

# 테이타베이스

## Chap 8. 데이터 모델링



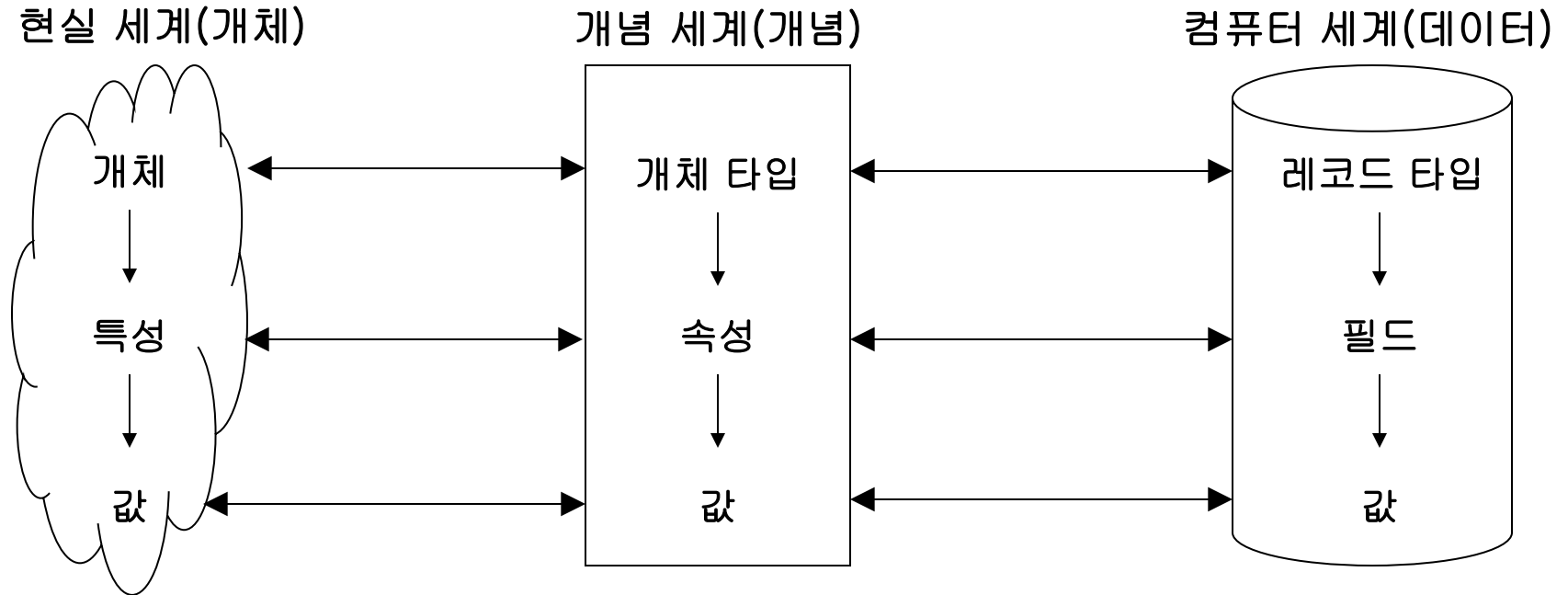
2014.05.12.

오 병 우

컴퓨터공학과

# Computerization

## 3개의 데이터 세계



값  
사실

추상화  
개념적  
모델링

개념적 구조

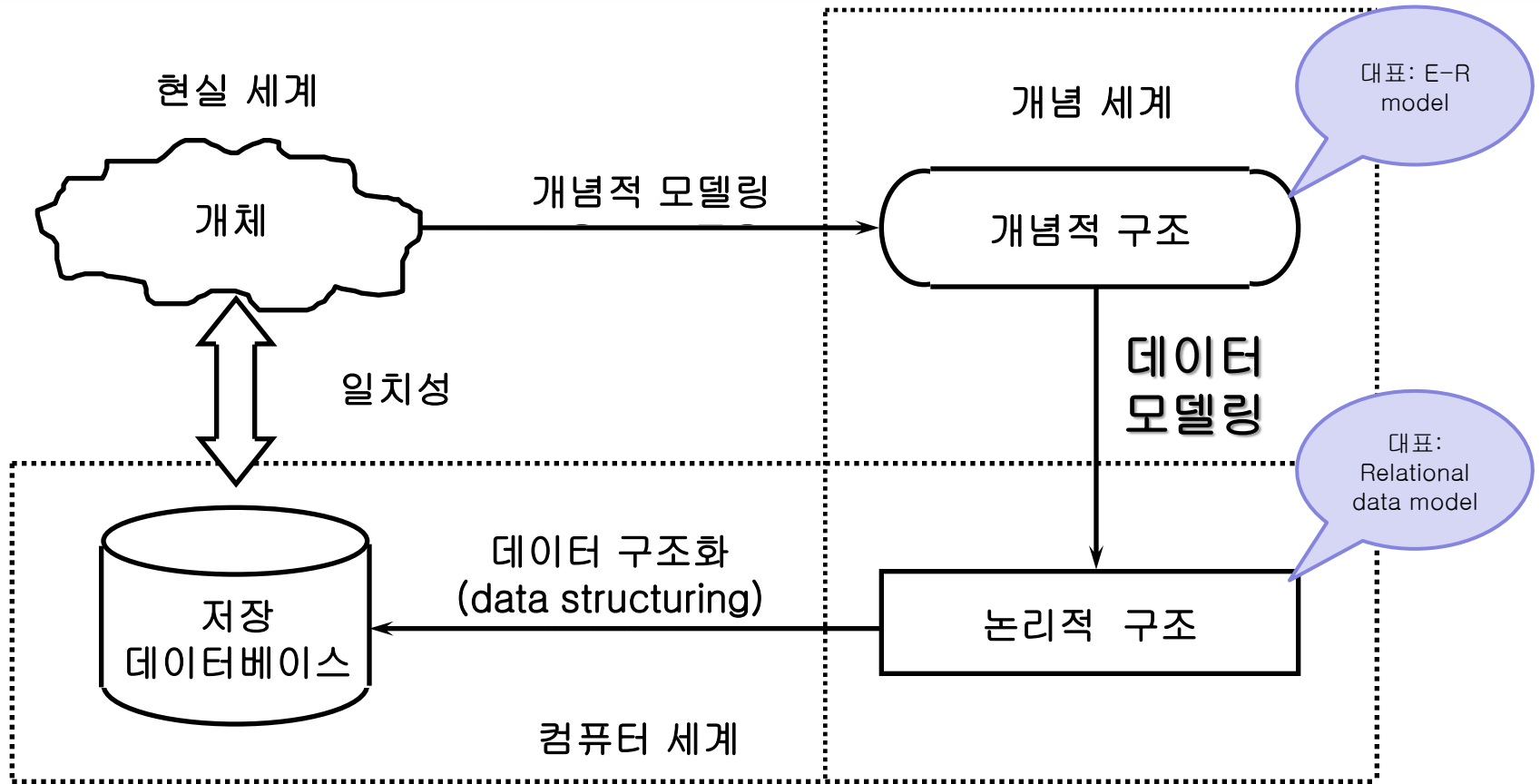
변환  
데이터  
모델링

논리적 구조  
(데이터 모델)

# 데이터베이스 설계

- 현실 세계의 구조적 표현 → 데이터베이스 설계
  - ◆ 무한성과 계속성의 현실 세계를 이해하기 위하여 추상화 기법 사용
- i. 개념적 모델링 → 개념적 구조
  - ◆ 개념적 설계 (conceptual design)
  - ◆ Entity, attribute 및 relationship 으로 구성
- ii. 데이터 모델링 → 논리적 구조
  - ◆ 논리적 설계 (logical design)
  - ◆ 테이블로 구성
    - 접근방법(access method)에 독립적 표현
- iii. 데이터 구조화 → 물리적 구조
  - ◆ 물리적 설계 (physical design)
  - ◆ 저장 장치에서의 데이터 표현

# 데이터 모델링



# 데이터 모델의 개념

- 데이터베이스 설계(database design)
  - ◆ 개념적인 구조와 논리적인 구조를 거쳐 실제 데이터를 저장할 수 있는 물리적 구조로 변환
  - ◆ 특정 RDBMS에 특화된 모델
- 개념적 데이터 모델(conceptual data model)
  - ◆ Attribute(속성)들로 기술된 개체 타입(entity type)과 이 개체 타입들 간의 관계를 이용하여 현실 세계를 표현하는 방법
  - ◆ 개체-관계 모델(**E-R : Entity-Relationship model**)
    - Entity & Relationship
- 논리적 데이터 모델(logical data model)
  - ◆ 데이터 필드로 기술된 데이터 타입(data type)과 이 데이터 타입들 간의 관계를 이용하여 현실 세계를 표현하는 방법
  - ◆ Relational data model

# 데이터 모델

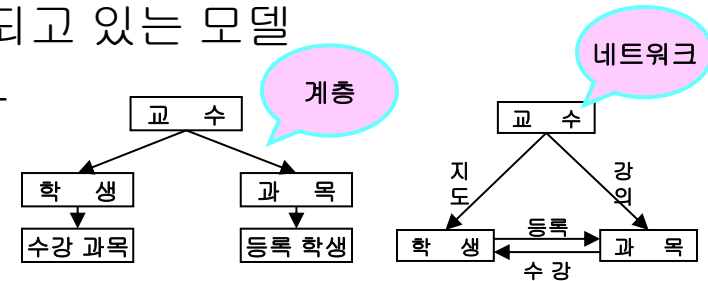
## ● 데이터 모델 (D)

$$D = \langle S, O, C \rangle$$

- ◆ S : 데이터의 구조(structure)
  - 정적 성질 (추상적 개념)
  - 개체타입과 이들 간의 관계를 명세
- ◆ O : 연산(operation)
  - 동적 성질 (임의의 상태에서 다른 상태로 전이)
  - 개체 인스턴스를 처리하는 작업에 대한 명세
  - 데이터의 조작 기법
- ◆ C : 제약 조건(constraint)
  - 데이터베이스에 허용될 수 있는 개체 인스턴스에 대한 논리적 제약 명세
    - 구조(S)로부터 파생된 구조적 제약
    - 실제 값에서 오는 의미상 제약
  - 데이터 조작의 한계를 표현한 규정

# 다양한 논리적 데이터 모델

- 논리적 데이터 모델은 개념적이므로, 특정 물리적 저장 구조를 포함하지 않음
- 관계 데이터 모델(relational data model)
  - ◆ 논리적 데이터 모델 중에서 가장 많이 사용되고 있는 모델
- 계층 데이터 모델(hierarchical data model) 과 네트워크 데이터 모델(network data model)
  - ◆ 이전에는 사용됨
- 객체 지향 데이터 모델(object-oriented data model)
  - ◆ 사용되지 않음
- 객체-관계 데이터 모델(object-relational data model)
  - ◆ 최근에는 RDB가 ORDB의 이름을 달고 나옴
- DBMS는 하나의 논리적 데이터 모델을 구현
  - ◆ 모델 간의 주요 차이점은 데이터 요소간의 관계(relationship)를 표현하는 방식



# Entity Type

## ● 개체 (entity)

- ◆ 단독으로 존재하며 다른 것과 구별되는 객체

## ● 개체 타입(entity type)

- ◆ 이름과 attribute(속성)들로 정의됨
  - Attribute는 개체의 특성을 기술함
- ◆ 개체 집합(entity set) : 특정 개체 타입에 대한 인스턴스 집합
  - 같은 attribute로 표현되는 개체 인스턴스의 집합

## ● Attribute의 유형

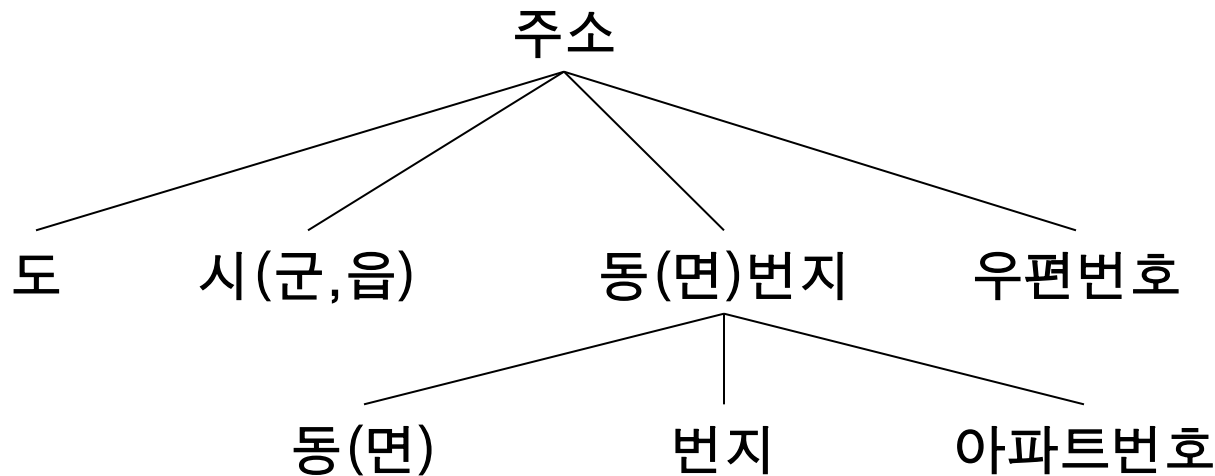
- ◆ i. Simple (단순) attribute와 composite (복합) attribute
- ◆ ii. Single-valued (단일 값) attribute와 multi-valued (다중 값) attribute
- ◆ iii. Derived (유도) attribute와 stored (저장) attribute
- ◆ iv. NULL attribute



# Attribute 유형 (1)

## Simple attribute와 Composite attribute

- ◆ Simple attribute : 더 이상 작은 구성 요소로 분해할 수 없는 attribute
- ◆ Composite attribute: 몇 개의 기본적인 simple attribute들로 분해할 수 있는 attribute



복합 attribute

# Attribute 유형 (2)

## ● 단일 값 attribute과 다중 값 attribute

### ◆ Single-valued attribute

- 대부분의 attribute들은 특정 개체에 대해 하나의 값을 갖음
- 학생 개체 타입의 이름 attribute : 각 개체에 대해 하나의 값만을 갖음

### ◆ Multi-valued attribute

- 한 개체에 대해서 몇 개의 값을 가지고 있을 수 있음
- 학생 개체 타입에 취미 attribute : 학생에 따라서는 두 개 이상의 취미를 가질 수 있음
- 현실적으로 multi-valued attribute으로 처리하기 위해서 값의 최대 수 제한 가능

# Attribute 유형 (3)

## 유도 attribute과 저장 attribute

### ◆ Derived attribute

- 어떤 attribute의 값은 다른 관련된 attribute이나 개체가 가지고 있는 값으로부터 derived(유도)되어 결정되는 경우

### ◆ Stored attribute

- Derived attribute을 만들 때 사용되는 attribute

### ◆ 과목별 평균 성적을 표현하는 과목-성적 개체 타입이 평균 성적 attribute을 포함하는 경우

- 평균 성적 attribute의 값은 등록 개체 타입의 성적 attribute 값을 가지고 계산한 결과로 결정
- 평균 성적 attribute : derived attribute
- 이를 위해 사용된 성적 attribute : stored attribute

# Attribute 유형 (4)

## Null attribute

- ◆ 널 값(null value)을 갖는 attribute
- ◆ 널 값은 어떤 개체 인스턴스가 어느 특정 attribute에 대한 값을 가지고 있지 않을 때 이를 명시적으로 표시하기 위해 사용

## 널 값을 갖는 경우

- ◆ 그 attribute 값이 그 개체에 “해당되지 않는”(not applicable) 경우
  - 여학생에 대한 병역 의무
- ◆ 그 attribute 값을 “알 수 없는”(unknown) 경우
  - 그 값이 존재하지만 값이 “누락”(missing)인 경우
  - 그 값이 존재하고 있는지조차 알 수 없어 “모르는”(not known) 경우

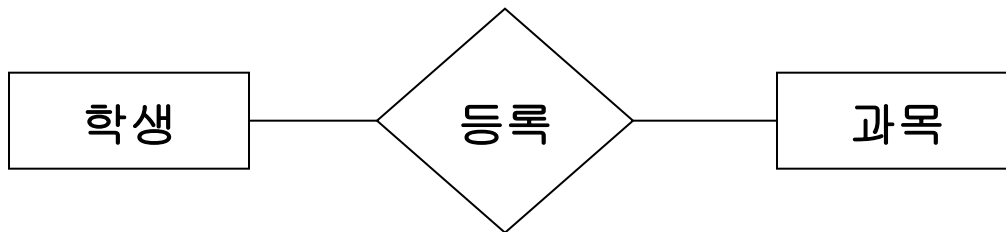
## 공백 또는 0과는 상이함

- ◆ 추가의 flag로 구현하는 경우 많음

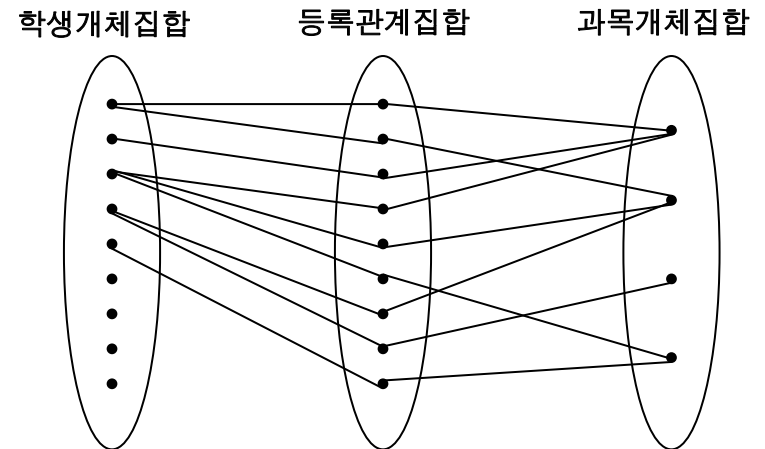
# Relationship Type

## ● 관계 타입 (relationship type)

- ◆ Entity set (개체 집합) 들 사이의 대응성 (correspondence)
- ◆ 사상 (mapping)



등록 관계 타입



등록 관계 인스턴스

# 관계 타입의 유형 (1)

## ● Mapping Cardinality 사상 원소수 - 관계의 분류 기준

◆ 4가지 중의 하나

- 1:1, 1:n, n:1, n:m

◆ 사상의 함수성(functionality)

-  $f_x: X \rightarrow Y, f_y: Y \rightarrow X$

## ● 유형

◆ 1 : 1 (일 대 일)

$f_x: x \rightarrow y$

and  $f_y: y \rightarrow x$

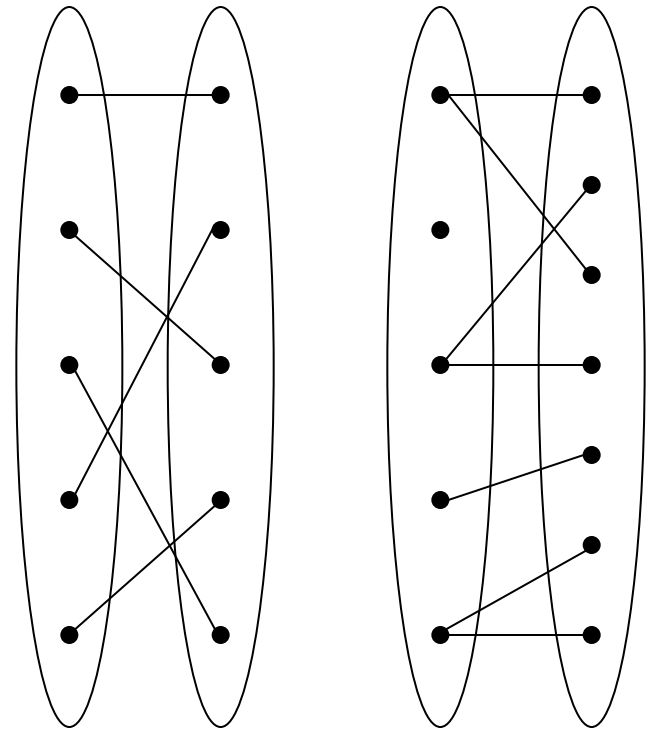
marriage : bridegroom  $\leftrightarrow$  bride

◆ 1 : n (일 대 다)

not  $f_x: x \rightarrow y$

but  $f_y: y \rightarrow x$

motherhood : mother  $\rightarrow$  children



(a) 일 대 일(1 : 1) (b) 일 대 다(1 : n)

# 관계 타입의 유형 (2)

◆ n : 1 ( 다 대 일 )

$fx : x \rightarrow y$

not  $fy : y \rightarrow x$

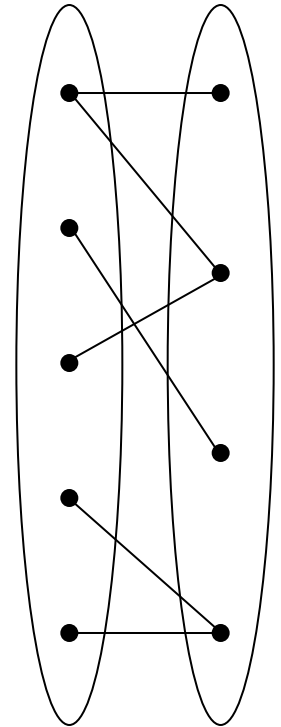
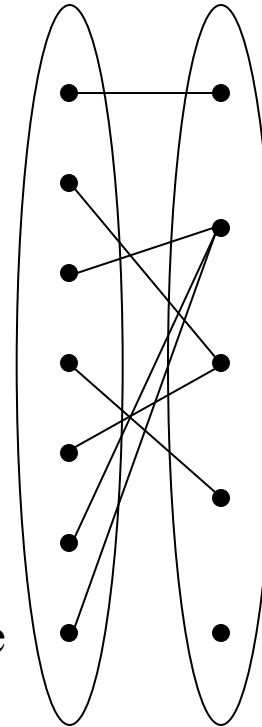
faculty : professor  $\rightarrow$  department

◆ n : m ( 다 대 다 )

not  $fx : x \rightarrow y$

not  $fy : y \rightarrow x$

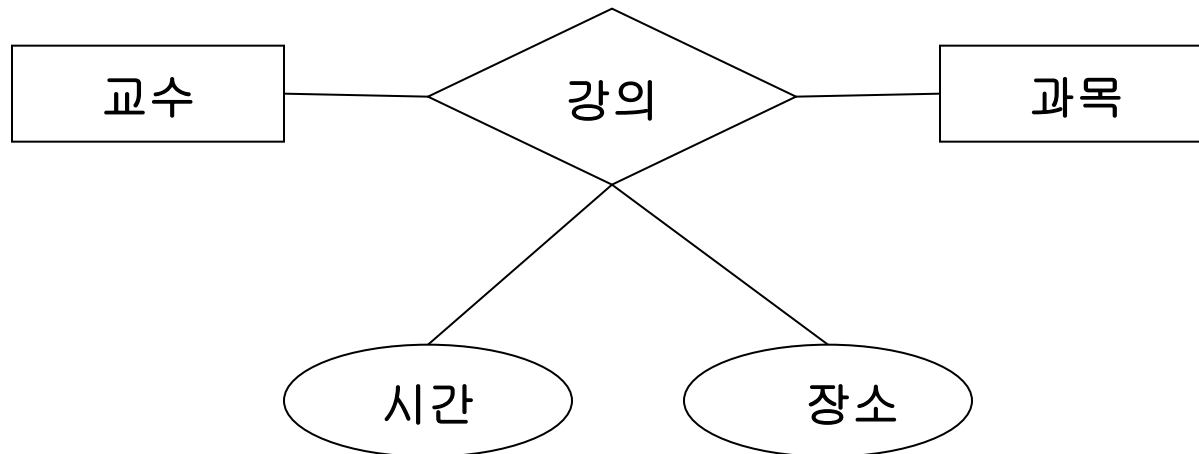
enrollment (등록) : student  $\leftrightarrow$  course



(c) 다 대 일(n : 1) (d) 다 대 다(n : m)

# 관계 타입의 특성 (1)

- Relationship도 Entity와 같이 Attribute 가질 수 있음



Attribute를 가진 관계 타입



## 관계 타입의 특성 (2)

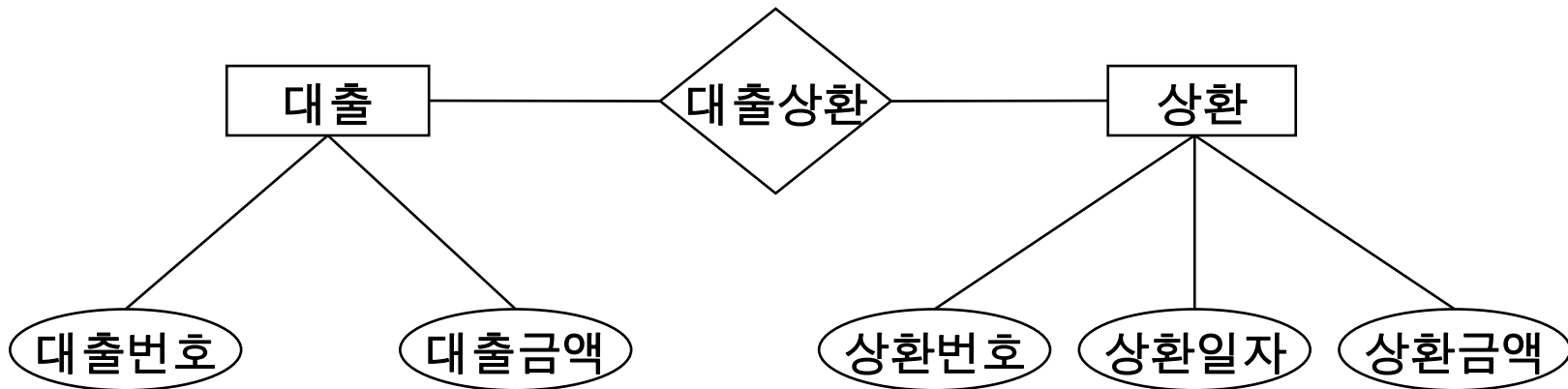
- Entity set A와 B 사이에 정의된 A-B relationship
- 전체 참여 (total participation)
  - ◆ A-B 관계에서 개체 집합A의 모든 개체가 A-B 관계에 참여
  - ◆ ex) 교수 – 학과
- 부분 참여 (partial participation)
  - ◆ A-B 관계에서 개체 집합A의 일부 개체만 A-B 관계에 참여
  - ◆ ex) 학생 (휴학생 허용시) – 과목

# 관계 타입의 특성 (3)

## 존재 종속 (existence dependence)

- ◆ 어떤 개체 b의 존재가 개체 a의 존재에 종속됨
- ◆ 즉 a가 삭제되면 b도 삭제되어야 함
- ◆ b는 a에 존재 종속
- ◆ a : 주 개체(dominant entity) b : 종속 개체(subordinate entity)

Constraint rule



대출 상환 관계 ( 주 개체: 대출, 종속 개체: 상환)

# 개체 – 관계 모델(E-R model)

## Entity-Relationship model

- ◆ 현실세계의 개념적 표현
- ◆ 개체 타입과 관계 타입을 기본 개념으로 현실 세계를 개념적으로 표현하는 방법
- ◆ 개체 집합 : 한 개체 타입에 속하는 모든 개체 인스턴스
- ◆ 관계 집합 : 한 관계 타입에 속하는 모든 관계 인스턴스

# E-R Diagram

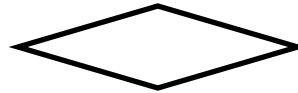
1976. Peter Chen

E-R 모델의 그래픽 표현

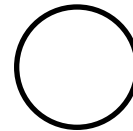
◆ 개체 타입 (entity type)



◆ 관계 타입 (relationship type)



◆ attribute(attribute)



◆ 링크(link)

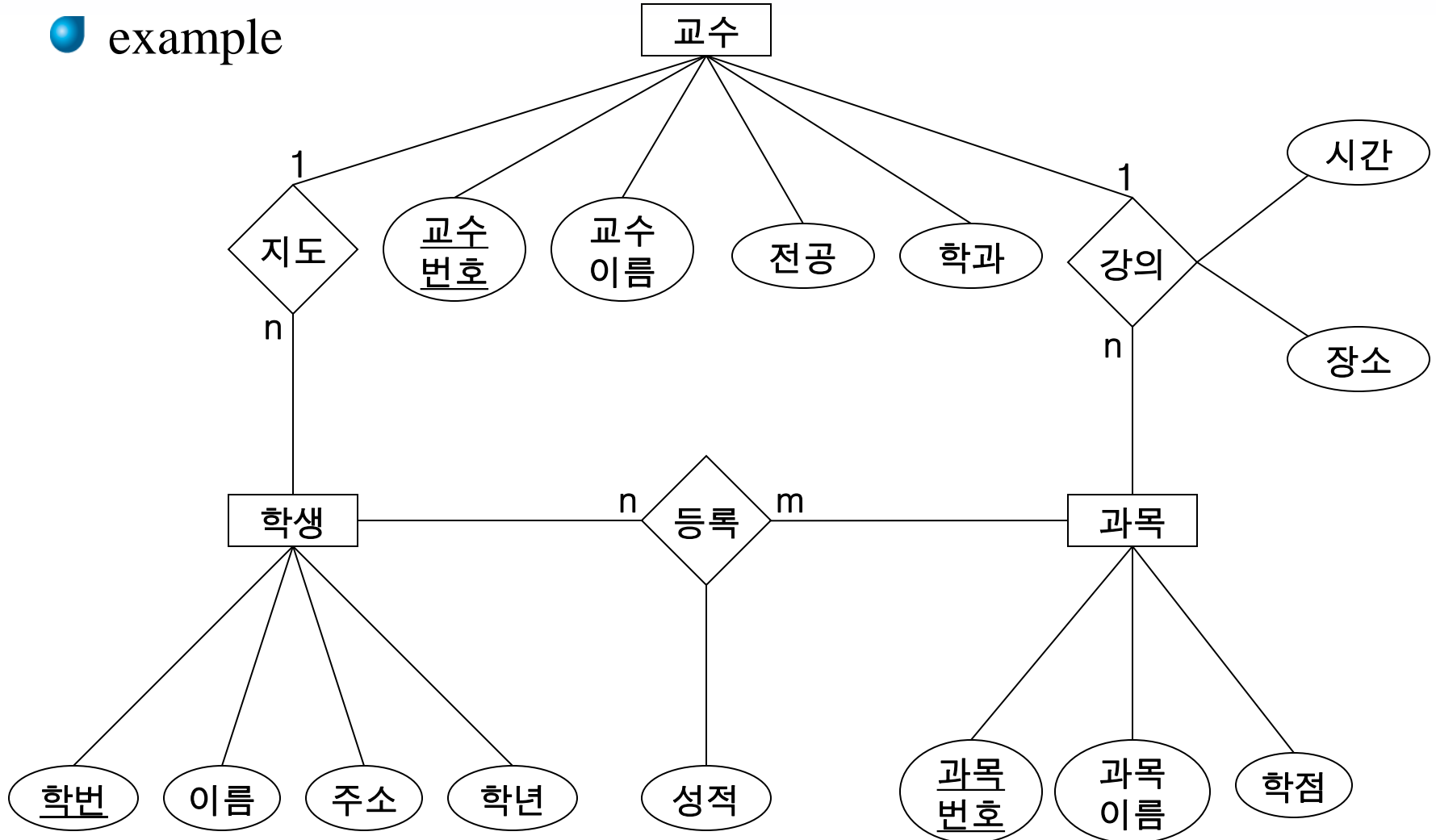


◆ 레이블(label) : 관계의 사상 원소 수 표현

- 일대일(1:1), 일대다(1:n), 다대일(n:1), 다대다(n:m)

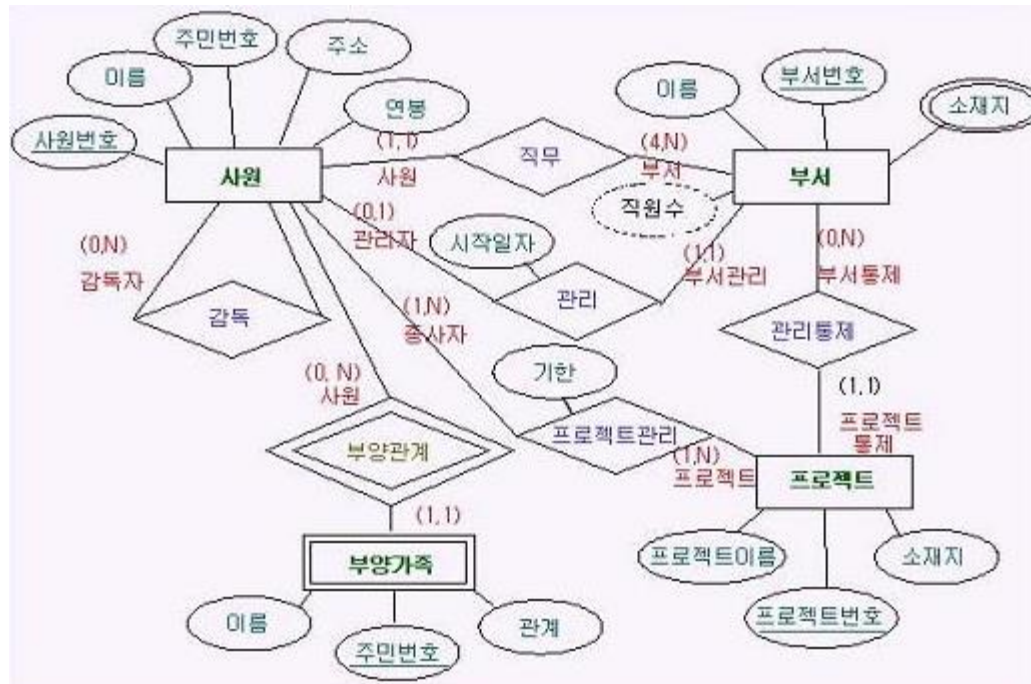
# E-R Diagram의 예제

example



# E-R Diagram의 예제

example

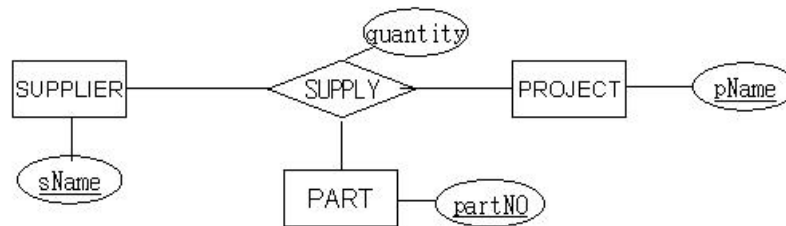


# E-R Diagram의 특징

## 특징

- ◆ 다 대 다(n : m) 관계 표현
- ◆ 다원 관계(n-ary relationship) 표현
  - 두 개 이상의 개체 타입이 하나의 관계에 관련 가능
  - e.g., Suppliers-Parts-Projects

Unary  
Binary  
Ternary  
Quaternary  
...  
n-ary

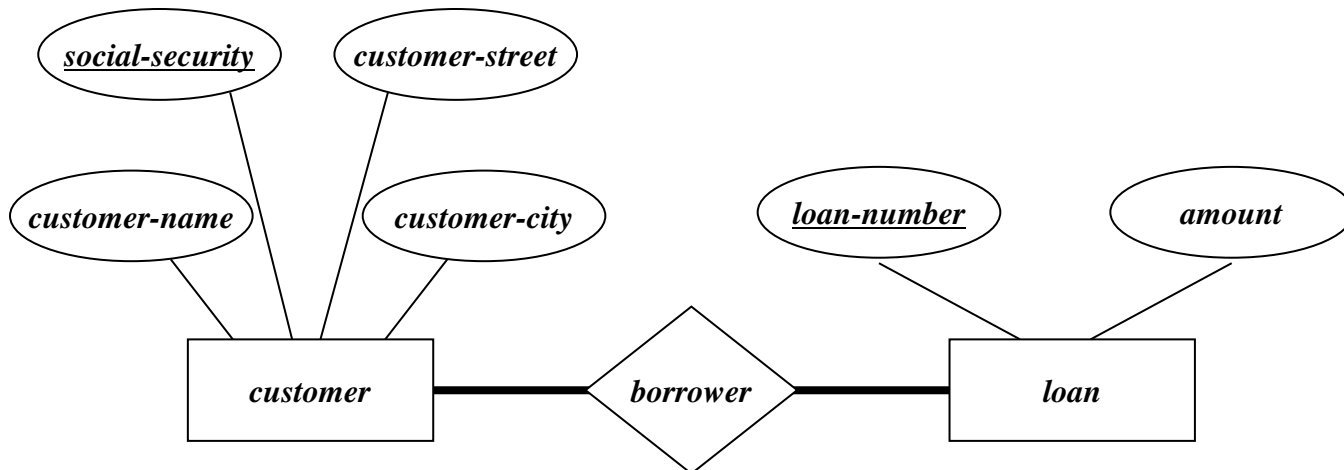


- ◆ 다중 관계(multiple relationship) 표현
  - 두 개체 타입 사이에 둘 이상의 관계가 존재 가능
- ◆ 관계 타입도 attribute를 가질 수 있음

# Key Attribute

## Key attribute

- ◆ 개체 집합 내에 각 개체마다 상이한 값을 갖는 attribute
- ◆ 키 (key)
  - 개체 타입내의 모든 개체 인스턴스들을 유일하게 식별
  - 동일한 키 값을 갖는 두 개의 개체 인스턴스는 없음
- ◆ E-R 다이어그램 상에서 밑줄로 표시





# Weak Entity Type

## ● 약한 개체 타입(weak entity type)

- ◆ 존재 의존성을 가진 개체 (예:대출 상환, 부양가족)
- ◆ 주 개체 – 강한 개체 타입, 종속 개체 – 약한 개체 타입
- ◆ 자기자신의 attribute로만 키를 명세할 수 없는 개체 타입  
↔강한 개체 타입(strong entity type)

## ● 구별자(discriminator)

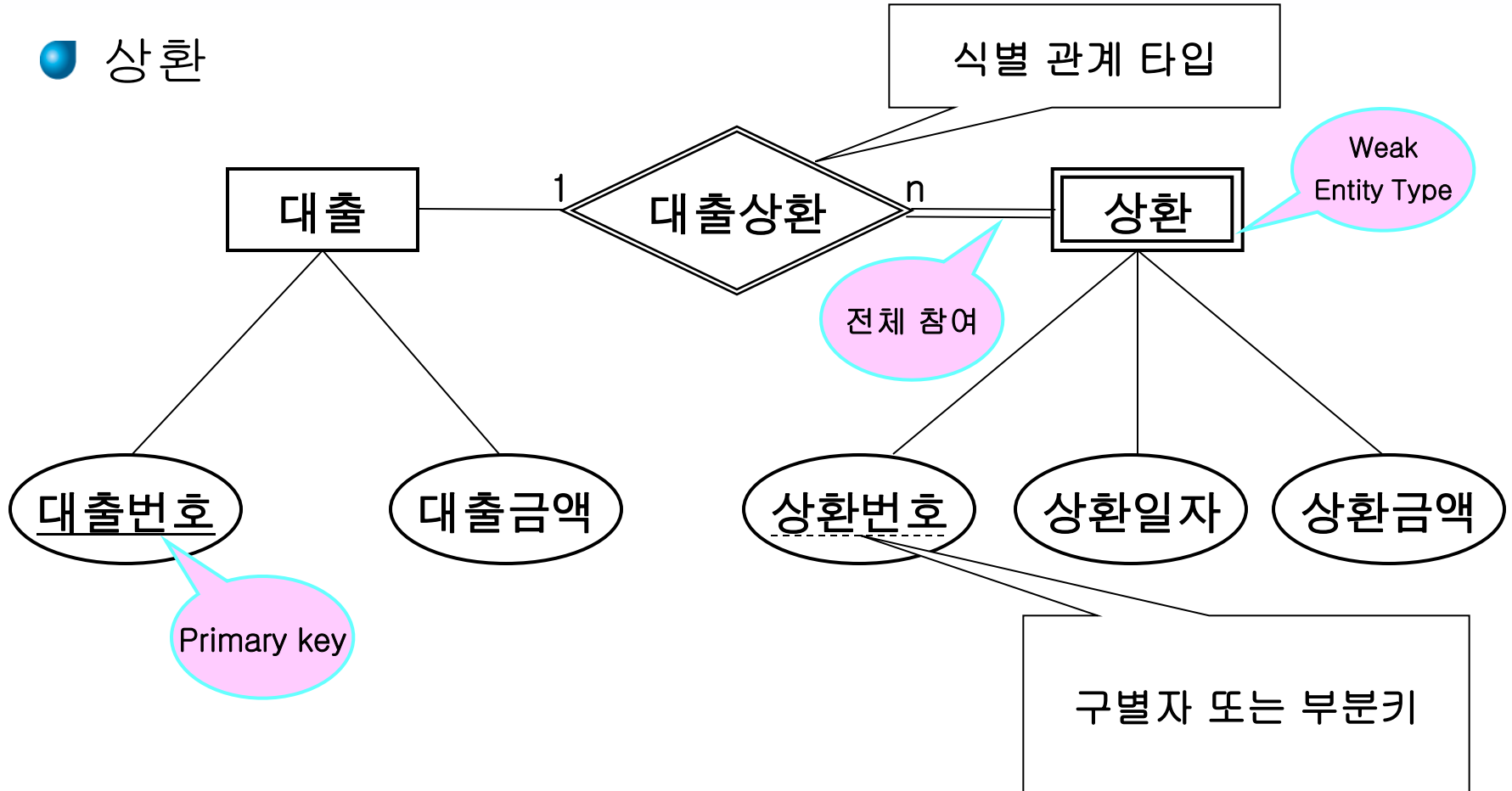
- ◆ 강한 개체와 연관된 약한 개체집합에서 이들을 서로 구별할 수 있는 attribute
- ◆ 부분키(partial key)라고도 함
- ◆ 실제 키: {강한 개체 키 + 부분키}

## ● 식별 관계 타입(identifying relationship type)

- ◆ 약한 개체를 강한 개체에 연관

# Weak Entity Type 예제

## 상환

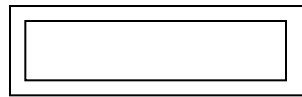


대출 상환 관계의 E-R 다이어그램

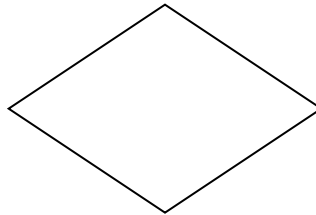
# E-R 다이어그램 표기법



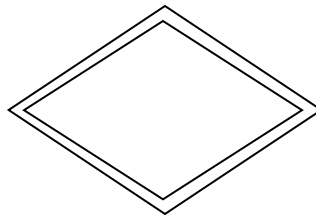
개체 타입



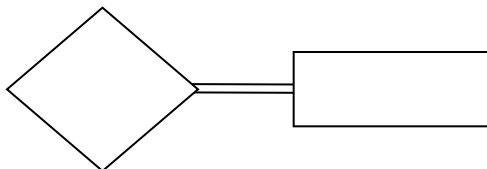
약한 개체 타입



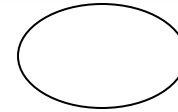
관계 타입



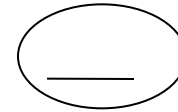
식별 관계 타입



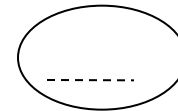
전체 참여 개체 타입



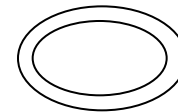
attribute



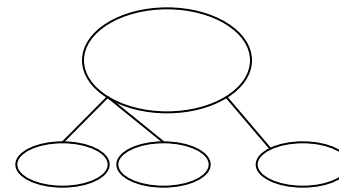
키 attribute



부분키 attribute



다중값 attribute



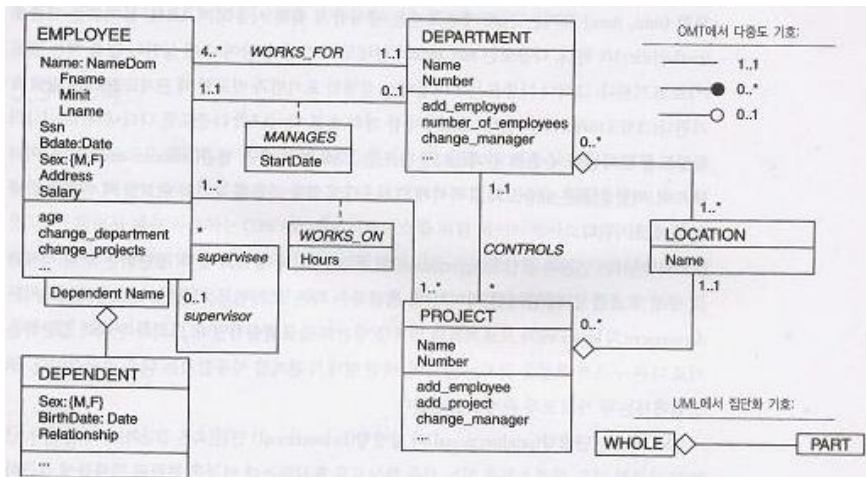
복합 attribute



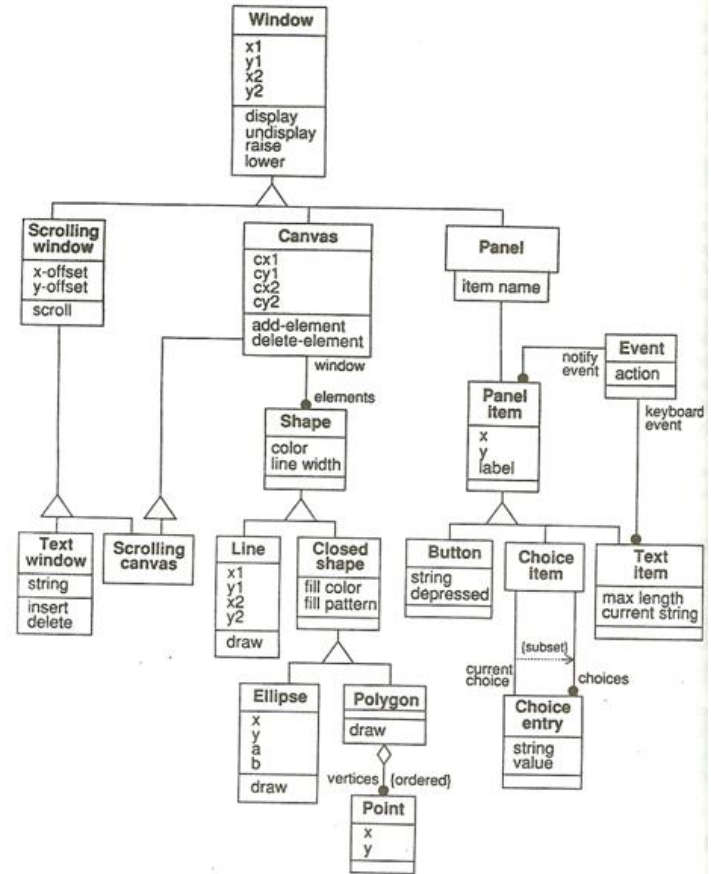
유도 attribute

# 그 밖의 다이어그램

UML 클래스 다이어그램



OMT (Object Modeling Technique)



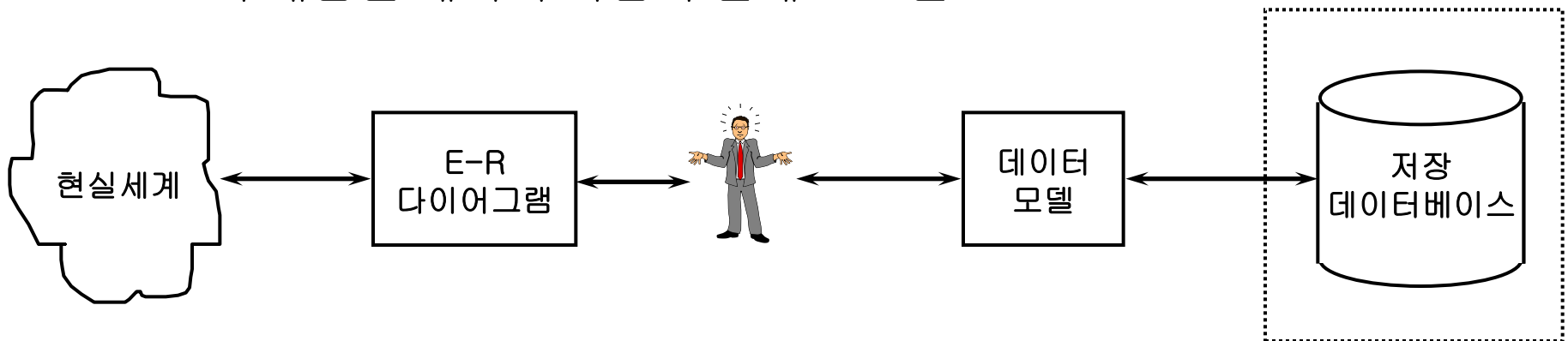
# 논리적 데이터 모델

## 개념적 데이터 모델

- ◆ 개체-관계 데이터 모델
- ◆ 현실 세계를 추상적 개념인 개체 타입과 관계 타입으로 표현
  - 특정 DBMS를 고려한 것이 아님

## 논리적 데이터 모델 (DBMS 고려)

- ◆ 개념적 구조를 데이터베이스로 구현하기 위한 중간 단계로 논리적 개념인 데이터 타입과 관계로 표현

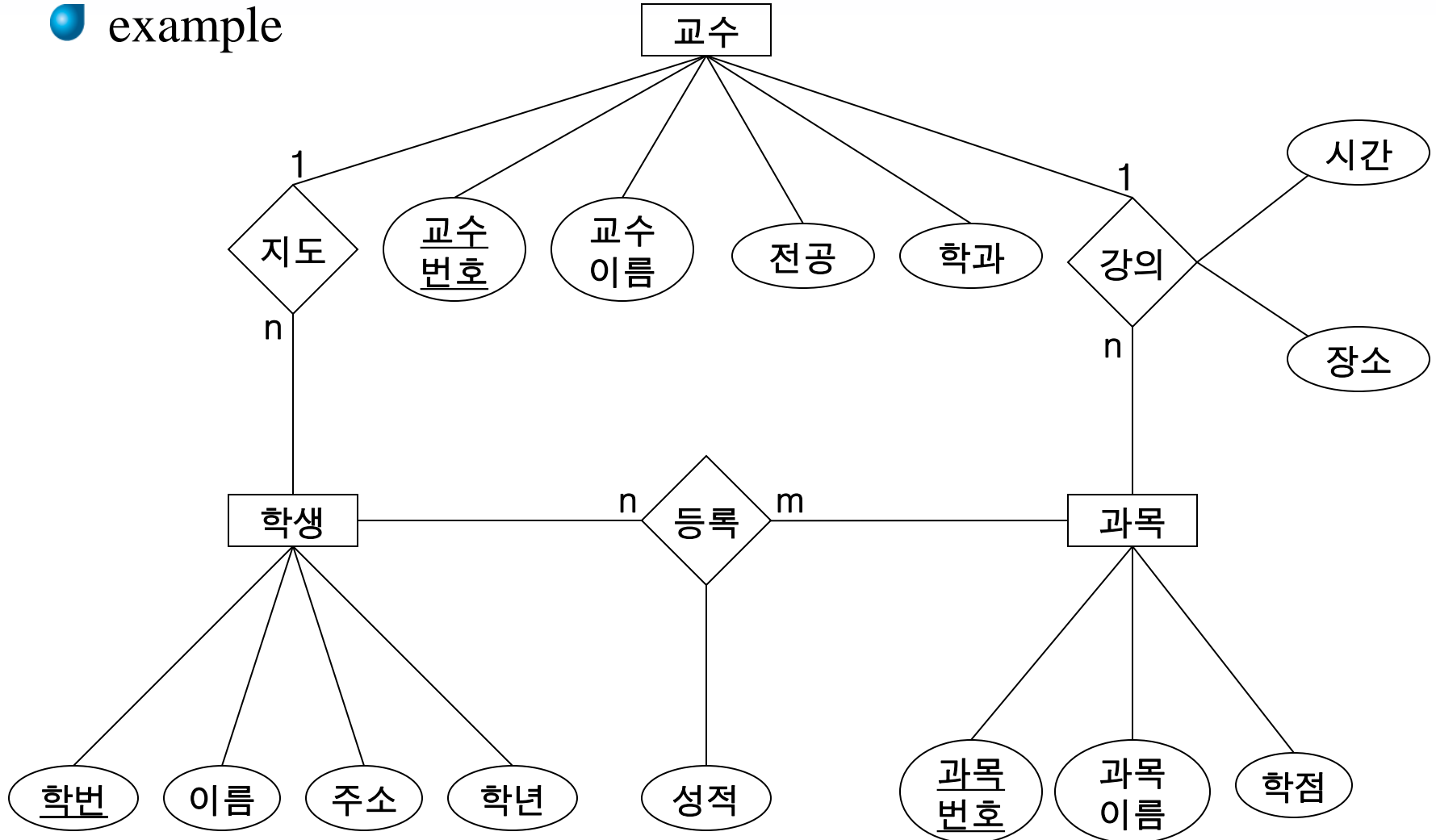


# 데이터 모델

- 관계를 나타내는 방법에 따라
  - ◆ 관계 데이터 모델 (Relational Data Model)
  - ◆ 계층 데이터 모델 (Hierarchical Data Model)
  - ◆ 네트워크 데이터 모델 (Network Data Model)
- 객체 지향을 지원하는 방법에 따라
  - ◆ 객체지향 데이터 모델 (Object-Oriented Data Model)
  - ◆ 객체관계 데이터 모델 (Object-Relational Data Model)

# 개념적 데이터 모델

example



# 관계 데이터 모델

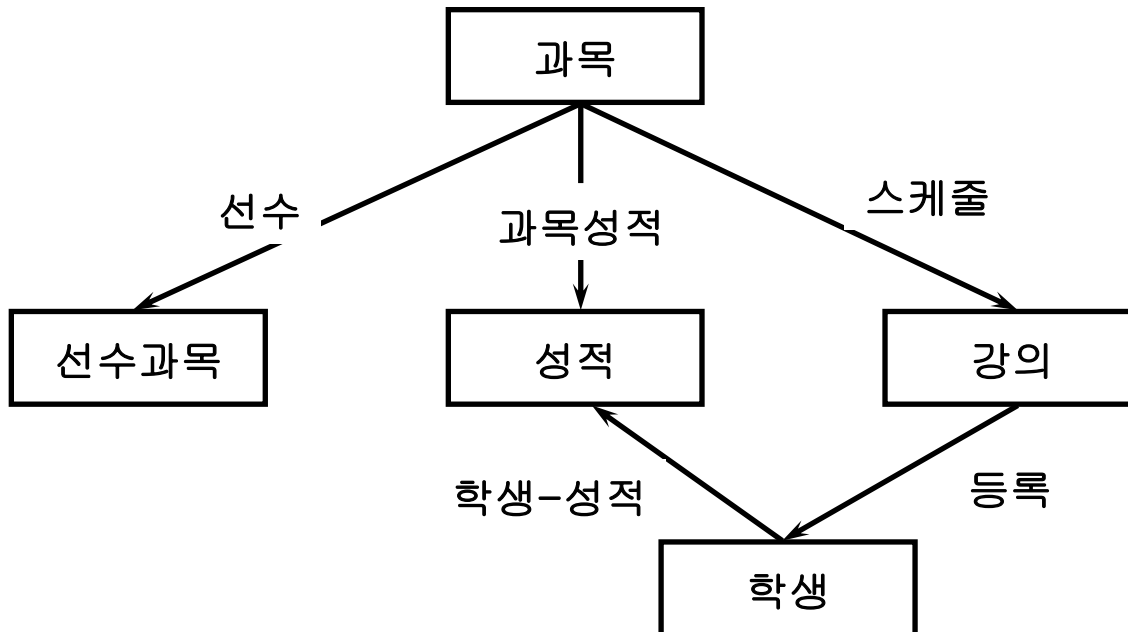
- 데이터베이스
  - ◆ 관계 (테이블)의 집합
  - ◆ 개체 릴레이션, 관계 릴레이션
- 관계 스키마 (Relation Scheme)
  - ◆ 개체와 관계성을 테이블로 정의

|    |    |       |       |     |     |
|----|----|-------|-------|-----|-----|
| 개체 | 학생 | 학 번   | 이 름   | 주 소 | 학 년 |
|    | 교수 | 교수 번호 | 교수 이름 | 전 공 | 학 과 |
|    | 과목 | 과목 번호 | 과목 이름 | 학 점 |     |
| 관계 | 지도 | 교수 번호 |       | 학 번 |     |
|    | 등록 | 학 번   | 과목 번호 | 성 적 |     |
|    | 강의 | 교수 번호 | 과목 번호 | 시 간 | 장 소 |



# 자료 구조도 (1)

- Bachman 다이어그램 (1969)
- 레코드 타입 간의 관계에 대한 도형적 표현
- 네트워크 데이터 모델과 계층 데이터 모델은 자료 구조도로 표현 가능



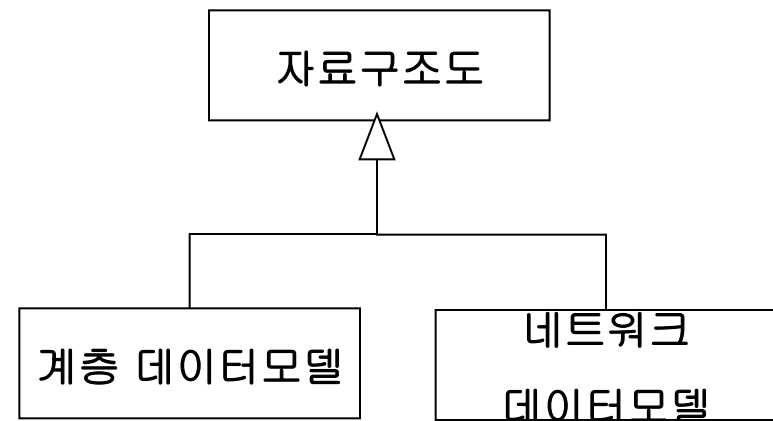
# 자료 구조도 (2)

## 구성 요소

- ◆ 사각형 노드 : 레코드 타입 (개체 타입)
- ◆ 링크(link;arc) : 레코드 타입 간의 일대다 (1:n) 관계
- ◆ 레이블 : 관계 이름

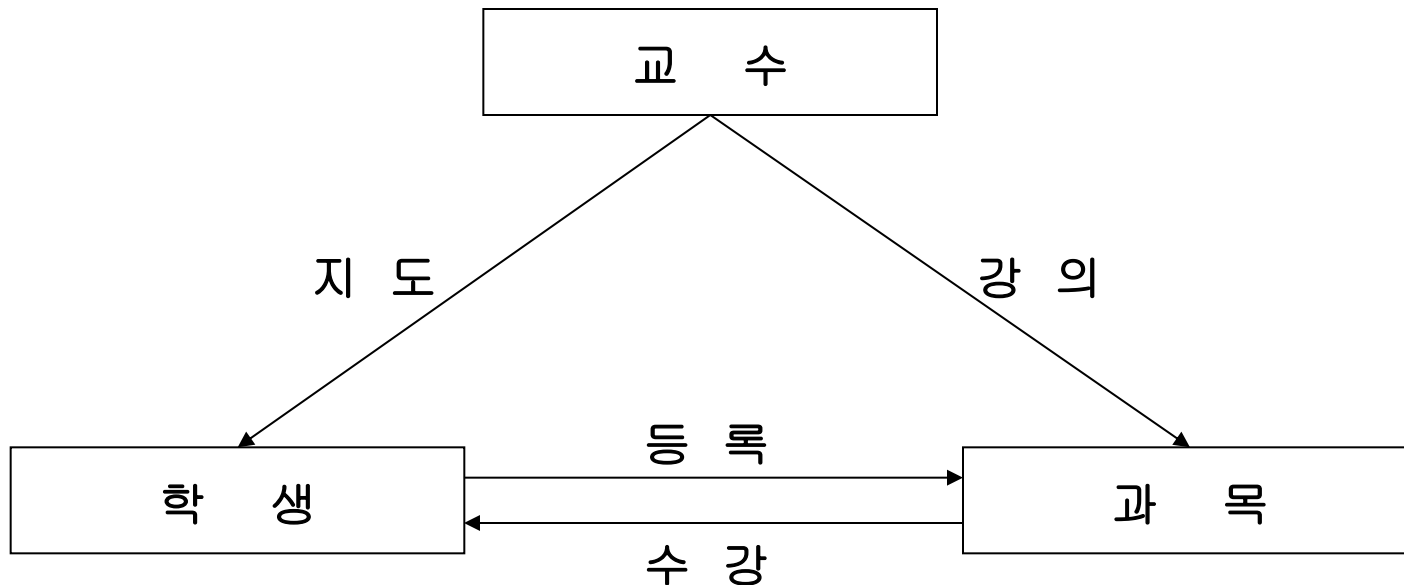
## 스키마 다이어그램

- ◆ 데이터베이스 스키마를 데이터 구조도로 표현
  - - 스키마의 도형적 표현
- ◆ 트리 형태 : 계층 데이터 모델
- ◆ 그래프 형태 : 네트워크 데이터 모델



# 네트워크 데이터 모델

- 스키마 다이어그램이 Network(그래프)
- 허용되는 레코드 타입, 관계성을 명세
- Owner-member 관계
  - ◆ 두 레코드 타입간의 1:n 관계
  - ◆ n:m (교수 학생 관계) → 등록(1:n)과 수강(1:n)으로 표현



# 계층 데이터 모델

- 스키마 다이어그램이 트리, No Cycle
- 루트 레코드, 자식 레코드, 레벨
- 자식-부모(Parent-Child)관계
  - ◆ 1:n 관계 만 가능
  - ◆ n:m (교수 학생 관계) → 교수 -> 과목 -> 등록학생으로 변환

