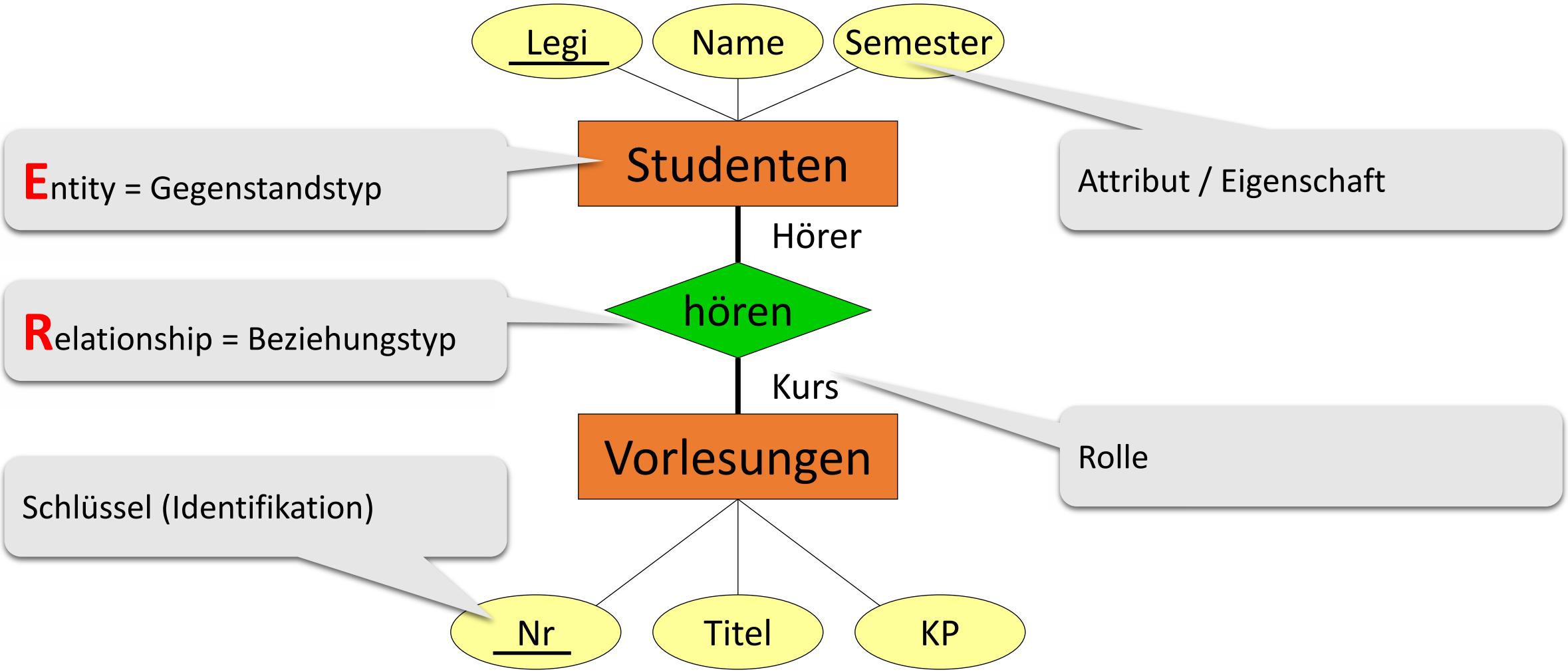
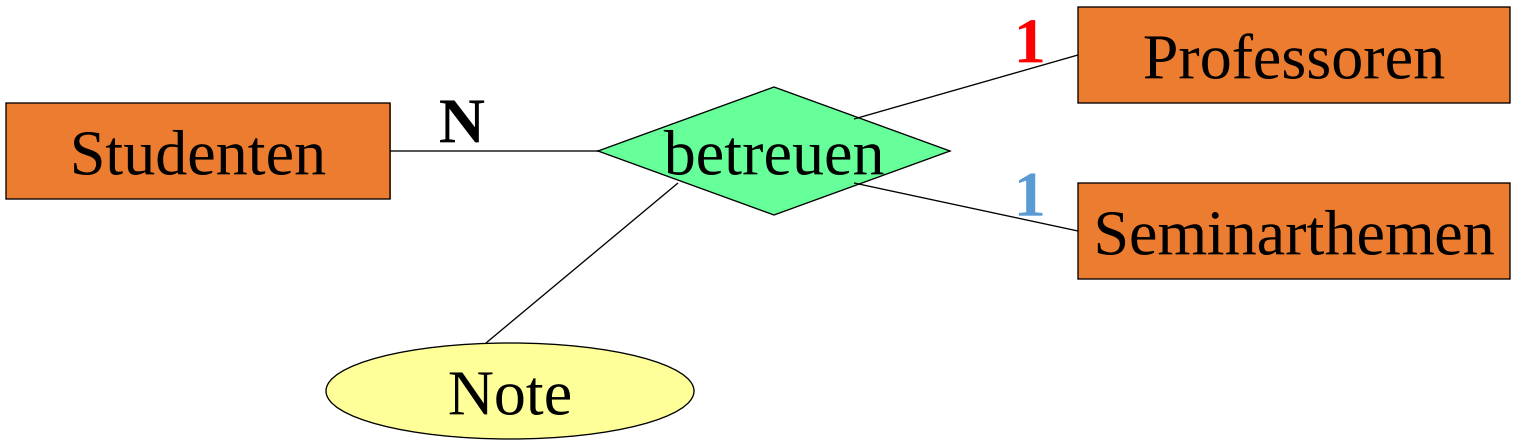
# Entity Relationship Modell

# Entity/Relationship (ER) Modell



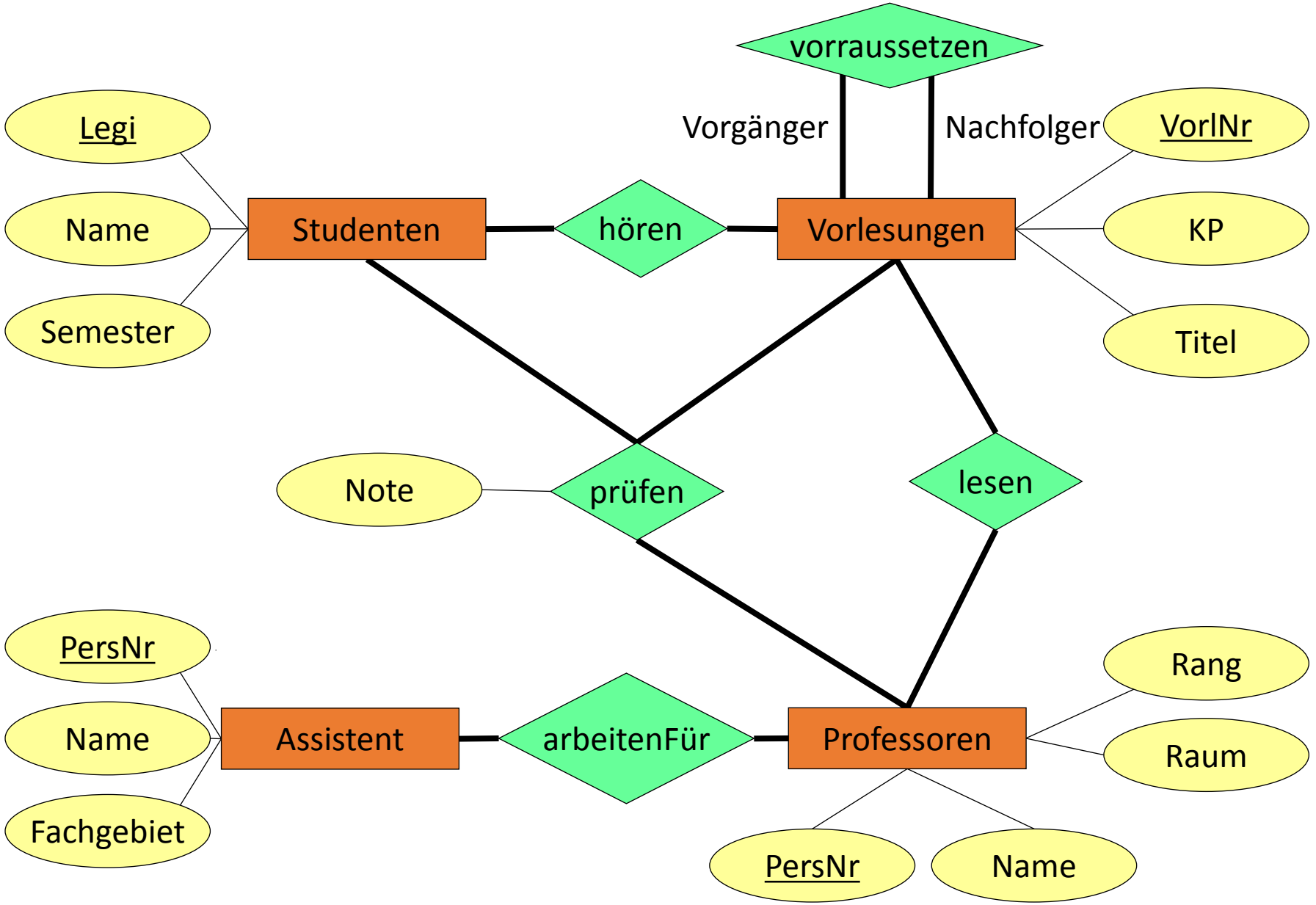
# Beispiel: Seminar



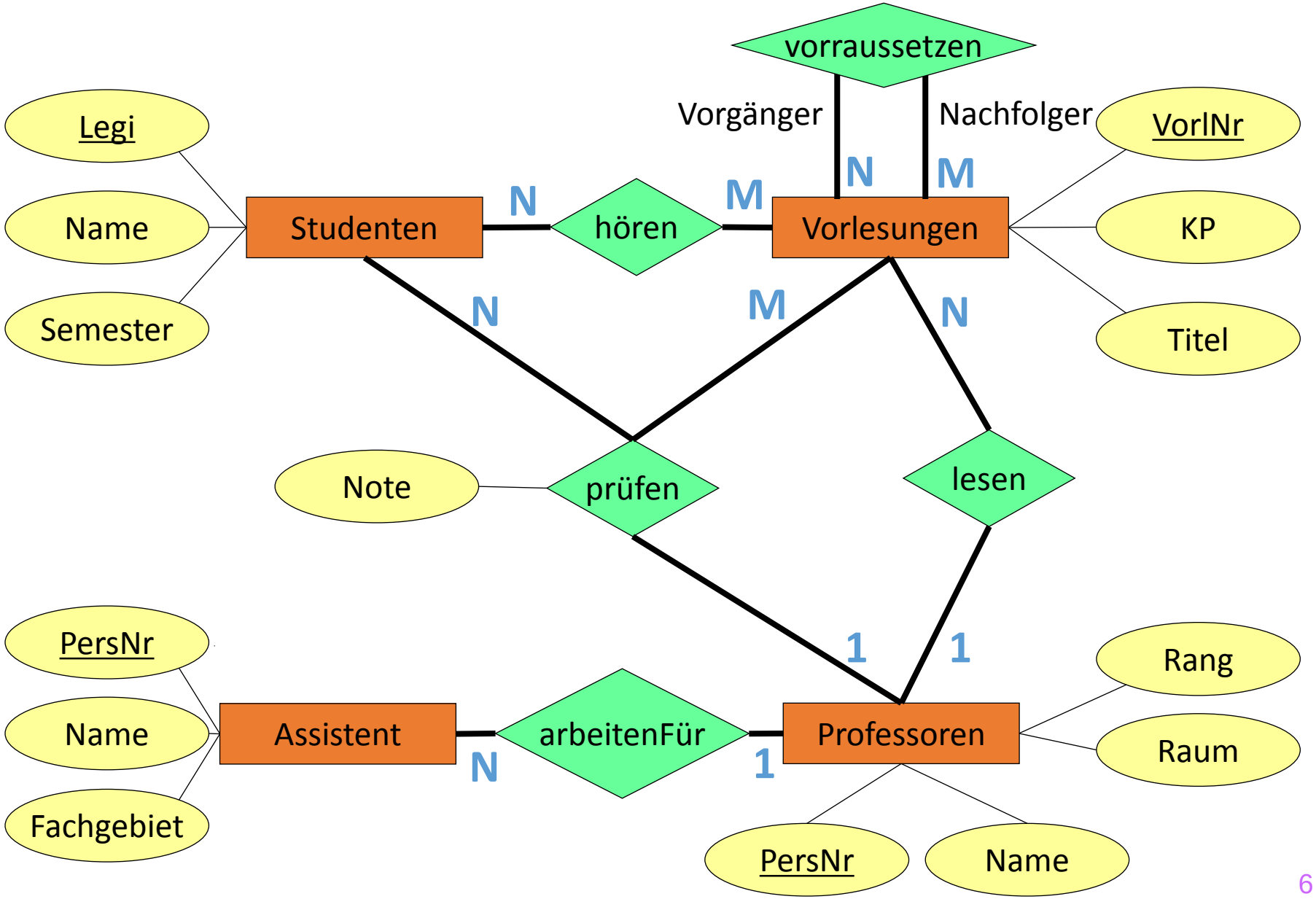

betreuen: Professoren x Studenten → Seminarthemen

betreuen: Seminarthemen x Studenten → Professoren

# Modell einer Universität

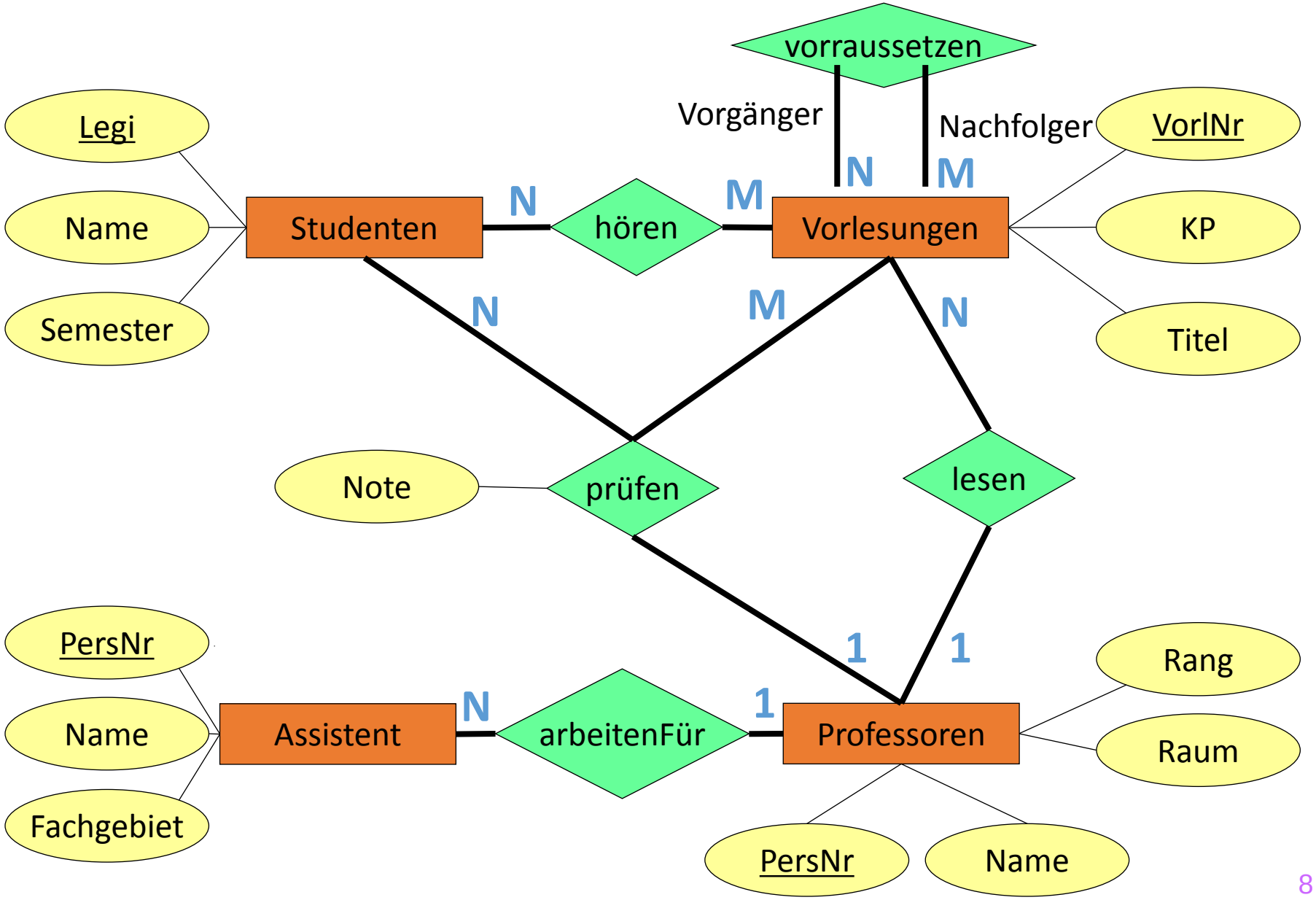


# Modell einer Universität



# Relationales Modell

# Uni Schema


vorraussetzen
Vorgänger
Nachfolger
N
M
Legi
Name
Semester
Studenten
N
hören
M
Vorlesungen
VorlNr
KP
Titel
N
M
N
Note
prüfen
lesen
1
1
PersNr
Name
Fachgebiet
Assistent
N
arbeitenFür
1
Professoren
Rang
Raum
PersNr
Name

# 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*]}

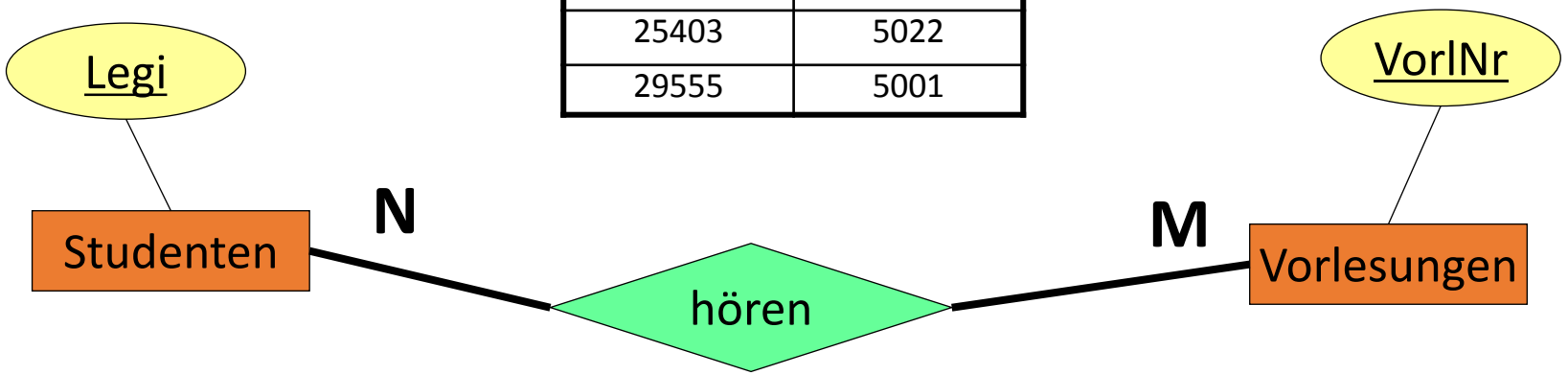
**Assistenten**:
{[PersNr:integer, *Name*: *string, Fachgebiet*: *string*]}

# Ausprägung von *hören*

| Studenten | |
|---|---|
| *Legi* | ... |
| 26120 | ... |
| 27550 | ... |
| ... | ... |

| hören | |
|---|---|
| Legi | VorlNr |
| 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 | |
|---|---|
| *VorlNr* | ... |
| 5001 | ... |
| 4052 | ... |
| ... | ... |



# Darstellung von Beziehungen

**hören**:

{[Legi: integer, VorlNr: integer]}

**lesen** :

{[PersNr: integer, VorlNr: integer]}

**arbeitenFür** :

{[AssiPersNr: integer, ProfPersNr: integer]}

**voraussetzen**:

{[Vorgänger: integer, Nachfolger: integer]}

**prüfen** :

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

# Relationales Modell der Uni-DB

| 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 |

| Studenten | | |
|---|---|---|
| Legi | 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 |

| 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 |
| | Die 3 Kritiken | 4 | 2137 |

| hören | |
|---|---|
| Legi | VorlNr |
| 26120 | 5001 |
| 27550 | 5001 |
| 27550 | 4052 |
| 28106 | 5041 |
| 28106 | 5052 |
| 28106 | 5216 |
| 28106 | 5259 |
| 29120 | 5001 |
| 29120 | 5041 |
| 29120 | 5049 |
| 29555 | 5022 |

| Assistenten | | | |
|---|---|---|---|
| PerslNr | Name | Fachgebiet | 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 |

| voraussetzen | |
|---|---|
| Vorgänger | Nachfolger |
| 5001 | 5041 |
| 5001 | 5043 |
| 5001 | 5049 |
| 5041 | 5216 |
| 5043 | 5052 |
| 5041 | 5052 |
| 5052 | 5259 |

| prüfen | | | |
|---|---|---|---|
| Legi | Nr | PersNr | Note |
| 28106 | 5001 | 2126 | 1 |
| 25403 | 5041 | 2125 | 2 |
| 27550 | 4630 | 2137 | 2 |

# Relationale Algebra & SQL

# Die relationale Algebra

- $\sigma$ Selektion
- $\pi$ Projektion
- $\times$ kartesisches Produkt
- ⋈ Join (Verbund)
- $\rho$ Umbenennung

# Relational Algebra und SQL

## Welcher Professor liest "Mäeutik"?

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

Projektion

Kreuzprodukt

Selektion

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

# Kartesisches Produkt

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

entspricht dem kartesischen Produkt

```
SELECT * FROM studenten, hören
```

| Legi | Name | Semester | Legi | VorlNr |
|---|---|---|---|---|
| 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 | VorlNr |
|---|---|---|---|---|
| 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 | VorlNr |
|---|---|---|---|
| 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

# Mengenoperationen

## Mengenoperation: **union**

( **select** PersNr, Name **from** Assistenten )
**union**
( **select** PersNr, Name **from** Professoren);

union funktioniert nur auf Relationen mit gleichem Schema!

# 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

# 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) >= 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 );
```