



Department of Data Engineering  
Subject: Big Data Analytics Lab (CSL702)

Experiment No: 03

Roll No: 75	Name: Ishan Shirode	Div: C	Batch: C3
-------------	---------------------	--------	-----------

**Aim:** To Install and configure MongoDB and Execute NoSQL Commands.

**Code & Output:**

1) db.help

Database Class	
getMongo	Returns the current database connection
getName	Returns the name of the DB
getCollectionNames	Returns an array containing the names of all collections in the current database.
getCollectionInfos	Returns an array of documents with collection information, i.e. collection name and options, for the current database.
runCommand	Runs an arbitrary command on the database.
adminCommand	Runs an arbitrary command against the admin database.
aggregate	Runs a specified admin/diagnostic pipeline which does not require an underlying collection.
getSiblingDB	Returns another database without modifying the db variable in the shell environment.
getCollection	Returns a collection or a view object that is functionally equivalent to using the db.<collectionName>.
dropDatabase	Removes the current database, deleting the associated data files.
createUser	Creates a new user for the database on which the method is run. db.createUser() returns a duplicate user error if the user already exists on the database.
updateUser	Updates the user's profile on the database on which you run the method. An update to a field completely replaces the previous field's values. This includes updates to the user's roles array.
changeUserPassword	Updates a user's password. Run the method in the database where the user is defined, i.e. the database you created the user.
logout	Ends the current authentication session. This function has no effect if the current session is not authenticated.
dropUser	Removes the user from the current database.
dropAllUsers	Removes all users from the current database.
auth	Allows a user to authenticate to the database from within the shell.
grantRolesToUser	Grants additional roles to a user.
revokeRolesFromUser	Removes a one or more roles from a user on the database.
getUser	Returns user information for a specified user. Run this method on the user's database. The user must exist on the database on which the method runs.
getUsers	Returns information for all the users in the database.
createCollection	Create new collection

2) show dbs

Janhavi	88.00 KiB
Rahul	8.00 KiB
Tej	8.00 KiB
admin	40.00 KiB
config	108.00 KiB
dd	8.00 KiB
local	72.00 KiB
mitesh	72.00 KiB
myrahul	40.00 KiB

3) use Tej

switched to db Tej



**Department of Data Engineering**  
**Subject: Big Data Analytics Lab (CSL702)**

4) db

```
Tej
```

5) db.dropDatabase()

```
{ ok: 1, dropped: 'Tej' }
```

6) db.createCollection("UCOE")

```
{ ok: 1 }
```

7) show collections

```
UCOE
```

8) db.Save.drop()

```
show collections
Save
UCOE
db.Save.drop()
true
show collections
UCOE
```

9) db.UCOE.insert([{name: 'Tej',address: 'Virar'}, {name: 'Pradeep',address: 'Kaman'}])

```
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('68c8f20380eed7e48bcf0aaa'),
    '1': ObjectId('68c8f20380eed7e48bcf0aab')
  }
}
```

10) db.UCOE.find()

```
{
  _id: ObjectId('68c8f20380eed7e48bcf0aaa'),
  name: 'Tej',
  address: 'Virar'
}
{
  _id: ObjectId('68c8f20380eed7e48bcf0aab'),
  name: 'Pradeep',
  address: 'Kaman'
}
```



**Department of Data Engineering**  
**Subject: Big Data Analytics Lab (CSL702)**

11) db.UCOE.update({'name':'Tej'},{\$set: {'name':'New Tej'}})

```
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

12) db.UCOE.remove({'name':'Srushti'})

```
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.
{
  acknowledged: true,
  deletedCount: 0
}
db.UCOE.find()
{
  _id: ObjectId('68c8f20380eed7e48bcf0aaa'),
  name: 'New Tej',
  address: 'Virar'
}
{
  _id: ObjectId('68c8f20380eed7e48bcf0aab'),
  name: 'Pradeep',
  address: 'Kaman'
}
```