Microsoft Windows [Version 10.0.14393]

(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\MY PC>mongo

MongoDB shell version v5.0.5

connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb

Implicit session: session { "id" : UUID("b7d4a290-eca5-483e-8ad2-b50223242991") }

MongoDB server version: 5.0.5

================

Warning: the "mongo" shell has been superseded by "mongosh",

which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in

an upcoming release.

For installation instructions, see

https://docs.mongodb.com/mongodb-shell/install/

================

---

The server generated these startup warnings when booting:

2022-01-09T18:12:03.751+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

---

---

Enable MongoDB's free cloud-based monitoring service, which will then receive and display

metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you

and anyone you share the URL with. MongoDB may use this information to make product

improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()

To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

---

> show dbs

admin 0.000GB

config 0.000GB

local 0.000GB

> use student

switched to db student

> show dbs

admin 0.000GB

config 0.000GB

local 0.000GB

> db.studentcollection.insertOne({name:"sumeetjumde",email:"jumdesumeet@gmail.com",active:true,phno:4234242344})

{

"acknowledged" : true,

"insertedId" : ObjectId("61dad9cf0823a3a4267d9e88")

}

> show dbs

admin 0.000GB

config 0.000GB

local 0.000GB

student 0.000GB

> show collections

studentcollection

> db.studentcollection.find()

{ "\_id" : ObjectId("61dad9cf0823a3a4267d9e88"), "name" : "sumeetjumde", "email" : "jumdesumeet@gmail.com", "active" : true, "phno" : 4234242344 }

> db.studentcollection.find().pretty()

{

"\_id" : ObjectId("61dad9cf0823a3a4267d9e88"),

"name" : "sumeetjumde",

"email" : "jumdesumeet@gmail.com",

"active" : true,

"phno" : 4234242344

}

>



There are two ways to insert data in mongodb

MongoDB provides the following methods to insert documents into a collection:

* [db.collection.insertOne()](https://docs.mongodb.com/manual/reference/method/db.collection.insertOne/#mongodb-method-db.collection.insertOne) *New in version 3.2*
* [db.collection.insertMany()](https://docs.mongodb.com/manual/reference/method/db.collection.insertMany/#mongodb-method-db.collection.insertMany) *New in version 3.2*

> db

student

> db.studentcollection.insertMany([{name:"Anshul Jumde",email:"anshuljumde@gmail.com",active:true,phno:42342342},{name:"Pravin Jumde",email:"pravinjumde@gmail.com",active:true,phno:234234234},{name:"sangitajumde",email:"sangitajumde@gmail.com",active:true,phno:23542545432}])

{

"acknowledged" : true,

"insertedIds" : [

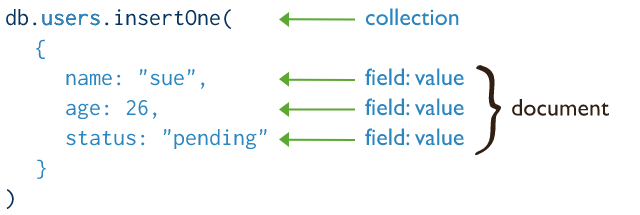
ObjectId("61dae1810823a3a4267d9e89"),

ObjectId("61dae1810823a3a4267d9e8a"),

ObjectId("61dae1810823a3a4267d9e8b")

]

}



> db.studentcollection.find()

{ "\_id" : ObjectId("61dad9cf0823a3a4267d9e88"), "name" : "sumeetjumde", "email" : "jumdesumeet@gmail.com", "active" : true, "phno" : 4234242344 }

{ "\_id" : ObjectId("61dae1810823a3a4267d9e89"), "name" : "Anshul Jumde", "email" : "anshuljumde@gmail.com", "active" : true, "phno" : 42342342 }

{ "\_id" : ObjectId("61dae1810823a3a4267d9e8a"), "name" : "Pravin Jumde", "email" : "pravinjumde@gmail.com", "active" : true, "phno" : 234234234 }

{ "\_id" : ObjectId("61dae1810823a3a4267d9e8b"), "name" : "sangitajumde", "email" : "sangitajumde@gmail.com", "active" : true, "phno" : 23542545432 }

> db.studentcollection.find().pretty()

{

"\_id" : ObjectId("61dad9cf0823a3a4267d9e88"),

"name" : "sumeetjumde",

"email" : "jumdesumeet@gmail.com",

"active" : true,

"phno" : 4234242344

}

{

"\_id" : ObjectId("61dae1810823a3a4267d9e89"),

"name" : "Anshul Jumde",

"email" : "anshuljumde@gmail.com",

"active" : true,

"phno" : 42342342

}

{

"\_id" : ObjectId("61dae1810823a3a4267d9e8a"),

"name" : "Pravin Jumde",

"email" : "pravinjumde@gmail.com",

"active" : true,

"phno" : 234234234

}

{

"\_id" : ObjectId("61dae1810823a3a4267d9e8b"),

"name" : "sangitajumde",

"email" : "sangitajumde@gmail.com",

"active" : true,

"phno" : 23542545432

}

>

**CRUD IN MONGO**

**db.collection.find(query,projection)**

> db.studentcollection.find({name:"sangitajumde"})

{ "\_id" : ObjectId("61dae1810823a3a4267d9e8b"), "name" : "sangitajumde", "email" : "sangitajumde@gmail.com", "active" : true, "phno" : 23542545432 }

> db.studentcollection.find({name:"sangitajumde"}).pretty()

{

"\_id" : ObjectId("61dae1810823a3a4267d9e8b"),

"name" : "sangitajumde",

"email" : "sangitajumde@gmail.com",

"active" : true,

"phno" : 23542545432

}

> db.studentcollection.find({name:"sangitajumde"},{name:1}).pretty()

{ "\_id" : ObjectId("61dae1810823a3a4267d9e8b"), "name" : "sangitajumde" }

> db.studentcollection.find({name:"sangitajumde"},{name:0}).pretty()

{

"\_id" : ObjectId("61dae1810823a3a4267d9e8b"),

"email" : "sangitajumde@gmail.com",

"active" : true,

"phno" : 23542545432

}

> db.studentcollection.find({name:"sangitajumde"},{\_id:0 ,name:1}).pretty()

{ "name" : "sangitajumde" }

>

**LIMITS**

> db.studentcollection.find({active:true}).pretty()

{

"\_id" : ObjectId("61dad9cf0823a3a4267d9e88"),

"name" : "sumeetjumde",

"email" : "jumdesumeet@gmail.com",

"active" : true,

"phno" : 4234242344

}

{

"\_id" : ObjectId("61dae1810823a3a4267d9e89"),

"name" : "Anshul Jumde",

"email" : "anshuljumde@gmail.com",

"active" : true,

"phno" : 42342342

}

{

"\_id" : ObjectId("61dae1810823a3a4267d9e8a"),

"name" : "Pravin Jumde",

"email" : "pravinjumde@gmail.com",

"active" : true,

"phno" : 234234234

}

{

"\_id" : ObjectId("61dae1810823a3a4267d9e8b"),

"name" : "sangitajumde",

"email" : "sangitajumde@gmail.com",

"active" : true,

"phno" : 23542545432

}

> db.studentcollection.find({active:true}).pretty().limit(1)

{

"\_id" : ObjectId("61dad9cf0823a3a4267d9e88"),

"name" : "sumeetjumde",

"email" : "jumdesumeet@gmail.com",

"active" : true,

"phno" : 4234242344

}

**FindOne**

> db.studentcollection.findOne({active:true})

{

"\_id" : ObjectId("61dad9cf0823a3a4267d9e88"),

"name" : "sumeetjumde",

"email" : "jumdesumeet@gmail.com",

"active" : true,

"phno" : 4234242344

}

**Skip**

> db.studentcollection.find({active:true}).pretty().limit(1).skip(1)

{

"\_id" : ObjectId("61dae1810823a3a4267d9e89"),

"name" : "Anshul Jumde",

"email" : "anshuljumde@gmail.com",

"active" : true,

"phno" : 42342342

}

> db.studentcollection.find({active:true}).pretty().limit(1).skip(2)

{

"\_id" : ObjectId("61dae1810823a3a4267d9e8a"),

"name" : "Pravin Jumde",

"email" : "pravinjumde@gmail.com",

"active" : true,

"phno" : 234234234

}

**Update**

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

**NOTE:Updates the fields that match first.**

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

**NOTE:Updates many fields at once.**

* [**db.collection.updateOne(<filter>, <update>, <options>)**](https://docs.mongodb.com/manual/reference/method/db.collection.updateOne/#mongodb-method-db.collection.updateOne)
* [**db.collection.updateMany(<filter>, <update>, <options>)**](https://docs.mongodb.com/manual/reference/method/db.collection.updateMany/#mongodb-method-db.collection.updateMany)
* [**db.collection.replaceOne(<filter>, <update>, <options>)**](https://docs.mongodb.com/manual/reference/method/db.collection.replaceOne/#mongodb-method-db.collection.replaceOne)

| **{** |
| --- |
| **<update operator>: { <field1>: <value1>, ... },** |
| **<update operator>: { <field2>: <value2>, ... },** |
| **...** |
| **}** |

> db.studentcollection.updateOne({name:"sangitajumde"},{$set:{phno:12345678}})

{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

> db.studentcollection.find({active:true}).pretty()

{

"\_id" : ObjectId("61dad9cf0823a3a4267d9e88"),

"name" : "sumeetjumde",

"email" : "jumdesumeet@gmail.com",

"active" : true,

"phno" : 4234242344

}

{

"\_id" : ObjectId("61dae1810823a3a4267d9e89"),

"name" : "Anshul Jumde",

"email" : "anshuljumde@gmail.com",

"active" : true,

"phno" : 42342342

}

{

"\_id" : ObjectId("61dae1810823a3a4267d9e8a"),

"name" : "Pravin Jumde",

"email" : "pravinjumde@gmail.com",

"active" : true,

"phno" : 234234234

}

{

"\_id" : ObjectId("61dae1810823a3a4267d9e8b"),

"name" : "sangitajumde",

"email" : "sangitajumde@gmail.com",

"active" : true,

"phno" : 12345678

}

**DeleteMany**

**deleteMany()=> db.collectionname.deleteMany(DELETION CRITERIA**

* [db.collection.deleteMany()](https://docs.mongodb.com/manual/reference/method/db.collection.deleteMany/#mongodb-method-db.collection.deleteMany)
* [db.collection.deleteOne()](https://docs.mongodb.com/manual/reference/method/db.collection.deleteOne/#mongodb-method-db.collection.deleteOne)

db.inventory.deleteMany({ status : "A" })

| db.inventory.deleteOne( { status: "D" } ) |
| --- |

**To delete entire DOCUMENTS(Entries) in the collection(part of database)**

db.studentcollection.deleteMany({}) //This will delete the entire data in the collection