MongoDB

차례

1. MongoDB 개요
2. Mongo 셸 환경에서 CRUD 사용
3. 쿼리 작성
4. Index
5. 데이터 모델링
6. Sharding
7. 맵리듀스
8. aggregate
9. mongoDB활용

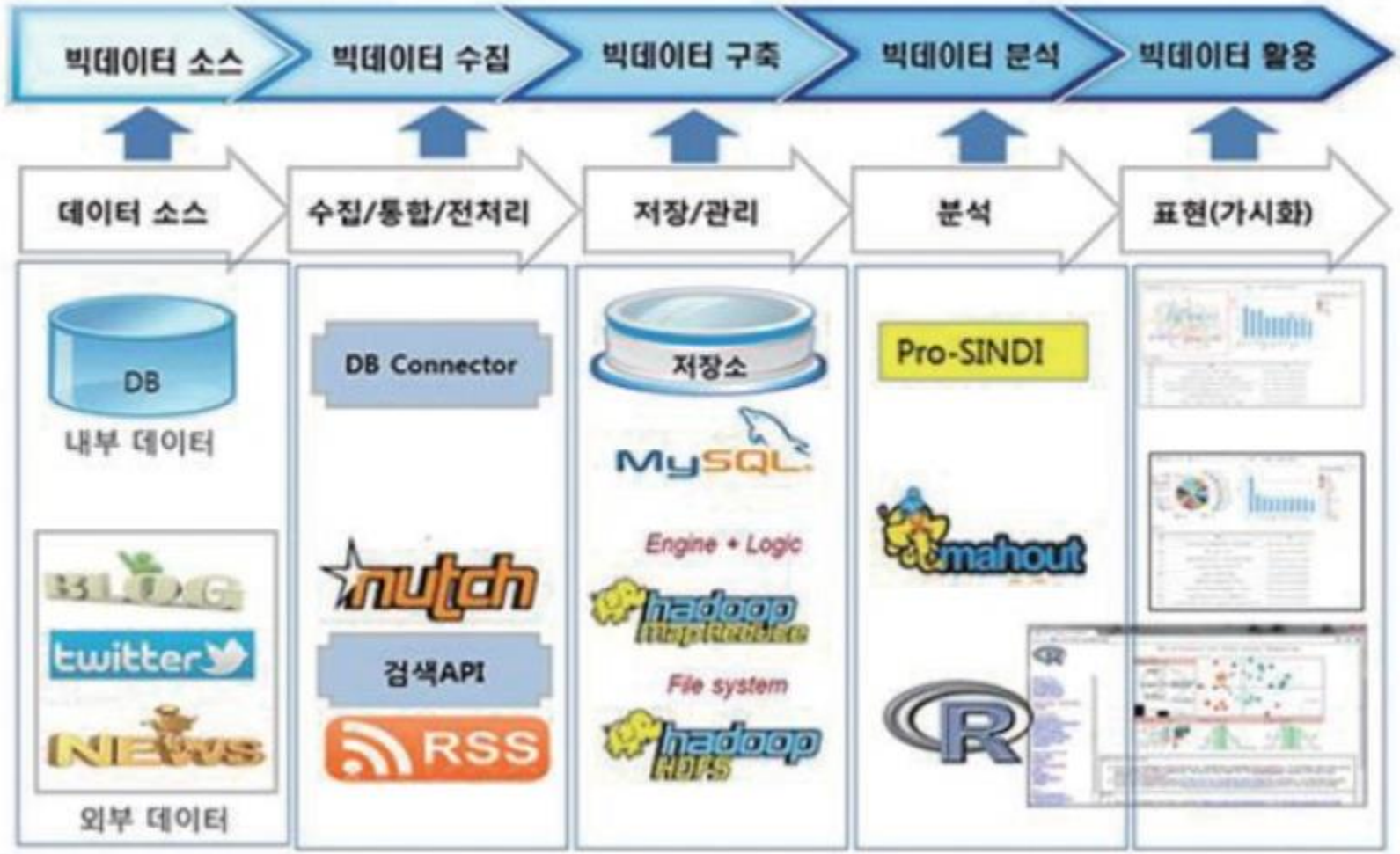
1. MongoDB 개요

차례

1. NoSQL 개요
2. MongoDB소개
3. 개발환경 구축
4. MongoDB 구조
5. 데이터 타입

1. NoSQL 개요

- Big Data Processing Flow



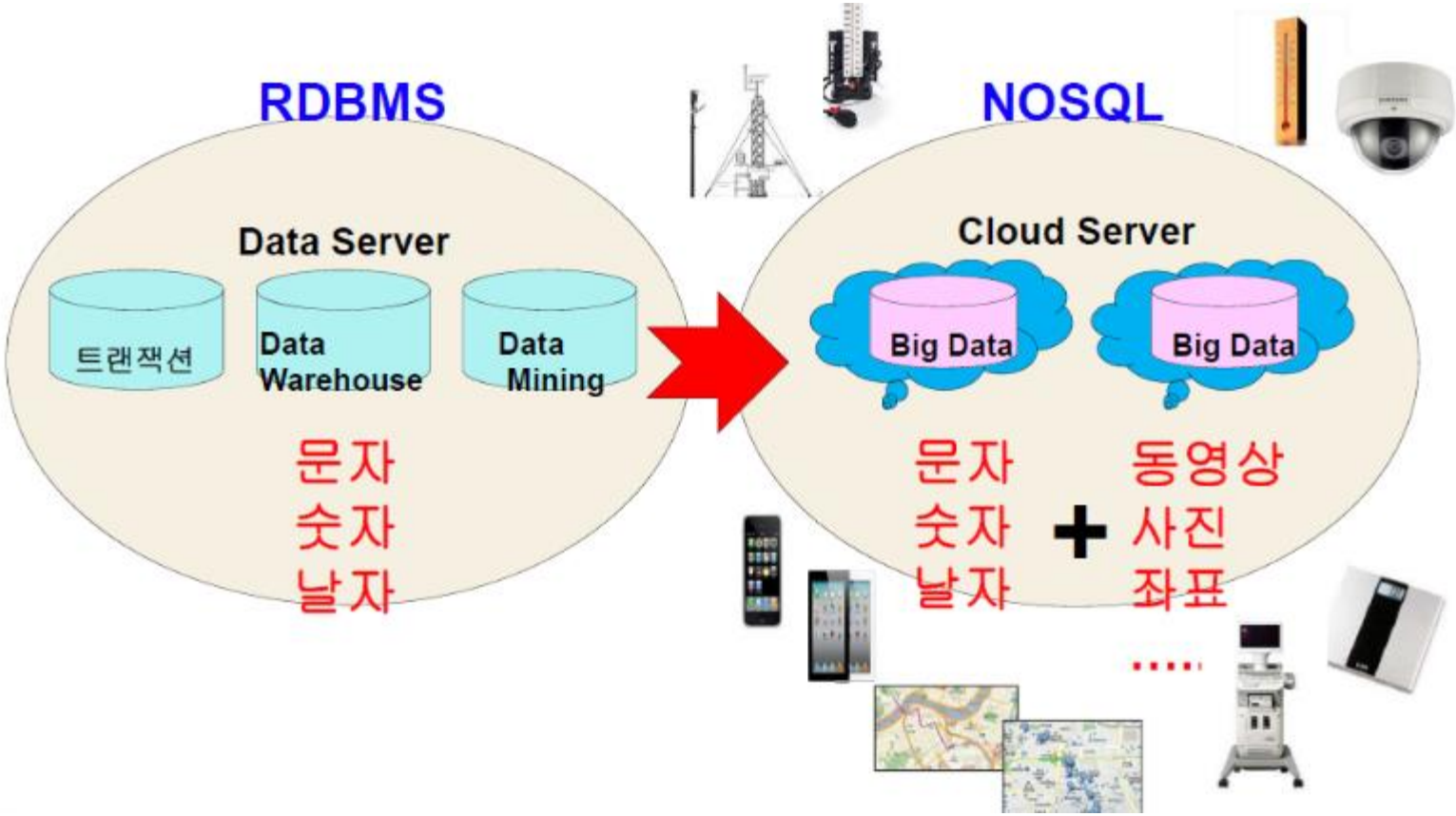
1. NoSQL 개요

- NoSQL?
  - NoSQL 데이터베이스는 전통적인 관계형 데이터베이스보다 덜 제한적인 일관성 모델을 이용하는 데이터의 저장 및 검색을 위한 매커니즘을 제공

- No SQL(X), Not Only SQL(O)
- Non-Relational Operational Database SQL

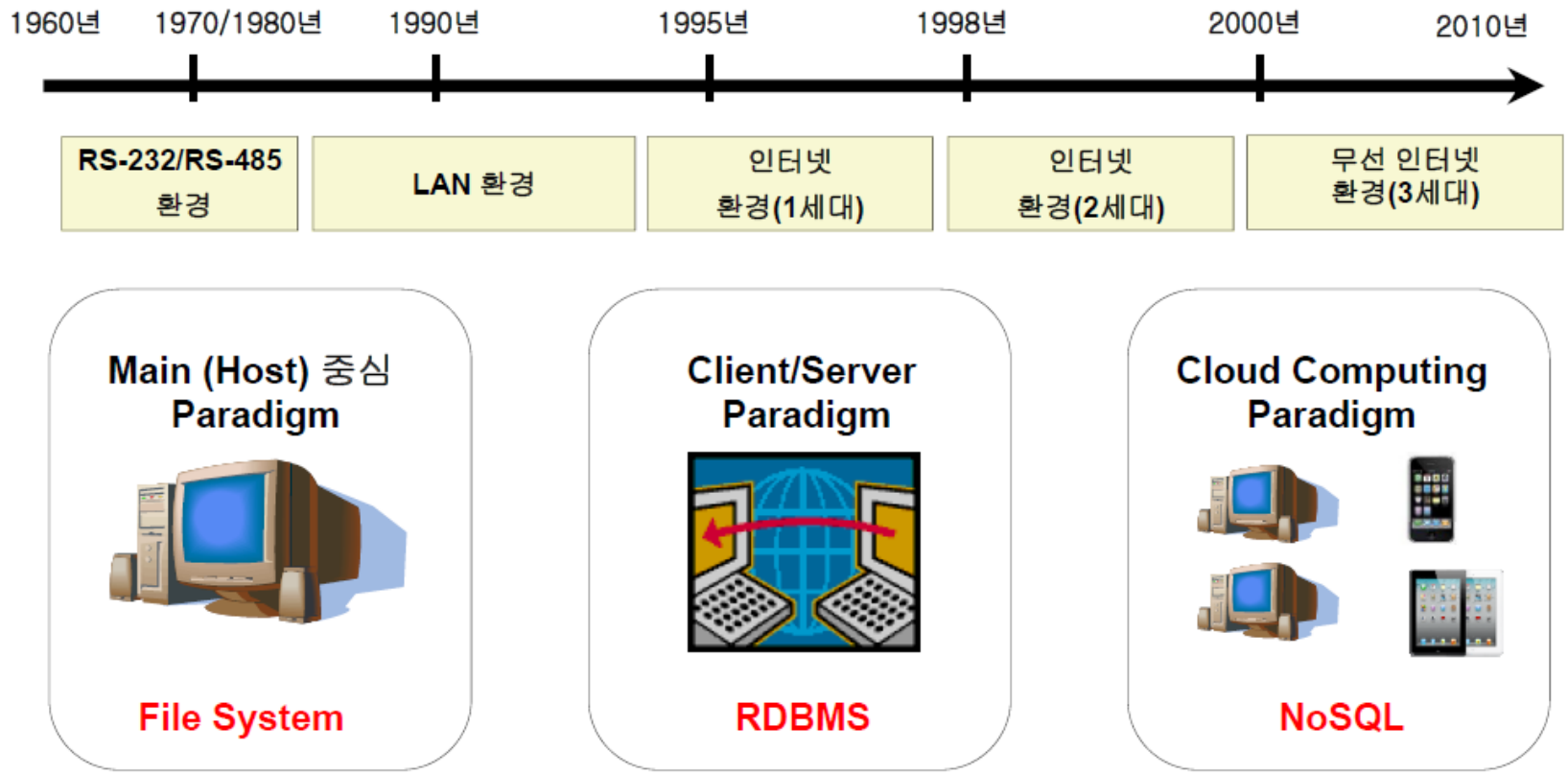
1. NoSQL 개요

- NoSQL : RDBMS vs NoSQL



1. NoSQL 개요

- NoSQL의 시대적 요구



1. NoSQL 개요

- NoSQL의 시대적 요구
  - 카를로스 트로찌(Carlo Strozzi)
    - 1998년 표준 SQL 인터페이스를 채용하지 않은 자신의 경량 오픈 소스 관계형데이터베이스를 NoSQL이라 명명
  - 2009년 라스트 FM의 요한 오스칼손(Johan Oskarsson)
    - 오픈소스 분산 데이터베이스를 논하기 위한 미트업 행사를 조직하면서, 이와 같은 데이터베이스를 NoSQL이라고 부름.
  - 관계형 데이터베이스 시스템의 주요 특성을 보장하는 ACID 제공을 주로 시도 하지않은 수 많은 비관계형, 분산 데이터 자료 공간의 등장에 따라 NoSQL이라는 명칭 사용
    - ACID: 원자성(Atomicity), 일관성(Consistency), 독립성(Isolation), 지속성(Durability)

1. NoSQL 개요

NoSQL 특징

1. RDBMS와 달리 데이터 간의 관계를 정의하지 않는다.
   - RDBMS는 데이터 관계를 외래키 등으로 정의하고 JOIN 연산을 수행할 수 있지만, NoSQL은 JOIN 연산이 불가능하다.
2. RDBMS에 비해 대용량의 데이터를 저장할 수 있다.
   - 페타바이트 급의 대용량 데이터를 저장할 수 있다.
3. 분산형 구조이다.
   - 여러 곳의 서버에 데이터를 분산 저장해 특정 서버에 장애가 발생했을 때도 데이터 유실 혹은 서비스 중지가 발생하지 않도록 한다.
4. 고정되지 않은 테이블 스키마를 갖는다.
   - RDBMS와 달리 테이블의 스키마가 유동적이다. 데이터를 저장하는 칼럼이 각기 다른 이름과 다른 데이터 타입을 갖는 것이 허용된다.

1. NoSQL 개요

- NoSQL의 장점

1. 클라우드 컴퓨팅 환경에 적합

   1) Open Source

   2) 하드웨어 확장에 유연한 대처 가능

   3) RDBMS에 비해 저렴한 비용으로 분산 처리와 병렬처리 가능

2. 유연한 데이터 모델

   1) 비정형 데이터 구조 설계로 설계 비용감소

   2) 관계형 데이터베이스의 Relationship과 Join 구조를 Linking과 Embedded로 구현하여 성능이 빠름

3. Big Data 처리에 효과적

   1) Memory Mapping 기능을 통해 Read/Write가 빠름.

   2) 전형적인 OS와 Hardware에 구축 가능.

   3) 기존 RDB와 동일하게 데이터 처리 가능

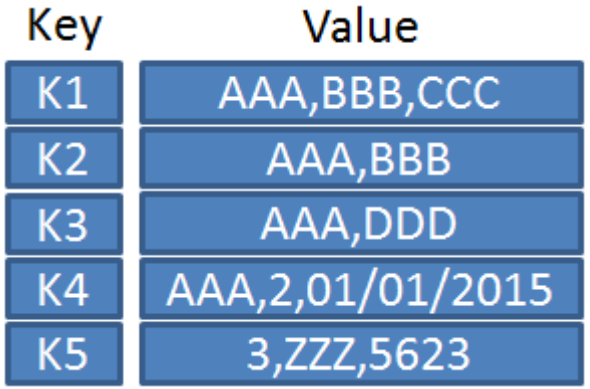
1. NoSQL 개요

- NoSQL의 단점
  - 데이터 업데이트 중 장애가 발생하면 데이터 손실 발생 가능
  - 많은 인덱스를 사용하려면 충분한 메모리가 필요. 인덱스 구조가 메모리에 저장
  - 데이터 일관성이 항상 보장되지 않음

- NoSQL의 종류
  - **Key-Value Database**
  - **Wide-Column Database**
  - **Wide-Column Database**
  - **Graph Database**

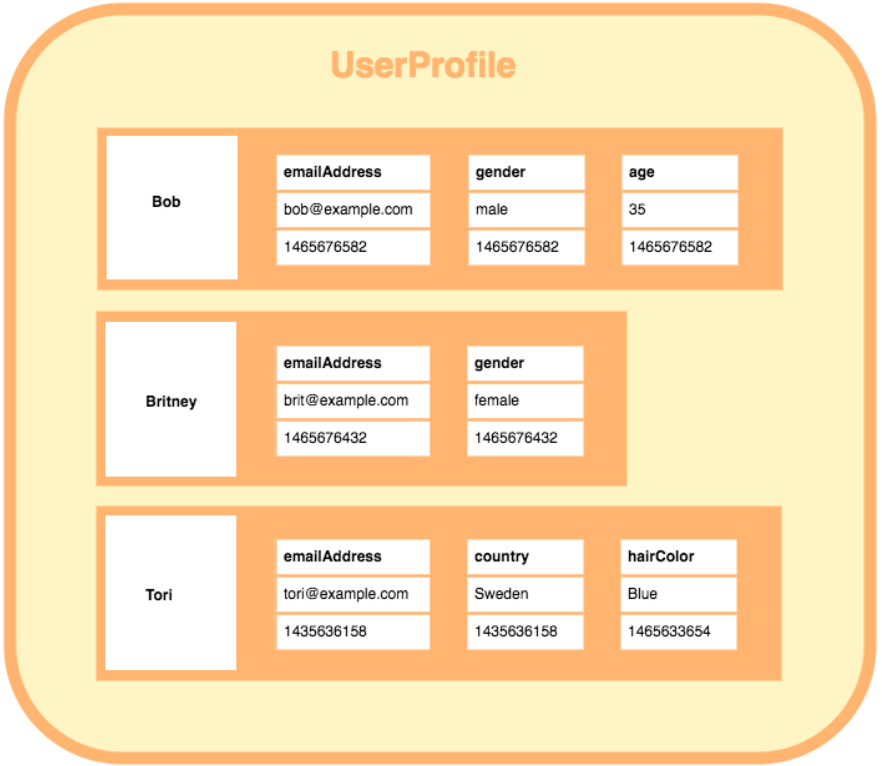
1. NoSQL 개요

- NoSQL의 종류
  - **Key-Value Database**
    - 기본적인 패턴으로 **KEY-VALUE 하나의 묶음(Unique)**으로 저장되는 구조로 단순한 구조이기에 속도가 빠르며 분산 저장 시 용이하다.
    - Key 안에 (COLUMN, VALUE) 형태로 된 여러 개의 필드, 즉 COLUMN FAMILIES 갖는다.
    - 주로 SERVER CONFIG, SESSION CLUSTERING등에 사용되고 엑세스 속도는 빠르지만, SCAN에는 용이하지 않다.
    - Ex) Redis, Oracle NoSQL Database, VoldeMorte



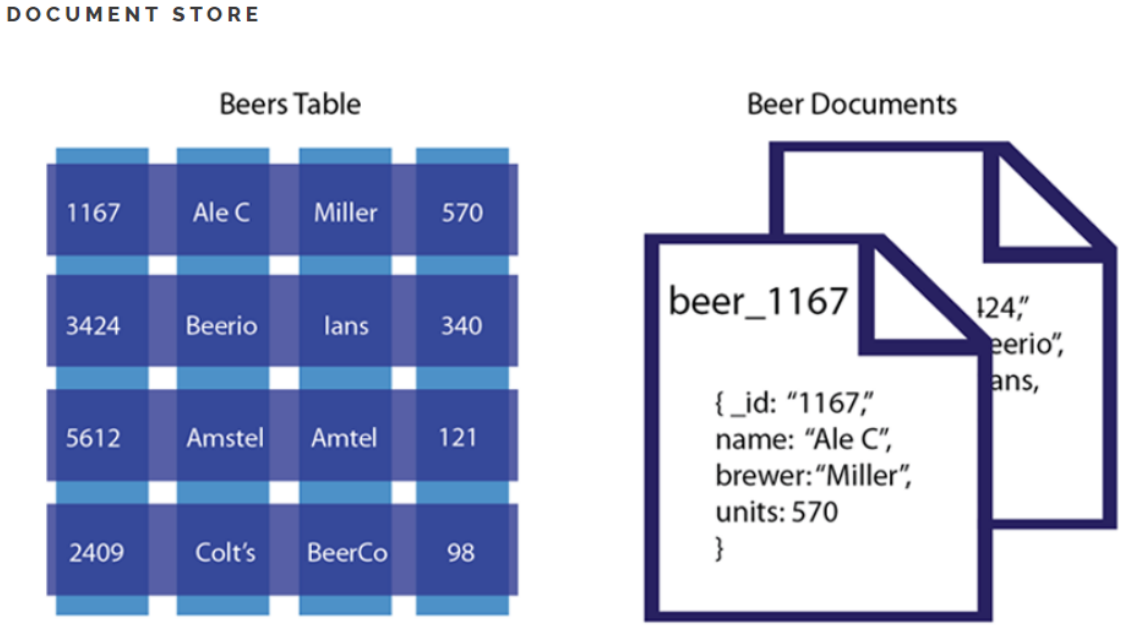
1. NoSQL 개요

- NoSQL의 종류
  - Wide-Column Database
    - 행마다 키와 값을 저장할 때 각각 다른 값의 다른 수의 스키마를 가질 수 있다.
    - 사용자의 이름(key)에 해당하는 값에 스키마들이 각각 다름을 볼 수 있다.
    - 이러한 구조를 갖는 WIDE COLUMN DATABASE는 대량의 데이터의 압축, 분산처리, 집계 쿼리 (SUM, COUNT, AVG 등)및 쿼리 동작 속도 그리고 확장성이 뛰어난 것이 그 대표적 특징이라 할 수 있다.
    - EX) Hbase, GoogleBigTable, Vertica



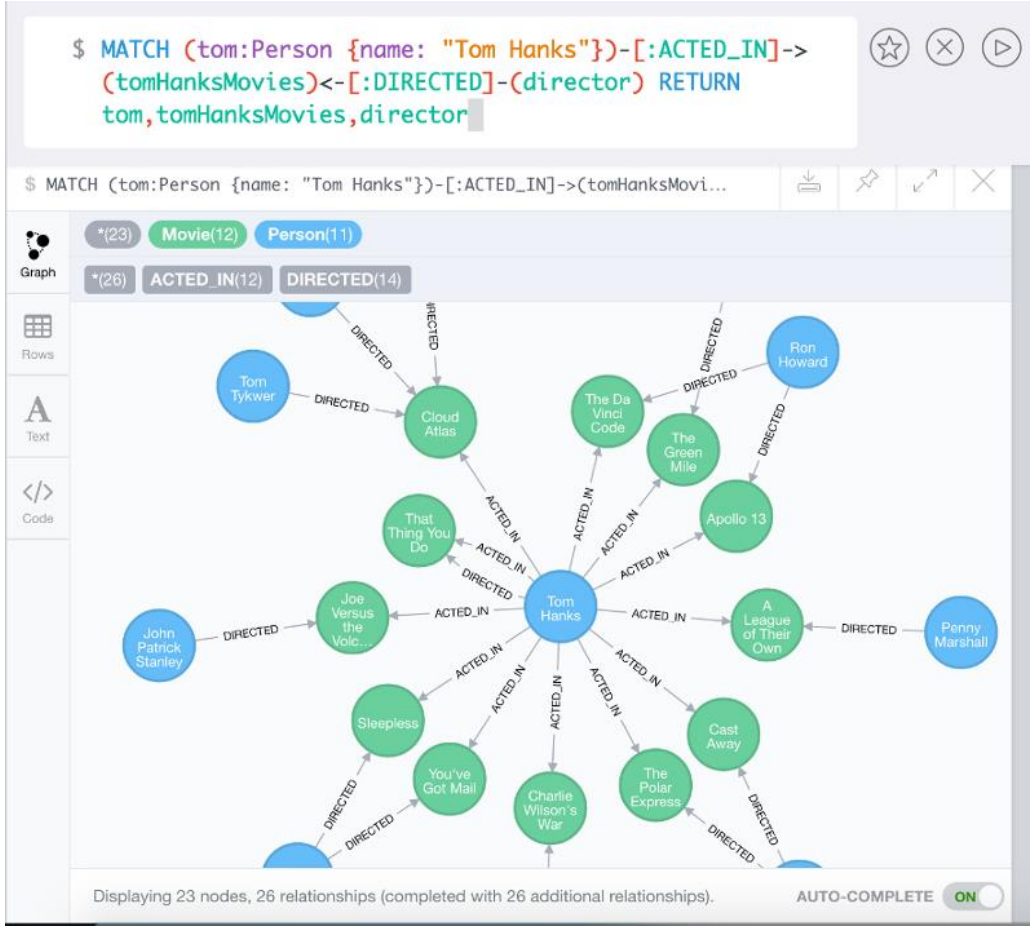
1. NoSQL 개요

- NoSQL의 종류
  - **Document Database**
    - 테이블의 스키마가 유동적, 즉 레코드마다 각각 다른 스키마를 가질 수 있다.
    - 보통 XML, JSON과 같은 DOCUMENT를 이용해 레코드를 저장한다.
    - 트리형 구조로 레코드를 저장하거나 검색하는 데 효과적이다.
    - Ex) MongoDB, CouchDB, Azure Cosmos DB



1. NoSQL 개요

- NoSQL의 종류
  - **Graph Database**
    - 데이터를 노드로(그림에서 파란, 녹색 원) 표현하며 노드 사이의 관계를 엣지(그림에서 화살표)로 표현
    - 일반적으로 RDBMS 보다 성능이 좋고 유연하며 유지보수에 용이한 것이 특징.
    - Social networks, Network diagrams 등에 사용할 수 있다.
    - Ex) Neo4j, BlazeGraph, OrientDB



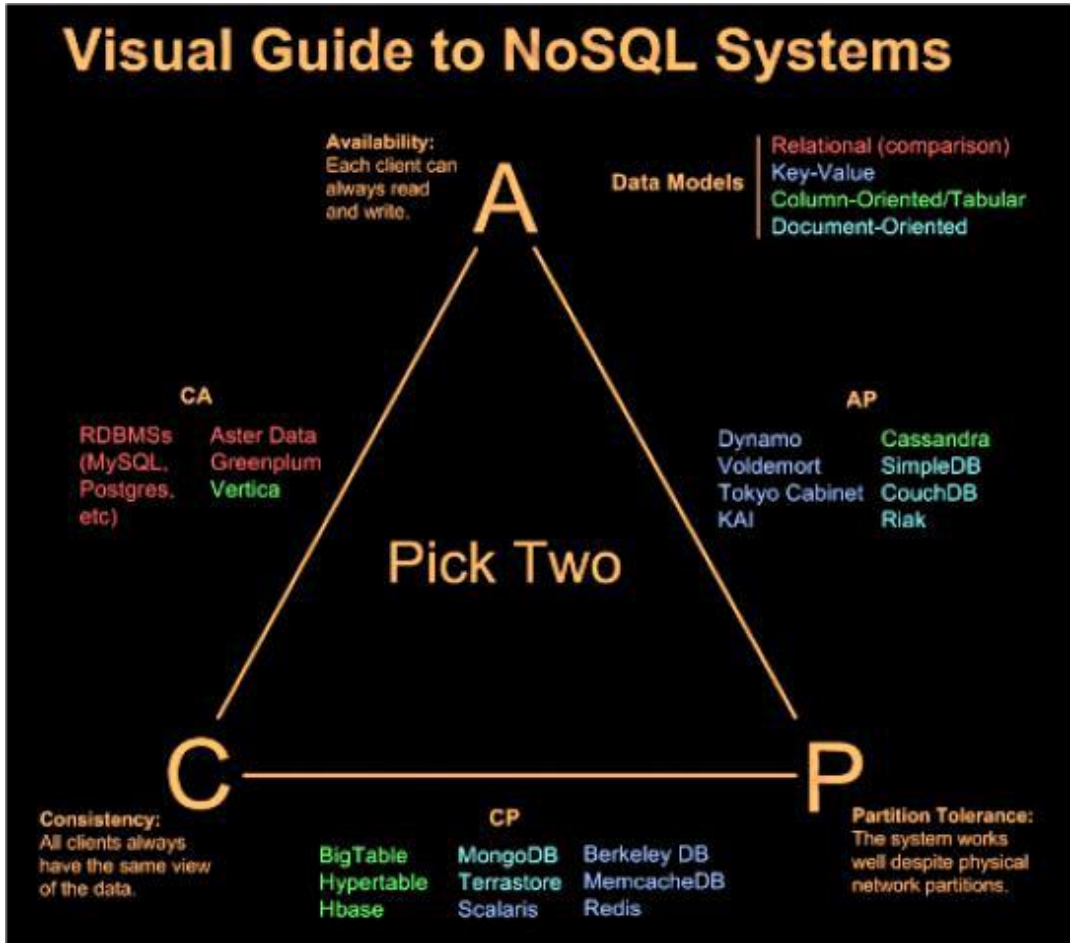
1. NoSQL 개요

- DBMS Ranking(참조: https://db-engines.com/en/ranking)

398 systems in ranking, June 2022

Rank			DBMS	Database Model	Score		
Jun 2022	May 2022	Jun 2021			Jun 2022	May 2022	Jun 2021
1.	1.	1.	Oracle	Relational, Multi-model	1287.74	+24.92	+16.80
2.	2.	2.	MySQL	Relational, Multi-model	1189.21	-12.89	-38.65
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	933.83	-7.37	-57.25
4.	4.	4.	PostgreSQL	Relational, Multi-model	620.84	+5.55	+52.32
5.	5.	5.	MongoDB	Document, Multi-model	480.73	+2.49	-7.49
6.	6.	↑ 7.	Redis	Key-value, Multi-model	175.31	-3.71	+10.06
7.	7.	↓ 6.	IBM Db2	Relational, Multi-model	159.19	-1.14	-7.85
8.	8.	8.	Elasticsearch	Search engine, Multi-model	156.00	-1.70	+1.29
9.	9.	↑ 10.	Microsoft Access	Relational	141.82	-1.62	+26.88
10.	10.	↓ 9.	SQLite	Relational	135.44	+0.70	+4.90
11.	11.	11.	Cassandra	Wide column	115.45	-2.56	+1.34
12.	12.	12.	MariaDB	Relational, Multi-model	111.58	+0.45	+14.79
13.	↑ 14.	↑ 26.	Snowflake	Relational	96.42	+2.91	+61.67
14.	↓ 13.	↓ 13.	Splunk	Search engine	95.56	-0.79	+5.30
15.	15.	15.	Microsoft Azure SQL Database	Relational, Multi-model	86.01	+0.68	+11.22
16.	16.	16.	Amazon DynamoDB	Multi-model	83.88	-0.58	+10.12
17.	17.	↓ 14.	Hive	Relational	81.58	-0.03	+1.89
18.	18.	↓ 17.	Teradata	Relational, Multi-model	70.41	+2.02	+1.07
19.	19.	↓ 18.	Neo4j	Graph	59.53	-0.61	+3.78
20.	20.	20.	Solr	Search engine, Multi-model	56.61	-0.64	+4.52

- NoSQL과 CAP 이론
  - NoSQL은 분산형 구조를 띠고 있기 때문에 분산 시스템의 특징을 그대로 반영하는데, 그 특성 중의 하나가 CAP 이론이다.
  - 2002년 버클리대학의 Eric Brewer 교수에 의해 발표된 분산 컴퓨팅 이론으로, **분산 컴퓨팅 환경**은 **일관성(Consistency)**, **가용성(Availability), 분산 가용성(Partitioning)** 세 가지 특징을 가지고 있으며, **이중 두 가지만 만족**할 수 있다는 이론이다.



1. NoSQL 개요

- CAP이론의 3가지 특징

데이터 일관성 (Consistency)	- 모든 노드들은 같은 시간에 같은 데이터를 보여줘야 함 (각각의 사용자가 항상 동일한 데이터를 조회함)
가용성 (Availability)	- 몇몇 노드가 다운되어도 다른 노드들에게 영향을 주지 않아야 함 (모든 사용자가 항상 읽고 쓸 수 있음)
단절 내성 (Partition Tolerance)	- NW장애로 메시지를 손실하더라도 시스템은 정상 동작을 해야 함 (물리적 네트워크 분산 환경에서 시스템이 잘 동작함)

- CAP의 분류

구분	특성	예
C + A	- 시스템이 죽더라도 메시지 손실은 방지하는 강한 신뢰형 - 트랜잭션이 필요한 경우 필수적	- 일반 RDBMS
C + P	- 모든 노드가 함께 퍼포먼스를 내야하는 성능형	- 구글의 BigTable, HyperTable, HBase, MongoDB
A + P	- 비 동기화된 서비스 스토어에 적합	- Dynamo, Apache Cassandra, CouchDB, Oracle Coherence

1. NoSQL 개요

- NoSQL & RDBMS

	RDBMS	NoSQL
적합한 사용례	데이터 정합성이 보장되어야 하는 은행 시스템	낮은 지연 시간, 가용성이 중요한 SNS 시스템
데이터 모델	정규화와 참조 무결성이 보장된 스키마	스키마가 없는 자유로운 데이터 모델
트랜젝션	강력한 ACID 지원	완화된 ACID(BASE)
확장	하드웨어 강화(Scale up)	수평 확장 가능한 분산 아키텍처 (Scale out)
API	SQL 쿼리	객체 기반 API 제공

1. NoSQL 개요

- MongoDB 적용 사례
  - Disney Interactive Media Group
    - Mysql -> MongoDB
    - Mysql 바이너리 데이터 저장 한계 및 성능 문제
    - 다양한 Game, Media Data 관리시스템에 적용
    - ReplicaSets & Auto Sharding 유연성과 확장성 활용
  - Music Television
    - 비디오/오디오 Content Management System에 적용
    - MySQL -> NoSQL로 전환
    - MTV의 계층적 데이터구조에 적합한 데이터 모델 활용
    - 쉬운 Query와Index를 이용한 빠른 검색기능 활용

1. NoSQL 개요

- MongoDB 적용 사례
  - Business Media Company(Forbes)
    - 원고 자동 수집 및 발행 시스템에 적용
    - Oracle DB -> NoSQL로 전환
    - 정형적인 Static Data관리에서 Dynamic Data 관리로 전환하면서 발생하는 재설계 및 구축비용 절감 목적으로 활용
  - Shutterfly
    - 인터넷 기반 사진 정보 및 개인출판 서비스 사이트
    - Oracle DB를 NoSQL로 전환(20TB)
    - 100만명의 고객/60억개의 이미지/초당10,000개 트랜잭션 처리에서 발생하는 구축/관리 비용 및 성능 문제가 이슈

1. NoSQL 개요

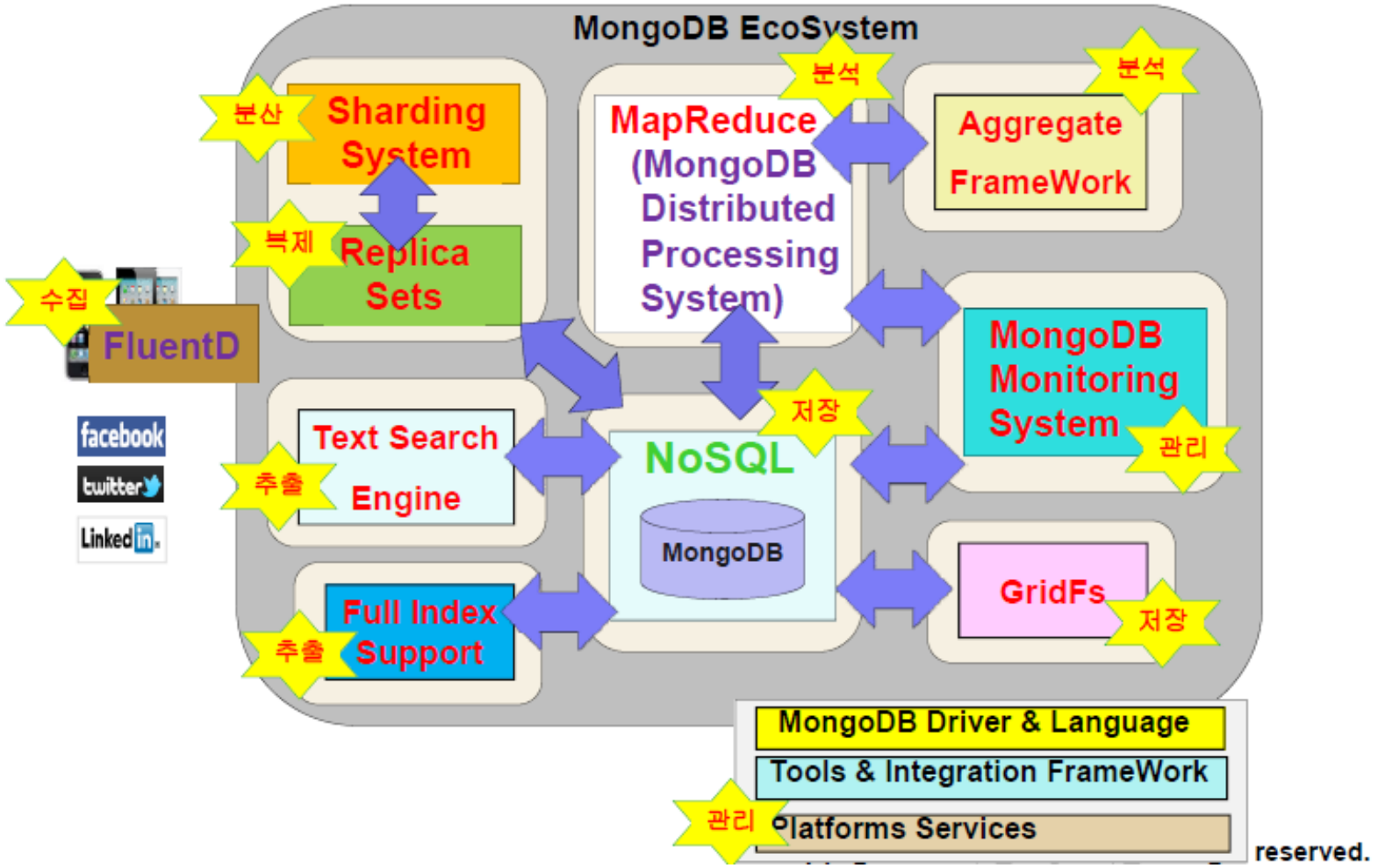
- 국내 적용사례

2. MongoDB 소개

- MongoDB란?
  - Humongos라는 회사의 제품명이 였으며 10gen으로 회사명이 변경 -> 현재 Mongodb.inc로 변경
  - JSON Type의 데이터 저장 구조를 제공
  - Sharding(분산)/Replica(복제) 기능을 제공.
  - MapReduce(분산/병렬처리) 기능을 제공.
  - CRUD(Create, Read, Update, Delete)위주의 다중 트랜잭션처리도 가능
  - Memory Mapping기술을기반으로 Big Data 처리에 탁월한 성능을 제공

2. MongoDB 소개

- MongoDB EcoSystem



2. MongoDB 소개

- MongoDB 장점
  - MongoDB의 인기
  - 빠른 속도와 확장성
  - 친숙함과 이용의 편리성
  - 쉽고 빠른 분산 컴퓨팅 환경 구성

- 언제 MongoDB사용 해야하는가?
  - 스키마가 자주 바뀌는 환경
  - 분산 컴퓨팅 환경

3. MongoDB 환경 구축

- download url: https://www.mongodb.com/try/download/community
  - Version : 5.0.9
  - Platform : Windows
  - Package : zip
- Download
- 압축해제 : c:₩monogdb
- Path 연결 : 고급시스템 설정보기-> 환경변수 ->path 추가
- 데이터베이스 폴더 작성 : c:₩mongodb₩data
- 버전 확인 및 도움말 보기

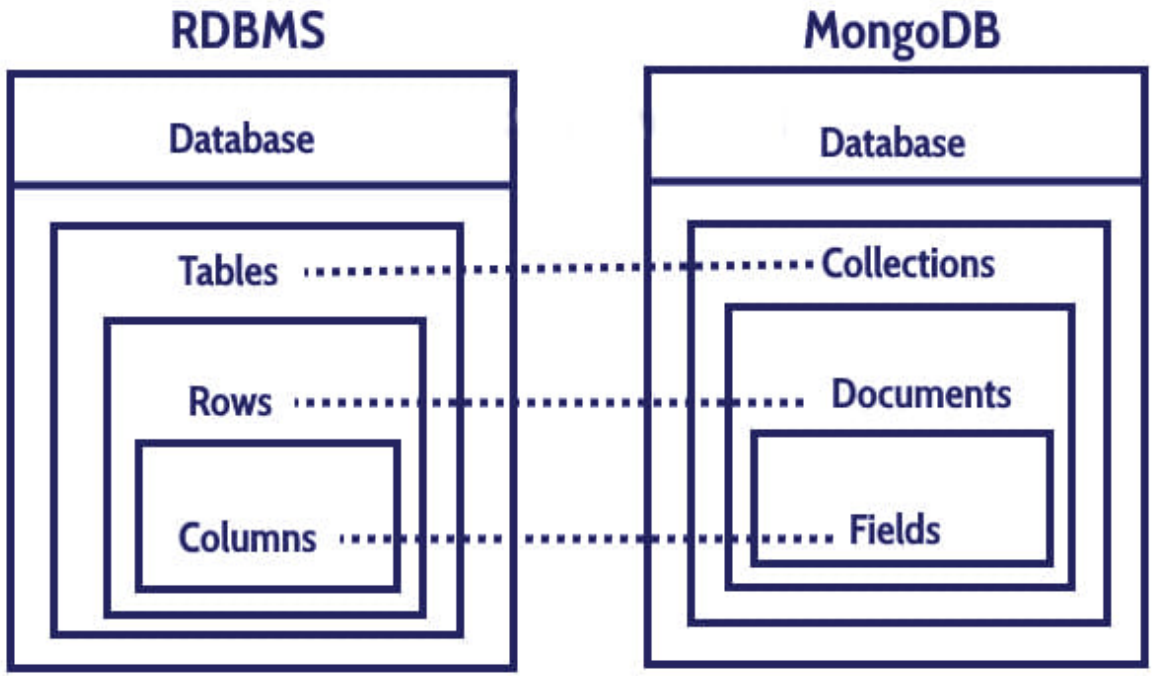
```
C:\>mongod --help
C:\>mongod --version
```

3. MongoDB 환경 구축

- Mongodb 서버 실행
  - 새 console 실행
  - c:\>mongod --dbpath c:\mongodb\data
- Mongodb 클라이언트 실행
  - 새 console 실행
  - C:\>mongo
- 서버 종료
  - C:\>use admin
  - C:\>db.shutdownServer()
- 클라이언트 종료
  - Exit 또는 ctrl+c

3. MongoDB 환경 구축

- MongoDB 구조
  - Document 기반 데이터베이스
  - Database > Collection > Document > Field 계층으로 구성

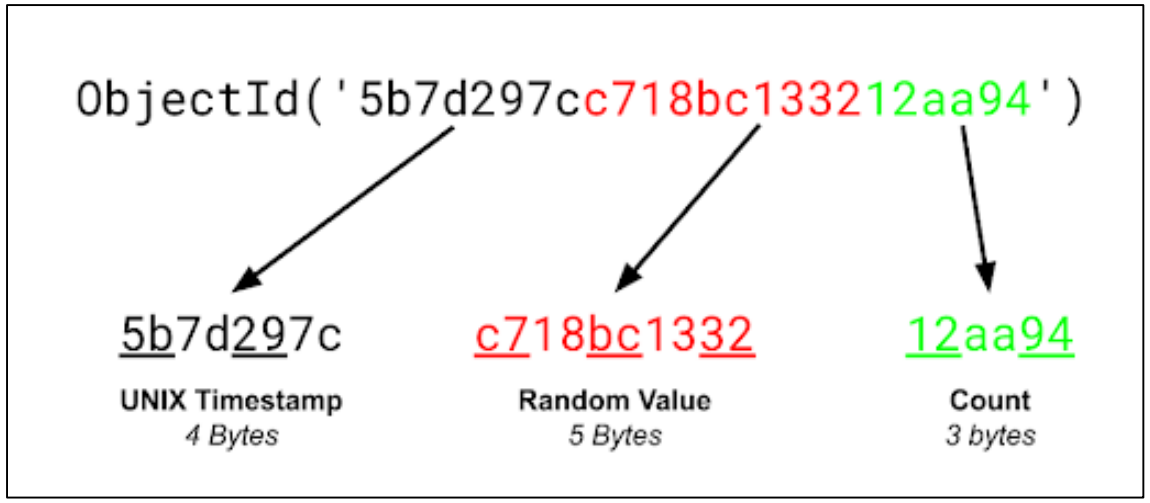


RDMBS와 MongoDB 데이터 계층 구조

3. MongoDB 환경 구축

- Document
  - RDBMS의 Row에 해당
  - Key, Value들의 쌍으로 이루어짐
  - ObjectId
    - RDBMS의 Primary Key와 같이 고유한 키를 의미
    - 차이점은 Primary Key는 DBMS가 직접 부여, ObjectId는 클라이언트에서 생성
    - 유닉스 시간, (기기Id+프로세스Id), 카운트로 구성

JSON(BSON)의 형태



4. 데이터 타입

- BSON(Binary JSON)
  - Json형태의 문서를 바이너리 형태로 인코딩한 바이트 문자열

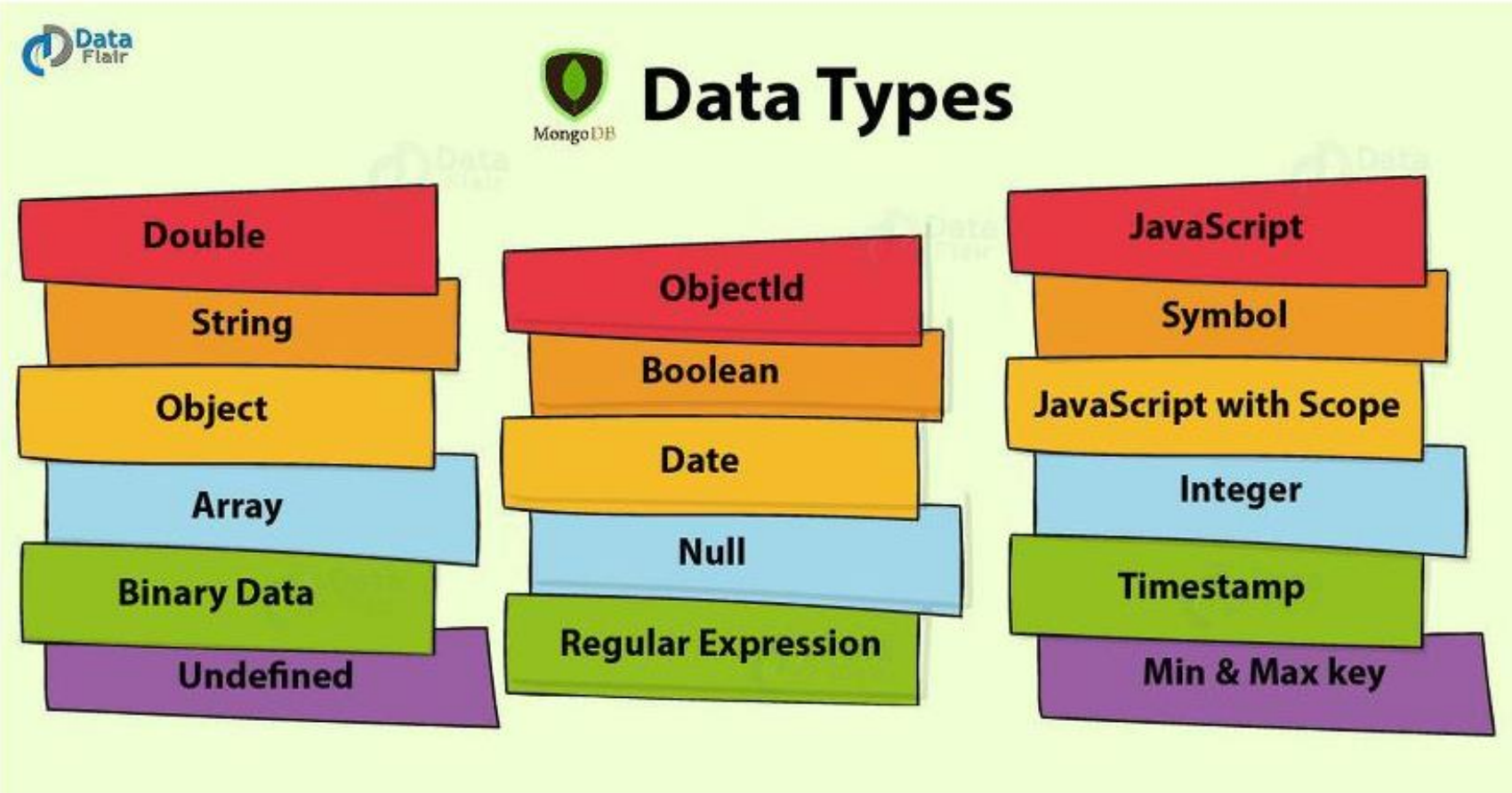
```
 {"hello": "world"}
→

  \x16\x00\x00\x00                  // total document size
  \x02                              // 0x02 = type String
  hello\x00                         // field name
  \x06\x00\x00\x00world\x00         // field value
  \x00                              // 0x00 = type EOO ('end of object')

 {"BSON": ["awesome", 5.05, 1986]}
→
```

4. 데이터 타입

- MongoDB 데이터 타입



4. 데이터 타입

- MongoDB 데이터

요소명		Data Type	
기능		포멧	설명
null		{"x" : null}	null값과 존재하지 않는 Field를 표현하는데 사용.
불린		{"x" : true}	참과 거짓을 구분할 때 사용.
숫자	실수	{"x" : 3.14}	숫자는 8바이트 부동소수점이 기본 형
	정수	{"x" : 3}	일반적인 정수도 8바이트 부동소수점을 사용함.
		{"x" : NumberInt("3")}	4바이트 정수 표현
		{"x" : NumberLong("3")}	8바이트 정수 표현
문자열		{"x" : "foobar"}	UTF-8 문자열을 표현할 때 사용.
날짜		{"x" : new Date()}	1970년 1월 1일부터의 시간을 1/1000 초 단위로 저장
정규표현식		{"x" :/foobar/i}	자바스크립트 정규표현식 문법사용이 가능함.
배열		{"x" : ["a", "b", "c"]}	값의 셋트나 리스트를 배열로 표현
내장 문서		{"x" : {"foo" : "bar"}}	문서는 다른 문서를 포함할 수도 있음.
객체 ID		{"x" : ObjectID()}	문서용 12바이트 ID. RDB의 PK와 같은 개념
코드		{"x" : function() { /* ..... */ }}	임의의 코드를 포함 할 수도 있음.