UPPSALA
UNIVERSITET

Lectures 2 and 3 – ER and EER Model

1DL301 Database Design I

Jan Kudlicka (jan.kudlicka@it.uu.se)

Fall 2019, Period 2

VOX AULAE

INTENDED LEARNING OUTCOMES

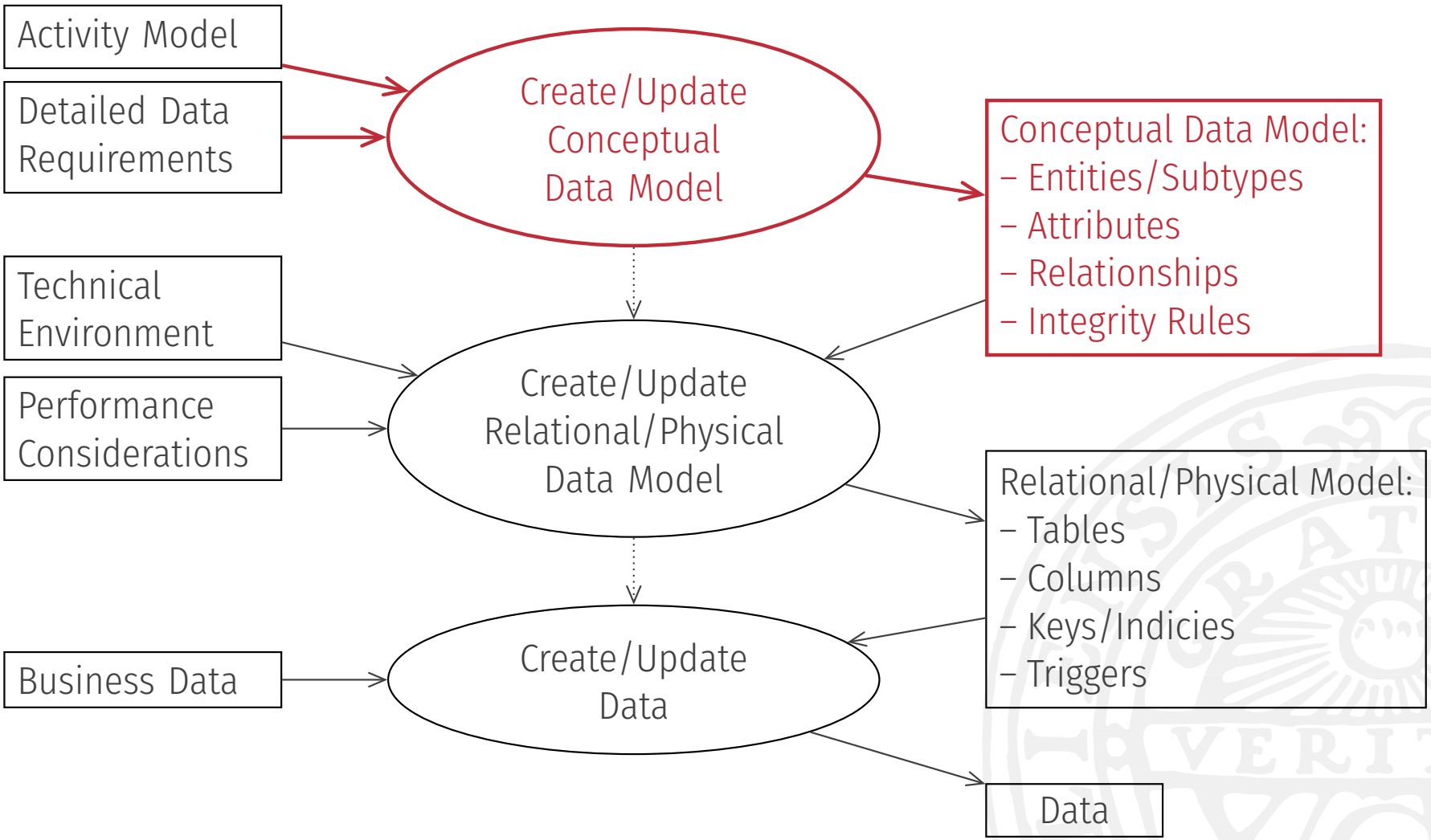
Intended learning outcomes for this and the next lecture

Design a conceptual data model based on textual requirements.

To understand and to be able to use an entity-relationship (ER) constructs:

- Entities and their attributes, keys
- Relationships and their attributes, cardinalities and roles
- Subtypes

WHERE WE ARE



Source: EPISTLE, modified

WHY CONCEPTUAL MODEL?

- Comparing to the natural language:
  - More formal
  - Not ambiguous
- Comparing to a relational model (Lecture 4):
  - Neither implementation nor (R)DBMS specific details
- High-level description of the mini world (a part of the real world which we wish to model).
- Easy to understand by people without technical background.
- Can be used as the documentation.
- ER models can be easily transformed to a relational model (and then implemented in a relational database).

HISTORY

Proposed in:

CHEN, Peter Pin-Shan. **The entity-relationship model – toward a unified view of data**. ACM Transactions on Database Systems (TODS), 1976, 1.1: 9-36.

Link to the paper

THREE MAIN COMPONENTS

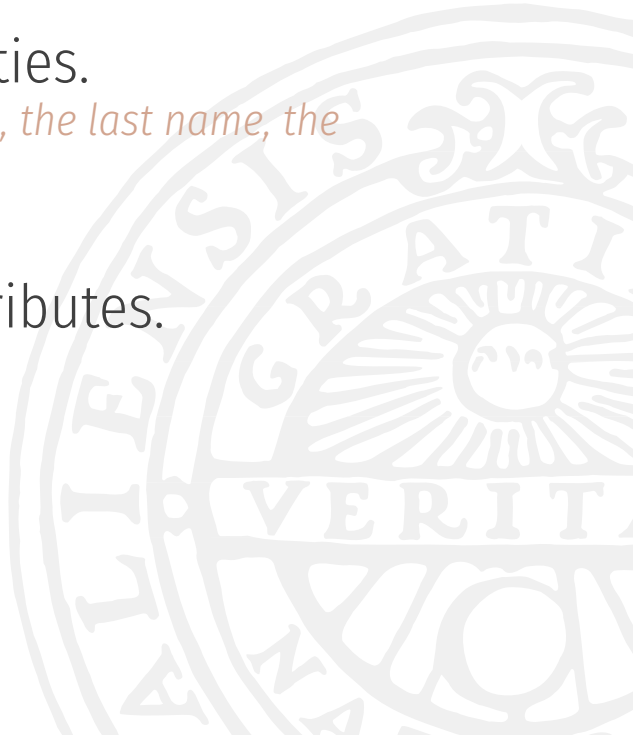
- Entities
- Attributes
- Relationships

ENTITY TYPE AND ENTITY

Entity – a “thing” or “object” in the mini world (for which we want to store some data) that is capable of an independent existence.
Examples of entities in an university IT system: each student, teacher, course, department, building, classroom, etc.

Attribute – a (relevant) property or feature of entities.
Examples of attributes for a teacher: the personal no., the first name, the last name, the birthdate, etc.

Entity type – a set of entities having the same attributes.
Entity type Student is a set of all students (all entities).

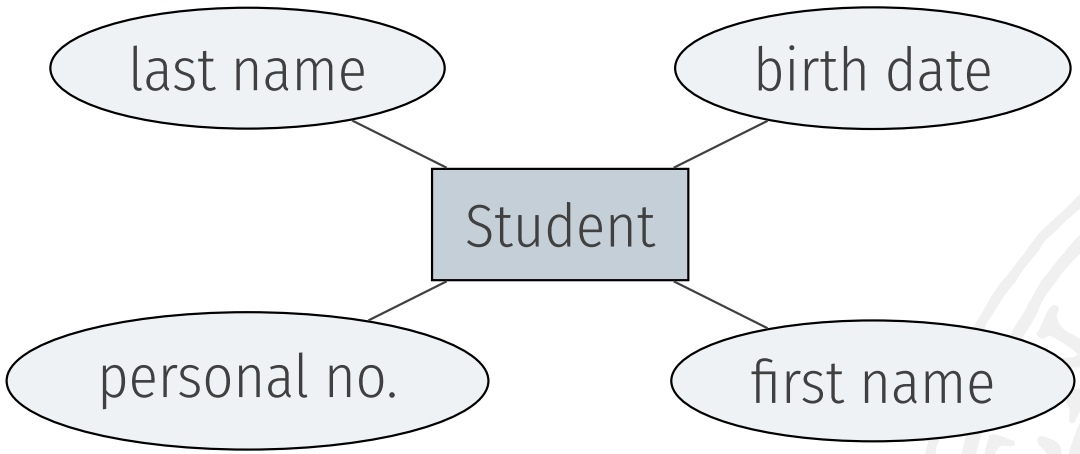


ENTITY TYPE AND ENTITY, CONT'D

- Entities do not have to exist physically.
  E. g. a department, a course or an order in an online store.
- It must be possible to distinguish entities from each other, i. e. each entity must have some identity.
  E. g. personal no. for teachers or students, combination of building name and room no. for classrooms.
- If you are familiar with object-oriented programming: Entity type corresponds to a class and entities to instances of this class.

GRAPHICAL NOTATION FOR ENTITY TYPES AND ATTRIBUTES

In the ER diagram we use rectangles for entity types and ovals for attributes. The attributes are connected to the entity type by lines.



ATTRIBUTE VALUE AND DOMAIN

Attribute value – value which an entity has for a given attribute.

The code (attribute) of the Database Design I course (entity) is 1DL301 (attribute value).

Attribute values are elements from a **domain** – a set of all possible values for the given attribute.

The domain of "result" in an experiment with a die is {1, 2, 3, 4, 5, 6}.

Very often we represent domains by datatypes, e.g., the domain of all real numbers is represented by the datatype *float*.

TYPES OF ATTRIBUTES

- Simple – each entity has a single atomic (= indivisible) value.

  E. g. "age", "first name" (although it can be split into letters, these letters do not have any meaning on their own).

- Composite (or compound) – the attribute can be divided into several components.

  E. g. "name" consists of the first, middle and last name, "address" consists of the street name, the house no., the city and the postal no.

- Multi-valued – the attribute might have multiple values.

  E. g. "email address" for a person or "keyword" for a blog post.

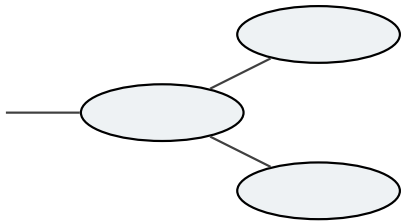
- Derived – the value of the attribute can be determined from the other attributes.

  E. g. "age" if there is an attribute for the birth date.

GRAPHICAL NOTATION FOR DIFFERENT TYPES OF ATTRIBUTES

Simple

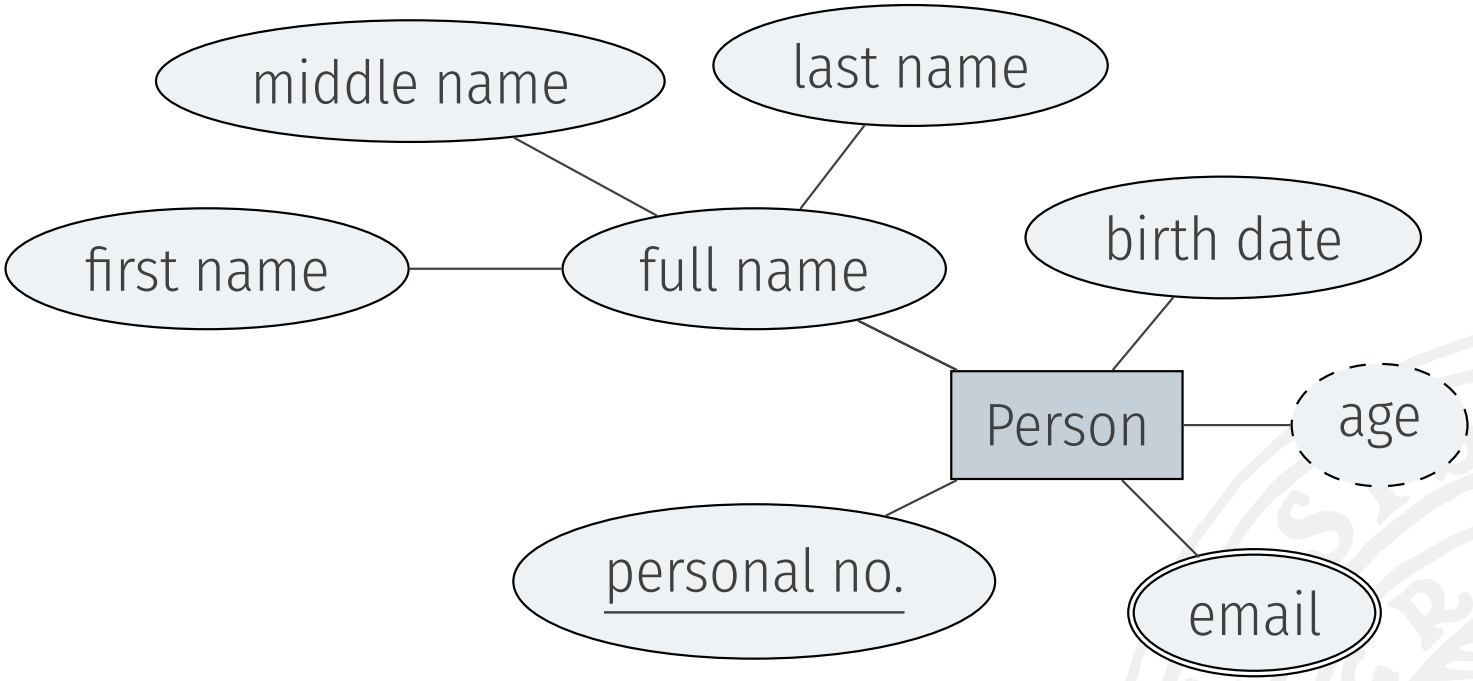
Composite



Multi-valued

Derived

AN EXAMPLE WITH SEVERAL TYPES OF ATTRIBUTES



RELATIONSHIP TYPE AND RELATIONSHIP

Relationship – a relation between entities of given entity types.

Note: Distinguish between a relation (in the meaning of a table) and a relation(ship) between entities.

Jan Kudlicka (entity of type Teacher) teaches (relationship) the Database Design I course (entity of type Course).

Johan Johansson (entity of type Student) takes (relationship) the Database Design I course (entity of type Course).

Relationship type – a set of all relations between members of given entity types.

RELATIONSHIP TYPE AND RELATIONSHIP, CONT'D

Types of relationships

- Binary – involving 2 entity types (most common)
- Ternary – involving 3 entity types
- N-ary – involving N entity types

Mathematically, a binary relationship type R between entity types E_1 and E_2 is a subset of the cartesian product of E_1 and E_2, i. e. $R \subset E_1 \times E_2$.

GRAPHICAL NOTATION FOR RELATIONSHIP TYPES

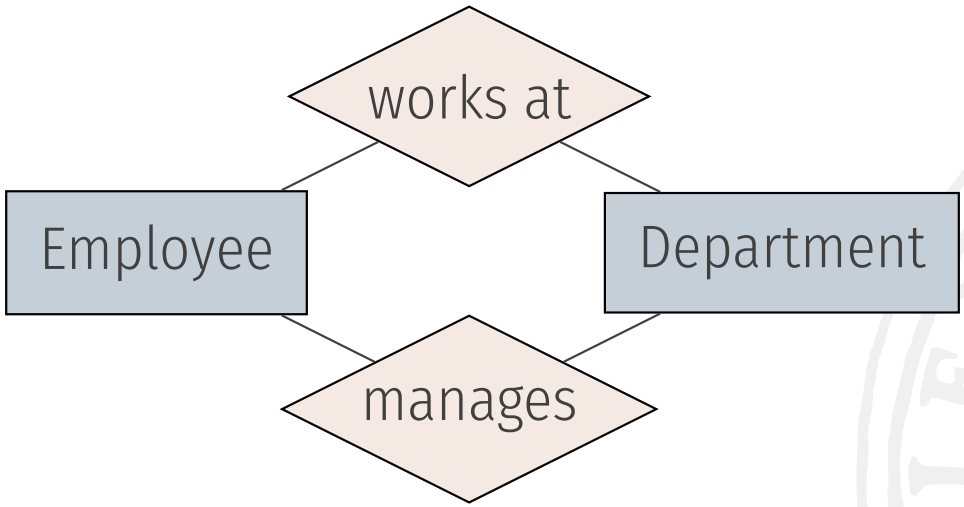
A diamond with lines to the involved entity types.

SPECIAL CASES

Relationships might also exist between entities of the same type.

Homer Simpson (entity of type Person) is a father (relationship) of Lisa Simpson (entity of type Person).

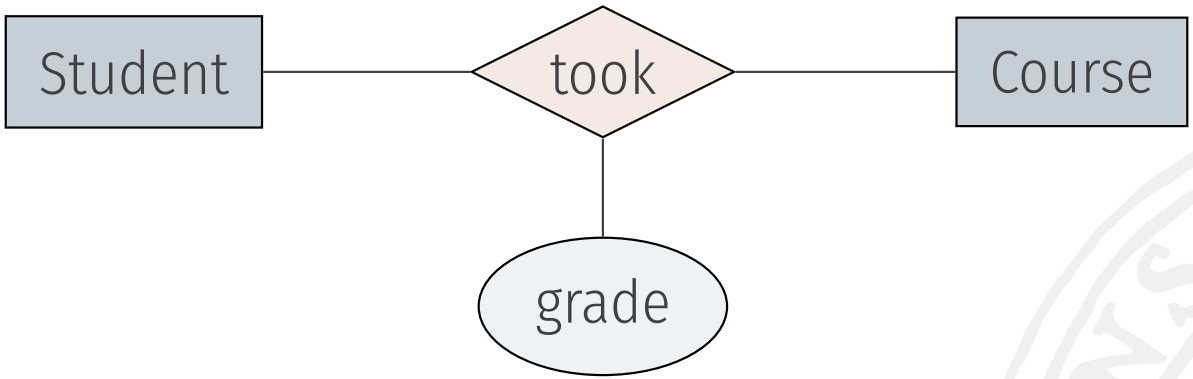
There might be several relationships between the same entity types.



RELATIONSHIP ATTRIBUTES

Relationships might have attributes too.

Example: Johan Johansson (entity of type Student) took (relationship) the Database Design I course (entity of type Course) and got a grade of 4 (attribute of the relationship). Note that the grade is not an attribute of a student, nor of a a course.



Example: Uppsala stadsteater (entity of type Theater) plays (relationship) Hamlet (entity of type Play) this Saturday at 7pm (attribute of the relationship).

RULES OF THUMB

Chen proposed the following “rules of thumb” for mapping natural language descriptions into ER diagrams:

English grammar structure	ER structure
Common noun	Entity type
Proper noun	Entity
Transitive verb	Relationship type
Intransitive verb	Attribute
Adjective	Attribute for entity
Adverb	Attribute for relationship

DISCUSSION: ATTRIBUTE OR ENTITY?

We want to store information about students and their names, surnames, birth dates and study programs.

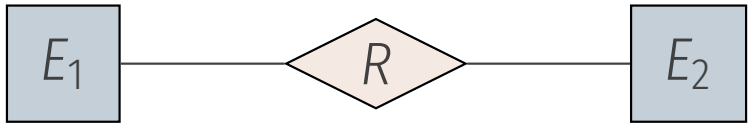
We create entity type Student with attributes Name, Surname and Birth date.

What about the study program?

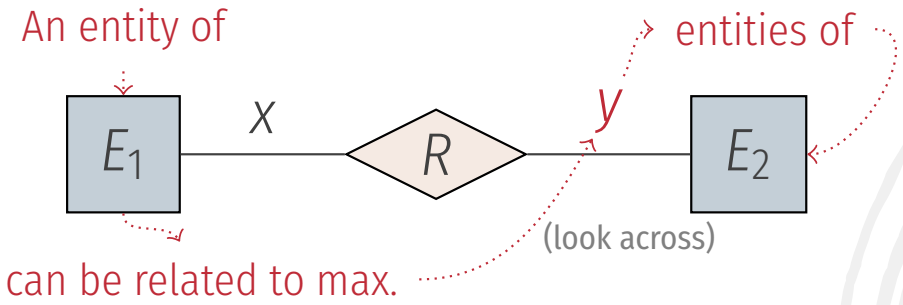
The difference between an attribute and an entity is sometimes subjective.

Or we want to store information about cars and their make (brand), model, max. speed and color. We create entity type Car with attribute Max speed. But what about the make, model and color?

CARDINALITIES



- Can any E_1 entity be related to more than one E_2 entity?
- Can any E_2 entity be related to more than one E_1 entity?



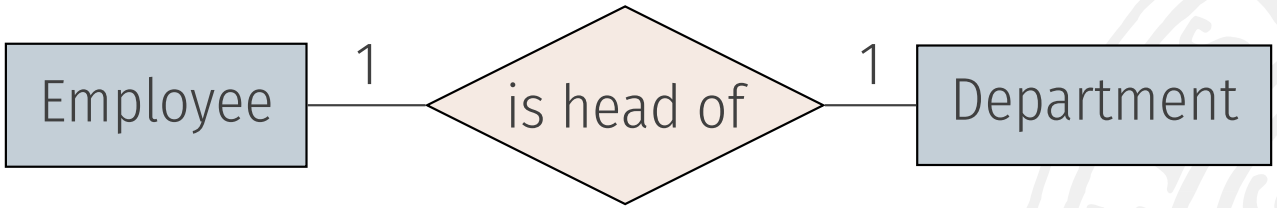
Values x and y are usually 1 (= max one) or N (= many).

TYPES OF RELATIONSHIPS BASED ON CARDINALITIES

Three different types of relationships based on cardinalities:

- One-to-one (1:1)
- One-to-many (1:N)
- Many-to-many (N:N)

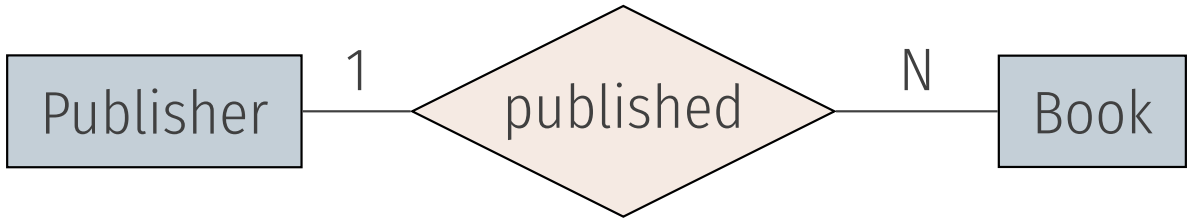
One-to-one (1:1)



No employee can be head of more than one department and no department can have more than one head.

TYPES OF RELATIONSHIPS BASED ON CARDINALITIES, CONT'D

One-to-many (1:N)

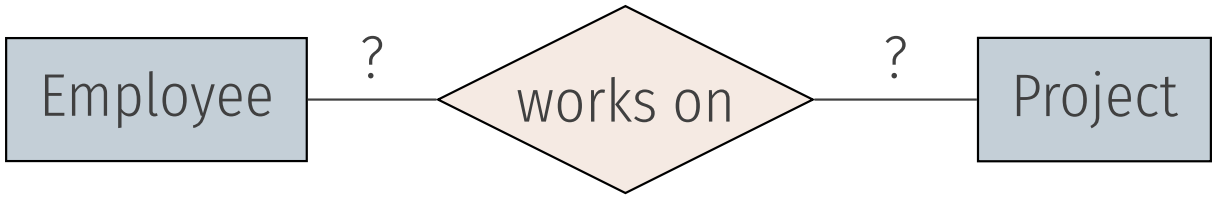


A publisher can publish many books, but no books has more than one publisher.

Many-to-many (N:N)

An author can write many books and a book may have many authors.

EXERCISE 1



Assume that an employee can work on several projects, and that a project can have several employees working on it.

1. 1:1
2. 1:N
3. N:1
4. N:N

EXERCISE 2

Assume that no employee will manage more than one department and that each department can only have one manager.

1. 1:1
2. 1:N
3. N:1
4. N:N

TODAY'S PROBLEM

The Internet Movie Database (IMDb) is an online database of information related to movies:

- Each movie has a title, release year, length in minutes, production company, one or more directors, one or more writers and one or more actors.
- Each actor, writer or director has a name (consisting of first, middle and last name), date and place of birth and a short biography.
- For each movie, IMDb shows also names of the characters being played by actors (e.g., Bruce Willis played character of John McClane in Die Hard). Assume that no actor plays several characters in one movie.

PARTICIPATION

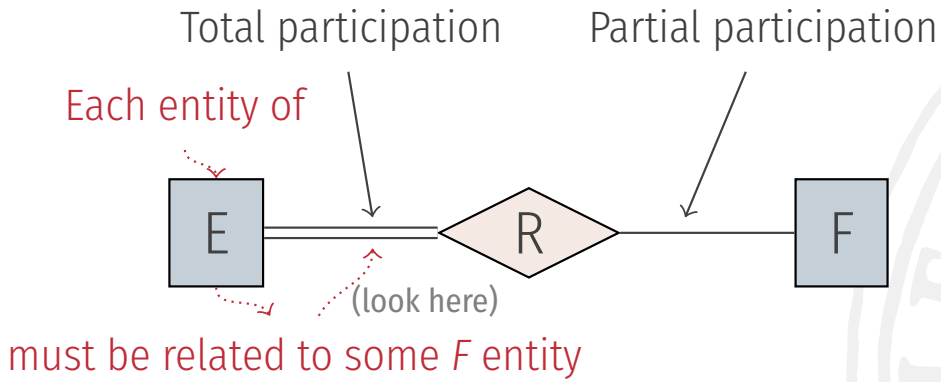
- **Total participation** (mandatory participation) – each entity must be related to at least one entity of the other type.

  A book has to have at least one author.

- **Partial participation** (optional participation) – an entity does not have to be related to any entity of the other type.

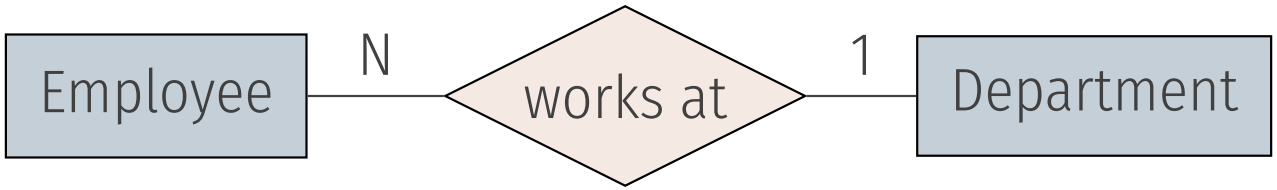
  An employee does not have to be a manager of any project.

Notation:



Note the “look here” notation for the participation!

EXERCISE 3



Assume that each employee must work at some department and each department must have some employees.

1.


2.


3.


4.


EXERCISE 4

Assume that not every employee is a manager, but every department has a manager.

1.

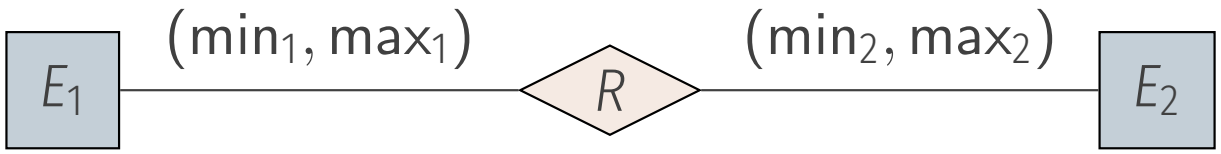

2.


3.


4.


MIN-MAX NOTATION

There are (unfortunately) many alternative notations for cardinality and participation. The min-max notation is used quite often:



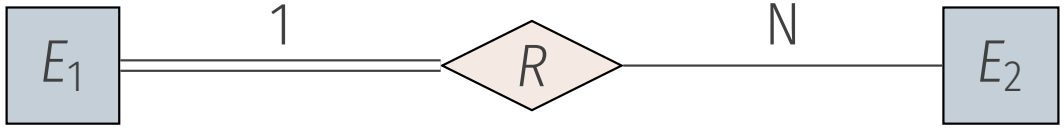
A "look here" notation (for both cardinality and participation).

$(\mathbf{min}_1, \mathbf{max}_1)$ – limits on how many times each E_1 entity can participate in R (= how many related E_2 entities).

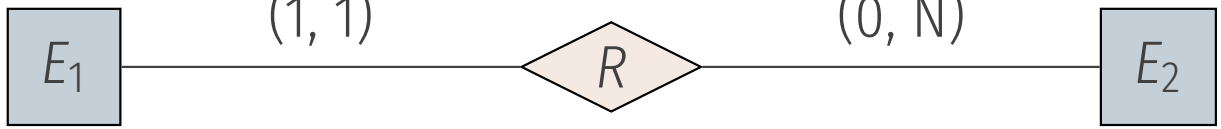
Min must be a concrete number (usually 0 or 1).

Max might be a concrete number or N (some also use M or *) for no limit (usually 0 or N).

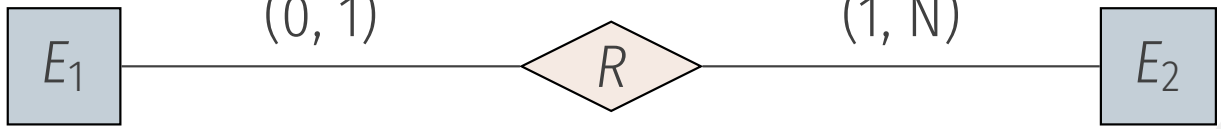
EXERCISE 5



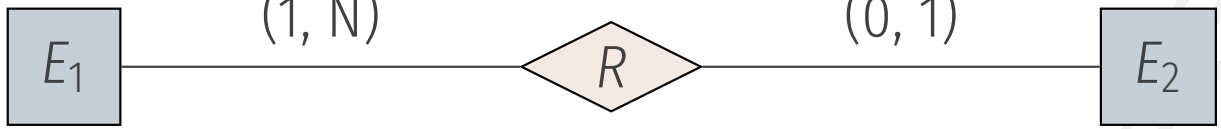
1.



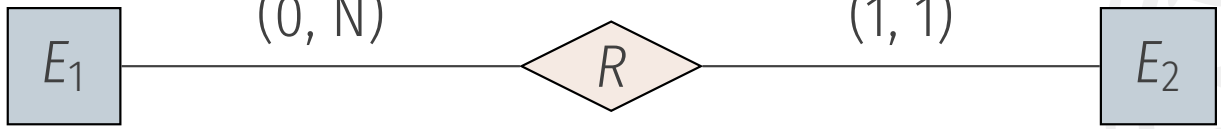
2.



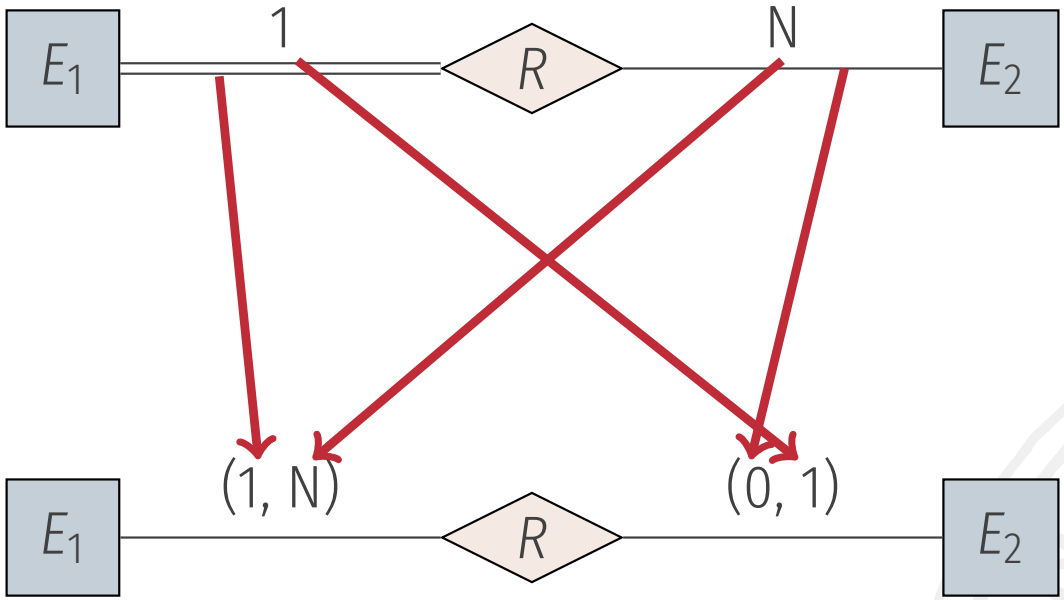
3.



4.



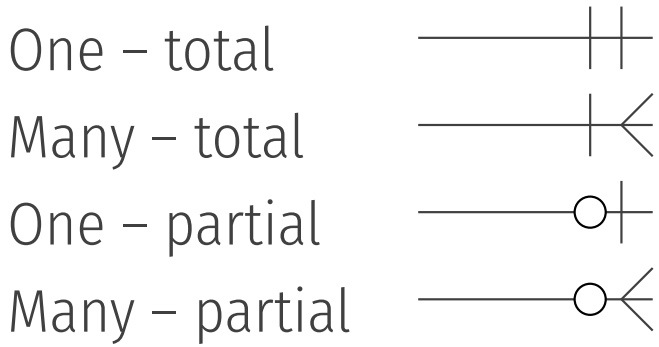
TRANSLATION TO THE MIN-MAX NOTATION

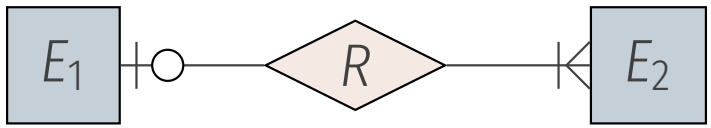


CROW'S FOOT NOTATION

ERDPlus uses yet another notation: the crow's foot notation.

A "look-across" notation for both cardinality and participation.



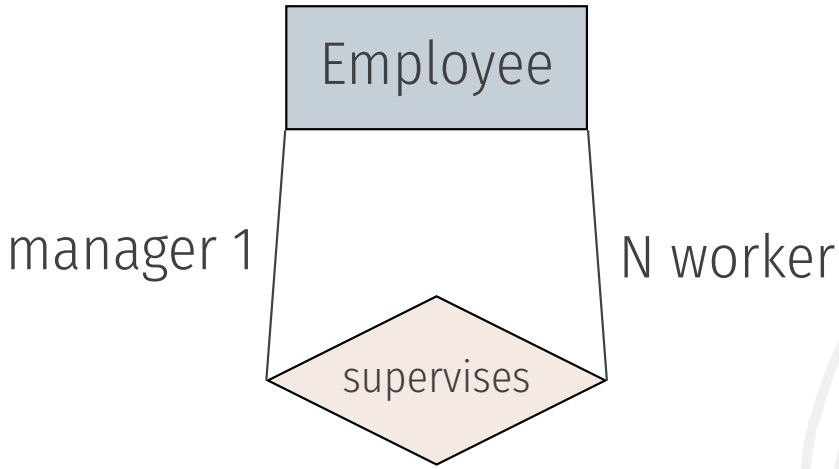


An E_1 entity must be related at least to one E_2 entity, an E_2 entity might be related to one E_1 entity.

ROLES

A **role** describes what function an entity has in a relationship.

Usually not specified, except for relationships between entities of the same type:

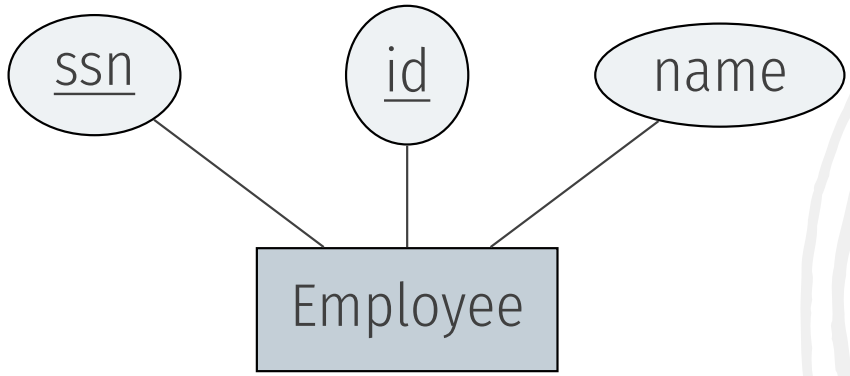


KEY ATTRIBUTES

Key attribute – an attribute that uniquely identifies entities of a given type (i.e., each entity has a unique value of this attribute).

Several attributes together might be used as a key (with the help of a composite attribute).

Each (strong) entity type has to have at least one key. If none of the attributes can be used as a (natural) key, you can add a "surrogate" key attribute (auto-incremented key or UUID).



WEAK ENTITY TYPES

Weak entities – entities that cannot be uniquely identified by its own attributes and must rely on the identity of an "owner" entity.

Examples: invoice items on an invoice, monthly payments of a loan, rooms in a building, departments in an organization.

A relationship to the owner entity is called the **identifying relationship** and it might only be 1:1 or 1:N.

In the 1:N case, the weak entity type needs a **partial key** – an attribute which together with the owner's key can uniquely identify the weak entities.

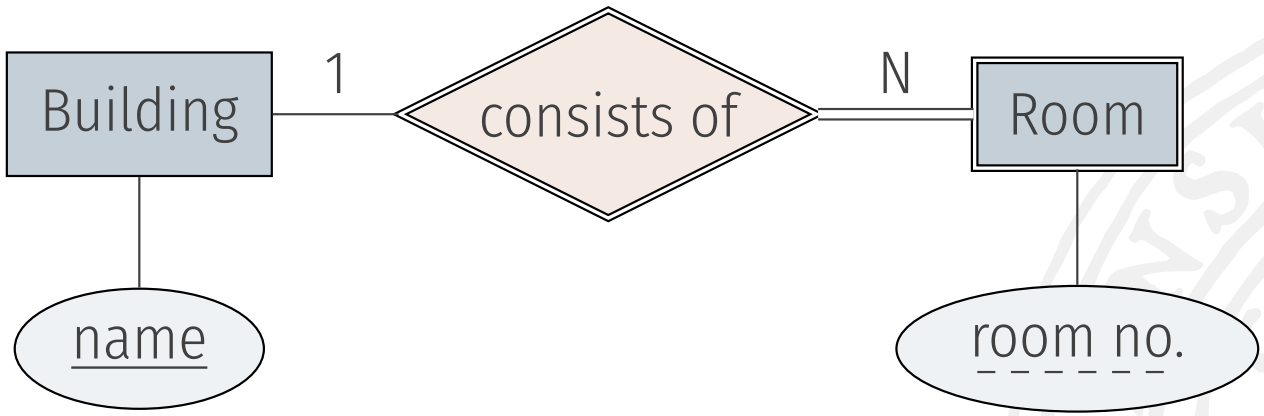
(Weak entities can be owners of other weak entities.)

GRAPHICAL NOTATION FOR WEAK ENTITIES

Weak entity type is represented by a double-lined rectangle.

Identifying relationship is represented by a double-lined diamond.

Partial key is underlined using a dashed line.



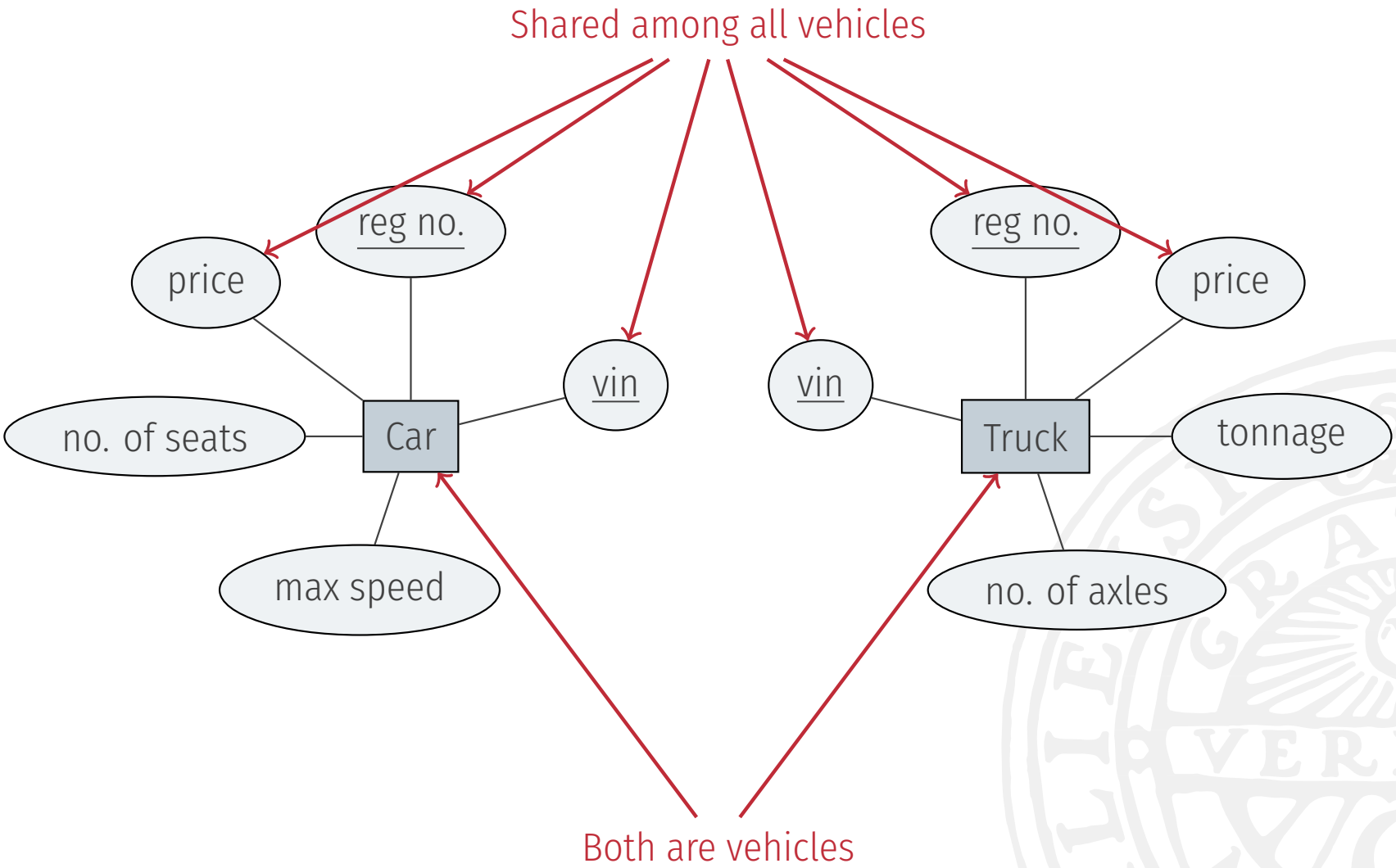
EXTENDED ENTITY-RELATIONSHIP (EER) MODEL

Extended entity-relationship (EER) model extends ER with more complex concepts.

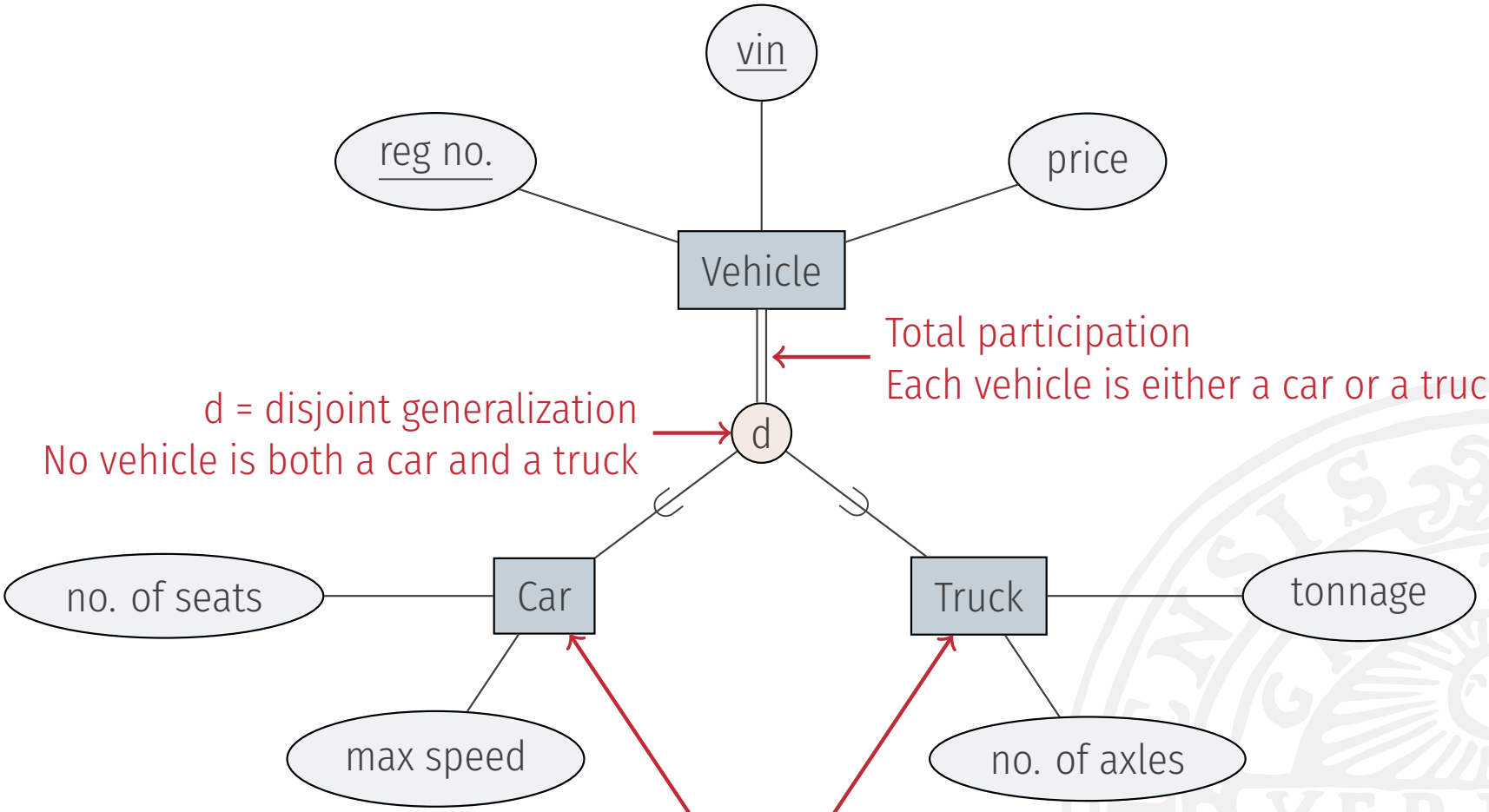
We will only cover the most important concept: **supertype / subtype** (superclass / subclass; is-a relationship).

Same idea as the class inheritance in object-oriented programming.

MOTIVATING EXAMPLE

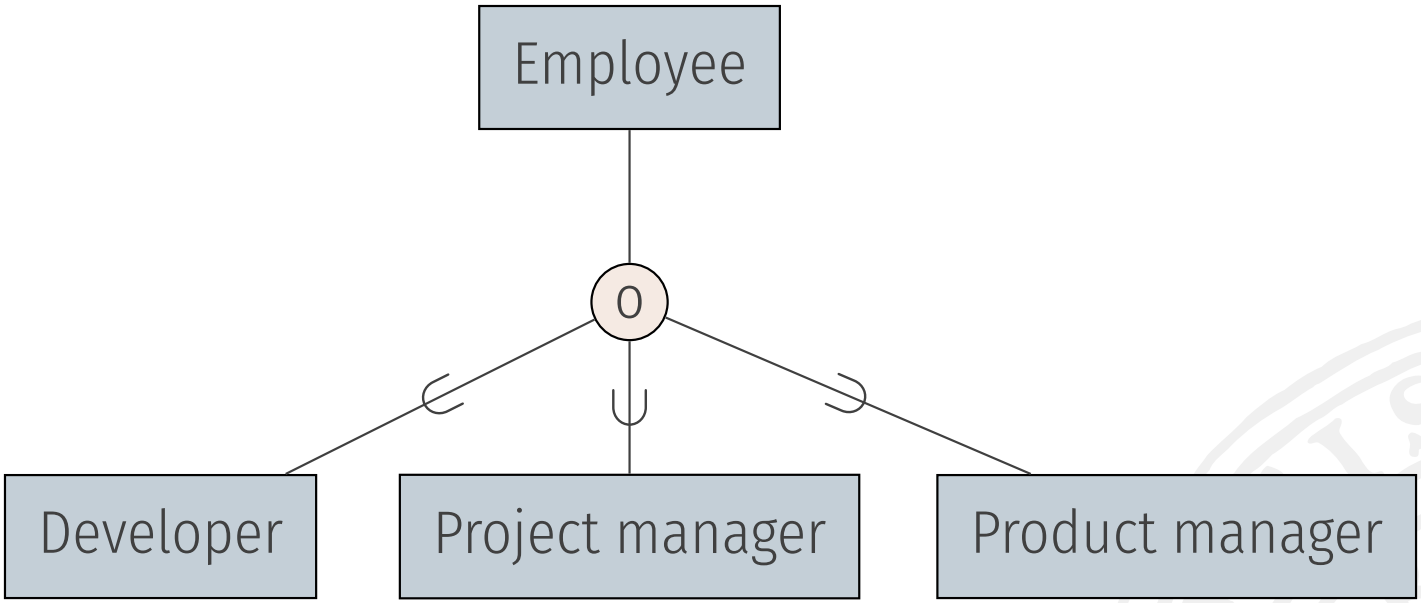


MOTIVATING EXAMPLE, CONT'D



OVERLAPPING VS. DISJOINT SUBTYPES

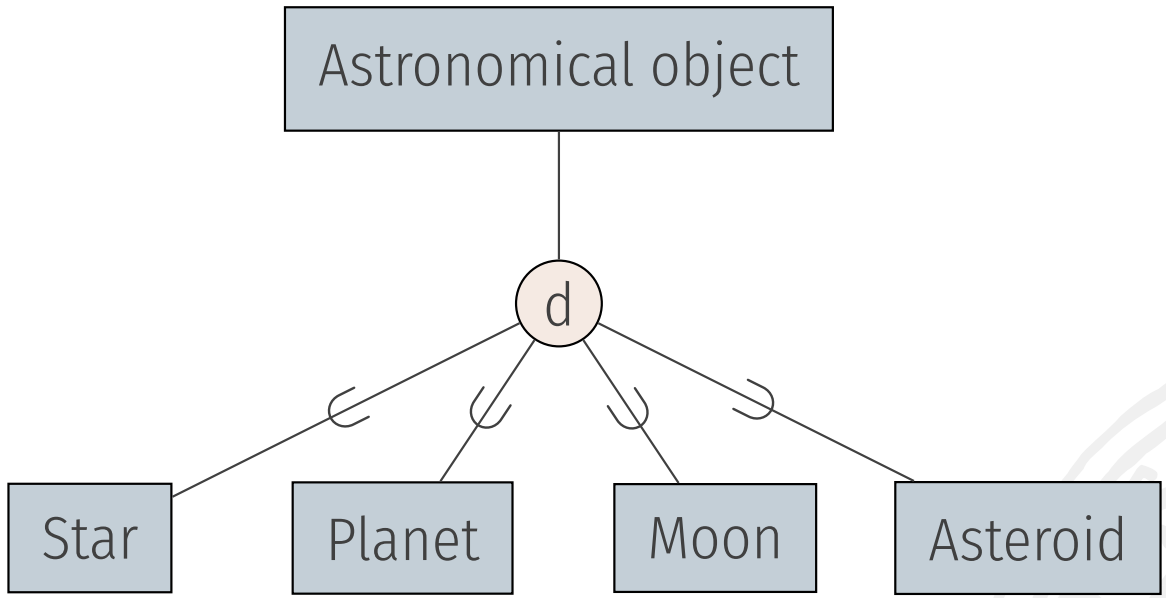
Overlapping subtypes (o)



An employee can be a developer, a project manager, a product manager, or **several of these subtypes** at the same time.

OVERLAPPING VS. DISJOINT SUBTYPES

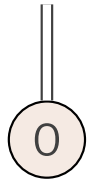
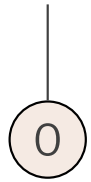
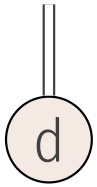
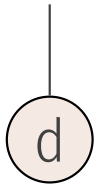
Disjoint subtypes (d)



An astronomical object can be either a star, a planet, a moon or an asteroid, but it **cannot be several of these subtypes** at the same time.

SUMMARY

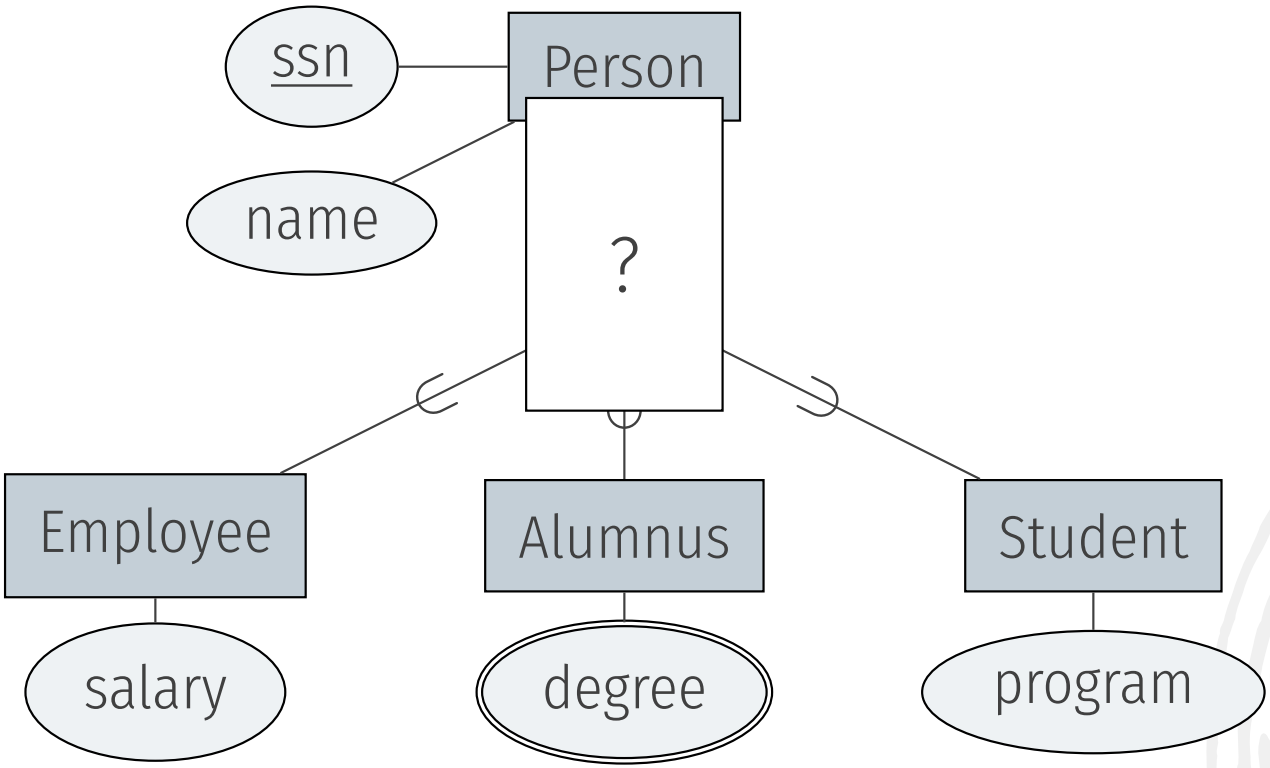
Four different combinations:

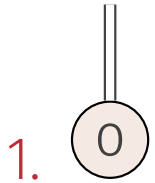
	Total	Partial
Overlapping		
Disjoint		

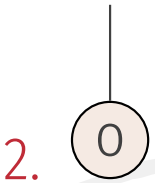
Adding a set of subtypes of an entity type is called **specialization**.
Creating a supertype for a set of entity types is called **generalization** (ref. the motivating example).

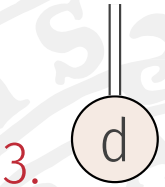
EXERCISE 6

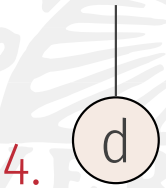
A university database



1. 

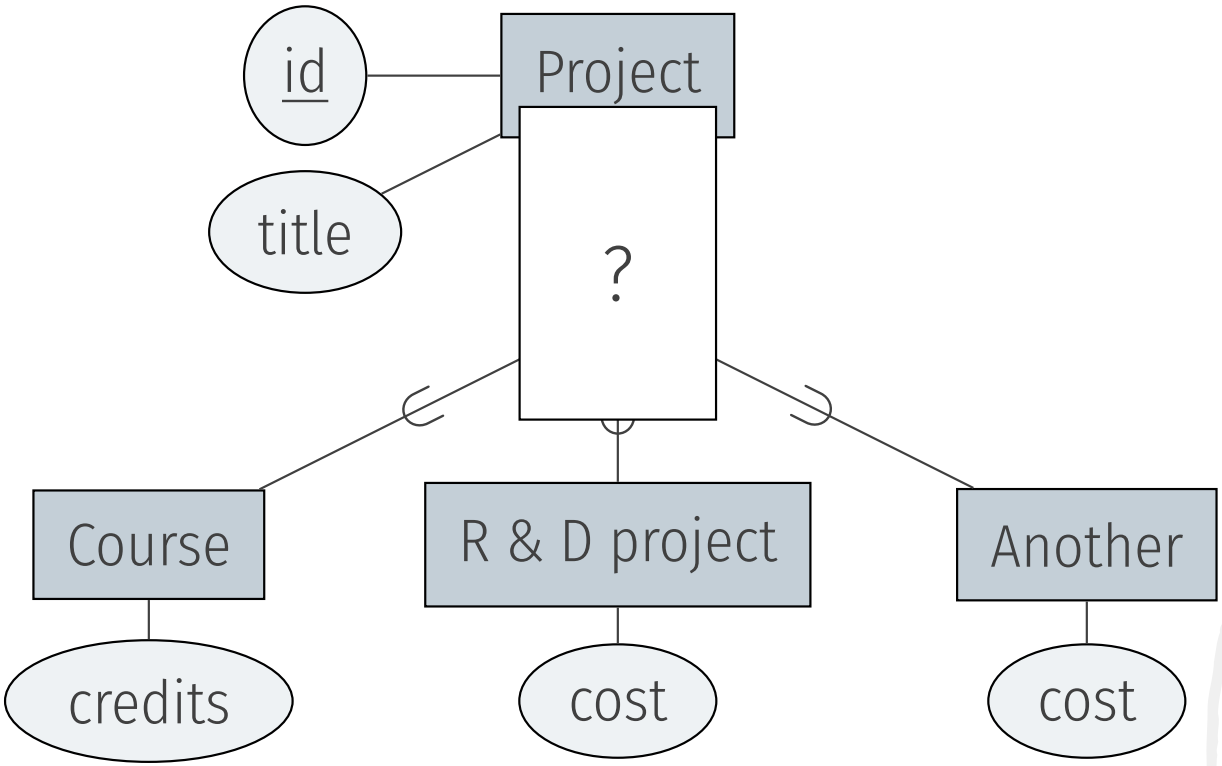
2. 

3. 

4. 

EXERCISE 7

A university database



1. o
2. o
3. d
4. d

SUMMARY OF THE GRAPHICAL NOTATION

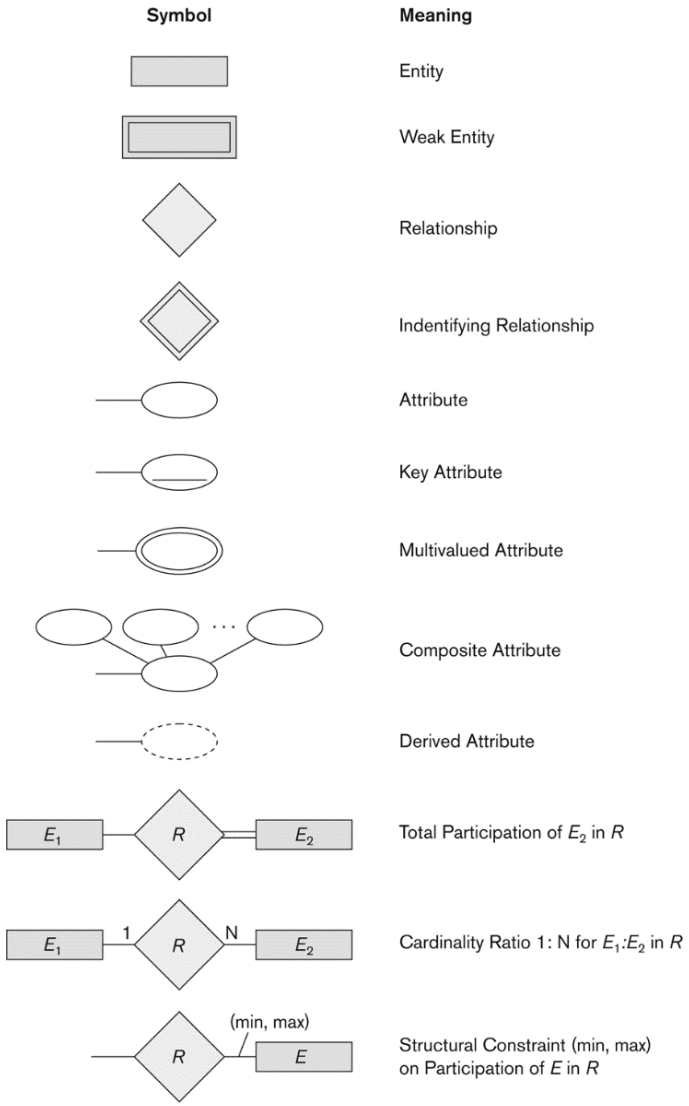


Figure by Ramez Elmasri, Shamkant B. Navathe

ALTERNATIVE NOTATIONS

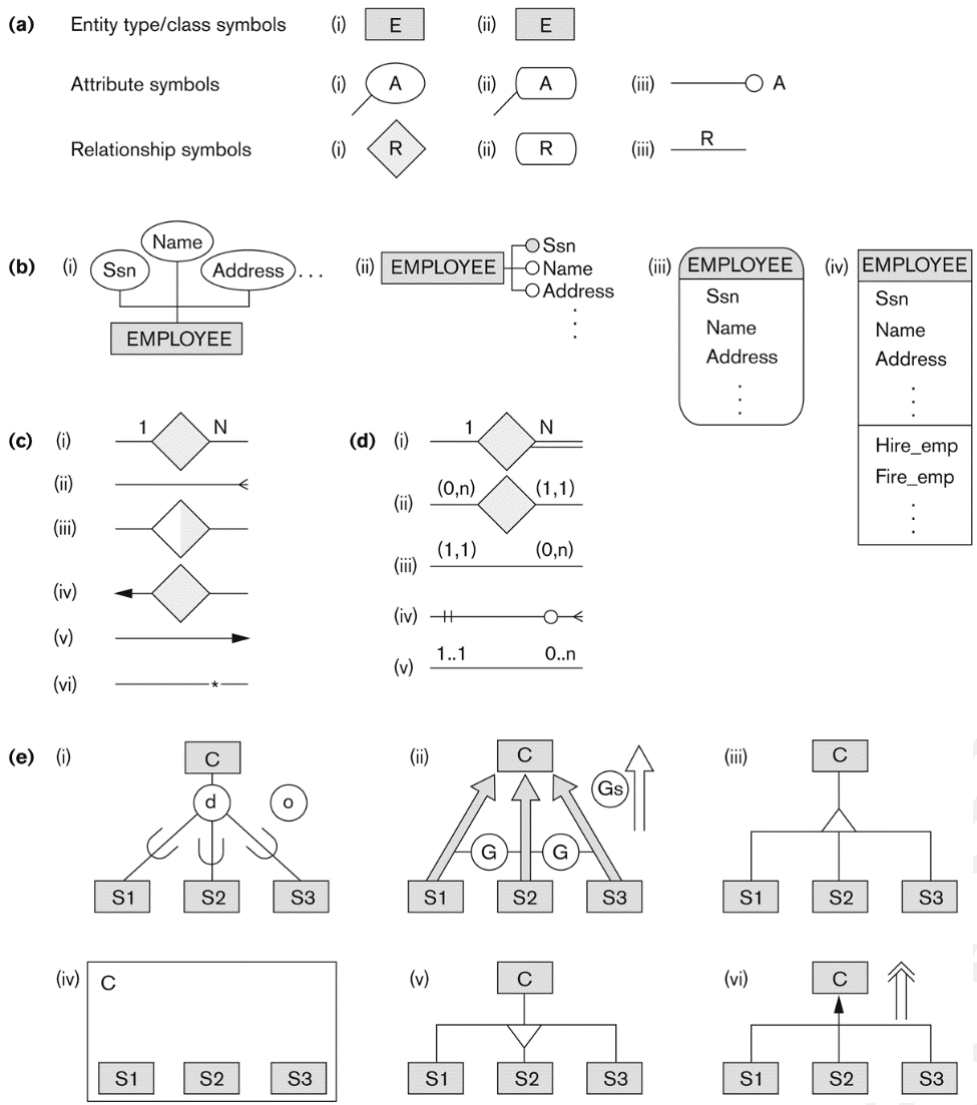


Figure by Ramez Elmasri, Shamkant B. Navathe