

Nama : Jihad Satrio Utama
NIM : 17050623024
Prodi : D3 Manajemen Informatika

TUGAS BASIS DATA LANJUT ONE TO MANY DAN MANY TO MANY MENGGUNAKAN MONGODB

1. One to many (satu ke banyak) saya memiliki contoh pada 2 tabel yaitu dosen dan kelas dimana satu dosen dapat memilih banyak kelas

Dosen :	Kelas :
- id	- id
- nama	- Mata Kuliah
- kelas [id]	

Membuat database

```
> use tugas
switched to db tugas
```

Insert data

```
> db.dosen.insert({_id:1,nama:"Jihad Satrio",kelas:[1,3]})
WriteResult({ "nInserted" : 1 })
```

Melihat data

```
> db.dosen.find()
{ "_id" : 1, "nama" : "Jihad Satrio", "kelas" : [ 1, 3 ] }
> db.kelas.find()
{ "_id" : "1", "matkul" : "Basdat MI17" }
{ "_id" : 2, "matkul" : "Basdat MI18" }
{ "_id" : 3, "matkul" : "Pemrograman MI18" }
```

Update Data

```
> db.dosen.update({_id:1},{ $set : { nama : "Abrisam" } })
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.dosen.find()
{ "_id" : 1, "nama" : "Abrisam", "kelas" : [ 1, 3 ] }
```

Delete Data

```
> db.kelas.deleteOne({_id:"1"})
{ "acknowledged" : true, "deletedCount" : 1 }
> db.kelas.find()
{ "_id" : 2, "matkul" : "Basdat MI18" }
{ "_id" : 3, "matkul" : "Pemrograman MI18" }
>
```

2. Many to many (banyak ke banyak) saya memili contoh dua tabel product dan customer dimana banyak product dapat dimiliki oleh banyak customer begitu juga sebaliknya

Product :

-id
-nmbrg
-harga
-customer [id]

customer :

-id
-nama

- Insert dan Read

```
> db.product.insertMany([
...   _id:1,
...   nmbrg:"helm lvl3",
...   harga:100,
...   customer:[1,2]},{
...   _id:2,
...   nmbrg:"peta pubg",
...   harga:200,
...   customer:[1,3]},{
...   _id:3
... },
...   nmbrg:"SLR Sniper",
...   harga:200,
...   customer:[3,2]})
{ "acknowledged" : true, "insertedIds" : [ 1, 2, 3 ] }
> db.product.find()
{ "_id" : 1, "nmbrg" : "helm lvl3", "harga" : 100, "customer" : [ 1, 2 ] }
{ "_id" : 2, "nmbrg" : "peta pubg", "harga" : 200, "customer" : [ 1, 3 ] }
{ "_id" : 3, "nmbrg" : "SLR Sniper", "harga" : 200, "customer" : [ 3, 2 ] }
>
```

In what situation should I use each one?

The `insert()` method is deprecated in major driver so you should use `insertMany()` method whenever you want to insert a single document into your collection. When you want to insert multiple documents into your collection, you should use `insertMany()` the documentation but the fact is that nobody really writes an article about this. This thing applies to `updateOne`, `updateMany`, `deleteOne`, `deleteMany`, `findOneAndUpdate` and `findOneAndReplace`. See [Write C...](#)

Share Improve this answer

edited May 1 '16 at 8:38

```
> db.customer.insertMany([
...   _id:1,
...   nama:"harmoko"},{
...   _id:2,
...   nama:"sugeng priambodo"},{
...   _id:3,
...   nama:"pablo galgadot"})
{ "acknowledged" : true, "insertedIds" : [ 1, 2, 3 ] }
> db.customer.find()
{ "_id" : 1, "nama" : "harmoko" }
{ "_id" : 2, "nama" : "sugeng priambodo" }
{ "_id" : 3, "nama" : "pablo galgadot" }
>
```

Page 2 of 2

119 words, 60 characters

Default Style