

Sort and Limit Documents in MongoDB:

```
> mongosh
> show dbs
> use school
> db.students.find()
```

```
[
  {
    _id: ObjectId('69804ab8864a660af4628ca0'),
    name: 'spongebob',
    age: 30,
    gpa: 3.2
  },
  {
    _id: ObjectId('698075a0c41c5afdcf628ca0'),
    name: 'patrick',
    age: 38,
    gpa: 1.5
  },
  {
    _id: ObjectId('698075a0c41c5afdcf628ca1'),
    name: 'sandy',
    age: 27,
    gpa: 4
  },
  {
    _id: ObjectId('698075a0c41c5afdcf628ca2'),
    name: 'gary',
    age: 18,
    gpa: 2.5
  },
  {
    _id: ObjectId('69807998aa017ce650cd8097'),
    name: 'mongo',
    age: 32,
    gpa: 3.2
  },
  {
    _id: ObjectId('69807a96aa017ce650cd8099'),
    name: 'Name1',
    age: 12,
    gpa: 1.2
  },
]
```

```
{
  _id: ObjectId('69807a96aa017ce650cd809a'),
  name: 'Name2',
  age: 22,
  gpa: 2.4
},
{
  _id: ObjectId('69807a96aa017ce650cd809b'),
  name: 'Name3',
  age: 42,
  gpa: 6.2
},
{
  _id: ObjectId('69808144c41c5afdcf628ca4'),
  name: 'Larry',
  age: 32,
  gpa: 3.2,
  fullTime: false,
  registerDate: ISODate('2026-02-02T10:49:40.654Z'),
  graduationDate: null,
  courses: [ 'Biology', 'Chemistry', 'Calculus' ],
  address: { street: '123 Fake St.', city: 'Bik Bottom', zip: 12345 }
}
]
```

To sort these documents, some sort of order:

```
> db.students.find().sort({name:1})
```

Note: name: 1 <- for alphabetical order

name: -1 <- reverse alphabetical order

```
[
  {
    _id: ObjectId('69808144c41c5afdcf628ca4'),
    name: 'Larry',
    age: 32,
    gpa: 3.2,
    fullTime: false,
    registerDate: ISODate('2026-02-02T10:49:40.654Z'),
    graduationDate: null,
    courses: [ 'Biology', 'Chemistry', 'Calculus' ],
    address: { street: '123 Fake St.', city: 'Bik Bottom', zip: 12345 }
  },
  {
    _id: ObjectId('69807a96aa017ce650cd8099'),
    name: 'Name1',
    age: 12,
```

```
    gpa: 1.2
  },
  {
    _id: ObjectId('69807a96aa017ce650cd809a'),
    name: 'Name2',
    age: 22,
    gpa: 2.4
  },
  {
    _id: ObjectId('69807a96aa017ce650cd809b'),
    name: 'Name3',
    age: 42,
    gpa: 6.2
  },
  {
    _id: ObjectId('698075a0c41c5afdcf628ca2'),
    name: 'gary',
    age: 18,
    gpa: 2.5
  },
  {
    _id: ObjectId('69807998aa017ce650cd8097'),
    name: 'mongo',
    age: 32,
    gpa: 3.2
  },
  {
    _id: ObjectId('698075a0c41c5afdcf628ca0'),
    name: 'patrick',
    age: 38,
    gpa: 1.5
  },
  {
    _id: ObjectId('698075a0c41c5afdcf628ca1'),
    name: 'sandy',
    age: 27,
    gpa: 4
  },
  {
    _id: ObjectId('69804ab8864a660af4628ca0'),
    name: 'spongebob',
    age: 30,
    gpa: 3.2
  }
}
```

```
]
> db.students.find().sort({name:-1})
[
  {
    _id: ObjectId('69804ab8864a660af4628ca0'),
    name: 'spongebob',
    age: 30,
    gpa: 3.2
  },
  {
    _id: ObjectId('698075a0c41c5afdcf628ca1'),
    name: 'sandy',
    age: 27,
    gpa: 4
  },
  {
    _id: ObjectId('698075a0c41c5afdcf628ca0'),
    name: 'patrick',
    age: 38,
    gpa: 1.5
  },
  {
    _id: ObjectId('69807998aa017ce650cd8097'),
    name: 'mongo',
    age: 32,
    gpa: 3.2
  },
  {
    _id: ObjectId('698075a0c41c5afdcf628ca2'),
    name: 'gary',
    age: 18,
    gpa: 2.5
  },
  {
    _id: ObjectId('69807a96aa017ce650cd809b'),
    name: 'Name3',
    age: 42,
    gpa: 6.2
  },
  {
    _id: ObjectId('69807a96aa017ce650cd809a'),
    name: 'Name2',
    age: 22,
    gpa: 2.4
  }
]
```

```

    },
    {
      _id: ObjectId('69807a96aa017ce650cd8099'),
      name: 'Name1',
      age: 12,
      gpa: 1.2
    },
    {
      _id: ObjectId('69808144c41c5afdcf628ca4'),
      name: 'Larry',
      age: 32,
      gpa: 3.2,
      fullTime: false,
      registerDate: ISODate('2026-02-02T10:49:40.654Z'),
      graduationDate: null,
      courses: [ 'Biology', 'Chemistry', 'Calculus' ],
      address: { street: '123 Fake St.', city: 'Bik Bottom', zip: 12345 }
    }
  ]

```

Similarly, we can sort by gpa:

```

> db.students.find().sort({gpa:1}) <- for Ascending order
> db.students.find().sort({gpa:-1}) <- reverse order

```

To limit a Document:

To get one document:

```

> db.students.find().limit(1)
[
  {
    _id: ObjectId('69804ab8864a660af4628ca0'),
    name: 'spongebob',
    age: 30,
    gpa: 3.2
  }
]

```

To get three document:

```

> db.students.find().limit(3)
[
  {
    _id: ObjectId('69804ab8864a660af4628ca0'),
    name: 'spongebob',
    age: 30,
    gpa: 3.2
  },

```

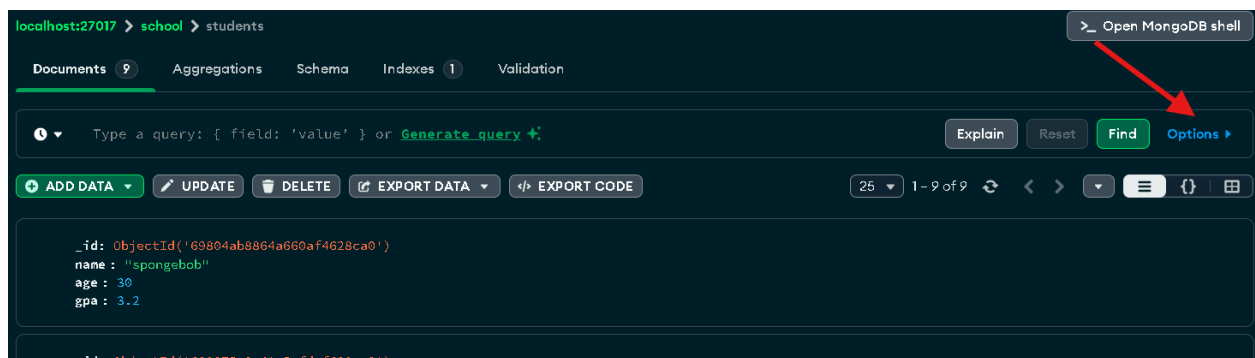
```
{
  _id: ObjectId('698075a0c41c5afdcf628ca0'),
  name: 'patrick',
  age: 38,
  gpa: 1.5
},
{
  _id: ObjectId('698075a0c41c5afdcf628ca1'),
  name: 'sandy',
  age: 27,
  gpa: 4
}
]
```

To get one student with highest gpa:

school> **db.students.find().sort({gpa:-1}).limit(1)**

```
[
  {
    _id: ObjectId('69807a96aa017ce650cd809b'),
    name: 'Name3',
    age: 42,
    gpa: 6.2
  }
]
```

To Sort by MongoDB Compass:



{name: 1}:

The screenshot shows the MongoDB Compass interface. At the top, there's a query editor with the following fields:

- Project: { field: 0 }
- Sort: {name:1} (highlighted with a red box)
- Collation: { locale: 'simple' }
- Index Hint: { field: -1 } or "indexName"

Buttons at the top right include Explain, Reset, and Find (highlighted with a red arrow). Below these are input fields for Max Time MS (60000), Skip (0), and Limit (0). A tooltip titled "Introducing insights" is visible, with a "See insights in action" button.

Below the query editor, there are buttons for ADD DATA, UPDATE, DELETE, EXPORT DATA, and EXPORT CODE. The results section shows three documents:

```
{
  "_id": ObjectId('69808144c41c5afdcf628ca4'),
  "name": "Larry",
  "age": 32,
  "gpa": 3.2,
  "fullTime": false,
  "registerDate": "2026-02-02T10:49:40.654+00:00",
  "graduationDate": null,
  "courses": Array (3),
  "address": Object
}
```

```
{
  "_id": ObjectId('69807a96aa017ce650cd8099'),
  "name": "Name1",
  "age": 12,
  "gpa": 1.2
}
```

```
{
  "_id": ObjectId('69807a96aa017ce650cd809a'),
  "name": "Name2"
}
```

To change the limit:

This close-up screenshot focuses on the top right of the MongoDB Compass interface. It shows the Explain, Reset, Find (highlighted with a red arrow), and Options buttons. Below these are input fields for Max Time MS (60000), Skip (0), and Limit (1, highlighted with a red box).