Basic MongoDB Commands

show databases

Show All Databases-Displays all existing databases.

use unidb

Switch to a Database-If unidb exists, it switches to it. If not, MongoDB creates it when you insert data

Show Collections

Lists all collections (like tables) in the current database.

MongoDB Compass Commands (GUI)

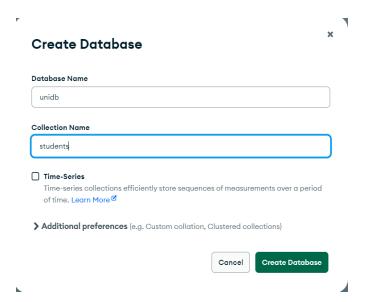
Command	Meaning
{ age: { "\$gt": 25 } }	Finds people older than 25.
{ age: { "\$It": 25 } }	Finds people younger than 25.
{ age: 25 }	Finds people whose age is 25.
{ name: 1 }	Shows only name (with _id).
{ name: 1, _id: 0 }	Shows name, hides _id field.
{ skills: { "\$in": ["NoSql"] } }	Finds if skills include "NoSql".
{ skills: { "\$in": ["NoSql", "JS"] } }	Finds if skills have either value.
{ skills: { "\$nin": ["NoSql"] } }	Finds if skills do not include "NoSql".
{ gender: "female" }	Finds documents where gender is female.
{ age: 1 }	Sorts by age ascending.
{ age: -1 }	Sorts by age descending.

MongoDB Console (mongosh) Commands

These are typed in the terminal or shell.

Command	Meaning
db.students.insertMany([])	Adds multiple documents.
db.students.find()	Shows all documents.
db.students.findOne()	Shows one document.
db.students.find({ regno: "2021ict126" })	Finds document where regno matches.
db.students.find({}, { name: 1 })	Show only the name field.
db. students .find({}, { name: 1, _id: 0 })	Show name only, hide _id.
db. students. find().sort({ age: 1 })	Sorts documents by age ascending.
db.students.find().sort({ age: -1 })	Sorts documents by age descending.
db.students.find({ age: { "\$gt": 25 } })	Finds people with age greater than 25.
db.students.find({ age: { "\$It": 25 } })	Finds people with age less than 25.
db.students.find({ age: 25 })	Finds people with age equal to 25.
db.students.find({ skills: { "\$in": ["NoSql"] } })	Finds those with "NoSql" in skills.
db.students.find({ skills: { "\$nin": ["NoSql"] } })	Finds those without "NoSql" skill.
db.students.find({ degree: "CSC", gender: "male" }).sort({ age: -1 })	Find CSC male students, sort by age descending.

Create a database and collection



Show databases

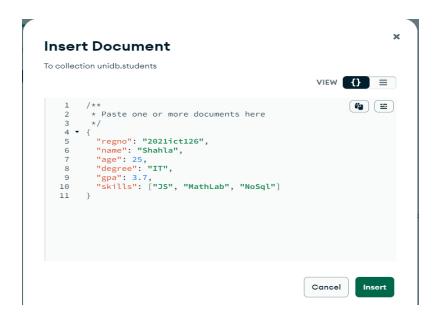
use unidb

```
> use unidb
< switched to db unidb
unidb>
```

show collections

```
> show collections
< students
unidb>
```

Insert document-Using Compass



Insert document-Using Console -----> "insertOne"

```
> db.students.insertOne(
{
    "regno": "2021ict001",
    "name": "Bandara",
    "age": 26,
    "degree": "CSC",
    "gpa": 3.1,
    "skills": [
        "35",
        "MathLab",
        "NoSql"
    ],
        "gender": "male"

}
}
```

Insert document-Using Console -----> "insertMany"

```
>_MONGOSH
> db.students.insertMany(
 "regno": "2021ict002",
   "name": "Shashi",
   "age": 23,
  "degree": "CSC",
   "gpa": 2.8,
    "MathLab",
   "gender": "female"
 "regno": "2021ict003",
  "name": "Raween",
   "age": 20,
   "degree": "CSC",
   "gpa": 3.9,
   "skills": [
    "JS",
    "MathLab",
    "NoSql"
   "gender": "male"
```

Display all documents - Click the button find

```
localhost:27017 > unidb > students
                                                                                                                                     >_ Open MongoDB shell
  Documents (1) Aggregations Schema Indexes (1) Validation
                                                                                                              Explain Reset Find ← Options ▼
  Type a query: { field: 'value' } or Generate query ★;
  Project { field: 0 }
                                                                                                                    Max Time MS 60000
             { field: -1 } or [['field', -1]]
  Collation { locale: 'simple' }
                                                                                           Skip 0
                                                                                                                          Limit 0
  Index Hint { field: -1 }
O ADD DATA 

© EXPORT DATA 

UPDATE

DELETE
                                                                                                       _id: ObjectId('680bbbacb2334dba91ceaa39')
       regno: "2021ict126"
name: "Shahla"
age: 25
degree: "IT"
      gpa: 3.7
▶ skills: Array (3)
     _id: ObjectId('680bbd754le98ef3f45b1203')
regno: "262lict801"
name: "Bandara"
age: 26
degree: "CSC"
gpa: 3.1
> skills: Array (3)
gender: "male"
       _id: ObjectId('680bbdd641e98ef3f45b1204')
regno: "2021ict002"
name: "Shashi"
```

```
_id: ObjectId('680bbdd641e98ef3f45b1204')
regno: "2021ict002"
name: "Shashi"
age: 23
degree: "CSC"
gpa: 2.8

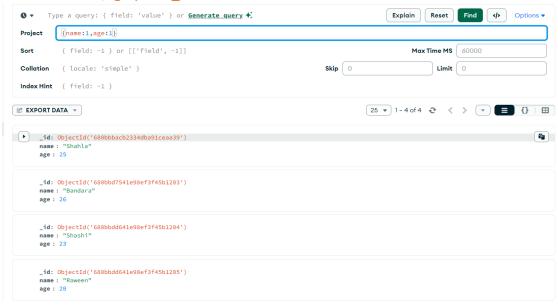
> skills: Array (2)
gender: "female"

_id: ObjectId('680bbdd641e98ef3f45b1205')
regno: "2021ict003"
name: "Raween"
age: 20
degree: "CSC"
gpa: 3.9

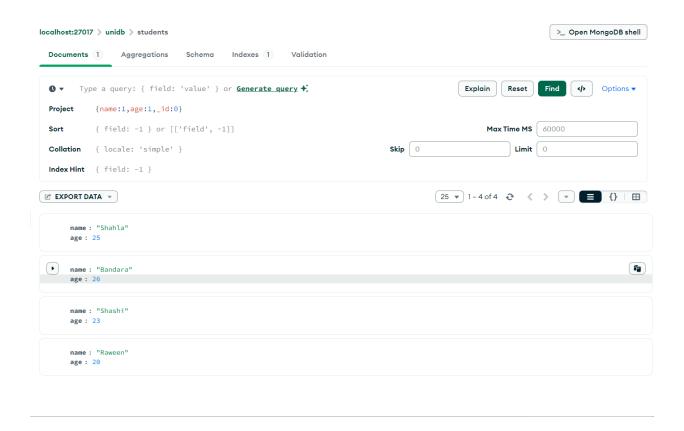
> skills: Array (3)
gender: "male"
```

Retrieve specific fields only

1.Shows name,age (with _id

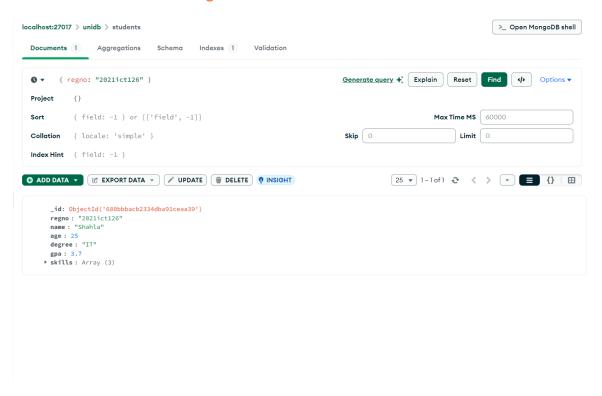


2.Shows name,age hides _id field.

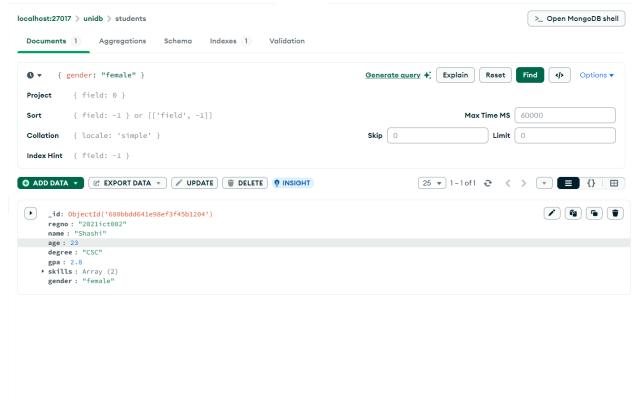


Retrieve documents using specific value

1. Shows documents which regno "2021ict126".



2. Shows documents those are female.



Retrieve documents using specific value-----> using console

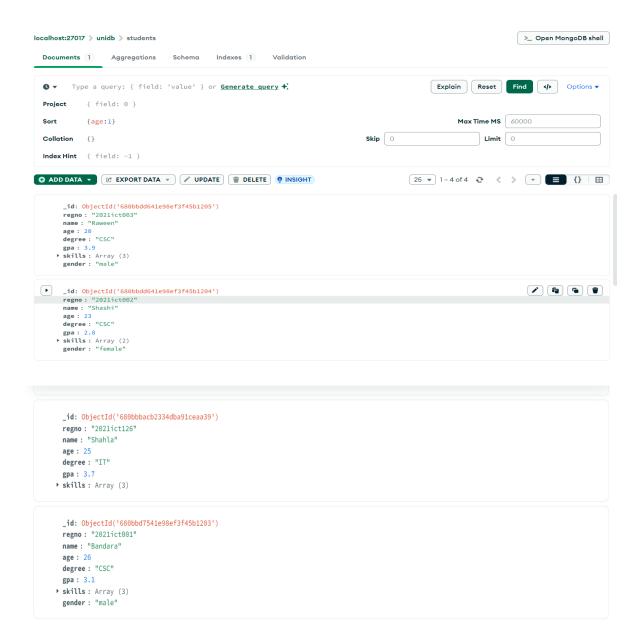
1. Shows documents those are male

2.Shows documents those are male-----> "findOne"

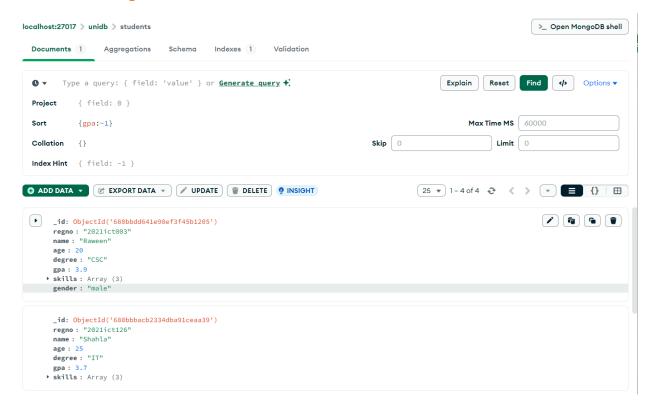
```
> db.students.findOne({"gender":"male"})

{
    _id: ObjectId('680bbd7541e98ef3f45b1203'),
    regno: '2021ict001',
    name: 'Bandara',
    age: 26,
    degree: 'CSC',
    gpa: 3.1,
    skills: [
        'JS',
        'MathLab',
        'NoSql'
    ],
    gender: 'male'
}
```

Sort in ascending Order

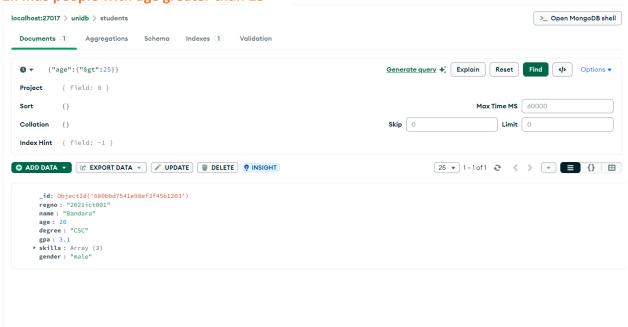


Sort in descending order

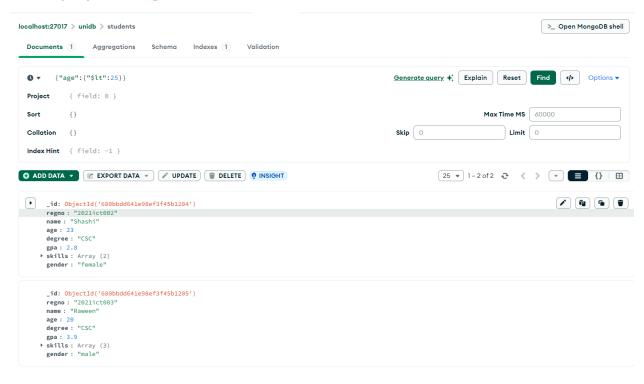


Find the documents with comparision

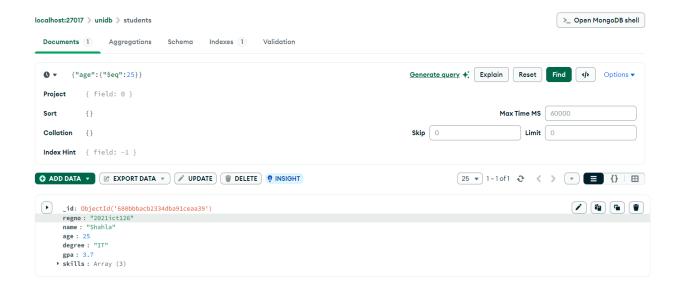
1. Finds people with age greater than 25



2. Finds people with age less than 25.

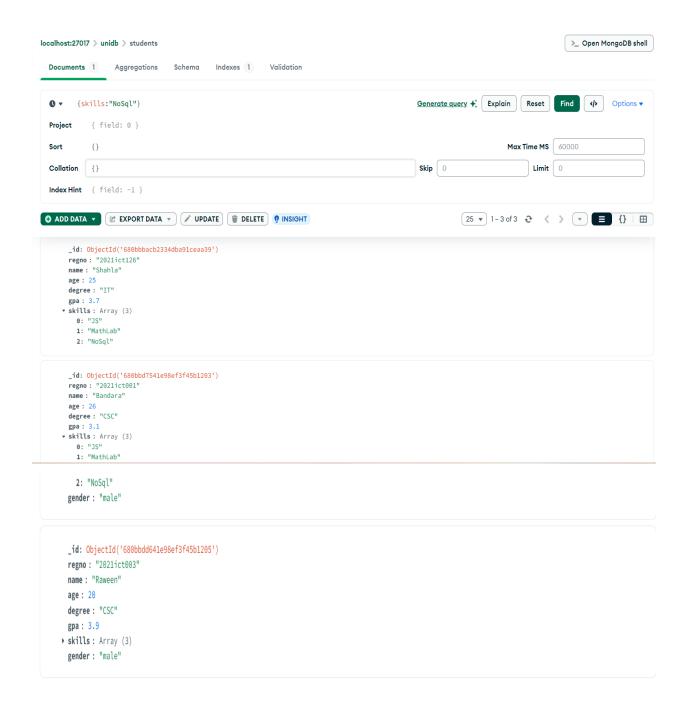


3. Finds people with age equal to 25.

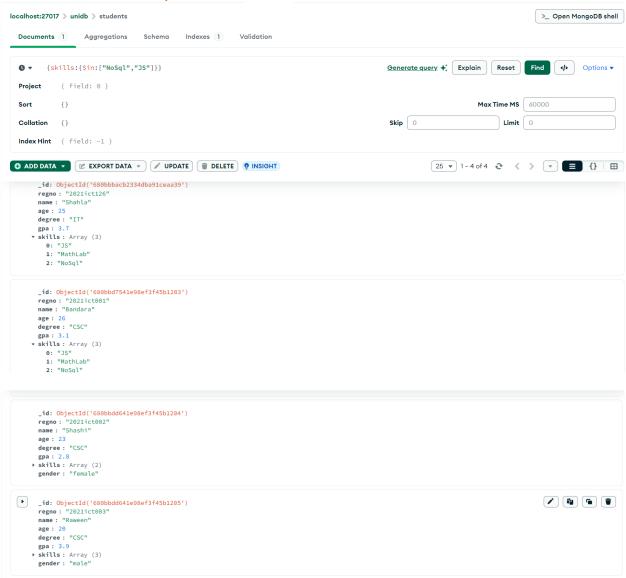


Find the documents using \$in,\$nin

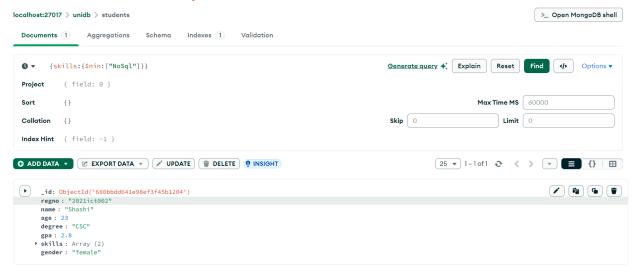
1.Finds those with "NoSql" skills.



2.Finds those with "NoSql" OR JS skills.



3. Finds those without "NoSql" skill.



Sort in descending Order ---->using console

```
{
    _id: ObjectId('680bbdr54le98ef3f45b1203'),
    regmo: '2021ct001',
    name: 'Bandara',
    age: 26,
    degree: 'CSC',
    gmai 3.1,
    skillo: [
        '35',
        'Mathlab',
        'Nosql'
],
    gender: 'male'
}
{
    _id: ObjectId('680bbdd64le98ef3f45b1204'),
    regmo: '2021ct062',
    name: 'Shashi',
    age: 23,
    degree: 'CSC',
    gmai 2.8,
    skillo: [
        '35',
        'Mathlab'
],
    gender: 'female'
}
```

Find who follow csc dgree, sort their gpa in ascending order

Find CSC male students, sort their age in descending.

Find the details with comparision---->using console

```
> db.students.find({"age":{"$gt":25}})

< {
    _id: ObjectId('680bbd7541e98ef3f45b1203'),
    regno: '2021ct001',
    name: 'Bandara',
    age: 26,
    degree: 'CSC',
    gpa: 3.1,
    skills: [
        'JS',
        'MathLab',
        'NoSql'
    ],
    gender: 'male'
}</pre>
```