```
> mongo
> show dbs
> use coursedb
> db.movie.insert({"name" : "Lagaan"})


--Drop Database
> show dbs/databases
> use sampledb1
> db.dropDatabase()
> use coursedb
> db.createCollection("April")
> show collections
> db.April.drop()
> show collections

--Insert/CREATE--
> db.createCollection("student")
> db.student.insert({ name: "Arati", id: 1, address:
{street:"4",area:"juhu",state:"maharashtra"}})
> db.student.find().pretty()
{
 "_id" : ObjectId("58bfe3e74e63ce9e85fb52e9"),
 "name" : "omkar",
 "id" : 1,
 "address" : {
 "street" : "4",
 "area" : "juhu",
 "state" : "maharashtra"
 }
}
db.student.insert({ name: "Dipti", _id: 2, address:
{street:"4",area:"juhu",state:"maharashtra"}})
--Ex 2
db.inventory.insert(
 {
 item: "ABC1",
 details: {
 model: "14Q3",
 manufacturer: "XYZ Company"
 },
 stock: [ { size: "S", qty: 25 }, { size: "M", qty: 50 } ],
 category: "clothing"
 }
)
>db.inventory.find()
>db.inventory.find().pretty()
--Insert array of documents
1. Store in Variable
> var mydocuments =
 [
 {
 item: "ABC2",
 details: { model: "14Q3", manufacturer: "M1 Corporation" },
 stock: [ { size: "M", qty: 50 } ],
 category: "clothing"
 },
 {
 item: "MNO2",
 details: { model: "14Q3", manufacturer: "ABC Company" },
 stock: [ { size: "S", qty: 5 }, { size: "M", qty: 5 }, { size:
"L", qty: 1 } ],
 category: "clothing"
 },
 {
 item: "IJK2",
 details: { model: "14Q2", manufacturer: "M5 Corporation" },
 stock: [ { size: "S", qty: 5 }, { size: "L", qty: 1 } ],
 category: "houseware"
 }
 ]
2. pass variable to command
> db.inventory.insert( mydocuments );
BulkWriteResult({
 "writeErrors" : [ ],
 "writeConcernErrors" : [ ],
 "nInserted" : 3,
 "nUpserted" : 0,
 "nMatched" : 0,
 "nModified" : 0,
 "nRemoved" : 0,
 "upserted" : [ ]
})
-- checking variable value - just type variable name
> mydocuments
db.inventory.insertMany([
{ item: "journal", qty: 25, tags: ["blank", "red"], size: { h: 14, w: 21,
uom: "cm" } },
{ item: "mat", qty: 85, tags: ["gray"], size: { h: 27.9, w: 35.5, uom:
"cm" } },
{ item: "mousepad", qty: 25, tags: ["gel", "blue"], size: { h: 19, w:
22.85, uom: "cm" } } ])
--INSERT-- https://docs.mongodb.com/manual/tutorial/insert-documents/
# db.collection.insertOne() //supported by v3.2 and above
> db.inventory.insertOne( { item: "canvas", qty: 100, tags: ["cotton"],
size: { h: 28, w: 35.5, uom: "cm" } } )
> db.inventory.find({ item: "canvas" }).pretty()
--InsertMany
db.collection.insertMany() //supported by v3.2 and above
--Insert for practice
db.inventory.insert([
 { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status:
"A" },
 { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" },
status: "A" },
 { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status:
"D" },
 { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" },
status: "D" },
 { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" },
status: "A" }
]);
> db.inventory.find( { status: "D" } )
//SELECT * FROM inventory WHERE status = "D"
db.inventory.insert([
 { item: "journal", qty: 25, tags: ["blank", "red"], dim_cm: [ 14, 21 ]
},
 { item: "notebook", qty: 50, tags: ["red", "blank"], dim_cm: [ 14, 21
] },
 { item: "paper", qty: 100, tags: ["red", "blank", "plain"], dim_cm: [
14, 21 ] },
 { item: "planner", qty: 75, tags: ["blank", "red"], dim_cm: [ 22.85,
30 ] },
 { item: "postcard", qty: 45, tags: ["blue"], dim_cm: [ 10, 15.25 ] }
])
> db.inventory.find({ tags: ["red", "blank"] }) //Match exact same tags
(red,blank)
------------------------Project Fields to Return from Query--------------
-------- ---
db.inventory1.insert( [
 { item: "journal", status: "A", size: { h: 14, w: 21, uom: "cm" },
instock: [ { warehouse: "A", qty: 5 } ] },
 { item: "notebook", status: "A", size: { h: 8.5, w: 11, uom: "in" },
instock: [ { warehouse: "C", qty: 5 } ] },
 { item: "paper", status: "D", size: { h: 8.5, w: 11, uom: "in" },
instock: [ { warehouse: "A", qty: 60 } ] },
 { item: "planner", status: "D", size: { h: 22.85, w: 30, uom: "cm" },
instock: [ { warehouse: "A", qty: 40 } ] },
 { item: "postcard", status: "A", size: { h: 10, w: 15.25, uom: "cm" },
instock: [ { warehouse: "B", qty: 15 }, { warehouse: "C", qty: 35 } ] }
])
> db.inventory1.find( { status: "A" } ) // Return all feild
> db.inventory1.find( { status: "A" }, { item: 1, status: 1 } )
//Return the Specified Fields and the _id Field Only
> db.inventory1.find( { status: "A" }, { item: 1, status: 1, _id: 0 } )
//Suppress _id Field
--Return All But the Excluded Fields--
db.inventory1.find( { status: "A" }, { size: 0, instock: 0 } )
* With the exception of the _id field, you can NOT combine inclusion and
exclusion statements in projection documents.
-------------------------UPDATE-----------------------------------------
----- --
db.additionalsubject.insert({name:'student1',subject:['arts','music']})
db.additionalsubject.insert({name:'student2',subject:['sports','arts']})
db.additionalsubject.insert({name:'student3',subject:['sports','cooking',
'music']})
db.additionalsubject.insert({name:'student4',subject:['arts','craft','mus
ic']})
db.additionalsubject.find({subject:'arts'})
--update --- whole replacement
db.additionalsubject.update({name:'student1'},{subject:['craft']})
db.additionalsubject.find().pretty()
----specific update
db.additionalsubject.update({name:'student2'},{$set:{name:'xyz'}})
db.inventory.insert( [
 { item: "canvas", qty: 100, size: { h: 28, w: 35.5, uom: "cm" },
status: "A" },
 { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status:
"A" },
 { item: "mat", qty: 85, size: { h: 27.9, w: 35.5, uom: "cm" }, status:
"A" },
 { item: "mousepad", qty: 25, size: { h: 19, w: 22.85, uom: "cm" },
status: "P" },
 { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" },
status: "P" },
 { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status:
"D" },
 { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" },
status: "D" },
 { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" },
status: "A" },
 { item: "sketchbook", qty: 80, size: { h: 14, w: 21, uom: "cm" },
status: "A" },
 { item: "sketch pad", qty: 95, size: { h: 22.85, w: 30.5, uom: "cm" },
status: "A" }
]);
---- db.collection.updateOne() ----- //update first record
db.inventory.find({item:"paper"})
db.inventory.updateOne( { item: "paper" }, { $set: { "size.uom": "cm",
status: "P" }})
db.inventory.find({item:"paper"})
-- // Reverting -- db.inventory.updateOne( { item: "paper" }, { $set: {
"size.uom": "in", status: "A" }})
-- db.collection.updateMany() -- //update all record as per condition
db.inventory.find({item:"paper"})
db.inventory.updateMany( { item: "paper" }, { $set: { "size.uom": "cm",
status: "P" }})
db.inventory.find({item:"paper"})
--db.collection.update-- single/multi
db.inventory.update( { "status": "P" } , { $set: { status: "D" }})
//single update
db.inventory.update( { "status": "P" }, { $set: { status: "D" }}, {
multi: true } ) // multi mentioned
--------------------- Delete document -----------------------------------
-------------
db.inventory.insert( [
 { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status:
"A" },
 { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" },
status: "P" },
 { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status:
"D" },
 { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" },
status: "D" },
 { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" },
status: "A" },
]);
--first method -- delete all from collection
> db.collection.deleteMany({})
--second method -- delete all from collection
> db.collection.remove({})
db.inventory.deleteMany({ status : "A" })
db.inventory.remove( { status : "P" } )
db.inventory.deleteOne( { status: "D" } ) //one record
db.inventory.remove({ status: "D" }, 1) // first record
db.inventory.find({ status: "D" }).count()
--------------LIMIT()---------------
db.COLLECTION_NAME.find().limit(NUMBER)
db.marks.find()
db.marks.find().limit(5)
---limit---ex2----
> db.mycol.insertMany([
{title:"MongoDB Overview"},
{title:"NoSQL Overview"},
{title:"Tutorials Point Overview"}
])
```

Following example will display only two documents while querying the document.

```
> db.mycol.find({},{"title":1,_id:0}).limit(2)
{"title":"MongoDB Overview"}
```

{"title":"NoSQL Overview"}