```
switched to db mydb
>use mydb
```

```
>show dbs
```

```
>db.movie.insert({"name":"tutorials point"})
```

```
>db.COLLECTION_NAME.drop()
```

```
db.createCollection(name, options)
```

```
db.sabpaisa.find()
```

```
>db.COLLECTION_NAME.insertOne(document)
```

```
>db.COLLECTION_NAME.find()
```

```
>db.COLLECTION_NAME.find().pretty()
```

```
>db.COLLECTIONNAME.findOne()
```

- To query documents based on the AND condition, you need to use $and keyword. Following is the basic syntax of **AND –**

  ```
  >db.mycol.find({ $and: [ {<key1>:<value1>}, { <key2>:<value2>} ] })
  ```

- To query documents based on the OR condition, you need to use **$or** keyword. Following is the basic syntax of **OR** –

  ```
  >db.mycol.find(
    {
      $or: [
        {key1: value1}, {key2:value2}
      ]
    }
  ).pretty()
  ```

- The following example will show the documents that have likes greater than 10 and whose title is either 'MongoDB Overview' or by is 'tutorials point'. Equivalent SQL where clause is **'where likes>10 AND (by = 'tutorials point' OR title = 'MongoDB Overview')'**

```
>db.mycol.find({"likes": {$gt:10}, $or: [{"by": "tutorials point"},{"title": "MongoDB
Overview"}]}).pretty()
```

- The update() method updates the values in the existing document.
    - db.sabpaisa.update({"name":"sonu"},{$set:{"name":"kr sonu"}})

```
>db.COLLECTION_NAME.update(SELECTION_CRITERIA, UPDATED_DATA)
```

- By default, MongoDB will update only a single document. To update multiple documents, you need to set a parameter 'multi' to true

```
>db.mycol.update({'title':'MongoDB Overview'},{$set:{'title':'New MongoDB
Tutorial'}},{multi:true})
```

- db.sabpaisa.update({"_id": ObjectId("6332a3245938c86bcc5b1004")}, {$set:{"date_of_resignation":"working"}})

- db.sabpaisa.deleteOne({ _id: ObjectId("6332c01c5938c86bcc5b1005")})

- The basic syntax of **createIndex()** method is as follows().

```
>db.COLLECTION_NAME.createIndex({KEY:1})
```