DB

7 NoSQL

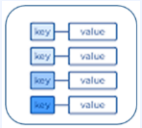
주요 질문

- ✓ NoSQL의 개념에 대해서 설명해 보세요.
- ✓ NoSQL 종류에 대해서 설명해 보세요.
- ✓ CAP 이론에 대해서 설명해 보세요.
- ✓ BASE 를 설명하고, ACID와 비교해 보세요.

관계형 DB의 한계를 벗어나 빅데이터 처리를 위해 데이터의 읽기보다 쓰기에 중점을 둔, 수평적 확장이 가능하며 다수 서버들에 데이터 복제 및 분산저장이 가능한 DBMS (Not Only SQL)

특징	설명
Schema-less	▪ 데이터 모델링을 위한 고정된 데이터 스키마 없이 키(Key)값을 이용해 다양한 형태의 데이터 저장과 접근이 가능한 기능을 이용 ▪ 데이터를 저장하는 방식에는 크게 컬럼(Column), 값(Value), 문서 (Document), 그래프(Graph)의 4가지로 나뉨
캐싱(Caching)	▪ 대규모의 질의에도 고성능 응답속도를 제공할 수 있는 메모리 기반의 캐싱 기술의 적용이 매우 중요하고 개발 및 운영에서도 일관 되게 적용할 수 있는 구조
탄력성(Elasticity)	▪ 시스템의 일부 장애에도 불구하고 시스템에 접근하는 클라이언트, 응용시스템의 다운타임이 없도록 하는 동시에 대용량 데이터의 생성, 업데이트, 질의에 대응할 수 있도록 시스템 규모와 성능 확장이 용이하고 입출력의 부하 분산에도 용이한 구조를 갖춤
저렴한 클러스터	▪ PC 수준의 하드웨어를 활용하여 다수의 노드를 구성하고, 수평적 확장 및 데이터 복제 및 분산 저장 가능

NoSQL 은 크게 4가지 종류가 존재

종류	설명	예시	제품
Key-Value 기반	▪ 단순하고 빠른 Get, Put, Delete 기능을 가장 기본 DB ▪ 메모리 기반에서 성능을 우선하는 시스템과 빅데이터를 저장·처리할 수 있는 방식		Dynamo, Redis
Column Family 기반	▪ 관계형 DB의 테이블에 대응되는 컬럼 패밀리의 행으로 데이터를 저장하는 구조		Cassandra, Hbase, SimpleDB
Document 기반	▪ XML, JSON 등의 문서를 Key-Value 의 value 부분에 저장하는 구조		MongoDB, CouchDB
Graph 기반	▪ RDB의 엔트리 속성을 노드로 표현하고, 관계를 node간의 edge 로 표현하는 구조		Neo4J, OrientDB

출처 : https://docs.microsoft.com/ko-kr/dotnet/architecture/cloud-native/relational-vs-nosql-data

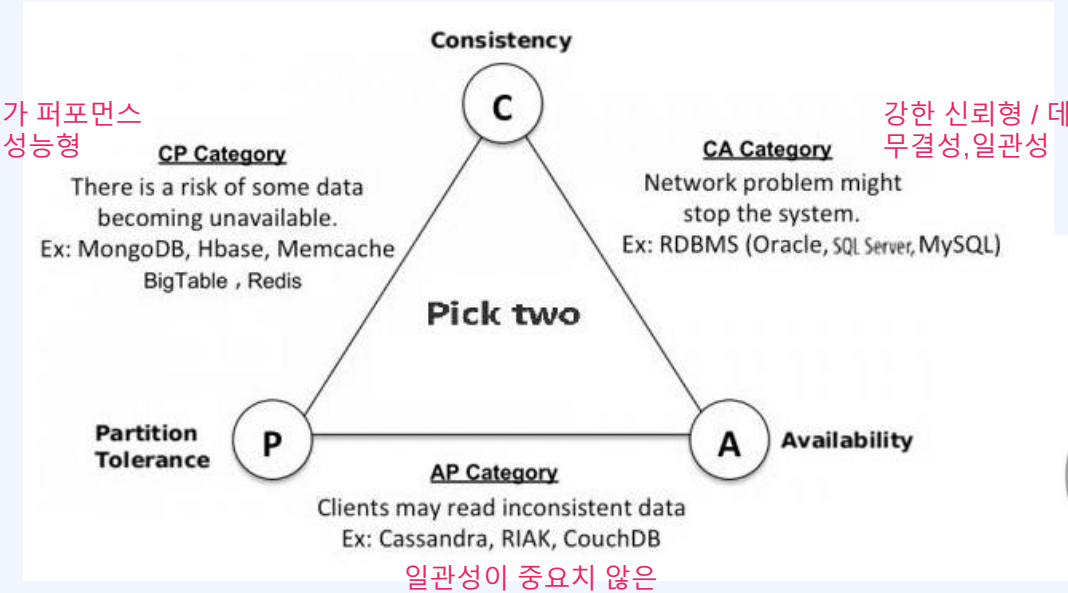
관계형 DBMS의 ACID와 대조적으로 가용성과 성능을 중시하는 NoSQL - 분산 시스템의 특성

특징	설명
Basically Availiable (기본적인 가용성)	▪ 다수의 실패에도 가용성을 보장, 다수의 스토리지에 복사본 저장 (주 서버가 안되더라도 백업 서버는 동작한다) ▪ 부분적인 고장은 있을 수 있으나, 나머지는 사용이 가능하다.
Soft-State (소프트 상태)	▪ 저장소는 쓰기 일관성이 있을 필요가 없으며 서로 다른 복제본이 항상 상호 일관성이 있을 필요도 없다. ▪ 분산 노드 간 업데이트는 데이터가 노드에 도달한 시점에 갱신
Eventually Consistent (최종 일관성)	▪ 일시적으로 일관성이 깨지는 상태가 되어도 일정시간 후에는 일관성이 있는 상태가 되는 성질 ▪ 저장소는 나중에 어떤 시점에서 일관성을 나타낸다.

BASE 와 ACID 의 비교

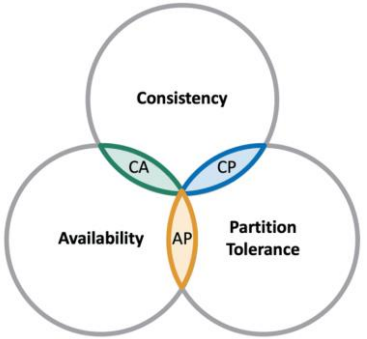
속성	BASE	ACID
적용분야	NoSQL	RDBMS
일관성 측면	약한 일관성	강한 일관성
중점 사항	Availability	Commit
시스템 측면	성능	데이터 무결성, 정합성
효율성	쿼리 디자인이 중요	테이블 디자인이 중요
범위	시스템 전체 특성	트랜잭션에 한정

분산 시스템에서 CAP 3가지 모두 만족 시키는 어렵고, 2가지만 선택 가능



모든 노드가 퍼포먼스 내야하는 성능형

강한 신뢰형 / 데이터 무결성,일관성



일관성이 중요치 않은 서비스

출처 : https://itwiki.kr/w/CAP_%EC%9D%B4%EB%A1%A0

CAP 이론

- 2000년 Eric Brewer가 “Towards Robust Distributed Systems” 첫 소개
- 2002년 MIT의 Seth Gilbert와 Nancy Lynch에 의해서 증명

Consistency	모든 노드들은 같은 시간에 같은 데이터를 보여줘야 함. (각각의 사용자가 항상 동일한 데이터를 조회함)
Availability	몇몇 노드가 다운되어도 다른 노드들에게 영향을 주지 않아야 함. (모든 사용자가 항상 읽고 쓸 수 있음)
Partition Tolerance	일부 메시지를 손실하더라도 시스템은 정상 동작을 해야 함 (물리적 네트워크 분산 환경에서 시스템이 잘 동작함)
C + A	-시스템이 죽더라도 메시지 손실은 방지하는 강한 신뢰형 -트랜잭션이 필요한 경우 필수적 -일반 RDBMS
C + P	-모든 노드가 함께 퍼포먼스를 내야하는 성능형 - 구글의 BigTable, HyperTable, HBase
A + P	-비동기화된 스토어 작업에 필수적 - Dynamo, Apache Cassandra, CouchDB, Oracle Coherence