Chapter 7: Entity-Relationship (E-R) Model – Part 2

Revision by Gun-Woo Kim

Dept. of Computer Science and Engineering

Hanyang University

Database System Concepts, 6th Ed.

Contents

7.6 Reduction to Relational Schemas

7.7 Entity-Relationship Design Issues

참고 *Alternative Natation for Modeling Data*

7.6 Reduction to Relation Schemas

E-R 다이어그램으로부터 릴레이션 스키마를 만드는 과정을 의미

Entity sets and relationship sets can be expressed uniformly as *relation schemas* that represent the contents of the database.

Representing strong entity sets with simple attributes

Representing strong entity sets with complex attributes

Representing weak entity sets

Representing relationship sets

A database which conforms to an E-R diagram can be represented by a collection of schemas.

For each entity set and relationship set there is a unique schema that is assigned the name of the corresponding entity set or relationship set.

Each schema has a number of columns (generally corresponding to attributes), which have unique names.

Representing Strong Entity Sets With Simple Attributes

student
ID
name
tot_cred

A strong entity set reduces to a schema with the same attributes

student(ID, name, tot_cred)

Representing Strong Entity Sets With Complex Attributes

instructor

ID
name
    first_name
    middle_initial
    last_name
address
    street
        street_number
        street_name
        apt_number
    city
    state
    zip
{ phone_number }
date_of_birth
age ( )

Composite attributes are flattened out by creating a separate attribute for each component attribute

Example: given entity set *instructor* with composite attribute *name* with component attributes *first_name* and *last_name,* the schema corresponding to the entity set has two attributes *name_first_name* and *name_last_name*

▸ *Prefix omitted if there is no ambiguity*

Except for 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)

유도된 값, 다중 값 은 또다른 임의의 스키마를 만들어 표현함

Representing Strong Entity Sets With Complex Attributes (Cont.)

A multivalued attribute *M* of an entity *E* is represented by a separate schema *EM*

- Schema *EM* has attributes corresponding to the primary key of *E* and an attribute corresponding to multivalued attribute *M*

- Example: Multivalued attribute *phone_number* of *instructor* is represented by a schema:
  inst_phone= (<u>ID</u>, <u>phone_number</u>)

- Each value of the multivalued attribute maps to a separate tuple of the relation on schema *EM*

  - For example, an *instructor* entity with primary key 22222 and phone numbers 456-7890 and 123-4567 maps to two tuples:
   (22222, 456-7890) and (22222, 123-4567)

Representing Strong Entity Sets With Complex Attributes (Cont.)

Special case:entity *time_slot* has only one attribute other than the primary-key attribute, and that attribute is multivalued

Optimization: Don't create the relation corresponding to the entity, just create the attribute corresponding to the multivalued attribute

time_slot(time_slot_id, day, start_time, end_time)

Caveat: *time_slot_id* attribute of *section (*from *sec_time_slot)* cannot be a foreign key due to this optimization →더 이상 time_slot_id가 주키가 아님

section		
sec_id		
semester		
year		
building		
room_no		
time_slot_id		

sec_time_slot

time_slot
time_slot_id
{ day
start_time
end_time
}
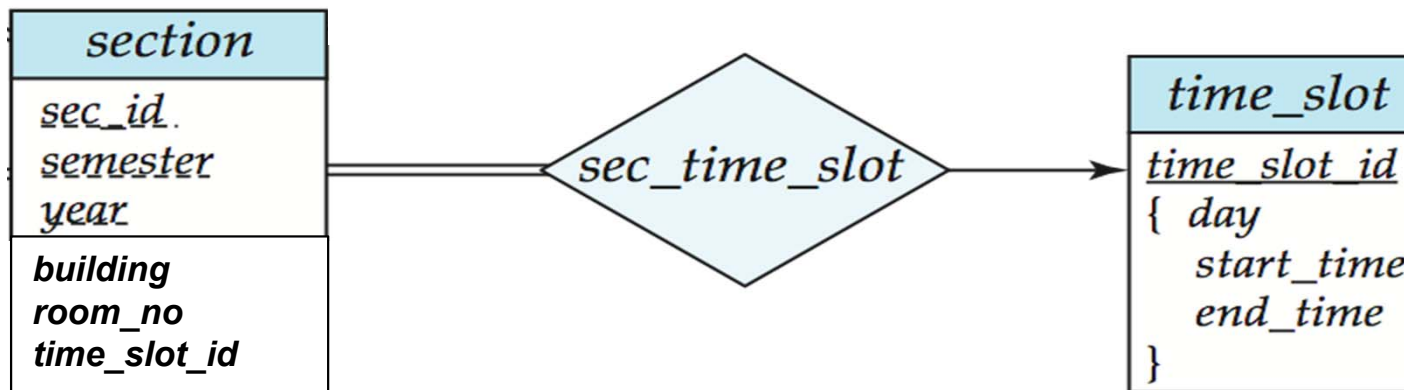
왜 이렇게 할까? → 만약 다중-값 속성을 앞과 동일한 방법으로 하나의 독립된 릴레이션으로 만들면 결국 동일한 형태의 스키마가 됨. 따라서 의미 없음!
그래서 속성의 특징을 이용하여 복합 속성을 flat화 시키는 방법을 이용할 수 있다.

time_slot은 분반시간표이므로 같은 요일, 같은 시간에 시작될 수 없다
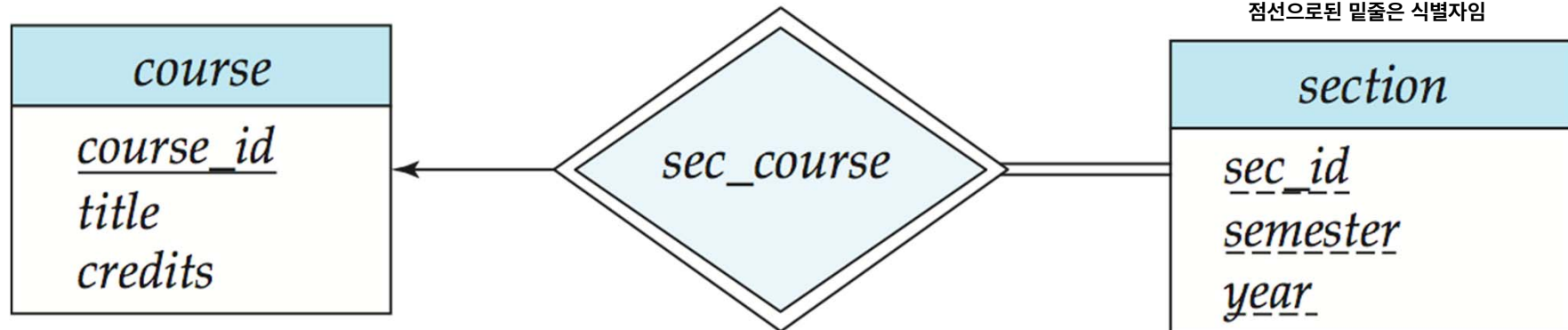
Representing Weak Entity Sets

관련된 문제가 나올수 있음

A weak entity set becomes a table that includes a column for the primary key of the identifying strong entity set

section (course_id, sec_id, sem, year)

주키가 없고 식별자만 있는 경우, 식별자인 것을 전부
묶어서 하나의 주키로 취급

점선으로된 밑줄은 식별자임

sec_course 스키마 = (*course_id, sec_id, sem, year*)

객체집합인 section의 스키마와 관계집합인 sec_course의 스키마가 동일하다.
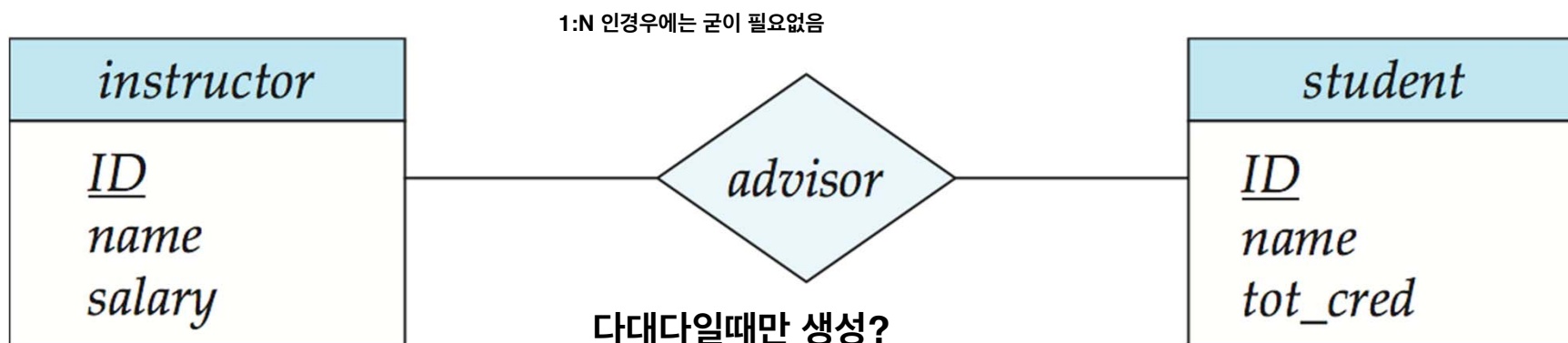→ 중복 제거 필요성   즉, **cid**와 **sid**만 있으면 되는데 불필요한 중복이 발생됨

Representing Relationship Sets

A many-to-many relationship set is represented as a schema with attributes for the primary keys of the two participating entity sets, and any descriptive attributes of the relationship set.

Example: schema for relationship set *advisor*

advisor = (*s_id, i_id*) + (설명속성, 예 date) 이때, **advisor은 N:M관계를 가질때만 가질수 있음**

1:N 인경우에는 굳이 필요없음

instructor		student
ID name salary	advisor	ID name tot_cred

다대다일때만 생성?

1:N: s_id

Redundancy of Schemas

Many-to-one and one-to-many relationship sets that are total on the many-side can be represented by adding an extra attribute to the "many" side, containing the primary key of the "one" side

Example: Instead of creating a schema for relationship set *inst_dept*, add an attribute *dept_name* to the schema arising from entity set *instructor*

몇개를 지울수있는지 문제가 나올수있음
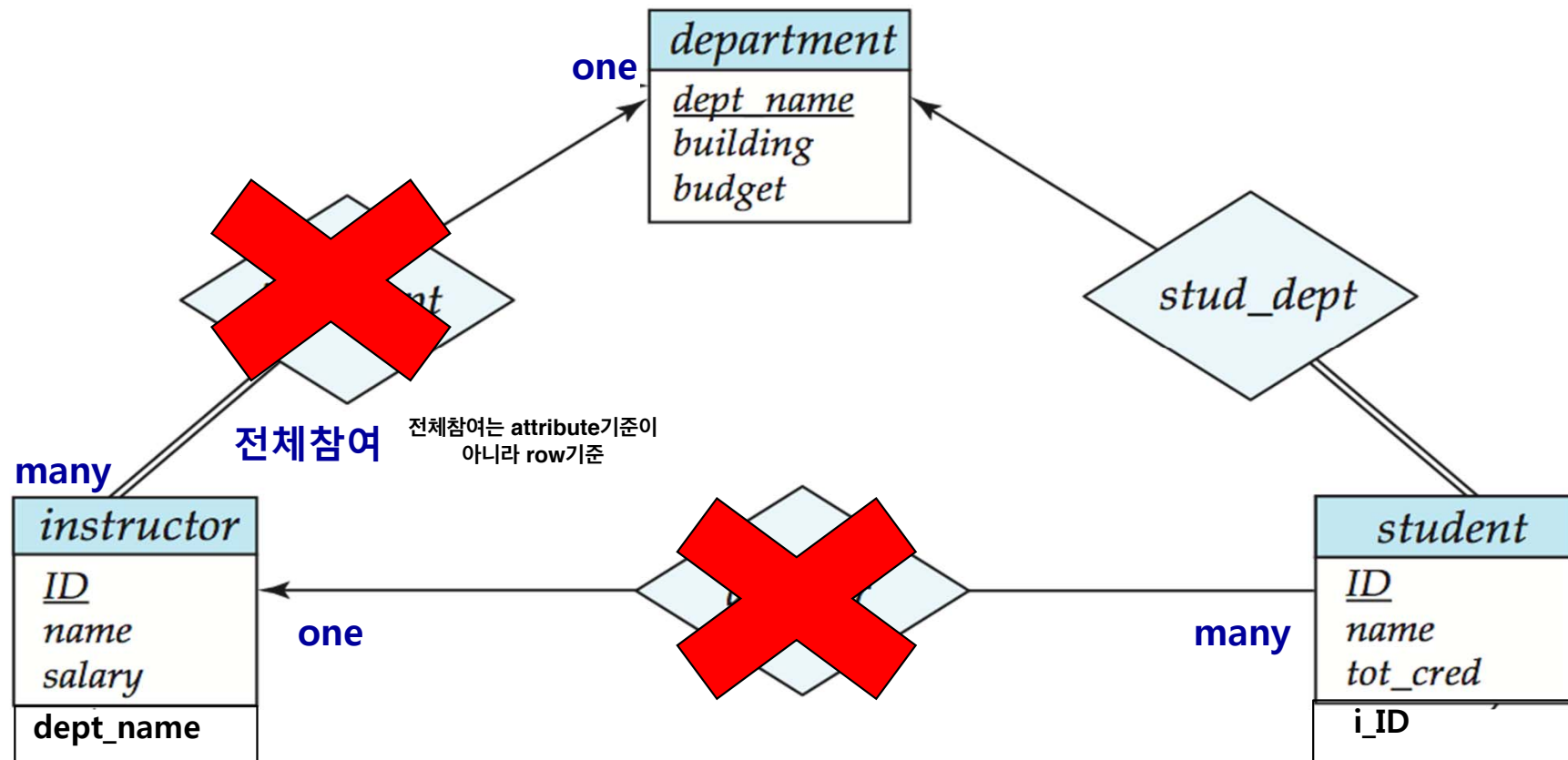
Redundancy of Schemas (Cont.)

For one-to-one relationship sets, either side can be chosen to act as the "many" side   즉, 비교하는 키를 어느쪽에 둬도 상관없음

That is, extra attribute can be added to either of the tables corresponding to the two entity sets

→ 관계에 참여하는 양쪽 릴레이션 중 한쪽에 다른쪽 주키를 넣고, 관계를 삭제!

If participation is *partial* on the "many" side, replacing a schema by an extra attribute in the schema corresponding to the "many" side could result in null values   1:M 인경우 many쪽에 관계되는 1쪽의 키를 넣음

→ instructor(one)와 student(many) 에서 student가 부분 참여이면, 지도교수님이 없는 학생이 있을 수도 있다는 것을 의미하며 이 경우 student에 삽입된 instructor의 ID는 null이 된다

The schema corresponding to a relationship set linking a weak entity set to its identifying strong entity set is redundant.

Example: The *section* schema already contains the attributes that would appear in the *sec_course* schema

→ 앞의 예에서 course와 section을 연결하는 sec_course 스키마의 경우 이미 section schema와 동일

7.7 Entity-Relationship Design Issues

Use of entity sets vs. attributes
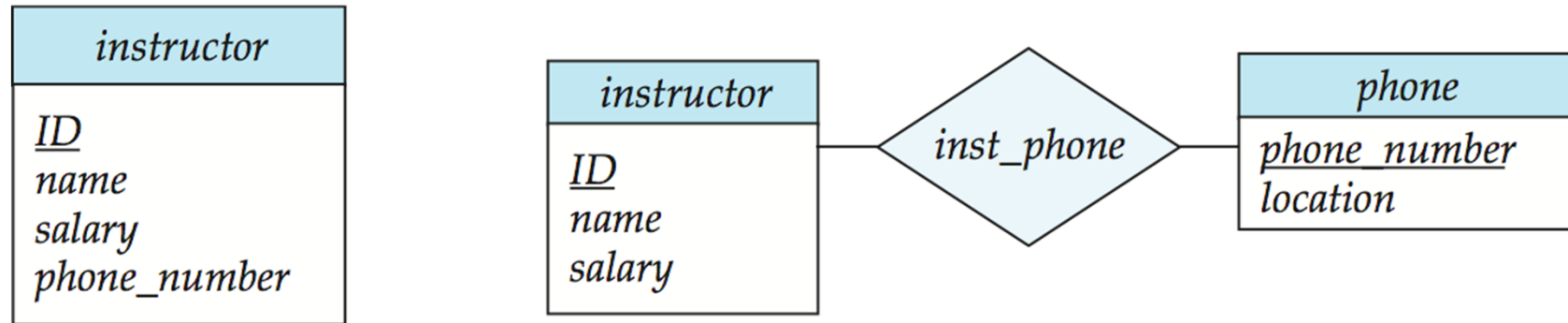
Use of phone as an entity allows extra information about phone numbers (plus multiple phone numbers)
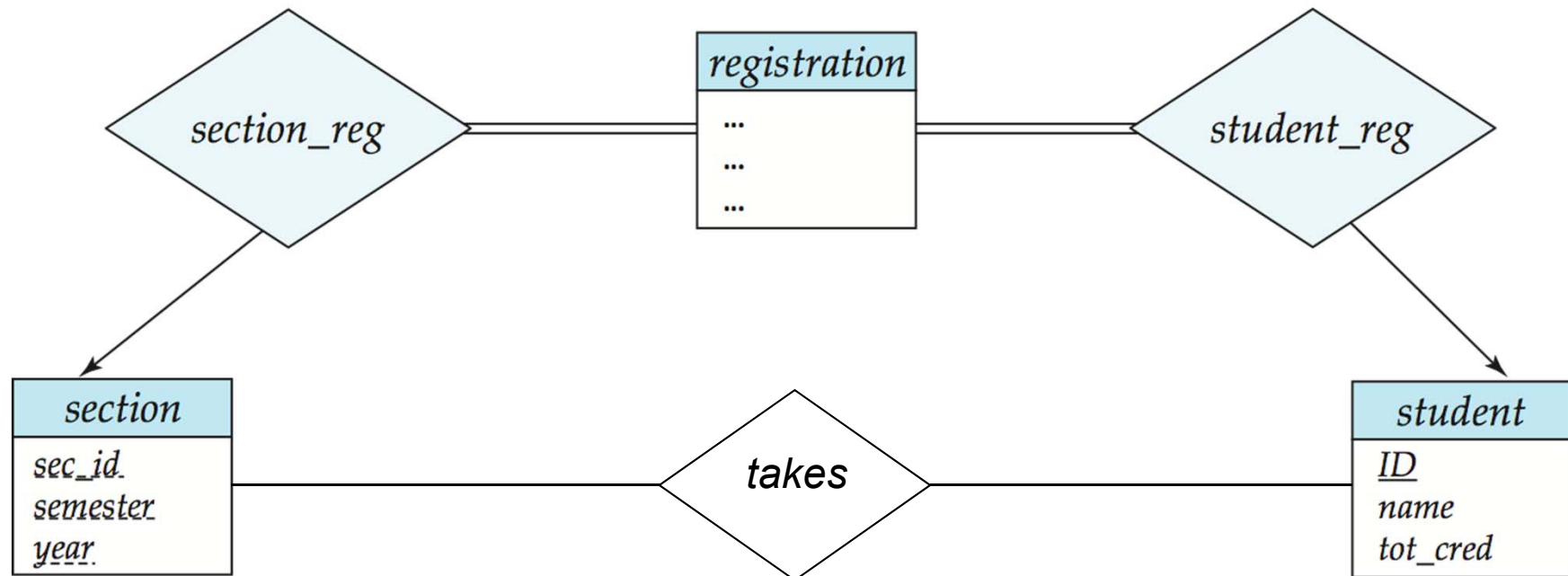
Entity-Relationship Design Issues (Cont.)

Use of entity sets vs. relationship sets

Possible guideline is to designate a relationship set to describe an action that occurs between entities

수업등록기록→ 각 section에 대한 학생들의
등록현황을 파악하는데 더 많은 정보 제공 가능

takes 냐 registration ?

Entity-Relationship Design Issues (Cont.)

Binary versus n-ary relationship sets

Although it is possible to replace any nonbinary (*n*-ary, for *n* > 2) relationship set by a number of distinct binary relationship sets, a *n*-ary relationship set shows more clearly that several entities participate in a single relationship.

Placement of relationship attributes

e.g., attribute *date* as attribute of *advisor* or as attribute of *student*

instructor-to-student (one-to-many) 의 경우
→ advisor의 설명속성 date를 many측(student)의 속성으로 옮기는 것이 가능
→ 일대일의 경우는 어느 쪽으로든 옮길 수 있다.

N:M관계에서는 date를 옮기는 것이 불가능

Binary Vs. Non-Binary Relationships

Some relationships that appear to be non-binary may be better represented using binary relationships

E.g., A ternary relationship *parents*, relating a child to his/her father and mother, is best replaced by two binary relationships, *father* and *mother*

▸ Using two binary relationships allows partial information (e.g., only mother being know)

But there are some relationships that are naturally non-binary

▸ Example: *proj_guide*

Converting Non-Binary Relationships to Binary Form

In general, any non-binary relationship can be represented using binary relationships by creating an artificial entity set.

Replace R between entity sets A, B and C by an entity set E, and three relationship sets:

1. R_A, relating E and A 2. R_B, relating E and B
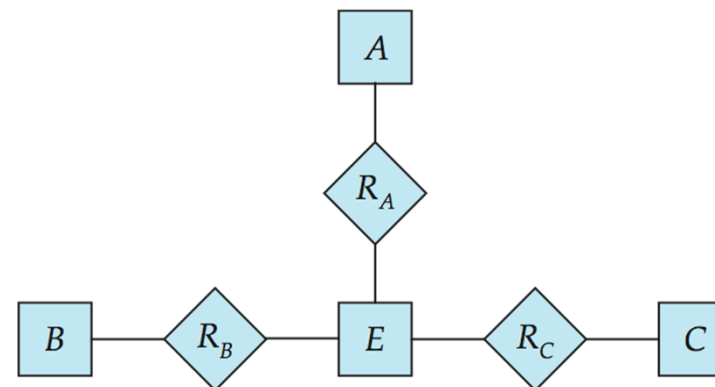3. R_C, relating E and C

Create a special identifying attribute for E

Add any attributes of R to E

For each relationship (a_i, b_i, c_i) in R, create

1. a new entity e_i in the entity set E 2. add (e_i, a_i) to R_A

3. add (e_i, b_i) to R_B 4. add (e_i, c_i) to R_C
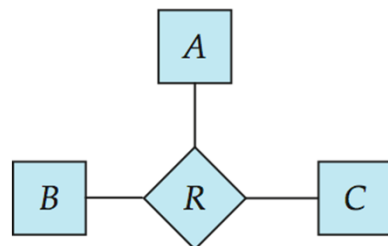
(a)

(b)

참고: Alternative Notations for Modeling Data

Summary of Symbols Used in E-R Notation

Symbol	Description
E (rectangle)	entity set
R (diamond)	relationship set
R (double diamond)	identifying relationship set for weak entity set
R (diamond) = E (rectangle)	total participation of entity set in relationship

E	
A1	attributes: simple (A1), composite (A2) and multivalued (A3) derived (A4)
A2	
A2.1	
A2.2	
{A3}	
A4()	

E	
A1	primary key

E	
A1	discriminating attribute of weak entity set
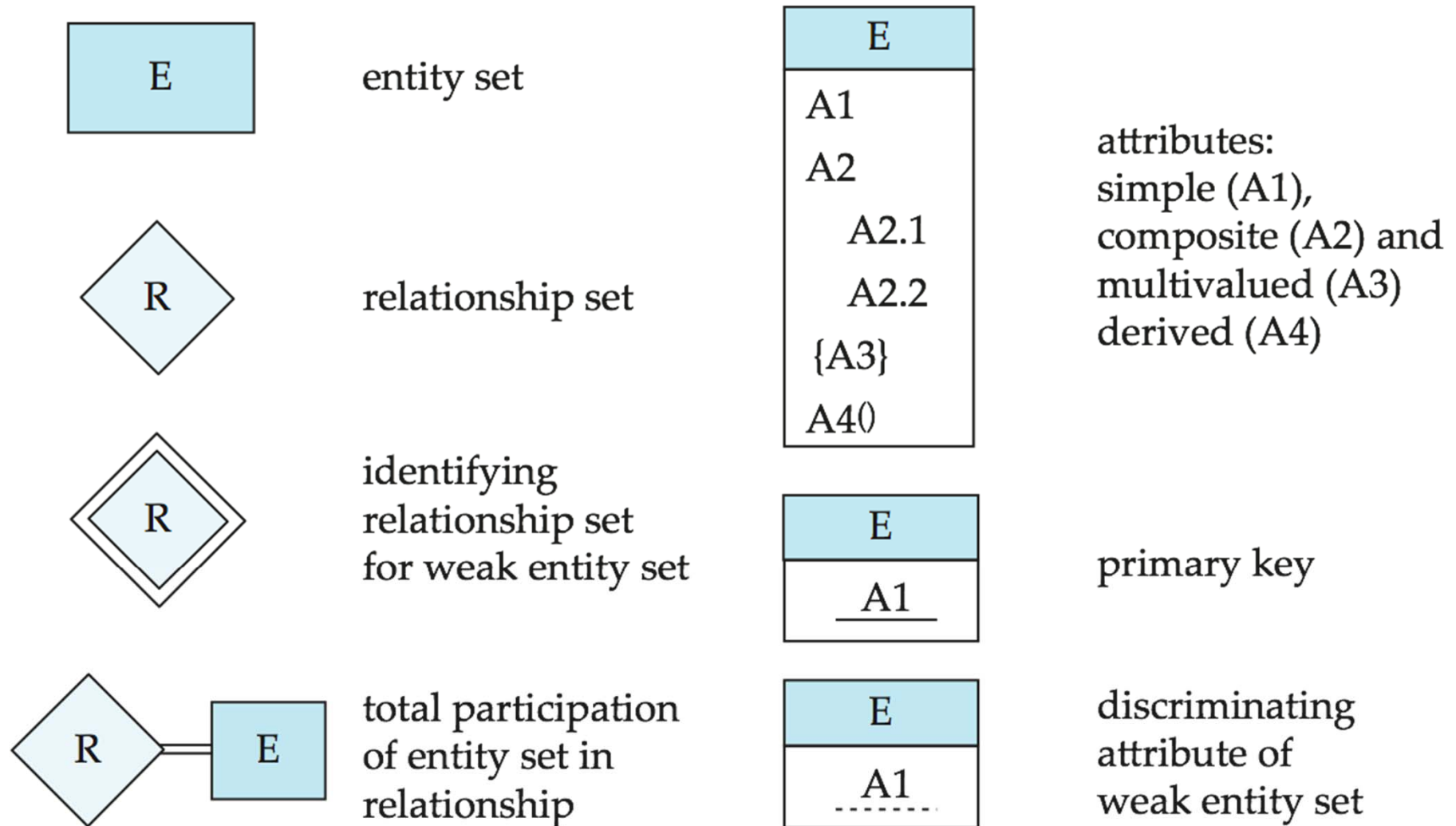
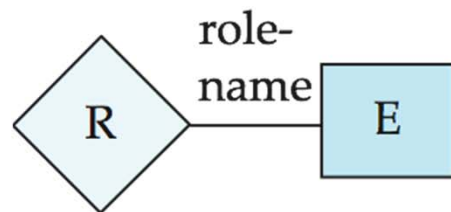# Symbols Used in E-R Notation (Cont.)
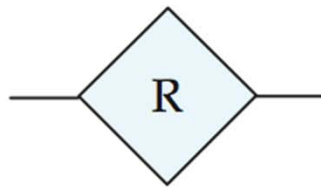
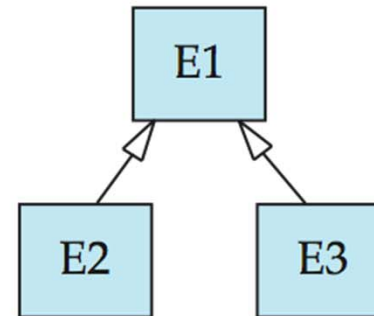many-to-many relationship
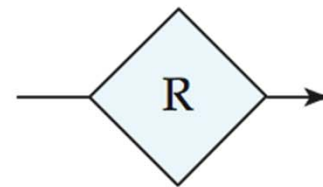
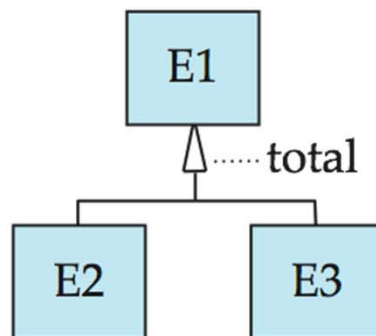many-to-one relationship
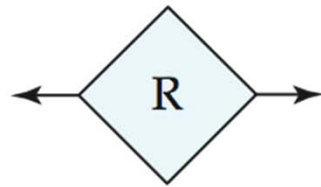
one-to-one relationship
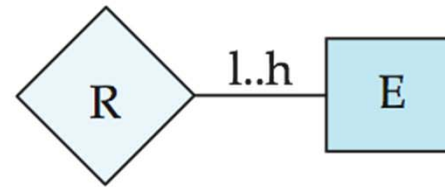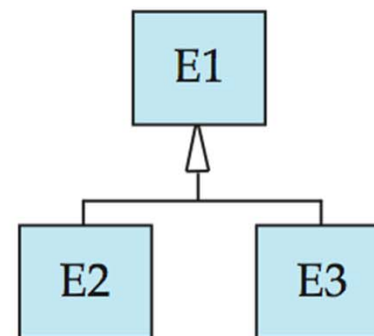
cardinality limits

role indicator

ISA: generalization or specialization

total (disjoint) generalization

disjoint generalization