

Clarification: final

- “final” prevents further modification, such as subclassing a class, overriding a method, or reassigning a variable.

```
create type person as (
    name varchar(10),
    birth_date DATE
);
```

```
create type student as (
    person_data person,
    student_id varchar(10)
) final;
```

- However, many DBMS including PostgreSQL does not support final.

지원하지 않음

Clarification: wasNull()

- wasNull() checks if the last retrieved column was NULL

검색된

```
ResultSet rs = stmt.executeQuery("SELECT id, name, age FROM users");
while (rs.next()) {
    int id = rs.getInt("id");
    String name = rs.getString("name");
    int age = rs.getInt("age");

    // Check if 'age' was NULL
    if (rs.wasNull()) {
        System.out.println("ID: "+id+", Name: "+name+", Age: NULL");
    } else {
        System.out.println("ID: "+id+", Name: "+name+", Age: " + age);
    }
}
```

Database Systems

Lecture09 – Ch 6. Database Design Using the E-R Model

Beomseok Nam (남범석)
bnam@skku.edu

Modeling

- A *database* can be modeled as:
 - a collection of entities,
 - relationship among entities.
- An **entity** is an object distinguishable from other objects.
 - Example: specific student, e.g., Williams
- Entities have **attributes**
 - Example: students have *names* and *phone numbers*
- An entity set is a set of entities of the same type that share the same properties.
 - equivalent of relation (**table**)
 - Example: set of all students, departments, etc

Entity Sets *instructor* and *student*

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)
student entity

advisor
relationship set

22222 (Einstein)
instructor entity

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

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

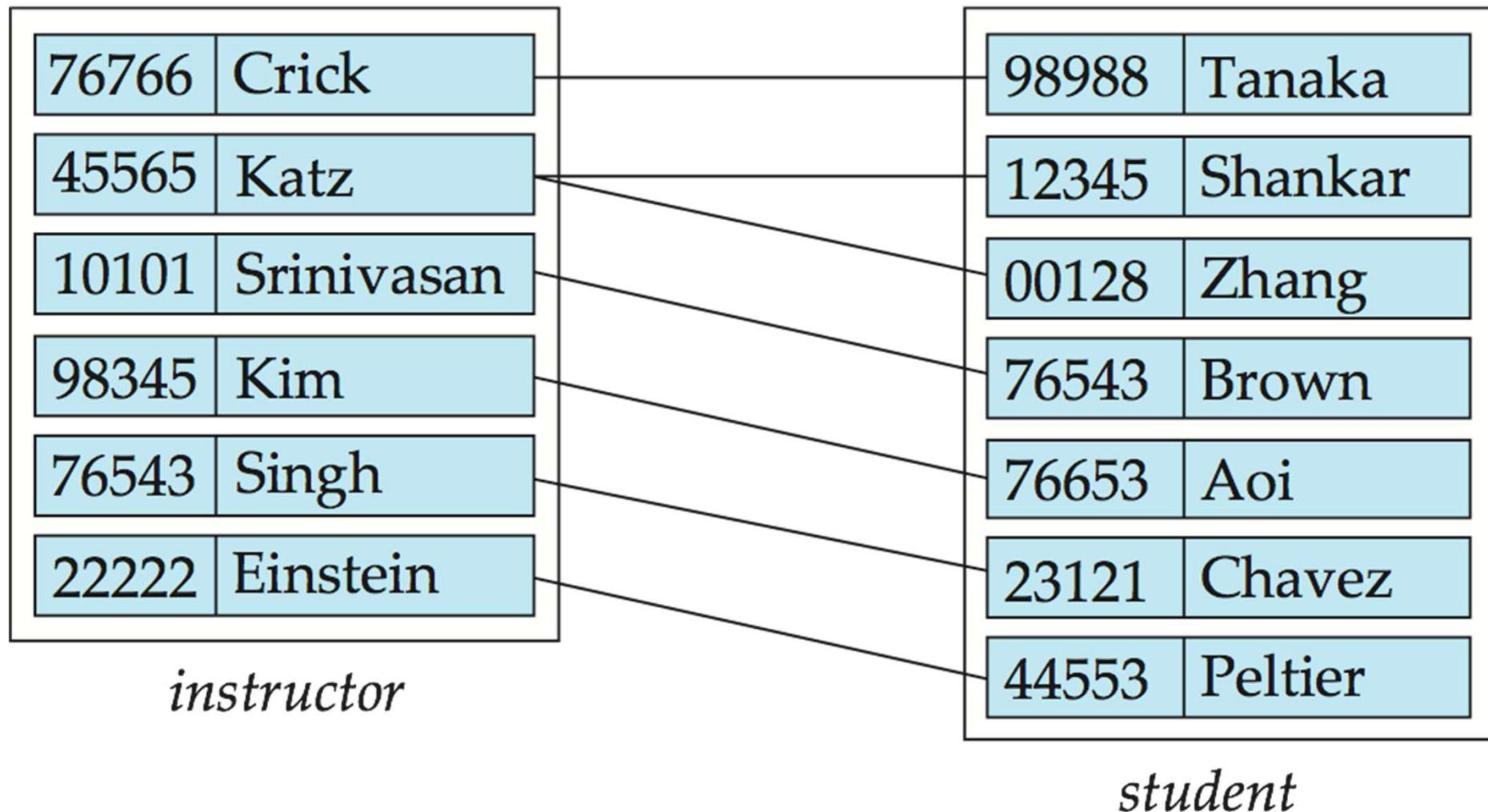
Entity Set (table)

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

- Example:

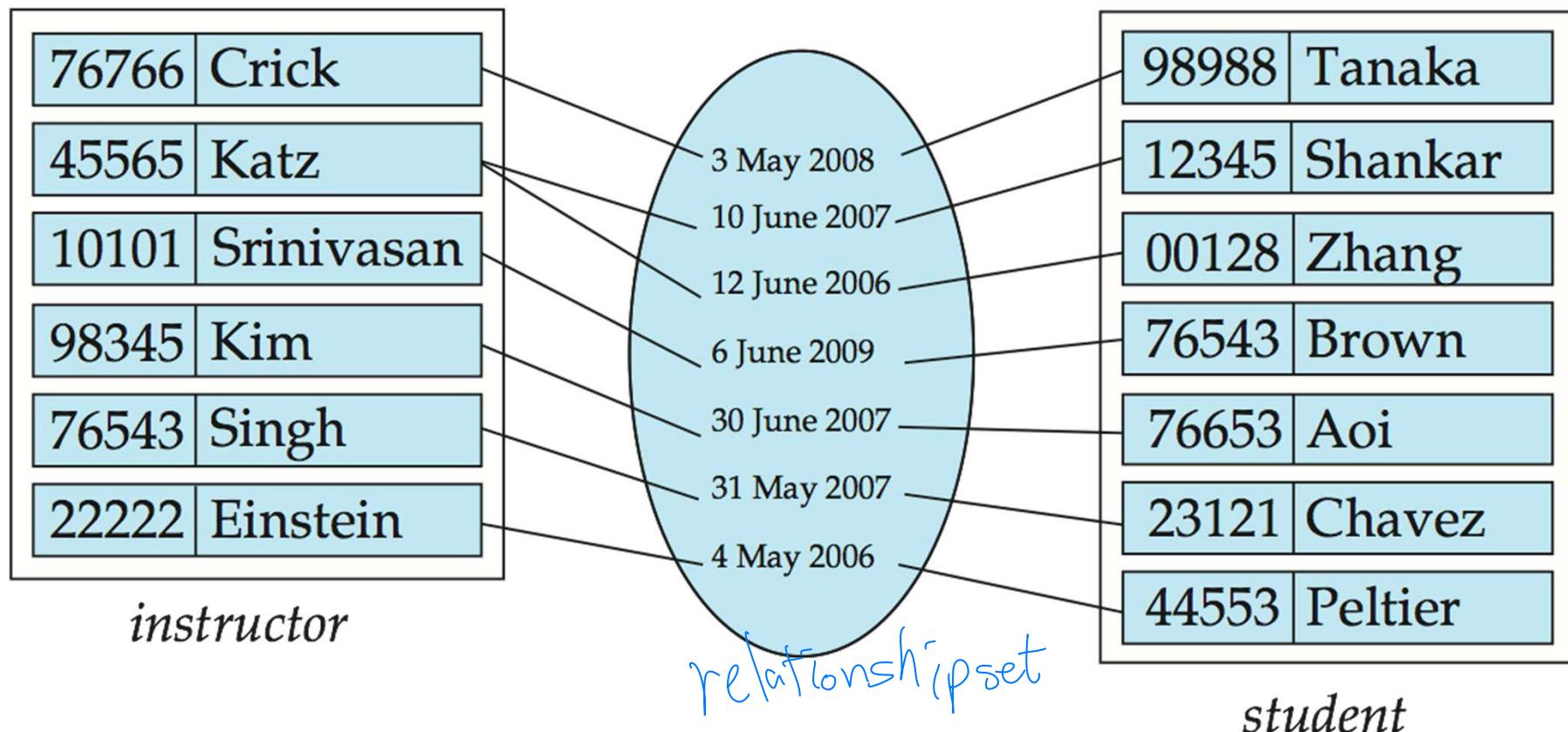
$(44553, 22222) \in \underline{\text{advisor}}$

Relationship Set *advisor*



Relationship Sets (Cont.)

- A relationship set can have **attributes**.
 - e.g., the *advisor* relationship set between *instructor* and *student* may have the attribute date which tracks when the student started being advised by the advisor



Degree of a Relationship Set

- **binary relationship**

2개의 관계

- involve two entity sets (or degree two).

- Most relationships are binary.

2개 이상 관계

- Relationships between more than two entity sets are rare.

- ▶ Example: *students* work on research *projects* under the guidance of an *instructor*.

- ▶ relationship *proj_guide* is a ternary relationship between *instructor*, *student*, and *project*

Attributes

- An entity is represented by a set of attributes.

- Example:

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

course= (course_id, title, credits)

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

- Attribute types:
ex Id : // *ex address : 경기도 / 서울시 / 목동 ...*

- Simple** and **composite** attributes.

처음 구분됨

- Single-valued** and **multivalued** attributes

- (개인)* – Example: multivalued attribute: *phone_numbers*

- a person can have more than one phone number.*

- Derived** attributes

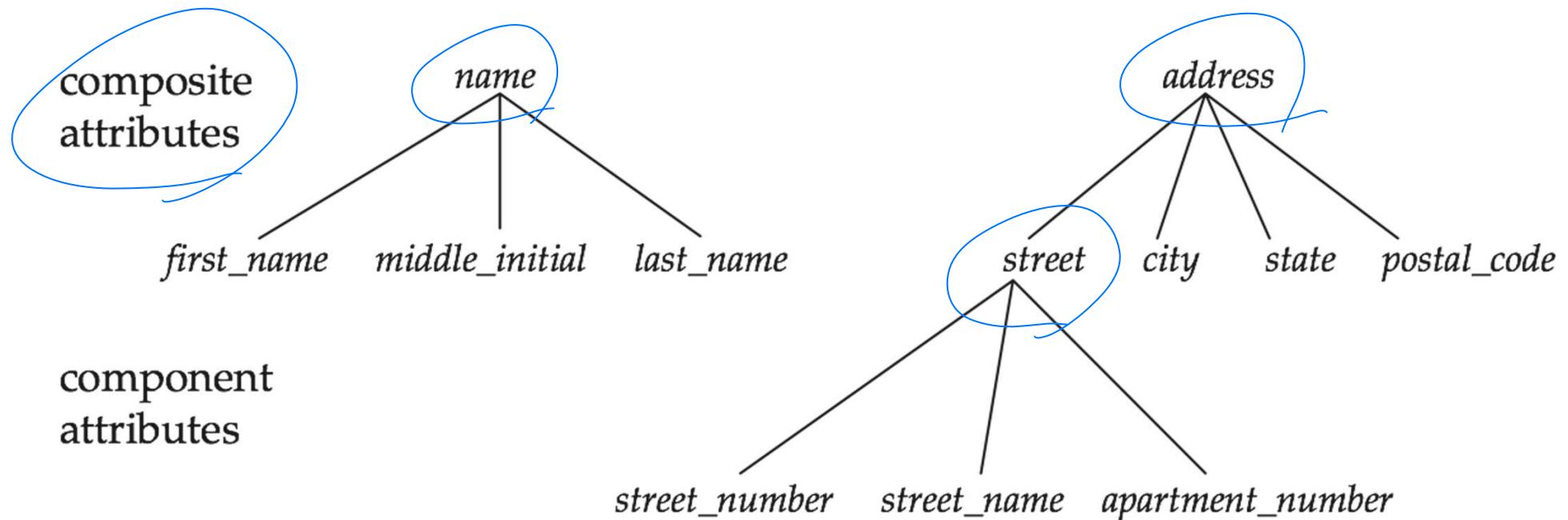
추가됨

- Can be computed from other attributes*

- Example: age, given date_of_birth*

↑
계산됨

Composite Attributes



Mapping Cardinality Constraints

- **Mapping Cardinality:**

한개체가 다른개체
몇개랑 연관되는지?

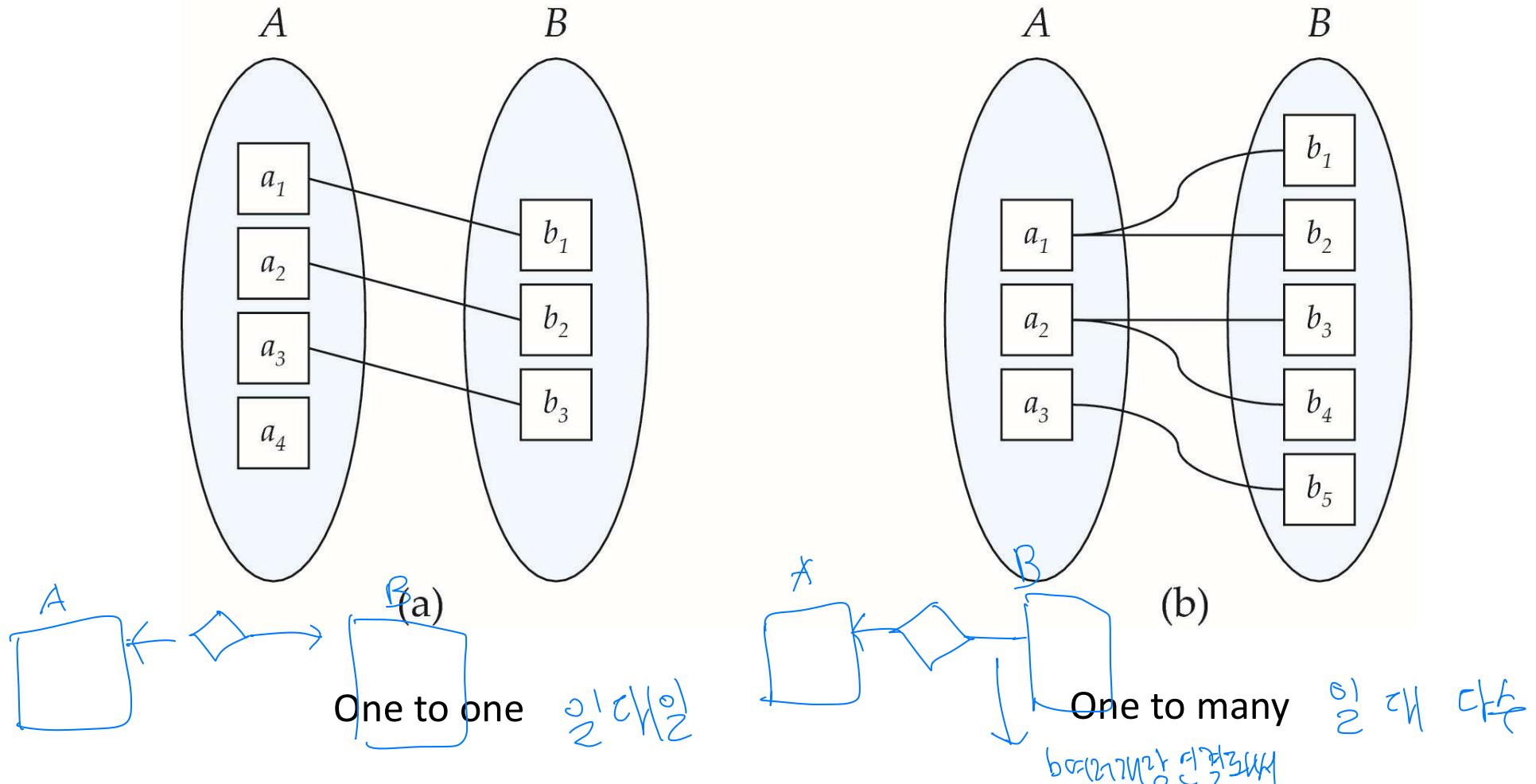
Einstein < Peltier
Bob

- The number of entities associated to another entity in a relationship set.
- useful in describing binary relationship sets.

- For a binary relationship set the mapping cardinality must be one of the following types:

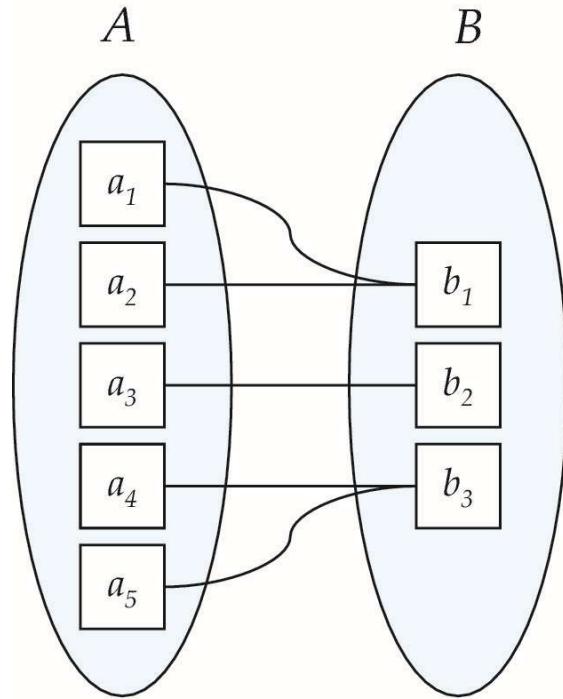
- One to one
- One to many
- Many to one
- Many to many

Mapping Cardinalities



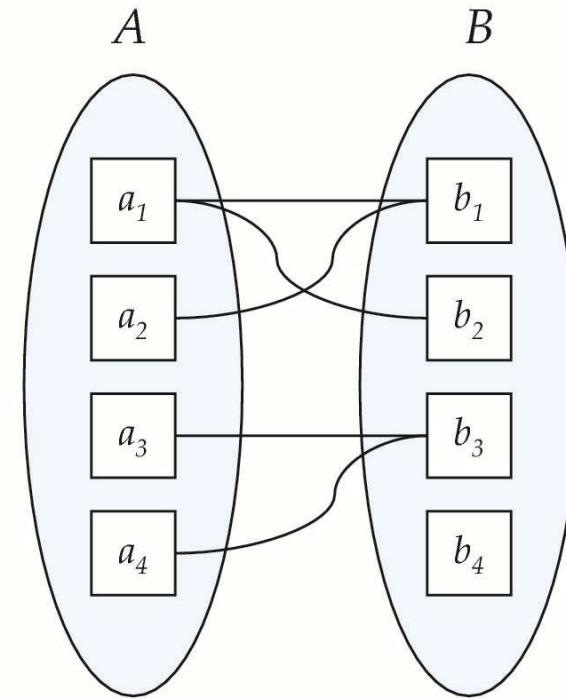
Note: Some elements in A and B may not be mapped to any elements in the other set

Mapping Cardinalities



(a)

Many to one



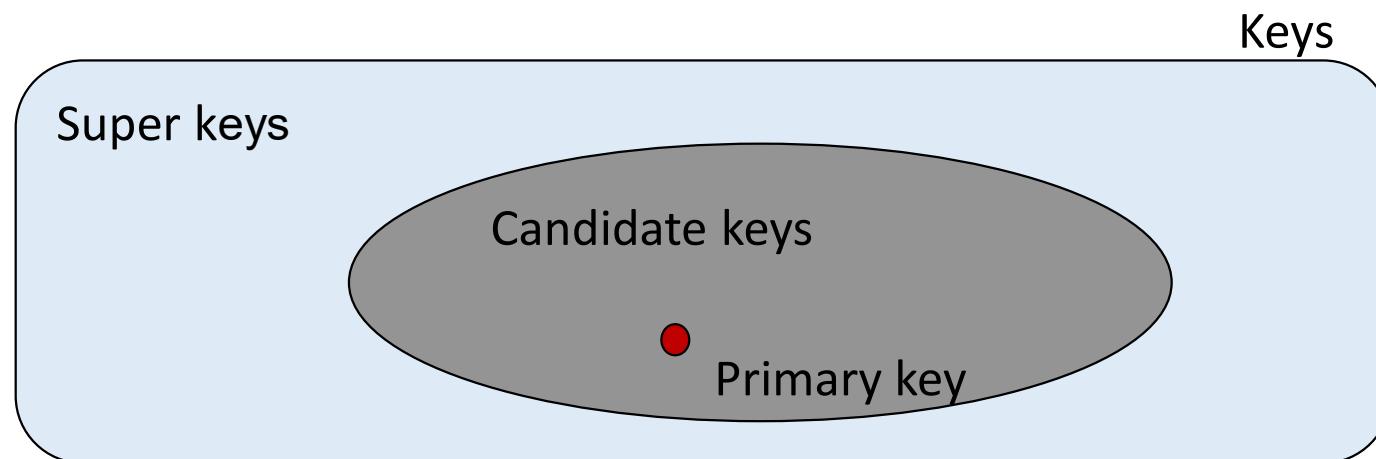
(b)

Many to many

Note: Some elements in A and B may not be mapped to any elements in the other set

Keys

- **Key** is a set of attributes that uniquely identifies a record.
 - **Super key** ^{부수키} is a set of attributes that uniquely determines each entity.
 - $(ID, name)$ is a super key of *instructor*
 - A **candidate key** ^{후보키} is a minimal super key
 - ID is a candidate key of *instructor*
 - $course_id$ is a candidate key of *course*
 - Several candidate keys may exist
 - One of the candidate keys is selected to be the **primary key**.^{주요키 (대표로 지정한 키)}



Keys for Relationship Sets

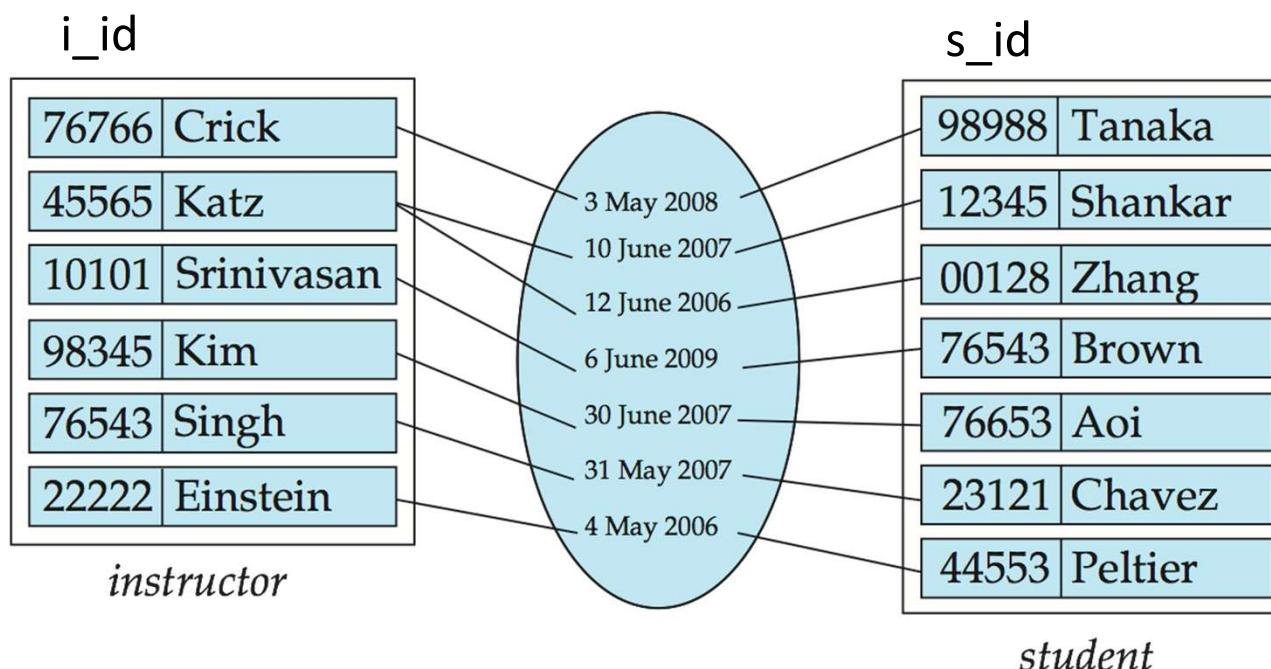
- The combination of primary keys forms a super key of a relationship set.

- E.g.)

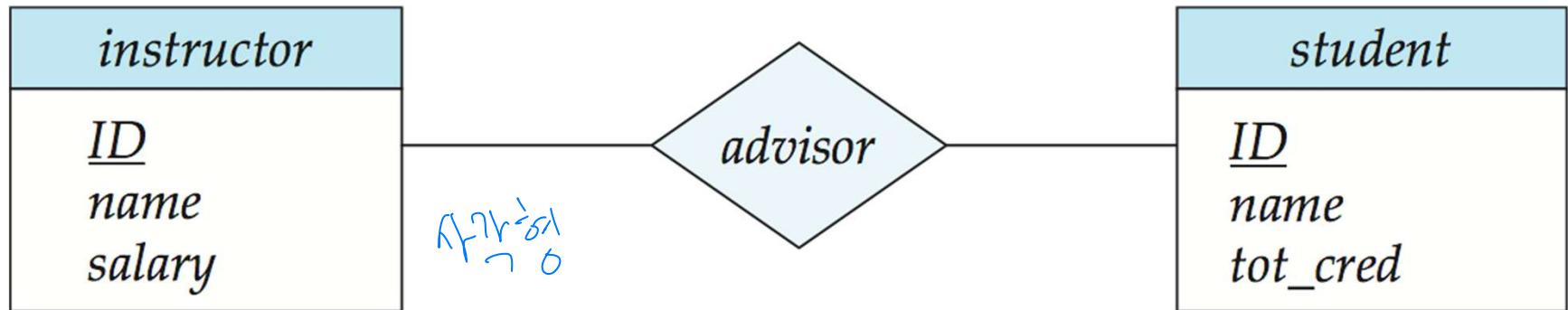
s_id is the primary key of student

i_id is the primary key of instructor

→ (s_id, i_id) is the super key of advisor



E-R Diagrams



- Rectangles represent entity sets.

- Diamonds represent relationship sets.

- Attributes listed inside entity rectangle
- Underline indicates primary key attributes

일본

Entity With Composite, Multivalued, and Derived Attributes

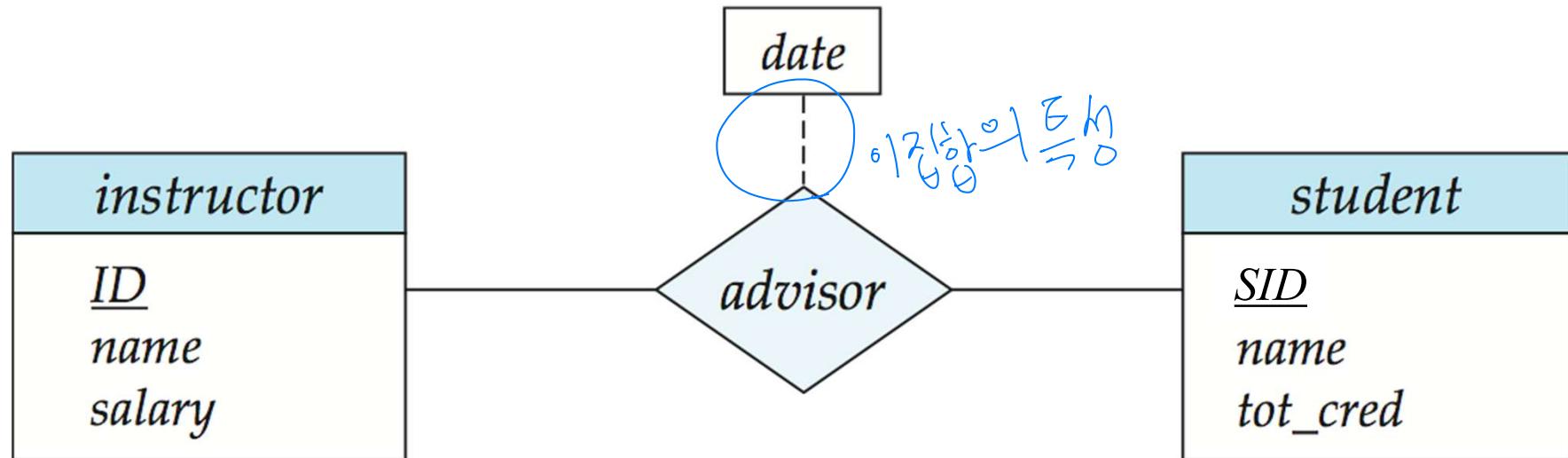
<i>instructor</i>
<u>ID</u>
<u>name</u>
first_name
middle_initial
last_name
<u>address</u>
street
street_number
street_name
apt_number
city
state
zip
{ phone_number }
date_of_birth
age () ← calculate : ()

- Composite attributes are flattened out by creating a separate attribute for each component attribute
- Ignoring multivalued attributes, extended instructor schema is
 - *instructor(ID, first_name, middle_initial, last_name, street_number, street_name, apt_number, city, state, zip_code, date_of_birth)*

mult(value : { })

calculate : ()

Relationship Sets with Attributes



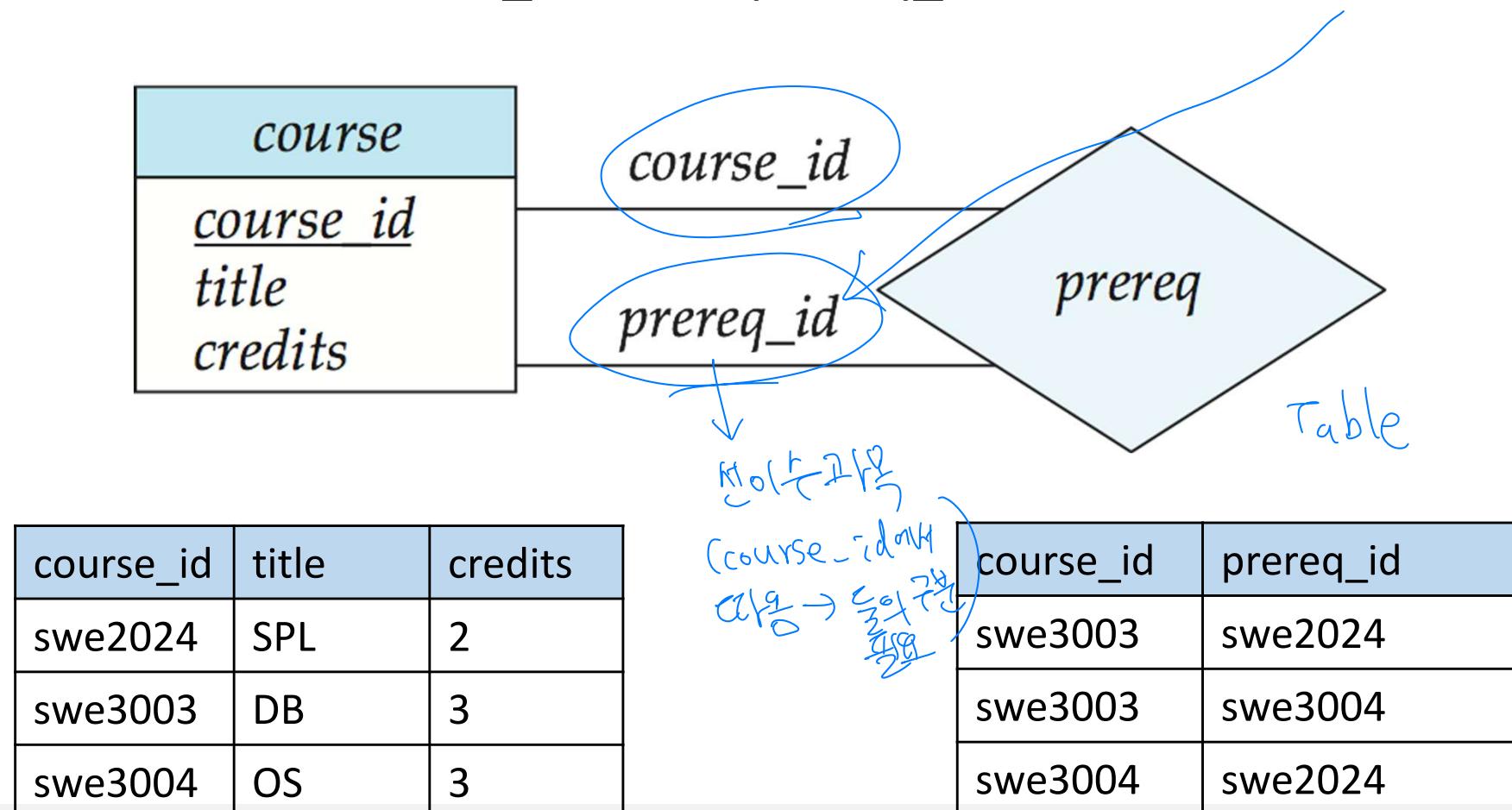
ID	name	salary
10	Alice	1000
20	Bob	2000

ID	SID	date
10	102	...
10	105	
20	105	

SID	name	tot_cred
102	John	123
105	Tom	108

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**.



Cardinality Constraints

- We express cardinality constraints by line types
 - directed line (\rightarrow), signifying “one”
 - undirected line ($-$), signifying “many,”

P.13 27

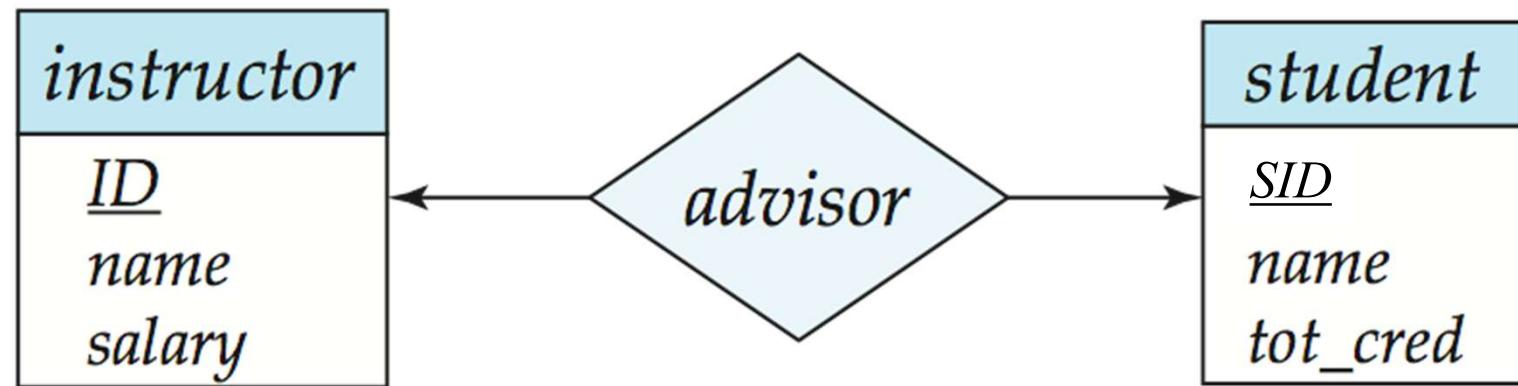


- One-to-one relationship:

- 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*



ID	name	salary
10	Alice	1000
20	Bob	2000

A table illustrating a violation of the one-to-one constraint. It shows three rows of data:

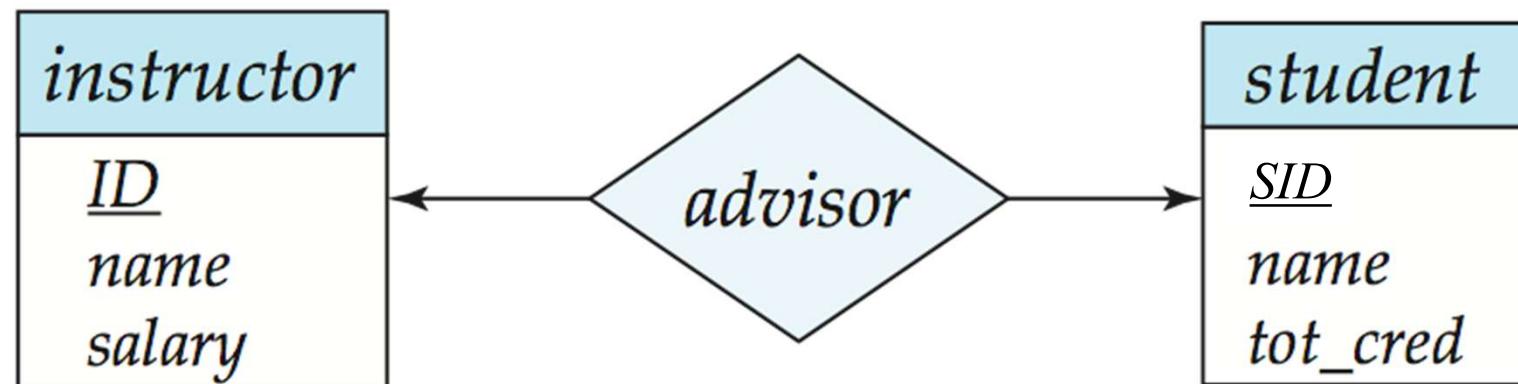
ID	SID	date
10	102	...
20	105	...

A large orange 'X' is drawn across the entire table, indicating that both rows 10 and 20 share the same SID value (102), which violates the one-to-one constraint for the advisor relationship.

SID	name	tot_cred
102	John	123
105	Tom	108

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*



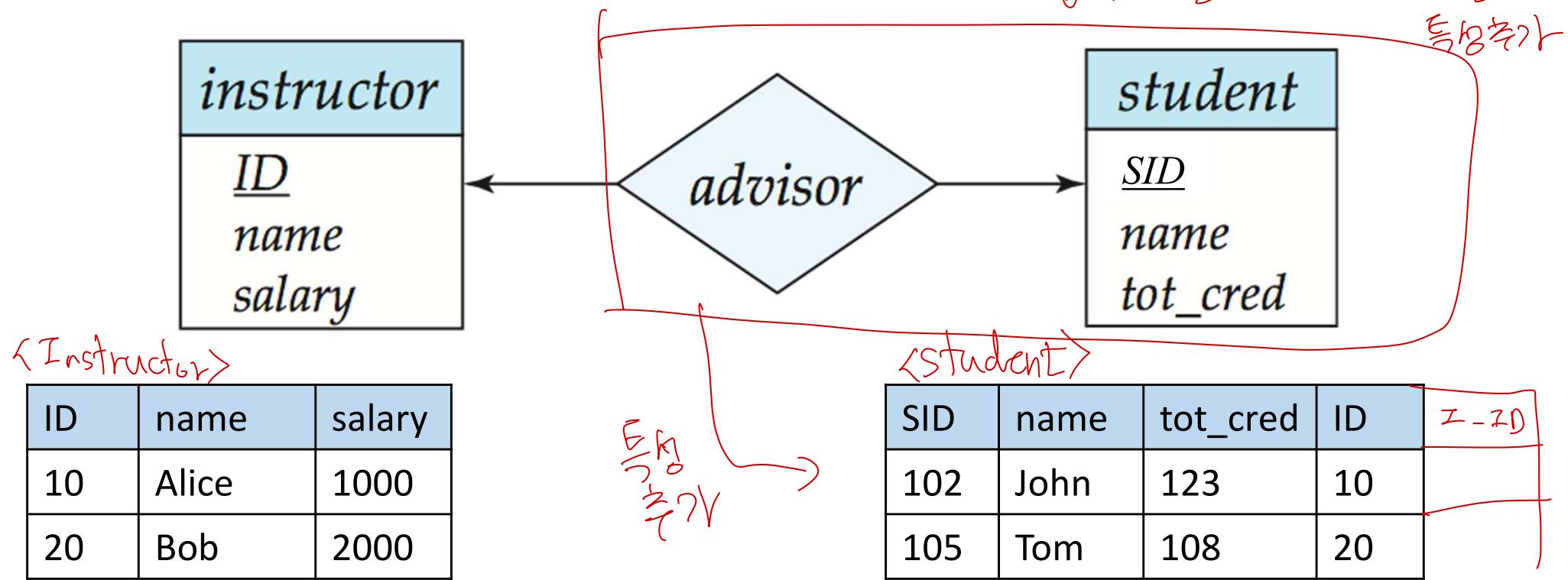
ID	name	salary	SID
10	Alice	1000	102
20	Bob	2000	105

SID	name	tot_cred
102	John	123
105	Tom	108

One-to-One Relationship

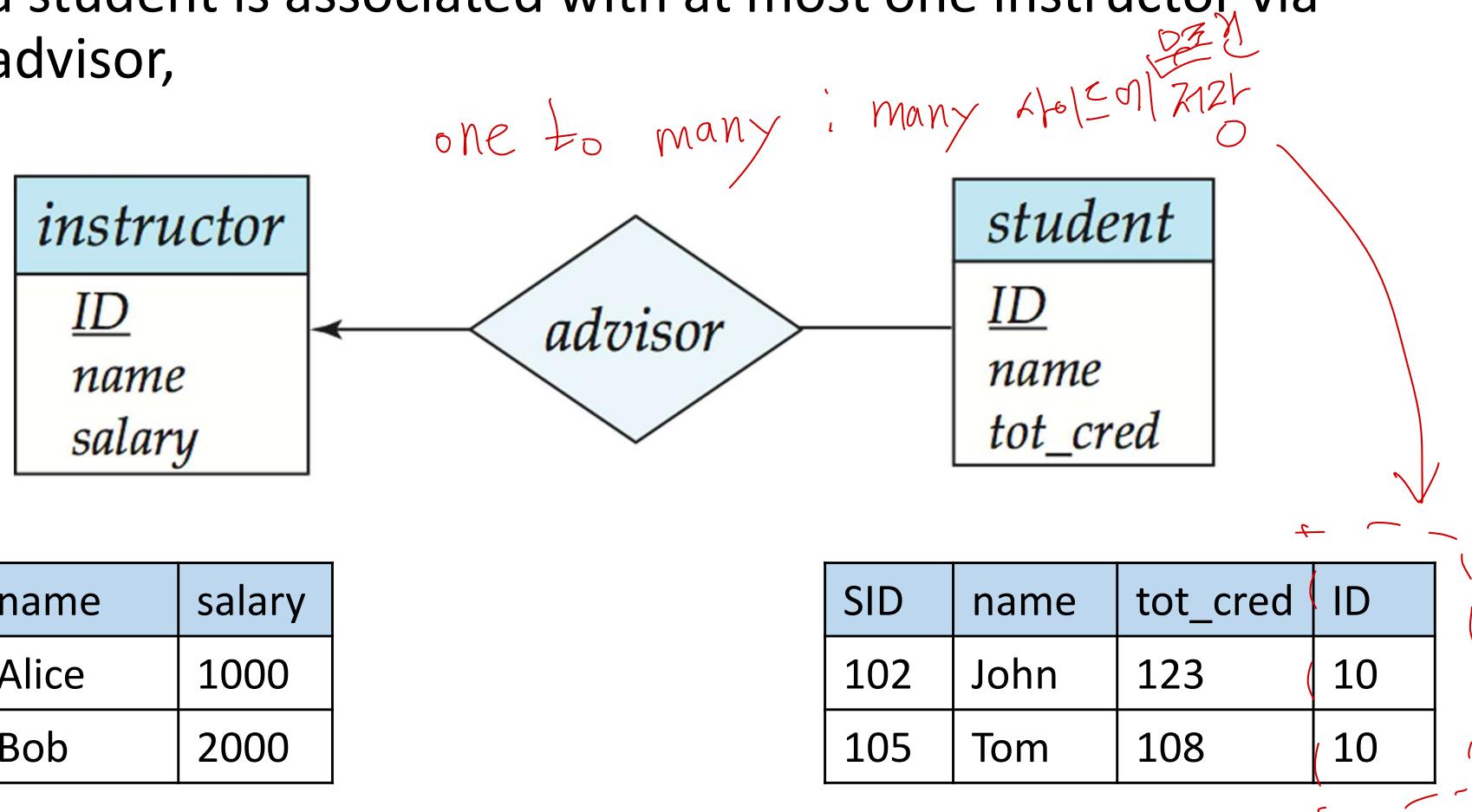
- one-to-one relationship between an *instructor* and a *student*
 - an instructor is associated with at most one student via *advisor*
 - and a student is associated with at most one instructor via *advisor*

one-to-one, 서로에게 대처(불생성 X) 기존테이블에
(동다상관 X) 틸당첨자



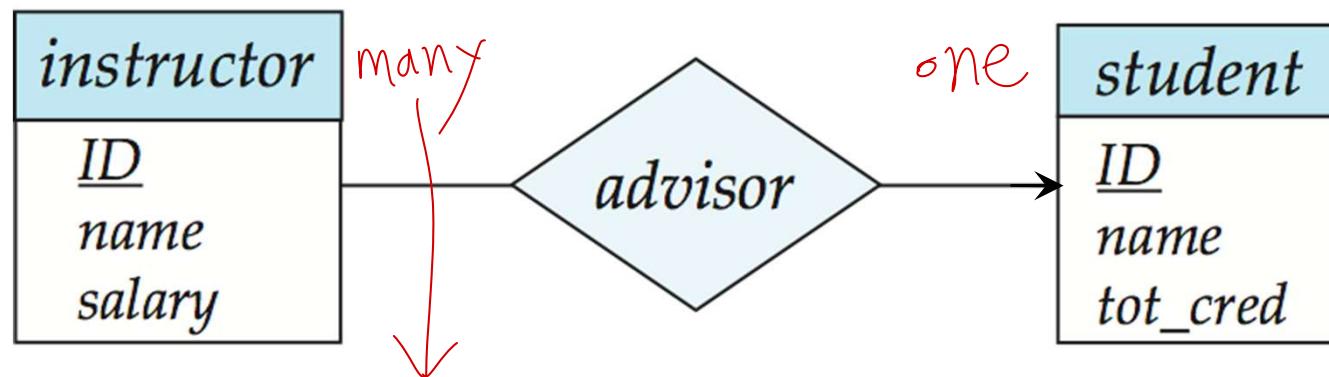
One-to-Many Relationship

- 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*

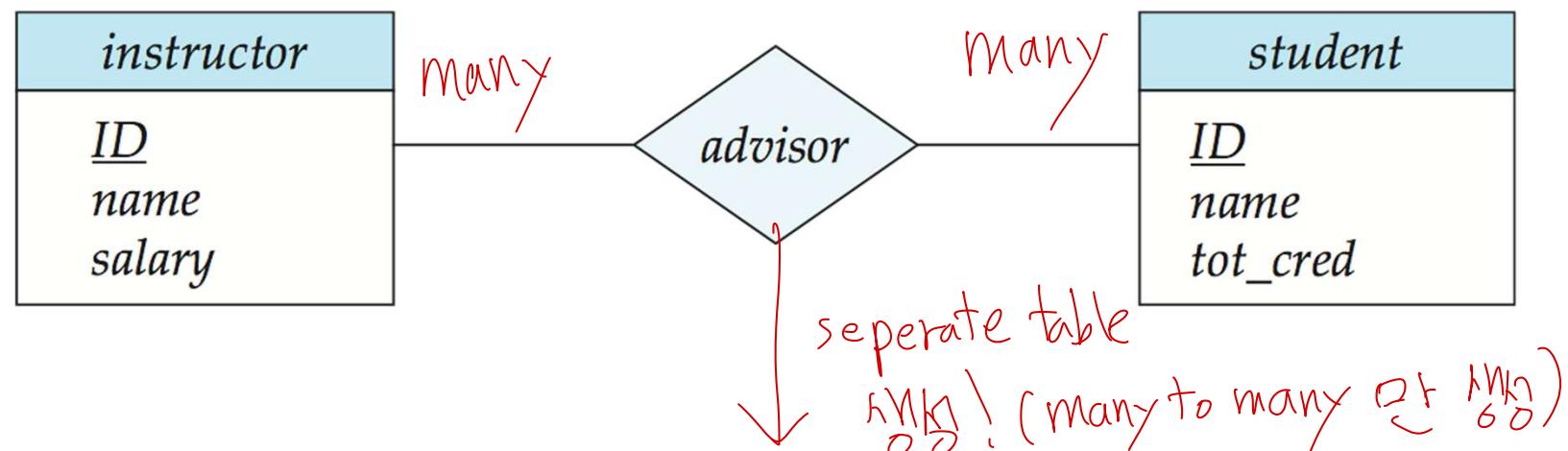


ID	name	salary	SID
10	Alice	1000	102
20	Bob	2000	102

SID	name	tot_cred
102	John	123
105	Tom	108

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*



ID	name	salary
10	Alice	1000
20	Bob	2000

ID	SID
10	102
10	105
20	105

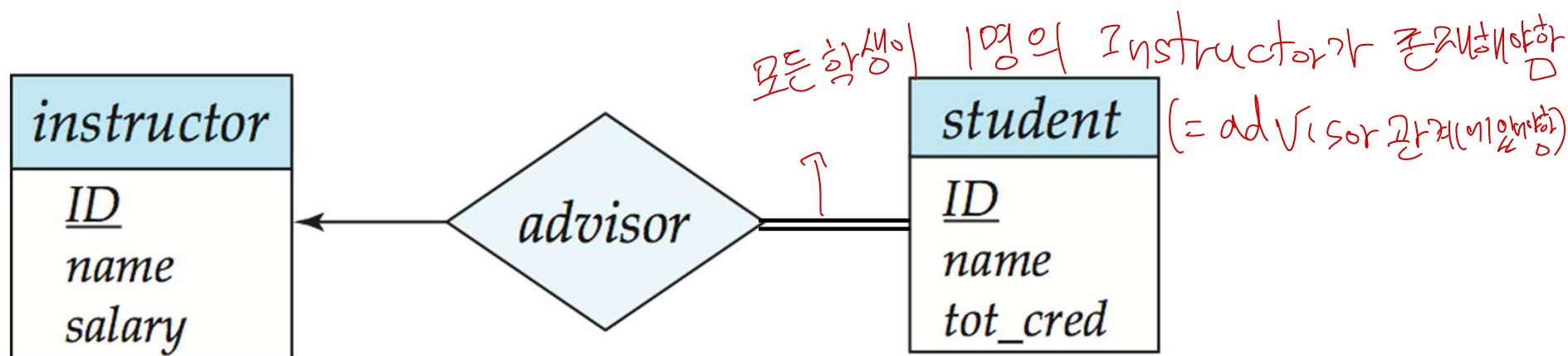
SID	name	tot_cred
102	John	123
105	Tom	108

Participation of an Entity Set in a Relationship Set

↳ 이론적이라 이해되여

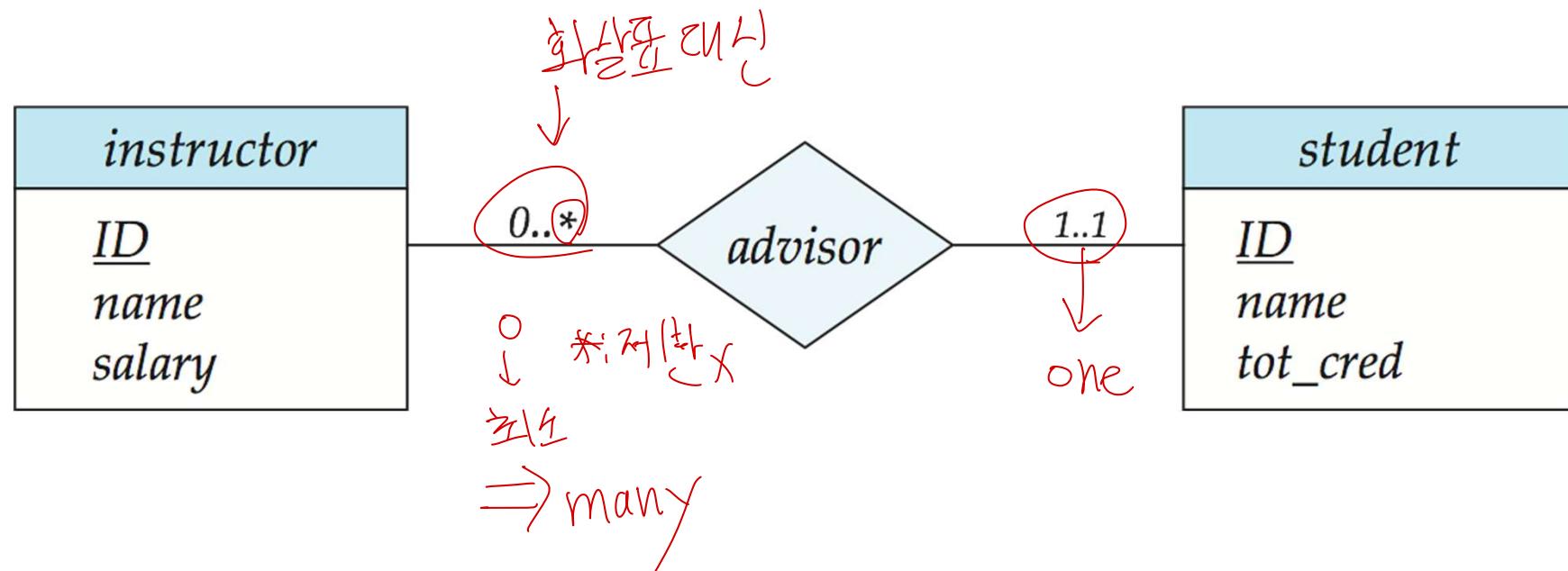
■ Total participation (indicated by double line):

- every entity in the entity set participates in at least one relationship
- E.g., participation of student in *advisor* is total
 - every student must have an advisor
- Partial participation: The opposite of total participation



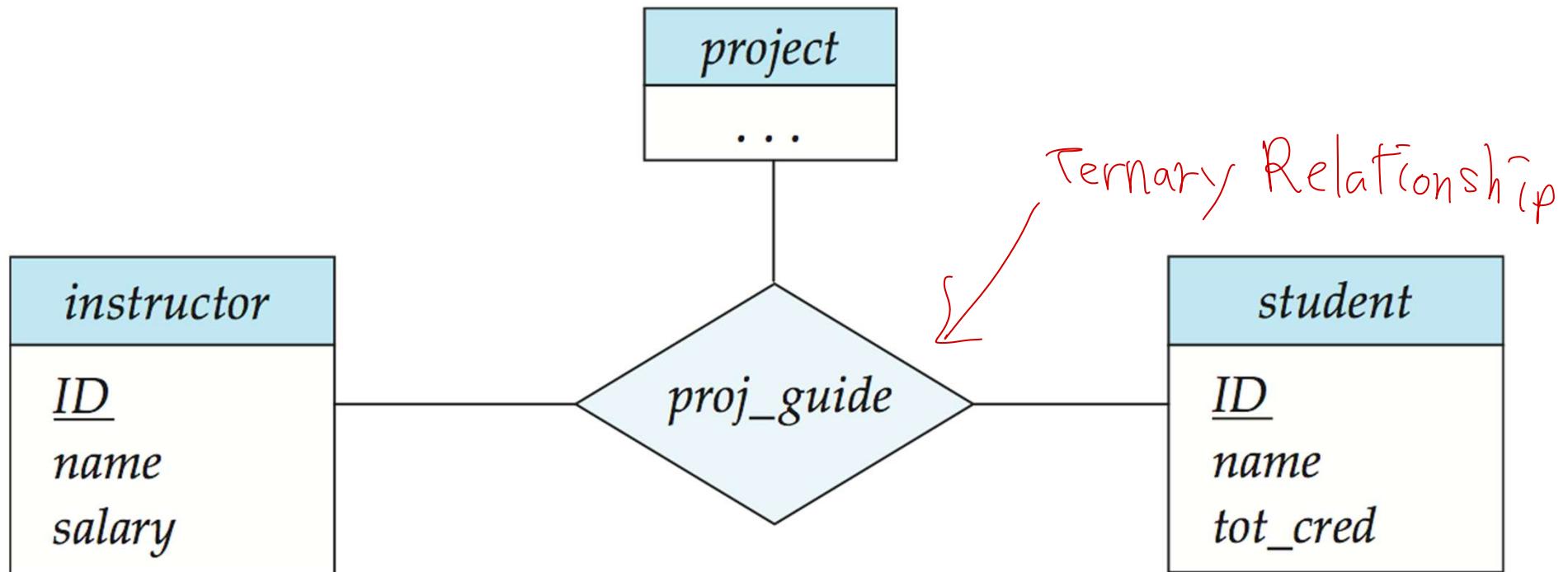
Alternative Notation for Cardinality Limits

- Cardinality limits can express participation constraints



E-R Diagram with a Ternary Relationship

Bad!



Cardinality Constraints on Ternary Relationship

- No more than one arrow is allowed for a ternary (or greater degree) relationship
- If there is more than one arrow, there are two **confusing** ways of defining the meaning.
 - E.g., a ternary relationship R between A , B and C with arrows to B and C could mean
 1. each A entity is associated with a unique entity from B and C or
 2. each pair of entities from (A, B) is associated with a unique C entity, and each pair (A, C) is associated with a unique B
 - To avoid confusion ***we outlaw more than one arrow***

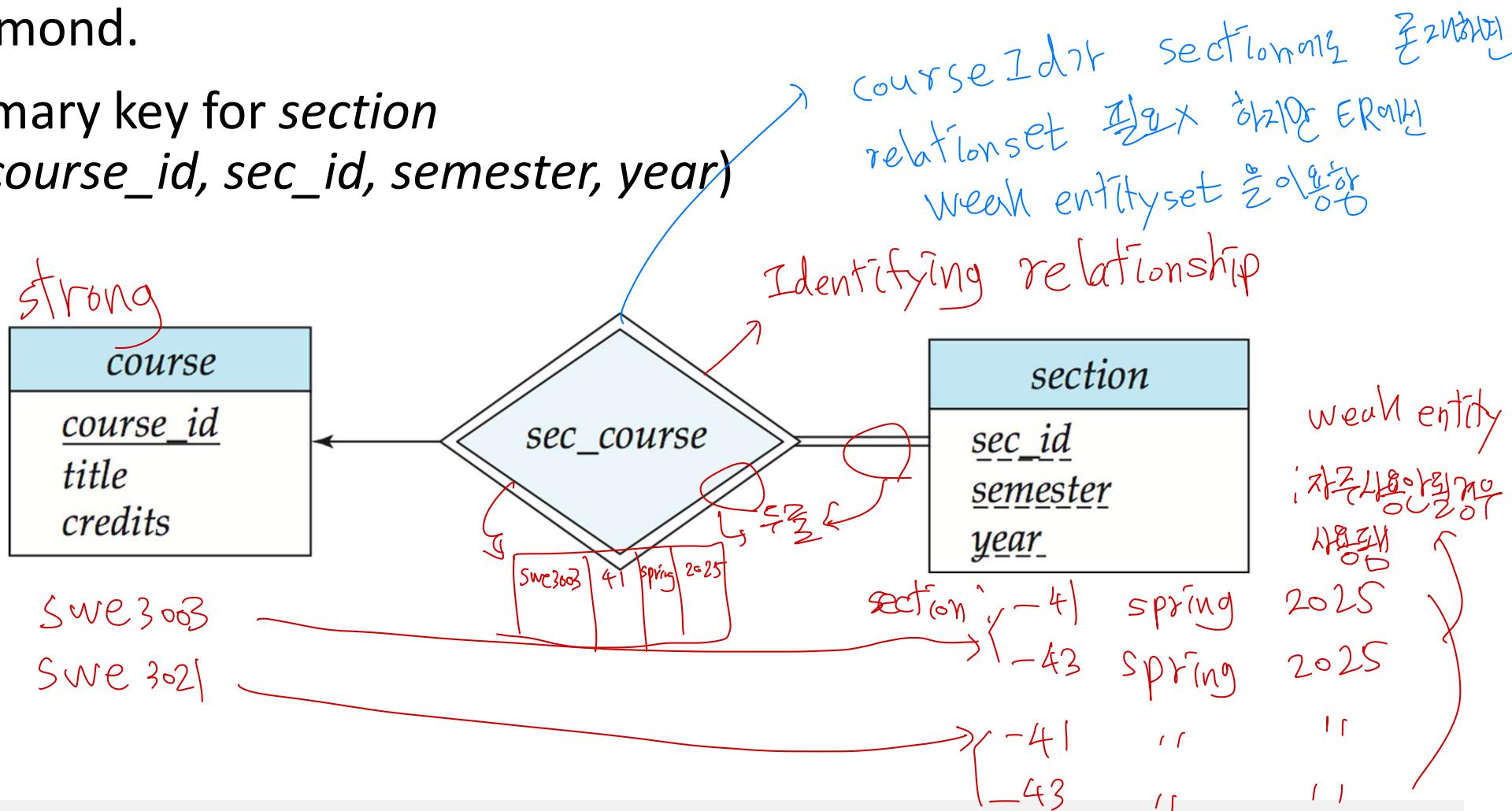
Bad 이유 : 화살표가 2개 이상일 시 의미가 혼동됨
∴ 화살표 하나일 때만 사용!

Weak Entity Sets

- An entity set that does not have a primary key is a **weak entity set**.
- Weak entity set depends on **identifying entity set**
 - It relates to the identifying entity set via a total, one-to-many relationship set
 - **Identifying relationship** depicted using a double diamond
- The **discriminator** (*or partial key*) of a weak entity set is the set of attributes that distinguishes an entity of a weak entity set.
- The primary key of a weak entity set is formed by the primary key of the identifying entity set plus the weak entity set's discriminator.

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 included in the weak entity set in ER-diagram (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*

E-R Diagram for a University Enterprise

