

RDBMS 와 NoSQL

웹 앱을 개발할 때, 데이터베이스를 선택할 때 고민하게 된다.

MySQL과 같은 RDBMS을 사용할까? 아니면 MongoDB와 같은 NoSQL을 사용할까?

보통 Spring에서 개발할 때는 MySQL을, Node.js에서는 MongoDB를 주로 사용했을 것이다.

하지만 그냥 단순히 프레임워크에 따라 결정하는 것이 아니다. 프로젝트를 진행하기에 앞서 적합한 데이터베이스를 택해야 한다. 차이점을 알아보자

SQL (관계형 DB)

SQL을 사용하면 RDBMS에서 데이터를 저장, 수정, 삭제 및 검색 할 수 있음

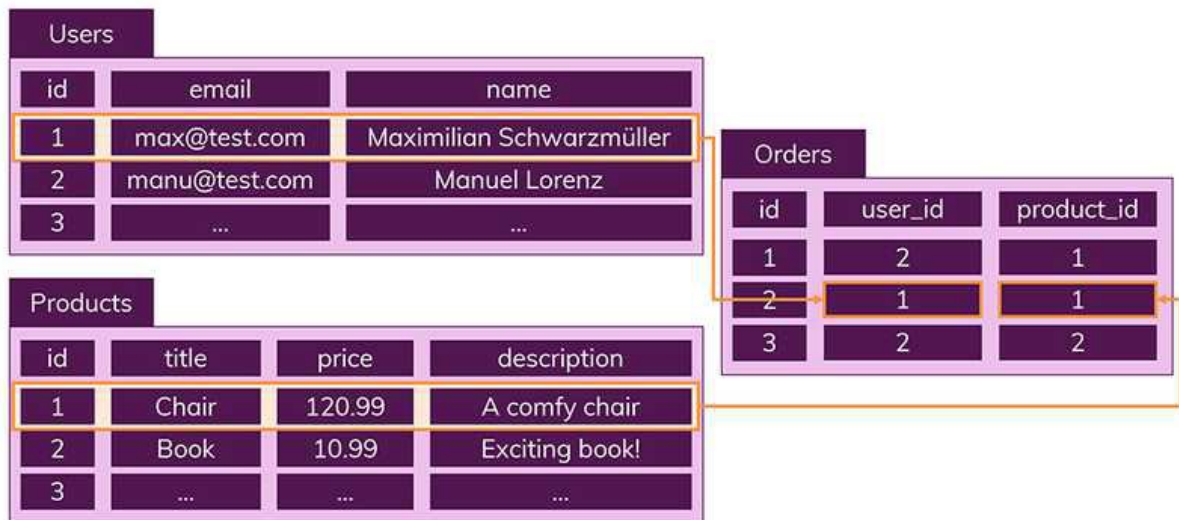
관계형 데이터베이스에는 핵심적인 두 가지 특징이 있다.

- 데이터는 정해진 데이터 스키마에 따라 테이블에 저장된다.
- 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.



NoSQL (비관계형 DB)

말그대로 관계형 DB의 반대다.
스키마도 없고, 관계도 없다!

NoSQL에서는 레코드를 문서(documents)라고 부른다.

여기서 SQL과 핵심적인 차이가 있는데, 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.

확장 개념

두 데이터베이스를 비교할 때 중요한 Scaling 개념도 존재한다.

데이터베이스 서버의 확장성은 '수직적' 확장과 '수평적' 확장으로 나뉘어진다.

- 수직적 확장 : 단순히 데이터베이스 서버의 성능을 향상시키는 것 (ex. CPU 업그레이드)
- 수평적 확장 : 더 많은 서버가 추가되고 데이터베이스가 전체적으로 분산됨을 의미 (하나의 데이터베이스에서 작동하지만 여러 호스트에서 작동)

데이터 저장 방식으로 인해 SQL 데이터베이스는 일반적으로 수직적 확장만 지원함

수평적 확장은 NoSQL 데이터베이스에서만 가능

둘다 훌륭한 솔루션이고 어떤 데이터를 다루느냐에 따라 선택을 고려해야한다.

RDBMS 장점

- 명확하게 정의된 스키마, 데이터 무결성 보장
- 관계는 각 데이터를 중복없이 한번만 저장

SQL 단점

- 덜 유연함. 데이터 스키마를 사전에 계획하고 알려야 함. (나중에 수정하기 힘들)
- 관계를 맺고 있어서 조인문이 많은 복잡한 쿼리가 만들어질 수 있음
- 대체로 수직적 확장만 가능함

NoSQL 장점

- 스키마가 없어서 유연함. 언제든지 저장된 데이터를 조정하고 새로운 필드 추가 가능
- 데이터는 애플리케이션이 필요로 하는 형식으로 저장됨. 데이터 읽어오는 속도 빨라짐
- 수직 및 수평 확장이 가능해서 애플리케이션이 발생시키는 모든 읽기/쓰기 요청 처리 가능

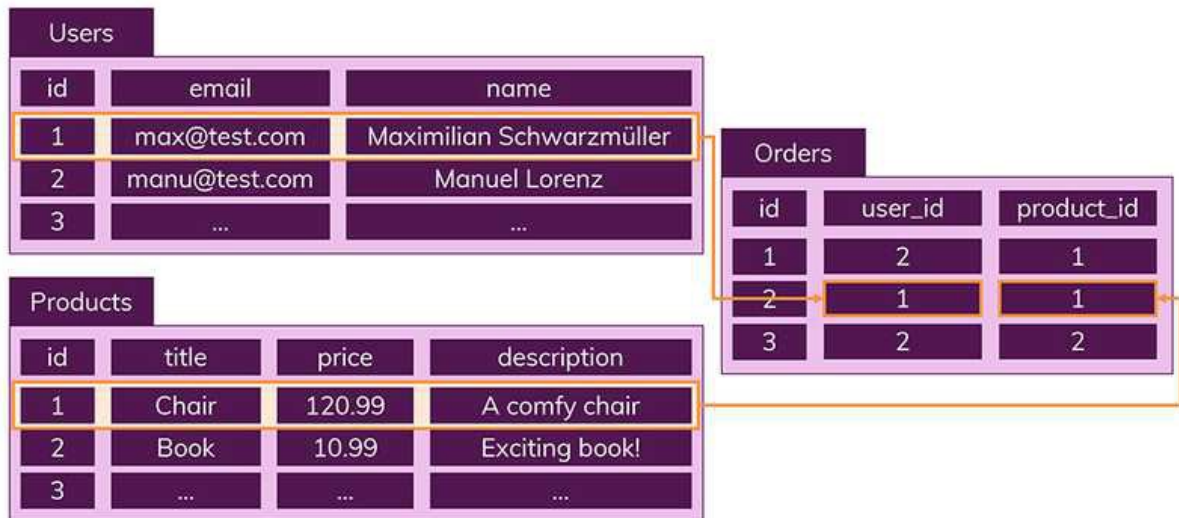
NoSQL 단점

- 유연성으로 인해 데이터 구조 결정을 미루게 될 수 있음
- 데이터 중복을 계속 업데이트 해야 함
- 데이터가 여러 컬렉션에 중복되어 있기 때문에 수정 시 모든 컬렉션에서 수행해야 함
(SQL에서는 중복 데이터가 없으므로 한번만 수행이 가능)

RDBMS 사용이 더 좋을 때

관계를 맺고 있는 데이터가 자주 변경되는 애플리케이션의 경우 NoSQL에서는 여러 컬렉션을 모두 수정해야 하기 때문에 비효율적

- 변경될 여지가 없고, 명확한 스키마가 사용자와 데이터에게 중요한 경우



NoSQL 데이터베이스 사용이 더 좋을 때

- 정확한 데이터 구조를 알 수 없거나 변경/확장 될 수 있는 경우
- 읽기를 자주 하지만, 데이터 변경은 자주 없는 경우
- 데이터베이스를 수평으로 확장해야 하는 경우 (막대한 양의 데이터를 다뤄야 하는 경우)

하나의 제시 방법이지 완전한 정답이 정해져 있는 것은 아니다.

SQL을 선택해서 복잡한 JOIN문을 만들지 않도록 설계하여 단점을 없앨 수도 있고

NoSQL을 선택해서 중복 데이터를 줄이는 방법으로 설계해서 단점을 없앨 수도 있다.