

Database & Collection Setup

Insert Operations

```
mongosh mongodb://127.0.0.1:27017/
Microsoft Windows [Version 10.0.26100.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SNEKA>mongosh
Current Mongosh Log ID: 693d75f84c44350a711e2620
Connecting to:  mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.10
Using MongoDB:  8.2.2
Using Mongosh:  2.5.10

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

-----
The server generated these startup warnings when booting
2025-12-11T13:05:26.609+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

test> use studentDB
switched to db studentDB
studentDB> db.createCollection("students")
{ ok: 1 }
studentDB> db.student.insertOne({name:"Rahul",age:23,course:"MERN Stack",status:"active"})
{ acknowledged: true, insertedId: ObjectId('693d76294c44350a711e2621') }
studentDB> db.student.insertMany([
  {name:"Anita",age:21,course:"Python",status:"active"},
  {name:"Vikram",age:23,course:"MERN Stack",status:"inactive"},
  {name:"sneha",age:23,status:"active"}
])
{ acknowledged: true, insertedIds: { '0': ObjectId('693d778e4c44350a711e2622'), '1': ObjectId('693d778e4c44350a711e2623'), '2': ObjectId('693d778e4c44350a711e2624') } }
studentDB> |
```

Read Operations

```
studentDB> db.student.find({course:"MERN Stack"})
[
  {
    _id: ObjectId('693d744cf144fb34f71e2621'),
    name: 'Rahul',
    age: 23,
    course: 'MERN Stack',
    status: 'active'
  },
  {
    _id: ObjectId('693d76294c44350a711e2621'),
    name: 'Rahul',
    age: 23,
    course: 'MERN Stack',
    status: 'active'
  },
  {
    _id: ObjectId('693d778e4c44350a711e2623'),
    name: 'Vikram',
    age: 23,
    course: 'MERN Stack',
    status: 'inactive'
  }
]
studentDB> |
```

```

studentDB> db.student.find({status:"active"})
[
  {
    _id: ObjectId('693d744cf144fb34f71e2621'),
    name: 'Rahul',
    age: 23,
    course: 'MERN Stack',
    status: 'active'
  },
  {
    _id: ObjectId('693d76294c44350a711e2621'),
    name: 'Rahul',
    age: 23,
    course: 'MERN Stack',
    status: 'active'
  },
  {
    _id: ObjectId('693d778e4c44350a711e2622'),
    name: 'Anita',
    age: 21,
    course: 'Python',
    status: 'active'
  },
  {
    _id: ObjectId('693d778e4c44350a711e2624'),
    name: 'Sneha',
    age: 23,
    status: 'active'
  }
]
studentDB> |

```

Update Operations

```

mongosh mongodb://127.0.0.1
]
studentDB> db.student.updateOne({name:"Rahul"},{$set:{age:22}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
studentDB> db.student.updateOne({course:"MERN Stack"},{$set:{status:"active"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}
studentDB> db.student.updateMany({course:"MERN Stack"},{$set:{status:"active"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 1,
  upsertedCount: 0
}
studentDB> db.student.find({course:"MERN Stack"})
[
  {
    _id: ObjectId('693d744cf144fb34f71e2621'),
    name: 'Rahul',
    age: 22,
    course: 'MERN Stack',
    status: 'active'
  },
  {
    _id: ObjectId('693d76294c44350a711e2621'),
    name: 'Rahul',
    age: 23,
    course: 'MERN Stack',
    status: 'active'
  },
  {
    _id: ObjectId('693d778e4c44350a711e2622'),
    name: 'Anita',
    age: 21,
    course: 'Python',
    status: 'active'
  },
  {
    _id: ObjectId('693d778e4c44350a711e2624'),
    name: 'Sneha',
    age: 23,
    status: 'active'
  }
]

```

Delete Operations

```
mongosh mongodb://127.0.0.1 x + v
studentDB> db.student.deleteOne({ name: "Vikram" })
{ acknowledged: true, deletedCount: 1 }
studentDB> db.student.find({name: "Vikram"})

studentDB> db.student.deleteMany({name:"Rahul",name:"Rahul"})
{ acknowledged: true, deletedCount: 2 }
studentDB> db.student.find({name: "Rahul"})

studentDB> db.student.find({ age: { $gt: 21 } })
[
  {
    _id: ObjectId('693d778e4c44358a711e2624'),
    name: 'sneha',
    age: 23,
    status: 'active'
  }
]
studentDB> db.student.find({ age: { $lt: 22 } })
[
  {
    _id: ObjectId('693d778e4c44358a711e2622'),
    name: 'Anita',
    age: 21,
    course: 'Python',
    status: 'active'
  }
]
studentDB> db.student.find({ course: { $in: ["MERN Stack", "Python"] } })
[
  {
    _id: ObjectId('693d778e4c44358a711e2622'),
    name: 'Anita',
    age: 21,
    course: 'Python',
    status: 'active'
  }
]
```

Query Operators

```
studentDB> db.student.find({$and:[{ course: "Python" },{ status: "active" }]}))
[
  {
    _id: ObjectId('693d778e4c44358a711e2622'),
    name: 'Anita',
    age: 21,
    course: 'Python',
    status: 'active'
  }
]
studentDB> db.student.find({$or:[{ course: "Python" },{ age:{ $lt:21} }]}))
[
  {
    _id: ObjectId('693d778e4c44358a711e2622'),
    name: 'Anita',
    age: 21,
    course: 'Python',
    status: 'active'
  }
]
studentDB> db.student.find({ age: { $exists: true } })
[
  {
    _id: ObjectId('693d778e4c44358a711e2622'),
    name: 'Anita',
    age: 21,
    course: 'Python',
    status: 'active'
  },
  {
    _id: ObjectId('693d778e4c44358a711e2624'),
    name: 'sneha',
    age: 23,
    status: 'active'
  }
]
studentDB> |
```

Use Case: Library System

```
mongosh mongodb://127.0.0.1:27020
studentDB> use libraryDB
switched to db libraryDB
libraryDB> db.createCollection("books")
{ ok: 1 }
libraryDB> db.books.insertMany([
  { title: "MongoDB Basics", author: "Jhon Doe", price: 450, available: true },
  { title: "Node.js Guide", author: "Jane smith", price: 600, available: true },
  { title: "React", author: "Mark Lee", price: 550, available: true }
])
TypeError: db.books.insertMany is not a function
libraryDB> db.books.insertMany([
  { title: "MongoDB Basics", author: "Jhon Doe", price: 450, available: true },
  { title: "Node.js Guide", author: "Jane smith", price: 600, available: true },
  { title: "React", author: "Mark Lee", price: 550, available: true }
])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('693d80e54c44350a711e2625'),
    '1': ObjectId('693d80e54c44350a711e2626'),
    '2': ObjectId('693d80e54c44350a711e2627')
  }
}
libraryDB> db.books.updateOne({ title: "Node.js Guide" }, { $set: { available: true } })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
libraryDB> db.books.updateOne({ title: "Node.js Guide" }, { $set: { available: false } })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
libraryDB> db.books.find({ available: true })
[
  {
    _id: ObjectId('693d80e54c44350a711e2625'),
    title: 'MongoDB Basics',
    author: 'Jhon Doe',
    price: 450,
    available: true
  },
  {
    _id: ObjectId('693d80e54c44350a711e2627'),
    title: 'React',
    author: 'Mark Lee',
    price: 550,
    available: true
  }
]
libraryDB> db.books.deleteOne({ title: "MongoDB Basics" })
{ acknowledged: true, deletedCount: 1 }
```