Deriving relational schemas from ER designs

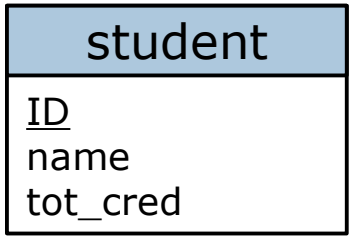
Motivation

- The point of conceptual design is to put stakes in the ground and get agreement.
- How to get from the conceptual design to the logical design?

Key idea: you can follow a well-defined procedure to go from an ER model to a relational schema.

You just need to learn the rules.

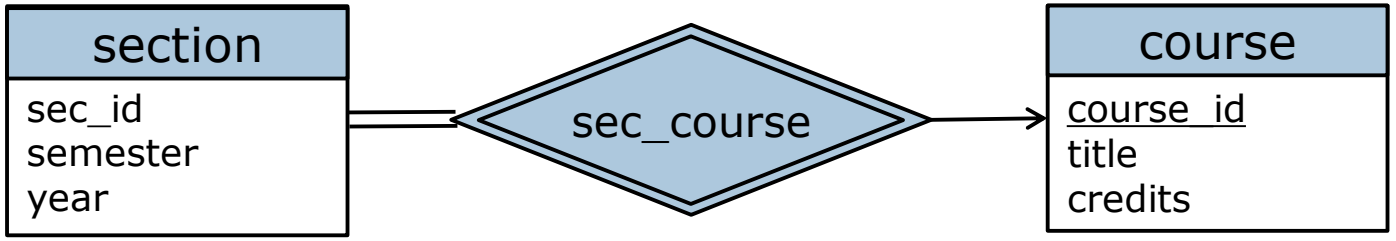
Translating a strong entity set



student(ID, name, tot_cred)

This is the rule for strong entity sets with simple attributes

Translating a weak entity set



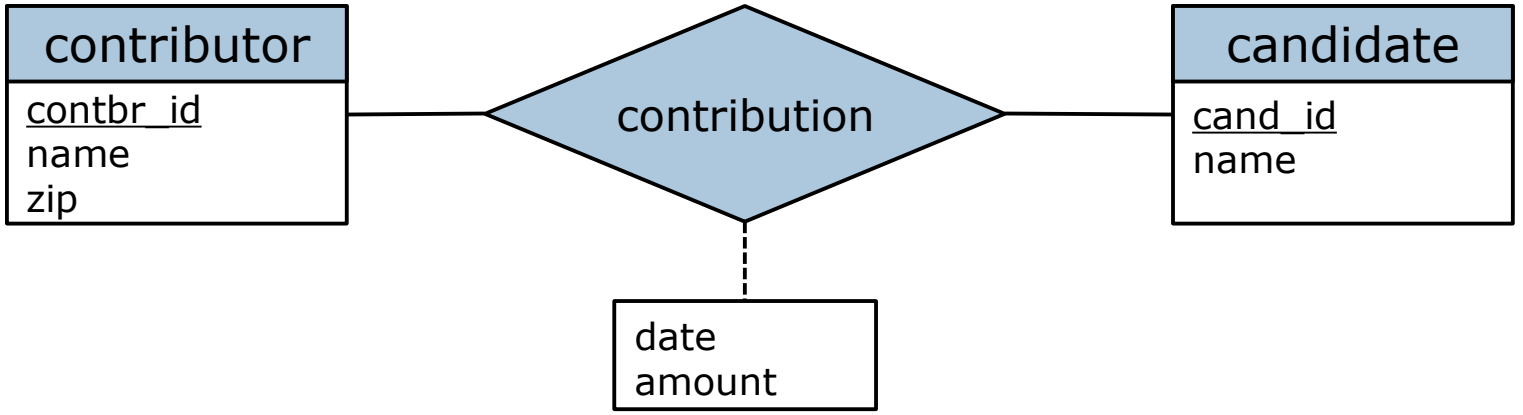
section(course_id, sec_id, semester, year)

\+ foreign key (course_id) references course

(foreign key constraint on section)

Remember that the primary key of a weak entity set = partial key of the weak entity set + primary key of the identifying set

Translating a relationship set, case 1



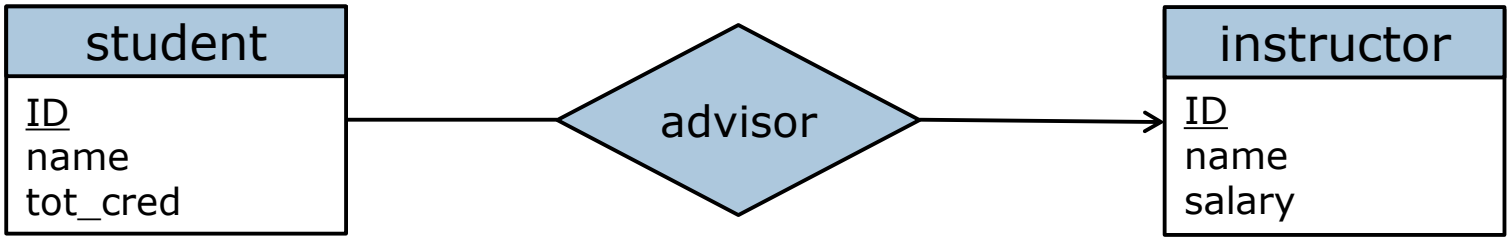
contribution(contbr_id, cand_id, date, amount)

\+

foreign key (contbr_id) references contributor
foreign key (cand_id) references candidate

In many-to-many relationships, use primary keys of related entity sets, plus attributes of the relationship set

Translating a relationship set, case 2



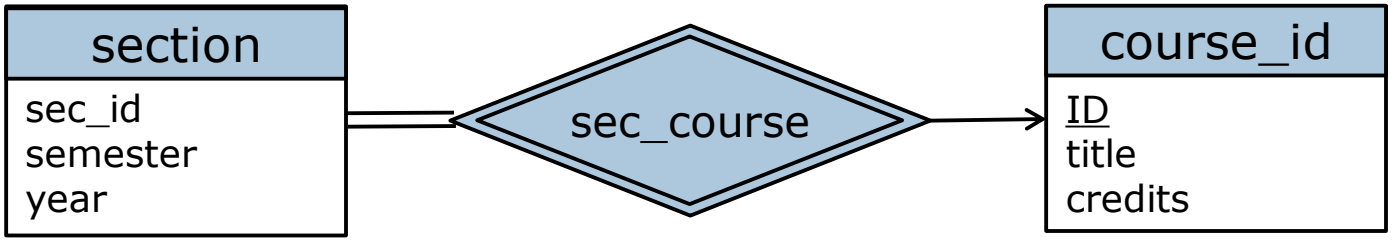
advisor(student_ID, instructor_ID)

(foreign key constraints not shown)

Primary keys of the entity sets become attributes, but since mapping cardinality is many-to-one, instructor_ID is not part of primary key.

If the mapping cardinality were one-to-one, either student_ID or instructor_ID could be key of advisor.

Translating a relationship set, case 3



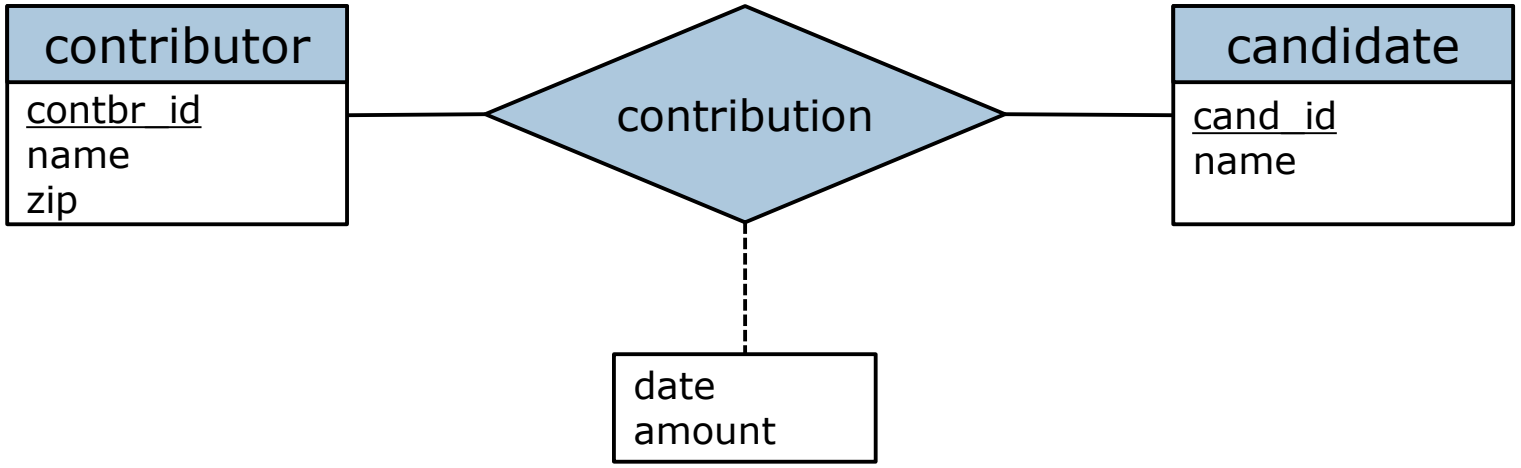
sec_course(ID, sec_id, semester, year) ?

Does this make sense?

No – such a relational schema would duplicate the relational schema derived from 'section'.

So no relational schema is produced. This is true for any relationship set linking a weak entity set to its corresponding strong entity set.

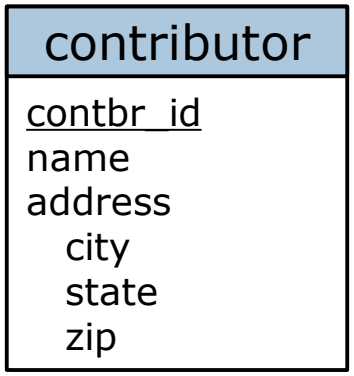
Exercise



?

contributor(<u>contbr_id</u>, zip)
candidate(<u>cand_id</u>, name)
contribution(<u>contbr_id</u>, <u>cand_id</u>, date, amount)

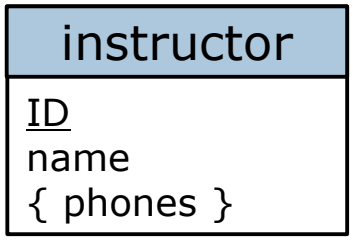
Composite attributes



contributor(contbr_id, name, cite, state, zip)

This composite attribute itself goes away

Multi-valued attributes

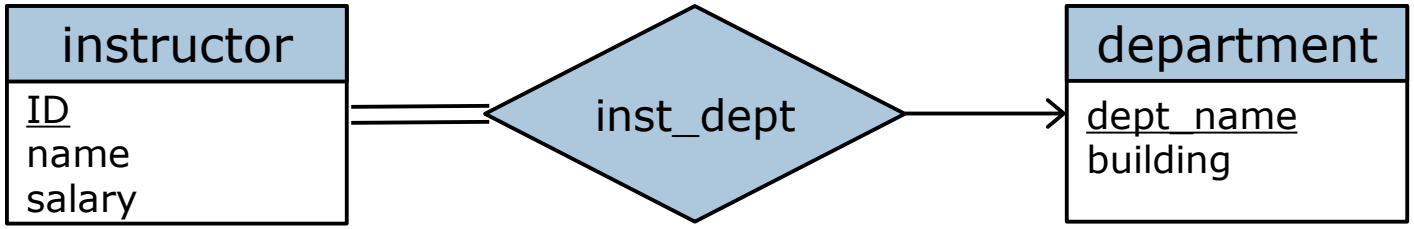


instructor(ID, name)
instructor_phone(ID, phone_number)

\+ foreign key (ID) references instructor

(foreign key constraint on instructor_phone)

Combining schemas

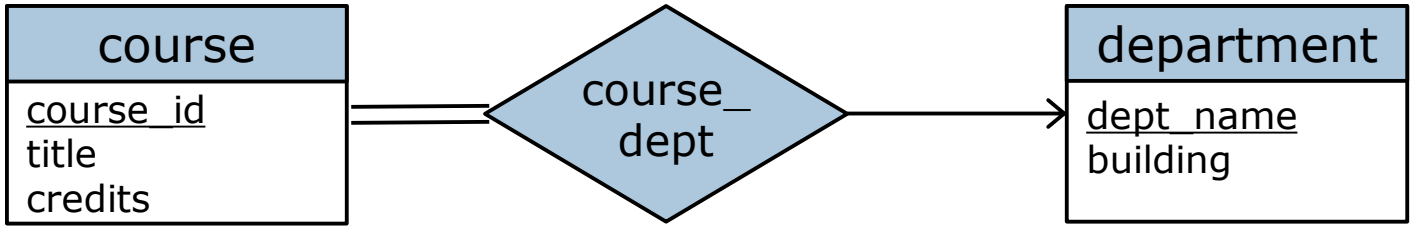


instructor(ID, name, salary)
inst_dept(ID, dept_name)
department(dept_name, building)

combine

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

Exercise: combining schemas



course(course_id, title, credits)
course_dept(course_id, dept_name)
department(dept_name, building)

combine

course(course_id, title, credits, dept_name)
department(dept_name, building)