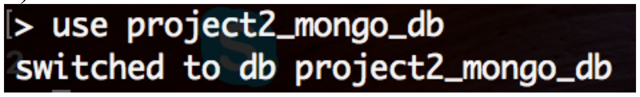
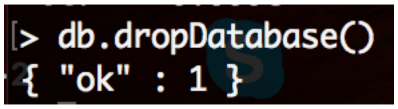
1) Create a database



```
> use project2_mongo_db
switched to db project2_mongo_db
```

(If you use the *show dbs* command again, you will see that the database hasn't been created yet.)
I create a database by a use command. If there's no same name database, it will create one.

2) Drop a database



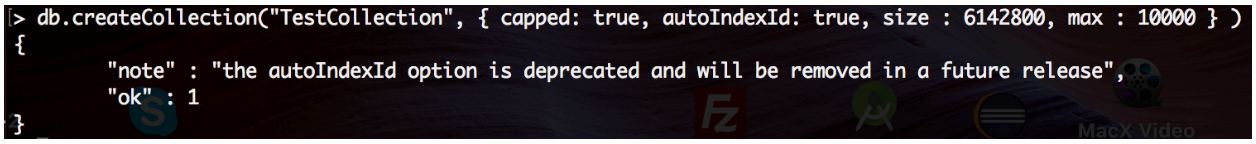
```
> db.dropDatabase()
{ "ok" : 1 }
```

I drop a database by doing the command with a function, dropDatabase()

- A **database** holds a set of collections
- A **collection** holds a set of documents (table)
- A **document** is a set of fields
- A **field** is a key-value pair
- A **key** is a name (string)
- A **value** is a - basic type like string, integer, float, timestamp, binary, etc.,

```
db.createCollection(<name>, { capped: <boolean>,
                              autoIndexId: <boolean>,
                              size: <number>,
                              max: <number>,
                              storageEngine: <document>,
                              validator: <document>,
                              validationLevel: <string>,
                              validationAction: <string>,
                              indexOptionDefaults: <document>,
                              viewOn: <string>,
                              pipeline: <pipeline>,
                              collation: <document> } )
```

3) Creating a collection (table)



```
[> db.createCollection("TestCollection", { capped: true, autoIndexId: true, size : 6142800, max : 10000 } )
{
        "note" : "the autoIndexId option is deprecated and will be removed in a future release",
        "ok" : 1
}
```

I create a collection(table) and give some initial options along with the “create” because it will useful.

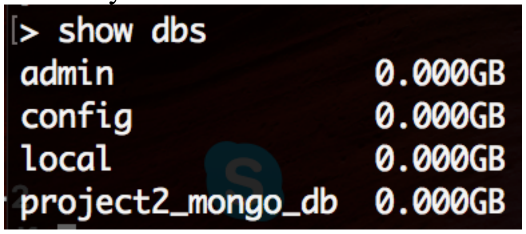
Capped: To create a capped collection, specify true. If you specify true, you must also set a maximum size in the sizefield.

AutoIndexId: Specify false to disable the automatic creation of an index on the _id field

Max: The maximum number of documents allowed in the capped collection

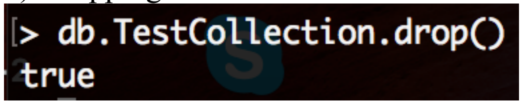
Size: Specify a maximum size in bytes for a capped collection

Note: if you create a collection or insert a document, then you can see your new database.
To verify:



```
[> show dbs
admin             0.000GB
config            0.000GB
local             0.000GB
project2_mongo_db 0.000GB
```

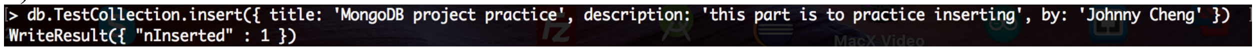
4) Dropping a collection



```
[> db.TestCollection.drop()
true
```

Since I’m still in selected database, then I can do drop() for the collection

5) Insert a document



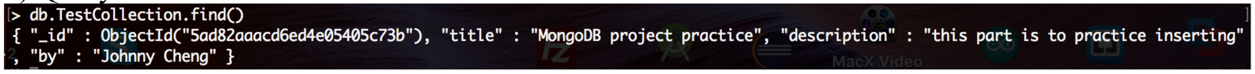
```
[> db.TestCollection.insert({ title: 'MongoDB project practice', description: 'this part is to practice inserting', by: 'Johnny Cheng' })
WriteResult({ "nInserted" : 1 })
```

I insert a document with title, description, and by whom into a collection.

Note: If the collection doesn't exist in the database, then MongoDB will create this collection and then insert a document into it.

Note: if we don't specify the _id parameter, then MongoDB assigns a unique ObjectId for this document.

6) Query a document

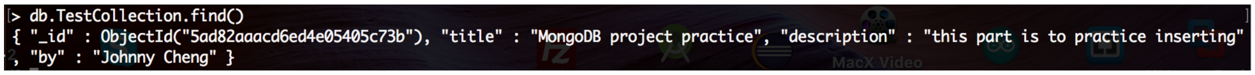


```
[> db.TestCollection.find()
{ "_id" : ObjectId("5ad82aaacd6ed4e05405c73b"), "title" : "MongoDB project practice", "description" : "this part is to practice inserting"
, "by" : "Johnny Cheng" }
```

I test a Query for the document I just inserted.

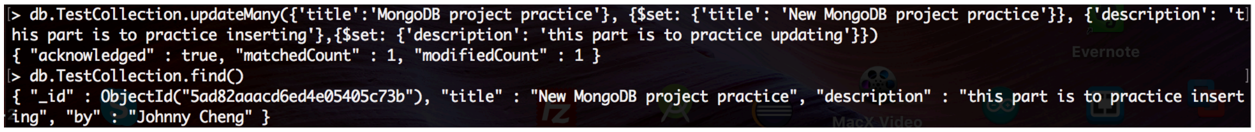
7) Update a document

Before:



```
> db.TestCollection.find()
{ "_id" : ObjectId("5ad82aaacd6ed4e05405c73b"), "title" : "MongoDB project practice", "description" : "this part is to practice inserting"
, "by" : "Johnny Cheng" }
```

After:

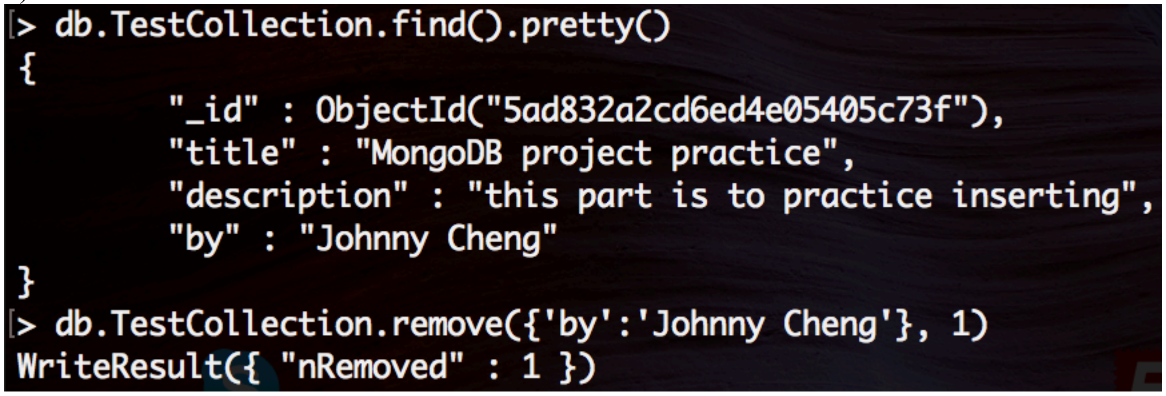


```
> db.TestCollection.updateMany({'title':'MongoDB project practice'}, {$set: {'title': 'New MongoDB project practice'}}, {'description': 't
his part is to practice inserting'},{$set: {'description': 'this part is to practice updating'}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.TestCollection.find()
{ "_id" : ObjectId("5ad82aaacd6ed4e05405c73b"), "title" : "New MongoDB project practice", "description" : "this part is to practice insert
ing", "by" : "Johnny Cheng" }
```

I update title with a net set and description with a new set.

Note: By default, MongoDB will update only a single document. I thought using updateMany() can update multiple documents but it still cannot, same as setting a parameter ‘multi’ to true

8) Delete Document:



```
> db.TestCollection.find().pretty()
{
        "_id" : ObjectId("5ad832a2cd6ed4e05405c73f"),
        "title" : "MongoDB project practice",
        "description" : "this part is to practice inserting",
        "by" : "Johnny Cheng"
}
> db.TestCollection.remove({'by':'Johnny Cheng'}, 1)
WriteResult({ "nRemoved" : 1 })
```

I delete the document and return 1, meaning it’s successful