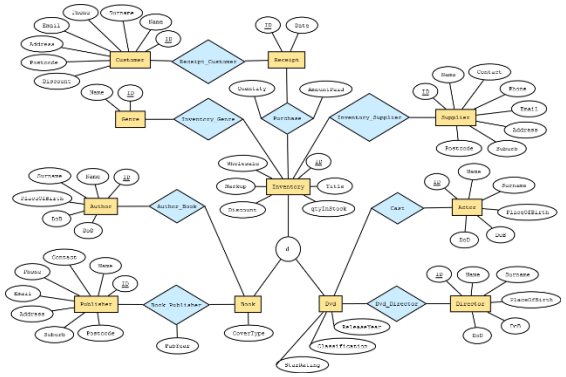
Lecture 11 - Database Modelling & Database Design

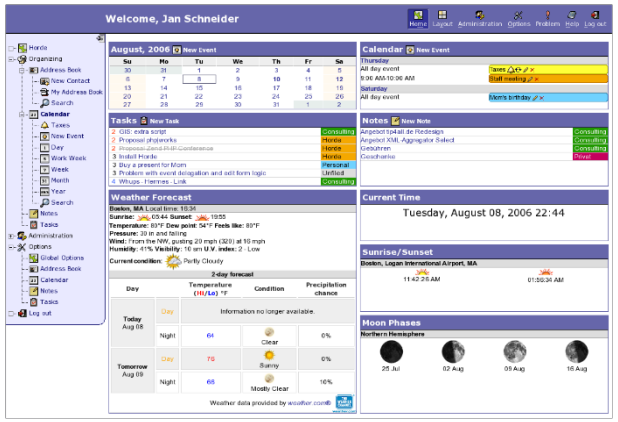
CS211 - Introduction to Database

Database Modelling

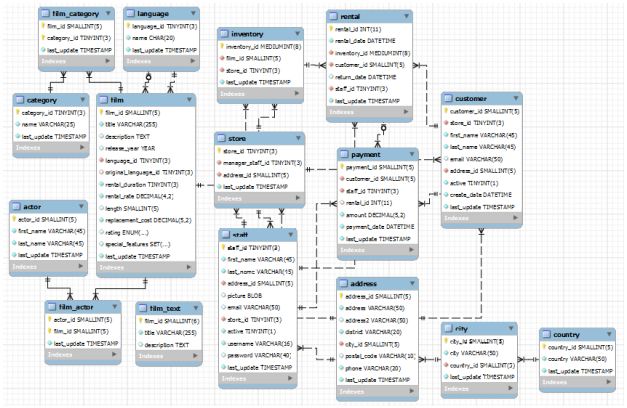
Data Modelling



Database Design



Implementation



Transaction
Indexing
Concurrency
etc.

Database Design Phases

1. **Specification of user requirements**

   - interact extensively with domain experts and users

2. **Conceptual-design**

   - translates user requirements into a conceptual schema of the database using a conceptual data model (ER-model)

3. **Specification of functional requirements**

   - users describe the kinds of operations that will be performed on the data

   - designer can review the schema to ensure it meets functional requirements

4. **Implementation of the database**

   - ***logical-design phase*** maps the high-level conceptual schema onto the implementation data model (relational data model)

   - ***physical-design phase*** specifies the physical features of the database like form of file organization and choice of index structures

Conceptual data Model vs. Data Model

Conceptual data Model:

E-R model
O-O model …

Data Model: Logic + physical

Relational model
Network
Hierarchical

- **Conceptual data Model**
  - Abstract of the real world
  - Independent of Computer World
- **Implementation Data Model**
  - Data/Computer World

Database Design Goals

1. **Deciding how to represent *entities***

   - any distinctly identifiable item in the real world

2. **Eliminate redundancy**

   - redundant representation of information may lead to data inconsistency among the various copies of information

3. **Avoid incompleteness**

   - a bad design may make certain aspects difficult or impossible to model

4. **Choose the best design from the available designs**

The Entity-Relationship Model

- Introduced in 1976 by P.P.S.Chen

- It models the **Entities** and the **Relationship** between them

- The 4 Concepts describing the world
  - Entity
  - Attribute
  - Relationship
  - Key

Concept	Instance
Entity	**Student, Course**
Attribute	**Student (sID, sName,…) Course (cID, cName,…)**
Relationship	**Takes (sID, cID, …)**
Key	**sID, cID**

Entity Sets

- An **entity** is a “thing” or “object” in the real world that is distinguishable from all other objects.

  - An entity has a set of attributes (properties), and the values for some set of properties may uniquely identify an entity (key)

- An **entity set E** is a set of entities of the same type that share the same properties, or attributes.

  - Student, Instructor

- An **attribute** is used for describing Entity
  - *Entity* (*attributes1, attributes2, …)*
  - Student (id, name, gender …)

- An **instance** lists the actual values of attributes
  - *A student instance* – 00128, Zhang, Male…

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

instructor

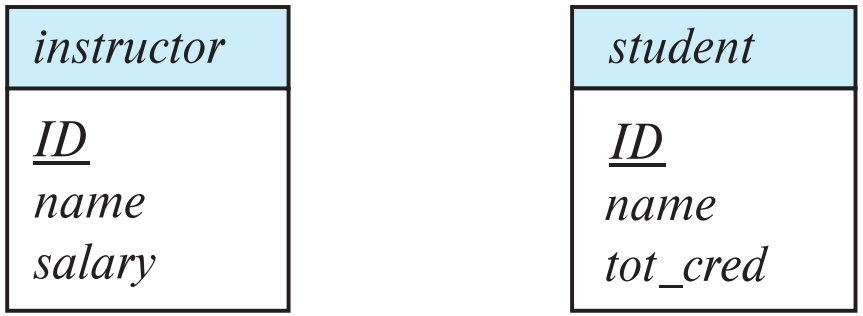
98988	Tanaka
12345	Shankar
00128	Zhang
76543	Brown
76653	Aoi
23121	Chavez
44553	Peltier

student

Representing Entity Sets via ER Diagrams

Entity sets can be represented graphically as follows:

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



Relationship Sets

- A **relationship** is an association among several entities.
  - *Example:*

44553 (John) — *advisor* — 22222 (Einstein)
student entity — relationship set — *instructor* entity

- A **relationship set** is a set of relationships of the same type.

- Formally a **relationship set** **R** is a mathematical relation on $n \geq 2$ (possibly non-distinct) entity sets.

If $E_1, E_2, \ldots E_n$ are entity sets, then **R** is a subset of

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

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

  - *Example:*

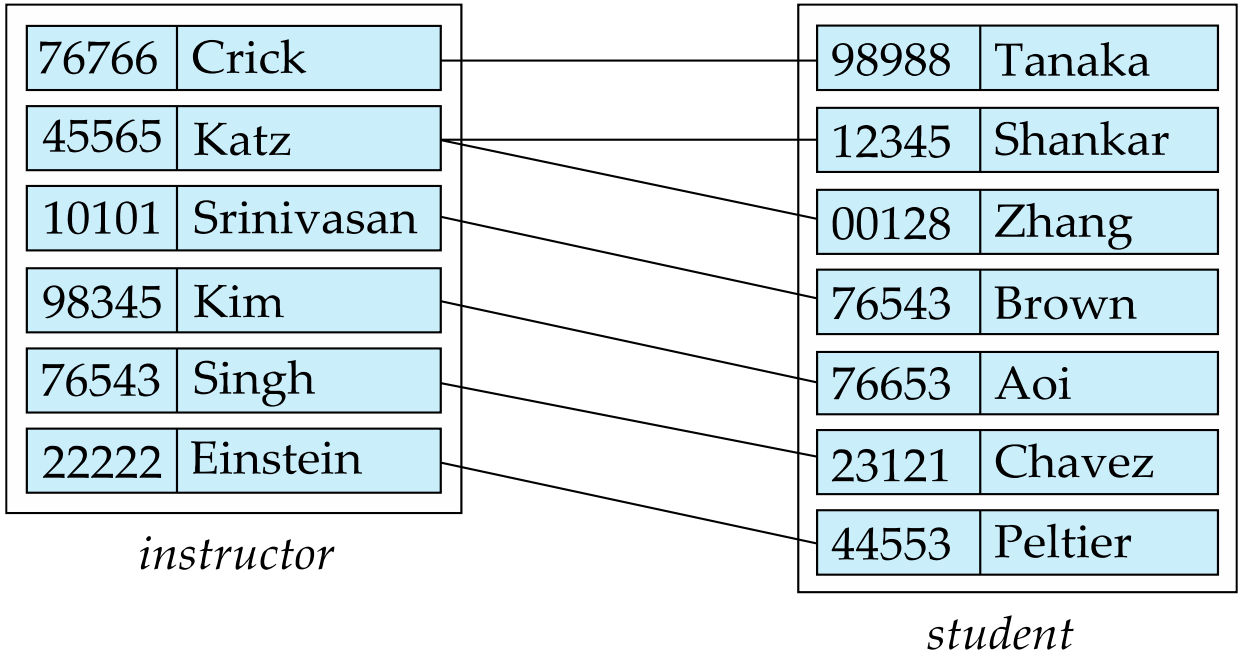
$(44553,22222) \in advisor$

Relationship Sets

- The association between entity sets is referred to as **participation**.

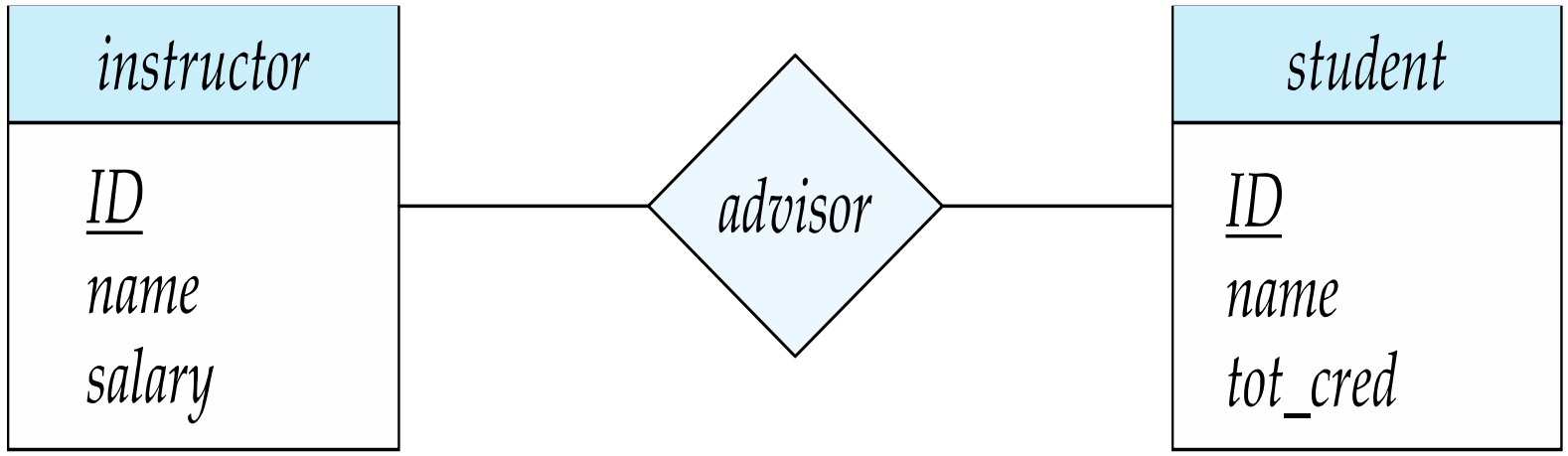
  - *Example:* we define the relationship set *advisor* to denote the associations between students and the instructors who act as their advisors.

- A **relationship instance** in an E-R schema represents an association between the named entities in the real-world that is being modeled.



Representing Relationship Sets via ER Diagrams

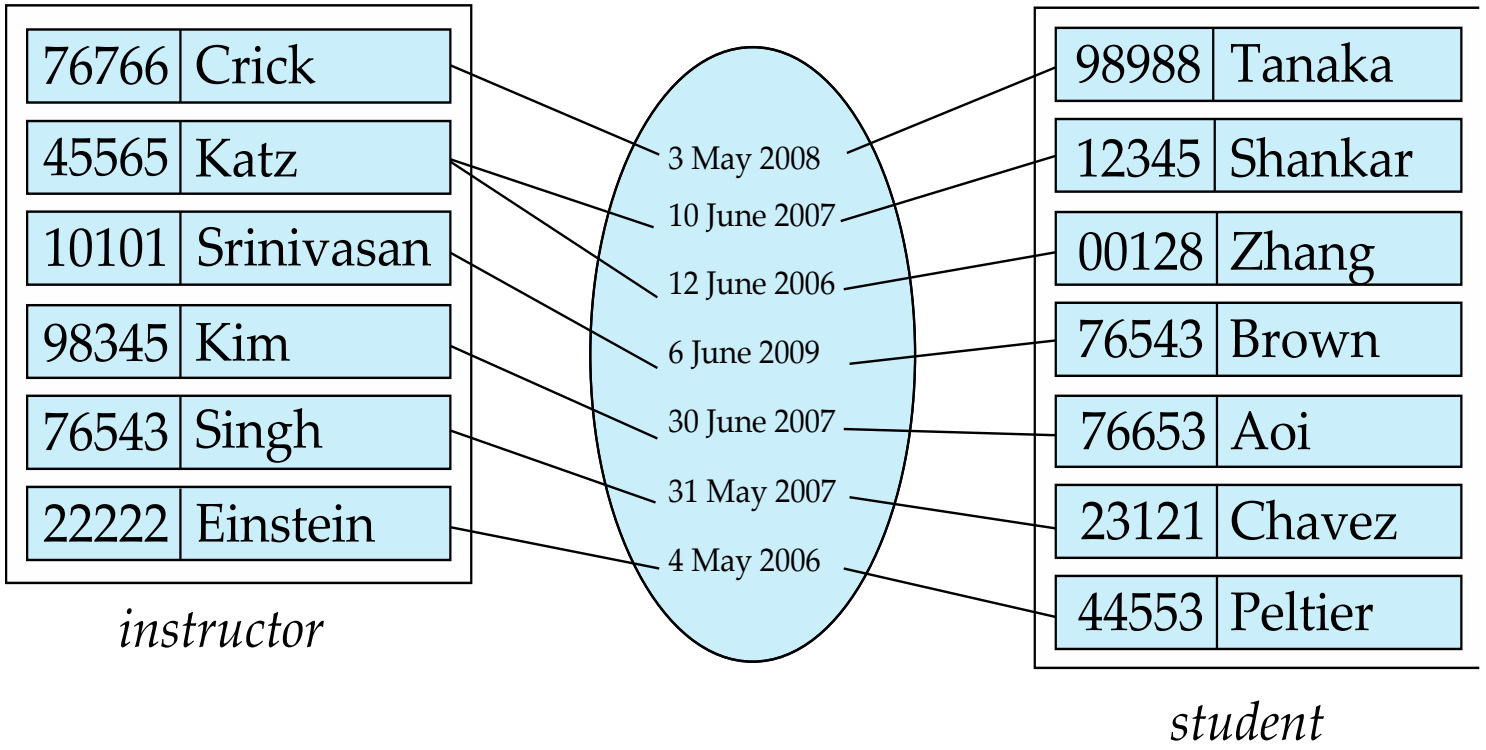
- Diamonds represent relationship sets.



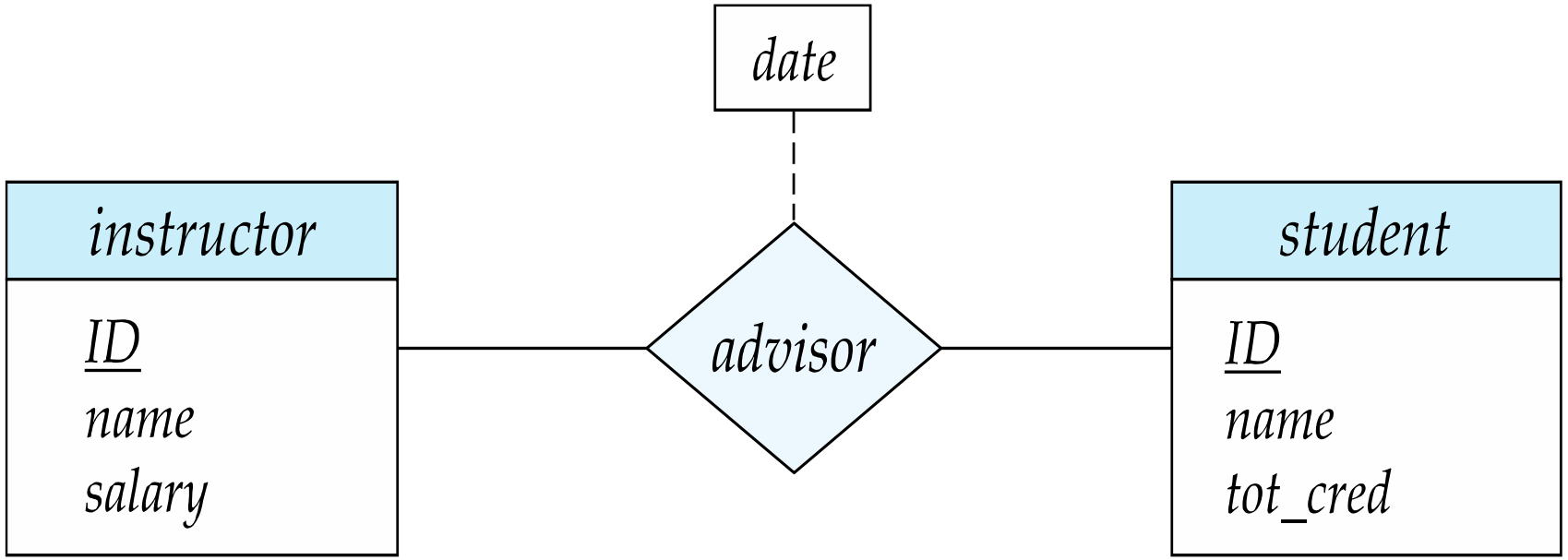
Relationship Set Attribute

- An **attribute** can also be associated with 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



Representing Relationship Set Attributes via ER Diagrams

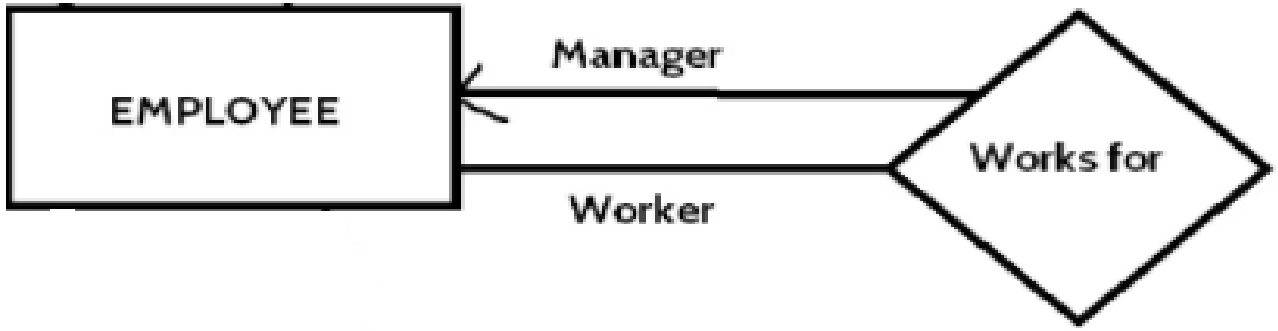


Entity Roles

- The function that an entity plays in a relationship is called that entity's **role**.

- Since entity sets participating in a relationship set are generally distinct, roles are implicit and are not usually specified.

- ***Entity sets of a relationship need not be distinct.***

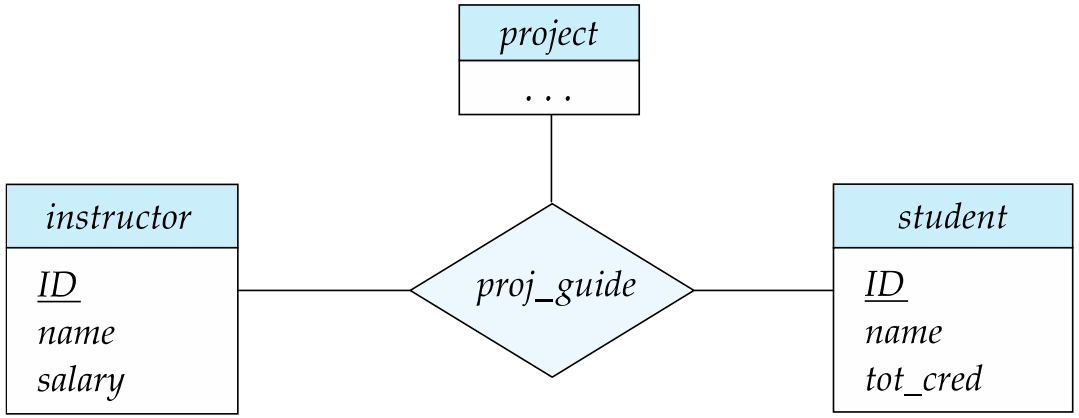


The labels "*Manager*" and "Worker" are called **roles**.

Degree of a Relationship Set

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

- There are occasions when it is more convenient to represent relationships as non-binary.



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

Simple Attributes / Complex Attributes

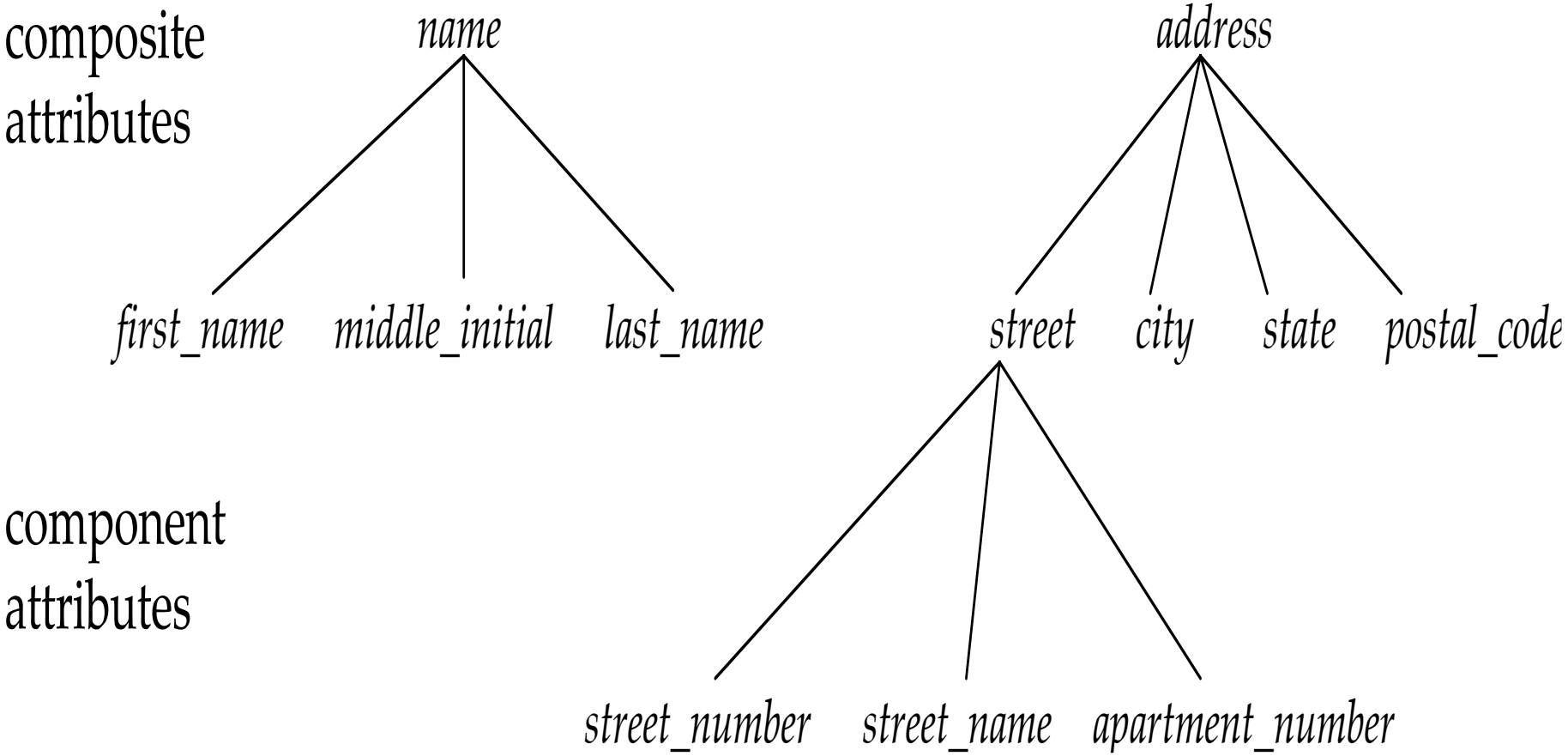
- For each attribute, there is a set of permitted values, called the **domain**, or **value set**, of that attribute.

  Example: the domain of attribute **semester** might be strings from the set **{ Fall, Winter, Spring, Summer }**

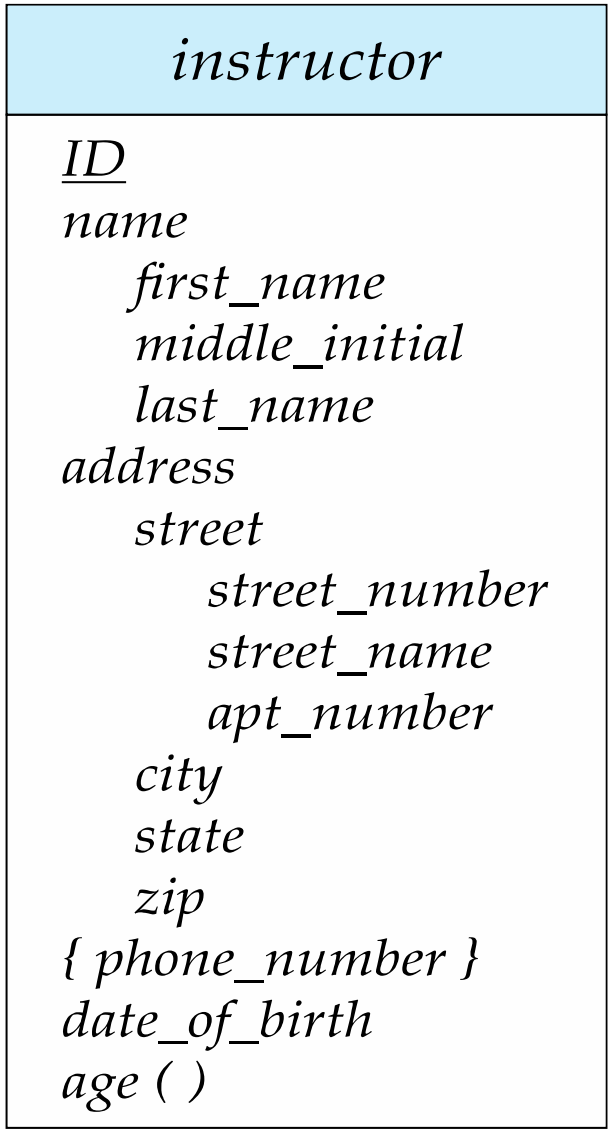
- **Attribute types:**
  - **Simple** and **composite** attributes
    - Example: composite attribute: *address*
  - **Single-valued** and **multivalued** attributes
    - Example: multivalued attribute: *phone_numbers*
  - **Derived** attributes
    - ➢ Can be computed from other attributes
    - Example: *age*, given date_of_birth

Composite Attributes

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



Representing Complex Attributes in ER Diagram



Constraints

- We study the following constraints defined by E-R model
    1. Mapping Cardinalities
    2. Participation Constraints
    3. Primary Key constraints

Mapping Cardinality Constraints

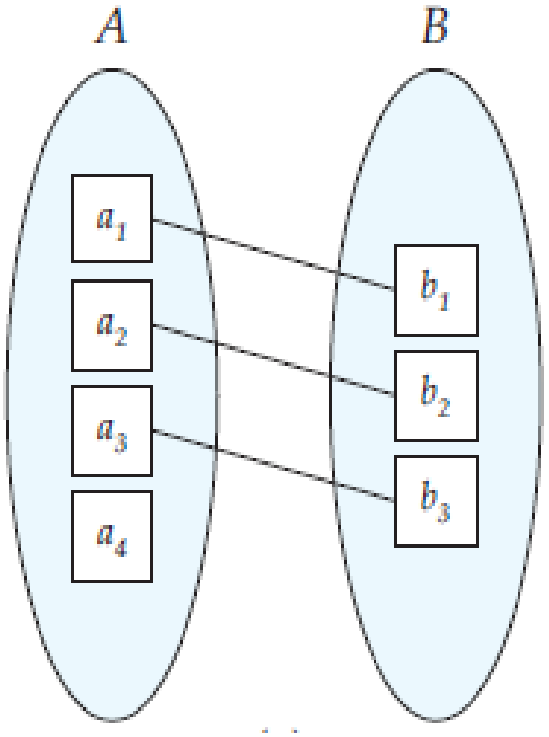
- **Mapping cardinalities**, or **cardinality ratios**, express the number of entities to which another entity can be associated via a relationship set.

- For a binary relationship set ***R*** between entity sets ***A*** and ***B***, the mapping cardinality must be one of the following:
  - One to one
  - One to many
  - Many to one
  - Many to many

Mapping Cardinality Constraints

- **One-to-one**:

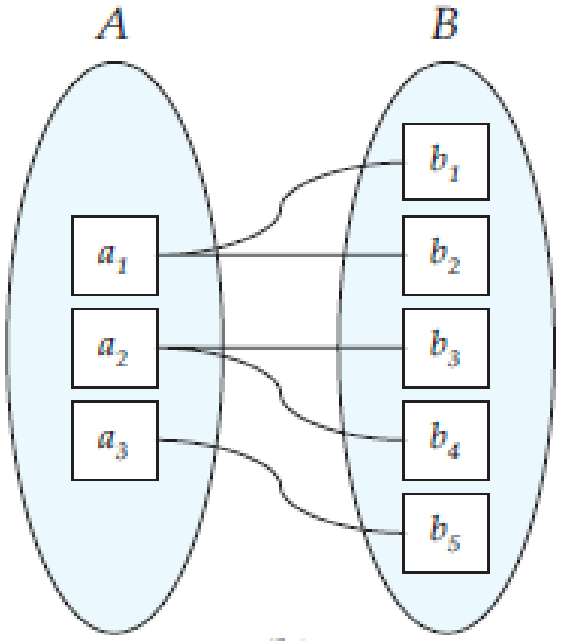
  An entity in *A* is associated with *at most* one entity in *B*, and an entity in *B* is associated with *at most* one entity in *A*.



Mapping Cardinality Constraints

- **One-to-many**:

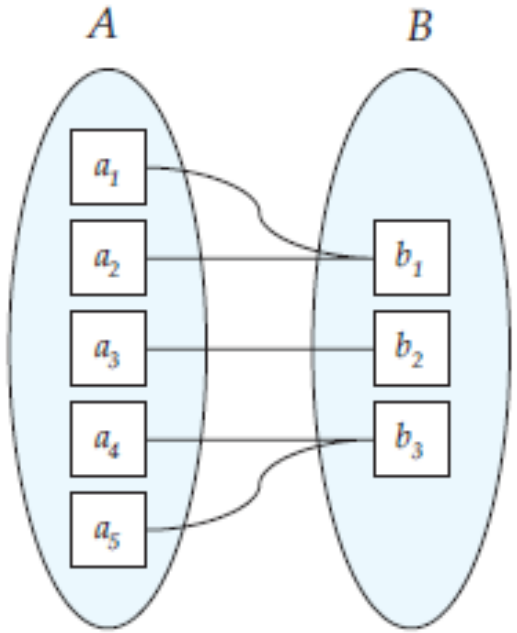
  An entity in *A* is associated with any number (zero or more) of entities in *B*. An entity in *B*, however, can be associated with *at most* one entity in *A*.



Mapping Cardinality Constraints

- **Many-to-one**:

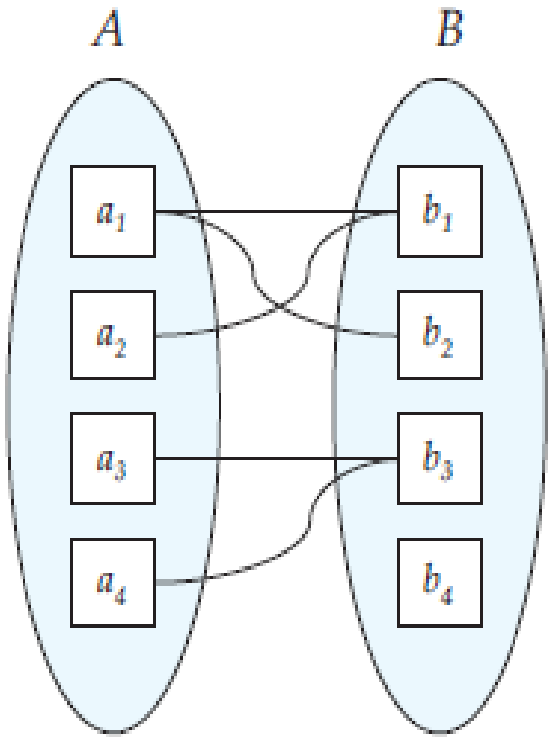
  An entity in *A* is associated with *at most* one entity in *B*. An entity in *B*, however, can be associated with any number (zero or more) of entities in *A*.



Mapping Cardinality Constraints

- **Many-to-many**:

  An entity in *A* is associated with any number (zero or more) of entities in *B*, and an entity in *B* is associated with any number (zero or more) of entities in *A*.

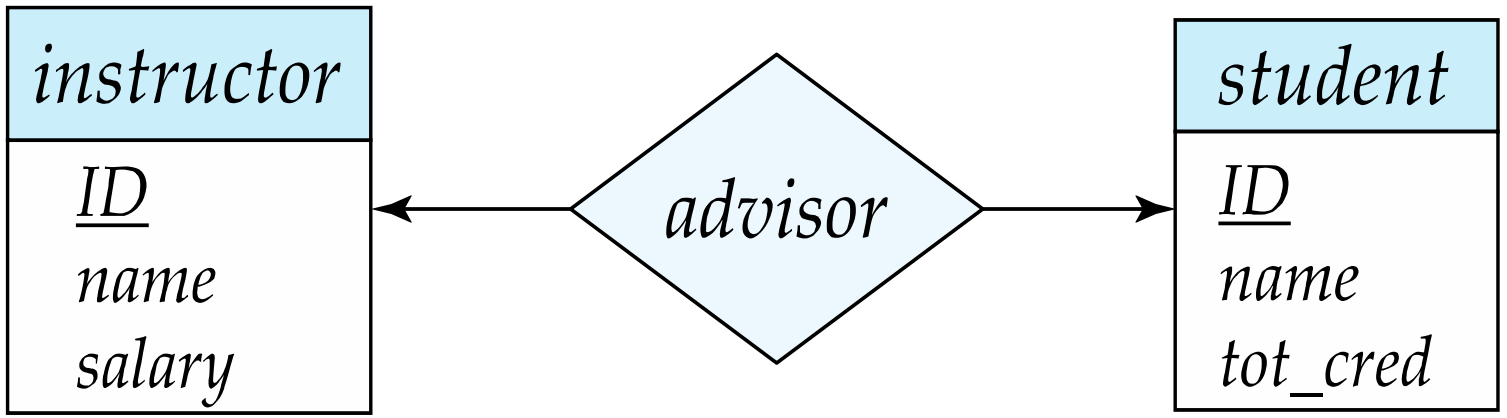


Representing Cardinality Constraints in ER Diagram

- 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.

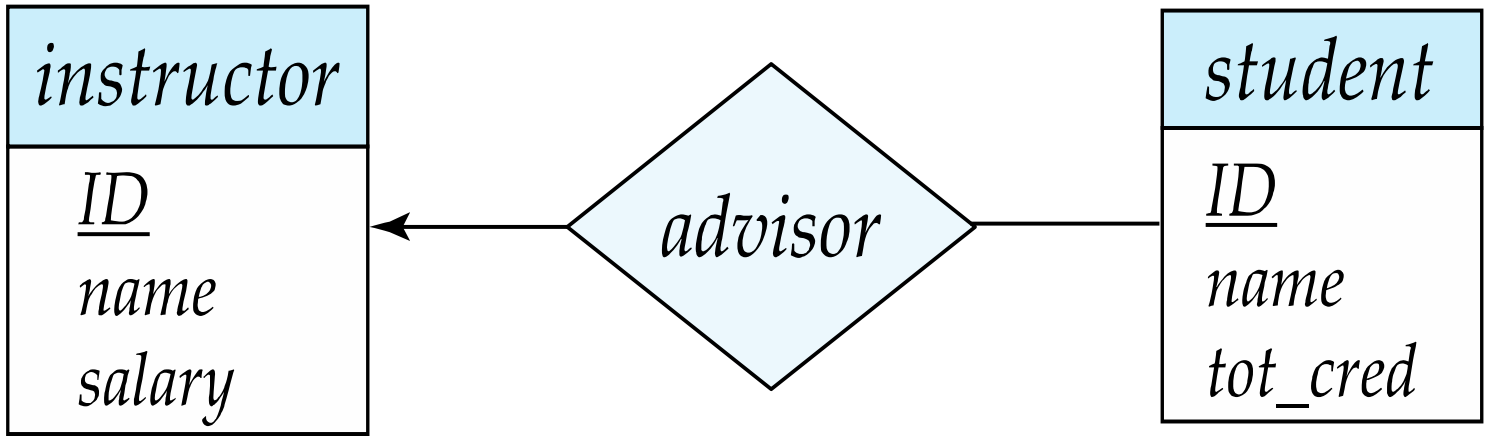
- **One-to-one** relationship between an *instructor* and a *student* :

  A student is associated with at most one *instructor* via the relationship *advisor*



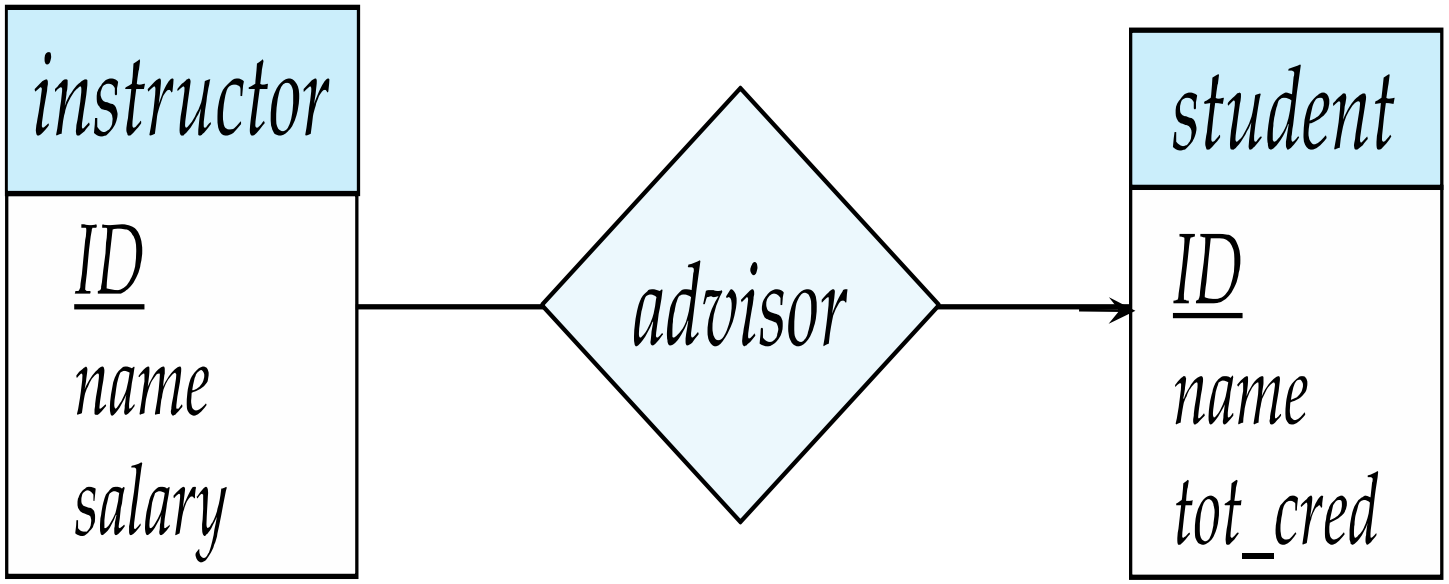
Representing Cardinality Constraints in ER Diagram

- **One-to-many** relationship between an *instructor* and a *student:*
  - an instructor is associated with several (including ZERO) students via *advisor*



Representing Cardinality Constraints in ER Diagram

- **Many-to-one** relationship between an *instructor* and a *student:*
  - many instructors are associated with one student via *advisor*



Representing Cardinality Constraints in ER Diagram

- **Many-to-many** relationship between an *instructor* and a *student:*
  - An instructor is associated with several (possibly ZERO) students via *advisor*
  - A student is associated with several (possibly ZERO) instructors via *advisor*