

Redis

2020.10.09

이지현

SQL vs NoSQL

SQL



NoSQL

Key-Value Store



Document DB



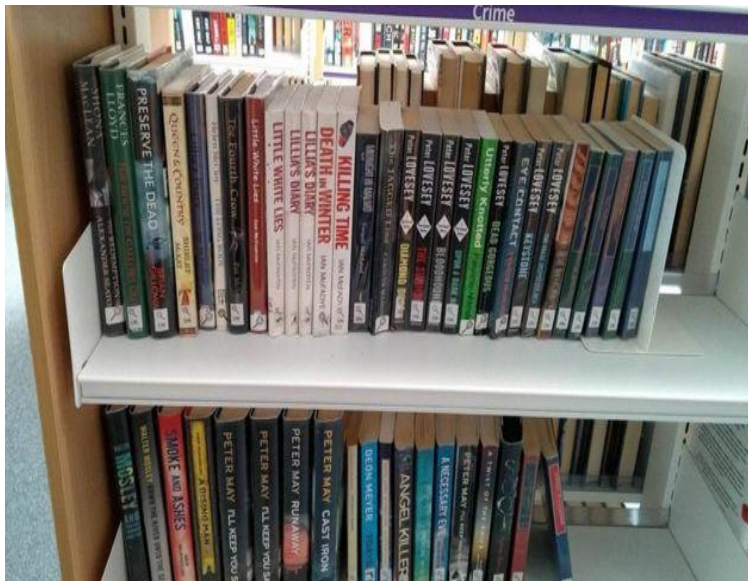
Column family Store



Graph Database



SQL vs NoSQL



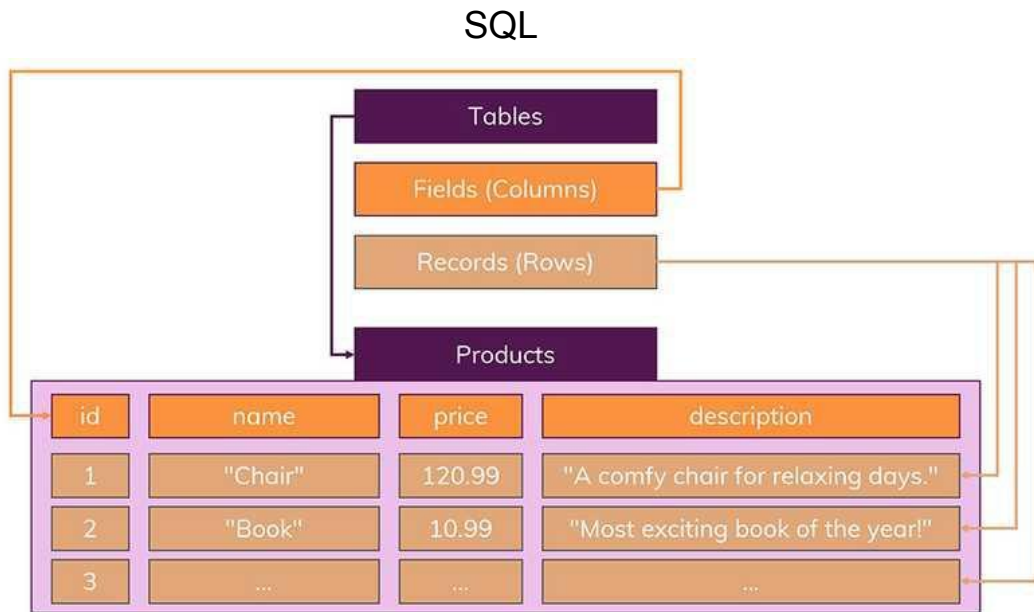
출처 : <https://www.bbc.com/korean/news-52423098>



출처 : <https://m.blog.naver.com/gonckey/221777289604>

SQL vs NoSQL

- 스키마 정의에 일치하지 않은 데이터는 저장할 수 없음
- Table 에 Record 로 저장
- Table 마다 구조(Structure) 를 가짐



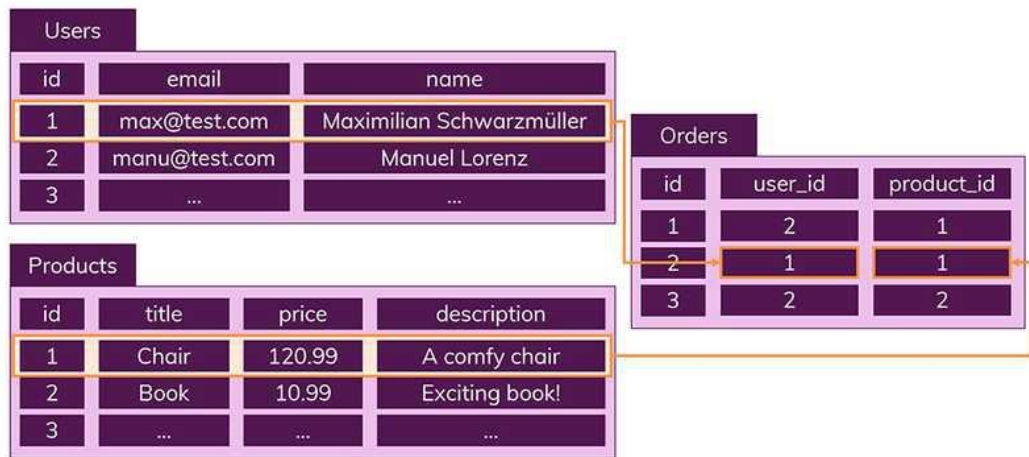
출처 : <https://siyoon210.tistory.com/130>

* 스키마 : 개체, 속성, 관계에 대한 정의와 제약조건

SQL vs NoSQL

- 데이터를 여러 **Table** 에 나누어
중복없이 저장
- **Table** 간의 관계를 나타냄
- 다른 **Table** 에 데이터를
저장하기 위해 구조(**Structure**)
정의가 필요

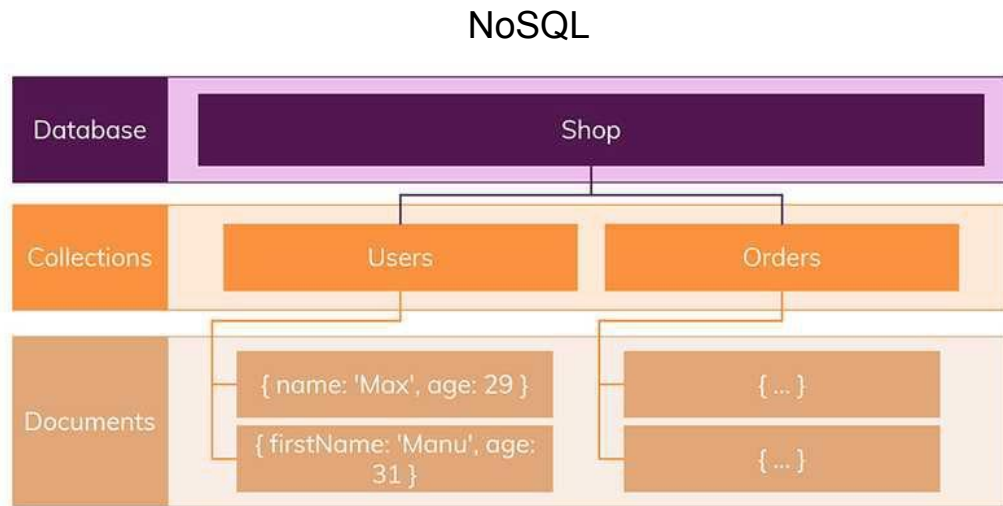
SQL



출처 : <https://siyoon210.tistory.com/130>

SQL vs NoSQL

- 스키마 , 관계 개념 없음
- Table (SQL) => Collections (NoSQL)
Record (SQL) => Documents (NoSQL)
- 구조에 따르지 않고 데이터를 저장할 수 있음
- Documents 는 JSON 과 비슷한 형태

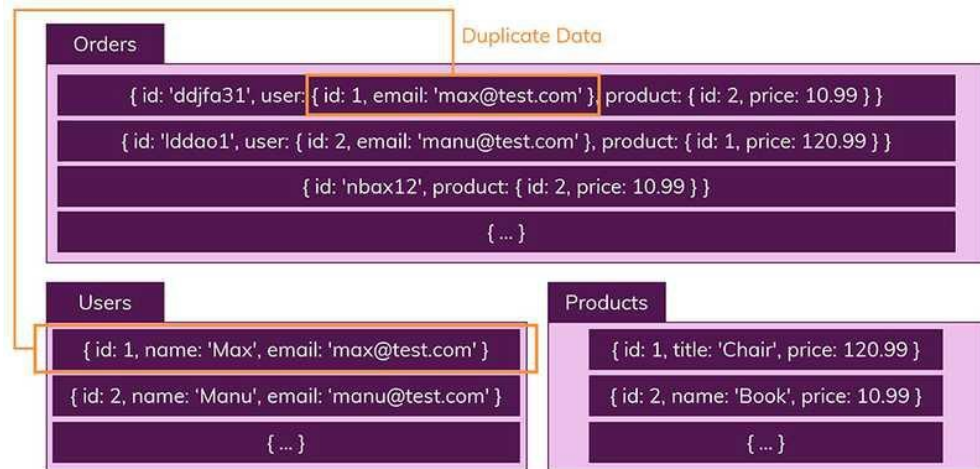


출처 : <https://siyoon210.tistory.com/130>

SQL vs NoSQL

- Join 개념 없음
- 데이터를 여러 Collections 에 나누어 저장하지 않고 필요한 모든 데이터를 포함시켜 저장
- 중복된 데이터를 수정할 때 동일하게 업데이트 필요

NoSQL



출처 : <https://siyoon210.tistory.com/130>

SQL vs NoSQL

SQL



NoSQL

Key-Value Store



Document DB



Column family Store



Graph Database



Redis ?

- Open source, NoSQL Database
- **Key-Value** 구조로 데이터 저장 및 관리
- 디스크가 아닌 메모리 기반
- 빠른 처리 속도



* Redis : Remote Dictionary Server 의 약자

* NoSQL : Not Only SQL, 비관계형 Database

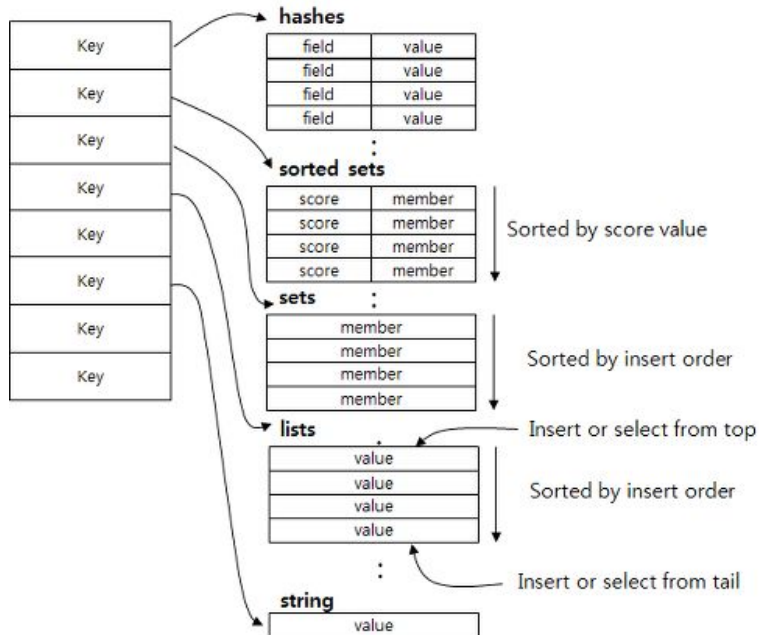
Redis의 특징

- 다양한 Value data type을 지원
 - string, list, set, sorted set, hash
(list data 입력과 삭제가 MySQL 보다 10배 빠름)
- 서버를 종료해도 Data는 사라지지 않음
 - Disk에서 읽어서 Memory에 올림
Snapshot 기능 제공 (파일로 저장, 복구 가능)
 - 명시적으로 삭제 명령을 하지 않으면
데이터는 영구적 보존

* Snapshot : 메모리에 있는 데이터들을
디스크에

옮겨 담는 방식

Redis structure



Data type: String

- 최대 512 MB
- 문자열 뿐만 아니라 **integer**와 같은 숫자나 JPEG 같은 Binary File 까지 저장 가능

String

"Hello, World!"

Data type: Set

- 데이터 정렬, 중복되지 않은 집합
- 여러개의 값을 하나의 **Value**에 저장
- 교집합, 합집합, 차이를 매우 빠른 시간내에 추출

Set

{A, C, B, E, D}

Data type: Sorted set

- Set 에 score 가 추가된 데이터 형
- score 기준으로 오름차순 정렬
- score는 실수 값을 가지며 중복될 수 있음

Sorted set

{ A: 0.1, B: 0.3, C: 100, D: 1337 }

Sorted by score value

Data type: Hash

- value 내에 “field name”: string value 쌍으로 이루어진 데이터 형

Hash

```
{ A: “Hi”, B: “Hello”, C: “Bye” }
```

- 객체를 나타내는데 사용 가능

Data type: List

- 일종의 양방향 Linked List
- PUSH/POP 연산 또는 index 값을 이용하여 데이터 삽입 또는 삭제 가능

List

[A <-> B <-> C <-> D <-> E]

Sorted by insert order

Reference

- **Dave Nielsen: Top 5 uses of Redis as a Database | PyData Seattle 2015**
(<https://www.youtube.com/watch?v=jTTIBc2-T9Q>)
- **레디스(Redis)란 무엇인가?**
(<https://medium.com/@jyejye9201/%EB%A0%88%EB%94%94%EC%8A%A4-redis-%EB%9E%80-%EB%A C%B4%EC%97%87%EC%9D%B8%EA%B0%80-2b7af75fa818>)
- **In memory dictionary Redis 소개**
(<https://bcho.tistory.com/654>)

감사합니다