

```

mongosh mongodb://127.0.0.1:27021/
NTI-TRAINING> use FacultySystemDB
switched to db FacultySystemDB
FacultySystemDB> db.createCollection("Students")
{ ok: 1 }
FacultySystemDB> show collections
Students
FacultySystemDB> db.deleteOne(Students)
ReferenceError: Students is not defined
FacultySystemDB> db.deleteOne("Students")
TypeError: db.deleteOne is not a function
FacultySystemDB> dp.Students.drop()
ReferenceError: dp is not defined
FacultySystemDB> db.Students.drop()
true
FacultySystemDB> db.createCollection("student")
{ ok: 1 }
FacultySystemDB> show collections
student
FacultySystemDB> db.student.insertOne({fName:"Mrihan",lName:"Mohamed",age:26,faculty:{name:"FCI",address:}
...  "Minia"},grades:[{cname:"c#",grade:60,pass:true},{cname:"JS",grade:100,
...
Uncaught:
SyntaxError: Unexpected character '}'. (1:28)

> 1 | db.student.insertOne({fName:"Mrihan",lName:"Mohamed",age:26,faculty:{name:"FCI",address:}
    |                                     ^
    |
    |
2 |   "Minia"},grades:[{cname:"c#",grade:60,pass:true},{cname:"JS",grade:100,
3 |
4 |

FacultySystemDB> db.student.insrertOne({fname:"Ahmed", lname:"Toba", age:22, faculty:{name:"BSU", address:"BeniSuef"}, grades:[{cname:"mongoDB", grade: 100,
pass:true}, {cname:"nodejs", grade: 45, pass:true}], isFired:false})
TypeError: db.student.insrertOne is not a function
FacultySystemDB> db.student.insertOne({fname:"Ahmed", lname:"Toba", age:22, faculty:{name:"BSU", address:"BeniSuef"}, grades:[{cname:"mongoDB", grade: 100,
pass:true}, {cname:"nodejs", grade: 45, pass:true}], isFired:false})
{
  acknowledged: true,
  insertedId: ObjectId('686e50c96abcb3d174718dca')
}
FacultySystemDB> db.student.find()
[

```

```

{
  _id: ObjectId('686e50c96abcb3d174718dca'),
  fname: 'Ahmed',
  lname: 'Toba',
  age: 22,
  faculty: { name: 'BSU', address: 'BeniSuef' },
  grades: [
    { cname: 'mongoDB', grade: 100, pass: true },
    { cname: 'nodejs', grade: 45, pass: true }
  ],
  isFired: false
}
]
FacultySystemDB> db.insertMany({fname:"Ahmed", lname:"Toba", age:22, faculty:{name:"BSU", address:"BeniSuef"}, grades:[{cname:"mongoDB", grade: 100, pass:true}, {cname:"nodejs", grade: 45, pass:true}], isFired:false})
... TypeError: db.student.insrertOne is not a function
... FacultySystemDB> db.student.insertOne({fname:"Ahmed", lname:"Toba", age:22, faculty:{name:"BSU", address:"BeniSuef"}, grades:[{cname:"mongoDB", grade: 100, pass:true}, {cname:"nodejs", grade: 45, pass:t
Uncaught:
SyntaxError: Missing semicolon. (2:32)

  1 | db.insertMany({fname:"Ahmed", lname:"Toba", age:22, faculty:{name:"BSU", address:"BeniSuef"}, grades:[{cname:"mongoDB", grade: 100, pass:true}, {cname:"nodejs", grade: 45, pass:true}], isFired:false})
> 2 | ^
    |
  3 | FacultySystemDB> db.student.insertOne({fname:"Ahmed", lname:"Toba", age:22, faculty:{name:"BSU", address:"BeniSuef"}, grades:[{cname:"mongoDB", grade: 100, pass:true}, {cname:"nodejs", grade: 45, pass:t
  4 |

FacultySystemDB> db.student.insertMany({fname:"Aya", lname:"Ahmed", age:20, faculty:{name:"BSU", address:"BeniSuef"}, grades:[{cname:"mongoDB", grade: 20, pass:flase}, {cname:"nodejs", grade: 80, pass:true}], isFired:false})
ReferenceError: flase is not defined
FacultySystemDB> db.student.insertMany({fname:"Aya", lname:"Ahmed", age:20, faculty:{name:"BSU", address:"BeniSuef"}, grades:[{cname:"mongoDB", grade: 20, pass:false}, {cname:"nodejs", grade: 80, pass:true}], isFired:false})
MongoInvalidArgumentError: Argument "docs" must be an array of documents
FacultySystemDB> db.student.insertMany([{{fname:"Aya", lname:"Ahmed", age:20, faculty:{name:"BSU", address:"BeniSuef"}, grades:[{cname:"mongoDB", grade: 20, pass:false}, {cname:"nodejs", grade: 80, pass:true}], isFired:false},{fname:"Nada", lname:"Yasser", age:21, faculty:{name:"CS", address:"BeniSuef"}, grades:[{cname:"dataStructure", grade: 10, pass:false}, {cname:"C++", grade: 8, pass:false}], isFired:true}]]
{

```

```

acknowledged: true,
insertedIds: {
  '0': ObjectId('686e51ea6abcb3d174718dcb'),
  '1': ObjectId('686e51ea6abcb3d174718dcc')
}
}
FacultySystemDB> db.student.find()
[
  {
    _id: ObjectId('686e50c96abcb3d174718dca'),
    fname: 'Ahmed',
    lname: 'Toba',
    age: 22,
    faculty: { name: 'BSU', address: 'BeniSuef' },
    grades: [
      { cname: 'mongoDB', grade: 100, pass: true },
      { cname: 'nodejs', grade: 45, pass: true }
    ],
    isFired: false
  },
  {
    _id: ObjectId('686e51ea6abcb3d174718dcb'),
    fname: 'Aya',
    lname: 'Ahmed',
    age: 20,
    faculty: { name: 'BSU', address: 'BeniSuef' },
    grades: [
      { cname: 'mongoDB', grade: 20, pass: false },
      { cname: 'nodejs', grade: 80, pass: true }
    ],
    isFired: false
  },
  {
    _id: ObjectId('686e51ea6abcb3d174718dcc'),
    fname: 'Nada',
    lname: 'Yasser',
    age: 21,
    faculty: { name: 'CS', address: 'BeniSuef' },
    grades: [

```

```

    grades: [
      { cname: 'dataStructure', grade: 10, pass: false },
      { cname: 'C++', grade: 8, pass: false }
    ],
    isFired: true
  }
]
FacultySystemDB>
(To exit, press Ctrl+C again or Ctrl+D or type .exit)
FacultySystemDB> db.student.insertOne({fname:"Aly", lname:"Nasser", age: 18, faculty:{name:"Science", address:"Cairo"}, grades:[{cname:"Physics", grade: 85, pass:true}, {cname:"Math", grade: 65, pass:true}], isFired:false})
{
  acknowledged: true,
  insertedId: ObjectId('686e52996abcb3d174718dcd')
}
FacultySystemDB> db.student.find()
[
  {
    _id: ObjectId('686e50c96abcb3d174718dca'),
    fname: 'Ahmed',
    lname: 'Toba',
    age: 22,
    faculty: { name: 'BSU', address: 'BeniSuef' },
    grades: [
      { cname: 'mongoDB', grade: 100, pass: true },
      { cname: 'nodejs', grade: 45, pass: true }
    ],
    isFired: false
  },
  {
    _id: ObjectId('686e51ea6abcb3d174718dcb'),
    fname: 'Aya',
    lname: 'Ahmed',
    age: 20,
    faculty: { name: 'BSU', address: 'BeniSuef' },
    grades: [
      { cname: 'mongoDB', grade: 20, pass: false },
      { cname: 'nodejs', grade: 80, pass: true }
    ],
    isFired: false
  }
]

```



```

},
{
  _id: ObjectId('686e51ea6abcb3d174718dcc'),
  fname: 'Nada',
  lname: 'Yasser',
  age: 21,
  faculty: { name: 'CS', address: 'BeniSuef' },
  grades: [
    { cname: 'dataStructure', grade: 10, pass: false },
    { cname: 'C++', grade: 8, pass: false }
  ],
  isFired: true
},
{
  _id: ObjectId('686e52996abcb3d174718dcd'),
  fname: 'Aly',
  lname: 'Nasser',
  age: 18,
  faculty: { name: 'Science', address: 'Cairo' },
  grades: [
    { cname: 'Physics', grade: 85, pass: true },
    { cname: 'Math', grade: 65, pass: true }
  ],
  isFired: false
}
]
FacultySystemDB> db.student.find({fname:"Ahmed"})
[
  {
    _id: ObjectId('686e50c96abcb3d174718dca'),
    fname: 'Ahmed',
    lname: 'Toba',
    age: 22,
    faculty: { name: 'BSU', address: 'BeniSuef' },
    grades: [
      { cname: 'mongoDB', grade: 100, pass: true },
      { cname: 'nodejs', grade: 45, pass: true }
    ],
    isFired: false
  }
]

```

```

FacultySystemDB> db.student.find({fname:"Aya"})
[
  {
    _id: ObjectId('686e51ea6abcb3d174718dcb'),
    fname: 'Aya',
    lname: 'Ahmed',
    age: 20,
    faculty: { name: 'BSU', address: 'BeniSuef' },
    grades: [
      { cname: 'mongoDB', grade: 20, pass: false },
      { cname: 'nodejs', grade: 80, pass: true }
    ],
    isFired: false
  }
]
FacultySystemDB> db.student.find({$or:[{fname:"Ahmed"}, {lname:"Ahmed"}]})
[
  {
    _id: ObjectId('686e50c96abcb3d174718dca'),
    fname: 'Ahmed',
    lname: 'Toba',
    age: 22,
    faculty: { name: 'BSU', address: 'BeniSuef' },
    grades: [
      { cname: 'mongoDB', grade: 100, pass: true },
      { cname: 'nodejs', grade: 45, pass: true }
    ],
    isFired: false
  },
  {
    _id: ObjectId('686e51ea6abcb3d174718dcb'),
    fname: 'Aya',
    lname: 'Ahmed',
    age: 20,
    faculty: { name: 'BSU', address: 'BeniSuef' },
    grades: [
      { cname: 'mongoDB', grade: 20, pass: false },
      { cname: 'nodejs', grade: 80, pass: true }
    ],
  }
]

```

```

    isFired: false
  }
]
FacultySystemDB> db.student.find({$ne:[fname:"Ahmed"]})
Uncaught:
SyntaxError: Unexpected token, expected ",", (1:27)
> 1 | db.student.find({$ne:[fname:"Ahmed"]})
    |                               ^
    2 |

FacultySystemDB> db.student.find({$ne:[{fname:"Ahmed"}]})
MongoServerError[BadValue]: unknown top level operator: $ne. If you have a field name that starts with a '$' symbol, consider using $getField or $setField.
FacultySystemDB> db.student.find({fname:{ $ne:"Ahmed"}})
[
  {
    _id: ObjectId('686e51ea6abcb3d174718dcb'),
    fname: 'Aya',
    lname: 'Ahmed',
    age: 20,
    faculty: { name: 'BSU', address: 'BeniSuef' },
    grades: [
      { cname: 'mongoDB', grade: 20, pass: false },
      { cname: 'nodejs', grade: 80, pass: true }
    ],
    isFired: false
  },
  {
    _id: ObjectId('686e51ea6abcb3d174718dcc'),
    fname: 'Nada',
    lname: 'Yasser',
    age: 21,
    faculty: { name: 'CS', address: 'BeniSuef' },
    grades: [
      { cname: 'dataStructure', grade: 10, pass: false },
      { cname: 'C++', grade: 8, pass: false }
    ],
    isFired: true
  },
]

```

```

{
  _id: ObjectId('686e52996abcb3d174718dcd'),
  fname: 'Aly',
  lname: 'Nasser',
  age: 18,
  faculty: { name: 'Science', address: 'Cairo' },
  grades: [
    { cname: 'Physics', grade: 85, pass: true },
    { cname: 'Math', grade: 65, pass: true }
  ],
  isFired: false
}
]
FacultySystemDB> db.student.find({age:{$lt:21}})
[
  {
    _id: ObjectId('686e51ea6abcb3d174718dcb'),
    fname: 'Aya',
    lname: 'Ahmed',
    age: 20,
    faculty: { name: 'BSU', address: 'BeniSuef' },
    grades: [
      { cname: 'mongoDB', grade: 20, pass: false },
      { cname: 'nodejs', grade: 80, pass: true }
    ],
    isFired: false
  },
  {
    _id: ObjectId('686e52996abcb3d174718dcd'),
    fname: 'Aly',
    lname: 'Nasser',
    age: 18,
    faculty: { name: 'Science', address: 'Cairo' },
    grades: [
      { cname: 'Physics', grade: 85, pass: true },
      { cname: 'Math', grade: 65, pass: true }
    ],
    isFired: false
  }
]

```



```

]
FacultySystemDB> db.student.find({isFired:true})
[
  {
    _id: ObjectId('686e51ea6abcb3d174718dcc'),
    fname: 'Nada',
    lname: 'Yasser',
    age: 21,
    faculty: { name: 'CS', address: 'BeniSuef' },
    grades: [
      { cname: 'dataStructure', grade: 10, pass: false },
      { cname: 'C++', grade: 8, pass: false }
    ],
    isFired: true
  }
]
FacultySystemDB> db.student.find({$and:[{age:{$lte:21}}, {faculty:{$ne:NULL}}]})
ReferenceError: NULL is not defined
FacultySystemDB> db.student.find({$and:[{age:{$lte:21}}, {faculty:{$ne:"NULL"}}]})
[
  {
    _id: ObjectId('686e51ea6abcb3d174718dcb'),
    fname: 'Aya',
    lname: 'Ahmed',
    age: 20,
    faculty: { name: 'BSU', address: 'BeniSuef' },
    grades: [
      { cname: 'mongoDB', grade: 20, pass: false },
      { cname: 'nodejs', grade: 80, pass: true }
    ],
    isFired: false
  },
  {
    _id: ObjectId('686e51ea6abcb3d174718dcc'),
    fname: 'Nada',
    lname: 'Yasser',
    age: 21,
    faculty: { name: 'CS', address: 'BeniSuef' },
    grades: [

```

```

    { cname: 'dataStructure', grade: 10, pass: false },
    { cname: 'C++', grade: 8, pass: false }
  ],
  isFired: true
},
{
  _id: ObjectId('686e52996abcb3d174718dcd'),
  fname: 'Aly',
  lname: 'Nasser',
  age: 18,
  faculty: { name: 'Science', address: 'Cairo' },
  grades: [
    { cname: 'Physics', grade: 85, pass: true },
    { cname: 'Math', grade: 65, pass: true }
  ],
  isFired: false
}
]
FacultySystemDB> db.student.find({$and:[{age:{$gte:21}}, {faculty:{$ne:"NULL"}}]})
[
  {
    _id: ObjectId('686e50c96abcb3d174718dca'),
    fname: 'Ahmed',
    lname: 'Toba',
    age: 22,
    faculty: { name: 'BSU', address: 'BeniSuef' },
    grades: [
      { cname: 'mongoDB', grade: 100, pass: true },
      { cname: 'nodejs', grade: 45, pass: true }
    ],
    isFired: false
  },
  {
    _id: ObjectId('686e51ea6abcb3d174718dcc'),
    fname: 'Nada',
    lname: 'Yasser',
    age: 21,
    faculty: { name: 'CS', address: 'BeniSuef' },
    grades: [

```

```

    { cname: 'dataStructure', grade: 10, pass: false },
    { cname: 'C++', grade: 8, pass: false }
  ],
  isFired: true
}
]
FacultySystemDB> db.student.find({fname:"Nada"}, {fname:1, lname:1, isFired:1, _id:0})
[ { fname: 'Nada', lname: 'Yasser', isFired: true } ]
FacultySystemDB> db.student.updateOne({fname:"Nada"}, {lname:"Ibrahim"})
MongoInvalidArgumentError: Update document requires atomic operators
FacultySystemDB> db.student.updateOne({fname:"Nada"}, {$set:{lname:"Ibrahim"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
FacultySystemDB> db.student.find({fname:"Nada"})
[
  {
    _id: ObjectId('686e51ea6abcb3d174718dce'),
    fname: 'Nada',
    lname: 'Ibrahim',
    age: 21,
    faculty: { name: 'CS', address: 'BeniSuef' },
    grades: [
      { cname: 'dataStructure', grade: 10, pass: false },
      { cname: 'C++', grade: 8, pass: false }
    ],
    isFired: true
  }
]
FacultySystemDB> db.student.deleteOne({isFired:true})
{ acknowledged: true, deletedCount: 1 }
FacultySystemDB> db.student.find()
[
  {
    _id: ObjectId('686e50c96abcb3d174718dca'),

```

```

    fname: 'Ahmed',
    lname: 'Toba',
    age: 22,
    faculty: { name: 'BSU', address: 'BeniSuef' },
    grades: [
      { cname: 'mongoDB', grade: 100, pass: true },
      { cname: 'nodejs', grade: 45, pass: true }
    ],
    isFired: false
  },
  {
    _id: ObjectId('686e51ea6abcb3d174718dcb'),
    fname: 'Aya',
    lname: 'Ahmed',
    age: 20,
    faculty: { name: 'BSU', address: 'BeniSuef' },
    grades: [
      { cname: 'mongoDB', grade: 20, pass: false },
      { cname: 'nodejs', grade: 80, pass: true }
    ],
    isFired: false
  },
  {
    _id: ObjectId('686e52996abcb3d174718dcd'),
    fname: 'Aly',
    lname: 'Nasser',
    age: 18,
    faculty: { name: 'Science', address: 'Cairo' },
    grades: [
      { cname: 'Physics', grade: 85, pass: true },
      { cname: 'Math', grade: 65, pass: true }
    ],
    isFired: false
  }
]
FacultySystemDB> db.student.updateMany({}, {$set:{lname:"Tamer"}})
TypeError: db.student.updateMany is not a function
FacultySystemDB> db.student.updateMany({}, {$set:{lname:"Tamer"}})
{

```

```

    isFired: false
  }
]
FacultySystemDB> dp.student.drop()
ReferenceError: dp is not defined
FacultySystemDB> db.student.drop()
true
FacultySystemDB> db.dropDatabase()
{ ok: 1, dropped: 'FacultySystemDB' }
FacultySystemDB> show dbs
NTI-TRAINING  72.00 KiB
NTI-TRANING   8.00 KiB
admin         40.00 KiB
config        108.00 KiB
local         40.00 KiB
natours-test  40.00 KiB
FacultySystemDB> use FacultySystemV2
switched to db FacultySystemV2
FacultySystemV2> db.createCollection("student")
{ ok: 1 }
FacultySystemV2> db.createCollection("faculty")
{ ok: 1 }
FacultySystemV2> db.createCollection("course")
{ ok: 1 }
FacultySystemV2> db.student.insertOne({fname:"Ahmed", lname:"Toba", isFired:false, facultyID: 101, [{courseID:504, greade:74}, {courseID: 404, grade:82}]})
Uncaught:
SyntaxError: Unexpected token, expected "]" (1:108)

> 1 | db.student.insertOne({fname:"Ahmed", lname:"Toba", isFired:false, facultyID: 101, [{courseID:504, greade:74}, {courseID: 404, grade:82}]})
    |
    2 |

FacultySystemV2> db.student.insertOne({fname:"Ahmed", lname:"Toba", isFired:false, facultyID: 101, courses:[{courseID:504, greade:74}, {courseID: 404, grade:82}]})
{
  acknowledged: true,
  insertedId: ObjectId('686e59e26abcb3d174718dce')
}
FacultySystemV2> db.student.find()

```



```

[
  {
    _id: ObjectId('686e59e26abcb3d174718dce'),
    fname: 'Ahmed',
    lname: 'Toba',
    isFired: false,
    facultyID: 101,
    courses: [ { courseID: 504, greade: 74 }, { courseID: 404, grade: 82 } ]
  }
]
FacultySystemV2> db.faculty.insertOne({fname:"Engineering", address:"BeniSuef"})
{
  acknowledged: true,
  insertedId: ObjectId('686e5a316abcb3d174718dcf')
}
FacultySystemV2> db.faculty.find()
[
  {
    _id: ObjectId('686e5a316abcb3d174718dcf'),
    fname: 'Engineering',
    address: 'BeniSuef'
  }
]
FacultySystemV2> db.course.insertOne({cname:"Math", fmark: 89})
{
  acknowledged: true,
  insertedId: ObjectId('686e5a706abcb3d174718dd0')
}
FacultySystemV2> db.course.find()
[
  {
    _id: ObjectId('686e5a706abcb3d174718dd0'),
    cname: 'Math',
    fmark: 89
  }
]
FacultySystemV2> |

```