in emp collection

[ { _id: 4, name: 'D' } ]

facultydb> db.emp.updateOne({_id:1},{$set:{salary:1000.5}}) //updating single document salary as 1000.5 which matches the _id:1 in emp collection

{

  acknowledged: true,

  insertedId: null,

  matchedCount: 1,

  modifiedCount: 1,

  upsertedCount: 0

}

facultydb> db.emp.updateOne({_id:2},{$set:{salary:2000.5}})

{

  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
facultydb> db.emp.updateOne({_id:3},{$set:{salary:3000.5}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
facultydb> db.emp.updateOne({_id:4},{$set:{salary:4000.5}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
facultydb> db.emp.find()
[
  { _id: 1, name: 'A', salary: 1000.5 },
  { _id: 2, name: 'B', salary: 2000.5 },
  { _id: 3, name: 'C', salary: 3000.5 },
  { _id: 4, name: 'D', salary: 4000.5 }
```

```
]
facultydb> db.emp.updateMany({},{$set:{gender:"FEMALE"}}) //updating the documents gender as FEMALE in emp collection
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 4,
  modifiedCount: 4,
  upsertedCount: 0
}
facultydb> db.emp.find()
[
  { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' },
  { _id: 2, name: 'B', salary: 2000.5, gender: 'FEMALE' },
  { _id: 3, name: 'C', salary: 3000.5, gender: 'FEMALE' },
  { _id: 4, name: 'D', salary: 4000.5, gender: 'FEMALE' }
]
facultydb> db.emp.updateOne({_id:2},{$set:{gender:"MALE"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
facultydb> db.emp.find()
[
  { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' },
  { _id: 2, name: 'B', salary: 2000.5, gender: 'MALE' },

  { _id: 3, name: 'C', salary: 3000.5, gender: 'FEMALE' },

  { _id: 4, name: 'D', salary: 4000.5, gender: 'FEMALE' }

]

facultydb> db.emp.updateOne({_id:4},{$set:{name:"C",gender:"MALE"}})

{

  acknowledged: true,

  insertedId: null,

  matchedCount: 1,

  modifiedCount: 1,

  upsertedCount: 0

}

facultydb> db.emp.find()

[

  { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' },

  { _id: 2, name: 'B', salary: 2000.5, gender: 'MALE' },

  { _id: 3, name: 'C', salary: 3000.5, gender: 'FEMALE' },

  { _id: 4, name: 'C', salary: 4000.5, gender: 'MALE' }

]

facultydb> db.emp.countDocuments({gender:"FEMALE"})

2

facultydb> db.emp.countDocuments({gender:"MALE"})

2

facultydb> db.emp.find()

[

  { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' },

  { _id: 2, name: 'B', salary: 2000.5, gender: 'MALE' },
  { _id: 3, name: 'C', salary: 3000.5, gender: 'FEMALE' },
  { _id: 4, name: 'C', salary: 4000.5, gender: 'MALE' }
]
facultydb> db.emp.find({},{name:1}) //it retrieves the documents with its name and _id .also know as projection
[
  { _id: 1, name: 'A' },
  { _id: 2, name: 'B' },
  { _id: 3, name: 'C' },
  { _id: 4, name: 'C' }
]
facultydb> db.emp.find({_id:4})
[ { _id: 4, name: 'C', salary: 4000.5, gender: 'MALE' } ]
facultydb> db.emp.find({_id:4},{salary:1}) //used to retrieve the document salary that matches _id:4
[ { _id: 4, salary: 4000.5 } ]
facultydb> db.emp.find({_id:4},{_id:0,salary:1}) //inclusion of salary and exclusion of _id of a document that matches the _id:4
[ { salary: 4000.5 } ]
facultydb> db.emp.find({_id:3},{_id:0,salary:1,name:1}) //inclusion of salary and name exclusion of a _id of a document that ,matches the _id:3
[ { name: 'C', salary: 3000.5 } ]
facultydb> db.emp.find({salary:{$gt:2000}}) //used to retrieve the documents that salary is greater than the  2000 in emp collection
[
  { _id: 2, name: 'B', salary: 2000.5, gender: 'MALE' },
  { _id: 3, name: 'C', salary: 3000.5, gender: 'FEMALE' },
  { _id: 4, name: 'C', salary: 4000.5, gender: 'MALE' }
]
```

```
facultydb> db.emp.find({salary:{$lt:3000}}) //used to retrieve the documents that salary is less than the 3000

[

  { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' },

  { _id: 2, name: 'B', salary: 2000.5, gender: 'MALE' }

]

facultydb> db.emp.find({salary:{$gte:3000}}) //used to retrieve the documents that salary is greater than or equal to 3000

[

  { _id: 3, name: 'C', salary: 3000.5, gender: 'FEMALE' },

  { _id: 4, name: 'C', salary: 4000.5, gender: 'MALE' }

]

facultydb> db.emp.find({salary:{$lte:3000}}) //used to retrieve the documents that salary is less than or equal to 3000 in emp collection

[

  { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' },

  { _id: 2, name: 'B', salary: 2000.5, gender: 'MALE' }

]

facultydb> db.emp.find({salary:{$ne:3000.5}}) //used to retrieve the documents that salary is not equal to 3000.5

[

  { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' },

  { _id: 2, name: 'B', salary: 2000.5, gender: 'MALE' },

  { _id: 4, name: 'C', salary: 4000.5, gender: 'MALE' }

]

facultydb> db.emp.find()

[

  { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' },

  { _id: 2, name: 'B', salary: 2000.5, gender: 'MALE' },
  { _id: 3, name: 'C', salary: 3000.5, gender: 'FEMALE' },

  { _id: 4, name: 'C', salary: 4000.5, gender: 'MALE' }

]

facultydb> db.emp.updateMany({salary:{$gt:3000}},{$set:{salary:5000}})  //used to update the salary as 5000 to those which are greater than the 3000 in emp collection

{

  acknowledged: true,

  insertedId: null,

  matchedCount: 2,

  modifiedCount: 2,

  upsertedCount: 0

}

facultydb> db.emp.find()

[

  { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' },

  { _id: 2, name: 'B', salary: 2000.5, gender: 'MALE' },

  { _id: 3, name: 'C', salary: 5000, gender: 'FEMALE' },

  { _id: 4, name: 'C', salary: 5000, gender: 'MALE' }

]

facultydb> db.emp.find({gender:{$in:["MALE","FEMALE"]}}) //used to retrieve the documents  that gender field matches either MALE or FEMALE

[

  { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' },

  { _id: 2, name: 'B', salary: 2000.5, gender: 'MALE' },

  { _id: 3, name: 'C', salary: 5000, gender: 'FEMALE' },

  { _id: 4, name: 'C', salary: 5000, gender: 'MALE' }

]

facultydb> db.emp.find({gender:{$in:["MALE","FEMALE","OTHERS"]}}) //used to retrieve the documents  that gender field matches either MALE or FEMALE or OTHERS
```

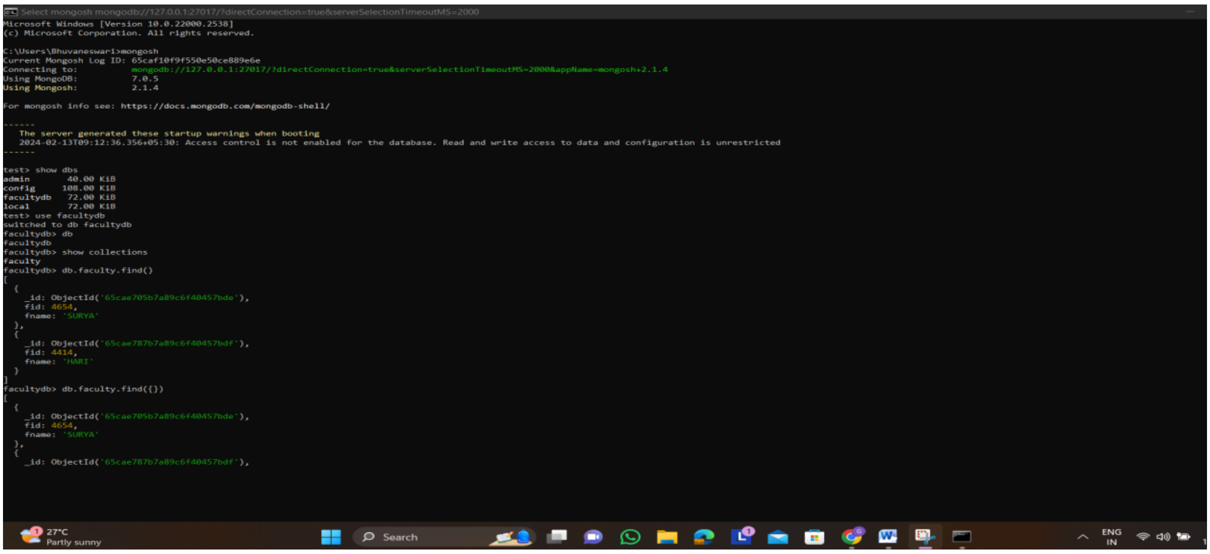
```
[
  { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' },
  { _id: 2, name: 'B', salary: 2000.5, gender: 'MALE' },
  { _id: 3, name: 'C', salary: 5000, gender: 'FEMALE' },
  { _id: 4, name: 'C', salary: 5000, gender: 'MALE' }
]
facultydb> db.emp.find({gender:{$nin:["MALE"]}}) //used to retrieve the documents that gender is not equal to MALE
[
  { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' },
  { _id: 3, name: 'C', salary: 5000, gender: 'FEMALE' }
]
facultydb> db.emp.renameCollection("employee") //renaming emp collection as employee
{ ok: 1 }
facultydb> show collections
employee
faculty
facultydb> db.employee.renameCollection("emp")
{ ok: 1 }
facultydb> db.emp.updateMany({},{$rename:{salary:"sal"}}) //updating the documents that renaming the salary as sal
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 4,
  modifiedCount: 4,
  upsertedCount: 0
}
facultydb> db.emp.find()
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },
  { _id: 4, name: 'C', gender: 'MALE', sal: 5000 }
]
facultydb> db.emp.updateMany({_id:4},{$set:{sal:7000}}) //updating the documents sal as 7000 that matches the field _id:4
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
facultydb> db.emp.find()
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 }
]
facultydb> db.emp.find({$and:[{gender:"FEMALE"},{sal:{$gt:2000}}]}) //retrieving the documents with gender as FEMALE and sal is greater than the 2000
[ { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 } ]
facultydb> db.emp.find({$or:[{gender:"FEMALE"},{sal:{$gt:2000}}]}) //retrieving the documents with gender as FEMALE or sal is greater than the 2000 in emp collection
```

```
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 }
]
```

facultydb> db.emp.find({$or:[{gender:"FEMALE"},{sal:{$lt:2000}}]}) //used to retrieve the documents with gender as FEMALE or sal is less than the 2000 in emp collection

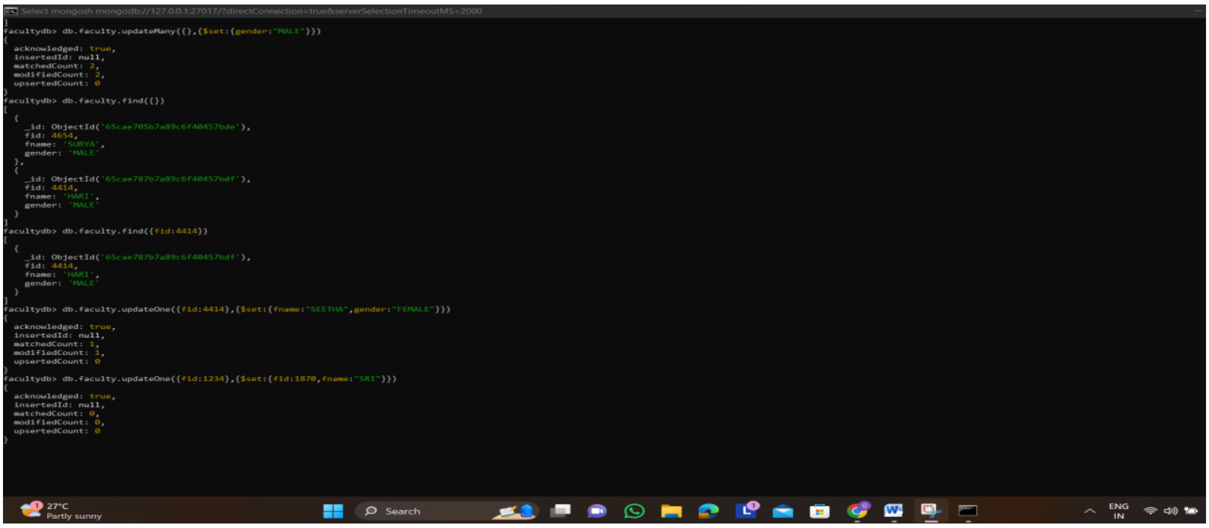
```
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },
  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 }
]
facultydb>
```

Screenshots of the above data:-

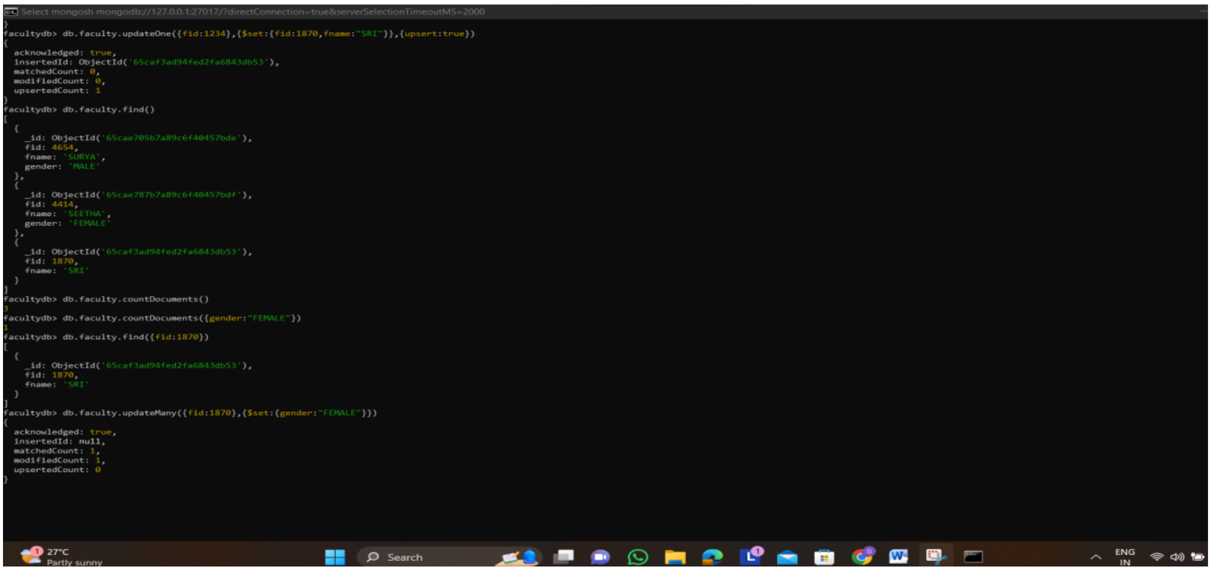


Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

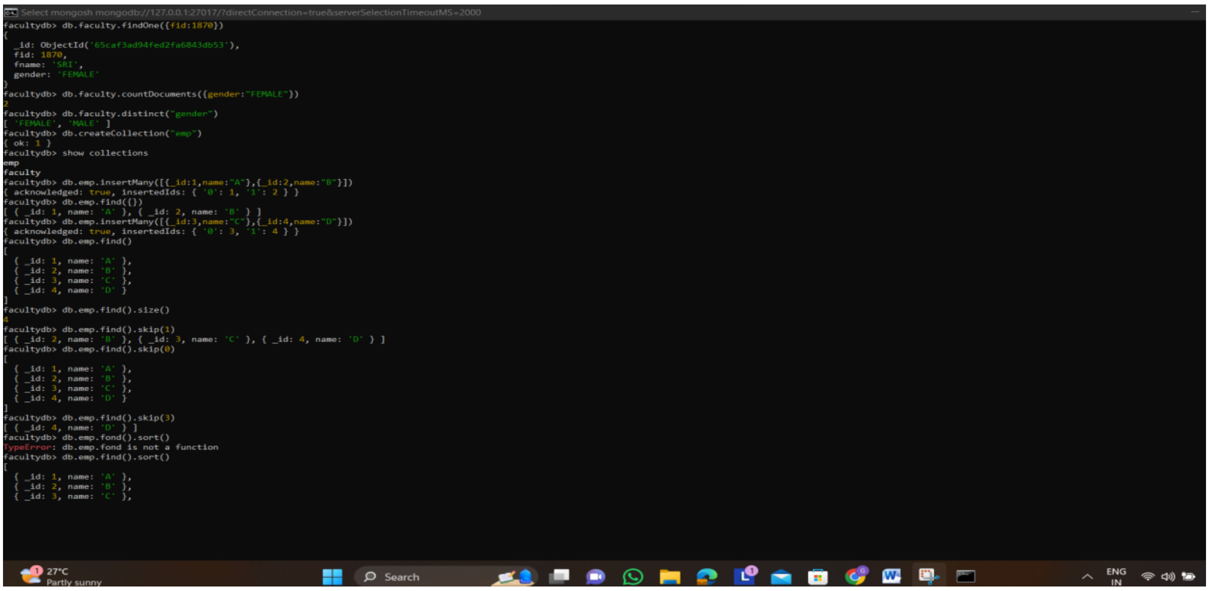
```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Bhuvaneswari>mongosh
Current Mongosh Log ID: 65caf10f9f550e50ce889e6e
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.4
Using MongoDB:          7.0.5
Using Mongosh:          2.1.4

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

------
   The server generated these startup warnings when booting
   2024-02-13T09:12:36.356+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

test> show dbs
admin        40.00 KiB
config      108.00 KiB
facultydb    72.00 KiB
local        72.00 KiB
test> use facultydb
switched to db facultydb
facultydb> db
facultydb
facultydb> show collections
faculty
facultydb> db.faculty.find()
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI'
  }
]
facultydb> db.faculty.find({})
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
```


Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
27°C
Partly sunny
Search
ENG
IN

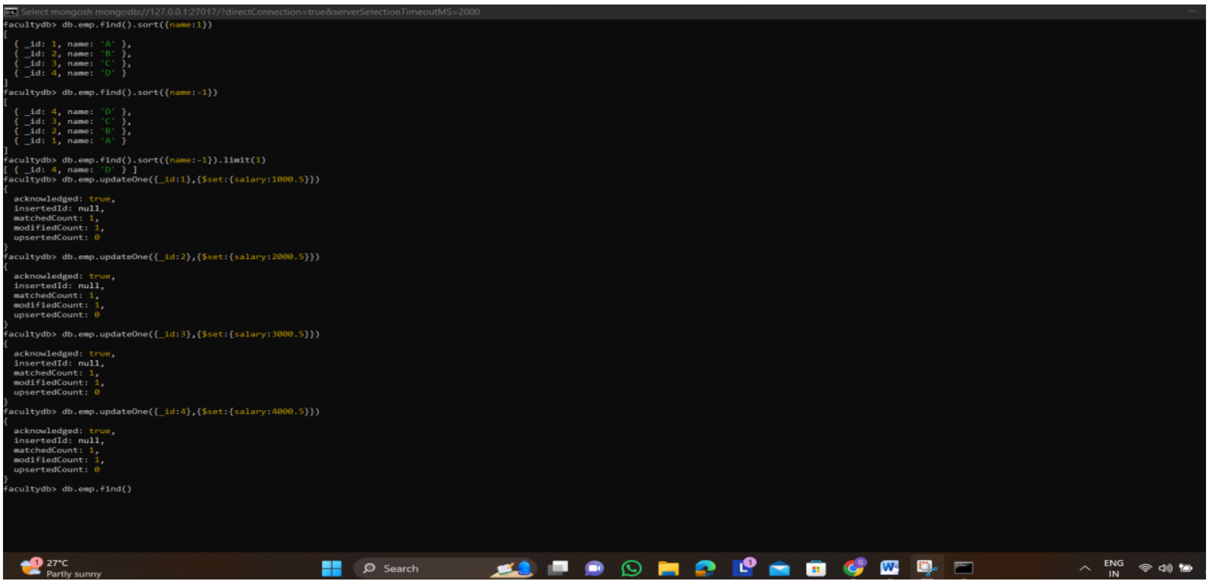
```
]
facultydb> db.faculty.updateMany({},{$set:{gender:"MALE"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
facultydb> db.faculty.find({})
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA',
    gender: 'MALE'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI',
    gender: 'MALE'
  }
]
facultydb> db.faculty.find({fid:4414})
[
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'HARI',
    gender: 'MALE'
  }
]
facultydb> db.faculty.updateOne({fid:4414},{$set:{fname:"SEETHA",gender:"FEMALE"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
facultydb> db.faculty.updateOne({fid:1234},{$set:{fid:1870,fname:"SRI"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```


Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
27°C
Partly sunny
Search
ENG
IN

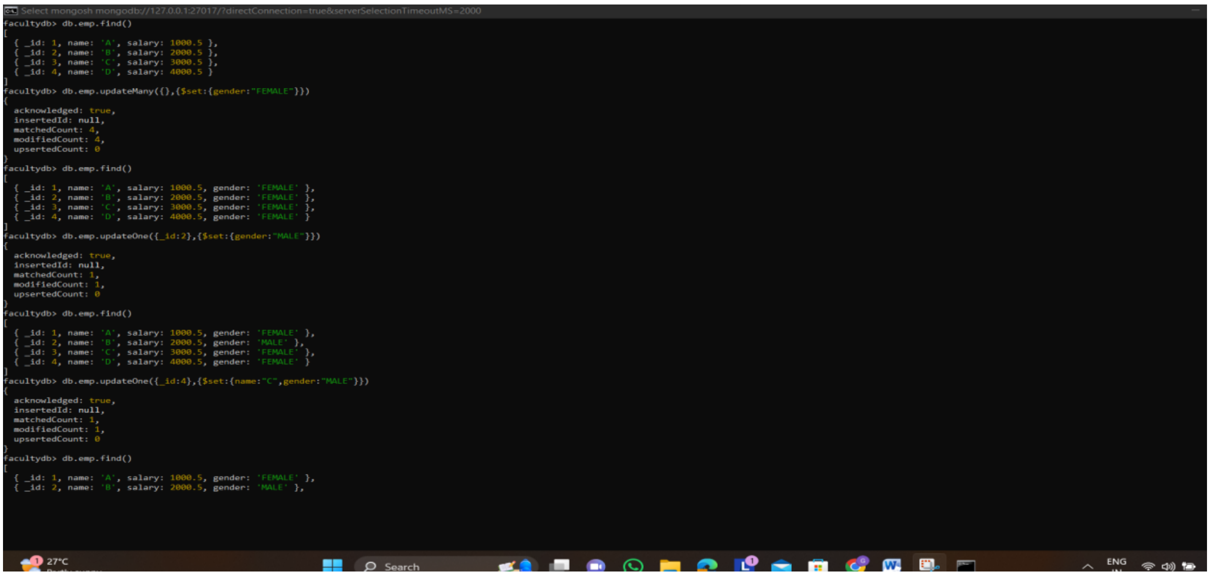
```
}
facultydb> db.faculty.updateOne({fid:1234},{$set:{fid:1870,fname:"SRI"}},{upsert:true})
{
  acknowledged: true,
  insertedId: ObjectId('65caf3ad94fed2fa6843db53'),
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1
}
facultydb> db.faculty.find()
[
  {
    _id: ObjectId('65cae705b7a89c6f40457bde'),
    fid: 4654,
    fname: 'SURYA',
    gender: 'MALE'
  },
  {
    _id: ObjectId('65cae787b7a89c6f40457bdf'),
    fid: 4414,
    fname: 'SEETHA',
    gender: 'FEMALE'
  },
  {
    _id: ObjectId('65caf3ad94fed2fa6843db53'),
    fid: 1870,
    fname: 'SRI'
  }
]
facultydb> db.faculty.countDocuments()
3
facultydb> db.faculty.countDocuments({gender:"FEMALE"})
1
facultydb> db.faculty.find({fid:1870})
[
  {
    _id: ObjectId('65caf3ad94fed2fa6843db53'),
    fid: 1870,
    fname: 'SRI'
  }
]
facultydb> db.faculty.updateMany({fid:1870},{$set:{gender:"FEMALE"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```


Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
27°C
Partly sunny
Search
ENG
IN

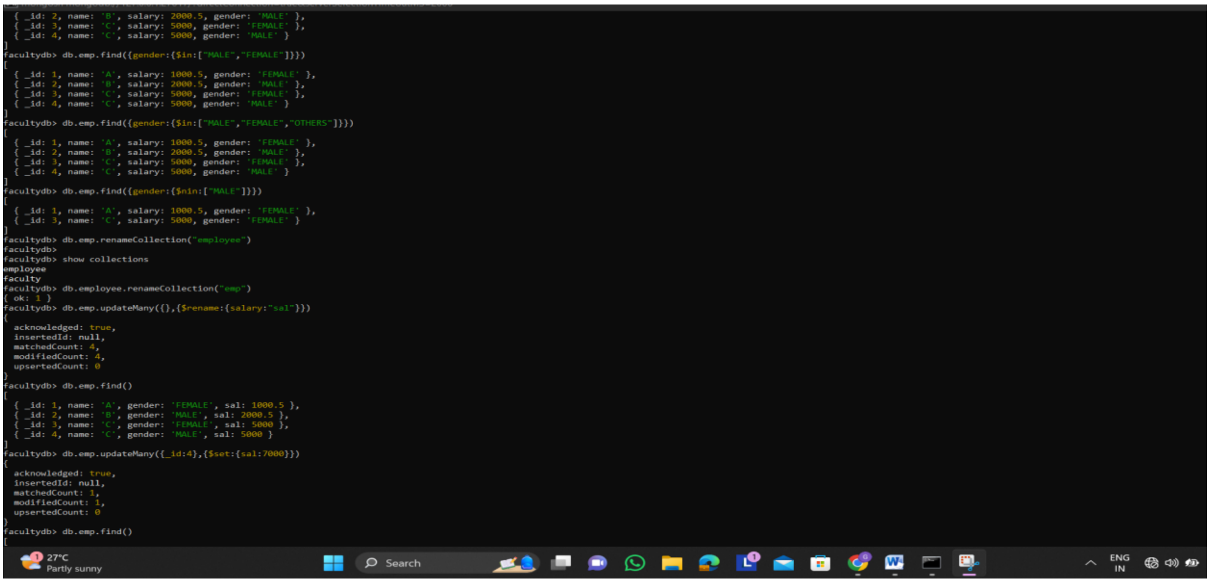
```
facultydb> db.faculty.findOne({fid:1870})
{
  _id: ObjectId('65caf3ad94fed2fa6843db53'),
  fid: 1870,
  fname: 'SRI',
  gender: 'FEMALE'
}
facultydb> db.faculty.countDocuments({gender:"FEMALE"})
2
facultydb> db.faculty.distinct("gender")
[ 'FEMALE', 'MALE' ]
facultydb> db.createCollection("emp")
{ ok: 1 }
facultydb> show collections
emp
faculty
facultydb> db.emp.insertMany([{_id:1,name:"A"},{_id:2,name:"B"}])
{ acknowledged: true, insertedIds: { '0': 1, '1': 2 } }
facultydb> db.emp.find({})
[ { _id: 1, name: 'A' }, { _id: 2, name: 'B' } ]
facultydb> db.emp.insertMany([{_id:3,name:"C"},{_id:4,name:"D"}])
{ acknowledged: true, insertedIds: { '0': 3, '1': 4 } }
facultydb> db.emp.find()
[
  { _id: 1, name: 'A' },
  { _id: 2, name: 'B' },
  { _id: 3, name: 'C' },
  { _id: 4, name: 'D' }
]
facultydb> db.emp.find().size()
4
facultydb> db.emp.find().skip(1)
[ { _id: 2, name: 'B' }, { _id: 3, name: 'C' }, { _id: 4, name: 'D' } ]
facultydb> db.emp.find().skip(0)
[
  { _id: 1, name: 'A' },
  { _id: 2, name: 'B' },
  { _id: 3, name: 'C' },
  { _id: 4, name: 'D' }
]
facultydb> db.emp.find().skip(3)
[ { _id: 4, name: 'D' } ]
facultydb> db.emp.fond().sort()
TypeError: db.emp.fond is not a function
facultydb> db.emp.find().sort()
[
  { _id: 1, name: 'A' },
  { _id: 2, name: 'B' },
  { _id: 3, name: 'C' },
```


Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
27°C
Partly sunny
Search
ENG
IN

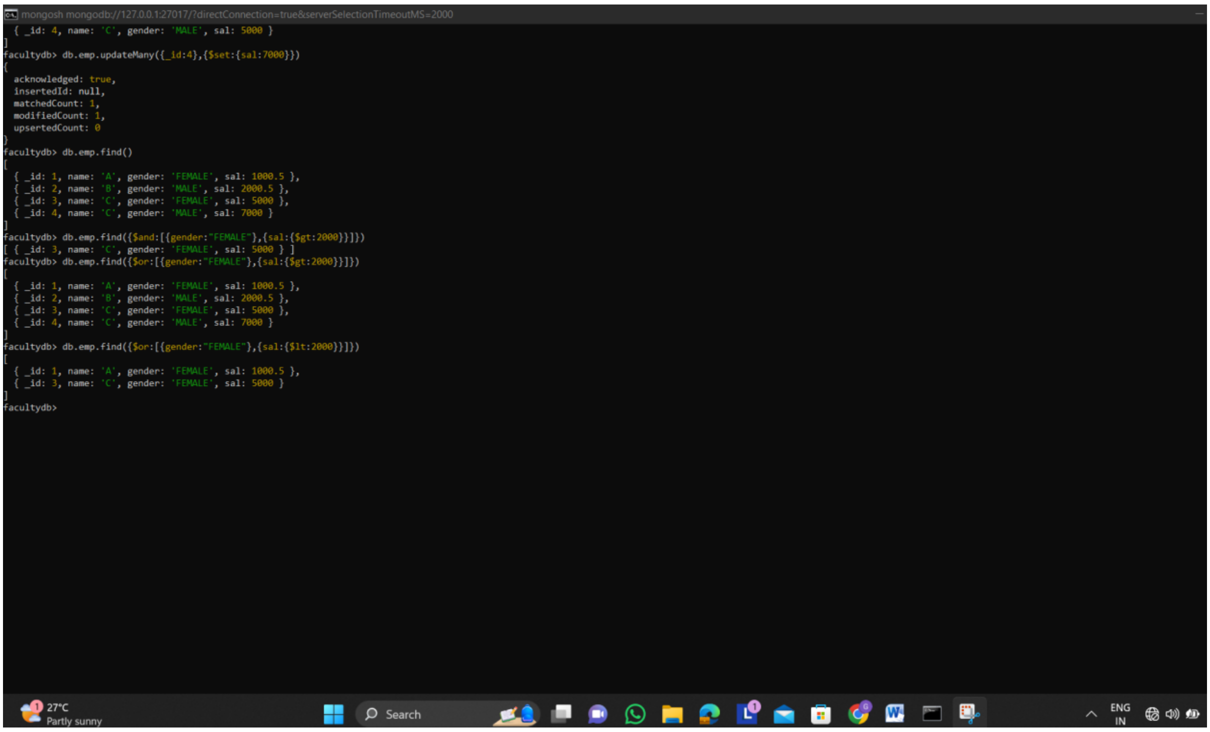
```
facultydb> db.emp.find().sort({name:1})
[
  { _id: 1, name: 'A' },
  { _id: 2, name: 'B' },
  { _id: 3, name: 'C' },
  { _id: 4, name: 'D' }
]
facultydb> db.emp.find().sort({name:-1})
[
  { _id: 4, name: 'D' },
  { _id: 3, name: 'C' },
  { _id: 2, name: 'B' },
  { _id: 1, name: 'A' }
]
facultydb> db.emp.find().sort({name:-1}).limit(1)
[ { _id: 4, name: 'D' } ]
facultydb> db.emp.updateOne({_id:1},{$set:{salary:1000.5}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
facultydb> db.emp.updateOne({_id:2},{$set:{salary:2000.5}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
facultydb> db.emp.updateOne({_id:3},{$set:{salary:3000.5}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
facultydb> db.emp.updateOne({_id:4},{$set:{salary:4000.5}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
facultydb> db.emp.find()
```


Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
27°C
Search
ENG

```
facultydb> db.emp.find()
[
  { _id: 1, name: 'A', salary: 1000.5 },
  { _id: 2, name: 'B', salary: 2000.5 },
  { _id: 3, name: 'C', salary: 3000.5 },
  { _id: 4, name: 'D', salary: 4000.5 }
]
facultydb> db.emp.updateMany({},{$set:{gender:"FEMALE"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 4,
  modifiedCount: 4,
  upsertedCount: 0
}
facultydb> db.emp.find()
[
  { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' },
  { _id: 2, name: 'B', salary: 2000.5, gender: 'FEMALE' },
  { _id: 3, name: 'C', salary: 3000.5, gender: 'FEMALE' },
  { _id: 4, name: 'D', salary: 4000.5, gender: 'FEMALE' }
]
facultydb> db.emp.updateOne({_id:2},{$set:{gender:"MALE"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
facultydb> db.emp.find()
[
  { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' },
  { _id: 2, name: 'B', salary: 2000.5, gender: 'MALE' },
  { _id: 3, name: 'C', salary: 3000.5, gender: 'FEMALE' },
  { _id: 4, name: 'D', salary: 4000.5, gender: 'FEMALE' }
]
facultydb> db.emp.updateOne({_id:4},{$set:{name:"C",gender:"MALE"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
facultydb> db.emp.find()
[
  { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' },
  { _id: 2, name: 'B', salary: 2000.5, gender: 'MALE' },
```



```
  { _id: 2, name: 'B', salary: 2000.5, gender: 'MALE' },
  { _id: 3, name: 'C', salary: 5000, gender: 'FEMALE' },
  { _id: 4, name: 'C', salary: 5000, gender: 'MALE' }
]
facultydb> db.emp.find({gender:{$in:["MALE","FEMALE"]}})
[
  { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' },
  { _id: 2, name: 'B', salary: 2000.5, gender: 'MALE' },
  { _id: 3, name: 'C', salary: 5000, gender: 'FEMALE' },
  { _id: 4, name: 'C', salary: 5000, gender: 'MALE' }
]
facultydb> db.emp.find({gender:{$in:["MALE","FEMALE","OTHERS"]}})
[
  { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' },
  { _id: 2, name: 'B', salary: 2000.5, gender: 'MALE' },
  { _id: 3, name: 'C', salary: 5000, gender: 'FEMALE' },
  { _id: 4, name: 'C', salary: 5000, gender: 'MALE' }
]
facultydb> db.emp.find({gender:{$nin:["MALE"]}})
[
  { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' },
  { _id: 3, name: 'C', salary: 5000, gender: 'FEMALE' }
]
facultydb> db.emp.renameCollection("employee")
facultydb>
facultydb> show collections
employee
faculty
facultydb> db.employee.renameCollection("emp")
{ ok: 1 }
facultydb> db.emp.updateMany({},{$rename:{salary:"sal"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 4,
  modifiedCount: 4,
  upsertedCount: 0
}
facultydb> db.emp.find()
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },
  { _id: 4, name: 'C', gender: 'MALE', sal: 5000 }
]
facultydb> db.emp.updateMany({_id:4},{$set:{sal:7000}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
facultydb> db.emp.find()
[
```



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

```
  { _id: 4, name: 'C', gender: 'MALE', sal: 5000 }
]
facultydb> db.emp.updateMany({_id:4},{$set:{sal:7000}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
facultydb> db.emp.find()
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 }
]
facultydb> db.emp.find({$and:[{gender:"FEMALE"},{sal:{$gt:2000}}]})
[ { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 } ]
facultydb> db.emp.find({$or:[{gender:"FEMALE"},{sal:{$gt:2000}}]})
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 }
]
facultydb> db.emp.find({$or:[{gender:"FEMALE"},{sal:{$lt:2000}}]})
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },
  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 }
]
facultydb>
```

DAY-3

Microsoft Windows [Version 10.0.22000.2538]

(c) Microsoft Corporation. All rights reserved.

C:\Users\Bhuvaneswari>mongosh

Current Mongosh Log ID: 65cb4073f8c747d6891952ea

Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.4

Using MongoDB: 7.0.5

Using Mongosh: 2.1.4

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

The server generated these startup warnings when booting 2024-02-13T15:41:24.889+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

------

```
test> show dbs

admin       40.00 KiB

config      84.00 KiB

facultydb  144.00 KiB

local       72.00 KiB

test> use facultydb

switched to db facultydb

facultydb> db.emp.find({})

[

  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },

  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },

  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },

  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 }

]

facultydb> db.emp.find({}).sort({salary:-1})

[

  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },

  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },

  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },

  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 }

]

facultydb> db.emp.find({}).sort({salary:-1}).limit(1)

[ { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 } ]

facultydb> db.emp.find({}).sort({sal:-1})

[

  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 },

  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },

  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 }

]

facultydb> db.emp.find({}).sort({sal:-1}).limit(1)

[ { _id: 4, name: 'C', gender: 'MALE', sal: 7000 } ]

facultydb> db.emp.find({},{name:1}).sort({sal:-1}).limit(1)

[ { _id: 4, name: 'C' } ]

facultydb> db.emp.find({},{name:1,_id:0}).sort({sal:-1}).limit(1)

[ { name: 'C' } ]

facultydb> //aggregate functions or multi-document functions

facultydb>

facultydb> db.emp.aggregate({$count:"totalcount"}) //that counts total number of documents in emp collection and returns the count field as totalcount

[ { totalcount: 4 } ]

facultydb> db.emp.aggregate({$group:{_id:null,totalsal:{$sum:"$sal"}}}) //that goups all the documents and calculate the total sal of all documents in emp collection

[ { _id: null, totalsal: 15001 } ]

facultydb> db.emp.aggregate({$group:{_id:null,avgsal:{$avg:"$sal"}}}) //that groups all the documents together and calculate the average to the salary of all documents in emp collection

[ { _id: null, avgsal: 3750.25 } ]

facultydb> db.emp.aggregate({$group:{_id:null,minsal:{$min:"$sal"}}}) //that groups all the docuemnts together and display the min value among all the sal

[ { _id: null, minsal: 1000.5 } ]

facultydb> db.emp.aggregate({$group:{_id:null,maxsal:{$max:"$sal"}}}) //that groups all the documents together and display the max value among all the sal

[ { _id: null, maxsal: 7000 } ]

facultydb> db.emp.aggregate({$group:{_id:null,totalsal:{$sum:"$sal"}}})

[ { _id: null, totalsal: 15001 } ]

facultydb> db.emp.aggregate({$group:{_id:"$gender",totalsal:{$sum:"$sal"}}}) //that groups the documents based on the gender field and then calculate the totalsal for each category
```

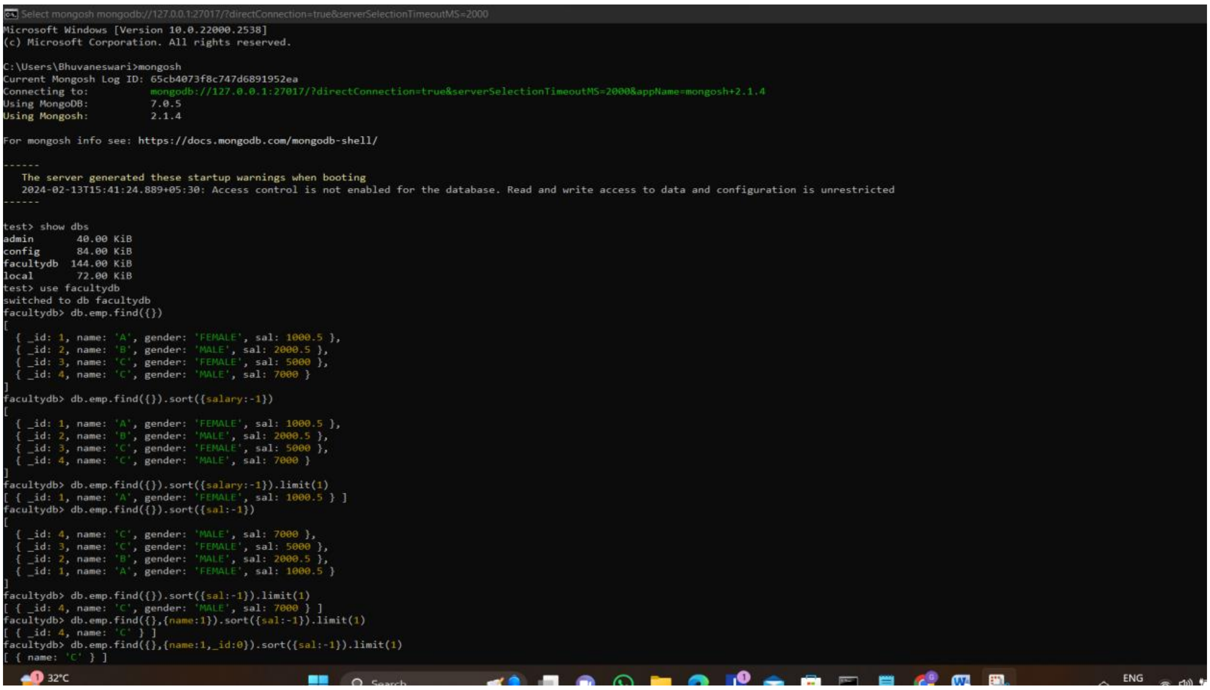
```
[
  { _id: 'FEMALE', totalsal: 6000.5 },
  { _id: 'MALE', totalsal: 9000.5 }
]
facultydb> db.emp.aggregate({$group:{_id:"$gender",avgsal:{$avg:"$sal"}}}) //that groups the documents based on the gender field and then performs avg to the sal for each category
[
  { _id: 'FEMALE', avgsal: 3000.25 },
  { _id: 'MALE', avgsal: 4500.25 }
]
facultydb> db.emp.aggregate({$group:{_id:"$gender",minsal:{$min:"$sal"}}}) //that groups the document based on the gender field and then display the minimum value among all the sal for each category
[ { _id: 'FEMALE', minsal: 1000.5 }, { _id: 'MALE', minsal: 2000.5 } ]
facultydb> db.emp.aggregate({$group:{_id:"$gender",maxsal:{$max:"$sal"}}}) //that groups tha documents based on the genderfield and then display the maximum value among all the sal for each category
[ { _id: 'MALE', maxsal: 7000 }, { _id: 'FEMALE', maxsal: 5000 } ]
facultydb> db.emp.aggregate({$match:{gender:"MALE"}}) //that displays the documents that matches the gender as MALE
[
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 }
]
facultydb> db.emp.aggregate([ {$group:{_id:"$gender",maxsal:{$max:"$sal"}}}])
[ { _id: 'FEMALE', maxsal: 5000 }, { _id: 'MALE', maxsal: 7000 } ]
facultydb>db.emp.aggregate([{$match:{gender:"MALE"}},{$group:{_id:"$gender",maxsal:{$max:"$sal"}}}]) //that filters the documents based on the gender as MALE and then displays the max sal among them
[ { _id: 'MALE', maxsal: 7000 } ]
facultydb> db.emp.aggregate({$match:{}}) //that displays all the documents with no condition
```

```
[

  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },

  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },

  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },

  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 }

]

facultydb> db.emp.aggregate([{$match:{gender:"MALE"}},{$group:{_id:"$gender",maxsal:{$max:"$sal"}}}])

[ { _id: 'MALE', maxsal: 7000 } ]

facultydb> db.emp.aggregate([{$match:{gender:"MALE"}},{$group:{_id:"$gender",maxsal:{$max:"$sal"}}},{$project:{maxsal:1,_id:0}}]) //that filters the documents based on the gender as MALE and then displays the max sal among them while excluding the _id



[ { maxsal: 7000 } ]

facultydb> db.emp.aggregate([{$match:{}},{$group:{_id:"$gender",maxsal:{$max:"$sal"}}},{$project:{maxsal:1,_id:0}}])  //that filters the documents based on the gender and then displays the max sal among them while excluding the _id

[ { maxsal: 7000 }, { maxsal: 5000 } ]

facultydb> db.emp.aggregate([{$match:{}},{$group:{_id:"$gender",maxsal:{$max:"$sal"}}},{$project:{maxsal:1,_id:0}},{$sort:{sal:1}}])

[ { maxsal: 5000 }, { maxsal: 7000 } ]

facultydb> db.emp.aggregate([{$match:{}},{$group:{_id:"$gender",maxsal:{$max:"$sal"}}},{$project:{_id:0}}])

[ { maxsal: 5000 }, { maxsal: 7000 } ]

facultydb> db.emp.aggregate([{$match:{}},{$group:{_id:"$gender",maxsal:{$max:"$sal"}}},{$project:{_id:0}},{$sort:{maxsal:1}}])  //that filters the documents based on the gender and then displays the max sal among them while excluding the _id and then finally sort in ascending order



[ { maxsal: 5000 }, { maxsal: 7000 } ]
```

```
facultydb> db.emp.aggregate([{$match:{}},{$group:{_id:"$gender",maxsal:{$max:"$sal"}}},{$project:{_id:0}},{$sort:{maxsal:-1}}])  //that filters the documents based on the gender and then displays the max sal among them while excluding the _id and then finally sort in descending order


[ { maxsal: 7000 }, { maxsal: 5000 } ]

facultydb> db.emp.aggregate([{$match:{}},{$group:{_id:"$gender",maxsal:{$max:"$sal"}}},{$project:{_id:0}},{$sort:{maxsal:-1}},{$limit:1}]) //that filters the documents based on the gender and then displays the max sal among them while excluding the _id and then finally sort in descending order and then displays the first document maxsal


[ { maxsal: 7000 } ]

facultydb> db.emp.find({})

[

  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },

  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },

  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },

  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 }

]

facultydb> db.emp.find({sal:{$not:{$gt:2000}}}) //retrieves the document that is not greater than the sal 2000

[ { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 } ]

facultydb> db.emp.find({})

[

  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },

  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },

  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },

  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 }

]
```

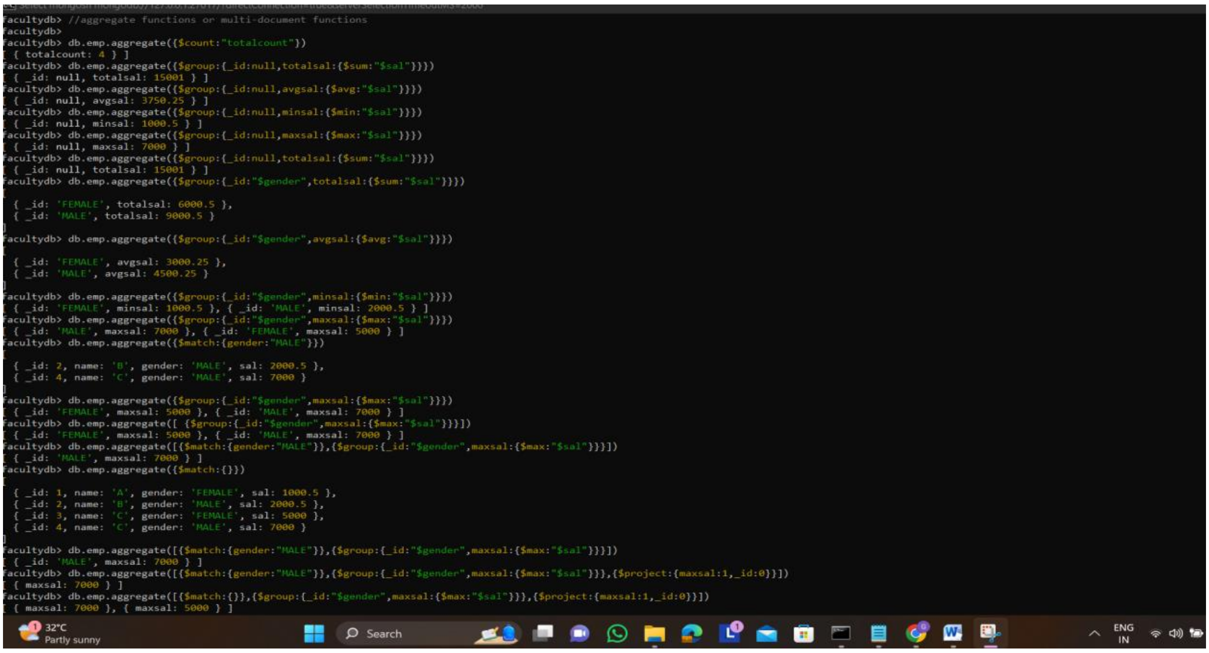
```
facultydb> db.emp.updateMany({},{$set:{location:"VIZAG"}}) //updating the documents with location as VIZAG in emp collection

{

  acknowledged: true,

  insertedId: null,

  matchedCount: 4,

  modifiedCount: 4,

  upsertedCount: 0

}

facultydb> db.emp.find({})

[

  {

    _id: 1,

    name: 'A',

    gender: 'FEMALE',

    sal: 1000.5,

    location: 'VIZAG'

  },

  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5, location: 'VIZAG' },

  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000, location: 'VIZAG' },

  { _id: 4, name: 'C', gender: 'MALE', sal: 7000, location: 'VIZAG' }

]

facultydb> db.emp.updateMany({},{$unset:{location:""}}) //updates the documents while removing the collection

{

  acknowledged: true,

  insertedId: null,

  matchedCount: 4,
  modifiedCount: 4,
  upsertedCount: 0
}
facultydb> db.emp.find({})
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 }
]
facultydb> db.emp.aggregate([ {$match:{}} , {$out:"person"}  ]) //that copies all the documents to the person collection from emp collection
facultydb> show collections
emp
faculty
person
facultydb> db.person.find({}) //that displays all the documents in the person collection
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 }
]
facultydb>
```

Screenshots of the above data:-

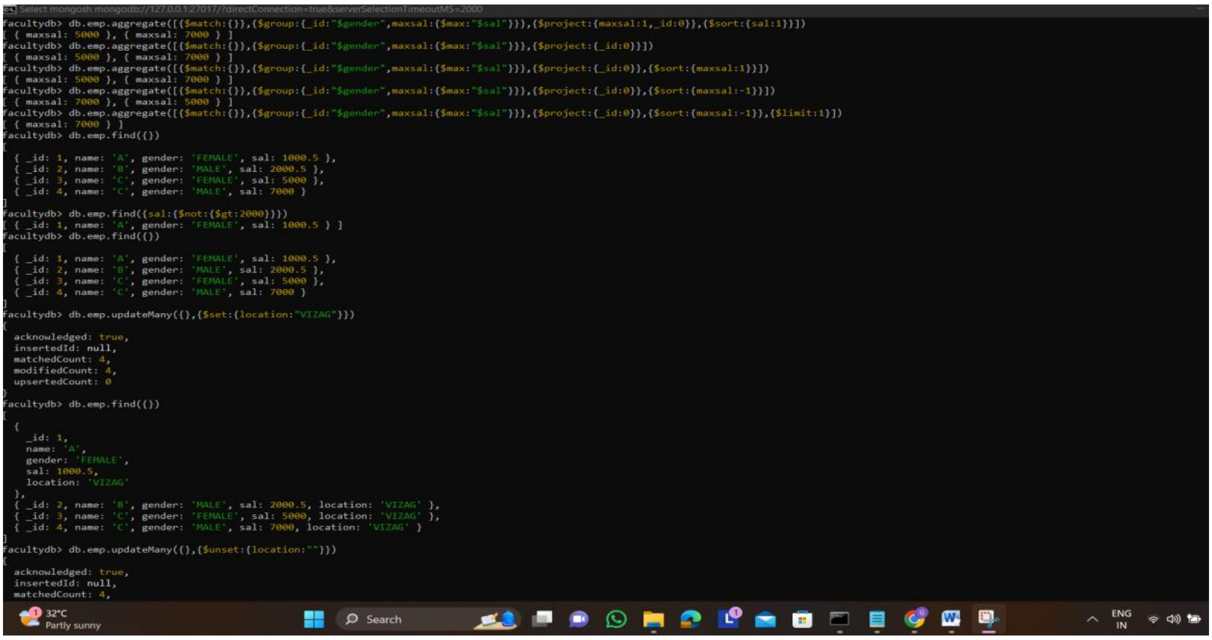


Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
32°C
Search
ENG

```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Bhuvaneswari>mongosh
Current Mongosh Log ID: 65cb4073f8c747d6891952ea
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.4
Using MongoDB:          7.0.5
Using Mongosh:          2.1.4

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

------
   The server generated these startup warnings when booting
   2024-02-13T15:41:24.889+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

test> show dbs
admin       40.00 KiB
config      84.00 KiB
facultydb  144.00 KiB
local       72.00 KiB
test> use facultydb
switched to db facultydb
facultydb> db.emp.find({})
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 }
]
facultydb> db.emp.find({}).sort({salary:-1})
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 }
]
facultydb> db.emp.find({}).sort({salary:-1}).limit(1)
[ { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 } ]
facultydb> db.emp.find({}).sort({sal:-1})
[
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 },
  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 }
]
facultydb> db.emp.find({}).sort({sal:-1}).limit(1)
[ { _id: 4, name: 'C', gender: 'MALE', sal: 7000 } ]
facultydb> db.emp.find({},{name:1}).sort({sal:-1}).limit(1)
[ { _id: 4, name: 'C' } ]
facultydb> db.emp.find({},{name:1,_id:0}).sort({sal:-1}).limit(1)
[ { name: 'C' } ]
```

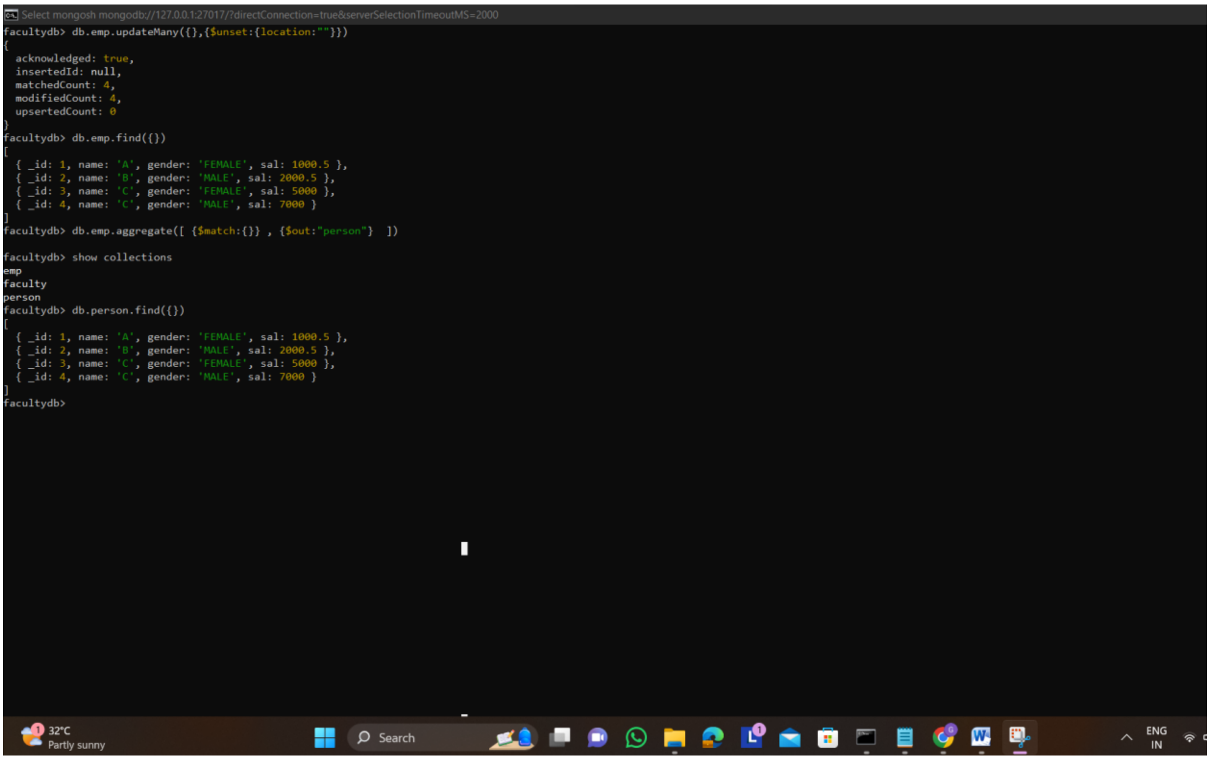


```
facultydb> //aggregate functions or multi-document functions
facultydb>
facultydb> db.emp.aggregate({$count:"totalcount"})
[ { totalcount: 4 } ]
facultydb> db.emp.aggregate({$group:{_id:null,totalsal:{$sum:"$sal"}}})
[ { _id: null, totalsal: 15001 } ]
facultydb> db.emp.aggregate({$group:{_id:null,avgsal:{$avg:"$sal"}}})
[ { _id: null, avgsal: 3750.25 } ]
facultydb> db.emp.aggregate({$group:{_id:null,minsal:{$min:"$sal"}}})
[ { _id: null, minsal: 1000.5 } ]
facultydb> db.emp.aggregate({$group:{_id:null,maxsal:{$max:"$sal"}}})
[ { _id: null, maxsal: 7000 } ]
facultydb> db.emp.aggregate({$group:{_id:null,totalsal:{$sum:"$sal"}}})
[ { _id: null, totalsal: 15001 } ]
facultydb> db.emp.aggregate({$group:{_id:"$gender",totalsal:{$sum:"$sal"}}})
[
  { _id: 'FEMALE', totalsal: 6000.5 },
  { _id: 'MALE', totalsal: 9000.5 }
]
facultydb> db.emp.aggregate({$group:{_id:"$gender",avgsal:{$avg:"$sal"}}})
[
  { _id: 'FEMALE', avgsal: 3000.25 },
  { _id: 'MALE', avgsal: 4500.25 }
]
facultydb> db.emp.aggregate({$group:{_id:"$gender",minsal:{$min:"$sal"}}})
[ { _id: 'FEMALE', minsal: 1000.5 }, { _id: 'MALE', minsal: 2000.5 } ]
facultydb> db.emp.aggregate({$group:{_id:"$gender",maxsal:{$max:"$sal"}}})
[ { _id: 'MALE', maxsal: 7000 }, { _id: 'FEMALE', maxsal: 5000 } ]
facultydb> db.emp.aggregate({$match:{gender:"MALE"}})
[
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 }
]
facultydb> db.emp.aggregate({$group:{_id:"$gender",maxsal:{$max:"$sal"}}})
[ { _id: 'FEMALE', maxsal: 5000 }, { _id: 'MALE', maxsal: 7000 } ]
facultydb> db.emp.aggregate([ {$group:{_id:"$gender",maxsal:{$max:"$sal"}}}])
[ { _id: 'FEMALE', maxsal: 5000 }, { _id: 'MALE', maxsal: 7000 } ]
facultydb> db.emp.aggregate([{$match:{gender:"MALE"}},{$group:{_id:"$gender",maxsal:{$max:"$sal"}}}])
[ { _id: 'MALE', maxsal: 7000 } ]
facultydb> db.emp.aggregate({$match:{}})
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 }
]
facultydb> db.emp.aggregate([{$match:{gender:"MALE"}},{$group:{_id:"$gender",maxsal:{$max:"$sal"}}}])
[ { _id: 'MALE', maxsal: 7000 } ]
facultydb> db.emp.aggregate([{$match:{gender:"MALE"}},{$group:{_id:"$gender",maxsal:{$max:"$sal"}}},{$project:{maxsal:1,_id:0}}])
[ { maxsal: 7000 } ]
facultydb> db.emp.aggregate([{$match:{}},{$group:{_id:"$gender",maxsal:{$max:"$sal"}}},{$project:{maxsal:1,_id:0}}])
[ { maxsal: 7000 }, { maxsal: 5000 } ]
```



```
facultydb> db.emp.aggregate([{$match:{}},{$group:{_id:"$gender",maxsal:{$max:"$sal"}}},{$project:{maxsal:1,_id:0}},{$sort:{sal:1}}])
[ { maxsal: 5000 }, { maxsal: 7000 } ]
facultydb> db.emp.aggregate([{$match:{}},{$group:{_id:"$gender",maxsal:{$max:"$sal"}}},{$project:{_id:0}}])
[ { maxsal: 5000 }, { maxsal: 7000 } ]
facultydb> db.emp.aggregate([{$match:{}},{$group:{_id:"$gender",maxsal:{$max:"$sal"}}},{$project:{_id:0}},{$sort:{maxsal:1}}])
[ { maxsal: 5000 }, { maxsal: 7000 } ]
facultydb> db.emp.aggregate([{$match:{}},{$group:{_id:"$gender",maxsal:{$max:"$sal"}}},{$project:{_id:0}},{$sort:{maxsal:-1}}])
[ { maxsal: 7000 }, { maxsal: 5000 } ]
facultydb> db.emp.aggregate([{$match:{}},{$group:{_id:"$gender",maxsal:{$max:"$sal"}}},{$project:{_id:0}},{$sort:{maxsal:-1}},{$limit:1}])
[ { maxsal: 7000 } ]
facultydb> db.emp.find({})
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 }
]
facultydb> db.emp.find({sal:{$not:{$gt:2000}}})
[ { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 } ]
facultydb> db.emp.find({})
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 }
]
facultydb> db.emp.updateMany({},{$set:{location:"VIZAG"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 4,
  modifiedCount: 4,
  upsertedCount: 0
}
facultydb> db.emp.find({})
[
  {
    _id: 1,
    name: 'A',
    gender: 'FEMALE',
    sal: 1000.5,
    location: 'VIZAG'
  },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5, location: 'VIZAG' },
  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000, location: 'VIZAG' },
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000, location: 'VIZAG' }
]
facultydb> db.emp.updateMany({},{$unset:{location:""}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 4,
```

Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
32°C
Partly sunny
Search
ENG
IN



```
facultydb> db.emp.updateMany({},{$unset:{location:""}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 4,
  modifiedCount: 4,
  upsertedCount: 0
}
facultydb> db.emp.find({})
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 }
]
facultydb> db.emp.aggregate([ {$match:{}} , {$out:"person"}  ])

facultydb> show collections
emp
faculty
person
facultydb> db.person.find({})
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 }
]
facultydb>
```

Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
32°C
Partly sunny
Search
ENG
IN

DAY-4

Microsoft Windows [Version 10.0.22000.2538]

(c) Microsoft Corporation. All rights reserved.

```
C:\Users\Bhuvaneswari>mongosh
Current Mongosh Log ID: 65cda61f7030bc56672e2e3e
Connecting to:
mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.4
Using MongoDB:        7.0.5
Using Mongosh:        2.1.4

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

------
   The server generated these startup warnings when booting
   2024-02-15T11:18:50.754+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

test> show dbs
admin      40.00 KiB
config     72.00 KiB
facultydb  184.00 KiB
local      72.00 KiB
test> use facultydb
switched to db facultydb
facultydb> show collections
emp
faculty
person
facultydb> db.emp.find({})
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },

  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },

  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },

  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 }

]

facultydb> db.emp.updateOne({_id:1},{$set:{location:"VJA"}})

{

  acknowledged: true,

  insertedId: null,

  matchedCount: 1,

  modifiedCount: 1,

  upsertedCount: 0

}

facultydb> db.emp.updateOne({_id:2},{$set:{location:"VIZAG"}})

{

  acknowledged: true,

  insertedId: null,

  matchedCount: 1,

  modifiedCount: 1,

facultydb>

}

facultydb> db.emp.updateOne({_id:3},{$set:{location:"GUNTUR"}})

{

  acknowledged: true,

  insertedId: null,

  matchedCount: 1,

  modifiedCount: 1,
  upsertedCount: 0
}
facultydb> db.emp.updateOne({_id:4},{$set:{location:"SHIRIDI"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
facultydb> db.emp.find({})
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5, location: 'VJA' },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5, location: 'VIZAG' },
  {
    _id: 3,
    name: 'C',
    gender: 'FEMALE',
    sal: 5000,
    location: 'GUNTUR'
  },
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000, location: 'SHIRIDI' }
]
facultydb> db.emp.find({},{name:1,location:1})
[
  { _id: 1, name: 'A', location: 'VJA' },
  { _id: 2, name: 'B', location: 'VIZAG' },
  { _id: 3, name: 'C', location: 'GUNTUR' },
  { _id: 4, name: 'C', location: 'SHIRIDI' }
]
facultydb> db.emp.find({location:/^V/})  //it displays the location that starts with the v
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5, location: 'VJA' },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5, location: 'VIZAG' }
]
facultydb> db.emp.find({location:/I$/})  //it displays the location that ends with i
[
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000, location: 'SHIRIDI' }
]
facultydb> db.emp.find({location:/^VI/})  //it displays the location that  starts with VI
[
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5, location: 'VIZAG' }
]
facultydb> db.emp.find({location:/UR$/})  //it displays the location that  ends with UR
[
  {
    _id: 3,
    name: 'C',
    gender: 'FEMALE',
    sal: 5000,
    location: 'GUNTUR'
  }
]
facultydb> db.emp.find({location:/NT/})  //it displays the location that contains NT
```

```
[
  {
    _id: 3,
    name: 'C',
    gender: 'FEMALE',
    sal: 5000,
    location: 'GUNTUR'
  }
]
facultydb> db.emp.find({location:/I.*A/})  // * contains  means zero or more character between I and A
 [
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5, location: 'VIZAG' }
]
facultydb> db.emp.find({location:/^S.*I$/})   //starts with S and ends with I and in between can have n characters
[
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000, location: 'SHIRIDI' }
]
facultydb> db.createCollection("student")
{ ok: 1 }
facultydb> show collections
emp
faculty
person
student
facultydb> db.student.insertOne({_id:101,name:"MONIKA",skills:["AWS","MSWD"]})
{ acknowledged: true, insertedId: 101 }
```

```
facultydb> db.student.insertOne({_id:102,name:"TEJU",skills:["AWS"]})

{ acknowledged: true, insertedId: 102 }

facultydb> db.student.insertOne({_id:103,name:"PRABHAS",skills:["AWS","PFSD"]})

{ acknowledged: true, insertedId: 103 }

facultydb> db.student.find({})

[

  { _id: 101, name: 'MONIKA', skills: [ 'AWS', 'MSWD' ] },

  { _id: 102, name: 'TEJU', skills: [ 'AWS' ] },

  { _id: 103, name: 'PRABHAS', skills: [ 'AWS', 'PFSD' ] }

]

facultydb> db.emp.find({})

[

  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5, location: 'VJA' },

  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5, location: 'VIZAG' },

  {

    _id: 3,

    name: 'C',

    gender: 'FEMALE',

    sal: 5000,

    location: 'GUNTUR'

  },

  { _id: 4, name: 'C', gender: 'MALE', sal: 7000, location: 'SHIRIDI' }

]

facultydb> db.emp.find({}).forEach( (e) => {

... print(e)

... })   //it displays emp collection through for each loop with no condition

{ _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5, location: 'VJA' }
{ _id: 2, name: 'B', gender: 'MALE', sal: 2000.5, location: 'VIZAG' }

{ _id: 3, name: 'C', gender: 'FEMALE', sal: 5000, location: 'GUNTUR' }

{ _id: 4, name: 'C', gender: 'MALE', sal: 7000, location: 'SHIRIDI' }


facultydb> db.emp.find({sal:{$gt:2000}}).forEach((e) => { print(e); }) //it displays emp collection through for each loop with  condition that sal should be greater than the 2000


{ _id: 2, name: 'B', gender: 'MALE', sal: 2000.5, location: 'VIZAG' }

{ _id: 3, name: 'C', gender: 'FEMALE', sal: 5000, location: 'GUNTUR' }

{ _id: 4, name: 'C', gender: 'MALE', sal: 7000, location: 'SHIRIDI' }


facultydb> db.emp.find({sal:{$gt:2000}}).forEach((e) => { print(e._id); })

2

3

4


facultydb> db.emp.find({sal:{$gt:2000}}).forEach((e) => { print(e._id+","+e.name); }) //it displays  _id and name of emp collection through for each loop with  condition that sal should be greater than the 2000


2,B

3,C

4,C


facultydb>
```