

**CSE231**

**데이터베이스 설계  
(Database Design)**

**Lecture 02: 데이터베이스 시스템**

**담당교수: 전강욱(컴퓨터공학부)**

**kw.chon@koreatech.ac.kr**

# NCS 정보

- 능력 단위명 : 개념 데이터 모델링
- 능력 단위요소 : 핵심개체 정의하기
- 학습목표(수행 준거) :
  - 2.2 후보 개체에서 주제영역을 대표할 수 있는 행위의 주체 또는 대상이 될 수 있는 독립적인 기본 개체를 정의할 수 있다.

# NCS 정보(계속)

## ■ 지식

- 개념적 데이터 모델과 논리적 데이터 모델
- ER 표기법
- 속성 종류별 특성

## ■ 기술

- 핵심개체를 식별하는 능력
- ER 다이어그램 작성 능력

## ■ 태도

- ER 표기법에 맞게 표현하려는 자세
- ER 다이어그램을 재검토하는 습관

# 개요

- 데이터 모델
- 데이터베이스 시스템
  - 3단계-스키마 아키텍처와 데이터 독립성
  - 시스템 데이터베이스
  - 데이터베이스 사용자
  - DBMS
- 강의요약

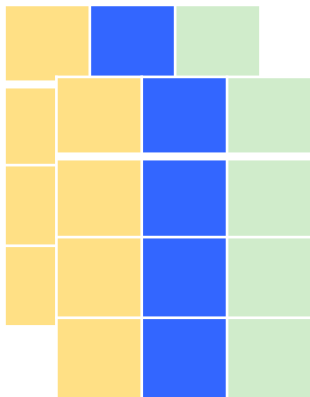
# 데이터 모델

- 실세계 데이터를 저장 및 관리할 때, **데이터베이스 구조를 명시하기 위해 사용할 수 있는 개념들의 집합**
  - 데이터베이스의 저장구조와 연산을 포함

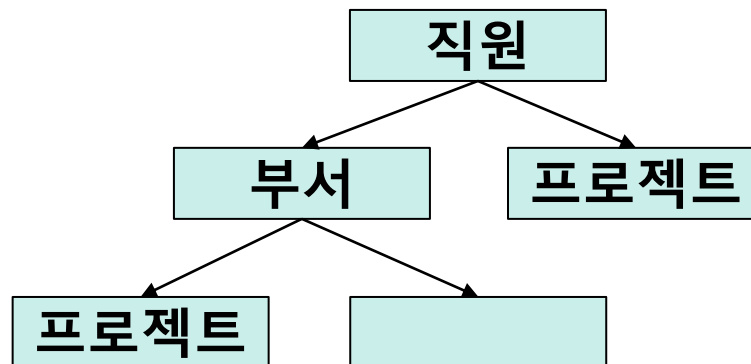
## 실세계정보

직원(사번, 이름, 전화번호, 부서,...)  
부서(부서명, 대표전화, 소속직원, 사무실위치,...)  
프로젝트(프로젝트명, 규모, 참여직원,...)

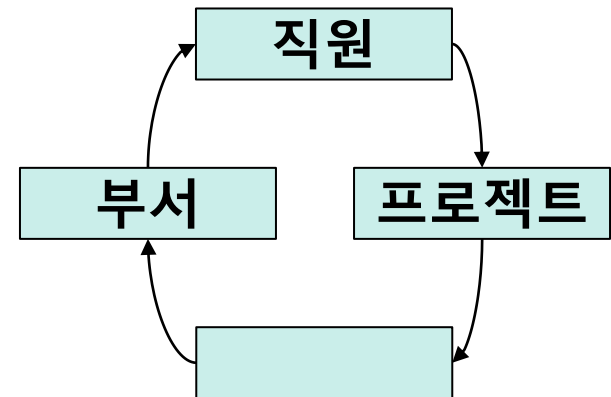
관계형모델



계층형모델



망형모델



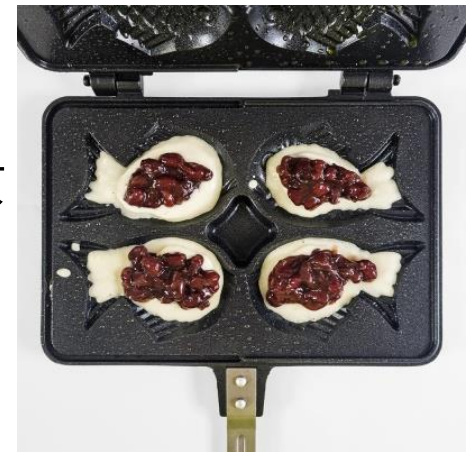
# 스키마 및 인스턴스

## ■ 데이터베이스 스키마 (또는 메타데이터)

- 데이터베이스에 저장되는 데이터 구조와 제약조건
- 데이터베이스 설계 과정에서 명시하며, 한번 정의되면 잘 바뀌지 않음
  - 최근에 big data 시기가 도래하면서, 자주 바뀌는 경향이 있음
- 예: 고객(고객번호 INTEGER, 이름 VARCHAR(10), 나이 INTEGER)

## ■ 인스턴스

- 스키마에 따라 데이터베이스에 실제로 저장된 값
- 인스턴스는 계속 변함
- 예: 고객1(0705, 전강욱, 25)

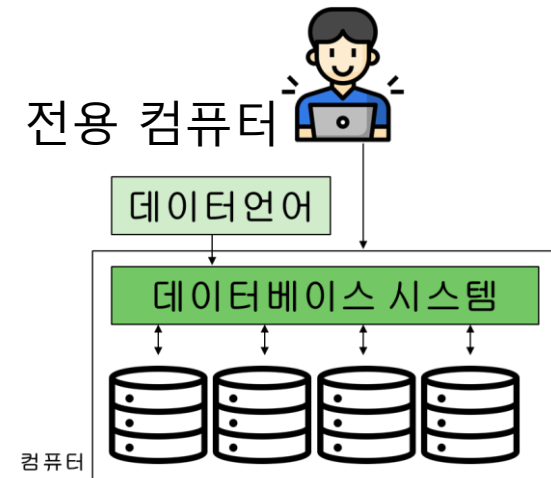


출처:  
<https://hottracks.kyobobook.co.kr/ht/gift/detail/2310046784610>.

# 데이터베이스 시스템의 정의

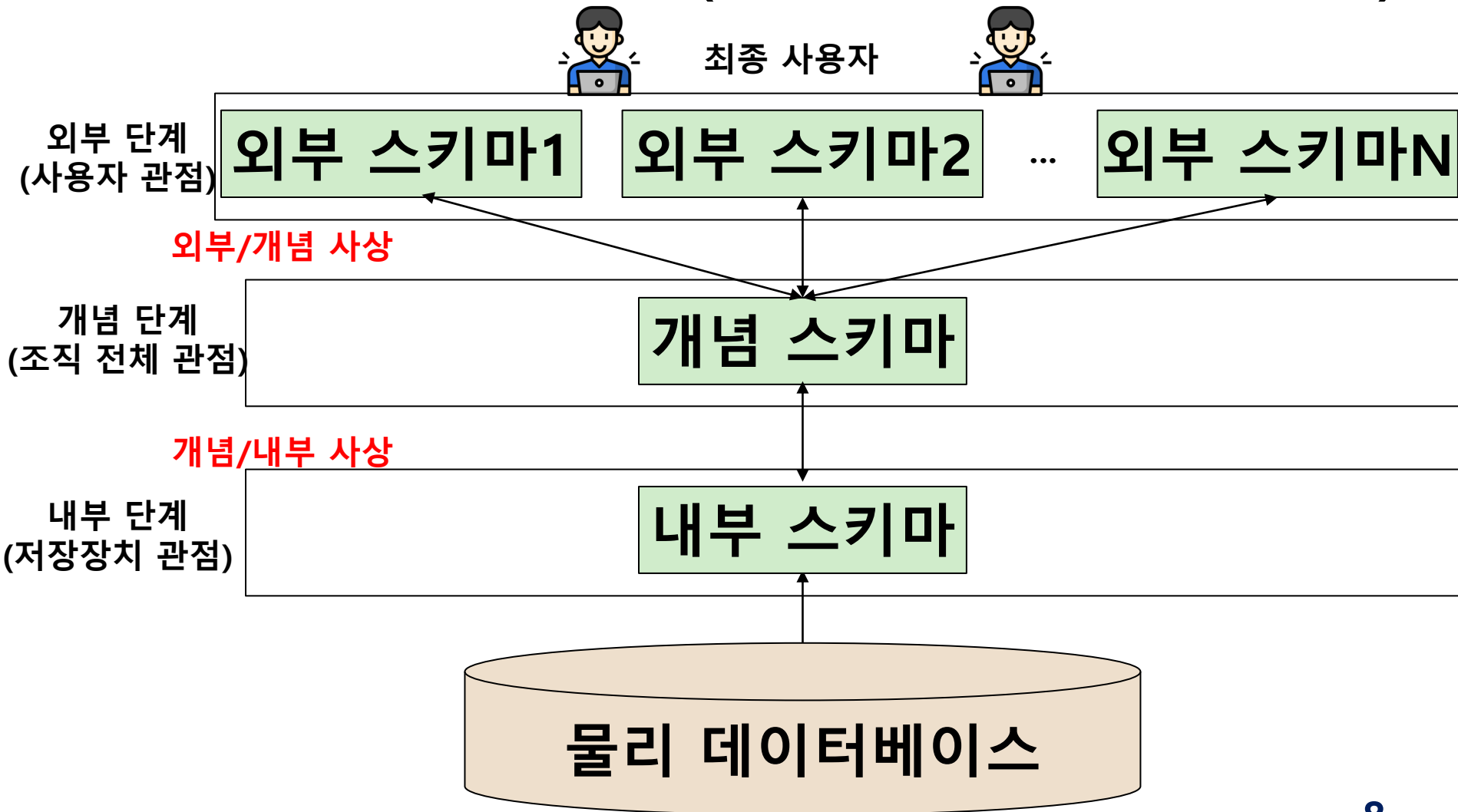
## ■ 데이터베이스 시스템

- 데이터베이스에 데이터를 저장하고, 데이터를 관리하여 조직에 필요한 정보를 생성해주는 시스템
- 구성요소
  - 데이터베이스(DB): 데이터를 저장
  - 데이터베이스 관리 시스템(DBMS): DB를 생성, 관리, 조작함으로써 사용자와 DB를 연결해주는 S/W
  - 데이터언어 (DDL, SQL 등): DB 정의와 조작, 제어를 위한 DB 전용 언어
  - DB 사용자: DB에 접근하는 사람
    - 일반 사용자, 응용 프로그래머, DB 관리자 등
  - DB 컴퓨터: DB에 대한 연산을 전담하는 DB 관리 전용 컴퓨터



# 3단계-스키마 아키텍처의 도식화

## ■ 3단계-스키마 아키텍처(three-schema architecture)





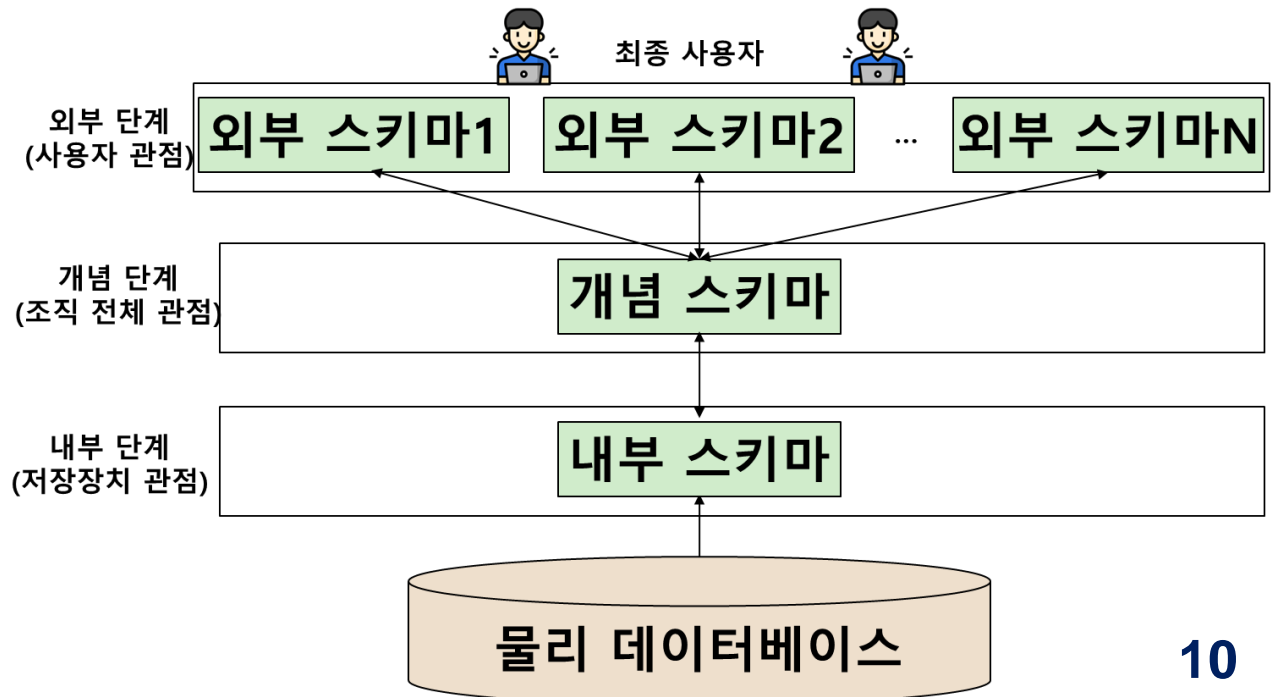
# 3단계-스키마 아키텍처

- ANSI/SPARC에서 제안
- 데이터베이스를 쉽게 이해하고 활용할 수 있도록 하나의 데이터베이스를 관점에 따라 세 단계로 나눈 것
  - 외부 단계 (external level) 또는 뷰 단계 (view level): 개별 사용자 관점
  - 개념 단계 (conceptual level): 조직 전체의 관점
  - 내부 단계 (internal level): 물리적인 저장 장치 관점
- 단계별로 다른 추상화 레벨 제공

# 3단계-스키마 아키텍처 (계속)

## ■ 외부 단계 또는 뷰 단계

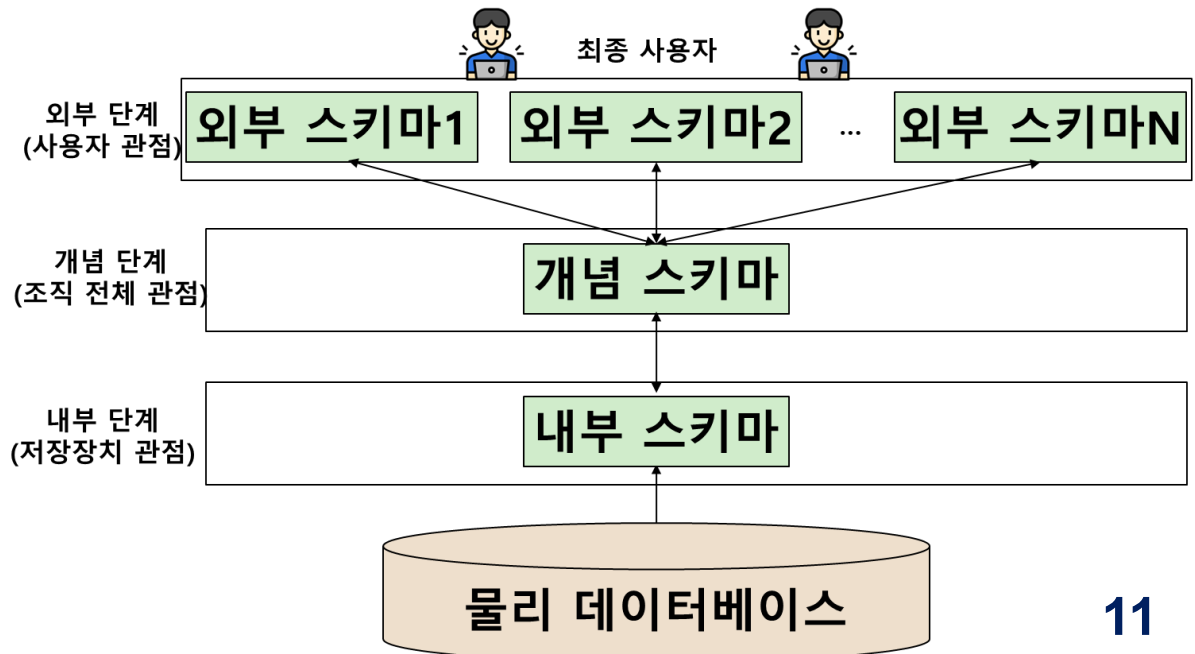
- 개별 사용자 관점에서 이해 및 표현
- 데이터베이스 하나에 외부 스키마가 여러 개 존재 가능
  - 외부 스키마(external schema)
  - 외부 단계에서 사용자에게 필요한 데이터베이스를 정의
  - 논리적 구조로 사용자마다 다름
  - 예: 학교정보시스템의 DB에서 학생 관점, 교직원 관점



# 3단계-스키마 아키텍처 (계속)

## ■ 개념 단계

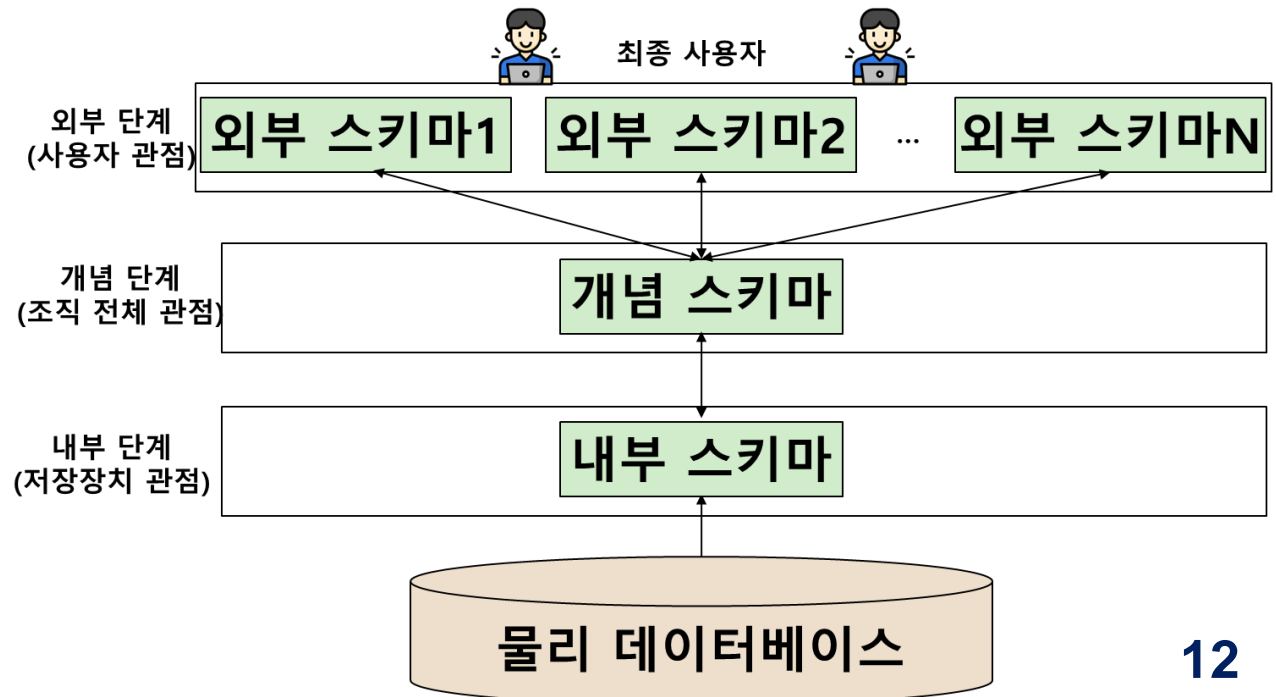
- 데이터베이스를 조직 전체의 관점에서 이해 및 표현
- 데이터베이스 하나에 개념 스키마(conceptual schema) 하나만 존재
  - 개념 단계에서 데이터베이스 전체의 논리적 구조를 정의
  - 조직 전체의 관점에서 생각하는 데이터베이스의 모습
  - 개체 관계, 데이터 타입, 관계, 제약조건 등을 나타내는데 중점
- 물리적 저장 구조의 세부 구조는 고려하지 않음



# 3단계-스키마 아키텍처 (계속)

## ■ 3단계 데이터베이스 구조: 내부 단계

- 데이터베이스를 저장장치 관점에서 이해 및 표현
- 데이터베이스 하나에 내부 스키마(internal schema) 하나만 존재
  - 전체 데이터베이스가 저장 장치에 실제로 저장되는 방법을 정의
  - 레코드 구조, 필드 크기, 레코드 접근 경로 등의 물리적인 저장 구조



# 3단계-스키마 아키텍처 (계속)

## ■ 3단계 데이터베이스 구조 예제

외부 단계  
(사용자 관점)

고객분석팀 외부 스키마			상품배송팀 외부 스키마			
성별 CHAR(2)	나이 INT	직업 CHAR(10)	고객번호 INT	고객이름 CHAR(10)	주소 CHAR(20)	연락처 CHAR(20)

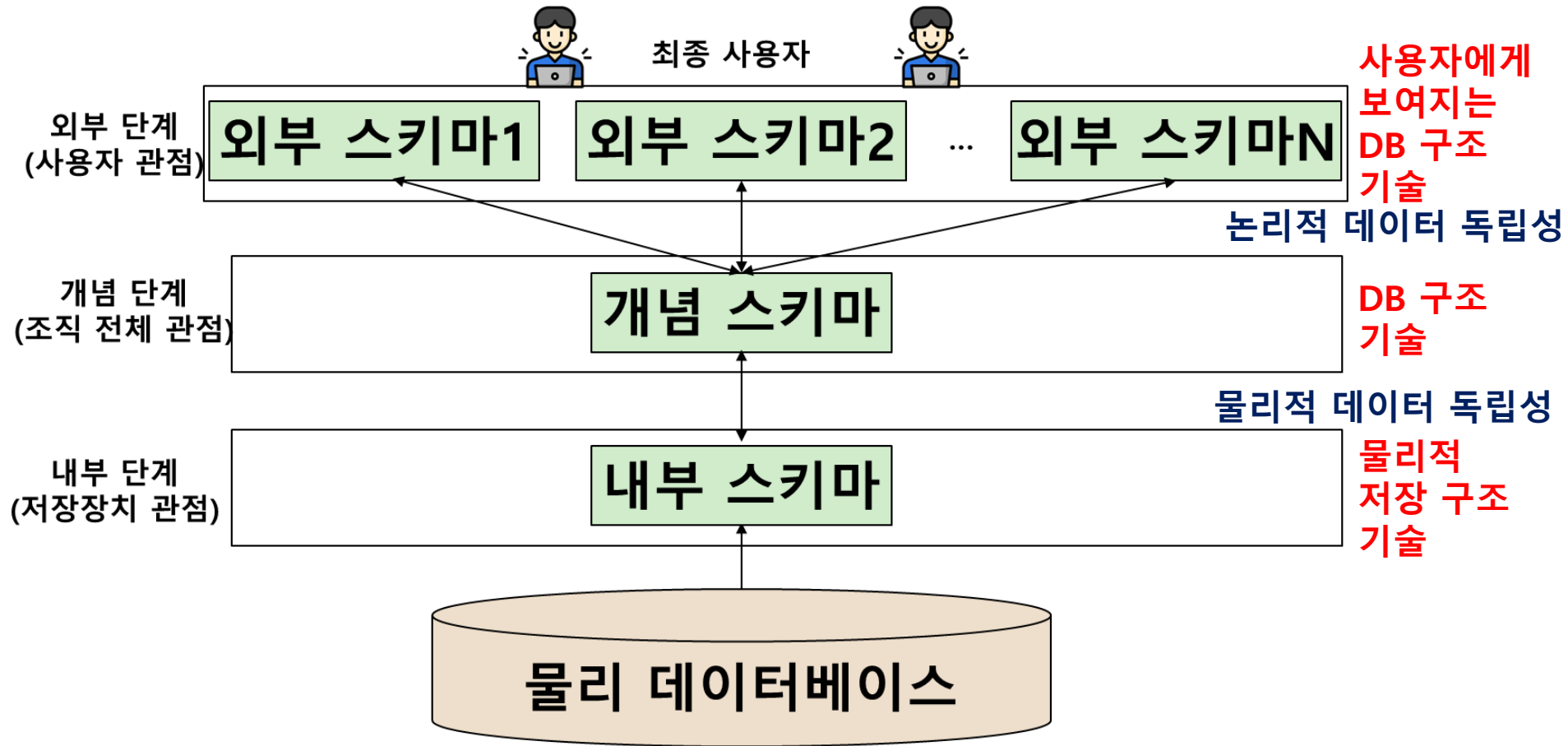
개념 단계  
(조직 전체 관점)

개념스키마						
번호 INT	이름 CHAR(10)	성별 CHAR(2)	나이 INT	직업 CHAR(10)	주소 CHAR(20)	연락처 CHAR(20)

내부 단계  
(저장장치 관점)

내부스키마(레코드 길이: 70Byte)			
필드이름	필드 크기	OFFSET	인덱스여부
번호	4Byte	0	존재
이름	10Byte	4	
성별	2Byte	14	
나이	4Byte	16	
직업	10Byte	20	
주소	20Byte	30	
연락처	20Byte	50	존재

# 3단계-스키마 아키텍처 (계속)



# 데이터 독립성

- 하위 스키마가 변경되어도 상위 스키마는 영향을 받지 않음
  - 논리적 데이터 독립성 (logical data independence)
    - 개념 스키마가 변경되어도 외부 스키마는 영향을 받지 않음
    - 개념 스키마가 변경되면 관련된 외부/개념 사상의 수정이 필요
  - 물리적 데이터 독립성 (physical data independence)
    - 내부 스키마가 변경되어도 개념 스키마는 영향을 받지 않음
    - 내부 스키마가 변경되면 관련된 개념/내부 사상의 수정이 필요

# 3단계-스키마 아키텍처 장/단점

## ■ 장점

- 물리적 및 논리적 독립성 확보가 용이함
- 사상(mapping) 변경만을 통해, 각 단계의 독립성이 유지

## ■ 단점

- 두 번의 사상에 따른 DBMS 비효율성 야기
  - 질의/프로그램 컴파일/실행에 많은 오버헤드



# 시스템 데이터베이스

- 데이터베이스는 실제 데이터 외에 저장된 데이터를 관리하기 위한 부가 정보도 포함
  - 메타 데이터(meta data): 데이터를 위한 데이터
- 사용자 데이터베이스(user database)
  - 사용자가 실제로 이용하는 데이터가 저장되어 있는 일반 데이터베이스
- 시스템 데이터베이스(system database)
  - DBMS를 사용하기 위한 부가 정보가 저장된 데이터베이스

# 시스템 데이터베이스 (계속)

## ■ 데이터 사전 (data dictionary)

- 메타 데이터를 유지하는 시스템 DB
- 스키마, 사상 정보, 제약조건 등을 저장
- DBMS가 자동으로 생성 및 유지
- 일반 사용자 접근 가능하지만, 수정 불가능
- a.k.a., 시스템 카탈로그

## ■ 데이터 디렉토리 (data directory)

- 데이터 사전에 있는 데이터에 실제로 접근하는데 필요한 위치 정보 저장하는 시스템 데이터베이스
- 일반 사용자 접근 불가

# 데이터베이스 사용자

- 데이터베이스를 이용하기 위해 접근하는 모든 사람
- 이용목적에 따라 분류
  - 데이터베이스 관리자
  - 최종 사용자(일반 사용자)
  - 응용 프로그래머

# 데이터베이스 사용자 (계속)

## ■ 데이터베이스 관리자 (Database Administrator, DBA)

- 데이터베이스 시스템을 운영 및 관리하는 사람
- 주로 데이터 정의어와 데이터 제어어를 사용
- 주요 업무
  - 데이터베이스 구성 요소 선정
  - 데이터베이스 스키마 정의
  - 물리적 저장 구조와 접근 방법의 결정
  - 무결성 유지를 위한 제약조건 정의
  - 보안 및 접근권한 정책 결정
  - 백업 및 회복 기법 정의
  - 시스템 데이터베이스 관리
  - 시스템 성능 감시 및 성능 분석
  - 데이터베이스 재구성

# 데이터베이스 사용자 (계속)

- **최종 사용자(end user)**
  - 데이터베이스에 접근하여 데이터를 조작하는 사람
  - 주로 데이터 조작어를 사용
- **응용 프로그래머(application programmer)**
  - 데이터 언어를 삽입하여 응용 프로그램을 작성하는 사람
  - 주로 데이터 조작어를 사용

# 데이터언어

## ■ 사용 목적에 따라 분류

- 데이터 정의어 (DDL: Data Definition Language)
  - 스키마 정의, 수정, 삭제
- 데이터 조작어 (DML: Data Manipulation Language)
  - 데이터의 삽입, 삭제, 수정, 검색
- 데이터 제어어 (DCL: Data Control Language)
  - 내부적으로 필요한 규칙, 기법 정의

# 데이터언어 (계속)

- 데이터 정의어(Data Definition Language, DDL)
  - 개요
    - 새로운 데이터베이스를 구축하기 위해 스키마 정의
    - 기존 스키마의 정의를 수정 또는 삭제
  - 동작방식
    - 데이터 정의어로 명시된 문장이 입력되면 DBMS는 스키마에 대한 명세를 시스템 카탈로그 또는 데이터 사전에 저장
  - 기본기능과 예제
    - 데이터 모델에서 지원하는 데이터 구조를 생성 (e.g., CREATE TABLE)
    - 데이터 구조의 변경 (e.g., ALTER TABLE)
    - 데이터 구조의 삭제 (e.g., DROP TABLE)
    - 효율적인 데이터 접근을 위한 인덱스 정의 (e.g., CREATE INDEX)

# 데이터언어 (계속)

- 데이터 조작어 (Data Manipulation Language, DML)
  - 개요
    - 데이터의 삽입, 삭제, 수정, 검색 등을 위해 사용
  - 절차적 데이터 조작어(procedural DML)
    - 데이터를 얻기 위한 절차를 기술
  - 비절차적 데이터 조작어(nonprocedural DML)
    - 원하는 데이터만 명시하며 대표적인 예로 SQL 언어
    - a.k.a., 선언적 언어(declarative language)



# 데이터언어 (계속)

- 데이터 제어어 (Data Control Language, DCL)
  - 개요
    - 데이터베이스에 저장된 데이터를 여러 사용자가 무결성과 일관성을 유지하며 문제 없이 공유할 수 있도록, 내부적으로 필요한 규칙이나 기법을 정의
  - 사용 목적
    - 무결성: 정확하고 유효한 데이터만 유지
    - 보안: 허가 받지 않은 사용자의 데이터 접근 차단, 허가된 사용자에게 권한 부여
    - 회복: 장애가 발생해도 데이터 일관성 유지
    - 동시성 제어: 동시 공유 지원

# 데이터베이스 관리 시스템 (DBMS)

- 데이터베이스 관리와 사용자의 데이터 처리 요구 수행
- 주요 구성 요소
  - 질의 처리기(query processor)
    - 사용자의 데이터 처리 요구를 해석 및 처리
      - DDL 컴파일러, DML 프리 컴파일러, DML 컴파일러, 런타임 데이터 베이스 처리기, 트랜잭션 관리자 등을 포함
    - 저장 데이터 관리자(stored data manager)
      - 사용자 데이터베이스와 데이터 사전을 관리
      - 데이터베이스에 실제로 접근하는 역할 담당

# 데이터베이스 관리 시스템 (계속)



DDL 컴파일러

런타임 데이터베이스 처리기

DML 프리컴파일러

런타임 데이터베이스 처리기

트랜잭션 관리자

질의처리기

저장 데이터 관리자

DBMS

데이터베이스

# 데이터베이스 관리 시스템 (계속)

## ■ 질의처리기

### □ DDL 컴파일러

- 데이터 정의어로 작성된 스키마의 정의를 해석
- 저장 데이터 관리자의 도움을 받아 새로운 데이터베이스 구축, 스키마의 정의를 데이터 사전에 저장, 기존 스키마의 삭제나 수정 요청도 처리하여 변경된 내용을 데이터 사전에 적용

### □ DML 프리 컴파일러

- 응용 프로그램에 삽입된 데이터 조작어를 추출하여 DML 컴파일러에 전달

### □ DML 컴파일러

- 데이터 조작어로 작성된 데이터의 처리 요구를 분석하여 런타임 데이터베이스 처리기가 이해할 수 있도록 해석

### □ 런타임 데이터베이스 처리기

- 저장 데이터 관리자를 통해 데이터베이스에 접근
- DML 컴파일러로부터 받은 데이터 처리 요구를 데이터베이스에서 실제로 실행

### □ 트랜잭션 관리자

- 데이터베이스에 접근하는 과정에서 사용자의 접근 권한의 유효성 검사, 제약조건 위반 여부 확인, 회복, 병행과 관련한 작업 수행

# 데이터베이스 관리 시스템 (계속)

## ■ 질의처리기

- DDL 컴파일러
  - 데이터 정의어로 작성된 스키마의 정의를 해석
  - 저장 데이터 관리자의 도움을 받아 새로운 데이터베이스 구축, 스키마의 정의를 데이터 사전에 저장, 기존 스키마의 삭제나 수정 요청도 처리하여 변경된 내용을 데이터 사전에 적용
- DML 프리 컴파일러
  - 응용 프로그램에 삽입된 데이터 조작어를 추출하여 DML 컴파일러에 전달
- DML 컴파일러
  - 데이터 조작어로 작성된 데이터의 처리 요구를 분석하여 런타임 데이터베이스 처리기가 이해할 수 있도록 해석
- 런타임 데이터베이스 처리기
  - 저장 데이터 관리자를 통해 데이터베이스에 접근
  - DML 컴파일러로부터 받은 데이터 처리 요구를 데이터베이스에서 실제로 실행
- 트랜잭션 관리자
  - 데이터베이스에 접근하는 과정에서 사용자의 접근 권한의 유효성 검사, 제약조건 위반 여부 확인, 회복, 병행과 관련한 작업 수행

# 데이터베이스 유틸리티

## ■ 데이터베이스 관리자의 시스템 운영을 도와주는 S/W

### □ 적재 (loading)

- 데이터 파일을 자동적으로 데이터베이스 내 파일의 형식으로 변환해서 저장
- 예: 엑셀의 텍스트 파일을 데이터베이스 내 테이블로 적재

### □ 백업 (backup)

- 전체 데이터베이스를 디스크에 복사하여 백업 사본을 만듦

### □ 파일 재조직 (file reorganization)

- 성능 향상을 위해 데이터베이스 파일 구조를 변경

### □ 성능 모니터링 (performance monitoring)

- 데이터베이스의 사용을 모니터해서 사용 통계를 데이터베이스 관리자에게 제공
  - 성능을 향상시키기 위해서 화일들을 재조직할 것인지를 결정하는데 사용

# DBMS 아키텍처

## ■ 중앙 집중식 데이터베이스 시스템

- 하나의 컴퓨터 시스템에서 DBMS가 운영
  - 모든 작업은 메인 프레임에서 수행
  - 데이터를 입력하고 필요한 정보를 출력하여 표시해주는 일은 터미널이 수행
- 초기에 주로 사용되던 DBMS 아키텍처

## ■ 분산 데이터베이스 시스템

- 네트워크로 연결된 여러 사이트에 DB 자체가 분산
- 데이터베이스 시스템도 여러 컴퓨터 시스템에서 운영
- 사용자는 다른 사이트에 저장된 DB도 접근 가능

# DBMS 아키텍처 (계속)

- 클라이언트-서버 데이터베이스 시스템 (client-server database system)
  - 자체 컴퓨팅 능력을 가진 클라이언트(PC)를 통해 DB 서버를 접근
  - DB가 하나의 데이터베이스 서버에 저장
  - 데이터베이스 시스템의 기능이 서버와 클라이언트에 분산
    - 서버는 DB를 저장, DBMS를 운영하면서 여러 클라이언트들의 질의를 최적화, 권한 검사, 동시성 제어와 회복 기능을 수행, DB의 무결성 유지, 데이터베이스 접근을 관리
    - 클라이언트는 사용자 인터페이스를 관리하고 응용 프로그램들을 수행



# 요약

- 데이터 모델: 저장구조 + 데이터연산
- 3단계 스키마: 외부 스키마, 개념 스키마, 내부 스키마
  - 데이터 독립성 제공 가능
- 데이터언어: DDL, DML, DCL
- 시스템 데이터베이스: DBMS를 사용하기 위한 부가 정보가 저장된 데이터베이스
- 데이터베이스 사용자: 데이터베이스 관리자, 최종 사용자(일반 사용자), 응용 프로그래머
- DBMS 구조 및 DBMS 아키텍처

# 감사합니다!

담당교수: 전강욱(컴퓨터공학부)

[kw.chon@koreatech.ac.kr](mailto:kw.chon@koreatech.ac.kr)

