Principles of Database Systems

Database Design and the E-R Model

Database Design and the E-R Model

- **Overview of the Design Process**
- **The Entity-Relationship Model**
- **Constraints**
- **Removing Redundant Attributes in Entity Sets**
- **Entity-Relationship Diagrams**
- **Reduction to Relational Schemas**
- **Entity-Relationship Design Issues**
- Extended E-R Features
- Alternative Notations for Modeling Data
- Other Aspects of Database Design

Overview of the Design Process

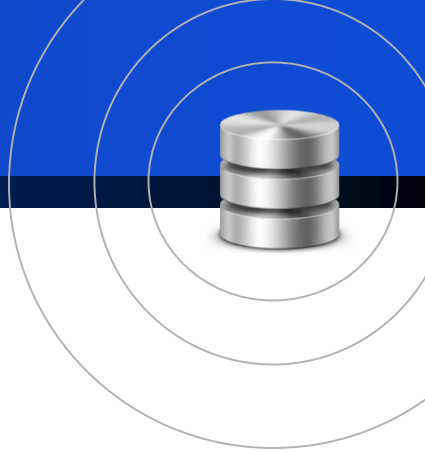
Design Phases

- characterize the data needs
- conceptual-design
  - entity-relationship model
  - specification of functional requirements(功能需求规格说明)
- implementation of the database
  - logical-design phase
  - physical-design phase

Design Alternatives

- must avoid two major pitfalls:
  - **Redundancy**(冗余)
    - A bad design may repeat information.
  - **Incompleteness**(不完整)
    - A bad design may make certain aspects of the enterprise difficult or impossible to model.

SCT（学号，课程号，任课老师，老师的办公地点）

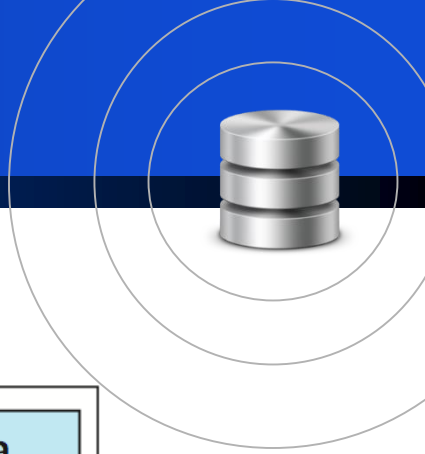


The Entity-Relationship Model

Entity Sets

- **Entity(实体)：** a “thing” or “object” in the real world that is distinguishable from all other objects.
- **entity set(实体集)：** a set of entities of the same type that share the same properties, or attributes.
- An entity is represented by a set of **attributes (属性)**. Attributes are descriptive properties possessed by each member of an entity set.
- Each entity has a **value(值)** for each of its attributes.

Entity Sets



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

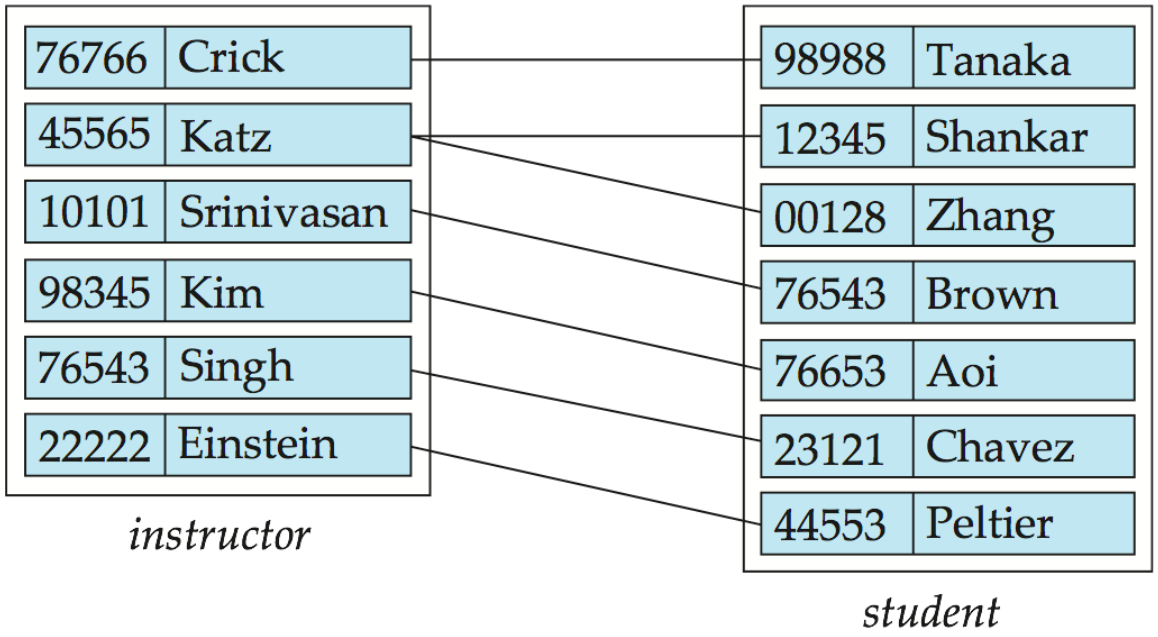
instructor

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

student

Relationship Sets

- **Relationship(联系)** :an association among several entities.
- **relationship set(联系集)** : a set of relationships of the same type.

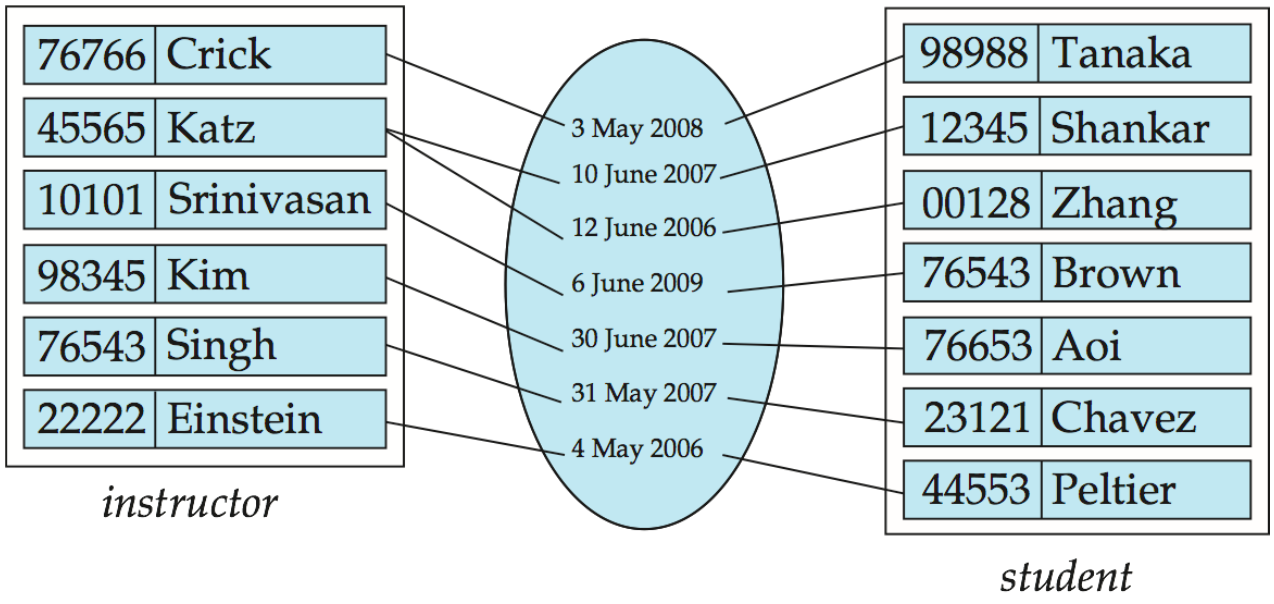


Relationship Sets

- The association between entity sets is referred to as participation; that is, the entity sets $E1, E2, \ldots, En$ **participate(参与)** in relationship set R.
- A **relationship instance(联系实例)** in an E-R schema represents an association between the named entities in the real-world enterprise that is being modeled.
- The function that an entity plays in a relationship is called that entity's **role(角色)**.

Relationship Sets

- A relationship may also have attributes called **descriptive attributes(描述性属性).**



Relationship Sets

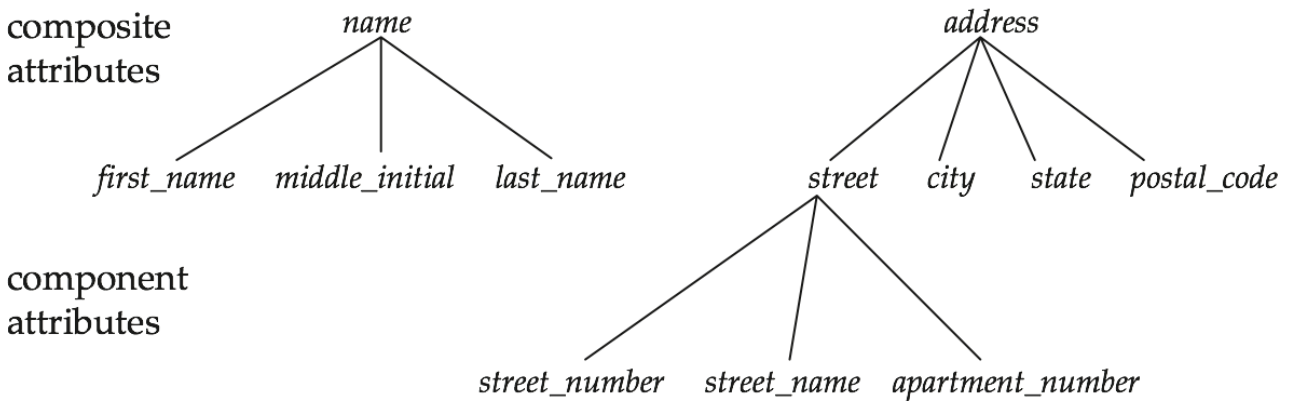
- Most of the relationship sets in a database system are binary(二元的).
- Occasionally, however, relationship sets involve more than two entity sets.
  - *instructor*, *student* and *project*
  - Each project can have multiple associated students and multiple associated instructors.
  - the relationship set *proj_guide*, which indicates that a particular student is guided by a particular instructor on a particular project.

Attributes

- For each attribute, there is a set of permitted values, called the **domain**(域), or **value set**(值集), of that attribute.
- An attribute of an entity set is a function that maps from the entity set into a domain.
- each entity can be described by a set of (attribute, data value) pairs
  - {(*ID*, 76766), (*name*, Crick), (*dept name*, Biology), (*salary*, 72000)}

Attributes

- **Simple** and **composite** attributes
  - Simple attributes(简单属性): have not been divided into subparts
  - composite attributes(复合属性): can be divided into subparts
    - a composite attribute may appear as a hierarchy



Attributes

- **Single-valued** and **multivalued** attributes
  - **Single-valued** attributes(单值属性): have a single value for a particular entity
  - **multivalued** attributes(多值属性): have a set of values for a specific entity
    - An *instructor* may have zero, one, or several phone numbers
    - any particular instructor may have zero, one, or more dependents

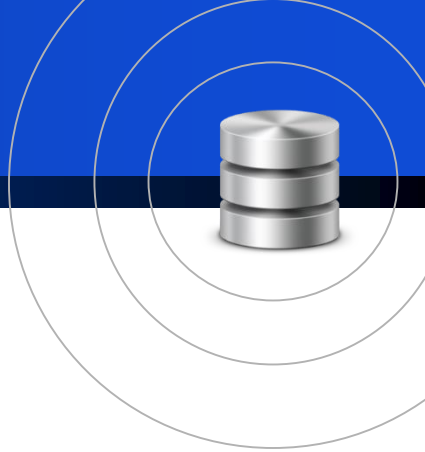
{*phone_number*} or {*dependent_name*}.

    - upper and lower bounds may be placed on the number of values

Attributes

- **Derived** attribute(派生属性)
  - The value can be derived from the values of other related attributes or entities.
    - *instructor* 's attribute: *students_advised*
    - *students*'s attribute: *age*
  - The value of a derived attribute is not stored but is computed when required.

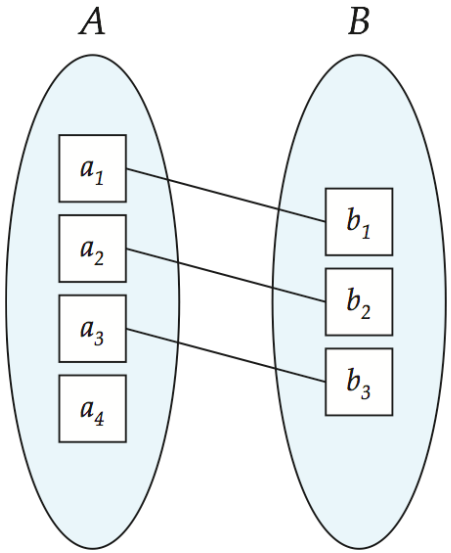
- An attribute takes a **null** value when an entity does not have a value for it.



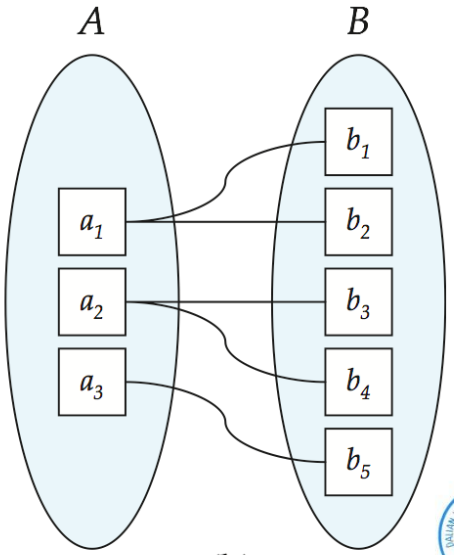
Constraints

Mapping Cardinalities

- **Mapping cardinalities**(映射基数), or **cardinality ratios**(基数比率), express the number of entities to which another entity can be associated via a relationship set.
  - One-to-one
  - One-to-many



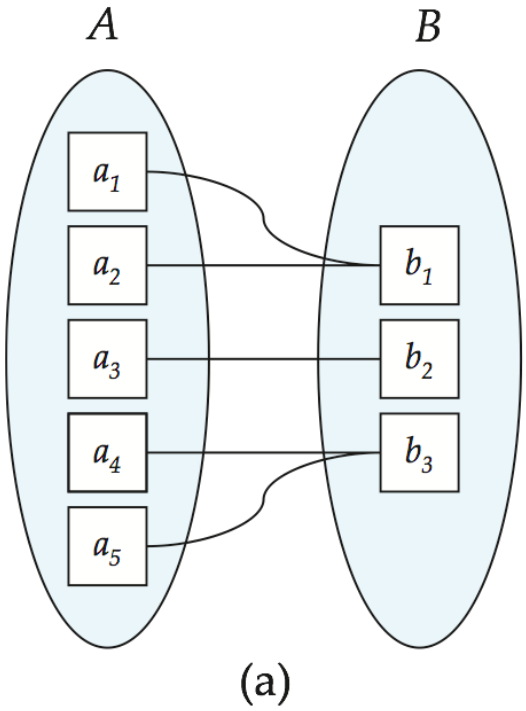
(a)



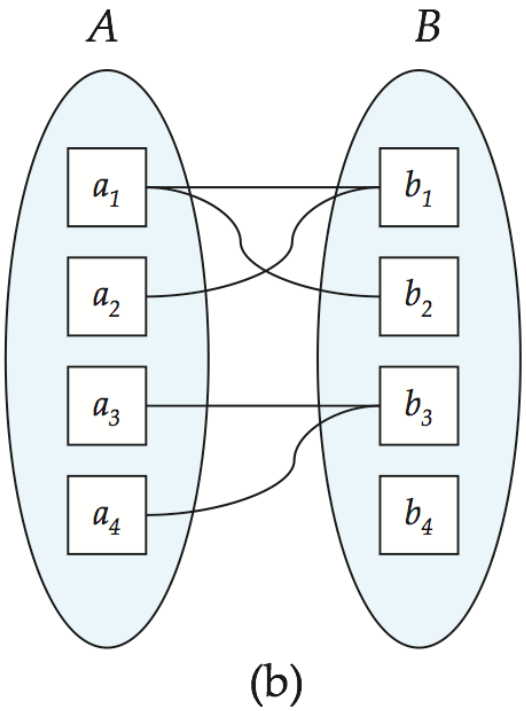
(b)

Mapping Cardinalities

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



(a)

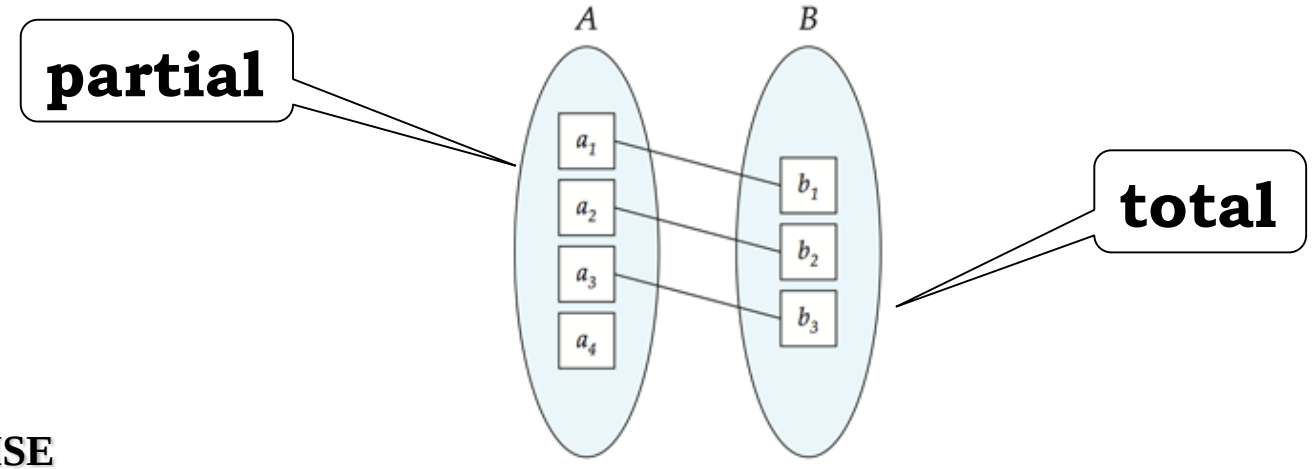


(b)

Participation Constraints

- The participation of an entity set *E* in a relationship set *R* is said to be **total(全部的)** if every entity in *E* participates in at least one relationship in *R*.

- If only some entities in *E* participate in relationships in *R*, the participation of entity set *E* in relationship *R* is said to be **partial(部分的)**.



Keys

- A key for an entity is a set of attributes that suffice to distinguish entities from each other.
- Keys also help to identify relationships uniquely, and thus distinguish relationships from each other.

Keys

- If the relationship set R has no attributes associated with it, then the set of attributes

  $primary\text{-}key(E1) \cup primary\text{-}key(E2) \cup \cdots \cup primary\text{-}key(En)$

  describes an individual relationship in set R.

- If the relationship set R has attributes $a1, a2, \ldots, am$ associated with it, then the set of attributes

  $primary\text{-}key(E1) \cup primary\text{-}key(E2) \cup \cdots \cup primary\text{-}key(En) \cup \{a1, a2, \ldots, am\}$

  describes an individual relationship in set R.

Removing Redundant Attributes in Entity Sets

Removing Redundant Attributes in Entity Sets

- When designing a database using the E-R model
  - Usually start by identifying entity sets
  - Then choose the appropriate attributes.
  - Then the relationship sets among the various entities are formed.
  - These relationship sets may result in a situation where attributes in the various entity sets are redundant(冗余的) and need to be removed from the original entity sets.

Removing Redundant Attributes in Entity Sets

- *Example*
  - *Instructor*(*ID*, *name*, *dept_name*, *salary*)
  - *department*(*dept_name*, *building*, *budget*)
  - A relationship set *inst_dept* relating *instructor* and *department*.
  - The attribute *dept_name* appears in both entity sets. Since it is the primary key for the entity set *department*, it is redundant in the entity set *instructor* and needs to be removed.

WHY???

Removing Redundant Attributes in Entity Sets

- *Another example*
  - *Section*(*course_id*, *sec_id*, *semester*, *year*, *building*, *room_number*, *time_slot_id*)
  - *Time_slot*(*time_slot_id*,{(*day*, *start_time*, *end_time*)})
    - {(*day*, *start_time*, *end_time*)}, multivalued composite attribute
  - The attribute *time_slot_id* appears in both entity sets. Since it is the primary key for the entity set *time_slot*, it is redundant in the entity set *section* and needs to be removed.

Removing Redundant Attributes in Entity Sets

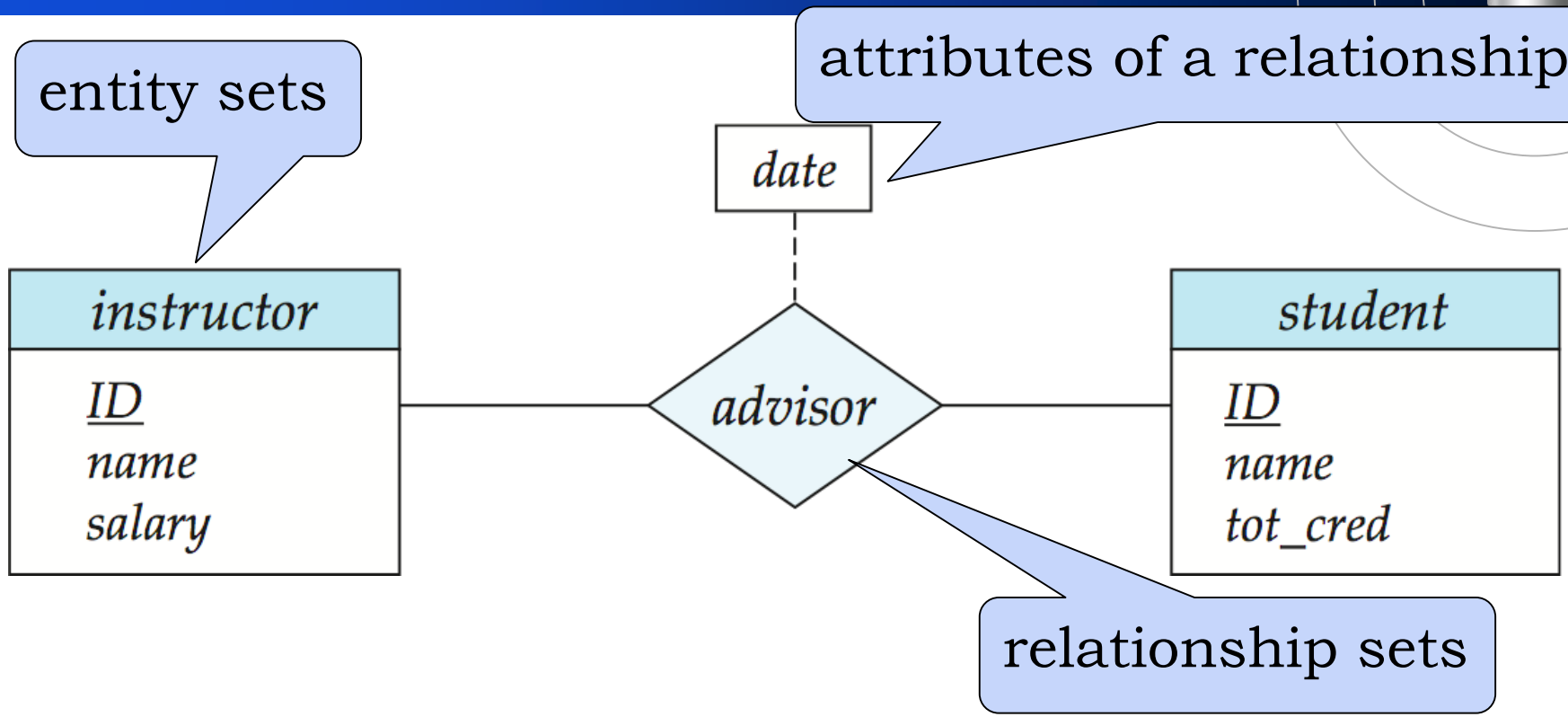
- entity sets in the university example
  - **classroom**(*building*, *room_number*, *capacity*).
  - **department**(*dept_name*, *building*, *budget*).
  - **course**(*course_id*, *title*, *credits*).
  - **instructor** (*ID*, *name*, *salary*).
  - **section** (*course_id*, *sec_id*, *semester*, *year*).
  - **student** (*ID*, *name*, *tot_cred*).
  - **time_slot** (*time_slot_id*, {(*day*, *start time*, *end time*) }).

Removing Redundant Attributes in Entity Sets

- relationship sets in the university example
  - **inst_dept**: relating instructors with departments.
  - **stud_dept**: relating students with departments.
  - **teaches**: relating instructors with sections.
  - **takes**: relating students with sections, with a descriptive attribute *grade*.
  - **course_dept**: relating courses with departments.
  - **sec_course**: relating sections with courses.
  - **sec_class**: relating sections with classrooms.
  - **sec_time_slot**: relating sections with time slots.
  - **advisor**: relating students with instructors.
  - **prereq**: relating courses with prerequisite courses.

Entity-Relationship Diagrams

Basic Structure



- **Double lines** indicate total participation of an entity in a relationship set.
- **Double diamonds** represent identifying relationship sets linked to weak entity sets

Mapping Cardinality

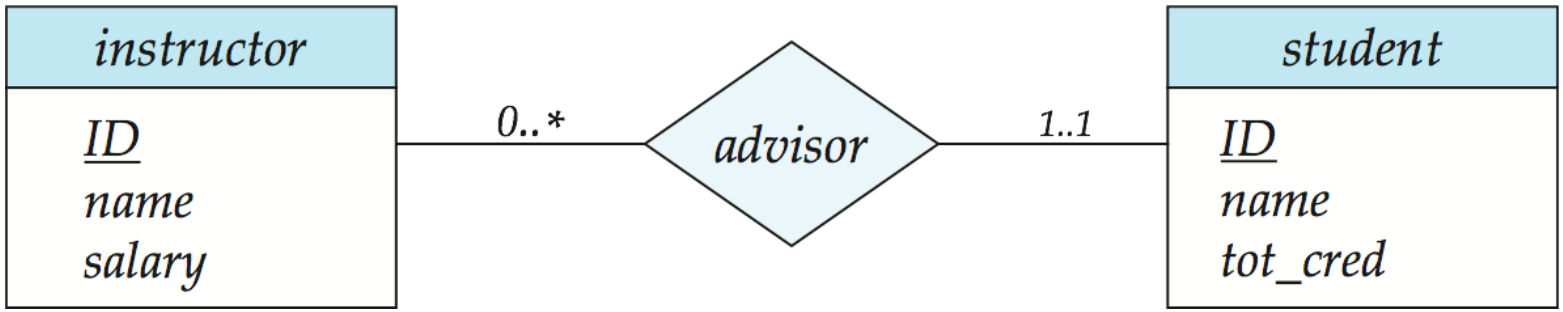
- **One-to-one**

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

- **Many-to-many**

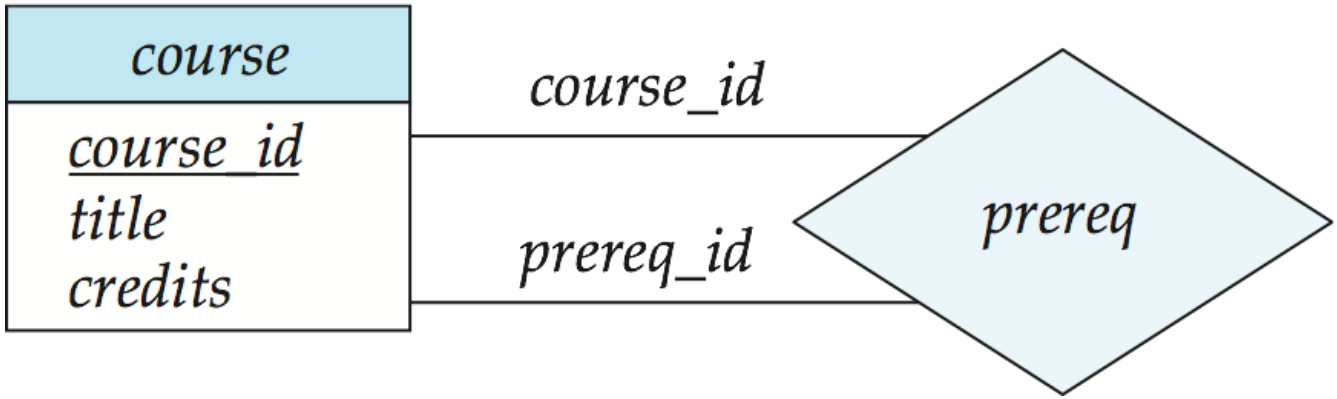
Mapping Cardinality

- A line may have an associated minimum and maximum cardinality, shown in the form *l..h*, where *l* is the minimum and *h* the maximum cardinality.



Roles

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



Weak Entity Sets

- **section** (*course_id, sec_id, semester, year*).
- **course**(*course_id*, *title*, *credits*).

- **sec_course**: relating sections with courses.

The information in *sec_course* is redundant!!!

Weak Entity Sets

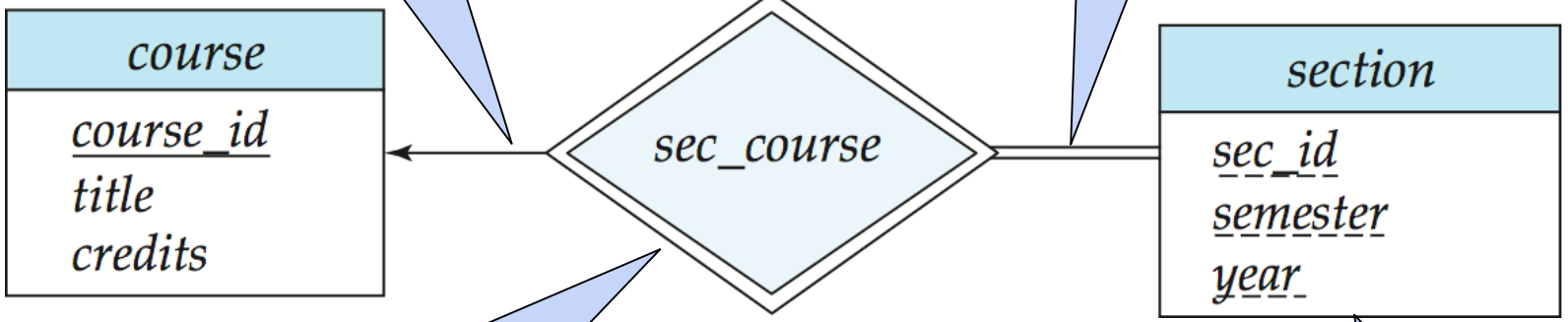
- **SOLUTION**
  - Getting rid of the relationship *sec_course*
  - Do not store the attribute *course_id* in the *section* entity

- **section** (*sec_id, semester, year*).
  - Another problem: entity *section* does not have enough attributes to identify a particular *section* entity uniquely

Weak Entity Sets

each section is related to a single course

total participation (全部参与).



The relationship set connecting the weak entity set to the identifying strong entity set is depicted by a double diamond(双菱形).

Discriminator is underlined with a dashed line(虚线).

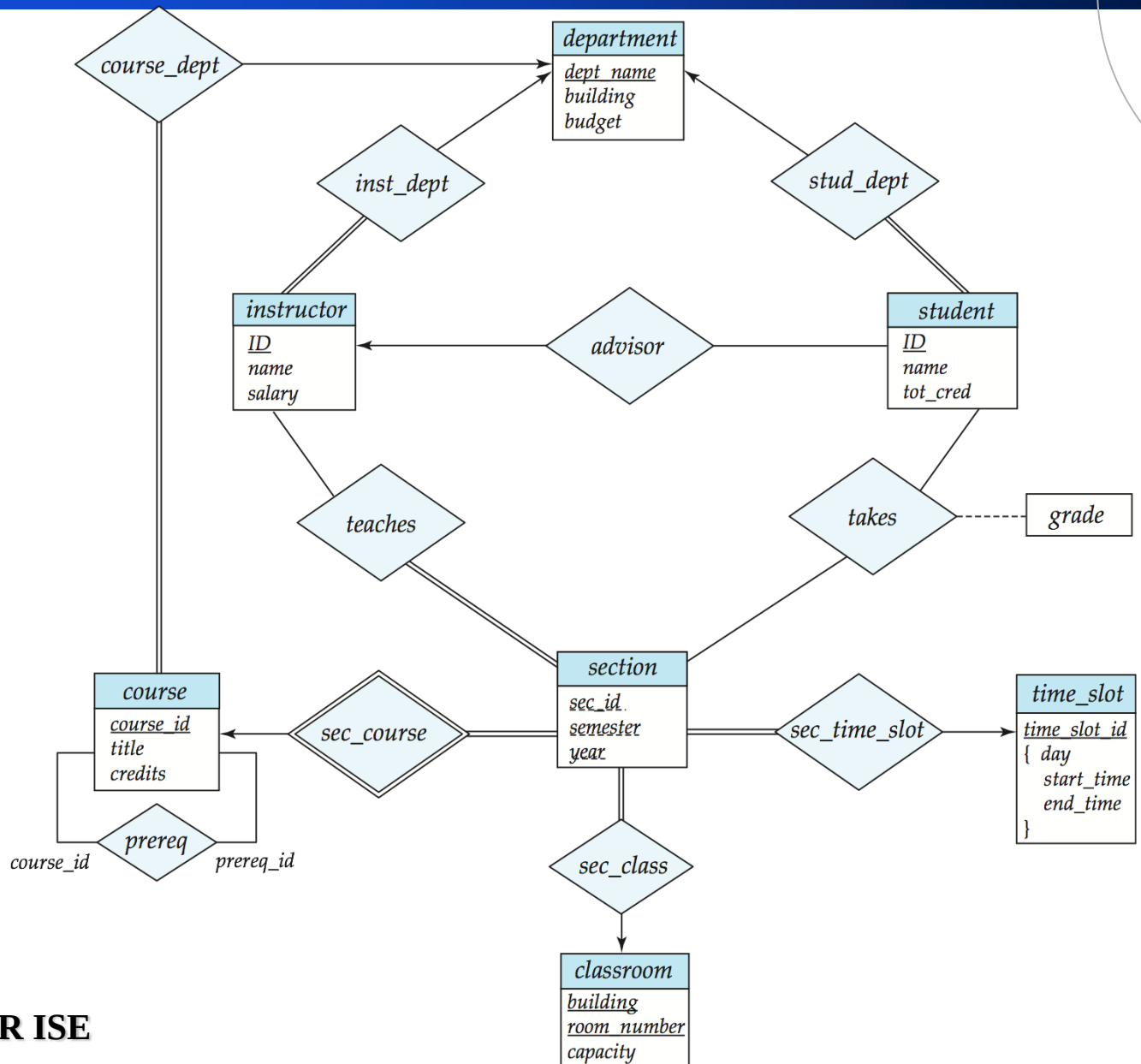
Weak Entity Sets

- **weak entity set**(弱实体集): An entity set that does not have sufficient attributes to form a primary key.
- **strong entity set**(强实体集) : An entity set that has a primary key.

- For a weak entity set to be meaningful, it must be associated with another entity set, called the **identifying**(标识) or **owner entity set**(属主实体集).

- **discriminator**(分辨符): a set of attributes that distinguishes among all those entities in the weak entity set.

E-R diagram for the University Enterprise



Reduction to Relational Schemas

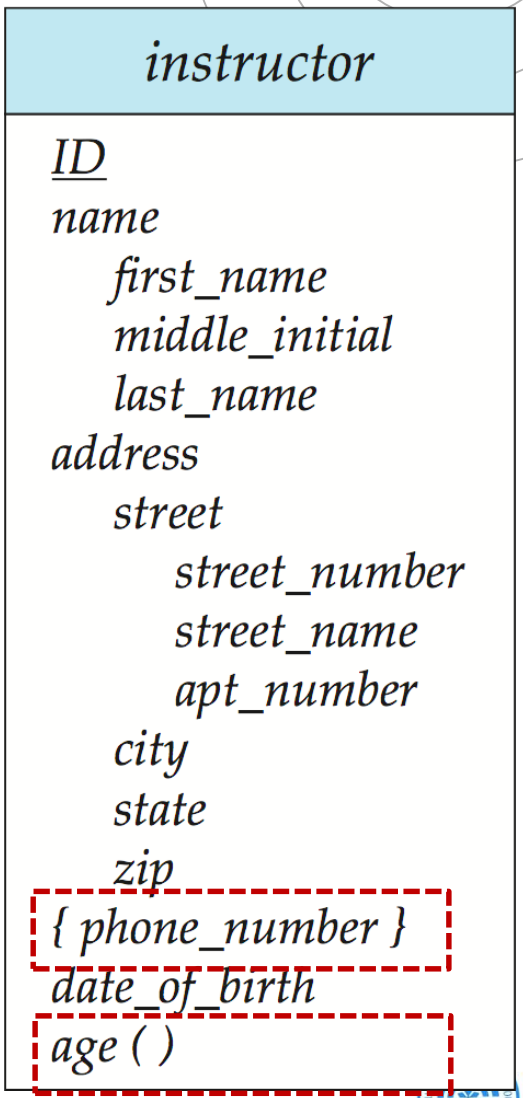
Representation of Strong Entity Sets with Simple Attributes

- Let E be a strong entity set with only simple descriptive attributes $a1, a2, \ldots, an$. We represent this entity by a schema called E with n distinct attributes.
- The primary key of the entity set serves as the primary key of the resulting schema.
  - *student* (*ID*, *name*, *tot cred*)
  - *classroom* (*building*, *room number*, *capacity*)
  - *department* (*dept_name*, *building*, *budget*)
  - *course* (*course_id*, *title*, *credits*)
  - *instructor* (*ID*, *name*, *salary*)

Representation of Strong Entity Sets with Complex Attributes

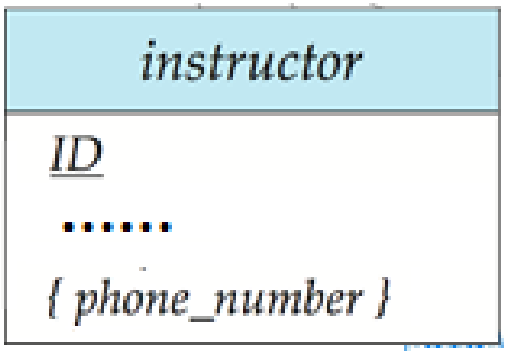
- composite attributes(复合属性)
  - creating a separate attribute for each of the component attributes;
  - do not create a separate attribute for the composite attribute itself.

instructor (*ID*, *first_name*, *middle_name*, *last_name*, *street number*, *street_name*, *apt_number*, *city*, *state*, *zip code*, *date_of_birth*)



Representation of Strong Entity Sets with Complex Attributes

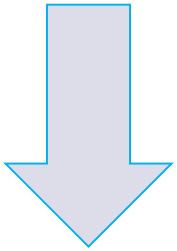
- multivalued attribute(多值属性) M
  - create a relations chema R with an attribute A that corresponds to M , and attributes corresponding to the primary key of the entity set or relationship set.



instructor_phone (*ID*, *phone_number*)

Representation of Strong Entity Sets with Complex Attributes

- **time_slot** (*time_slot_id*, {(*day*, *start time*, *end time*) }).



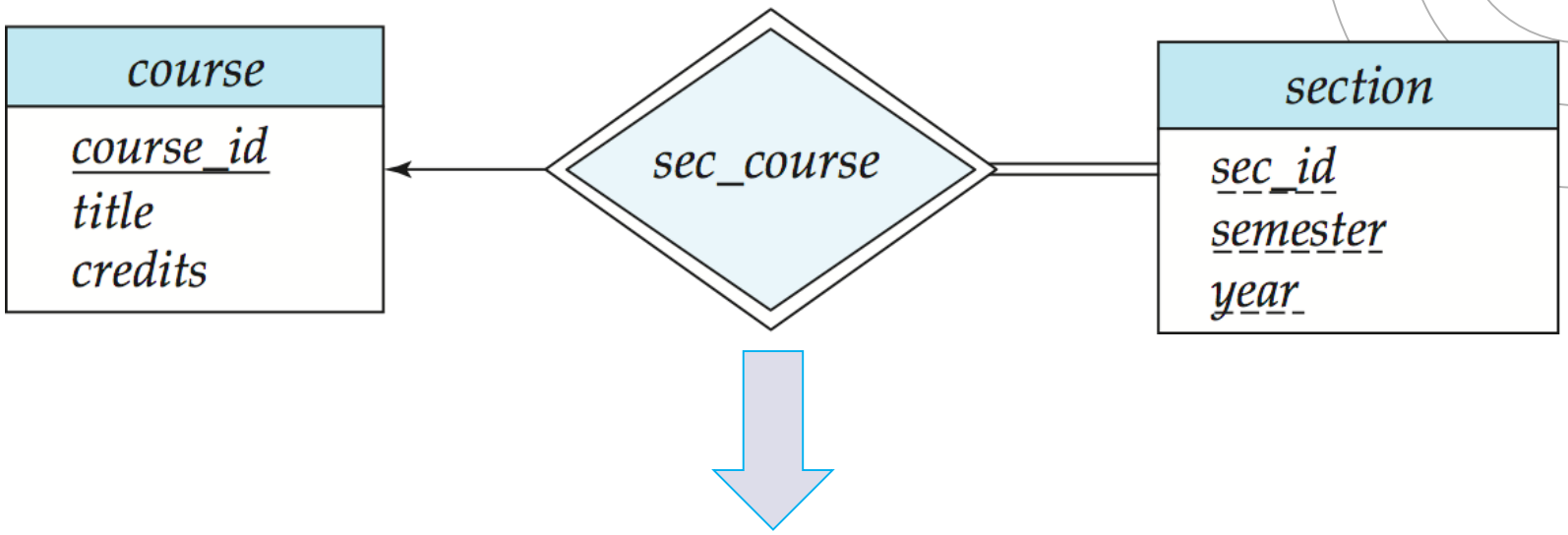
time_slot (*time_slot_id*, *day*, *start_time*, *end_time*)

Representation of Weak Entity Sets

- Let A be a weak entity set with attributes $a1, a2, \ldots, am$.
- Let B be the strong entity set on which A depends.
- Let the primary key of B consist of attributes $b1, b2, \ldots, bn$.
- We represent the entity set A by a relation schema called A with one attribute for each member of the set:

$$\{a1, a2, \ldots, am\} \cup \{b1, b2, \ldots, bn\}$$

Representation of Weak Entity Sets



section(course_id, sec_id, semester, year**)**

Can you conclude the primary key and foreign key?

Representation of Relationship Sets

- Let R be a relationship set, let $a1, a2, \ldots, am$ be the set of attributes formed by the union of the primary keys of each of the entity sets participating in R.
- Let the descriptive attributes (if any) of R be $b1, b2, \ldots, bn$.
- We represent this relationship set by a relation schema called R with one attribute for each member of the set:

$$\{a1, a2, \ldots, am\} \cup \{b1, b2, \ldots, bn\}$$

Representation of Relationship Sets

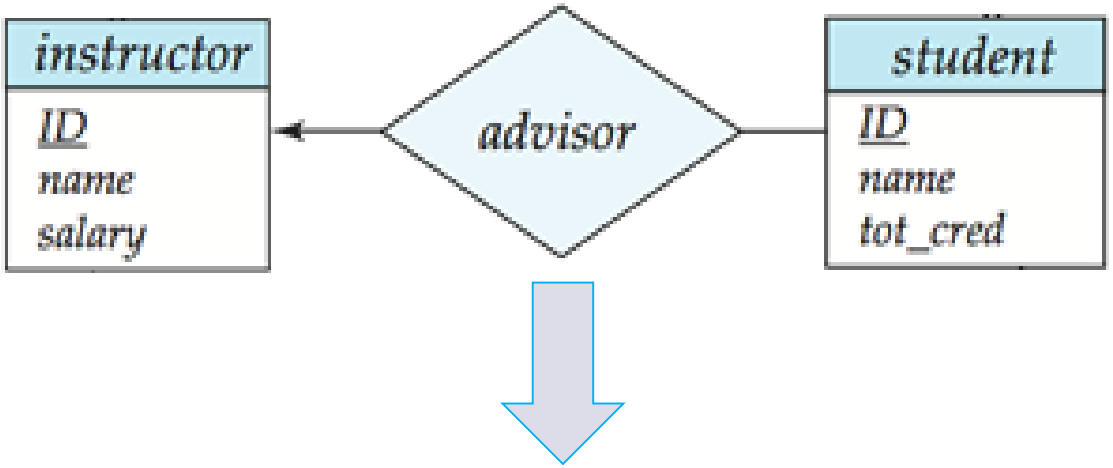
- The primary key is chosen as follows:
  - binary many-to-many relationship:
    - the union of the primary-key attributes
  - binary one-to-one relationship :
    - the primary key of either entity set
  - binary many-to-one or one-to-many relationship :
    - the primary key of the entity set on the “many” side

Representation of Relationship Sets

- The primary key is chosen as follows:
  - For an *n*-ary relationship set without any arrows on its edges
    - the union of the primary key-attributes from the participating entity sets becomes the primary key.
  - For an *n*-ary relationship set with an arrow on one of its edges
    - the primary keys of the entity sets not on the “arrow” side of the relationship set serve as the primary key for the schema.

Representation of Relationship Sets

- For example

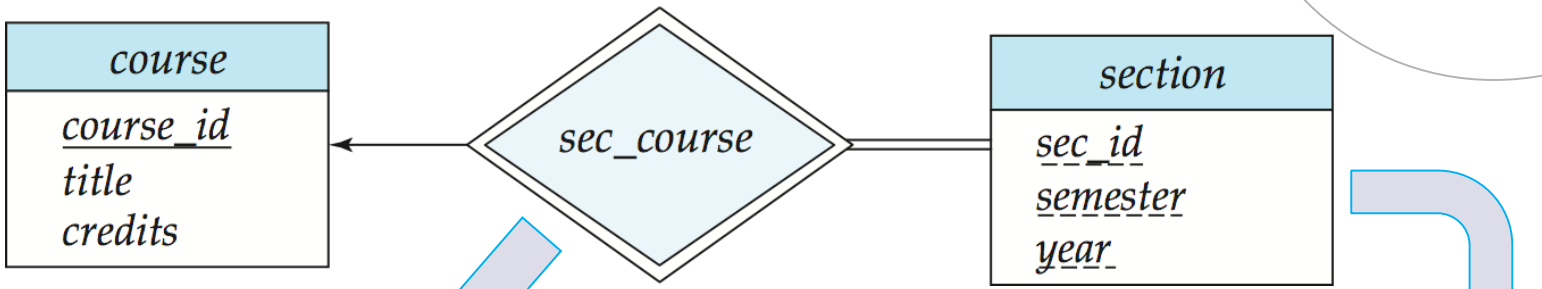


advisor(s_id, i_id**)**

Any foreign keys?

Representation of Relationship Sets

- Redundancy(冗余) of Schemas



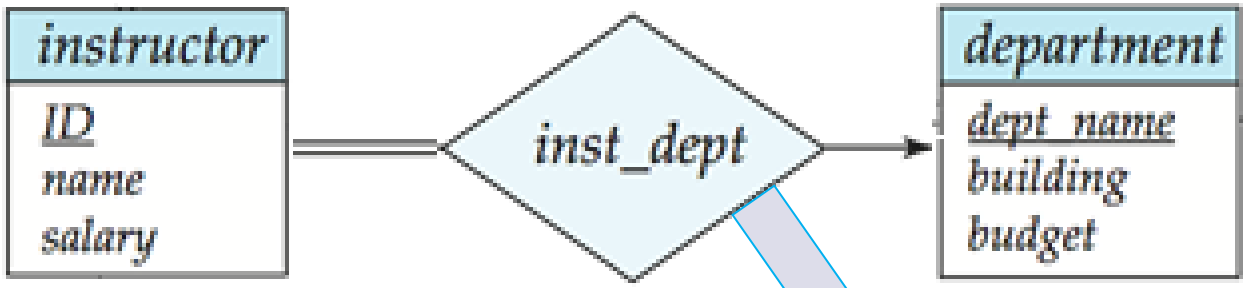
section(course_id, sec_id, semester, year**)**

sec_course(course_id, sec_id, semester, year**)**

In general, the schema for **the relationship set** linking a weak entity set to its corresponding strong entity set **is redundant.**

Representation of Relationship Sets

- Combination(合并) of Schemas



instructor(ID, name, salary)

inst_dept (ID, dept_name)

instructor(ID, name, salary, dept_name)

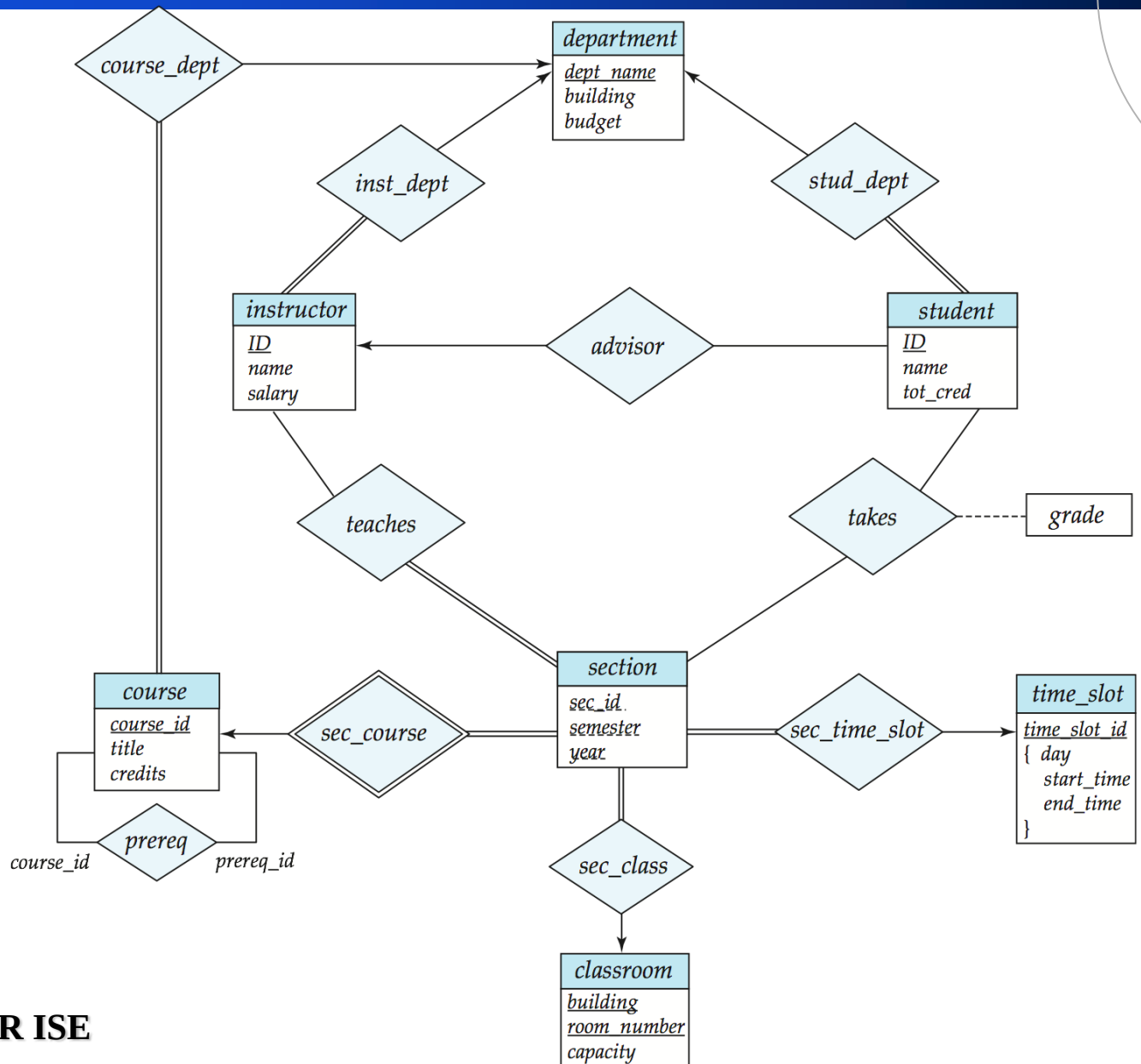
Representation of Relationship Sets

- Combination(合并) of Schemas
  - Consider a many-to-one relationship set *AB* from entity set *A* to entity set *B*.
  - we get three schemas: *A*, *B*, and *AB*.
  - Suppose further that the participation of *A* in the relationship is total; that is, every entity *a* in the entity set *A* must participate in the relationship *AB*.
  - Then we can combine the schemas *A* and *AB* to form a single schema consisting of the union of attributes of both schemas.

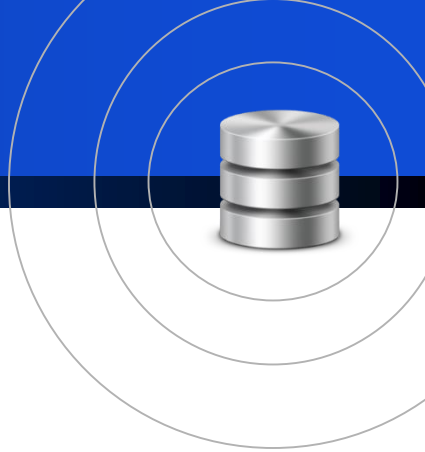
Representation of Relationship Sets

- Combination(合并) of Schemas
  - In the case of one-to-one relationships, the relation schema for the relationship set can be combined with the schemas for either of the entity sets.
  - We can combine schemas even if the participation is partial by using null values.
  - Drop the constraint referencing the entity set into whose schema the relationship set schema is merged, and add the other foreign-key constraints to the combined schema.

E-R diagram for the University Enterprise



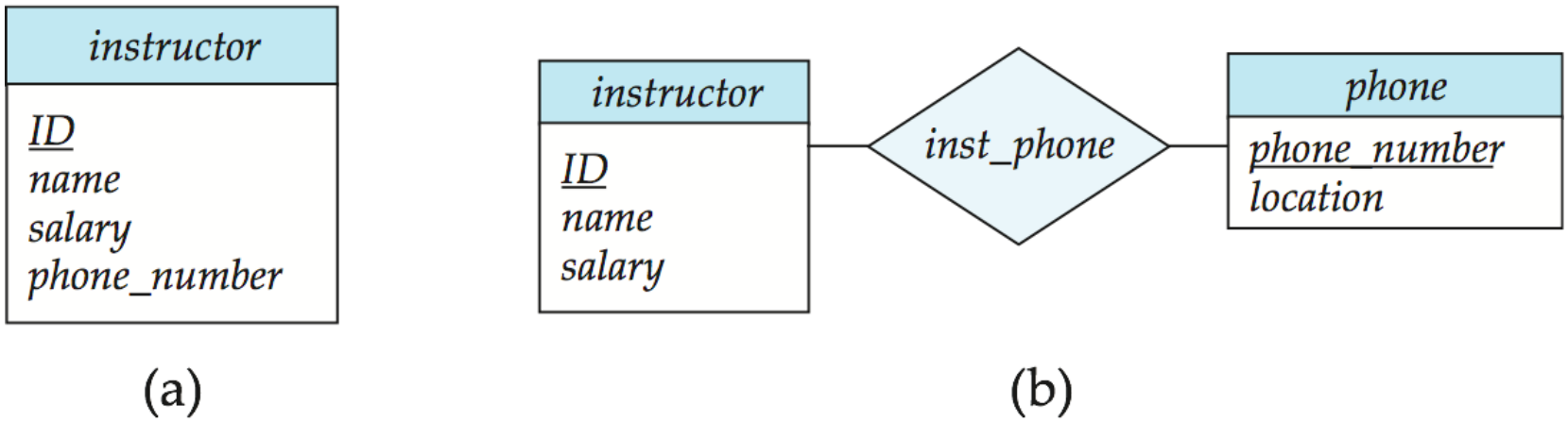
classroom(*building*, *room_number*, *capacity*)
department(*dept_name*, *building*, *budget*)
course(*course_id*, *title*, *dept_name*, *credits*)
instructor(*ID*, *name*, *dept_name*, *salary*)
section(*course_id*, *sec_id*, *semester*, *year*, *building*, *room_number*, *time_slot_id*)
teaches(*ID*, *course_id*, *sec_id*, *semester*, *year*)
student(*ID*, *name*, *dept_name*, *tot_cred*)
takes(*ID*, *course_id*, *sec_id*, *semester*, *year*, *grade*)
advisor(*s_ID*, *i_ID*)
time_slot(*time_slot_id*, *day*, *start_time*, *end_time*)
prereq(*course_id*, *prereq_id*)



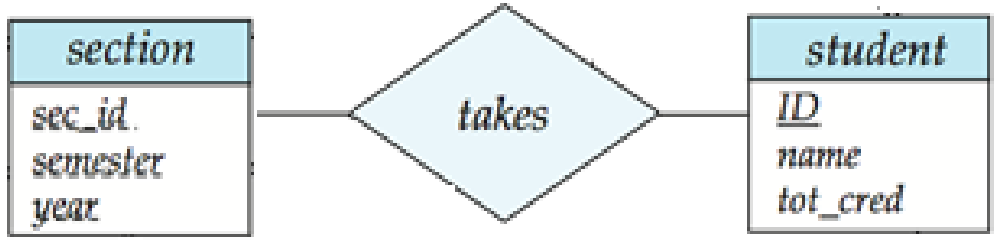
Entity-Relationship Design Issues

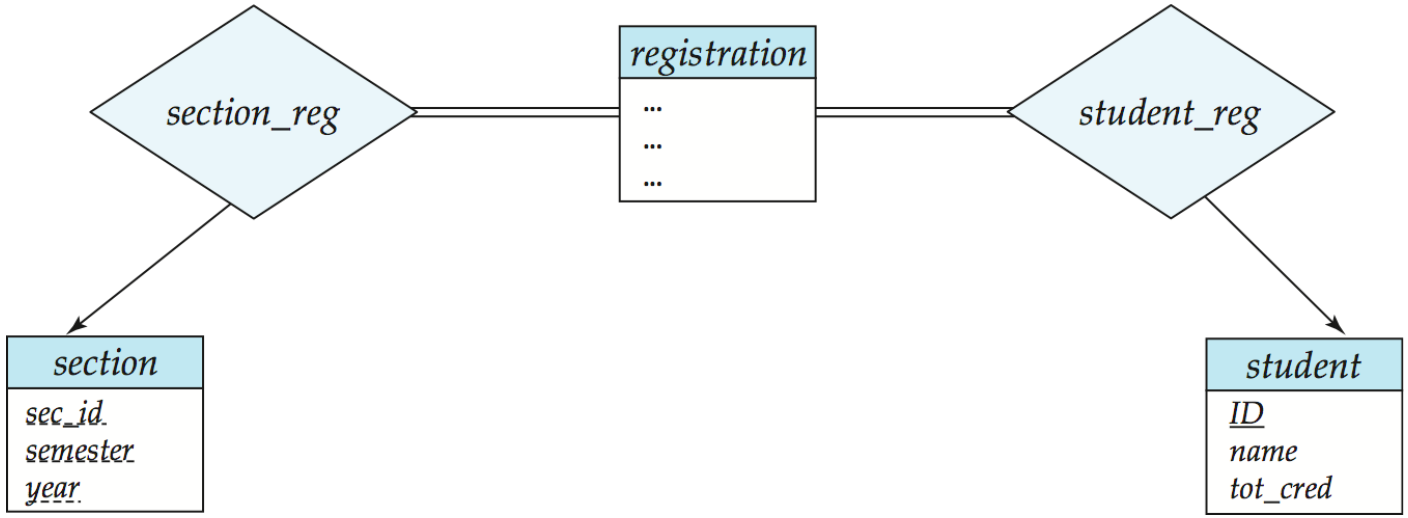
Use of Entity Sets versus Attributes

- What constitutes an attribute, and what constitutes an entity set?
  - depend on the structure of the real-world enterprise being modeled, and on the semantics (语义) associated with the attribute in question.

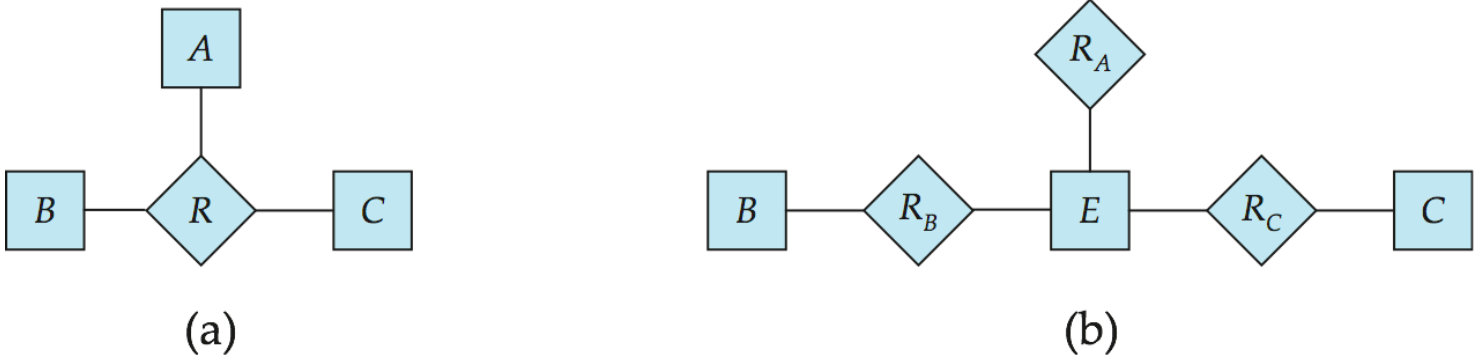


Use of Entity Sets versus Relationship Sets





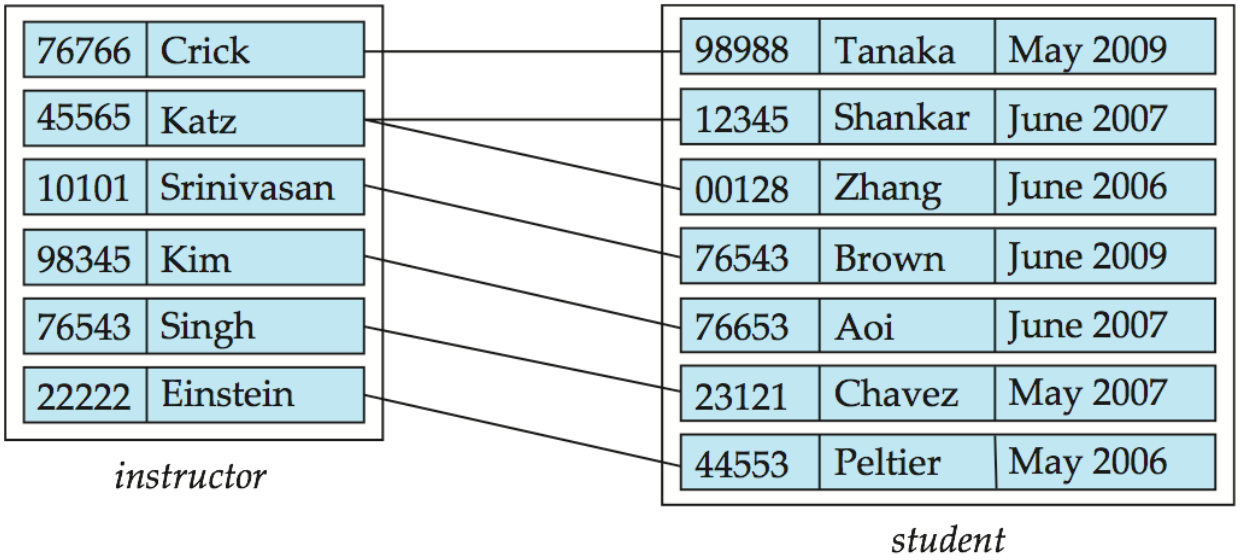
Binary versus *n*-ary Relationship Sets



replace a nonbinary (*n*-ary, for $n > 2$) relationship set by a number of distinct binary relationship sets.

Placement of Relationship Attributes

- Attributes of a one-to-many relationship set can be repositioned to only the entity set on the “many” side of the relationship.
- For one-to-one relationship sets, the relationship attribute can be associated with either one of the participating entities.



Thanks