MongoDB

```
db.posts.insertMany([
  {
    title: 'Post Two',
    body: 'Body of post two',
    category: 'Technology',
    date: Date()
  },
  {
    title: 'Post Three',
    body: 'Body of post three',
    category: 'News',
    date: Date()
  },
  {
    title: 'Post Four',
    body: 'Body of post three',
    category: 'Entertainment',
    date: Date()
  }
 ])
```

Login into the database with username and password:

mongo -u USERNAME -p PASSWORD --authenticationDatabase DATABASENAME

start/stop

sudo service mongod start

sudo service mongod stop

basic commands

db	Show name of current database
mongod	Start database
mongo	Connect to database
show dbs	Show databases
use db	Switch to database db
show collections	Display current database collections

create

insert(data)	insert document(s) returns write result
insertOne (data, options)	insert one document
insertMany(data, options)	insert many documents
insertMany([{},{},{}])	needs square brackets

update

updateOne(filter, data, options)	Change one document
updateMany(filter, data, options)	Change many documents
replaceOne(filter, data, options)	Replace document entirely

read

db.collection.find()	Display documents from collection
find(filter, options)	find all matching documents
findOne(filter, options)	find first matching document

delete

deleteOne(filter, options)	Delete one document
deleteMany(filter, options)	Delete many documents

delete row

```
db.posts.remove({ title: 'Post Four' })
```

filters 1

{"key": "value"}	Used for filter arguments to filter collection
{key: {$operator: value} }	Operators for querying data
{key: {$exists: true}}	Matches all documents containing subdocument key
$eq	Matches values that are equal to a specified value.
$gt	Matches values that are greater than a specified value.
$gte	Matches values that are greater than or equal to a specified value.
$in	Matches any of the values specified in an array
syntax:	{key:{$in: [array of values] } }

filters 2

$lt	Matches values that are less than a specified value.
$lte	Matches values that are less than or equal to a specified value.
$ne	Matches all values that are not equal to a specified value.
$nin	Matches none of the values specified in an array.
$and	Performs AND operation
syntax:	{$and: [{},{}] }
{key: {$op: filter}, {filter}}	$and operator is necessary when the same field or operator has to be specified in multiple expressions
find({doc.subdoc: value})	

Insert Row

```
db.posts.insert({
  title: 'Post One',
  body: 'Body of post one',
  category: 'News',
  tags: ['news', 'events'],
  user: {
    name: 'John Doe',
    status: 'author'
  },
  date: Date()
 })
```

Insert Multiple Rows

```
db.posts.insertMany([
  {
    title: 'Post Two',
    body: 'Body of post two',
    category: 'Technology',
    date: Date()
  },
  {
    title: 'Post Three',
    body: 'Body of post three',
    category: 'News',
    date: Date()
  },
  {
    title: 'Post Four',
    body: 'Body of post three',
    category: 'Entertainment',
    date: Date()
  }
 ])
```

Sort Rows

```
# asc
db.posts.find().sort({
title: 1 }).pretty()
# desc
 db.posts.find().sort({
 title: -1 }).pretty()
```

Count Rows

```
db.posts.find().count()
 db.posts.find({
 category: 'news'
 }).count()
```

Limit Rows

```
db.posts.find().limit(2).pretty()
```

Chaining

```
db.posts.find().limit(2).sort({ title: 1 }).pretty()
```

Foreach

```
db.posts.find().forEach(function(doc) {
  print("Blog Post: " + doc.title)
  })
```

Find One Row

```
db.posts.findOne({ category: 'News' })
```

Find Specific Fields

```
db.posts.find({ title: 'Post One' }, {
  title: 1,
  author: 1
  })
```

Update Row

```
db.posts.update({ title: 'Post
Two' },
{
  title: 'Post Two',
  body: 'New body for post 2',
  date: Date()
},
{
  upsert: true
  })
```

Update Specific Field

```
db.posts.update({ title: 'Post Two' },
{
  $set: {
    body: 'Body for post 2',
    category: 'Technology'
  }
  })
```

Increment Field ($inc)

```
db.posts.update({ title: 'Post Two' },
{
  $inc: {
    likes: 5
  }
  })
```

Rename Field

```
db.posts.update({ title: 'Post Two' },
{
  $rename: {
    likes: 'views'
  }
  })
```

Sub-Documents

```
db.posts.update({ title: 'Post One' },
{
  $set: {
    comments: [
      {
        body: 'Comment One',
        user: 'Mary Williams',
        date: Date()
      },
      {
        body: 'Comment Two',
        user: 'Harry White',
        date: Date()
      }
    ]
  }
  })
```

Find By Element in Array ($elemMatch)

```
db.posts.find({
  comments: {
     $elemMatch: {
       user: 'Mary Williams'
       }
    }
  }
 )
```

Text Search

```
db.posts.find({
  $text: {
    $search: "\"Post O\""
    }
 })
```

Greater & Less Than

```
db.posts.find({ views: { $gt: 2 } })
db.posts.find({ views: { $gte: 7 } })
db.posts.find({ views: { $lt: 7 } })
db.posts.find({ views: { $lte: 7 } })
```