

BIG DATA ANALYTICS LAB-1

13-04-2022

Name: Aruna Ravi K R

USN: 1BM19CS225

Batch: B4

```
Q bmsce@bmsce-Precision-T1700: ~
(base) bmsce@bmsce-Precision-T1700:~$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("2139b1da-2835-4627-bad9-167fee5d8c1c") }
MongoDB server version: 3.6.8
Server has startup warnings:
2022-04-13T19:39:10.171+0530 I STORAGE [initandlisten]
2022-04-13T19:39:10.171+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is
strongly recommended with the WiredTiger storage engine
2022-04-13T19:39:10.171+0530 I STORAGE [initandlisten] ** See http://dochub.mongodb.or
g/core/prodnotes-filesystem
2022-04-13T19:39:14.150+0530 I CONTROL [initandlisten]
2022-04-13T19:39:14.150+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enable
d for the database.
2022-04-13T19:39:14.150+0530 I CONTROL [initandlisten] ** Read and write access to dat
a and configuration is unrestricted.
2022-04-13T19:39:14.150+0530 I CONTROL [initandlisten]
> use arunaDB;
switched to db arunaDB
> db;
arunaDB
> show dbs;
admin 0.000GB
config 0.000GB
local 0.000GB
> db.createCollection("Student");
{ "ok" : 1 }
> db.Student.insert({id:1,name:"Aruna Ravi K R",sem:6,hobbies:"cricket"});
WriteResult({ "nInserted" : 1 })
> db.Student.update({id:3,name:"Manjunatha",sem:1},{ $set:{hobbies:"skating"}},{upsert:true})
WriteResult({
  "nMatched" : 0,
  "nUpserted" : 1,
  "nModified" : 0,
  "_id" : ObjectId("6256945713cdf85df653a993")
})
> db.Student.find({name:"Manjunatha"});
{ "_id" : ObjectId("6256945713cdf85df653a993"), "id" : 3, "name" : "Manjunatha", "sem" : 1, "hob
bies" : "skating" }
> db.Student.find({}, {name:1});
{ "_id" : ObjectId("625693993478889c646fe83f"), "name" : "Aruna Ravi K R" }
{ "_id" : ObjectId("6256945713cdf85df653a993"), "name" : "Manjunatha" }
> db.Student.find({sem:6}).pretty();
```

```

bmsce@bmsce-Precision-T1700: ~
{ "_id" : ObjectId("625693993478889c646fe83f"), "name" : "Aruna Ravi K R" }
{ "_id" : ObjectId("6256945713cdf85df653a993"), "name" : "Manjunatha" }
> db.Student.find({sem:6}).pretty();
{
  "_id" : ObjectId("625693993478889c646fe83f"),
  "id" : 1,
  "name" : "Aruna Ravi K R",
  "sem" : 6,
  "hobbies" : "cricket"
}
> db.Student.find({hobbies:{$in:['skating','cricket']}}).pretty();
{
  "_id" : ObjectId("625693993478889c646fe83f"),
  "id" : 1,
  "name" : "Aruna Ravi K R",
  "sem" : 6,
  "hobbies" : "cricket"
}
> db.Student.find({name:/^M/}).pretty();
{
  "_id" : ObjectId("6256945713cdf85df653a993"),
  "id" : 3,
  "name" : "Manjunatha",
  "sem" : 1,
  "hobbies" : "skating"
}
> db.Student.find({name:/R/}).pretty();
{
  "_id" : ObjectId("625693993478889c646fe83f"),
  "id" : 1,
  "name" : "Aruna Ravi K R",
  "sem" : 6,
  "hobbies" : "cricket"
}
> db.Student.count();
2
> db.Student.find.sort({name:-1}).pretty();
2022-04-13T14:53:27.847+0530 E QUERY [thread1] TypeError: db.Student.find.sort is not a function :
@shell):1:1
> db.Student.find().sort({name:-1}).pretty();
{
  "_id" : ObjectId("6256945713cdf85df653a993"),

```

```
bmsce@bmsce-Precision-T1700: ~  
    "name" : "Aruna Ravi K R",  
    "sem" : 6,  
    "hobbies" : "cricket"  
  }  
> db.Student.count();  
2  
> db.Student.find.sort({name:-1}).pretty();  
2022-04-13T14:53:27.847+0530 E QUERY [thread1] TypeError: db.Student.find.sort is not a function :  
@shell):1:1  
> db.Student.find().sort({name:-1}).pretty();  
{  
  "_id" : ObjectId("6256945713cdf85df653a993"),  
  "id" : 3,  
  "name" : "Manjunatha",  
  "sem" : 1,  
  "hobbies" : "skating"  
}  
{  
  "_id" : ObjectId("625693993478889c646fe83f"),  
  "id" : 1,  
  "name" : "Aruna Ravi K R",  
  "sem" : 6,  
  "hobbies" : "cricket"  
}  
> db.Student.save({name:"amar",sem:5});  
WriteResult({ "nInserted" : 1 })  
> db.Student.update({id:3},{ $set:{Location:"Network"}});  
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })  
> db.Student.find({name:/r$/}).pretty();  
{ "_id" : ObjectId("625696fe3478889c646fe840"), "name" : "amar", "sem" : 5 }  
> db.Student.update({id:3},{ $set:{Location:null}});  
2022-04-13T14:58:18.222+0530 E QUERY [thread1] SyntaxError: missing : after property id @shell):1:30  
> db.Student.update({id:3},{ $set:{Location:null}});  
2022-04-13T14:58:32.140+0530 E QUERY [thread1] SyntaxError: missing : after property id @shell):1:30  
> db.Student.update({id:3},{ $set:{Location:null}});  
2022-04-13T14:59:05.114+0530 E QUERY [thread1] SyntaxError: missing ) after argument list @shell):1:47  
> db.Student.update({id:3},{ $set:{Location:null}});  
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })  
> db.Student.count();
```

```
bmsce@bmsce-Precision-T1700: ~  
> db.Student.update({id:3},{ $set:{Location:null}});  
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })  
> db.Student.count();  
3  
> db.Student.count({sem:6});  
1  
> db.Student.find({sem:6}).limit(2).pretty();  
{  
  "_id" : ObjectId("625693993478889c646fe83f"),  
  "id" : 1,  
  "name" : "Aruna Ravi K R",  
  "sem" : 6,  
  "hobbies" : "cricket"  
}  
> db.Student.find().sort({name:1}).pretty();  
{  
  "_id" : ObjectId("625693993478889c646fe83f"),  
  "id" : 1,  
  "name" : "Aruna Ravi K R",  
  "sem" : 6,  
  "hobbies" : "cricket"  
}  
{  
  "_id" : ObjectId("6256945713cdf85df653a993"),  
  "id" : 3,  
  "name" : "Manjunatha",  
  "sem" : 1,  
  "hobbies" : "skating",  
  "Location" : null  
}  
{ "_id" : ObjectId("625696fe3478889c646fe840"), "name" : "amar", "sem" : 5 }  
> db.Student.find().skip(2).pretty();  
{ "_id" : ObjectId("625696fe3478889c646fe840"), "name" : "amar", "sem" : 5 }  
>  
> db.createcollection("food");  
2022-04-13T15:03:36.470+0530 E QUERY [thread1] TypeError: db.createcollection is not a function :  
@(shell):1:1  
> db.createCollection("food");  
{ "ok" : 1 }  
> db.food.insert({id:1,fruits:['m','a','g']});  
WriteResult({ "nInserted" : 1 })  
> db.food.insert({id:2,fruits:['g','m','c']});
```



```

bmsce@bmsce-Precision-T1700: ~
> db.food.insert({id:1,fruits:['m','a','g']});
WriteResult({ "nInserted" : 1 })
> db.food.insert({id:2,fruits:['g','m','c']});
WriteResult({ "nInserted" : 1 })
> db.food.insert({id:3,fruits:['b','m']});
WriteResult({ "nInserted" : 1 })
> db.food.find({fruits:['g','m','c']});
{ "_id" : ObjectId("625699523478889c646fe842"), "id" : 2, "fruits" : [ "g", "m", "c" ] }
> db.food.find({fruits:{$size:2}}).pretty();
{
  "_id" : ObjectId("625699603478889c646fe843"),
  "id" : 3,
  "fruits" : [
    "b",
    "m"
  ]
}
> db.food.find({id:1,{fruits:{$slice:2}}).pretty();
...
...
>
>
> db.food.find({fruits:{$all:['m','g']}}).pretty();
{
  "_id" : ObjectId("6256993d3478889c646fe841"),
  "id" : 1,
  "fruits" : [
    "m",
    "a",
    "g"
  ]
}
{
  "_id" : ObjectId("625699523478889c646fe842"),
  "id" : 2,
  "fruits" : [
    "g",
    "m",
    "c"
  ]
}
> db.food.update({id:3},{set:fruits:['b','m','o']});
2022-04-13T15:11:26.973+0530 E QUERY [thread1] SyntaxError: missing } after property list @(<s

```

```

}
{
  "_id" : ObjectId("625699523478889c646fe842"),
  "id" : 2,
  "fruits" : [
    "g",
    "m",
    "c"
  ]
}
> db.food.update({id:3},{set:fruits:['b','m','o']});
2022-04-13T15:11:26.973+0530 E QUERY [thread1] SyntaxError: missing } after property list @(shell):1:34
> db.food.update({id:3},{set:fruits:['b','m','o']});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.food.update({id:2},{push:price:{g=80,m:100,c:150,b:60,o:70}});
...
...
> db.createCollection("Customer");
{ "ok" : 1 }
> db.Customer.aggregate({$group:{id:"$custID",TotAccBal:{$sum:"$AccBal"}}});

assert: command failed: {
  "ok" : 0,
  "errmsg" : "The field 'id' must be an accumulator object",
  "code" : 40234,
  "codeName" : "Location40234"
} : aggregate failed
_getErrorWithCode@src/mongo/shell/utils.js:25:13
doassert@src/mongo/shell/assert.js:16:14
assert.commandWorked@src/mongo/shell/assert.js:403:5
DB.prototype._runAggregate@src/mongo/shell/db.js:260:9
DBCollection.prototype.aggregate@src/mongo/shell/collection.js:1212:12
@(shell):1:1

2022-04-13T15:16:14.450+0530 E QUERY [thread1] Error: command failed: {
  "ok" : 0,
  "errmsg" : "The field 'id' must be an accumulator object",
  "code" : 40234,
  "codeName" : "Location40234"
} : aggregate failed :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
doassert@src/mongo/shell/assert.js:16:14

```