| **Description** | **Command** |
| --- | --- |
| show all database | show dbs |
| show current database | db.getName() |
| switch(and create if doesn’t exit) to database ‘userdb’ | use usersdb |
| drop current database | db.dropDatabase() |

### Database

### Collection

| **Description** | **Command** |
| --- | --- |
| show all the collection under current database | db.getCollectionNames() |
| create collection ‘users’ | db.createCollection("users") |
| drop collection ‘users’ | db.users.drop() |

**Document: CRUD**

| **Description** | **Command** |
| --- | --- |
| insert document into ‘users’ collection | db.users.insert({  userid: "123",  age: 18,  name: "vikash"  }); |
| find all documents in ‘users’ collection | db.users.find() |
| find all documents from collection ‘users’ but select field ‘\_id’, ‘name’ & ‘age’ only | db.users.find(  {},  {  name: 1,  age: 1  }  ) |
| find all documents from collection ‘users’ but select field ‘name’ & ‘age’ only. By default ‘\_id’ field is always selected, so to remove it. | db.users.find(  {},  {  \_id: 0,  name: 1,  age: 1  }  ) |
| find all documents from collection ‘users’ where ‘name’=‘vikash’ | db.users.find({  name: "vikash"  }) |
| find all documents from collection ‘users’ where ‘name’=‘vikash’ and select field ‘userid’ only | db.users.find(  {  name: "vikash"  },  {  \_id: 0  userid: 1  }  ) |
| find all documents from collection ‘users’ where ‘name’!=‘vikash’ | db.users.find({  name: {  $ne: "vikash"  }  }) |
| find all documents from collection ‘users’ where ‘name’=‘vikash’ and age=18 | db.users.find({  name: "vikash",  age: 18  }) |
| find all documents from collection ‘users’ where ‘name’=‘vikash’ or age=18 | db.users.find({  $or: [  {  name: "vikash"  },  {  age: 18  }  ]  }) |
| find all documents from collection ‘users’ where ‘age > 18’ | db.users.find({  age: {  $gt: 18  }  }) |
| find all documents from collection ‘users’ where ‘age < 18’ | db.users.find({  age: {  $lt: 18  }  }) |
| find all documents from collection ‘users’ where ‘age >= 18’ | db.users.find({  age: {  $gte: 18  }  }) |
| find all documents from collection ‘users’ where ‘name is like %ind%’ | db.users.find({  name: /ind/  }) |
| find all documents from collection ‘users’ where ‘name is like ind%’ | db.users.find({  name: /^ind/  }) |
| find all documents from collection ‘users’ where ‘name is like %ind%’ and order(ASC) by field ‘age’ | db.users.find({  name: /ind/  })  .sort({  age: 1  }) |
| find all documents from collection ‘users’ where ‘name is like %ind%’ and order(DESC) by field ‘age’ | db.users.find({  name: /ind/  })  .sort({  age: -1  }) |
| find the number of documents in collection ‘users’ | db.users.find().count() |
| find the number of documents in collection ‘users’ where field ‘name’ exist | db.users  .find({  name: {  $exists: **true**  }  })  .count() |
| show distinct value for field ‘name’ of collection ‘users’ | db.users.distinct('name') |
| fetch 2 document skipping first 5 documents from collection ‘users’ | db.users.find().limit(2).skip(5) |
| updated field ‘age’ to 19 of collection ‘users’ where name = ‘vikash’ | db.users.update(  {  name: "vikash"  },  {  $set: {age: 19}  },  {  multi: **true**  }  ) |
| increase current value of field ‘age’ by 5 of collection ‘users’ where name = ‘vikash’ | db.users.update(  {  name: "vikash"  },  {  $inc: {age: 5}  },  {  multi: **true**  }  ) |
| delete all documents from ‘users’ collection | db.users.remove({}) |
| delete all the documents of collection ‘users’ where name = ‘vikash’ | db.users.remove(  {  name: "vikash"  }  ) |