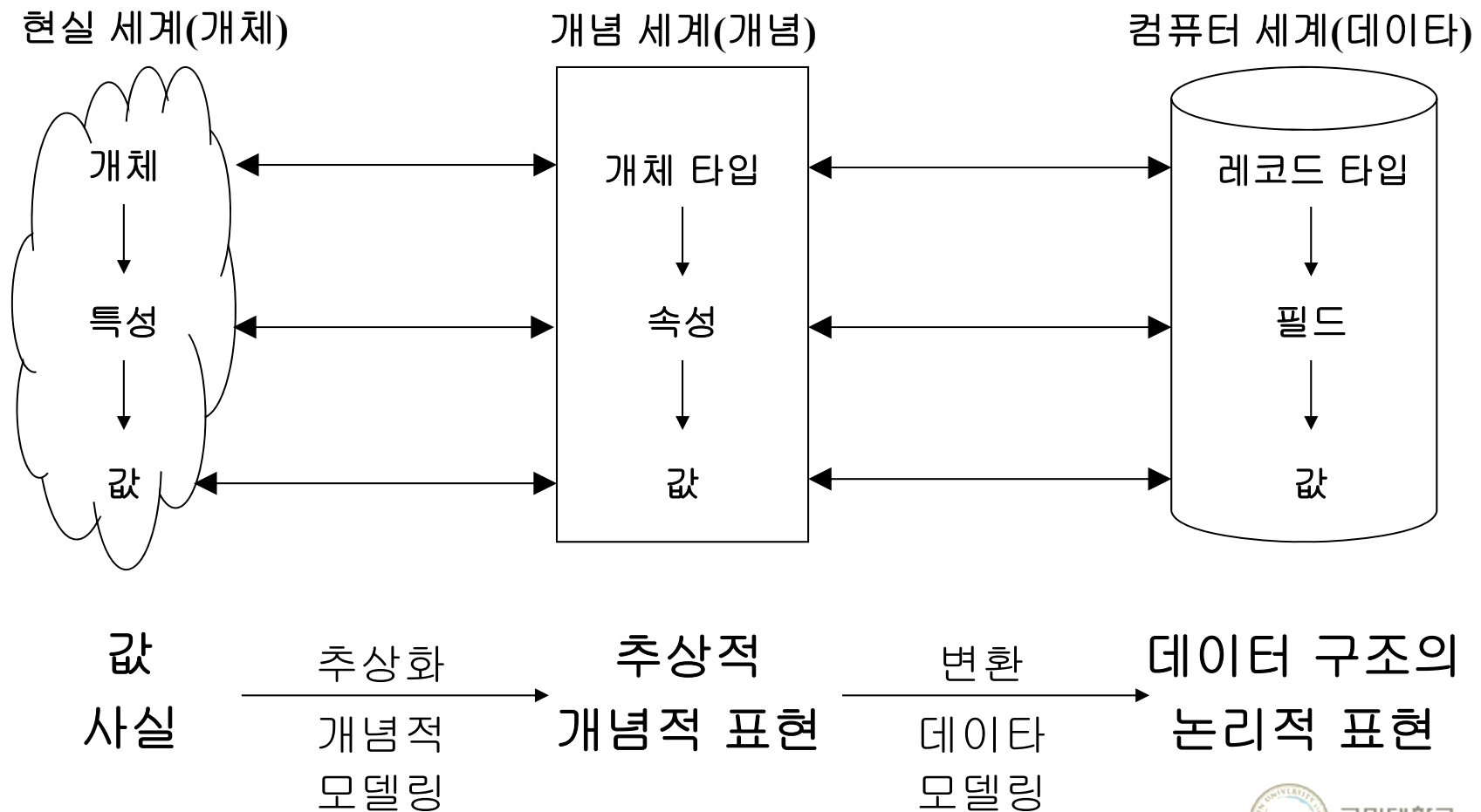


5. 데이터베이스 모델링

5.1 데이터의 세계

□ 3개의 데이터 세계



현실 세계의 모델링

i . 개념적 설계 (conceptual design)

- 개념적 모델링 : 개념적 표현
- Entity type & attribute

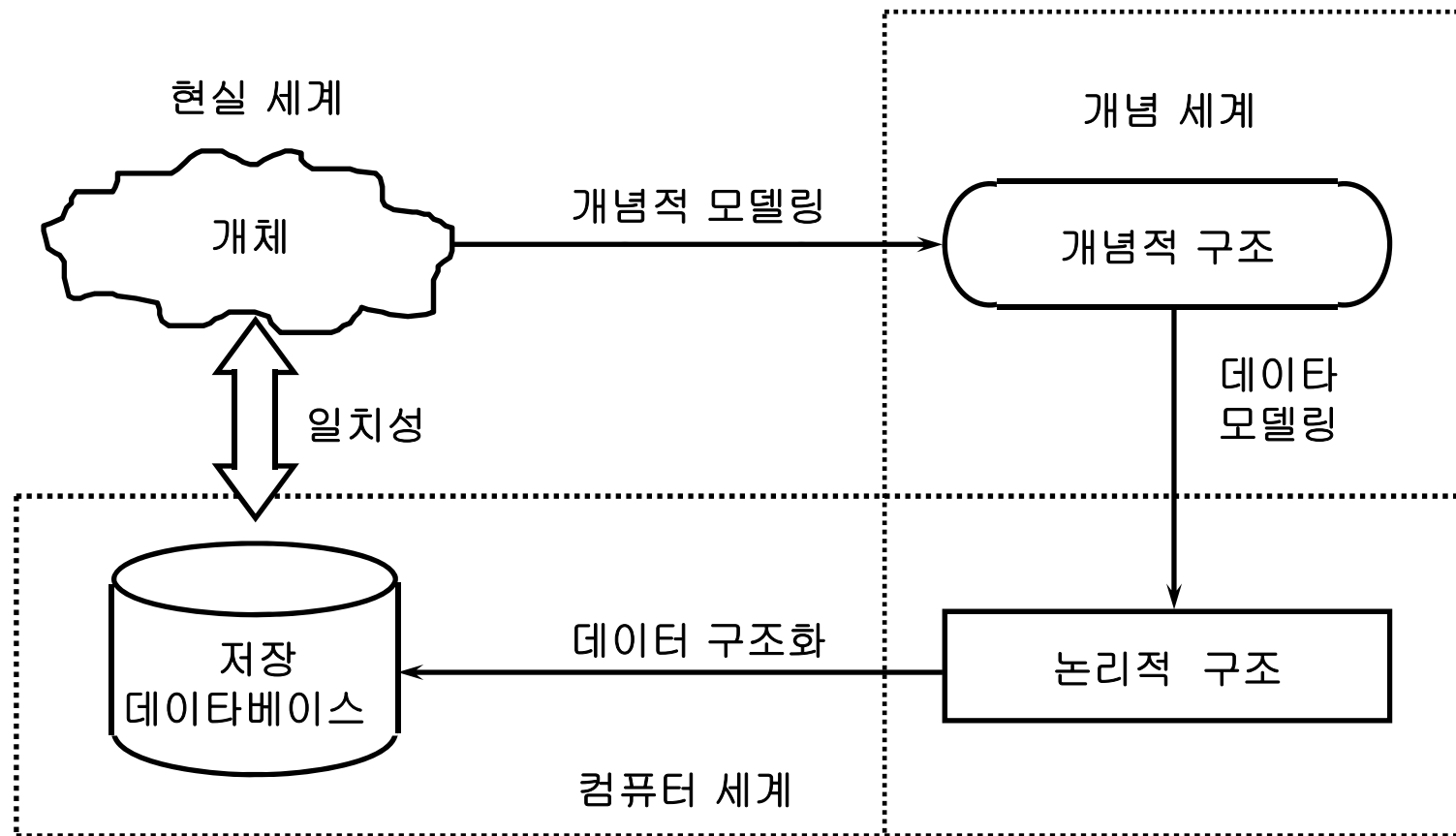
ii . 논리적 설계 (logical design)

- 데이터 모델링(data modeling) : 논리적 표현
→ 접근방법(access method)에 독립적인 표현

iii . 물리적 설계 (physical design)

- 데이터 구조화(data structuring) : 구현
- 저장 장치에서의 데이터 표현

데이터베이스와 현실 세계



5.2 데이터 모델의 개념

□ 데이터 모델

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

- S : 데이터의 구조(structure)
 - ◆ 정적 성질 (추상적 개념)
 - ◆ 개체타입과 이들 간의 관계를 명세
- O : 연산(operation)
 - ◆ 동적 성질
 - ◆ 개체 인스턴스를 처리하는 작업에 대한 명세
 - ◆ 데이터의 조작 기법
- C : 제약 조건(constraint)
 - ◆ 데이터의 논리적 제약
 - 개체 인스턴스의 허용 조건
 - 구조(S)로부터 파생
 - 의미상 제약
 - ◆ 데이터 조작의 한계를 표현한 규정

5.3 개체 타입

□ 개체 타입(entity type)

- 이름과 애트리뷰트들로 정의됨
- 개체 집합(entity set) : 특정 개체 타입에 대한 인스턴스 집합

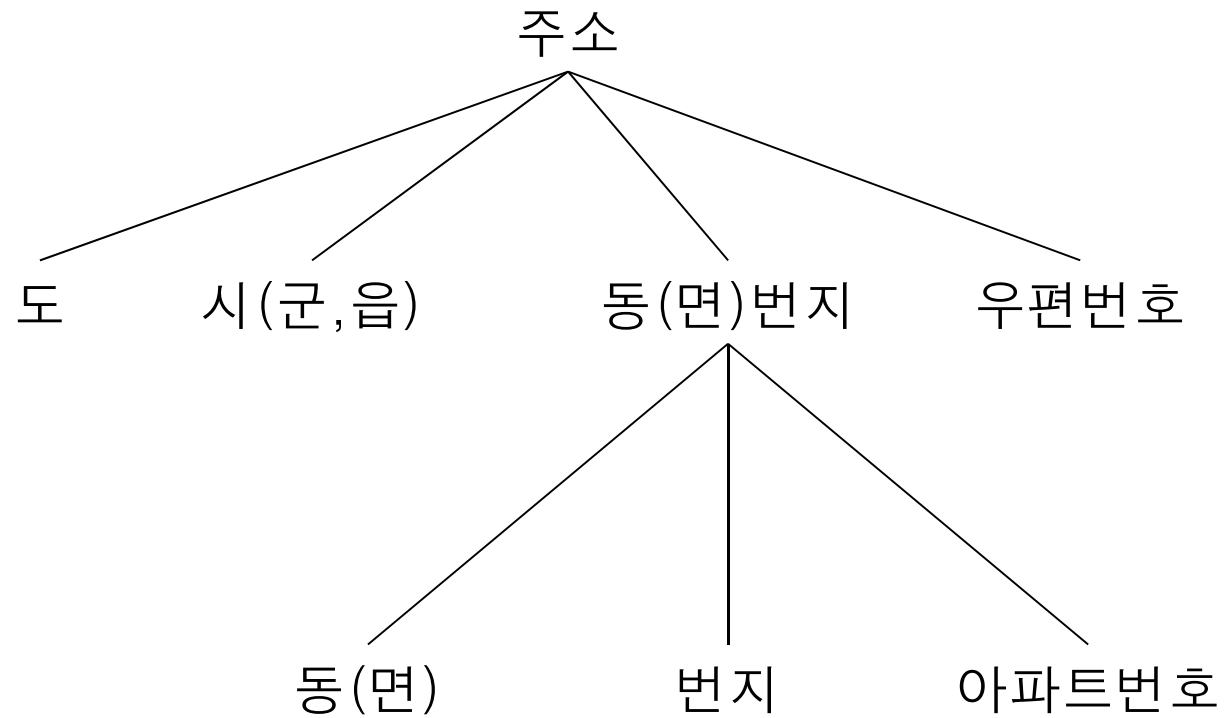
□ 애트리뷰트(attribute)의 유형

- i. 단순(simple) 애트리뷰트와 복합(composite) 애트리뷰트
- ii. 단일값(single-valued) 애트리뷰트와 다중값(multivalued) 애트리뷰트
- iii. 저장(stored) 애트리뷰트와 유도(derived) 애트리뷰트

Note: 널(null) 애트리뷰트

- not applicable
- unknown: missing, not known

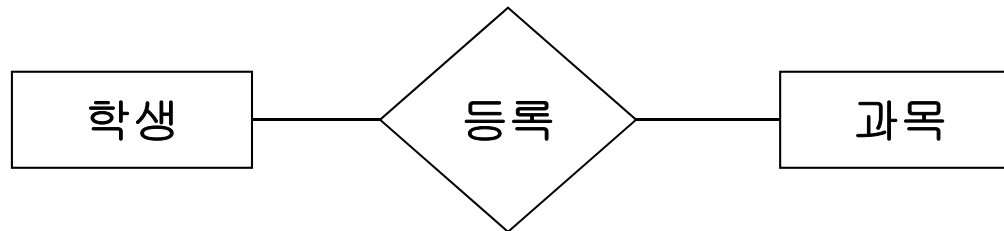
복합 애트리뷰트



복합 애트리뷰트

5.4 관계 타입과 구조적 제약조건 (Structural Constraints)

- 관계 타입 (relationship type)
 - 개체 집합들 사이의 대응성 (correspondence)
 - 사상 (mapping)



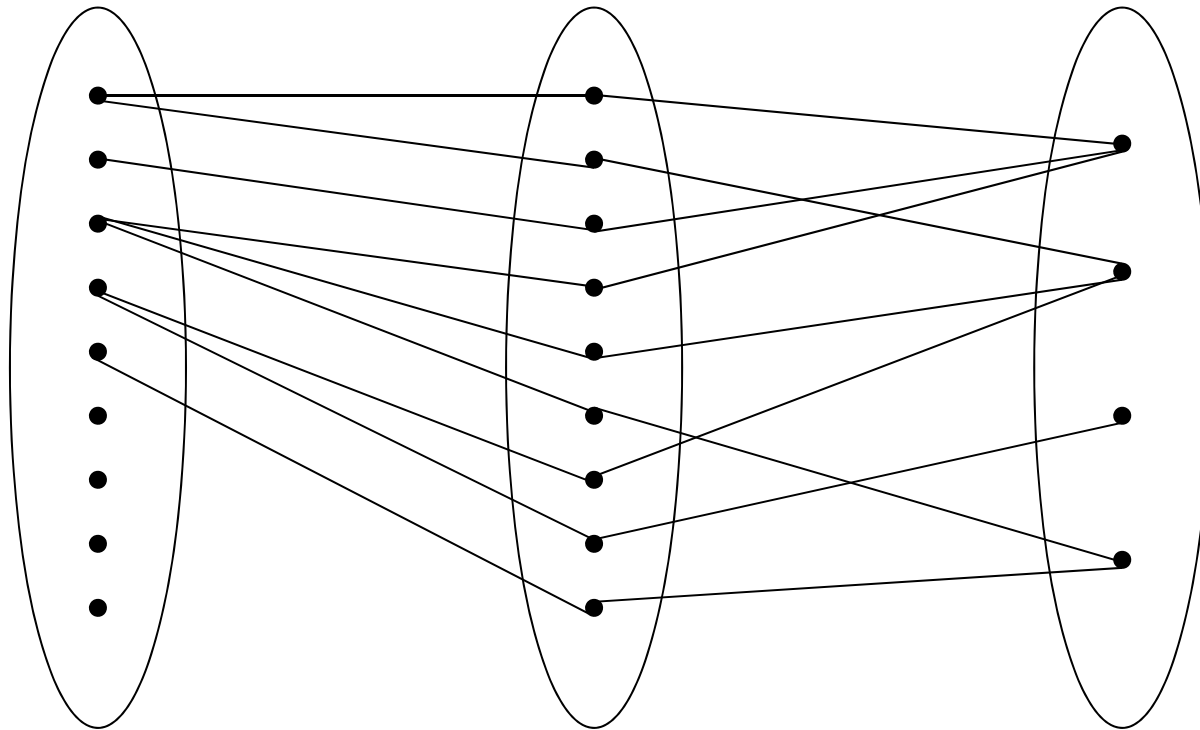
등록 관계 타입

관계 타입과 인스턴스

학생개체 집합

등록관계 집합

과목관계 집합



관계 타입과 인스턴스

관계타입의 구조적 제약조건

- 카디널리티 비율(cardinality ratios)
- 참여(participations)

관계 타입의 카디널리티 비율(Cardinality ratio)

□ 사상 원소수 – 관계의 분류 기준

– 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)

$f_x : x \rightarrow y$ or

$f_y : y \rightarrow x$ (but not both)

motherhood : mother \rightarrow children

-
- $n : 1$ (다 대 일)
 $fx : x \rightarrow y$ or
 $fy : y \rightarrow x$ (but not both)
position : professor \rightarrow department
 - $n : m$ (다 대 다)
 $fx : x \rightarrow y$
 $fy : y \rightarrow x$ (neither)
enrollment : student \leftrightarrow course

👉 Note : 사상의 함수성(functionality)

관계 타입에의 참여(Participation)

□ 전체 참여(total participation)

- A-B 관계에서 개체 집합B의 모든 개체가 A-B 관계에 참여해야 된다면 B의 A-B에의 참여를 전체 참여라 함.
- ex) 학과 - 교수

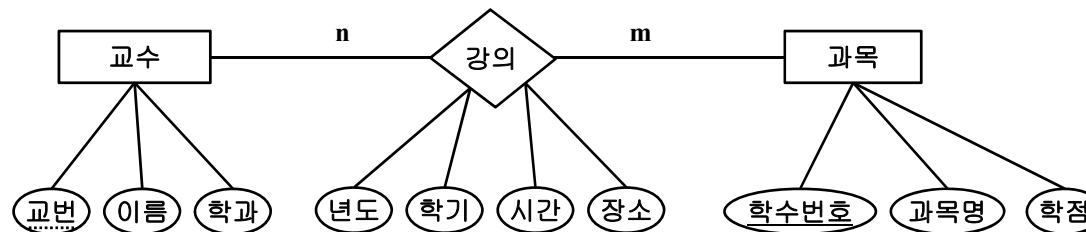
□ 부분 참여(partial participation)

- A-B 관계에서 개체 집합B의 일부 개체만 A-B 관계에 참여해도 된다면 B의 A-B 관계에의 참여를 부분 참여라 함.
- ex) 과목 - 학생 (휴학생 허용시)

관계 타입의 특성

□ 애트리뷰트를 가진 관계 타입

- 관계 타입도 개체 타입과 같이 애트리뷰트를 가질 수 있다.



5.5 개체 – 관계 모델

- 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 다이어그램

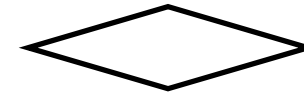
□ 1976. Peter Chen

□ E-R 모델의 그래픽 표현

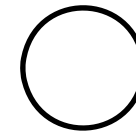
– 개체 타입 (entity type)



– 관계 타입 (relationship type)



– 속성 (attribute)



– 링크 (link)



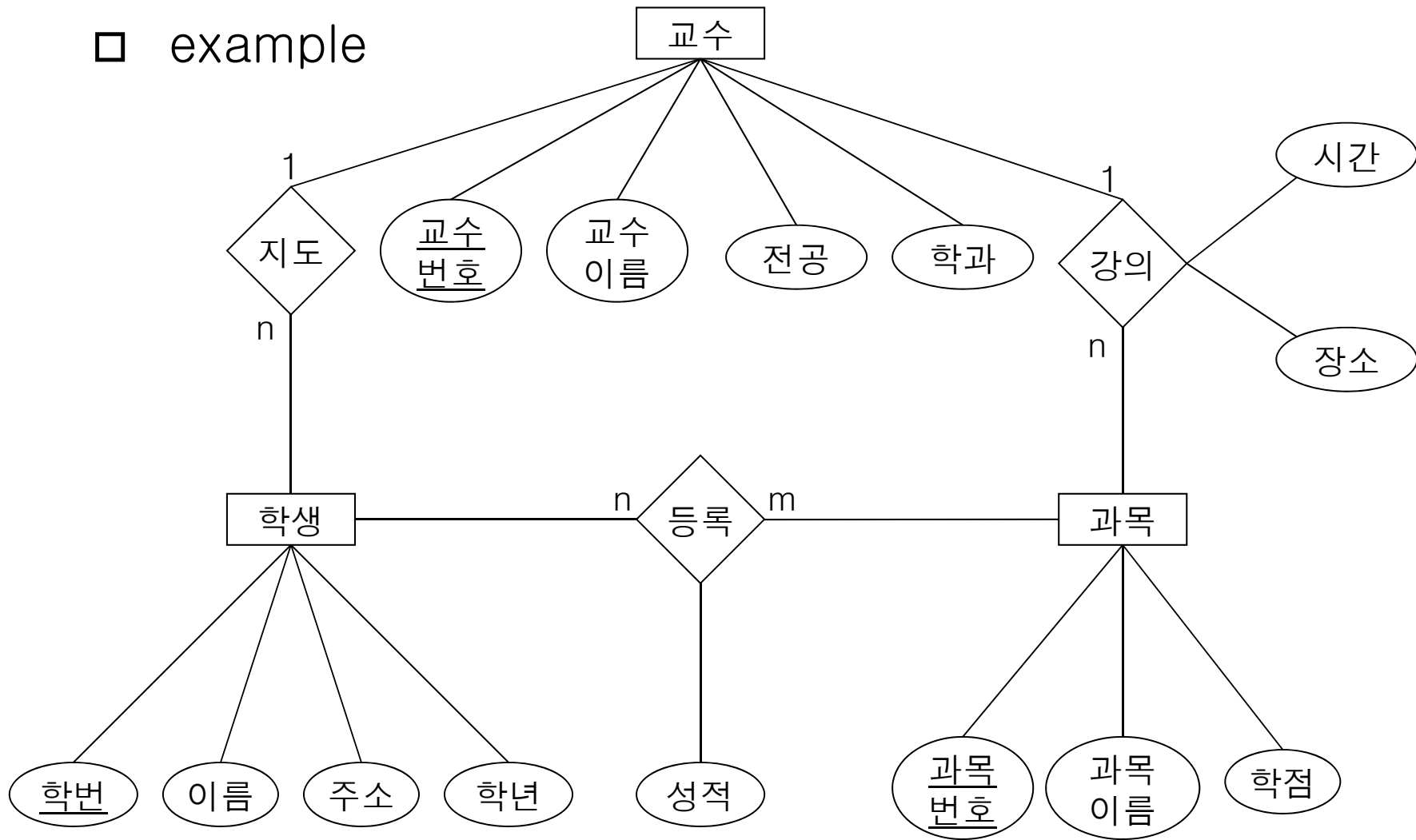
– 레이블 (label) : 관계의 사상, 원소수를 표현,
일대일(1:1), 일대다(1:n), 다대다(n:m)

개체 타입과 키 애트리뷰트

□ 키 애트리뷰트(key attribute)

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

□ example

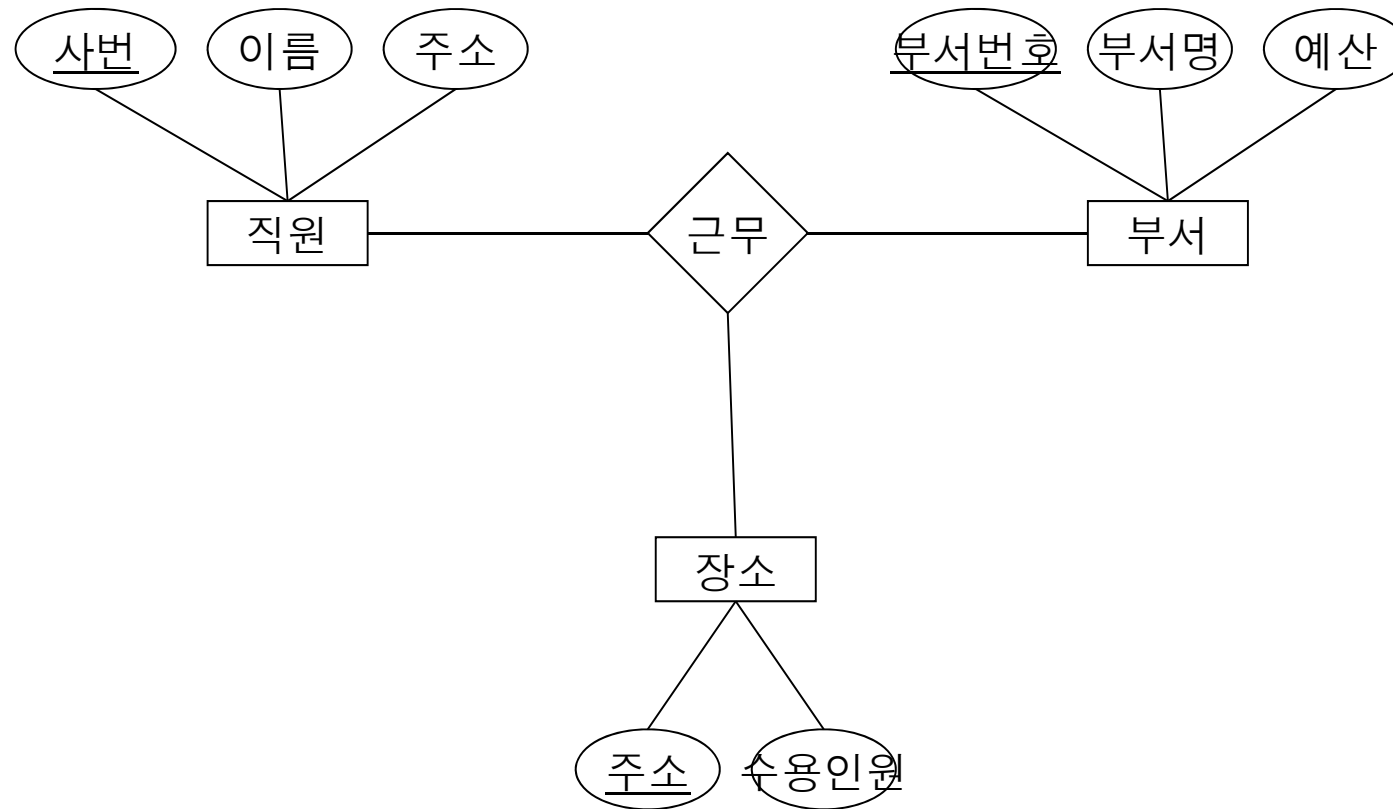


E-R 다이어그램의 특징

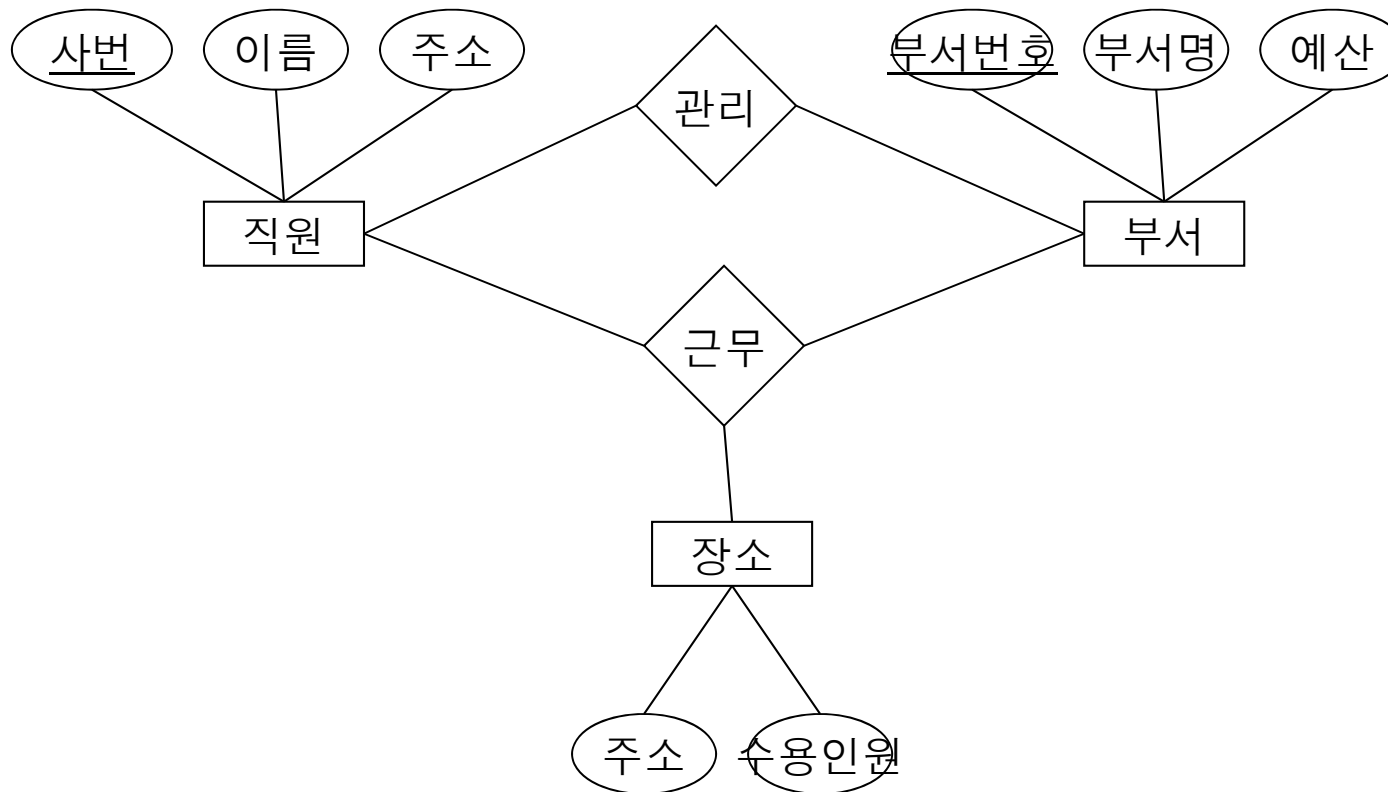
- 관계 타입도 속성(attribute)을 가질 수 있음
- 다대다(m:n) 관계 표현
- 다원 관계(n-ary relationship) 표현
 - 두개 이상의 개체 타입이 하나의 관계에 관련 가능
- 다중 관계(multiple relationship) 표현
 - 두 개체 타입 사이에 둘 이상의 관계가 존재 가능
- 존재 종속(existence dependency)
 - 어떤 객체 b의 존재가 객체 a의 존재에 달려있다.
 - 만일 객체 a를 삭제해야 한다면 객체 b도 삭제되어야 한다는 제약이 적용됨.
- 순환적 관계(recursive relationship)

□ Example: 다원 관계 (n-ary relationship)

– 삼진(ternary) 관계

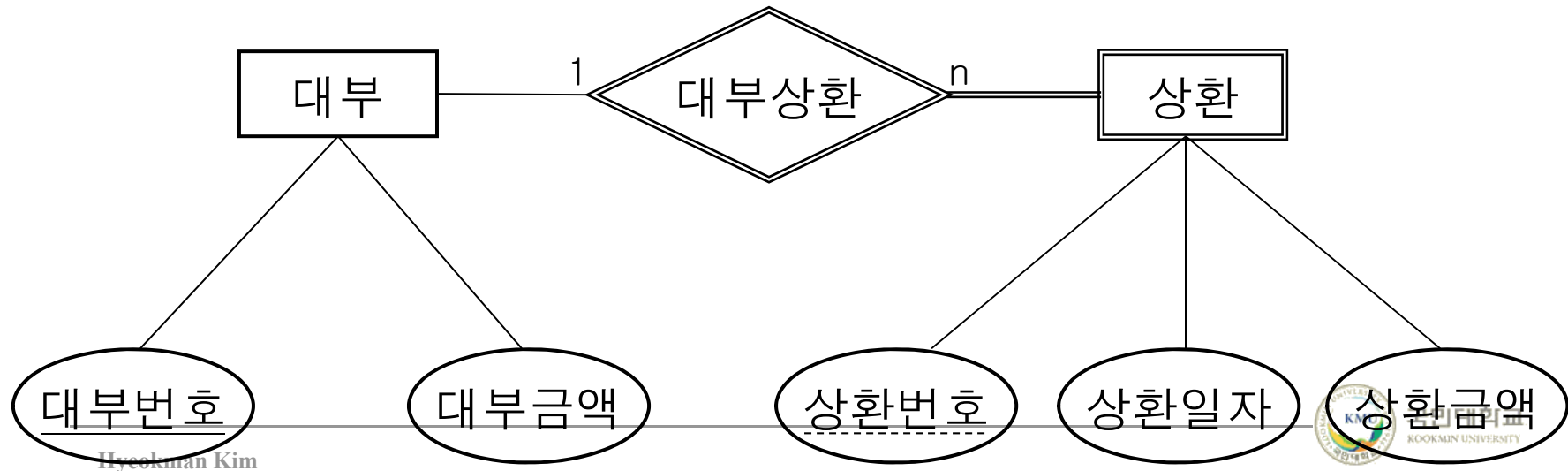


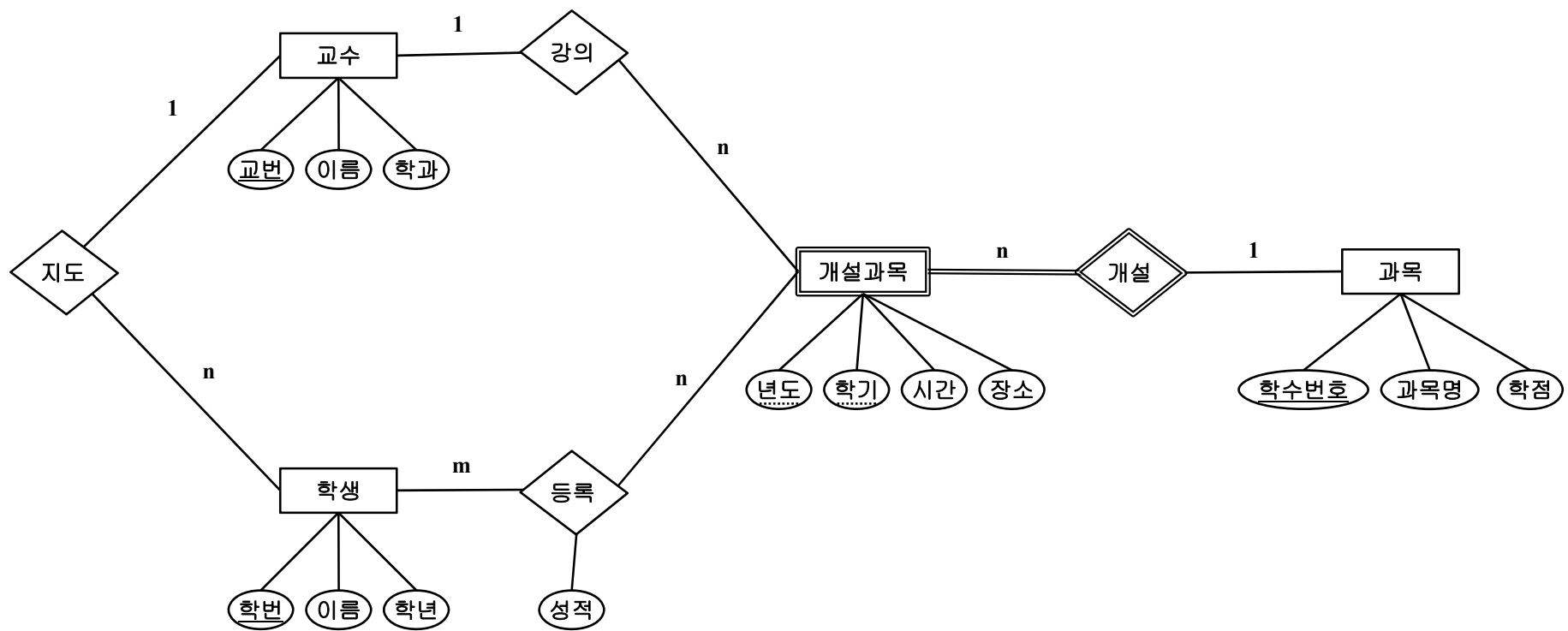
□ Example: 다중 관계 (multiple relationship)



□ 존재 종속(existence dependence)

- 개체 b의 존재가 개체 a의 존재에 좌우됨(b는 a에 존재 종속). 즉, 객체 a를 삭제하면, 객체 b도 함께 삭제되어야 함.
 - ◆ a : 주개체(dominant entity), b : 종속 개체(subordinate entity)
- Strong(regular/owner) entity type : 기본키가 있는 엔티티
- Weak entity type : 기본키는 없고, partial key만 있는 엔티티
- Identifying relationship : strong-weak 엔티티를 연결하는 릴레이션쉽, strong 엔티티의 기본키를 identifying attribute라 함.
- Weak entity type은 identifying relationship에 항상 전체 참여함.

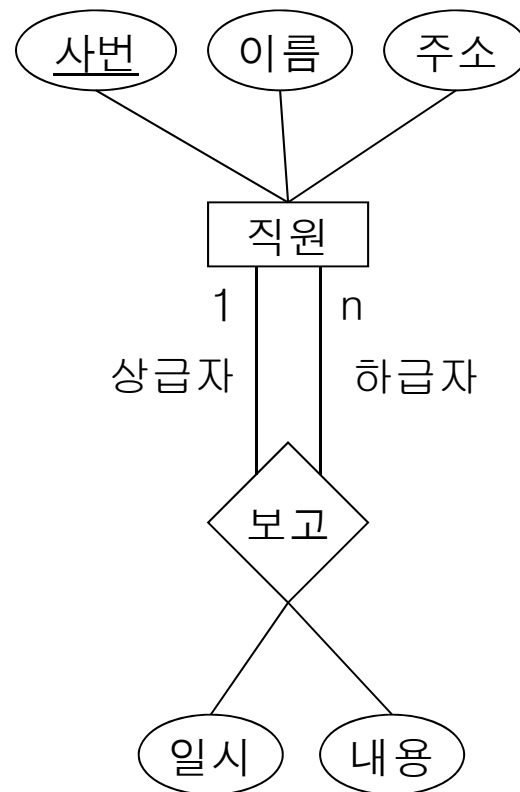




* “과목”을 “과목”과 “개설과목”으로 분리하고,
 * “개설과목”과 “개설”을 존재 종속으로 정의 !!

□ 순환적 관계 (recursive relationship)

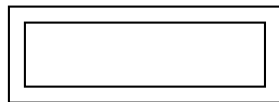
- 역할 이름(role name)



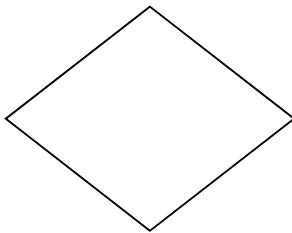
E-R 다이어그램 표기법



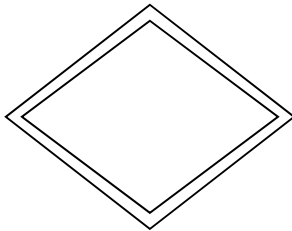
개체 타입



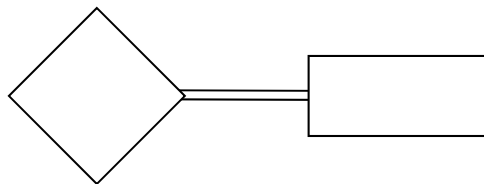
약한 개체 타입



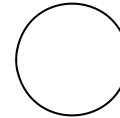
관계 타입



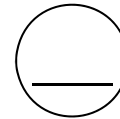
약한 관계 타입



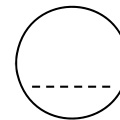
전체 참여 개체 타입



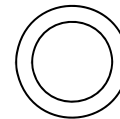
애트리뷰트



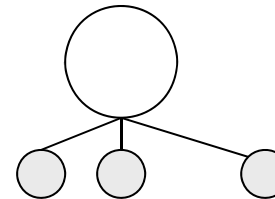
키 애트리뷰트



부분키 애트리뷰트



다중값 애트리뷰트



복합 애트리뷰트



유도 애트리뷰트

확장된 개체-관계 모델

- 확장된 개체-관계 모델(EER, Extended E-R model)
 - 복잡한 현실세계를 보다 정교하게 모델링
 - 주요 개념
 - ◆ 세분화(specialization)
 - ◆ 일반화(generalization)
 - ◆ 집단화(aggregation)
 - 복잡한 표기법 필요

5.6 논리적 데이터 모델

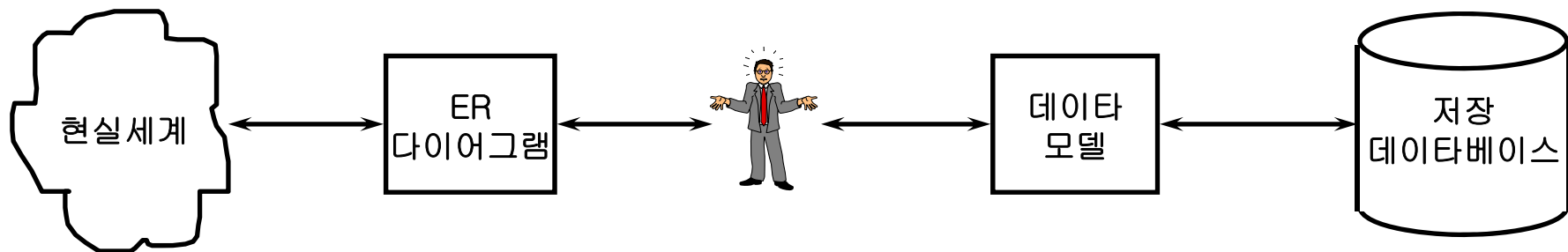
□ 개념적 데이터 모델

– 개체-관계 데이터 모델

- ◆ 현실 세계를 추상적으로 표현한 개념적 구조

□ 논리적 데이터 모델

- 개념적 구조를 저장 데이터베이스에 사상시키기 위해,
논리적 구조로 표현하기 위한 일련의 규칙



논리적 데이터 모델의 종류

- 관계 데이터 모델 (Relational Data Model)
- 계층 데이터 모델 (Hierarchical Data Model)
- 네트워크 데이터 모델 (Network Data Model)

5.6.1 관계 데이터 모델

□ 데이터베이스

- 테이블(릴레이션)의 집합
- 개체 릴레이션, 관계 릴레이션

□ 관계 스키마(Relation Scheme)

- 개체와 관계성의 테이블 정의 (검색 효율성을 고려안한 경우)

학생

<u>학번</u>	이름	주소	학년
-----------	----	----	----

 기본키(Primary Key)

교수

<u>교수번호</u>	교수 이름	전공	학과
-------------	-------	----	----

과목

<u>과목번호</u>	과목 이름	학점
-------------	-------	----

지도

<u>교수번호*</u>	<u>학번*</u>
--------------	------------

 외래키(Foreign Key)가 기본키가 됨

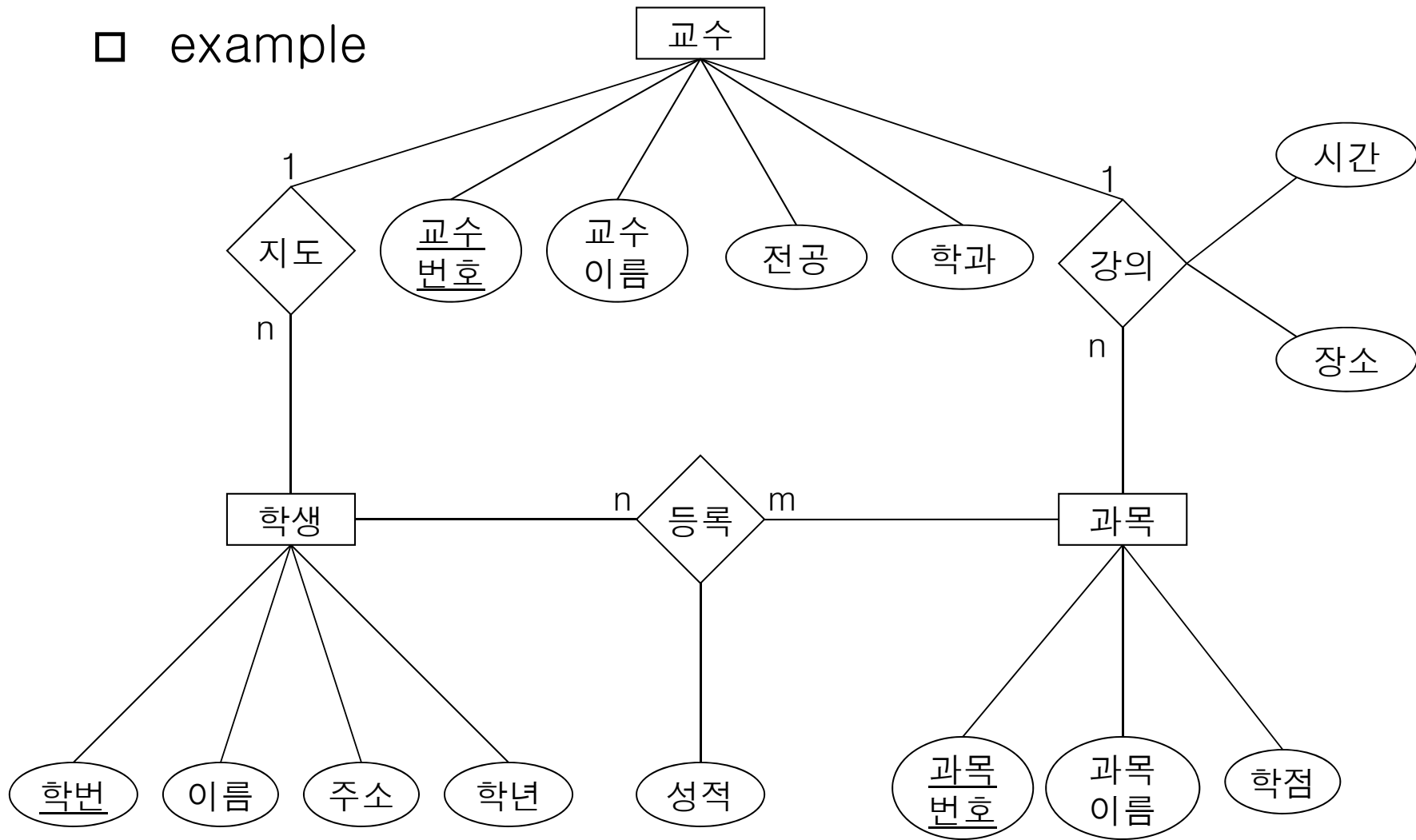
등록

<u>학번*</u>	<u>과목번호*</u>	성적
------------	--------------	----

강의

<u>교수번호*</u>	<u>과목번호*</u>	시간	장소
--------------	--------------	----	----

□ example



ER 다이어그램에서 관계 스키마로의 변환

□ 원칙

- Entity type을 릴레이션으로 변환 : entity relation
- Relationship을 릴레이션으로 변환 : relationship relation

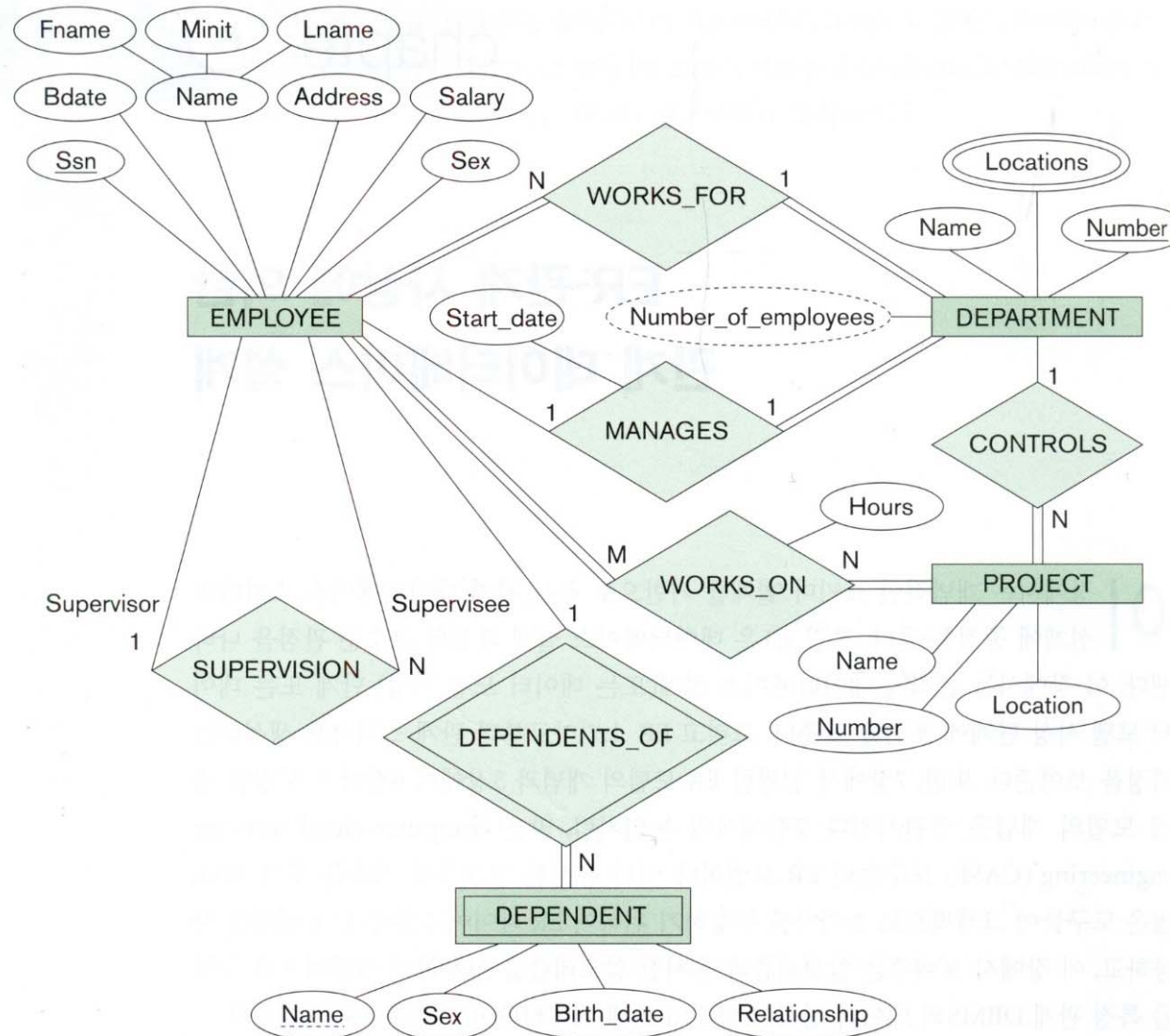
□ 검색 효율성을 고려

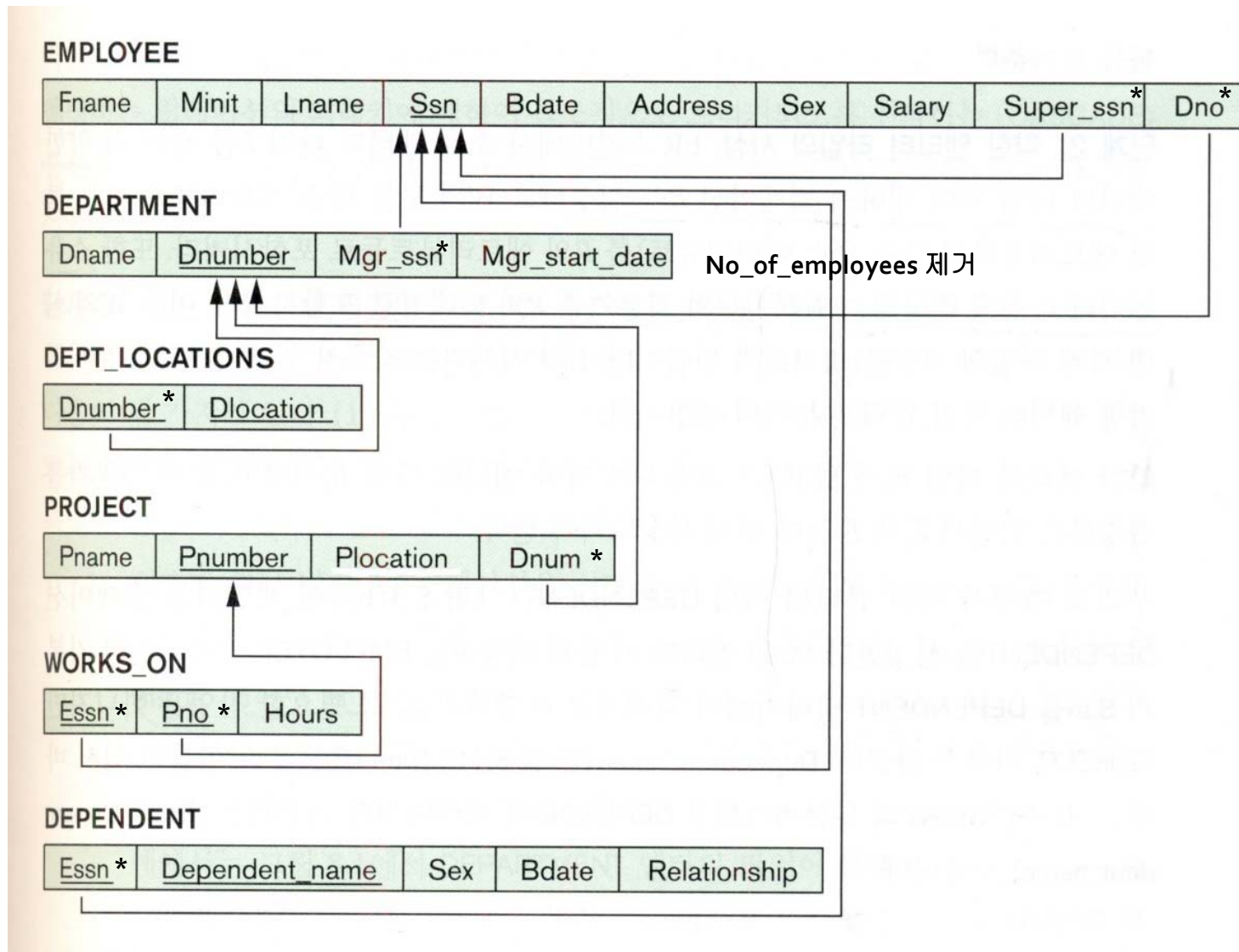
- 릴레이션의 수가 작을수록 검색 연산이 효율적임
 - ◆ 전체 릴레이션의 수를 줄이는 방향으로 변환

□ Note

- 관계 모델에서는 다중값 애트리뷰트를 허용하지 않음.

변환 예





<Entities>

EMPLOYEE (SSN, Bdate, Fname, Minit, Lname, Sex, Salary, Address,
DepartmentNumber*, SupervisorSSN*)

DEPARTMENT (Number, Name, ManagerSSN*, ManagesStartDate)

PROJECT (Number, Name, Location, ControllingDepartmentNumber*)

DEPENDENT (EmployeeSSN*, DependentName, Sex, BirthDate, Relationship)

<Relationships>

WORKS_ON (EmployeeSSN*, ProjectNumber*, Hours)

<Multi-valued attribute>

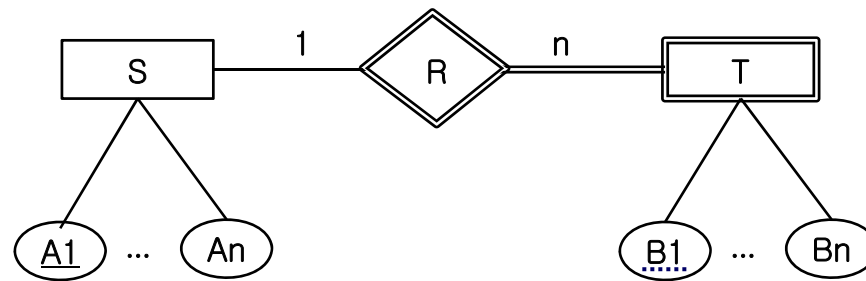
DEPARTMENT_LOCATIONS (DepartmentNumber*, Location)

변환 알고리즘

- 1. 정규 엔티티 타입(regular/strong entity type)은 릴레이션으로 생성. (엔티티 릴레이션)
 - 기본키는 정규 엔티티 타입의 키 애트리뷰트를 그대로 유지
 - 복합 애트리뷰트는 단순 애트리뷰트만 포함시킴.

- 2. 약한 엔티티 타입(weak entity type)은 릴레이션으로 생성. (엔티티 릴레이션)
 - Owner entity type의 키 애트리뷰트를 릴레이션에 포함시킴.
 - 기본키는 owner entity type의 기본키와 약한 엔티티 타입의 부분키 조합으로 구성함.

□ Rule 1 & 2



<u>A1</u>	A2	...	An

<u>A1*</u>	<u>B1</u>	B2	...	Bn

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary
-------	-------	-------	------------	-------	---------	-----	--------

DEPARTMENT

Dname	<u>Dnumber</u>
-------	----------------

No_of_employees 제거, Locations는 6번에서 별도 처리

PROJECT

Pname	<u>Pnumber</u>	Plocation
-------	----------------	-----------

DEPENDENT

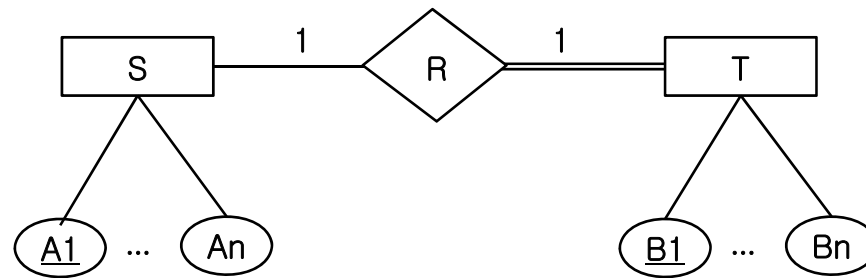
<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

-
- 3. 이진 1:1 관계 타입 R은 별도의 릴레이션을 생성하지 않고, 외래키를 추가함.
 - 한 엔티티 타입 T를 선택하여, 다른 엔티티 타입 S의 기본키를 T의 릴레이션에 애트리뷰트로 추가함. (외래키, 기본키는 아님)
 - R에 속한 애트리뷰트는 모두 T의 릴레이션에 포함시킴.
 - T 역할은 참여하는 두 엔티티 타입 모두 가능함. 그러나 두 엔티티 타입 중에 **완전 참여**가 있다면, 그 엔티티 타입을 T로 하는 것이 유리함. (널값의 최소화)
 - 4. 이진 1:n 관계 타입 R은 별도의 릴레이션을 생성하지 않고, 외래키를 추가함.
 - n-측의 엔티티 타입을 T라 할 때, S의 기본키를 T의 릴레이션에 애트리뷰트로 추가함. (외래키, 기본키는 아님)
 - R에 속한 애트리뷰트는 모두 T의 릴레이션에 포함시킴.

□ Note

- 3,4에서 R에 속한 애트리뷰트의 갯수가 많다면 (예를 들어 3 이상), 별도의 릴레이션으로 생성할 수도 있음 (관계 릴레이션)

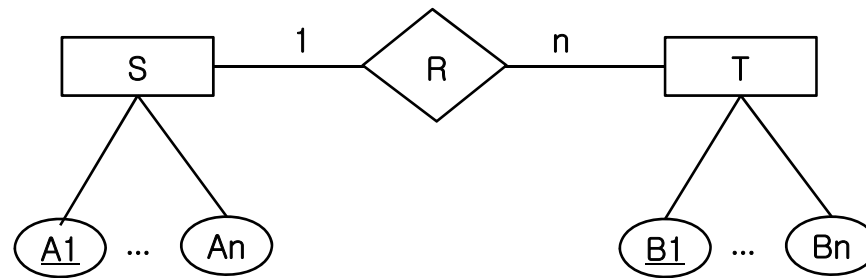
□ Rule 3



<u>A1</u>	A2	...	An

<u>B1</u>	B2	...	Bn	A1*

□ Rule 4



<u>A1</u>	A2	...	An

<u>B1</u>	B2	...	Bn	A1*

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn*	Dno*
-------	-------	-------	------------	-------	---------	-----	--------	------------	------

SUPERVISION WORKS_FOR

DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn*	Mgr_start_date
-------	----------------	----------	----------------

No_of_employees 제거

MANAGES

PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum*
-------	----------------	-----------	-------

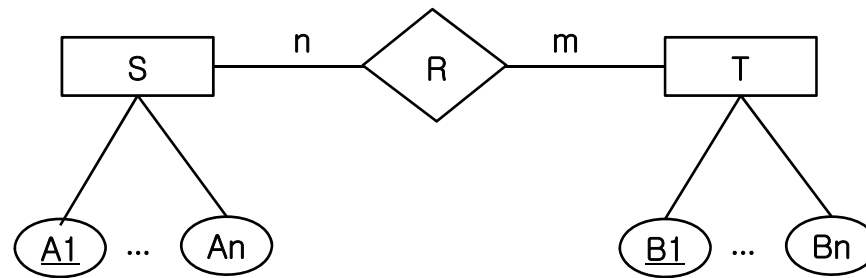
CONTROLS

-
- 5. 이진 n:m 관계 타입 R은 별도의 릴레이션으로 생성. (관계 릴레이션)
- 양 쪽 엔티티 타입 S와 T의 키 애트리뷰트를 R의 릴레이션에 애트리뷰트로 추가함. (외래키, 기본키로 설정)
 - R에 속한 애트리뷰트는 모두 R의 릴레이션에 포함시킴.
 - 기본키는 S와 T의 키 애트리뷰트 조합으로 구성함.

WORKS_ON

<u>Essn</u> *	<u>Pno</u> *	Hours
---------------	--------------	-------

□ Rule 5



<u>A1</u>	A2	...	An

<u>A1</u> *	<u>B1</u> *

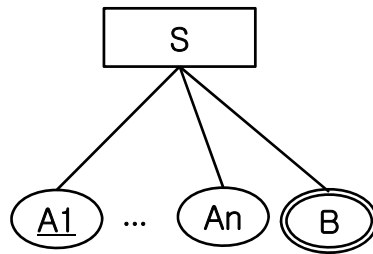
<u>B1</u>	B2	...	Bn

-
- 6. 다중값 애트리뷰트 A는 별도의 릴레이션으로 생성. (애트리뷰트 릴레이션)
- 별도의 릴레이션은 다중값 애트리뷰트 A와 A가 속한 엔티티 타입의 키 애트리뷰트 K로 구성됨.
 - 기본키는 A와 K로 구성함.
 - K는 외래키임.

DEPT_LOCATIONS

<u>Dnumber</u> *	<u>Dlocation</u>
------------------	------------------

□ Rule 6

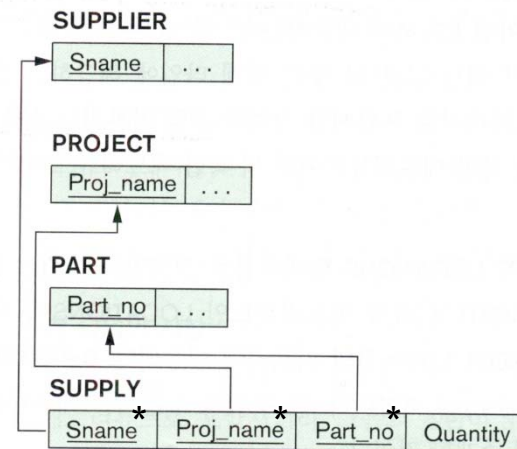
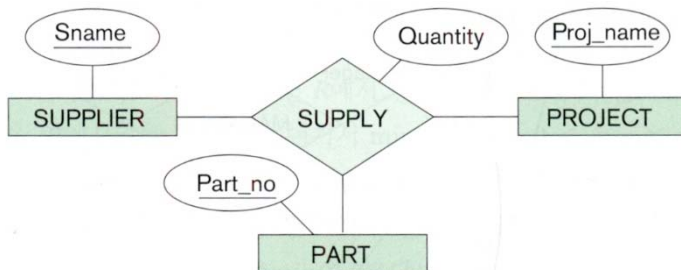


<u>A1</u>	A2	...	An

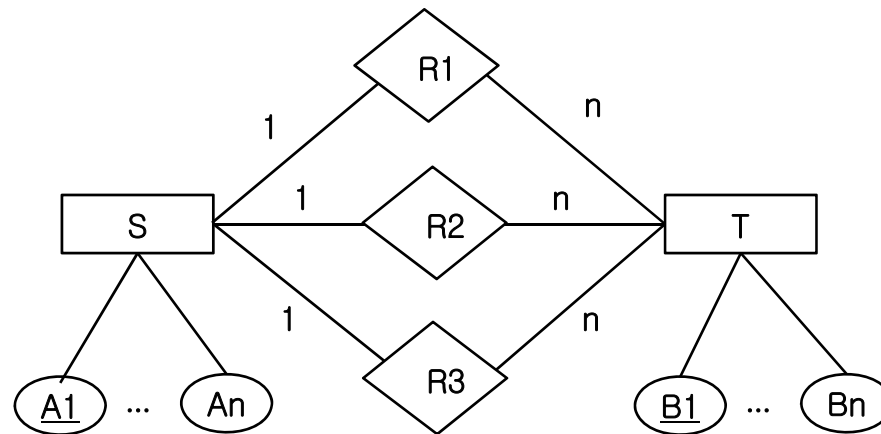
<u>A1</u> *	<u>B</u>

□ 7. 3차 이상의 다원 관계 타입 R은 별도의 릴레이션으로 생성. (관계 릴레이션)

- 모든 참여 엔티티 타입의 키 애트리뷰트를 R의 릴레이션에 포함시킴.
- R에 속한 애트리뷰트는 모두 R의 릴레이션에 포함시킴.
- 기본키는 참여 엔티티 타입의 키 애트리뷰트 조합으로 구성함.

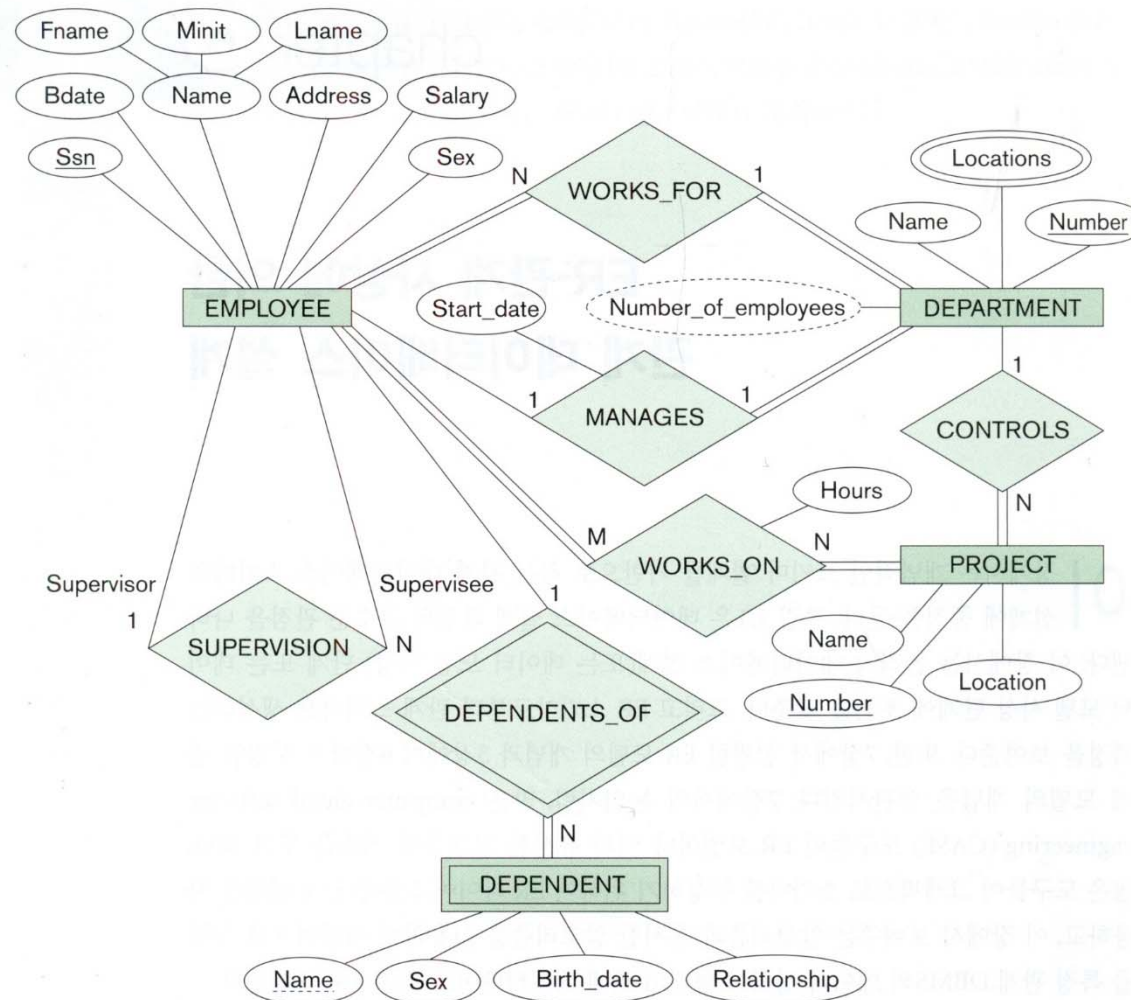


□ Example

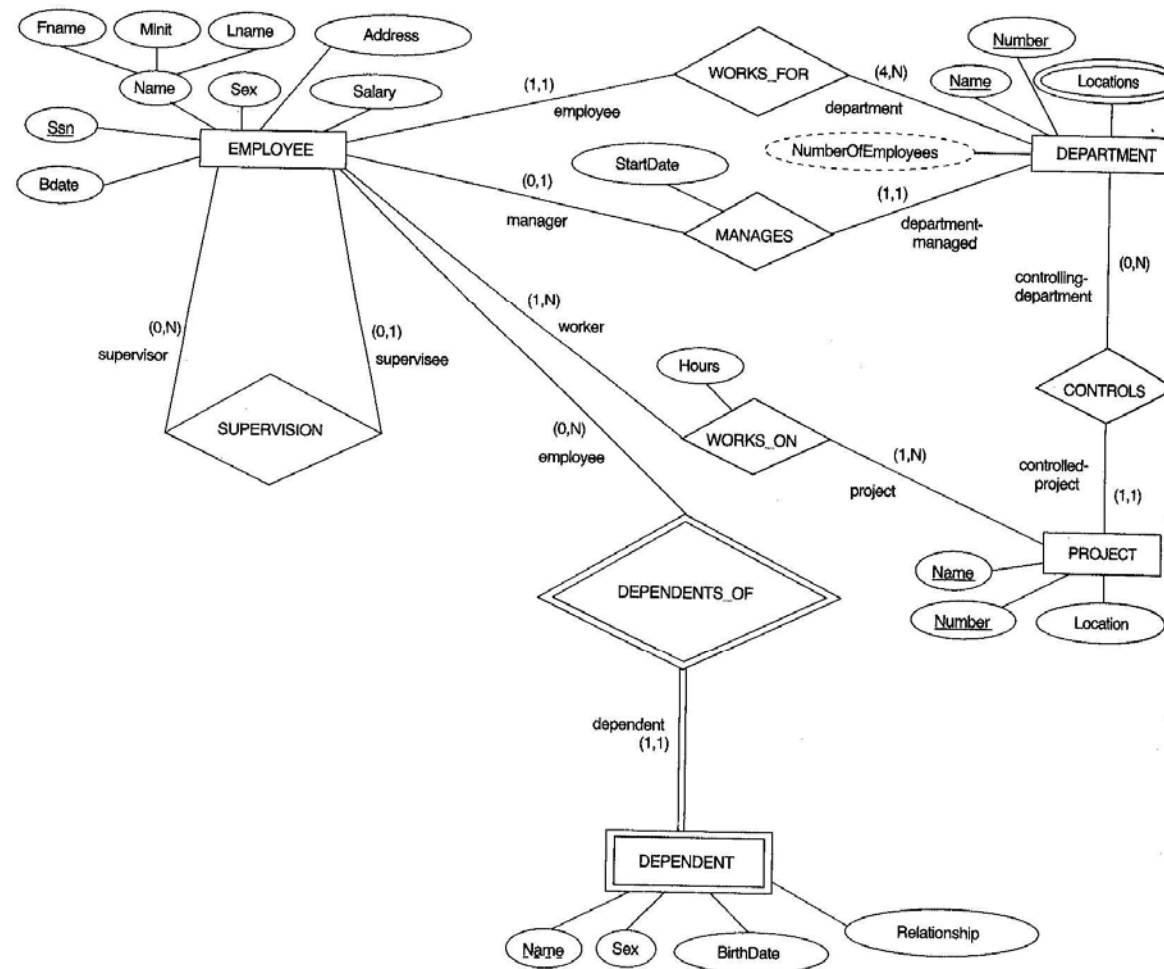


E-R 다이어그램의 다른 표기법

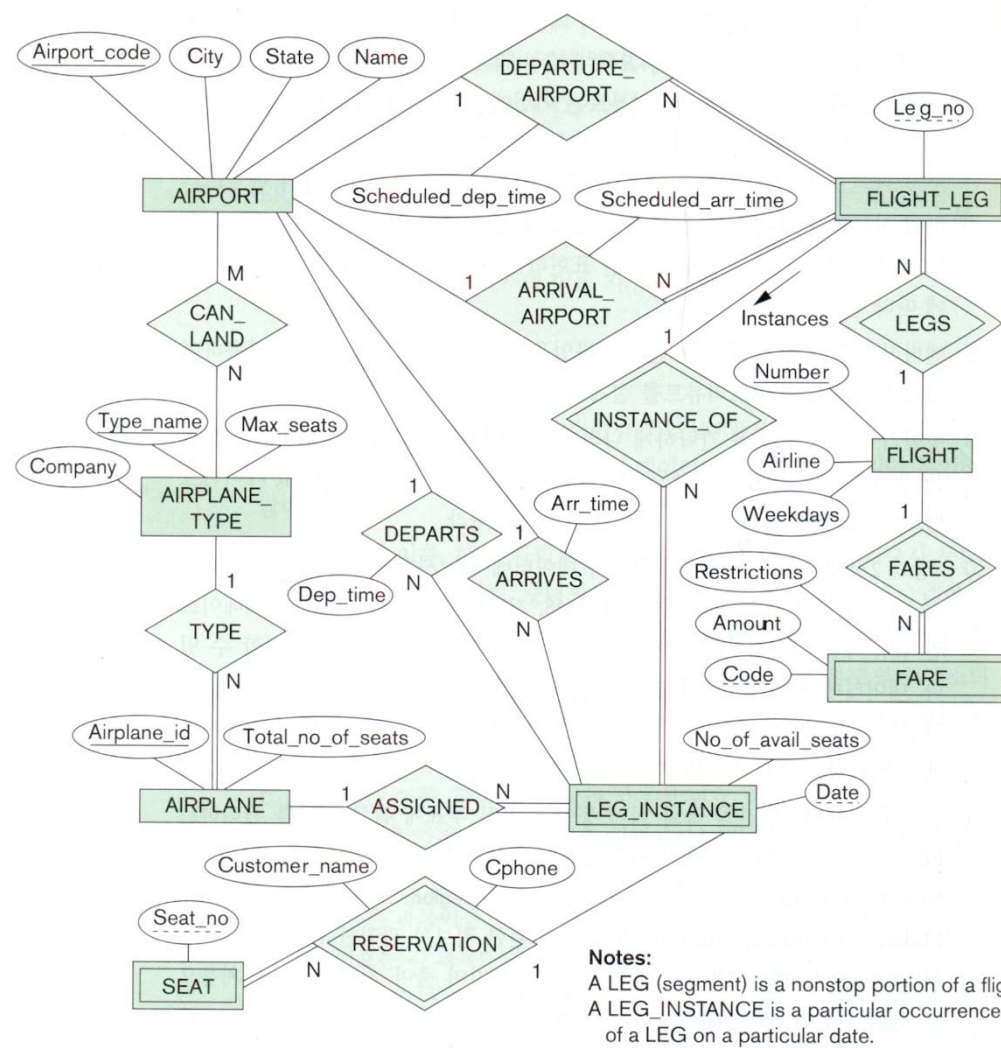
□ 단일선/이중선 표기법 - Relationship 기준



□ (min,max) 표기법 – Entity type 기준

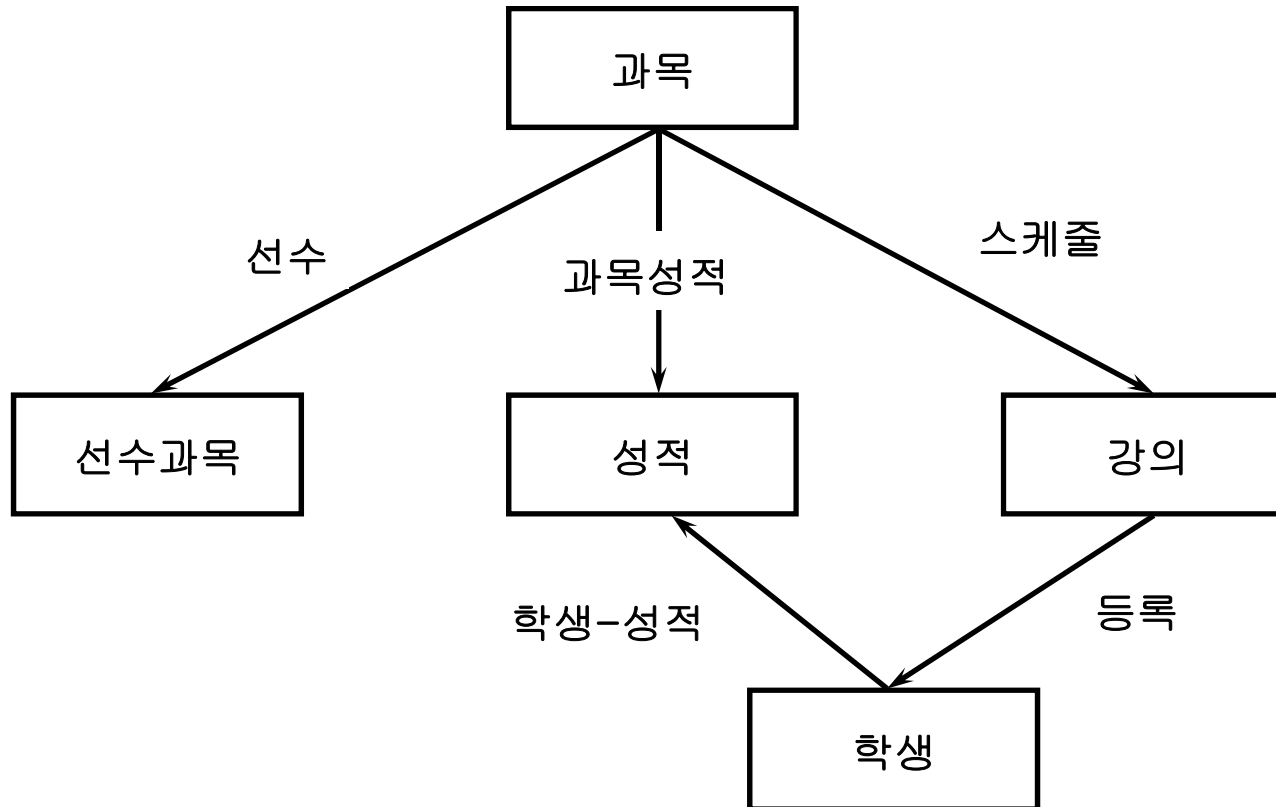


E-R 다이어그램의 예



Note: 자료 구조도

- Bachman 다이어그램 (1969)
- 레코드 타입 간의 관계에 대한 도형적 표현
 - 화살표: 레코드 타입간의 일대다(1:n) 관계를 의미



□ 구성 요소

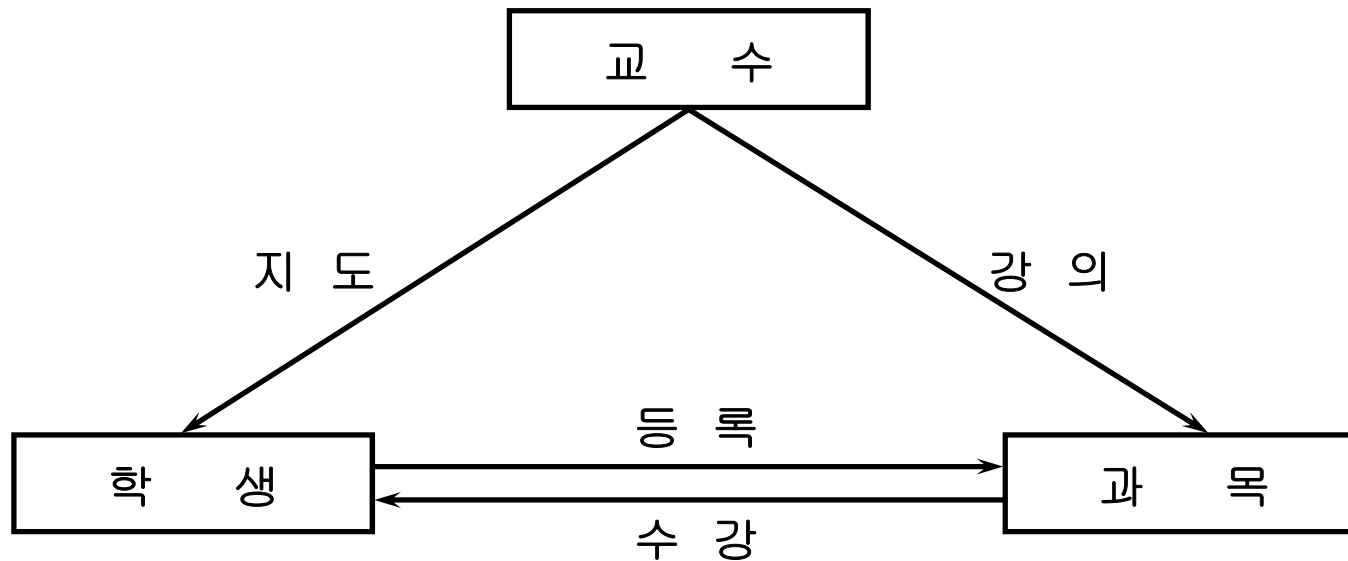
- 사각형 노드 : 레코드 타입 (개체 타입)
- 직선(link;arc) : 레코드 타입 간의 일대다 (1:n) 관계
- 레이블 : 관계 이름

□ 스키마 다이어그램으로 사용 가능

- 데이터베이스의 논리적 구조를 표현한 데이터 구조도, 즉 스키마의 도형적 표현
- 트리 형태 : 계층 데이터 모델
- 그래프 형태 : 네트워크 데이터 모델

5.6.2 네트워크 데이터 모델

- 스키마 다이어그램이 Network(그래프)
- 허용되는 레코드 타입, 관계성을 명시
- Owner-member 관계
 - 두 레코드 타입간의 1:n 관계



5.6.3 계층 데이터 모델

- 스키마 다이어그램이 트리
- No Cycle
- 루트 레코드, 자식 레코드, 레벨
- 자식-부모(Parent-Child) 관계
 - 1:n 관계의 두 레코드 타입

