Deriving relational schemas from ER designs

CST 363

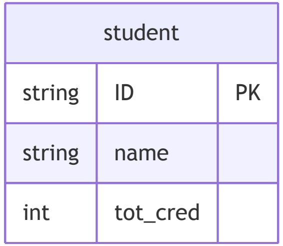
Motivation

- 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 know the rules.

Translating a strong entity set

This is the rule for strong entity sets with simple attributes

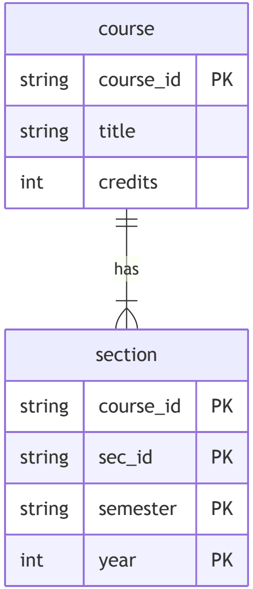


student		
string	ID	PK
string	name	
int	tot_cred	

➡ `student(ID, name, tot_cred)`

This is the rule for strong entity sets with simple attributes

Translating a weak entity set



course		
string	course_id	PK
string	title	
int	credits	

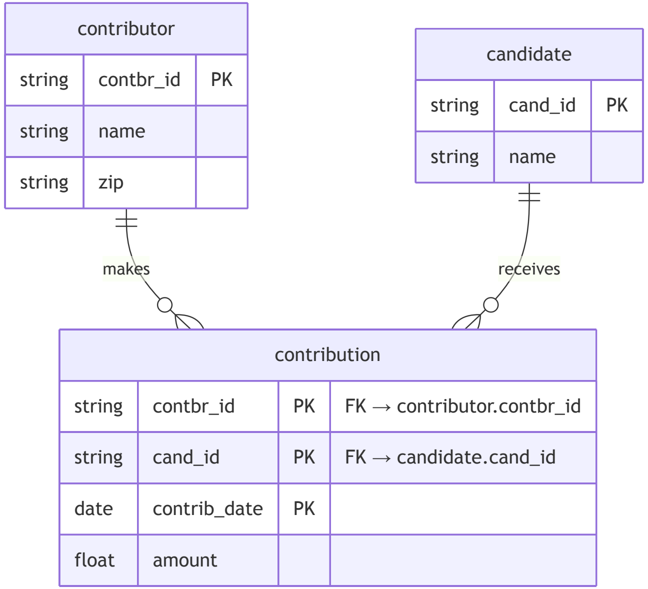
section		
string	course_id	PK
string	sec_id	PK
string	semester	PK
int	year	PK

➡ `section(course_id, sec_id, semester, year)`

+ foreign key (`course_id`) references `course(course_id)`

> Remember: the primary key of a weak entity set =
>
> PK of the identifying (owner) entity set + partial key of the weak entity set.
>
> Here: `PK(section) = course_id + (sec_id, semester, year)`

Translating a relationship set, case 1



contributor

string	contbr_id	PK
string	name	
string	zip	

candidate

string	cand_id	PK
string	name	

contribution

string	contbr_id	PK	FK → contributor.contbr_id
string	cand_id	PK	FK → candidate.cand_id
date	contrib_date	PK	
float	amount		

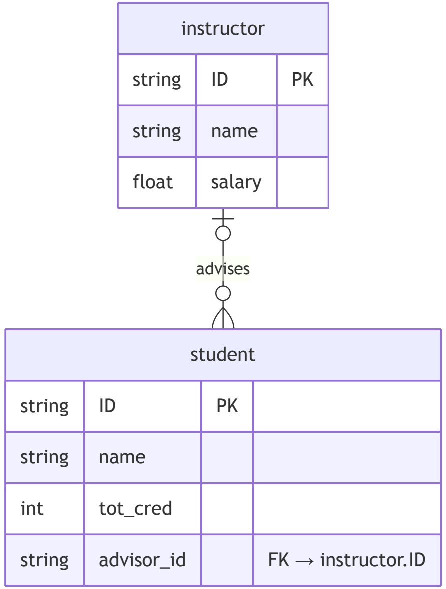
➡ `contribution(contbr_id, cand_id, contrib_date, amount)`

+ foreign key (`contbr_id`) references `contributor(contbr_id)`

+ foreign key (`cand_id`) references `candidate(cand_id)`

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

Translating a relationship set, case 2



instructor		
string	ID	PK
string	name	
float	salary	

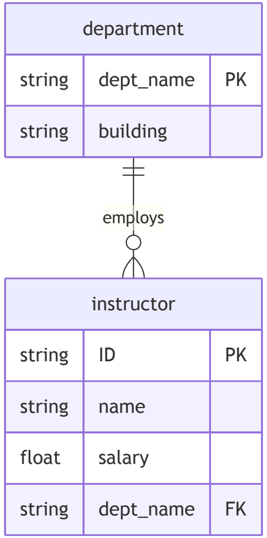
student			
string	ID	PK	
string	name		
int	tot_cred		
string	advisor_id		FK → instructor.ID

➡ `student(ID, name, tot_cred, advisor_id)`

+ foreign key (`advisor_id`) references `instructor(ID)`

In a 1:N relationship, add the primary key of the “1” side as a foreign key to the “N” side

Translating a relationship set, exercise



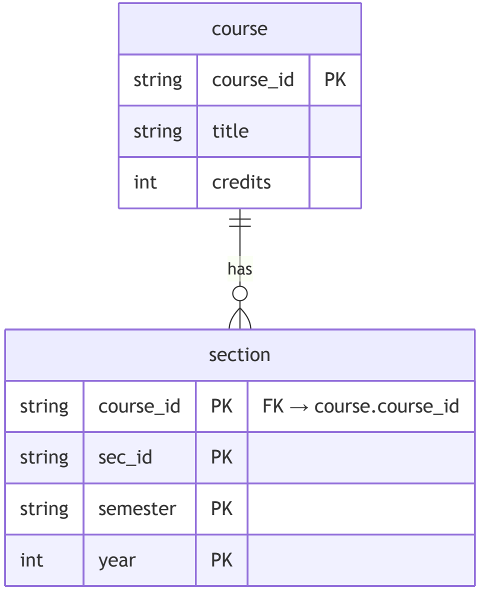
department		
string	dept_name	PK
string	building	

instructor		
string	ID	PK
string	name	
float	salary	
string	dept_name	FK

```
department(dept_name PK, building)

instructor(ID PK, name, salary, dept_name FK NOT NULL)
  FK (dept_name) → DEPARTMENT(dept_name)
```

Translating a relationship set, case 3



course		
string	course_id	PK
string	title	
int	credits	

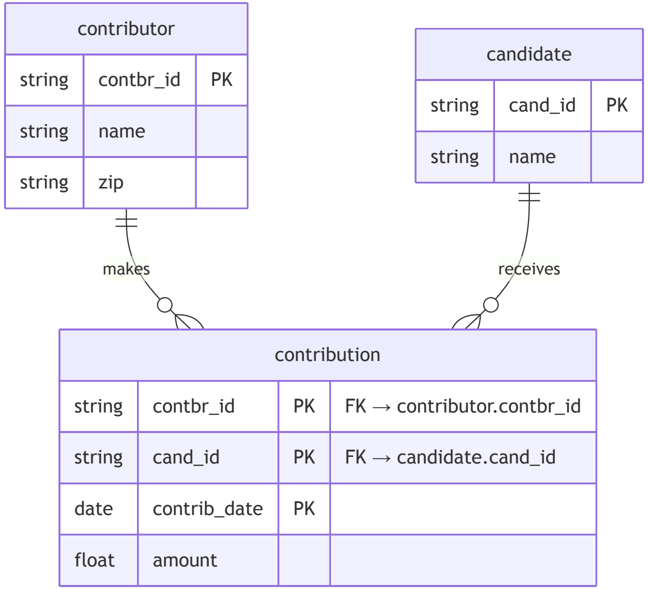
section			
string	course_id	PK	FK → course.course_id
string	sec_id	PK	
string	semester	PK	
int	year	PK	

➡ `sec_course(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		
string	contbr_id	PK
string	name	
string	zip	

candidate		
string	cand_id	PK
string	name	

contribution			
string	contbr_id	PK	FK → contributor.contbr_id
string	cand_id	PK	FK → candidate.cand_id
date	contrib_date	PK	
float	amount		

```
contributor(contbr_id, name, zip)
candidate(cand_id, name)
contribution(contbr_id, cand_id, contrib_date, amount)
```

Summary

You can translate an ER model to a relational schema by following a set of rules.

- an entity set maps to a relational schema
- attributes of schema are attributes of the entity set
- a relationship set maps to a relational schema
- attributes are keys of the related entity sets, plus attributes of the relationship set