Database Systems Concepts and Design

CSC201S2/G2

Database Design Using the E-R Model

Outline

- The Entity-Relationship Model
- Complex Attributes
- Mapping Cardinalities
- Primary Key
- Removing Redundant Attributes in Entity Sets
- Reducing ER Diagrams to Relational Schemas
- Extended E-R Features
- Entity-Relationship Design Issues
- Alternative Notations for Modeling Data

Overview

Entity Relationship Model

- Models an enterprise as a collection of *entities* and *relationships*
  - Entity: a “thing” or “object” in the enterprise that is distinguishable from other objects. Described by a set of *attributes*
  - Relationship: an association among several entities
- Represented diagrammatically by an *entity-relationship diagram*

▪Normalization Theory

- Formalize what designs are bad, and test for them

ER model: Database Modeling

- Facilitate database design by allowing specification of an enterprise schema
- Represents the overall logical structure of a database.
- The ER data model employs three basic concepts:

  entity sets, relationship sets, and attributes.
- Express the overall logical structure of a database graphically

Entity Sets

- An entity: object that exists and is distinguishable from other objects.

  Eg: specific person, company, event, plant

- An entity set: set of entities of the same type that share the same properties.

  Eg: set of all persons, companies, trees, holidays

- An entity is represented by a set of attributes

  Eg: instructor = (ID, name, salary), course= (course_id, title, credits)

- A subset of the attributes form a primary key: uniquely identifying each member of the set.

Entity Sets: Instructor and Student

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

instructor

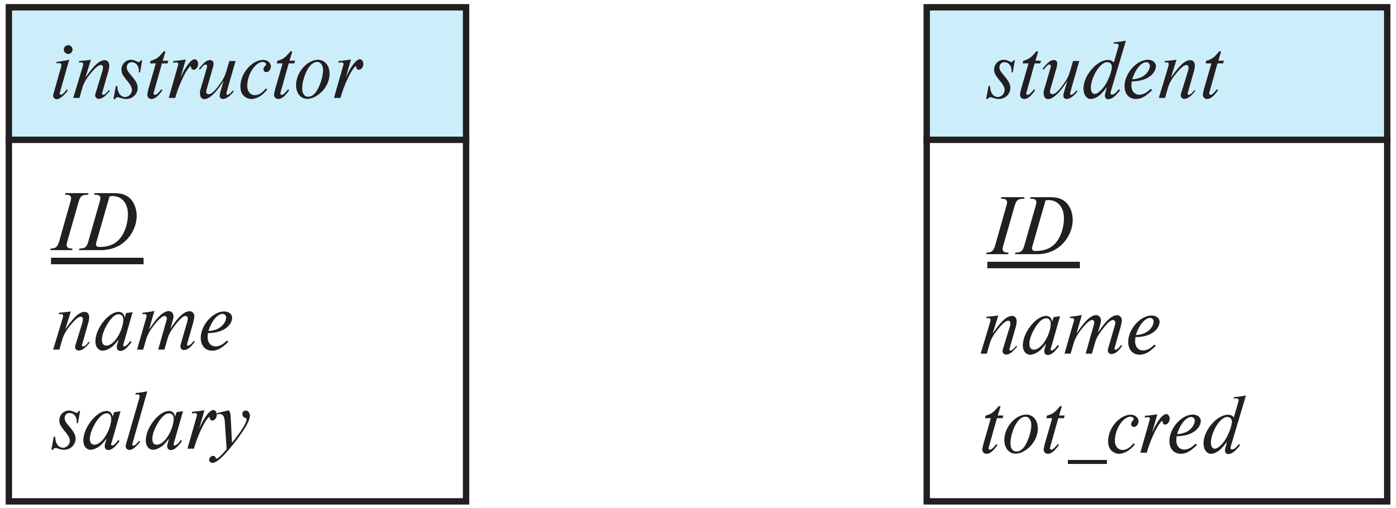
98988	Tanaka
12345	Shankar
00128	Zhang
76543	Brown
76653	Aoi
23121	Chavez
44553	Peltier

student

Representing Entity sets in ER Diagram

Entity sets can be represented graphically as follows:

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



Relationship Sets

Eg: 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

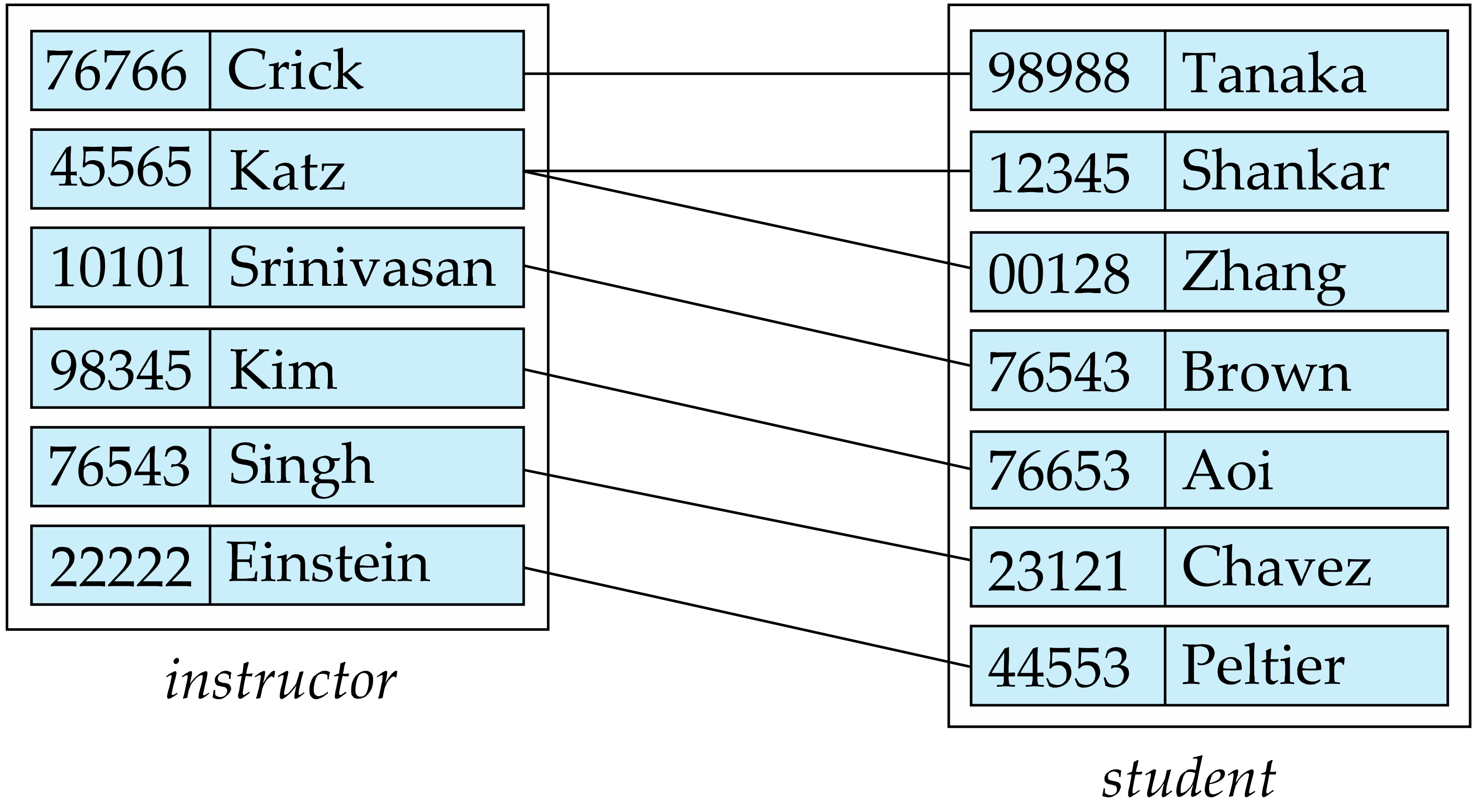
$$\{(e_1, e_2, \ldots e_n) \mid e_1 \in E_1, e_2 \in E_2, \ldots, e_n \in E_n\}$$

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

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

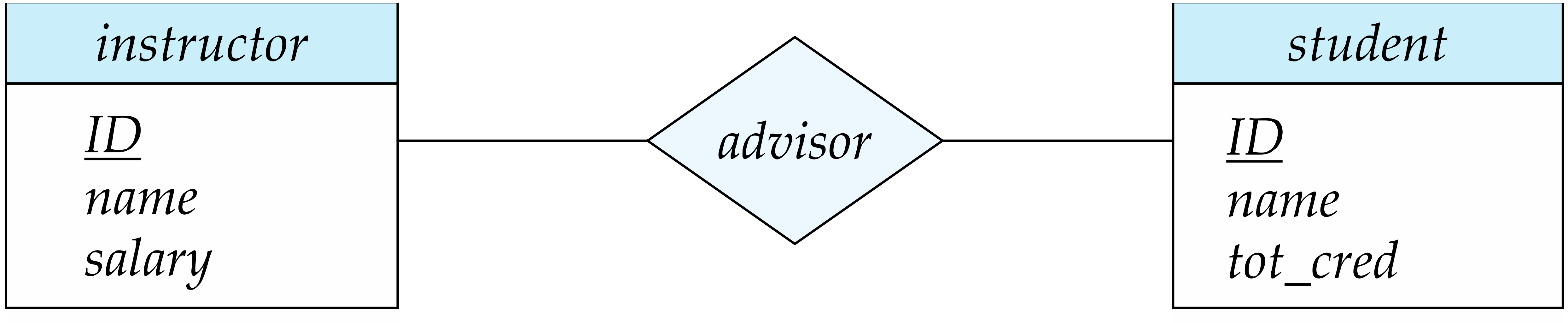
Relationship Sets

- Example: we define the relationship set *advisor* to denote the associations between students and the instructors who act as their advisors.



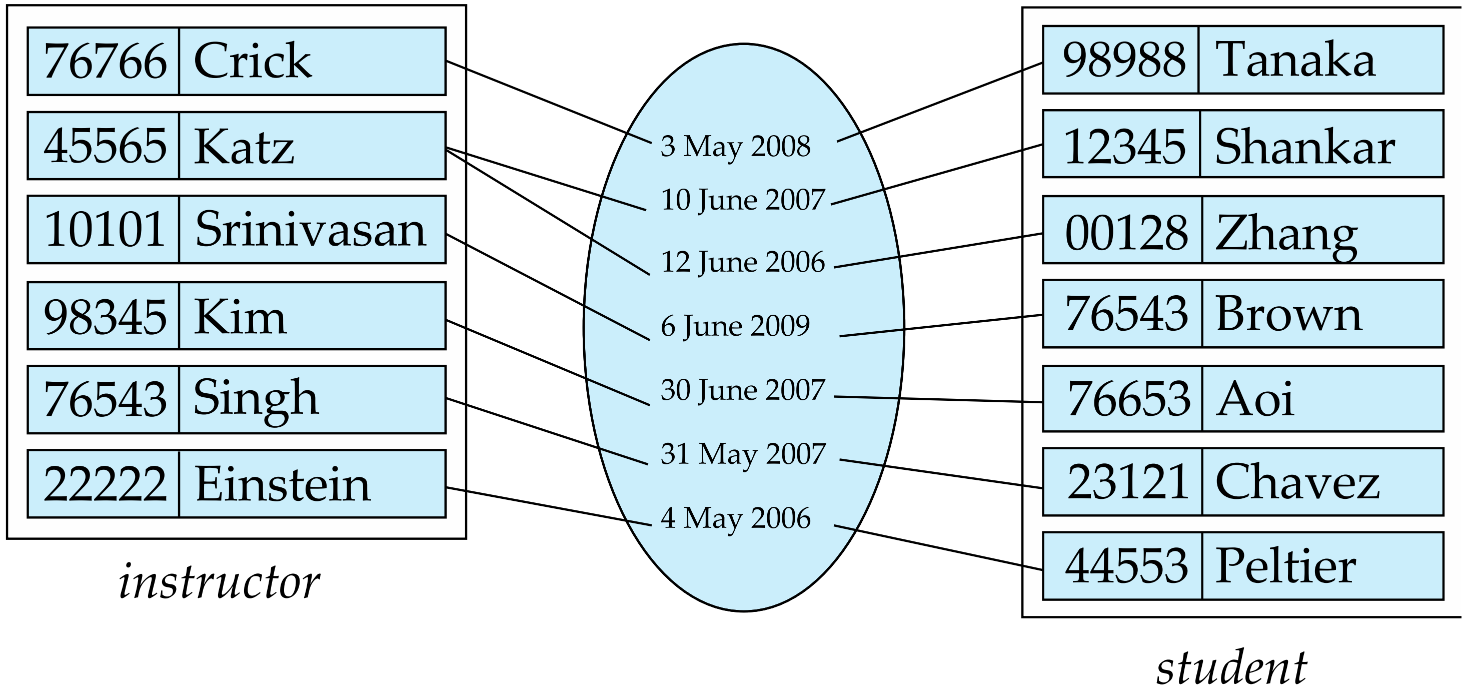
Representing Relationship Sets via ER Diagrams

- Diamonds represent relationship sets.



Relationship Sets

- An attribute can also be associated with a relationship set.
- Eg: 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

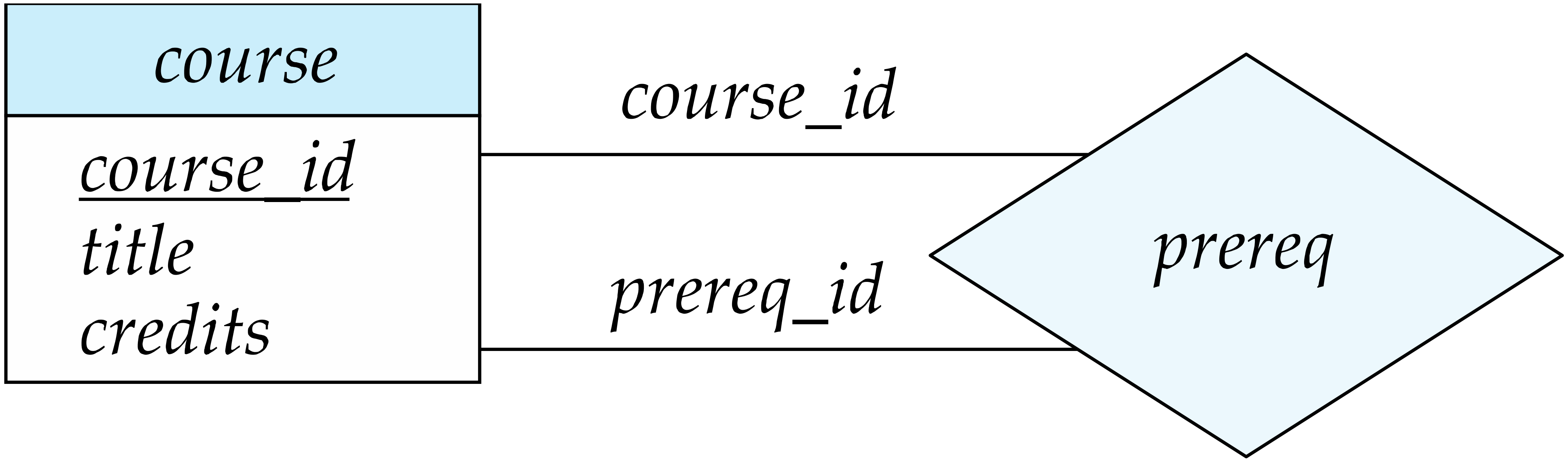


Roles

- Entity sets of a relationship need not be distinct

  Each occurrence of an entity set plays a “role” in the relationship

Eg: The labels “*course_id*” and “*prereq_id*” are called **roles**.



Q
&
A

Degree of Relationship Set

Binary relationship

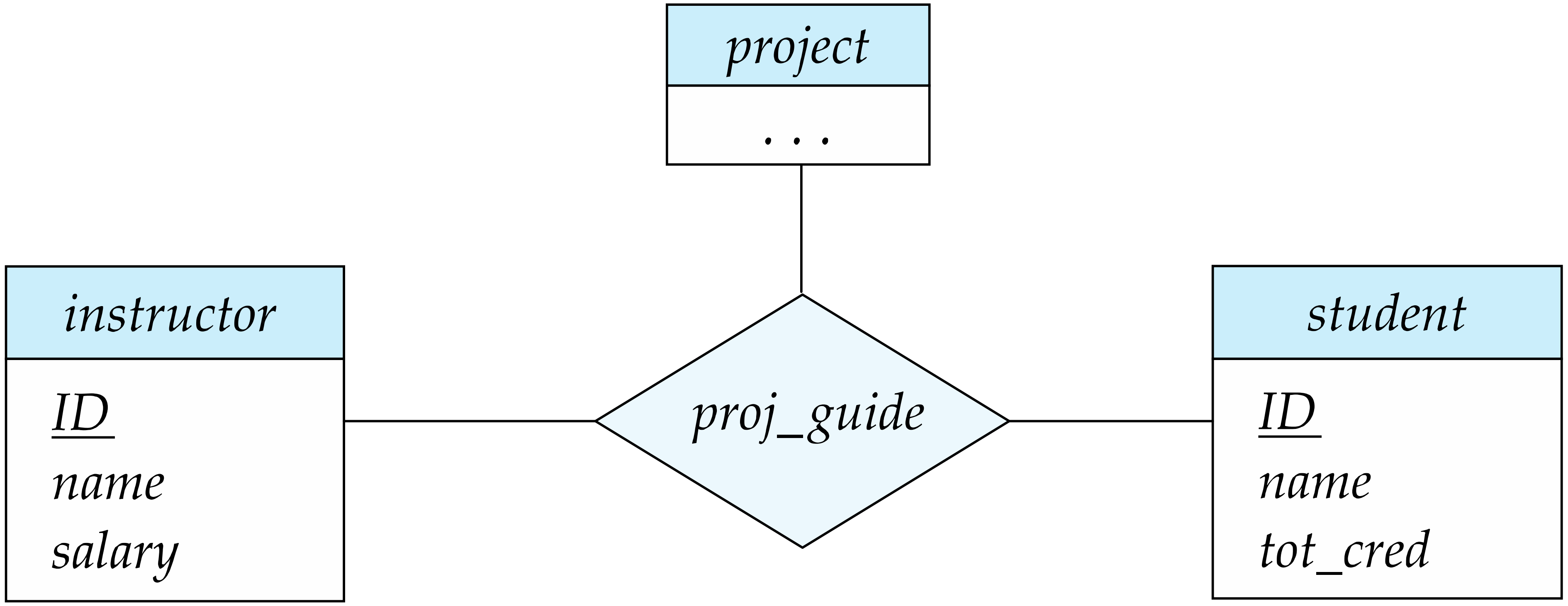
- involve two entity sets (or degree two).
- most relationship sets in a database system are binary.

Example: students work on research projects under the guidance of an instructor.

- relationship *proj_guide* is a ternary relationship between instructor, student, and project

Non-binary Relationship Set

E-R Diagram with a Ternary Relationship



Complex Attributes

- Simple and composite attributes

  Single-valued and multivalued attributes

  Example: multivalued attribute: phone_numbers

- Derived attributes

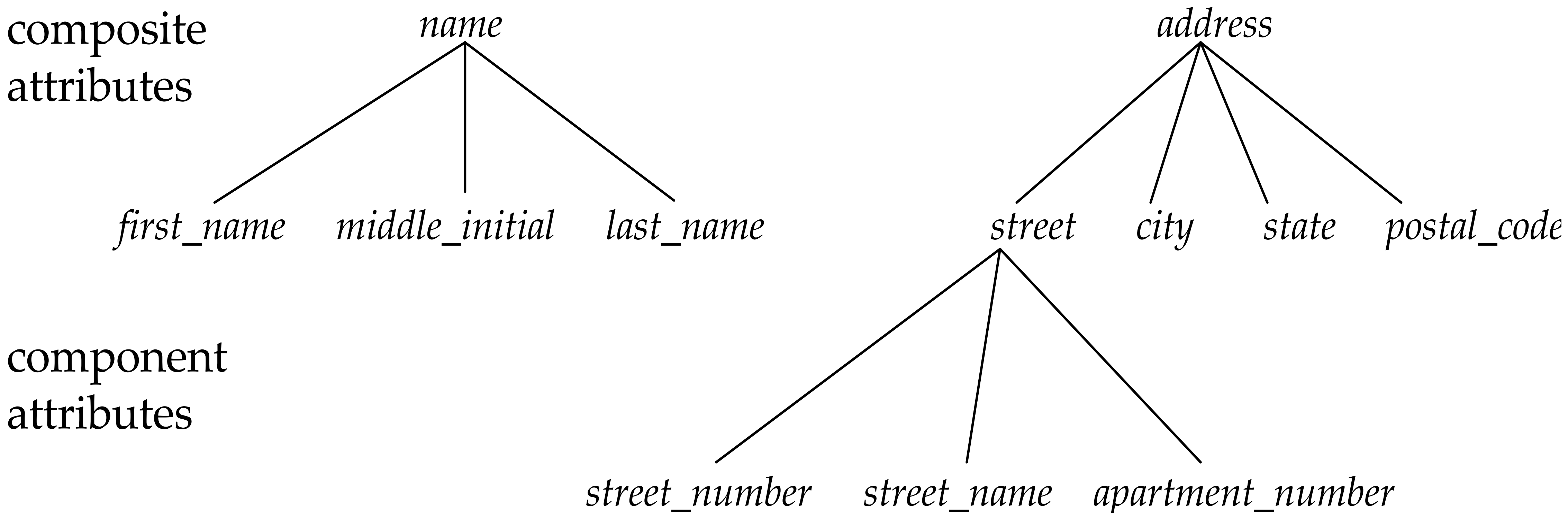
  Can be computed from other attributes

  Example: age, given date_of_birth

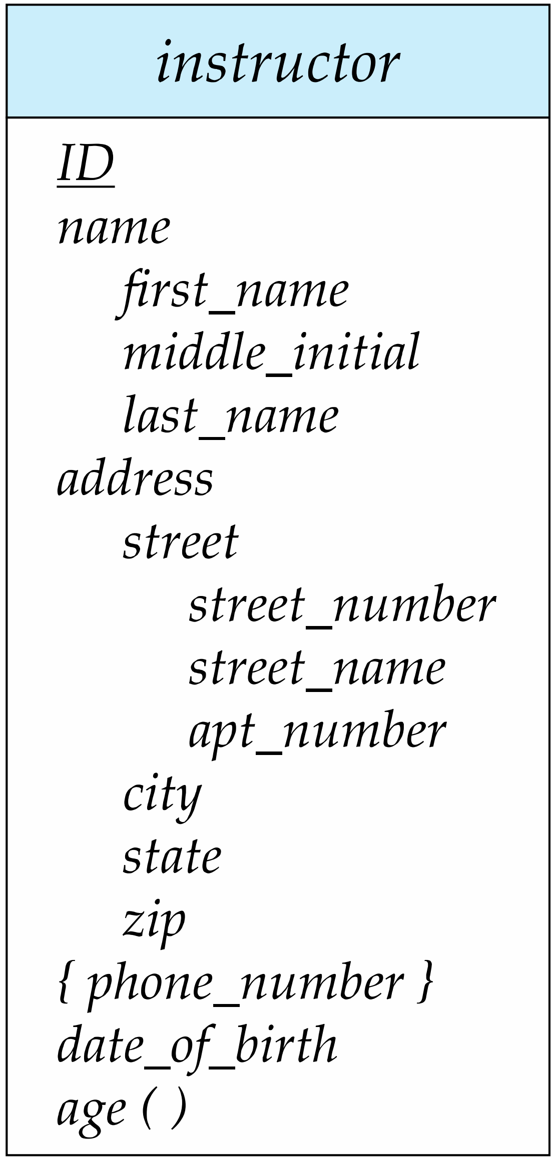
- Domain – the set of permitted values for each attribute

Composite Attributes

- Composite attributes allow us to divided attributes into subparts (other attributes).



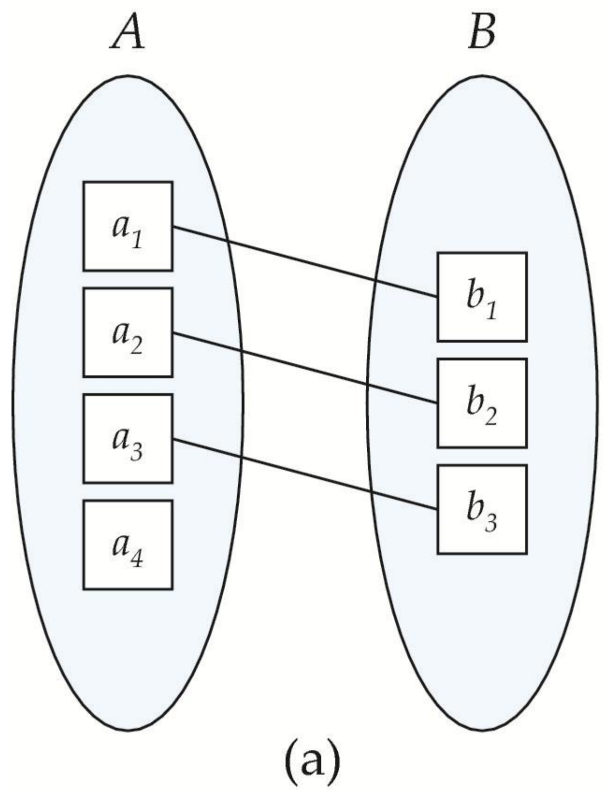
Representing complex attributes in ER-Diagram



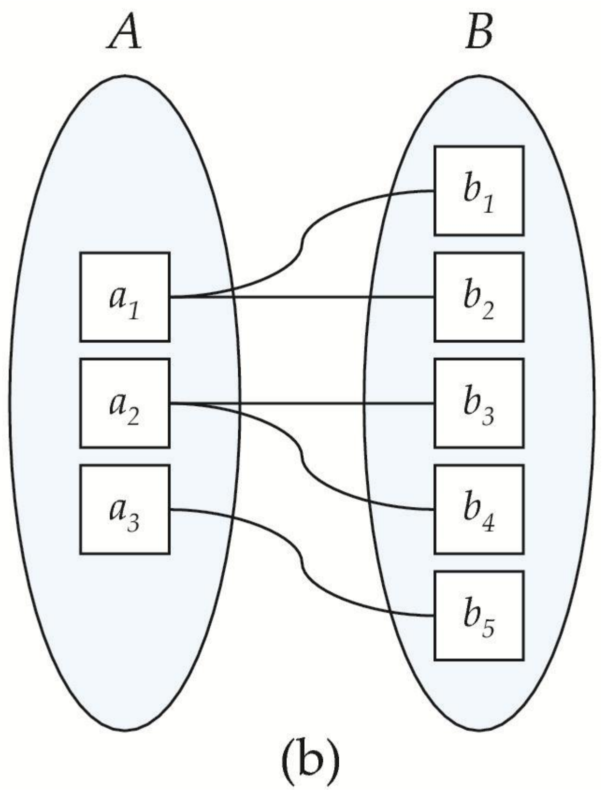
Mapping Cardinality Constraints

- 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 types are:
  - One to one
  - One to many
  - Many to one
  - Many to many

Mapping Cardinalities



One to one



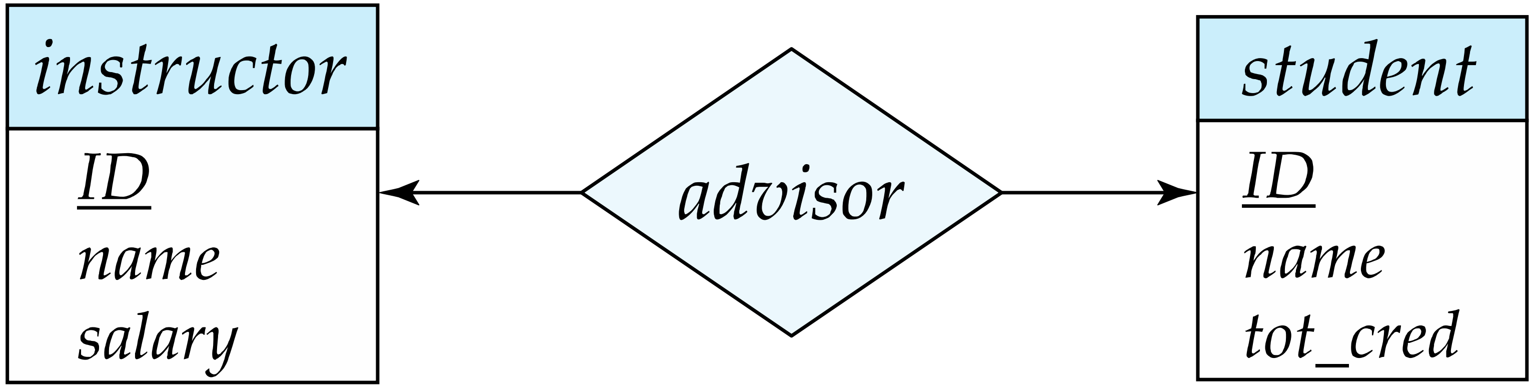
One to many

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

Representing Cardinality Constraints in ER-Diagram

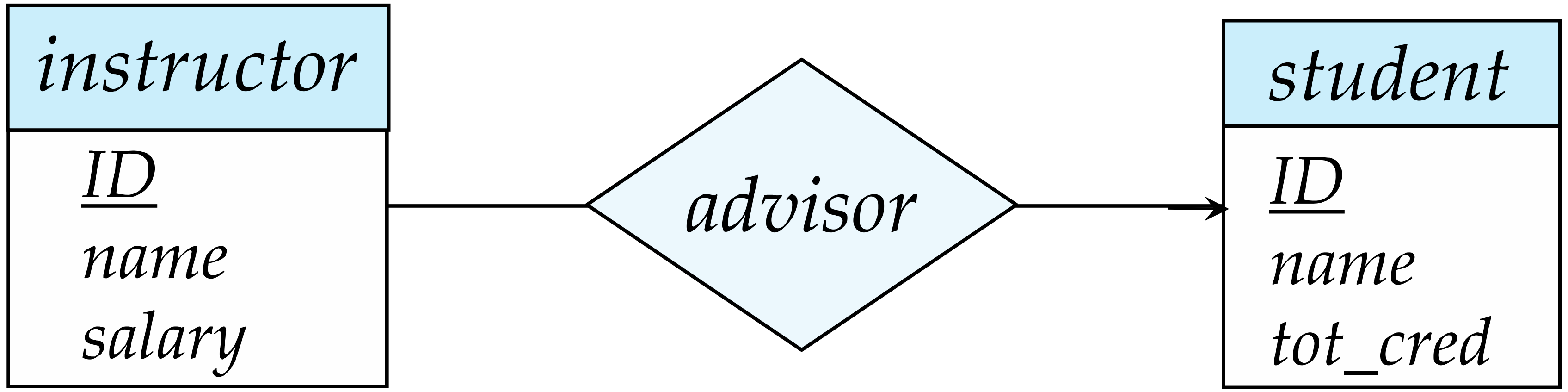
- Express cardinality constraints by drawing either a directed line (→), signifying "one," or an undirected line (—), signifying "many," between the relationship set and the entity set.

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



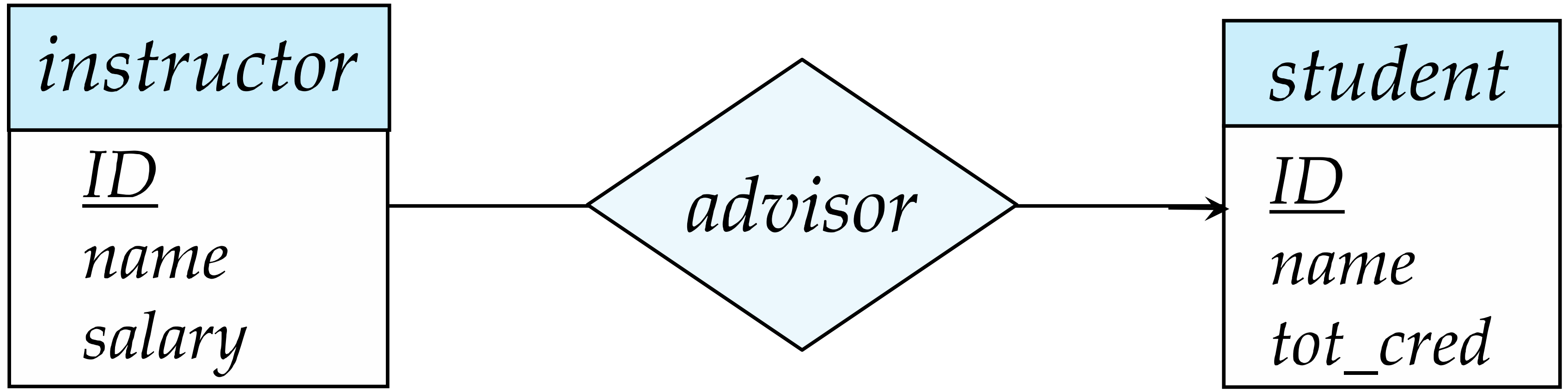
One-to-Many Relationship

- 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*



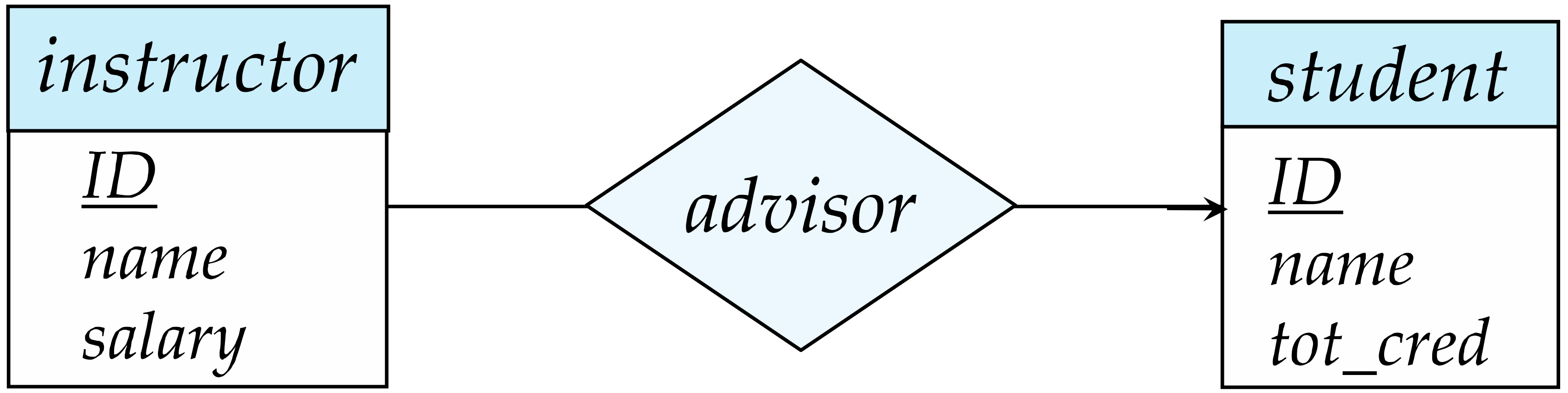
Many-to-One Relationship

- 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 Relations

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



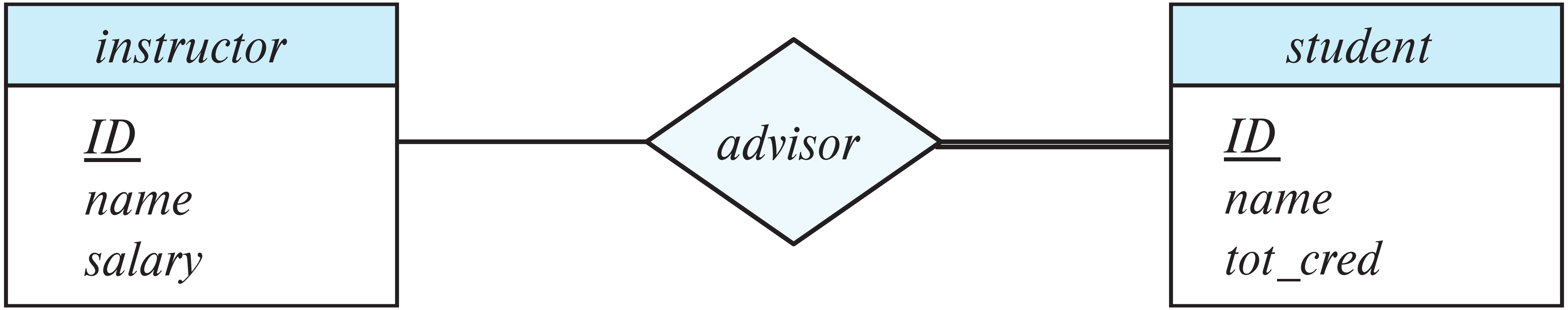
Total and Partial Participation

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

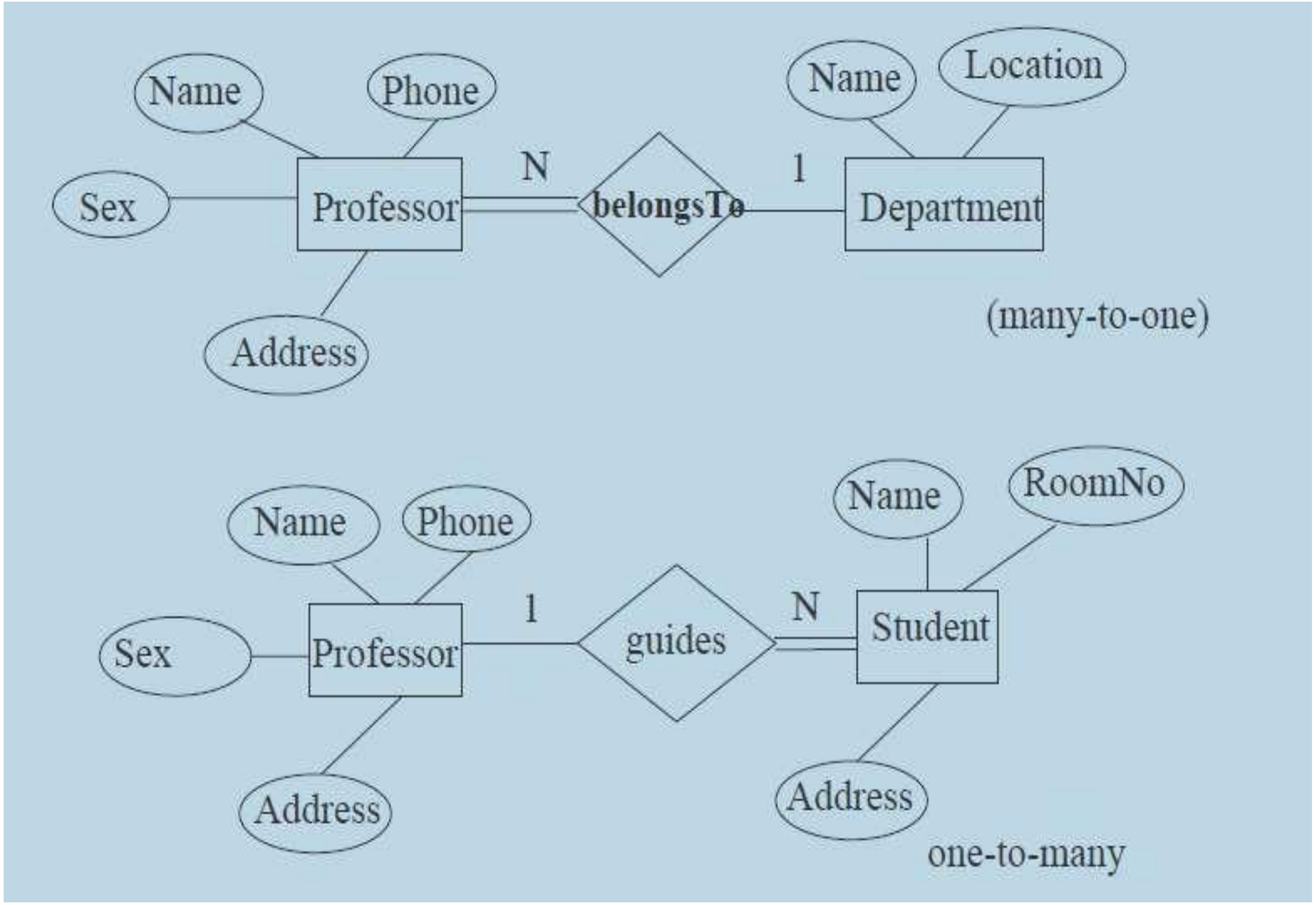
  participation of student in *advisor* relation is total

- **Partial participation**: some entities may not participate in any relationship in the relationship set

  participation of instructor in *advisor* is partial

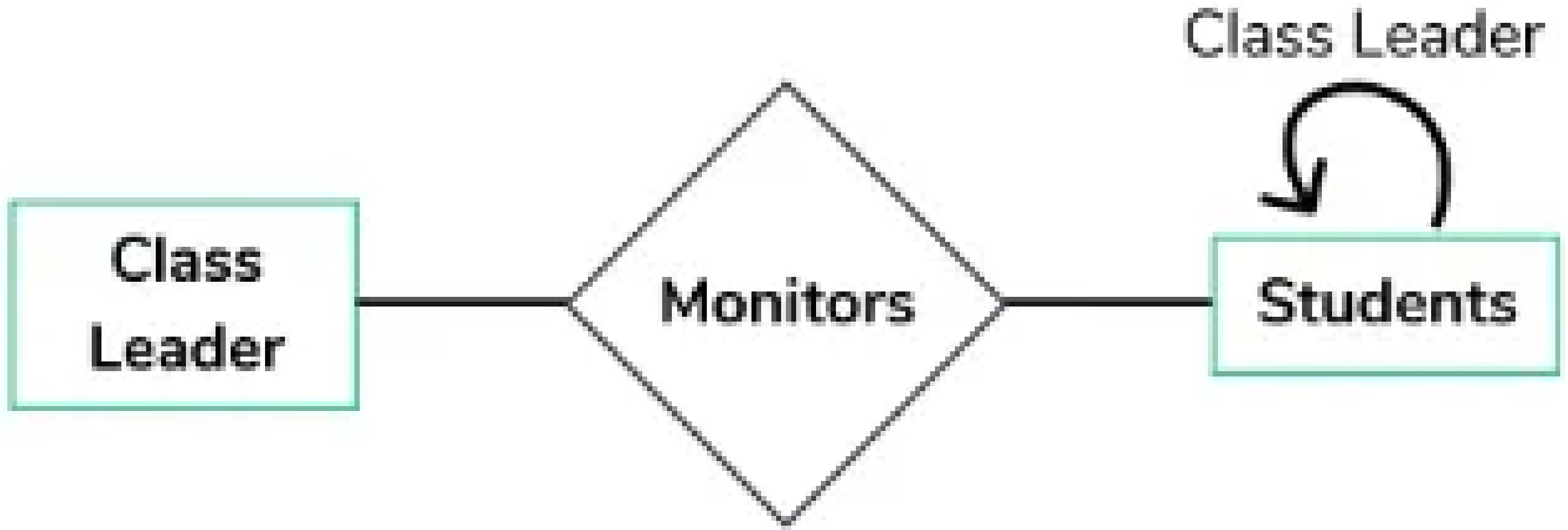


Example


Name
Phone
Sex
Professor
Address
N
belongsTo
1
Name
Location
Department
(many-to-one)
Name
Phone
Sex
Professor
Address
1
guides
N
Name
RoomNo
Student
Address
one-to-many

Recursive Relationship

- An entity in a database is related to itself
- One instance of an entity can be linked to another instance of the same entity type
- Example:



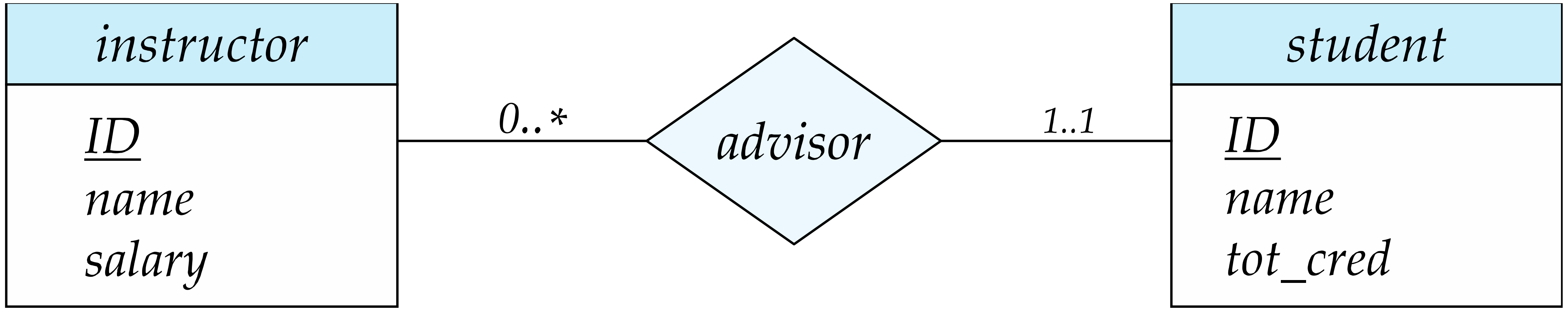
Notation for Expressing More Complex Constraints

- 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
- Minimum value of 1 indicates total participation
- 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

Example

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



Example 1:

A company has several departments. Each department has a supervisor and at least one employee. Every supervisor has only one department under him Employees must be assigned to at least one, but possibly more departments. At least one employee is assigned to a project, but an employee may be on vacation and not assigned to any projects. The important data fields are the names of the departments, projects, supervisors and employees, as well as the supervisor and employee number and a unique project number.

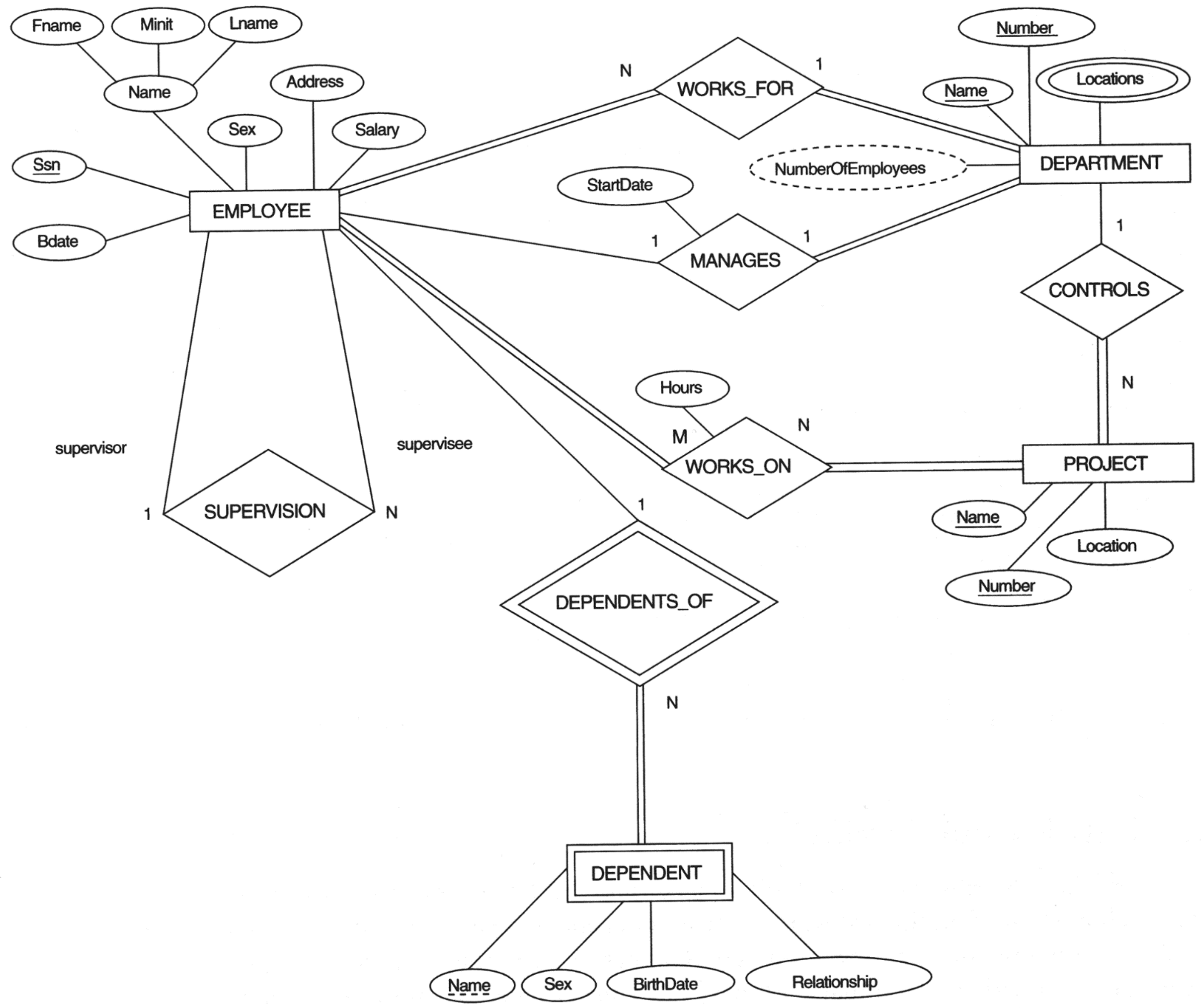
Identify Relationships

- A Department is assigned an employee.
- A Department is run by a supervisor.
- An employee belongs to a department.
- An employee works on a project.
- A supervisor runs a department.
- A project uses an employee.

Fill in Cardinality

- **Supervisor:**
  - ✓Each department has one supervisor.
- **Department:**
  - ✓Each supervisor has one department
  - ✓Each employee can belong to one or more departments
- **Employee:**
  - ✓Each department must have one or more employees
  - ✓Each project must have one or more employees
- **Project:**
  - ✓Each employee can have 0 or more projects.

Example 2: ER-diagram for the COMPANY database



Example 2: ER-schema for the COMPANY database

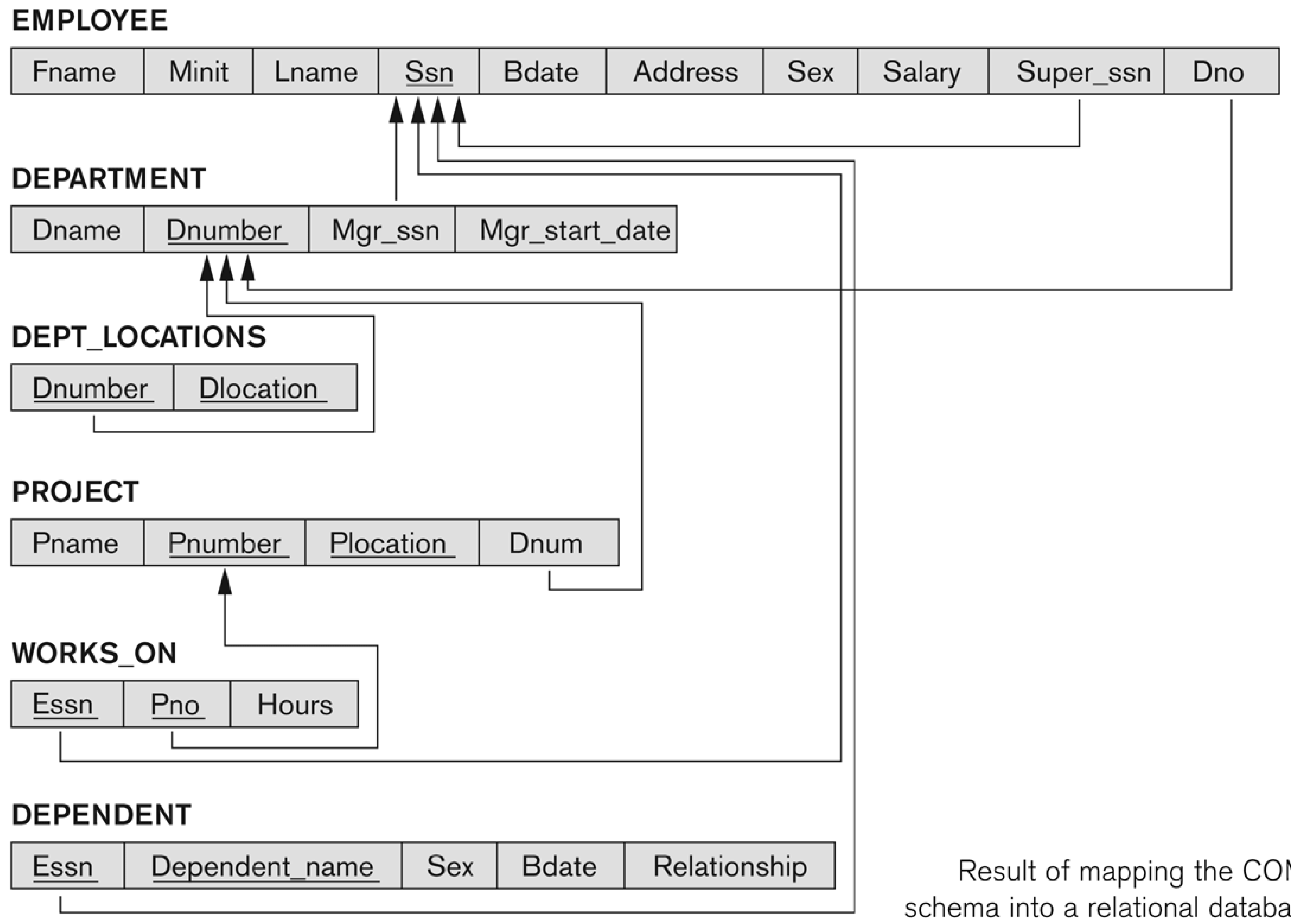


Figure 7.2
Result of mapping the COMPANY ER schema into a relational database schema.

Step 1: Mapping of Regular Entity Types.

- For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes of E.

- Choose one of the key attributes of E as the primary key for R

- If the chosen key of E is composite, the set of simple attributes that form it will together form the primary key of R.

Example:

- We create the relations EMPLOYEE, DEPARTMENT, and PROJECT in the relational schema corresponding to the regular entities in the ER diagram. SSN, DNUMBER, and PNUMBER are the primary keys for the relations EMPLOYEE, DEPARTMENT, and PROJECT as shown.

Step 2: Mapping of Weak Entity Types

- For each weak entity type W in the ER schema with owner entity type E, create a relation R & include all simple attributes (or simple components of composite attributes) of W as attributes of R.

- Also, include as foreign key attributes of R the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s).

- The primary key of R is the combination of the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any.

Example:

- Create the relation DEPENDENT in this step to correspond to the ***weak entity type*** DEPENDENT. Include the primary key SSN of the EMPLOYEE relation as a foreign key attribute of DEPENDENT (renamed to ESSN).

- The primary key of the DEPENDENT relation is the combination {ESSN, DEPENDENT_NAME} because DEPENDENT_NAME is the partial key of DEPENDENT.

Step 3: Mapping of Binary 1:1 Relation Types

- For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R.
- **Foreign Key approach:** Choose one of the relations-say S-and include a foreign key in S the primary key of T. It is better to choose an entity type with total participation in R in the role of S.

  Example: 1:1 relation MANAGES is mapped by choosing the participating entity type DEPARTMENT to serve in the role of S, because its participation in the MANAGES relationship type is total.

- **Merged relation option:** An alternate mapping of a 1:1 relationship type is possible by merging the two entity types and the relationship into a single relation. **This may be appropriate when both participations are total.**

- **Cross-reference or relationship relation option:** The third alternative is to set up a third relation R for the purpose of cross-referencing the primary keys of the two relations S and T representing the entity types.

Step 4: Mapping of Binary 1:N Relationship Types

- For each regular binary 1:N relationship type R, identify the relation S that represent the participating entity type at the N-side of the relationship type.

- Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R.

- Include any simple attributes of the 1:N relation type as attributes of S.

Example:

1:N relationship types WORKS_FOR, CONTROLS, and SUPERVISION in the figure. For WORKS_FOR we include the primary key DNUMBER of the DEPARTMENT relation as foreign key in the EMPLOYEE relation and call it DNO.

Step 5: Mapping of Binary M:N Relationship Types

- For each regular binary M:N relationship type R, create a new relation S to represent R.

- Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; their combination will form the primary key of S.

- Also include any simple attributes of the M:N relationship type (or simple components of composite attributes) as attributes of S.

Example: The M:N relationship type WORKS_ON from the ER diagram

- The primary keys of the PROJECT and EMPLOYEE relations are included as foreign keys in WORKS_ON and renamed PNO and ESSN, respectively.

- Attribute HOURS in WORKS_ON represents the HOURS attribute of the relation type. The primary key of the WORKS_ON relation is the combination of the foreign key attributes {ESSN, PNO}.

Step 6: Mapping of Multivalued attributes

- For each multivalued attribute A, create a new relation R.
- This relation R will include an attribute corresponding to A, plus the primary key attribute K-as a foreign key in R-of the relation that represents the entity type of relationship type that has A as an attribute.
- The primary key of R is the combination of A and K. If the multivalued attribute is composite, we include its simple components.

Example: The relation DEPT_LOCATIONS is created.

- The attribute DLOCATION represents the multivalued attribute LOCATIONS of DEPARTMENT, while DNUMBER-as foreign key-represents the primary key of the DEPARTMENT relation.
- The primary key of R is the combination of {DNUMBER, DLOCATION}.

Q
&
A