

# 데이터베이스 시스템



충북대학교 컴퓨터과학과

이종윤 교수님

# 목차

- **PART 1 데이터베이스 소개**
  - 제1장 데이터베이스와 데이터베이스 사용자
  - 2장 데이터베이스 시스템 개념 및 아키텍처
- **PART 2 개념적 데이터 모델링 및 데이터베이스 설계**
  - 3장 엔티티-관계(ER) 모델을 사용한 데이터 모델링
  - 4장 향상된 엔티티 관계(EER) 모델

# 목차

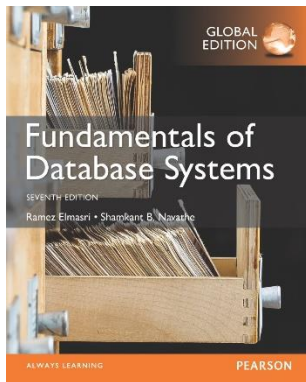
- **PART 3 개념적 데이터 모델링 및 데이터베이스 설계**
  - 5장 관계형 데이터 모델과 관계형 데이터베이스 제약 조건
  - 6장 기본 SQL
  - 7장 더 많은 SQL: 복잡한 쿼리, 트리거, 뷰
  - 8장 관계 대수와 관계 미적분
  - 제9장 ER 및 EER-관계 매핑을 통한 관계형 데이터베이스 설계
- **PART 6 설계 이론 및 정규화**
  - 제14장 관계형 데이터베이스의 함수 종속성 및 정규화 기본
  - 제15장 추가 종속성

# 목차

- **PART 7 파일 구조, 해싱, 인덱싱 및 물리적 데이터베이스 설계**
  - 16장 디스크 저장, 파일 구조, 해싱
  - 17장 파일 및 물리적 DD에 대한 인덱싱 구조
- **PART 8 쿼리 처리 및 최적화**
  - 제18장 쿼리 처리 전략
  - 19장 쿼리 최적화
- **PART 9 트랜잭션 처리, 동시성 제어 및 복구**
  - 제20장 TP 개념 및 이론 소개
  - 21장 동시성 제어 기술
  - 제22장 데이터베이스 복구 기술

# 제1장:

## 데이터베이스 및 데이터베이스 사용자



**소개**

# 개요

- 데이터베이스 유형 및 데이터베이스 애플리케이션
- 기본 정의
- 일반적인 DBMS 기능
- 데이터베이스의 예 (대학)
- 데이터베이스 접근 방식의 주요 특징
- 데이터베이스 사용자 유형
- 데이터베이스 접근 방식을 사용하는 이점
- 데이터베이스 기술의 역사적 발전
- 데이터베이스 기능 확장
- 데이터베이스를 사용하지 말아야 할 때

# 데이터베이스 유형 및 데이터베이스 애플리케이션

- 전통적인 응용 프로그램:

- 숫자 및 텍스트 데이터베이스

- 최근의 응용 프로그램:

- 멀티미디어 데이터베이스

- 지리정보시스템(GIS)

- 생물학 및 계놈 데이터베이스

- 데이터웨어하우스

- 모바일 데이터베이스

- 실시간 및 활성 데이터베이스

- 책의 첫 번째 부분은 전통적인 응용 프로그램에 초점을 맞춥니다.

- ~~최근의 여러 응용 프로그램은 책의 뒷부분에서 설명됩니다(예:  
24,25,26,27,28,29장)~~



# 최근 개발 사항 (1)

- 소셜 네트워크 시작많은 정보를 수집하다 사람들에 대한 것과 사람들 간의 커뮤니케이션에 대한 것 - 게시물, 트윗, 사진, 비디오와 같은 시스템:
  - 페이스북
  - 트위터
  - 링크드인
- 위의 모든 것데이터를 구성합니다
- 검색 엔진 - Google, Bing, Yahoo :자신의 저장소를 수집하다 검색 목적을 위한 웹 페이지

## 최근 개발 사항 (2)

- 소위 말하는 새로운 기술이 등장하고 있습니다.빅데이터베이스 소프트웨어와 공급업체는 방대한 양의 데이터를 관리해야 합니까? 웹에서 생성됨:
- 빅데이터 저장 시스템 대규모 분산 컴퓨터 클러스터를 포함함(25장)
- NoSQL(Not Only SQL) 시스템(24장)
- 큰 규모 유데이터 n의 nt 유형w는 "클라우드"에 상주합니다 즉, 수천 대의 기계를 사용하는 거대한 데이터 센터에 있다는 뜻입니다.

# 기본 정의

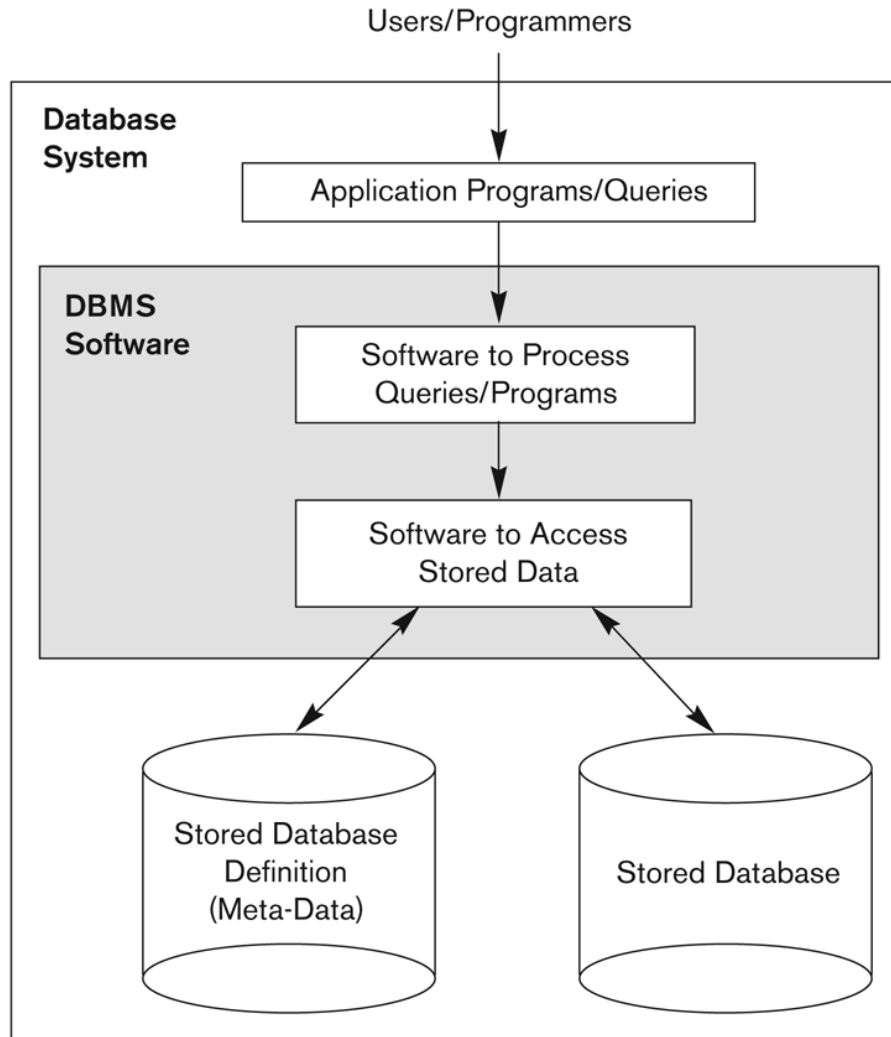
- **데이터 베이스:**
  - 관련 데이터의 모음.
  - 조직을 운영하고 관리하는 데 필요한 데이터 집합
- **데이터:**
  - 알려진 사실 기록될 수 있고 암묵적인 의미를 갖는 것.
- **미니월드:**
  - 일부의 현실 세계 데이터베이스에 어떤 데이터가 저장되어 있는지에 대한 정보. 예를 들어, 대학의 학생 성적과 성적 증명서.
- **데이터베이스 관리 시스템(DBMS):**
  - 컴퓨터화된 데이터베이스의 정의 및 조작(삽입, 업데이트, 삭제, 검색)을 용이하게 해주는 소프트웨어 패키지/시스템.
- **데이터베이스 시스템:**
  - 데이터 자체와 함께 DBMS 소프트웨어. 때로는 애플리케이션도 포함됩니다.
  - DBS = 데이터베이스 + DBMS + 응용 프로그램 + 사용자 + ...

# 데이터베이스 및 데이터베이스 기술의 영향

- **사업체:**
  - 은행, 보험, 소매, 운송, 의료, 제조
- **서비스 산업:**
  - 금융, 부동산, 법률, 전자상거래, 중소기업
- **교육 :**
  - 콘텐츠 및 전달을 위한 리소스
- **최근:**
  - 소셜 네트워크, 환경 및 과학적 응용, 의학 및 유전학
- **개인화된 애플리케이션:**
  - 스마트 모바일 기기 기반



# 단순화된 데이터베이스 시스템 환경



**Figure 1.1**  
A simplified database  
system environment.

# 일반적인 DBMS 기능

- 정의하다 특정 데이터베이스 데이터 유형, 구조 및 제약 조건 측면에서
- 건설하다 또는 초기 데이터베이스를 로드합니다. 데이터 내용 2차 저장 매체에
- 조작하다 데이터베이스 :
  - 검색: 쿼리, 보고서 생성
  - 수정: 콘텐츠의 삽입, 삭제 및 업데이트
  - 웹 애플리케이션을 통한 데이터베이스 액세스
- 처리 중 데이터 공유하기 동시 사용자 및 애플리케이션 프로그램 집합에 의해 - 그러나 모든 데이터는 유효하고 일관되게 유지됩니다.

예

# 데이터베이스의 예(개념적 데이터 모델 포함)

- 예를 들어 미니 월드:
  - 대학 환경의 일부.
- 어떤 미니월드 *엔/E/타*:
  - 학생들
  - 코스
  - 섹션(과정)
  - (학술) 학과
  - 강사



# 데이터베이스의 예(개념적 데이터 모델 포함)

- 어떤 미니월드 관계:

- 섹션 **특정하다** 행동
- 학생들 **가져간다** 섹션
- 코스 **필수조건이 있다** 행동
- 강사 **가르치다** 섹션
- 코스 **에 의해 제공됩니다** 부서
- 학생들 **전공하다** 부서

- 참고사항: 위의 엔티티 및 아르 자형 관계는 일반적으로 개념적으로 표현됩니다. 예이 데이터 모델, 예를 들면 엔티티-관계 나 P 데이터 모델 (3장, 4장 참조)

# 간단한 데이터베이스의 예

## 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

그림 1.2 학생 및 과정 정보를 저장하는 데이터베이스.

# 데이터베이스 접근 방식의 주 요 특징

# 데이터베이스 접근 방식의 주요 특징

- 자기 설명 비자연을 잉하다 데이터베이스 시스템의:

- DBMS **목록** 백화점 **특정** 데이터베이스의 설명 (예: 데이터 구조, 유형 및 제약 조건)
- 설명은 다음과 같습니다. **메타데이터\***.
- 이를 통해 DBMS 소프트웨어는 다양한 데이터베이스 애플리케이션과 함께 작동할 수 있습니다.

# 단순화된 데이터베이스 카탈로그의 예

## 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
Course_number	XXXXNNNN	COURSE
....	....	....
....	....	....
....	....	....
Prerequisite_number	XXXXNNNN	PREREQUISITE

Note: Major\_type is defined as an enumerated type with all known majors. XXXXNNNN is used to define a type with four alpha characters followed by four digits

**Figure 1.3**

An example of a database catalog for the database in Figure 1.2.

테이블(카탈로그)은 테이블을 설명합니다.

### 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

**그림 1.2**

# 데이터베이스 접근 방식의 주요 특징

- 프로그램간의 절연 에이<sub>ms</sub>와 데이터 :

- 라고 불리는 **프로그램-데이터 독립성**.
- 데이터 구조 및 저장 구성을 변경하지 않고도 가능  
DBMS 접속 프로그램을 변경해야 함.

Data Item Name	Starting Position in Record	Length in Characters (bytes)
Name	1	30
Student_number	31	4
Class	35	1
Major	36	4

그림 1.3의 데이터베이스 카탈로그를 기반으로 한 STUDENT 레코드의 내부 저장 형식입니다.

# 데이터베이스 접근 방식의 주요 특징

(계속되는)

- 데이터 추상화:

- 데이터 모델~에 사용된다 저장소 세부 정보 숨기기 사용자에게 다음을 제공합니다.  
데이터베이스의 개념적 관점 .
- 프로그램은 데이터 저장소가 아닌 데이터 모델 구조를 참조합니다.  
세부

# 데이터베이스 접근 방식의 주요 특징

(계속되는)

- 지원여러 번 보기 와데이터의 s :

- 각 사용자는데이터베이스의 다른 보기를 보세요 , 설명하는 **오직**해당 사용자에게 관심 있는 데이터.

**TRANSCRIPT**

Student_name	Student_transcript				
	Course_number	Grade	Semester	Year	Section_id
Smith	CS1310	C	Fall	08	119
	MATH2410	B	Fall	08	112
Brown	MATH2410	A	Fall	07	85
	CS1310	A	Fall	07	92
	CS3320	B	Spring	08	102
	CS3380	A	Fall	08	135

(a)

**COURSE\_PREREQUISITES**

Course_name	Course_number	Prerequisites
Database	CS3380	CS3320
		MATH2410
Data Structures	CS3320	CS1310

(b)

그림 1.2의 데이터베이스에서 파생된 두 가지 뷰. (a) TRANSCRIPT 뷰.  
(b) COURSE\_PREREQUISITES 보기.



# 데이터베이스 접근 방식의 주요 특징

(계속되는)

- 데이터 공유 그리고 다중 사용자 티거래 처리 중:

- 일련의 허용 **동시 사용자** 데이터베이스에서 검색하고 업데이트합니다.
- **동시성 제어** DBMS 내에서 각각을 보장합니다 **거래**  
올바르게 실행되거나 중단됩니다
- **회복** 서브시스템은 완료된 각 트랜잭션이 효과를 갖도록 보장합니다.  
데이터베이스에 영구적으로 기록됨
- **온라인**(온라인 거래 처리)는 데이터베이스의 주요 부분입니다.  
응용 프로그램. 이를 통해 수백 개의 동시 트랜잭션을 실행할 수 있습니다.  
초당.

# 현장의 배우들

# 데이터베이스 사용자

- 사용자는 다음과 같이 구분될 수 있습니다.
  - 그 사람들은 실제로 데이터베이스 콘텐츠를 사용하고 제어합니다. , 그리고 데이터베이스 애플리케이션을 설계, 개발 및 유지 관리하는 사람들("배우"라고 불리는 현장에서 ), 그리고
  - 그 사람들은 DBMS 소프트웨어를 설계하고 개발하다 및 관련 도구, 그리고 컴퓨터 시스템 운영자(라고 함) "뒤에 있는 노동자들 장면" ).

사용자  
↓  
데이터베이스 시스템과 관련된 사람들

# 데이터베이스 사용자

- 데이터베이스 관리자:

- 접근 권한 부여를 담당합니다 데이터베이스에 조정하다 그리고 모니터링 그것의 사용, 소프트웨어 및 하드웨어 리소스 획득,  
그 사용을 통제하고 운영 효율성 모니터링.

- 데이터베이스 설계자:

- 내용, 구조, 제약 조건, 기능 또는 거래를 정의하는 책임 데이터베이스에 대해. 그들은 소통해야 합니다.  
최종 사용자와 협력하여 그들의 요구 사항을 파악합니다.

# 데이터베이스 최종 사용자

- **최종 사용자:**

- 쿼리, 보고서에 데이터를 사용하고, 일부는 데이터베이스 콘텐츠를 업데이트합니다.

- **최종 사용자는 다음과 같이 분류될 수 있습니다.**

- **평상복:** 데이터베이스 접속가끔 필요할 때

- **순진한또는 매개변수적:** 이들은 최종 사용자 인구의 상당 부분을 차지합니다.

- 그들은 이전에 잘 정의된 함수를 다음과 같은 형태로 사용합니다. "통조림 거래" 데이터베이스에 대하여.

- 모바일 앱 사용자는 대부분 이 범주에 속합니다.

- 은행원 또는 예약 직원 전체 작업 교대 동안 이 활동을 수행하는 매개변수 사용자입니다.

- 소셜 미디어 사용자는 웹사이트에서 정보를 게시하고 읽습니다.

# 데이터베이스 최종 사용자(계속되는)

## - 정교한:

- 여기에는 다음이 포함됩니다. 비즈니스 분석가, 과학자, 엔지니어 , 다른 사람들은 철저히 시스템 기능에 익숙합니다.
- 많은 사람들이 소프트웨어 패키지 형태의 도구를 사용하여 긴밀하게 작동합니다.  
저장된 데이터베이스.

## - 독립형:

- 주로 개인 데이터베이스를 유지하다 바로 사용할 수 있는 포장을 사용하여 응용 프로그램.
- 예를 들어 세금 프로그램 사용자가 자체 내부를 생성하는 경우가 있습니다.  
데이터 베이스.
- 또 다른 예는 사용자입니다. 개인 사진의 데이터베이스를 유지 관리합니다.  
비디오.

# 데이터베이스 사용자

- 시스템 분석가 및 애플리케이션 개발자

- 이 범주는 현재 IT 인력의 매우 큰 비중을 차지하고 있습니다.

- 시스템 분석가:

- 그들은 naï의 사용자 요구 사항을 이해합니다 ve 및 정교한 사용자 및  
디자인 응용 프로그램 이러한 요구 사항을 충족하기 위해 통조림 거래도 포함됩니다.

- 애플리케이션 프로그래머:

- 분석가가 개발한 사양을 구현하고 테스트하고 디버깅합니다.  
배치 전.

- 비즈니스 분석가:

- 이러한 사람들이 점점 더 필요해지고 있습니다.방대한 양의 분석  
비즈니스 데이터와 실시간 데이터("빅 데이터") 더 나은 의사결정을 위해  
기획, 광고, 마케팅 등과 관련됨

# 현장 뒤의 근로자들



# 현장 뒤의 근로자들

- **시스템 설계자 및 구현자:**

- DBMS 패키지 설계 및 구현 모듈과 인터페이스 형태로 테스트하고 디버깅합니다.  
DBMS는 애플리케이션, 언어 컴파일러, 운영 체제 구성 요소 등과 인터페이스해야 합니다.

- **도구 개발자:**

- 소프트웨어 시스템을 설계하고 구현합니다.모델링 도구 그리고 데이터베이스를 설계하고,성능 모니터링, 프로토타입 제작, 테스트 데이터 생성, 사용자 인터페이스 생성, 시뮬레이션 등을 통해 애플리케이션 구축이 용이해지고 데이터베이스의 효과적인 활용이 가능해집니다.

- **운영자 및 유지 보수 인력:**

- 그들은 실제 실행을 관리하고데이터베이스 시스템 하드웨어 및 소프트웨어 환경 유지 관리.

# DBMS 사용의 장점 접근하다

# 데이터베이스 접근 방식을 사용하는 이점

- 중복성 제어(감소) 데이터 저장, 개발 및 유지 관리 작업에 있어서.
  - 여러 사용자 간 데이터 공유.

**GRADE\_REPORT**

Student_number	Student_name	Section_identifier	Course_number	Grade
17	Smith	112	MATH2410	B
17	Smith	119	CS1310	C
8	Brown	85	MATH2410	A
8	Brown	92	CS1310	A
8	Brown	102	CS3320	B
8	Brown	135	CS3380	A

(a)

**GRADE\_REPORT**

Student_number	Student_name	Section_identifier	Course_number	Grade
17	Brown	112	MATH2410	B

(b)

그림 1.6 GRADE\_REPORT에서 Student\_name과 Course\_name의 중복 저장. (a) 일관된 데이터.(b) 일관되지 않은 기록.

# 데이터베이스 접근 방식을 사용하는 이점

- 데이터 공유 여러 사용자로부터
- u를 제한하다 N데이터에 대한 승인된 액세스 DBA 직원만 권한이 있는 명령과 기능을 사용할 수 있습니다.
- 프로그램 객체에 대한 영구 저장소 제공
  - 예를 들어, 객체 지향 DBMS는 프로그램 객체를 영구적으로 만듭니다. 12장을 참조하세요.
- 저장 구조(예: 인덱스) 제공효율적인 쿼리 처리 - 17장을 참조하세요.

# 데이터베이스 접근 방식을 사용하는 이점

(계속되는)

- 제공최적화 N쿼리의 효율적인 처리를 위해.
- 제공백업하다 디회복 서비스.
- 제공다중 인터페이스 기음es는 다르다 기음사용자의 아가씨들 .
- 복잡한 r을 표현하다 이자형관계 데이터 중에서.
- 시행하다 정직성 c 영형제약 데이터베이스에.
- 그림추론 에이nd 행동 저장된 데이터에서 사용 연역적 규칙과 능동적 규칙 에스그리고 트리거 .

# 데이터베이스 접근 방식 사용의 추가 적 의미

- **표준 시행의 잠재력:**

- 이는 대규모 조직의 데이터베이스 애플리케이션 성공에 매우 중요합니다.**표준** 데이터 항목 이름, 표시 형식을 참조하세요.  
화면, 보고서 구조, 메타데이터(데이터 설명), 웹 페이지 레이아웃 등

- **애플리케이션 개발 시간 단축:**

- 새로운 애플리케이션을 추가하는 데 걸리는 시간이 줄어듭니다.

# 추가 의미

## 데이터베이스 접근 방식 사용(계속)

- 데이터 변경에 대한 유연성 구조:

- 새로운 요구 사항이 정의됨에 따라 데이터베이스 구조가 진화할 수 있습니다.

- **최신 정보의 가용성:**

- 쇼핑, 항공, 호텔, 자동차 예약 등 온라인 거래 시스템에 매우 중요합니다.

- **규모의 경제성:**

- 자원과 인력의 낭비적인 중복을 피할 수 있습니다.

부서 간 데이터와 애플리케이션을 통합합니다.

# **데이터베이스 애플리케이션의 간략한 역사**



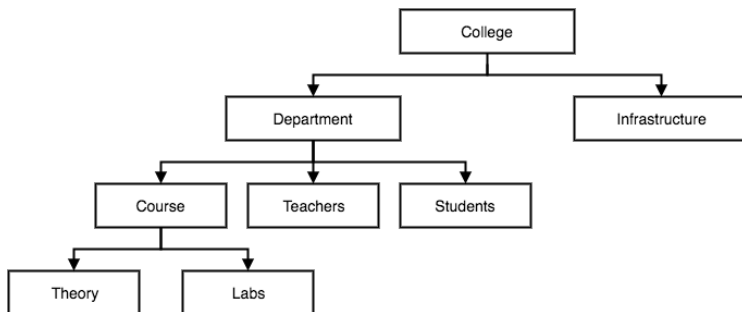
# 데이터베이스 기술의 역사적 발전

## • 초기 데이터베이스 응용 프로그램:

단단히 결합됨

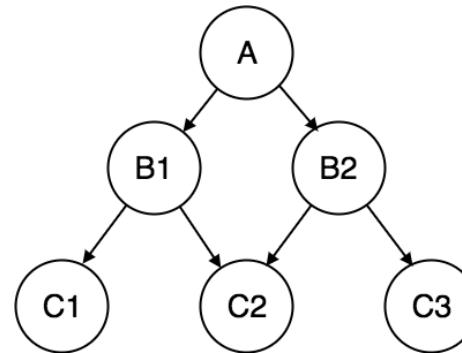
개념 설계 - 물리적 설계

- 그계층적 모델과 네트워크 모델은 1960년대 중반에 도입되었습니다. 70년대에 지배적인 위치를 차지했습니다.
- 전 세계 데이터베이스 처리의 대부분은 여전히 다음을 사용하여 수행됩니다.  
모델, 특히 IBM의 IMS 시스템을 사용한 계층적 모델입니다.



계층적 모델

- 상향식 구조
- 일대일을 지원하며
- 일대다 관계



네트워크 모델

- 일대일 지원
- 일대다
- 그리고 다대다 관계
- 문제는 임시 쿼리를 지원할 수 없다는 것입니다.

찰스 윌리엄 바흐만III(1924년 12월 11일 ~ 2017년 7월 13일)은 미국의 컴퓨터 과학자였다.



# 데이터베이스 기술의 역사적 발전

- 관계형 모델 기반 시스템:

독립

개념 설계 - 물리적 설계

- 관계형 모델은 원래 1970년에 도입되었으며 IBM Research와 여러 기관에서 심도 있게 연구되고 실험되었습니다.

대학.

- 관계형 DBMS 제품은 1980년대 초반에 등장했습니다.

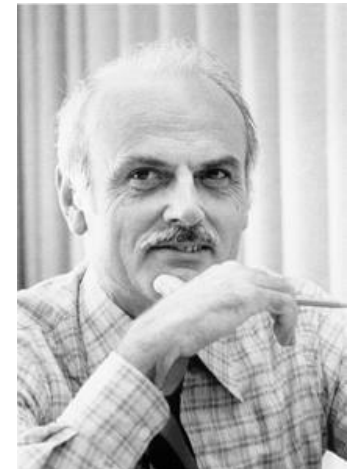
Relational Model

Activity Code	Activity Name
23	Patching
24	Overlay
25	Crack Sealing

Key = 24

Activity Code	Date	Route No.
24	01/12/01	I-95
24	02/08/01	I-66

Date	Activity Code	Route No.
01/12/01	24	I-95
01/15/01	23	I-495
02/08/01	24	I-66



에드거 프랭크 "테드" 코드(1923년 8월 19일 ~ 2003년 4월 18일)은 영국의 컴퓨터 과학자였습니다. IBM에서 근무하는 동안 데이터베이스 관리를 위한 관계형 모델을 발명했습니다. , 관계형 데이터베이스와 관계형 데이터베이스 관리 시스템의 이론적 기초입니다.

# 데이터베이스 기술의 역사적 발전

(계속되는)

- 객체 지향 및 새로운 응용 프로그램:

- 객체 지향 데이터베이스 관리 시스템(OODBMS)은 1980년대 후반과 1990년대 초반에 다음과 같은 요구 사항을 충족하기 위해 도입되었습니다.

CAD 및 기타 응용프로그램에서의 복잡한 데이터 처리.

- 그 사용은 크게 늘어나지 않았습니다.

- 많은 관계형 DBMS는 객체 데이터베이스 개념을 통합했습니다.

새로운 카테고리로 이어짐 *객체 관계* DBMS(ORDBMS)

- *확장된 관계형*시스템은 추가 기능을 추가합니다(예: 멀티미디어용)  
데이터, 텍스트, XML 및 기타 데이터 유형)

# 데이터베이스 기술의 역사적 발전

(계속되는)

- 웹 및 전자 상거래 애플리케이션의 데이터:

- 웹은 페이지 간의 링크가 포함된 HTML(하이퍼텍스트 마크업 언어)로 데이터를 포함합니다.
- 이로 인해 새로운 일련의 응용 프로그램이 생겨났으며 전자 상거래가 생겨났습니다.  
XML(eXtended Markup Language)과 같은 새로운 표준을 사용합니다. (Ch. 참조) 13).
- PHP 및 JavaScript와 같은 스크립트 프로그래밍 언어를 사용하면 부분적으로 생성된 동적 웹 페이지를 생성할 수 있습니다.

데이터베이스(11장 참조)

- 웹 페이지를 통한 데이터베이스 업데이트도 허용합니다.

# 데이터베이스 기능 확장(1)

- 다음 영역의 DBMS에 새로운 기능이 추가되고 있습니다.
  - 과학적 응용 - 물리학, 화학, 생물학 - 유전학
  - 지구 및 대기 과학 및 천문학
  - **엑스엠L**(확장형 마크업 언어)
  - 이미지 저장 및 관리
  - 오디오 및 비디오 데이터 관리
  - 데이터 웨어하우징 및 데이터 마이닝 - 새로운 기술을 사용한 미래 개발을 위한 매우 중요한 분야(28-29장 참조)
  - 공간 데이터 관리 및 위치 기반 서비스
  - 시계열 및 과거 데이터 관리
- 위의 내용은 다음과 같습니다. **새로운 연구 및 개발** 데이터베이스 시스템에 새로운 데이터 유형, 복잡한 데이터 구조, 새로운 작업, 저장 및 인덱싱 체계를 통합하는 것.

# 데이터베이스 기능 확장(2)

- 21세기의 배경성세기:

- 21세기의 첫 10년성세기에는 사용자 생성 데이터와 애플리케이션에서 자동으로 수집된 데이터가 엄청나게 증가했습니다.  
검색 엔진.
- Facebook, Twitter와 같은 소셜 미디어 플랫폼 생성 중입니다  
하루에 수백만 건의 거래가 이루어지고 있으며 기업들은 이를 활용하고자 합니다.  
이 데이터를 사용하여 사용자를 "이해"합니다.
- 클라우드 스토리지 및 백업 무제한 저장공간을 만들고 있습니다  
사용자와 애플리케이션에서 사용 가능

# 데이터베이스 기능 확장(3)

- 빅데이터 기술의 등장과 **한국어:데이터베이스**

- 일부 애플리케이션에서는 하루에 페타바이트( $10^{15}$ 바이트 또는 1000테라바이트) 규모의 데이터 폭증을 처리하기 위해 새로운 데이터 저장, 관리 및 분석 기술이 필요했습니다. 이를 일반적으로 "빅 데이터"라고 부르기 시작했습니다.
- Hadoop(Yahoo에서 유래)과 MapReduce 분산 데이터 처리에 대한 프로그래밍 방식(Google에서 유래)과 Google 파일 시스템은 빅 데이터 기술을 탄생시켰습니다 (25장). Spark 기반 기술의 형태로 추가적인 향상이 이루어지고 있습니다.
- NOSQL(Not Only SQL- SQL은 관계형 DBMS의 사실상 표준 언어임) 시스템은 문서에서의 빠른 검색 및 수집, 소셜 네트워크에서 발생하는 거대한 그래프의 처리, 유연한 트랜잭션 처리 모델을 갖춘 기타 형태의 비정형 데이터를 위해 설계되었습니다 (24장).

**DBMS를 사용하지 말아야 할 때**



# DBMS를 사용하지 않는 경우

- **DBMS 사용의 주요 저해 요소(비용):**
  - 초기 투자 비용이 높고 추가 하드웨어가 필요할 수 있음.
  - 일반성, 보안, 동시성 제어, 복구 및 무결성 기능을 제공하기 위한 오버헤드.
- **DBMS가 불필요할 수 있는 경우:**
  - 데이터베이스와 애플리케이션이 간단하고, 잘 정의되어 있으며, 변경될 가능성이 없는 경우.
  - 여러 사용자가 데이터에 액세스할 필요가 없는 경우.
- **DBMS가 실행 불가능한 경우:**
  - 일반 용도 DBMS가 사용 가능한 저장소에 맞지 않는 임베디드 시스템에서

# DBMS를 사용하지 않는 경우

- 어떤 DBMS도 충분하지 않은 경우:
  - DBMS 오버헤드로 인해 충족되지 못할 수 있는 엄격한 실시간 요구 사항이 있는 경우(예: 전화 교환 시스템)
  - 데이터베이스 시스템이 데이터의 복잡성을 처리할 수 없는 경우  
모델링 제한으로 인해(예: 복잡한 계층 및 단백질의 경우)  
데이터베이스)
  - 데이터베이스 사용자에게 DBMS에서 지원하지 않는 특수 작업(예: GIS 및 위치 기반 서비스)이 필요한 경우

# 장 요약

- 데이터베이스 유형 및 데이터베이스 애플리케이션
- 기본 정의
- 일반적인 DBMS 기능
- 데이터베이스의 예 (대학)
- 데이터베이스 접근 방식의 주요 특징
- 데이터베이스 사용자 유형
- 데이터베이스 접근 방식을 사용하는 이점
- 데이터베이스 기술의 역사적 발전
- 데이터베이스 기능 확장
- 데이터베이스를 사용하지 말아야 할 때