

CSE 403

Software Engineering

Spring 2023

#10: Data modelling

Logistics

WEEK 4

04/17 L: Data modeling

04/18 T: DUE: [GPS!!!](#)

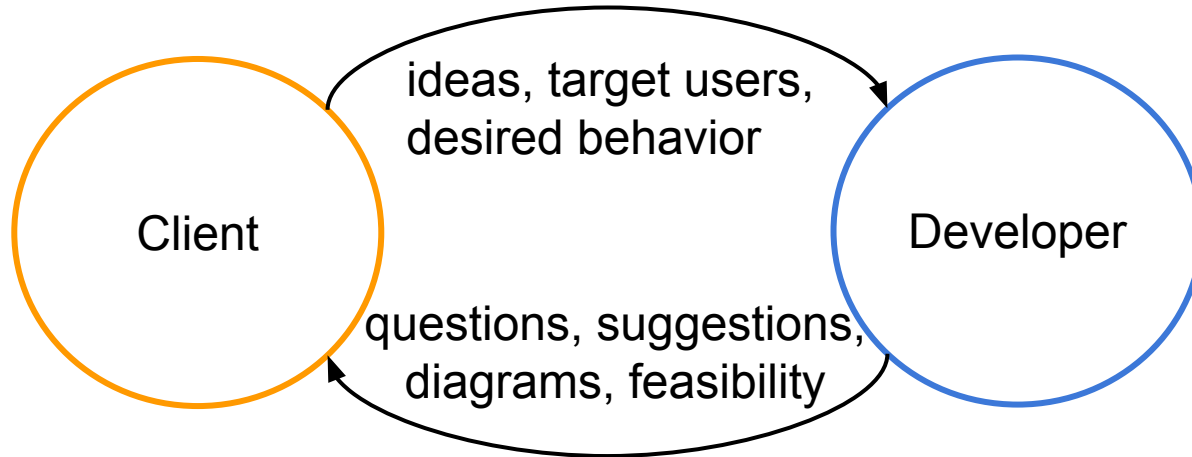
04/19 L: Architecture [Design & Architecture \(DnA\)](#)

04/20 P:

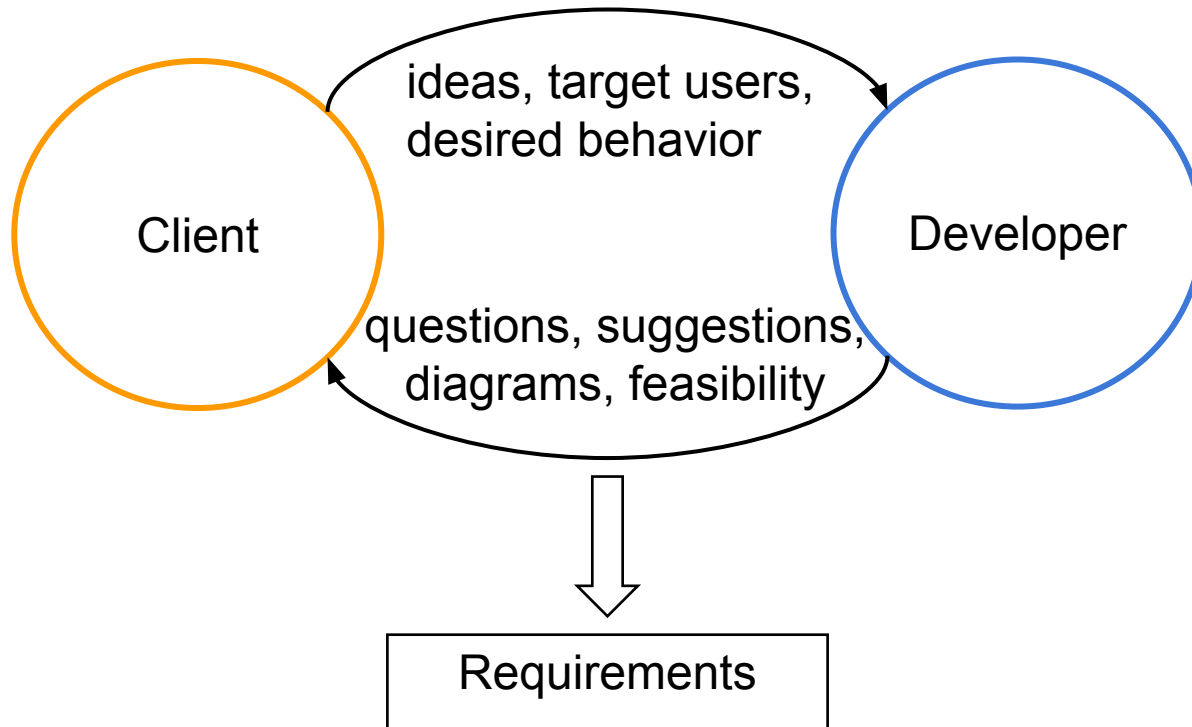
04/21 L: Design

From Requirements to System Design

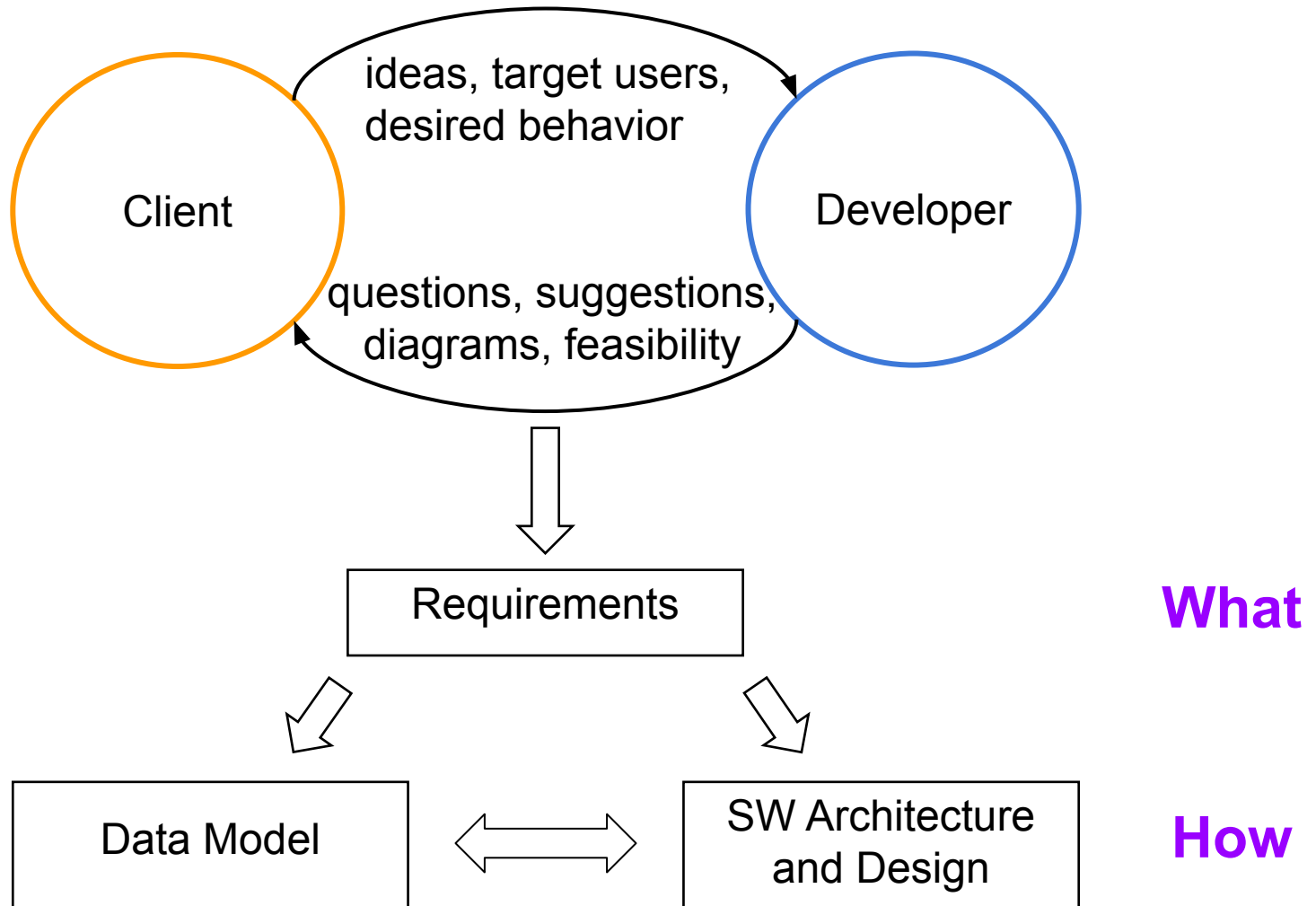
From Requirements to System Design



From Requirements to System Design



From Requirements to System Design



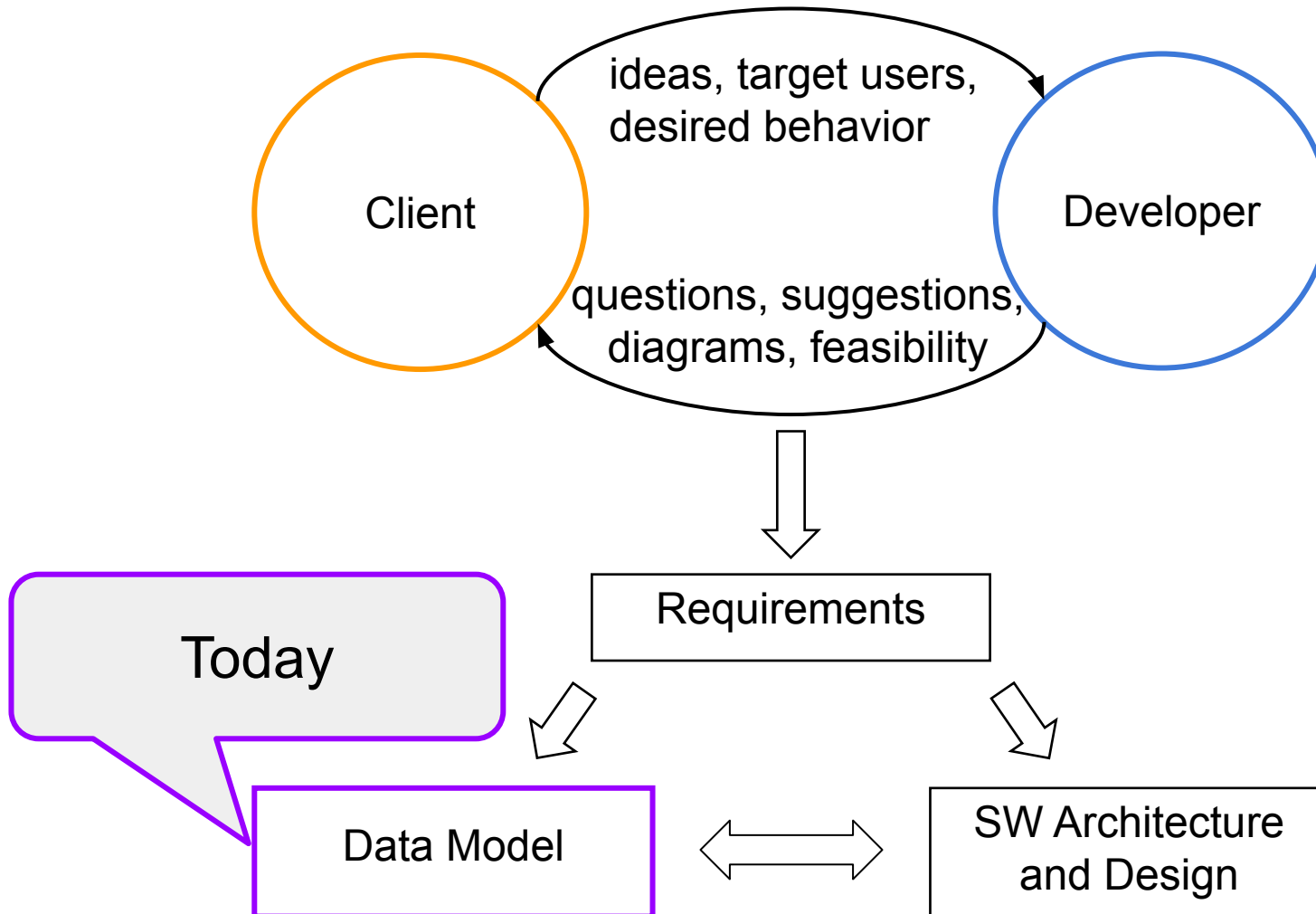
Life-cycle stages

Virtually all SDLC models have the following stages

- Requirements
- Design
- Implementation
- Testing
- Maintenance

Do you remember this?

From Requirements to System Design



Data Modelling

Goals for today

- How to model data?
 - Identify Entities
 - Identify Attributes
 - Identify Relationships
 - Assign Keys
 - ~~○ (Normalization to reduce redundancy)~~
 - ~~○ (Denormalization to improve performance)~~
- Common “language” for data modelling
 - ER (Entity-Relationship) diagrams
 - Just one out of many possibilities (diagrams, tables, text)
- Develop a data model for a course-registration system

ER diagrams: overview

- An Entity Relationship (ER) diagram is a **graphical representation** of a **data model**.
- It shows the **relationship** between **entities** (e.g., people, objects, events, or concepts) within a system.
- It **can be mapped** to a **relational** (database) **schema**.

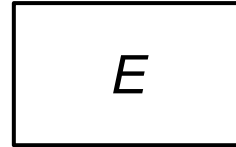
ER diagrams: graphical syntax

- An entity E

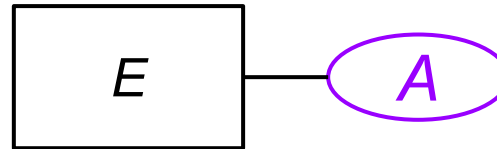


ER diagrams: graphical syntax

- An entity E

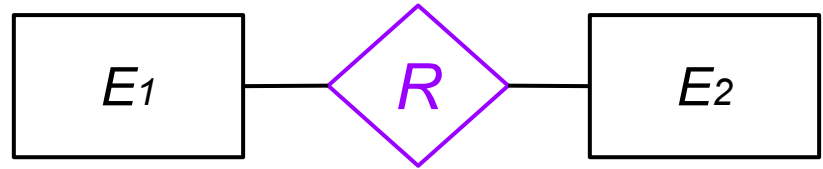
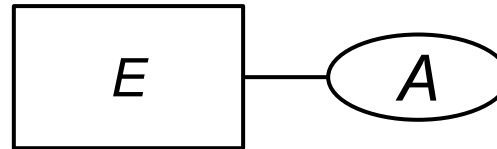
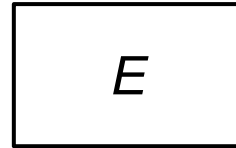


- An attribute A of entity E



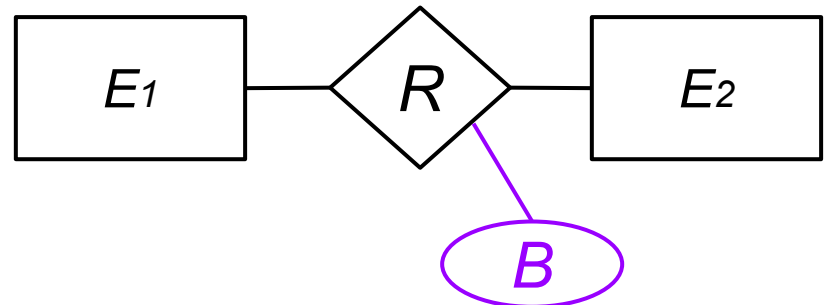
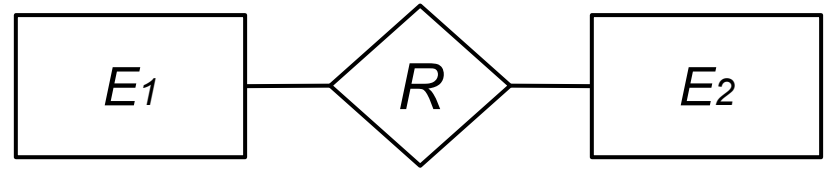
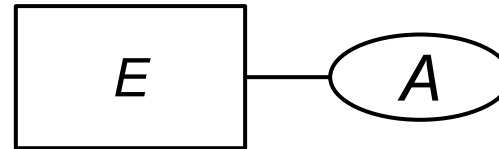
ER diagrams: graphical syntax

- An entity E
- An attribute A of entity E
- A relationship R between two entities E_1 and E_2



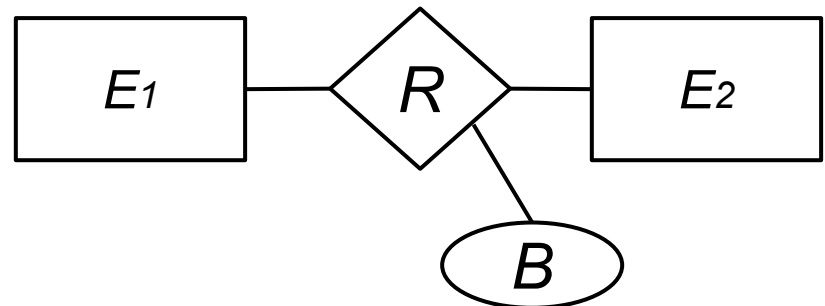
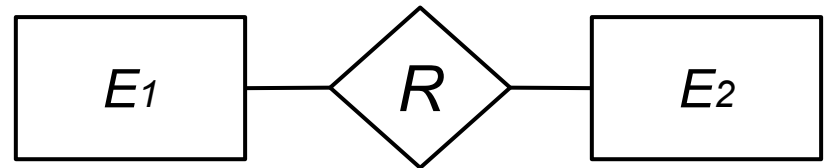
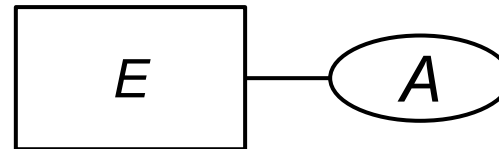
ER diagrams: graphical syntax

- An entity E
- An attribute A of entity E
- A relationship R between two entities E_1 and E_2
- An attribute B of relationship R



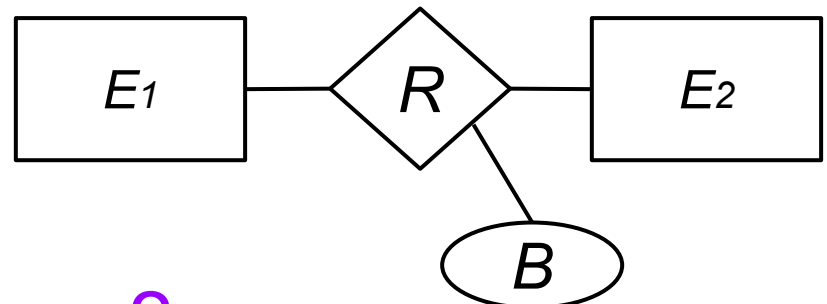
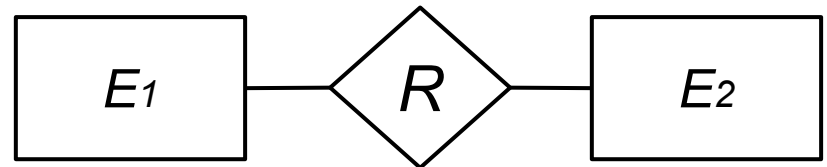
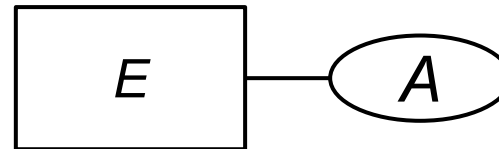
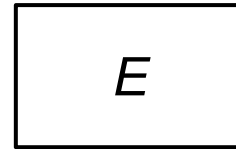
ER diagrams: rules

- An interconnecting line is only allowed between:
 - a box and a diamond,
 - a box and an oval,
 - a diamond and an oval.
- An oval must have exactly one connecting line.
- Names of boxes must be unique in the diagram.
- Names of ovals must be unique per box/diamond.



ER diagrams: rules

- An interconnecting line is only allowed between:
 - a box and a diamond,
 - a box and an oval,
 - a diamond and an oval.
- An oval must have exactly one connecting line.
- Names of boxes must be unique in the diagram.
- Names of ovals must be unique per box/diamond.



Questions?

A first example

Let's model a simple course registration system:

- **Students**
- **Instructors**
- **Courses**

A first example: identify entities

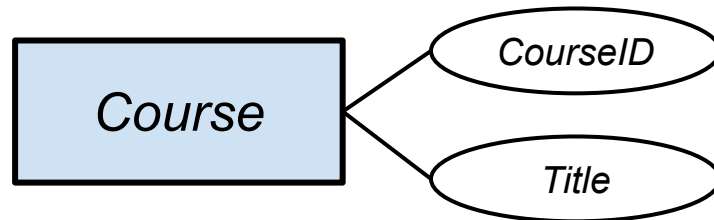
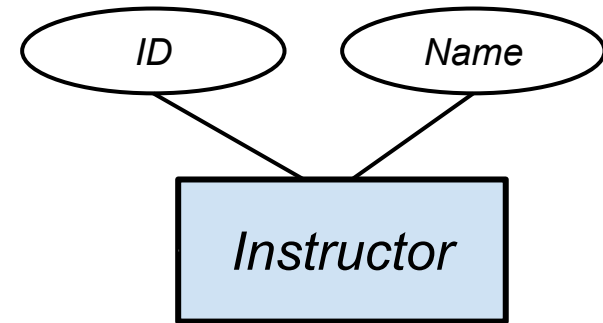
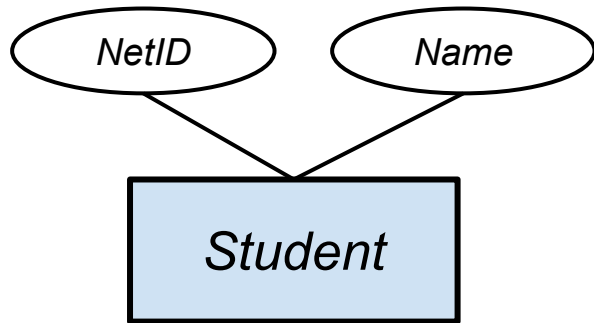
Student

Instructor

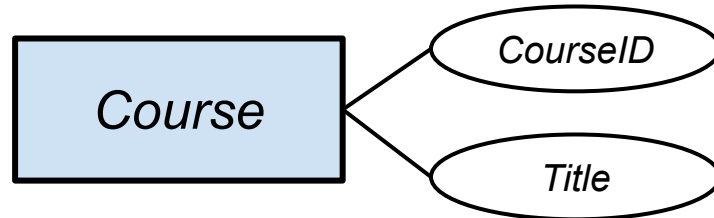
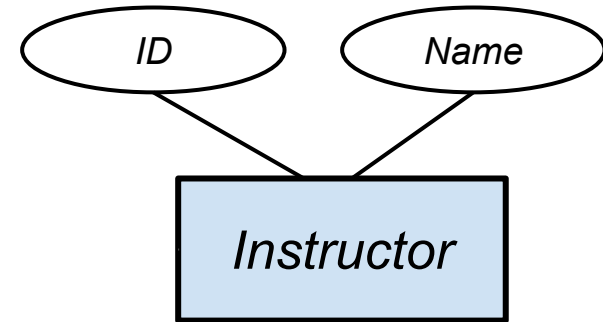
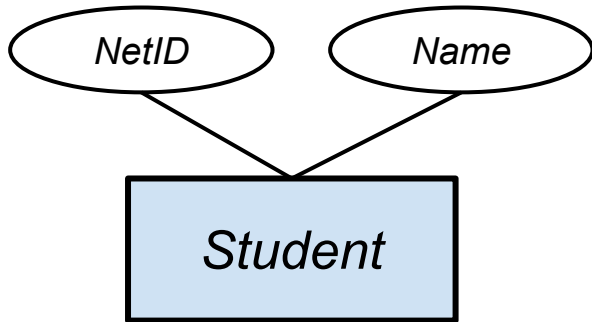
Course

What attributes should we add?

A first example: identify attributes

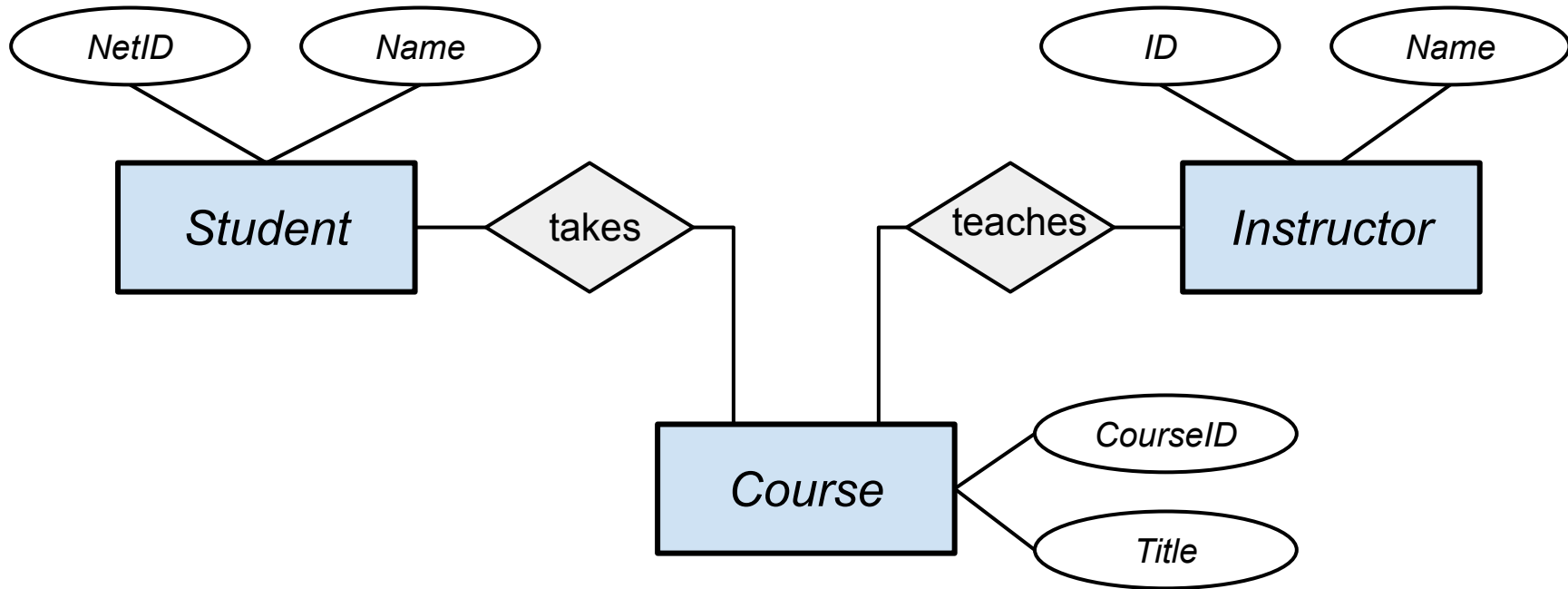


A first example: identify attributes



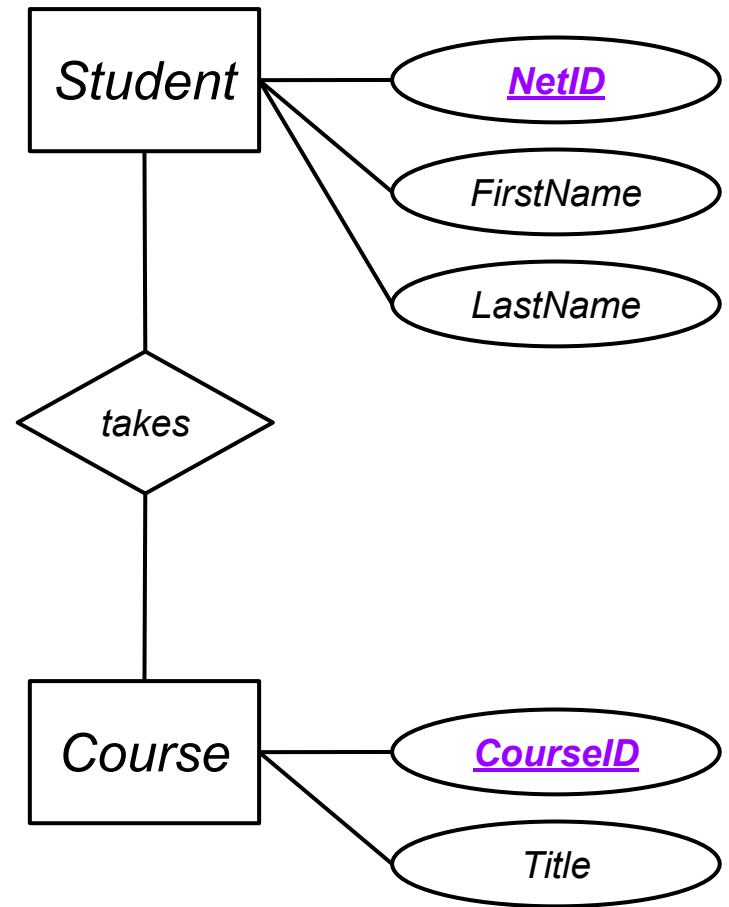
What relationships should we add?

A first example: identify relationships



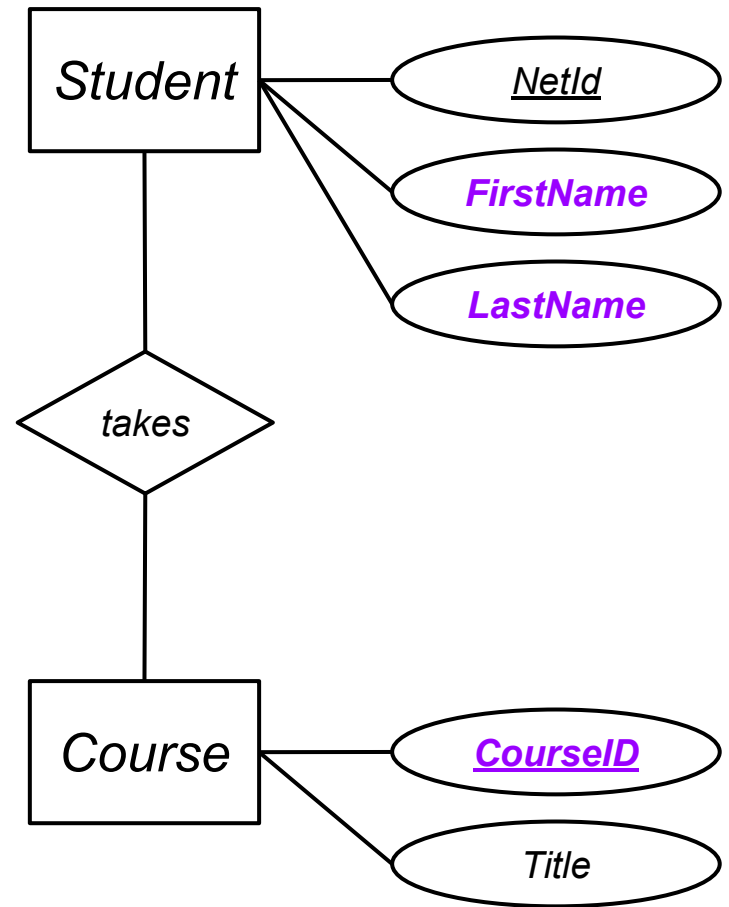
ER diagrams: keys and cardinalities

- A **key** is an (underlined) attribute, or a set of attributes, which **uniquely identifies an entity**.



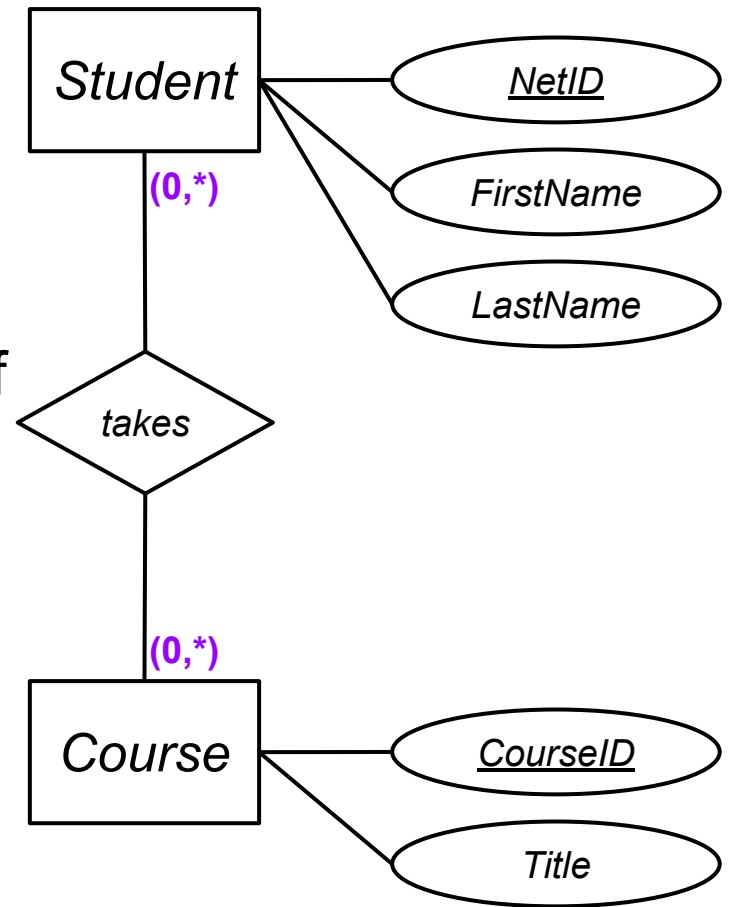
ER diagrams: keys and cardinalities

- A key is an (underlined) attribute, or a set of attributes, which uniquely identifies an entity.
- A key can be **artificial** or **natural**.



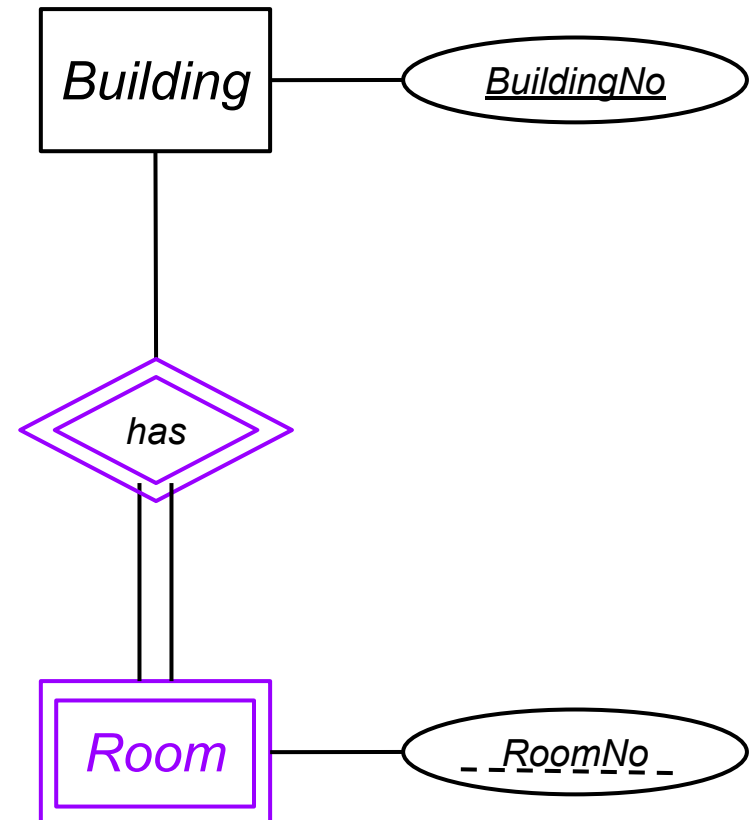
ER diagrams: keys and cardinalities

- A key is an (underlined) attribute, or a set of attributes, which uniquely identifies an entity.
- A key can be artificial or natural.
- The **cardinalities** define the kind of relationship (**one-to-one**, **one-to-many**, or **many-to-many**).
- There are different notations for cardinalities. For example:
 - 1 = (1,1)
 - c = (0,1)
 - m = (1,*)
 - mc = (0,*)



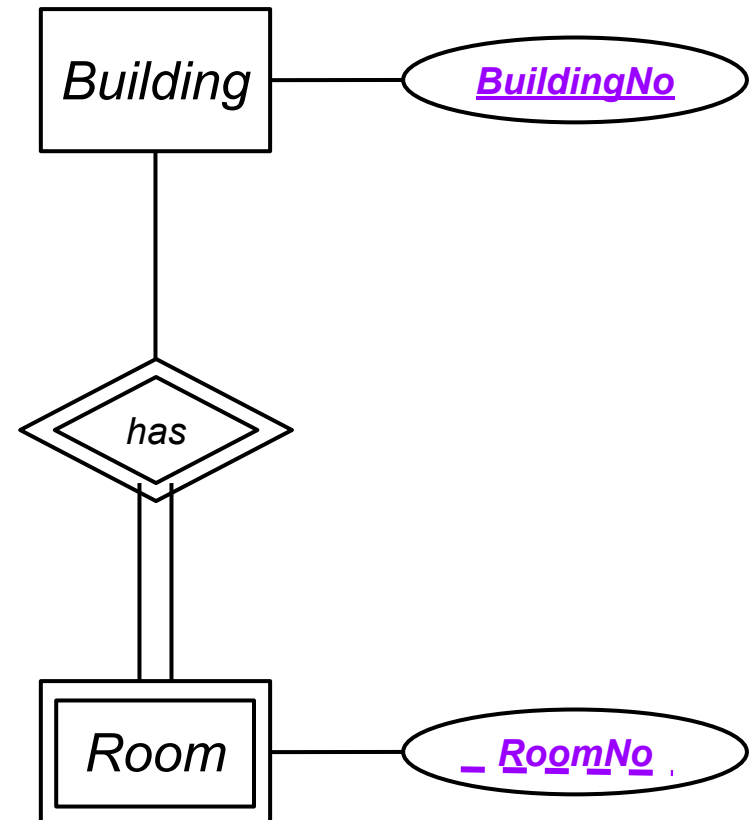
ER diagrams: weak entities

- A **weak entity** can't exist on its own (if a building is torn down, its rooms disappear).



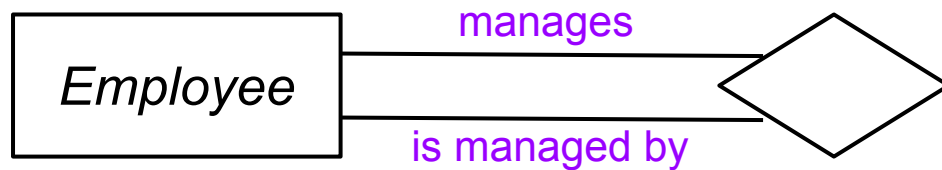
ER diagrams: weak entities

- A weak entity can't exist on its own (if a building is torn down, its rooms disappear).
- A weak entity is only **uniquely identifiable** in **reference** to another entity.

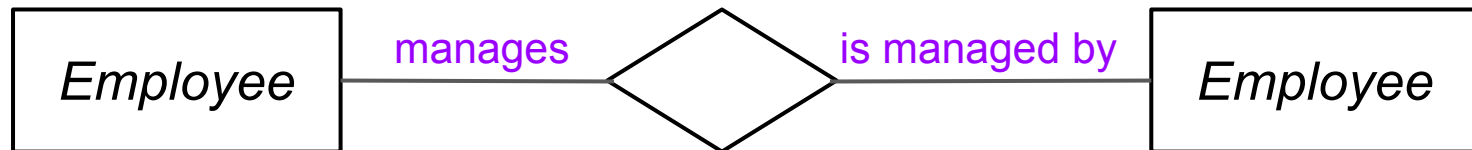


ER diagrams: self references and roles

- A **self reference** is usually explicitly annotated with **roles** to clarify the meaning of the self-referencing relationship.



Think about (but never draw) the following:

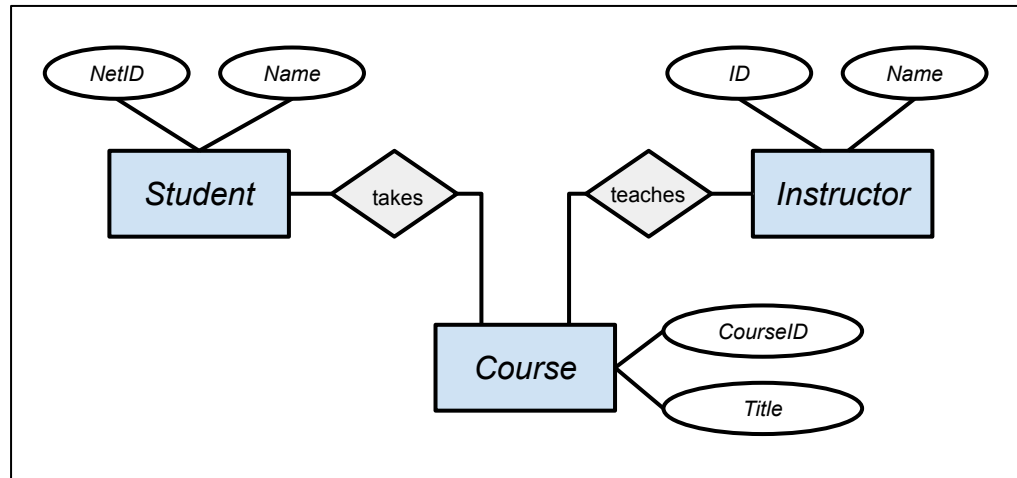




Putting it all together

Let's **augment** our **model** of a course registration system:

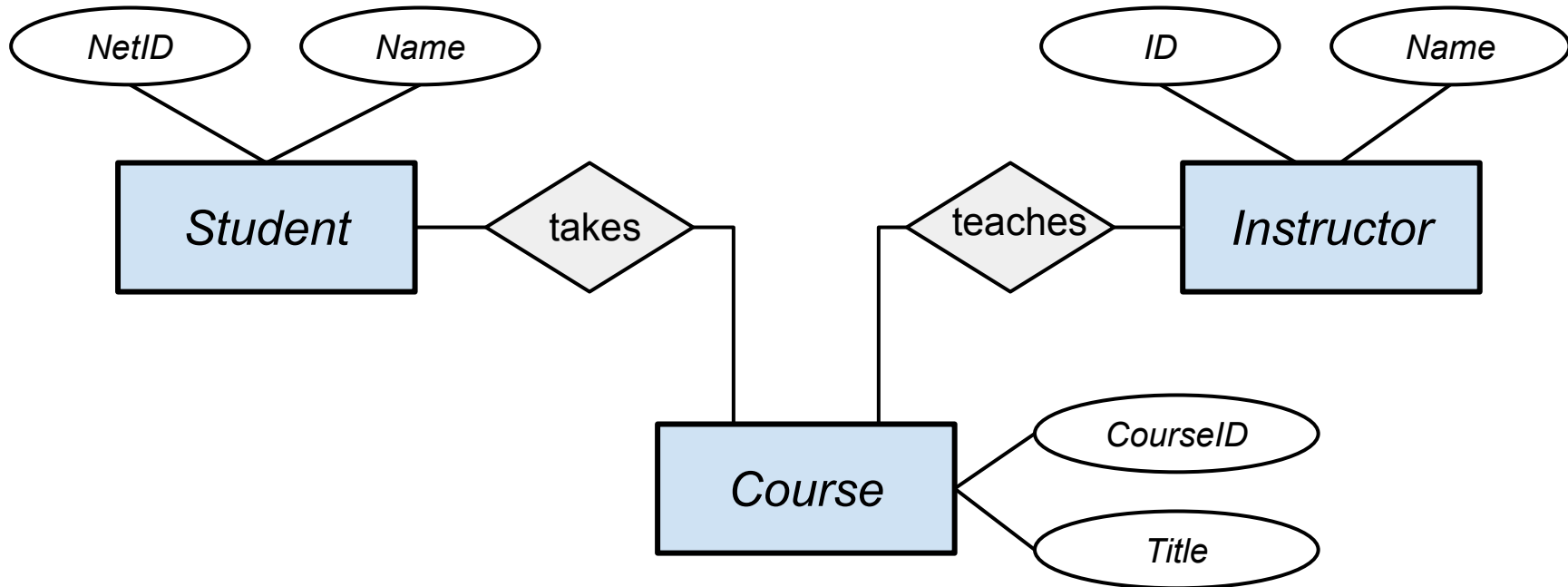
- Prerequisites
- Assignments
- Points/grades



Instructions

<https://tinyurl.com/cse403-ER>

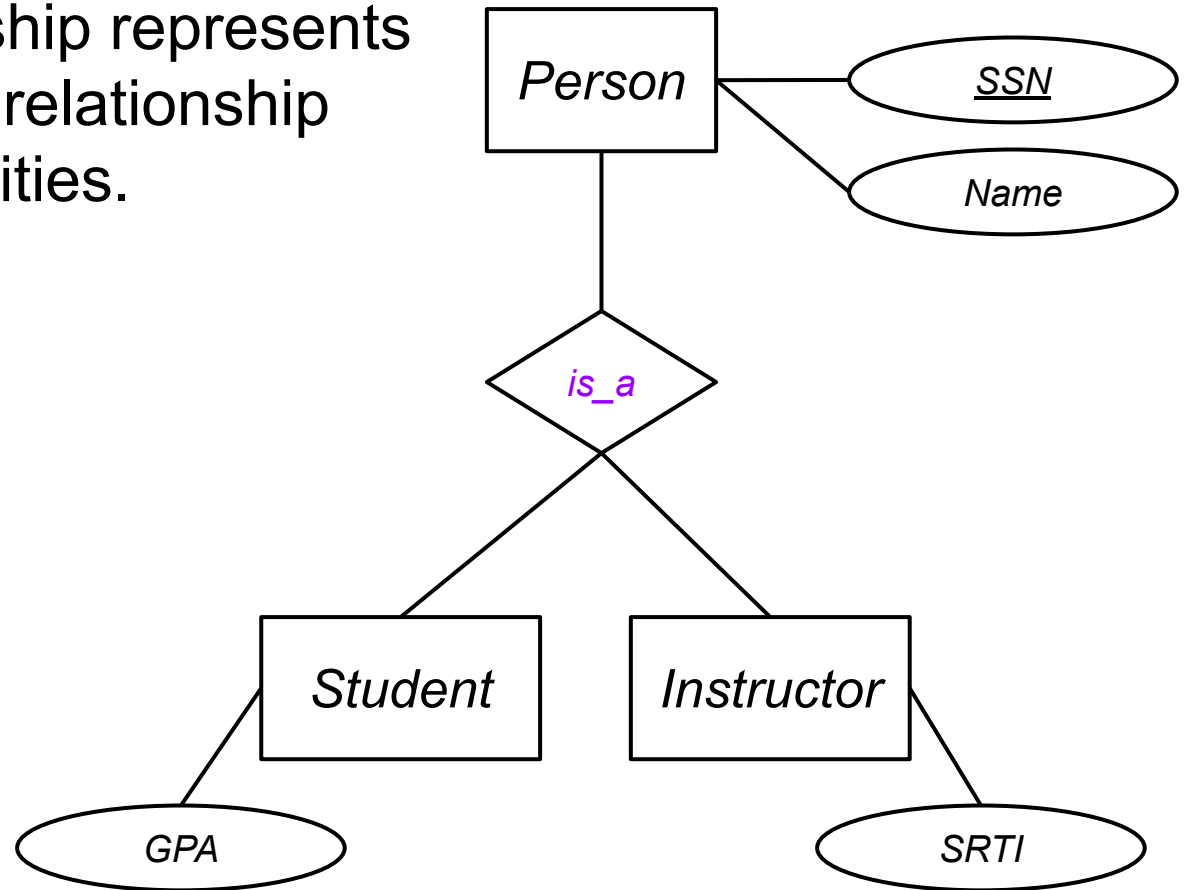
Putting it all together



Additional material, not discussed in class

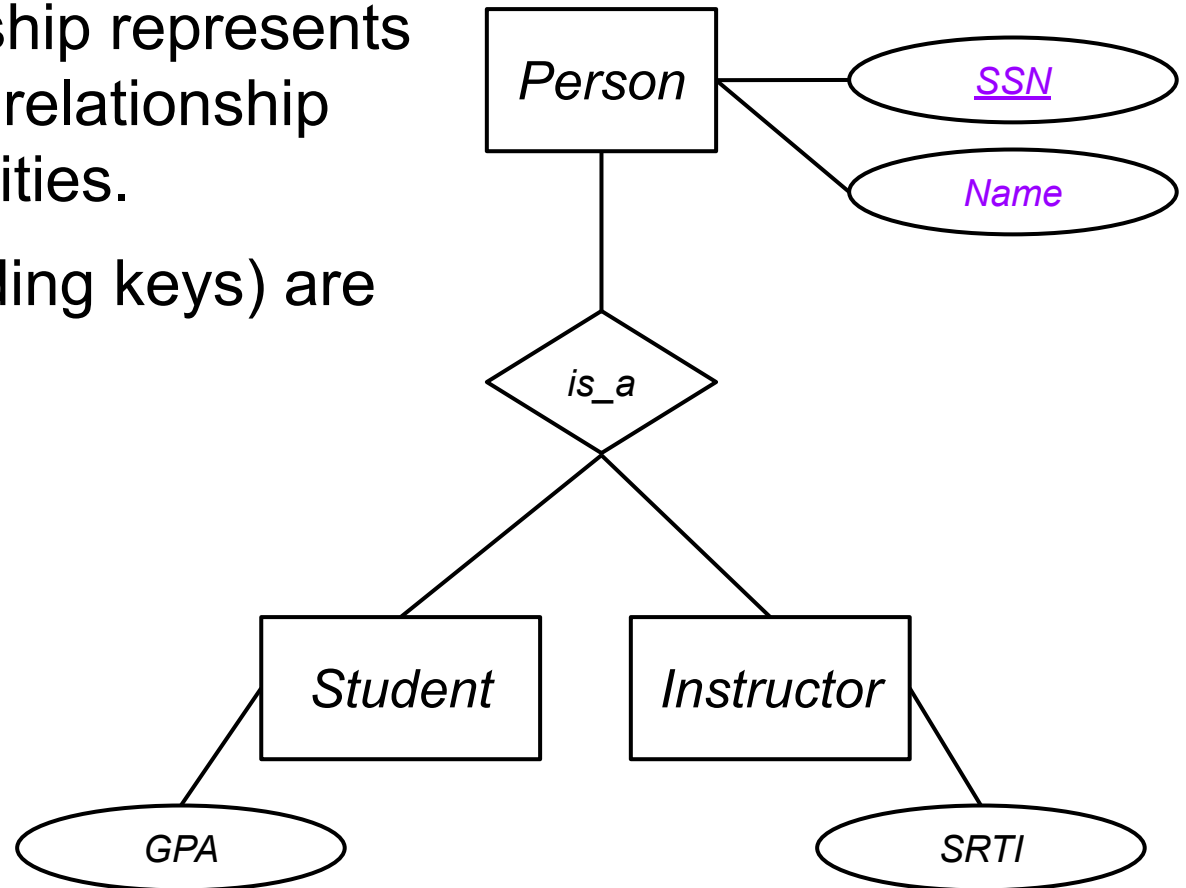
ER diagrams: generalization

- An *is_a* relationship represents a generalization relationship between two entities.



ER diagrams: generalization

- An *is_a* relationship represents a generalization relationship between two entities.
- **Attributes** (including keys) are “**inherited**”.



ER diagrams: generalization

- An *is_a* relationship represents a generalization relationship between two entities.
- Attributes (including keys) are “inherited”.
- **Additional attributes** can be defined.

