

3TEAM

DATABASE

아이온커뮤니케이션즈 SW 개발자 양성교육

김예지 구본아 이성호 이충무



1. 데이터베이스와 DBMS
2. DBMS의 종류 및 특징
3. SQL

1. 데이터베이스란?

데이터베이스

컴퓨터 시스템에 전자적으로 저장된 구조화된 정보 또는 데이터의 조직화된 모음

데이터베이스 관리 시스템(DBMS)

- 다수의 사용자들이 데이터베이스 내의 데이터에게 접근, 사용할 수 있도록 해주는 소프트웨어
- 파일시스템 구조의 문제점을 보완하기 위해 만들어짐

DBMS의 특징

1. 데이터 무결성 – 삽입, 삭제, 갱신 후 정해진 제약조건 만족
2. 데이터 일관성 – 질의에 대한 응답이 변함없이 일정
3. 데이터의 회복 – 시스템 장애 시 원래 상태로 복구
4. 데이터 보안 – 데이터의 노출, 손실로부터 보호
5. 데이터 효율성 – 응답시간, 생산성, 저장공간 등



DBMS

데이터베이스 관리 시스템

01

RDBMS

- RDB : 관계형 데이터 모델에 기초를 둔 데이터베이스
- 관계형 데이터 모델은 모든 데이터를 2차원 테이블 형태로 표현
- RDBMS는 관계형 모델을 기반으로 하는 DBMS 유형
- RDB(Relational DataBase)를 생성, 수정, 관리하는 소프트웨어

DBMS

데이터베이스 관리 시스템

01

RDBMS 특징

- 2차원 구조의 모델을 기반함
- 데이터 무결성, 트랜잭션 처리 등 기본적인 기능면에서 뛰어남
- SQL을 사용한 데이터 접근 방법

제품 : 오라클 데이터베이스, 마이크로소프트 SQL 서버, MySQL, IBM DB2, IBM Informix, SAP Sybase, Adaptive Server Enterprise, SAP Sybase IQ, 테라데이터

DBMS

데이터베이스 관리 시스템

02

OODBMS

- 객체 모델링과 데이터 생성을 지원하는 객체지향 DBMS
- RDB의 문제점을 보완하기 위해 만들어짐
- * RDB의 문제점
 - 데이터 타입의 제한과 새로운 타입의 생성 및 기존 타입의 확장 불가능
 - 비정형 복합 정보를 표현하기가 어려움
 - Impedance Mismatch 문제로 인해서 응용 개발 및 유지가 어려움

DBMS

데이터베이스 관리 시스템

02

OODBMS 특징

- 사용자 정의 타입을 지원하고 상속과 명세가 가능
- 비정형 복합 정보의 모델링이 가능
- 참조 구조를 이용해 네비게이션 기반 정보 접근 가능
- 프로그램 내의 정보 구조와 데이터베이스 스키마 구조가 유사

제품 : IRIS, UniSQL

DBMS

데이터베이스 관리 시스템

03

ORDBMS

- OORDBMS 모델을 가진 RDBMS이다.
- 객체지향 클래스를 자유롭게 정의하여 DB를 개발할 수 있다.
- 사용목적은 DB 설계의 추상화 수준을 높이는 것

제품 : Illustra, Informix Dynamic Server, PostgreSQL, IBM DB2, 티베로, 오라클 데이터베이스

DBMS

데이터베이스 관리 시스템

03

ORDBMS 특징

- 사용자 정의 타입 지원
- 참조 타입 지원
- 중첩된 테이블
- 대단위 객체 지원
- 테이블 사이의 상속 관계

DBMS

데이터베이스 관리 시스템

04

NoSQL

- Non SQL, Not Only SQL의 약자
- RDB와 반대되는 방식을 사용하여 데이터 간의 관계를 정의하지 않는다.
- 스키마가 없어 자유롭게 데이터를 관리 가능
- 데이터의 일관성을 포기하여 데이터를 분산시킴
- Scale-Out을 목표로 비용 문제를 해결

제품 : MongoDB, Cassand, Redis

DBMS

데이터베이스 관리 시스템

04

NoSQL 특징

- 데이터 간의 관계를 정의하지 않음 (테이블 간의 Join 불가능)
- 기하급수적인 대용량 데이터를 저장 가능
- 분산형 구조 : 여러 서버에 분산해 저장
- 스키마가 없어 다루기 쉬움
- Key에 대한 put/get 만 지원

DBMS

데이터베이스 관리 시스템

비교

구분	RDBMS	OODBMS	ORDBMS
데이터 모델	행/열, 스키마, 인덱스	사용자 정의 타입 및 비정형 복합 정보 타입	사용자 정의 타입 및 비정형 복합 정보 타입
질의 언어	SQL	OQL	SQL 확장
대규모 처리 능력	탁월	보통	탁월
시스템 안정성	탁월	보통	탁월
장점	오랜 기간 검증된 시스템 안정성과 대규모 정보 처리 성능	복잡한 정보 구조의 모델 링이 가능함	RDBMS 안정성 OODBMS 모델링의 장점을 고루 가짐
단점	제한된 형태의 정보만 처리 가능, 복잡한 정보 구조 모델링이 어려움	기본 기능의 안정성 및 성능의 검증이 미비	표준화가 되어있지 않음 사용자의 혼란 유발

DBMS

데이터베이스 관리 시스템

비교

SQL의 장점

- 명확하게 정의 된 스키마, 데이터 무결성 보장
- 관계는 각 데이터를 중복없이 한번만 저장

NoSQL의 장점

- 스키마가 없어 유연
- 데이터는 애플리케이션이 필요로 하는 형식으로 저장 → 데이터 읽는 속도 향상
- 수직 및 수평 확장이 가능하기에 모든 읽기/쓰기 요청 처리 가능

DBMS

데이터베이스 관리 시스템

비교

SQL의 단점

- 데이터 스키마는 사전에 계획해야 함
- 관계를 맺고 있어 매우 복잡한 쿼리가 만들어질 수 있음
- 수평적 확장이 어려움, 대체적으로 수직적 확장

NoSQL의 단점

- 데이터 구조 결정을 하지 못할 수 있음
- Update를 해야 하는 경우 모든 컬렉션에서 수행

SQL

데이터베이스를 정의, 제어, 관리하는 구조적 질의 언어

DDL

DDL(Data Definition Language) : 데이터 정의어

- 데이터베이스를 정의하는 언어
- 데이터의 생성, 수정, 삭제 기능으로 전체 구조를 결정하는 역할을 하는 언어
- SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 정의하거나 변경 또는 삭제할 때 사용

종류	역할
CREATE	데이터베이스, 테이블 등을 생성하는 역할을 합니다.
ALTER	테이블을 수정하는 역할을 합니다.
DROP	데이터베이스, 테이블을 삭제하는 역할을 합니다.
TRUNCATE	테이블을 초기화 시키는 역할을 합니다.

SQL

데이터베이스를 정의, 조작, 제어하는 구조적 질의 언어

DML

DML(Data Definition Language) : 데이터 조작어

- 정의된 데이터베이스에 입력된 레코드를 조회, 수정, 삭제 등의 역할
- 테이블에 있는 행과 열을 조작하는 언어
- 사용자가 응용 프로그램과 데이터 베이스 사이에 실질적인 데이터 처리를 위해 사용한다.
- 데이터베이스 사용자와 데이터베이스 관리 시스템 간의 인터페이스를 제공합니다.

종류	역할
SELECT	데이터를 조회하는 역할을 합니다.
INSERT	데이터를 삽입하는 역할을 합니다.
UPDATE	데이터를 수정하는 역할을 합니다.
DELETE	데이터를 삭제하는 역할을 합니다.

SQL

데이터베이스를 정의, 제어, 관리하는 구조적 질의 언어

DCL

DCL(Data Control Language) : 데이터 제어어

- 데이터베이스에 접근하거나 객체에 권한을 주는 역할

종류	역할
GRANT	특정 데이터베이스 사용자에게 특정 작업에 대한 수행권한 부여 합니다.
REVOKE	특정 데이터베이스 사용자에게 특정 작업에 대한 수행 권한을 박탈, 회수 합니다.

TCL(Transaction Control Language)

- DCL과 구분하여, 트랜잭션을 제어하고자 할 때 쓰이는 언어이다.
- 데이터의 보안, 무결성, 회복, 무결성 수행 제어 등을 정의하는데 사용한다

종류	역할
COMMIT	트랜잭션의 작업이 정상적으로 완료되었음을 관리자에게 알려준다.
ROLLBACK	트랜잭션의 작업을 취소 및 원래대로 복구하는 역할을 합니다.
SAVEPOINT	저장점(SAVEPOINT)을 정의하면 롤백(ROLLBACK)할 때 트랜잭션에 포함된 전체 작업을 롤백하는 것이 아니라 현 시점에서 SAVEPOINT까지 트랜잭션의 일부만 롤백할 수 있다.

감사합니다.