

**과목명 │ 데이터베이스**

**담당교수 │ 박 재 휘 교수님**

**학과 │ 컴퓨터공학부**

**학년 │ 3학년**

**학번 │ 201401438**

**이름 │ 남 궁 찬**

**제출일 │ 2018. 6. 24.**

**이메일 │** [**201401438@inu.ac.kr**](mailto:201401438@inu.ac.kr)

**연락처 │ 010-6847-6270**

**과제차수 │ # 1**

Nosql은 비관계형이고 open-source이며, cluster구조에 친화적이고 21세기 웹 문화에 잘 맞는 데이터베이스이다. 또한 스키마 없는 데이터에 최적화되어 있다.

Nosql데이터 모델로는 다양한 메타데이터를 저장할 수 있는 키-값 스토어, json, xml문서 스토어, 열-친화 데이터베이스, 그래프 데이터 모델 등 다양한 모델을 사용하며 암묵적인 스키마를 쓸 수 있다. 중요한 것은 Nosql이집합 중심의 데이터베이스라는 것인데, 키-값 스토어에서는 집합이 값이고 문서 스토어에서는 집합이 문서이다. 열-친화 데이터베이스는 row키로 구분된다. Nosql은 집합 중심 데이터베이스이므로 클러스터 구조에서 매우 효율적이다. 데이터베이스에 집합을 한 번에 저장할 수 있기 떄문이다. 하지만 데이터를 나누기가 까다롭기 때문에 집합을 나누어야 할 경우 비효율적이다.

그래프 데이터베이스는 그래프 구조를 가지고 있어서 관계를 표현하기에 매우 유용하다. 관계형 데이터베이스에서는 수업때 배웠듯이 관계를 표현하기에 많은 이상현상(삽입, 삭제, 수정이상)이 있고 이를 해결하기 위한 정규화가 까다로운 편이다.

관계형 데이터베이스는 ACID 속성을 갖는다. 관계형 데이터베이스에서는 데이터의 일관성(consistency)을 위해 무결성 제약조건과 동시성 제어를 한다는 것을 배웠다. 반면 Nosql데이터베이스에서는 ACID속성을 완벽하게 만족하지는 않는다. 조금이나마 만족하기 위해 그래프 데이터베이스를 사용하는 방법이 있다. 집합형 갱신이라면 ACID를 만족한다.

[[1]](#endnote-1)CAP이론은 분산 시스템에서는 아래 3개의 속성을 모두 가지는 것이 불가능하다는 이론이다.

C는 Consistency로 ACID의 C가 아니다. 수업 떄 배운 ACID의 C는 데이터는 항상 일관성 있는 상태를 유지해야 하고 데이터의 조작 후에도 무결성을 해치지 말아야 한다는 속성이다. 반면 CAP의 C는 쓰기 동작이 완료된 후 발생하는 읽기 동작은 마지막으로 쓰여진 데이터를 리턴해야 한다는 것이고, 모든 노드가 같은 시간에 같은 데이터를 보여줘야 한다는 것이다.

A는 Availability로 특정 노드에 장애가 생겨도 서비스가 가능해야 한다는 의미이다.

P는Partitions Tolerance로 노드의 상태는 정상이지만 네트워크 등의 문제로 서로간의 연결이 끊어진 상황에서라도 동작해야 한다는 의미이다.

예를 들어 두 개의 노드가 서로 통신하지 않는 경우 Consistency는 만족하지 않는 대신에 Availability는 올라간다. 반대로 두 개의 노드가 통신하는 경우에는 Consistency는 만족하지만 Availability는 감소하게 된다.

한 예로 Amazon을 들여다보자. amazon에서는 일관성을 지켜서 사용자들이 쇼핑을 못 하게 되는 상황보다는 비일관성에 대처하는 방법을 선택했다. 사용자들이 쇼핑을 계속 할 수 있게 하되 적은 확률로 비일관성 문제가 생길 경우. 고객과의 상담전화와 같은 다른 방법으로 비일관성 문제를 해결한다.

Nosql의 가장 중요한 장점은 효율적으로 Impedence mismatch(객체-관계 불일치)를 줄일 수 있다는 것이다. [[2]](#endnote-2)객체-관계 불일치란 관계형 모델과 메모리 내 데이터 구조 간의 차이이며, 튜플 안의 값에 중첩된 레코드나 다른 구조를 포함할수 없어 관계형 표현으로 변환해야 하는 문제를 가진다.

또 다른 장점은 빠른 처리와 쉬운 개발이다. 미래에 빠른 처리와 쉬운 개발이 중요해진다면, Nosql이 중요해질 것이다.

1. CAP : https://embian.wordpress.com/2013/06/27/nosql-2/ [↑](#endnote-ref-1)
2. Impedence mismatch : http://sungsoo.github.io/2013/08/21/impedance-mismatch.html [↑](#endnote-ref-2)