

# MONGO & MYSQL

## MONGO

**Ques 1. Install latest version of MongoDB from apt-get repository**

**ANS 1.**

```
kaushlendra@kaushlendra:~ $ sudo apt-get install mongodb
[sudo] password for kaushlendra:
Reading package lists... Done
Building dependency tree
Reading state information... Done
mongodb is already the newest version (1:3.6.3-0ubuntu1.1).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
kaushlendra@kaushlendra:~ $
```

**Ques 2. Create a database student**

**Ans 2.**

```
> use student
switched to db student
>
```

**Ques 3. Insert operation : 5 students data (Name, Contact, City, Roll No, Branch)**

**Ans 3.**

```
> db.students.find().pretty()
{
  "_id" : ObjectId("5e48286eabec100da0870eaf"),
  "student name" : "raj",
  "contact" : "1234",
  "city" : "lucknow",
  "Roll no" : "1",
  "Branch" : "devops"
}
{
  "_id" : ObjectId("5e48286eabec100da0870eb0"),
  "student name" : "sam",
  "contact" : "234",
  "city" : "delhi",
  "Roll no" : "2",
  "Branch" : "devops"
}
{
  "_id" : ObjectId("5e48286eabec100da0870eb1"),
  "student name" : "jhon",
  "contact" : "456",
  "city" : "jaipur",
  "Roll no" : "3",
  "Branch" : "devops"
}
{
  "_id" : ObjectId("5e48286eabec100da0870eb2"),
  "student name" : "harshit",
  "contact" : "567",
  "city" : "lucknow",
  "Roll no" : "4",
  "Branch" : "devops"
}
{
  "_id" : ObjectId("5e48286eabec100da0870eb3"),
  "student name" : "rishabh",
  "contact" : "678",
  "city" : "kanpur",
```

**Ques 4. Read operation : All the students belong to a particular city**

**Ans 4.**

```
> db.students.find({}, {"student name" :1, "city" : 1, "_id" : 0})
{ "student name" : "raj", "city" : "lucknow" }
{ "student name" : "sam", "city" : "delhi" }
{ "student name" : "jhon", "city" : "jaipur" }
{ "student name" : "harshit", "city" : "lucknow" }
{ "student name" : "rishabh", "city" : "kanpur" }
> █
```

**Ques 5. Update operation : Update the branch of all the students to CSE**

**Ans 5.**

```
> db.students.update( { "student name" : "sam"}, {$set : {"Branch" : "CSE"} } )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.students.update( { "student name" : "jhon"}, {$set : {"Branch" : "CSE"} } )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.students.update( { "student name" : "harshit"}, {$set : {"Branch" : "CSE"} } )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.students.update( { "student name" : "rishabh"}, {$set : {"Branch" : "CSE"} } )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
█
```

```
> db.students.find().pretty()
{
  "_id" : ObjectId("5e48286eabec100da0870eaf"),
  "student name" : "raj",
  "contact" : "1234",
  "city" : "lucknow",
  "Roll no" : "1",
  "Branch" : "CSE"
}
{
  "_id" : ObjectId("5e48286eabec100da0870eb0"),
  "student name" : "sam",
  "contact" : "234",
  "city" : "delhi",
  "Roll no" : "2",
  "Branch" : "CSE"
}
{
  "_id" : ObjectId("5e48286eabec100da0870eb1"),
  "student name" : "jhon",
  "contact" : "456",
  "city" : "jaipur",
  "Roll no" : "3",
  "Branch" : "CSE"
}
{
  "_id" : ObjectId("5e48286eabec100da0870eb2"),
  "student name" : "harshit",
  "contact" : "567",
  "city" : "lucknow",
  "Roll no" : "4",
  "Branch" : "CSE"
}
{
  "_id" : ObjectId("5e48286eabec100da0870eb3"),
  "student name" : "rishabh",
  "contact" : "678",
  "city" : "kanpur",
  "Roll no" : "5",
  "Branch" : "CSE"
}
```

**Ques 6. Take dump of the database**

**Ans 6.**

```
kaushlendra@kaushlendra:~ $ sudo mongodump --db student -o /home/kaushlendra/tan
[sudo] password for kaushlendra:
2020-02-16T00:18:16.391+0530    writing student.students to
2020-02-16T00:18:16.393+0530    done dumping student.students (5 documents)
kaushlendra@kaushlendra:~ $ cd tan
kaushlendra@kaushlendra:tan $ ls
student
```

**Ques 7. Delete operation : Delete the record of last 2 students according to the roll number**

**Ans 7.**

```
> db.students.remove(
... {
... "Roll no" : "5"
... }
... )
WriteResult({ "nRemoved" : 1 })
> db.students.find().pretty()
{
  "_id" : ObjectId("5e48286eabec100da0870eaf"),
  "student name" : "raj",
  "contact" : "1234",
  "city" : "lucknow",
  "Roll no" : "1",
  "Branch" : "CSE"
}
{
  "_id" : ObjectId("5e48286eabec100da0870eb0"),
  "student name" : "sam",
  "contact" : "234",
  "city" : "delhi",
  "Roll no" : "2",
  "Branch" : "CSE"
}
{
  "_id" : ObjectId("5e48286eabec100da0870eb1"),
  "student name" : "jhon",
  "contact" : "456",
  "city" : "jaipur",
  "Roll no" : "3",
  "Branch" : "CSE"
}
{
  "_id" : ObjectId("5e48286eabec100da0870eb2"),
  "student name" : "harshit",
  "contact" : "567",
  "city" : "lucknow",
  "Roll no" : "4",
  "Branch" : "CSE"
}
```



```

> db.students.remove( { "Roll no" : "4" } )
WriteResult({ "nRemoved" : 1 })
> db.students.find().pretty()
{
  "_id" : ObjectId("5e48286eabec100da0870eaf"),
  "student name" : "raj",
  "contact" : "1234",
  "city" : "lucknow",
  "Roll no" : "1",
  "Branch" : "CSE"
}
{
  "_id" : ObjectId("5e48286eabec100da0870eb0"),
  "student name" : "sam",
  "contact" : "234",
  "city" : "delhi",
  "Roll no" : "2",
  "Branch" : "CSE"
}
{
  "_id" : ObjectId("5e48286eabec100da0870eb1"),
  "student name" : "jhon",
  "contact" : "456",
  "city" : "jaipur",
  "Roll no" : "3",
  "Branch" : "CSE"
}
>

```

**Ques 8. Drop the database**

**Ans 8.**

```

> db
student
> db.dropDatabase()
{ "dropped" : "student", "ok" : 1 }

```

```

> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
>

```

## Ques 9. Restore the database again to have the full data

### Ans 9.

```
kaushlendra@kaushlendra:tan $ sudo mongorestore -d student /home/kaushlendra/tan
[sudo] password for kaushlendra:
2020-02-16T11:53:28.221+0530 the --db and --collection args should only be used when restoring from a BSON file. Other uses are deprecated
and will not exist in the future; use --nsInclude instead
2020-02-16T11:53:28.221+0530 building a list of collections to restore from /home/kaushlendra/tan dir
2020-02-16T11:53:28.221+0530 don't know what to do with subdirectory "tan/student", skipping...
2020-02-16T11:53:28.221+0530 done
kaushlendra@kaushlendra:tan $
```

```
> db.students.find().pretty()
{
  "_id" : ObjectId("5e48286eabec100da0870eaf"),
  "student name" : "raj",
  "contact" : "1234",
  "city" : "lucknow",
  "Roll no" : "1",
  "Branch" : "CSE"
}
{
  "_id" : ObjectId("5e48286eabec100da0870eb0"),
  "student name" : "sam",
  "contact" : "234",
  "city" : "delhi",
  "Roll no" : "2",
  "Branch" : "CSE"
}
{
  "_id" : ObjectId("5e48286eabec100da0870eb1"),
  "student name" : "jhon",
  "contact" : "456",
  "city" : "jaipur",
  "Roll no" : "3",
  "Branch" : "CSE"
}
{
  "_id" : ObjectId("5e48286eabec100da0870eb2"),
  "student name" : "harshit",
  "contact" : "567",
  "city" : "lucknow",
  "Roll no" : "4",
  "Branch" : "CSE"
}
{
  "_id" : ObjectId("5e48286eabec100da0870eb3"),
  "student name" : "rishabh",
  "contact" : "678",
  "city" : "kanpur",
```

**Ques 10. Enable authentication on the Mongo**

**Ans 10.**



```

> use admin
switched to db admin
> db.createUser(
...   {
...     user: "myUserAdmin",
...     pwd: "abc123",
...     roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]
...   }
... )
Successfully added user: {
  "user" : "myUserAdmin",
  "roles" : [
    {
      "role" : "userAdminAnyDatabase",
      "db" : "admin"
    }
  ]
}
>

```

```

kaushlendra@kaushlendra:~ $ mongo mongodb://127.0.0.1:27009
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27009
MongoDB server version: 3.6.3
> show dbs
2020-02-16T20:44:45.907+0530 E QUERY [thread1] Error: listDatabases failed:{
  "ok" : 0,
  "errmsg" : "not authorized on admin to execute command { listDatabases:
1.0, $db: \"admin\" }",
  "code" : 13,
  "codeName" : "Unauthorized"
} :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
Mongo.prototype.getDBs@src/mongo/shell/mongo.js:65:1
shellHelper.show@src/mongo/shell/utils.js:816:19
shellHelper@src/mongo/shell/utils.js:706:15
@(shellhelp2):1:1
>

```

**Ques 11. Install another version of MongoDB from source (Version 2.6.3) and run it on port 27009**

**Ans 11.**

```
kaushlendra@kaushlendra:~ $ sudo netstat -nptl | grep mongo
tcp        0      0 127.0.0.1:27009      0.0.0.0:*            LISTEN
8011/mongod
kaushlendra@kaushlendra:~ $
```

## MYSQL

**Ques 1. Install latest version of MYSQL from apt-get repository**

**Ans 1.**

```
kaushlendra@kaushlendra:~ $ sudo apt-get install mysql-server
[sudo] password for kaushlendra:
Reading package lists... Done
Building dependency tree
Reading state information... Done
mysql-server is already the newest version (5.7.29-0ubuntu0.18.04.1).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
kaushlendra@kaushlendra:~ $
```

**Ques 2. Create a database student**

**Ans 2.**

```
mysql> create database student;
Query OK, 1 row affected (0.00 sec)

mysql> use student;
Database changed
mysql>
```

**Ques 3. Insert operation : 5 students data (Name, Contact, City, Roll No, Branch)**

**Ans 3.**

```
mysql> select * from students;
+-----+-----+-----+-----+-----+
| student_name | contact | city    | roll_no | branch |
+-----+-----+-----+-----+-----+
| raj          | 345     | lucknow | 1        | devops |
| sam          | 3465    | chinnai | 2        | devops |
| harshit     | 34767   | jaipur  | 3        | devops |
| rahul        | 34767   | kanpur  | 4        | devops |
| pankaj       | 9876    | delhi   | 5        | devops |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

**Ques 4. Read operation : All the students belong to a particular city**

**Ans 4.**

```
mysql> mysql> select city,student_name from students;
+-----+-----+
| city   | student_name |
+-----+-----+
| lucknow | raj          |
| chinnai | sam          |
| jaipur  | harshit      |
| kanpur  | rahul        |
| delhi   | pankaj       |
+-----+-----+
5 rows in set (0.00 sec)

mysql> █
```

**Ques 5. Update operation : Update the branch of all the students to CSE**

**Ans 5.**

```
mysql> UPDATE students SET branch='CSE' WHERE branch='devops';
Query OK, 5 rows affected (0.01 sec)
Rows matched: 5  Changed: 5  Warnings: 0

mysql> select * from students;
+-----+-----+-----+-----+-----+
| student_name | contact | city   | roll_no | branch |
+-----+-----+-----+-----+-----+
| raj          | 345     | lucknow | 1       | CSE    |
| sam          | 3465    | chinnai | 2       | CSE    |
| harshit      | 34767   | jaipur  | 3       | CSE    |
| rahul        | 34767   | kanpur  | 4       | CSE    |
| pankaj       | 9876    | delhi   | 5       | CSE    |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> 
```

**Ques 6. Take dump of the database**

**Ans 6.**

```
kaushlendra@kaushlendra:~$ sudo mysqldump student > backup.sql
kaushlendra@kaushlendra:~$ ls
1280px-Monasterio_Khor_Virap,_Armenia,_2016-10-01,_DD_25.jpg Desktop exercise images output.txt Public Videos
backup.sql Documents gang Music pan tan wget
bootcamp Downloads https: new.txt Pictures Templates
kaushlendra@kaushlendra:~$ 
```

**Ques 7. Delete operation : Delete the record of last 2 students according to the roll number**

**Ans 7.**



```
mysql> DELETE FROM students WHERE roll_no='5';
Query OK, 1 row affected (0.01 sec)

mysql> DELETE FROM students WHERE roll_no='4';
Query OK, 1 row affected (0.01 sec)

mysql> select * from students;
+-----+-----+-----+-----+-----+
| student_name | contact | city    | roll_no | branch |
+-----+-----+-----+-----+-----+
| raj          | 345     | lucknow | 1       | CSE    |
| sam          | 3465    | chinnai | 2       | CSE    |
| harshit     | 34767   | jaipur  | 3       | CSE    |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> █
```

**Ques 8. Drop the database**

**Ans 8 .**

```
mysql> drop database student;
Query OK, 0 rows affected (0.00 sec)

mysql> use student;
ERROR 1049 (42000): Unknown database 'student'
mysql> █
```

**Ques 9. Restore the database again to have the full data**

**Ans 9.**

```
kaushlendra@kaushlendra:~ $ sudo mysql student < backup.sql
[sudo] password for kaushlendra:
kaushlendra@kaushlendra:~ $ █
```

```
mysql> select * from students;
+-----+-----+-----+-----+-----+
| student_name | contact | city    | roll_no | branch |
+-----+-----+-----+-----+-----+
| raj          | 345     | lucknow | 1        | devops |
| sam          | 3465    | chinnai | 2        | devops |
| harshit      | 34767   | jaipur  | 3        | devops |
| rahul        | 34767   | kanpur  | 4        | devops |
| pankaj       | 9876    | delhi   | 5        | devops |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> █
```

**Ques 10. Enable authentication on the Mongo**

**Ans 10.**

**Ques 11. Install another version of MongoDB from source (Version 2.6.3) and run it on port 27009**

**Ans 11.**

**Ques 12. Create init service of Mongo installed later\***

**Ans 12 .**