ER Modeling - constraints

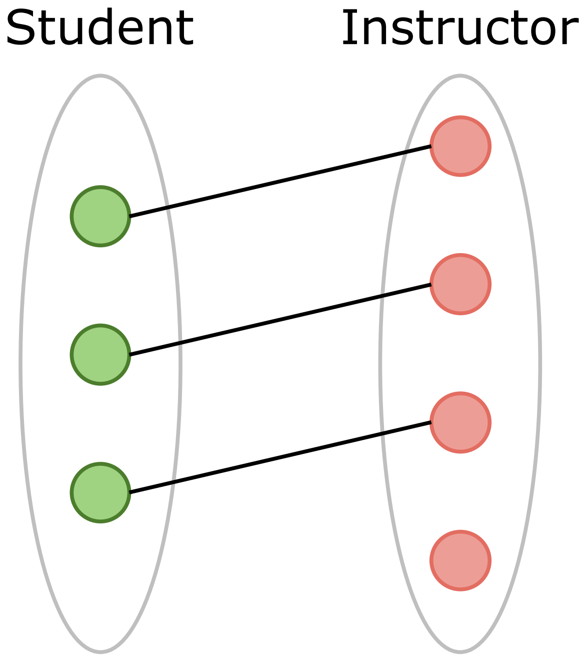
Motivation

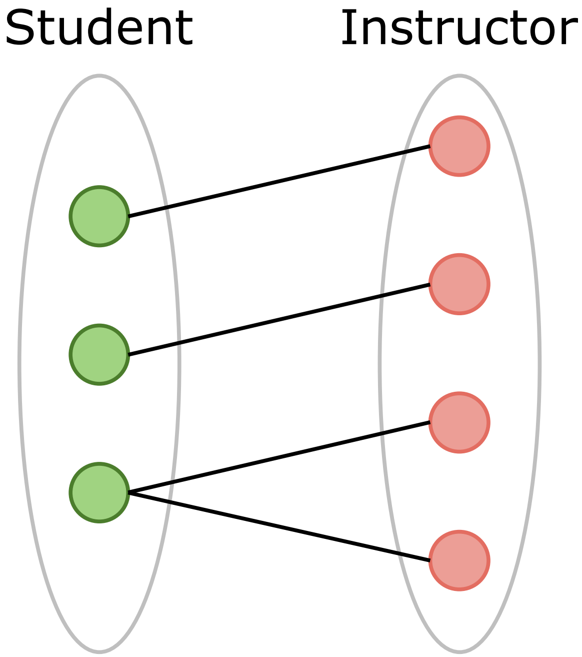
- So far, our ER models let us talk about:
  - some entities
  - their relationships
  - attributes of entities and relationships
- Can an ER model answer these questions?
  - can one session be taught by multiple instructors?
  - must every student have an advisor?

Mapping cardinalities

Can a student have more than one advisor?

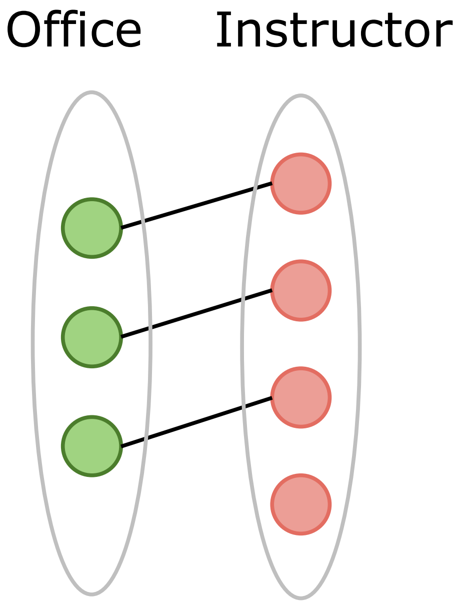
Can an instructor advise more than one student?





Mapping cardinalities: one-to-one

Example: office-of relationship set

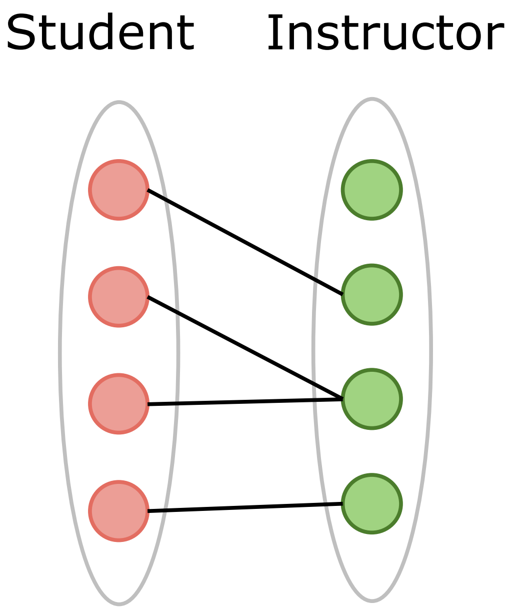


An office can be assigned to at most one instructor

An instructor can have at most one office

Mapping cardinalities: many-to-one

Example: adviser relationship set

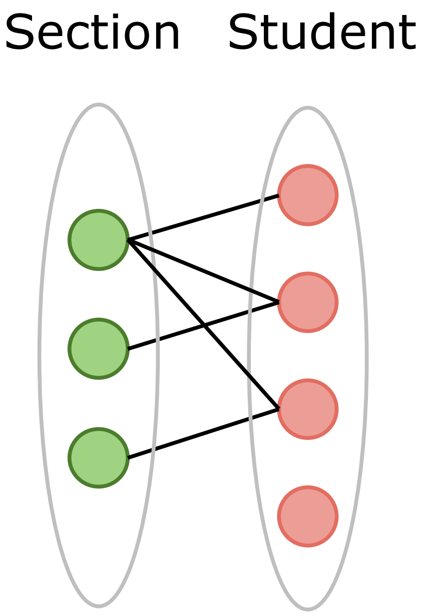


A student has at most one adviser

An instructor can have zero or more students

Mapping cardinalities: many-to-many

Example: takes relationship set



What are the mapping cardinalities for:

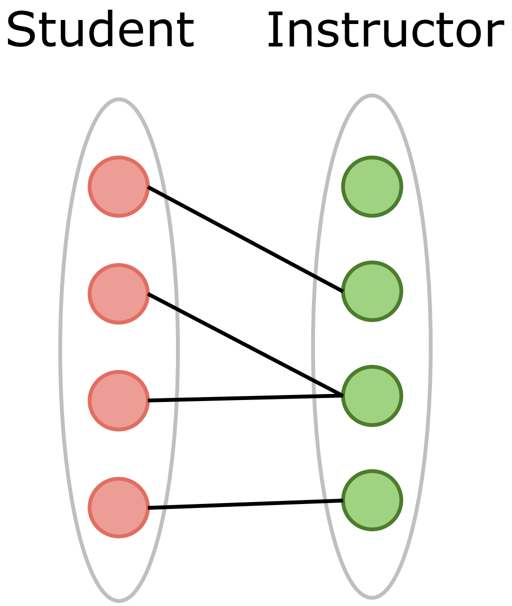
- teaches?
- pre-req?

Each section is taken by zero or more students

Each student takes zero or more sections

Participation constraints

- Only two cases:
  - total
  - partial
- Remember, this is a constraint
  - prescriptive, not descriptive



Participation by Student is total

Participation by Instructor is partial

Keys of entities

- No two entities in an entity set can have the same attribute values
- Definitions of key, superkey, candidate key all apply

Complication: keys of relationship sets

- Example: the advises relationship set between instructor and student
- primary keys of the entity sets:
  - instructor: inst_ID
  - student: student_ID
- So a superkey for ‘advises’ is:
  - inst_ID, student_ID

In other words: we can uniquely identify a relationship if we can uniquely identify the entities in the relationship

Primary keys for relationship sets

Mapping cardinality:	Primary key:
many-to-many (a student can have more than one advisor; an instructor can advise more than one student)	advises(inst_ID, student_ID)
many-to-one (a student can have at most one advisor; instructors can advise many students)	advises(student_ID, inst_ID)

one-to-one
(a student can have at most one advisor, similarly for instructors)
	advises(student_ID, inst_ID)
	or
	advises(inst_ID, student_ID)

Removing redundant attributes

Example:

- instructor(ID, name, dept_name, salary)
- department(dept_name, building, budget)

We might decide to add relationship set inst_dept between instructor and department.

Then dept_name appears in both entity sets.

It should be removed from instructor – it is the key of department.

A method for developing an ER model

1. identify entity sets
2. identify relationship sets
3. identify mapping cardinalities
4. identify participation constraints
5. identify attributes of entity and relationship sets (and their domains)
6. identify primary keys of entity sets
7. identify primary keys of relationship sets

This is not a rule (it's more of a guideline…)

Try it and see if it's helpful