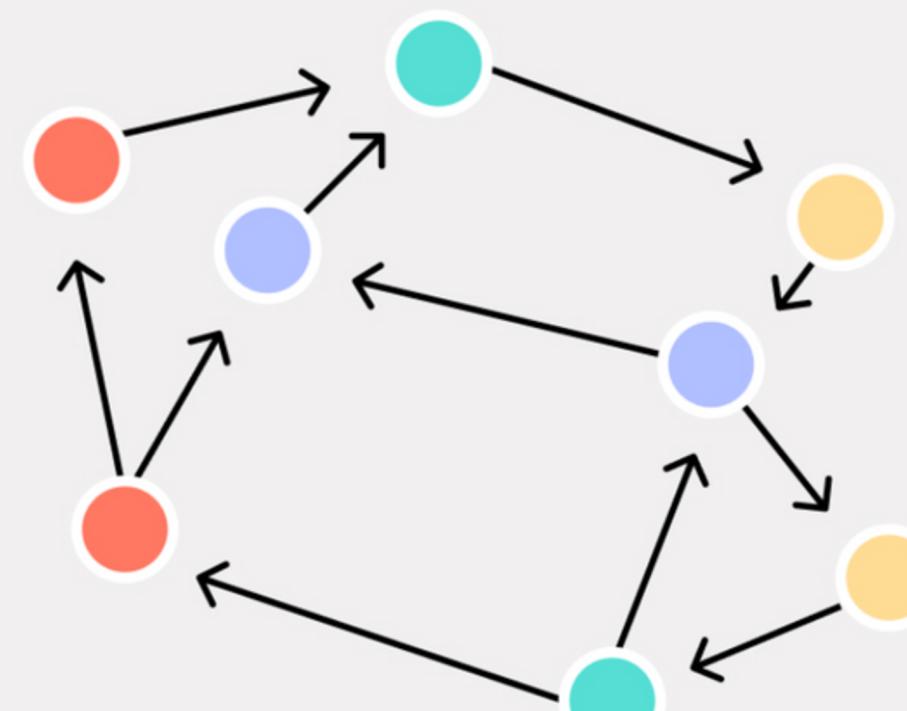


RDBMS vs NoSQL

co2plant

NoSQL이란?



Not Only SQL의 약자로, RDBMS와 다른 방식으로 데이터를 저장, 관리하는 데이터 베이스입니다.

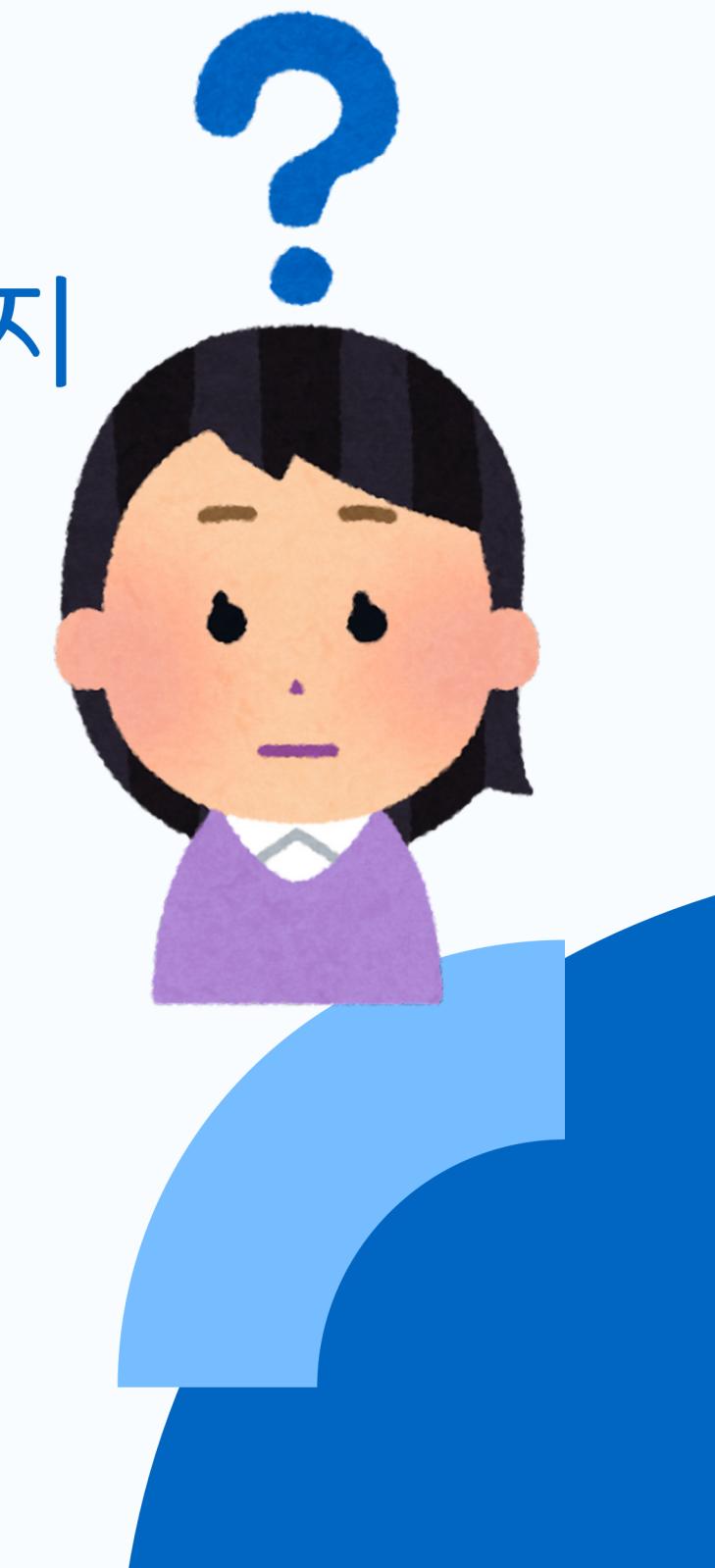
NoSQL 왜 필요한가요?

- 높은 트래픽과 빠른 처리 속도
- 수평적 확장의 가능성
- 대규모 비정형 또는 반정형 데이터 처리 능력
- 구조가 자주 변경될 경우

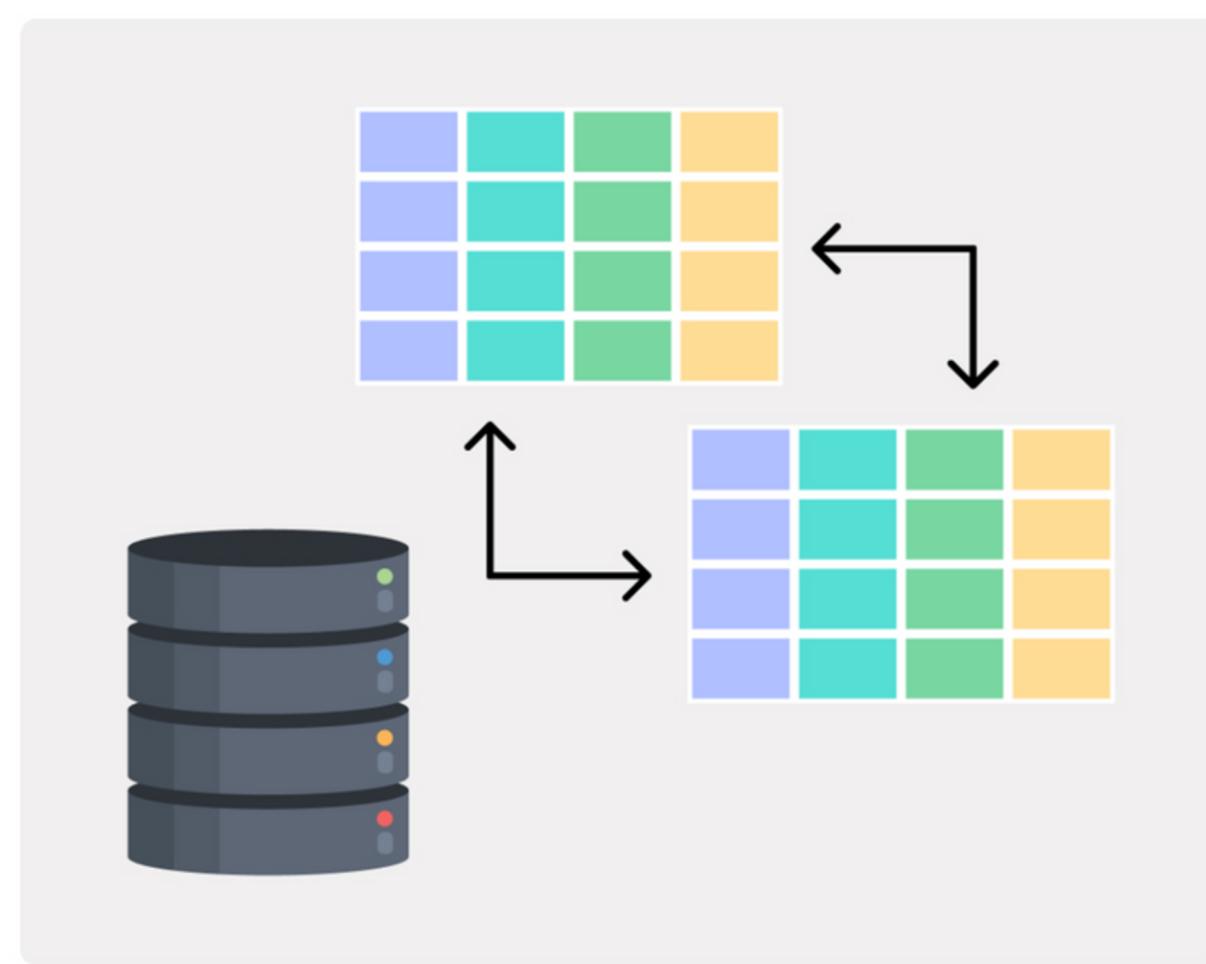


왜 RDBMS는 안되나요?

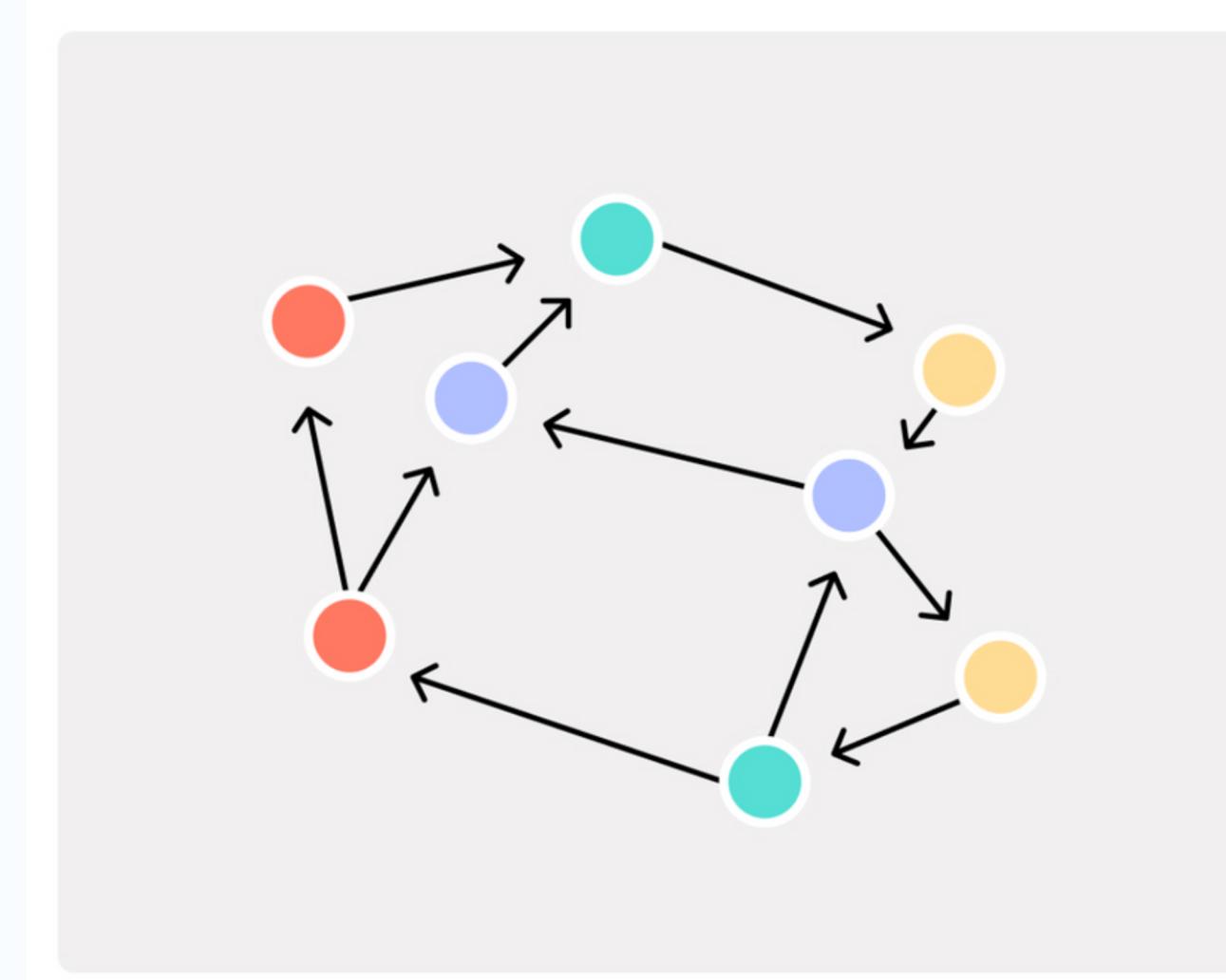
- 중복 최소화로 인한 JOIN 연산 증가(성능, 지연시간)
- 대규모 데이터가 쌓인 테이블의 스키마 변경은 위험함
- 비정형 데이터 처리 어려움



차이점



VS



스키마

RDBMS - 고정된 스키마

User table

ID	first_name	last_name	cell	city
1	Leslie	Yepp	8125552344	Pawnee

Hobbies table

ID	user_id	hobby
10	1	scrapbooking
11	1	eating waffles
12	1	working

데이터 구조와 제약조건을 미리 정의 해두고 사용
임의대로 사용자가 ID에 2.02 삽입시 오류발생

NoSQL - 유연한 스키마

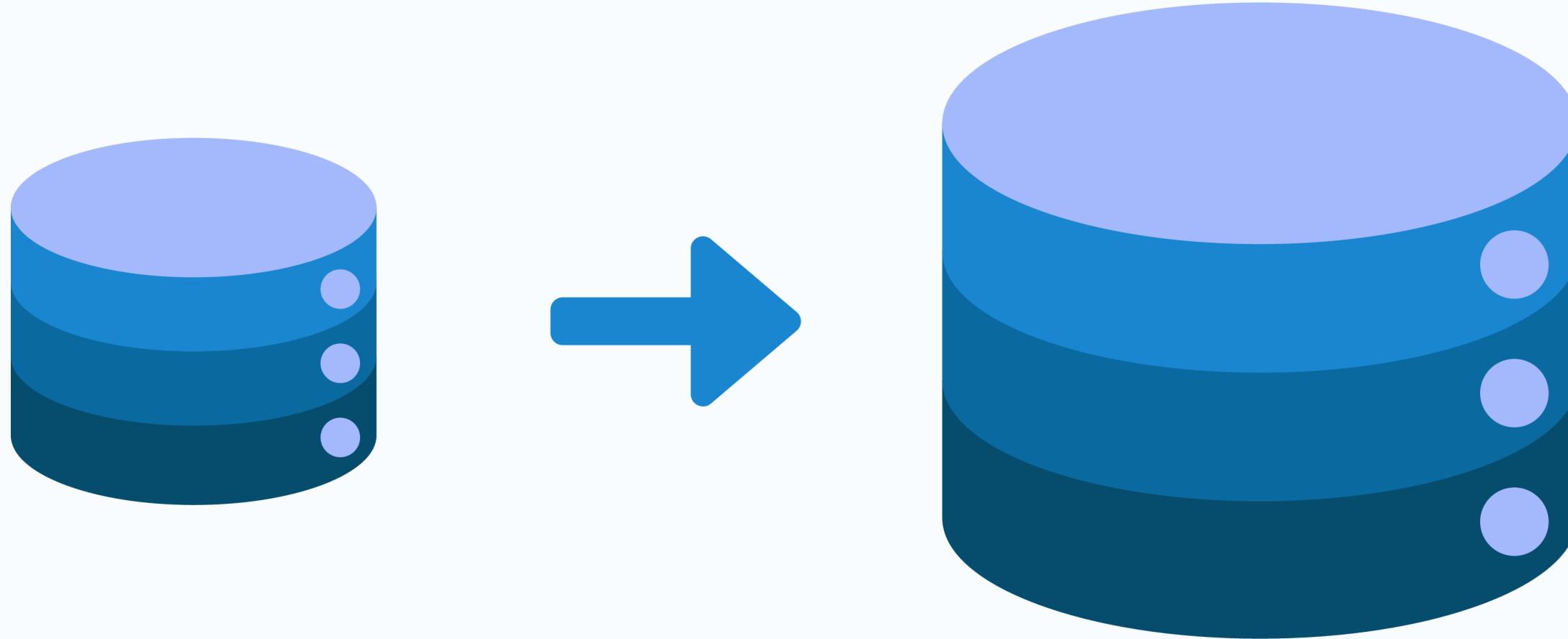
```
{  
    "_id": 1,  
    "first_name": "Leslie",  
    "last_name": "Yepp",  
    "cell": "8125552344",  
    "city": "Pawnee",  
    "hobbies": ["scrapbooking", "eating  
waffles", "working"]  
}
```

데이터를 정규화할 필요가 없음

hobbies처럼 값안에 값 묶음을 집어넣어도 문제가 생기지 않음

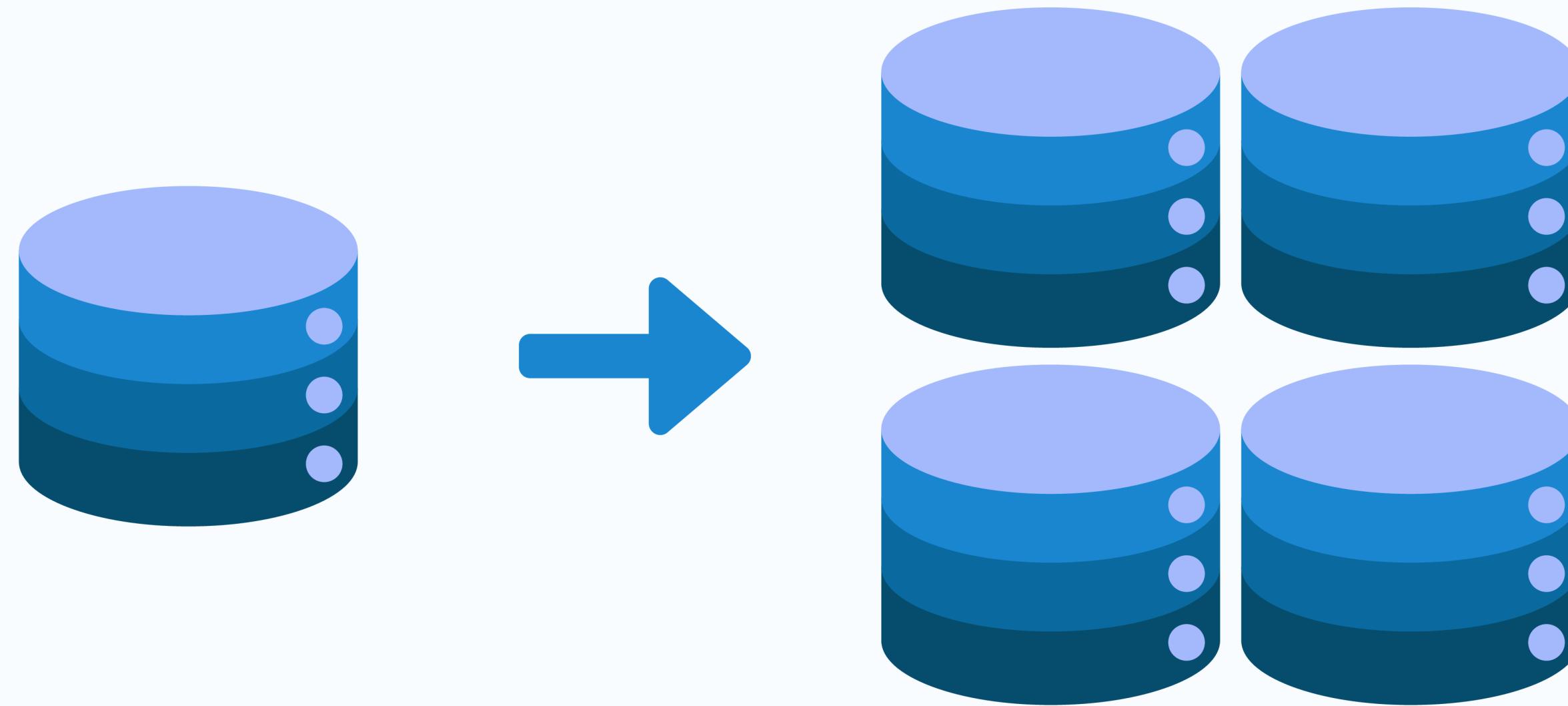
확장성

RDBMS - 스케일 업



성능, 용량의 증대를 위해 주로 수직확장을 함
복제를 통해 읽기 서버 추가 가능 But 결국 주 서버에 부하 집중

NoSQL - 스케일 아웃



설계부터 데이터 분산 저장을 염두에 둠

더 좋은 성능이 필요하면 새로운 서버를 클러스터에 추가하기만 하면됨

일관성 모델

RDBMS - ACID

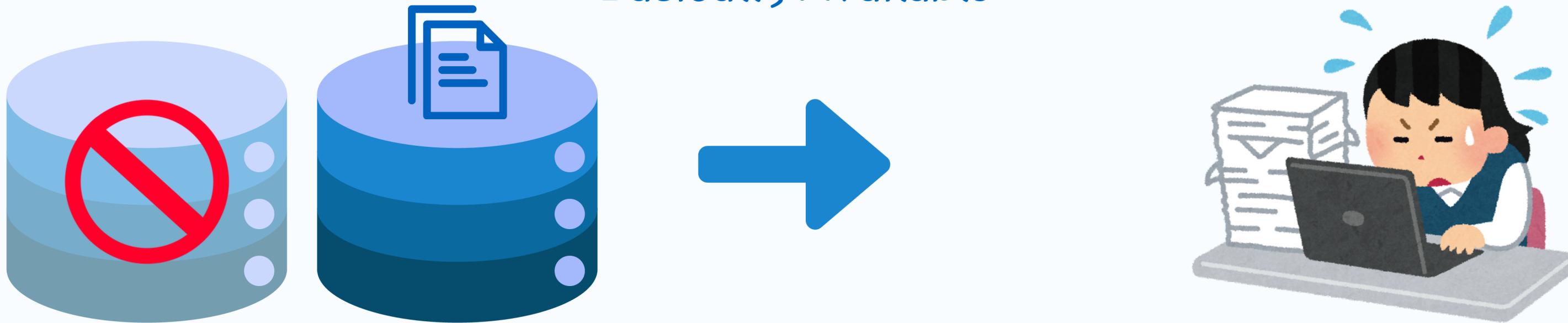
- 원자성(Atomicity)
- 일관성(Consistency)
- 격리성(Isolation)
- 내구성(Durability)

NoSQL - BASE

- Basically Available
- Soft-state
- Eventually consistency

기본적인 가용성

Basically Available



분산된 시스템의 특정 노드에 장애가 발생하더라도, 시스템 전체가 멈추지 않고 **항상 요청에 응답할 수 있는 상태**

응답이 항상 일관성을 가지지는 않음. 실패, 성공 상관없이 일단 보낼 수는 있는 상태

유연한 상태

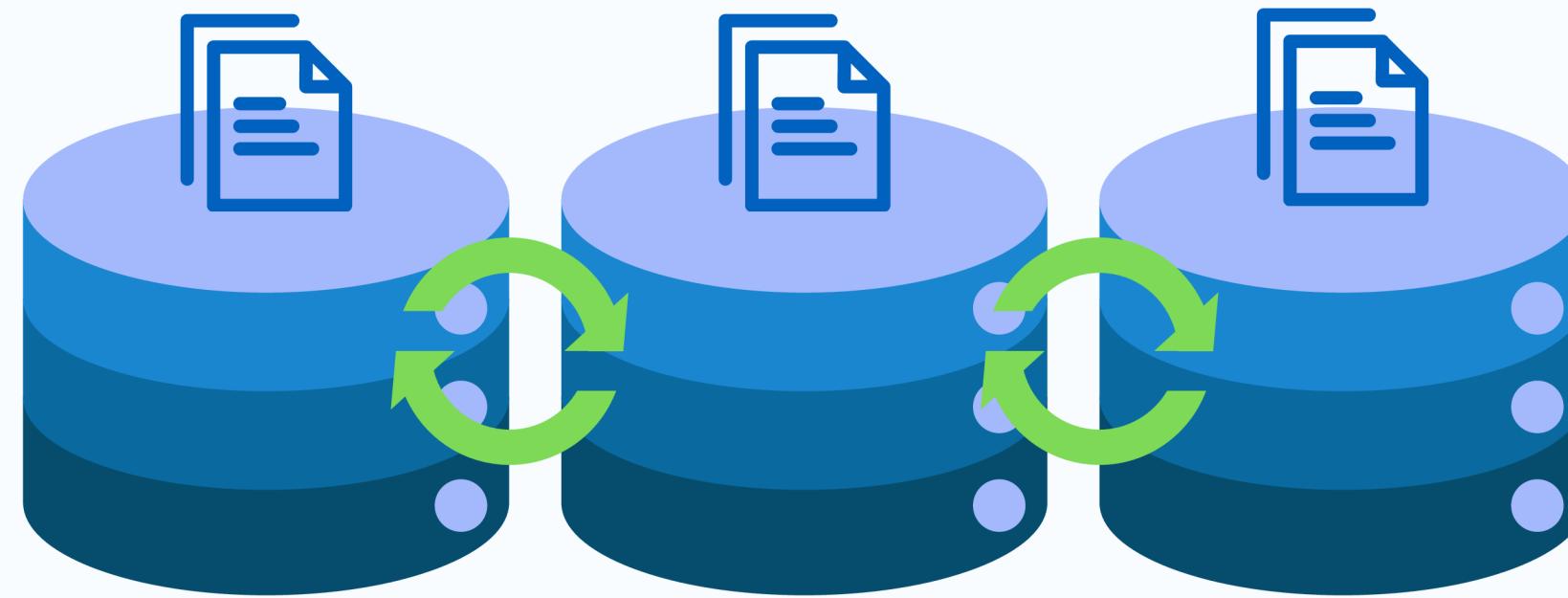


외부의 새로운 입력이 없더라도 시간이 지나면 **스**
스로 변할 수 있음

최종적 일관성을 맞추기 위해 내부적으로 계속 데이터를 동기화 하기 때문

최종적 일관성

Eventually consistency



시스템에 더 이상 새로운 데이터 수정이 없다면,
결국 언젠가는 모든 노드의 데이터가 같은 값으로
일관성을 유지하게 됨

그렇다면 단점도...

- 데이터 중복 발생 가능
- 데이터 처리 완결성 미보장
- 명확한 데이터 구조 보장 어려움
- 앱 레벨에서 스키마/중복 관리 필요

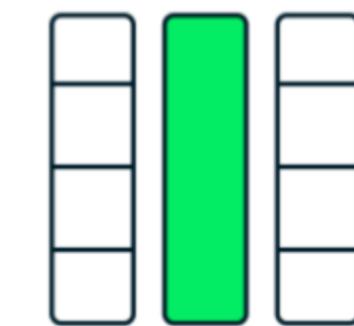
NoSQL 유형



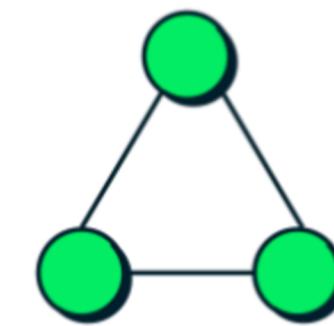
Key-value
database



Document
database



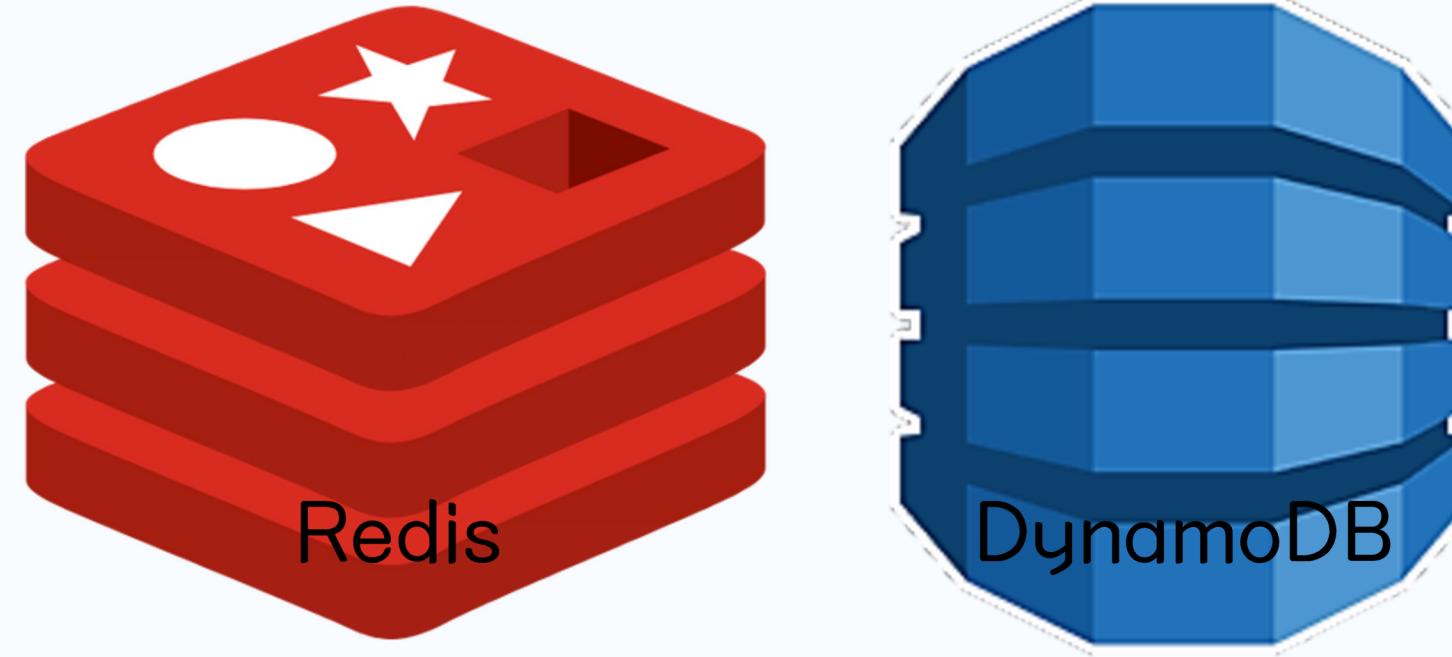
Wide-column
database



Graph database

Key-Value

키-값 저장소



JSON과 같이 Key와 Value 쌍으로 데이터를 저장하는 방식

Key-Value

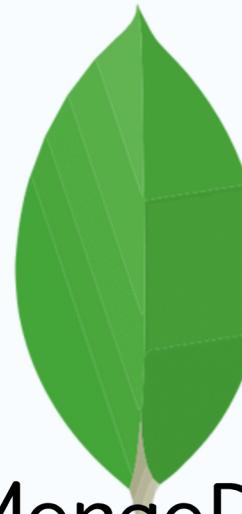
키-값 저장소

Key: user:12345

Value: {"name": "foo bar", "email": "foo@bar.com", "designation": "software developer"}

Document

문서 저장소



MongoDB



CouchDB

Key-Value 저장소의 확장으로 Value 부분에
JSON, XML 등의 구조화된 문서를 통으로 저장
할 수 있음

Document

문서 저장소

```
{  
  "id": "12345",  
  "name": "foo bar",  
  "email": "foo@bar.com",  
  "address": {  
    "street": "123 foo street",  
    "city": "some city",  
    "state": "some state",  
    "zip": "123456"  
  },  
  "hobbies": ["music", "guitar",  
  "reading"]  
}
```

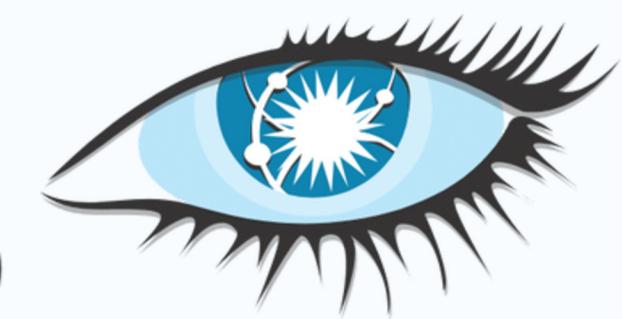
Column Family

컬럼 패밀리 저장소 (=Wide Column Store)



Apache

Hbase



Apache

Cassandra

RDBMS와 달리 데이터를 Row, Column 그리고
이를 묶는 Column 패밀리로 관리

Column Family

컬럼 패밀리 저장소 (=Wide Column Store)

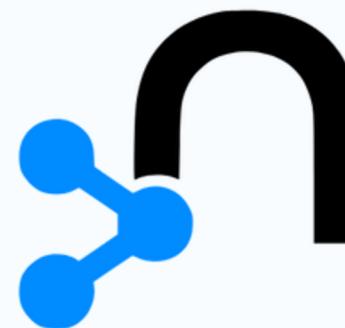
name	id	email	dob	city
Foo bar	12345	foo@bar.com		Some city
Carn Yale	34521	bar@foo.com	12-05-1972	

RDBMS와 달리 유연하기 때문에 서로 다른 열 집합을 가질 수 있음

dob, city를 보면 null로 유지될거 같지만, 열이 존재하지 않는 것으로 처리

Graph

그래프 데이터 베이스



Neo4j

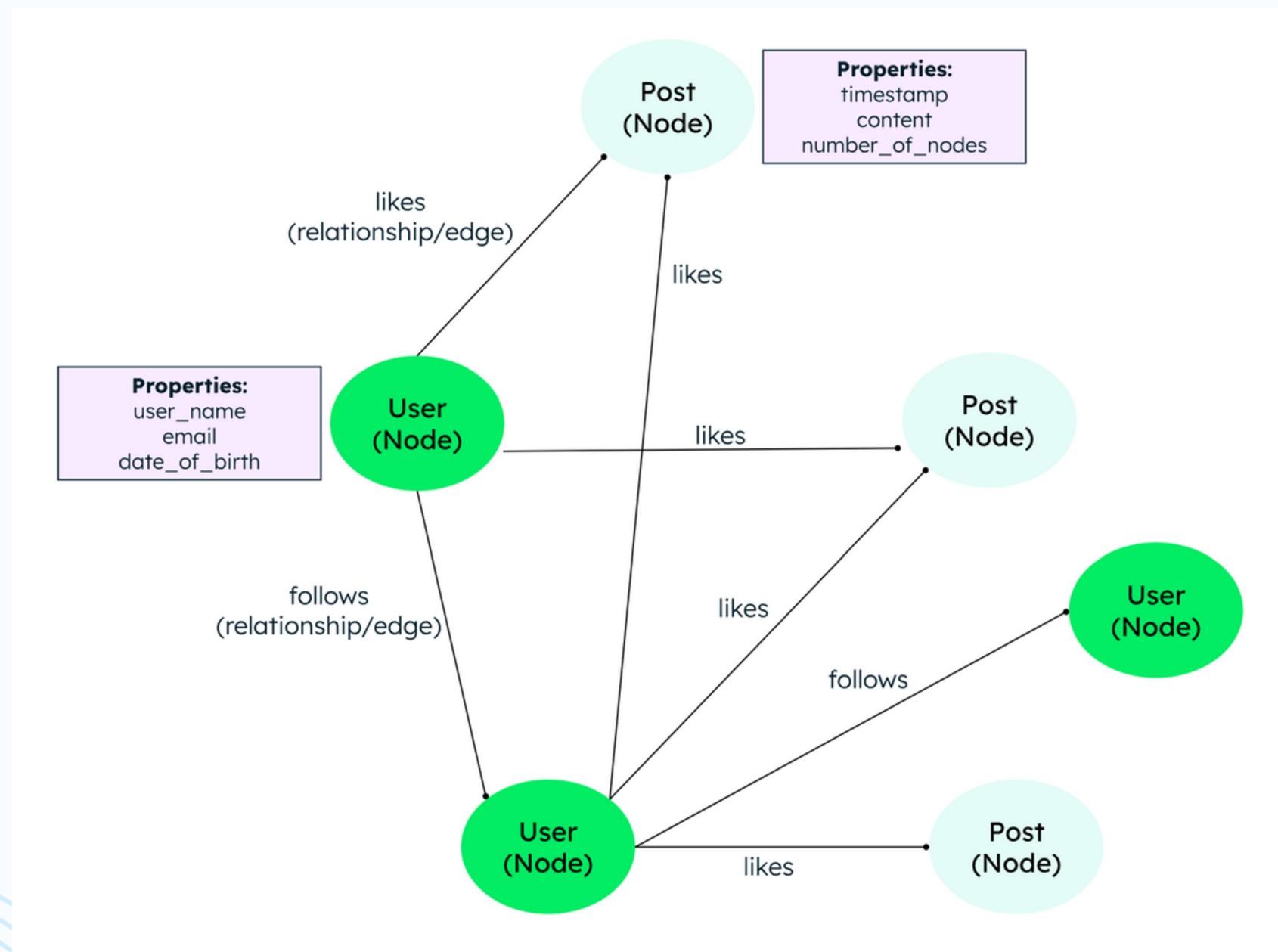


Amazon Neptune

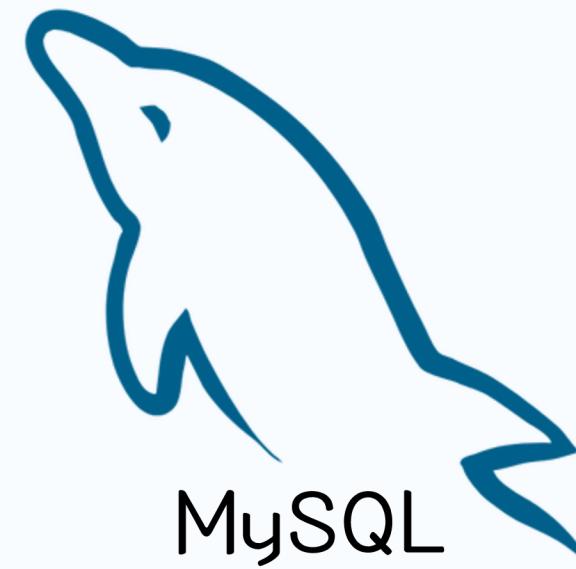
데이터를 노드로 보고, 노드를 잇는 간선(Edge)
로 표현하여 복잡한 관계를 표현하는 데 중점을 둠

Graph

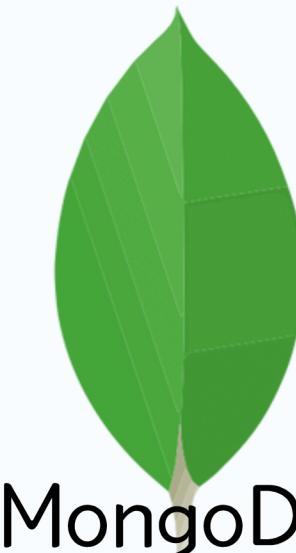
그래프 데이터 베이스



MySQL vs MongoDB



MySQL



MongoDB

정적 스키마 vs 동적 스키마

외래키 지원 0 vs 외래키 지원 X

기본키 별도 설정 vs 기본 키 자동 설정

사용자 수 > 사용자 수

성능 < 성능

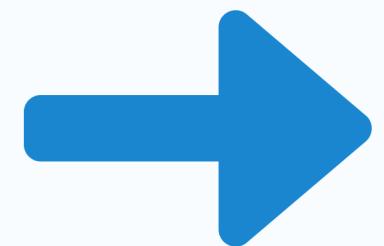
(업데이트는 MySQL이 빠름)

그럼 월 써야하나요?



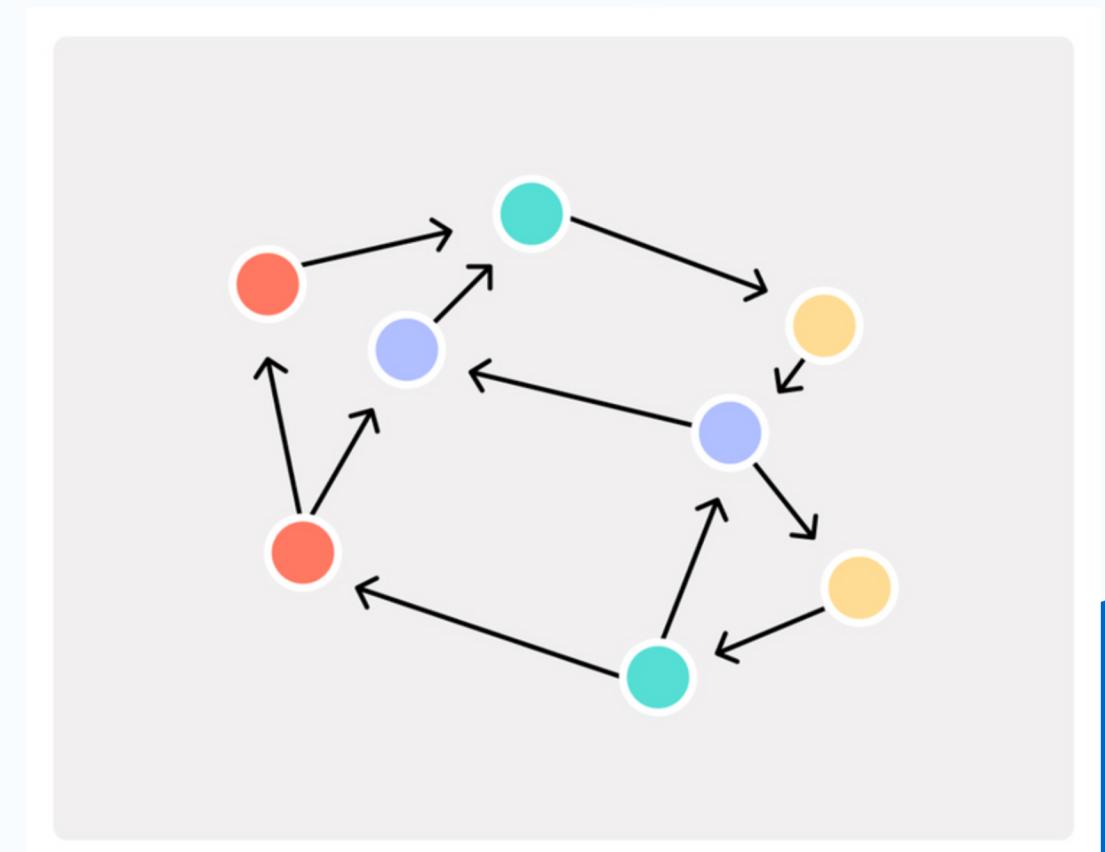
RDBMS 사용

정확성과 무결성이 중요
スキ마 변경이 적거나 확고
복잡한 쿼리 및 트랜잭션 관리가 필요
테이블 간의 관계가 중요



NoSQL 사용

높은 트래픽과 빠른 처리 속도
데이터 구조가 자주 변경되는 서비스
대규모의 비정형 또는 반정형 데이터 처리
수평적 확장이 필수

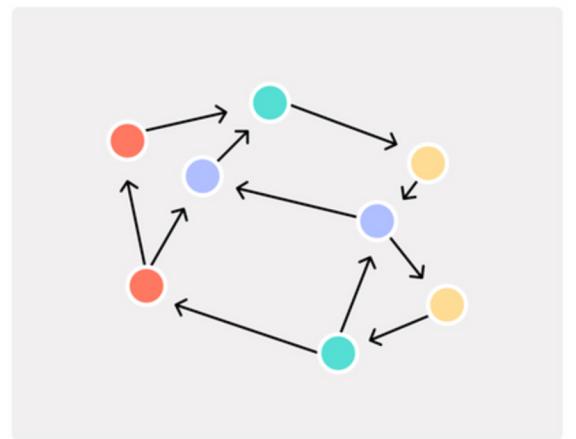
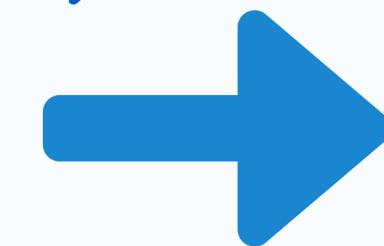


하이브리드 사용

예시 1) 전자 상거래 사이트

주문 및 결제 정보 RDBMS

장바구니 및 세션 캐싱 NoSQL(Redis)



예시2) 소셜 미디어 플랫폼

사용자 프로필 및 친구 관계 RDBMS

게시물, 좋아요, 댓글 등 NoSQL(MongoDB)



INTERVIEW



BASE가 뭔가요?



BASE

Soft-state

기본 가용성(Basically Available)

소프트 상태(Soft State)

최종 일관성(Eventually Consistent)

NoSQL의 속도가 빠른 이유



속도가 빠른 이유

JOIN이 필요 없는 모델

수평적 확장으로 부하 분산

완화된 일관성 모델(BASE)로 일단 처리하고 나중에 일관성을 맞춤

일관성



금융 거래 시스템에는 무엇을



RDBMS

일관성을 항상 유지해야함

지켜라 규칙

자주 변경되지 않음

일관성을 위해 가용성을 희생 가능(트랜잭션 등)

