

# DB & NoSQL

조운영

# 목차

- **DB**

- **RDBMS**

- RDBMS의 단점

- **NoSQL**

- 1. Document DB

- 2. Column-family DB

- 3. Key-Value DB

- 4. Graph DB

- 5. Search Engine

- 6. Vector DB

- 7. Time Series DB

# DB

## DB & DBMS & DB System

- **DB** (DataBase)
  - : electronically 저장되고 사용되는 관련된 데이터들의 orgainzed collection
- **DBMS** (DataBase Management Systems)
  - : 사용자에게 DB를 정의하고 만들고 관리하는 기능을 제공하는 소프트웨어 시스템
    - 종류들 매우 다양
    - metadata를 저장/관리
  - metadata: DB를 정의하거나 기술하는(descriptive) data / 데이터 유형, 구조, 제약 조건, 보안, 저장, 인덱스, 사용자 그룹 등
- **DB System**
  - : database + DBMS + 연관된 applications / 줄여서 database라고도 부름

# 다양한 DBMS

**ORACLE®**  
DATABASE

 **amazon**  
REDSHIFT

  
**redis**

 Cloud  
Firestore

  
**MySQL®**

 Google  
Big Query

 **neo4j**

  
**CouchDB**

 snowflake

 **mongoDB®**

  
Postgre**SQL**



  
**cassandra**

  
**Chroma**

  
Microsoft®  
**SQL Server®**

 **amazon**  
RDS

 **elastic**

# DB

## Data Model

- **Data Model**

: DB의 구조(structure)를 기술하는데 사용되는 개념들이 모인 집합, DB 구조를 추상화해서 표현

- conceptual data model: 가장 높은 수준의 추상화, 쉽게 이해할 수 있음

entity-relationship model(E-R diagram)

- logical data model: 이해하기 어렵지 않으면서도 디테일하게 DB를 구조화

relational data model, object data model, object-relational data model

- physical data model: 컴퓨터에 데이터가 어떻게 파일 형태로 저장되는지 기술

data format, data orderings, access path(데이터 검색을 빠르게 하기 위한 구조체, index 등)

# DB

## database schema & database state

- **database schema**  
: data model을 바탕으로 database 구조를 기술(description)  
database를 설계할 때 정해짐
- **database state**  
: 특정 시점에 database에 있는 데이터  
= database snapshot  
database에 있는 현재 instances의 집합

# DB

## database language

- **database language**
  - Data Definition Language(DDL): schema를 정의하는 언어
  - Data Manipulation Language(DML): database에 있는 data를 활용하기 위한 언어(CRUD: 추가, 검색, 수정, 삭제 등)

→ 오늘날에는 통합된 언어로 존재

ex) relational database language인 SQL(Structured Query Language)

# RDBMS

## Relational Database

**ORACLE®**  
D A T A B A S E

**MySQL®**

Microsoft®  
**SQL Server®**

  
PostgreSQL

 **amazon**  
RDS

RDS는 여러 데이터베이스 엔진 지원하는 클라우드 서비스

DBMS	제작사	작동 운영체제	기타
MySQL	Oracle	Unix, Linux, Windows, Mac	오픈소스(무료), 상용
MariaDB	MariaDB	Unix, Linux, Windows	오픈소스(무료)
PostgreSQL	PostgreSQL	Unix, Linux, Windows, Mac	오픈소스(무료)
Oracle	Oracle	Unix, Linux, Windows	상용 시장 점유율 1위
...			

- table(column & row)
  - row 하나 -> entity 하나
- RDBMS : excel, oracle, MySQL, PostgreSQL
- SQL 문법 사용
- 보통 데이터를 정규화해서 저장함



# RDBMS

## 단점

### 1. 경직된 스키마: 유연한 확장성의 부족

- 이미 잔뜩 존재하는 데이터에 대해 column을 추가한다거나 하는 작업을 하기는 어렵고 위험

### 2. 과도한 **JOIN**과 성능 하락

- 보통 데이터를 (중복 제거를 위해) 정규화해서 저장함 → 출력 문법 복잡

### 3. **scale-out**이 어려움

- scale-out(서버 수를 늘려서 확장)을 위해서는 (정규화되어 있으므로) 이미 존재하는 DB를 전부 복제해서 옮겨주거나 해야함

### 4. transaction의 **ACID**를 보장하는 과정에서 서버의 성능에 영향

# RDBMS

## 단점

특히, 2000년대 초중반부터 인터넷과 SNS

-> high-throughput(높은 처리량) & low-latency(짧은 응답시간) 요구됨

& 비정형 데이터의 증가

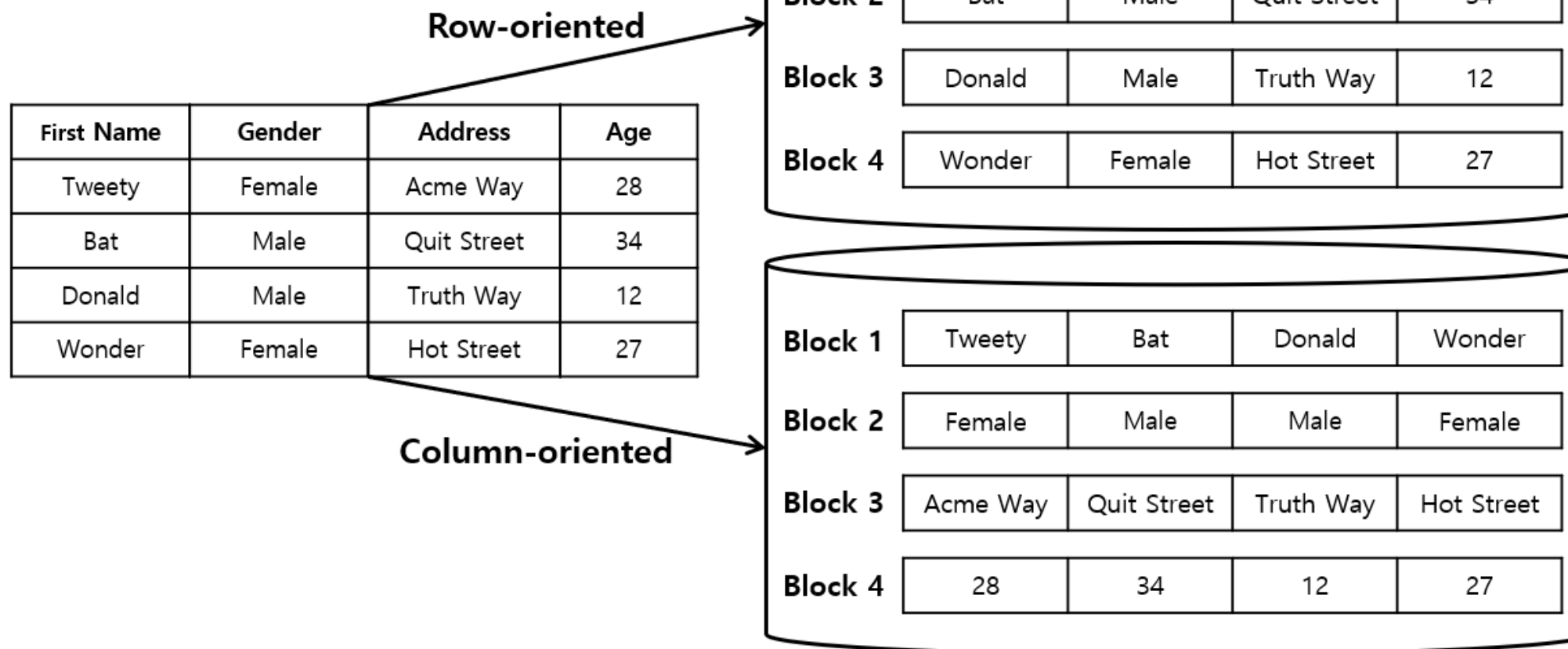
**(column-oriented DB or)**

-> **NoSQL**

# RDBMS

## row-oriented vs. column oriented

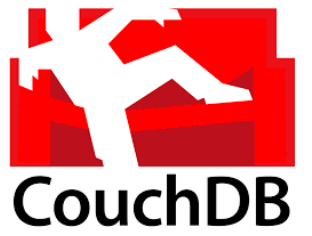
OLTP (Online Transaction Processing  
실시간 트랜잭션 처리)에 적합



OLAP (Online Analytical Processing  
대규모 데이터 분석 및 집계)에 적합

# NoSQL

## 장점



### 1. 유연한 스키마

- 주의사항: 스키마 설계 및 관리를 개발자가 잘 해야함

### 2. 중복 허용 (JOIN 회피)

- 주의사항: 중복된 데이터들이 모두 최신 데이터를 유지할 수 있도록 관리해야함

### 3. scale-out 편함

→ 서버 여러 대로 하나의 클러스터를 구성하여 사용

### 4. ACID를 일부 포기하고 high-throughput & low-latency 추구

- 금융시스템처럼 consistency가 중요한 환경에서는 사용하기 조심스러움

# Document DB

## 다양한 NoSQL DBMS



- mongoDB, Couch DB, Cloud Firestore
- collection 폴더 안에다가 document라는 파일들(json 형태로 데이터 저장)
  - document의 구조 바뀌어도 문제 없음
  - 정규화 안 함 (중복제거 안 함) → 입출력 쉬움
  - 분산처리 잘해줌
- 분산의 단점: 데이터베이스 간의 일관성이나 정확도가 떨어질 수 있음

# Column-family DB

## 다양한 NoSQL DBMS



- Cassandra, Google Cloud Bigtable
- 표 형식으로 저장하는데 유연하게 쓰고 싶다
- table 안에 row를 만들고 그 안에 column을 만들어서 씬

- row 안의 컬럼들 달라도 상관 없음

- CQL 등 자기들이 만든 언어 사용
- 정규화(중복 제거) 안 함 → 입출력 쉬움
- 분산처리 잘 함 (일관성 부족)

Row Key → Son	Gender	Age	Email	Column Key
	Male	24	son@exam.com	Data
	1465676582	1465676582	1465676582	Time Stamp
Tori	Gender	Age		
	Female	36		
	1465676421	1465676421		
John	Gender	Age	Country	
	Mail	36	Korea	
	1465672451	1465672451	146564512	

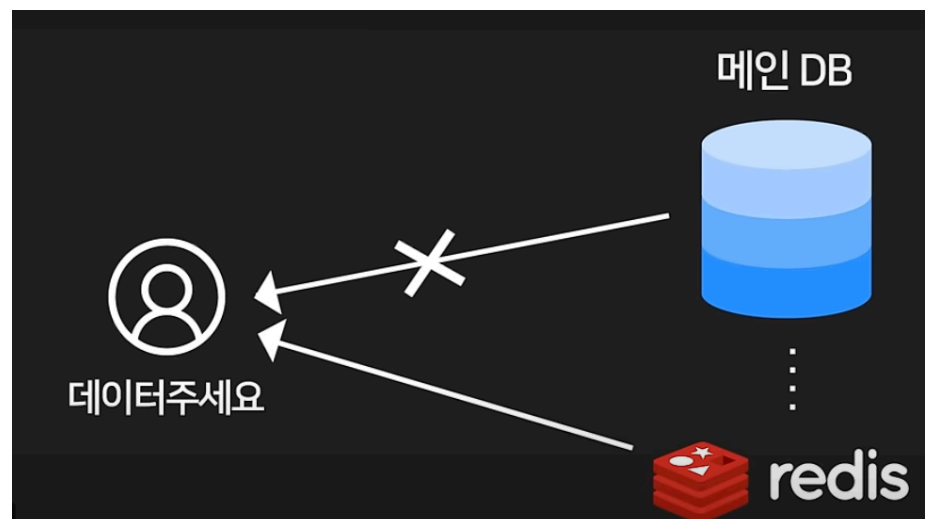
- 시간 기록 쉽게 해주는 기능 있음 → 시계열 데이터에 쓸 수 있음

# Key-Value DB

## 다양한 NoSQL DBMS



- Redis
- 서버용 DB로 많이 사용
  - 데이터를 하드디스크가 아니라 램에 먼저 저장 → 훨씬 빠름
  - 자주 필요한 데이터를 메인 DB에서 redis에 복사해두고 씬
- 자주쓰는 데이터를 캐싱하거나, 채팅을 위한 pub/sub 등에 사용

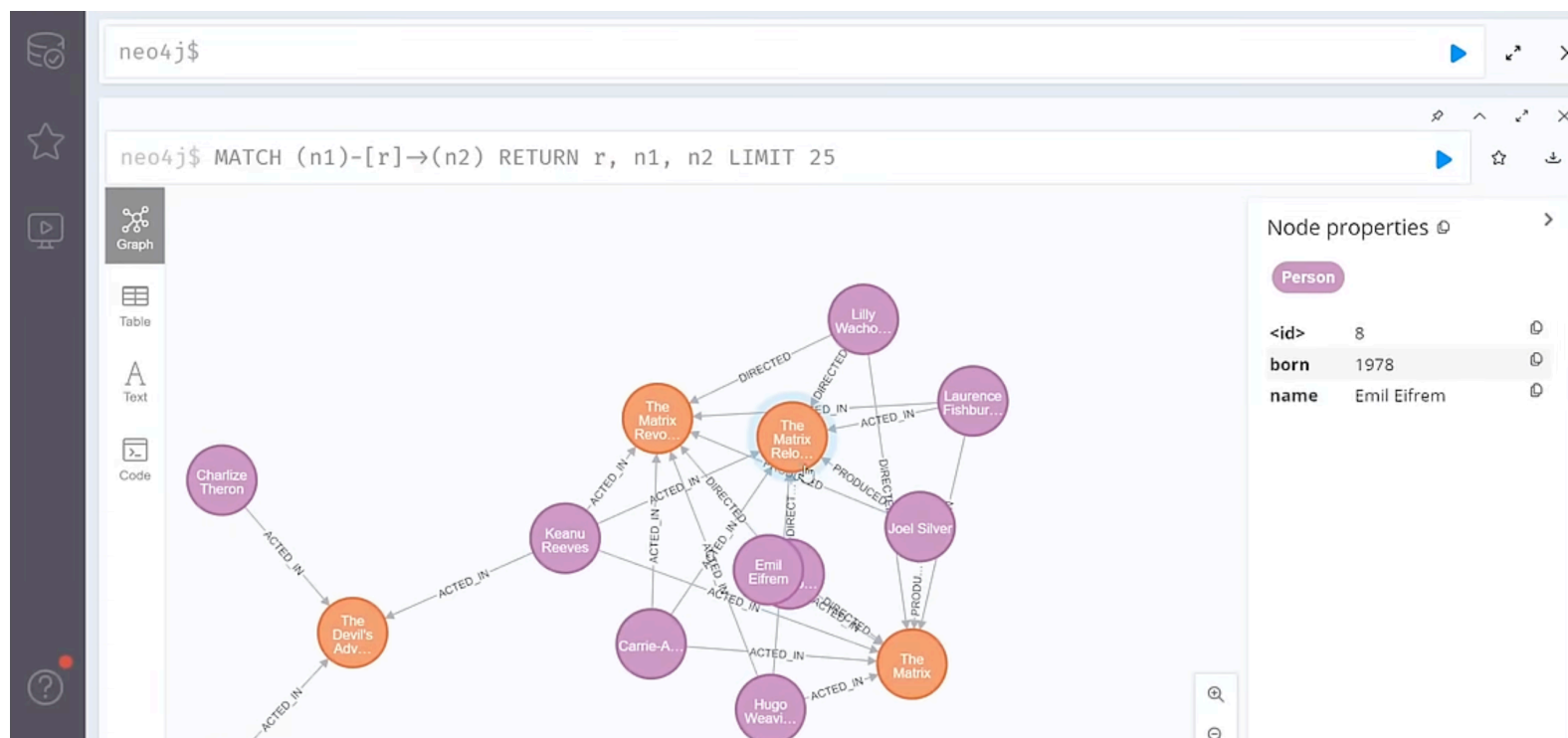


# Graph DB

## 다양한 NoSQL DBMS



- neo4j
- 자료(노드) 간의 관계와 방향 저장
- Graph Query Language 사용
- 비행기 노선, SNS 친구관계, 코로나전염맵, 추천서비스





# Search Engine

## 다양한 NoSQL DBMS



- elastic, Amazon CloudSearch, Google Cloud Search
- index(빠른 검색 도와주는 목차 같은 거)들 안에 document를 넣음
  - 검색용 인덱스를 보관하기 위해서 씬
- 기존 DB에서 데이터를 뽑아서 입력해서 사용
- 실시간 검색어, 추천 검색어, 검색어 오타 교정 등의 기능 만들기 쉬움

# Vector DB

## 다양한 NoSQL DBMS



**Chroma**

- Chroma
- ML 모델의 벡터화/임베딩 기술 활용하여 텍스트/이미지/영상/오디오 등의 비정형 데이터 저장 가능
- 메타데이터 저장 및 필터링: 벡터 항목 관련 메타데이터를 저장할 수 있다. 필터를 이용하여 DB쿼리 가능
  - 내장된 알고리즘으로 유사한 벡터들을 검색해줌

# Time Series DB

다양한 NoSQL DBMS



= TSDB

- InfluxDB
- 시계열 데이터를 처리하기 위해 최적화된 데이터베이스: 빠르고 정확하게 실시간으로 쌓이는 대규모 데이터들을 처리
- 데이터들과 시간이 함께 저장
  - timestamp를 기반으로 저장
- 주기마다 자동으로 데이터들을 처리하는 기능, 오래된 데이터를 자동으로 삭제하도록 도와주는 기능 등

# 실습!!

## MongoDB & Chroma



	MongoDB	Chroma
유형	Document DB	Vector DB
데이터 저장 형식	BSON(Binary JSON)	벡터
사용 목적	다양한 비정형 데이터 저장	벡터 기반 유사성 검색
데이터 질의 방식	MongoDB Query Language(MQL)	벡터 유사성 계산 알고리즘 사용

### MongoDB 세팅

<https://www.notion.so/yonseibigdata/NoSQL-3a185e37177d4f909334b05cae87cae1?pvs=4>

# Reference

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

Chroma 공식문서 <https://docs.trychroma.com/guides>

23기 오재현 선배님 발제자료

쉬운코드

[https://www.youtube.com/watch?](https://www.youtube.com/watch?v=aL0XXc1yGPs&list=PLcXyemr8ZeoREWGhhZi5FZs6cvymjIBVe&index=1)

[v=aL0XXc1yGPs&list=PLcXyemr8ZeoREWGhhZi5FZs6cvymjIBVe&index=1](https://www.youtube.com/watch?v=aL0XXc1yGPs&list=PLcXyemr8ZeoREWGhhZi5FZs6cvymjIBVe&index=1)

[https://www.youtube.com/watch?](https://www.youtube.com/watch?v=sqVByJ5tbNA&list=PLcXyemr8ZeoREWGhhZi5FZs6cvymjIBVe&index=33)

[v=sqVByJ5tbNA&list=PLcXyemr8ZeoREWGhhZi5FZs6cvymjIBVe&index=33,](https://www.youtube.com/watch?v=sqVByJ5tbNA&list=PLcXyemr8ZeoREWGhhZi5FZs6cvymjIBVe&index=33)

코딩애플 <https://www.youtube.com/watch?v=ZVuHZ2Fjkl4&t=360s>

cheonbi 벨로그 <https://velog.io/@vrooming13/DB-NoSQL-Column-oriented-Column-Family-DB>

김진만 님 벨로그 <https://velog.io/@jinman-kim/VectorDB-벡터-데이터베이스-개념>

망나니개발자 님 블로그 <https://mangkyu.tistory.com/188>

끝