

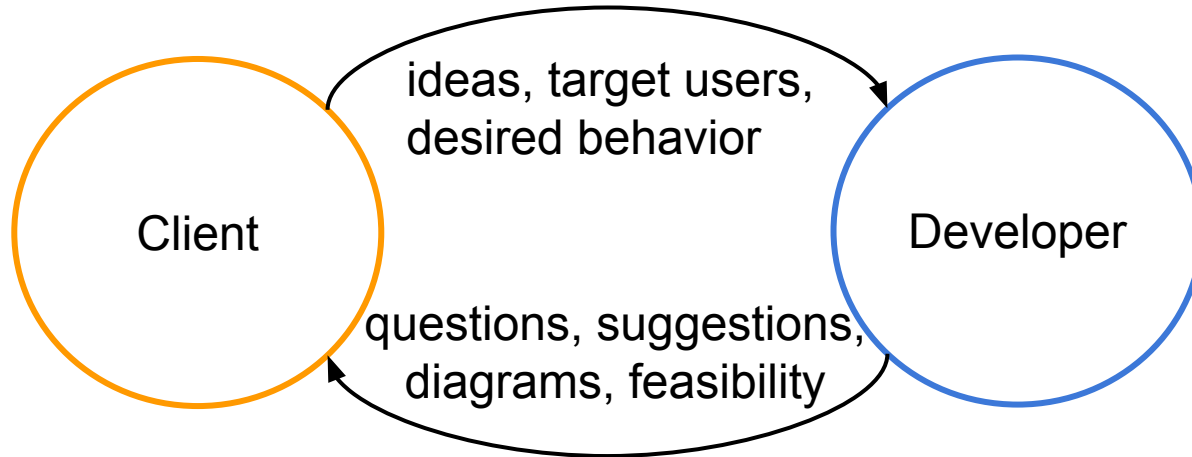
CSE 403

Software Engineering

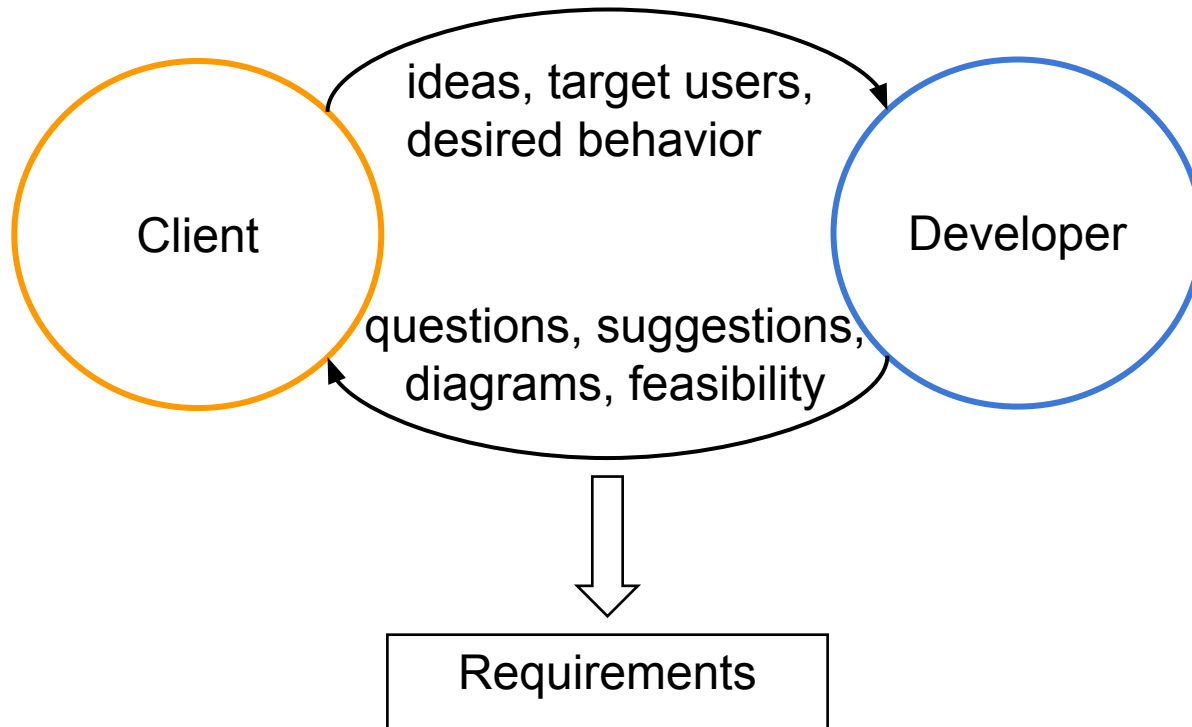
Data modelling

From Requirements to System Design

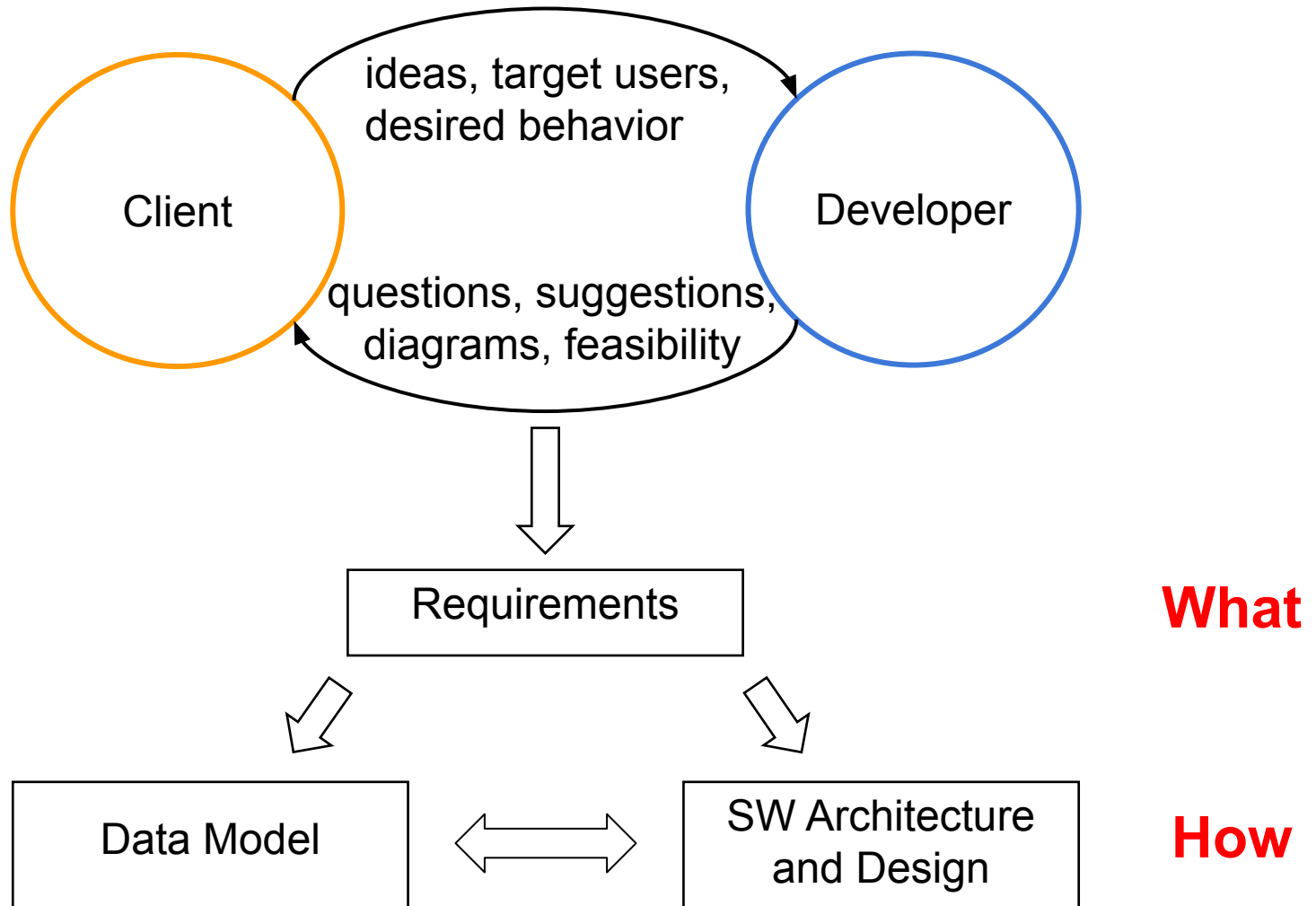
From Requirements to System Design



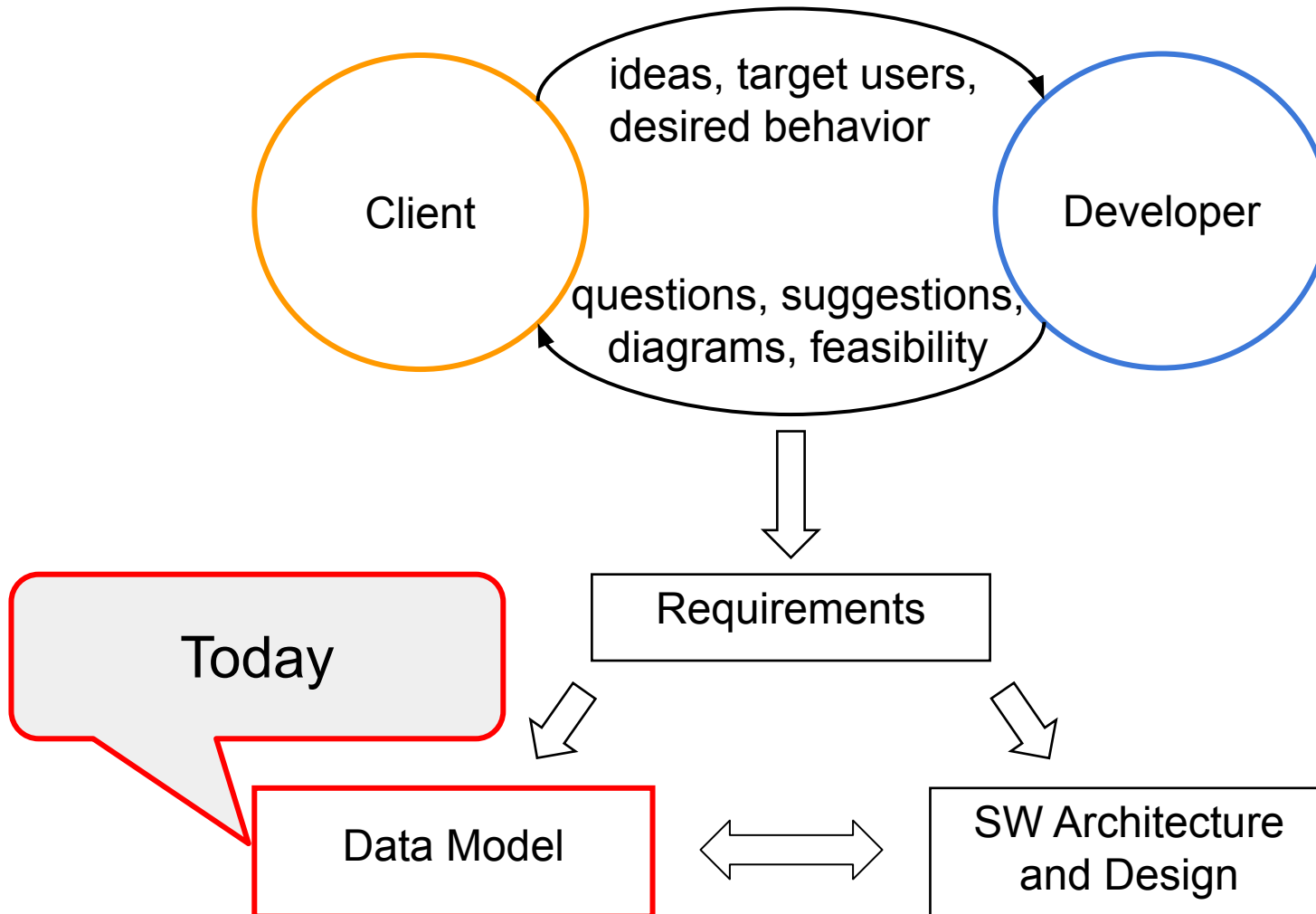
From Requirements to System Design



From Requirements to System Design



From Requirements to System Design



Data Modelling

Your program represents things in the world

Your program contains data structures

- Equivalently, a database contains your data (CSE 344)

How should you design those data structures?

Data modeling expresses:

- What data is important
- How the data are related to one another

Avoids redundancy, errors, and inefficiency

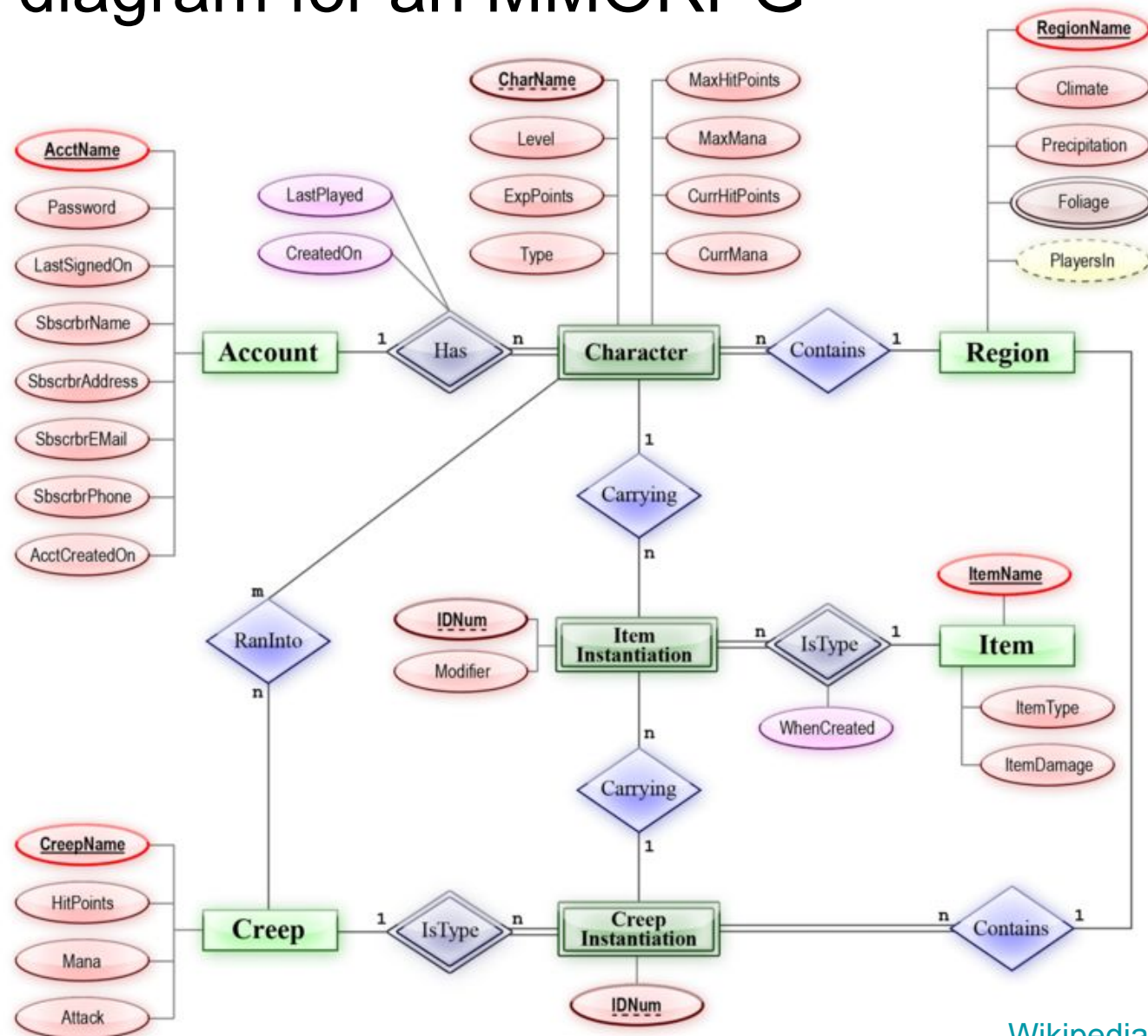
How to model data

- Identify Entities
- Identify Attributes
- Identify Relationships
- Assign Keys
- *(Normalization to reduce redundancy)*
- *(Denormalization to improve performance)*

ER (entity-relationship) diagrams

- ER diagrams are a common language for data modelling
 - Other possibilities exist: diagrams, tables, text
- 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 diagram for an MMORPG



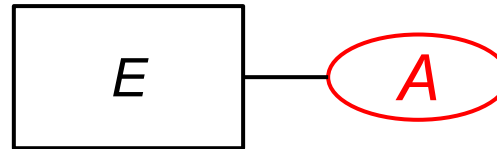
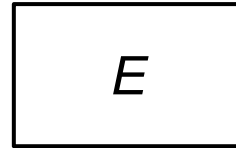
ER diagrams: graphical syntax

- An entity *E*
 - A physical or logical thing



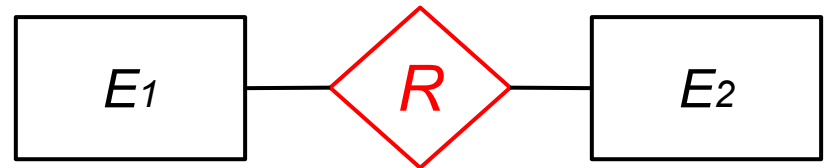
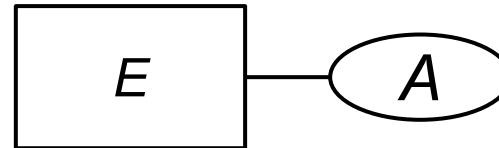
ER diagrams: graphical syntax

- An entity E
 - A physical or logical thing
- An **attribute** A of entity E
 - A property; like an object field



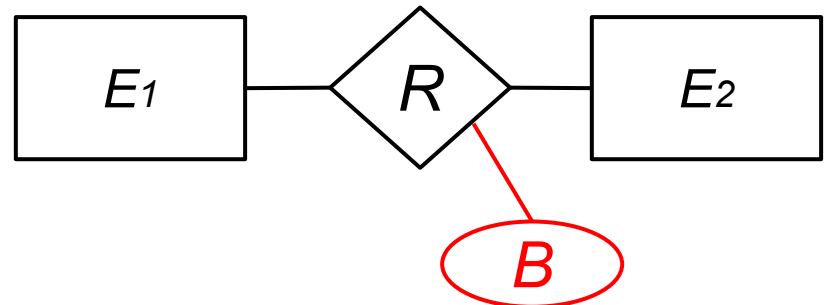
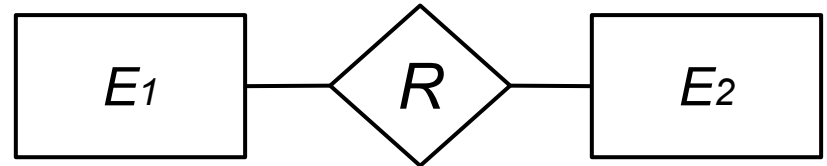
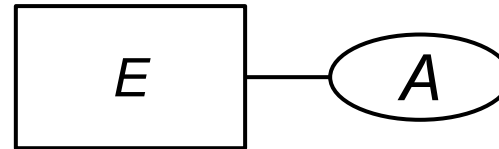
ER diagrams: graphical syntax

- An entity E
 - A physical or logical thing
- An attribute A of entity E
 - A property; like an object field
- A **relationship** R between two entities E_1 and E_2
 - Associations, dependencies



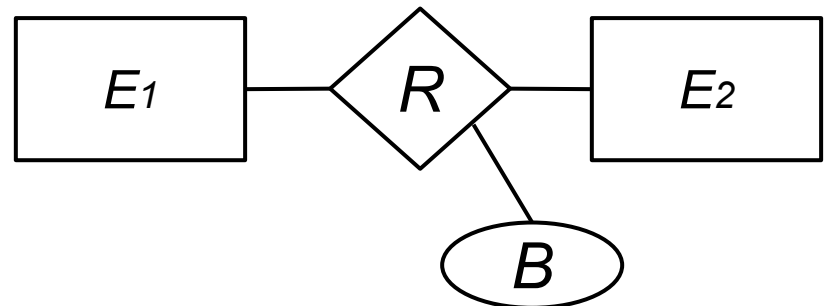
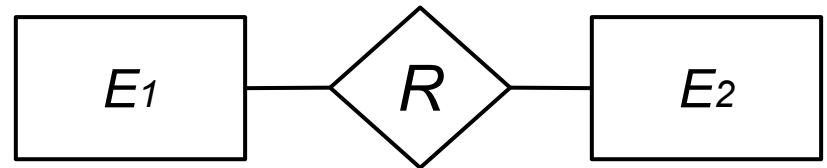
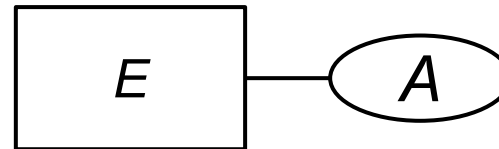
ER diagrams: graphical syntax

- An entity E
 - A physical or logical thing
- An attribute A of entity E
 - A property; like an object field
- A relationship R between two entities $E1$ and $E2$
 - Associations, dependencies
- An **attribute B** of relationship **R**



ER diagrams: syntactic rules

- An interconnecting line is only allowed between:
 - entity (box) and relationship (diamond)
 - entity (box) and attribute (oval)
 - relationship (diamond) and attribute (oval)
- An attribute (oval) has exactly 1 connecting line.
- Names of entities (boxes) are unique in the diagram.
- Names of attributes (ovals) are unique per entity/ relation (box/diamond).



Exercise: Model a course registration system

What are the entities?

Model a course registration system

What are the entities?

- **Students**
- **Instructors**
- **Courses**
- ...
- Prerequisites
- Assignments
- Points/grades
- ...
- Rooms
- Times
- Credits
- ...

Step 1: identify entities

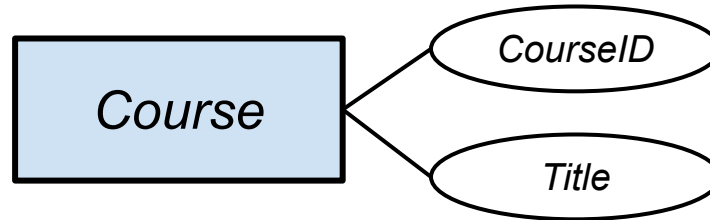
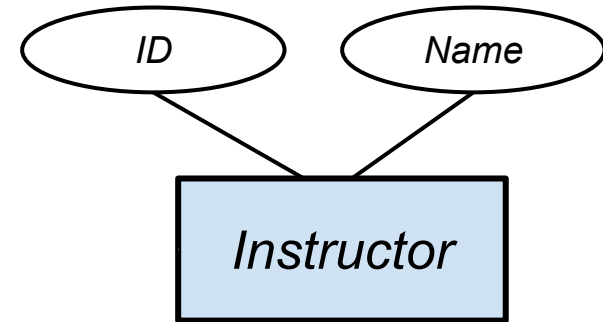
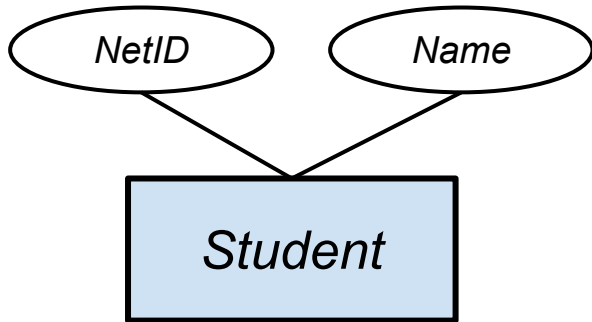
Student

Instructor

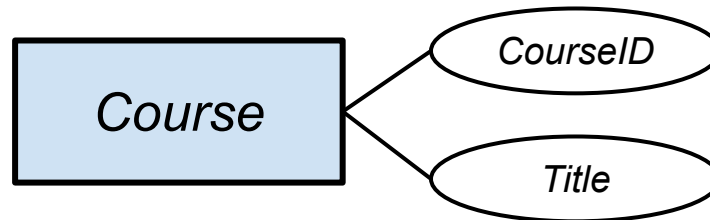
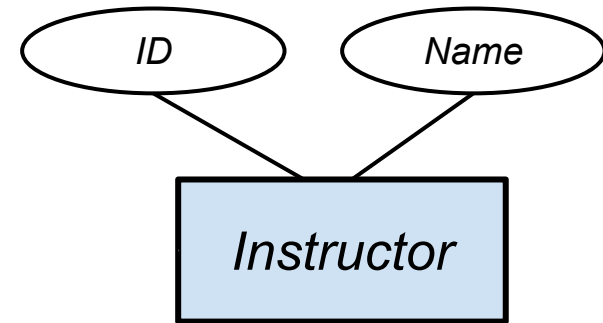
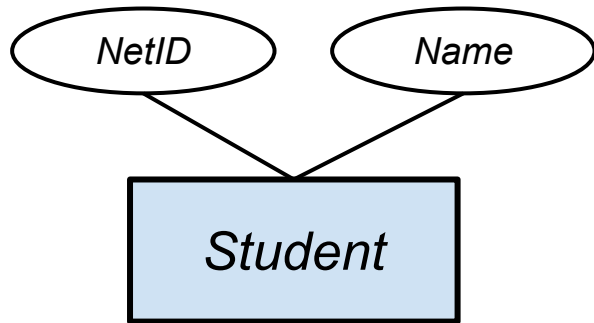
Course

What attributes should we add?

Step 2: identify attributes

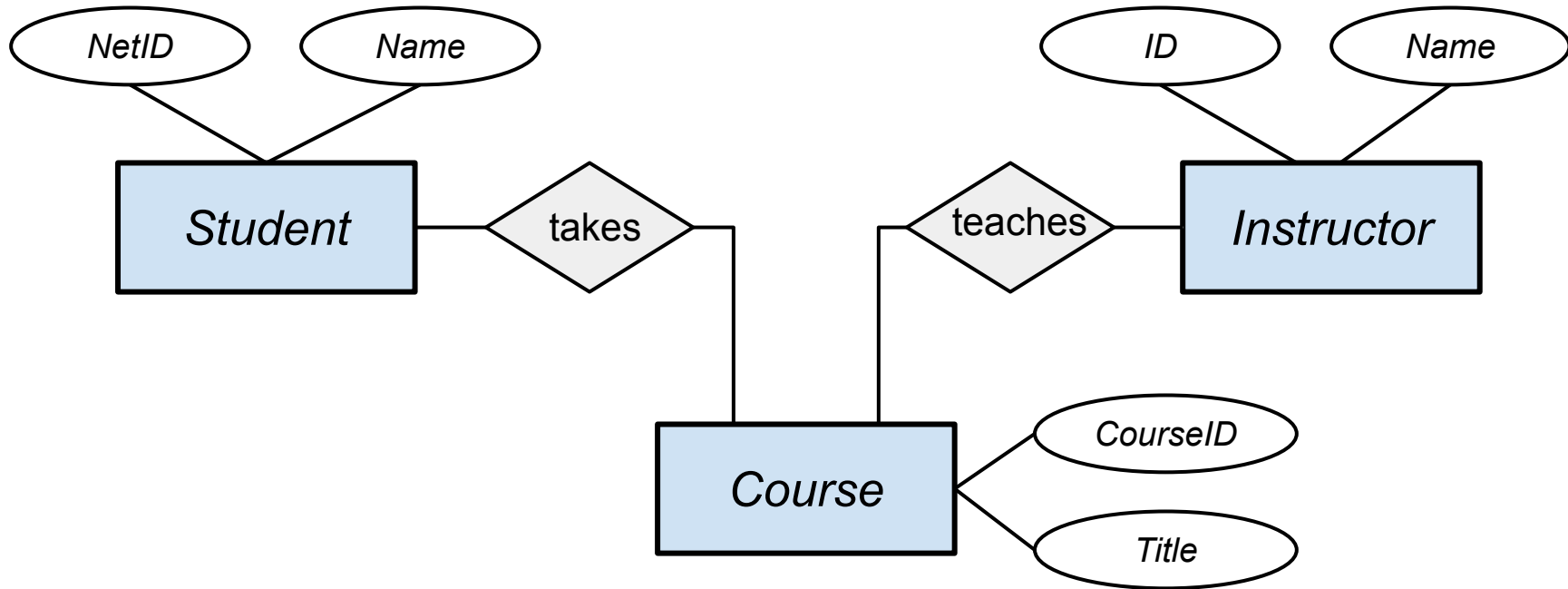


Step 3: identify relationships



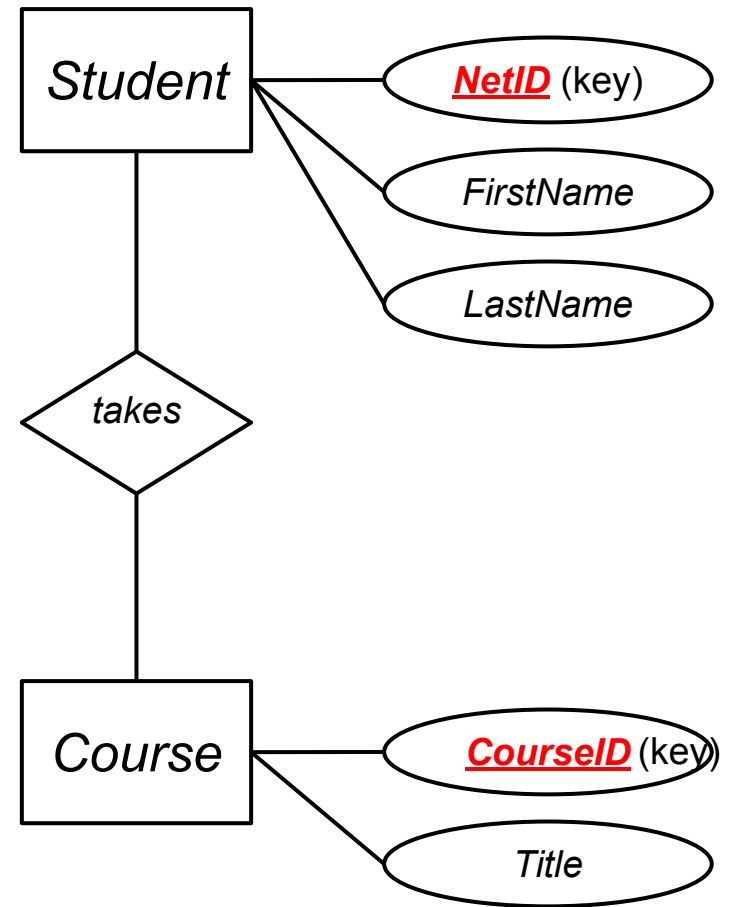
What relationships should we add?

A first example: identify relationships



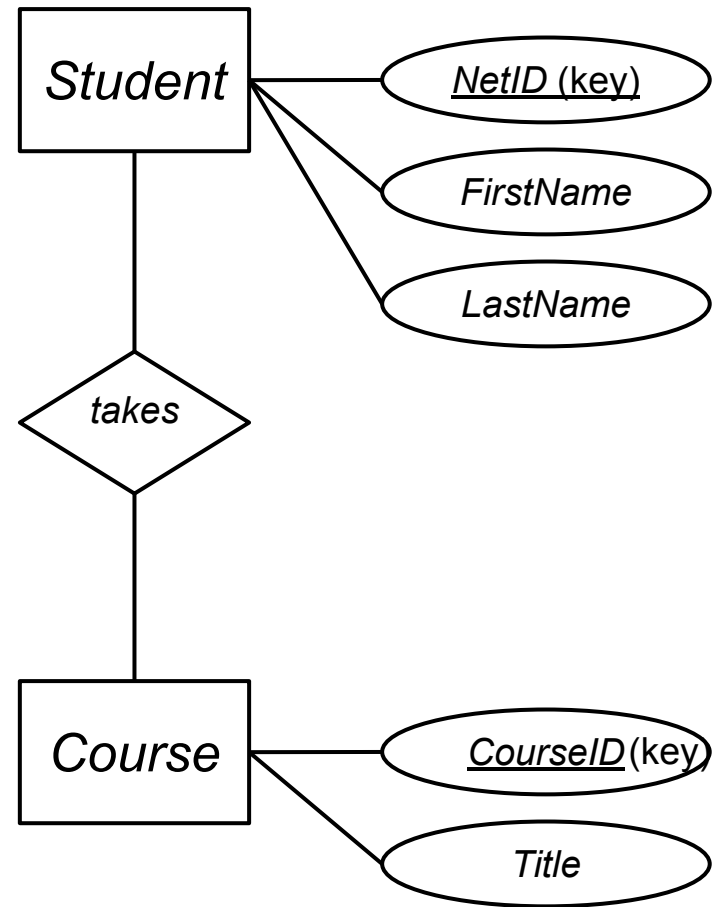
ER diagrams: keys and cardinalities

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



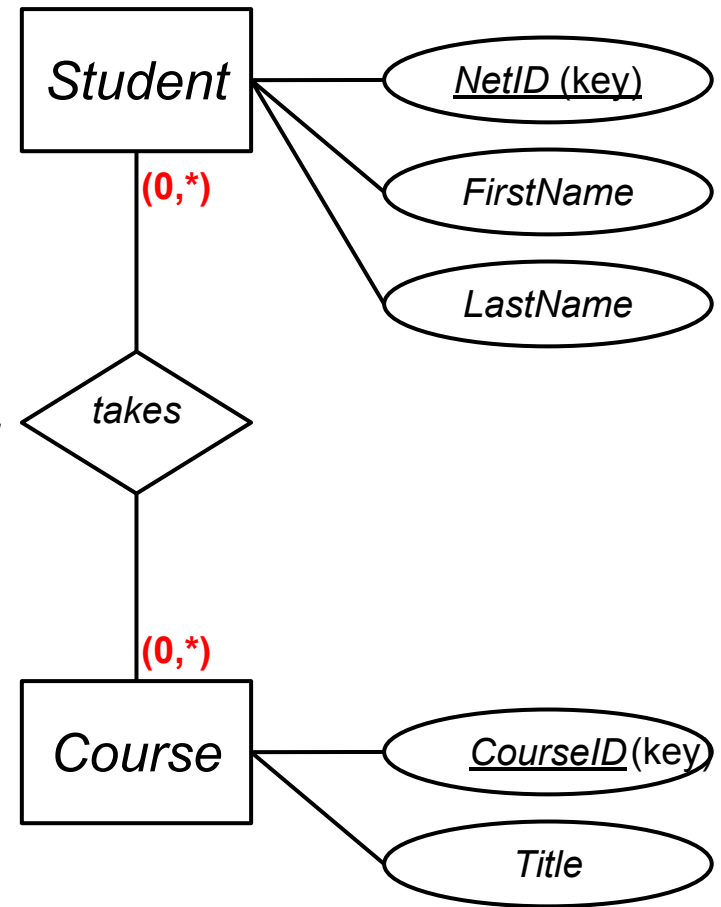
ER diagrams: keys and cardinalities

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



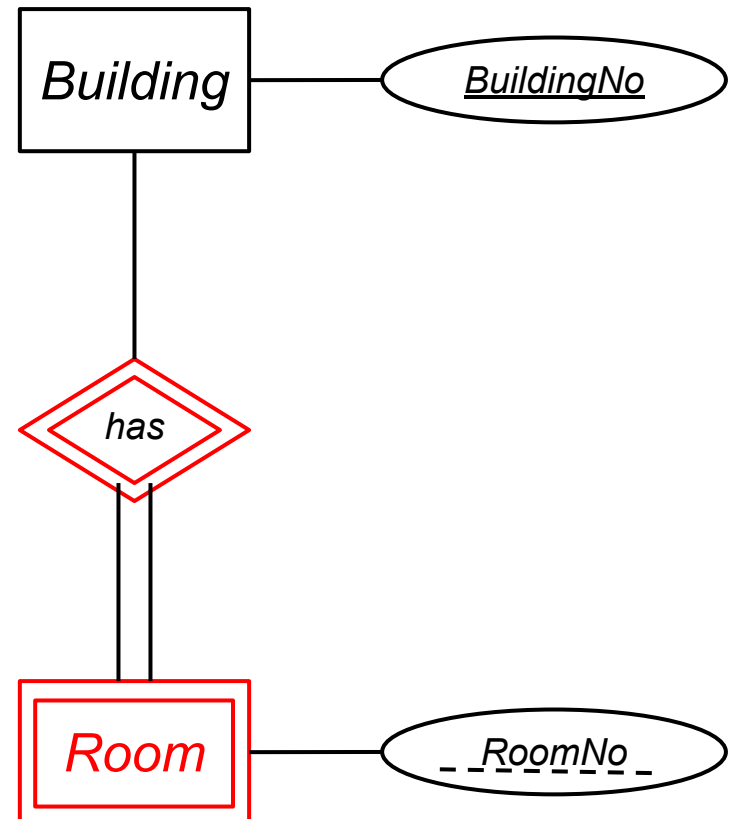
ER diagrams: keys and cardinalities

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



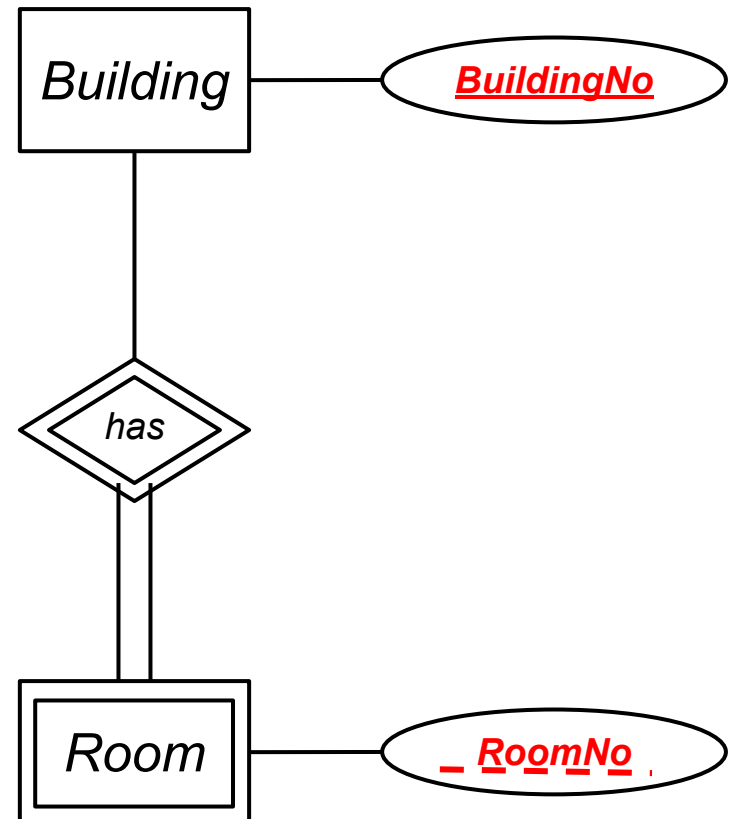
ER diagrams: weak entities

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



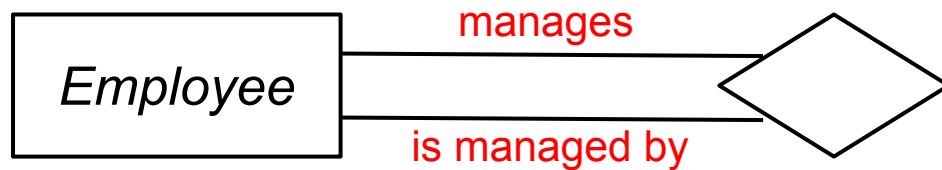
ER diagrams: weak entities

- 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.

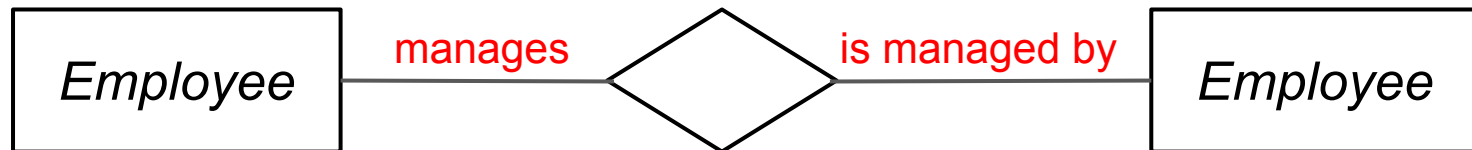


ER diagrams: self references and roles

- 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:



ER diagrams vs. code

ER diagrams vs. code

ER diagrams:

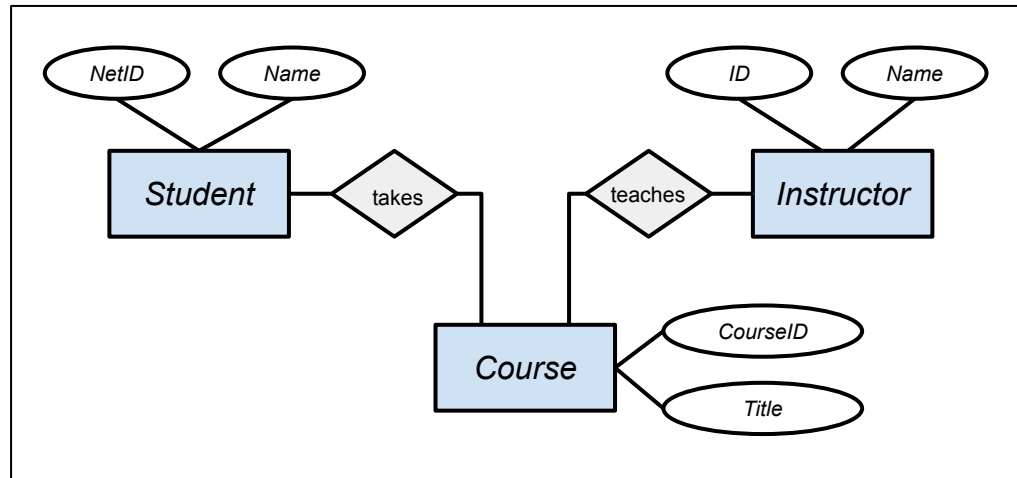
- Omit implementation details (caching, bookkeeping)
- Show cardinality (e.g., may a field be null?)
- Concise

Practice



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

- Prerequisites
- Assignments
- Points/grades



ER diagram: course registration system

Set up:

1. Start from the diagram on [the next slide](#), or draw your diagram in any other way you want.

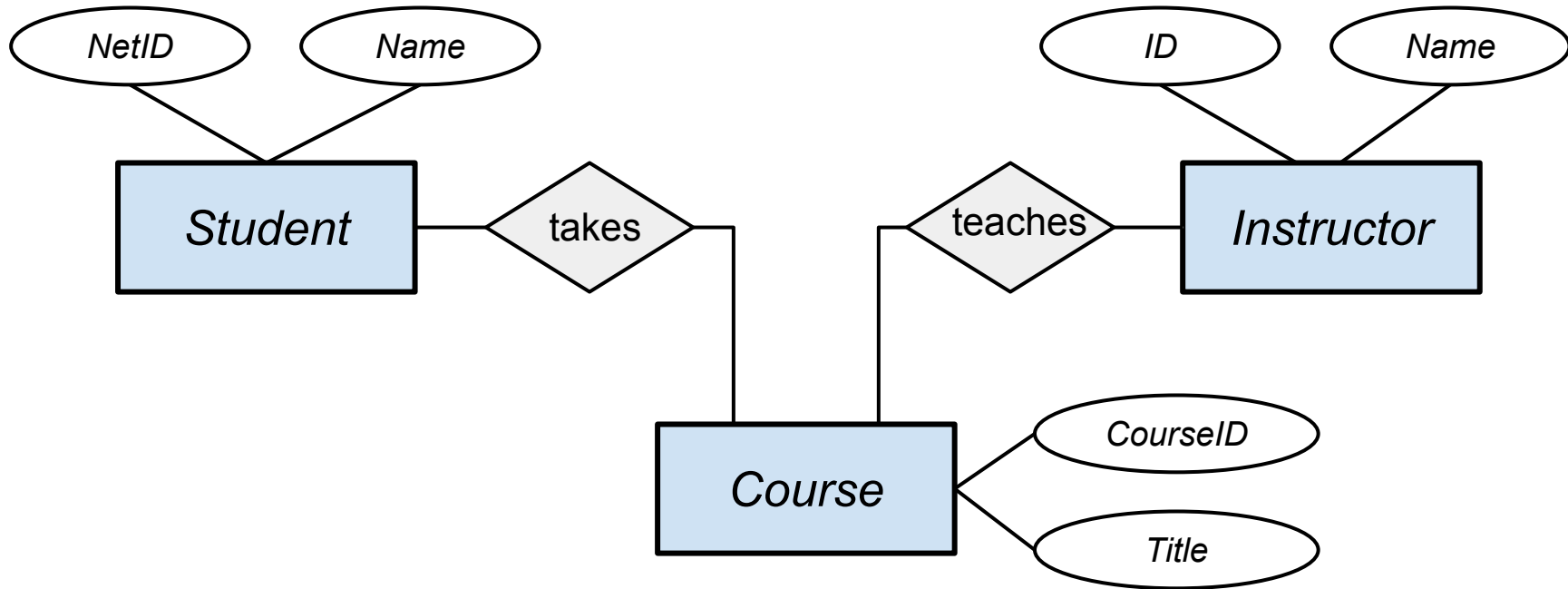
Instructions:

1. Augment the simple ER diagram on the next slide by
 - a. **modelling** the following concepts:
 - i. **Overall grade for a taken course**
 - ii. **Course prerequisites**
 - iii. **Assignments (and points)**
 - b. **adding keys and cardinalities**
2. Think about:
 - b. Which concepts should be modeled as (weak) entities vs. relationships vs. attributes?
 - c. Are there any self-references? What are reasonable cardinalities (reflecting reality)?

Additional Information:

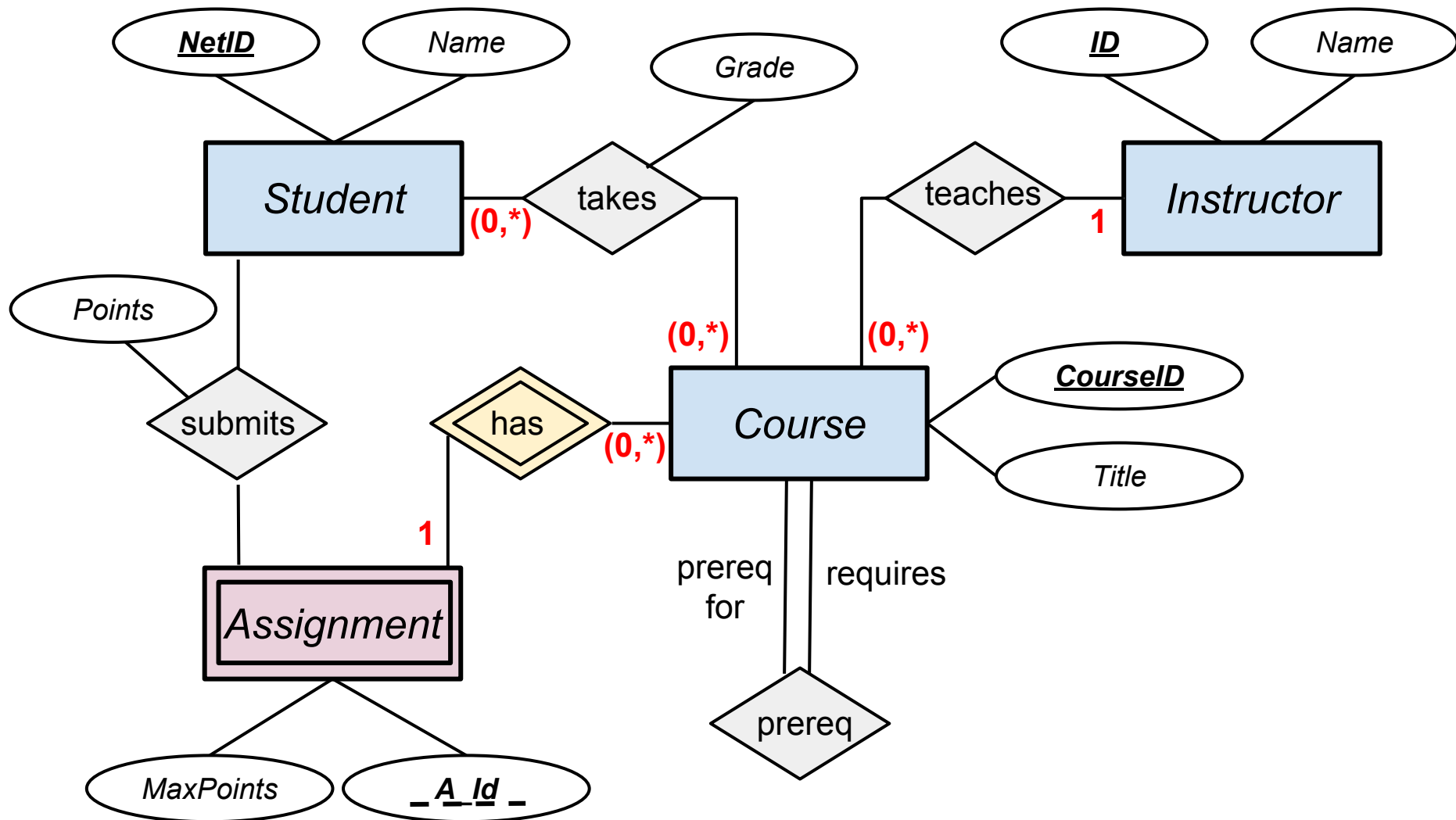
- ER syntax and rules were given on previous slides.

Course registration system: extend this ER model



Spoilers ahead!

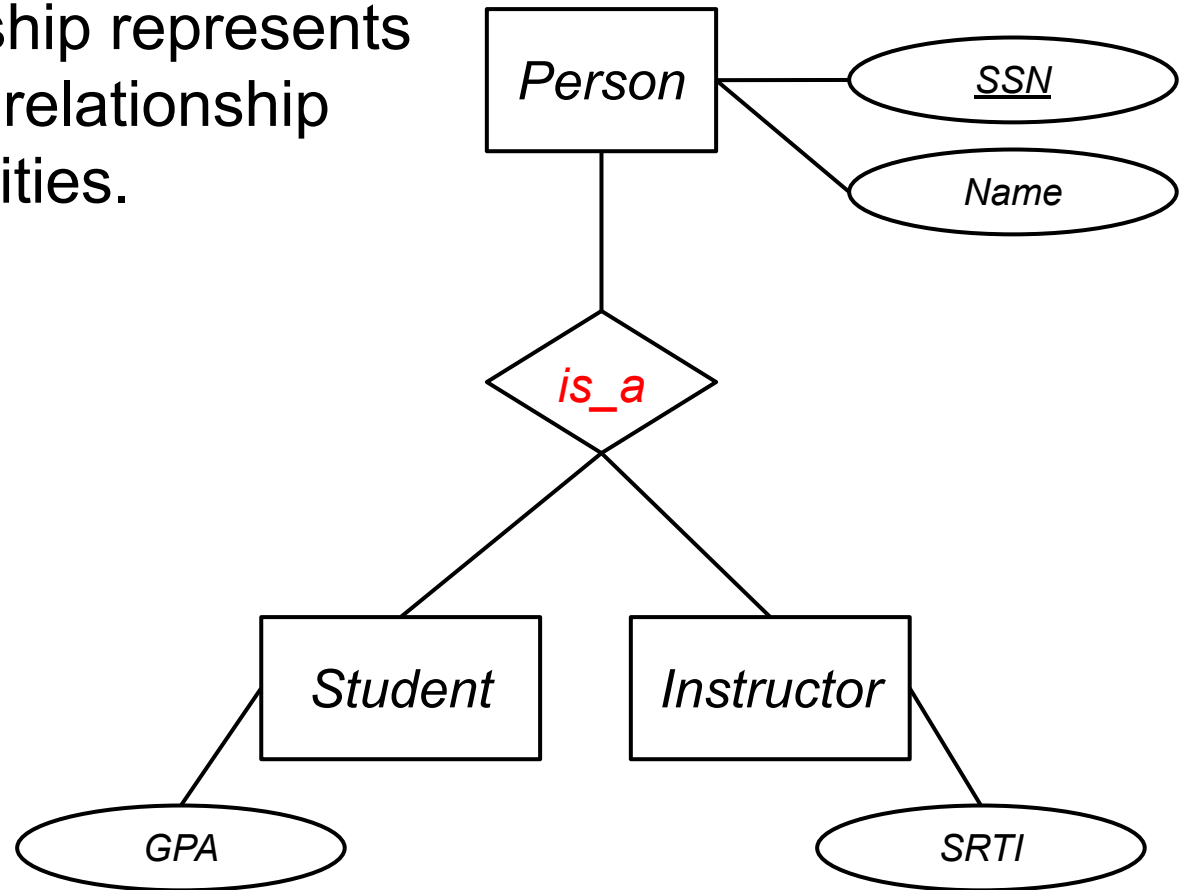
Extended course registration system (example)



Additional material, not discussed in class

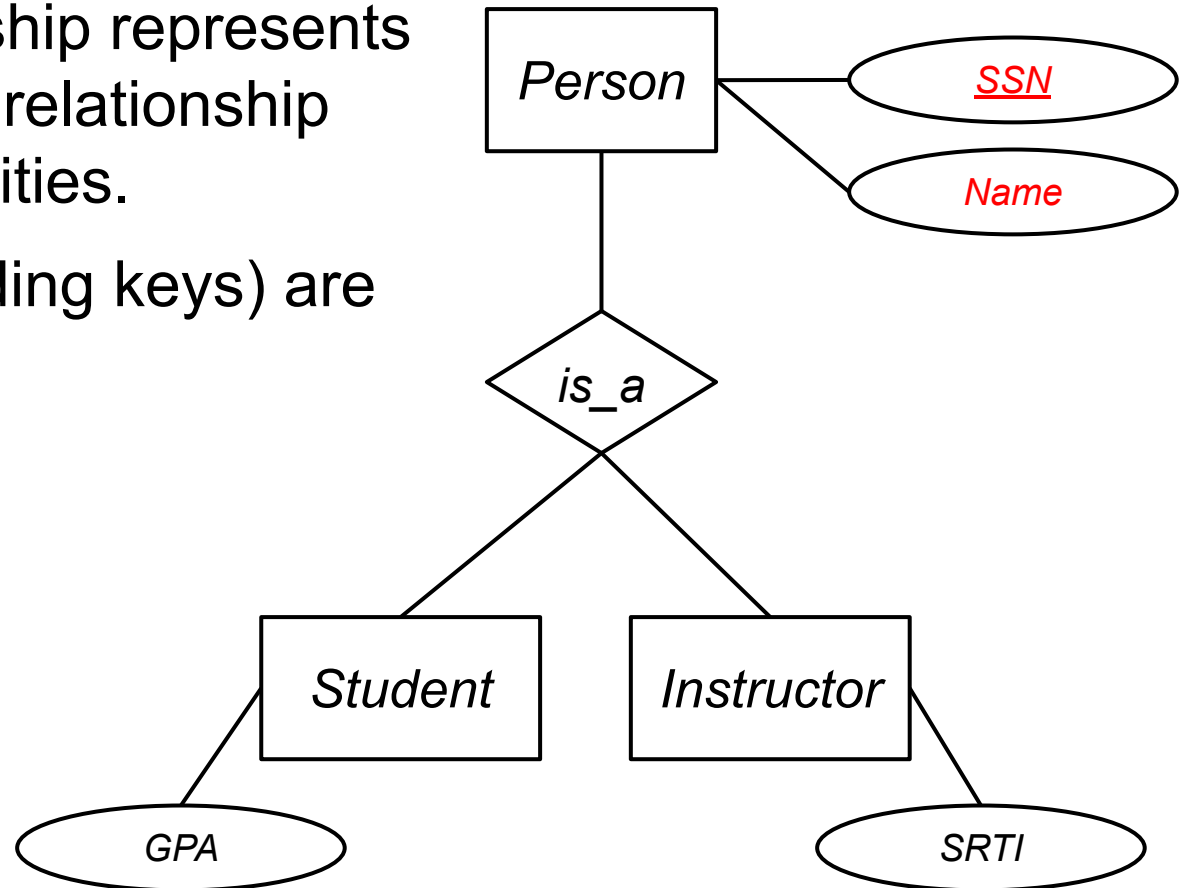
ER diagrams: modeling OO

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



ER diagrams: modeling OO

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



ER diagrams: modeling OO

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

