

```

test> db.createCollection("students")
{ ok: 1 }

test>
db.students.insertMany([
  {name:"Raj",age:22,subject:["Math",
"Science"],gratuated:false},
{name:"Priya",age:24,subject:["Hi
story","Literature"],gratuated:true},
{name:"Anil",age:23,subject:["Math",
"Literature"],gratuated:false},
{name:"Sita",age:25,subject:["Math",
"Science"],gratuated:true}
])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("669f47ef61bbe242162b2c18"),
    '1': ObjectId("669f47ef61bbe242162b2c19"),
    '2': ObjectId("669f47ef61bbe242162b2c1a"),
    '3': ObjectId("669f47ef61bbe242162b2c1b")
  }
}

test> db.students.find()
[
  {
    _id: ObjectId("669f47ef61bbe242162b2c18"),
    name: 'Raj',
    age: 22,
    subject: [ 'Math', 'Science' ],
    graduated: false
  },
  {
    _id: ObjectId("669f47ef61bbe242162b2c19"),
    name: 'Priya',
    age: 24,
    subject: [ 'History', 'Literature' ],
    graduated: true
  },
  {
    _id: ObjectId("669f47ef61bbe242162b2c1a"),
    name: 'Anil',
    age: 23,
    subject: [ 'Math', 'Literature' ],
    graduated: false
  },
  {
    _id: ObjectId("669f47ef61bbe242162b2c1b"),
    name: 'Sita',
    age: 25,
    subject: [ 'Math', 'Science' ],
    graduated: true
  }
]

test> var binsert = db.students.initializeUnorderedBulkOp();

test>
binsert.insert({name:"Reema",age:21,subject:["Math","Science"],gratuated:false});
{ nInsertOps: 1, nUpdateOps: 0, nRemoveOps: 0, nBatches: 1 }
test> binsert.execute()
{
  acknowledged: true,
  insertedCount: 1,
  insertedIds: [ { index: 0, _id: ObjectId("669f494c61bbe242162b2c1c") } ],
  matchedCount: 0,
  modifiedCount: 0,
  deletedCount: 0,
  upsertedCount: 0,
  upsertedIds: []
}

```

```

test> db.students.find()
[
  {
    _id: ObjectId("669f47ef61bbe242162b2c18"),
    name: 'Raj',
    age: 22,
    subject: [ 'Math', 'Science' ],
    graduated: false
  },
  {
    _id: ObjectId("669f47ef61bbe242162b2c19"),
    name: 'Priya',
    age: 24,
    subject: [ 'History', 'Literature' ],
    graduated: true
  },
  {
    _id: ObjectId("669f47ef61bbe242162b2c1a"),
    name: 'Anil',
    age: 23,
    subject: [ 'Math', 'Literature' ],
    graduated: false
  },
  {
    _id: ObjectId("669f47ef61bbe242162b2c1b"),
    name: 'Sita',
    age: 25,
    subject: [ 'Math', 'Science' ],
    graduated: true
  },
  {
    _id: ObjectId("669f494c61bbe242162b2c1c"),
    name: 'Reema',
    age: 21,
    subject: [ 'Math', 'Science' ],
    graduated: false
  }
]
test> db.runCommand( { collMod: "users",
...   validator: {
...     $jsonSchema: {
...       bsonType: "object",
...       required: [ "username", "password" ],
...       properties: {
...         username: {
...           bsonType: "string",
...           description: "must be a string and is required"
...         },
...         password: {
...           bsonType: "string",
...           minLength: 12,
...           description: "must be a string of at least 12
characters, and is required"
...         }
...       }
...     }
...   }
... }

test> db.runCommand({collMod:"students",validator:{$jsonSchema:
{bsonType:"object",required:["name","age","subject","gratu
ated"],properties:{name:{bsonType:"string"},age:{bsonType:"i
nt"},subject:{bsonType:"object"},gratuated:{bsonType:"bool"}}
}}})
{ ok: 1 }
test> db.students.deleteOne({name:"raj"})
{ acknowledged: true, deletedCount: 0 }
test> db.students.deleteOne({name:"Raj"})
{ acknowledged: true, deletedCount: 1 }
test> db.students.find()
[

```

```

{
  _id: ObjectId("669f47ef61bbe242162b2c19"),
  name: 'Priya',
  age: 24,
  subject: [ 'History', 'Literature' ],
  graduated: true
},
{
  _id: ObjectId("669f47ef61bbe242162b2c1a"),
  name: 'Anil',
  age: 23,
  subject: [ 'Math', 'Literature' ],
  graduated: false
},
{
  _id: ObjectId("669f47ef61bbe242162b2c1b"),
  name: 'Sita',
  age: 25,
  subject: [ 'Math', 'Science' ],
  graduated: true
},
{
  _id: ObjectId("669f494c61bbe242162b2c1c"),
  name: 'Reema',
  age: 21,
  subject: [ 'Math', 'Science' ],
  graduated: false
}
]
test> db.students.deleteOne({graduation:false})
{ acknowledged: true, deletedCount: 0 }
test> db.students.deleteOne({graduated:false})
{ acknowledged: true, deletedCount: 1 }
test> db.students.find()
[
  {
    _id: ObjectId("669f47ef61bbe242162b2c19"),
    name: 'Priya',
    age: 24,
    subject: [ 'History', 'Literature' ],
    graduated: true
  },
  {
    _id: ObjectId("669f47ef61bbe242162b2c1b"),
    name: 'Sita',
    age: 25,
    subject: [ 'Math', 'Science' ],
    graduated: true
  },
  {
    _id: ObjectId("669f494c61bbe242162b2c1c"),
    name: 'Reema',
    age: 21,
    subject: [ 'Math', 'Science' ],
    graduated: false
  }
]

test>
db.runCommand({collMod:"students",validator:{$jsonSchema:{bsonType:"object",required:["name","age","subject","graduated"],properties:{name:{bsonType:"string"},age:{bsonType:"int"},graduated:{bsonType:"bool"}}}}})
{ ok: 1 }
test>
db.students.updateOne({name:"Priya"},{$set:{subject:["History","Philosophy"]}})
{
  acknowledged: true,
  insertedId: null,

```

```

  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
test> db.students.find()
[
  {
    _id: ObjectId("669f47ef61bbe242162b2c19"),
    name: 'Priya',
    age: 24,
    subject: [ 'History', 'Philosophy' ],
    graduated: true
  },
  {
    _id: ObjectId("669f47ef61bbe242162b2c1b"),
    name: 'Sita',
    age: 25,
    subject: [ 'Math', 'Science' ],
    graduated: true
  },
  {
    _id: ObjectId("669f494c61bbe242162b2c1c"),
    name: 'Reema',
    age: 21,
    subject: [ 'Math', 'Science' ],
    graduated: false
  }
]
test> db.students.updateMany({},{$set:{hasMaths:false}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0
}
test>
db.students.updateMany({subject:{$in:["Math"]}},{$set:{hasMaths:true}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
test> db.students.find()
[
  {
    _id: ObjectId("669f47ef61bbe242162b2c19"),
    name: 'Priya',
    age: 24,
    subject: [ 'History', 'Philosophy' ],
    graduated: true,
    hasMaths: false
  },
  {
    _id: ObjectId("669f47ef61bbe242162b2c1b"),
    name: 'Sita',
    age: 25,
    subject: [ 'Math', 'Science' ],
    graduated: true,
    hasMaths: true
  },
  {
    _id: ObjectId("669f494c61bbe242162b2c1c"),
    name: 'Reema',
    age: 21,
    subject: [ 'Math', 'Science' ],
    graduated: false,

```

```

    hasMaths: true
  }
]
test> db.students.updateOne({name:"Sita"},{$set:{gratuated:false}})
MongoInvalidArgumentError: Update document requires
atomic operators
test>
db.students.updateOne({name:"Sita"},{$set:{gratuated:false}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
test> db.students.find()
[
  {
    _id: ObjectId("669f47ef61bbe242162b2c19"),
    name: 'Priya',
    age: 24,
    subject: [ 'History', 'Philosophy' ],
    graduated: true,
    hasMaths: false
  },
  {
    _id: ObjectId("669f47ef61bbe242162b2c1b"),
    name: 'Sita',
    age: 25,
    subject: [ 'Math', 'Science' ],
    graduated: false,
    hasMaths: true
  },
  {
    _id: ObjectId("669f494c61bbe242162b2c1c"),
    name: 'Reema',
    age: 21,
    subject: [ 'Math', 'Science' ],
    graduated: false,
    hasMaths: true
  }
]
test>
est> db.students.find()
[
  {
    _id: ObjectId('66a9cc19cf1de96595149f49'),
    name: 'Priya',
    age: 24,
    subject: [ 'History', 'Literature' ],
    graduated: true
  },
  {
    _id: ObjectId('66a9cc19cf1de96595149f4b'),
    name: 'Sita',
    age: 25,
    subject: [ 'Math', 'Science' ],
    graduated: true
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4c'),
    name: 'Raj',
    age: 22,
    subject: [ 'Math', 'Science' ],
    graduated: false
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4d'),
    name: 'Ravi',

```

```

    age: 25,
    subject: [ 'Literature', 'Science' ],
    graduated: true
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4e'),
    name: 'Sohama',
    age: 20,
    subject: [ 'Math', 'Science' ],
    graduated: false
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4f'),
    name: 'Roshan',
    age: 23,
    subject: [ 'AI', 'Science' ],
    graduated: false
  }
]
test> db.collections.find()
test> db.students.find()
[
  {
    _id: ObjectId('66a9cc19cf1de96595149f49'),
    name: 'Priya',
    age: 24,
    subject: [ 'History', 'Literature' ],
    graduated: true
  },
  {
    _id: ObjectId('66a9cc19cf1de96595149f4b'),
    name: 'Sita',
    age: 25,
    subject: [ 'Math', 'Science' ],
    graduated: true
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4c'),
    name: 'Raj',
    age: 22,
    subject: [ 'Math', 'Science' ],
    graduated: false
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4d'),
    name: 'Ravi',
    age: 25,
    subject: [ 'Literature', 'Science' ],
    graduated: true
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4e'),
    name: 'Sohama',
    age: 20,
    subject: [ 'Math', 'Science' ],
    graduated: false
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4f'),
    name: 'Roshan',
    age: 23,
    subject: [ 'AI', 'Science' ],
    graduated: false
  }
]
test> db.students.find({name:"Raj"})
[
  {
    _id: ObjectId('66a9d741c63be471b5149f4c'),

```

```

        name: 'Raj',
        age: 22,
        subject: [ 'Math', 'Science' ],
        graduated: false
    }
]
test> db.students.find({name:"Anil"})

test> db.students.find({$gt:{age:22}})
test> db.students.find({age:{$gt:22}})
[
  {
    _id: ObjectId('66a9cc19cf1de96595149f49'),
    name: 'Priya',
    age: 24,
    subject: [ 'History', 'Literature' ],
    graduated: true
  },
  {
    _id: ObjectId('66a9cc19cf1de96595149f4b'),
    name: 'Sita',
    age: 25,
    subject: [ 'Math', 'Science' ],
    graduated: true
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4d'),
    name: 'Ravi',
    age: 25,
    subject: [ 'Literature', 'Science' ],
    graduated: true
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4f'),
    name: 'Roshan',
    age: 23,
    subject: [ 'AI', 'Science' ],
    graduated: false
  }
]
test> db.students.find({$or:[{name:"Sita"},{age:{$gt:23}}]})
[
  {
    _id: ObjectId('66a9cc19cf1de96595149f49'),
    name: 'Priya',
    age: 24,
    subject: [ 'History', 'Literature' ],
    graduated: true
  },
  {
    _id: ObjectId('66a9cc19cf1de96595149f4b'),
    name: 'Sita',
    age: 25,
    subject: [ 'Math', 'Science' ],
    graduated: true
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4d'),
    name: 'Ravi',
    age: 25,
    subject: [ 'Literature', 'Science' ],
    graduated: true
  }
]
test> db.students.find({$or:[{name:"Raj"},{age:{$gt:23}}]})
[
  {
    _id: ObjectId('66a9cc19cf1de96595149f49'),
    name: 'Priya',
    age: 24,

```

```

        subject: [ 'History', 'Literature' ],
        graduated: true
    },
    {
      _id: ObjectId('66a9cc19cf1de96595149f4b'),
      name: 'Sita',
      age: 25,
      subject: [ 'Math', 'Science' ],
      graduated: true
    },
    {
      _id: ObjectId('66a9d741c63be471b5149f4c'),
      name: 'Raj',
      age: 22,
      subject: [ 'Math', 'Science' ],
      graduated: false
    },
    {
      _id: ObjectId('66a9d741c63be471b5149f4d'),
      name: 'Ravi',
      age: 25,
      subject: [ 'Literature', 'Science' ],
      graduated: true
    }
  ]
test> db.students.find({age:{$lte:23}})
[
  {
    _id: ObjectId('66a9d741c63be471b5149f4c'),
    name: 'Raj',
    age: 22,
    subject: [ 'Math', 'Science' ],
    graduated: false
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4e'),
    name: 'Sohama',
    age: 20,
    subject: [ 'Math', 'Science' ],
    graduated: false
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4f'),
    name: 'Roshan',
    age: 23,
    subject: [ 'AI', 'Science' ],
    graduated: false
  }
]
test> db.students.find({age:{$ne:23}})
[
  {
    _id: ObjectId('66a9cc19cf1de96595149f49'),
    name: 'Priya',
    age: 24,
    subject: [ 'History', 'Literature' ],
    graduated: true
  },
  {
    _id: ObjectId('66a9cc19cf1de96595149f4b'),
    name: 'Sita',
    age: 25,
    subject: [ 'Math', 'Science' ],
    graduated: true
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4c'),
    name: 'Raj',

```

```

    age: 22,
    subject: [ 'Math', 'Science' ],
    graduated: false
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4d'),
    name: 'Ravi',
    age: 25,
    subject: [ 'Literature', 'Science' ],
    graduated: true
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4e'),
    name: 'Sohama',
    age: 20,
    subject: [ 'Math', 'Science' ],
    graduated: false
  }
]
test> db.students.find({subject:{$in:["Science"]}})
MongoServerError[BadValue]: $in needs an array
test> db.students.find({subject:{$in:["Science"]}})
[
  {
    _id: ObjectId('66a9cc19cf1de96595149f4b'),
    name: 'Sita',
    age: 25,
    subject: [ 'Math', 'Science' ],
    graduated: true
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4c'),
    name: 'Raj',
    age: 22,
    subject: [ 'Math', 'Science' ],
    graduated: false
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4d'),
    name: 'Ravi',
    age: 25,
    subject: [ 'Literature', 'Science' ],
    graduated: true
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4e'),
    name: 'Sohama',
    age: 20,
    subject: [ 'Math', 'Science' ],
    graduated: false
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4f'),
    name: 'Roshan',
    age: 23,
    subject: [ 'AI', 'Science' ],
    graduated: false
  }
]
test> db.students.find({$ne:{$nickname:""}})
MongoServerError[BadValue]: unknown top level operator:
$ne. If you have a field name that starts with a '$' symbol,
consider using $getField or $setField.
test> db.students.find({nickname:{$exist:false}})
MongoServerError[BadValue]: unknown operator: $exist
test> db.students.find({nickname:{$exists:false}})
[
  {
    _id: ObjectId('66a9cc19cf1de96595149f49'),
    name: 'Priya',

```

```

    age: 24,
    subject: [ 'History', 'Literature' ],
    graduated: true
  },
  {
    _id: ObjectId('66a9cc19cf1de96595149f4b'),
    name: 'Sita',
    age: 25,
    subject: [ 'Math', 'Science' ],
    graduated: true
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4c'),
    name: 'Raj',
    age: 22,
    subject: [ 'Math', 'Science' ],
    graduated: false
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4d'),
    name: 'Ravi',
    age: 25,
    subject: [ 'Literature', 'Science' ],
    graduated: true
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4e'),
    name: 'Sohama',
    age: 20,
    subject: [ 'Math', 'Science' ],
    graduated: false
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4f'),
    name: 'Roshan',
    age: 23,
    subject: [ 'AI', 'Science' ],
    graduated: false
  }
]
test> db.students.find({nickname:{$exists:true}})

test> db.students.find({names:{$regex:.a}})
Uncaught:
SyntaxError: Unexpected token (1:32)

> 1 | db.students.find({names:{$regex:.a}})
    |                               ^
    2 |

test> db.students.find({names:{$regex:".a"}})

test> db.students.find({names:{$regex:". "}})

test> db.students.find({names:{$regex:"*a"}})
MongoServerError[Location51091]: Regular expression is
invalid: quantifier does not follow a repeatable item
test> db.students.find({names:{$regex:"^a"}})

test> db.students.find({names:{$regex:"/*a/g"}})

test> db.students.find({names:{$regex:"/*n/g"}})

test>
db.students.find({age:{$gt:21}}).limit(2).skip(1).sort({name:1})
[
  {
    _id: ObjectId('66a9d741c63be471b5149f4c'),
    name: 'Raj',
    age: 22,

```

```

      subject: [ 'Math', 'Science' ],
      graduated: false
    },
    {
      _id: ObjectId('66a9d741c63be471b5149f4d'),
      name: 'Ravi',
      age: 25,
      subject: [ 'Literature', 'Science' ],
      graduated: true
    }
  ]
}
test>
db.students.find({age:{$gt:21}}).limit(2).skip(1).sort({name:-1})
)
[
  {
    _id: ObjectId('66a9d741c63be471b5149f4d'),
    name: 'Roshan',
    age: 23,
    subject: [ 'AI', 'Science' ],
    graduated: false
  },
  {
    _id: ObjectId('66a9d741c63be471b5149f4d'),
    name: 'Ravi',
    age: 25,
    subject: [ 'Literature', 'Science' ],
    graduated: true
  }
]
test>
(To exit, press Ctrl+C again or Ctrl+D or type .exit)
test>
dit-23@dit23-OptiPlex-5070:~/Desktop$

```