KB금융그룹 | 국민의 평생 금융파트너

또 다른 DBMS

no-sql

2025년 상반기 K-디지털 트레이닝

최신 웹을 위한 도큐먼트 데이터베이스

[KB] IT's Your Life

nosql에서 대표적인 몽고디비

커뮤니티 버전과 상업버전이 있다.

빅데이터가 나오면서
주목받은.

꼭 nosql라 해서 꼭 빅데이터와
연관되어있진 않다.

멀티미디어에 대한
데이터는비정형 데이터.

RDBMS는 정형 데이터 관리 특화.

웹서비스는 비정형 데이터가 많이 등장함.
OODB패러다임으로 관리하다가
OR-DBMS패러다임으로 관리하다가
NOSQL패러다임으로 묶어버림.

KB국민은행

nosql기반 대표 디비 == 몽고 디비

MongoDB 소개

KB 국민은행

MongoDB의 특징

도큐먼트 기반.

- **웹 애플리케이션과 인터넷 기반을 위해 설계된 데이터베이스 관리 시스템**

번외 레디스 디비
키 – 밸류 기반.

- **데이터 모델과 지속성 전략**
  - 높은 읽기/쓰기 효율
  - 자동 장애 조치를 통한 확장의 용이성
- **직관적인 데이터 모델**
  - 정보를 행(row) 대신 도큐먼트(document)에 저장

이때 말하는 JSON이라고 생각하면 돼
json자체를 하나의 행으로 간주.

```
{
    _id: 10,
    username: 'peter',
    email: 'pbbakkum@gmail.com'
}
```

속성(필드)(칼럼)

JSON형태로 데이터를 표현

특징 형태가 고정되어있지 않다. 비정형이다.` 어떤 행은 특정 속성이 있고 어떤 행은 없고.`

  - 여러 개의 이메일을 저장하는 경우
    - 관계 모델에서는 조인을 위한 이메일 주소와 사용자 테이블을 각각 만들어야 함

```
{
    _id: 10,
    username: 'peter',
    email: [ 'pbbakkum@gmail.com', 'pbb7c@virginia.edu' ]
}
```

또한 속성의 데이터 타입이 아주 다양함
배열도 된다!

스키마==구조

MongoDB의 도큐먼트 형식

- 임의의 구조를 저장하는 스키마 == 비정형 데이터다
- JSON에 기반 사람이 표현할때는 텍스트기바의 JSON으로 표현. 실제로는 BSON으로 내부로 처리
- 키(key)와 키에 대한 값(value)로 구성, 중첩에 제한이 없음

js 객체 형태처럼

몽고디비에서는
이런 중첩된
도큐먼트를
임베디드
문서라고 부른다

- 관계 데이터베이스에서 필요한 여러 테이블 간의 복잡한 조인 연산이 없음
  - 관계 데이터베이스
    - 완전히 정규화된 데이터 모델에서 한 상품의 정보는 여러 개의 테이블에 나뉘어 저장
    - 조인 연산으로 가득 찬 복잡한 SQL 쿼리를 사용
  - 도큐먼트 모델
    - 대부분의 상품 정보를 하나의 도큐먼트로 표현

RDBMS에서는 VO객체를 생각해 보면
jdbc travel실습 기억나죠잉?

1대n 관계를 가진 테이블 표현시

한 테이블들의 인스턴스들을 담는
리스트형식이 필드로 자리잡고 있는 형태이다.
그대로 저장X.

또한 해당 테이블도 또다른 VO객체로
매핑해야함

- 객체지향 언어의 객체에 잘 매핑되는 데이터 저장 구조
  - 프로그래밍 언어에서 정의한 객체가 그대로 저장
  - 객체 매퍼의 복잡성이 사라짐

객체지향 프로그래밍과
궁합이 잘맞다

고대로 저장하면 되는 것이
몽고디비의 장점!. RDBMS는
테이블로 분리 매핑 해야하는 과정 필요.
조인결과에 맞게금 바꿔야하기도

MongoDB 소개

MongoDB의 역사

- 2007년 중반 10gen이라는 회사에서 웹 애플리케이션을 위한 데이터베이스로서 시작
- 이 후 MongoDB로 회사명 바뀜
- 오픈 소스 정책

- 매우 간단하지만 유연하고 웹 애플리케이션 계층의 일부로 개발 되었음

MongoDB의 핵심 기능

몽고디비 하나의 도큐먼트==RDBMS의 하나의 행.

도큐먼트 데이터 모델

소셜 뉴스 사이트의 기사를 나타내는 JSON 도큐먼트

```
{
  _id: ObjectID('4bd9e8e17cefd64410896bb'),   // _id 필드가 프라이머리 키
  title: 'Adventures in Databases',
  url: 'http://example.com/databases.txt',
  author: 'msmith',
  vote_count: 20,
  tags: ['database', 'mongodb', 'indexing'],  // 태그는 문자열의 배열로 저장
  image: {  // 임베디드 문서(또 다른 문서)
    url: 'http://example.com/db.jpg',
    caption: 'A database.',
    type: 'jpg',
    size: 75381,
    data: 'Binary'
  },
  comments: [    // 코멘트는 코맨트 객체(도큐먼트)의 배열로 저장
    { user: 'bjones', text: 'Interesting article' },
    { user: 'sverch', text: 'Color me skeptical' },
  ]
}
```

몽고디비의 기본키. 정해져있음. UUID
universal unique id 전세계적으로 중복 없는

내부 검색 가능

내부 객체
내부 배열
등등

내부 배열의 내부객체 등등

다 접근 검색 기능 등등

다 가능

RDBMS에서는 n:m 관계이므로
적어도 테이블이 3개 등장해야죠??
3개 만들고 찾아낼때도
조인 연산이 필요하고 하...

속성 필드로 이미지 객체를 가지고 있네용

RDBMS에서는 1:N관계겠다

코멘트 속성은
객체들의 배열로 구성됨

이게 하나의 도큐먼트고 RDBMS에서는 이에 대응하는게 하나의 행에 해당됨

MongoDB의 핵심 기능

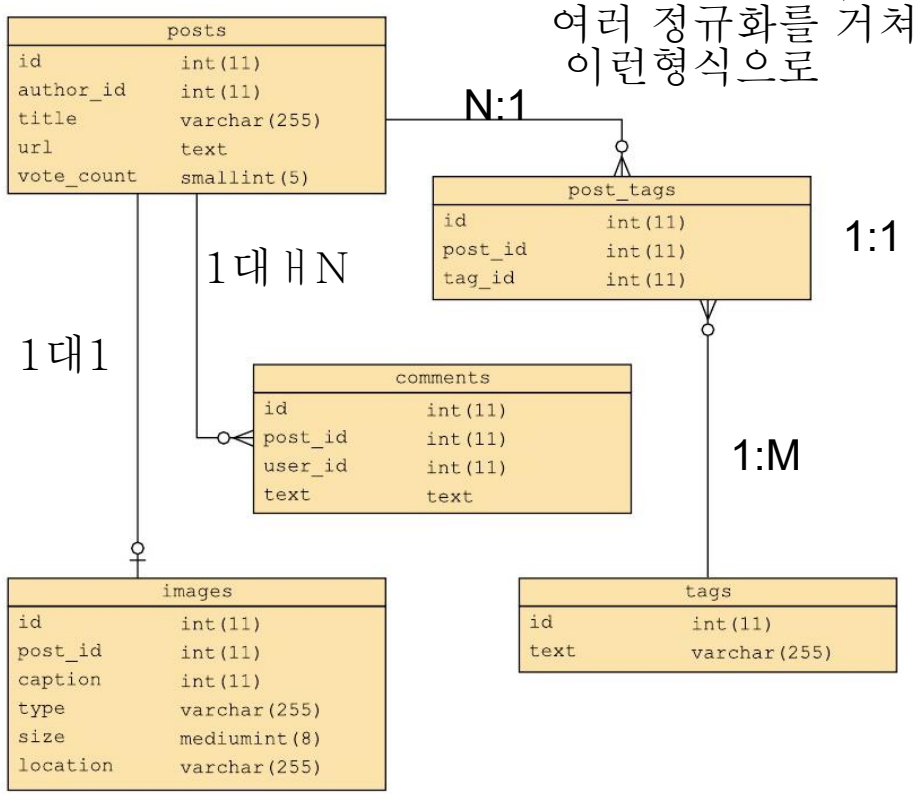
도큐먼트 모델

- 도큐먼트
  - 속성의 이름과 값으로 이루어진 쌍의 집합
- 하나의 도큐먼트로 다양한 구조의 데이터를 표현
  - 고정된 스키마가 없음
- MongoDB는 내부적으로 Binary JSON 혹은 BSON의 형태로 도큐먼트를 저장
- 컬렉션(collection)에 도큐먼트를 저장
  - 관계 데이터베이스에서 테이블에 해당

도큐먼트는 행에 해당

컬렉션은 도큐먼트들을 모아서 관리
==
테이블에 해당

RDBMS로 설계한다면
여러 정규화를 거쳐서
이런형식으로



MongoDB의 핵심 기능

몽고디비의 또다른 장점.

스키마가 없는 모델의 장점

- 데이터베이스가 아닌 애플리케이션이 데이터 구조를 결정
- 데이터 구조가 빈번히 변경되는 개발 초기 단계에서 개발 속도를 단축시켜 줌
- 가변적인 속성을 갖는 데이터를 표현할 수 있음
  - 예) 전자 상거래 상품 카탈로그
  - 추후에 필요한 데이터 필드가 무엇인지에 대해서 걱정할 필요없음

개발 과정에서
좋다. 굉장히.

트예)
IOT 센서 데이터를 수집하고 있음

sensors컬렉션을 만들어서 관리중

한 도큐먼트 안에는
온도 센서
터치센서
켜짐센서

다양한 타입을 가짐.
실수, 정수, 불린, 객체, 배열로
표현해야함

관계형 디비는 센서별로 테이블을
만들어서 관리해야함

하지만 IOT센서 종류는
유동적으로
사라지고
추가됨.
그럴때마다 해야한다? 오바.

몽고디비는
구조가 없기에 그냥 추가하면 돼

MongoDB의 핵심 기능

쿼리 예

- 추천수가 10이상의 politics라는 용어로 태그된 모든 포스트 찾기
- SQL 쿼리

```
SELECT * FROM posts
  INNER JOIN posts_tags ON posts.id = posts_tags.post_id
  INNER JOIN tags ON posts_tags.tag_id = tags.id
  WHERE tags.text = 'politics' AND posts.vote_count > 10;
```

N:M관계

- MongoDB

```
db.posts.find({ tag: 'politics', vote_count: { $gt: 10 }});
```

→ 태그가 각 도큐먼트에 포함되어 있다고 가정

데이터베이스.컬렉션(테이블).찾아라명령(JSON문서객체 == where 절 == 조건 도큐먼트);

조건 JSON 도큐먼트 { 검색 조건들 }

검색 조건들
키 값 쌍. 대상 속성명:조건속성값. 대상속성명:{비교조건} 대상속성이 해당 값을 가정햐한다

비교조건 : $gt(특정변수):10//10보다 커야한다

MongoDB의 핵심 기능

- **인덱스**
  - B-Tree 기반
  - _id 프라이머리 키에 대해서는 자동으로 인덱스 생성
  - 세컨더리 인덱스 생성 가능
  - 컬렉션 별로 64개까지 인덱스 생성 가능

필ㄹㄷ별로.

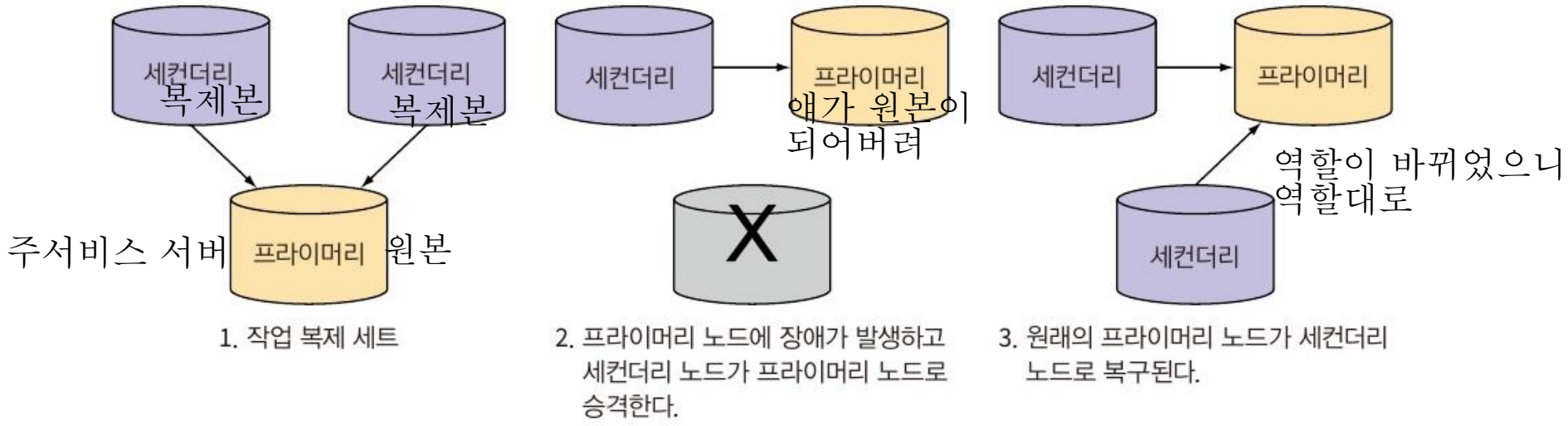
DBMS 분산 분산 시스템. distributed.
물리적으로는 분산된 애들이 연결되어 하나의 가상 컴퓨터처럼 사용가능한

MongoDB의 핵심 기능

클러스터 : 여러대의 DBMS서버를 한대의 DBMS서버인것처럼 제공하는 것. 2가지 기법이 있다,

1. **복제(replica)** 똑같은 애를 복제(미러링)하는 것.

- 복제 세트(replica set)
  - 서버와 네트워크 장애가 발생할 경우를 대비해 중복성과 장애 조치 자동화를 위해 데이터를 여러 대의 서버에 분산
  - 읽기에 대한 확장 제공
- 하나의 프라이머리 노드와 여러 대의 세컨더리 노드로 구성
- 프라이머리 노드 : 읽기/쓰기 모두 가능 수정될 경우. 세컨더리도 같이 진도 맞춰야함
- 세컨더리 노드 : 읽기만 가능
- 복제 세트의 자동 장애 복구



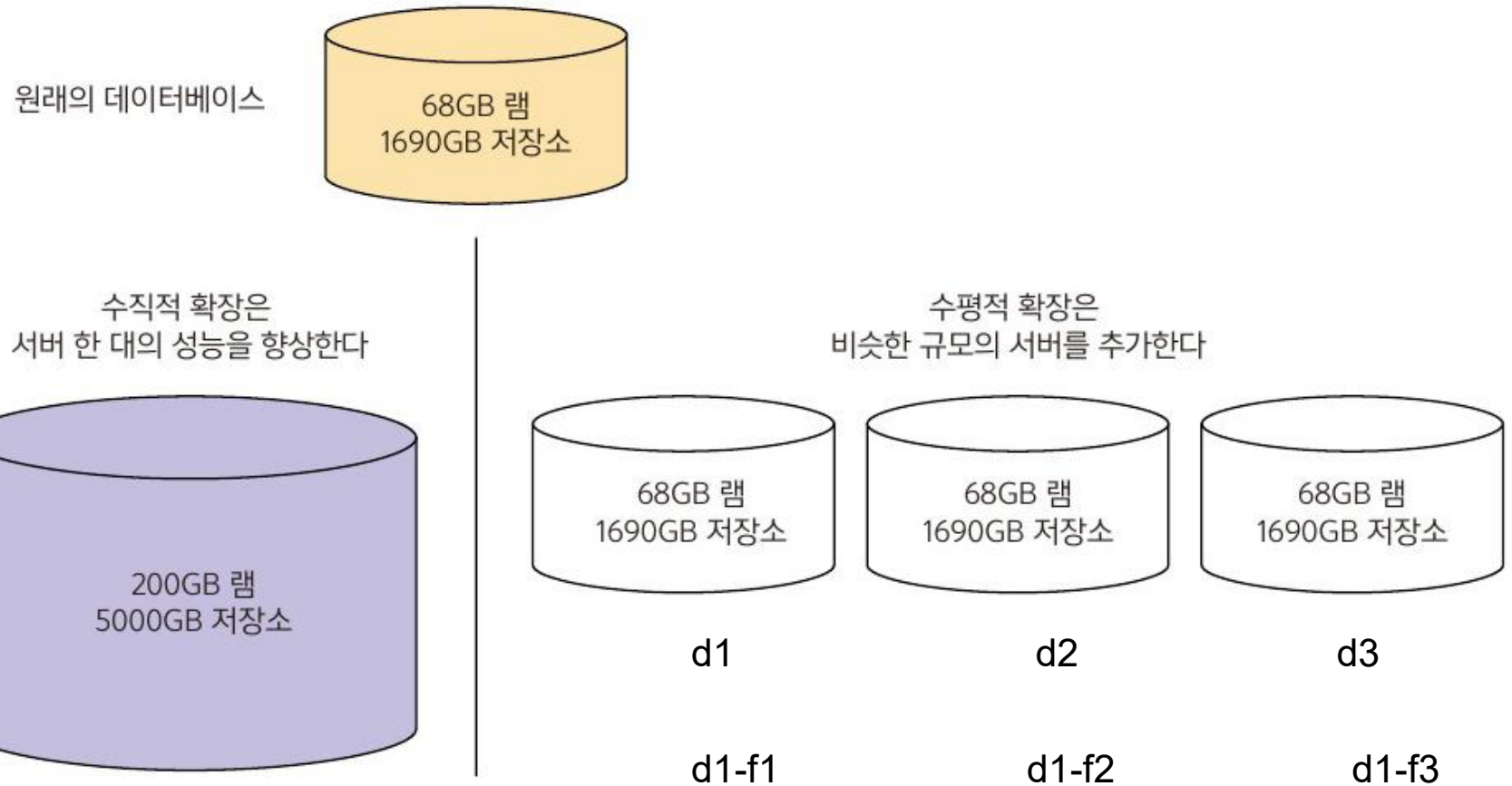
메모리 차지율이 높긴해

MongoDB의 핵심 기능

샤딩(shading)

데이터를 여러 서버에 분산해서 저장하자.
복제랑 달라요잉

수직적 확장 vs 수평적 확장



oracle mysql를 사용하면서 분산 기능을 사용하려면
돈이 아주 만힝 드는데 몽고디비는 무료다1

MongoDB의 핵심 기능

샤딩(shading)

- 범위 기반 파티션 메커니즘을 통해 여러 노드에 걸쳐 분산하는 것을 자동으로 관리

제품	가격
머플러	20,000
핸드크림	13,000
케이크	35,000
한우세트	100,000
향수	60,000
무드등	40,000

한예시

조건 별로 분산 따로 저장

0 ~ 30,000

제품	가격
핸드크림	13,000
머플러	20,000

30,000 ~ 60,000

제품	가격
케이크	35,000
무드등	40,000

60,000 ~ 100,000

제품	가격
향수	60,000
한우세트	100,000

건색 범위가
확주는
장점`

하지미ㅏㄴ 전ㄴ체
탐색때는
추가적인 활동필요
리듀스앤맵