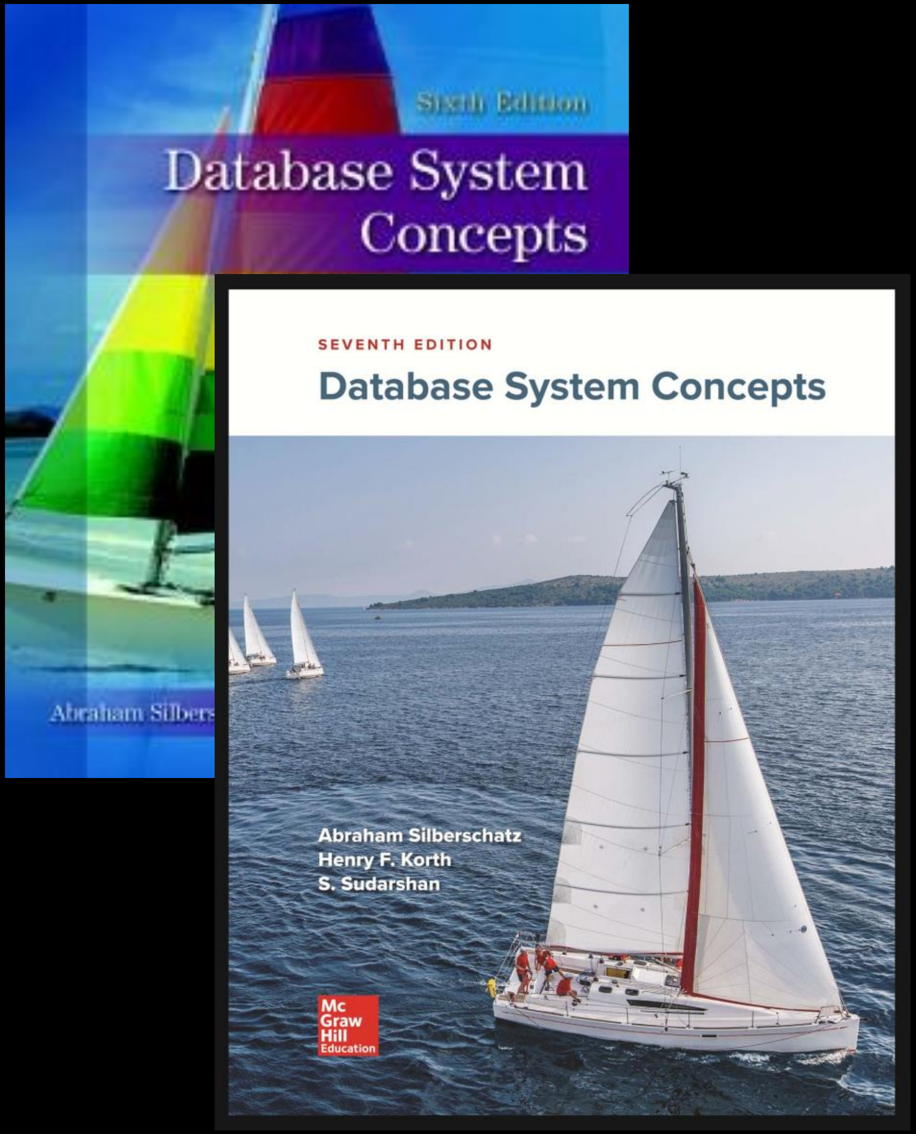
IF2240 – Basis Data

Database Design using E-R Model

Modified form Silberschatz's slides, Database System Concept, 7th edition



References

Abraham Silberschatz, Henry F. Korth, S. Sudarshan : "**Database System Concepts**", 6th Edition

- Chapter 7: Database Design and the E-R Model

Abraham Silberschatz, Henry F. Korth, S. Sudarshan : "**Database System Concepts**", 7th Edition

- Chapter 6: Database Design using E-R Model

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.
- Physical Design – Deciding on the physical layout of the database

Design Alternatives

In designing a database schema, we must ensure that we avoid two major pitfalls:

Redundancy: a bad design may result in repeat information Redundant representation of information may lead to data inconsistency among the various copies of information

Incompleteness: a bad design may make certain aspects of the enterprise difficult or impossible to model.

Design Approaches

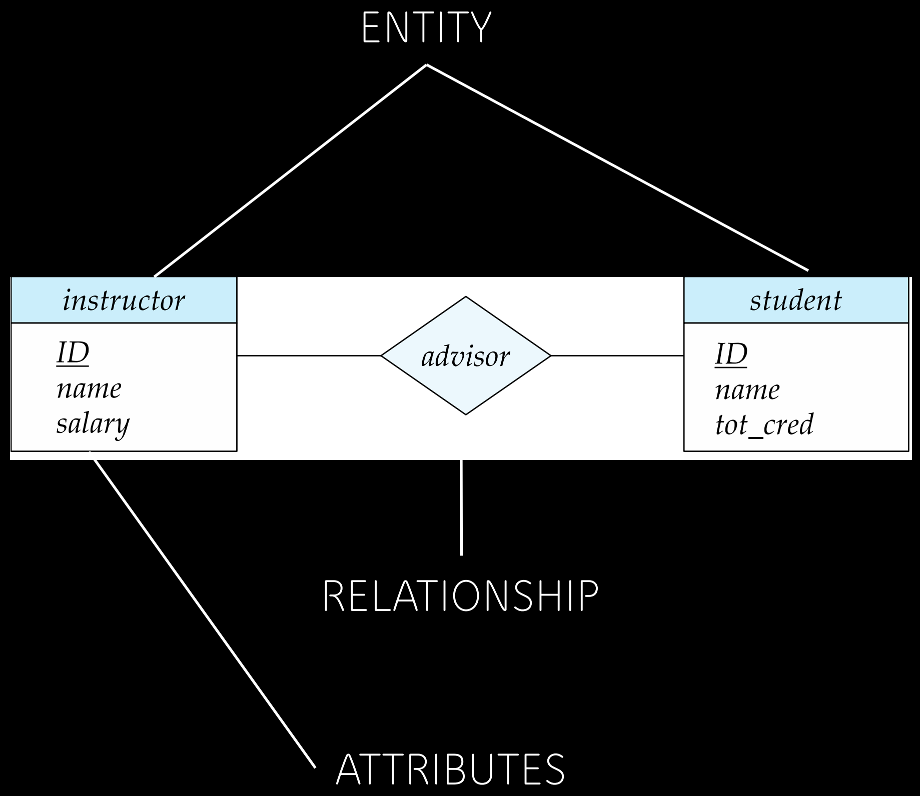
Entity Relationship Model (covered in this material)

- 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 (next topic)

- Formalize what designs are bad, and test for them

Entity-Relationship Model



ER model -- Database Modeling

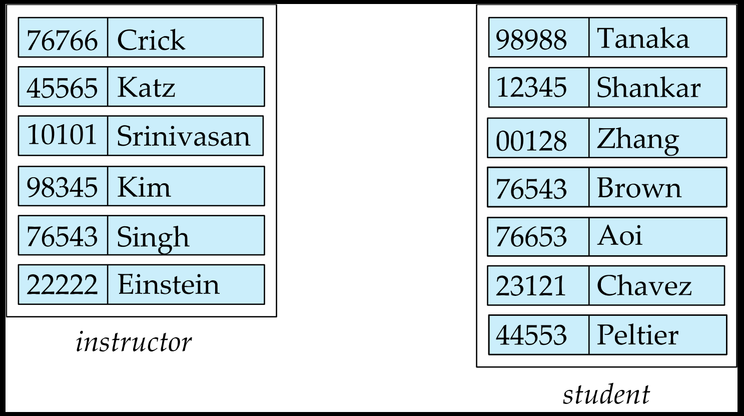
The ER data model was developed to facilitate database design by allowing specification of an **enterprise schema** that represents the overall logical structure of a database.

The ER data model employs three basic concepts:

- entity sets,
- relationship sets,
- attributes.

The ER model also has an associated diagrammatic representation, the **ER diagram**, which can express the overall logical structure of a database graphically.

Entity Sets

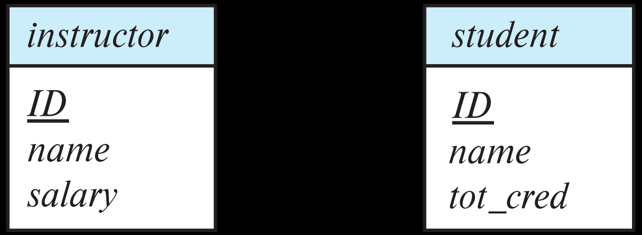


76766	Crick
45565	Katz
10101	Srinivasan
98345	Kim
76543	Singh
22222	Einstein

instructor

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

student



Entity sets can be represented graphically as follows:

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

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

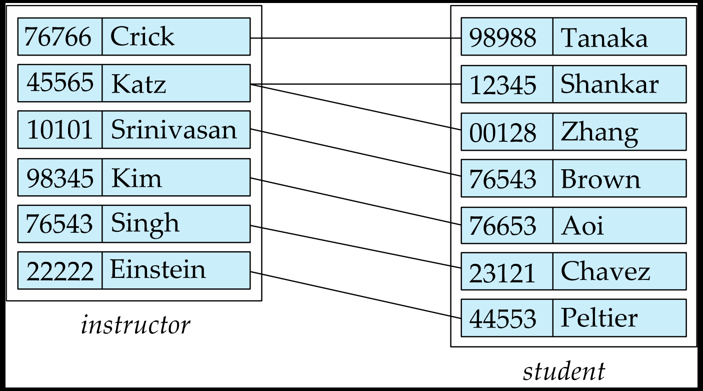
An entity is represented by a set of attributes; i.e., descriptive properties possessed by all members of an entity set.

- Example:

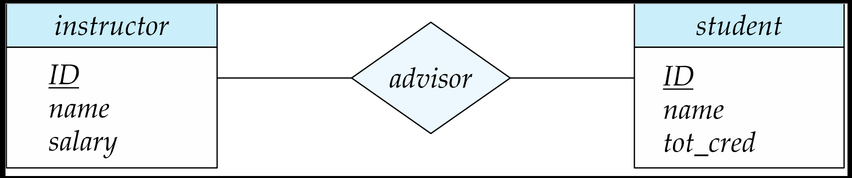
  instructor = (*ID*, *name*, *salary*)
  course= (*course_id*, *title*, *credits*)

- *Domain* – the set of permitted values for each attribute

A subset of the attributes form a **primary key** of the entity set; i.e., uniquely identifying each member of the set.



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



Diamonds represent relationship sets.

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

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

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

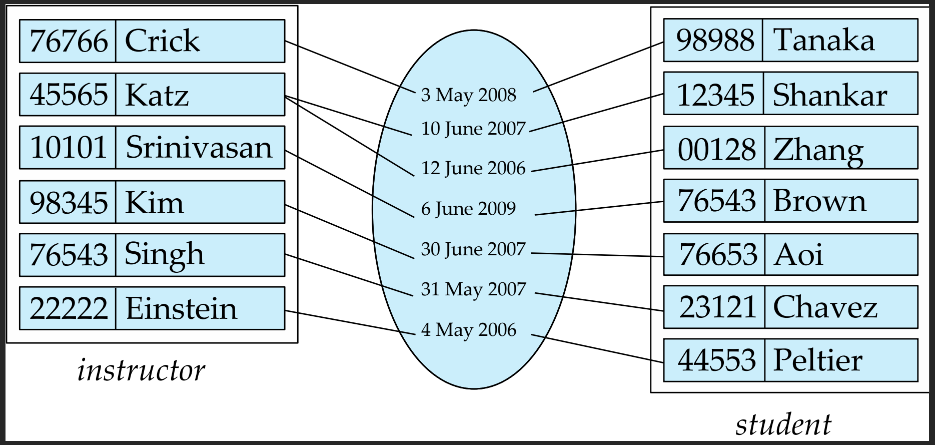
- Example:

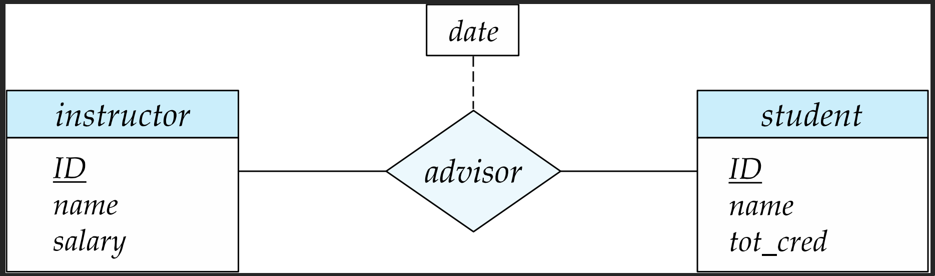
  (44553,22222) ∈ *advisor*

Relationship Sets with Attributes

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



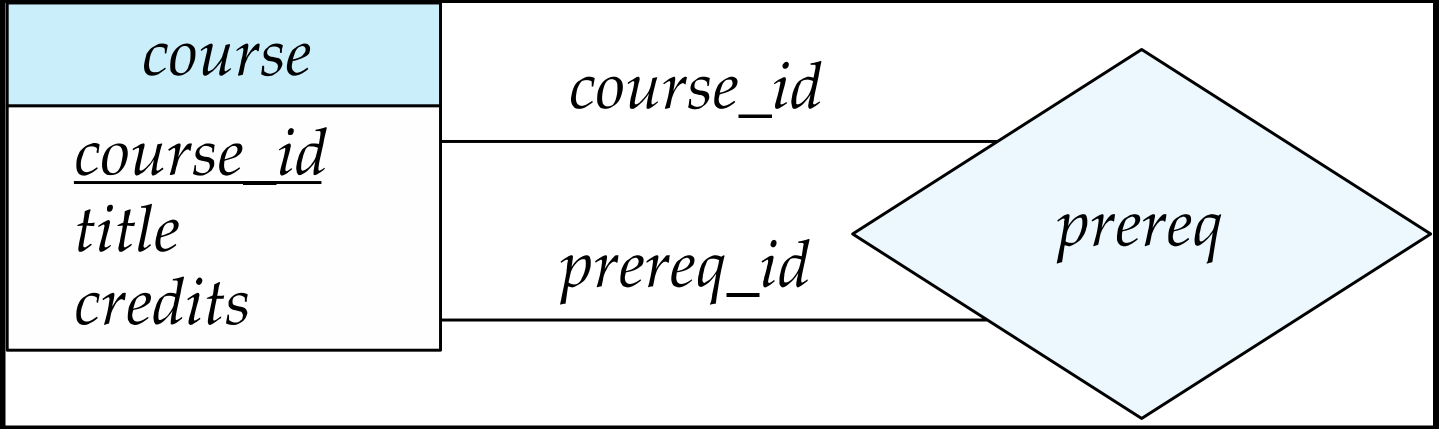


Roles

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**.



Degree of a Relationship Set

Binary Relationship

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

Non-Binary Relationship

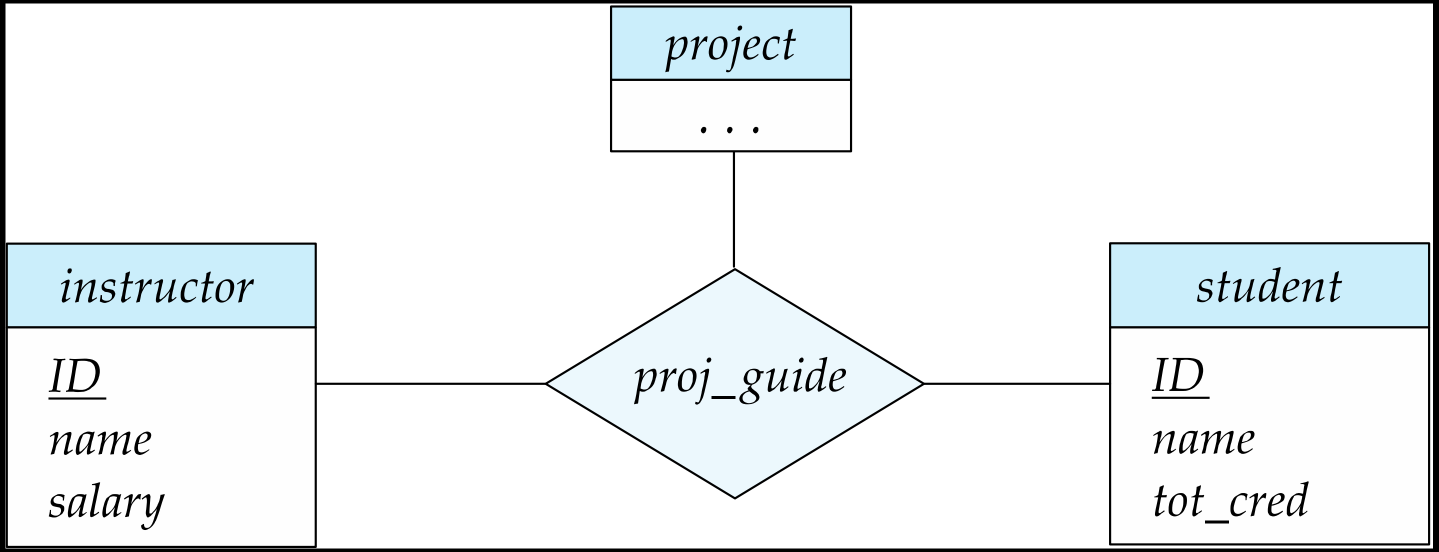
- Relationships between more than two entity sets are rare.
  - Example: *students* work on research *projects* under the guidance of an *instructor*. Relationship *proj_guide* is a ternary relationship between *instructor, student,* and *project*

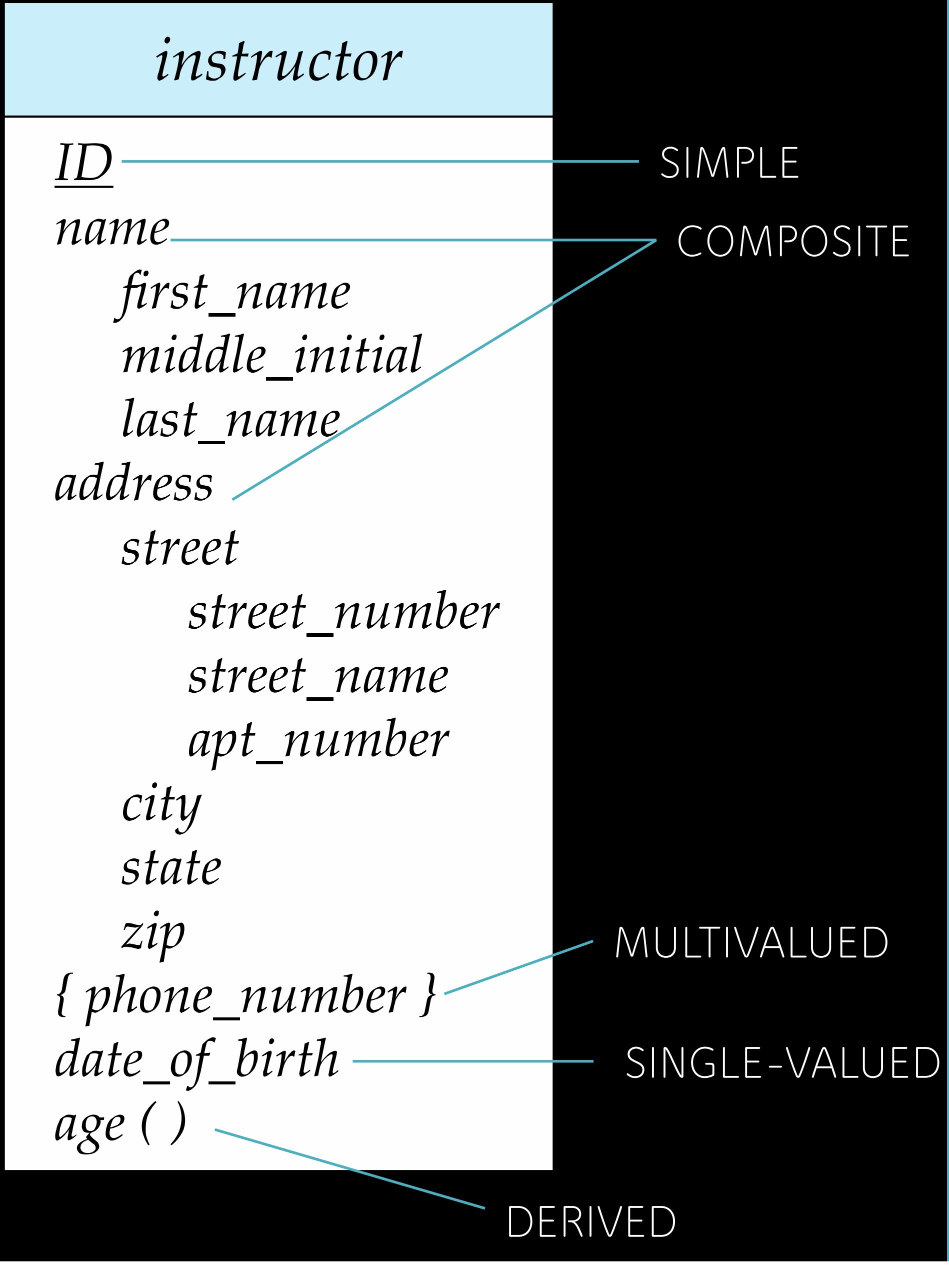
Non-binary Relationship Sets

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

KNOWLEDGE & SOFTWARE ENGINEERING

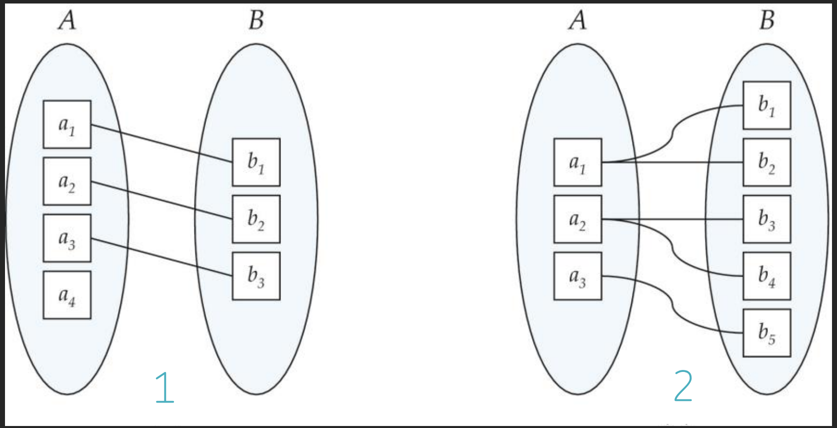
Mapping Cardinality Constraints

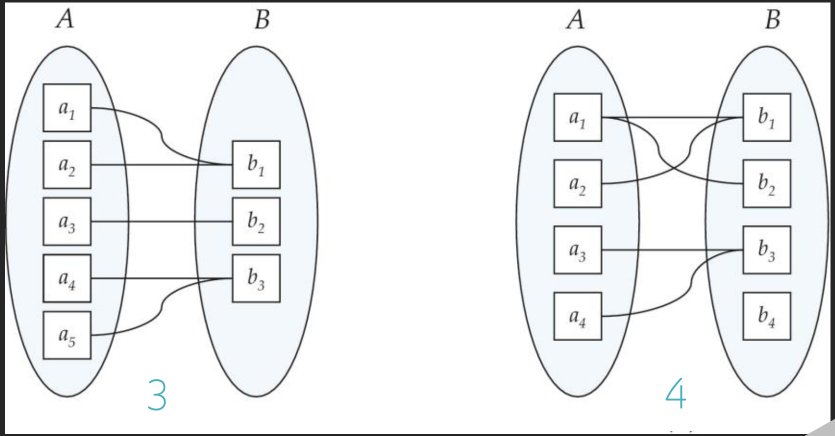
Express the number of entities to which another entity can be associated via a relationship set.

Most useful in describing binary relationship sets.

For a binary relationship set the mapping cardinality must be one of the following types:

1. One to one
2. One to many
3. Many to one
4. Many to many





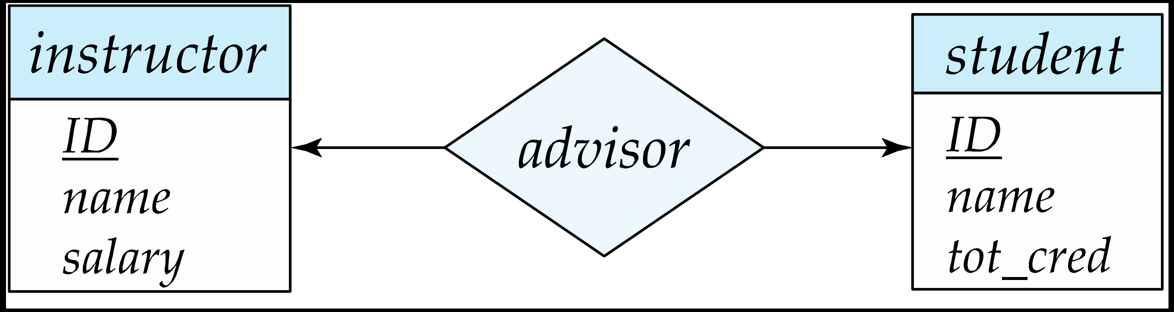
Note: Some elements in *A* and *B* may not be mapped to any elements in the other set

Representing Cardinality Constraints in ER Diagram (1/2)

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.

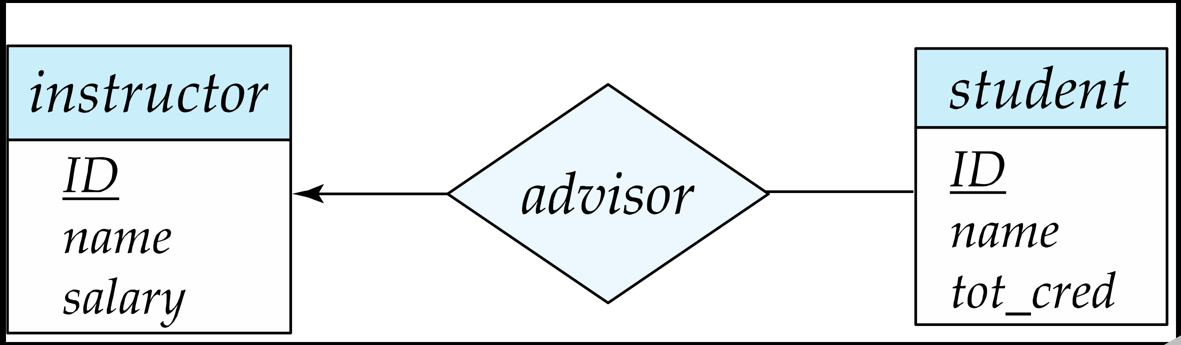
One-to-one relationship between an *instructor* and a *student* :

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



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,

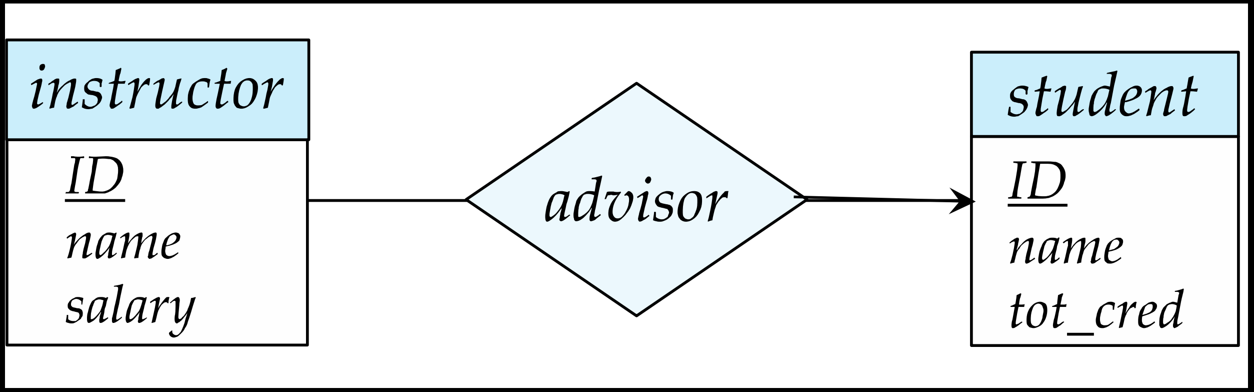


Representing Cardinality Constraints in ER Diagram (2/2)

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.

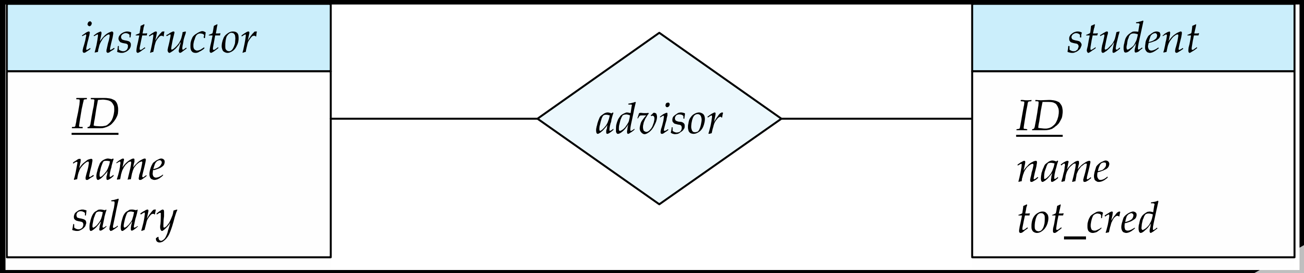
Many-to-one relationship between an instructor and a student,

- an instructor is associated with at most one student via advisor,
- and a student is associated with several (including 0) instructors via advisor



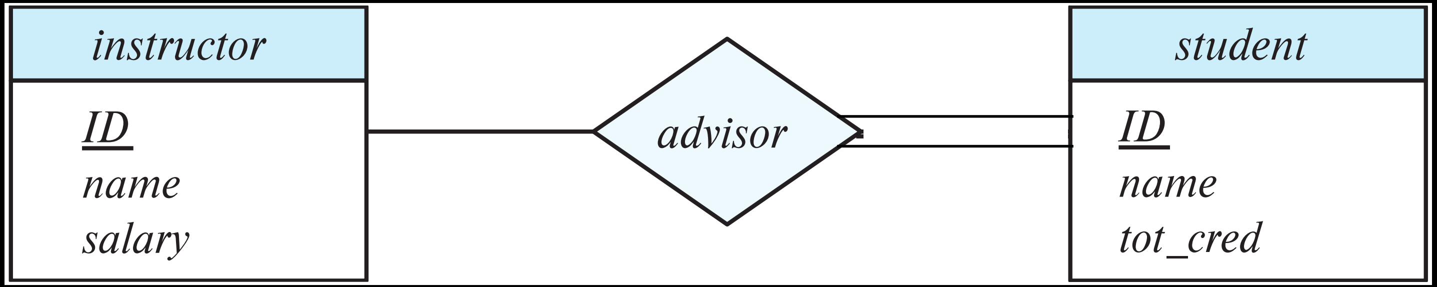
Many-to-many relationship between an instructor and a student

- 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

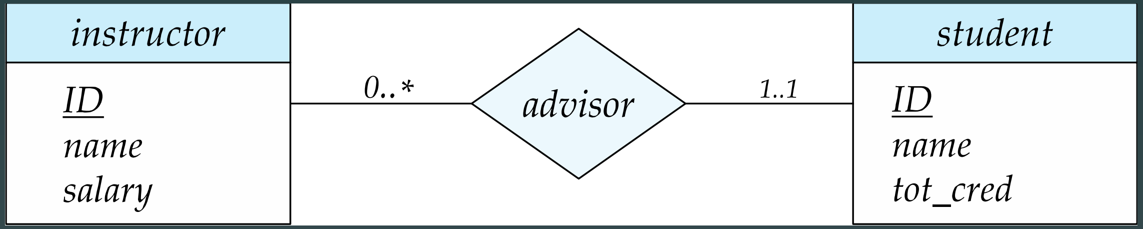
- **Total participation** (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set
- **Partial participation**: some entities may not participate in any relationship in the relationship set



- Example : participation of *student* in *advisor* relation is total
  - every *student* must have an associated instructor
- Example : participation of *instructor* in *advisor* relation is partial
  - some *instructor* may not have an associated student

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.
- Example



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

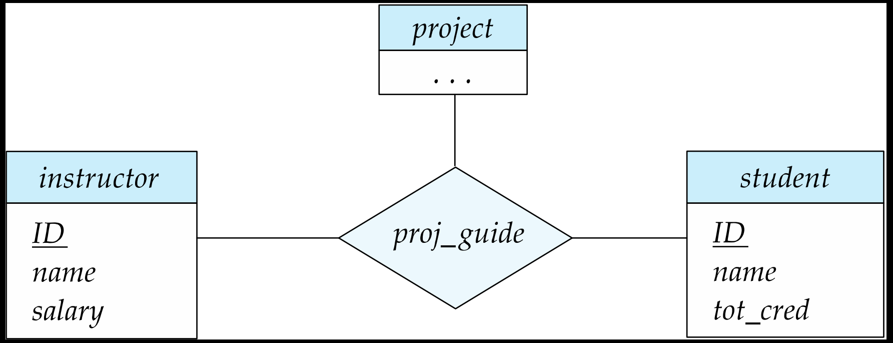
Cardinality Constraints on Ternary Relationship

We allow at most one arrow out of a ternary (or greater degree) relationship to indicate a cardinality constraint

For example, 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.

- For example, 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 outlaw more than one arrow



Primary Key

Primary keys provide a way to specify how entities and relations are distinguished. We will consider:

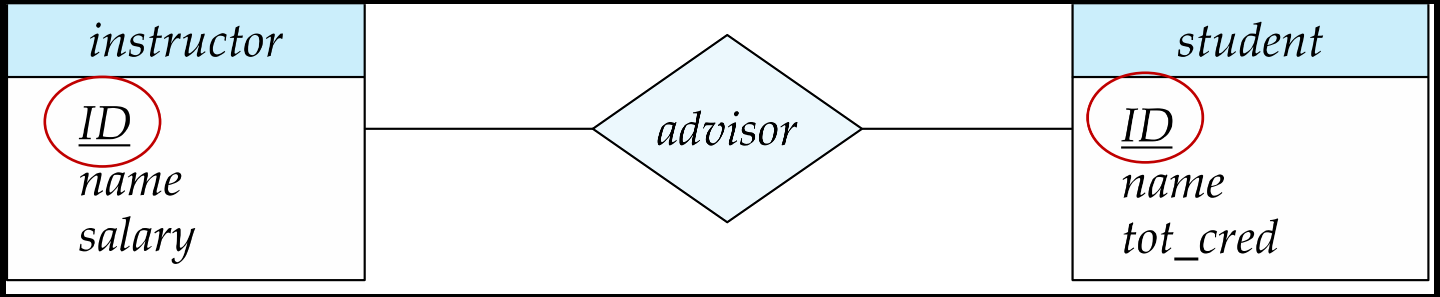
- Entity sets
- Relationship sets.
- *later on.. Weak entity sets*

By definition, individual entities are distinct.

The values of the attribute values of an entity must be such that they can uniquely identify the entity.

- No two entities in an entity set are allowed to have exactly the same value for all attributes.

A key for an entity is a set of attributes that suffice to distinguish entities from each other



Primary Key

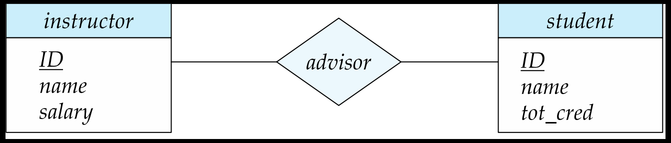
Primary keys provide a way to specify how entities and relations are distinguished. We will consider:

- Entity sets
- Relationship sets
- *later on.. Weak entity sets*

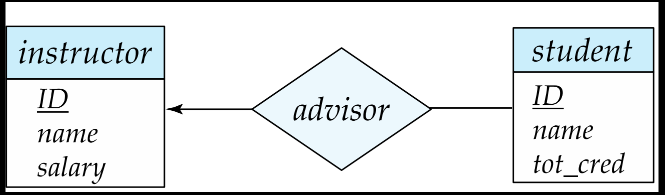
To distinguish among the various relationships of a relationship set we use the individual primary keys of the entities in the relationship set.

The choice of the primary key for a relationship set depends on the mapping cardinality of the relationship set.

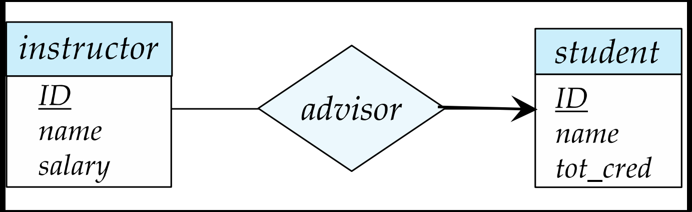
Many-to-Many relationships. The preceding union of the primary keys is a minimal superkey and is chosen as the primary key.



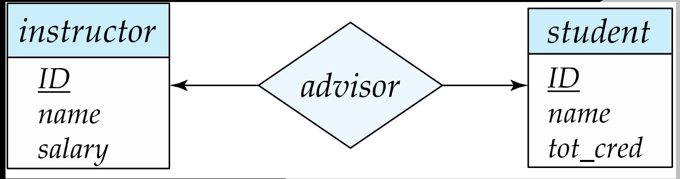
One-to-Many relationships. The primary key of the "Many" side is a minimal superkey and is used as the primary key.



Many-to-one relationships. The primary key of the "Many" side is a minimal superkey and is used as the primary key.



One-to-one relationships. The primary key of either one of the participating entity sets forms a minimal superkey, and either one can be chosen as the primary key.



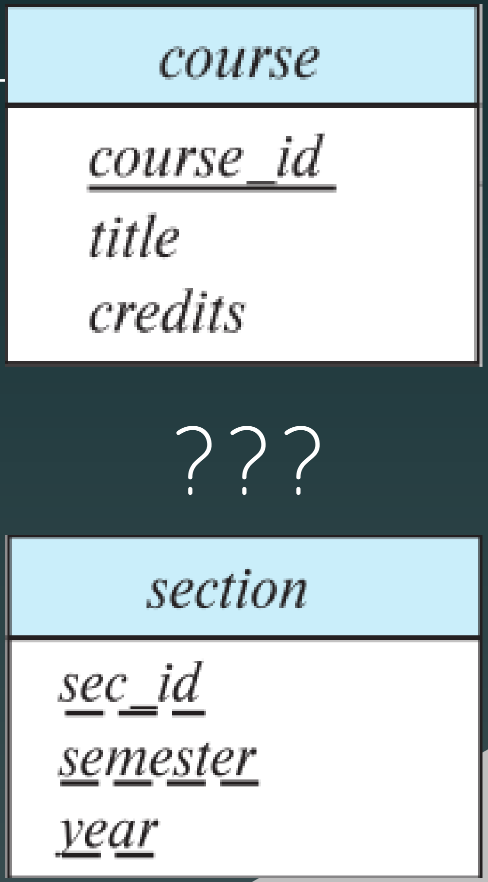
Weak Entity Sets

Consider a *section* entity, which is uniquely identified by a *course_id, semester, year*, and *sec_id*.. Clearly, section entities are related to course entities. Suppose we create a relationship set *sec_course* between entity sets *section* and *course*.. Note that the information in *sec_course* is redundant, since *section* already has an attribute *course_id*, which identifies the course with which the section is related. One option to deal with this redundancy is to get rid of the relationship *sec_course*; however, by doing so the relationship between *section* and *course* becomes implicit in an attribute, which is not desirable. An alternative way to deal with this redundancy is to not store the attribute course_id in the section entity and to only store the remaining attributes section_id, year, and semester. However, the entity set section then does not have enough attributes to identify a particular section entity uniquely

To deal with this problem, we treat the relationship sec_course as a special relationship that provides extra information, in this case, the course_id, required to identify section entities uniquely.

A weak entity set is one whose existence is dependent on another entity, called its identifying entity

Instead of associating a primary key with a weak entity, we use the identifying entity, along with extra attributes called discriminator to uniquely identify a weak entity.



Weak Entity Sets (Cont.)

An entity set that is not a weak entity set is termed a strong entity set.

Every weak entity must be associated with an identifying entity; that is, the weak entity set is said to be existence dependent on the identifying entity set.

The identifying entity set is said to own the weak entity set that it identifies.

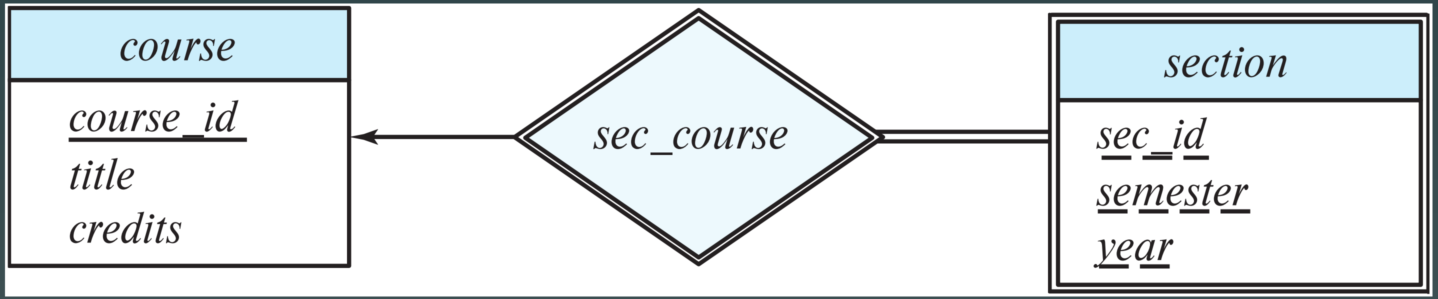
The relationship associating the weak entity set with the identifying entity set is called the identifying relationship.

In E-R diagrams, a weak entity set is depicted via a double rectangle.

We underline the discriminator of a weak entity set with a dashed line.

The relationship set connecting the weak entity set to the identifying strong entity set is depicted by a double diamond.

Primary key for section – (course_id, sec_id, semester, year)


course
course_id
title
credits
sec_course
section
sec_id
semester
year