MongoDB 기초

2016년도 2학기 2학년 방과후학교

MongoDB 실행

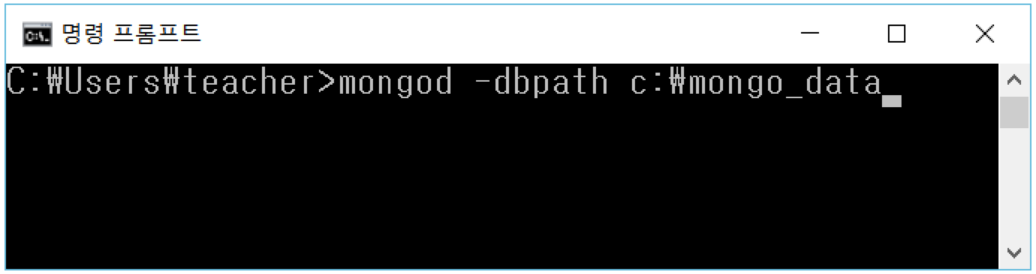
- Mongo DB 데이터 저장 폴더 생성

  > C:₩mongo_data

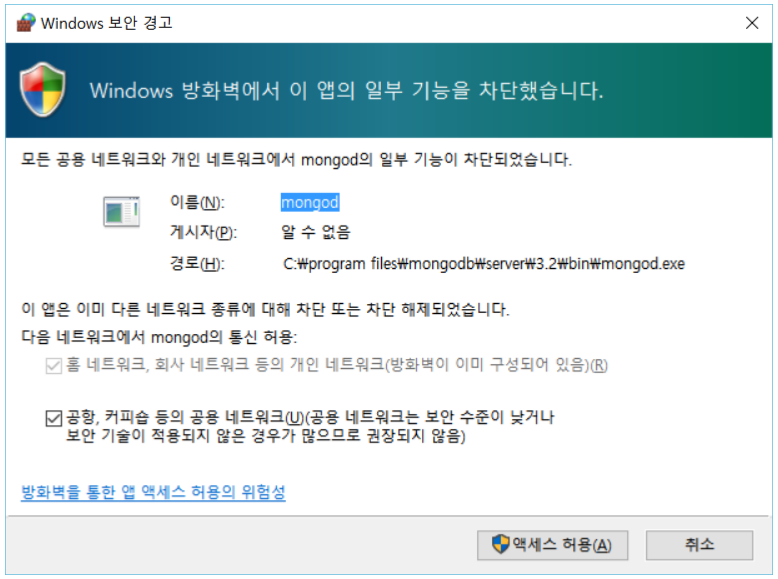
- Mongo DB 실행

  > mongod -dbpath c:₩mongo_data

- mongod : MongoDB의 서버 프로그램 (Daemon)



```
C:₩Users₩teacher>mongod -dbpath c:₩mongo_data_
```



Windows 방화벽에서 이 앱의 일부 기능을 차단했습니다.

모든 공용 네트워크와 개인 네트워크에서 mongod의 일부 기능이 차단되었습니다.

이름(N): mongod
게시자(P): 알 수 없음
경로(H): C:₩program files₩mongodb₩server₩3.2₩bin₩mongod.exe

이 앱은 이미 다른 네트워크 종류에 대해 차단 또는 차단 해제되었습니다.

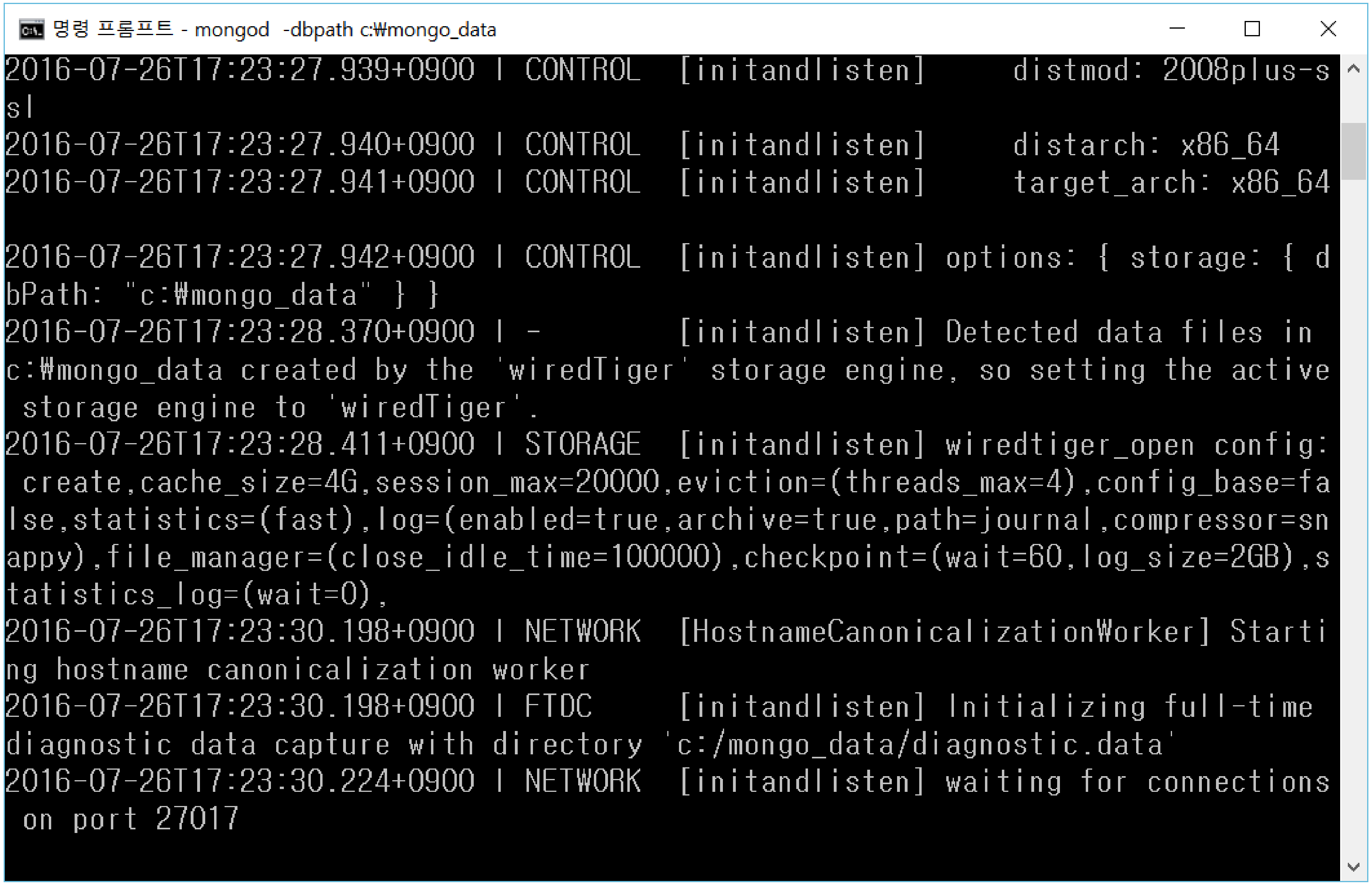
다음 네트워크에서 mongod의 통신 허용:

[✓] 홈 네트워크, 회사 네트워크 등의 개인 네트워크(방화벽이 이미 구성되어 있음)(R)

[✓] 공항, 커피숍 등의 공용 네트워크(U)(공용 네트워크는 보안 수준이 낮거나 보안 기술이 적용되지 않은 경우가 많으므로 권장되지 않음)

방화벽을 통한 앱 액세스 허용의 위험성

MongoDB 실행



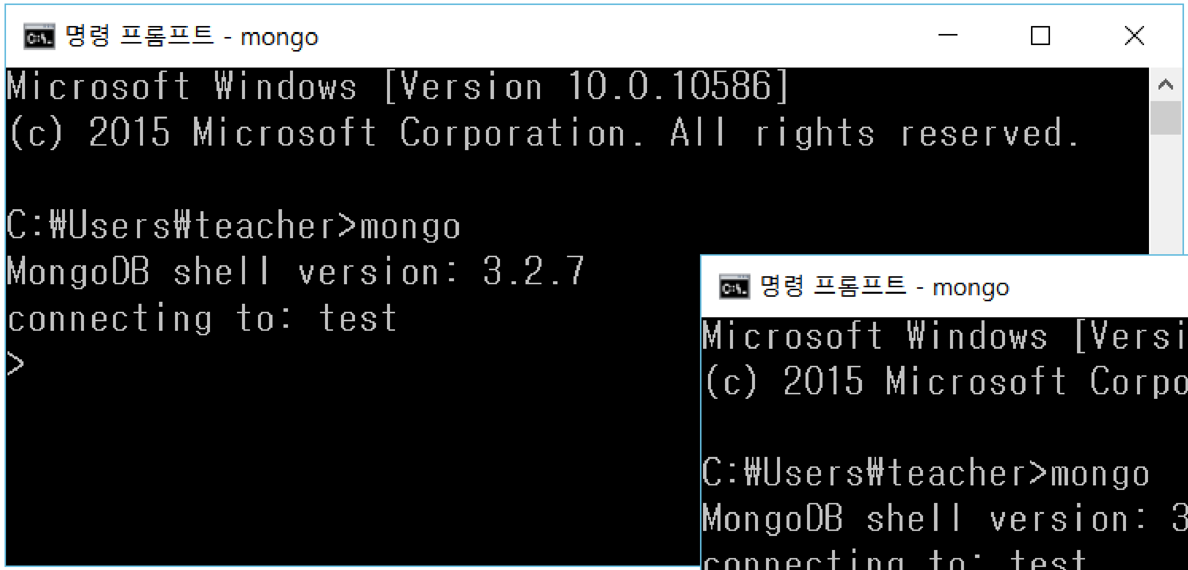
```
2016-07-26T17:23:27.939+0900 I CONTROL  [initandlisten]     distmod: 2008plus-s
sl
2016-07-26T17:23:27.940+0900 I CONTROL  [initandlisten]     distarch: x86_64
2016-07-26T17:23:27.941+0900 I CONTROL  [initandlisten]     target_arch: x86_64

2016-07-26T17:23:27.942+0900 I CONTROL  [initandlisten] options: { storage: { d
bPath: "c:₩mongo_data" } }
2016-07-26T17:23:28.370+0900 I -        [initandlisten] Detected data files in
c:₩mongo_data created by the 'wiredTiger' storage engine, so setting the active
 storage engine to 'wiredTiger'.
2016-07-26T17:23:28.411+0900 I STORAGE  [initandlisten] wiredtiger_open config:
 create,cache_size=4G,session_max=20000,eviction=(threads_max=4),config_base=fa
lse,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=sn
appy),file_manager=(close_idle_time=100000),checkpoint=(wait=60,log_size=2GB),s
tatistics_log=(wait=0),
2016-07-26T17:23:30.198+0900 I NETWORK  [HostnameCanonicalizationWorker] Starti
ng hostname canonicalization worker
2016-07-26T17:23:30.198+0900 I FTDC     [initandlisten] Initializing full-time
diagnostic data capture with directory 'c:/mongo_data/diagnostic.data'
2016-07-26T17:23:30.224+0900 I NETWORK  [initandlisten] waiting for connections
 on port 27017
```

MongoDB 실행

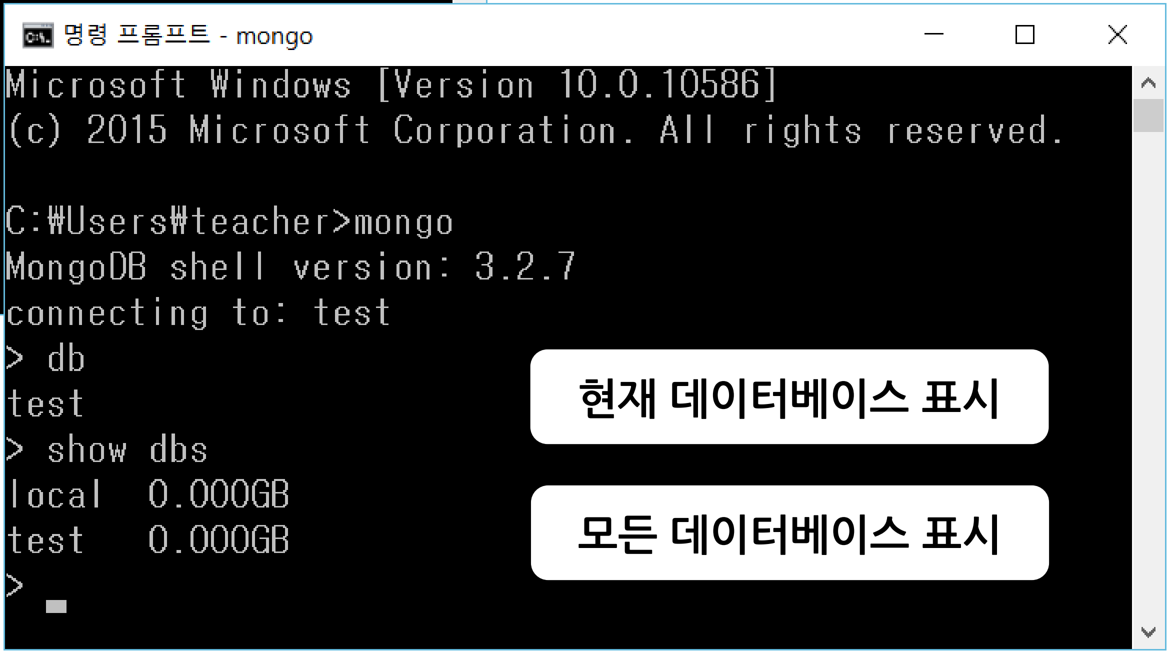
- MongoDB Shell 프로그램 실행

  > mongo



```
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:₩Users₩teacher>mongo
MongoDB shell version: 3.2.7
connecting to: test
>
```



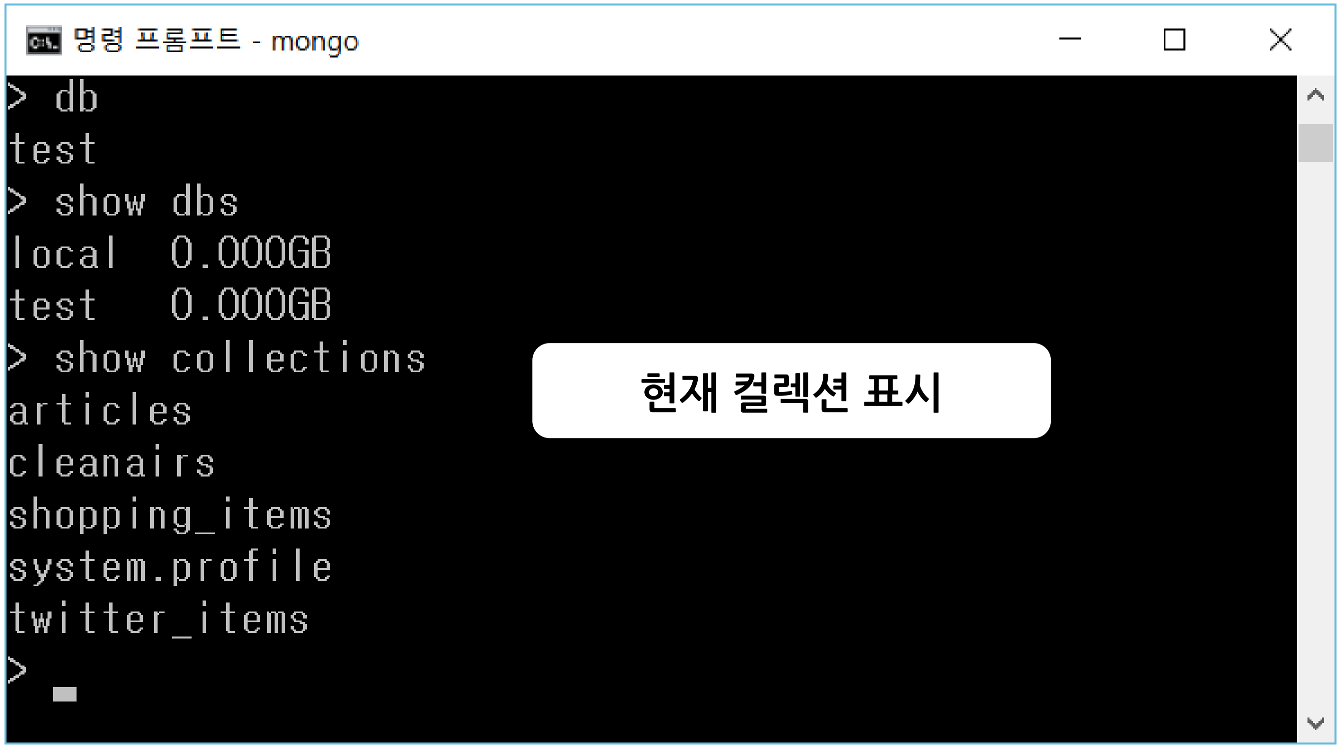
```
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:₩Users₩teacher>mongo
MongoDB shell version: 3.2.7
connecting to: test
> db
test
> show dbs
local   0.000GB
test    0.000GB
>
```

MongoDB 실행

- 전체 Collection 정보 보기

  > show collections



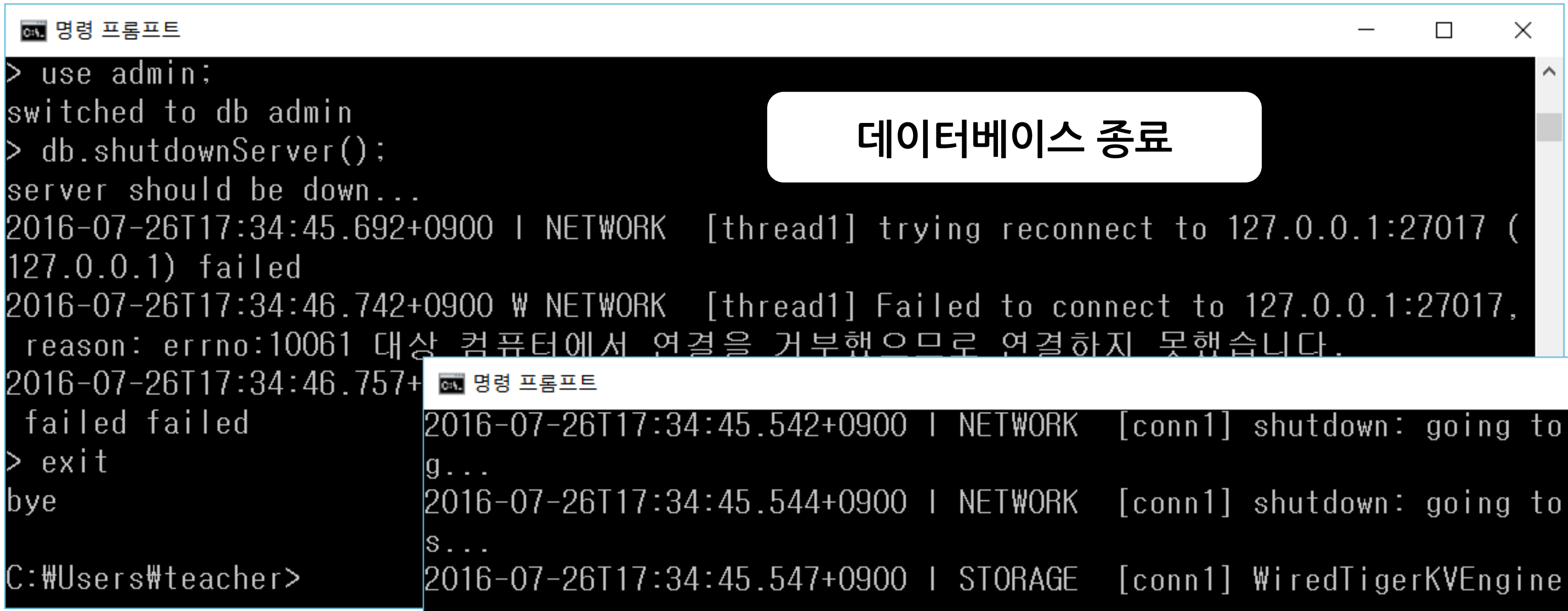
```
> db
test
> show dbs
local   0.000GB
test    0.000GB
> show collections
articles
cleanairs
shopping_items
system.profile
twitter_items
>
```

MongoDB 종료

- MongoDB Clean Exit

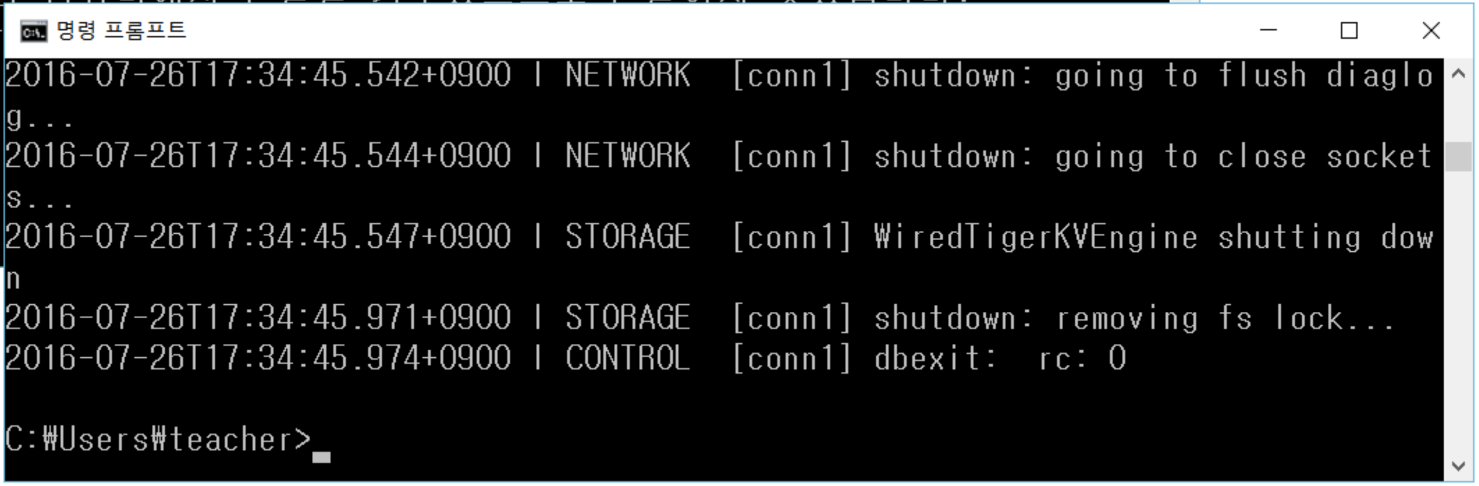
> use admin;   // admin 데이터베이스로의 전환

> db.shutdownServer();   // DB 서버 stop



```
> use admin;
switched to db admin
> db.shutdownServer();
server should be down...
2016-07-26T17:34:45.692+0900 I NETWORK  [thread1] trying reconnect to 127.0.0.1:27017 (
127.0.0.1) failed
2016-07-26T17:34:46.742+0900 W NETWORK  [thread1] Failed to connect to 127.0.0.1:27017,
 reason: errno:10061 대상 컴퓨터에서 연결을 거부했으므로 연결하지 못했습니다.
2016-07-26T17:34:46.757+
 failed failed
> exit
bye

C:\Users\teacher>
```



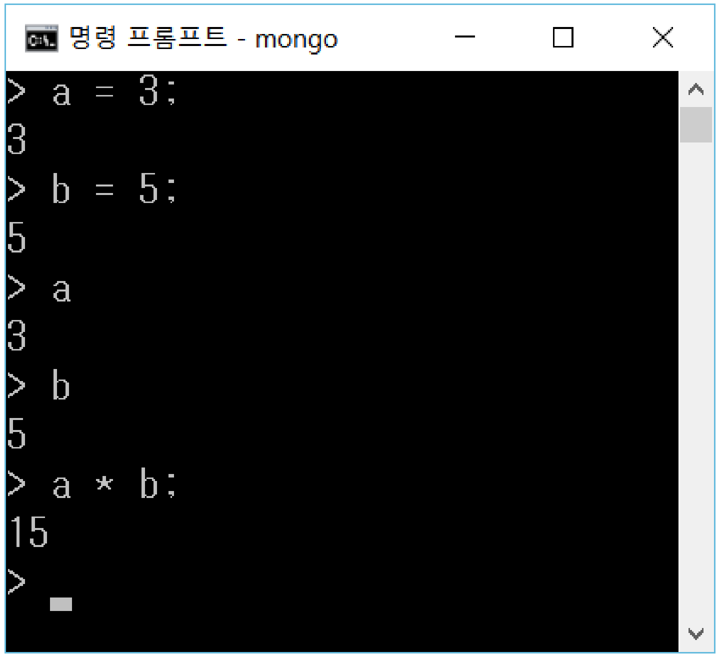
```
2016-07-26T17:34:45.542+0900 I NETWORK  [conn1] shutdown: going to flush diaglo
g...
2016-07-26T17:34:45.544+0900 I NETWORK  [conn1] shutdown: going to close socket
s...
2016-07-26T17:34:45.547+0900 I STORAGE  [conn1] WiredTigerKVEngine shutting dow
n
2016-07-26T17:34:45.971+0900 I STORAGE  [conn1] shutdown: removing fs lock...
2016-07-26T17:34:45.974+0900 I CONTROL  [conn1] dbexit:  rc: 0

C:\Users\teacher>_
```

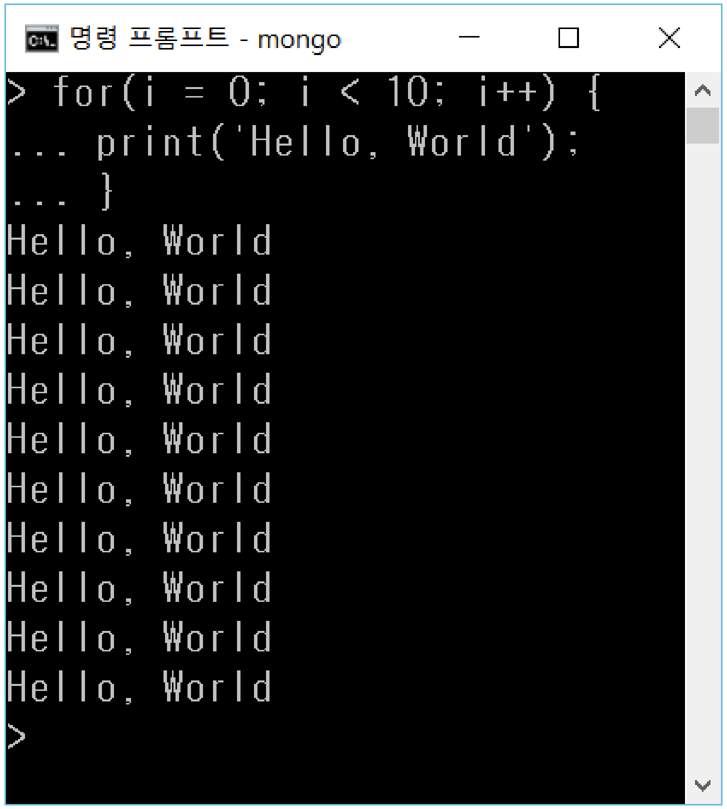
[실습] MongoDB 실행/종료

- MongoDB 데몬 구동
- MongoDB 쉘 프로그램 실행
- 모든 데이터베이스 검색
- MongoDB 종료

JavaScript 실행

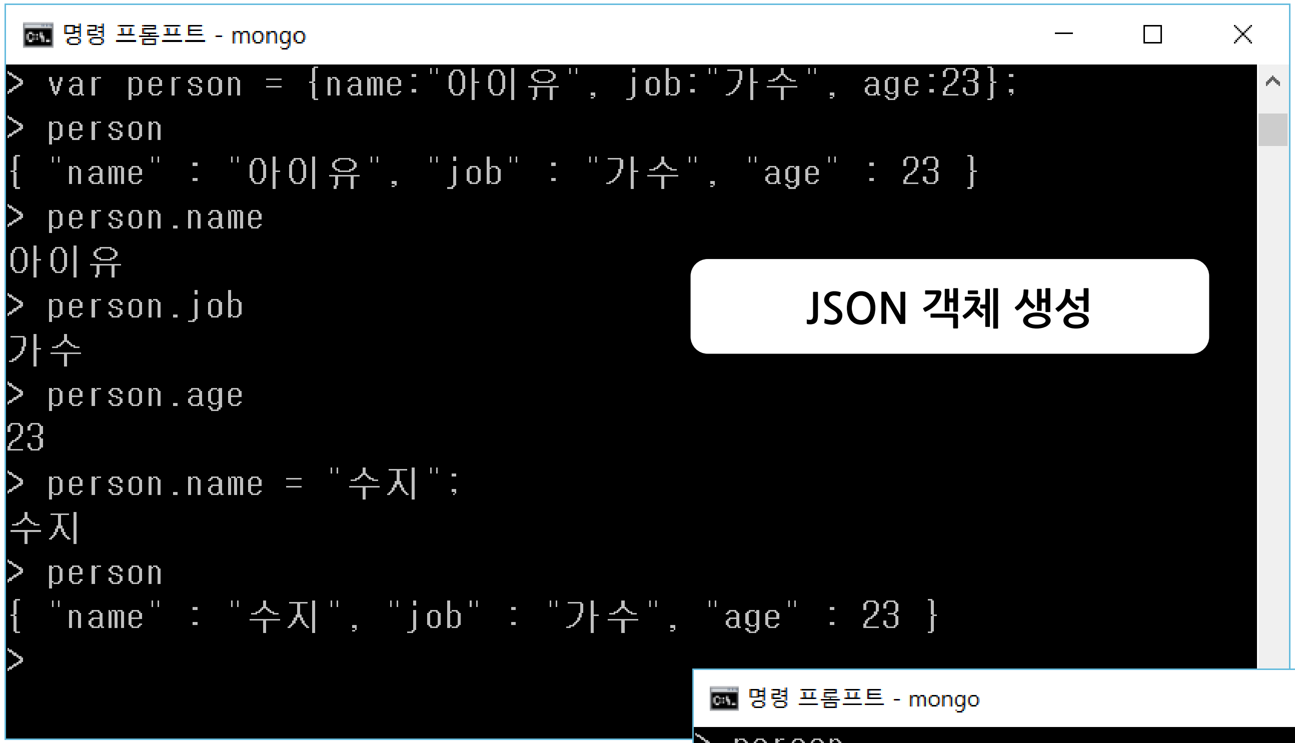


```
> a = 3;
3
> b = 5;
5
> a
3
> b
5
> a * b;
15
>
```

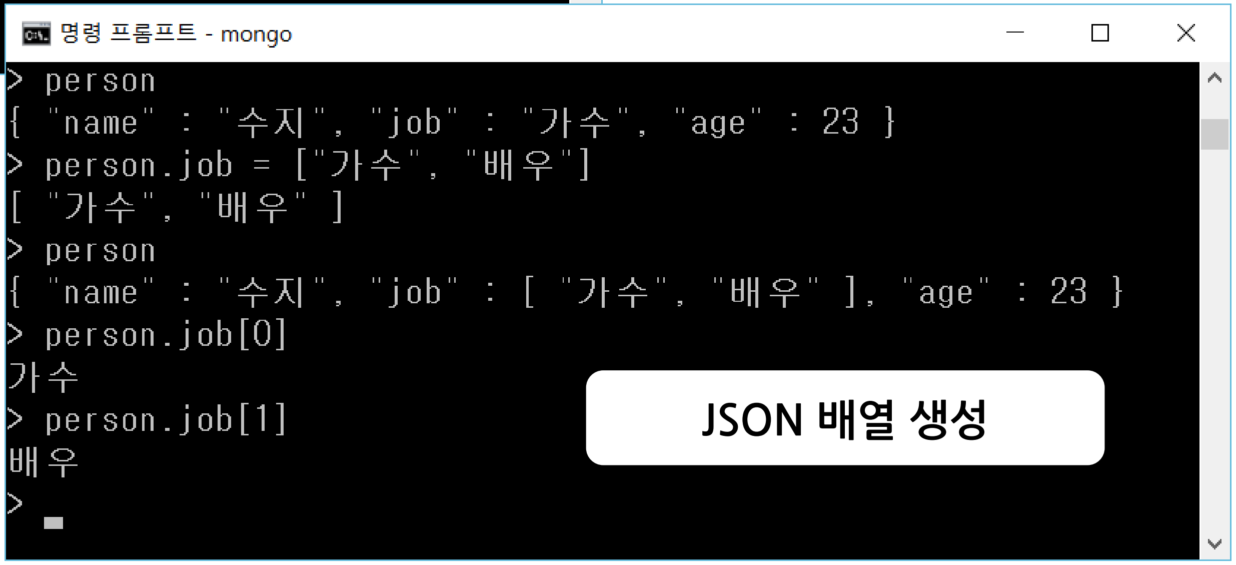


```
> for(i = 0; i < 10; i++) {
... print('Hello, World');
... }
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
>
```

JavaScript 실행


명령 프롬프트 - mongo

```
> var person = {name:"아이유", job:"가수", age:23};
> person
{ "name" : "아이유", "job" : "가수", "age" : 23 }
> person.name
아이유
> person.job
가수
> person.age
23
> person.name = "수지";
수지
> person
{ "name" : "수지", "job" : "가수", "age" : 23 }
>
```

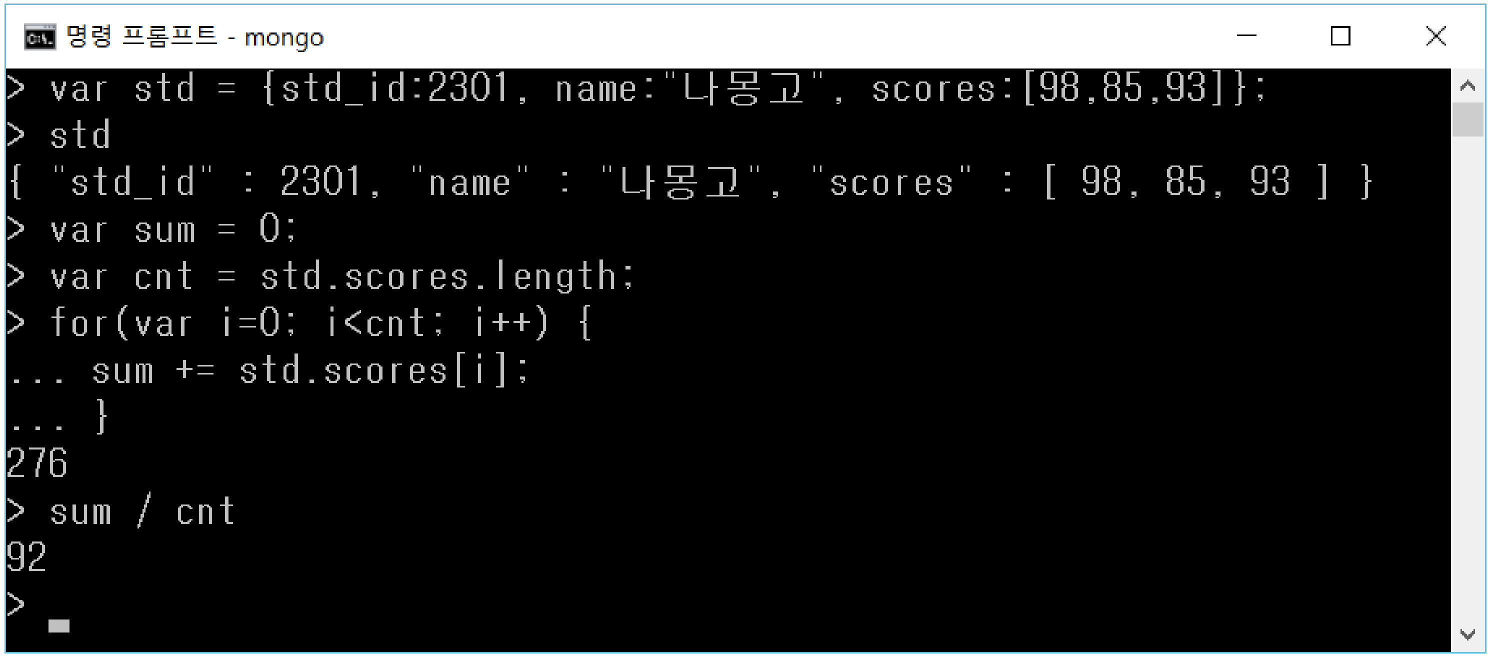

명령 프롬프트 - mongo

```
> person
{ "name" : "수지", "job" : "가수", "age" : 23 }
> person.job = ["가수", "배우"]
[ "가수", "배우" ]
> person
{ "name" : "수지", "job" : [ "가수", "배우" ], "age" : 23 }
> person.job[0]
가수
> person.job[1]
배우
>
```

[실습] 점수 평균 계산

- 학생 JSON 객체를 생성한 후 점수 평균을 계산해보자!

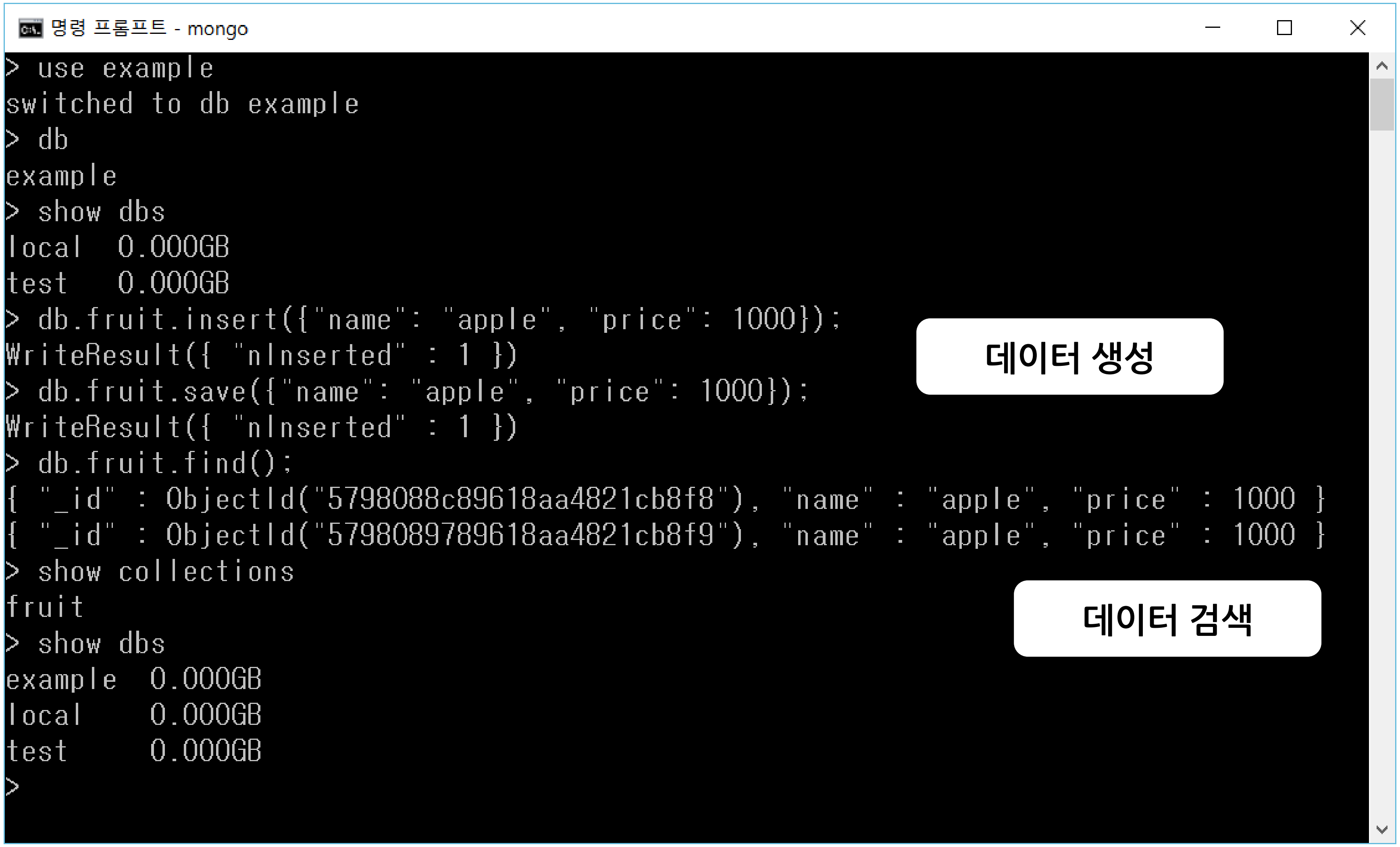
- std_id : 2301, name : 나몽고, scores : [98, 85, 93]



```
> var std = {std_id:2301, name:"나몽고", scores:[98,85,93]};
> std
{ "std_id" : 2301, "name" : "나몽고", "scores" : [ 98, 85, 93 ] }
> var sum = 0;
> var cnt = std.scores.length;
> for(var i=0; i<cnt; i++) {
... sum += std.scores[i];
... }
276
> sum / cnt
92
>
```

데이터베이스 생성

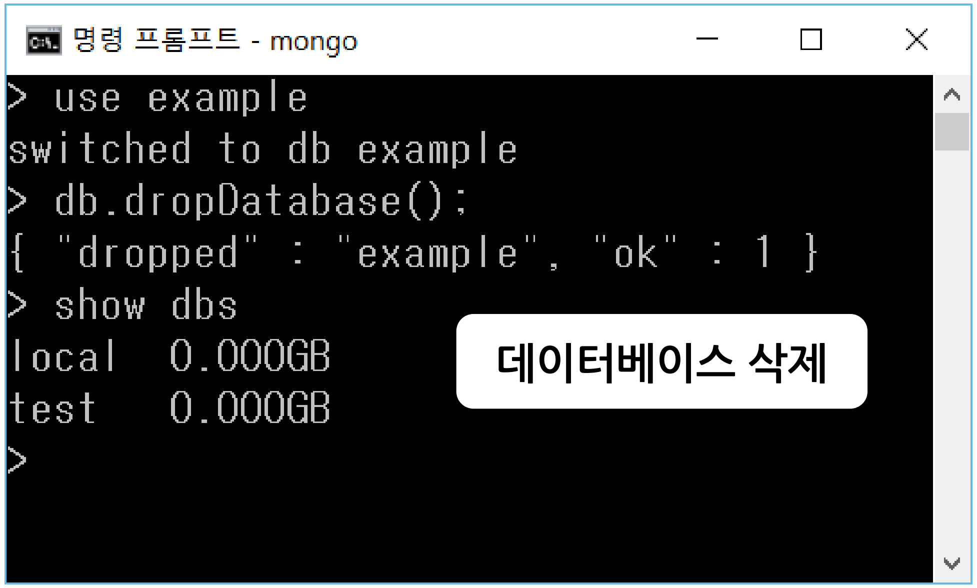
- use 데이터베이스명 // 없으면 생성, 있으면 전환



```
> use example
switched to db example
> db
example
> show dbs
local   0.000GB
test    0.000GB
> db.fruit.insert({"name": "apple", "price": 1000});
WriteResult({ "nInserted" : 1 })
> db.fruit.save({"name": "apple", "price": 1000});
WriteResult({ "nInserted" : 1 })
> db.fruit.find();
{ "_id" : ObjectId("5798088c89618aa4821cb8f8"), "name" : "apple", "price" : 1000 }
{ "_id" : ObjectId("5798089789618aa4821cb8f9"), "name" : "apple", "price" : 1000 }
> show collections
fruit
> show dbs
example   0.000GB
local     0.000GB
test      0.000GB
>
```

데이터베이스 삭제

- use 데이터베이스명
- db.dropDatabase( );

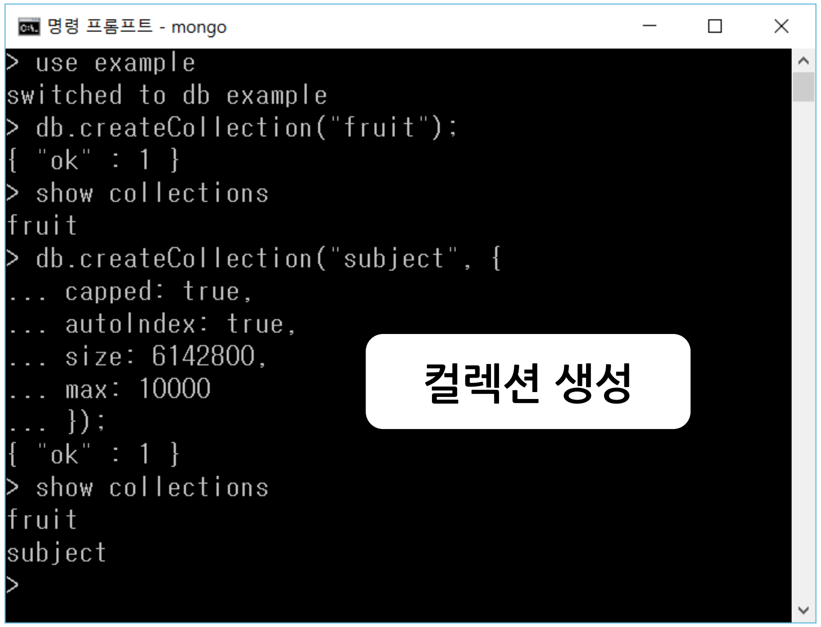


```
> use example
switched to db example
> db.dropDatabase();
{ "dropped" : "example", "ok" : 1 }
> show dbs
local   0.000GB
test    0.000GB
>
```

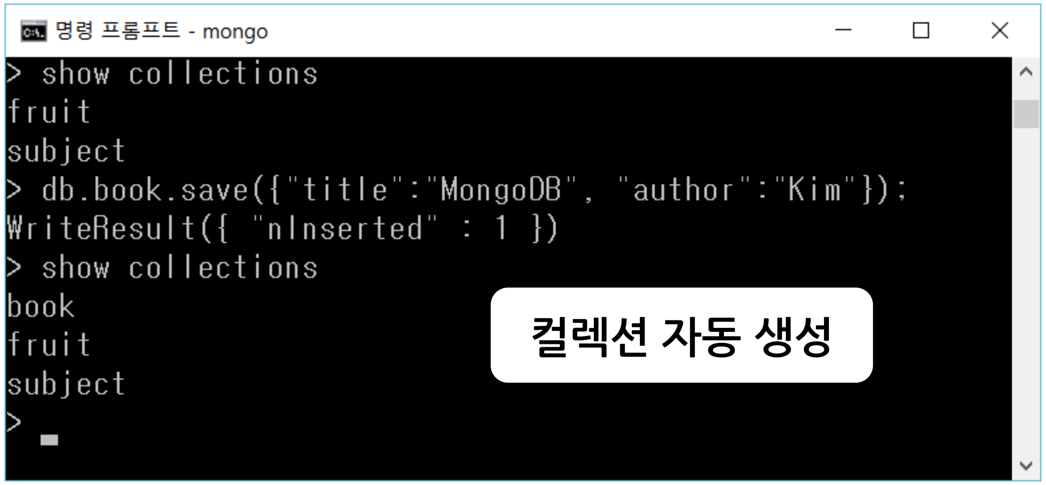
Collection 생성

- db.createCollection(name, [options]) 또는 Document 추가 시 컬렉션 자동 생성

Field	Type	설명
capped	Boolean	이 값을 true 로 설정하면 capped collection 을 활성화 시킵니다. Capped collection 이란, 고정된 크기(fixed size)를 가진 컬렉션으로서, size 가 초과되면 가장 오래된 데이터를 덮어씁니다. **이 값을 true로 설정하면 size 값을 꼭 설정해야합니다.**
autoIndex	Boolean	이 값을 true로 설정하면, _id 필드에 index를 자동으로 생성합니다. 기본값은 false 입니다.
size	Boolean	Capped collection 을 위해 해당 컬렉션의 최대 사이즈(maximum size)를 ~ bytes로 지정합니다.
max	Boolean	해당 컬렉션에 추가 할 수 있는 최대 갯수를 설정합니다.



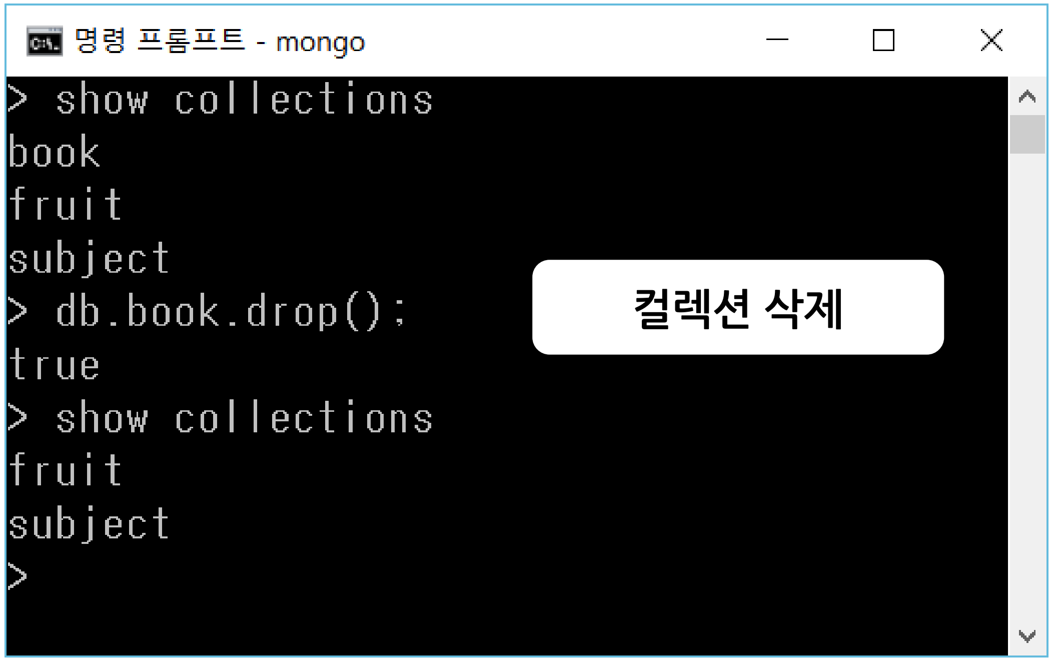
```
> use example
switched to db example
> db.createCollection("fruit");
{ "ok" : 1 }
> show collections
fruit
> db.createCollection("subject", {
... capped: true,
... autoIndex: true,
... size: 6142800,
... max: 10000
... });
{ "ok" : 1 }
> show collections
fruit
subject
>
```



```
> show collections
fruit
subject
> db.book.save({"title":"MongoDB", "author":"Kim"});
WriteResult({ "nInserted" : 1 })
> show collections
book
fruit
subject
>
```

Collection 삭제

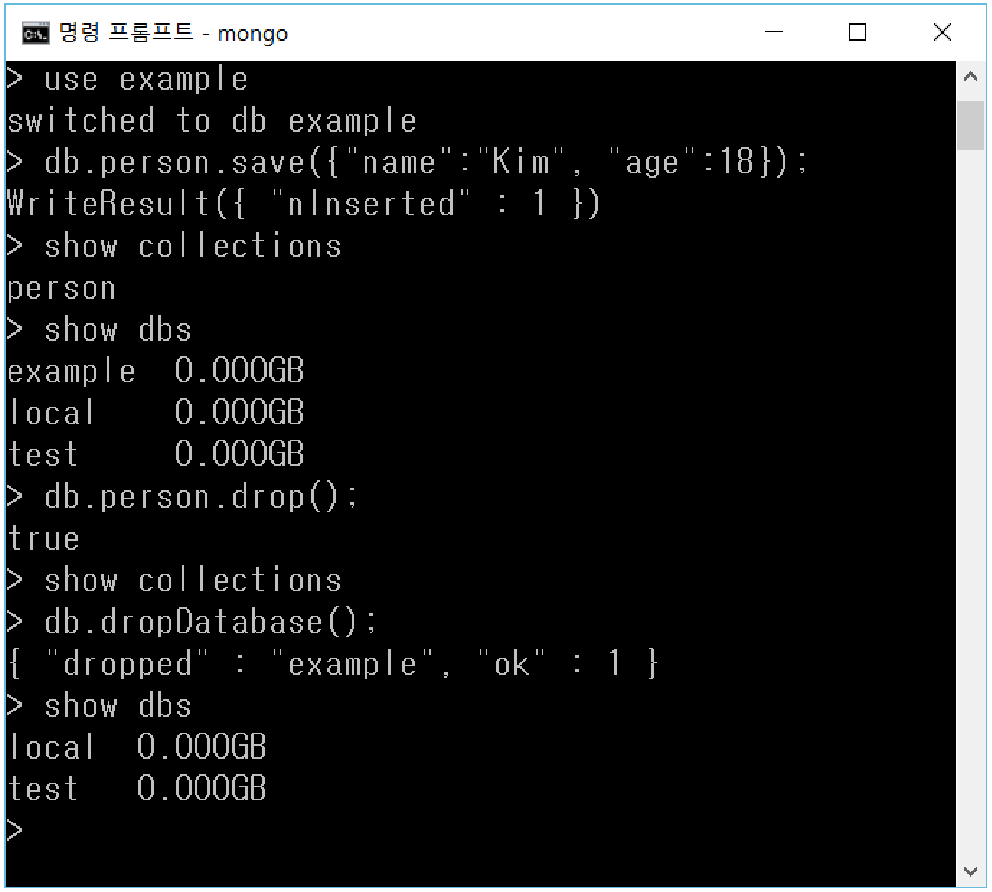
- db.컬렉션명.drop()



```
> show collections
book
fruit
subject
> db.book.drop();
true
> show collections
fruit
subject
>
```

[실습] 데이터베이스/컬렉션 생성, 삭제

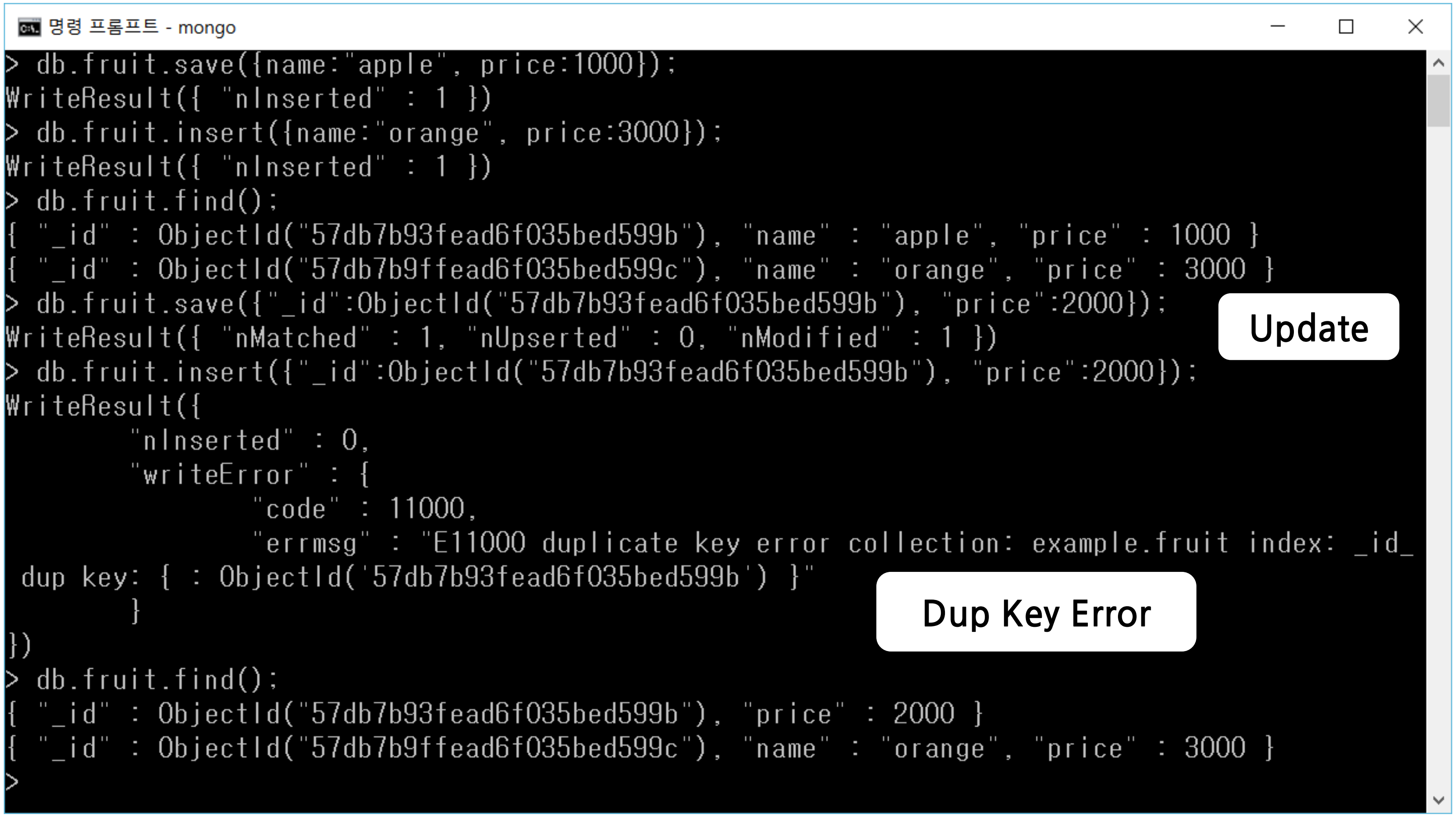
- example 데이터베이스 전환 (생성)
- person 컬렉션 생성 (document 생성 방식)
- show collections / show dbs
- person 컬렉션 삭제
- show collections
- example 데이터베이스 삭제
- show dbs



```
> use example
switched to db example
> db.person.save({"name":"Kim", "age":18});
WriteResult({ "nInserted" : 1 })
> show collections
person
> show dbs
example  0.000GB
local    0.000GB
test     0.000GB
> db.person.drop();
true
> show collections
> db.dropDatabase();
{ "dropped" : "example", "ok" : 1 }
> show dbs
local  0.000GB
test   0.000GB
>
```

데이터 저장 (한건)

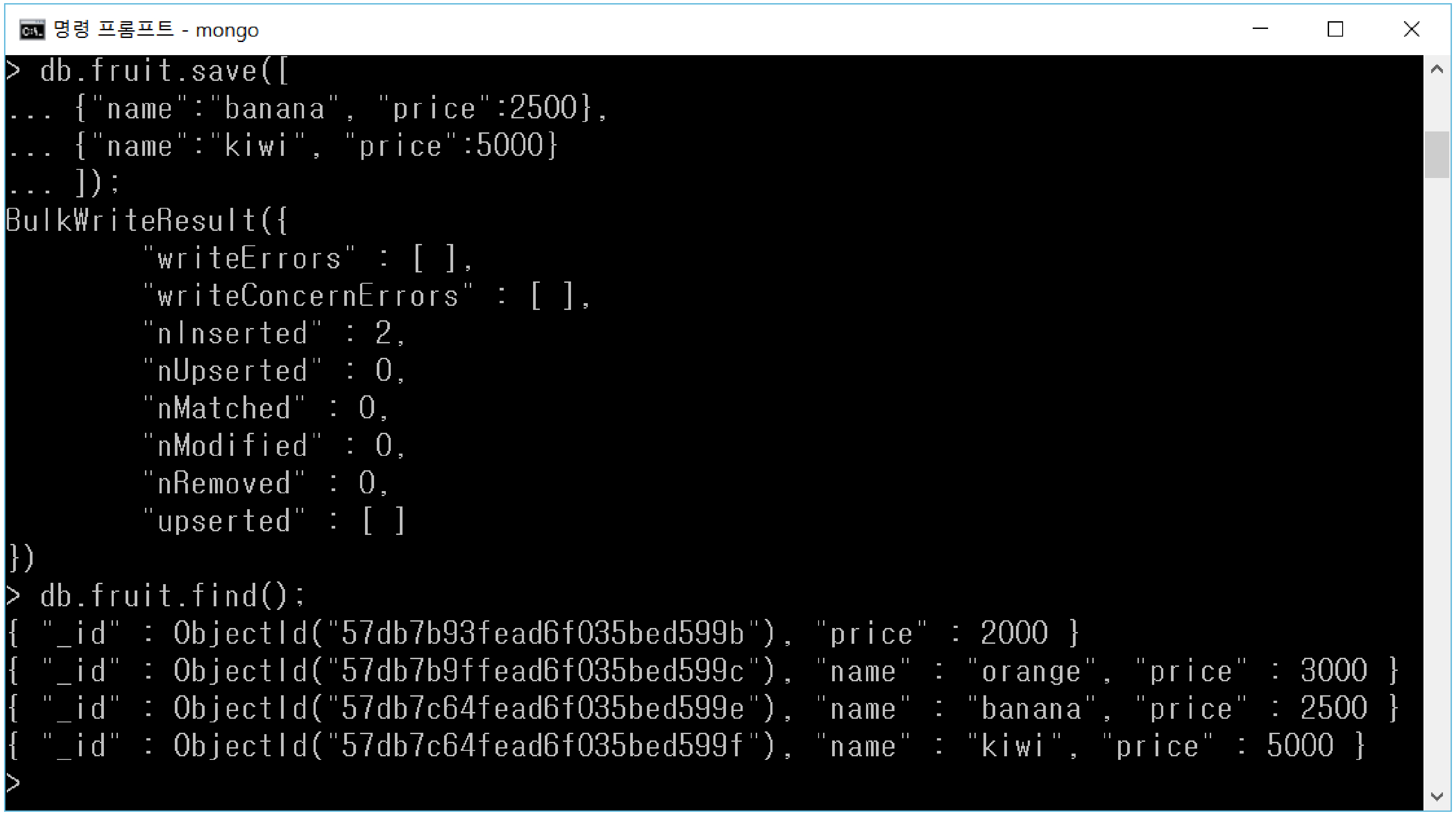
- db.컬렉션명.save(문서); // 문서 내 _id값과 동일 _id값이 있으면 Update
- db.컬렉션명.insert(문서); // 문서 내 _id값과 동일 _id값이 있으면 Dup Key



```
> db.fruit.save({name:"apple", price:1000});
WriteResult({ "nInserted" : 1 })
> db.fruit.insert({name:"orange", price:3000});
WriteResult({ "nInserted" : 1 })
> db.fruit.find();
{ "_id" : ObjectId("57db7b93fead6f035bed599b"), "name" : "apple", "price" : 1000 }
{ "_id" : ObjectId("57db7b9ffead6f035bed599c"), "name" : "orange", "price" : 3000 }
> db.fruit.save({"_id":ObjectId("57db7b93fead6f035bed599b"), "price":2000});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.fruit.insert({"_id":ObjectId("57db7b93fead6f035bed599b"), "price":2000});
WriteResult({
        "nInserted" : 0,
        "writeError" : {
                "code" : 11000,
                "errmsg" : "E11000 duplicate key error collection: example.fruit index: _id_
 dup key: { : ObjectId('57db7b93fead6f035bed599b') }"
        }
})
> db.fruit.find();
{ "_id" : ObjectId("57db7b93fead6f035bed599b"), "price" : 2000 }
{ "_id" : ObjectId("57db7b9ffead6f035bed599c"), "name" : "orange", "price" : 3000 }
>
```

데이터 저장 (다건)

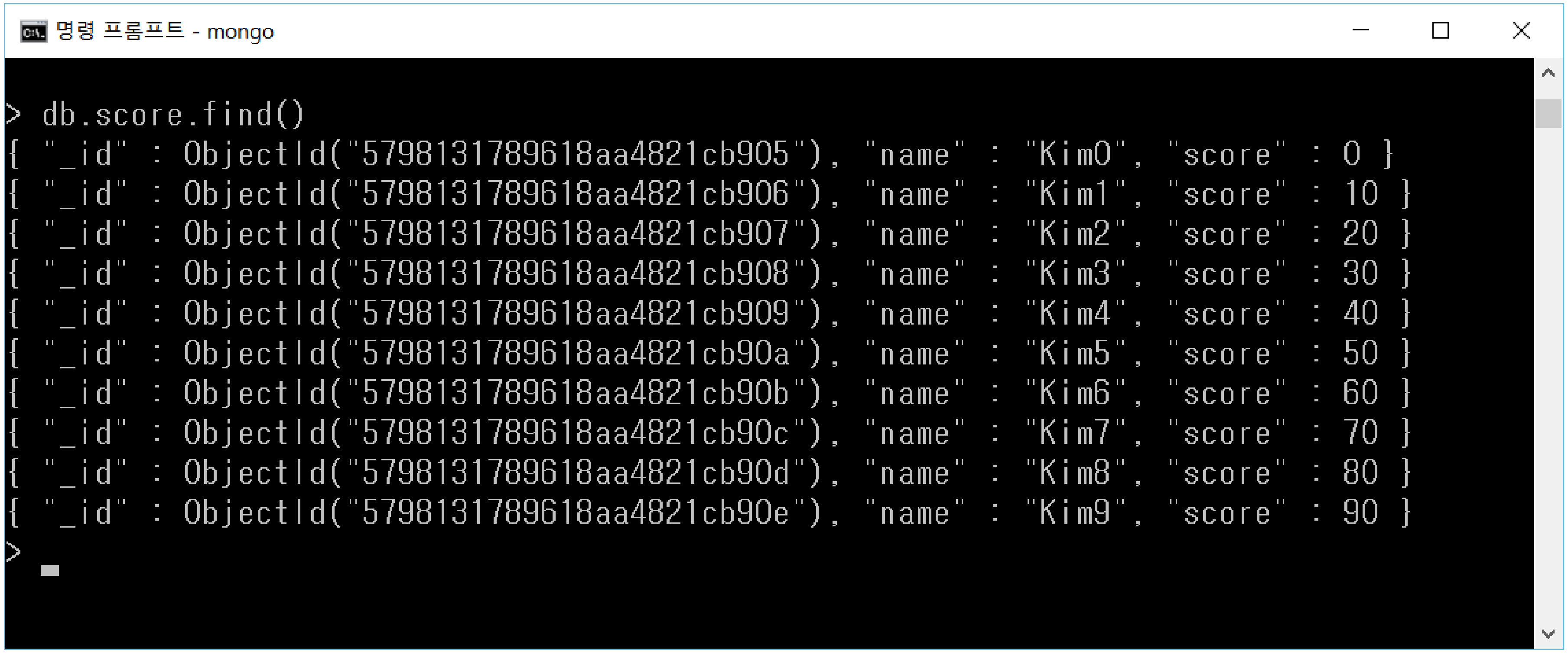
- db.컬렉션명.save([
    { 문서 } ..
  ]);


명령 프롬프트 - mongo

```
> db.fruit.save([
... {"name":"banana", "price":2500},
... {"name":"kiwi", "price":5000}
... ]);
BulkWriteResult({
        "writeErrors" : [ ],
        "writeConcernErrors" : [ ],
        "nInserted" : 2,
        "nUpserted" : 0,
        "nMatched" : 0,
        "nModified" : 0,
        "nRemoved" : 0,
        "upserted" : [ ]
})
> db.fruit.find();
{ "_id" : ObjectId("57db7b93fead6f035bed599b"), "price" : 2000 }
{ "_id" : ObjectId("57db7b9ffead6f035bed599c"), "name" : "orange", "price" : 3000 }
{ "_id" : ObjectId("57db7c64fead6f035bed599e"), "name" : "banana", "price" : 2500 }
{ "_id" : ObjectId("57db7c64fead6f035bed599f"), "name" : "kiwi", "price" : 5000 }
>
```

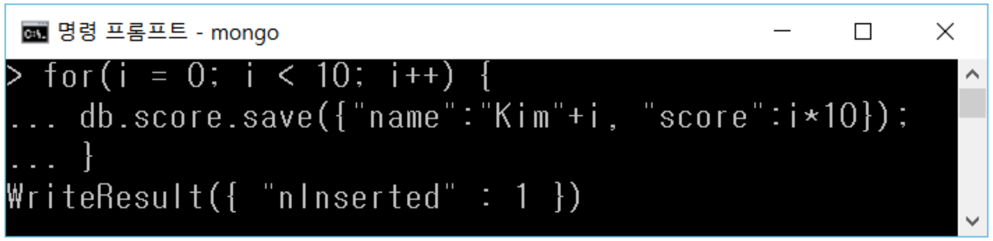
[실습] 데이터 다건 저장

- 아래 데이터가 저장되도록 데이터를 저장해보자!! (JavaScript 반복문 이용)



명령 프롬프트 - mongo

```
> db.score.find()
{ "_id" : ObjectId("5798131789618aa4821cb905"), "name" : "Kim0", "score" : 0 }
{ "_id" : ObjectId("5798131789618aa4821cb906"), "name" : "Kim1", "score" : 10 }
{ "_id" : ObjectId("5798131789618aa4821cb907"), "name" : "Kim2", "score" : 20 }
{ "_id" : ObjectId("5798131789618aa4821cb908"), "name" : "Kim3", "score" : 30 }
{ "_id" : ObjectId("5798131789618aa4821cb909"), "name" : "Kim4", "score" : 40 }
{ "_id" : ObjectId("5798131789618aa4821cb90a"), "name" : "Kim5", "score" : 50 }
{ "_id" : ObjectId("5798131789618aa4821cb90b"), "name" : "Kim6", "score" : 60 }
{ "_id" : ObjectId("5798131789618aa4821cb90c"), "name" : "Kim7", "score" : 70 }
{ "_id" : ObjectId("5798131789618aa4821cb90d"), "name" : "Kim8", "score" : 80 }
{ "_id" : ObjectId("5798131789618aa4821cb90e"), "name" : "Kim9", "score" : 90 }
>
```

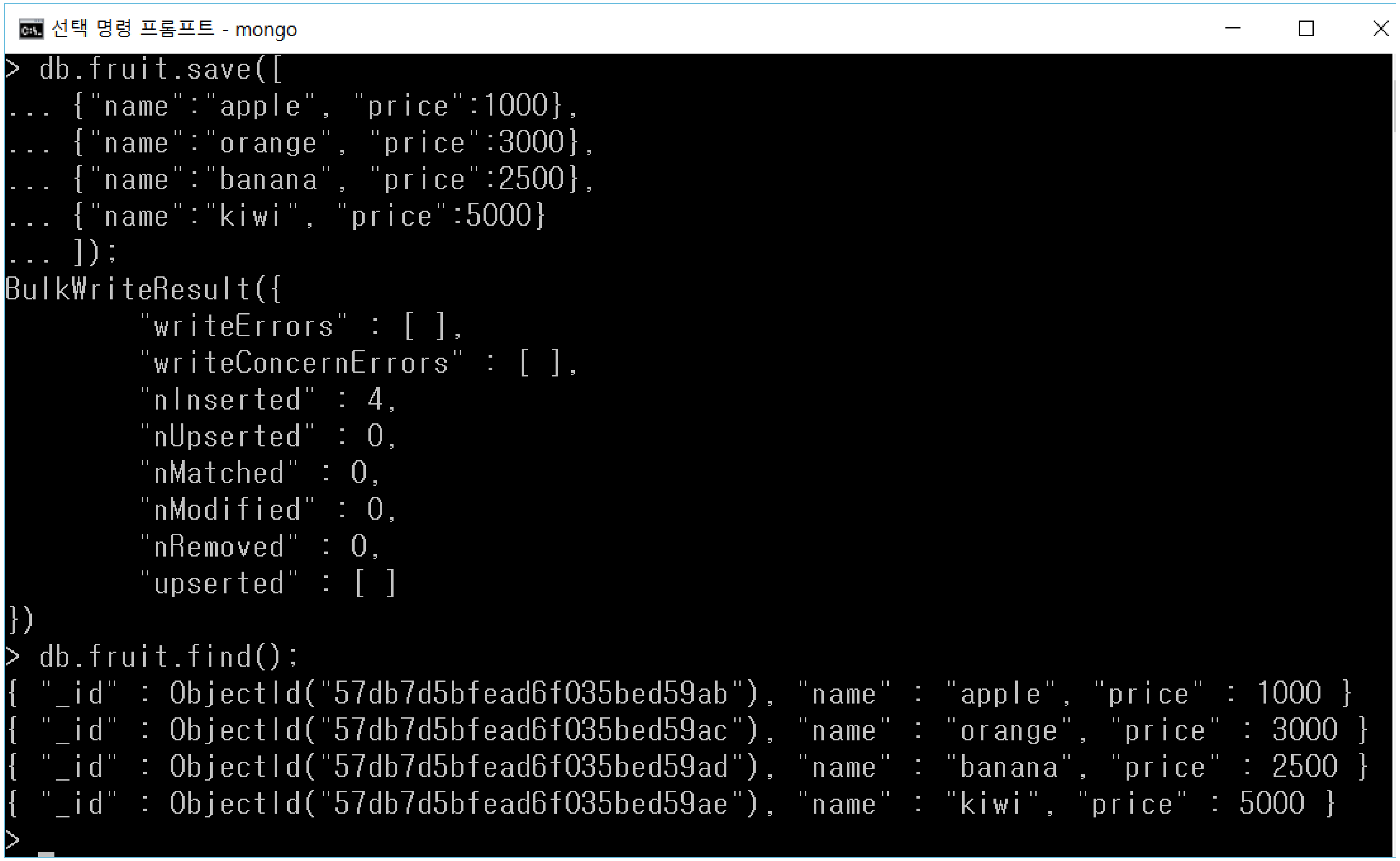


명령 프롬프트 - mongo

```
> for(i = 0; i < 10; i++) {
... db.score.save({"name":"Kim"+i, "score":i*10});
... }
WriteResult({ "nInserted" : 1 })
```

데이터 검색 (전체)

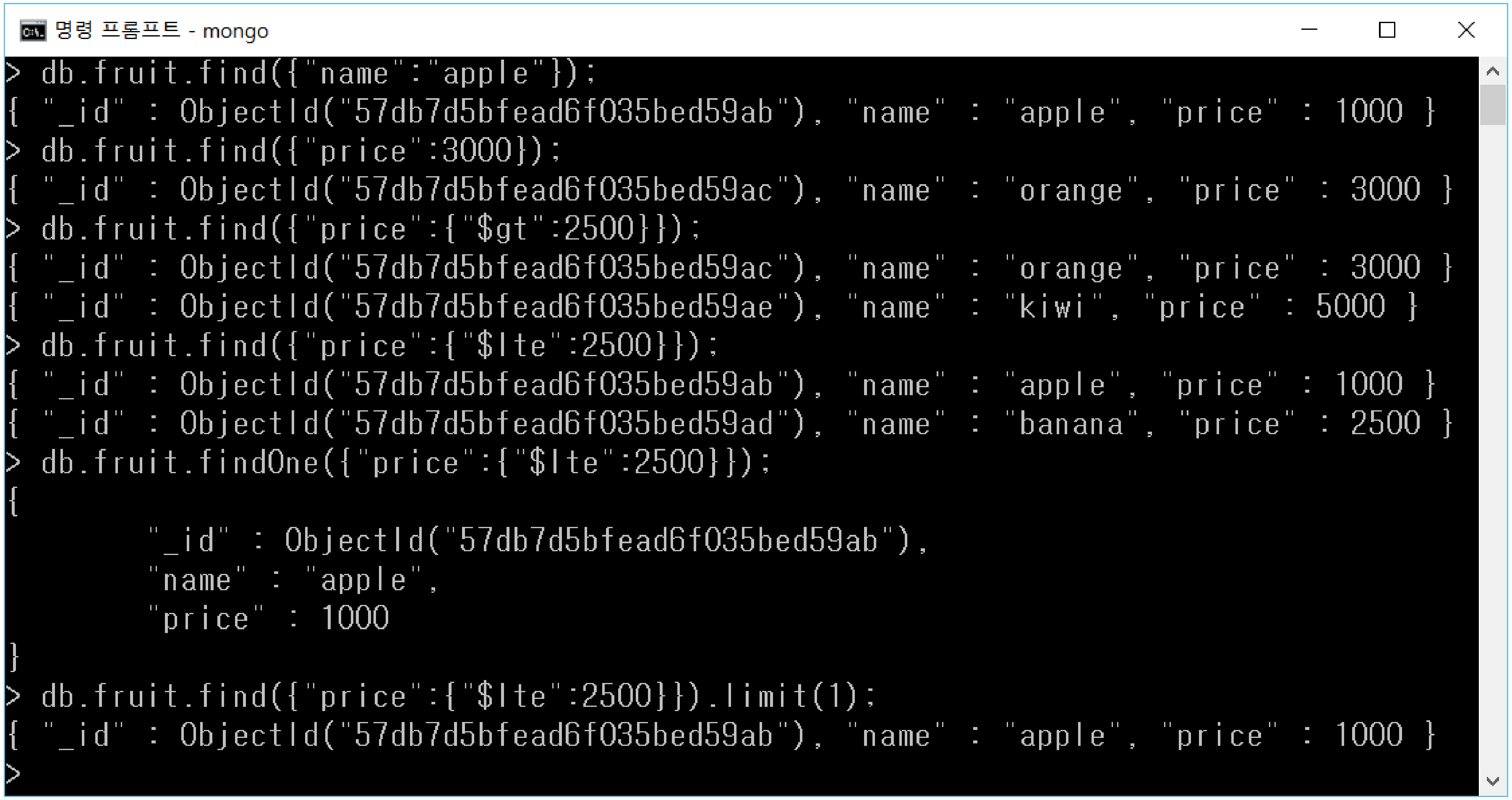
- db.컬렉션명.find( )



```
> db.fruit.save([
... {"name":"apple", "price":1000},
... {"name":"orange", "price":3000},
... {"name":"banana", "price":2500},
... {"name":"kiwi", "price":5000}
... ]);
BulkWriteResult({
        "writeErrors" : [ ],
        "writeConcernErrors" : [ ],
        "nInserted" : 4,
        "nUpserted" : 0,
        "nMatched" : 0,
        "nModified" : 0,
        "nRemoved" : 0,
        "upserted" : [ ]
})
> db.fruit.find();
{ "_id" : ObjectId("57db7d5bfead6f035bed59ab"), "name" : "apple", "price" : 1000 }
{ "_id" : ObjectId("57db7d5bfead6f035bed59ac"), "name" : "orange", "price" : 3000 }
{ "_id" : ObjectId("57db7d5bfead6f035bed59ad"), "name" : "banana", "price" : 2500 }
{ "_id" : ObjectId("57db7d5bfead6f035bed59ae"), "name" : "kiwi", "price" : 5000 }
> _
```

데이터 검색 (조건)

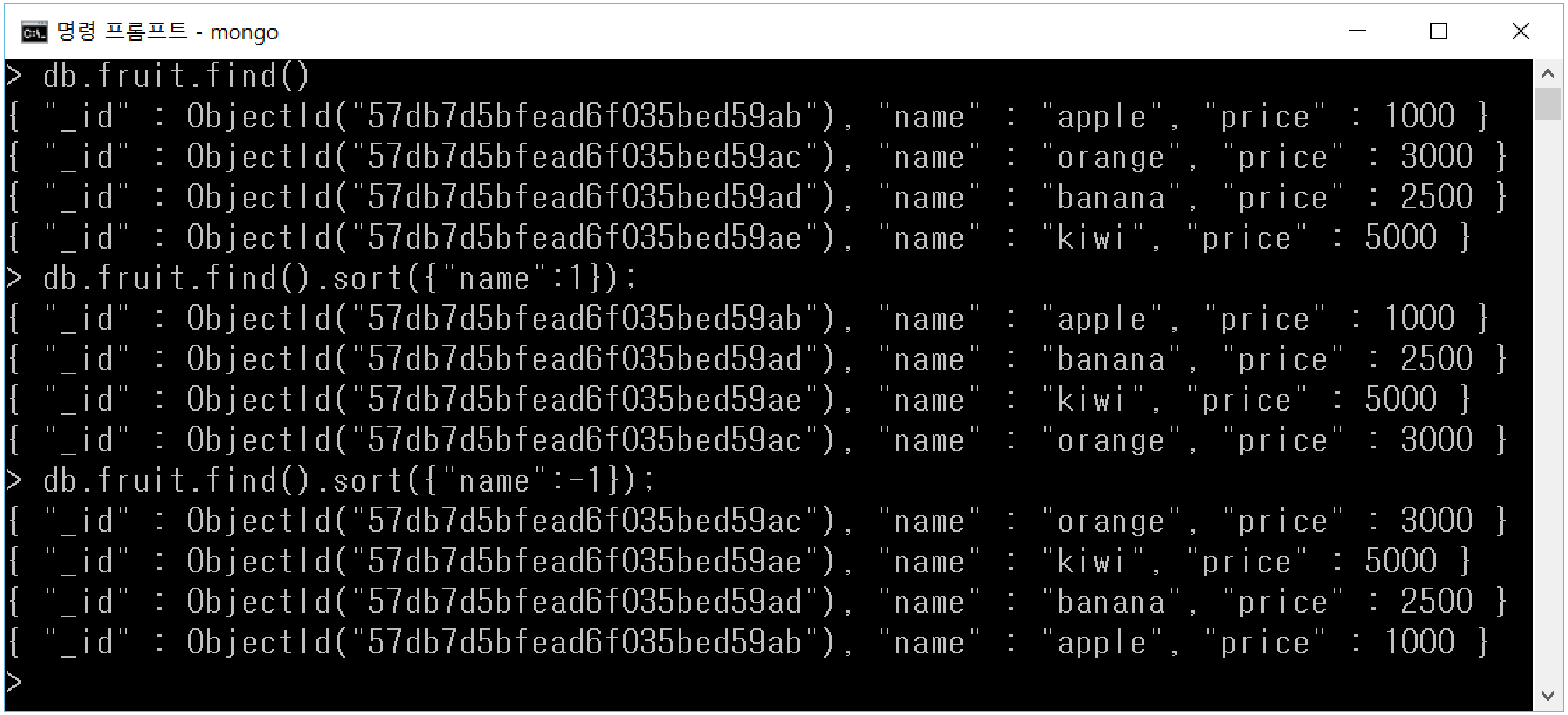
- db.컬렉션명.find({ 검색조건 })


명령 프롬프트 - mongo

```
> db.fruit.find({"name":"apple"});
{ "_id" : ObjectId("57db7d5bfead6f035bed59ab"), "name" : "apple", "price" : 1000 }
> db.fruit.find({"price":3000});
{ "_id" : ObjectId("57db7d5bfead6f035bed59ac"), "name" : "orange", "price" : 3000 }
> db.fruit.find({"price":{"$gt":2500}});
{ "_id" : ObjectId("57db7d5bfead6f035bed59ac"), "name" : "orange", "price" : 3000 }
{ "_id" : ObjectId("57db7d5bfead6f035bed59ae"), "name" : "kiwi", "price" : 5000 }
> db.fruit.find({"price":{"$lte":2500}});
{ "_id" : ObjectId("57db7d5bfead6f035bed59ab"), "name" : "apple", "price" : 1000 }
{ "_id" : ObjectId("57db7d5bfead6f035bed59ad"), "name" : "banana", "price" : 2500 }
> db.fruit.findOne({"price":{"$lte":2500}});
{
        "_id" : ObjectId("57db7d5bfead6f035bed59ab"),
        "name" : "apple",
        "price" : 1000
}
> db.fruit.find({"price":{"$lte":2500}}).limit(1);
{ "_id" : ObjectId("57db7d5bfead6f035bed59ab"), "name" : "apple", "price" : 1000 }
>
```

데이터 검색 (정렬)

- db.컬렉션명.find( ).sort({ 필드 : 1|-1 })

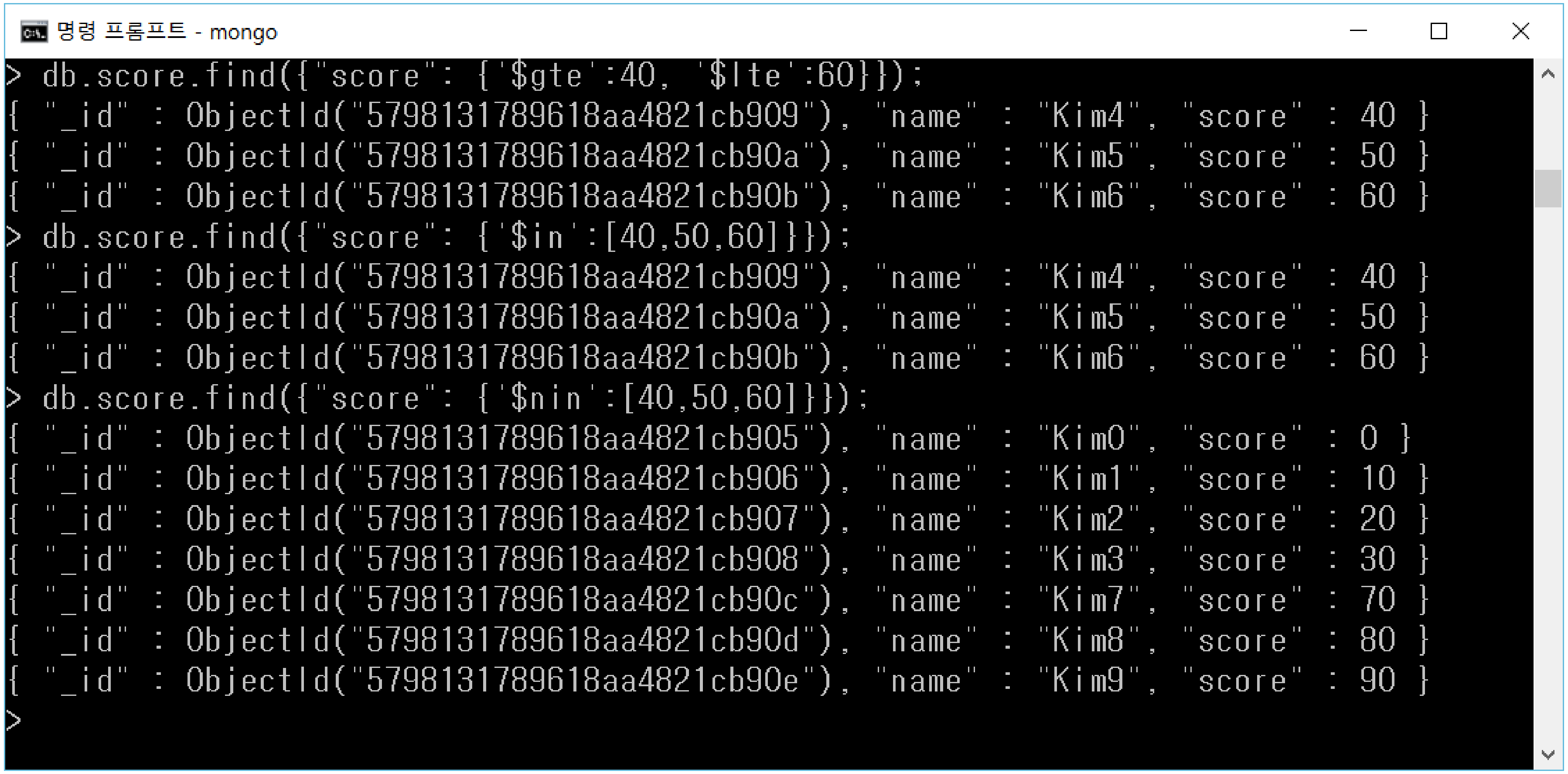


명령 프롬프트 - mongo

```
> db.fruit.find()
{ "_id" : ObjectId("57db7d5bfead6f035bed59ab"), "name" : "apple", "price" : 1000 }
{ "_id" : ObjectId("57db7d5bfead6f035bed59ac"), "name" : "orange", "price" : 3000 }
{ "_id" : ObjectId("57db7d5bfead6f035bed59ad"), "name" : "banana", "price" : 2500 }
{ "_id" : ObjectId("57db7d5bfead6f035bed59ae"), "name" : "kiwi", "price" : 5000 }
> db.fruit.find().sort({"name":1});
{ "_id" : ObjectId("57db7d5bfead6f035bed59ab"), "name" : "apple", "price" : 1000 }
{ "_id" : ObjectId("57db7d5bfead6f035bed59ad"), "name" : "banana", "price" : 2500 }
{ "_id" : ObjectId("57db7d5bfead6f035bed59ae"), "name" : "kiwi", "price" : 5000 }
{ "_id" : ObjectId("57db7d5bfead6f035bed59ac"), "name" : "orange", "price" : 3000 }
> db.fruit.find().sort({"name":-1});
{ "_id" : ObjectId("57db7d5bfead6f035bed59ac"), "name" : "orange", "price" : 3000 }
{ "_id" : ObjectId("57db7d5bfead6f035bed59ae"), "name" : "kiwi", "price" : 5000 }
{ "_id" : ObjectId("57db7d5bfead6f035bed59ad"), "name" : "banana", "price" : 2500 }
{ "_id" : ObjectId("57db7d5bfead6f035bed59ab"), "name" : "apple", "price" : 1000 }
>
```

데이터 검색 (AND 연산)

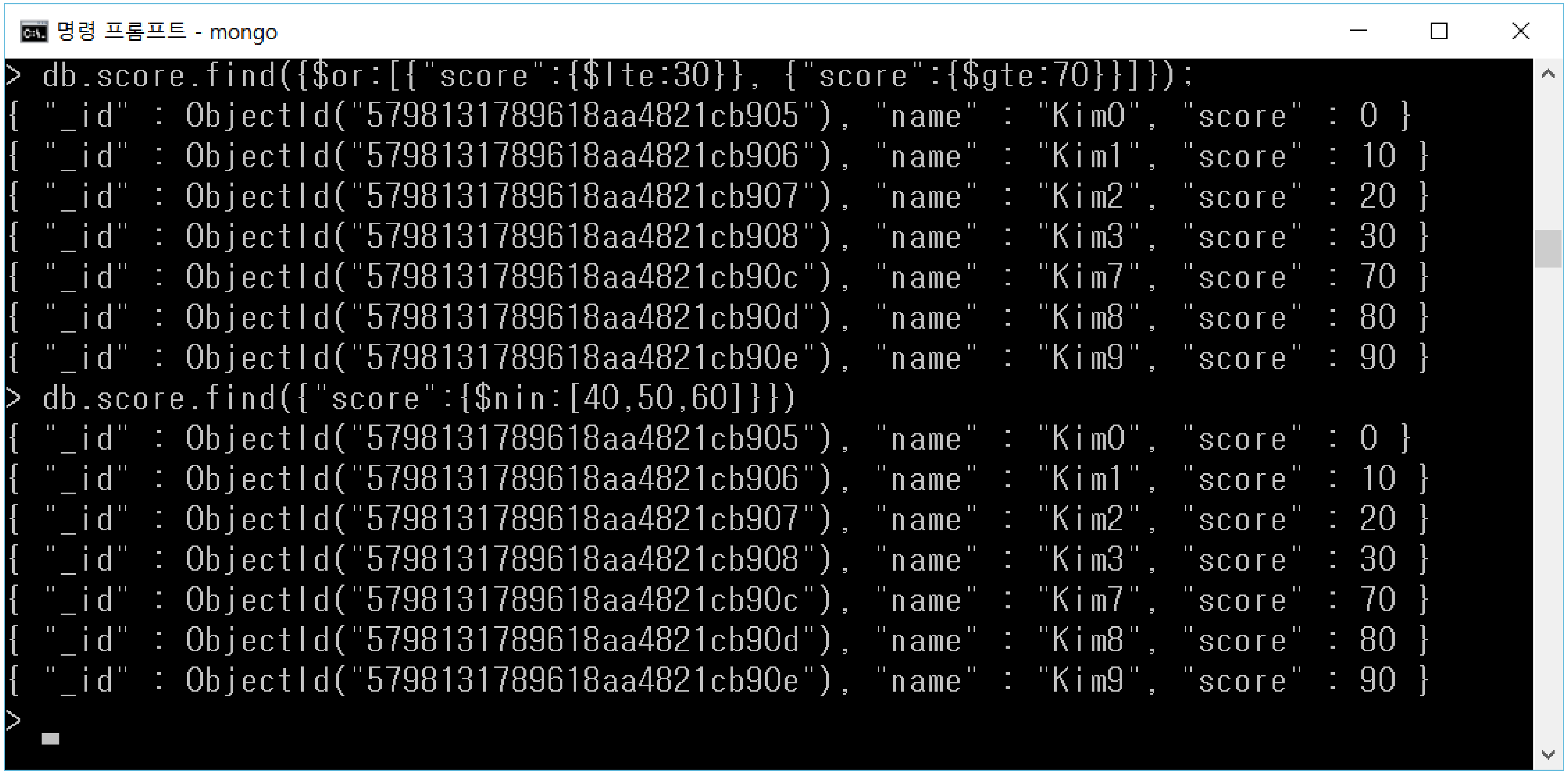
- 비교 연산자 - $gt : '>', $lt : '<', $gte : '>=', $lte : '<=', $ne : '!=‘
  $in : 'is in array', $nin : '! in array'



```
> db.score.find({"score": {'$gte':40, '$lte':60}});
{ "_id" : ObjectId("5798131789618aa4821cb909"), "name" : "Kim4", "score" : 40 }
{ "_id" : ObjectId("5798131789618aa4821cb90a"), "name" : "Kim5", "score" : 50 }
{ "_id" : ObjectId("5798131789618aa4821cb90b"), "name" : "Kim6", "score" : 60 }
> db.score.find({"score": {'$in':[40,50,60]}});
{ "_id" : ObjectId("5798131789618aa4821cb909"), "name" : "Kim4", "score" : 40 }
{ "_id" : ObjectId("5798131789618aa4821cb90a"), "name" : "Kim5", "score" : 50 }
{ "_id" : ObjectId("5798131789618aa4821cb90b"), "name" : "Kim6", "score" : 60 }
> db.score.find({"score": {'$nin':[40,50,60]}});
{ "_id" : ObjectId("5798131789618aa4821cb905"), "name" : "Kim0", "score" : 0 }
{ "_id" : ObjectId("5798131789618aa4821cb906"), "name" : "Kim1", "score" : 10 }
{ "_id" : ObjectId("5798131789618aa4821cb907"), "name" : "Kim2", "score" : 20 }
{ "_id" : ObjectId("5798131789618aa4821cb908"), "name" : "Kim3", "score" : 30 }
{ "_id" : ObjectId("5798131789618aa4821cb90c"), "name" : "Kim7", "score" : 70 }
{ "_id" : ObjectId("5798131789618aa4821cb90d"), "name" : "Kim8", "score" : 80 }
{ "_id" : ObjectId("5798131789618aa4821cb90e"), "name" : "Kim9", "score" : 90 }
>
```

데이터 검색 (OR 연산)

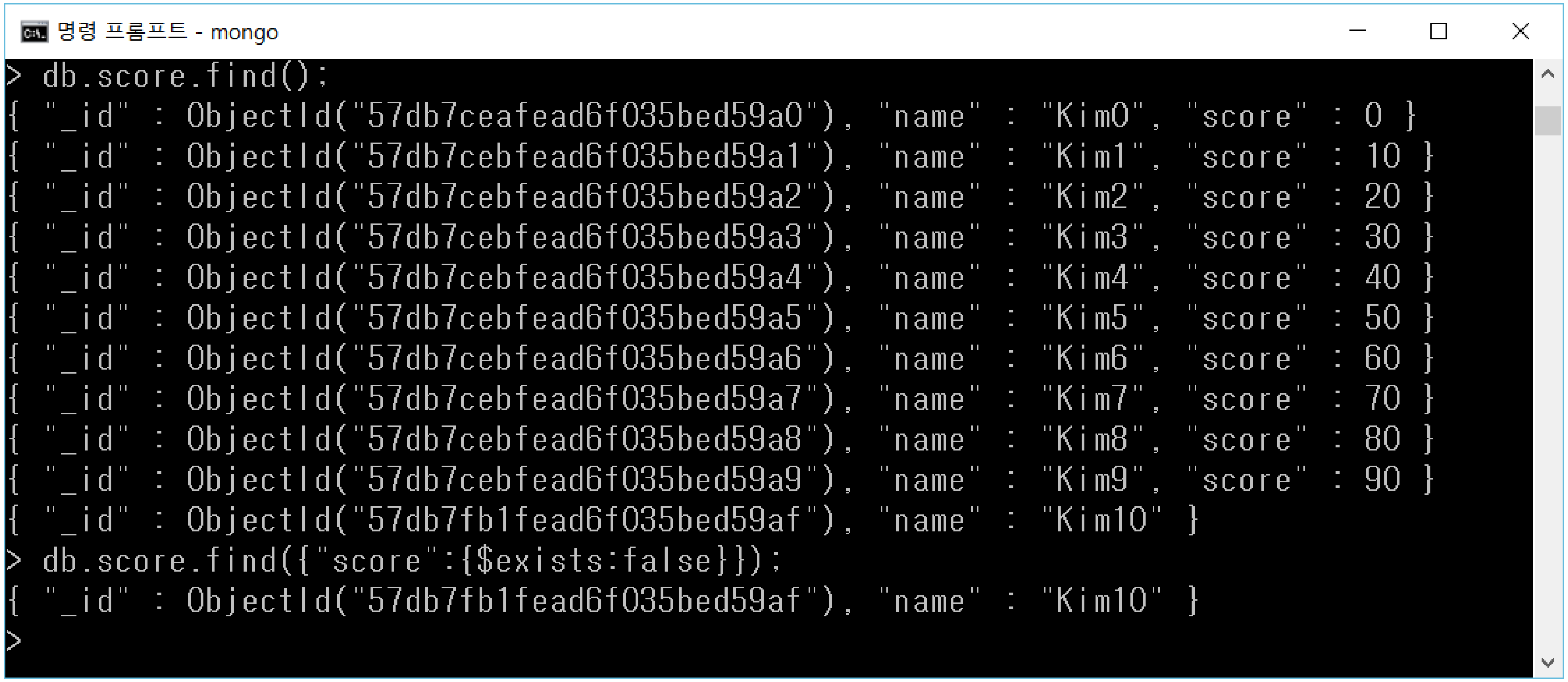
- 비교 연산자 - $gt : '>', $lt : '<', $gte : '>=', $lte : '<=', $ne : '!='
  $in : 'is in array', $nin : '! in array'



```
> db.score.find({$or:[{"score":{$lte:30}}, {"score":{$gte:70}}]});
{ "_id" : ObjectId("5798131789618aa4821cb905"), "name" : "Kim0", "score" : 0 }
{ "_id" : ObjectId("5798131789618aa4821cb906"), "name" : "Kim1", "score" : 10 }
{ "_id" : ObjectId("5798131789618aa4821cb907"), "name" : "Kim2", "score" : 20 }
{ "_id" : ObjectId("5798131789618aa4821cb908"), "name" : "Kim3", "score" : 30 }
{ "_id" : ObjectId("5798131789618aa4821cb90c"), "name" : "Kim7", "score" : 70 }
{ "_id" : ObjectId("5798131789618aa4821cb90d"), "name" : "Kim8", "score" : 80 }
{ "_id" : ObjectId("5798131789618aa4821cb90e"), "name" : "Kim9", "score" : 90 }
> db.score.find({"score":{$nin:[40,50,60]}})
{ "_id" : ObjectId("5798131789618aa4821cb905"), "name" : "Kim0", "score" : 0 }
{ "_id" : ObjectId("5798131789618aa4821cb906"), "name" : "Kim1", "score" : 10 }
{ "_id" : ObjectId("5798131789618aa4821cb907"), "name" : "Kim2", "score" : 20 }
{ "_id" : ObjectId("5798131789618aa4821cb908"), "name" : "Kim3", "score" : 30 }
{ "_id" : ObjectId("5798131789618aa4821cb90c"), "name" : "Kim7", "score" : 70 }
{ "_id" : ObjectId("5798131789618aa4821cb90d"), "name" : "Kim8", "score" : 80 }
{ "_id" : ObjectId("5798131789618aa4821cb90e"), "name" : "Kim9", "score" : 90 }
>
```

데이터 검색 (필드 존재 여부)

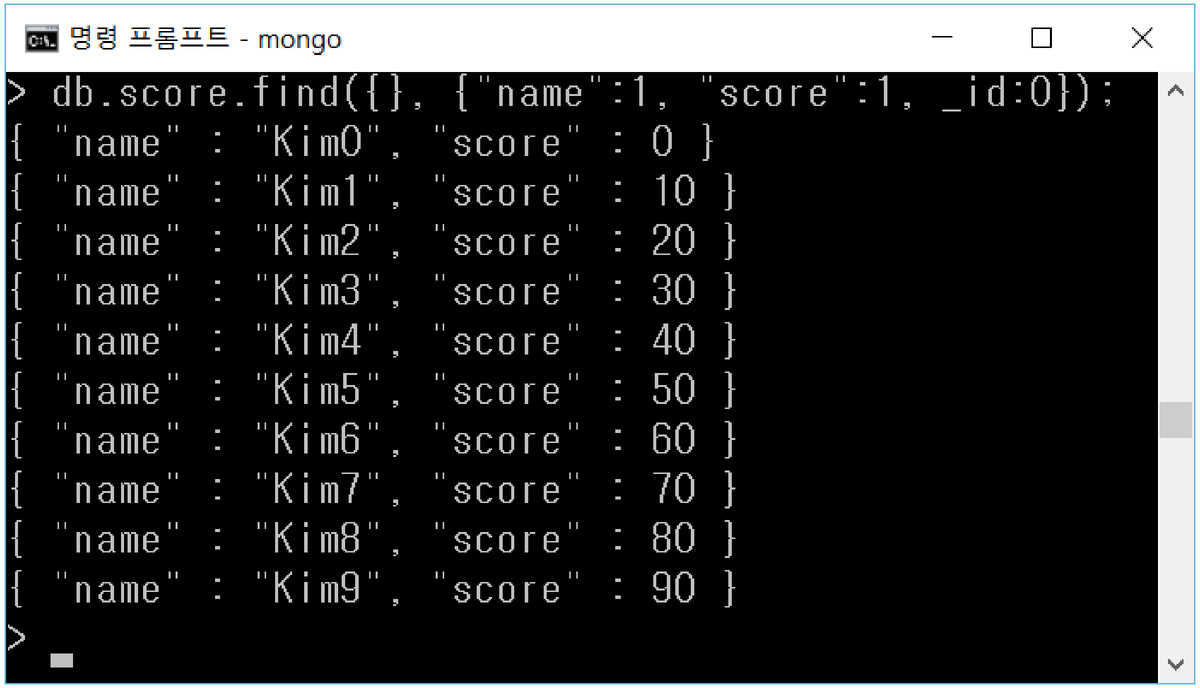
- $exists : true / false



```
> db.score.find();
{ "_id" : ObjectId("57db7ceafead6f035bed59a0"), "name" : "Kim0", "score" : 0 }
{ "_id" : ObjectId("57db7cebfead6f035bed59a1"), "name" : "Kim1", "score" : 10 }
{ "_id" : ObjectId("57db7cebfead6f035bed59a2"), "name" : "Kim2", "score" : 20 }
{ "_id" : ObjectId("57db7cebfead6f035bed59a3"), "name" : "Kim3", "score" : 30 }
{ "_id" : ObjectId("57db7cebfead6f035bed59a4"), "name" : "Kim4", "score" : 40 }
{ "_id" : ObjectId("57db7cebfead6f035bed59a5"), "name" : "Kim5", "score" : 50 }
{ "_id" : ObjectId("57db7cebfead6f035bed59a6"), "name" : "Kim6", "score" : 60 }
{ "_id" : ObjectId("57db7cebfead6f035bed59a7"), "name" : "Kim7", "score" : 70 }
{ "_id" : ObjectId("57db7cebfead6f035bed59a8"), "name" : "Kim8", "score" : 80 }
{ "_id" : ObjectId("57db7cebfead6f035bed59a9"), "name" : "Kim9", "score" : 90 }
{ "_id" : ObjectId("57db7fb1fead6f035bed59af"), "name" : "Kim10" }
> db.score.find({"score":{$exists:false}});
{ "_id" : ObjectId("57db7fb1fead6f035bed59af"), "name" : "Kim10" }
>
```

데이터 검색 (검색 필드 선택)

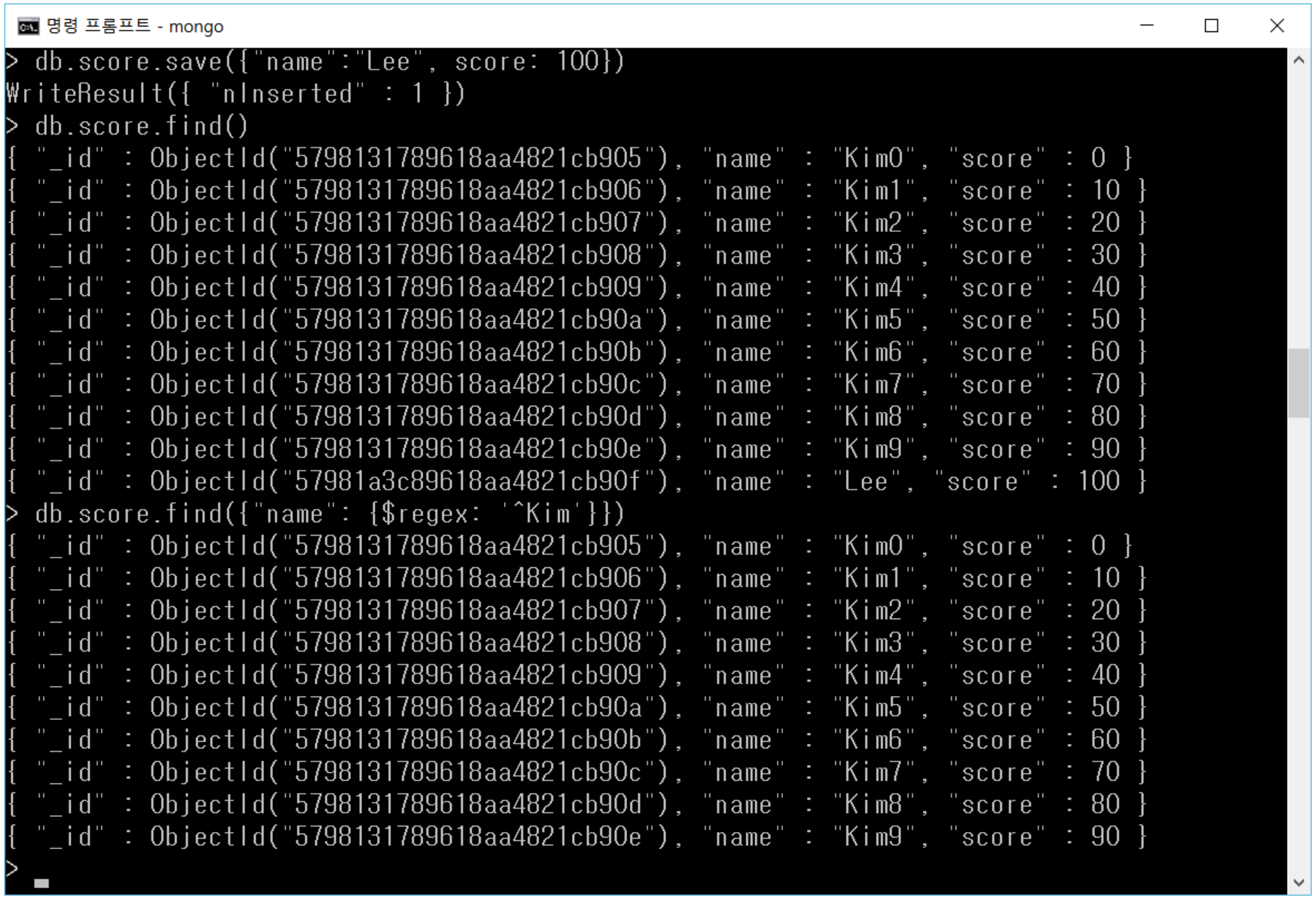
- 1: 출력, 0: 출력 안함



```
> db.score.find({}, {"name":1, "score":1, _id:0});
{ "name" : "Kim0", "score" : 0 }
{ "name" : "Kim1", "score" : 10 }
{ "name" : "Kim2", "score" : 20 }
{ "name" : "Kim3", "score" : 30 }
{ "name" : "Kim4", "score" : 40 }
{ "name" : "Kim5", "score" : 50 }
{ "name" : "Kim6", "score" : 60 }
{ "name" : "Kim7", "score" : 70 }
{ "name" : "Kim8", "score" : 80 }
{ "name" : "Kim9", "score" : 90 }
>
```

데이터 검색 (정규식 표현 1)

- $regex

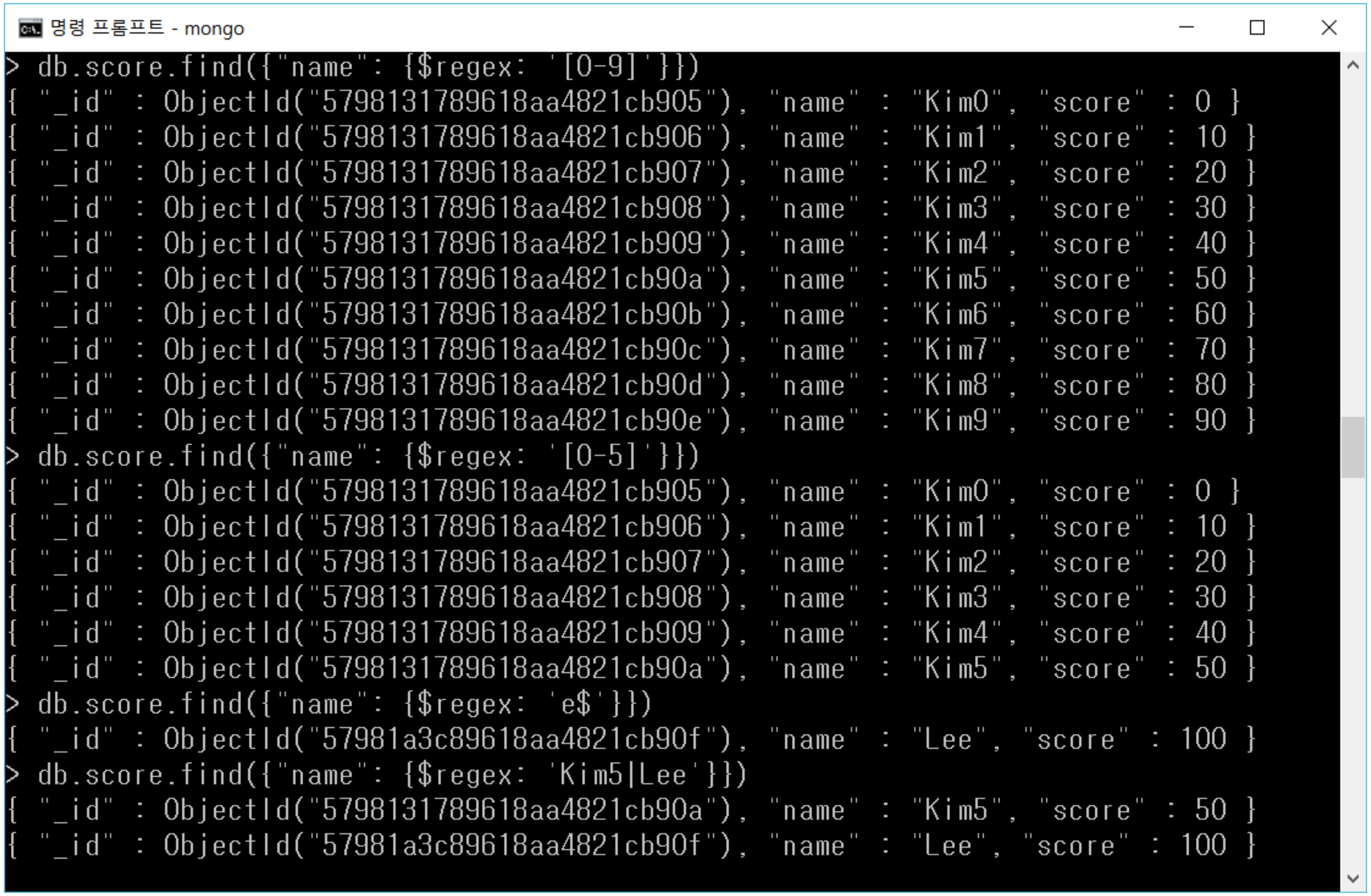


```
> db.score.save({"name":"Lee", score: 100})
WriteResult({ "nInserted" : 1 })
> db.score.find()
{ "_id" : ObjectId("5798131789618aa4821cb905"), "name" : "Kim0", "score" : 0 }
{ "_id" : ObjectId("5798131789618aa4821cb906"), "name" : "Kim1", "score" : 10 }
{ "_id" : ObjectId("5798131789618aa4821cb907"), "name" : "Kim2", "score" : 20 }
{ "_id" : ObjectId("5798131789618aa4821cb908"), "name" : "Kim3", "score" : 30 }
{ "_id" : ObjectId("5798131789618aa4821cb909"), "name" : "Kim4", "score" : 40 }
{ "_id" : ObjectId("5798131789618aa4821cb90a"), "name" : "Kim5", "score" : 50 }
{ "_id" : ObjectId("5798131789618aa4821cb90b"), "name" : "Kim6", "score" : 60 }
{ "_id" : ObjectId("5798131789618aa4821cb90c"), "name" : "Kim7", "score" : 70 }
{ "_id" : ObjectId("5798131789618aa4821cb90d"), "name" : "Kim8", "score" : 80 }
{ "_id" : ObjectId("5798131789618aa4821cb90e"), "name" : "Kim9", "score" : 90 }
{ "_id" : ObjectId("57981a3c89618aa4821cb90f"), "name" : "Lee", "score" : 100 }
> db.score.find({"name": {$regex: '^Kim'}})
{ "_id" : ObjectId("5798131789618aa4821cb905"), "name" : "Kim0", "score" : 0 }
{ "_id" : ObjectId("5798131789618aa4821cb906"), "name" : "Kim1", "score" : 10 }
{ "_id" : ObjectId("5798131789618aa4821cb907"), "name" : "Kim2", "score" : 20 }
{ "_id" : ObjectId("5798131789618aa4821cb908"), "name" : "Kim3", "score" : 30 }
{ "_id" : ObjectId("5798131789618aa4821cb909"), "name" : "Kim4", "score" : 40 }
{ "_id" : ObjectId("5798131789618aa4821cb90a"), "name" : "Kim5", "score" : 50 }
{ "_id" : ObjectId("5798131789618aa4821cb90b"), "name" : "Kim6", "score" : 60 }
{ "_id" : ObjectId("5798131789618aa4821cb90c"), "name" : "Kim7", "score" : 70 }
{ "_id" : ObjectId("5798131789618aa4821cb90d"), "name" : "Kim8", "score" : 80 }
{ "_id" : ObjectId("5798131789618aa4821cb90e"), "name" : "Kim9", "score" : 90 }
>
```

http://www.w3schools.com/js/js_regexp.asp
https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/RegExp

데이터 검색 (정규식 표현 2)

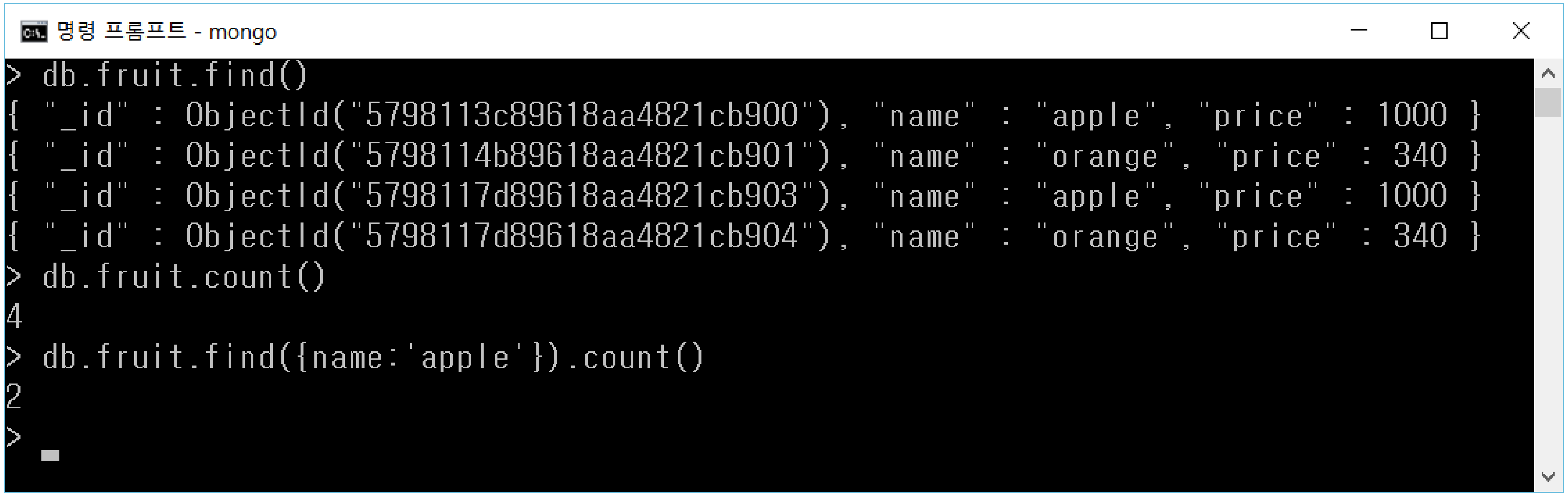
- $regex


명령 프롬프트 - mongo

```
> db.score.find({"name": {$regex: '[0-9]'}})
{ "_id" : ObjectId("5798131789618aa4821cb905"), "name" : "Kim0", "score" : 0 }
{ "_id" : ObjectId("5798131789618aa4821cb906"), "name" : "Kim1", "score" : 10 }
{ "_id" : ObjectId("5798131789618aa4821cb907"), "name" : "Kim2", "score" : 20 }
{ "_id" : ObjectId("5798131789618aa4821cb908"), "name" : "Kim3", "score" : 30 }
{ "_id" : ObjectId("5798131789618aa4821cb909"), "name" : "Kim4", "score" : 40 }
{ "_id" : ObjectId("5798131789618aa4821cb90a"), "name" : "Kim5", "score" : 50 }
{ "_id" : ObjectId("5798131789618aa4821cb90b"), "name" : "Kim6", "score" : 60 }
{ "_id" : ObjectId("5798131789618aa4821cb90c"), "name" : "Kim7", "score" : 70 }
{ "_id" : ObjectId("5798131789618aa4821cb90d"), "name" : "Kim8", "score" : 80 }
{ "_id" : ObjectId("5798131789618aa4821cb90e"), "name" : "Kim9", "score" : 90 }
> db.score.find({"name": {$regex: '[0-5]'}})
{ "_id" : ObjectId("5798131789618aa4821cb905"), "name" : "Kim0", "score" : 0 }
{ "_id" : ObjectId("5798131789618aa4821cb906"), "name" : "Kim1", "score" : 10 }
{ "_id" : ObjectId("5798131789618aa4821cb907"), "name" : "Kim2", "score" : 20 }
{ "_id" : ObjectId("5798131789618aa4821cb908"), "name" : "Kim3", "score" : 30 }
{ "_id" : ObjectId("5798131789618aa4821cb909"), "name" : "Kim4", "score" : 40 }
{ "_id" : ObjectId("5798131789618aa4821cb90a"), "name" : "Kim5", "score" : 50 }
> db.score.find({"name": {$regex: 'e$'}})
{ "_id" : ObjectId("57981a3c89618aa4821cb90f"), "name" : "Lee", "score" : 100 }
> db.score.find({"name": {$regex: 'Kim5|Lee'}})
{ "_id" : ObjectId("5798131789618aa4821cb90a"), "name" : "Kim5", "score" : 50 }
{ "_id" : ObjectId("57981a3c89618aa4821cb90f"), "name" : "Lee", "score" : 100 }
```

데이터 집계 함수

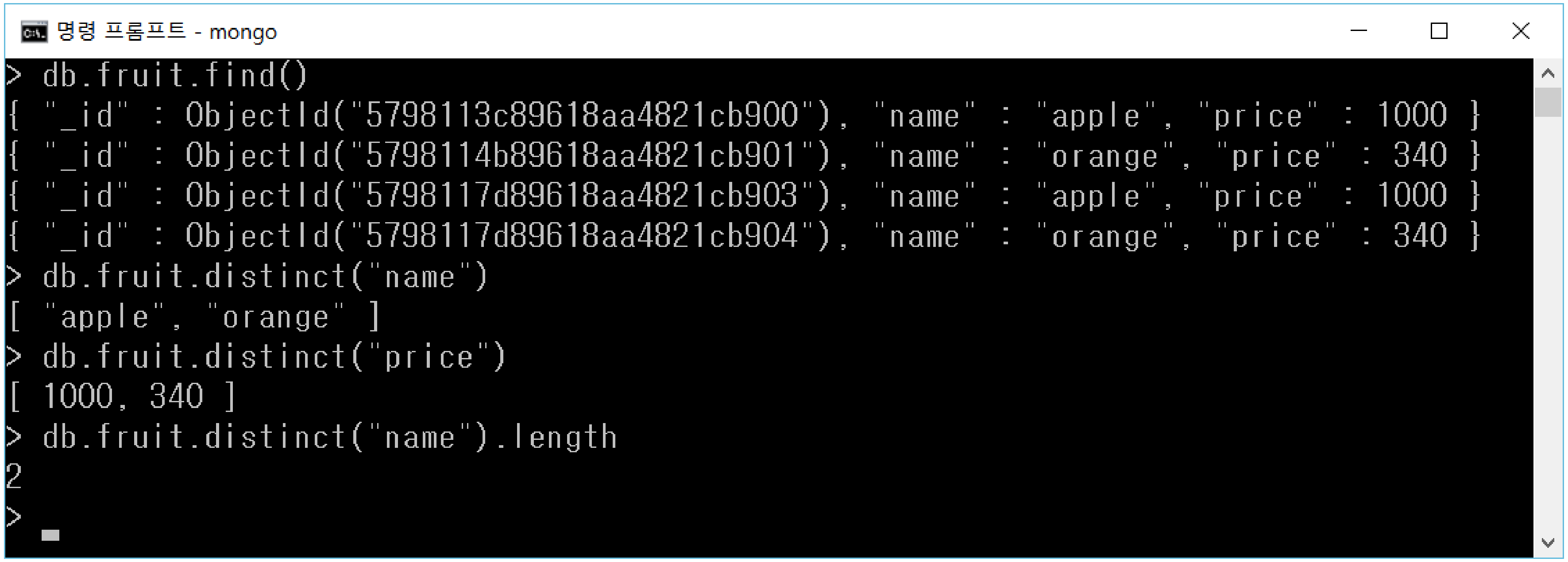
- count : 컬렉션 내 문서의 개수를 조회



```
> db.fruit.find()
{ "_id" : ObjectId("5798113c89618aa4821cb900"), "name" : "apple", "price" : 1000 }
{ "_id" : ObjectId("5798114b89618aa4821cb901"), "name" : "orange", "price" : 340 }
{ "_id" : ObjectId("5798117d89618aa4821cb903"), "name" : "apple", "price" : 1000 }
{ "_id" : ObjectId("5798117d89618aa4821cb904"), "name" : "orange", "price" : 340 }
> db.fruit.count()
4
> db.fruit.find({name:'apple'}).count()
2
>
```

데이터 집계 함수

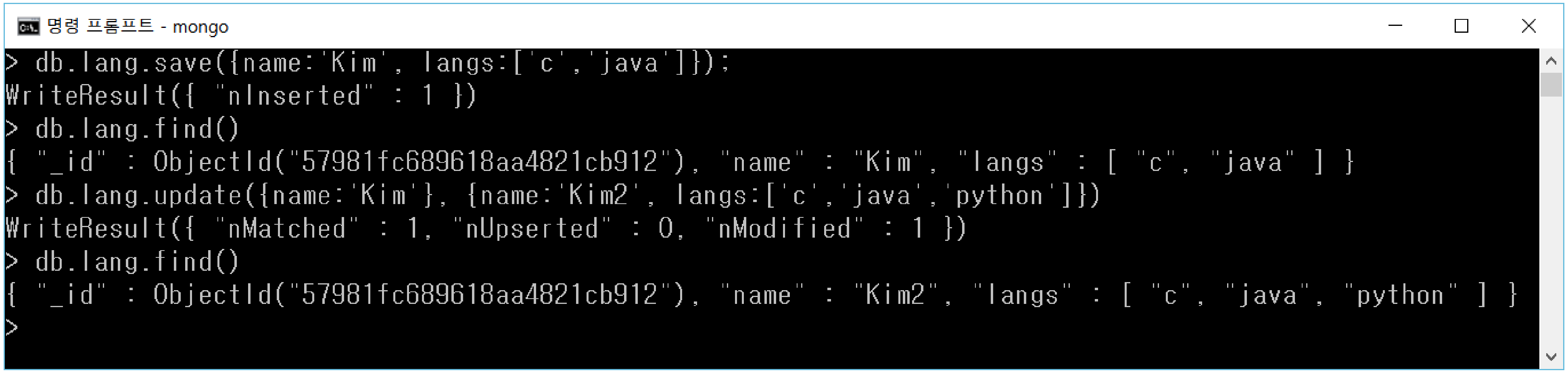
- distinct : 지정한 키에 대한 중복 값 제거



```
> db.fruit.find()
{ "_id" : ObjectId("5798113c89618aa4821cb900"), "name" : "apple", "price" : 1000 }
{ "_id" : ObjectId("5798114b89618aa4821cb901"), "name" : "orange", "price" : 340 }
{ "_id" : ObjectId("5798117d89618aa4821cb903"), "name" : "apple", "price" : 1000 }
{ "_id" : ObjectId("5798117d89618aa4821cb904"), "name" : "orange", "price" : 340 }
> db.fruit.distinct("name")
[ "apple", "orange" ]
> db.fruit.distinct("price")
[ 1000, 340 ]
> db.fruit.distinct("name").length
2
>
```

데이터 수정 (문서 전체)

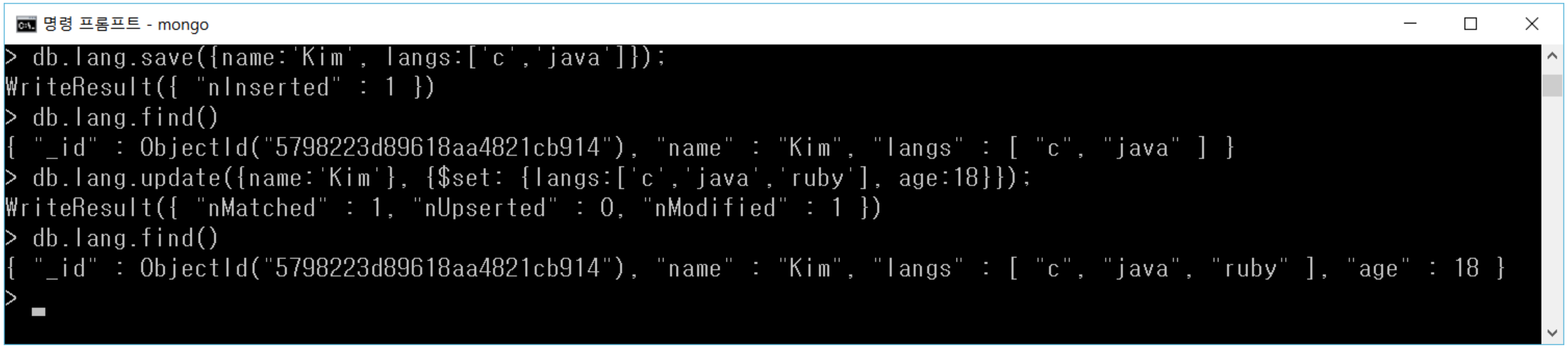
- db.컬렉션명.update({변경대상 문서}, {변경할 문서 전체})



```
> db.lang.save({name:'Kim', langs:['c','java']});
WriteResult({ "nInserted" : 1 })
> db.lang.find()
{ "_id" : ObjectId("57981fc689618aa4821cb912"), "name" : "Kim", "langs" : [ "c", "java" ] }
> db.lang.update({name:'Kim'}, {name:'Kim2', langs:['c','java','python']})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.lang.find()
{ "_id" : ObjectId("57981fc689618aa4821cb912"), "name" : "Kim2", "langs" : [ "c", "java", "python" ] }
>
```

데이터 수정 (필드 변경/추가)

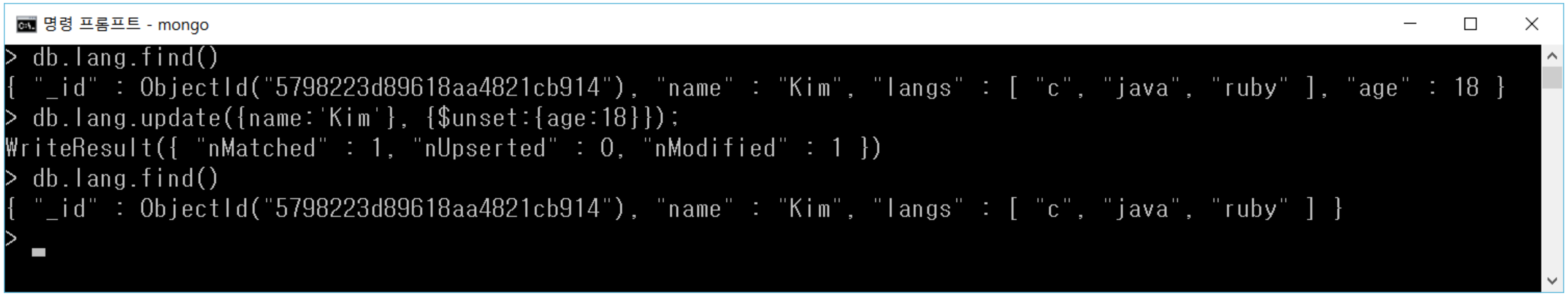
- db.컬렉션명.update({변경대상 문서}, {'$set' : {필드 변경/추가} })



```
> db.lang.save({name:'Kim', langs:['c','java']});
WriteResult({ "nInserted" : 1 })
> db.lang.find()
{ "_id" : ObjectId("5798223d89618aa4821cb914"), "name" : "Kim", "langs" : [ "c", "java" ] }
> db.lang.update({name:'Kim'}, {$set: {langs:['c','java','ruby'], age:18}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.lang.find()
{ "_id" : ObjectId("5798223d89618aa4821cb914"), "name" : "Kim", "langs" : [ "c", "java", "ruby" ], "age" : 18 }
>
```

데이터 수정 (필드 삭제)

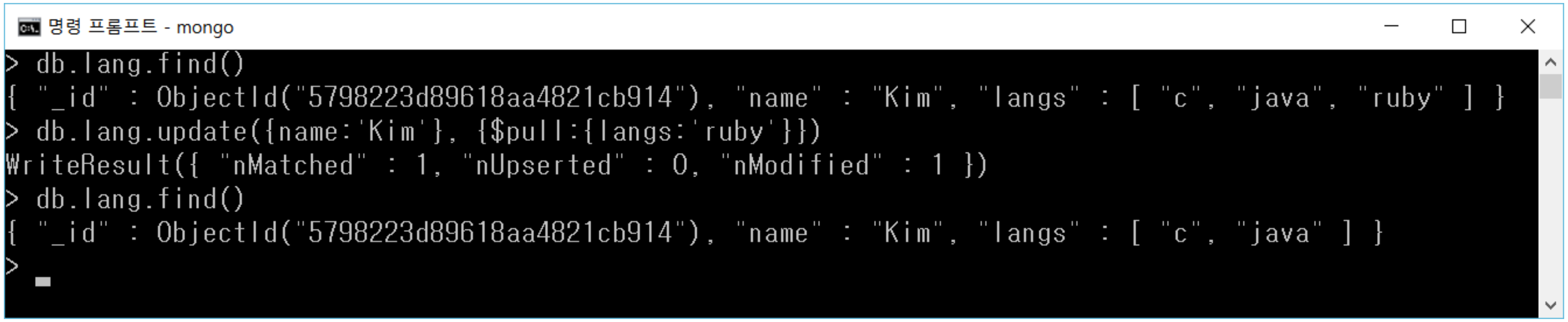
- db.컬렉션명.update({변경대상 문서}, {'$unset' : {필드 변경/추가} })



```
> db.lang.find()
{ "_id" : ObjectId("5798223d89618aa4821cb914"), "name" : "Kim", "langs" : [ "c", "java", "ruby" ], "age" : 18 }
> db.lang.update({name:'Kim'}, {$unset:{age:18}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.lang.find()
{ "_id" : ObjectId("5798223d89618aa4821cb914"), "name" : "Kim", "langs" : [ "c", "java", "ruby" ] }
>
```

데이터 수정 (배열값 삭제)

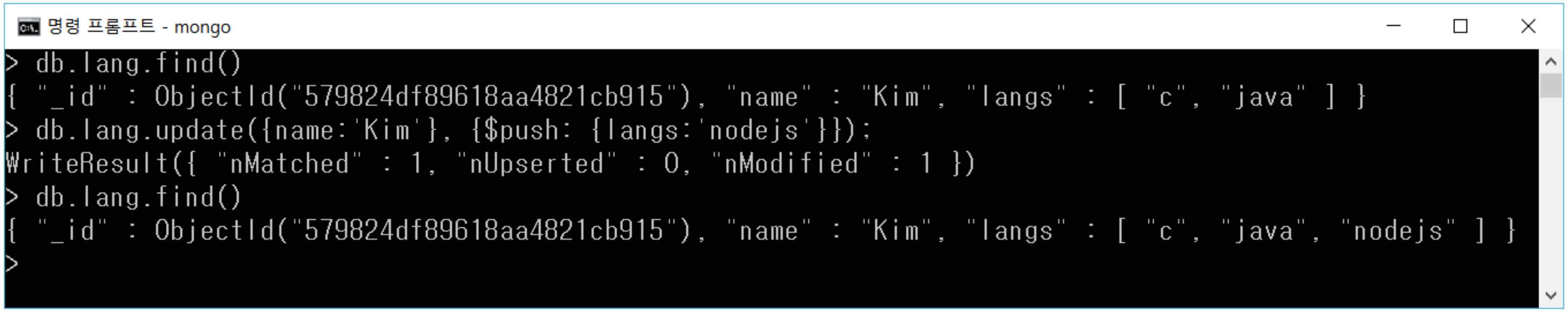
- db.컬렉션명.update({변경대상 문서}, {'$pull' : {key : value} })


명령 프롬프트 - mongo

```
> db.lang.find()
{ "_id" : ObjectId("5798223d89618aa4821cb914"), "name" : "Kim", "langs" : [ "c", "java", "ruby" ] }
> db.lang.update({name:'Kim'}, {$pull:{langs:'ruby'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.lang.find()
{ "_id" : ObjectId("5798223d89618aa4821cb914"), "name" : "Kim", "langs" : [ "c", "java" ] }
>
```

데이터 수정 (배열값 추가)

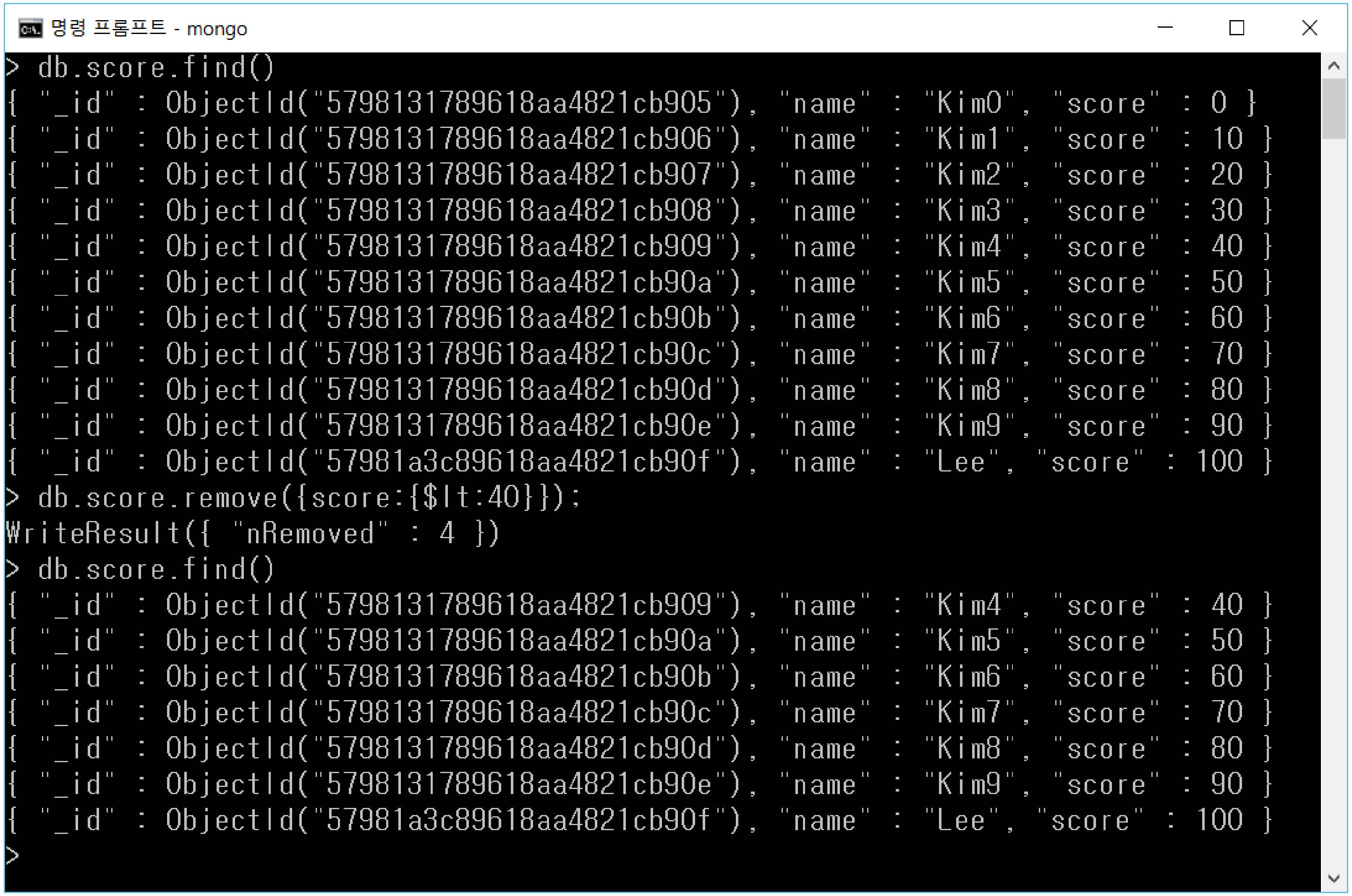
- db.컬렉션명.update({변경대상 문서}, {'$push' : {key : value} })



```
> db.lang.find()
{ "_id" : ObjectId("579824df89618aa4821cb915"), "name" : "Kim", "langs" : [ "c", "java" ] }
> db.lang.update({name:'Kim'}, {$push: {langs:'nodejs'}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.lang.find()
{ "_id" : ObjectId("579824df89618aa4821cb915"), "name" : "Kim", "langs" : [ "c", "java", "nodejs" ] }
>
```

데이터 삭제

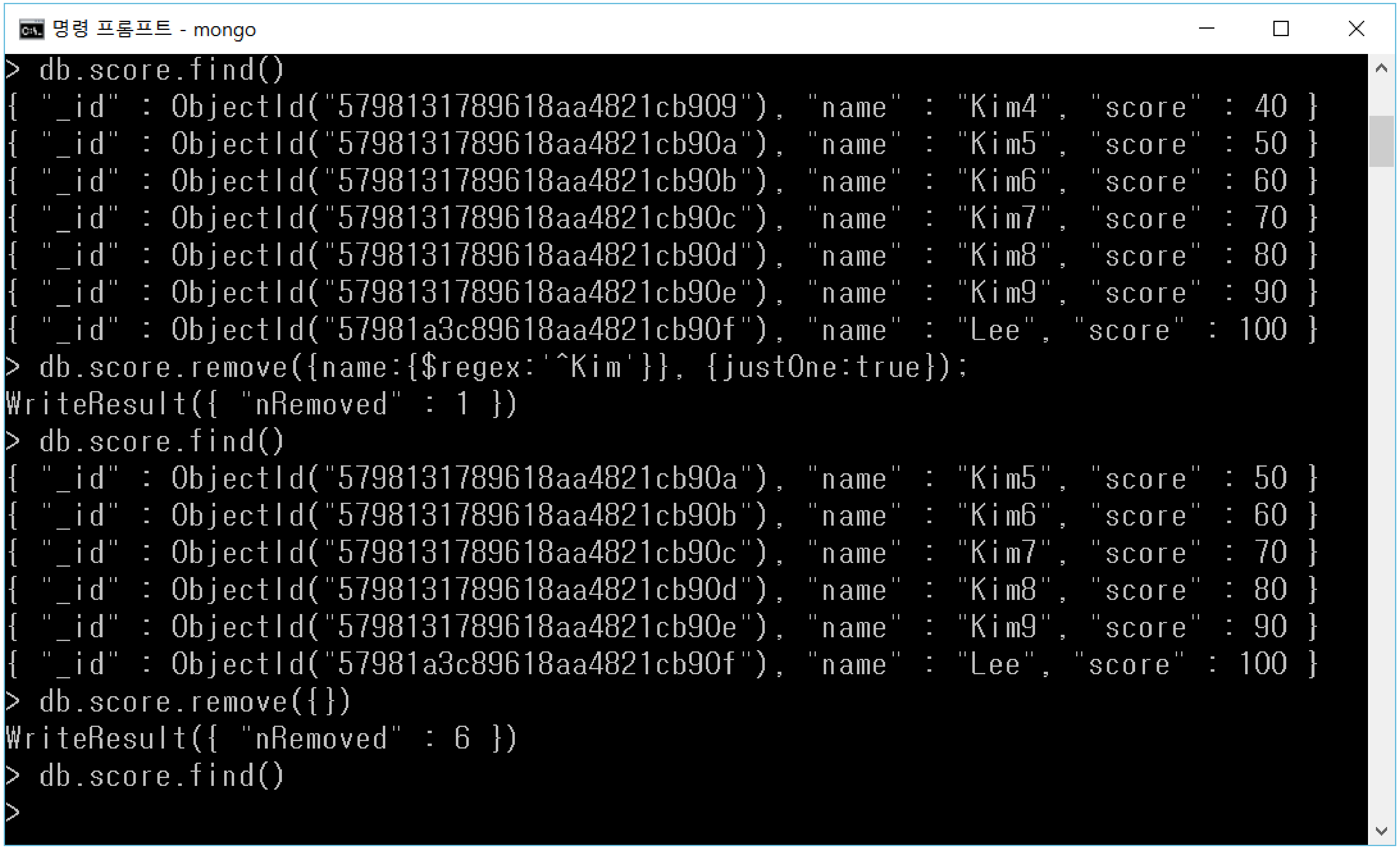
- db.컬렉션명.remove({ 삭제할 문서 })



```
> db.score.find()
{ "_id" : ObjectId("5798131789618aa4821cb905"), "name" : "Kim0", "score" : 0 }
{ "_id" : ObjectId("5798131789618aa4821cb906"), "name" : "Kim1", "score" : 10 }
{ "_id" : ObjectId("5798131789618aa4821cb907"), "name" : "Kim2", "score" : 20 }
{ "_id" : ObjectId("5798131789618aa4821cb908"), "name" : "Kim3", "score" : 30 }
{ "_id" : ObjectId("5798131789618aa4821cb909"), "name" : "Kim4", "score" : 40 }
{ "_id" : ObjectId("5798131789618aa4821cb90a"), "name" : "Kim5", "score" : 50 }
{ "_id" : ObjectId("5798131789618aa4821cb90b"), "name" : "Kim6", "score" : 60 }
{ "_id" : ObjectId("5798131789618aa4821cb90c"), "name" : "Kim7", "score" : 70 }
{ "_id" : ObjectId("5798131789618aa4821cb90d"), "name" : "Kim8", "score" : 80 }
{ "_id" : ObjectId("5798131789618aa4821cb90e"), "name" : "Kim9", "score" : 90 }
{ "_id" : ObjectId("57981a3c89618aa4821cb90f"), "name" : "Lee", "score" : 100 }
> db.score.remove({score:{$lt:40}});
WriteResult({ "nRemoved" : 4 })
> db.score.find()
{ "_id" : ObjectId("5798131789618aa4821cb909"), "name" : "Kim4", "score" : 40 }
{ "_id" : ObjectId("5798131789618aa4821cb90a"), "name" : "Kim5", "score" : 50 }
{ "_id" : ObjectId("5798131789618aa4821cb90b"), "name" : "Kim6", "score" : 60 }
{ "_id" : ObjectId("5798131789618aa4821cb90c"), "name" : "Kim7", "score" : 70 }
{ "_id" : ObjectId("5798131789618aa4821cb90d"), "name" : "Kim8", "score" : 80 }
{ "_id" : ObjectId("5798131789618aa4821cb90e"), "name" : "Kim9", "score" : 90 }
{ "_id" : ObjectId("57981a3c89618aa4821cb90f"), "name" : "Lee", "score" : 100 }
>
```

데이터 삭제

- db.컬렉션명.remove({ 삭제할 문서 })


명령 프롬프트 - mongo

```
> db.score.find()
{ "_id" : ObjectId("5798131789618aa4821cb909"), "name" : "Kim4", "score" : 40 }
{ "_id" : ObjectId("5798131789618aa4821cb90a"), "name" : "Kim5", "score" : 50 }
{ "_id" : ObjectId("5798131789618aa4821cb90b"), "name" : "Kim6", "score" : 60 }
{ "_id" : ObjectId("5798131789618aa4821cb90c"), "name" : "Kim7", "score" : 70 }
{ "_id" : ObjectId("5798131789618aa4821cb90d"), "name" : "Kim8", "score" : 80 }
{ "_id" : ObjectId("5798131789618aa4821cb90e"), "name" : "Kim9", "score" : 90 }
{ "_id" : ObjectId("57981a3c89618aa4821cb90f"), "name" : "Lee", "score" : 100 }
> db.score.remove({name:{$regex:'^Kim'}}, {justOne:true});
WriteResult({ "nRemoved" : 1 })
> db.score.find()
{ "_id" : ObjectId("5798131789618aa4821cb90a"), "name" : "Kim5", "score" : 50 }
{ "_id" : ObjectId("5798131789618aa4821cb90b"), "name" : "Kim6", "score" : 60 }
{ "_id" : ObjectId("5798131789618aa4821cb90c"), "name" : "Kim7", "score" : 70 }
{ "_id" : ObjectId("5798131789618aa4821cb90d"), "name" : "Kim8", "score" : 80 }
{ "_id" : ObjectId("5798131789618aa4821cb90e"), "name" : "Kim9", "score" : 90 }
{ "_id" : ObjectId("57981a3c89618aa4821cb90f"), "name" : "Lee", "score" : 100 }
> db.score.remove({})
WriteResult({ "nRemoved" : 6 })
> db.score.find()
>
```

[실습] 데이터 CRUD (생성)

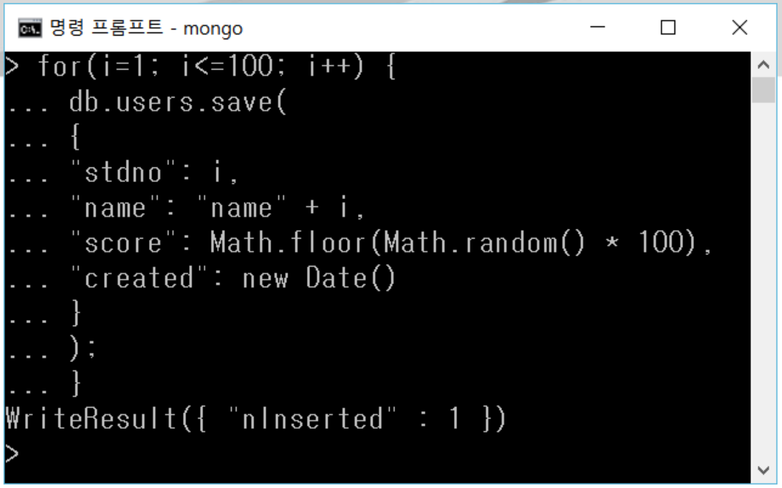
- “users” 컬렉션 사용
- 100명에 대한 Document Insert

  - stdno : 학생번호

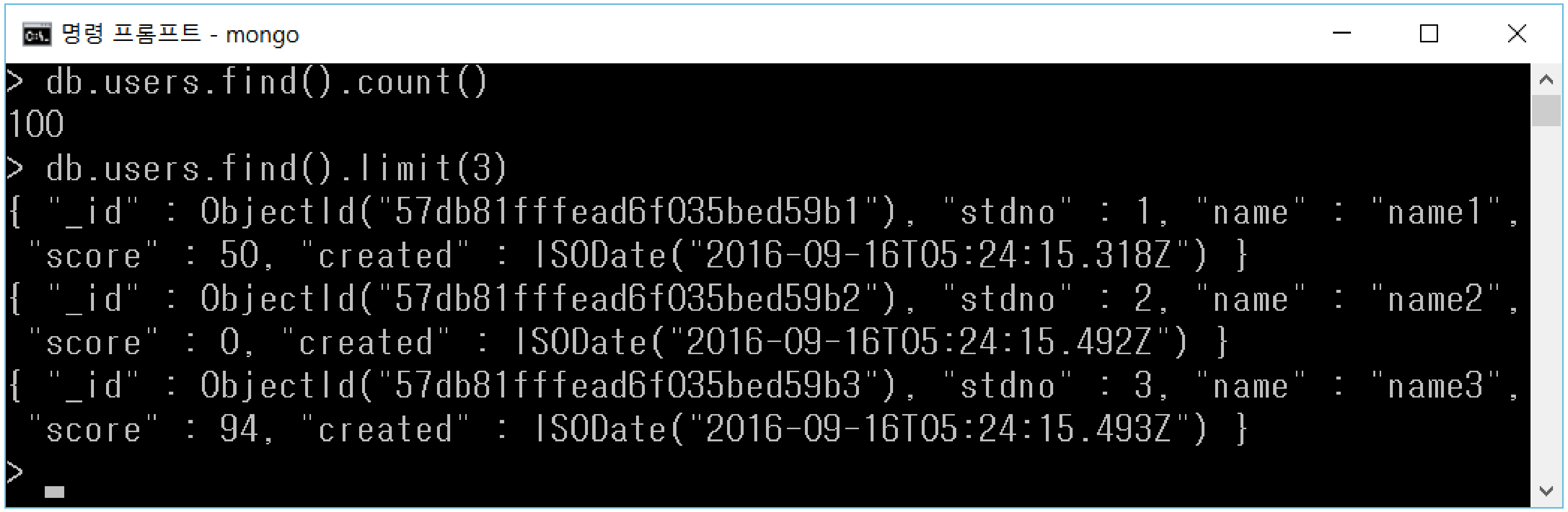
  - name : 이름

  - score : 점수 → Math.floor(Math.random( )*100)

  - created : 생성일자 → new Date( )


명령 프롬프트 - mongo

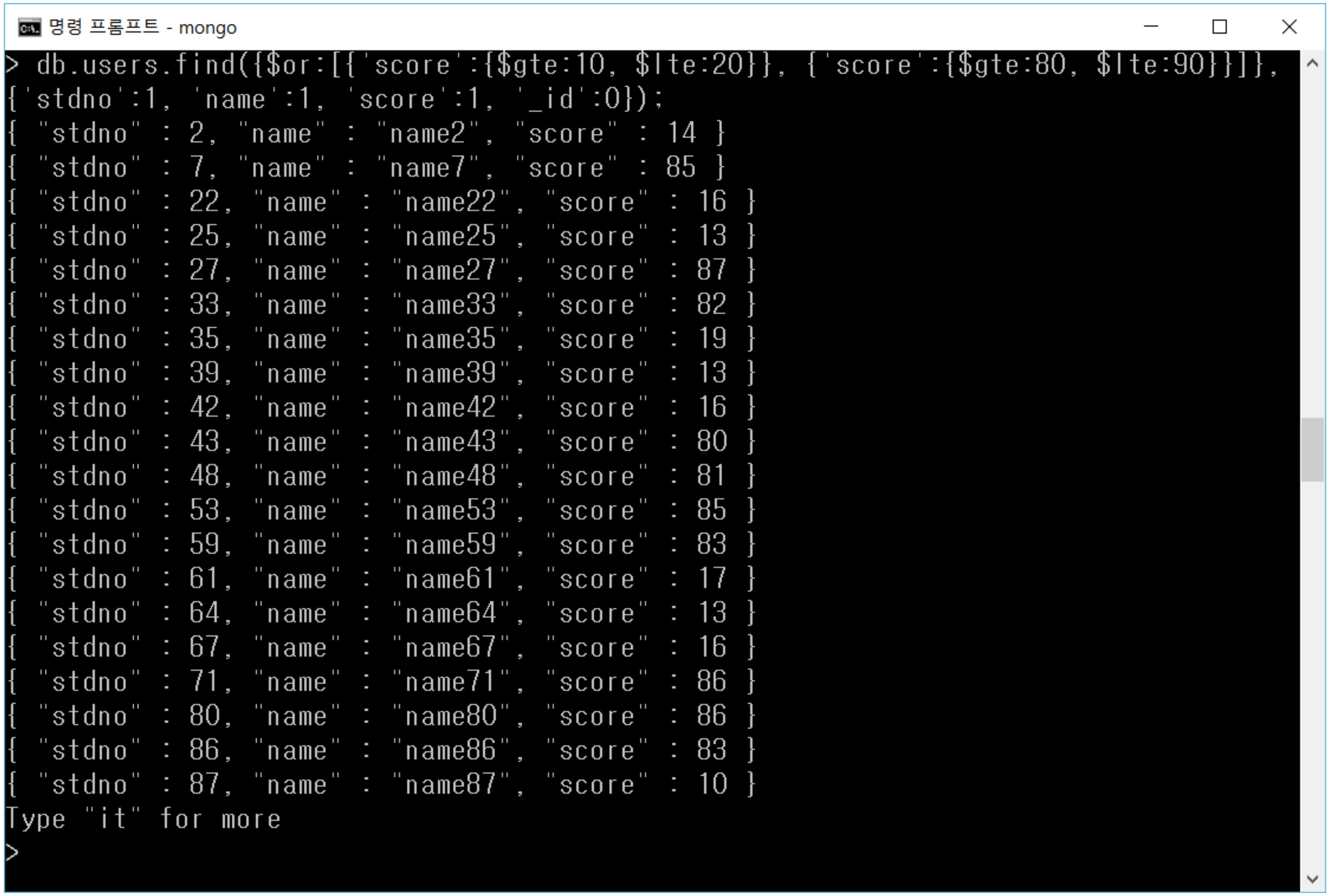
```
> for(i=1; i<=100; i++) {
... db.users.save(
... {
... "stdno": i,
... "name": "name" + i,
... "score": Math.floor(Math.random() * 100),
... "created": new Date()
... }
... );
... }
WriteResult({ "nInserted" : 1 })
>
```


명령 프롬프트 - mongo

```
> db.users.find().count()
100
> db.users.find().limit(3)
{ "_id" : ObjectId("57db81fffead6f035bed59b1"), "stdno" : 1, "name" : "name1",
 "score" : 50, "created" : ISODate("2016-09-16T05:24:15.318Z") }
{ "_id" : ObjectId("57db81fffead6f035bed59b2"), "stdno" : 2, "name" : "name2",
 "score" : 0, "created" : ISODate("2016-09-16T05:24:15.492Z") }
{ "_id" : ObjectId("57db81fffead6f035bed59b3"), "stdno" : 3, "name" : "name3",
 "score" : 94, "created" : ISODate("2016-09-16T05:24:15.493Z") }
>
```

[실습] 데이터 CRUD (검색)

- 10 <= score <= 20 이거나, 80 <= score <= 90인 stdno, name, score 검색

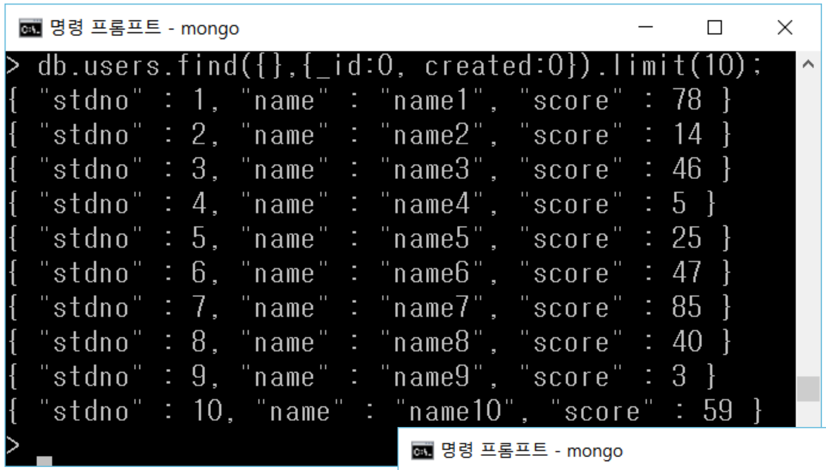


```
> db.users.find({$or:[{'score':{$gte:10, $lte:20}}, {'score':{$gte:80, $lte:90}}]},
{'stdno':1, 'name':1, 'score':1, '_id':0});
{ "stdno" : 2, "name" : "name2", "score" : 14 }
{ "stdno" : 7, "name" : "name7", "score" : 85 }
{ "stdno" : 22, "name" : "name22", "score" : 16 }
{ "stdno" : 25, "name" : "name25", "score" : 13 }
{ "stdno" : 27, "name" : "name27", "score" : 87 }
{ "stdno" : 33, "name" : "name33", "score" : 82 }
{ "stdno" : 35, "name" : "name35", "score" : 19 }
{ "stdno" : 39, "name" : "name39", "score" : 13 }
{ "stdno" : 42, "name" : "name42", "score" : 16 }
{ "stdno" : 43, "name" : "name43", "score" : 80 }
{ "stdno" : 48, "name" : "name48", "score" : 81 }
{ "stdno" : 53, "name" : "name53", "score" : 85 }
{ "stdno" : 59, "name" : "name59", "score" : 83 }
{ "stdno" : 61, "name" : "name61", "score" : 17 }
{ "stdno" : 64, "name" : "name64", "score" : 13 }
{ "stdno" : 67, "name" : "name67", "score" : 16 }
{ "stdno" : 71, "name" : "name71", "score" : 86 }
{ "stdno" : 80, "name" : "name80", "score" : 86 }
{ "stdno" : 86, "name" : "name86", "score" : 83 }
{ "stdno" : 87, "name" : "name87", "score" : 10 }
Type "it" for more
>
```

[실습] 데이터 CRUD (수정)

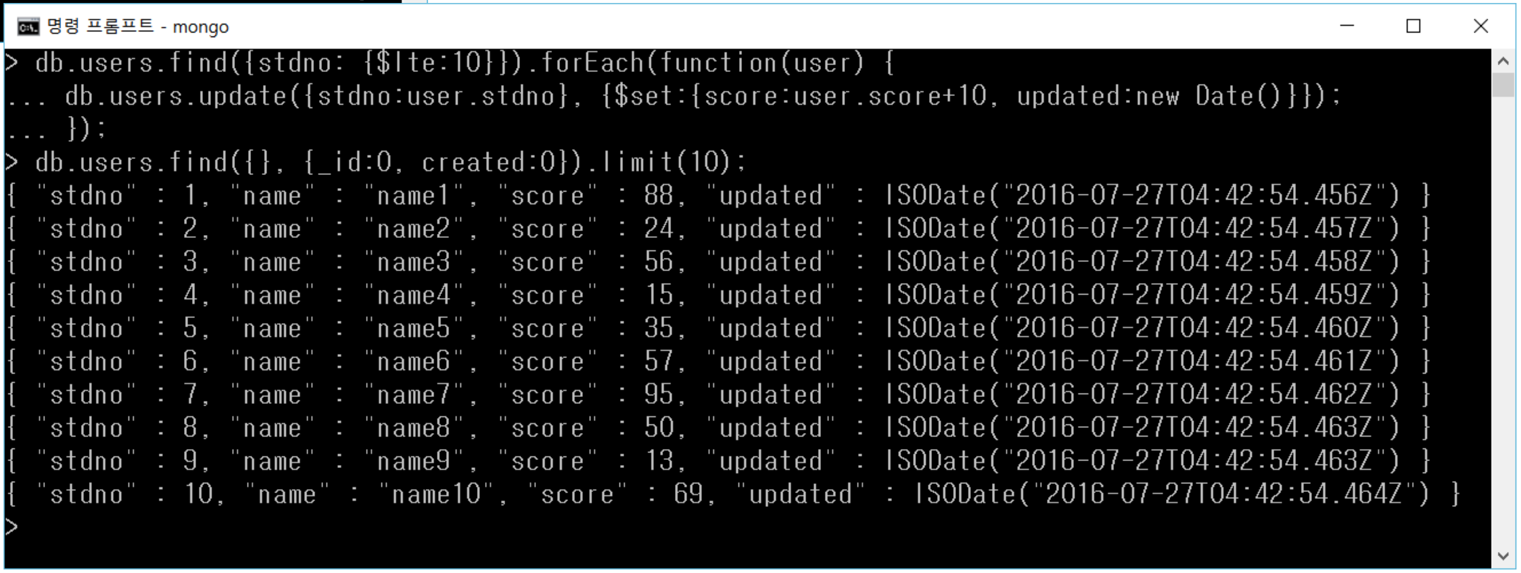
- stdno가 1~10번인 학생들의 score 점수를 10점씩 올리고, 변경일자 추가하기

  - updated : 변경일자



명령 프롬프트 - mongo

```
> db.users.find({},{_id:0, created:0}).limit(10);
{ "stdno" : 1, "name" : "name1", "score" : 78 }
{ "stdno" : 2, "name" : "name2", "score" : 14 }
{ "stdno" : 3, "name" : "name3", "score" : 46 }
{ "stdno" : 4, "name" : "name4", "score" : 5 }
{ "stdno" : 5, "name" : "name5", "score" : 25 }
{ "stdno" : 6, "name" : "name6", "score" : 47 }
{ "stdno" : 7, "name" : "name7", "score" : 85 }
{ "stdno" : 8, "name" : "name8", "score" : 40 }
{ "stdno" : 9, "name" : "name9", "score" : 3 }
{ "stdno" : 10, "name" : "name10", "score" : 59 }
>
```

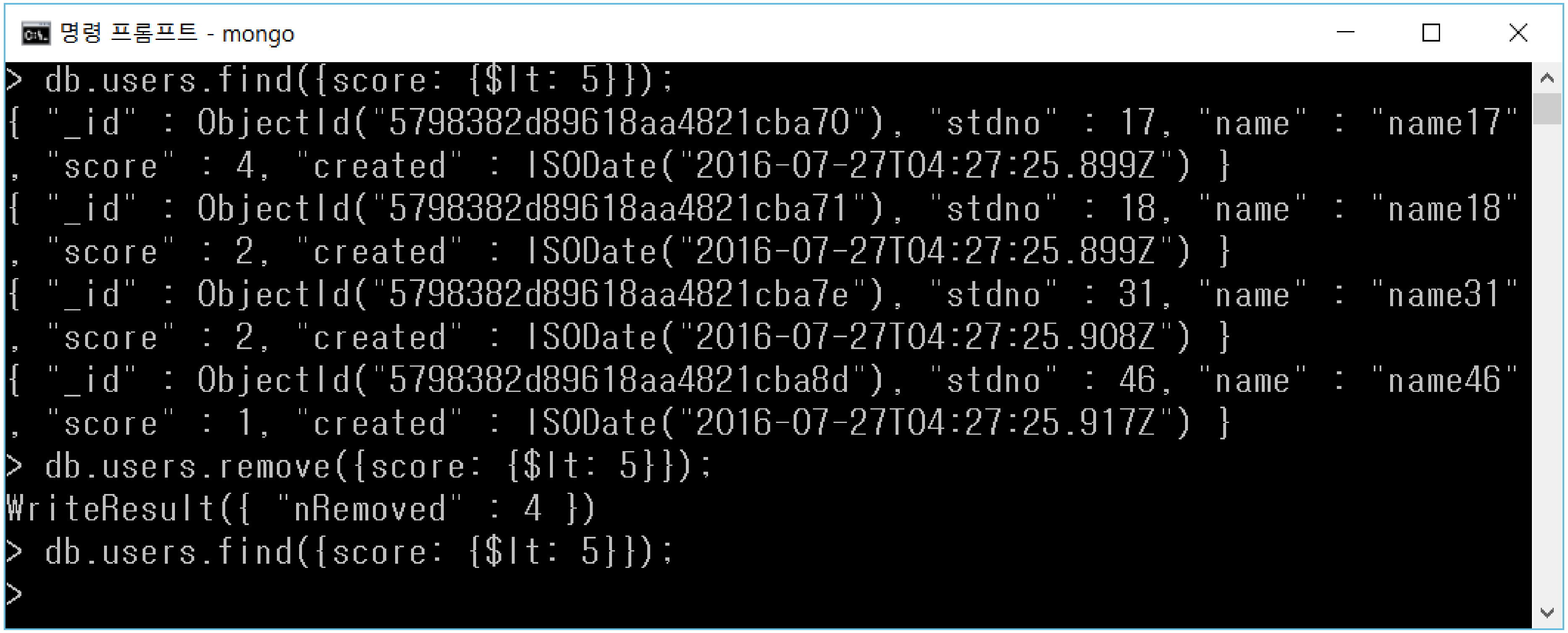


명령 프롬프트 - mongo

```
> db.users.find({stdno: {$lte:10}}).forEach(function(user) {
... db.users.update({stdno:user.stdno}, {$set:{score:user.score+10, updated:new Date()}});
... });
> db.users.find({}, {_id:0, created:0}).limit(10);
{ "stdno" : 1, "name" : "name1", "score" : 88, "updated" : ISODate("2016-07-27T04:42:54.456Z") }
{ "stdno" : 2, "name" : "name2", "score" : 24, "updated" : ISODate("2016-07-27T04:42:54.457Z") }
{ "stdno" : 3, "name" : "name3", "score" : 56, "updated" : ISODate("2016-07-27T04:42:54.458Z") }
{ "stdno" : 4, "name" : "name4", "score" : 15, "updated" : ISODate("2016-07-27T04:42:54.459Z") }
{ "stdno" : 5, "name" : "name5", "score" : 35, "updated" : ISODate("2016-07-27T04:42:54.460Z") }
{ "stdno" : 6, "name" : "name6", "score" : 57, "updated" : ISODate("2016-07-27T04:42:54.461Z") }
{ "stdno" : 7, "name" : "name7", "score" : 95, "updated" : ISODate("2016-07-27T04:42:54.462Z") }
{ "stdno" : 8, "name" : "name8", "score" : 50, "updated" : ISODate("2016-07-27T04:42:54.463Z") }
{ "stdno" : 9, "name" : "name9", "score" : 13, "updated" : ISODate("2016-07-27T04:42:54.463Z") }
{ "stdno" : 10, "name" : "name10", "score" : 69, "updated" : ISODate("2016-07-27T04:42:54.464Z") }
>
```

[실습] 데이터 CRUD (삭제)

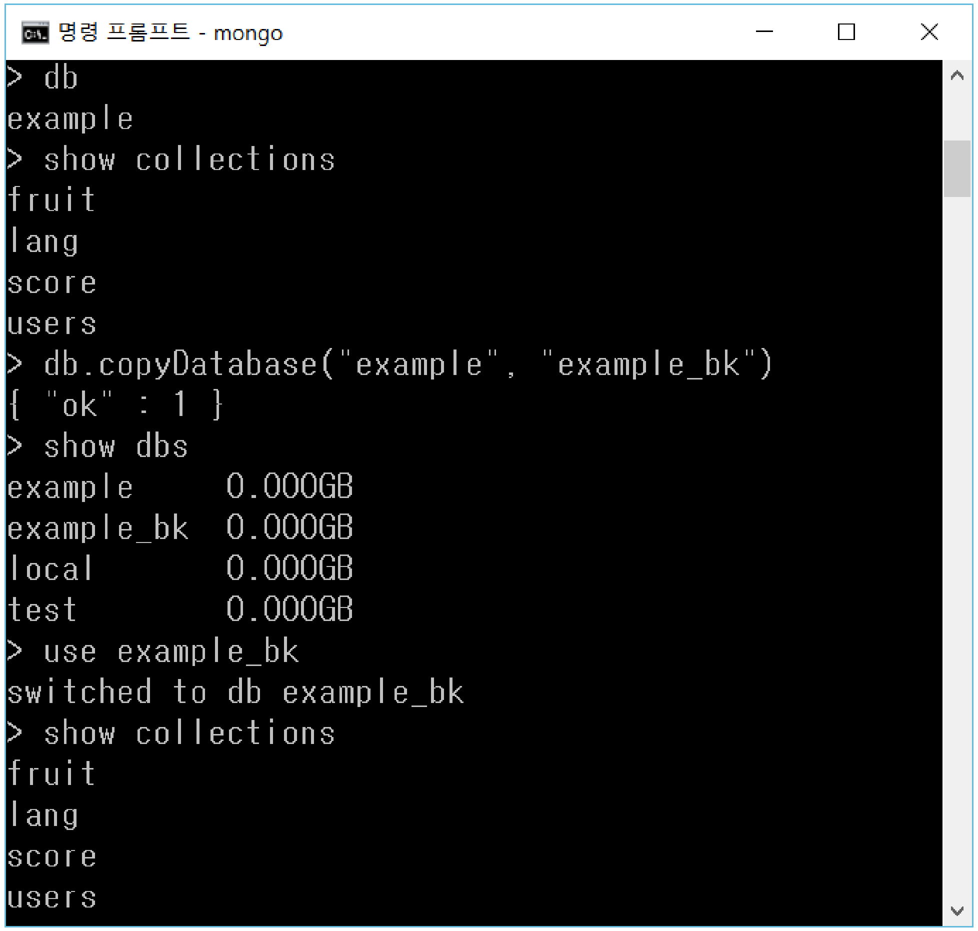
- score 점수가 5점 미만인 데이터 삭제



```
> db.users.find({score: {$lt: 5}});
{ "_id" : ObjectId("5798382d89618aa4821cba70"), "stdno" : 17, "name" : "name17"
, "score" : 4, "created" : ISODate("2016-07-27T04:27:25.899Z") }
{ "_id" : ObjectId("5798382d89618aa4821cba71"), "stdno" : 18, "name" : "name18"
, "score" : 2, "created" : ISODate("2016-07-27T04:27:25.899Z") }
{ "_id" : ObjectId("5798382d89618aa4821cba7e"), "stdno" : 31, "name" : "name31"
, "score" : 2, "created" : ISODate("2016-07-27T04:27:25.908Z") }
{ "_id" : ObjectId("5798382d89618aa4821cba8d"), "stdno" : 46, "name" : "name46"
, "score" : 1, "created" : ISODate("2016-07-27T04:27:25.917Z") }
> db.users.remove({score: {$lt: 5}});
WriteResult({ "nRemoved" : 4 })
> db.users.find({score: {$lt: 5}});
>
```

데이터베이스 백업

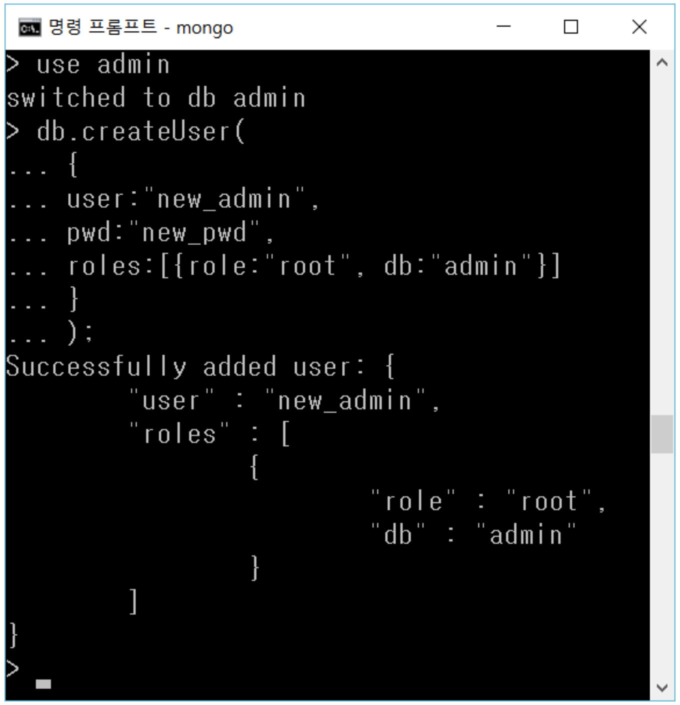
- 로컬에서 데이터베이스 복사 : db.copyDatabase(“db명”, “copy db명”);
- 원격지로 데이터베이스 복사 : db.copyDatabase(“db명", “copy db명", "192.168.1.77:27017");



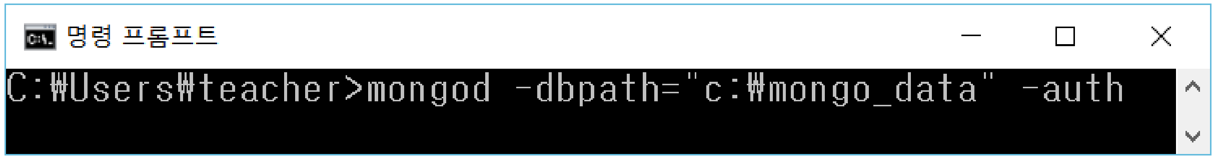
```
> db
example
> show collections
fruit
lang
score
users
> db.copyDatabase("example", "example_bk")
{ "ok" : 1 }
> show dbs
example     0.000GB
example_bk  0.000GB
local       0.000GB
test        0.000GB
> use example_bk
switched to db example_bk
> show collections
fruit
lang
score
users
```

데이터베이스 보안

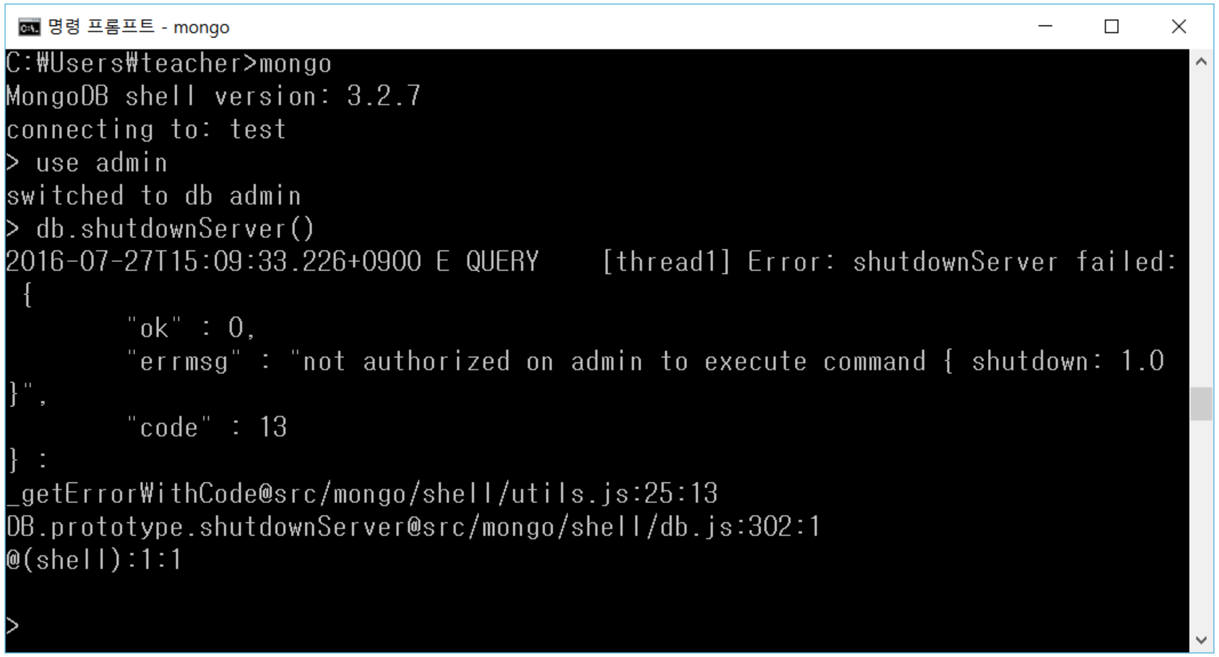
- createUser( ) : user 생성, removeUser( ) : user 삭제



```
> use admin
switched to db admin
> db.createUser(
... {
... user:"new_admin",
... pwd:"new_pwd",
... roles:[{role:"root", db:"admin"}]
... }
... );
Successfully added user: {
        "user" : "new_admin",
        "roles" : [
                {
                        "role" : "root",
                        "db" : "admin"
                }
        ]
}
> ▬
```



```
C:₩Users₩teacher>mongod -dbpath="c:₩mongo_data" -auth
```

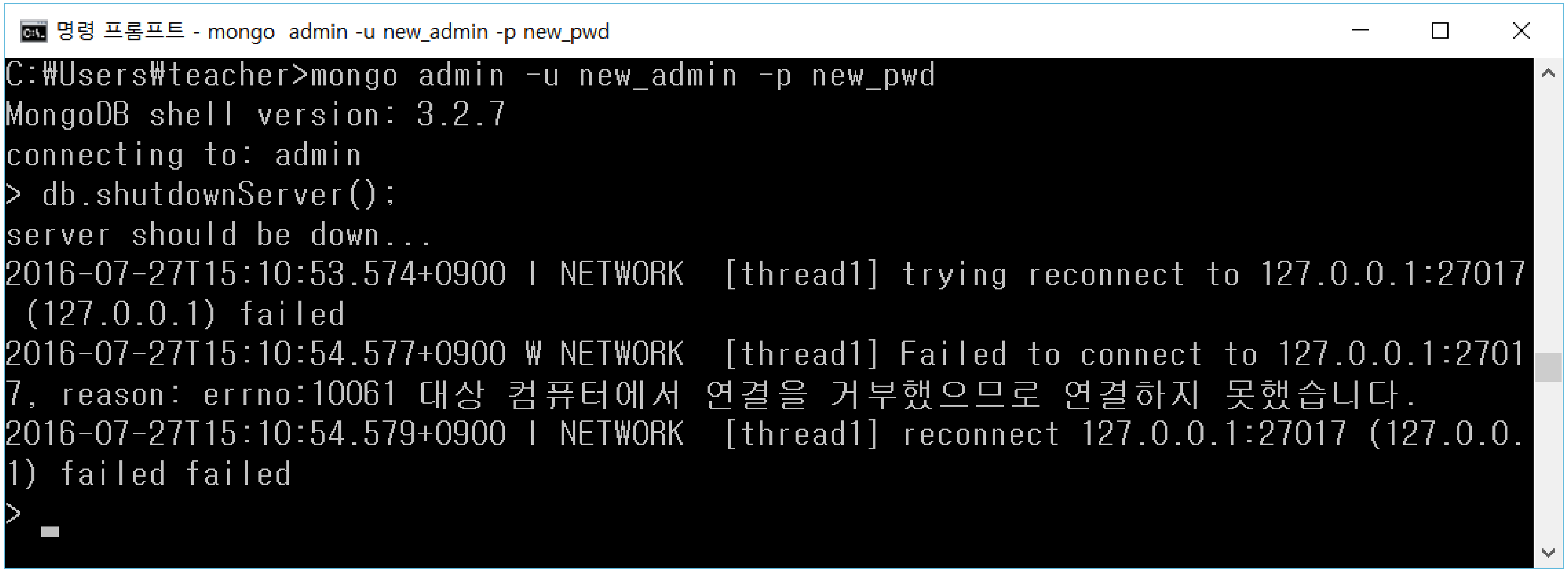


```
C:₩Users₩teacher>mongo
MongoDB shell version: 3.2.7
connecting to: test
> use admin
switched to db admin
> db.shutdownServer()
2016-07-27T15:09:33.226+0900 E QUERY    [thread1] Error: shutdownServer failed:
 {
        "ok" : 0,
        "errmsg" : "not authorized on admin to execute command { shutdown: 1.0
}",
        "code" : 13
} :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
DB.prototype.shutdownServer@src/mongo/shell/db.js:302:1
@(shell):1:1

>
```

데이터베이스 보안

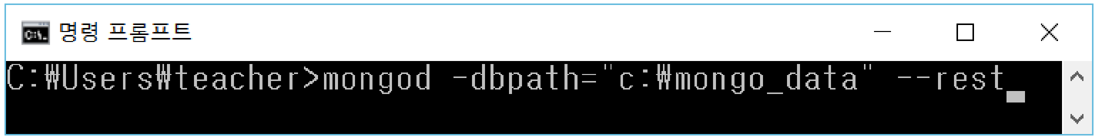
- createUser( ) : user 생성, removeUser( ) : user 삭제



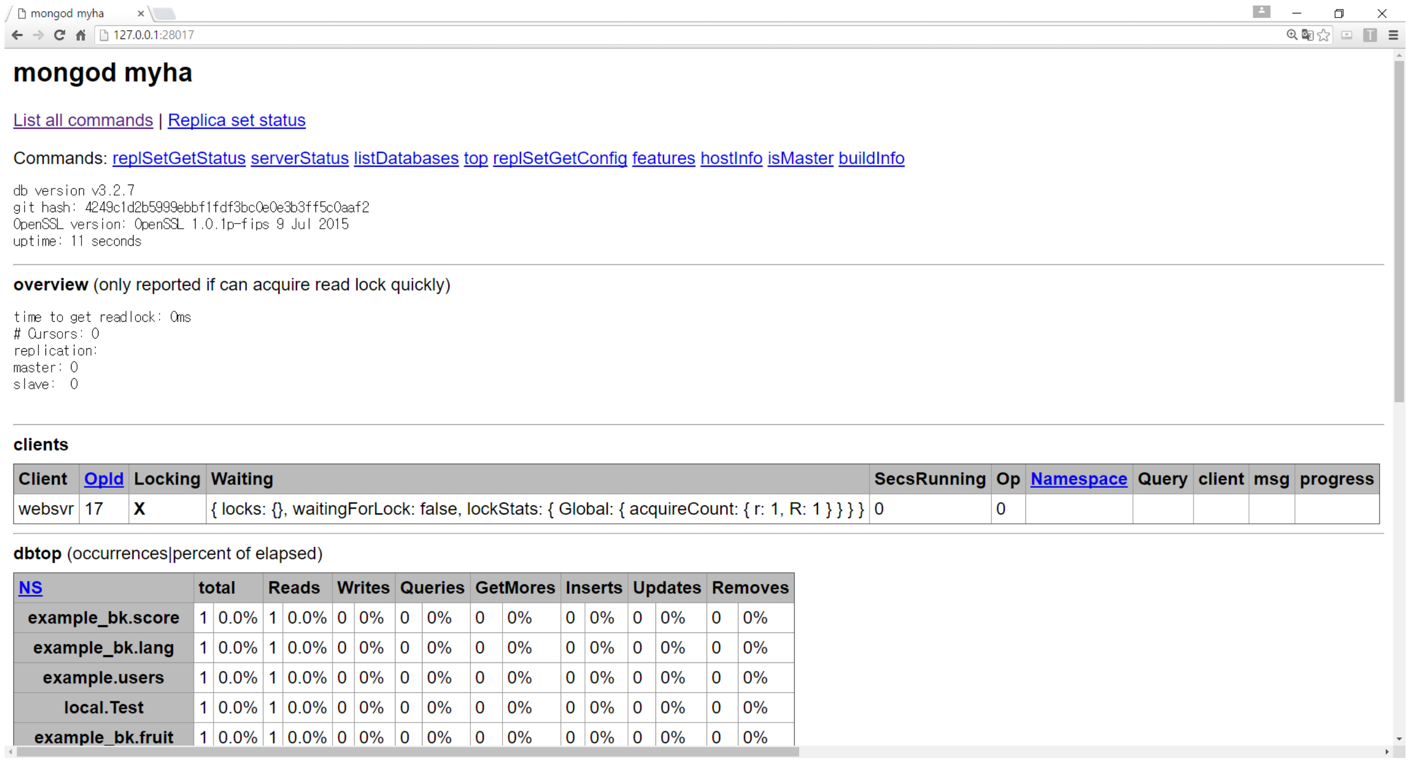
```
C:₩Users₩teacher>mongo admin -u new_admin -p new_pwd
MongoDB shell version: 3.2.7
connecting to: admin
> db.shutdownServer();
server should be down...
2016-07-27T15:10:53.574+0900 I NETWORK  [thread1] trying reconnect to 127.0.0.1:27017
 (127.0.0.1) failed
2016-07-27T15:10:54.577+0900 W NETWORK  [thread1] Failed to connect to 127.0.0.1:2701
7, reason: errno:10061 대상 컴퓨터에서 연결을 거부했으므로 연결하지 못했습니다.
2016-07-27T15:10:54.579+0900 I NETWORK  [thread1] reconnect 127.0.0.1:27017 (127.0.0.
1) failed failed
>
```

웹 모니터링 도구

- 웹으로 MongoDB의 상황을 모니터링 할 수 있는 환경을 제공함
- mongod 실행 시 --rest 옵션 추가 → http://127.0.0.1:28017



```
C:₩Users₩teacher>mongod -dbpath="c:₩mongo_data" --rest
```



mongod myha

List all commands | Replica set status

Commands: replSetGetStatus serverStatus listDatabases top replSetGetConfig features hostInfo isMaster buildInfo

```
db version v3.2.7
git hash: 4249c1d2b5999ebbf1fdf3bc0e0e3b3ff5c0aaf2
OpenSSL version: OpenSSL 1.0.1p-fips 9 Jul 2015
uptime: 11 seconds
```

overview (only reported if can acquire read lock quickly)

```
time to get readlock: 0ms
# Cursors: 0
replication:
master: 0
slave:  0
```

clients

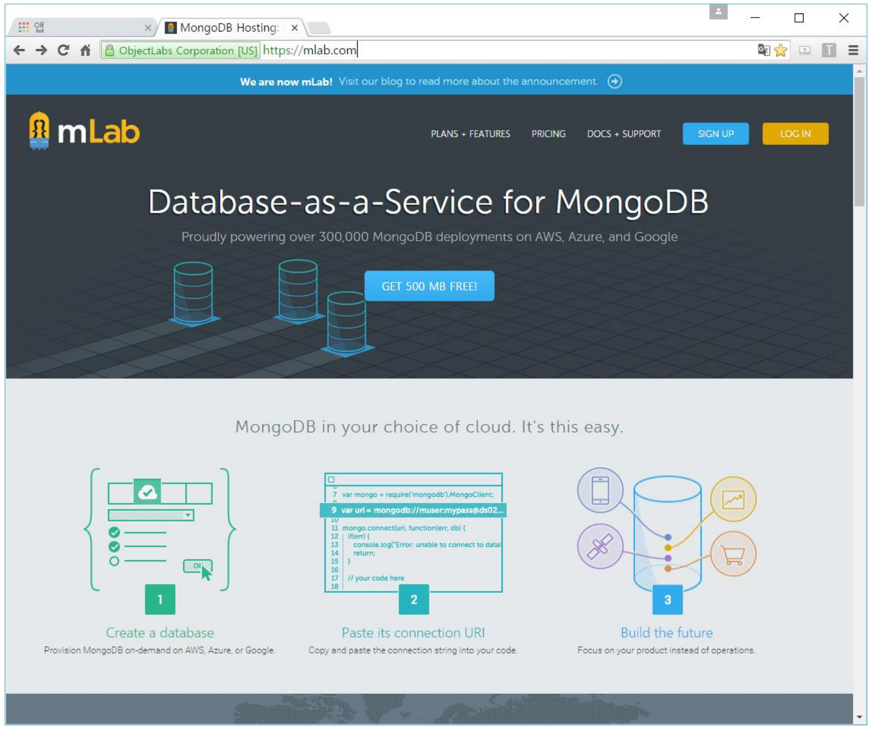
Client	OpId	Locking	Waiting	SecsRunning	Op	Namespace	Query	client	msg	progress
websvr	17	X	{ locks: {}, waitingForLock: false, lockStats: { Global: { acquireCount: { r: 1, R: 1 } } } }	0	0					

dbtop (occurrences\|percent of elapsed)

NS	total		Reads		Writes		Queries		GetMores		Inserts		Updates		Removes	
example_bk.score	1	0.0%	1	0.0%	0	0%	0	0%	0	0%	0	0%	0	0%	0	0%
example_bk.lang	1	0.0%	1	0.0%	0	0%	0	0%	0	0%	0	0%	0	0%	0	0%
example.users	1	0.0%	1	0.0%	0	0%	0	0%	0	0%	0	0%	0	0%	0	0%
local.Test	1	0.0%	1	0.0%	0	0%	0	0%	0	0%	0	0%	0	0%	0	0%
example_bk.fruit	1	0.0%	1	0.0%	0	0%	0	0%	0	0%	0	0%	0	0%	0	0%

클라우드 호스팅

- 몽고랩 : https://mlab.com
- 0.5GB까지 저장공간 무료 사용



MongoDB Hosting:

ObjectLabs Corporation [US] https://mlab.com

We are now mLab! Visit our blog to read more about the announcement.

mLab

PLANS + FEATURES PRICING DOCS + SUPPORT SIGN UP LOG IN

Database-as-a-Service for MongoDB

Proudly powering over 300,000 MongoDB deployments on AWS, Azure, and Google

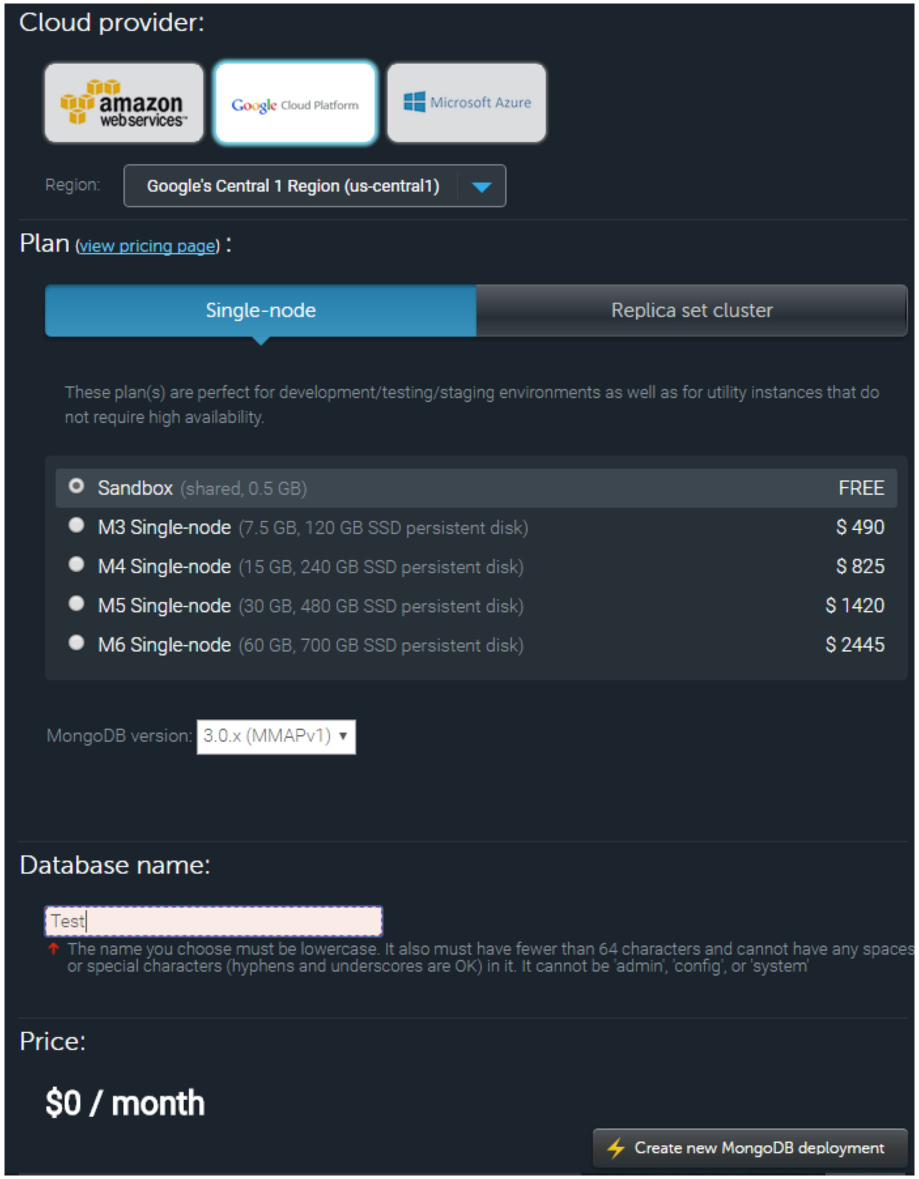
GET 500 MB FREE!

MongoDB in your choice of cloud. It's this easy.

1 Create a database
Provision MongoDB on-demand on AWS, Azure, or Google.

2 Paste its connection URI
Copy and paste the connection string into your code.

3 Build the future
Focus on your product instead of operations.



Cloud provider:

amazon web services | Google Cloud Platform | Microsoft Azure

Region: Google's Central 1 Region (us-central1)

Plan (view pricing page) :

Single-node | Replica set cluster

These plan(s) are perfect for development/testing/staging environments as well as for utility instances that do not require high availability.

Sandbox (shared, 0.5 GB)	FREE
M3 Single-node (7.5 GB, 120 GB SSD persistent disk)	$ 490
M4 Single-node (15 GB, 240 GB SSD persistent disk)	$ 825
M5 Single-node (30 GB, 480 GB SSD persistent disk)	$ 1420
M6 Single-node (60 GB, 700 GB SSD persistent disk)	$ 2445

MongoDB version: 3.0.x (MMAPv1)

Database name:

Test

The name you choose must be lowercase. It also must have fewer than 64 characters and cannot have any spaces or special characters (hyphens and underscores are OK) in it. It cannot be 'admin', 'config', or 'system'

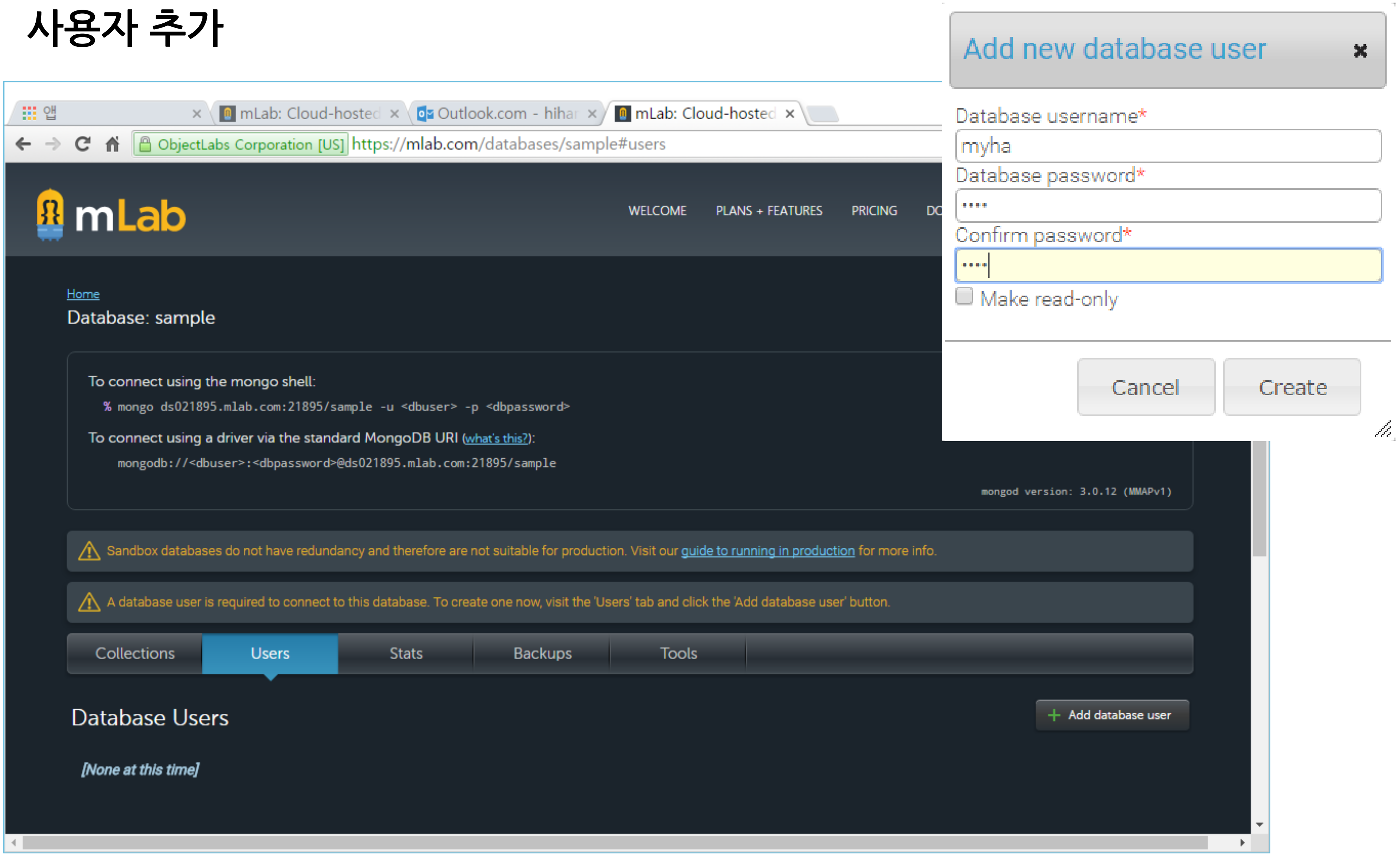
Price:

$0 / month

Create new MongoDB deployment

클라우드 호스팅

- 사용자 추가



ObjectLabs Corporation [US] https://mlab.com/databases/sample#users

Database: sample

To connect using the mongo shell:

```
% mongo ds021895.mlab.com:21895/sample -u <dbuser> -p <dbpassword>
```

To connect using a driver via the standard MongoDB URI (what's this?):

```
mongodb://<dbuser>:<dbpassword>@ds021895.mlab.com:21895/sample
```

```
mongod version: 3.0.12 (MMAPv1)
```

Sandbox databases do not have redundancy and therefore are not suitable for production. Visit our guide to running in production for more info.

A database user is required to connect to this database. To create one now, visit the 'Users' tab and click the 'Add database user' button.

Database Users

[None at this time]

Add new database user

Database username*: myha

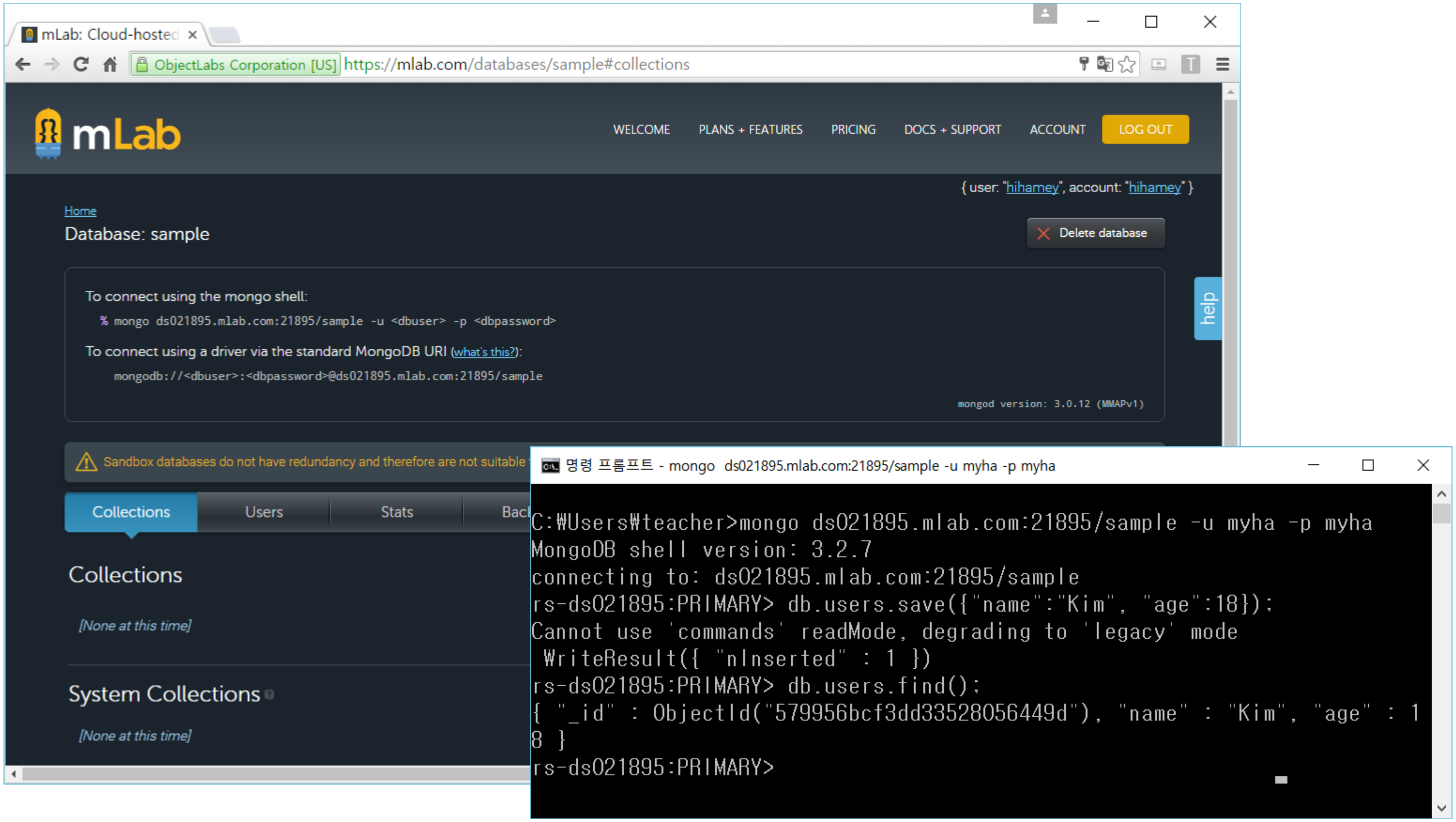
Database password*: ••••

Confirm password*: ••••

Make read-only: [ ]

클라우드 호스팅

- Mongo 쉘 로그인



mLab: Cloud-hosted
ObjectLabs Corporation [US] https://mlab.com/databases/sample#collections
mLab
WELCOME
PLANS + FEATURES
PRICING
DOCS + SUPPORT
ACCOUNT
LOG OUT
Delete database
help
Collections
Users
Stats
Bac
명령 프롬프트 - mongo ds021895.mlab.com:21895/sample -u myha -p myha

{ user: "hihamey", account: "hihamey" }

Home
Database: sample

To connect using the mongo shell:

```
% mongo ds021895.mlab.com:21895/sample -u <dbuser> -p <dbpassword>
```

To connect using a driver via the standard MongoDB URI (what's this?):

```
mongodb://<dbuser>:<dbpassword>@ds021895.mlab.com:21895/sample
```

```
mongod version: 3.0.12 (MMAPv1)
```

Sandbox databases do not have redundancy and therefore are not suitable

Collections

[None at this time]

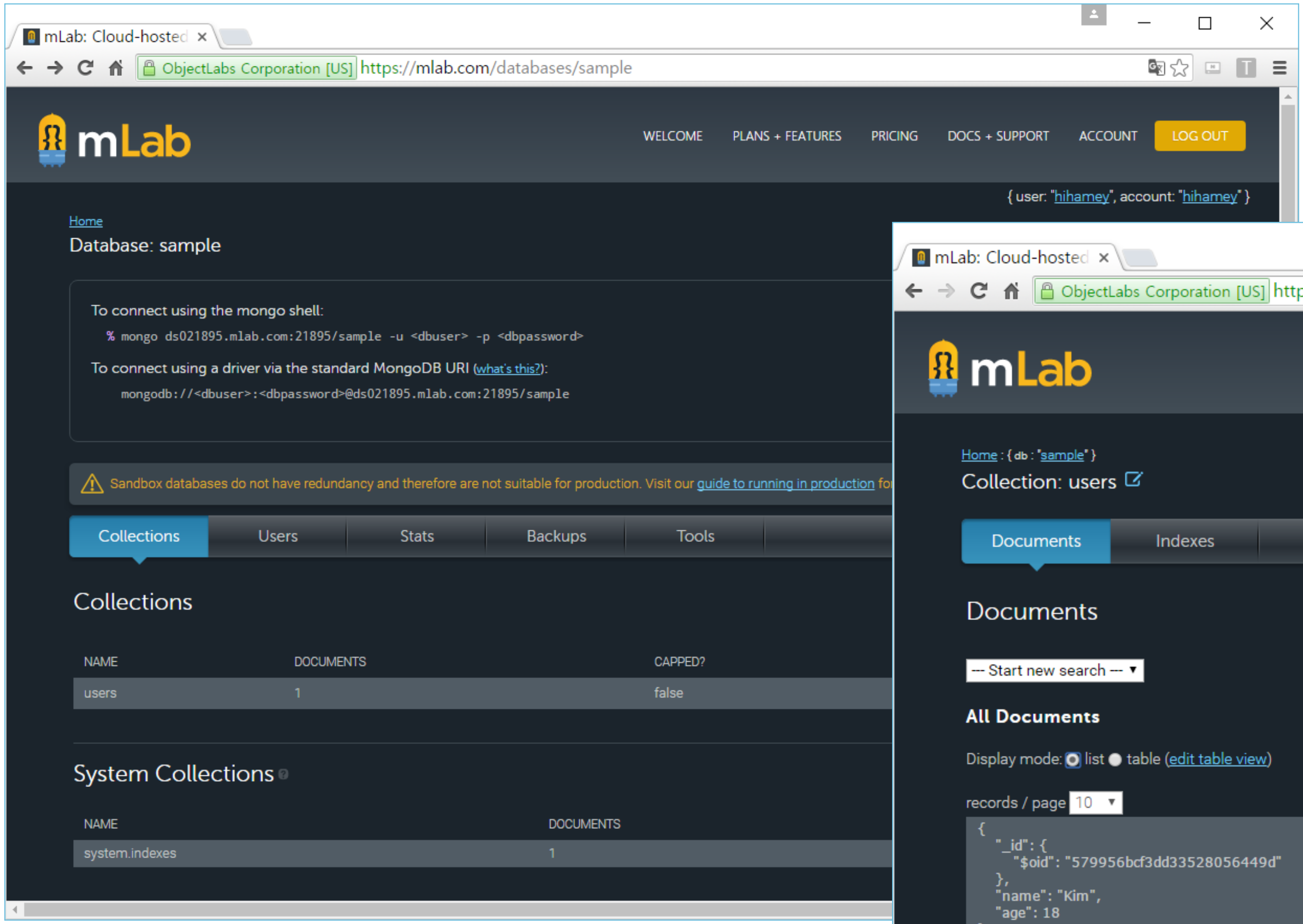
System Collections

[None at this time]

```
C:₩Users₩teacher>mongo ds021895.mlab.com:21895/sample -u myha -p myha
MongoDB shell version: 3.2.7
connecting to: ds021895.mlab.com:21895/sample
rs-ds021895:PRIMARY> db.users.save({"name":"Kim", "age":18});
Cannot use 'commands' readMode, degrading to 'legacy' mode
 WriteResult({ "nInserted" : 1 })
rs-ds021895:PRIMARY> db.users.find();
{ "_id" : ObjectId("579956bcf3dd33528056449d"), "name" : "Kim", "age" : 1
8 }
rs-ds021895:PRIMARY>
```

클라우드 호스팅

- Collections → Documents


mLab: Cloud-hosted ×

ObjectLabs Corporation [US] https://mlab.com/databases/sample

{ user: "hihamey", account: "hihamey" }

Home

Database: sample

To connect using the mongo shell:

```
% mongo ds021895.mlab.com:21895/sample -u <dbuser> -p <dbpassword>
```

To connect using a driver via the standard MongoDB URI (what's this?):

```
mongodb://<dbuser>:<dbpassword>@ds021895.mlab.com:21895/sample
```

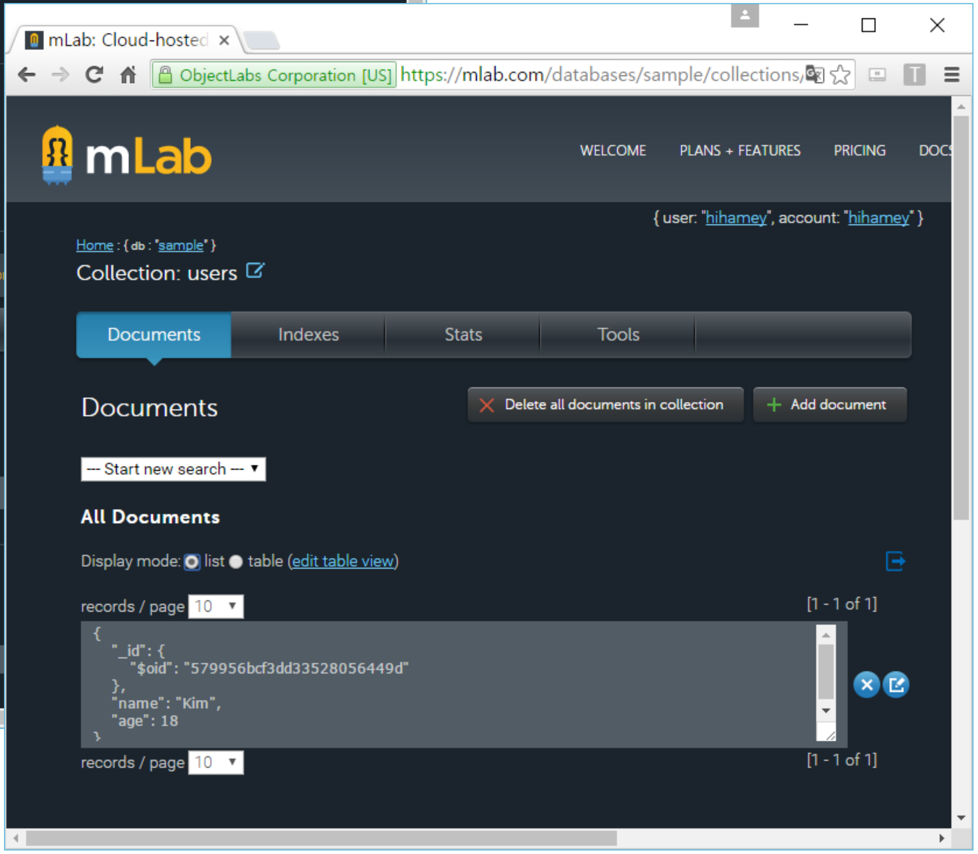
Sandbox databases do not have redundancy and therefore are not suitable for production. Visit our guide to running in production fo

Collections

NAME	DOCUMENTS	CAPPED?
users	1	false

System Collections

NAME	DOCUMENTS
system.indexes	1


mLab: Cloud-hosted ×

ObjectLabs Corporation [US] https://mlab.com/databases/sample/collections/

{ user: "hihamey", account: "hihamey" }

Home : { db : "sample" }

Collection: users

Documents

All Documents

Display mode: [✓ list] table (edit table view)

records / page 10 [1 - 1 of 1]

```
{
    "_id": {
        "$oid": "579956bcf3dd33528056449d"
    },
    "name": "Kim",
    "age": 18
}
```

records / page 10 [1 - 1 of 1]