SESSIONS 2

mongoDB

Working with mongoDB Database

Data Science Program

Purwadhika
Startup and Coding School

MongoDB Query

NoSQL Databases

Purwadhika
Startup and Coding School

Show Databases

Write on your terminal

- Show all database :

  `$ show dbs`

- Show avtive databse (default :test) :

  `$ db`

- Create and use automatically with databases “toko” :

  `$ use toko`

- Delete database “toko” :

  ```
  $ use toko
  $ db.dropDatabase()
  ```

Collection

Write on your terminal

- Creating database “user” :

  ```
  $ db.createUser({user : “George”, pwd : “123456”, roles : [“dataAnalyst”,”databaseAdministrator”]})
  ```

- Create collection “karyawan” :

  ```
  $ db.createCollection(‘karyawan’)
  ```

- Show all collection in db:

  ```
  $ show collections
  ```

- Delete collection “karyawan” :

  ```
  $ db.karyawan.drop()
  ```

Add Data Record

Write on your terminal

- Add 1 data JSON to collection “karyawan”:

```
$ db.karyawan.insert({name : “George”, age : 24})
```

- Add multiple data to collection “karyawan” :

```
$ db.karyawan.insert([
{name : “Milo”, age : 27},
{name : “Moly”, age : 26})
```

- Show all data in collection “karyawan”:

```
$ db.karyawan.find({})
$ db.karyawan.find().pretty()
$ db.karyawan.find({name: ”George”}).pretty()
```

Update Data

Write on your terminal

- Update all property data:

```
$ db.karyawan.update({name : “George”},{name : “George”, age : 28})
```

- Update specific property data :

```
$ db.karyawan.update(
{name : “Milo”},
{$set : {age : 30}})
```

- Update “name” property:

```
$ db.karyawan.update(
{name:”Moly”},
{$rename : {“age” : “usia”}})
```

Update Many Data

mongoDB

Write on your terminal

- Update all data just with data age = 21 :

```
$ db.karyawan.updateMany({},{age : 21})
```

- Update all data property with age = 21 :

```
$ db.karyawan.updateMany({},
{$set : {age : 21}})
```

- Update property name to all data:

```
$ db.karyawan.updateMany({},
{$rename : {“age” : “umur”}})
```

Purwadhika
Startup and Coding School

Remove

Write on your terminal

- Delete 1 property from data:

```
$ db.karyawan. update({name: “George”},
{$unset:”{age :true}})
```

or

```
$ db.karyawan. update({name: “George”},
{$unset:”{age :1}})
```

- Delete 1 data :

```
$ db.karyawan. remove({name: “George”})
```

- Delete all data :

```
$ db.karyawan. remove({})
```

And & Or

Write on your terminal

- Show data that have value property name = “George” ***and*** age = 28:

  ```
  $ db.karyawan. find({$and:
  [{name:”George”},{age :28}]}).pretty()
  ```

- Show data that have value property name = “George” ***and*** name = “Moly”:

  ```
  $ db.karyawan. find({$or:
  [{name: ”George”},{name :”Moly”}]}).pretty()
  ```

Lower and Greater Than

Write on your terminal

- Show all data with property age < 25:

```
$ db.karyawan. find({age : {$lt:25}}).pretty()
```

- Show all data with property age < 25:

```
$ db.karyawan. find({age : {$gt:25}}).pretty()
```

Lower and Greater Than Equal

Write on your terminal

- Show all data with property age < 25:

  ```
  $ db.karyawan. find({age : {$lte:25}}).pretty()
  ```

- Show all data with property age < 25:

  ```
  $ db.karyawan. find({age : {$gte:25}}).pretty()
  ```

Limit & Skip

Write on your terminal

- Show the first 2 data:

```
$ db.karyawan. find().limit(2)
```

- Show 1 data after the first 2 data:

```
$ db.karyawan. find().limit(1).skip(2)
```

- Show 3 data after the first 3 data

```
$ db.karyawan. find().limit(3).skip(3)
```

Sort

Write on your terminal

- Sort the data with ***ascending*** based on property name:

  ```
  $ db.karyawan. find().sort({name:1})
  ```

- Sort the data with ***descending*** based on property name :

  ```
  $ db.karyawan. find(). sort({name:-1})
  ```

- Sort the data with ***ascending*** based on property age:

  ```
  $ db.karyawan. find().sort({age:1})
  ```

- Sort the data with ***descending*** based on property age :

  ```
  $ db.karyawan. find(). sort({age:-1})
  ```

Count

Write on your terminal

- Count total data in collection “karyawan”:

```
$ db.karyawan. find().count()
```

- Count total data with property name = ”George” :

```
$ db.karyawan. find({name:”George”}).count()
```

Regex

Write on your terminal

- SQL :

  $ select * from karyawan where name like ‘%a’

- mongoDB :

  $ db.karyawan.find({name:{$regex : /a$/}})

- SQL :

  $ select * from karyawan where name like ‘%g%’

- mongoDB :

  $ db.karyawan.find({name:{$regex : ‘g’ , $options : ‘a’ }})

Reference

https://docs.mongodb.com/manual/core/databases-and-collections/

https://docs.mongodb.com/manual/tutorial/insert-documents/

https://docs.mongodb.com/manual/tutorial/update-documents/

https://docs.mongodb.com/manual/tutorial/query-documents/

https://docs.mongodb.com/manual/reference/operator/query/regex/

https://docs.mongodb.com/manual/tutorial/remove-documents/

https://docs.mongodb.com/manual/tutorial/update-documents/