

CRUD

1. Starting MongoDB Instance. Open a terminal window and type the following

```
[tom@localhost ~]$ su
Password:
[root@localhost tom]# mongod --dbpath /var/lib/mongo

2014-07-28T06:39:49.221+0200 [initandlisten] MongoDB starting : pid=2107
port=27017 dbpath=/var/lib/mongo 64-bit host=localhost.localdomain
2014-07-28T06:39:49.222+0200 [initandlisten] db version v2.6.3
2014-07-28T06:39:49.222+0200 [initandlisten] git version:
255f67a66f9603c59380b2a389e386910bbb52cb
2014-07-28T06:39:49.222+0200 [initandlisten] build info: Linux
build12.njl.10gen.cc 2.6.32-431.3.1.el6.x86_64 #1 SMP Fri Jan 3 21:39:27
UTC 2014 x86_64 BOOST_LIB_VERSION=1_49
2014-07-28T06:39:49.222+0200 [initandlisten] allocator: tcmalloc
2014-07-28T06:39:49.222+0200 [initandlisten] options: { storage: { dbPath:
"/var/lib/mongo" } }
2014-07-28T06:39:49.239+0200 [initandlisten] journal
dir=/var/lib/mongo/journal
2014-07-28T06:39:49.239+0200 [initandlisten] recover : no journal files
present, no recovery needed
2014-07-28T06:39:50.107+0200 [initandlisten] waiting for connections on
port 27017
```

2. Open another Terminal window and type the following.

```
[tom@localhost ~]$ mongo

MongoDB shell version: 2.6.3
connecting to: test
```

3. Display all existing databases

```
> show dbs

admin                (empty)
bookstore            0.078GB
local                0.078GB
sample_database     0.078GB
test                 0.078GB
```

CRUD

4. Switch to one of the databases

```
> use bookstore
```

```
switched to db bookstore
```

5. Display all collections in the database

```
> show collections
```

```
cats  
imported_new  
map_reduce_example  
map_reduce_example1  
mybookstore  
mybookstore1  
session_stat  
sessions  
system.indexes  
system.js  
system.profile
```

6. Drop a collection

```
> db.cats.drop()
```

```
true
```

```
> show collections
```

```
imported_new  
map_reduce_example  
map_reduce_example1  
mybookstore  
mybookstore1  
session_stat  
sessions  
system.indexes  
system.js
```

CRUD

7. Create a new database.

```
> use sample_database
```

```
switched to db sample_database
```

8. Create a new collection.

```
> db.sample_collection.insert({_id:1,stud_name:"suresh",score:40})
```

```
WriteResult({ "nInserted" : 1 })
```

9. Insert a new document using “update()”

```
>db.sample_collection.update({_id:2},{ $set:{stud_name:"ramesh",score:71}},  
{upsert:true})
```

```
WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : 2  
})
```

10. Insert a new document using “save()”

```
> db.sample_collection.save({_id:3,stud_name:"sachin",score:35})
```

```
WriteResult({ "nInserted" : 1 })
```

11. Insert a new document using “insert()”

```
> db.sample_collection.insert({stud_name:"sameer",score:99})
```

```
WriteResult({ "nInserted" : 1 })
```

CRUD

12. Display all documents

```
> db.sample_collection.find().pretty()

{ "_id" : 1, "stud_name" : "suresh", "score" : 40 }
{ "_id" : 2, "stud_name" : "ramesh", "score" : 71 }
{ "_id" : 3, "stud_name" : "sachin", "score" : 35 }
{
  "_id" : ObjectId("53d5c6ad595947aa08fcfb94"),
  "stud_name" : "sameer",
  "score" : 99
}
```

13. Display all documents that match a specific condition.

```
> db.sample_collection.find({stud_name:"sachin"})

{ "_id" : 3, "stud_name" : "sachin", "score" : 35 }
```

14. Display all documents that match a specific condition using operators.

```
> db.sample_collection.find({stud_name:{ $in:["suresh","ramesh"]}})

{ "_id" : 1, "stud_name" : "suresh", "score" : 40 }
{ "_id" : 2, "stud_name" : "ramesh", "score" : 71 }
```

15. Display all documents that match a specific condition using "and"

```
> db.sample_collection.find({stud_name:{
  $in:["suresh","ramesh"]},score:40})

{ "_id" : 1, "stud_name" : "suresh", "score" : 40 }
```

CRUD

16. Display all documents that match a specific condition using “or”

```
>db.sample_collection.find({$or:[{stud_name:{ $in:["suresh","ramesh"] }},{score:35}]})
```

```
{ "_id" : 1, "stud_name" : "suresh", "score" : 40 }
{ "_id" : 2, "stud_name" : "ramesh", "score" : 71 }
{ "_id" : 3, "stud_name" : "sachin", "score" : 35 }
```

17. Specify the fields to display in the output

```
> db.sample_collection.find({stud_name:"suresh"},{score:1})
```

```
{ "_id" : 1, "score" : 40 }
```

18. Display only the specified fields

```
> db.sample_collection.find({stud_name:"suresh"},{score:1,_id:0})
```

```
{ "score" : 40 }
```

19. Specify the fields not to be displayed in the output

```
> db.sample_collection.find({stud_name:"suresh"},{score:0})
```

```
{ "_id" : 1, "stud_name" : "suresh" }
```

20. Add new field to the existing document

```
> db.sample_collection.update({stud_name:"ramesh"},{$set:{marks:09}})
```

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

```
> db.sample_collection.find().pretty()
```

```
{ "_id" : 1, "stud_name" : "suresh", "score" : 40 }
{ "_id" : 2, "stud_name" : "ramesh", "score" : 71, "marks" : 9 }
{ "_id" : 3, "stud_name" : "sachin", "score" : 35 }
{
  "_id" : ObjectId("53d5c6ad595947aa08fcfb94"),
  "stud_name" : "sameer",
  "score" : 99
}
```

21. Replace entire document using “update”

```
> db.sample_collection.update({stud_name:"ramesh"},{"marks":09})
```

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

```
> db.sample_collection.find().pretty()
```

```
{ "_id" : 1, "stud_name" : "suresh", "score" : 40 }
{ "_id" : 2, "marks" : 9 }
{ "_id" : 3, "stud_name" : "sachin", "score" : 35 }
{
  "_id" : ObjectId("53d5c6ad595947aa08fcfb94"),
  "stud_name" : "sameer",
  "score" : 99
}
```

CRUD

22. Modify the values of the existing fields in the document

```
> db.sample_collection.update({stud_name:"suresh"},{$set:{"score":99}})
```

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

```
> db.sample_collection.find().pretty()
```

```
{ "_id" : 1, "stud_name" : "suresh", "score" : 99 }
{ "_id" : 2, "marks" : 9 }
{ "_id" : 3, "stud_name" : "sachin", "score" : 35 }
{
  "_id" : ObjectId("53d5c6ad595947aa08fcfb94"),
  "stud_name" : "sameer",
  "score" : 99
}
```

23. Delete a collection

```
> db.sample_collection.remove({marks:9})
```

24. Exit from MongoDB shell

```
> exit
```

```
Bye
```

25. Shutdown MongoDB Instance

```
[tom@localhost ~]$ sudo mongod --dbpath /var/lib/mongo --shutdown
```

```
[sudo] password for tom:
killing process with pid: 2127
[tom@localhost ~]$ exit
```