



DataBase

1. Database 종류

2. Database 명령어

< 2조 >

조철호 박은진

이용준 황보은영



Database 종류

- ORDBMS

OODBMS의 한계를 극복하기 위해서 OODBMS와 RDBMS의 기술을 접목한 DBMS

현재 데이터베이스 제품의 주종을 이루는 DBMS

- NOSQL

NoSQL은 Not Only SQL의 약자

기존 RDBMS 형태의 관계형 데이터베이스가 아닌 다른 형태의 데이터 저장 기술



ORDBMS



ORDBMS장점

- 사용자 정의형 지원

사용자 정의형 데이터 타입의 저장 및 검색 가능

- 참조 타입 지원

하나의 객체 레코드가 다른 객체 레코드를 참조함으로써 참조구현을 이용한 네비게이션 기반 접근 가능



ORDBMS장점

- 중첩된 테이블

테이블 안의 하나의 컬럼이 또 다른 테이블로 구성됨으로써 복합 구조의 모델링 가능

- 대단위 객체 지원

이미지, 오디오, 비디오 등의 대단위 비정형 데이터를 위한 **Large Object**를 기본형으로 지원함

- 테이블간 상속관계

테이블 간의 상속 관계 지정함으로써 객체지향의 장점 수용



ORDBMS 종류

- PostgreSQL.
- Oracle Corporation의 Oracle 데이터베이스.
- IBM의 Informix
- Microsoft의 SQL Server
- Pivotal Software의 Greenplum 데이터베이스



NOSQL



NOSQL

- 관계형 데이터 베이스의 한계를 극복하기 위한 데이터 저장소의 새로운 형태로, 수평적 확장성을 가짐
- 문서, 그래프, 키 값, 인 메모리, 검색을 포함해 다양한 데이터 모델을 사용한다.
- NoSQL은 클라우드 환경에 맞는 저장 기술이다.



NOSQL 장점

- ✓ 유연성
NoSQL 데이터베이스는 일반적으로 유연한 스키마를 제공해 빠르고 반복적인 개발 가능
- ✓ 확장성
NoSQL 데이터베이스는 분산형 하드웨어 클러스터를 이용해 확장하도록 설계



NOSQL 장점

- ✓ **고성능**
NoSQL 데이터베이스는 특정 데이터 모델(문서, 키 값, 그래프 등) 및 액세스 패턴에 대해 최적화되어 관계형 데이터베이스를 통해 유사한 기능을 충족하려 할 때보다 뛰어난 성능을 얻게 해준다.
- ✓ **고기능성**
NoSQL 데이터베이스는 각 데이터 모델에 맞추어 특별히 구축된 뛰어난 기능의 **API**와 데이터 유형을 제공



NOSQL 장점 더있어!

- ❖ NoSQL은 RDBMS와는 달리 데이터 간의 관계를 정의하지 않는다.
 - 관계형 데이터 베이스인 RDBMS는 데이터의 관계를 Foreign Key 등으로 정의하고 Join 등의 관계형 연산을 한다고 하면, NoSQL은 데이터 간의 관계를 정의하지 않는다.
- ❖ RDBMS에 비해 훨씬 더 대용량의 데이터를 저장할 수 있다.
 - RDBMS의 복잡도와 용량의 한계를 극복하기 위한 목적으로 등장한 만큼 페타바이트급의 대용량 데이터를 저장할 수 있다.



NOSQL 장점 더있어!

❖ 분산형 구조이다.

- NoSQL은 기존 RDBMS와는 다르게 분산형 구조를 통해 데이터를 여러 대의 서버에 분산해 저장한다.

❖ 고정되지 않은 테이블 스키마를 갖는다.

- RDBMS와는 다르게 테이블의 스키마가 유동적이다. 데이터를 저장하는 컬럼은 각기 다른 이름과 다른 데이터 타입을 갖는 것이 허용된다.



NOSQL종류

- Document DB
- Wide Column DB
- Key-Value DB
- Graph DB

Document DB

- 일반적으로 가장 많이 알려진 NoSQL이다
- 테이블이 스키마가 유동적이다. 즉 레코드마다 각각 다른 스키마를 가질 수 있습니다. XML, JSON과 같은 Document를 이용해 레코드를 저장한다. 트리형 구조로 레코드를 저장하거나 검색하는데 좋은 Database이다.
- 종류: MongoDB, Azure Cosmos DB, CouchDB, MarkLogic, OrientDB

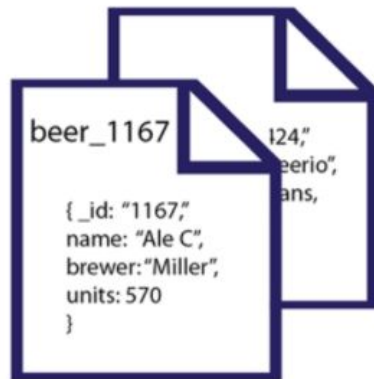
* Document Database

DOCUMENT STORE

Beers Table

1167	Ale C	Miller	570
3424	Beerio	Ians	340
5612	Amstel	Amtel	121
2409	Colt's	BeerCo	98

Beer Documents

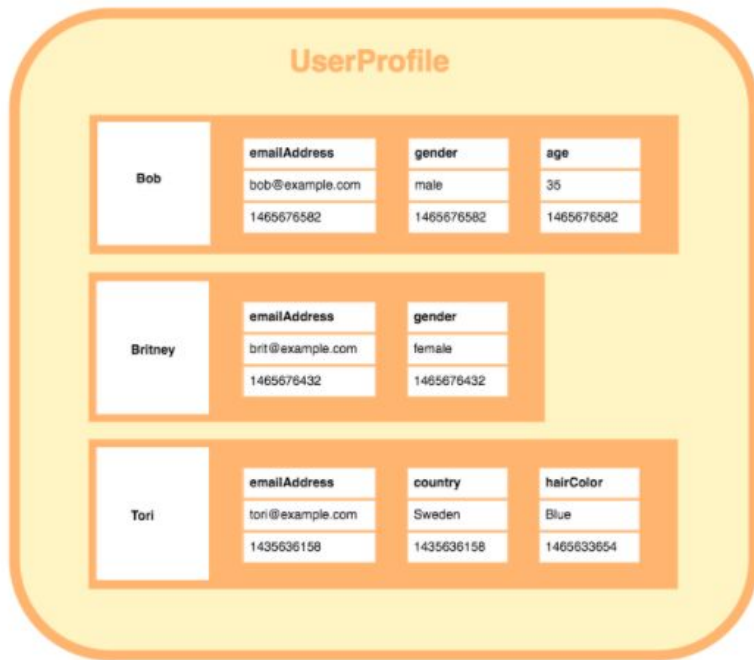


Document Database

wide Column DB

- 행마다 키와 해당값을 저장할 때마다 각각 다른값의 다른 수의 스키마를 가질 수 있다.
- 사용자 이름(Key)에 해당하는 값에 스키마들이 각각 다르다는 것을 알 수 있고. 대량의 데이터의 압축, 분산처리, 집계 처리(sum, count, avg 등) 및 쿼리 동작 속도 그리고 확장성이 뛰어나다.
- 종류:Cassandra, HBase, Google BigTable, Vertica, Druid, Accumulo, HyperTable

* Wide Column Database



Wide Column Database



Key-Value DB

- 기본적인 패턴으로 Key,value가 하나의 묶음으로 저장되는 구조로 단순한 구조이기에 속도가 빠르며 분산 저장 시 용이하다. Key 안에 (Column, Value) 형태로 된 여러개의 필드를 갖는다. 주로 Serverconfig, Session Clustering 등에 사용되고 액세스 속도는 빠르지만, Scan에는 용이하지 않습니다
- 종류: Redis, Oracle NoSQL Database, Voldmorte, Oracle Berkeley DB, Memcached, Hazelcast

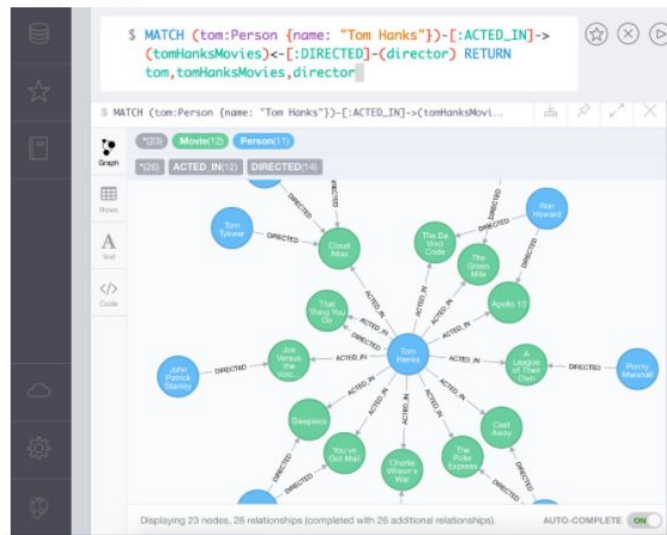
* Key Value Database

Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623

Key Value Database

- 데이터를 노드로 표현하며, 노드 사이의 관계를 엣지로 표현, RDBMS 보다 Performance가 좋고 유연하며 유지보수에 용이한것이 특징. Social networks, Network diagrams 등에 사용할 수 있다.
- 종류: Neo4j, Blazegraph, OrientDB, AgensGraph(국내솔루션)

* Graph Database



Graph Database



Database 명령어



DDL(Data Definition Language) - 데이터 정의어

데이터베이스를 정의할 때 사용하는 명령어

- **CREATE** : 데이터베이스,테이블등을 생성하는 명령어
- **ALTER** : 테이블을 수정하는 명령어
- **DROP** : 데이터베이스,테이블을 삭제하는 명령어
- **TRUNCATE** : 테이블을 초기화 시키는 명령어



DML(Data Manipulation Language) - 데이터 조작용어

테이블에 있는 행과 열을 조작하는 명령어

- **SELECT** : 데이터를 조회하는 명령어
- **INSERT** : 데이터를 삽입하는 명령어
- **UPDATE** : 데이터를 수정하는 명령어
- **DELETE** : 데이터를 삭제하는 명령어



DCL(Data Control Language) - 데이터 제어어

데이터베이스에 객체가 접근하고 사용하는 권한을 부여하고 회수하는 명령어

- **GRANT** : 특정 데이터베이스 사용자에게 특정 작업에 대한 수행권한 부여
- **REVOKE** : 특정 데이터베이스 사용자에게 특정 작업에 대한 권한을 박탈, 회수



TCL(Transaction Control Language)

- Transaction 제어어

데이터의 보안, 무결성, 회복 등을 정의하는데 사용하는 명령어

- COMMIT : 트랜잭션의 작업이 정상적으로 완료되었음을 관리자에게 알려줌
- ROLLBACK : 트랜잭션의 작업이 비정상적으로 종료되었을 때 원래의 상태로 복구



감사합니다.