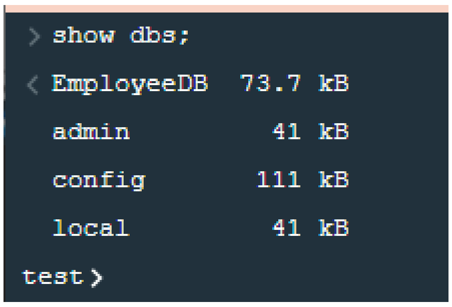
MongoDB commands for database

1. show dbs – show all databases.



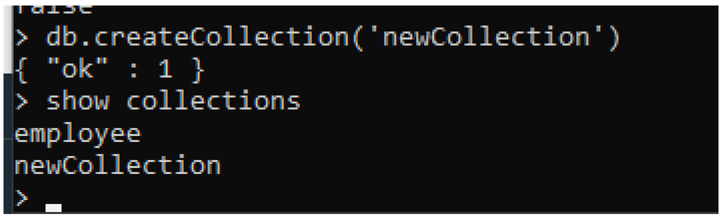
```
> show dbs;
< EmployeeDB   73.7 kB
  admin          41 kB
  config        111 kB
  local          41 kB
test>
```

2. use EmployeeDB – To switch to new database or create one.
3. db – List out the current db name.
4. db.dropDatabase() – To delete or drop the current database.

MongoDB commands for Collections

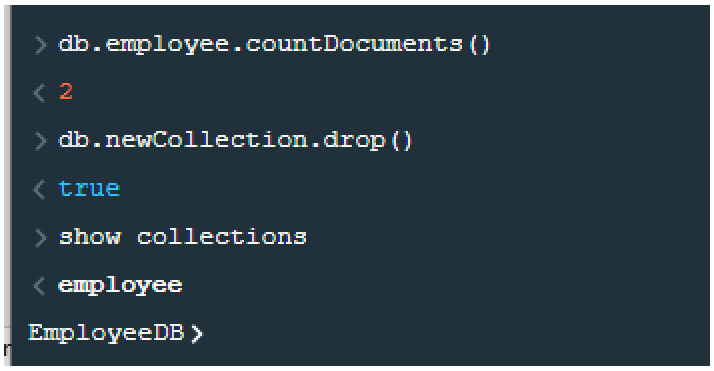
5. show collections – to list out the collections in the current database.

6. db.createCollection('newCollection') – Creating a new collection inside a database.



```
false
> db.createCollection('newCollection')
{ "ok" : 1 }
> show collections
employee
newCollection
> _
```

7. db.newCollection.drop() – To drop a collection



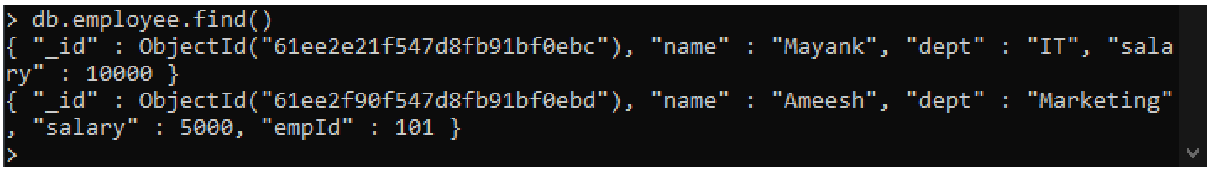
```
> db.employee.countDocuments()
< 2
> db.newCollection.drop()
< true
> show collections
< employee
EmployeeDB>
```

MongoDB commands for documents

8. db.<Collection Name>.insert({'name':'Harry', 'lang':'Javascript','member_since':5}) – Insert document in a collection

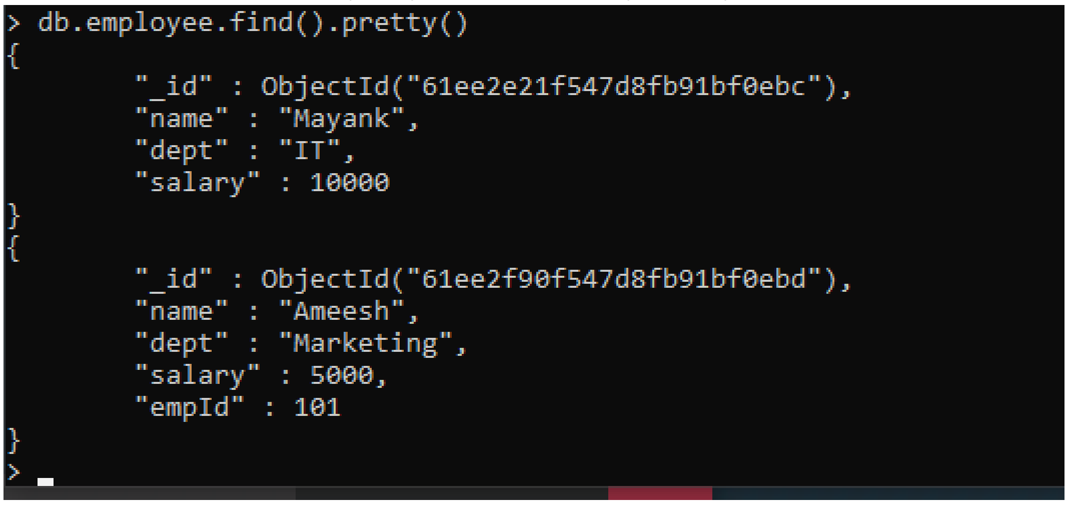
9. db.comments.insertMany([{'name':'Harry', 'lang':'Javascript','member_since':5},{'name':'Mayank', 'lang':'C++','member_since':3},{'name':'Swapnil', 'lang':'Python','member_since':1}]) – Insert many documents at once.

10. db.<Collection name>.find() – Display all documents inside the collection.



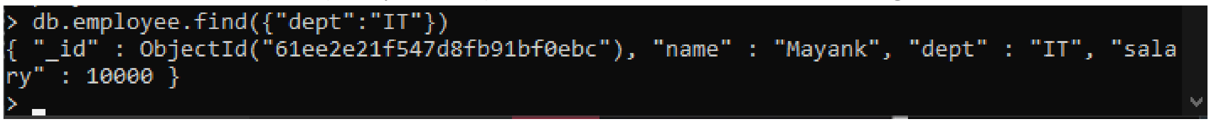
```
> db.employee.find()
{ "_id" : ObjectId("61ee2e21f547d8fb91bf0ebc"), "name" : "Mayank", "dept" : "IT", "sala
ry" : 10000 }
{ "_id" : ObjectId("61ee2f90f547d8fb91bf0ebd"), "name" : "Ameesh", "dept" : "Marketing"
, "salary" : 5000, "empId" : 101 }
>
```

11. db.<Collection Name>.find().pretty() – Gives the output in a prettified manner



```
> db.employee.find().pretty()
{
        "_id" : ObjectId("61ee2e21f547d8fb91bf0ebc"),
        "name" : "Mayank",
        "dept" : "IT",
        "salary" : 10000
}
{
        "_id" : ObjectId("61ee2f90f547d8fb91bf0ebd"),
        "name" : "Ameesh",
        "dept" : "Marketing",
        "salary" : 5000,
        "empId" : 101
}
> 
```

12. db.<Collection Name>.find({“dept”:”IT”}) – To find the document with the given JSON.

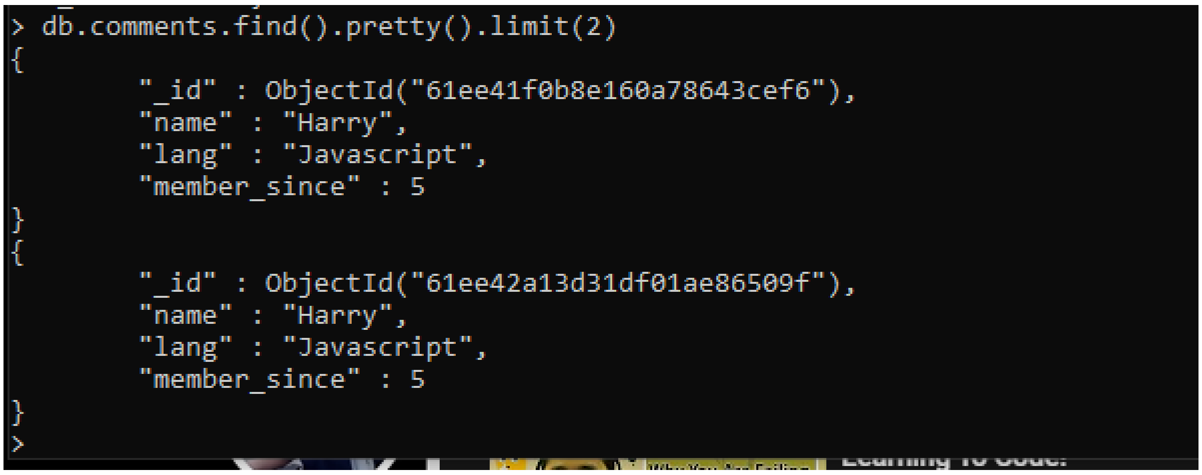


```
> db.employee.find({"dept":"IT"})
{ "_id" : ObjectId("61ee2e21f547d8fb91bf0ebc"), "name" : "Mayank", "dept" : "IT", "sala
ry" : 10000 }
> 
```

13. db.employee.find().count() – Count the number of documents inside the collection.

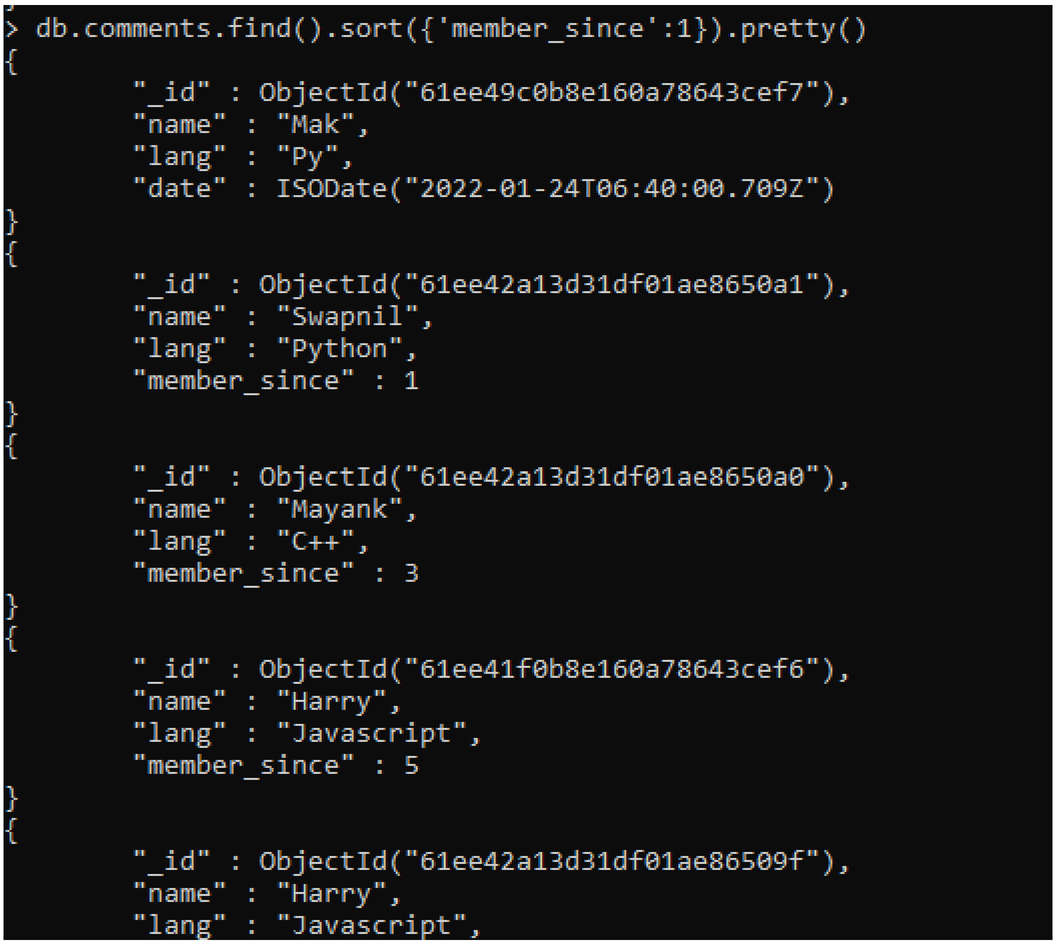
14. db.<collection_name>.findOne({key: 'value'}) - Find the first row matching the object

15. db.comments.find().pretty().limit(2) – Limit the output rows


```
> db.comments.find().pretty().limit(2)
{
        "_id" : ObjectId("61ee41f0b8e160a78643cef6"),
        "name" : "Harry",
        "lang" : "Javascript",
        "member_since" : 5
}
{
        "_id" : ObjectId("61ee42a13d31df01ae86509f"),
        "name" : "Harry",
        "lang" : "Javascript",
        "member_since" : 5
}
>
```

16. db.comments.find().limit(2).count() – It will count the total number of rows in the output without being limited. So the answer will not be 2 but whatever is the count of the find function. So limit function only limits the output to be displayed.

17. Sort according to a field


```
> db.comments.find().sort({'member_since':1}).pretty()
{
        "_id" : ObjectId("61ee49c0b8e160a78643cef7"),
        "name" : "Mak",
        "lang" : "Py",
        "date" : ISODate("2022-01-24T06:40:00.709Z")
}
{
        "_id" : ObjectId("61ee42a13d31df01ae8650a1"),
        "name" : "Swapnil",
        "lang" : "Python",
        "member_since" : 1
}
{
        "_id" : ObjectId("61ee42a13d31df01ae8650a0"),
        "name" : "Mayank",
        "lang" : "C++",
        "member_since" : 3
}
{
        "_id" : ObjectId("61ee41f0b8e160a78643cef6"),
        "name" : "Harry",
        "lang" : "Javascript",
        "member_since" : 5
}
{
        "_id" : ObjectId("61ee42a13d31df01ae86509f"),
        "name" : "Harry",
        "lang" : "Javascript",
```

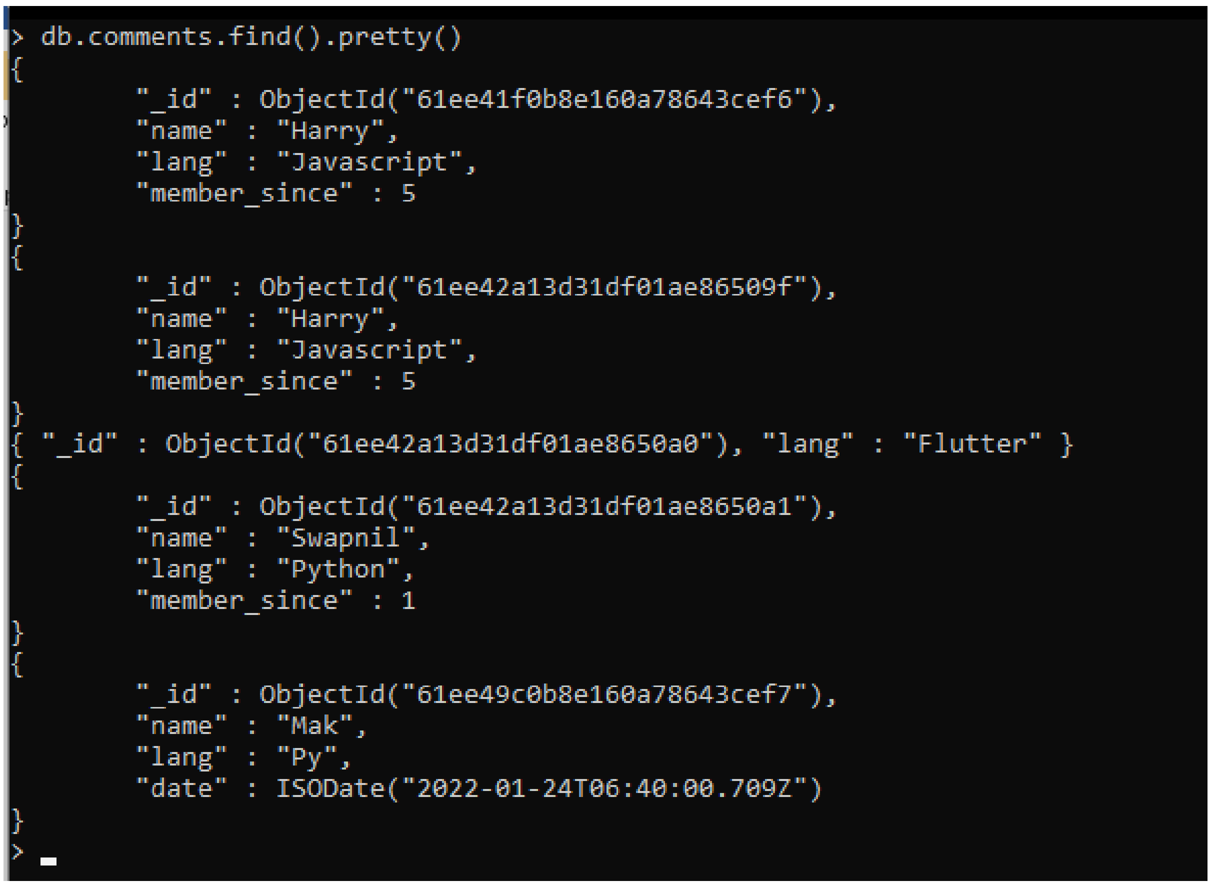
For descending order, give -1 instead of 1.

18. db.comments.findOne({‘name’:’Harry’}) – Find the first row matching the object

19. db.comments.find({'member_since':{$gt:3}}).pretty() – find documents having field value greater than 3.

    $gt – greater than, $gte – greater than equal to, $lt – less than, $lte – less than equal to

20. db.comments.update({"name":"Mayank"},{"lang":"Flutter"}) – To update the document. This will overwrite all the fields inside the matching document with the second argument. By default it only updates the first matching document.

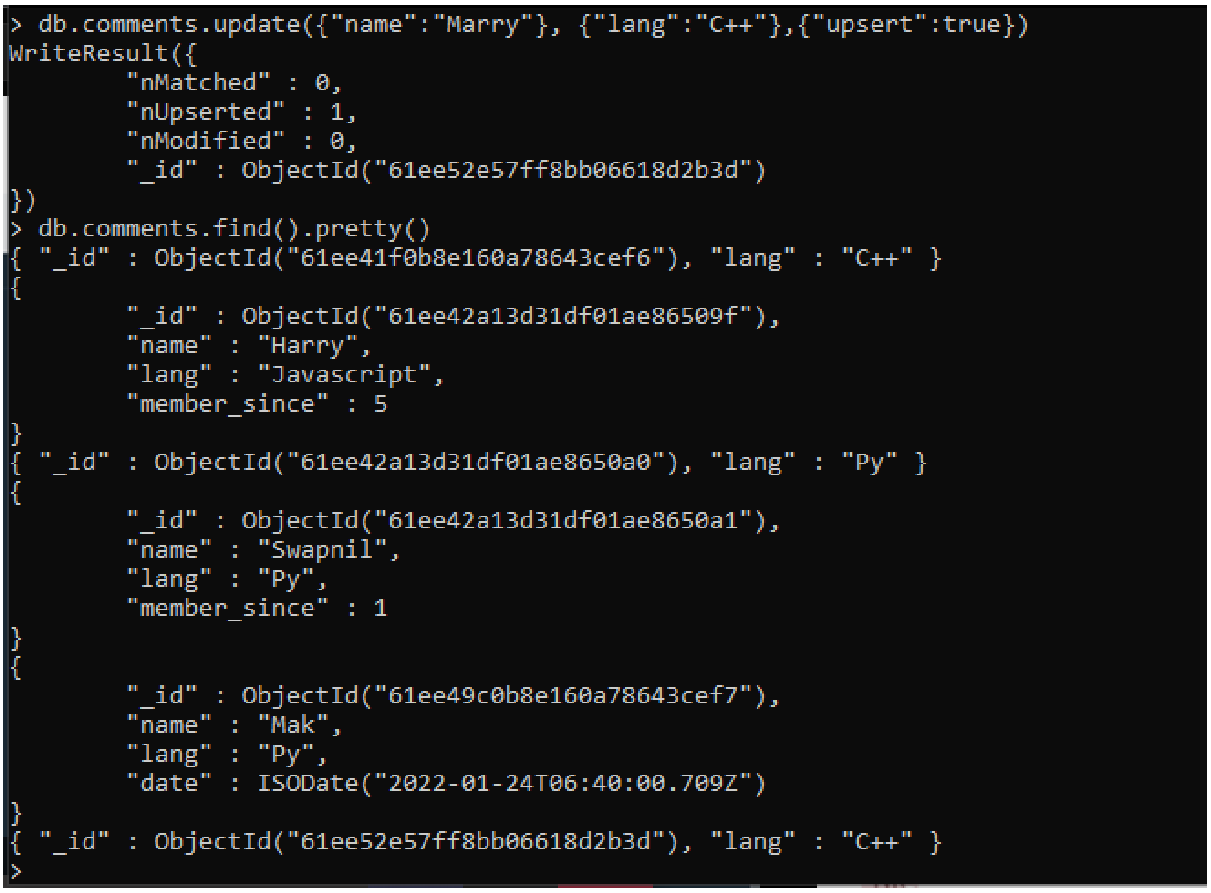


```
> db.comments.find().pretty()
{
        "_id" : ObjectId("61ee41f0b8e160a78643cef6"),
        "name" : "Harry",
        "lang" : "Javascript",
        "member_since" : 5
}
{
        "_id" : ObjectId("61ee42a13d31df01ae86509f"),
        "name" : "Harry",
        "lang" : "Javascript",
        "member_since" : 5
}
{ "_id" : ObjectId("61ee42a13d31df01ae8650a0"), "lang" : "Flutter" }
{
        "_id" : ObjectId("61ee42a13d31df01ae8650a1"),
        "name" : "Swapnil",
        "lang" : "Python",
        "member_since" : 1
}
{
        "_id" : ObjectId("61ee49c0b8e160a78643cef7"),
        "name" : "Mak",
        "lang" : "Py",
        "date" : ISODate("2022-01-24T06:40:00.709Z")
}
>
```

You can see the document of Mayank has been overwritten and it contains only the json that we provided.

21. db.comments.update({"name":"Swapnil"}, {$set:{"lang":"Py"}}) – In the document where name is swapnil, the lang will be updated to Py.

22. db.comments.update({"name":"Marry"}, {"lang":"C++"},{"upsert":true}) – Now “Marry” is not there so a new document will be added as upsert is set to true.



```
> db.comments.update({"name":"Marry"}, {"lang":"C++"},{"upsert":true})
WriteResult({
        "nMatched" : 0,
        "nUpserted" : 1,
        "nModified" : 0,
        "_id" : ObjectId("61ee52e57ff8bb06618d2b3d")
})
> db.comments.find().pretty()
{ "_id" : ObjectId("61ee41f0b8e160a78643cef6"), "lang" : "C++" }
{
        "_id" : ObjectId("61ee42a13d31df01ae86509f"),
        "name" : "Harry",
        "lang" : "Javascript",
        "member_since" : 5
}
{ "_id" : ObjectId("61ee42a13d31df01ae8650a0"), "lang" : "Py" }
{
        "_id" : ObjectId("61ee42a13d31df01ae8650a1"),
        "name" : "Swapnil",
        "lang" : "Py",
        "member_since" : 1
}
{
        "_id" : ObjectId("61ee49c0b8e160a78643cef7"),
        "name" : "Mak",
        "lang" : "Py",
        "date" : ISODate("2022-01-24T06:40:00.709Z")
}
{ "_id" : ObjectId("61ee52e57ff8bb06618d2b3d"), "lang" : "C++" }
>
```

Notes: https://www.codewithharry.com/blogpost/mongodb-cheatsheet

Sir Notes

#Mango Db Commands Rows

#show all the row in collections
>db.<collection_name>.find()

#insert single row

>db.<collection_name>.insert({});

db.employee.insert(
{
"name": "ramesh",
"dept":"Account",

```
"salary":5000
});
```

#insert multiple rows

```
>db.<collection_name>.insertMany([{},{},{}])
```

```
db.employee.insertMany([
{
"name": "c",
"dept":"c",
"salary":2000
},
{
"name": "python",
"dept":"python",
"salary":1000
}
]);
```

#Show all Rows in a Collection (Prettified)

```
>db.<collection_name>.find().pretty()
```

#Find the first row matching the object

```
db.<collection_name>.findOne({key: 'value'})
```

#Search in a MongoDb Database

```
db.<collection_name>.find({key:'value'})
```

#Limit the number of rows in output

```
>db.comments.find().limit(2)
```

#Count the number of rows in the output

```
>db.comments.find().count()
```