

CCT396, Fall 2011

# Database Design and Implementation

Yuri Takhteyev  
University of Toronto



This presentation is licensed under Creative Commons Attribution License, v. 3.0. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/>. This presentation incorporates images from the Crystal Clear icon collection by Everaldo Coelho, available under LGPL from <http://everaldo.com/crystal/>.

**Week 3**

**The ER Model**

# The Relational Model

species

Human	humanoid	1.7
Hutt	gastropod	3.5

persona

Jabba	Hutt
Obiwan Kenobi	Human

species\_type

gastropod	0
humanoid	2

# Good DB Design

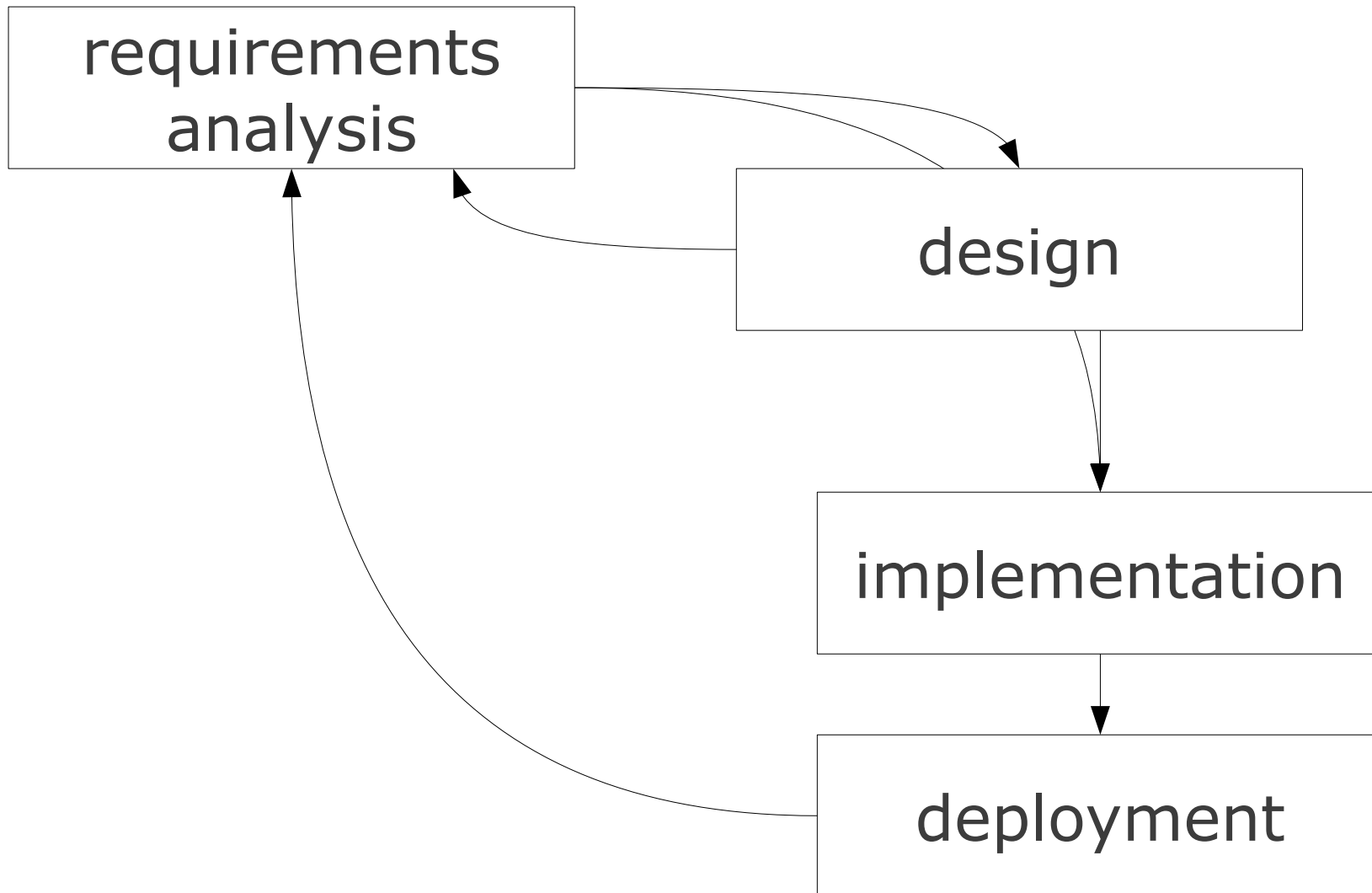
1. Serves the purpose.
2. Simplifies updates.
3. Avoids “anomalies”.

# Two Approaches

1. Start with whatever, then fix it
2. Start by modeling

## ER Modeling

# The DB Lifecycle



client

you

database

users



# Functional Specifications

What is the system going to do?  
(Not **how**, but **what**.)



# Course Enrollment

What do we want from a course enrollment system?

# “Use Cases”

A student wants to enroll in a course for the next semester. The student goes to the website for the enrollment system and logs in. The student is presented with a list of courses there he or she is enrolled in. There is a button next to each course to un-enroll from it. There is also a search box that the student can use to look for courses that they are not enrolled in. The student can find the course by the course code or course title...

# ER Model

E is for “Entities”

R is for “Relationships”

(“relationships” != “relations”)

# “Use Cases”

A **student** wants to **enroll** in a **course** for the next semester. The student goes to the website for the enrollment system and logs in. The student is presented with a list of courses there he or she is enrolled in. There is a button next to each course to un-enroll from it. There is also a search box that the student can use to look for courses that they are not enrolled in. The student can find the course by the course code or course title...

# Entities

The “things” we need to keep track of in our database:

students

courses

instructors

rooms

time slots

attributes

last name

first name

utorid

date of birth

**student**

program

payment

course

other entities

# Entities vs Attributes

## **Attributes**

usually atomic values, e.g.  
a number, a string, a date

## **Entities**

bundles of values

(But are dates really atomic?)

# One Entity or Two?

The book and its author?

The book and its title?

The film and its director?

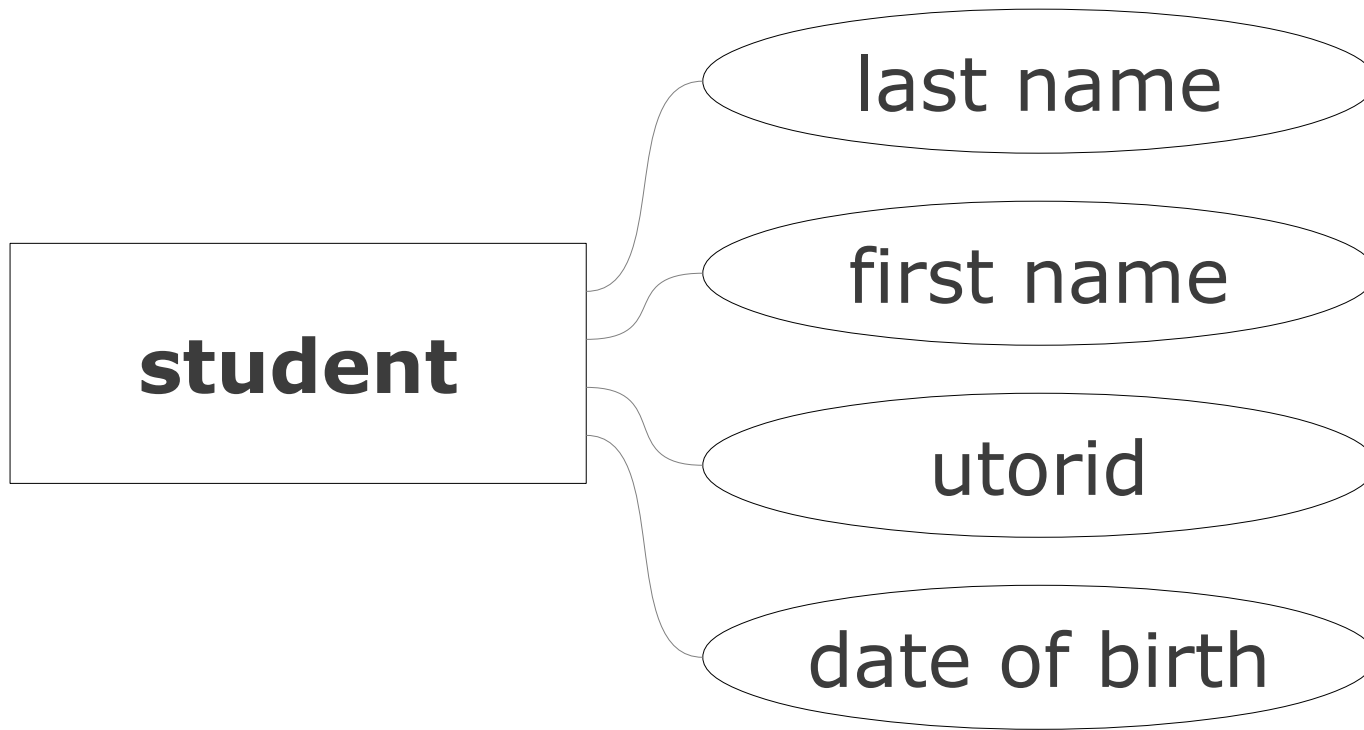
The film and its title?

The city and its population?

The city and its mayor?

The user and their account?





“Chen’s notation”

**student**

---

last name

first name

utorid

date of birth

“UML notation”  
(simplified)

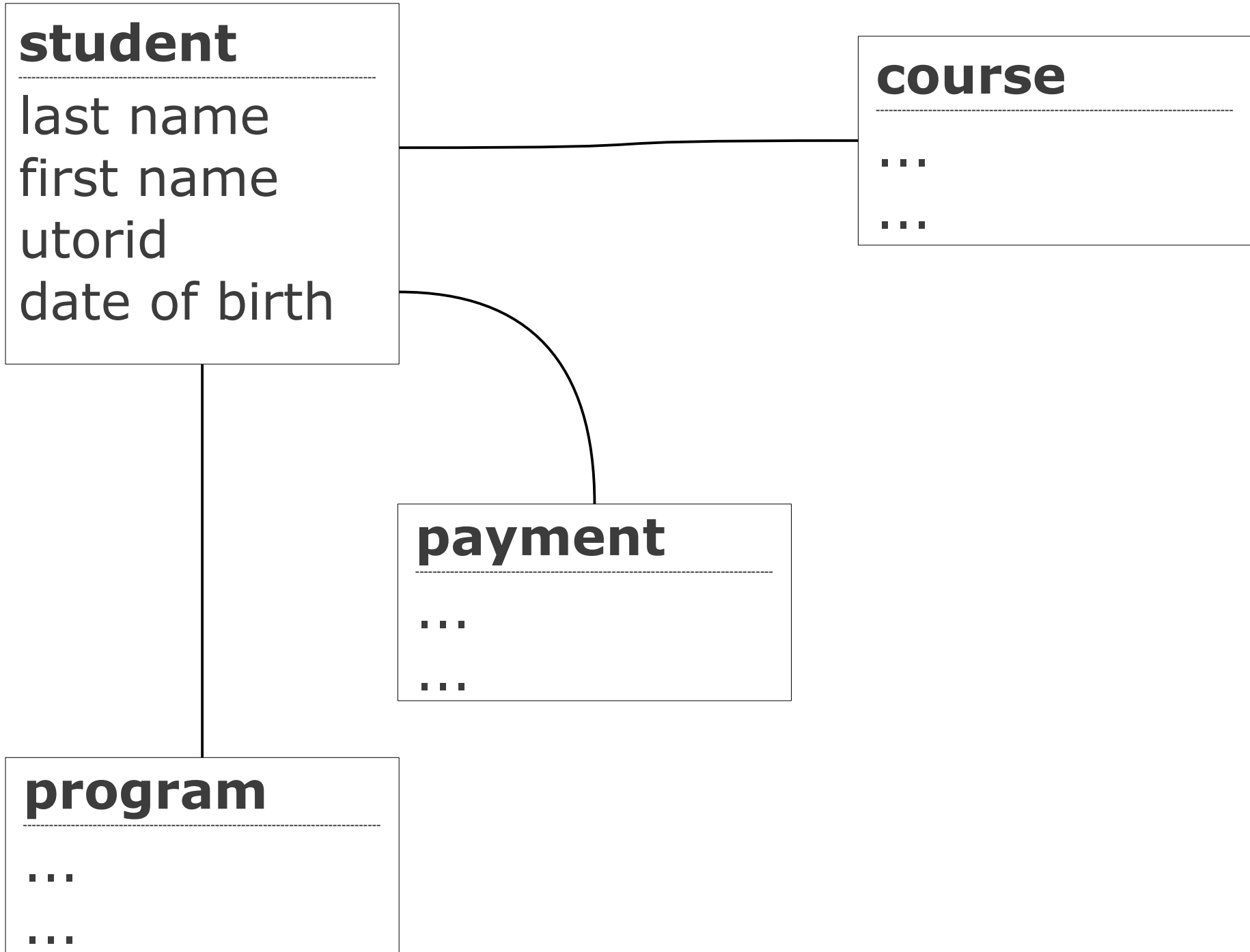
# Type vs Instance

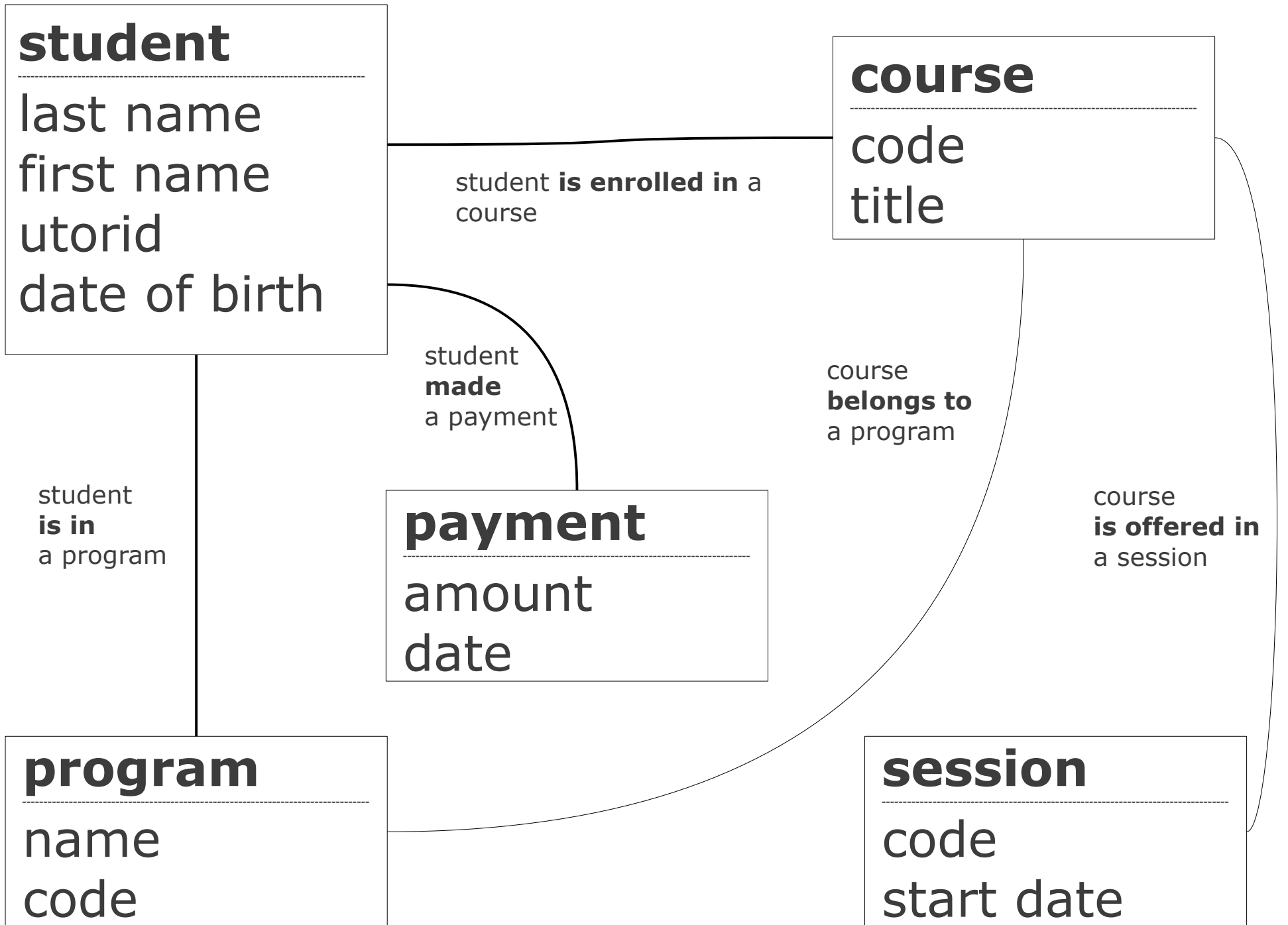
## **Entity as a type**

When we say “student” we really mean “students” as a type.

## **Instances of entities**

Individual students’ records are instances of the “student” entity.





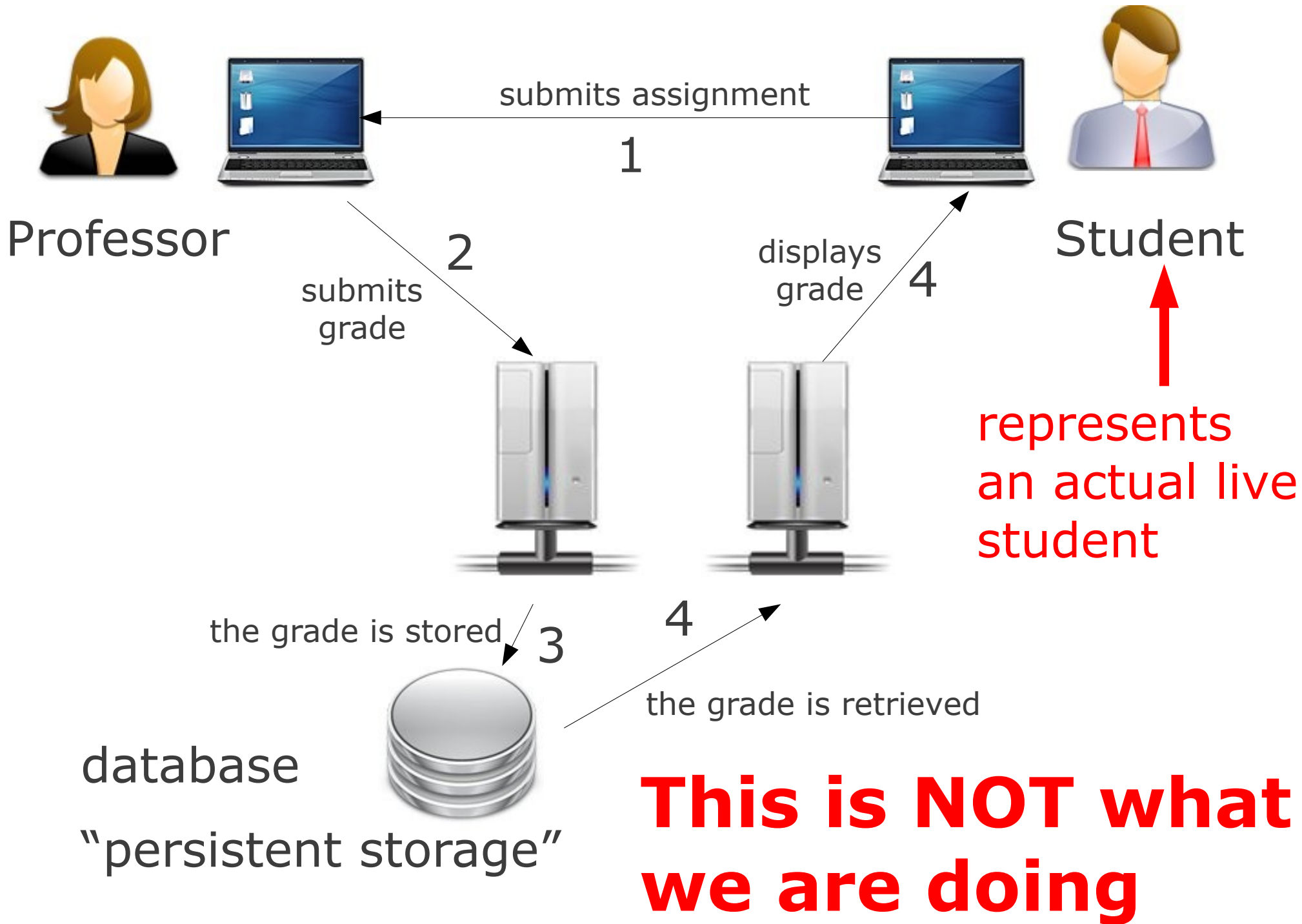
# ER vs Data Flow Diagrams

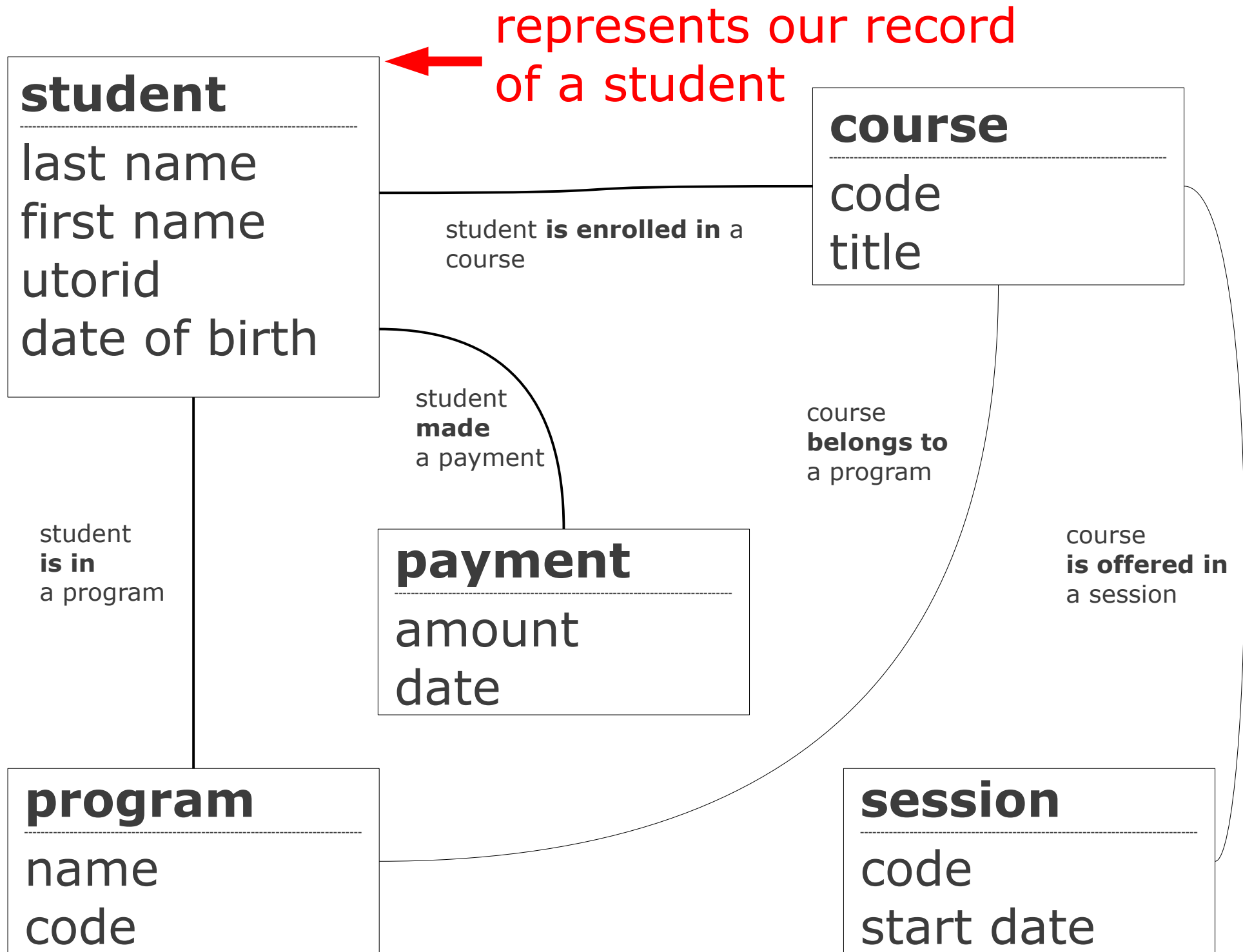
## **ER Diagrams**

The structure of the data itself

## **Data Flow Diagrams**

The flow of data between the different processing steps.







# Two Degrees

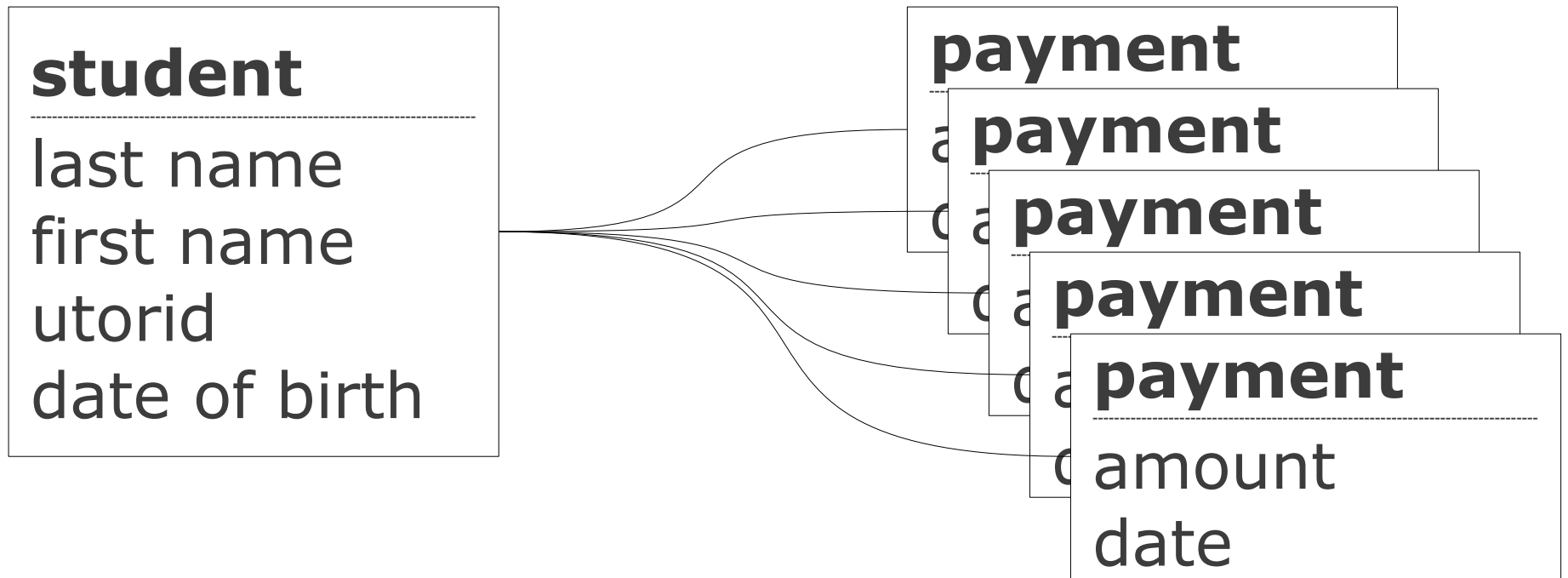
1. A box in an IR diagram represents a type of record.
2. Records stores information about things in the real world.

# A Movie Database

What are the entities?  
Which are related?



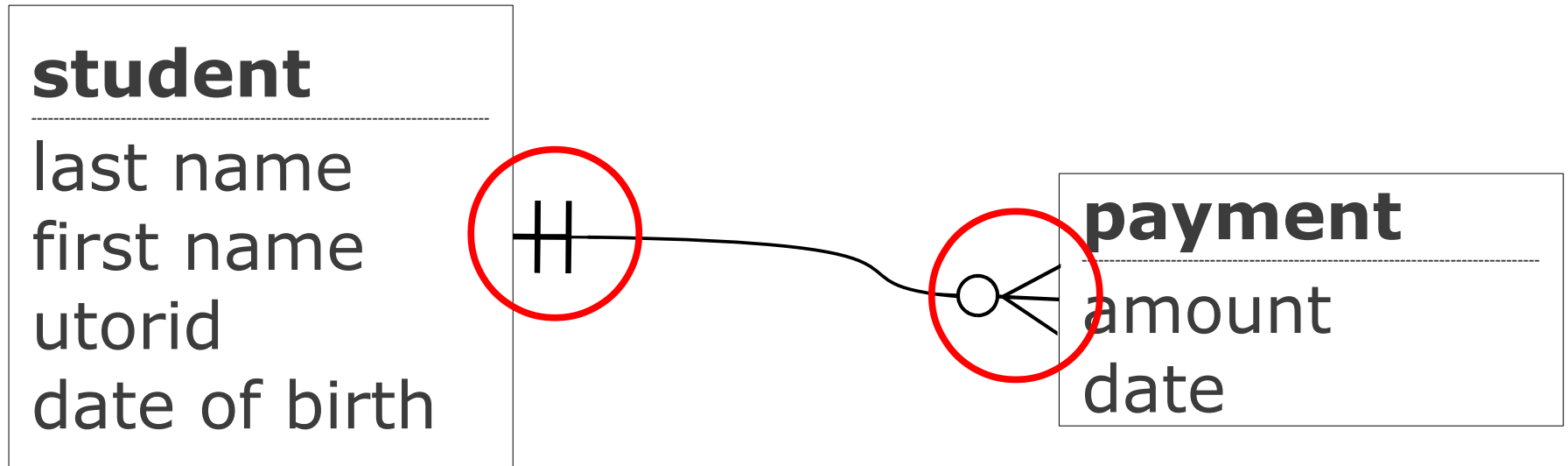
# Cardinality



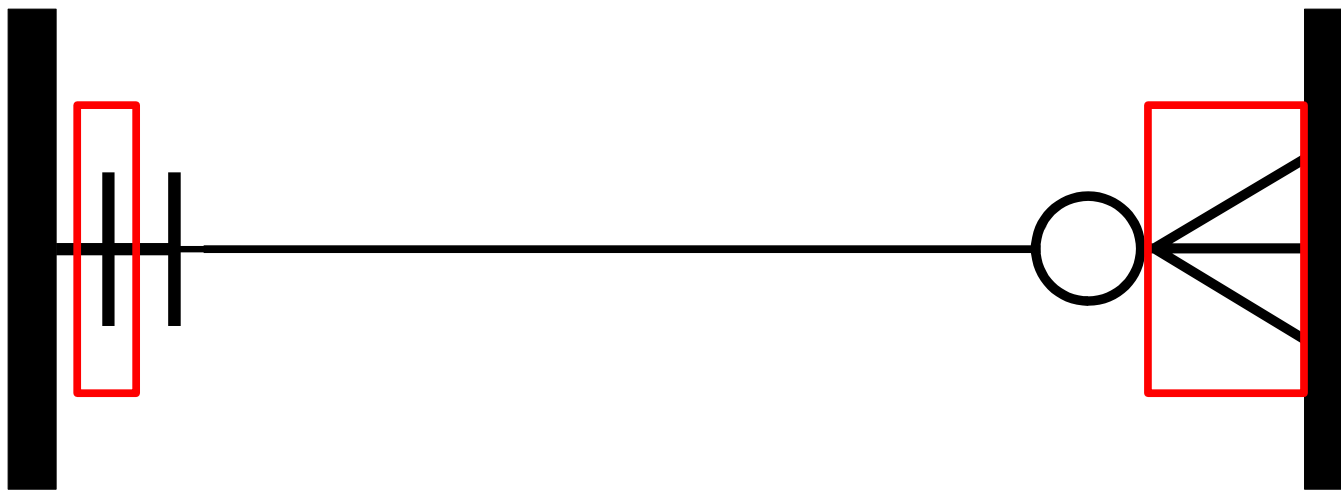
one student, many payments

“one-to-many” relationship

# “Crow’s Foot Notation”



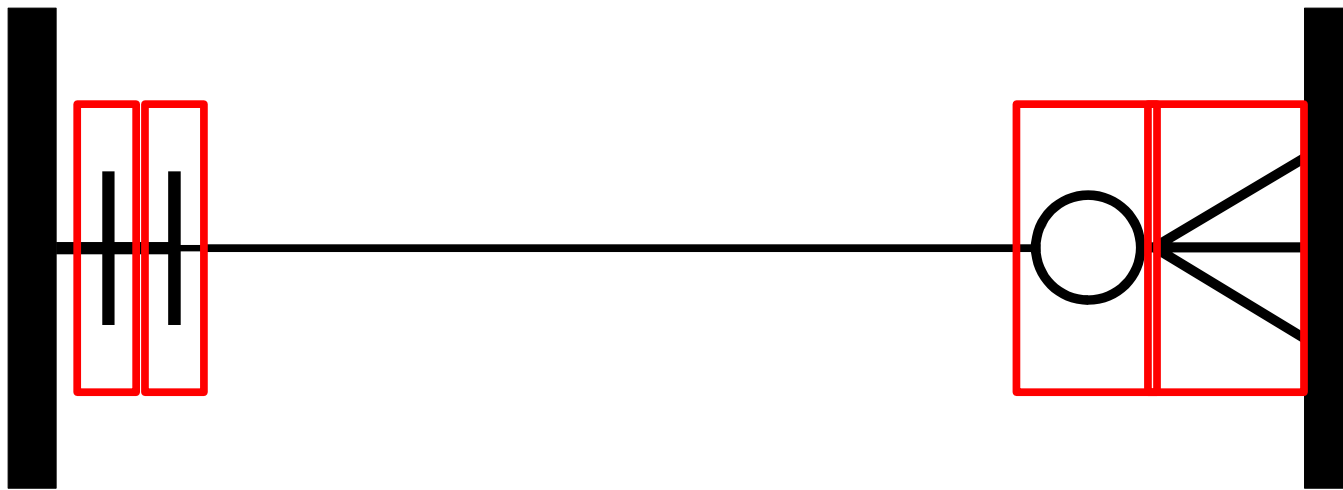
# “Crow’s Foot Notation”



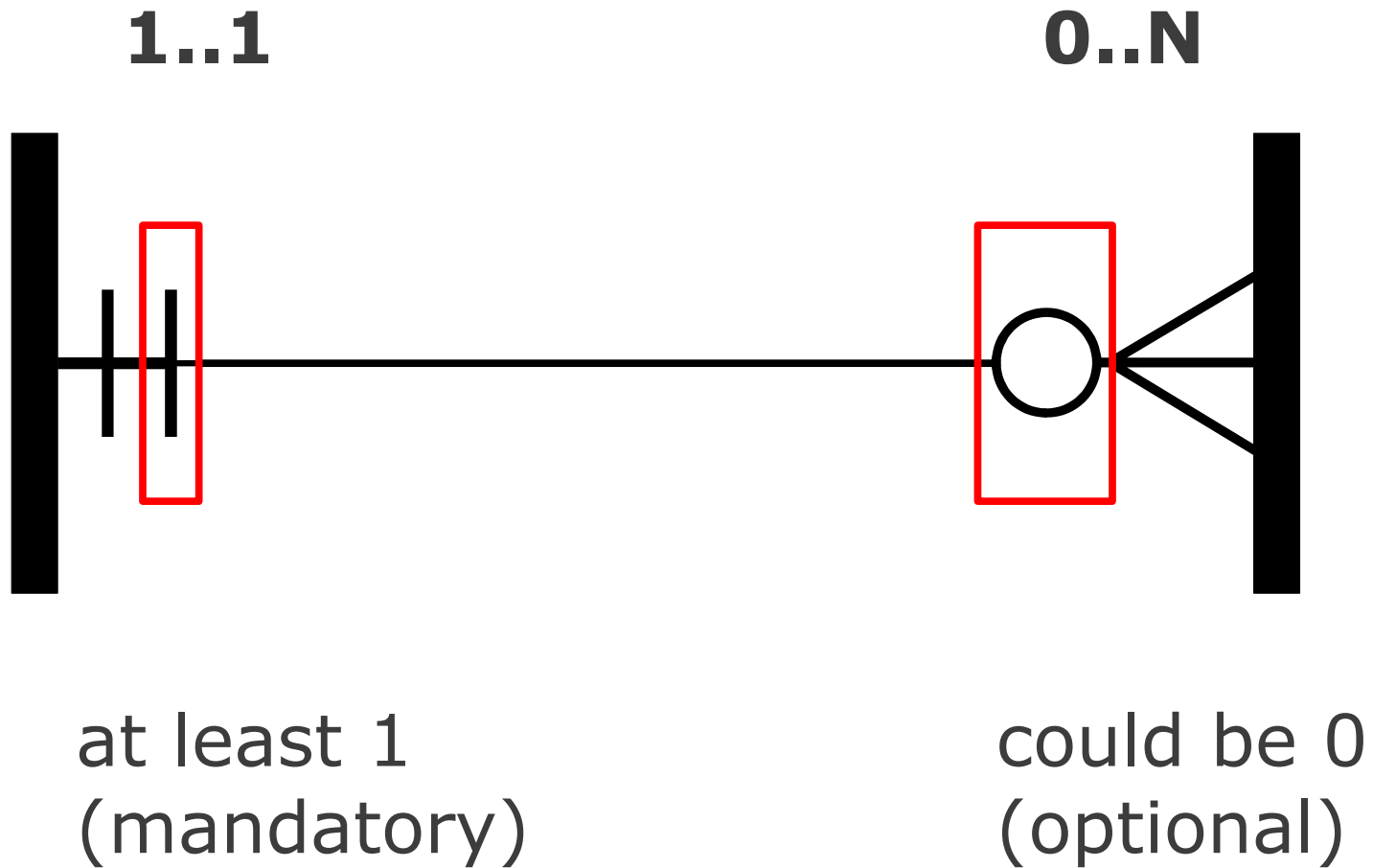
not more than 1

many

# “Crow’s Foot Notation”



# “Crow’s Foot Notation”





# “UML Notation”

**1..1**

**0..N**

**1..1**

**0..N**

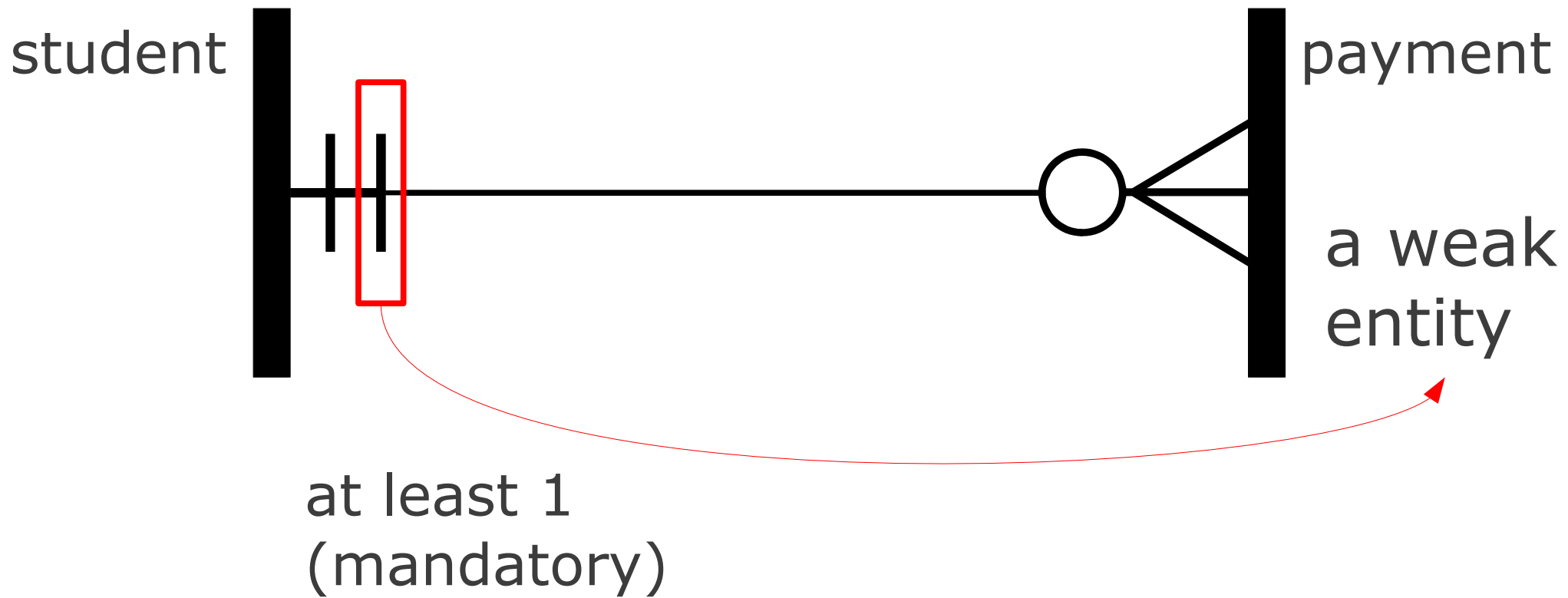
or just “1”

or “0..\*”

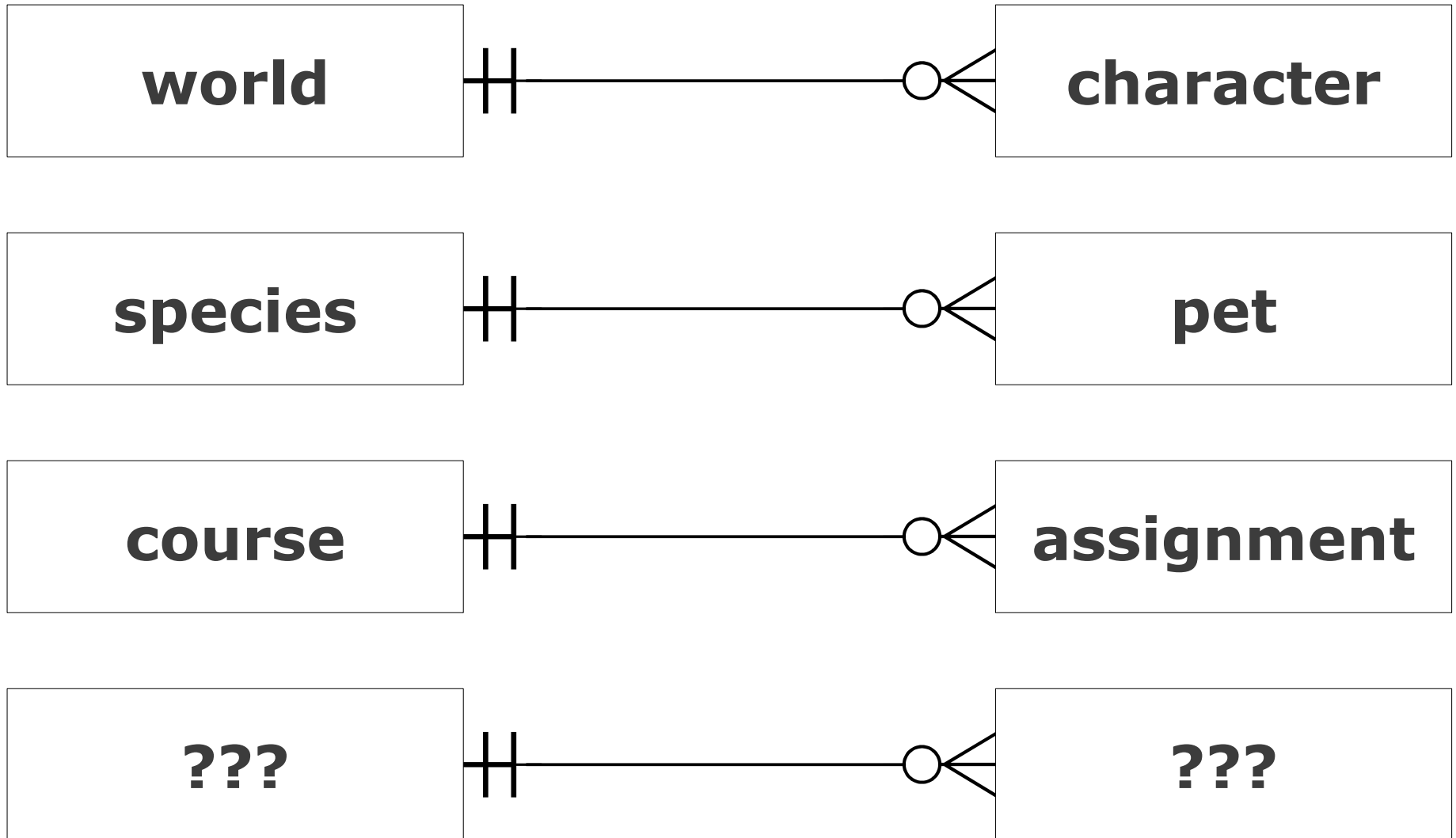
just  
one

zero or  
many

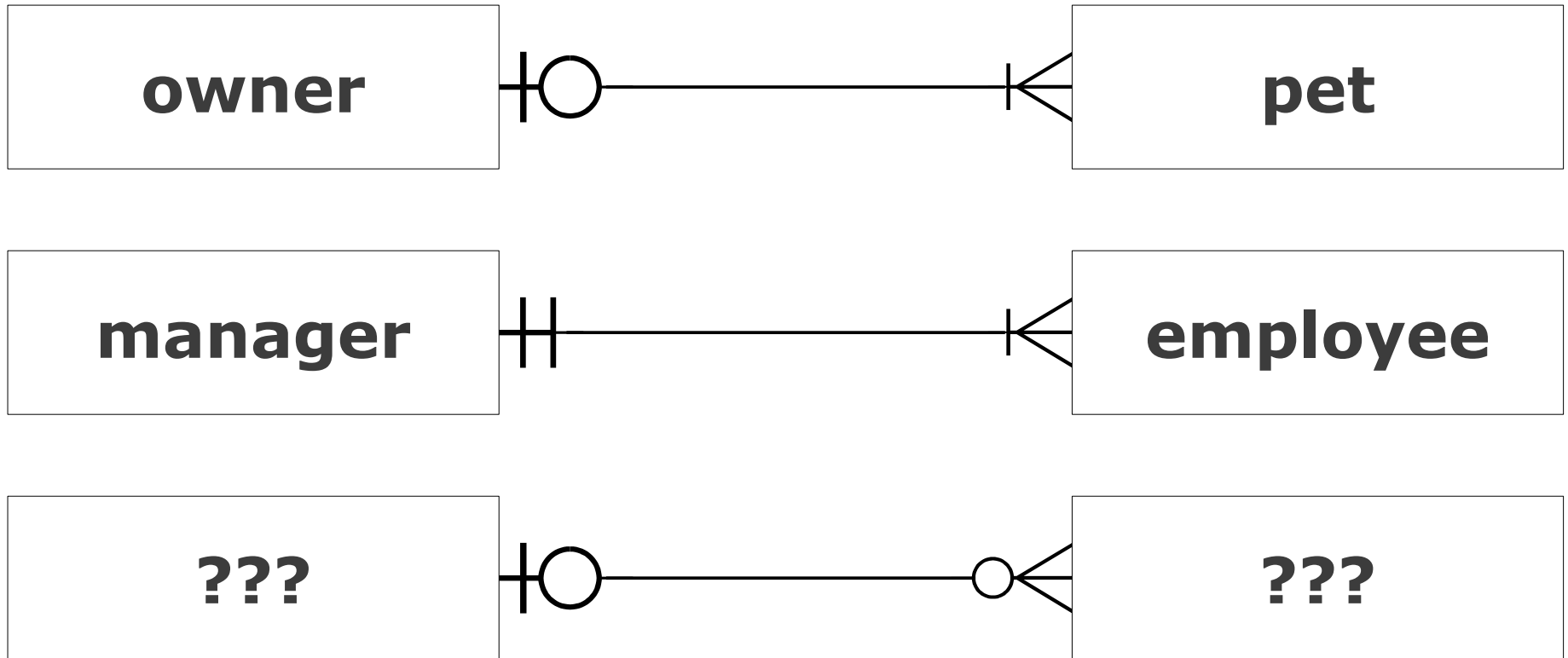
# “Weak Entity”



# More Examples



# Variations



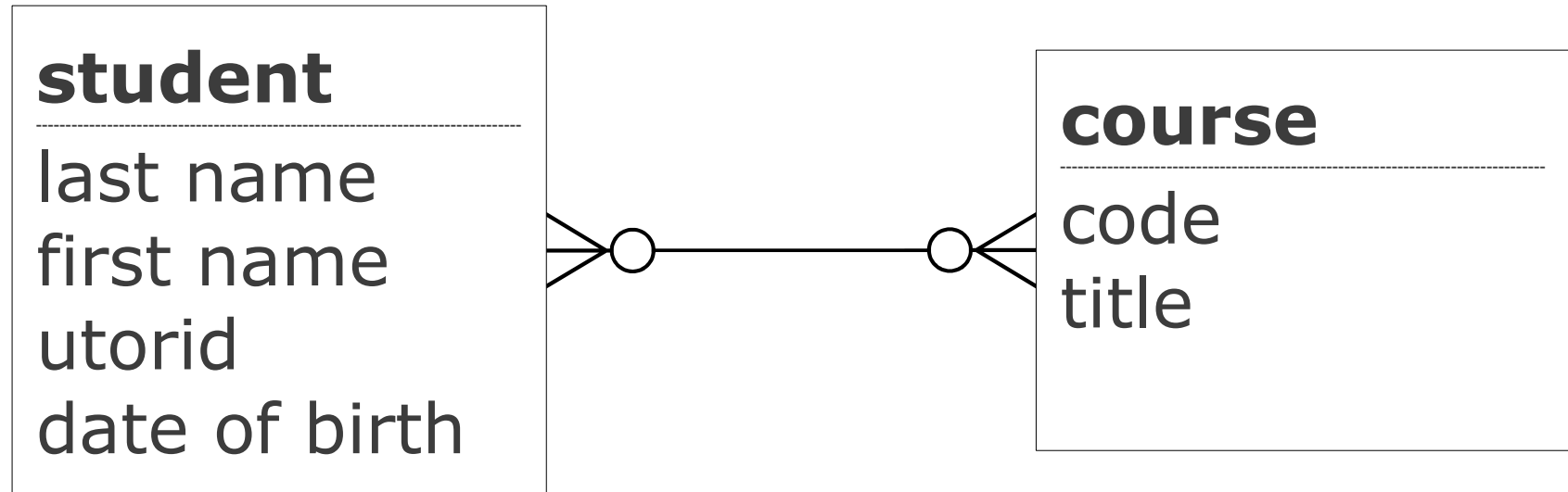
Questions on 1:M?

We 

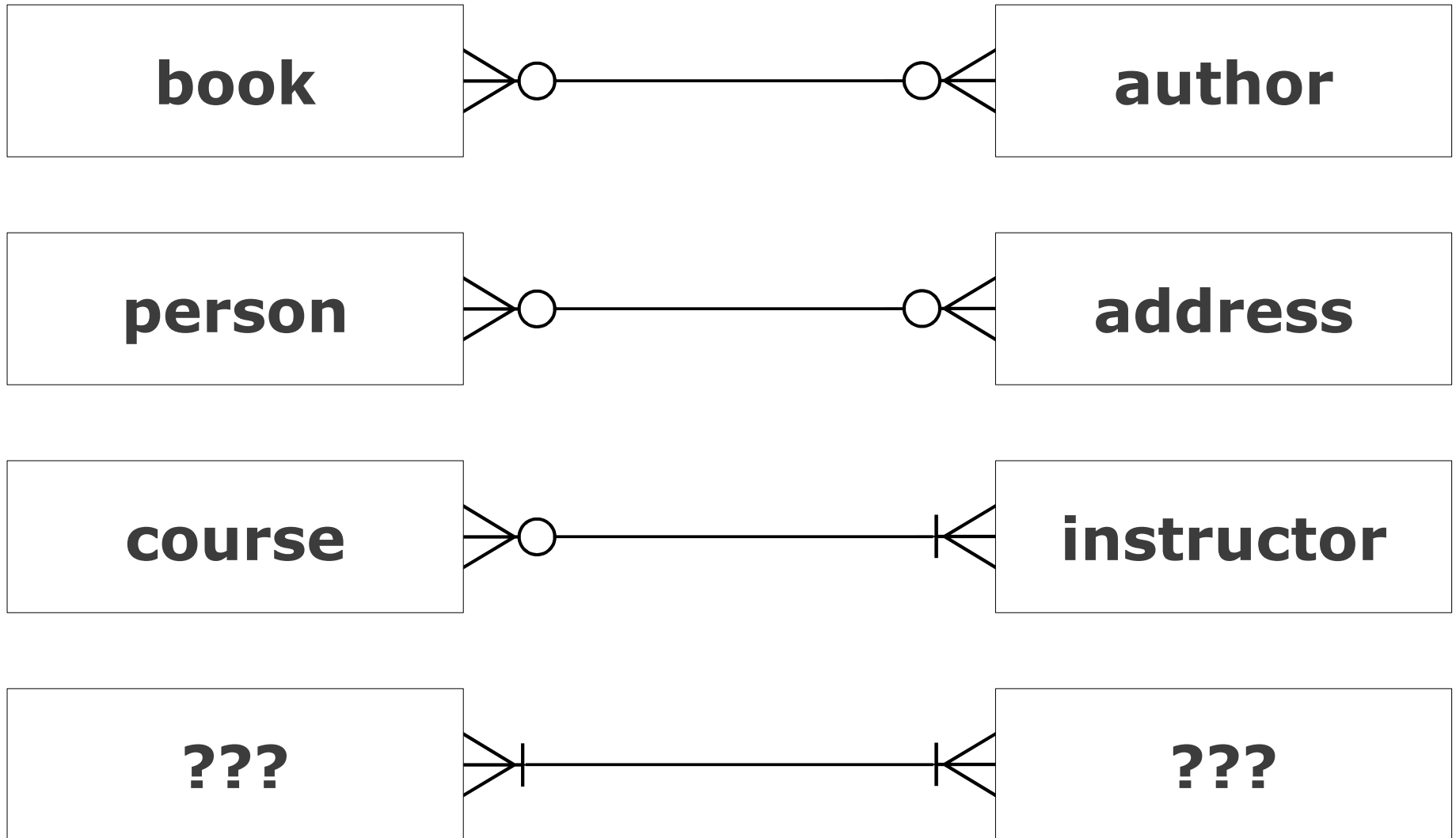
1:M

Because they are really  
easy to represent in a  
relational database.

# Many-to-Many



# Examples





# Movie Database

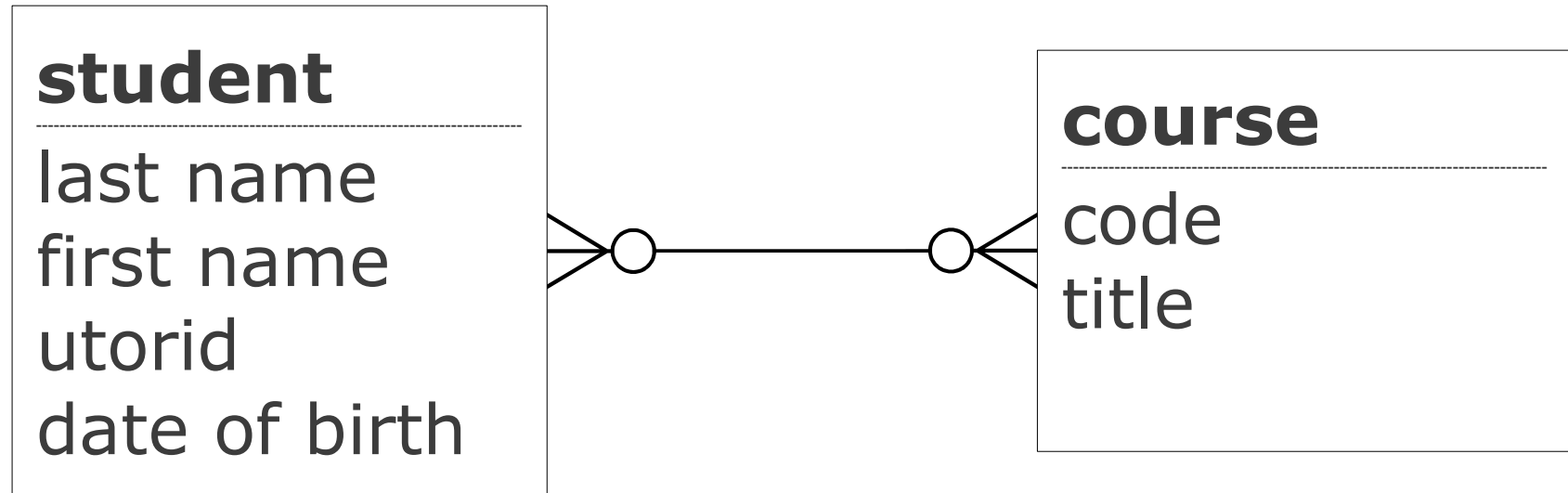
Which relationships are 1:M and  
which are M:M?

We! 

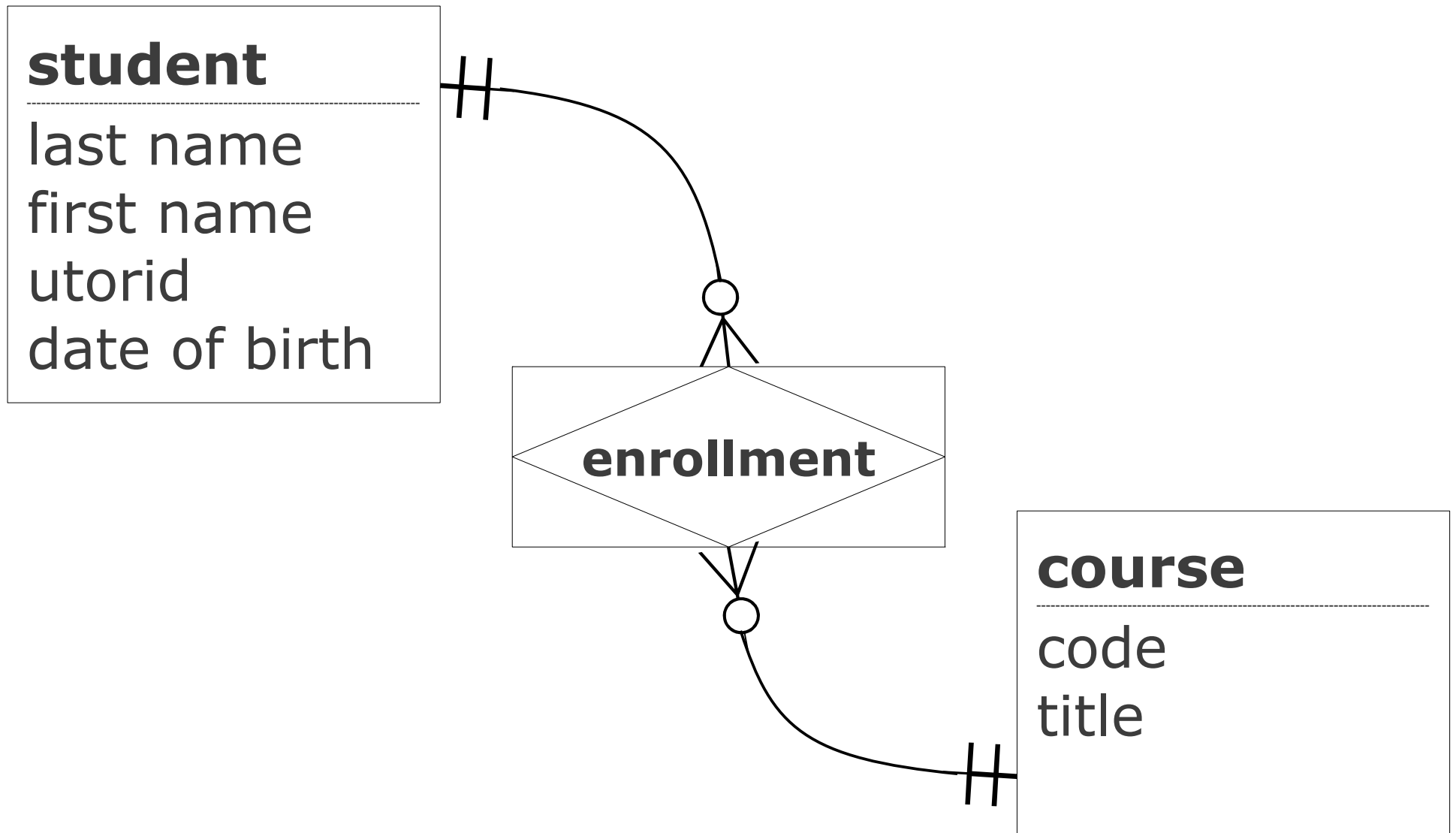
M:M

Because they **cannot**  
be represented in a  
relational database.

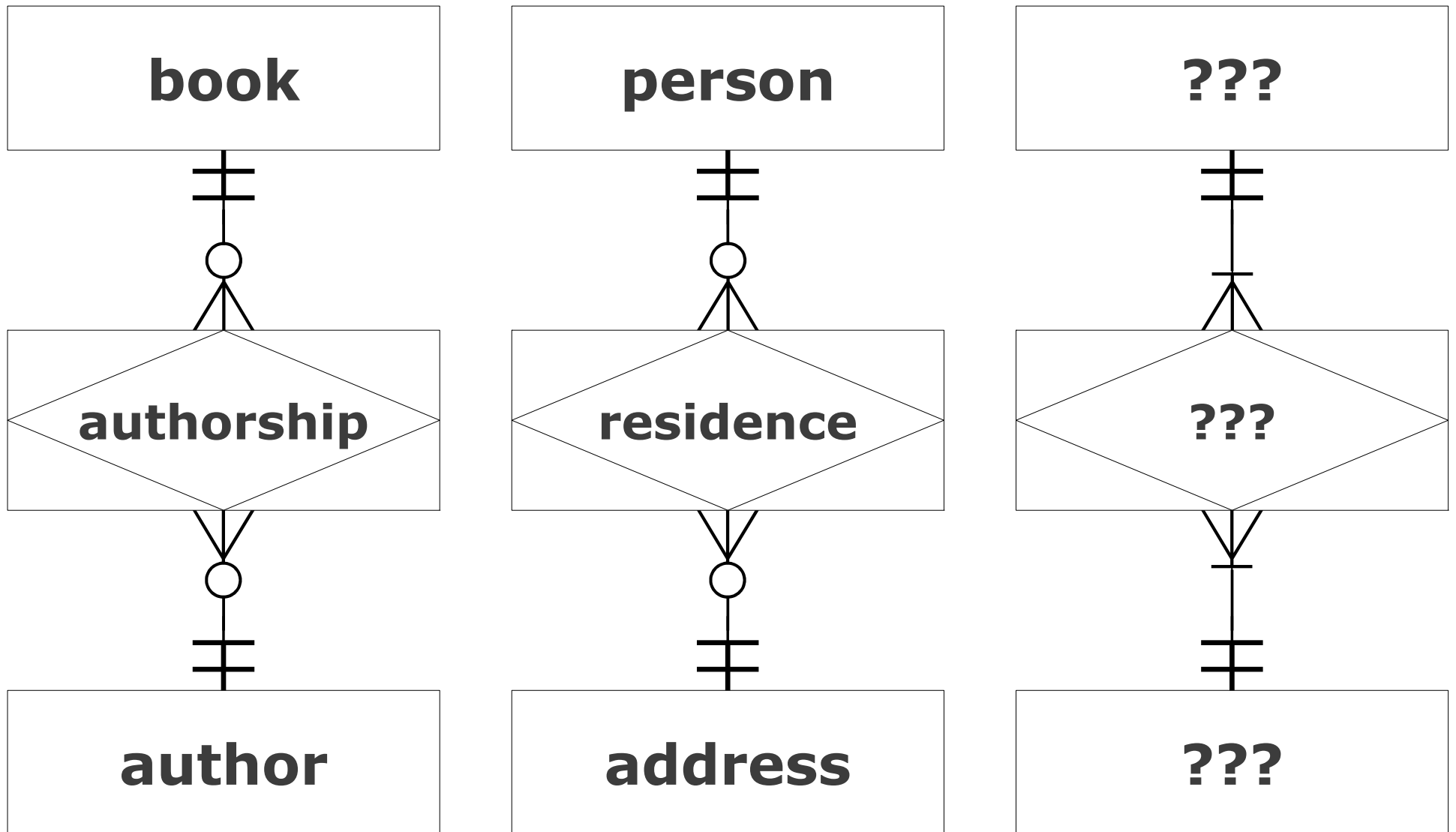
# Breaking Up M:M



# “Associative Entity”

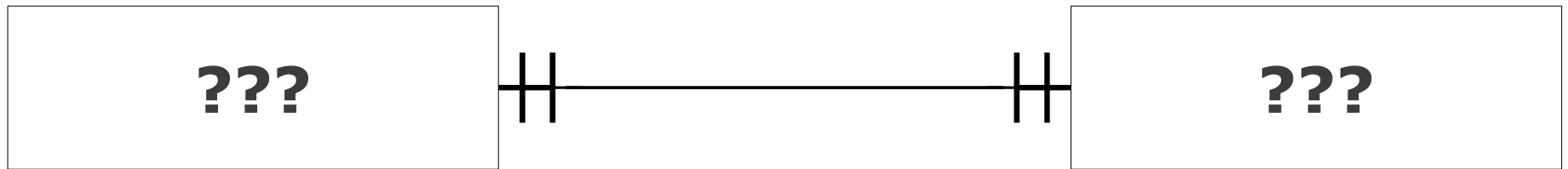


# Examples

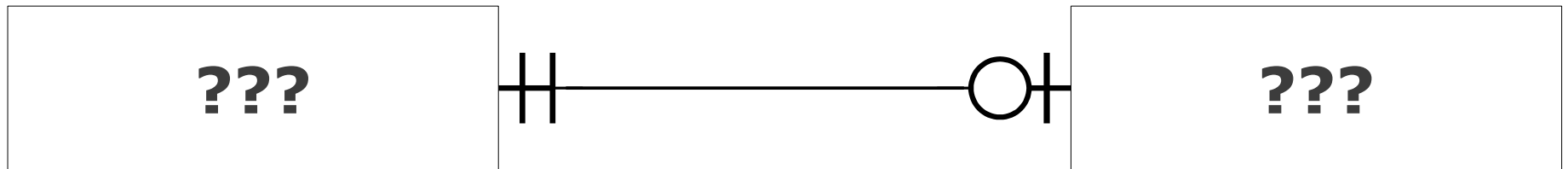


Questions on M:M?

# One-to-One

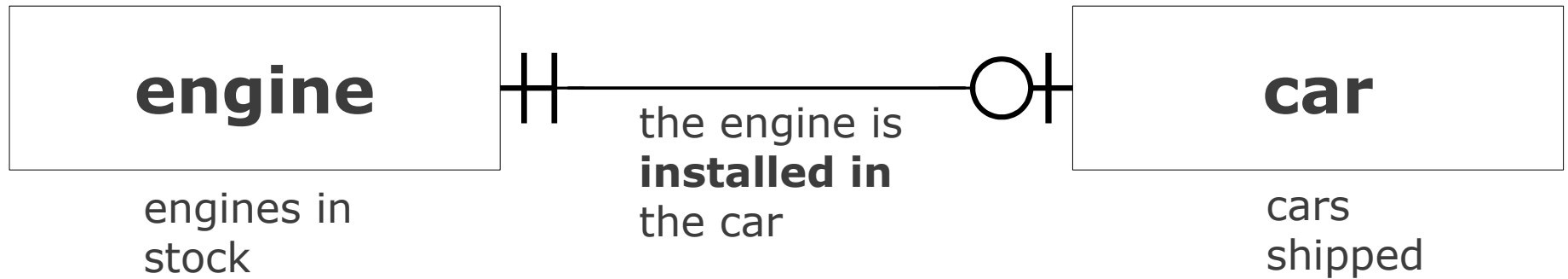


# One-to-One

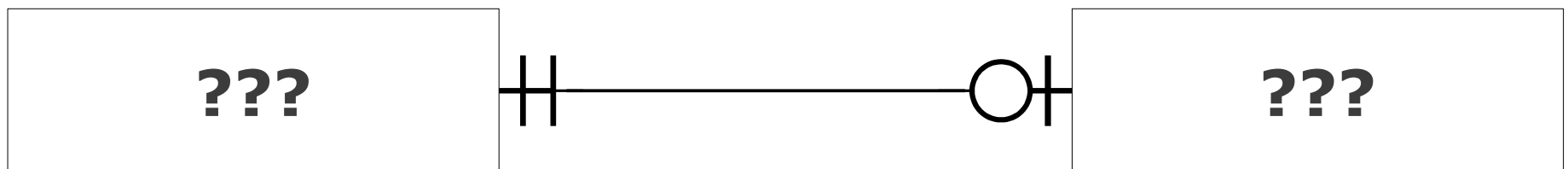
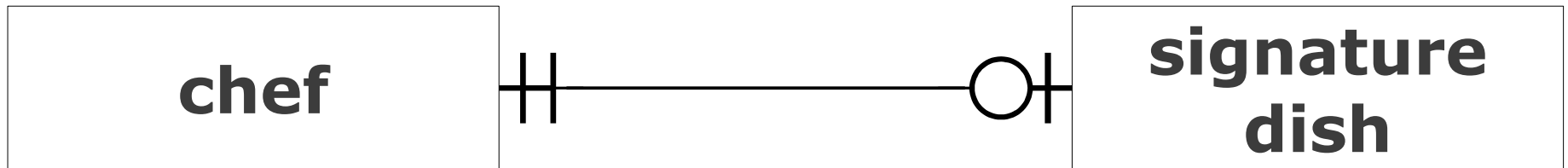
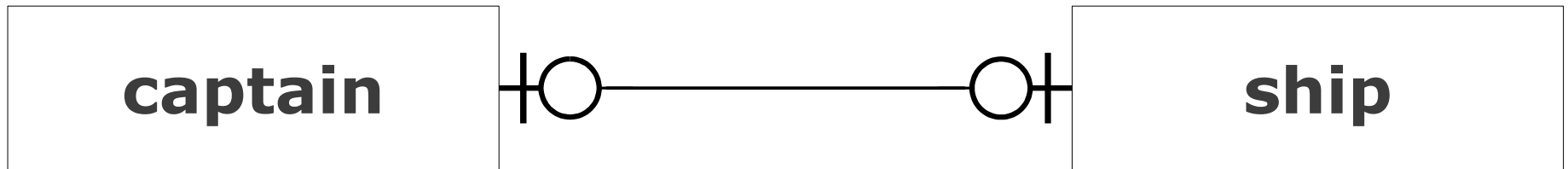
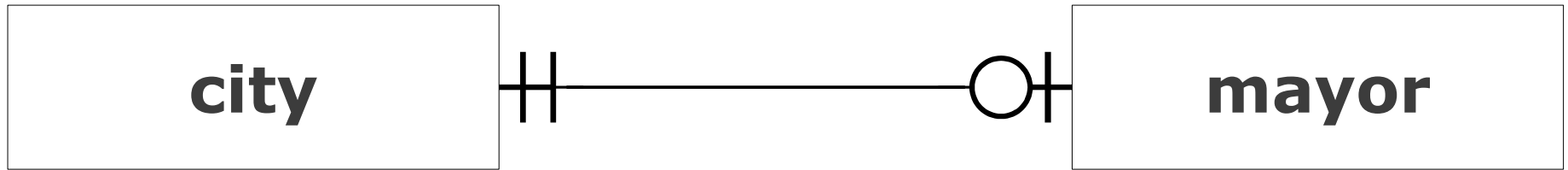




# One-to-One



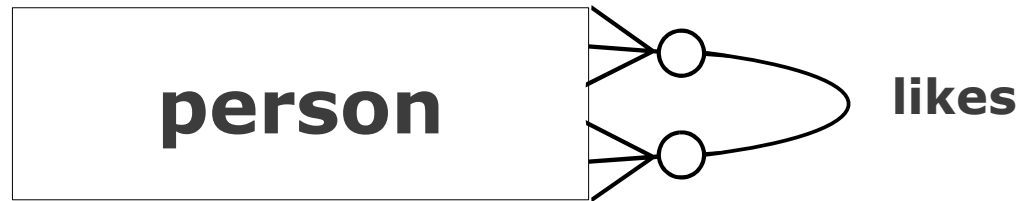
# Examples



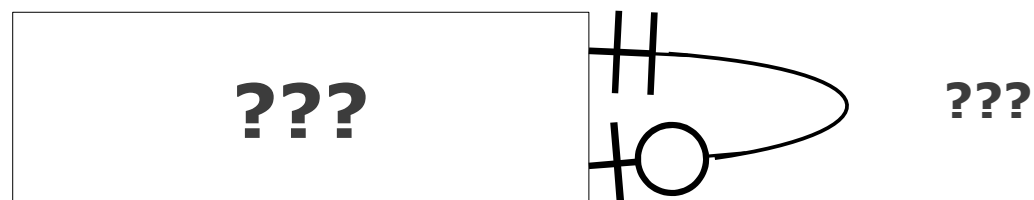
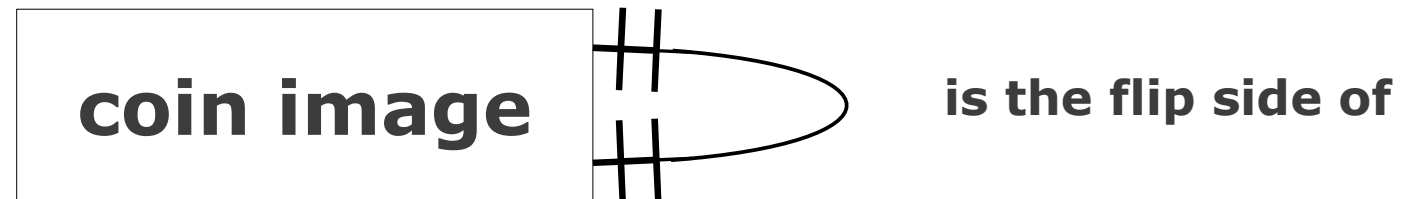
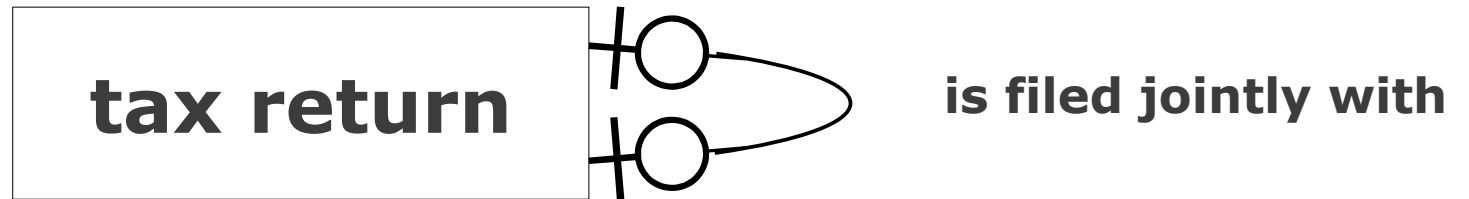
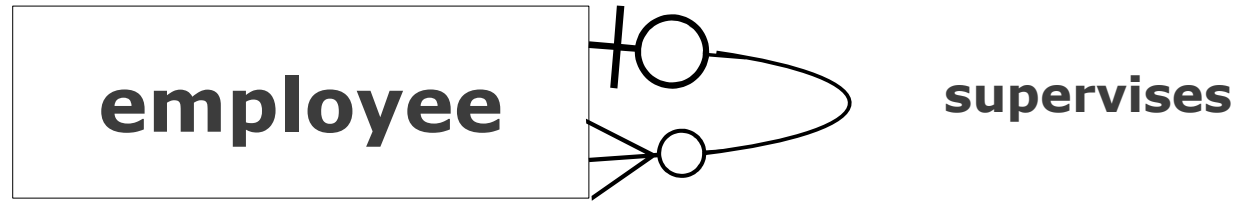
# Recursive Relationships



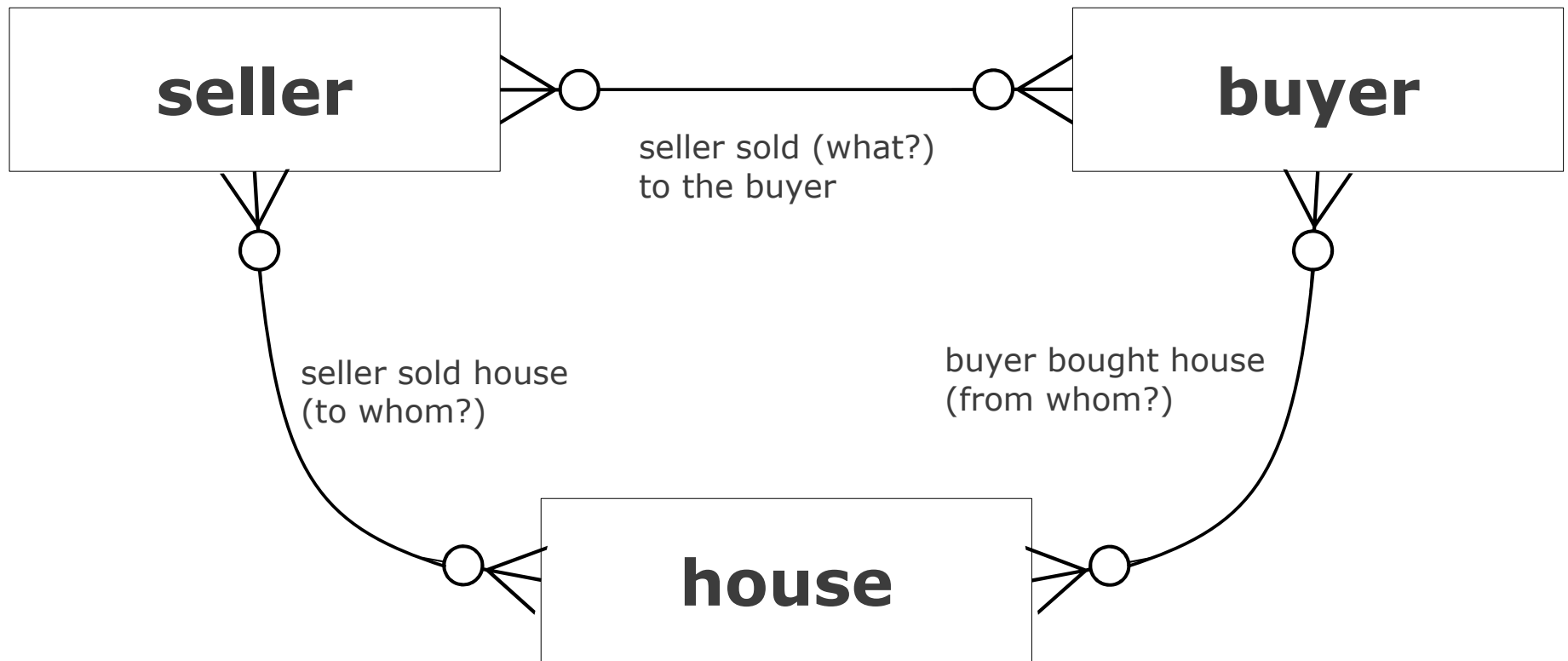
# Recursive Relationships



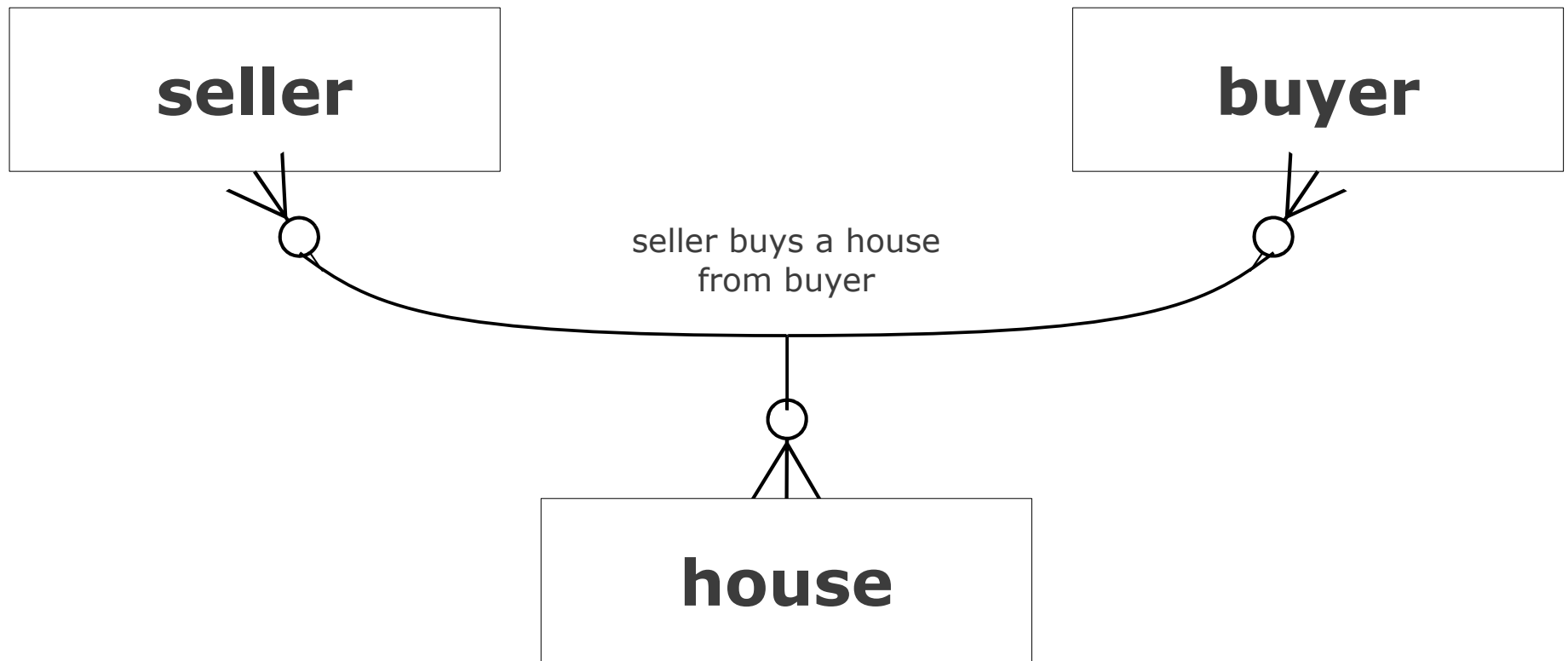
# Examples



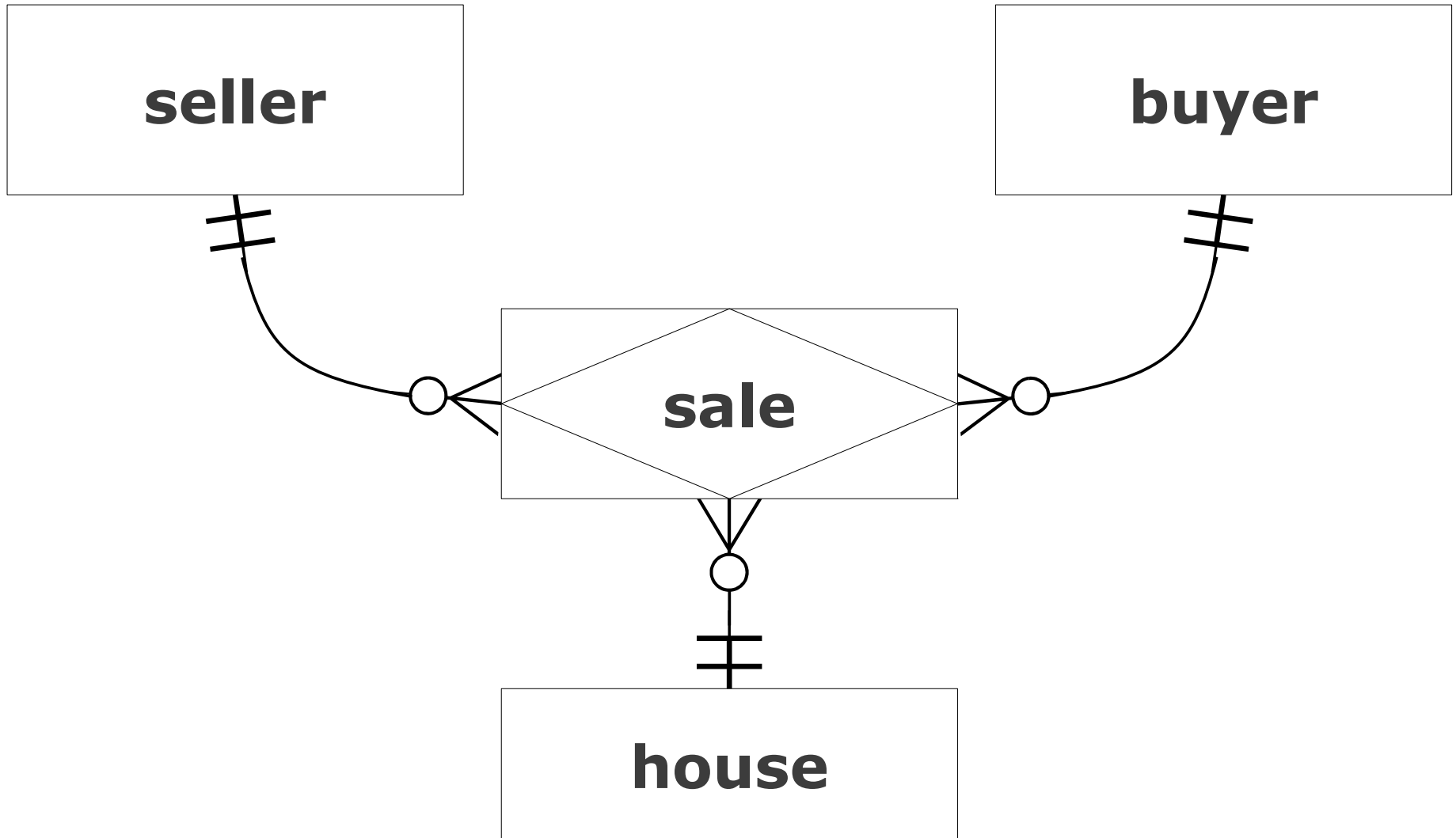
# N-nary Relationships



# N-nary Relationships



# N-nary Relationships



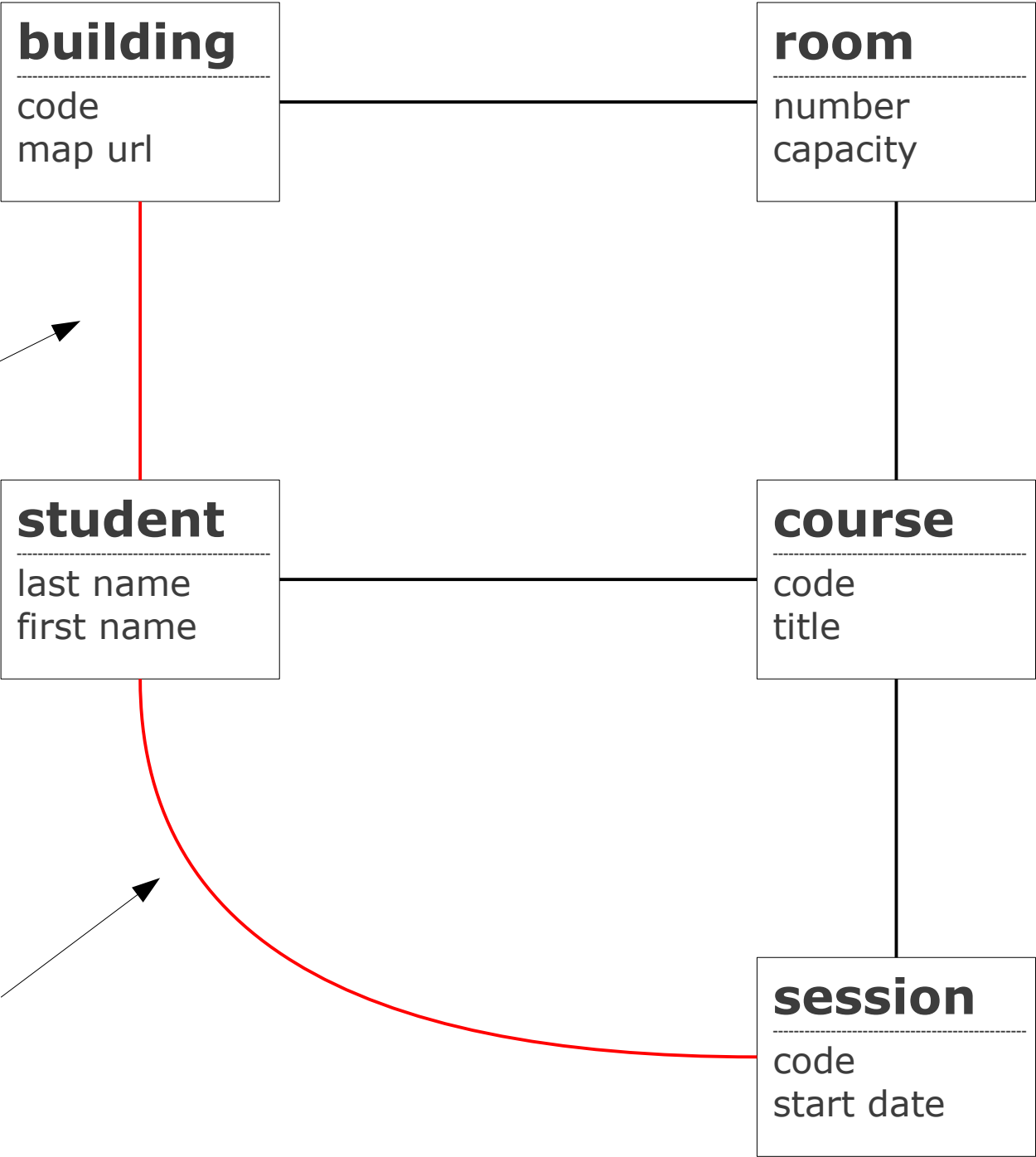


# Putting It Together

## Options for software:

- OpenOffice Draw
  - Free / open source
  - Available in the lab
  - You can get "Crow's Foot" templates at <http://www.thinktek.ca/articles/article2.php>
  - Alternatively, do UML notation ("n..m") by hand
- Microsoft Visio
- Your favorite software

Questions?



**building**

code  
map url

**room**

number  
capacity

**student**

last name  
first name

**course**

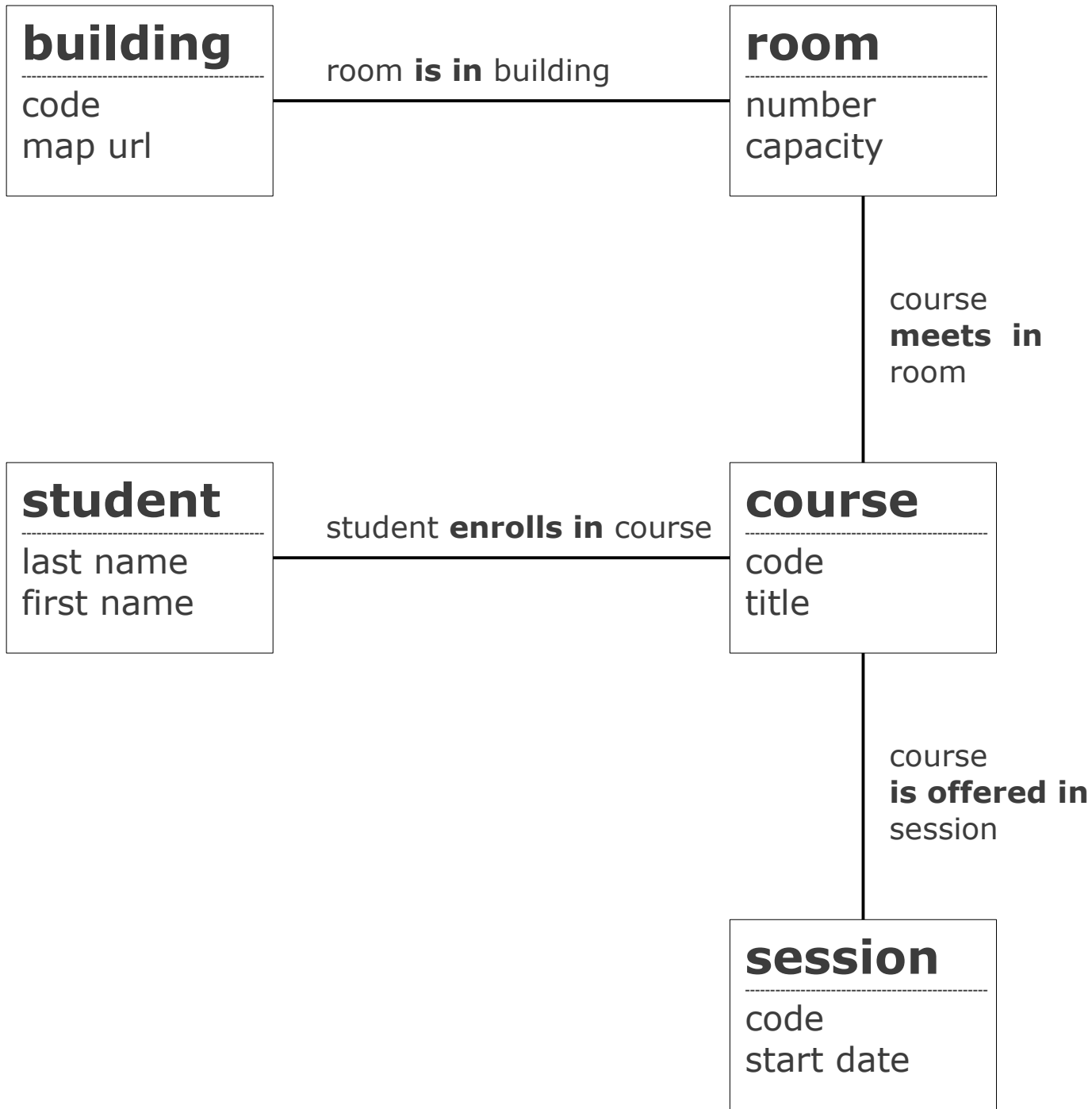
code  
title

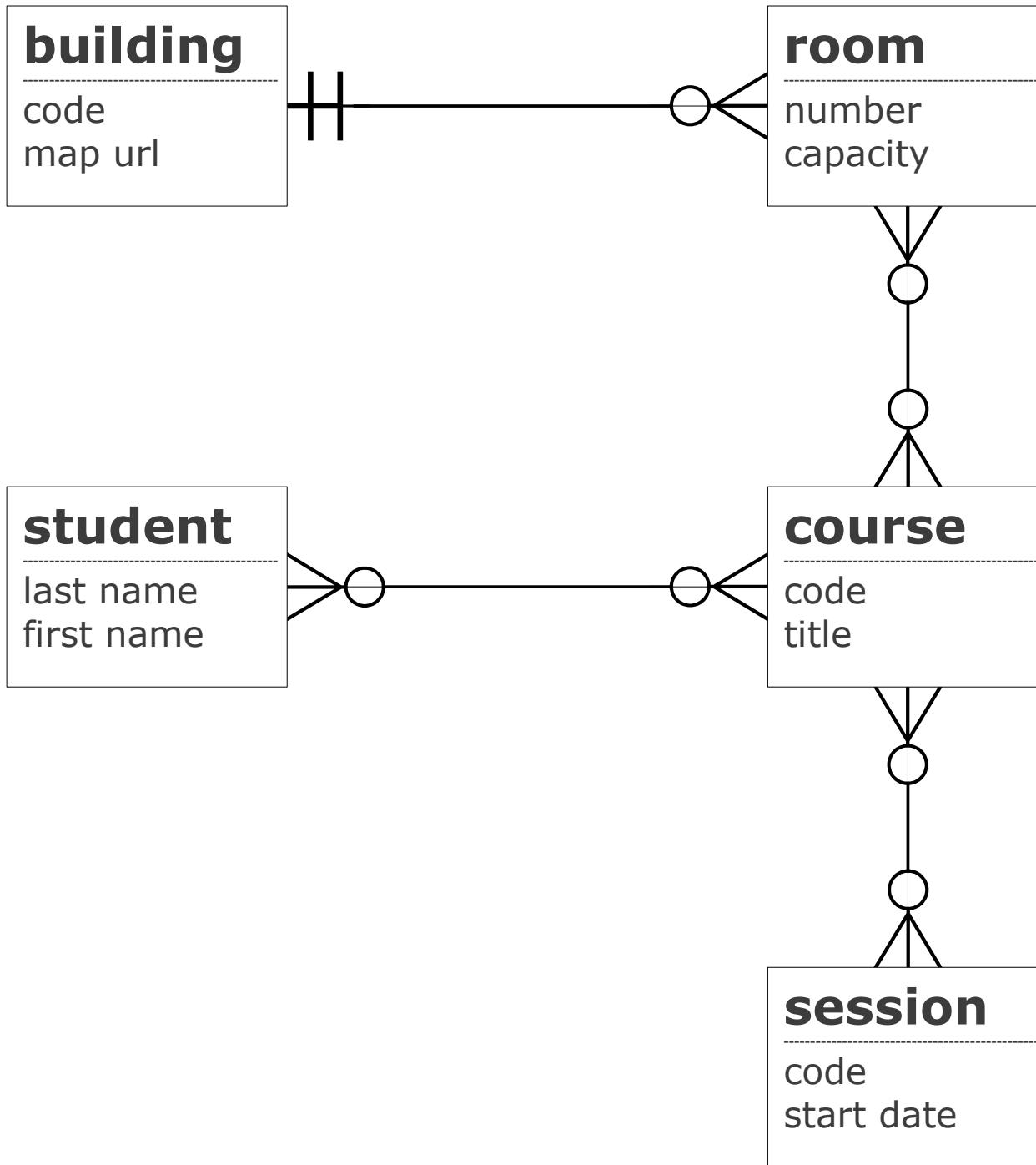
**session**

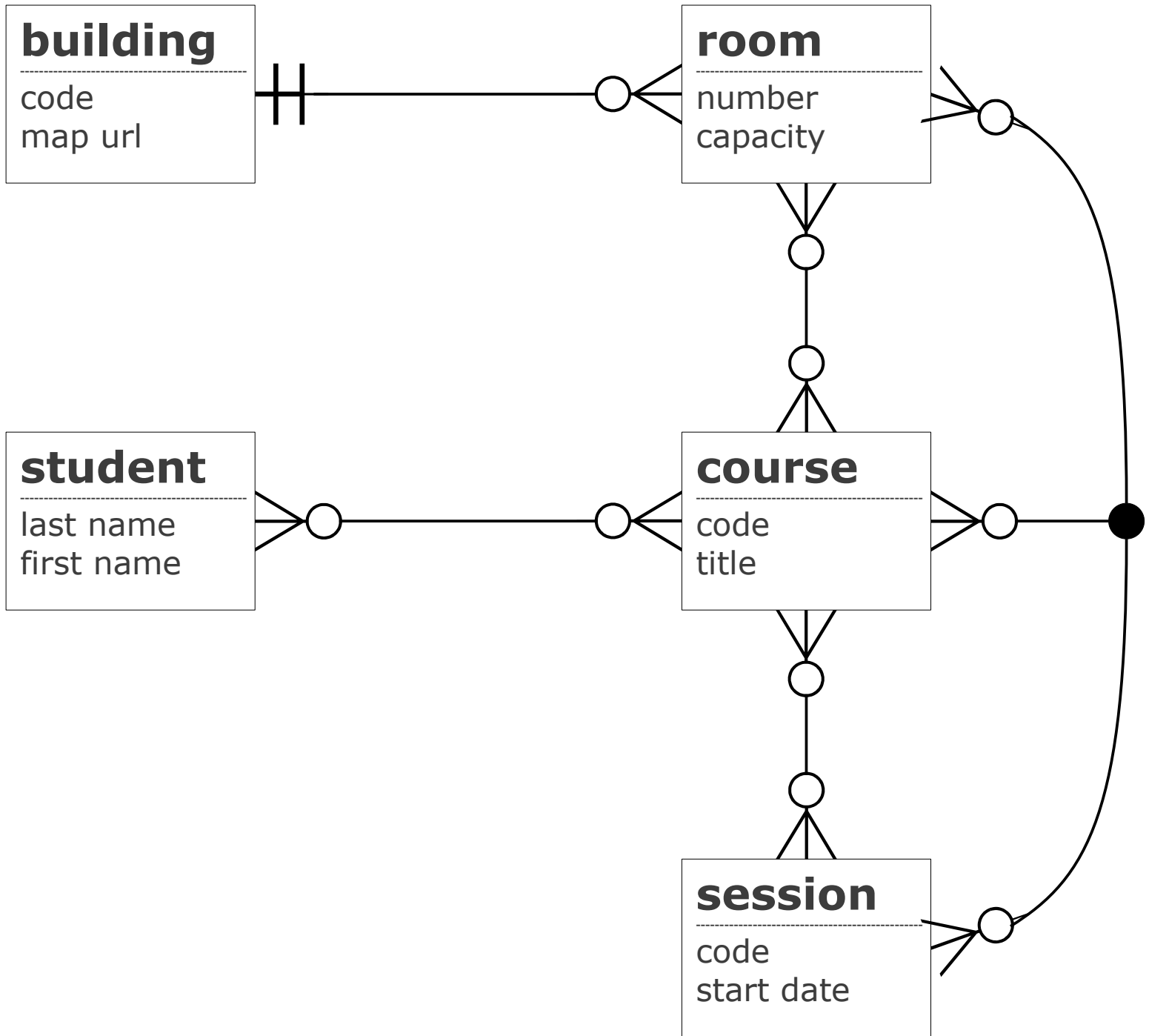
code  
start date

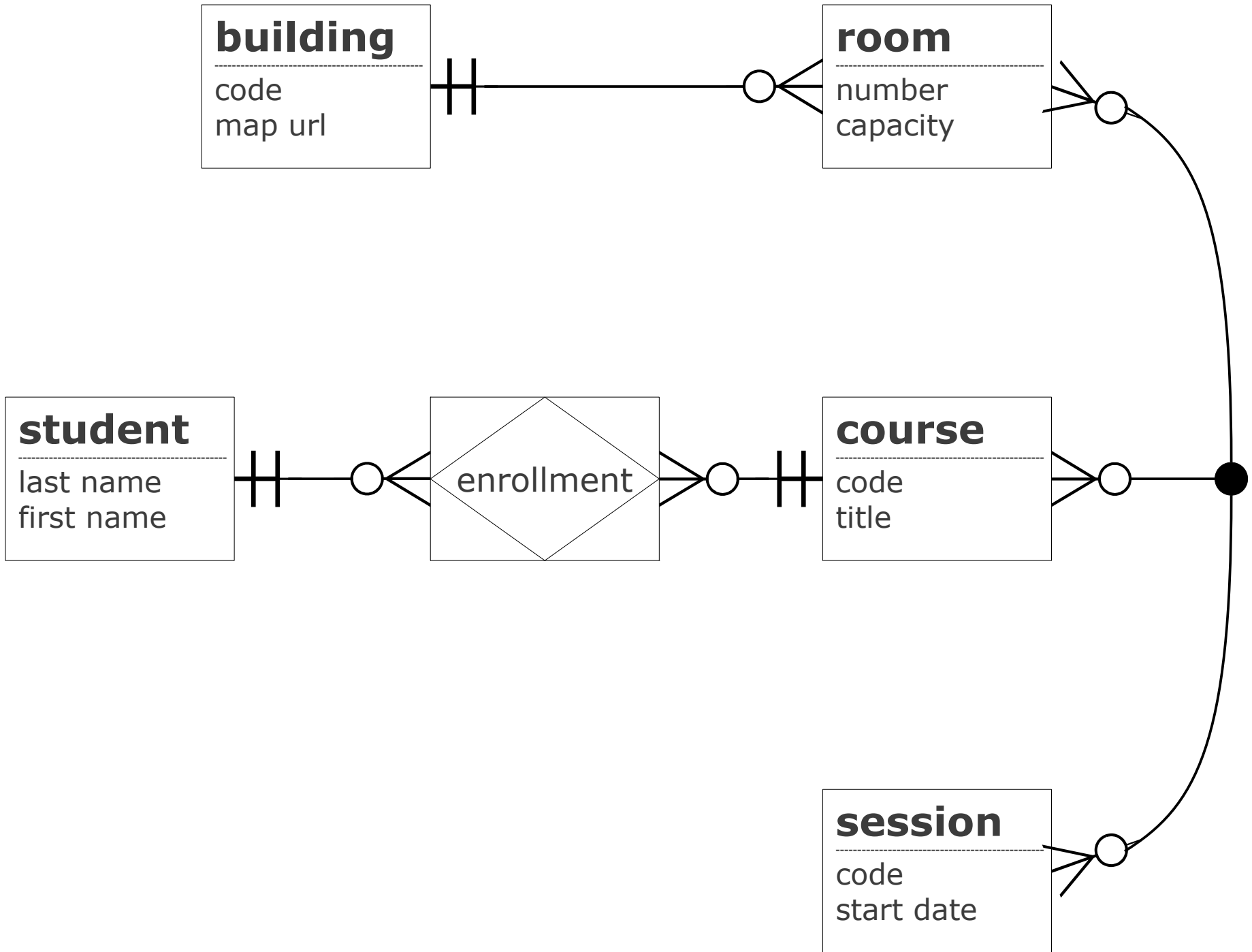
probably  
redundant

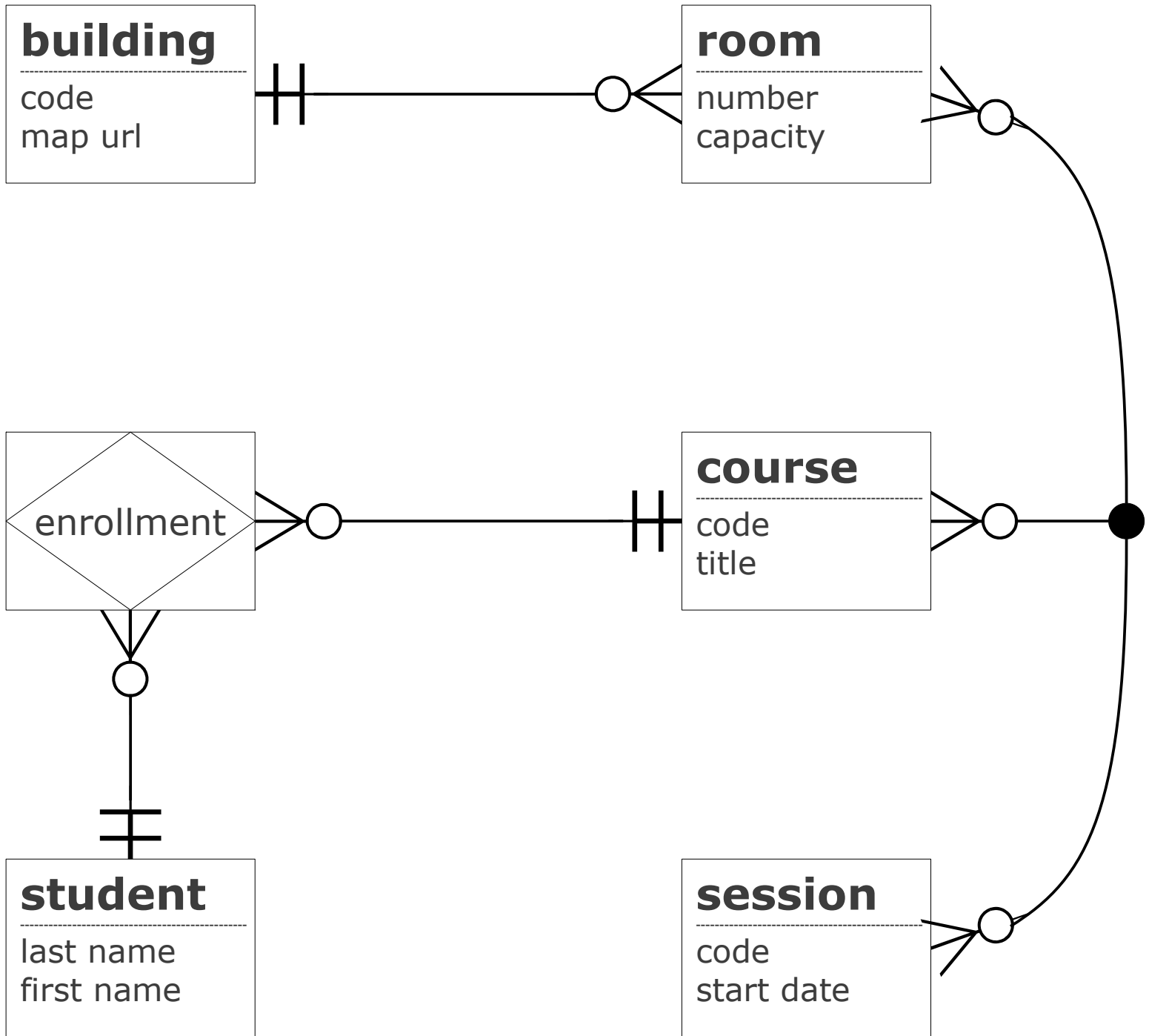
might be  
redundant



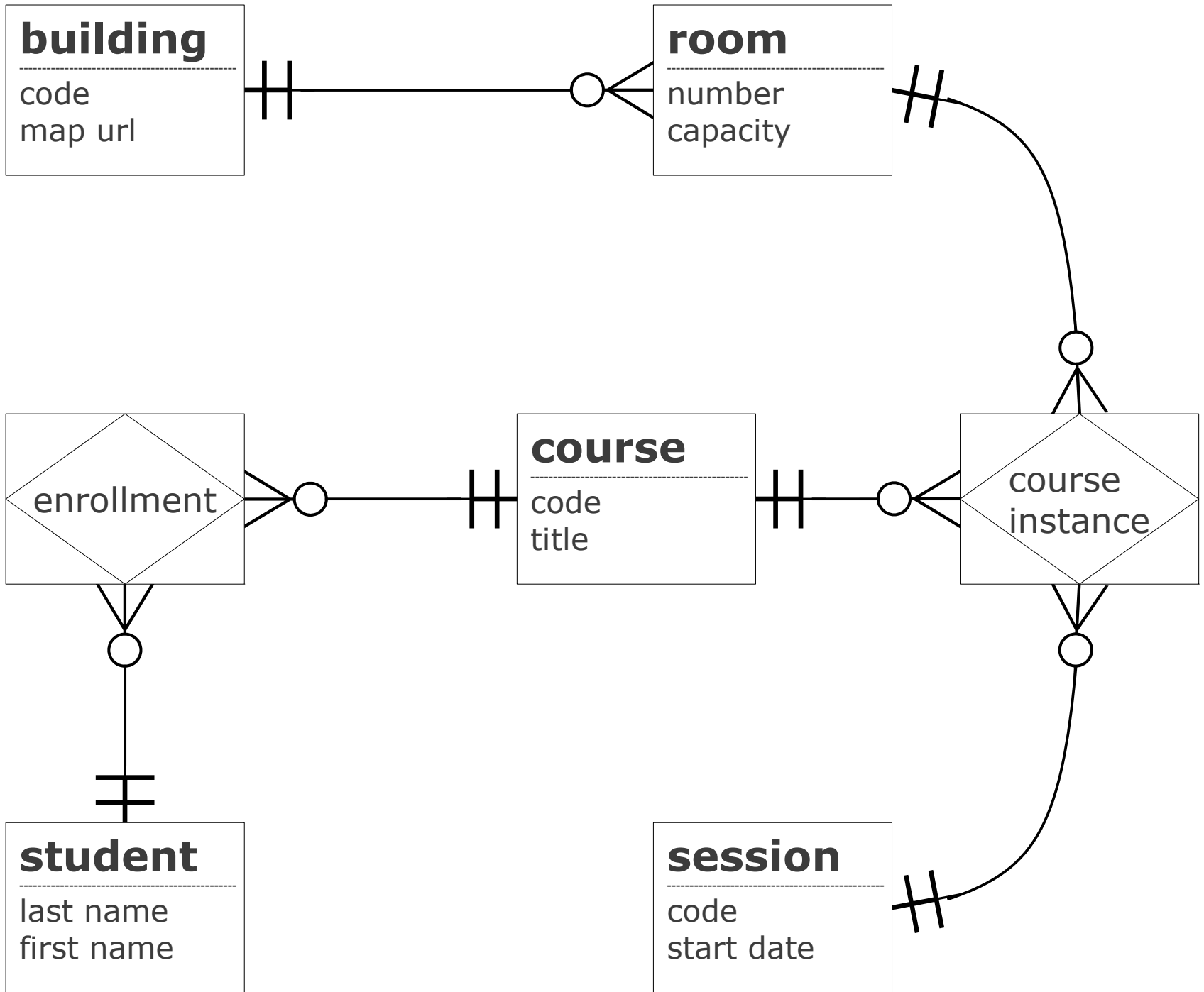


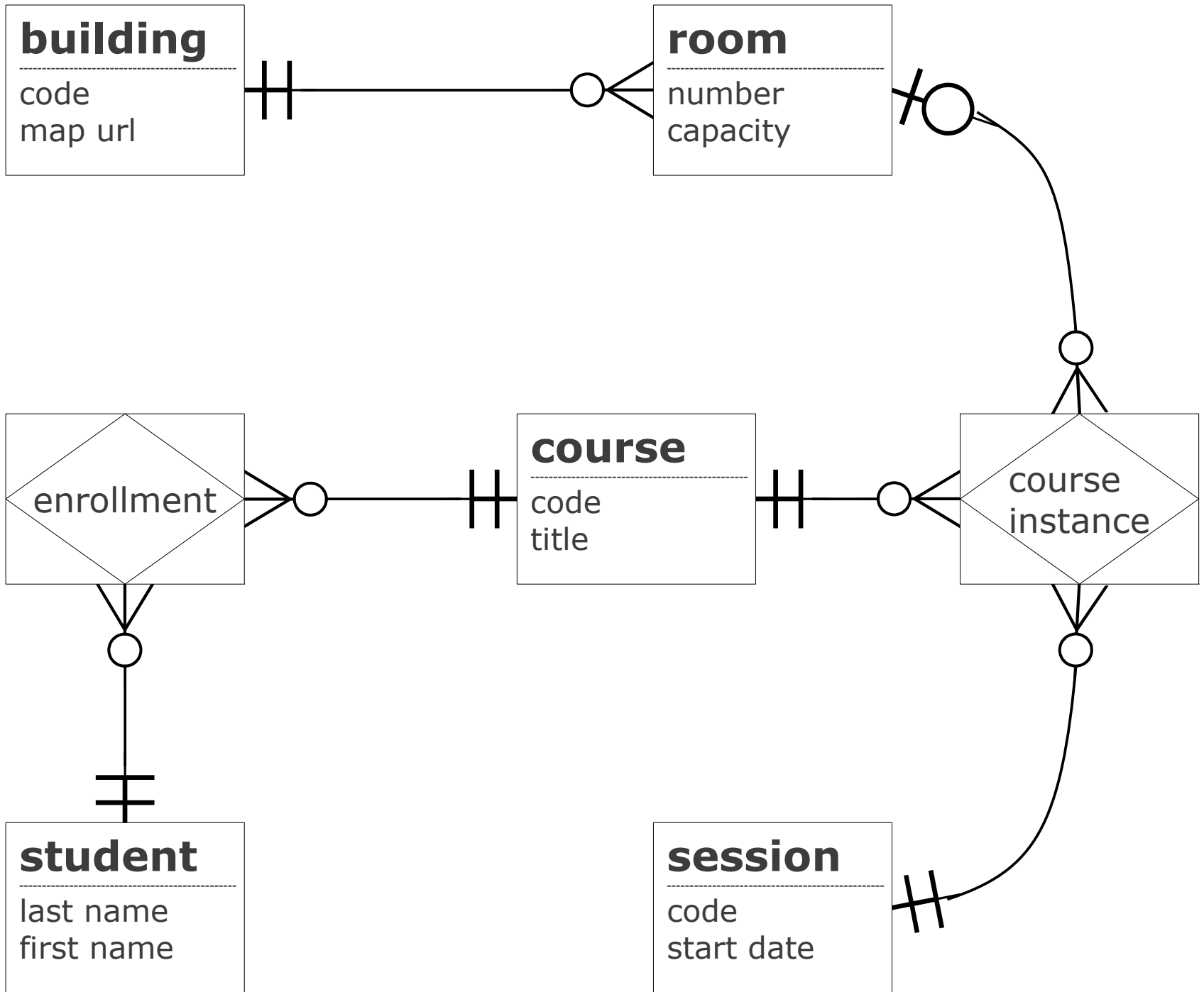


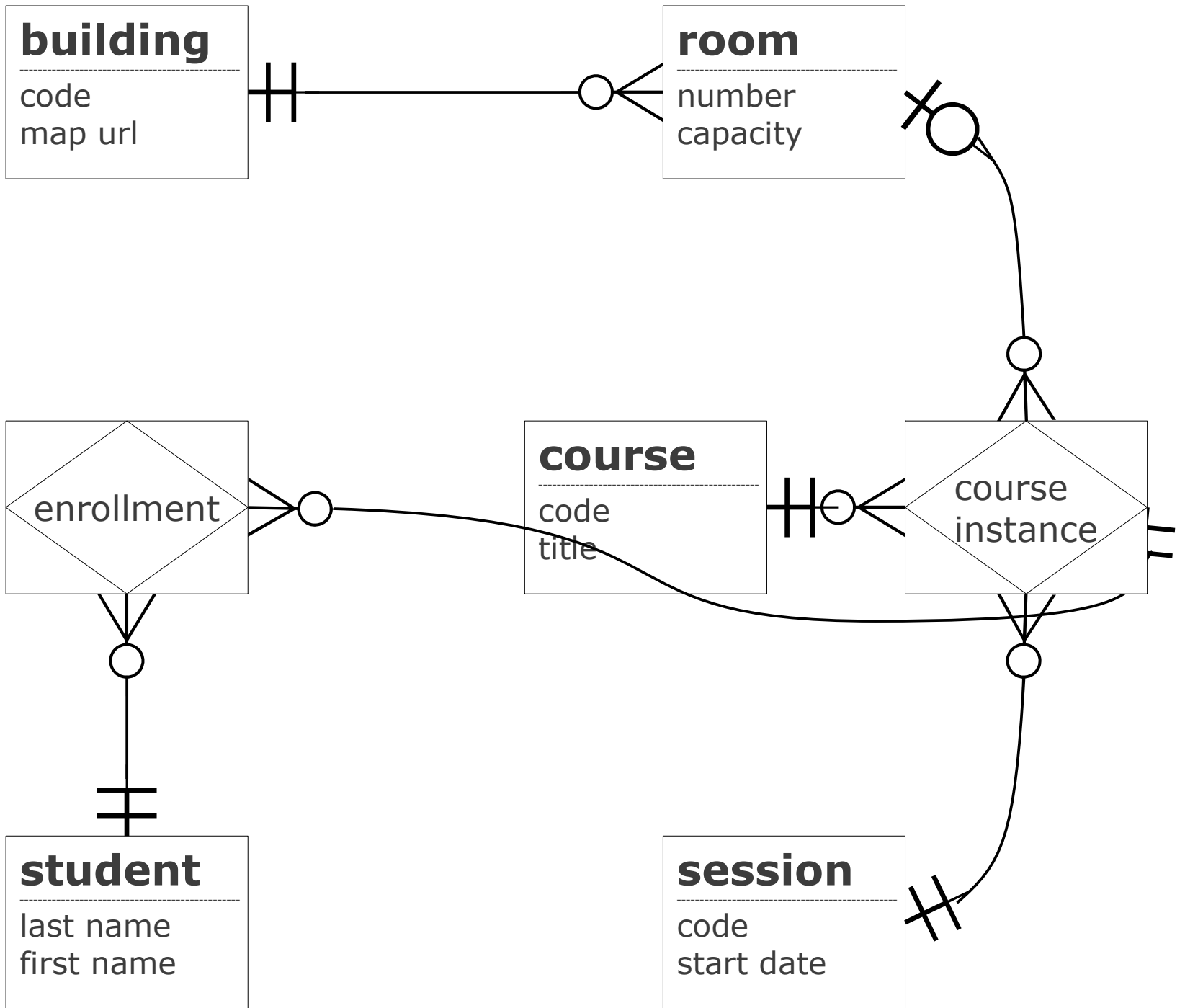


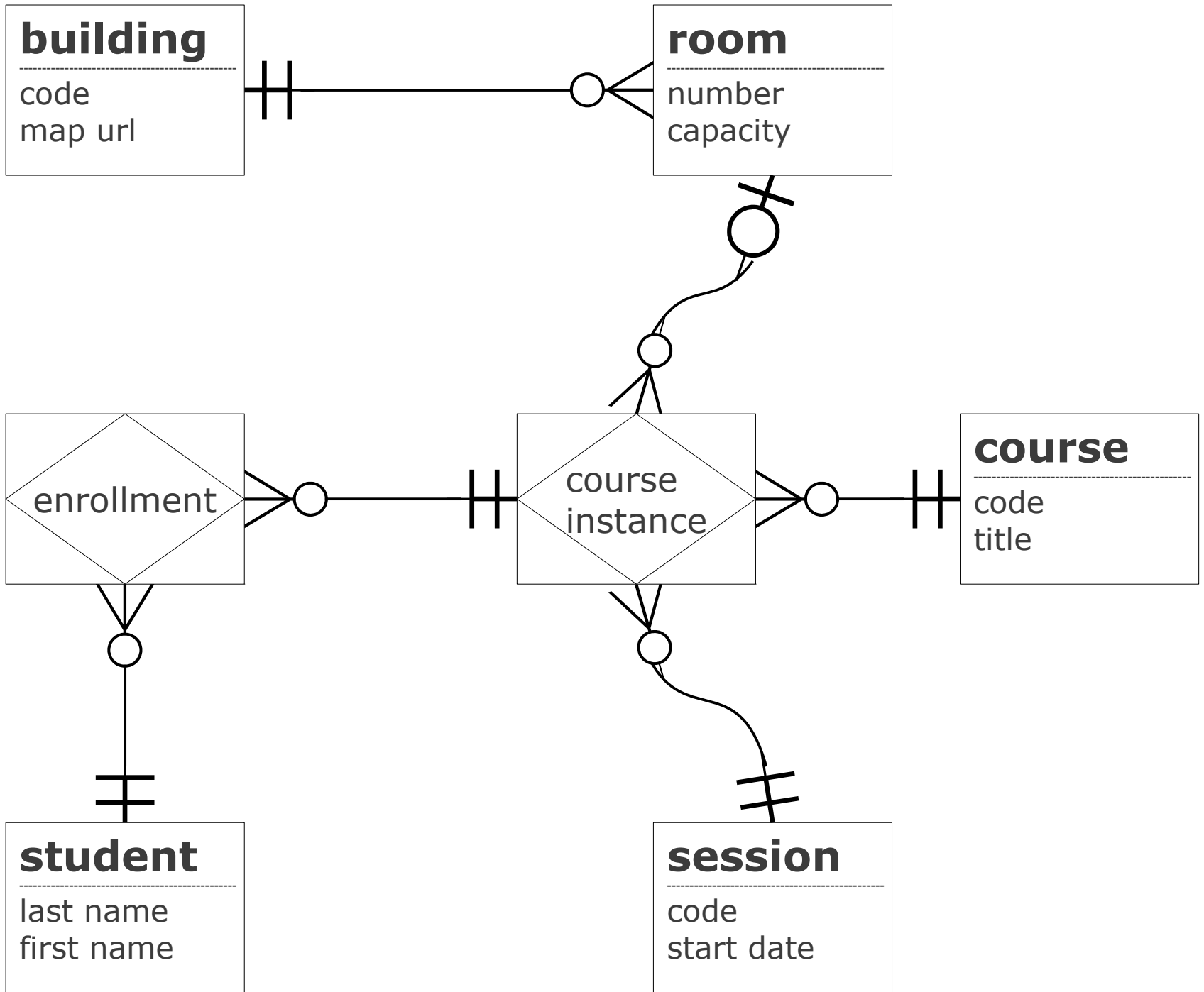


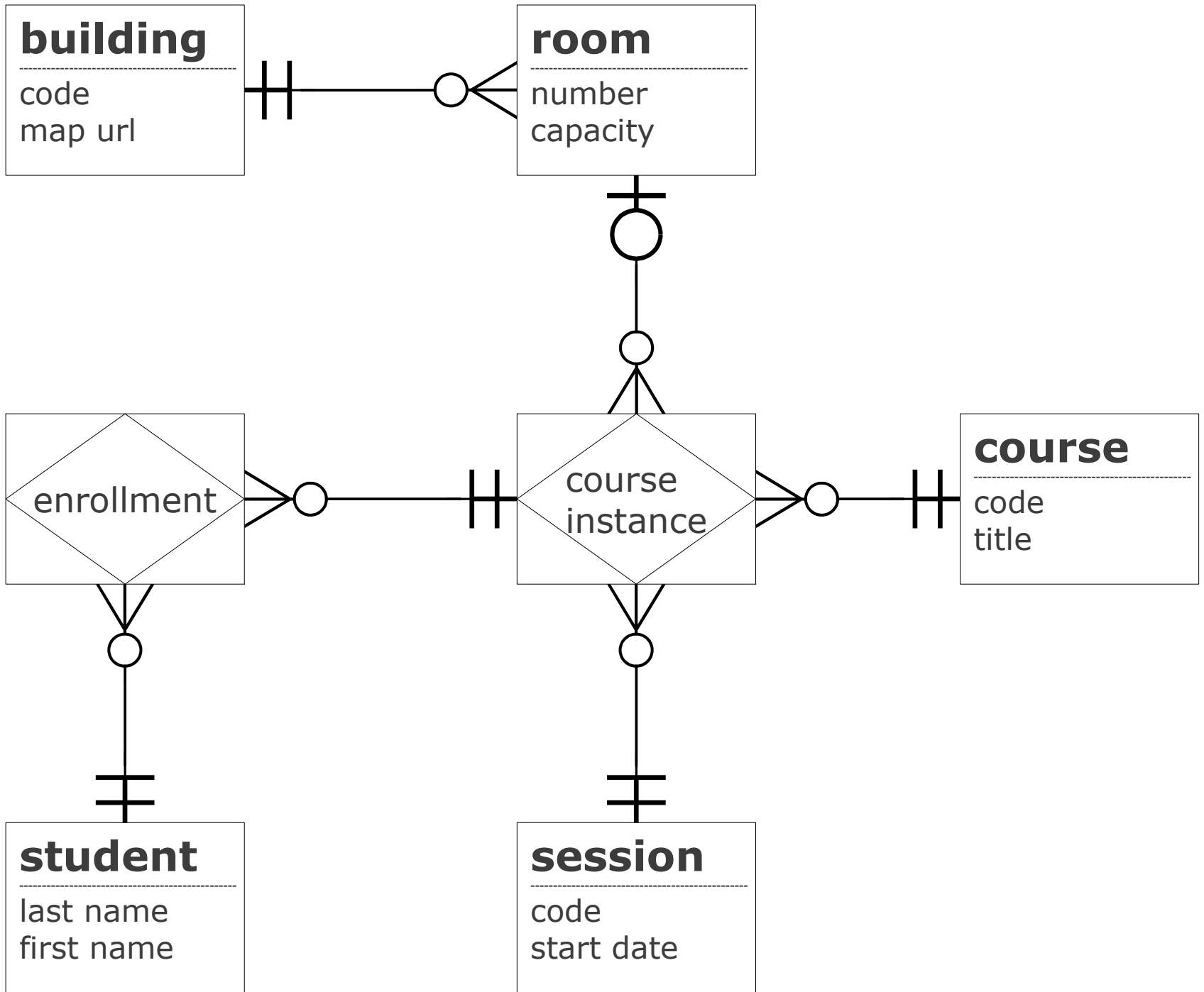


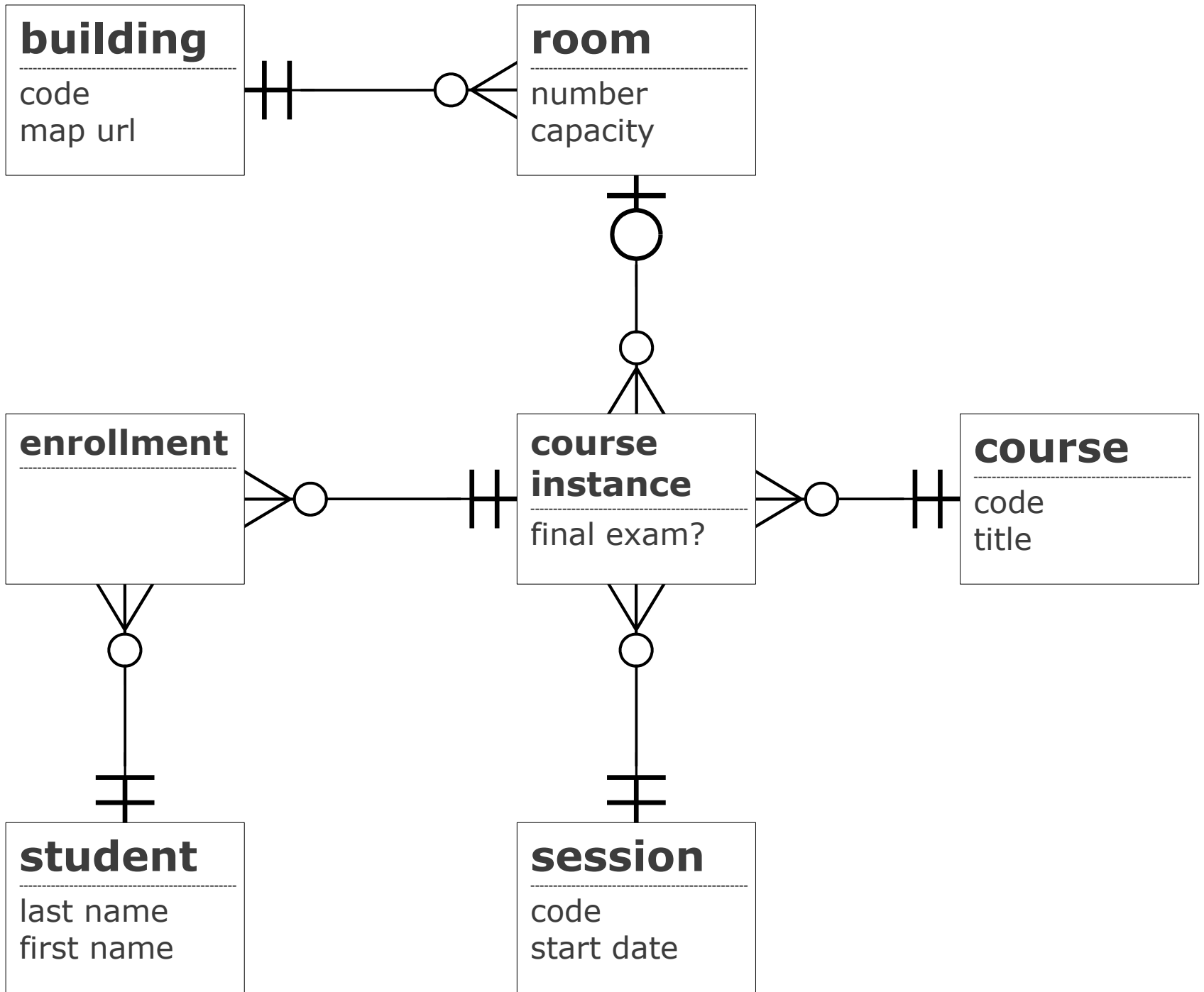


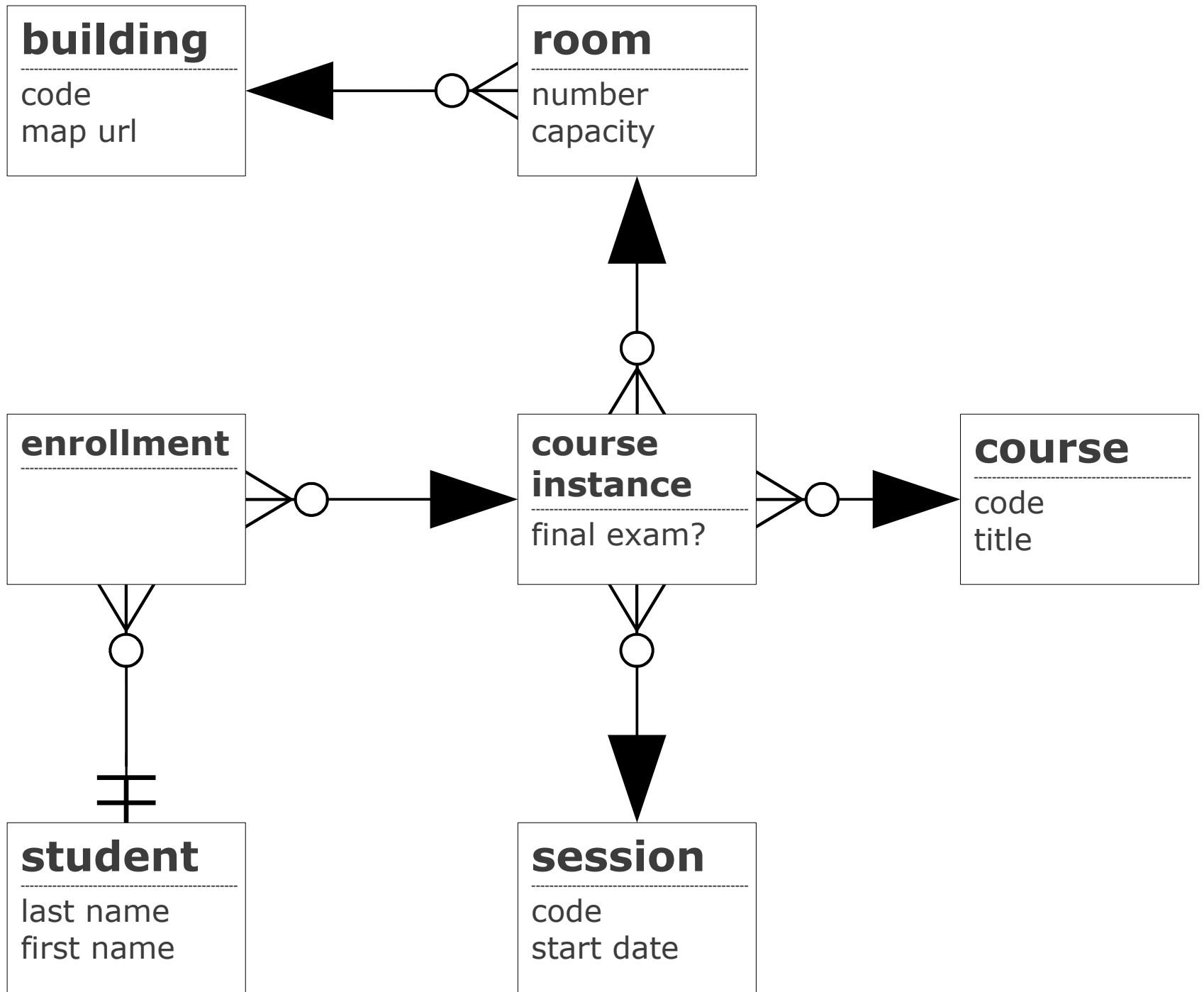












Questions?



