

# 9. NoSQL

## 1. 비관계형DB (비정형화된 DB)

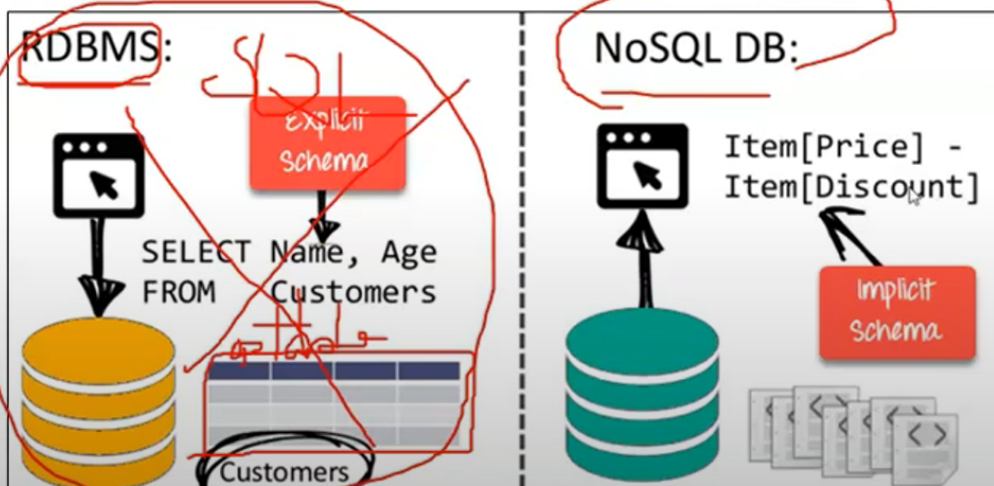
: SQL처럼 정형화되지 않은 SQL(스키마x) => No가 아니라 Not임 => Not Only SQL

cf) RDBMS - 정형화된 틀(스키마, 미리 정해진), 행과 열로 이루어짐

- RDBMS는 목적이 있음 (-> 정형화 해놓고 쓰는 이유)
  - ex) 은행, 성적

## 비관계형 데이터베이스: NoSQL, Not Only SQL

- Schema-Free: 보다 유연한 데이터 모델
- 분산다수의 하드웨어에 걸쳐 저장된 데이터
- 용이한 데이터 규모 확장성
- 대용량의 구조적, 반구조적 데이터들을 다룸(웹, 소셜 미디어, 그래픽)

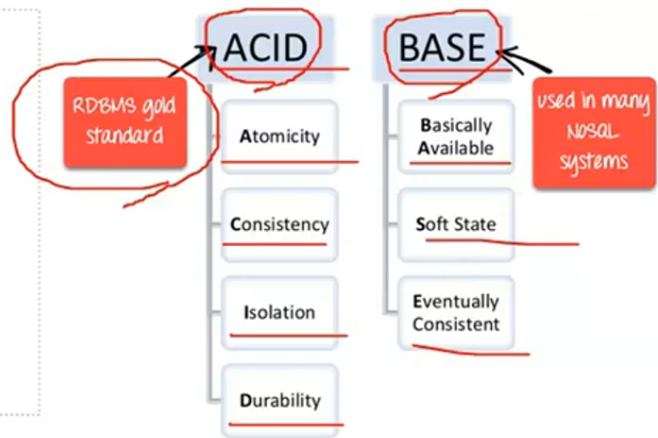


## ■ NoSQL

- 빅데이터를 처리하기 위한 분산 데이터 저장소의 통칭
- BASE

## ■ RDBMS

- ACID 중심



속성	BASE	ACID
적용분야	NOSQL	RDBMS
범위	시스템 전체에 대한 특성	트랜잭션에 한정
일관성측면	약한 일관성	강한 일관성
중점사항	Availability	'Commit'에 집중
시스템측면	성능에 초점	엄격한 데이터관리
효율성	쿼리디자인이 중요	테이블 디자인이 중요

- ACID를 지원한다~, BASE를 지원한다~
- 빅데이터는 BASE 성격이 강함

## 2. 모자이론 (CAP이론)

: 속성을 뜻하는 CAP 모두를 갖고 있을 수는 없다. 2가지를 갖고 있음

## 2002년 버클리대학의 Eric Brewer 발표

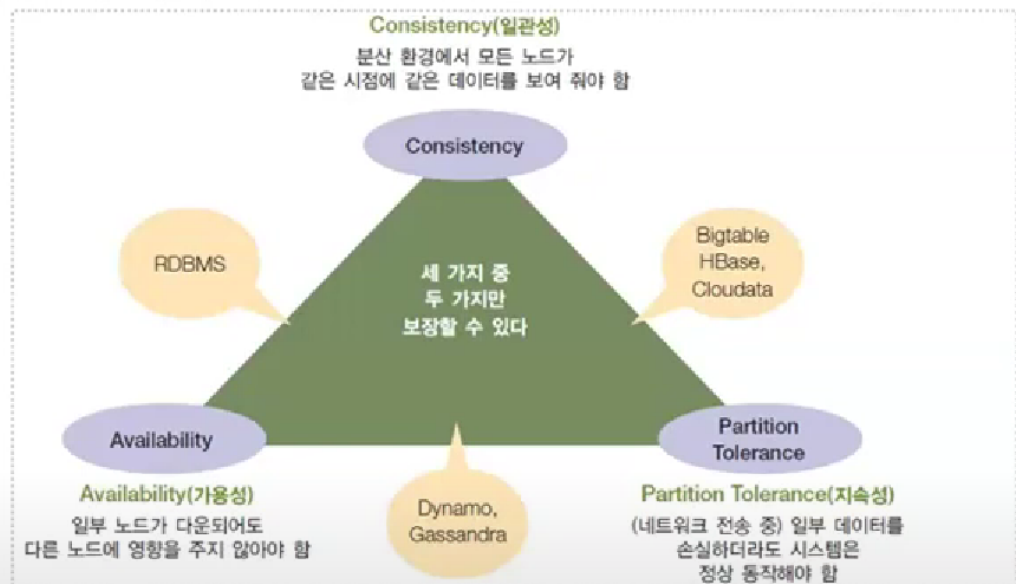


표 3-1 CAP 이론을 기준으로 한 RDBMS와 NoSQL의 비교

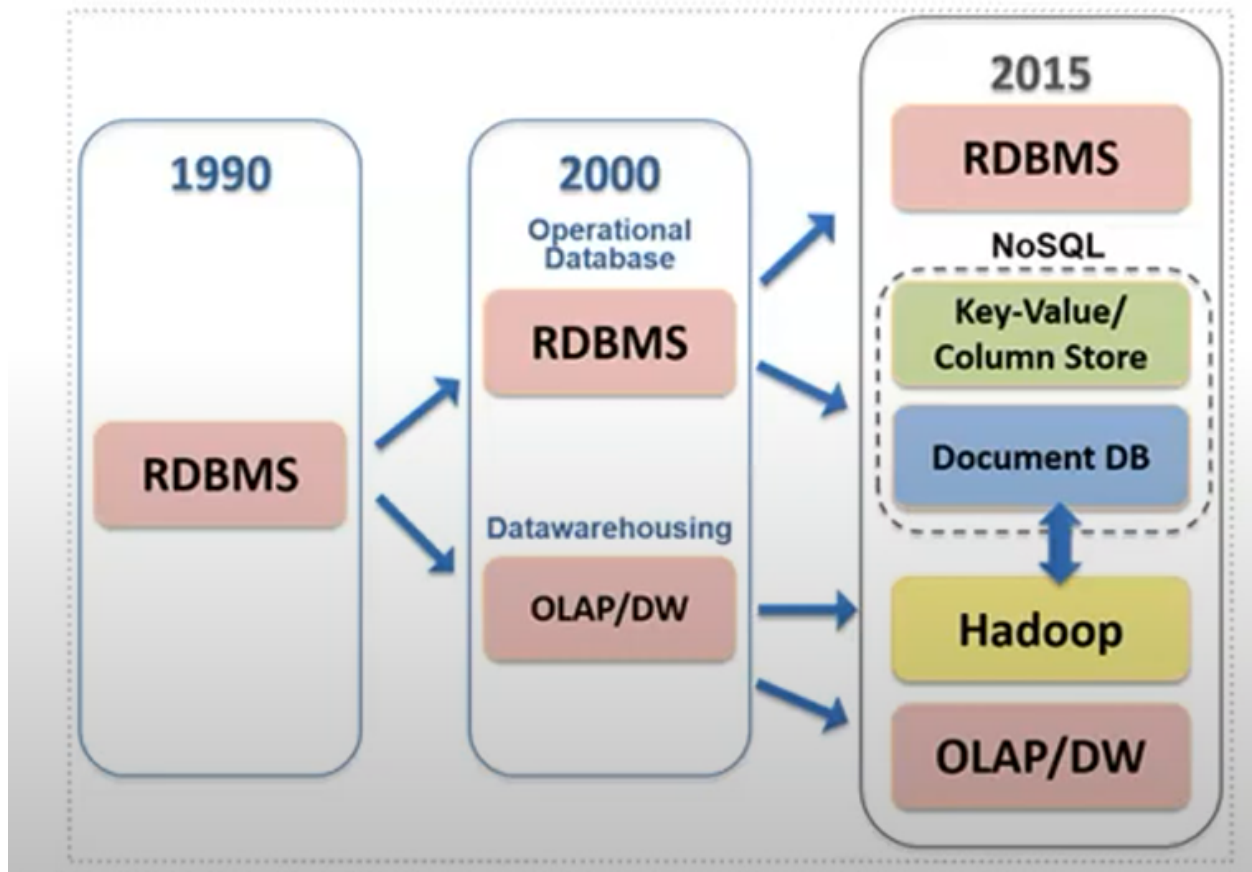
구분	설명	적용 예
RDBMS	일관성(C)과 가용성(A)을 선택	트랜잭션 ACID의 보장(금융 서비스)
NoSQL	일관성(C)이나 가용성(A) 중 하나를 포기하고, 지속성(P)을 보장	<ul style="list-style-type: none"> <li>■ C+P형 : 대용량 분산 파일 시스템(성능 보장)</li> <li>■ A+P형 : 비동기식 서비스(아마존, 트위터 등)</li> </ul>

- RDBMS는 'C, A'를 갖고 있다는 식으로 해석

- 구체적으로는 몰라도 됨 => RDBMS와 NoSQL이 상호 보완된다는 것만 알면 됨









### 3. RDBMS vs. NoSQL

- 서로 대치 하는 것이 아님 -> 서로 보완
- RDBMS : 트랜잭션이나 무결성을 보장해야하는 데이터 처리
- NoSQL : 배치성으로 데이터를 저장하고 처리



- RDBMS - 운영계 시스템(실제로 운영하는 여러가지 현상들 저장)
- OLAP/DW
  - 필요없는 예전 데이터들을 따로 모아서 관리해서 쓰기 위해
  - 분석하기 쉽게, 시간의 관점에서 따로 저장해서 분석용으로 관리하는 시스템
- NoSQL - 종류나 특징, 용도에 따라서 다양해져서 생긴 것  
=> 상호 보완적

## 2. NoSQL의 분류

Type	Example	
<u>Key-Value Store</u>	 redis	 riak
Wide Column Store	 HBASE	 cassandra
Document Store	 mongoDB	 CouchDB relax
Graph Store	 Neo4j	 InfiniteGraph The Distributed Graph Database

- 다양한 형태로 저장
- column (key와 value를 묶어 놓은 것), document (문서 자체)
- HBase, cassandra, mongoDB가 많이 쓰임

Key	Value
Name	Joe Bloggs
Age	42
Occupation	Stunt Double
Height	175cm
Weight	77kg

### Key-Value

Document 1	Document 2	Document 3
{ "prop1": data, "prop2": data, "prop3": data, "prop4": data }	{ "prop1": data, "prop2": data, "prop3": data, "prop4": data }	{ "prop1": data, "prop2": data, "prop3": data, "prop4": data }

**Document-Based**

- key-value (table과 같은 형태와는 다르다는 것)

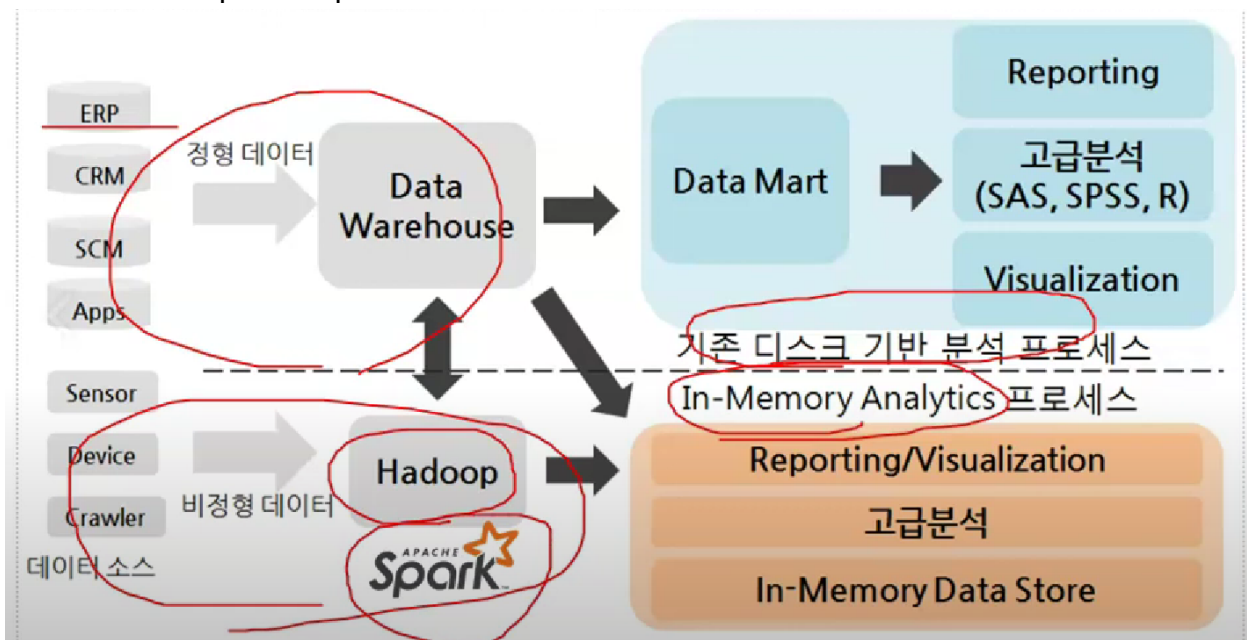
ColumnFamily			
Row Key	Column Name		
	Key	Key	Key
	Value	Value	Value
	Column Name		
	Key	Key	Key
	Value	Value	Value

## Column-Based



## Graph-Based

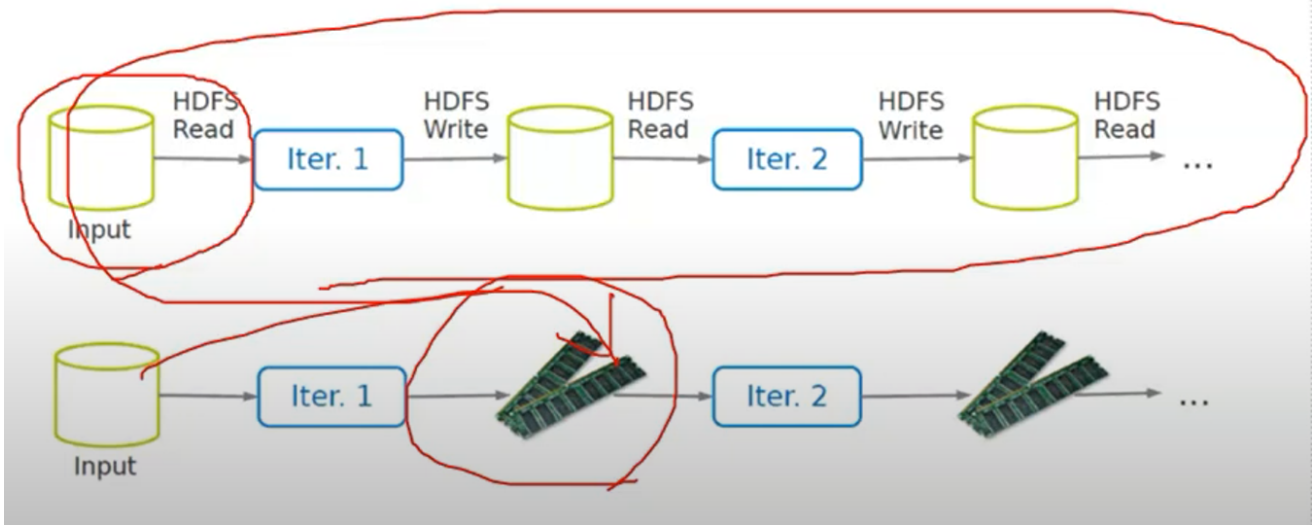
### 3. 아파치 스파크(Apache Spark)



- 아파치 스파크 : 하둡말고, 하둡 시스템을 부르는 말 / 하둡 시스템을 인메모리 기술로 쓴다는 말

## Apache Spark

- In-메모리 기반 분산 고성능 클러스터 플랫폼
- 기존 하둡의 MapReduce 작업을 디스크 기반이 아닌 메모리 기반으로 옮겨 가속화(In-Memory)



- 속도 향상 (하드디스크에서 꺼내씀 -> 메모리에다가 올려 놓고 씀)
- HDD(대용량) -> 메모리값 감소 -> Memory