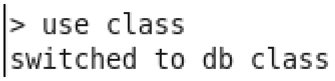
MongoDB

Use MongoDB command line for all the assignments.

1. Create a database called *class*.

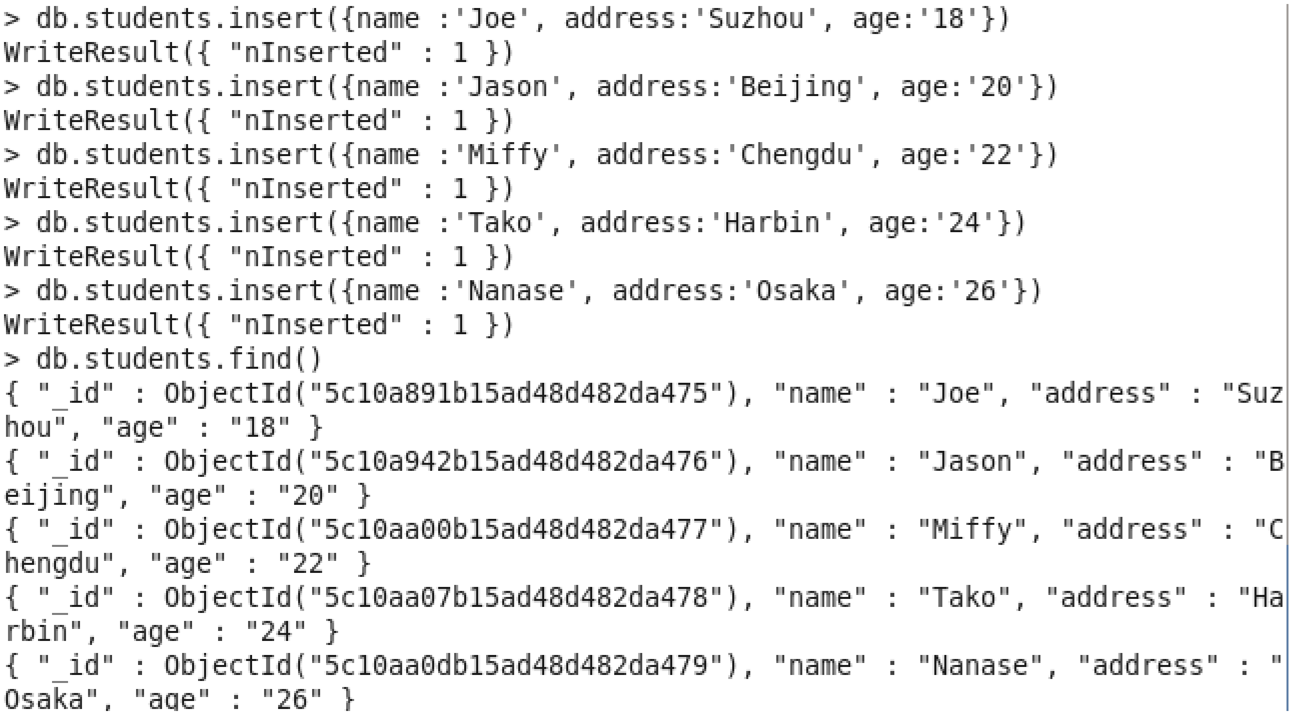
   $ sudo service mongod start
   $Mongo // start MongoDB and Login
   Use class // create a database called class



```
> use class
switched to db class
```

2. Create a collection called *students* and Insert few students with their *name, address* and *age*

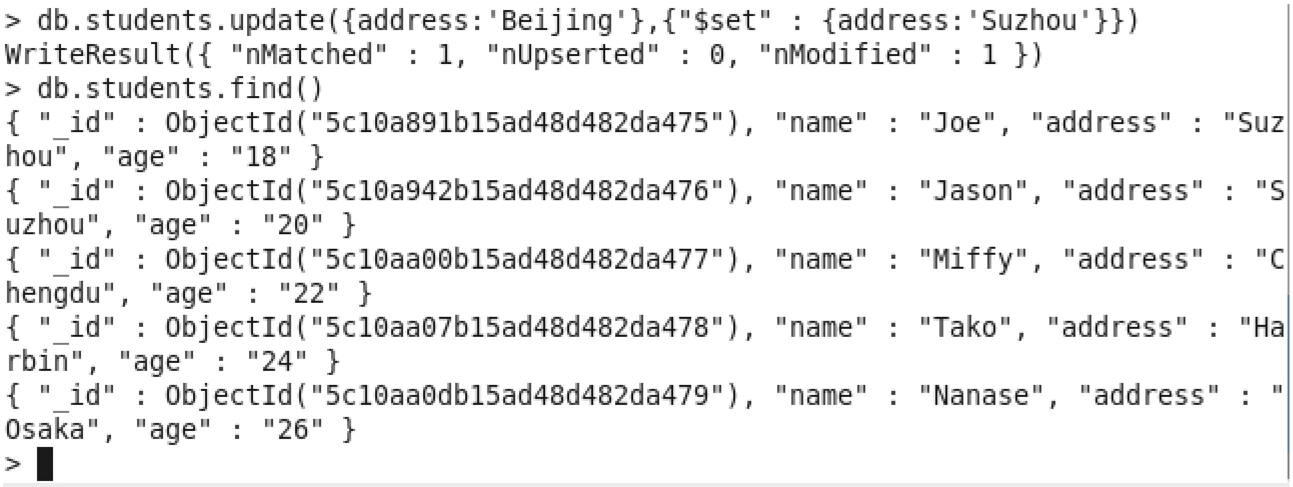
   db.createCollection("students") // create a collection called students
   db.students.insert({name :'Joe', address:'Suzhou', age:'18'})
   db.students.insert({name :'Jason', address:'Beijing', age:'20'})
   db.students.insert({name :'Miffy', address:'Chengdu', age:'22'})
   db.students.insert({name :'Tako', address:'Harbin', age:'24'})
   db.students.insert({name :'Nanase', address:'Osaka', age:'26'}) // insert five students with details



```
> db.students.insert({name :'Joe', address:'Suzhou', age:'18'})
WriteResult({ "nInserted" : 1 })
> db.students.insert({name :'Jason', address:'Beijing', age:'20'})
WriteResult({ "nInserted" : 1 })
> db.students.insert({name :'Miffy', address:'Chengdu', age:'22'})
WriteResult({ "nInserted" : 1 })
> db.students.insert({name :'Tako', address:'Harbin', age:'24'})
WriteResult({ "nInserted" : 1 })
> db.students.insert({name :'Nanase', address:'Osaka', age:'26'})
WriteResult({ "nInserted" : 1 })
> db.students.find()
{ "_id" : ObjectId("5c10a891b15ad48d482da475"), "name" : "Joe", "address" : "Suz
hou", "age" : "18" }
{ "_id" : ObjectId("5c10a942b15ad48d482da476"), "name" : "Jason", "address" : "B
eijing", "age" : "20" }
{ "_id" : ObjectId("5c10aa00b15ad48d482da477"), "name" : "Miffy", "address" : "C
hengdu", "age" : "22" }
{ "_id" : ObjectId("5c10aa07b15ad48d482da478"), "name" : "Tako", "address" : "Ha
rbin", "age" : "24" }
{ "_id" : ObjectId("5c10aa0db15ad48d482da479"), "name" : "Nanase", "address" : "
Osaka", "age" : "26" }
```

3. Find students with a given *address*. Ex: Find all student whose *address* is Beijing then update the students address to Suzhou.

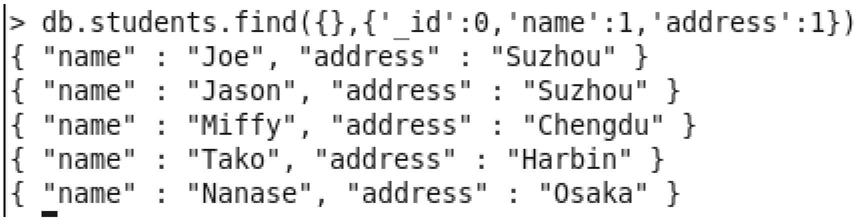
   db.students.update({address:'Beijing'},{"$set" : {address:'Suzhou'}})



```
> db.students.update({address:'Beijing'},{"$set" : {address:'Suzhou'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.students.find()
{ "_id" : ObjectId("5c10a891b15ad48d482da475"), "name" : "Joe", "address" : "Suz
hou", "age" : "18" }
{ "_id" : ObjectId("5c10a942b15ad48d482da476"), "name" : "Jason", "address" : "S
uzhou", "age" : "20" }
{ "_id" : ObjectId("5c10aa00b15ad48d482da477"), "name" : "Miffy", "address" : "C
hengdu", "age" : "22" }
{ "_id" : ObjectId("5c10aa07b15ad48d482da478"), "name" : "Tako", "address" : "Ha
rbin", "age" : "24" }
{ "_id" : ObjectId("5c10aa0db15ad48d482da479"), "name" : "Nanase", "address" : "
Osaka", "age" : "26" }
> 
```

4. Print all students' *name* and *address* but not *age* and *_id*.

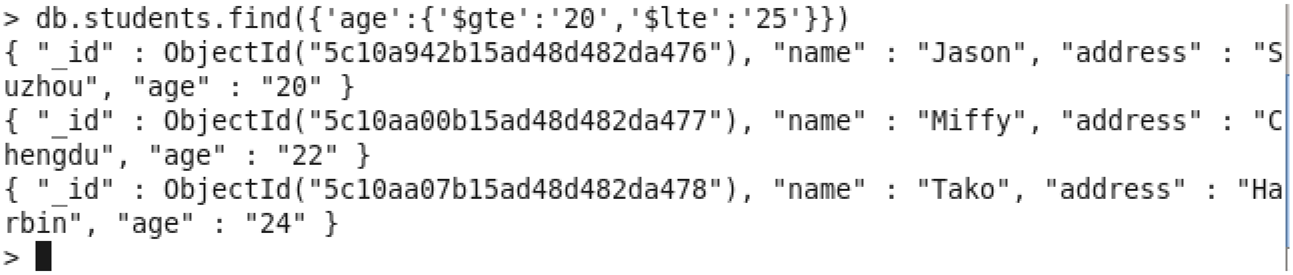
   db.students.find({},{'_id':0,'name':1,'address':1})



```
> db.students.find({},{'_id':0,'name':1,'address':1})
{ "name" : "Joe", "address" : "Suzhou" }
{ "name" : "Jason", "address" : "Suzhou" }
{ "name" : "Miffy", "address" : "Chengdu" }
{ "name" : "Tako", "address" : "Harbin" }
{ "name" : "Nanase", "address" : "Osaka" }
```

5. Find all the students *age* between age 20 to 25.

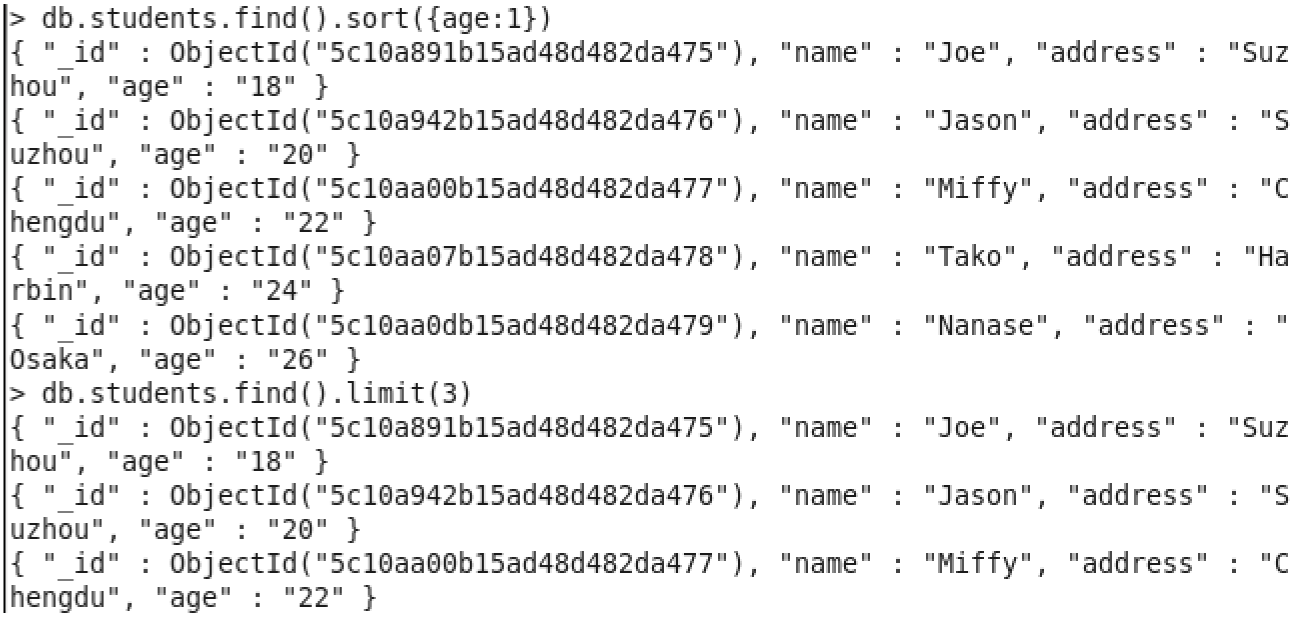
   db.students.find({'age':{'$gte':'20','$lte':'25'}})



```
> db.students.find({'age':{'$gte':'20','$lte':'25'}})
{ "_id" : ObjectId("5c10a942b15ad48d482da476"), "name" : "Jason", "address" : "S
uzhou", "age" : "20" }
{ "_id" : ObjectId("5c10aa00b15ad48d482da477"), "name" : "Miffy", "address" : "C
hengdu", "age" : "22" }
{ "_id" : ObjectId("5c10aa07b15ad48d482da478"), "name" : "Tako", "address" : "Ha
rbin", "age" : "24" }
> 
```

6. Sort the students by the age (low to high) and print first 3 students.

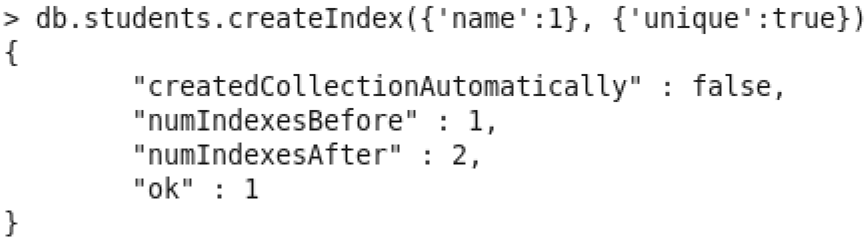
   db.students.find().sort({age:1}) // sort the students by the age (low to high)
   db.students.find().limit(3) // print first 3 students



```
> db.students.find().sort({age:1})
{ "_id" : ObjectId("5c10a891b15ad48d482da475"), "name" : "Joe", "address" : "Suz
hou", "age" : "18" }
{ "_id" : ObjectId("5c10a942b15ad48d482da476"), "name" : "Jason", "address" : "S
uzhou", "age" : "20" }
{ "_id" : ObjectId("5c10aa00b15ad48d482da477"), "name" : "Miffy", "address" : "C
hengdu", "age" : "22" }
{ "_id" : ObjectId("5c10aa07b15ad48d482da478"), "name" : "Tako", "address" : "Ha
rbin", "age" : "24" }
{ "_id" : ObjectId("5c10aa0db15ad48d482da479"), "name" : "Nanase", "address" : "
Osaka", "age" : "26" }
> db.students.find().limit(3)
{ "_id" : ObjectId("5c10a891b15ad48d482da475"), "name" : "Joe", "address" : "Suz
hou", "age" : "18" }
{ "_id" : ObjectId("5c10a942b15ad48d482da476"), "name" : "Jason", "address" : "S
uzhou", "age" : "20" }
{ "_id" : ObjectId("5c10aa00b15ad48d482da477"), "name" : "Miffy", "address" : "C
hengdu", "age" : "22" }
```

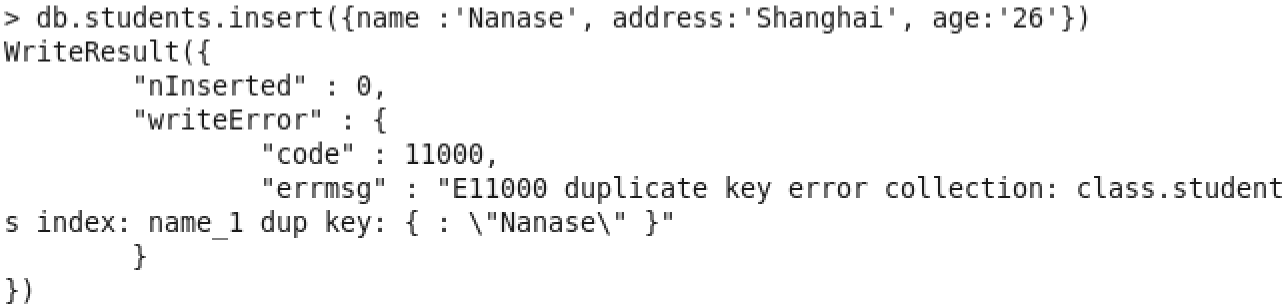
7. Update the *students* collection such a way that same student *name* can not be inserted twice in two records (i.e. ensure student name is the unique in the collection)

   db.students.createIndex({'name':1}, {'unique':true})



```
> db.students.createIndex({'name':1}, {'unique':true})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
```

   Check:
   db.students.insert({name :'Nanase', address:'Shanghai', age:'26'})
   // insert a student with the same name as the last student inserted



```
> db.students.insert({name :'Nanase', address:'Shanghai', age:'26'})
WriteResult({
        "nInserted" : 0,
        "writeError" : {
                "code" : 11000,
                "errmsg" : "E11000 duplicate key error collection: class.student
s index: name_1 dup key: { : \"Nanase\" }"
        }
})
```

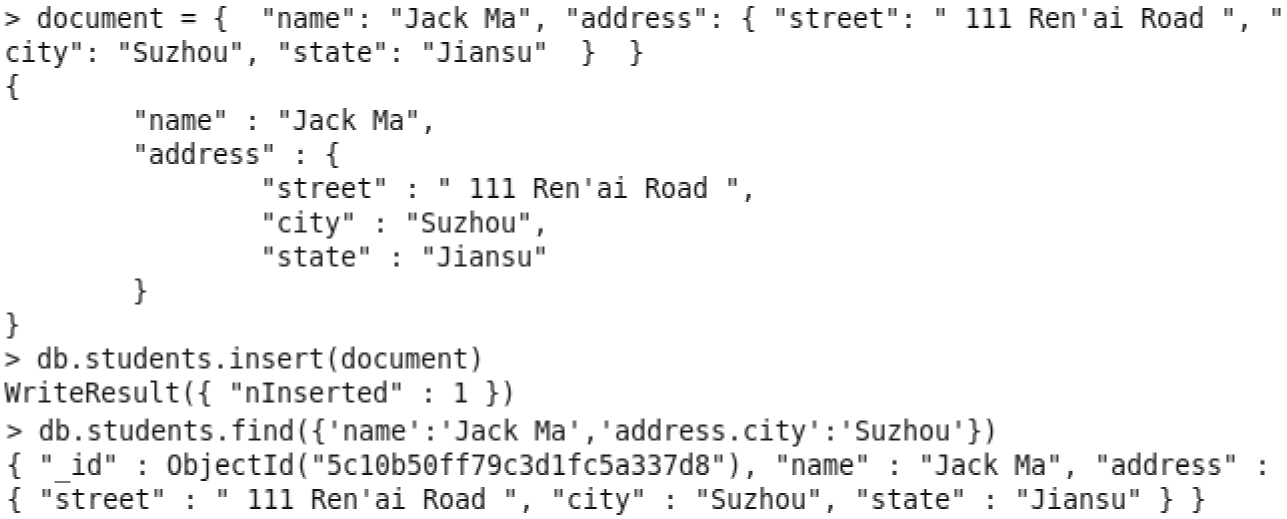
8. Consider a student Jack Ma’s address is an embedded document as shown below:

```
{
"name": "Jack Ma",
"address": {
"street": " 111 Ren'ai Road ",
"city": "Suzhou",
"state": "Jiansu"
}
}
```

Insert this document into MongoDB.

Write a query to search for all students who lives in the city *Suzhou* and name is *Jack Ma*. Remember *city* field is inside address document.

```
document = { "name": "Jack Ma", "address": { "street": " 111 Ren'ai Road ", "city":
"Suzhou", "state": "Jiansu"  }} // define the document
db.students.insert(document) // insert
db.students.find({'name':'Jack Ma','address.city':'Suzhou'})
//query with combine inside field
```



```
> document = {  "name": "Jack Ma", "address": { "street": " 111 Ren'ai Road ", "
city": "Suzhou", "state": "Jiansu"  }  }
{
        "name" : "Jack Ma",
        "address" : {
                "street" : " 111 Ren'ai Road ",
                "city" : "Suzhou",
                "state" : "Jiansu"
        }
}
> db.students.insert(document)
WriteResult({ "nInserted" : 1 })
> db.students.find({'name':'Jack Ma','address.city':'Suzhou'})
{ "_id" : ObjectId("5c10b50ff79c3d1fc5a337d8"), "name" : "Jack Ma", "address" :
{ "street" : " 111 Ren'ai Road ", "city" : "Suzhou", "state" : "Jiansu" } }
```