

MongoDB

<NoSQL의 개요>

NoSQL은 "Not Only SQL" 또는 "Non-SQL"의 줄임말로, 관계형 데이터베이스 관리 시스템(RDBMS)과는 다른 형태의 데이터 저장 및 관리 방식을 지칭하는 용어이다.

<MongoDB의 개요>

MongoDB는 NoSQL 데이터베이스의 한 종류로, 문서 지향(document-oriented) 데이터베이스 시스템

문서 지향 데이터 모델: MongoDB는 JSON과 유사한 BSON(Binary JSON) 형식의 문서를 사용한다.

동적 스키마 : 데이터 모델을 자유롭게 수정하고 확장할 수 있다.

대규모 데이터 처리 : 빠른 쓰기 및 읽기 작업을 처리하는 데 유용하다.

<MongoDB 구동 및 사용>

[MongoDB 버전 확인]

```
C:\Users\admin>mongod --version
```

```
db version v7.0.0
```

```
Build Info: {
```

```
  "version": "7.0.0",
```

```
  "gitVersion": "37d84072b5c5b9fd723db5fa133fb202ad2317f1",
```

```
  "modules": [],
```

```
  "allocator": "tcmalloc",
```

```
  "environment": {
```

```
    "distmod": "windows",
```

```
    "distarch": "x86_64",
```

```
    "target_arch": "x86_64"
```

```
  }
```

```
}
```

[DB 조작하기]

```
use testDB // testDB 사용
```

[도큐먼트에 DB삽입]

```
db.myCollection.insertOne({x: 1})
```

[현재 DB확인]

```
db
```

[DB 목록보기]

```
show dbs = show databases
```

[DB의 컬렉션 목록 보기]

```
show collections
```

[단일 문서 삭제]

```
db.collection.deleteOne({ 조건 })
```

[여러 문서 삭제]

```
db.collection.deleteMany({ 조건 })
```

[전체 문서 삭제]

```
db.collection.deleteMany({})
```

[단일 문서 수정]

```
db.collection.updateOne({ 조건 }, { $set: { 필드: 값 } })
```

[여러 문서 수정]

```
db.collection.updateMany({ 조건 }, { $set: { 필드: 값 } })
```

[조건 대체하기]

```
db.collection.replaceOne({ 조건 }, { 새로운 문서 })
```

[전체 문서 조회]

```
db.collection.find()
```

[조건에 맞는 문서 조회]

```
db.collection.find({ 조건 })
```

[문서 작성 - 컬렉션 생성]

```
use board
```

```
switched to db board
```

[자유게시판 - 문서 board에 컬렉션 생성]

```
freeboard_result = db.board.insertOne({name: "자유게시판"})
```

```
{
  acknowledged: true,
  insertedId: ObjectId("650bda605fdb941e926cc241")
}
```

```
freeboard_id = freeboard_result.insertedId
ObjectId("650bda605fdb941e926cc241")
```

[자유게시판 글 작성]

```
db.article.insertMany([
  {
    board_id: "freeboard_id",
    title: 'hello',
    content: 'hi, hello1',
    author: 'Karoid'
  },
  {
    board_id: "freeboard_id",
    title: 'hi',
    content: 'hi, hello2',
    author: 'Jeong'
  },
  {
    board_id: "freeboard_id",
    title: 'hi',
    content: 'hi, hello3',
    author: 'Hong',
    comments: [
      {
        author: 'karoid',
        content: 'hello Hong!'
      }
    ]
  }
])
```

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("650bda765fdb941e926cc242"),
    '1': ObjectId("650bda765fdb941e926cc243"),
    '2': ObjectId("650bda765fdb941e926cc244")
  }
}
```

[비밀게시판 작성]

```
secretboard_id = db.board.insertOne({name: '비밀게시판'}).insertedId
ObjectId("650bda7e5fdb941e926cc245")
```

[비밀게시판 글 작성]

```
db.article.insertOne({
  board_id: "secretboard_id",
  title: 'my Secret Title',
  content: 'hi, hello1',
  author: 'noname'
})
{
  acknowledged: true,
  insertedId: ObjectId("650bda965fdb941e926cc246")
}
```

[자유게시판 documento 글 불러오기]

```
freeboard_id = db.board.find({name: '자유게시판'}).toArray()[0]._id
ObjectId("650bda605fdb941e926cc241")
```

[자유게시판 글만 나태내게 하기]

```
db.article.find({board_id: "freeboard_id"},{
  _id: false,
  board_id: false,
  author: false,
  comments: false})
{
  title: 'hello',
  content: 'hi, hello1'
}
{
  title: 'hi',
  content: 'hi, hello2'
}
{
  title: 'hi',
  content: 'hi, hello3'
}
```

[Karoid라는 특정 documento 찾기]

```
db.article.find({"comments.0.author": "Karoid"})
```

[모든글에 대한 추천수 필드 추가]

```
db.article.updateMany({},{$set: {upvote: 0}})
```

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 4,
  modifiedCount: 4,
  upsertedCount: 0
}
```

[비밀게시판 _id를 찾아서 게시판 추천수 모두 +1 증가하기]

```
secretboard_id = db.board.find({name: "비밀게시판"}).toArray()[0]._id
ObjectId("650bda7e5fdb941e926cc245")
```

```
db.article.updateMany({board_id: secretboard_id}, {$inc: {upvote: 1}})
```

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

[댓글이 달린 자유게시판 글의 _id(작성자 Hong) 저장(ObjectId 값 저장)]

```
freeboard_id = db.board.find({name: "자유게시판"}).toArray()[0]._id
ObjectId("650bda605fdb941e926cc241")
```

```
doc_id = db.article.find({board_id: "freeboard_id", author: "Hong"}).toArray()[0]._id
ObjectId("650bda765fdb941e926cc244")
```

[Content 값 수정]

```
db.articel.updateOne({_id: 'doc_id'}, {$set: {content: 'updated'}})
```

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

[필드 안에 있는 요소 추가]

```
db.article.updateOne(
  { _id: 'doc_id' }, {
```

```
$push: {comments: {author: "Quote", content: "reply"}}
})
```

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

[Comment 필드 안에 있는 요소 추가]

```
db.article.updateOne(
  { _id: 'doc_id' }, {
    $set: { "comments.$[karoidcomment].upvote": 0 }
  },
  {
    arrayFilters: [{ "karoidcomment.author": "Quote" }]
  }
)
```

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

[board의 article 컬렉션 삭제]

```
db.article.deleteMany({})
{
  acknowledged: true,
  deletedCount: 4
}
```

[board의 article 도큐먼트 삭제]

```
db.article.drop()
true
```

[board 도큐먼트 삭제]

```
db.board.drop()
```

```
true
```

[전체 도큐먼트 삭제]

```
db.dropDatabase()
```

```
{ ok: 1, dropped: 'board' }
```