

01-25 [DB]

관계형 데이터베이스의 주요 개념

[데이터베이스의 역사]

컴퓨터 시스템은 과거부터 자료를 분석하고 처리하는 일에 본격적으로 사용
이 시기에는 '파일'을 통해 분석한 자료를 저장

파일시스템에서는 개별 응용프로그램이 직접 파일에 접근하여 기록, 갱신, 삭제 가능.
파일에 있는 데이터는 응용프로그램에 무조건 의존

시간이 지남에 따라 파일에 기초한 정보 시스템에서 데이터는 급속하게 증가하고 이로 인해 많은

문제점이 발생

1. 데이터 종속성(data dependency)

만약 학생의 이름을 저장하는 필드의 길이를 20자리 → 30자리로 변경해야 한다면, 학생 정보 파일을 읽어서 처리하는 **모든 응용프로그램을 변경해야 하는 문제가 발생**
프로그램의 구조가 데이터 구조에 강하게 종속 됨.

2. 데이터 무결성 (data integrity)의 침해

파일시스템을 이용하는 경우 **무결성에 대한 책임은 오로지 개발자에게 있었음.**
응용 프로그램만이 파일내부에 데이터를 조작할 수 있기에 잘못된 데이터가 저장될 수 있는 여지를 비즈니스 로직에서 무조건 잡아야했음.

- 파일 : 자료를 저장하는 기본적인 방법
- 파일 시스템 : 파일에 기초하여 자료나 정보를 처리하는 시스템
- 종속성 : 데이터를 사용하는 프로그램의 구조가 데이터 구조(파일 구조)의 영향을 받는다는 것
- 무결성 : 저장된 데이터의 내용이 본래 의도했던 데이터의 형식, 범위를 준수해야함
Ex.) Age필드는 숫자 형식이며 20 ~ 60 범위를 갖는다.
이러한 조건을 위배하는 데이터가 저장되면 데이터의 무결성이 침해된 것
- 중복성 : 같은 내용의 데이터가 여러 곳에 중복하여 저장되는 것

3. 데이터 중복성(data redundancy)

과거의 정보시스템에서는 개별부서나 응용 프로그램에서 필요로 하는 데이터 파일을 각각 만들어 사용하는 일이 많았기에 동일 데이터가 여러 파일에 중복 저장되는 경우가 많았는데 이 경우 **저장 공간의 낭비, 데이터의 불일치** 등으로 이어짐

4. 데이터 표준화(data standardization)의 어려움

표준화 : 특정 무언가를 표현할 때 모두가 알아볼 수 있는 표준적인 규칙으로 만드는 것

Ex.) 학생 이름은 S-NAME으로 표준화 하여 사용

5. 데이터 보안성(data security)

데이터가 저장되어있는 파일은 그 내용이 Text등의 잘 알려진 형식으로 저장되기에 응용프로그램 없이도 쉽게 파일을 열어볼 수 있고, 파일의 공유를 위해 접근이 쉬운 위치에 파일을 저장했기에 보안을 유지하는 것이 어려웠음.

위와 같은 문제로 데이터베이스가 등장.

[데이터베이스의 등장]

파일 시스템의 단점을 극복하고 다수의 사용자들이 정보를 공유할 수 있어야 한다.

- 파일 형태로 여기저기에 흩어져 있는 데이터, 정보 취합
- 응용프로그램들이 운영체제를 통해 시스템 자원을 이용하는 것 처럼 모아놓은 데이터들을 관리하고 사용자(응용프로그램)와 데이터 사이에 인터페이스 역할을 할 수 있는 S/W

이 때 모아놓은 데이터의 집합을 [데이터베이스(database)]

데이터를 관리하는 S/W를 [데이터베이스관리시스템(DBMS : Database Management System)]

데이터베이스에 기초해서 데이터나 정보를 처리하는 체제 [데이터베이스 시스템(Database System)]

- 파일 시스템 개요

응용프로그램 ↔ 운영체제 ↔ 파일 / 응용프로그램이 운영체제를 거쳐 파일에 접근하듯

- DB 시스템 개요

응용프로그램 ↔ DBMS ↔ DB / 응용프로그램이 DBMS를 통해 DB에 접근할 수 있게 구성
DB시스템은 기존의 파일 시스템의 문제점을 보완

- DB : **단순히 데이터의 집합**. 이 데이터는 구조화되어 있을 수 있으며, 특정 주제나 목적에 따라 체계적으로 구성될 수 있음. 데이터베이스는 순수한 정보의 집합체로써, 데이터를 저장하고 관리하는데 사용.
- DBMS : **데이터베이스에 접근하여 데이터를 조회, 삽입, 수정, 삭제하는 등의 실질적인 작업을 수행하는 소프트웨어 시스템**. DBMS는 데이터베이스의 구조를 정의하고 데이터를 관리. 사용자나 응용 프로그램은 DBMS를 통해 데이터베이스와 상호 작용하여 필요한 작업을 수행.

1. 데이터 독립성(independency) 지원

DB시스템에서는 사용자 혹은 응용프로그램이 직접 데이터베이스에 접근할 수 없고 반드시 DBMS를 통해서만 접근이 가능,
DBMS는 데이터베이스 내에 있는 데이터의 물리적, 논리적 변화가 응용 시스템에 영향을 미치지 않도록 함으로써 데이터 독립성 보장.

2. 데이터 무결성 유지

DBMS는 데이터베이스 내에 저장될 데이터에 대하여 데이터의 타입, 길이, 값의 범위 등에 대한 정보를 가지고 있으며, 이를 위반하는 데이터가 들어올 경우 처리를 거절함으로써 데이터 무결성 지원

3. 데이터 중복성 및 불일치 최소화

DB내에 데이터는 한 개인의 관점이나 특정 부서의 관점이 아닌 DB를 사용하는 조직 전체의 관점에서 관리하므로, 동일 데이터가 여러 부서에서 사용되거나 하는 중복을 방지하고 이를 통해 데이터 간의 불일치 문제를 해결

4. 데이터 표준화의 용이성

DBA(데이터 베이스 관리자)가 설계과정을 주도함으로써 데이터의 구조에 관한 정보를 정의하여 DB를 사용하는 모든 개발자들에게 표준화된 규약을 제공

5. 높은 데이터 보안성

DBMS는 사용자 권한에 따라 DB내에 있는 데이터에 대한 접근을 제한할 수 있고,
DB는 일반적으로 DBMS를 통하지 않고는 외부에서 내용을 알아내기가 매우 어려우므로 데이터 보호 가능

6. 데이터 공유의 용이성

DB는 데이터를 통합 관리하며 여러 부서, 사용자들이 공유

[관계형 데이터베이스 모델]

데이터 베이스모델은 **사용자의 입장에서 데이터가 어떤 모양으로 표현되고 관리되는가에 따라 분류 가능**

대표적으로, '계층형 모델', '네트워크 모델', '관계형 모델'로 분류
최근에는 '객체지향 모델', '객체-관계형'모델도 발표

관계형 데이터베이스 모델은 **[2차원 구조의 테이블 형태를 통해 자료를 표현]**

데이터가 하나 이상의 열과 행의 테이블(또는 '관계')에 저장되어 서로 다른 데이터 구조가 어떻게 관련되어 있는지 쉽게 파악

관계형 데이터베이스는 왜 관계라는 이름이 붙여졌을까? 결론부터 말하자면 각 테이블의 행과 행이 연결되는 관계를 맺을 수 있기 때문이다.

하나의 테이블에 필요한 모든 정보를 전부 기입하면 필연적으로 중복이 발생.
관계형 DB모델은 이 때 테이블을 분리하고, 테이블간의 관계를 잡아줌으로써 해당 문제를 해결하고 유연한 설계를 가져갈 수 있음.