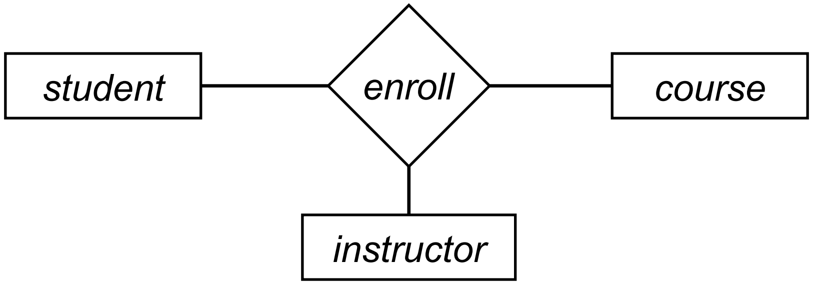
Lecture 4 Entity-Relationship Model Extended Features

Outline

- Aggregation
- Weak Entity Sets
- Generalization and Specialization

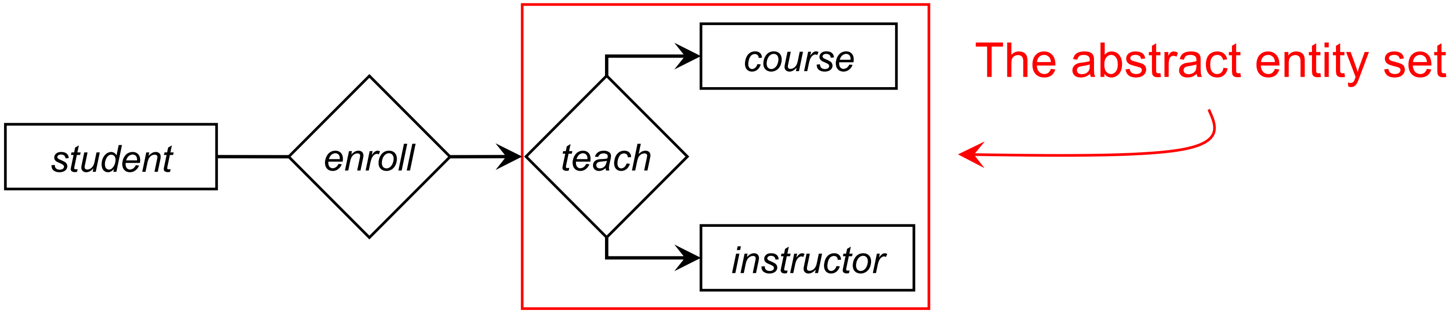
Aggregation

- In the last lecture, we saw the example to constraint ternary relationship sets.
- But there were some constraints which were not discussed.
- Recall the ternary relationship set and the constraints.
  1. One course is instructed by at most one instructor.
  2. One instructor can instruct at most one course.
  3. One student is associated with at most one combination of instructors and courses.



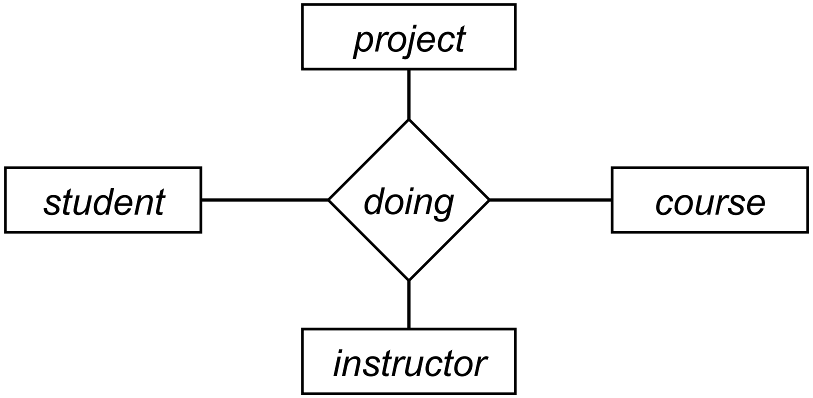
Aggregation

- The difficulty in this example is that the ER diagram wants to express two types of constraints at the same time,
  - a constraint between two entity sets in a multi-ary relationship set, and
  - a constraint between an entity set and a combination of multiple entity sets.
- To handle this issue, ER diagrams use ***aggregations***.
  - Treat relationship as an abstract entity.
  - Allows relationships between relationships.
  - Abstraction of relationship into new entity.
- Back to the example, the following design is much better.



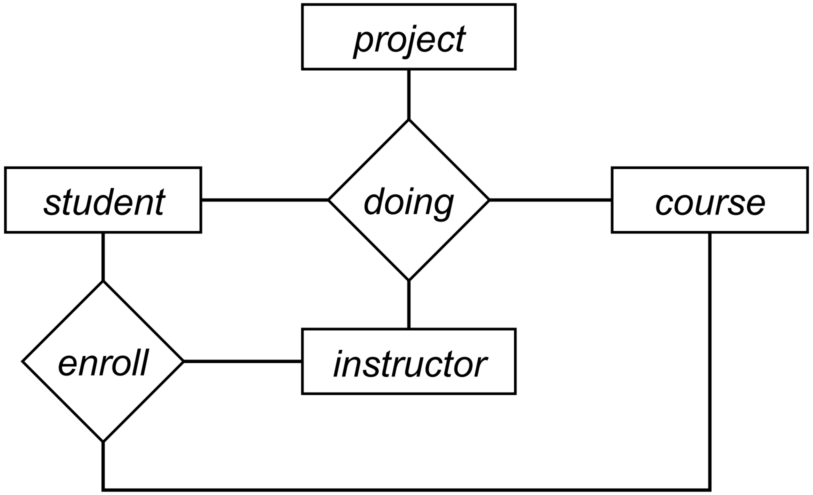
Aggregation

- Sometimes, aggregations are also used to remove redundancies.
- First, let’s go back to the ternary relationship among “student”, “instructor”, and “course” without any constraint.
- This time, we want to model a new entity set, course projects.
- Since each course project is supervised by an instructor and done by some students for a particular course, then a quaternary (4-ary) relationship set is reasonable.



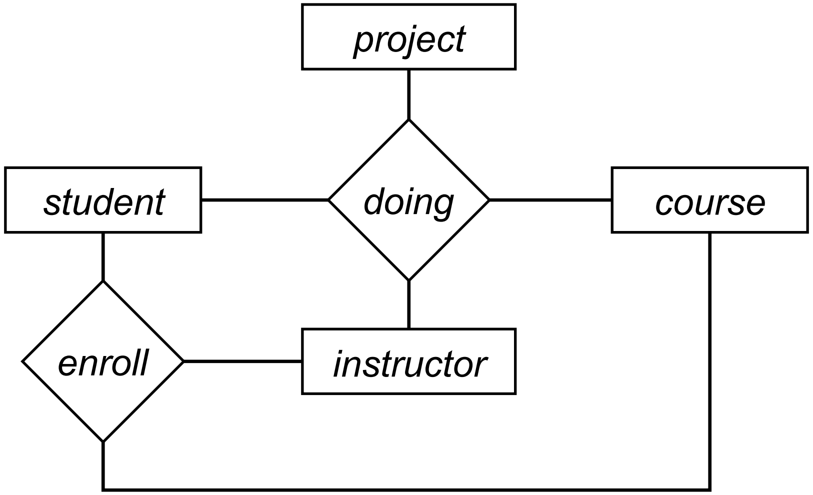
Aggregation

- However, only one quaternary relationship cannot cover the original relationship set “enroll”.
- The two relationship sets are of different meanings.
- More importantly, some courses may not have any project.
- Thus, the ER diagram also needs “enroll”.



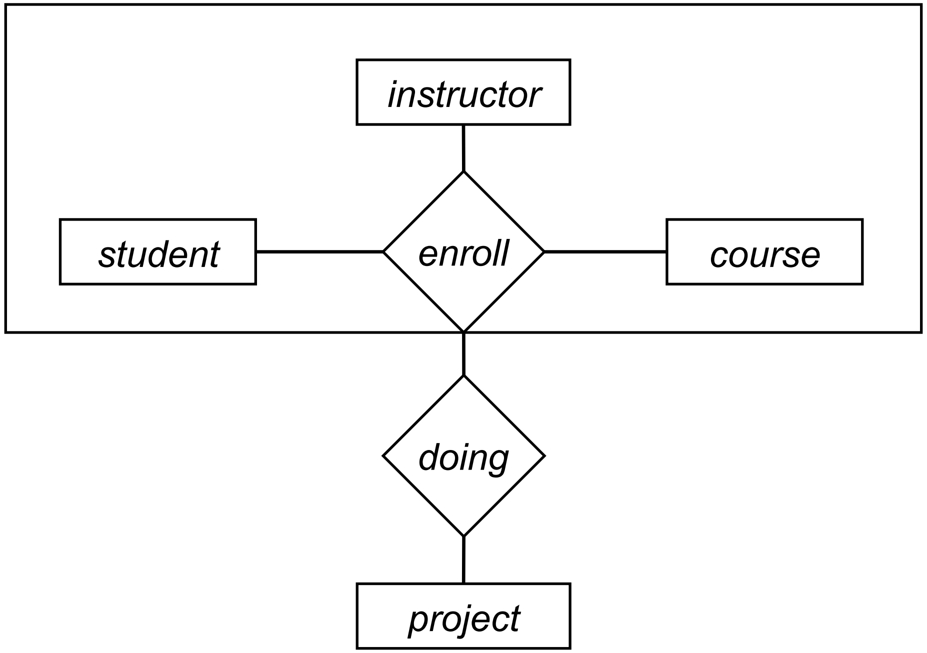
Aggregation

- But this model contains redundancies.
- If a student does a project for a course under an instructor’s supervision, this student and the instructor must enroll the course.
- Each project must be with an enrollment.



Aggregation

- A better model is let “student”, “instructor”, “course”, and “enroll” form an abstract entity as an aggregation.
- Then “project” only need to associate with the courses which has a course project.



Weak Entity Sets

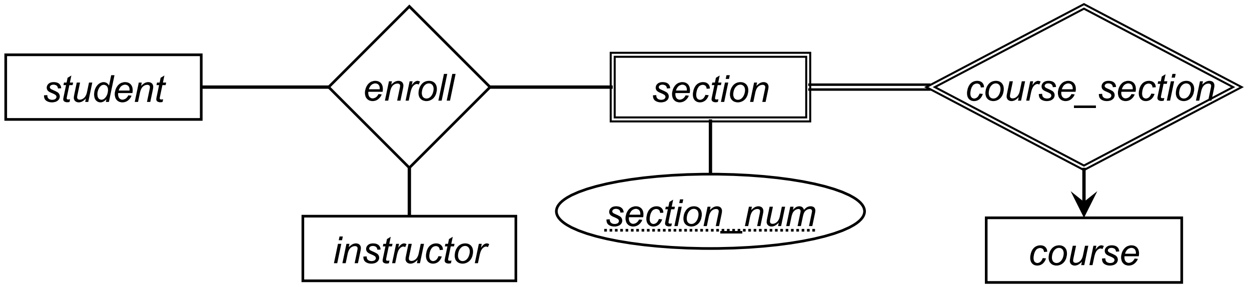
- A ***weak entity*** is an entity cannot exist alone. The existence depends on another entity of a different type.
- A set of weak entities is a ***weak entity set***.
- Since a weak entity cannot exist alone, it does not have a key.
- The existence of a weak entity set depends on the existence of an **identifying entity set.**
  - A weak must relate to the identifying entity set via a **total**, **one-to-many** relationship set from the identifying to the weak entity set.
  - An **identifying relationship** depicted using a double diamond.
- The **discriminator** (*or partial key)* of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set.
- To distinguish weak entities, one must combine the key of the identifying entity set and the discriminator.

Weak Entity Sets

- Our original modeling for courses was not accurate.
- Instead of saying “a student is enrolled to a course”, it’s better to say, “a student is enrolled to a section of a course”.
- Same for instructors, one instructor may teach multiple sections of one course.
- An entity set “section” is needed.
- But a section needs to be with a course.
- Thus, “section” is a weak entity set which depends on “course”.

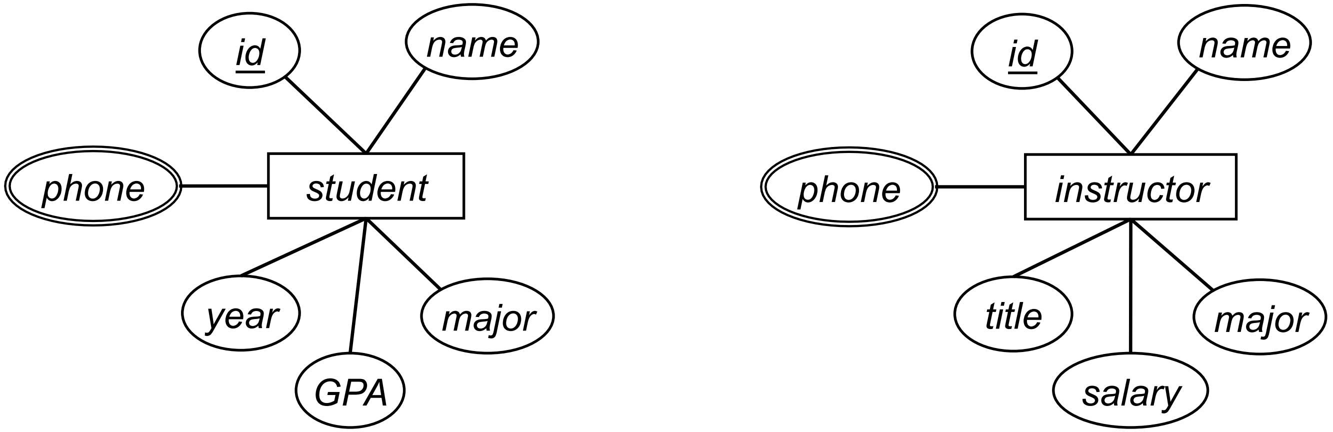
Weak Entity Sets

- ***Double rectangles*** denote weak entity sets.
- ***Dashed underlines*** denote discriminators of weak entity sets.
- ***Double diamonds*** denote identifying relationship sets.
- Assume we model the original “enroll” relationship set without constraints.
- After adding the weak entity set “section”,



Generalization and Specialization

- Let’s look at the entity sets “student” and “instructor” again.



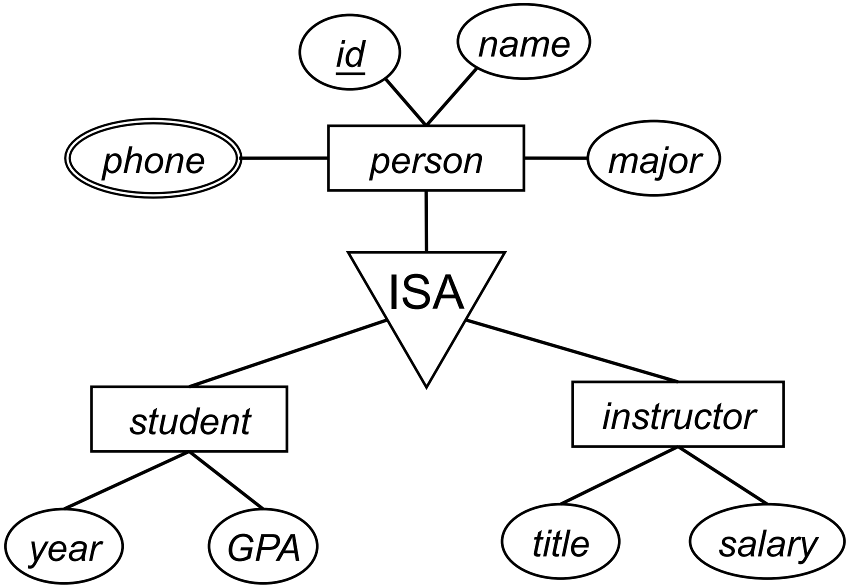
- Both of them have attributes id, name, major, and phone number because they are all persons.
- At the same time, these two entity sets cannot be merged as one because students are not instructors.

Generalization and Specialization

- For this case, an entity set “person” is usually created to hold the common attributes.
- This is called ***generalization***, finding common properties among several different entity sets.
- The entity sets “student” and “instructor” are also remained to keep their special attributes.
- A special relationship set, ***ISA*** is used to show the connection among “person”, “student”, and “instructor”.
- The intuition is that a student ***is a*** person.
- The entity set “person” is also a super set of “student” and “instructor”.

Generalization and Specialization

- ISA relationship sets are denoted by ***triangles***.
- The super set is usually placed above the triangle.



Generalization and Specialization

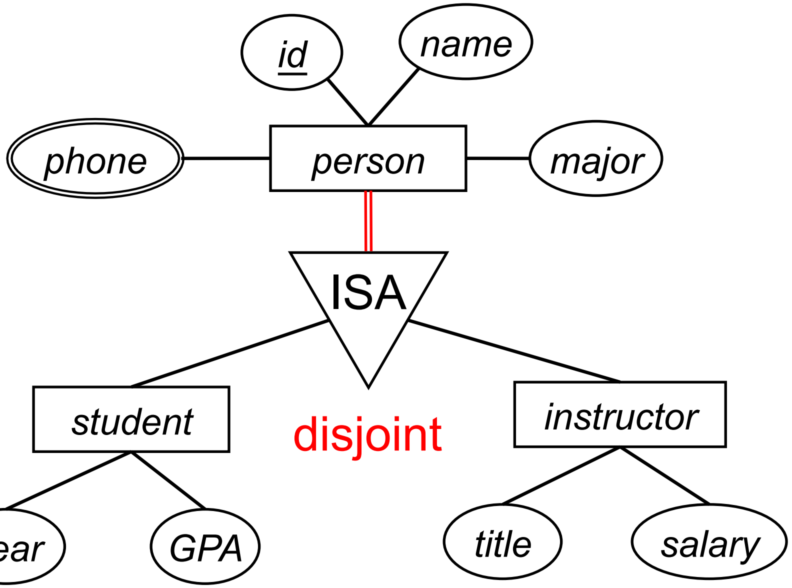
- The above design process is ***bottom-up***, combining several entity sets with same attributes into higher-level entity set.
- But sometimes the process is reversed.
- Some entity sets can be split into some lower-level entity sets with specific attributes.
- This ***top-down*** process is called ***specialization***.

Constraints on Generalization/Specialization

- Cardinality constraints and participation constraints can be applied on ISA relationship sets.
- But some constraints are omitted because
  - each entity on the lower-level is one entity on the higher-level (the entity sets on the lower-level always fully participate ISA);
  - one entity cannot be associated with multiple entities in the same entity set on the lower-level.
- Thus, the constraints on ISA only discuss
- Does one higher-level entity belong to at least one lower-level entity set?
  - If yes, this ISA is a ***total generalization***.
- Does one higher-level entity belong to multiple lower-level entity set?
  - If No, it is a ***disjoint generalization***.

Constraints on Generalization/Specialization

- Consider the “student” and “instructor” example.
- If we want to express every person is either a student or an instructor, then this is a total generalization.
- And if we assume nobody can be a student and an instructor at the same time, this is a disjoint generalization.

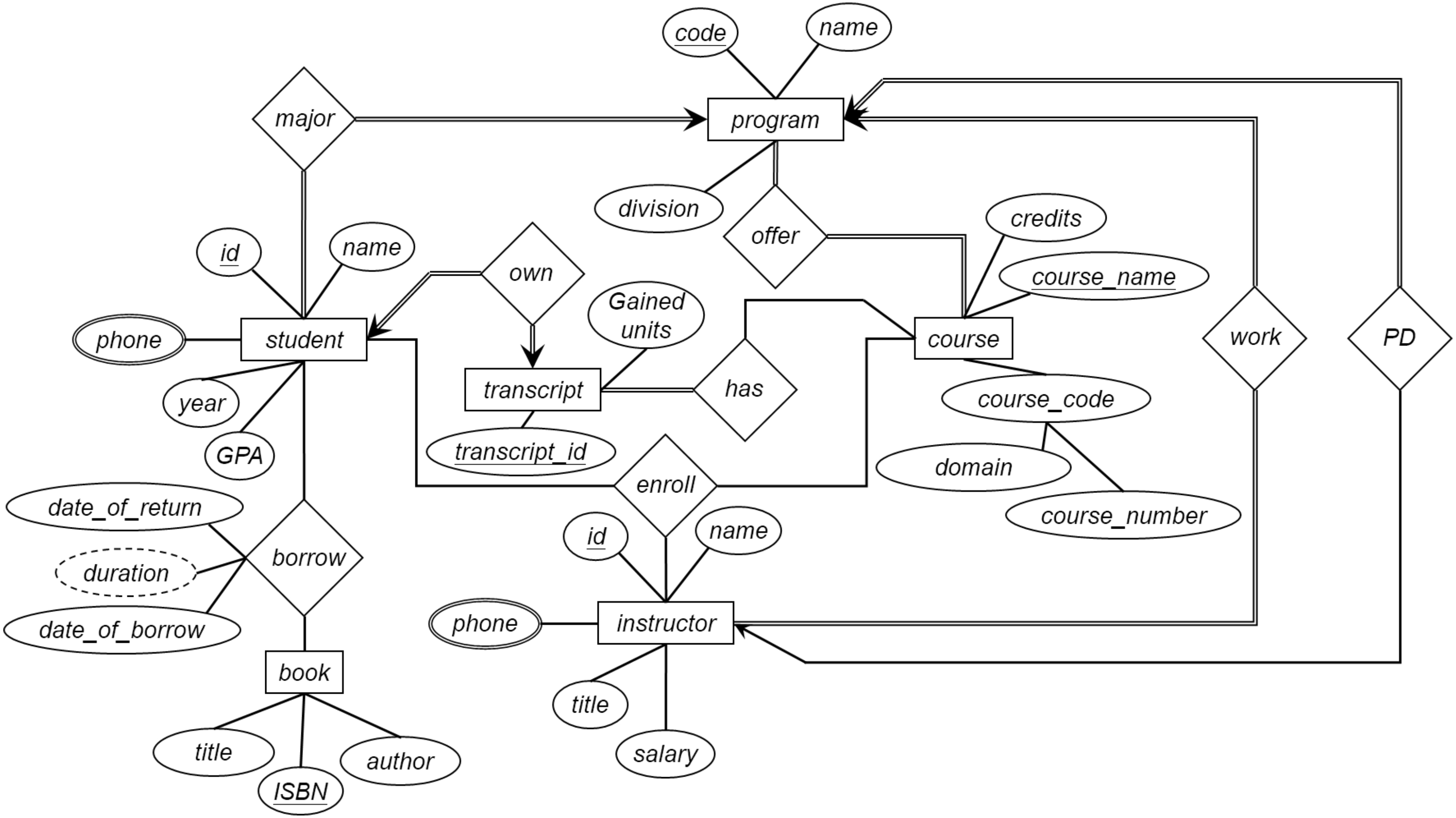


Exercises

- Recall the ER diagram of the exercise from the last lecture.

1. Add the weak entity set “section”.
2. Generalize “student” and “instructor”.
3. Associate the entity sets by relationship sets properly.

Exercises


code
name
major
program
division
offer
credits
course_name
id
name
own
Gained units
student
phone
course
work
PD
transcript
has
year
course_code
GPA
transcript_id
domain
enroll
date_of_return
id
name
course_number
borrow
duration
date_of_borrow
phone
instructor
book
title
title
salary
author
ISBN

End of Lecture 4