

Informatik II

Vorlesung am D-BAUG der ETH Zürich

Vorlesung 11, 23.5.2016

Datenbanksysteme: Das Entity Relationship (ER) Modell,
das Relationale Modell und SQL.

Literatur, Quellen

Literatur: Kemper, Eickler: Datenbanksysteme: Eine Einführung. Oldenbourg Verlag, 9. Auflage, 2013.

Quellen: Basismaterial wurde von Prof. Donald Kossmann & Martin Kaufmann freundlicherweise zur Verfügung gestellt.

Weitere Quelle: Folien zu *Datenbanksysteme: Eine Einführung*, Lehrstuhl III Datenbanksysteme, Prof. Kemper, TU München

Ziele

Nutzen von Datenbanksystemen verstehen, Modellierungskennntnisse

- ER Modell (Modellierung der Weltsicht)
- Relationales Modell (Modellierung für die DB)

Datenbanksystem anwenden

- SQL

Datenbankverwaltungssysteme

Ein Datenbankverwaltungssystem (DBMS) ist ein Werkzeug zur Erstellung und Ausführung datenintensiver Anwendungen

grosse Datenmengen

grosse Datenströme



Typische Anwendungen

Bank

z.B. Konten / „Geldtransfer“

Bibliothek

z.B. Bücher / „Ausleihen“

Facebook, Twitter, ...

z.B. Freunde, „Sende Tweet“

Geoinformationssysteme

z.B. Topographische Information, "Erzeuge Karte"

Wozu Datenbanksysteme?

Vermeide Redundanz und Inkonsistenz

Deklarativer Zugriff auf die Daten und Unabhängigkeit von der Implementation (physische Datenunabhängigkeit)

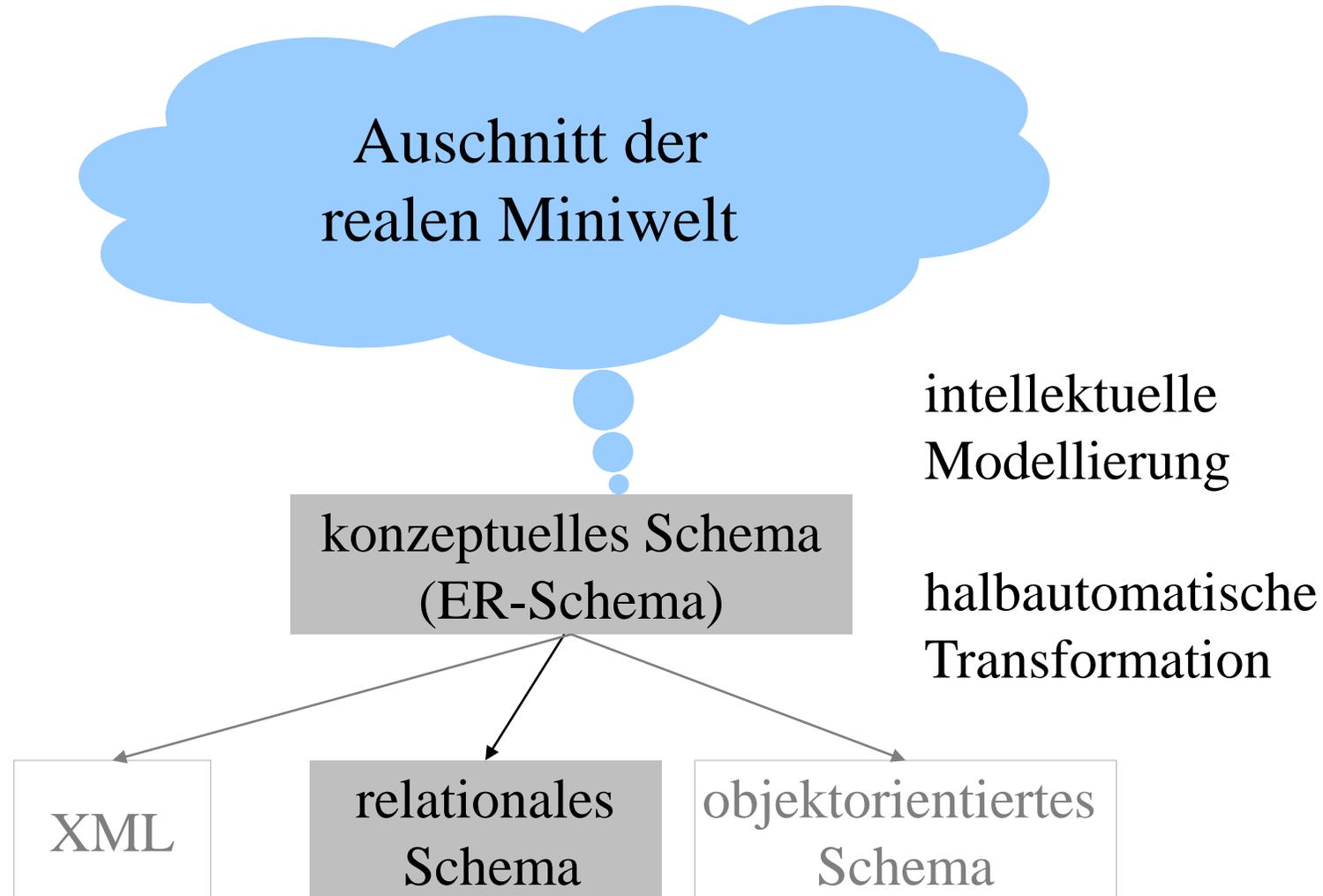
Synchronisiere gleichzeitigen Datenzugriff

Sicherheit, Vertraulichkeit

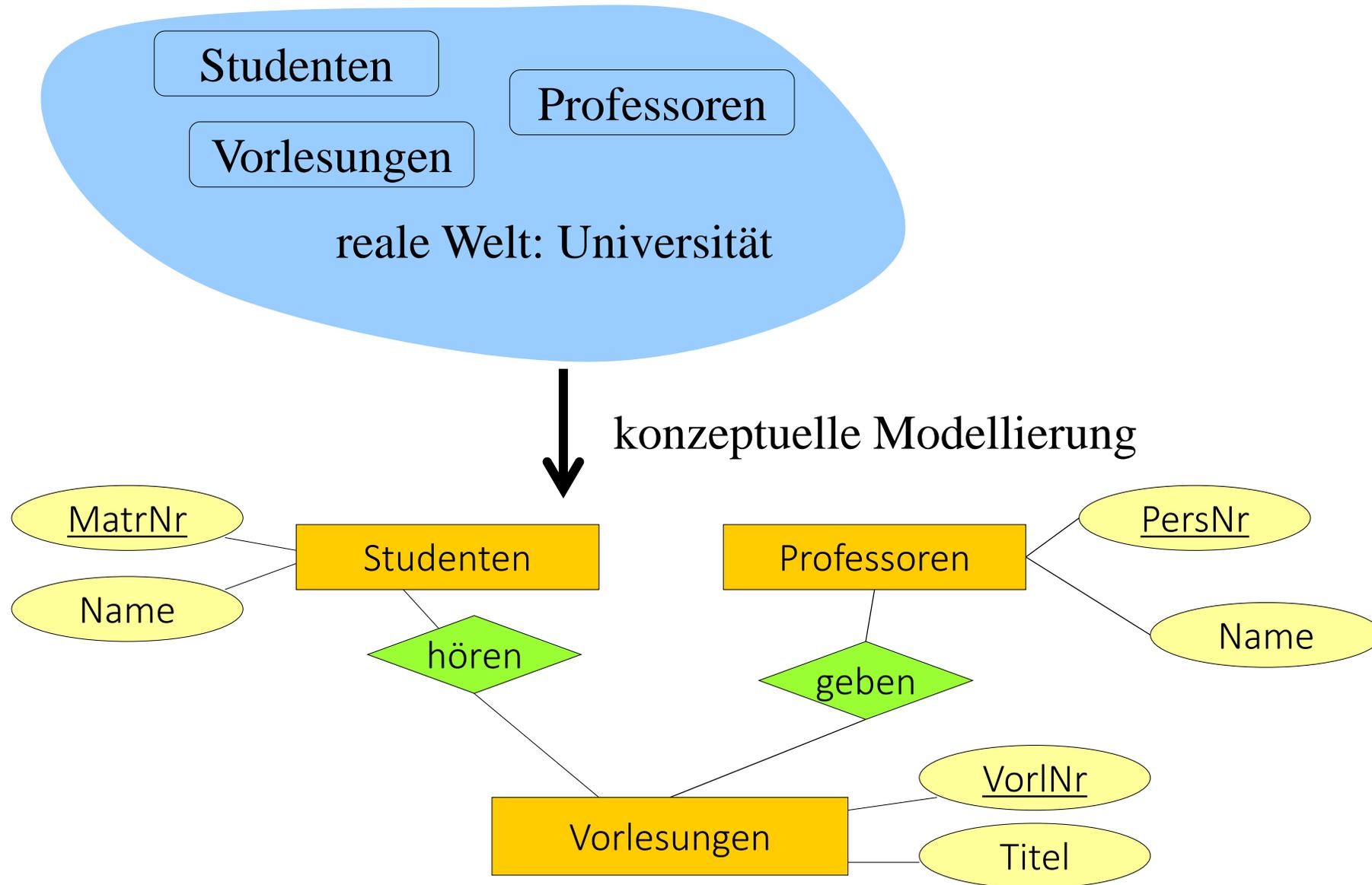
Minimiere Kosten und Aufwand

Ähnliche Funktionalität selbst zu implementieren würde Jahre in Anspruch nehmen

Datenmodellierung



1. Teil heutige Vorlesung: Modellierung (ER-Modell)



2. Teil heutige Vorlesung: Relationales Datenmodell

Studenten	
Legi	Name
26120	Fichte
25403	Jonas
...	...

hören	
Legi	VorlNr
25403	5022
26120	5001
...	...

Vorlesungen	
VorlNr	Titel
5001	Grundzüge
5022	Glaube und Wissen
...	...

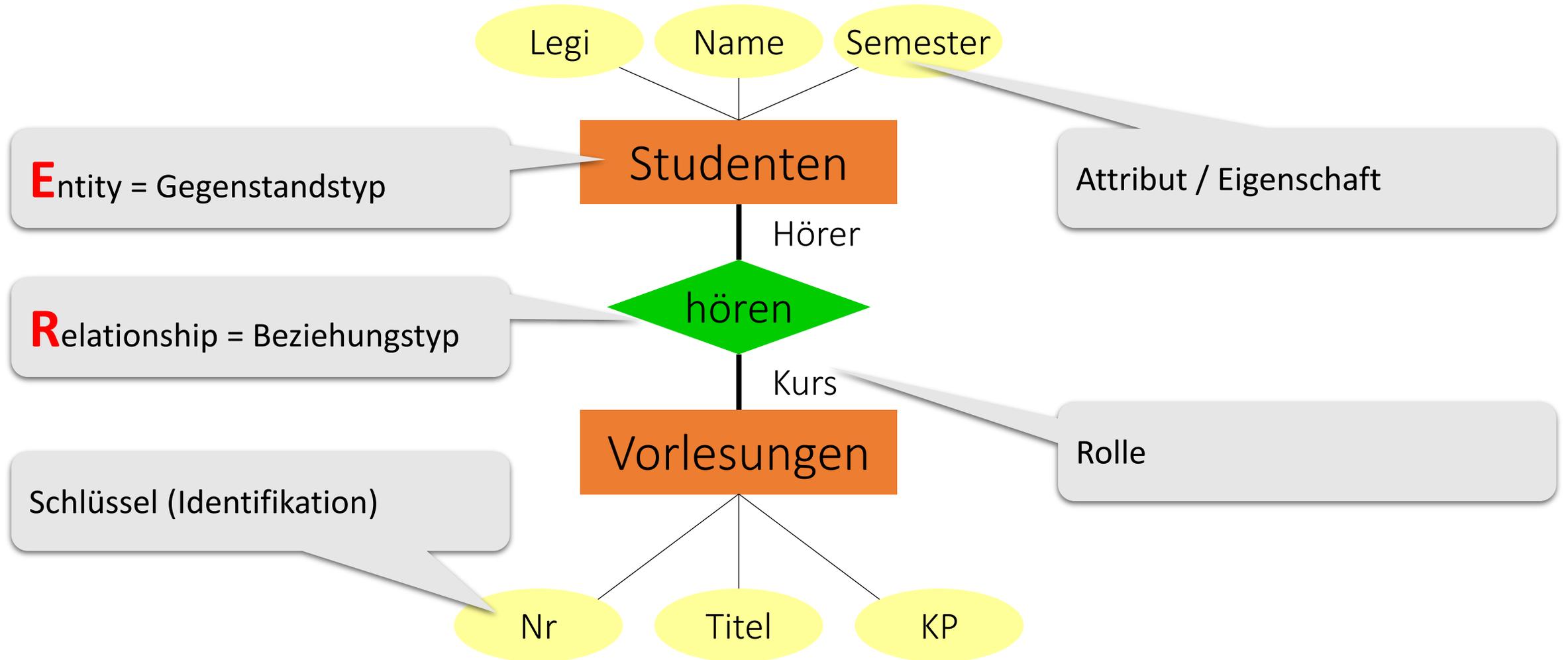
```
select Name
from Studenten, hören, Vorlesungen
where Studenten.Legi= hören.Legi and
        hören.VorlNr= Vorlesungen.VorlNr and
        Vorlesungen.Titel = `Grundzüge`;
```

$\pi_{name} (\text{studenten} \bowtie \text{hören} \bowtie \text{vorlesungen})$

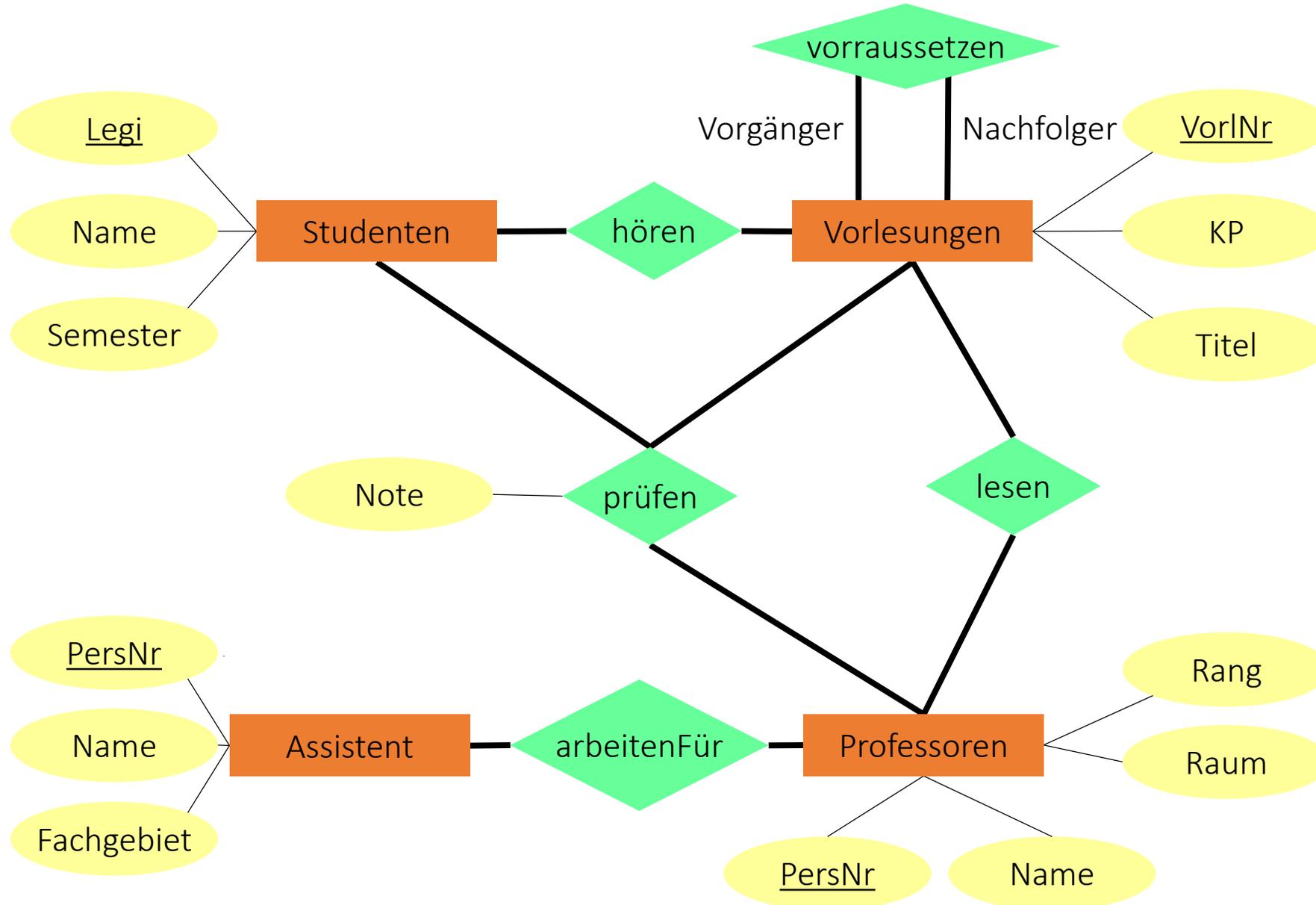
```
update Vorlesungen
set     Title = `Grundzüge der Logik`
where   VorlNr = 5001;
```

Entity Relationship Modell

Entity/Relationship (ER) Modell



Modell einer Universität



... in natürlicher Sprache

Studenten haben LegiNr, Name und Semester. Die LegiNr identifiziert einen Studenten eindeutig.

Vorlesungen haben eine VorlNr, Kreditpunkte und einen Titel. VorlNr identifiziert eine Vorlesung eindeutig.

Professoren haben PersNr, Name, Rang und Raum. PersNr identifiziert einen Professor eindeutig.

Assistenten haben PersNr, Name und Fachgebiet. PersNr. identifiziert einen Assistenten eindeutig.

Studenten hören Vorlesungen

Vorlesungen können Voraussetzung für andere Vorlesungen sein.

Professoren lesen Vorlesungen.

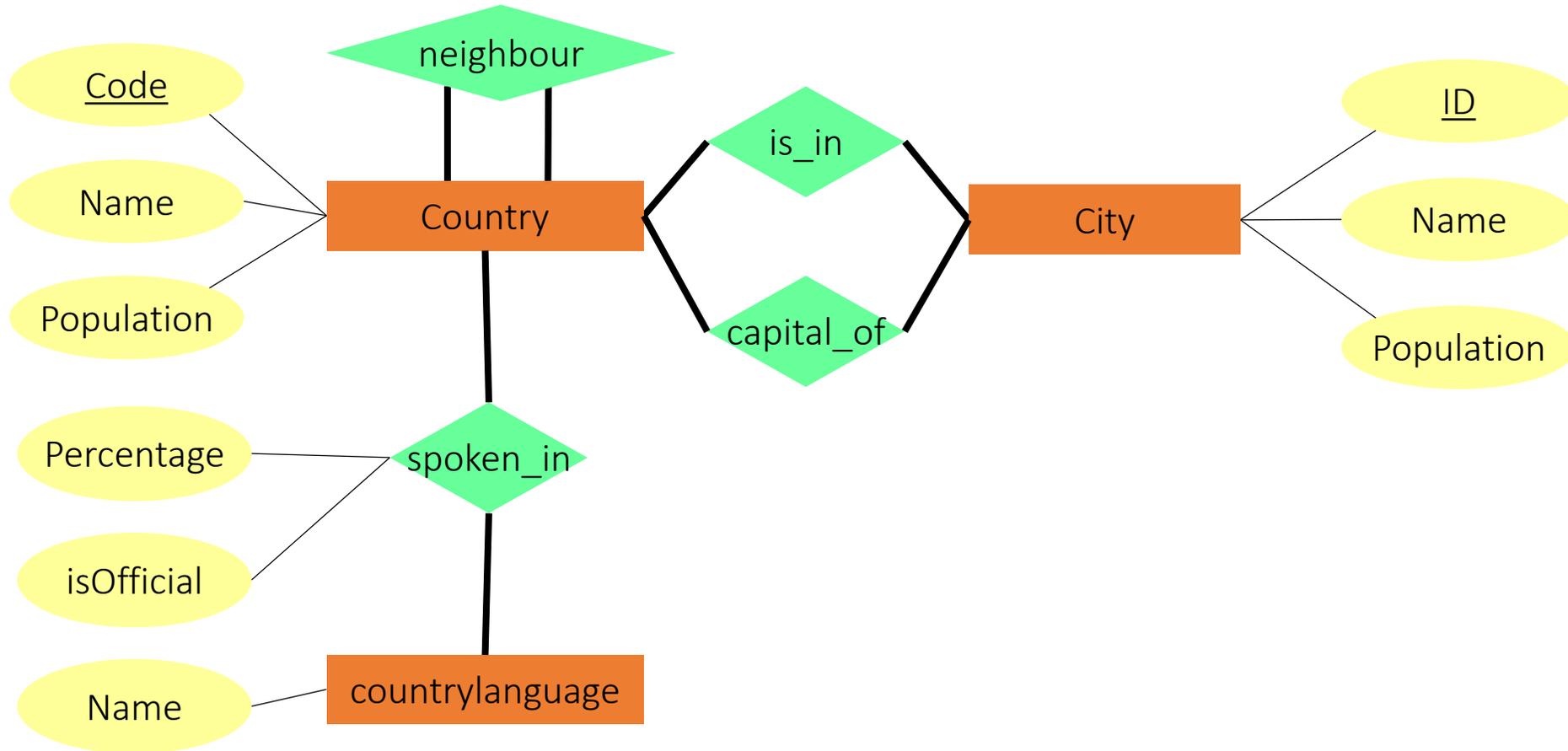
Assistenten arbeiten für Professoren

Studenten werden von Professoren über Vorlesungen geprüft. Studenten erhalten Noten als Teil dieser Prüfungen.

Ist das die einzig mögliche Interpretation?

Nein: zu einem ER-Model gehört immer noch Dokumentation und/oder gesunder Menschenverstand

Ein Modell der Welt



Warum ER?

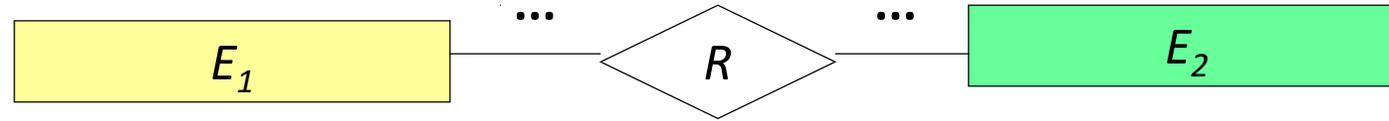
Vorteile

- ER Diagramme sind einfach zu erstellen und editieren
- ER Diagramme sind aufgrund der grafischen Darstellung einfach zu verstehen (vom Laien)
- ER Diagramme beschreiben alle Informationsanforderungen

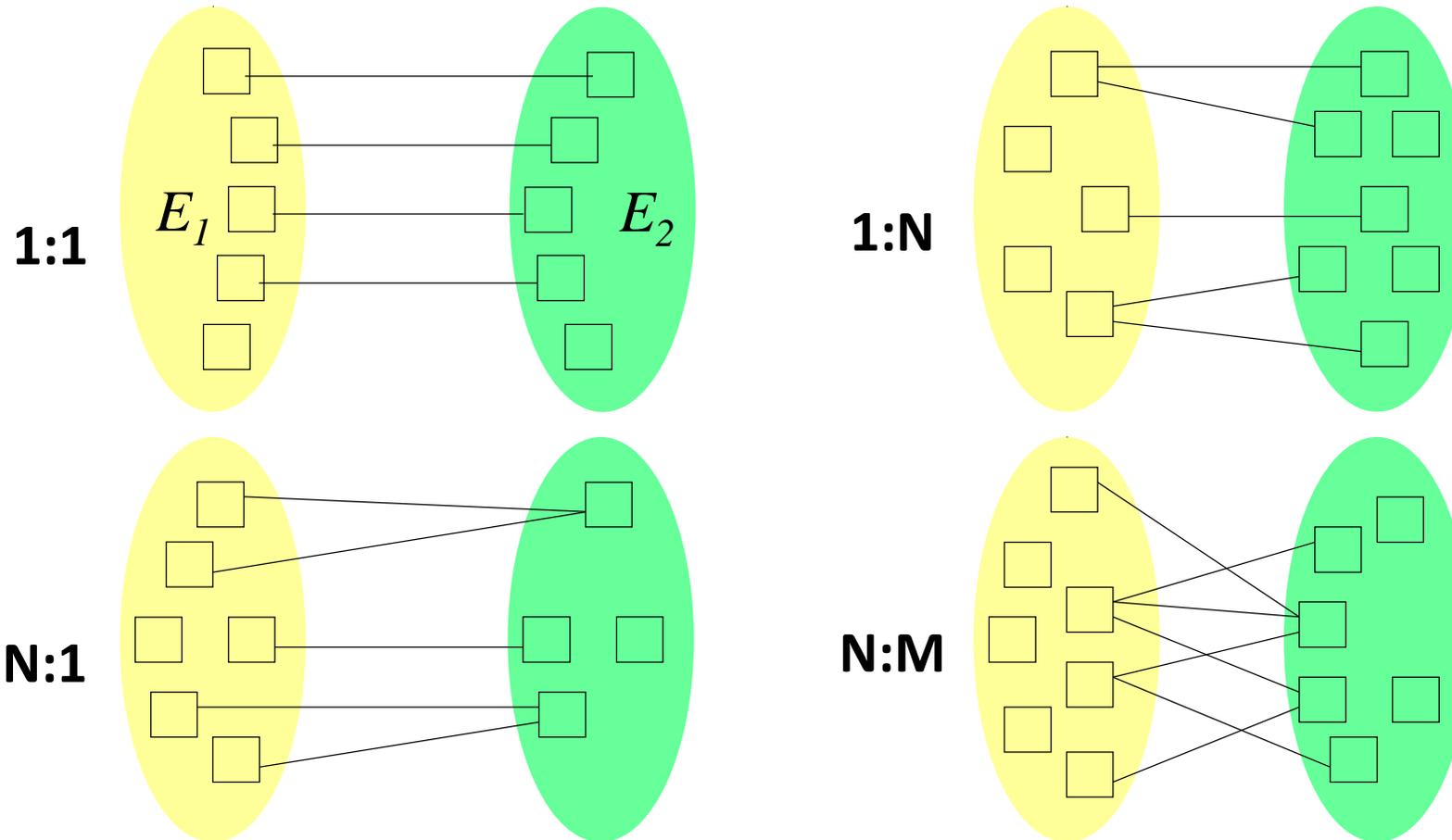
Allgemeines

- Viele Tools verfügbar
- Kontroverse, ob ER/UML in der Praxis von Nutzen ist
- Keine Kontroverse, dass jeder ER/UML lernen sollte

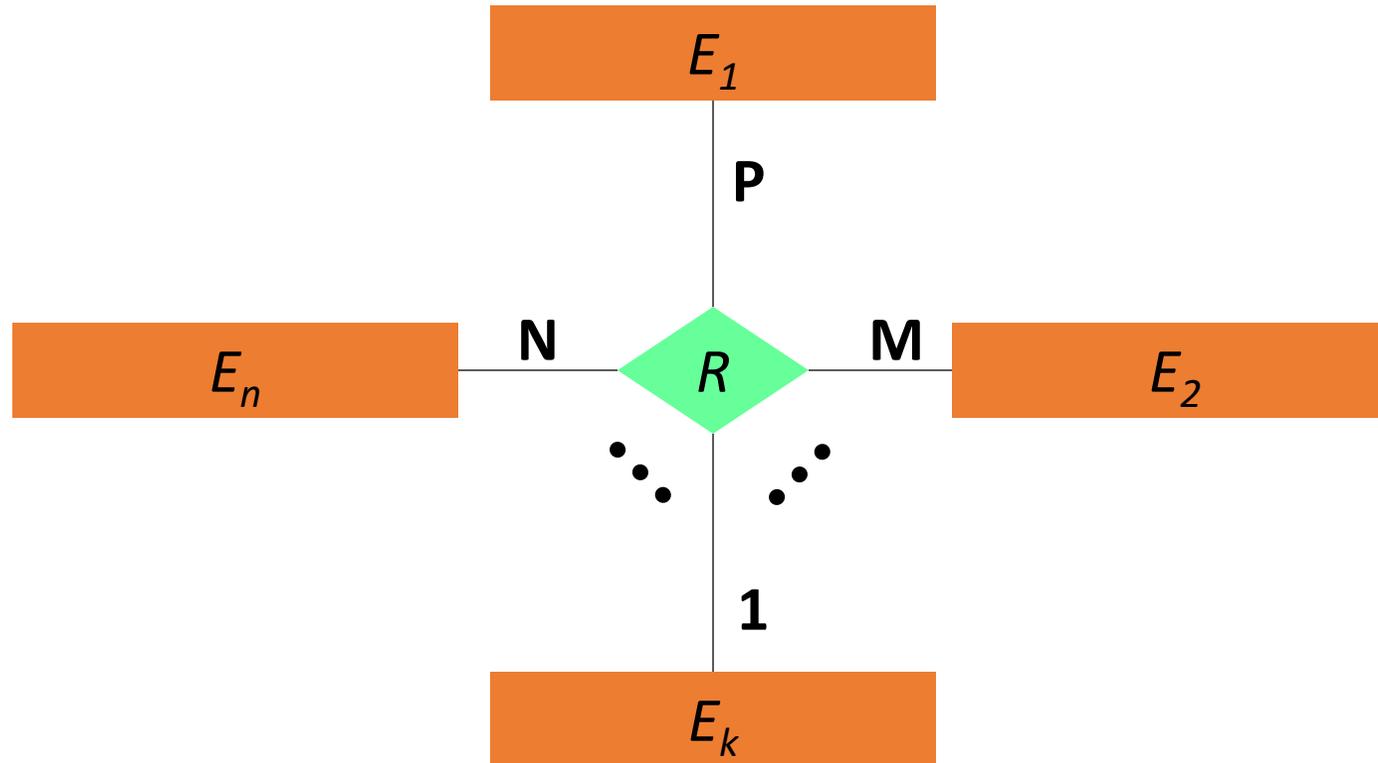
Funktionalitätsangaben



$$R \subseteq E_1 \times E_2$$



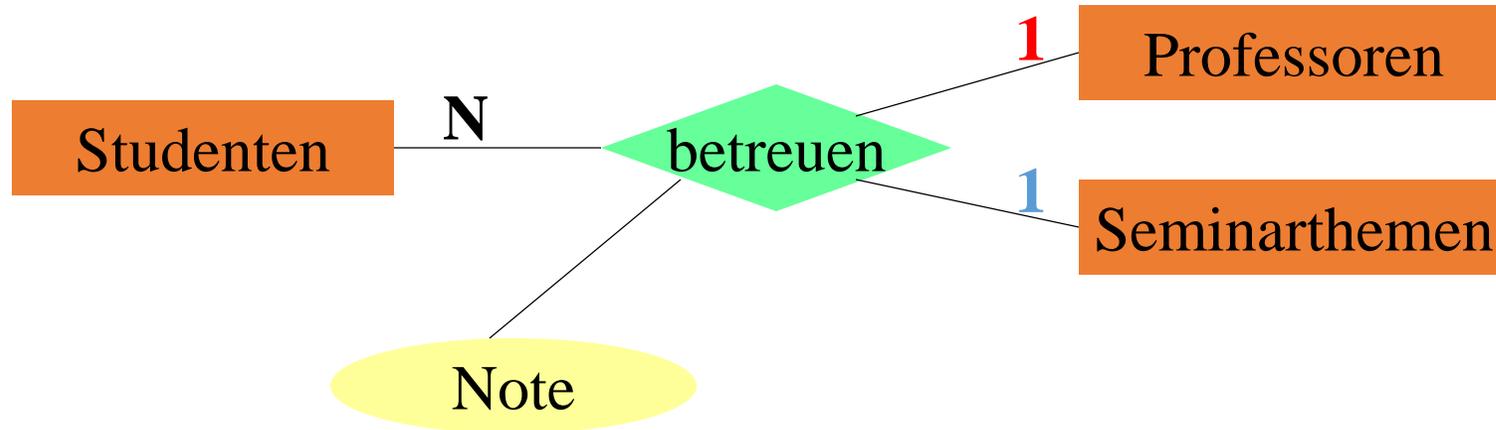
Funktionalitäten bei n-stelligen Beziehungen



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

Beispiel: Seminar

Um das zu verstehen, denken Sie am besten in Abbildungen (Funktionen): eine Abbildung $f(x,y)$ hat pro Parameter x,y nur einen Wert.



betreuen: Professoren \times Studenten \rightarrow Seminarthemen

betreuen: Seminarthemen \times Studenten \rightarrow Professoren

Konsistenzbedingungen des Seminar

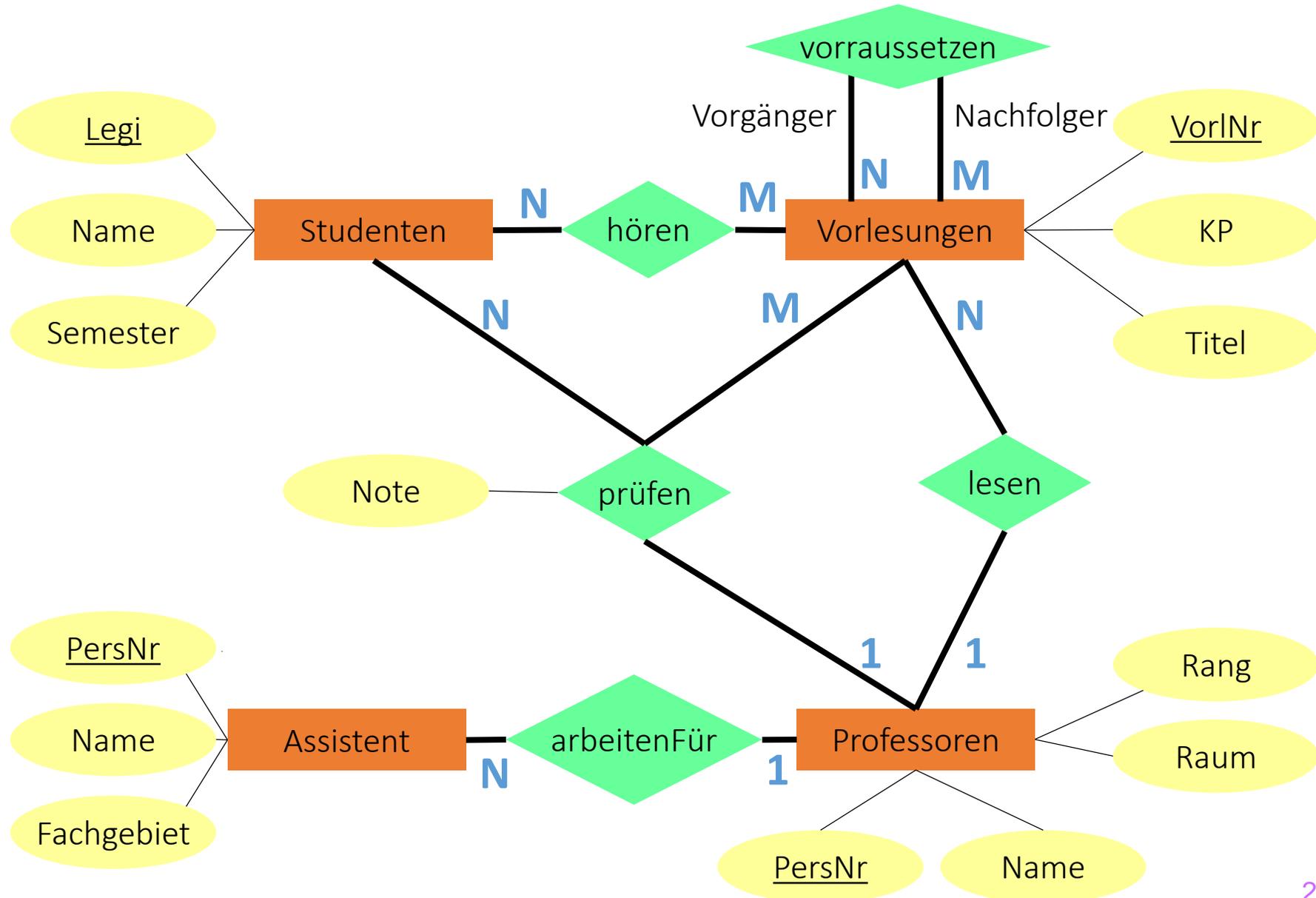
Einschränkungen

1. Studenten dürfen bei einem Professor nur ein Seminarthema bearbeiten
2. Studenten können dasselbe Seminarthema nur einmal bearbeiten

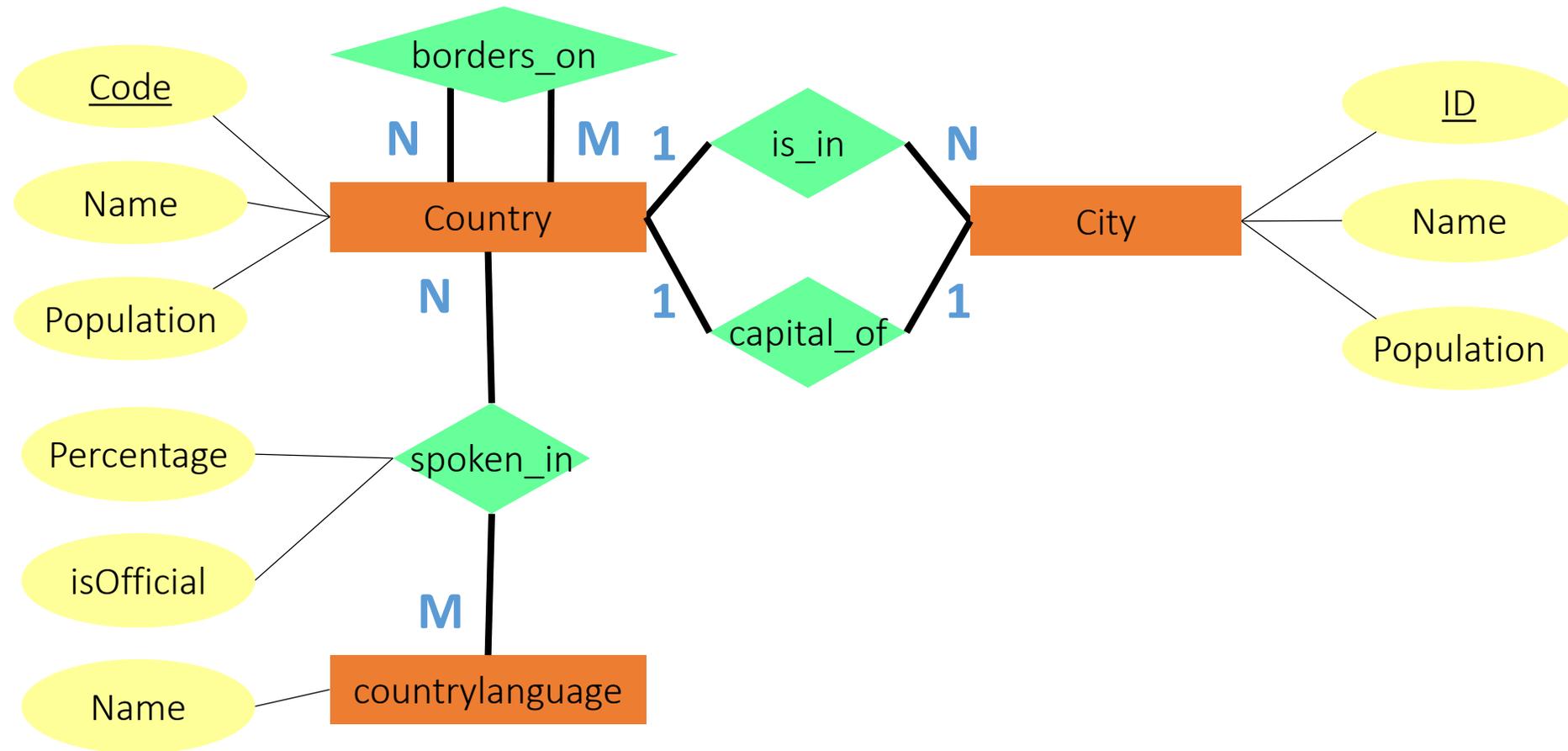
Möglichkeiten

1. Professoren können das Seminarthema für andere Studenten wiederverwenden
2. Dasselbe Thema kann von verschiedenen Professoren verwendet werden

Universität mit Funktionalitäten



Modell der Welt mit Funktionalitäten



Daumenregeln

Wann Attribut, wann Entität?

- Entität, wenn das Konzept mehr als eine Beziehung hat
- Attribut, wenn das Konzept nur eine 1:1 Beziehung hat

Partitionierung von ER-Modellen

- Realistische Modelle sind grösser als eine Seite
- Nach Bereichen / Organisationseinheiten partitionieren
- Kein gutes automatisches Graphenpartitionierungstool bekannt

Tipps

- Keine Redundanz modellieren, keine vermeintliche Leistungsverbesserung anbringen
- Je weniger Entitäten desto besser
- **konzis, vollständig, nachvollziehbar, korrekt**

ER Modellierung: Zusammenfassung

ER beschreibt eine Miniwelt

- Das "was" und die Regeln
- ER ist statisch. Es beschreibt keine Übergänge

Nützlich zum Erstellen von Software zur Beantwortung von (An)fragen über die Miniwelt

- es folgt nun: ER-Modell → relationales Modell

Ähnliche Modellierungsmöglichkeiten bietet **UML** (mehr auf OOP zugeschnitten)

Auch andere graphische Darstellungen des ER Modells gebräuchlich, z.B. "Krähenfussnotation" optisch näher bei UML

Relationales Modell

Die Welt in Tabellen

Relationales Modell, Formalismus

Relation R

- $R \subseteq D_1 \times \dots \times D_n$
- D_1, D_2, \dots, D_n sind Domänen

Beispiel: Telefonbuch \subseteq string \times string \times integer

Tupel: $t \in R$

Beispiel: $t =$ („Mickey Mouse“, „Main Street“, 4711)

Relationenschemata werden wie folgt beschrieben

Telefonbuch: $\{[Name: string, Strasse: string, \underline{Telefon#:integer}]\}$

$\{[...]\}$ deuten an, dass ein Schema eine Menge von Tupeln $[\]$ ist

Name des Attributes

Typ des Attributes

Relationales Modell

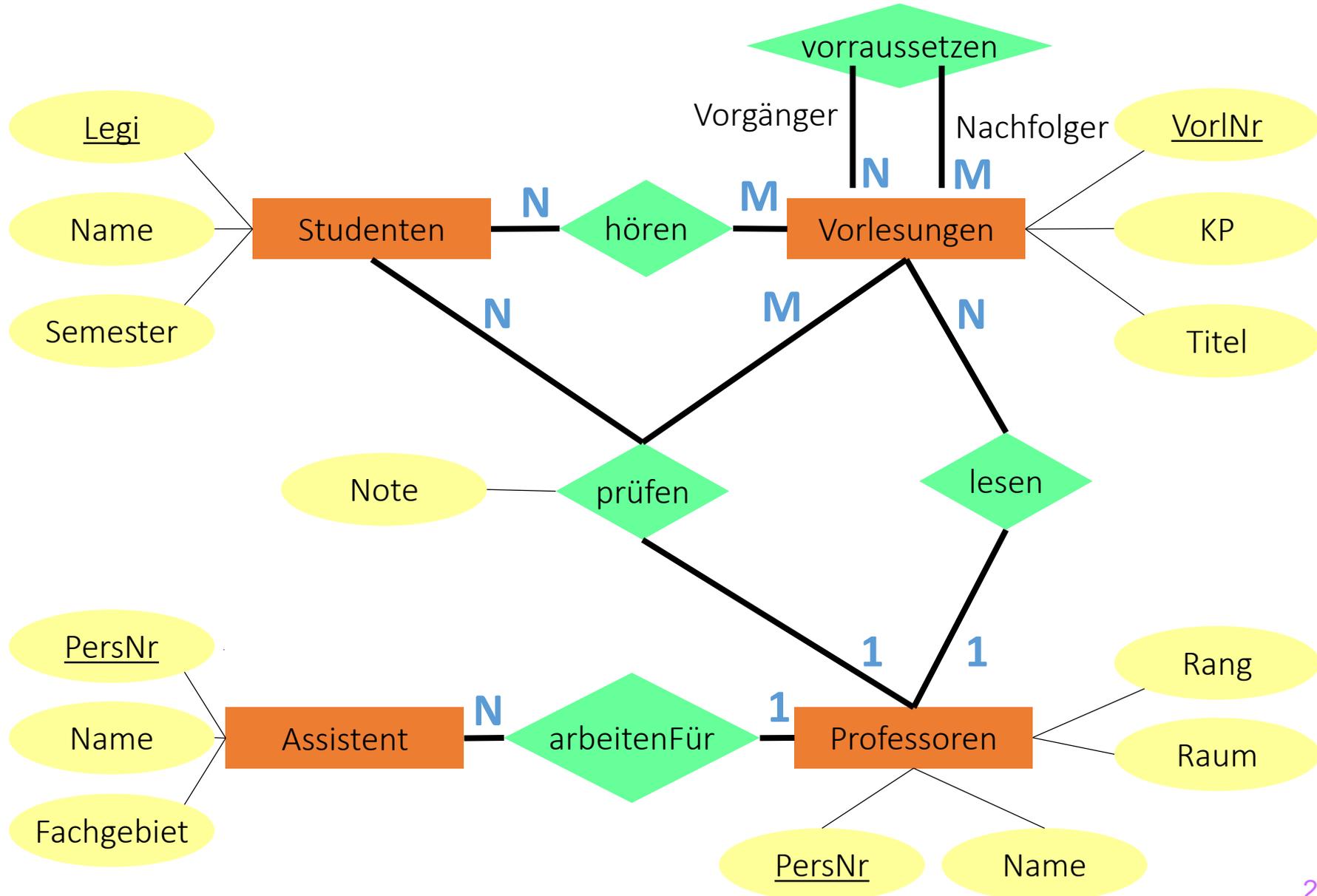
Telefonbuch		
Name	Strasse	<u>Telefon#</u>
Mickey Mouse	Main Street	4711
Minnie Mouse	Broadway	94725
Donald Duck	Broadway	95672
...

Ausprägung: Zustand der Datenbank

Schlüssel: minimale Menge von Attributen, welche ein Tupel eindeutig identifizieren
z.B. {Telefon#} oder {Name, Geburtstag}

Primärschlüssel (durch Unterstreichung hervorgehoben): Ausgewählter Schlüssel, welcher üblicherweise zur Identifikation eines Tupels in einer Relation verwendet wird.

Uni Schema



Regel #1: Darstellung von Entities

Studenten:

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

Vorlesungen:

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

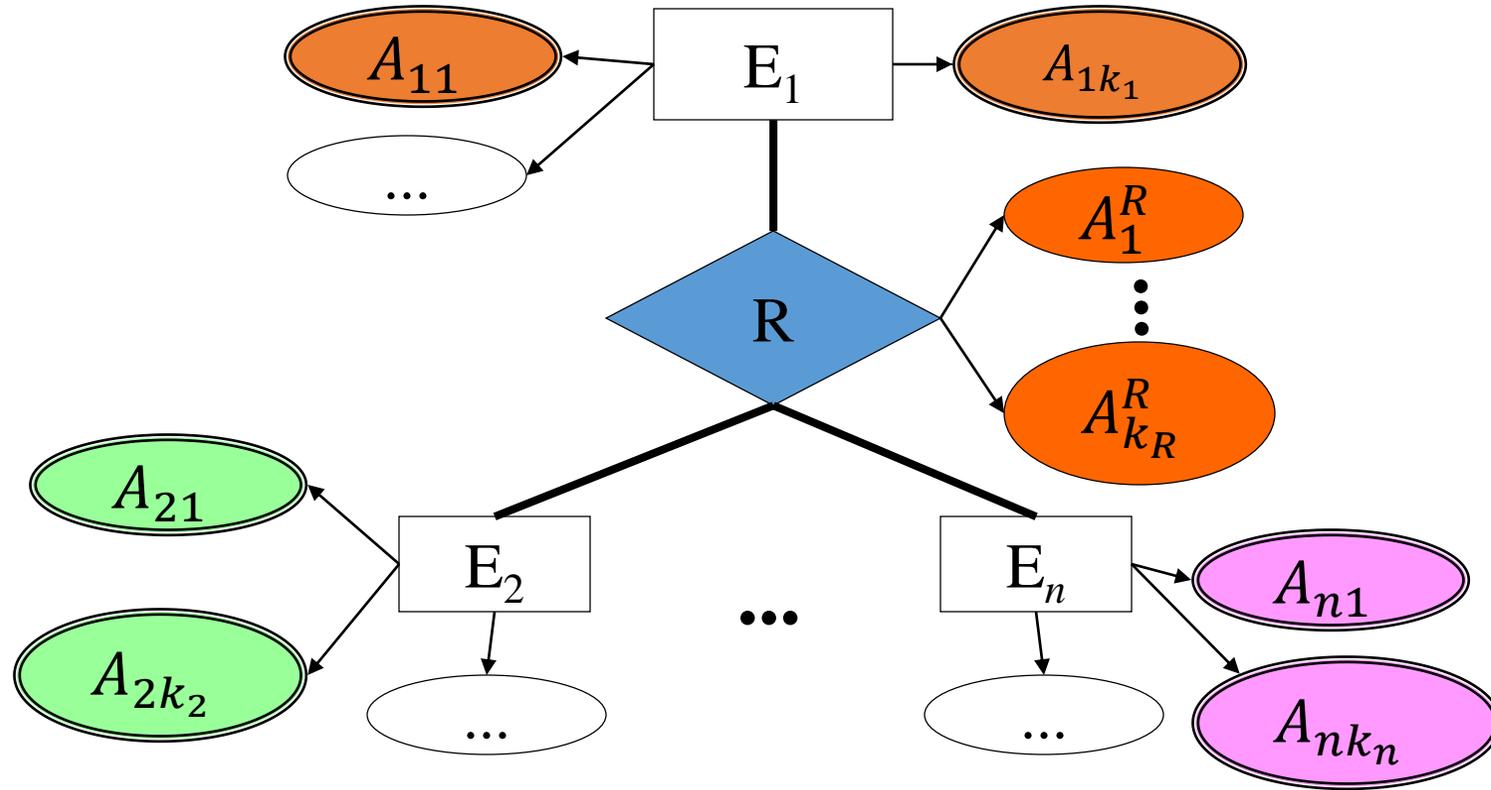
Professoren:

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

Assistenten:

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

Regel #2: Darstellung von Beziehungen



$$R: \left\{ \underbrace{[A_{11}, \dots, A_{1k_1}]}_{\text{Schlüssel } E_1}, \underbrace{[A_{21}, \dots, A_{2k_2}]}_{\text{Schlüssel } E_2}, \dots, \underbrace{[A_{n1}, \dots, A_{nk_n}]}_{\text{Schlüssel } E_n}, \underbrace{[A_1^R, \dots, A_{k_R}^R]}_{\text{Attribute von } R} \right\}$$

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]}



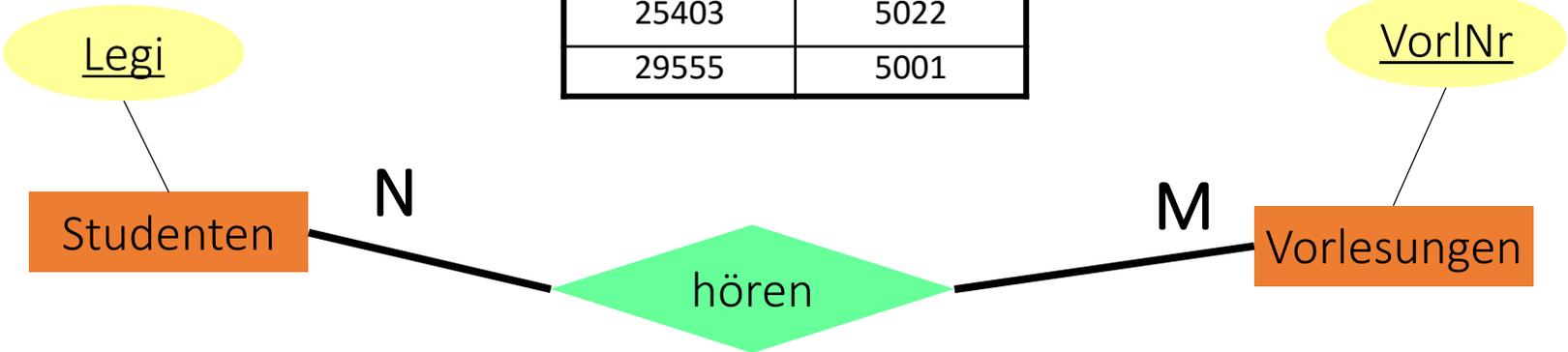
Fremdschlüssel, identifizieren
Tupel aus anderen Entitäten

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	...
...	...

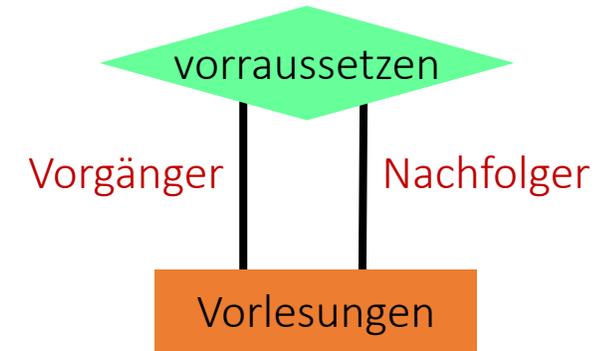


Zur Regel #2: Benennung der Attribute?

Spezifiziert das ER-Modell **Rollen**, dann

- nimm den Namen der jeweiligen Rolle

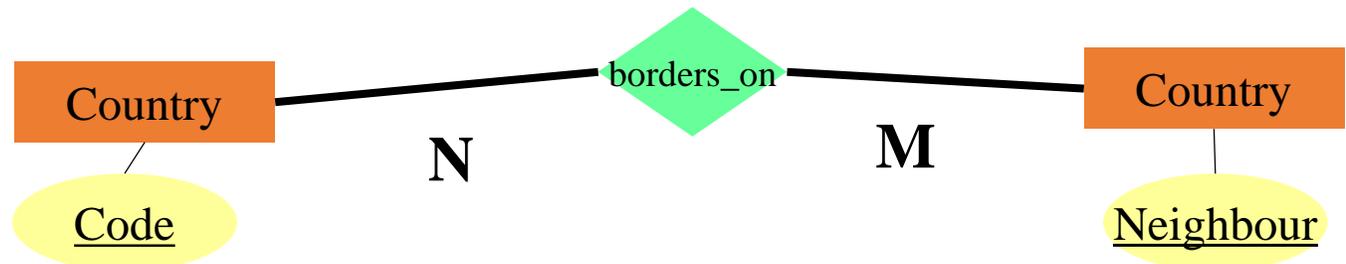
`voraussetzen: {[Vorgänger:string, Nachfolger:string]}`



andernfalls

- benutze die Namen der Schlüsselattribute der Entitäten
- bei Mehrdeutigkeit erfinde aussagekräftigen Namen

`borders_on: {[Code:string, Neighbour:string]}`



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

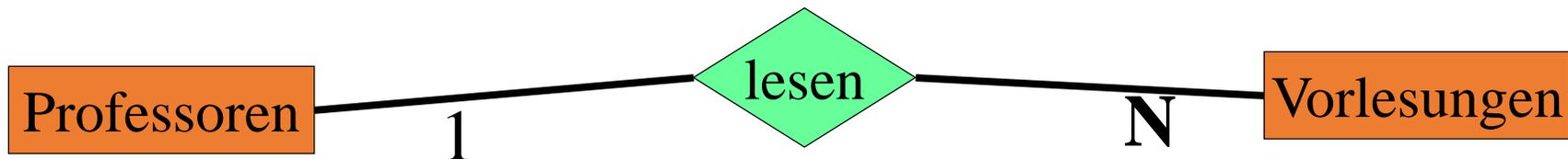


Das funktioniert NICHT

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



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

Die Welt in Tabellen

ID	Name	CountryCode ▲	District	Population
3248	Bern	CHE	Bern	122700
3247	Basel	CHE	Basel-Stadt	166700
3246	Geneve	CHE	Geneve	173500
3245	Zürich	CHE	Zürich	336800
3249	Lausanne	CHE	Vaud	114500
3117	Oldenburg	DEU	Niedersachsen	154125
3130	Remscheid	DEU	Nordrhein-Westfalen	120125
3131	Heilbronn	DEU	Baden-Württemberg	119526
3132	Pforzheim	DEU	Baden-Württemberg	117227
3133	Offenbach am Main	DEU	Hessen	116627
3134	Ulm	DEU	Baden-Württemberg	116103

CountryCode	Language	IsOfficial	Percentage
ABW	Dutch	T	5.3
ABW	English	F	9.5
ABW	Papiamentu	F	76.7
ABW	Spanish	F	7.4
AFG	Balochi	F	0.9
AFG	Dari	T	23.1

Code	Name	Continent	Region	SurfaceArea	IndepYear	Population	LifeExpectancy	GNP	GNPOid	LocalName
ABW	Aruba	North America	Caribbean	193.00	NULL	103000	78.4	828.00	793.00	Aruba
AFG	Afghanistan	Asia	Southern and Central Asia	652090.00	1919	22720000	45.9	5976.00	NULL	Afganistan/Afq
AGO	Angola	Africa	Central Africa	1246700.00	1975	12878000	38.3	6648.00	7984.00	Angola
AIA	Anguilla	North America	Caribbean	96.00	NULL	8000	76.1	63.20	NULL	Anguilla
ALB	Albania	Europe	Southern Europe	28748.00	1912	3401200	71.6	3205.00	2500.00	Shqipëria
AND	Andorra	Europe	Southern Europe	468.00	1278	78000	83.5	1630.00	NULL	Andorra
ANT	Netherlands Antilles	North America	Caribbean	800.00	NULL	217000	74.7	1941.00	NULL	Nederlandse A
ARE	United Arab Emirates	Asia	Middle East	82800.00	1971	2441000	74.1	27066.00	26946.00	Al Emirat al 'Ar

(borders_on nicht modelliert)

Die Relationale Algebra und SQL

Die relationale Algebra

σ Selektion

π Projektion

\times kartesisches Produkt

\bowtie Join (Verbund)

ρ Umbenennung

\cup Vereinigung

$-$ Mengendifferenz

\div Division

\cap Durchschnitt

F Semi-Join (links)

E Semi-Join (rechts)

C linker äusserer Join

D rechter äusserer Join

werden wir nicht diskutieren

SQL (Structured Query Language)

Nachfolger von
Sequel = Structured English Query Language

Familie von Standards

- Anfrage- (Query)-Sprache – Anfragen
- Datendefinitionssprache (DDL) – Schemas
- Datenmanipulationssprache (DML) – Updates

SQL implementiert die Relationale Algebra

Selektion σ

Auswahl von Tupeln (**Zeilen**) der Relation (Tabelle), so dass das **Selektionsprädikat** jeweils erfüllt ist.

$\sigma_{\text{Semester} > 10}$ (Studenten)		
Legi	Name	Semester
24002	Xenokrates	18
25403	Jonas	12

SQL: Anzeigen einer existierenden Tabelle

`SELECT * FROM Tabellename`

gibt den Inhalt einer Tabelle (Relation) aus

`SELECT * FROM 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

SQL Selektion

SELECT * FROM *Tabellen* WHERE *Selektionsprädikat*
entspricht der **Selektion** σ

SELECT * FROM country WHERE SurfaceArea < 10

Code	Name	Continent	Region	SurfaceArea	IndepYear	Population	LifeExpectancy	GNP	GNPOld	LocalName	GovernmentForm
GIB	Gibraltar	Europe	Southern Europe	6.00	NULL	25000	79.0	258.00	NULL	Gibraltar	Dependent Territory of the UK
MCO	Monaco	Europe	Western Europe	1.50	1861	34000	78.8	776.00	NULL	Monaco	Constitutional Monarchy
VAT	Holy See (Vatican City State)	Europe	Southern Europe	0.40	1929	1000	NULL	9.00	NULL	Santa Sede/Città del Vaticano	Independent Church State

$\sigma_{\text{SurfaceArea} < 10}(\text{country})$

Projektion π

Extraktion von Attributen (**Spalten**) der Relation (Tabelle)

$\pi_{\text{Rang}}(\text{Professoren})$
Level
FP
AP

Einfaches SQL Statement

SELECT *Spaltennamen* FROM *Tabellennamen*

entspricht der **Projektion** π ohne Duplikatelimination:

SELECT Rang FROM 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



Rang
FP
FP
AP
AP
AP
FP
FP

Projektion

SELECT DISTINCT *Spaltennamen* FROM *Tabellennamen*

entspricht der **Projektion π** :

SELECT DISTINCT Rang FROM 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



Rang
FP
AP

$\pi_{\text{Rang}}(\text{Professoren})$

Projektion

Anderes Beispiel

```
SELECT Name, Population, SurfaceArea FROM country
```

Name	Population	SurfaceArea
Aruba	103000	193.00
Afghanistan	22720000	652090.00
Angola	12878000	1246700.00
Anguilla	8000	96.00
Albania	3401200	28748.00
Andorra	78000	468.00
Netherlands Antilles	217000	800.00

$\pi_{\text{Name,Population,SurfaceArea}}(\text{country})$

Selektion und Projektion

SELECT *Spalten* FROM *Tabellen* WHERE *Selektionsprädikat*
entspricht der **Projektion** auf das Ergebnis der **Selektion**

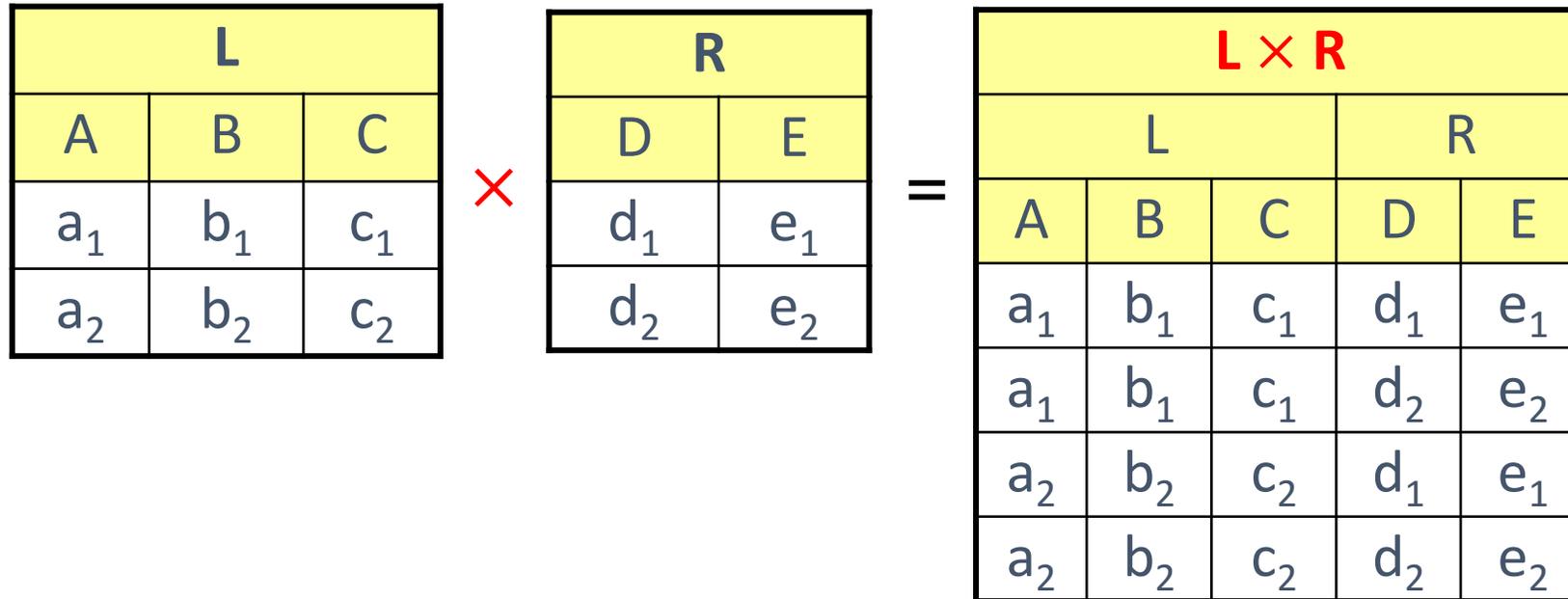
```
SELECT Name, Population FROM country WHERE SurfaceArea < 10
```

Name	Population
Gibraltar	25000
Monaco	34000
Holy See (Vatican City State)	1000

$\pi_{\text{Name,Population}}(\sigma_{\text{SurfaceArea}<10}(\text{country}))$

Kartesisches Produkt \times

enthält alle $|L| \cdot |R|$ mögliche Paare von Tupeln aus L und R



Kartesisches Produkt

Professoren × hören					
Professoren				hören	
PersNr	Name	Rang	Raum	Legi	Nr
2125	Sokrates	FP	226	26120	5001
...
2125	Sokrates	FP	226	29555	5001
...
2137	Kant	FP	7	29555	5001

Riesiges Ergebnis ($| \text{Professoren} | \cdot | \text{hören} |$)

Wird meist in Verbindung mit Selektion eingesetzt
→ Vermeidung riesiger Zwischenergebnisse durch Einführung eines separaten Operators (Join)

Kartesisches Produkt

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

entspricht dem kartesischen Produkt $Tabelle1 \times Tabelle2$

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 \times hören

Umbenennung ρ

Umbenennung von Relationen

Ermittlung indirekter Vorgänger 2. Stufe der Vorlesung 5216

$$\Pi_{V1. \text{Vorgänger}}(\sigma_{V2. \text{Nachfolger}=5216 \wedge V1. \text{Nachfolger} = V2. \text{Vorgänger}}(\rho_{V1}(\text{voraussetzen}) \times \rho_{V2}(\text{voraussetzen})))$$

Umbenennung von Attributen ρ

$$\rho_{\text{Voraussetzung} \leftarrow \text{Vorgänger}}(\text{voraussetzen})$$

Umbenennung von Attributen

SELECT ... *Name as NeuerName* ... FROM Tabellen ...

entspricht der **Umbenennung** ρ .

SELECT Titel, **gelesenVon** as **Dozent** FROM Vorlesungen

VorNr	Titel	KP	gelesenVon
5001	Grundzuege	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Maeeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126



Titel	Dozent
Grundzuege	2137
Ethik	2125
Erkenntnistheorie	2126
Maeeutik	2125
Logik	2125
Wissenschaftstheorie	2126
Bioethik	2126

$\pi_{\text{Titel,Dozent}}(\rho_{\text{gelesenVon} \leftarrow \text{Dozent}}(\text{Vorlesungen}))$

Umbenennung von Tabellen

SELECT ... FROM *Tabellenname Alias* ...
entspricht der Umbenennung ρ .

SELECT v.Titel FROM Vorlesungen v



VorNr	Titel	KP	gelesenVon
5001	Grundzuege	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Maeeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5048	Biethik	2	2126



Titel
Grundzuege
Ethik
Erkenntnistheorie
Maeeutik
Logik
Wissenschaftstheorie
Biethik

$\pi_{\text{Titel}}(\rho_v(\text{Vorlesungen}))$

Kartesisches Produkt plus Selektion

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

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

Etwas kürzer mit Umbenennung

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

Legi	Name	Semester	Legi	VorNr
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

$$\sigma_{s.Legi = h.Legi}(\rho_s(\text{studenten}) \times \rho_h(\text{hören}))$$

Der natürliche Verbund (Join) \bowtie

Gegeben Relationen: $R(A_1, \dots, A_m, B_1, \dots, B_k)$

$S(B_1, \dots, B_k, C_1, \dots, C_n)$

$$R \bowtie S = \Pi_{A_1, \dots, A_m, R.B_1, \dots, R.B_k, C_1, \dots, C_n} \left(\sigma_{R.B_1=S.B_1 \wedge \dots \wedge R.B_k=S.B_k} (R \times S) \right)$$

R \bowtie S											
R - S				R \cap S				S - R			
A ₁	A ₂	...	A _m	B ₁	B ₂	...	B _k	C ₁	C ₂	...	C _n
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Beispiel: hören \bowtie Vorlesungen

Legi	VorNr
25403	5022
26120	5001
27550	4052
27550	5001
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5001
29555	5022

L



VorNr	Titel	KP	gelesenVon
5001	Grundzuege	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Maeeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126

R

=

Legi	VorNr	Titel	KP	gelesenVon
25403	5022	Glaube und Wissen	2	2134
26120	5001	Grundzuege	4	2137
27550	4052	Logik	4	2125
27550	5001	Grundzuege	4	2137
28106	5041	Ethik	4	2125
28106	5052	Wissenschaftstheorie	3	2126
28106	5216	Bioethik	2	2126
28106	5259	Der Wiener Kreis	2	2133
29120	5001	Grundzuege	4	2137
29120	5041	Ethik	4	2125
29120	5049	Maeeutik	2	2125
29555	5001	Grundzuege	4	2137
29555	5022	Glaube und Wissen	2	2134

$L - R$

$R \cap L$

$R - L$

Beispiel: Studenten \bowtie hören

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
4711	Unbekannter	NULL



Legi	VorINr
25403	5022
26120	5001
27550	4052
27550	5001
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5001
29555	5022



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

Drei-Wege-Join: Studenten \bowtie hören \bowtie Vorlesungen

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
4711	Unbekannter	NULL



Legi	VorlNr
25403	5022
26120	5001
27550	4052
27550	5001
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5001
29555	5022



VorlNr	Titel	KP	gelesenVon
5001	Grundzuege	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Maeeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126



Legi	Name	Semester	VorlNr	Titel	KP	gelesenVon
25403	Jonas	12	5022	Glaube und Wissen	2	2134
26120	Fichte	10	5001	Grundzuege	4	2137
27550	Schopenhauer	6	4052	Logik	4	2125
27550	Schopenhauer	6	5001	Grundzuege	4	2137
28106	Carnap	3	5041	Ethik	4	2125
28106	Carnap	3	5052	Wissenschaftstheorie	3	2126
28106	Carnap	3	5216	Bioethik	2	2126
28106	Carnap	3	5259	Der Wiener Kreis	2	2133
29120	Theophrastos	2	5001	Grundzuege	4	2137
29120	Theophrastos	2	5041	Ethik	4	2125
29120	Theophrastos	2	5049	Maeeutik	2	2125
29555	Feuerbach	2	5001	Grundzuege	4	2137
29555	Feuerbach	2	5022	Glaube und Wissen	2	2134

Join ist assoziativ:

(Studenten \bowtie hören) \bowtie Vorlesungen =
 Studenten \bowtie (hören \bowtie Vorlesungen)

Join

SELECT ... FROM *Tabelle1 NATURAL JOIN Tabelle2* ...

entspricht dem Join 

SELECT *
FROM studenten NATURAL JOIN hören

Im Unterschied zu

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

kommt Legi nur einmal vor in der Tabelle:

Join fasst gleichnamige Attribute zusammen.

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 **A** hören

Allgemeiner Join (Theta-Join)

Gegeben Relationen(-Schemata)

$R(A_1, \dots, A_n)$ und
 $S(B_1, \dots, B_m)$

$$R \bowtie_{\theta} S = \sigma_{\theta} (R \times S)$$

$R \bowtie_{\theta} S$							
R				S			
A_1	A_2	...	A_n	B_1	B_2	...	B_m
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Beispiel: Allgemeiner Join

Relationen Züge(name, start, ziel, ..., länge)
 Gleise(station, nummer, ..., länge)

Finde alle möglichen Gleise für den „CIS Alpino“ in Zürich

$\sigma_{\text{station}=\text{„Zürich“}}$ (Gleise)

\bowtie Züge.länge < Gleise.länge

$\sigma_{\text{name}=\text{„CIS“}}$ (Züge)