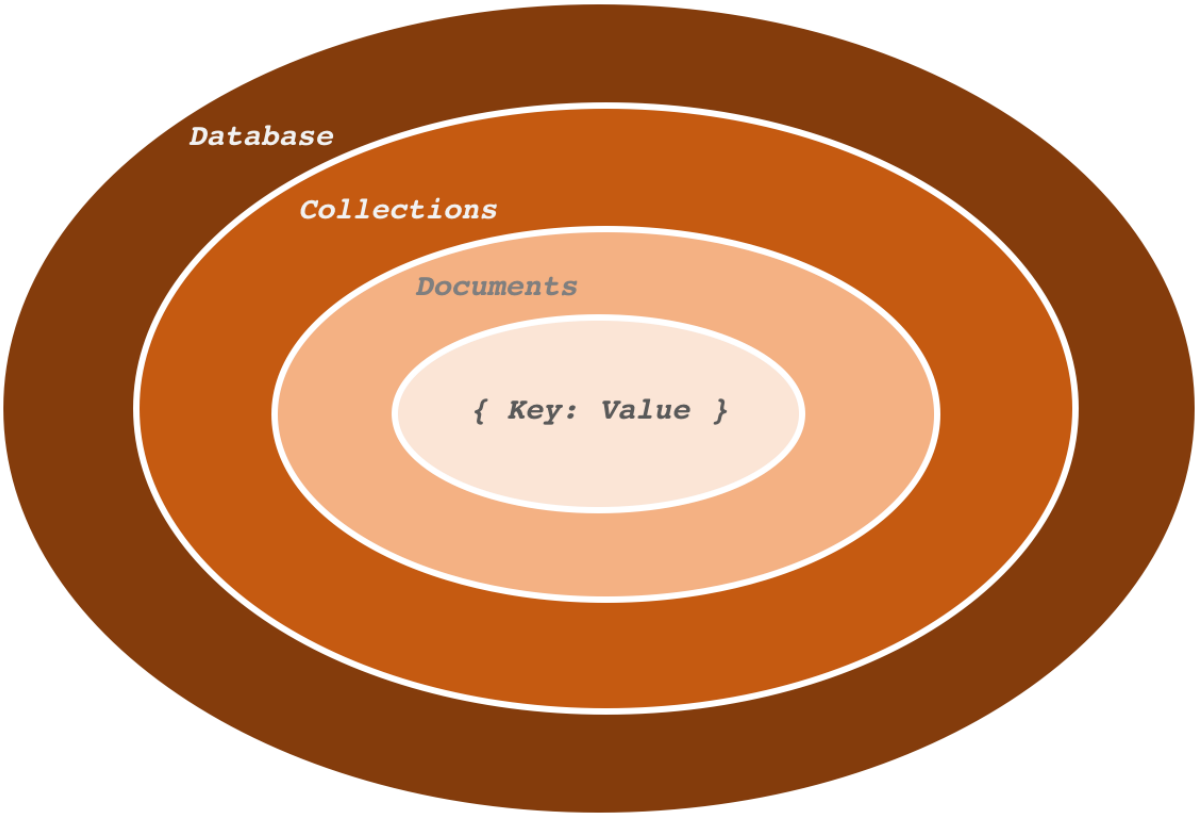
4. 기본 명령어 익히기

데이터베이스, 컬렉션, 도큐먼트



1. Database

- 데이터베이스는 컬렉션의 물리적 컨테이너 입니다. 하나의 데이터베이스에는 보통 여러개의 컬렉션을 가지고 있습니다.

2. Collection

- 컬렉션은 몽고DB Document 의 그룹이며 RDBMS 의 예를 들면 Table 과 개념과 유사합니다.
- 컬렉션은 단일 데이터베이스에 존재합니다.
- 컬렉션은 스키마를 강요하지 않습니다. 따라서 컬렉션 내부의 도큐먼트는 서로 다른 필드를 가질수 있습니다.
- 컬렉션 안에 도큐먼트는 일반적으로 서로 유사한 하거나 관련된 목적이 있습니다.

3. Document

- Docuemtn 는 하나의 키(key) 와 값(value)의 집합으로 이루어져 있으며 동적 스키마 입니다.
- 동적 스키마는 동일한 컬랙션 내의 도큐먼트가 동일한 필드 또는 구조를 가지필요 없음을 의미한다.
- 그리고 동일한 필드안에 다른타입의 데이타를 보유할수 있음을 의미합니다.

RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
Column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (Default _id)

기본적인 명령어 익히기

종류	설명
db.help	DB Method 종류 출력
db.<Coll>.help()	Mongo DB에서 제공하는 함수의 종류 출력
ls("디스크명")	현제 운영체제 경로에 존재하는 파일의 종류를 출력
getMemInfo()	현재 할당된 메모리 상태 출력
help keys	현재 라인에 작성된 내용을 편집하는 방법을 출력
show dbs	현제 생성되어 있는 데이터 명과 크기 출력
show collections	DB내에 생성되어 있는 COLLECTION 종류 출력
hostname	현재 서버의 호스트 명을 출력

. help

```
repmongo:PRIMARY> help  <-- mongo 툴에서 사용할 수 있는 환경 명령어를 출력한다.
        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
```

. db 종료

```
repmongo:SECONDARY> db.shutdownServer()
2021-01-10T23:01:51.896+0900 I  NETWORK  [js] DBClientConnection failed to receive message from localhost:10000 - HostUnreachab
server should be down...
2021-01-10T23:01:51.905+0900 I  NETWORK  [js] trying reconnect to localhost:10000 failed
2021-01-10T23:01:51.905+0900 I  NETWORK  [js] reconnect localhost:10000 failed failed
2021-01-10T23:01:51.908+0900 I  NETWORK  [js] trying reconnect to localhost:10000 failed
2021-01-10T23:01:51.908+0900 I  NETWORK  [js] reconnect localhost:10000 failed failed
>
```

. 현재 DB의 세션 정보

```
repmongo:PRIMARY> db.currentOp()
{
        "inprog" : [
                {
                        "type" : "op",
                        "host" : "mongodb:10000",
                        "desc" : "conn22",
                        "connectionId" : 22,
                        "client" : "127.0.0.1:36396",
                        "appName" : "MongoDB Shell",
                        "clientMetadata" : {
                                "application" : {
                                        "name" : "MongoDB Shell"
                                },
                                "driver" : {
                                        "name" : "MongoDB Internal Client",
                                        "version" : "4.2.12-rc0"
                                },
                                "os" : {
                                        "type" : "Linux",
                                        "name" : "CentOS Linux release 7.9.2009 (Core)",
                                        "architecture" : "x86_64",
                                        "version" : "Kernel 3.10.0-1160.el7.x86_64"
                                }
                        },
                        "active" : true,
                        "currentOpTime" : "2021-01-10T23:53:08.200+0900",
                        "opid" : 7713,
                        "lsid" : {
                                "id" : UUID("74d73543-c5a6-4e51-a46d-0eb37b62b4da"),
                                "uid" : BinData(0,"47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU=")
                        },
                        "secs_running" : NumberLong(0),
                        "microsecs_running" : NumberLong(202),
                        "op" : "command",
                        "ns" : "admin.$cmd.aggregate",
                        "command" : {
                                "currentOp" : 1,
                                "lsid" : {
                                        "id" : UUID("74d73543-c5a6-4e51-a46d-0eb37b62b4da")
                                },
                                "$clusterTime" : {
                                        "clusterTime" : Timestamp(1610290343, 1),
                                        "signature" : {
                                                "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                                                "keyId" : NumberLong(0)
                                        }
                                },
                                "$db" : "admin"
                        },
                        "numYields" : 0,
                        "locks" : {

                        },
                        "waitingForLock" : false,
                        "lockStats" : {

                        },
                        "waitingForFlowControl" : false,
                        "flowControlStats" : {

                        }
                },
                {
                        "type" : "op",
                        "host" : "mongodb:10000",
                        "desc" : "NoopWriter",
                        "active" : true,
                        "currentOpTime" : "2021-01-10T23:53:08.200+0900",
                        "opid" : 7679,
                        "op" : "none",
                        "ns" : "",
                        "command" : {

                        },
                        "numYields" : 0,
                        "locks" : {

                        },
                        "waitingForLock" : false,
                        "lockStats" : {
```

```
                },
                "waitingForFlowControl" : false,
                "flowControlStats" : {

                }
        },
        {
                "type" : "op",
                "host" : "mongodb:10000",
                "desc" : "WT-OplogTruncaterThread-local.oplog.rs",
                "active" : true,
                "currentOpTime" : "2021-01-10T23:53:08.200+0900",
                "opid" : 487,
                "op" : "none",
                "ns" : "",
                "command" : {

                },
                "numYields" : 0,
                "locks" : {

                },
                "waitingForLock" : false,
                "lockStats" : {
                        "ReplicationStateTransition" : {
                                "acquireCount" : {
                                        "w" : NumberLong(2)
                                }
                        },
                        "Global" : {
                                "acquireCount" : {
                                        "w" : NumberLong(2)
                                },
                                "acquireWaitCount" : {
                                        "w" : NumberLong(1)
                                },
                                "timeAcquiringMicros" : {
                                        "w" : NumberLong(37656)
                                }
                        },
                        "Database" : {
                                "acquireCount" : {
                                        "w" : NumberLong(1)
                                }
                        }
                },
                "waitingForFlowControl" : false,
                "flowControlStats" : {
                        "acquireCount" : NumberLong(1),
                        "timeAcquiringMicros" : NumberLong(1)
                }
        },
        {
                "type" : "op",
                "host" : "mongodb:10000",
                "desc" : "waitForMajority",
                "active" : true,
                "currentOpTime" : "2021-01-10T23:53:08.200+0900",
                "opid" : 2,
                "op" : "none",
                "ns" : "",
                "command" : {

                },
                "numYields" : 0,
                "waitingForLatch" : {
                        "timestamp" : ISODate("2021-01-10T14:35:41.622Z"),
                        "captureName" : "WaitForMaoritySerivce::_mutex"
                },
                "locks" : {

                },
                "waitingForLock" : false,
                "lockStats" : {

                },
                "waitingForFlowControl" : false,
                "flowControlStats" : {

                }
        },
        {
                "type" : "op",
                "host" : "mongodb:10000",
                "desc" : "rsSync-0",
                "active" : true,
```

```
                "currentOpTime" : "2021-01-10T23:53:08.200+0900",
                "effectiveUsers" : [
                        {
                                "user" : "__system",
                                "db" : "local"
                        }
                ],
                "opid" : 7707,
                "op" : "none",
                "ns" : "",
                "command" : {

                },
                "numYields" : 0,
                "locks" : {

                },
                "waitingForLock" : false,
                "lockStats" : {
                        "ParallelBatchWriterMode" : {
                                "acquireCount" : {
                                        "r" : NumberLong(1)
                                }
                        },
                        "ReplicationStateTransition" : {
                                "acquireCount" : {
                                        "w" : NumberLong(1)
                                }
                        },
                        "Global" : {
                                "acquireCount" : {
                                        "r" : NumberLong(1)
                                }
                        },
                        "Database" : {
                                "acquireCount" : {
                                        "r" : NumberLong(1)
                                }
                        },
                        "Collection" : {
                                "acquireCount" : {
                                        "r" : NumberLong(1)
                                }
                        },
                        "Mutex" : {
                                "acquireCount" : {
                                        "r" : NumberLong(1)
                                }
                        }
                },
                "waitingForFlowControl" : false,
                "flowControlStats" : {

                }
        },
        {
                "type" : "op",
                "host" : "mongodb:10000",
                "desc" : "conn23",
                "connectionId" : 23,
                "client" : "10.0.2.15:45590",
                "clientMetadata" : {
                        "driver" : {
                                "name" : "NetworkInterfaceTL",
                                "version" : "4.2.12-rc0"
                        },
                        "os" : {
                                "type" : "Linux",
                                "name" : "CentOS Linux release 7.9.2009 (Core)",
                                "architecture" : "x86_64",
                                "version" : "Kernel 3.10.0-1160.el7.x86_64"
                        }
                },
                "active" : true,
                "currentOpTime" : "2021-01-10T23:53:08.200+0900",
                "opid" : 7687,
                "secs_running" : NumberLong(4),
                "microsecs_running" : NumberLong(4500724),
                "op" : "getmore",
                "ns" : "local.oplog.rs",
                "command" : {
                        "getMore" : NumberLong("2458148265191136876"),
                        "collection" : "oplog.rs",
                        "batchSize" : 13981010,
                        "maxTimeMS" : NumberLong(5000),
                        "term" : NumberLong(2),
                        "lastKnownCommittedOpTime" : {
```

```
                        "ts" : Timestamp(1610290383, 1),
                        "t" : NumberLong(2)
                },
                "$replData" : 1,
                "$oplogQueryData" : 1,
                "$readPreference" : {
                        "mode" : "secondaryPreferred"
                },
                "$clusterTime" : {
                        "clusterTime" : Timestamp(1610290383, 1),
                        "signature" : {
                                "hash" : BinData(0,"q4U8SjI3qR8a+s6eGGT8hdGzvaw="),
                                "keyId" : NumberLong("6916140262850822147")
                        }
                },
                "$db" : "local"
        },
        "planSummary" : "COLLSCAN",
        "cursor" : {
                "cursorId" : NumberLong("2458148265191136876"),
                "createdDate" : ISODate("2021-01-10T14:38:44.956Z"),
                "lastAccessDate" : ISODate("2021-01-10T14:53:03.699Z"),
                "nDocsReturned" : NumberLong(91),
                "nBatchesReturned" : NumberLong(280),
                "noCursorTimeout" : false,
                "tailable" : true,
                "awaitData" : true,
                "originatingCommand" : {
                        "find" : "oplog.rs",
                        "filter" : {
                                "ts" : {
                                        "$gte" : Timestamp(1610289504, 1)
                                }
                        },
                        "tailable" : true,
                        "oplogReplay" : true,
                        "awaitData" : true,
                        "maxTimeMS" : NumberLong(60000),
                        "batchSize" : 13981010,
                        "term" : NumberLong(2),
                        "readConcern" : {
                                "afterClusterTime" : Timestamp(0, 1)
                        },
                        "$replData" : 1,
                        "$oplogQueryData" : 1,
                        "$readPreference" : {
                                "mode" : "secondaryPreferred"
                        },
                        "$clusterTime" : {
                                "clusterTime" : Timestamp(1610289523, 1),
                                "signature" : {
                                        "hash" : BinData(0,"exXLH1+JIFgpV2iG6qedOvgrkPM="),
                                        "keyId" : NumberLong("6916140262850822147")
                                }
                        },
                        "$db" : "local"
                },
                "operationUsingCursorId" : NumberLong(7687)
        },
        "numYields" : 2,
        "locks" : {

        },
        "waitingForLock" : false,
        "lockStats" : {
                "ReplicationStateTransition" : {
                        "acquireCount" : {
                                "w" : NumberLong(2)
                        }
                },
                "Global" : {
                        "acquireCount" : {
                                "r" : NumberLong(2)
                        }
                },
                "Database" : {
                        "acquireCount" : {
                                "r" : NumberLong(2)
                        }
                },
                "Mutex" : {
                        "acquireCount" : {
                                "r" : NumberLong(1)
                        }
                },
                "oplog" : {
                        "acquireCount" : {
```

```
                                        "r" : NumberLong(2)
                                }
                        }
                },
                "waitingForFlowControl" : false,
                "flowControlStats" : {

                }
        },
        {
                "type" : "op",
                "host" : "mongodb:10000",
                "desc" : "monitoring-keys-for-HMAC",
                "active" : true,
                "currentOpTime" : "2021-01-10T23:53:08.200+0900",
                "opid" : 1409,
                "op" : "none",
                "ns" : "",
                "command" : {

                },
                "numYields" : 0,
                "waitingForLatch" : {
                        "timestamp" : ISODate("2021-01-10T14:37:53.628Z"),
                        "captureName" : "PeriodicRunner::_mutex"
                },
                "locks" : {

                },
                "waitingForLock" : false,
                "lockStats" : {

                },
                "waitingForFlowControl" : false,
                "flowControlStats" : {

                }
        },
        {
                "type" : "op",
                "host" : "mongodb:10000",
                "desc" : "ReplBatcher",
                "active" : true,
                "currentOpTime" : "2021-01-10T23:53:08.200+0900",
                "opid" : 7709,
                "op" : "none",
                "ns" : "",
                "command" : {

                },
                "numYields" : 0,
                "locks" : {

                },
                "waitingForLock" : false,
                "lockStats" : {

                },
                "waitingForFlowControl" : false,
                "flowControlStats" : {

                }
        }
    ],
    "ok" : 1,
    "$clusterTime" : {
            "clusterTime" : Timestamp(1610290383, 1),
            "signature" : {
                    "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                    "keyId" : NumberLong(0)
            }
    },
    "operationTime" : Timestamp(1610290383, 1)
}
```


. Collection 생성

capped : 제한 크기 Collection

일정 크기 안에 데이터만 사용하고 꽉차면 앞에 생성한 데이터 덮어 씀

※ 주의 : 삭제 불가능, document가 이동되는 갱신 불가능

장점 : 입력 속도 빠름

```
repmongo:PRIMARY> db.createCollection("emp",{capped:true, size: 100000,max:200000});
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1610290561, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1610290561, 1)
}
--capped : 해당 공간이 모두 사용되면 다시 처음부터 재 사용할 수 있는 데이터구조를 생성할때 해당 Collecion의 최초 크기 지정 가능
```

. collection 조회

```
repmongo:PRIMARY> show collections
emp
```

데이터 베이스 생성, 조회, 삭제

. 생성

```
repmongo:PRIMARY> use dropDB
switched to db dropDB
repmongo:PRIMARY> show dbs -- use 사용하여 만들었지만 나타나지 않음
MongoDB  0.000GB
admin    0.000GB
config   0.000GB
local    0.000GB
repmongo:PRIMARY> db.createCollection("emp",{capped:true, size: 100000,max:200000});
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1610290960, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1610290960, 1)
}
```

. 조회

```
repmongo:PRIMARY> show dbs
MongoDB  0.000GB
admin    0.000GB
config   0.000GB
dropDB   0.000GB
local    0.000GB
```

. 삭제

```
repmongo:PRIMARY> db.dropDatabase();
{
        "dropped" : "dropDB",
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1610291104, 2),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
```

```
        },
        "operationTime" : Timestamp(1610291104, 2)
}
```

컬렉션 명령어

. 컬렉션 현재 상태 및 정보분석

```
repmongo:PRIMARY> db.emp.stats()
{
        "ns" : "MongoDB.emp",
        "size" : 0,
        "count" : 0,
        "storageSize" : 4096,
        "capped" : true,
        "max" : 200000,
        "maxSize" : 100096,
        "sleepCount" : 0,
        "sleepMS" : 0,
        "wiredTiger" : {
                "metadata" : {
                        "formatVersion" : 1
                },
                "creationString" : "access_pattern_hint=none,allocation_size=4KB,app_metadata=(formatVersion=1),assert=(commit_
                "type" : "file",
                "uri" : "statistics:table:collection-23--944410322461979814",
                "LSM" : {
                        .....
                },
                "block-manager" : {
                     .....
                },
                "btree" : {
                      .......
                },
                "cache" : {
               ........
                },
                "compression" : {
                       ......
                },
                "cursor" : {
                     ......
                },
                "reconciliation" : {
                        .....
                },
                "session" : {
                        "object compaction" : 0
                },
                "transaction" : {
                        "update conflicts" : 0
                }
        },
        "nindexes" : 1,
        "indexBuilds" : [ ],
        "totalIndexSize" : 4096,
        "indexSizes" : {
                "_id_" : 4096
        },
        "scaleFactor" : 1,
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1610291493, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1610291493, 1)
}
```

. 컬렉션 명 변경

```
repmongo:PRIMARY> db.emp.renameCollection("employee")
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1610291663, 1),
```

```
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1610291663, 1)
}
```

. 컬렉션 삭제

```
repmongo:PRIMARY> db.employee.drop()
true
```

도큐먼트 생성

. 단일 도큐먼트 생성

```
repmongo:PRIMARY> db.emp.insert({empno:3,ename:'홍길동'})
WriteResult({ "nInserted" : 1 })
```

. 다수 도큐먼트 생성

```
repmongo:PRIMARY> db.ACCOUNT.insert([{ACTNO:13265487,CUSNM:'다라마'},{ACTNO:87654321, CUSNO:'라마바'}]);
BulkWriteResult({
        "writeErrors" : [ ],
        "writeConcernErrors" : [ ],
        "nInserted" : 2,
        "nUpserted" : 0,
        "nMatched" : 0,
        "nModified" : 0,
        "nRemoved" : 0,
        "upserted" : [ ]
})

repmongo:PRIMARY> var heightSize = 170
repmongo:PRIMARY> for(var i = 0 ; i < 3 ; i++){     db.employee.insertOne({ ename : "김포문" + i, depart : "영업팀", status : "B",
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5ffb25f3a7b91cb7c3815cf3")
}
repmongo:PRIMARY>
repmongo:PRIMARY>
repmongo:PRIMARY> db.employee.find()
{ "_id" : ObjectId("5ffb25f3a7b91cb7c3815cf1"), "ename" : "김포문0", "depart" : "영업팀", "status" : "B", "height" : 160 }
{ "_id" : ObjectId("5ffb25f3a7b91cb7c3815cf2"), "ename" : "김포문1", "depart" : "영업팀", "status" : "B", "height" : 163 }
{ "_id" : ObjectId("5ffb25f3a7b91cb7c3815cf3"), "ename" : "김포문2", "depart" : "영업팀", "status" : "B", "height" : 166 }
```

. 시퀀스 생성 및 insert

```
--함수를 통해 시퀀스 생성하여 insert
repmongo:PRIMARY> function seq_no(name){
...  var ret = db.seq_no.findAndModify({query : {_id : name},
...  return ret.next; }
repmongo:PRIMARY> db.ord_no.insert({_id:seq_no("ord_no"), name: "스타워즈레고"})
WriteResult({ "nInserted" : 1 })
repmongo:PRIMARY>  db.ord_no.insert({_id:seq_no("ord_no"), name:"닌자고레고"})
WriteResult({ "nInserted" : 1 })
repmongo:PRIMARY> db.ord_no.find()
{ "_id" : 1, "name" : "스타워즈레고" }
{ "_id" : 2, "name" : "닌자고레고" }
```

. 도큐먼트 조회

```
repmongo:PRIMARY> db.emp.find()
{ "_id" : ObjectId("5ffb1a75be60e03d4945f42c"), "empno" : 3, "ename" : "홍길동" }

repmongo:PRIMARY> db.emp.findOne({empno:3})
{
        "_id" : ObjectId("5ffb1a75be60e03d4945f42c"),
        "empno" : 3,
        "ename" : "홍길동"
}
```

. Update

```
repmongo:PRIMARY> db.emp.find()
{ "_id" : ObjectId("5ffb1a75be60e03d4945f42c"), "empno" : 3, "ename" : "홍길동" }
{ "_id" : ObjectId("5ffb1b38a7b91cb7c3815ce9"), "empno" : 1, "ename" : "가나다" }
repmongo:PRIMARY> db.emp.update({empno:1},{$set:{ename:'라마바'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
repmongo:PRIMARY> db.emp.find()
{ "_id" : ObjectId("5ffb1a75be60e03d4945f42c"), "empno" : 3, "ename" : "홍길동" }
{ "_id" : ObjectId("5ffb1b38a7b91cb7c3815ce9"), "empno" : 1, "ename" : "라마바" }
```

. Delete

```
repmongo:PRIMARY> db.createCollection("ACCOUNT")
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1610292377, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1610292377, 1)
}
repmongo:PRIMARY> db.ACCOUNT.insert({ACTNO:12345678,CUSNM:'홍길동'});
WriteResult({ "nInserted" : 1 })
repmongo:PRIMARY> db.ACCOUNT.insert({ACTNO:23456789,CUSNM:'가나다'});
WriteResult({ "nInserted" : 1 })
repmongo:PRIMARY> db.ACCOUNT.remove({ACTNO:12345678})
WriteResult({ "nRemoved" : 1 })
repmongo:PRIMARY> db.ACCOUNT.find()
{ "_id" : ObjectId("5ffb1cf4a7b91cb7c3815ceb"), "ACTNO" : 23456789, "CUSNM" : "가나다" }
repmongo:PRIMARY>
```

연산자

$gt

해당 값보다 더 큰 값을 가진 필드를 찾습니다. 숫자 뿐만 아니라 날짜와 ObjectId도 비교할 수 있습니다.

```
{ 필드: { $gt: 값 } }
```

$lt

해당 값보다 작은 값을 가진 필드를 찾습니다

```
{ 필드: { $lt: 값 } }
```

$gte

해당 값보다 크거나 같은 값을 가진 필드를 찾습니다

```
{ 필드: { $gte: 값 } }
```

$lte

해당 값보다 작거나 같은 값을 가진 필드를 찾습니다

```
{ 필드: { $lte: 값 } }
```

$eq

해당 값과 일치하는 값을 가진 필드를 찾습니다. 사실 그냥 { 필드: 값 } 하는 것과 같습니다.

```
{ 필드: { $eq: 값 } }
```

$ne

해당 값과 일치하지 않는 값을 가진 필드를 찾습니다.

```
{ 필드: { $ne: 값 } }
```

$in

필드의 값이 $in 안에 들어있는 값들 중 하나인 필드를 찾습니다. 아래의 예를 보면 값1, 값2, 값3, ...이 있는데 필드의 값이 그 중 하나면 반환하는 겁니다.

```
{ 필드: { $in: [ 값1, 값2, 값3, ... ] }
```

$nin

필드의 값이 $nin 안에 값들이 아닌 필드를 찾습니다. 아래의 예를 보면 값1, 값2, 값3, ...이 있는데 필드의 값이 그 값들이 아니어야 합니다.

```
{ 필드: { $nin: [ 값1, 값2, 값3, ... ] }
```

논리 쿼리

$or

$or은 여러 개의 조건 중에 적어도 하나를 만족하는 다큐먼트를 찾습니다.

```
{ $or: [{ 조건1 }, { 조건2 }, ...] }
```

$and

$and는 여러 개의 조건을 모두 만족하는 다큐먼트를 찾습니다. 조건이 간단하면 그냥 *{ 필드: 값, 필드: 값 }* 이렇게 $and가 없어도 되지만, 주로 다음과 같은 경우 때문에 $and가 필요합니다.

```
{ $and: [
  { $or: [{ 조건1 }, { 조건2 }] },
  { $or: [{ 조건3 }, { 조건4 }] }
] }
```

$nor

$nor은 여러 개의 조건을 모두 만족하지 않는 다큐먼트를 찾습니다. 조건1, 조건2 등등을 모두 만족하지 않아야합니다.

```
{ $nor: [{ 조건1 }, { 조건2 }, ...] }
```

$not

$not은 뒤의 조건을 만족하지 않는 필드를 찾습니다. $nor의 단일 버전이라고 보시면 됩니다.

```
{ $not: { 조건 } }
```

Document 조회

collection 전체 조회

```
db.wow.find({})
```

ObjectId로 조회

```
db.getCollection('wow').find({_id:ObjectId('5bee16ac39ab840015adbec3')})
```

특정 값 조회

```
db.getCollection('wow').find({"korean":"사이즈"})
```

조건 여러개로 조회

```
db.getCollection('wow').find({"korean":"사이즈","english" : "size" })
```

특정 조회 값의 특정 필드만 조회

- english 필드만 노출
- _id는 0으로 설정하지 않으면 항상 노출됨

```
db.getCollection('wow').find({"korean":"사이즈"},{_id:0, "english":1})
```

OR 쿼리

- korean 이 사이즈이거나, english가 color인 것 조회

```
db.getCollection('wow').find({
$or: [
  {"korean": "사이즈"},
  {"english":"color"}
  ]
})
```

AND 쿼리

- korean 이 사이즈이고, english가 size인 것 조회

```
db.getCollection('wow').find({
 $and : [
  {"korean":"사이즈"},
  {"english": "size"}
  ]
})
```

기간 조회

- $gte : 이상
- $lte : 이하
- $gt : 초과
- $lt : 미만
- created_at key는 추가된 상태어야 함

```
db.getCollection('wow').find({created_at:{$gte:ISODate("2019-08-27T14:00:35.386Z"),
$lte:ISODate("2019-08-29T14:00:35.386Z") }})
```

정규식 활용 조회

- english 가 s로 시작하지 않는 것 조회

```
db.products.find({english: { $not: /^s*/ }})
```

- code 가 특정글자(code_)로 시작하는 대상 조회

```
db.getCollection('products').find({"code":{$regex : /code_\w+/}})
db.getCollection('products').find({"code":{$regex : "merge_"}})
```

collection 내림차순 조회

```
db.getCollection('wow').find({}).sort({_id:-1})
```

Index 조회

```
db.getCollection('wow').getIndexes()
```

key가 존재하는 것 조회

```
db.getCollection('wow').find({"korean":{$exists:true}})
```

key 가 존재하는 결과 갯수 조회

- mongodb find 개수를 count()로 find 로 조회한 결과의 갯수를 확인 가능

```
db.getCollection('wow').find({"korean":{$exists:true}}).count()
```

Cursors

커서는 find()가 반환하는 객체이다. 3-2번 참고. 아래 함수들도 cursor를 반환하므로 중첩해서 쓸 수 있다.

5-1. 정렬

`cursor.sort(document)` 는 커서 객체를 정렬할때 사용한다. 매개변수로 들어가는 document는 `{ KEY: value }` 의 구조를 가진다. key에는 데이터 필드명이 들어가고, value에는 1 또는 -1이 들어간다. 1은 오름차순이고 -1은 내림차순이다. key에 여러 값을 넣었을 경우, 먼저 입력한 key가 우선순위가 높다.

ex. `> db.orders.find().sort( { "_id": 1 } )`

5-2. 출력갯수 제한

`cursor.limit(value)` 로 출력할 데이터 갯수를 제한할 수 있다. value에는 출력할 개수를 넣어준다.

ex. `> db.orders.find().limit(3)`

5-3. 출력시작할 부분 설정

`cursor.skip(value)` 로 출력할 데이터의 시작부분을 설정해줄 수 있다. value값 이후부터 출력한다.

ex. `> db.orders.find().skip(2)` 는 _id 3번부터 출력한다.

※ 출처

- http://blog.naver.com/PostView.nhn?blogId=bitofsky&logNo=221065925719 : cursor
- https://poiemaweb.com/mongdb-basics : Database, collection, Document
- 빅데이터 저장 및 분석을 위한 New NoSQL & mongoDB :capped, Document CRUD
- https://www.zerocho.com/category/MongoDB/post/57a17d114105f0a03bc55f74 : 연산자
- https://minimilab.tistory.com/43 : Document 조회