

[DB] DB구조&설계 - RDB vs NoSQL

👤 생성자	👤 정희주
🕒 생성 일시	@2023년 12월 13일 오전 2:53
🏷 태그	

[RDB \(관계형 데이터베이스\)](#)

[NoSQL Not only SQL](#)

[RDB vs NoSQL 한눈에 비교하기](#)

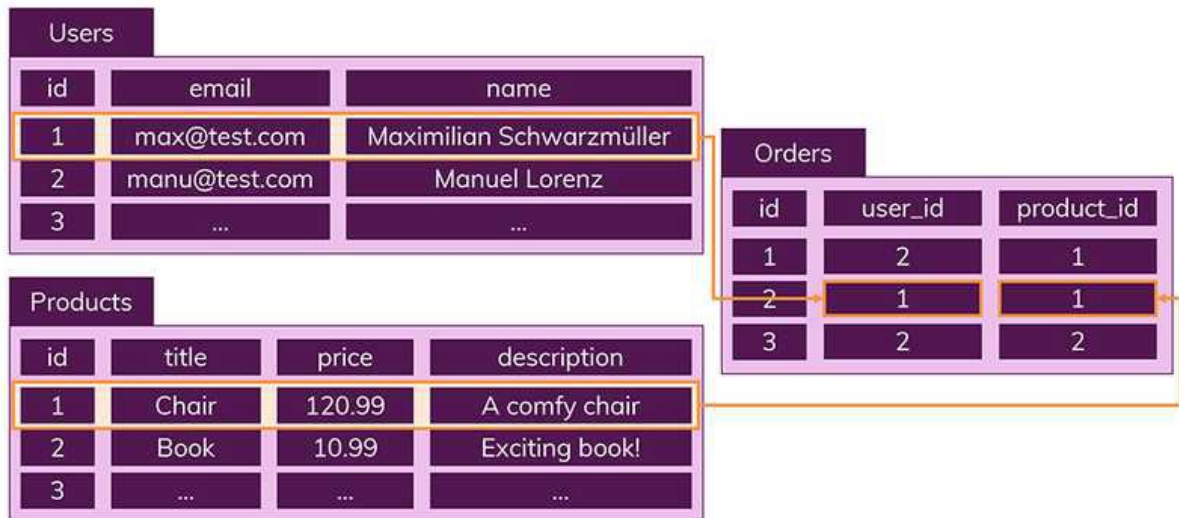
RDB (관계형 데이터베이스)

- SQL을 사용하면 RDBMS에서 데이터를 저장, 수정, 삭제 및 검색 할 수 있음
- 특징
 - 데이터는 사전에 엄격하게 **정의된 DB 스키마에 따라 테이블에 저장된다.**
 - 엄격한 스키마로 인해 **데이터 중복이 없기 때문에** 데이터 UPDATE가 많을 때 유리하다.
 - 여기서 말하는 데이터 중복은 한 칼럼에 같은 데이터를 여러개 입력해도 되고 이런 게 아니라, 정규화에서도 말하는 **테이블과 테이블 간의 중복 데이터**를 말하는 것으로 이해했다.
 - 데이터는 **관계를 통해 여러 테이블에 분산된다.**

데이터는 테이블에 레코드로 저장되는데, 각 테이블마다 명확하게 정의된 구조가 있다. 해당 구조는 필드의 이름과 데이터 유형으로 정의된다.

따라서 **스키마를 준수하지 않은 레코드는 테이블에 추가할 수 없다.** 즉, 스키마를 수정하지 않는 이상은 정해진 구조에 맞는 레코드만 추가가 가능한 것이 관계형 데이터베이스의 특징 중 하나다.

또한, 데이터의 중복을 피하기 위해 '관계'를 이용한다.



하나의 테이블에서 중복 없이 하나의 데이터만을 관리하기 때문에 다른 테이블에서 부정확한 데이터를 다룰 위험이 없어지는 장점이 있다.

NoSQL Not only SQL

- 특징
 - 데이터를 다양한 방식으로 표현할 수 있다.
 - 도큐먼트 모델 / 키-값 모델 / 그래프 모델 / 와이드 컬럼 모델 등 다양한 데이터 모델을 가짐
 - 스키마가 고정적이지 않고, 매우 유연하다
 - **데이터 중복을 허용하기 때문에**, 데이터 UPDATE시 모든 컬렉션에서 수정이 필요하다. 따라서, UPDATE가 적고 조회가 많은 경우 사용하는 것이 유리하다.
 - 수평적 확장이 용이하다
 - 연산이 빨라 실시간 연산에 적합하다

SQL은 정해진 스키마를 따르지 않으면 데이터 추가가 불가능했다. 하지만 NoSQL에서는 다른 구조의 데이터를 같은 컬렉션에 추가가 가능하다.

문서(documents)는 Json과 비슷한 형태로 가지고 있다. 관계형 데이터베이스처럼 여러 테이블에 나누어담지 않고, 관련 데이터를 동일한 '컬렉션'에 넣는다.

따라서 위 사진에 SQL에서 진행한 Orders, Users, Products 테이블로 나눈 것을 NoSQL에서는 Orders에 한꺼번에 포함해서 저장하게 된다.

따라서 여러 테이블에 조인할 필요없이 이미 필요한 모든 것을 갖춘 문서를 작성하는 것이 NoSQL이다. (NoSQL에는 조인이라는 개념이 존재하지 않음)

그러면 조인하고 싶을 때 NoSQL은 어떻게 할까?

컬렉션을 통해 데이터를 복제하여 각 컬렉션 일부분에 속하는 데이터를 정확하게 산출하도록 한다.

하지만 이러면 데이터가 중복되어 서로 영향을 줄 위험이 있다. 따라서 조인을 잘 사용하지 않고 자주 변경되지 않는 데이터일 때 NoSQL을 쓰면 상당히 효율적이다.

RDB vs NoSQL 한눈에 비교하기

특징과 장단점, 언제 사용하는지 알아두고,

단순히 몽고디비 써보고싶어서, 이런 장점이 있어서 선택했다가 아닌, 프로젝트 요구사항에 따라 trade-off를 고려해서 이 기술이 적합하다는 판단에 따라 선택했다 라는 명확한 근거가 있어야 한다.

	RDB (SQL)	NoSQL
데이터 저장 모델	table	json document / key-value / 그래프 등 여러가지
개발 목적	데이터 중복 감소	애자일 / 확장가능성 / 수정가능성
예시	Oracle, MySQL, PostgreSQL 등	MongoDB, DynamoDB 등
Schema	엄격한 데이터 구조	유연한 데이터 구조
장점	<ul style="list-style-type: none">- 명확한 데이터구조 보장- 무결성: 데이터 중복 없이 한 번만 저장- 데이터 중복이 없어서 데이터 update 용이	<ul style="list-style-type: none">- 유연하고 자유로운 데이터 구조- 새로운 필드 추가 자유로움- 수평적 확장(scale out) 용이
단점	<ul style="list-style-type: none">- 시스템이 커지면 Join문이 많은 복잡한 query가 필요- 수평적 확장이 까다로워 비용이 큰 수직적 확장(Scale up)이 주로 사용됨.- 데이터 구조가 유연하지 못함	<ul style="list-style-type: none">- 데이터 중복 발생 가능- 중복 데이터가 많기 때문에 데이터 변경 시 모든 컬렉션에서 수정이 필요함- 명확한 데이터구조 보장 X

	RDB (SQL)	NoSQL
사용	<ul style="list-style-type: none"> - 데이터 구조가 변경될 여지가 없이 명확한 경우 - 데이터 update가 잦은 시스템 (중복 데이터가 없으므로 변경에 유리) 	<ul style="list-style-type: none"> - 정확한 데이터 구조가 정해지지 않은 경우 - Update가 자주 이루어지지 않는 경우 (조회가 많은 경우) - 데이터 양이 매우 많아 수평적 확장 (Scale out)이 필요한 경우