D:\Users\mpochira\Desktop\mongodb-win32-x86\_64-enterprise-windows-64-3.0.15\bin>

**mongo**

2018-08-13T12:04:14.565+0530 I CONTROL Hotfix KB2731284 or later update is not

installed, will zero-out data files

MongoDB shell version: 3.0.15

connecting to: test

> **show dbs**

local 0.078GB

myData 0.078GB

> **use mydata**

switched to db mydata

> **db.author.find()**

> **db.author.insert({name:"Mahitha", id:1})**

WriteResult({

"nInserted" : 0,

"writeError" : {

"code" : 13297,

"errmsg" : "db already exists with different case already have:

[myData] trying to create [mydata]"

}

})

> **use myData**

switched to db myData

> **show collections**

author

system.indexes

> **db.author.find()**

{ "\_id" : ObjectId("5b7124bd43031646f82c22a4"), "authorid" : 3, "name" : "Raghu"

, "date" : "01/07/1997" }

> **use mydata**

switched to db mydata

> **ds.dropDatabase()**

2018-08-13T12:07:31.181+0530 E QUERY ReferenceError: ds is not defined

at (shell):1:1

> **db.dropDatabase()**

{ "ok" : 1 }

> **use myData**

switched to db myData

> **db.author.find(name:"mahitha")**

2018-08-13T12:08:12.922+0530 E QUERY SyntaxError: Unexpected token :

> db.author.find(name:mahitha)

2018-08-13T12:08:24.202+0530 E QUERY SyntaxError: Unexpected token :

> db.author.find(name:raghu)

2018-08-13T12:08:41.995+0530 E QUERY SyntaxError: Unexpected token :

> db.author.find(name:"raghu")

2018-08-13T12:08:49.276+0530 E QUERY SyntaxError: Unexpected token :

> db.author.find({name:"raghu"})

> db.author.find({name:raghu})

2018-08-13T12:09:50.383+0530 E QUERY ReferenceError: raghu is not defined

at (shell):1:22

> db.author.find()

{ "\_id" : ObjectId("5b7124bd43031646f82c22a4"), "authorid" : 3, "name" : "Raghu"

, "date" : "01/07/1997" }

> **db.author.find({name:Raghu})**

2018-08-13T12:10:11.744+0530 E QUERY ReferenceError: Raghu is not defined

at (shell):1:22

> **db.author.find({name:"Raghu"})**

{ "\_id" : ObjectId("5b7124bd43031646f82c22a4"), "authorid" : 3, "name" : "Raghu"

, "date" : "01/07/1997" }

> **db.author.remove({})**

WriteResult({ "nRemoved" : 1 })

**> db.author.insert({name:"Mahitha", id:1},{name:"Bharani",id:2})**

WriteResult({ "nInserted" : 1 })

> db.author.find()

**{ "\_id" : ObjectId("5b7128a712d537207ead4d78"), "name" : "Mahitha", "id" : 1 }**

> db.author.find({name:"Bharani"})

> db.author.insert({name:"Bharani",id:2})

WriteResult({ "nInserted" : 1 })

**> db.author.find()**

{ "\_id" : ObjectId("5b7128a712d537207ead4d78"), "name" : "Mahitha", "id" : 1 }

{ "\_id" : ObjectId("5b7128df12d537207ead4d79"), "name" : "Bharani", "id" : 2 }

**> db.author.insert({name:"Raghu",id:3})**

WriteResult({ "nInserted" : 1 })

**> db.author.remove({name:"Raghu"})**

WriteResult({ "nRemoved" : 1 })

**> db.author.find()**

{ "\_id" : ObjectId("5b7128a712d537207ead4d78"), "name" : "Mahitha", "id" : 1 }

{ "\_id" : ObjectId("5b7128df12d537207ead4d79"), "name" : "Bharani", "id" : 2 }

**> show collections**

author

system.indexes

> db.employees.remove({

... "firstname": /^J/

... })

WriteResult({ "nRemoved" : 0 })

> db.employees.remove({ "firstname": /^M/ })

WriteResult({ "nRemoved" : 0 })

**> db.author.remove({ "name": /^M/ })**

WriteResult({ "nRemoved" : 1 })

**> db.author.find()**

{ "\_id" : ObjectId("5b7128df12d537207ead4d79"), "name" : "Bharani", "id" : 2 }

> **db.author.update({id:2},{name:"Bharani Madap"})**

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.author.find()

{ "\_id" : ObjectId("5b7128df12d537207ead4d79"), "name" : "Bharani Madap" }

> db.employees.find( { $or: [ { age:{ $lte: 34 } }, { salary:{$gt:50000}}]})

**> db.author.find( { $or: [ { age:{ $lte: 34 } }, { name:"Mahitha"}]})**

{ "\_id" : ObjectId("5b712eed7c0a2398444264d5"), "name" : "Mahitha", "age" : 21,

"foodILike" : [ [ "pizza", "idali" ] ] }

**> db.author.find({name:{$all : ["Mahitha"]}})**

{ "\_id" : ObjectId("5b712eed7c0a2398444264d5"), "name" : "Mahitha", "age" : 21,

"foodILike" : [ [ "pizza", "idali" ] ] }

**> db.author.find({name:{$all : ["Mahitha"]}}).pretty()**

{

"\_id" : ObjectId("5b712eed7c0a2398444264d5"),

"name" : "Mahitha",

"age" : 21,

"foodILike" : [

[

"pizza",

"idali"

]

]

}

**> db.author.find({name:{$all : ["Mahitha"]}}).count()**

1

> db.mycol.find({},{"title":1,\_id:0})

> db.author.find({},{name:0,\_id:1,age:0,foodilike:0})

Error: error: {

"$err" : "Can't canonicalize query: BadValue Projection cannot have a mi

x of inclusion and exclusion.",

"code" : 17287

}

> db.mycol.find({})

> db.author.find({})

{ "\_id" : ObjectId("5b7128df12d537207ead4d79"), "name" : "Bharani Madap" }

{ "\_id" : ObjectId("5b712eed7c0a2398444264d5"), "name" : "Mahitha", "age" : 21,

"foodILike" : [ [ "pizza", "idali" ] ] }

> db.author.find({},{name:0,\_id:1,age:0,foodILike:0})

Error: error: {

"$err" : "Can't canonicalize query: BadValue Projection cannot have a mi

x of inclusion and exclusion.",

"code" : 17287

}

> db.author.find({},{\_id:1})

{ "\_id" : ObjectId("5b7128df12d537207ead4d79") }

{ "\_id" : ObjectId("5b712eed7c0a2398444264d5") }

> db.author.find({\_id:1, name:0})

> db.author.find({}{\_id:1, name:0})

2018-08-13T15:21:59.702+0530 E QUERY SyntaxError: Unexpected token {

> db.author.find({},{\_id:1, name:0})

Error: error: {

"$err" : "Can't canonicalize query: BadValue Projection cannot have a mi

x of inclusion and exclusion.",

"code" : 17287

}

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

Error: error: {

"$err" : "Can't canonicalize query: BadValue Projection cannot have a mi

x of inclusion and exclusion.",

"code" : 17287

}

> db.author.find({},{\_id:1,"name":1})

{ "\_id" : ObjectId("5b7128df12d537207ead4d79"), "name" : "Bharani Madap" }

{ "\_id" : ObjectId("5b712eed7c0a2398444264d5"), "name" : "Mahitha" }

> db.author.find({},{\_id:1,"name":1}).limit(5)

{ "\_id" : ObjectId("5b7128df12d537207ead4d79"), "name" : "Bharani Madap" }

{ "\_id" : ObjectId("5b712eed7c0a2398444264d5"), "name" : "Mahitha" }

> db.author.find({},{\_id:1,"name":1}).limit(1)

{ "\_id" : ObjectId("5b7128df12d537207ead4d79"), "name" : "Bharani Madap" }

> db.author.find({}).sort({name:-1})

{ "\_id" : ObjectId("5b712eed7c0a2398444264d5"), "name" : "Mahitha", "age" : 21, "foodILike" : ["pizza","idali" ] ] }

{ "\_id" : ObjectId("5b7128df12d537207ead4d79"), "name" : "Bharani Madap" }

> db.author.find({}).sort({name:1})

{ "\_id" : ObjectId("5b7128df12d537207ead4d79"), "name" : "Bharani Madap" }

{ "\_id" : ObjectId("5b712eed7c0a2398444264d5"), "name" : "Mahitha", "age" : 21,

"foodILike" : [ [ "pizza", "idali" ] ] }

>