



Chapter 7: Entity-Relationship (E-R) Model – Part 1

Revision by Gun-Woo Kim

Dept. of Computer Science and Engineering
Hanyang University

Database System Concepts, 6th Ed.

©Silberschatz, Korth and Sudarshan
See www.db-book.com for conditions on re-use



Contents

- 7.1 Overview of Design Process
- 7.2 Entity-Relationship Modeling
- 7.3 Constraints
- 7.4 Removing Redundant Attributes in Entity Sets
- 7.5 Entity-Relationship Diagram (E-R Diagram)

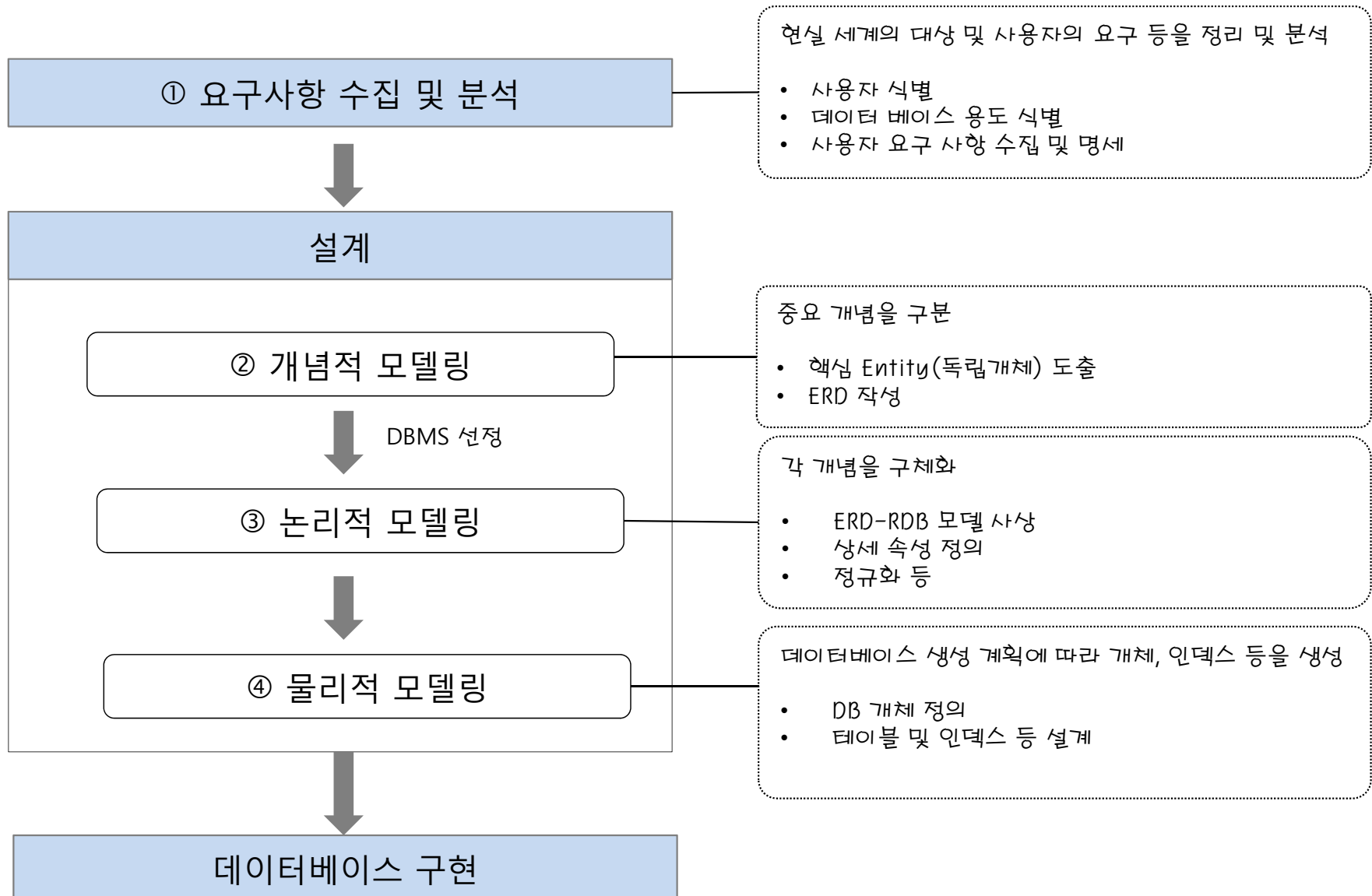


7.1 Overview of Design Process

- Design phases
 - Specification of user requirements
 - Conceptual design
 - ▶ Provide a detailed overview of the enterprise
 - ▶ Use **entity-relationship model** to represent the conceptual design
 - ▶ Its result is a graphic representation of the schema (E-R diagram)
 - Logical design
 - ▶ Map the conceptual schema onto the implementation data model of the database system that will be used
 - Physical design
 - ▶ The physical feature of the database such as file organization and choice of **index structure** are specified



7.1 Overview of Design Process



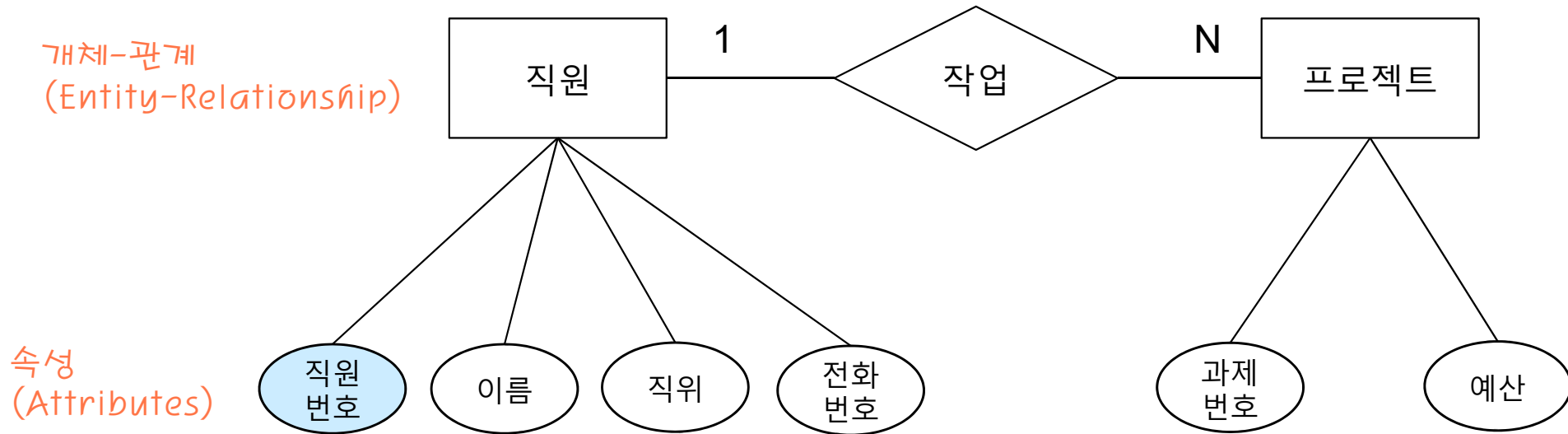


Example of E-R Diagram

■ ER 다이어그램이란?

- ER 모델은 개체 (Entity)와 개체 간의 관계(Relationship)를 표준화된 그림으로 나타냄.

1명의 직원이 다수의 프로젝트를 가질수있다는 말, 즉 1대 다 의 관계





7.2 Entity-Relationship Model - Entity Sets

- A *database* can be modeled as:
 - a collection of entities,
 - relationship among entities.
- An **entity** is an object that exists and is distinguishable from other objects.
 - Example: specific person, company, event, plant
- Entities have **attributes**
 - Example: people have *names* and *addresses*
- An **entity set** is a set of entities of the same type that share the same properties.
 - Example: set of all persons, companies, trees, holidays



Entity Sets (Cont.)

instructor_ID instructor_name

76766	Crick
45565	Katz
10101	Srinivasan
98345	Kim
76543	Singh
22222	Einstein

instructor

student-ID student_name

98988	Tanaka
12345	Shankar
00128	Zhang
76543	Brown
76653	Aoi
23121	Chavez
44553	Peltier

student



Relationship Sets

- A **relationship** is an association among several entities

Example:

44553 (Peltier)	<u>advisor</u>	22222 (<u>Einstein</u>)
<i>student</i> entity	relationship set	<i>instructor</i> entity

- A **relationship set** is a mathematical relation among $n \geq 2$ entities, each taken from entity sets

$$\{(e_1, e_2, \dots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$$

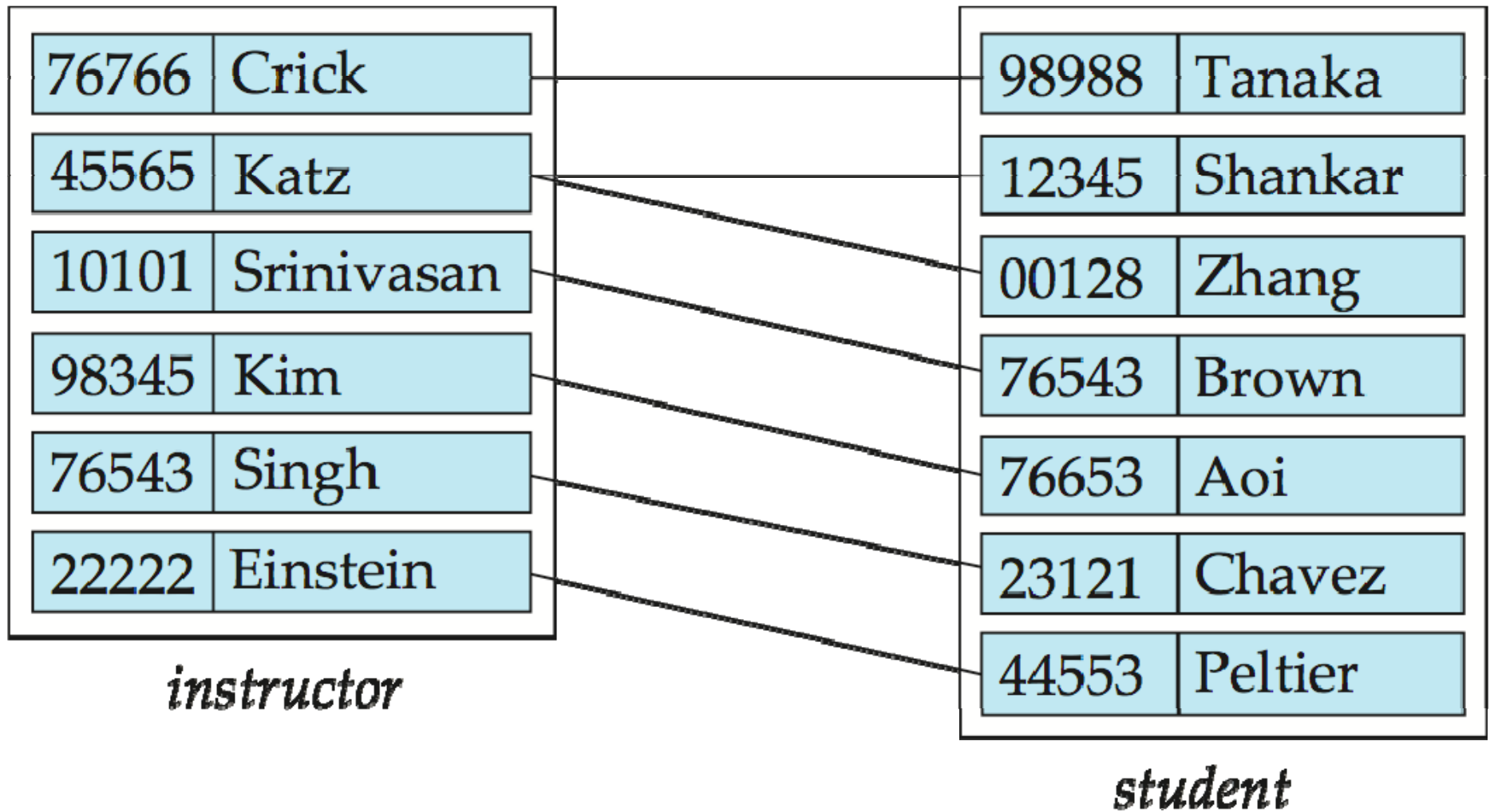
where (e_1, e_2, \dots, e_n) is a relationship

- Example:

$$(44553, 22222) \in \text{advisor}$$



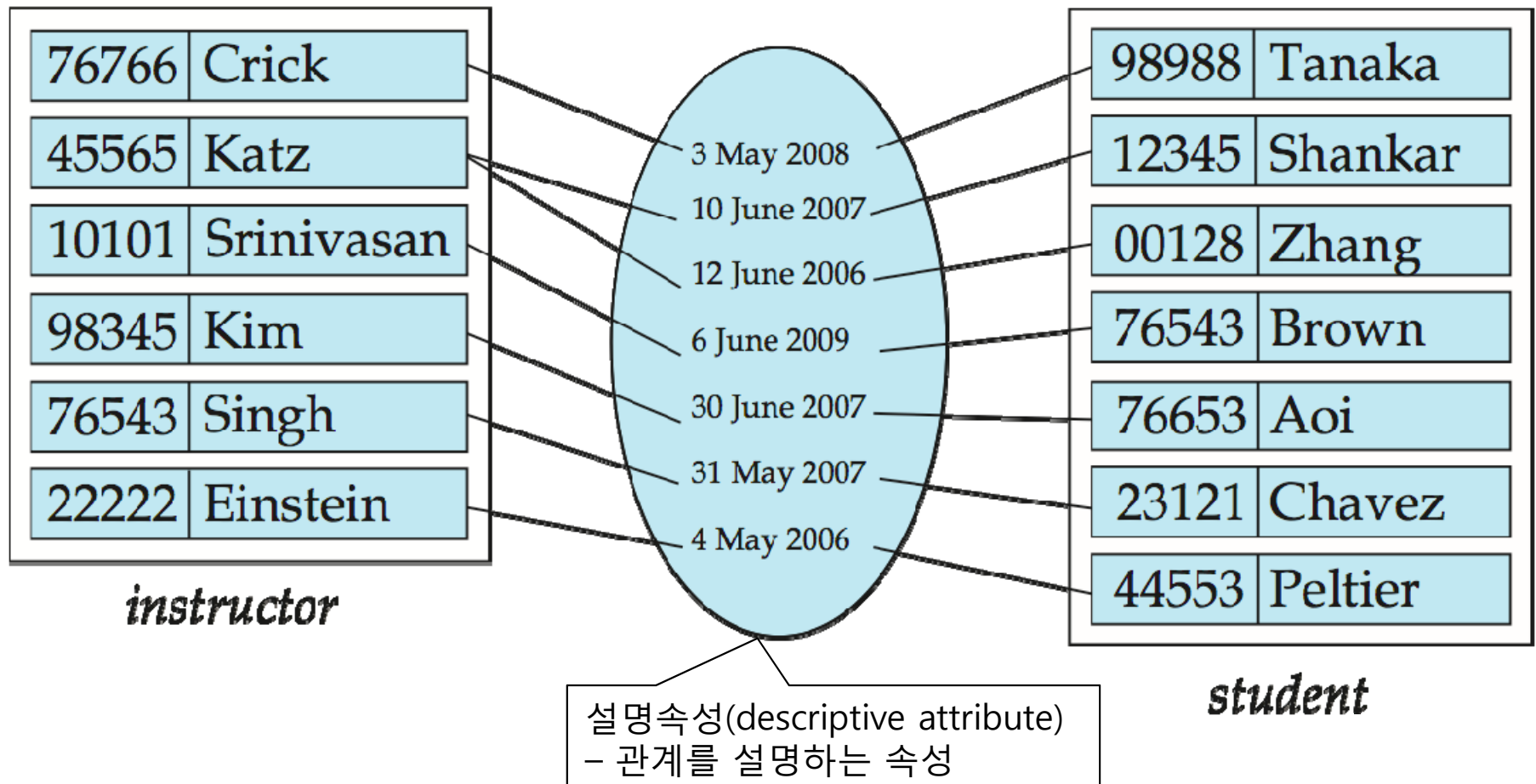
Relationship Set (Cont.)





Relationship Sets (Cont.)

- An **attribute** can also be property of a relationship set.
- For instance, the *advisor* relationship set between entity sets *instructor* and *student* may have the attribute *date* which tracks when the student started being associated with the advisor


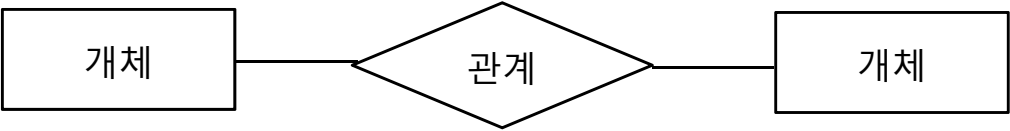
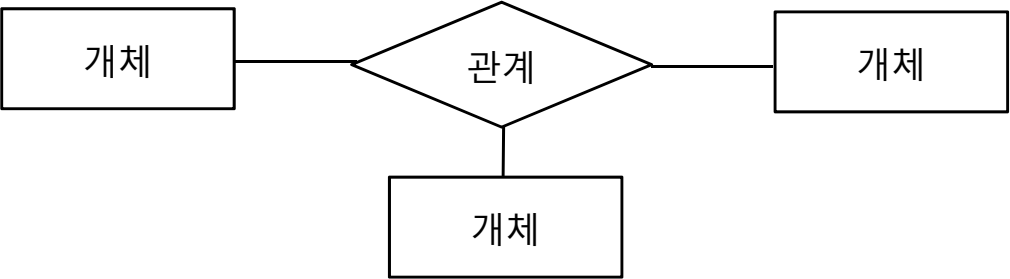




Degree of a Relationship Set

■ 차수에 따른 유형

관계 집합에 참가하는 개체 타입의 수를 관계 타입의 차수(degree)라고 함.

기호	의미	설명
	1진 관계	한 개의 개체가 자기 자신과 관계를 맺음
	2진 관계	두 개의 개체가 관계를 맺음
	3진 관계	세 개의 개체가 관계를 맺음

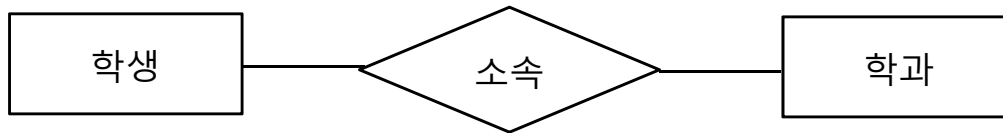


Degree of a Relationship Set (Cont.)

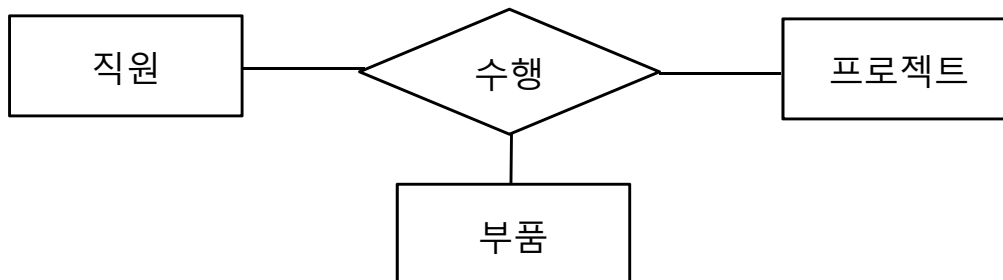
- ① 1진 관계(recursive relationship) : 한 개의 개체가 자기 자신과 관계를 맺는 경우



- ② 2진 관계(binary relationship) : 두 개의 개체가 관계를 맺는 경우



- ③ 3진 관계(ternary relationship) : 세 개의 개체가 관계를 맺는 경우





Attributes

- An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.

- Example:

instructor = (ID, name, street, city, salary)

course = (course_id, title, credits)

- **Domain** – the set of permitted values for each attribute

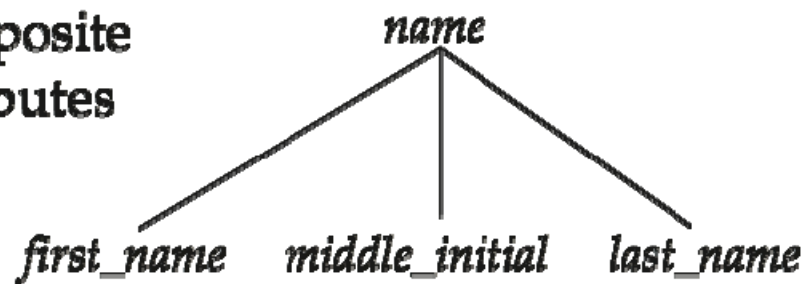
- Attribute types:

- **Simple** and **composite** attributes. (단순, 복합) 더이상 분해할수 있느냐 없느냐로 simple과 composite를 나눔
- **Single-valued** and **multivalued** attributes (단일-값, 다중-값)
 - ▶ Example: multivalued attribute: *phone_numbers* (집, 사무실, 핸드폰)
- **Derived** attributes (유도된 속성)
 - ▶ Can be computed from other attributes
 - ▶ Example: age, given date_of_birth

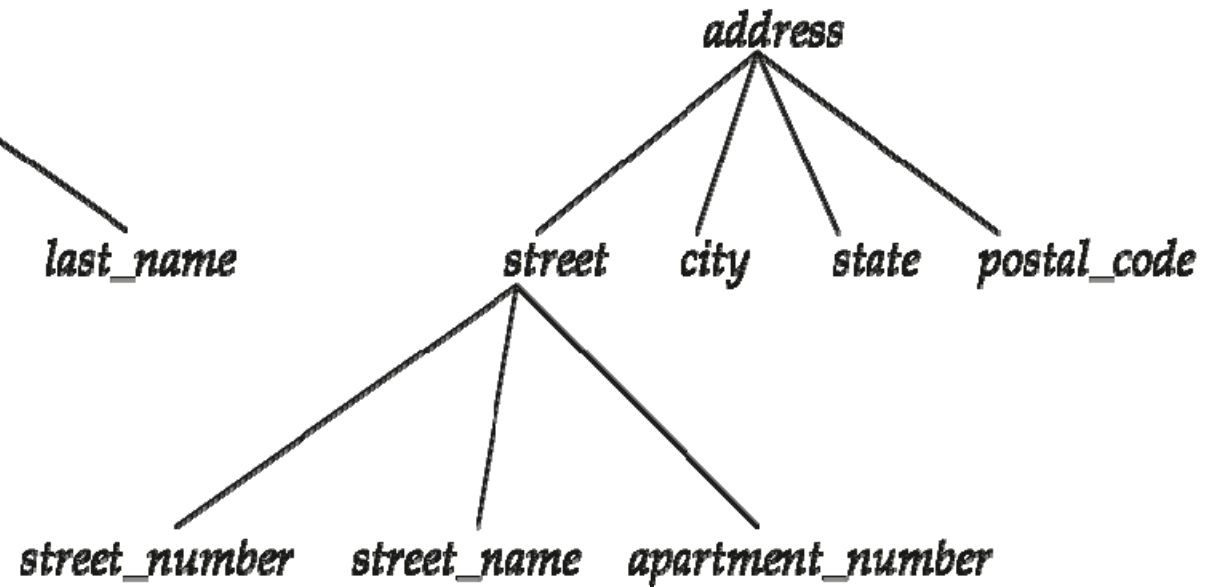


Composite Attributes

composite
attributes



component
attributes





7.3 Constraints – Mapping Cardinalities

대응수

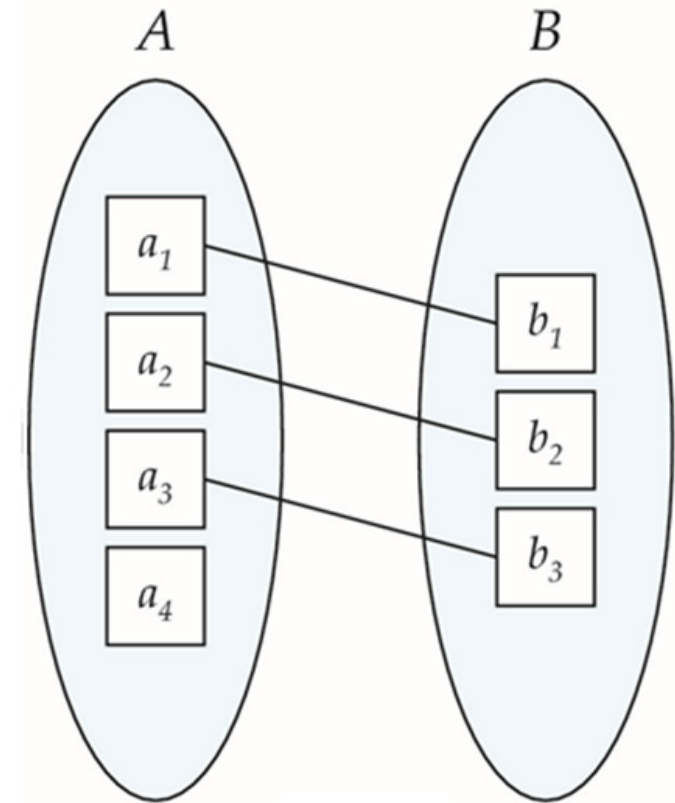
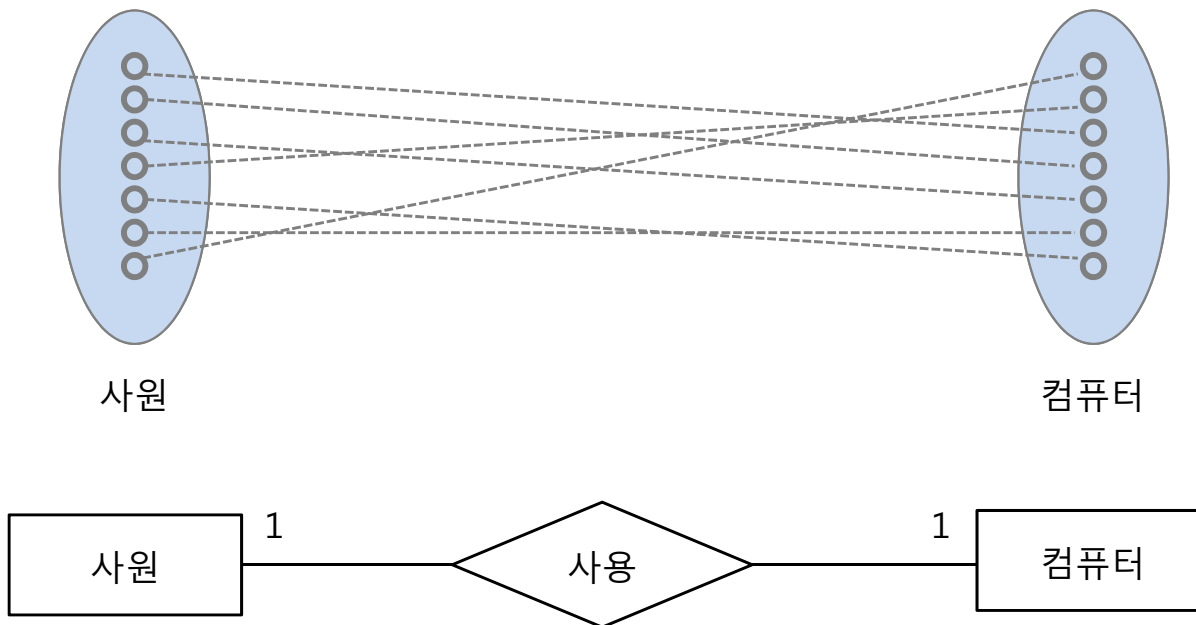
- Express the number of entities to which another entity can be associated via a relationship set.
- Most useful in describing binary relationship sets.
- For a binary relationship set the mapping cardinality must be one of the following types:
 - One to one
 - One to many
 - Many to one
 - Many to many



Mapping Cardinalities (Cont.)

■ One to One:

- 좌측 개체 타입에 포함된 개체가 우측 개체 타입에 포함된 개체와 일대일로 대응하는 관계

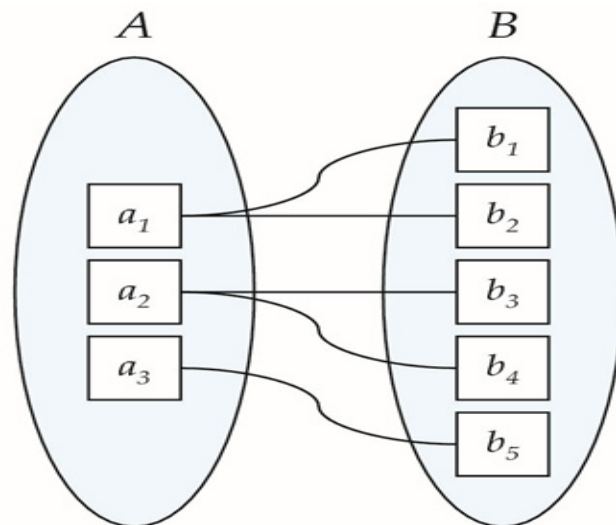
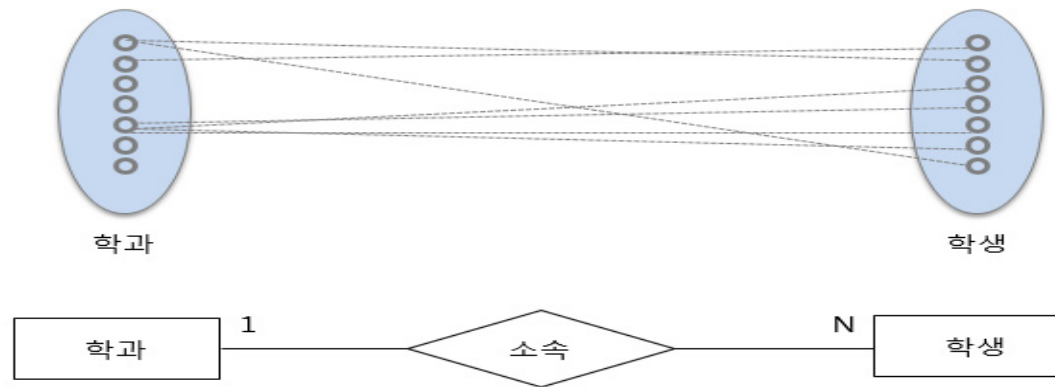




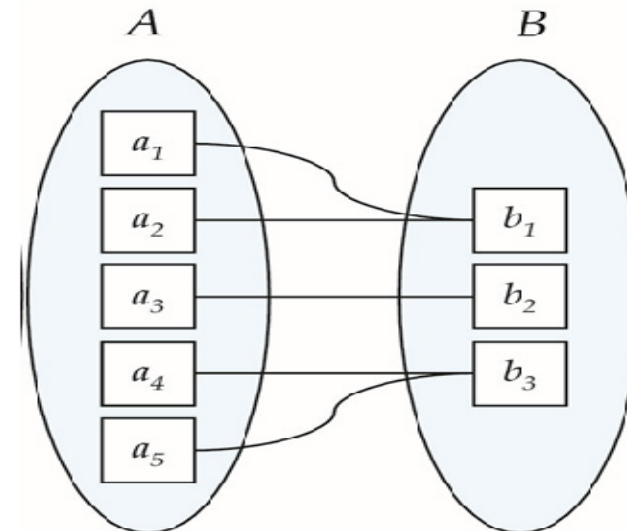
Mapping Cardinalities (Cont.)

■ One to Many or Many to One:

- 실제 일상생활에서 가장 많이 볼 수 있는 관계로, 한쪽 개체 타입의 개체 하나가 다른 쪽 개체 타입의 여러 개체와 관계를 맺음.



One to many



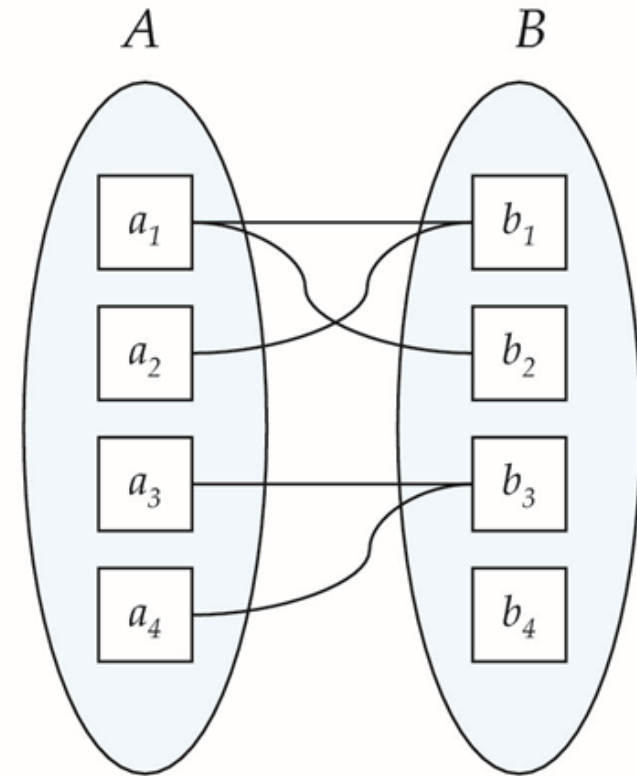
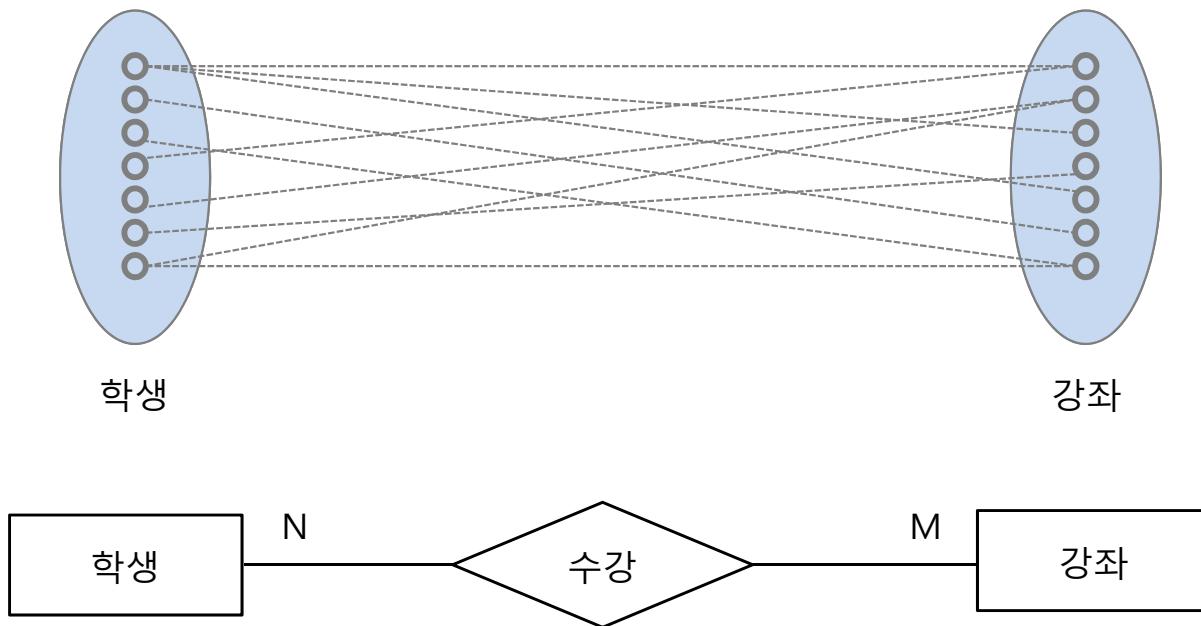
Many to One



Mapping Cardinalities (Cont.)

■ Many to Many: 요약하면 N:M관계라고 할 수 있음

- 각 개체 타입의 개체들이 서로 임의의 개수의 개체들과 서로 복합적인 관계를 맺고 있는 관계를 말함





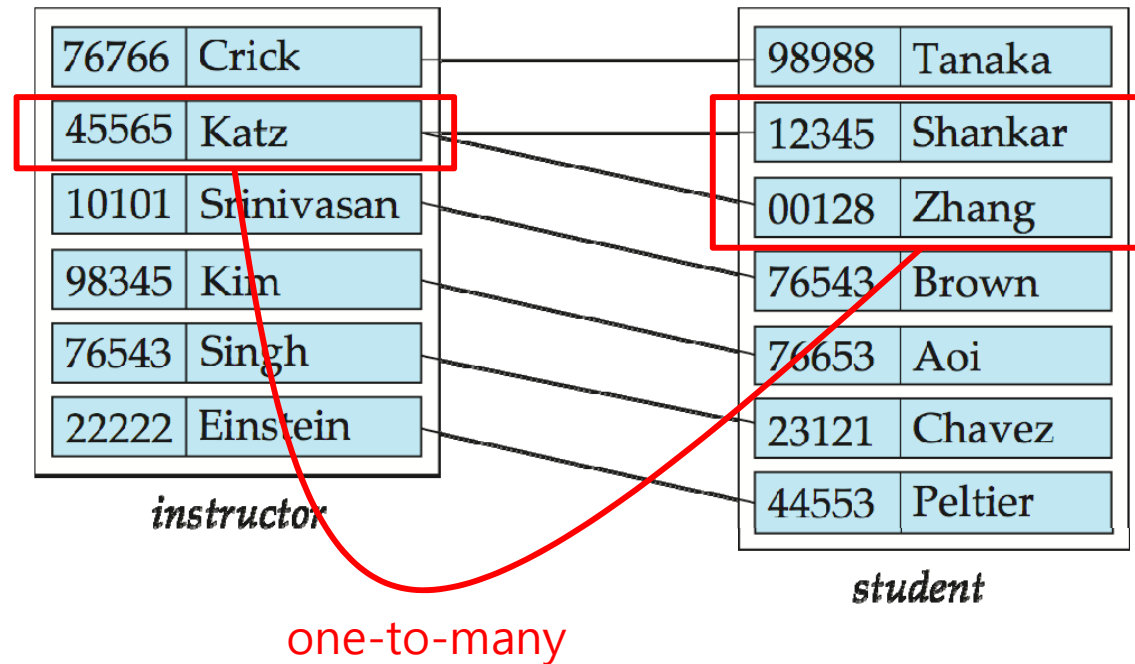
Keys

- A **super key** of an entity set is a set of one or more attributes whose values uniquely determine each entity.
- A **candidate key** of an entity set is a minimal super key
 - *ID* is candidate key of *instructor*
 - *course_id* is candidate key of *course*
- Although several candidate keys may exist, one of the candidate keys is selected to be the **primary key**.



Keys for Relationship Sets

- The combination of primary keys of the participating entity sets forms a super key of a relationship set.
 - (s_id, i_id) is the super key of *advisor*
- Must consider the mapping cardinality of the relationship set when deciding what are the candidate keys
 - If the relationship of *student-to-instructor* is
 - ▶ many-to-one, the primary key : (s_id)
 - ▶ one-to-many, the primary key : (i_id)
 - ▶ one-to-one, the primary key : (s_id) or (i_id)
 - ▶ many-to-many, the primary key : (s_id, i_id)
- Need to consider semantics of relationship set in selecting the *primary key* in case of more than one candidate key
 - 왜 그럴까? → 실제 관계 relation을 잘 생각해 보면 당연!



instructor-to-student

i_id	s_id
76766	98988
45565	12345
45565	00128
10101	76543
.....

결국, many측 참여 개체의 Primary key가 관계 릴레이션의 Primary key로 사용됨



7.4 Removing Redundant Attributes

- Suppose we have entity sets
 - *instructor*, with attributes including *dept_name*
 - *department*and a relationship
 - *inst_dept* relating *instructor* and *department*
- Attribute *dept_name* in entity *instructor* is redundant since there is an explicit relationship *inst_dept* which relates instructors to departments
 - The attribute replicates information present in the relationship, and should be removed from *instructor*
 - BUT: when converting back to tables, in some cases the attribute gets reintroduced, as we will see.



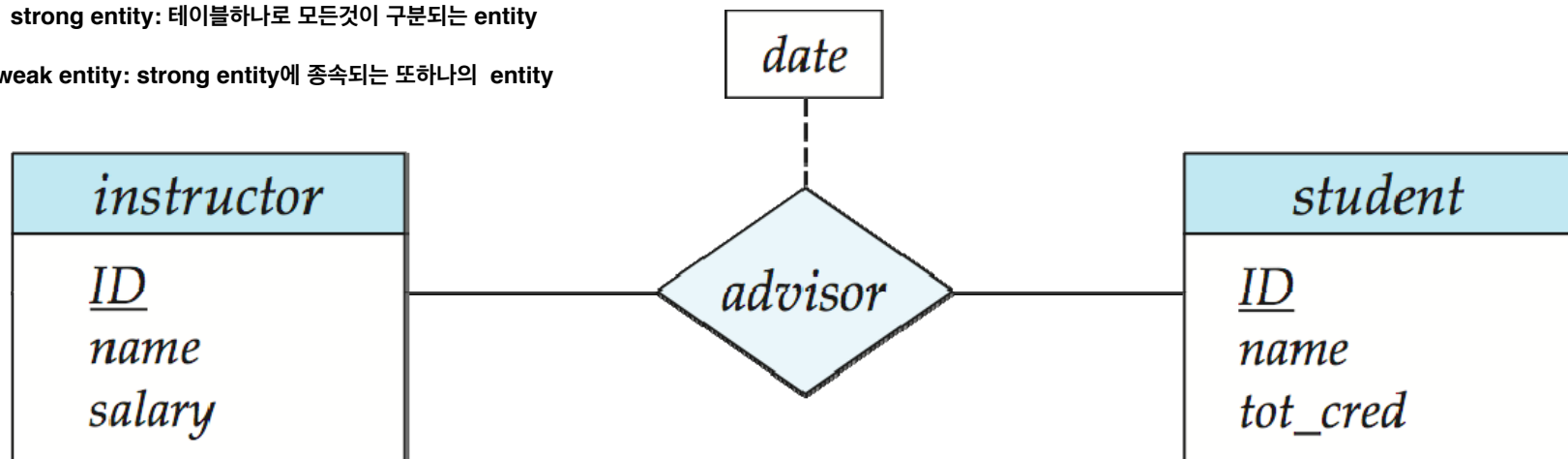
7.5 Entity-Relationship Diagrams

■ Basic structure

- **Rectangle** represent entity sets
- **Diamonds** represent relationship sets
- **Attributes** listed inside entity rectangle
- **Underline** indicates primary key attributes
- **Lines** link entity sets to relationship sets
- **Dashed lines** link attributes of a relationship set to the relationship set
- **Double lines** indicate total participation of an entity in a relationship set
- **Double diamonds** represent **identifying relationship sets** linked to **weak entity sets**

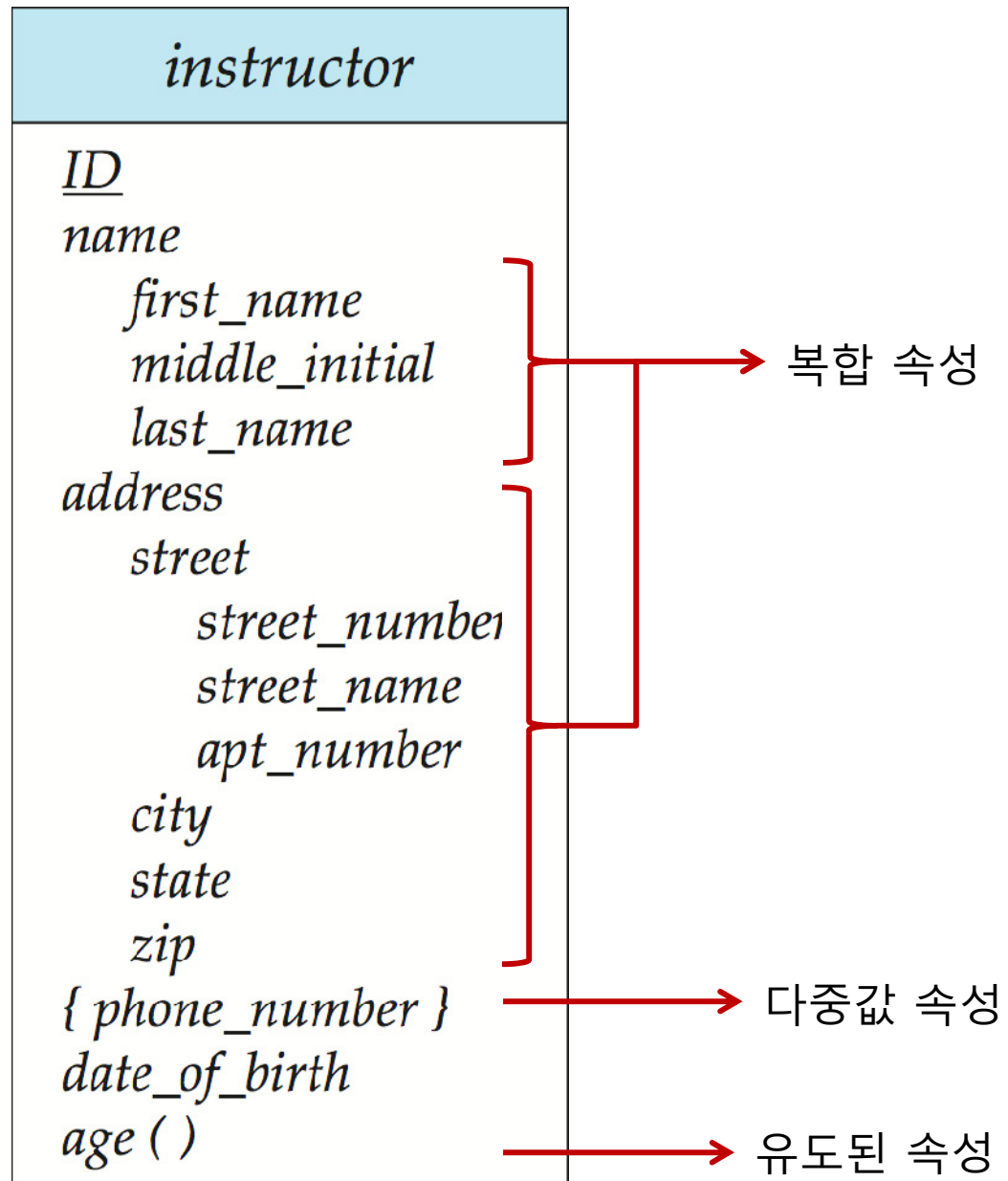
strong entity: 테이블 하나로 모든것이 구분되는 entity

weak entity: strong entity에 종속되는 또하나의 entity





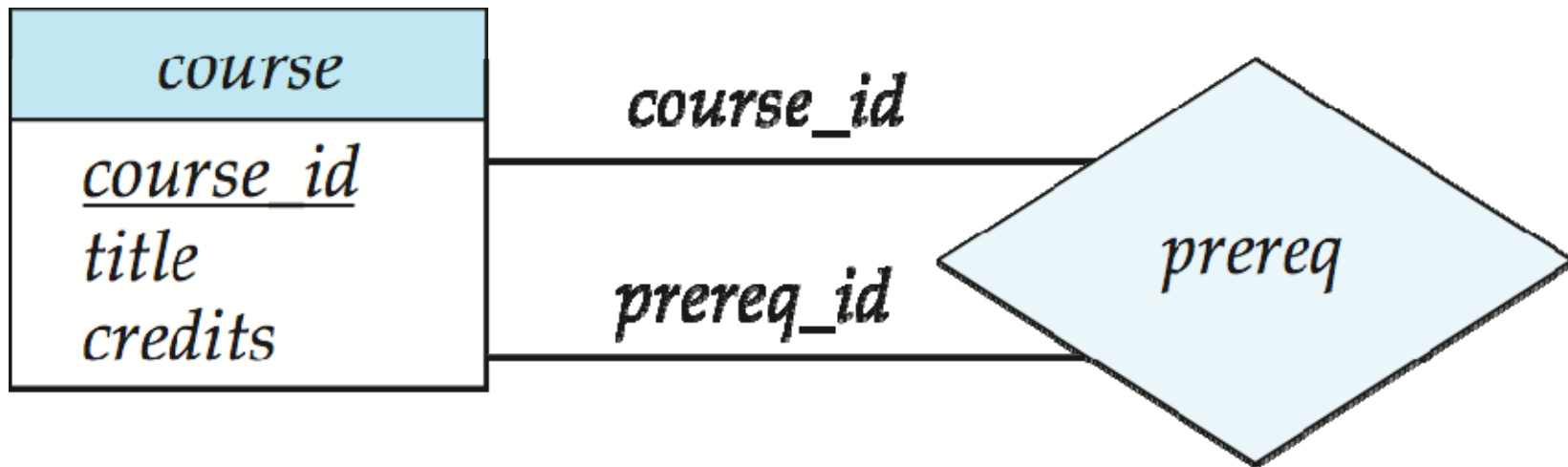
Complex Attributes





Roles

- Entity sets of a relationship need not be distinct
 - Each occurrence of an entity set plays a “role” in the relationship
- The labels “*course_id*” and “*prereq_id*” are called **roles**.



선위의 label → role(역할)을 표시



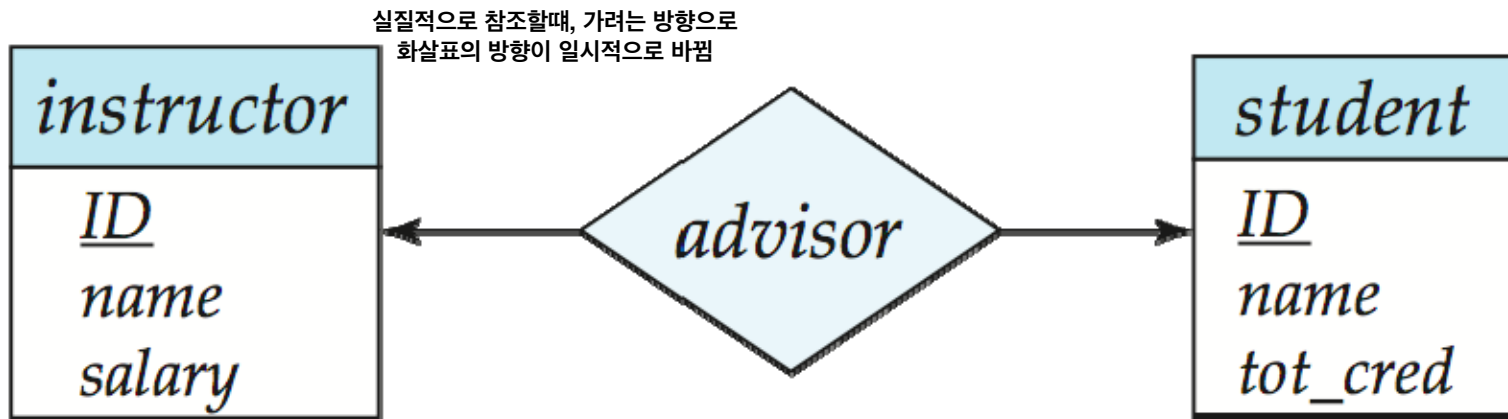
Cardinality Constraints

- We express cardinality constraints by drawing either a directed line (\rightarrow), signifying “one,” or an undirected line (—), signifying “many,” between the relationship set and the entity set. directed line=1 관계, undirected line: M(다)관계
- One-to-one relationship:
 - A student is associated with at most one *instructor* via the relationship *advisor*
 - A *student* is associated with at most one *department* via *stud_dept*



One-to-One Relationship

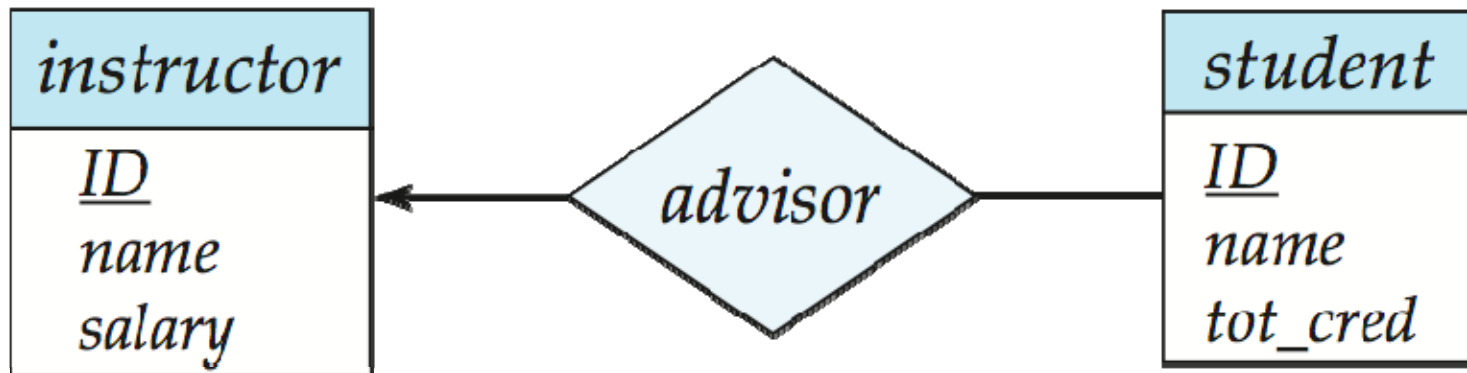
- one-to-one relationship between an *instructor* and a *student*
 - an instructor is associated with at most one student via *advisor*
 - and a student is associated with at most one instructor via *advisor*





One-to-Many Relationship

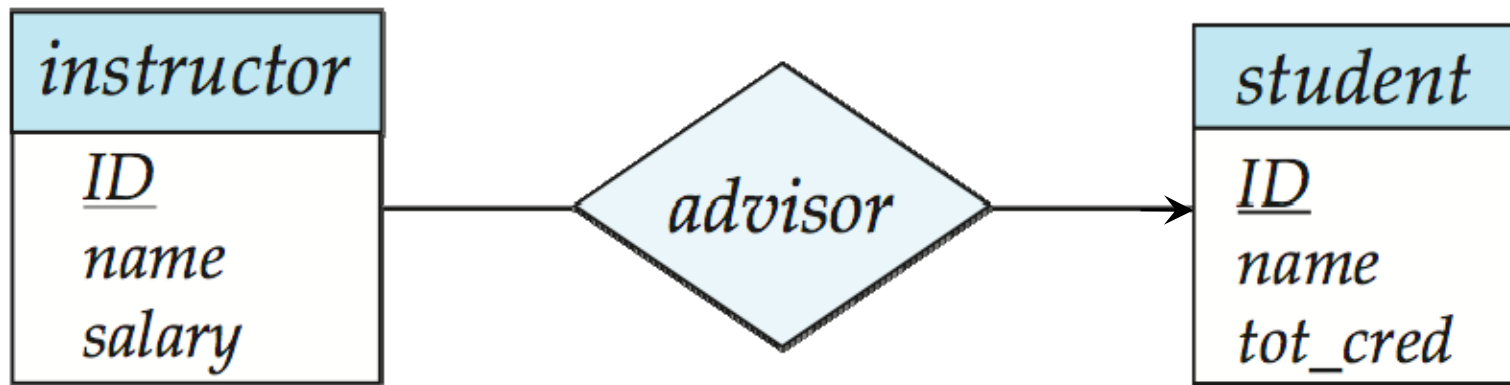
- one-to-many relationship between an *instructor* and a *student*
 - an instructor is associated with several (including 0) students via *advisor*
 - a student is associated with at most one instructor via *advisor*,





Many-to-One Relationships

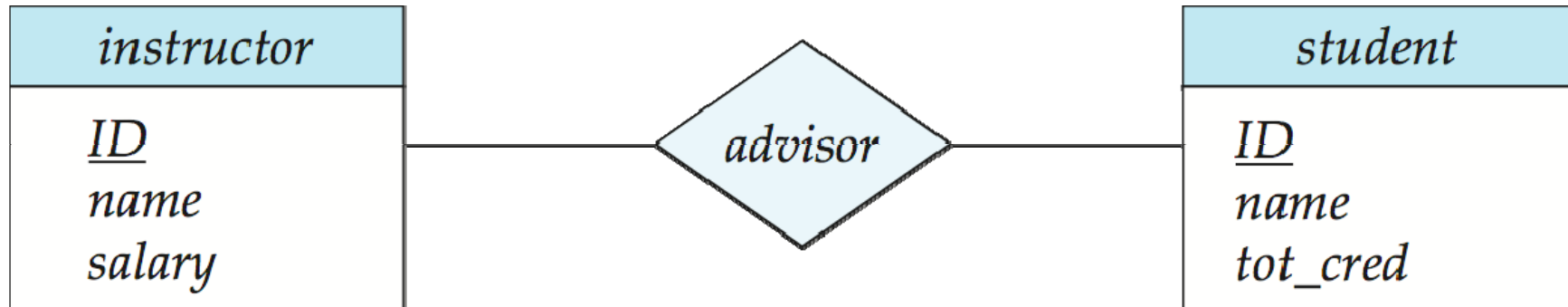
- In a many-to-one relationship between an *instructor* and a *student*,
 - an instructor is associated with at most one student via *advisor*,
 - and a student is associated with several (including 0) instructors via *advisor*





Many-to-Many Relationship

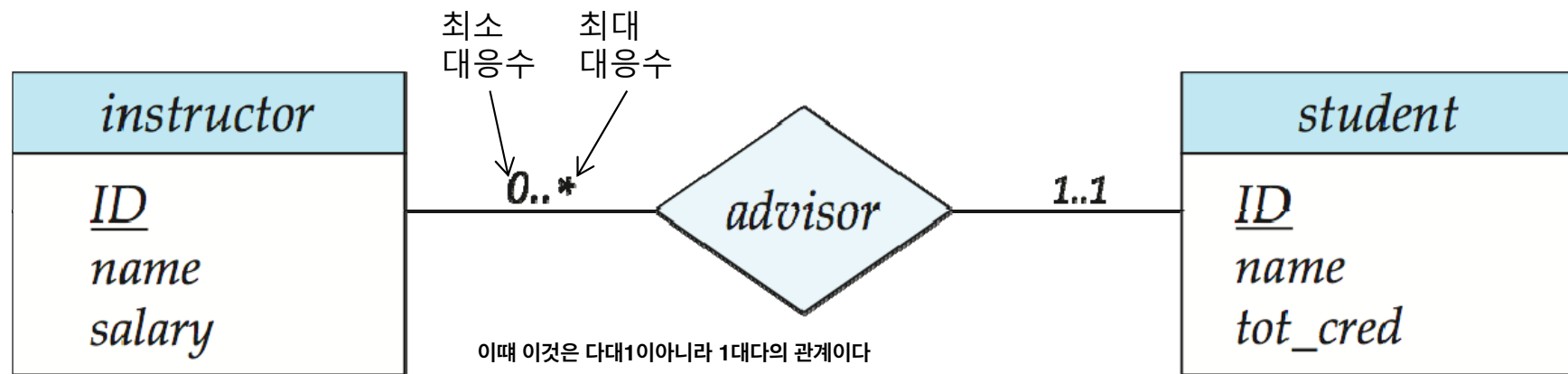
- An instructor is associated with several (possibly 0) students via *advisor*
- A student is associated with several (possibly 0) instructors via *advisor*





Alternative Notation for Cardinality Limits

- Cardinality limits can also express participation constraints



$0..*$ → 각 교수님은 지도학생이 한 명도 없을 수도 있고, 여러 명 있을 수도 있다.

$1..1$ → 학생은 반드시 한 명의 지도교수님이 있어야 한다.

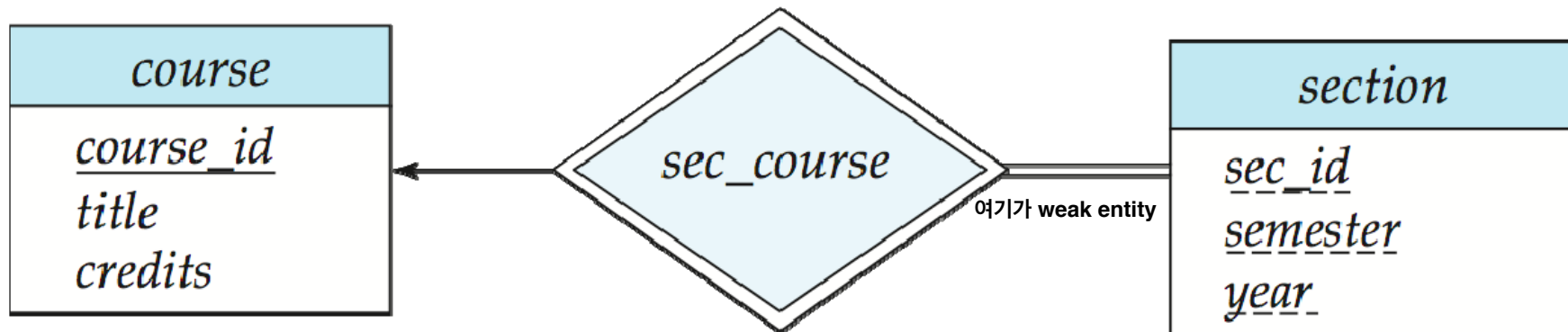
(학생 객체에 있는 모든 튜플들은 *advisor* 관계에 모두 참여 = “전체참여”)



Participation of an Entity Set in a Relationship Set

- **Total participation** (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set
 - E.g., participation of *section* in *sec_course* is total
 - ▶ every *section* must have an associated course
- **Partial participation**: some entities may not participate in any relationship in the relationship set
 - Example: participation of *instructor* in *advisor* is partial → 앞의 예 참고

이는 Total participation임

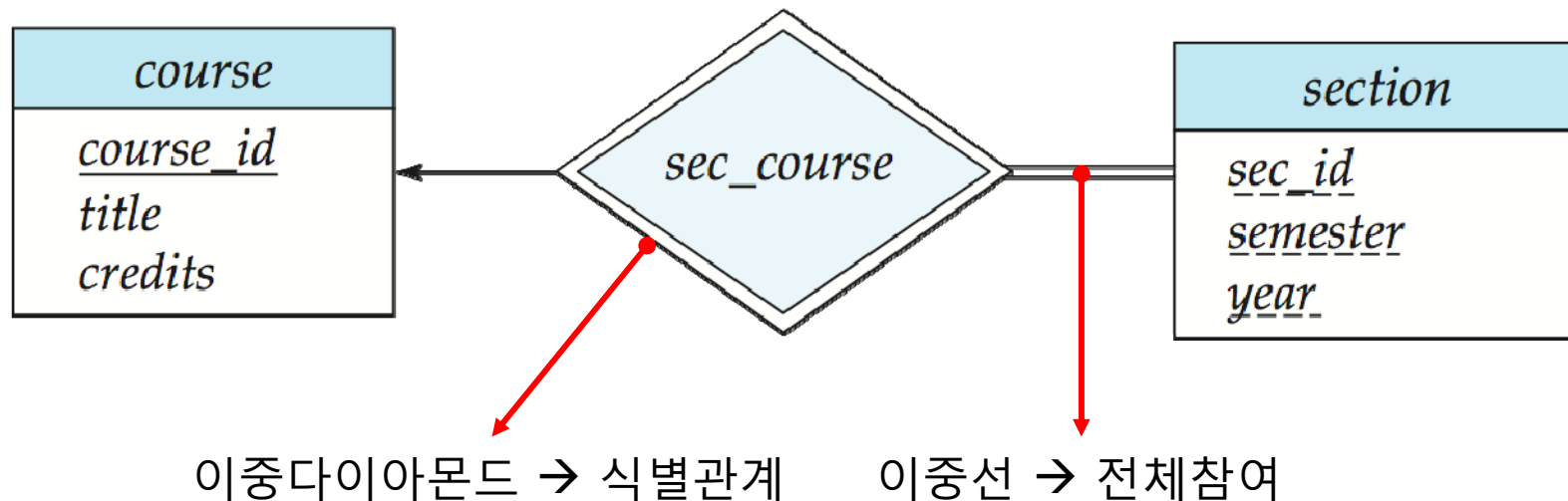


야매로 1의 관계를 가지면 대부분
double line을 가질수 있는 확률이 높음



Weak Entity Sets (Cont.)

- We underline the discriminator of a weak entity set with a dashed line.
- We put the identifying relationship of a weak entity in a double diamond.
- Primary key for *section* – (*course_id*, *sec_id*, *semester*, *year*)





Weak Entity Sets (Cont.)

- Note: the primary key of the strong entity set is not explicitly stored with the weak entity set, since it is implicit in the identifying relationship.
- If *course_id* were explicitly stored, *section* could be made a strong entity, but then the relationship between *section* and *course* would be duplicated by an implicit relationship defined by the attribute *course_id* common to *course* and *section*

참고1: E-R 다이어그램에서 section에서 course_id는 생략 가능하지만, 실제 DB 구현시 course_id가 있어야 한다.

참고2: sec_id를 unique하게 만들면 section 객체집합은 중복되지 않는 주키를 가질 수 있다. 그러나 개념적으로 section의 존재는 course에 의존적이기 때문에 바람직하지 않다.



E-R Diagram for a University Enterprise

