

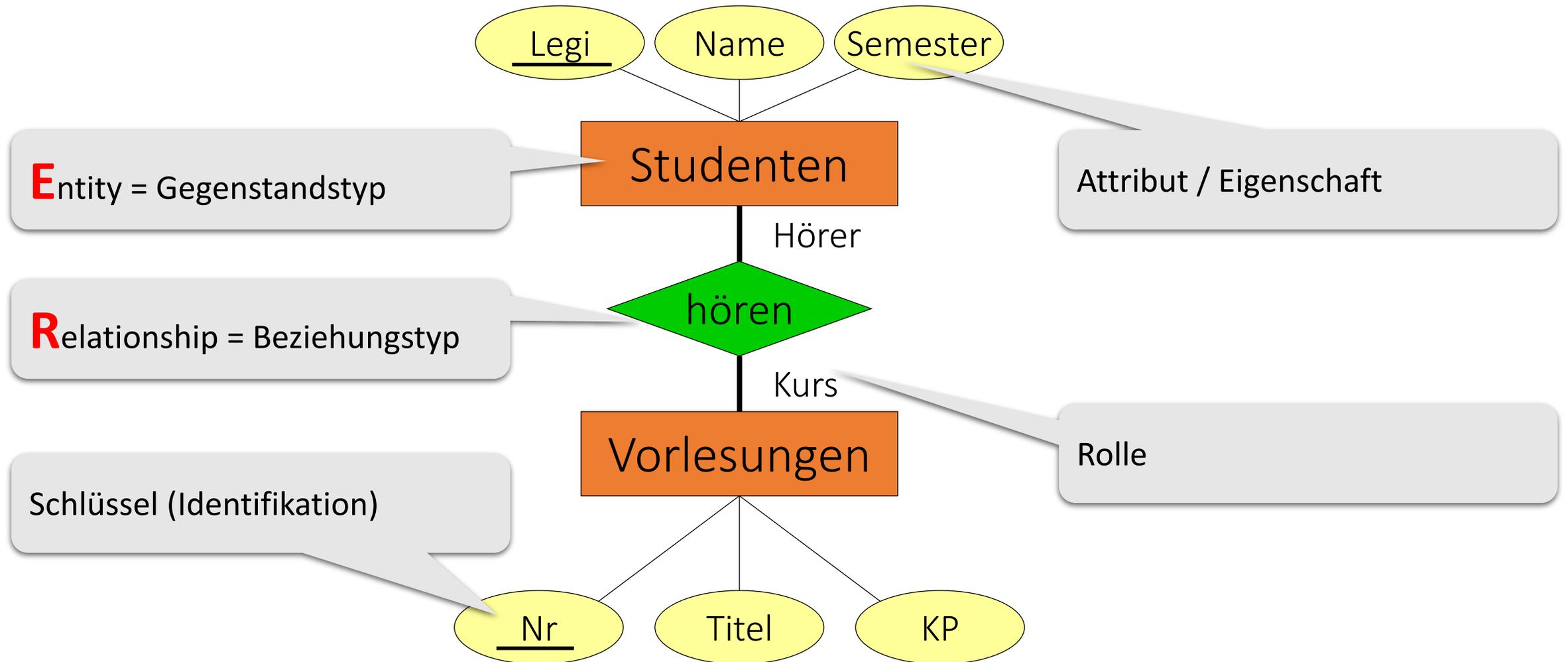
1.6.2016

INFORMATIK II

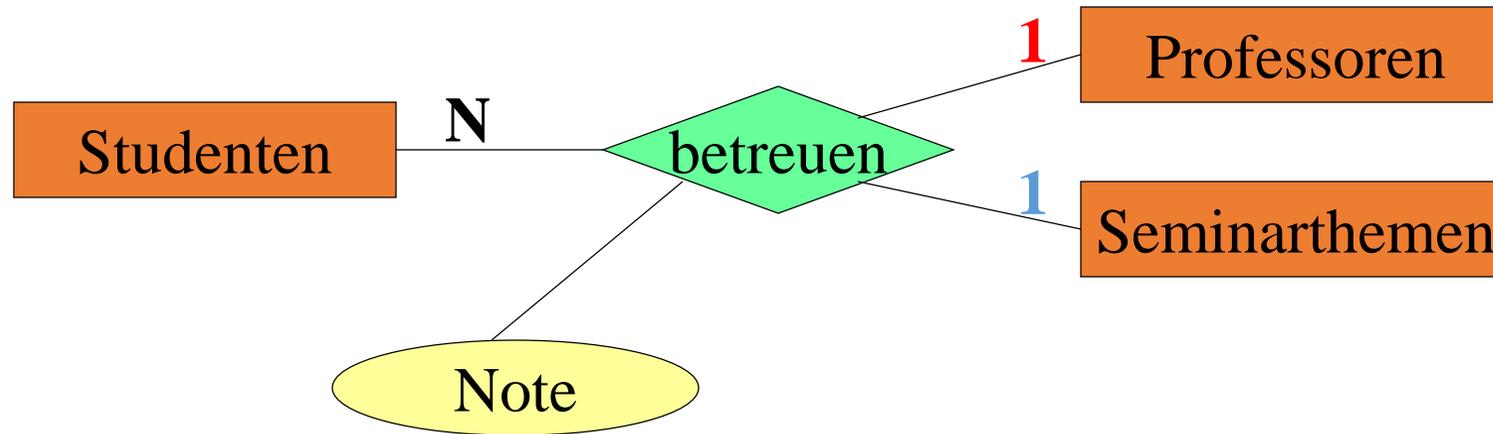
EXERCISE SESSION 11

Entity Relationship Modell

Entity/Relationship (ER) Modell



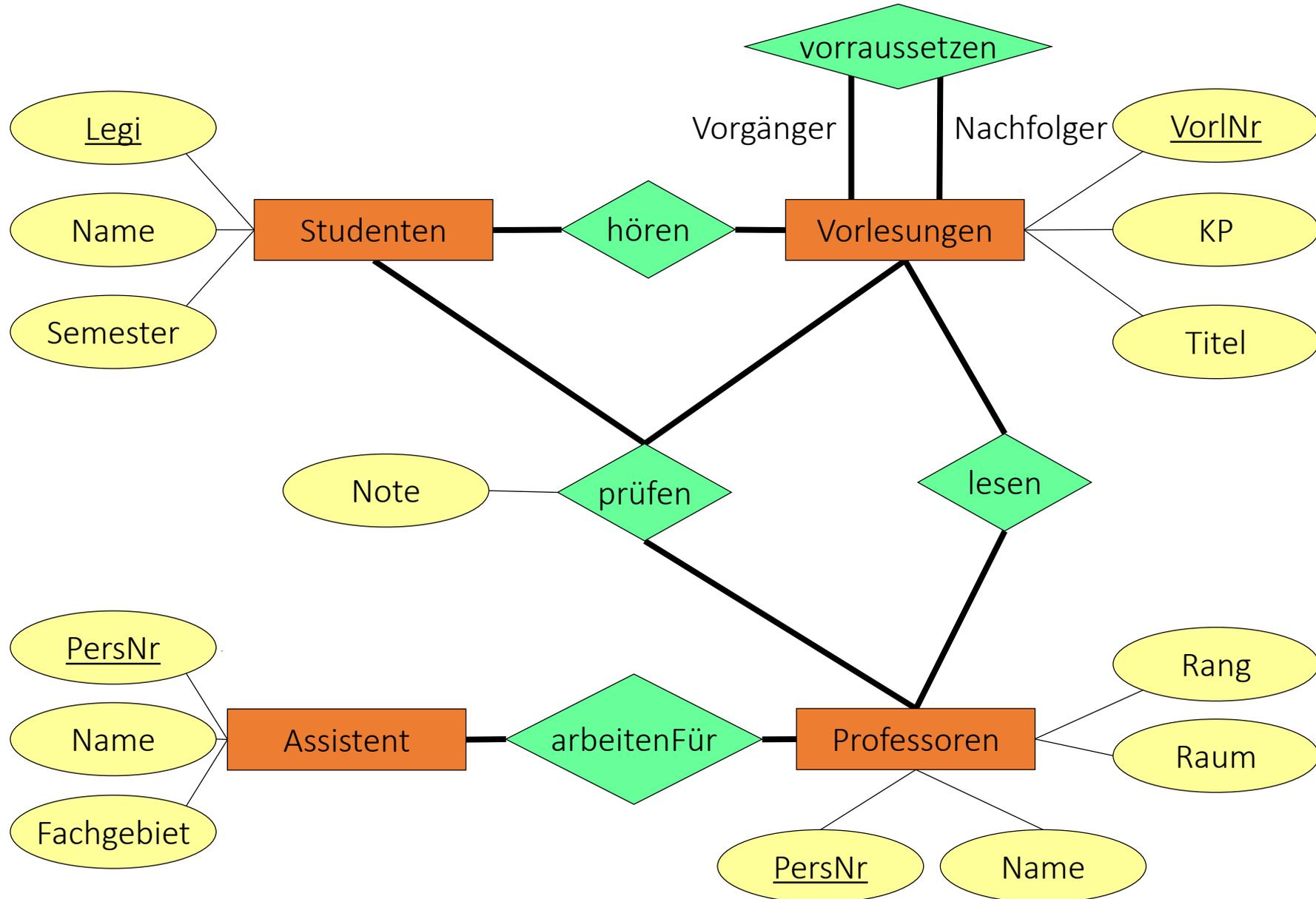
Beispiel: Seminar



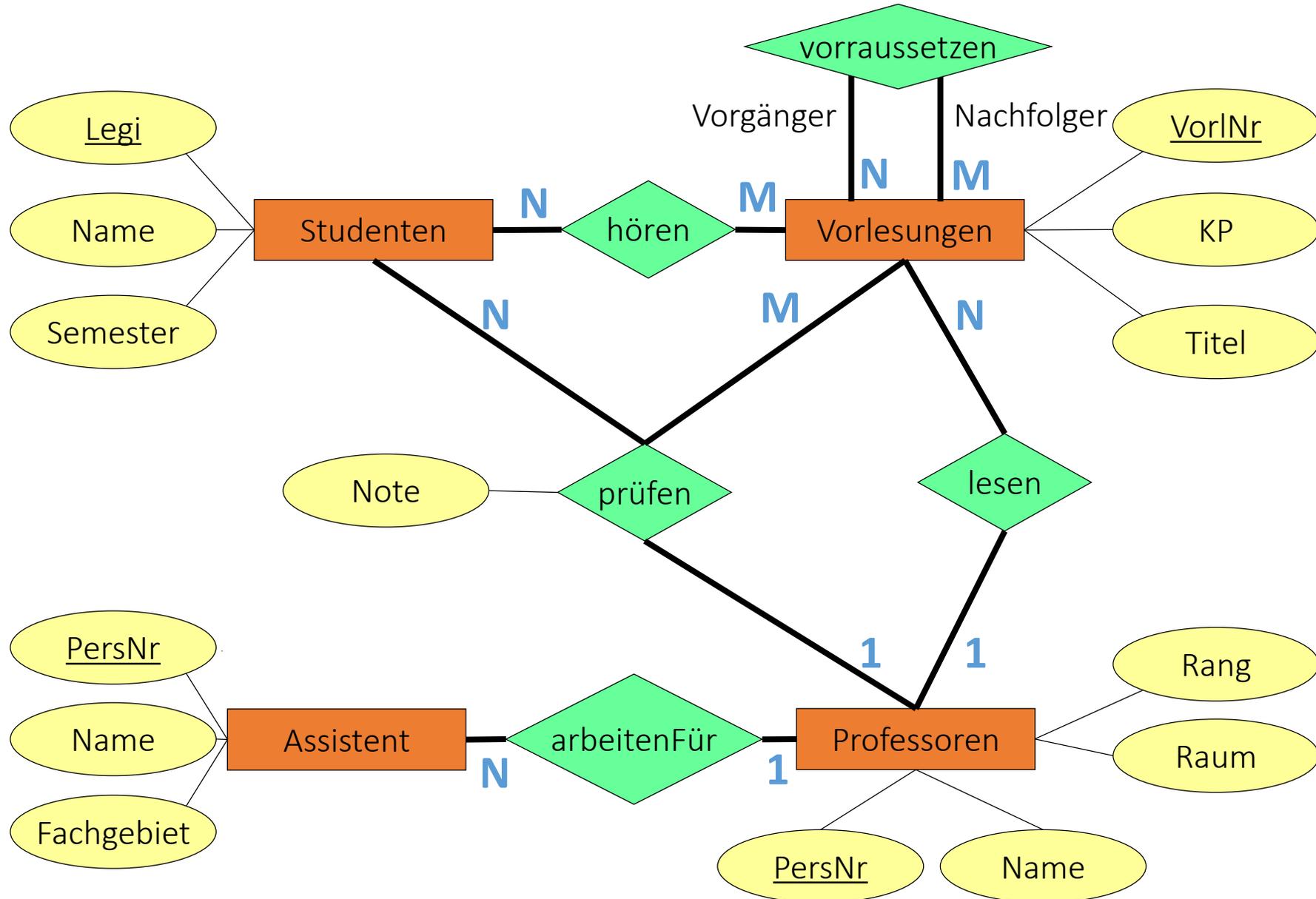
betreuen: Professoren \times Studenten \rightarrow Seminarthemen

betreuen: Seminarthemen \times Studenten \rightarrow Professoren

Modell einer Universität



Modell einer Universität



Relationales Modell

Regel #1: Darstellung von Entities

Studenten:

{[Legi:integer], *Name:string, Semester: integer*}

Vorlesungen:

{[VorlNr:integer], *Titel: string, KP: integer*}

Professoren:

{[PersNr:integer], *Name: string, Rang: string, Raum: integer*}

Assistenten:

{[PersNr:integer], *Name: string, Fachgebiet: string*}

Regel #2: Darstellung von Beziehungen

hören:

{[Legi: integer, VorlNr: integer]}

lesen :

{[PersNr: integer, VorlNr: integer]}

Fremdschlüssel, identifizieren
Tupel aus anderen Entitäten

arbeitenFür :

{[AssiPersNr: integer, ProfPersNr: integer]}

voraussetzen:

{[Vorgänger: integer, Nachfolger: integer]}

prüfen :

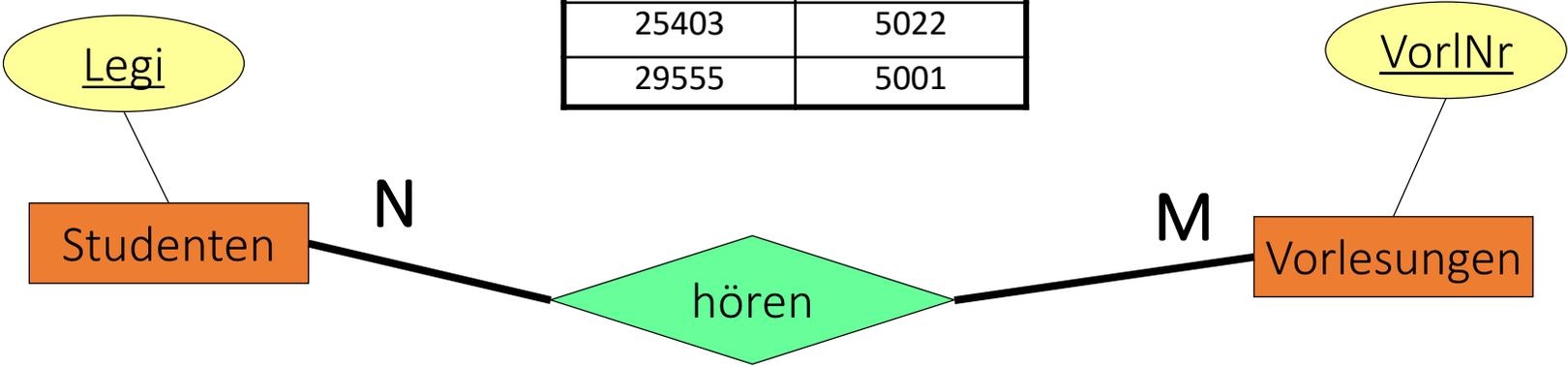
{[Legi: integer, VorlNr: integer, PersNr: integer, Note: decimal]}

Ausprägung von *hören*

Studenten	
<i>Legi</i>	...
26120	...
27550	...
...	...

hören	
<i>Legi</i>	<i>VorlNr</i>
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022
29555	5001

Vorlesungen	
<i>VorlNr</i>	...
5001	...
4052	...
...	...



Regel #3: Zusammenfassung von Relationen

Vorlesungen : {[VorlNr, Title, CP]}

Professoren: {[PersNr, Name, Level, Room]}

lesen: {[VorlNr, PersNr]}

Fasse (nur) Relationen mit **gleichem Schlüssel** zusammen
(also auch nur (N:1), (1:N) oder (1:1) Beziehungen)

Zusammenfassen:

Vorlesungen : {[VorlNr, Title, CP, **gelesenVon**]}

Professoren : {[PersNr, Name, Level, Room]}



Ausprägung von *Professoren* und Vorlesungen

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	FP	226
2126	Russel	FP	232
2127	Kopernikus	AP	310
2133	Popper	AP	52
2134	Augustinus	AP	309
2136	Curie	FP	36
2137	Kant	FP	7

Vorlesungen			
VorlNr	Titel	KP	gelesenVon
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



Das funktioniert NICHT

Professoren				
PersNr	Name	Rang	Raum	liest
2125	Sokrates	FP	226	5041
2125	Sokrates	FP	226	5049
2125	Sokrates	FP	226	4052
...
2134	Augustinus	AP	309	5022
2136				

Vorlesungen		
VorlNr	Titel	KP
5001	Grundzüge	4
5041	Ethik	4
5043	Erkenntnistheorie	3
5049	Mäeutik	2
4052	Logik	4
		3
		2
		2
		2
		4

Problem: Redundanz und Anomalien
PersNr ist kein gültiger Schlüssel für
Professoren mehr



Relationale Algebra & SQL

Die relationale Algebra

σ Selektion

π Projektion

\times kartesisches Produkt

\bowtie Join (Verbund)

ρ Umbenennung

Relational Algebra und SQL

Welcher Professor liest "Mäeutik"?

```
select      Name, Titel
from        Professoren , Vorlesungen
where       PersNr = gelesenVon and Titel = "Mäeutik" ;
```

Projektion

Kreuzprodukt

Selektion

$\Pi_{\text{Name, Titel}}$
 $(\sigma_{\text{PersNr=gelesenVon} \wedge \text{Titel='Mäeutik'}}$
 $(\text{Professoren} \times \text{Vorlesungen}))$

Kartesisches Produkt

SELECT ... FROM *Tabelle1, Tabelle2* ...

entspricht dem kartesischen Produkt

SELECT * FROM studenten, hören

Legi	Name	Semester	Legi	VorNr
24002	Xenokrates	18	26120	5001
25403	Jonas	12	26120	5001
26120	Fichte	10	26120	5001
26830	Aristoxenos	8	26120	5001
27550	Schopenhauer	6	26120	5001
28106	Carnap	3	26120	5001
29120	Theophrastos	2	26120	5001
29555	Feuerbach	2	26120	5001
4711	Unbekannter	NULL	26120	5001
24002	Xenokrates	18	27550	5001
25403	Jonas	12	27550	5001
26120	Fichte	10	27550	5001

studenten × hören

Kartesisches Produkt plus Selektion

```
SELECT * FROM studenten s, hören h  
WHERE s.Legi = h.Legi
```

Legi	Name	Semester	Legi	VorINr
25403	Jonas	12	25403	5022
26120	Fichte	10	26120	5001
27550	Schopenhauer	6	27550	4052
27550	Schopenhauer	6	27550	5001
28106	Carnap	3	28106	5041
28106	Carnap	3	28106	5052
28106	Carnap	3	28106	5216
28106	Carnap	3	28106	5259
29120	Theophrastos	2	29120	5001
29120	Theophrastos	2	29120	5041
29120	Theophrastos	2	29120	5049
29555	Feuerbach	2	29555	5001
29555	Feuerbach	2	29555	5022

$\sigma_{\text{studenten.legi} = \text{hören.Legi}}(\text{studenten} \times \text{hören})$

Kartesisches Produkt plus Selektion = Join

SELECT ... FROM *Tabelle1 JOIN Tabelle2 ON condition*

entspricht dem Join ⋈

```
SELECT *  
FROM studenten s  
JOIN hören h ON s.Legi = h.Legi
```

```
SELECT * FROM studenten s, hören h  
WHERE s.Legi = h.Legi
```

Legi	Name	Semester	VorINr
25403	Jonas	12	5022
26120	Fichte	10	5001
27550	Schopenhauer	6	4052
27550	Schopenhauer	6	5001
28106	Carnap	3	5041
28106	Carnap	3	5052
28106	Carnap	3	5216
28106	Carnap	3	5259
29120	Theophrastos	2	5001
29120	Theophrastos	2	5041
29120	Theophrastos	2	5049
29555	Feuerbach	2	5001
29555	Feuerbach	2	5022

studenten⋈hören

Aggregatfunktion und Gruppierung

Aggregatfunktionen: **avg, max, min, count, sum**

```
select v.gelesenVon, p.Name, sum(v.KP)
from Vorlesungen v, Professoren p
where v.gelesenVon = p.PersNr and p.Rang = 'FP'
group by v.gelesenVon, p.Name
having avg(v.KP) >= 3;
```

SQL weiss nicht, dass sich der Name innerhalb der Gruppe nicht ändern kann, wenn nur gelesenVon angegeben ist

(Fast) Alles

select [distinct]

column1 [**as** *alias1*], *column2* [**as** *alias2*], [avg, max, min, count, sum]...

from *table1* [*alias1*], *table2* [*alias2*],...

[**where** *condition(s)*]

[**group by** *column(s)*]

[**having** *condition(s)*]

[**order by** *column(s)* [**DESC**]];

Projektion π
[+Umbenennung ρ]

Kreuzprodukt \times
[+Umbenennung ρ]

Selektion σ

Abarbeitung der Anfrage in SQL

1. Schritt: *Kreuzprodukt und Selektion*

from Vorlesungen, Professoren
where gelesenVon = PersNr **and** Rang = 'FP'

2. Schritt: *Gruppierung*

group by gelesenVon, Name

3. Schritt: *Selektion der Gruppierung*

having avg (KP) \geq 3

4. Schritt: *Projektion*

select gelesenVon, Name, **sum** (KP)

Geschachtelte Anfrage

Unteranfrage in der **where**-Klausel

Welche Prüfungen sind besser als durchschnittlich verlaufen?

```
select *  
from prüfen  
where Note > ( select avg (Note)  
                from prüfen );
```

Zur Prüfung

Informatik II: 1h nach Informatik I Prüfung
am 14.8.2016 von 11:00 – 13:00.

Erlaubt: 4 A4 Seiten (rsp. 2 Blätter) handgeschrieben oder min. 11 Punkt Schriftart. Kopieren ist erlaubt aber nicht schlau.
Handgeschriebenes vom Tablet drucken ist auch erlaubt, wenn dabei die Schrift nicht wesentlich verkleinert wird (in Relation zur sonst üblichen Schreibschrift).

Was ist prüfungsrelevant?

Prüfungsrelevant: Inhalt aller Vorlesungen und aller Übungen dieses Kurses.

In 2017 neu prüfungsrelevant, aber nicht in 2016:

- Komplexität
- Quickselect und Sortieren durch Auswahl, Einfügen und Quicksort

Nicht prüfungsrelevant: Material von 2016, welches 2017 nicht gelehrt wurde:

- OOP (Vererbung und Polymorphie)
- Point-In-Polygon Algorithmus und Bresenham

Prüfung 2017 Vorbereitung

Lösen alter Prüfungsaufgaben und der Übungen

NIEMALS (!) vorher die Lösung ansehen. Dann ist der Lerneffekt = 0!
Aufgabe komplett schriftlich ausarbeiten und danach (!) die Lösung ansehen.*

Lernen in Gruppen mit gegenseitigem Erklären kann sehr hilfreich sein.

Es wird natürlich nicht erwartet, dass Sie sämtlichen Quelltext auswendig hinschreiben können. Es wird aber erwartet, dass Sie Quelltext lesen, verstehen und ergänzen können.

*Einzige Ausnahme: wenn man nicht vorwärts kommt, in die Lösung schauen und dann nochmal komplett selbstständig ausarbeiten.