



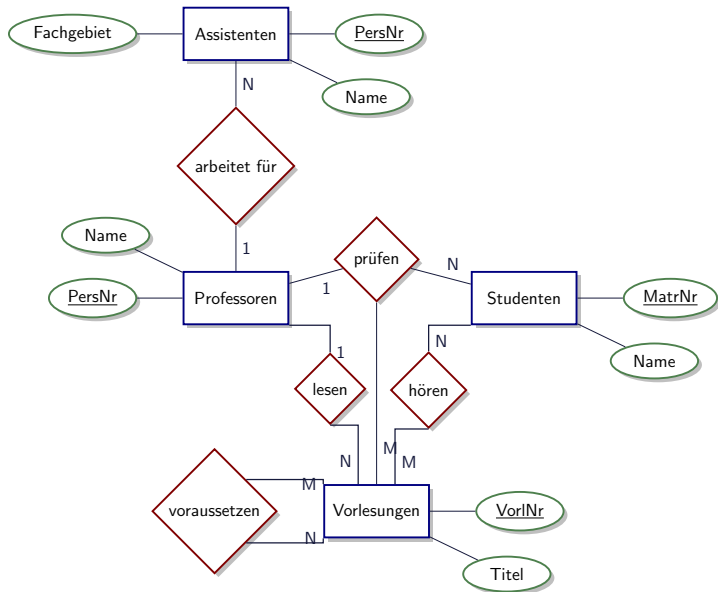
# Informationssysteme

Sommersemester 2016

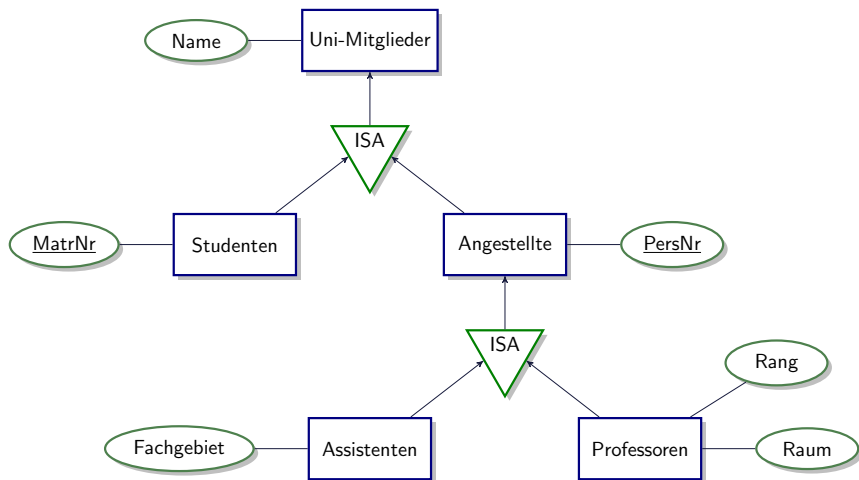
Prof. Dr.-Ing. Sebastian Michel  
TU Kaiserslautern

[smichel@cs.uni-kl.de](mailto:smichel@cs.uni-kl.de)

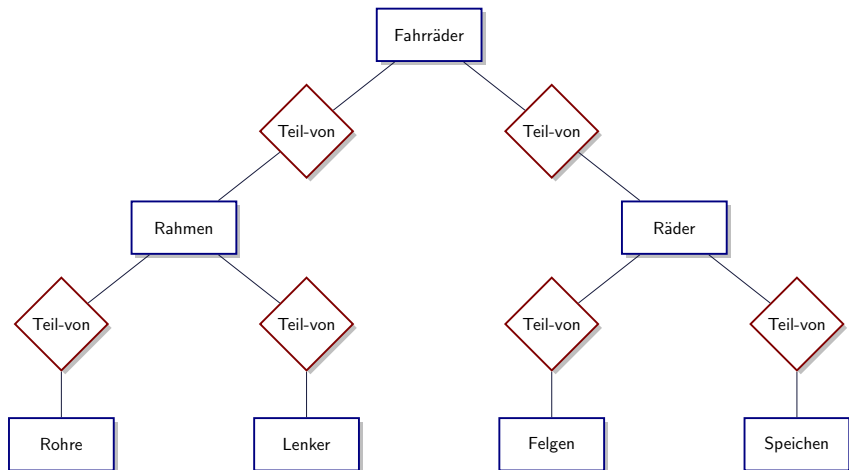
# Wiederholung: Das E/R-Modell der Universität



# Generalisierung

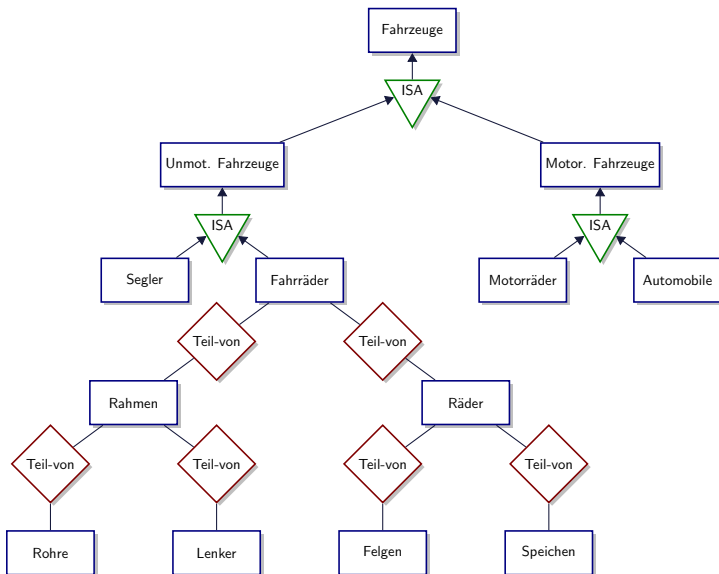


# Aggregation

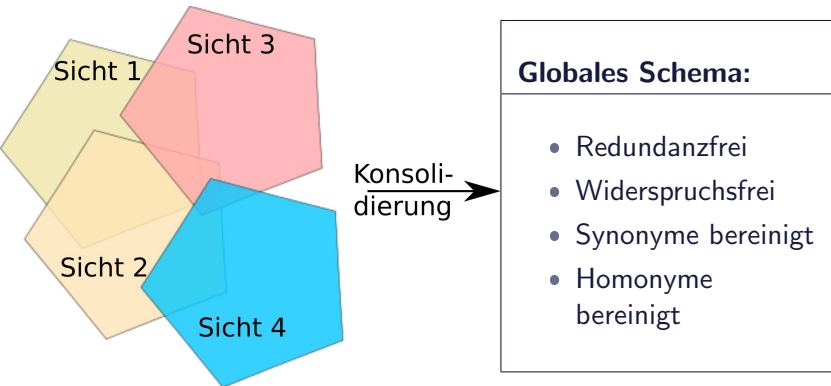


und so weiter ...

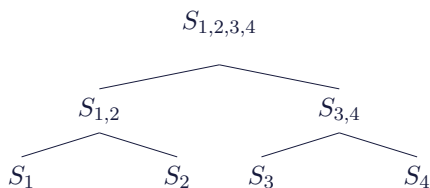
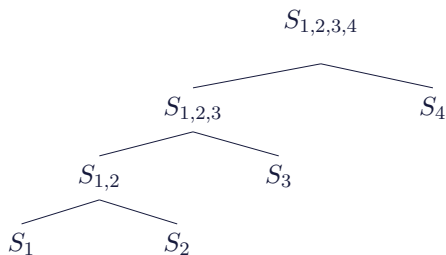
# Generalisierung und Aggregation



# Konsolidierung von Teilschemata oder Sichtenintegration

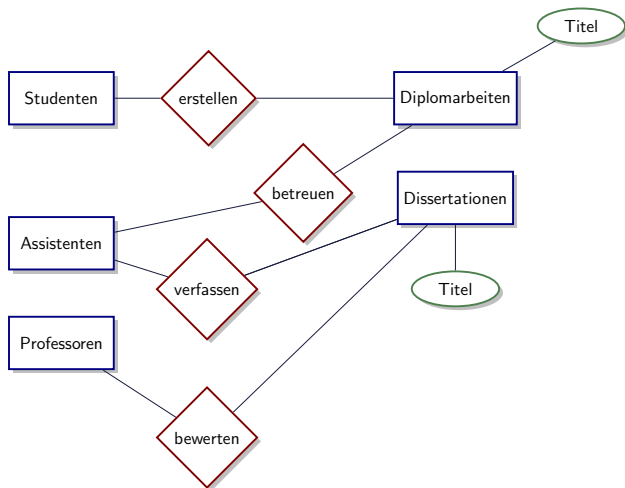


# Mögliche Konsolidierungsbäume



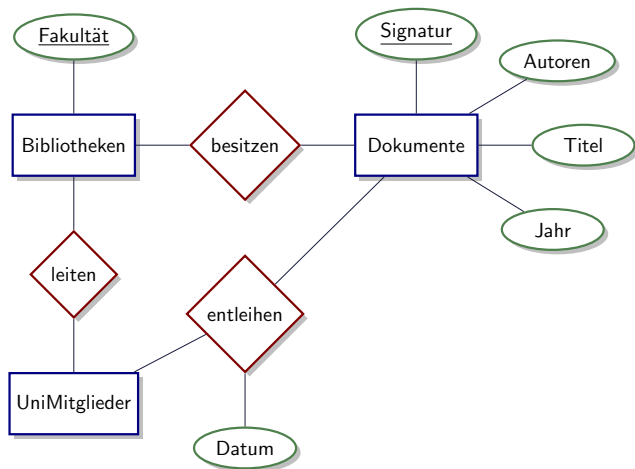
- Gegeben: Teilschemata  $S_1$ ,  $S_2$ ,  $S_3$  und  $S_4$ .
- Mögliche Konsolidierungsbäume zur Herleitung des globalen Schemas  $S_{1,2,3,4}$
- Oben: maximal hoher Konsolidierungsbaum (links-tief).
- Unten: minimal hoher Konsolidierungsbaum (balanciert).
- Vorgehensweise hängt vom Einzelfall ab.

# Drei Sichten einer Universitäts-Datenbank

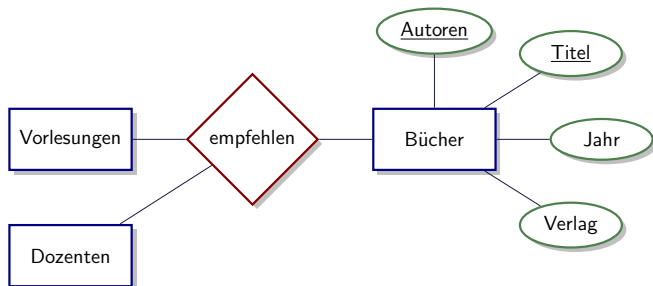


## Sicht 1: Erstellung von Dokumenten als Prüfungsleistung





## Sicht 2: Bibliotheksverwaltung

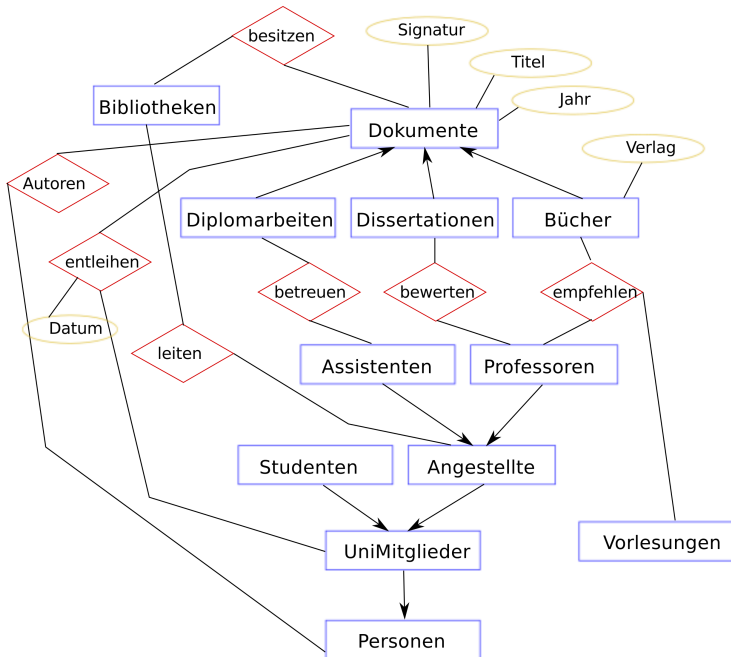


### Sicht 3: Buchempfehlungen für Vorlesung

# Beobachtungen für die Konsolidierung

- Die Begriffe **Dozenten** und **Professoren** sind synonym verwendet worden.
- Der Entitytyp **UniMitglieder** ist eine **Generalisierung** von **Studenten**, **Professoren** und **Assistenten**.
- Fakultätsbibliotheken werden sicherlich von Angestellten (und nicht von Studenten) geleitet. → Beziehung **leiten** in Sicht 2 muss angepasst werden. Insbesondere da im globalen Schema eine Spezialisierung von UniMitglieder in **Studenten** und **Angestellte** vorgenommen wird.
- **Dissertationen**, **Diplomarbeiten** und **Bücher** sind Spezialisierungen von **Dokumenten**, die in den Bibliotheken verwaltet werden.

- Wir können davon ausgehen, dass alle an der Universität erstellten Diplomarbeiten und Dissertationen in Bibliotheken verwaltet werden.
- Die in Sicht 1 festgelegten Beziehungen **erstellen** und **verfassen** modellieren denselben Sachverhalt wie das Attribut **Autoren** von **Bücher** in Sicht 3.
- Alle in einer Bibliothek verwalteten Dokumente werden durch die Signatur identifiziert.



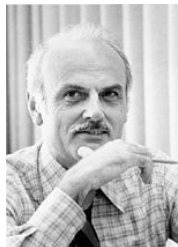
# Datenmodellierung mit UML (nur ganz kurz erwähnt)

- Unified Modelling Language UML
- De-facto Standard für den objektorientierten Software- Entwurf
- Zentrales Konstrukt ist die Klasse (class) mit der gleichartige Objekte modelliert werden hinsichtlich:
- Struktur ( $\approx$ Attribute)
- Verhalten ( $\approx$ Operationen/Methoden)
- Assoziationen zwischen Klassen entsprechen Beziehungstypen
- Generalisierungshierarchien
- Aggregation
- für Datenbankentwurf ist strukturelle Modellierung entscheidend
- Modellierung von Verhalten weniger wichtig

# Das relationale Modell

# Grundlagen des relationalen Modells

- Entwickelt von Edgar. F. Codd in den Siebzigern ( $\Rightarrow$  Turing-Award)
- Das am weitesten verbreitete Datenmodell
- Wird aber nach und nach um objektorientierte Konzepte erweitert





# Grundlagen des relationalen Modells

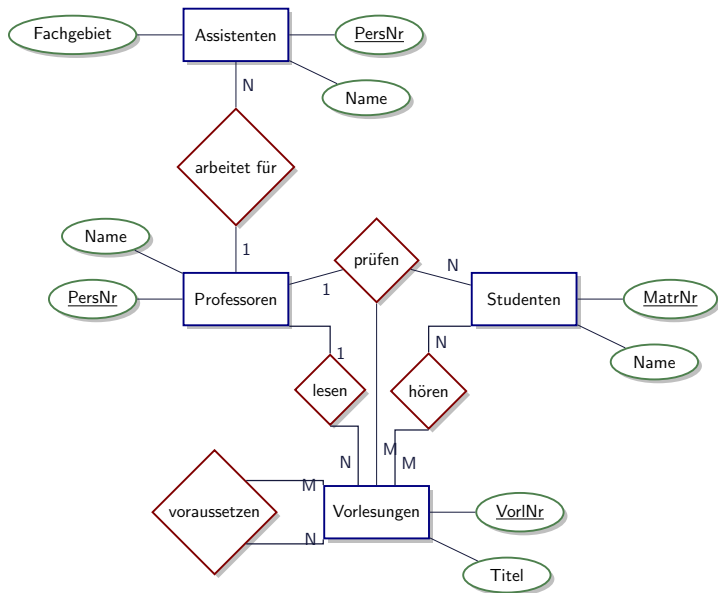
Seien  $D_1, D_2, \dots, D_n$  Domänen (=Wertebereiche)

- **Relation:**  $R \subseteq D_1 \times \dots \times D_n$   
Beispiel:  $Telefonbuch \subseteq string \times string \times integer$
- Wertebereiche dürfen identisch sein:  $D_i = D_j$  für  $i \neq j$
- **Relationenschema:** legt die Struktur der gespeicherten Daten fest.  
Wird mit  $sch(R)$  oder  $\mathcal{R}$  bezeichnet.  
Beispiel: Telefonbuch:  
 $\{[Name : string, Adresse : string, \underline{Telefon : integer}]\}$
- **Tupel:**  $t \in R$   
Beispiel:  $t = ("Mickey Mouse", "Main Street", 4711);$

Telefonbuch		
Name	Straße	<u>Telefon</u>
Mickey Mouse	Main Street	4711
Minnie Mouse	Broadway	94725
Donald Duck	Broadway	95622
...	...	...

- **Ausprägung:** der aktuelle Zustand der Datenbasis, Teilmenge des Kreuzproduktes
- **Schlüssel:** minimale Menge von Attributen, deren Werte ein Tupel eindeutig identifizieren
- **Primärschlüssel:** wird unterstrichen
  - Einer der Schlüsselkandidaten wird ausgewählt
  - Hat eine besondere Bedeutung bei der Referenzierung von Tupeln
  - später mehr dazu
- **Fremdschlüssel** verweist auf den Primärschlüssel einer anderen Relation.

## Universitätsschema



# Von Entitätstypen zu Relationenschemata

- **Studenten:** {[MatrNr: integer, Name: string, Semester: integer]}
- **Vorlesungen:** {[VorlNr: integer, Titel: string, SWS: integer]}
- **Professoren:** {[PersNr: integer, Name: string, Rang: string, Raum: integer]}
- **Assistenten:** {[PersNr: integer, Name: string, Fachgebiet: string]}

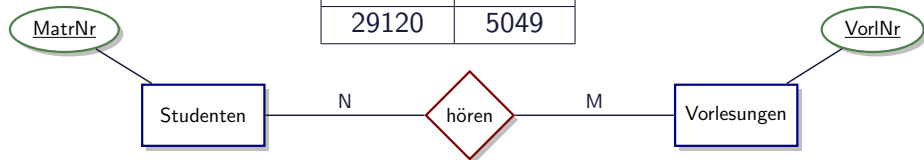
Im Folgenden lassen wir die Datentypen (z.B. integer) meist weg.

## Umsetzung von N:M-Beziehungen

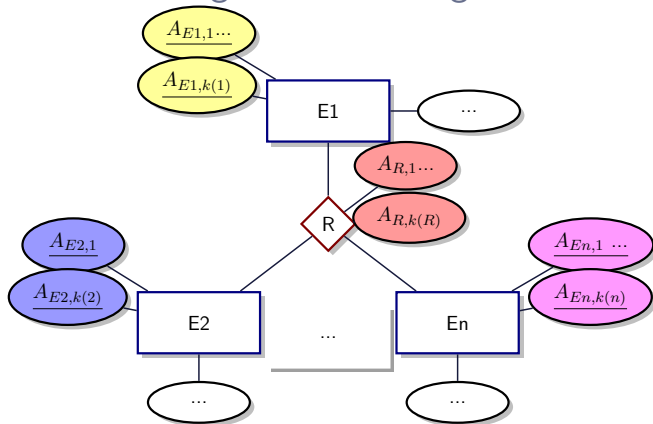
Studenten	
MatrNr	...
26120	...
27550	...
...	...

hören	
MatrNr	VorlNr
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049

Vorlesungen	
VorlNr	...
5001	...
4052	...
...	...

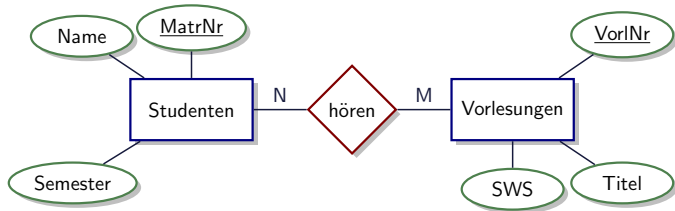


# Relationale Darstellung von Beziehungen



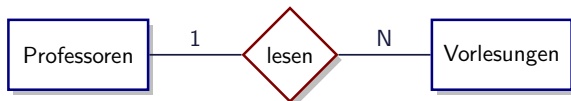
$R: \{ [ \underbrace{A_{E1,1}, \dots, A_{E1,k(1)}}_{\text{Schlüssel von } E_1}, \underbrace{A_{E2,1}, \dots, A_{E2,k(2)}}_{\text{Schlüssel von } E_2}, \dots, \underbrace{A_{En,1}, \dots, A_{En,k(n)}}_{\text{Schlüssel von } E_n}, \underbrace{A_{R,1}, \dots, A_{R,k(R)}}_{\text{Attribute von } R} ] \}$

# Beziehungen unseres Beispiel-Schemas



- **hören:** { MatrNr: integer, VorlNr: integer }
- **lesen:** { [ PersNr: integer, VorlNr: integer ] }
- **arbeitenFür:** { [ AssistentenPersNr: integer, ProfPersNr: integer ] }
- **voraussetzen:** { [ Vorgänger: integer, Nachfolger: integer ] }
- **prüfen:** { [ MatrNr: integer, VorlNr: integer, PersNr: integer, Note: decimal ] }

# Umsetzung von 1:N-Beziehungen



## Initial-Entwurf:

- **Vorlesungen:** {[ VorlNr, Titel, SWS ]}
- **Professoren:** {[ PersNr, Name, Rang, Raum ]}
- **lesen:** {[ VorlNr, PersNr ]}



## Umsetzung von 1:N-Beziehungen (2)

### Initial-Entwurf:

- **Vorlesungen:** {[ VorlNr, Titel, SWS ]}
- **Professoren:** {[ PersNr, Name, Rang, Raum ]}
- **lesen:** {[ VorlNr, PersNr ]}

### Verfeinerung durch Zusammenfassung:

- **Vorlesungen:** {[ VorlNr, Titel, SWS, gelesenVon ]}
- **Professoren:** {[ PersNr, Name, Rang, Raum ]}

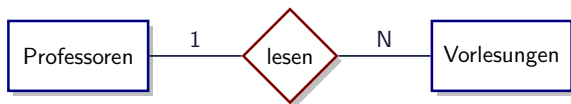
### Regel:

Relationen mit gleichem Schlüssel können zusammengefasst werden ...  
**aber nur diese und keine anderen!**

# Äußerung von Professoren und Vorlesung

Professoren			
PersNr	Name	Rang	Raum
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

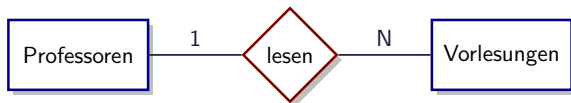
Vorlesungen			
VorlNr	Titel	SWS	GelesenVon
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137



# Vorsicht: So geht es NICHT

Professoren				
PersNr	Name	Rang	Raum	liest
2125	Sokrates	C4	226	5041
2125	Sokrates	C4	226	5049
2125	Sokrates	C4	226	4052
...	...	...	...	...
2134	Augustinus	C3	309	5022
2136	Curie	C4	36	??
...	...	...	...	...

Vorlesungen		
VorlNr	Titel	SWS
5001	Grundzüge	4
5041	Ethik	4
5043	Erkenntnistheorie	3
5049	Mäeutik	2
4052	Logik	4
5052	Wissenschaftstheorie	3
5216	Bioethik	2
5259	Der Wiener Kreis	2
5022	Glaube und Wissen	2
4630	Die 3 Kritiken	4



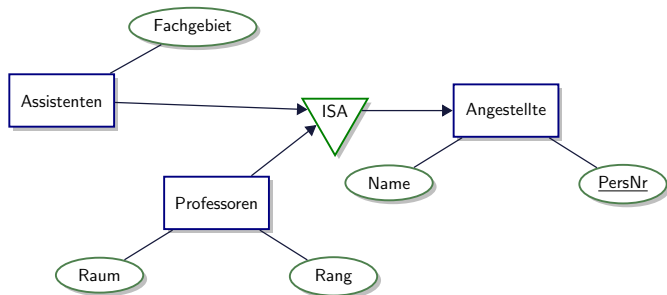
# Wieso kann (wird) es Probleme geben?

Professoren				
PersNr	Name	Rang	Raum	liest
2125	Sokrates	C4	226	5041
2125	Sokrates	C4	226	5049
2125	Sokrates	C4	226	4052
...	...	...	...	...
2134	Augustinus	C3	309	5022
2136	Curie	C4	36	??
...	...	...	...	...

Vorlesungen		
VorlNr	Titel	SWS
5001	Grundzüge	4
5041	Ethik	4
5043	Erkenntnistheorie	3
5049	Mäeutik	2
4052	Logik	4
5052	Wissenschaftstheorie	3
5216	Bioethik	2
5259	Der Wiener Kreis	2
5022	Glaube und Wissen	2
4630	Die 3 Kritiken	4

- **Update-Anomalie:** Was passiert wenn Sokrates umzieht?
- **Lösch-Anomalie:** Was passiert wenn "Glaube und Wissen wegfällt?"
- **Einfüge-Anomalie:** Curie ist neu und liest noch keine Vorlesungen
- ...

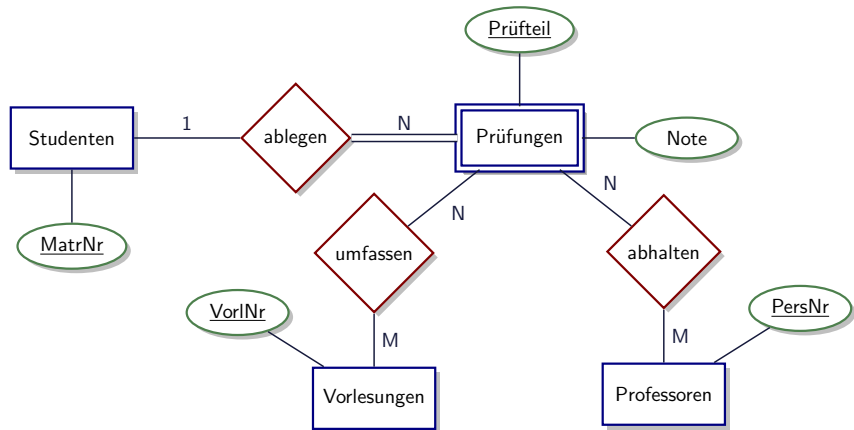
# Relationale Modellierung der Generalisierung



- Angestellte: {[PersNr, Name]}
- Professoren: {[PersNr, Rang, Raum]}
- Assistenten: {[PersNr, Fachgebiet]}

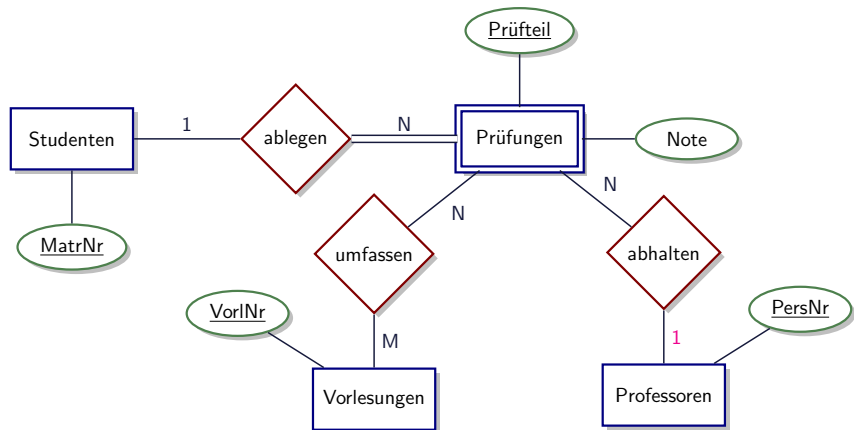
Dies ist nur eine Möglichkeit der Abbildung der Generalisierung auf Relationen. **Welche Alternativen gibt es?**

# Relationale Modellierung schwacher Entitytypen



- Prüfungen: {[MatrNr: integer, Prüfteil: string, Note: integer]}
- umfassen: {[MatrNr: integer, Prüfteil: string, VorlNr: integer]}
- abhalten: {[MatrNr: integer, Prüfteil: string, PersNr: integer]}

# Nur ein Prüfer je Prüfung: Was passiert?



- Prüfungen: {[MatrNr: integer, Prüfteil: string, Note: integer, PersNr: integer]}
- umfassen: {[MatrNr: integer, Prüfteil: string, VorlNr: integer]}

# Ergebnis: Die relationale Uni-DB

Professoren			
PersNr	Name	Rang	Raum
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

Studenten		
MatrNr	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	SWS	gelesen von
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

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

hören	
MatrNr	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

prüfen			
MatrNr	VorlNr	PersNr	Note
28106	5001	2126	1
25403	5041	2125	2
27550	4630	2137	2

Assistenten			
PersNr	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



# Formulierung und einfache Auswertung von Anfragen

# Übersicht

Inhalte der nächsten Vorlesungen:

- **Konjunktive regelbasierte Anfragen:**

$ans(x_{na}) \leftarrow Professoren(x_{pn}, x_{na}, 'C4', x_{ra})$

- **Relationenkalküle:**  $\{t.name \mid t \in Professoren \wedge t.rang = 'C4'\}$
- **Relationale Algebra:**  $\pi_{name}(\sigma_{rang='C4'}(Professoren))$
- **SQL:** SELECT name FROM Professoren WHERE rang='C4'

Wir betrachten die Datenbank

CINEMA = {Movies, Location, Pariscope}

wobei die Relationen **Movies**, **Location** und **Pariscop**e die folgenden Schemata haben:

$sch(Movies) = \{Title, Director, Actor\}$

$sch(Location) = \{Theater, Address, Phone Number\}$

$sch(Pariscop) = \{Theater, Title, Schedule\}$

Beispiele und Notationen folgen weitestgehend dem Buch "Foundations of Databases" S.Abiteboul, R. Hull und V. Vianu. PDF ist verfügbar unter <http://webdam.inria.fr/Alice/>.

<b>Movies</b>		
<b>Title</b>	<b>Director</b>	<b>Actor</b>
Immer Ärger mit Harry	Hitchcock	Gwenn
Immer Ärger mit Harry	Hitchcock	Forsythe
Immer Ärger mit Harry	Hitchcock	MacLaine
Immer Ärger mit Harry	Hitchcock	Hitchcock
.....	....	....
Schreie und Flüstern	Bergman	Andersson
Schreie und Flüstern	Bergman	Sylwan
Schreie und Flüstern	Bergman	Thulin
Schreie und Flüstern	Bergman	Ullman

<b>Location</b>		
<b>Theater</b>	<b>Address</b>	<b>Phone Number</b>
Gaumont Opéra	31 bd. des Italiens	47 42 60 33
Saint André des Arts	30 rue Saint André des Arts	43 26 48 18
Le Champo	51 rue des Ecoles	43 54 51 60
....	....	...
Georges V	144 av. des Champs-Élysées	45 62 41 46
Les 7 Montparnassiens	98 bd. du Montparnassiens	43 20 32 20

<b>Pariscope</b>		
<b>Theater</b>	<b>Title</b>	<b>Schedule</b>
Gaumont Opéra	Schreie und Flüstern	20:30
Saint André des Arts	Immer Ärger mit Harry	20:15
Georges V	Schreie und Flüstern	22:15
....	....	....
Les 7 Montparnassiens	Schreie und Flüstern	20:45

# Anfragen

- Wer ist der Regisseur von “Schreie und Flüstern”?
- In welchem Theater läuft “Schreie und Flüstern”?
- Was ist die Adresse und die Telefonnummer des Theaters “Le Champo”?

## Gib Namen und Adressen der Theater aus, die einen Bergman-Film spielen.

Zur Erinnerung:

$sch(Movies) = \{Title, Director, Actor\}$

$sch(Location) = \{Theater, Address, Phone Number\}$

$sch(Pariscope) = \{Theater, Title, Schedule\}$

### Kann wie folgt berechnet werden:

Falls es jeweils Tupel  $r_1$ ,  $r_2$ ,  $r_3$  aus den Relationen  $Movies$ ,  $Pariscope$ ,  $Location$  gibt, so dass  
der Regisseur in  $r_1$  "Bergman" ist  
und die Titel der Tupel  $r_1$  und  $r_2$  gleich sind  
und die Theater in Tupel  $r_2$  und  $r_3$  gleich sind  
dann möchten wir Theater und Adresse von Tupel  $r_3$ .



Falls es jeweils in den Relationen Movies, Pariscopie und Location Tupel  $\langle x_{ti}, \text{"Bergman"}, x_{ac} \rangle$ ,  $\langle x_{th}, x_{ti}, x_s \rangle$  und  $\langle x_{th}, x_{ad}, x_s \rangle$  gibt dann nehme das Tupel  $\langle \text{Theater} : x_{th}, \text{Address} : x_{ad} \rangle$  in die Antwort auf.

$x_{ti}$ ,  $x_{th}$  etc. sind Variablen.

## Wir können die Anfrage umschreiben als

$$ans(x_{th}, x_{ad}) \leftarrow \text{Movies}(x_{ti}, \text{"Bergman"}, x_{ac}), \text{Pariscopie}(x_{th}, x_{ti}, x_s), \\ \text{Location}(x_{th}, x_{ad}, x_p)$$

wobei  $ans$  (für Answer/Antwort) eine Relation über  $\{\text{Theater}, \text{Address}\}$  ist

## Kopf (Head) und Rumpf (Body)

Der Ausdruck links von  $\leftarrow$  wird als Kopf bezeichnet. Der Ausdruck rechts von  $\leftarrow$  heisst Rumpf.

$$\text{ans}(x_{th}, x_{ad}) \leftarrow \text{Movies}(x_{ti}, \text{"Bergman"}, x_{ac}), \text{Pariscope}(x_{th}, x_{ti}, x_s), \\ \text{Location}(x_{th}, x_{ad}, x_p)$$

## Der obige Ausdruck kann noch vereinfacht werden als

$$\text{ans}(x_{th}, x_{ad}) \leftarrow \text{Movies}(x_{ti}, \text{'Bergman'}, \_), \text{Pariscope}(x_{th}, x_{ti}, \_), \\ \text{Location}(x_{th}, x_{ad}, \_)$$

wobei das Zeichen  $\_$  benutzt wird, um alle Variablen zu ersetzen, die nur ein Mal auftreten.

# Regelbasierte Konjunktive Anfrage

Sei  $\mathbf{R}$  ein Datenbankschema. Eine **regelbasierte konjunktive Anfrage** über  $\mathbf{R}$  hat die Form

$$ans(u) \leftarrow R_1(u_1), \dots, R_n(u_n)$$

- $u, u_1, \dots, u_n$  sind Tupel
- **Notation:** falls  $v = \langle x_1, \dots, x_m \rangle$  dann schreiben wir  $R(v)$  anstelle von  $R(x_1, \dots, x_m)$
- $u_i$  muss Stelligkeit (arity) passend zu  $R_i$  haben.
- Jede Variable aus  $u$  muss mindestens ein Mal in einem der  $u_1, \dots, u_n$  auftreten.

# Semantik von Regelbasierten Konjunktiven Anfragen

Betrachtet man eine **Instanz einer Datenbank**, also Ausprägungen von Tupeln in den einzelnen Relationen, dann kann man sich eine Regel so vorstellen, dass wenn man eine gültige Belegung des Rumpfes gefunden hat, der Kopf der Regel das Ergebnis darstellt.

Betrachten wir wieder Anfrage  $q$  als

$$ans(u) \leftarrow R_1(u_1), \dots, R_n(u_n)$$

und eine Instanz (Ausprägung)  $\mathbf{I}$  der Datenbank  $\mathbf{R}$ . **Das Abbild von  $\mathbf{I}$  unter Anfrage  $q$  ist**

$$q(\mathbf{I}) = \{\nu(u) \mid \nu \text{ ist eine Belegung der Variablen aus } q \\ \text{und } \nu(u_i) \in I(R_i), \forall i \in [1, n]\}$$

## Belegung von Variablen: Beispiel

$ans(x_{th}, x_{ad}) \leftarrow \text{Movies}(x_{ti}, \text{"Bergman"}, x_{ac}), \text{Pariscope}(x_{th}, x_{ti}, x_s),$   
 $\text{Location}(x_{th}, x_{ad}, x_p)$

### Betrachten wir folgende Belegung der Variablen:

$\nu(x_{ti}) = \text{"Schreie und Flüstern"}$

$\nu(x_{ac}) = \text{"Ullman"}$

$\nu(x_{th}) = \text{"Gaumont Opera"}$

$\nu(x_s) = \text{"20:30"}$

$\nu(x_{ad}) = \text{"31 bd. des Italiens"}$

$\nu(x_p) = \text{"47 42 60 33"}$

**Dies ist eine gültige Belegung, da**  $\text{Tupel}(\nu(x_{ti}), \text{"Bergman"}, \nu(x_{ac})) =$   
 $(\text{"Schreie und Flüstern"}, \text{"Bergman"}, \text{"Ullman"}) \in I(\text{Movies})$ , für unsere  
Beispiel Ausprägung  $I$ , analog für die beiden anderen Relationen bzw.  
Tupel.

## Weitere Beispiele

### Finde Name und Personalnummer aller C4 Professoren:

$$ans(x_{na}, x_{pn}) \leftarrow Professoren(x_{pn}, x_{na}, 'C4', x_{ra})$$

### Welche Vorlesungen bietet Prof. Sokrates an?

$$ans(x_{ti}) \leftarrow Professoren(x_{pn}, 'Sokrates', x_{rg}, x_{ra}), \\ Vorlesungen(x_{vn}, x_{ti}, x_{sw}), lesen(x_{pn}, x_{vn})$$

## Intensionale vs. Extensionale Relationen

Wir berechnen durch die Erstellung von Anfragen nicht nur Ergebnisse, sondern **definieren dabei implizit auch neue Relationen**, z.B. die neue Relation *C4Profes*, die alle Professoren mit Rang C4 enthält:

$$C4Profes(x_{pn}, x_{na}, x_{ra}) \leftarrow Professoren(x_{pn}, x_{na}, 'C4', x_{ra})$$

Man unterscheidet zwischen Relationen, die ursprünglich in der Datenbank vorhanden sind, den sogenannten **extensionalen** Relationen, und Relationen, die durch Regeln definiert werden, den sogenannten **intensionalen** Relationen.

## Auswertung (aka. Berechnung) der Anfrage

### Naive brute-force Methode zur Berechnung der Anfrageergebnisse:

- Betrachtung aller möglichen Belegung anhand der in den Relationen auftretenden Werte.
- Für jede dieser Belegungen schauen ob es eine gültige Belegung ist.

Diese Vorgehensweise ist natürlich sehr teuer. Z.B. für  $Movies(x_1, \text{"Bergman"}, x_2)$  machen nur Belegungen Sinn, die für dieses Tupel mit  $x_1$  und  $x_2$  aus  $Movies$  den Regisseur gleich "Bergman" haben. Idealerweise kann man mit sogenannten Indexen die Auswahl die Suche nach geeigneten Tupeln beschleunigen.

Dazu kommen wir später im **Abschnitt über Anfrageverarbeitung und Indexstrukturen**.



# Monotonie

- **Monotonie:** Eine Anfrage  $q$  über  $R$  ist monoton wenn für jede Ausprägungen (Instanzen)  $I, J$  über  $R$ , gilt: Falls  $I \subseteq J$  dann  $q(I) \subseteq q(J)$ .
- D.h. wenn neue Tupel zu den Relationen hinzukommen fallen keine Ergebnistupel weg.

**Konjunktive Anfragen sind monoton**

**Welche Anfragen sind somit nicht möglich?**

## Gleichheit bzw. Vergleichsoperatoren

$$\begin{aligned} ans(x_{th}, x_{ad}) \leftarrow & \text{Movies}(x_{ti}, x_d, x_{ac}), x_d = \text{'Bergman'}, \\ & \text{Pariscope}(x_{th}, x_{ti}, x_s), \\ & \text{Location}(x_{th}, x_{ad}, x_p) \end{aligned}$$

ist identisch zu

$$\begin{aligned} ans(x_{th}, x_{ad}) \leftarrow & \text{Movies}(x_{ti}, \text{'Bergman'}, x_{ac}), \\ & \text{Pariscope}(x_{th}, x_{ti}, x_s), \\ & \text{Location}(x_{th}, x_{ad}, x_p) \end{aligned}$$

Ebenfalls möglich Vergleichsoperatoren, z.B.

$$ans(x_{ma}) \leftarrow \text{Studenten}(x_{ma}, x_{na}, x_{se}), x_{se} \geq 12$$

# Anfrage-Programm

Wie erwähnt, kann konzeptionell eine Anfrage so aufgefasst werden, dass sie eine neue Relation erzeugt, die dann in nachfolgenden Anfragen benutzt werden kann.

Ein **konjunktives Anfrage-Programm** ist eine Sequenz von Regeln in der Form

$$\begin{aligned} S_1(u_1) &\leftarrow body_1 \\ S_2(u_2) &\leftarrow body_2 \\ &\dots \\ S_m(u_m) &\leftarrow body_m \end{aligned}$$

## Beispiel

$$\begin{aligned}
 S_1(x, z) &\leftarrow Q(x, y), R(y, z, w) \\
 S_2(x, y, z) &\leftarrow S_1(x, w), R(w, y, v), S_1(v, z) \\
 S_3(x, z) &\leftarrow S_2(x, u, v), Q(v, z)
 \end{aligned}$$

$Q$	
1	2
2	1
2	2

$R$		
1	1	1
2	3	1
3	1	2
4	4	1

$S_1$	
1	3
2	1
2	3

$S_2$		
1	1	1
1	1	3
2	1	1
2	1	3

$S_3$	
1	2
2	2

# Beispiel

$$\begin{aligned}S_1(x, z) &\leftarrow Q(x, y), R(y, z, w) \\S_2(x, y, z) &\leftarrow S_1(x, w), R(w, y, v), S_1(v, z) \\S_3(x, z) &\leftarrow S_2(x, u, v), Q(v, z)\end{aligned}$$

**Wir können für die ersten beiden Zeilen auch schreiben als**

$$\begin{aligned}S_2(x, y, z) &\leftarrow Q(x_1, y_1), R(y_1, z_1, w_1), x = x_1, w = z_1, \\&R(w, y, v), Q(x_2, y_2), R(y_2, z_2, w_2), v = x_2, z = z_2\end{aligned}$$

bzw. ohne Gleichheits-Operator

$$\begin{aligned}S_2(x, y, z) &\leftarrow Q(x, y_1), R(y_1, w, w_1), \\&R(w, y, v), Q(v, y_2), R(y_2, z, w_2)\end{aligned}$$