SQL vs NoSQL 비교하기

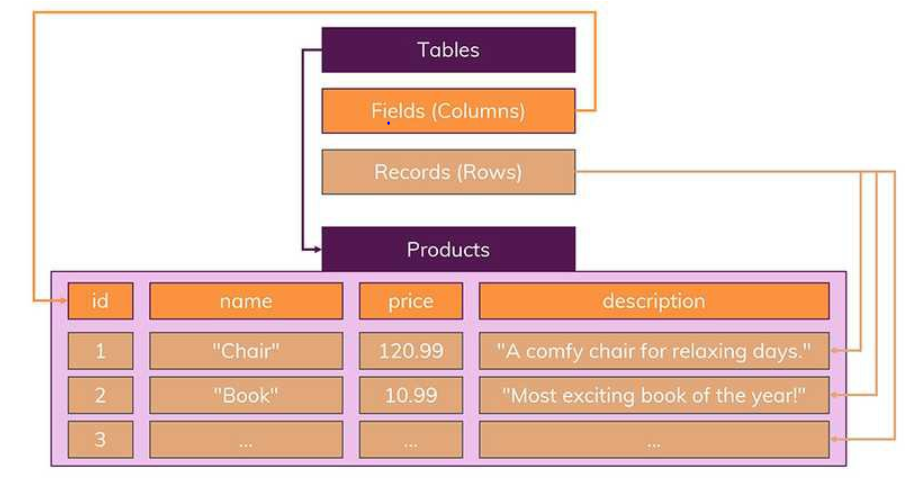
SQL은 '구조화 된 쿼리 언어 (Structured Query Language)'의 약자로 관계형 데이터베이스를 관리할 때 사용하는 언어이다. 그렇지만 통상적으로 SQL을 관계형 데이터베이스를 가리킬 때 사용하기도 한다. 관계형 데이터베이스가 데이터베이스의 큰 부분을 차지하는 만큼, SQL과 NoSQL로 데이터베이스를 나누어 보는 시각이 일반적이고, 각각의 특성을 정리해 보려고 한다.

주요 특징 비교

**SQL 관계형 데이터베이스**

1. 엄격한 스키마

* 관계형 데이터베이스에서 데이터는 테이블(table)에 저장되며, 정의된 구조(structure)를 따라야한다.
* 구조(structure)는 필드의 이름과 데이터 유형으로 정의됨

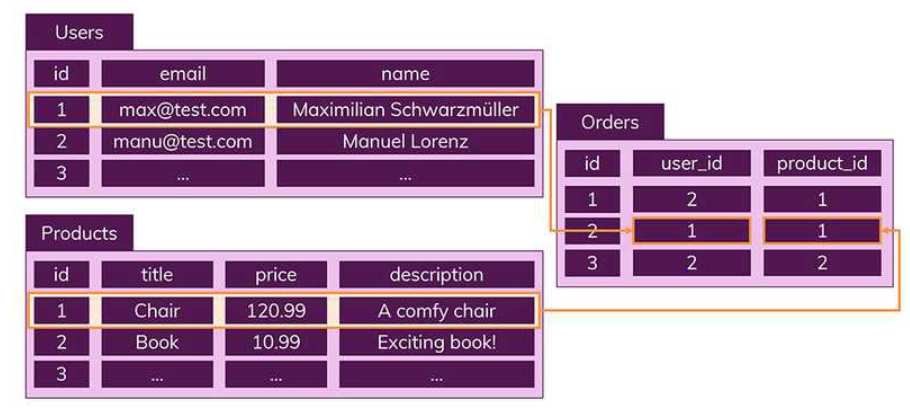


(주황색 부분이 필드, 필드 아래의 데이터가 레코드, 전체가 테이블)

스키마(structure)를 준수하지 않는 데이터는 추가할 수 없다.

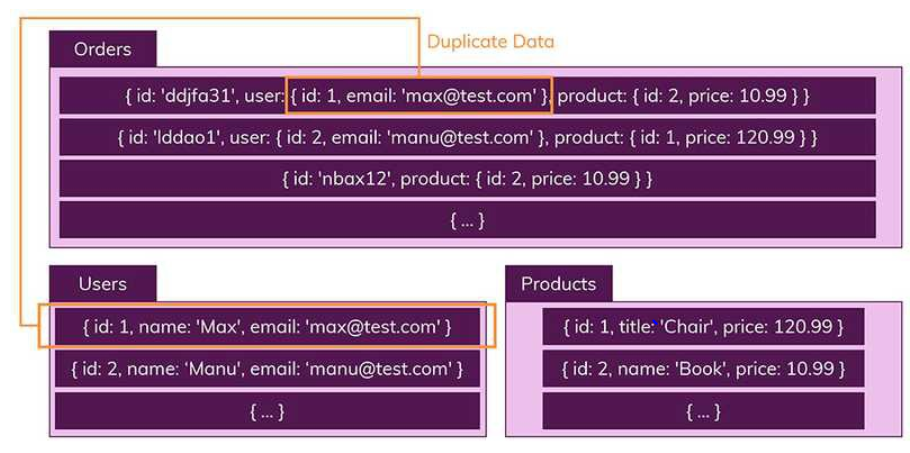
1. 관계

* 데이터들을 여러 개의 테이블에 나눈다.
* 각각의 테이블들은 다른 테이블에 저장되지 않은 데이터만을 가진다. (중복 x)



**NoSQL 비관계형 데이터베이스**

* NoSQL에서는 다른 구조의 데이터를 같은 컬렉션(SQL에서의 테이블)에 추가 가능
* 
* 일반적으로 관련 데이터를 동일한 컬렉션에 넣는다.
* 따라서 다른 컬렉션에(join) 하는 일이 거의 없다.
* 그 대신 데이터를 복제하여 각 컬렉션에 필요한 데이터를 산출한다.



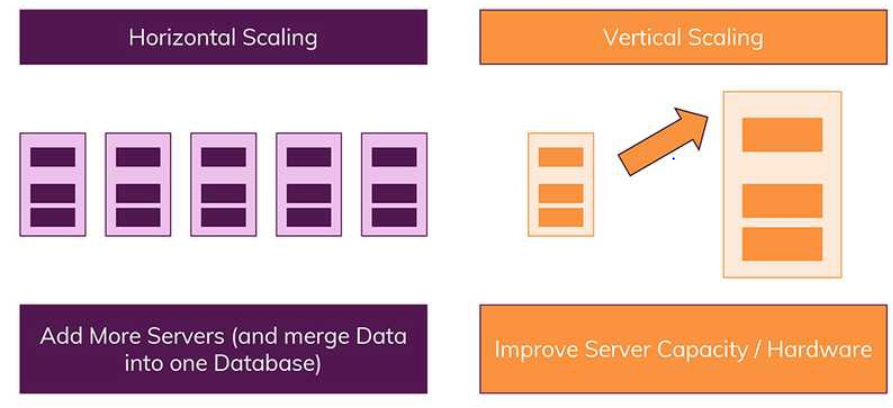
* 데이터가 중복되기 때문에, 한 쪽 데이터만 업데이트 하는 등의 위험이 따른다.

**수직적, 수평적 확장**

-확장(Scaling)은 데이터 베이스 서버의 확장성을 의미한다.

수직적 확장이란 데이터베이스 서버 성능의 향상을 의미

수평적 확장은 더 많은 서버가 추가되고 데이터베이스가 전체적으로 분산됨을 의미



SQL은 일반적으로 수직적 확장 만을 지원, sharding의 구현 힘듬

NoSQL은 수평적 확장도 가능, sharding 을 지원하여 데이터베이스 분리가 쉽다.

장단점 정리

**SQL 장점**

명확하게 정의 된 스키마(structure) , 데이터 무결성

관계를 통해, 데이터 중복 x

**단점**

데이터 스키마가 사전에 계획되어 있어야 해서 유연성이 떨어짐

관계를 맺고 있기 때문에, join을 해야하는데 복잡할 수 있음.

수평적 확장이 어렵기 때문에 데이터 양에 한계가 있음

**NoSQL의 장점**

데이터 조정이 쉬움(유연함)

Join이 없기 때문에, 데이터를 읽어오는 것이 쉬움.

수직 및 수평 확장이 가능하므로 데이터베이스가 애플리케이션에서 발생시키는 모든 읽기 / 쓰기 요청 처리 가능

**NoSQL의 단점**

유연성 때문에, 데이터 구조 결정을 하지 못하고 미루게 될 수 있음

데이터가 중복되어 있기 때문에, 수정할 때 모든 컬렉션에서 수정해야함.

* 관계를 맺고 있는 데이터가 자주 변경(수정)되는 애플리케이션일 경우

변경될 여지가 없고, 명확한 스키마가 사용자와 데이터에게 중요한 경우 에는 SQL이 유용함

* 정확한 데이터 구조를 알 수 없어서 변경 / 확장 될 수 있는 경우

읽기(read)처리를 자주하지만, 데이터를 자주 변경(update)하지 않는 경우

막대한 양의 데이터를 다루어야 하는 경우 에는 NoSQL이 유용함

NoSQL은 비관계형 데이터베이스로, 관계형 데이터베이스가 아닌 데이터베이스이다. 이 안에서도 키-벨류형, 컬럼형, 문서형, 그래프형으로 나뉘고, 또 각각을 지원하는 소프트웨어가 여러가지 존재한다.

사실 많은 종류의 소프트웨어가 있어서 전부 조사하지 못했거니와, 설명을 읽어도 잘 이해가 되지 않아 얻은 자료 사진만 제시하겠습니다.(죄송합니다 ㅜ). 우선 관계형으로 할지, 비관계형으로 할지 정하고, 비관계형으로 한다면 더 조사해보면 될 듯 합니다.

