Chapter 7: Entity-Relationship Model

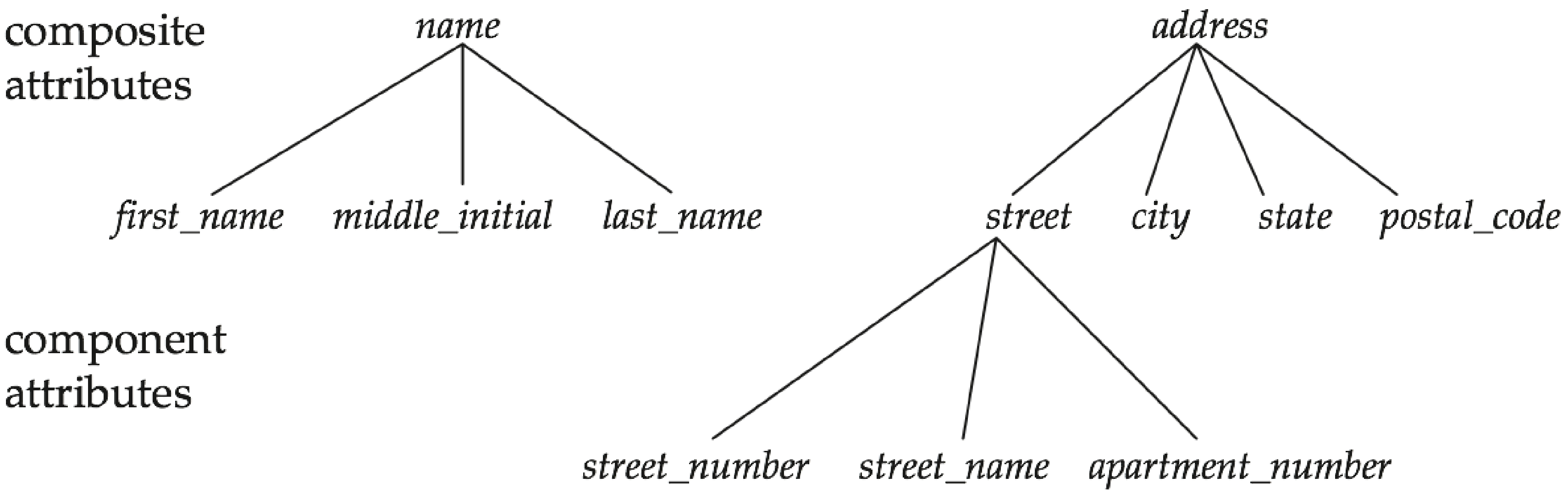
Edited by Radhika Sukapuram

Database System Concepts, 7th Ed.

Complex Attributes

- Attribute types:
  - **Simple** and **composite** attributes.
    - Composite attributes
      - can be divided into sub-parts
      - Help group related attributes
      - A good choice only when the user wishes to refer to the entire attribute on some occasions and to only a component of the attribute on other occasions
      - Example: address

Composite Attributes



Complex Attributes cont.

- **Single-valued** and **multivalued** attributes
  - Example of multivalued attributes:
    - {*phone_numbers*} 0 or more phone numbers
    - {*dependents*} : 0 or more dependents
  - Upper and lower bounds may be placed on the number of attributes

Complex Attributes cont.

  - **Derived** attributes
    - Can be computed from other attributes or entities
    - Example:
    - *age*, given *date_of_birth* (base or stored attribute)
    - Attribute *total_salary* of entity *instructor (*derived from *instructor*)
- **Domain** – the set of permitted values for each attribute
- Attributes can have null values
  - Not applicable (apartment number is not part of the address), unknown (not known if apartment number is part of the address), missing (name of instructor)

Redundant Attributes

- Suppose we have entity sets:
  - *instructor*, with attributes: *ID*, *name*, *dept_name, salary*
  - *department,* with attributes: *dept_name, building, budget*
- We model the fact that each instructor has an associated department
  - using a relationship set *inst_dept*
- Treating *inst_dept* as a relationship rather than as an attribute of *instructor* helps in
  - Making the logical relationship explicit
  - Removes the assumption that each instructor is associated with only one department

Redundant Attributes cont.

- The attribute *dept_name* appears in both entity sets.
  - Since it is the primary key for the entity set *department*, it replicates information present in the relationship
  - therefore redundant in the entity set *instructor*
  - therefore needs to be removed.
- BUT: when converting back to tables, in some cases the attribute gets reintroduced (covered later).
- A good entity-relationship design does not contain redundant attributes

Redundant attributes : example

classroom (building, room_number, capacity)

section (course_id, section_id, semester, year, building, room_number, timeslot_id)

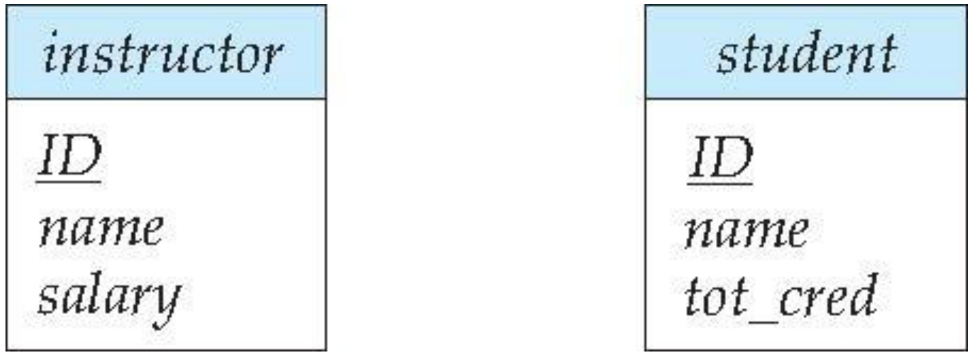
Relationship set *sec_class* relates *section* to *classroom*

{building, room_number} are redundant in *section* and must be removed

E-R Diagrams

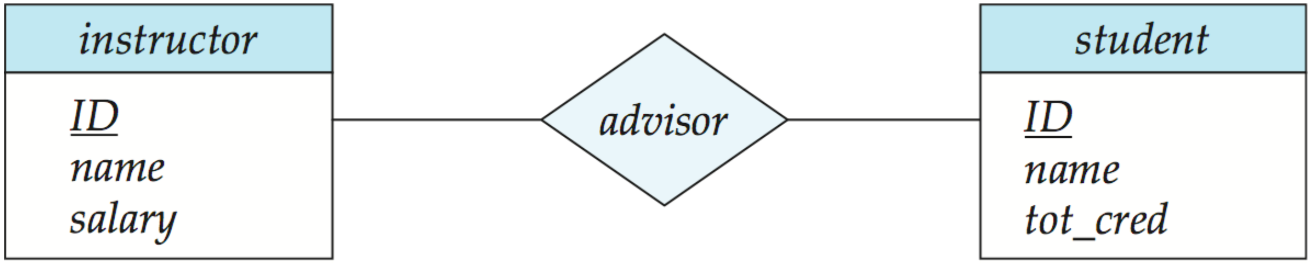
Entity Sets

- Entity sets can be represented graphically as follows:
  - Rectangles represent entity sets.
  - Attributes listed inside entity rectangle
  - Underline indicates primary key attributes

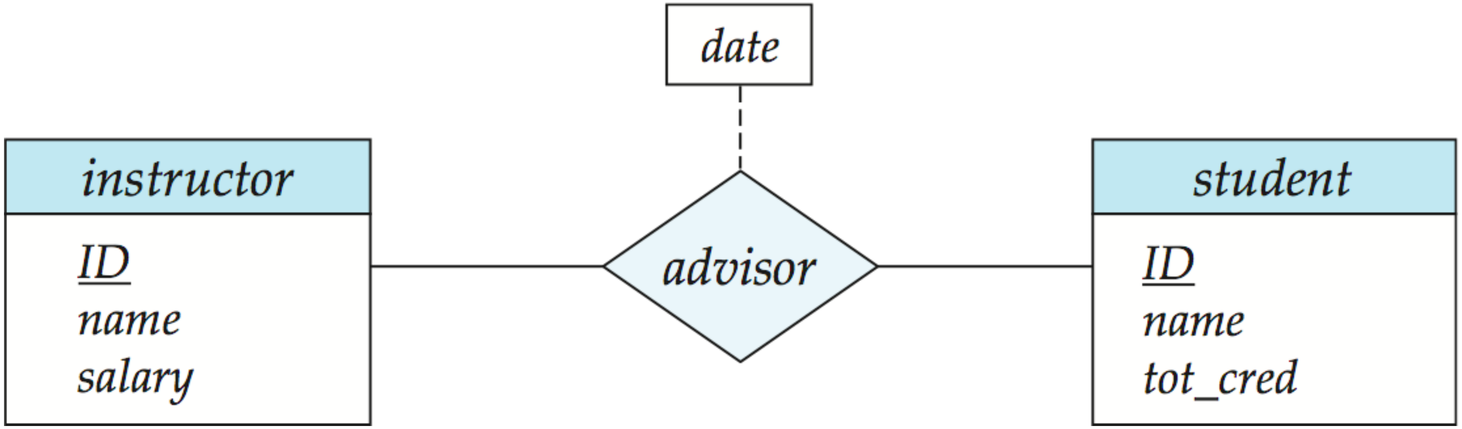


Relationship Sets

- Diamonds represent relationship sets.

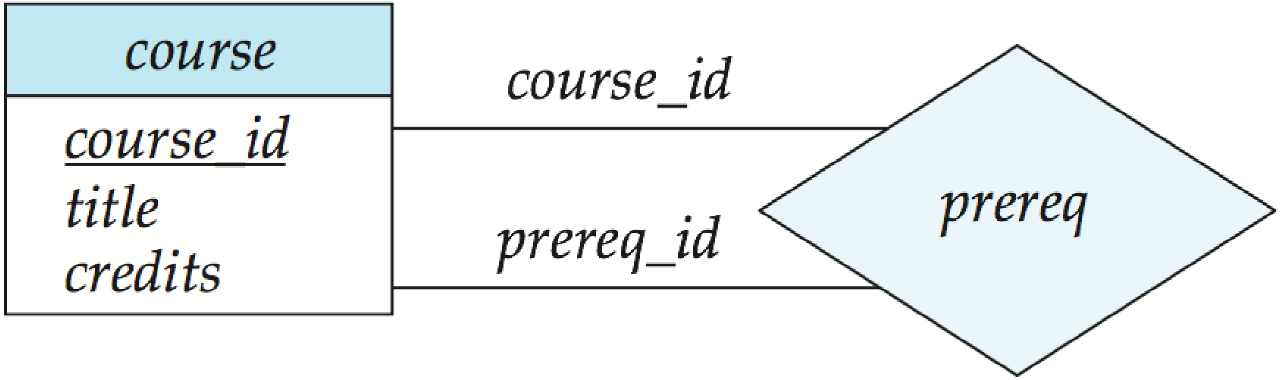


Relationship Sets with Attributes



Roles

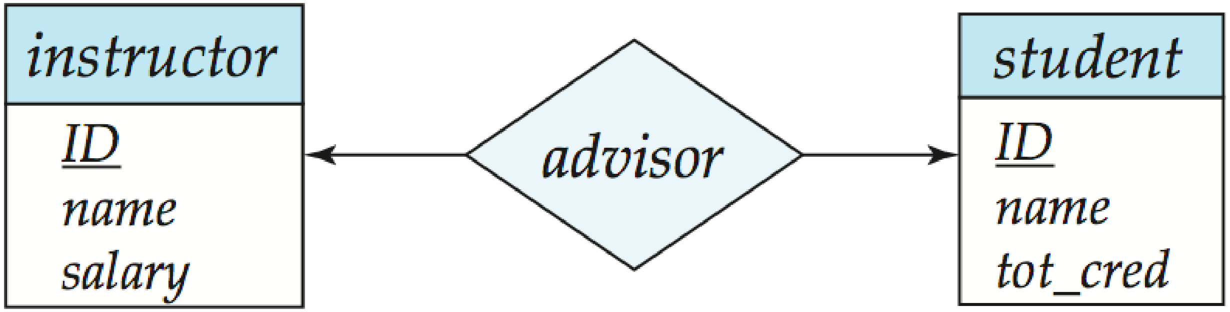
- Entity sets of a relationship need not be distinct
  - If distinct, the function an entity set plays is implicit
  - Otherwise each occurrence of an entity set plays a "role" in the relationship
  - The same entity set participates in a relationship set more than once, in different roles
- The labels "*course_id*" and "*prereq_id*" are **roles**



Cardinality Constraints

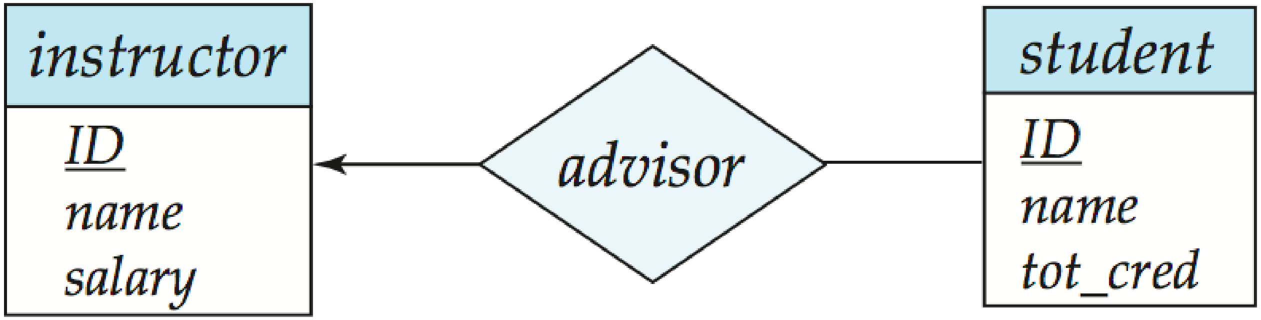
- We express cardinality constraints by drawing either a directed line (→), signifying "one," or an undirected line (—), signifying "many," between the relationship set and the entity set.

- One-to-one relationship between an *instructor* and a *student* :
  - A *student* is associated with at most one *instructor* via the relationship set *advisor*
  - An *instructor* is associated with at most one *student* via *advisor*



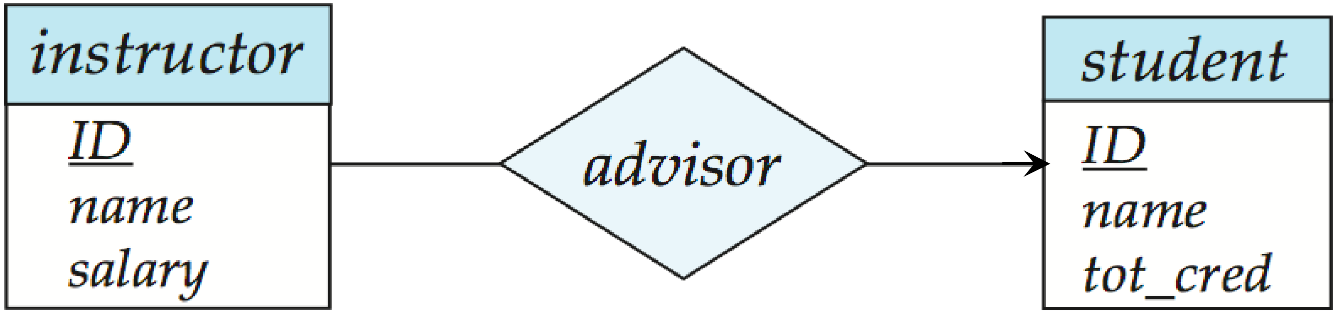
One-to-Many Relationship

- one-to-many relationship between an *instructor* and a *student*
  - an instructor is associated with zero or more students via *advisor*
  - a student is associated with at most one instructor via advisor,



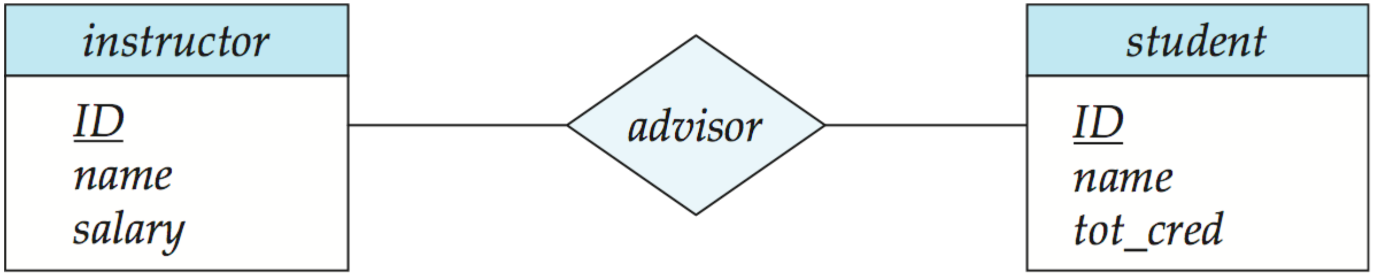
Many-to-One Relationships

- 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 zero or more instructors via *advisor*



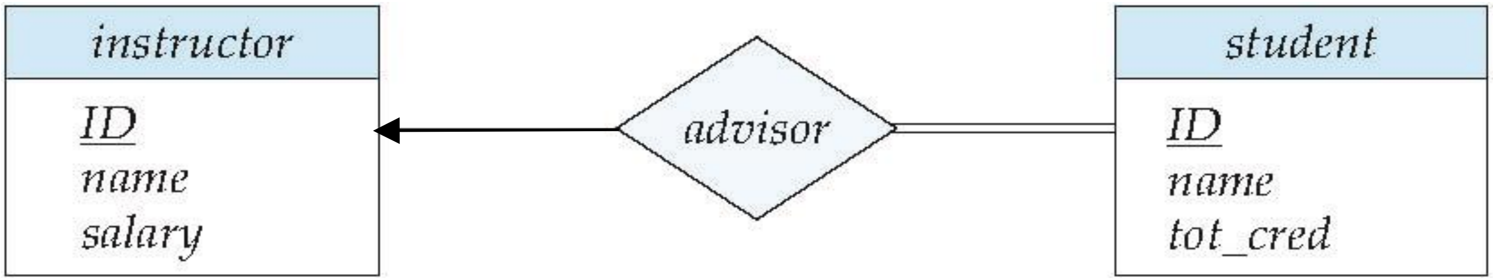
Many-to-Many Relationship

- An instructor is associated with zero or more students via *advisor*
- A student is associated with zero or more instructors via *advisor*



Total and Partial Participation

- Total participation (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set



participation of *student* in *advisor r*elation is total

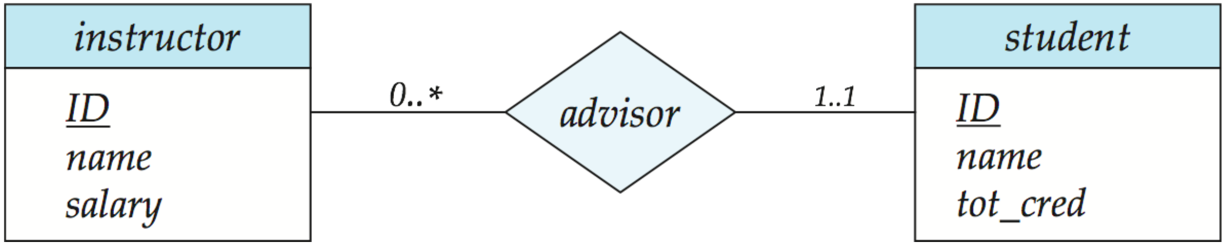
  - every *student* must have an associated instructor

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

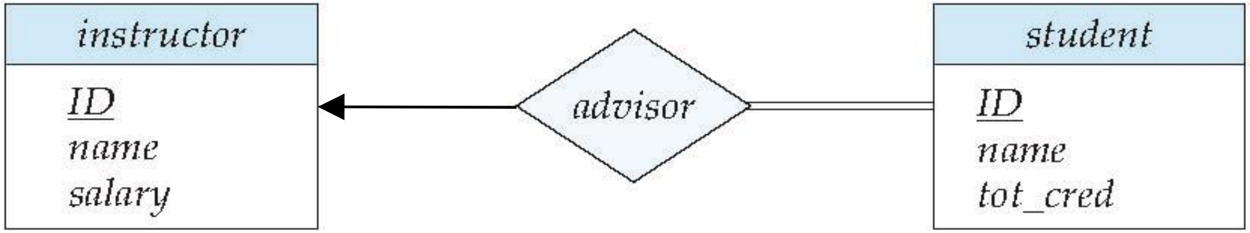
Notation for Expressing More Complex Constraints

- 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
    - A minimum value of 1 indicates total participation (alternative notation)
    - A maximum value of 1 indicates that the entity participates in at most one relationship
    - A maximum value of * indicates no limit.

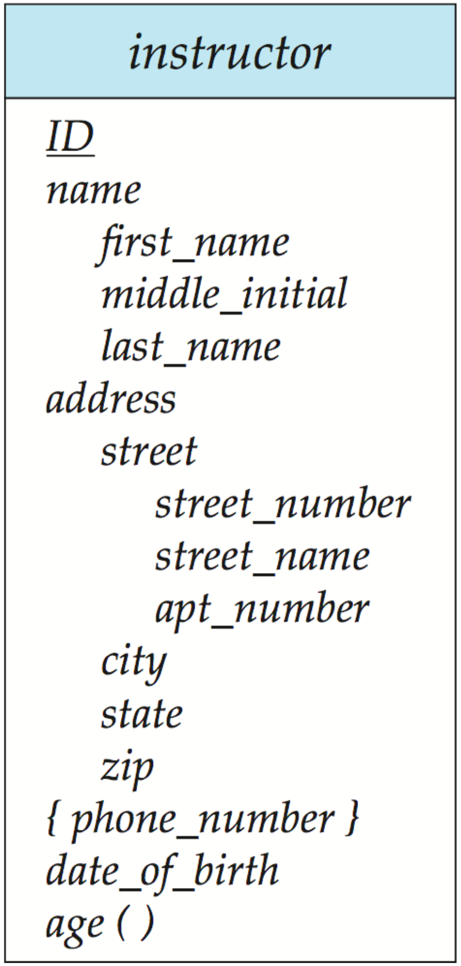
Notation for Expressing More Complex Constraints cont.



- Instructor can advise a minimum of 0 or a maximum of * students.
- A student must have a minimum of 1 advisor and a maximum of 1 advisor
- Same as



Notation to Express Entity with Complex Attributes



instructor

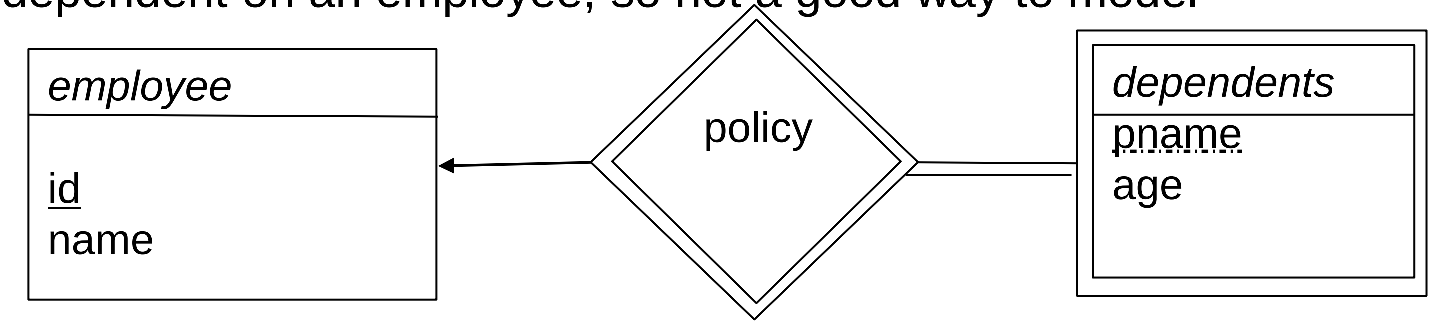
- *ID*
- *name*
  - *first_name*
  - *middle_initial*
  - *last_name*
- *address*
  - *street*
    - *street_number*
    - *street_name*
    - *apt_number*
  - *city*
  - *state*
  - *zip*
- *{ phone_number }*
- *date_of_birth*
- *age ( )*

Weak entity sets

employee (id, name)

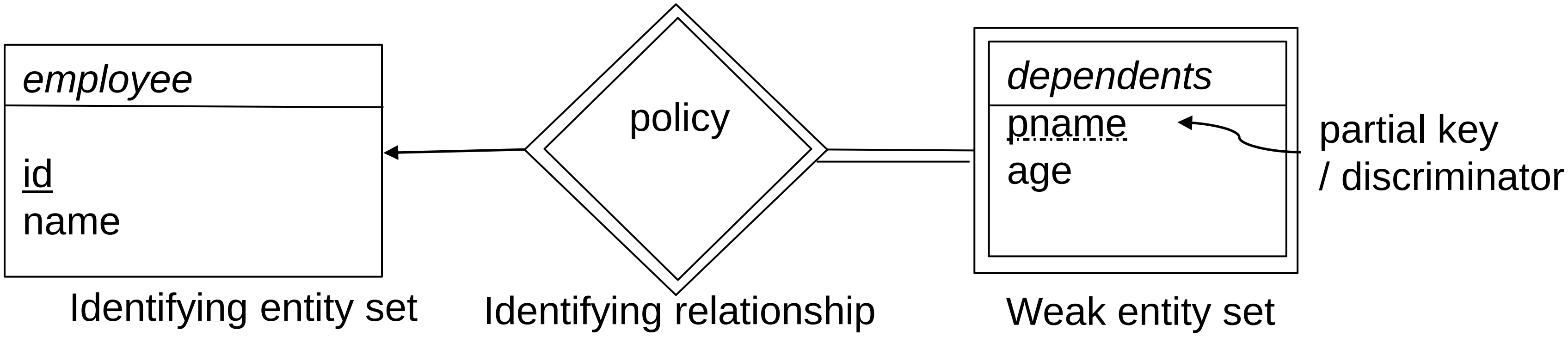
dependents (pname, age)

- An entity in *employee* has a policy covering his dependents
  - A given employee's dependents are likely to have unique names
- In *dependents*
  - each entity is distinct
  - but enough attributes are not present to identify a dependent uniquely
- Treat policy as a special relationship that provides extra information (*id*) required to identify entities in *dependents* uniquely
- pname could be made unique – but conceptually, a dependent is still dependent on an employee, so not a good way to model



Weak Entity Sets (Cont.)

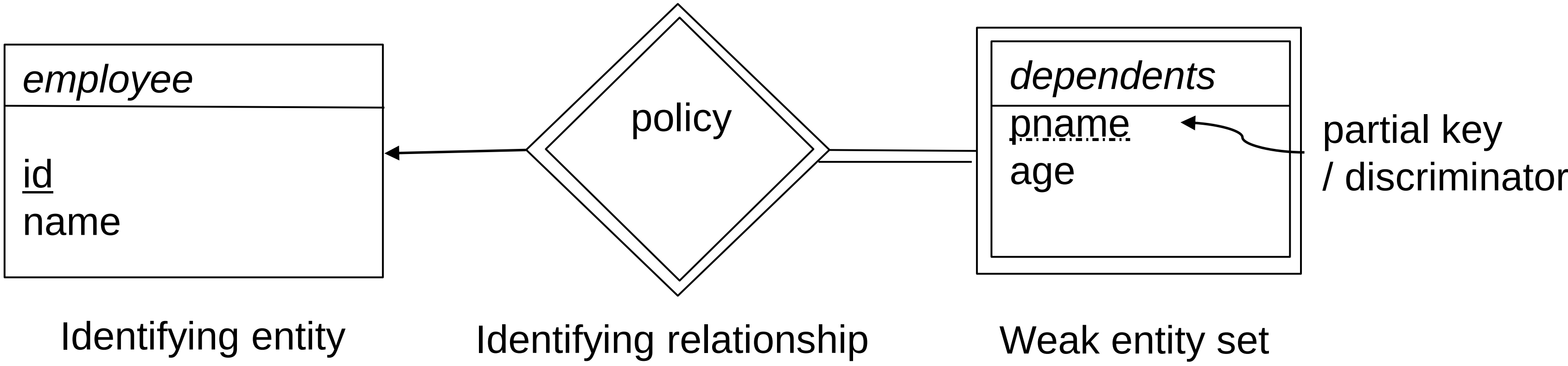
- The notion of **weak entity set** formalizes the above intuition.
- A weak entity set is one whose existence is dependent on another entity, called its **identifying entity set** (or **owner entity set**)
- A primary key is not associated with a weak entity
  - to uniquely identify a weak entity: primary key of the identifying entity and extra attributes called **discriminator** or **partial key**
- An entity set that is not a weak entity set is termed a **strong entity set.**



{id , pname } : uniquely identifies an entity in dependents

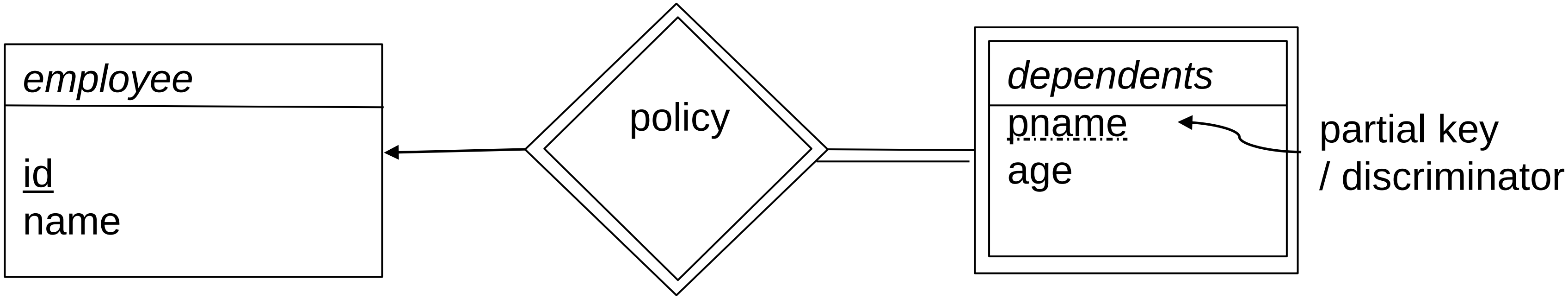
Weak Entity Sets (Cont.)

- Every weak entity must be associated with an identifying entity
  - the weak entity set is said to be **existence dependent** on the identifying entity set.
  - The identifying entity set is said to **own** the weak entity set that it identifies.
  - The relationship associating the weak entity set with the identifying entity set - **identifying relationship**.



Expressing Weak Entity Sets

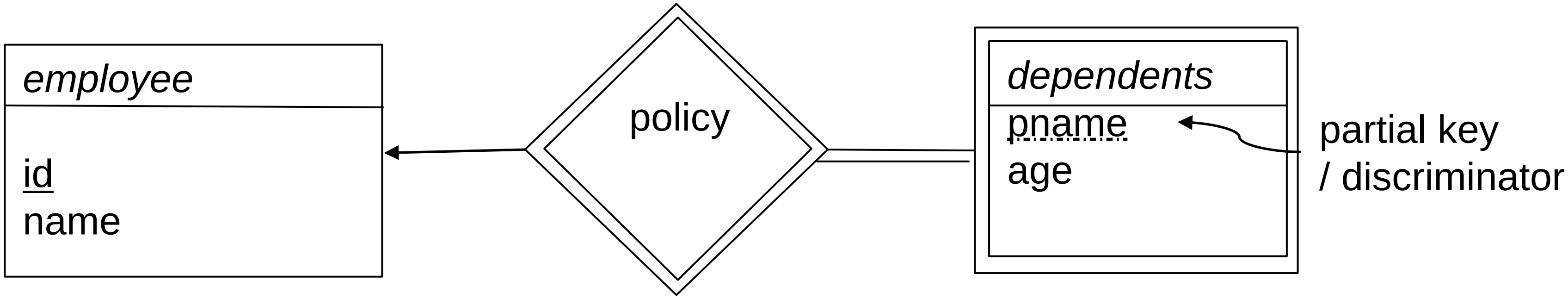
- In E-R diagrams, a weak entity set is depicted via a double rectangle.
- We underline the discriminator of a weak entity set with a dashed line.
- The relationship set connecting the weak entity set to the identifying strong entity set is depicted by a double diamond.



Identifying entity Identifying relationship Weak entity set

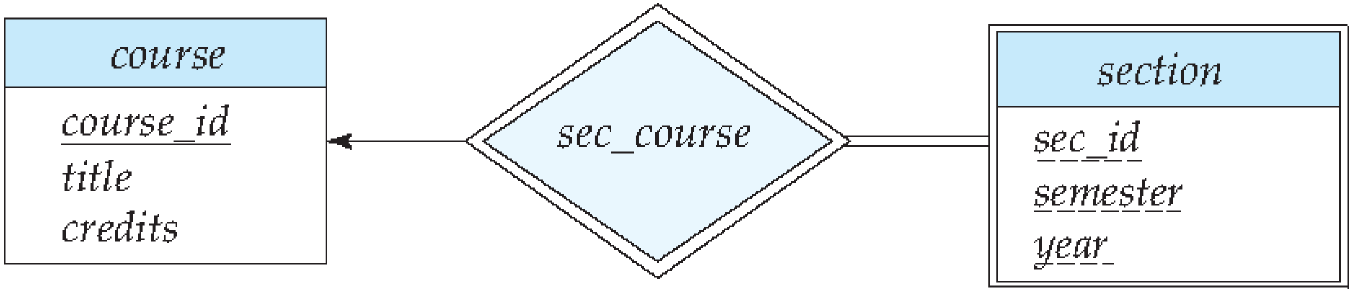
Weak entity sets cont.

- Identifying relationship is many-to-one from the weak entity set to the identifying entity set
  - one-to-one is a special case of many-to-one
- Participation of the weak entity set in the identifying relationship is total
- Identifying relationship
  - should not have any descriptive attributes
  - such attributes can be associated with the weak entity set instead



Identifying entity   Identifying relationship   Weak entity set

Weak entity set – another example

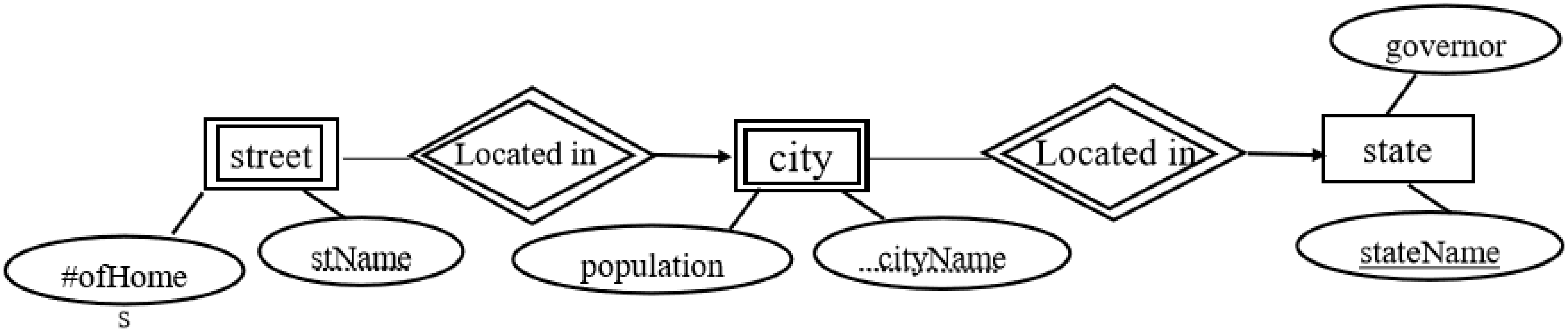


- Primary key for *section* – (*course_id, sec_id, semester, year*)

Note : the relational schema we eventually create from the entity set *section* does have the attribute *course_id* (explained later)*.*

Weak entity set - participation

- A weak entity set can participate
  - In relationships other than the identifying relationships
  - As owner in an identifying relationship of another weak entity set

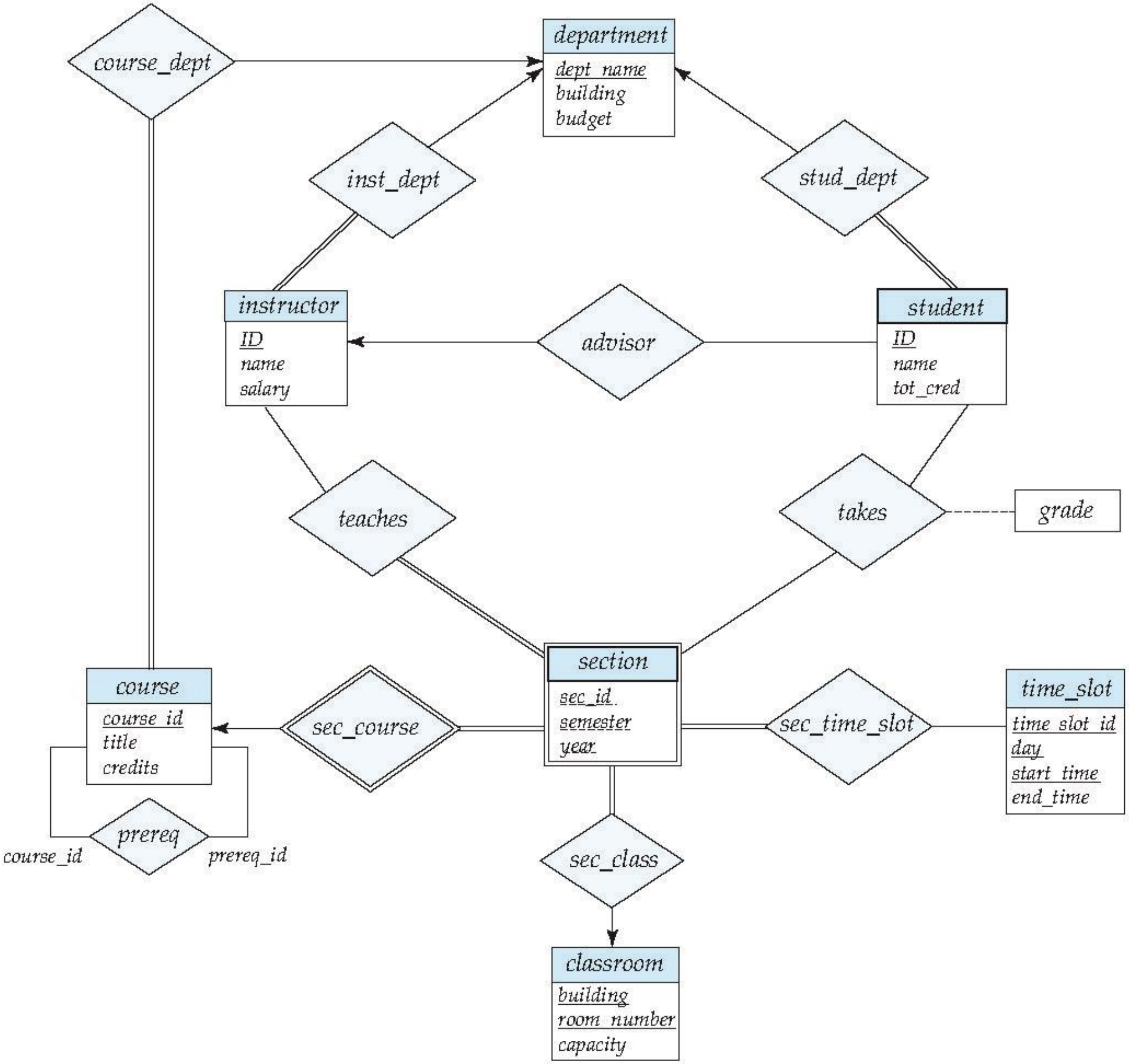


- A weak entity set can have more than one identifying set
  - The weak entity is identified by a combination of entities
  - Primary key: union of primary keys of the identifying entity sets and the discriminator of the weak entity set

Alternative to weak entity set

- A weak entity set may be expressed as a multivalued composite attribute of the owner entity set
- *section* may be expressed as a multivalued composite attribute of *course*
  - A w.e.s may be modelled as a composite attribute when
    - It participates in only the identifying relationship
    - And if it has a few attributes
  - *section* participates in other relationships
  - Therefore section is better modelled as a weak entity set

E-R Diagram for a University Enterprise


course_dept
department
dept_name
building
budget
inst_dept
stud_dept
instructor
ID
name
salary
advisor
student
ID
name
tot_cred
teaches
takes
grade
section
sec_id
semester
year
course
course_id
title
credits
sec_course
sec_time_slot
time_slot
time_slot_id
day
start_time
end_time
prereq
course_id
prereq_id
sec_class
classroom
building
room_number
capacity