Unit I - Introduction

Purpose of Database System - Views of data – Data models, Database Management system - Three-schema architecture of DBMS, Components of DBMS. **Entity – Relationship Model - Conceptual data modelling - motivation**, entities, entity types, attributes, relationships, relationship types, E/R diagram notations, Examples

Recap

- **Data Modelling** - process of creating a data model for the data to be stored in a database.
- Two Types
  - ER
  - UML

Design Phases

- Initial phase -- characterize fully the data needs of the prospective database users.
- Second phase -- choosing a data model
- Final Phase -- Moving from an abstract data model to the implementation of the database
  - Logical Design - Deciding on the database schema.
    - **Redundancy**
    - **Incompleteness**
  - Physical Design - Deciding on the physical layout of the database

Conceptual data modelling

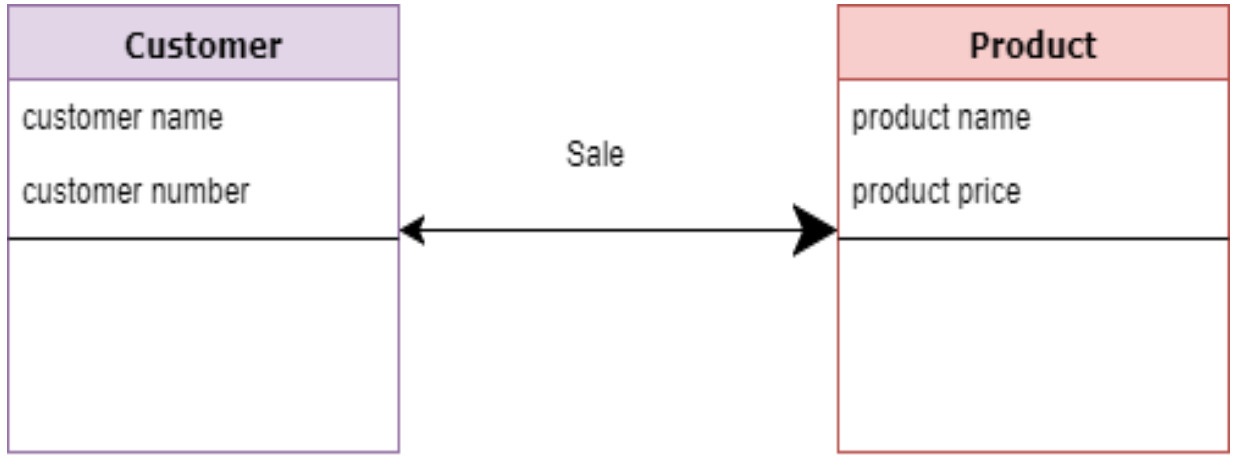
- **Conceptual Data Model**
  - Organized view of database concepts and their relationships.
  - The purpose of creating a conceptual data model is to establish entities, their attributes, and relationships.

Design Approaches

- Entity Relationship Model
  - Models an enterprise as a collection of *entities* and *relationships*
    - Entity: a “thing” or “object”
      - Described by a set of *attributes*
    - Relationship: an association among several entities
  - Represented diagrammatically by an ***entity-relationship diagram***

Entity Relationship Model

- Represents the overall logical structure of a database.
- The ER data model employs three basic concepts:
  - **Entity**: A real-world thing
  - **Attribute**: Characteristics or properties of an entity
  - **Relationship**: Dependency or association between two entities



Entity Sets

- An **entity** is an object that exists and is distinguishable from other objects.
  - Example: specific person, company, event, plant
- 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
- An entity is represented by a set of attributes; i.e., descriptive properties possessed by all members of an entity set.
  - Example:

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

- **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

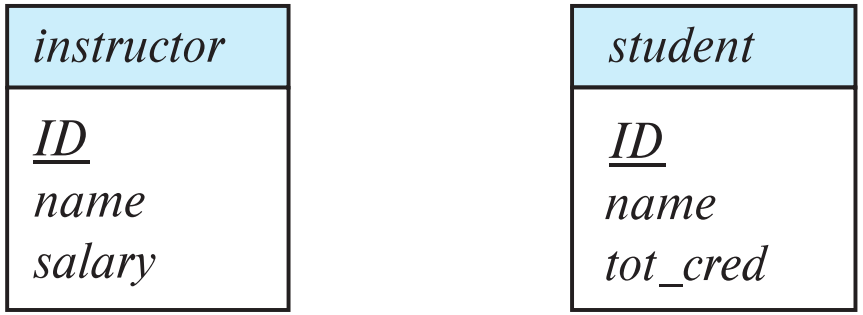
instructor

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

- A **relationship** is an association among several entities

  Example:

  44553 (Peltier) — *advisor* — 22222 (Einstein)

  student entity — relationship set — *instructor* entity

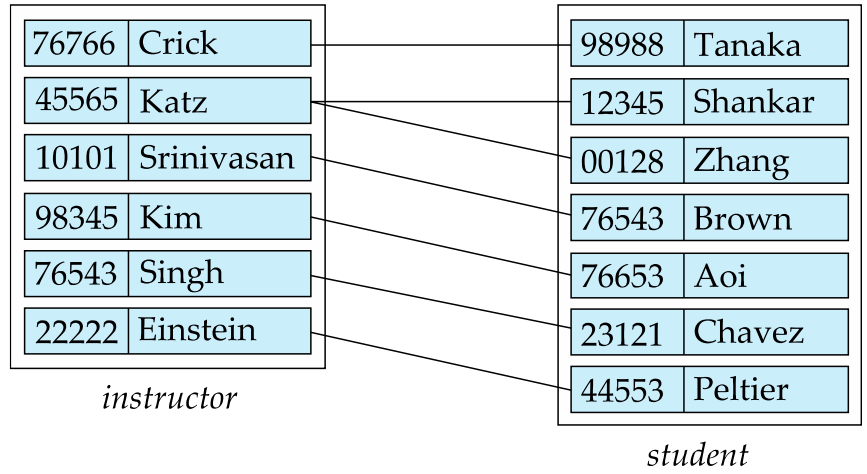
- A **relationship set** is a mathematical relation among $n \geq 2$ entities, each taken from entity sets

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

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

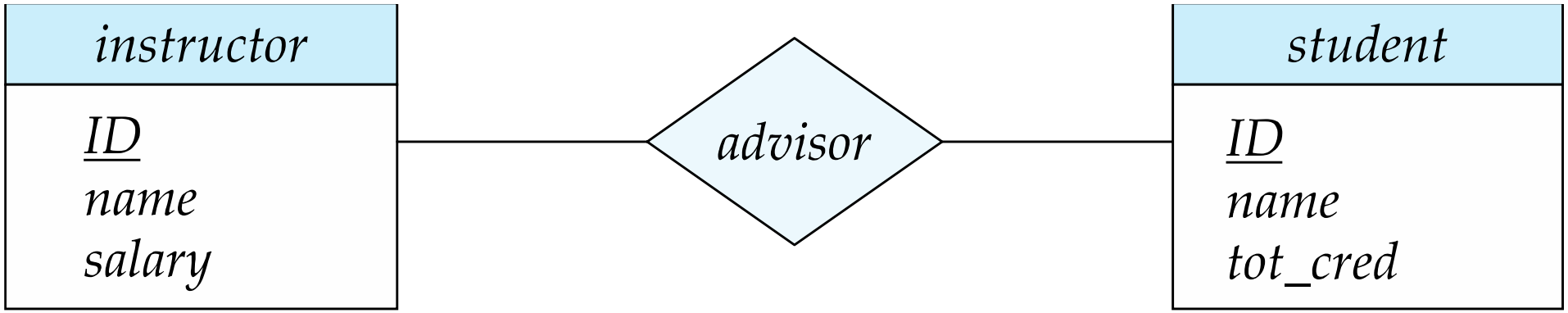
  - Example:

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



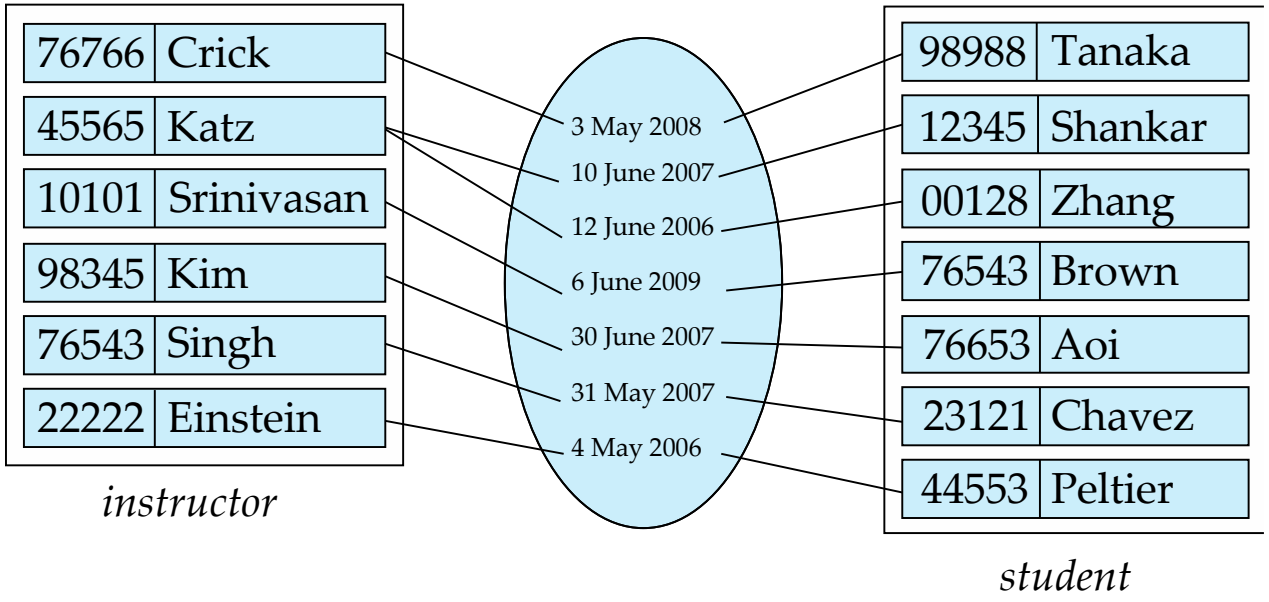
Representing Relationship Sets via ER Diagrams

- Diamonds represent relationship sets.

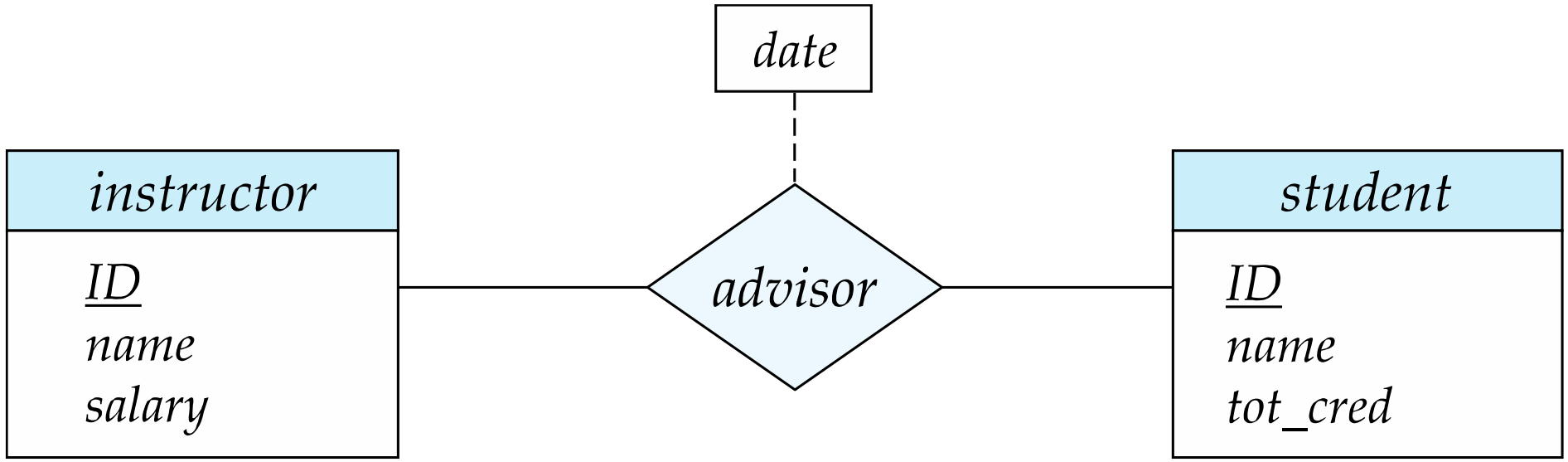


Relationship Sets

- An attribute can also be associated with a relationship set

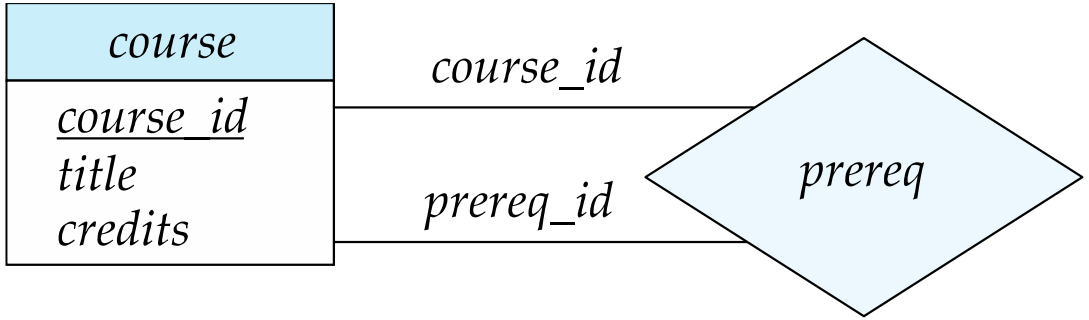


Relationship Sets with Attributes



Roles

- Entity sets of a relationship need not be distinct

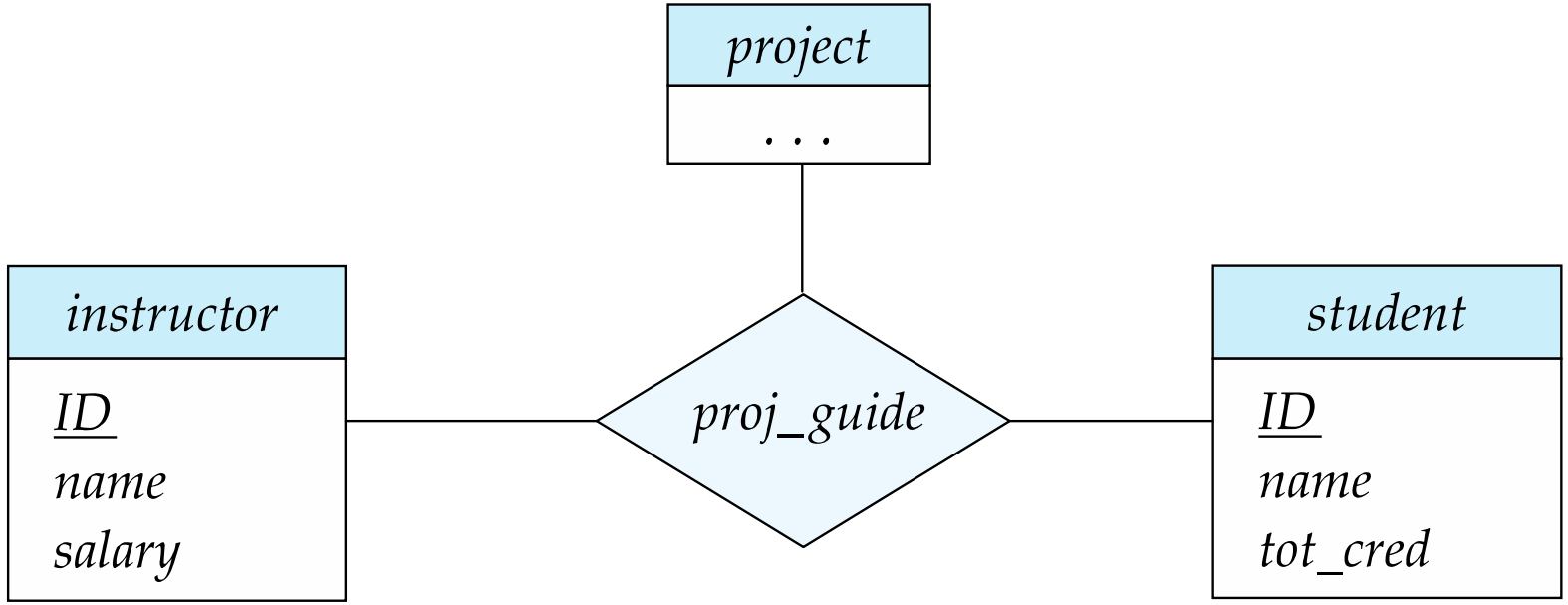


Degree of a Relationship Set

- 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 Sets

E-R Diagram with a Ternary Relationship

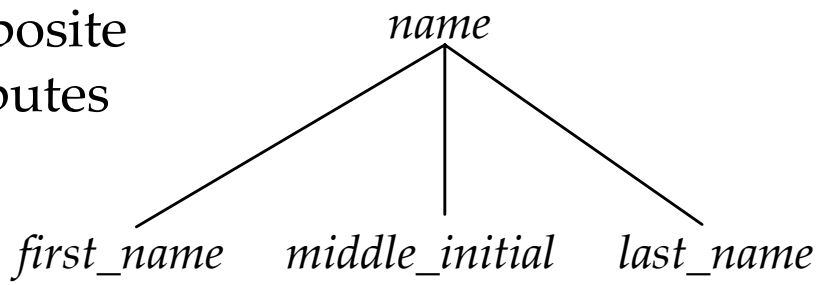


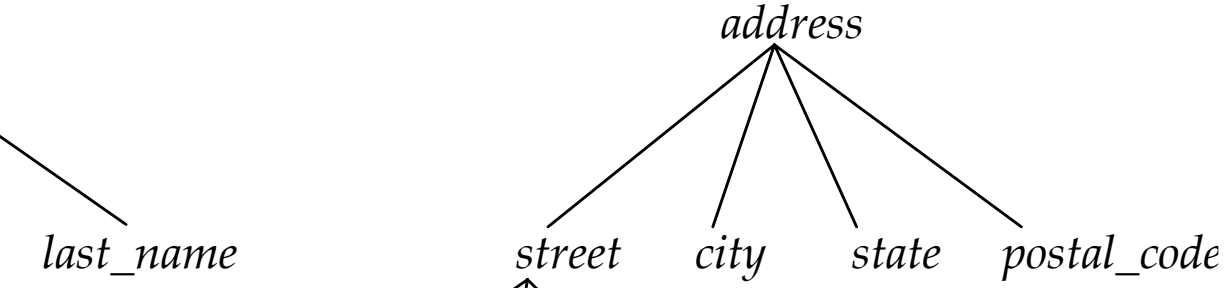
Attributes

- Attribute types:
  - **Simple** and **composite** attributes.
    - Composite Attributes- further divide into more simple attributes.
    - Student Roll Number, Employee id, Account balance, Salary, Account number, and Aadhar number are an example of simple attributes.
  - **Complex Attributes -** Name and 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
  - **Domain –** the set of permitted values for each attribute

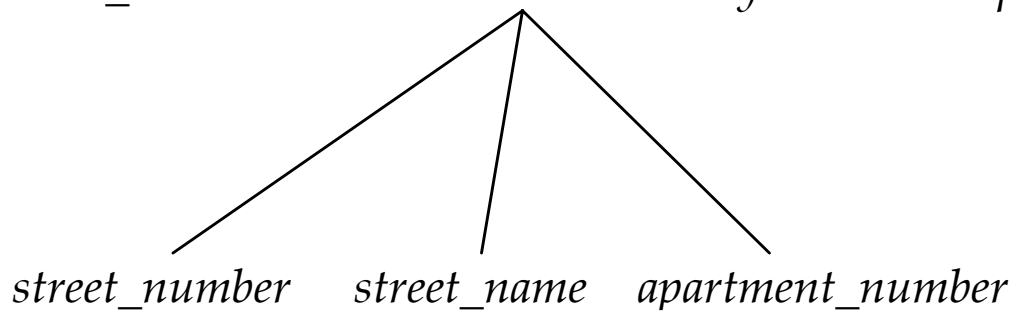
Composite Attributes

composite attributes



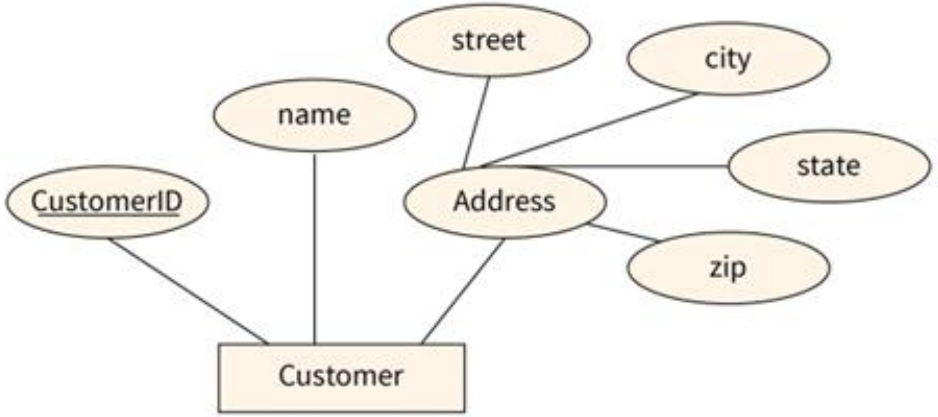


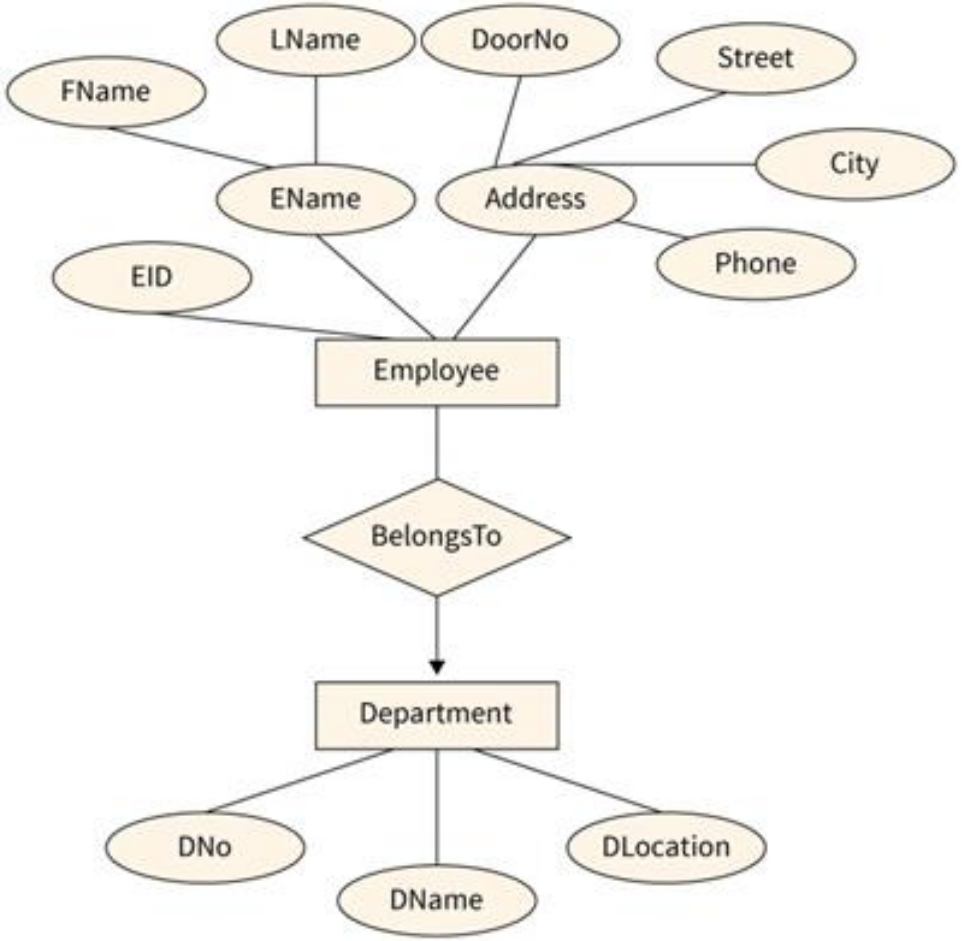
component attributes



E-R Diagram with 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 ( )*

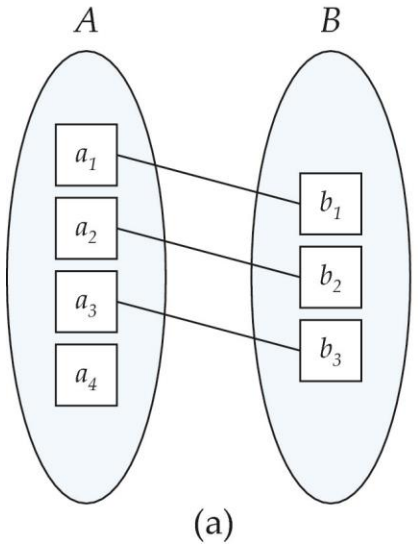




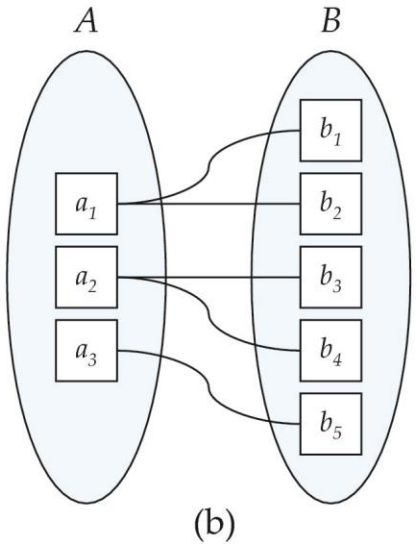
Mapping Cardinality Constraints

- Express the number of entities to which another entity can be associated via a relationship set.
- 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**

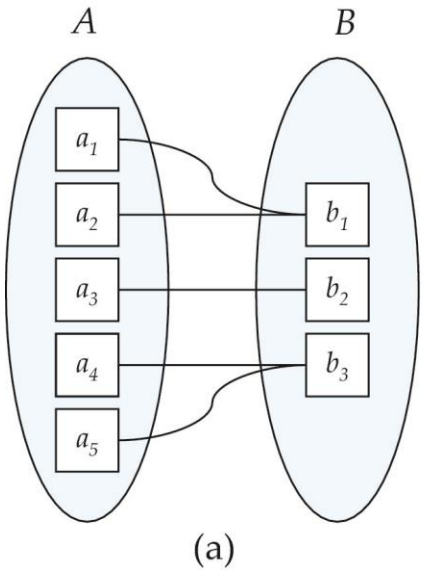
Mapping Cardinality



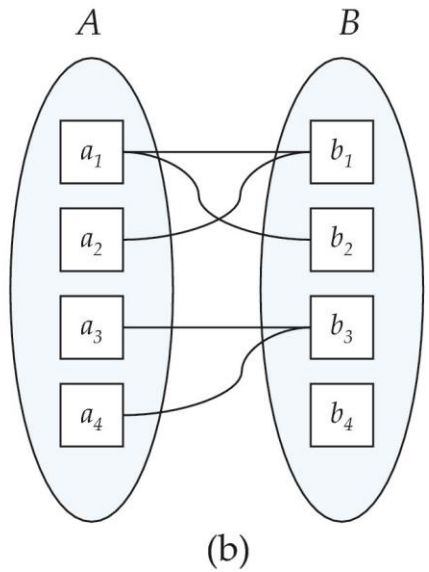
One to one



One to many



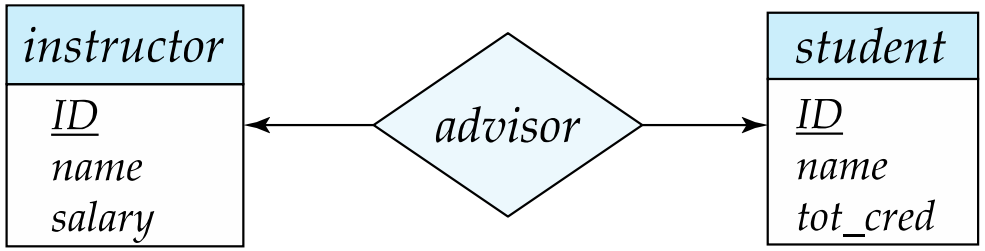
Many to one

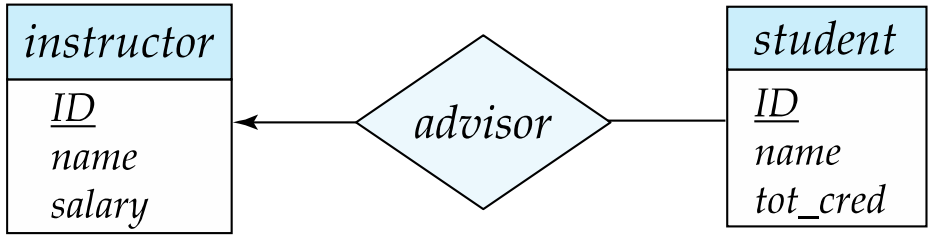


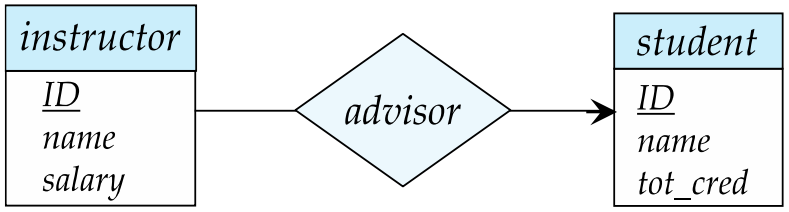
Many to many

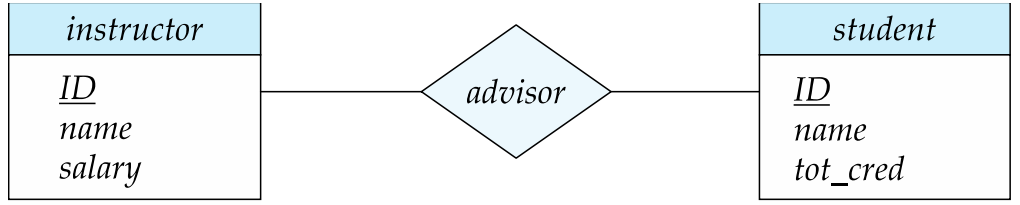
Representing Cardinality Constraints in ER Diagram

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

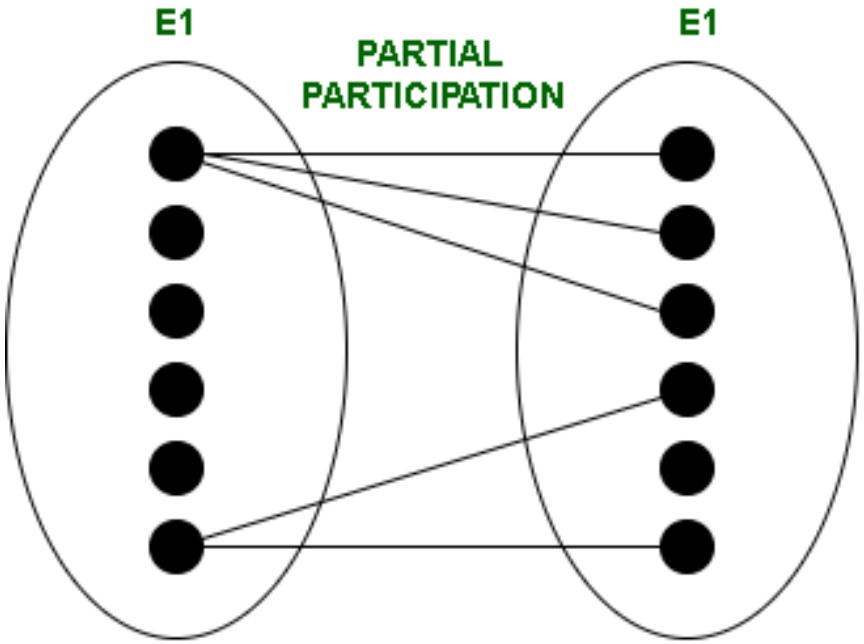


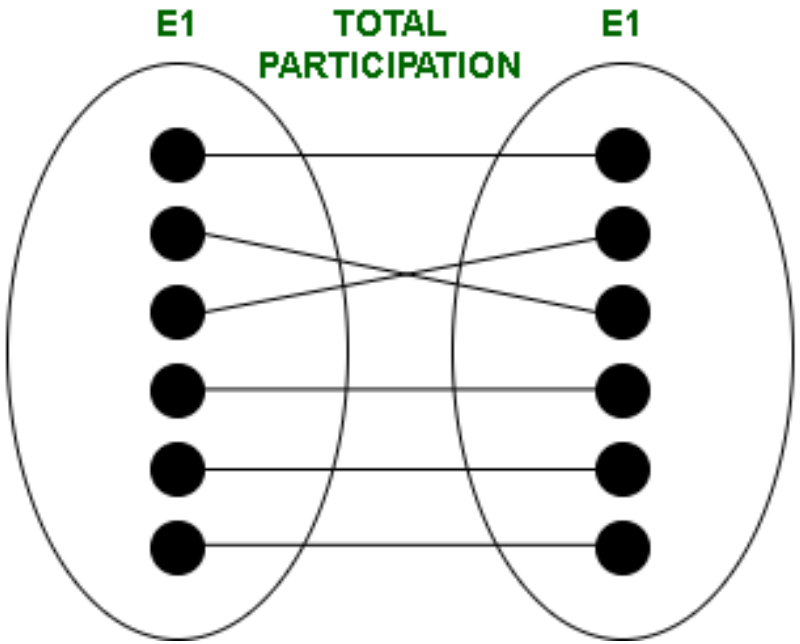






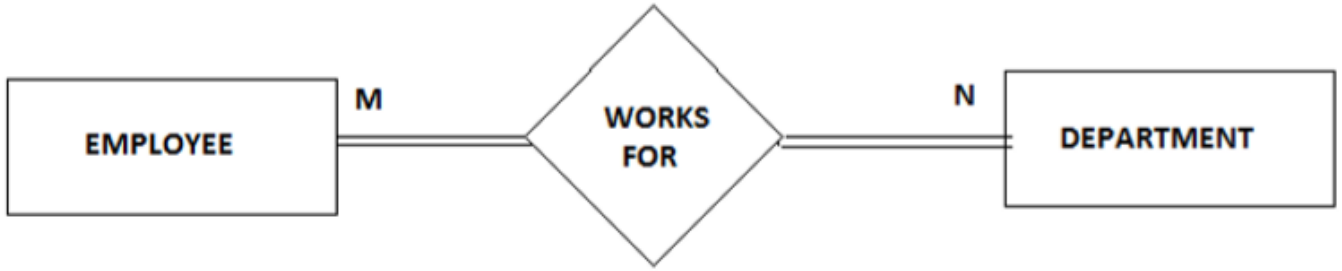
Total and Partial Participation



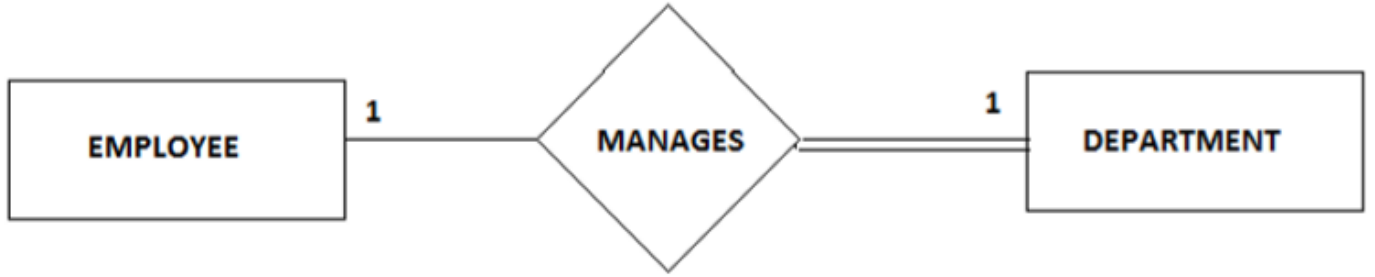


Total and Partial Participation

Total Participation

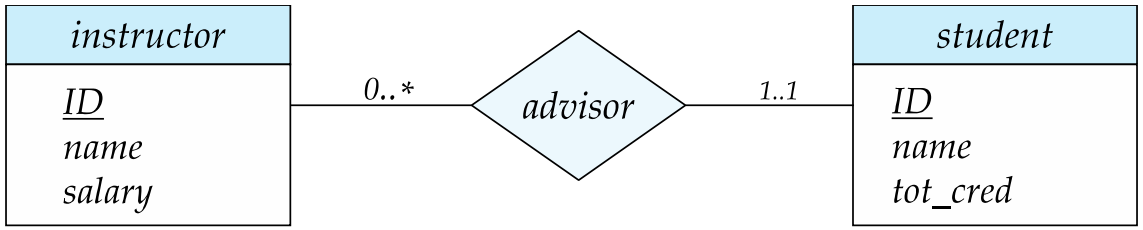


Partial Participation



Notation for Expressing More Complex Constraints

- A line may have an associated minimum and maximum cardinality (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

Thank You!