Entity-Relationship Model

MSCI346
Lukasz Golab

Acknowledgement: slides derived from material provided by Sliberschatz, Korth and Sudarshan, copyright 2019, www.db-book.com, and by Prof. Wojciech Golab, ECE356

Learning Outcomes

- Building blocks of the entity relationship (E-R) model:
  - entities and entity sets
  - relationships and relationship sets
  - constraints: cardinality, participation
  - primary keys
- E-R diagrams

- Textbook sections (6th ed.): 7.1 to 7.5, 7.7, 7.8

Data in the real world

- A database organizes information about a particular enterprise.
  - potentially large collection of interrelated data
  - set of programs to access the data
  - environment that is **convenient**, **efficient** and **reliable**
- Example applications:
  - Banking – transactions
  - Airlines – reservations, schedules
  - Universities – registration, grades
  - Sales – customers, products, purchases
  - Online retailers – order tracking, customized recommendations
  - Manufacturing – production, inventory, orders, supply chain
  - Human resources – employee records, salaries, tax deductions

Database modeling

- To a first approximation, 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*

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

instructor

ID	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

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

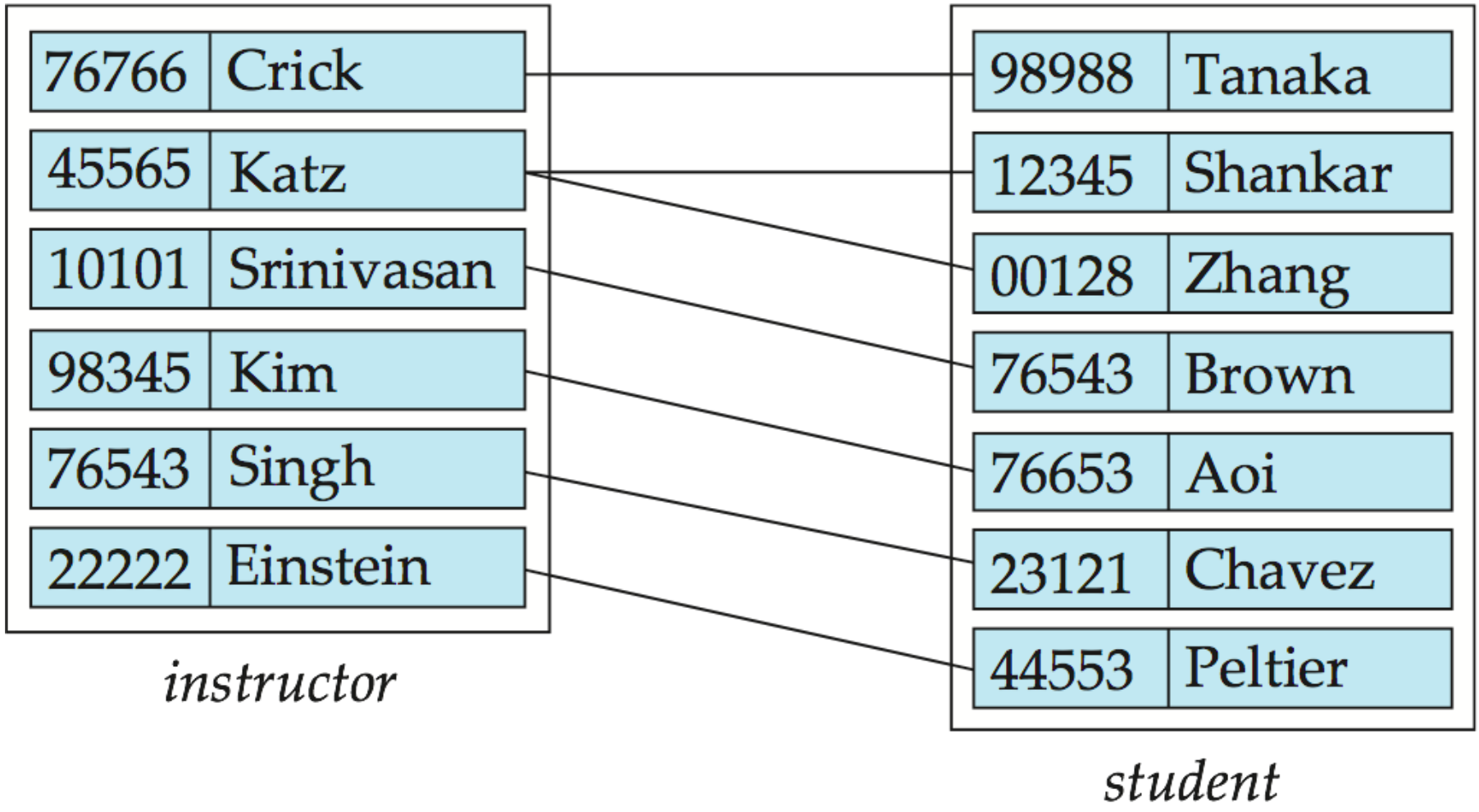
  where $(e_1, e_2, \ldots, e_n)$ is a relationship

  - Example:

$$(44553, 22222) \in advisor$$

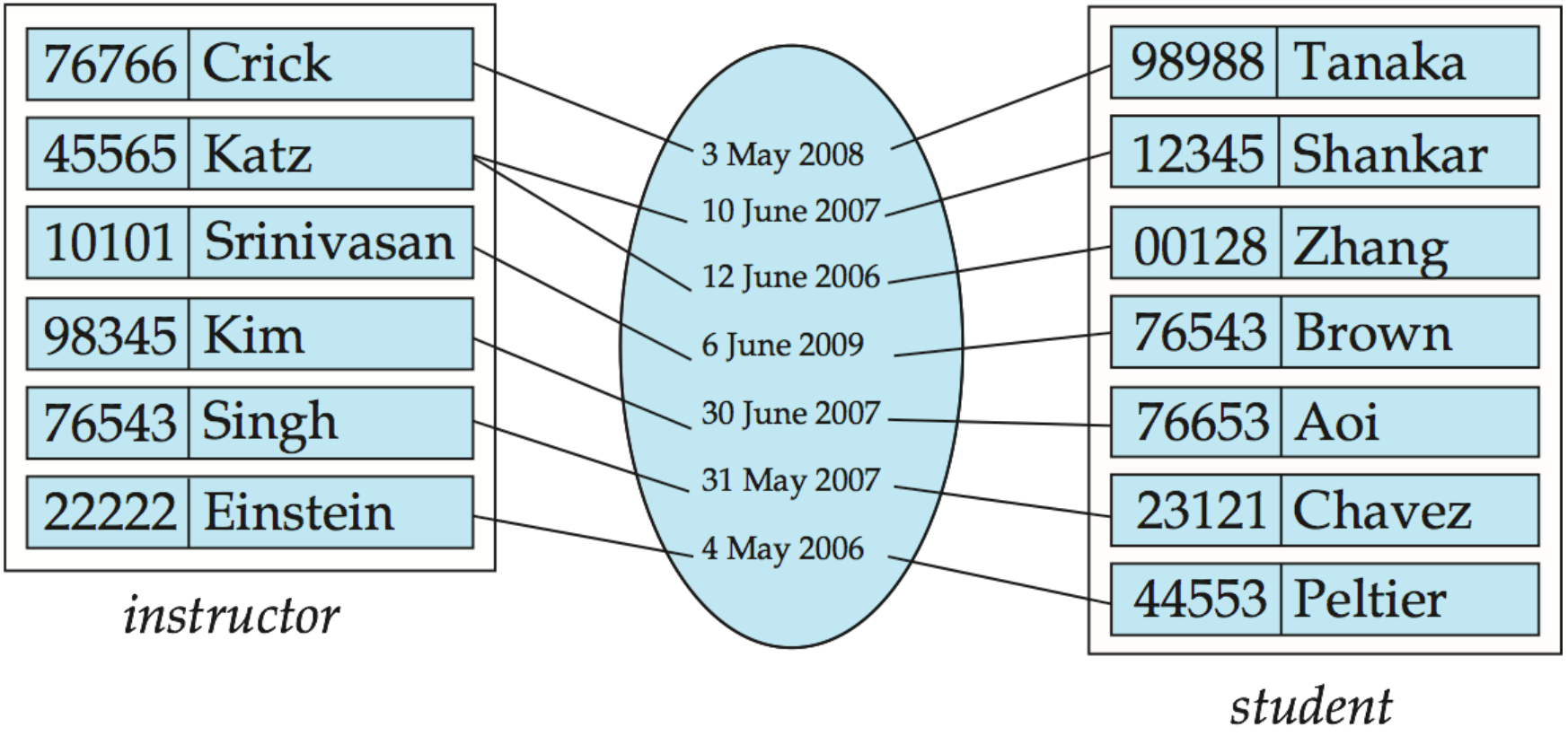
  Note: the ID attribute uniquely identifes students and instructors.

Relationship Set *advisor*



Relationship Sets (Cont.)

- An **attribute** can also be a property of a relationship set.
- For instance, the *advisor* relationship set between entity sets *instructor* and *student* may have a **descriptive attribute** *date* that indicates when the student started being associated with the advisor.



Degree of a Relationship Set

- **binary relationship**
  - involves two entity sets (or degree two)
  - binary relationships are the simplest and work well in practice
  - relationships between more than two entity sets are rare
    - Example: *students* work on research *projects* under the guidance of an *instructor*.

      Relationship *proj_guide* (discussed later on) is a ternary relationship between *instructor, student,* and *project*.

      Food for thought: in this particular example, can an instructor have a project without any students?

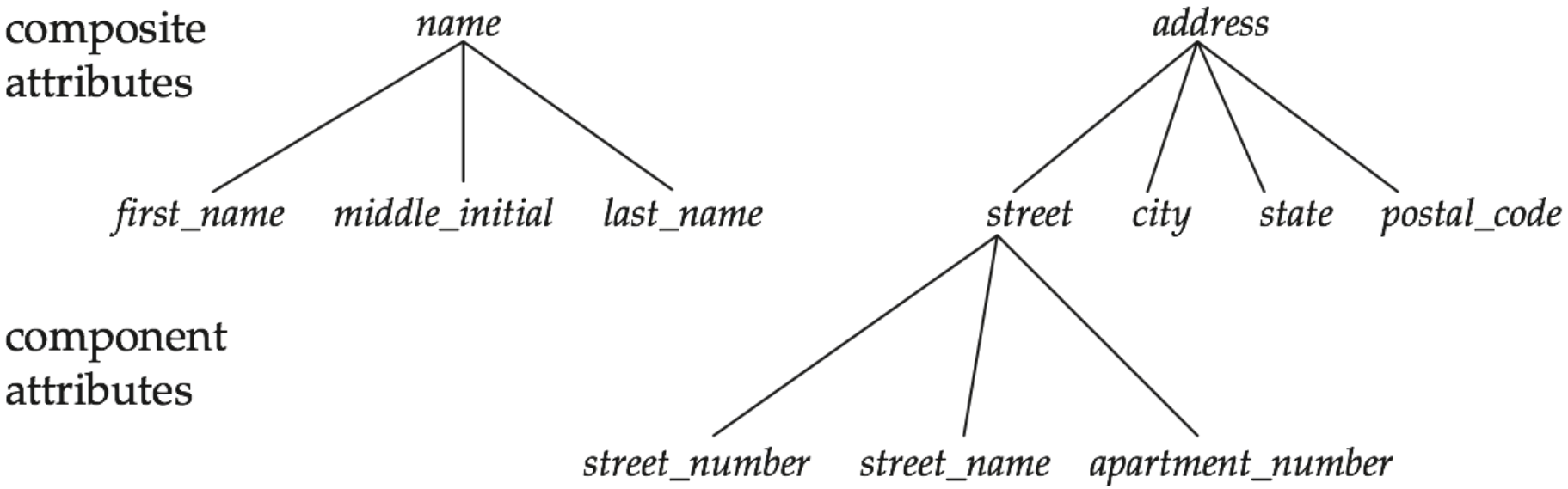
Attributes

- An entity is represented by a set of attributes, which are 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 an attribute
  - example: salary is a non-negative integer
- Attribute types:
  - **simple** versus **composite**
    - example of simple attribute: *credits*
    - example of composite attribute: *address*
  - **single-valued** versus **multivalued**
    - example of single-valued attribute: *course_ID*
    - example of multivalued attribute: *phone_numbers*

Composite Attributes

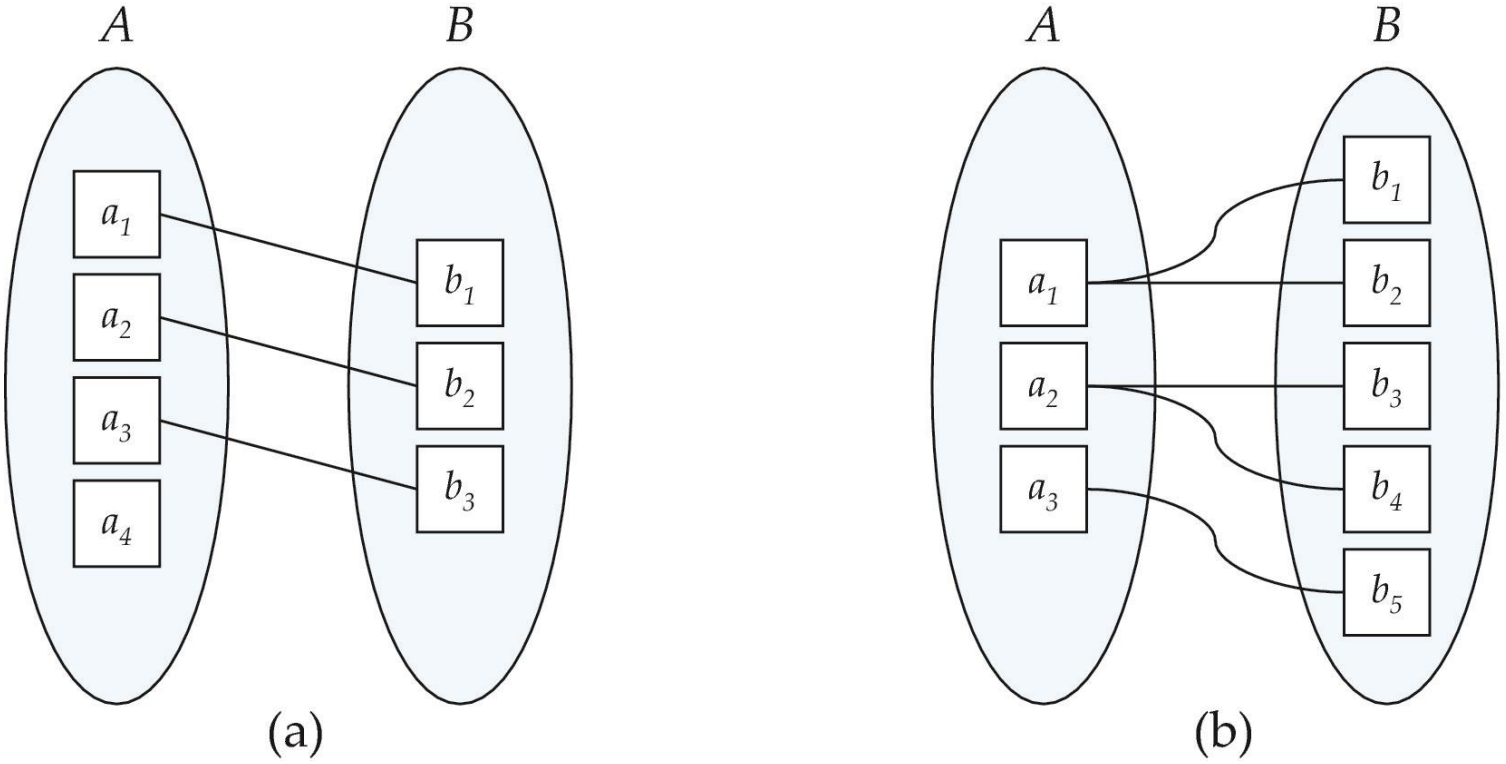
- A **composite attribute** comprises a collection of **component attributes**. Such attributes may be composed recursively, as shown below (*address* contains *street*, which contains *street_name*).



Mapping Cardinality Constraints

- **Cardinality constraints** express the number of entities to which another entity can be associated via a relationship set.
- The constraints reflect the general structure of the data rather than the structure in a specific data set.
  - example: *advisor* relationship (from instructor to student) is generally 1:N or N:N even though at a given time each instructor may have only one student
- They are 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 (1:1)
  - one-to-many (1:N)
  - many-to-one (N:1)
  - many-to-many (N:N)

Mapping Cardinalities

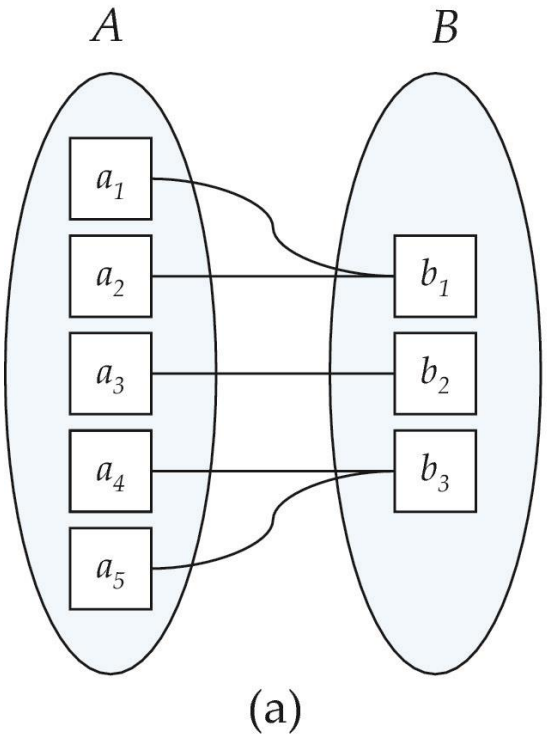


one to one

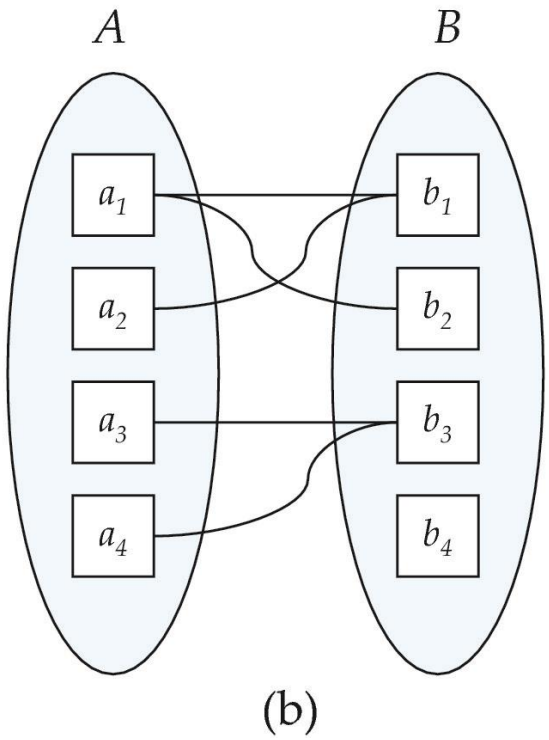
one to many

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

Mapping Cardinalities



many to one



many to many

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

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*

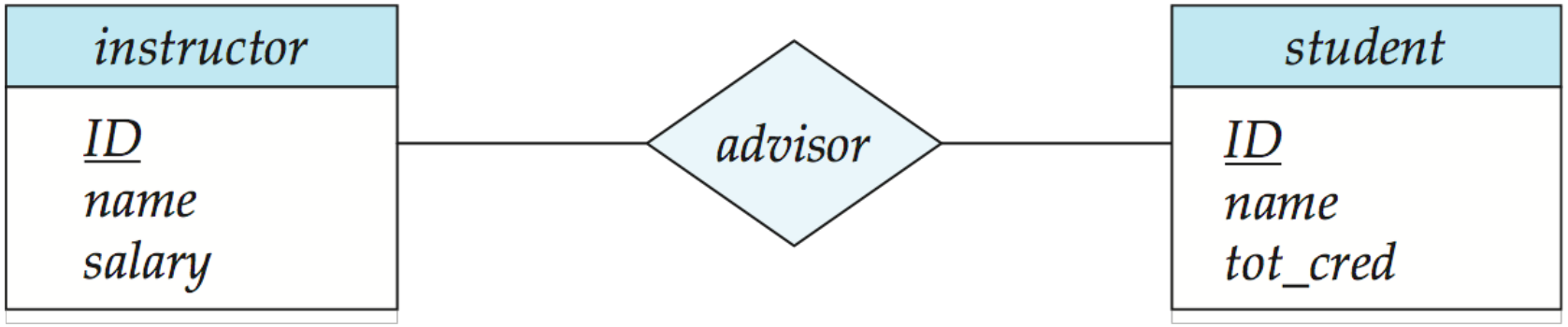
  Note: “minimal” in this context means that if we take out any attribute, we end up with something that is no longer a super key.

- Although several candidate keys may exist, one of the candidate keys is selected by the database designer to be the **primary key**.

Keys for Relationship Sets

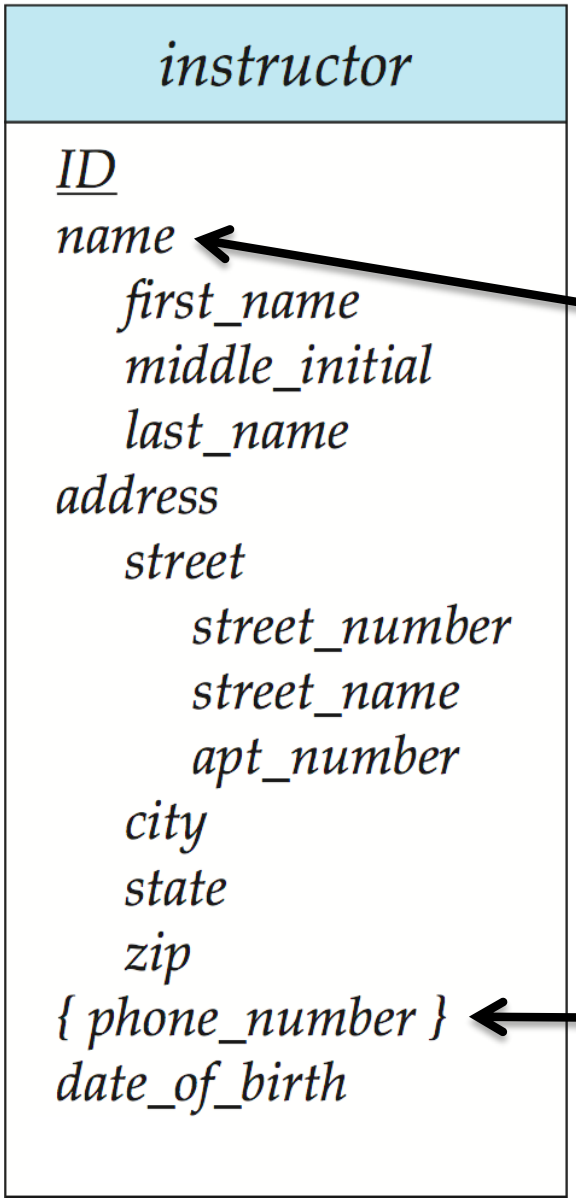
- The combination of primary keys of the participating entity sets forms a super key of a relationship set.
  - (*student.ID, instructor.ID*) is the super key of *advisor*
  - Note 1: **dotted notation** (e.g., *student.ID* vs. *instructor.ID*) is used to resolve attribute name collisions
  - Note 2: **a pair of entities can participate at most once in a particular binary relationship set.**
    - Example: if we wish to track multiple meeting dates between a student and her advisor, we cannot assume a distinct relationship for each meeting. Instead, we can add a multivalued descriptive attribute to the relationship set. (More on this coming later.)
- Must consider the mapping cardinality of the relationship set when deciding what are the candidate keys.
- Need to consider semantics of relationship set in selecting the primary key if there is more than one candidate key.

E-R Diagrams



- Rectangles represent entity sets.
- Diamonds represent relationship sets.
- Attributes are listed inside entity rectangle.
- Underlined attributes are part of the primary key.

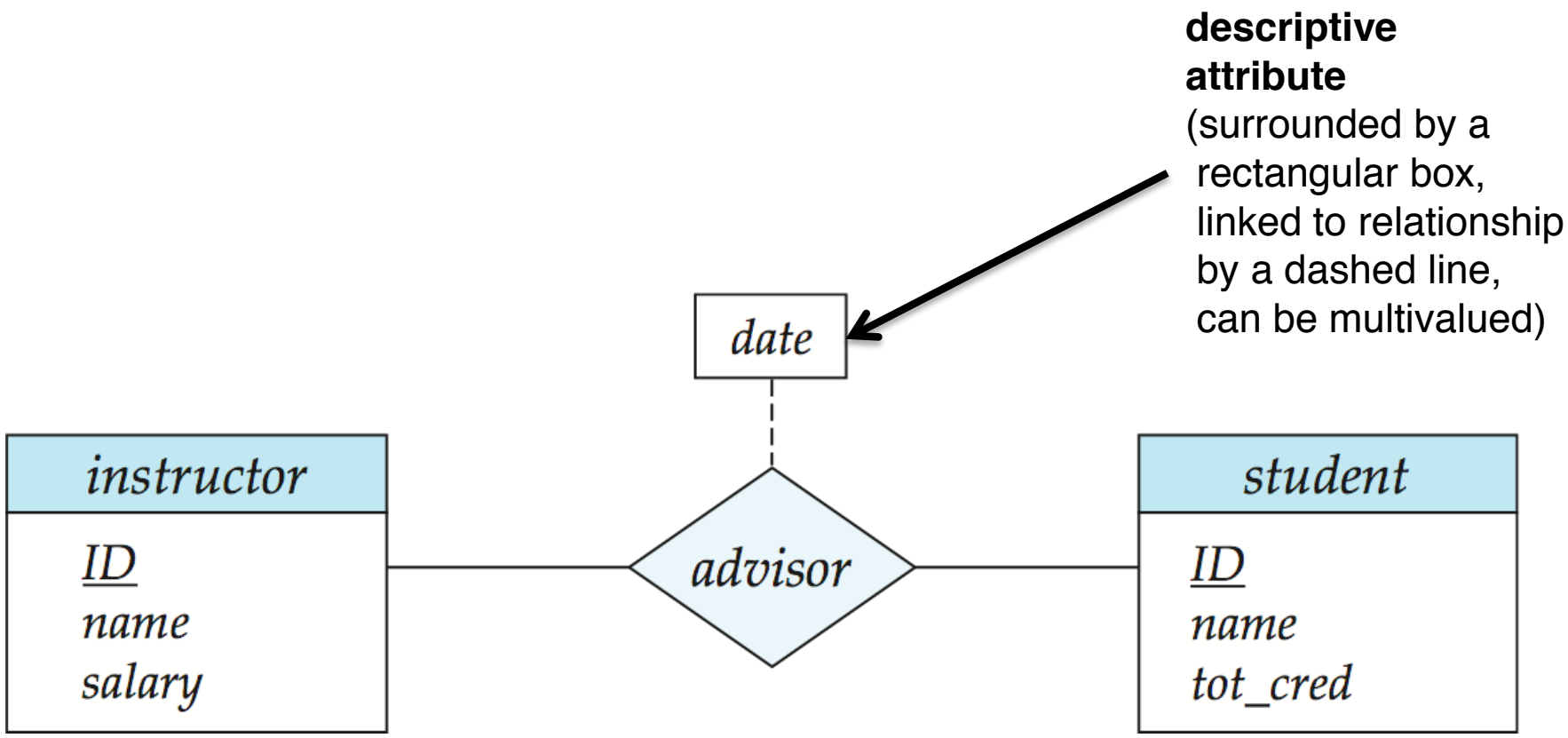
Entity Set with Composite & Multivalued Attributes



composite attribute (components are given in an indented list)

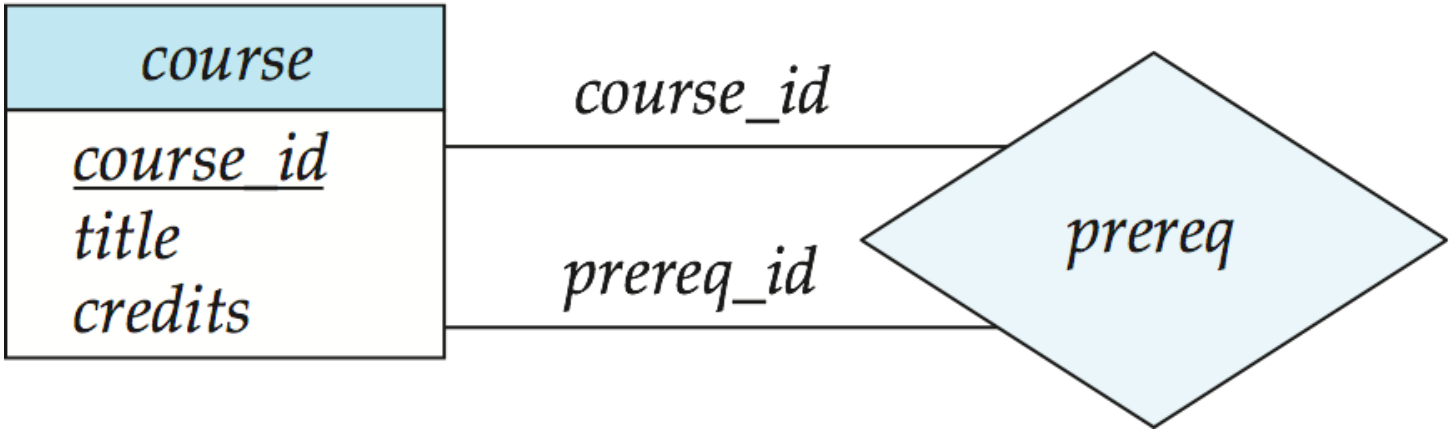
multivalued attribute (surrounded by curly braces)

Relationship Set with Descriptive Attributes



Roles

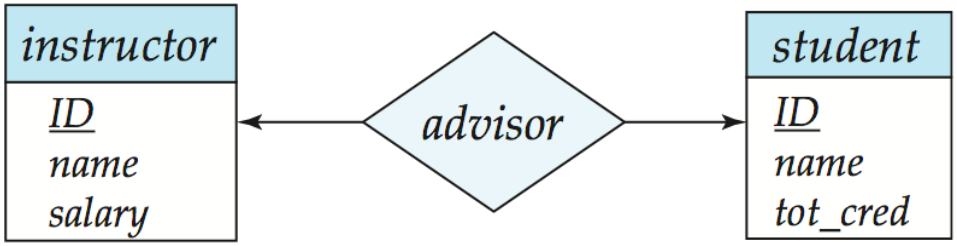
- The entity sets of a relationship need not be distinct.
- Each occurrence of an entity set plays a “role” in the relationship.
- The labels *“course_id”* and *“prereq_id”* below are called **roles**.



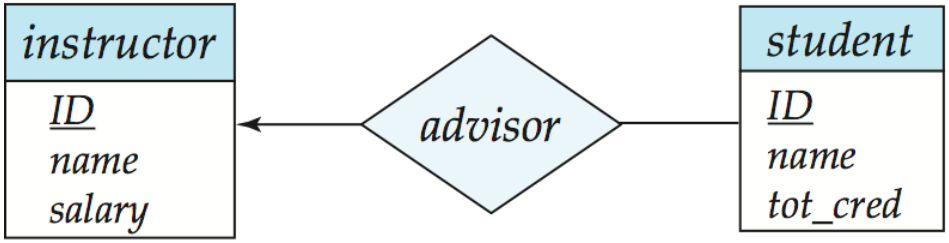
(*course_id.course_id, prereq_id.course_id*) is the super key of *prereq*

Cardinality Constraints

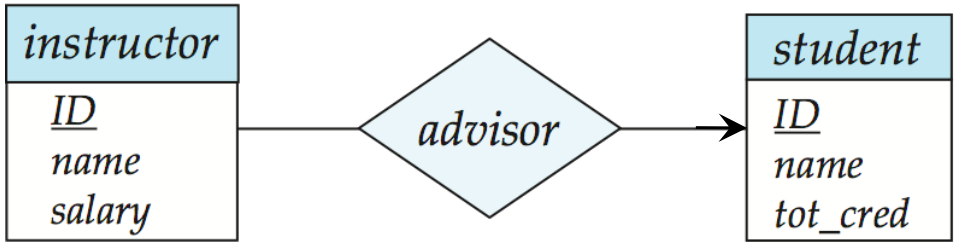
- We express cardinality constraints by drawing either a directed line (→), signifying "one," or an undirected line (—), signifying "many", between the relationship set and the entity set.



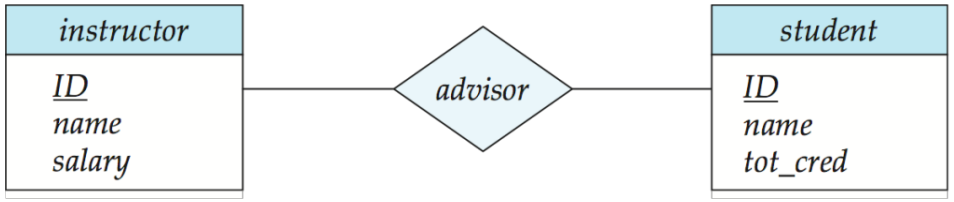
one to one
- at most one instructor per student
- at most one student per instructor



one to many
- at most one instructor per student



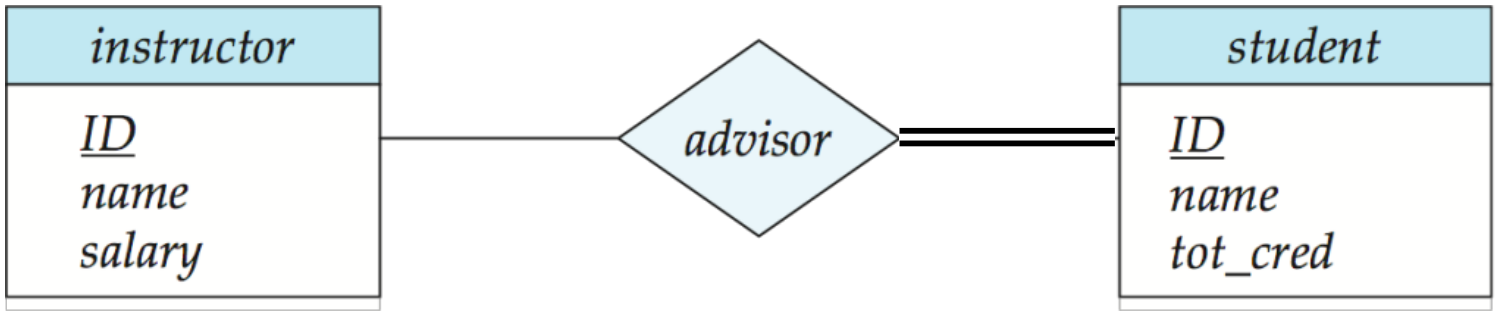
many to one
- at most one student per instructor



many to many

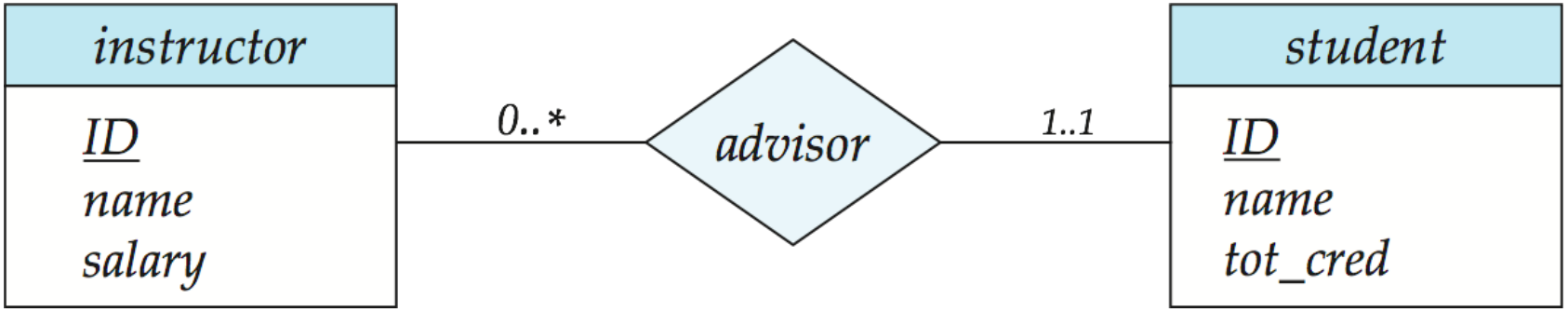
Participation of an Entity Set in a Relationship Set

- **Total participation** (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set.
- Example:
  - At Waterloo a graduate student may have multiple advisors, an advisor may have multiple graduate students, and every graduate student must have at least one advisor.
  - On the other hand an instructor is not required to advise any graduate students.



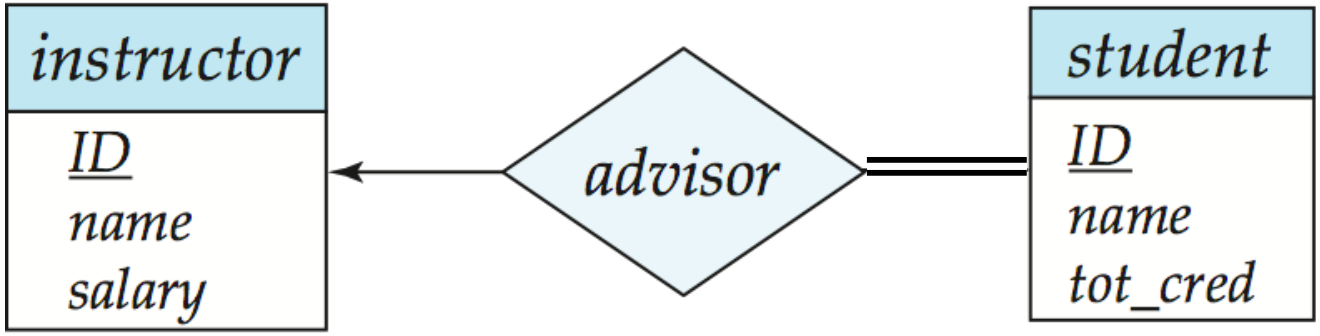
Alternative Notation for Cardinality Limits

- **Cardinality limits** can also express participation constraints.

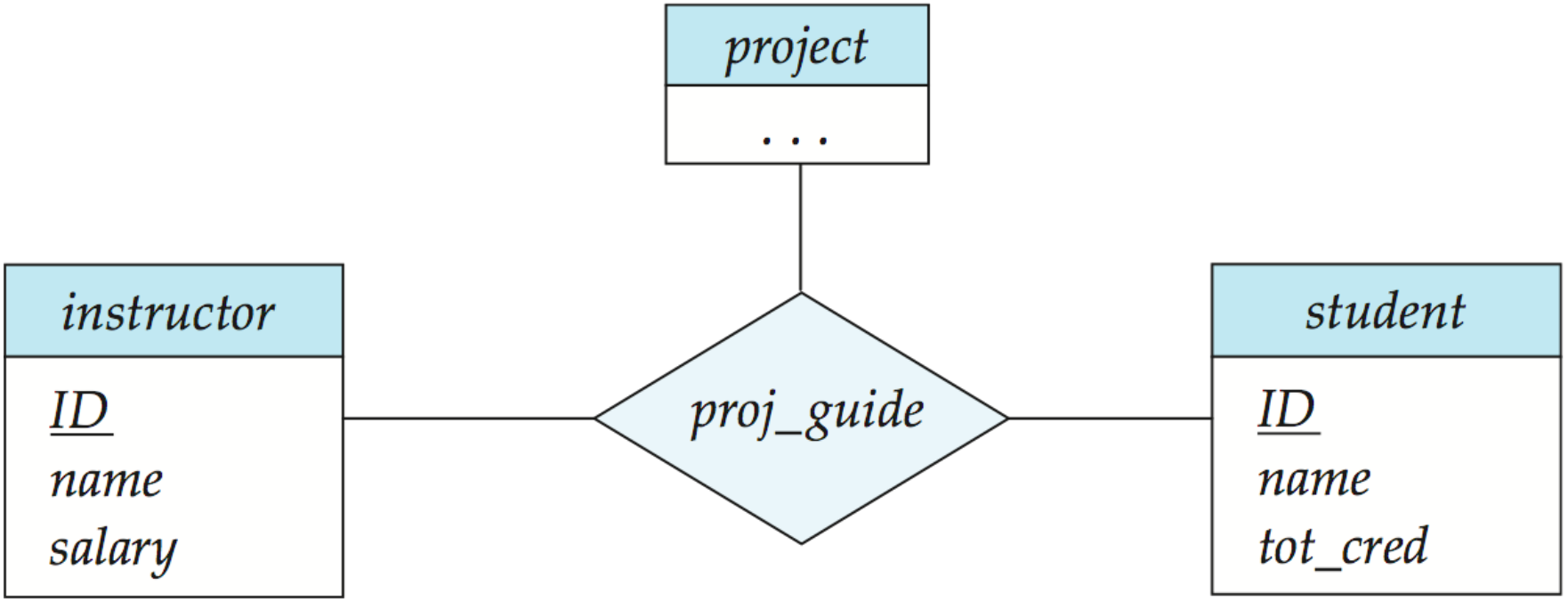


(Interpretation: every student must be advised by exactly one instructor, and an instructor may advise any number of students including possibly zero.)

is equivalent to



E-R Diagram with a Ternary Relationship



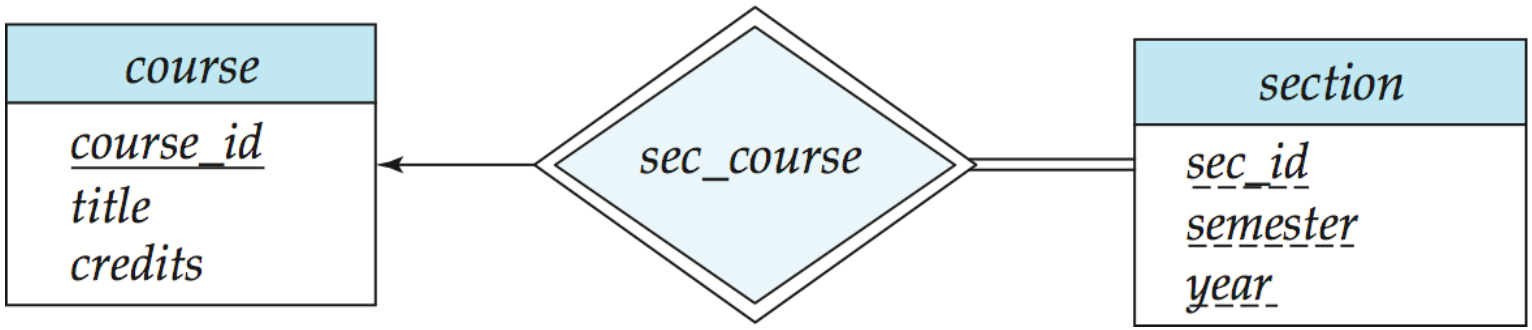
Food for thought: What do cardinality constraints mean for a ternary relationship?

Weak Entity Sets

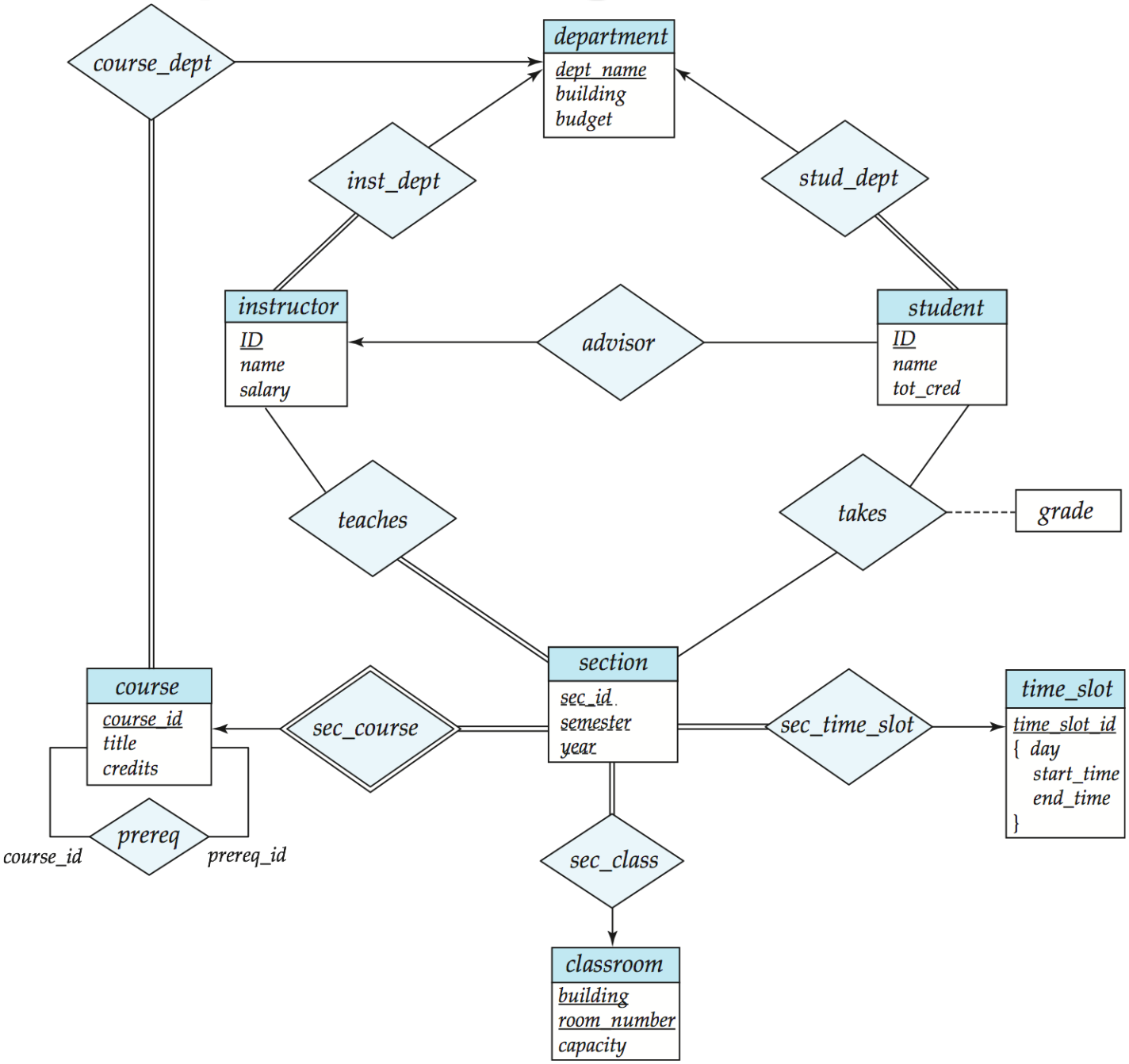
- An entity set whose attributes do not provide a primary key is referred to as a **weak entity set**. (Example on next slide.)
- The existence of a weak entity set depends on the existence of an **identifying entity set**.
  - The weak entity set must relate to the identifying entity set via a one-to-many relationship set from the identifying to the weak entity set, with total participation of 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 a given weak entity among all the other entities that map to the same identifying entity.
- The primary key of a weak entity set comprises the primary key of the identifying entity set plus the discriminator of the weak entity set. (Thus, the primary key of the weak entity set does exist but cannot be formed using only the attributes of the weak entity set.)

Weak Entity Sets (Cont.)

- We underline the discriminator of a weak entity set with a dashed line.
- We place the identifying relationship of a weak entity in a double diamond.
- Primary key for *section:* (*course_id, sec_id, semester, year*)
- The primary key of the weak entity is a super key of the identifying relationship.

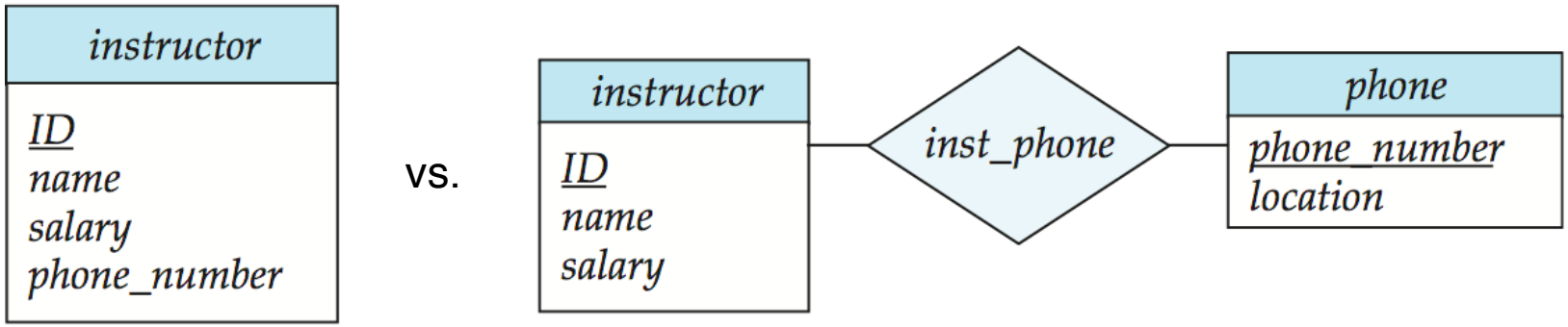


Example E-R Diagram for a University



Design Issues

Use of entity sets vs. attributes.

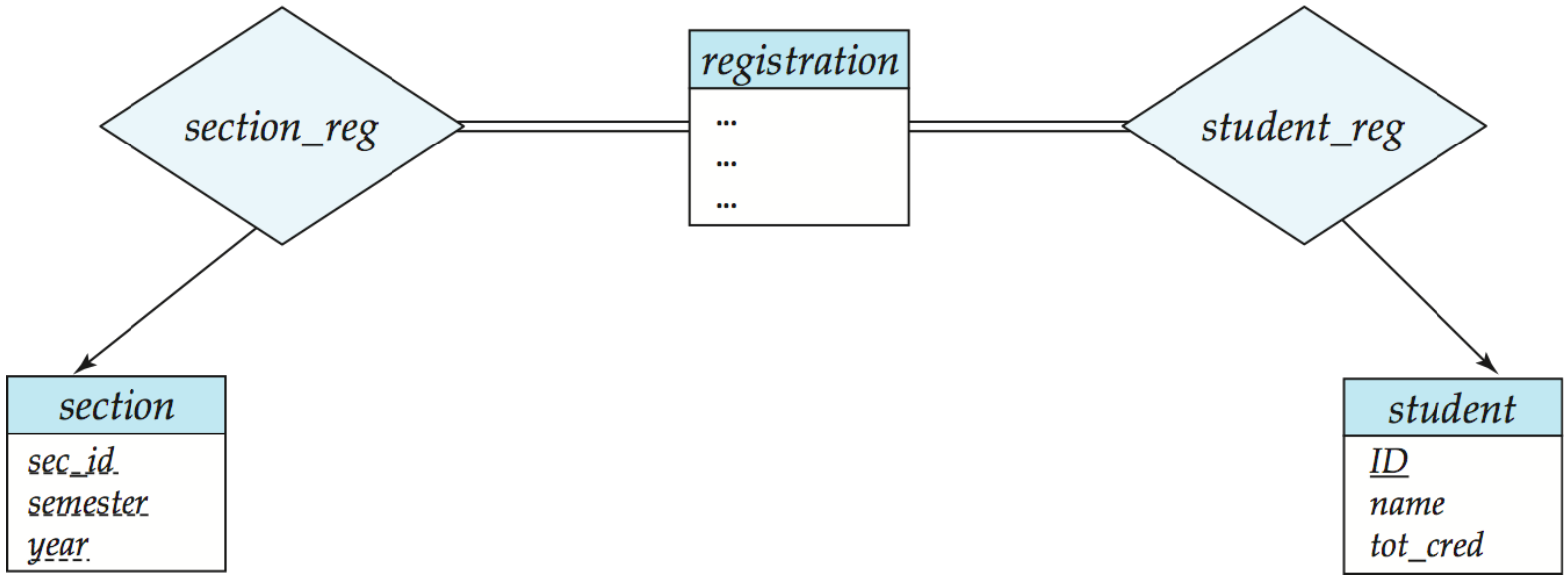


- Use of *phone* as an attribute is simpler.
- Use of *phone* as an entity is more expressive because it allows representing additional information about phone numbers, such as location (e.g,. home, office, or mobile).

Design Issues (Cont.)

Use of entity sets vs. relationship sets (actors vs. actions).

- Possible guideline is to use a relationship set to describe an action that occurs between entities.
- Example: *student takes section.*
  - If the registrar's office associates more complex information with each registration record, it might be best to replace the "takes" relationship set with a registration entity set as shown below:



Design Issues (Cont.)

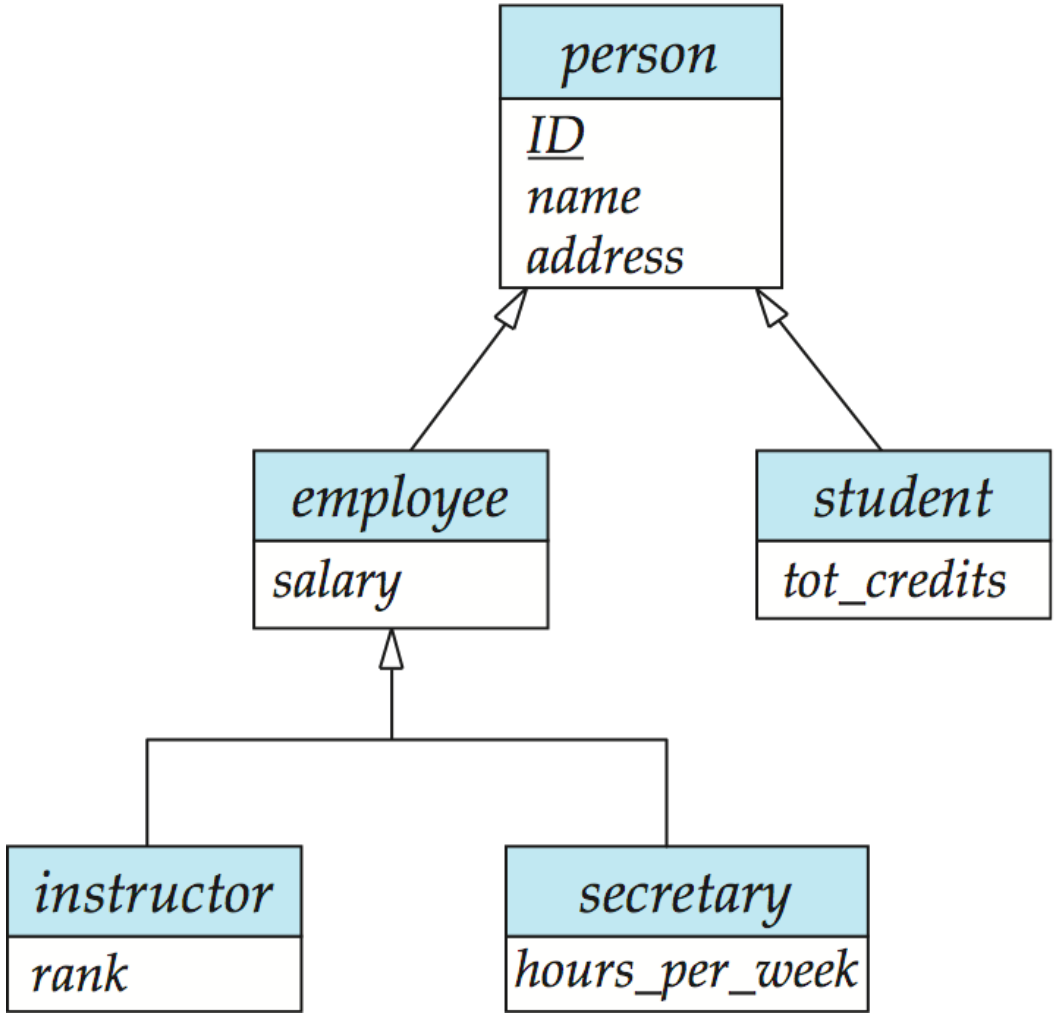
Binary versus n-ary relationship sets.

- Although it is possible to replace any nonbinary (*n*-ary, for $n > 2$) relationship set by a collection of distinct binary relationship sets, an *n*-ary relationship set shows more clearly that several entities participate in a single relationship.
- Some relationships that appear to be non-binary may be better represented using binary relationships.
  - Example: 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 the mother being known).
- Nevertheless, there are some relationships that are naturally non-binary.
  - Example: *proj_guide* (see earlier slide).

Specialization and Generalization

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

Specialization Example

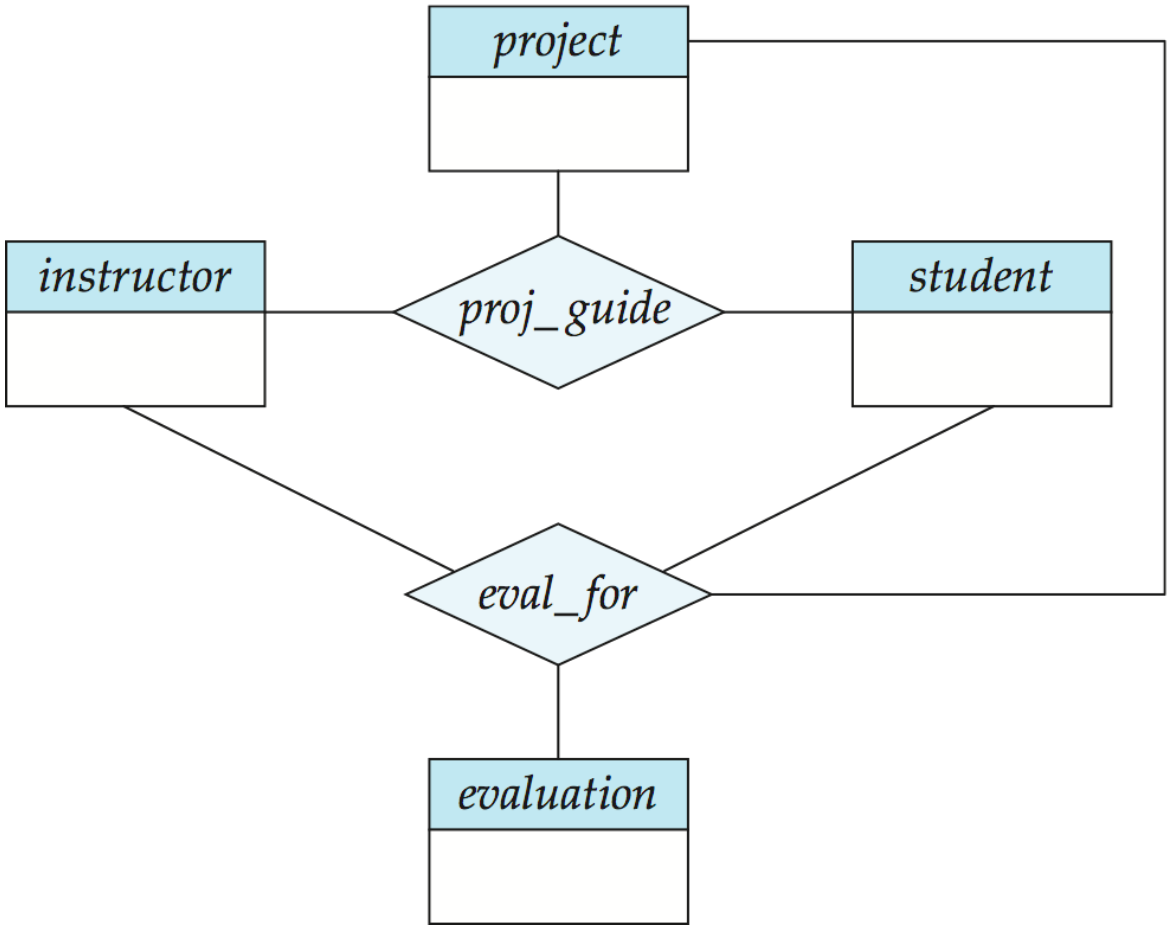


(**ordinary generalization is indicated by separate arrows from lower-level entities to higher-level entities,** a person can be both an employee and a student)

(**disjoint generalization is indicated by combined arrows from lower-level entities to higher-level entities,** an employee cannot be both an instructor and a secretary)

Aggregation

- Consider the ternary relationship *proj_guide*, discussed earlier.
- Suppose we want to record evaluations of a student by a guide on a project.

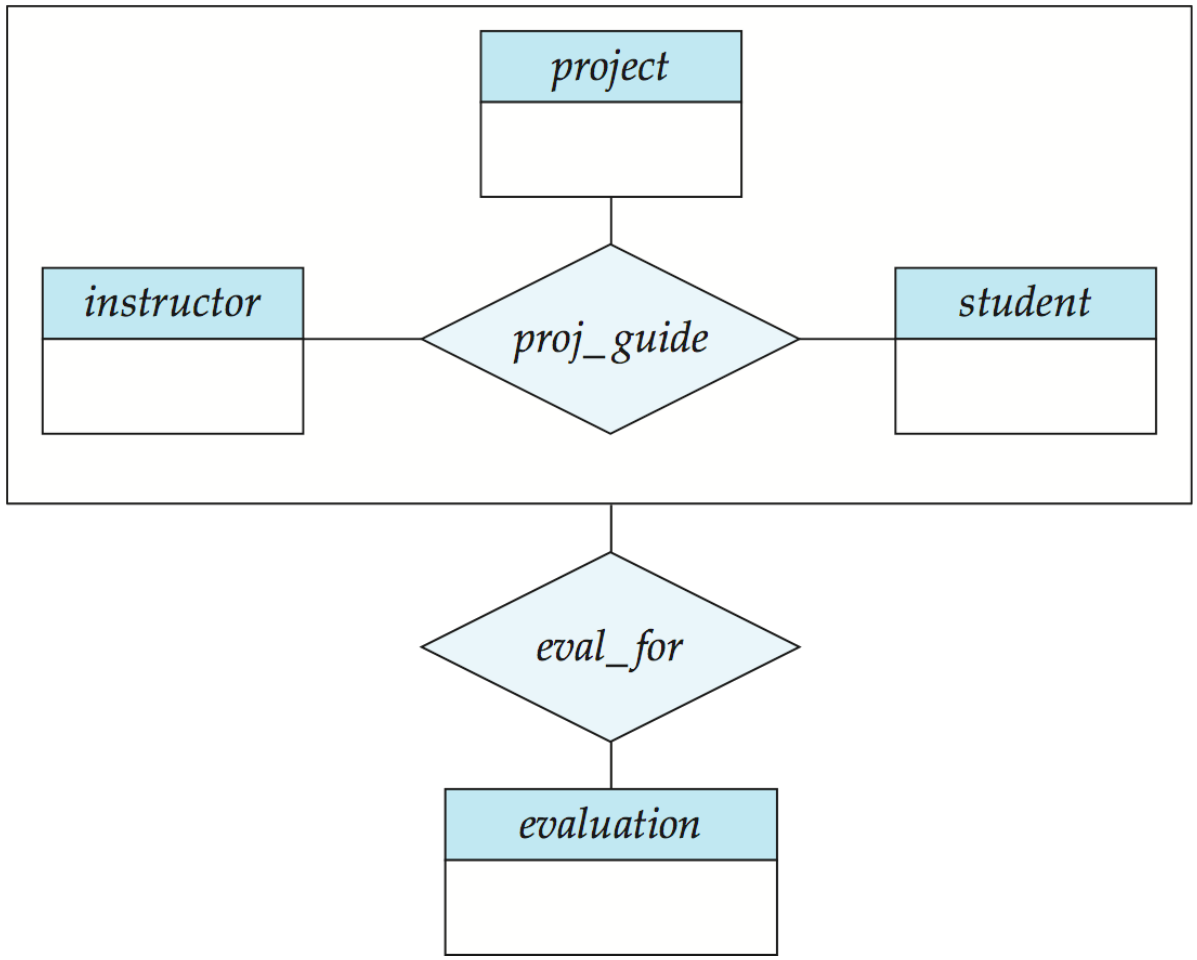


Aggregation (Cont.)

- Relationship sets *eval_for* and *proj_guide* represent overlapping information:
  - every *eval_for* relationship ties together a student, instructor, and project, just like the *proj_guide* relationship
  - that said, we cannot discard *proj_guide* since some *proj_guide* relationships may not correspond to any *eval_for* relationships
- **Aggregation** – treat the entire relationship as an abstract entity.
  - removes redundancy
  - allows relationships between entities and relationships (or between relationships and relationships)

Aggregation (Cont.)

(**aggregate entity indicated by outer rectangle**)



Summary of E-R Design Decisions

- The use of an attribute vs. 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 weak vs. ordinary entity set.
- The use of specialization/generalization, which contributes to modularity in the design.
- The use of aggregation, which allows us to treat the aggregate entity set as a single unit without concern for the details of its internal structure.