Getting Started with MongoDB

Ondrej Sika
ondrej@ondrejsika.com

Data Script s.r.o., Praha, 30. 3. 2017

https://go.sika.guru/mongo

Agenda

- Conventions
- Databases
- Collections
- Insert
- Select (Find)
- Update
- Delete
- Indexes
- Aggregation

Conventions

Conventions

- Database -> Database
- Table -> Collection
- Row-> Document
- Column -> Field (key)
- Index -> Index

MongoDB has no joins or relations

Databases

Select & Show Databases

```
> use mydb
switched to db mydb

> show dbs
admin  0.000GB
local  0.000GB
test   0.000GB
```

Create Databases

Create by insert of first document

```
> db.pets.insert({name: 'Pista', kind:
'dog'})
WriteResult({ "nInserted" : 1 })
> show dbs
admin  0.000GB
local  0.000GB
mydb   0.000GB
test   0.000GB
```

Drop Databases

```
> db.dropDatabase()
{ "dropped" : "mydb", "ok" : 1 }
> show dbs
admin  0.000GB
local  0.000GB
test   0.000GB
```

Collections

Create Collection

```
> use test
switched to db test
> db.createCollection("mycol")
{ "ok" : 1 }
```

Create Capped Collection

```
> db.createCollection("cappedcol", {
    capped: true,
    autoIndexID: true,
    size: 1024,
    max: 1
})
{ "ok" : 1 }
```

List Collection

```
> show collections
mycol
crappedcol
```

Drop Collection

```
> db.mycol.drop()
true
```

Insert

Insert

```
db.pets.insert({
    name: 'Fista',
    kind: 'cat',
    age: 2,
    colors: ['black', 'white'],
})
```

InsertMany

```
> db.pets.insertMany([
    {name: "Ben"},
    {name:"Zen"}
])
{"acknowledged" : true,
 "insertedIds" : [
  ObjectId("58..b9"), ObjectId("58..ba")]
}
```

Select (Find)

Find

```
> db.pets.find()
{ "name" : "Fista", "kind" : "cat", ...}
{ "name" : "Pista", "kind" : "dog", ...}
{ "name" : "Mista", "kind" : "rat", ... }
```

Pretty output

```
> db.pets.find().pretty()
{
    "_id" : ObjectId("58..."),
    "name" : "Fista",
    "kind" : "cat",
    "age" : 2,
    "colors" : ["black", "white"]
}
```

Operators

```
- {kind: 'rat'}
- {age: {$lt: 2}}
- {age: {$gt: 2}}
- {age: {$gte: 2}}
- {age: {$ne: 2}}
- {age: {$in: [1, 2, 3]}}
- {age: {$nin: [1, 2, 3]}}
- {age: {$exists: true}}
```

Operators Example

```
> db.pets.find({name: 'Pista'})

> db.pets.find({kind: 'dog'})

> db.pets.find({colors: 'black'})

> db.pets.find({colors: {$ne: 'black'}})
```

Columns

You shourld select some colums by:

```
db.COLLECTION_NAME.find(QUERY, COLLUMNS)
```

Example:

```
> db.pets.find({}, {name: 1})
> db.pets.find({}, {name: 1, _id: 0})
> db.pets.find({}, {_id: 0})
```

AND & OR

AND

```
> db.pets.find({age: {$gte: 2}, kind:
'dog'})
```

OR

```
> db.pets.find({$or: [{kind: 'rat'}, {kind:
'cat'}]})
```

Limit & Offset

```
> db.pets.find().limit(1)

> db.pets.find().skip(1)

> db.pets.find().skip(1).limit(1)
```

Sort

```
> db.pets.find().sort({age:1})

> db.pets.find().sort({age:-1})

>
db.pets.find().sort({kind:1}).sort({age:1})
```

Update

Update

Single Document Update

```
db.COLLECTION_NAME.update(SELECTIOIN_CRITERIA, UPDATED_DATA)
```

Multiple Documents Update

```
db.COLLECTION_NAME.update(SELECTIOIN_CRITERIA, UPDATED_DATA, {multi: true})
```

Update Example

```
> db.pets.update({name: 'Ben'},
  {$set: {age: 13}})
WriteResult({ "nMatched" : 1, "nUpserted" :
0, "nModified" : 1 })
> db.pets.find({name: 'Ben'})
{ "name" : "Ben", "kind" : "dog", "age" :
13, "colors" : [ "white" ] }
```

Unset Example

```
> db.pets.update({name: 'Ben'},
  {$unset: {age: 1}})
WriteResult({ "nMatched" : 1, "nUpserted" :
0, "nModified" : 1 })
> db.pets.find({name: 'Ben'})
{ "name" : "Ben", "kind" : "dog", "colors" :
[ "white" ] }
```

Insert Update

```
> db.collection.replaceOne({name: "Bak"},
   {name: "Bak", age: 8},
   {upsert: 1})
{
   "acknowledged" : true,
   "matchedCount" : 0,
   "modifiedCount" : 0,
   "upsertedId" : ObjectId("58d..82f")
}
```

Delete

Delete

Single Document Delete

```
db.COLLECTION_NAME.remove(DELETION_CRITERIA, 1)
```

Multiple Documents Delete

```
db.COLLECTION_NAME.remove(DELETION_CRITERIA)
```

Delete Example

```
> db.pets.remove({age: {$gt: 1}}, 1)
WriteResult({ "nRemoved" : 1 })

> db.pets.remove({age: {$gt: 1}})
WriteResult({ "nRemoved" : 2 })
```

Delete All Documents

```
> db.pets.remove({})
WriteResult({ "nRemoved" : 1 })
```

Index

Get Indexes

```
> db.pets.getIndexes()
[
    {
        "v" : 1,
        "key" : {"_id" : 1},
        "name" : "_id_",
        "ns" : "test.pets"
    }
]
```

Create Index

```
> db.pets.ensureIndex({age:1})
{
    "createdCollectionAutomatically" :
false,
    "numIndexesBefore" : 1,
    "numIndexesAfter" : 2,
    "ok" : 1
}
```

Create Index with parameters

```
> db.pets.ensureIndex({name: 1}, {
    unique: 1,
    backgroud: 1,
})
```

Drop Index

```
> db.pets.dropIndex({age:1})
{ "nIndexesWas" : 3, "ok" : 1 }
```

Aggregation

Count

```
> db.pets.count()
4
> db.pets.find().count()
4
> db.pets.find({kind: 'dog'}).count()
2
```

Aggregation Operators

- $sum
- $avg
- $min
- $max
- $push
- $addToSet
- $first
- $last

Aggregation Example

```
> db.pets.aggregate([{$group : {
    _id : "$kind",
    count: {$sum : 1},
    age: {$avg: '$age'}
}}])
{ "_id" : "cat", "count" : 1, "age" : 2 }
{ "_id" : "rat", "count" : 1, "age" : 1 }
{ "_id" : "dog", "count" : 2, "age" : 8 }
```

Aggregation by Multiple Keys

```
> db.pets.aggregate([{$group : {_id : {kind:
'$kind', age: '$age'}, count: {$sum : 1}}}])

{ "_id" : { "kind" : "cat", "age" : 2 },
"count" : 1 }
{ "_id" : { "kind" : "rat", "age" : 1 },
"count" : 1 }
...
```

Resources

- https://docs.mongodb.com
- https://go.bysika.cz/mongo
  - Slides
  - Links to related content, eg.: tools, docs, books, ...

Thank you & Questions

Ondrej Sika

ondrej@ondrejsika.com
@ondrejsika

https://go.bysika.cz/mongo