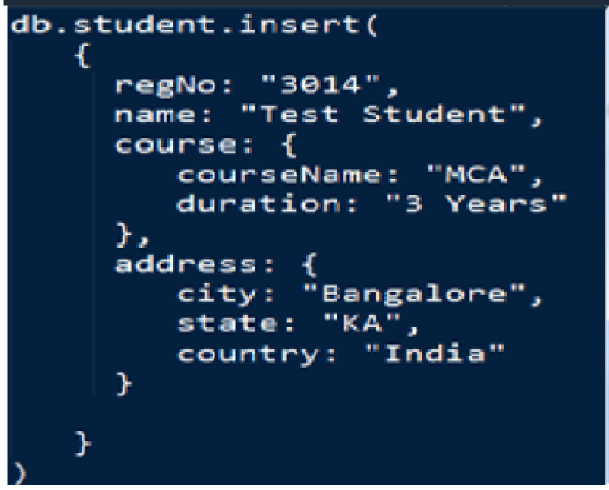
Inserting a document into a collection (Create)

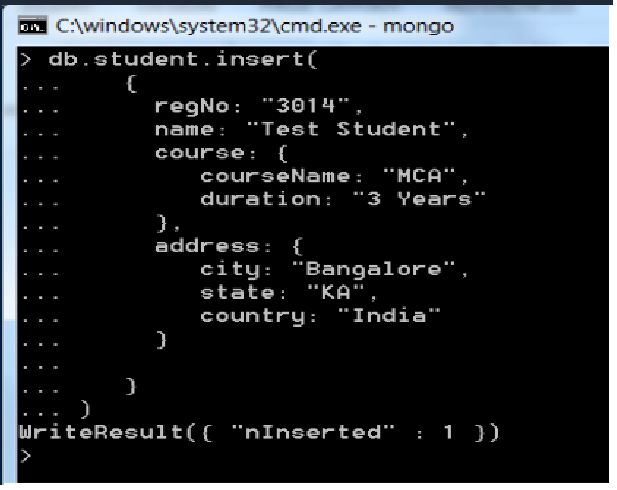
The command `db.collection.insert()` will perform an insert operation into a collection of a document.

Let us insert a document to a `student` collection. You must be connected to a database for doing any insert. It is done as follows:

```
db.student.insert({

          regNo: "3014",

          name: "Test Student",

          course: {

                    courseName: "MCA",

                    duration: "3 Years"

          },

          address: {

                    city: "Bangalore",

                    state: "KA",

                    country: "India"

          }

})
```



```
db.student.insert(
    {
        regNo: "3014",
        name: "Test Student",
        course: {
            courseName: "MCA",
            duration: "3 Years"
        },
        address: {
            city: "Bangalore",
            state: "KA",
            country: "India"
        }

    }
)
```



```
> db.student.insert(
...     {
...         regNo: "3014",
...         name: "Test Student",
...         course: {
...             courseName: "MCA",
...             duration: "3 Years"
...         },
...         address: {
...             city: "Bangalore",
...             state: "KA",
...             country: "India"
...         }
...
...     }
... )
WriteResult({ "nInserted" : 1 })
>
```

Note that an entry has been made into the collection called student.

Add MongoDB Array using insert()

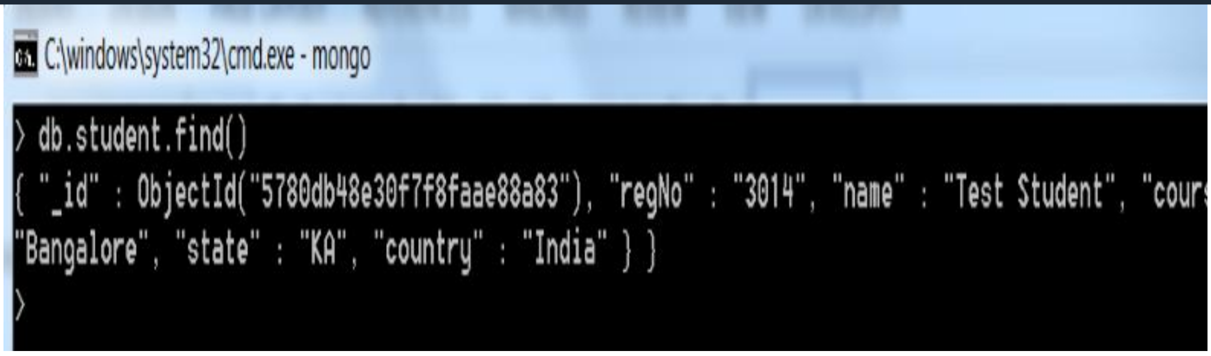
```
var myEmployee=
        [

                {
                        "Employeeid" : 1,
                        "EmployeeName" : "Smith"
                },
                {
                        "Employeeid"   : 2,
                        "EmployeeName" : "Mohan"
                },
                {
                        "Employeeid"   : 3,
                        "EmployeeName" : "Joe"
                },

        ];

        db.Employee.insert(myEmployee);
```

Querying a document from a collection (Read)

To retrieve (Select) the inserted document, run the below command.

The `find()` command will retrieve all the documents of the given collection.

```
db.collection_name.find()
```



```
C:\windows\system32\cmd.exe - mongo
> db.student.find()
{ "_id" : ObjectId("5780db48e30f7f8faae88a83"), "regNo" : "3014", "name" : "Test Student", "cours
"Bangalore", "state" : "KA", "country" : "India" } }
>
```

NOTE : Please observe that the record retrieved contains an attribute called `_id` with some unique identifier value called **ObjectId** which acts as a document identifier.

If a record is to be retrieved based on some criteria, the `find()` method should be called passing parameters, then the record will be retrieved based on the attributes specified.

```
db.collection_name.find({"fieldname":"value"})
```

For Example : Let us retrieve the record from the **student** collection where the attribute **regNo** is **3014**and the query for the same is as shown below:

```
db.students.find({"regNo":"3014"})
```

Printing in JSON format

```
db.Employee.find().forEach(printjson)

db.Employee.find().limit(2).forEach(printjson);

db.Employee.find().sort({Employeeid:-1}).forEach(printjson) //descending order
db.userdetails.find({"education":"M.C.A."},{"user_id" : 1,"password":1,"date_of_join":1
,_id:0}).pretty();//tofetch specific fields only
```

Apart from find() method, there is **findOne()** method, that returns only one document.

Operation	Syntax	Example	RDBMS Equivalent
Equality	{<key>:<value>}	db.student.find({"by":"Asreet"}).pretty()	where by = 'Asreet'
Less Than	{<key>:{$lt:<value>}}	db.student.find({"marks":{$lt:50}}).pretty()	where marks < 50
Less Than Equals	{<key>:{$lte:<value>}}	db.student.find({"marks":{$lte:50}}).pretty()	where marks <= 50
Greater Than	{<key>:{$gt:<value>}}	db.student.find({"marks":{$gt:50}}).pretty()	where marks > 50
Greater Than Equals	{<key>:{$gte:<value>}}	db.student.find({"marks":{$gte:50}}).pretty()	where marks >= 50
Not Equals	{<key>:{$ne:<value>}}	db.student.find({"marks":{$ne:50}}).pretty()	where marks != 50