



# Database Design Using the E-R Model

Delivered by:

Dr. Saad Adnan Abed

## Outline



**Overview of the  
design process**

**The entity  
relationship model**

**Attribute types**

# The design process

- **Initial phase** -- characterize fully the data needs of the prospective database users.
- **Second phase** -- choosing a data model
  - Applying the concepts of the chosen data model
  - Translating these requirements into a conceptual schema of the database.
  - A fully developed conceptual schema indicates the functional requirements of the enterprise.
    - Describe the kinds of operations (or transactions) that will be performed on the data.

# The design process

- **Final Phase** -- Moving from an abstract data model to the implementation of the database
  - Logical Design – Deciding on the database schema.
    - Database design requires that we find a “good” collection of relation schemas.
      - Business decision – What attributes should we record in the database?
      - Computer Science decision – What relation schemas should we have and how should the attributes be distributed among the various relation schemas?
  - Physical Design – Deciding on the physical layout of the database

# Design Approaches

- **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**

# 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 logical container for instances of an entity type.
  - 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)*
- A subset of the attributes form a **primary key** of the entity set; i.e., uniquely identifying each member of the set.

# Entity sets (instructor and student)

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

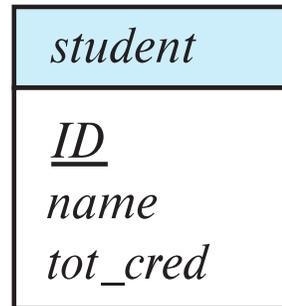
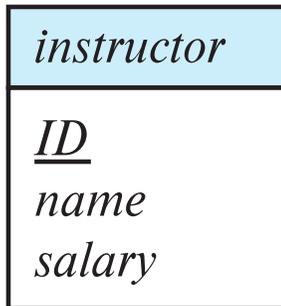
*instructor*

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)	<u>advisor</u>	22222 (Einstein)
<i>student</i> entity	relationship set	<i>instructor</i> entity

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

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

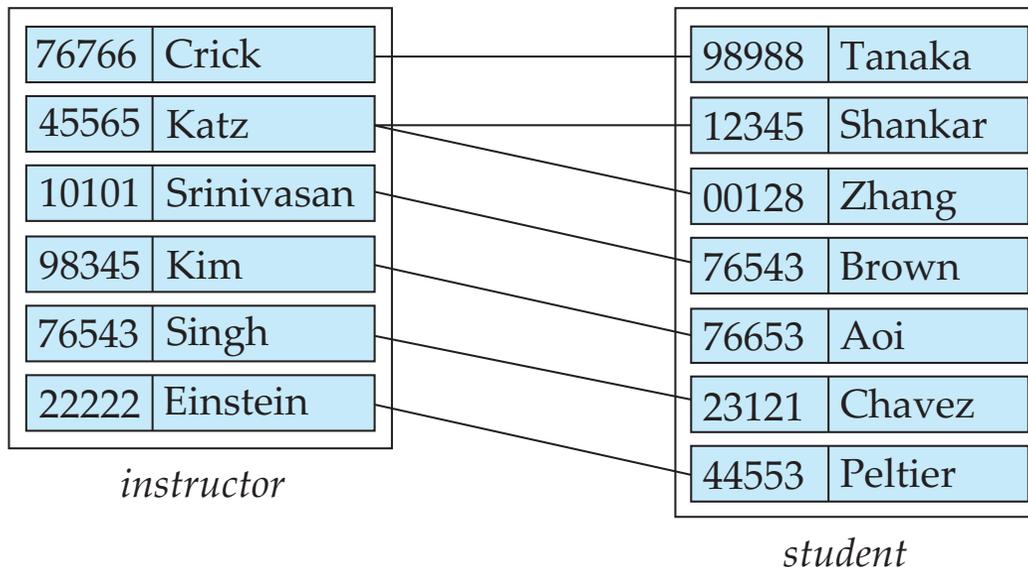
where  $(e_1, e_2, \dots, e_n)$  is a relationship

- Example:

$(44553, 22222) \in \text{advisor}$

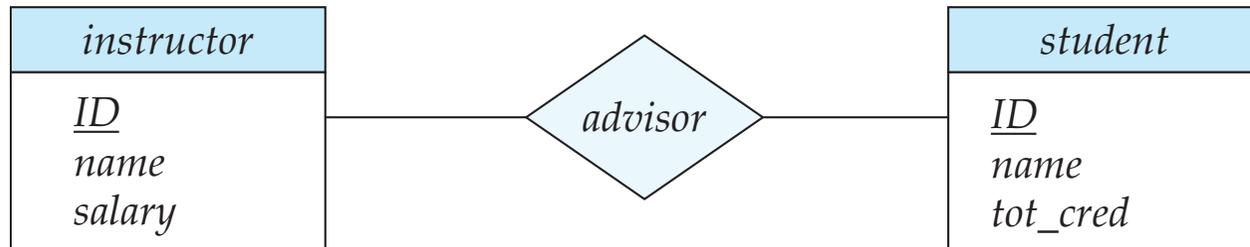
# Relationship Sets

- Example: we define the relationship set *advisor* to denote the associations between students and the instructors who act as their advisors.
- Pictorially, we draw a line between related entities.



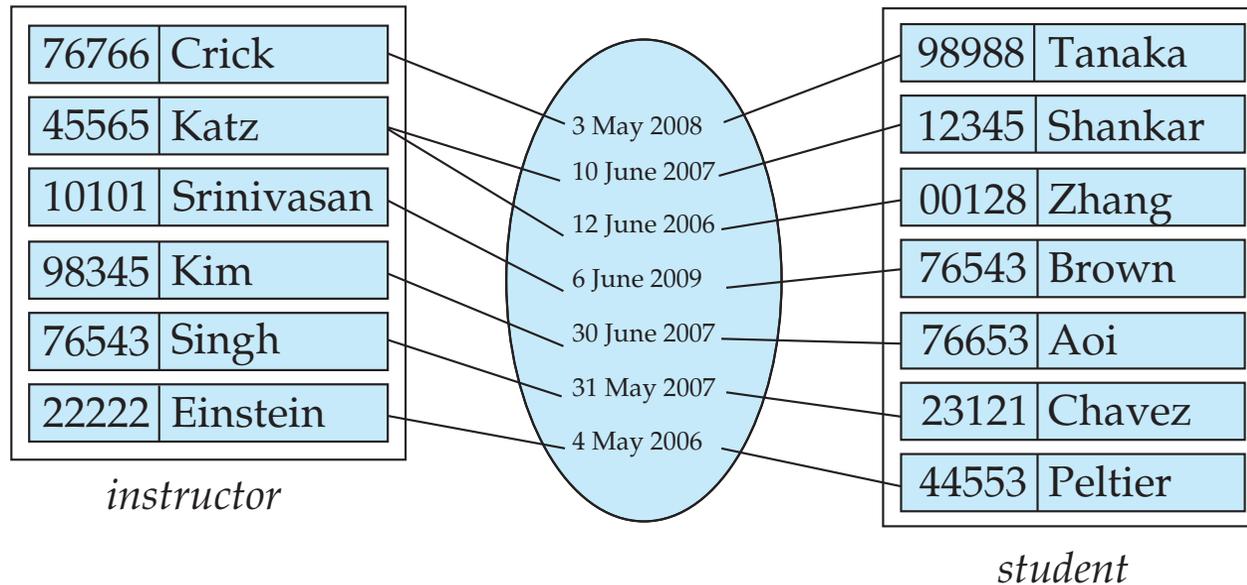
# Relationship Sets via ER Diagrams

- Diamonds represent relationship sets.

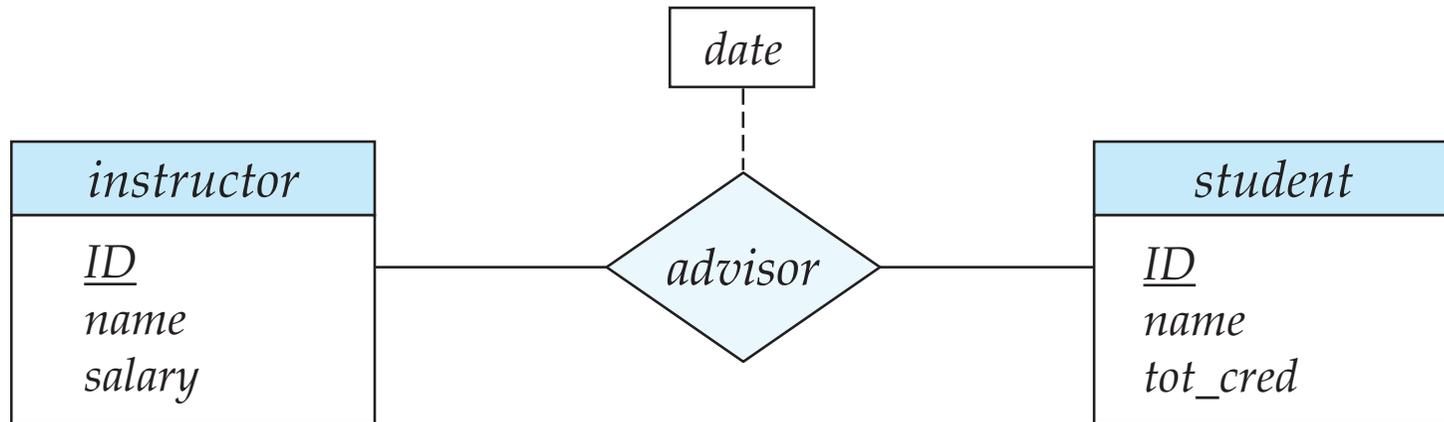


# Relationship Sets

- 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



# Relationship Sets

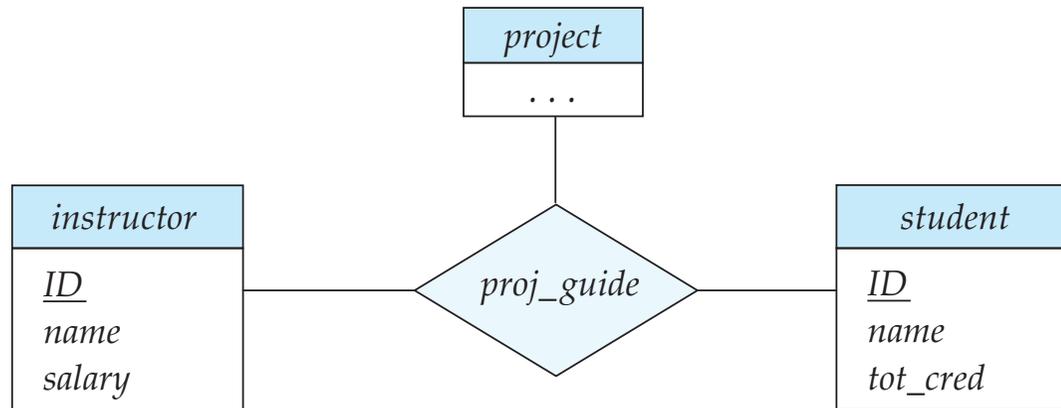


# Degree of a Relationship Set

- Binary relationship
  - involve two entity sets (or degree two).
  - most relationship sets in a database system are binary.
- Relationships between more than two entity sets are rare. Most relationships are binary. (More on this later.)
  - 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

- Most relationship sets are binary
- There are occasions when it is more convenient to represent relationships as non-binary.
- E-R Diagram with a Ternary Relationship



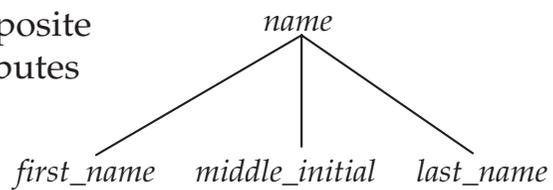
# Complex Attributes

- Attribute types:
  - **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

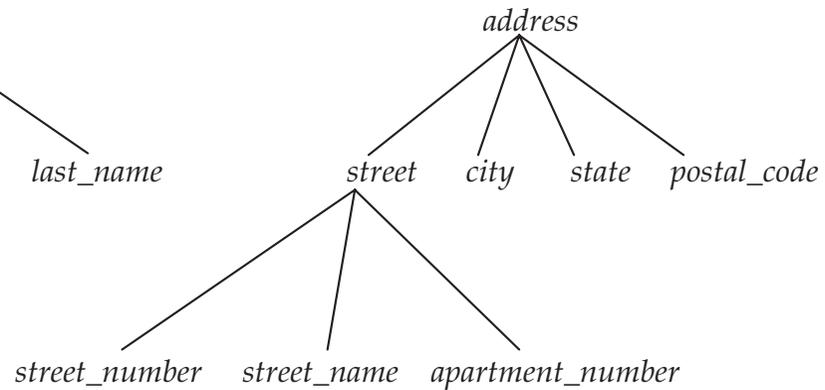
# Composite Attributes

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

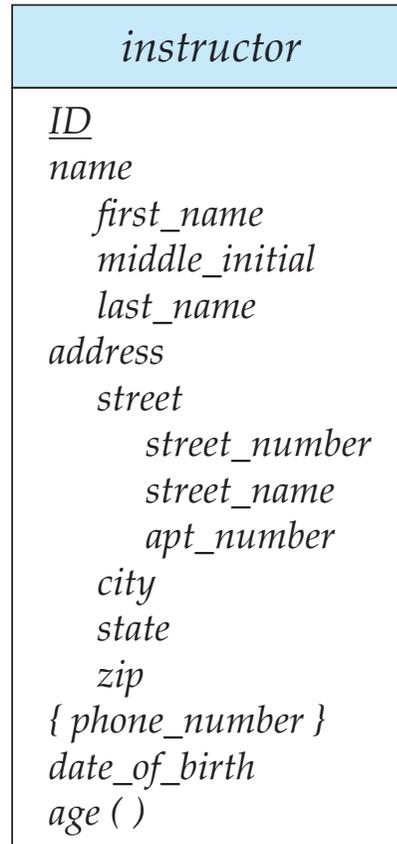
composite  
attributes



component  
attributes



# Representing Complex Attributes in ER Diagram



# Discussion