Data != DBMS

DBMS는 데이터를 관리하기 위한 시스템이며, 절대 필수적이지 않음

DBMS의 특성상 빠르거나 가볍지 않으며, 대신 편리함

따라서 DBMS는 선택사항(스마트폰(전화번호부)에는 있지만 컴퓨터에는 내장되어있지 않음)

DBMS는 고가치의 크고, 많은 사용자가 접근하는 데이터에 주로 사용함.

파일 형태로 데이터를 저장하면 데이터가 redundant(중복)될 수 있고, inconsistent해질 수 있음.

또한 접근이 어려울 수 있음(확장자 차이 등)

따라서 데이터가 있음에도 사용이 불가능함(isolated) (DBMS를 사용하지 않는 경우)

Integrity 문제(데이터 일관성?) 데이터가 지녀야 하는 통상적 기준(이름이라면 홍길동 같이)을 충족하는지를 DBMS가 대신 체크해줄 수 있음.

Atomic한 업데이트, 어떤 행위를 했을 때 모두 되거나 하나도 진행되지 않아야 함. 절반만 진행되면 안됨.

Concurrent한 접근 다수 사용자가 데이터베이스에 동시적으로 접근할 수 있어야 함. (통장 잔고가 여러 곳에서 빠져나가는 경우 등)

Security 문제, 사용자가 모든 데이터에 쉽게 접근할 수 없어야 함.

이 모든 것들을 충족하려면 DBMS가 필연적으로 무겁고 느려지게 됨.

수업 중에는 University DB를 주로 다루게 될 것(학생, 교수, 수업 등이 있는 데이터)

익숙해지는게 좋을거시다

View of data

Data abstraction -> 보여줄 것은 보여주고 나머지는 추상화. -> DBMS의 가장 대표적인 기능

Data model -> 데이터를 어떻게 설명, 어떻게 추상화해 줄 것인지에 대한 방법론. 크게 두 가지 레벨에 대한 수업을 들을 것. Conception한 모델과 Code 모델

Conception -> 개념적으로 모델을 미리 설계해보는 것. 모듈화를 하고, 플로우차트를 그리는 등. 대표적인 것이 Entity – Relationship model (ER모델)

Xml -> 구조가 있지만 flexible한 형태

Relational Database -> 열과 행이 있는 테이블의 형태

Abstraction의 레벨

최하층 = Physical level(OS). 가장 기계적인 부분. 데이터가 실제로 기억되는 것 (몇 비트냐, 어떤 저장공간에 있느냐, 자료형은 어떻게 되느냐) -> 인간이 신경쓰는 것이 아니라 DBMS와 OS가 해 줌

Logical level(=Conceptual level) = 우리한테 익숙한 것

View level = 구성원마다 다르게 보는 데이터베이스(학부생, 교수가 동일한 데이터베이스를 다른 형태로 보게 됨) 각 구성원은 전체의 일부만 볼 수 있음. 이 밑에 있는 레벨들은 보는 것이 아님.

Schema = 껍데기, 틀(데이터를 처리하는 방법), 설계도의 개념

거기서 찍어낸 것 = Instance (각 데이터마다 값이 다름)

새로운 하드웨어를 도입할 때 physical level은 바뀔 수 있으나, Logical level은 바뀌어선 안됨.

* Physical data independence
* 기계를 바꿨다고 해서 데이터의 내용이 바뀌어서는 안됨.

Data Definition Language(DDL)과 Data Manifulation Language(DML) 합쳐서 SQL

DDL => 스키마를 정의하기 위한 언어, metadata 등

DML => CRUD를 진행하는 언어

사실상 구분은 무의미함.

Functional -> c언어 등. 모든 과정을 다 하나하나 거쳐서 결과를 끄집어내기

Declarative lang -> 제일 좋은 거 줘, 뭐가 좋은지는 니가 알아서 해. 컴파일러가 알아서 해 줘.

그럼 SQL 언어는 무엇이냐? -> 전자보다는 후자에 가깝지만 완벽하진 않다.

Structured Query Lang. 구조화된 질의 언어. 시컬이라고 읽는 경우가 많음

Id name department salary

Conception -> 내가 그린 내 집의 그림, 내가 데이터베이스에 요구하는 요건들

Logical -> 설계사가 그린 집의 설계. 요건들을 어떻게 효율적으로 해결할 것이냐.

디자인에는 정답이 업다. 오답만 있을 뿐

// Database Design 파트 = 이번 학기에 할 것.

DB system = > DB engine 파트