Business Intelligence

ER model

Lecture outline

- Database design
- The Entity-Relationship (ER) model
- Relationships in the ER model
- Attributes and identifiers in the ER model
- Hierarchies
- Quality of a conceptual schema

Database design

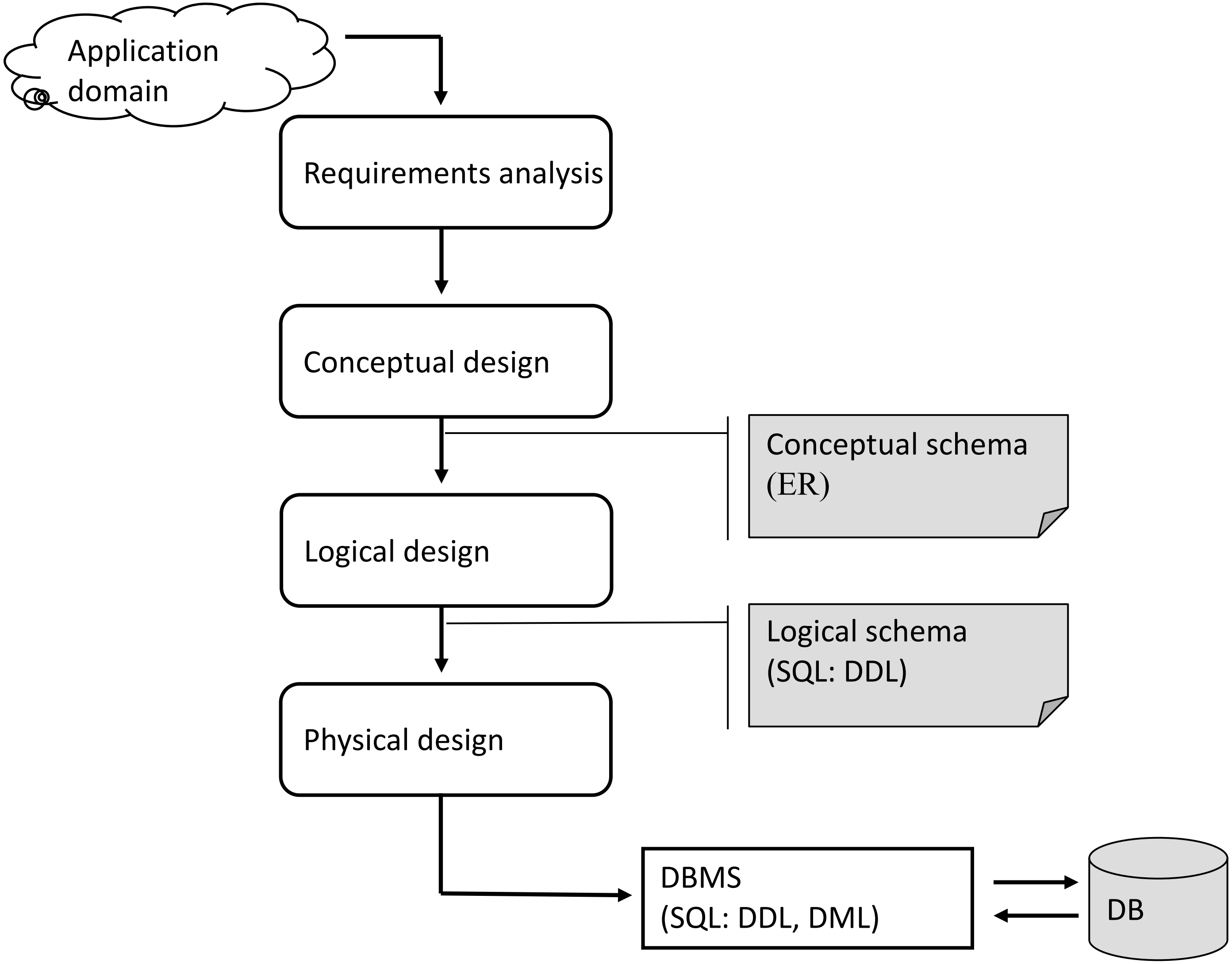
- Part of the life cycle of a database, comprising:
    - Requirements analysis
    - Schema and application design
    - Implementation
    - Validation and testing
    - Running

- Here we focus on schema design

Design phases

- Conceptual design
- Logical design
- Physical design

Design process



Conceptual design

- Goal: translating requirements analysis into a formal description, independent of the DBMS
- This makes use of a conceptual schema, by means of a conceptual data model

Logical design

- Goal: translating the conceptual schema into a logical schema among, e.g.:
  - hierarchical
  - network
  - relational
  - object-oriented
  - XML

Physical design

- Goal: producing a physical design allowing the best performance by choosing the right physical access structures

- The physical design is different on each product

Dependencies on model and DBMS

	Depends on data model	Depends on DBMS
Conceptual design	N	N
Logical design	Y	N
Physical design	Y	Y

ENTITY-RELATIONSHIP MODEL

- Due to P. P. Chen (1976)
  - Peter P. Chen: *The Entity-Relationship Model - Toward a Unified View of Data*. **ACM Trans. Database Syst.** 1(1): 9-36 (1976)
- Industrial standard since then
- It’s a graphical model

ER is a time-proof model

P. Chen

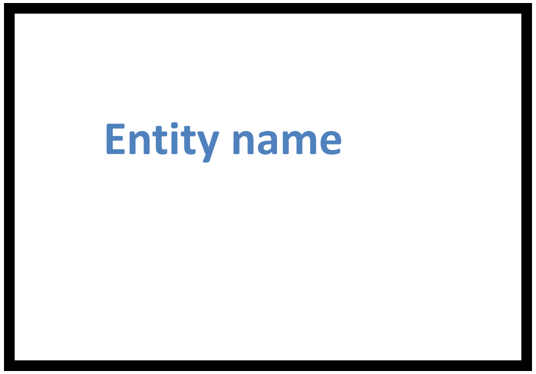
ER conference (october 2008)

ER conference (november 2016)

Main elements

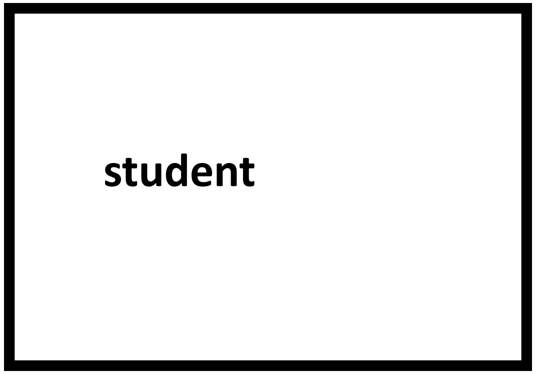
- Entity: a thing (object, person, …) that exists in its own right in the application domain. We want to store specific facts about it and identify it
- Relationship: fact describing an action or situation establishing associations between entities
- Property (or attribute): fact describing characteristics of entities and relationships

Entities



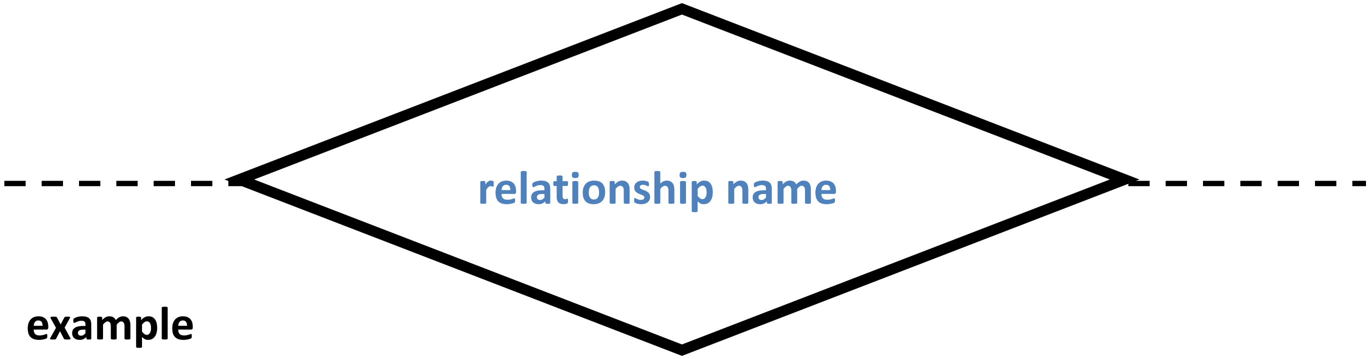
Entity name

Example:

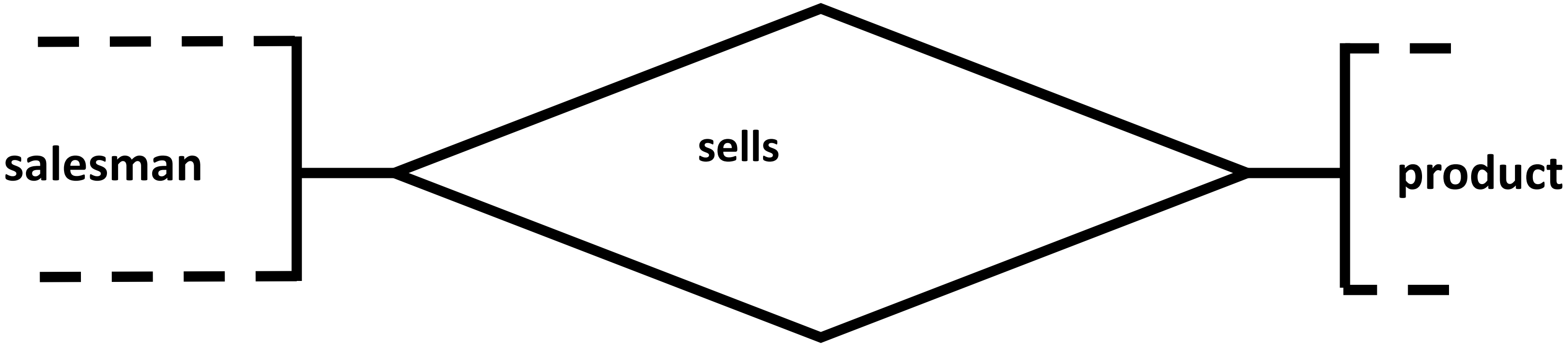


student

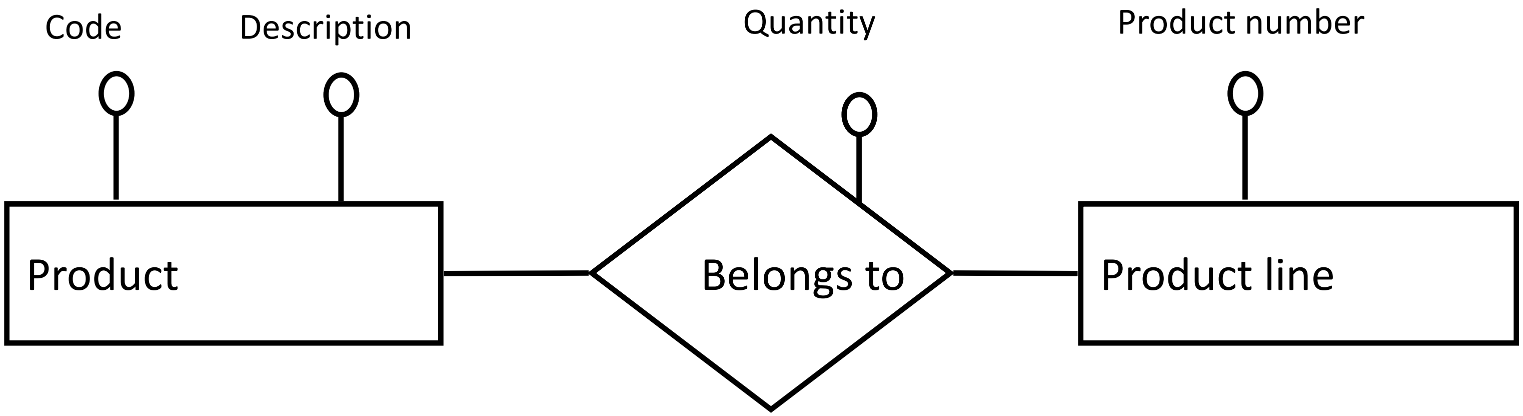
Relationships



example



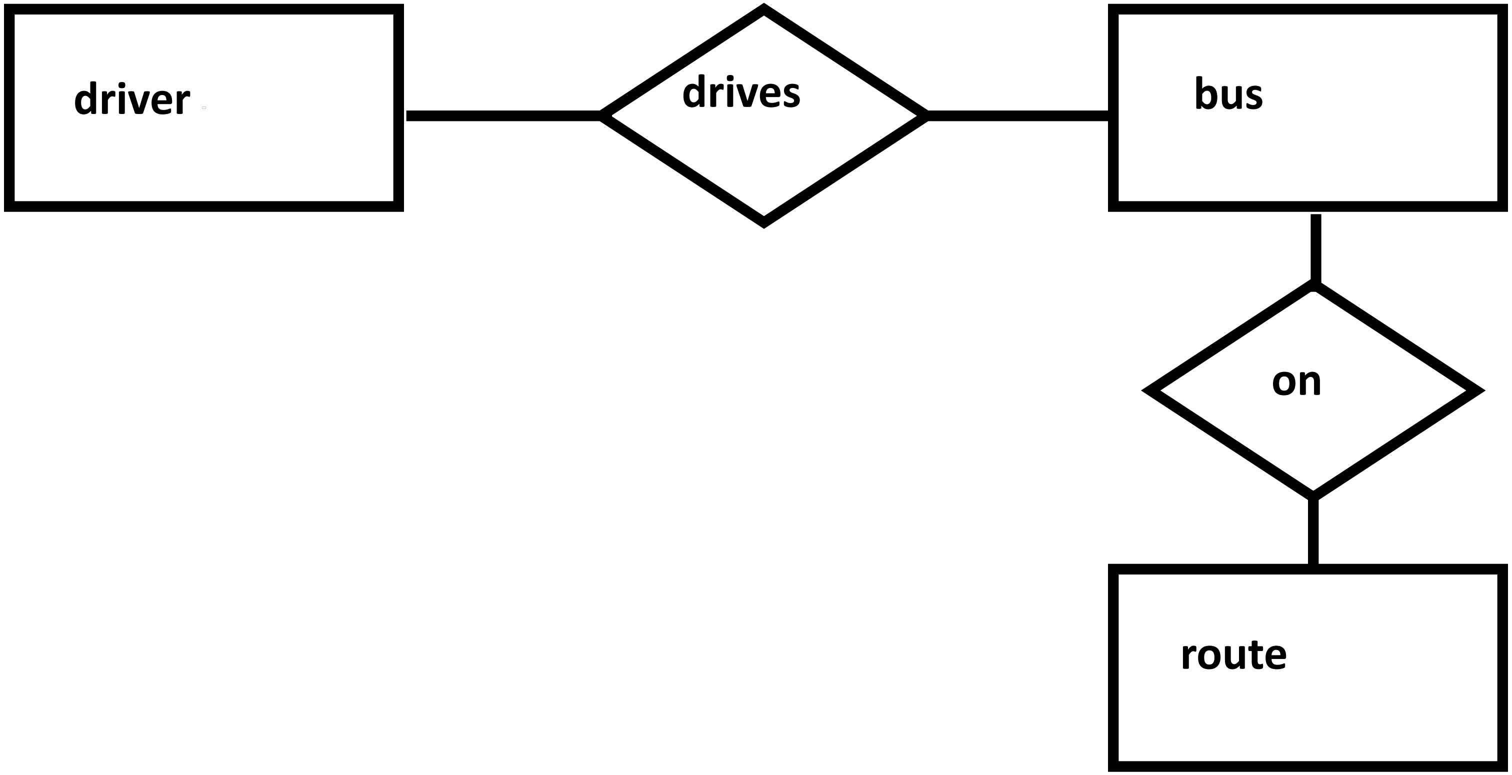
Attributes


Code
Description
Quantity
Product number
Product
Belongs to
Product line

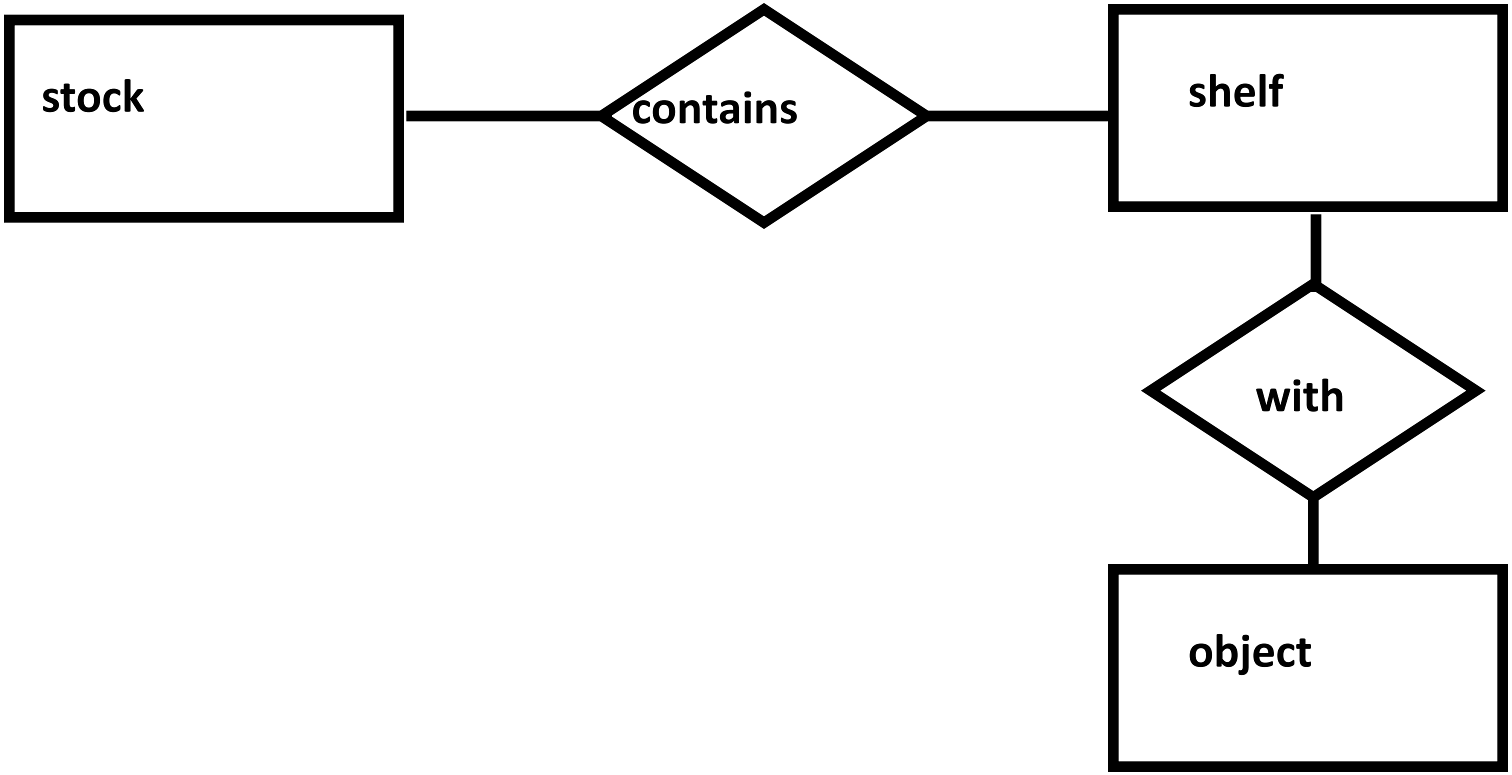
Design guidelines

- If concept significant -> entity
- If concept describable by an elementary value -> attribute
- If concept relates entities -> relationship

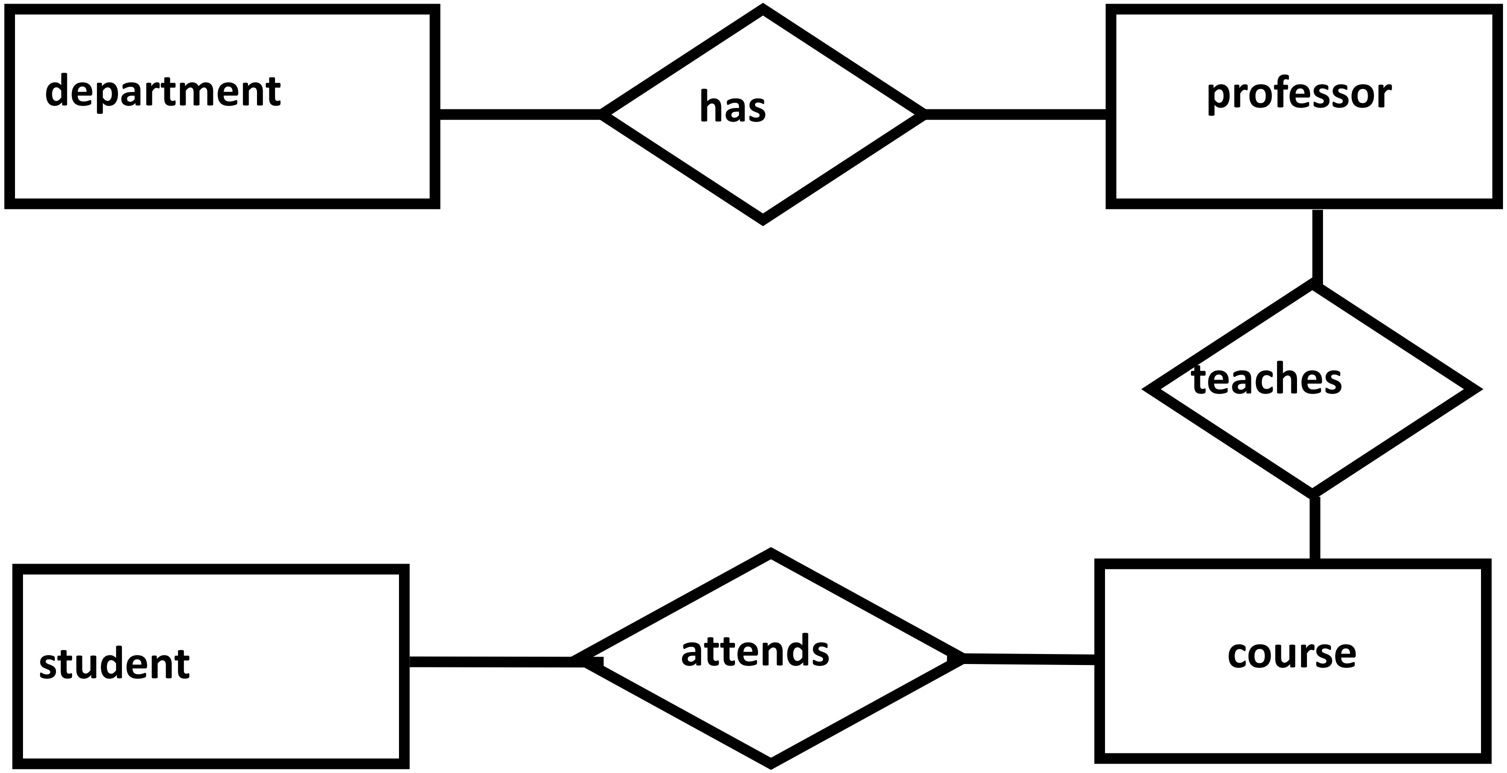
Example: travels



Example: stock



Example: university



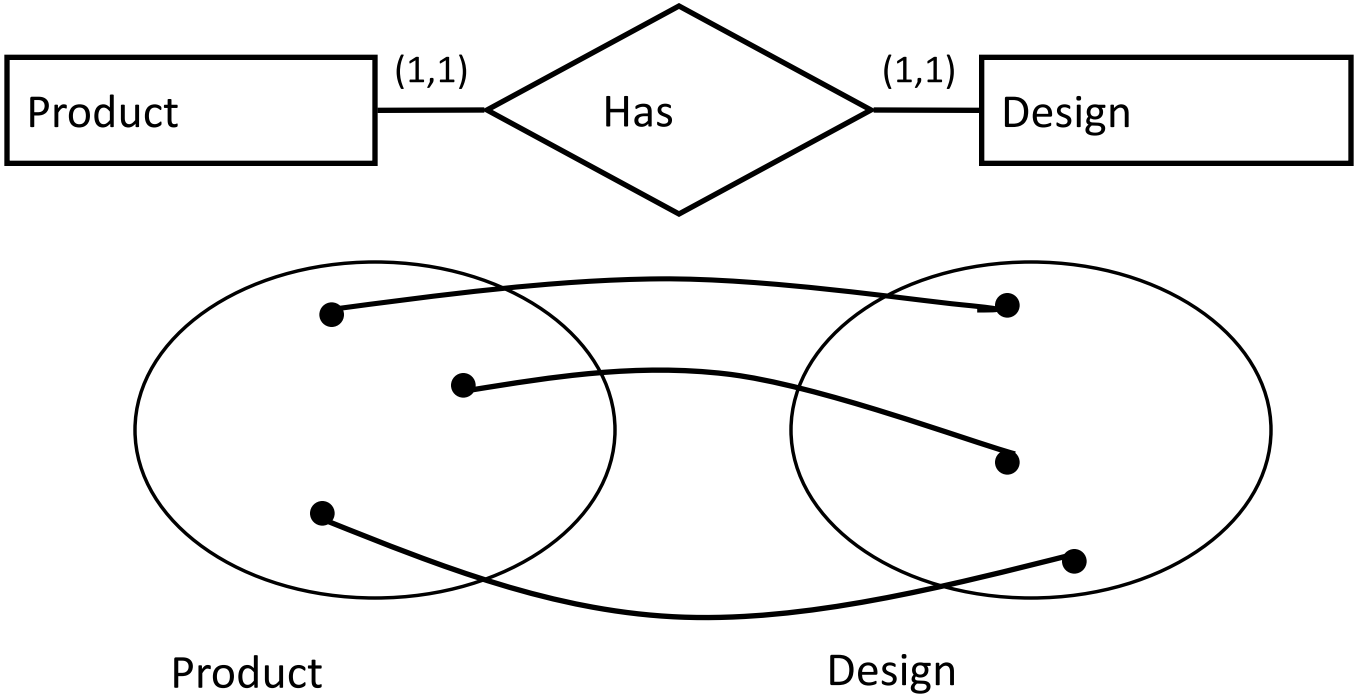
Cardinality of relationships

- Cardinality: a constraint on the number of association instances in which each entity must participate
- It’s a pair (MIN-CARD, MAX-CARD)

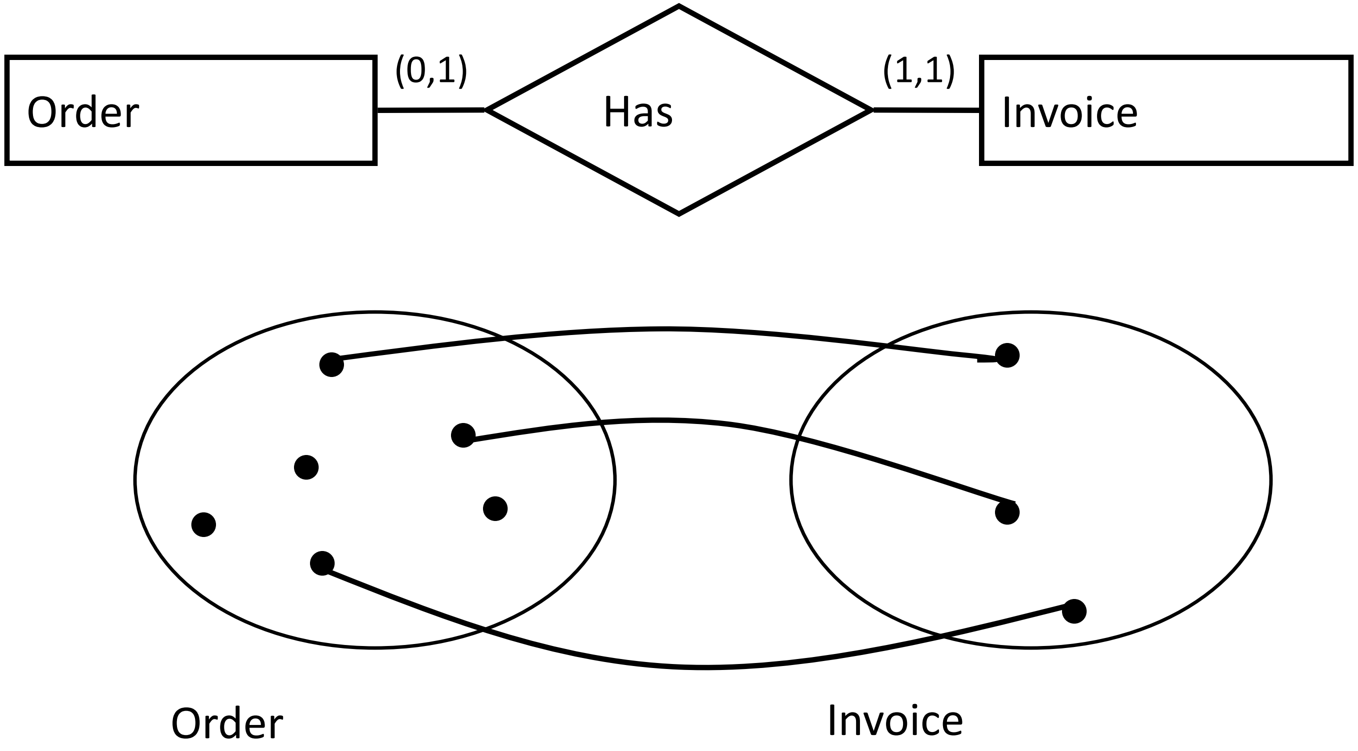
```
MIN-CARD  = 0  (optional)
          = 1 (mandatory)
MAX-CARD  = 1 (one)
          = N (many)
```

- MAX-CARD determines whether a relationship is one-to-one, one-to-many, or many-to-many

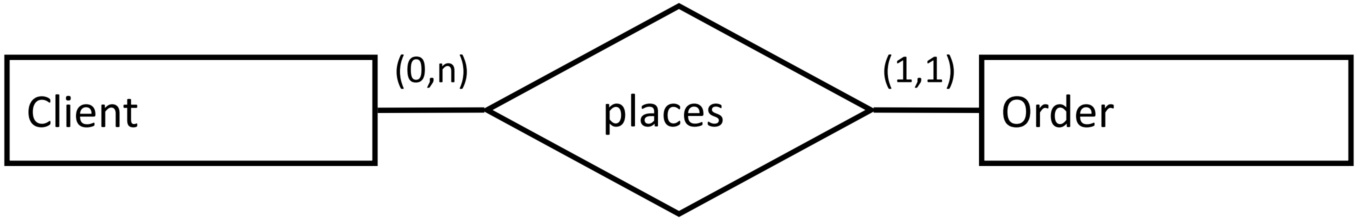
1:1 relationship



Optional 1:1 relationship



1:N Relationship



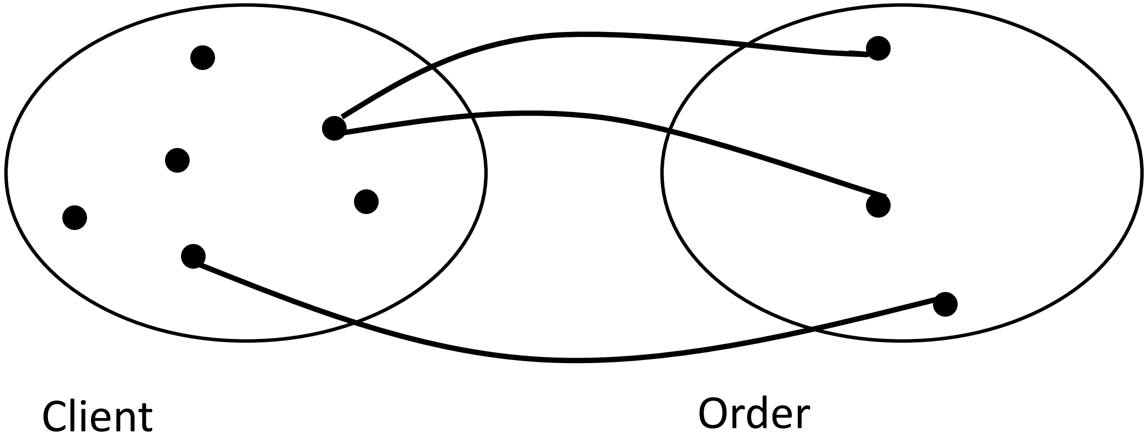
Client

(0,n)

places

(1,1)

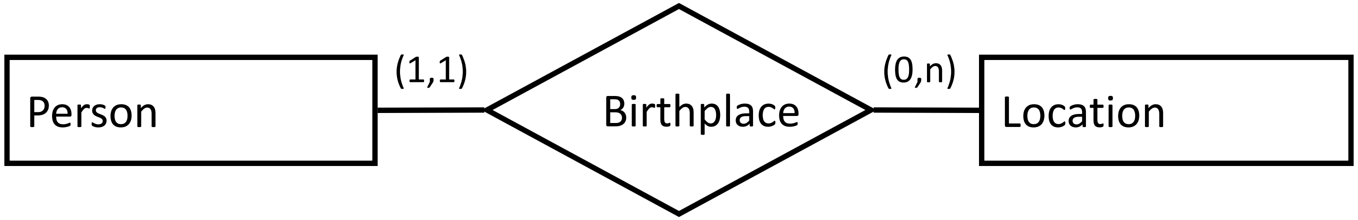
Order

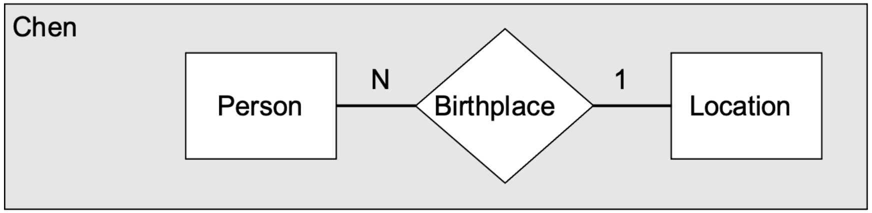


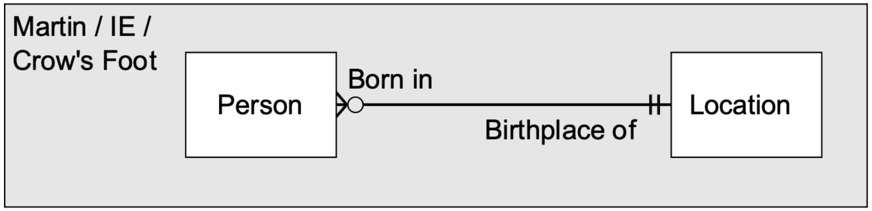
Client

Order

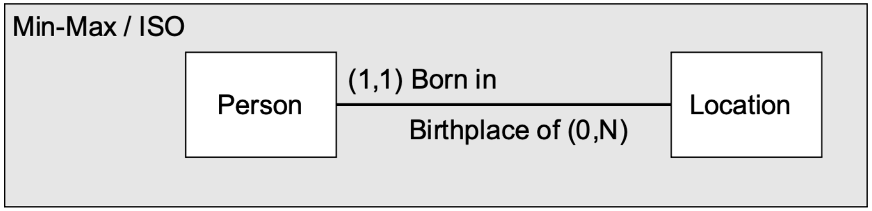
Caution: many different notations!

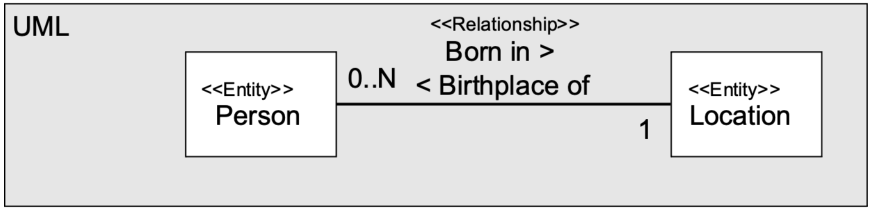




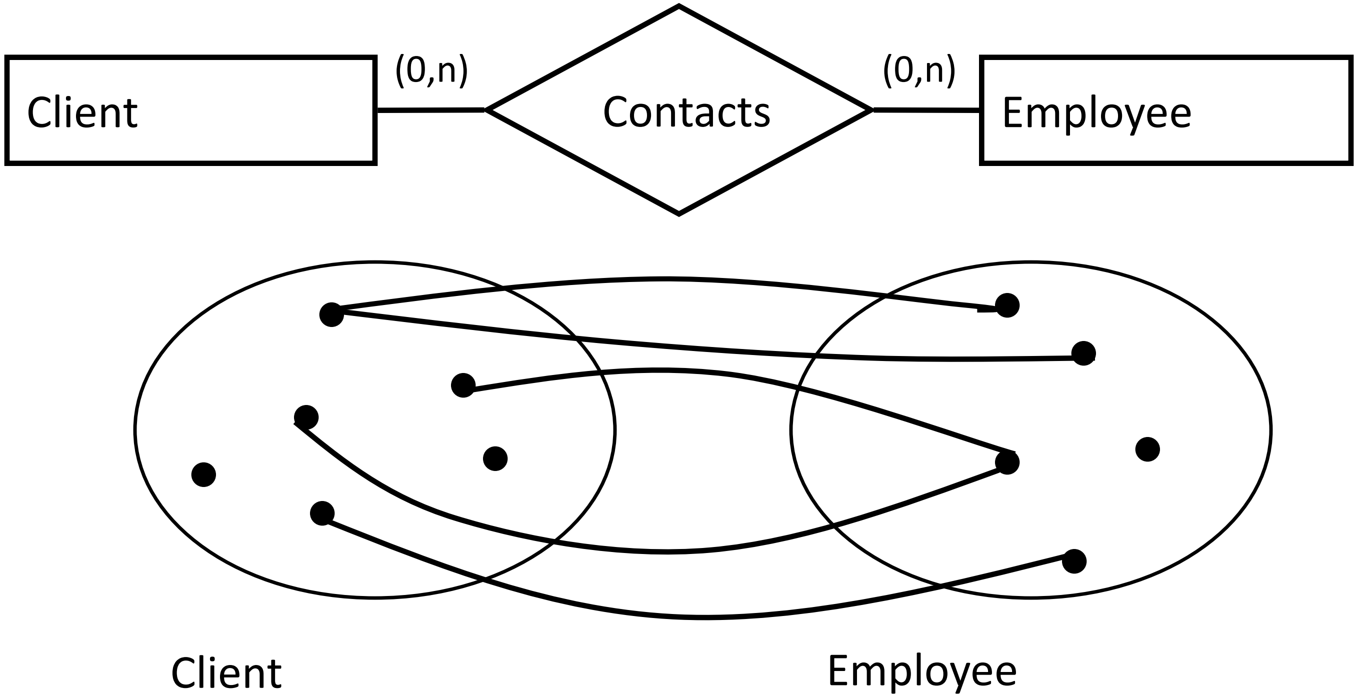


Used by ERDPlus

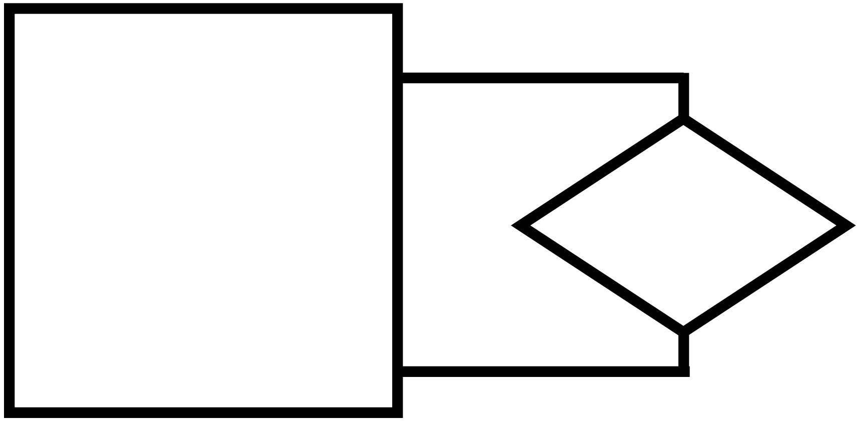




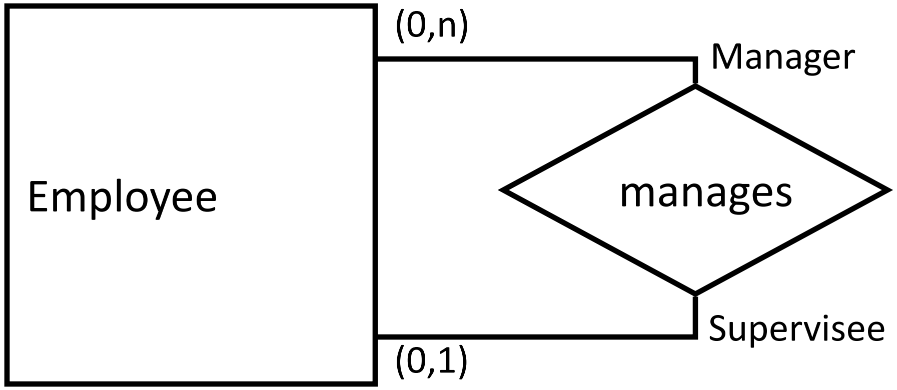
Many-to-many relationship

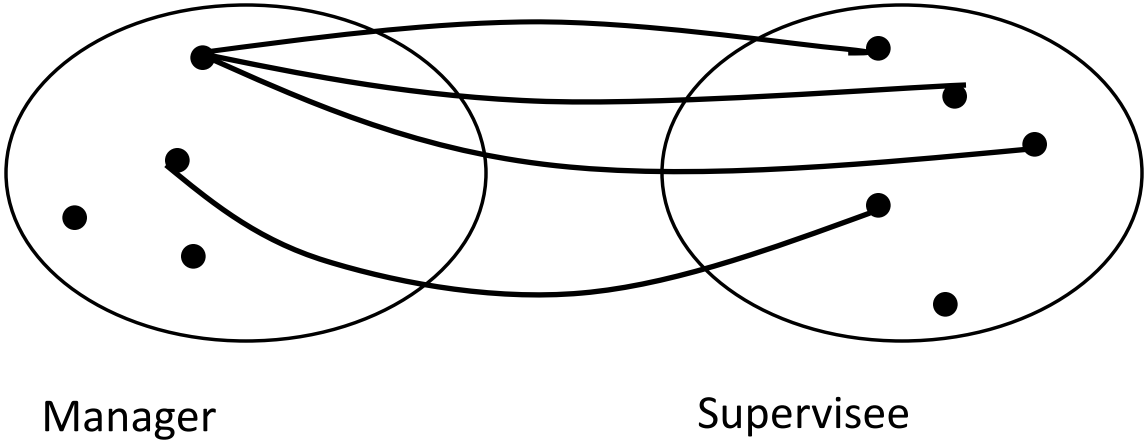


Self-relationships

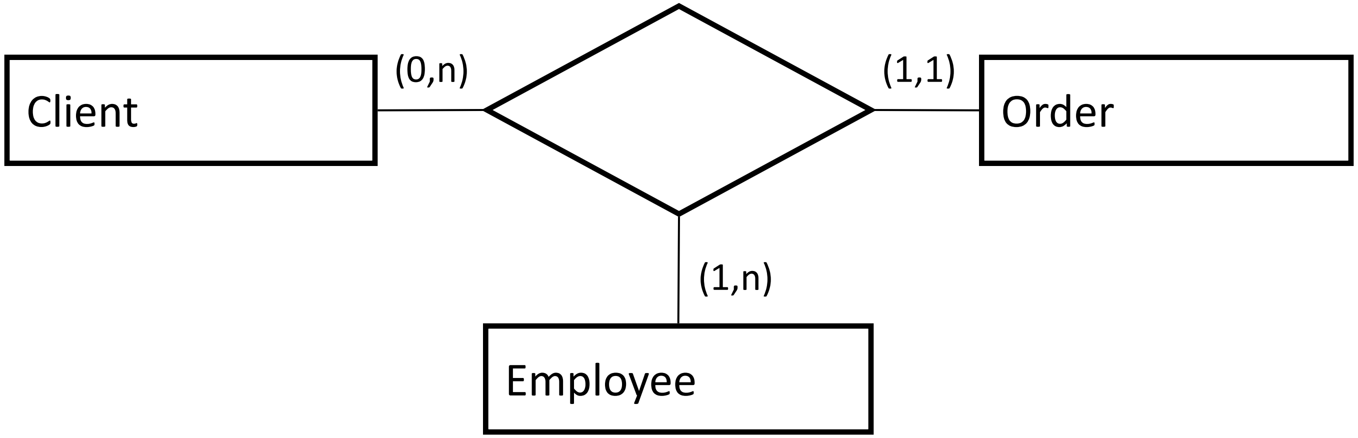


Self-relationships

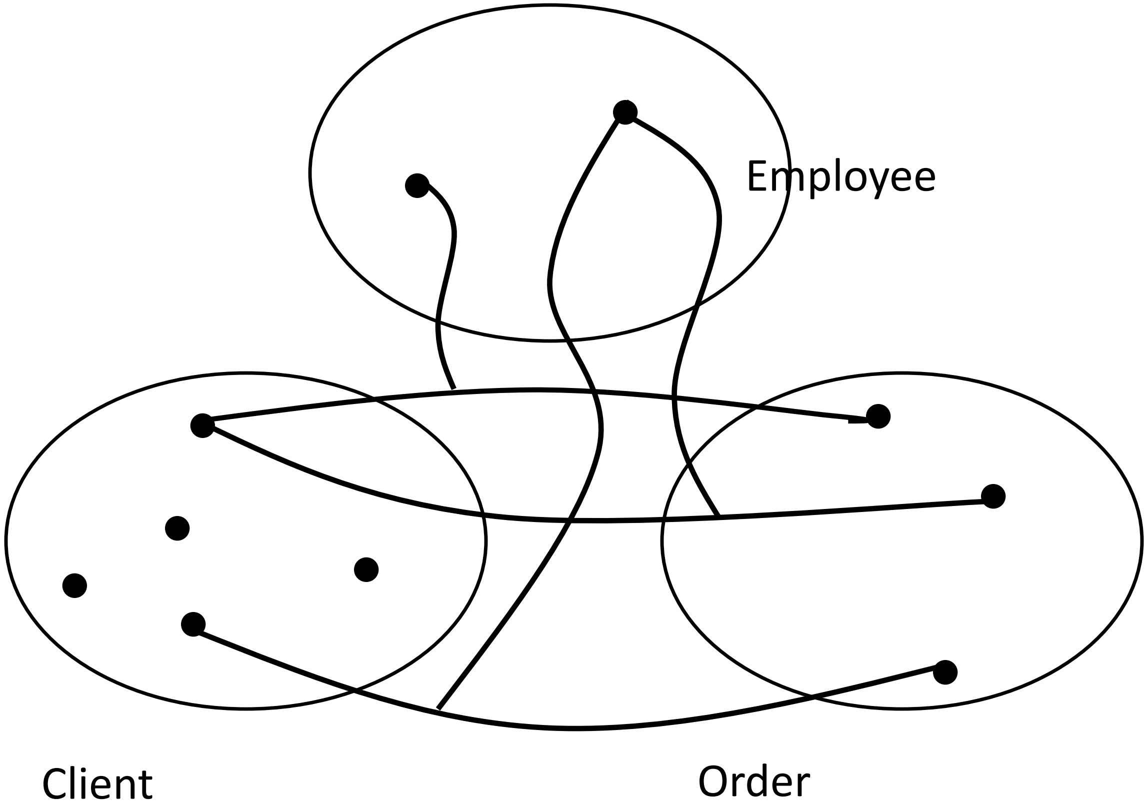




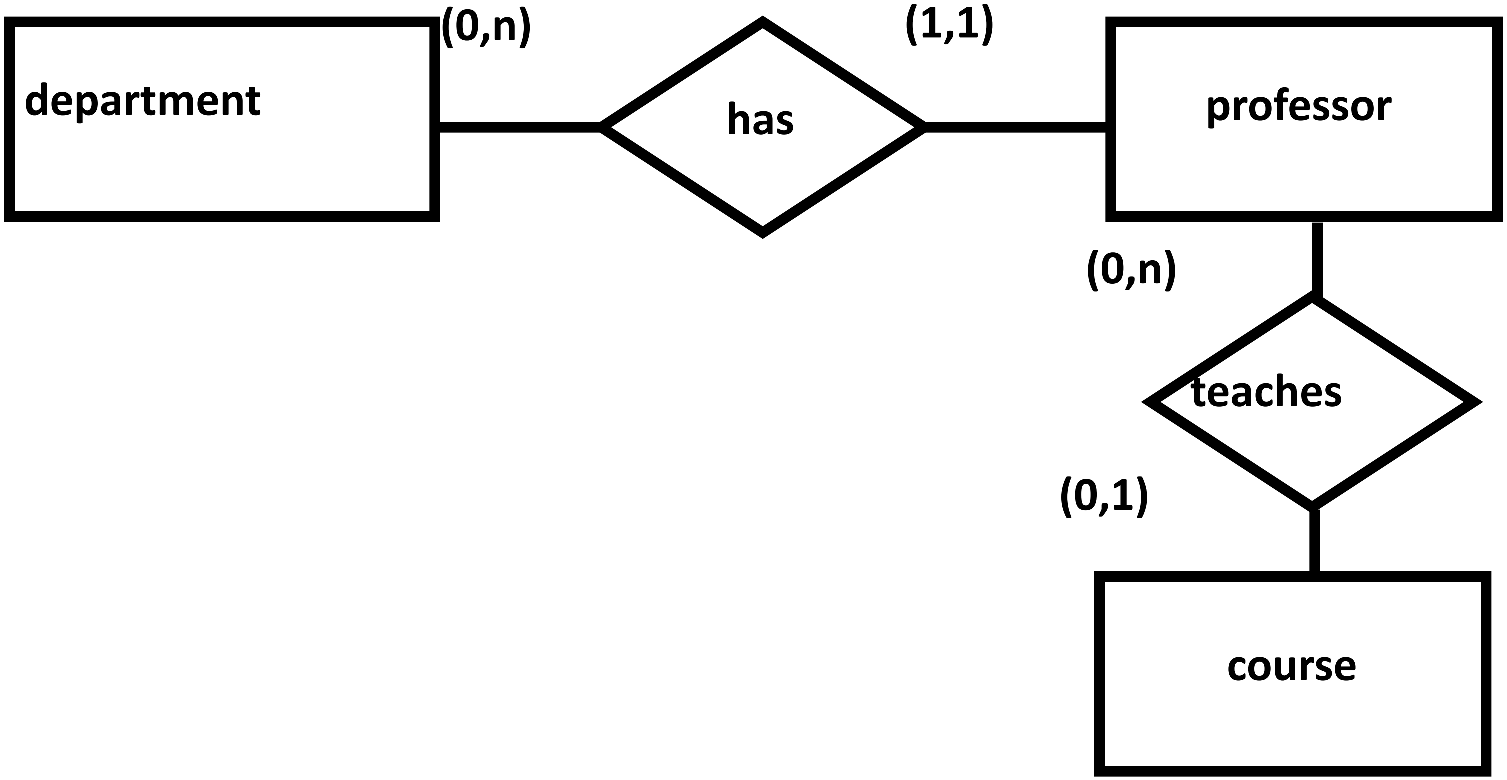
Ternary relationships



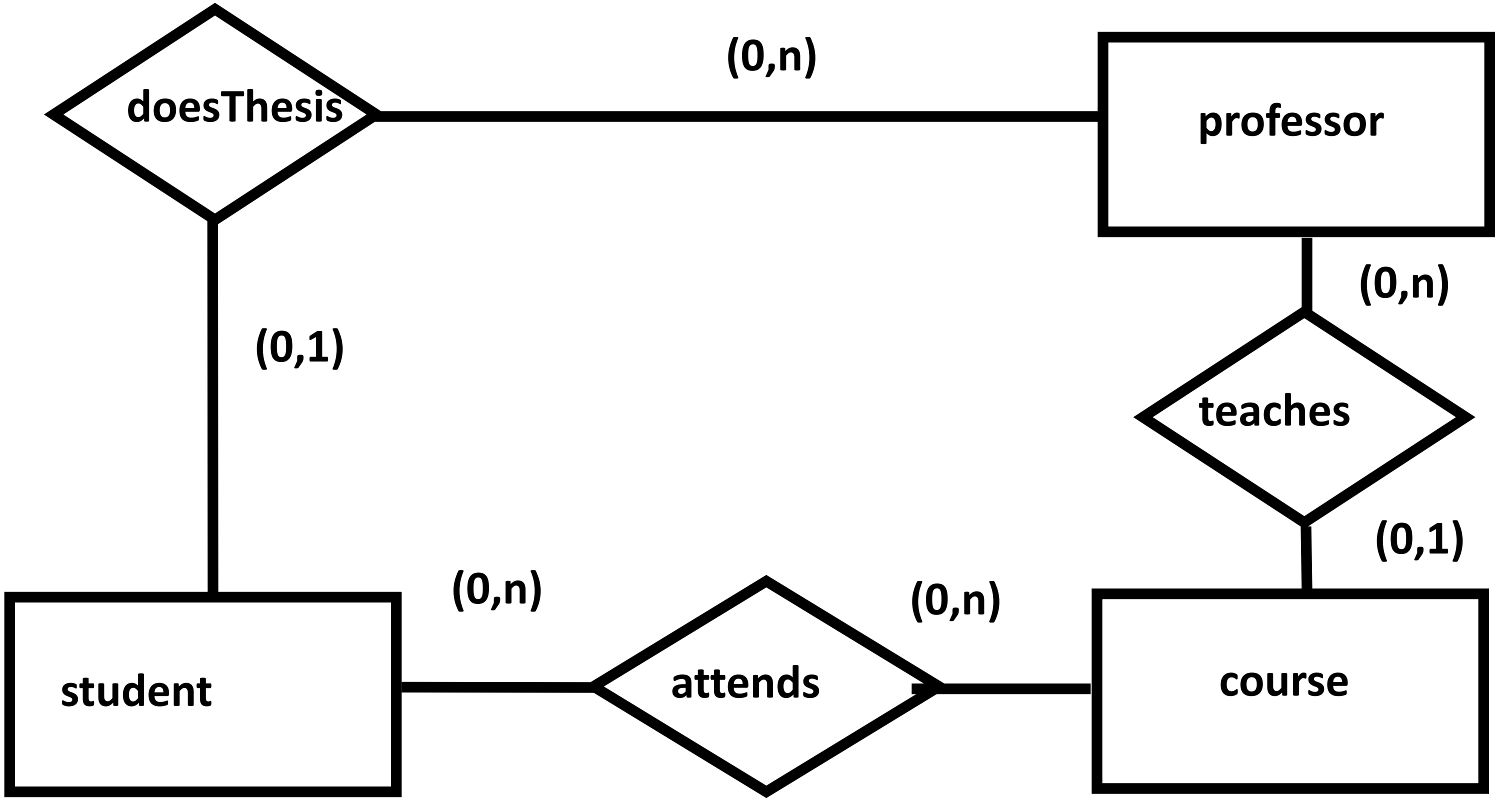
Ternary relationships



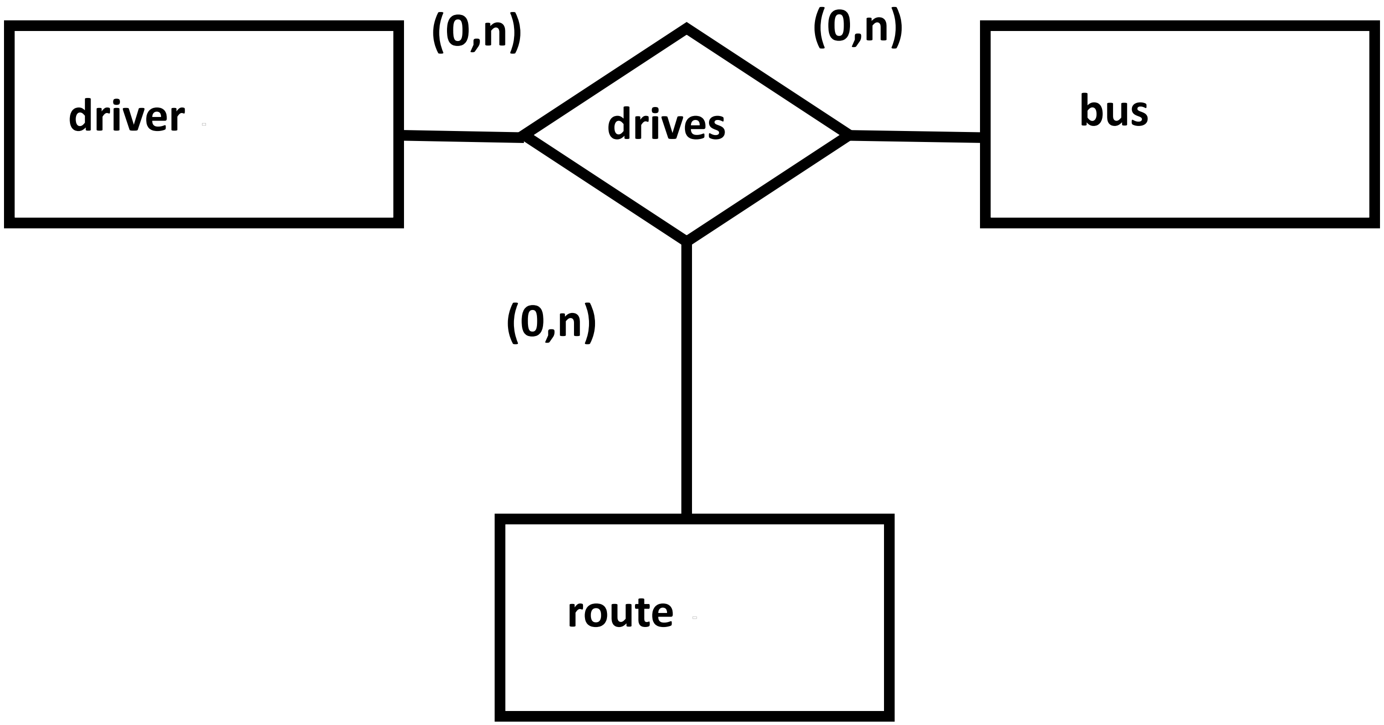
Example: university



Example: university

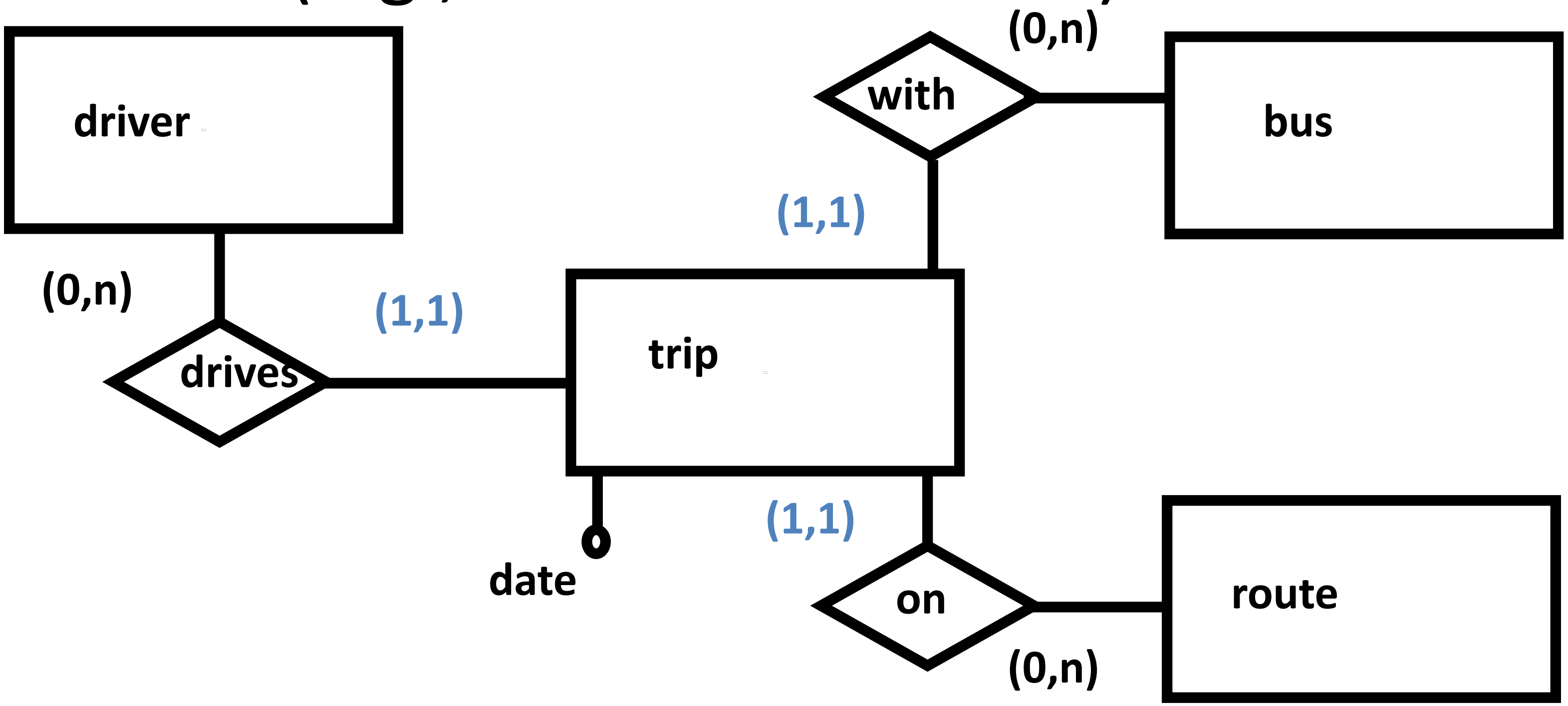


Example: buses



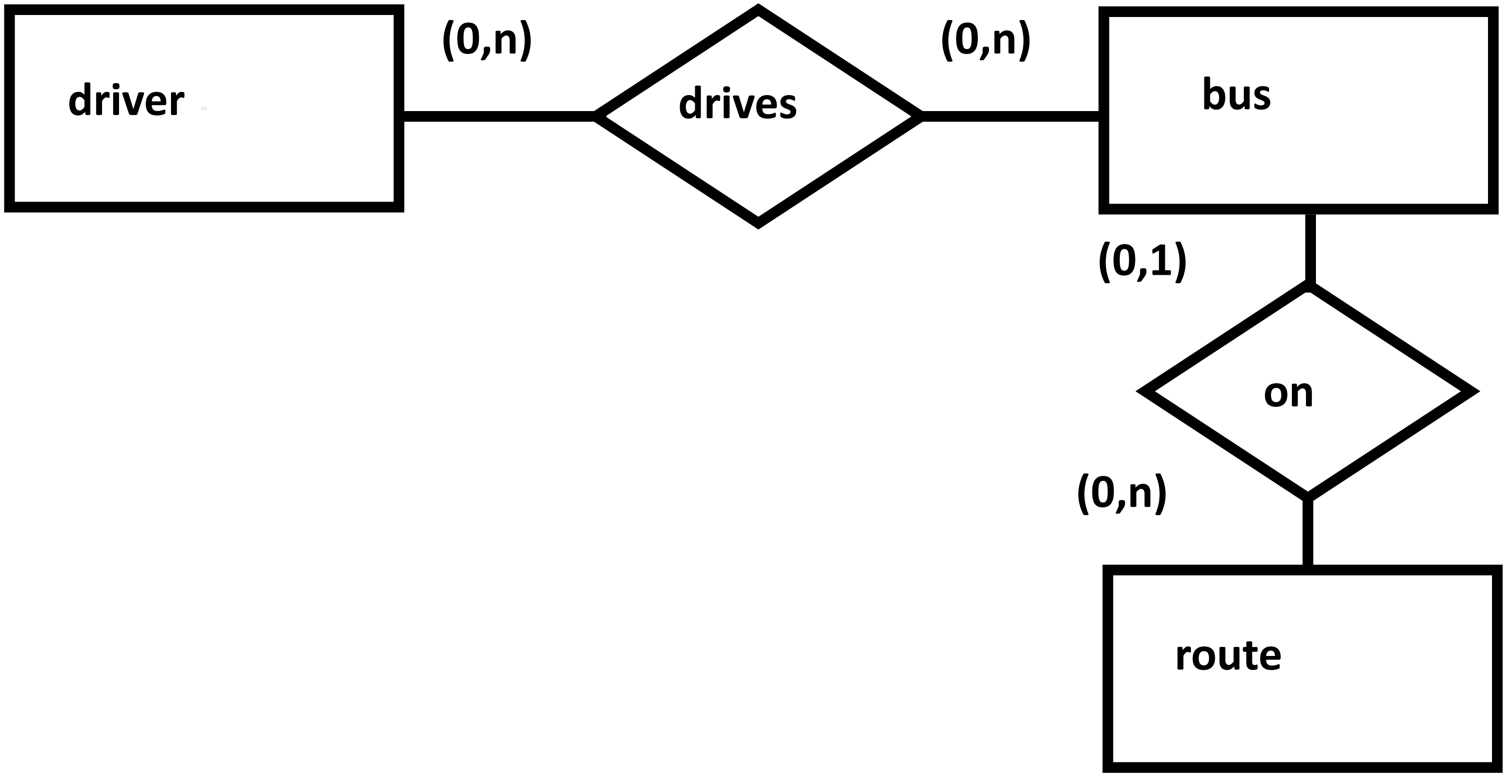
Bus example: reification

- This solution is needed if the same driver may use the same route on the same bus several times (e.g., on different dates)



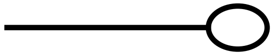
Bus example: removing one end

- This solution works if every bus can only use one route



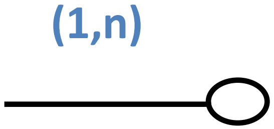
Cardinality of attributes

- **scalar (simple, one-valued)**



ex.: lastname, mark

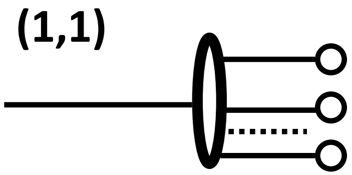
multiple (n values are possible)



ex.: qualification, title

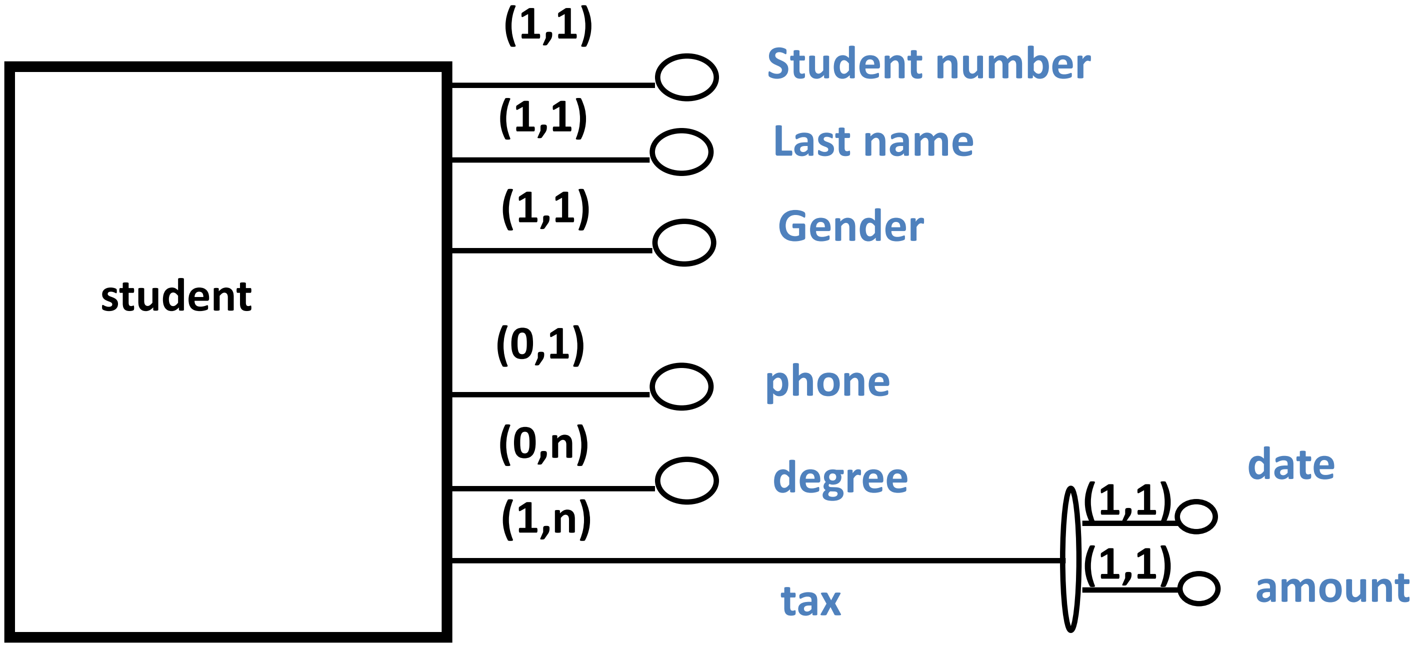
Compound attributes

compound



ex.: date (dd,mm,yyyy)

Example

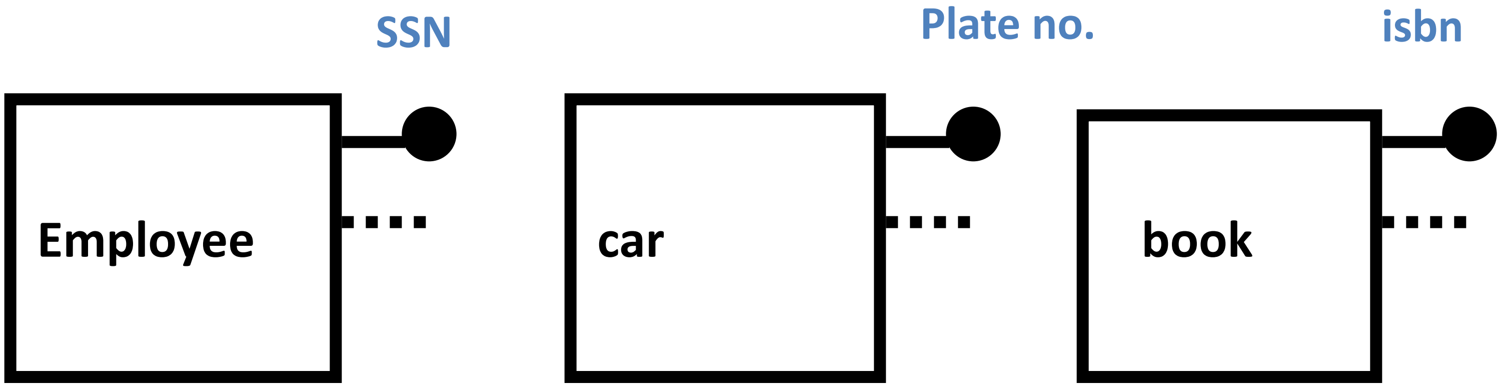

(1,1)
Student number
(1,1)
Last name
(1,1)
Gender
student
(0,1)
phone
(0,n)
degree
(1,n)
tax
date
(1,1)
(1,1)
amount

Identifier

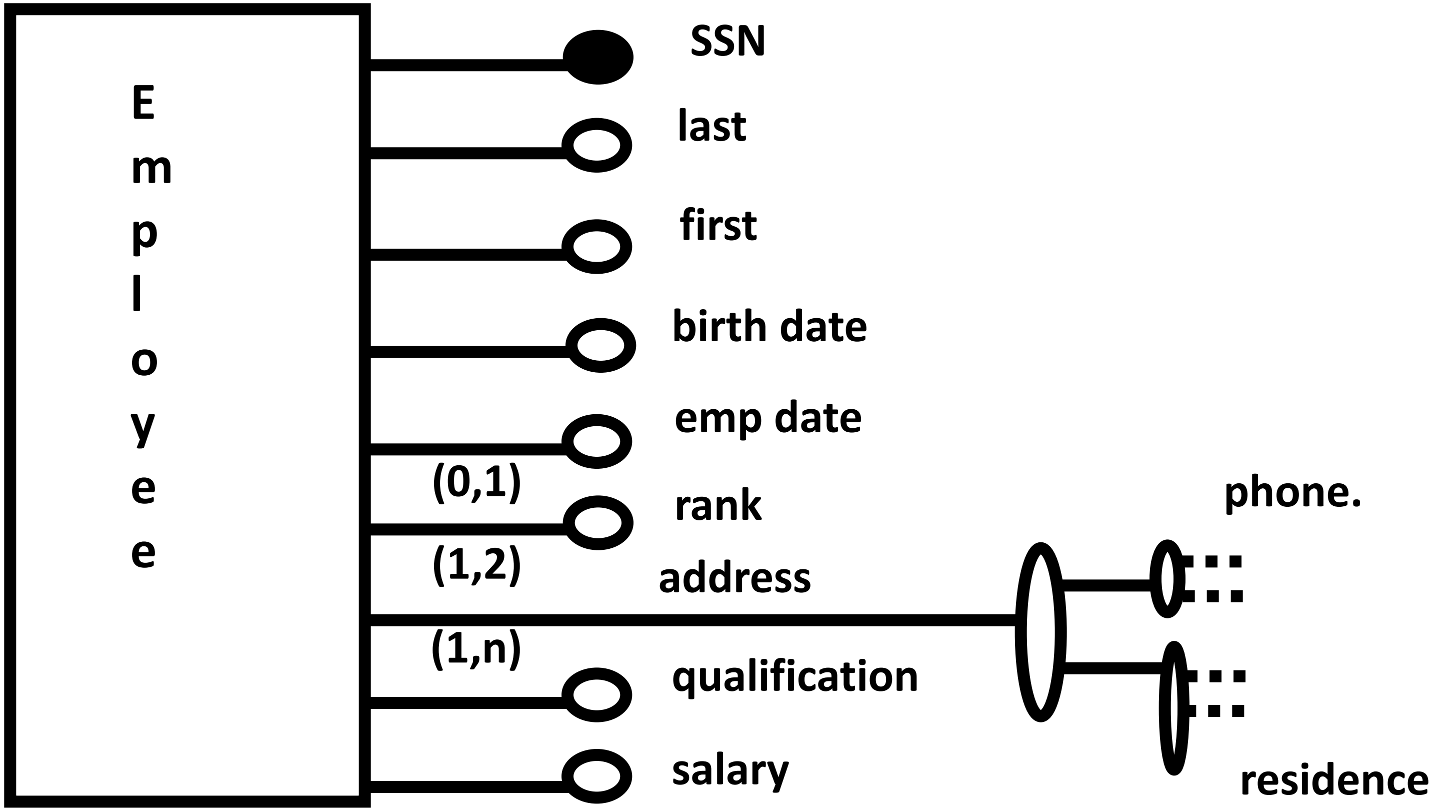
An identifier univocally characterizes every single entity

symbol

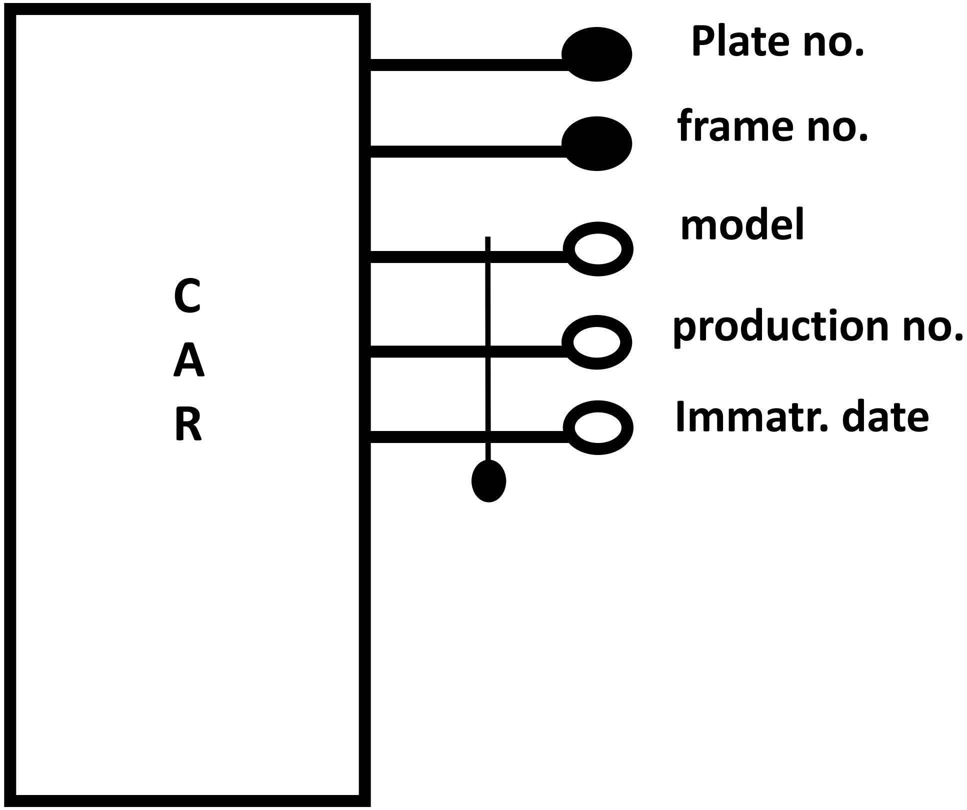
—●



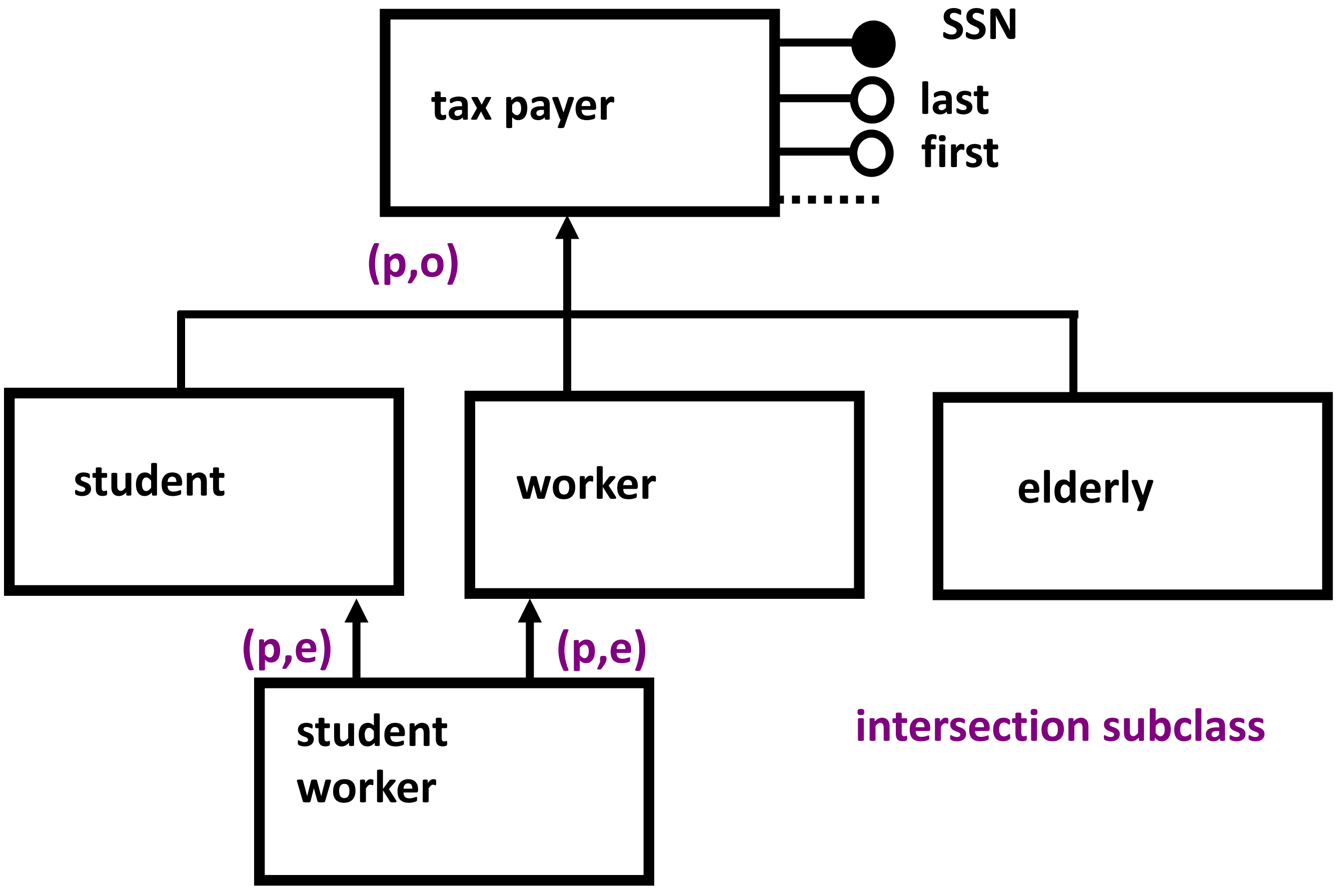
Example


E
m
p
l
o
y
e
e
SSN
last
first
birth date
emp date
(0,1)
rank
(1,2)
address
(1,n)
qualification
salary
phone.
residence

Example with compound identifier



Example of hierarchy

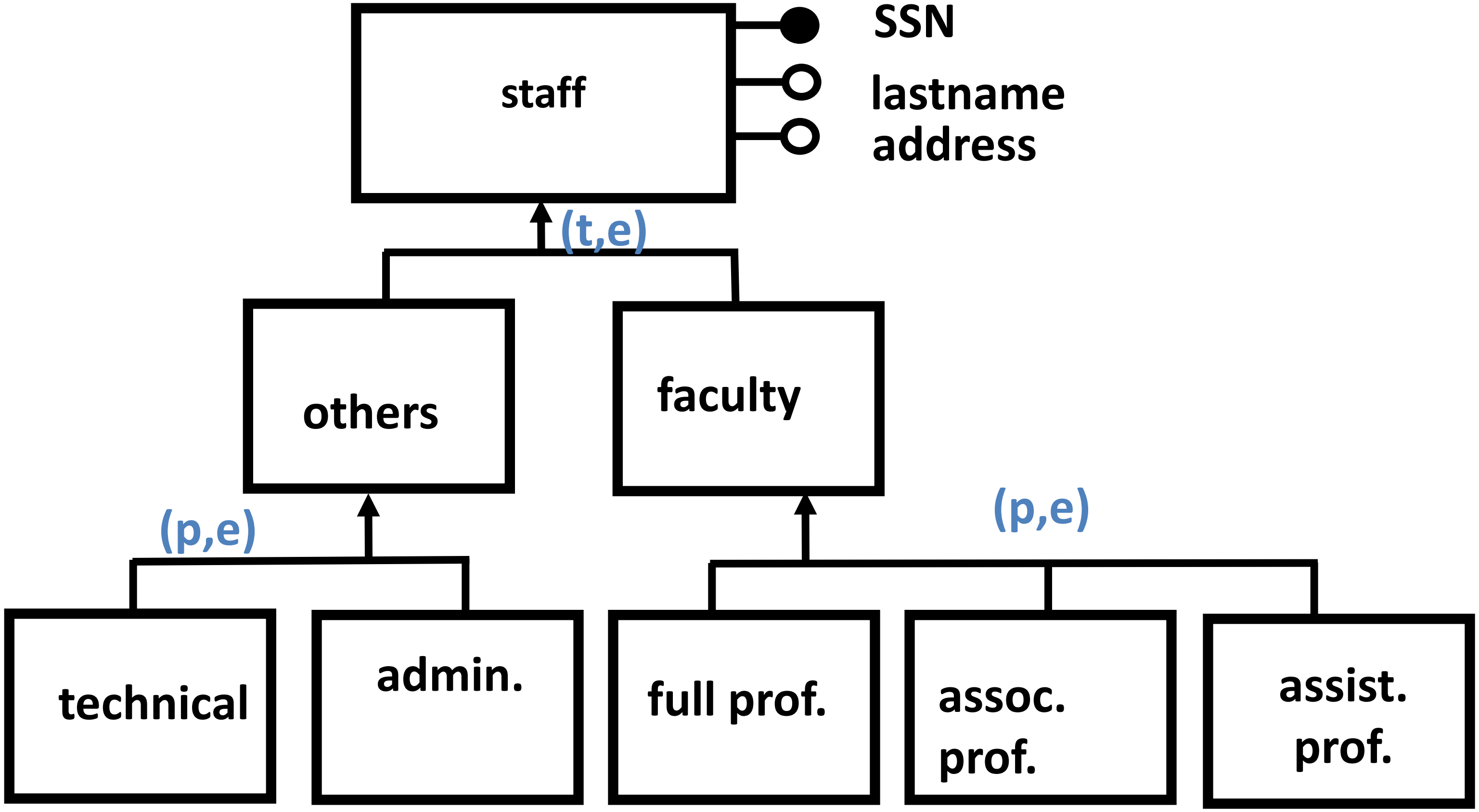


intersection subclass

Properties of hierarchies

- ***t*** **- total: every instance of the parent entity must belong to one of the child entities**
- ***p*** **- partial: the instances of the parent entity may be part of one of the child entities**
- ***e*** **- exclusive: every instance of the parent entity cannot be part of more than one of the child entities**
- ***o*** **- overlapping: every instance of the parent entity may be part of several child entities**

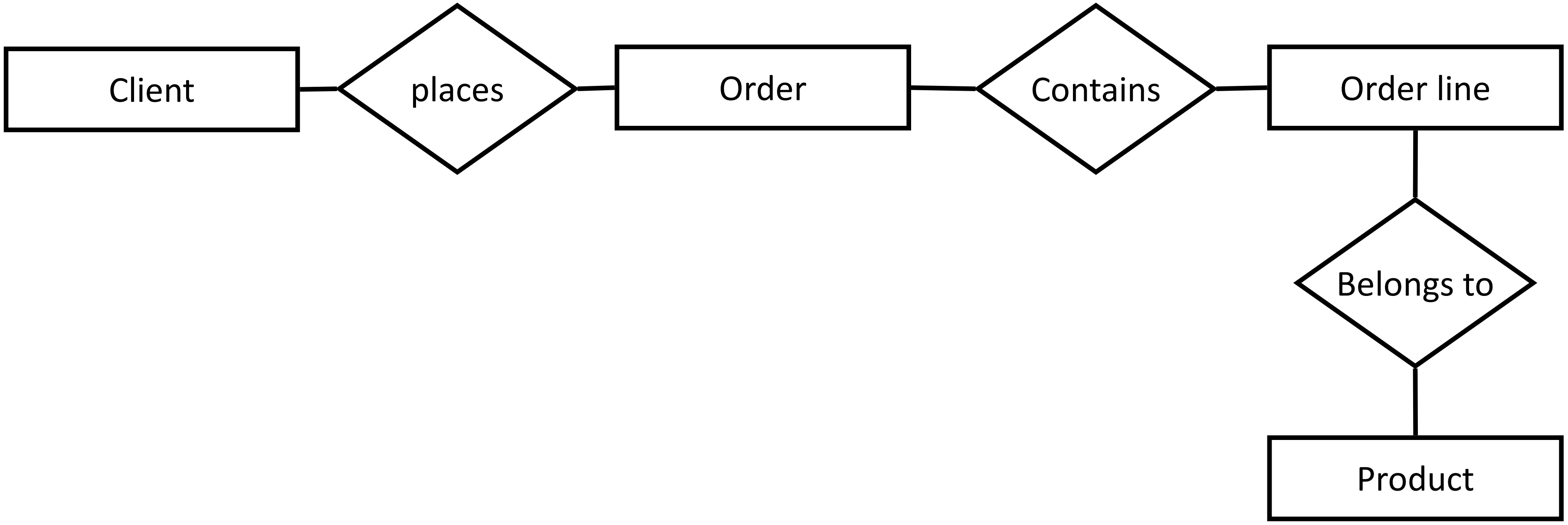
Example: university



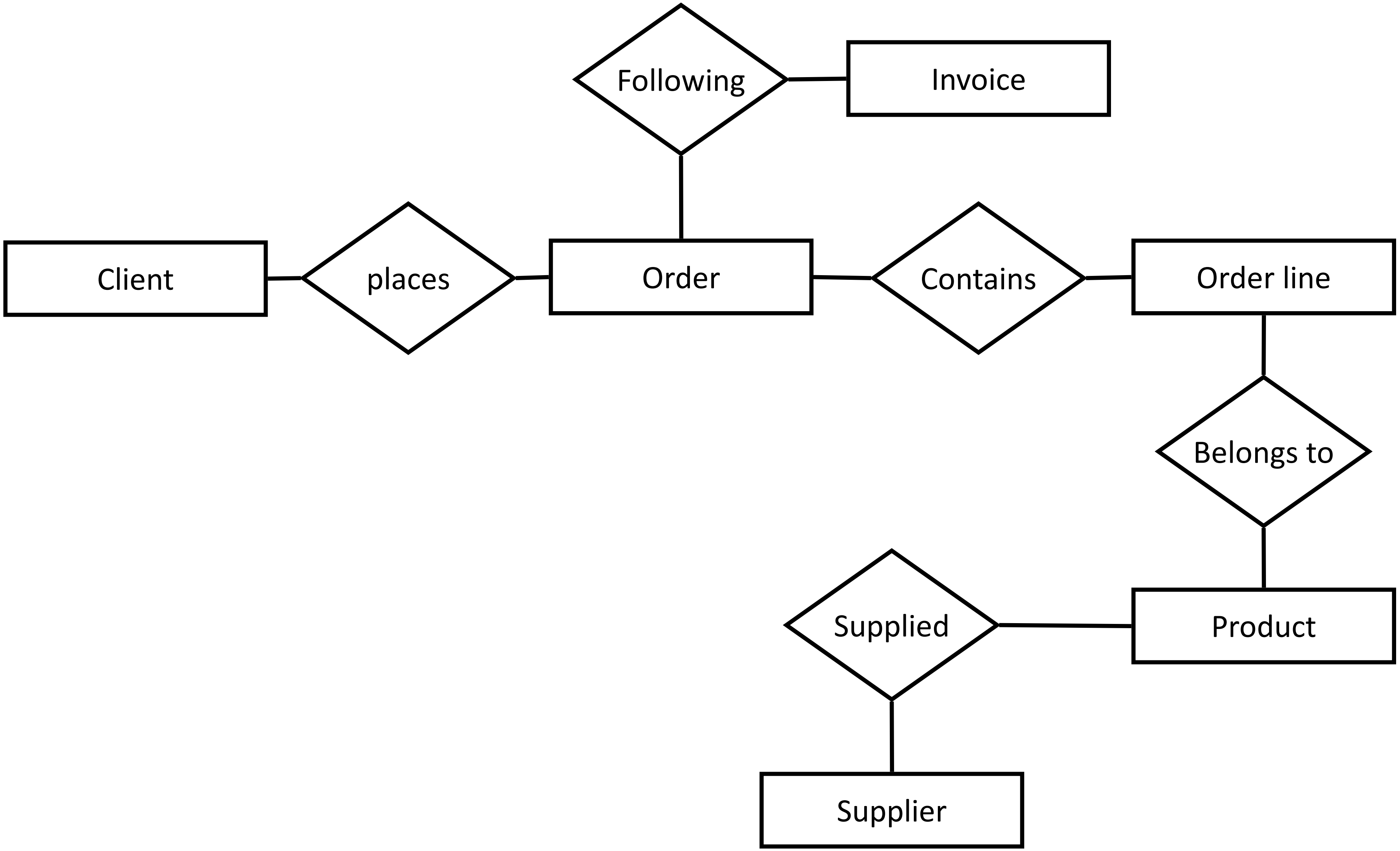
Top-down vs bottom-up design

- Top-down: refining in different steps, starting from an overall description of the domain of interest
- Bottom-up: different aspects are designed separately, and then integrated into a single schema

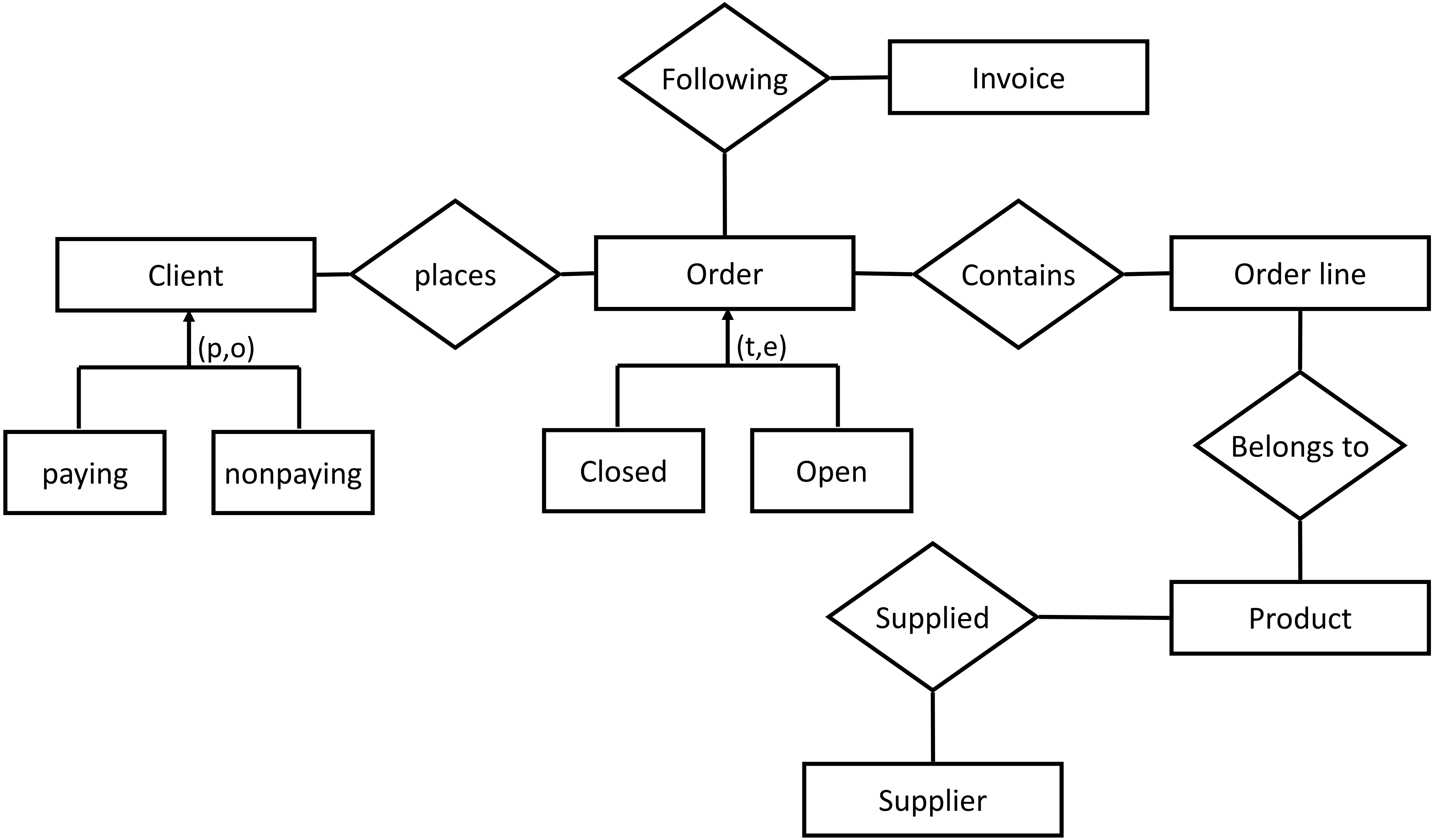
Example: pure top-down



Example: pure top-down



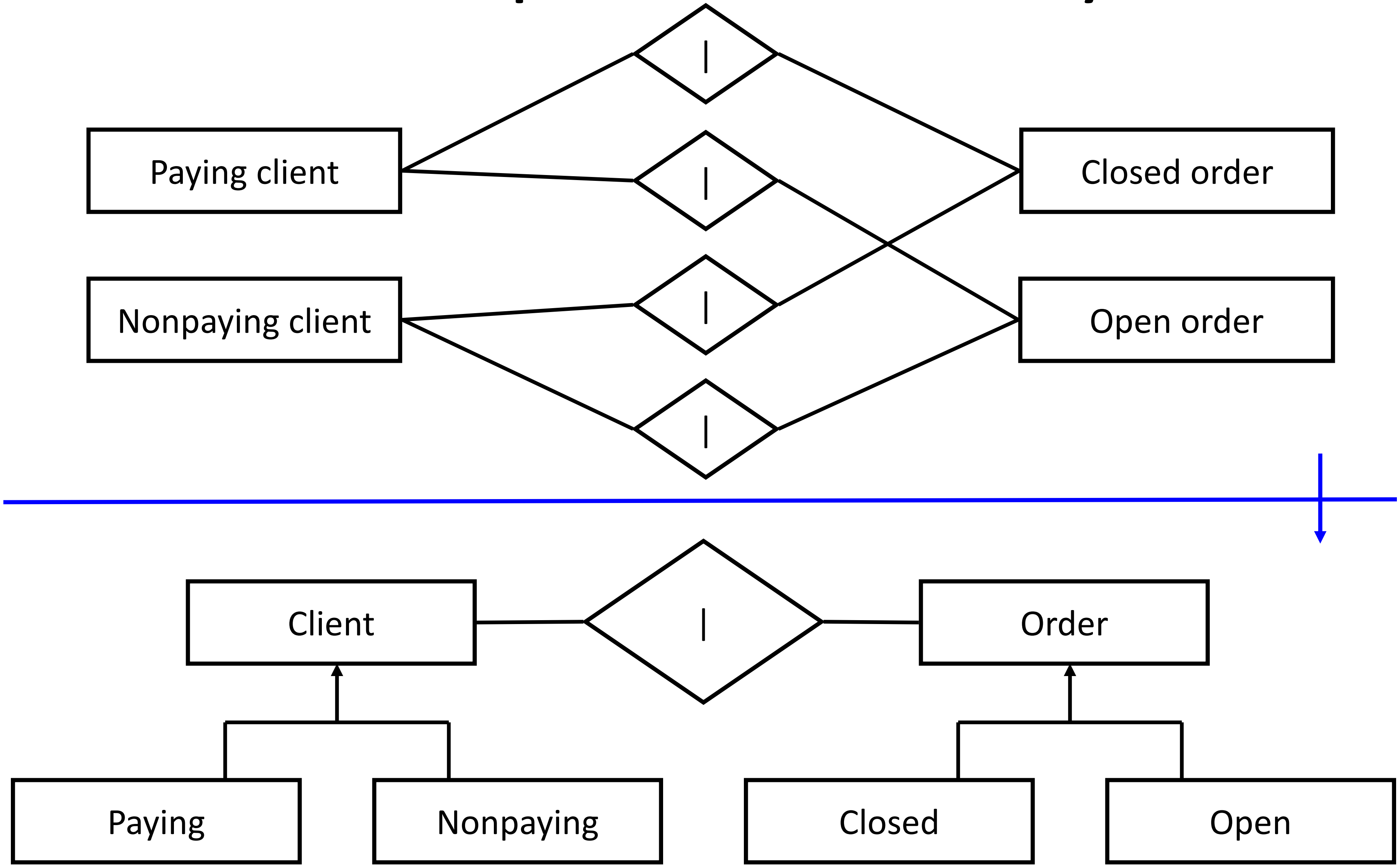
Example: pure top-down



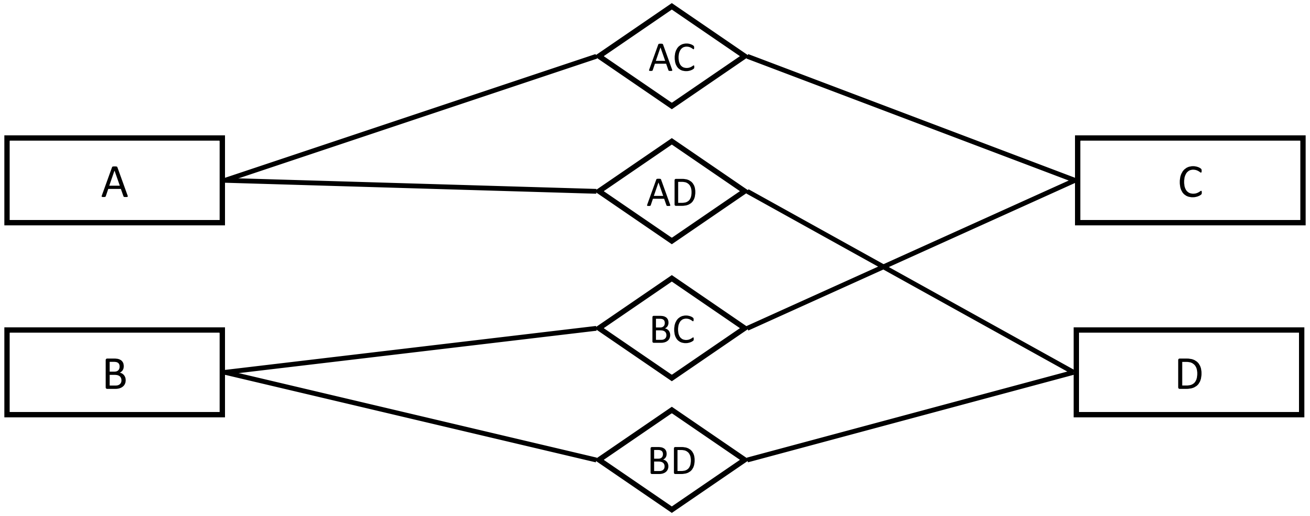
Quality of a conceptual schema

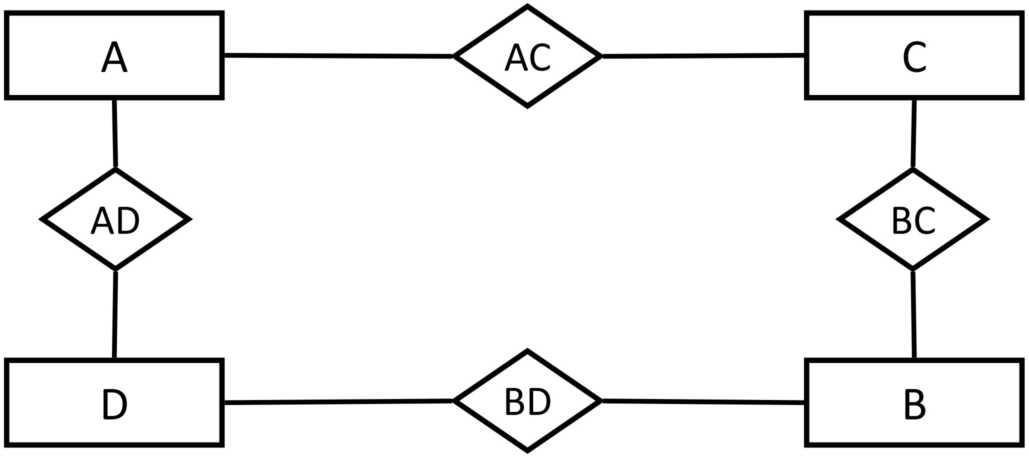
- Completeness
- Correctness
- Readability
- Minimality
- Self-explanatoriness

Conceptual readability

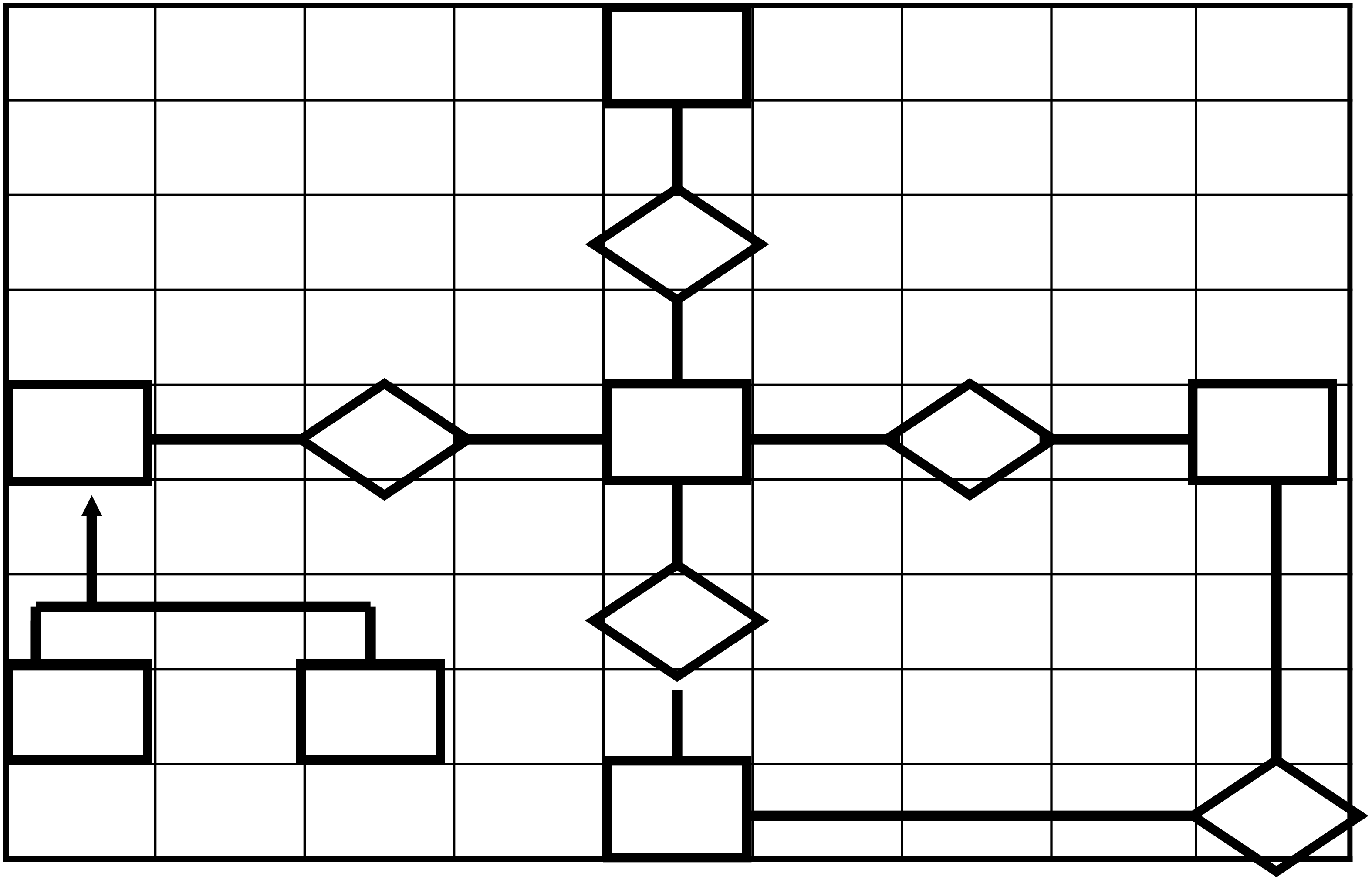


Graphical readability

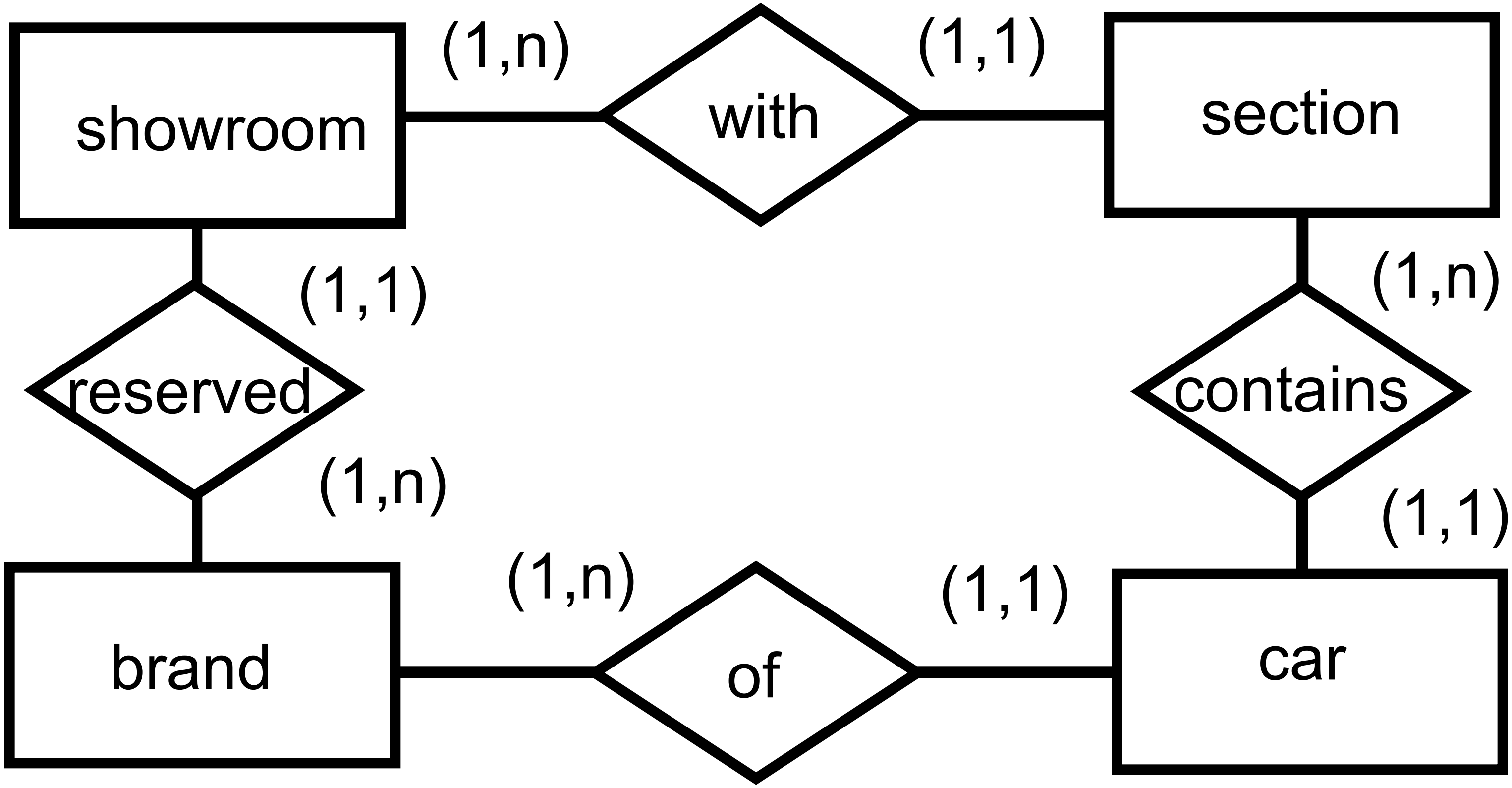




Draw in a grid!



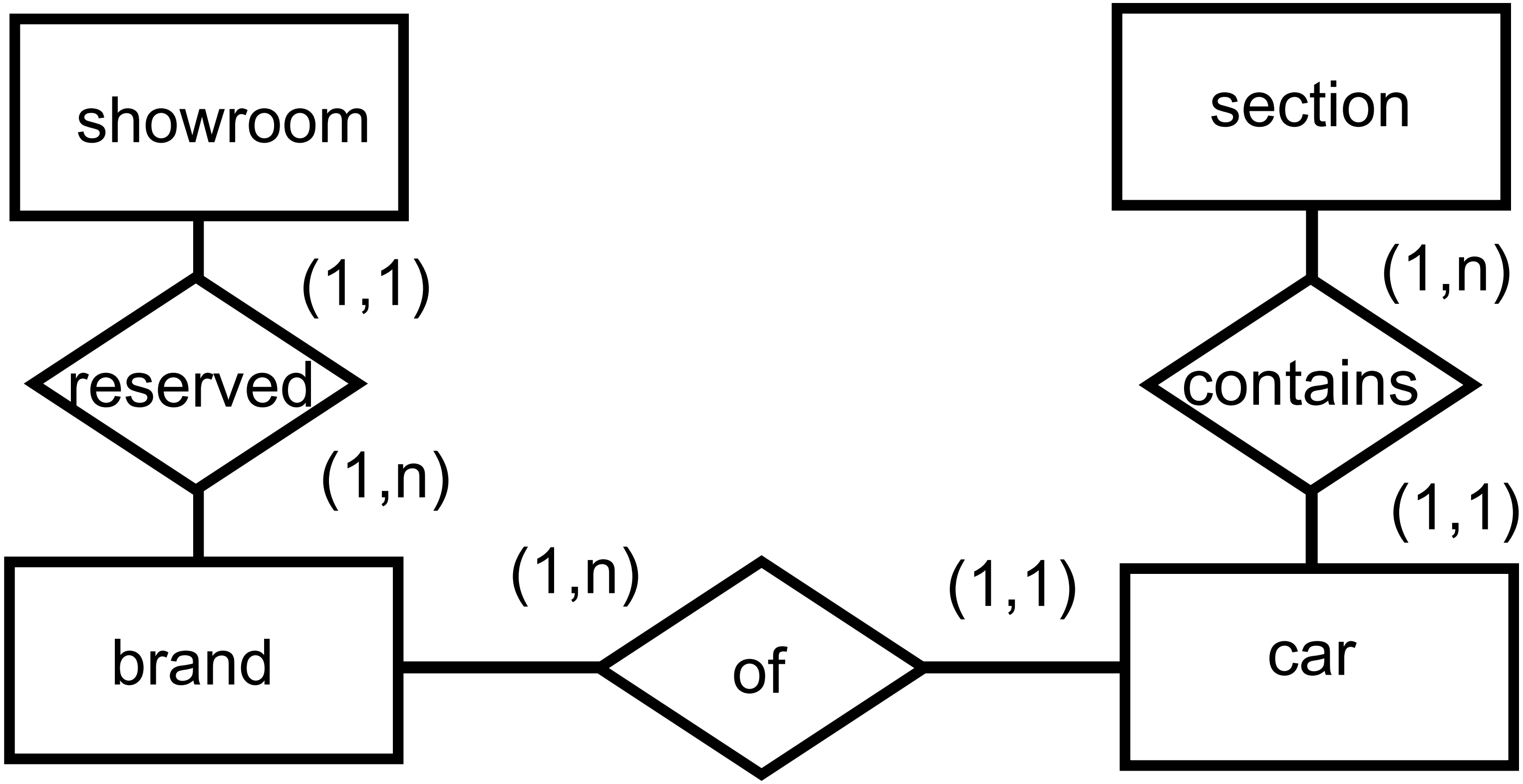
Minimality vs redundancy



- Showrooms are reserved for certain brands of car
- Showrooms consist of sections, which contain cars
- Cars are of specific brands

There might be some redundancy: let’s see

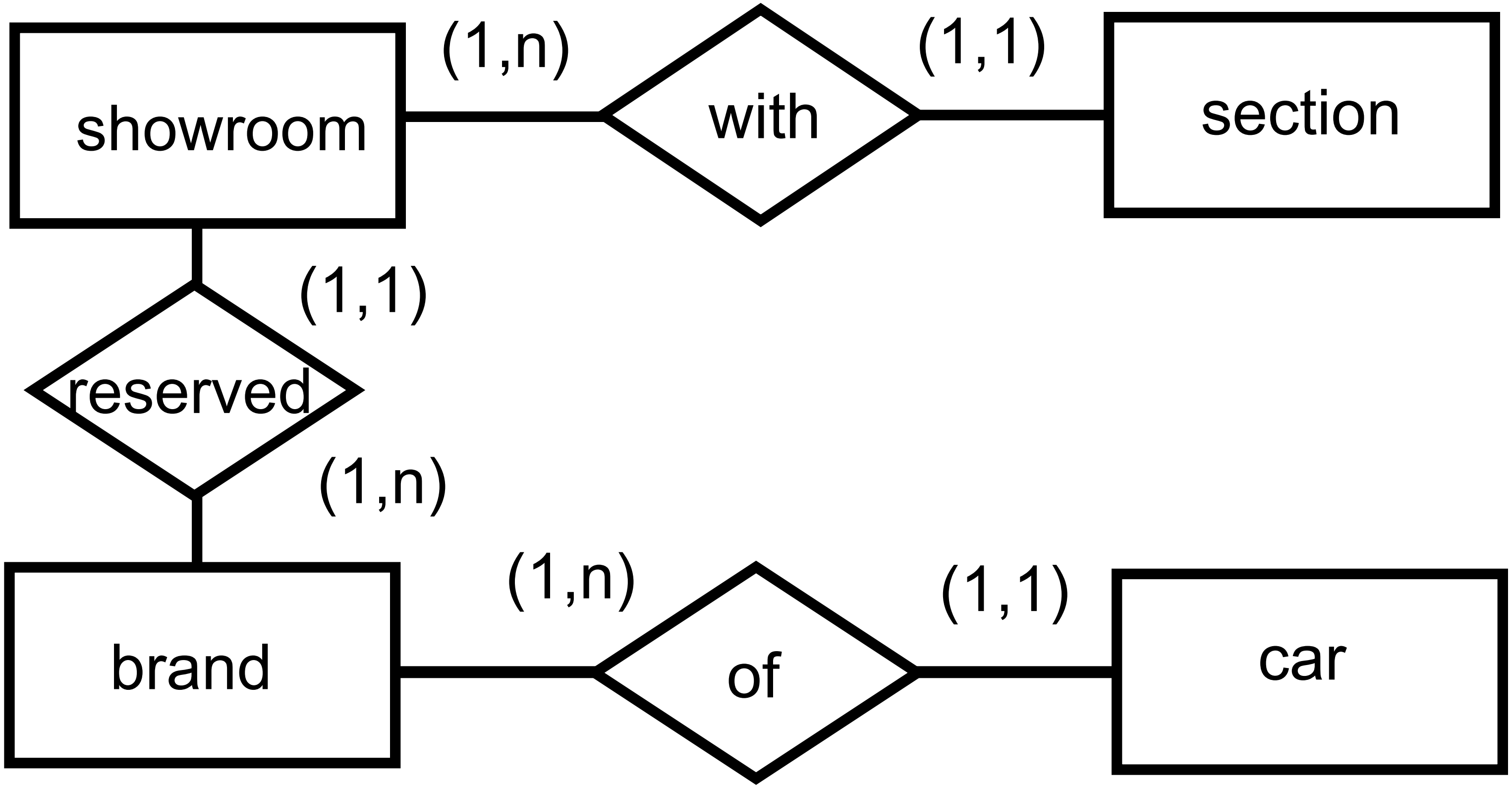
Minimality vs redundancy



- Showrooms are reserved for certain brands of car
- Showrooms consist of sections, which contain cars
- Cars are of specific brands

Not ok: given a section, can’t find its showroom

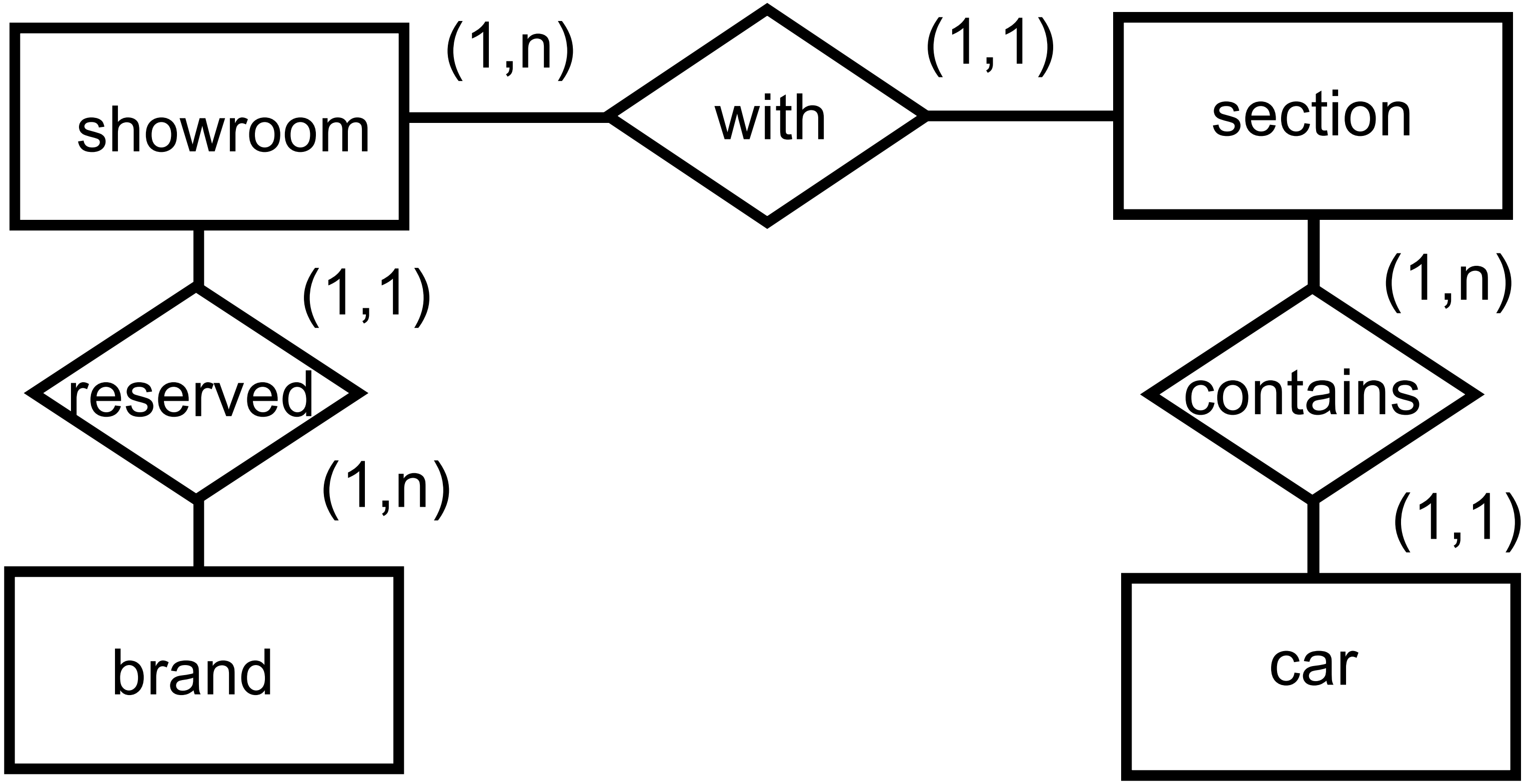
Minimality vs redundancy



- Showrooms are reserved for certain brands of car
- Showrooms consist of sections, which contain cars
- Cars are of specific brands

Not ok: given a car, can’t find its section

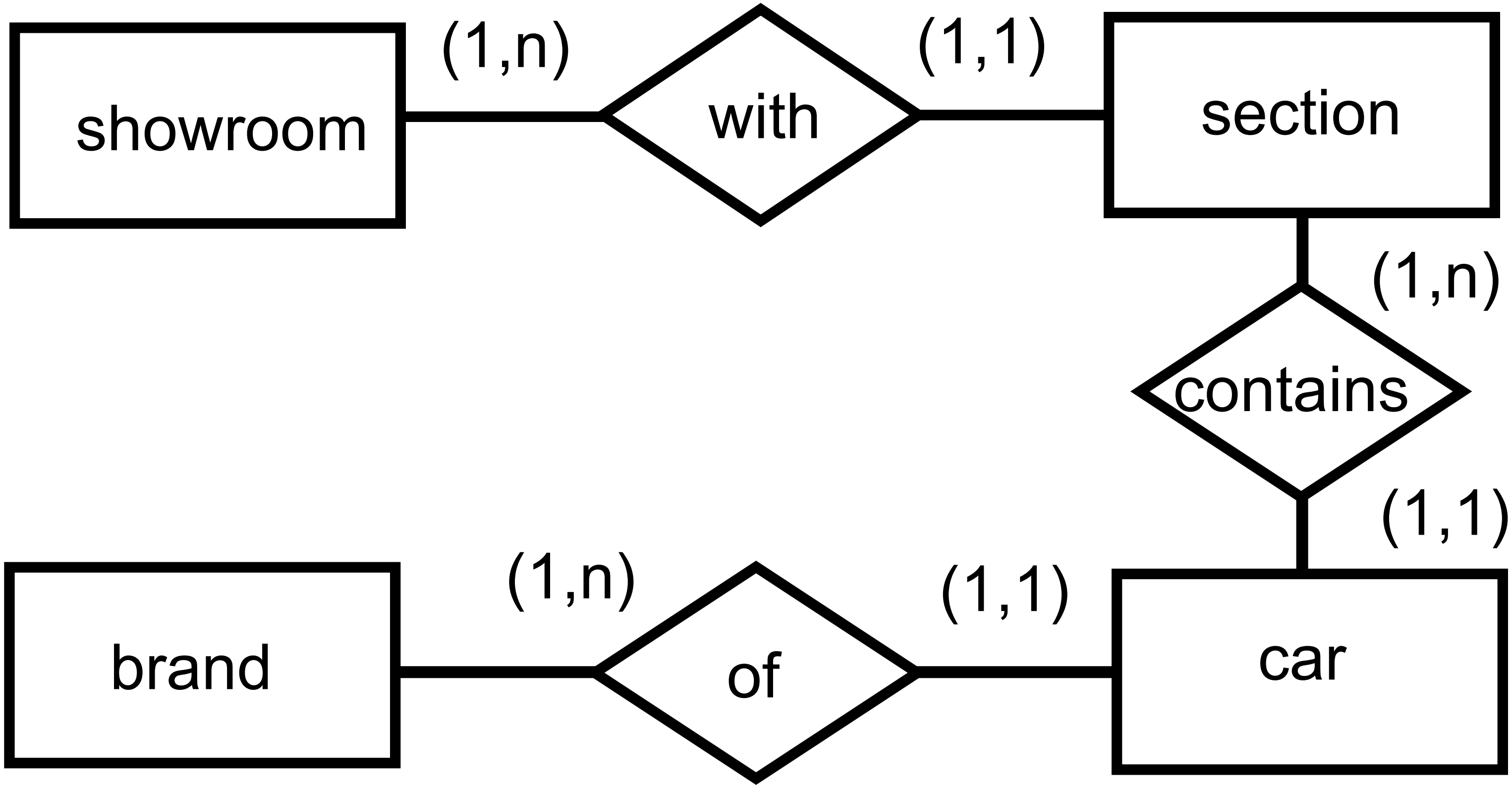
Minimality vs redundancy



- Showrooms are reserved for certain brands of car
- Showrooms consist of sections, which contain cars
- Cars are of specific brands

Ok: no loss of information

Minimality vs redundancy



- Showrooms are reserved for certain brands of car
- Showrooms consist of sections, which contain cars
- Cars are of specific brands

Ok: no loss of information (less obvious)

FROM ER TO RELATIONAL SCHEMA

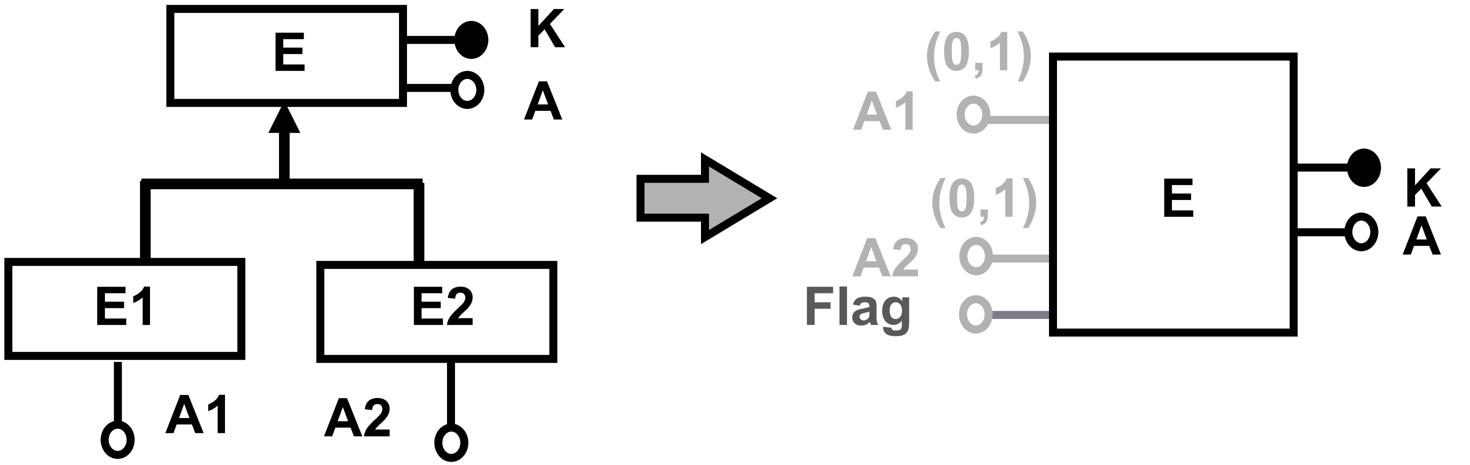
Phases of logical design

- Elimination of hierarchies
- Selection of keys
- Normalization of multiple attributes
- Elimination of relationships

- The result is a set of tables with Primary Key and Foreign Key constraints

Elimination of hierarchies: option 1

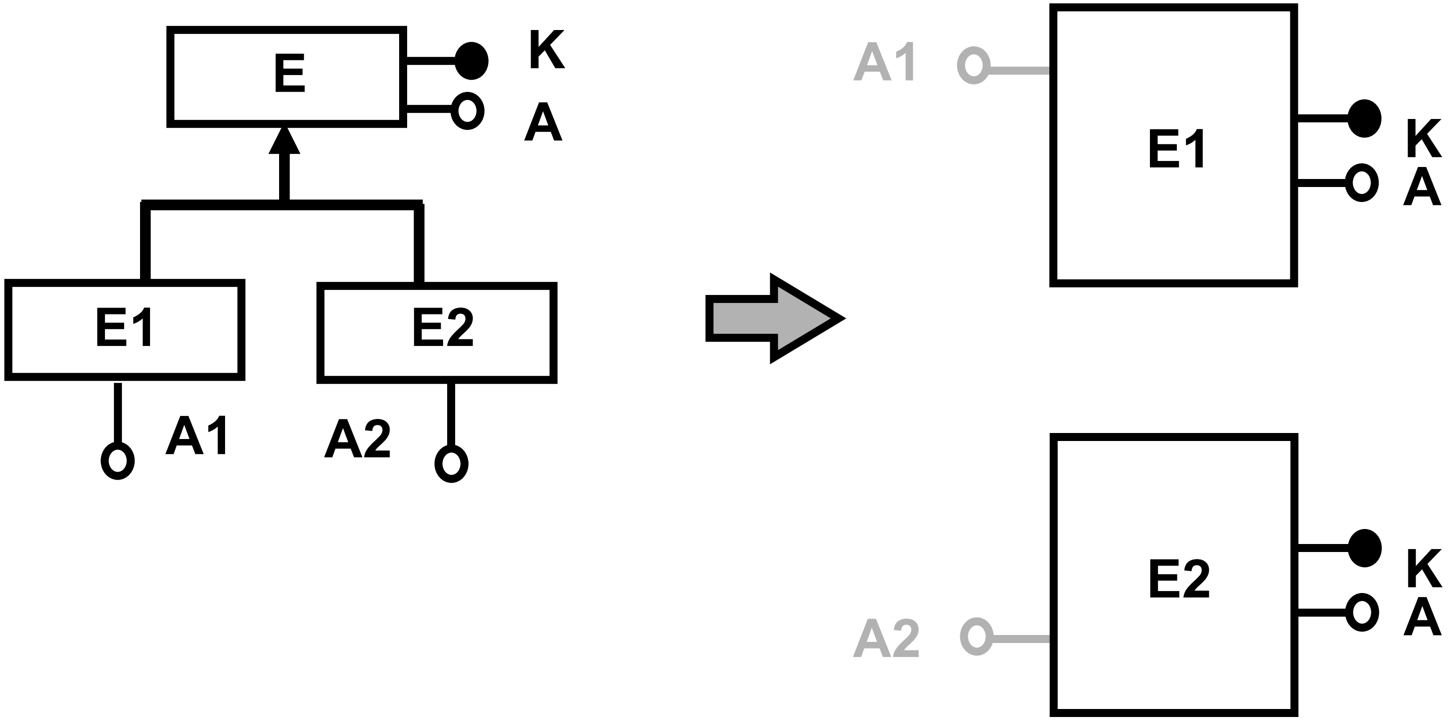
- Horizontal solution: one single entity at the parent level



- Flag: indicates whether an instance is E1 or E2
- Example: male/female

Elimination of hierarchies: option 2

- Vertical solution: two entities corresponding to the children (only if the hierarchy is total)

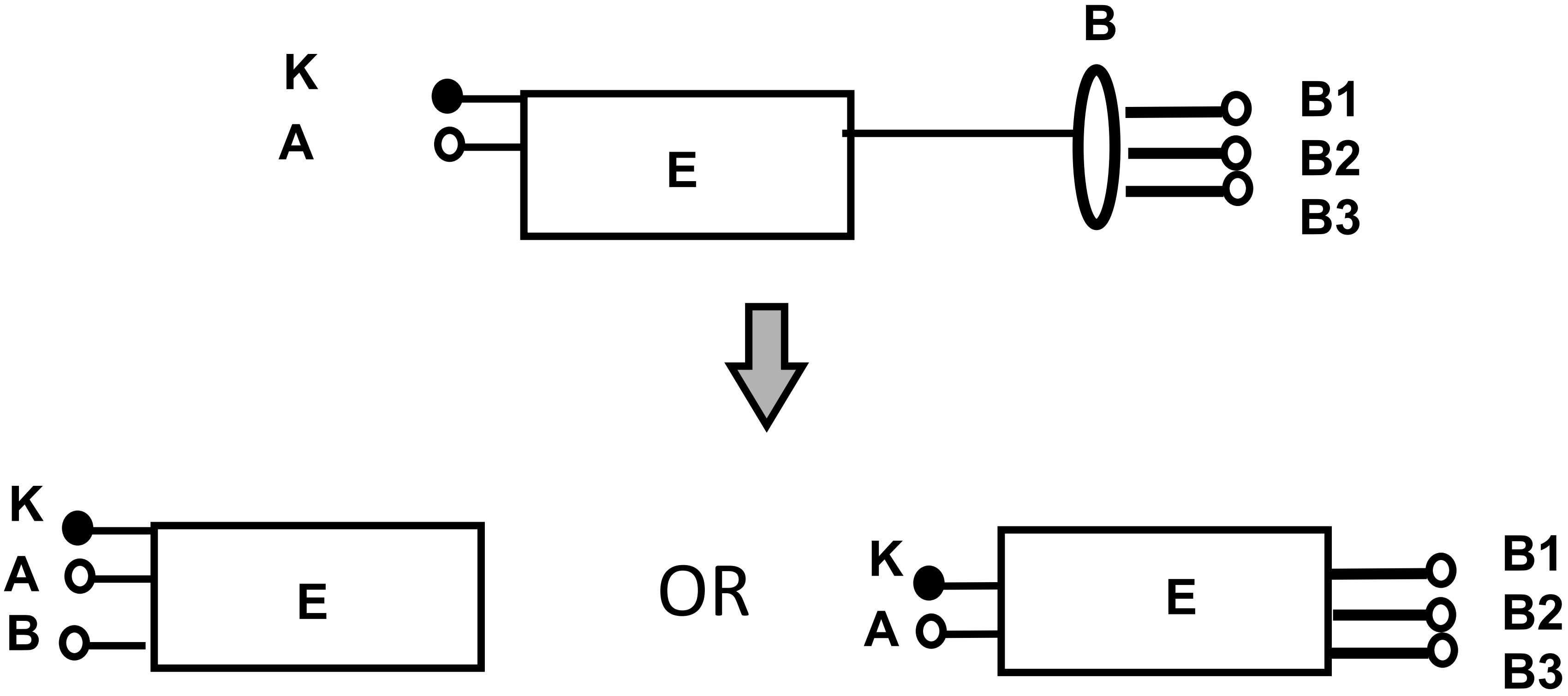


Selection of primary key

- Choosing the most used identifier, as long as it remains simple
- If it’s too complex, one can always introduce a code

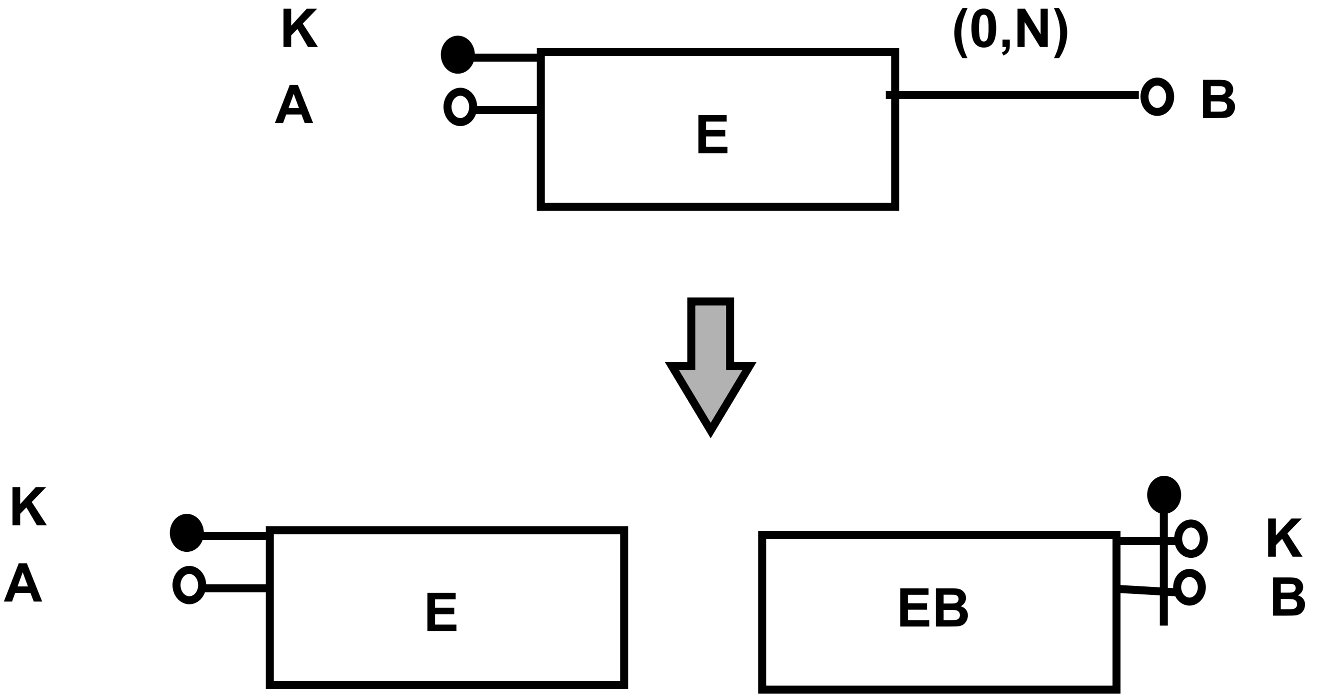
Compound attributes

- Making them simple: one attribute vs several attributes

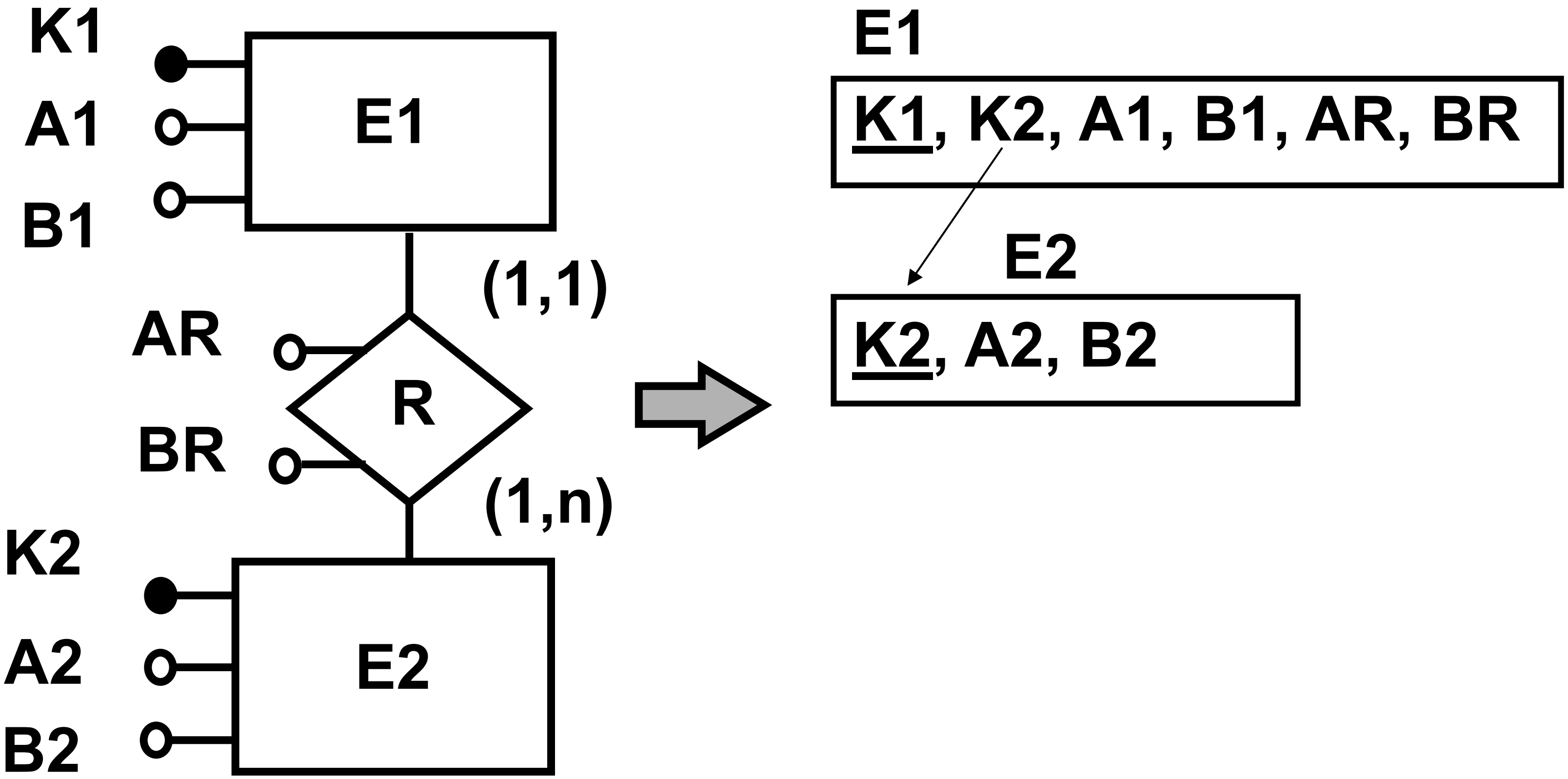


Multiple attributes

- We need a separate entity



Translating one-to-many relationships



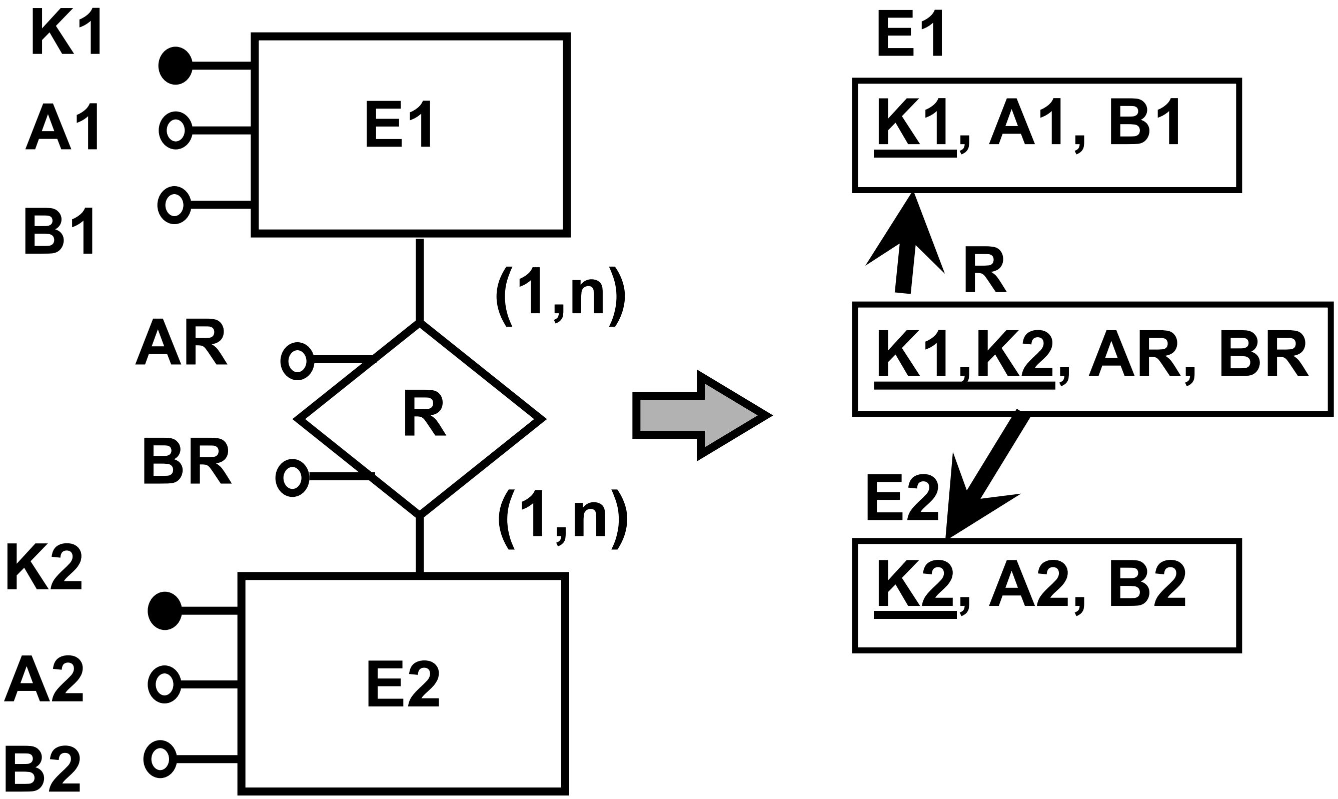
E1

K1, K2, A1, B1, AR, BR

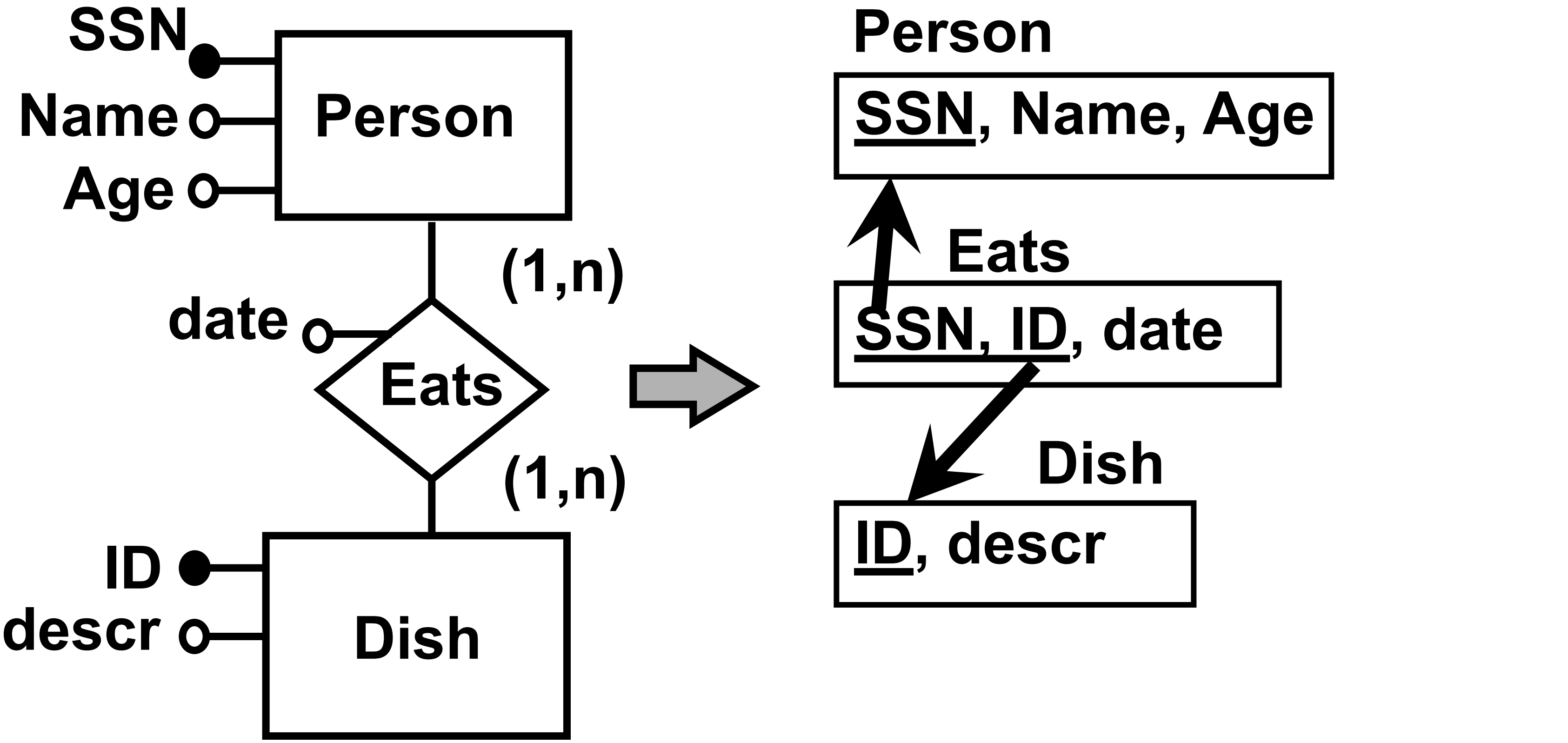
E2

K2, A2, B2

Translating many-to-many relationships



Warning: attributes in many-to-many relationships



SSN

Name

Age

Person

(1,n)

date

Eats

(1,n)

ID

descr

Dish

Person

SSN, Name, Age

Eats

SSN, ID, date

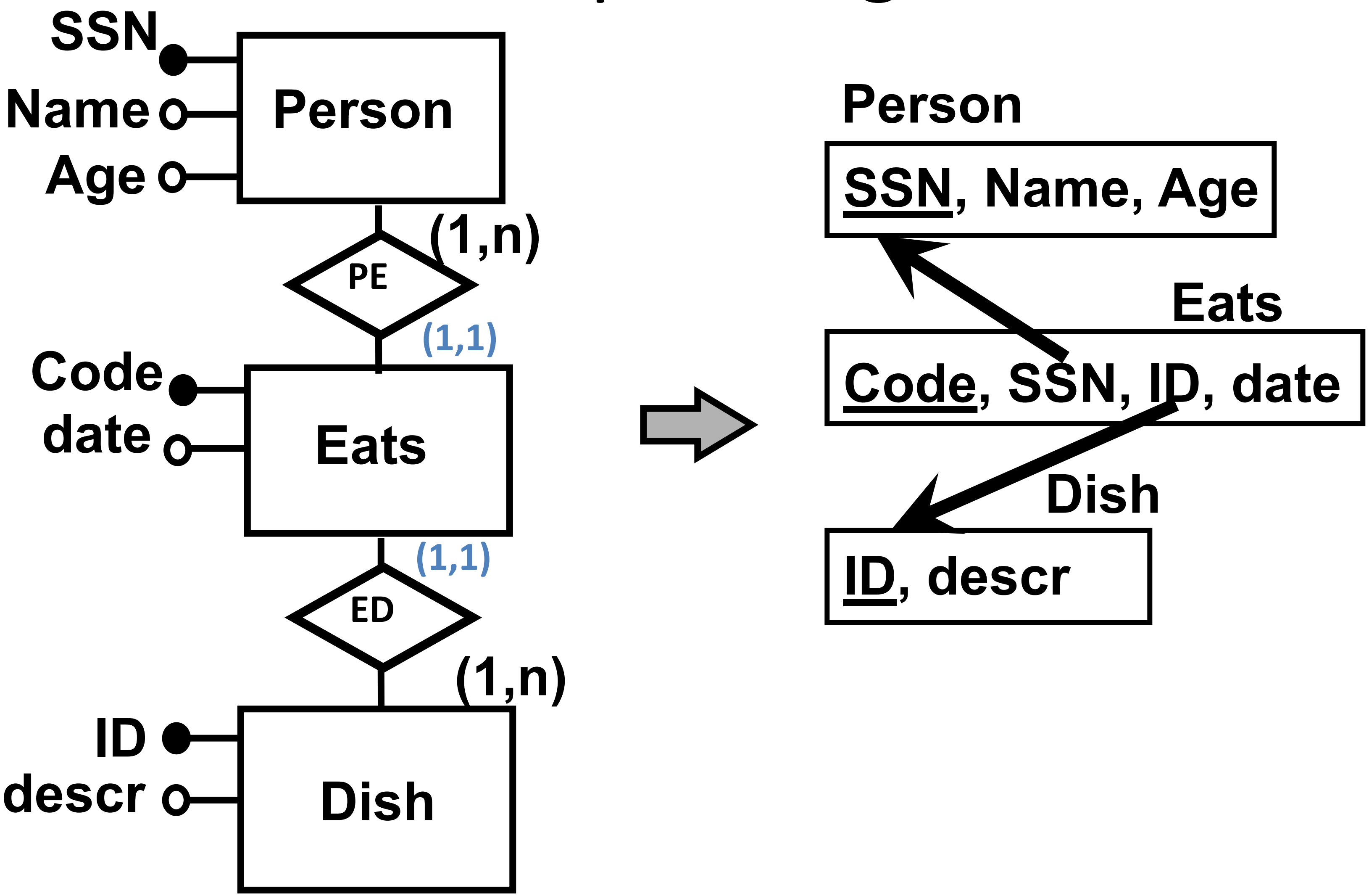
Dish

ID, descr

Correct?

Eats is translated to a relation, and a relation is a set!

Allowing repetitions of the same relationship through reification



Person

SSN, Name, Age

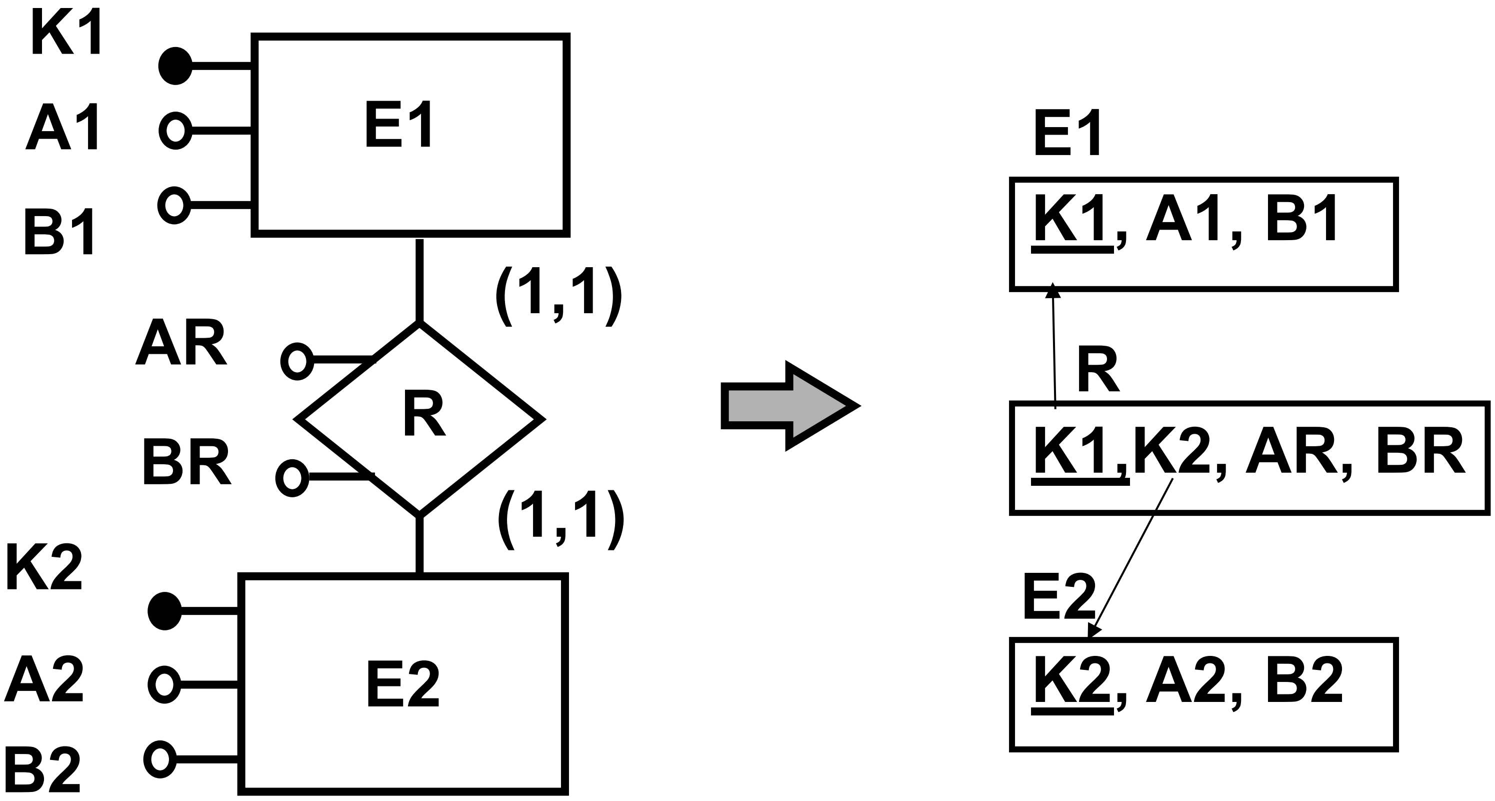
Eats

Code, SSN, ID, date

Dish

ID, descr

Translating one-to-one relationships: option 1



E1

<u>K1</u>, A1, B1

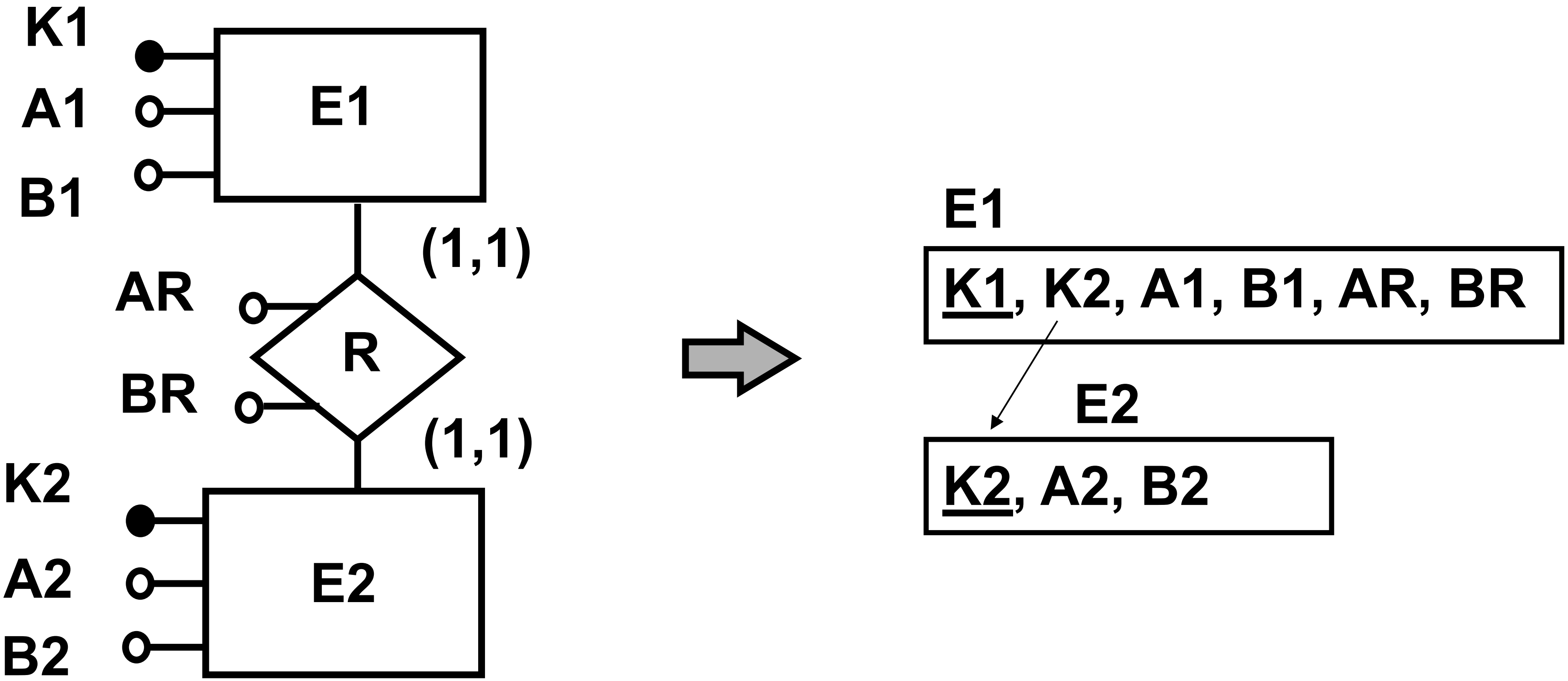
R

<u>K1</u>,K2, AR, BR

E2

<u>K2</u>, A2, B2

Translating one-to-one relationships: option 2



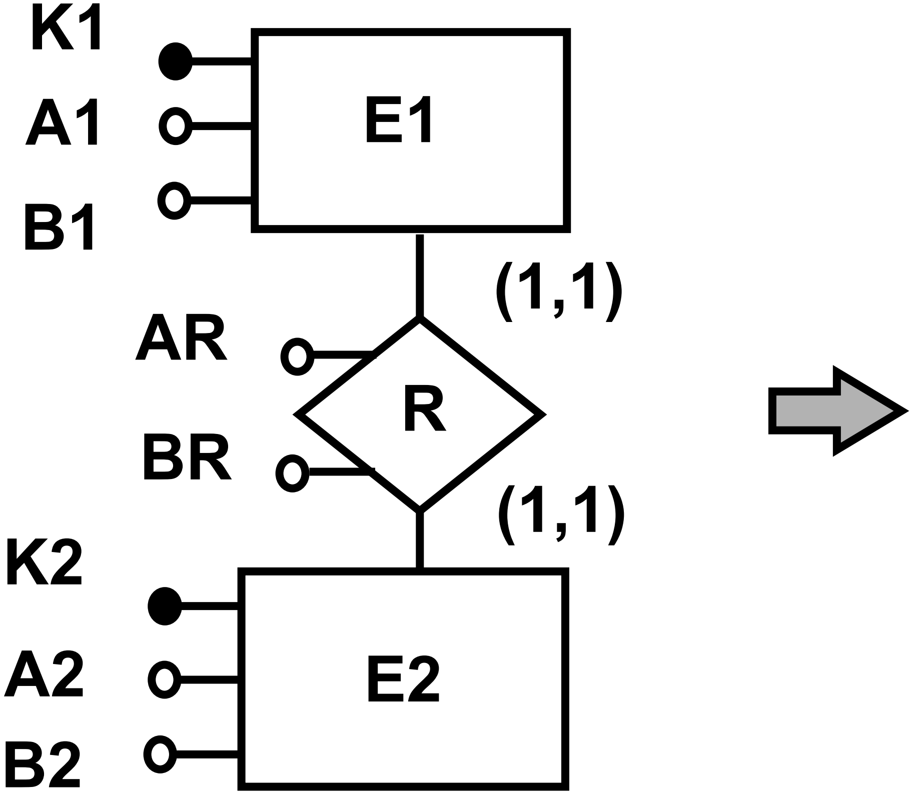
E1

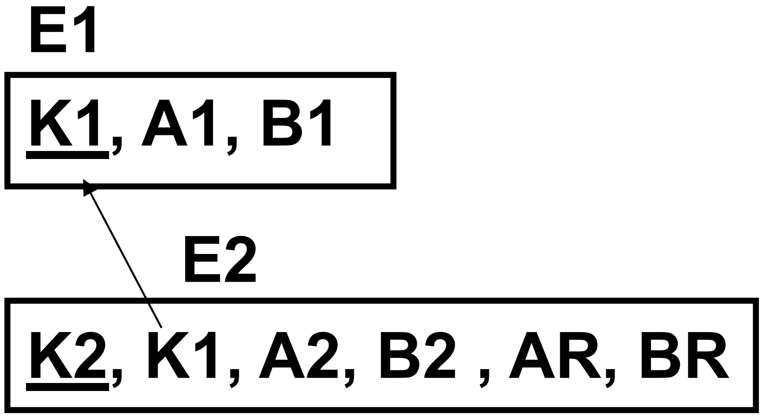
K1, K2, A1, B1, AR, BR

E2

K2, A2, B2

Translating one-to-one relationships: option 3




E1

K1, A1, B1

E2

K2, K1, A2, B2 , AR, BR