NoSQL

https://shuu.tistory.com/135

NoSQL

: Not Only SQL의 약자로 비관계형 데이터베이스를 의미

NoSQL 특징

- RDBMS와 달리 데이터 간의 관계를 포함하지 않음
  - Key - Value 형태로 저장되고 Foreign Key가 없기 때문에 Join 연산을 수행할 수 없다.
- RDBMS에 비해 대용량의 데이터 저장 가능
- 분산형 구조로 설계되어 있음
  - 여러 곳의 서버에 데이터를 분산 저장하여 특정 서버에 장애가 발생해도 데이터 유실 또는 서비스 중지가 발생하지 않는다.
- 고정되어 있지 않은 데이터 스키마를 가짐
  - RDBMS와 달리 스키마가 유동적이어서 데이터를 저장하는 칼럼이 다른 데이터 타입을 가지는 것이 허용된다.

Key	Value
사과	빨간색
수박	[초록색, 검은색]
딸기	{ 산딸기: 빨간색 }

NoSQL 장점

- RDBMS에 비해 저렴한 비용으로 분산 처리와 병렬 처리가 가능하다.
- 비정형 데이터 구조 설계로 설계 비용이 감소한다.

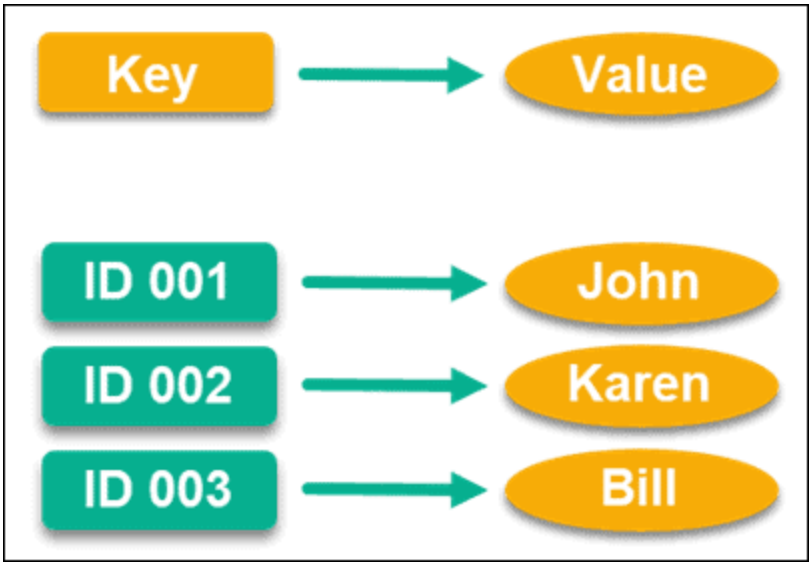
- Big Data 처리에 효과적이다
- 가변적인 구조로 데이터 저장이 가능하다
- 데이터 모델의 유연한 변화가 가능

NoSQL 단점

- 데이터 업데이트 중 장애가 발생하면 데이터 손실 발생 가능
- 많은 인덱스를 사용하려면 충분한 메모리가 필요, 인덱스 구조가 메모리에 저장됨
- 데이터 일관성이 항상 보장되지는 않음
    - 최종적 일관성을 지향한다

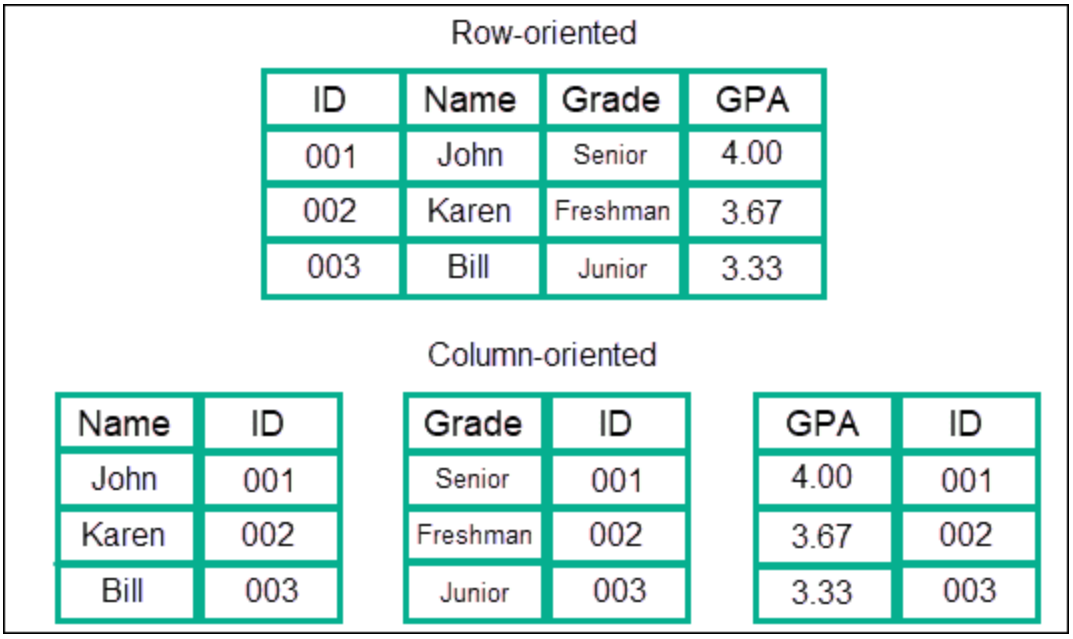
NoSQL 종류 4가지

- Key - Valule Database
    - 데이터가 Key와 Value로만 이루어져 있는 데이터베이스
    - 정렬이나 조인 연산이 불가능함
    - Key를 통한 고속 읽기, 쓰기가 가능함

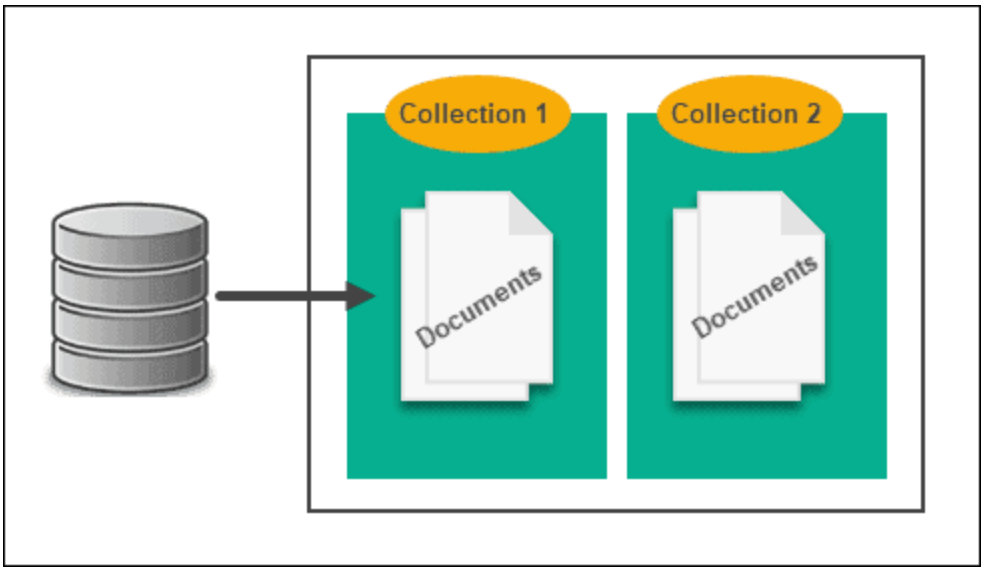


- Big - Table (Wide-Column) Database

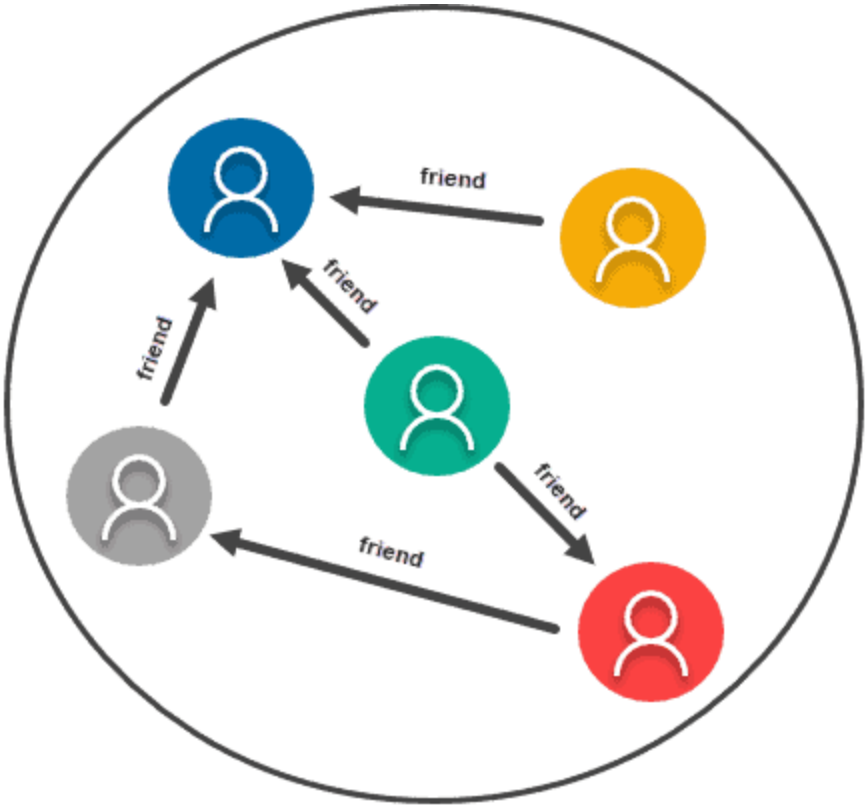
- Key - Value의 형태로 이루어져 있는 데이터베이스
- 내부적으로 Key를 정렬하여 데이터 제공



- Document Database
  - value에 Document라는 타입을 저장 (XML, JSON, YAML)
  - Document id 또는 속성값 기준으로 인덱스 생성
  - Sorting, Join, Grouping 등이 가능
  - B트리 인덱스를 사용하여 2차 인덱스 생성
    - 크기가 커질수록 insert delete 성능이 떨어진다
    - 변하지 않는 정보를 저장하고 조회하는 데 적합



- Graph Database
  - node들과 Relationship 들로 구성된 관계
  - key - value 방식이며 모든 노드는 끊기지 않고 연결되어 있음
  - relationship은 direction, type, start node, end node에 대한 속성등을 가짐



CAP 이론

분산형 구조의 3가지 특징

- 일관성(Consistency)
  - 분산된 노드 중 어느 노드로 접근하더라도 데이터 값이 같아야 한다.
- 가용성(Availability)
  - 클러스터링된 노드 중 하나 이상의 노득 실패라도 정상적으로 요청을 처리할 수 있는 기능을 제공한다
- 분산 허용(Partitioning Tolerance)
  - 클러스터링 노드간에 통신하는 네트워크 장애가 나더라도 정상적으로 서비스를 수행한다.
  - 노드 간 물리적으로 전혀 다른 네트워크 공간에 위치도 가능하다.

CAP 이론은 이 중 2가지만 만족할 수 있다는 이론이며 NoSQL은 대부분 _CAP 이론을 따름

(RDBMS는 주로 일관성, 가용성 만족)

NoSQL 데이터 모델링 패턴

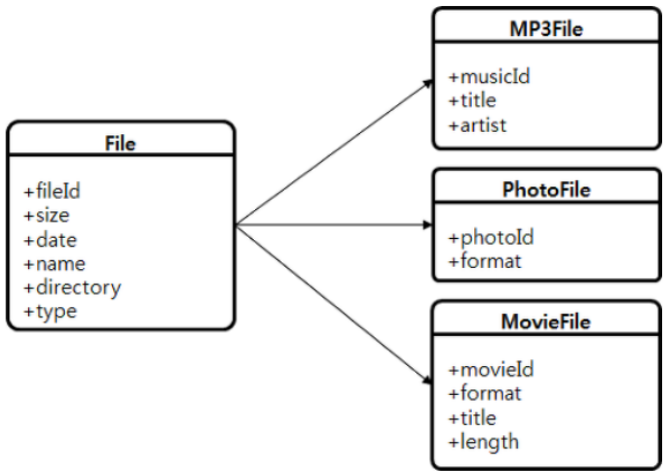
Denormalization (비정규화)

- 특징
  - 같은 데이터를 중복해서 저장하는 방식
  - 비정규화 진행 시 두 테이블간의 Join을 없앨 수 있다.
  - Join을 하지 않게 되는 경우 IO 한번으로 모든 데이터 가져올 수 있음
- 장점
  - 쿼리 당 IO 횟수 감소
  - 쿼리 데이터 사이즈 감소
  - 쿼리 수행 복잡도 감소
- 단점

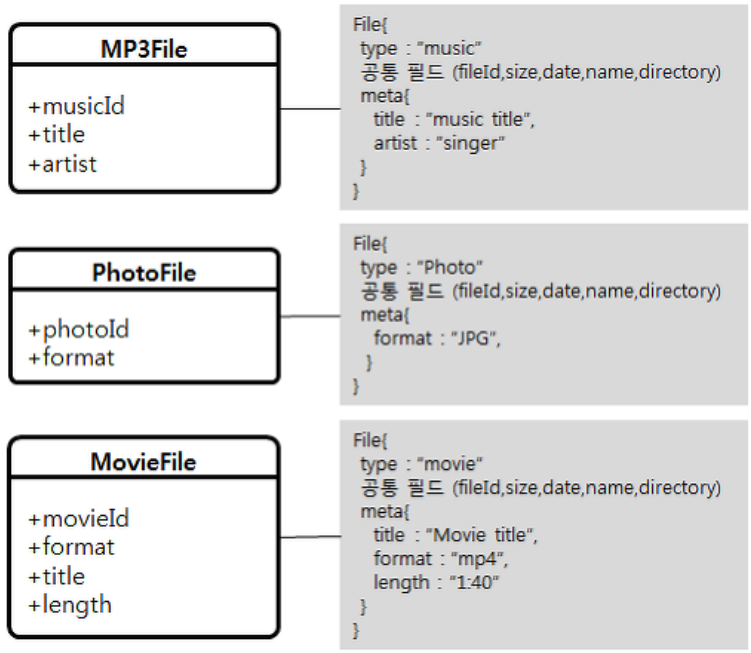
- 전체 데이터 사이즈 증가
- 데이터 일관성 문제 발생 가능

Aggregation

- 특징
  - NoSQL 특성 중 하나인 Scheme-less / Soft Scheme 특성을 이용하여 1:N 같은 복잡한 entity들의 관계를 손쉽게 하나의 테이블로 바꿀 수 있음



RDBMS



NoSQL

Application Side Join

- 특징
  - 어쩔 수 없이 Join 기능을 구현해야 하는 경우,
    NoSQL을 사용하는 client application 단에서 Join 로직을 처리해야함
  - Join이 필요한 테이블의 수 만큼 NoSQL로의 Request/Response IO가 발생하긴 하지만 Denomalization 등에 비해서 스토리지 사용량은 감소

MongoDB

특징

- 신뢰성 (Reliability)
  - 서버 장애에도 서버가 유동적으로 분담하여 서비스는 계속 동작 유지
  - 몽고디비는 기본적으로 1개의 primary와 2개의 secondary로 ReplicaSet을 구성
  - primary : 데이터 쓰기 요청
  - secondary : primary에서 변경된 데이터 복제

    - primary 서버에 문제가 생기면 secondary가 primary로 전환되어 서버를 유지시킨다.
    - 이후 빈 secondary 서버를 몽고디비가 복구 시켜줌으로 서버가 유지할 수 있게 된다.
- 유연성 (Flexibility)
    - 여러가지 형태의 데이터를 손쉽게 저장
    - 다양한 종류의 데이터가 추가되어도 스키마 변경 과정 없이 필요한 데이터 바로 저장 가능
- 확장성 (Scalability)
    - 데이터와 트래픽이 증가하여 Replica Set에 담을 수 없게 되면 몽고 DB는 데이터를 샤딩하여 분산시켜줄 수 있음 (밸런싱)
- Index 지원 (Index Support)
    - 다양한 조건으로 빠른 데이터 검색
    - 대부분 NoSQL들과 다른 몽고디비 만의 큰 차이점
    - 대용량 데이터에서 다양한 조건의 조회 가능
    - 다양한 형태의 Index 제공
        - Hashed Index
        - Multikey Index
        - Partial Index
        - TTL Index (제한시간 설정하여 오래된 데이터를 자동으로 지워주는 인덱스)
        - Geospatial Index (공간내에 거리나 범위 계산)
- BSON 형식
    - BSON : Binary JSON으로 JSON과 동일한 구조이지만 Binary 형태로 변경된 구조
    - JSON의 경우 텍스트 기반으로 구문 분석이 느리고 공간 효율성과 거리가 멀기 때문에 기계가 빠르게 읽을 수 있는 binary 형태로 변경하여 저장

MongoDB 데이터 모델링

- Database : Collection의 물리적 컨테이너

- Collection : RDBMS table 과 비슷함
- Document : RDBMS Tuple과 유사
- Key/Field : RDBMS Column