

# CSE460/560 DATA MODELS AND QUERY LANGUAGES

Entity-Relationship Data Model

Cheng-En Chuang

(Slides Adopted from Jan Chomicki and Ning Deng)



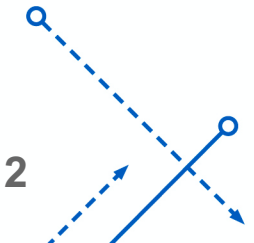
University at Buffalo

Department of Computer Science  
and Engineering

School of Engineering and Applied Sciences

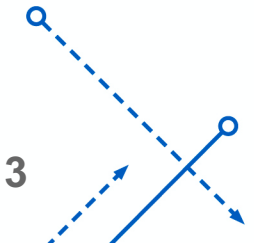
# Announcement

- Remember to enroll on Piazza
- Project 1 A Group Project
  - 3 – 5 people
    - Email the TA your group members (ubit & person#)
  - Hopefully will be published before next lecture (Feb. 4)



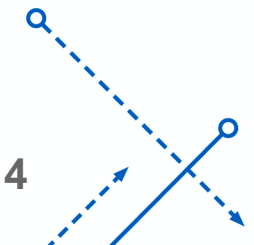
# Outline

1. **Level of Abstraction**
2. ER Model Introduction
3. ER Model Basics
4. Conceptual Design in ER Model



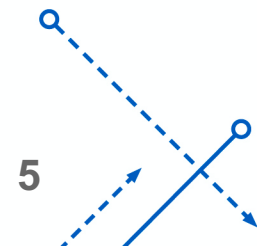
# Level of Abstraction

- Ideas
- High-Level Design (ER Model)
- Relational Database Schema (Relational Schema)
  - The global view of the whole application
  - Uses a high-level data mode, e.g. relational
  - Created and maintained by the database administrator
- Relational DBMS
  - How the database is actually stored on disk(s)
  - Hidden from users
  - Uses low-level concepts: files, indexes, ...
  - Created and maintained by the DBA with high-level tools



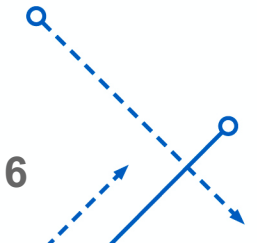
# Outline

1. Level of Abstraction
- 2. ER Model Introduction**
3. ER Model Basics
4. Conceptual Design in ER Model



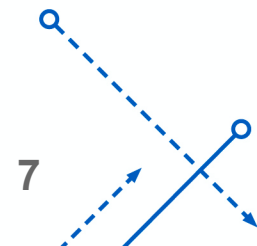
# ER Model Introduction

- Scheme
  - Structure of data
- How to produce a schema?
  - Data represents concepts
  - Model the concepts



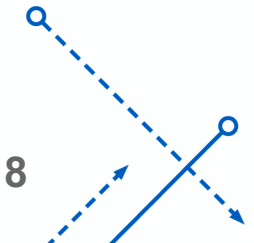
# ER Model Introduction

- A pictorial (graphical) representation of a schema
  - Enumerates all entities in the schema
  - Shows how entities are related
  - Shows what is stored for each entity
  - Shows restrictions (integrity constraints)



# Outline

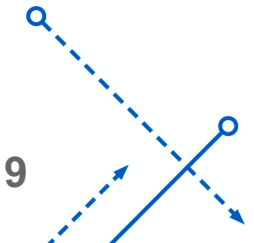
1. Level of Abstraction
2. ER Model Introduction
- 3. ER Model Basics**
4. Conceptual Design in ER Model





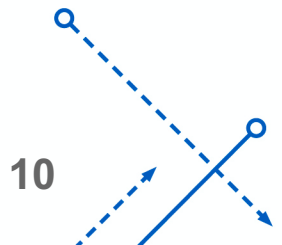
# ER Model Basics – Principle Elements Types

- Entity sets
- Attributes
- Relationships



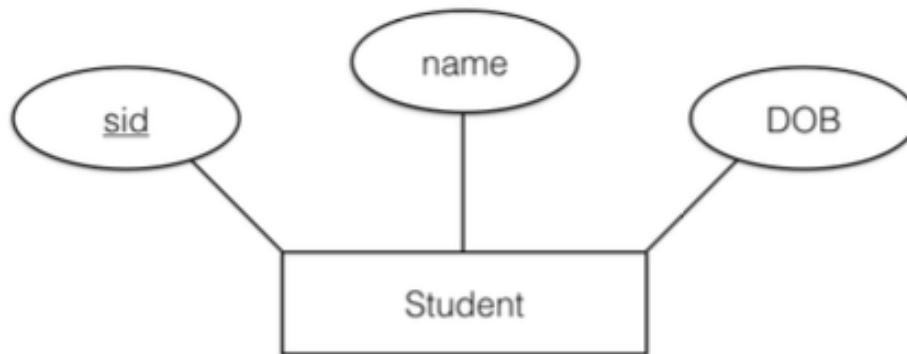
# ER Model Basics – Entity

- A abstract object of some sort
  - a concrete one: Apartments
  - an abstract one: Moods
- An entity is described through a set of attributes
  - The properties of the entity



# ER Model Basics – Entity Set, Attributes

- A collection of similar entities (e.g. students)
  - Represented by *rectangles* in ER diagrams
  - Entities in an entity set have the same set of attributes
- An attribute is a property of the entities in an entity set
  - Represented by *ovals* in ER diagrams
  - Each attribute has a domain (e.g. integers, string)

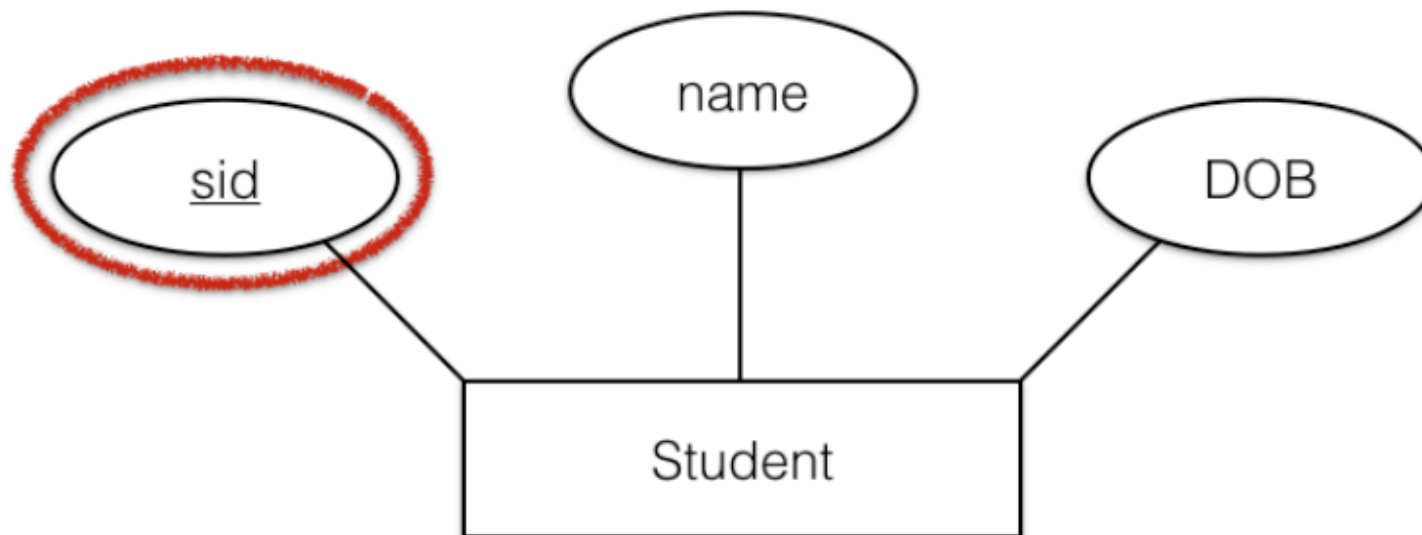


# ER Model Basics – Domain of attributes

Type class	Description	Example Entities
Primitive	A value that can not be subdivided further	1701, 2.4, or 'John Smith'
Tuple	A fixed-length "struct" of attribute-value pairs (attribute names optional)	<Name: 'John', ID:1234>,
Collection	A collection of variable number of records	['John', 'Mary'] [<'John', 1234>, <'Mary', 2234> ...]

# ER Model Basics – Key Constraint

- A **key** for an entity set  $E$  is a set  $K$  of one or more attributes such that
  - Given any two distinct entities  $e_1, e_2 \in E$ ,  $e_1$  and  $e_2$  cannot have **identical values** for each of the attributes in the key  $K$
  - Example



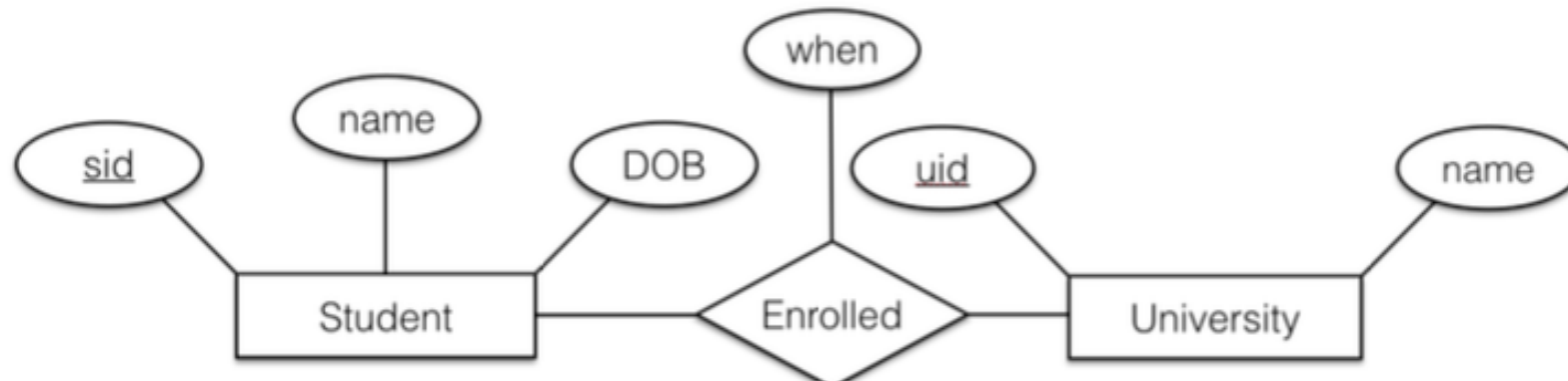
# ER Model Basics – Key Constraints

- Some important points of key constraints
  - Every entity **must** have a key
    - Although in some cases the key actually belongs to another entity set
    - E.g. ISA-Hierarchies, weak entity sets
  - There might be more than one candidate key for an entity set
    - Pick one from candidate keys as **primary key**, act as if it is the only key
  - Example?



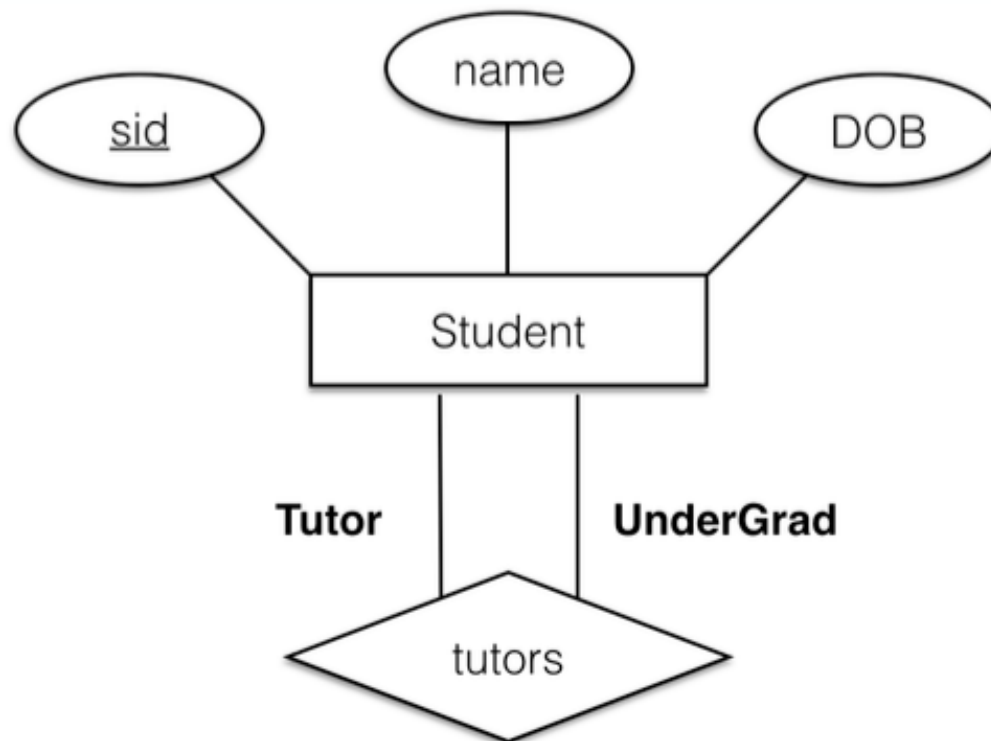
# ER Model Basics – Relationships

- Relationship
  - connections among two or more entity sets
- Relationship Set
  - A collection of similar relationships
    - Represented by diamonds in ER diagrams
  - An *n-ary relationship* relates  $n$  entity sets  $E_1, \dots, E_n$
  - May have their own attributes



## ER Model Basics – Relationships

- There can be relationships between entities in the same entity sets
  - Any other examples?



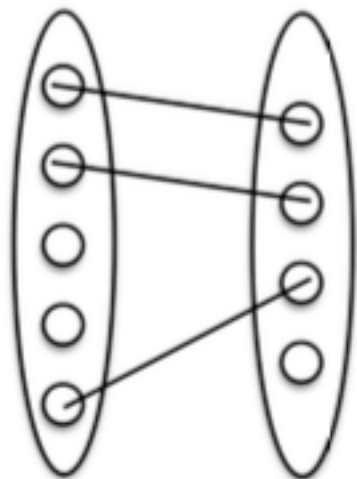


# ER Model Basics – Multiplicity

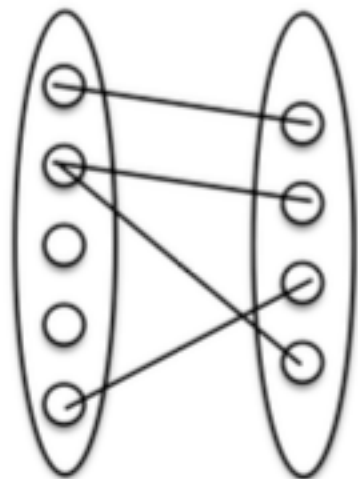
- Consider if we have the following relationships
  - 1-to-1
    - One university can have only one president
  - 1-to-many
    - One university can have many students
    - But each student has only one university
  - Many-to-many
    - A tutor can tutor many students
    - A student can have many tutors



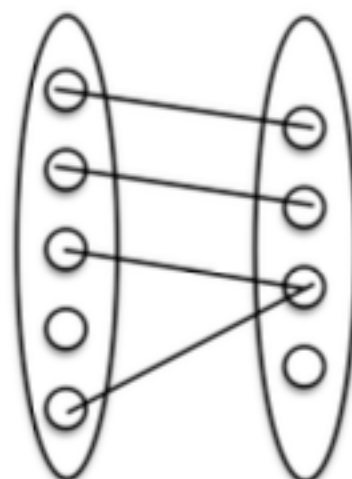
# ER Model Basics – Multiplicity



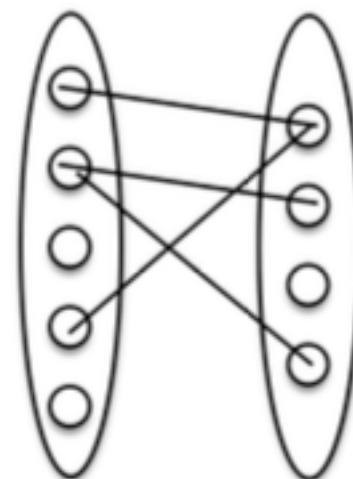
1-to-1



1-to-Many



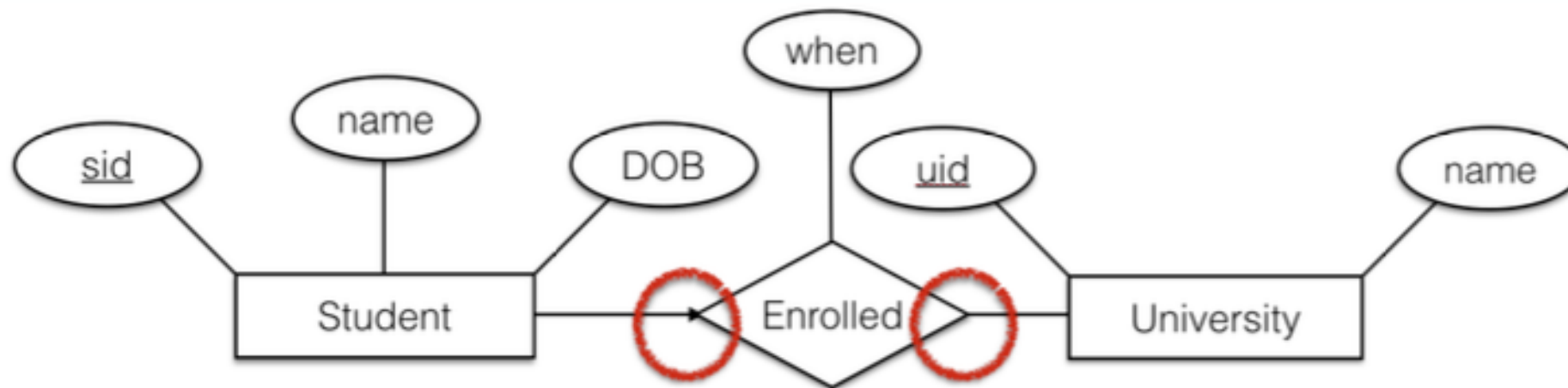
Many-to-1



Many-to-Many

# ER Model Basics – Multiplicity in Binary Relationships

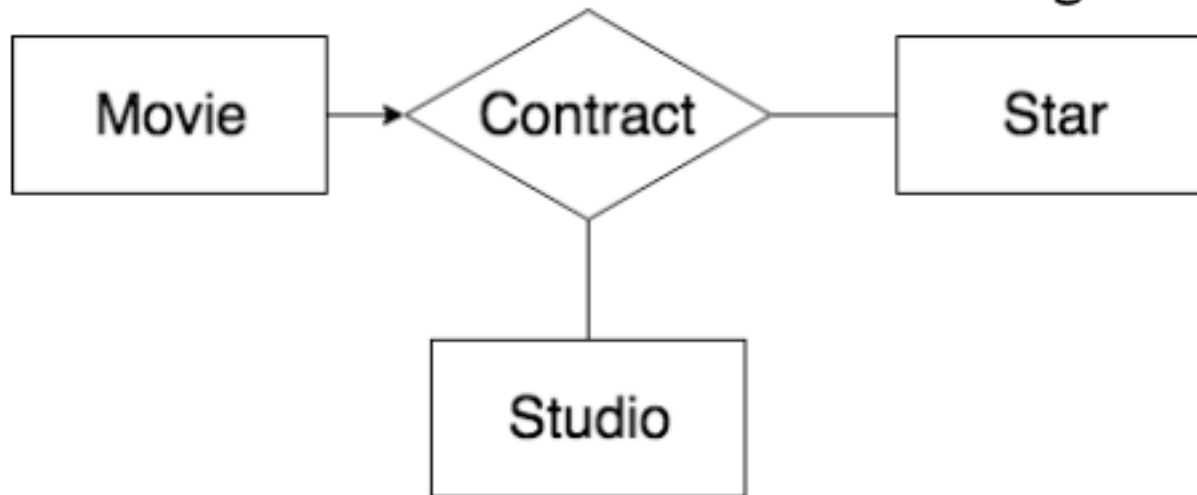
- "one-X" in binary relationship
  - Identifies entities that participate in **at most once**
    - Denote that "many" side with an *entering arrow*
  - There are many other notations for multiplicity constraints
  - What happen when we remove a student or an university?



# ER Model Basics – Multiplicity in Multiway Relationships

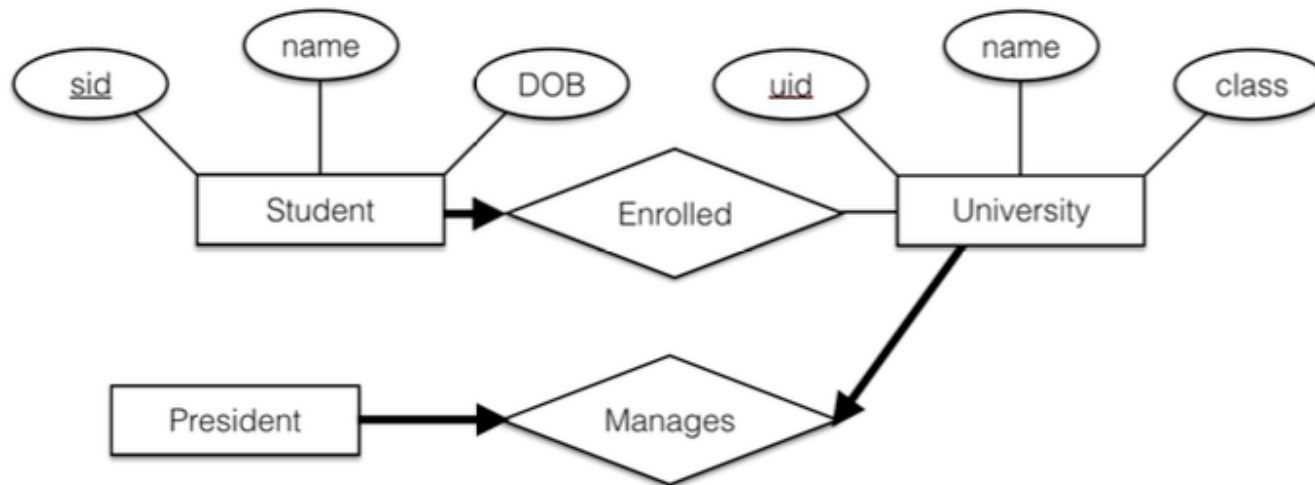
- A ternary relationship
- What are the pros and cons in this schema?

*We denote the 'one' side with an entering arrow*



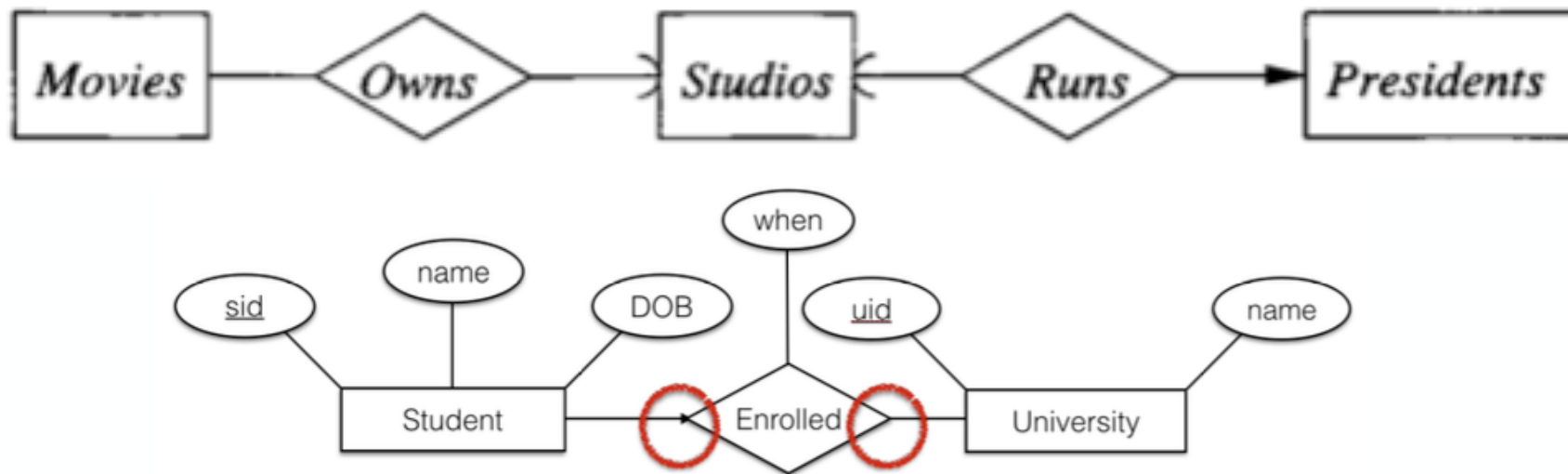
## ER Model Basics – Participation Constraints

- Require participation in a relationship, denoted as bold line in ER diagrams
  - Every student must be enrolled in an university
  - Every president must manage a univerisy
  - Every university must have a president
  - Any issues?



## ER Model Basics – Referential Integrity

- The referenced entity **must exist in the instance** of the ER diagram
  - There is a president -> There must be a studio
    - Not vice versa
  - Note the different targets of the arrow



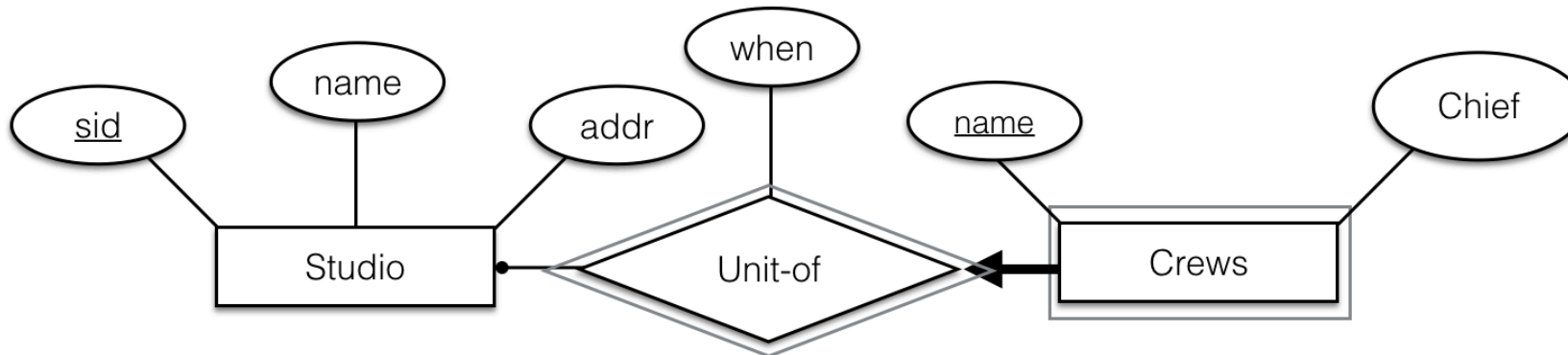
# ER Model Basics – Degree Constraints

- Indicating limits on the number of entities that can be connected to any one entity of the related entity set



## ER Model Basics – Weak Entities

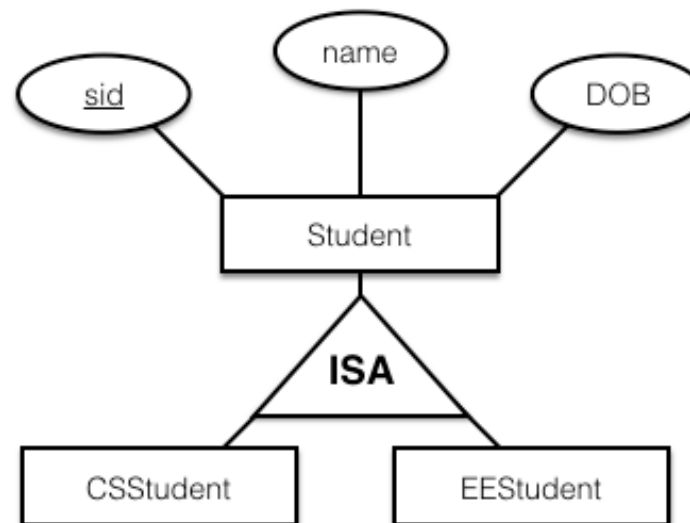
- Some of all of the attributes in an entity set's key belongs to another entity set
- Identified only to the primary key of another (owner) unique entity
  - Sounds familiar? What kind of relationship?
  - The weak entity must participate in a one-to-many relationship
- If there is a studio? If there is a crew?





# ER Model Basics – ISA Hierarchies

- Define entity inheritance
  - By **A ISA B**, then every A is also considered to be a B
  - Every ISA relationship is one-to-one, represented by triangle in ER diagram
- Overlap constraints
  - Can a student be a CSStudent and an EESStudent?
- Covering constraints
  - Does every student have to be a CSStudent or an EESStudent?

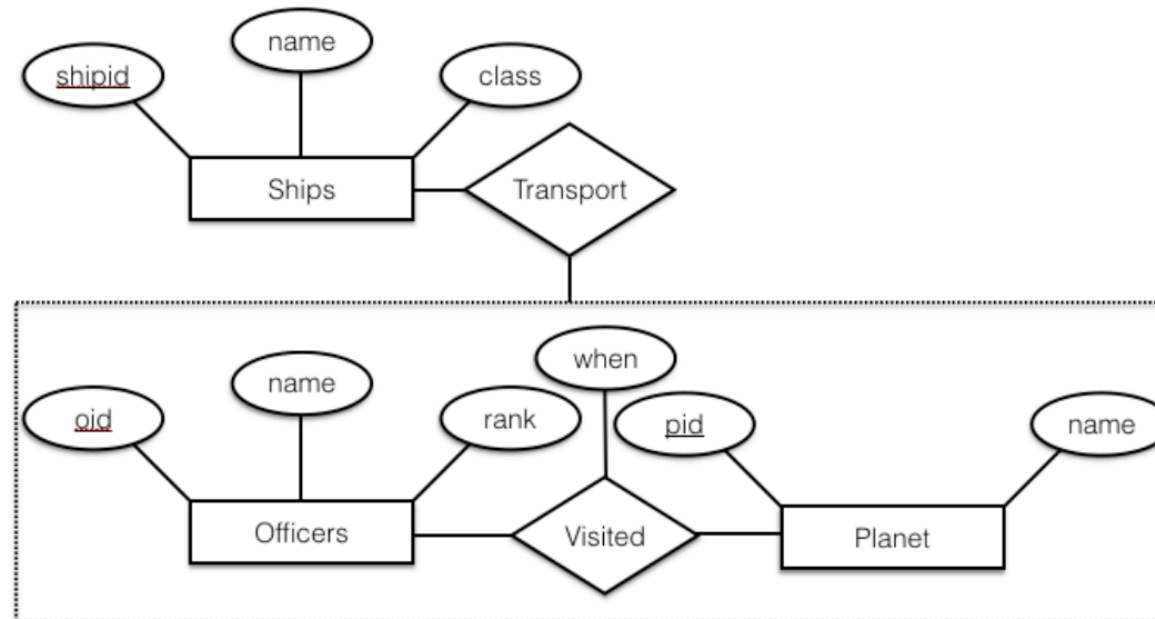


# ER Model Basics – ISA Hierarchies

- Reasons for using ISA
  - Adding descriptive attributes specific to a subclass, e.g. CSEIT
  - Identifying entities in a specific type of relationship
    - A relationship only connected to a subset of an entity set
    - Examples?

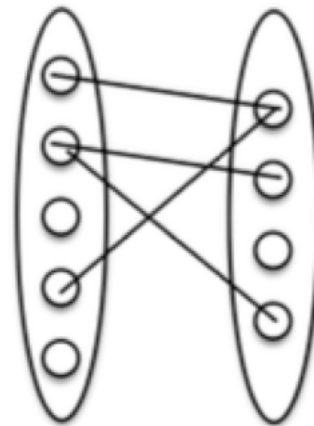
# ER Model Basics – Aggregation

- Allows us to treat a relationship as an entity set
- What does it imply?



## ER Model Basics – More on Relationships

- Many-to-Many can be decoupled into ?
  - Two 1-to-Many relationships plus an new entity set



Many-to-Many

# Outline

1. Level of Abstraction
2. ER Model Introduction
3. ER Model Basics
- 4. Conceptual Design in ER Model**

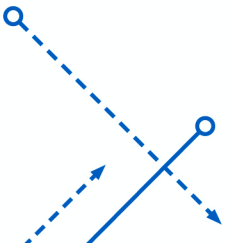
# Conceptual Design in ER Model

- Murphy's Law: Anything that can go wrong will go wrong
  - Design a foolproof DB starting from ER model
- Design choices
  - Should a concept be modeled as an entity or an attribute of another entity?
    - What is the difference?
  - Should a concept be model as an entity or a relationship between entities?
    - A noun? Or a verb?
  - What kind of relationship?
    - Binary, Ternary, N-ary?
- Constraints:
  - A lot of data semantics can (and should) be captured
  - Not all constraints are expressible in ER diagrams

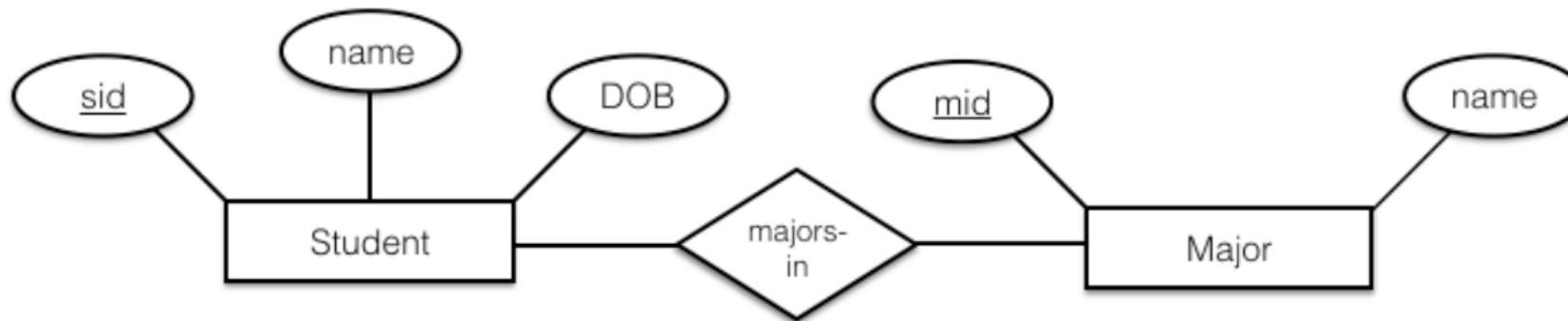


# Conceptual Design in ER Model

- Example
  - Expressing the Major of a student
    - Option 1: Major as an attribute of Students
    - Option 2: Majors entity set and a relationship set *majors-in*
  - Vote your option
  - Depends on semantics
    - Can a student have multiple majors?
    - Are there another detail we want beside major's name?



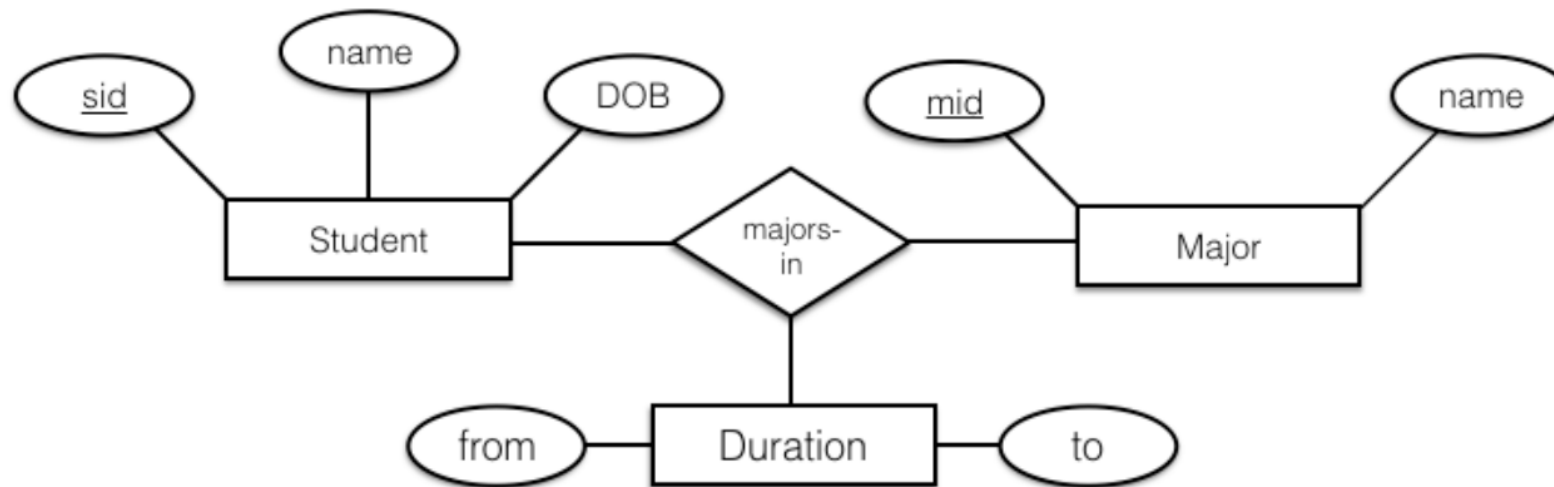
# Conceptual Design in ER Model



**Problem:** No details on relationship *majors-in*



## Conceptual Design in ER Model



**Solution:** Add a duration entity and make *majors-in* a ternary relationship.

📍 Is this good enough? Can we do better?

# Recommended Reading

Database Systems: The Complete Book  
Chapter 4.1 – 4.4