



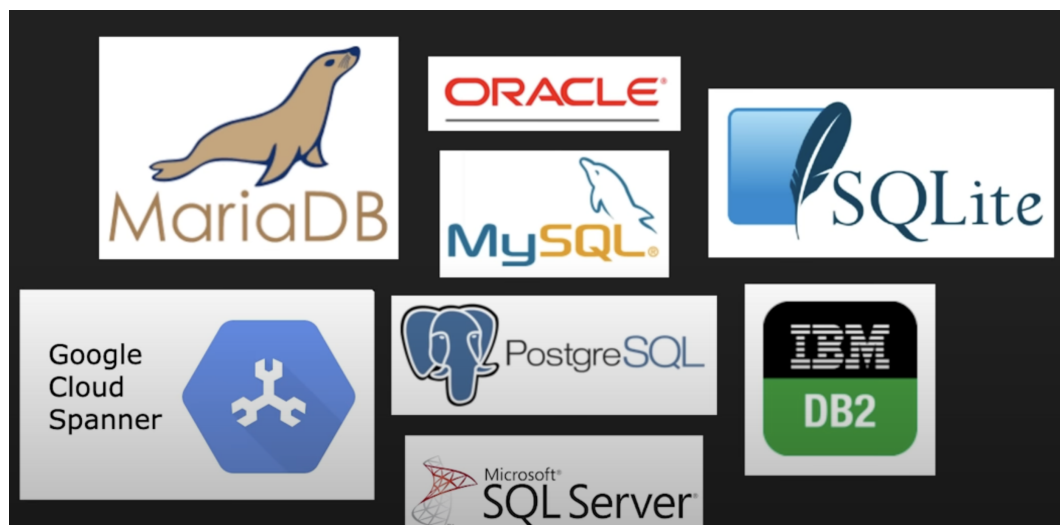
면접을 위한 CS 전공지식 노트

▼ CHAPTER4. 데이터베이스

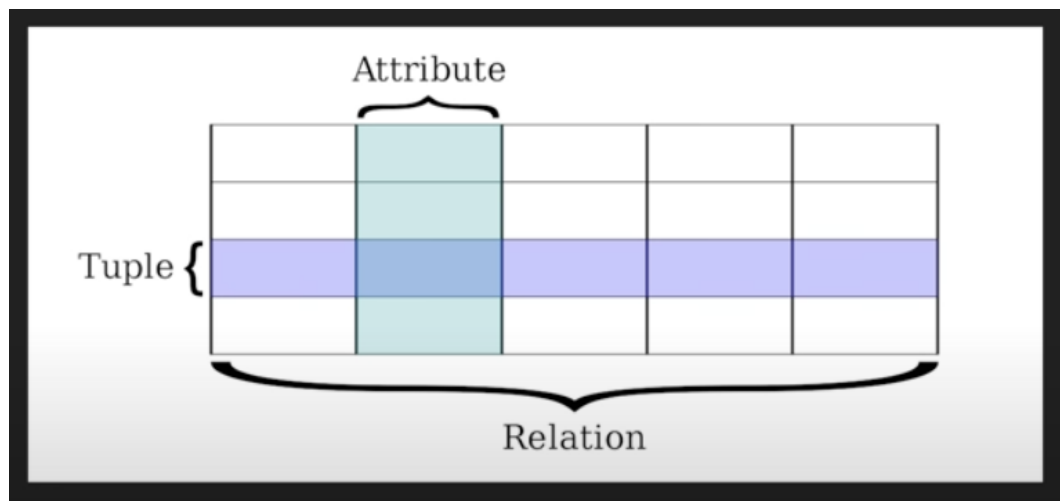
▼ 4.4 데이터베이스의 종류

관계형 데이터베이스 (RDB, Relational Database)

- 관계형 모델을 기반으로 하는 데이터베이스



RDB의 종류



데이터의 관계를 나타내기보다 행렬의 이름에서 따옴

특징

- 고정된 행(row)과 열(column)로 구성된 테이블에 데이터 저장
 - e.g. '옷가게' 테이블

상품명	가격	색상
셔츠	6000	Red
바지	7000	Blue
양말	8000	Green
뽀죤	9000	Yellow

테이블을 만들고 행에 데이터 저장(엑셀)

- RDBMS (이를 유지, 관리하기 위한 시스템)
- SQL(Structured Query Language)를 사용해 데이터를 조회, 관리(DML, DDL, DCL 등)

```
CREATE TABLE 옷가게 ( # 테이블의 열(Column) 정의
    상품명 VARCHAR(50), # 데이터의 타입, 크기 지정
    가격 INT,
    색상 VARCHAR(50)
);
```

```
INSERT INTO 옷가게 VALUES ('셔츠', 6000, 'Red'); # 데이터
INSERT INTO 옷가게 VALUES ('바지', 7000, 'Blue');
INSERT INTO 옷가게 VALUES ('양말', 8000, 'Green');
INSERT INTO 옷가게 VALUES ('뽀죤', 9000, 'Yellow');
```

- 데이터를 정규화해서 저장한다.

프로그램	가격	강사	출신대학	중복발견
스쿼시	5000	김을용	서울대	
헬스	6000	박덕팔	연세대	
골프	8000	이상구	고려대	
골프중급	9000	이상구	고려대	
개인피티	6000	박덕팔	연세대	

프로그램	가격	강사	강사	출신대학
스쿼시	5000	김을용	김을용	서울대
헬스	6000	박덕팔	박덕팔	연세대
골프	8000	이상구	이상구	고려대
골프중급	9000	이상구		
개인피티	6000	박덕팔		

(다른 테이블로 빼서 중복제거)

단점 : 데이터 출력 문법이 복잡해진다.

장점

- ACID 성질을 갖는다.
 - **ACID**(원자성, 일관성, 고립성, 지속성)는 데이터베이스 트랜잭션이 안전하게 수행된다는 것을 보장하기 위한 성질을 가리키는 약어
 - 트랜잭션
 - 데이터베이스에서 수행되는 작업의 단위
 - 데이터베이스에서 정보를 읽거나 쓸 때 일어나는 하나의 논리적인 작업
 - 일련의 연산을 원자적(Atomic), 일관적(Consistent), 고립적(Isolated), 지속적(Durable)으로 수행하는 개념



예시)

현금을 뽑아 송금하는 작업.

1. 계좌에서 돈을 빼고

2. 받는 계좌에 돈을 송금

두 가지 단계 모두가 성공해야 전체 송금 트랜잭션이 완료

하나라도 실패하면 전체 트랜잭션이 취소되고 원래 상태로 돌아감

이를 통해 DB의 일관성을 유지, 무결성 보장

- 데이터의 일관성과 무결성을 유지할 수 있다.
 - 데이터 무결성 :

데이터가 전송, 저장되고 처리되는 모든 과정에서 변경되거나 손상되지 않고 완전성, 정확성, 일관성 유지를 보장하는 특성.
- 각 데이터를 중복 없이 한 번만 저장한다. (데이터 정규화)

단점

- 스키마 변경이 어렵다.
 - 스키마
 - DB에서 데이터의 구조, 제약 조건, 관계 등을 정의한 것
 - DB의 '계획도'로 생각할 수 있다.
 - 스키마 변경이 어려운 이유
 - 데이터 무결성

ACID 특성을 유지하기 위해 데이터 무결성을 보장한다.

스키마 변경 시 기존 데이터와의 일관성을 유지하기 위해 추가 작업이 필요하다.
 - 의존성 문제

다수의 테이블이 서로 의존하는 관계때문에 스키마 변경이 일어나면 의존성을 고려해야 한다.
- 대체적으로 수직적 확장만 가능하다.
 - 데이터가 여러 테이블에 의존해있기 때문
 - 수직적 확장(Vertical Scaling)

- 단일 서버에 더 강력한 하드웨어를 추가하거나 업그레이드하여 성능을 향상시키는 방법
 - 하드웨어 스펙의 확장을 의미
- 대용량 및 유연한 데이터에 대한 확장이 어렵다.

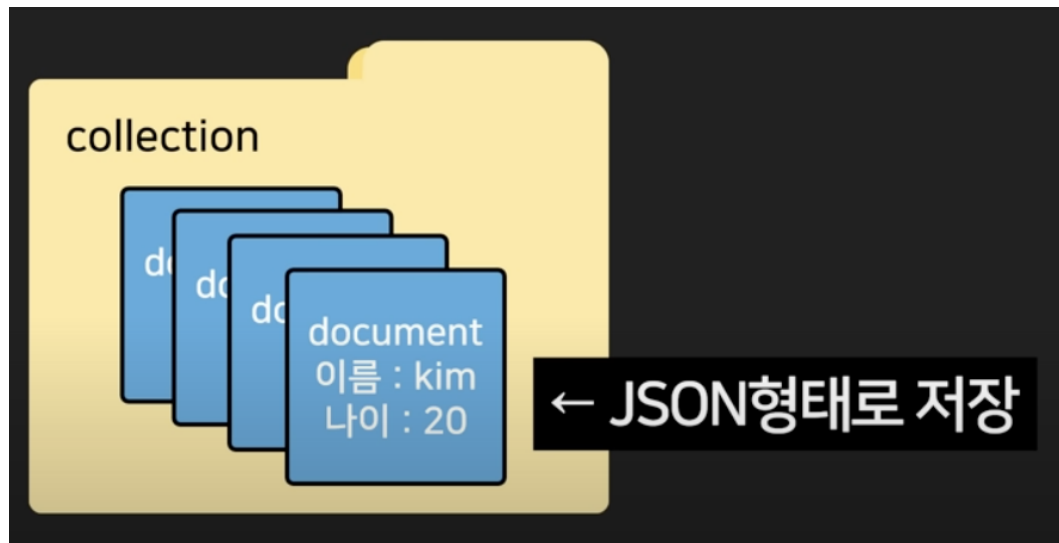
NoSQL Database

- 관계형이 아닌 데이터 모델의 총칭 (No SQL, Not Only SQL)
- Document 모델 / Key-Value 모델 / Graph 모델 등 다양한 모델이 있다.



1. Document Database (mongoDB)

- 레코드 하나를 오브젝트(도큐먼트) 형식으로 표현한다.
- 자유로운 스키마 구조를 가진다.
- collection이라는 폴더에 document라는 폴더를 만들고 그 안에 JSON 형식으로 데이터를 저장



```
1  [
2    {
3      "_id": {
4        "$oid": "5cdd3b56753ab5d4ac278710"
5      },
6      "humidity": 56,
7      "temperature_k": 288.6439936,
8      "conditions": "few clouds",
9      "wind_direction": 262,
10     "wind_speed": 6,
11     "city": "New York City",
12     "datetime": "2012-10-01T19:00:00Z",
13     "weather_station": {
14       "name": "Central Park",
15       "location": {
16         "type": "Point",
17         "coordinates": [
18           -73.9709045,
19           40.7789523
20         ]
21       }
22     }
23   },
24   {
25     "_id": {
26       "$oid": "5cdd3b56753ab5d4ac27871a"
27     },
28     "humidity": 54,
29     "temperature_k": 289.1988381,
30     "conditions": "few clouds",
31     "wind_direction": 265,
32     "wind_speed": 6,
33     "city": "New York City",
34     "datetime": "2012-10-02T02:00:00Z",
35     "weather_station": {
36       "name": "Central Park",
```

- 데이터의 중복을 제거하지 않는다.

프로그램	가격	강사	출신대학		프로그램	가격	강사	강사	출신대학
스쿼시	5000	김을용	서울대	→	스쿼시	5000	김을용	김을용	서울대
헬스	6000	박덕팔	연세대		헬스	6000	박덕팔	박덕팔	연세대
골프	8000	이상구	고려대		골프	8000	이상구	이상구	고려대
골프중급	9000	이상구	고려대		골프중급	9000	이상구		
개인피티	6000	박덕팔	연세대		개인피티	6000	박덕팔		

- RDB와 다르게 어떤 데이터를 저장할 지 지정하지 않아도 된다.
- 분산처리를 매우 잘 한다.
 - 입출력이 많은 SNS, 실시간 채팅, 게시판, 온라인 게임
 - 정확도, 일관성이 떨어진다.

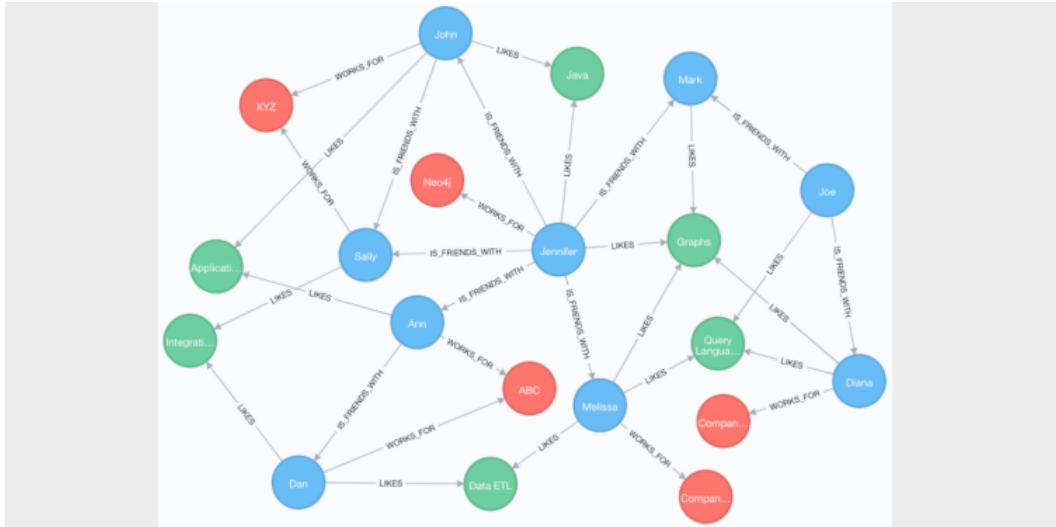
2. Key - Value Database (redis)

- 데이터를 Key - Value 형태로 저장
- 데이터에 빠르게 접근할 수 있다.
- 자주쓰는 데이터 캐싱, 영상 스트리밍, 로그인 기록 저장 등

key	value
이름	홍길동
나이	20
...	...

3. Graph Database (neo4j)

- 노드를 만들고 노드 안에 데이터를 저장
- 노드끼리의 관계 또한 저장
- Graph Query Language 사용
- 비행기 노선, SNS 친구 관계, 코로나 전염 맵, 추천 서비스 등



4. Column-family Database (Cassandra)

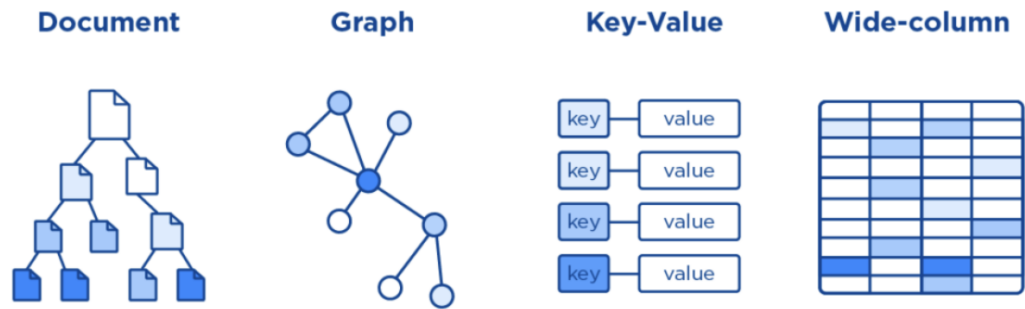
- RDB처럼 표 형식으로 데이터를 저장한다.(보다 유연하다.)
- 테이블, row를 만들고 데이터를 저장한다.



- Cassandra Query Language를 사용한다.
- 데이터 정규화(중복제거)없이 사용한다.
- 분산처리가 우수하다. (많은 입출력 처리 가능)

특징

- 다양한 방식으로 데이터를 표현한다.



- 테이블 사이에 명시된 제약이나 규칙이 없다.
- 스키마가 고정적이지 않고 매우 유연하다.
- 데이터의 수평적 확장이 가능하다.
 - 수평적 확장 (Horizontal Scaling)
 - 시스템의 성능을 향상시키기 위해 서버나 노드를 추가함으로써 데이터 베이스 시스템의 성능 확장하는 방법.
 - 양적 확장을 의미
- 연산이 빠르고 빅데이터, 실시간 연산에 적합하다.
- 확장성과 유연성이 필요한 대규모 및 다양한 데이터 요구 사항에 적합하다.

장점

- 유연한 데이터 모델과 스키마
- 빅데이터 및 분산 데이터에 대한 확장성

단점

- 일관성 보장이 RDB보다 어렵다.
- 특정 쿼리 언어가 없어 다양한 데이터 모델에 대한 표준이 없다.

RDB vs NoSQL

RDB	NoSQL
스키마가 정적이다.	유연한 스키마 구조를 갖는다.

RDB	NoSQL
수직 확장이 용이하다. (서버 용량을 늘리는게 쉽다.)	수평 확장이 용이하다. (서버를 여러대 늘리는게 쉽다.)
복잡한 쿼리와 Join 연산 가능	구조화된 쿼리 언어가 없는 경우도 많고 일반적으로 Join이 없다.
트랜잭션 처리에 용이하다.	분석 처리에 용이하다.
일관성과 안정성이 중요한 경우에 적합하다.	확장성과 유연성이 필요한 대규모 및 다양한 데이터 요구 사항에 적합하다.