RDBMS vs NoSQL

용어 정리

Database : 여러 사람이 공유하여 사용할 목적으로 체계화해 통합, 관리하는 **데이터의 집합**

Relational Database : **키**(key)와 **값**(value)들의 간단한 **관계**를 테이블화 시킨 매우 간단한 원칙의 데이터베이스

DBMS : 데이터베이스를 만들고 관리하기위한 **시스템 소프트웨어 (관리 시스템)**

RDBMS : 관계형 데이터베이스에 데이터를 저장하는 관계형 모델을 기반으로하는 데이터베이스 관리 시스템

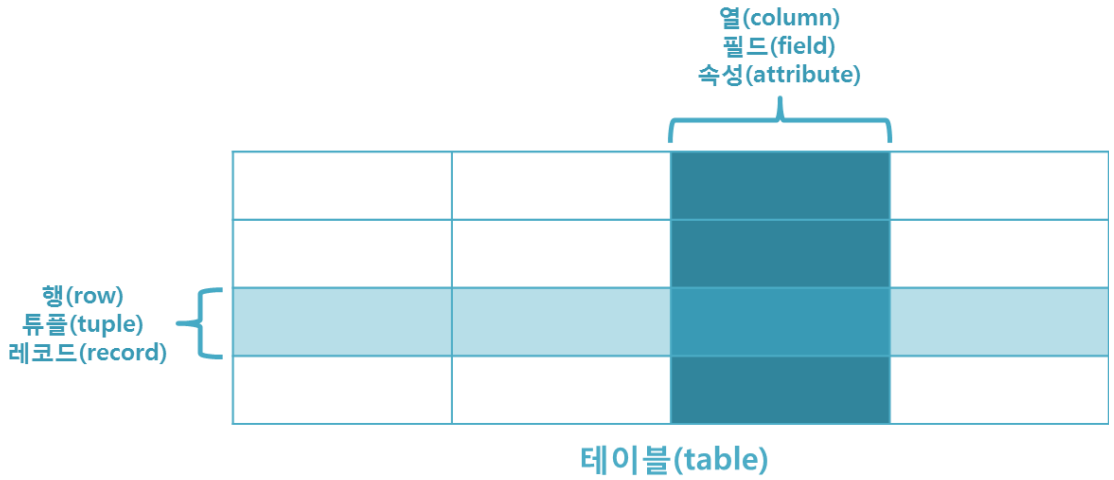
SQL : 관계형 데이터베이스를 처리하기위한 **표준 언어**

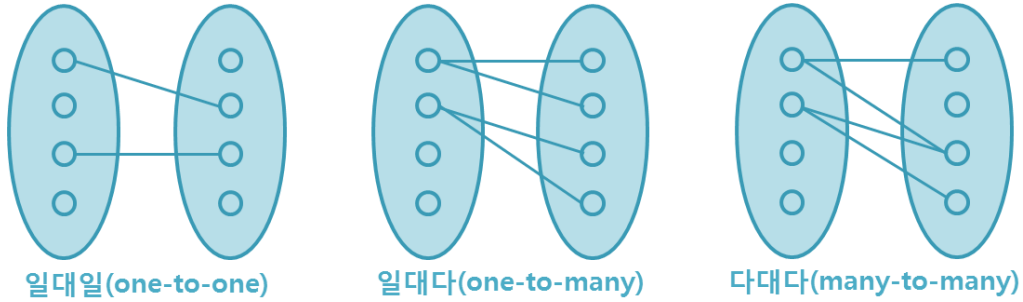
NoSQL
전통적인 관계형 데이터베이스 보다 덜 제한적인 일관성 모델을 이용하는 데이터의 저장 및 검색을 위한 매커니즘을 제공
No SQL, 즉 SQL을 사용하지 않는다는 의미보다는 Not Only SQL, SQL 계열 쿼리 언어를 사용할 수 있다는 사실을 강조하는 분류

NoSQL -> No RDMBS? ~~예외) BerkelyDB~~
No RDBMS -> NoSQL? ~~예외) KV store~~

각 위키, NoSQL은 sds 페이지

RDB





열(column)
각각의 열은 유일한 이름과 자신만의 타입을 가지고 있다.
이러한 열은 필드(field) 또는 속성(attribute)이라고도 불린다.

행(row)
관계된 데이터의 묶음을 의미하며, 한 테이블의 모든 행은 같은 수의 열을 가지고 있다.
이러한 행은 튜플(tuple) 또는 레코드(record)라고도 불린다.

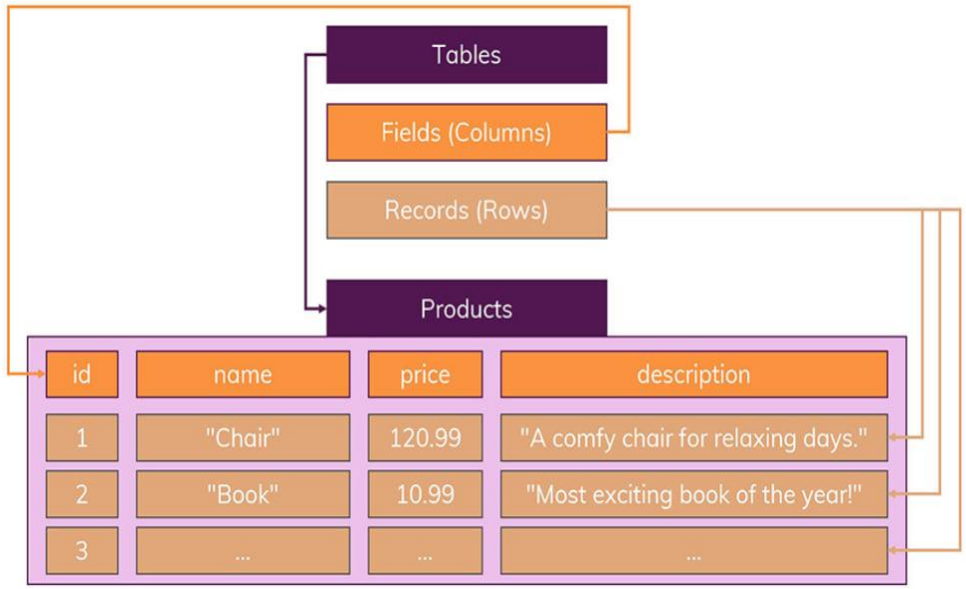
값(value)
테이블은 각각의 행과 열에 대응하는 값을 가지고 있다.
이러한 값은 열의 타입에 맞는 값이어야 한다.

키(key)
테이블에서 행의 식별자로 이용되는 열을 키(key) 또는 기본 키(primary key)라고 한다.
즉, 테이블에 저장된 레코드를 고유하게 식별하는 후보 키(candidate key) 중에서 데이터베이스 설계자가 지정한 속성을 의미

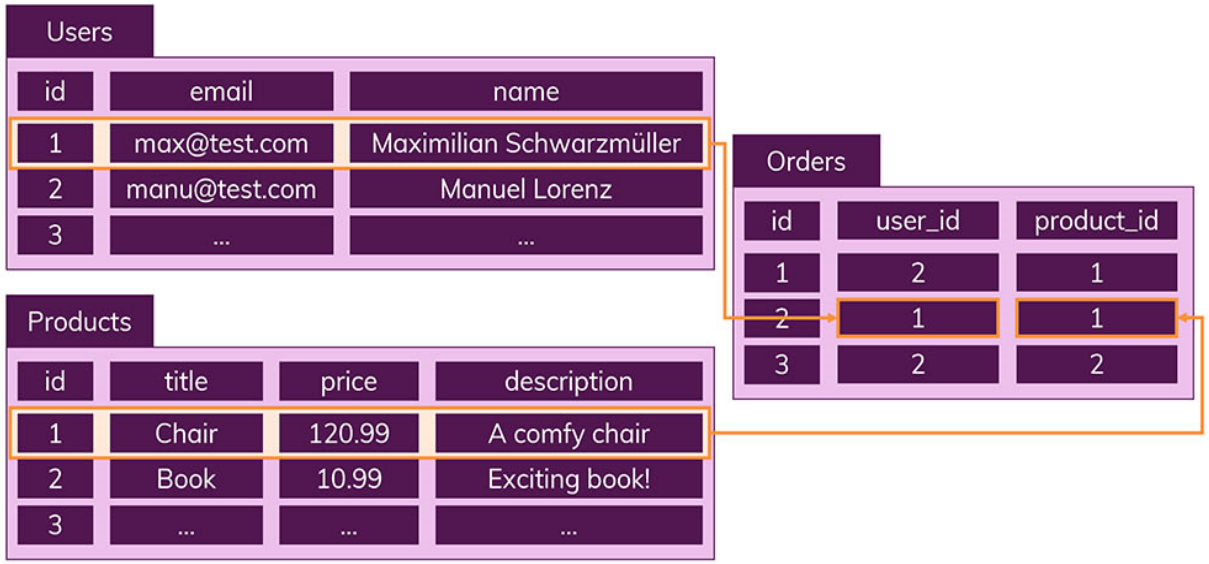
출처 TCPschool

RDB

엄격한 스키마 (schema, data structure)



관계 (Relation)



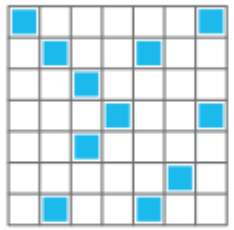
출처 academind, 번역

NoSQL

Not Only SQL, 데이터를 저장하는데에는 SQL 외에 다른 방법들도 있다.

<공통적인 성향>
- **대부분**은 **관계형 모델 X** (클러스터에서 실행할 목적)
- **스키마 없이** 동작 (자유롭게 필드 추가 가능)

NoSQL Database



Column-Family

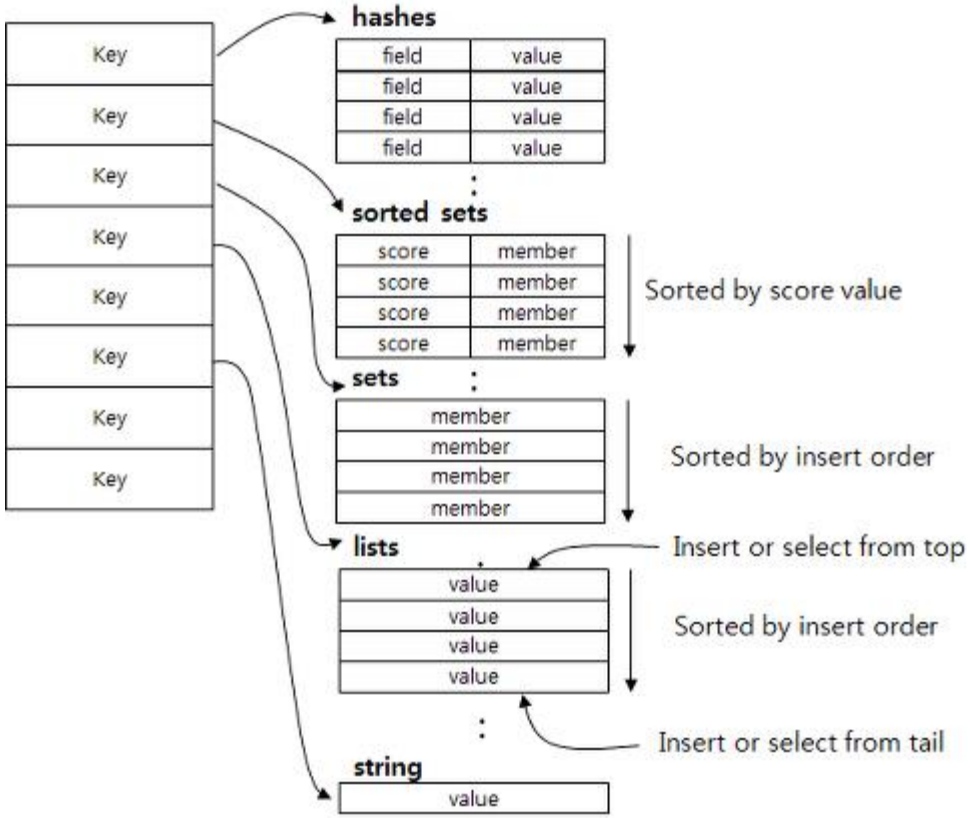
Graph

Document

Key-Value

출처 devcom

NoSQL – Key-Value



대표적인 예시 Redis

키를 고유한 식별자로 사용하는
키-값 쌍의 집합으로 데이터를 저장

데이터를 하나의 불투명한 컬렉션으로 처리하여
각 레코드마다 각기 다른 필드를 가질 수 있음
(객체지향에 밀접, 유연성 제공)

사용 사례 : 세션 스토어
세션은 사용자가 로그인~로그아웃 혹은 세션초과까지 활성화
그 사이 애플리케이션은 모든 세션 관련 데이터를 주 메모리 또는 데이터베이스에 저장
대체로 사용자 프로필 정보, 메시지, 개인 설정 데이터 및 테마, 권장 사항, 사용자 대상 판촉 및 할인이 세션 데이터에 해당된다.
각 사용자 세션에는 고유 식별자가 있다.
세션 데이터는 기본 키 이외의 다른 키에 의해 쿼리되지 않으므로 속도가 빠른 키-값 저장소가 세션 데이터에 더 적합합니다. 일반적으로 키-값 데이터베이스는 관계형 데이터베이스에 비하여 페이지당 제공하는 오버헤드가 적다.

출처 aws, 조대협님 블로그

NoSQL – Wide Column store (Column-Family)

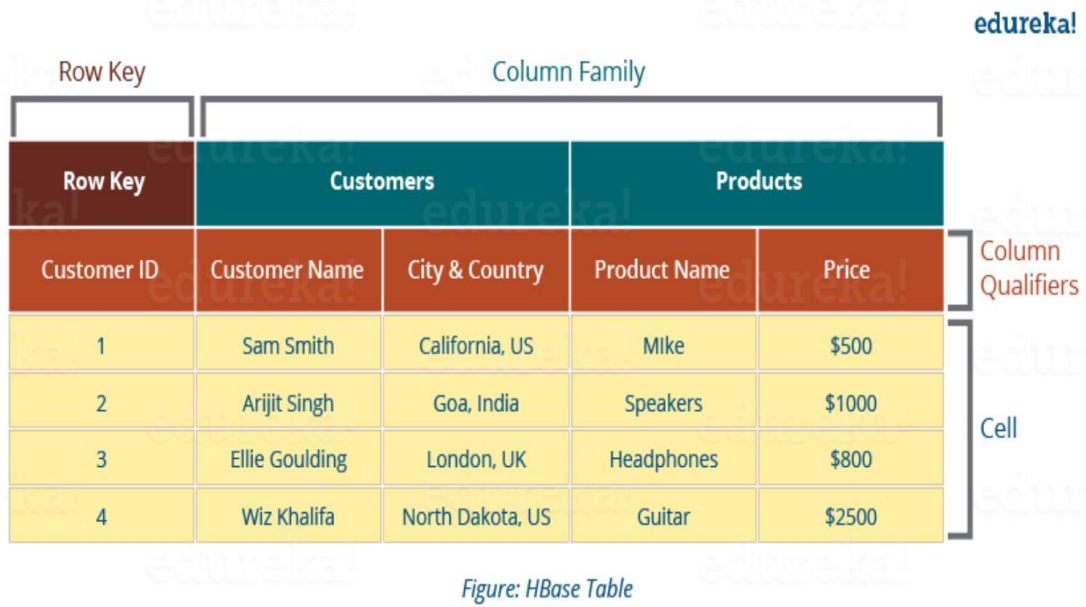


Figure: HBase Table

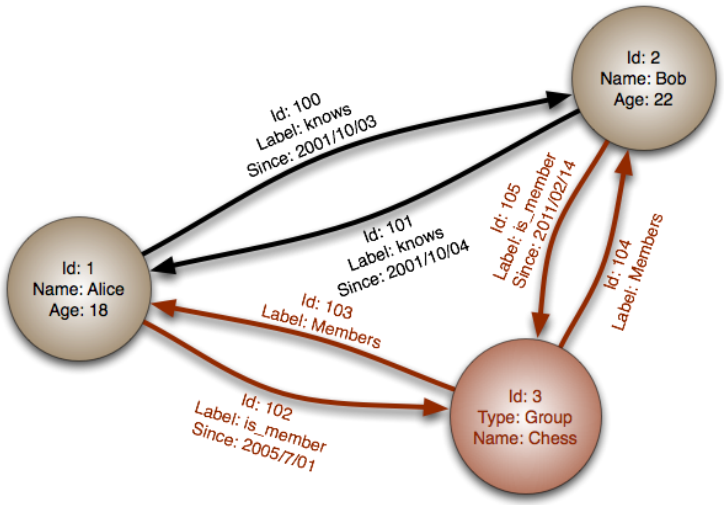
기본 단위는 **Column**,
Column들이 모여서 **Column Family**를 구성,
Column Family가 모여서 **Table**을 구성

Table에 들어가는 각 row는 **rowkey**로 식별

대표적인 예시 HBase

출처 블로그

NoSQL – Graph



대표적인 예시 Neo4j

- 점(node): 추적 대상이 되는 사람, 기업, 계정 등 의 실체를 대표, RDB에서의 row와 유사
- 선(edge): 그래프(graph)나 관계(relationship)이라고도 하며 노드를 다른 노드에 연결하는 선이며 관계를 표현, RDB에서의 join연산과 유사
- 속성(property): 노드의 정보와 밀접한 관련, RDB에서의 column과 유사

사용 사례

이상 탐지
관계를 사용하여 거의 실시간으로 금융이나 구매 트랜잭션을 처리할 수 있다. 빠른 그래프 쿼리를 사용하면 구매하려는 사람이 이미 알려진 이상 사례에 포함된 것과 같은 이메일 주소와 신용 카드를 사용하는지 여부를 감지할 수 있다.
또한, 그래프 데이터베이스는 하나의 개인 이메일 주소와 연결된 여러 사람들 또는 IP 주소는 같지만 실제로는 서로 다른 주소에 거주하는 사람들 간의 관계 패턴을 파악하는 데 도움이 될 수 있다.

추천 엔진
그래프 데이터베이스를 사용하면 고객 관심 분야, 친구, 구매 이력과 같은 정보 카테고리들 사이의 그래프 관계를 저장할 수 있다. 고도의 가용성을 가진 그래프 데이터베이스를 사용하여 같은 운동을 즐기고 비슷한 구매 이력을 가진 다른 사용자가 어떤 제품을 구입했는지에 따라 사용자에게 제품을 추천할 수 있다. 또는 공통의 친구가 있지만 서로는 아직 모르는 사람들을 찾아내어 친구를 추천할 수도 있다.

출처 aws, 위키

NoSQL – Document

Relational

ID	first_name	last_name	cell	city	year_of_birth	location_x	location_y
1	'Mary'	'Jones'	'516-555-2048'	'Long Island'	1986	'-73.9876'	'40.7574'

ID	user_id	profession
10	1	'Developer'
11	1	'Engineer'

ID	user_id	name	version
20	1	'MyApp'	1.0.4
21	1	'DocFinder'	2.5.7

ID	user_id	make	year
30	1	'Bentley'	1973
31	1	'Rolls Royce'	1965

대표적인 예시 MongoDB

MongoDB

```
{
first_name: "Mary",
last_name: "Jones",
cell: "516-555-2048",
city: "Long Island",
year_of_birth: 1986,
location: {
        type: "Point",
        coordinates: [-73.9876, 40.7574]
},
profession: ["Developer", "Engineer"],
apps: [
 { name: "MyApp",
   version: 1.0.4 },
 { name: "DocFinder",
   version: 2.5.7 }
],
cars: [
   { make: "Bentley",
     year: 1973 },
   { make: "Rolls Royce",
     year: 1965 }
 ]
}
```

문서 데이터베이스는 JSON 유사 형식의 문서로 데이터를 저장 및 쿼리하도록 설계된 비관계형 데이터베이스 유형

애플리케이션 코드에서 사용하는 것과 동일한 문서 모델 형식을 사용하여 데이터베이스에서 보다 손쉽게 데이터를 저장하고 쿼리할 수 있다.

사용 사례

콘텐츠 관리

블로그 및 비디오 플랫폼과 같은 콘텐츠 관리에 용이하다. 애플리케이션이 추적하는 각 엔터티를 **단일 문서**로 저장할 수 있다. 요구사항이 발전함에 따라 문서 데이터베이스는 개발자가 애플리케이션을 업데이트하는데 더욱 직관적으로 사용할 수 있게 되었다. 게다가, 데이터 모델을 변경해야 하는 경우에도 영향을 받는 문서만 업데이트하면 된다. 스키마를 업데이트하거나 데이터베이스 가동 중단이 필요하지 않다.

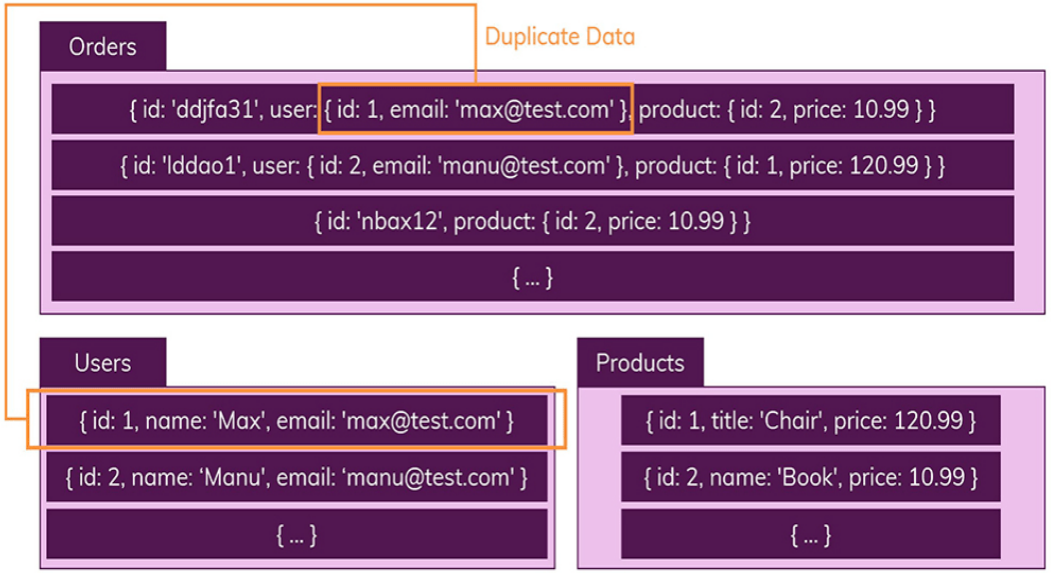
카탈로그

전자 상거래 애플리케이션에서 다른 제품의 개별 속성의 수는 서로 다르다. RDB에서는 수많은 속성을 관리하는 것이 비효율적이며 읽기 성능에도 영향을 주는가 반면, document DB를 사용하면 각 제품의 **속성을 단일 문서**로 기술하여 관리가 쉽고 읽기 속도도 빨라집니다. 한 제품의 속성을 변경해도 다른 제품에는 영향을 주지 않는다.

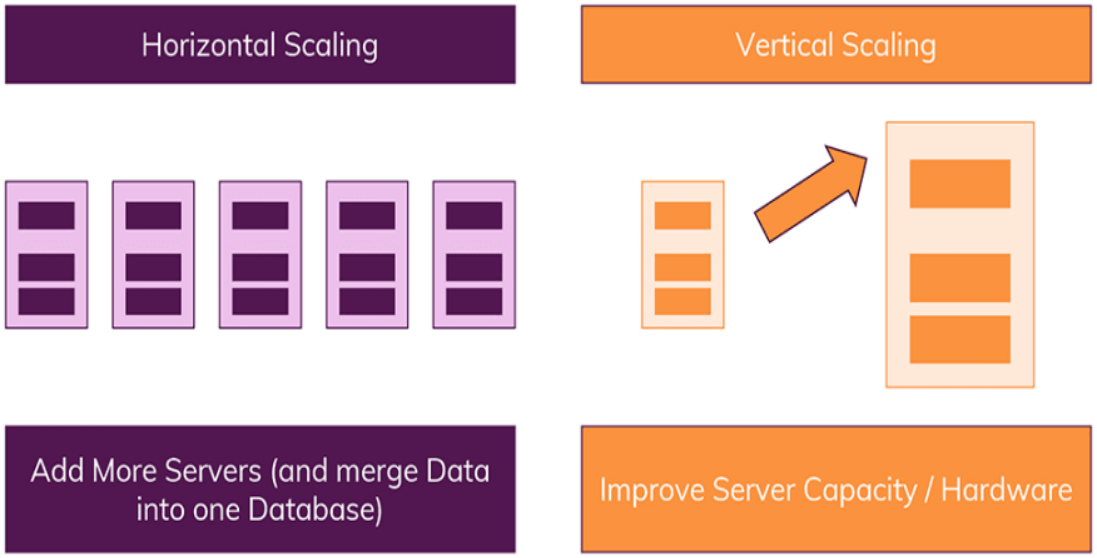
출처 mongodb, aws

NoSQL

관련 데이터, 동일 컬렉션



수평 확장(Horizontal Scaling)



출처 academind, 번역

RDMBS(SQL) vs NoSQL?

	SQL	NoSQL
Database Type	Relational Databases	Non-relational Databases / Distributed Databases
Structure	Table-based	• Key-value pairs • Document-based • Graph databases • Wide-column stores
Scalability	Designed for scaling up vertically by upgrading one expensive custom-built hardware	Designed for scaling out horizontally by using shards to distribute load across multiple commodity (inexpensive) hardware
Strength	• Great for highly structured data and don't anticipate changes to the database structure • Working with complex queries and reports	• Pairs well with fast paced, agile development teams • Data consistency and integrity is not top priority • Expecting high transaction load

출처 blog, academind, 번역

감사합니다! QnA