

## Group Details

Nimish T Shah : 18BCI0197  
Siddhant Tiwary : 18BCI0181  
Deep N Golani : 18BCI0116

## Digital Assignment- I

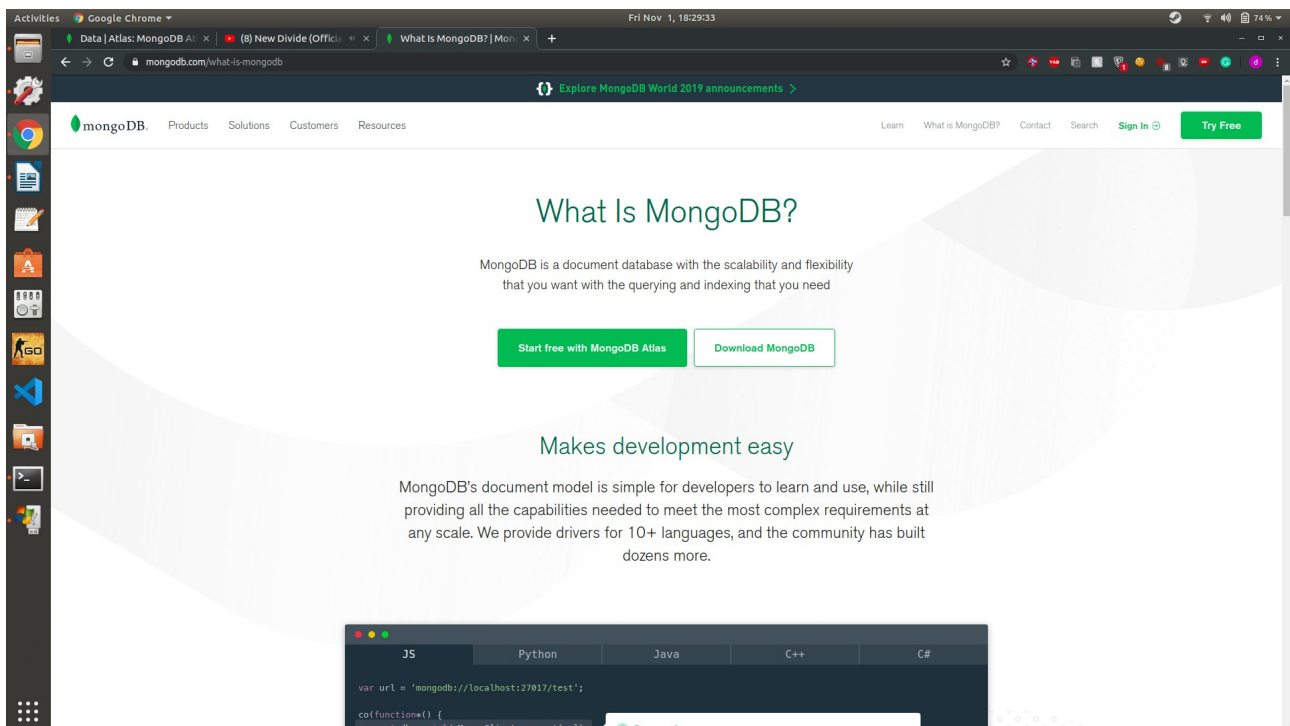
## DBMS

### Instruction Query for installation and basic command of MONGODB

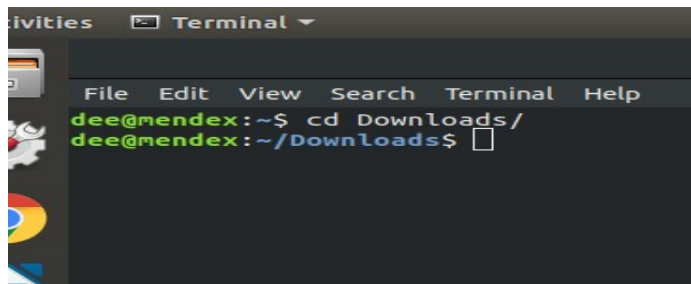
This process is only for Operating System UBUNTU 18.04 LTS

### Installation :

(1) Download a .deb file from mongodb official website.



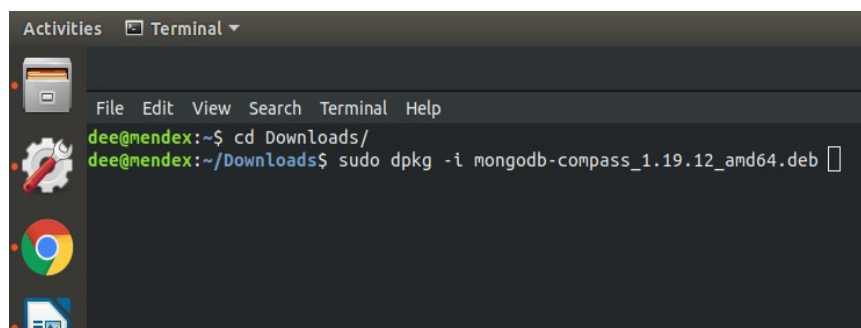
(2) Now go to terminal And change a directory to Downloads



```
File Edit View Search Terminal Help
dee@mendex:~$ cd Downloads/
dee@mendex:~/Downloads$
```

(3) Now install a Downloaded package by command

**sudo dpkg -i <package name>**

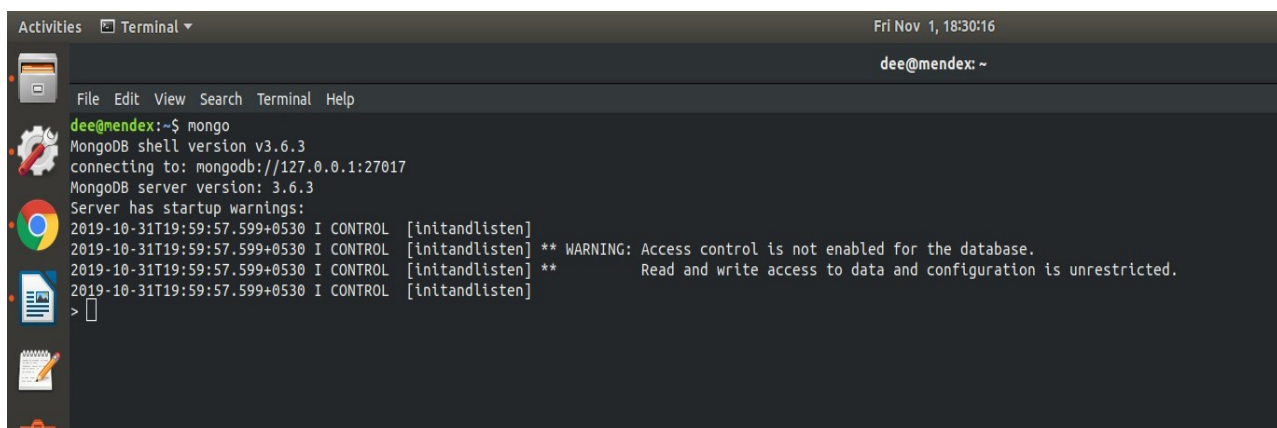


```
File Edit View Search Terminal Help
dee@mendex:~$ cd Downloads/
dee@mendex:~/Downloads$ sudo dpkg -i mongodb-compass_1.19.12_amd64.deb
```

**basic command:**

(1) now to launch a mongodb type

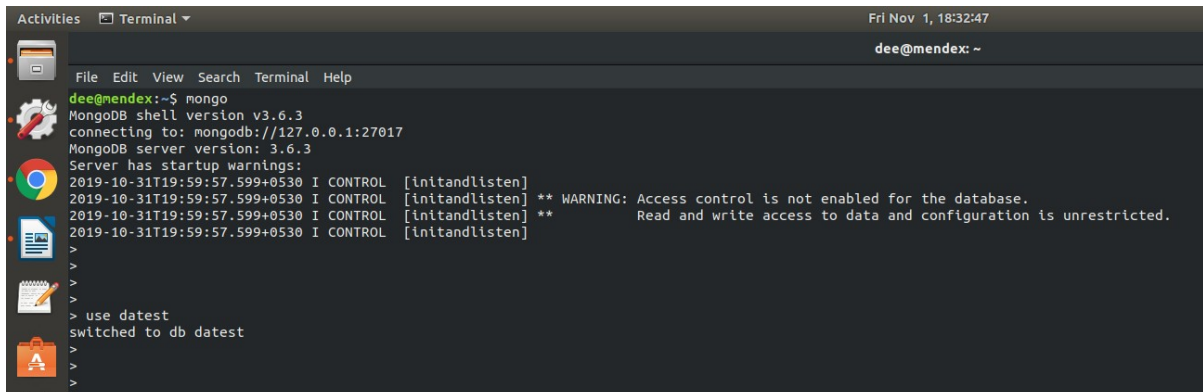
**mongo**



```
File Edit View Search Terminal Help
dee@mendex:~$ mongo
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.3
Server has startup warnings:
2019-10-31T19:59:57.599+0530 I CONTROL [initandlisten]
2019-10-31T19:59:57.599+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2019-10-31T19:59:57.599+0530 I CONTROL [initandlisten] **           Read and write access to data and configuration is unrestricted.
2019-10-31T19:59:57.599+0530 I CONTROL [initandlisten]
>
```

(2) to create a database use this command in mongodb-shell

**use <database name>**

A terminal window titled 'Terminal' with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (Fri Nov 1, 18:32:47, dee@mendex: ~). The terminal shows the command 'mongo' being executed, which starts the MongoDB shell. It displays the version (v3.6.3), connection details (localhost:27017), and server version (3.6.3). Startup warnings are shown, including a warning about access control. The user then enters 'use datest', and the shell responds 'switched to db datest'.

```
dee@mendex:~$ mongo
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.3
Server has startup warnings:
2019-10-31T19:59:57.599+0530 I CONTROL [initandlisten]
2019-10-31T19:59:57.599+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2019-10-31T19:59:57.599+0530 I CONTROL [initandlisten] **           Read and write access to data and configuration is unrestricted.
2019-10-31T19:59:57.599+0530 I CONTROL [initandlisten]
>
>
>
> use datest
switched to db datest
>
>
```

(3) to insert a data in database use this command-shell

**db.<collection name>.insert({....})**

A terminal window showing the MongoDB shell. The user enters the command 'db.datest.insert({Name:"deep",RegNo:"18BCI0116",Address:"surat"})'. The shell returns 'WriteResult({ "nInserted" : 1 })'.

```
>
>
> db.datest.insert({Name:"deep",RegNo:"18BCI0116",Address:"surat"})
WriteResult({ "nInserted" : 1 })
>
>
>
```

(4) to show the data use this command in mongodb-shell

**db.<collection name>.find()**

**or**

**db.<collection name>.find().pretty()**

```
> db.datest.insert({Name:"deep",RegNo:"18BCI0116",Address:"surat"})
WriteResult({ "nInserted" : 1 })

> db.datest.find().pretty()
{
  "_id" : ObjectId("5dbc2cf19598e6907dbd9aab"),
  "Name" : "deep",
  "RegNo" : "18BCI0116",
  "Address" : "surat"
}
```

(5) to delete the data use this command in mongodb-shell

**db.<collection name>.remove({...})**

```
> db.datest.remove({Name:"nimish"})
WriteResult({ "nRemoved" : 1 })

> db.datest.find().pretty()
{
  "_id" : ObjectId("5dbc2cf19598e6907dbd9aab"),
  "Name" : "deep",
  "RegNo" : "18BCI0116",
  "Address" : "surat"
}
```

(6) to update the data use this command in mongodb-shell

**db.<collection name>.save({...})**

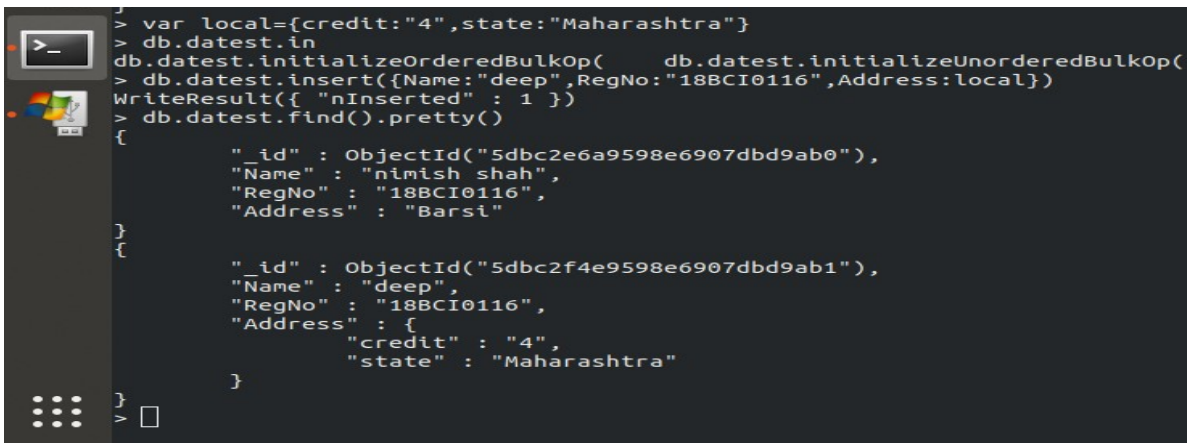
```
> db.datest.find().pretty()
{
  "_id" : ObjectId("5dbc2e6a9598e6907dbd9ab0"),
  "Name" : "deep",
  "RegNo" : "18BCI0116",
  "Address" : "surat"
}

> db.datest.save({"_id" : ObjectId("5dbc2e6a9598e6907dbd9ab0"), "Name": "nimish shah", "RegNo": "18BCI0116", "Address": "Barsi"})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.datest.find().pretty()
{
  "_id" : ObjectId("5dbc2e6a9598e6907dbd9ab0"),
  "Name" : "nimish shah",
  "RegNo" : "18BCI0116",
  "Address" : "Barsi"
}
```

(7) to use the foreign key the data use this command in mongodb-shell

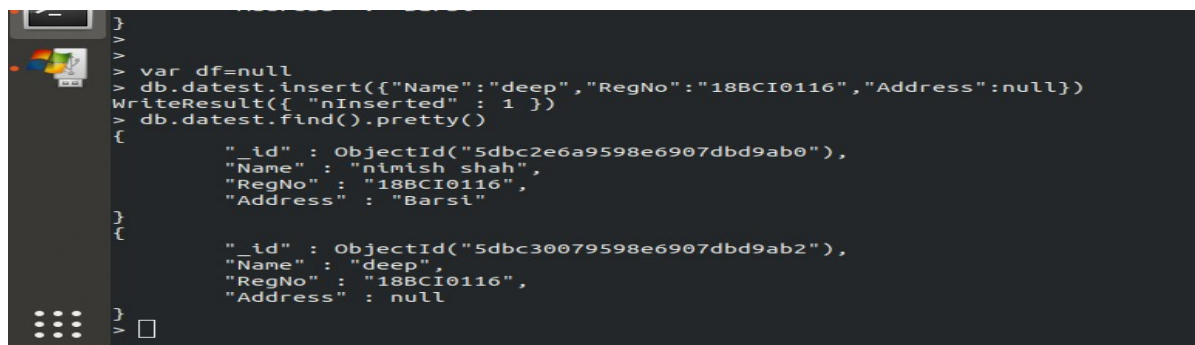
```
var <name>={...}  
db.<collection name>.insert {...})
```



```
> var local={credit:"4",state:"Maharashtra"}  
> db.datest.in  
db.datest.initializeOrderedBulkOp(    db.datest.initializeUnorderedBulkOp(  
> db.datest.insert({Name:"deep",RegNo:"18BCI0116",Address:local})  
WriteResult({ "nInserted" : 1 })  
> db.datest.find().pretty()  
{  
  "_id" : ObjectId("5dbc2e6a9598e6907dbd9ab0"),  
  "Name" : "nimish shah",  
  "RegNo" : "18BCI0116",  
  "Address" : "Barsi"  
}  
{  
  "_id" : ObjectId("5dbc2f4e9598e6907dbd9ab1"),  
  "Name" : "deep",  
  "RegNo" : "18BCI0116",  
  "Address" : {  
    "credit" : "4",  
    "state" : "Maharashtra"  
  }  
}
```

(8) to delete the data use this command in mongodb-shell

```
db.<collection name>.remove({...})
```



```
>  
>  
> var df=null  
> db.datest.insert({"Name":"deep","RegNo":"18BCI0116","Address":null})  
WriteResult({ "nInserted" : 1 })  
> db.datest.find().pretty()  
{  
  "_id" : ObjectId("5dbc2e6a9598e6907dbd9ab0"),  
  "Name" : "nimish shah",  
  "RegNo" : "18BCI0116",  
  "Address" : "Barsi"  
}  
{  
  "_id" : ObjectId("5dbc30079598e6907dbd9ab2"),  
  "Name" : "deep",  
  "RegNo" : "18BCI0116",  
  "Address" : null  
}
```

(9) to use the greater than and less than in mongodb-shell

**db.<collection name>.find({...:{\$gt:...}})**

**db.<collection name>.find({...:{\$lt:...}})**

```
>
>
>
> db.datest.find({"RegNo":{$gt:"3"}})
{ "_id" : ObjectId("5dbc3146289d2dd2ffe1fbaa"), "Name" : "bagga", "RegNo" : "5", "Address" : "shirdi" }
> db.datest.find({"RegNo":{$gt:"3"}}).pretty()
{
  "_id" : ObjectId("5dbc3146289d2dd2ffe1fbaa"),
  "Name" : "bagga",
  "RegNo" : "5",
  "Address" : "shirdi"
}
```

```
    "Name" : "bagga",
    "RegNo" : "5",
    "Address" : "shirdi"
  }
  >
  >
  >
  > db.datest.find({"RegNo":{$lt:"11"}}).pretty()
{
  "_id" : ObjectId("5dbc313d289d2dd2ffe1fba9"),
  "Name" : "soumya",
  "RegNo" : "10",
  "Address" : "alhabad"
}
```

(10) to use the or function in mongodb-shell

**db.<collection name>.find({...:{\$or:...}})**

```
>
>
>
> db.datest.find({$or:[{"RegNo":"5"}, {"Name":"soumya"}]}).pretty()
{
  "_id" : ObjectId("5dbc313d289d2dd2ffe1fba9"),
  "Name" : "soumya",
  "RegNo" : "10",
  "Address" : "alhabad"
}
{
  "_id" : ObjectId("5dbc3146289d2dd2ffe1fbaa"),
  "Name" : "bagga",
  "RegNo" : "5",
  "Address" : "shirdi"
}
> □
```

(11) to use the andfunction in mongodb-shell

**db.<collection name>.find({...},{...})**

```
Address : shirdi
}
> db.datest.find({"RegNo":"5"}, {"Name":"bagga"}).pretty()
{ "_id" : ObjectId("5dbc3146289d2dd2ffe1fbaa"), "Name" : "bagga" }
> □
```