[RDBMS vs MONGODB 용어비교

|  |  |
| --- | --- |
| RDBMS SQL Term/Concepts | MongoDB Term/Concepts |
| database | database |
| table | collection |
| row | document or BSON document |
| column | field |
| index | index |
| primary key | primary key  (Mongo DB에서는 primary key가 \_id 라는 필드로 자동 생성됨) |

[MONGODB설치]

**vi /etc/yum.repos.d/mongodb-org-3.4.repo**

[mongodb-org-3.4]

name=MongoDB Repository

baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/3.4/x86\_64/

gpgcheck=0

enabled=1

**vi /etc/yum.conf -> yum update시 버젼업그레이드 방지**

exclude=mongodb-org,mongodb-org-server,monogdb-org-shell,mongodb-org-mongos,mongodb-org-tools

|  |  |
| --- | --- |
| mongodb-org | A metapackage that will automatically install the four component packages listed below |
| mongodb-org-server | Contains the mongod daemon and associated configuration and init scripts |
| mongodb-org-mongos | Contains the mongos daemon. |
| mongodb-org-shell | Contains the mongo shell. |
| mongodb-org-tools | Contains the following MongoDB tools: mongoimport bsondump, mongodump, mongoexport, mongofiles, mongooplog, mongoperf, mongorestore, mongostat, and mongotop |

gpgk[root@centos-master ~]# **yum install mongodb-org**

Loaded plugins: fastestmirror

Loading mirror speeds from cached hostfile

\* base: ftp.kaist.ac.kr

\* extras: ftp.kaist.ac.kr

\* updates: ftp.kaist.ac.kr

Resolving Dependencies

--> Running transaction check

---> Package mongodb-org.x86\_64 0:3.4.9-1.el7 will be installed

--> Processing Dependency: mongodb-org-tools = 3.4.9 for package: mongodb-org-3. 4.9-1.el7.x86\_64

--> Processing Dependency: mongodb-org-shell = 3.4.9 for package: mongodb-org-3. 4.9-1.el7.x86\_64

--> Processing Dependency: mongodb-org-server = 3.4.9 for package: mongodb-org-3 .4.9-1.el7.x86\_64

--> Processing Dependency: mongodb-org-mongos = 3.4.9 for package: mongodb-org-3 .4.9-1.el7.x86\_64

--> Running transaction check

---> Package mongodb-org-mongos.x86\_64 0:3.4.9-1.el7 will be installed

---> Package mongodb-org-server.x86\_64 0:3.4.9-1.el7 will be installed

---> Package mongodb-org-shell.x86\_64 0:3.4.9-1.el7 will be installed

---> Package mongodb-org-tools.x86\_64 0:3.4.9-1.el7 will be installed

--> Finished Dependency Resolution

Dependencies Resolved

================================================================================

Package Arch Version Repository Size

================================================================================

Installing:

mongodb-org x86\_64 3.4.9-1.el7 mongodb-org-3.4 5.8 k

Installing for dependencies:

mongodb-org-mongos x86\_64 3.4.9-1.el7 mongodb-org-3.4 12 M

mongodb-org-server x86\_64 3.4.9-1.el7 mongodb-org-3.4 20 M

mongodb-org-shell x86\_64 3.4.9-1.el7 mongodb-org-3.4 11 M

mongodb-org-tools x86\_64 3.4.9-1.el7 mongodb-org-3.4 49 M

Transaction Summary

================================================================================

Install 1 Package (+4 Dependent packages)

Total download size: 91 M

Installed size: 258 M

Is this ok [y/d/N]: y

Downloading packages:

warning: /var/cache/yum/x86\_64/7/mongodb-org-3.4/packages/mongodb-org-3.4.9-1.el 7.x86\_64.rpm: Header V3 RSA/SHA1 Signature, key ID a15703c6: NOKEY

Public key for mongodb-org-3.4.9-1.el7.x86\_64.rpm is not installed

(1/5): mongodb-org-3.4.9-1.el7.x86\_64.rpm | 5.8 kB 00:00

(2/5): mongodb-org-mongos-3.4.9-1.el7.x86\_64.rpm | 12 MB 00:08

(3/5): mongodb-org-server-3.4.9-1.el7.x86\_64.rpm | 20 MB 00:12

(4/5): mongodb-org-shell-3.4.9-1.el7.x86\_64.rpm | 11 MB 00:08

(5/5): mongodb-org-tools-3.4.9-1.el7.x86\_64.rpm | 49 MB 00:29

--------------------------------------------------------------------------------

Total 2.2 MB/s | 91 MB 00:42

Retrieving key from https://www.mongodb.org/static/pgp/server-3.4.asc

Importing GPG key 0xA15703C6:

Userid : "MongoDB 3.4 Release Signing Key <packaging@mongodb.com>"

Fingerprint: 0c49 f373 0359 a145 1858 5931 bc71 1f9b a157 03c6

From : https://www.mongodb.org/static/pgp/server-3.4.asc

Is this ok [y/N]: y

Running transaction check

Running transaction test

Transaction test succeeded

Running transaction

Installing : mongodb-org-mongos-3.4.9-1.el7.x86\_64 1/5

Installing : mongodb-org-tools-3.4.9-1.el7.x86\_64 2/5

Installing : mongodb-org-shell-3.4.9-1.el7.x86\_64 3/5

Installing : mongodb-org-server-3.4.9-1.el7.x86\_64 4/5

Created symlink from /etc/systemd/system/multi-user.target.wants/mongod.service to /usr/lib/systemd/system/mongod.service.

Installing : mongodb-org-3.4.9-1.el7.x86\_64 5/5

Verifying : mongodb-org-server-3.4.9-1.el7.x86\_64 1/5

Verifying : mongodb-org-3.4.9-1.el7.x86\_64 2/5

Verifying : mongodb-org-shell-3.4.9-1.el7.x86\_64 3/5

Verifying : mongodb-org-tools-3.4.9-1.el7.x86\_64 4/5

Verifying : mongodb-org-mongos-3.4.9-1.el7.x86\_64 5/5

Installed:

mongodb-org.x86\_64 0:3.4.9-1.el7

Dependency Installed:

mongodb-org-mongos.x86\_64 0:3.4.9-1.el7

mongodb-org-server.x86\_64 0:3.4.9-1.el7

mongodb-org-shell.x86\_64 0:3.4.9-1.el7

mongodb-org-tools.x86\_64 0:3.4.9-1.el7

Complete!ey=https://www.mongodb.org/static/pgp/server-3.4.asc

[root@centos-master ~]# **setenforce 0**

[root@centos-master ~]# **getenforce**

Permissive

[root@centos-master ~]# **systemctl status firewalld.service**

● firewalld.service - firewalld - dynamic firewall daemon

Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)

Active: inactive (dead)

Docs: man:firewalld(1)

**vi /etc/mongod.conf**

systemLog:

destination: file

logAppend: true

**path: /var/log/mongodb/mongod.log**

# Where and how to store data.

storage:

**dbPath: /var/lib/mongo**

journal:

enabled: true

# engine:

# mmapv1:

# wiredTiger:

# how the process runs

processManagement:

fork: true # fork and run in background

**pidFilePath: /var/run/mongodb/mongod.pid # location of pidfile**

# network interfaces

net:

**port: 27017**

**bindIp: 127.0.0.1 # Listen to local interface only, comment to listen on all interfaces.**

[root@centos-master mongo]# **systemctl start mongod.service**

[root@centos-master mongo]# **systemctl status mongod.service**

● mongod.service - High-performance, schema-free document-oriented database

Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; vendor preset: disabled)

**Active: active (running)** since Fri 2017-10-20 02:59:29 EDT; 4s ago

Docs: https://docs.mongodb.org/manual

Process: 1510 ExecStartPre=/usr/bin/chmod 0755 /var/run/mongodb (code=exited, status=0/SUCCESS)

Process: 1507 ExecStartPre=/usr/bin/chown mongod:mongod /var/run/mongodb (code=exited, status=0/SUCCESS)

Process: 1506 ExecStartPre=/usr/bin/mkdir -p /var/run/mongodb (code=exited, status=0/SUCCESS)

Main PID: 1516 (mongod)

Memory: 35.3M

CGroup: /system.slice/mongod.service

└─1516 /usr/bin/mongod -f /etc/mongod.conf

Oct 20 02:59:29 centos-master systemd[1]: Starting High-performance, schema-free document-oriented d...e...

Oct 20 02:59:29 centos-master systemd[1]: Started High-performance, schema-free document-oriented database.

Oct 20 02:59:29 centos-master mongod[1513]: about to fork child process, waiting until server is rea...ons.

Oct 20 02:59:29 centos-master mongod[1513]: forked process: 1516

Hint: Some lines were ellipsized, use -l to show in full.

[root@centos-master mongo]# **ps -ef | grep mongo**

mongod 1516 1 0 02:59 ? 00:00:00 /usr/bin/mongod -f /etc/mongod.conf

[root@centos-master mongo]# **netstat -anlp | grep mongod**

tcp 0 0 127.0.0.1:27017 0.0.0.0:\* LISTEN 1516/mongod

unix 2 [ ACC ] STREAM LISTENING 27521 1516/mongod /tmp/mongodb-27017.sock

[root@centos-master mongo]# **mongo**

MongoDB shell version v3.4.9

connecting to: mongodb://127.0.0.1:27017

MongoDB server version: 3.4.9

Welcome to the MongoDB shell.

For interactive help, type "help".

For more comprehensive documentation, see

http://docs.mongodb.org/

Questions? Try the support group

http://groups.google.com/group/mongodb-user

Server has startup warnings:

2017-10-20T02:59:29.814-0400 I CONTROL [initandlisten]

2017-10-20T02:59:29.814-0400 I CONTROL [initandlisten] \*\* WARNING: Access control is not enabled for the d atabase.

2017-10-20T02:59:29.814-0400 I CONTROL [initandlisten] \*\* Read and write access to data and confi guration is unrestricted.

2017-10-20T02:59:29.814-0400 I CONTROL [initandlisten]

2017-10-20T02:59:29.815-0400 I CONTROL [initandlisten]

2017-10-20T02:59:29.815-0400 I CONTROL [initandlisten] \*\* WARNING: /sys/kernel/mm/transparent\_hugepage/ena bled is 'always'.

2017-10-20T02:59:29.815-0400 I CONTROL [initandlisten] \*\* We suggest setting it to 'never'

2017-10-20T02:59:29.815-0400 I CONTROL [initandlisten]

2017-10-20T02:59:29.815-0400 I CONTROL [initandlisten] \*\* WARNING: /sys/kernel/mm/transparent\_hugepage/def rag is 'always'.

2017-10-20T02:59:29.815-0400 I CONTROL [initandlisten] \*\* We suggest setting it to 'never'

2017-10-20T02:59:29.815-0400 I CONTROL [initandlisten]

> **show dbs;**

admin 0.000GB

local 0.000GB

> **use myNewDatabase**

switched to db myNewDatabase

> **db.myCollection.insertOne({x:1});**

{

"acknowledged" : true,

"insertedId" : ObjectId("59e9af55b3e024e17140f8ea")

}

> **db.myCollection.find()**

{ "\_id" : ObjectId("59e9af55b3e024e17140f8ea"), "x" : 1 }

>exit

[root@centos-master mongo]# **wget** [**https://raw.githubusercontent.com/mongodb/docs-assets/primer-dataset/primer-dataset.json**](https://raw.githubusercontent.com/mongodb/docs-assets/primer-dataset/primer-dataset.json)

[root@centos-master mongo]# **mongoimport --db test --collection restaurants --drop --file primer-dataset.json**

2017-10-20T04:31:19.624-0400 connected to: localhost

2017-10-20T04:31:19.624-0400 dropping: test.restaurants

2017-10-20T04:31:20.974-0400 imported 25359 documents

> **help**

db.help() help on db methods

db.mycoll.help() help on collection methods

sh.help() sharding helpers

rs.help() replica set helpers

help admin administrative help

help connect connecting to a db help

help keys key shortcuts

help misc misc things to know

help mr mapreduce

show dbs show database names

show collections show collections in current database

show users show users in current database

show profile show most recent system.profile entries with time >= 1ms

show logs show the accessible logger names

show log [name] prints out the last segment of log in memory, 'global' is default

use <db\_name> set current database

db.foo.find() list objects in collection foo

db.foo.find( { a : 1 } ) list objects in foo where a == 1

it result of the last line evaluated; use to further iterate

DBQuery.shellBatchSize = x set default number of items to display on shell

exit quit the mongo shell

> **use test**

switched to db test

> **db.restaurants.insert(**

**... {**

**... "address" : {**

**... "street" : "2 Avenue",**

**... "zipcode" : "10075",**

**... "building" : "1480",**

**... "coord" : [ -73.9557413, 40.7720266 ]**

**... },**

**... "borough" : "Manhattan",**

**... "cuisine" : "Italian",**

**... "grades" : [**

**... {**

**... "date" : ISODate("2014-10-01T00:00:00Z"),**

**... "grade" : "A",**

**... "score" : 11**

**... },**

**... {**

**... "date" : ISODate("2014-01-16T00:00:00Z"),**

**... "grade" : "B",**

**... "score" : 17**

**... }**

**... ],**

**... "name" : "Vella",**

**... "restaurant\_id" : "41704620"**

**... }**

**... )**

WriteResult({ "nInserted" : 1 })

**rs0:PRIMARY> use test**

switched to db test

**rs0:PRIMARY> var bulk = db.test\_collection.initializeUnorderedBulkOp();**

**rs0:PRIMARY> people = ["Marc", "Bill", "George", "Eliot", "Matt", "Trey", "Tracy", "Greg", "Steve", "Kristina", "Katie", "Jeff"];**

**[**

**"Marc",**

**"Bill",**

**"George",**

**"Eliot",**

**"Matt",**

**"Trey",**

**"Tracy",**

**"Greg",**

**"Steve",**

**"Kristina",**

**"Katie",**

**"Jeff"**

**]**

**rs0:PRIMARY> for(var i=0; i<1000000; i++){**

**... user\_id = i;**

**... name = people[Math.floor(Math.random()\*people.length)];**

**... number = Math.floor(Math.random()\*10001);**

**... bulk.insert( { "user\_id":user\_id, "name":name, "number":number });**

**... }**

**rs0:PRIMARY> bulk.execute();**

BulkWriteResult({

"writeErrors" : [ ],

"writeConcernErrors" : [ ],

"nInserted" : 1000000,

"nUpserted" : 0,

"nMatched" : 0,

"nModified" : 0,

"nRemoved" : 0,

"upserted" : [ ]

})

**rs0:PRIMARY> show collections;**

restaurants

test

test\_collection

**>db.restaurants.find()**

**>db.restaurants.find( { "borough": "Manhattan" } )**

**>db.restaurants.find( { "address.zipcode": "10075" } )**

**>db.restaurants.find( { "grades.grade": "B" } )**

**> db.restaurants.find( { "grades.score": { $gt: 30 } } )**

**> db.restaurants.find( { "grades.score": { $lt: 10 } } )  
> db.restaurants.find( { "cuisine": "Italian", "address.zipcode": "10075" } )** --> AND조건쿼리

**> db.restaurants.find(**

**{ $or: [ { "cuisine": "Italian" }, { "address.zipcode": "10075" } ] }**

**)** --> OR조건쿼리

**>db.restaurants.find().sort( { "borough": 1, "address.zipcode": 1 } )**

**> db.restaurants.update(**

**... { "name" : "Juni" },**

**... {**

**... $set: { "cuisine": "American (New)" },**

**... $currentDate: { "lastModified": true }**

**... }**

**... )**

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

**> db.restaurants.update(**

**... { "restaurant\_id" : "41156888" },**

**... { $set: { "address.street": "East 31st Street" } }**

**... )**

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

**> db.restaurants.update(**

**... { "address.zipcode": "10016", cuisine: "Other" },**

**... {**

**... $set: { cuisine: "Category To Be Determined" },**

**... $currentDate: { "lastModified": true }**

**... },**

**... { multi: true} --> multiple documnet에 업데이트 하기위해**

**... )**

WriteResult({ "nMatched" : 20, "nUpserted" : 0, "nModified" : 20 })

**> db.restaurants.update(**

**... { "restaurant\_id" : "41704620" },**

**... {**

**... "name" : "Vella 2",**

**... "address" : {**

**... "coord" : [ -73.9557413, 40.7720266 ],**

**... "building" : "1480",**

**... "street" : "2 Avenue",**

**... "zipcode" : "10075"**

**... }**

**... }**

**... )**

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

**> db.restaurants.remove( { "borough": "Manhattan" } )**

WriteResult({ "nRemoved" : 10259 })

**> db.restaurants.find( { "borough": "Manhattan" } ).pretty()**

**> db.restaurants.remove( { "borough": "Queens" }, { justOne: true } ) --> 1개 다큐먼트만 삭제**

WriteResult({ "nRemoved" : 1 })

**> db.restaurants.find( { "borough": "Queens" } ).pretty()**

**> db.restaurants.remove({}) --> 모든 다큐먼트 삭제**

WriteResult({ "nRemoved" : 15098 })

**> db.restaurants.drop()**

true

**> show dbs;**

admin 0.000GB

local 0.000GB

myNewDatabase 0.000GB

**> show collections;**

**> db.restaurants.aggregate(**

**... [**

**... { $group: { "\_id": "$borough", "count": { $sum: 1 } } }**

**... ]**

**... );**

{ "\_id" : "Missing", "count" : 51 }

{ "\_id" : "Staten Island", "count" : 969 }

{ "\_id" : "Brooklyn", "count" : 6086 }

{ "\_id" : "Bronx", "count" : 2338 }

{ "\_id" : "Queens", "count" : 5656 }

{ "\_id" : "Manhattan", "count" : 10259 }

**> db.restaurants.aggregate(**

**... [**

**... { $match: { "borough": "Queens", "cuisine": "Brazilian" } },**

**... { $group: { "\_id": "$address.zipcode" , "count": { $sum: 1 } } }**

**... ]**

**... );**

{ "\_id" : "11377", "count" : 1 }

{ "\_id" : "11103", "count" : 1 }

{ "\_id" : "11106", "count" : 3 }

{ "\_id" : "11368", "count" : 1 }

{ "\_id" : "11101", "count" : 2 }

**> db.restaurants.createIndex( { "cuisine": 1 } ) --> cuisine로 오름차순 인덱스생성**

{

"createdCollectionAutomatically" : false,

"numIndexesBefore" : 1,

"numIndexesAfter" : 2,

"ok" : 1

}

**> db.restaurants.createIndex( { "cuisine": 1, "address.zipcode": -1 } )**

{

"createdCollectionAutomatically" : false,

"numIndexesBefore" : 2,

"numIndexesAfter" : 3,

"ok" : 1

}

**> db.restaurants.getIndexes()**

[

{

"v" : 2,

"key" : {

"\_id" : 1

},

"name" : "\_id\_",

"ns" : "test.restaurants"

},

{

"v" : 2,

"key" : {

"cuisine" : 1

},

"name" : "cuisine\_1",

"ns" : "test.restaurants"

},

{

"v" : 2,

"key" : {

"cuisine" : 1,

"address.zipcode" : -1

},

"name" : "cuisine\_1\_address.zipcode\_-1",

"ns" : "test.restaurants"

}

]

**> db.restaurants.dropIndex( "cuisine\_1\_address.zipcode\_-1")**

{ "nIndexesWas" : 3, "ok" : 1 }

**> db.restaurants.getIndexes()**

[

{

"v" : 2,

"key" : {

"\_id" : 1

},

"name" : "\_id\_",

"ns" : "test.restaurants"

},

{

"v" : 2,

"key" : {

"cuisine" : 1

},

"name" : "cuisine\_1",

"ns" : "test.restaurants"

}

]

**> db.restaurants.dropIndexes()**

{

"nIndexesWas" : 2,

"msg" : "non-\_id indexes dropped for collection",

"ok" : 1

}

**> db.restaurants.getIndexes() -->\_id index는 삭제불가**

[

{

"v" : 2,

"key" : {

"\_id" : 1

},

"name" : "\_id\_",

"ns" : "test.restaurants"

}

]

**> use admin**

**> db.shutdownServer()**

**> exit**