

NoSQL

몽고디비

E조

데이터사이언스전공 20221608 권유나
데이터사이언스전공 20221612 김서윤
데이터사이언스전공 20221635 이유나
데이터사이언스전공 20221655 황진하

Part 1

- NoSQL 등장배경
- 몽고디비의 정의 및 특징
- 장단점
- 사용추세 및 사용사례

NoSQL 등장배경



대표적인 데이터베이스
관리 시스템은 관계
DBMS라는 인식



소셜 네트워크 서비스
이용 폭발적으로 증가
-> 사진, 동영상, 검색
로그와 같은 비정형 데
이터가 대량생산



클라우드 컴퓨팅, 빅데
이터의 개념이 등장



여러 대의 서버 컴퓨터
에 데이터를 분산하여
저장 및 처리하는 환경



구글, 페이스북, 트위터
같은 SNS를 제공하는
회사들이 NoSQL의 필
요성을 강조

MongoDB의 정의



- **도큐먼트 지향** 데이터 베이스 시스템
- **JSON**과 같은 **동적 스키마형 도큐먼트**들을 선호
- 특정한 종류의 애플리케이션을 더 쉽고 빠르게 데이터 통합 가능
- Memory Mapping 기술을 기반으로 빅데이터 처리에 성능이 탁월
- mongoDB <-- **HUMONGOUS**



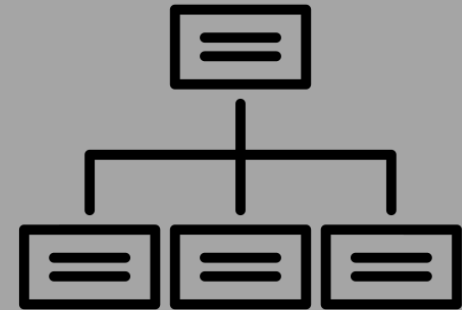
MongoDB의 특징



도큐먼트 지향적인 데이터베이스

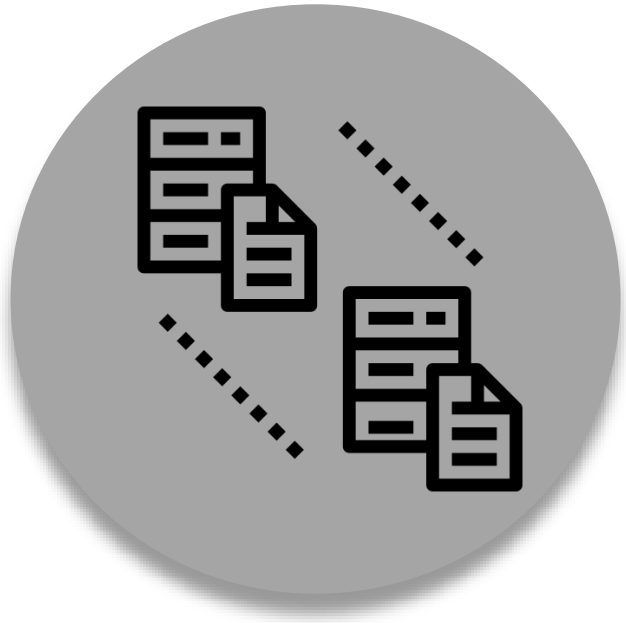


광범위한 언어 지원



B-Tree 인덱스 지원

MongoDB의 특징



자동 장애 조치



데이터 집계를 위한 기본 도구 제공

MongoDB의 장점



- 스키마가 없는 데이터 모델을 통해 다양한 형태의 데이터를 표현 가능
- 뛰어난 유연성, 데이터 구조의 변경 용이
- 쉬운 사용 방법, 편리한 개발
- 여러개의 세컨더리 인덱스를 허용



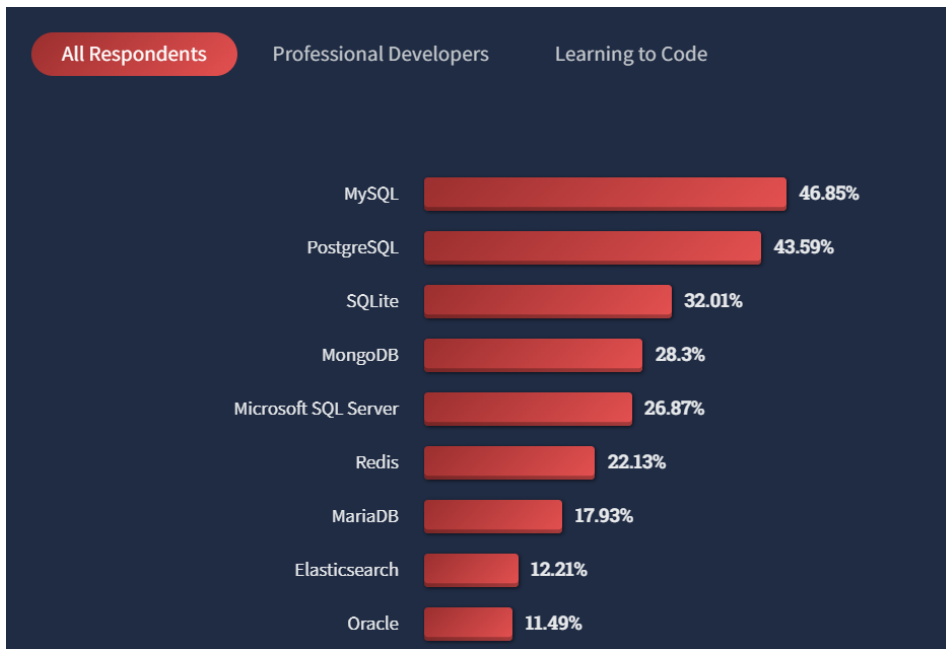
MongoDB의 단점



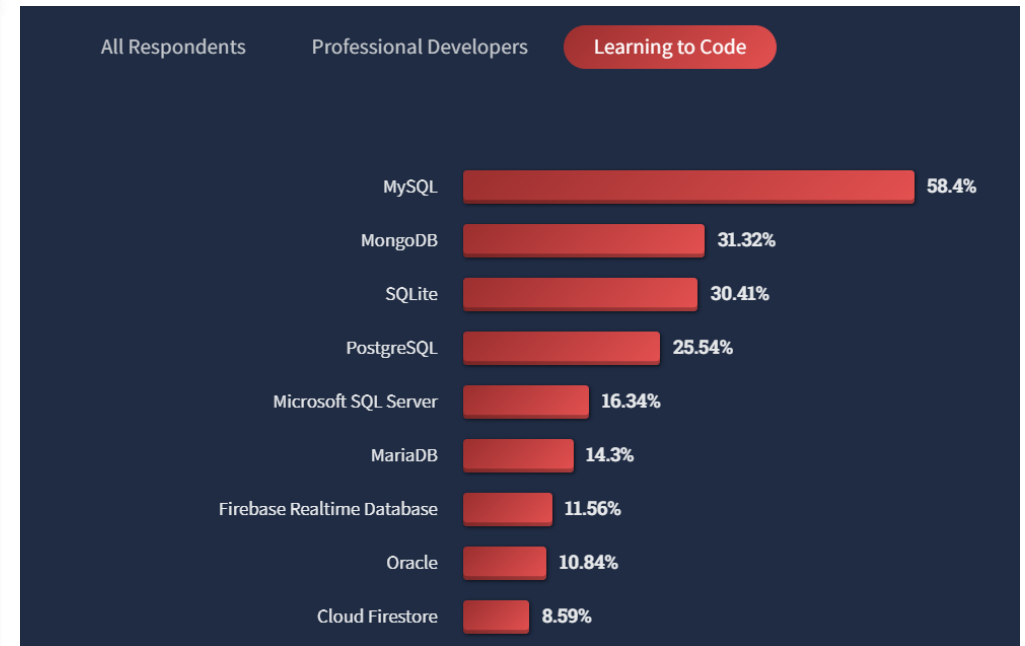
- 많은 인덱스 사용 시, 충분한 메모리 확보 필요 (B트리는 크기가 커질수록 새로운 데이터를 입력하거나 삭제할 때 성능이 저하)
- RDBMS에 비해 더 많은 데이터 공간 소모 (비효율적인 Key 중복 입력)



사용추세



데이터베이스 기술 중 4위



가장 배우길 원하는 데이터베이스 중 2위



사용사례 및 기업

NAVER



Google Cloud

kakao



NEXON



LG

aws



사용사례 및 기업

helvetia 

Boots

V O L V O

 thinkable

» Bendigo and
Adelaide Bank

V O L V O

65M+

daily events processed

40%

more resource efficient

| MANUFACTURING

“We expect to process 20 times as many events in just two years’ time – that’s two billion per day and Atlas is well placed to handle that massive increase without a problem.”

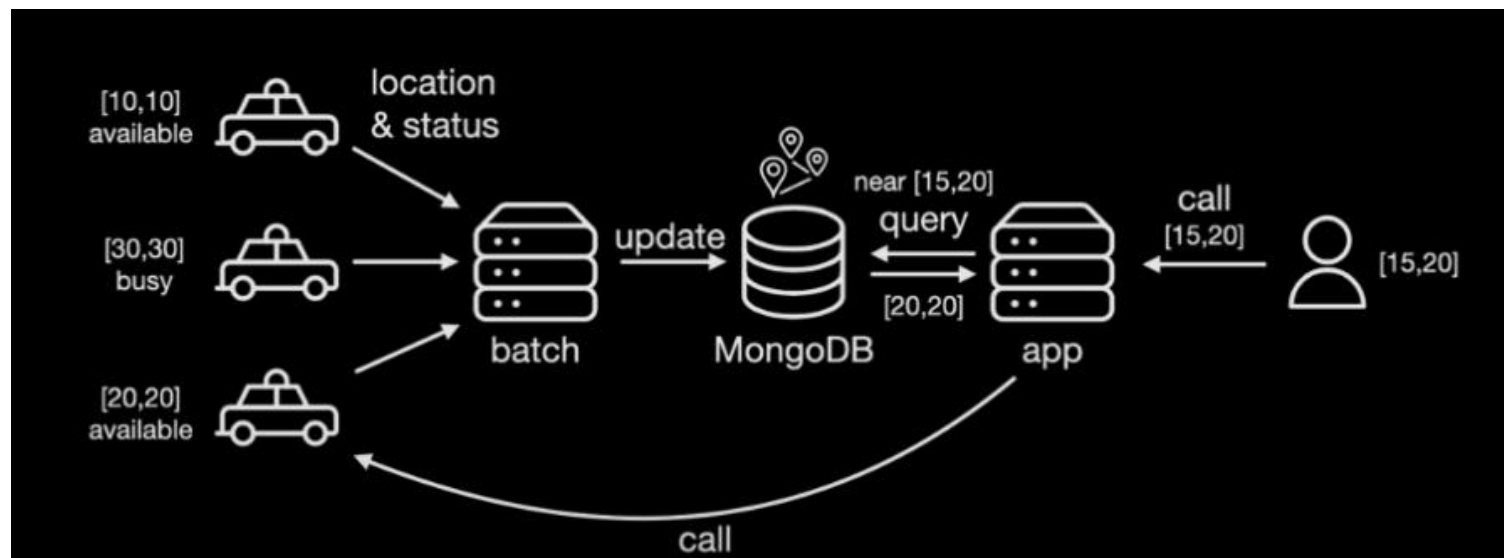
[Read Case Study](#)

[MongoDB for Manufacturing >](#)

volvo와 카카오



사용사례 및 기업



카카오 모빌리티: 복합인덱스를 이용

| 퀴즈1



- 몽고디비의 특징 2가지?

Part 2

- 몽고디비 구조 및 용어
- BSON이란?
- BSON의 데이터 타입

몽고디비 구조



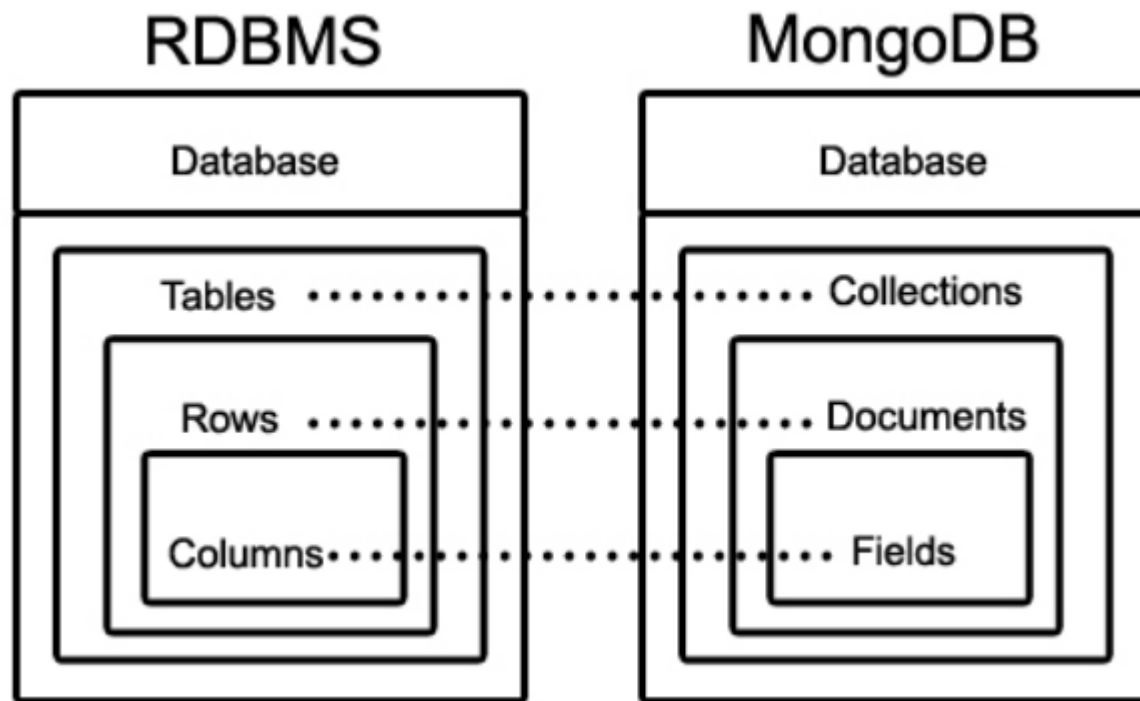
- Document가 모여서 Collection이 되고, Collection들이 모여서 Database를 이룹니다.





몽고디비의 용어

- Table은 MongoDB에서는 Collection
- Tuple/Row은 MongoDB에서는 Document라고 부릅니다.





몽고디비 구조

MongoDB Compass - localhost:27017/local.startup_log

Connect View Collection Help

localhost:27017 Documents local.startup_log

My Queries local.startup_log 1 DOCUMENTS 1 INDEXES

데이터베이스 (database)

admin

config

local

startup_log

컬렉션 (collection)

Filter Type a query: { field: 'value' } Reset Find More Options

ADD DATA EXPORT COLLECTION

1-1 of 1

```
{
  "_id": "PC051-1672082475362",
  "hostname": "PC051",
  "startTime": "2022-12-26T19:21:15.000+00:00",
  "startTimeLocal": "Tue Dec 27 04:21:15.362",
  "cmdLine": Object,
  "pid": 12624,
  "buildinfo": Object
}
```

도큐먼트 (문서, document)

> MONGOSH

BSON이란?



- Binary JSON(JavaScript Object Notation)의 의미 JSON의 일부지만 MongoDB에서 JSON으로는 부족함이 있어, 데이터를 저장하기 위한 **BSON**을 만든다.

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```

← field: value
← field: value
← field: value
← field: value

BSON의 데이터 타입



Type	Number	Alias
Double	1	"double"
String	2	"string"
Object	3	"object"
Array	4	"array"
Binary data	5	"binData"
Undefined	6	"undefined"
ObjectId	7	"objectId"
Boolean	8	"bool"
Date	9	"date"
Null	10	"null"



BSON의 데이터 타입

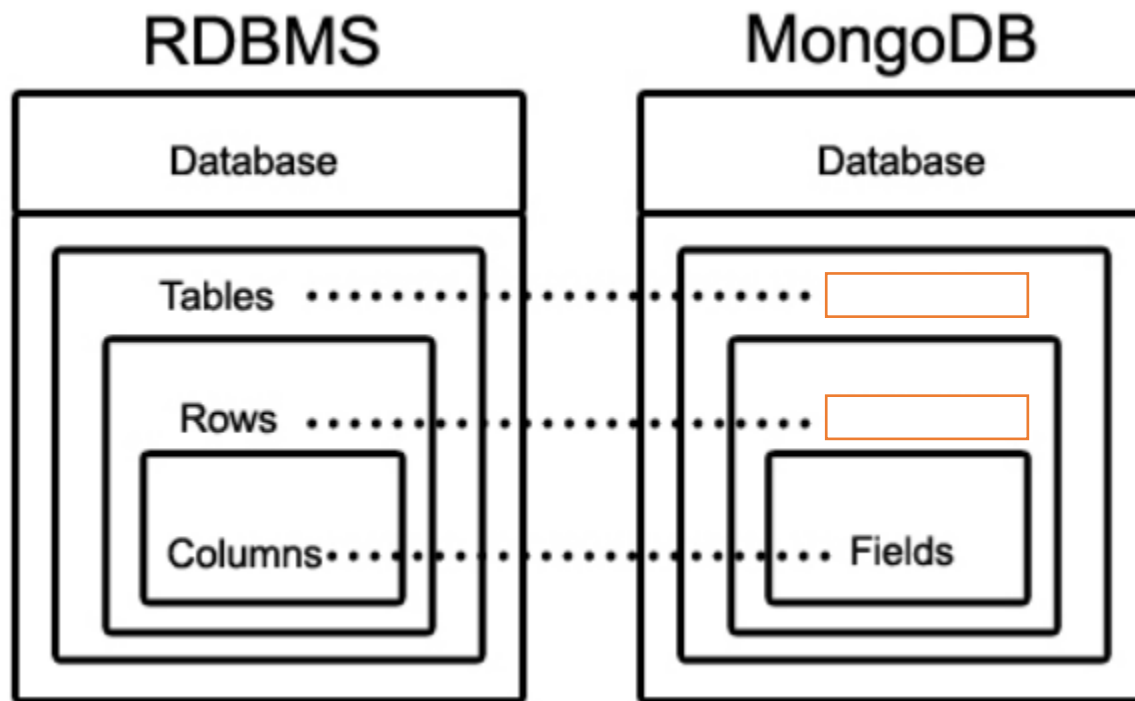
Type	Number	Alias
Regular Expression	11	"regex"
DBPointer	12	"dbPointer"
JavaScript	13	"javascript"
Symbol	14	"symbol"
JavaScript (with scope)	15	"javascriptWithScope"
32-bit integer	16	"int"
Timestamp	17	"timestamp"
64-bit integer	18	"long"
Decimal128	19	"decimal"
Min key	-1	"minkey"
Max key	127	"maxkey"



많이 사용되는 데이터 타입

타입	종류	예시
Null	null	null
정수형	int, long	1, 1004
실수형	double	1.24, 3.14
문자형	string	"kkoma", "developer"
Object	object	{field: 'value', number: 1}
Array	array	[1, 3.14, null, {x: 1}, true]
Boolean	boolean	true, false

퀴즈2









Part 3

- 몽고디비 설치


몽고디비 설치



 mongoDB     

[🔍 전체](#) [🖼️ 이미지](#) [📰 뉴스](#) [📺 동영상](#) [📍 지도](#) [⋮ 더보기](#) [도구](#)

검색결과 약 110,000,000개 (0.37초)

 **mongodb.com**
<https://www.mongodb.com>

MongoDB: The Developer Data Platform | MongoDB

Get your ideas to market faster with a developer data platform built on the leading modern database. **MongoDB** makes working with data easy.

[Documentation](#)
Manual - Drivers - Get Started with Atlas - Tools and Connectors - ...

[Log in to your account](#)
Or with email and password. Email Address. Next. Discover What's ...


[Community Edition](#)
Try MongoDB Community Edition. The community version of our ...

[애플리케이션 데이터 플랫폼](#)
업계 최고의 최신 데이터베이스를 토대로 구축된 애플리케이션 데 ...

[mongodb.com](#) 검색결과 더보기 »

몽고DB (MongoDB)

소프트웨어



몽고DB는 크로스 플랫폼 문서 지향 데이터베이스 시스템이다. NoSQL 데이터베이스로 분류되는 몽고DB는 JSON과 같은 동적 스키마형 문헌들을 선호함에 따라 전통적인 테이블 기반 관계형 데이터베이스 구조의 사용을 삼간다. 위키백과

프로그래밍 언어: 자바스크립트, 파이썬, 자바, Go, PHP, C, C++, 펄, 루비

개발자: 몽고DB

라이선스: SSPL

안정화 버전: 6.0.5 / 2023년 3월 13일 (2개월 전)

몽고디비 설치



이벤트 마지막 호출! 5월 12일까지 MongoDB.local NYC에 등록하면 등록 비용을 50% 절약할 수 있습니다. 자세히 알아보기 >

MongoDB. Products제품 Solutions솔루션 Resources자원 Company회사 Pricing가격

Try Free무료로 체험하다

Sign In로그인

Atlas아틀라스 → Developer data platform개발자 데이터 플랫폼

Database데이터 베이스

Search찾다

Data Lake 데이터 레이 (Preview) 크(미리 보기)

Charts차트

Device Sync장치 동기화

APIs, Triggers, API, 트리 Functions 거, 함수

Enterprise 엔터프라이즈 Advanced 고급 → Enterprise software and support엔터프라이즈 소프트웨어 및 지원

Enterprise 엔터프라이즈 Server 서버

Ops Manager운영 관리자

Enterprise 엔터프라이즈 Kubernetes Kubernetes Operator 운영자

Community 커뮤니티 Edition 에디션 → Free software used by millions수백만 명이 사용하는 무료 소프트웨어

Community 커뮤니티 Server 서버

Cloud 클라우드 관리자

Community 커뮤니티 Kubernetes Kubernetes Operator 운영자

Tools도구 → Build faster더 빠르게 구축

Compass나침반

Shell셸데기

VS Code VS 코드 플러그인

Atlas CLI아틀라스 CLI

Database 데이터베이스 Connectors 스 커넥터

Cluster-to-Cluster Sync 클러스터 간 동기화

Mongoose 몽구스 ODM Support ODM 지원

Relational 관계형 데이터 마이그레이션

+srv://cluster0.ab123.mongodb.net/

america-west1

남동부

america-northeast2

america-northeast2

asia-south2

neast2

부2

p-northeast-3

est-1

- us-gov-east-1

문의하기

몽고디비 설치

[Products제품](#)[Solutions솔루션](#)[Resources자원](#)[Company회사](#)[Pricing가격](#)[Sign In로그인](#)[Try Free무료로 체험하다](#)[MongoDB Atlas몽고디비 아틀라스](#)[MongoDB Enterprise Advanced몽고DB 엔터프라이즈 어드밴스드](#)[MongoDB Community EditionMongoDB 커뮤니티 에디션](#)[MongoDB Community ServerMongoDB 커뮤니티 서버](#)[MongoDB Community Kubernetes OperatorMongoDB 커뮤니티 Kubernetes 운영자](#)[Tools도구](#)[Atlas SQL Interface아틀라스 SQL 인터페이스](#)[Mobile & Edge모바일 및 엣지](#)[MONGODB COMMUNITY SERVERMONGODB 커뮤니티 서버](#)

MongoDB Community Server DownloadMongoDB 커뮤니티 서버 다운로드

The Community version of our distributed database offers a flexible document data model along with support for ad-hoc queries, secondary indexing, and real-time aggregations to provide powerful ways to access and analyze your data. 분산 데이터베이스의 커뮤니티 버전은 데이터에 액세스하고 분석하는 강력한 방법을 제공하기 위해 임시 쿼리, 보조 인덱싱 및 실시간 집계에 대한 지원과 함께 유연한 문서 데이터 모델을 제공합니다.

The database is also offered as a fully-managed service with [몽고디비 아틀라스](#). Get access to advanced functionality such as auto-scaling, serverless instances, full-text search, and data distribution across regions and clouds. Deploy in minutes on AWS, Google Cloud, and/or Azure, with no downloads necessary.

[가용성이 높은 무료 512MB 클러스터로 사용해 보십시오.](#) or get started from your terminal with the following two commands:

```
$ brew install mongodb - 아틀라스
$ 아틀라스 설정
```



몽고디비 설치



MongoDB. Products제품 Solutions솔루션 Resources자원 Company회사 Pricing가격

Try Free무료로 체험하다

몽고디비 아틀라스

몽고DB 엔터프라이즈 어드밴스드

MongoDB 커뮤니티 에디션

MongoDB 커뮤니티 서버

MongoDB 커뮤니티
Kubernetes 운영자

도구

아틀라스 SQL 인터페이스

모바일 및 예지

몽고디비 아틀라스

가용성이 높은 무료 512MB 클러스터로 사용해 보십시오.

```
$ brew install mongodb - 아틀라스
$ 아틀라스 설정
```

Version버전
6.0.6 (current)6.0.6(현재)

Platform플랫폼
Windows윈도우

Package패키지
msmsi

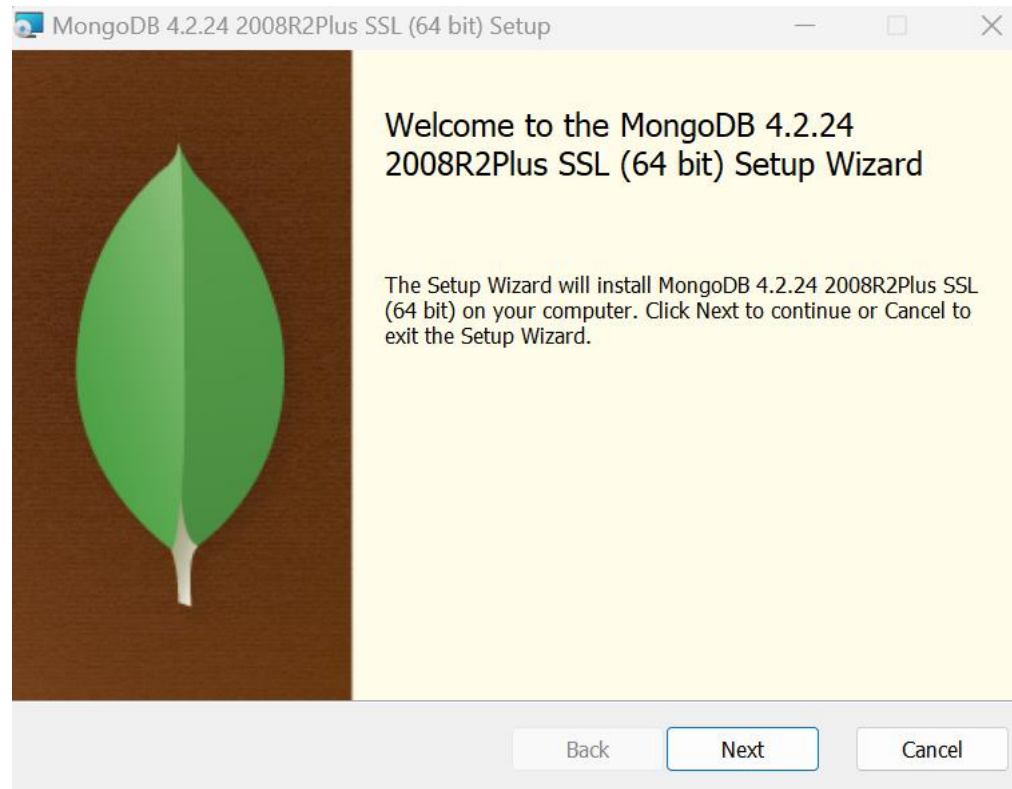
Download다운로드

Copy link링크 복사

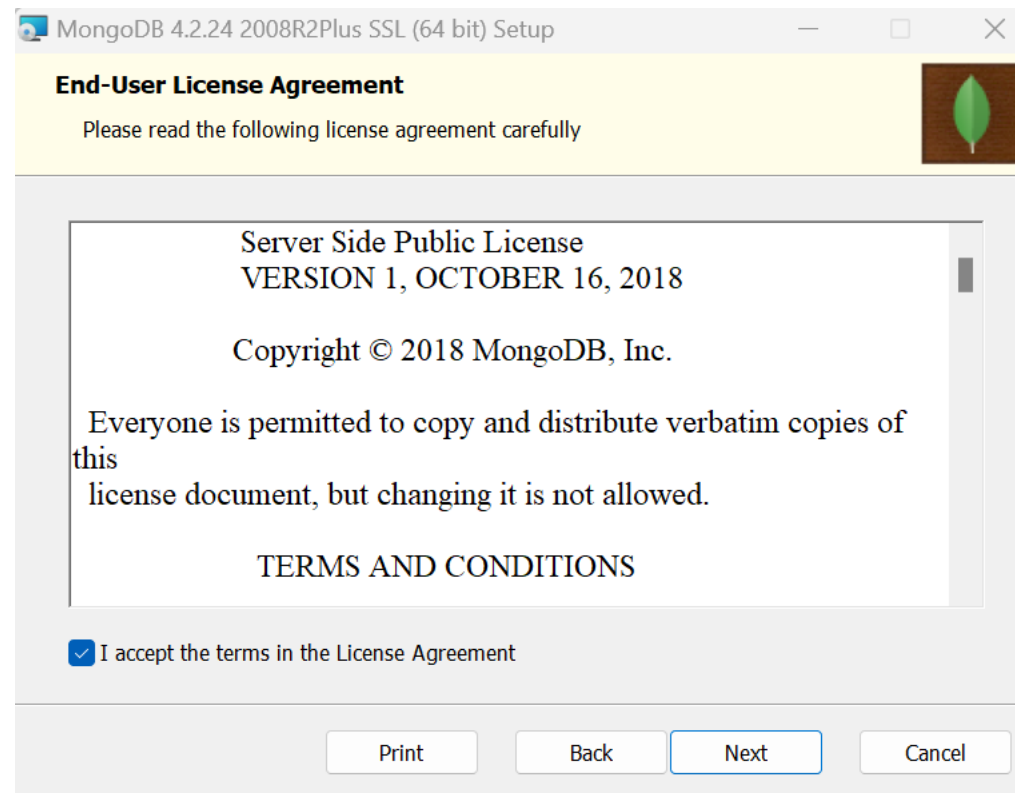
More
Options

추가 옵션

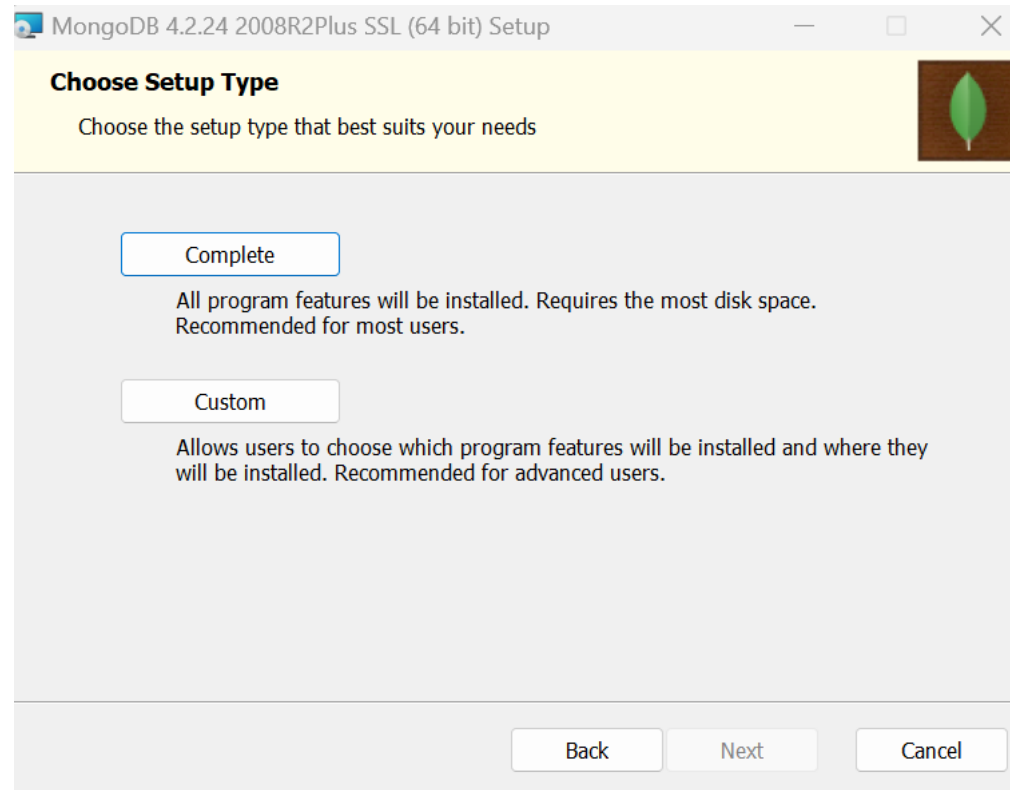
몽고디비 설치



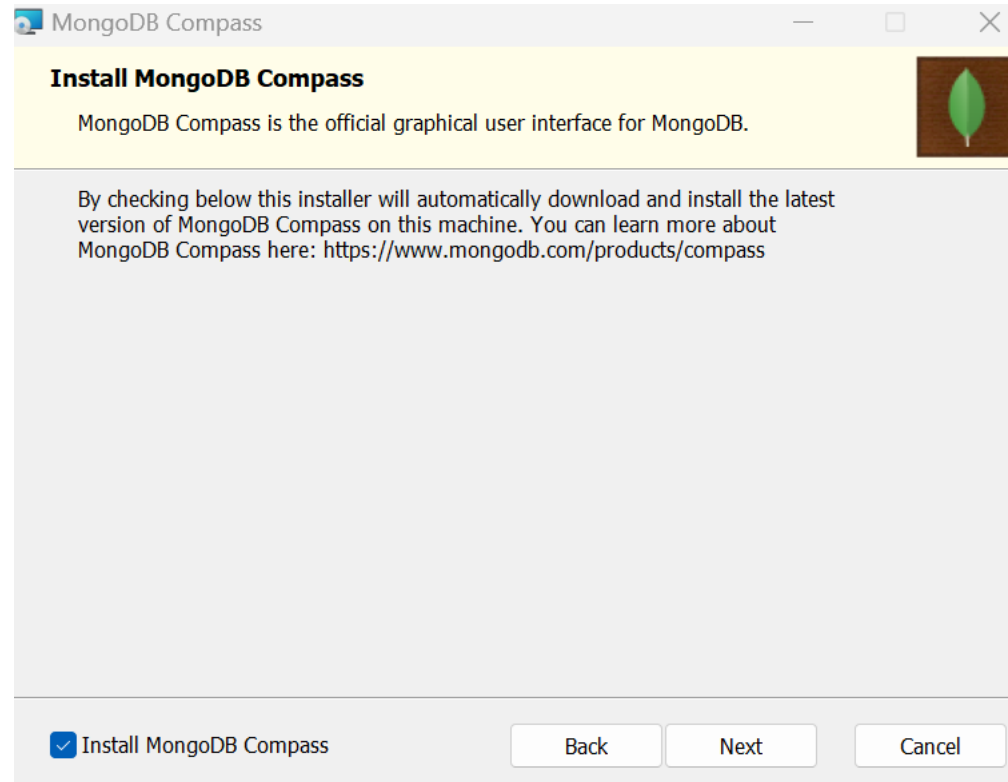
몽고디비 설치



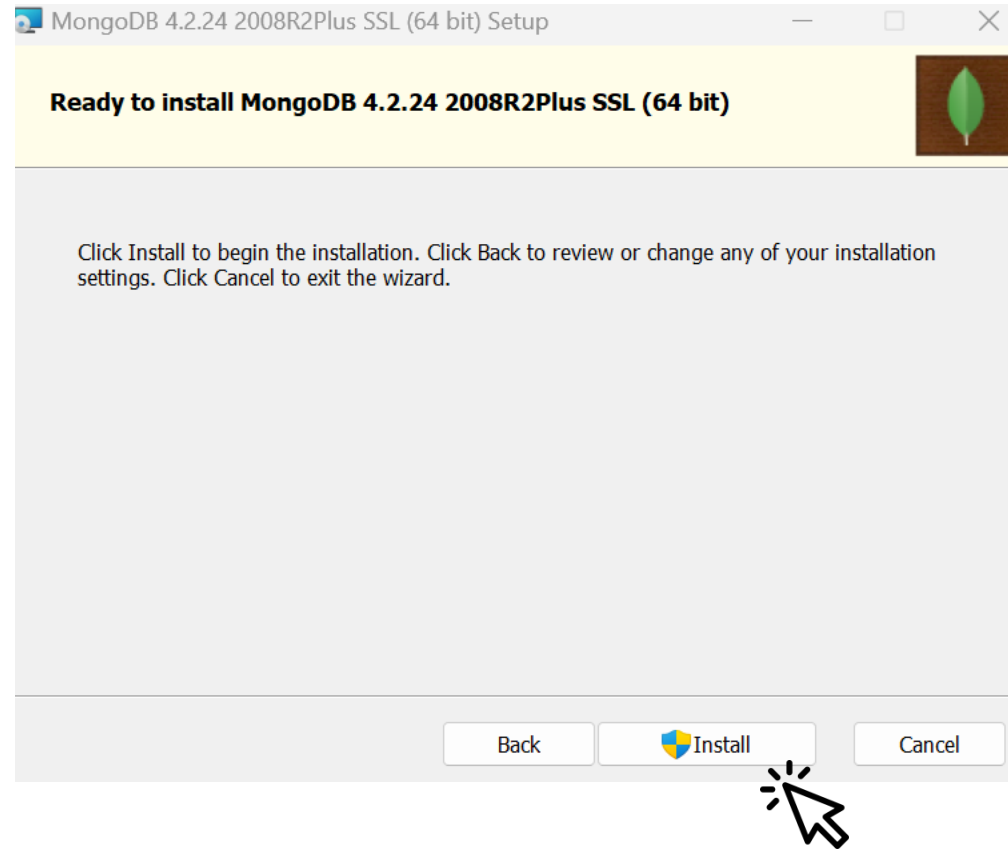
몽고디비 설치



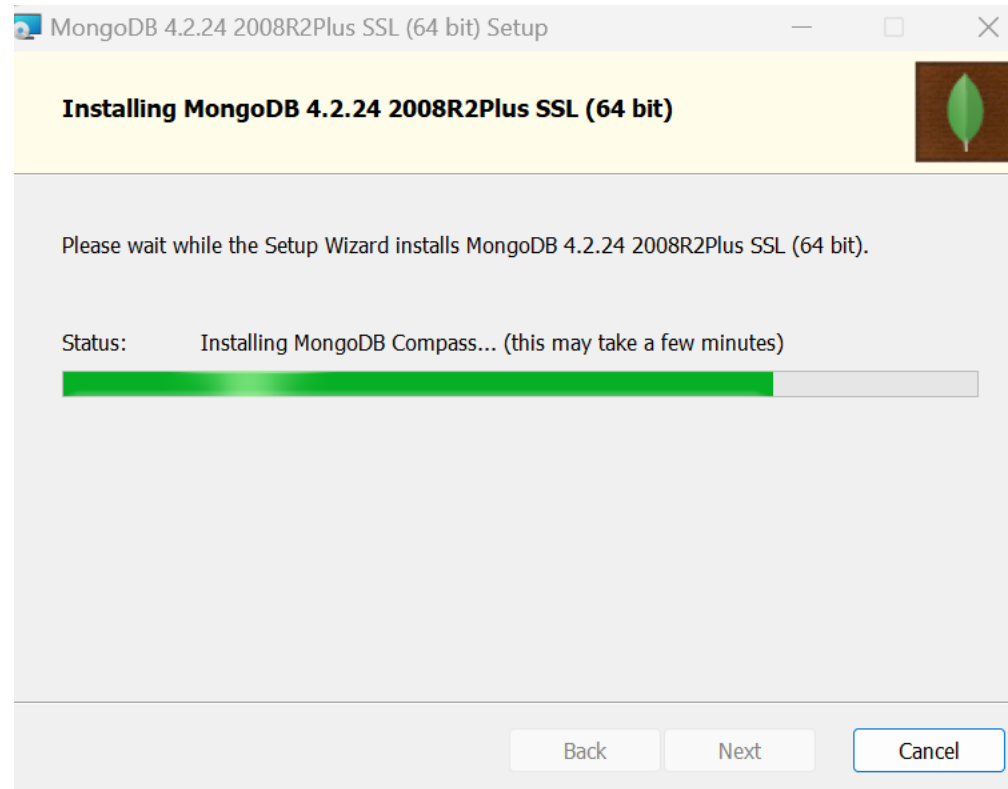
몽고디비 설치



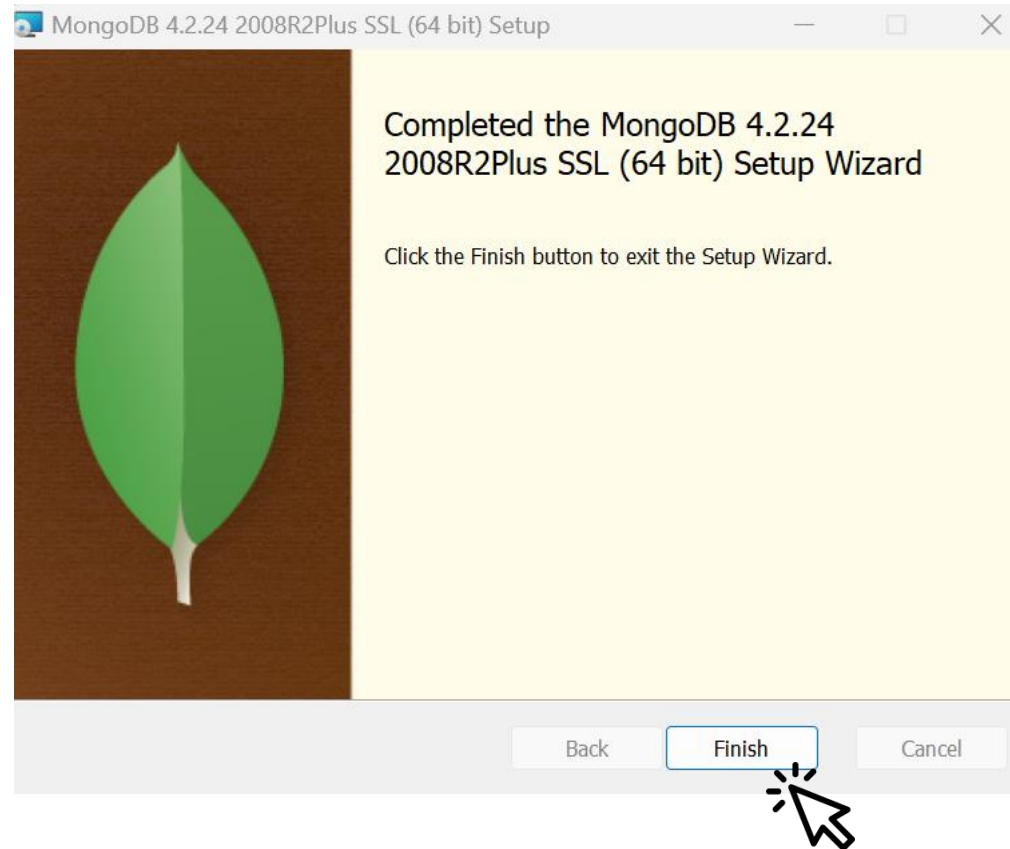
몽고디비 설치



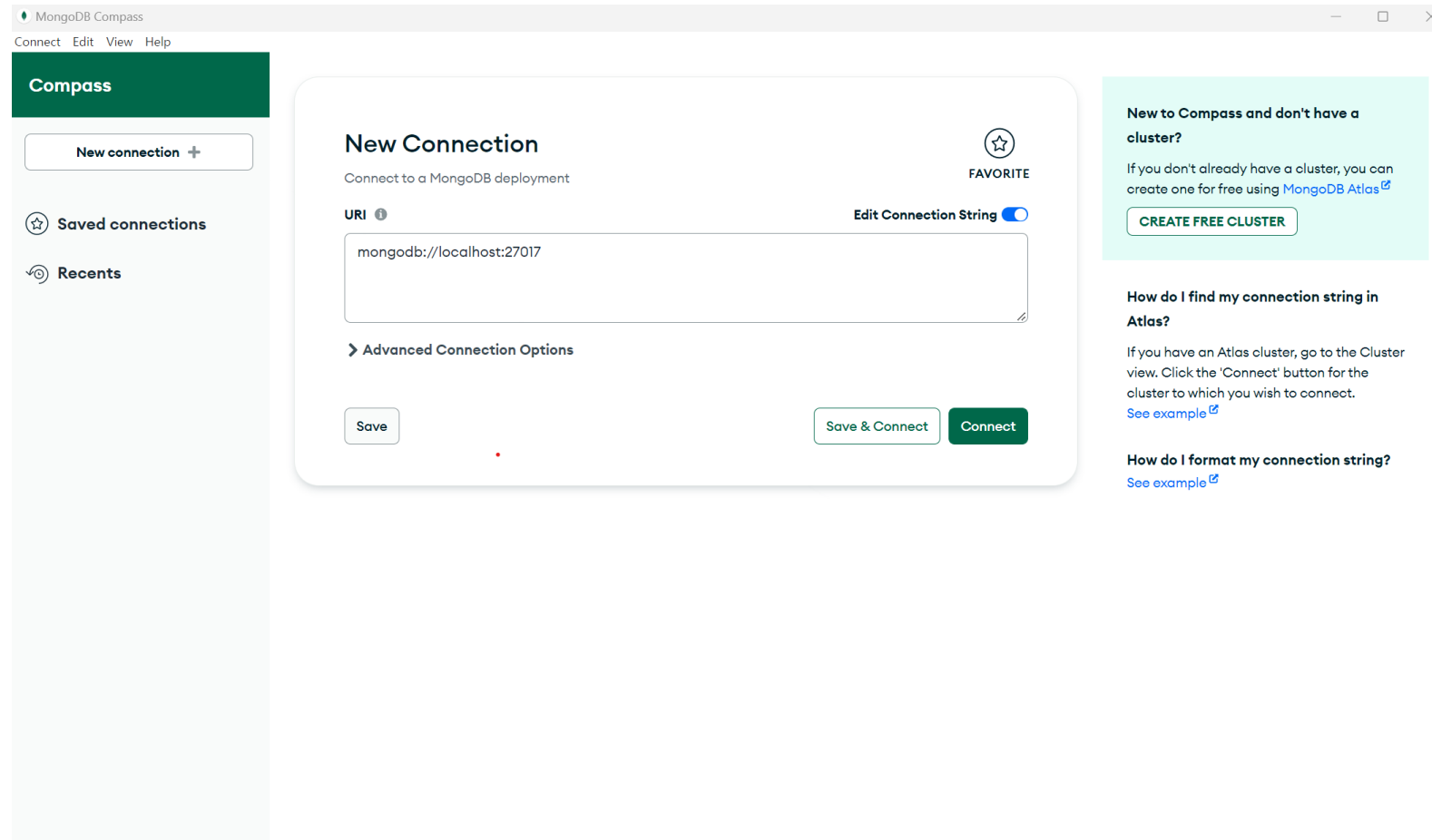
몽고디비 설치



몽고디비 설치



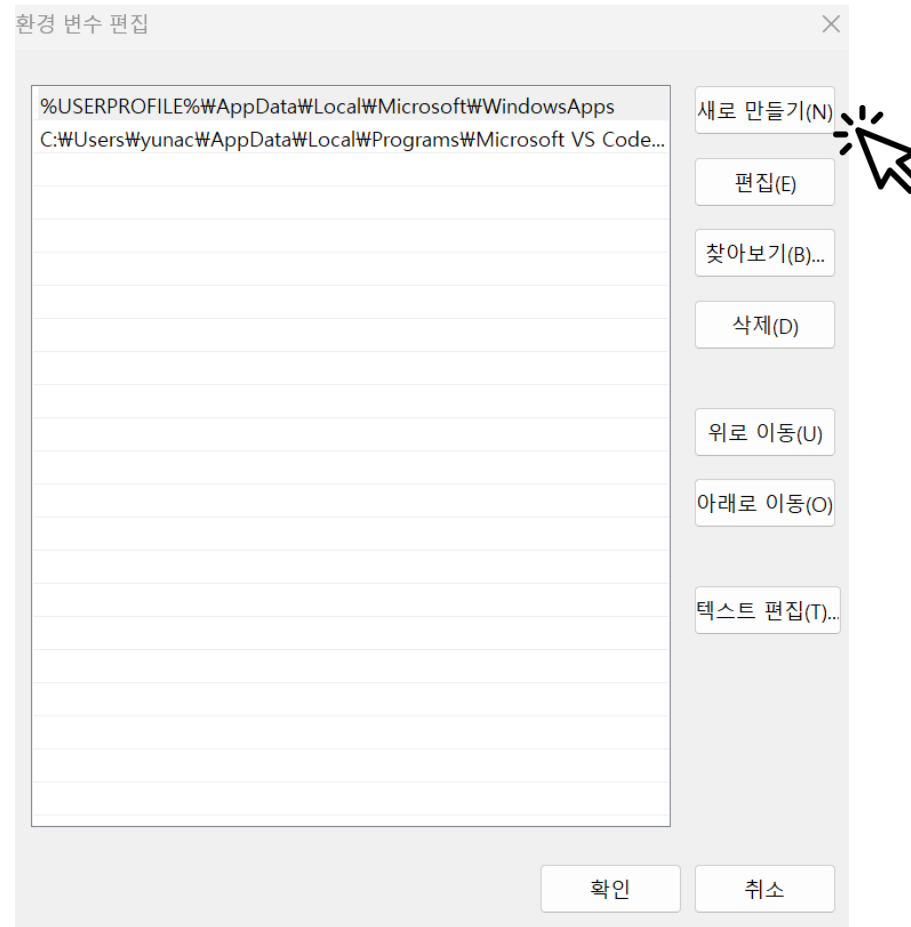
몽고디비 설치



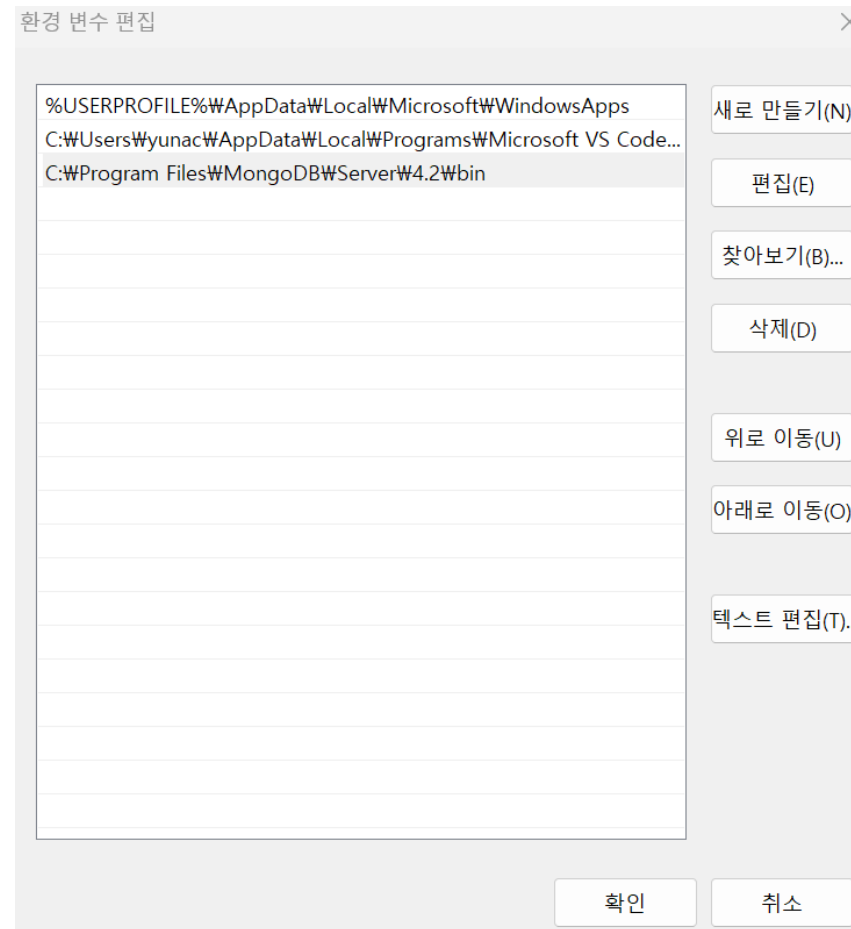
몽고디비 설치



몽고디비 설치



몽고디비 설치



몽고디비 설치



```
명령 프롬프트
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\yunac>mongod
2023-05-16T06:54:34.529+0900 I CONTROL [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDi
sabledProtocols 'none'
2023-05-16T06:54:34.885+0900 W ASIO [main] No TransportLayer configured during NetworkInterface startup
2023-05-16T06:54:34.887+0900 I CONTROL [initandlisten] MongoDB starting : pid=14484 port=27017 dbpath=C:\data\db\ 64-b
it host=kwonyunagram
2023-05-16T06:54:34.887+0900 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2023-05-16T06:54:34.887+0900 I CONTROL [initandlisten] db version v4.2.24
2023-05-16T06:54:34.887+0900 I CONTROL [initandlisten] git version: 5e4ec1d24431fcdd28b579a024c5c801b8cde4e2
2023-05-16T06:54:34.887+0900 I CONTROL [initandlisten] allocator: tcmalloc
2023-05-16T06:54:34.887+0900 I CONTROL [initandlisten] modules: none
2023-05-16T06:54:34.887+0900 I CONTROL [initandlisten] build environment:
2023-05-16T06:54:34.887+0900 I CONTROL [initandlisten] distmod: 2012plus
2023-05-16T06:54:34.887+0900 I CONTROL [initandlisten] distarch: x86_64
2023-05-16T06:54:34.887+0900 I CONTROL [initandlisten] target_arch: x86_64
2023-05-16T06:54:34.887+0900 I CONTROL [initandlisten] options: {}
2023-05-16T06:54:34.888+0900 I STORAGE [initandlisten] exception in initAndListen: NonExistentPath: Data directory C:\
data\db\ not found. Create the missing directory or specify another path using (1) the --dbpath command line option, or
(2) by adding the 'storage.dbPath' option in the configuration file., terminating
2023-05-16T06:54:34.888+0900 I REPL [initandlisten] Stepping down the ReplicationCoordinator for shutdown, waitTime
: 10000ms
2023-05-16T06:54:34.888+0900 I SHARDING [initandlisten] Shutting down the WaitForMajorityService
2023-05-16T06:54:34.888+0900 I NETWORK [initandlisten] shutdown: going to close listening sockets...
2023-05-16T06:54:34.888+0900 I NETWORK [initandlisten] Shutting down the global connection pool
2023-05-16T06:54:34.888+0900 I STORAGE [initandlisten] Shutting down the FlowControlTicketHolder
2023-05-16T06:54:34.888+0900 I - [initandlisten] Stopping further Flow Control ticket acquisitions.
2023-05-16T06:54:34.888+0900 I INDEX [initandlisten] Shutting down the IndexBuildsCoordinator
```

몽고디비 설치



```
명령 프롬프트 - mongo
2023-05-16T06:54:34.889+0900 I - [initandlisten] Dropping the scope cache for shutdown
2023-05-16T06:54:34.889+0900 I CONTROL [initandlisten] now exiting
2023-05-16T06:54:34.889+0900 I CONTROL [initandlisten] shutting down with code:100

C:\Users\yunac>

C:\Users\yunac>mongo
MongoDB shell version v4.2.24
connecting to: mongod://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongod
Implicit session: session { "id" : UUID("aa4a888d-9584-4b21-89f7-1ba21bfa3afb") }
MongoDB server version: 4.2.24
Server has startup warnings:
2023-05-14T17:43:30.017+0900 I CONTROL [initandlisten]
2023-05-14T17:43:30.017+0900 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2023-05-14T17:43:30.017+0900 I CONTROL [initandlisten] ** Read and write access to data and configuration is
unrestricted.
2023-05-14T17:43:30.019+0900 I CONTROL [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible 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.disableFreeMonitoring()
---
```


Part 4

- 몽고디비 실습

동덕여대 데사과 명단을 관리해보자



1. 동덕여대 데이터베이스를 만들기
(DB생성: use DATABASE_NAME)
2. 데이터사이언스전공 신설 및 10명의 학생이름 넣기
(컬렉션,도큐먼트 생성 :
db.datascience.insertOne/insertMany)
3. 학생 1명 자퇴
(도큐먼트 삭제 :deleteOne)
4. 교환학생을 간 마크와 편입한 Alice를 수정해보자
(데이터 수정 : replaceOne)
5. Alice의 이름을 한국이름인 예리로 변경해보자
(데이터 수정 : updateOne)
- 6.인덱스 생성,조회,삭제(createIndex,getIndexes,dropIndexes)



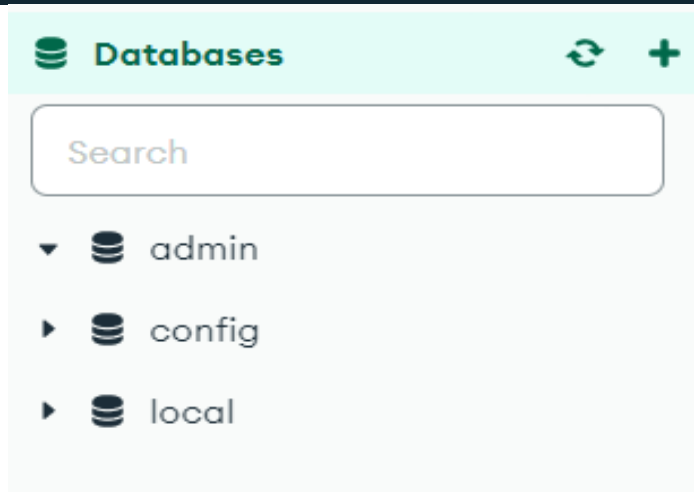
MongoDB 구동하기



1. dongduk_wuniversity 데이터 베이스 생성

-> use dongduk_wuniversity

```
> use dongduk_wuniversity  
< 'switched to db dongduk_wuniversity'  
dongduk_wuniversity>
```



MongoDB 구동하기



2-1 Datascience 컬렉션 생성 및 문서 하나 삽입

db.datascience.insertOne

- **insertOne** - 단일 Document를 입력할 때 사용
- **insertMany** - 다수의 Document를 입력할 때 사용

The screenshot displays the MongoDB Compass web interface. On the left, the 'Databases' sidebar shows the 'dongduk_wuniversity' database expanded, with the 'datascience' collection selected. The main panel shows the 'Documents' tab for the 'dongduk_wuniversity.datascience' collection, indicating 1 document and 1 index. A filter bar is present with a query field. Below the filter, there are buttons for 'ADD DATA' and 'EXPORT COLLECTION'. The document viewer shows a single document with the following structure:

```
{
  "_id": ObjectId('64622aa0066e72655dc93940'),
  "univ_num": "20221608",
  "name": "권유나",
  "age": 21,
  "etc": Array,
  "hobby": "유튜브 시청"
}
```

At the bottom, a terminal window titled '_MONGOSH' shows the following commands and output:

```
> use dongduk_wuniversity
< 'switched to db dongduk_wuniversity'
> db.datascience.insertOne({univ_num: "20221608", name: "권유나", age: 21, etc: [{hobby: "유튜브 시청"}]})
< {
  acknowledged: true,
  insertedId: ObjectId("64622aa0066e72655dc93940")
}
```

MongoDB 구동하기



2-2 생성된 데이터베이스 보여주기

show dbs

```
> show dbs
< admin          72.00 KiB
  config         108.00 KiB
  dongduk_wuniversity 40.00 KiB
  local          40.00 KiB
```

MongoDB 구동하기



2-3 **도큐먼트 여러 개 생성**

db.datascience.insertMany

```
> db.datascience.insertMany([ { univ_num: "20221612", name: "김서윤", age: 21, etc: [ { hobby: "잠자기" } ] },
  { univ_num: "20221635", name: "이유나", age: 22, etc: [ { hobby: "밴드 공연 보기" } ] },
  { univ_num: "20221655", name: "황진하", age: 21, etc: [ { hobby: "게임" } ] },
  { univ_num: "20220219", name: "이강인", age: 23, etc: [ { hobby: "축구" } ] },
  { univ_num: "20221234", name: "김영훈", age: 27, etc: [ { hobby: "출추기" } ] },
  { univ_num: "20220987", name: "마크", age: 25, etc: [ { hobby: "노래" } ] },
  { univ_num: "20221357", name: "김수지", age: 22, etc: [ { hobby: "넷플릭스" } ] },
  { univ_num: "20220875", name: "유리", age: 23, etc: [ { hobby: "헬스" } ] },
  { univ_num: "20223579", name: "철수", age: 21, etc: [ { hobby: "독서" } ] } ] )
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("646223bdf3896812e271468a"),
    '1': ObjectId("646223bdf3896812e271468b"),
    '2': ObjectId("646223bdf3896812e271468c"),
    '3': ObjectId("646223bdf3896812e271468d"),
    '4': ObjectId("646223bdf3896812e271468e"),
    '5': ObjectId("646223bdf3896812e271468f"),
    '6': ObjectId("646223bdf3896812e2714690"),
    '7': ObjectId("646223bdf3896812e2714691"),
    '8': ObjectId("646223bdf3896812e2714692")
  }
}
```

MongoDB 구동하기



```
_id: ObjectId('64622aa0066e72655dc93940')
univ_num: "20221608"
name: "권유나"
age: 21
▼ etc: Array
  ▼ 0: Object
    hobby: "유튜브 시청"
```

```
_id: ObjectId('64622b49066e72655dc93941')
univ_num: "20221612"
name: "김서윤"
age: 21
▼ etc: Array
  ▼ 0: Object
    hobby: "잠자기"
```

```
_id: ObjectId('64622b49066e72655dc93942')
univ_num: "20221635"
name: "이유나"
age: 22
▼ etc: Array
  ▼ 0: Object
    hobby: "밴드 공연 보기"
```

```
_id: ObjectId('64622b49066e72655dc93943')
univ_num: "20221655"
name: "황진하"
age: 21
▼ etc: Array
  ▼ 0: Object
    hobby: "게임"
```

```
_id: ObjectId('64622b49066e72655dc93944')
univ_num: "20220219"
name: "이강인"
age: 23
▼ etc: Array
  ▼ 0: Object
    hobby: "축구"
```

```
_id: ObjectId('64622b49066e72655dc93945')
univ_num: "20221234"
name: "김영훈"
age: 27
▼ etc: Array
  ▼ 0: Object
    hobby: "춤추기"
```

```
_id: ObjectId('64622b49066e72655dc93946')
univ_num: "20220987"
name: "마크"
age: 25
▼ etc: Array
  ▼ 0: Object
    hobby: "노래"
```

```
_id: ObjectId('64622b49066e72655dc93947')
univ_num: "20221357"
name: "김수지"
age: 22
▼ etc: Array
  ▼ 0: Object
    hobby: "넷플릭스"
```

```
_id: ObjectId('64622b49066e72655dc93948')
univ_num: "20220875"
name: "유리"
age: 23
▼ etc: Array
  ▼ 0: Object
    hobby: "헬스"
_id: ObjectId('64622b49066e72655dc93949')
univ_num: "20223579"
name: "철수"
age: 21
▼ etc: Array
  ▼ 0: Object
    hobby: "독서"
```

MongoDB 구동하기



데이터 조회 (find)

3.나이 21 찾기

db.datascience.find({age:21})

```
< {
  _id: ObjectId("646bd3db5bd55cd457bc0032"),
  univ_num: '20221608',
  name: '권유나',
  age: 21,
  etc: [
    {
      hobby: '유튜브 시청'
    }
  ]
}
{
  _id: ObjectId("646bd3f35bd55cd457bc0033"),
  univ_num: '20221612',
  name: '김서윤',
  age: 21,
  etc: [
    {
      hobby: '잠자기'
    }
  ]
}

{
  _id: ObjectId("64622b49066e72655dc93943"),
  univ_num: '20221655',
  name: '황진하',
  age: 21,
  etc: [
    {
      hobby: '게임'
    }
  ]
}
{
  _id: ObjectId("64622b49066e72655dc93949"),
  univ_num: '20223579',
  name: '철수',
  age: 21,
  etc: [
    {
      hobby: '독서'
    }
  ]
}
```


MongoDB 구동하기



데이터 조회 (find)

3-1 취미가 독서 찾기

db.datascience.find({etc:{\$elemMatch:{hobby:독서}}})

```
> db.datascience.find({etc:{$elemMatch:{hobby:"독서"}}})
< {
  _id: ObjectId("64622b49066e72655dc93949"),
  univ_num: '20223579',
  name: '철수',
  age: 21,
  etc: [
    {
      hobby: '독서'
    }
  ]
}
```

MongoDB 구동하기



데이터 조회 (find)

3-2 나이가 23이상 26이하 찾기

db.datascience.find({age:{\$gte:23,\$lte:26}})

```
> db.datascience.find({age:{$gte:23,$lte:26}})
< {
  _id: ObjectId("64622b49066e72655dc93944"),
  univ_num: '20220219',
  name: '이강인',
  age: 23,
  etc: [
    {
      hobby: '축구'
    }
  ]
}
{
  _id: ObjectId("64622b49066e72655dc93946"),
  univ_num: '20220987',
  name: '마크',
  age: 25,
  etc: [
    {
      hobby: '노래'
    }
  ]
}
```

```
{
  _id: ObjectId("64622b49066e72655dc93948"),
  univ_num: '20220875',
  name: '유리',
  age: 23,
  etc: [
    {
      hobby: '헬스'
    }
  ]
}
```

MongoDB 구동하기



```
> db.datascience.find({$and:[{age:{$gte:23}},{age:{$lte:26}}]})
```

```
< {
  _id: ObjectId("64622b49066e72655dc93944"),
  univ_num: '20220219',
  name: '이강인',
  age: 23,
  etc: [
    {
      hobby: '축구'
    }
  ]
}
{
  _id: ObjectId("64622b49066e72655dc93946"),
  univ_num: '20220987',
  name: '마크',
  age: 25,
  etc: [
    {
      hobby: '노래'
    }
  ]
}
```

데이터 조회 (find)

3-3 and 이용해서 3-2와 똑같은 결과 나오게 하기

```
db.datascience.find({$and:[{age:{$gte:23}},{age:{$lte:26}}]})
```

```
{
  _id: ObjectId("64622b49066e72655dc93948"),
  univ_num: '20220875',
  name: '유리',
  age: 23,
  etc: [
    {
      hobby: '헬스'
    }
  ]
}
```

MongoDB 구동하기



```
> db.datascience.find({name: {$regex: "^김"}})
< {
  _id: ObjectId("64622b49066e72655dc93941"),
  univ_num: '20221612',
  name: '김서윤',
  age: 21,
  etc: [
    {
      hobby: '잠자기'
    }
  ]
}
{
  _id: ObjectId("64622b49066e72655dc93945"),
  univ_num: '20221234',
  name: '김영훈',
  age: 27,
  etc: [
    {
      hobby: '충추기'
    }
  ]
}
```

데이터 조회 (find)

3-4 성이 김씨인 사람 찾기

`db.datascience.find({name: {$regex: "^김"}})`

```
{
  _id: ObjectId("64622b49066e72655dc93947"),
  univ_num: '20221357',
  name: '김수지',
  age: 22,
  etc: [
    {
      hobby: '넷플릭스'
    }
  ]
}
```

MongoDB 구동하기



4 김수지 자퇴해서 삭제
db.datascience.deleteOne

```
> db.datascience.deleteOne({name:"김수지"})  
< {  
  acknowledged: true,  
  deletedCount: 1  
}
```

MongoDB 구동하기



+ ADD DATA ▾

EXPORT COLLECTION

```
_id: ObjectId('64622aa0066e72655dc93940')
univ_num: "20221608"
name: "권유나"
age: 21
▶ etc: Array
```

```
_id: ObjectId('64622b49066e72655dc93941')
univ_num: "20221612"
name: "김서윤"
age: 21
▶ etc: Array
```

```
_id: ObjectId('64622b49066e72655dc93942')
univ_num: "20221635"
name: "이유나"
age: 22
▶ etc: Array
```

```
_id: ObjectId('64622b49066e72655dc93943')
univ_num: "20221655"
name: "황진하"
age: 21
▶ etc: Array
```

```
_id: ObjectId('64622b49066e72655dc93944')
univ_num: "20220219"
name: "이강인"
age: 23
▶ etc: Array
```

```
_id: ObjectId('64622b49066e72655dc93945')
univ_num: "20221234"
name: "김영훈"
age: 27
▶ etc: Array
```

```
_id: ObjectId('64622b49066e72655dc93946')
univ_num: "20220987"
name: "마크"
age: 25
▶ etc: Array
```

```
_id: ObjectId('64622b49066e72655dc93948')
univ_num: "20220875"
name: "유리"
age: 23
▶ etc: Array
```

```
_id: ObjectId('64622b49066e72655dc93949')
univ_num: "20223579"
name: "철수"
age: 21
▶ etc: Array
```

MongoDB 구동하기



5-1

마크가 교환학생을 가서 과를 떠나고 그 자리에 Alice가 데사과로

1) 마크를 Alice로 교체 : **replaceOne**

```
> db.datascience.replaceOne({name:"마크"},{univ_num:"20221010",name:"Alice",age:29})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
_id: ObjectId('64622910e45b123891cb642b')
univ_num: "20220987"
name: "마크"
age: 25
etc: Array
  0: Object
    hobby: "노래"
```

```
_id: ObjectId('64622910e45b123891cb642b')
univ_num: "20221010"
name: "Alice"
age: 29
```

MongoDB 구동하기



5-1

2) Alice에 배열 럭비 추가

```
> db.datascience.updateOne({name:"Alice"},{$addToSet:{etc":{"hobby":"럭비"}}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
_id: ObjectId('64622910e45b123891cb642b')
univ_num: "20221010"
name: "Alice"
age: 29
▼ etc: Array
  ▼ 0: Object
    hobby: "럭비"
```


MongoDB 구동하기



5-2 etc배열에 거주지 배열 추가

```
> db.datascience.updateOne({name:"김서윤"},{$push:{"etc":{"residence":["서울시 양천구"]}}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
_id: ObjectId('64622910e45b123891cb6426')
univ_num: "20221612"
name: "김서윤"
age: 21
etc: Array
  0: Object
    hobby: "잠자기"
```

```
_id: ObjectId('64622910e45b123891cb6426')
univ_num: "20221612"
name: "김서윤"
age: 21
etc: Array
  0: Object
    hobby: "잠자기"
  1: Object
    residence: Array
      0: "서울시 양천구"
```

MongoDB 구동하기



5-2-1 거주지 배열 삭제

```
> db.datascience.updateOne({name:"김서윤"},{$pull:{"etc":{"residence":"서울시 양천구"}}})  
< {  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}
```

MongoDB 구동하기



5-3 Alice이름을 예리로 수정 (Alice가 한국와서)

```
> db.datascience.updateOne({name:"Alice"},{$set:{name:"예리"}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
_id: ObjectId('64622910e45b123891cb642b')
univ_num: "20221010"
name: "예리"
age: 29
▼ etc: Array
  ▼ 0: Object
    hobby: "러닝"
```

MongoDB 구동하기



5-4. 취미 바꾸기

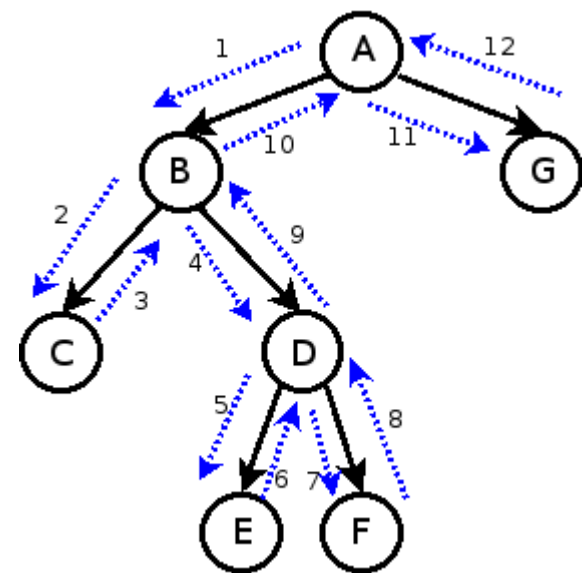
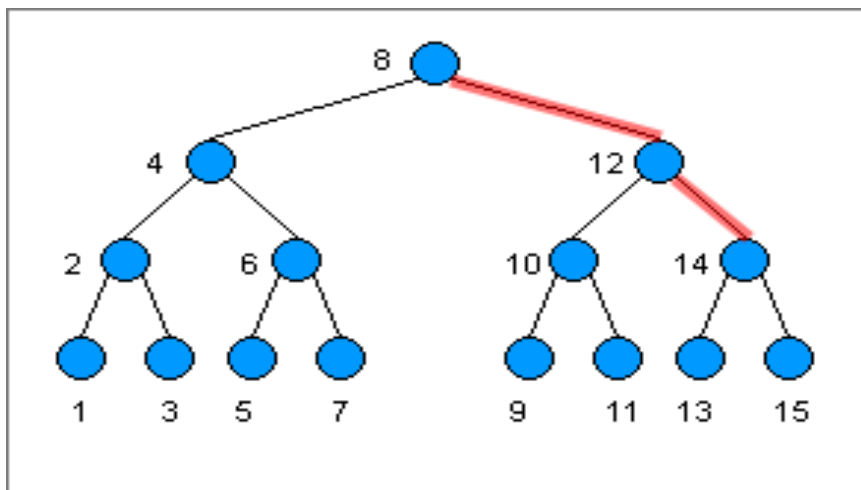
```
> db.datascience.updateOne( {name: "예리"}, {$set: {"etc.$[mod].hobby": "축구"}}, {arrayFilters: [{"mod.hobby": "러비"}]})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
_id: ObjectId('64622910e45b123891cb642b')
univ_num: "20221010"
name: "예리"
age: 29
▼ etc: Array
  ▼ 0: Object
    hobby: "축구"
```

인덱스란?



- Index는 MongoDB에서 데이터 쿼리를 더욱 효율적으로 이용할 수 있게 해줍니다.
- 인덱스를 사용하면 더 작은 횟수의 조회로 원하는 데이터를 찾을 수 있습니다.





인덱스란?

1. 인덱스 생성

- 인덱스를 생성할 땐 `createIndex()` 메소드를 사용
- 파라미터는 인덱스를 적용할 필드를 전달
- 값이 1일때 오름차순으로, -1일때 내림차순으로 정렬

2. text 인덱스 설정

- 인덱스를 지정할 때 +1, -1 대신 `text`라고 넣는다.
- 이 경우 `$text` 쿼리 연산자를 사용할 수 있다.



인덱스란?

3. 생성된 인덱스 조회

- 생성된 인덱스를 조회할 땐 `getIndexes()` 메소드를 사용

4. 인덱스 삭제

- 특정 인덱스를 제거하려면 `db.datascience.dropIndex(필드이름);`
- 모두 제거하려면 `db.datascience.dropIndexes();`

MongoDB 구동하기



6-1. 인덱스 생성 (text인덱스 설정)

```
> db.datascience.createIndex({name:"text"})  
< 'name_text'
```

```
> db.datascience.find({$text: {$search: "이강인"}})  
< {  
  _id: ObjectId("64621938a56761a70ddc9aa3"),  
  univ_num: '20220219',  
  name: '이강인',  
  age: 23,  
  etc: [  
    {  
      hobby: '축구'  
    }  
  ]  
}
```


MongoDB 구동하기



name필드에 대해 오름차순1 정렬

```
> db.datascience.createIndex({ name: 1 })  
< 'name_1'
```

내림차순 -1 정렬

```
> db.datascience.createIndex({ name: -1 })  
< 'name_-1'
```

MongoDB 구동하기



6-2 인덱스 조회 및 삭제

```
> db.datascience.getIndexes()
< [
  { v: 2, key: { _id: 1 }, name: '_id_' },
  {
    v: 2,
    key: { _fts: 'text', _ftsx: 1 },
    name: 'name_text',
    weights: { name: 1 },
    default_language: 'english',
    language_override: 'language',
    textIndexVersion: 3
  },
  { v: 2, key: { name: 1 }, name: 'name_1' },
  { v: 2, key: { name: -1 }, name: 'name_-1' }
]
```

```
> db.datascience.dropIndexes()
< {
  nIndexesWas: 4,
  msg: 'non-_id indexes dropped for collection',
  ok: 1
}
```



퀴즈3

- 도큐먼트 하나 넣을 때는 insertOne이다.
여러 개 넣을 때는?
insert___ (빈칸)

출처



<https://hoing.io/archives/1379>

데이터베이스개론(김연희, 한빛아카데미, 2022)

<https://sjh836.tistory.com/98>

<https://www.mongodb.com/ko-kr/evolved>

<https://zdnet.co.kr/view/?no=20221019135141>

https://www.dt.co.kr/contents.html?article_no=2019090902101431650001

<https://survey.stackoverflow.co/2022/>

<https://tv.kakao.com/channel/3693125/cliplink/414072595>

<http://www.koit.co.kr>

<https://www.mongodb.com/ko-kr>

<https://penguigoon.tistory.com/246>

<https://blog.naver.com/ehdwns3487/222248910517>

<https://league-cat.tistory.com/24>

<https://blog.naver.com/efforthye/223066920630>

<https://penguigoon.tistory.com/246>

<https://blog.naver.com/ehdwns3487/222248910517>

<https://league-cat.tistory.com/24>

<https://velopert.com/560>

<https://kay-log.tistory.com/24>

<http://https://www.youtube.com/watch?v=VcgEkIEReJk&t=54s>

Slides

THANK YOU