Chapter 2: ER-Diagrams

Content:

- Learn how to draw ER diagrams
- Useful to model a database

Database Design

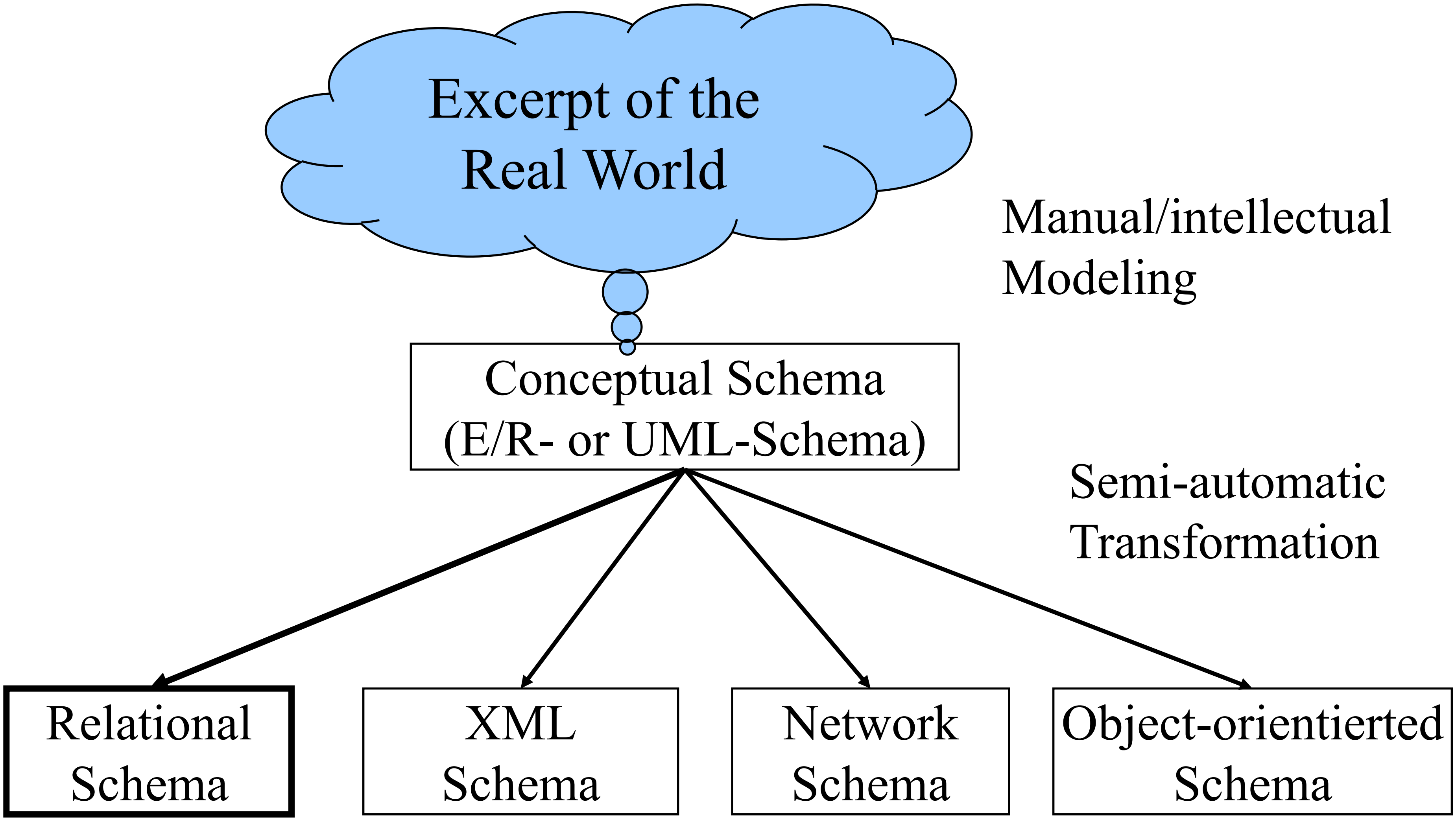
DBS can take care automatically of many things – but the user has to specify

- Requirements of the application
- Characteristics of the data

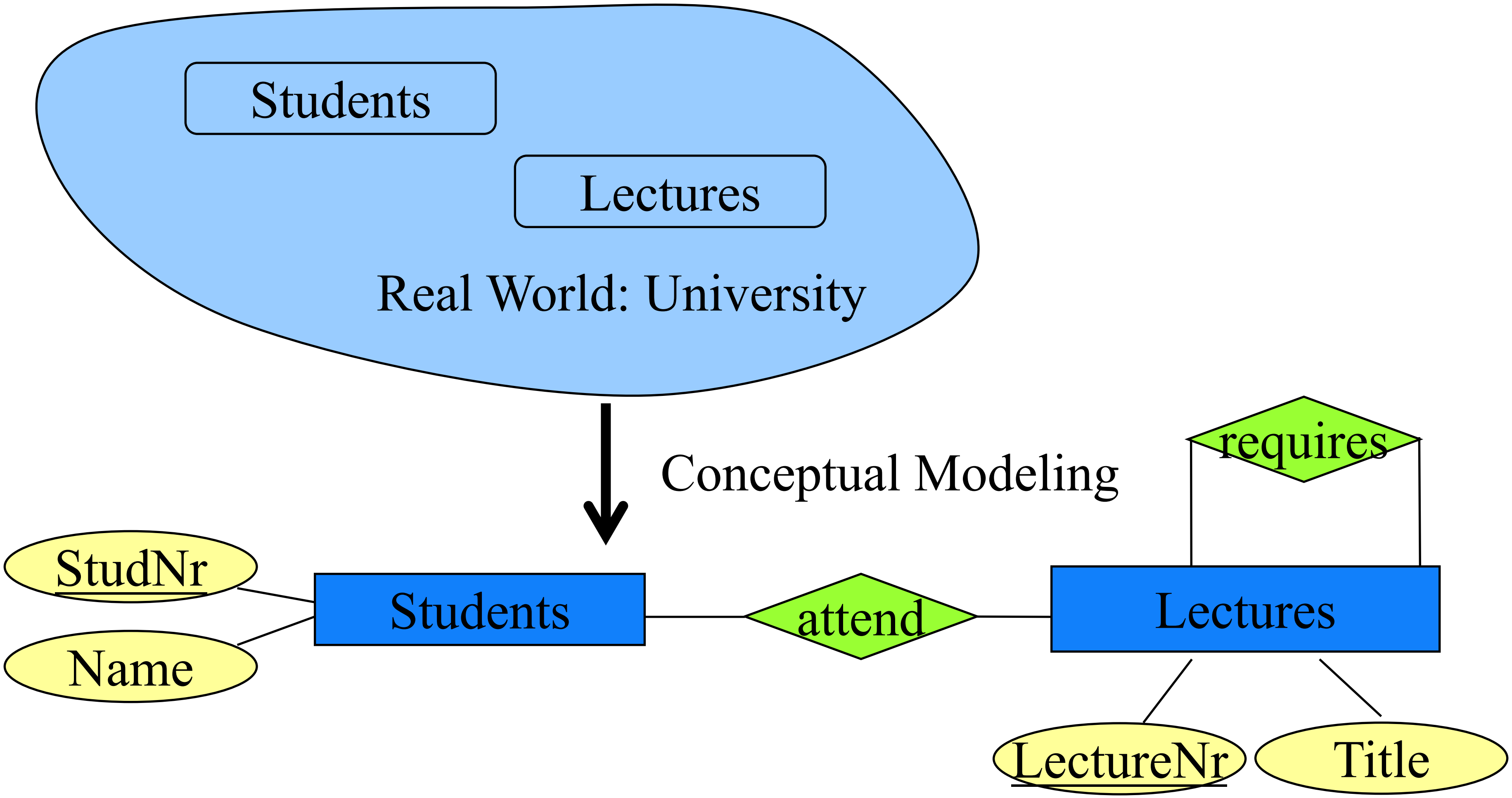
Two important concepts during DBS design:

- Data Model: How to describe the data?
- Data Schema: Concrete description of the data (using the chosen data model)

Data modeling



Modeling a small example application: E/R



Logical Data Models

- Network Model
- Hierarchical Model
- **Relational Data Model**
- XML Model
- Object-orientierted Data Model
    - Object-relational Schema
- Deductive Data Model

* [Michael Stonebraker: What Goes Around Comes Around]

Relational Data Model

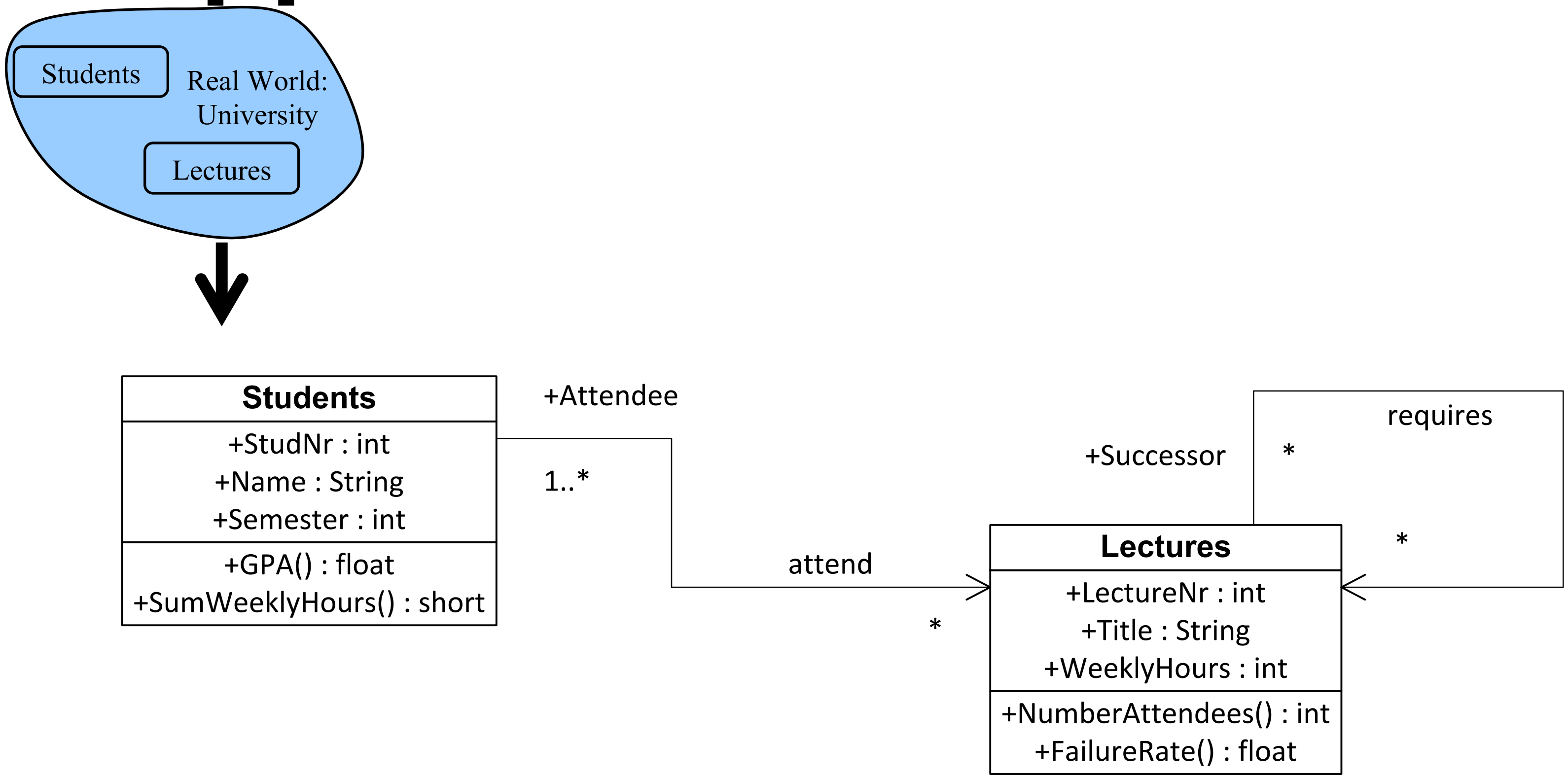
Students	
StudNr	**Name**
26120	Fichte
25403	Jonas
...	...

attend	
StudNr	**Lecture Nr**
25403	5022
26120	5001
...	...

Lectures	
Lecture Nr	**Title**
5001	Grundzüge
5022	Glaube und Wissen
...	...

Select Name
From Students, attend, Lectures
Where Students.StudNr = attend.StudNr **and**
attend.LectureNr = Lectures.LectureNr **and**
Lectures.Title = ‘Grundzüge‘;

Modeling a small example application: UML



Database Design

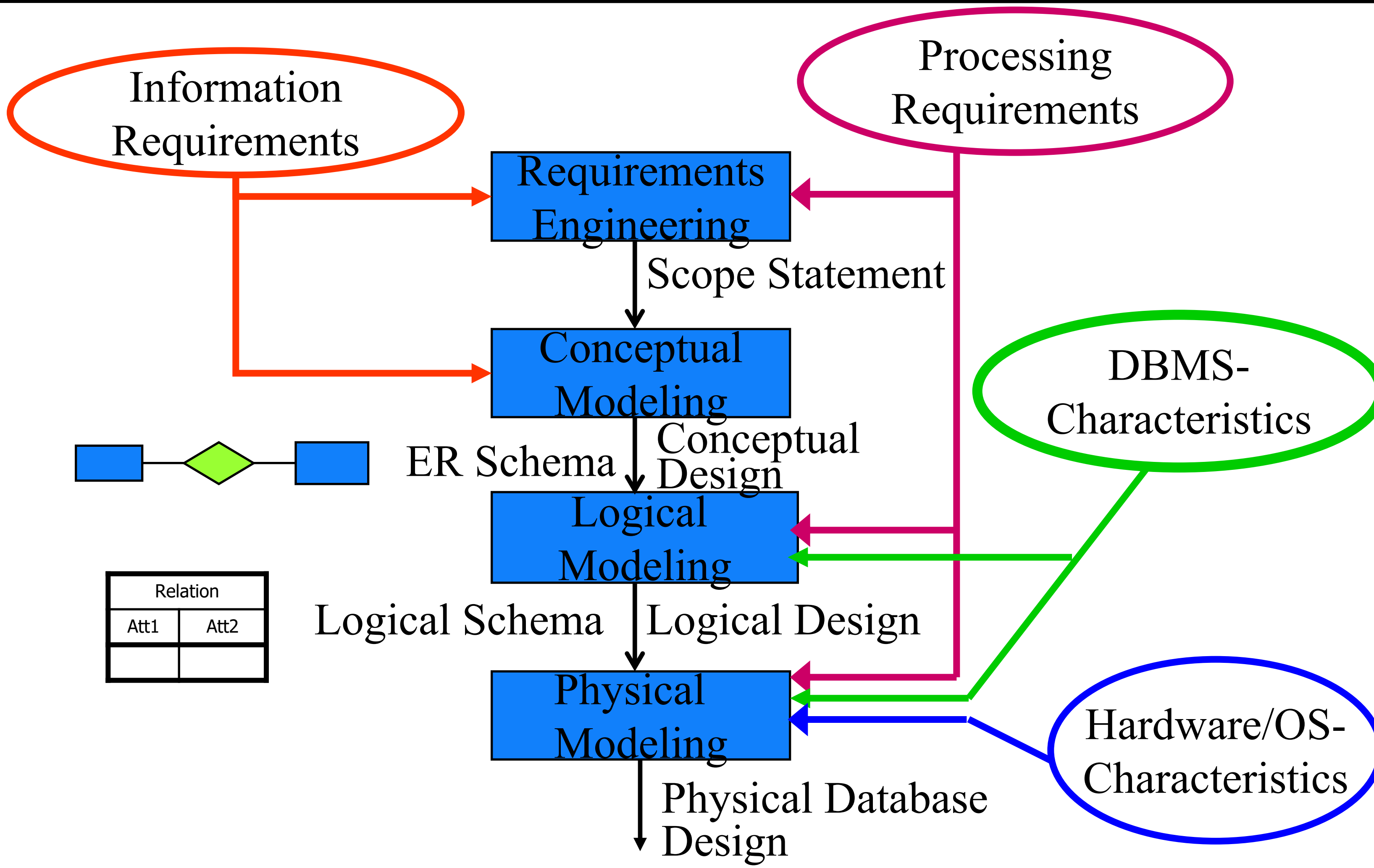
Database Abstraction Layers

1. Conceptual Design

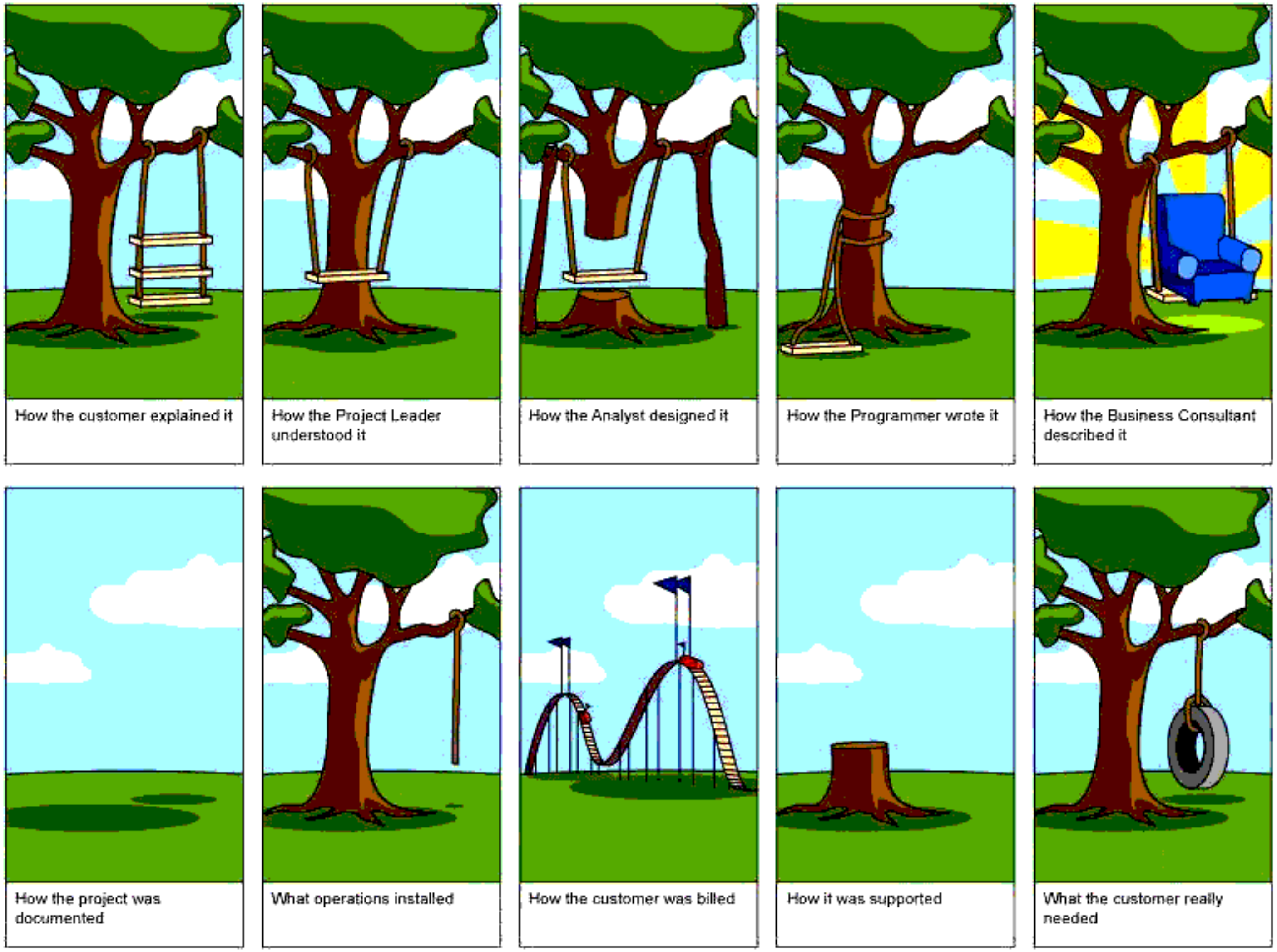
2. Logical Design

3. Physical Database Design

Phases of Database Design



Software Development and Ability to Communicate



Schema Design

Approach in principle:

Information Requirements	Semantical Data Modeling	Logical Data Modeling	Database Installation / Tuning
Semantical Analysis	Coarse Grain Data Modeling Fine Grain Data Modeling		Time →
- Interview - Brainstorming - Document‘s Analy. - …	- ERM - UML - …	- Hierarchical - network - relational - object-oriented - ….	- IMS - UDS - DB2 - Ozone - …
Conceptual Schema Design		Logical Schema Design	Physical Schema Design

Requirements Engineering

Entity description

Relation description

Process description

…

Entity Description

University Employees

- -Quantity: 1000
- -Attributes
  - ❖ EmpNumber
    - •Type: Integer
    - •Domain: 0...999.999.99
    - •Defined: 100%
    - •Identifying: yes
    - •Example: 007
  - ❖ Salary
    - •Type: decimal
    - •Length: (7,2)
    - •Unit: Euro per month
    - •Defined: 10%
    - •Identifying: no
  - ❖ Level
    - •Type: String
    - •Length: 2
    - •Defined: 100%
    - •Identifying: no
    - •Example: W2

Relation Description: *exam*

Involved Objects:

- Professor as Tester
- Student as Testee
- Lecture as Test Subject

Attributes of the Relation:

- Date
- Time
- Grade

Quantity: 100 000 per year

Process Description : *Issue a Certificate*

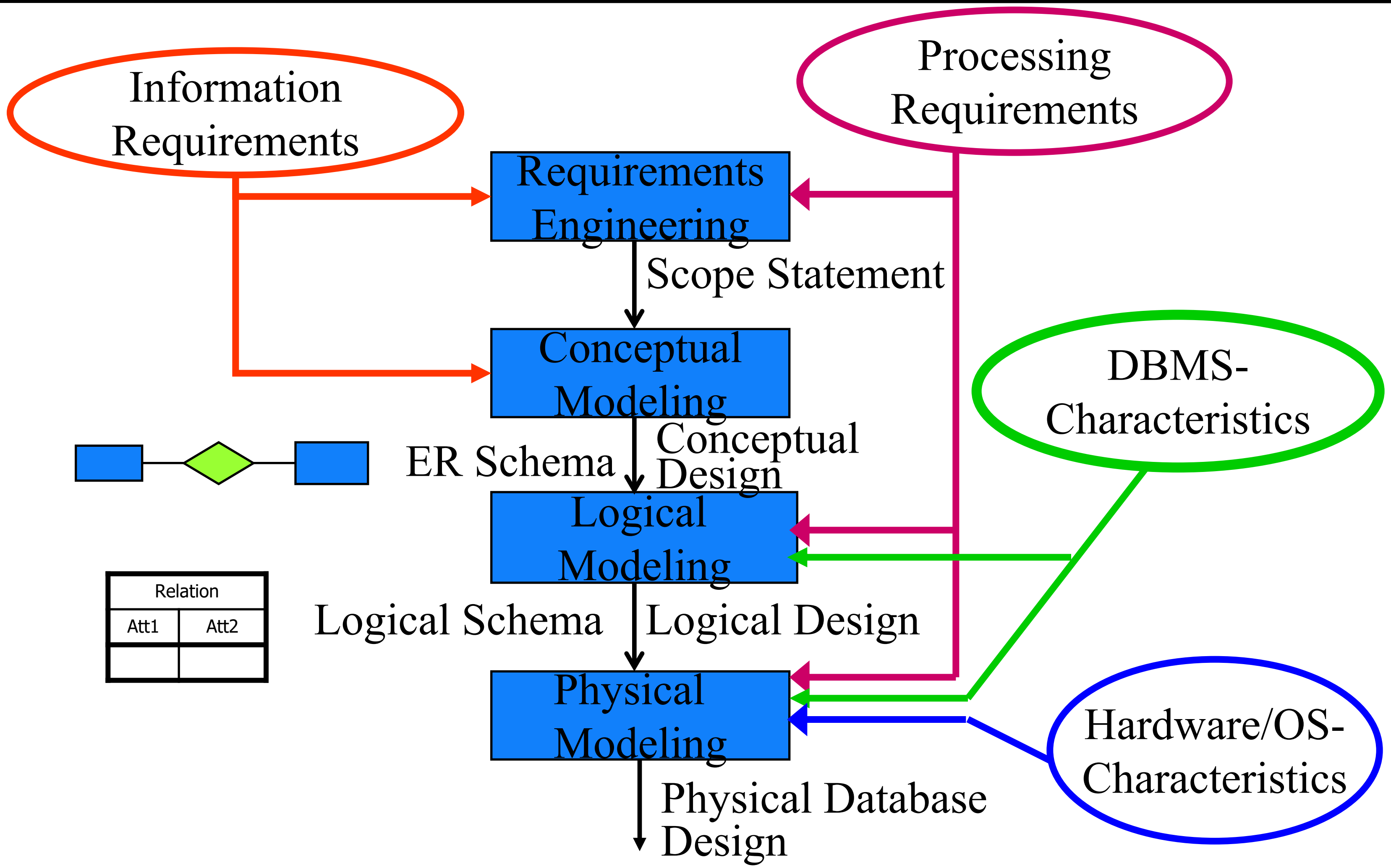
- Frequency: semiannually
- Required Data
  - Tests
  - Examination Rules
  - Student‘s Records
  - ...
- Priority: high
- Data Volume to be processed
  - 500 Students
  - 3000 Tests
  - 10 Versions of Examination Rules

Creating a Specification

The actual analysis is an iterative process:

- Customer tells developer his/her needs
- Developer notes everything down (s/he understood) in his/her „language" . . .
- . . . and translates it into the " language" of the customer
- This is shown to the customer who does not agree with everything
- Change requests are agreed on
- Back to step 2

Phases of Database Design



Conceptual Design

The ideal design (the ideal specification) is

- unique
- complete
- comprehensible (for all participants)
- nonredundant
- . . . and not reachable in reality

Entity/Relationship-Modeling

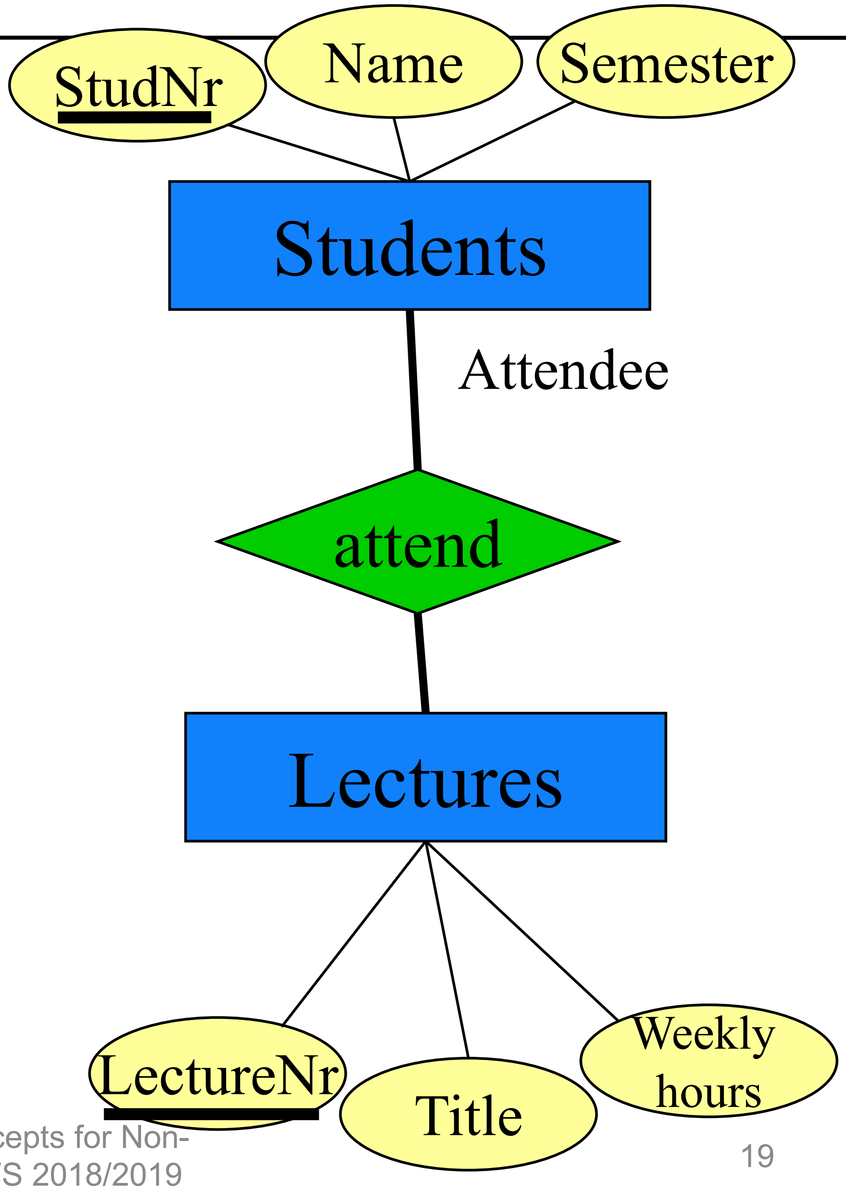
Entity

Relationship

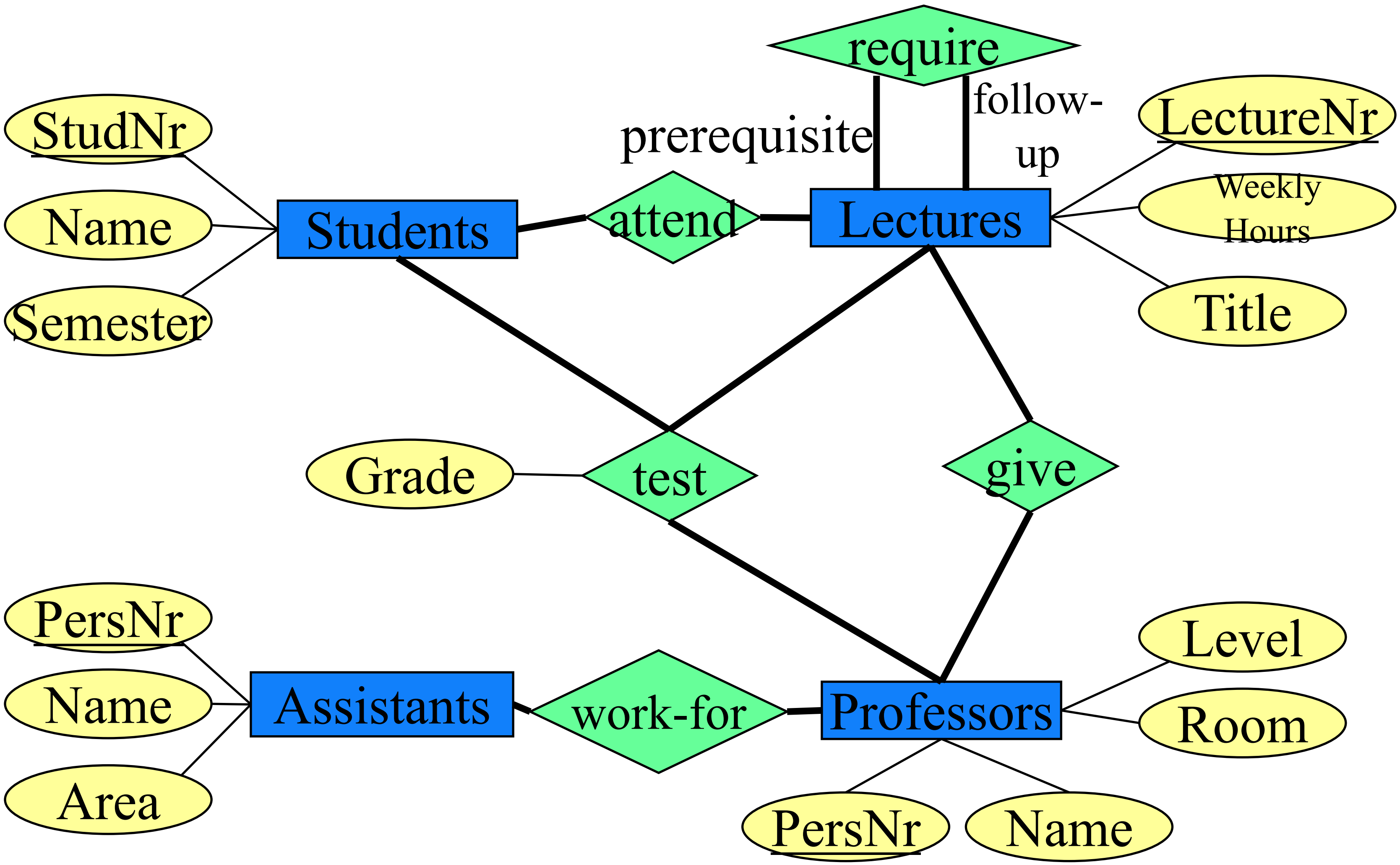
Attribute (property)

Key (identification)

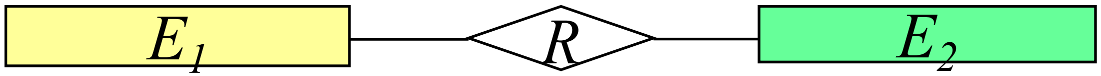
Role



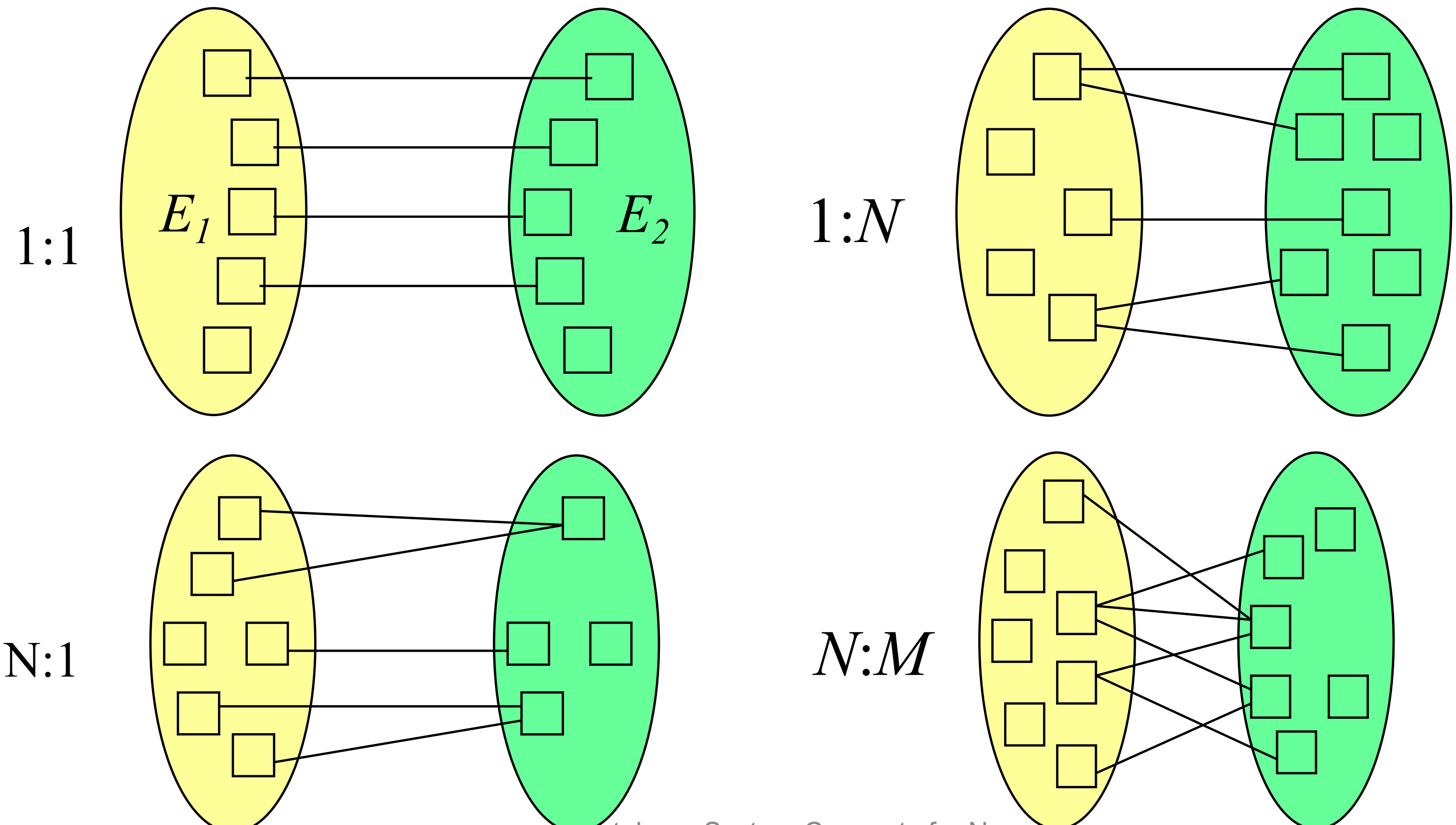
University Schema



Functionalities

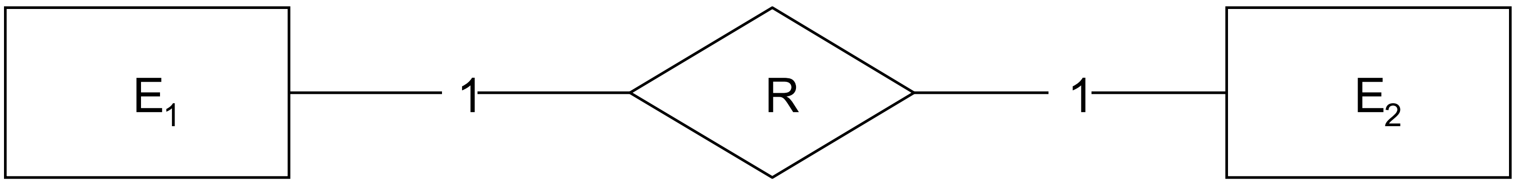


$R \subseteq E_1 \times E_2$



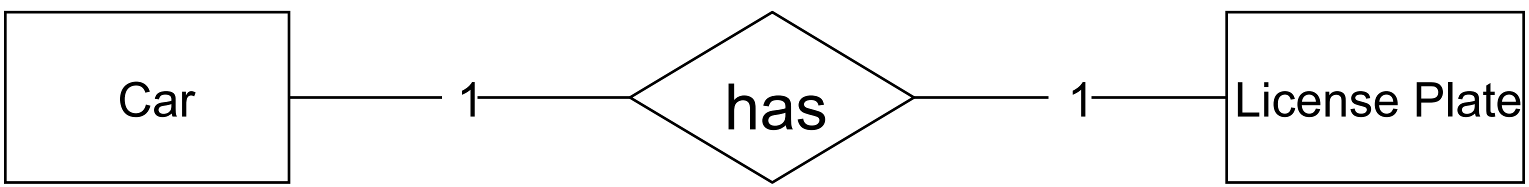
Relationship 1:1

Relationship 1:1



e_1 out of E_1 takes part in 1 relation of type R
e_2 out of E_2 takes part in 1 relation of type R

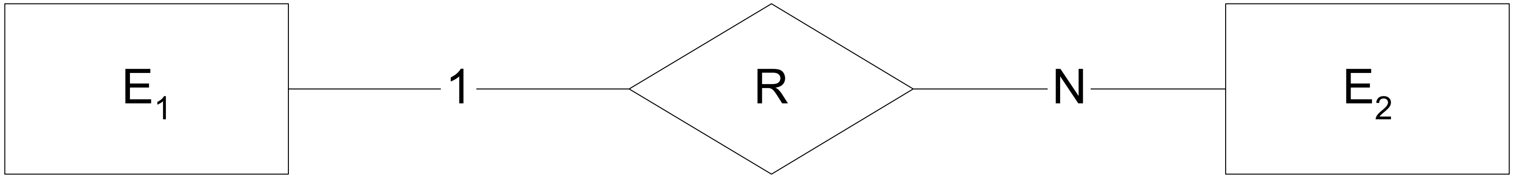
Example:



one car has one license plate
one license plate belongs to one car

Relationship 1:N

Relationship 1:N



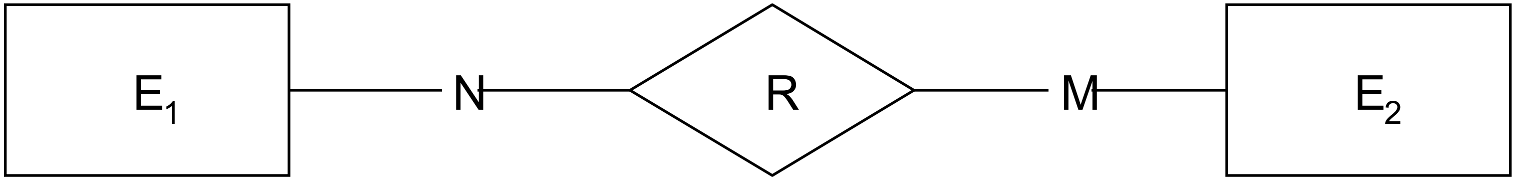
e_1 out of E_1 takes part in N relations of type R
e_2 out of E_2 takes part in 1 relation of type R

Example:

one mentor advises several students
one student is advised by one mentor

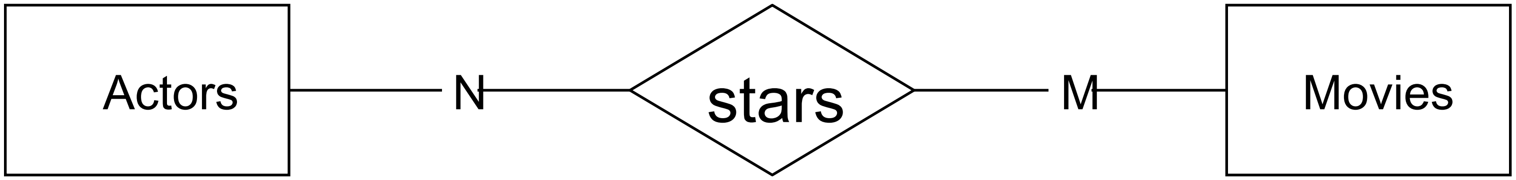
Relationship N:M

Relationship N:M



e_1 out of E_1 takes part in M relations of type R
e_2 out of E_2 takes part in N relation of type R

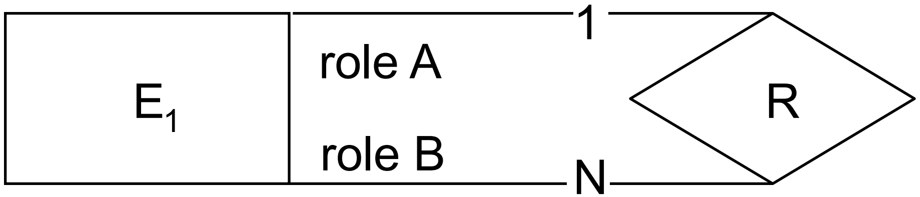
Example:



one actor stars in several movies
one movie has several actors

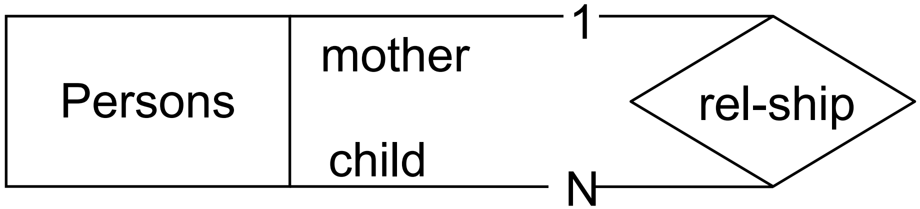
Recursive Relationship 1:N

Relationship 1:N



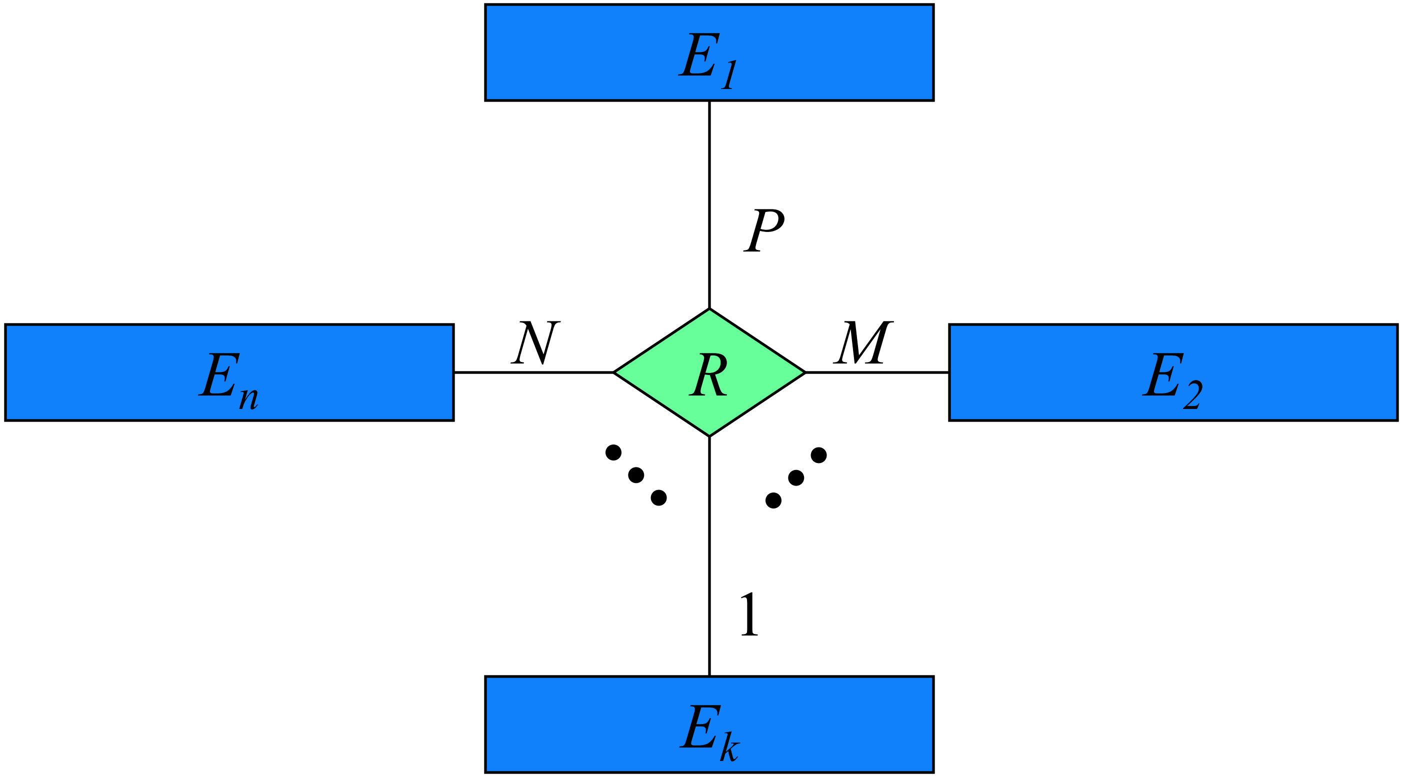
e_1 out of E_1 takes part in role A in N relations of type R
e_1 out of E_1 takes part in role B in 1 relation of type R

Example:



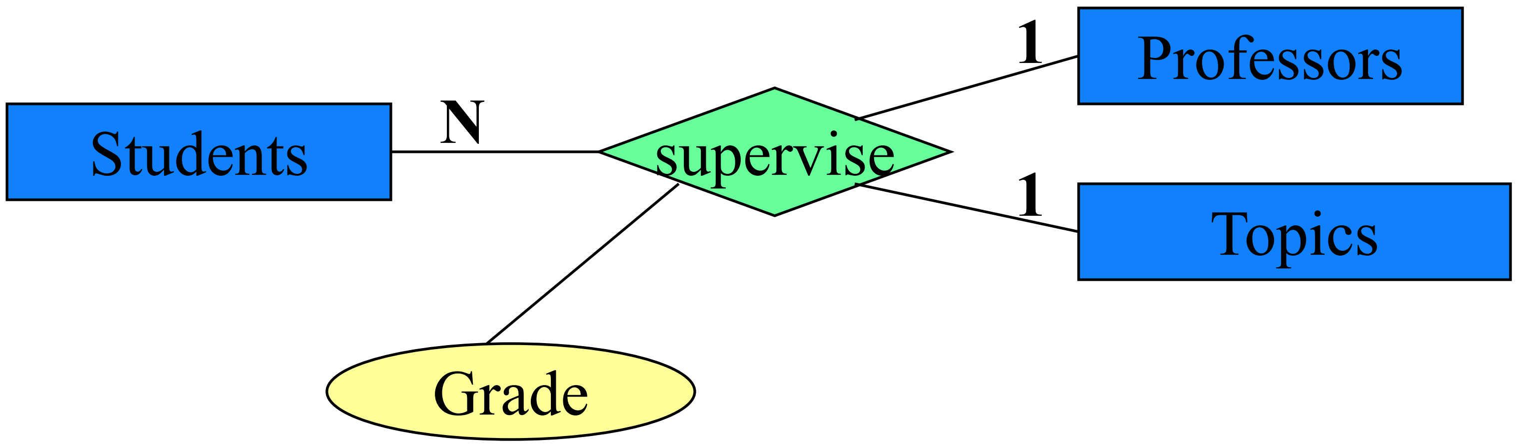
one person is mother of several persons (children)
one person is child of one person (mother)

Functionalities in *n*-ary Relationships



$$R : E_1 \times \ldots \times E_{k-1} \times E_{k+1} \times \ldots \times E_n \rightarrow E_k$$

Example Seminar



supervise : Professors x Students → Topics

supervise : Topics x Students → Professors

Thereby induced Consistency Constraints

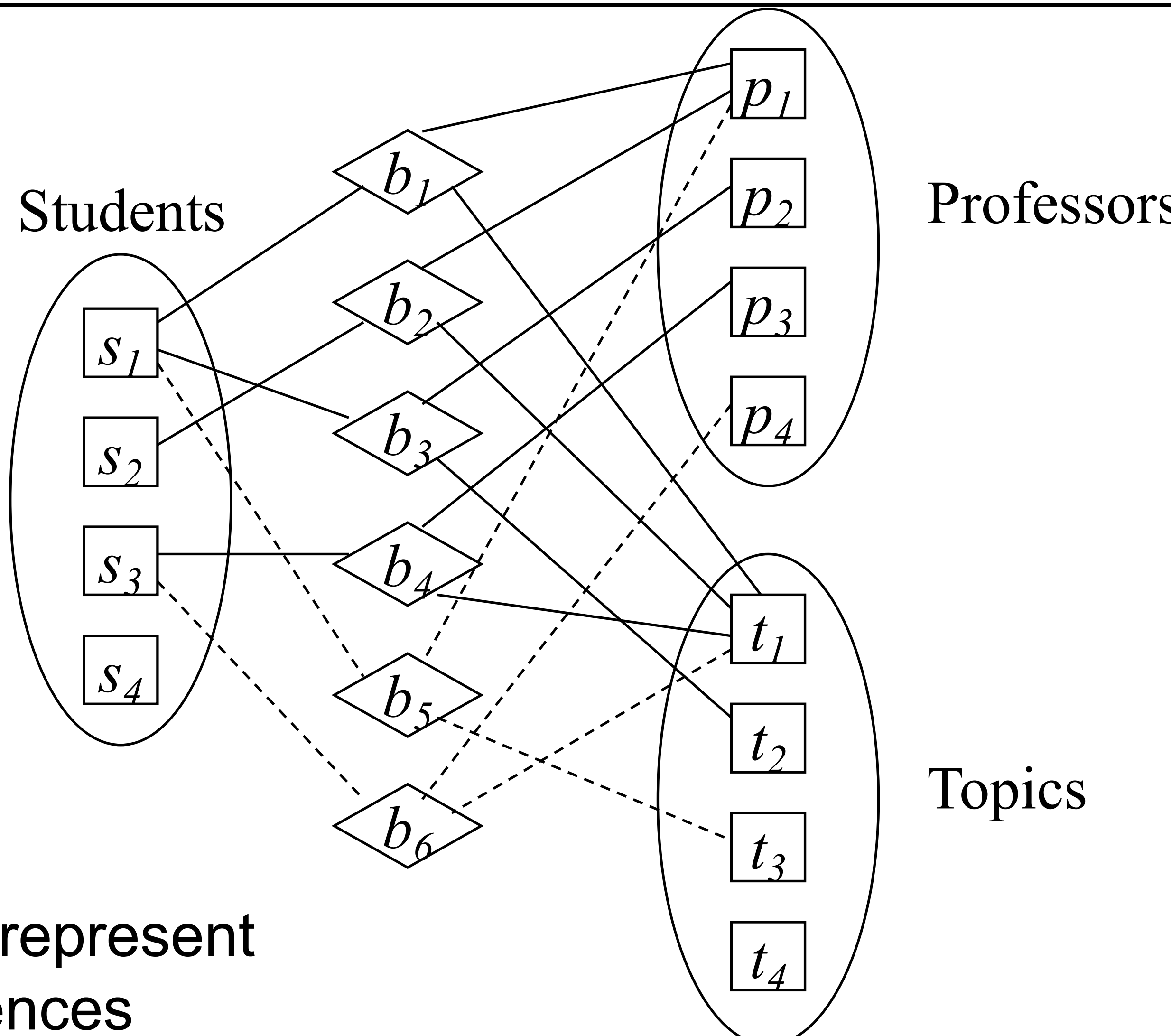
1. Students may work on only one topic with the same professor (to cover a broad spectrum)

2. Students may work on the same topic only once – thus they may not work on the same topic again with another professor

3. Professors can reuse the same topic – i.e. give the same topic to different students

4. One topic can be given by different professors – but to different students

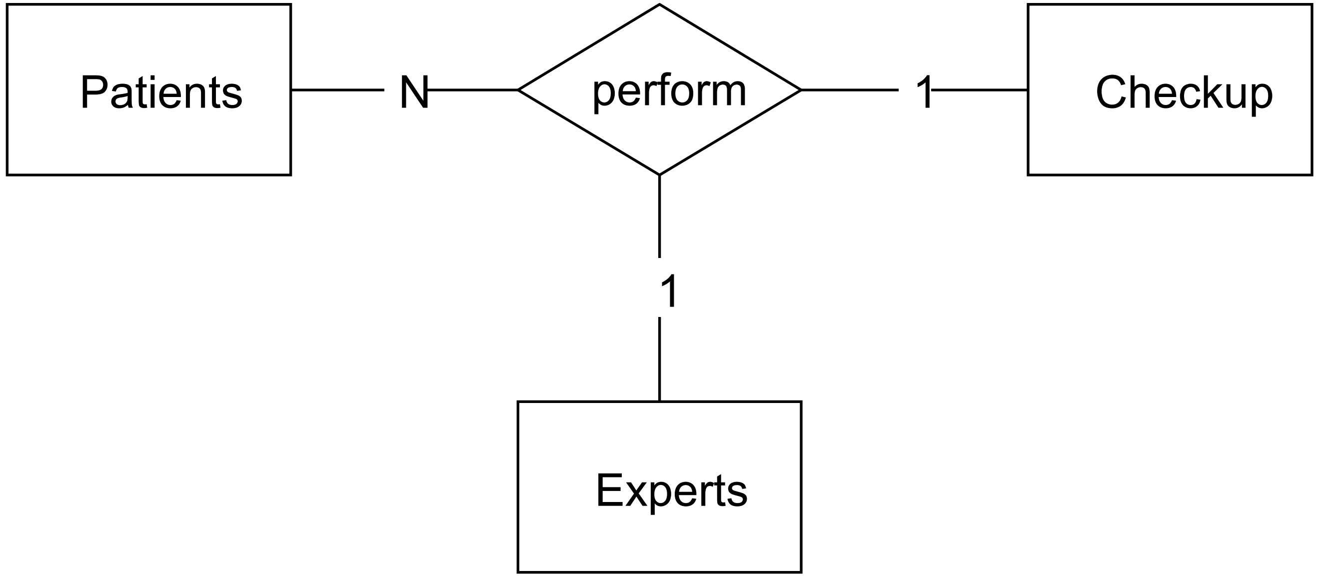
Occurrence of the Relationship *supervise*



Dashed lines represent illegal occurrences

One more Example

3-ary relationship:

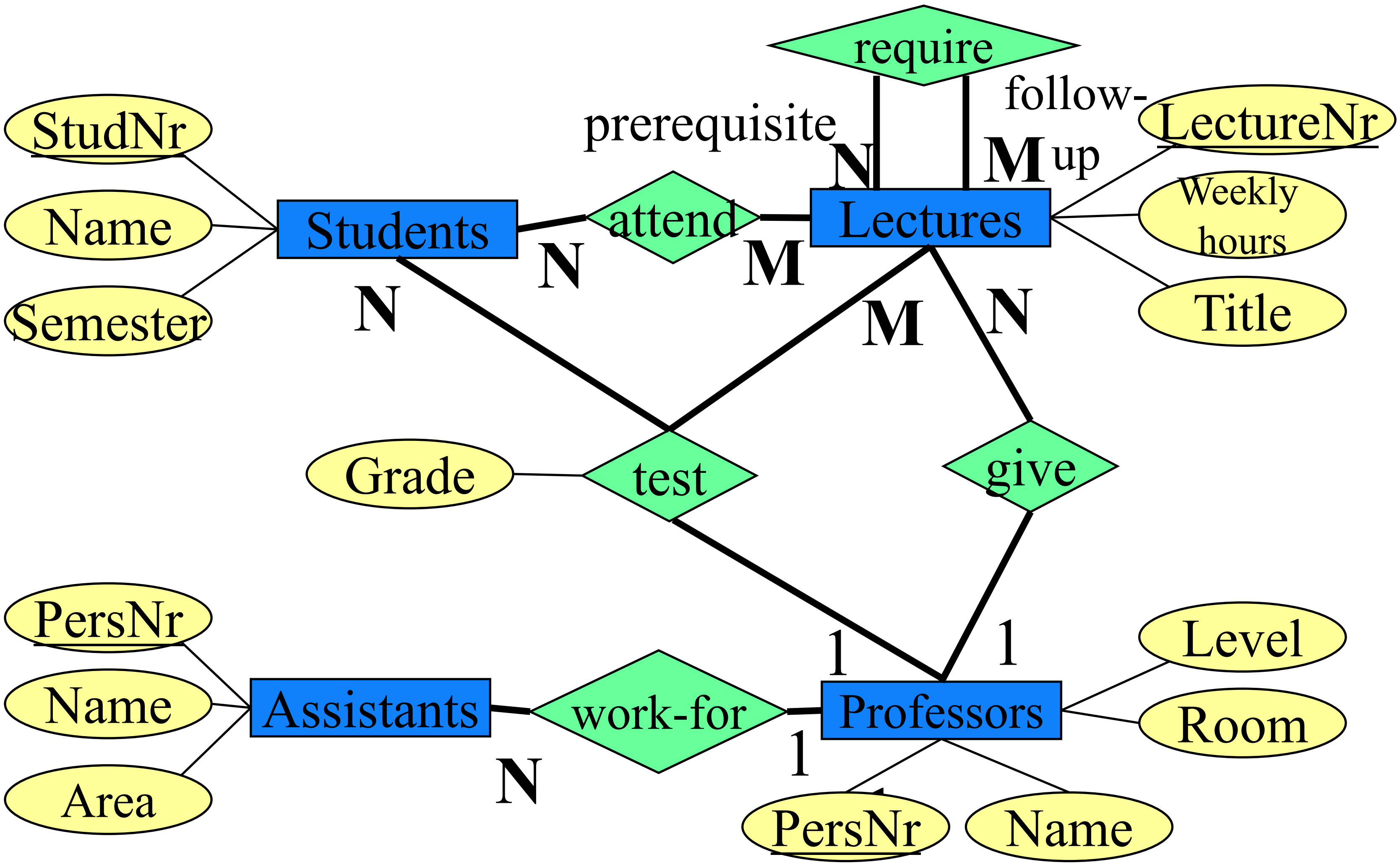


One checkup is performed by one expert with several patients

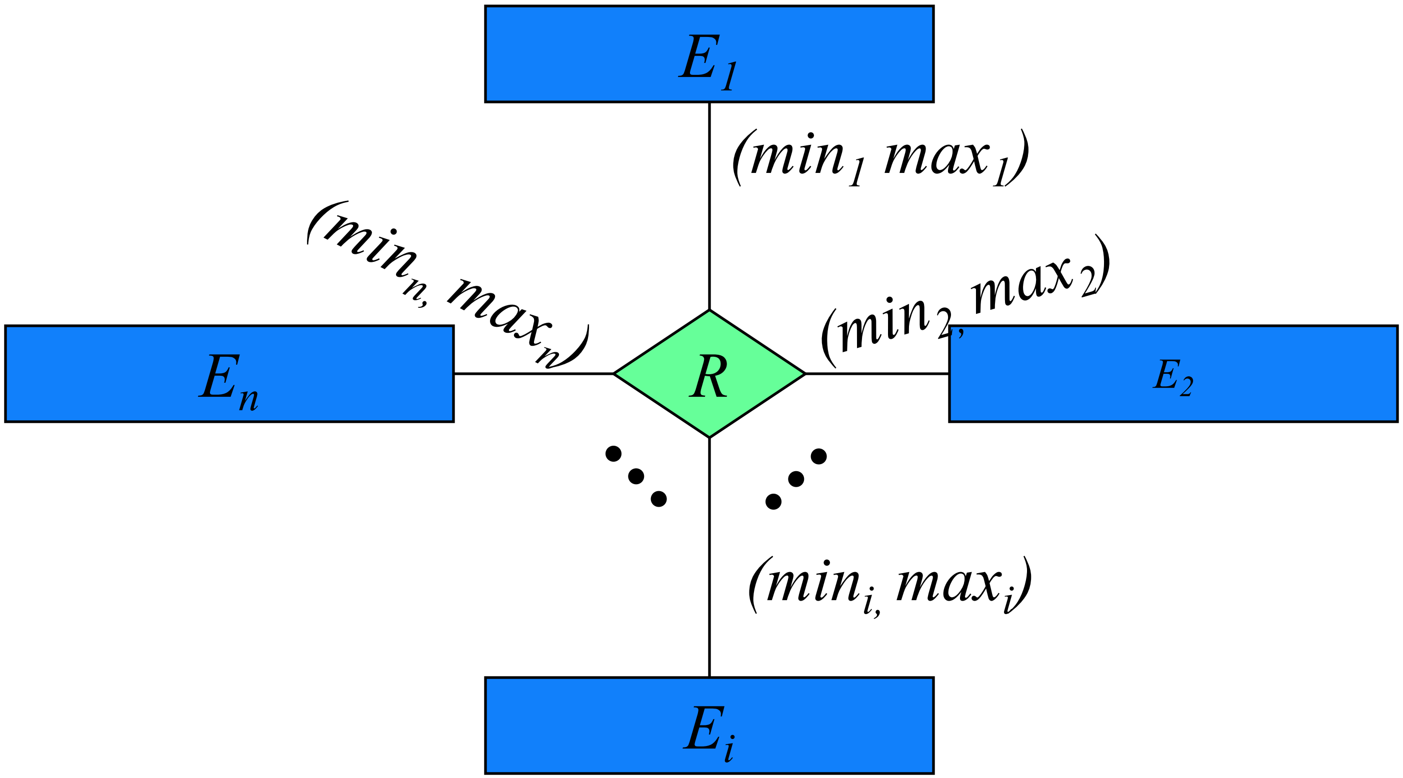
One Patient gets only one checkup from one expert

One checkup is performed at one patient only by one expert

University Schema



(min, max)-Notation

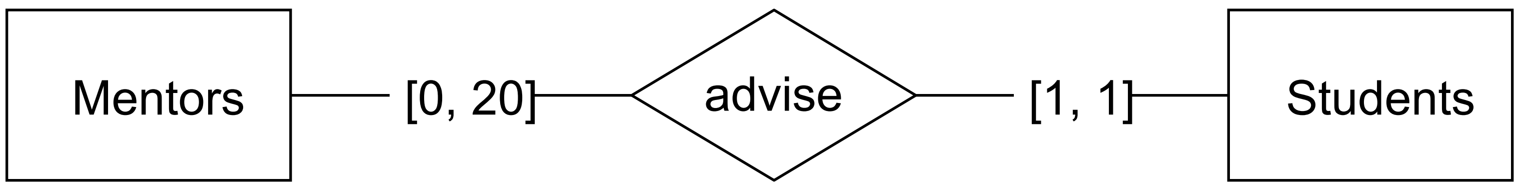


$R \subseteq E_1 \times \ldots \times E_i \times \ldots \times E_n$

For every $e_i \in E_i$ *there are*

- *at least* min_i tuples $(\ldots, e_i, \ldots) \in R$ and
- at most max_i tuples $(\ldots, e_i, \ldots) \in R$

Example (min, max)



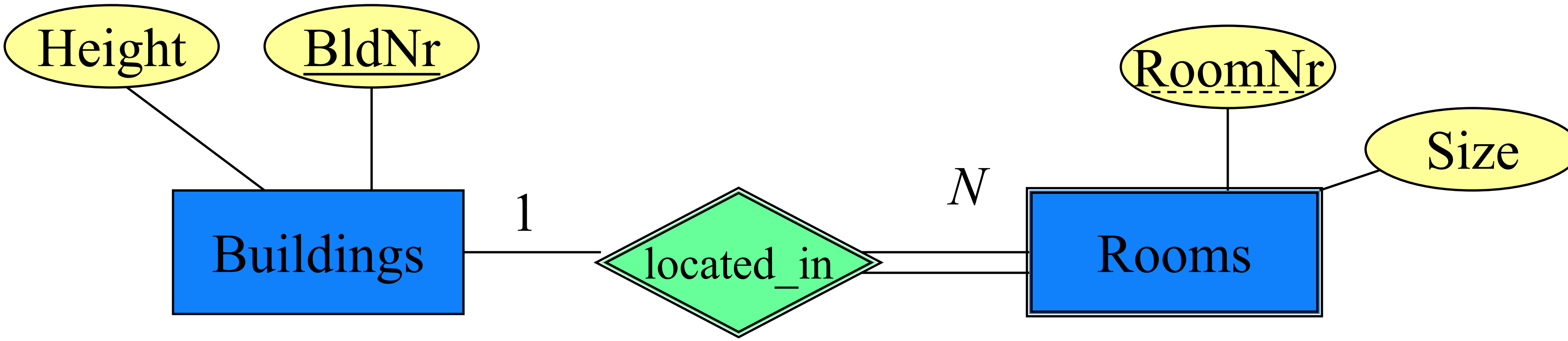
one mentor advises up to 20 students
one student is advised by exactly one mentor

Excercise for next class

Inform yourself about unary – binary – ternary relationships

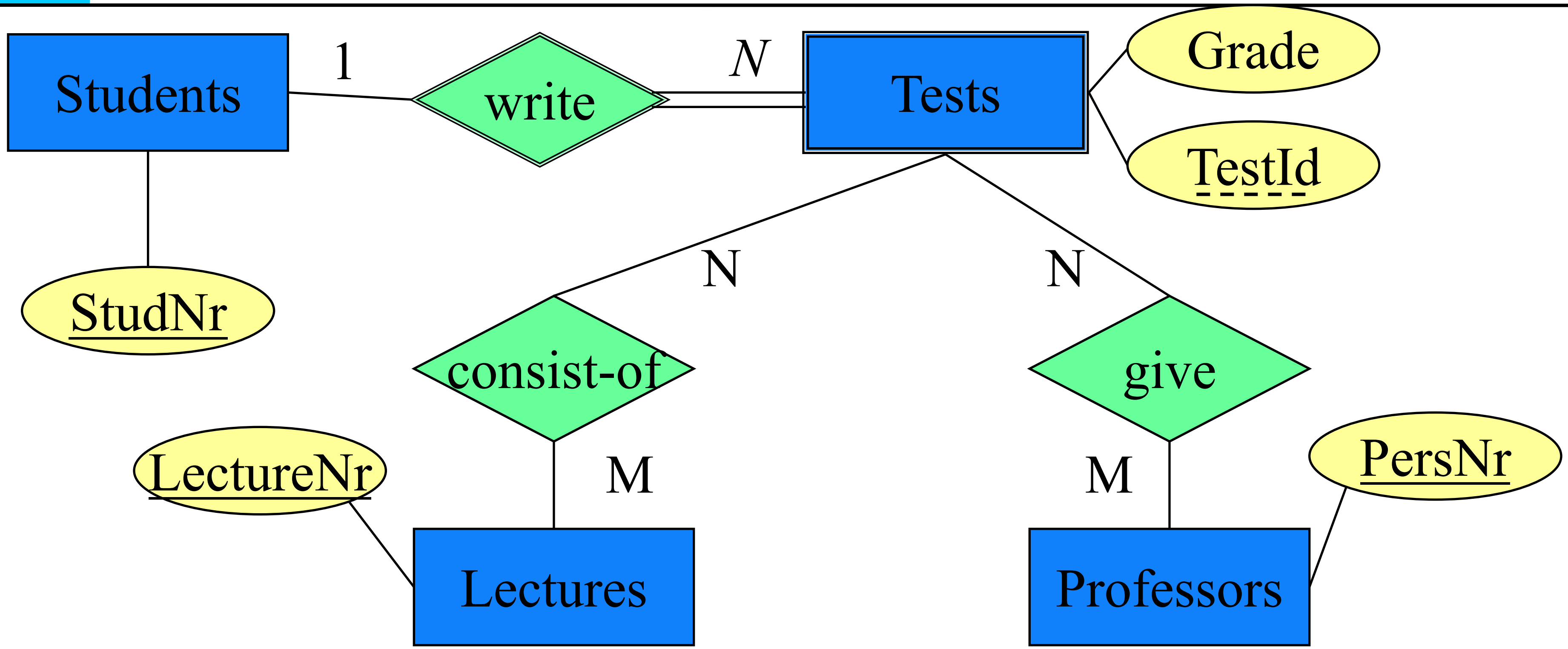
Discussion / new examples next class!

Weak Entities



- Relationship between "strong" and "weak " type is 1:*N* (or 1:1 in rare cases) - why not *N:M*?
- The existence of a room depends on the existence of the associated building
- RoomNr is unique only within the building
- Key of Rooms is: RoomNr **and** BldNr

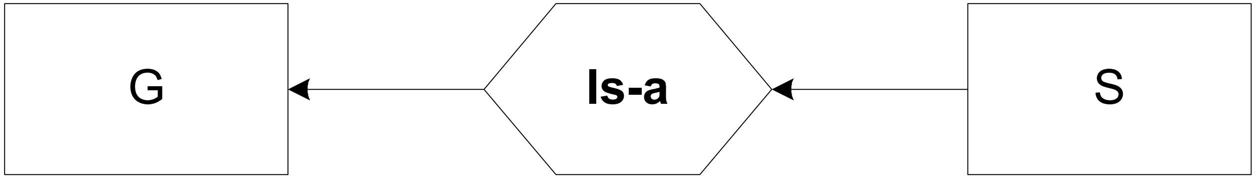
Tests as weak entity type



- Several professors design one test
- Several lectures are inquired in one test

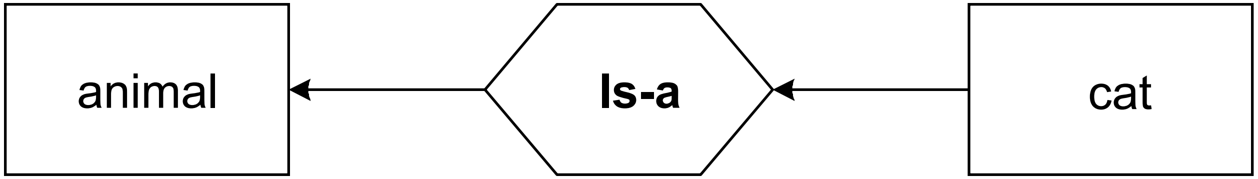
Generalization

Generalization / Specialization:

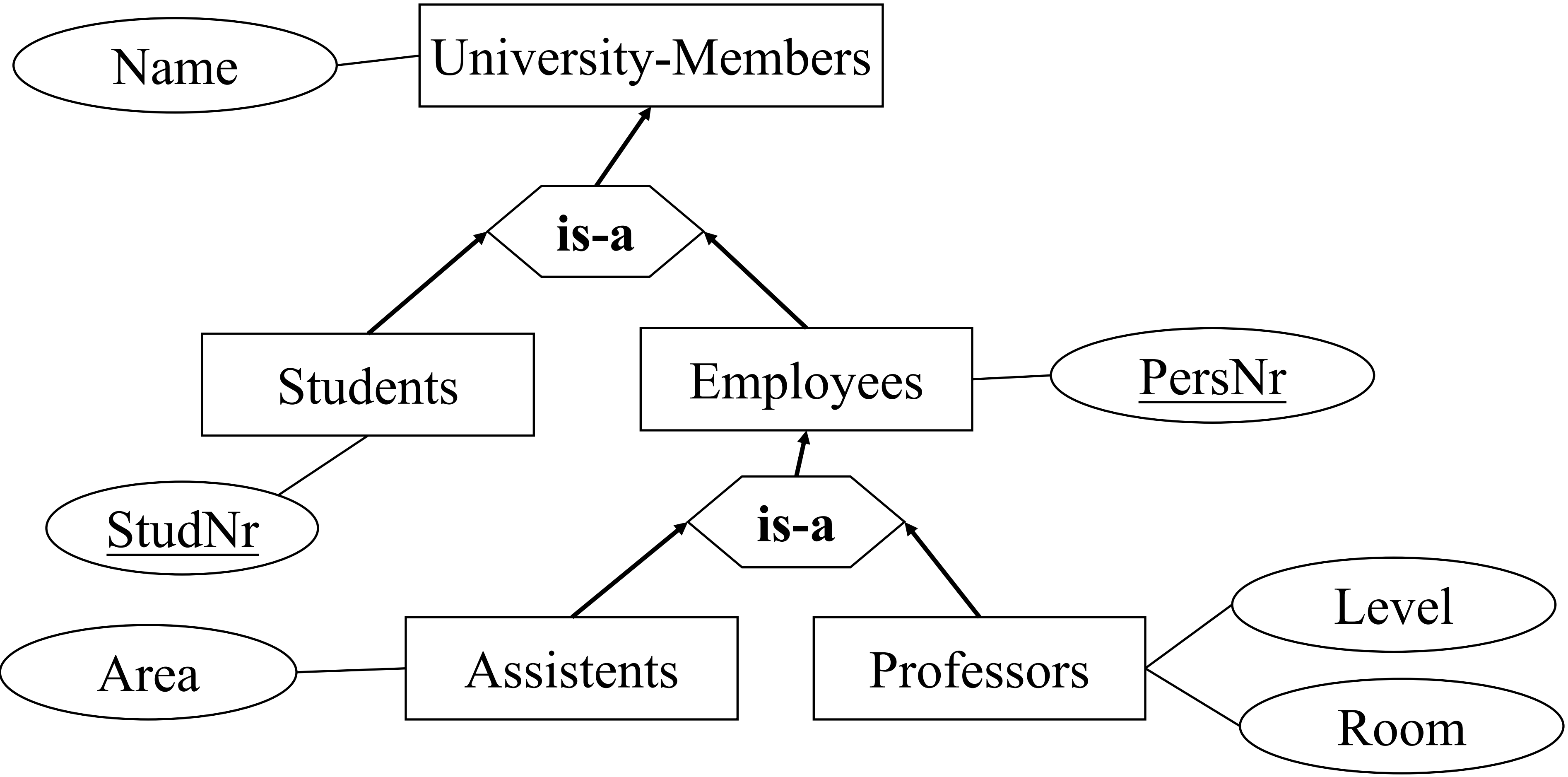


S is a specialization of G

Example:



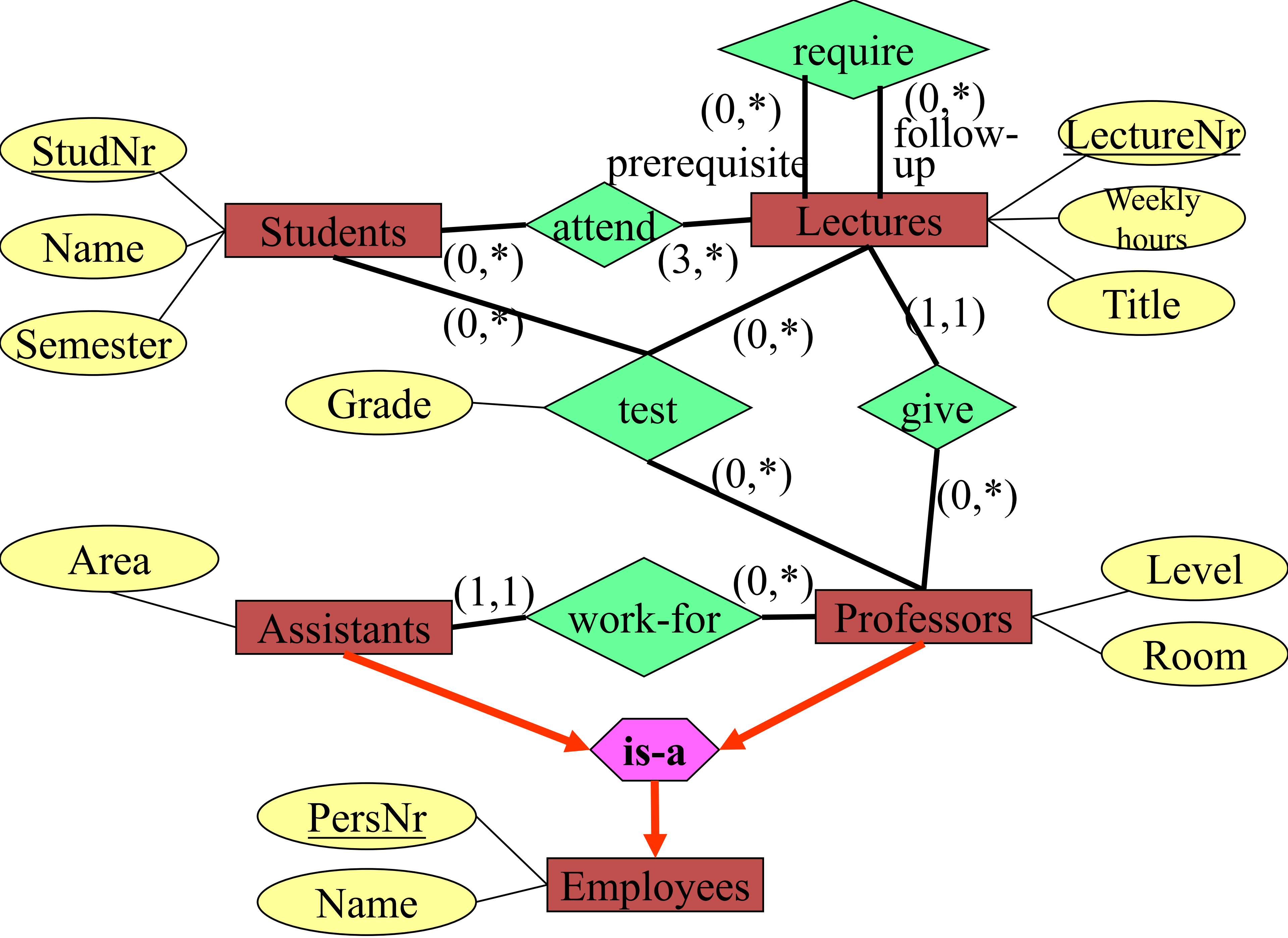
Generalization University



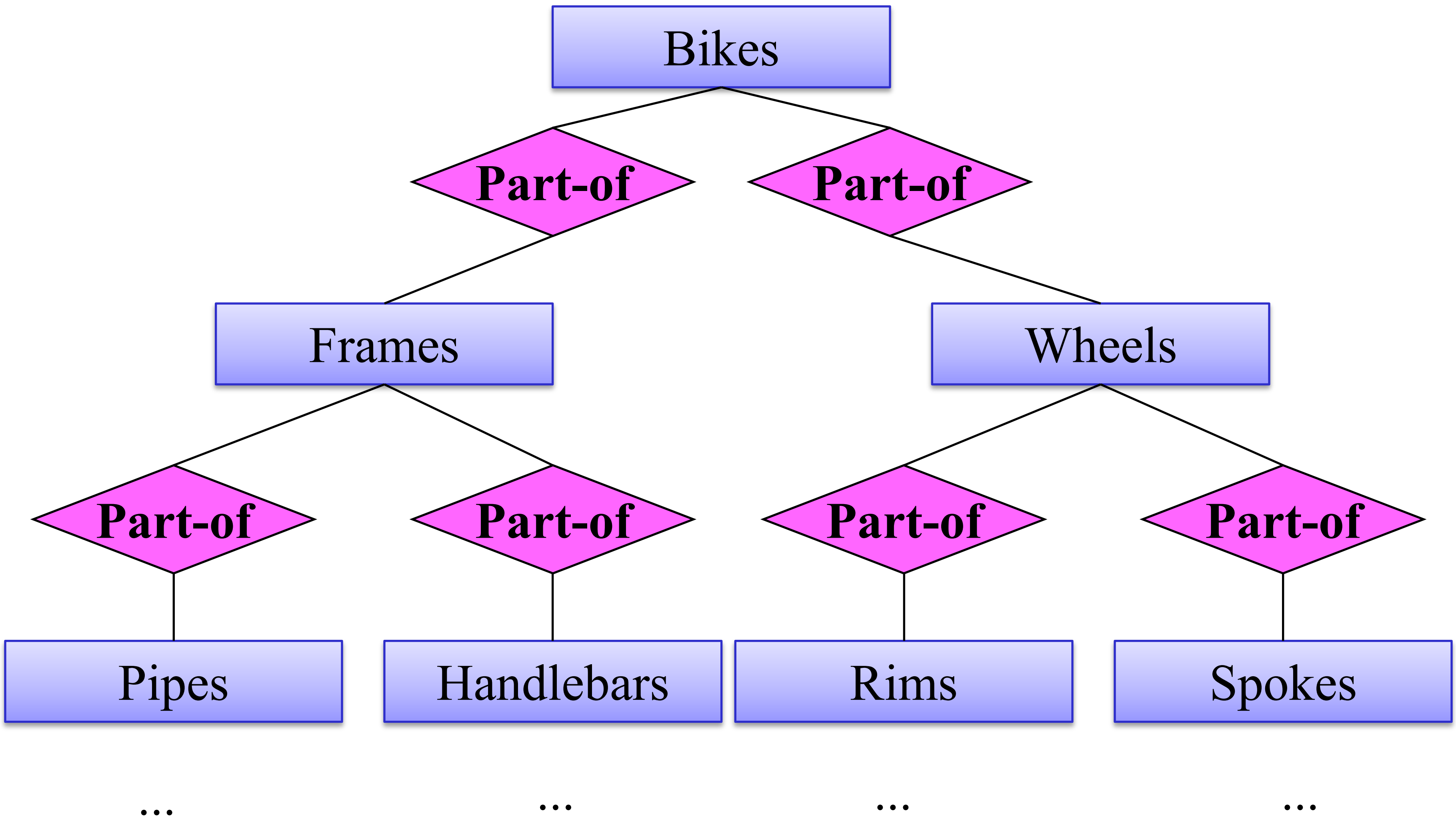
Conclusion

University schema with generalization and (min, max)-notation

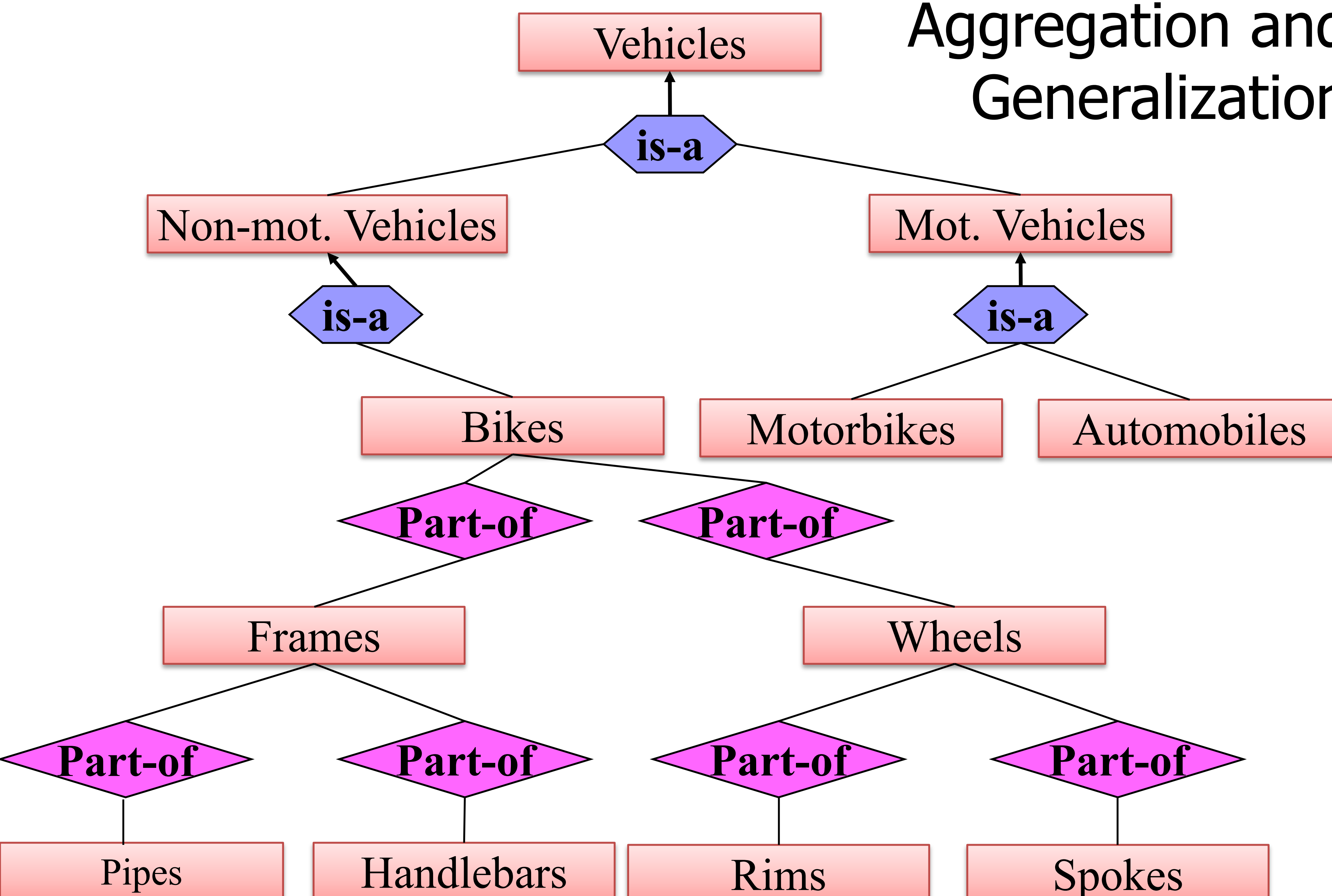
→ Nextpage


require
(0,*)
prerequisite
(0,*)
follow-up
StudNr
Name
Semester
Students
attend
(0,*)
(3,*)
Lectures
LectureNr
Weekly hours
Title
(0,*)
(0,*)
(1,1)
Grade
test
give
(0,*)
(0,*)
Area
Assistants
(1,1)
work-for
(0,*)
Professors
Level
Room
is-a
PersNr
Name
Employees

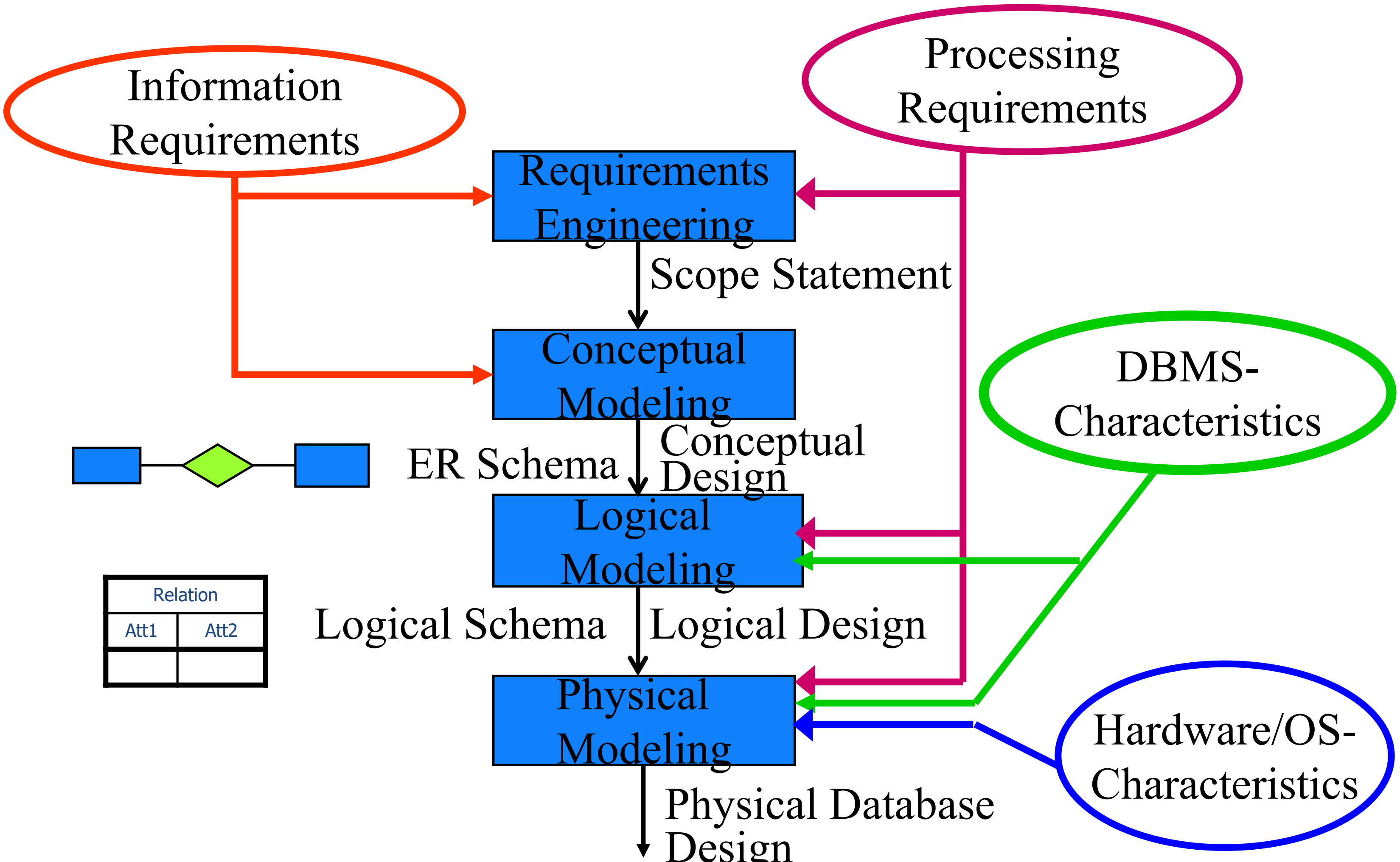
Aggregation



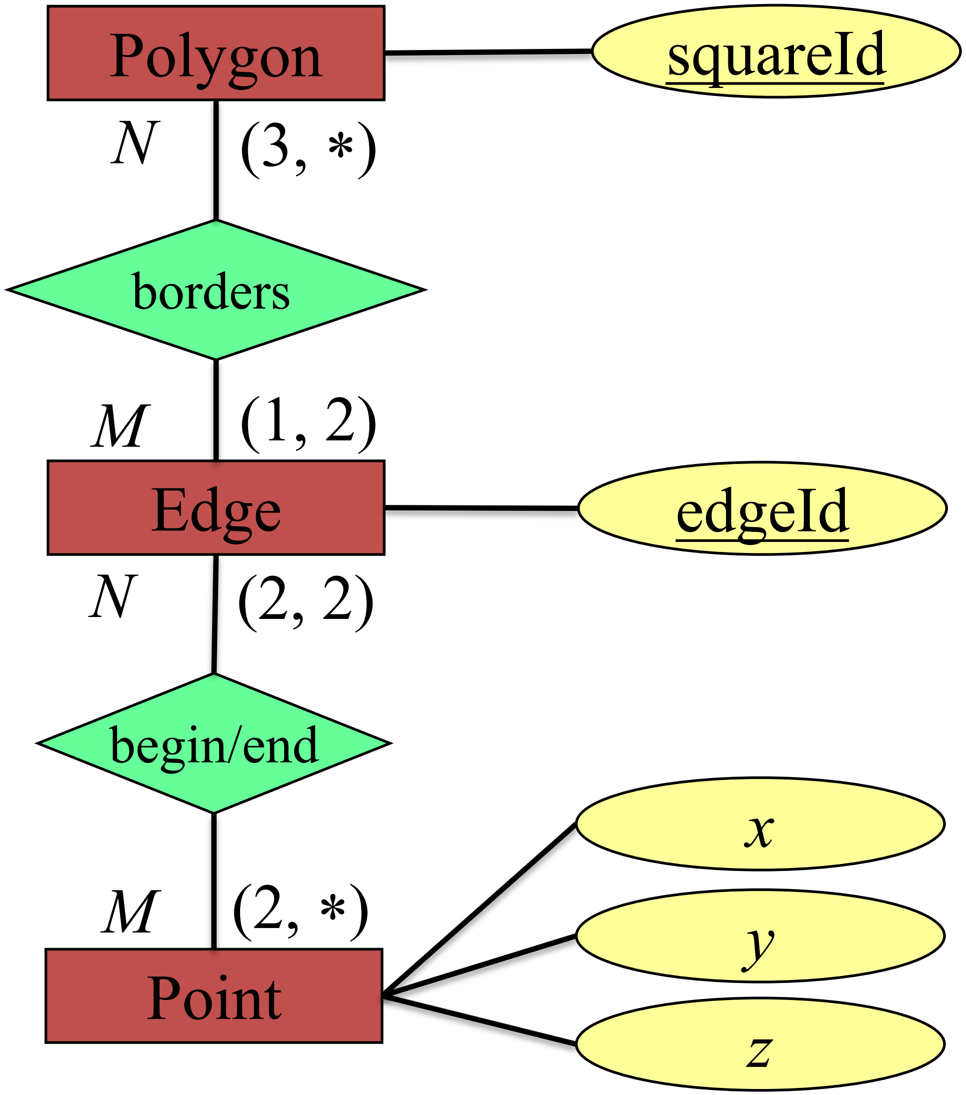
Aggregation and Generalization

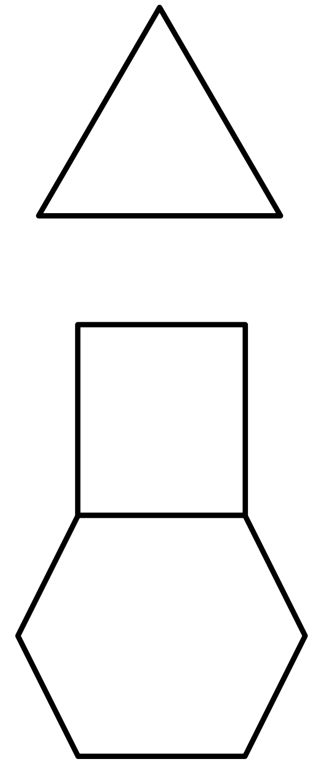


Where are we?


Information Requirements
Processing Requirements
Requirements Engineering
Scope Statement
Conceptual Modeling
DBMS-Characteristics
ER Schema
Conceptual Design
Logical Modeling
Relation
Att1
Att2
Logical Schema
Logical Design
Physical Modeling
Hardware/OS-Characteristics
Physical Database Design

Min,max Notation and Functionalities





Min-max:
A Polygon has at least 3 Edges.
An Edge has 1 or 2 Polygons.

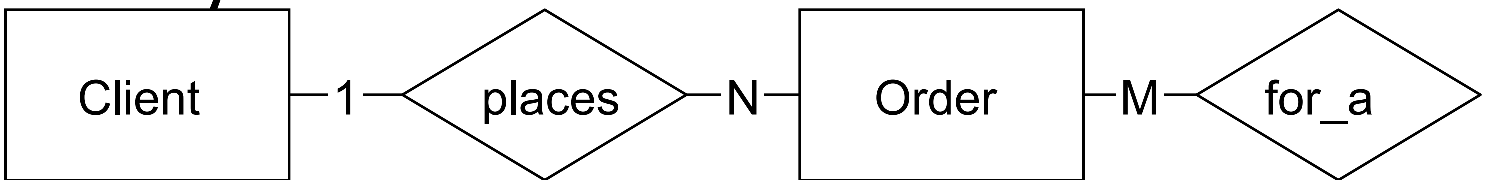
Design criteria

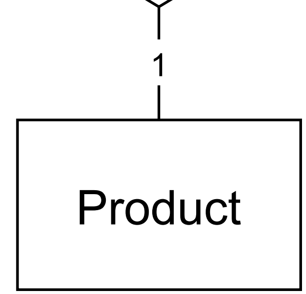
- Rules for classification of entities and attributes:
  - Entities should contain descriptive information
  - Multi valued attributes should be classified as entities
  - Attribute should be assigned to that Entity which describes it most directly
  - Redundant relationships should be avoided

– *However, it always depends on the application*

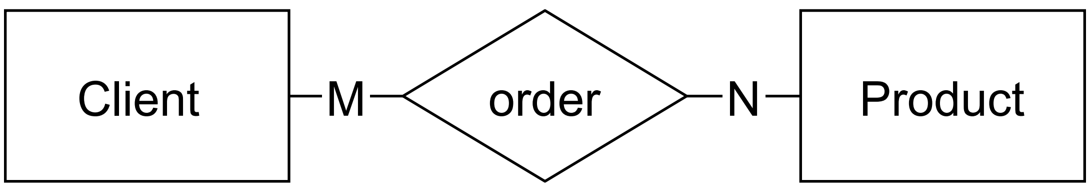
Example: Order

As entity:

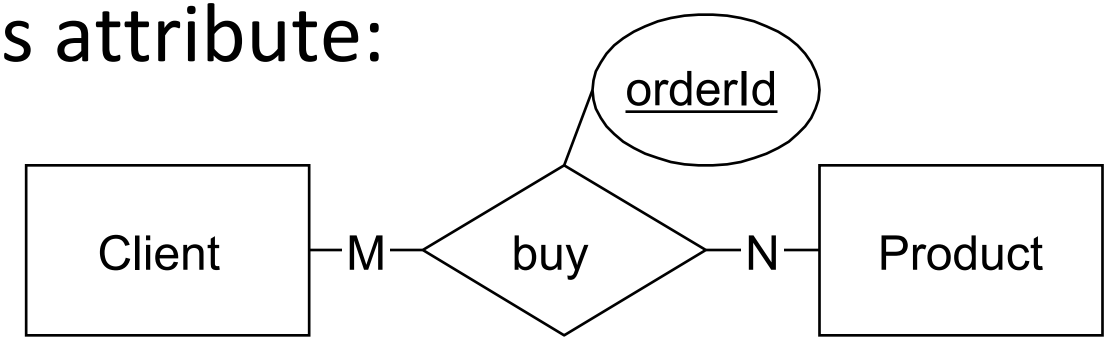




As relationship:

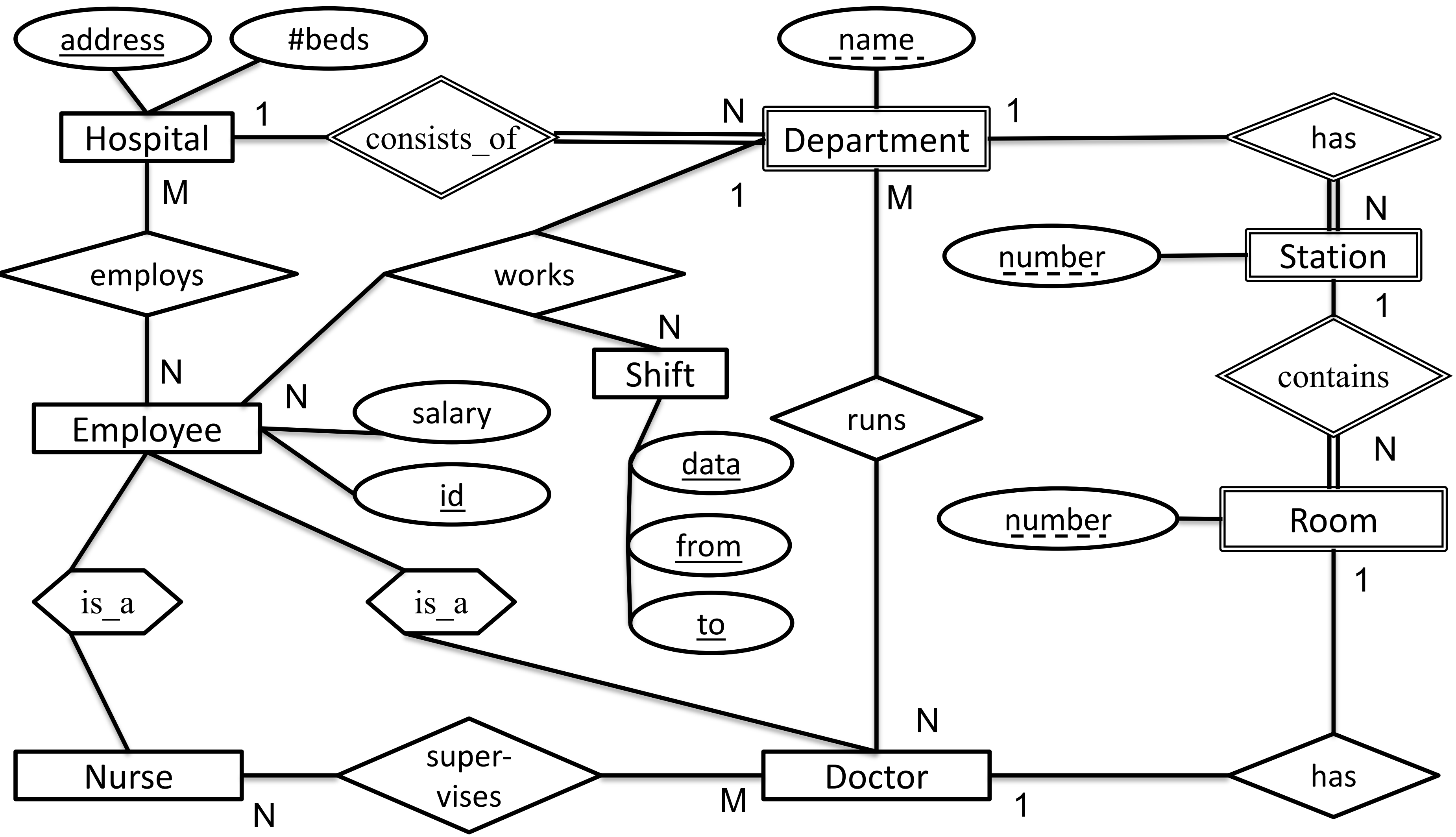


As attribute:



Exercise Sheet 2

Exercise Sheet 2



Exercise Sheet 2