

데이터베이스_220526_한지희

데이터베이스 용어

- **독립성 (Data Independence)**
 - 논리적 독립성 : 응용 프로그램과 데이터베이스를 독립시킴으로써, 데이터의 논리적 구조를 변경시키더라도 응용 프로그램은 변경되지 않는다.
 - 물리적 독립성 : 응용 프로그램과 보조기억장치 같은 물리적 장치를 독립시킴으로써, 데이터베이스 시스템의 성능 향상을 위해 새로운 디스크를 도입하더라도 응용 프로그램에는 영향을 주지 않고 데이터의 물리적 구조만을 변경한다.
- **무결성 (Data Integrity)** : 삽입, 삭제, 갱신 등의 연산 후에도 데이터베이스에 저장된 데이터가 정해진 제약 조건을 항상 만족해야 한다.
- **일관성 (Data Consistency)** : 데이터베이스에 저장된 데이터와 특정 질의에 대한 응답이 변함 없이 일정해야 한다.
- **종속성 (Data Dependency)** : (⇔ 독립성) 응용 프로그램의 구조가 데이터의 구조에 영향을 받는다.
- **중복성 (Data Redundancy)** : 같은 데이터가 중복되어 저장되는 것으로, 데이터 수정/삭제 시 연결된 모든 데이터를 수정/삭제해줘야 하는 문제점이 있다.
- **비일관성 (Data Inconsistency)** : 동일한 데이터의 여러 사본이 서로 다른 값을 보유하고 있는 상태로, 데이터 중복성에 이어서 발생할 수 있다.

데이터베이스 시스템의 목적

데이터베이스 시스템이 등장하기 전까지는 운영체제의 의해 지원되는 파일 시스템을 통해 데이터를 관리하였다. 이 때 여러가지 문제점이 존재하는데, 데이터베이스 시스템의 목적은 이 문제점을 해결하기 위하는 데 있다. **파일 시스템을 사용했을 때의 주요 문제점**은 다음과 같다.

- 데이터의 중복과 비일관성
- 데이터 접근 시 필요한 데이터를 편리하고 효율적으로 검색하기 힘들다.
- 데이터가 여러 파일에 흩어져 있고 파일 형식이 서로 다를 수 있다.
- 데이터 동시 접근 시 데이터가 잘못 업데이트될 수 있다.
- 무결성 문제
- 보안 문제

데이터베이스는 여러 사람들이 공유하여 사용할 목적으로 체계화해 통합, 관리하는 데이터의 집합을 말한다. DBMS를 통해 요구를 처리하고, SQL을 사용해 데이터에 접근한다. **데이터베이스**는 다음의 **장점**을 가진다.

- 데이터의 논리적, 물리적 독립성, 일관성, 무결성, 보안성 보장
- 데이터 중복 최소화
- 저장된 자료를 공동으로 이용할 수 있다.
- 데이터를 통합하여 관리할 수 있다.
- 데이터의 실시간 처리가 가능하다.

한편 **단점**도 존재한다.

- 전산화 비용이 증가한다.
- 대용량 디스크로의 집중적인 Access로 과부하(Overhead)가 발생할 수 있다.

관계형 데이터베이스 시스템 (RDBMS)

RDBMS는 테이블 기반(Table based) DBMS로, 테이블들의 집합으로 데이터들의 관계를 표현한다.

- 데이터를 테이블 단위로 관리
- 중복 데이터 최소화 (정규화)
- 여러 테이블에 분산되어 있는 데이터를 검색 시 테이블 간의 관계(join)를 이용해 필요한 데이터를 검색

데이터베이스 언어

DDL (Data Definition Language) : 데이터 정의어

- CREATE, ALTER, DROP, RENAME
- 데이터베이스 객체(table, view, index...)의 구조를 정의 (생성, 변경, 제거)

DML (Data Manipulation Language) : 데이터 조작어

- SELECT, INSERT, DELETE, UPDATE
- 데이터베이스 테이블의 레코드를 CRUD (Create, Read, Update, Delete)

DCL (Data Control Language) : 데이터 제어어

- GRANT, REVOKE

- 데이터베이스와 그 구조에 대한 접근 권한을 부여 혹은 회수

TCL (Transaction Control Language) : 트랜잭션 제어어

- COMMIT, ROLLBACK, SAVEPOINT
- DML 명령문으로 수행한 변경을 관리 (트랜잭션 관리)

CHAR과 VARCHAR의 차이

다른점	VARCHAR	CHAR
저장	크기보다 작은 값이 들어오면 해당 크기만큼만 저장한다.	크기보다 작은 값이 들어오면 해당 크기까지 공백을 채워서 크기만큼 저장한다.
비교	한 문자씩 나누어 앞부터 비교한다. 공백 또한 하나의 문자로 표현한다.	짧은 쪽 끝에 공백을 추가해서 2개의 데이터의 길이를 맞춘다. 그 후에 앞부터 한글자씩 비교한다.

DROP vs TRUNCATE vs DELETE

NATHAN

이름	나이
손영배	31
한지희	26

DROP NATHAN;

테이블 삭제해서 흔적조차 안 남음 ㅎㅎ!

테이블의 저장 공간도 release함.

TRUNCATE NATHAN;

테이블을 초기화함. 뼈대 빼고 다 지움.

DELETE NATHAN;

데이터만 남기고 삭제함. 데이터가 있던 자리는 그대로 있음.

영배랑 지희 왔다간 자리만 있음.

이름	나이
----	----

이름	나이

