

MongoDB

Installation

Link download:

https://fastdl.mongodb.org/windows/mongodb-windows-x86_64-6.0.1-signed.msi

To check properly install MongoDB

After download open go to c driver and
programfile-->mongodb-->version-->bin copy bin path

Method 1

```
C:\Users\Shubham>"C:\Program Files\MongoDB\Server\6.0\bin\mongod.exe" --version
db version v6.0.1
Build Info: {
  "version": "6.0.1",
  "gitVersion": "32f0f9c88dc44a2c8073a5bd47cf779d4bfdee6b",
  "modules": [],
  "allocator": "tcmalloc",
  "environment": {
    "distmod": "windows",
    "distarch": "x86_64",
    "target_arch": "x86_64"
  }
}
```

Second download tools boost productivity

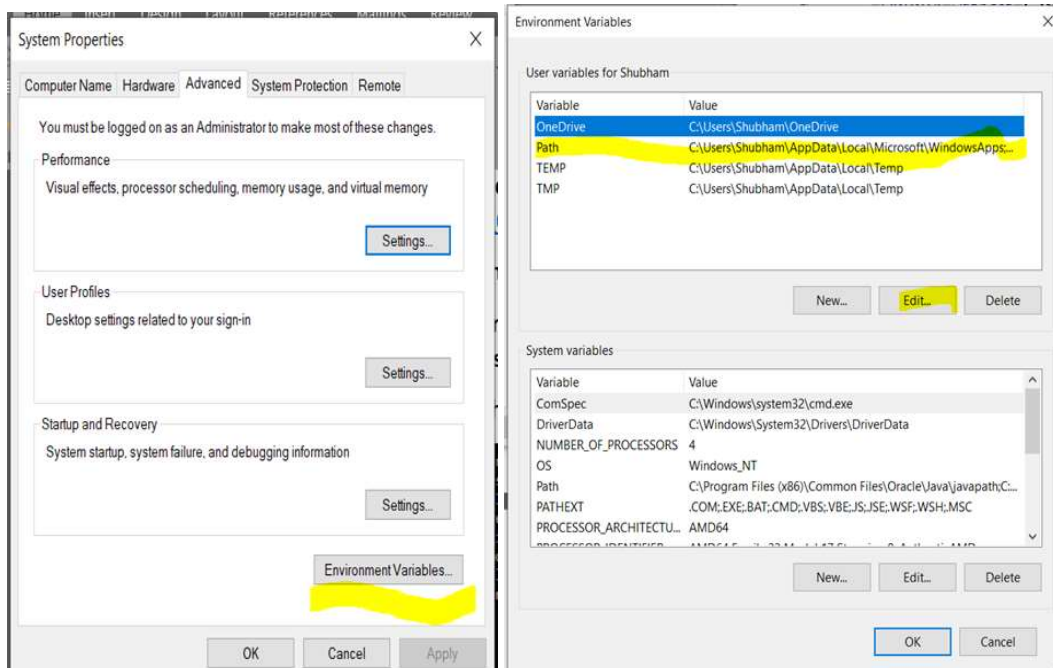
Link:

<https://downloads.mongodb.com/compass/mongosh-1.5.4-win32-x64.zip>

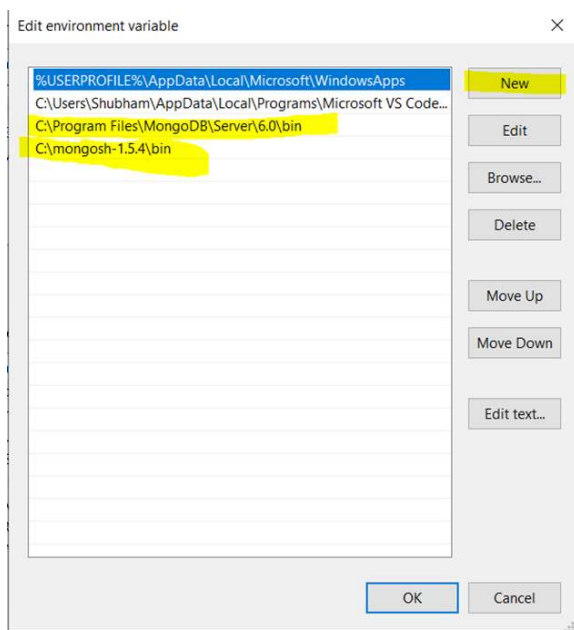
Extract the file and copy the file past it local c driver

Direct use mongo dbs

1)to click window button and search "edit the system environment variable"



Click à edit



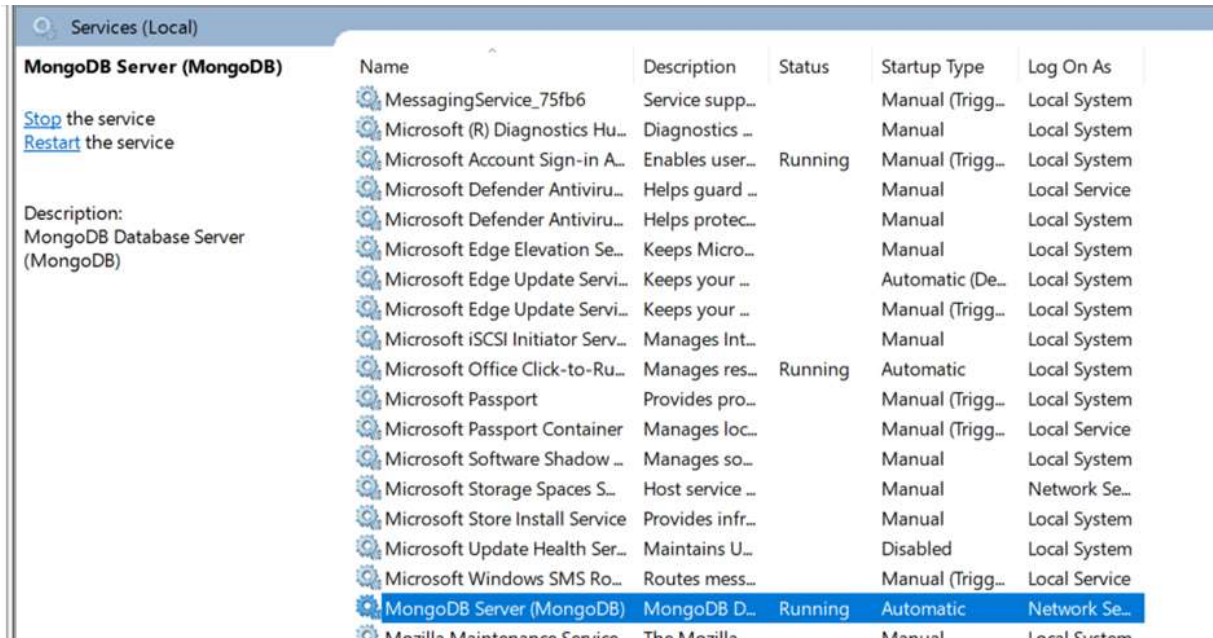
Go the local c driveràprogramàserveràversionàbinàcopy the path

Go the local c driveràmongosh-1.5.4àbinàcopy path

Past it ss

Click ok ok ok

Its always run the mongod server to check click the window and search services its always run



Complete the install part

Create Collection part

Create data base

Use anyname eg ss

```
test> use cscorner
switched to db cscorner
cscorner> db
cscorner
```

To show database

```
cscorner> show dbs
admin    40.00 KiB
config  60.00 KiB
local    40.00 KiB
```

To create the collection

“db.createCollection(“collection name”)”

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

Insert collection

“Db.collection name.insertOne({“rollno”:9231,”name”:”shubham”})”

```
cscorner> db.students.insertOne({"rollno":9231,"name":"shubham"})
{
  acknowledged: true,
  insertedId: ObjectId("6305bccc152e052821b5bfd7")
}
```

Show collection

“Show collection”

```
cscorner> show collections
students
cscorner> db.students.find().pretty()
```

To view collection

“Db.collection name .find().pretty()”

```
cscorner> db.students.find().pretty()
[
  {
    _id: ObjectId("6305bccc152e052821b5bfd7"),
    rollno: 9231,
    name: 'shubham'
  }
]
```

To delete and drop the collection

“Db.colection name.drop()”

“Db.dropDatabase”

```
cscorner> db.students.drop()
true
```

```
cscorner> db.dropDatabase()
{ ok: 1, dropped: 'cscorner' }
```

```
cscorner> show dbs
admin      40.00 KiB
config    108.00 KiB
local      40.00 KiB
cscorner>
```

Complete create the collection and delete or drop the collection

Important syntax

1)how to databases

Show dbs

```
test> show dbs
admin    40.00 KiB
config  60.00 KiB
local   40.00 KiB
test>
```

2)how to create database

Use student

```
test> use student
switched to db student
student>
```

3)how to create collection

db.10thclass.insertOne({"roll":21,"name":"shubham"})

```
student> db.tenclass.insertOne({name:"shubham", "name":"shubham"})
{
  acknowledged: true,
  insertedId: ObjectId("630654191a0466f0161502fc")
}
student>
```

4)how to show collection

Show collections

db.tenclass.find().pretty()

```
student> db.tenclass.find().pretty()
[
  {
    _id: ObjectId("630654191a0466f0161502fc"), name: 'shubham' },
  {
    _id: ObjectId("630654ff1a0466f0161502fd"),
    name: 'shubham',
    mobile: 9082361867
  },
  {
    _id: ObjectId("630655301a0466f0161502fe"),
    name: 'shubham',
    mobile: 9082361867,
    fess: true
  }
]
student>
```

```
student> show collections
tenclass
student>
```

5)how to delete the database

“**Db.colection name.drop()**”

“**Db.dropDatabase**”

```
cscorner> db.students.drop()
true
```

```
cscorner> db.dropDatabase()
{ ok: 1, dropped: 'cscorner' }
```

```
cscorner> show dbs
admin      40.00 KiB
config    108.00 KiB
local      40.00 KiB
cscorner>
```

Create database
Create collection
Create document

❖ insertMany

Syntax “**db.collection name.insertMany([{}, {}, {}, {}])**”


```
college> db.student1.insertMany([{name:"shubham",Roll:21},{name:"suraj",Roll:22}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("63065bbe1a0466f0161502ff"),
    '1': ObjectId("63065bbe1a0466f016150300")
  }
}
college> show dbs
admin      40.00 KiB
college    40.00 KiB
config     108.00 KiB
local      40.00 KiB
college>
```

Read

db.collection.find(query,projection)

Query→ pass and find the particular data

Projection→you can see data or not with the use of projection

Question to solve

1. Find all results to collection.

```
colleges> db.students.insertMany([
  {name:"shubham",email:"shubham@gmail.com",mno:87867767,fees:true},
  {name:"suraj",email:"suraj@gmail.com",mno:9777985,fees:true},
  {name:"aftab",email:"aftab@gmail.com",mno:78865868,fees:true},
  {name:"kuldeep",email:"kuldeep@gmail.com",mno:787885868,fees:true}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("630704ae63256fbc59f0fe68"),
    '1': ObjectId("630704ae63256fbc59f0fe69"),
    '2': ObjectId("630704ae63256fbc59f0fe6a"),
    '3': ObjectId("630704ae63256fbc59f0fe6b")
  }
}
colleges> db.students.find()
[
  {
    _id: ObjectId("630704ae63256fbc59f0fe68"),
    name: 'shubham',
    email: 'shubham@gmail.com',
    mno: 87867767,
    fees: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe69"),
    name: 'suraj',
    email: 'suraj@gmail.com',
    mno: 9777985,
    fees: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6a"),
    name: 'aftab',
    email: 'aftab@gmail.com',
    mno: 78865868,
    fees: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6b"),
    name: 'kuldeep',
    email: 'kuldeep@gmail.com',
    mno: 787885868,
    fees: true
  }
]
```

2. Show result in pretty format


```
colleges> db.students.find().pretty()
[
  {
    _id: ObjectId("630704ae63256fbc59f0fe68"),
    name: 'shubham',
    email: 'shubham@gmail.com',
    mno: 87867767,
    fees: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe69"),
    name: 'suraj',
    email: 'suraj@gmail.com',
    mno: 9777985,
    fess: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6a"),
    name: 'aftab',
    email: 'aftab@gmail.com',
    mno: 78865868,
    fees: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6b"),
    name: 'kuldeep',
    email: 'kuldeep@gmail.com',
    mno: 787885868,
    fees: true
  }
]
colleges>
```

3. Show particular document.

Query use

Only show one name particular data

Syntax

db.students.find({name:"shubham"}).pretty()

```
colleges> db.students.find({name:"shubham"}).pretty()
[
  {
    _id: ObjectId("630704ae63256fbc59f0fe68"),
    name: 'shubham',
    email: 'shubham@gmail.com',
    mno: 87867767,
    fees: true
  }
]
colleges>
```

But you see all the field like name,email,mno,fess

4. Show particular field in one document.

Projection

Only show one name particular data and one field

Syntax

db.students.find({name:"shubham"},{name:1}).pretty()

But projection 0 than false and 1 than true

You write 0 they do not print but you write 1 the only print when you written 1

```
colleges> db.students.find({name:"shubham"},{name:1}).pretty()
[ { _id: ObjectId("630704ae63256fbc59f0fe68"), name: 'shubham' } ]
colleges>
```

This ss only print name filed because you written 1

```
colleges> db.students.find({name:"shubham"},{name:0}).pretty()
[
  {
    _id: ObjectId("630704ae63256fbc59f0fe68"),
    email: 'shubham@gmail.com',
    mno: 87867767,
    fees: true
  }
]
colleges>
```

Note : id is default every where print

5. Show particular filed in one document without ID.

You not see the id then write **_id:0**

```
db.students.find({name:"shubham"},{_id:0,name:1}).pretty()
```

```
colleges> db.students.find({name:"shubham"},{name:1,_id:0}).pretty()
[ { name: 'shubham' } ]
colleges>
```

6. Set the query "fees:true" and get the one document.

Limit

Can se limit document but print all the feld

```
colleges> db.students.find({fees:true}).pretty()
[
  {
    _id: ObjectId("630704ae63256fbc59f0fe68"),
    name: 'shubham',
    email: 'shubham@gmail.com',
    mno: 87867767,
    fees: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6a"),
    name: 'aftab',
    email: 'aftab@gmail.com',
    mno: 78865868,
    fees: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6b"),
    name: 'kuldeep',
    email: 'kuldeep@gmail.com',
    mno: 787885868,
    fees: true
  }
]
```

Limit syntax

```
db.students.fint({fees:true}).pretty().limit(2)
```

Only 2 limit print

```
colleges> db.students.find({fees:true}).pretty().limit(2)
[
  {
    _id: ObjectId("630704ae63256fbc59f0fe68"),
    name: 'shubham',
    email: 'shubham@gmail.com',
    mno: 87867767,
    fees: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6a"),
    name: 'aftab',
    email: 'aftab@gmail.com',
    mno: 78865868,
    fees: true
  }
]
```

7. Please use another method for 6 question and get the result.
Not use limit find one field

```
colleges> db.students.findOne({fees:true}).pretty()
TypeError: db.students.f ... true}).pretty is not a function
colleges>
```

Note :not use pretty because its not a function pretty is show collection is good format

```
colleges> db.students.findOne({fees:true})
{
  _id: ObjectId("630704ae63256fbc59f0fe68"),
  name: 'shubham',
  email: 'shubham@gmail.com',
  mno: 87867767,
  fees: true
}
```

8. Apply on 6 question and skip 1 document and get next result

Skip

Skip the document

synatx

db.student.find().pretty().skip(1)

```

colleges> db.students.find({fees:true}).pretty().limit(1).skip(1)
[
  {
    _id: ObjectId("630704ae63256fbc59f0fe6a"),
    name: 'aftab',
    email: 'aftab@gmail.com',
    mno: 78865868,
    fees: true
  }
]
colleges> db.students.find().pretty().skip(1)
[
  {
    _id: ObjectId("630704ae63256fbc59f0fe69"),
    name: 'suraj',
    email: 'suraj@gmail.com',
    mno: 9777985,
    fess: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6a"),
    name: 'aftab',
    email: 'aftab@gmail.com',
    mno: 78865868,
    fees: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6b"),
    name: 'kuldeep',
    email: 'kuldeep@gmail.com',
    mno: 787885868,
    fees: true
  }
]
colleges>

```

For eg first document skip and find only one document

Syntax

db.student.find({fess:true}).pretty().limit(1).skip(1)

```
colleges> db.students.find({fees:true}).pretty().limit(1).skip(1)
[
  {
    _id: ObjectId("630704ae63256fbc59f0fe6a"),
    name: 'aftab',
    email: 'aftab@gmail.com',
    mno: 78865868,
    fees: true
  }
]
colleges>
```

Update

updateOne()=>**db.collection_name.updateOne(<filter>,<update>)**

updateMany()=>**db.collection_name.updateMany(<filter>,<update>)**

Question

1. how to update one particular filed in one document

Only one document change the number

Syntax:

db.students.updateOne({name:"shubham"},{\$set{mno:9082361867}})


```
colleges> db.students.find().pretty()
[
  {
    _id: ObjectId("630704ae63256fbc59f0fe68"),
    name: 'shubham',
    email: 'shubham@gmail.com',
    mno: 9082361867,
    fees: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe69"),
    name: 'suraj',
    email: 'suraj@gmail.com',
    mno: 9777985,
    fess: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6a"),
    name: 'aftab',
    email: 'aftab@gmail.com',
    mno: 78865868,
    fees: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6b"),
    name: 'kuldeep',
    email: 'kuldeep@gmail.com',
    mno: 787885868,
    fees: true
  }
]
colleges> db.students.updateOne({name:"shubham"},{$set:{mno:9082361867}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}
colleges>
```

Shubham mobile number change


```
colleges> db.students.find().pretty()
[
  {
    _id: ObjectId("630704ae63256fbc59f0fe68"),
    name: 'shubham',
    email: 'shubham@gmail.com',
    mno: 9082361867,
    fees: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe69"),
    name: 'suraj',
    email: 'suraj@gmail.com',
    mno: 9777985,
    fess: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6a"),
    name: 'aftab',
    email: 'aftab@gmail.com',
    mno: 78865868,
    fees: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6b"),
    name: 'kuldeep',
    email: 'kuldeep@gmail.com',
    mno: 787885868,
    fees: true
  }
]
colleges>
```

2. How to update one particular field in two or more documents.
Syntax:

db.students.updateOne({name:"shubham"},{\$set{mno:9082361867}})

Many

db.students.updateMany({mno:9082361867},{\$set{mno:false}})

Shubham and suraj number same

```
]
colleges> db.students.updateOne({name:"suraj"},{$set:{mno:9082361867}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
colleges> db.students.find().pretty()
[
  {
    _id: ObjectId("630704ae63256fbc59f0fe68"),
    name: 'shubham',
    email: 'shubham@gmail.com',
    mno: 9082361867,
    fees: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe69"),
    name: 'suraj',
    email: 'suraj@gmail.com',
    mno: 9082361867,
    fess: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6a"),
    name: 'aftab',
    email: 'aftab@gmail.com',
    mno: 78865868,
    fees: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6b"),
    name: 'kuldeep',
    email: 'kuldeep@gmail.com',
    mno: 787885868,
    fees: true
  }
]
```

```
colleges> db.students.updateMany({mno:9082361867},{ $set:{fess:false}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 1,
  upsertedCount: 0
}
colleges> db.students.find().pretty()
[
  {
    _id: ObjectId("630704ae63256fbc59f0fe68"),
    name: 'shubham',
    email: 'shubham@gmail.com',
    mno: 9082361867,
    fees: true,
    fess: false
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe69"),
    name: 'suraj',
    email: 'suraj@gmail.com',
    mno: 9082361867,
    fess: false
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6a"),
    name: 'aftab',
    email: 'aftab@gmail.com',
    mno: 78865868,
    fees: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6b"),
    name: 'kuldeep',
    email: 'kuldeep@gmail.com',
    mno: 787885868,
    fees: true
  }
]
colleges>
```

3. How to update more than one field in one document.
 One document to field change by using updateOne
db.students.updateOne({name:"suraj"},{\$set{mno:9082361867,fess:true}})

```

colleges> db.students.updateOne({name:"suraj"},{$set:{fess:true,mno:7886778}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
colleges> db.students.find().pretty()
[
  {
    _id: ObjectId("630704ae63256fbc59f0fe68"),
    name: 'shubham',
    email: 'shubham@gmail.com',
    mno: 9082361867,
    fees: true,
    fess: false
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe69"),
    name: 'suraj',
    email: 'suraj@gmail.com',
    mno: 7886778,
    fess: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6a"),
    name: 'aftab',
    email: 'aftab@gmail.com',
    mno: 78865868,
    fees: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6b"),
    name: 'kuldeep',
    email: 'kuldeep@gmail.com',
    mno: 787885868,
    fees: true
  }
]
colleges>

```

4. How to update more than one field in two or more document.

db.students.updateMany({fess:true},{ \$set{mno:9082361867,fess:false}})

```

colleges> db.students.updateMany({fess:true},{ $set:{fess:false,mno:7886778}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
colleges> db.students.find().pretty()
[
  {
    _id: ObjectId("630704ae63256fbc59f0fe68"),
    name: 'shubham',
    email: 'shubham@gmail.com',
    mno: 9082361867,
    fees: true,
    fess: false
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe69"),
    name: 'suraj',
    email: 'suraj@gmail.com',
    mno: 7886778,
    fess: false
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6a"),
    name: 'aftab',
    email: 'aftab@gmail.com',
    mno: 78865868,
    fees: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6b"),
    name: 'kuldeep',
    email: 'kuldeep@gmail.com',
    mno: 787885868,
    fees: true
  }
]
colleges>

```

Note :\$ set is compulsory

**db.students.updateMany({fess:true},{ \$set{mno:9082361867,f
ess:false}})**

Delete

deleteOne()--> db.collection_name.deleteOne(<filter>)
deleteMany()--> db.collection_name.deleteMany(<filter>)

Question

1. How to delete one document using filter

Delete one field to the document

DeleteOne

syntax

db.students.deleteOne({name:"aftab"})

```
colleges> db.students.find().pretty()
[
  {
    _id: ObjectId("630704ae63256fbc59f0fe68"),
    name: 'shubham',
    email: 'shubham@gmail.com',
    mno: 9082361867,
    fees: true,
    fess: false
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe69"),
    name: 'suraj',
    email: 'suraj@gmail.com',
    mno: 7886778,
    fess: false
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6a"),
    name: 'aftab',
    email: 'aftab@gmail.com',
    mno: 78865868,
    fees: true
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6b"),
    name: 'kuldeep',
    email: 'kuldeep@gmail.com',
    mno: 787885868,
    fees: true
  }
]
```

Delete one filed


```
colleges> db.students.deleteOne({name:"aftab"})
{ acknowledged: true, deletedCount: 1 }
colleges> db.students.find().pretty()
[
  {
    _id: ObjectId("630704ae63256fbc59f0fe68"),
    name: 'shubham',
    email: 'shubham@gmail.com',
    mno: 9082361867,
    fees: true,
    fess: false
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe69"),
    name: 'suraj',
    email: 'suraj@gmail.com',
    mno: 7886778,
    fess: false
  },
  {
    _id: ObjectId("630704ae63256fbc59f0fe6b"),
    name: 'kuldeep',
    email: 'kuldeep@gmail.com',
    mno: 787885868,
    fees: true
  }
]
colleges>
```

2. How to delete one document without using filter

Delete only one document not using filter

Syntax

db.students.deleteOne({})

```
colleges> db.students.find().pretty()
[
  {
    _id: ObjectId("630704ae63256fbc59f0fe6b"),
    name: 'kuldeep',
    email: 'kuldeep@gmail.com',
    mno: 787885868,
    fees: true
  },
  {
    _id: ObjectId("6307260e63256fbc59f0fe6c"),
    name: 'subham',
    email: 'shubham@gmail.com',
    mno: 67767637,
    fess: true
  },
  {
    _id: ObjectId("6307260e63256fbc59f0fe6d"),
    name: 'suraj',
    email: 'suraj@gmail.com',
    mno: 89847837,
    fess: true
  },
  {
    _id: ObjectId("6307263d63256fbc59f0fe6e"),
    name: 'aftab',
    email: 'aftab@gmail.com',
    mno: 677687
  }
]
```

Delete one document

```

]
colleges> db.students.deleteOne({})
{ acknowledged: true, deletedCount: 1 }
colleges> db.students.find().pretty()
[
  {
    _id: ObjectId("6307260e63256fbc59f0fe6c"),
    name: 'subham',
    email: 'shubham@gmail.com',
    mno: 67767637,
    fess: true
  },
  {
    _id: ObjectId("6307260e63256fbc59f0fe6d"),
    name: 'suraj',
    email: 'suraj@gmail.com',
    mno: 89847837,
    fess: true
  },
  {
    _id: ObjectId("6307263d63256fbc59f0fe6e"),
    name: 'aftab',
    email: 'aftab@gmail.com',
    mno: 677687
  }
]
colleges>

```

3. How to delete multiple document

Many document delete but one filed same both the document then its delete another one document is delete

deleteMany

Syntax

db.students.deleteMany({fess:true})

```
colleges> db.students.deleteMany({fess:true})
{ acknowledged: true, deletedCount: 2 }
colleges> db.students.find().pretty()
[
  {
    _id: ObjectId("6307263d63256fbc59f0fe6e"),
    name: 'aftab',
    email: 'aftab@gmail.com',
    mno: 677687
  }
]
colleges>
```

4. How to delete multiple document without using filter

All document delete use

Syntax:

db.deleteMan

Commands

1) View all database

Show dbs

2) Create a new or Switch database

Use dbName

3) view current database

Db

4) delete database

db.dropDatabase()

5) mongoDb Commands For collection

Show collections

6) Create a collection name "student"

db.createCollection("student")

7) delete and drop the collection

```
db.student.drop()
```

Mongodb commands for Rows

Show all rows in a collections

```
db.student.find().pretty()
```

Insert one rows

```
db.student.insert({name:"shubham",roll:21})
```

insertMany

```
db.student.insertMany([
    {name:"suraj",roll:21},
    {name:"shubham",roll:22},
    {name:"vikas",roll:23},
    {name:"karan",roll:24}
])
```

Search

```
db.student.find({name:"shubham"})
```

Search only one when one find then stop

```
db.student.findOne({name:"shubham"})
```

Limit

```
db.student.find().limit(1) //only one document show
```

Count document

```
db.student.find().pretty().count()
```

Which document as same name eg two document same name

```
db.student.find({name:"shubham"}).count()
```

Sort ascending order

```
db.student.find().sort({roll:1}).pretty()//always run eg  
Screenshot
```

```

college> db.student.find().sort({"roll":21}).pretty()
MongoInvalidArgumentError: Invalid sort direction: 21
college> db.student.find().sort({"roll":22}).pretty()
MongoInvalidArgumentError: Invalid sort direction: 22
college> db.student.find().sort({"roll":1}).pretty()
[
  {
    _id: ObjectId("630c7f71963cfd5ad9ffd14a"),
    name: 'suraj',
    roll: 21
  },
  {
    _id: ObjectId("630c7f71963cfd5ad9ffd14c"),
    name: 'shubham',
    roll: 22
  },
  {
    _id: ObjectId("630c7f71963cfd5ad9ffd14d"),
    name: 'vikas',
    roll: 23
  },
  {
    _id: ObjectId("630c7f71963cfd5ad9ffd14b"),
    name: 'karan',
    roll: 24
  }
]
college>

```

Update

db.students.updateOne({name:"suraj"},{\$set{mno:9082361867,fess:true}})

Increment update

db.student.update({name:"shubham"},{\$inc:{"roll":2}})
 //Shubhma roll no 22 and output is 24

Rename update

db.student.update({name:"shubham"},{\$rename:{"roll":"not declared"}})

Imp:only change filed not value

Remove

db.student.remove({name:"suraj"})

Less

```
db.student.find({"roll":{"lt:23}})
```

Greater than

```
db.student.find({"roll":{"gt:23}})
```