

CSE235

**데이터베이스 시스템
(Database Systems)**

Lecture 03: DB 및 DBMS 개요

담당교수: 전강욱(컴퓨터공학부)

kw.chon@koreatech.ac.kr

세부 학습목표

- 1. 데이터(Data)와 데이터베이스(Database)를 정의할 수 있다.
- 2. 데이터베이스 정의에 함축된 4가지 개념을 설명할 수 있다.
- 3. 데이터베이스의 특징을 나열할 수 있다.
- 4. 데이터베이스 구축의 필요성을 설명할 수 있다.
- 5. 데이터베이스의 개념적 구성요소를 나열할 수 있다

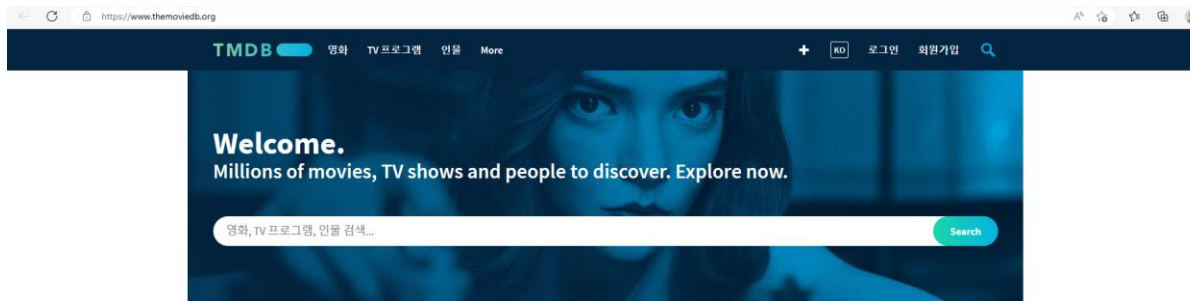
데이터베이스 개요

- 데이터 (data)
 - 의미를 가지면서 기록될 수 있는 알려진 사실
 - 행정 데이터, 의료 데이터, 성적 데이터, 실험 데이터
- 정보 (information)
 - 데이터가 어떤 목적에 의해 해석되거나 가공된 형태를 의미
- 데이터베이스 (database)
 - 서로 연관이 있는 데이터들의 모임
 - 간단한 데이터베이스의 예: 스마트폰에 저장된 전화번호 데이터
- 데이터베이스 관리 시스템 (DBMS: database management system)
 - 데이터베이스의 생성과 관리를 담당하는 소프트웨어 패키지
 - DBMS는 운영체제와 함께 중요한 시스템 소프트웨어 패키지로 분류됨
 - 대표적 DBMS 상품: Oracle, Microsoft MS-SQL (MySQL)

데이터베이스 시스템

■ 작은 세계 (mini-world, Universe of Database)

- 데이터베이스 구축의 대상이 되는 실세계(real world)의 일부분
- 데이터베이스는 특정한 목적을 위해 구축 및 운용되며, 이는 그 목적에 적합한 작은 세계를 구축하는 것에 해당
- 예제1: 스마트폰의 전화번호부 데이터베이스
 - 전화 번호에 대한 작은 세계를 이룸
- 예제2: 영화 데이터베이스
 - 영화 정보에 대한 독특한 도메인을 가짐

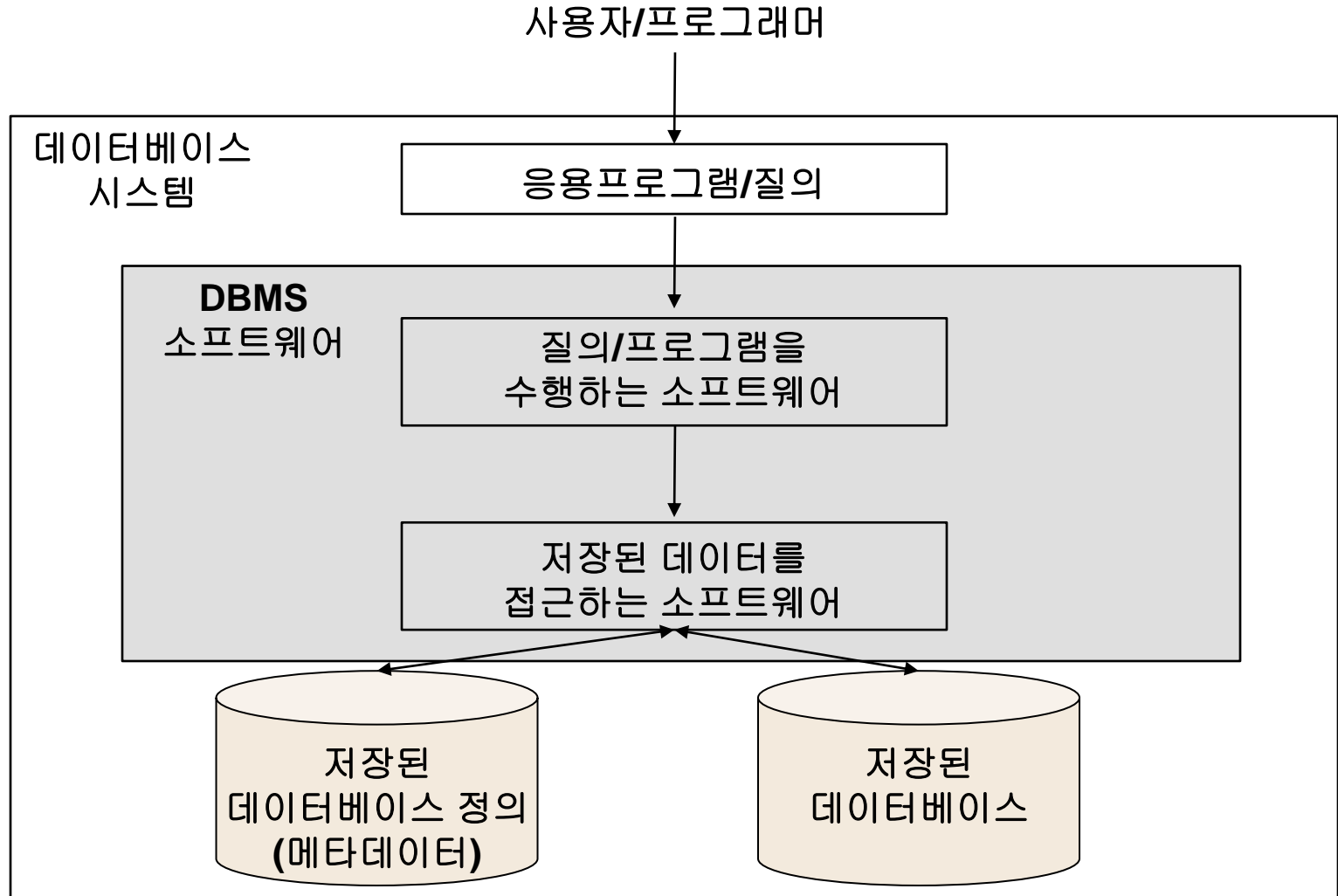


출처: <https://www.themoviedb.org/>.



출처: <https://danbis.net/10874>.

데이터베이스 시스템 환경



대학 정보 데이터베이스 예제

- 데이터베이스 파일의 종류

- STUDENT
- COURSE
- (COURSE의) SECTION
- GRADE_REPORT
- PREREQUISITE

- 파일의 설명 (각 파일은 동일한 유형의 데이터 레코드를 저장한다.)

- STUDENT: 학생 관련 데이터를 저장한다.
- COURSE: 개설된 과목에 관한 데이터를 저장한다.
- SECTION: 과목의 각 강좌에 대한 데이터를 저장한다.
- GRADE_REPORT: 학생이 수강한 강좌의 성적을 저장한다.
- PREREQUISITE: 각 과목의 선수 과목을 저장한다.

대학 정보 데이터베이스 예제(계속)

[그림 1.2]

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

GRADE REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

대학 정보 데이터베이스 예제(계속)

■ 몇 가지 용어의 (비정형적, informal) 정의

- 데이터 항목(data item): 레코드를 저장하는 항목(attribute)
 - 예: Name, Student_number, Class, Credit_hours 등
- 데이터 타입(data type): 데이터 항목이 가질 수 있는 값
 - 예: 정수, 문자열, {'A', 'B', 'C', 'D', 'F', 'I'} 등
- 관계(relationship): 여러 파일에 속하는 레코드들 사이의 연관성
 - 예: STUDENT와 GRADE_REPORT의 Student_number

■ 데이터베이스 조작(질의, 갱신)의 예

- (질의) 'Smith'의 성적을 검색
- (질의) 데이터베이스 과목의 선수 과목을 검색
- (갱신) 'Smith'를 2학년 학생으로 변경
- (갱신) 지난 학기 '데이터베이스' 과목에서 'Smith'의 성적을 'A'로 변경

파일 처리와 데이터베이스 비교

■ 파일 처리 (엑셀, 아래한글 파일 등)

- 예제: 성적 관리부와 회계 부서가 각기 데이터를 구축하고, 각기 응용 프로그램을 작성하여 사용함
 - 학생/직원 정보의 중복

■ 데이터베이스

- 데이터는 한 곳에 저장되고, 여러 응용 프로그램이 목적에 맞게 이를 사용함
- 예제: 데이터는 데이터베이스에 관리하고, 각각의 응용(성적 관리, 회계)은 이를 필요에 따라서 사용함

비교	DB	파일처리
공유	여러 사람이 효과적으로 공유	공유 불편(email, 파일전송 등)
최신성	변경 즉시 모든 사람이 변경된 데이터 활용	각자 가진 파일에서 변경된 부분을 갱신 또는 수정본 재-전송
뷰	거대한 DB에서 각자가 관심있는 부분을 뷰로 정의하여 활용	관심있는 부분만 보는 것이 어려움
접근권한	사용자 접근 제어, 암호화 등의 기능 제공	접근 제어가 어려움
복구	데이터 복구 가능	복구가 어려움

데이터베이스의 특징

■ 데이터베이스 시스템의 자기 기술성 (self-describing)

- 데이터베이스 카탈로그(catalog)에는 메타 데이터 (meta-data)가 저장
- 메타 데이터 (meta-data)
 - 데이터베이스 자체에 대한 정보
 - 테이블 구조, 테이블들이 저장된 위치, 인덱스 형성 체계 등 내부적 정보 관리

■ DBMS는 다수의 데이터베이스를 생성하고 관리 가능

- 각 데이터베이스에 대한 메타 데이터를 유지하기 때문임

RELATIONS

Relation_name	No_of_columns
STUDENT	4
COURSE	4
SECTION	5
GRADE_REPORT	3
PREREQUISITE	2

COLUMNS

Column_name	Data_type	Belongs_to_relation
Name	Character(30)	STUDENT
Student_number	Character(4)	STUDENT
Class	Integer(1)	STUDENT
Major	Major_type	STUDENT
Course_name	Character(10)	COURSE
...
Prerequisite_number	XXXXNNN	PREREQUISITE

데이터베이스를 위한 데이터베이스 카탈로그 예제:
Major_type은 전공명들로 구성되는 enumerated type으로 정의, XXXXNNN은 4개의 문자 뒤에 3자리 숫자가 나타난다는 의미.

데이터베이스의 특징(계속)

- 프로그램과 데이터의 분리(격리)로 프로그램-데이터 독립성 확보
 - 데이터베이스 내의 데이터 저장 구조가 변경되어도 데이터베이스 응용 프로그램은 영향을 받지 않는 성질
 - 데이터 독립성을 높임
 - 데이터에는 정해진 인터페이스(데이터베이스 질의)를 통해 접근
 - 병원 DB 예제: CT 촬영 데이터는 하나이나 이를 이용하는 사용자(SW)는 다양함

데이터 항목명	레코드 내 시작위치	길이(Byte)
Name	1	30
Student_number	31	4
Class	35	1
Major	36	4

STUDENT 레코드에 대한 내부 저장 형식.

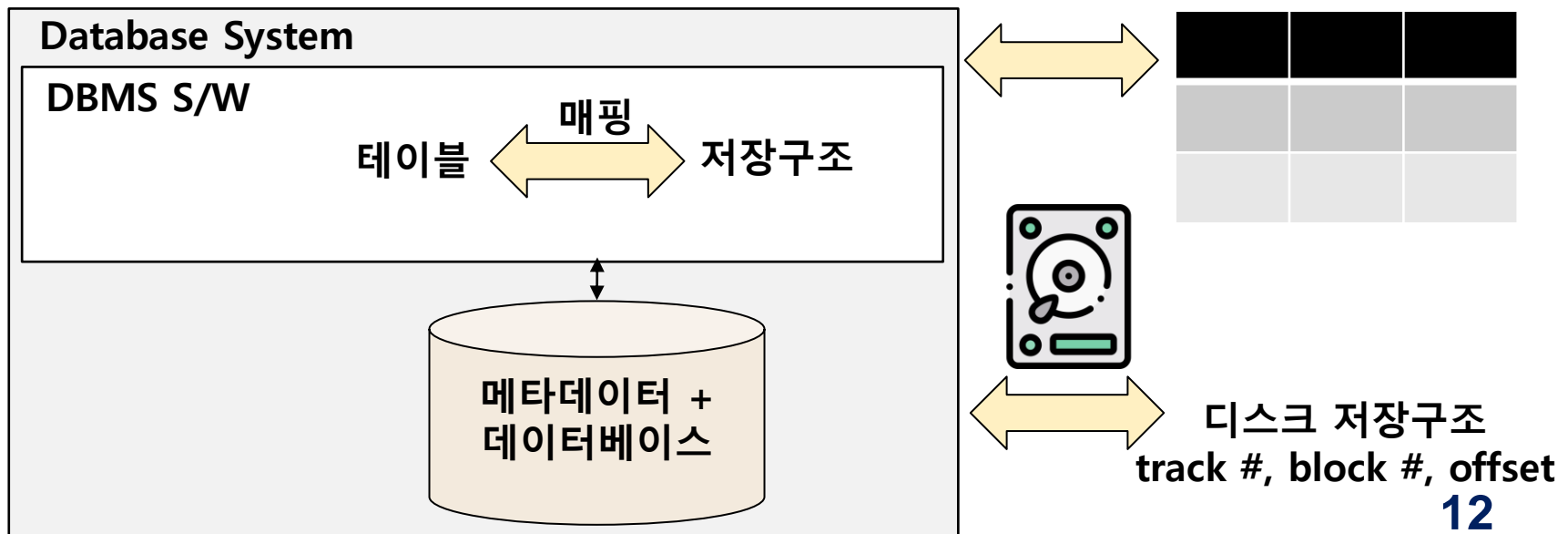
데이터베이스의 특징(계속)

■ 데이터 추상화 (data abstraction)

- 데이터모델(data model)을 사용하여 저장 구조의 자세한 내용은 사용자로부터 은닉시키고, 사용자에게는 각자의 요구에 맞는 개념적인 뷰(view)만을 제공
- DB뿐 아니라, 운영체제 등 전산 분야에서 추상화가 많이 사용

■ 데이터에 대한 다양한 뷰(view) 제공

- 사용자는 전체 데이터베이스 보다는 관심이 있는 데이터베이스의 일부를 뷰로 정의할 수 있음
- 병원 DB 예제: 의사는 진료 기록, CT 촬영 데이터에 관심이 있는 반면에, 원무과에서는 접수, 수납 등의 데이터에 관심이 있음



데이터베이스의 특징(계속)

- (그림 1.2로 부터 만들어지는) 대학교 데이터베이스에 대한 두 가지 뷰: (a) 학생의 성적 뷰(transcript: 미국의 성적증명서), (b) 과목의 선수과목 뷰
- DBMS는 테이블 정보를 가공하여 다양한 뷰로 사용자에게 필요한 데이터셋을 제공 가능

(a)

		Student	Section	Grade_Report		
TRANSCRIPT	StudentNAME	Course Number	Grade	Semester	Year	SectionId
	Smith	MATH2410	C	Fall	92	119
		COSC1310	B	Fall	92	112
	Brown	COSC3320	A	Fall	91	85
		MATH2410	A	Fall	91	92
		COSC1310	B	Spring	92	102
		COSC3380	A	Fall	92	135

(b)

	Course	Prerequisite	
PREREQUISITES	CourseName	CourseNumber	Prerequisites
	Database	3380	COSC3320
			MATH2410
	Data Structures	3320	COSC1310

데이터베이스의 특징(계속)

■ 데이터의 공유와 다수 사용자 트랜잭션 처리

- 여러 사용자가 (동시에) 동일한 데이터베이스 공유 가능하도록 지원
- 동시에 사용하더라도 일관성(consistency)을 보장하기 위한 동시성 제어(concurrency control) 기능 제공
 - 예제: 비행기 좌석 예약 시, 한 좌석은 한 사람에게만 배정되어야 함
- 트랜잭션: DB 작업을 수행하는 단위 프로세스
 - 예: 계좌이체, 좌석예약
- 트랜잭션의 주요 성질:
 - 고립성(isolation): 트랜잭션이 마치 혼자서 수행된 것 같아야 함
 - 원자성(atomicity): 트랜잭션은 수행되었거나, 수행되지 않아야 함

데이터베이스 사용자

- 데이터베이스 관리자 (database administrator, DBA)
 - 데이터베이스 시스템의 관리를 총괄하여 책임진 사람
 - DBMS 자체는 물론 데이터베이스 구축, 관리에 해박한 지식과 많은 경험이 요구됨

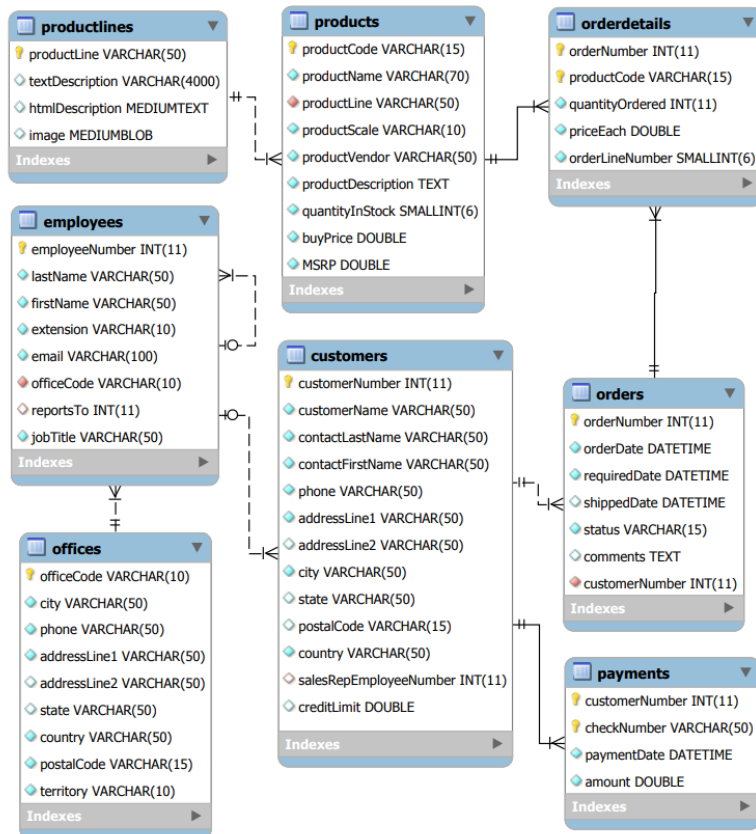
<p>데이터베이스 관리(DBA)</p> <ul style="list-style-type: none"> • 기업정보 데이터베이스 관리 • DBMS 구축 및 운영 	<p>필수요건</p> <ul style="list-style-type: none"> • 대졸 이상(4년제) • DBA 또는 엔지니어 경력 5년 이상 (ORACLE or SYBASE or MSSQL) • 대용량 DB운영 실무 경력자 • 시스템 장애/성능 분석 및 쿼리 튜닝 능력 보유자 <p>우대요건</p> <ul style="list-style-type: none"> • 이기종 DBMS간 마이그레이션 경력자 • SYBASE/HANA DB 능숙자 	<p>경력</p>	<p>0명</p>
---	---	-----------	-----------

2023년 NICE평가정보 상반기 신입/경력 직원 수시채용.

데이터베이스 사용자 (계속)

■ 데이터베이스 설계자 (database designer)

- 데이터베이스의 설계를 책임지는 사람
 - (요구 분석 과정을 거쳐서) 관리할 데이터를 선정하고, 저장할 구조를 결정
- 실세계 현상을 모델링하는 기술이 요구됨 (관계형 데이터베이스 이해)



Book DB의 Enhanced Entity-Relationship 모델.

데이터베이스 사용자 (계속)

- 최종 사용자 (end user): 데이터베이스에 대하여 질의하고, 변경하고, 보고서를 작성하는 사람
 - 캐주얼 사용자(casual end users)
 - 비정기적인 데이터베이스 사용자, 매번 다른 형태의 정교한 질의를 수행, 중상급의 관리자
 - 초보 사용자(parametric or naive users)
 - 미리 일정한 용도로 작성된 프로그램을 사용하는 사용자; 은행 점원이나 여행사 예약 담당자 등 (기작성 트랜잭션 사용)
 - 전문 사용자(sophisticated end users)
 - 엔지니어, 과학자, 비즈니스 분석가 등으로, DBMS의 고급 기능을 이용하여 복잡한 응용을 개발
 - 독자적 사용자(stand-alone end user)
 - 자신만의 개인 DB를 구축 관리하는 사용자
 - 메뉴나 GUI를 사용하여 자신만의 데이터베이스를 관리/이용

데이터베이스 사용자 (계속)

- 시스템 분석가 / 응용 프로그래머 (소프트웨어 공학자)
 - 초보 사용자를 위하여 잘 정의된 기능의 응용을 분석/설계하고 구현하는 사람
 - (초보 사용자의) 요구 분석을 통한 트랜잭션의 명세를 작성
 - 그래픽 인터페이스 등의 구현을 통해 최종 사용자의 이용 편의성을 제공
 - DBMS의 내부 기능을 잘 이해하고 있어야 함

DBMS 시스템 및 도구 개발자

■ DBMS 설계 및 구현자

- DBMS 소프트웨어 자체를 설계하고 구현하는 업무를 담당하는 사람들
- 용용 및 시스템 프로그래밍에 익숙하고, DB 이론에 해박한 전문가
- 개발하는 모듈의 예
 - 카타로그 구현, 질의어 처리, 데이터 접근과 버퍼링, 동시성 제어, 회복, 보안 관리

DBMS 시스템 및 도구 개발자 (계속)

■ 도구 개발자

- 데이터베이스를 사용하는 데에 필요한 도구들을 설계하고 구현하는 사람들
- 도구의 예
 - 데이터베이스 설계 및 구축 도구, 성능 모니터링 도구, 자연어 혹은 그래픽 인터페이스, 시뮬레이션 및 테스트 데이터 생성기
- 일반적으로 DBMS 엔진 업체와는 별도의 (많은) 업체가 개발하여 판매

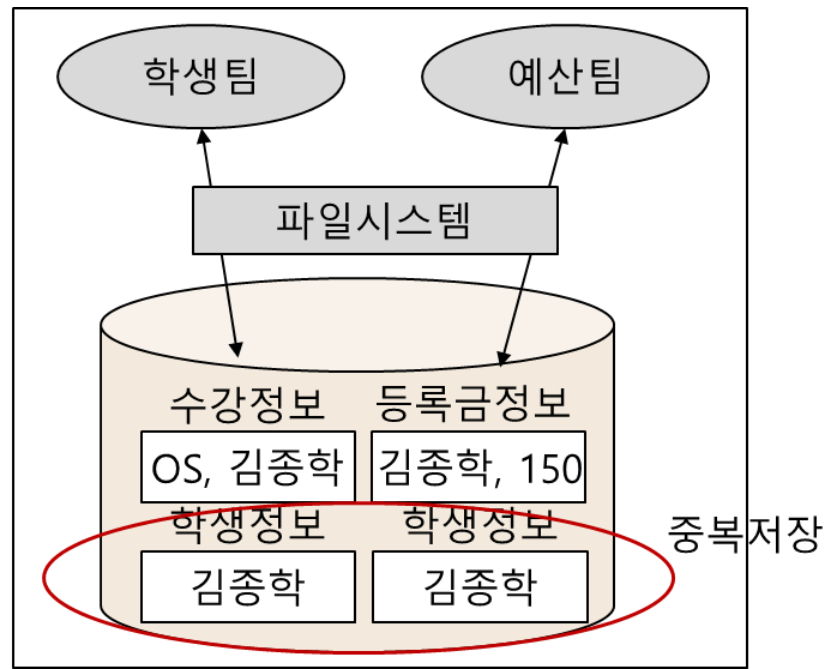
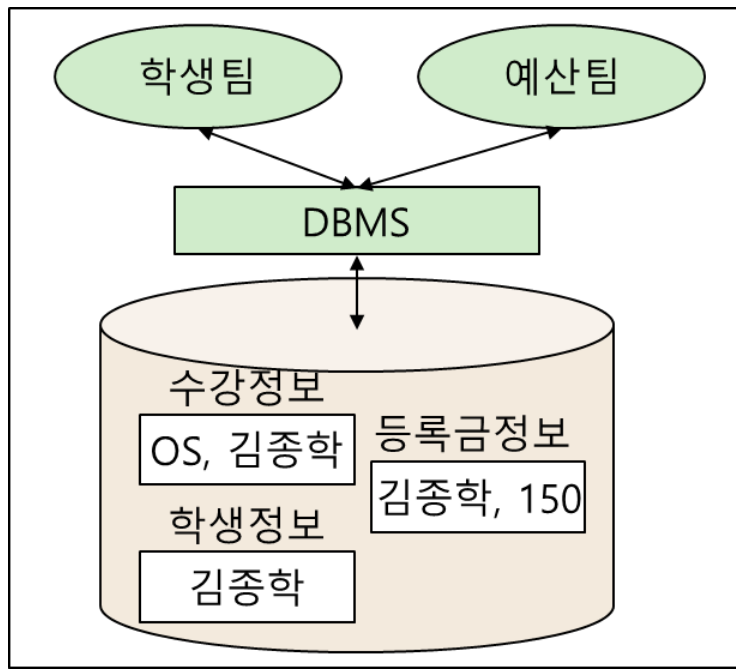
■ 운영 및 유지 보수 요원

- 데이터베이스 시스템을 운영하는 데 필요한 하드웨어 및 소프트웨어의 운영 및 유지 보수 담당 요원들
- 전문적 지식은 부족해도, 시스템의 청소, 백업 파일 관리 등 꼭 필요한 업무 수행

DBMS의 장점

■ 데이터 중복성(redundancy)의 제어 및 중복의 최소화

- 동일한 데이터가 중복되어 저장되는 것을 방지
- 중복 제어를 통한 데이터의 일치성(consistency) 보장
- 중복 최소화를 통한 메모리 낭비 방지
- 제어된 중복과 데이터 불일치
 - 제어된 중복: 성능 향상을 위해서, 응용 프로그램(혹은 사용자) 책임하에 데이터를 중복 관리한다.
 - 데이터 불일치: 중복 저장으로 인해 발생하는 불일치



DBMS의 장점 (계속)

■ 제어된 중복과 데이터 불일치의 예

- (a) 제어된 중복성: 성능 향상을 위하여 GRADE_REPORT 화일에 StudentName과 CourseNumber를 포함시키고, 두 속성의 값이 Student에서의 두 속성값과 일치하도록 DBMS가 보장함
- (b) 제어되지 않은 중복성: 그림 1.2의 STUDENT 레코드와 불일치하는 GRADE_REPORT 레코드의 예

(a)

GRADE REPORT	StudentNumber	StudentName	SectionIdentifier	CourseNumber	Grade
	17	Smith	112	MATH2410	B
	17	Smith	119	COSC1310	C
	8	Brown	85	MATH2410	A
	8	Brown	92	COSC1310	A
	8	Brown	102	COSC3320	B
	8	Brown	135	COSC3380	A

(b)

GRADE REPORT	StudentNumber	StudentName	SectionIdentifier	CourseNumber	Grade
	17	Brown	112	MATH2410	B

(17번 학생은 Brown이 아니라 Smith 임)

DBMS의 장점 (계속)

■ 보안 기능

- 권한 없는 사용자의 데이터 접근을 통제
 - 예: 재정 정보에 대한 접근 권한, 성적 입력 및 수정 권한
- 데이터의 중요성에 따른 다양한 형태의 접근 권한을 부여
- 다양한 권리를 가지는 다양한 접근 제어
 - 대표적 특권: DBA가 가진 신규 사용자 생성, 데이터 접근 권한 부여

■ 지속성 기억 공간 제공

- 프로그램 수행이 끝나더라도 관련 데이터/객체는 그 값을 저장하고 있어야 함
- 데이터를 데이터베이스에 영구적으로 보관/저장
- 데이터를 관리하는 프로그램 객체 자체도 지속성 기억 공간에 저장/관리

■ 효율적 질의처리를 위한 저장 구조 제공

- 레코드의 신속한 검색을 위해 인덱스 제공
- 질의 처리 최적화 모듈을 통해 최적의 질의 수행 계획 선택

DBMS의 장점 (계속)

- **백업(backup)과 회복(recovery) 기능 제공**
 - 시스템 고장 시에도 데이터의 일관성을 보장
- **다수의 사용자 인터페이스 제공**
 - 캐주얼 사용자에게는 질의어 I/F
 - 응용 프로그래머에게는 프로그래밍 언어 I/F
 - 초보 사용자에게는 폼 등의 GUI I/F 제공
- **데이터 간 복잡한 관계의 체계적 표현**
 - 테이블 간의 복잡하고 다양한 관련성을 표시
 - 그림 1.2에서 STUDENT는 GRADE_REPORT와, SECTION은 COURSE와 관련됨
- **데이터베이스의 무결성(integrity) 제약 조건의 시행**
 - 각 속성 값이 가져야 하는 제약 조건 (예: 나이는 0세 – 200세)
 - 테이블 간의 가져야 하는 제약 조건
 - 또 다른 예제:
 - 생년월일과 나이의 관계 (생년월일에 따라서 나이는 결정)
 - 키 값에 대한 제약 (키가 3M인 사람?)

DBMS의 장점 (계속)

■ 규칙을 사용한 추론과 수행

- 연역적 규칙을 이용하여 데이터베이스에 저장된 사실로부터 새로운 정보를 추론
 - 예: 생년월일을 바탕으로 자동적으로 나이를 갱신
- 자동으로 수행되는 능동 규칙의 정의 및 실행
 - 트리거 기능 제공, 저장된 절차(stored procedure)의 수행
- 예: 학생의 과목 성적이 변경되면, 그 학생의 총점과 평균을 변경한다.

■ 동시성 제어 기능 제공 (트랜잭션 개념)

- 여러 사용자가 동시에 DB를 접근하여 효율적으로 처리
- 동시 접근 시에도 데이터의 손실 방지, 일관성 보장

■ 데이터 독립성 제공 (데이터와 프로그램의 분리)

- 데이터베이스를 쉽게 사용할 수 있게 함
- 응용 프로그램 개발이 용이
- 내부 저장 구조를 변경하기 용이
- 프로그램 및 데이터베이스 유지 보수가 용이

데이터베이스 활용의 효과

■ 표준화된 데이터 관리

- 조직 내 모든 부서에서 표준화된 문서 관리로 업무 효율성 증대
- 부서, 프로젝트, 사용자 사이의 의사 교환 및 협조가 용이

■ 응용 프로그램의 개발 시간 단축

- 응용 프로그램의 상당한 부분을 DBMS 및 관련 소프트웨어가 처리함
- DBMS 사용하면 전통적 파일 시스템 사용시보다 개발 시간 단축

■ 데이터 구조 변경에 융통성 부여

- DB 내부의 자료 구조가 변경되어도 사용자에게 대한 영향은 거의 없음
- 현재의 DBMS는 기존 데이터나 응용 프로그램에 영향을 주지 않고 내부 구조 변경이 가능함

데이터베이스 활용의 효과 (계속)

■ 항상 최신의 정보를 제공 (실시간성)

- 사용자 중에서 한 사람의 갱신으로 나머지 사람은 즉시 변경된 값을 접근 가능
- 예약 시스템, 은행 데이터베이스 등에서 필수적 기능임

■ 규모의 경제성 (economics of scale)

- 부서마다 다른 방식으로 자료를 관리하는 것보다 통합 DB로 관리하는 것이 전체적인 관점에서 저 비용임
- 5명 미만 회사에서는 DB 사용 시 비용이 더 필요
 - 통상적으로 5명 이상이 되면 DB 사용시 비용 절감

DBMS를 사용하지 않아도 좋을 경우

■ DBMS를 사용하면 비용이 높아짐

- 높은 초기 투자 비용과 추가적인 (고사양, 고성능) 하드웨어 필요
- 데이터의 보안, 동시성 제어, 회복, 무결성 조건 등의 기능이 필요하지 않은 응용
 - 불필요한 기능이 오히려 오버헤드가 됨

■ 언제 DBMS가 필요치 않은가?

- 데이터베이스와 응용이 단순하고 잘 정의되어 있으며, 변경될 가능성이 적은 경우
 - 예: 교환기 시스템의 전화 경로 설정
 - 자주 변경되지 않으며, 응용이 제한적임
- DBMS 오버헤드로 인해 엄격한 실시간 데이터 처리 요구사항을 만족시키기 힘든 경우
- 다수의 사용자 데이터 접근이 필요하지 않은 경우

데이터베이스의 개념적 구성요소

■ 개체(entity)

- DB가 저장하는 유형, 무형의 정보 대상으로, 존재하면서 서로 구별 될 수 있는 요소 (개념 or 정보의 단위)
- 파일 구조에서 레코드에 대응
- 단독으로 존재 가능, 정보로서의 역할 가능
- 둘 이상의 속성(attribute)으로 구성됨

■ 속성(attribute)

- 개체의 특성이나 상태를 나타내는 요소로, 이름을 가진 정보의 가장 작은 논리적인 단위이지만, 단독으로는 아무런 정보도 제공하지 못 함

■ 관계(relationship)

- 일반적으로 개체들 간의 의미 있는 연결 또는 연관성을 나타내는 요소
- 관계 유형은 일 대 일 (1:1), 일 대 다 (1:n), 다 대 다 (m:n)로 구별

데이터베이스의 개념적 구성요소 (계속)

STUDENT

[그림 1.2]

Name	Student number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

속성

개체정보

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

관계정보

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

GRADE REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

데이터베이스의 저장 구조

■ DB의 논리적 구조

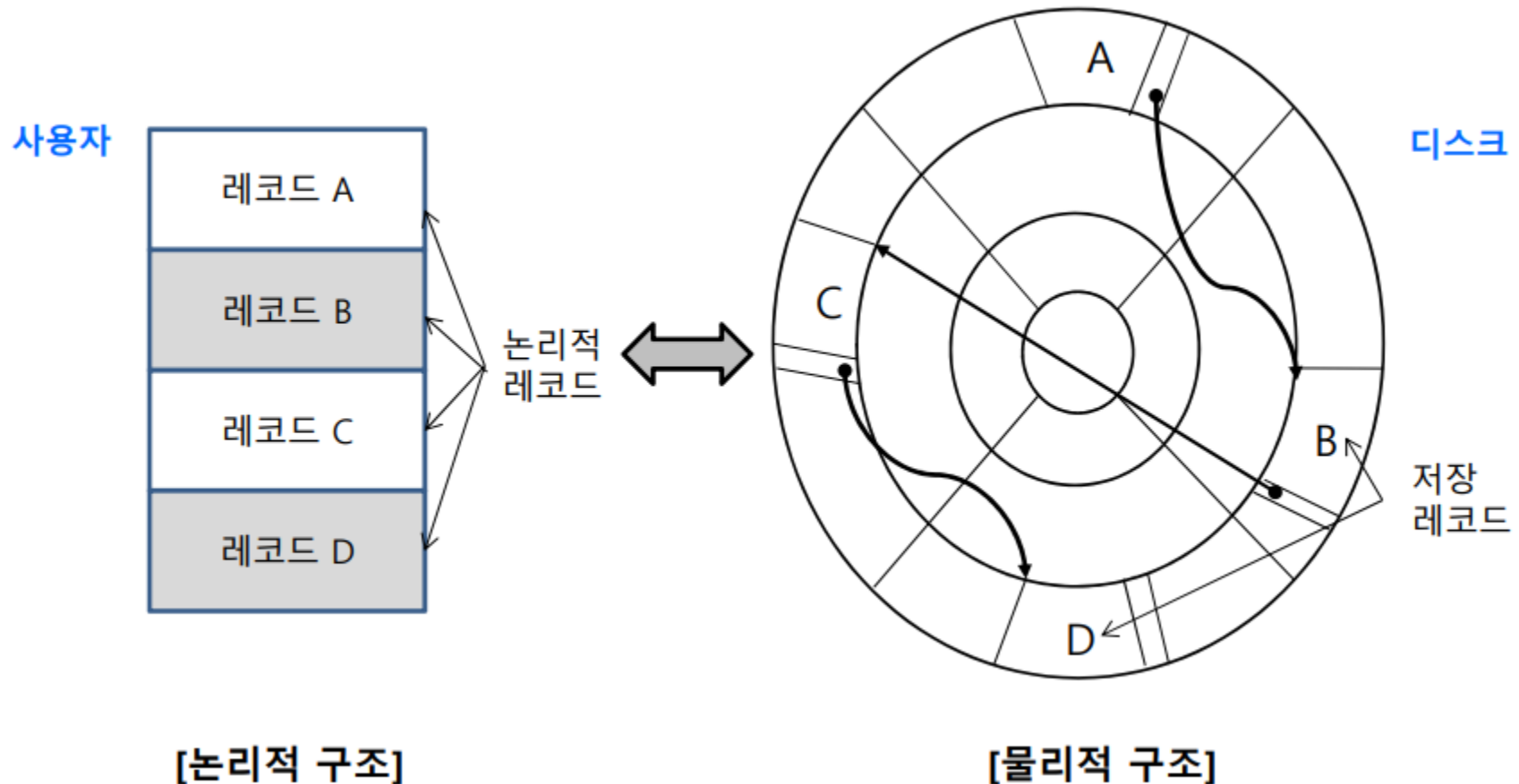
- 사용자 관점에서 본 DB 구조
- 데이터 사용자나 응용 프로그램 개발자가 생각하는 데이터 레코드(논리적 레코드)의 구조
- 물리적 저장 장치에 저장된 DB의 저장 구조를 사용자 관점에서 본 것

■ DB의 물리적 구조

- 시스템(저장 장치) 관점에서 본 DB의 구조
- 디스크 같은 물리적 저장 장치에 저장된 저장 레코드의 물리적인 배치
 - 블록, 인덱스, 포인터, 체인, 오버플로우 영역 등 포함

데이터베이스의 저장 구조

■ DB의 논리적 구조와 물리적 구조



요약

■ 용어 정의

- 데이터베이스, DBMS, 데이터베이스 시스템

■ 기존 파일 처리 시스템 대비 데이터베이스의 특징

- 카탈로그(메타 데이터), 프로그램-데이터의 독립성, 프로그램-연산의 독립성
- 데이터 추상화, 다중 뷰 지원, 여러 트랜잭션 간 데이터 공유

■ 데이터베이스 사용자의 분류

- 데이터베이스 관리자, 설계자, 최종 사용자, 시스템 분석가와 응용 프로그래머
- DBMS 설계자 및 개발자, 데이터베이스 도구의 개발자, 운영자와 유지보수 인력

요약 (계속)

■ DBMS의 기능들

- 중복성 제어 및 권한 검사
- 프로그램 객체와 데이터 구조에 대한 지속성 기억 공간 제공
- 규칙을 사용한 추론과 수행
- 다중 사용자 인터페이스 제공
- 데이터 사이의 복잡한 관련성 표현
- 무결성 제약조건 처리
- 백업과 회복 기능

■ 기존 파일 처리시스템 대비 데이터베이스의 장점

- 표준화 강화, 응용 개발 시간의 단축, 융통성 증가
- 최신 정보를 즉시 이용, 규모의 경제성

감사합니다!

담당교수: 전강욱(컴퓨터공학부)

kw.chon@koreatech.ac.kr