#데이터 베이스의 특징

- 자기기술성(Self Description)

- 프로그램과 데이터의 분리

- 데이터 추상화(Abstract)

- …

# 시스템의 자기기술성

‘메타 데이터’를 통하여 데이터의 구조를 “직접” 알고 있지 않아도 데이터를 엑세스 할 수 있다.

* 데이터의 구조를 직접 알고 있어야 한다? -> 프로그램이 데이터에 의존한다.

# 프로그램과 데이터의 분리

데이터를 직접 아는 것이 아니라 간접적으로 알고 있는다.

* 프로그램과 데이터의 분리: 데이터베이스 내의 데이터 저장 구조가 변경되어도 Database 응용 프로그램은 영향을 받지 않는다. (변경할 필요가 없음, 업데이트 할 필요가 없음)

# 데이터 추상황

데이터 모델을 사용함으로써 저장구조와는 “별도로” 데이터의 의미를 표현하는 방법을 제공함.

* 사용자는 그 구조를 알 필요없이 그냥 잘 쓰기만 하면 된다.

# 데이터에 대한 다양한 뷰

사용자는 전체 데이터베이스 보다는 관심이 있는 데이터베이스의 일부를 뷰로 정의할 수 있음

* 데이터베이스를 도형으로 봤을 때 그걸 보는 사용자가 보는 뷰에 따라 도형이 다르게 보일 수 있다.

# 데이터베이스 스키마

스키마 : 데이터를 읽기 위한 틀 (형식에 맞게 넣어라), 구조

상태 : 이름, 학번, 학과, 전화번호등 스키마에 들어가는 정보

변동이 많은 것은 상태이다. (스키마의 변동은 새로운 필드가 생기거나 지워질 때)

스키마의 설계는 대단히 꼼꼼히 해야한다. (변화를 최소화 시킬 수 있도록)

# DBMS의 사용효과

* 표준화된 데이터 : 자동으로 표준화 => 효율성 증가.
* 데이터 구조 변경에 융통성 부여 => 데이터를 분리
* 개발 시간 단축 => 자연어 처리
* 항상 최신의 정보 제공 => 갱신 시 즉시 변경된 값을 접근가능
* 규모의 경제성 => 적정 규모에 맞으면 비용이 절감된다. 규모가 적으면 비용이 증가한다.

# 항상 DBMS를 사용해야 하는가?

DBMS의 문제점.

* 비용 증가.
* 성능 감소 (그 용도에 맞게 사용하지 않았을때…)
* 고장의 영향 확대 (파손 회복의 기능이 있지만..) (고장의 여파가 대단할 수 있다.)

# 데이터베이스 사용자

DBMS 개발자 => DB 관리자, DB 설계자, 시스템 분석가, 운영자, 최종 사용자, 도구 개발자.

# 데이터베이스 응용의 역사

60년대 - 계층 모델(네트워크 모델)

70~80년대 – 관계형 데이터 모델(연관성을 표현하는 방식의 모델) , 쉽고 간단하고 빠름

* 우리는 이것을 배울것이다.

80년대 – 객체 지향 붐이 일어나 객체 지향 응용 요구 등장

90년대 – XML을 이용한 데이터

2000년대 – 이미지, 오디오, 비디오 등 데이터 관리 (최신 DB기능 확장)

현재 – 빅데이터의 시대 (스키마 X -> 그냥 다 모으자! (비정형 데이터))