■ NoSQL 등장 배경

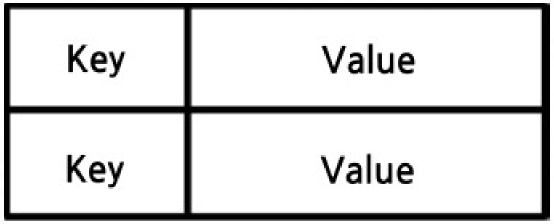
- 기존 컴퓨팅 시스템은 기업의 업무를 자동화하고 효율화하는데 목적이 있음
- 기업의 복잡한 데이터를 저장하고 데이터 간의 관계를 정의하고 분석
- 생성되는 데이터의 양은 한계를 가지고 있음
- 2000년대에 들어서면서 인터넷의 발전과 함께 SNS 활성화
- 기존의 기업 시스템에서 볼 수 없었던 대규모 데이터 생산
- 한정된 규모의 복잡성이 높은 데이터에서 단순한 대량의 데이터로 넘어감
- 기존 데이터 저장 시스템으로는 소화하기 힘든 여러가지 한계가 발생되었고 새로운 형태의 데이터 저장 기술을 요구하게 됨
- 구글의 Bigtable, 아마존의 Dynamo 논문이 발표되면서 새로운 데이터 저장 기술을 만들어 내는 시발점이 되었고 NoSQL 등장
- NoSQL은 Not Only SQL의 약자

■ NoSQL 특징

- NoSQL은 RDBMS와 다른 형태의 데이터 저장 구조를 총칭
- 제품에 따라 특성이 매우 달라서 하나의 제품군으로 정의할 수 없음
- 하나의 고성능 머신을 사용하지 않고 평범한 다수의 일반 PC를 연결하여 데이터를 처리하는 구조를 가짐
- 다수의 서버에 데이터를 상호 복제하여 분산 저장할 수 있기 때문에 특정 서버에 장애가 발생하여도 데이터 유실이 거의 없음
- 테이블 설계 시 스키마가 정해지지 않기 때문에 유동적으로 데이터 저장 가능

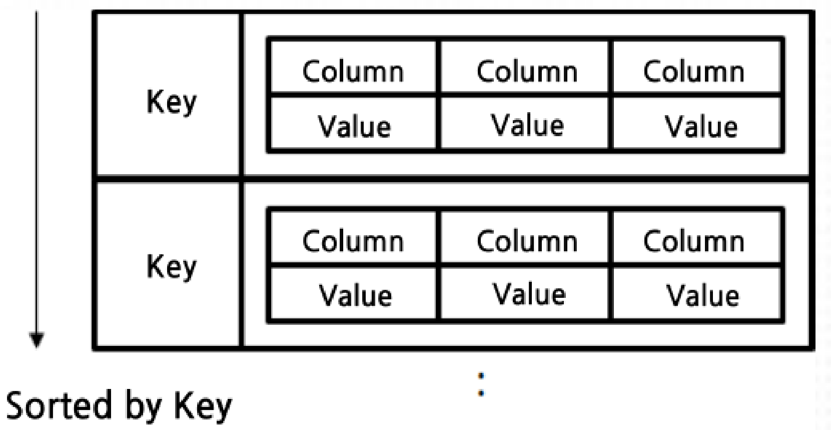
■ NoSQL 종류

- ● Key / Value Store
  - – 대부분의 NoSQL은 Key/Value 개념 지원
  - – Unique Key에 하나의 Value를 가지고 있는 형태
  - – Redis



Key	Value
Key	Value

- ● Ordered Key / Value Store
  - – 내부적으로 Key를 순서대로 정렬하여 저장
  - – Key 안에 Column:Value 조합으로 된 여러개의 필드를 가지는 구조
  - – HBase, Cassandra



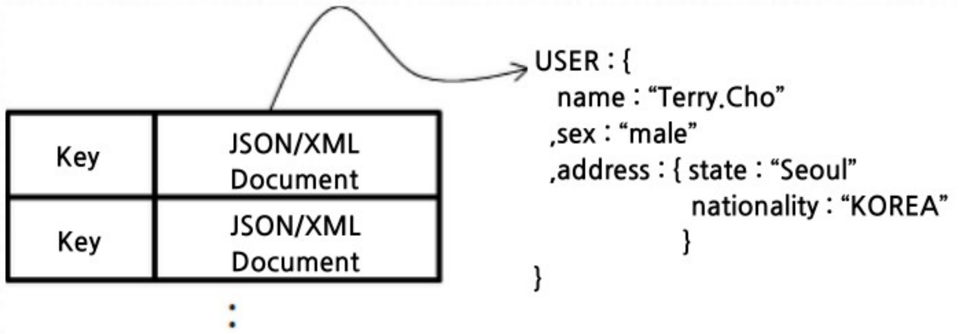
Key	Column	Column	Column
	Value	Value	Value
Key	Column	Column	Column
	Value	Value	Value

:

Sorted by Key

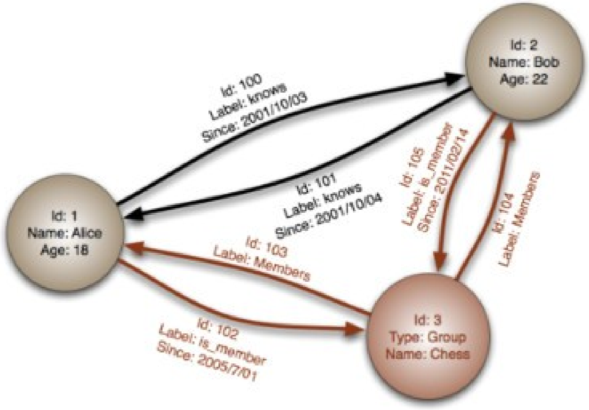
■ NoSQL 종류

- ● Document Key / Value Store
  - - Key/Value Store의 확장된 형태
  - - Document 라는 형태로 구조화된 데이터 형태(JSON, XML, YAML 등) 사용
  - - 복잡한 계층구조 표현 가능
  - - MongoDB, CouchDB



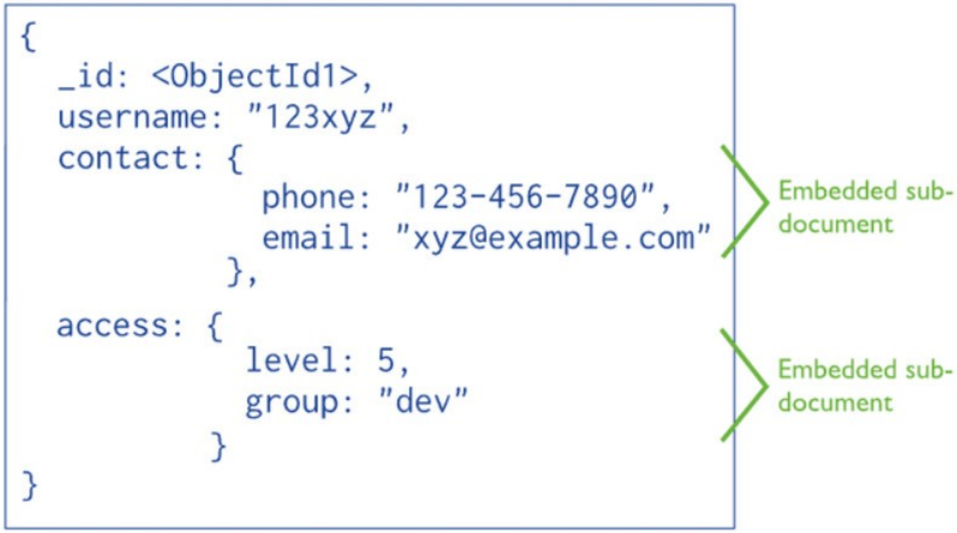
```
USER : {
  name : "Terry.Cho"
 ,sex : "male"
 ,address : { state : "Seoul"
              nationality : "KOREA"
            }
}
```

- ● Graph Store
  - - 모델링 하기 어려운 복잡한 계층 구조를 단순하고 신속하게 사용
  - - Node, Edge, Property 로 구성
  - - Neo4j



■ NoSQL 장점

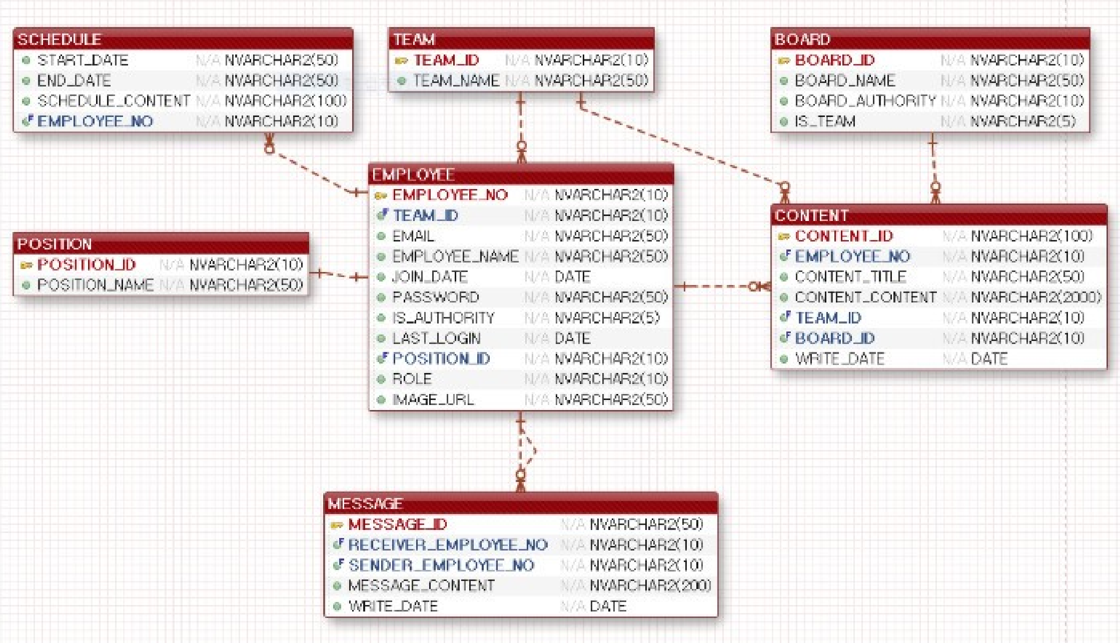
- ● 클라우드 환경에 적합
  - – 하드웨어 확장에 유연한 대처 가능
  - – 분산 / 병렬 처리 가능 (데이터가 독립적인 모델로 설계)
- ● 유연한 데이터 모델
  - – 비정형 데이터 구조
  - – 정해진 스키마를 따르지 않고 다양한 모든 형태를 저장



```
{
   _id: <ObjectId1>,
   username: "123xyz",
   contact: {
               phone: "123-456-7890",
               email: "xyz@example.com"
            },
   access: {
              level: 5,
              group: "dev"
           }
}
```

■ NoSQL 단점

- ● 각 제품별 이해 / 학습 필요
- ● 일반적인 서버 – 클라이언트 환경에 적합하지 않음
  - – 회원정보 → 게시글 작성
  - – 게시글번호 → 댓글 작성



SCHEDULE		
START_DATE	N/A	NVARCHAR2(50)
END_DATE	N/A	NVARCHAR2(50)
SCHEDULE_CONTENT	N/A	NVARCHAR2(100)
EMPLOYEE_NO	N/A	NVARCHAR2(10)

TEAM		
TEAM_ID	N/A	NVARCHAR2(10)
TEAM_NAME	N/A	NVARCHAR2(50)

BOARD		
BOARD_ID	N/A	NVARCHAR2(10)
BOARD_NAME	N/A	NVARCHAR2(50)
BOARD_AUTHORITY	N/A	NVARCHAR2(10)
IS_TEAM	N/A	NVARCHAR2(5)

EMPLOYEE		
EMPLOYEE_NO	N/A	NVARCHAR2(10)
TEAM_ID	N/A	NVARCHAR2(10)
EMAIL	N/A	NVARCHAR2(50)
EMPLOYEE_NAME	N/A	NVARCHAR2(50)
JOIN_DATE	N/A	DATE
PASSWORD	N/A	NVARCHAR2(50)
IS_AUTHORITY	N/A	NVARCHAR2(5)
LAST_LOGIN	N/A	DATE
POSITION_ID	N/A	NVARCHAR2(10)
ROLE	N/A	NVARCHAR2(10)
IMAGE_URL	N/A	NVARCHAR2(50)

POSITION		
POSITION_ID	N/A	NVARCHAR2(10)
POSITION_NAME	N/A	NVARCHAR2(50)

CONTENT		
CONTENT_ID	N/A	NVARCHAR2(100)
EMPLOYEE_NO	N/A	NVARCHAR2(10)
CONTENT_TITLE	N/A	NVARCHAR2(50)
CONTENT_CONTENT	N/A	NVARCHAR2(2000)
TEAM_ID	N/A	NVARCHAR2(10)
BOARD_ID	N/A	NVARCHAR2(10)
WRITE_DATE	N/A	DATE

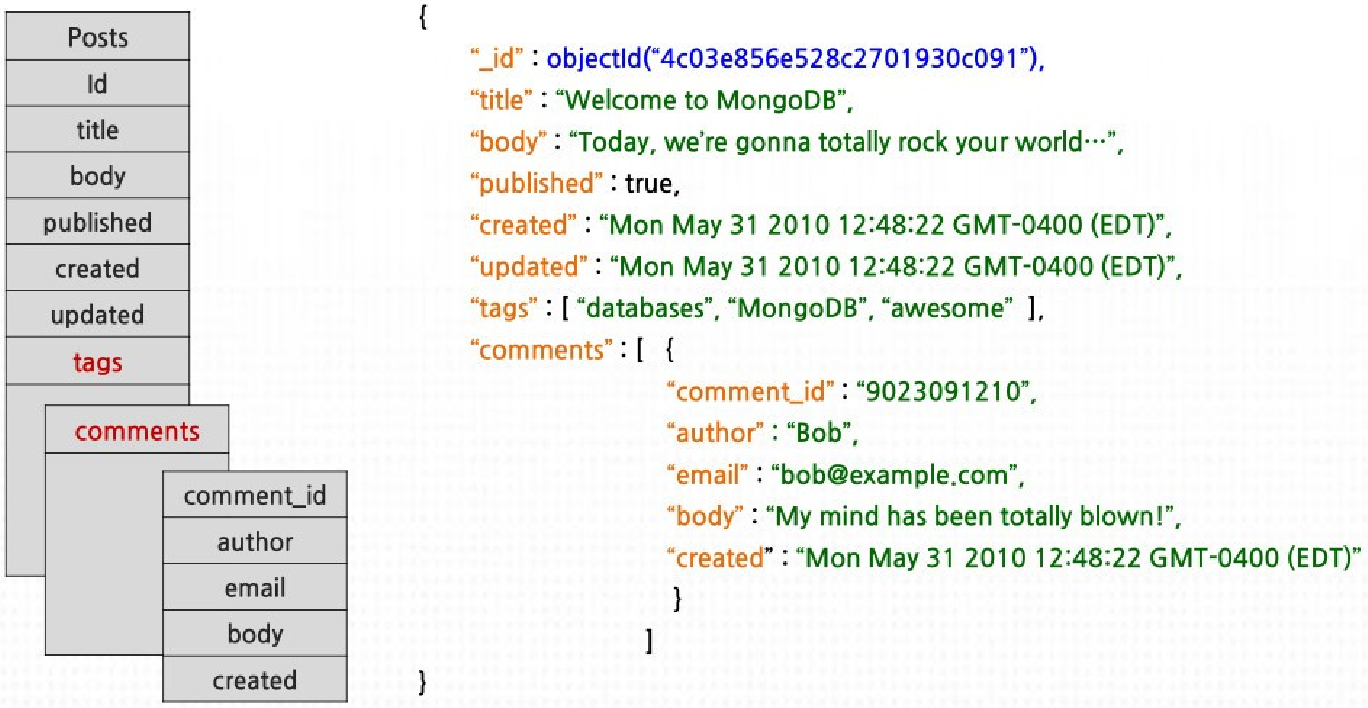
MESSAGE		
MESSAGE_ID	N/A	NVARCHAR2(50)
RECEIVER_EMPLOYEE_NO	N/A	NVARCHAR2(10)
SENDER_EMPLOYEE_NO	N/A	NVARCHAR2(10)
MESSAGE_CONTENT	N/A	NVARCHAR2(200)
WRITE_DATE	N/A	DATE

■ MongoDB

- 10gen 사에서 개발한 제품 (개발언어 C++)
- 스키마를 고정하지 않으므로 다양한 형태의 데이터 저장 가능
- JSON 형태의 데이터 사용
- Join이 불가능하기 때문에 Join이 필요없도록 데이터를 설계해야 됨
- 메모리에 의존적이라서 메모리의 크기로 성능이 좌우
- 주로 데이터를 저장하여 조회만 하고 삭제나 수정이 없는 업무에 적합
- 관계나 트랜잭션이 중요한 금융 / 결제 / 게시판 등의 서비스에는 부적합

■ Document Data Model

- ● Key / Value로 이루어진 데이터 집합
- ● Value는 일반적으로 문자열, 숫자, 날짜를 사용하며 배열([]) 또는 다른 Document를 지정하는 것도 가능
- ● 하나의 문서에 필요한 정보를 모두 담아야 함
- ● "종이문서" 한장에 모든 내용을 볼 수 있도록 설계

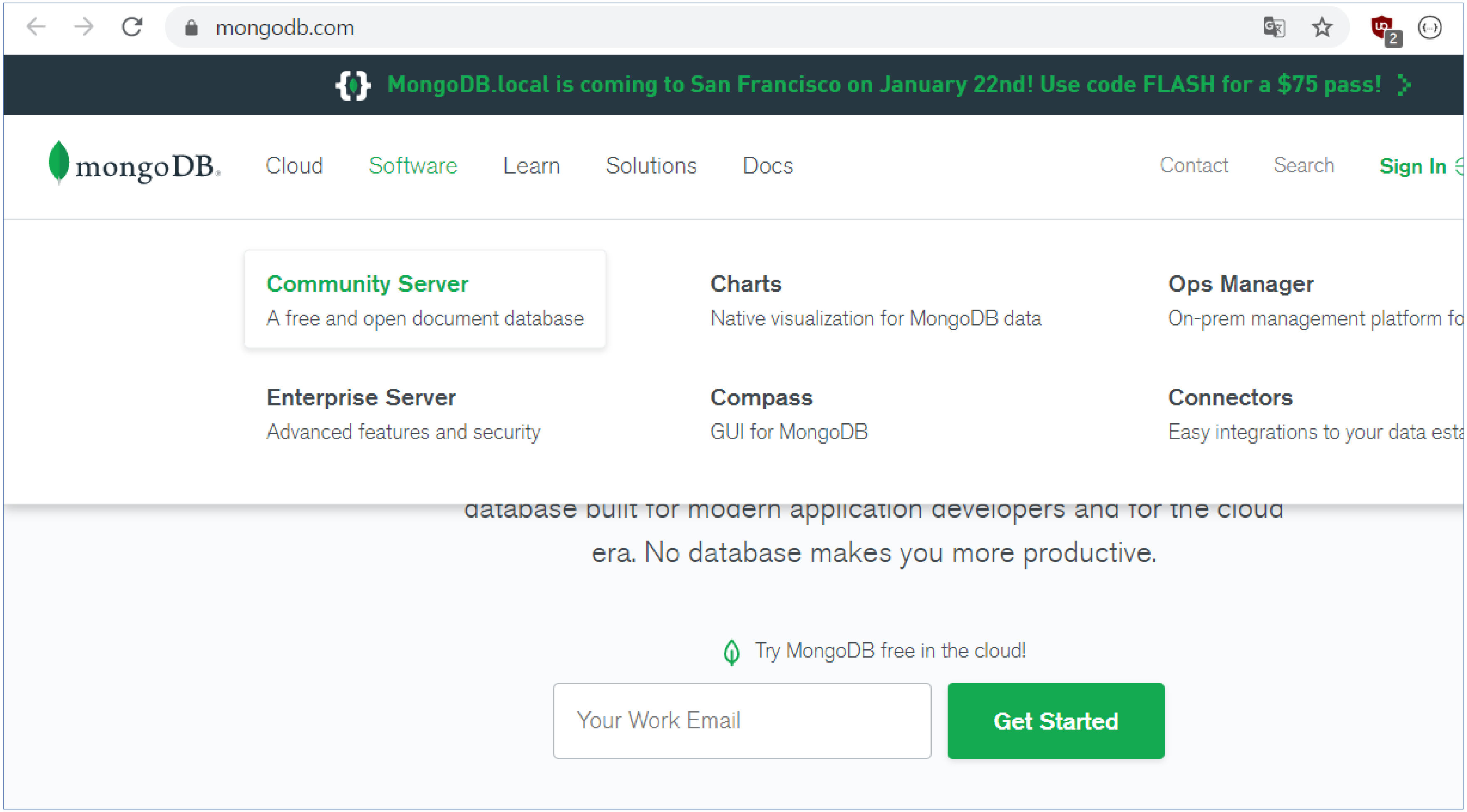


Posts
Id
title
body
published
created
updated
tags
comments
comment_id
author
email
body
created

```
{
      "_id" : objectId("4c03e856e528c2701930c091"),
      "title" : "Welcome to MongoDB",
      "body" : "Today, we're gonna totally rock your world…",
      "published" : true,
      "created" : "Mon May 31 2010 12:48:22 GMT-0400 (EDT)",
      "updated" : "Mon May 31 2010 12:48:22 GMT-0400 (EDT)",
      "tags" : [ "databases", "MongoDB", "awesome"  ],
      "comments" : [  {
                  "comment_id" : "9023091210",
                  "author" : "Bob",
                  "email" : "bob@example.com",
                  "body" : "My mind has been totally blown!",
                  "created" : "Mon May 31 2010 12:48:22 GMT-0400 (EDT)"
                  }
            ]
}
```

■ MongoDB 설치

- https://mongodb.com → Community Server



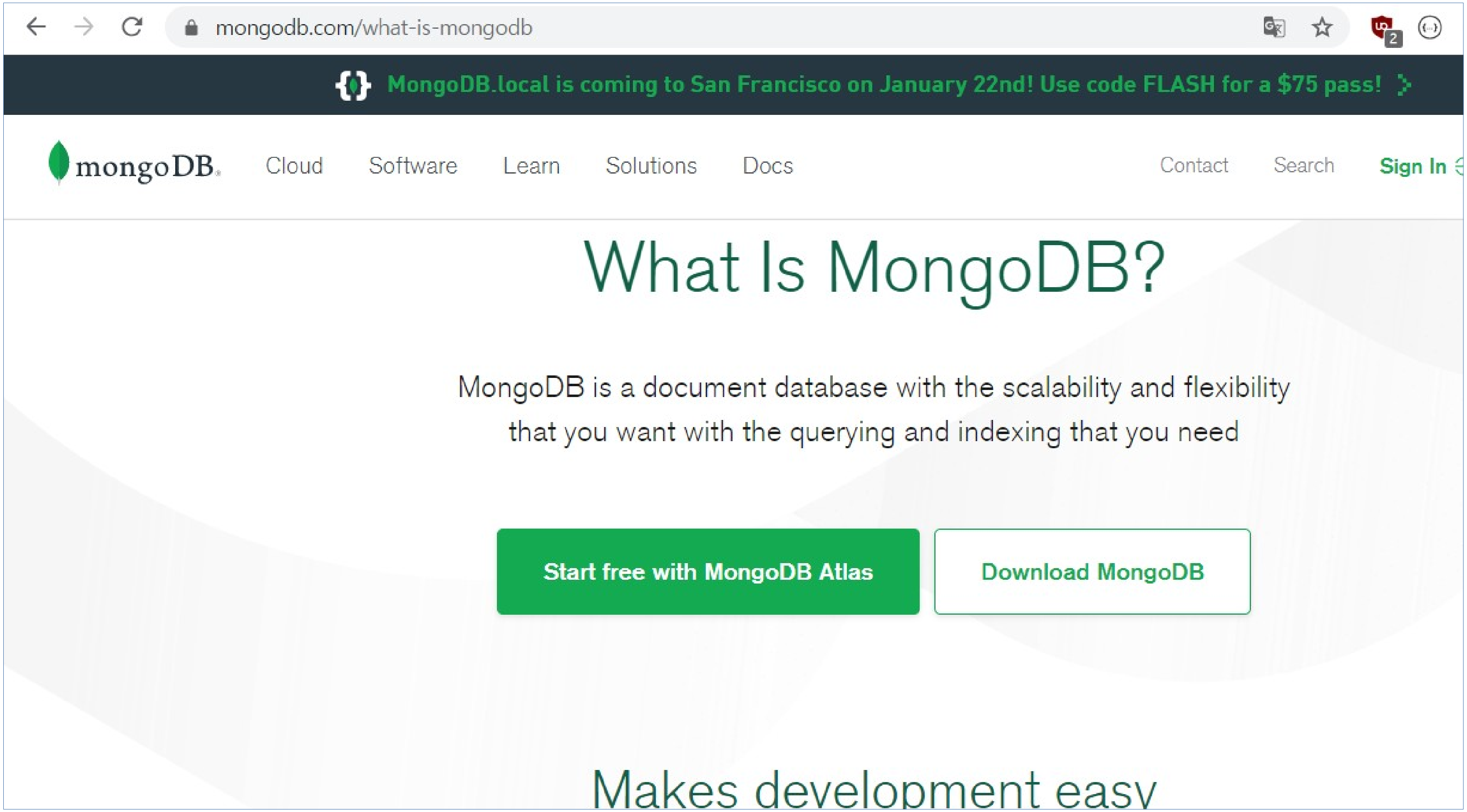
MongoDB.local is coming to San Francisco on January 22nd! Use code FLASH for a $75 pass!

database built for modern application developers and for the cloud era. No database makes you more productive.

Try MongoDB free in the cloud!

■ MongoDB 설치

- Download MongoDB



mongodb.com/what-is-mongodb

MongoDB.local is coming to San Francisco on January 22nd! Use code FLASH for a $75 pass!

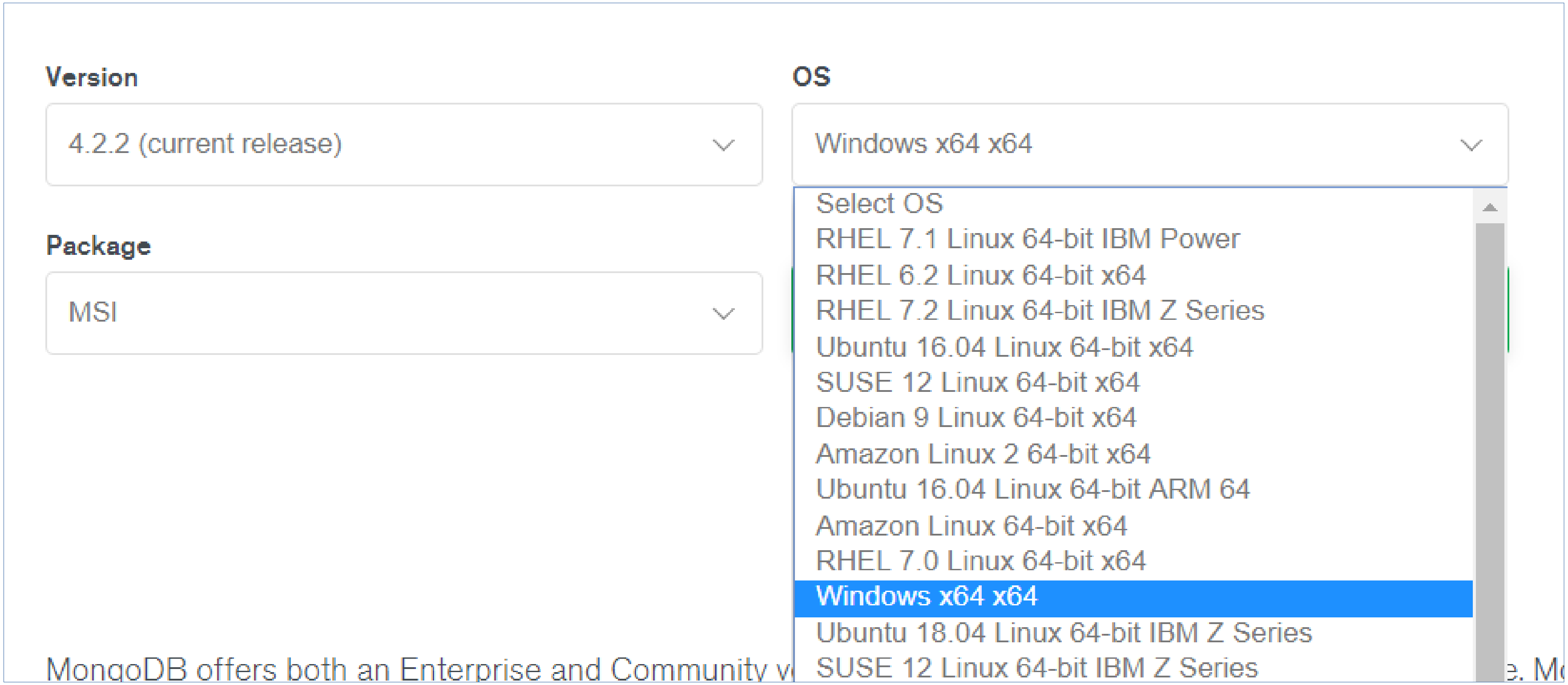
What Is MongoDB?

MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that you need

Makes development easy

■ MongoDB 설치

- ● OS 및 Version 선택 → 다운로드



Version

4.2.2 (current release)

OS

Windows x64 x64

Package

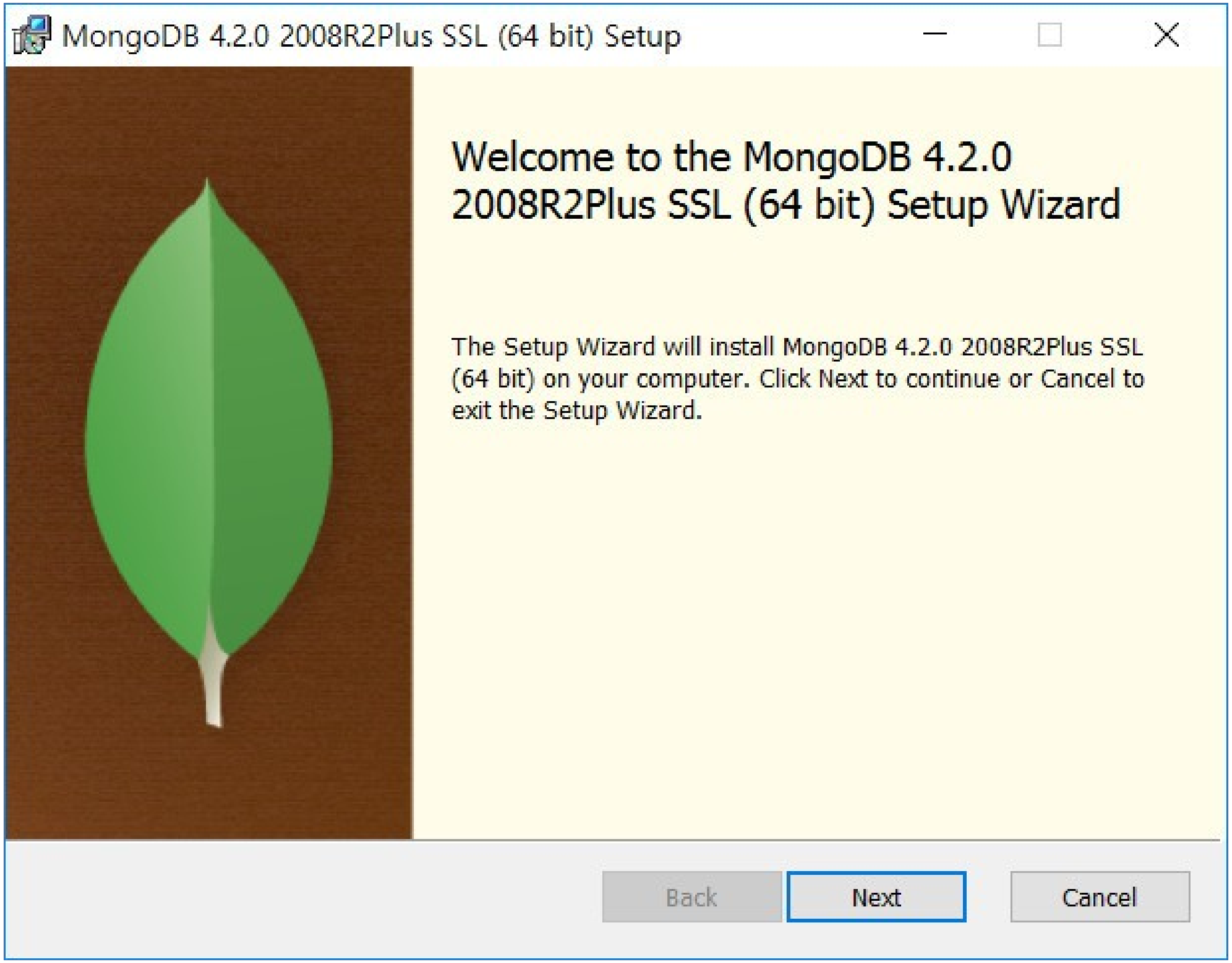
MSI

Select OS
RHEL 7.1 Linux 64-bit IBM Power
RHEL 6.2 Linux 64-bit x64
RHEL 7.2 Linux 64-bit IBM Z Series
Ubuntu 16.04 Linux 64-bit x64
SUSE 12 Linux 64-bit x64
Debian 9 Linux 64-bit x64
Amazon Linux 2 64-bit x64
Ubuntu 16.04 Linux 64-bit ARM 64
Amazon Linux 64-bit x64
RHEL 7.0 Linux 64-bit x64
Windows x64 x64
Ubuntu 18.04 Linux 64-bit IBM Z Series
SUSE 12 Linux 64-bit IBM Z Series

MongoDB offers both an Enterprise and Community v e. Mc

■ MongoDB 설치

● Install



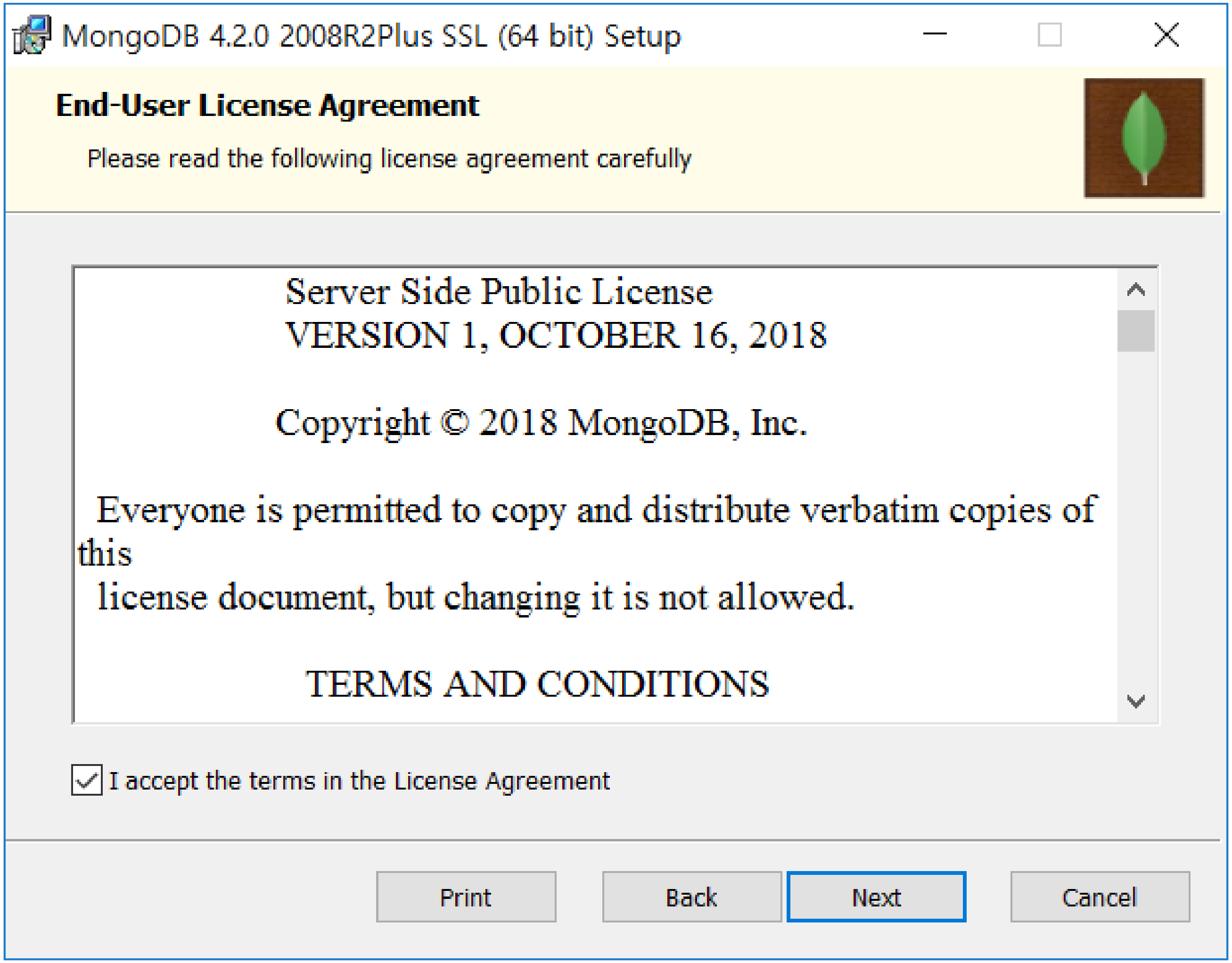
MongoDB 4.2.0 2008R2Plus SSL (64 bit) Setup

Welcome to the MongoDB 4.2.0 2008R2Plus SSL (64 bit) Setup Wizard

The Setup Wizard will install MongoDB 4.2.0 2008R2Plus SSL (64 bit) on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

■ MongoDB 설치

- ● Install



MongoDB 4.2.0 2008R2Plus SSL (64 bit) Setup

End-User License Agreement

Please read the following license agreement carefully

Server Side Public License
VERSION 1, OCTOBER 16, 2018

Copyright © 2018 MongoDB, Inc.

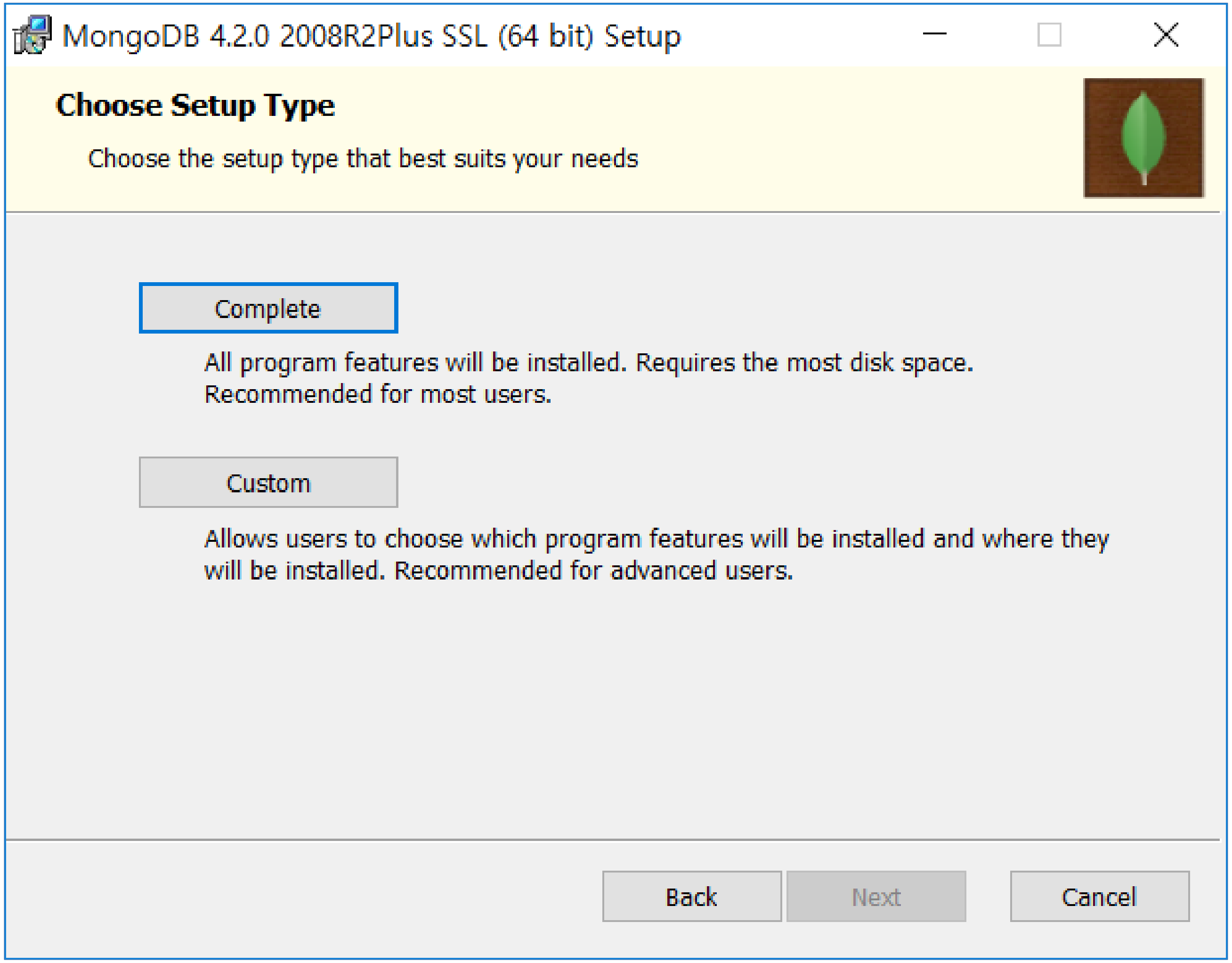
Everyone is permitted to copy and distribute verbatim copies of
this
license document, but changing it is not allowed.

TERMS AND CONDITIONS

☑ I accept the terms in the License Agreement

■ MongoDB 설치

- Install



MongoDB 4.2.0 2008R2Plus SSL (64 bit) Setup

Choose Setup Type

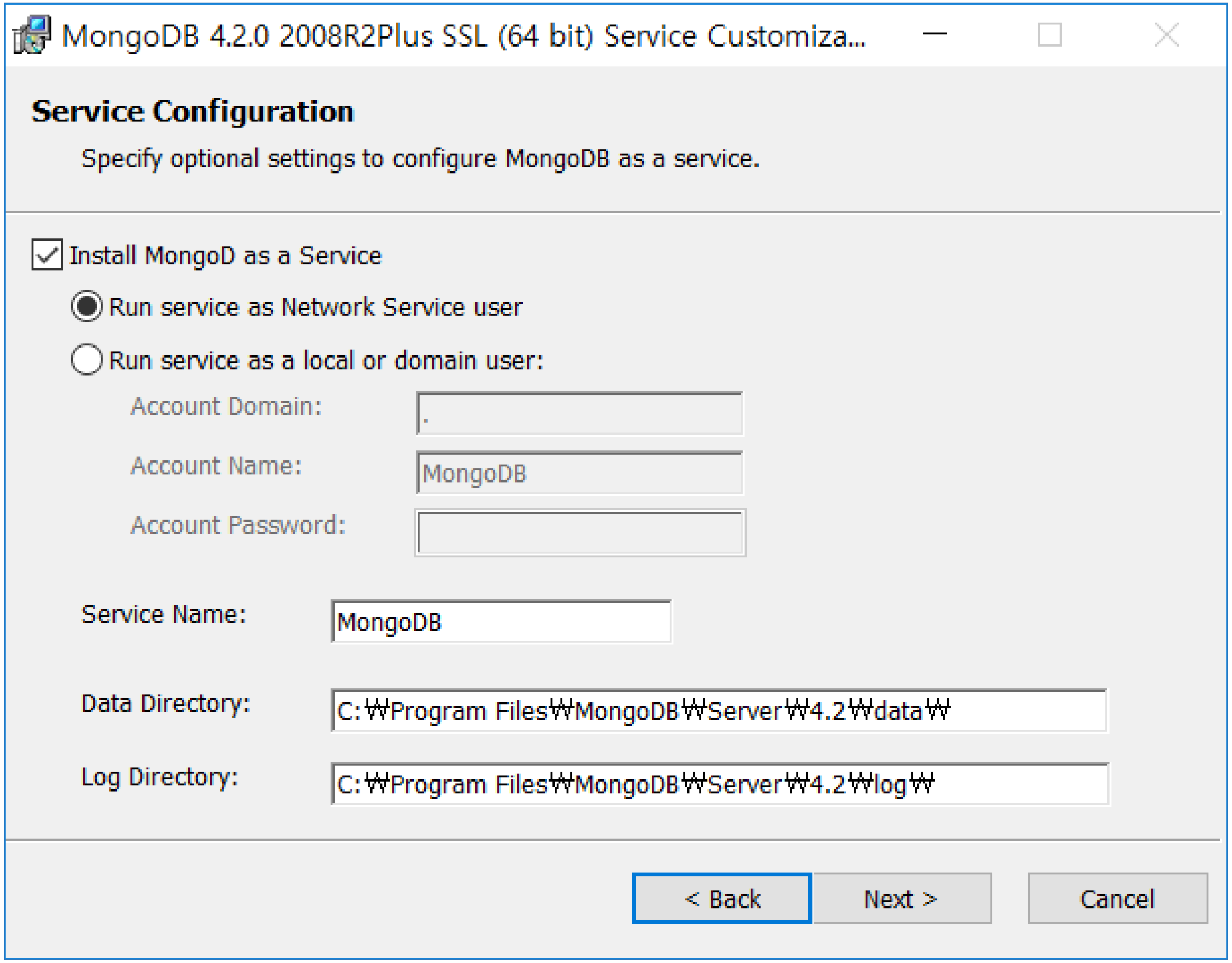
Choose the setup type that best suits your needs

All program features will be installed. Requires the most disk space. Recommended for most users.

Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.

■ MongoDB 설치

● Install



MongoDB 4.2.0 2008R2Plus SSL (64 bit) Service Customiza...

Service Configuration

Specify optional settings to configure MongoDB as a service.

☑ Install MongoD as a Service

● Run service as Network Service user

○ Run service as a local or domain user:

Account Domain: .

Account Name: MongoDB

Account Password:

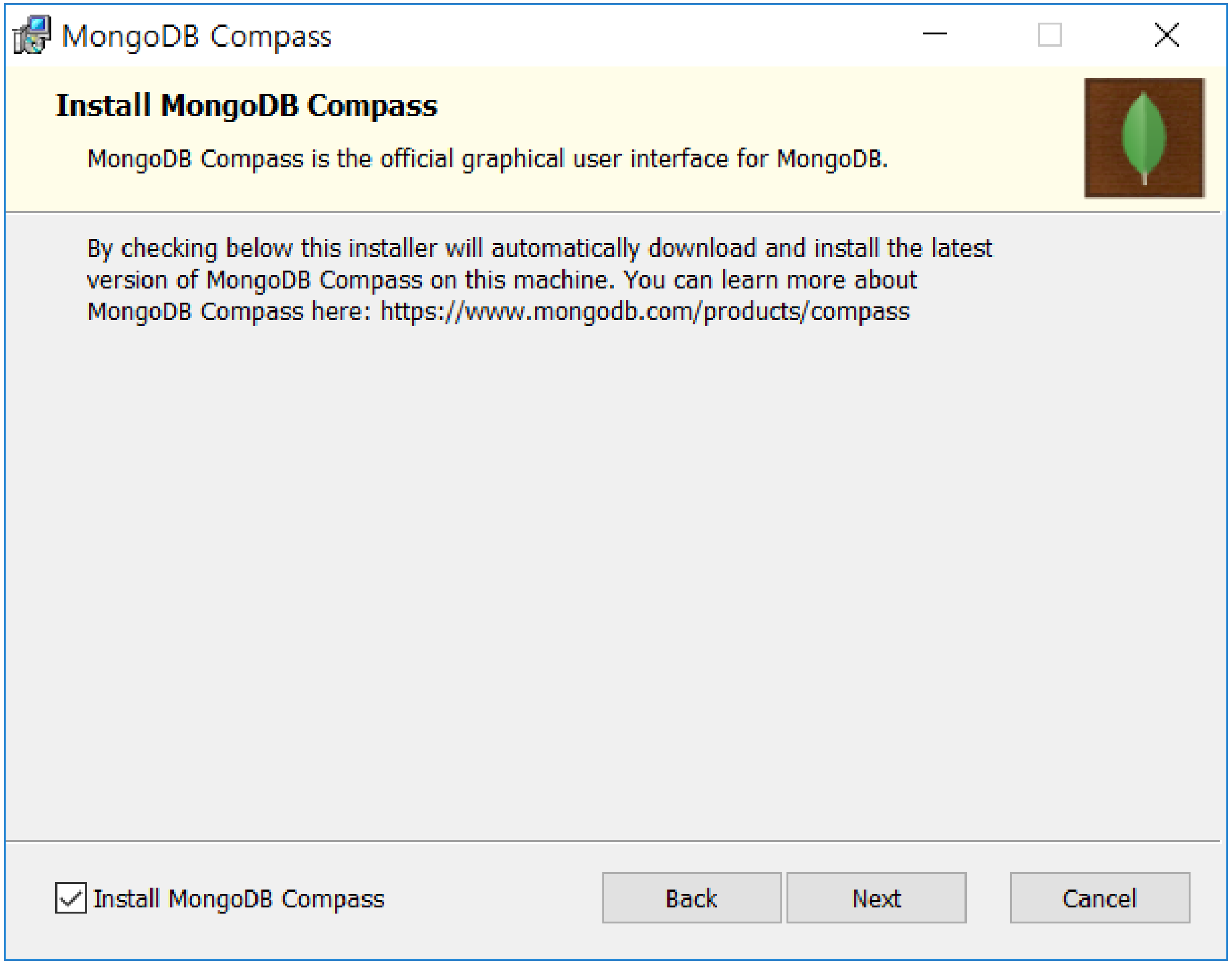
Service Name: MongoDB

Data Directory: C:₩Program Files₩MongoDB₩Server₩4.2₩data₩

Log Directory: C:₩Program Files₩MongoDB₩Server₩4.2₩log₩

■ MongoDB 설치

- ● Install



Install MongoDB Compass

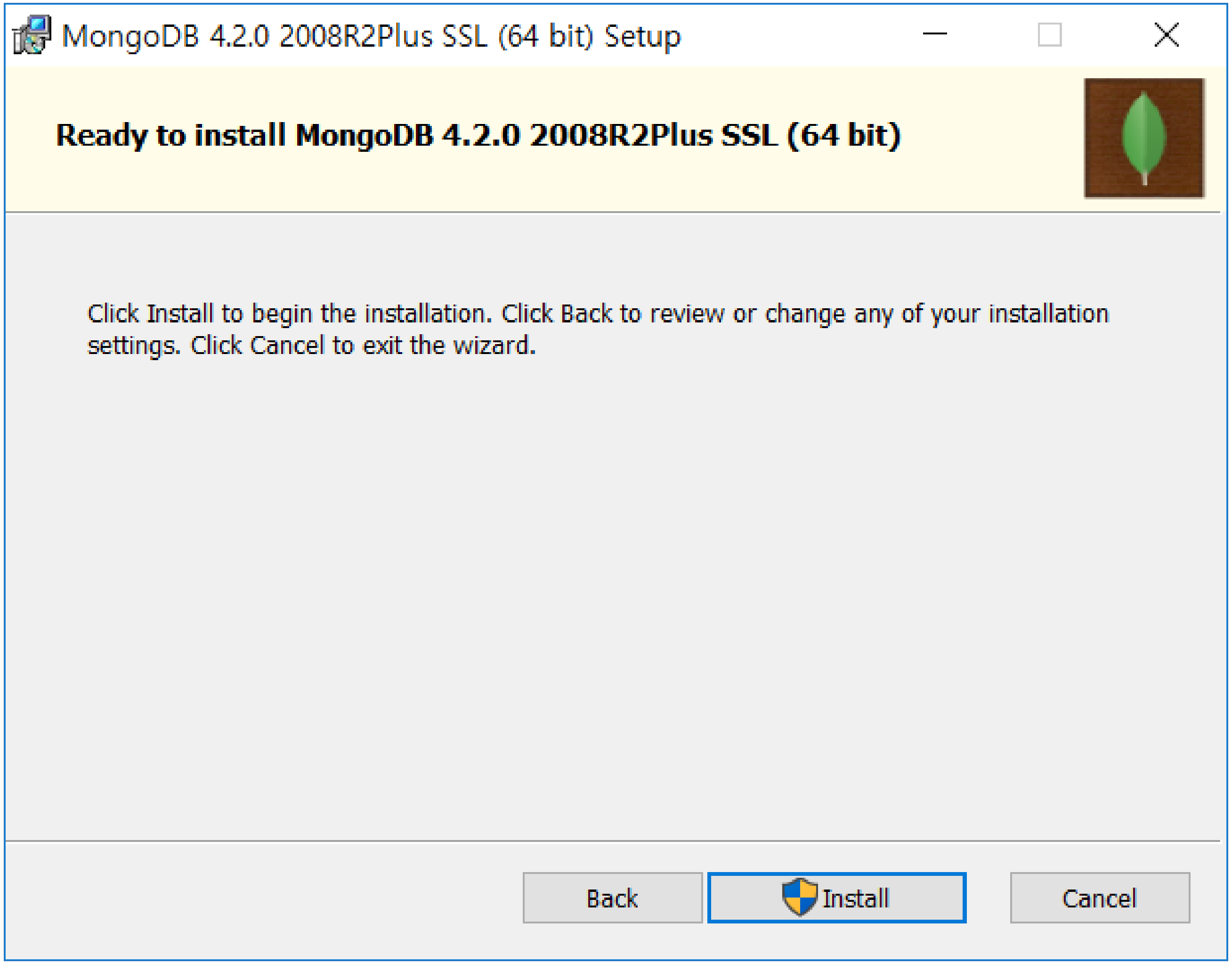
MongoDB Compass is the official graphical user interface for MongoDB.

By checking below this installer will automatically download and install the latest version of MongoDB Compass on this machine. You can learn more about MongoDB Compass here: https://www.mongodb.com/products/compass

☑ Install MongoDB Compass

■ MongoDB 설치

- Install



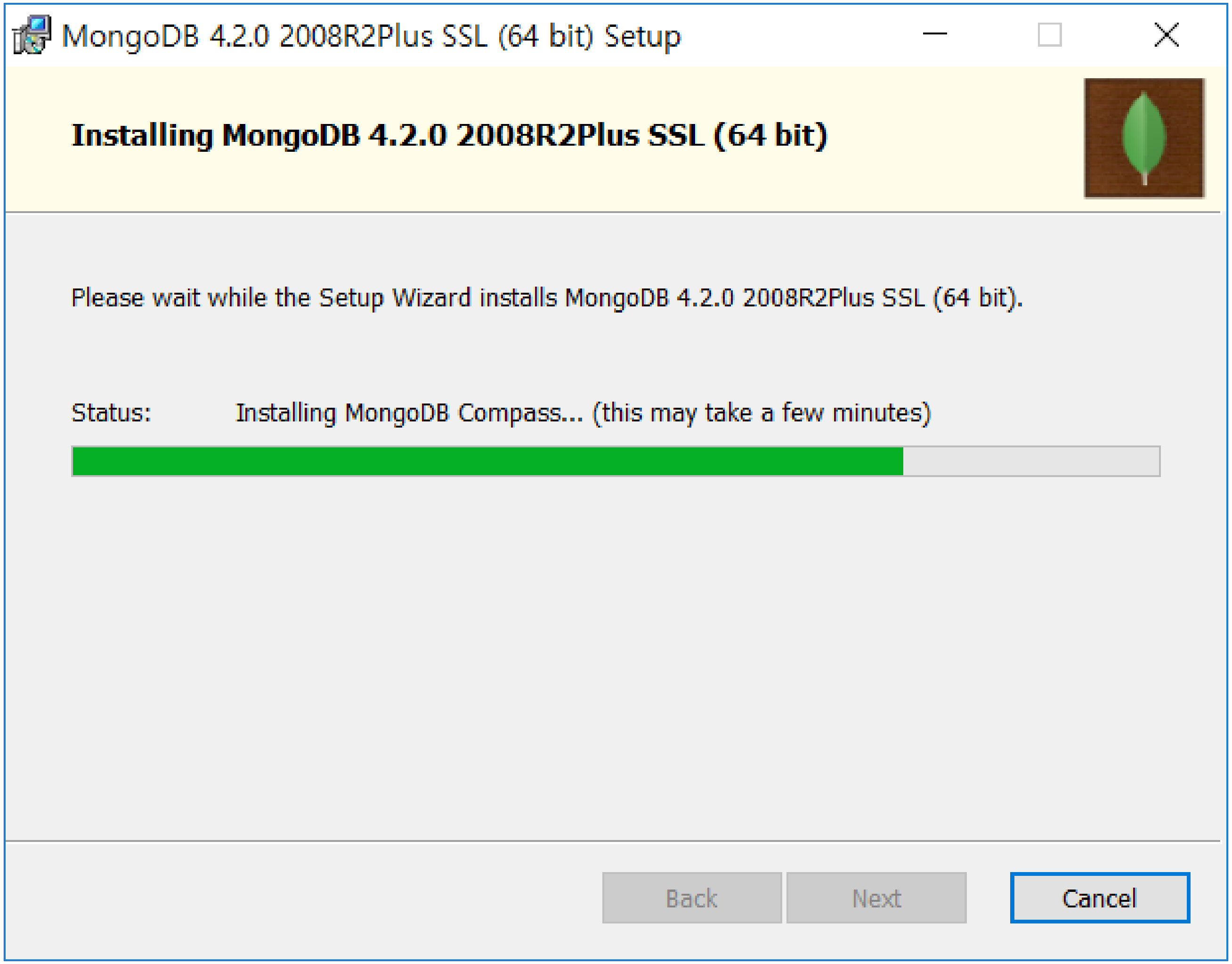
MongoDB 4.2.0 2008R2Plus SSL (64 bit) Setup

Ready to install MongoDB 4.2.0 2008R2Plus SSL (64 bit)

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

■ MongoDB 설치

- Install



MongoDB 4.2.0 2008R2Plus SSL (64 bit) Setup

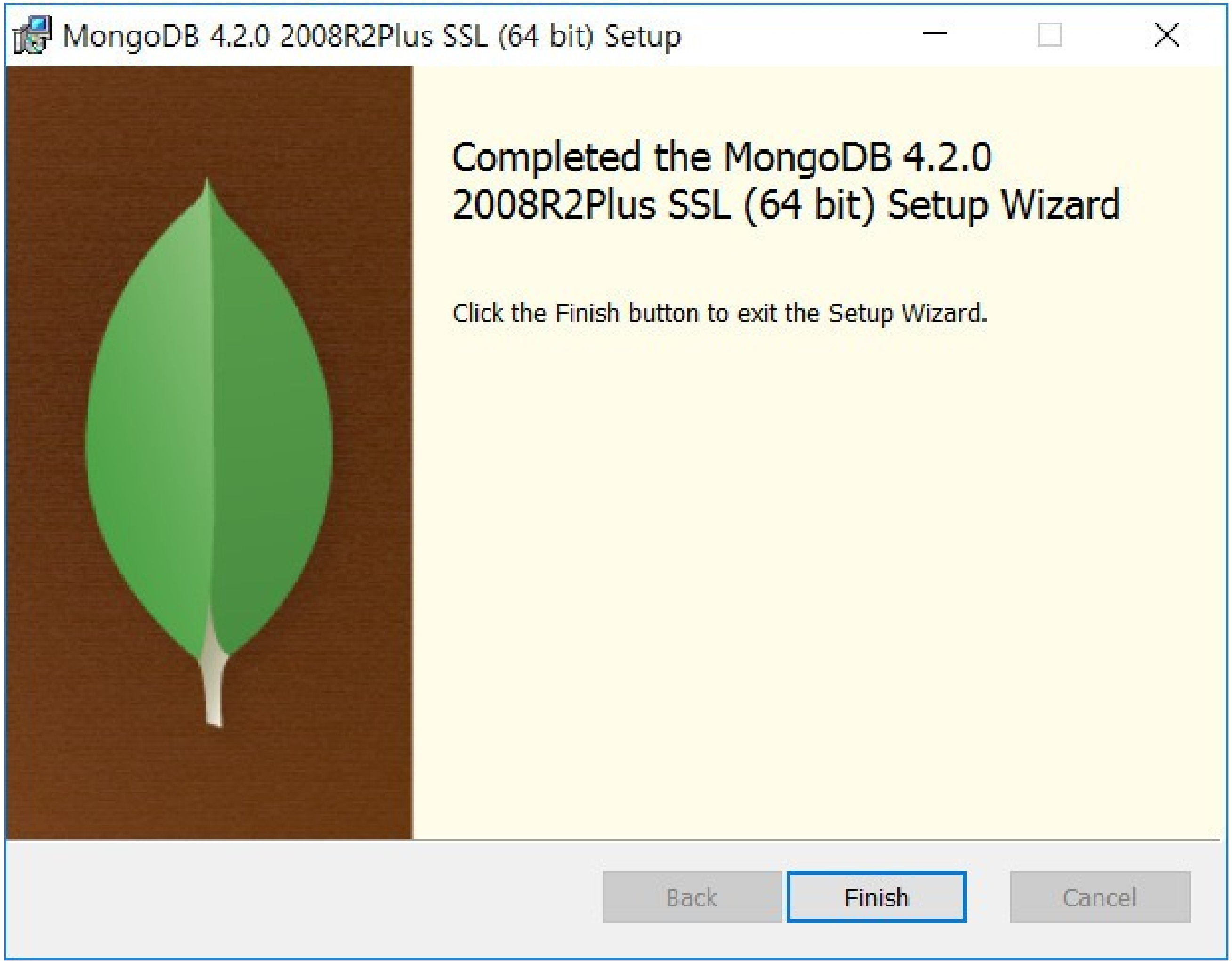
Installing MongoDB 4.2.0 2008R2Plus SSL (64 bit)

Please wait while the Setup Wizard installs MongoDB 4.2.0 2008R2Plus SSL (64 bit).

Status: Installing MongoDB Compass... (this may take a few minutes)

■ MongoDB 설치

- Install



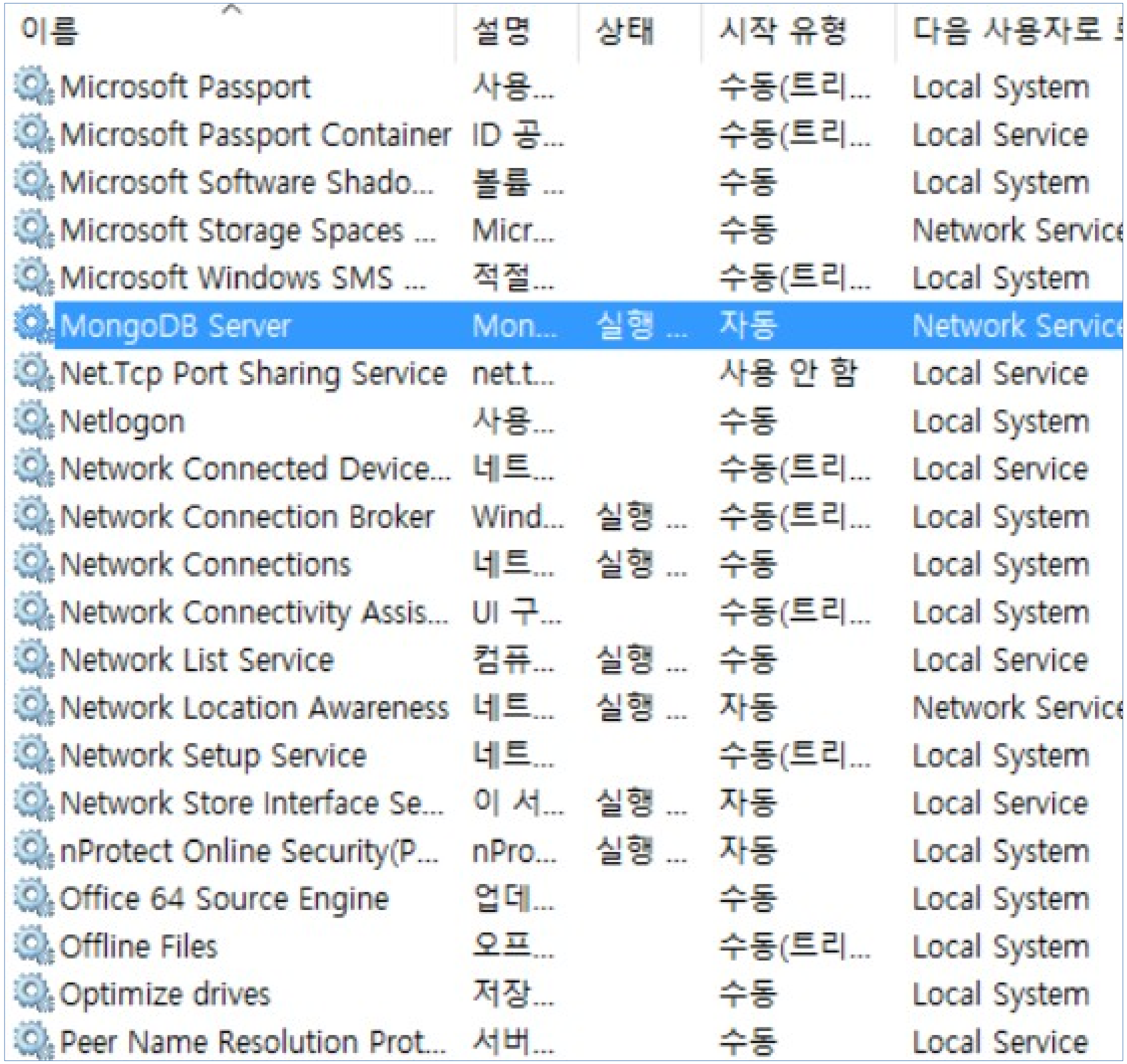
MongoDB 4.2.0 2008R2Plus SSL (64 bit) Setup

Completed the MongoDB 4.2.0 2008R2Plus SSL (64 bit) Setup Wizard

Click the Finish button to exit the Setup Wizard.

■ MongoDB 서비스(인스턴스)

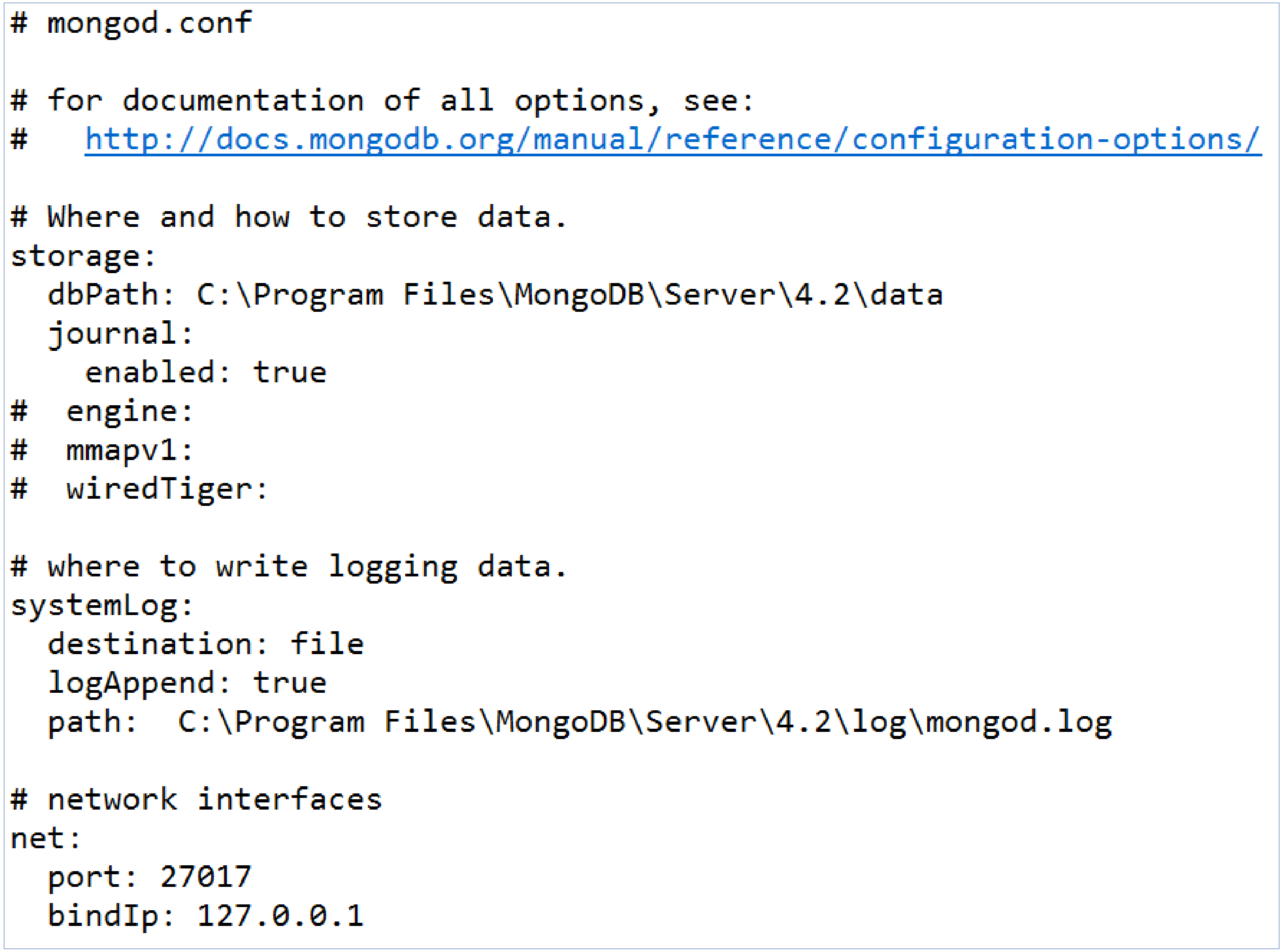
● 제어판 – 관리도구 – 서비스 (services.msc)



이름	설명	상태	시작 유형	다음 사용자로
Microsoft Passport	사용...		수동(트리...	Local System
Microsoft Passport Container	ID 공...		수동(트리...	Local Service
Microsoft Software Shado...	볼륨 ...		수동	Local System
Microsoft Storage Spaces ...	Micr...		수동	Network Service
Microsoft Windows SMS ...	적절...		수동(트리...	Local System
MongoDB Server	Mon...	실행 ...	자동	Network Service
Net.Tcp Port Sharing Service	net.t...		사용 안 함	Local Service
Netlogon	사용...		수동	Local System
Network Connected Device...	네트...		수동(트리...	Local Service
Network Connection Broker	Wind...	실행 ...	수동(트리...	Local System
Network Connections	네트...	실행 ...	수동	Local System
Network Connectivity Assis...	UI 구...		수동(트리...	Local System
Network List Service	컴퓨...	실행 ...	수동	Local Service
Network Location Awareness	네트...	실행 ...	자동	Network Service
Network Setup Service	네트...		수동(트리...	Local System
Network Store Interface Se...	이 서...	실행 ...	자동	Local Service
nProtect Online Security(P...	nPro...	실행 ...	자동	Local System
Office 64 Source Engine	업데...		수동	Local System
Offline Files	오프...		수동(트리...	Local System
Optimize drives	저장...		수동	Local System
Peer Name Resolution Prot...	서버...		수동	Local Service

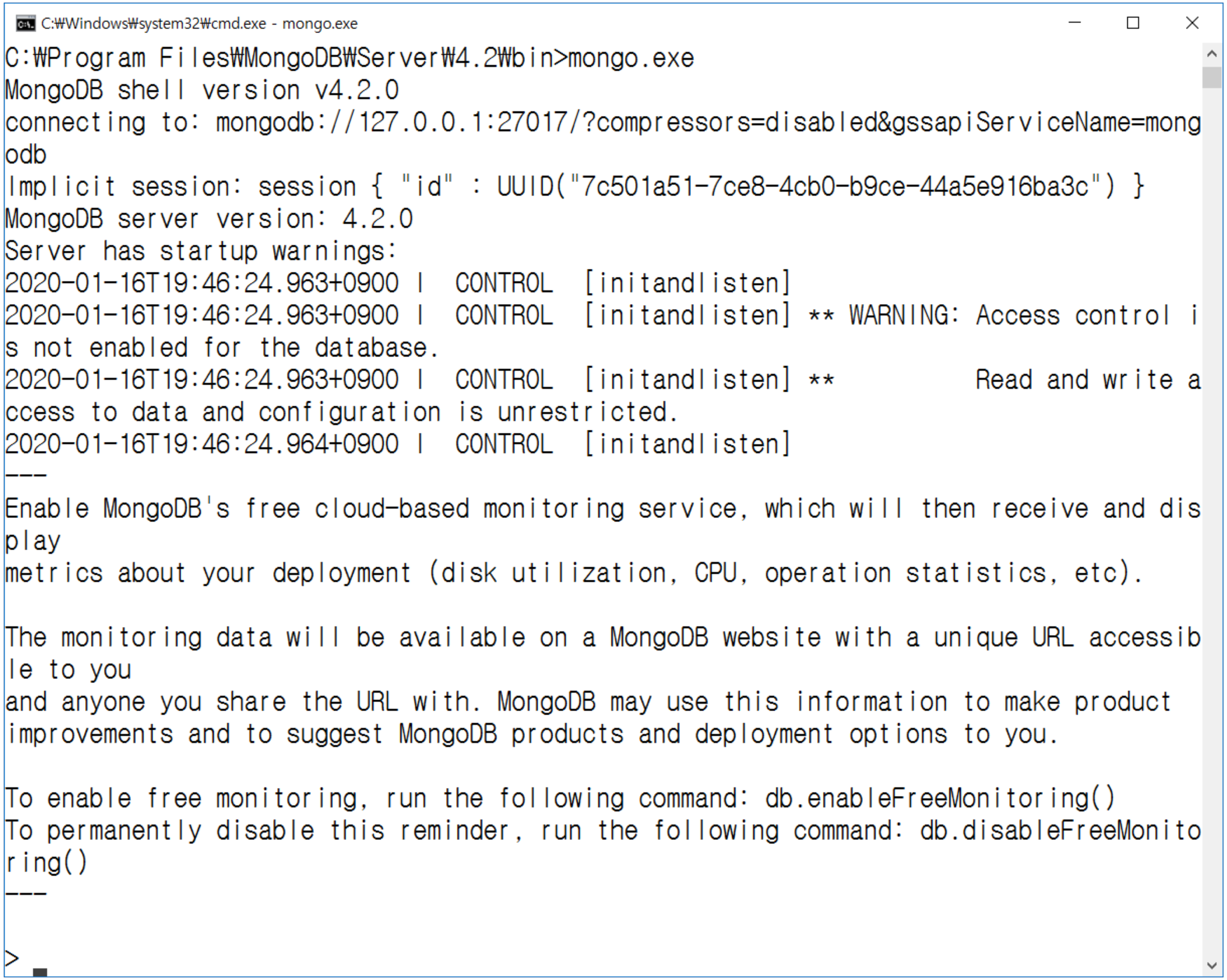
■ MongoDB 환경설정

● [MongoDB 설치경로]/bin/mongod.cfg



```
# mongod.conf

# for documentation of all options, see:
#   http://docs.mongodb.org/manual/reference/configuration-options/

# Where and how to store data.
storage:
  dbPath: C:\Program Files\MongoDB\Server\4.2\data
  journal:
    enabled: true
#  engine:
#  mmapv1:
#  wiredTiger:

# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path:  C:\Program Files\MongoDB\Server\4.2\log\mongod.log

# network interfaces
net:
  port: 27017
  bindIp: 127.0.0.1
```

- MongoDB 실행
  - [MongoDB 설치경로]/bin/mongo.exe



```
C:₩Program Files₩MongoDB₩Server₩4.2₩bin>mongo.exe
MongoDB shell version v4.2.0
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mong
odb
Implicit session: session { "id" : UUID("7c501a51-7ce8-4cb0-b9ce-44a5e916ba3c") }
MongoDB server version: 4.2.0
Server has startup warnings:
2020-01-16T19:46:24.963+0900 I  CONTROL  [initandlisten]
2020-01-16T19:46:24.963+0900 I  CONTROL  [initandlisten] ** WARNING: Access control i
s not enabled for the database.
2020-01-16T19:46:24.963+0900 I  CONTROL  [initandlisten] **          Read and write a
ccess to data and configuration is unrestricted.
2020-01-16T19:46:24.964+0900 I  CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and dis
play
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessib
le to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonito
ring()
---

>
```

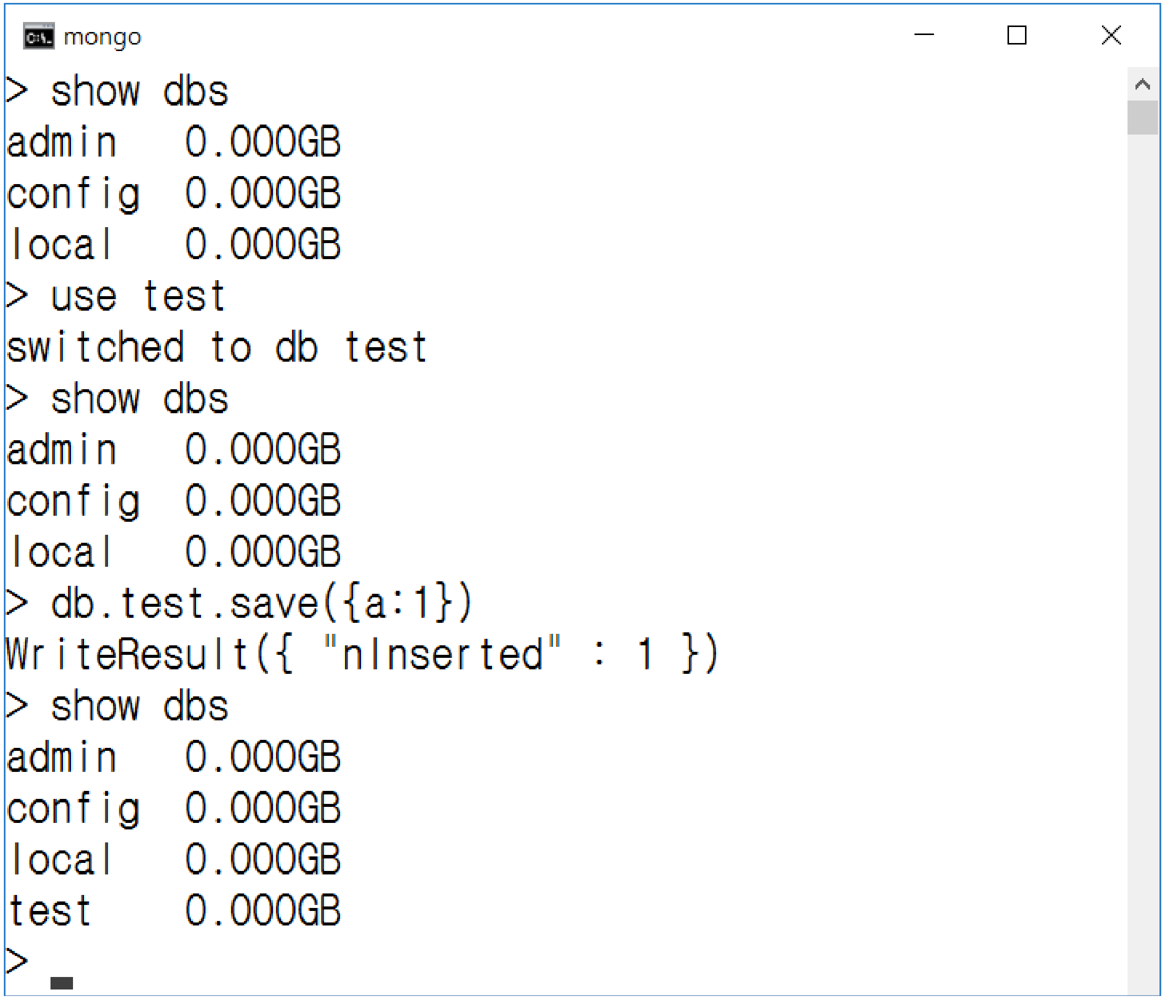
■ MongoDB

● RDBMS와 MongoDB의 용어 비교

SQL 사용 용어	MongoDB 사용 용어
데이터베이스(database)	데이터베이스(database)
테이블(table)	컬렉션(collection)
행(row)	문서(document) 또는 BSON 문서
열(column)	필드(field)
색인(index)	색인(index)
테이블 조인(table joins)	임베디드 문서 & 링킹(linking)
기본(주) 키(primary key, 유일한 고유 칼럼)	기본(주) 키(primary key, _id 필드 자동 생성)
집합(aggregation, 예: group by)	집합(aggregation) 프레임워크

■ MongoDB

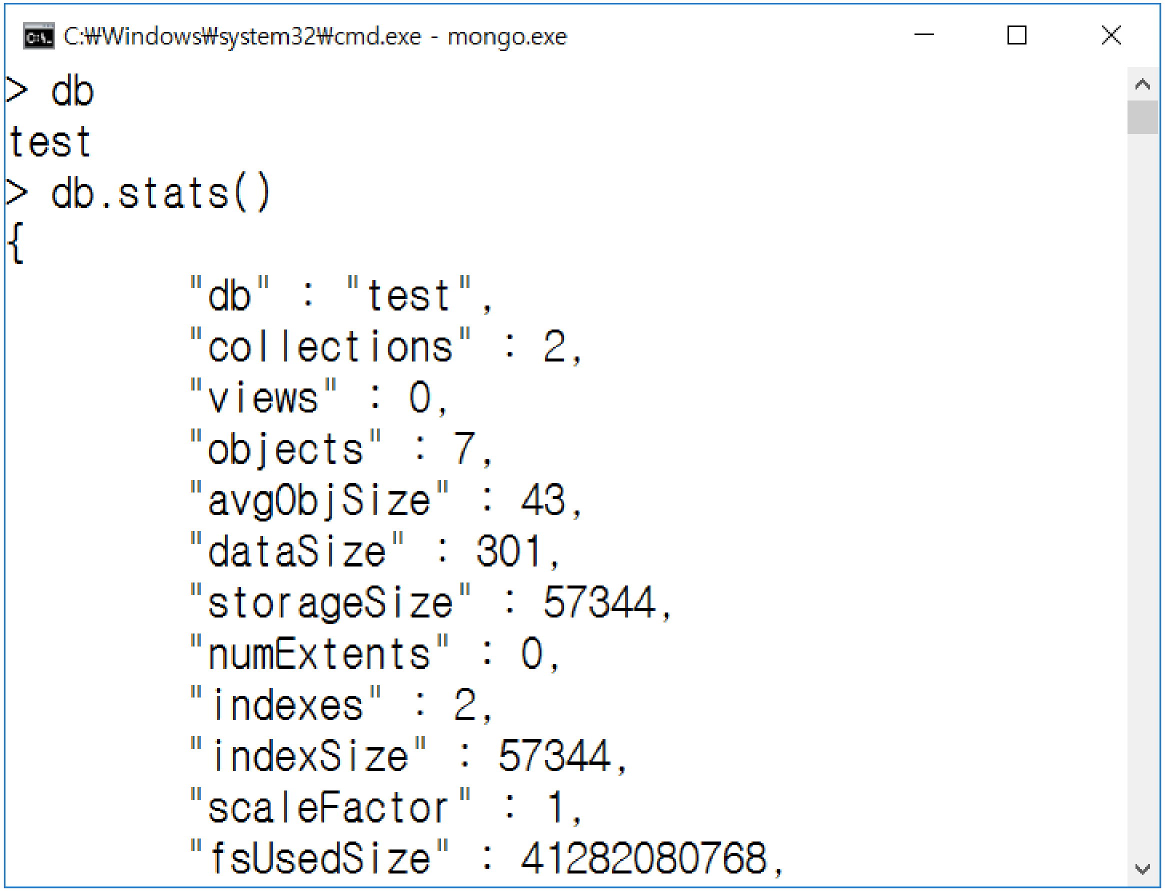
● 데이터베이스 조회 / 사용



```
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
> use test
switched to db test
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
> db.test.save({a:1})
WriteResult({ "nInserted" : 1 })
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
test    0.000GB
>
```

■ MongoDB

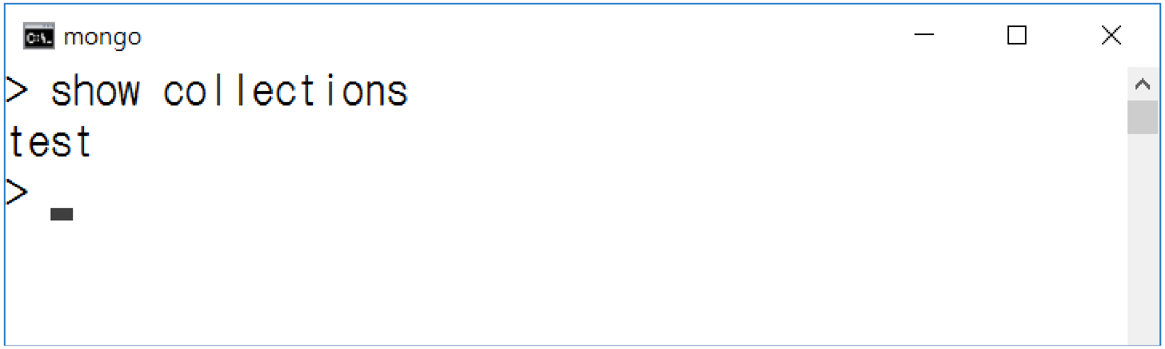
● 데이터베이스 확인



```
> db
test
> db.stats()
{
        "db" : "test",
        "collections" : 2,
        "views" : 0,
        "objects" : 7,
        "avgObjSize" : 43,
        "dataSize" : 301,
        "storageSize" : 57344,
        "numExtents" : 0,
        "indexes" : 2,
        "indexSize" : 57344,
        "scaleFactor" : 1,
        "fsUsedSize" : 41282080768,
```

■ MongoDB

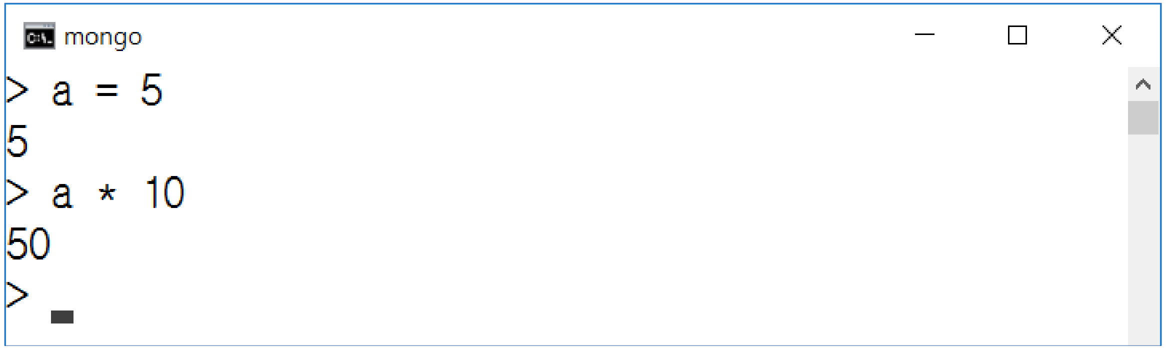
● 컬렉션 조회



```
> show collections
test
>
```

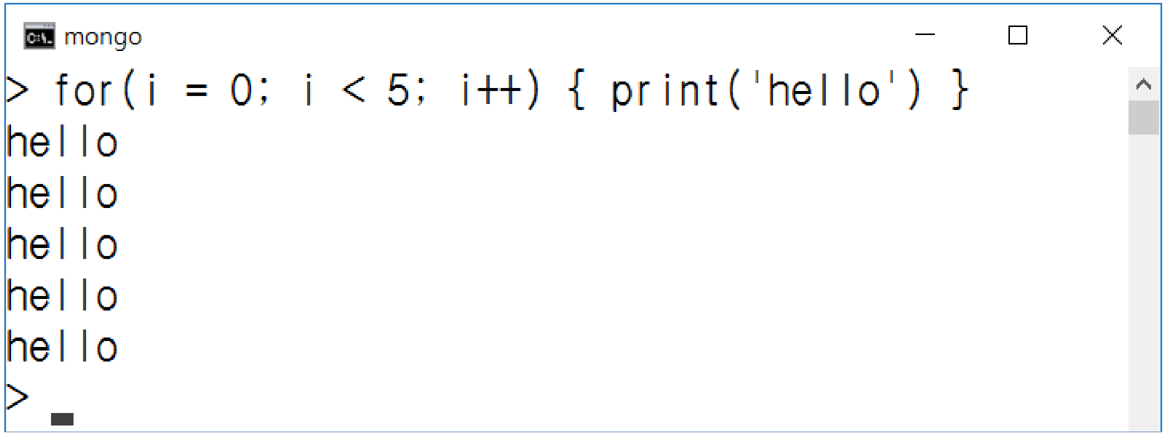
■ 자바스크립트 문법

● 변수 생성 및 연산



```
> a = 5
5
> a * 10
50
>
```

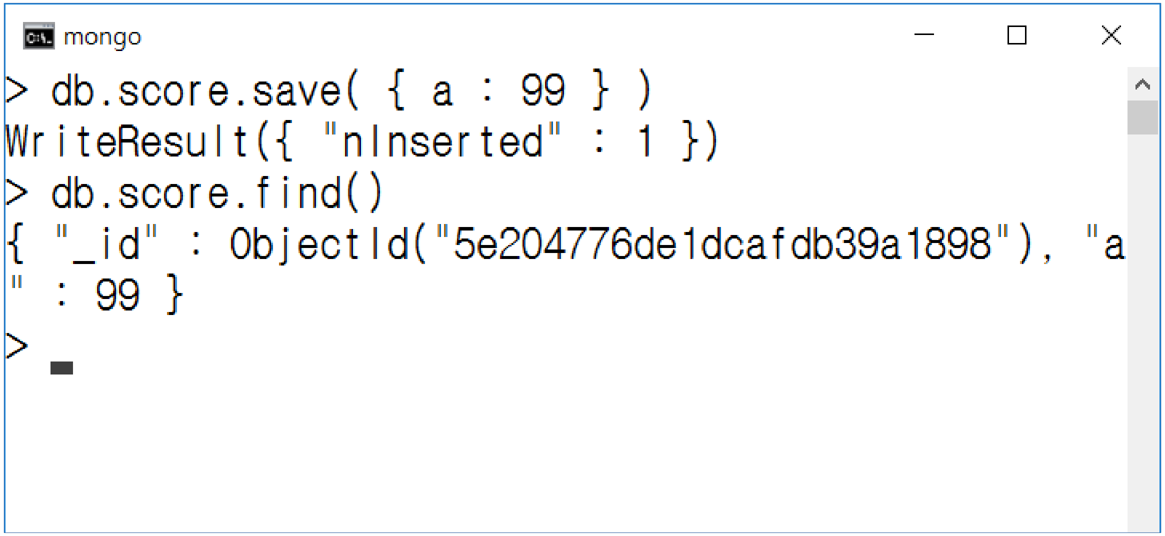
● 반복문



```
> for(i = 0; i < 5; i++) { print('hello') }
hello
hello
hello
hello
hello
>
```

■ MongoDB CRUD

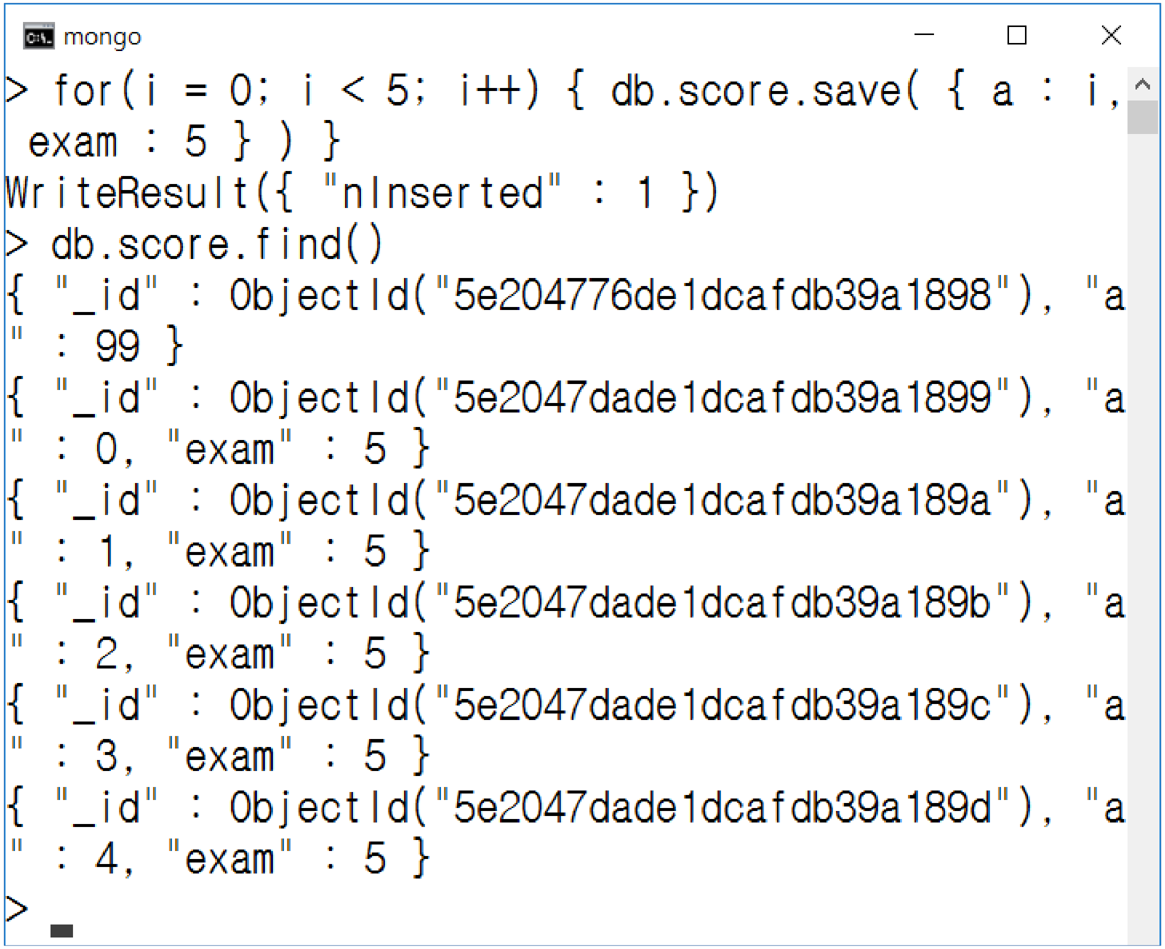
● 데이터 저장



```
> db.score.save( { a : 99 } )
WriteResult({ "nInserted" : 1 })
> db.score.find()
{ "_id" : ObjectId("5e204776de1dcafdb39a1898"), "a
" : 99 }
> 
```

■ MongoDB CRUD

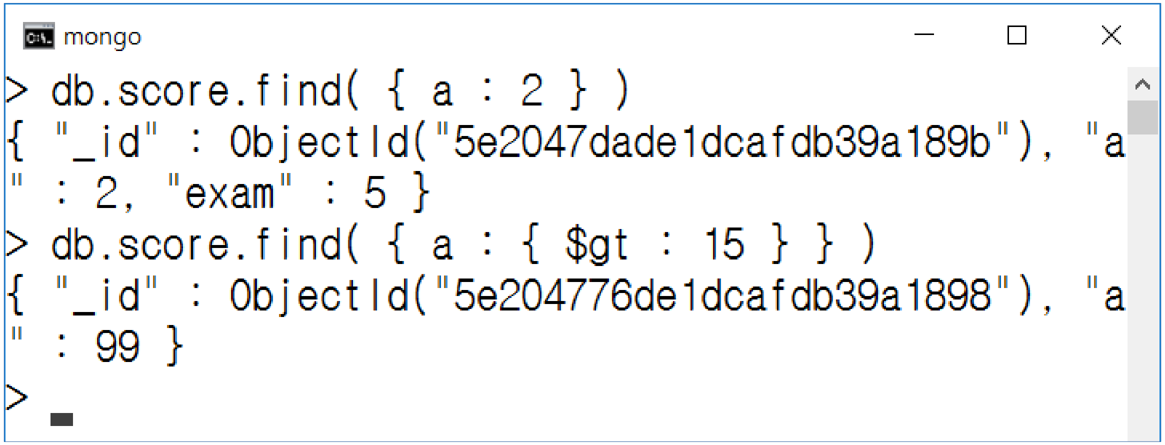
● 데이터 저장



```
> for(i = 0; i < 5; i++) { db.score.save( { a : i,
 exam : 5 } ) }
WriteResult({ "nInserted" : 1 })
> db.score.find()
{ "_id" : ObjectId("5e204776de1dcafdb39a1898"), "a
" : 99 }
{ "_id" : ObjectId("5e2047dade1dcafdb39a1899"), "a
" : 0, "exam" : 5 }
{ "_id" : ObjectId("5e2047dade1dcafdb39a189a"), "a
" : 1, "exam" : 5 }
{ "_id" : ObjectId("5e2047dade1dcafdb39a189b"), "a
" : 2, "exam" : 5 }
{ "_id" : ObjectId("5e2047dade1dcafdb39a189c"), "a
" : 3, "exam" : 5 }
{ "_id" : ObjectId("5e2047dade1dcafdb39a189d"), "a
" : 4, "exam" : 5 }
> _
```

■ MongoDB CRUD

● 데이터 조회 / where

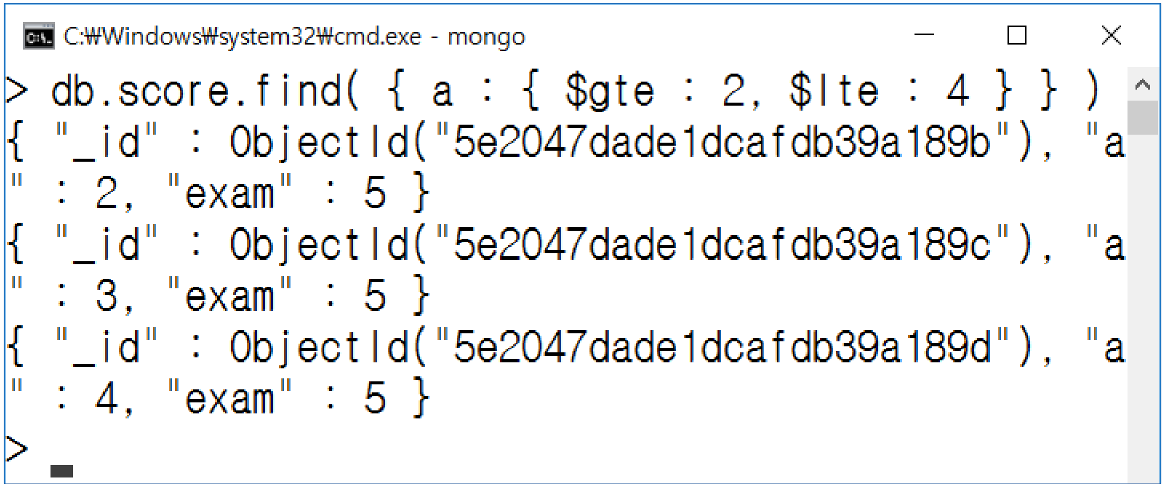


```
> db.score.find( { a : 2 } )
{ "_id" : ObjectId("5e2047dade1dcafdb39a189b"), "a
" : 2, "exam" : 5 }
> db.score.find( { a : { $gt : 15 } } )
{ "_id" : ObjectId("5e204776de1dcafdb39a1898"), "a
" : 99 }
> 
```

```
db.score.find( { a : 2 } )

db.score.find( { a : { $gt : 15 } } )
```

■ MongoDB CRUD

● 데이터 조회 / AND



```
> db.score.find( { a : { $gte : 2, $lte : 4 } } )
{ "_id" : ObjectId("5e2047dade1dcafdb39a189b"), "a
" : 2, "exam" : 5 }
{ "_id" : ObjectId("5e2047dade1dcafdb39a189c"), "a
" : 3, "exam" : 5 }
{ "_id" : ObjectId("5e2047dade1dcafdb39a189d"), "a
" : 4, "exam" : 5 }
>
```

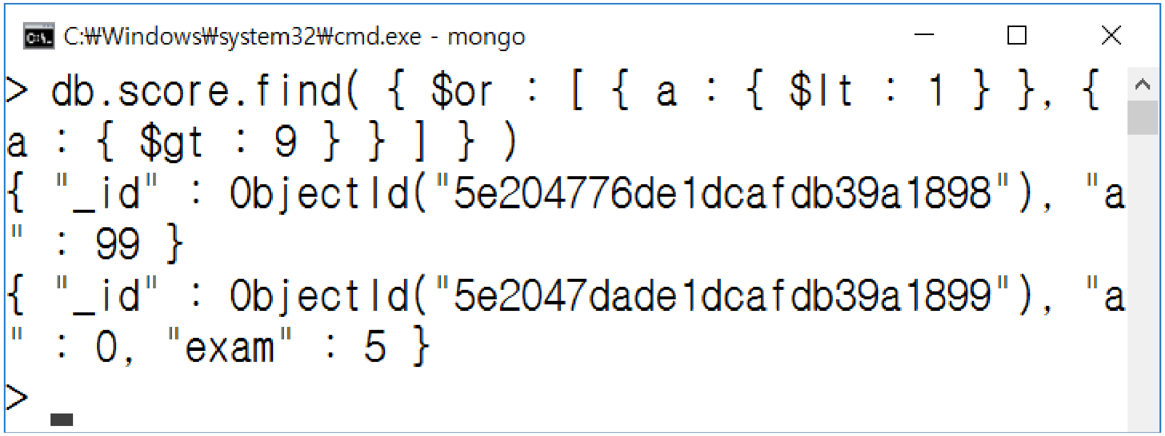
```
db.score.find( { a : { $gte : 2, $lte : 4 } } )
```

※ 비교 연산자

$gt → >, $lt → <, $gte → >=, $lte → <=, $ne → !=

■ MongoDB CRUD

● 데이터 조회 / OR



```
> db.score.find( { $or : [ { a : { $lt : 1 } }, {
a : { $gt : 9 } } ] } )
{ "_id" : ObjectId("5e204776de1dcafdb39a1898"), "a
" : 99 }
{ "_id" : ObjectId("5e2047dade1dcafdb39a1899"), "a
" : 0, "exam" : 5 }
> _
```

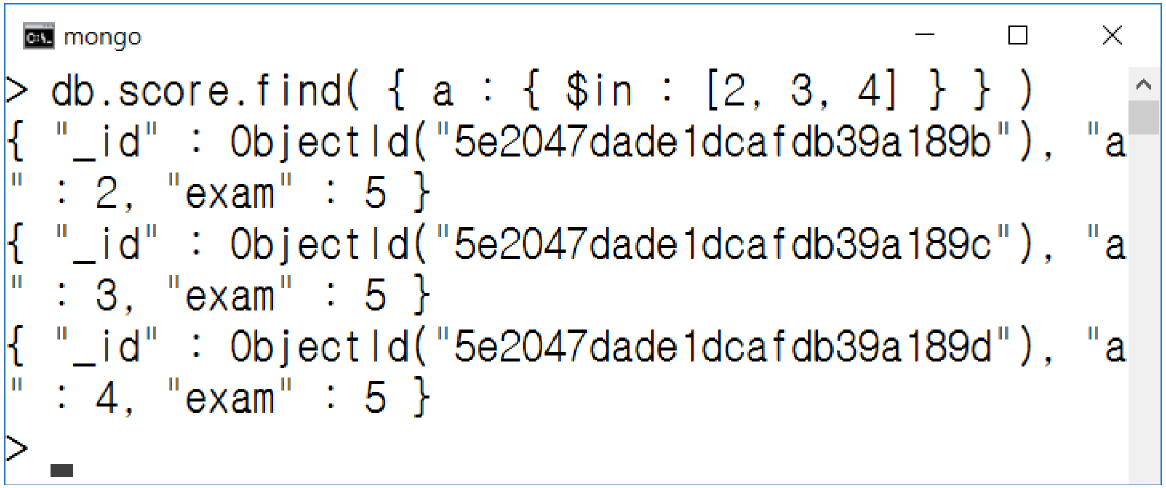
```
db.score.find( { $or : [ { a : { $lt : 1 } }, { a : { $gt : 9 } } ] } )
```

※ 비교 연산자

$gt → >, $lt → <, $gte → >=, $lte → <=, $ne → !=

■ MongoDB CRUD

● 데이터 조회 / IN – is in array

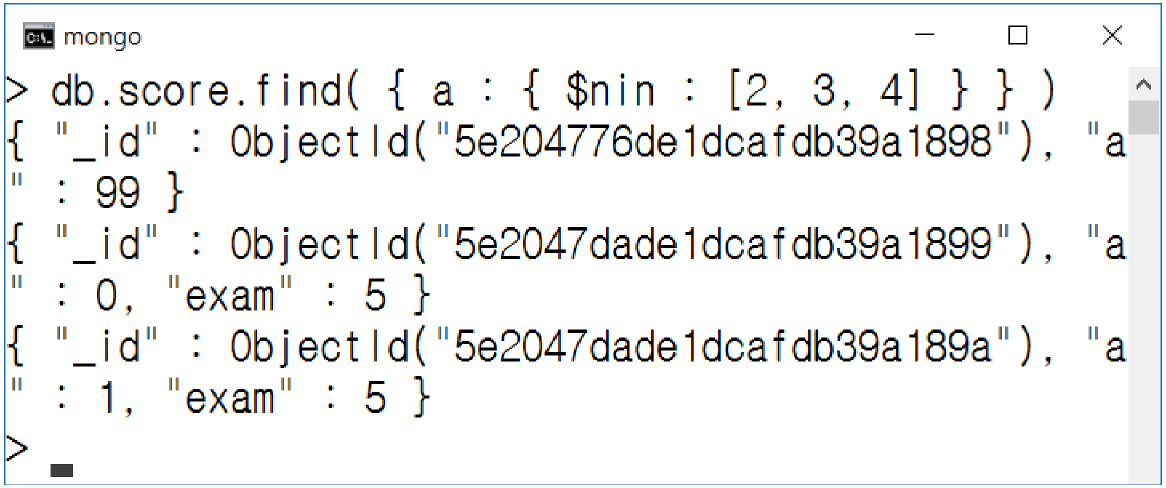


```
> db.score.find( { a : { $in : [2, 3, 4] } } )
{ "_id" : ObjectId("5e2047dade1dcafdb39a189b"), "a
" : 2, "exam" : 5 }
{ "_id" : ObjectId("5e2047dade1dcafdb39a189c"), "a
" : 3, "exam" : 5 }
{ "_id" : ObjectId("5e2047dade1dcafdb39a189d"), "a
" : 4, "exam" : 5 }
> _
```

```
db.score.find( { a : { $in : [2, 3, 4] } } )
```

■ MongoDB CRUD

● 데이터 조회 / NOT IN – is not in array

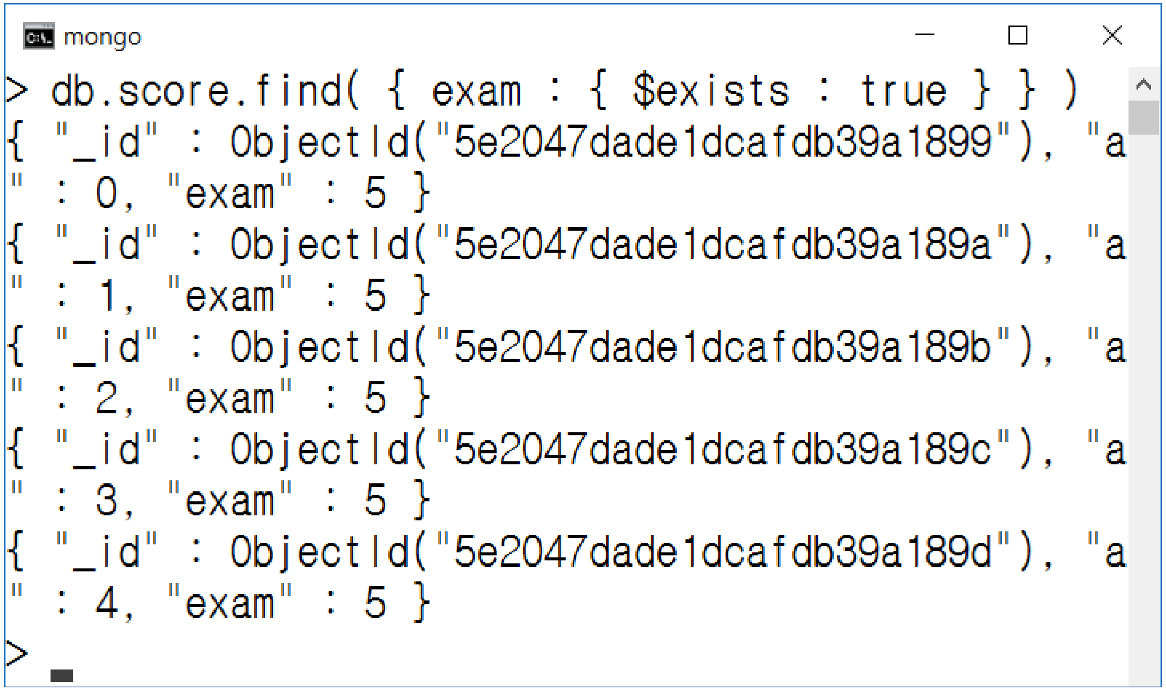


```
> db.score.find( { a : { $nin : [2, 3, 4] } } )
{ "_id" : ObjectId("5e204776de1dcafdb39a1898"), "a
" : 99 }
{ "_id" : ObjectId("5e2047dade1dcafdb39a1899"), "a
" : 0, "exam" : 5 }
{ "_id" : ObjectId("5e2047dade1dcafdb39a189a"), "a
" : 1, "exam" : 5 }
>
```

```
db.score.find( { a : { $nin : [2, 3, 4] } } )
```

■ MongoDB CRUD

● 데이터 조회 / 필드 존재 유무 확인

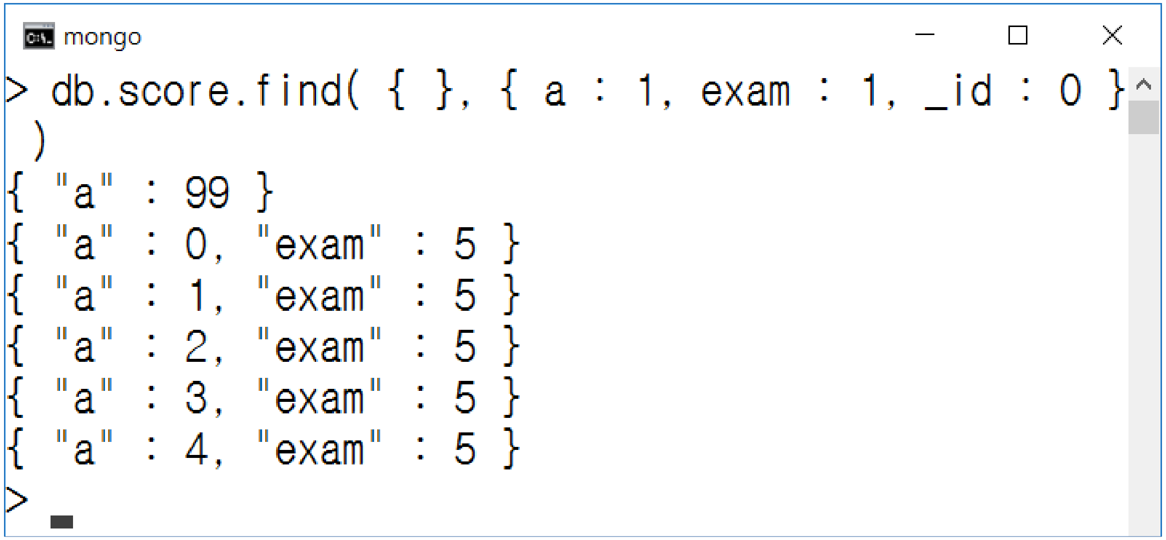


```
> db.score.find( { exam : { $exists : true } } )
{ "_id" : ObjectId("5e2047dade1dcafdb39a1899"), "a
" : 0, "exam" : 5 }
{ "_id" : ObjectId("5e2047dade1dcafdb39a189a"), "a
" : 1, "exam" : 5 }
{ "_id" : ObjectId("5e2047dade1dcafdb39a189b"), "a
" : 2, "exam" : 5 }
{ "_id" : ObjectId("5e2047dade1dcafdb39a189c"), "a
" : 3, "exam" : 5 }
{ "_id" : ObjectId("5e2047dade1dcafdb39a189d"), "a
" : 4, "exam" : 5 }
> _
```

```
db.score.find( { exam : { $exists : true } } )
```

■ MongoDB CRUD

● 데이터 조회 / 조회할 필드 선택 (0 : 해제, 1 : 선택)

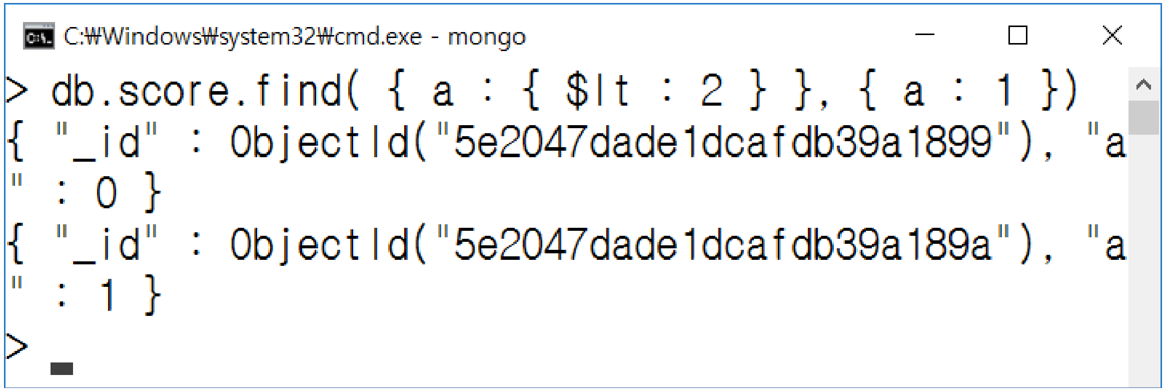


```
> db.score.find( { }, { a : 1, exam : 1, _id : 0 }
 )
{ "a" : 99 }
{ "a" : 0, "exam" : 5 }
{ "a" : 1, "exam" : 5 }
{ "a" : 2, "exam" : 5 }
{ "a" : 3, "exam" : 5 }
{ "a" : 4, "exam" : 5 }
>
```

```
db.score.find( { }, { a : 1, exam : 1, _id : 0 }
```

■ MongoDB CRUD

● 데이터 조회 / 조회할 필드 선택 (a가 2보다 적은 문서의 a 필드 조회)



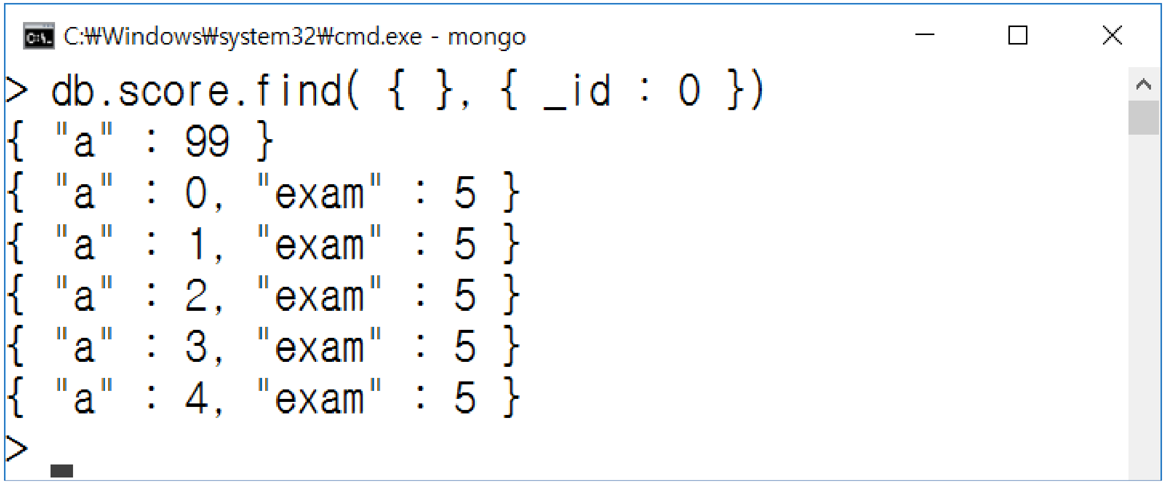
```
> db.score.find( { a : { $lt : 2 } }, { a : 1 })
{ "_id" : ObjectId("5e2047dade1dcafdb39a1899"), "a
" : 0 }
{ "_id" : ObjectId("5e2047dade1dcafdb39a189a"), "a
" : 1 }
> 
```

```
db.score.find( { }, { a : 1, exam : 1, _id : 0 }
```

- _id 필드는 명시하지 않을 경우 기본 선택

■ MongoDB CRUD

● 데이터 조회 / 조회할 필드 선택 (모든 문서의 모든 필드 조회, _id 필드 해제)

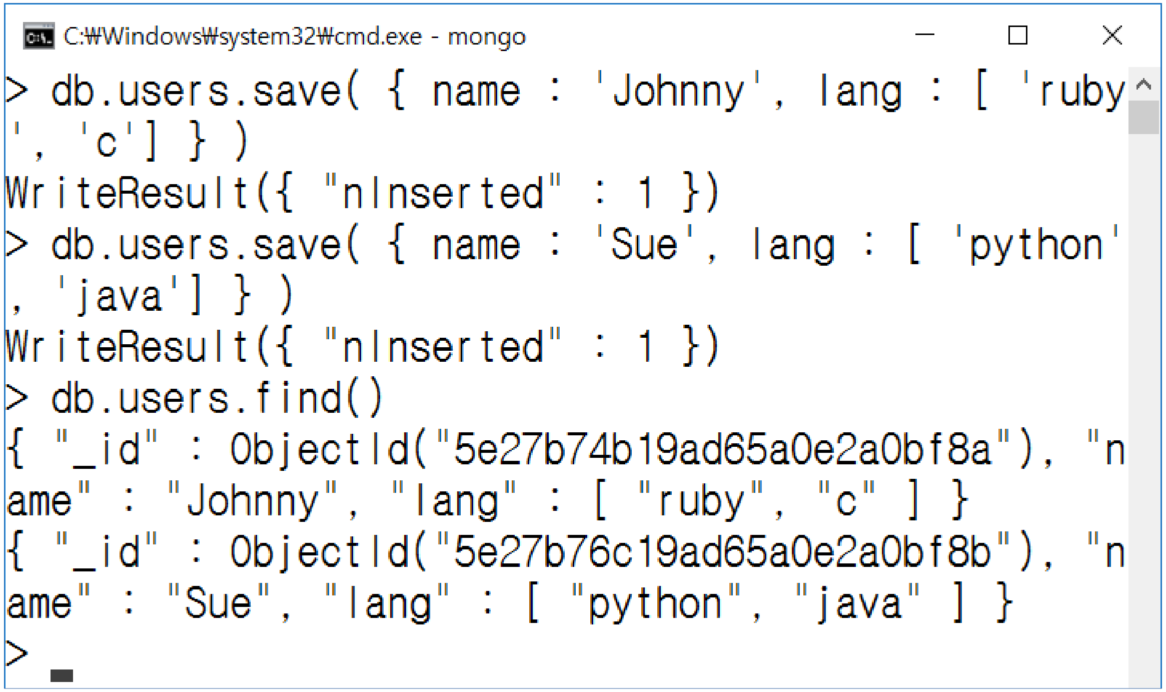


```
> db.score.find( { }, { _id : 0 })
{ "a" : 99 }
{ "a" : 0, "exam" : 5 }
{ "a" : 1, "exam" : 5 }
{ "a" : 2, "exam" : 5 }
{ "a" : 3, "exam" : 5 }
{ "a" : 4, "exam" : 5 }
>
```

```
db.score.find( { }, { _id : 0 })
```

■ MongoDB CRUD

● 데이터 수정 (수정 연습용 데이터 입력)

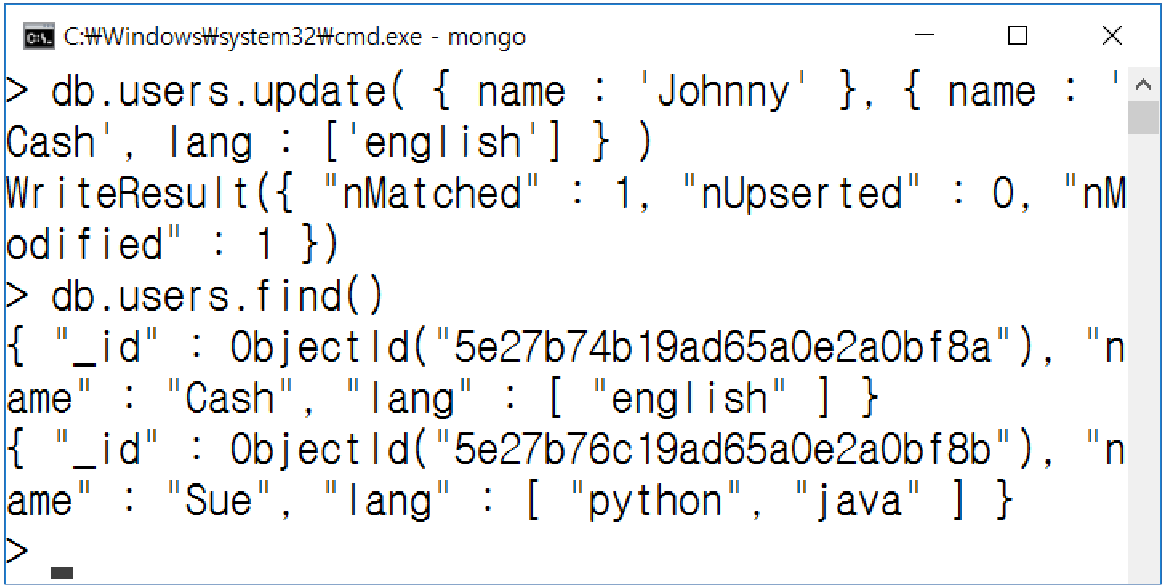


```
> db.users.save( { name : 'Johnny', lang : [ 'ruby
', 'c'] } )
WriteResult({ "nInserted" : 1 })
> db.users.save( { name : 'Sue', lang : [ 'python'
, 'java'] } )
WriteResult({ "nInserted" : 1 })
> db.users.find()
{ "_id" : ObjectId("5e27b74b19ad65a0e2a0bf8a"), "n
ame" : "Johnny", "lang" : [ "ruby", "c" ] }
{ "_id" : ObjectId("5e27b76c19ad65a0e2a0bf8b"), "n
ame" : "Sue", "lang" : [ "python", "java" ] }
> _
```

```
db.users.save( { name : 'Johnny', lang : [ 'ruby', 'c' ] } )

db.users.save( { name : 'Sue', lang : [ 'python', 'java' ] } )
```

■ MongoDB CRUD

● 데이터 수정 (name이 Johnny인 문서의 전체 내용 변경)

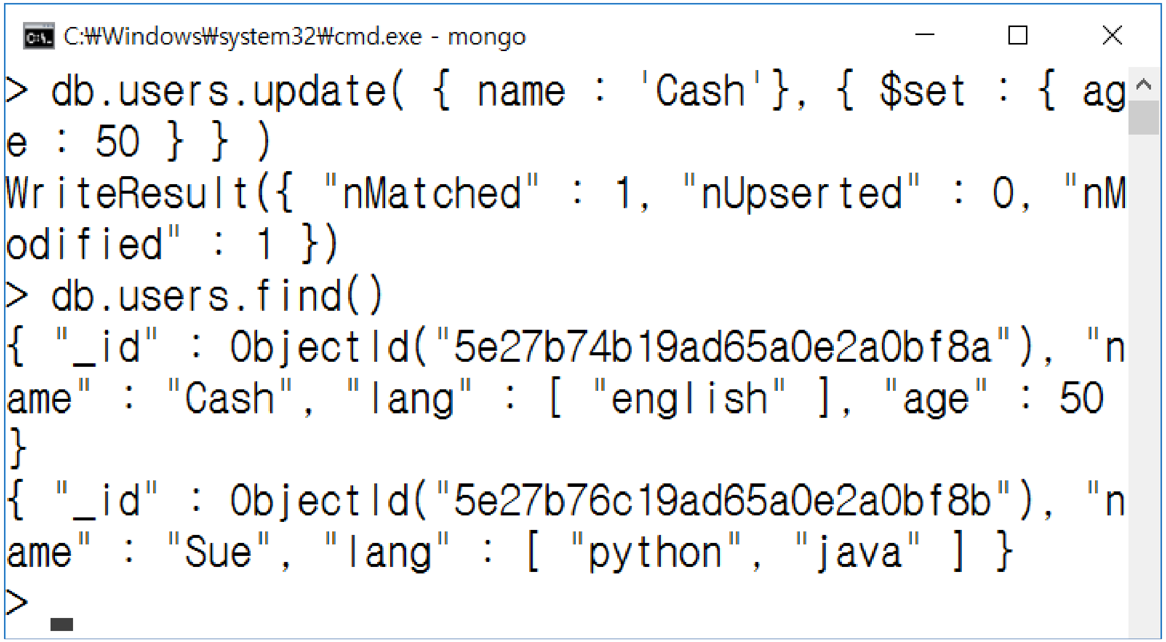


```
> db.users.update( { name : 'Johnny' }, { name : '
Cash', lang : ['english'] } )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nM
odified" : 1 })
> db.users.find()
{ "_id" : ObjectId("5e27b74b19ad65a0e2a0bf8a"), "n
ame" : "Cash", "lang" : [ "english" ] }
{ "_id" : ObjectId("5e27b76c19ad65a0e2a0bf8b"), "n
ame" : "Sue", "lang" : [ "python", "java" ] }
> _
```

```
db.users.update( { name : 'Johnny' }, { name : 'Cash', lang : ['english'] } )
```

■ MongoDB CRUD

● 데이터 수정 (name이 Cash인 문서에 필드 추가 또는 수정)

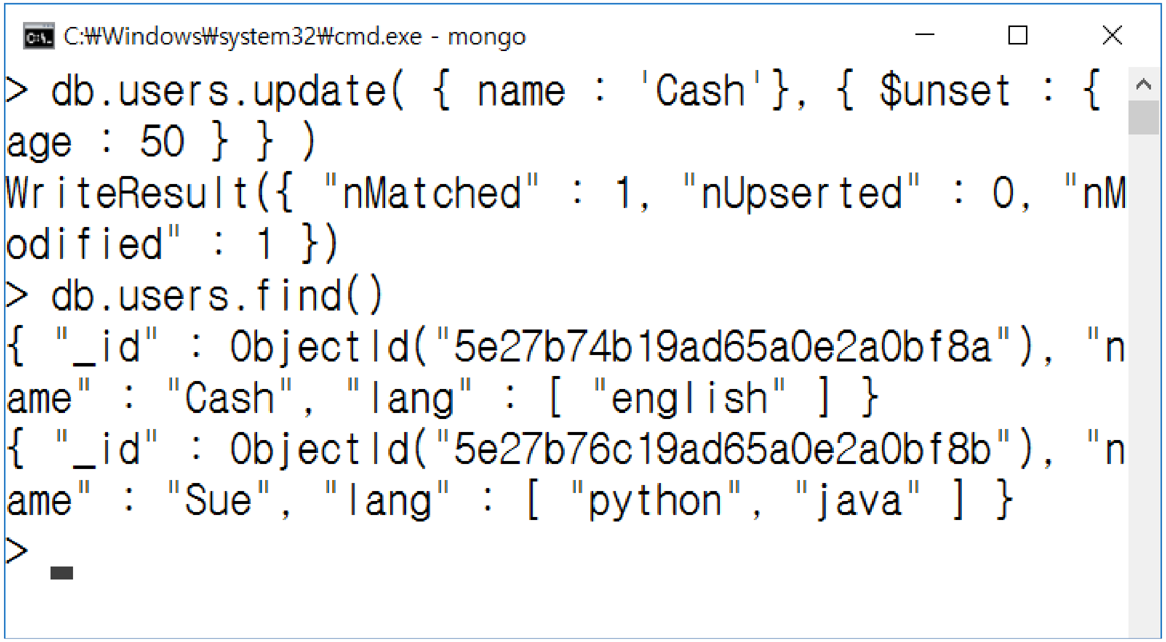


```
> db.users.update( { name : 'Cash'}, { $set : { ag
e : 50 } } )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nM
odified" : 1 })
> db.users.find()
{ "_id" : ObjectId("5e27b74b19ad65a0e2a0bf8a"), "n
ame" : "Cash", "lang" : [ "english" ], "age" : 50
}
{ "_id" : ObjectId("5e27b76c19ad65a0e2a0bf8b"), "n
ame" : "Sue", "lang" : [ "python", "java" ] }
>
```

```
db.users.update( { name : 'Cash'}, { $set : { age : 50 } } )
```

■ MongoDB CRUD

● 데이터 수정 (name이 Cash인 문서에 필드 삭제)



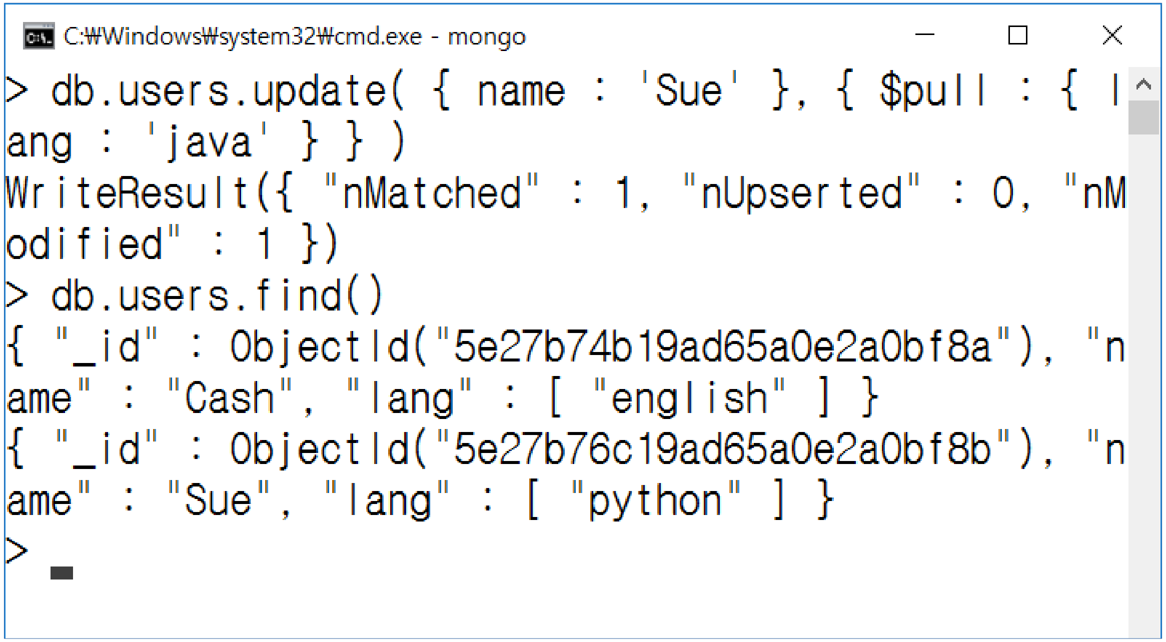
```
> db.users.update( { name : 'Cash'}, { $unset : {
age : 50 } } )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nM
odified" : 1 })
> db.users.find()
{ "_id" : ObjectId("5e27b74b19ad65a0e2a0bf8a"), "n
ame" : "Cash", "lang" : [ "english" ] }
{ "_id" : ObjectId("5e27b76c19ad65a0e2a0bf8b"), "n
ame" : "Sue", "lang" : [ "python", "java" ] }
>
```

```
db.users.update( { name : 'Cash'}, { $unset : { age : 50 } } )
```

- age 필드의 정확한 값이 아니라도 상관없음 ex) age : 0

■ MongoDB CRUD

● 데이터 수정 (리스트 요소 삭제)

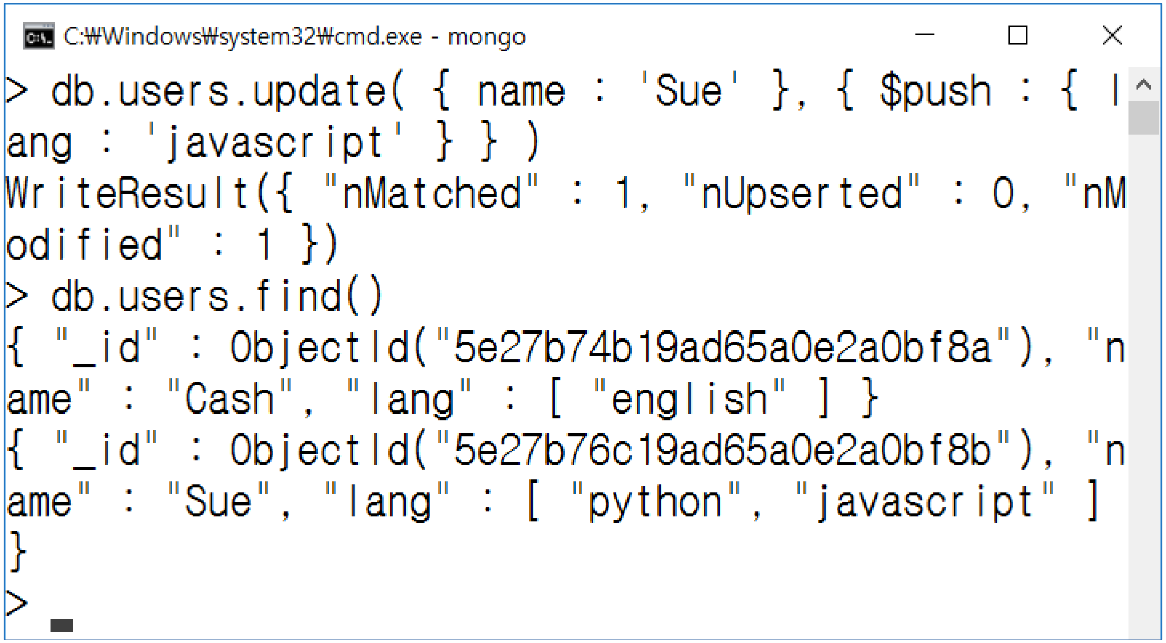


```
> db.users.update( { name : 'Sue' }, { $pull : { lang : 'java' } } )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.users.find()
{ "_id" : ObjectId("5e27b74b19ad65a0e2a0bf8a"), "name" : "Cash", "lang" : [ "english" ] }
{ "_id" : ObjectId("5e27b76c19ad65a0e2a0bf8b"), "name" : "Sue", "lang" : [ "python" ] }
>
```

```
db.users.update( { name : 'Sue' }, { $pull : { lang : 'java' } } )
```

■ MongoDB CRUD

● 데이터 수정 (리스트 요소 추가)

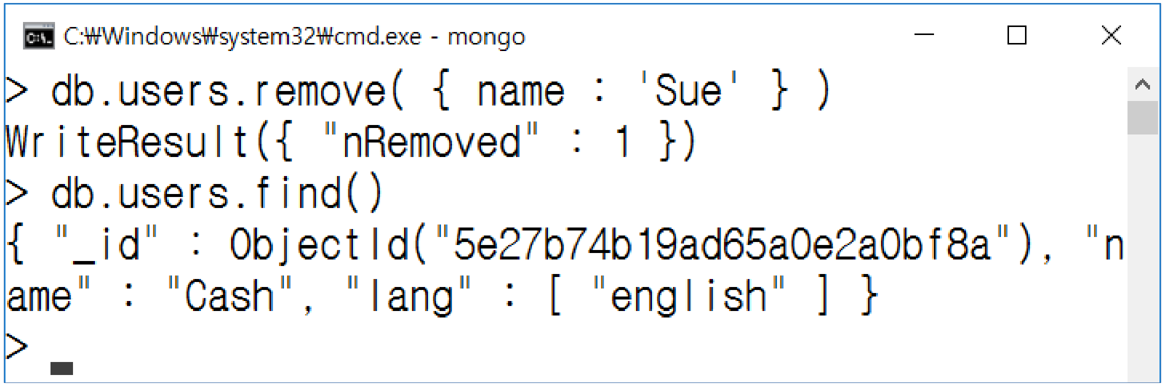


```
> db.users.update( { name : 'Sue' }, { $push : { l
ang : 'javascript' } } )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nM
odified" : 1 })
> db.users.find()
{ "_id" : ObjectId("5e27b74b19ad65a0e2a0bf8a"), "n
ame" : "Cash", "lang" : [ "english" ] }
{ "_id" : ObjectId("5e27b76c19ad65a0e2a0bf8b"), "n
ame" : "Sue", "lang" : [ "python", "javascript" ]
}
>
```

```
db.users.update( { name : 'Sue' }, { $push : { lang : 'javascript' } } )
```

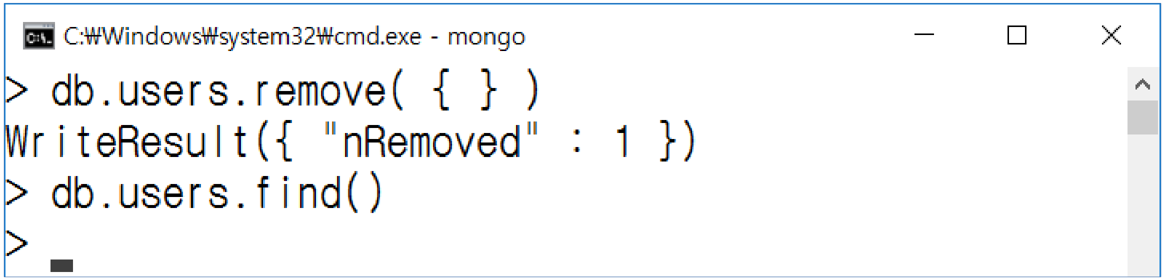
■ MongoDB CRUD

● 데이터 삭제



```
> db.users.remove( { name : 'Sue' } )
WriteResult({ "nRemoved" : 1 })
> db.users.find()
{ "_id" : ObjectId("5e27b74b19ad65a0e2a0bf8a"), "n
ame" : "Cash", "lang" : [ "english" ] }
> _
```

```
db.users.remove( { name : 'Sue' } )
```

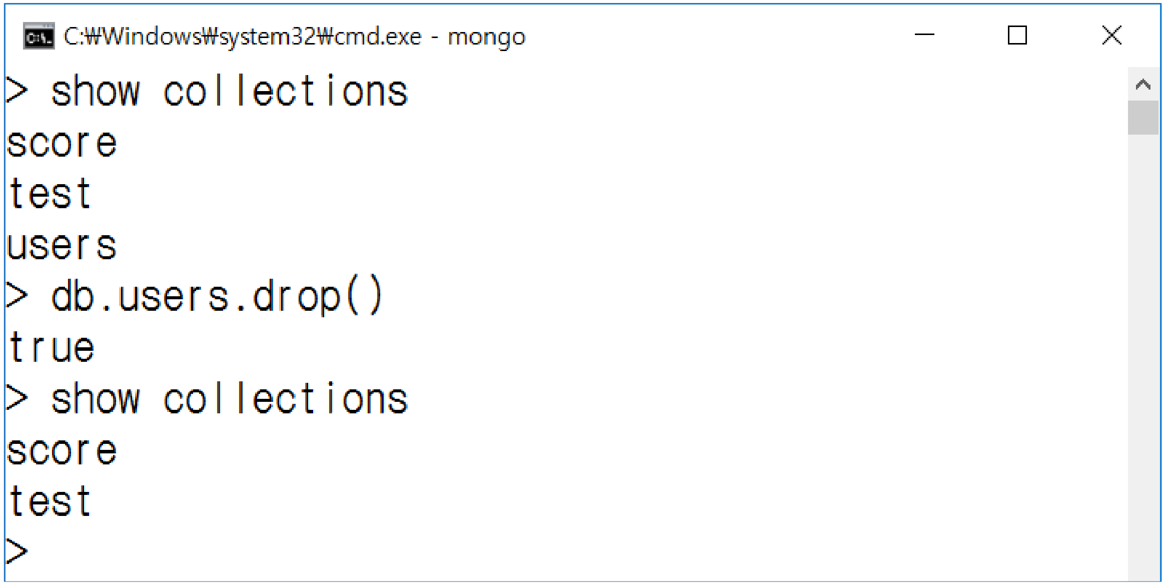


```
> db.users.remove( { } )
WriteResult({ "nRemoved" : 1 })
> db.users.find()
> _
```

```
db.users.remove( { } )
```

■ MongoDB CRUD

● 컬렉션 삭제



```
> show collections
score
test
users
> db.users.drop()
true
> show collections
score
test
>
```

```
db.users.drop()
```

■ MongoDB CRUD – 연습문제

- ● prod collection 사용
- ● 100개의 제품 정보를 document에 저장
  - – for 반복문 사용
  - – 제품정보 필드 : name / price / count
    - · name : name-1 ~ name-100
    - · price : 1001 ~ 1100
    - · count : 10 ~ 1000

      ex) {name : 'name-1', price : 1001, count : 10}, …,

      …, {name : 'name-100', price : 1100, count : 1000}
- ● price가 1010원 보다 크거나 같고 1020원 보다 적거나 같은 제품 또는 price가 1070원 보다 크고 1100원 보다 적은 제품 조회

■ Index

- ● Document를 query 하는 작업량을 줄이기 위해 사용

  - index가 없으면 조건을 만족할 때까지 모든 Document를 순차적으로 조회

- ● 2개 이상의 index가 필요하다면 복합 index를 사용

  - a와 b 필드로 구성된 복합 index를 가지고 있다면
    a 또는 b에 대한 단일 index는 제거해도 상관없음

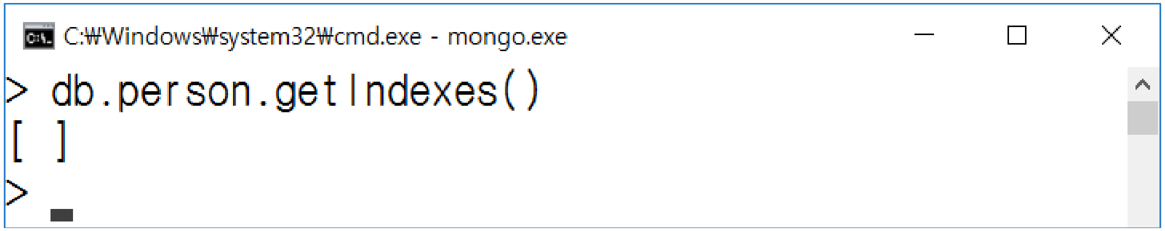
- ● _id 필드는 기본적으로 생성되는 index로 Document를 가리키는 Key로 사용

- ● index는 읽기 위주의 애플리케이션에서 유용하게 사용됨

- ● 쓰기/수정이 많다면 index를 사용하지 않거나 별도의 Index Collection 사용

■ Index

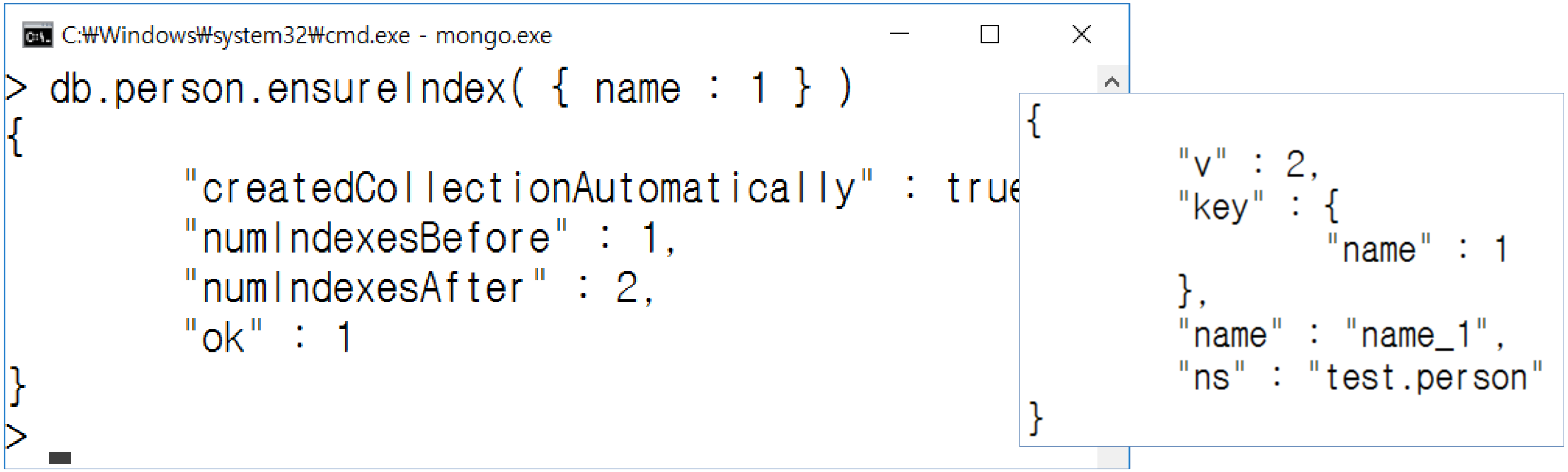
● Index 확인



```
> db.person.getIndexes()
[ ]
> _
```

```
db.person.getIndexes()
```

● Index 생성 (1 : 오름차순, -1 : 내림차순)



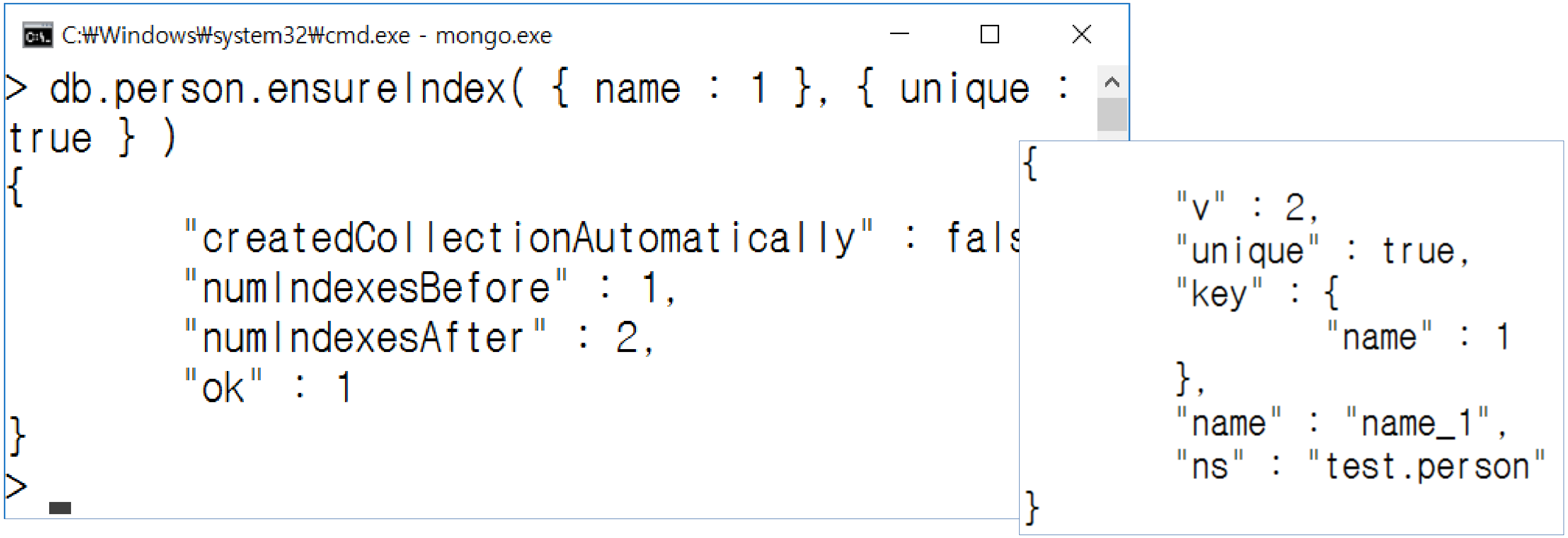
```
> db.person.ensureIndex( { name : 1 } )
{
        "createdCollectionAutomatically" : true
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
> _
```

```
{
        "v" : 2,
        "key" : {
                "name" : 1
        },
        "name" : "name_1",
        "ns" : "test.person"
}
```

```
db.person.ensureIndex( { name : 1 } )
```

■ Index

● Index 생성 (unique : 중복 데이터 저장 불가)

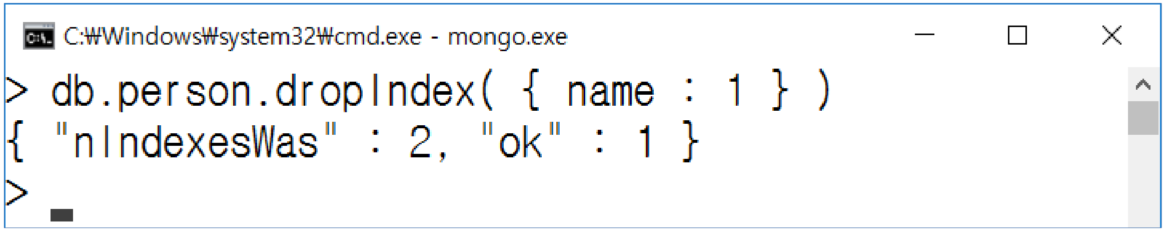


```
> db.person.ensureIndex( { name : 1 }, { unique :
true } )
{
        "createdCollectionAutomatically" : fals
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
>
```

```
{
        "v" : 2,
        "unique" : true,
        "key" : {
                "name" : 1
        },
        "name" : "name_1",
        "ns" : "test.person"
}
```

```
db.person.ensureIndex( { name : 1 }, { unique : true } )
```

● Index 삭제 (모든 인덱스 삭제 시 dropIndexes())

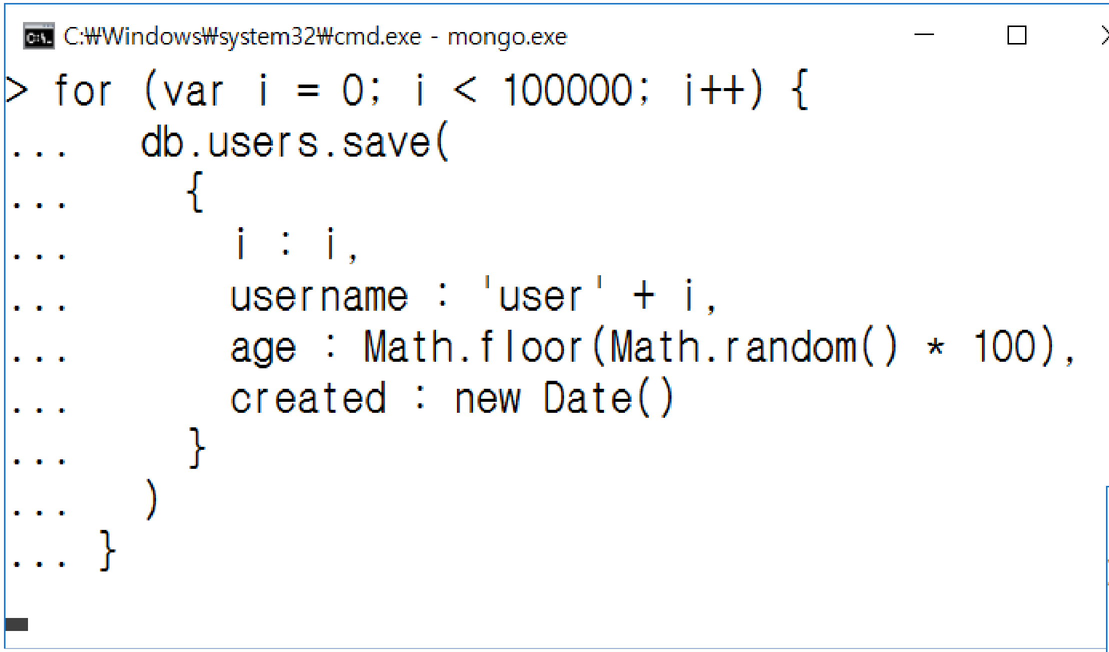


```
> db.person.dropIndex( { name : 1 } )
{ "nIndexesWas" : 2, "ok" : 1 }
>
```

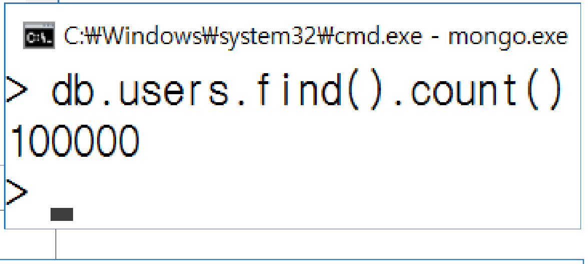
```
db.person.dropIndex( { name : 1 } )
```

■ Index 활용

- ● Document 10만개 생성

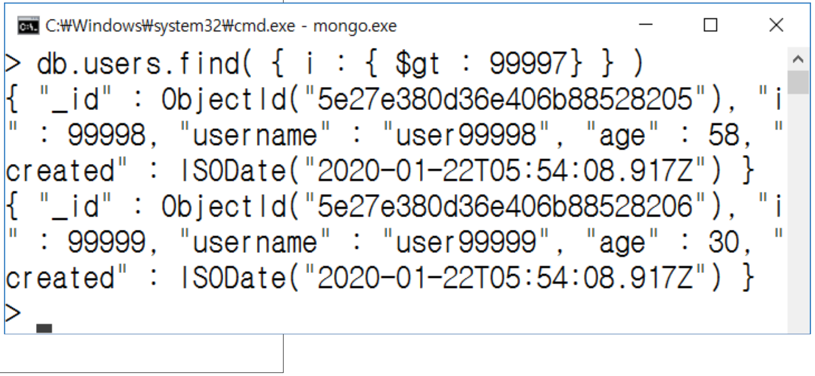


```
> for (var i = 0; i < 100000; i++) {
...     db.users.save(
...       {
...         i : i,
...         username : 'user' + i,
...         age : Math.floor(Math.random() * 100),
...         created : new Date()
...       }
...     )
... }
```



```
> db.users.find().count()
100000
>
```

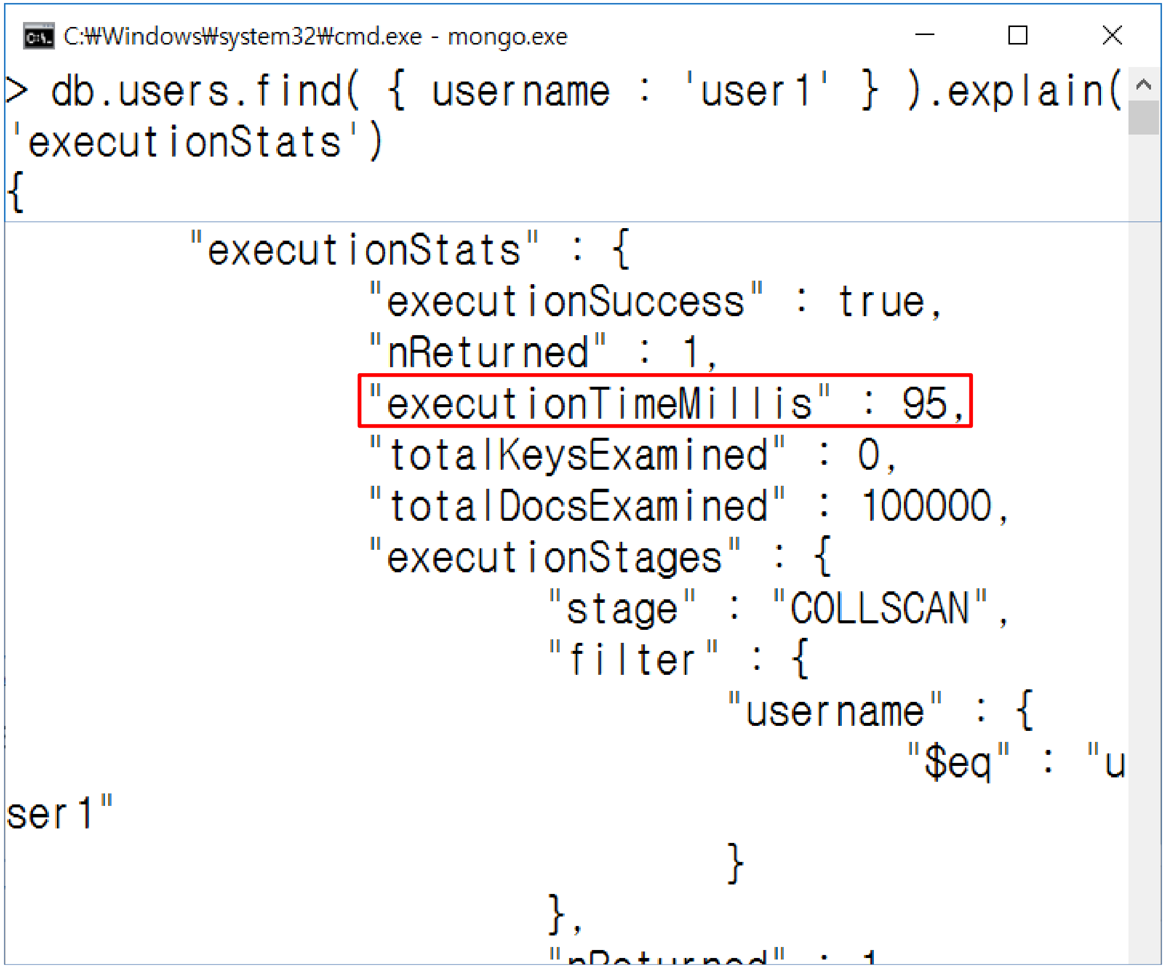
```
for (var i = 0; i < 100000; i++) {
  db.users.save(
    {
      i : i,
      username : 'user' + i,
      age : Math.floor(Math.random() * 100),
      created : new Date()
    }
  )
}
```



```
> db.users.find( { i : { $gt : 99997} } )
{ "_id" : ObjectId("5e27e380d36e406b88528205"), "i
" : 99998, "username" : "user99998", "age" : 58, "
created" : ISODate("2020-01-22T05:54:08.917Z") }
{ "_id" : ObjectId("5e27e380d36e406b88528206"), "i
" : 99999, "username" : "user99999", "age" : 30, "
created" : ISODate("2020-01-22T05:54:08.917Z") }
>
```

■ Index 활용

● Index 적용 전 검색 시간 확인

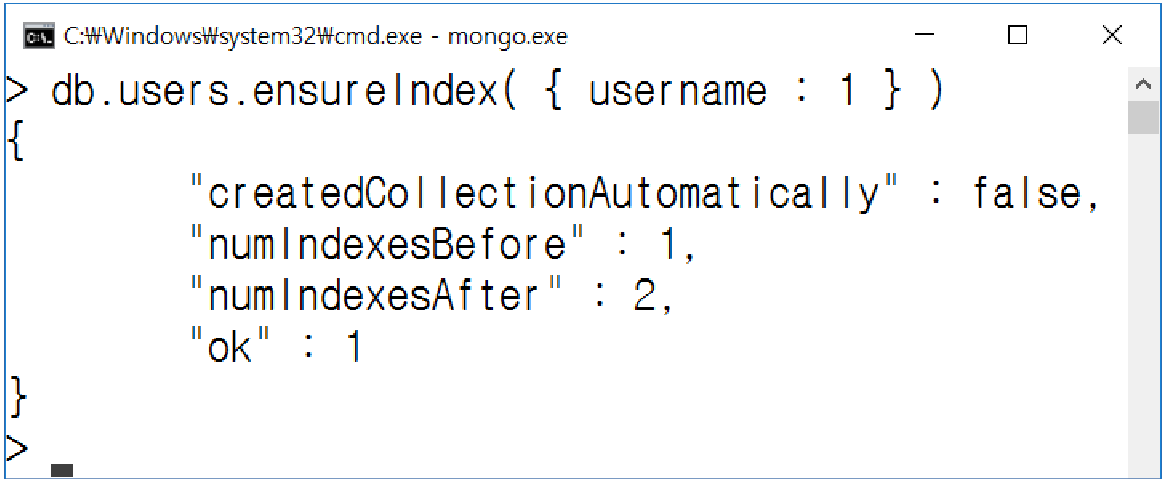


```
> db.users.find( { username : 'user1' } ).explain(
'executionStats')
{
        "executionStats" : {
                "executionSuccess" : true,
                "nReturned" : 1,
                "executionTimeMillis" : 95,
                "totalKeysExamined" : 0,
                "totalDocsExamined" : 100000,
                "executionStages" : {
                        "stage" : "COLLSCAN",
                        "filter" : {
                                "username" : {
                                        "$eq" : "u
ser1"
                                }
                        },
                        "nReturned" : 1
```

```
db.users.find( { username : 'user1' } ).explain('executionStats')
```

■ Index 활용

● Index 적용 (username 필드)

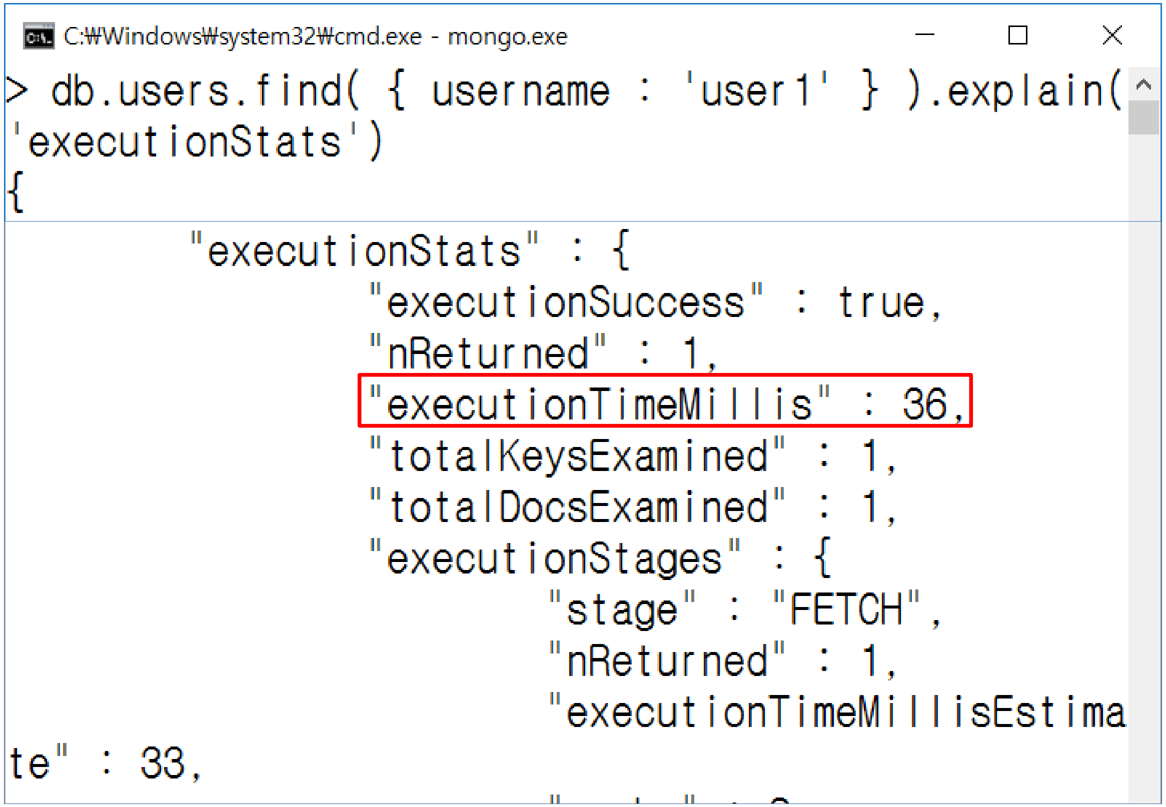


```
> db.users.ensureIndex( { username : 1 } )
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
> _
```

```
db.users.ensureIndex( { username : 1 } )
```

■ Index 활용

● Index 적용 후 검색 시간 확인



```
> db.users.find( { username : 'user1' } ).explain(
'executionStats')
{
        "executionStats" : {
                "executionSuccess" : true,
                "nReturned" : 1,
                "executionTimeMillis" : 36,
                "totalKeysExamined" : 1,
                "totalDocsExamined" : 1,
                "executionStages" : {
                        "stage" : "FETCH",
                        "nReturned" : 1,
                        "executionTimeMillisEstima
te" : 33,
```

```
db.users.find( { username : 'user1' } ).explain('executionStats')
```

■ pymongo 모듈

● 모듈 가져오기

```
from pymongo import MongoClient
```

● MongoDB 접속

```
client = MongoClient(host='localhost', port=27017)
```

● test 데이터베이스 지정

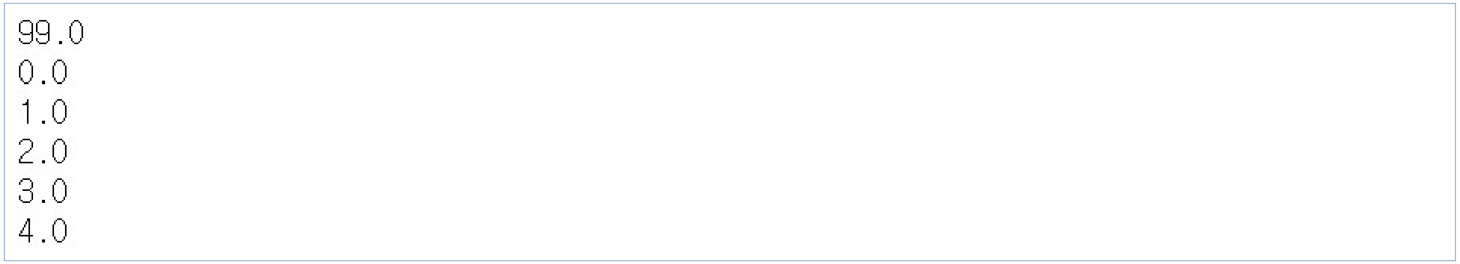
```
db = client['test']
```

● score 컬렉션 데이터 조회

```
cursor = db.score.find()
```

● 데이터 확인

```
for data in list(cursor):
    print(data['a'])
```



```
99.0
0.0
1.0
2.0
3.0
4.0
```

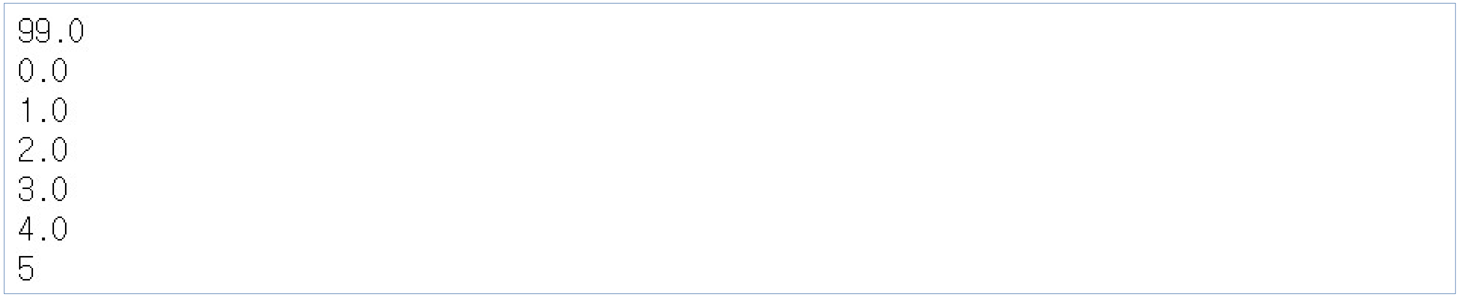
■ pymongo 모듈

● score 컬렉션 데이터 입력

```
db.score.save( { 'a' : 5, 'exam' : 6 } )
```

● score 컬렉션 데이터 조회

```
cursor.rewind()

for data in list(cursor):
    print(data['a'])
```



```
99.0
0.0
1.0
2.0
3.0
4.0
5
```

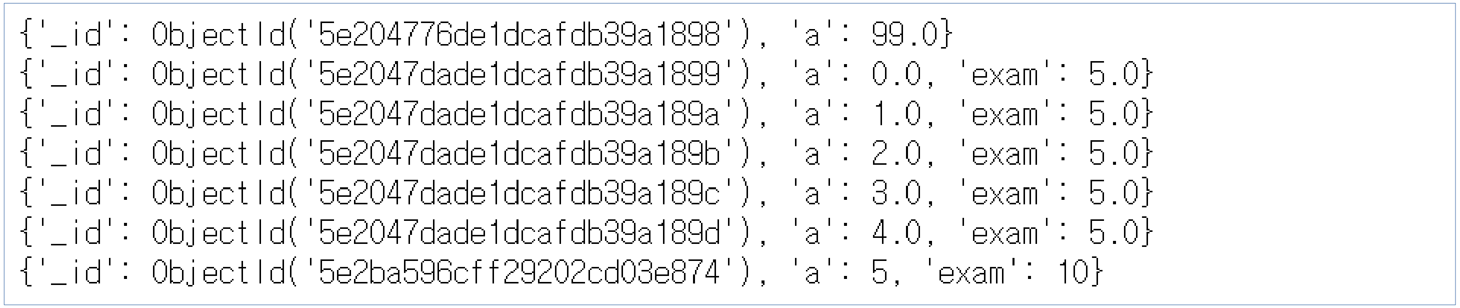
■ pymongo 모듈

● score 컬렉션 데이터 수정

```
db.score.update( { 'a' : 5 }, { '$set' : { 'exam' : 10 } } )
```

● score 컬렉션 데이터 조회

```
cursor.rewind()

for data in list(cursor):
    print(data)
```



```
{'_id': ObjectId('5e204776de1dcafdb39a1898'), 'a': 99.0}
{'_id': ObjectId('5e2047dade1dcafdb39a1899'), 'a': 0.0, 'exam': 5.0}
{'_id': ObjectId('5e2047dade1dcafdb39a189a'), 'a': 1.0, 'exam': 5.0}
{'_id': ObjectId('5e2047dade1dcafdb39a189b'), 'a': 2.0, 'exam': 5.0}
{'_id': ObjectId('5e2047dade1dcafdb39a189c'), 'a': 3.0, 'exam': 5.0}
{'_id': ObjectId('5e2047dade1dcafdb39a189d'), 'a': 4.0, 'exam': 5.0}
{'_id': ObjectId('5e2ba596cff29202cd03e874'), 'a': 5, 'exam': 10}
```

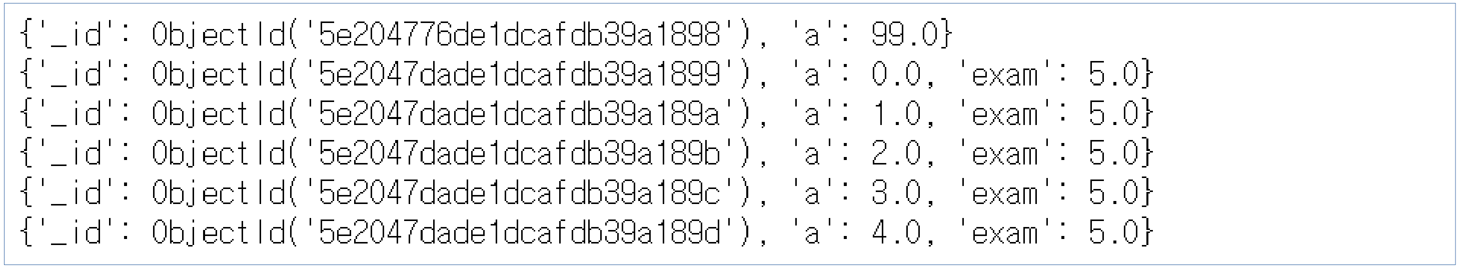
■ pymongo 모듈

● score 컬렉션 데이터 삭제

```
db.score.delete_one( { 'a' : 5 } )
```

● score 컬렉션 데이터 조회

```
cursor.rewind()

for data in list(cursor):
    print(data)
```

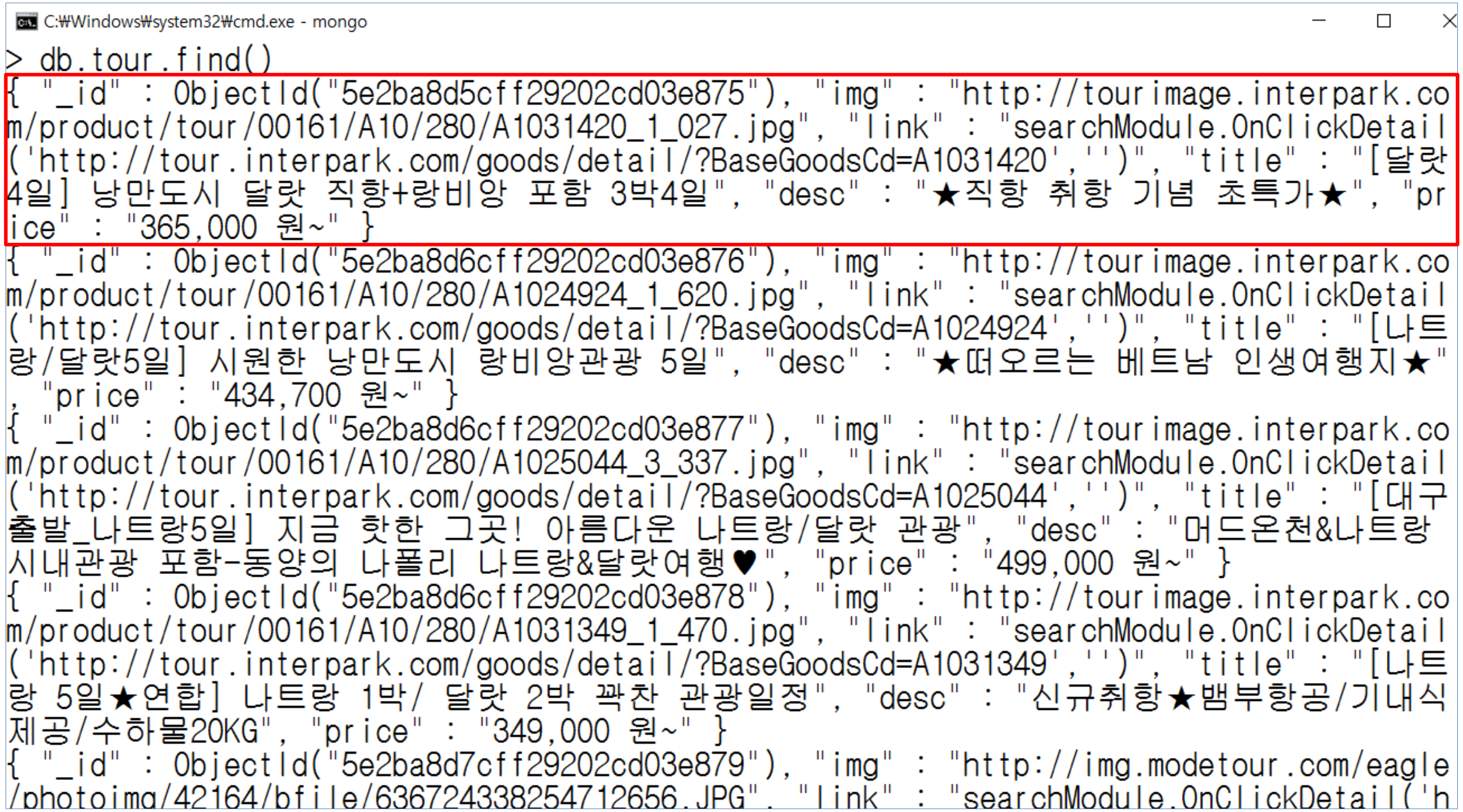


```
{'_id': ObjectId('5e204776de1dcafdb39a1898'), 'a': 99.0}
{'_id': ObjectId('5e2047dade1dcafdb39a1899'), 'a': 0.0, 'exam': 5.0}
{'_id': ObjectId('5e2047dade1dcafdb39a189a'), 'a': 1.0, 'exam': 5.0}
{'_id': ObjectId('5e2047dade1dcafdb39a189b'), 'a': 2.0, 'exam': 5.0}
{'_id': ObjectId('5e2047dade1dcafdb39a189c'), 'a': 3.0, 'exam': 5.0}
{'_id': ObjectId('5e2047dade1dcafdb39a189d'), 'a': 4.0, 'exam': 5.0}
```

■ pymongo 모듈

● 인터파크 투어에서 조회한 결과를 MongoDB로 저장하기

- 컬렉션 : tour
- 필드 : 이미지경로-img, 상세페이지경로-link, 상품명-title,
  상품설명-desc, 가격-price



```
> db.tour.find()
{ "_id" : ObjectId("5e2ba8d5cff29202cd03e875"), "img" : "http://tourimage.interpark.co
m/product/tour/00161/A10/280/A1031420_1_027.jpg", "link" : "searchModule.OnClickDetail
('http://tour.interpark.com/goods/detail/?BaseGoodsCd=A1031420','')", "title" : "[달랏
4일] 낭만도시 달랏 직항+랑비앙 포함 3박4일", "desc" : "★직항 취항 기념 초특가★", "pr
ice" : "365,000 원~" }
{ "_id" : ObjectId("5e2ba8d6cff29202cd03e876"), "img" : "http://tourimage.interpark.co
m/product/tour/00161/A10/280/A1024924_1_620.jpg", "link" : "searchModule.OnClickDetail
('http://tour.interpark.com/goods/detail/?BaseGoodsCd=A1024924','')", "title" : "[나트
랑/달랏5일] 시원한 낭만도시 랑비앙관광 5일", "desc" : "★떠오르는 베트남 인생여행지★"
, "price" : "434,700 원~" }
{ "_id" : ObjectId("5e2ba8d6cff29202cd03e877"), "img" : "http://tourimage.interpark.co
m/product/tour/00161/A10/280/A1025044_3_337.jpg", "link" : "searchModule.OnClickDetail
('http://tour.interpark.com/goods/detail/?BaseGoodsCd=A1025044','')", "title" : "[대구
출발_나트랑5일] 지금 핫한 그곳! 아름다운 나트랑/달랏 관광", "desc" : "머드온천&나트랑
시내관광 포함-동양의 나폴리 나트랑&달랏여행♥", "price" : "499,000 원~" }
{ "_id" : ObjectId("5e2ba8d6cff29202cd03e878"), "img" : "http://tourimage.interpark.co
m/product/tour/00161/A10/280/A1031349_1_470.jpg", "link" : "searchModule.OnClickDetail
('http://tour.interpark.com/goods/detail/?BaseGoodsCd=A1031349','')", "title" : "[나트
랑 5일★연합] 나트랑 1박/ 달랏 2박 꽉찬 관광일정", "desc" : "신규취항★뱀부항공/기내식
제공/수하물20KG", "price" : "349,000 원~" }
{ "_id" : ObjectId("5e2ba8d7cff29202cd03e879"), "img" : "http://img.modetour.com/eagle
/photoimg/42164/bfile/636724338254712656.JPG", "link" : "searchModule.OnClickDetail('h
```

■ pymongo 모듈

● 인터파크 투어에서 조회한 결과를 MongoDB로 저장하기

- 셀레니움 크롤링 소스 코드

```
from selenium import webdriver as wd
driver = wd.Chrome(executable_path='chromedriver.exe')

from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

import time

driver.get('http://tour.interpark.com')

driver.implicitly_wait(10)

driver.find_element_by_id('SearchGNBText').send_keys('달랏')

driver.find_element_by_css_selector('button.search-btn').click()

WebDriverWait(driver, 10).until(
    EC.presence_of_element_located((By.CLASS_NAME, 'oTravelBox'))
)
driver.find_element_by_css_selector('.oTravelBox .moreBtn').click()
```

■ pymongo 모듈

● 인터파크 투어에서 조회한 결과를 MongoDB로 저장하기

– 셀레니움 크롤링 소스 코드

```
for page in range(1, 2):
    driver.execute_script("searchModule.SetCategoryList(%s, '')" % page)
    time.sleep(2)
    boxItems = driver.find_elements_by_css_selector(
             '.panelZone > .oTravelBox > .boxList > li')
    for li in boxItems:
        print('이미지', li.find_element_by_css_selector('img.img').get_attribute('src'))
        print('링크', li.find_element_by_css_selector('a').get_attribute('onclick'))
        print('상품명', li.find_element_by_css_selector('h5.proTit').text)
        print('추가설명', li.find_element_by_css_selector('.proSub').text)
        print('가격', li.find_element_by_css_selector('strong.proPrice').text)
        for info in li.find_elements_by_css_selector('.info-row .proInfo'):
            print(info.text)
        print('=' * 20)
driver.close()
```