

## MongoDB introduction

MongoDB ဟာ Nosql လို့ခေါ်တဲ့ non-relational အမျိုးအစားဖြစ်ပါတယ်။ ဒီမှာကျတော့ table ဆိုတာမရှိဘဲ collection လို့ခေါ်ပါတယ်။ row ကို document လို့ခေါ်ပြီး column ကိုကျတော့ field လို့ခေါ်ပါတယ်။

MongoDB မှာ database က collections တွေအများအပြားပါဝင်ပြီး collections တိုင်းမှာ documents တွေပါဝင်ပါတယ်။ Document တွေကို database ထဲကို တိုက်ရိုက်ထည့်လို့မရတဲ့အတွက် collection လုပ်ပြီး ထည့်ပါတယ်။

Document တိုင်းမှာ မတူညီတဲ့ unique ID တခုစီရှိပါတယ်။ Documents တွေကို nested လုပ်ထားတဲ့ embedded ပုံစံနဲ့လဲရေးလို့ရသလို array ပုံစံနဲ့လဲရေးလို့ရပါတယ်။

MongoDB မှာရေးတဲ့ပုံစံက Javascript က json format နဲ့ရေးပြီး သူက python ရဲ့ dictionary နဲ့ဆင်တူပါတယ်။

VS code မှာ သုံးမယ်ဆိုရင် mongodb extension ကို install တင်ပြီးသုံးလို့ရပါတယ်။ VS code terminal မှာ mongosh လို့ရိုက်လိုက်တာနဲ့ mongo shell ကိုသုံးလို့ရမှာဖြစ်ပါတယ်။ ထို့အတူ Mongo Compass မှာလဲ လွယ်ကူစွာအသုံးပြုလို့ရပါတယ်။

ကျနော်အကြံချင်ပေးတာကတော့ mongodb ကို သုံးမယ်ဆိုရင် shell ရယ် ကိုယ်အသုံးများတဲ့ programming language ရယ် mongo compass ရယ်ကို မျှသုံးမယ်ဆိုရင် လွယ်ကူပါလိမ့်မယ်။ ပထမဆုံးအနေနဲ့တော့ localhost မှာအရင်စမ်းကြည့်ပြီးမှ MongoDB Atlas မှ သုံးကြည့်ပါ။ MongoDB Atlas နဲ့ သုံးမယ်ဆိုရင် အကောင့်အရင်ဖောက်ပြီးမှ connection string ကိုချိတ်ပြီးသုံးမှ ရမှာပါ။

သူ့ကို စသုံးမယ်ဆိုရင် stable ဖြစ်တဲ့ version တခုခုကို အရင် install တင်ပါ။ ကျနော်ကတော့ MongoDB compass , MongoDB version 5.0 နဲ့ mongodb shell ကို install တင်ထားပါတယ်။ mongodb extension ကို vscode မှာ install တင်ပြီးလဲ သုံးလို့ရပါတယ်။ ပထမဆုံး ဒီမှာ ကျနော်သိသလို ရှင်းသွားမယ့်ပုံစံက mongodb shell ကနေအရင်ရှင်းပြပြီးနောက် MongoDB ရဲ့ python library ဖြစ်တဲ့ pymongo ကို ရှင်းပြမှာဖြစ်ပါတယ်။

### 1) Mongo Shell အားအသုံးပြုခြင်း

Mongo Shell အား vs code မှာဖြစ်ဖြစ် သူ့ရဲ့ shell terminal မှာဖြစ်ဖြစ် အသုံးပြုလို့ရပါတယ်။ mongo shell ၏ အသုံးဝင်သော command များမှာအောက်ပါအတိုင်းဖြစ်ပါတယ်-

a) show dbs

**မိမိစက်ထဲတွင်ရှိသော database များအား ကြည့်ခြင်း**

b) use (database\_name)

ရှိပြီးသား database အသုံးပြုရာတွင် (သို့) data base အသစ်ပြုလုပ်ခြင်း  
အလွယ်လုပ်ချင်ရင် mongo compass ထဲမှာလုပ်လို့ရပါတယ်။

c) show collections

မိမိ database ထဲတွင် ရှိသော collection များအားကြည့်ခြင်း

d) data base အားဖျက်ခြင်း

db.dropDatabase() နည်းဖြင့် အနောက်ဆုံးမှာရှိသော database အားဖျက်ခြင်း  
ဒီလိုမဖျက်ချင်ဘူးဆိုရင် mongoDB compass ထဲကိုဝင်ပြီးလဲဖျက်လို့ရပါတယ်။

## 2) collection တစ်ခုပြုလုပ်ခြင်း

Sql နဲ့မတူတာက table ကို collection လို့ခေါ်ပါတယ်။ ထို့အတူ row ကို document လို့ခေါ်ပြီး column ကို field လို့ခေါ်ပါတယ်။

Mongo Compass မှာ collection တစ်ခုပြုလုပ်မယ်ဆိုရင် ကိုယ့်ရဲ့ database name ကိုထောက်ပြီး create collection ဆိုတာကိုနှိပ်လိုက်ပြီး နာမည်တခုပေးလိုက်ရင်ရပါပြီ။

Shell မှာကျတော့ db.createCollection('collection\_name') လို့ရိုက်ပေးရပါမယ်။

ဥပမာ - db.createCollection('fruits')

ကိုယ်လုပ်လိုက်တဲ့ collection တွေကို ပြန်ကြည့်မယ်ဆိုရင် shell ထဲမှာ show collections လို့ရိုက်ပြီးကြည့်လို့ရပါတယ်။ Mongo Compass မှာဆိုရင် တန်းပေါ် ခဲ့မှာဖြစ်ပါတယ်။ တကယ်လို့မပေါ် ခဲ့ရင်လဲ Databases ကို refresh လုပ်ရင်ပေါ် ခဲ့မှာဖြစ်ပါတယ်။

## 3) Useful CRUD methods of MongoDB

1) C-Create

insertOne(data,options)

insertMany(data,options)

2) R-Read

find(filter,options)

findOne(filter,options)

3) U-Update

updateOne(filter,data,options)

updateMany(filter,data,options)

replaceOne(filter,data,options)

4) Delete

deleteOne(filter,options)

deleteMany(filter,options)

အပေါ်က နည်းလေးတွေကိုဖတ်လိုက်တာနဲ့ ဘယ်လိုလုပ်ရမယ်ဆိုတာ သဘောပေါက် မယ်လို့ယူဆပါတယ်။ ပထမဆုံးအနေနဲ့ ကိုယ့်ရဲ့ collection ထဲကို တန်ဖိုးတွေကိုထည့်မယ် ဆိုရင် insert method ဖြင့်ထည့်လို့ရပါတယ်။ document တခုထဲဆိုရင်တော့ insert one ကို သုံးပြီး အများအပြားဆိုရင်တော့ insert many နဲ့သုံးပါတယ်။

```
db.fruit.insertOne({name:'apple',price:250,from:'kta'})
```

```
db.fruit.insertMany([{'name':'orange',price:350,from:'ygn'},
                      {'name':'grape',price:450,from:'strawberry'}])
```

ကိုယ်ထည့်လိုက်တဲ့ collection တွေကို ပြန်ကြည့်မယ်ဆိုရင်

```
db.products.find()
```

```
db.products.find().pretty()
```

ကိုယ်ထည့်မယ့်တန်ဖိုးတွေထဲမှာလဲ list, array, dictionary လို nested ဖြစ်နေတဲ့ data အမျိုးအစားတွေပါလဲ ရပါတယ်။ ပြီးတော့ MongoDB မှာ programming language အလိုက် driver တွေရှိပါတယ်။

Mongo Compass မှာဆိုရင်လဲ insert လုပ်မယ်ဆိုရင် လဲအလွယ်တကူလုပ်လို့ရပါ တယ်။ ထို့အတူ csv file တွေ json file တွေကိုလဲ ထည့်ပြီးသုံးလို့ရပါတယ်။

#### 4) sorting

```
db.fruit.find().sort({'name':1})
```

1 သည် ascending order ဖြစ်ပြီး -1 သည် descending order ဖြစ်သည်။

##### a) တန်ဖိုးအကြီးဆုံးအားရှာခြင်း

```
db.fruit.find().sort({'price':-1}).limit(1)
```

#### 5) using find method

```
a) db.fruit.find({'name':'apple'})
```

```
b) db.fruit.find({'price':300})
```

```
c) db.fruit.find({'name':'apple','price':300})
```

#### projection paramter

သူ့ကိုရေးတဲ့ပုံက db.fruit.find({query},{projection})

```
db.fruit.find({},{'name':true})
```

true မရေးချင်ရင် 1 ကိုရေးလို့ရတယ်။

```
db.fruit.find({},{'_id':false,'name':true})
```

## 6) update document

### a) updateOne

```
db.fruit.updateOne(filter,update)
```

```
db.fruit.updateOne({name:'apple'},{$set:{from:'Ygn'}})
```

### b) updateMany

```
db.fruit.updateMany({},{$set:{from:'Ygn'}})
```

### c) update လုပ်ထားသည့်များကို remove လုပ်ခြင်း

```
db.fruit.updateOne({name:'apple'},{$unset:{from:''}})
```

တခုသတိထားရမှာက crud လုပ်ရင် query မှာစိတ်ကြိုက်စစ်လို့ရတယ်ဆိုတာ မမေ့ဖို့လိုပါတယ်။

## 7) delete

```
db.fruit.deleteOne({name:'apple'})
```

```
db.fruit.deleteMany({from:'Katha'})
```

## 8) query operator

```
db.fruit.find({name:{$ne: 'apple'}})
```

```
db.fruit.find({price:{$lt: 20}})
```

```
db.fruit.find({price:{$gte:3,$lte:4}})
```

```
db.fruit.find({name:{$in:['apple','orange','grape']}})
```

*lt က less than, gt က greater than , gte က greater than or equal to ဖြစ်ပြီး lte က less than or equal to ဖြစ်ပါတယ်။ eq က equal ဖြစ်ပြီး ne က not equal ဖြစ်ပါတယ်။ ထို့အတူ in နဲ့ nin(not in) ကိုလည်း သုံးလို့ရပါတယ်။ mongo compass ထဲမှာဆိုရင် filter projection ကို သေချာသုံးဖို့တော့လိုတယ်။*

## 9) exporting data

Data တွေကို စိတ်ကြိုက် ကစားပြီးရင် mongo compass ကနေပဲ အလွယ်တကူ ကိုယ်ကြိုက်တဲ့ format (json မဟုတ်ရင် csv) နဲ့ export ကိုထုတ်လို့ရပါတယ်။