

CSC7072: Databases, fall 2015

Dr. Kim Bouters



ER model

ER model

retrieving data using SQL

don't forget:

```
SELECT {attribute [AS new_attribute_name]}  
      FROM {table [AS new_table_name]}  
      [{JOIN table ON attribute = attribute}]  
      [WHERE {condition}]  
      [GROUP BY {attribute}]  
      [HAVING {condition}]  
      [ORDER BY {attribute}]
```

where {argument} denotes you need to have at least one, and
where [argument] denotes a part that is optional and can be omitted

ER model

creating databases

so we know how to use databases ...

... but how do we create a database of our own?

multi-step process:

- conceptual database modelling phase
data requirements are expressed using an ER model
- logical database design phase
normalisation is used to remove redundancies, often by splitting tables, to make modifications to the information easier
- database description language (DDL)
SQL is used to define and create the database

ER model



bird's eye view of an ER model

what is an ER model?

used to express the data requirements as:

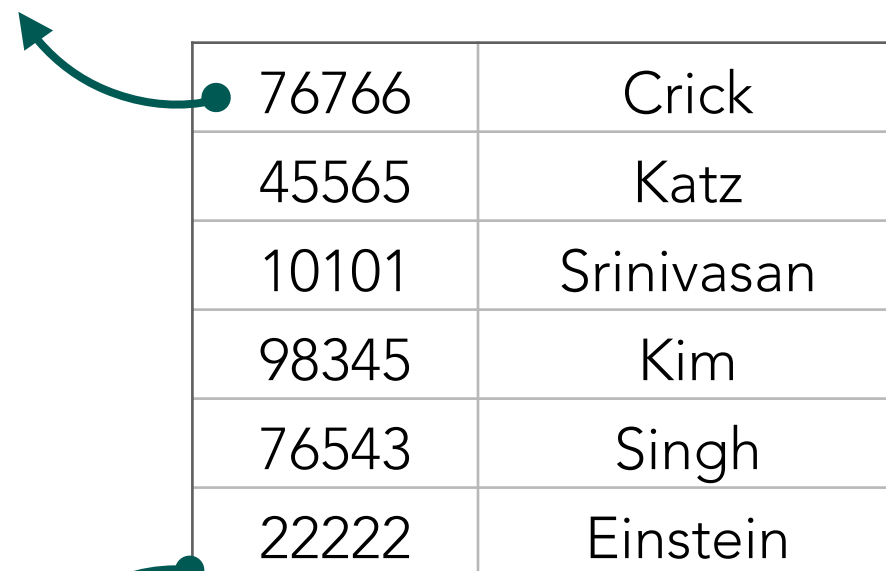
- a set of *entities*
something that exists, and is distinguishable from other entities
e.g. a student, a book, an order, an event, a company ...
 - each entity can have *attributes* (e.g. a name, a value ...)
 - an *entity set* is a group of entities of the same type
- *relationship* between those entities
used to link entities together and highlight the structure
 - can also have attributes,
called *relationship set* when linking entity sets

ER model

closer look at entities

defining an entity (set)

a single instructor,
i.e. an *entity*



76766	Crick
45565	Katz
10101	Srinivasan
98345	Kim
76543	Singh
22222	Einstein

instructor

an entity set, *i.e.*
a group of instructors

98988	Tanaka
12345	Shankar
00128	Zhang
76543	Brown
76653	Aoi
23121	Chavez
44553	Peltier

student

note: having the same attributes (an id and a name) does not mean they are necessarily the same type of entities!

ER model



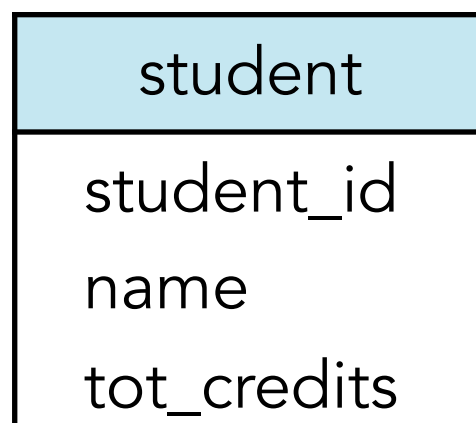
entity sets: notation

formal notation

an entity (set) is denoted by a box (typically a *noun*):



↗ descriptive property possessed by all members of the entity set
and attributes can be added in a UML-like way:



ER model

relationship sets: notation



formal notation

a relationship is denoted by a diamond (typically *conjugated verb*):



example of a *binary* relationship (linking together two entities):



by far the most common type of relationship; rare (but not impossible) to find a relationship that links together more than two entities

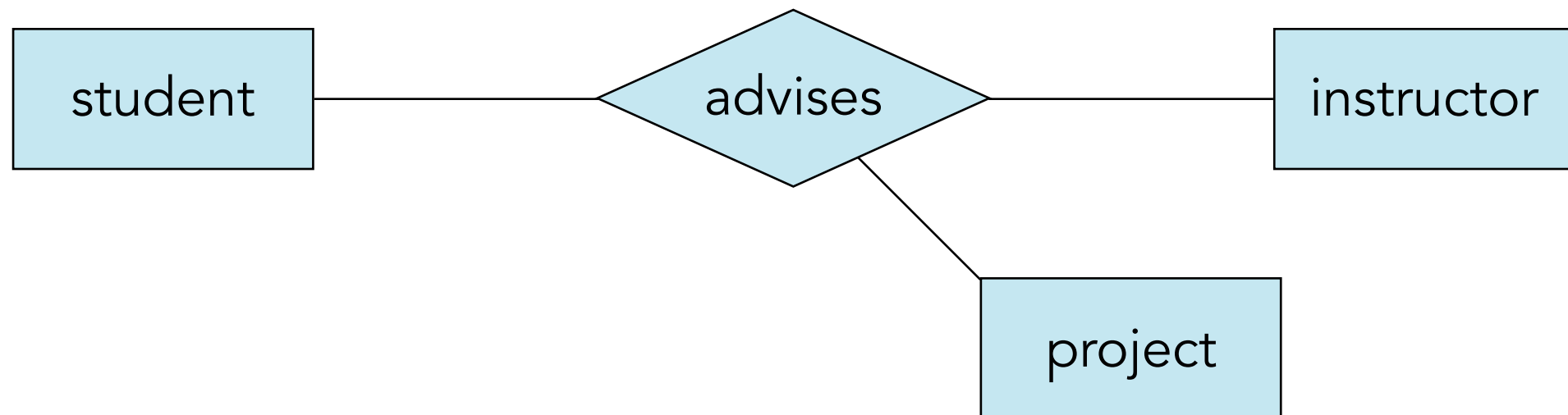
ER model

relationship sets: ternary relationships

formal notation

relationships can become very complicated (contrived?)

example of a *ternary* relationship (linking together three entities):



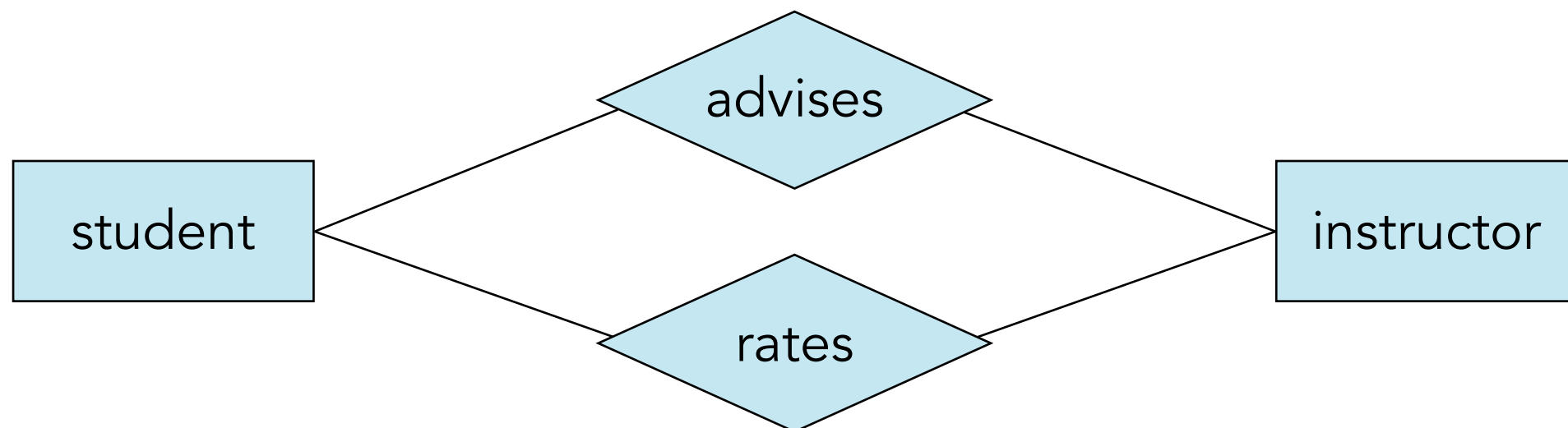
ER model

relationship sets: multiple relationships

formal notation

relationships can become very complicated (contrived?)

example of a *multiple* relationships between entities:



be careful with these!

ternary relationships can be ok,
multiple relationships could cause dependencies

ER model

your turn

try creating an ER diagram yourself!

problem

The work of a writer is to write novels.

ER model

your turn

try creating an ER diagram yourself!

problem

The work of a writer is to write novels.

The work of a writer is to write novels.

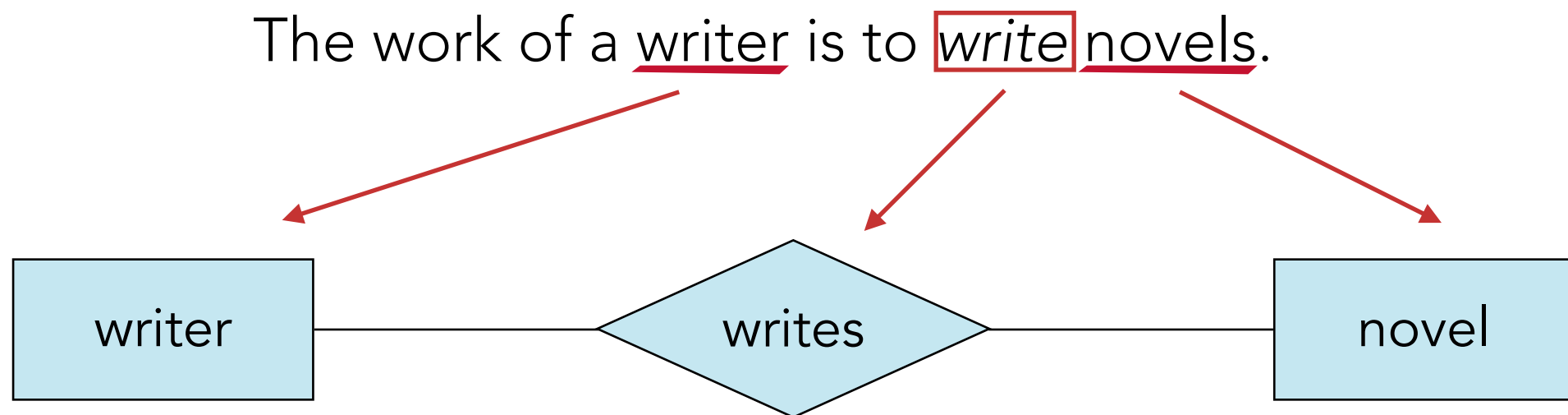
ER model

your turn

try creating an ER diagram yourself!

problem

The work of a writer is to write novels.



ER model

attributes



another look at attributes

what are good attributes?

properties possessed by all members of an entity/relationship set

each property has a *domain* of values



the set permitted values for a property

what types of attributes do we have?

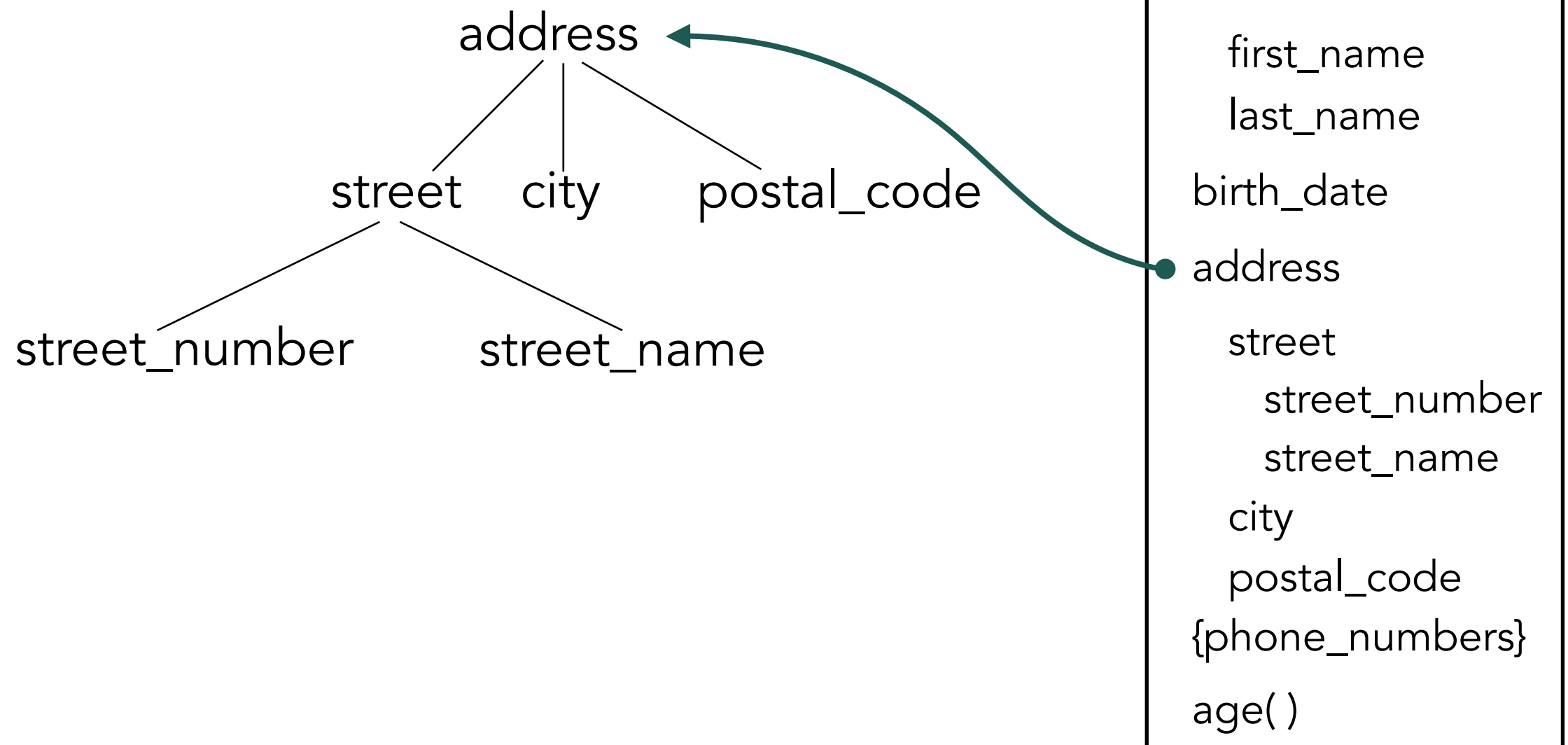
- simple (e.g. title) and composite (e.g. name, address)
- single-valued (e.g. title) and multi-valued (e.g. phone_numbers)
- derived (e.g. age, derived from birthdate)

ER model

attributes



decomposition of a (nested) composite attribute:

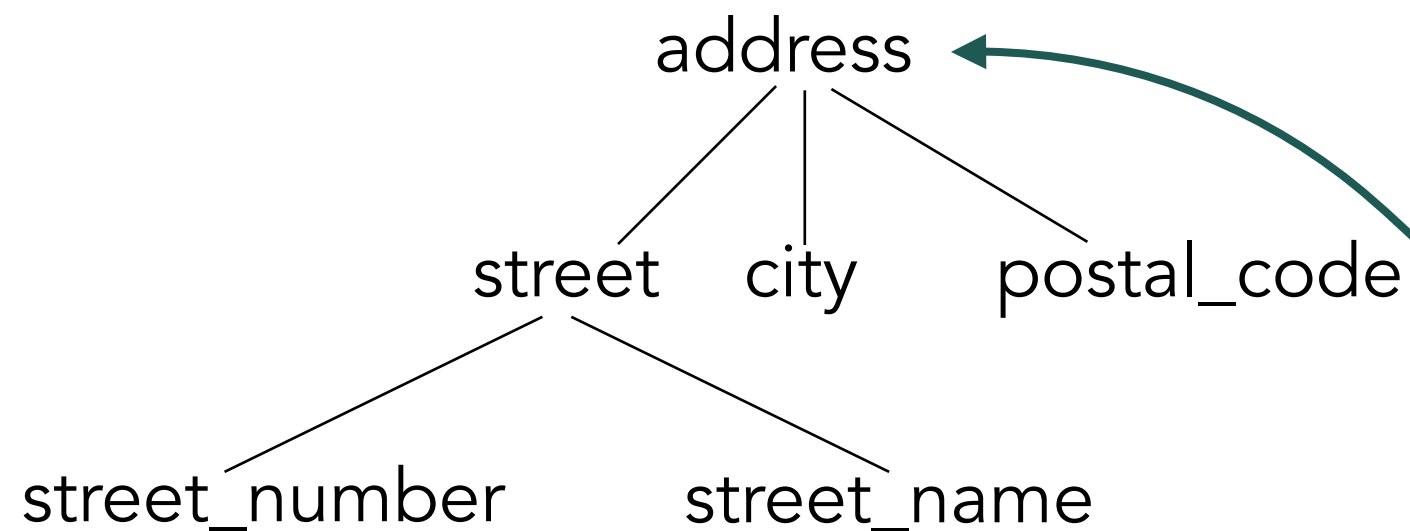


ER model

attributes



decomposition of a (nested) composite attribute:

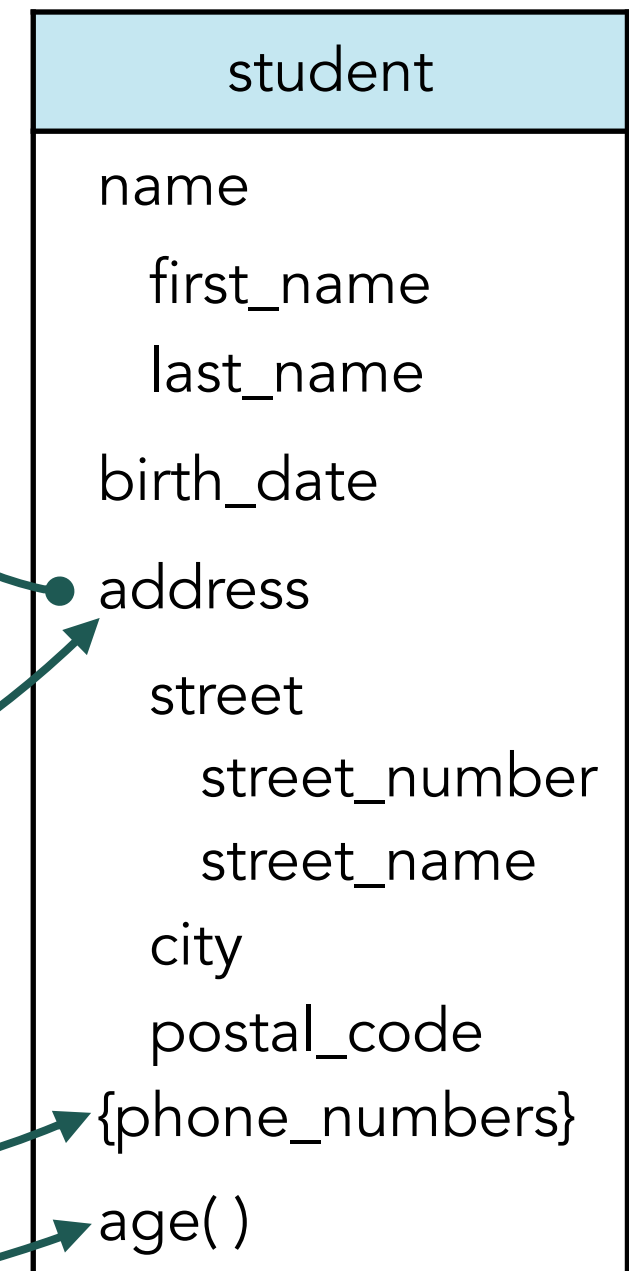


notation for:

■ composite

■ multi-valued

■ derived



ER model

your turn

try creating an ER diagram yourself!

problem

The work of a writer is to write novels. Every writer has an address and a unique name. Every novel has a unique title.

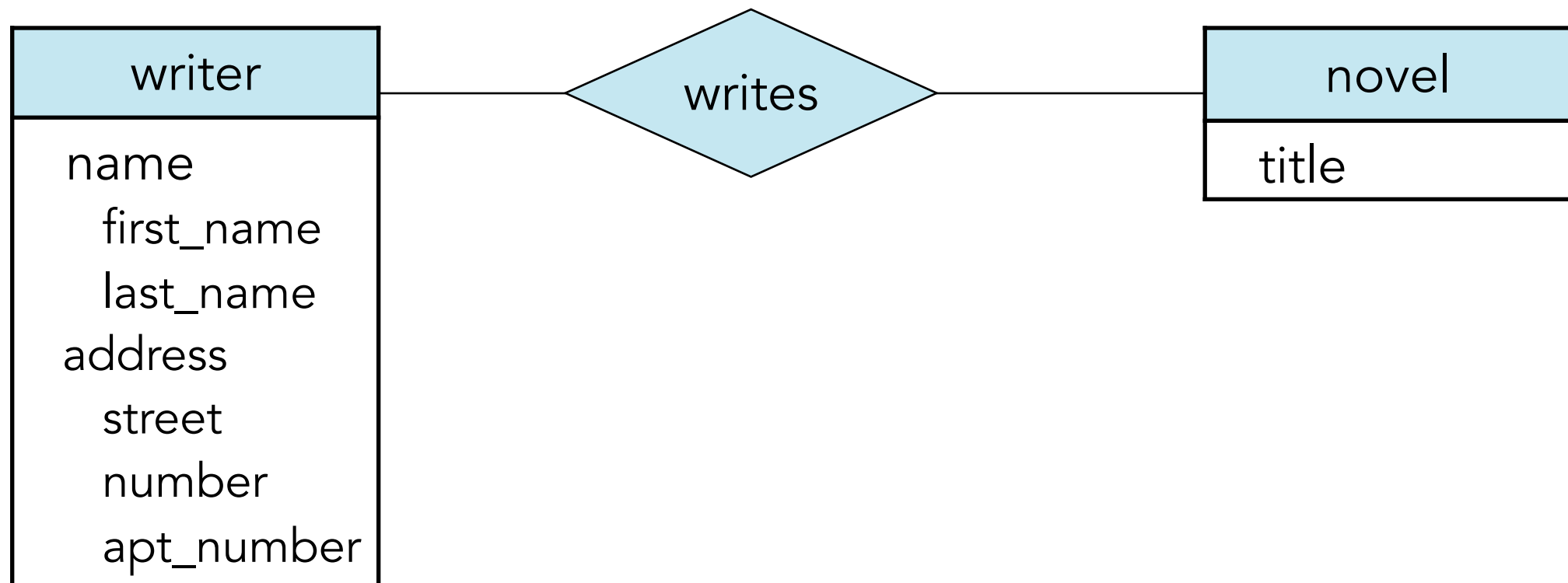
ER model

your turn

try creating an ER diagram yourself!

problem

The work of a writer is to write novels. Every writer has an address and a unique name. Every novel has a unique title.



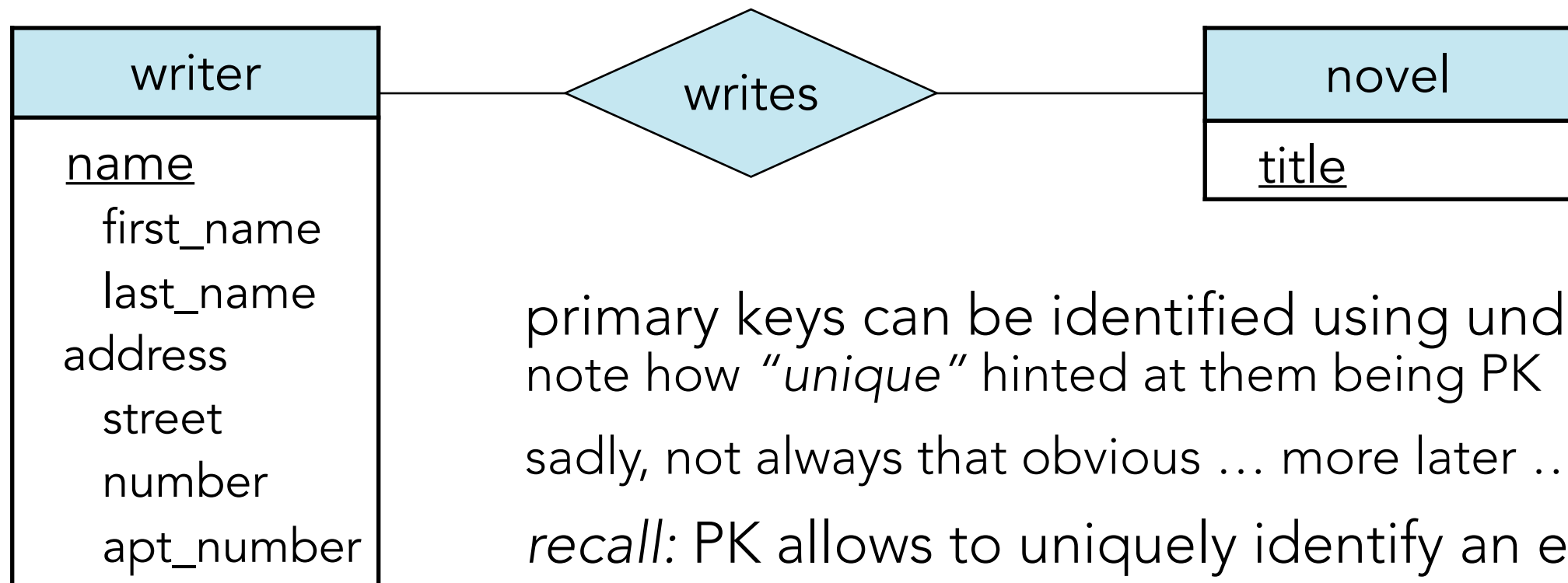
ER model

your turn

try creating an ER diagram yourself!

problem

The work of a writer is to write novels. Every writer has an address and a unique name. Every novel has a unique title.



primary keys can be identified using underline
note how "unique" hinted at them being PK
sadly, not always that obvious ... more later ...
recall: PK allows to uniquely identify an entity

ER model

relationship constraints

a useful property when describing relationships is their *cardinality*

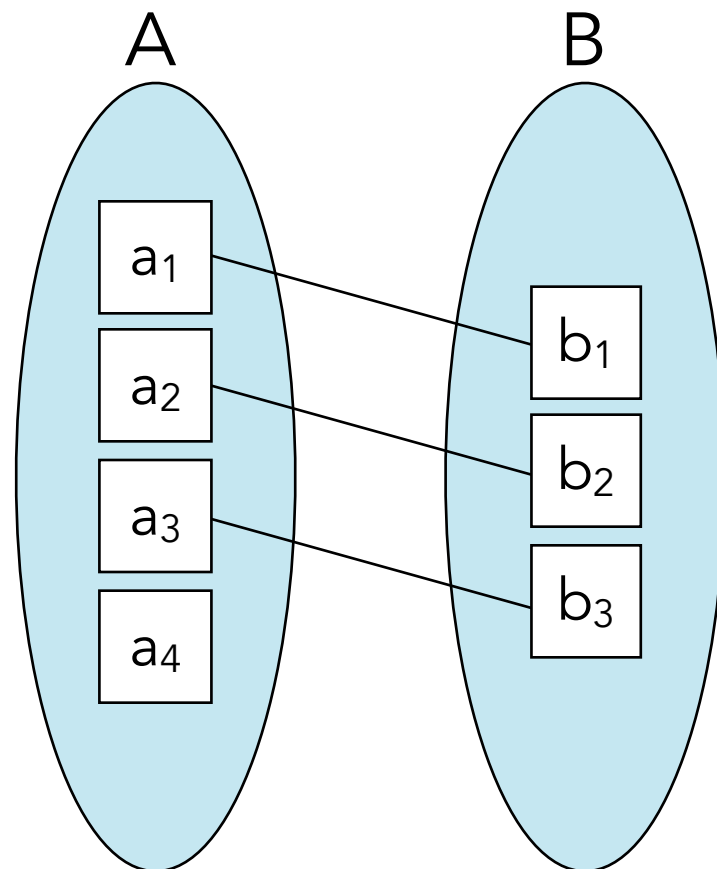
express the number of entities to which an entity can be associated via a relationship set

most useful when discussing binary relationships, which can be:

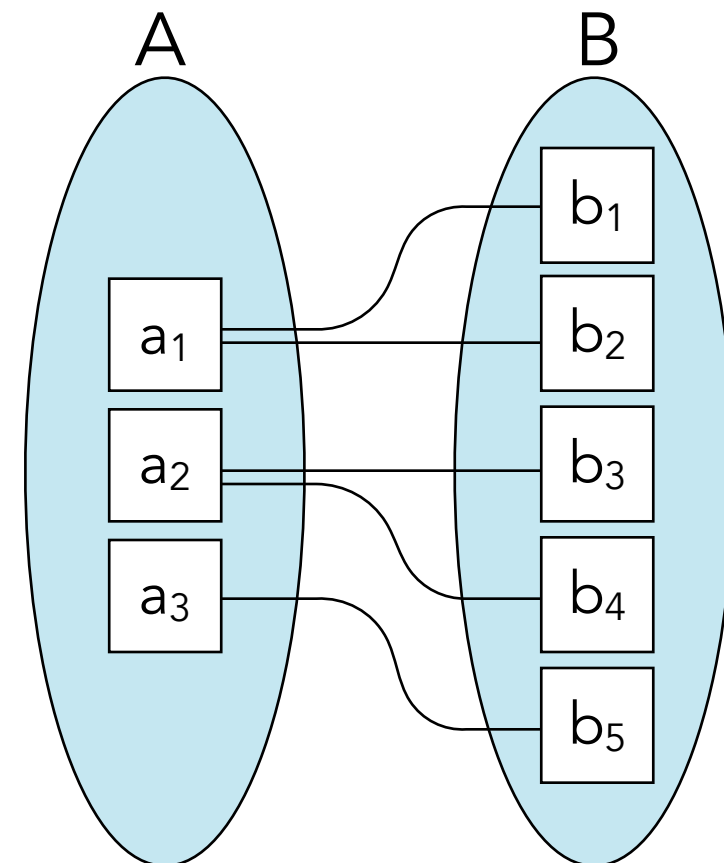
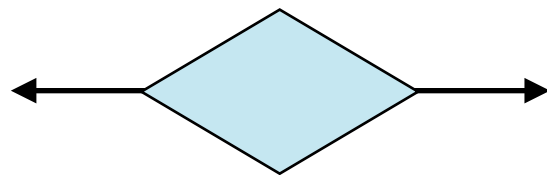
- one-to-one;
- one-to-many;
- many-to-one;
- many-to-many.

ER model

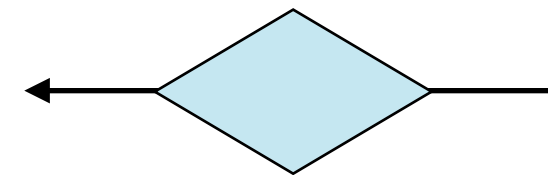
constraints: mapping cardinalities



one-to-one



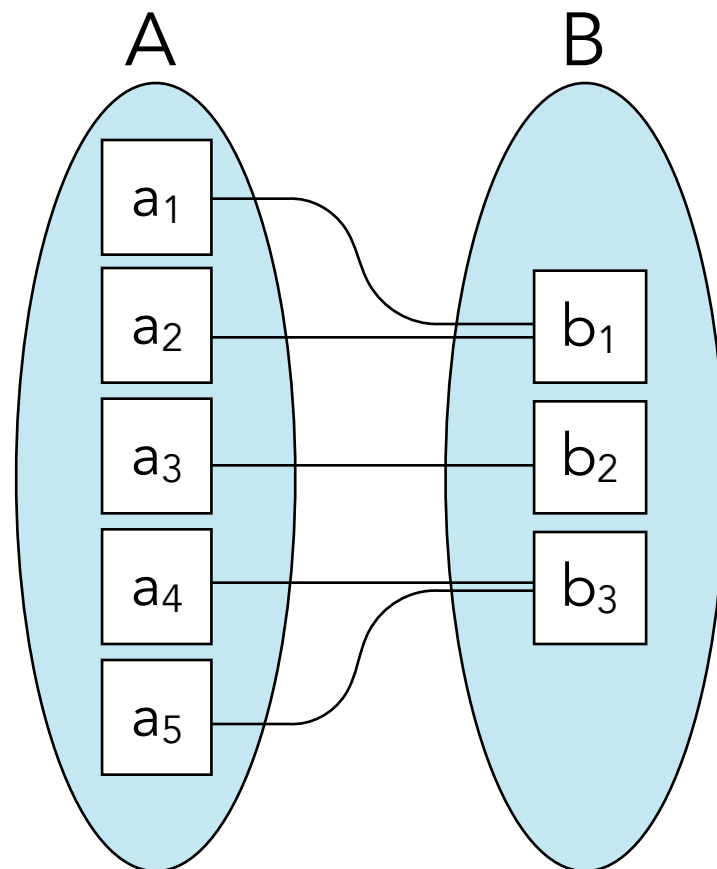
one-to-many



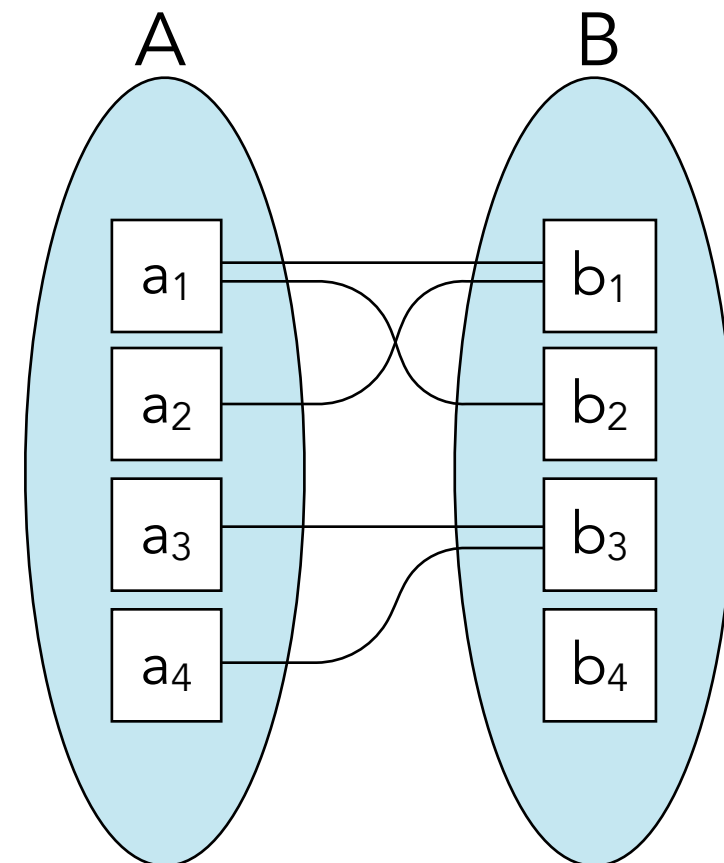
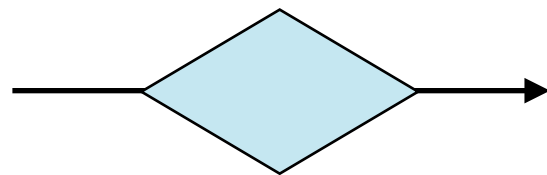
important: some elements in A (or B) may not be mapped to an element in B (or A)

ER model

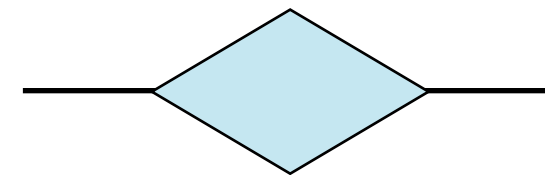
mapping cardinalities



many-to-one



many-to-many



important: some elements in A (or B) may not be mapped to an element in B (or A)

ER model

your turn

cardinality of relationship is problem-specific:

problem

Every student is assigned one lecturer to advise him/her.
Every lecturer only advises at most one student.



ER model

your turn

cardinality of relationship is problem-specific:

problem

Every student is assigned one lecturer to advise him/her.
A lecturer may be an advisor to many students.



ER model

your turn

cardinality of relationship is problem-specific:

problem

Every student is assigned a team of lecturers him/her.

Every lecturer only advises at most one student.



ER model

your turn

cardinality of relationship is problem-specific:

problem

Every student is assigned a team of lecturers him/her.
A lecturer may be an advisor to many students.



ER model



relationship constraints: total participation

notation for total entity set participation:

problem

Every student **must** be assigned one lecturer to advise him/her.
A lecturer may be an advisor to many students.



total participation (indicated by double line): every student must be linked with an instructor

partial participation (indicated by single line): some instructors may not have any students they need to supervise

ER model

relationship constraints: exact cardinality

different notation for cardinality:

problem

Every student is assigned one lecturer to advise him/her.
A lecturer may be an advisor to at most 4 students.



this notation is a lot more precise

but more arduous and such detailed information might not be available or needed

ER model

your turn

try creating an ER diagram yourself!

problem

The work of a writer is to write novels. Every writer has an address and a unique name. Every novel has a unique title. A novel has only one author.

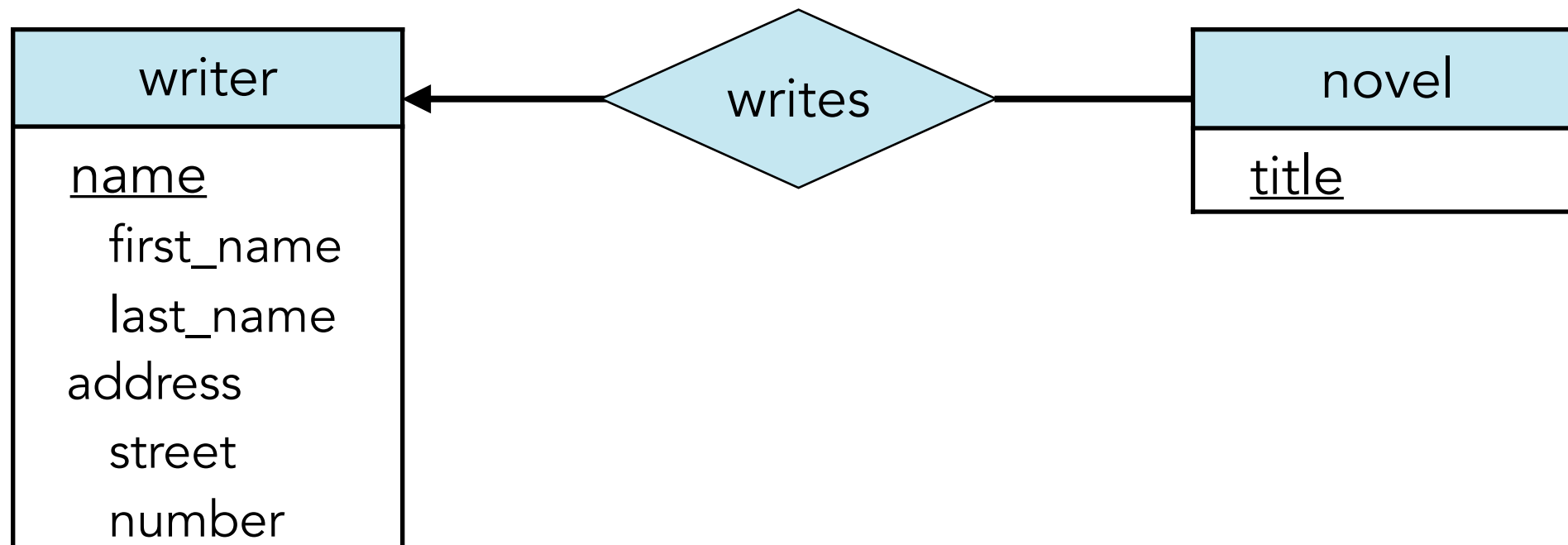
ER model

your turn

try creating an ER diagram yourself!

problem

The work of a writer is to write novels. Every writer has an address and a unique name. Every novel has a unique title. A novel has only one author.



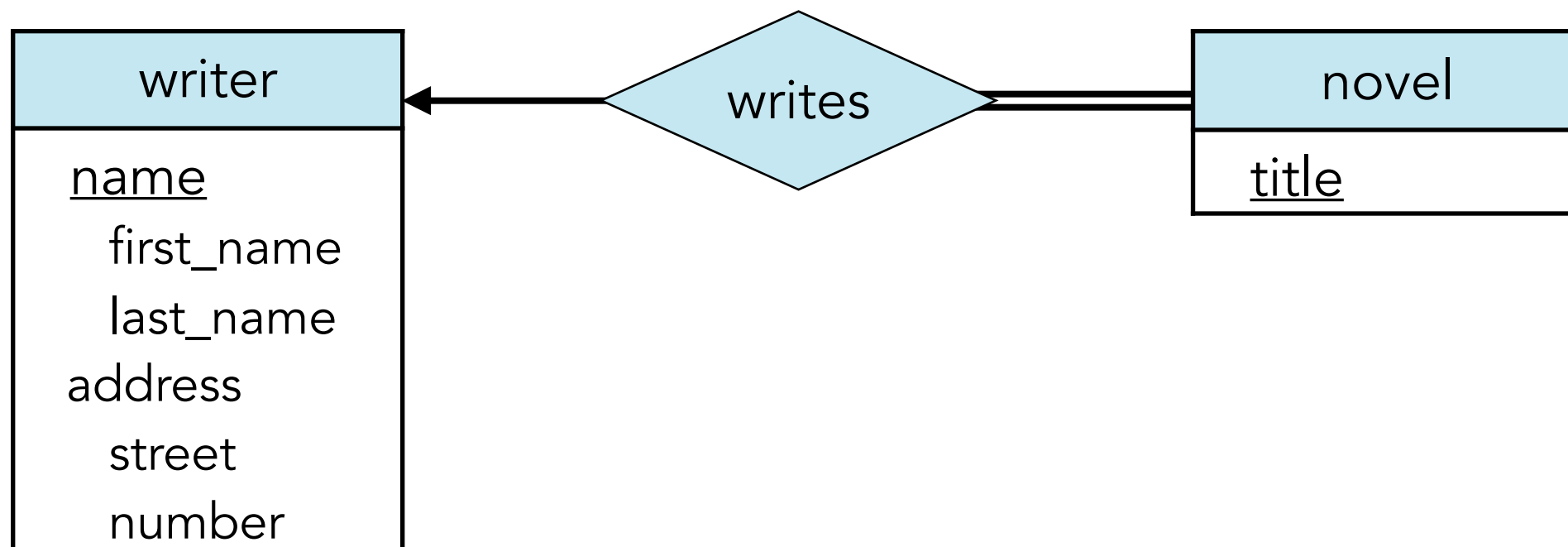
ER model

your turn

try creating an ER diagram yourself!

problem

The work of a writer is to write novels. Every writer has an address and a unique name. Every novel has a unique title. A novel has only one author.



ER model

your turn

try creating an ER diagram yourself!

problem

The work of a writer is to write novels. A novel has only one author. Novels have consumers who buy said novels.

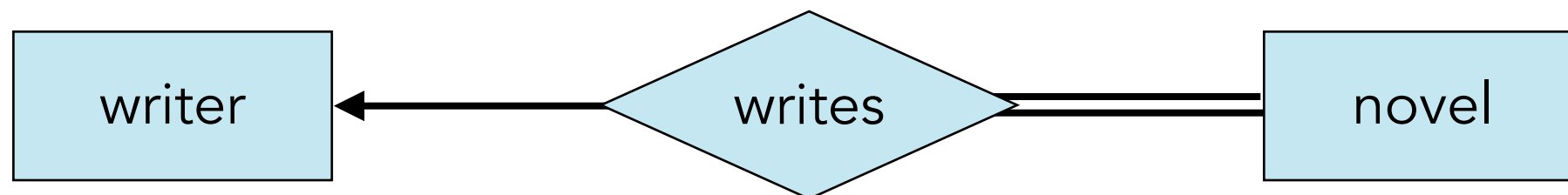
ER model

your turn

try creating an ER diagram yourself!

problem

The work of a writer is to write novels. A novel has only one author. Novels have consumers who buy said novels.



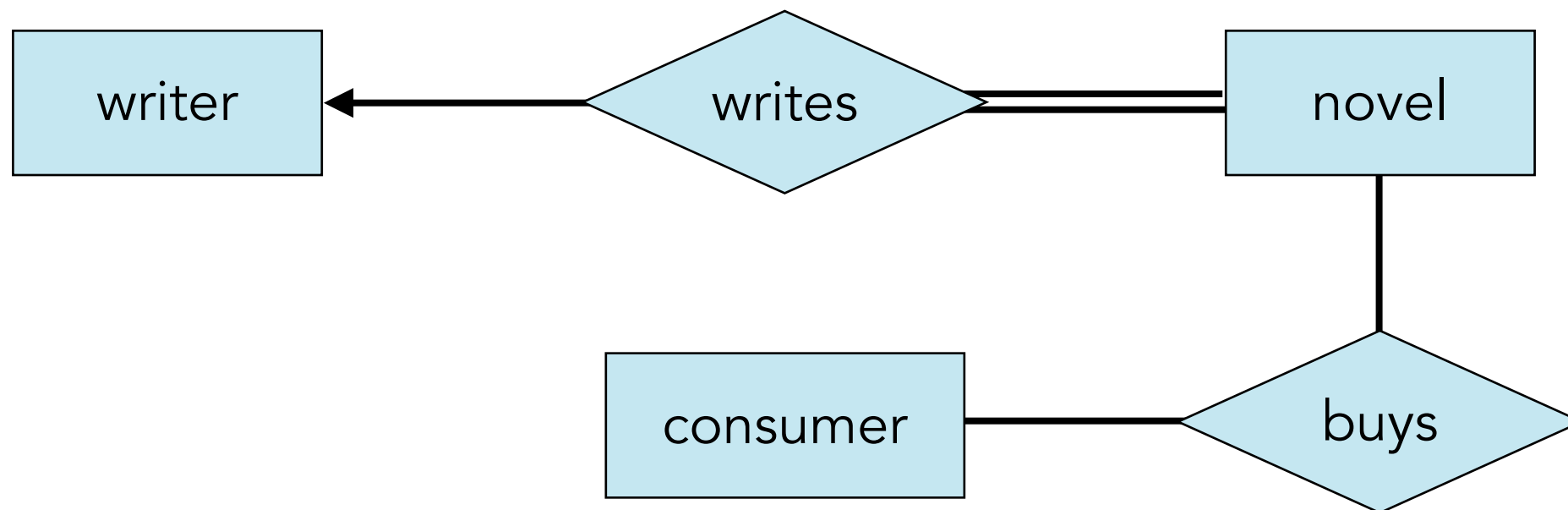
ER model

your turn

try creating an ER diagram yourself!

problem

The work of a writer is to write novels. A novel has only one author. Novels have consumers who buy said novels.



ER model

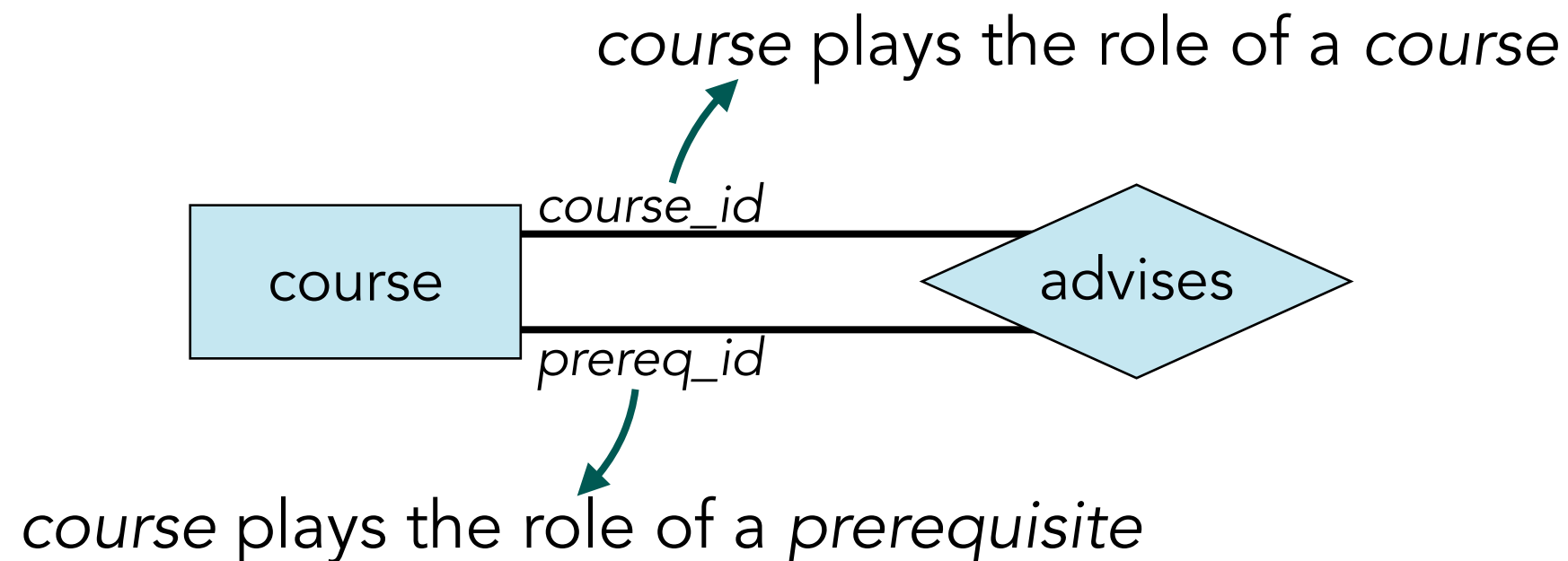
relationships: roles



entity sets need not be distinct

if the same entity set participates multiple times in a relationship, it is said to take on a **role** in that relationship

for example: a course has as a prerequisites other courses



ER model

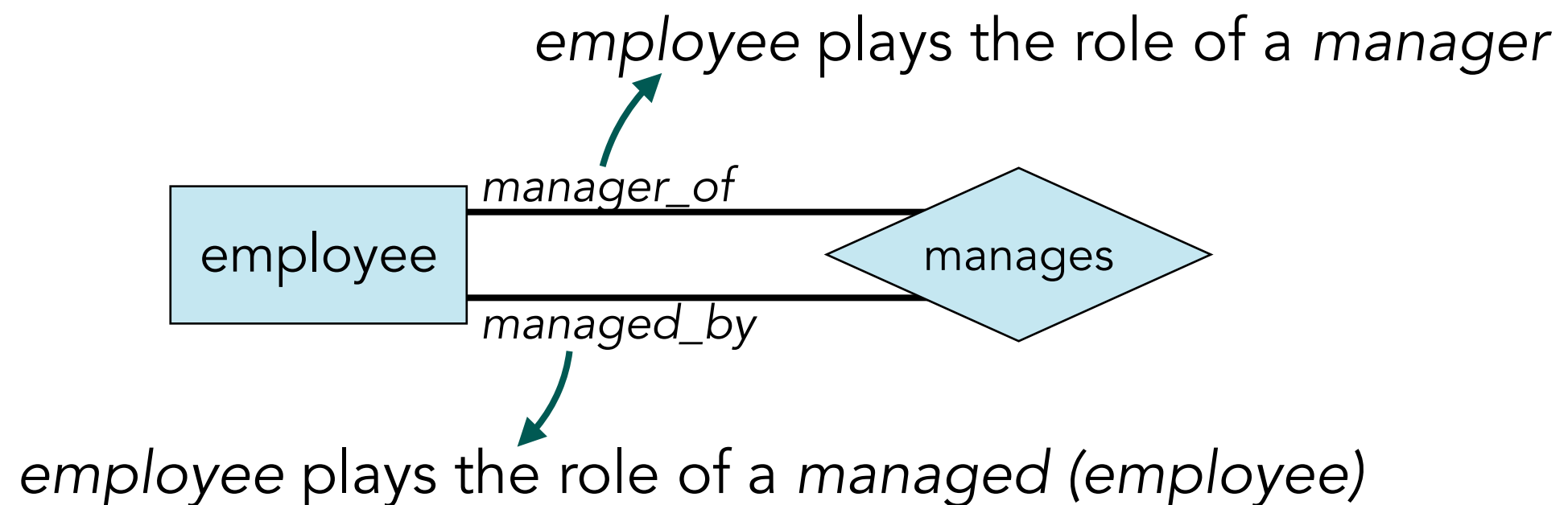
relationships: roles



entity sets need not be distinct

if the same entity set participates multiple times in a relationship, it is said to take on a **role** in that relationship

for example: an employee is managed by a manager (employee)

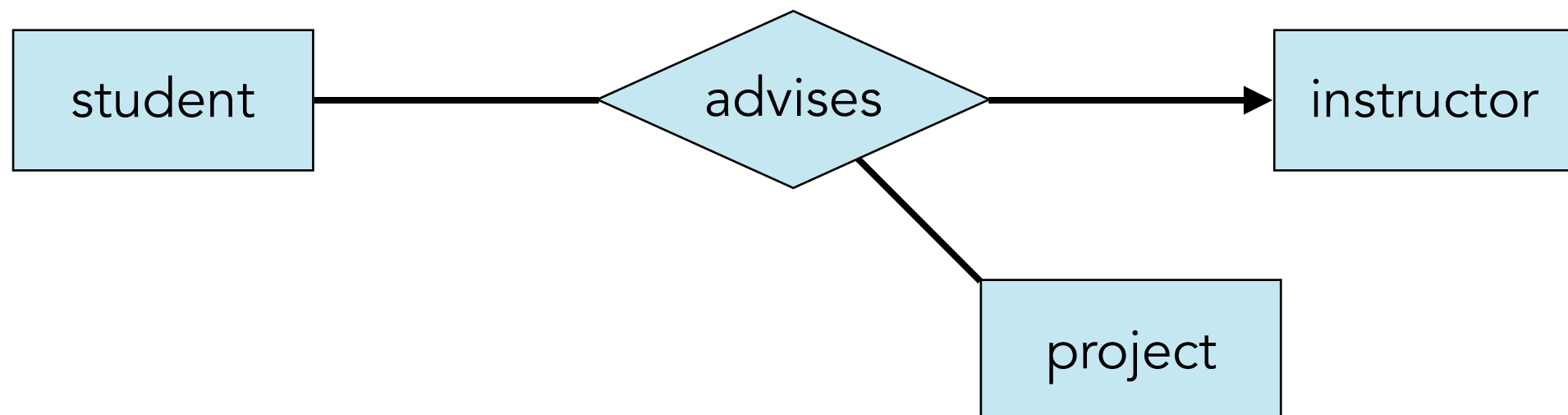


ER model

relationship constraints: ternary/multiple relationships

cardinality of ternary relationships

possible, but only allows **one arrow** to *avoid ambiguity*



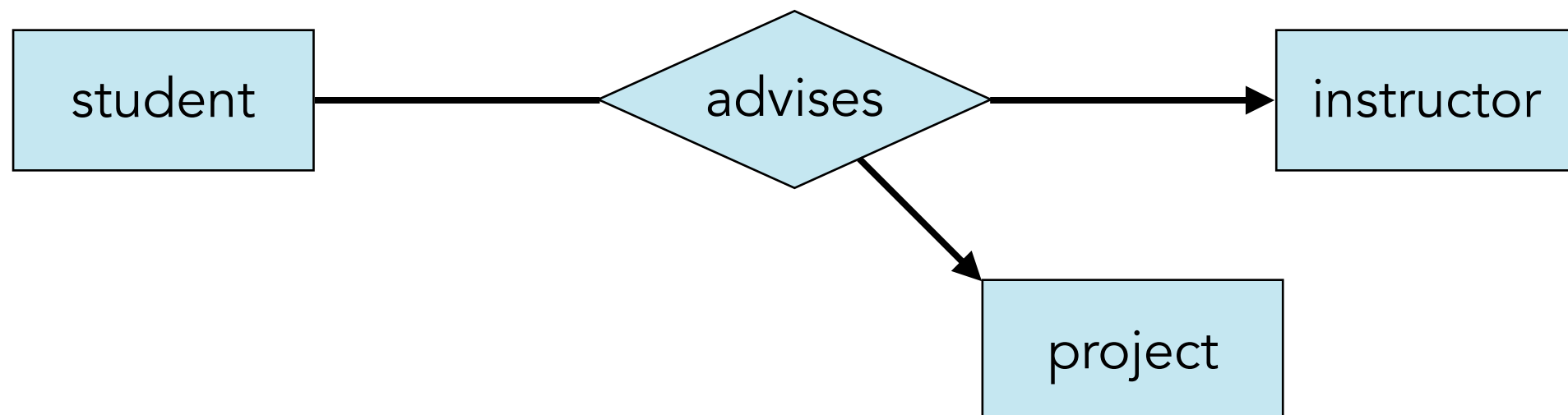
each student is assigned a project, and one project mentor

ER model

relationship constraints: ternary/multiple relationships ambiguity

cardinality of ternary relationships

ambiguous as soon as we have more than one arrow



each student is assigned one project, and one project mentor
or

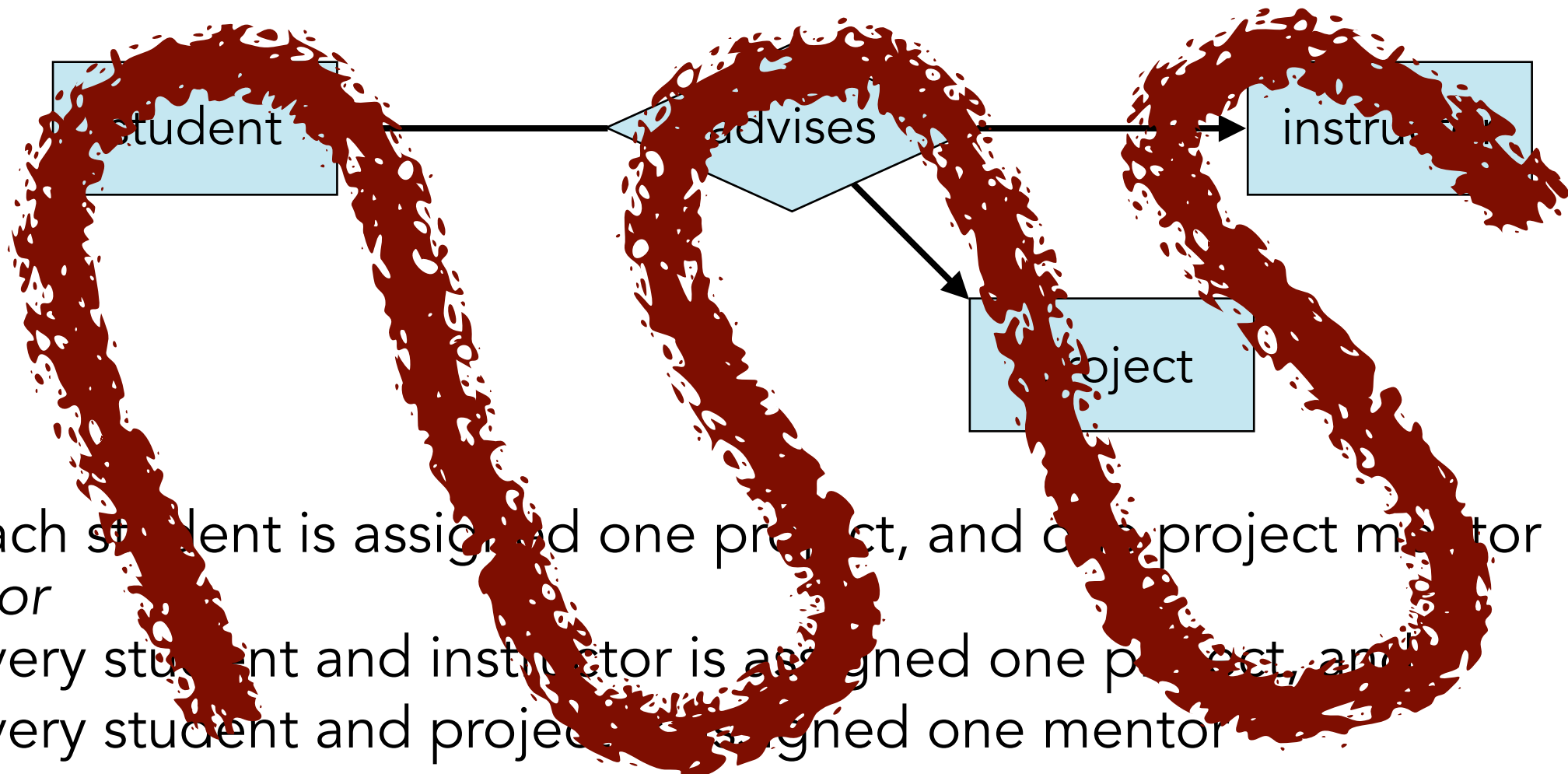
every student and instructor is assigned one project, and
every student and project is assigned one mentor

ER model

relationship constraints: ternary/multiple relationships resolution

cardinality of ternary relationships

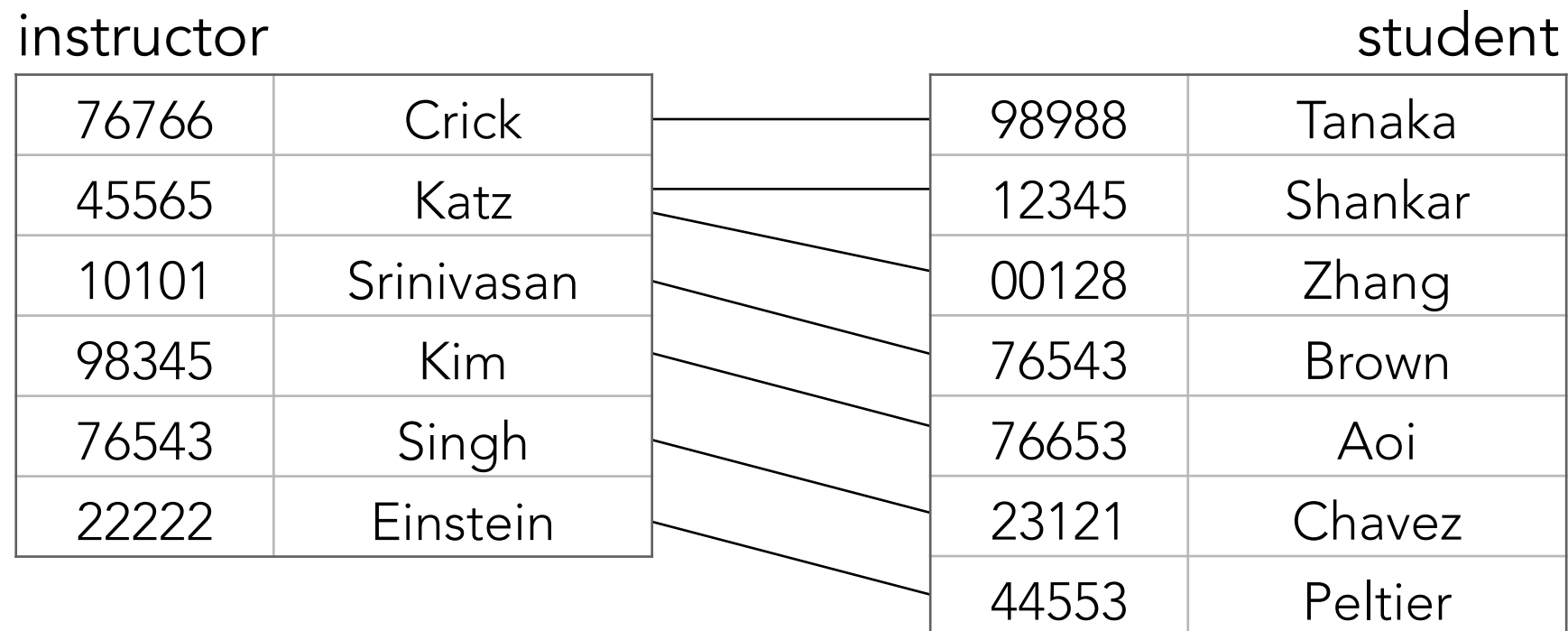
we only allow at most one arrow! no ambiguity then!



ER model

relationship attributes

more on relationships: example of *advisor* relationship



recall: it is a binary relationship, *i.e.* a relationship between two entities
such binary relationships are *by far* the most common

ER model

relationship attributes

but relationships can be more than just links!

for example, the advisor relationship could keep track of the date when the instructor took the task on him/her to become an advisor

instructor

76766	Crick
45565	Katz
10101	Srinivasan
98345	Kim
76543	Singh
22222	Einstein

3 May 2008
10 June 2007
12 June 2006
6 June 2009
30 June 2007
31 May 2007
4 May 2006

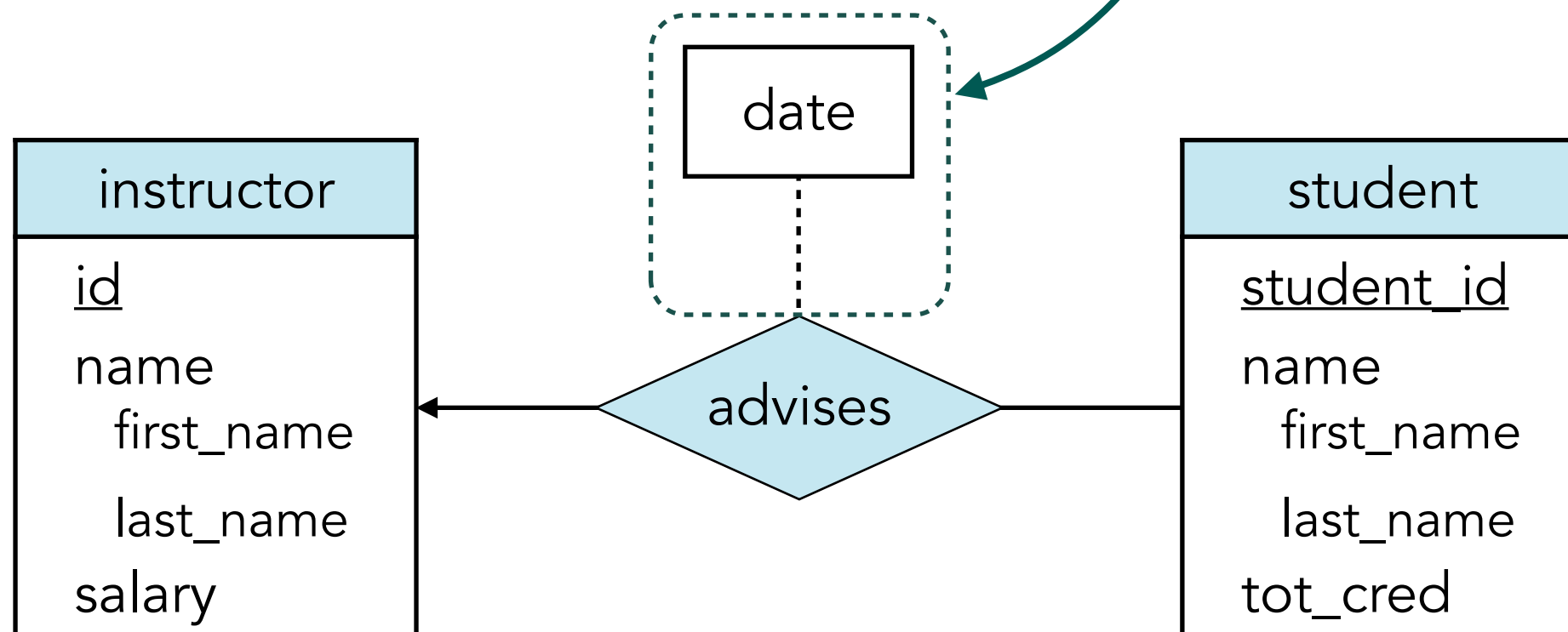
student

98988	Tanaka
12345	Shankar
00128	Zhang
76543	Brown
76653	Aoi
23121	Chavez
44553	Peltier

ER model

relationship attributes: notation

attributes of a relation can be denote using a *note*



ER model

your turn

try creating an ER diagram yourself!

problem

The work of a writer is to write novels. Sometimes there are more than one novelist with the same names.

Novels have readers who buy these novels to read them.

We would like to know what date it was purchased on.

ER model

your turn

try creating an ER diagram yourself!

The work of a writer is to write novels. Sometimes there are more than one novelist with the same names.

Novels have readers who buy these novels to read them.

We would like to know what date it was purchased on.

writer

novel

reader

ER model

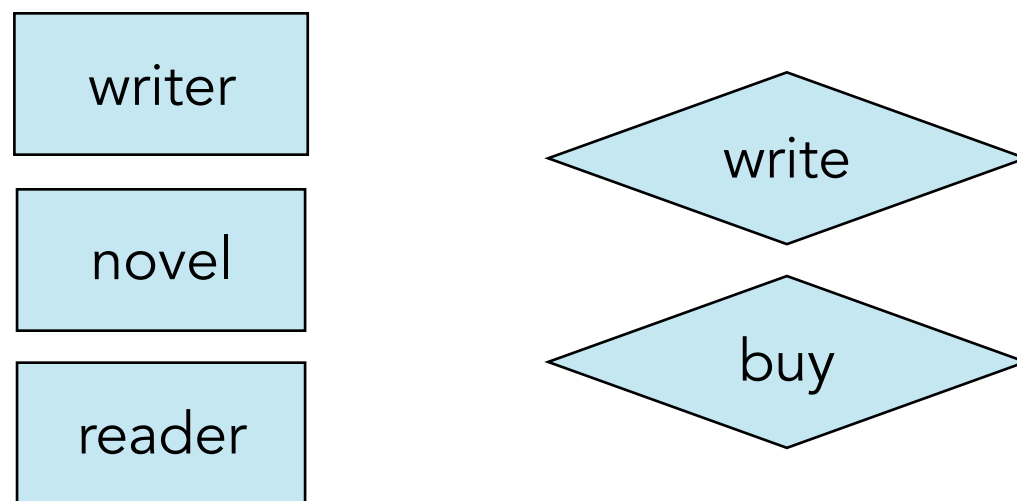
your turn

try creating an ER diagram yourself!

The work of a writer is to write novels. Sometimes there are more than one novelist with the same names.

Novels have readers who buy these novels to read them.

We would like to know what date it was purchased on.



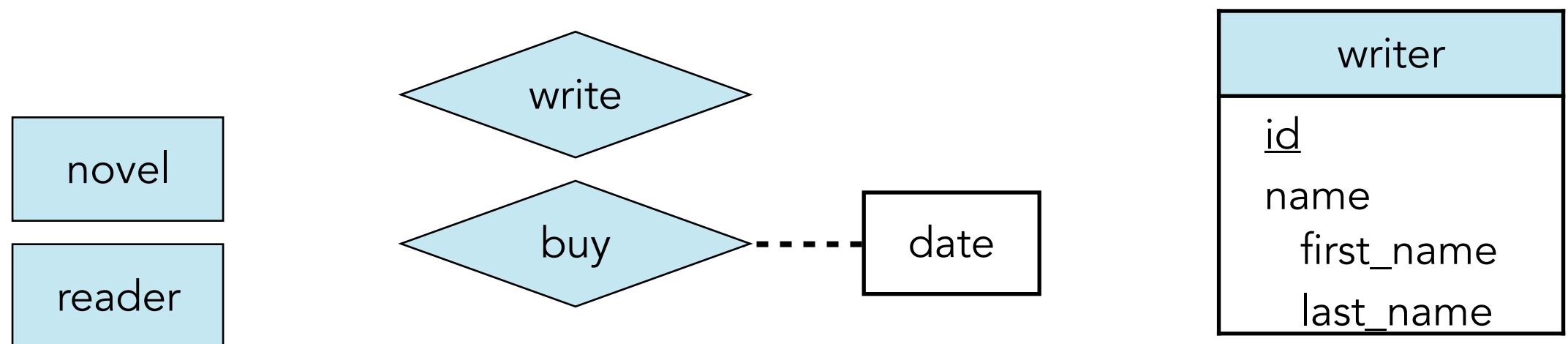
ER model

your turn

try creating an ER diagram yourself!

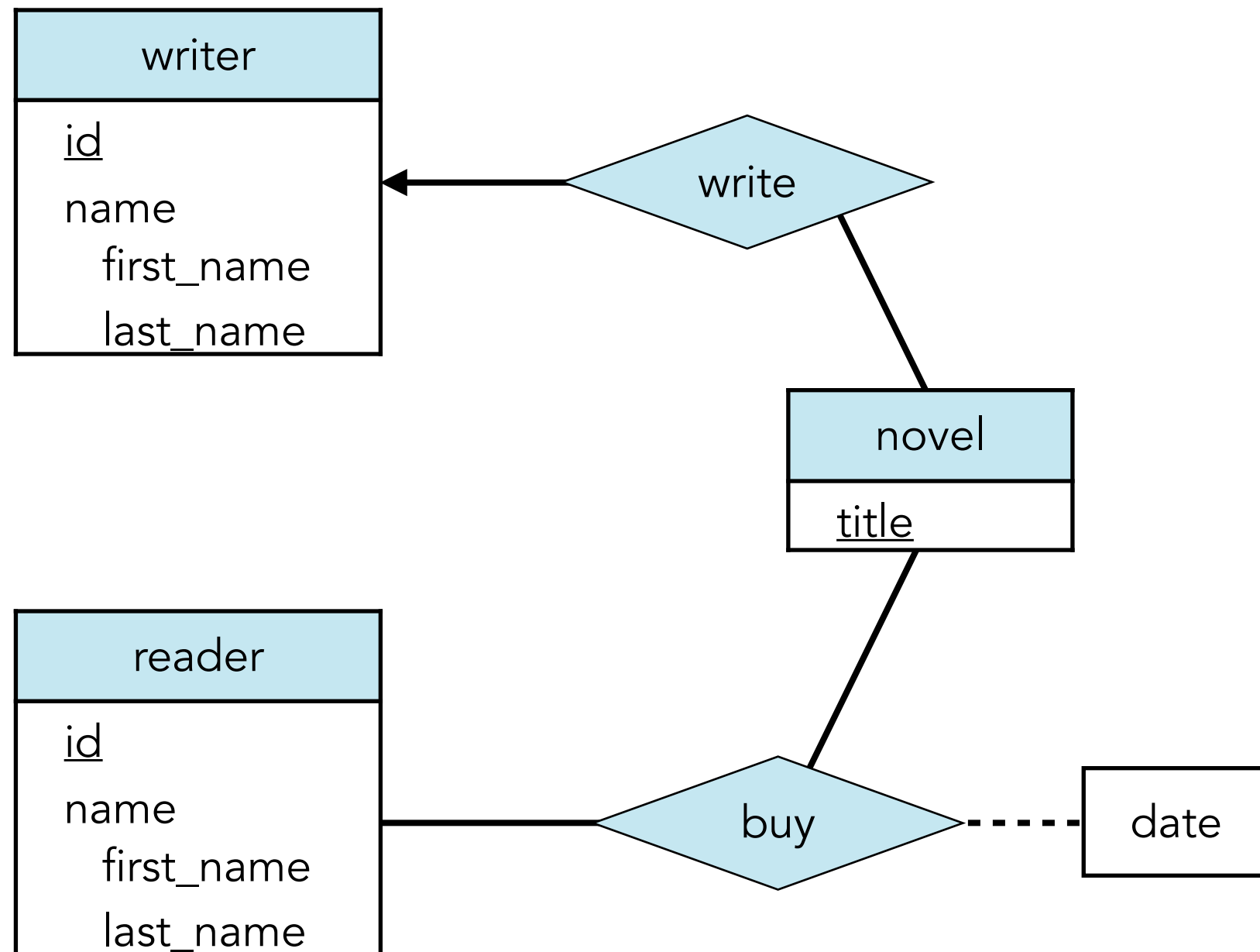
The work of a writer is to write novels. Sometimes there are ***more than one novelist with the same names.***

Novels have readers who buy these novels to read them.
We would like to know what ***date*** it was purchased on.



ER model

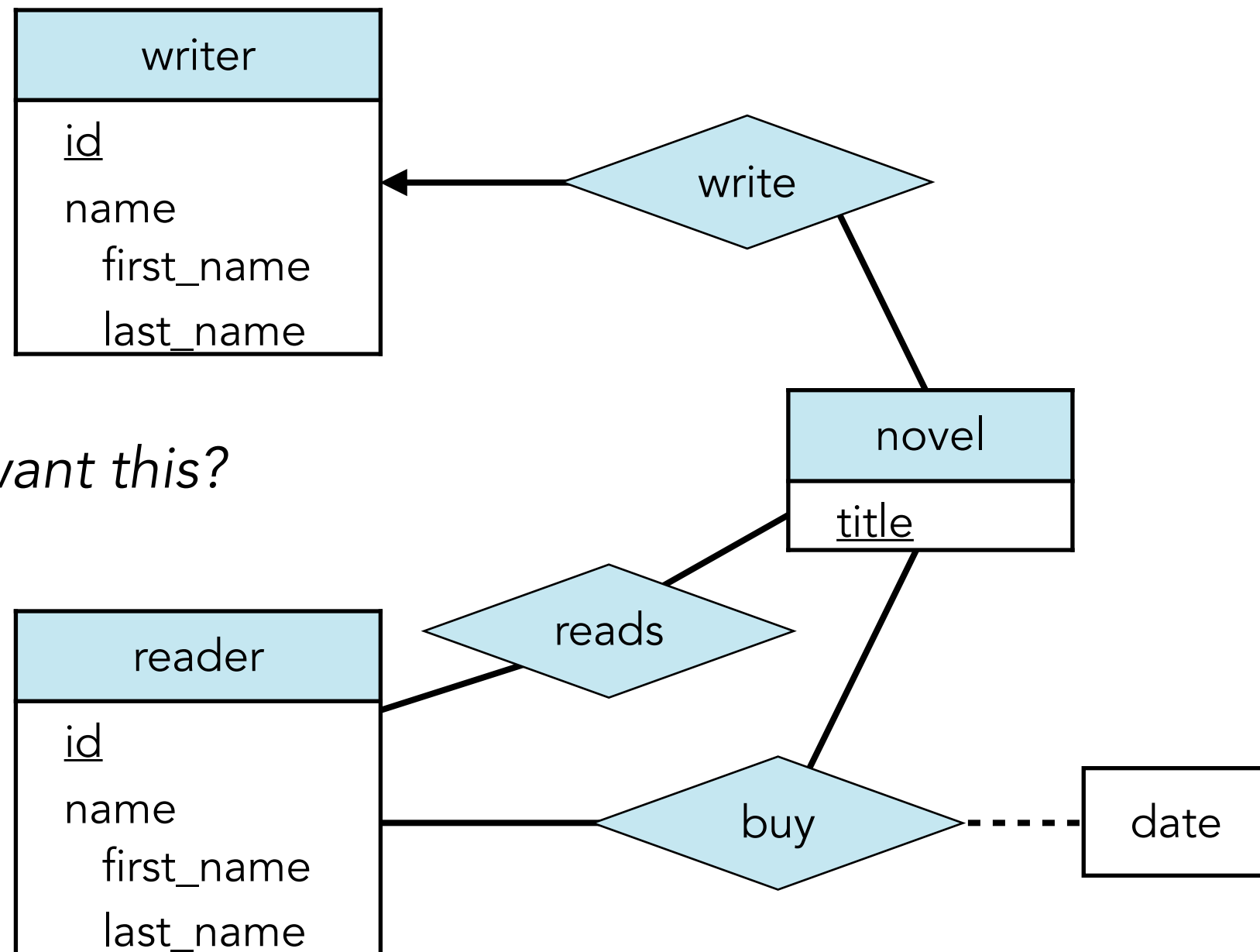
your turn



ER model

your turn

would we want this?



ER model

weak entity sets

one special kind of entity set ...

a *weak entity set* is an entity set without a **primary key**

cannot be uniquely identified
based solely on its attributes



ER model

weak entity sets

one special kind of entity set ...

a *weak entity set* is an entity set without a **primary key**

- └─ its existence depends on an identifying entity set
- └─ must relate to the identifying entity set using a:
 - total; and
 - one-to-manyrelationship set

ER model

weak entity sets

one special kind of entity set ...

a *weak entity set* is an entity set without a **primary key**

- its existence depends on an identifying entity set
- must relate to the identifying entity set using a:

- total; and
- one-to-many

relationship set

- called an identifying relationship set
and depicted using a double diamond

ER model

weak entity sets

one special kind of entity set ...

a *weak entity set* is an entity set without a **primary key**

- its existence depends on an identifying entity set
- must relate to the identifying entity set using a:

- total; and
- one-to-many

relationship set

- called an identifying relationship set
and depicted using a double diamond

the *discriminator* (or *partial key*) of a weak entity set is the set of attributes that distinguishes among all its entities

ER model

weak entity sets



one special kind of entity set ...

a *weak entity set* is an entity set without a **primary key**

→ its existence depends on an identifying entity set

→ must relate to the identifying entity set using a:

- total; and
- one-to-many

relationship set

→ called an identifying relationship set
and depicted using a double diamond

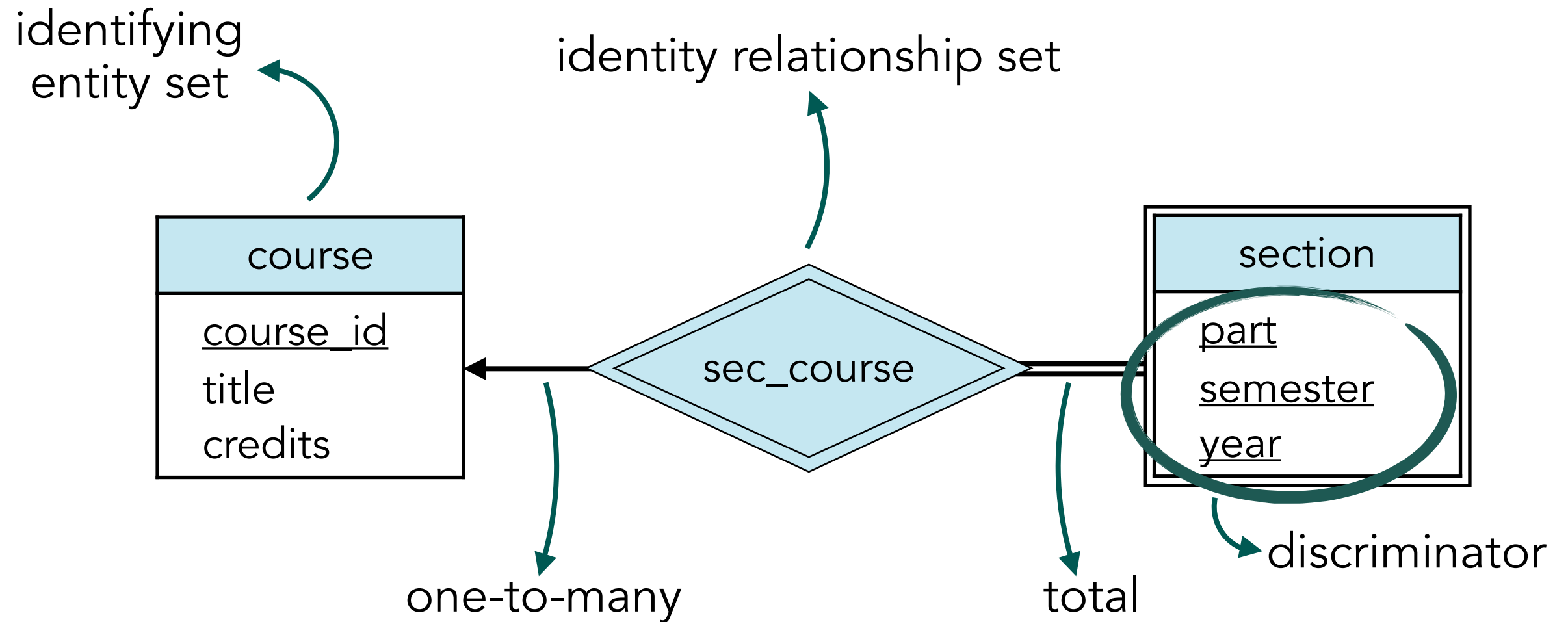
its PK is

- PK of ; and
- its discriminator

the *discriminator* (or *partial key*) of a weak entity set is the set of attributes that distinguishes among all its entities

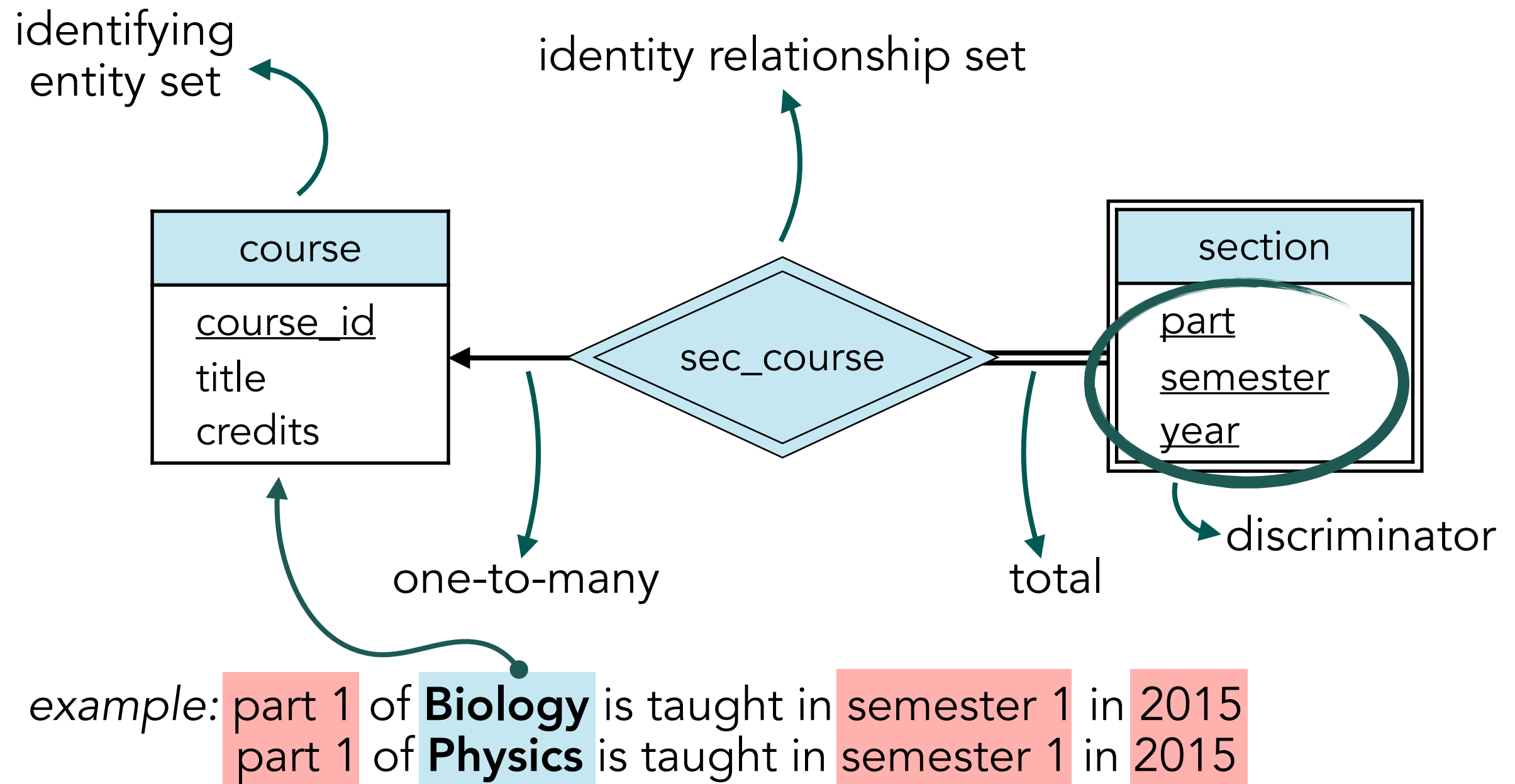
ER model

weak entity sets: notation



ER model

need for weak entity sets



ER model

are weak entity sets necessary?

some remarks:

- the *primary key* of the strong entity set is *not explicitly* stored within the weak entity set, and is *only implicitly* stored in the identifying relationship
- we can store *course_id* explicitly in *section*, which makes *section* a strong entity set. However, the relationship between *section* and *course* would then be duplicated by the implicit relationship between *section* and *course* defined by the attribute *course_id* which they both share.

remember: we try to avoid duplicates, so use weak entity set

ER model

your turn

try creating an ER diagram yourself!

problem

An employer is working for a property maintenance company and is responsible for a number of different buildings. Every employer is assigned a unique ID. We also keep track of the address of an employer, their DOB, sex, and age. Every company is identified by its name. Other properties include the insurance number of the company and the total number of employees. Buildings are identified by a unique name. We also keep track of the access code. Rooms in a building are identified by a room number. We also keep track of rooms' capacity.

ER model

your turn

try creating an ER diagram yourself!

An employer is working for a property maintenance company and is responsible for a number of different buildings. Every employer is assigned a unique ID. We also keep track of the address of an employer, their DOB, sex, and age. Every company is identified by its name. Other properties include the insurance number of the company and the total number of employees. Buildings are identified by a unique name. We also keep track of the access code. Rooms in a building are identified by a room number. We also keep track of rooms' capacity.

ER model

your turn

try creating an ER diagram yourself!

An employer is working for a property maintenance company and is responsible for a number of different buildings. Every employer is assigned a unique ID. We also keep track of the address of an employer, their DOB, sex, and age. Every company is identified by its name. Other properties include the insurance number of the company and the total number of employees. Buildings are identified by a unique name. We also keep track of the access code. Rooms in a building are identified by a room number. We also keep track of rooms' capacity.

ER model

your turn

try creating an ER diagram yourself!

An employer is working for a property maintenance company and is responsible for a number of different buildings. Every employer is assigned a **unique ID**. We also keep track of the **address** of an employer, their **DOB**, **sex**, and **age**. Every company is identified by its name. Other properties include the insurance number of the company and the total number of employees. Buildings are identified by a unique name. We also keep track of the access code. Rooms in a building are identified by a room number. We also keep track of rooms' capacity.

ER model

your turn

try creating an ER diagram yourself!

An employer is working for a property maintenance company and is responsible for a number of different buildings. Every employer is assigned a unique ID. We also keep track of the address of an employer, their DOB, sex, and age. Every company is **identified by** its **name**. Other properties include the **insurance number** of the company and the **total number of employees**. Buildings are identified by a unique name. We also keep track of the access code. Rooms in a building are identified by a room number. We also keep track of rooms' capacity.

ER model

your turn

try creating an ER diagram yourself!

An employer is working for a property maintenance company and is responsible for a number of different buildings. Every employer is assigned a unique ID. We also keep track of the address of an employer, their DOB, sex, and age. Every company is identified by its name. Other properties include the insurance number of the company and the total number of employees. Buildings are identified by a **unique name**. We also keep track of the **access code**. Rooms in a building are identified by a room number. We also keep track of rooms' capacity.

ER model

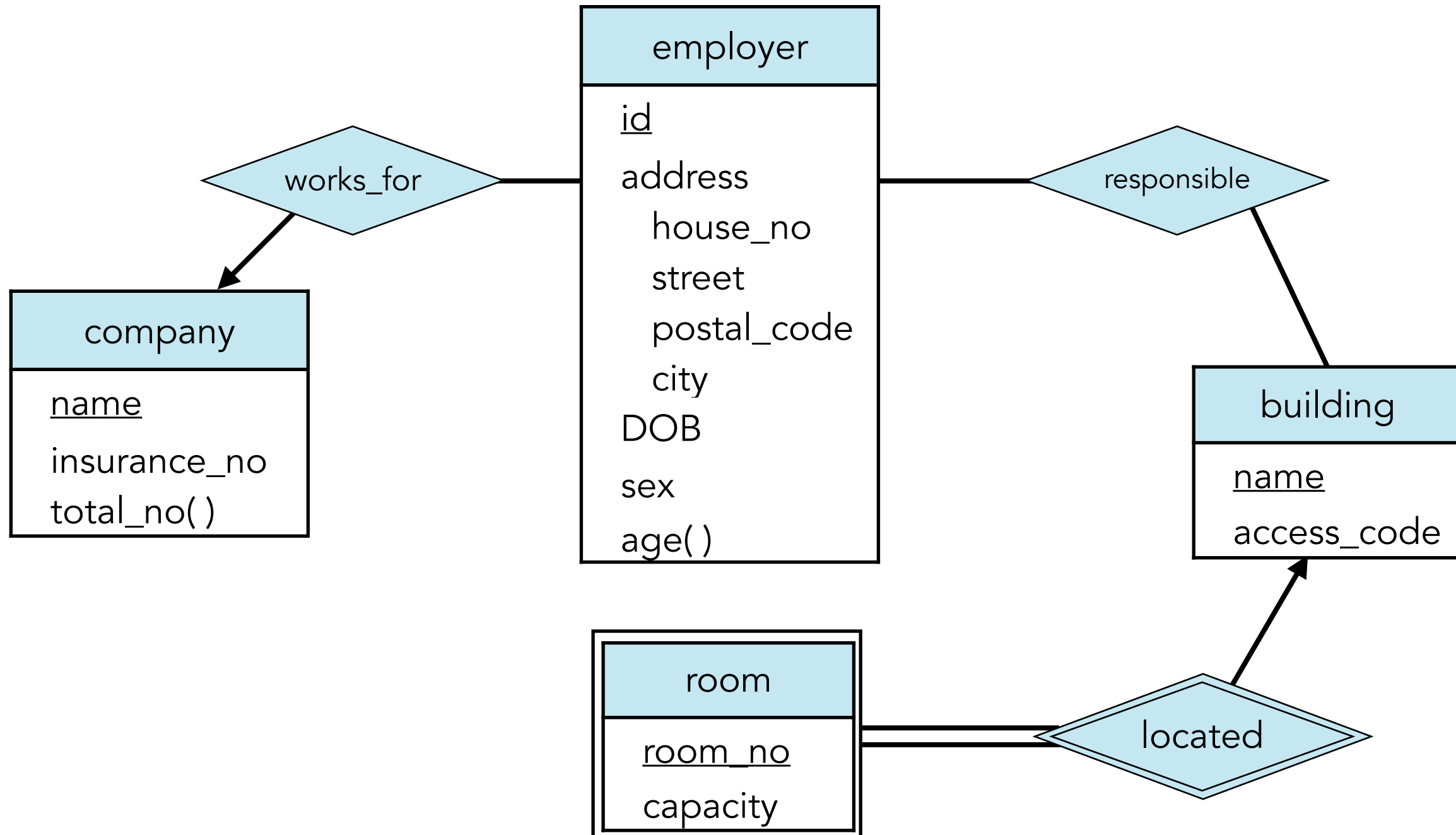
your turn

try creating an ER diagram yourself!

An employer is working for a property maintenance company and is responsible for a number of different buildings. Every employer is assigned a unique ID. We also keep track of the address of an employer, their DOB, sex, and age. Every company is identified by its name. Other properties include the insurance number of the company and the total number of employees. Buildings are identified by a unique name. We also keep track of the access code. Rooms in a building are identified by a **room number**. We also keep track of rooms' **capacity**.

ER model

your turn

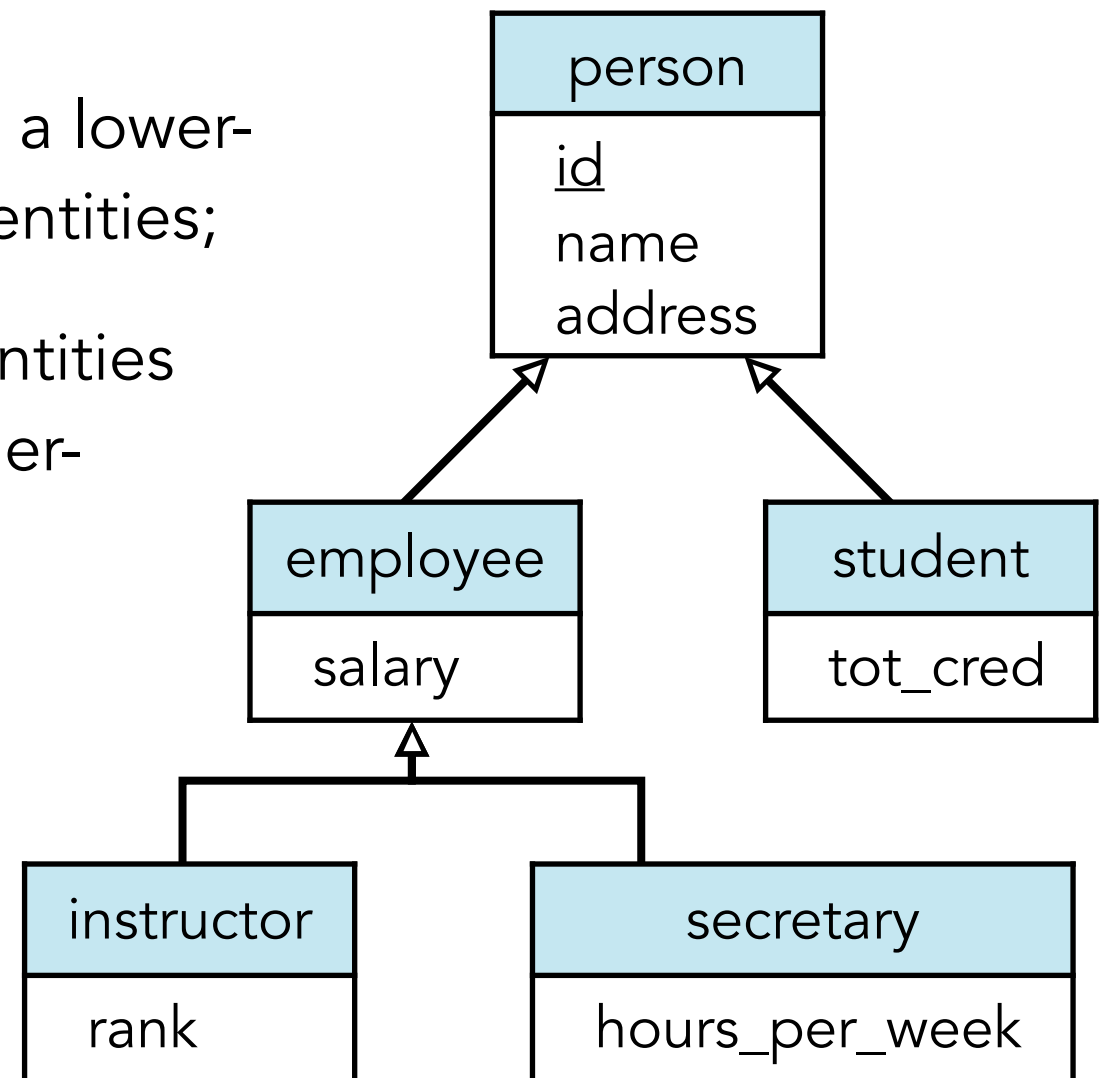


ER model

specialisation/generalisation in ER models

specialisation in ER models

- *top-down design*: we specialise to a lower-level entity distinctive of all other entities;
- *attribute inheritance*: lower-level entities have the attributes of all their higher-level entities, as well as their own
- lower-level entities can participate in their own relationships that do not apply to higher-level entities



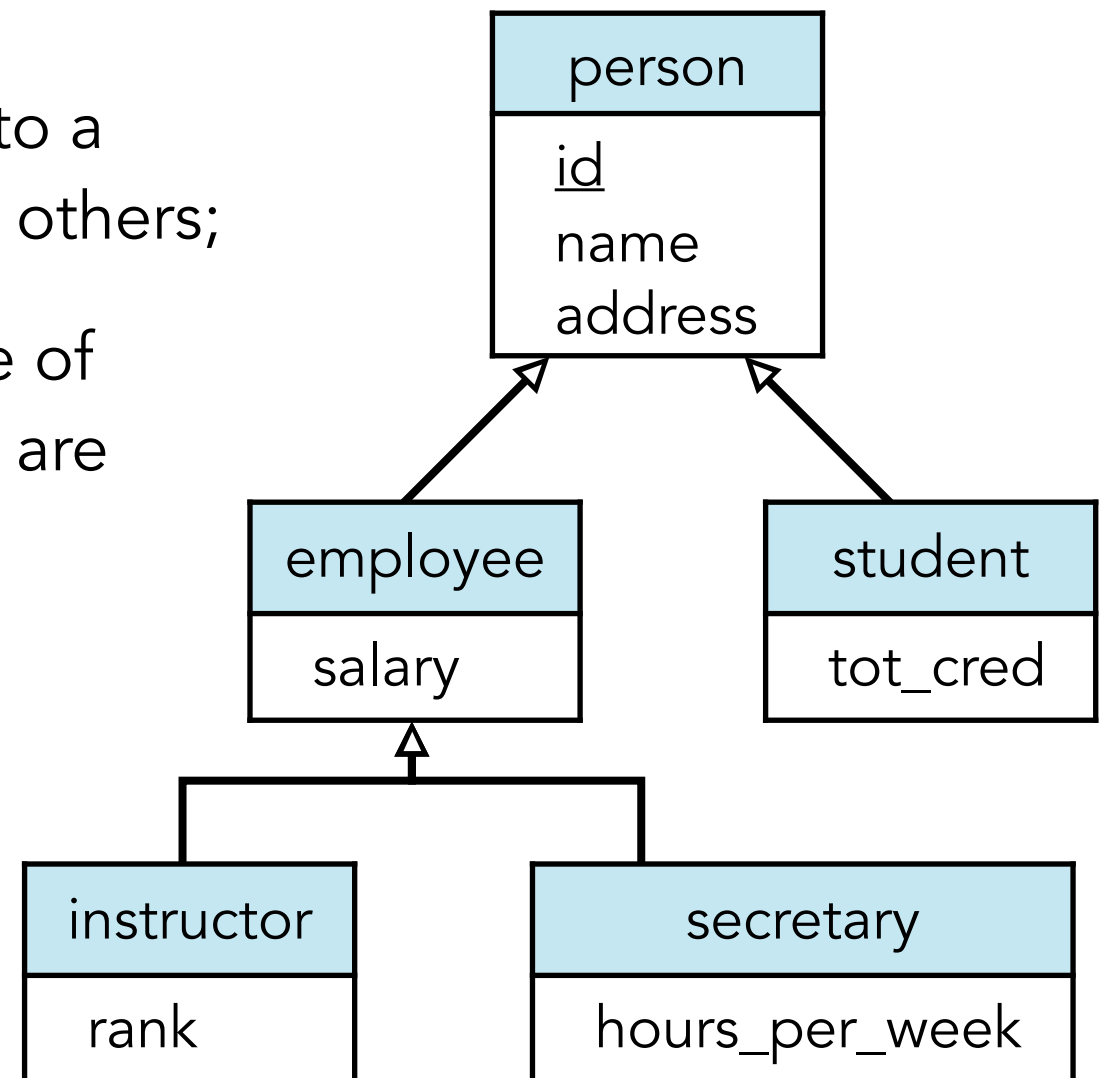
denoted by directed arrow: *is-a* relation

ER model

specialisation/generalisation in ER models

generalisations in ER models

- *bottom-up design*: we generalise to a higher-level entity distinctive of all others;
- generalisation is simply the inverse of specialisation and vice versa: both are represented in the same way;
- similar to their shared representation, the terms generalisation and specialisation are typically used interchangeably

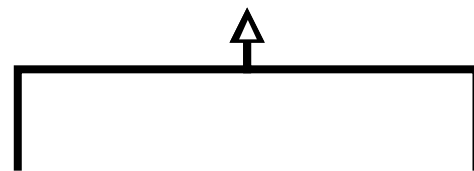


ER model

specialisation/generalisation constraints

some complications/added constraints:

- possible for an entity to have multiple specialisations
e.g. an employee is a lecturer/secretary, and an employee is permanent/temporary. Each employee is then a member of either permanent or temporary, **and** either lecturer or secretary
- constraint on whether or not an entity can belong to more than one lower-level entity set within a single generalisation:
 - *disjoint*: entity can belong to *only one* lower-level entity set
 - *overlapping*: entity can belong to *multiple* lower-level entity sets

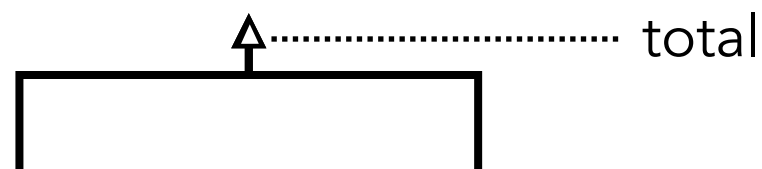


ER model

specialisation/generalisation constraints

some complications/added constraints (cont.):

- there can be constraints on which employees can be a member of a lower-level entity
 - can be condition-defined;
e.g. all citizens over 65 are a senior_citizen entity
 - or user-defined.
- completeness constraint as to whether an entity in the higher-level entity set needs to belong to one of the lower-entity sets
 - *total*: entity must belong to one of the lower-entity sets
 - *partial*: entity need not belong to one of the lower-entity sets



ER model



general algorithm for developing ER model

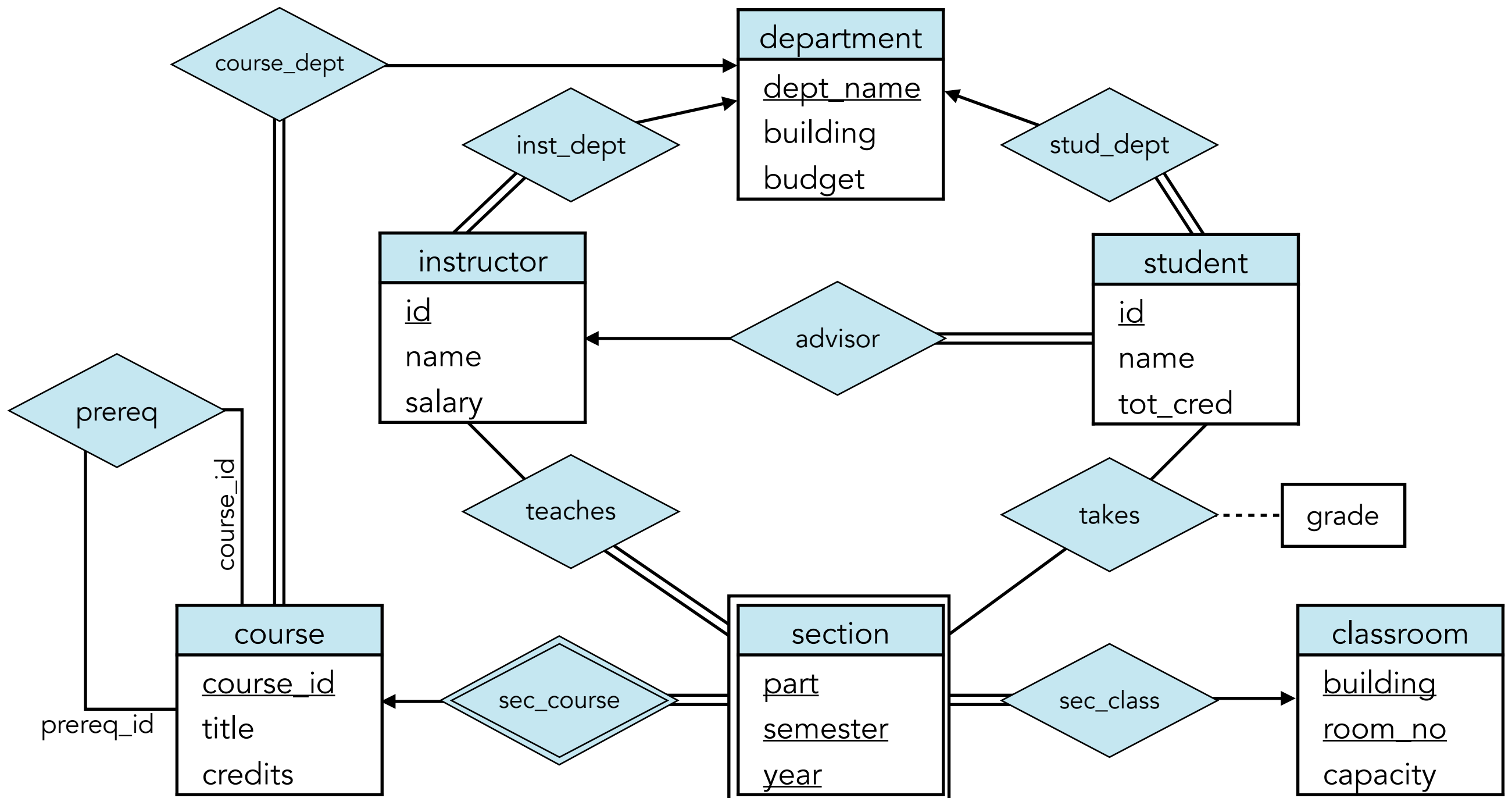
general approach:

- identify the entity sets;
- identify the relationship sets, and the entity sets that are participating in each relationship;
- identify the attributes;
- define dependencies, constraints, keys, etc.;
- draw the diagram.

important: for each step/choice, make note of your assumptions!
Some *will* become outdated as you explore the domain.
These notes make it easy to backtrack and apply changes.

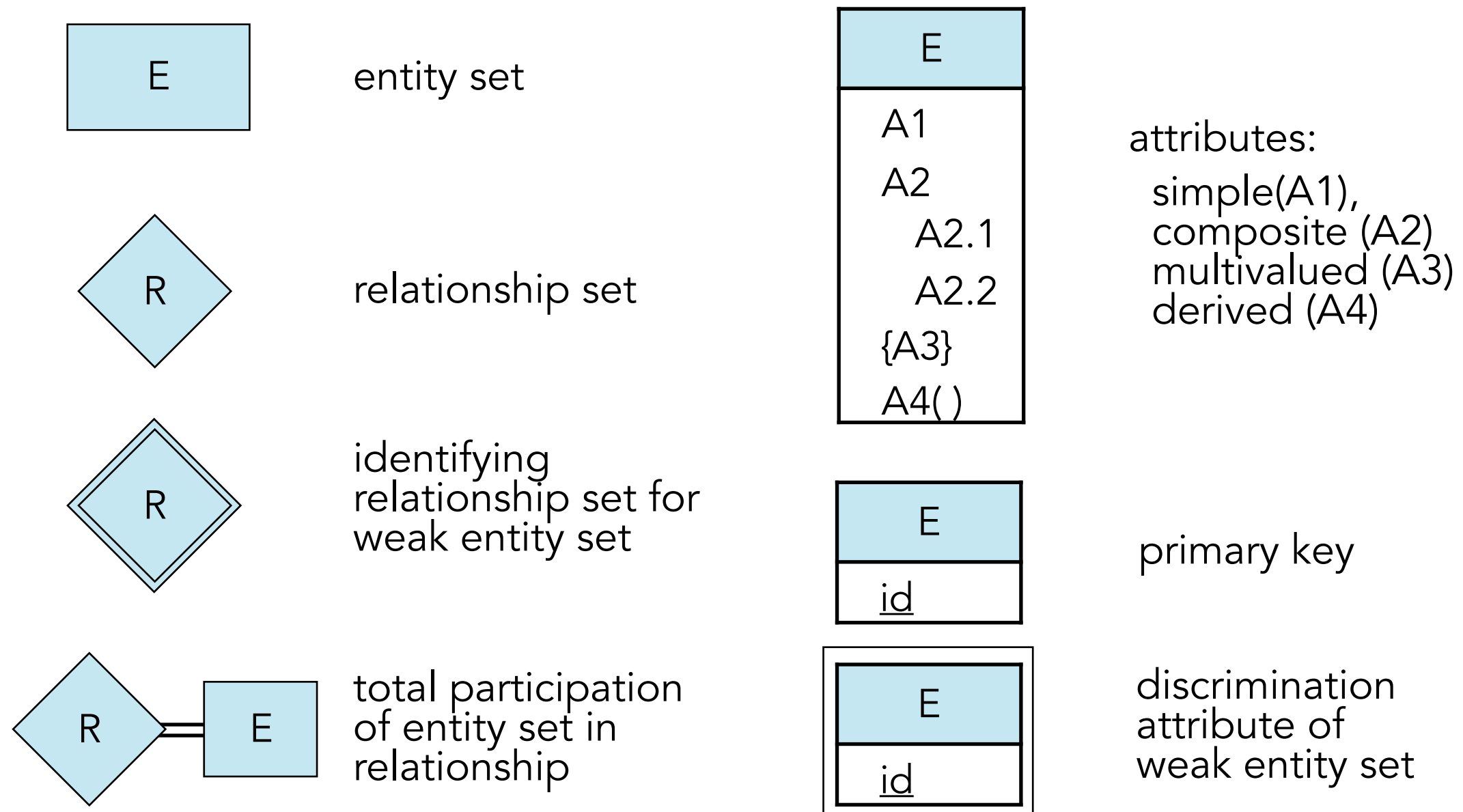
ER model

full model for the university setting



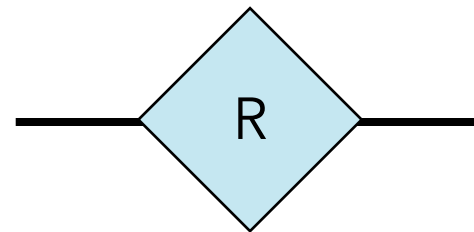
ER model

summary

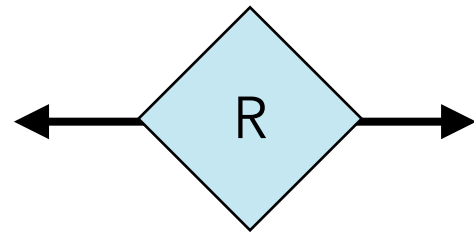


ER model

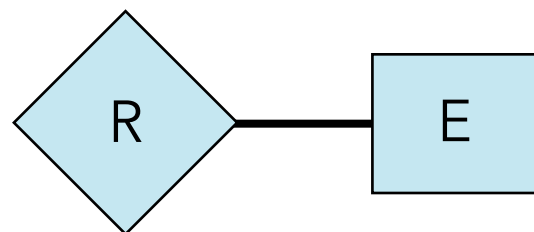
summary (cont.)



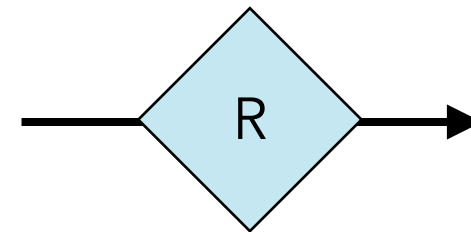
many-to-many
relationship



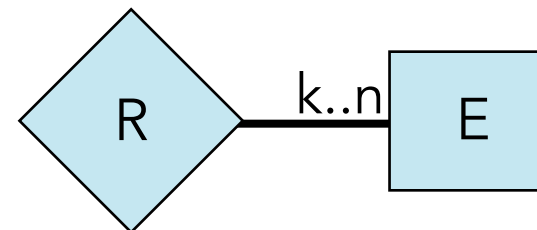
one-to-one
relationship



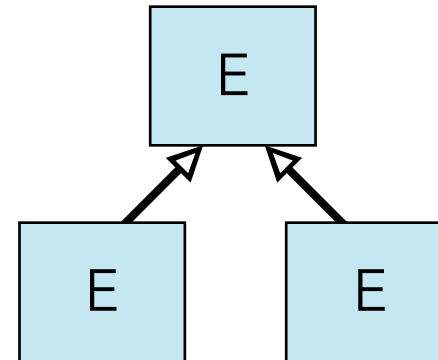
role indicator



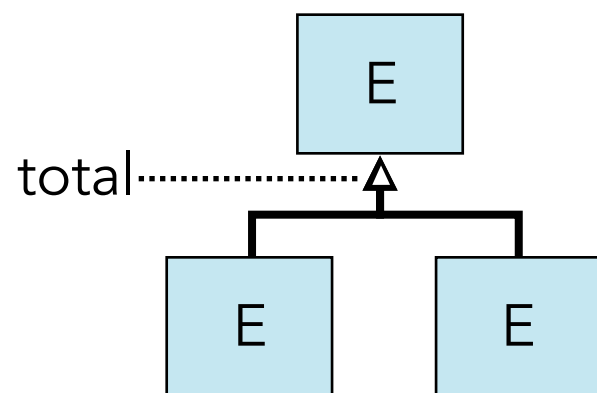
many-to-one
relationship



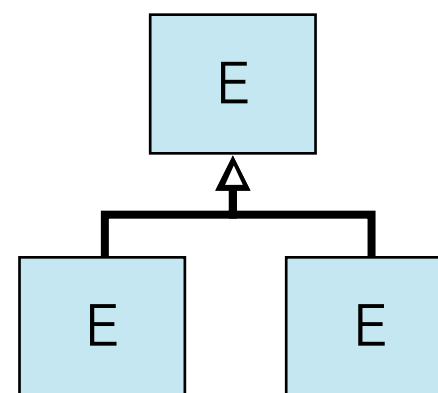
cardinality
limits



is-a (or ISA):
generalisation
or specialisation



total (disjoint)
generalisation



disjoint
generalisation