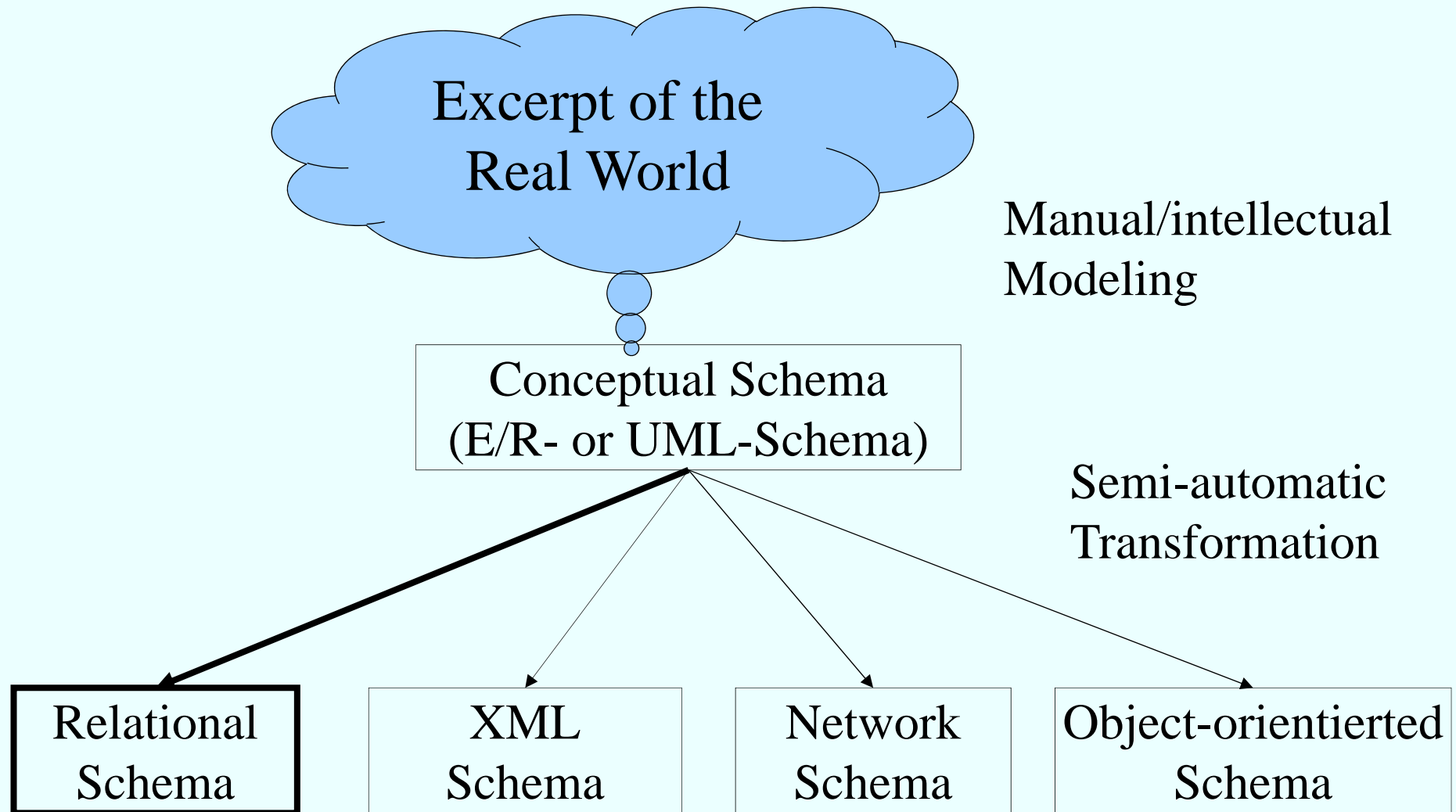
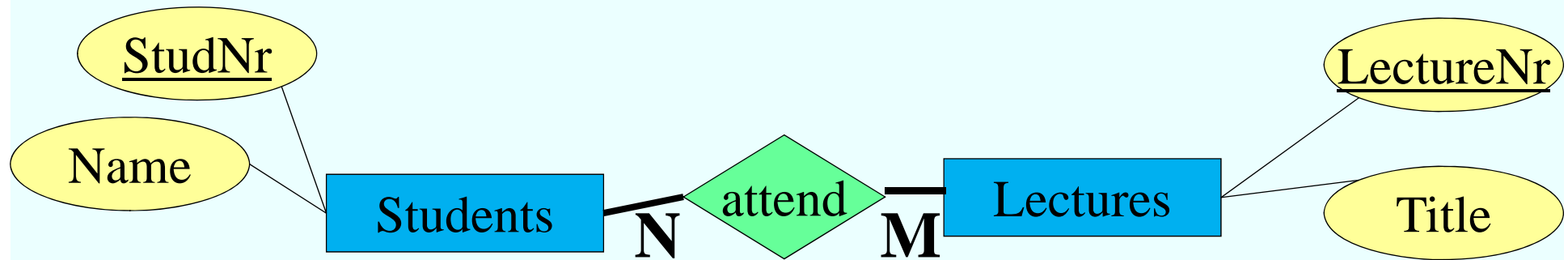


# Data modeling



# Relational Data Model



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

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

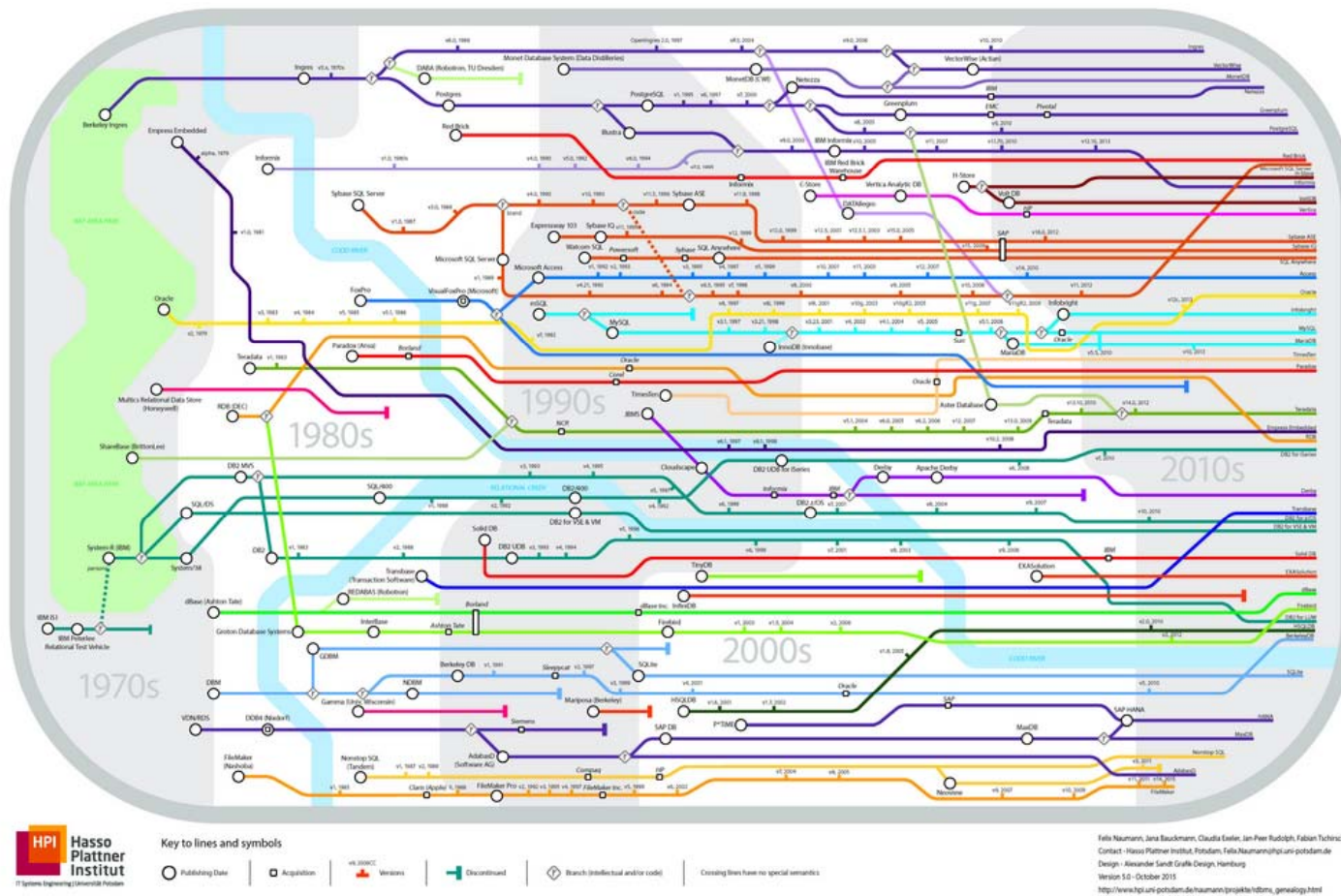
Lectures	
Lecture Nr	Title
5001	Grundzüge
5022	Glaube und Wissen
...	...

# Development of relational DBMS

- Codasyl, beginning 1960: network data model
- IMS, mid 1960: hierarchical data model
- Ted Codd, 1970: foundation relational data model
- System R, mid 1970: relational DBMS  
(research prototype)
- Genealogy poster:  
[www.hpi.de/naumann/projects/rdbms-genealogy.html](http://www.hpi.de/naumann/projects/rdbms-genealogy.html)

# Development of relational DBMS

Genealogy of Relational Database Management Systems



# Development of relational DBMS

## Commercial relational DBMS, Focus OLTP

- Oracle V2, end 1970
- Ingres (Berkeley), end 1970 → PostgreSQL
- SQL/DS, beginning 1980: IBM → DB2
- MS SQL Server, 1990 (out of Sybase)
- MySQL, end 1990

→ since end 1990: Object relational extensions

# Development of relational DBMS

## Specialised systems:

- Object orientierted DBMS (since end 1980)
- XML DBMS (since end 1990)
- Main memory DBMS (since end 1990)
- Analytical DBMS (Focus OLAP) (since beginning 2000)

## Hybrid systems:

- OLTP & OLAP: HyPer, SAP HANA (main memory DBMS, multi processor architecture)

# Foundation relational model

Let  $D_1, D_2, \dots, D_n$  be **domains**

**Relation:**  $R \subseteq D_1 \times \dots \times D_n$

e.g.: *Phone book*  $\subseteq$  *string*  $\times$  *string*  $\times$  *integer*

**Tupel:**  $t \in R$

e.g.:  $t = (\text{„Mickey Mouse“}, \text{„Main Street“}, 4711)$

**Schema:** *determines the structure of the stored data*

e.g.: *Phone book*:  $\{[Name: string, Street: string, \underline{phone\#: integer}]\}$

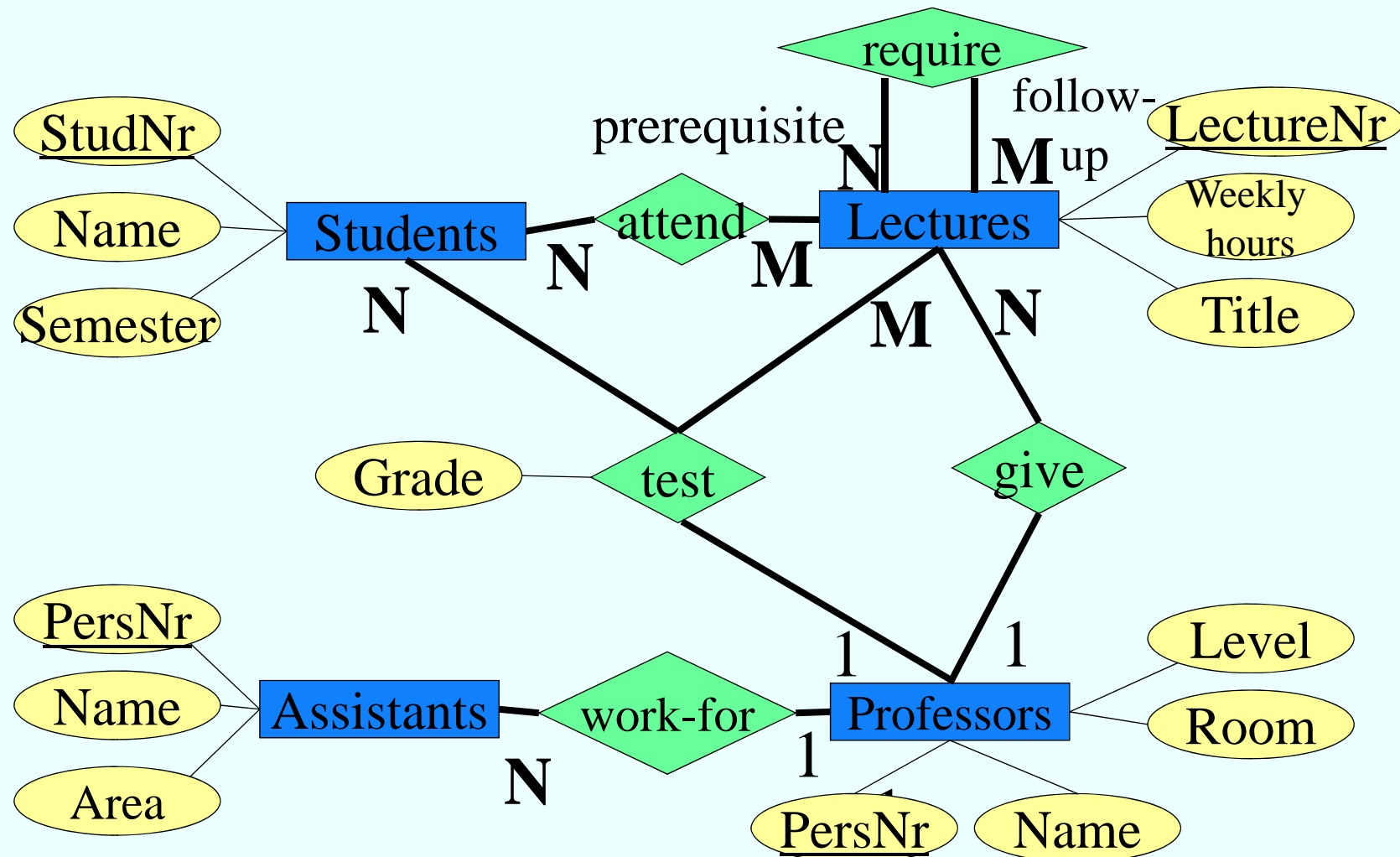
# Relation

Phone book		
Name	Street	<u>Phone#</u>
Mickey Mouse	Main Street	4711
Minnie Mouse	Broadway	94725
Donald Duck	Broadway	95672
...	...	...

- **Instance:** current state of the data base
- **Candidate Key:** minimal set of attributes whose values uniquely identify a tuple
- **Primary Key:** underlined
  - One of the candidate keys is chosen as primary key
  - Has a special meaning when referencing tuples



# University Schema



# Transformation rules (1)

## Entity sets to relations:

Entity set  $E$  with attributes  $A_i$  out of domains  $D_i$  ( $1 \leq i \leq k$ )  
 $\Rightarrow$   $k$ -ary relation  $E(A_1:D_1, \dots, A_k:D_k)$ .

Keep key attributes

# Relational depiction of Entity sets

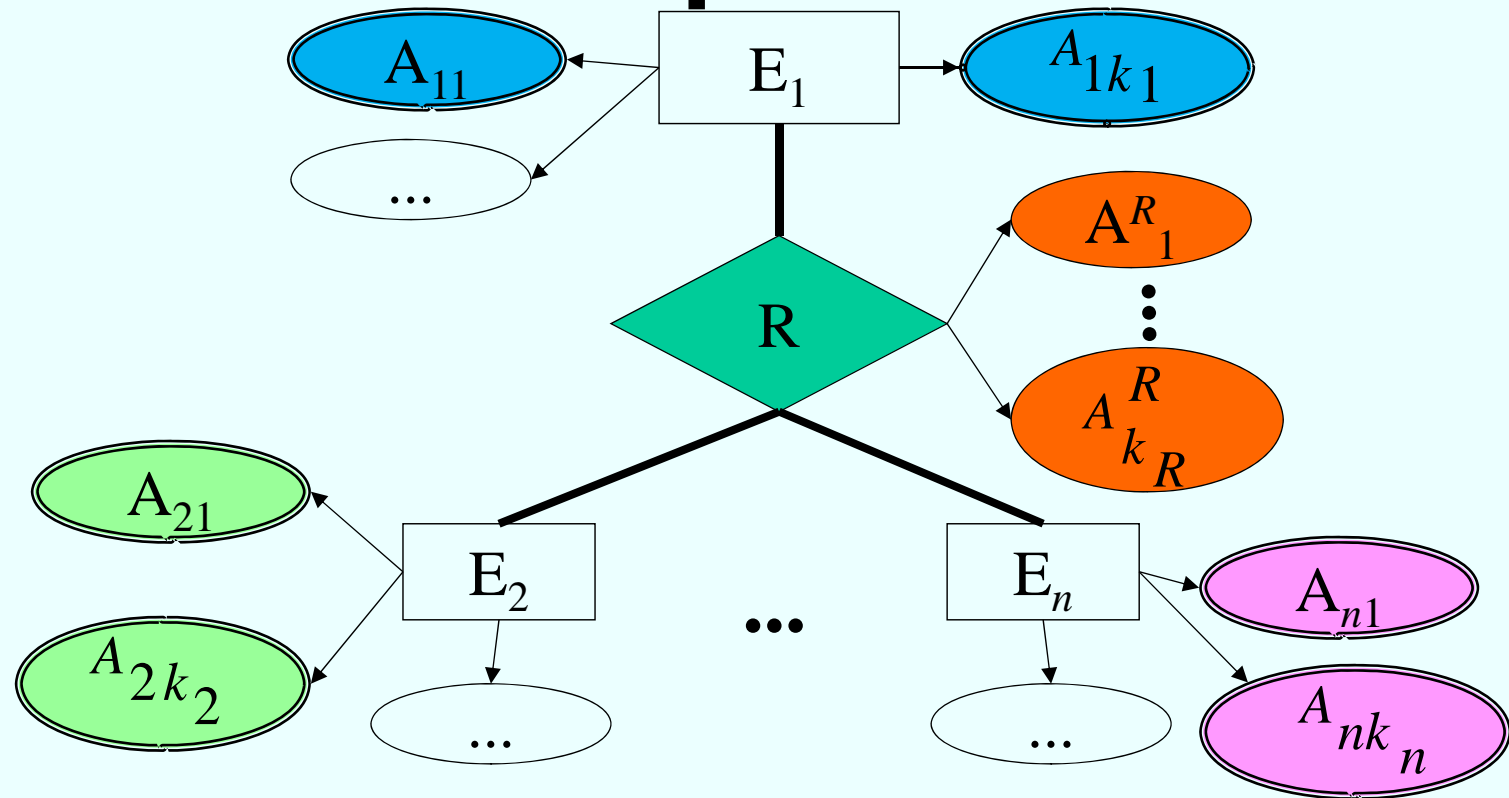
**Students:** {[StudNr:integer, *Name: string*, *Semester: integer*]}

**Lectures:** {[LectureNr:integer, *Title: string*, *WeeklyHours: integer*]}

**Professors:** {[PersNr:integer, *Name: string*, *Level: string*, *Room: integer*]}

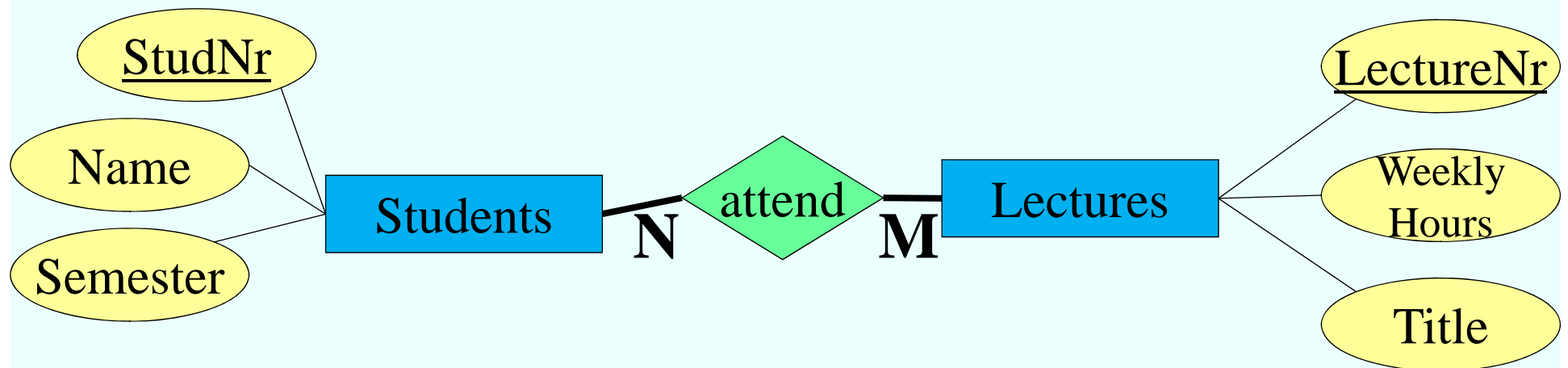
**Assistants:** {[PersNr:integer, *Name: string*, *Area: string*]}

# Relational depiction of Relationships



$$R: \left\{ \underbrace{[A_{11}, \dots, A_{1k_1}]}_{\text{Key of } E_1}, \underbrace{[A_{21}, \dots, A_{2k_2}]}_{\text{Key of } E_2}, \dots, \underbrace{[A_{n1}, \dots, A_{nk_n}]}_{\text{Key of } E_n}, \underbrace{[A_{11}^R, \dots, A_{k_R}^R]}_{\text{Attributes of } R} \right\}$$

# Relationships of our example schema



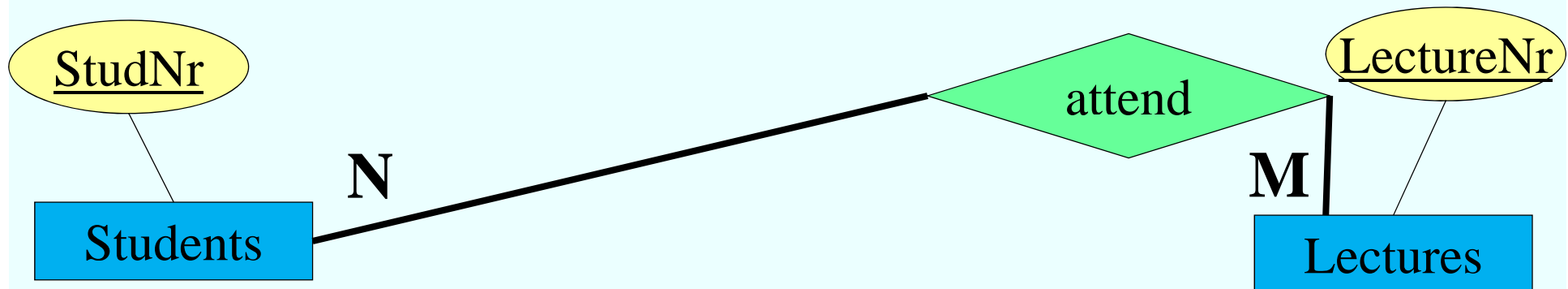
**attend** (N:M): {[StudNr: integer, LectureNr: integer]}

# Instance of Relationship *attend*

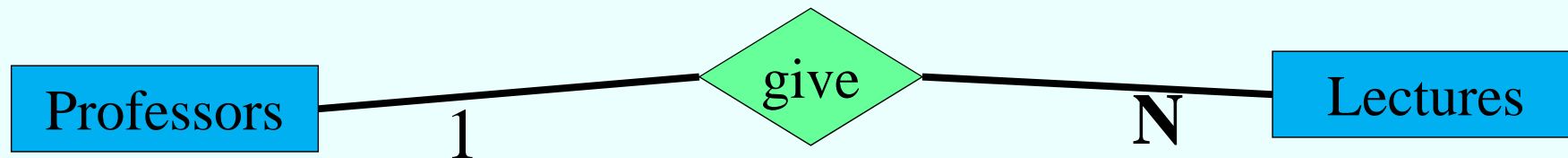
Students	
<i>StudNr</i>	...
26120	...
27550	...
...	...

attend	
StudNr	LectureNr
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
...	...

Lectures	
Lecture Nr	...
5001	...
4052	...
...	...



# Relationships of our example schema



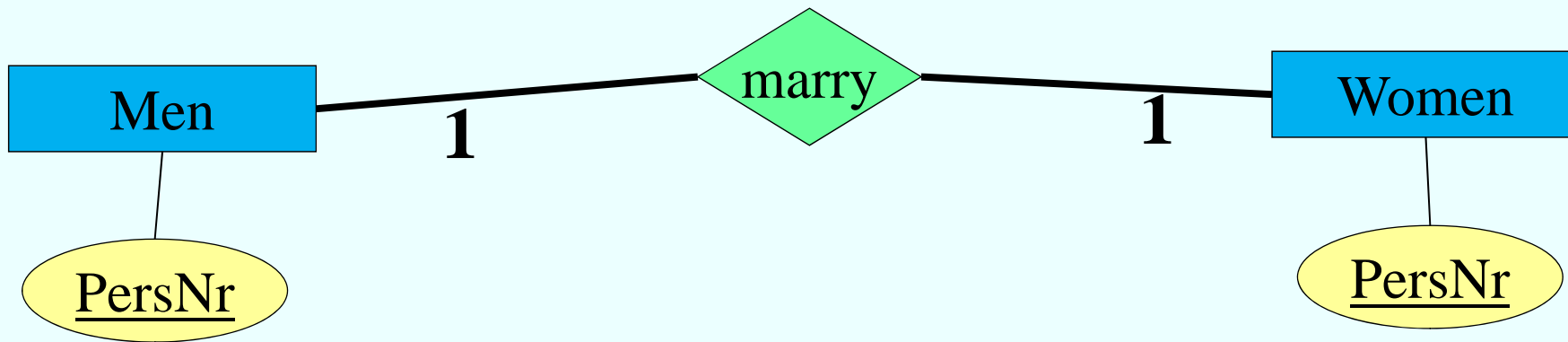
**give (1:N):** {[PersNr: integer, LectureNr: integer]}

**work\_for (N:1):** {[AssistantsPersNr: integer,  
*ProfPersNr: integer*]}

**require (N:M):** {[Predecessor: integer, Successor: integer]}

**test (N:M:1):** {[StudNr: integer, LectureNr: integer,  
PersNr: integer, Grade: decimal]}

# 1:1 Relationship?



**marry (1:1): {[??]}**



# Refined transformation rules

## Relationships to Relations (cont'd):

### 1:1-Relationship:

Relationship R between 2 Entities E and F.

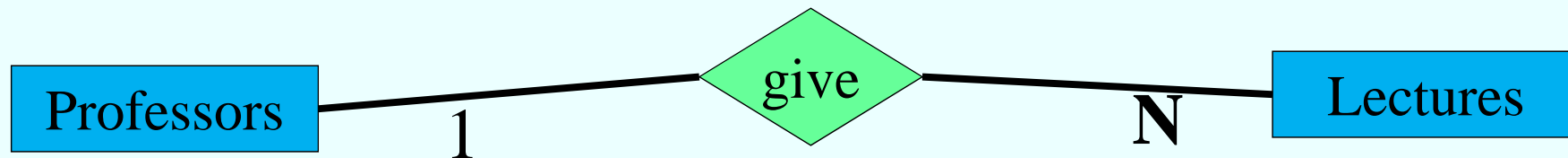
⇒ no Relation R, instead primary key of E in Relation in F or vice versa.

### 1:N-Relationship:

Relationship R between 2 Entities E and F.

⇒ no Relation R, instead primary key of E in Relation F als Fremdschlüssel aufnehmen. In case R has own attributes incorporate them into F.

# Refined relational schema



## 1:N-Relationship

Initial draft

**Professors** : {[PersNr, Name, Level, Room]}

**Lectures** : {[LectureNr, Title, WeeklyHours]}

**give**: {[PersNr , LectureNr]}

# Refined relational schema

## 1:N-Relationship

Initial draft

**Professors** : {[PersNr, Name, Level, Room]}

**Lectures** : {[LectureNr, Title, WeeklyHours]}

**give**: {[PersNr , LectureNr]}

Refinement by subsumption

**Professors** : {[PersNr, Name, Level, Room]}

**Lectures** : {[LectureNr, Title, WeeklyHours,  
*given\_by*]}

**Rule:** Relations with the same primary key can be subsumed  
... but only those and no others!

# Instance of *Professors* and Lectures

Professors			
PersNr	Name	Level	Room
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

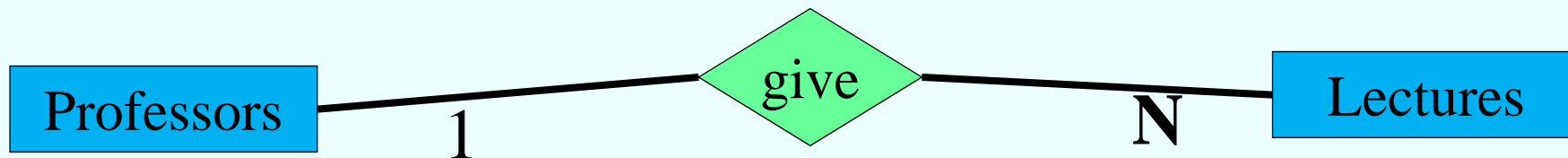
Lectures			
Lecture Nr	Title	Weekly Hours	given_by
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
...	...	...	...



# Why not the other way round?

Professors				
PersNr	Name	Level	Room	gives
2125	Sokrates	C4	226	5041
2125	Sokrates	C4	226	5049
2125	Sokrates	C4	226	4052
...	...	...	...	...
2134	Augustinus	C3	309	5022
2136	Curie	C4	36	??

Lectures		
Lecture Nr	Title	Weekly Hours
5001	Grundzüge	4
5041	Ethik	4
5043	Erkenntnistheorie	3
5049	Mäeutik	2
4052	Logik	4
5052	Wissenschaftstheorie	3
...	...	...



# Consequences → Anomalies

Professors				
PersNr	Name	Level	Room	gives
2125	Sokrates	C4	226	5041
2125	Sokrates	C4	226	5049
2125	Sokrates	C4	226	4052
...	...	...	...	...
2134	Augustinus	C3	309	5022
2136	Curie	C4	36	??

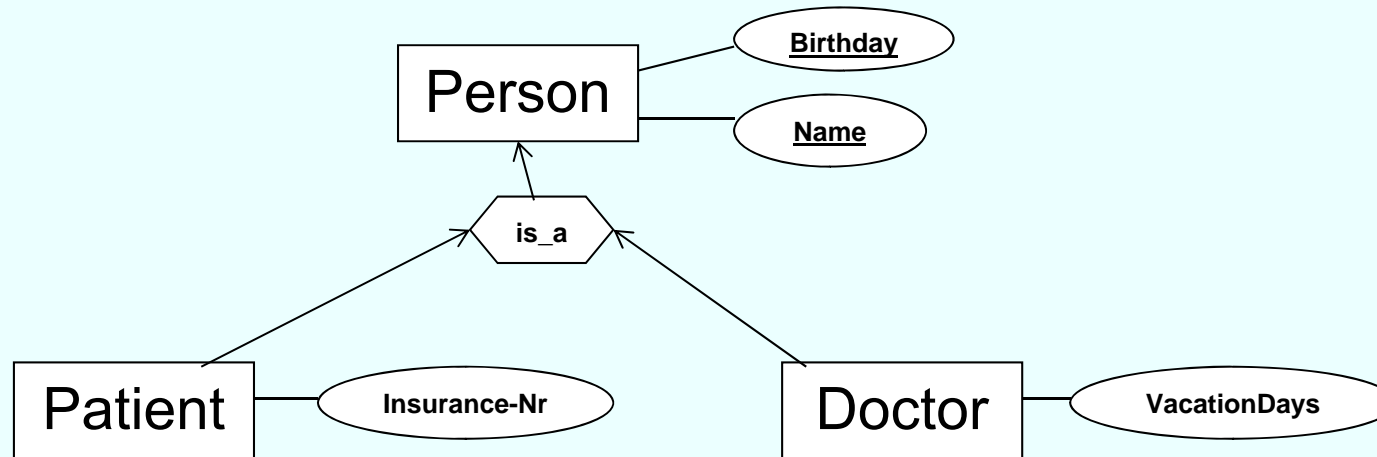
Lectures		
Lecture Nr	Title	Weekly Hours
5001	Grundzüge	4
5041	Ethik	4
5043	Erkenntnistheorie	3
5049	Mäeutik	2
4052	Logik	4
5052	Wissenschaftstheorie	3
...	...	...

Update anomaly: What if Sokrates moves?

Delete anomaly: What if „Glaube und Wissen“ is dropped?

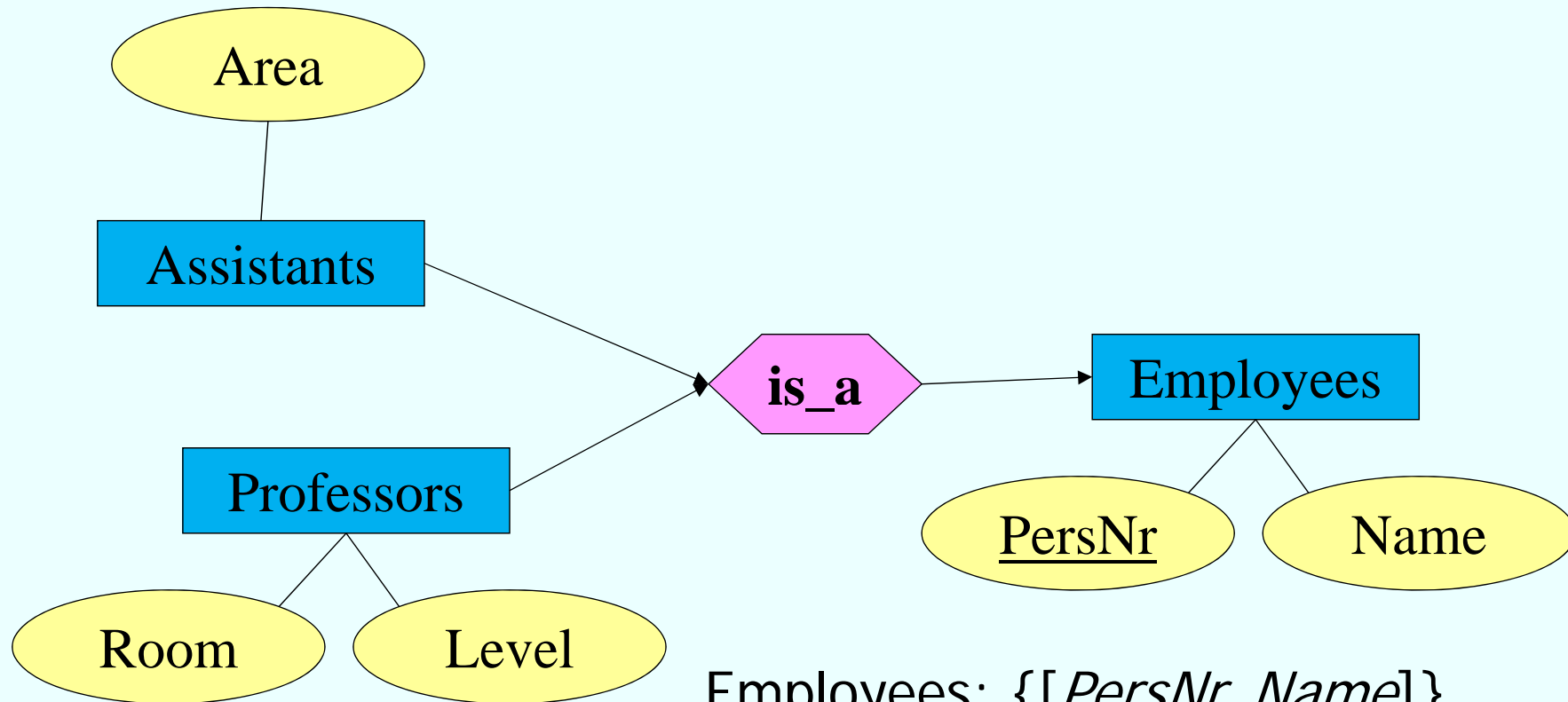
Insert anomaly: Curie is new and does not yet give lectures?

# Transformation Generalization



## Discussion

# Relational Modelling of the Generalization



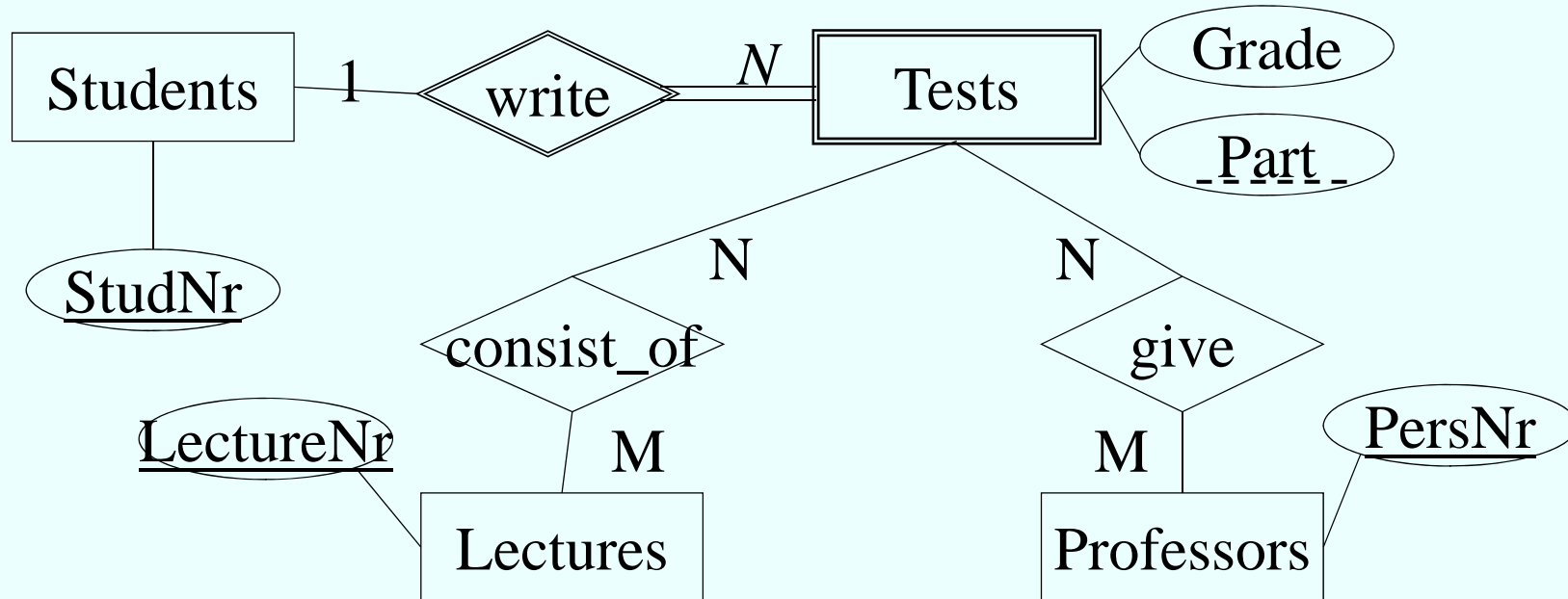
Employees:  $\{[\underline{PersNr}, Name]\}$

Professoren:  $\{[\underline{PersNr}, Level, Room]\}$

Assistants:  $\{[\underline{PersNr}, Area]\}$



# Relational Modelling of weak Entities

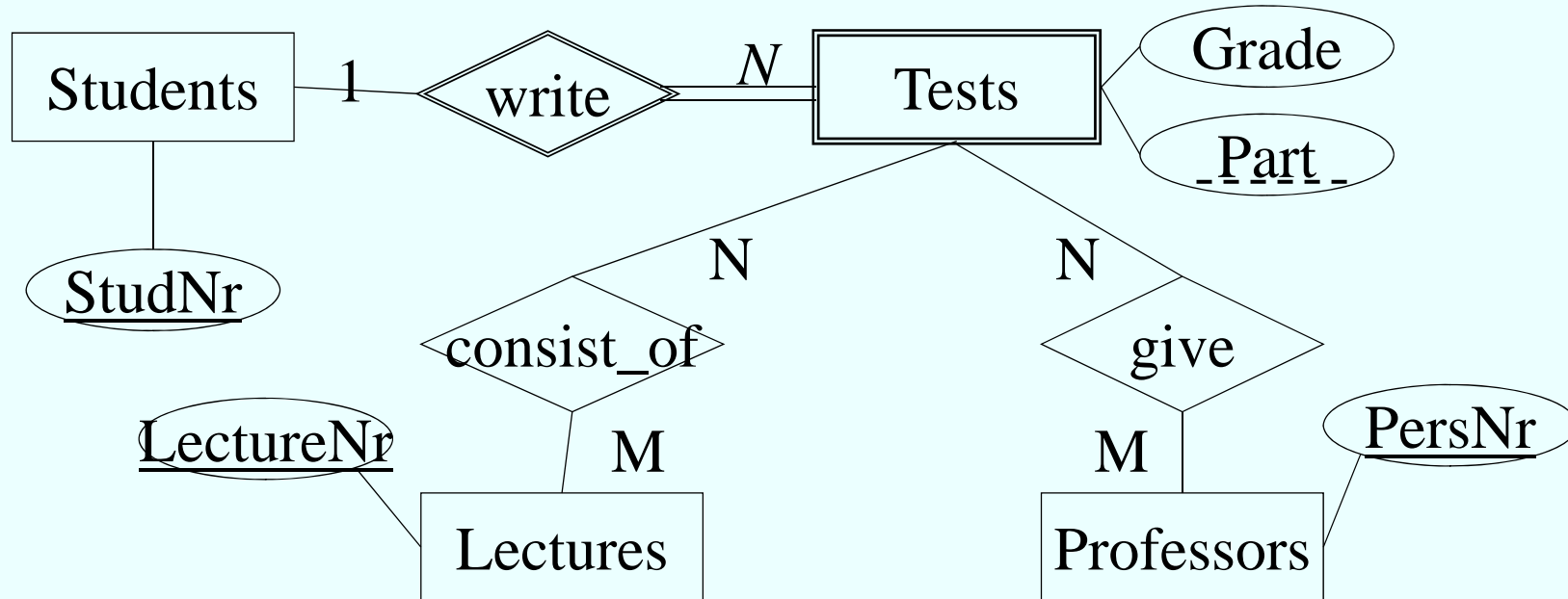


Entity sets Students, Tests:

Students: {[StudNr: integer, ... ]}

Tests: {[StudNr: integer, Part: string, Grade: integer]}

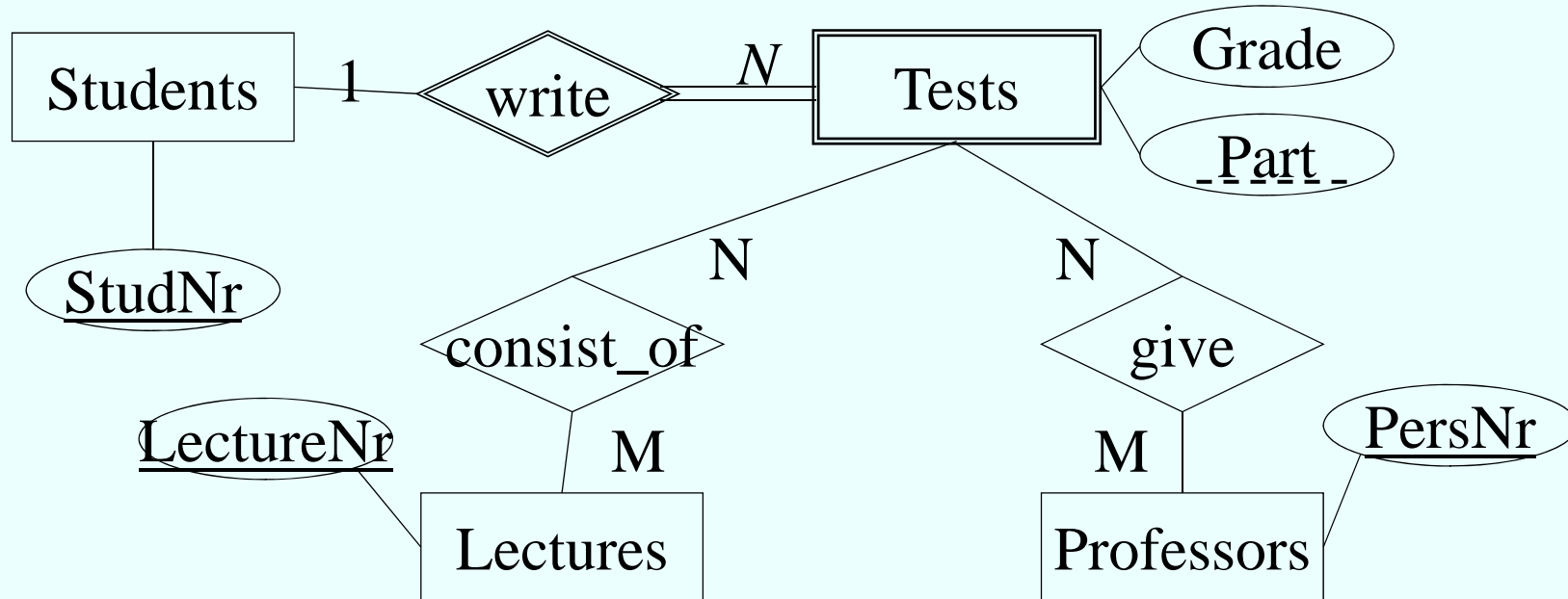
# Relational Modelling of weak Entities



RelShips *consist\_of* and *give*:

Globally unique primary key of the relation *Pests* = *StudNr* **and** *Part*  
must be transferred as foreign key into the relations *consist\_of* and *give*

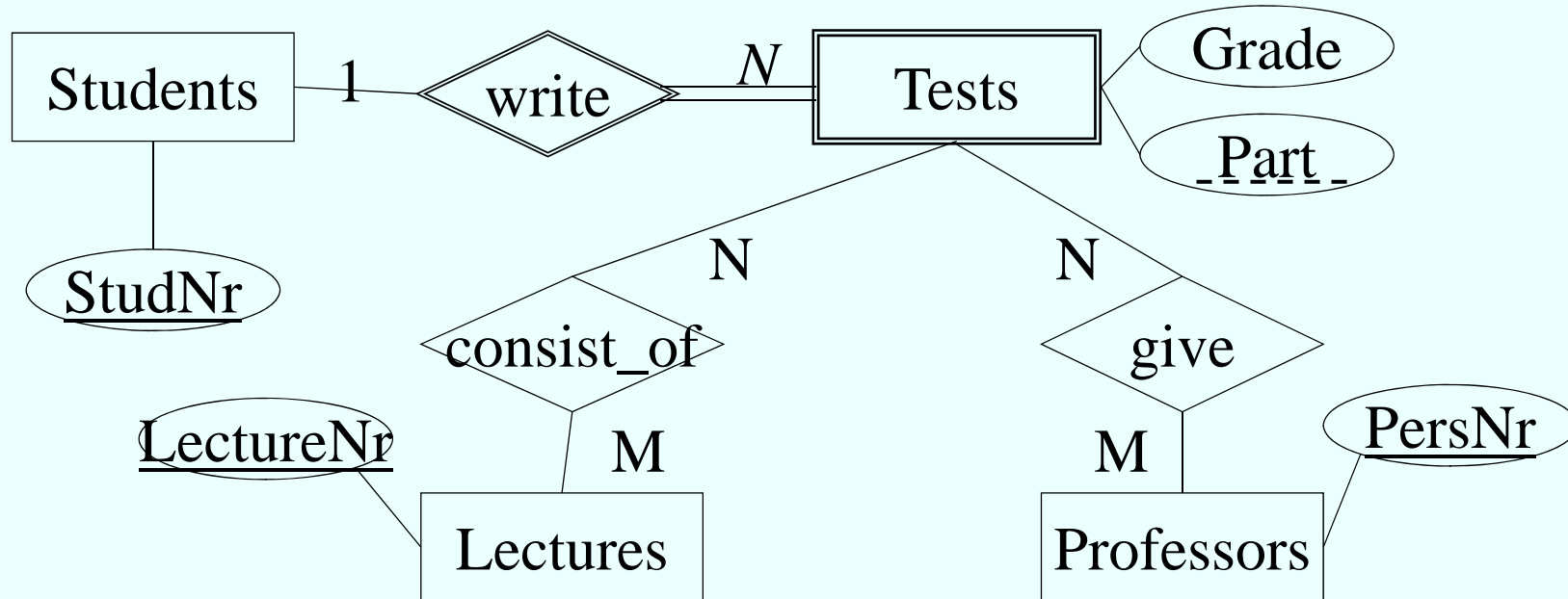
# Relational Modelling of weak Entities



consist\_of: {[StudNr: integer, Part: string, LectureNr: integer]}

give: {[StudNr: integer, Part: string, PersNr: integer]}

# Relational Modelling of weak Entities



RelShip *write* ?

Professors			
PersNr	Name	Level	Room
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

Students		
StudNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10
26830	Aristoxenos	8
27550	Schopenhauer	6
28106	Carnap	3
29120	Theophrastos	2
29555	Feuerbach	2

Lectures			
Lecture Nr	Title	Weekly Hours	Given_by
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137

attend	
StudNr	LectureNr
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
25403	5022
29555	5022
29555	5001

require	
Predecessor	Successor
5001	5041
5001	5043
5001	5049
5041	5216
5043	5052
5041	5052
5052	5259

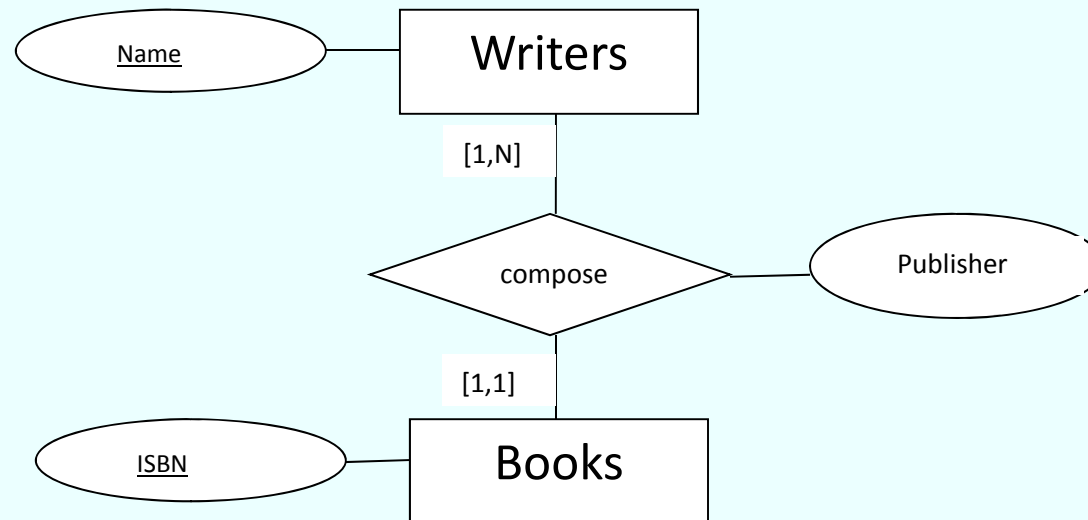
test			
StudNr	LectureNr	PersNr	Grade
28106	5001	2126	1
25403	5041	2125	2
27550	4630	2137	2

Assistants			
PersNr	Name	Area	Boss
3002	Platon	Ideenlehre	2125
3003	Aristoteles	Syllogistik	2125
3004	Wittgenstein	Sprachtheorie	2126
3005	Rhetikus	Planetenbewegung	2127
3006	Newton	Keplersche Gesetze	2127
3007	Spinoza	Gott und Natur	2126

# Sample assignment

## Assignment 1 (E/R-Modelling) 7 Points

Given the following E/R-Model (in the notation of the lecture) with an excerpt of the world of literature.



Can a book be composed by several writers?

yes  no

Can a writer compose several books?

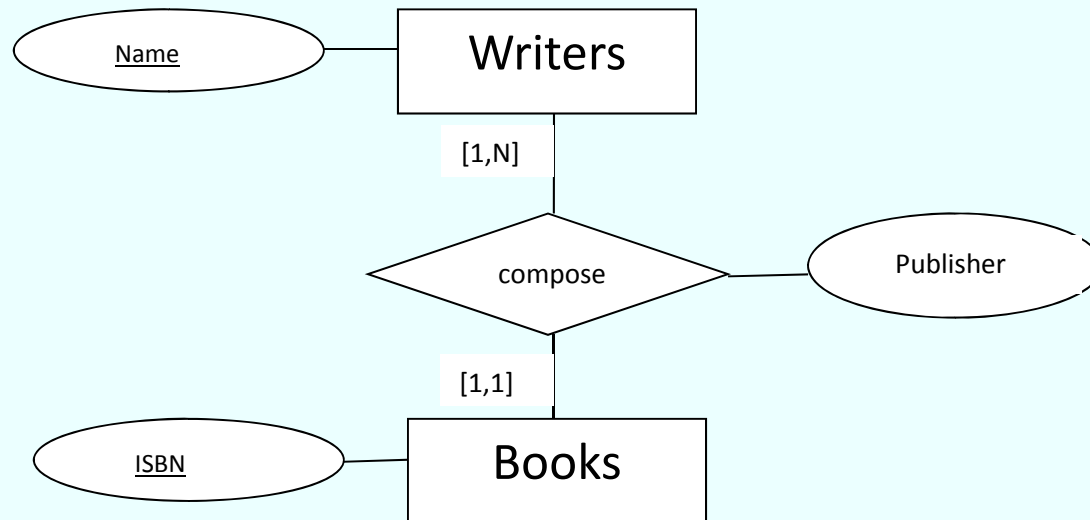
yes  no

Can a writer also compose no books?

yes  no

Give a **short** explanation in each case.

# Sample assignment (cont'd)



Describe the **two** tables which stem from the **refined** transformation of 1:N relationships to relational schemas for the above model.

Use the short notation  $T(\underline{A}, B, ..)$  with T: Table name, A and B attributes, A primary key.