Tutorial 1

CS4.301: Data and Applications

October 7, 2024

Agenda

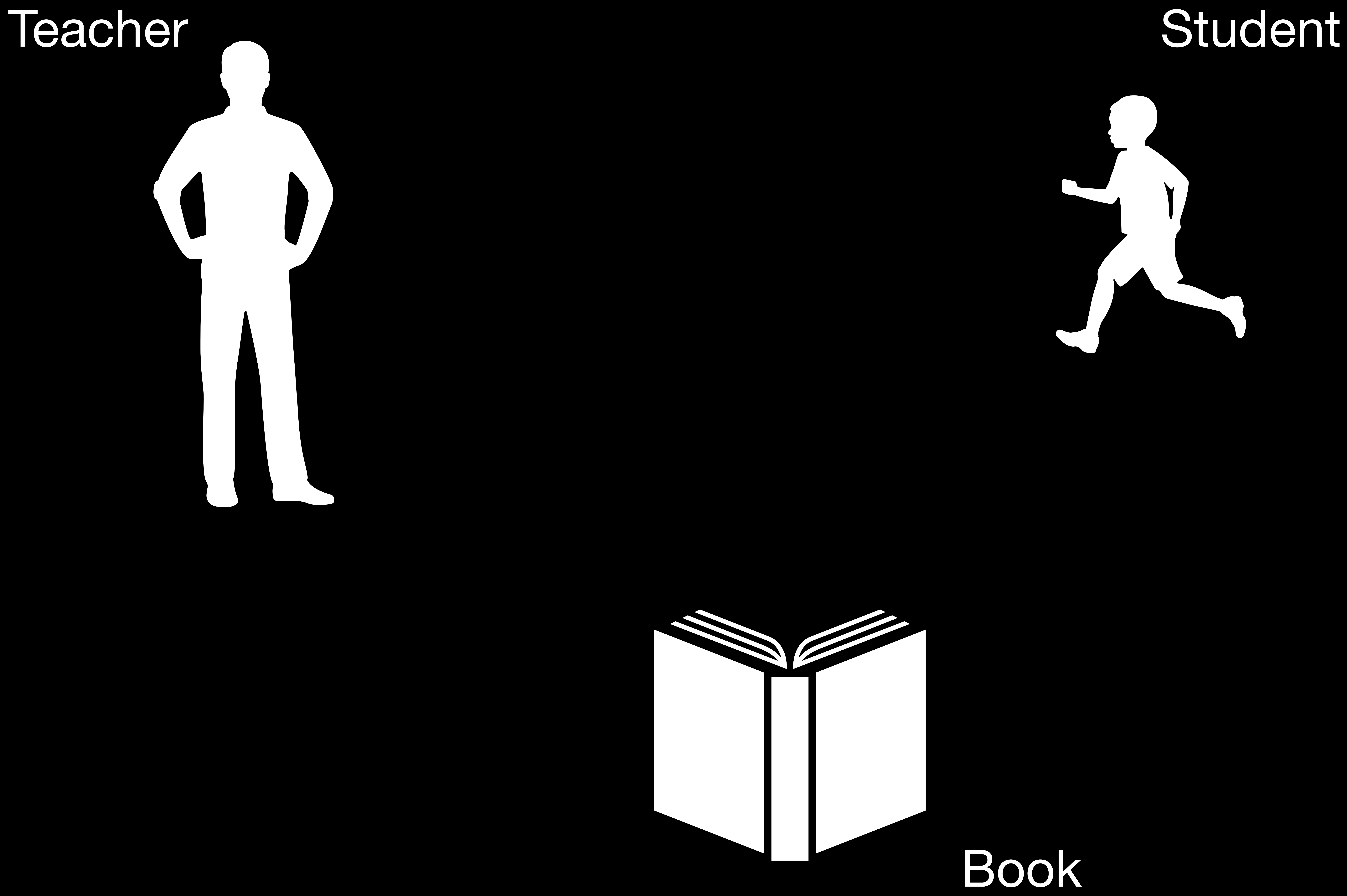
- ER Data Model
- Practice
- HW-1

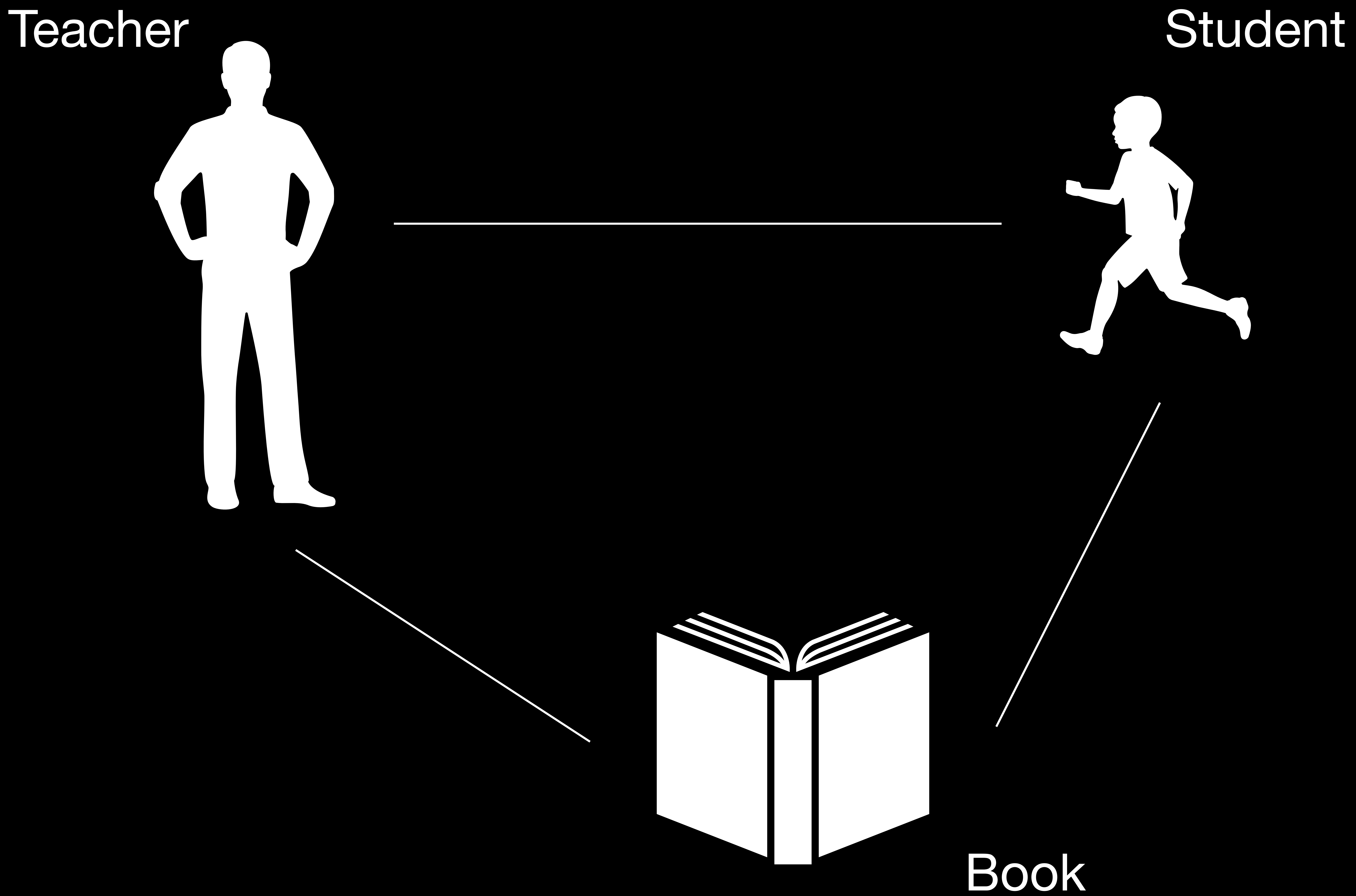
Miniworld / UoD → Data Requirements → Conceptual Design

Entity-relationship (ER) Model

ER Model

- Wikipedia: *“Describes interrelated things of interest in a specific domain of knowledge”*
- Designed by Peter Chen and published in a paper in 1976
  - https://dspace.mit.edu/bitstream/handle/1721.1/47432/entityrelationshx00chen.pdf
- Different sources might have slight variations, try to follow the course’s book (Elmasri) for this course


Teacher
Student
Book


Teacher
Student
Book

Components of an ER Model

- Entity sets (all entities of the same entity type)
- Relationship sets (all relationships of the same relationship type)
- Attributes

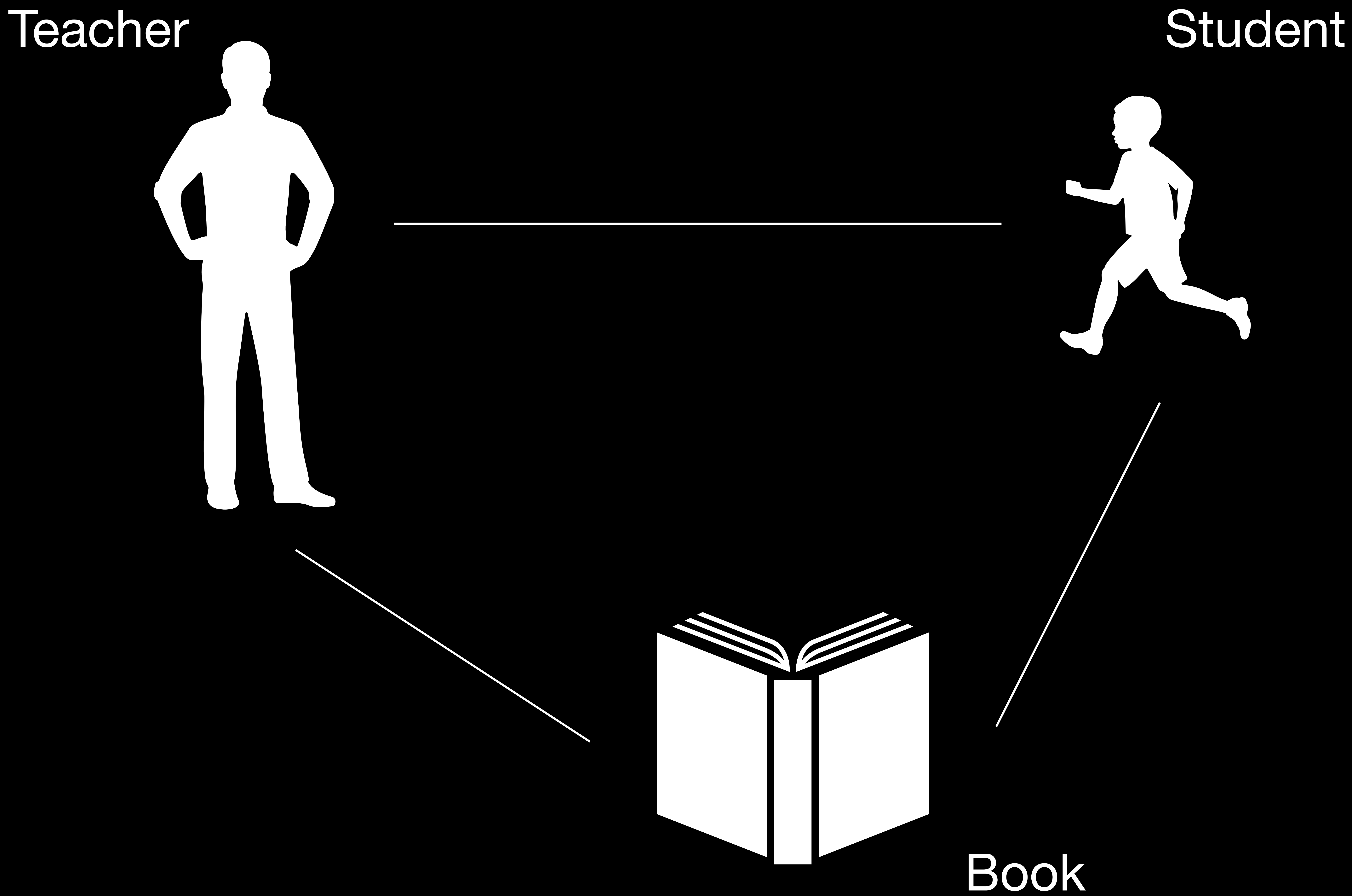
Entity & Entity type

Entity vs Entity Type

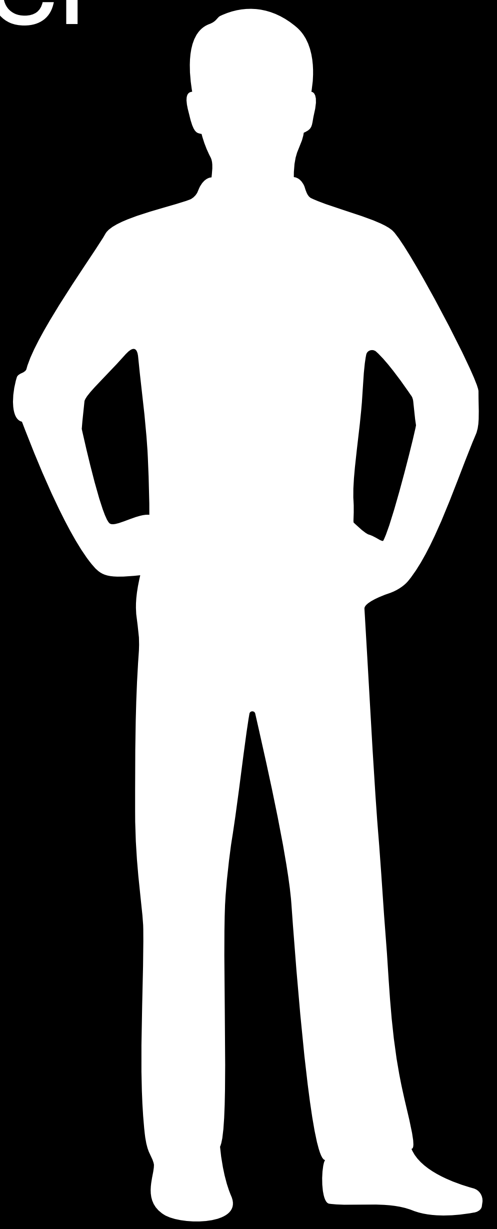
- Wikipedia: *“thing capable of an independent existence that can be uniquely identified”*
- Can be physical or logical
    - house/ car
    - house sale/ car service

Entity vs Entity Type

- Although the term is *entity* is most commonly used, we must distinguish between an **entity** and an **entity-type**
- **Entity-type** is a category
- **Entity** is an instance of a given entity-type
    - many such instances generally exist

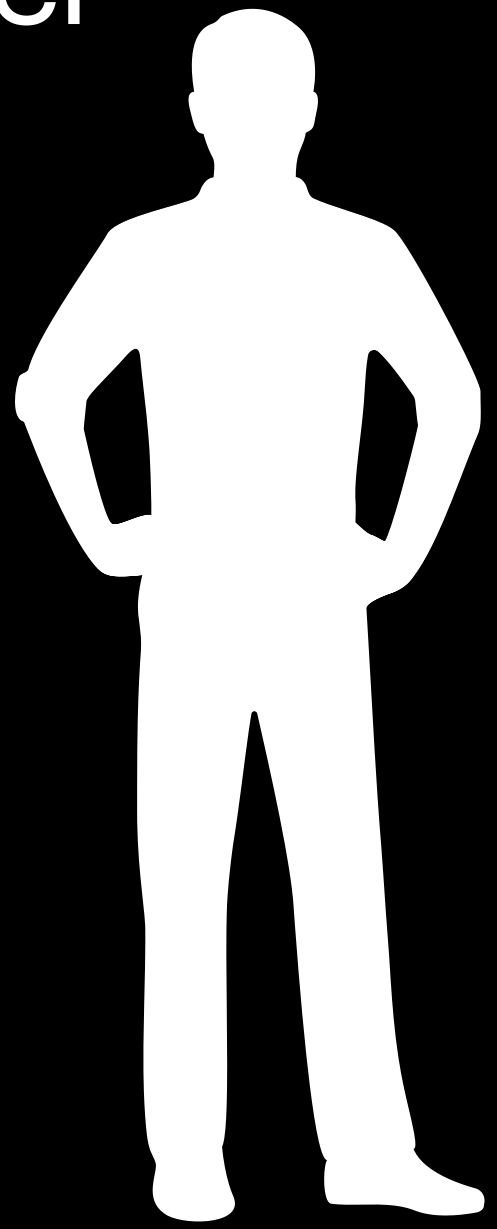

Teacher
Student
Book

Teacher



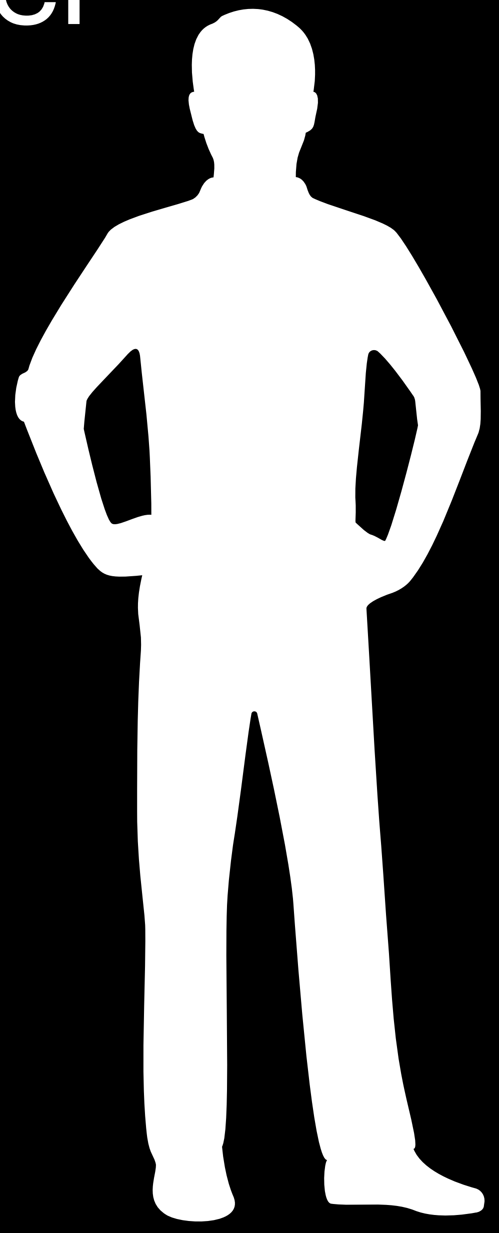
- Name
- Date of Birth
- Age
- Phone number
- Salary

Teacher

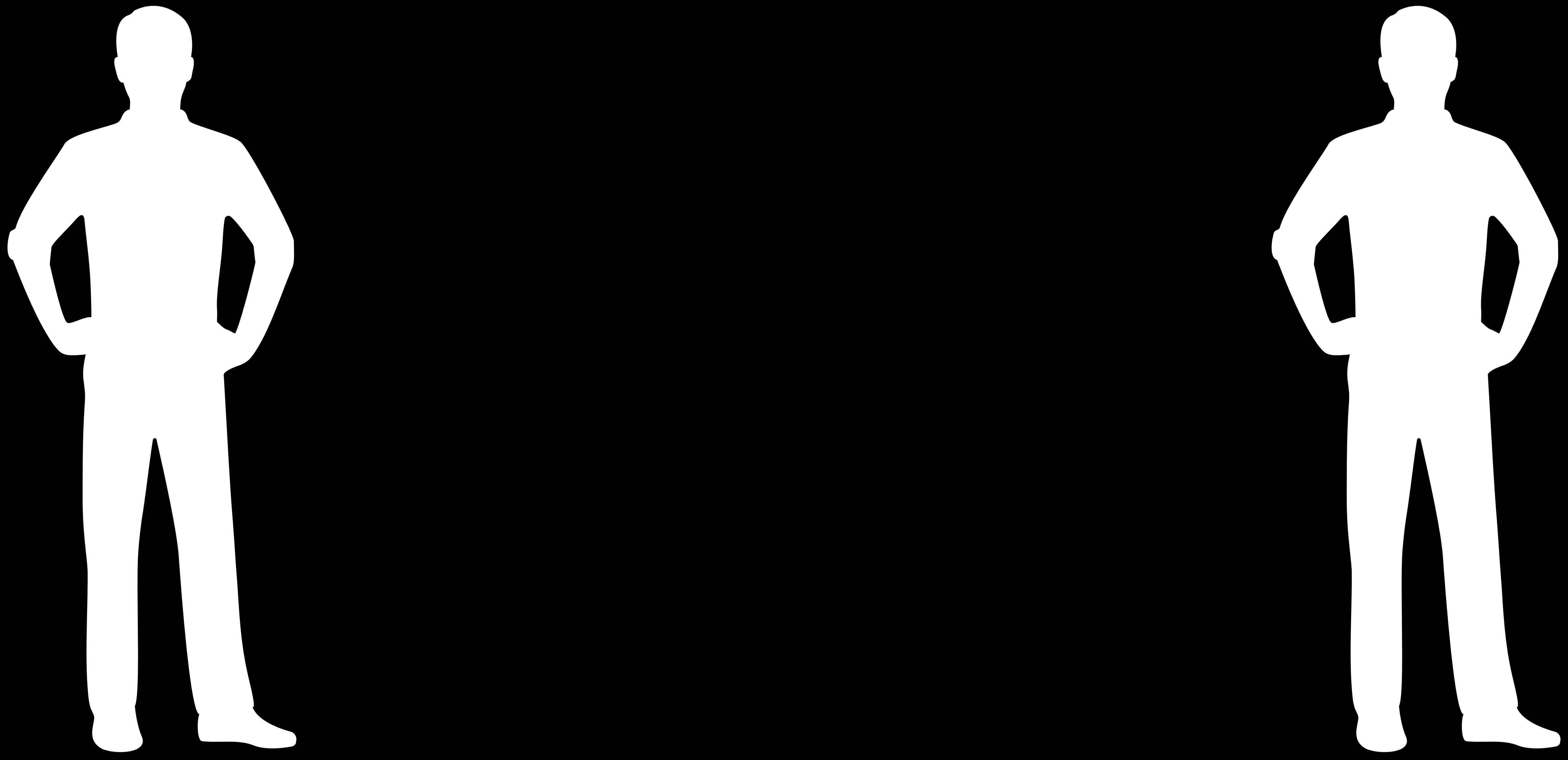


- Name
  - First Name
  - Last Name
- Date of Birth
- Age (can be derived from DoB)
- Phone number (can have multiple)
- Salary
- Emergency Contacts (comprised of other peo

Teacher

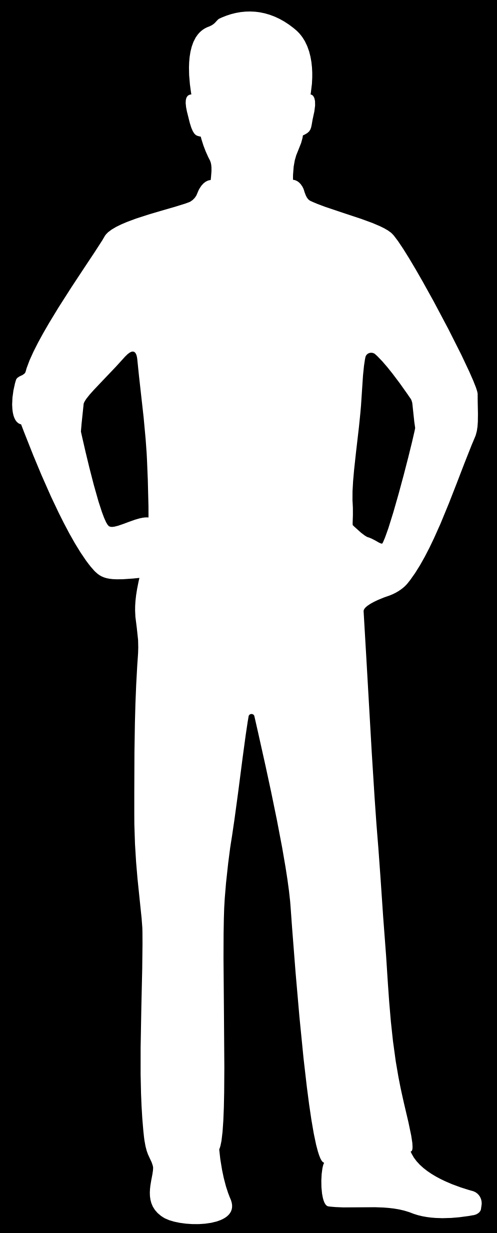


- Name [Composite Attribute]
  - First Name
  - Last Name
- Date of Birth
- Age (can be derived from DoB) [Derived Attribute]
- Phone number (can have multiple) [Multivalued Attribute]
- Salary [Simple Attribute]
- Emergency Contacts [Complex Attribute]

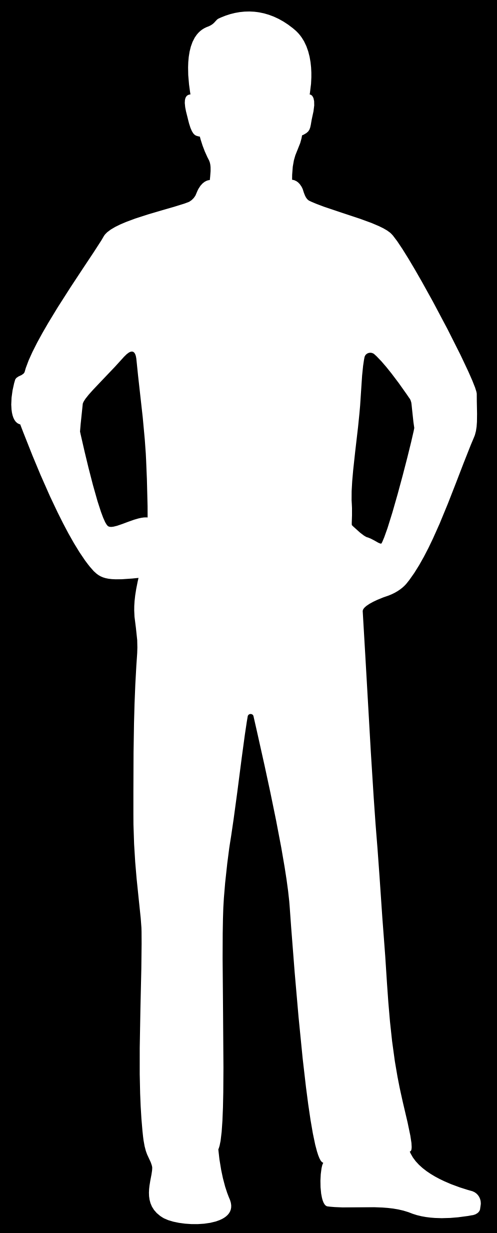


- How do we identify who is who?
- We need something to differentiate (uniquely identify) an entity

Anish



Anika



- How do we identify who is who?
- We need something to differentiate (uniquely identify) an entity

[Key Attribute] Can use phone number/ email ID/ employee ID, et cetera

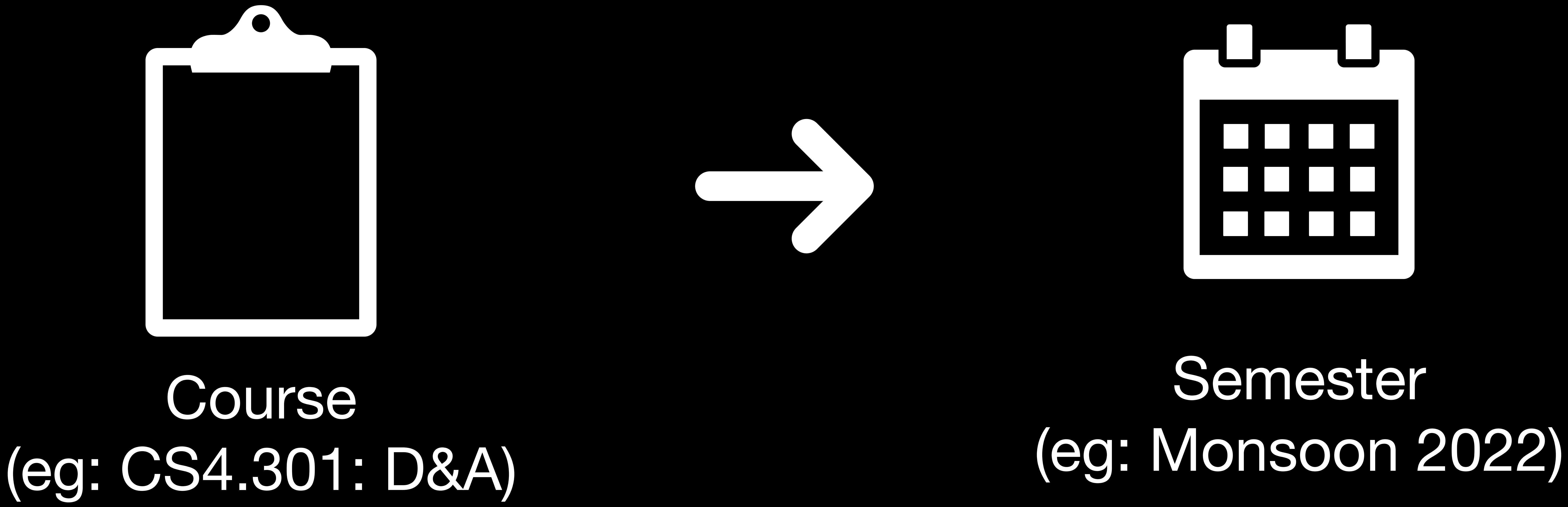
Weak Entity type

- Cannot be uniquely identified by its own attributes alone (no primary key)
- Needs another entity type to identify it uniquely (its own attributes + key attributes of the strong entity it depends on)

Weak Entity type

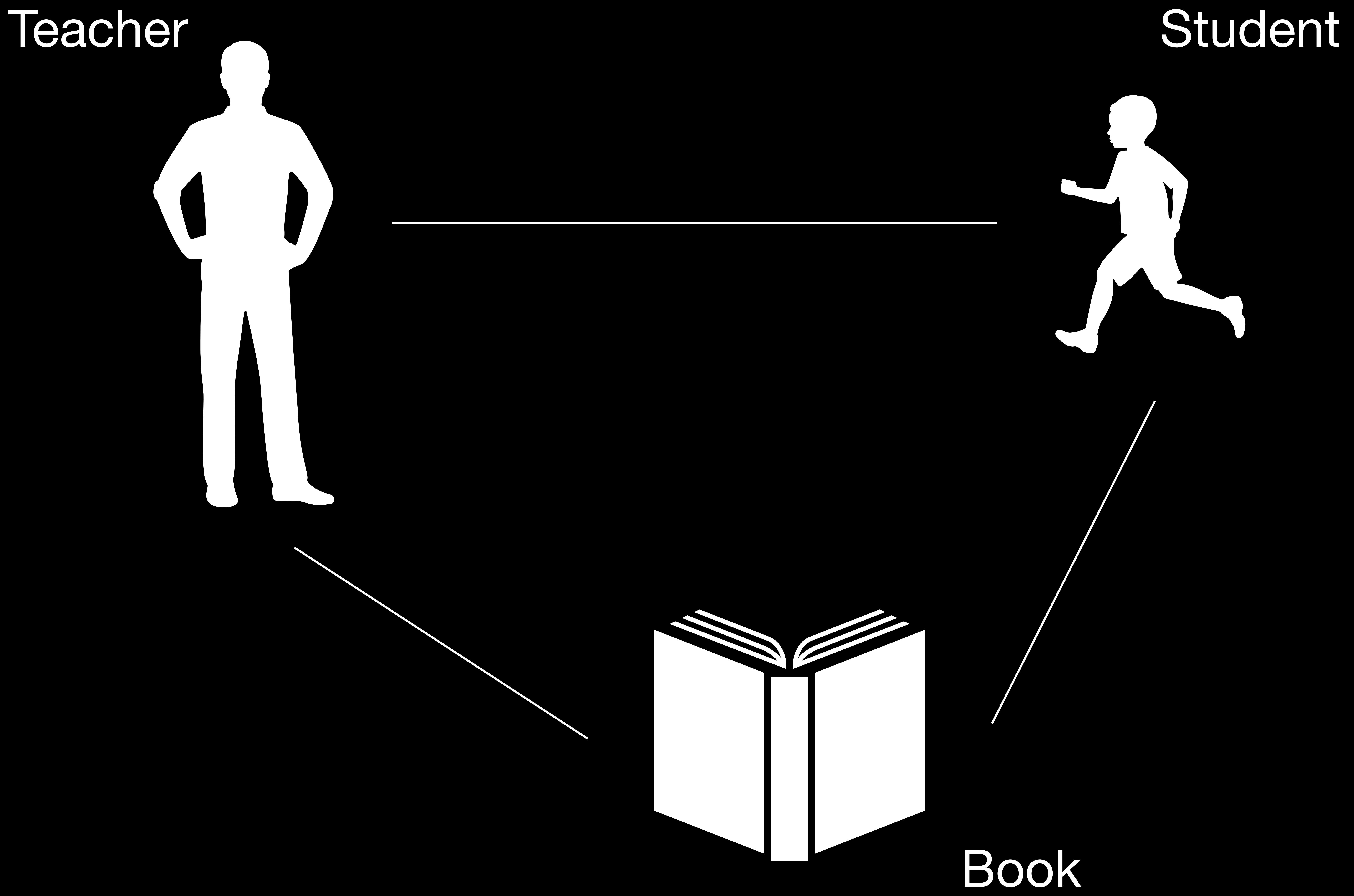
- Partial keys: Attribute of a weak entity that helps distinguish between entities dependent on the same strong entity

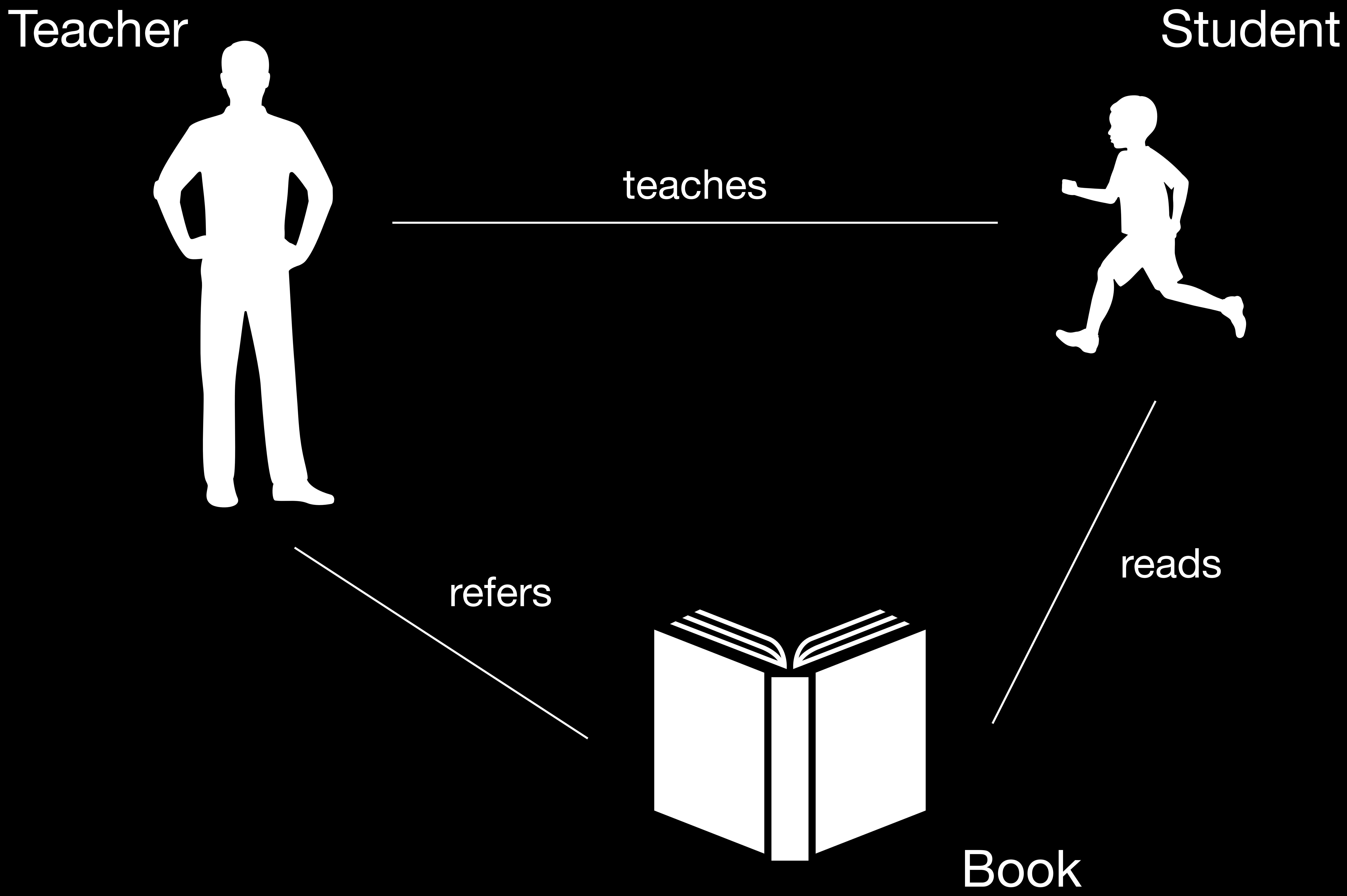
- No hard rule that some entity must be strong/weak. Depends on your design.

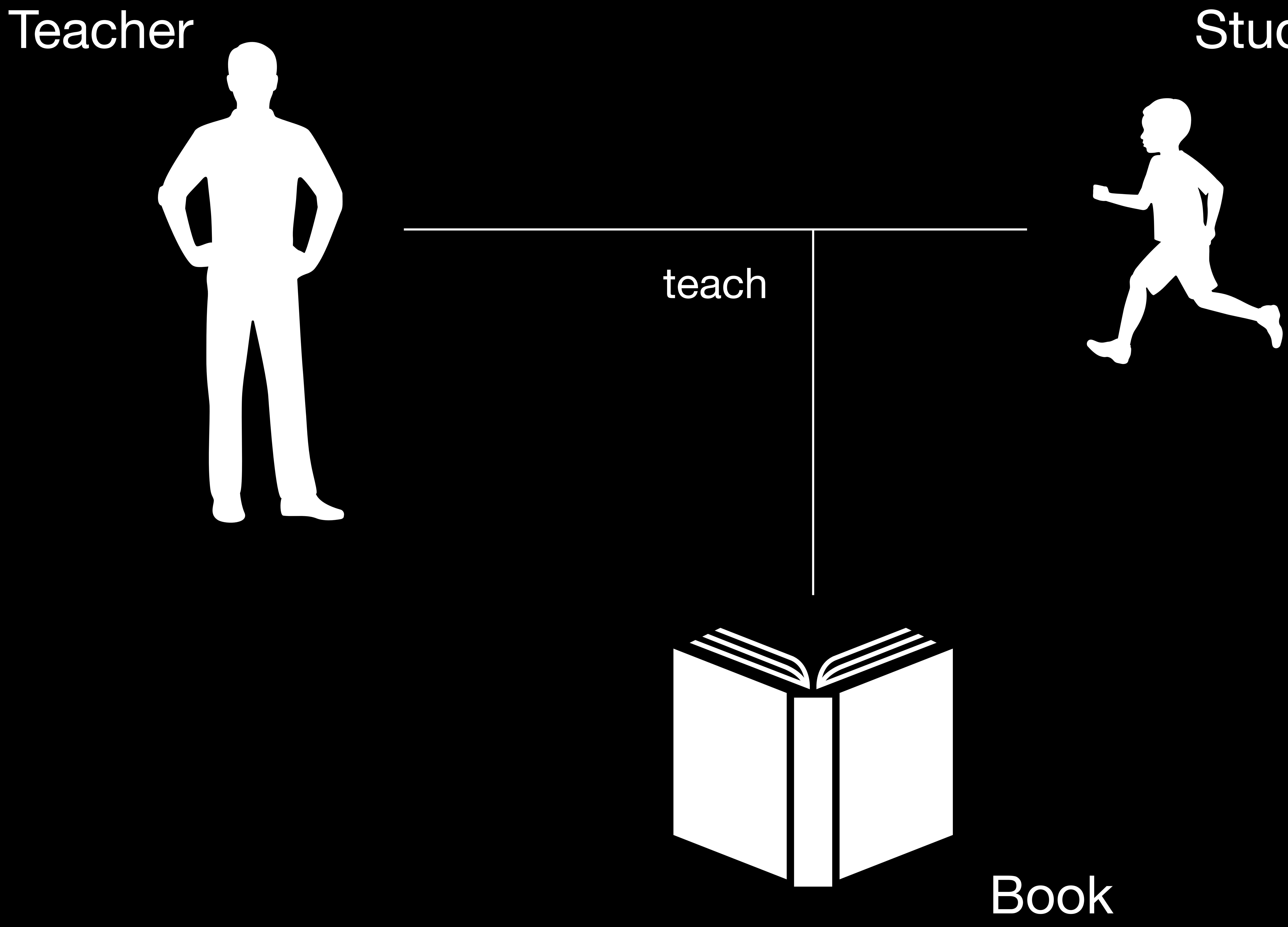


Relationship & Relationship type

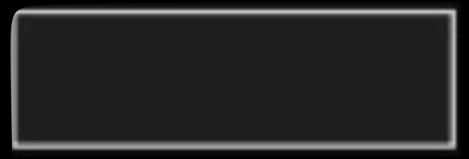
- Similar to entity vs entity-type: relationship-type is a category and relationship is an instance of a relationship-type
- A relationship-type gives a relationship between two (or more) entity-types
    - The entity-types are called as **roles** in this relationship-type

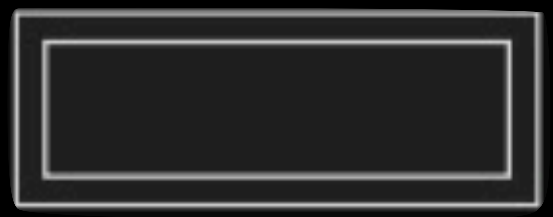

Teacher
Student
Book

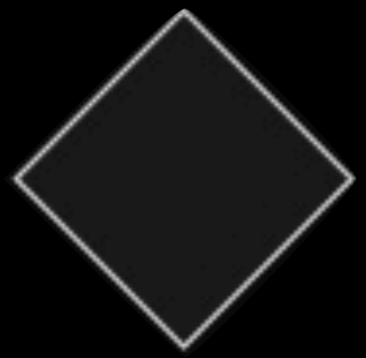

Teacher
Student
teaches
refers
reads
Book

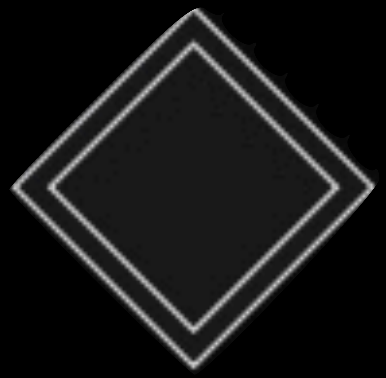

Teacher
Student
teach
Book

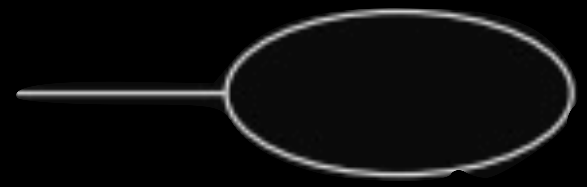
Notations

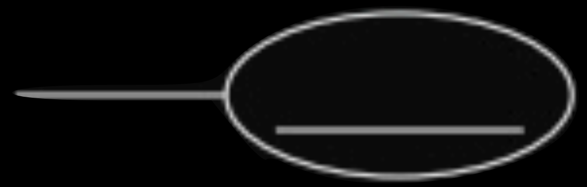

Entity

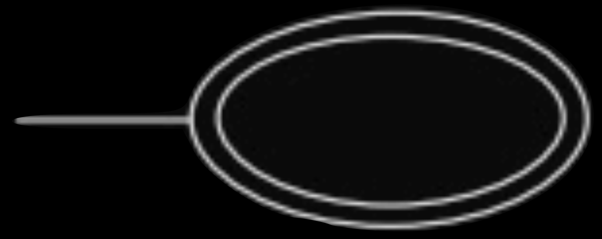

Weak Entity

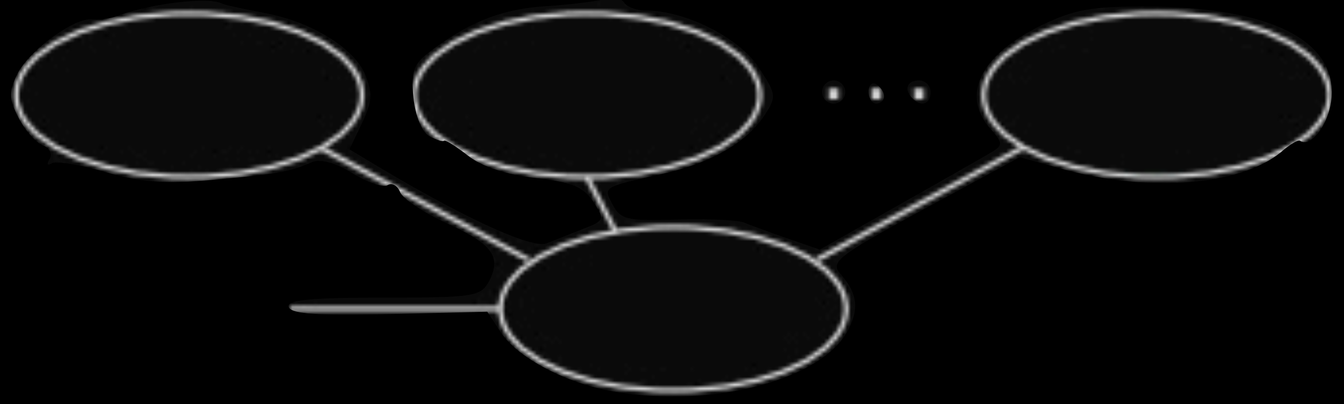

Relationship

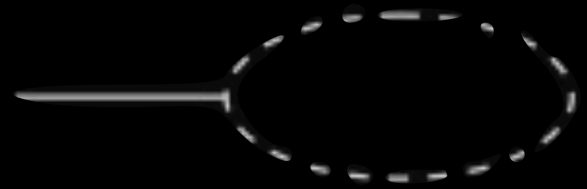

Indentifying Relationship


Attribute

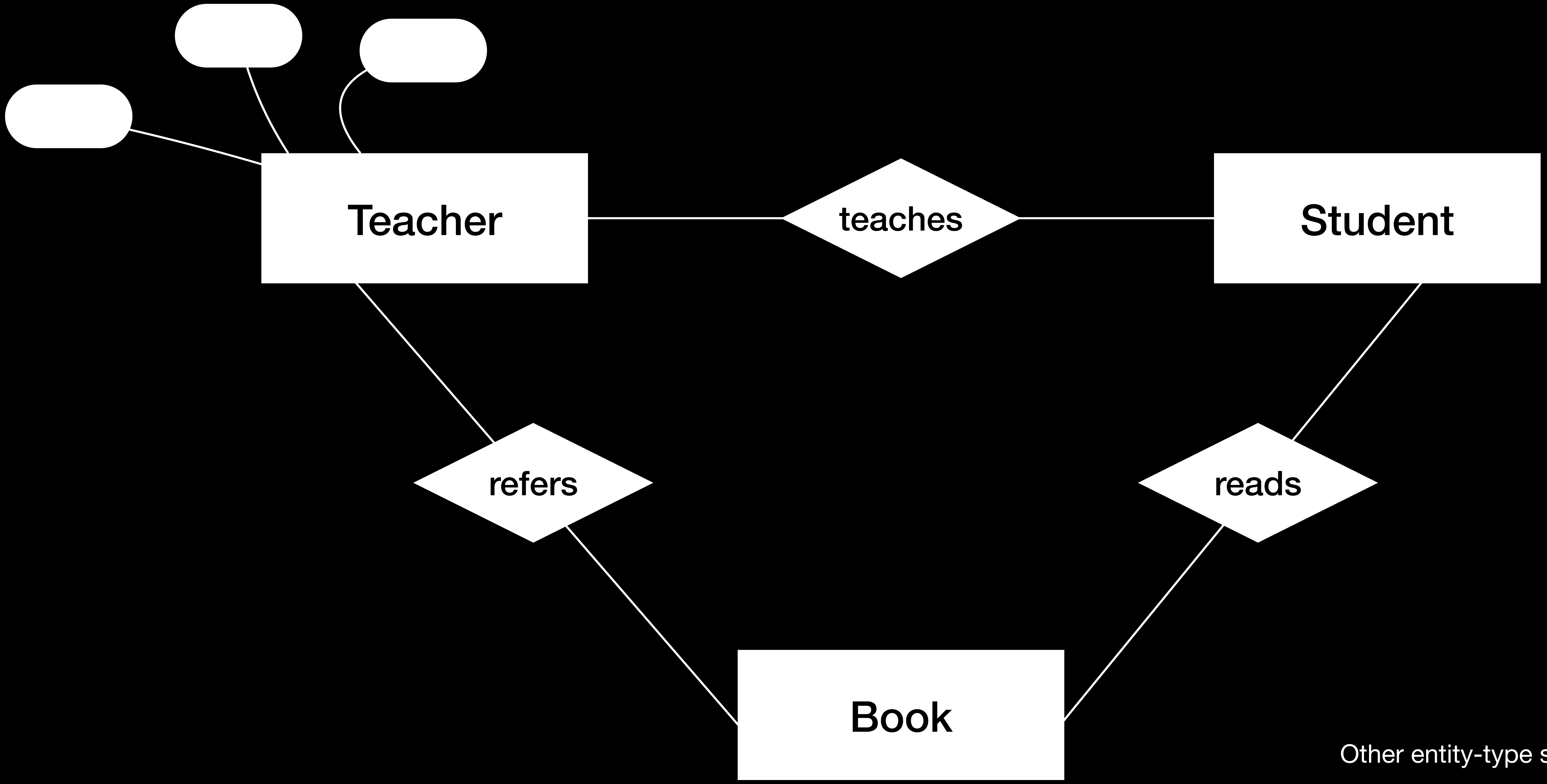

Key Attribute


Multivalued Attribute


Composite Attribute


Derived Attribute

Binary Relationships



Other entity-type sets will have attributes too

Ternary Relationship

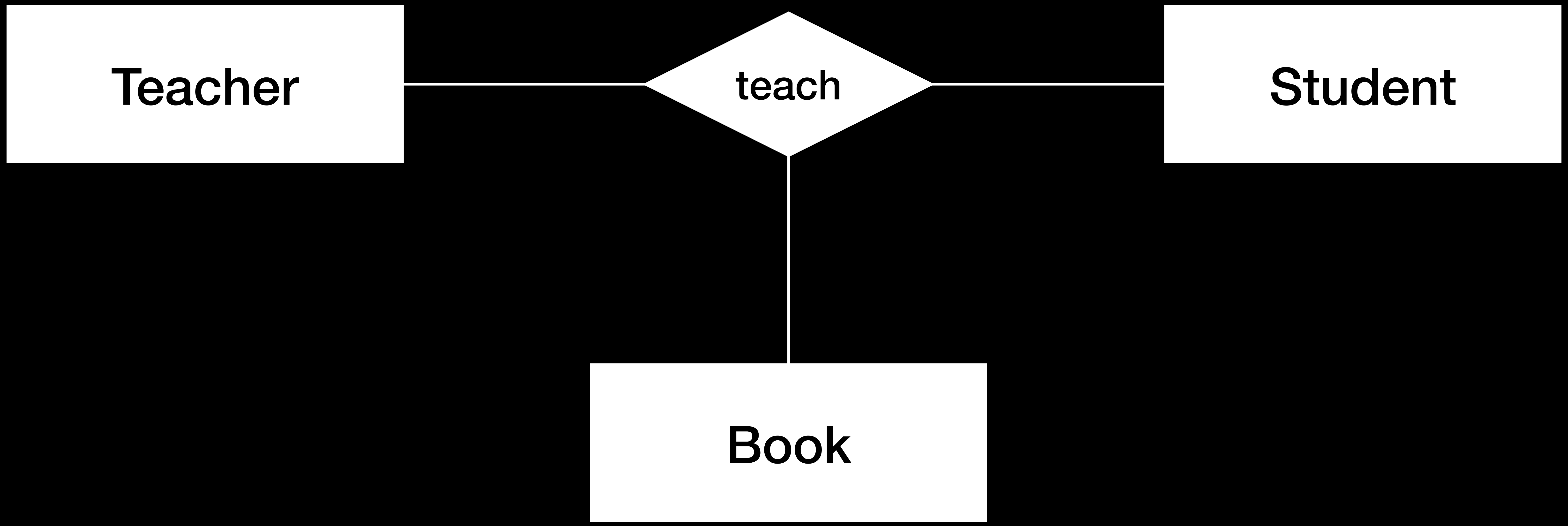
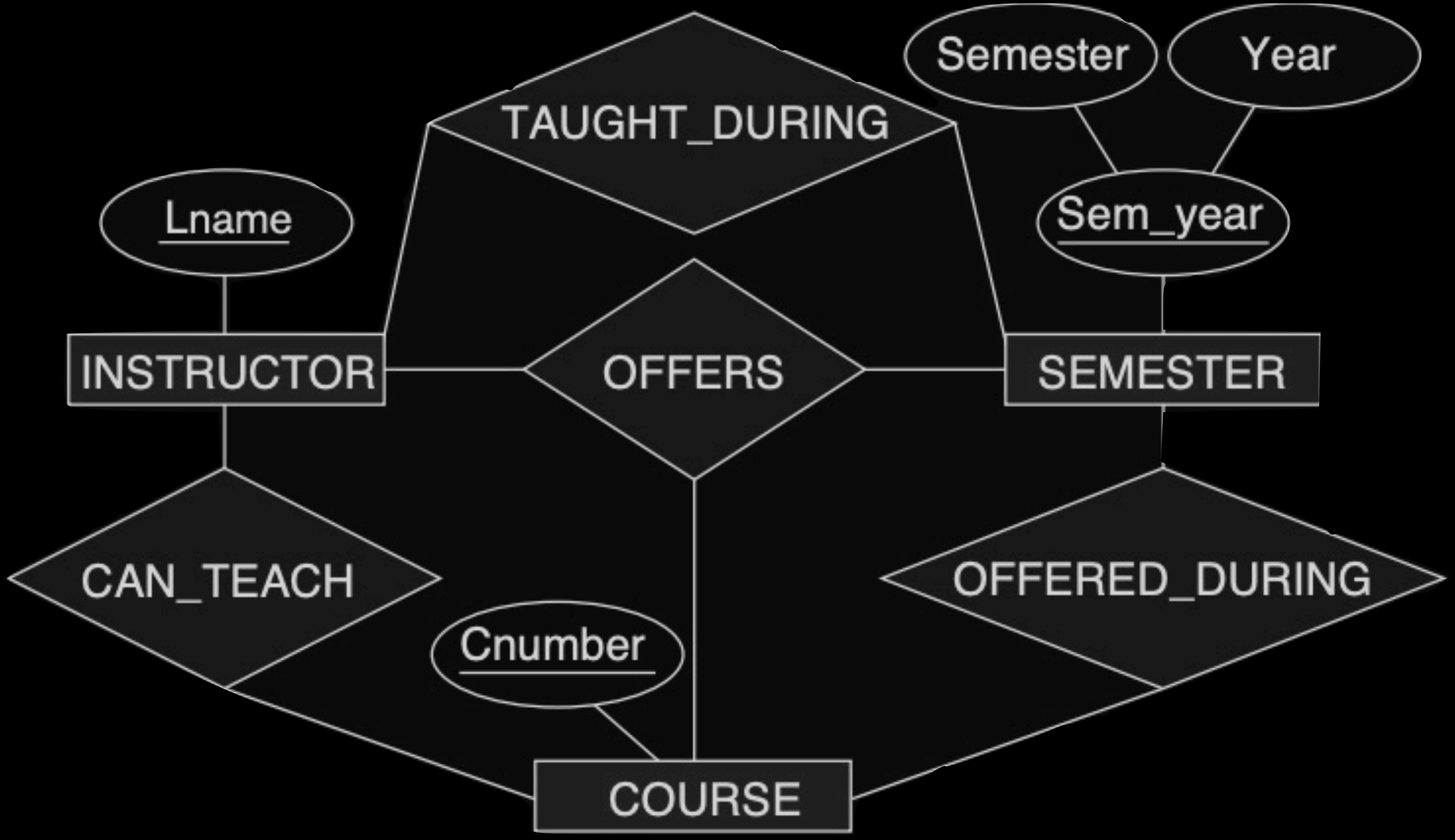
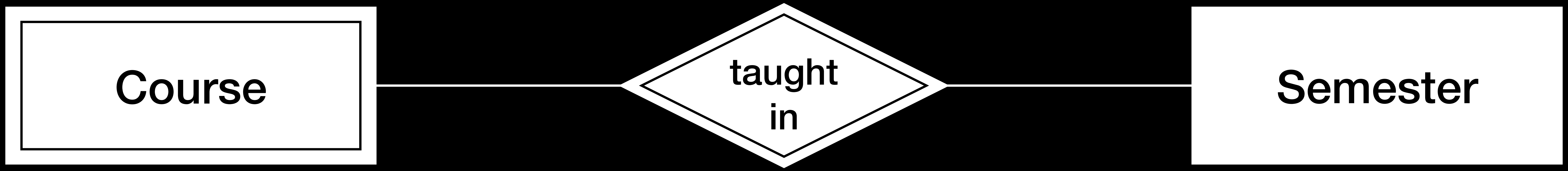


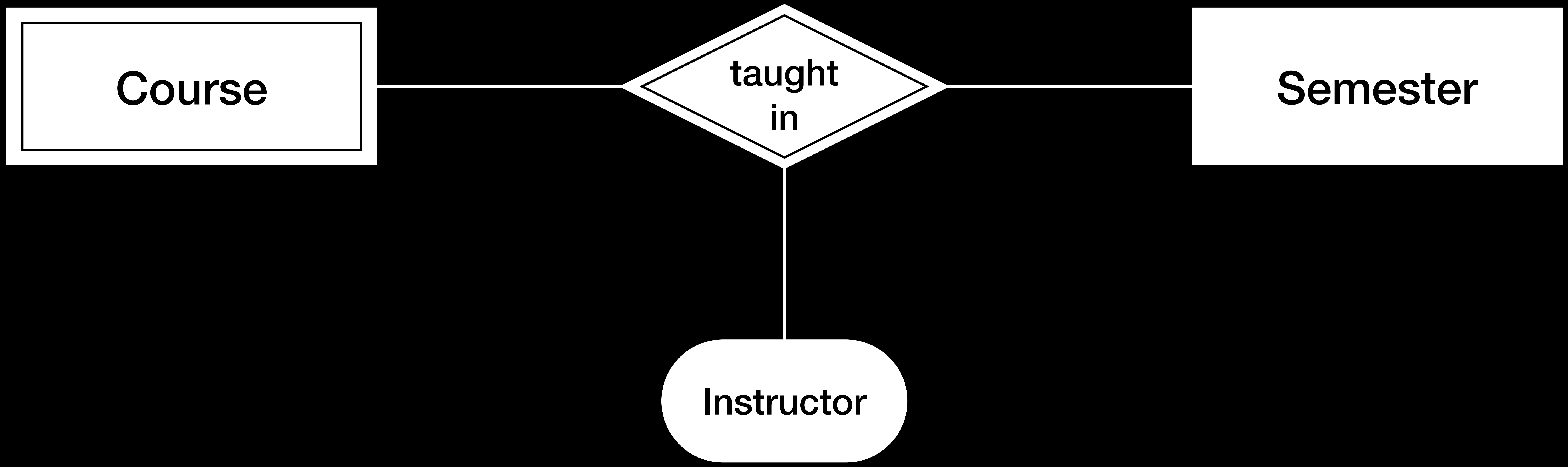
Figure 3.18
Another example of ternary versus binary relationship types.



Identifying Relationship



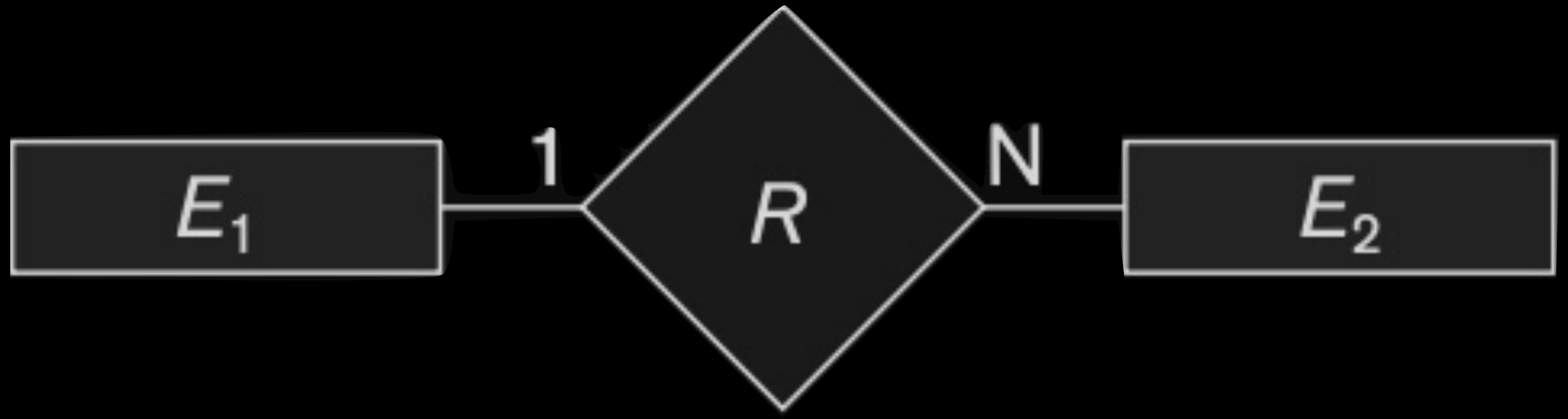
Relationship-types can have attributes!



Constraints on Relationship types

Cardinality Ratio

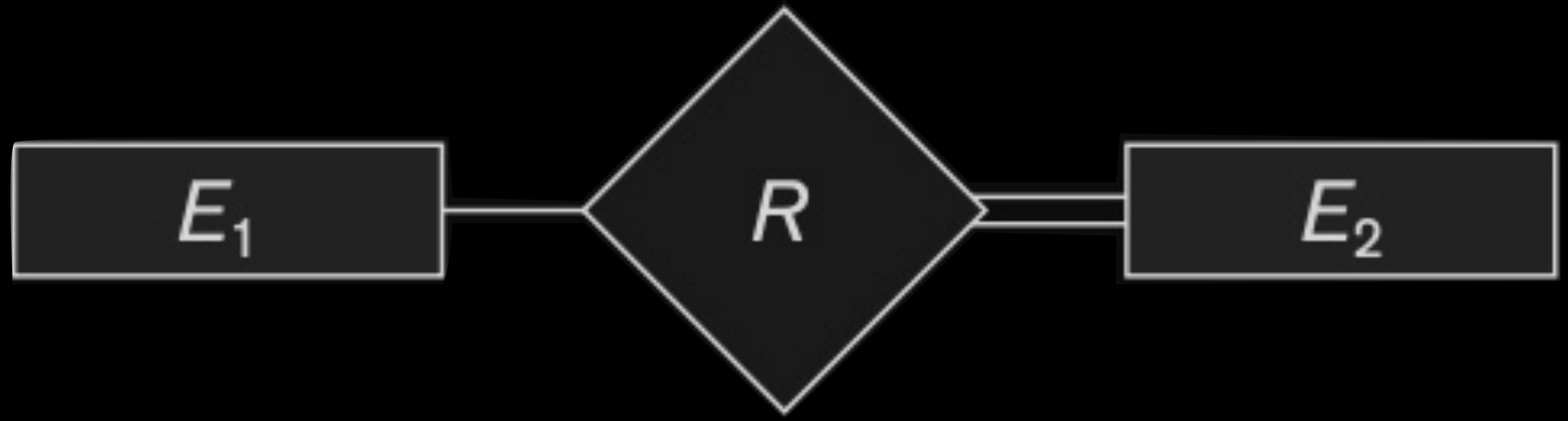
- Specifies the *maximum* number of relationship instances that an entity can participate in
  - 1:1
  - 1:N
  - N:1
  - M:N



Cardinality Ratio 1: N for $E_1 : E_2$ in R

Participation Constraint

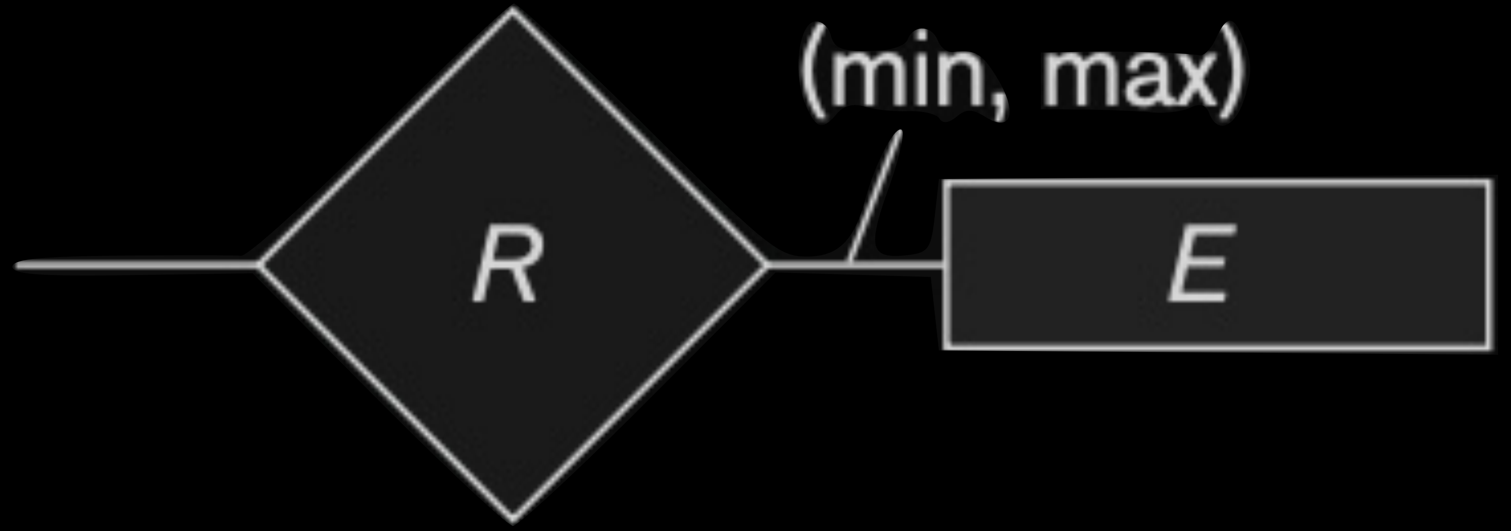
- Specifies whether the existence of an entity depends on its being related to another entity via the relationship type
- Specifies the *minimum* number of relationship instances that each entity must participate in
  - **total**: must participate in at least one relationship
  - **partial**: may/ may not participate in a relationship



Total Participation of E_2 in R

Alternative notation

- Problem arises that you can only mention min=1 and max=N with both cardinality ratios and partial/total participation
- Can user ***(min, max)*** notation
  - Each entity must participate in *min* and at most *max* relationships



Structural Constraint (min, max) on Participation of *E* in *R*

Example of a special type of relationship called **Self Relationship**

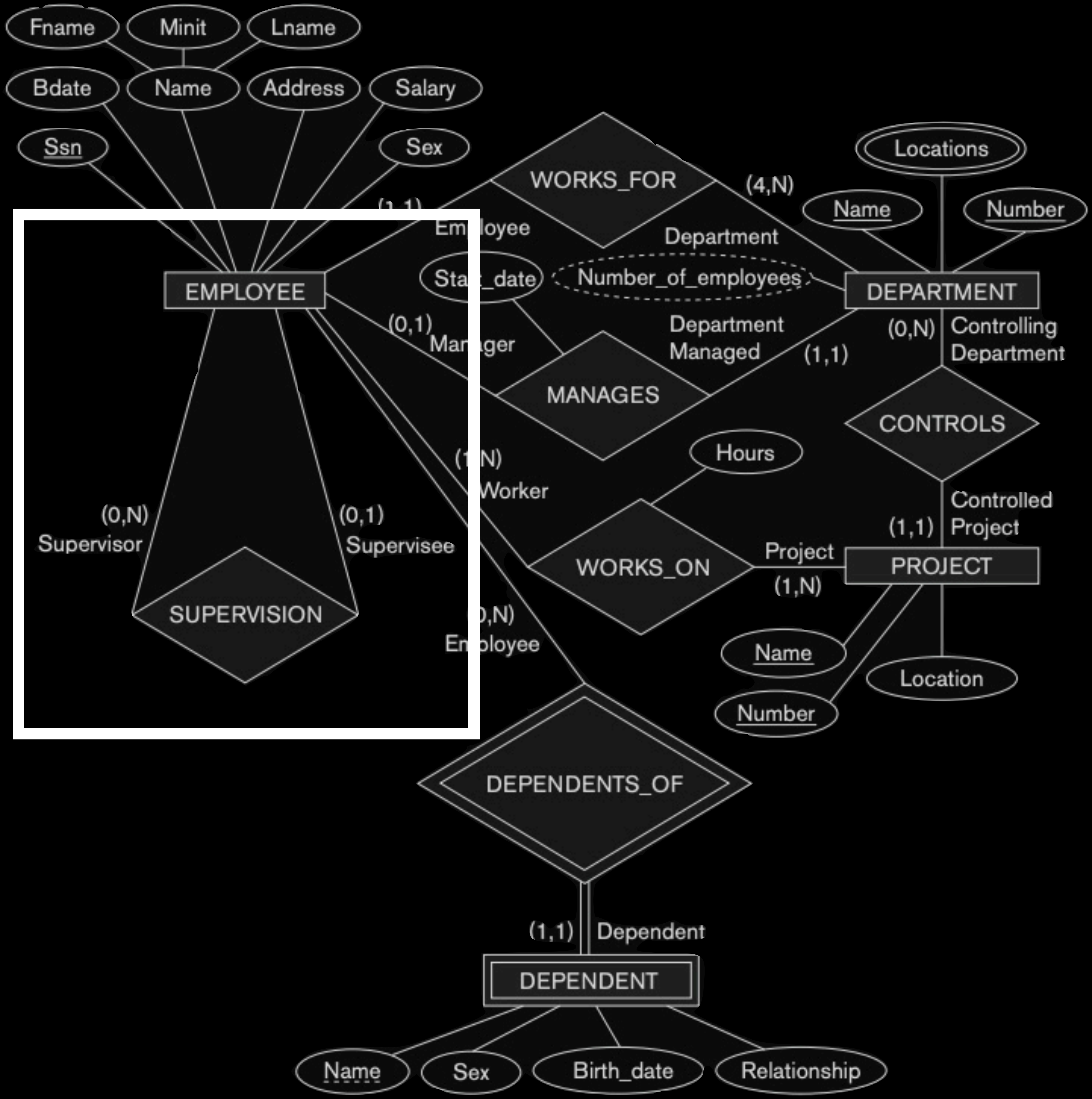


Figure 3.15
ER diagrams for the company schema, with structural constraints specified using (min, max) notation and role names.

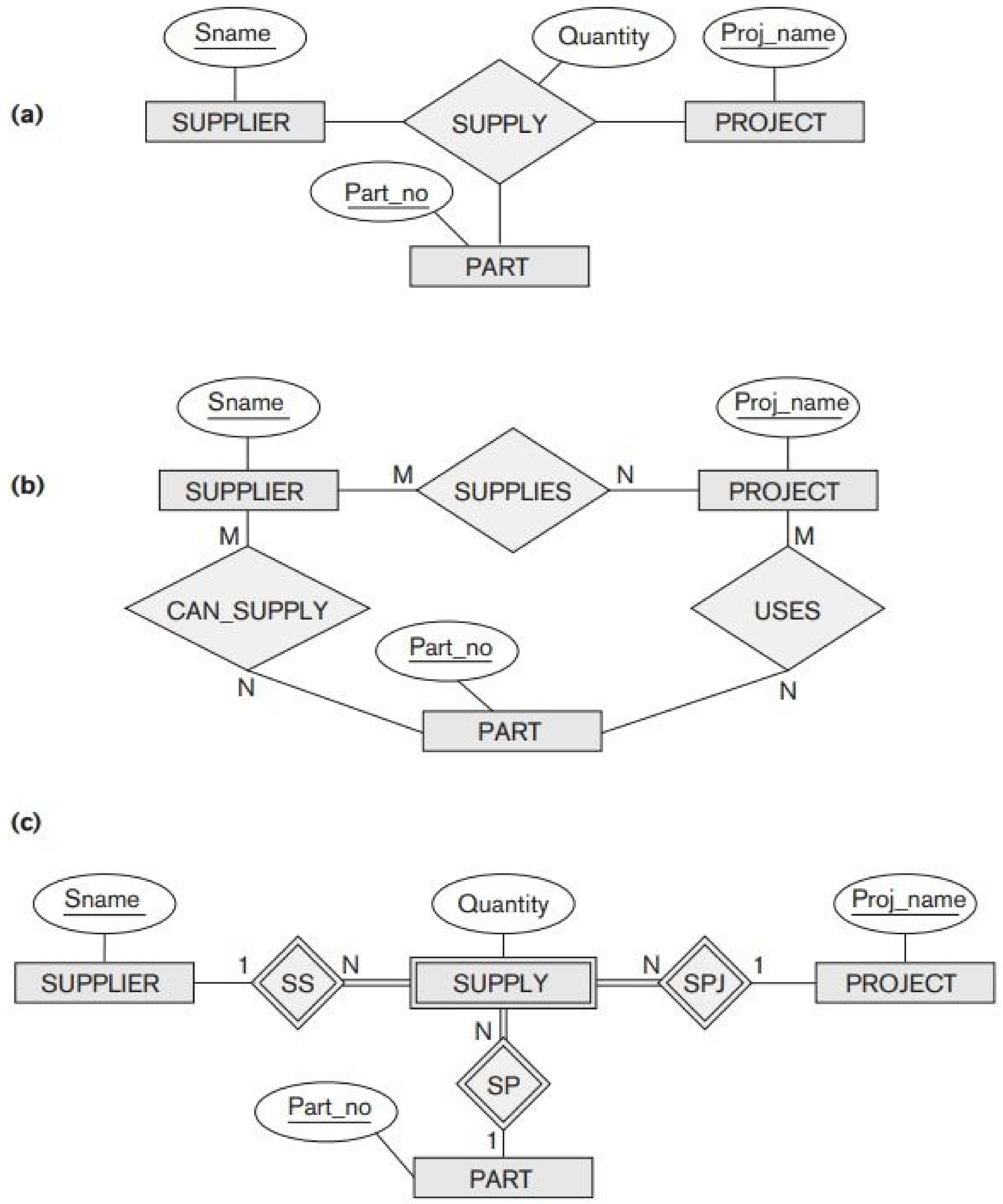


Figure 3.17
Ternary relationship types. (a) The SUPPLY relationship. (b) Three binary relationships not equivalent to SUPPLY. (c) SUPPLY represented as a weak entity type.

Homework 1

Requirements Documents

Objectives

- Define a mini-world
- Define the entity types of the mini-world
- Understand how they interact with each other
- Translate these interactions into relationships
- Define boundaries
- Define basic system behavior

Requirements Documents

Sections

- Introduction
  - define your mini-world, set boundaries
- Purpose of the DB
  - why does the DB exist? what does it offer that non-DB solutions don’t?
- Users of the DB
  - who uses it? what do they do with it?
- Applications of the DB
  - what all applications exist for your DB in the given mini-world?

Requirements Documents

Sections

- Database Requirements
- Functional Requirements
  - Descriptions of data to be entered into the system
  - Descriptions of system reports or other outputs
  - Access control
  - For this course, functional requirements should relate to the tasks that the database system will perform — usually in the form of **access**, **searching**, **reports** and **sorting** (queries). FRs may also provide details around the data that must be stored in the DB.

Administrative stuff

- Teams
  - 4 people in a team— already released on Moodle
  - will remain same throughout this course
- Approaching TAs
  - TA office hours shared on Moodle
  - mailing list shared on Moodle
  - be formal — WhatsApp messages will be ignored
- Late Days
  - 8 late days in total, 8 submissions (4 HW + 4 Project phases) in total
  - Max of 2 can be used on 1 submission