

2200030722

P.veda Vyshnavi

Section:-32

Microsoft Windows [Version 10.0.22631.3085]

(c) Microsoft Corporation. All rights reserved.

C:\Users\Dell>mongosh

Current Mongosh Log ID: 65c46c9921831ee232061d8b

Connecting to:

mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.3

Using MongoDB: 7.0.5

Using Mongosh: 2.1.3

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

The server generated these startup warnings when booting

2024-02-08T11:08:58.717+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

local 72.00 KiB

test> **use facultydb** // used to create a new db or to switch to an existing one

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 collection
```

```
{ ok: 1 }
```

```
facultydb> show dbs
```

```
admin    40.00 KiB
```

```
config   108.00 KiB
```

```
facultydb 8.00 KiB
```

```
local    72.00 KiB
```

```
facultydb> show collections //to display the collections
```

```
faculty
```

```
facultydb> db
```

```
facultydb
```

```
facultydb> db.faculty.drop() //to drop the collections it returns true or false
```

```
true
```

```
facultydb> show collections
```

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

```
{ ok: 1, dropped: 'facultydb' }
```

```
facultydb> show dbs
```

```
admin    40.00 KiB
```

```
config   108.00 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") // again creating the database called as faculty
```

```
{ ok: 1 }
```

```
facultydb> show collections
```

```
faculty
```

```
facultydb> //insert document
```

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

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

```
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('65c46c9921831ee232061d8b') }
}
```

```
facultydb> db.faculty.find() //used to find the faculty data
```

```
[
  {
    _id: ObjectId('65c4711695925835599bd6d8'),
    fid: 4654,
    fname: 'SURYA'
  }
]
```

```
facultydb> db.faculty.find().pretty() //use to find the data
```

```
[
  {
    _id: ObjectId('65c4711695925835599bd6d8'),
    fid: 4654,
    fname: 'SURYA'
  }
]
```

```
facultydb> db.faculty.find() //used to find the faculty data
```

```
[
  {
    _id: ObjectId('65c4711695925835599bd6d8'),
    fid: 4654,
    fname: 'SURYA'
  }
]
```

```

    }
  ]
facultydb> db.faculty.insertOne({fid:4414,fname:"HARI"}) // inserting the data contains one object
{
  acknowledged: true,
  insertedId: ObjectId('65c4729f95925835599bd6d9')
}
facultydb> db.faculty.insertMany([{fid:5000},{fid:1234,fname:"KLU"}]) //inserting the data
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('65c4734d95925835599bd6da'),
    '1': ObjectId('65c4734d95925835599bd6db')
  }
}
facultydb> db.faculty.find()
[
  {
    _id: ObjectId('65c4711695925835599bd6d8'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65c4729f95925835599bd6d9'),
    fid: 4414,
    fname: 'HARI'
  },
  { _id: ObjectId('65c4734d95925835599bd6da'), fid: 5000 },
  {
    _id: ObjectId('65c4734d95925835599bd6db'),
    fid: 1234,

```

```
    fname: 'KLU'
  }
]
facultydb> db.faculty.find().size() //it give the size of the faculty
4
facultydb> db.faculty.countDocuments() //count operation
4
facultydb> db.faculty.find().limit(2) //limit operation
[
  {
    _id: ObjectId('65c4711695925835599bd6d8'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65c4729f95925835599bd6d9'),
    fid: 4414,
    fname: 'HARI'
  }
]
facultydb> db.faculty.find().limit(140)
[
  {
    _id: ObjectId('65c4711695925835599bd6d8'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65c4729f95925835599bd6d9'),
    fid: 4414,
    fname: 'HARI'
```

```

},
{ _id: ObjectId('65c4734d95925835599bd6da'), fid: 5000 },
{
  _id: ObjectId('65c4734d95925835599bd6db'),
  fid: 1234,
  fname: 'KLU'
}
]

```

```
facultydb> db.faculty.find().sort({fid:1}) //asc //sort the data
```

```

[
  {
    _id: ObjectId('65c46c9921831ee232061d8b'),
    fid: 1234,
    fname: 'KLU'
  },
  {
    _id: ObjectId('65c46c9921831ee232061d8b'),
    fid: 4414,
    fname: 'HARI'
  },
  {
    _id: ObjectId('65c46c9921831ee232061d8b'),
    fid: 4654,
    fname: 'SURYA'
  },
  { _id: ObjectId('65c46c9921831ee232061d8c'), fid: 5000 }
]

```

```
facultydb> db.faculty.find().sort({fid:-1}) //descending order
```

```

[
  { _id: ObjectId('65c46c9921831ee232061d8b'), fid: 5000 },
  {

```

```

    _id: ObjectId('65c46c9921831ee232061d8b'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65c46c9921831ee232061d8b'),
    fid: 4414,
    fname: 'HARI'
  },
  {
    _id: ObjectId('65c46c9921831ee232061d8b'),
    fid: 1234,
    fname: 'KLU'
  }
]
facultydb> db.faculty.find().sort() //sort operation
[
  {
    _id: ObjectId('65c46c9921831ee232061d8b'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65c46c9921831ee232061d8b'),
    fid: 4414,
    fname: 'HARI'
  },
  { _id: ObjectId('65c46c9921831ee232061d8b'), fid: 5000 },
  {
    _id: ObjectId('65c46c9921831ee232061d8b'),
    fid: 1234,

```

```
    fname: 'KLU'
  }
]
facultydb> db.faculty.find({fid:1234}) //used to find the data
```

```
[
  {
    _id: ObjectId('65c46c9921831ee232061d8b'),
    fid: 1234,
    fname: 'KLU'
  }
]
```

```
facultydb> db.faculty.updateOne({fid:1234},{ $set:{fname:"KLEF"}}) //update operation
```

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
facultydb> db.faculty.find().limit(1) //limit the number of documents returned by a query
```

```
[
  {
    _id: ObjectId('65c46c9921831ee232061d8b'),
    fid: 4654,
    fname: 'SURYA'
  }
]
```

```
facultydb> db.faculty.find().limit(2) //limit the number of documents returned by a query
```

```
[
  {
    _id: ObjectId('65c46c9921831ee232061d8b'),
```



```
    fid: 4654,  
    fname: 'SURYA'  
  },  
  {  
    _id: ObjectId('65c46c9921831ee232061d8b'),  
    fid: 4414,  
    fname: 'HARI'  
  }  
]  
facultydb> db.faculty.find().limit(3) //limit the number of documents returned by a query
```

```
[  
  {  
    _id: ObjectId('65c46c9921831ee232061d8b'),  
    fid: 4654,  
    fname: 'SURYA'  
  },  
  {  
    _id: ObjectId('65c46c9921831ee232061d8b'),  
    fid: 4414,  
    fname: 'HARI'  
  },  
  { _id: ObjectId('65c46c9921831ee232061d8c'), fid: 5000 }  
]
```

```
facultydb> db.faculty.find().limit(4) //limit the number of documents returned by a query  
[  
  {  
    _id: ObjectId('65c46c9921831ee232061d8b'),  
    fid: 4654,  
    fname: 'SURYA'  
  },  
  {
```

```

    _id: ObjectId('65c46c9921831ee232061d8b'),
    fid: 4414,
    fname: 'HARI'
  },
  { _id: ObjectId('65c46c9921831ee232061d8c'), fid: 5000 },
  {
    _id: ObjectId('65c46c9921831ee232061d8b'),
    fid: 1234,
    fname: 'KLEF'
  }
]

facultydb> db.faculty.find({fid:5000}) //used to find data
[ { _id: ObjectId('65c46c9921831ee232061d8c'), fid: 5000 } ]

facultydb> db.faculty.updateOne({fid:5000},{ $set:{fname:"Harshitha",dept:"CSE"}}) //update
operation

{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

facultydb> db.faculty.find({fid:5000})

[
  {
    _id: ObjectId('65c46c9921831ee232061d8b'),
    fid: 5000,
    dept: 'CSE',
    fname: 'Harshitha'
  }
]

```

```

facultydb> db.faculty.countDocuments() //count operation
4
facultydb> db.faculty.find().size() //give the size of the faculty
4
facultydb> db.faculty.insertOne({fid:1764,fname:"Sanjana",dept:"CSE"}) //inserting data
{
  acknowledged: true,
  insertedId: ObjectId('65c46c9921831ee232061d8b')
}
facultydb> db.faculty.findOne({fid:1234})
{ _id: ObjectId('65c46c9921831ee232061d8b'), fid: 1234, fname: 'KLEF' }
facultydb> db.faculty.findOne({fid:1234,fname:"KLEF"})
{ _id: ObjectId('65c46c9921831ee232061d8b'), fid: 1234, fname: 'KLEF' }
facultydb> db.faculty.findOne({fid:1234,fname:"VIT"})
null
facultydb> db.faculty.find({dept:"CSE"})
[
  {
    _id: ObjectId('65c46c9921831ee232061d8b'),
    fid: 5000,
    dept: 'CSE',
    fname: 'Harshitha'
  },
  {
    _id: ObjectId('65c46c9921831ee232061d8b'),
    fid: 1764,
    fname: 'Sanjana',
    dept: 'CSE'
  }
]
facultydb> db.faculty.findOne({dept:"CSE"})

```

```
{
  _id: ObjectId('65c46c9921831ee232061d8b'),
  fid: 5000,
  dept: 'CSE',
  fname: 'Harshitha'
}
```

```
facultydb> db.faculty.find({fid:1234})
```

```
[
  {
    _id: ObjectId('65c46c9921831ee232061d8b'),
    fid: 1234,
    fname: 'KLEF'
  }
]
```

```
facultydb> db.faculty.deleteOne({fid:1234}) //delete operation
```

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

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

```
4
```

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

```
4
```

```
facultydb> db.faculty.find({dept:"CSE"})
```

```
[
  {
    _id: ObjectId('65c46c9921831ee232061d8b'),
    fid: 5000,
    dept: 'CSE',
    fname: 'Harshitha'
  },
  {
    _id: ObjectId('65c46c9921831ee232061d8b'),
    fid: 1764,
```

```

    fname: 'Sanjana',
    dept: 'CSE'
  }
]
facultydb> db.faculty.deleteMany({dept:"CSE"}) //delete many operation
{ acknowledged: true, deletedCount: 2 }
facultydb> db.faculty.find().size() //give size of the faculty
2
facultydb> db.faculty.countDocuments() //count operation
2
test> show dbs
admin    40.00 KiB
config   72.00 KiB
facultydb 72.00 KiB
local    184.00 KiB
test> use facultydb
switched to db facultydb
facultydb> db
facultydb
facultydb> show collections //used to show the collections
faculty
facultydb> db.faculty.find()
[
  {
    _id: ObjectId('65c4711e21831ee232061d8c'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65c4729b21831ee232061d8d'),
    fid: 4414,

```

```
    fname: 'HARI'
  }
]
facultydb> db.faculty.find({})
[
  {
    _id: ObjectId('65c4711e21831ee232061d8c'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65c4729b21831ee232061d8d'),
    fid: 4414,
    fname: 'HARI'
  }
]
facultydb> db.faculty.find()
[
  {
    _id: ObjectId('65c4711e21831ee232061d8c'),
    fid: 4654,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65c4729b21831ee232061d8d'),
    fid: 4414,
    fname: 'HARI'
  }
]
facultydb> db.faculty.updateMany({},{$set:{gender:"MALE"}}) //used to update data
{
```

```
    acknowledged: true,  
    insertedId: null,  
    matchedCount: 2,  
    modifiedCount: 2,  
    upsertedCount: 0  
  }  
}
```

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

```
[  
  {  
    _id: ObjectId('65c4711e21831ee232061d8c'),  
    fid: 4654,  
    fname: 'SURYA',  
    gender: 'MALE'  
  },  
  {  
    _id: ObjectId('65c4729b21831ee232061d8d'),  
    fid: 4414,  
    fname: 'HARI',  
    gender: 'MALE'  
  }  
]
```

```
facultydb> db.faculty.find({fid:4414})
```

```
[  
  {  
    _id: ObjectId('65c4729b21831ee232061d8d'),  
    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
}
```

```
facultydb> db.faculty.updateOne({fid:1234},{ $set:{fid:1870,fname:"SRI"}},{upsert:true})
```

```
{
  acknowledged: true,
  insertedId: ObjectId('65c5a6e584ef1aa33a8ae703'),
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1
}
```

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

```
[
  {
    _id: ObjectId('65c4711e21831ee232061d8c'),
    fid: 4654,
    fname: 'SURYA',
    gender: 'MALE'
  },

```



```

{
  _id: ObjectId('65c4729b21831ee232061d8d'),
  fid: 4414,
  fname: 'SEETHA',
  gender: 'FEMALE'
},
{
  _id: ObjectId('65c5a6e584ef1aa33a8ae703'),
  fid: 1870,
  fname: 'SRI'
}
]
facultydb> db.faculty.countDocuments() //used to count the data
3
facultydb> db.faculty.countDocuments({gender:"FEMALE"})
1
facultydb> db.faculty.find({fid:1870})
[
  {
    _id: ObjectId('65c5a6e584ef1aa33a8ae703'),
    fid: 1870,
    fname: 'SRI'
  }
]
facultydb> db.faculty.updateMany({fid:1870},{ $set:{gender:"FEMALE"}}) //used to update data
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

```

```

}
facultydb> db.faculty.findOne({fid:1870})
{
  _id: ObjectId('65c5a6e584ef1aa33a8ae703'),
  fid: 1870,
  fname: 'SRI',
  gender: 'FEMALE'
}
facultydb> db.faculty.countDocuments({gender:"FEMALE"})
2
facultydb> db.faculty.distinct("gender") //considers each element of the array as a separate
value
[ '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"} ]) //used to insert data
{ 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"} ]) //insert many operation
{ 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().skip(1) //used to skip the data
```

```
[ { _id: 2, name: 'B' }, { _id: 3, name: 'C' }, { _id: 4, name: 'D' } ]
```

```
facultydb> db.emp.find().skip(4)
```

```
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}) //sort the data
```

```
[
```

```
  { _id: 1, name: 'A' },
```

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

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

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

```
]
```

```
facultydb> db.emp.find({}).sort({name:-1}) //sort in descending order
```

```
[
```

```
  { _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}}) //used to updateone data
```

```
{  
  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"}})
```

```
{
  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.updateOne({_id:4},{$set:{name:"C",gender:"MALE"}})
```

```
{
  acknowledged: true,
```

```

insertedId: null,
matchedCount: 1,
modifiedCount: 1,
upsertedCount: 0
}
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})
[
  { _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})
[ { _id: 4, salary: 4000.5 } ]
facultydb> db.emp.find({_id:4},{_id:0,salary:1})
[ { salary: 4000.5 } ]
facultydb> db.emp.find({_id:3},{name:1,salary:1,_id:0}) //it return name and salary
[ { name: 'C', salary: 3000.5 } ]

```

```
facultydb> db.emp.find({salary:{$gt:2000}})
```

```
[
  { _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:2000}})
```

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

```
facultydb> db.emp.find({salary:{$gte:2000}})
```

```
[
  { _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:{$lte:2000}}) //lte:-lessthan or Equal to operation
```

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

```
facultydb> db.emp.find({salary:{$ne:2000.5}}) //ne:- selects the documents where the value of
the specified field is not equal to the specified value
```

```
[
  { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' },
  { _id: 3, name: 'C', salary: 3000.5, gender: 'FEMALE' },
  { _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.updateOne({salary:{$gt:3000}},{$set:{salary:5000}}) //gt:-greaterthan operation
```

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
facultydb> db.emp.updateMany({salary:{$gt:3000}},{$set:{salary:5000}}) //update manyoperation
```

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  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: 5000, gender: 'FEMALE' },
  { _id: 4, name: 'C', salary: 5000, gender: 'MALE' }
]
```

```
facultydb> db.emp.find({gender:{$in:["MALE","FEMALE"]}}) //comparision operator
```

```
[
  { _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:{$nin:["MALE"]}}) //nin is a notin operation
[
  { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' },
  { _id: 3, name: 'C', salary: 5000, gender: 'FEMALE' }
]
facultydb> db.emp.renameCollection("employee") //used to renaming
{ ok: 1 }
facultydb> show collections
employee
faculty
facultydb> db.employee.renameCollection("emp")
{ ok: 1 }
facultydb> db.emp.updateMany({},{$rename:{salary:"sal"}}) //used to renaming
{
  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({$and:[{gender:"FEMALE"},{sal:{gt:2000}}]}) //and operation
```

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

```
facultydb> db.emp.find({$or:[{gender:"FEMALE"},{sal:{gt:2000}}]}) //or operation
```

```
[
```

```
  { _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({$lt:[{gender:"FEMALE"},{sal:{gt:2000}}]}) //lt:-lessthen operation
```

MongoServerError: unknown top level operator: \$lt. If you have a field name that starts with a '\$' symbol, consider using \$getField or \$setField.

```
facultydb> db.emp.find({$and:[{gender:"FEMALE"},{sal:{lt:2000}}]})
```

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

```
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 }
```

```
]
```

```
test> show dbs
```

```
admin    40.00 KiB
```

```
config   60.00 KiB
```

facultydb 144.00 KiB

local 192.00 KiB

test> use facultydb

switched to db facultydb

facultydb> show collections

emp

faculty

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}) //sort based on salary

```
[
  { _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}).limit(1)

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

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> 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({$count:"totalcount"}) //it give total count
```

```
[ { totalcount: 4 } ]
```

```
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({$group:{_id:null,totalsal:{$sum:"$sal"}}}) //it give total salary
```

```
[ { _id: null, totalsal: 15001 } ]
```

```
facultydb> db.emp.aggregate({$group:{_id:null,avgsal:{$avg:"$sal"}}}) //it give avg salary
```

```
[ { _id: null, avgsal: 3750.25 } ]
```

```
facultydb> db.emp.aggregate({$group:{_id:null,minsal:{$min:"$sal"}}}) //return minimum value
```

```
[ { _id: null, minsal: 1000.5 } ]
```

```
facultydb> db.emp.aggregate({$group:{_id:null,maxsal:{$max:{$sal}}}}) //it give the 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: 'MALE', totalsal: 9000.5 },
  { _id: 'FEMALE', totalsal: 6000.5 }
]
```

```
facultydb> db.emp.aggregate({$group:{_id:{$gender},avgsal:{$avg:{$sal}}}}) //it give the avg sal and gender
```

```
[
  { _id: 'MALE', avgsal: 4500.25 },
  { _id: 'FEMALE', avgsal: 3000.25 }
]
```

```
facultydb> db.emp.aggregate({$group:{_id:{$gender},minsal:{$min:{$sal}}}}) it give gender and minsal
```

```
[ { _id: 'FEMALE', minsal: 1000.5 }, { _id: 'MALE', minsal: 2000.5 } ]
```

```
facultydb> db.emp.aggregate({$group:{_id:{$gender},maxsal:{$max:{$sal}}}}) it give gender and maxsal
```

```
[ { _id: 'FEMALE', maxsal: 5000 }, { _id: 'MALE', maxsal: 7000 } ]
```

```
facultydb> db.emp.aggregate({$match:{gender:"MALE"}}) //match used to filter documents
```

```
[
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _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:{}})
```

```
[
  { _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:{max:1,_id:0}}])
```

```
[ {} ]
```

facultydb>

```
db.emp.aggregate([{$match:{gender:"MALE"}},{ $group:{_id:"$gender",maxsal:{$max:"$sal"}}},{ $project:{maxsal:1,_id:0}}]) //group is used to separates documents into groups according to group key
```

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

facultydb>

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

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

facultydb>

```
db.emp.aggregate([{$match:{}},{ $group:{_id:"$gender",maxsal:{$max:"$sal"}}},{ $project:{maxsal:1,_id:0}},{$sort:{sal:1}}]) //project is used to extract only the fields that we need
```

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

facultydb>

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

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

facultydb>

```
db.emp.aggregate([{$match:{}},{ $group:{_id:"$gender",maxsal:{$max:"$sal"}}},{ $project:{maxsal:1,_id:0}},{$sort:{maxsal:-1}}]) //it returned maximum value
```

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

facultydb>

```
db.emp.aggregate([{$match:{}},{ $group:{_id:"$gender",maxsal:{$max:"$sal"}}},{ $project:{maxsal:1,_id:0}},{$sort:{maxsal:-1}},{$limit:1}]) //limit rows returned
```

```
[ { 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"}}) //it is used to modifies an existing documents
```

```
{
  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:""}}) //it remove the location(unset)

```

```

{
  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"}]) //out is used to take the documents
returned by aggregation and write them to a specified collections

```

```

facultydb> show collections

```

```

emp
faculty
person

```

```

facultydb> db.person.find({})

```

```

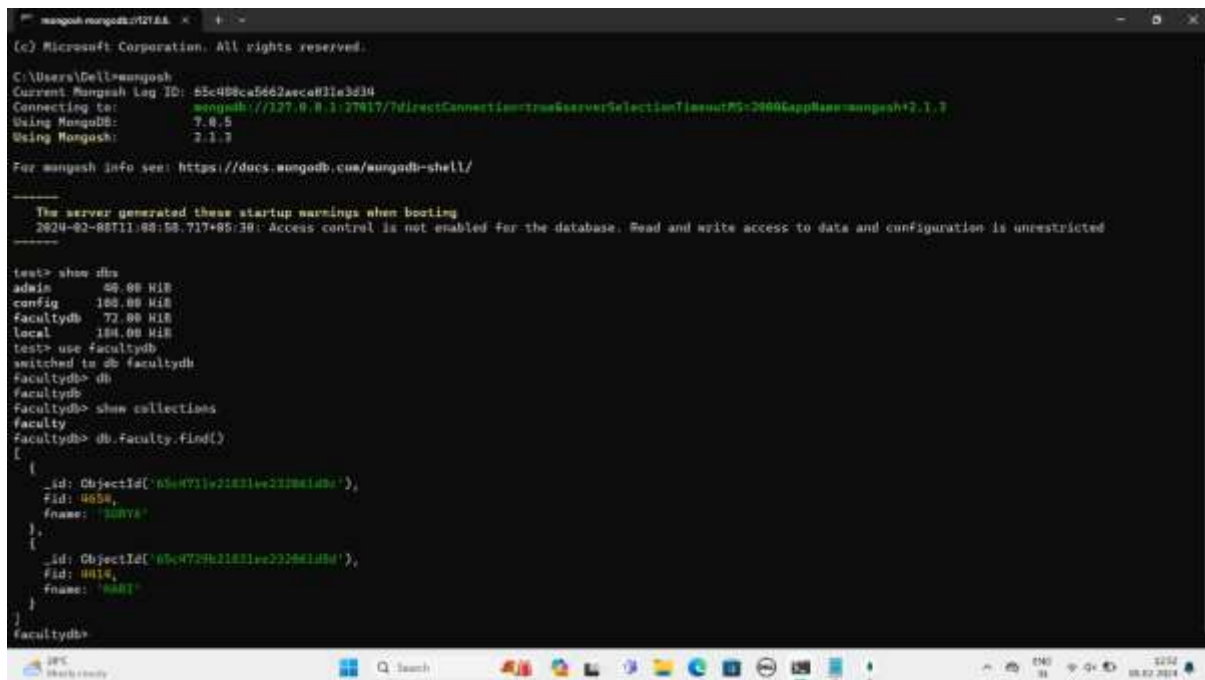
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },

```



```
{_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 }
]
```

screenshots of the above :-



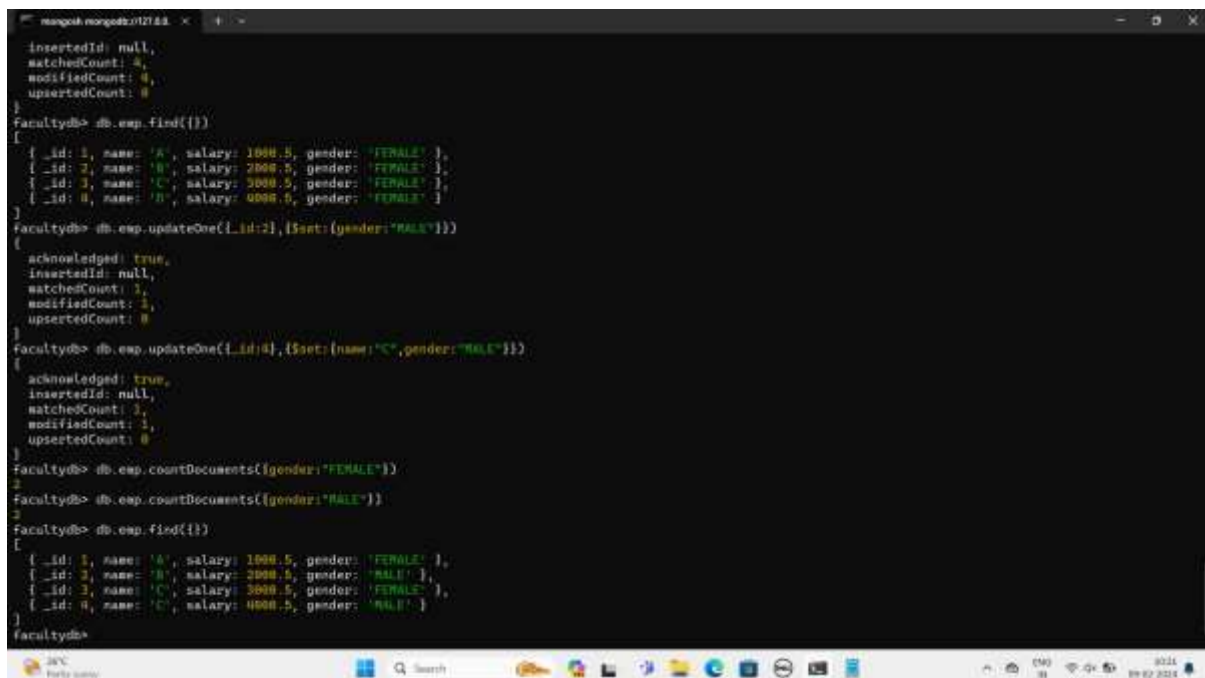
```
(c) Microsoft Corporation. All rights reserved.

C:\Users\Gell>mongosh
Current MongoDB Log ID: 85c488ca5d62aeca821e3234
Connecting to: mongosh://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.7
Using MongoDB: 7.0.5
Using Mongosh: 2.1.7

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

-----
The server generated these startup warnings when booting
2024-02-08T11:08:58.717+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

test> show dbs
admin      48.00 MiB
config     100.00 MiB
facultydb  72.00 MiB
local      104.00 MiB
test> use facultydb
switched to db facultydb
facultydb> db
facultydb
facultydb> show collections
faculty
facultydb> db.faculty.find()
[
  {
    _id: ObjectId('65c4711e21831e4232881d8c'),
    fid: 4058,
    fname: 'SURYA'
  },
  {
    _id: ObjectId('65c4729b21831e4232881d8d'),
    fid: 4014,
    fname: 'RAH'
  }
]
facultydb>
```



```
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: 'B', 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.updateOne({_id:4},{set:{name:'C',gender:'MALE'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
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>
```

```
mongosh mongosh:2.17.0.1 > use facultydb
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})
[ { _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})
[ { _id: 4, salary: 4000.5 } ]
facultydb> db.emp.find({_id:4}, {_id:0, salary:1})
[ { salary: 4000.5 } ]
facultydb> db.emp.find({_id:3}, {name:1, salary:1, _id:0})
[ { name: 'C', salary: 3000.5 } ]
facultydb> db.emp.find({salary:{$gt:3000}})
[ { _id: 3, name: 'B', salary: 2000.5, gender: 'MALE' },
  { _id: 3, name: 'C', salary: 3000.5, gender: 'FEMALE' },
  { _id: 4, name: 'C', salary: 4000.5, gender: 'MALE' } ]
facultydb> db.emp.find({salary:{$lt:2000}})
[ { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' } ]
facultydb> db.emp.find({salary:{$gte:2000}})
[ { _id: 2, name: 'B', salary: 2000.5, gender: 'MALE' },
  { _id: 3, name: 'C', salary: 3000.5, gender: 'FEMALE' },
  { _id: 4, name: 'C', salary: 4000.5, gender: 'MALE' } ]
facultydb> db.emp.find({salary:{$lte:2000}})
[ { _id: 1, name: 'A', salary: 1000.5, gender: 'FEMALE' } ]
facultydb>
```

```
Microsoft Windows [Version 10.0.22021.3085]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Dell>mongosh
Current Mongosh Log ID: 65cb506955505eb6478f549a
Connecting to:  mongosh://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.3
Using MongoDB: 7.0.5
Using Mongosh: 2.1.3
mongosh 2.1.4 is available for download: https://www.mongodb.com/try/download/shell

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

-----
The server generated these startup warnings when booting
2024-02-13T15:40:27.937485:38: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

test> show dbs
admin    48.00 MiB
config   48.00 MiB
facultydb 140.00 MiB
local    102.00 MiB
test> use facultydb
switched to db facultydb
facultydb> show collections
emp
faculty
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: 3000 },
  { _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: 3000 },
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 } ]
```

```

mongosh mongos2:27481 >
[ { _id: 0, name: 'C', gender: 'MALE', sal: 7000 },
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 5000 },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _id: 3, name: 'C', gender: 'FEMALE', sal: 1000.5 } ]
facultydb> db.emp.find().sort({sal:-1}).limit(1)
[ { _id: 0, name: 'C', gender: 'MALE', sal: 7000 } ]
facultydb> db.emp.find({name:'C'}).sort({sal:-1}).limit(1)
[ { _id: 0, name: 'C' } ]
facultydb> db.emp.find({name:'C',_id:0}).sort({sal:-1}).limit(1)
[ { name: 'C' } ]
facultydb> //Aggregate functions or multi-document functions
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({$count:'totalCount'})
[ { totalCount: 4 } ]
facultydb> db.emp.find()
[
  { _id: 1, name: 'A', gender: 'FEMALE', sal: 1000.5 },
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _id: 3, name: 'C', gender: 'FEMALE', sal: 5000 },
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 } ]
facultydb> db.emp.aggregate({$group:{_id:null,totalsal:{$sum:'$sal'}}})
[ { _id: null, totalsal: 15000 } ]
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: 15000 } ]
facultydb> db.emp.aggregate({$group:{_id:'$gender',totalsal:{$sum:'$sal'}}})

```

```

mongosh mongos2:27481 >
[ { _id: null, maxsal: 7000 } ]
facultydb> db.emp.aggregate({$group:{_id:null,totalsal:{$sum:'$sal'}}})
[ { _id: null, totalsal: 15000 } ]
facultydb> db.emp.aggregate({$group:{_id:'$gender',totalsal:{$sum:'$sal'}}})
[
  { _id: 'MALE', totalsal: 9000.5 },
  { _id: 'FEMALE', totalsal: 6000.5 } ]
facultydb> db.emp.aggregate({$group:{_id:'$gender',avgsal:{$avg:'$sal'}}})
[
  { _id: 'MALE', avgsal: 4500.25 },
  { _id: 'FEMALE', avgsal: 3000.25 } ]
facultydb> db.emp.aggregate({$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:'$sal'}}})
[ { _id: 'FEMALE', maxsal: 5000 }, { _id: 'MALE', maxsal: 7000 } ]
facultydb> db.emp.aggregate({$match:{gender:'MALE'}})
[
  { _id: 2, name: 'B', gender: 'MALE', sal: 2000.5 },
  { _id: 4, name: 'C', gender: 'MALE', sal: 7000 } ]
facultydb> db.emp.aggregate({$match:{gender:'MALE'}},{$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'}}},{$project:{maxsal:1,_id:0}}])
[ { } ]
facultydb> db.emp.aggregate([{$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: 5000 }, { maxsal: 7000 } ]
facultydb>

```

MongoDB Compass - localhost:27017/facultydb.EMP

localhost:27017

My Queries Performance Databases

admin config facultydb

emp faculty local

facultydb.EMP

Documents Aggregations Schema Indexes Validation

Find Type a query (Field: "value") or Demote to query

EXPLAIN RESET FIND ADVANCED

1-4 of 4

ADD DATA EXPORT DATA UPDATE DELETE

```
{ "_id": 1, "name": "A", "gender": "MALE", "sal": 1000.0 }
{ "_id": 2, "name": "B", "gender": "MALE", "sal": 2000.0 }
{ "_id": 3, "name": "C", "gender": "MALE", "sal": 3000.0 }
{ "_id": 4, "name": "D", "gender": "FEMALE", "sal": 1500.0 }
```

MongoDB Compass - localhost:27017/facultydb.faculty

localhost:27017

My Queries Performance Databases

admin config facultydb

emp faculty local

facultydb.faculty

Documents Aggregations Schema Indexes Validation

Find Type a query (Field: "value") or Demote to query

EXPLAIN RESET FIND ADVANCED

1-3 of 3

ADD DATA EXPORT DATA UPDATE DELETE

```
{ "_id": ObjectId("63c4713a2185a4220e030c7c"), "Fid": 1053, "Fname": "John", "gender": "MALE" }
{ "_id": ObjectId("63c4713a2185a4220e030c7d"), "Fid": 4434, "Fname": "John", "gender": "MALE" }
{ "_id": ObjectId("63c4713a2185a4220e030c7e"), "Fid": 1070, "Fname": "John", "gender": "MALE" }
```

