國立陽明交通大學

NATIONAL YANG MING CHIAO TUNG UNIVERSITY

Biological Databases: Theories and Practice

430032

Database Schema and Entity-Relationship Model

Instructor: Prof. LEE, Tzong-Yi (李宗夷)
Email: leetzongyi@nycu.edu.ctw

Professor
Institute of Bioinformatics and Systems Biology,
National Yang Ming Chiao Tung University

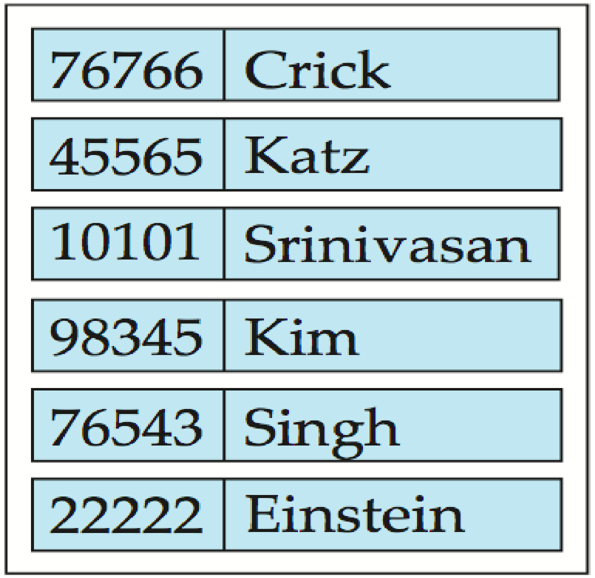
Outline

- Modeling
- Constraints
- E-R Diagram
- Weak Entity Sets
- Reduction to Relation Schemas
- Design Issues
- Extended E-R Features

Modeling

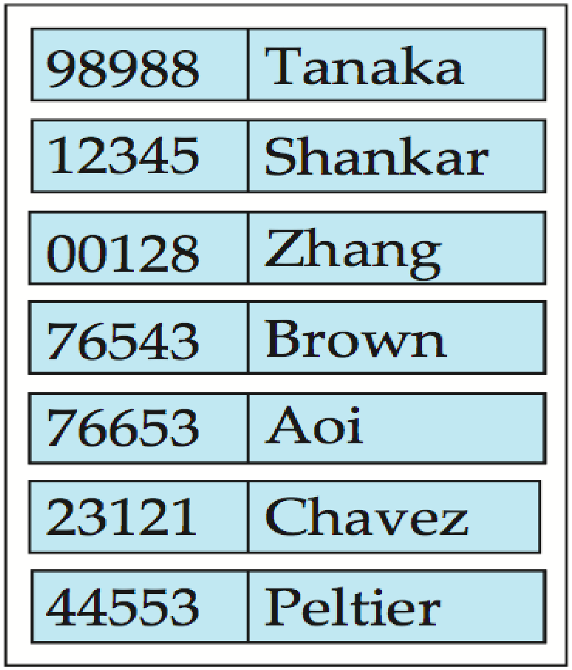
- A *database* can be modeled as:
  - a collection of entities,
  - relationship among entities.
- An **entity** is an object that exists and is distinguishable from other objects.
  - Example: specific person, company, event, plant
- Entities have **attributes**
  - Example: people have *names* and *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: *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)   *advisor*   22222 (Einstein)

  student entity   relationship set   *instructor* entity

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

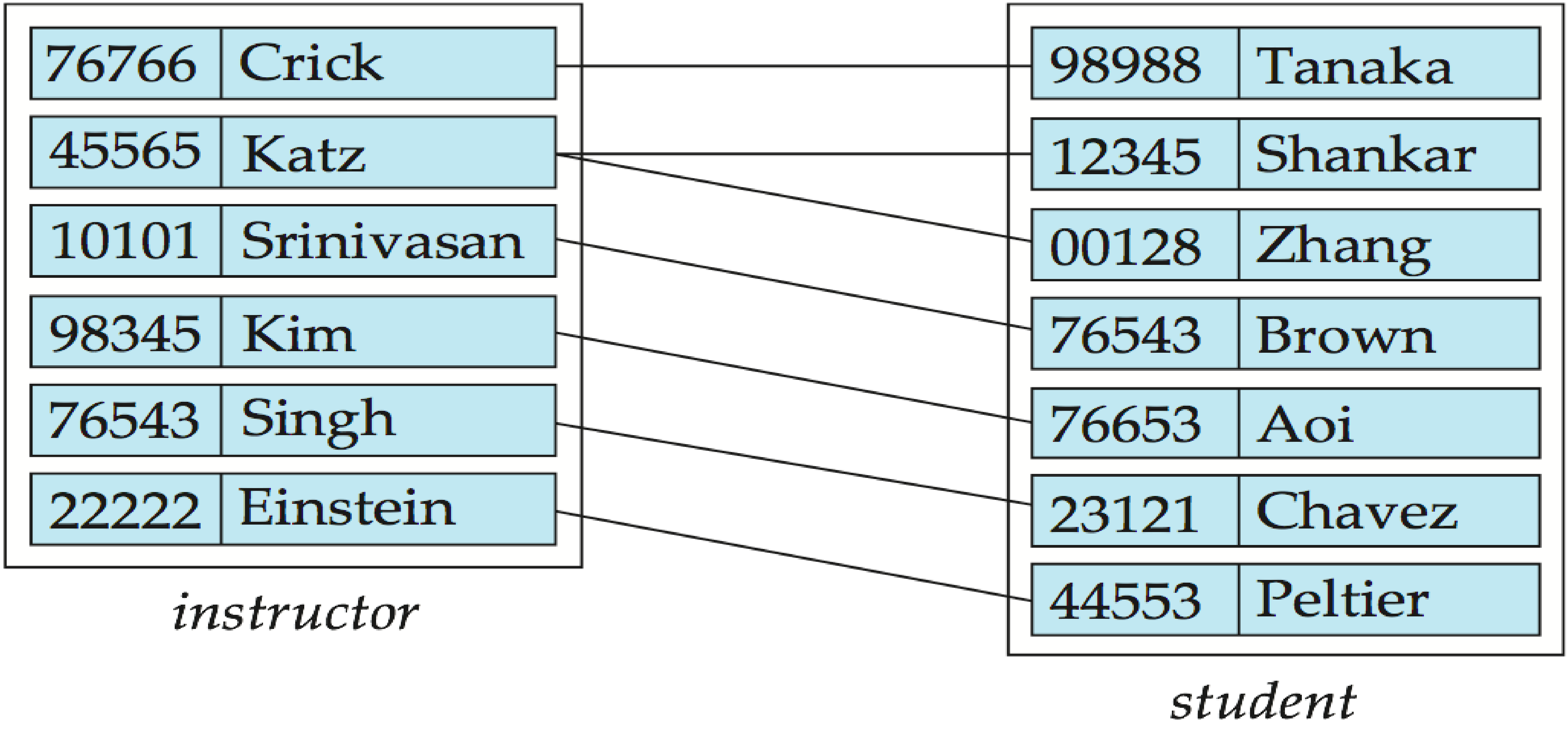
  $$\{(e_1, e_2, \ldots e_n) \mid e_1 \in E_1, e_2 \in E_2, \ldots, e_n \in E_n\}$$

  where $(e_1, e_2, \ldots, e_n)$ is a relationship

  - Example:

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

Relationship Set: *advisor*



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

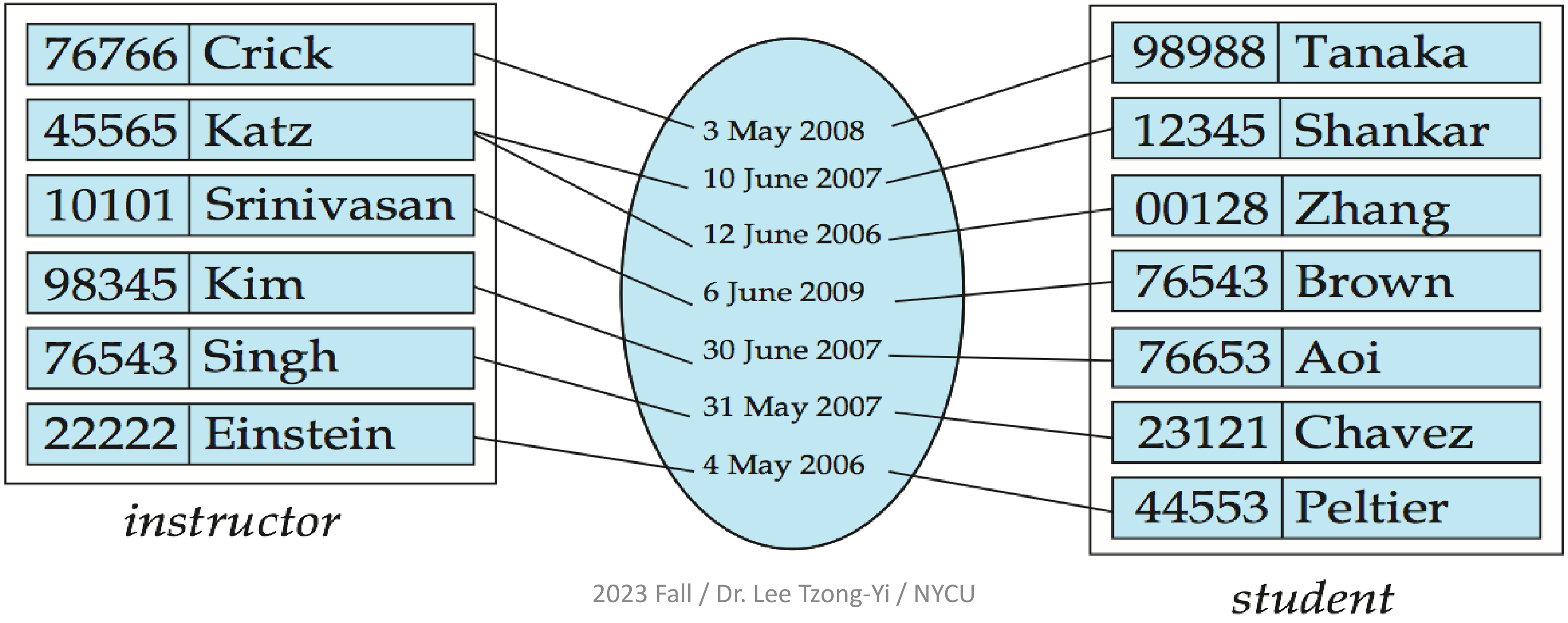
instructor

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

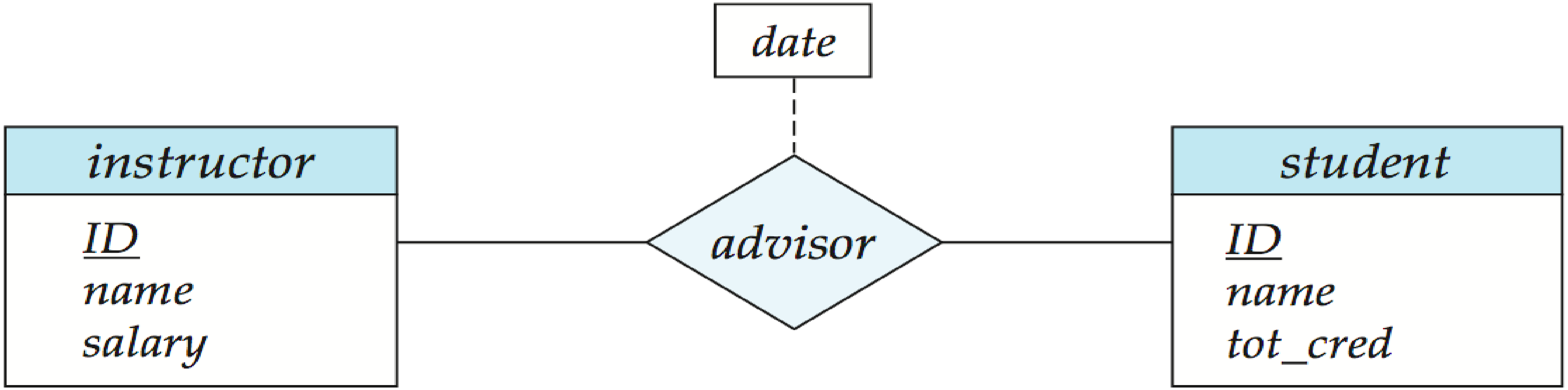
student

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



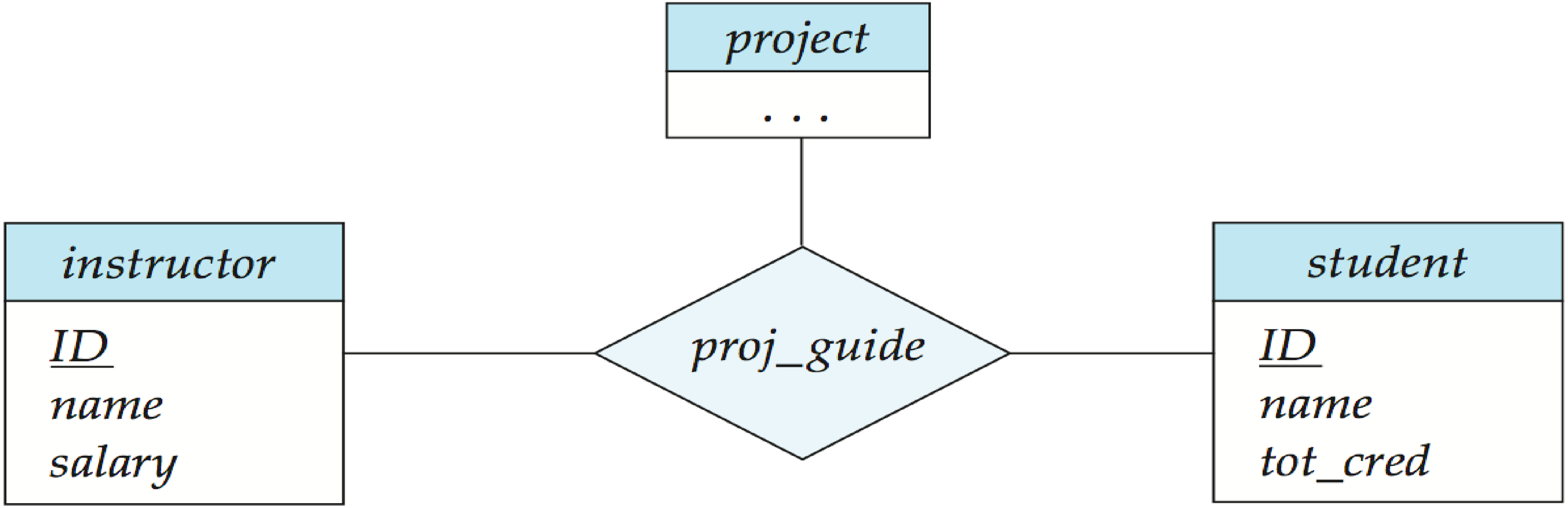
Relationship Sets with Attributes



Degree of a Relationship Set

- **Binary relationship**
  - involve two entity sets (or degree two).
  - most relationship sets in a database system are binary.
- Relationships between more than two entity sets are rare. Most relationships are binary.
- Example: *students* work on research *projects* under the guidance of an *instructor*.
  - relationship ***proj_guide*** is a **ternary relationship** between *instructor, student,* and *project*

E-R Diagram with a Ternary Relationship

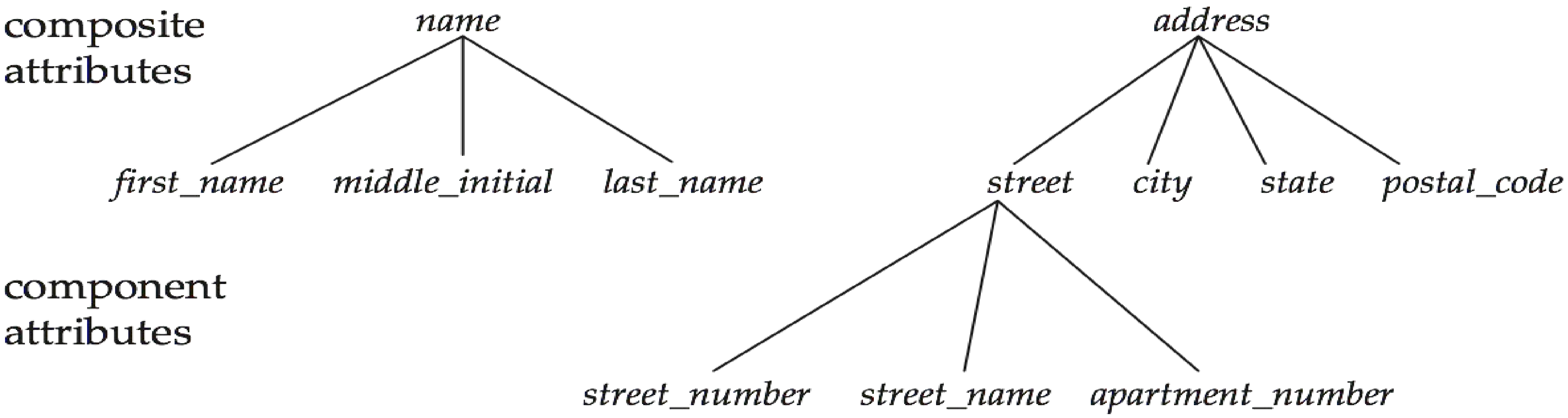


Attributes

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

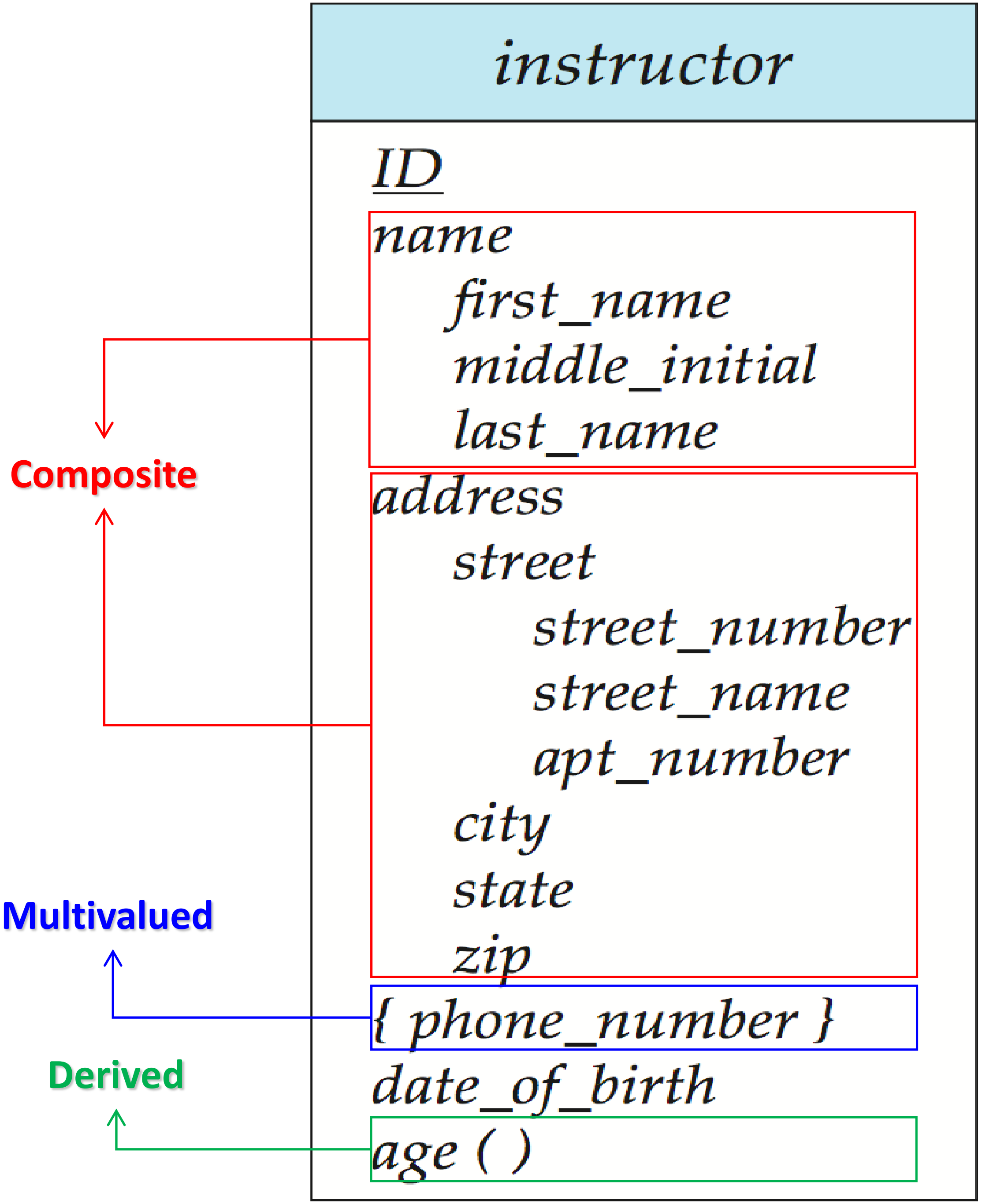
    instructor = (*ID, name, street, city, salary*)
    course= (*course_id, title, credits*)
- **Domain** – the set of permitted values for each attribute
- Attribute types:
  - **Simple** and **composite** attributes.
  - **Single-valued** and **multivalued** attributes
    - Example: multivalued attribute: *phone_numbers*
  - **Derived** attributes
    - Can be computed from other attributes
    - Example: ***age*** can be obtained from ***date_of_birth***

Composite Attributes



Example

The entity ***instructor*** contains composite, multivalued, and derived attributes.



instructor

ID

name

first_name

middle_initial

last_name

address

street

street_number

street_name

apt_number

city

state

zip

{ phone_number }

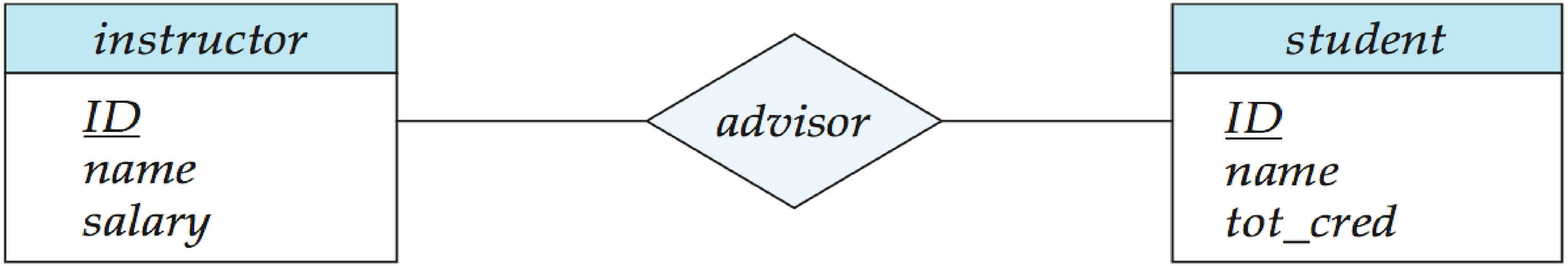
date_of_birth

age ( )

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**.

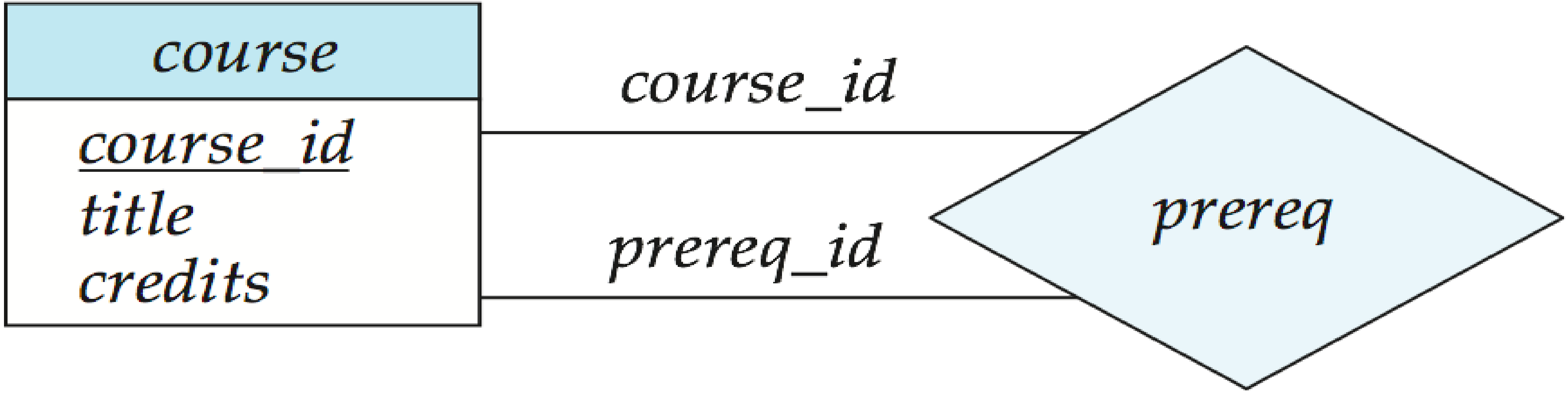
E-R Diagrams



- **Rectangles** represent **entity sets**.
- **Diamonds** represent **relationship sets**.
- **Attributes** listed inside entity rectangle
- **Underline** indicates **primary key** attributes

Roles

- We indicate roles in E-R diagram by labeling the lines that connect diamonds to rectangles.
- 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**.

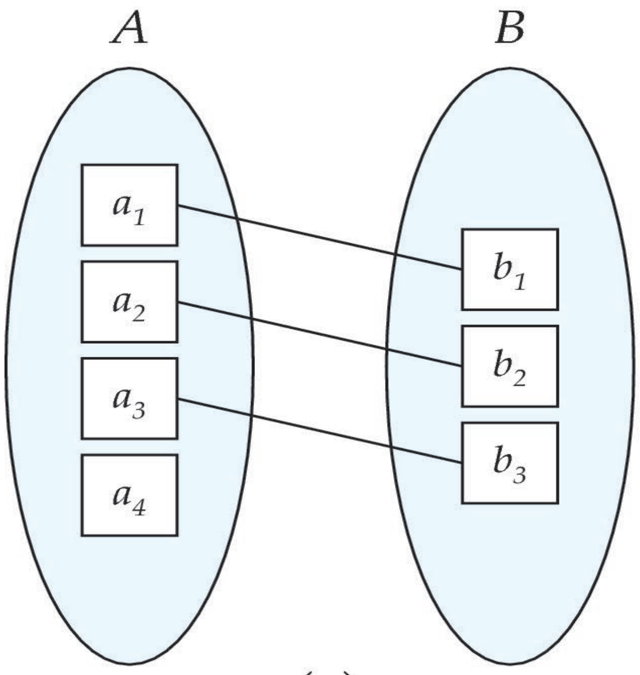


Mapping Cardinality Constraints

- 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**

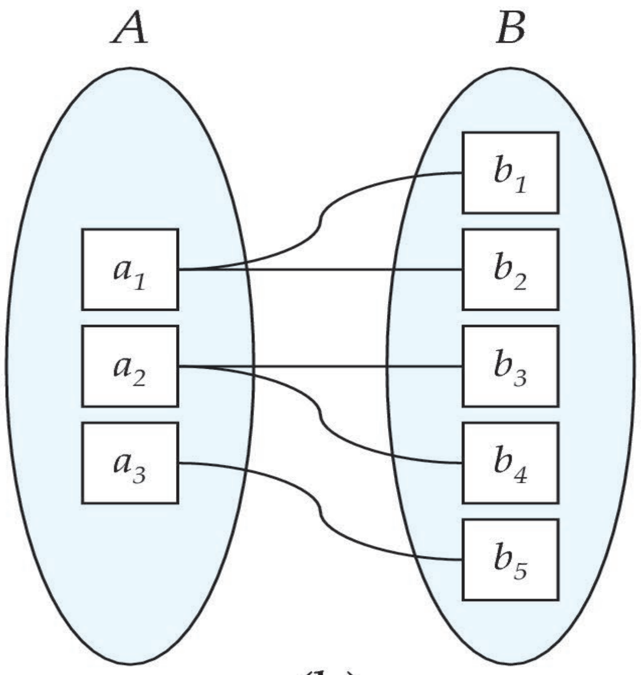
Mapping Cardinalities

One to one



(a)

One to many

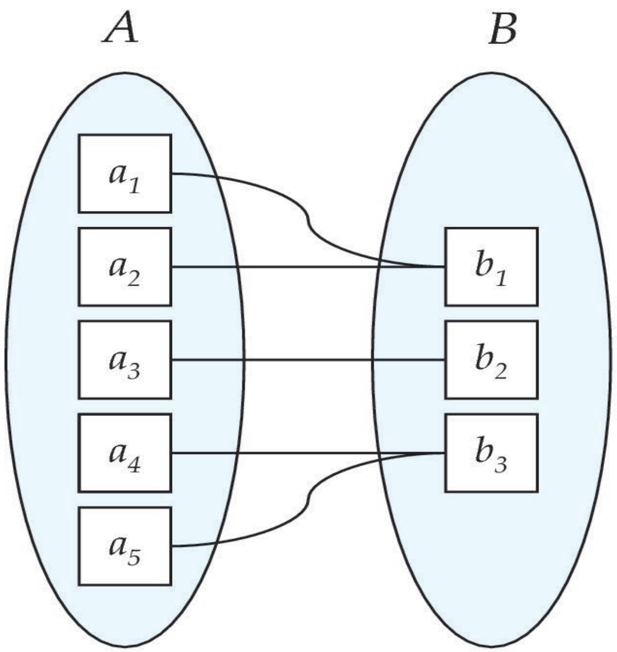


(b)

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

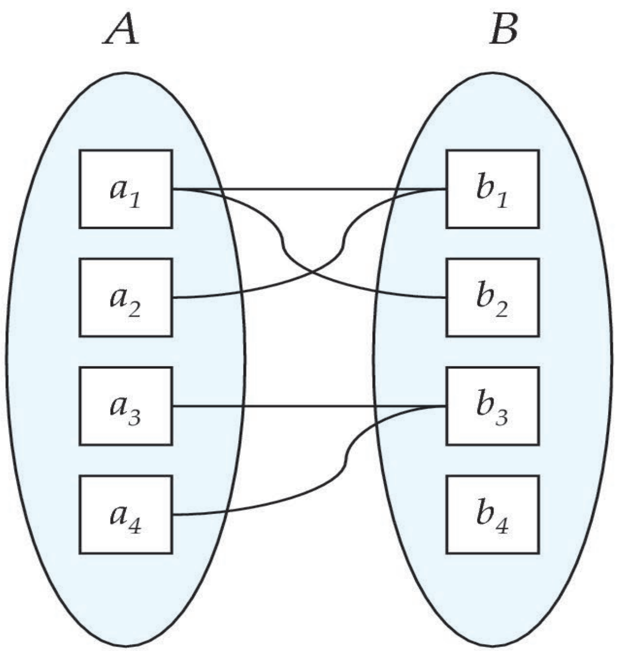
Mapping Cardinalities

Many to one



(a)

Many to many



(b)

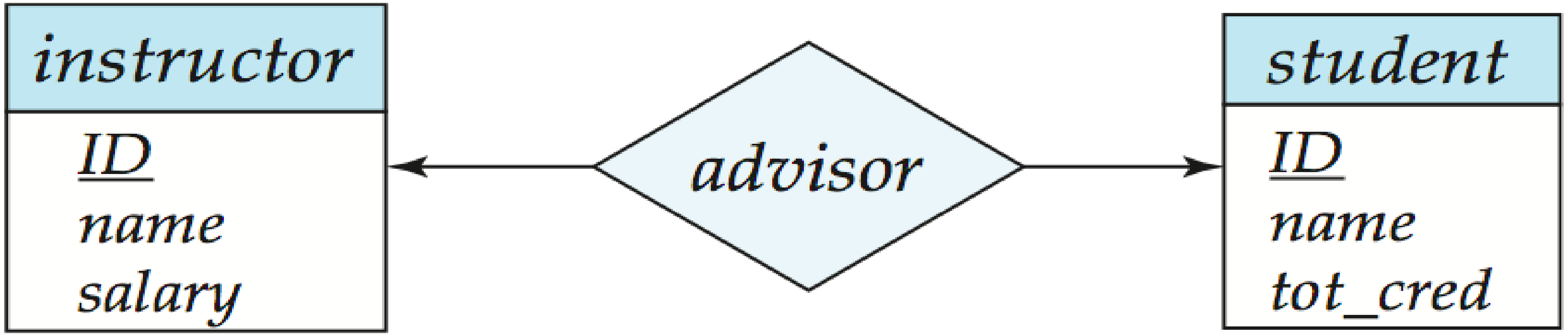
Note: Some elements in *A* and *B* may not be mapped to any elements in the other set

Cardinality Constraints

- We express cardinality constraints by drawing either a directed line (→), signifying "**one**," or an undirected line (—), signifying "**many**," between the relationship set and the entity set.
- 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 relationship ***stud_dept***

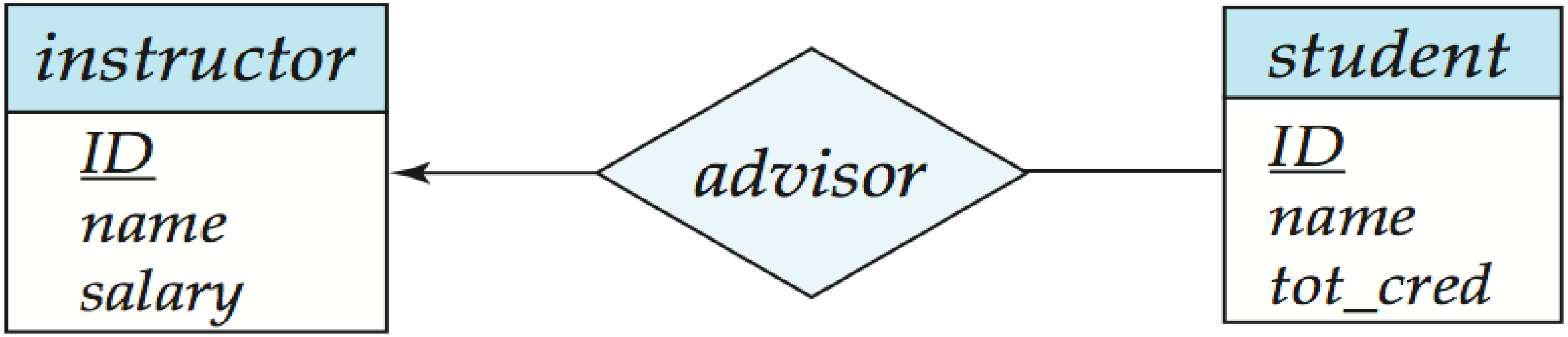
One-to-One Relationship

- 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***



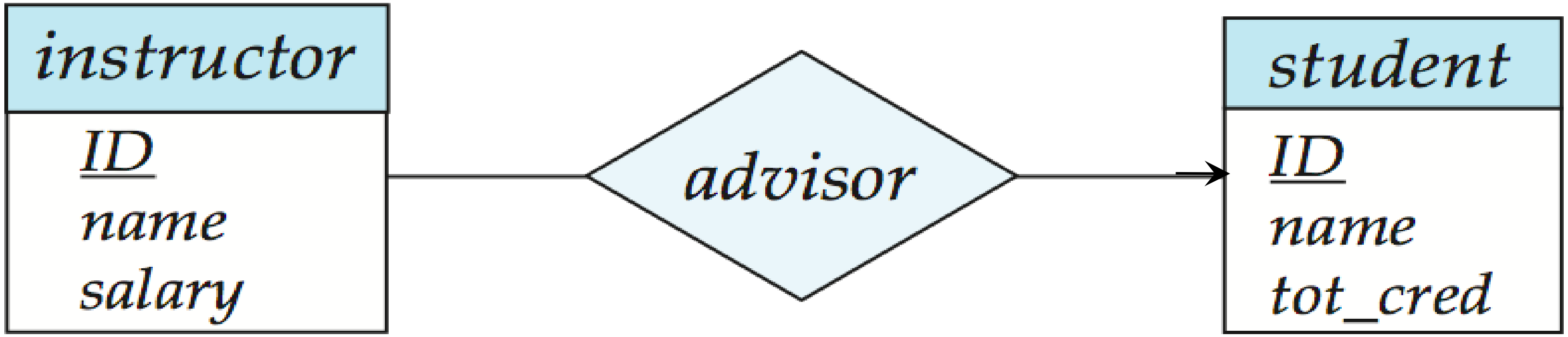
One-to-Many Relationship

- 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***



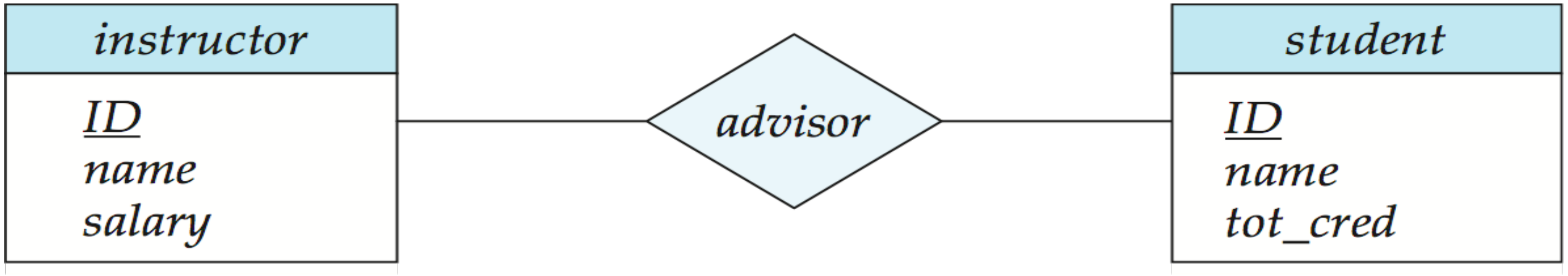
Many-to-One Relationships

- 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***



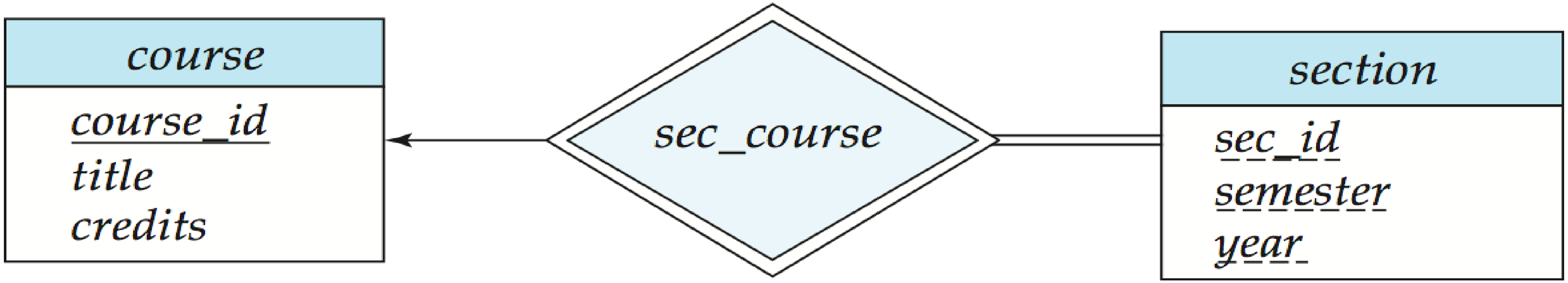
Many-to-Many Relationship

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



Participation of an Entity Set in a Relationship Set

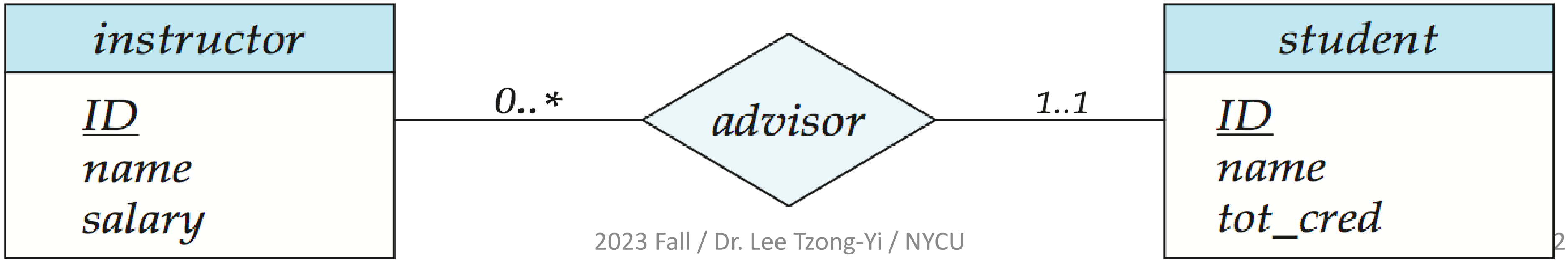
- **Total participation** (indicated by **double line**): every entity in the entity set participates in at least one relationship in the relationship set
  - E.g., participation of ***section*** in ***sec_course*** is total
    - every ***section*** must have an associated ***course***



- **Partial participation**: some entities may not participate in any relationship in the relationship set
  - Example: participation of ***instructor*** in ***advisor*** is partial

Alternative Notation for Cardinality Limits

- Cardinality limits can also express participation constraints:
  - The line with cardinality ***1…1*** between *advisor* and *student* means each student must have exactly one advisor.
  - The line with cardinality ***0…**** between *advisor* and *instructor* means an instructor can have zero or more students.
  - The mapping cardinality is **one-to-many** from *instructor* to *student*.
  - The participation of *student* in *advisor* is **total**.

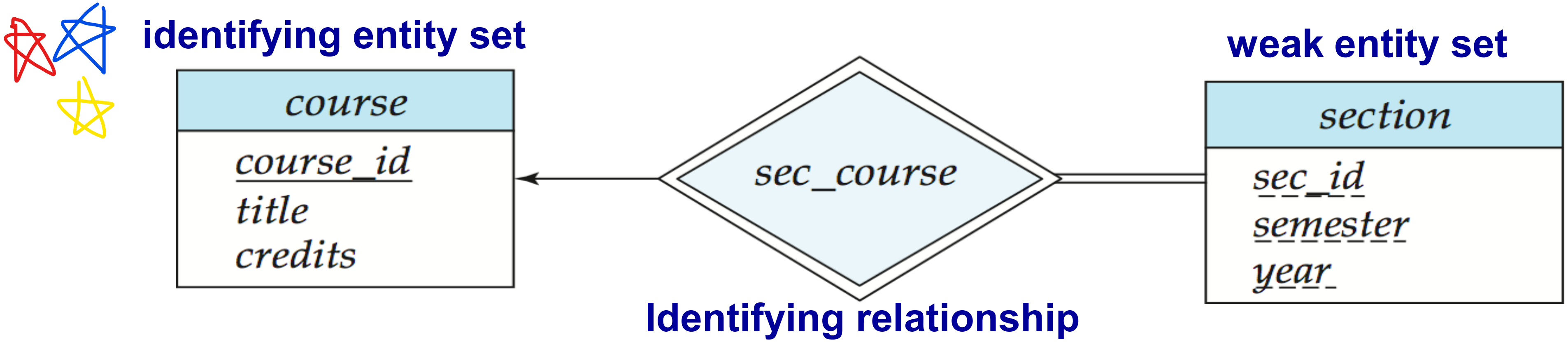


Weak Entity Sets

- An entity set that does not have a primary key is referred to as a **weak entity set**.
- The existence of a weak entity set depends on the existence of a **identifying entity set**
  - It must relate to the identifying entity set via a total, one-to-many relationship set from the identifying to the weak entity 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 among all the entities of a weak entity set.
- The primary key of a weak entity set is formed by the primary key of the **strong entity set** on which the weak entity set is existence dependent, plus the weak entity set's discriminator.

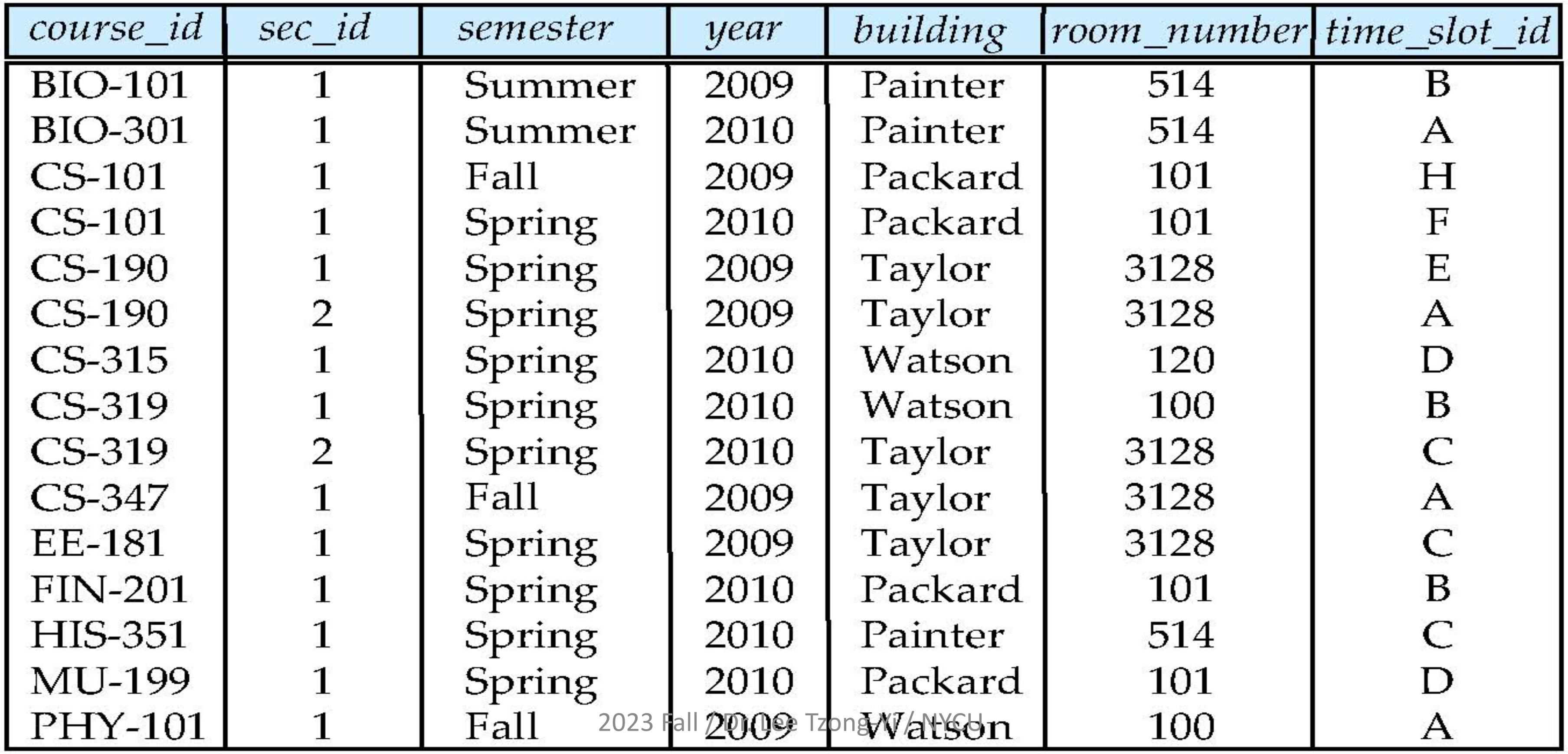
Weak Entity Sets (Cont.)

- 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*)



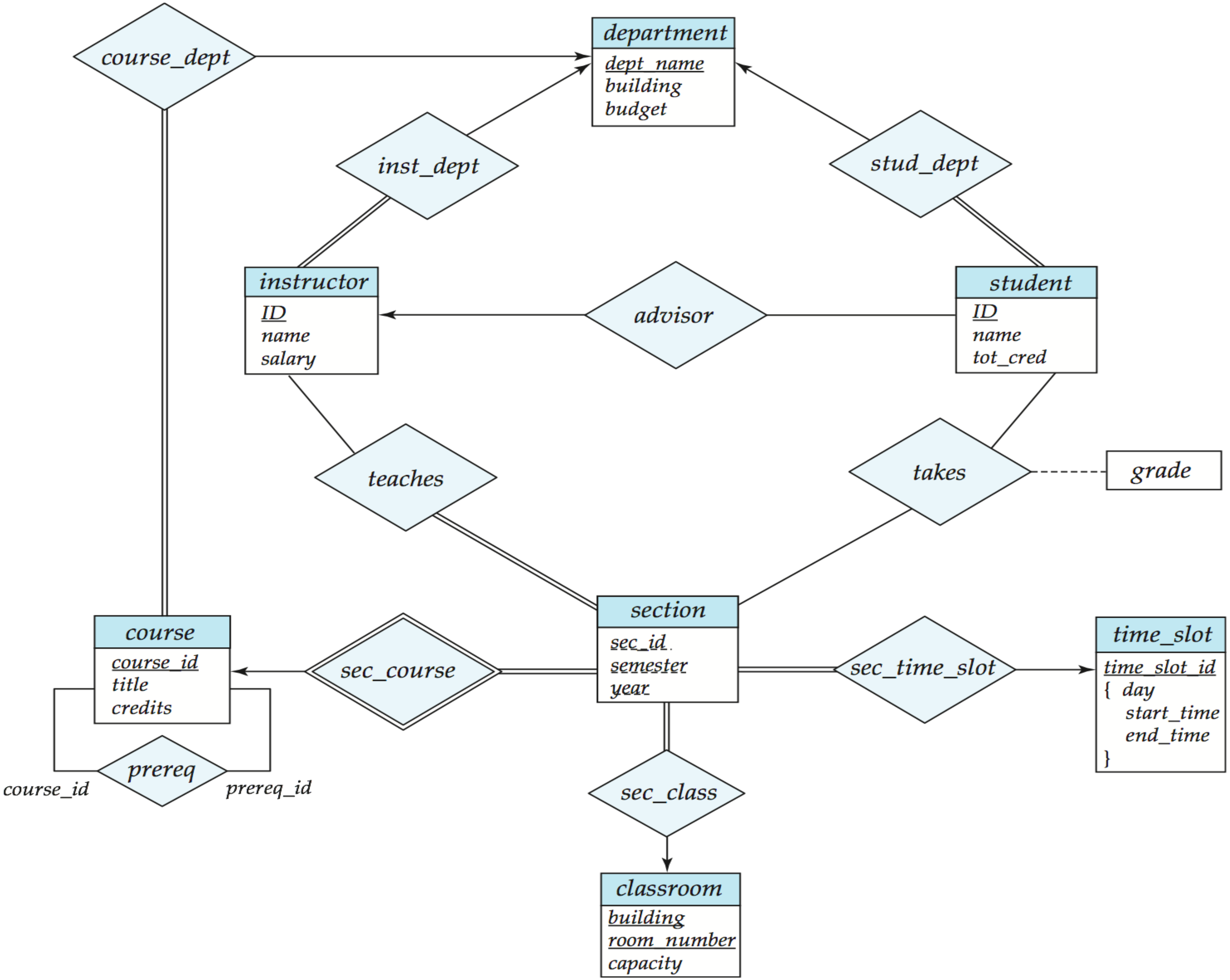
Weak Entity Sets (Cont.)

- Note: the primary key of the **strong entity set** is not explicitly stored with the **weak entity set**, since it is implicit in the **identifying relationship**.



course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2009	Painter	514	B
BIO-301	1	Summer	2010	Painter	514	A
CS-101	1	Fall	2009	Packard	101	H
CS-101	1	Spring	2010	Packard	101	F
CS-190	1	Spring	2009	Taylor	3128	E
CS-190	2	Spring	2009	Taylor	3128	A
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	Taylor	3128	C
CS-347	1	Fall	2009	Taylor	3128	A
EE-181	1	Spring	2009	Taylor	3128	C
FIN-201	1	Spring	2010	Packard	101	B
HIS-351	1	Spring	2010	Painter	514	C
MU-199	1	Spring	2010	Packard	101	D
PHY-101	1	Fall	2009	Watson	100	A

E-R Diagram for University Enterprise



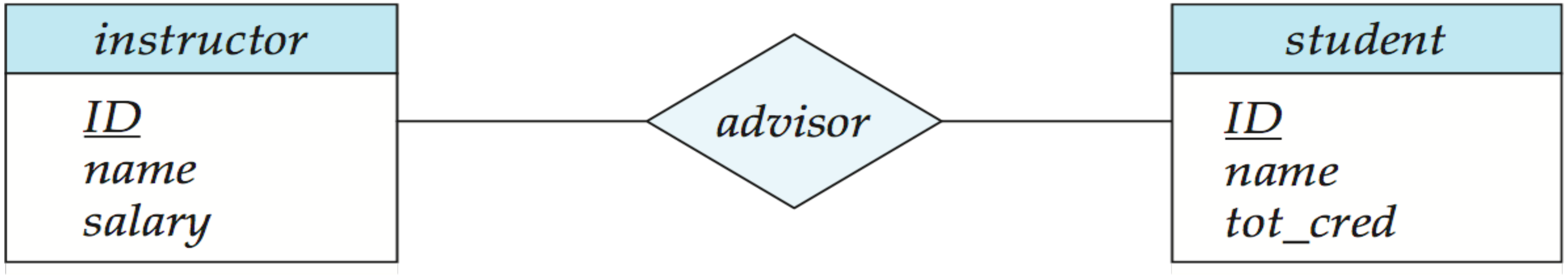
Reduction to Relation Schemas

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

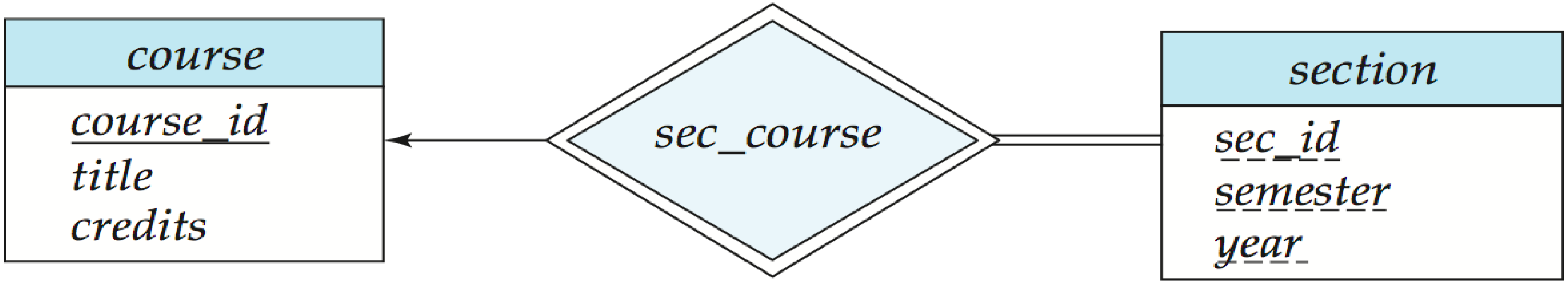
- A many-to-many relationship set is represented as a schema with attributes for the primary keys of the two participating entity sets, and any descriptive attributes of the relationship set.
- Example: schema for relationship set *advisor*

***advisor* = (*instructor_ID*, *student_ID*)**



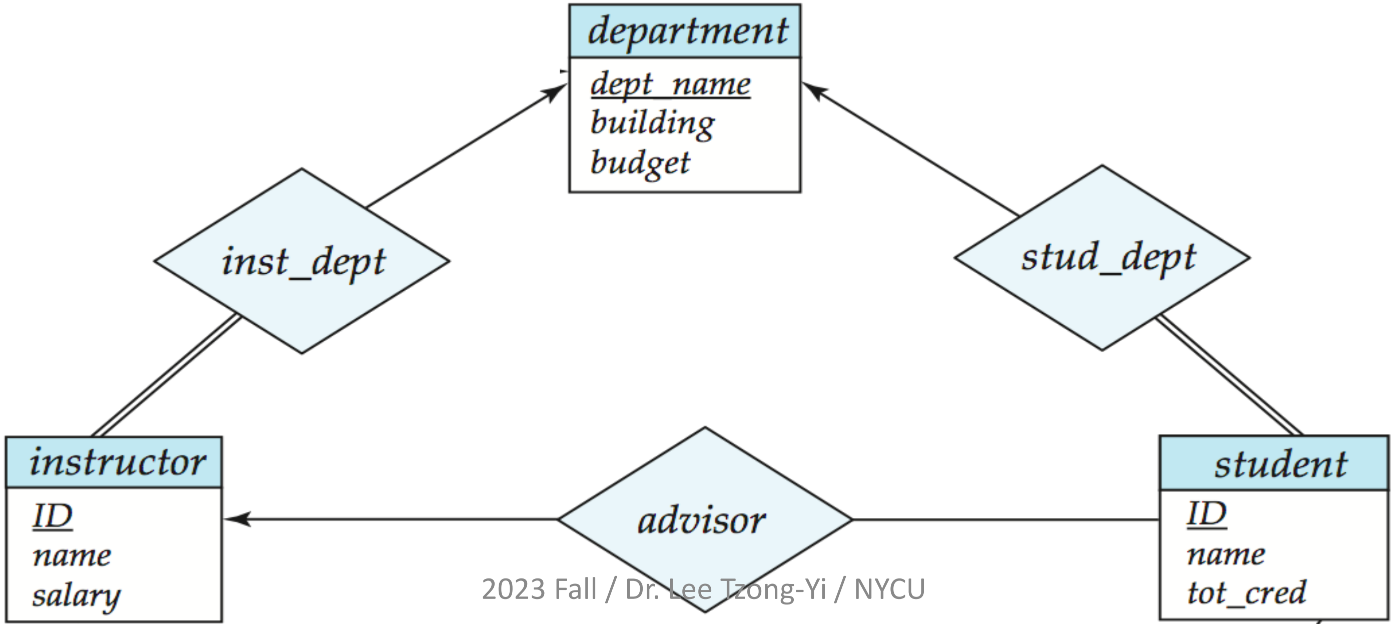
Representing Entity Sets With Simple Attributes

- A strong entity set reduces to a schema with the same attributes ***course = (course_id, title, credits)***
- 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, semester, year)***



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*



Redundancy of Schemas (Cont.)

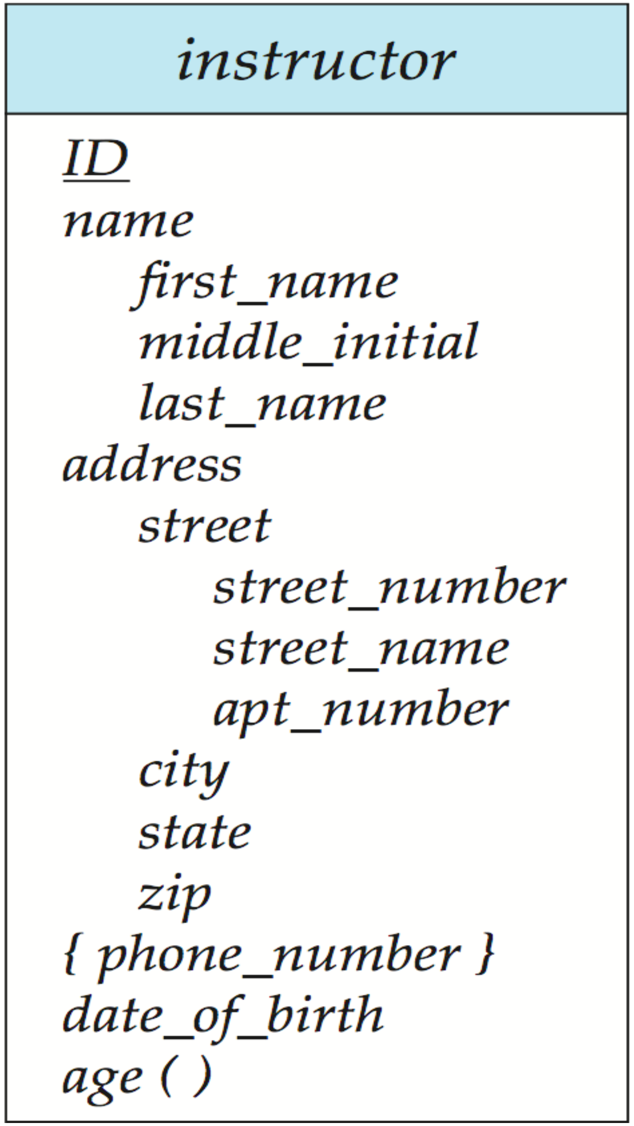
- For **one-to-one** relationship sets, either side can be chosen to act as the “many” side
  - That is, extra attribute can be added to either of the tables corresponding to the two entity sets
- If participation is ***partial*** on the “many” side, replacing a schema by an extra attribute in the schema corresponding to the “many” side could result in **null** values
- 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

Composite and Multivalued Attributes

- Composite attributes are flattened out by creating a separate attribute for each component attribute
  - Example: given entity set ***instructor*** with composite attribute ***name*** containing component attributes ***first_name***, ***middle_initial***, and ***last_name***
  - The schema has three attributes ***first_name***, ***middle_initial***, and ***last_name***
- Ignoring multivalued attributes, extended instructor schema is

  phone number is not here

  - *instructor(ID, first_name, middle_initial, last_name, street_number, street_name, apt_number, city, state, zip_code, date_of_birth)*



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 and Multivalued Attributes

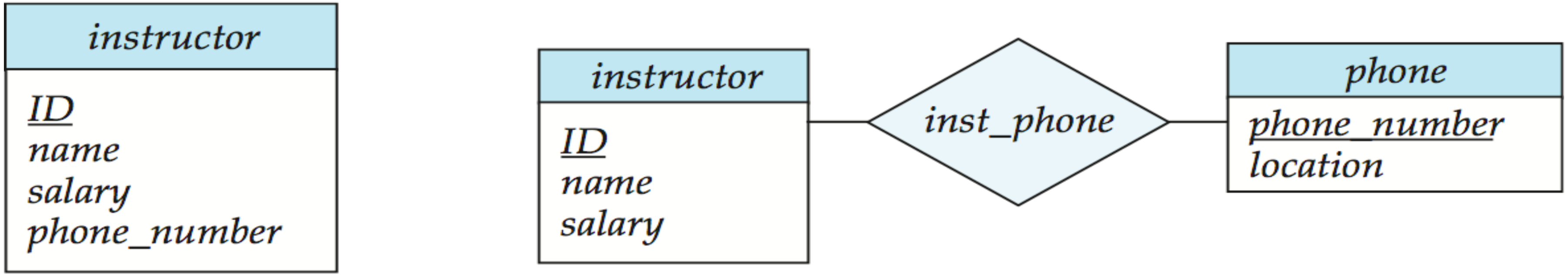
- A multivalued attribute *M* of an entity *E* is represented by a separate schema *EM*
  - Schema *EM* has attributes corresponding to the primary key of *E* and an attribute corresponding to multivalued attribute *M*
  - Example: Multivalued attribute ***phone_number*** of ***instructor*** is represented 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)

Design Issues

there is no certain answer

- **Use of entity sets vs. attributes**



- Use of phone as an entity allows extra information about phone numbers (plus multiple phone numbers)

Design Issues

- **Binary vs. n-ary relationship sets**
  Although it is possible to replace any non-binary (*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***

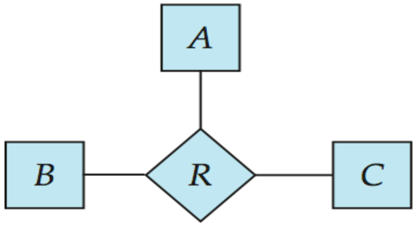
Converting Non-Binary Relationships to Binary Form

- 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:

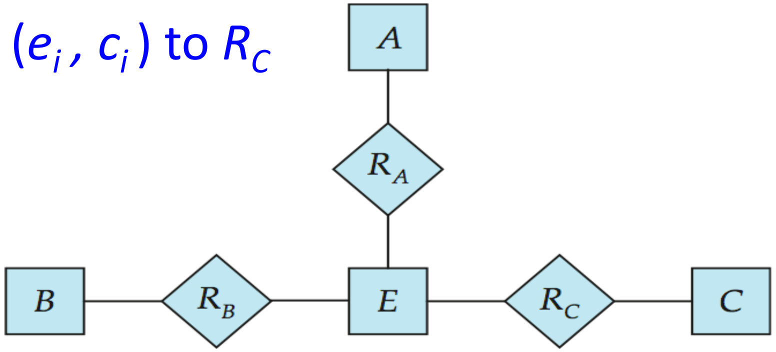
    1. R_A, relating *E* and *A* 2. R_B, relating *E* and *B* 3. R_C, relating *E* and *C*
  - Create a special identifying attribute for *E*
  - Add any attributes of *R* to *E*
  - For each relationship (a_i , b_i , c_i) in *R,* create

    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



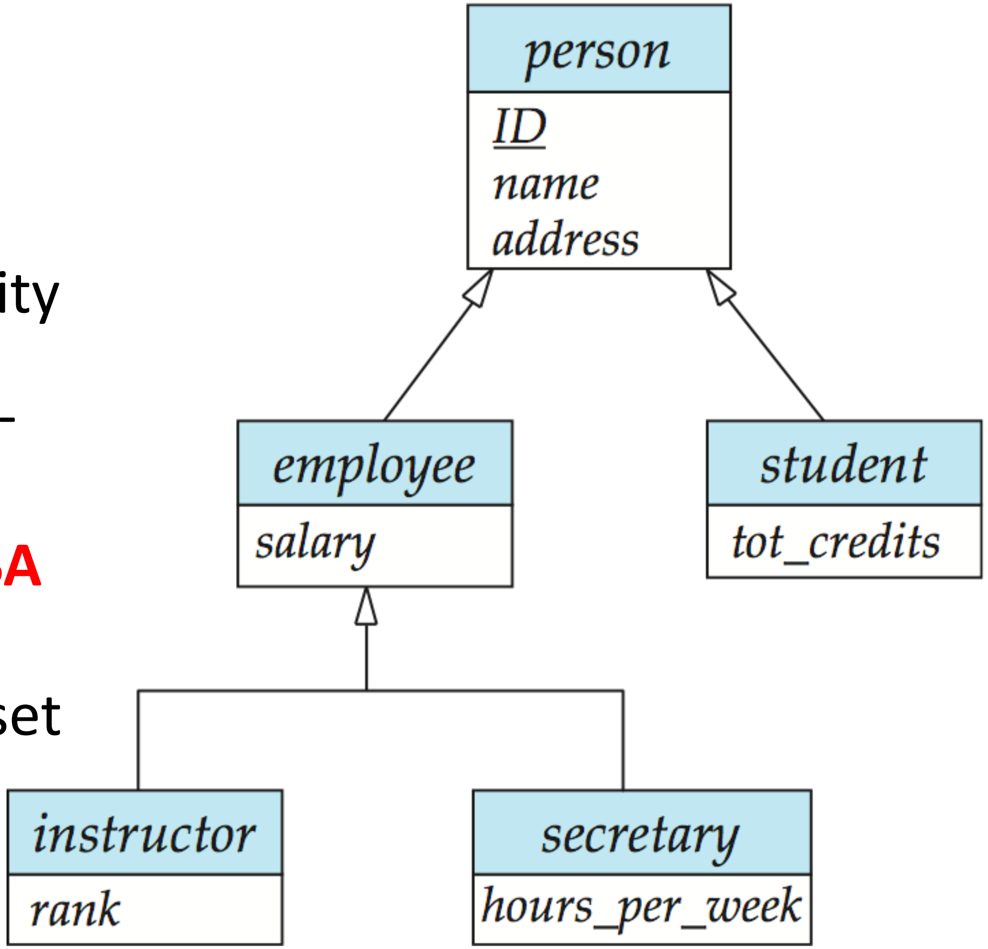
(a)



(b)

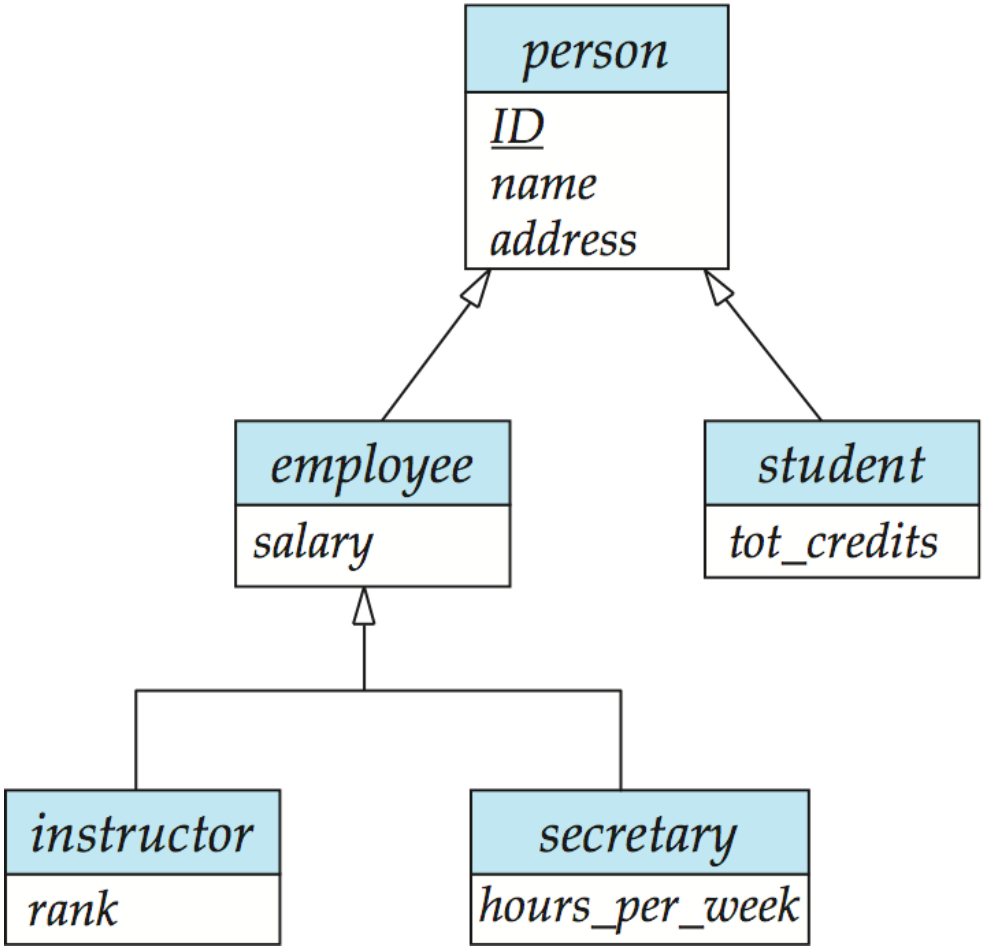
Extended E-R Features: Specialization

- **Top-down design process**: we designate subgroupings within an entity set that are distinctive from other entities in the set.
- These subgroupings become lower-level entity sets that have attributes or participate in relationships that do not apply to the higher-level entity set.
- Depicted by a *triangle* component labeled **ISA** (E.g., *instructor* "is a" *person*).
- **Attribute inheritance** – a lower-level entity set **inherits** all the attributes and relationship participation of the higher-level entity set to which it is linked.



Extended ER Features: Generalization

- A **bottom-up design process** – combine a number of entity sets that share the same features into a higher-level entity set.
- Specialization and generalization are simple inversions of each other; they are represented in an E-R diagram in the same way.
- The terms **specialization** and **generalization** are used interchangeably.



Specialization and Generalization

- Can have **multiple specializations** of an entity set based on different features.
- E.g., *permanent_employee* vs. *temporary_employee*, in addition to *instructor* vs. *secretary*
- Each particular employee would be
  - a member of one of *permanent_employee* or *temporary_employee*,
  - and also a member of one of *instructor*, *secretary*
- The ISA relationship also referred to as **superclass - subclass** relationship

Representing Specialization via Schemas

- Method 1:
  - Form a schema for the higher-level entity
  - Form a schema for each lower-level entity set, include **primary key** of higher-level entity set and local attributes

schema	attributes
person	*ID, name, street, city*
student	*ID, tot_cred*
employee	*ID, salary*

  - **Drawback**: getting information about, an *employee* requires accessing two relations, the one corresponding to the low-level schema and the one corresponding to the high-level schema

Representing Specialization as Schemas (Cont.)

- Method 2:
  - Form a schema for each entity set with all local and inherited attributes

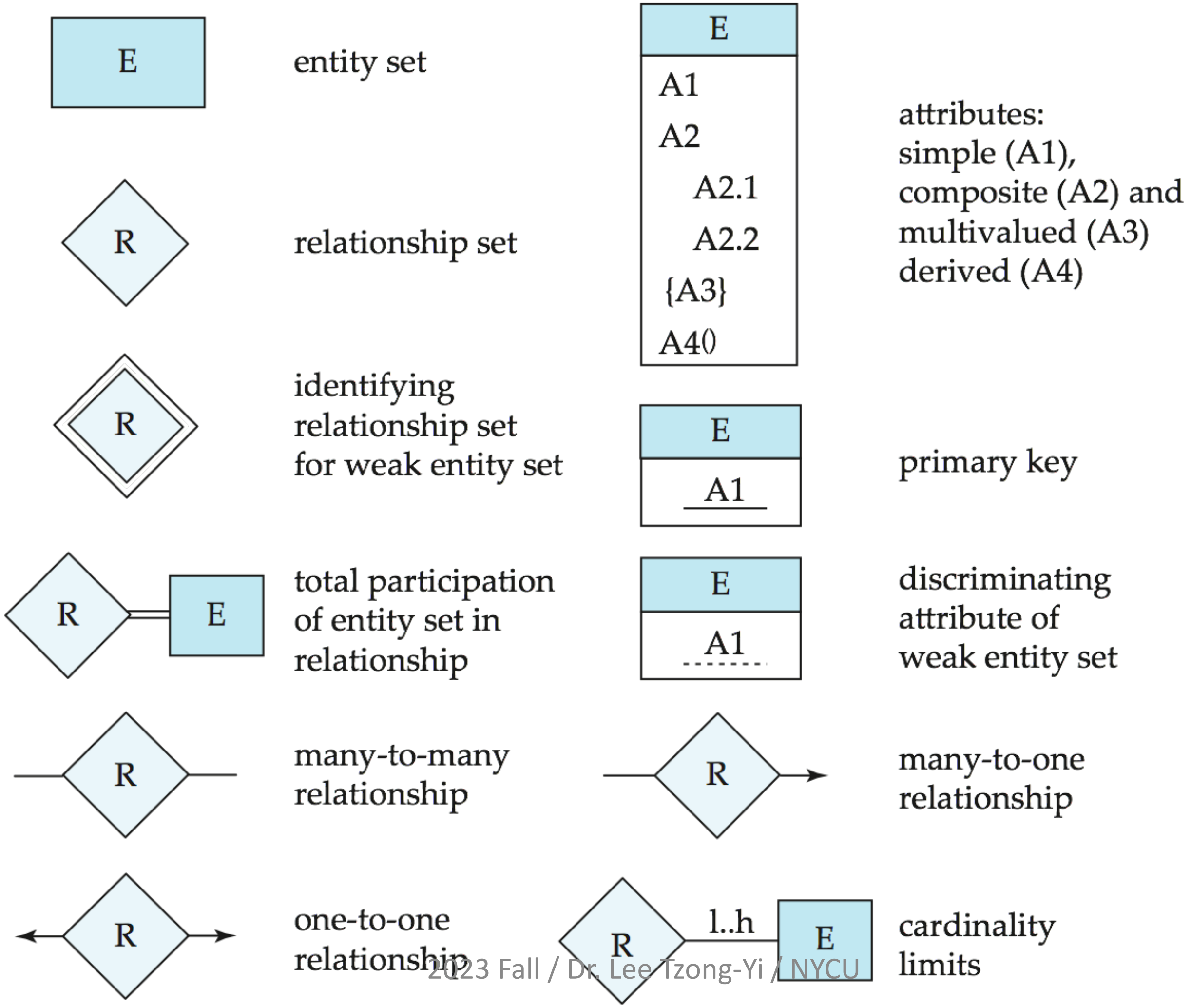
schema	attributes
person	*ID, name, street, city*
student	*ID, name, street, city, tot_cred*
employee	*ID, name, street, city, salary*

  - If specialization is total, the schema for the generalized entity set (*person*) not required to store information
    - Can be defined as a "view" relation containing union of specialization relations
    - But explicit schema may still be needed for foreign key constraints
  - **Drawback**: *name, street* and *city* may be stored redundantly for people who are both students and employees

E-R Design Issues

- The use of **an attribute** or **entity set** to represent an object.
- Whether a real-world concept is best expressed by an **entity set** or a **relationship set**.
- The use of a ternary relationship versus a pair of binary relationships.
- The use of a strong or weak entity set.
- The use of specialization/generalization – contributes to **modularity** in the design.

Summary of Symbols Used in E-R Notation



entity set

relationship set

identifying relationship set for weak entity set

total participation of entity set in relationship

many-to-many relationship

one-to-one relationship

attributes: simple (A1), composite (A2) and multivalued (A3) derived (A4)

primary key

discriminating attribute of weak entity set

many-to-one relationship

cardinality limits

Thank you and any questions?