

Below there are some queries done in mongoDB shell

```
> use education
< switched to db education
education> // Insert a single student document

db.students.insertOne({
  "regno": "2021IT001",
  "name": "Kolins",
  "age": 25,
  "degree": "IT",
  "gpa": 3.2,
  "gender": "Male",
  "skills": [
    "Java",
    "MatLap",
    "MySQL"
  ]
});
```

```
// Insert multiple student documents
db.students.insertMany([
  {
    "regno": "2021IT004",
    "name": "Kamal",
    "age": 25,
    "degree": "CSC",
    "gpa": 3.2,
    "gender": "Male",
    "skills": [
      "Java",
      "MatLap",
      "MySQL"
    ]
  },
  {
    "regno": "2021IT005",
    "name": "Mathavi",
    "age": 25,
    "degree": "IT",
    "gpa": 3.2,
    "gender": "Female",
    "skills": [
      "Java",
      "MatLap",
      "python"
    ]
  }
]);
```

```

// Find all female students
db.students.find({ "gender": "Female" });

/*
Find all students whose age is greater than 25
$gt = greater than
*/
db.students.find({ age: { $gt: 25 } });

/*
Find all students who have either "Java" or "MatLab" as a skill
$in = matches any value in the given array
*/
db.students.find({ skills: { $in: ["Java", "MatLab"] } });

/*
Find all IT students and sort them by GPA in ascending order (lowest to highest)
1 = ascending order
*/
db.students.find({ 'degree': 'IT' }).sort({ 'gpa': 1 });

/*
Find female students in IT degree and sort by GPA in ascending order
*/
db.students.find({ 'degree': 'IT', 'gender': 'Female' }).sort({ 'gpa': 1 });

```

```

/*
Find female students in IT degree and sort by GPA in ascending order
*/
db.students.find({ 'degree': 'IT', 'gender': 'Female' }).sort({ 'gpa': 1 });

/*
Sort all students by GPA in descending order and by name in ascending order
-1 = descending order, 1 = ascending order
*/
db.students.find().sort({ gpa: -1, name: 1 });
|

```

Let see how to do this in MongoDB compass

This is the way how we use find option

Documents 0 Aggregations Schema Indexes 1 Validation

🔍 {"gender": "Female"} [Generate query](#) [Explain](#) [Reset](#) [Find](#) [Options](#)

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#) [INSIGHT](#) 25 1 - 1 of 1

```
{
  "_id": ObjectId('680d103b133bbc3264fe26ed'),
  "regno": "2021IT005",
  "name": "Mathavi",
  "age": 25,
  "degree": "IT",
  "gpa": 3.2,
  "gender": "Female",
  "skills": Array(3)
}
```

Next we can see more options

localhost:27017 > mongosh: localhost:27017 students local +

localhost:27017 > education > students [Open MongoDB shell](#)

Documents 0 Aggregations Schema Indexes 1 Validation

🔍 {'degree': 'IT'} [Generate query](#) [Explain](#) [Reset](#) [Find](#) [Options](#)

Project { field: 0 }

Sort {'gpa': 1} Max Time MS 60000

Collation { locale: 'simple' } Skip 0 Limit 0

Index Hint { field: -1 }

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#) [INSIGHT](#) 25 1 - 2 of 2

students						
	_id ObjectId	regno String	name String	age Int32	degree String	gpa Doubl
1	ObjectId('680d103b133bbc3...	"2021IT001"	"Kolins"	25	"IT"	3.2
2	ObjectId('680d103b133bbc3...	"2021IT005"	"Mathavi"	25	"IT"	3.2