

# 제 1강

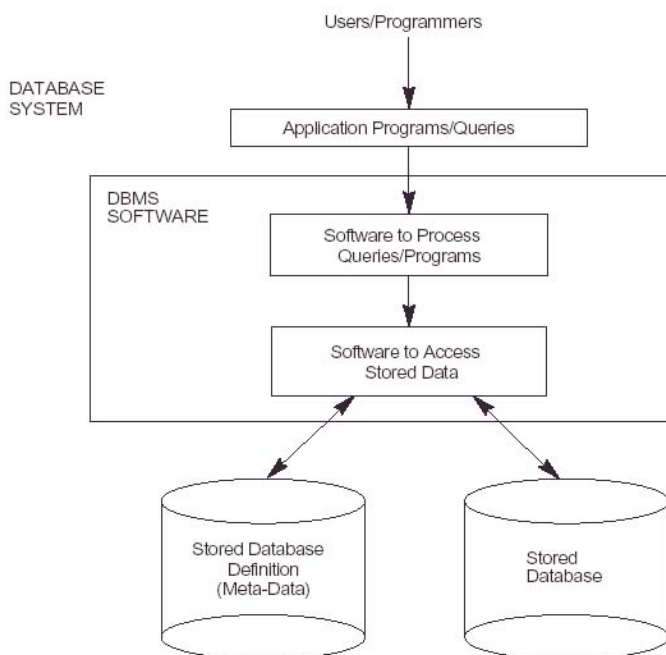
## 데이터베이스와 데이터베이스 사용자

- 강의 목표
  - 본 장에서는 데이터베이스와 관련된 기본 개념과 용어에 대하여 학습한다
- 기대 효과
  - 수강자는 다음과 같은 기본 개념을 올바르게 이해할 수 있다
    - ▷ 데이터베이스 / DBMS / 데이터베이스 시스템
    - ▷ 데이터베이스의 특징
    - ▷ 데이터베이스의 사용자
    - ▷ DBMS의 기능

### 개요

- 데이터(data)
  - 의미를 가지면서 기록될 수 있는 사실
    - ▷ 예: 사람 이름, 주소, 전화번호
- 데이터베이스(database)
  - 서로 연관이 있는 데이터의 모임
    - ▷ 예: 우리 가족의 전화번호부
- 작은세계(mini-world)
  - 전체 실세계의 일부분으로서 데이터베이스 구축의 대상
- 데이터베이스 관리 시스템(database management system: DBMS)
  - 데이터베이스를 생성하고, 데이터를 저장 및 관리할 수 있도록 기능을 제공하는 전문 프로그램
- 데이터베이스 시스템(database system)
  - 데이터베이스 관리 시스템 + 데이터베이스

### 데이터베이스 시스템 구조



[그림 1.1] 단순화된 데이터베이스 시스템 환경

## 데이터베이스의 예

- 데이터베이스 예제
  - 개체
    - ▷ STUDENT (Name, StudentNumber, Class, Major)
    - ▷ COURSE (CourseNumber, CreditHours, Department)
    - ▷ SECTION (SectionIdentifier, CourseNumber, Semester, Year, Instructor)
    - ▷ GRADE\_REPORT (StudentNumber, SectionIdentifier, Grade)
    - ▷ PrevREQUISITE (CourseNumber, PrevquisiteNumber)

STUDENT	Name	StudentNumber	Class	Major
	Smith	17	1	CS
	Brown	8	2	CS

COURSE	CourseName	CourseNumber	CreditHours	Department
	Intro to Computer Science	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Database	CS3380	3	CS

SECTION	SectionIdentifier	CourseNumber	Semester	Year	Instructor
	85	MATH2410	Fall	98	King
	92	CS1310	Fall	98	Anderson
	102	CS3320	Spring	99	Knuth
	112	MATH2410	Fall	99	Chang
	119	CS1310	Fall	99	Anderson
	135	CS3380	Fall	99	Stone

GRADE_REPORT	StudentNumber	SectionIdentifier	Grade
	17	112	B
	17	119	C
	8	85	A
	8	92	A
	8	102	B
	8	135	A

PREREQUISITE	CourseNumber	PrerequisiteNumber
	CS3380	CS3320
	CS3380	MATH2410
	CS3320	CS1310

[그림 1.2] 수강 데이터베이스의 예

## 데이터베이스의 특징

- 데이터베이스 시스템의 자기 기술성
  - 메타 데이터(meta-data): 기본 데이터베이스의 구성, 데이터 항목 타입, 저장 구조, 제약조건을 명시하는 데이터
  - 시스템 카탈로그(system catalog): 메타 데이터를 저장 및 관리하는 장소
  - 데이터베이스내의 사용자 데이터 자체 뿐만 아니라 데이터베이스 특성에 관한 데이터를 시스템에서 관리
- 프로그램과 데이터의 격리
  - 데이터베이스 시스템의 자기 기술성에 의한 장점
  - 데이터베이스 내의 데이터 저장 구조가 변경되어도 응용 프로그램은 영향을 받지 않음
  - 프로그램과 데이터의 독립성(program-data independence)이 높아짐

- 데이터 추상화
  - 복잡한 데이터베이스의 구조에 관한 상세한 정보를 감춤
  - 데이터 모델(data model) 사용자에게는 개념적인 뷰(conceptual view)만을 제공함
- 데이터에 대한 다중 뷰 제공
  - 뷰(view): 전체 데이터베이스로부터 추출된 가상의 데이터베이스
  - 동일한 데이터베이스인 경우에도 사용자들의 관점에 따라 다른 형태의 뷰를 제공
- 데이터의 공유
  - 다수의 사용자가 동시에 데이터베이스를 접근하는 것을 허용
  - 동시성 제어(concurrency control) 기능을 가짐

(a)

TRANSCRIPT	StudentName	Student Transcript				
		CourseNumber	Grade	Semester	Year	SectionId
	Smith	CS1310	C	Fall	99	119
		MATH2410	B	Fall	99	112
	Brown	MATH2410	A	Fall	98	85
		CS1310	A	Fall	98	92
		CS3320	B	Spring	99	102
		CS3380	A	Fall	99	135

(b)

PREREQUISITES	CourseName	CourseNumber	Prerequisites
Database	CS3380		CS3320
			MATH2410
Data Structures	CS3320		CS1310

[그림 1.4] 그림 1.2의 데이터베이스에 대한 두 가지 뷰

(a) 학생 성적의 뷰 (b) 선수과목의 뷰

## 데이터베이스 사용자의 분류

- 시스템 분석가(system analyst)
  - 사용자의 요구사항 분석 후, 이를 만족하는 트랜잭션 명세를 설계하는 사람
- 데이터베이스 설계자(database designer)
  - 데이터베이스에 저장된 데이터를 선정하고, 데이터베이스의 구조 및 특성을 정의하는 사람
- 응용 프로그래머(application programmer)
  - 트랜잭션 명세를 참고하여 응용 프로그램을 작성하는 사람
- 데이터베이스 관리자(database administrator: DBA)
  - 데이터베이스 시스템과 관련된 자원들을 관리하는 사람
- 최종 사용자(end users)
  - 데이터베이스에 대하여 질의, 변경, 보고서 등을 작성하는 사람
    - ▷ 캐주얼 사용자(casual end user): 비정기적으로 데이터베이스를 접근하는 전문 사용자
    - ▷ 초보 사용자(parametric or naive user): 미리 일정한 용도로 작성된 프로그램을 기계적으로 사용하는 사용자(은행 점원, 비행기 예약 직원)
    - ▷ 전문 사용자(sophisticated end user): 복잡한 응용을 개발하며, DBMS의 기능을 충분히 사용하는 전문가(엔지니어, 과학자, 비즈니스 분석가)
    - ▷ 독자적인 사용자(stand-alone end user): 편리한 패키지를 사용하여 개인 데이터베이스를 유지하는 사용자.

## 무대 뒤의 사람들

- DBMS 설계 및 구현자
  - DBMS 모듈들과 인터페이스들을 소프트웨어 패키지로 설계하고 구현하는 사람
- 도구 개발자
  - 데이터베이스 설계, 사용, 성능 개선등에 필요한 별도의 소프트웨어를 설계하고 구현하는 사람
- 운영 및 유지 보수자
  - 데이터베이스 시스템을 위한 하드웨어 및 소프트웨어 환경의 운영 및 유지 보수를 하는 사람

## DBMS의 기능

- 데이터 중복의 제어
  - 데이터 중복: 동일한 데이터가 여러 곳에서 저장 및 관리되는 것
  - 데이터 중복의 문제
    - ▷ 디스크의 낭비를 초래
    - ▷ 데이터 일관성(consistency)의 결여 유발
  - DBMS는 다양한 사용자들이 하나의 데이터베이스를 공유하도록 함으로서 이러한 중복의 문제를 해결
- 권한없는 접근의 통제
  - 접근이 허용된 사용자만 데이터 접근을 할 수 있도록 제어
  - 권한 관리 기능 제공
- 프로그램 데이터의 지속성 제공
  - 프로그램의 수행 및 종료와 관계없이 해당 프로그램이 관리하는 데이터가 보존되는 기능 제공
- 다양한 사용자 인터페이스 제공
  - 사용자들의 지식의 정도에 적합한 다양한 형태의 데이터베이스 접근 방법을 제공
  - 질의어, 프로그래밍 인터페이스, 메뉴기반 인터페이스 등
- 관계 표현 기능
  - 서로 다른 데이터간에 다양한 형태의 연관성을 표현할 수 있는 기능 제공
- 무결성 제약 조건의 시행
  - 무결성: 응용 데이터가 지정된 제약 조건을 만족해야 하는 성질
  - 무결성에 관한 제약 조건을 정의하는 기능 제공
  - 정의된 무결성 제약 조건을 검사하는 기능 제공
- 백업 및 회복
  - 하드웨어 및 소프트웨어의 고장으로부터 복구할 수 있는 기능 제공
  - 백업(backup): 데이터베이스내 데이터를 중복 저장하는 기능
  - 회복(recovery): 고장시 백업된 데이터로부터 최신 데이터를 복구시키는 기능

## 데이터베이스 사용의 효과

- 표준화에 공헌
  - 조직 내 모든 부서에서 표준화된 문서 관리로 업무 효율성 증대

- 응용 프로그램 개발 시간의 단축
  - 응용 프로그램에서 데이터 저장 및 관리 부분은 DBMS를 이용하여 처리하므로 응용 프로그램 개발의 부담이 줄어듦
- 데이터베이스 구조 수정에 융통성 제공
  - 데이터베이스 내의 자료 구조가 어떠한 이유로 변경되어도 사용자에게 대한 영향은 거의 없음
- 항상 최신의 정보를 제공
  - 한 사용자의 데이터베이스 변경을 나머지 사용자가 즉시 참조할 수 있음
- 규모의 경제성
  - 조직내 데이터와 응용이 통합되어 관리되므로 서로 다른 부서간의 업무 중복성이 줄어듦

## 데이터베이스를 사용하지 않아도 좋은 경우

- DBMS 사용의 오버헤드
  - 하드웨어, 소프트웨어, 교육 등의 측면에서 초기 투자 비용이 높음
  - 보안, 동시성 제어, 회복, 무결성 조건 등의 오버헤드가 큼
- DBMS가 적절하지 않은 경우
  - 데이터베이스와 응용이 단순하고 잘 정의되어 있으며, 변경될 가능성이 적은 경우
  - 실시간 데이터 처리 요구 사항이 엄격한 경우
  - 단일 사용자만이 접근하는 경우