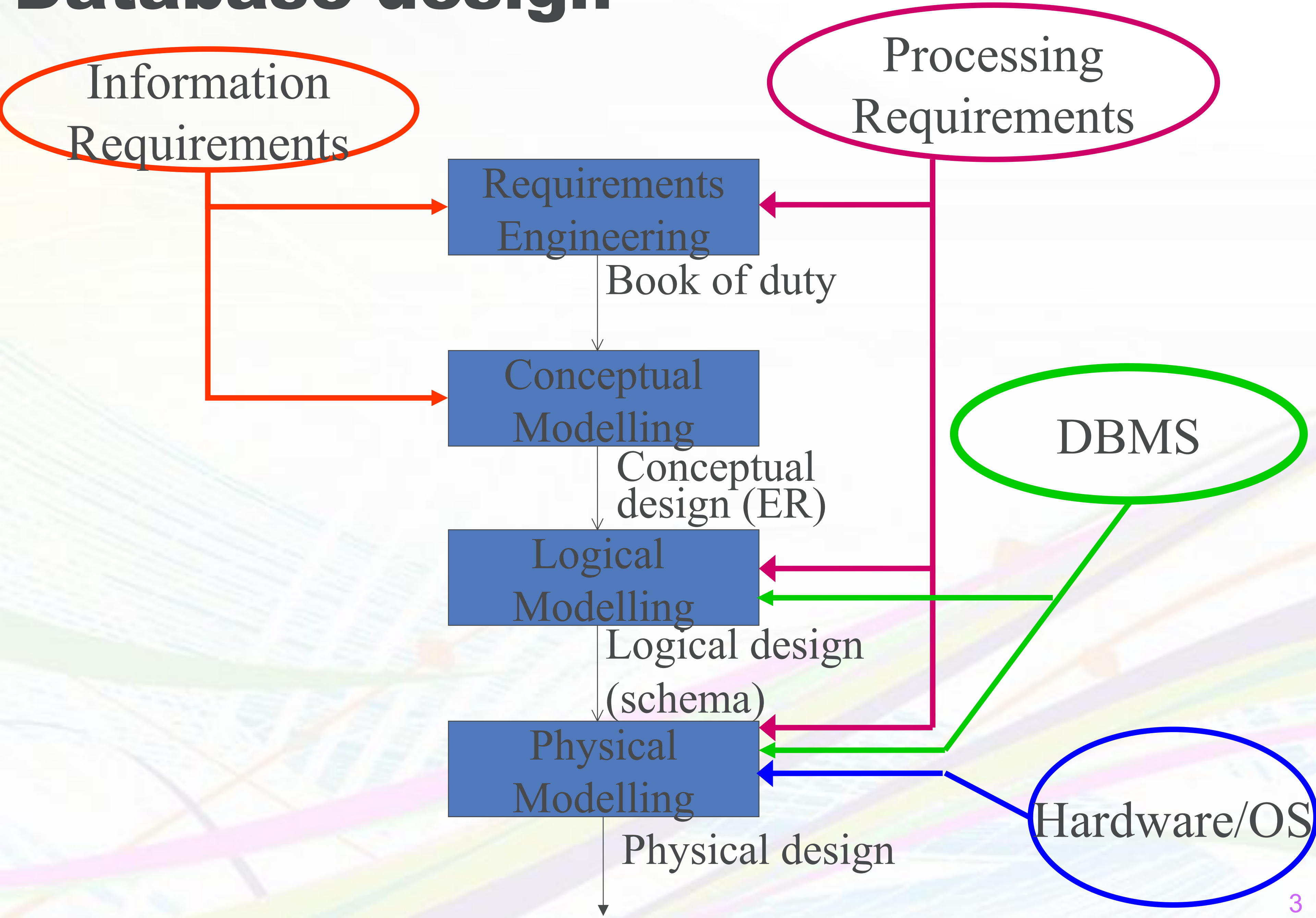
Entity–Relationship Model

Database Design

Database Abstraction Layers

1. Conceptual Model
2. Logical Model
3. Physical Database Design

Database design



Book of Duty

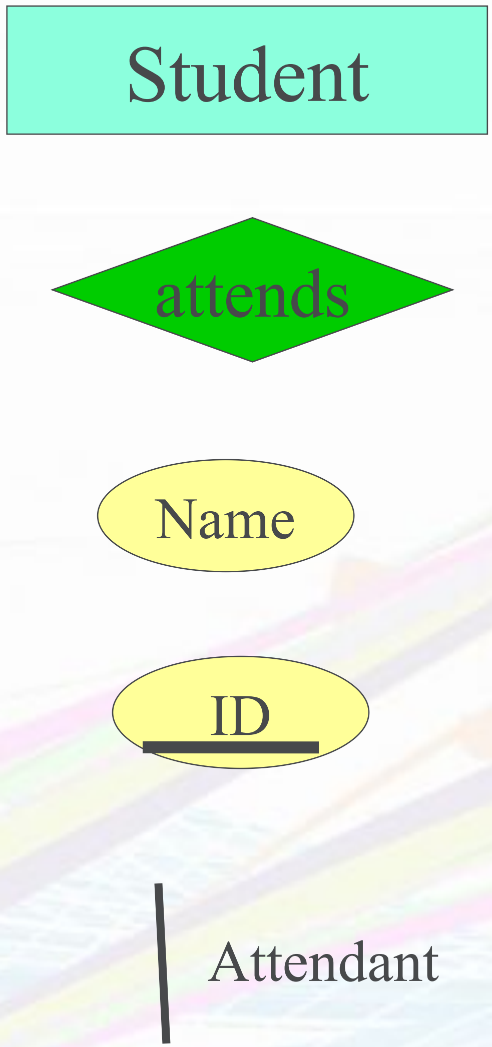
- **Describe information requirements (Entities)**
  - Objects used (e.g., student, professor, lecture)
  - Domains of attributes of objects
  - Identifiers, references / relationships
- Describe processes (relations)
  - E.g., examination, degree, register course
- Describe processing requirements (constraints)
  - Cardinalities: how many students?
  - Distributions: skew of lecture attendance
  - Workload: how often a process is carried out
  - Priorities and service level agreements

Entity/Relationship (ER) Models

- Entity
- Relationship
- Attribute
- Key
- Role

Entity/Relationship (ER) Models

- Entity
- Relationship
- Attribute
- Key
- Role



Student

attends

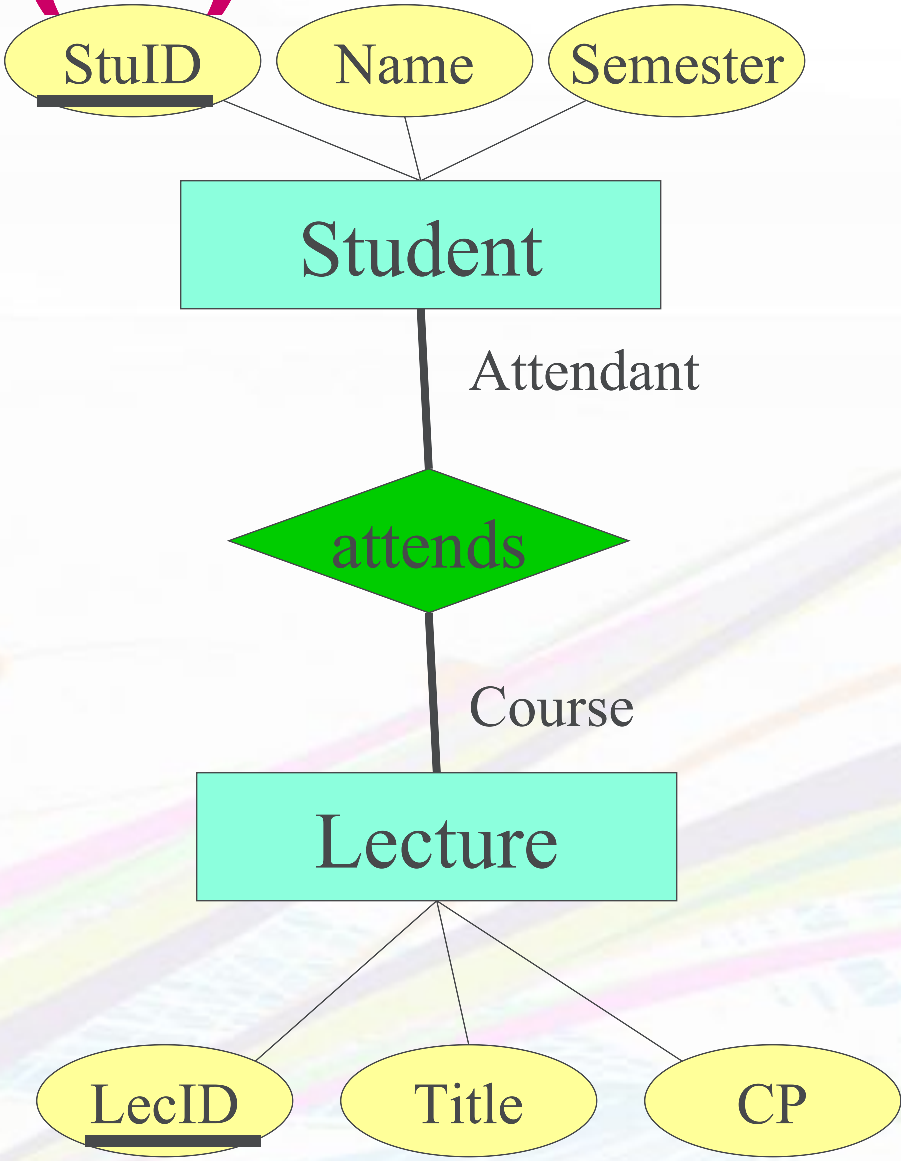
Name

ID

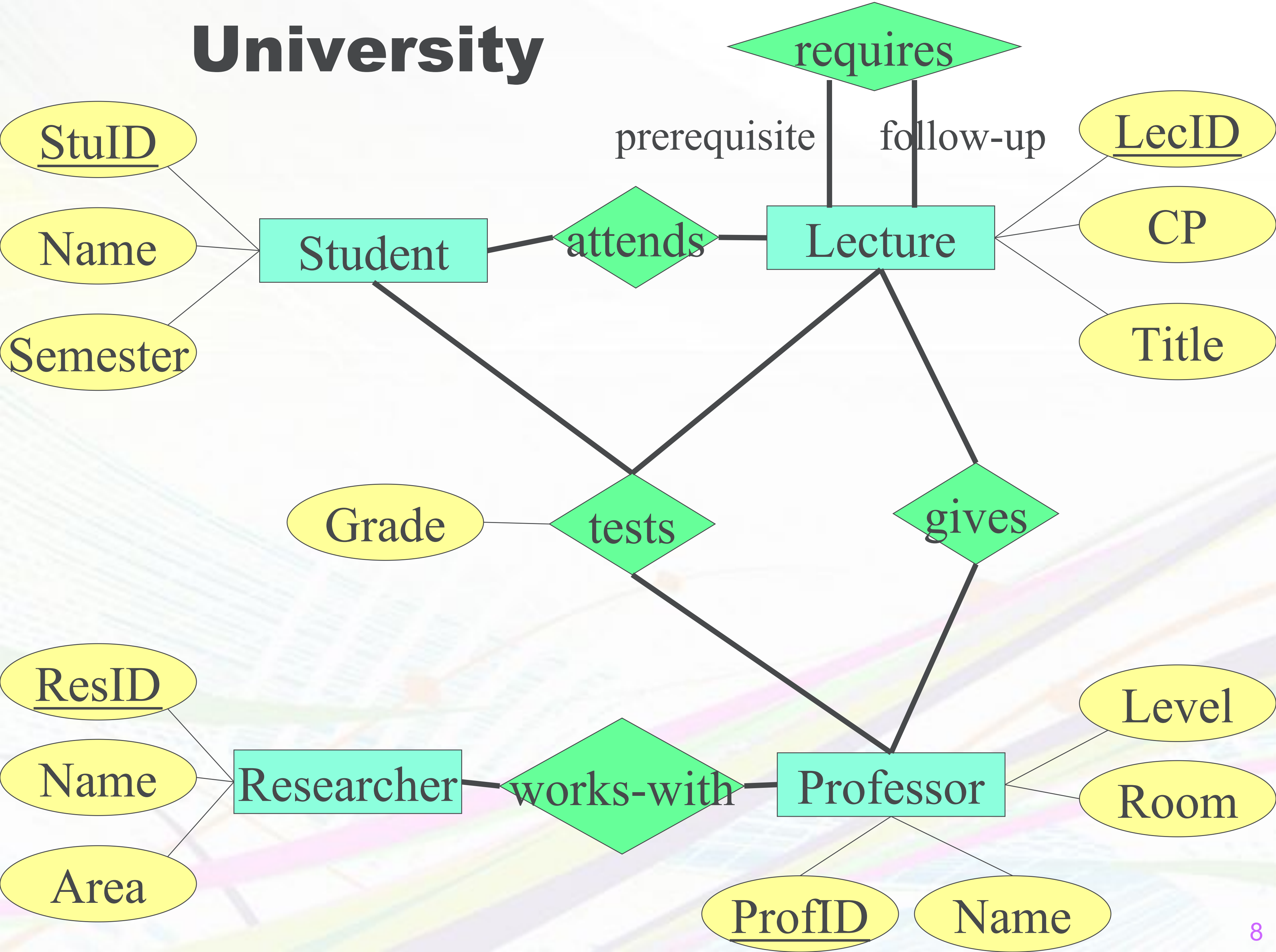
Attendant

Entity/Relationship (ER) Models

- Entity
- Relationship
- Attribute
- Key
- Role



University


requires
prerequisite
follow-up
StuID
Name
Semester
Student
attends
Lecture
LecID
CP
Title
Grade
tests
gives
ResID
Name
Area
Researcher
works-with
Professor
Level
Room
ProfID
Name

Natural Language Version

- Students have a StuID, Name and Semester. The StuID identifies a student uniquely.
- Lectures have a LecID, CP and Title. The LecID identifies a lecture uniquely.
- Professors have a ProfID, Name, Level and Room. The ProfID identifies a professor uniquely.
- Researchers have a ResD, Name and (research) Area. The AssiID identifies an assistent uniquely.
- Students attend lectures.
- Lectures can be prerequisites for other lectures.
- Professors give lectures.
- Researchers work with professors.
- Students are tested by professors about lectures. Students receive grades as part of these tests.
- Is this the only possible interpretation?

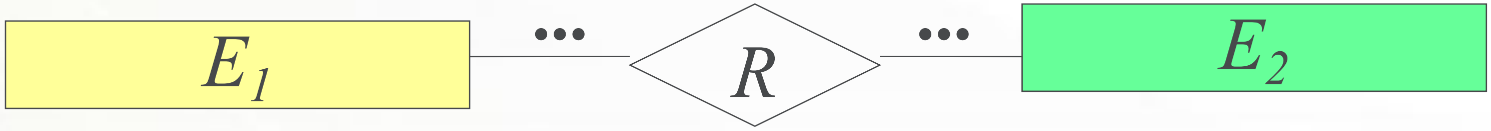
Why ER?

- Advantages
  - ER diagrams are easy to create
  - ER diagrams are easy to edit
  - ER diagrams are easy to read (from the layman)
  - ER diagrams express all information requirements

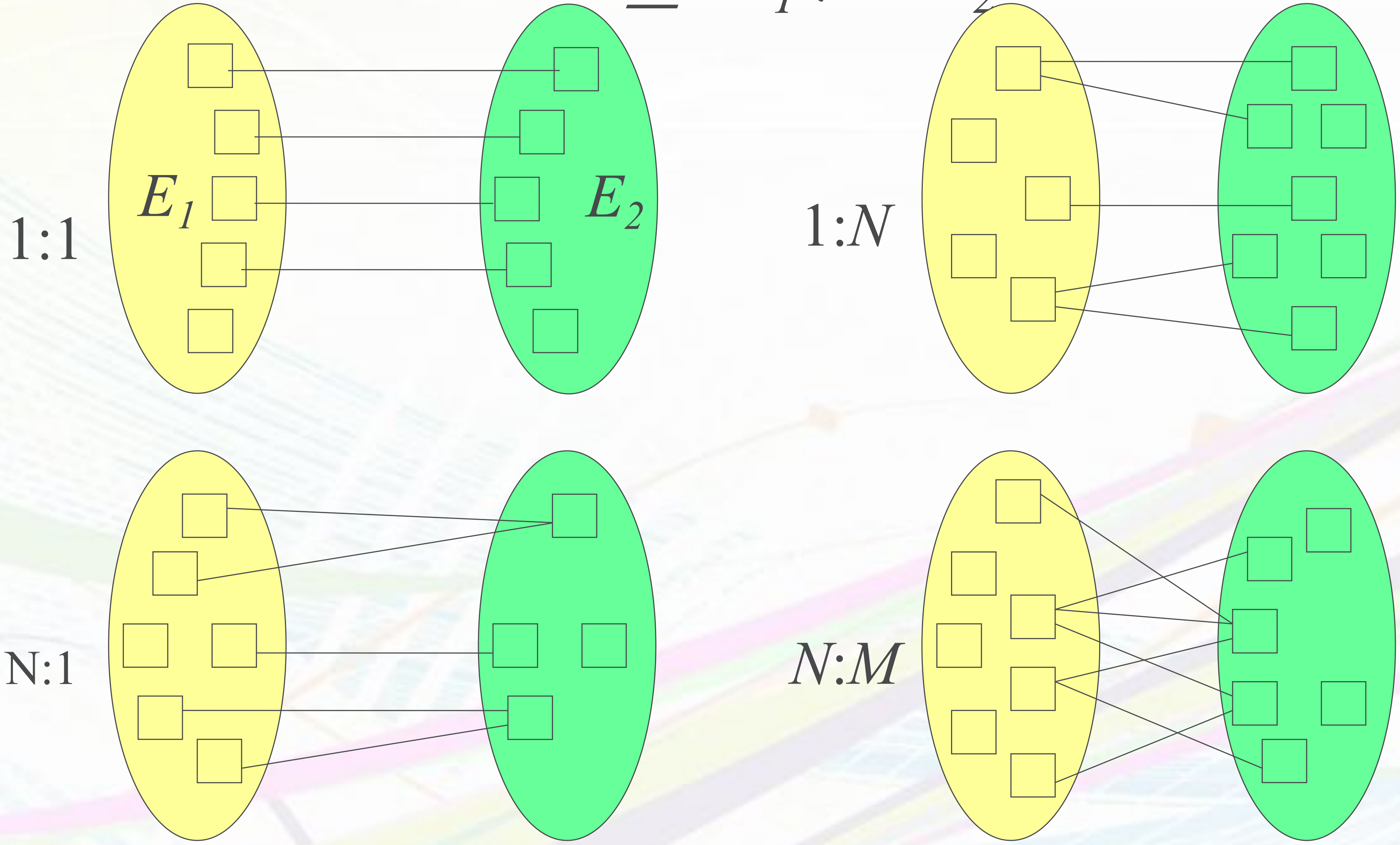
- Other aspects
  - Minimality
  - Tools (e.g., Visio)
  - Graphical representation

- General
  - Try to be concise, complete, comprehensible, and correct
  - *Controversy whether ER/UML is useful in practice?*
  - No controversy that everybody needs to learn ER/UML

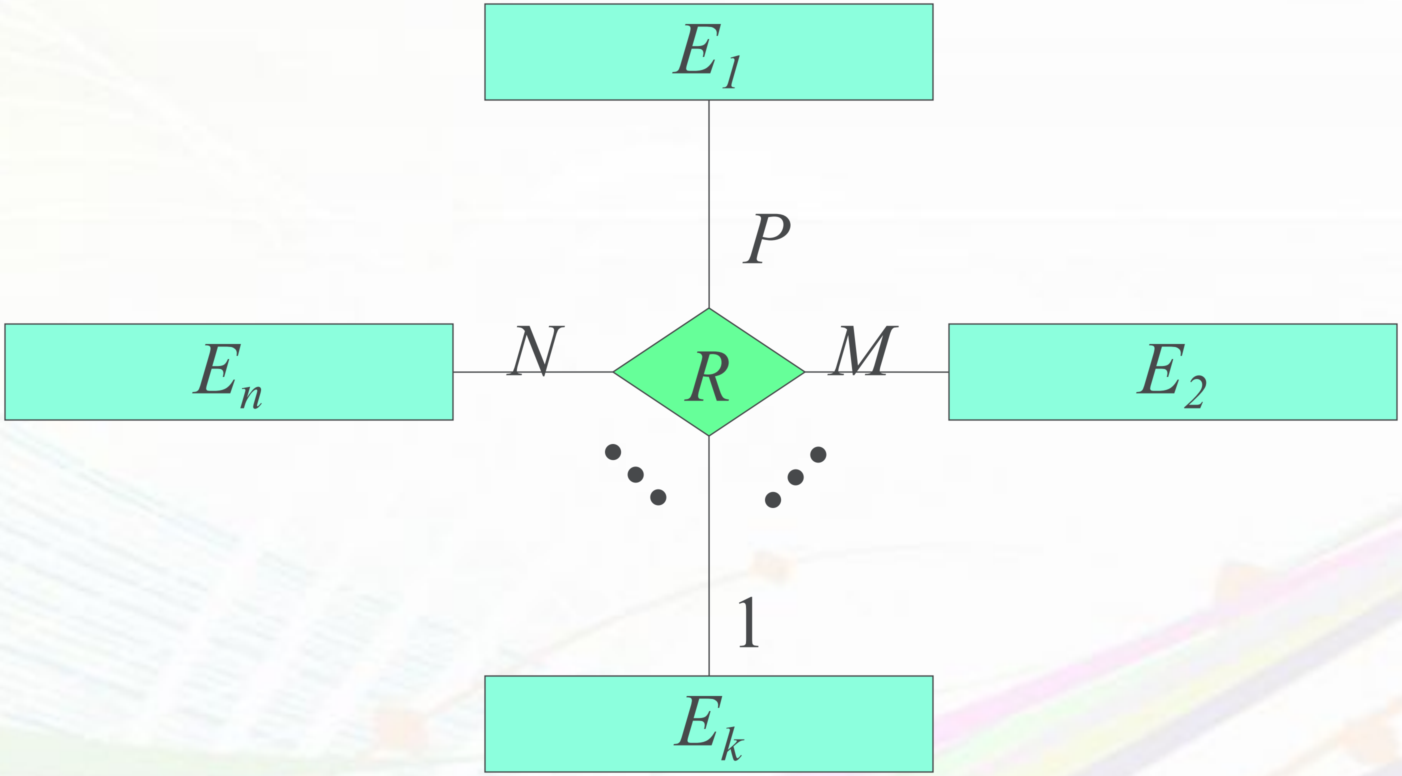
Functionalities



$$R \subseteq E_1 \times E_2$$

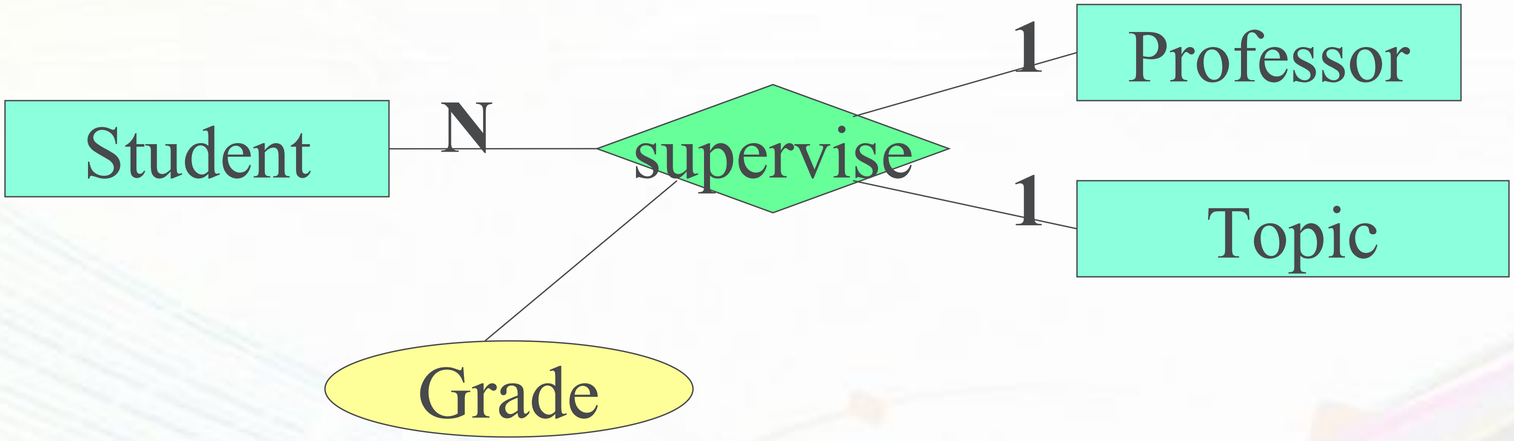


Functionalities of n-ary relationships



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

Example: *seminar*



supervise : Professor × Student → Topic

supervise : Topic × Student → Professor

Constraints

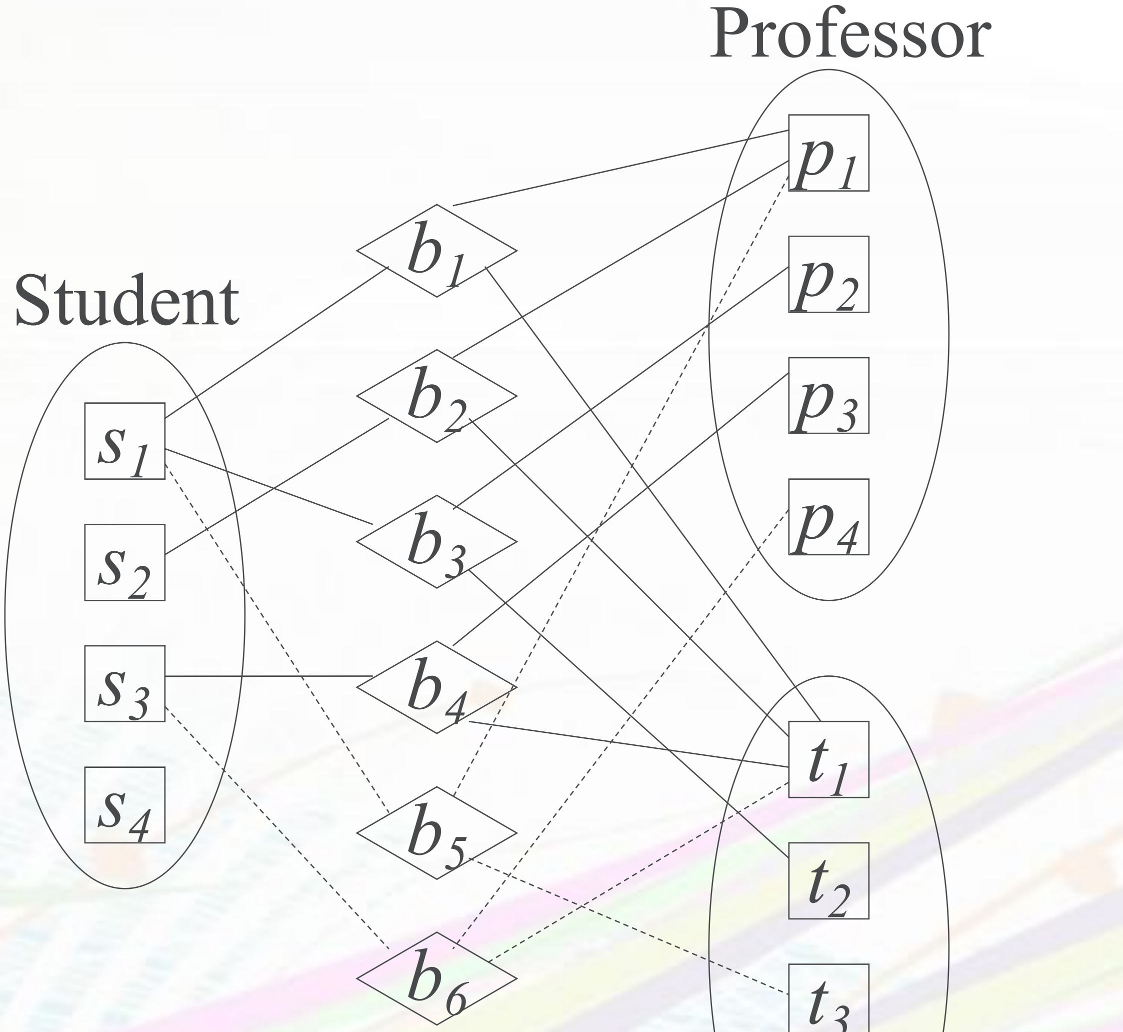
The following is not possible:

1. Students may only do at most one seminar with a prof.
2. Students may only work on a topic at most once.

The following is possible:

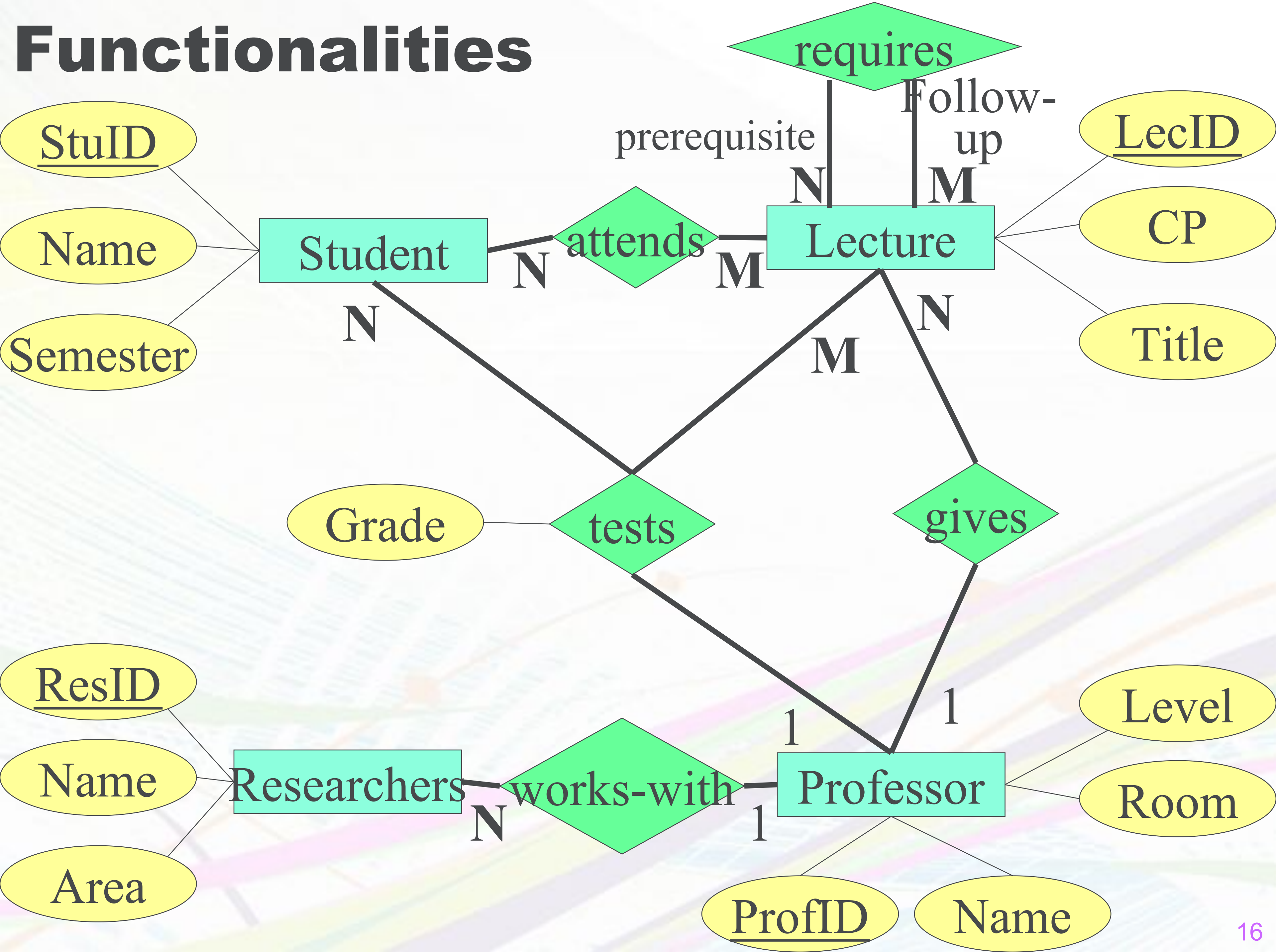
Profs may recycle topics and assign the same topic to several students.

Example



Dashed lines represent illegal references

Functionalities



Two Binary vs. One Ternary Relat.

- A thief steals a painting as part of a theft.

Model as two binary relationships

Model as one ternary relationship

What is better?

ER: Rules of thumb

- Attribute vs. Entity

  Entity if the concept has more than one relationship

  Attribute if the concept has only one 1:1 relationship

- Partitioning of ER Models

  Most realistic models are larger than a page

  Partition by domains (library, research, finances, ...)

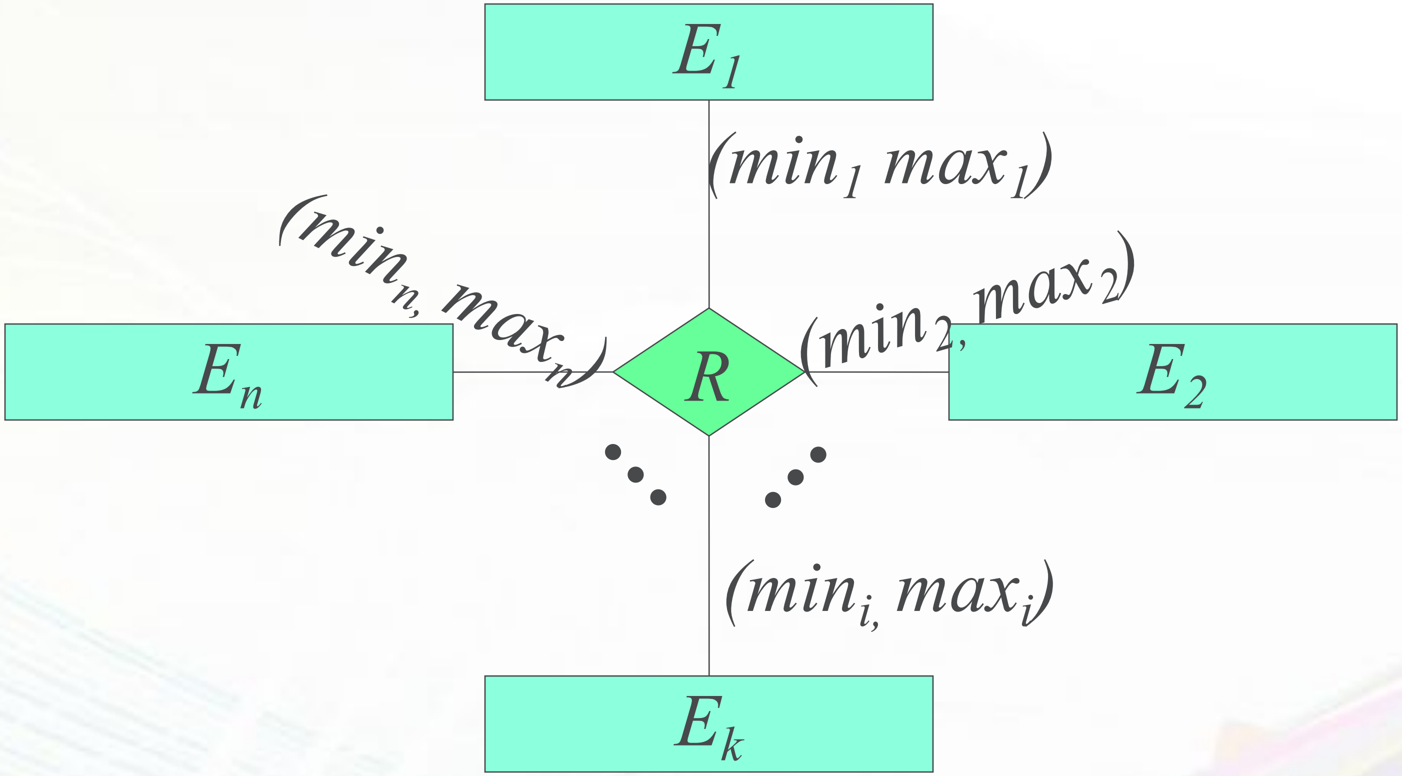
- Good vs. Bad models

  Do not model redundancy or tricks to improve performance

  Less entities is better (the fewer, the better!)

  Remember the C4 rule. (concise, correct, complete, compr.)

(min, max)-Notation

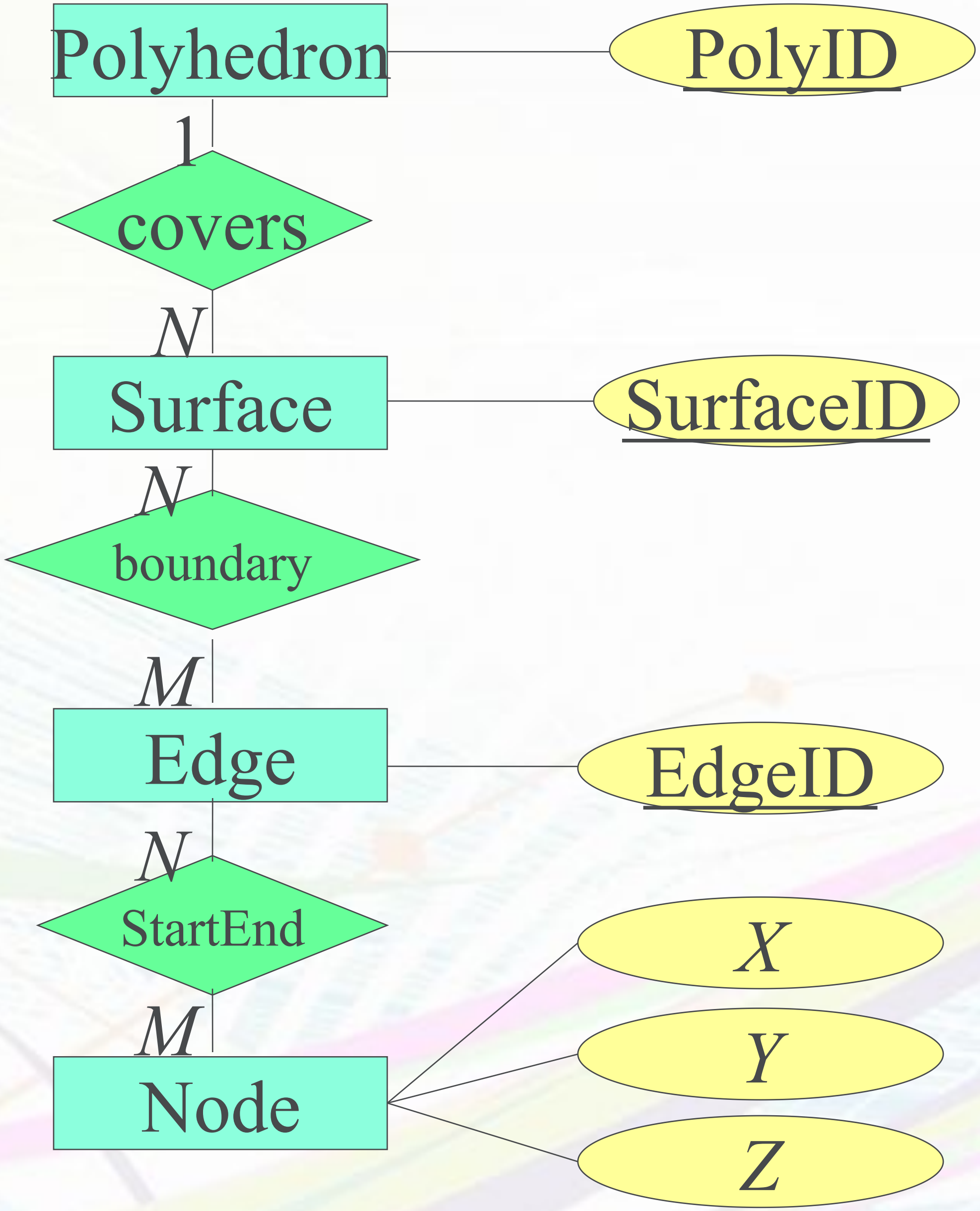


$R \subseteq E_1 \times \ldots \times E_i \times \ldots \times E_n$

For all $e_i \in E_i$:

- At least min_i records $(\ldots,\ e_i,\ \ldots)$ exist in R AND
- At most max_i records $(\ldots,\ e_i,\ \ldots)$ exist in R

Geometric Modelling



Polyhedron

PolyID

1

covers

N

Surface

SurfaceID

N

boundary

M

Edge

EdgeID

N

StartEnd

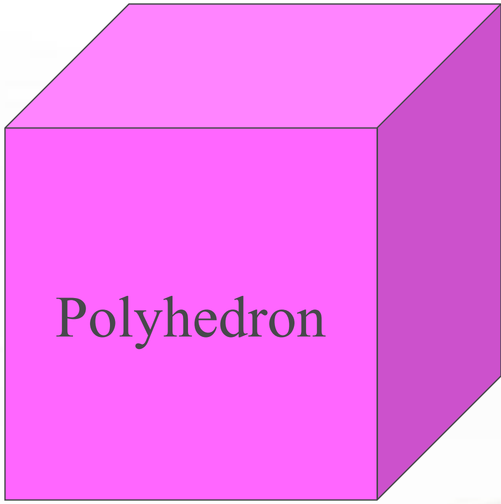
M

Node

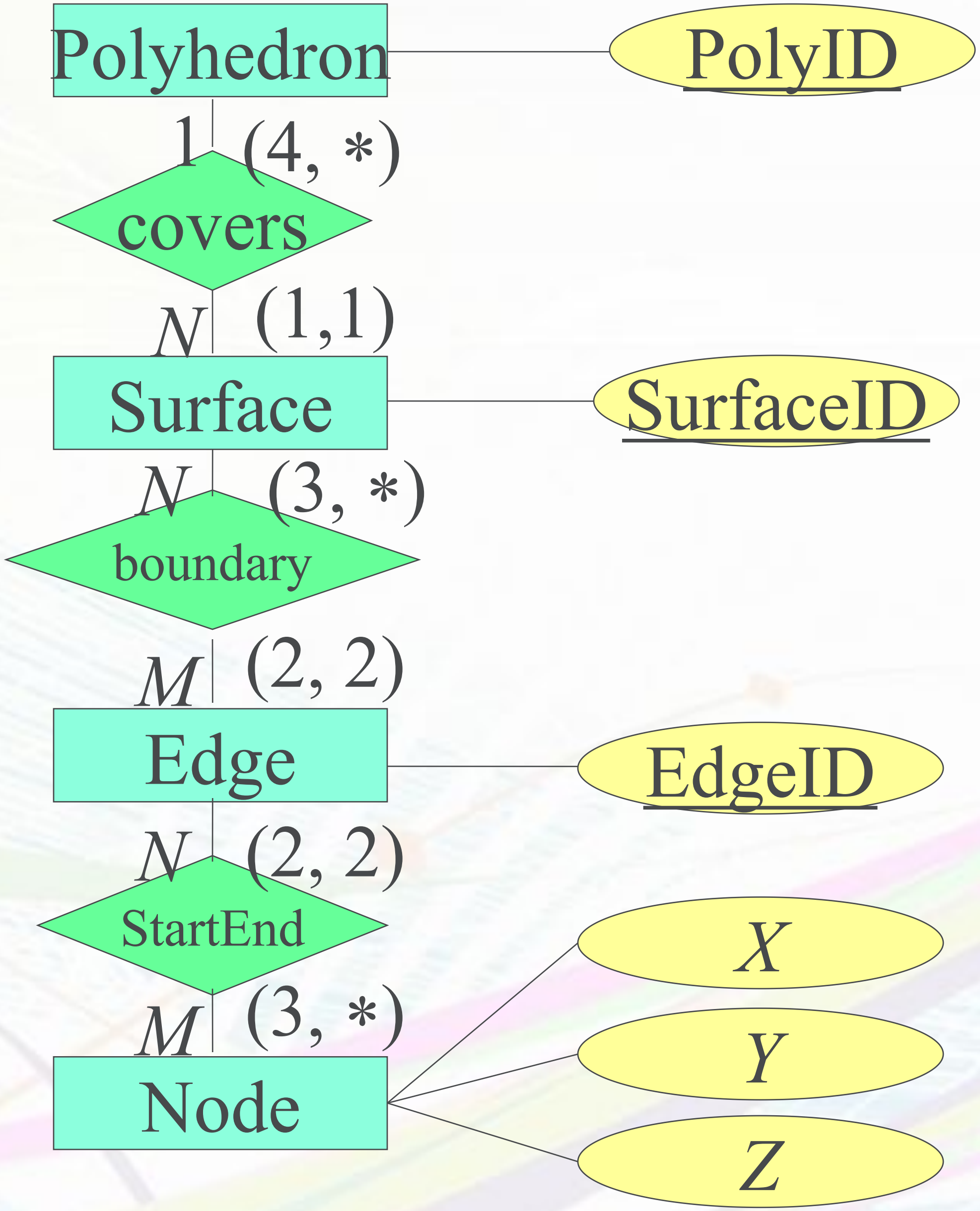
X

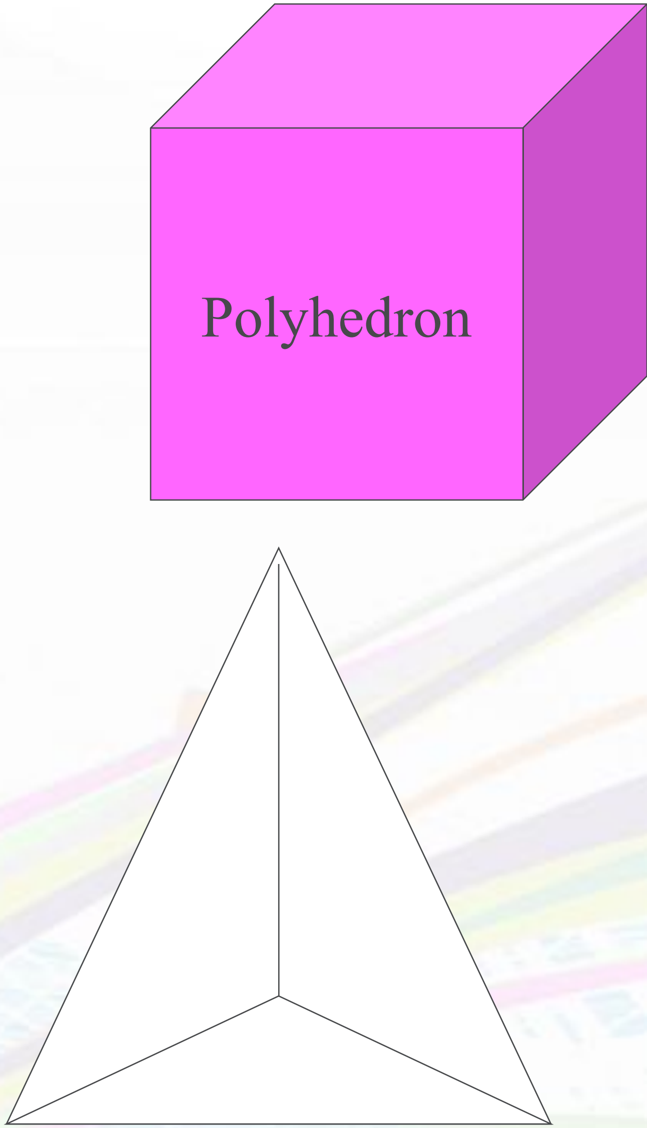
Y

Z

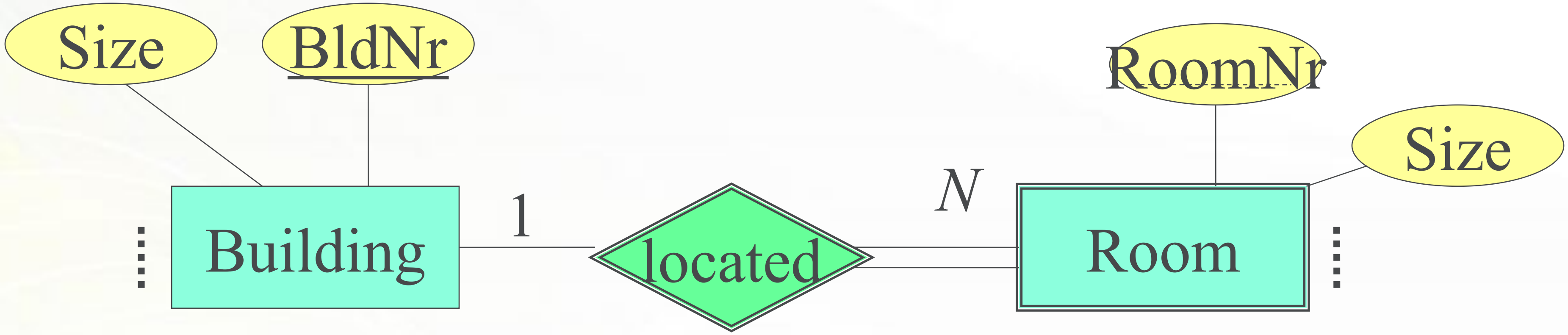


Geometric Modelling





Weak Entities



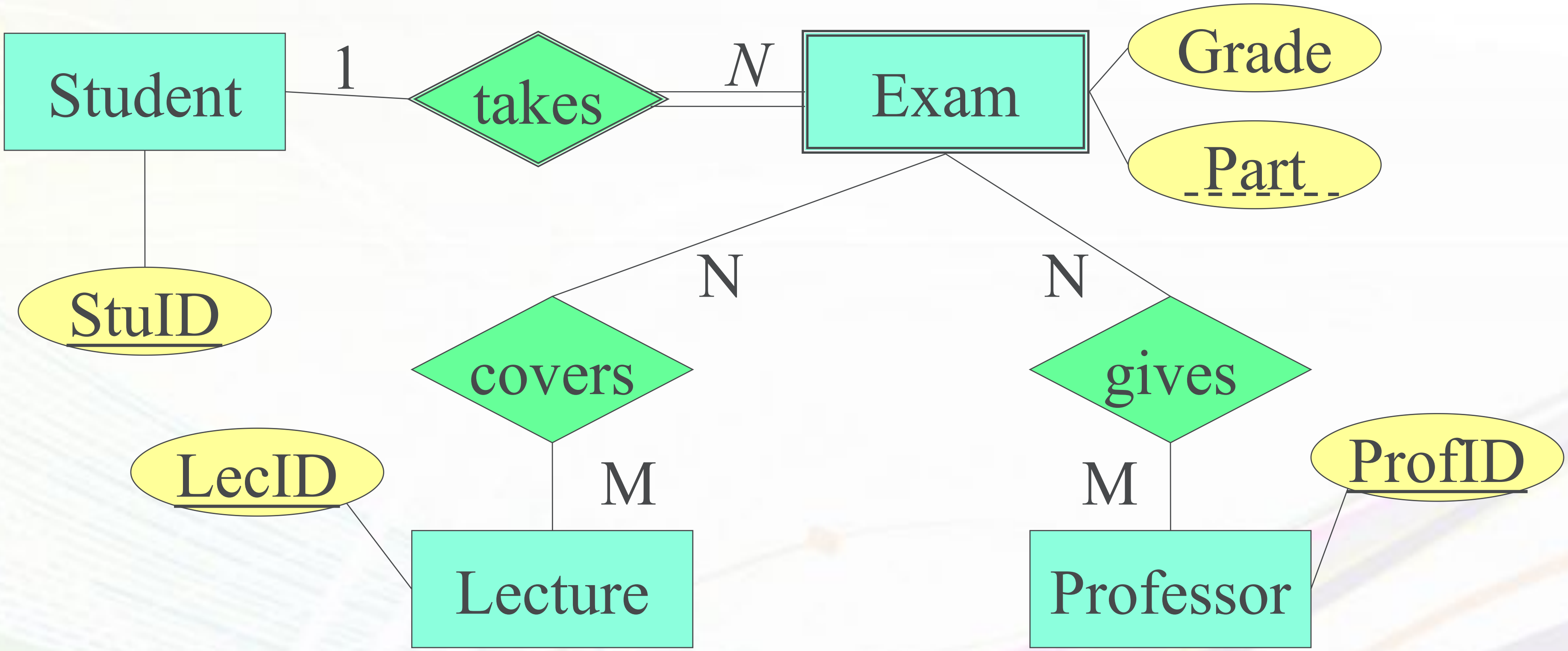
The existince of room depends on the existence of the associated building.

Why must such relationships be N:1 (or 1:1)?

RoomNr is only unique within a building.

Key of a room: BldNr **and** RoomNr

Exams depend on the student



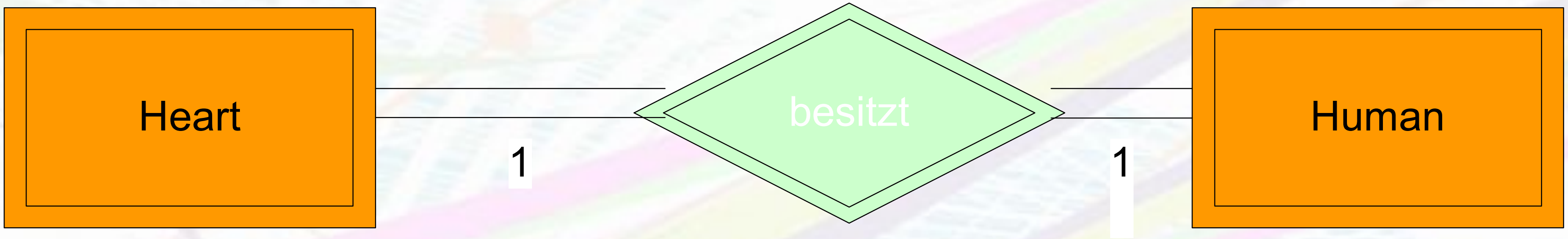
Can the existence of an entity depend on several other entities? (E.g., exam on student and prof?)

Corner Case 1

- A human cannot exist without a heart.
- A heart cannot exist without a human.
- Anne lives on Bob‘s heart. Bob lives on Anne‘s heart. Possible?
- Heart transplantation – possible?

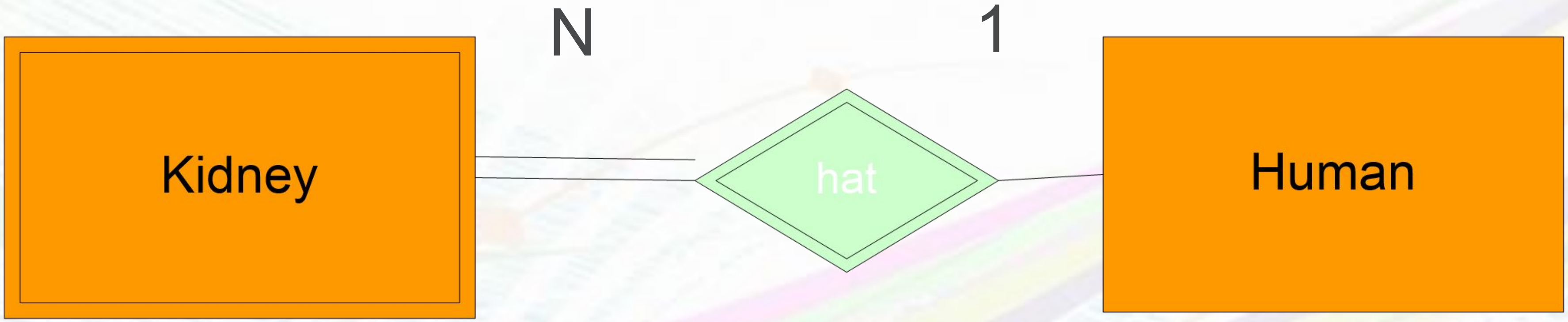
ER describes possible worlds and their rules

ER does not describe legal transitions!

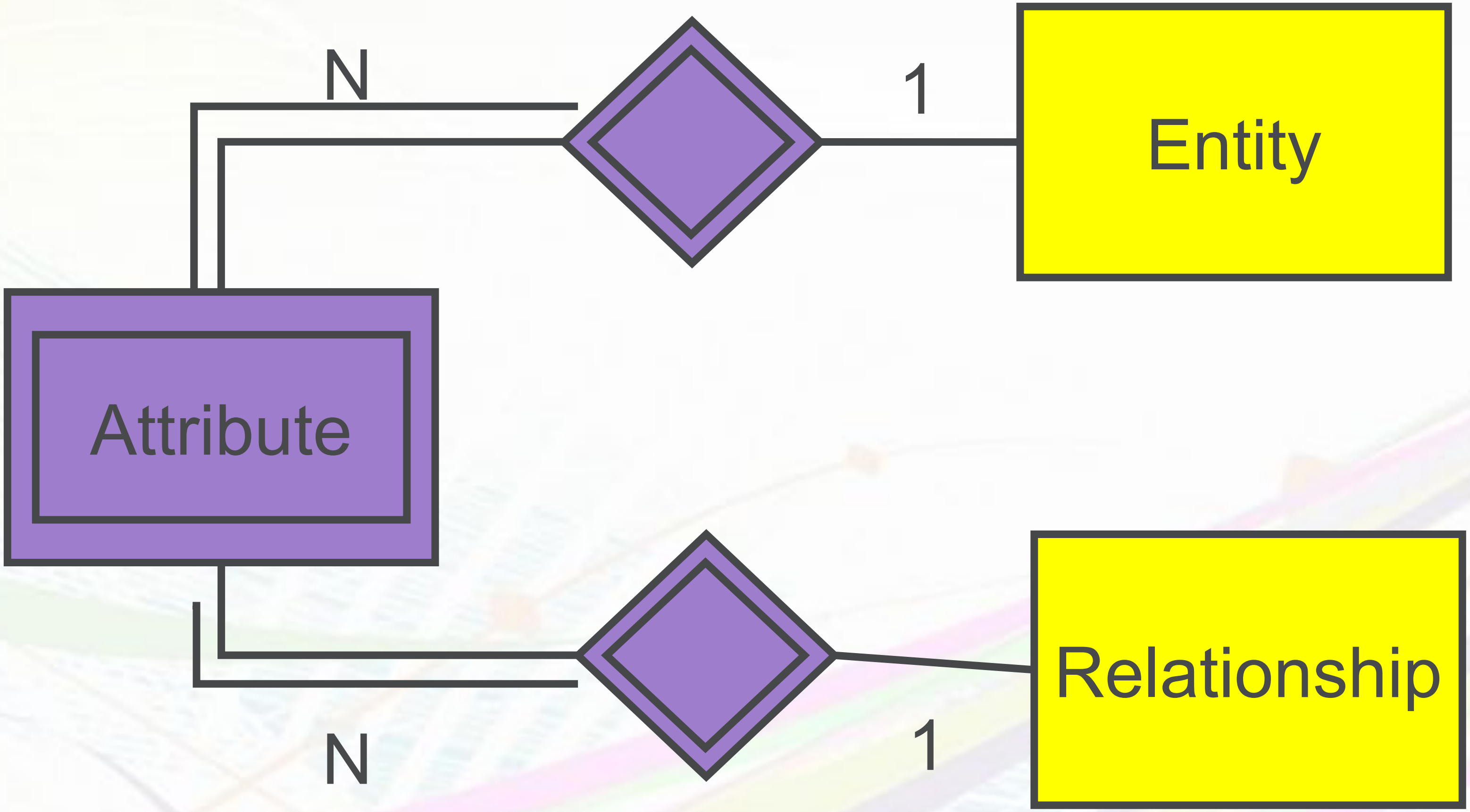


Corner Case 2

- A human can only survive with ***at least one*** kidney.
- A relationship can only survive with ***all*** its entities.
- **Not expressible with ER!** (Why not?)

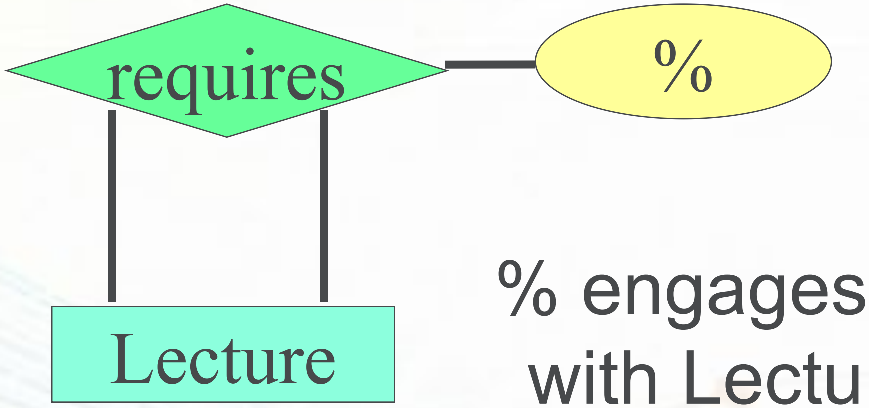


Why is this a bad model?



Why is this example so itchy?

- Is the following "instance" legal?



% engages in relationship with Lecture via requires

Answer: Yes!

When "requires" dies, "%" dies, too

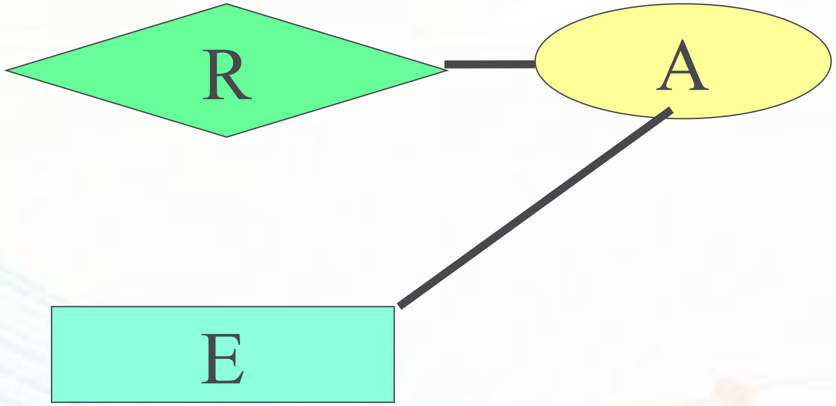
When "lecture" dies, "%" dies, too

but, not cleanly modelled – transitivity of weakness

N.B. weakness of "Relationship" cannot be modelled!!!

Why is this example so itchy?

- Is the following "instance" legal?



Answer: Yes!
when "R" dies, "A" dies, too
when "E" dies, "A" dies, too
no way to model that it is an **"either or"** relationship
yet another weakness of ER (lack of negation)

Solution

- Model attributes as two weak entities

  attributes of relationships

  attributes of entities

- (Give up on relationships as weak entities)
- Not perfect because redundant

  but the pricest way to model ER as ER