The picture can't be displayed.

Topic 2: Entity-Relationship Model (Chapter 6)

Database System Concepts

©Silberschatz, Korth and Sudarshan

(Modified for CS 4513)

Topic 2: Entity-Relationship Model

- Design Process
- Modeling
- Constraints
- E-R Diagram
- Design Issues
- Weak Entity Sets
- Extended E-R Features
- ER Design Examples
- Reduction to Relation Schemas (will be discussed in Topic 3)

©Silberschatz, Korth and Sudarshan

Design Process

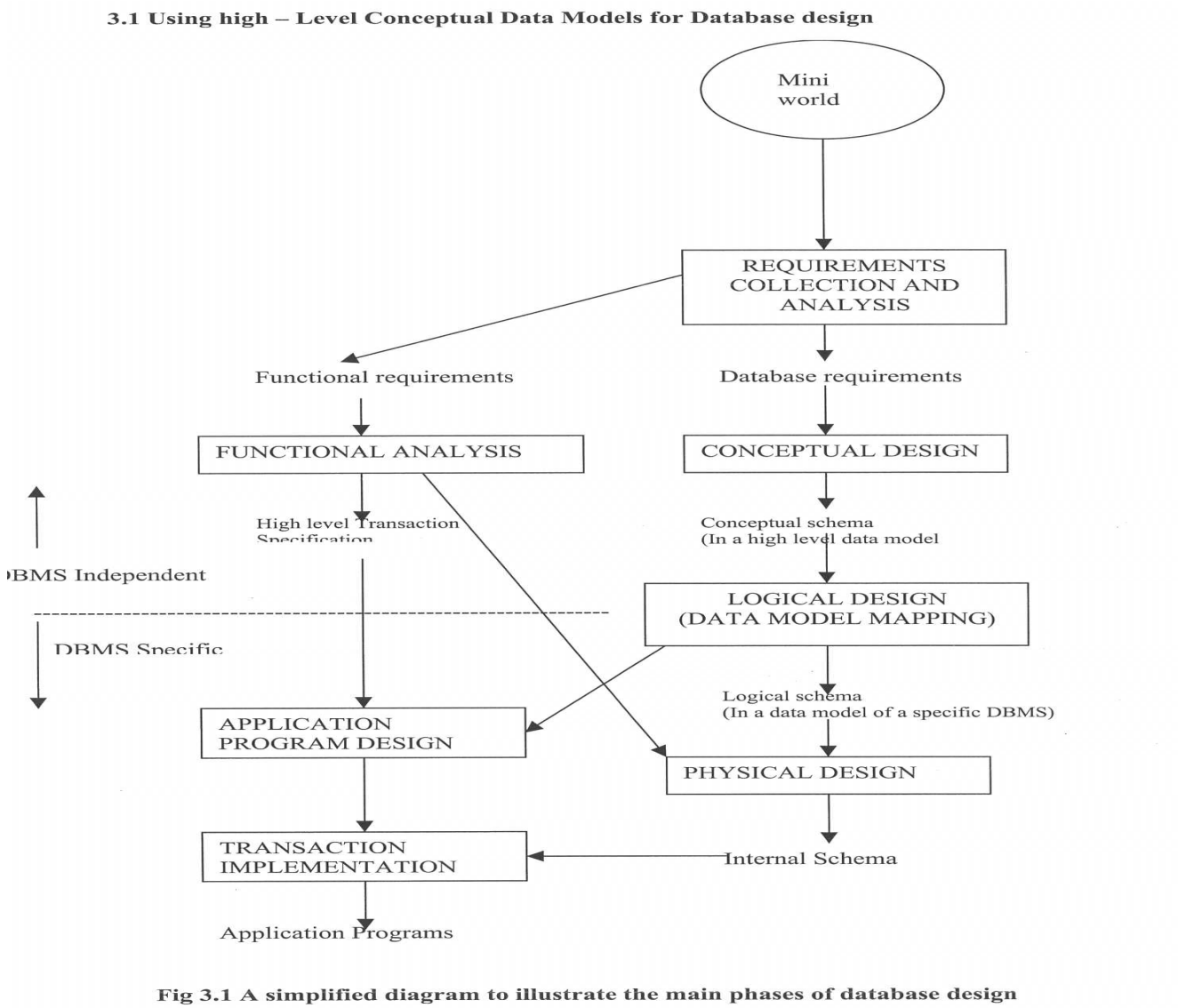


Fig 3.1 A simplified diagram to illustrate the main phases of database design

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

©Silberschatz, Korth and Sudarshan

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

©Silberschatz, Korth and Sudarshan

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

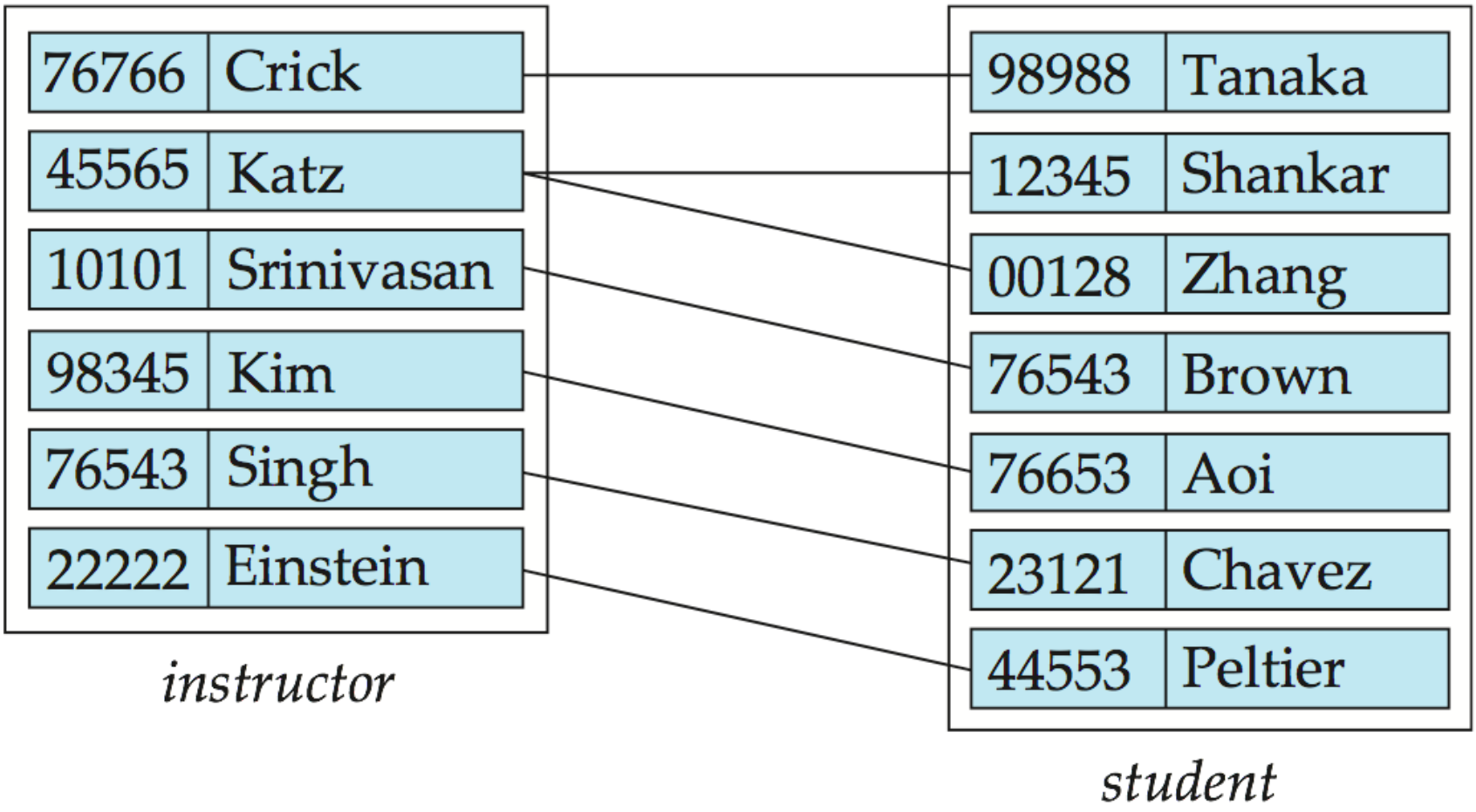
$$\{(e_1, e_2, \ldots e_n) \mid e_1 \in E_1, e_2 \in E_2, \ldots, e_n \in E_n\}$$

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

  - Example:

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

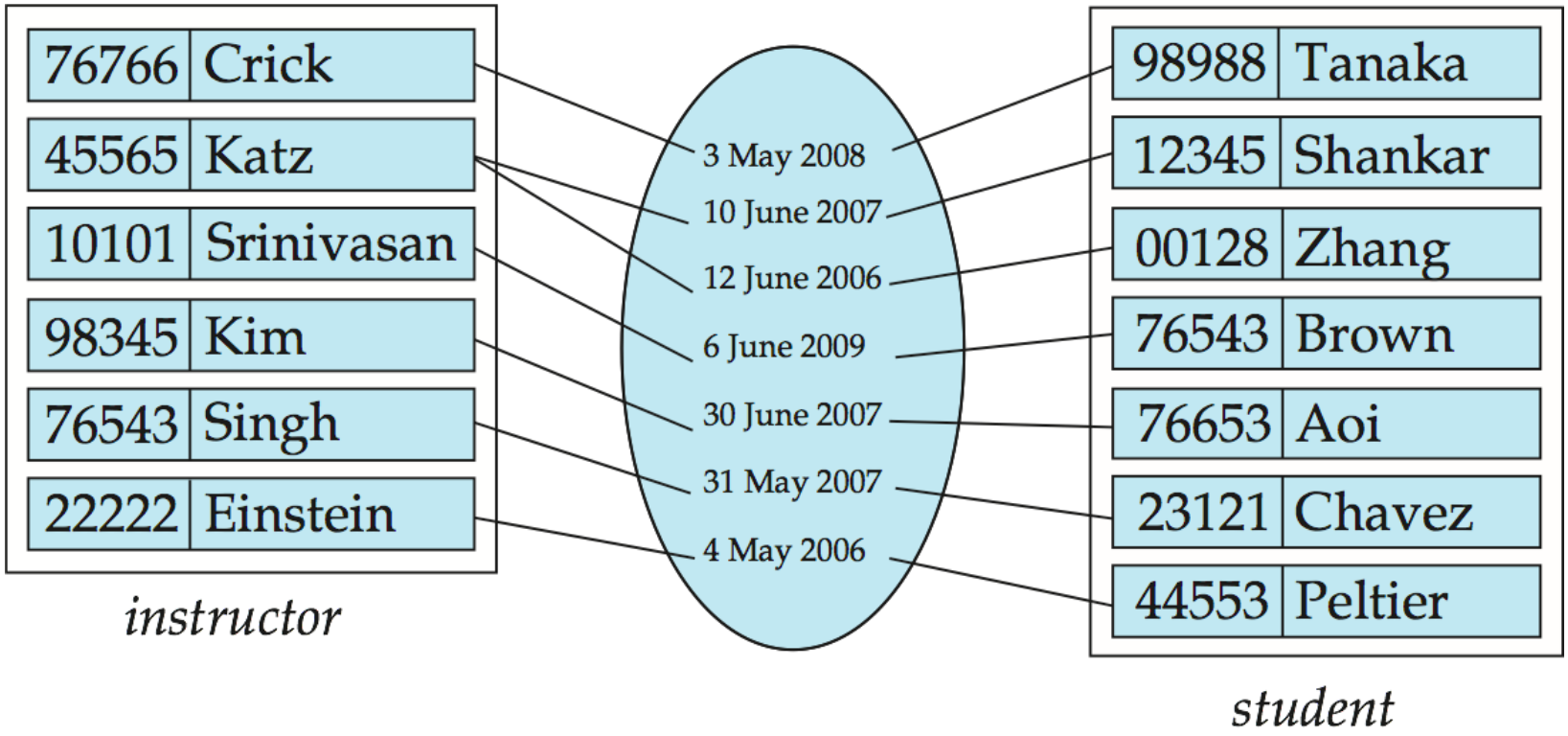
Relationship Set *advisor*



 ©Silberschatz, Korth and Sudarshan

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



Degree of a Relationship Set

- **binary relationship**
  - Involve two entity sets (or degree two).
  - Example: *instructors teach classes.*
    - Binary relationship *teach* between two entity sets *instructor* and *class*

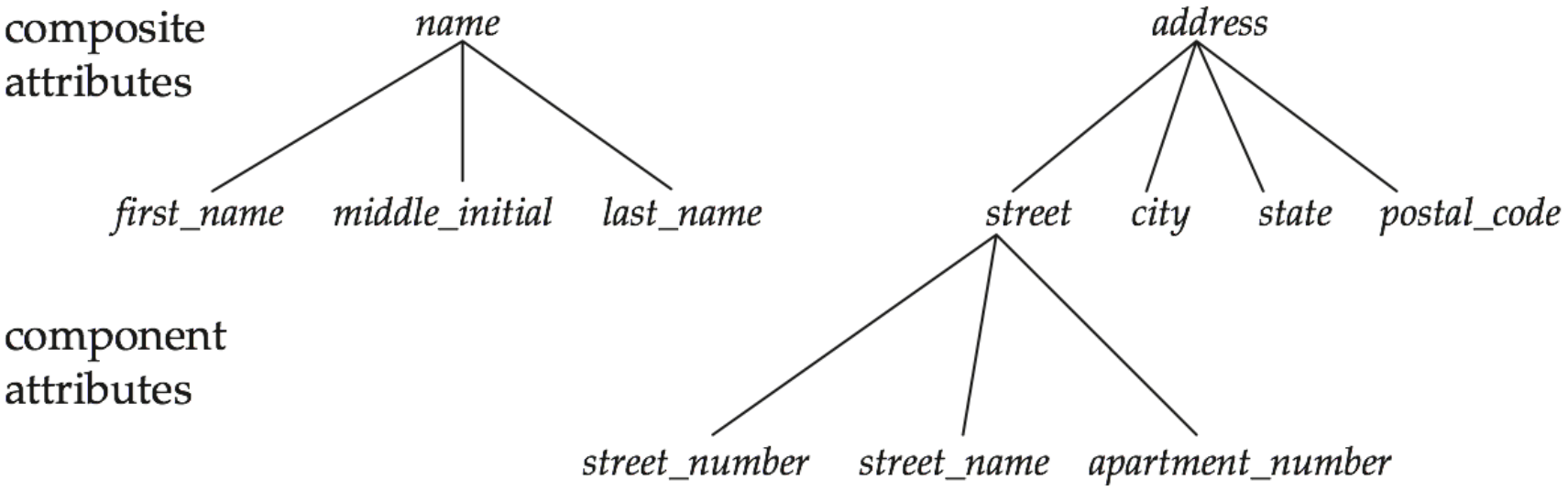
- Non-binary relationship:
  - Involve more than two entity sets.
  - example: *students* work on research *projects* under the guidance of an *instructor*.
    - relationship *proj_guide* is a ternary relationship between *instructor, student,* and *project.*

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 attribute:** composed of only one part (i.e. not divisible)
    - e.g. student id, instructor salary
  - **Composite attribute**: composed of several subparts

Composite Attributes



Attributes (cont.)

- Attribute types (cont):
  - **Single-valued attribute:** has only one value
    - e.g. first name, last name
  - **Multivalued attribute**: has multiple values
    - Example: phone_numbers, student_degree
  - **Derived attribute**:
    - Can be computed from other attributes
    - Example: age, given date_of_birth

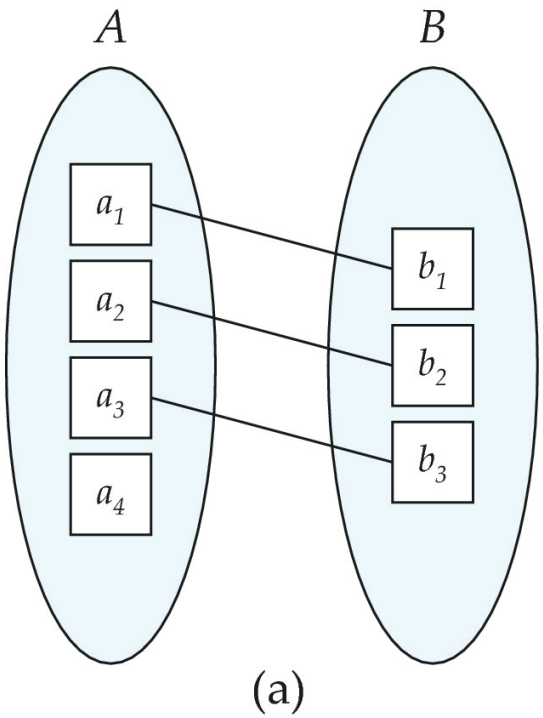
©Silberschatz, Korth and Sudarshan

Mapping Cardinality Constraints

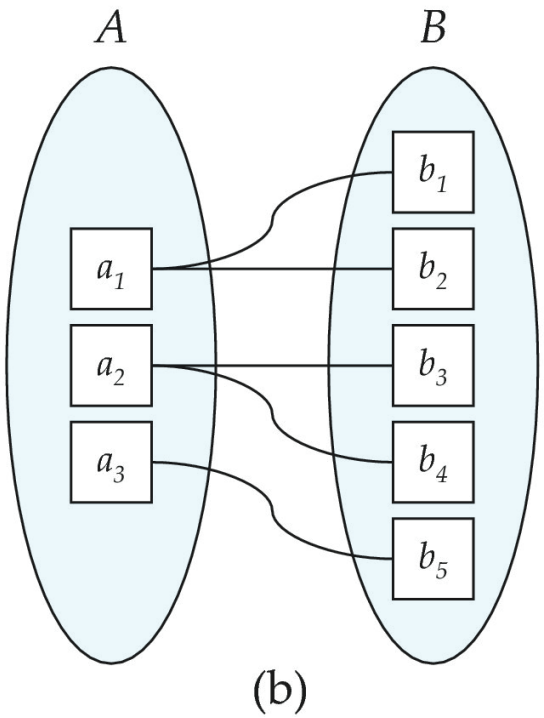
- Express the number of entities to which another entity can be associated via a relationship set.
- Most useful in describing binary relationship sets.
- For a binary relationship set the mapping cardinality must be one of the following types:
  - One to one
  - One to many
  - Many to one
  - Many to many

 ©Silberschatz, Korth and Sudarshan

Mapping Cardinalities



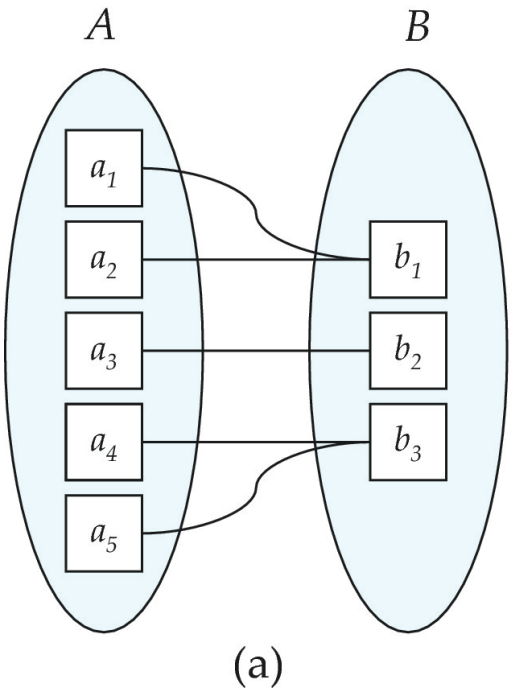
One to one



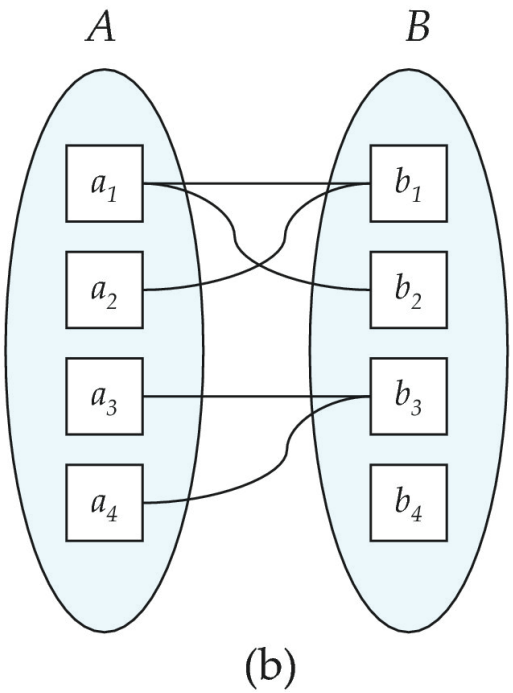
One to many

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

 ©Silberschatz, Korth and Sudarshan

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**.

©Silberschatz, Korth and Sudarshan

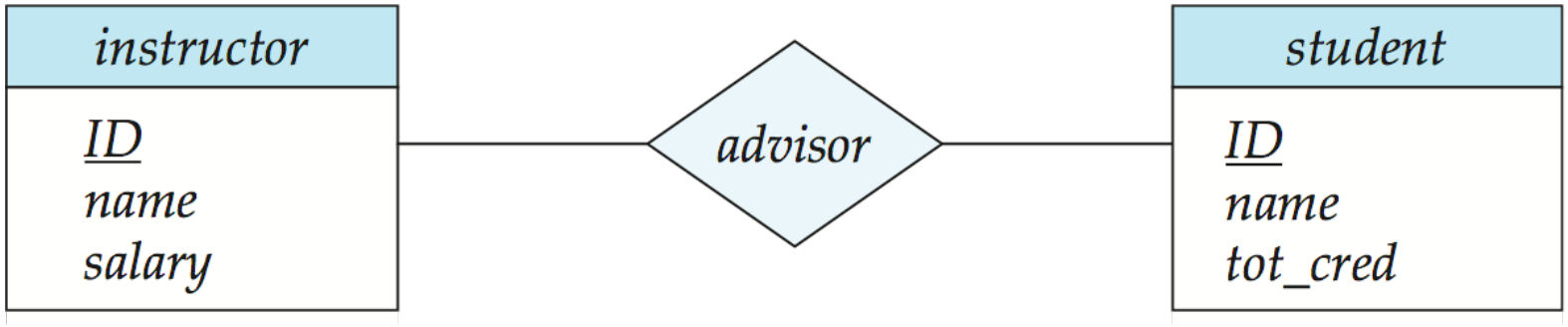
ER Model and external view

ER Model and Diagram

- ER Model:
  - A design tool describing at the high level data and relationships among data
- ER Diagram:
  - A graph describing data and relationships among data

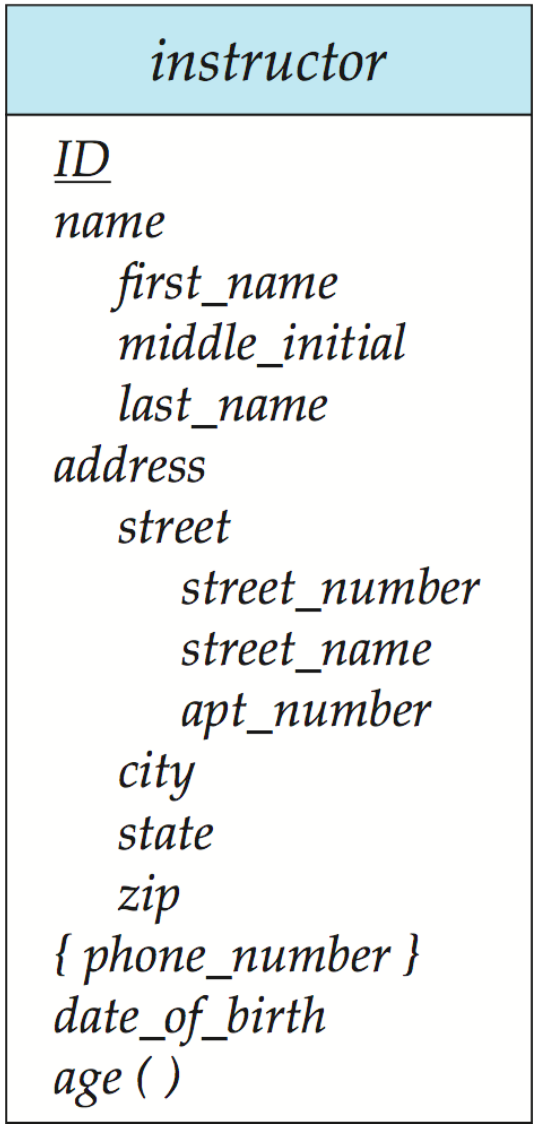
 ©Silberschatz, Korth and Sudarshan

E-R Diagrams



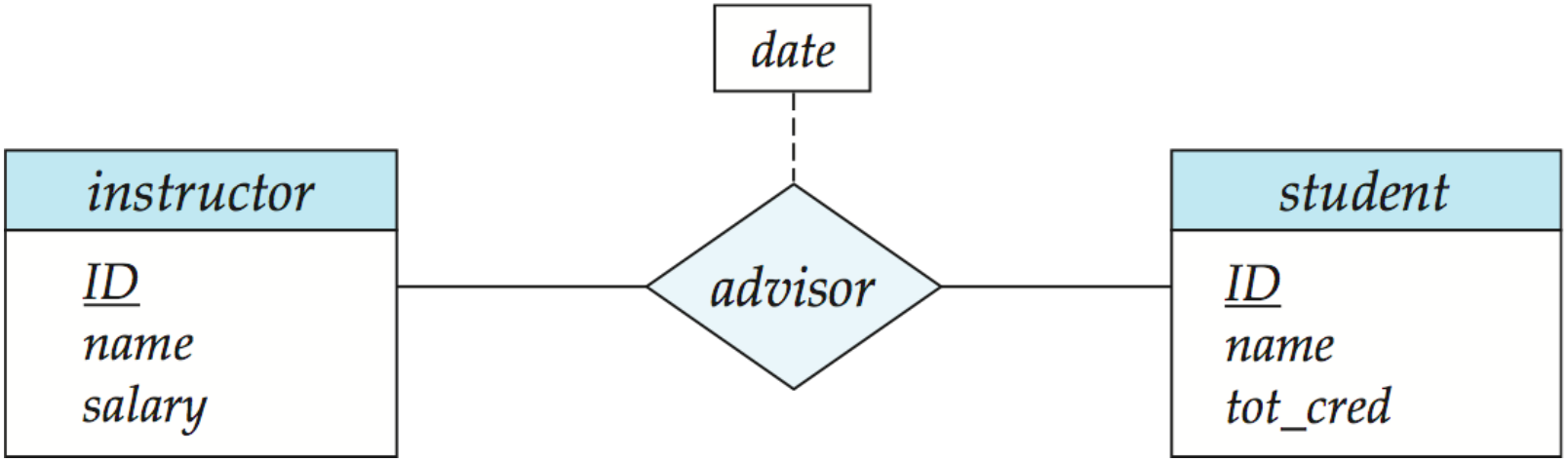
- Rectangles represent entity sets.
- Diamonds represent relationship sets.
- Attributes listed inside entity rectangle
- Underline indicates primary key attributes

Entity With Composite, Multivalued, and Derived Attributes



 ©Silberschatz, Korth and Sudarshan

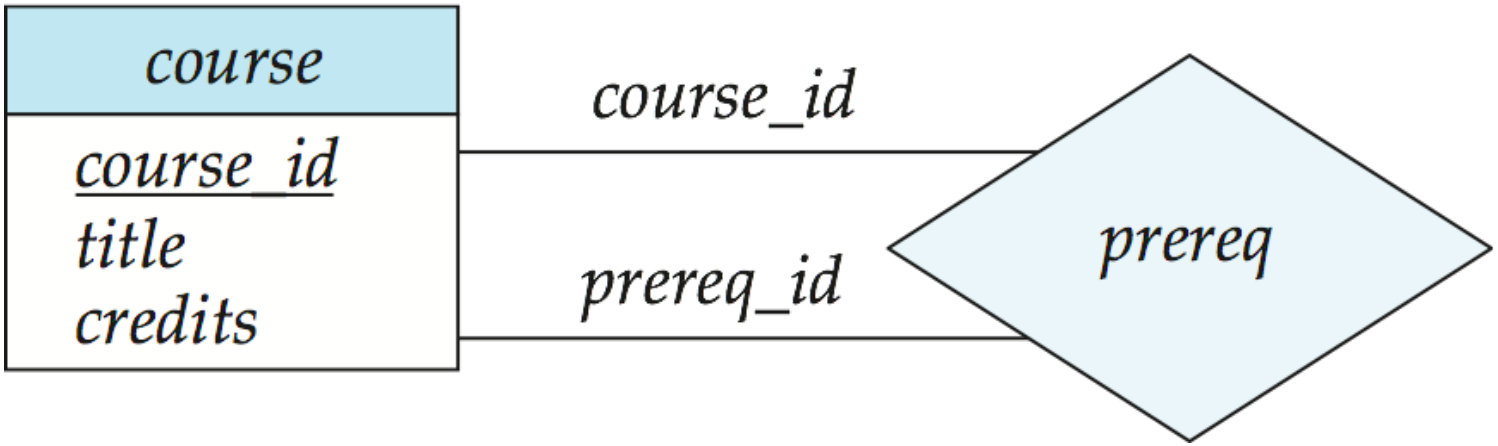
Relationship Sets may have Attributes



 ©Silberschatz, Korth and Sudarshan

Role Indicator (recursive relationship)

- 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**.



Another Example:

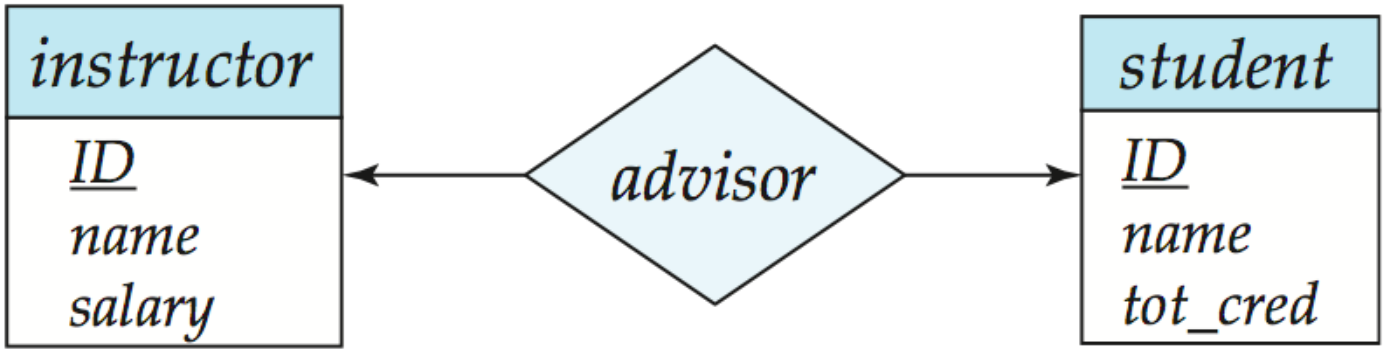
Cardinality Constraints

- 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 *stud_dept*

©Silberschatz, Korth and Sudarshan

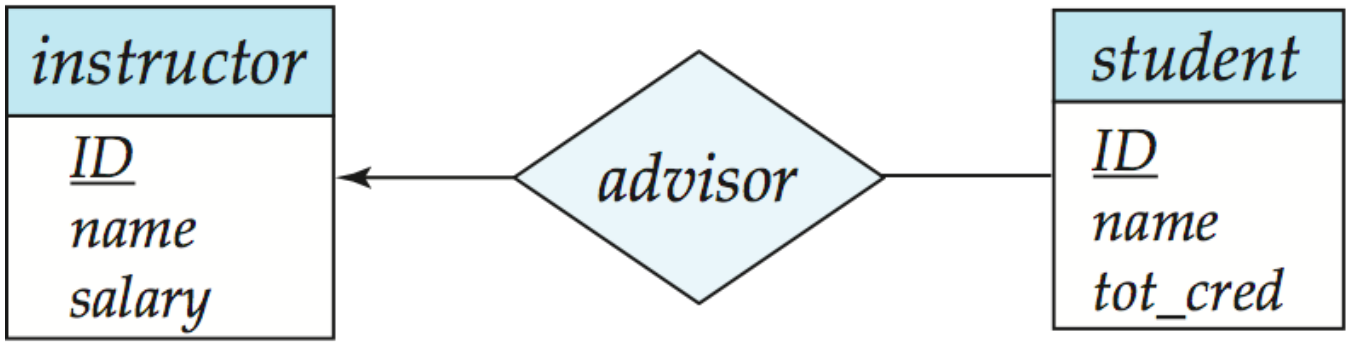
One-to-One Relationship

- 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*



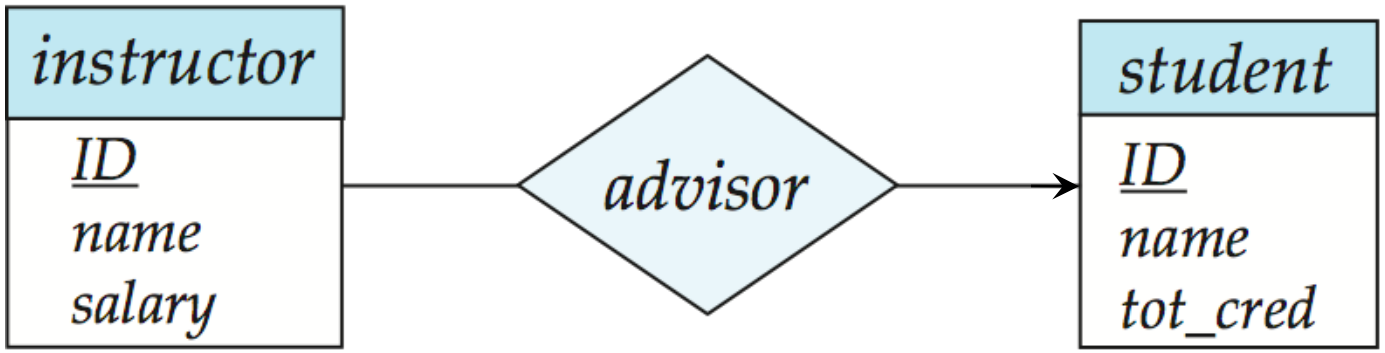
One-to-Many Relationship

- 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,



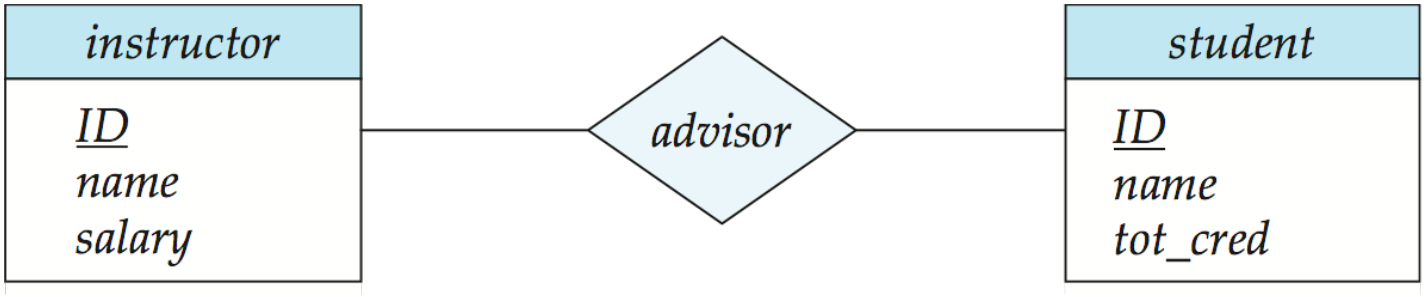
Many-to-One Relationships

- 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

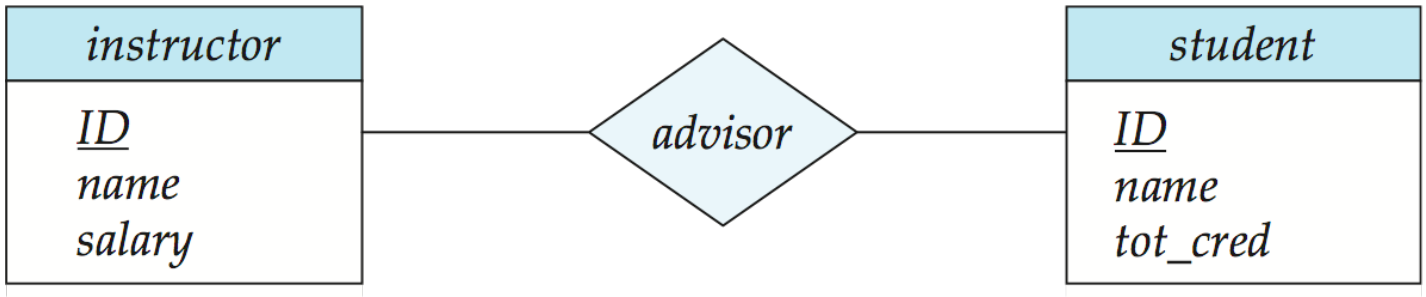
- An instructor is associated with several (possibly 0) students via *advisor*
- A student is associated with several (possibly 0) instructors via *advisor*



 ©Silberschatz, Korth and Sudarshan

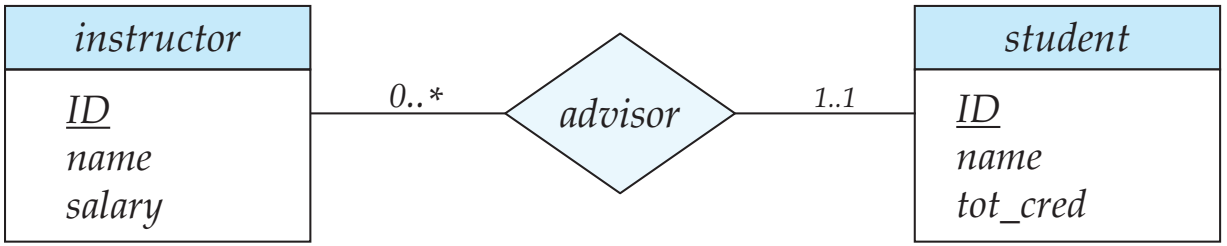
Participation of an Entity Set in a Relationship Set

- Total participation (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set
- Partial participation: some entities may not participate in any relationship in the relationship set
  - Example: modify the following ER diagram to show that every student must have an advisor



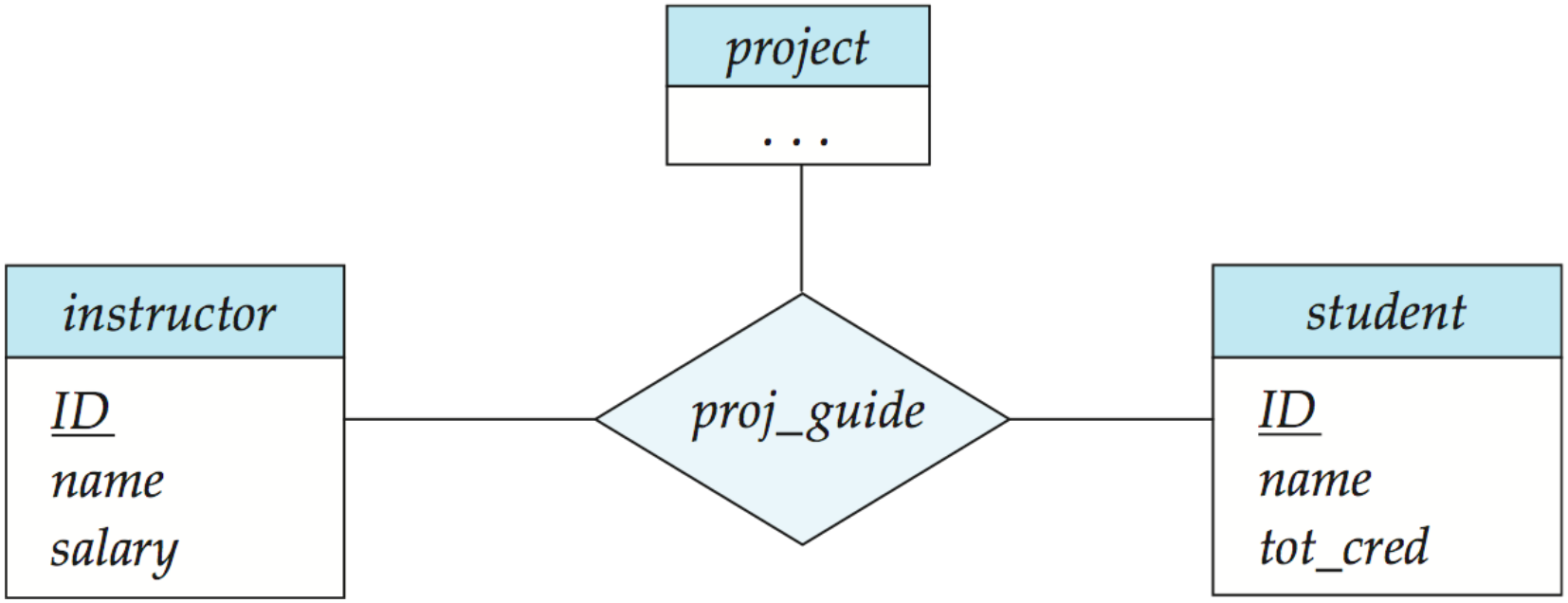
Notation for Expressing More Complex Constraints

- 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.
  - A maximum value of 1 indicates that the entity participates in at most one relationship
  - A maximum value of * indicates no limit.



Instructor can advise 0 or more students. A student must have 1 advisor; cannot have multiple advisors

E-R Diagram with a Ternary Relationship



Another example: professors teach classes at universities

Cardinality Constraints on Ternary Relationship

- We allow at most one arrow out of a ternary (or greater degree) relationship to indicate a cardinality constraint
- E.g., an arrow from *proj_guide* to *instructor* indicates each student has at most one guide for a project
- If there is more than one arrow, there are two ways of defining the meaning.
  - E.g., a ternary relationship *R* between *A*, *B* and *C* with arrows to *B* and *C* could mean

    1. each *A* entity is associated with a unique entity from *B* and *C* or

    2. each pair of entities from (*A, B*) is associated with a unique *C* entity, and each pair (*A, C*) is associated with a unique *B*
  - Each alternative has been used in different formalisms
  - To avoid confusion we avoid using more than one arrow

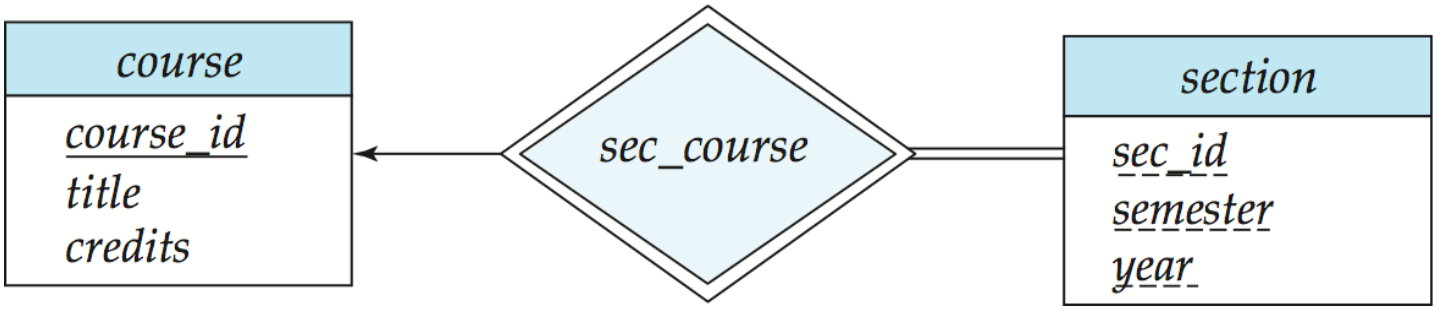
©Silberschatz, Korth and Sudarshan

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.

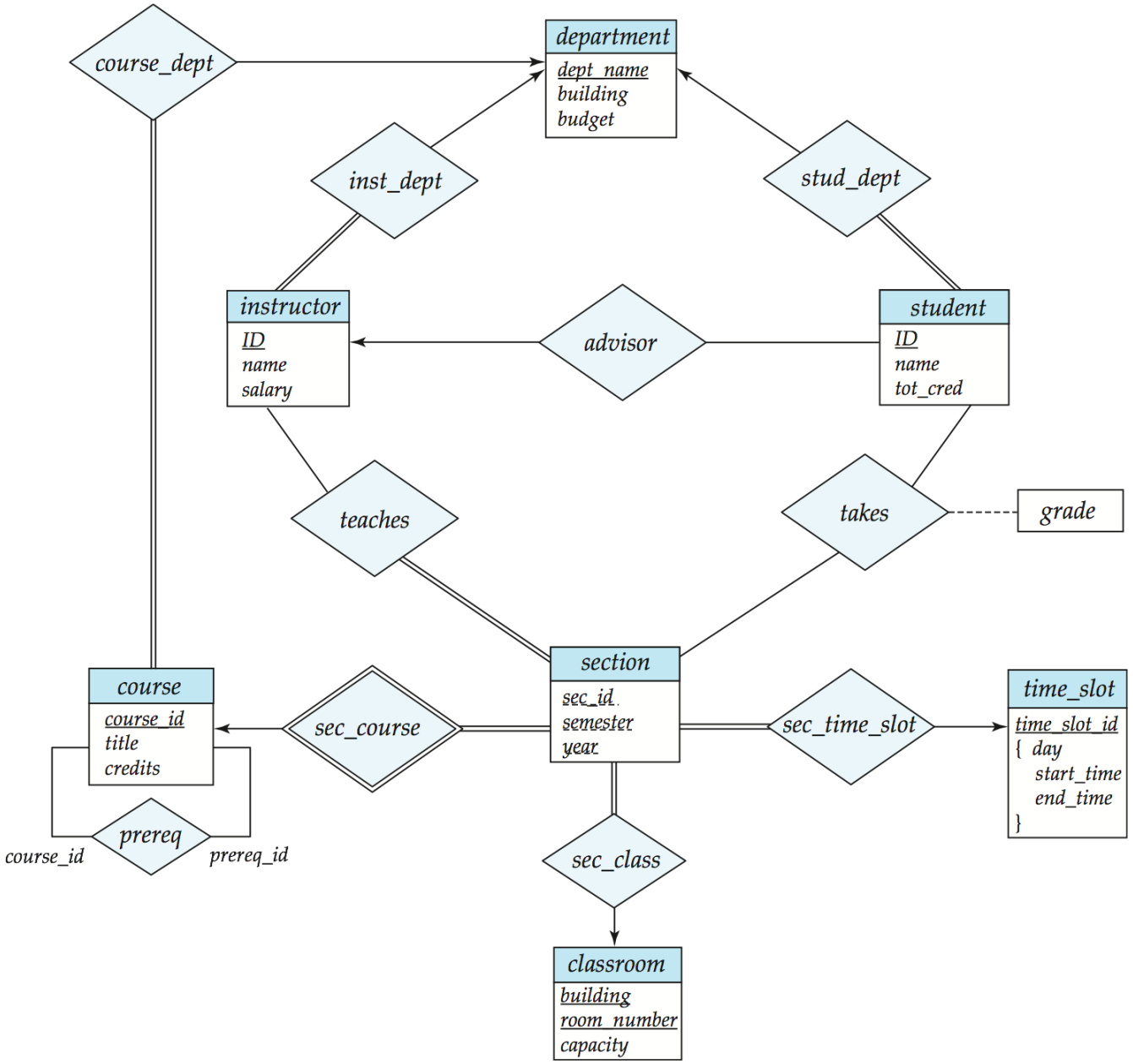
Weak Entity Sets (Cont.)

- 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 a weak entity set: combined discriminator and primary key of strong entity set on which the weak entity dependents.
  - Example: primary key for *section* is
    - (*course_id, sec_id, semester, year*)



 ©Silberschatz, Korth and Sudarshan

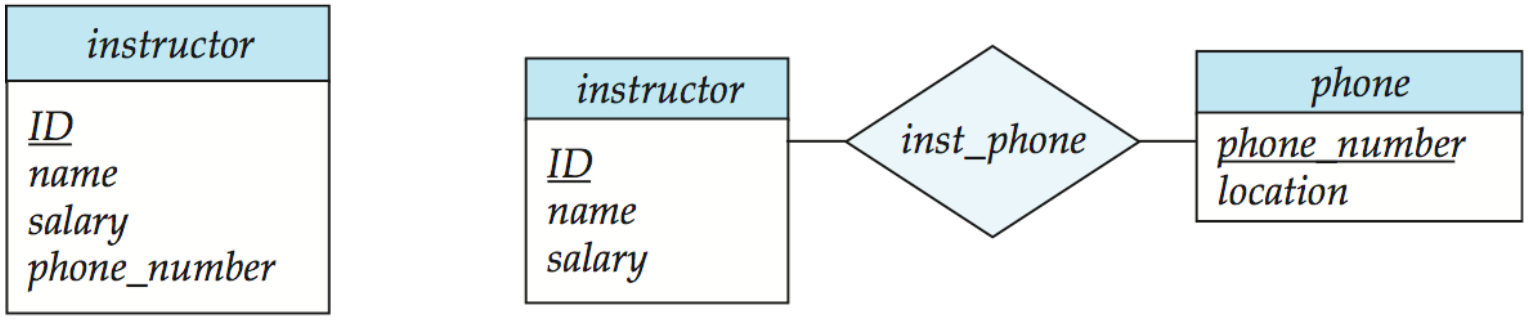
E-R Diagram for a University Enterprise



©Silberschatz, Korth and Sudarshan

Design Issues

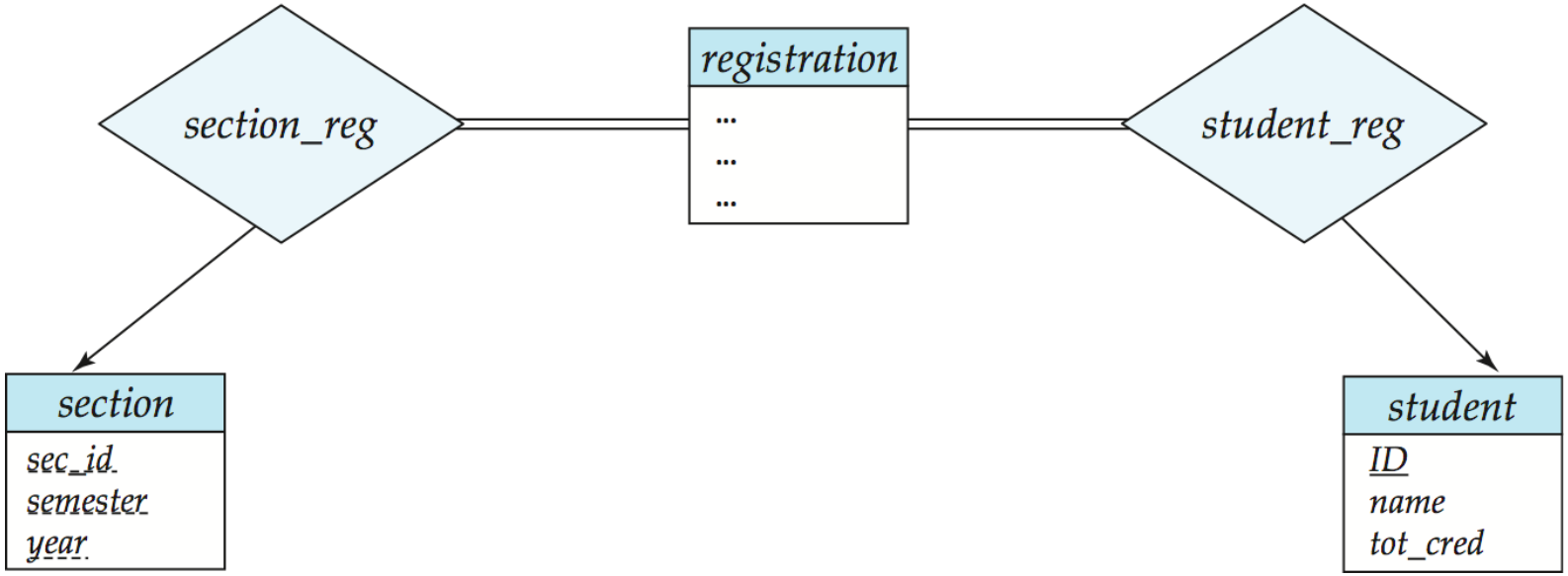
- **Use of entity sets vs. attributes**



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

Design Issues

- **Use of entity sets vs. relationship sets**
  Possible guideline is to designate a relationship set to describe an action that occurs between entities



Design Issue: Redundant Attributes

- Suppose we have entity sets
  - *instructor*, with attributes including *dept_name*
  - *department*

  and a relationship
  - *inst_dept* relating *instructor* and *department*
- Attribute *dept_name* in entity *instructor* is redundant since there is an explicit relationship *inst_dept* which relates instructors to departments
  - The attribute replicates information present in the relationship, and should be removed from *instructor*

©Silberschatz, Korth and Sudarshan

Design Issues

- **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?*

©Silberschatz, Korth and Sudarshan

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 known)
  - But there are some relationships that are naturally non-binary
    - Example: *proj_guide*

©Silberschatz, Korth and Sudarshan

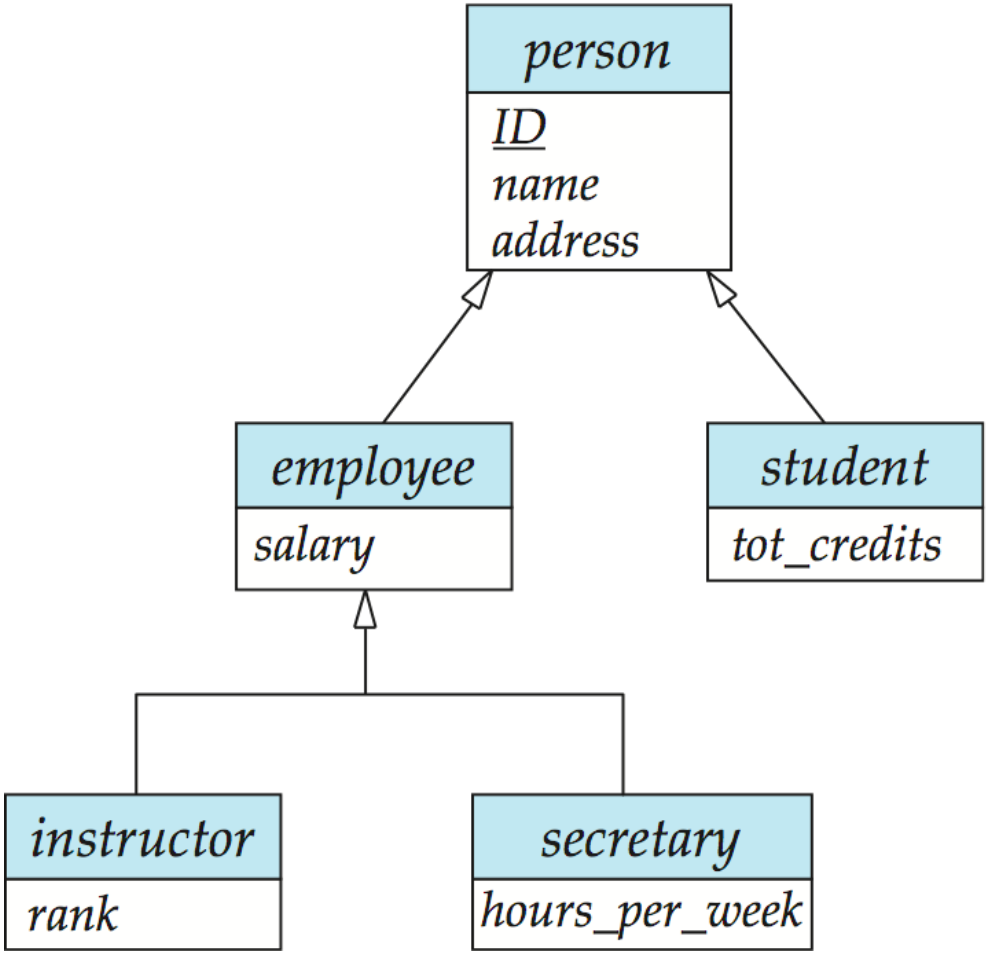
Extended ER Features

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.

©Silberschatz, Korth and Sudarshan

Specialization Example



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.

©Silberschatz, Korth and Sudarshan

Specialization and Generalization (Cont.)

- 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

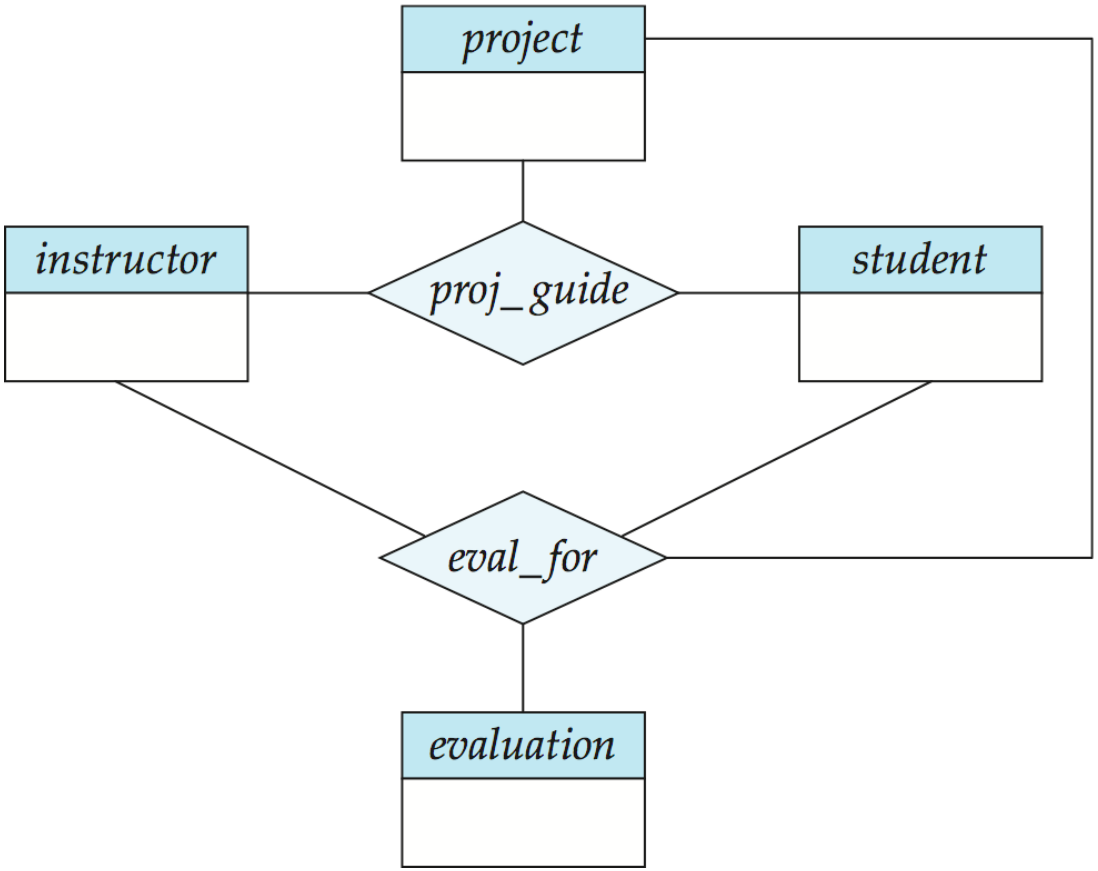
Design Constraints on a Specialization/Generalization

- Constraint on whether or not entities may belong to more than one lower-level entity set within a single generalization.
  - **Disjoint**
    - an entity can belong to only one lower-level entity set
    - Noted in E-R diagram by having multiple lower-level entity sets link to the same higher-level entity set via ONE ARROW
  - **Overlapping**
    - an entity can belong to more than one lower-level entity set

©Silberschatz, Korth and Sudarshan

Aggregation

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



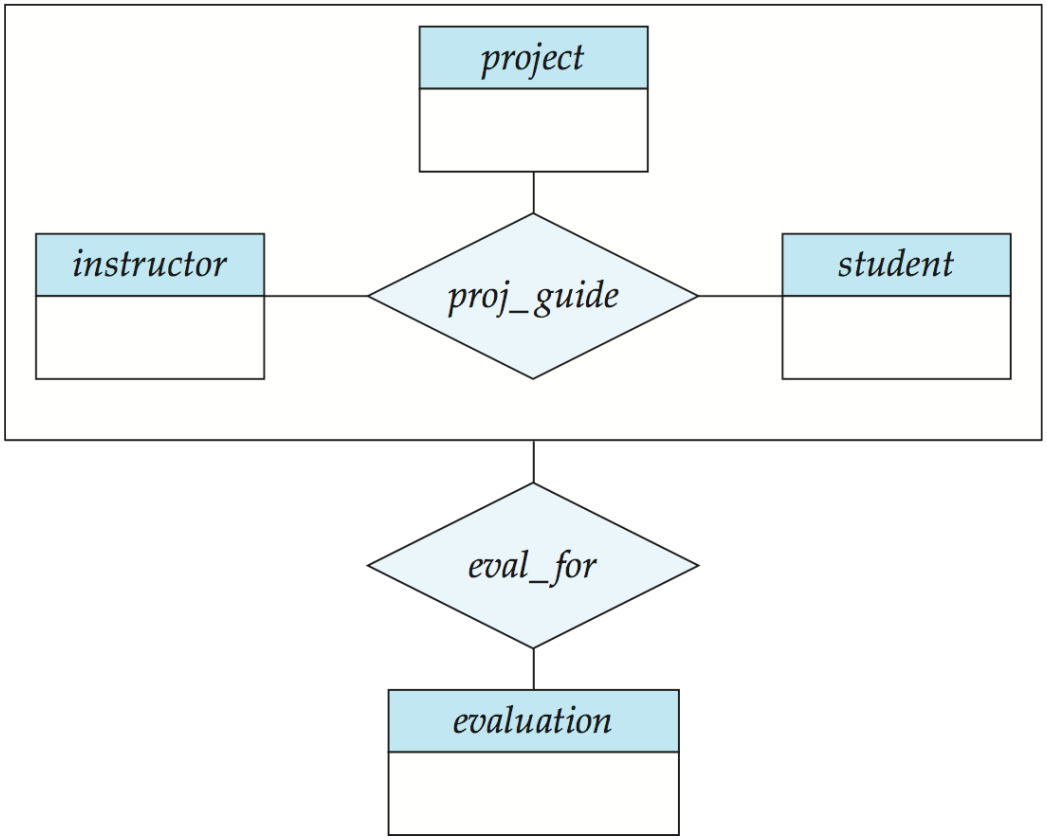
 ©Silberschatz, Korth and Sudarshan

Aggregation (Cont.)

- Relationship sets *eval_for* and *proj_guide* represent overlapping information
  - Every *eval_for* relationship corresponds to a *proj_guide* relationship
  - However, some *proj_guide* relationships may not correspond to any *eval_for* relationships
    - So we can't discard the *proj_guide* relationship
- Eliminate this redundancy via *aggregation*
  - Treat relationship as an abstract entity
  - Allows relationships between relationships
  - Abstraction of relationship into new entity

Aggregation (Cont.)

- Without introducing redundancy, the following diagram represents:
  - A student is guided by a particular instructor on a particular project
  - A student, instructor, project combination may have an associated evaluation

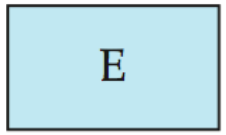


 ©Silberschatz, Korth and Sudarshan

E-R Design Decisions

- 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.
- The use of aggregation – can treat the aggregate entity set as a single unit without concern for the details of its internal structure.

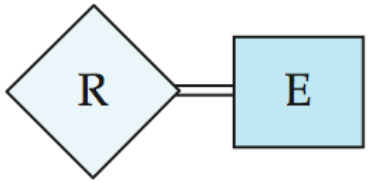
Summary of Symbols Used in E-R Notation



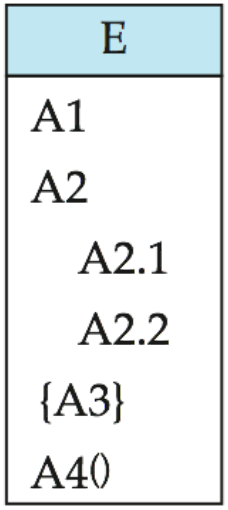
entity set

relationship set

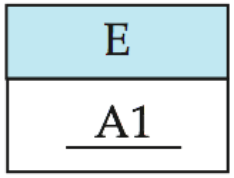
identifying relationship set for weak entity set



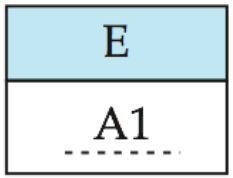
total participation of entity set in relationship



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

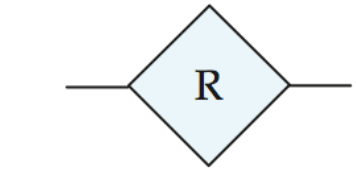


primary key

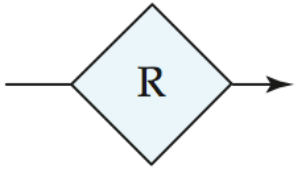


discriminating attribute of weak entity set

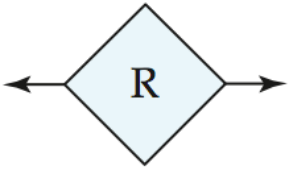
Symbols Used in E-R Notation (Cont.)



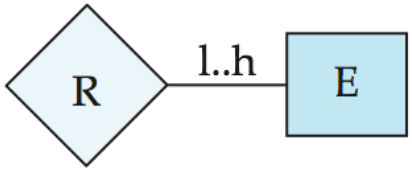
many-to-many relationship



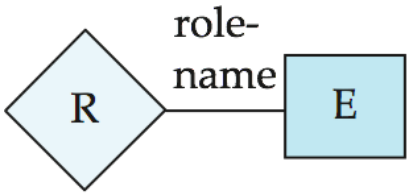
many-to-one relationship



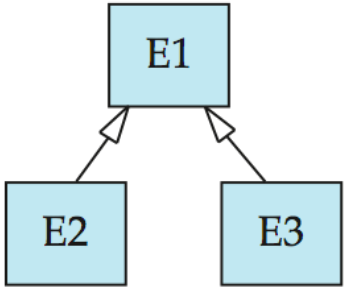
one-to-one relationship



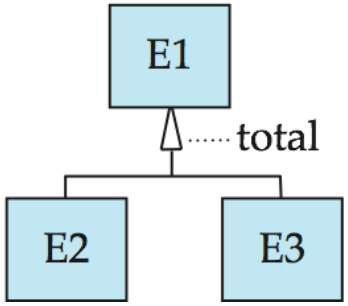
cardinality limits



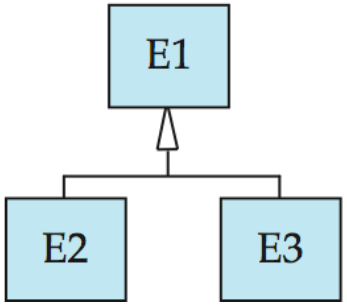
role indicator



ISA: generalization or specialization



total (disjoint) generalization

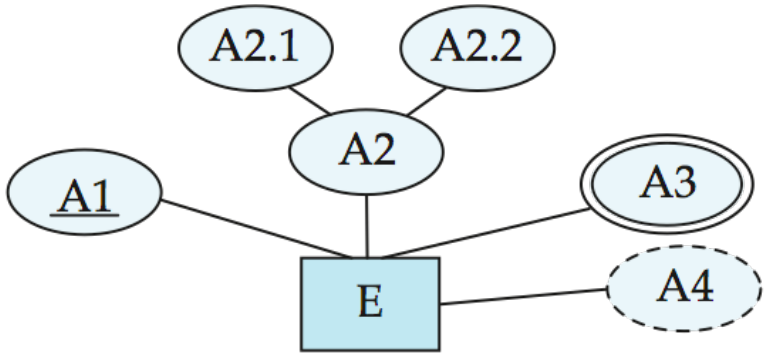


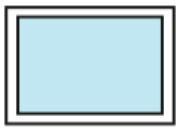
disjoint generalization

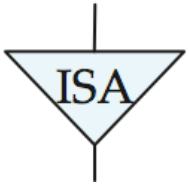
Alternative ER Notations

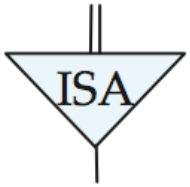
- Chen, IDE1FX, …

entity set E with
simple attribute A1,
composite attribute A2,
multivalued attribute A3,
derived attribute A4,
and primary key A1

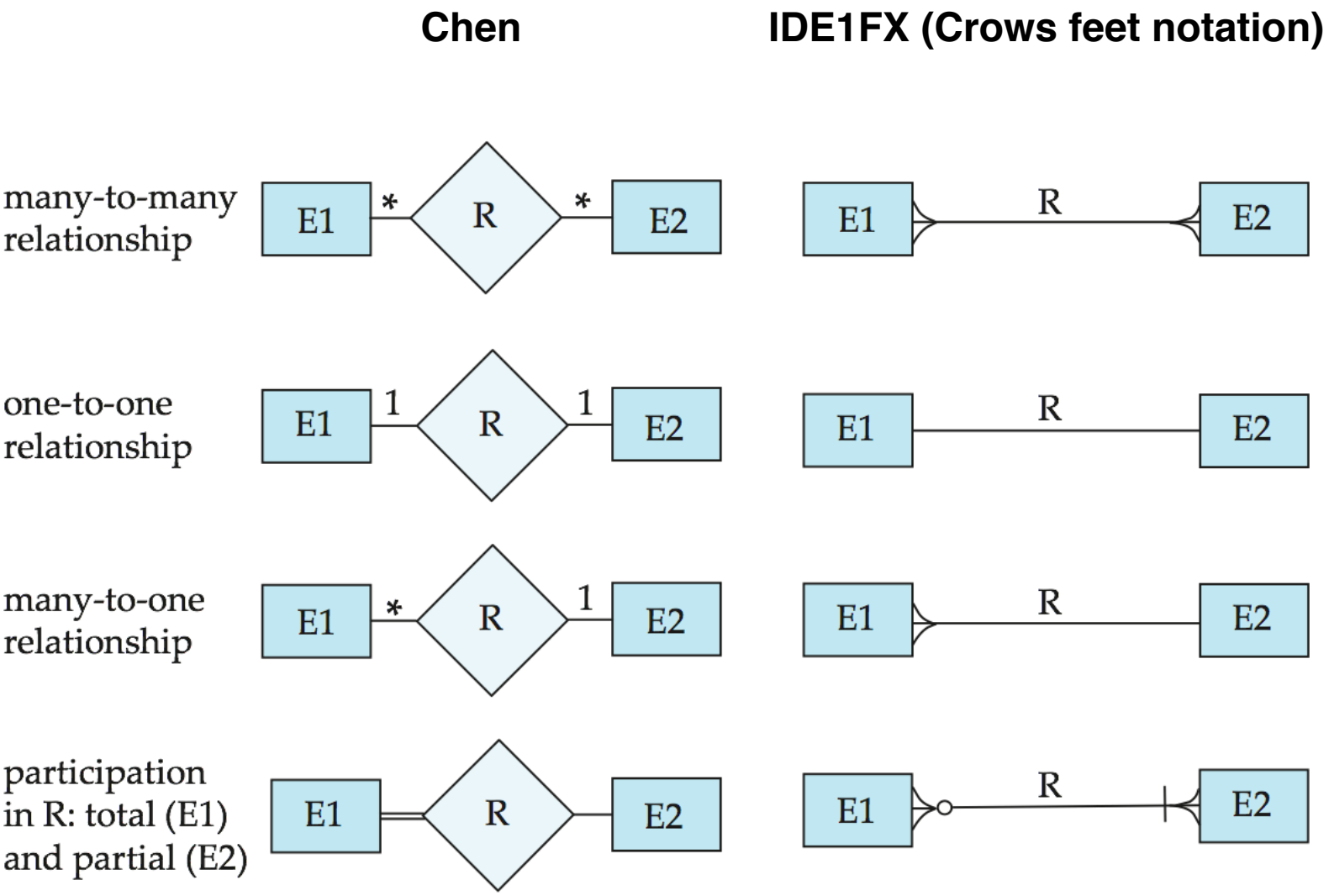


weak entity set 

generalization 

total generalization 

Alternative ER Notations



 ©Silberschatz, Korth and Sudarshan

The picture can't be displayed.

ER Database Examples

Database System Concepts

©Silberschatz, Korth and Sudarshan

(Modified for CS 4513)

- Construct an ER diagram that represents the database in each of the following problems; underline all primary keys; no additional attributes are allowed.

Problem 1: A Company Database

- A company is organized into departments. Each departments has a name and a number, and an employee who manages the department. A department controls a number of projects, each of which has a name, number and location. Each employee has a name, social security number, and address. Each employee is assigned to one department but may work on several projects. The database also keeps track of the number of hours per week an employee works on a project, and dependents of employees. Each dependent has a name and relationship.

Solution for Problem 1

 ©Silberschatz, Korth and Sudarshan

Problem 2: a Hospital Database

- A hospital is made up of persons each of whom is either a doctor, patient, or a staff member. Every person has a unique id, a name and an age. For each doctor, the hospital keeps track of the specialty and school where he/she obtained his/her Medical Doctor degree. For each patient, his/her symptom is also recorded. Each staff member has a rank and salary. A patient can be treated by several doctors. The description, time and date of each treatment that a doctor gives to a patient is also recorded. A patient can be either an inpatient or an outpatient. Each outpatient has an address and each inpatient has a date he/she was admitted to the hospital. Each inpatient is assigned to a particular award on a particular date. Each ward is identified by its ward id, is managed by a doctor, has a description and is associated with a number of rooms. A room in each ward has its own room number, type and cost per day. Some staff members are used to support doctors and patients in emergency cases. The hospital keeps track of the number of hours each of these members spent on an emergency case.

Solution for Problem 2

Homework 1

- Questions are on the class website;
- Not to be turned in;
- Answers will be posted on the class website later.

The picture can't be displayed.

End of Topic 2

Database System Concepts

©Silberschatz, Korth and Sudarshan

(Modified for CS 4513)