

Datenbanken I (T2INF2004)

Foliensatz 3: Datenbankentwurf II

Uli Seelbach, DHBW Mannheim, 2023

Foliensatz freundlicherweise zur
Verfügung gestellt von Mirko Schick



Datenbankentwurf

konzeptueller Entwurf ist fertig, und nun?

- Überführung des konzeptionellen (ER-)Modells in das relationale Datenmodell
- Dazu werden
 - Transformationsregeln und
 - Normalformenangewendet
- Modellierungstools sind dabei hilfreich
 - CASE-Tools (Computer-aided software engineering) gehen noch einen Schritt weiter und können unterstützend hinzugenommen werden



Relationales Modell

Grundlagen

- Beruht ausschließlich auf Relationen, deren Schema durch einen Namen und seine Attribute festgelegt wird
- Relationen mit n Attributen sind Mengen von n -Tupeln
- Ein n -Tupel ist eine Liste von n Objekten eines jeweils festgelegten Wertebereichs (Vgl. Record)
- Jede Relation besitzt einen eindeutigen Primärschlüssel
 - Manche Autoren fordern, dass sich dieser nicht ändern darf – dafür gibt es in der Theorie keinen Grund und in der Praxis keinen überzeugenden Grund
- Tupel in einer Relation sind paarweise verschieden und nicht geordnet



Relationales Modell

Grundlagen

- Darstellung einer Relation meist tabellenförmig, Darstellung des Relationenschemas auch einzeilig möglich
- Tupel einer Relation können auf Tupel anderer Relationen „verweisen“
- Der Datentyp der Attribute ist im grundlegenden relationalen Modell für uns erst einmal auf die bekannten elementaren Datentypen beschränkt. Wie wir diese genau nennen, ist theoretisch irrelevant, es gibt jedoch DBMS-spezifische Eigenheiten in der Benennung. Beispiele für primitive Datentypen sind:
 - Ganze Zahlen (INT)
 - Zeichen oder Zeichenketten: CHAR
 - Gleitpunktzahlen: DOUBLE, DECIMAL

Relationales Modell

Grundlagen

Beispielrelation mit Begriffen aus der Welt des „Relationenmodells“ und der Welt der Datenbanksysteme (in Klammern):

Name des Relationenschemas		Attribut (Spalte)		
↓		↓		
Markt:		<u>Bezeichnung</u>	<u>Standort</u>	Kategorie
		Internationaler Töpfermarkt	Krefeld	Töpfermarkt
		Töpfermarkt Sommerhausen	Sommerhausen	Töpfermarkt
		Internationaler Töpfermarkt	Hanau	Töpfermarkt

Relation (Tabelleninhalt)

Relationenschema

Tupel (Zeile)



Relationales Modell

Integritätsbestimmungen

- Innerhalb einer Relation:
 - Darauf achten, dass Datentypen eingehalten werden
 - Darauf achten, dass die als Schlüssel definierten Attributmengen auch wirklich Eindeutigkeit gewährleisten
 - Verstoß:

Telefonbuch		
Name	Straße	<u>Telefon#</u>
Mickey Mouse	Main Street	4711
Minnie Mouse	Broadway	4711
Donald Duck	Broadway	95672
...
Horst	Broadway	xxx
Hampelmann	Broadway	545456
Hampelfrau	Broadway	



Relationales Modell

Integritätsbestimmungen

- **Zwischen Relationen: Fremdschlüsselabhängigkeit**
 - Eine Attributmenge, die in einer Relation als Primärschlüssel definiert wurde, kann in einer assoziierten Relation als Fremdschlüssel definiert werden. Dies bedeutet, dass sich ein Tupel der assoziierten Relation auf ein Tupel der Relation mit dem Primärschlüssel bezieht
 - Tupel in der assoziierten Relation dürfen nur solche Werte als Fremdschlüsselattribut haben, die auch in der referenzierten Relation vorhanden sind / ggf. darf ein Fremdschlüssel aber auch leer gelassen werden
 - Die assoziierte und referenzierte Relation müssen nicht verschieden sein, d.h. ein Fremdschlüssel der Relation R kann durchaus auf den Primärschlüssel der Relation R verweisen
 - Hört sich komplizierter an, als es ist:

Relationales Modell

Fremdschlüssel – Warum eigentlich? Folgendes Problem:

Schüler			
<u>Vorname</u>	<u>Nachname</u>	Klasse	Klassenlehrer
Anja	Bla	3a	Herr A
Horst	Bli	3a	Herr A
Michelle	Blubb	3a	Herr A
Projr	Fla	3b	Frau B
...



Schüler			
<u>Vorname</u>	<u>Nachname</u>	Klasse	Klassenlehrer
Anja	Bla	3a	Herr A C
Horst	Bli Fli	3a	Herr A C
Michelle	Blubb	3a	Herr A C
Projr	Fla	3b	Frau B
...

Relationales Modell

Fremdschlüssel – Warum eigentlich? Lösung:

Schüler			
<u>Vorname</u>	<u>Nachname</u>	Klasse	Klassenlehrer
Anja	Bla	3a	Herr A
Horst	Bli	3a	Herr A
Michelle	Blubb	3a	Herr A
Projr	Fla	3b	Frau B
...



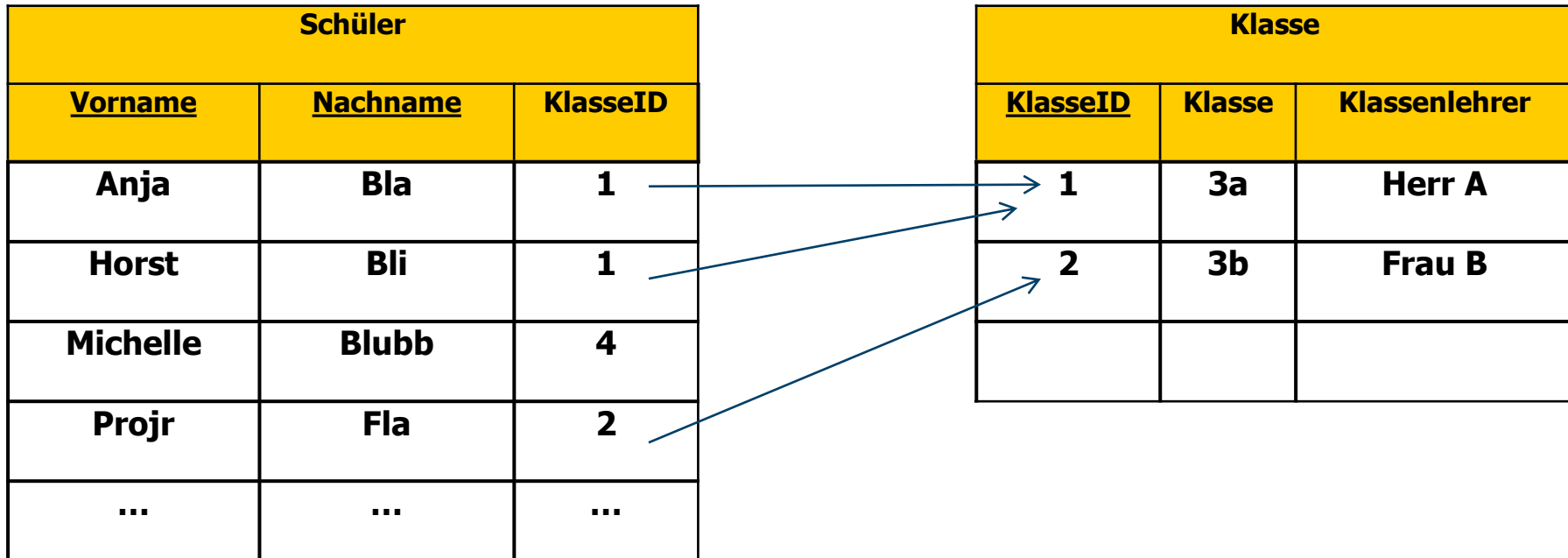
Schüler		
<u>Vorname</u>	<u>Nachname</u>	KlasseID
Anja	Bla	1
Horst	Bli	1
Michelle	Blubb	1
Projr	Fla	2
...

Klasse		
<u>KlasseID</u>	Klasse	Klassenlehrer
1	3a	Herr A
2	3b	Frau B
...



Relationales Modell

Fremdschlüssel – Warum eigentlich?



- KlasseID in der Relation Schüler ist ein Fremdschlüssel. Er bezieht sich auf den Primärschlüssel der Relation Klasse
 - Schüler.KlasseID darf also nur die Werte von Klasse.KlasseID annehmen
- ¹⁰ – Ggf. darf der Wert auch leer sein, wenn die Klasse noch nicht feststeht

Relationales Modell

Fremdschlüssel

- Was hat uns das gebracht?
 - Datenänderungen sind einfacher
 - Weniger Redundanz
 - Automatische Integritätsprüfung möglich



Ist das dargestellte Schema sinnvoll?

Schüler		
<u>Vorname</u>	<u>Nachname</u>	KlasseID
Anja	Bla	1
Horst	Bli	1
Michelle	Blubb	1
Projr	Fla	2
...

Klasse		
<u>KlasseID</u>	Klasse	Klassenlehrer
1	3a	Herr A C
2	3b	Frau B
...

Relationales Modell

Schema-Schreibweise

Klasse: {[KlasseID:integer, Klasse: string, Klassenlehrer: string]}



Schüler: {[Vorname: string, Nachname: string, KlasseID: integer]}

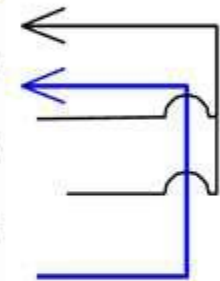
Schüler		
<u>Vorname</u>	<u>Nachname</u>	KlasseID
Anja	Bla	1
Horst	Bli	1
Michelle	Blubb	1
Projr	Fla	2
...

Klasse		
<u>KlasseID</u>	Klasse	Klassenlehrer
1	3a	Herr A
2	3b	Frau B
...

Relationales Modell

Fremdschlüssel auf Primärschlüssel in gleicher Relation

Arbeitnehmer		
<u>ID</u>	Name	VorgesetzterID
1	Herr A	NULL
2	Herr B	1
3	Frau A	1
4	Frau B	2
...



Arbeitnehmer: {[ID:integer, Name: string, VorgesetzterID: integer]}

Relationales Modell

Arten von Schlüsseln

- Nur einer der Schlüsselkandidaten kann als Primärschlüssel bezeichnet werden, die Auswahl fällt manchmal schwer. Es gibt:

Fachliche / natürliche / sprechende Schlüssel

- Schlüssel, die auf natürliche Weise vorhanden sind und nicht extra generiert wurden. Bsp.: Kontonummer. Falls diese nicht mit Bedacht gewählt werden, können Systemänderungen dazu führen, dass der verwendete Datentyp hinfällig wird (Kontonummer wurde nun durch IBAN inkl. Buchstaben abgelöst)

Technische / stellvertretende / Surrogatschlüssel

- Ein Attribut wird extra als Schlüssel angelegt.
- Sehr breite zusammengesetzte, natürlich vorkommende Schlüssel können so kompakt zusammengefasst werden (Fachwort: Surrogatschlüssel)



Relationales Modell

Arten von Schlüsseln

Rechnung				
KundeID	Kundenrechnung	Erfassungszeitpunkt	S1	S2
1	1	26.01.2014 20:53	1_1	1
1	2	26.01.2014 21:21	1_2	2
2	1	26.01.2014 21:50	2_1	3
4	1	28.01.2014 04:05	4_1	4
...

- Guter fachlicher Schlüssel: KundeID, Kundenrechnung
- Guter technischer Schlüssel: S2 (Fortlaufende Rechnungsnummer)
- Nicht guter fachlicher Schlüssel: Erfassungszeitpunkt
- ¹⁵ • Mäßig guter technischer Schlüssel: S1
– concat(KundeID, “_“, Kundenrechnung)



Fallen Ihnen weitere Anwendungsbeispiele ein, in denen ein fachlicher Primärschlüssel ggf. ersetzt werden sollte?



Relationales Modell

Überführung vom ER-Modell

- Zukünftig ein paar Abkürzungen:
 - Primärschlüssel = primary key = PK
 - Fremdschlüssel = foreign key = FK
- Dazu werden im ersten Schritt folgende Transformationsregeln angewendet:
 - Jeder Entity-Typ wird zu einem Relationenschema (auch schwache Entity-Typen)
 - Mehrwertige Attribute werden in ein Relationenschema überführt
 - Jeder Relationstyp wird zu einem Relationenschema
 - Relationen mit gleichem PK werden, falls kein Grund dagegen spricht, zusammengefasst
 - Generalisierungen werden modelliert



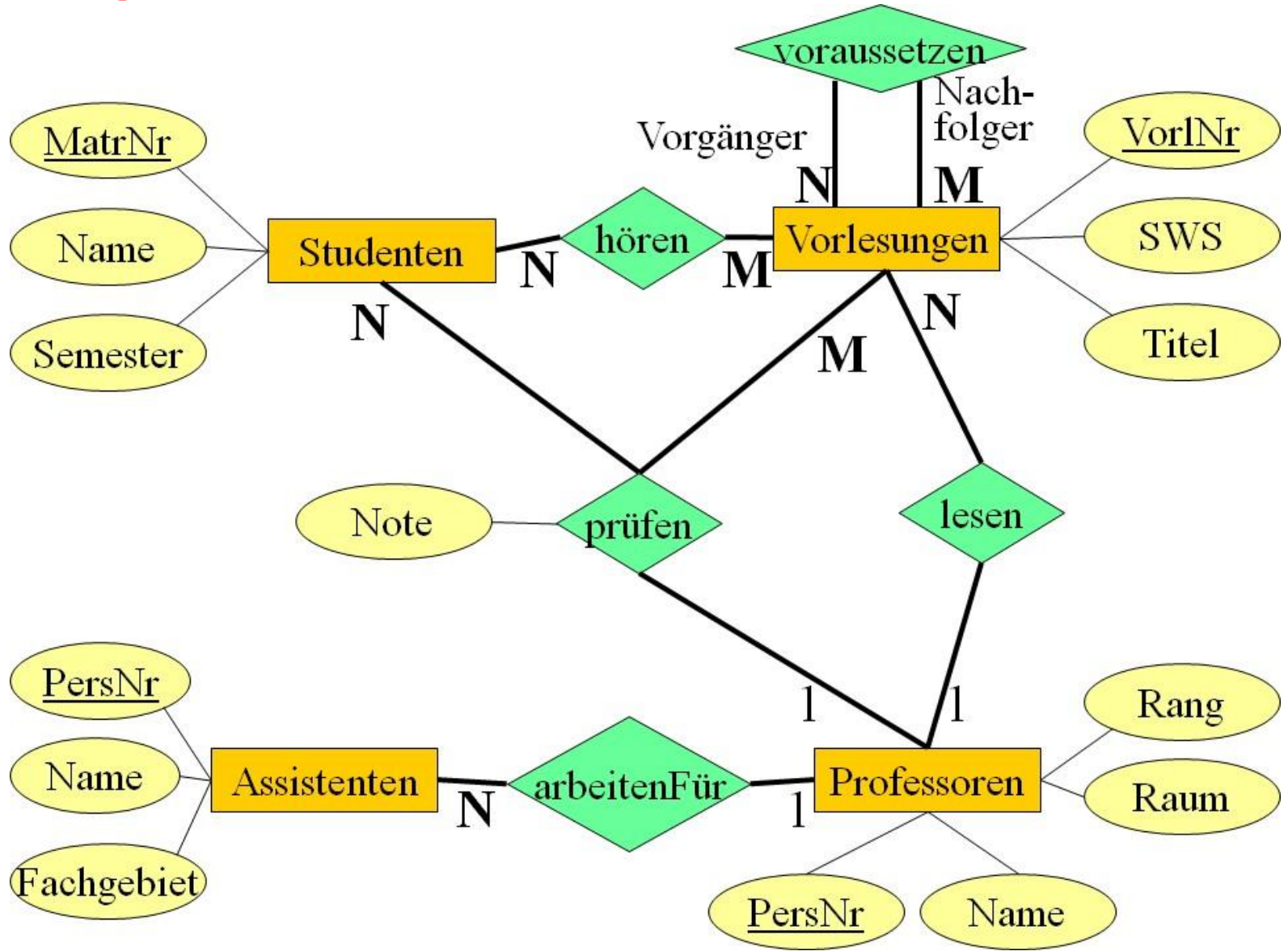
Relationales Modell

Überführung vom ER-Modell – Schritt 1 (Entity-Typen)

- Dazu werden im ersten Schritt folgende Transformationsregeln angewendet:
 - Jeder Entity-Typ wird zu einem Relationenschema (auch schwache Entity-Typen)
 - Mehrwertige Attribute werden in ein Relationenschema überführt
 - Jeder Relationshiptyp wird zu einem Relationenschema
 - Relationen mit gleichem PK werden, falls kein Grund dagegen spricht, zusammengefasst
 - Generalisierungen werden modelliert

Relationales Modell

Überführung vom ER-Modell – Schritt 1 (Entity-Typen)





Relationales Modell

Überführung vom ER-Modell – Schritt 1 (Entity-Typen)

- Entity-Typen werden in Relationen umgewandelt
- die Attribute der Entity-Typen werden als solche übernommen, Primärschlüssel als solche gekennzeichnet

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*]}

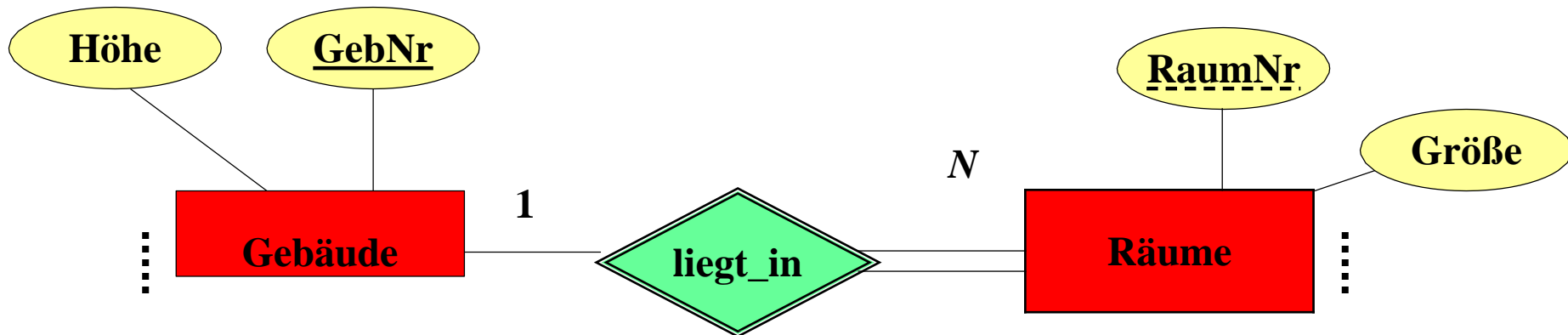
Relationales Modell

Überführung vom ER-Modell – Schritt 1 (Entity-Typen)

- Das Vorgehen ist bei schwachen Entity-Typen analog, nur dass der Primärschlüssel implizit aus dem identifizierenden Beziehungstypen ermittelt werden muss:

Gebäude: {[GebNr:integer, Höhe: Float]}

Räume: {[GebNr:integer, RaumNr:integer, Größe: Float]}





Relationales Modell

Überführung vom ER-Modell

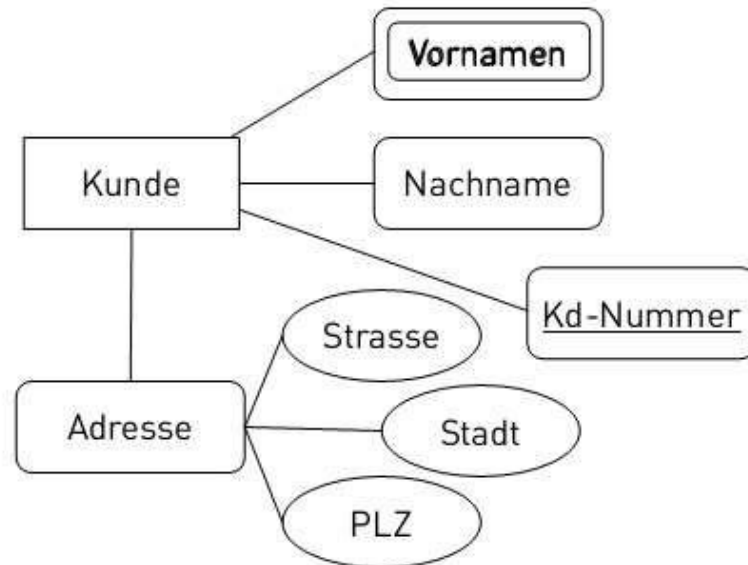
- Dazu werden im ersten Schritt folgende Transformationsregeln angewendet:
 - Jeder Entity-Typ wird zu einem Relationenschema (auch schwache Entity-Typen)
 - Mehrwertige Attribute werden in ein Relationenschema überführt
 - Jeder Relationshiptyp wird zu einem Relationenschema
 - Relationen mit gleichem PK werden, falls kein Grund dagegen spricht, zusammengefasst
 - Generalisierungen werden modelliert

Relationales Modell

Überführung vom ER-Modell – Schritt 2 (Mehrwertige Attr.)

Ein ungültiges Relationenschema, da die Länge der Tupel eindeutig sein muss („Array“ *Vornamen*) und keine komplexen Datentypen umfassen soll* („Record“ *Adresse*):

Entity-Typ (Beispiel):



Kunde:	<u>Kd-Nummer</u>	Nachname	Adresse			Vornamen			
			Strasse	PLZ	Stadt	Vorname_1	Vorname_2	...	Vorname_n



Relationales Modell

Überführung vom ER-Modell – Schritt 2 (Mehrwertige Attr.)

- Simple Records können sofort aufgelöst und in die Ursprungsrelation integriert werden
- Mehrwertige Attribute der Ursprungsrelation R müssen in eine gesonderte Relation S ausgelagert werden, welche die folgenden Attribute erhält:
 - Alle PK-Attribute von R (diese werden dann als FK deklariert und zeigen auf das assoziierte Tupel in R)
 - Das mehrwertige Attribut selbst
- Als PK von S wird dann die Gesamtmenge der Attribute von S definiert



Ist die hier aufgezeigte Lösung frei von Informationsverlust und in der Intention des Anforderers?

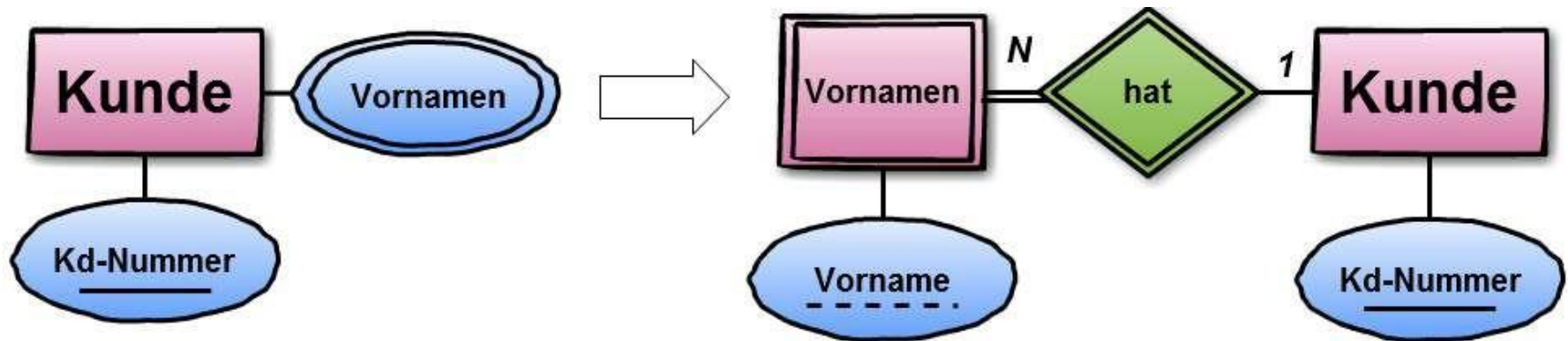
Vornamen: {[Kd-Nummer: integer, Vorname: string]}



Kunde: {[Kd-Nummer: integer, Nachname: string, Straße: String, Stadt: String, PLZ: String]}

Relationales Modell

Überführung vom ER-Modell – Schritt 2 (Mehrwertige Attr.)



- Etwas einfacher erklärt... was haben wir gemacht?
 - ➔ Wir haben aus einem mehrwertigen Attribut einen schwachen Entity-Typen gemacht und verfahren mit diesem wie im Schritt 1
 - ➔ Nun haben wir schon einen Vorgeschmack darauf bekommen, was mit Relationship-Typen passiert: dort spielen FKs eine Rolle

Vornamen: {[Kd-Nummer: integer, Vorname: string]}



Kunde: {[Kd-Nummer: integer, Nachname: string, Straße: String, Stadt: String, PLZ: String]}



Relationales Modell

Überführung vom ER-Modell

- Dazu werden im ersten Schritt folgende Transformationsregeln angewendet:
 - Jeder Entity-Typ wird zu einem Relationenschema (auch schwache Entity-Typen)
 - Mehrwertige Attribute werden in ein Relationenschema überführt
 - **Jeder Relationstyp wird zu einem Relationenschema**
 - Relationen mit gleichem PK werden, falls kein Grund dagegen spricht, zusammengefasst
 - Generalisierungen werden modelliert



Relationales Modell

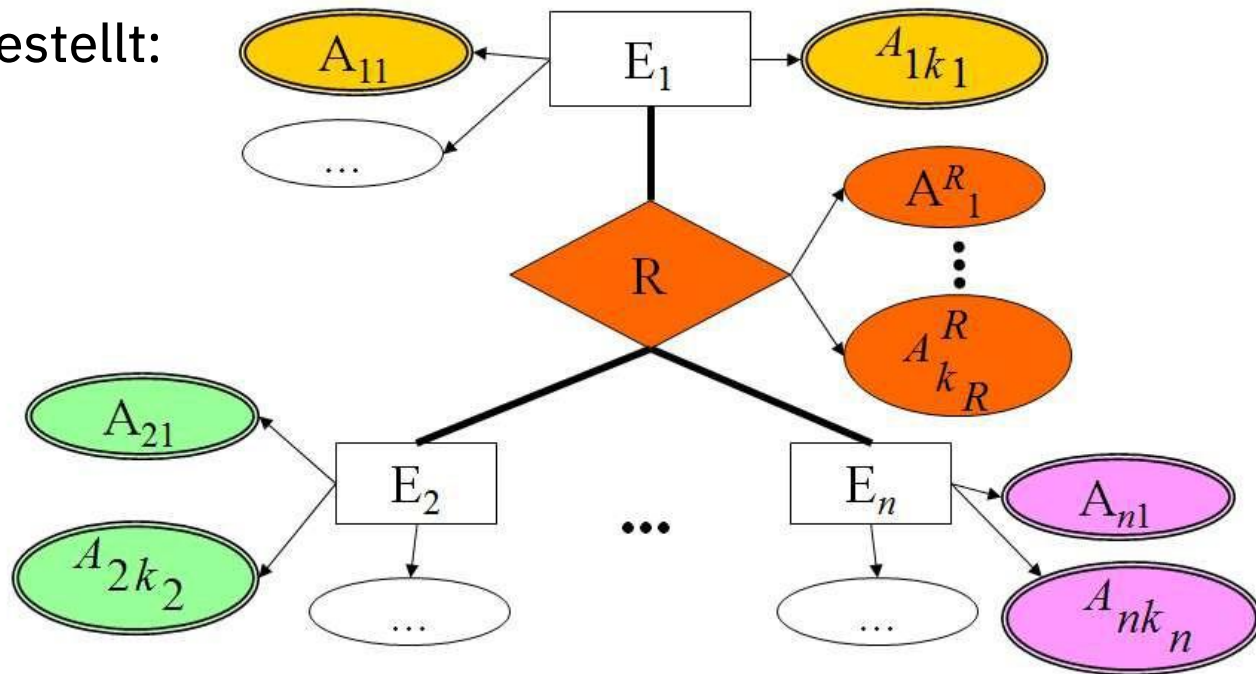
Überführung vom ER-Modell – Schritt 3 (Relationship-Typen)

- Aus der Sicht eines „einfachen“ Algorithmus wird jeder Beziehungstyp b zu einer Relation B
 - 1:1-Beziehungen, 1:N-Beziehungen, N:M-Beziehungen
- Die Relation B setzt sich zusammen aus den PKs aller an dem Beziehungstypen beteiligten, auf den Entity-Typen basierenden, Relationen und den Attributen von b . Jeweilige PKs werden zu FKs
 - Im Falle einer 1:N-Beziehung reicht es aus, dass der PK von B dem FK auf der Seite von N entspricht
 - Im Falle einer 1:1-Beziehung wird der PK definiert aus:
 - dem FK der Relation mit totaler Teilnahme, wenn Kardinalitätsstärke unterschiedlich
 - Dem FK der Relation kleinerer Kardinalität
 - Andernfalls entspricht der PK von B der Kombination aller FKs in B
- Aggregationen behandeln wir wie 1:N-Beziehungen

Relationales Modell

Überführung vom ER-Modell – Schritt 3 (Relationship-Typen)

Grafisch dargestellt:

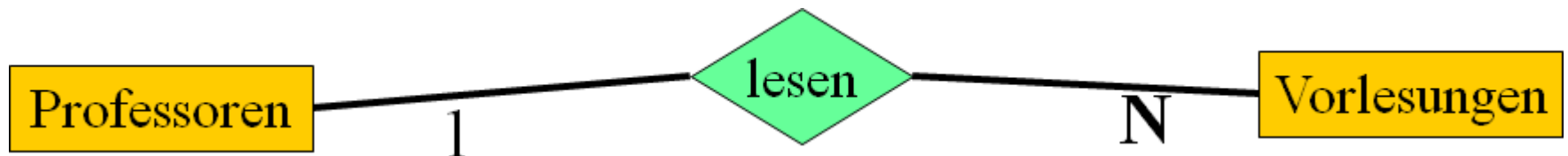


$$R: \left\{ \underbrace{[A_{11}, \dots, A_{1k_1}]}_{\text{Schlüssel von } E_1}, \underbrace{[A_{21}, \dots, A_{2k_2}]}_{\text{Schlüssel von } E_2}, \dots, \underbrace{[A_{n1}, \dots, A_{nk_n}]}_{\text{Schlüssel von } E_n}, \underbrace{[A_1^R, \dots, A_{k_R}^R]}_{\text{Attribute von } R} \right\}$$

Relationales Modell

Überführung vom ER-Modell – Schritt 3 (Relationship-Typen)

- 1:N-Relationen
 - Im Falle einer 1:N-Beziehung reicht es aus, dass der PK von B dem FK auf der Seite von N entspricht
 - Warum? Zu jeder Vorlesung kann es maximal einen Prof geben, der dann in der Relation „lesen“ stehen wird, also sind Vorlesungen einmalig



Vorlesungen: {[VorlNr:integer, Titel: string, SWS: integer]}

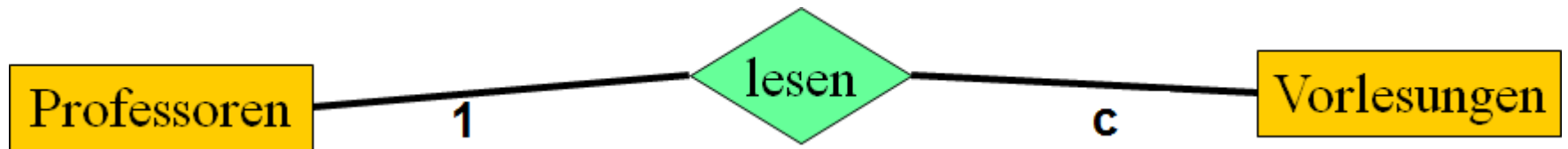
lesen: {[PersNr:integer, VorlNr:integer]}

Professoren : {[PersNr:integer, Name:String, Rang:String, Raum:String]}

Relationales Modell

Überführung vom ER-Modell – Schritt 3 (Relationship-Typen)

- 1:1-Relationen
 - Hier muss (leider) vorab definiert werden, ob die jeweiligen einsen „must“ oder „can“ sind
 - Ist das nicht bekannt, müssen wir implizit davon ausgehen, dass eine c:c-Beziehung vorliegt (selten!!!) oder den gesunden Menschenverstand benutzen und das konzeptuelle Schema anpassen
 - 1:c → PK von Relation „lesen“ ist FK auf Relation mit geringerer Kardinalität



Vorlesungen:

{[VorlNr:integer], Titel:string, SWS:integer}

lesen:

{[PersNr:integer], [VorlNr:integer]}

Professoren :

{[PersNr:integer], Name:String, Rang:String, Raum:String}

• Vorweg zu Schritt 4:
Relationen mit
gleichen PKs werden
zusammengefasst

• PersNr wäre
theoretisch ein
Schlüsselkandidat

Relationales Modell

Überführung vom ER-Modell – Schritt 3 (Relationship-Typen)

- 1:1-Relationen

- Hier muss (leider) vorab definiert werden, ob die jeweiligen einsen „must“ oder „can“ sind
- Ist das nicht bekannt, müssen wir implizit davon ausgehen, dass eine c:c-Beziehung vorliegt (selten!!!) oder den gesunden Menschenverstand benutzen und das konzeptuelle Schema anpassen
- c:c → Relation „nutzt“ bekommt bel. PK! Alternativ (und theretisch sauberer) gehen wir hier vor wie bei 1:c (PK = FK auf Relation mit geringerer Kardinalität)



Büro:

{BüroNr:integer, Größe: Float}

nutzt:

{PersNr:integer, BüroNr:integer}

Professoren :

{PersNr:integer, Name:String, Rang:String, Raum:String}

- da wir nicht wissen, welchen der beiden wir nehmen sollen, definieren wir irgendeinen PK oder definieren wie hier, dass PK auf die Relation mit geringerer Kardinalität zeigt

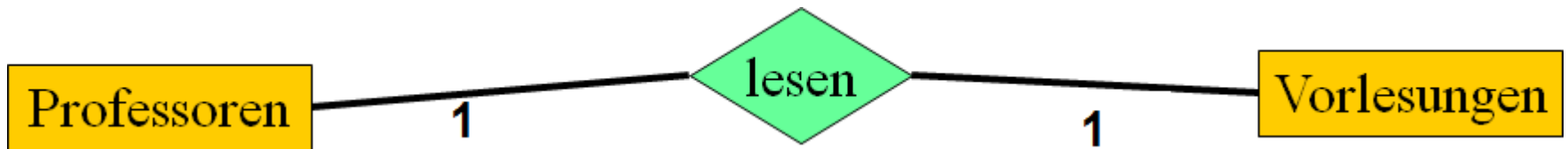
- PersNr und BüroNr wären theoretisch Schlüsselkandidaten

Relationales Modell

Überführung vom ER-Modell – Schritt 3 (Relationship-Typen)

- 1:1-Relationen

- Hier muss (leider) vorab definiert werden, ob die jeweiligen einsen „must“ oder „can“ sind
- Ist das nicht bekannt, müssen wir implizit davon ausgehen, dass eine c:c-Beziehung vorliegt (selten!!!) oder den gesunden Menschenverstand benutzen und das konzeptuelle Schema anpassen
- 1:1 → PK der Relation „lesen“ = FK auf eine beliebige Relation



Vorlesungen: {[VorlNr:integer, Titel: string, SWS: integer]}

lesen: {[PersNr:integer, VorlNr:integer]}

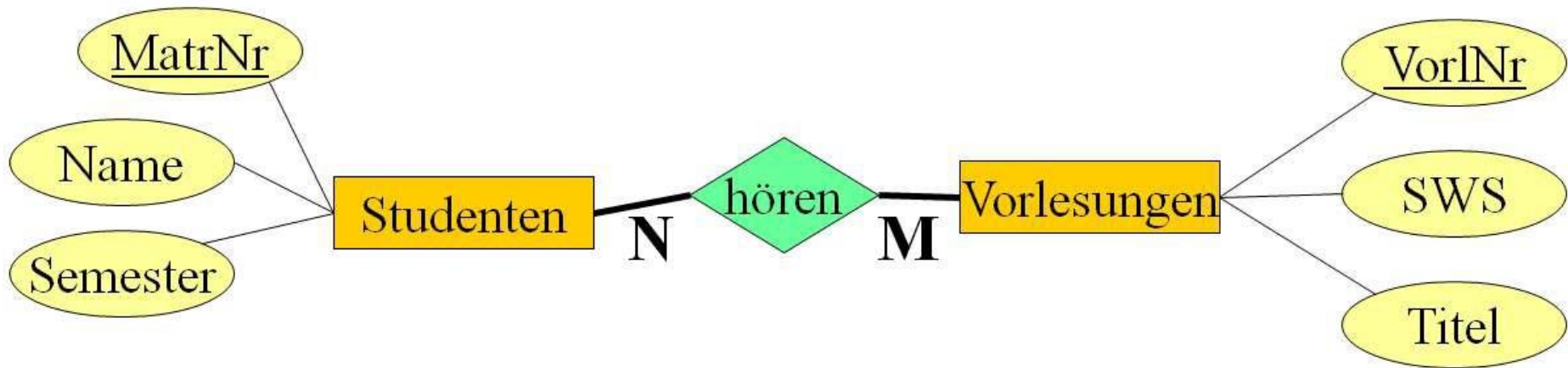
Professoren : {[PersNr:integer, Name:String, Rang:String, Raum:String]}

- Beides wären hier mögliche Schlüsselkandidaten

Relationales Modell

Überführung vom ER-Modell – Schritt 3 (Relationship-Types)

- N:M-Relationen
 - [...] andernfalls entspricht der PK von B der Kombination aller FKs in B



Studenten: {[MatrNr:integer, Name: string, Semester: integer]}

hören: {[MatrNr:integer, VorlNr:integer]}

Vorlesungen: {[VorlNr:integer, Titel: string, SWS: integer]}

Relationales Modell

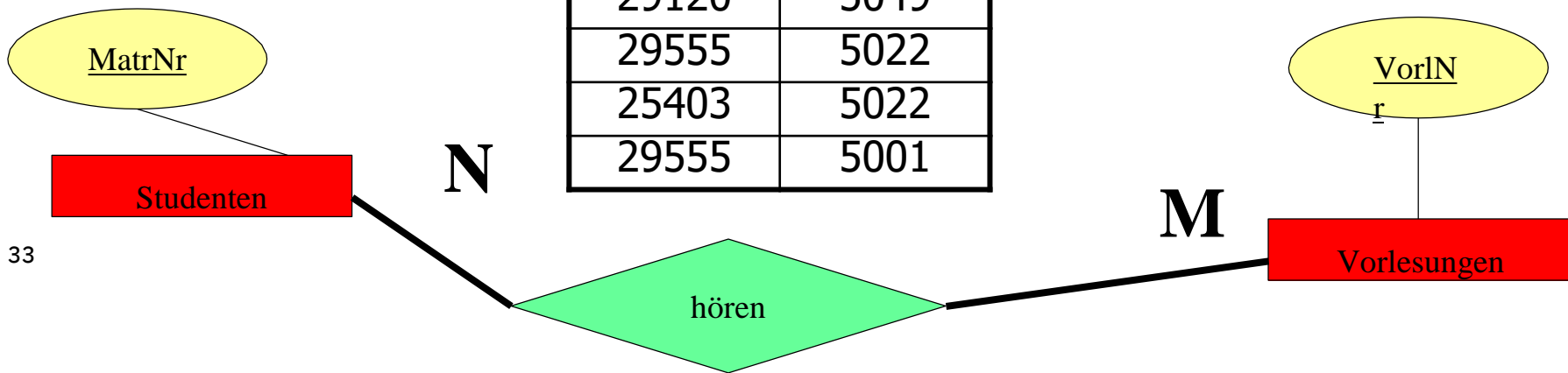
Überführung vom ER-Modell – Schritt 3 (Relationship-Typen)

- N:M-Relationen

Studenten	
<u>MatrNr</u>	...
26120	...
27550	...
...	...

hören	
<u>MatrNr</u>	<u>VorlNr</u>
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	
<u>VorlNr</u>	...
5001	...
4052	...
...	...





Relationales Modell

Überführung vom ER-Modell

- Dazu werden im ersten Schritt folgende Transformationsregeln angewendet:
 - Jeder Entity-Typ wird zu einem Relationenschema (auch schwache Entity-Typen)
 - Mehrwertige Attribute werden in ein Relationenschema überführt
 - Jeder Relationshiptyp wird zu einem Relationenschema
 - Relationen mit gleichem PK werden, falls kein Grund dagegen spricht, zusammengefasst
 - Generalisierungen werden modelliert



Relationales Modell

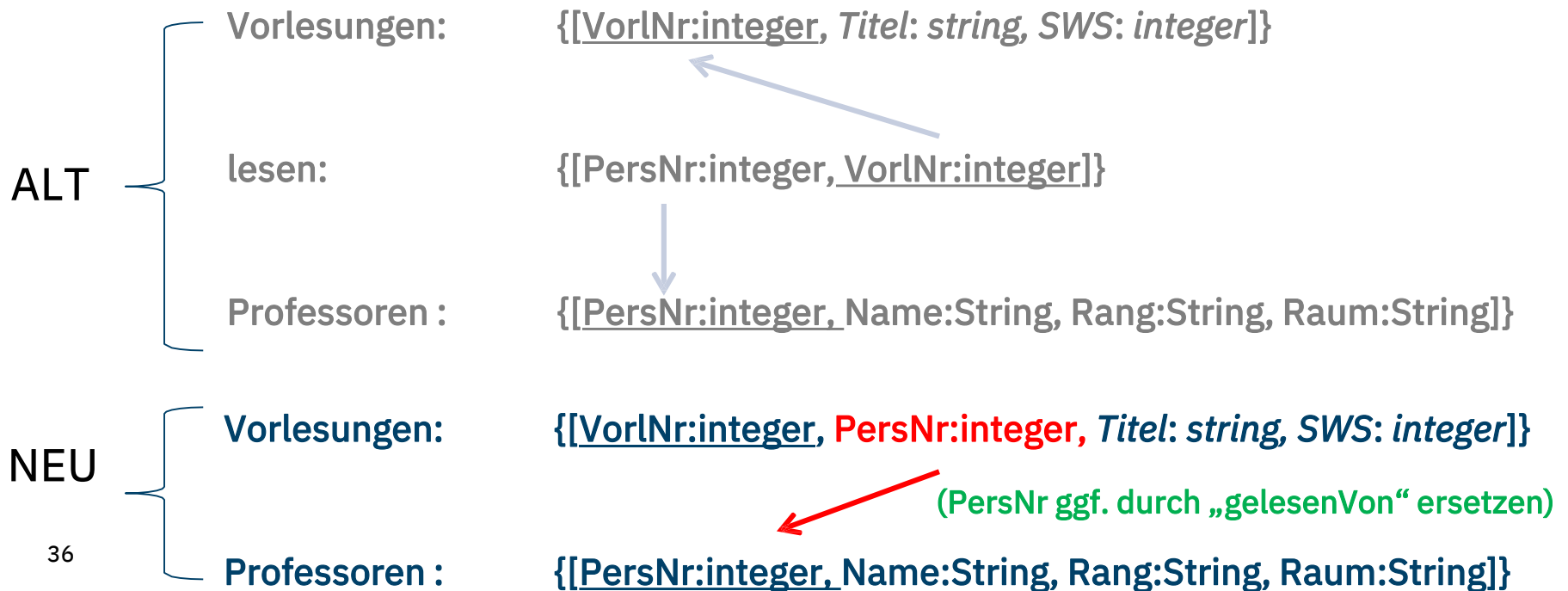
Überführung vom ER-Modell – Schritt 4 (Zusammenfassen)

- Relationen mit gleichem PK kann man unter bestimmten Voraussetzungen zusammenfassen, wenn kein bestimmter Grund dagegen spricht
- Gründe gegen ein Zusammenfassen könnten sein
 - Relationen klein halten (NULL-Werte vermeiden, Bsp. kommt gleich)
 - Relationen dürfen aus welchem Grund auch immer nicht geändert werden
 - Einfacher Datenschutz auf Objektebene

Relationales Modell

Überführung vom ER-Modell – Schritt 4 (Zusammenfassen)

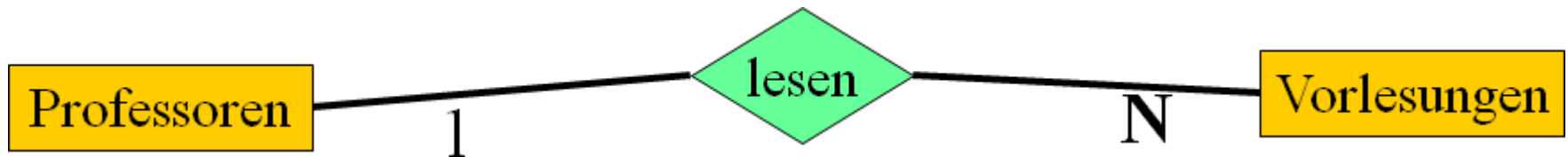
- Wie sieht so ein Zusammenfassen aus bei 1:N?



Relationales Modell

Überführung vom ER-Modell – Schritt 4 (Zusammenfassen)

- Ausprägungen von Professoren und Vorlesungen



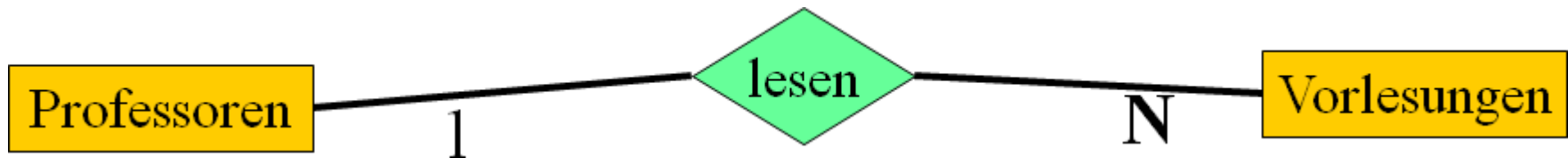
Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

Vorlesungen			
VorlNr	Titel	SWS	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

Relationales Modell

Überführung vom ER-Modell – Schritt 4 (Zusammenfassen)

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



Relationales Modell

Überführung vom ER-Modell – Schritt 4 (Zusammenfassen)

Vorsicht, so geht es nicht! Folge sind Anomalien.

- Lösch-Anomalie: Was passiert wenn „Glaube und Wissen“ wegfällt
- Update-Anomalie: Was passiert wenn Sokrates umzieht
- Einfügeanomalie: Curie ist neu und liest noch keine Vorlesungen

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

Relationales Modell

Überführung vom ER-Modell – Schritt 4 (Zusammenfassen)

- Wie sieht so ein Zusammenfassen aus bei 1:N?
 - Mit der modifizierten Chen-Notation können wir noch zusätzliche Integritätsbestimmungen definieren:



c	Vorlesungen:	{ <u>VorlNr:integer</u> , PersNr:integer , Titel: string, SWS: integer]}
	Professoren :	{ <u>PersNr:integer</u> , Name:String, Rang:String, Raum:String}
1	Vorlesungen:	{ <u>VorlNr:integer</u> , PersNr:integer NOT NULL , Titel: string, SWS: integer]}
	Professoren :	{ <u>PersNr:integer</u> , Name:String, Rang:String, Raum:String}

Relationales Modell

Überführung vom ER-Modell – Schritt 4 (Zusammenfassen)

- Beachten Sie: bei c:m-Beziehungen können NULL-Werte auftreten
- Bedenken Sie folgendes Beispiel mit einer 1:N-Relation:



- Aus „habilitiert an“ würden wir eine eigene Relation erstellen mit dem PK „Person.PersonId“
- Im Folgeschritt würden wir die Relationen „habilitiert an“ und „Person“ zusammenfassen wegen gleichem PK
- Das führt zu vielen NULL-Werten in „Person.habilitiertAnUniId“
- Hier besteht also aufgrund der sehr hohen Selektivität* ein besonderes Interesse, die beiden Relationen nicht zusammenzufassen!



Relationales Modell

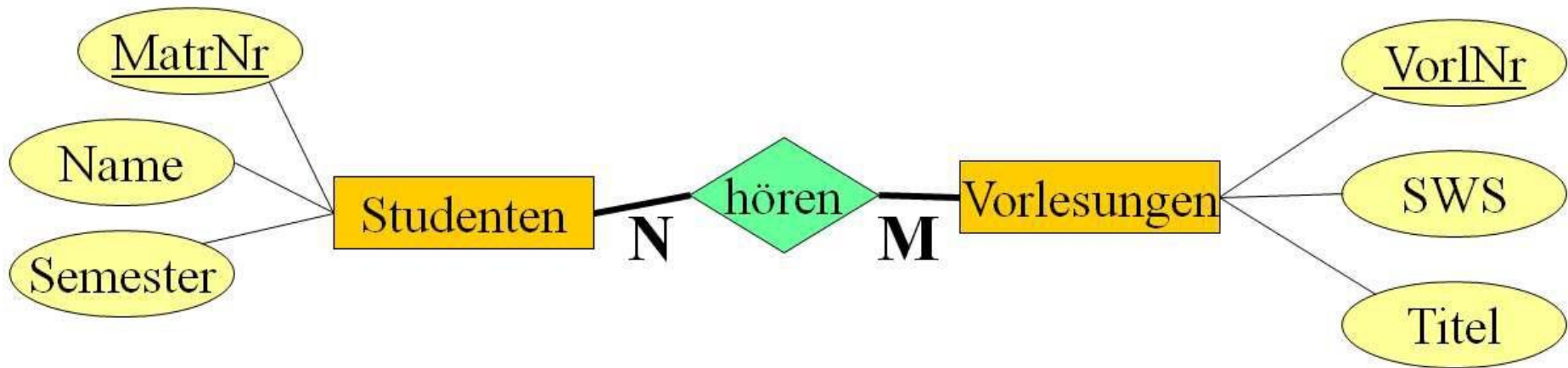
Überführung vom ER-Modell – Schritt 4 (Zusammenfassen)

- Wie sieht so ein Zusammenfassen aus bei 1:1?
 - Bei 1:c-Beziehungen, also einer totalen Teilnahme (bzw. in UML: Kompositionen) kein Problem
 - Bei (in Bezug auf die modifizierte Chen-Notation) 1:1-Beziehungen wird die Trennung der Relationen einen Grund gehabt haben. Theoretisch können beide Relationen komplett zusammengefasst werden, praktisch muss man hinterfragen, wie es erst zu 1:1 kommen konnte
 - Bei (in Bezug auf die modifizierte Chen-Notation) c:c-Beziehungen wird ein Zusammenfassen nicht möglich sein, da nicht sicher ist, welche Relation die größere Kardinalität aufweisen wird – darum haben wir auch im Schritt 3 definiert, dass in solchen Fällen ein PK am besten erst gar nicht automatisch gebildet wird, sondern manuell
 - Achtung: wenn es mehrere 1:c-Beziehungen von einer Relation aus gibt, spricht ggf. ein Grund gegen das Zusammenfassen wegen unnötiger „unbekannter“ Attributwerte (→ siehe Schritt 5)

Relationales Modell

Überführung vom ER-Modell – Schritt 4 (Zusammenfassen)

- Wie sieht so ein Zusammenfassen aus bei N:M?
 - Das geht nicht, da es hier keine „gleichen“ PKs gibt



Studenten: {[MatrNr:integer, Name: string, Semester: integer]}

hören: {[MatrNr:integer, VorlNr:integer]}

Vorlesungen: {[VorlNr:integer, Titel: string, SWS: integer]}



Relationales Modell

Überführung vom ER-Modell

- Dazu werden im ersten Schritt folgende Transformationsregeln angewendet:
 - Jeder Entity-Typ wird zu einem Relationenschema (auch schwache Entity-Typen)
 - Mehrwertige Attribute werden in ein Relationenschema überführt
 - Jeder Relationstyp wird zu einem Relationenschema
 - Relationen mit gleichem PK werden, falls kein Grund dagegen spricht, zusammengefasst
 - Generalisierungen werden modelliert



Relationales Modell

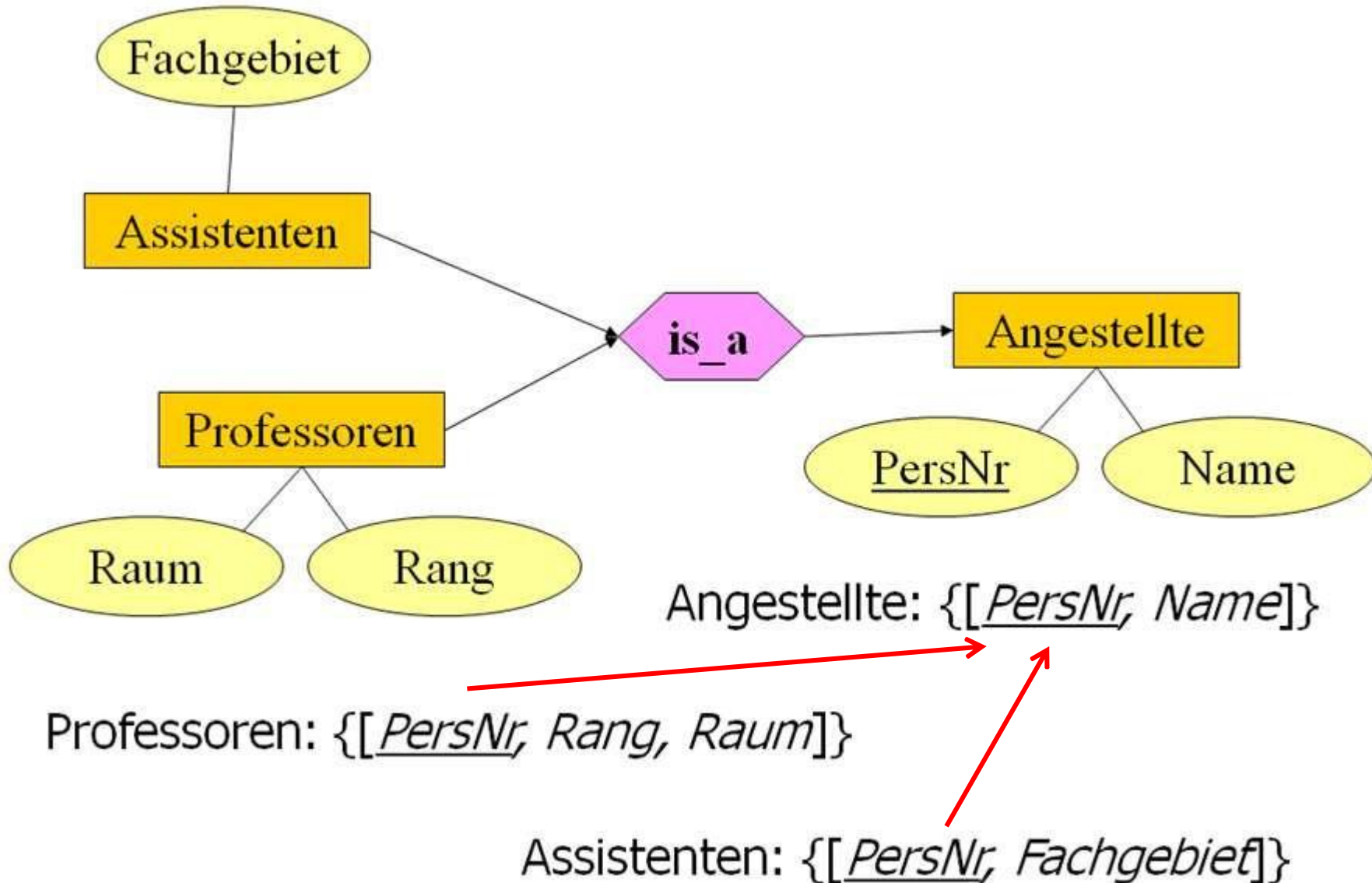
Überführung vom ER-Modell – Schritt 5 (Generalisierungen)

- Generalisierung/Spezialisierung waren im ursprünglichen ER-Modell von Chen nicht enthalten, wurden aber später eingeführt.
 - Darum gibt es auch keine einheitliche Schreibweise in der „Chen“-Notation des ERM
 - Bei Notation in UML ist Vererbung selbstsprechend
- Das relationale Modell ist für Vererbung auch nicht optimal ausgelegt
- Standardvorgehen, falls wir doch Vererbungen benötigen:
 - Eine Generalisierung wird wie eine 1:1-Beziehung behandelt
 - als Fremdschlüssel in der spezialisierten Relation dient der PK der generalisierten Relation
 - Um NULL-Werte zu vermeiden, werden die Relationen i.d.R. nicht zusammengefasst, obwohl der PK identisch ist
- Beispiel auf den Folgefolien:



Relationales Modell

Überführung vom ER-Modell – Schritt 5 (Generalisierungen)

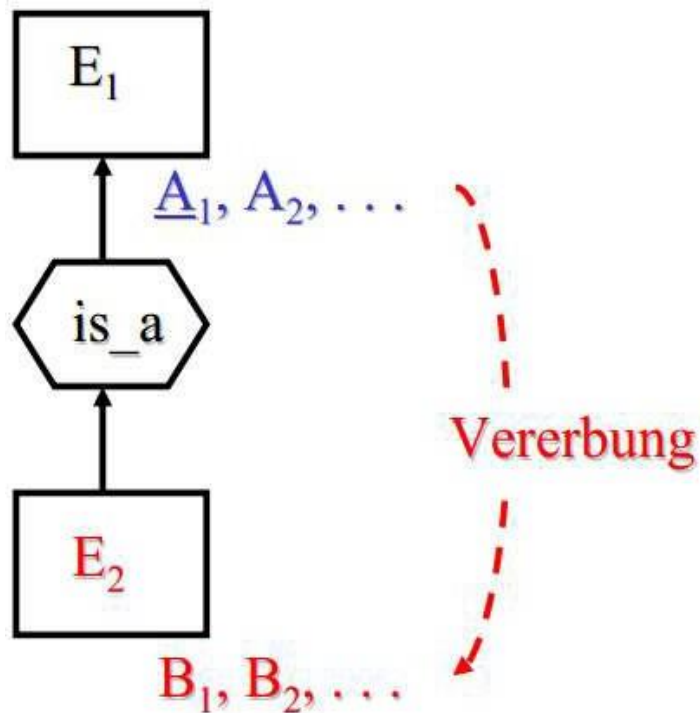




Relationales Modell

Überführung vom ER-Modell – Schritt 5 (Generalisierungen)

- Alternative zu diesem Vorgehen, die bei Abfragen schneller, jedoch nicht redundanzfrei ist und es wird zusätzlicher Aufwand bei Korrekturen benötigt:



E_1	<u>A_1</u>	A_2	\dots
	a_{11}	a_{12}	\dots
	a_{21}	a_{22}	\dots

E_2	<u>A_1</u>	A_2	\dots	B_1	B_2	\dots
	a_{11}	a_{12}	\dots	b_{11}	b_{12}	\dots

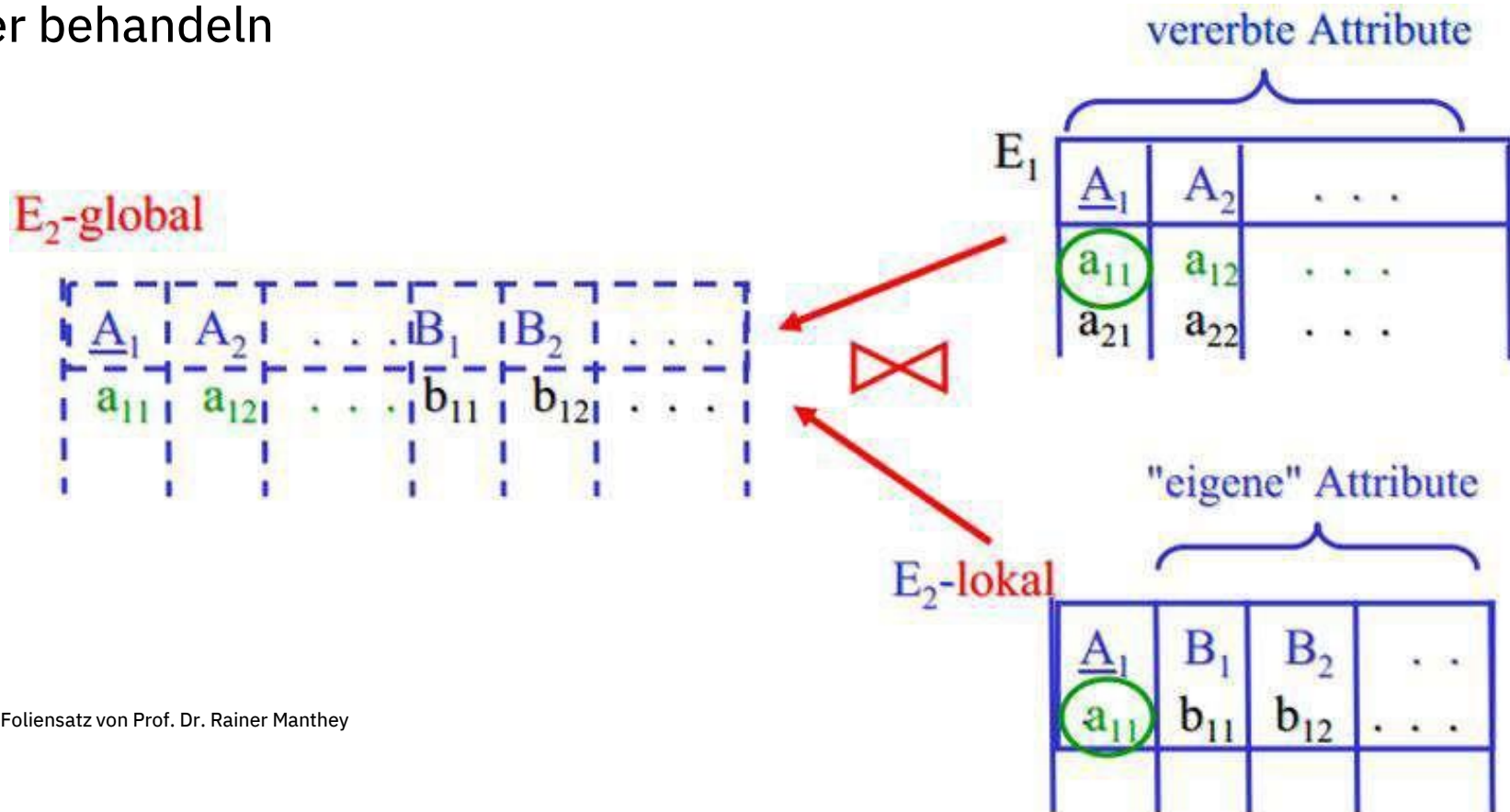
vererbte "eigene" Attribute



Relationales Modell

Überführung vom ER-Modell – Schritt 5 (Generalisierungen)

- Ausweg: Daten werden intern redundanzfrei hinterlegt, nach außen hin jedoch in Form einer vordefinierten Abfrage („Sicht“, „View“) bereitgestellt: – Dieses Vorgehen werden wir im Rahmen von „SQL“ näher behandeln

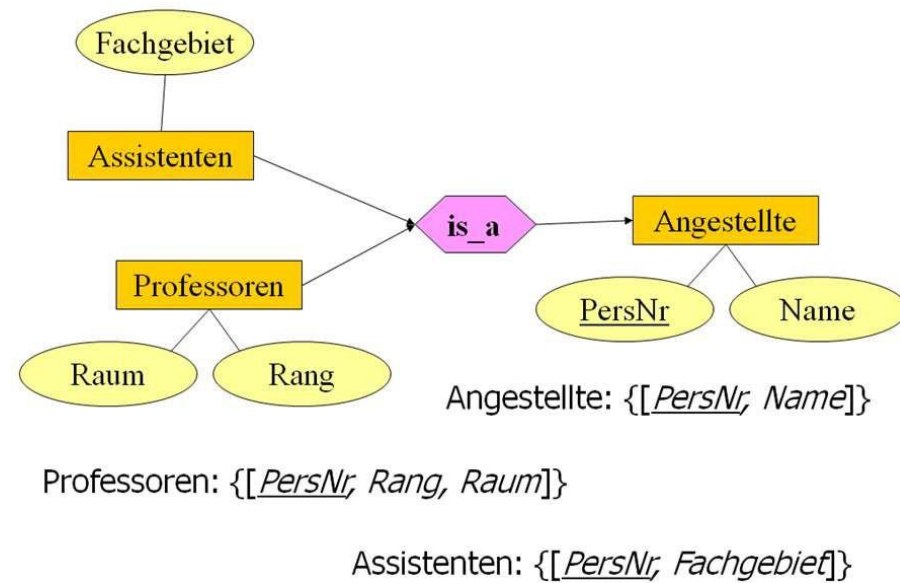
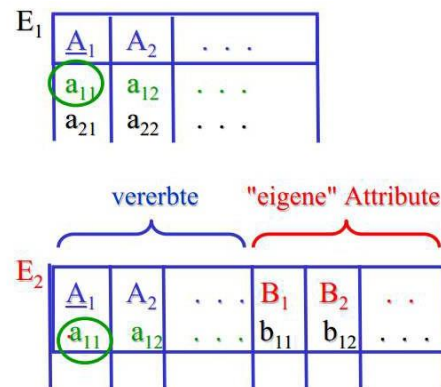
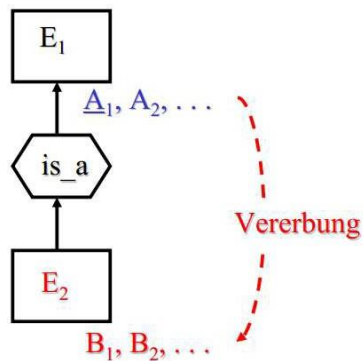




Relationales Modell

Überführung vom ER-Modell – Schritt 5 (Generalisierungen)

- Angestellte und Assistenten haben zwar den gleichen PK, werden aber nicht zusammengefasst, weil es einen Grund gibt, der dagegen spricht: Vermeidung von NULL-Werten in den „eigenen Attributen“ paralleler Spezialisierungszweige. Für E2.A1 gilt auch hier NOT NULL, da 1:c





Relationales Modell

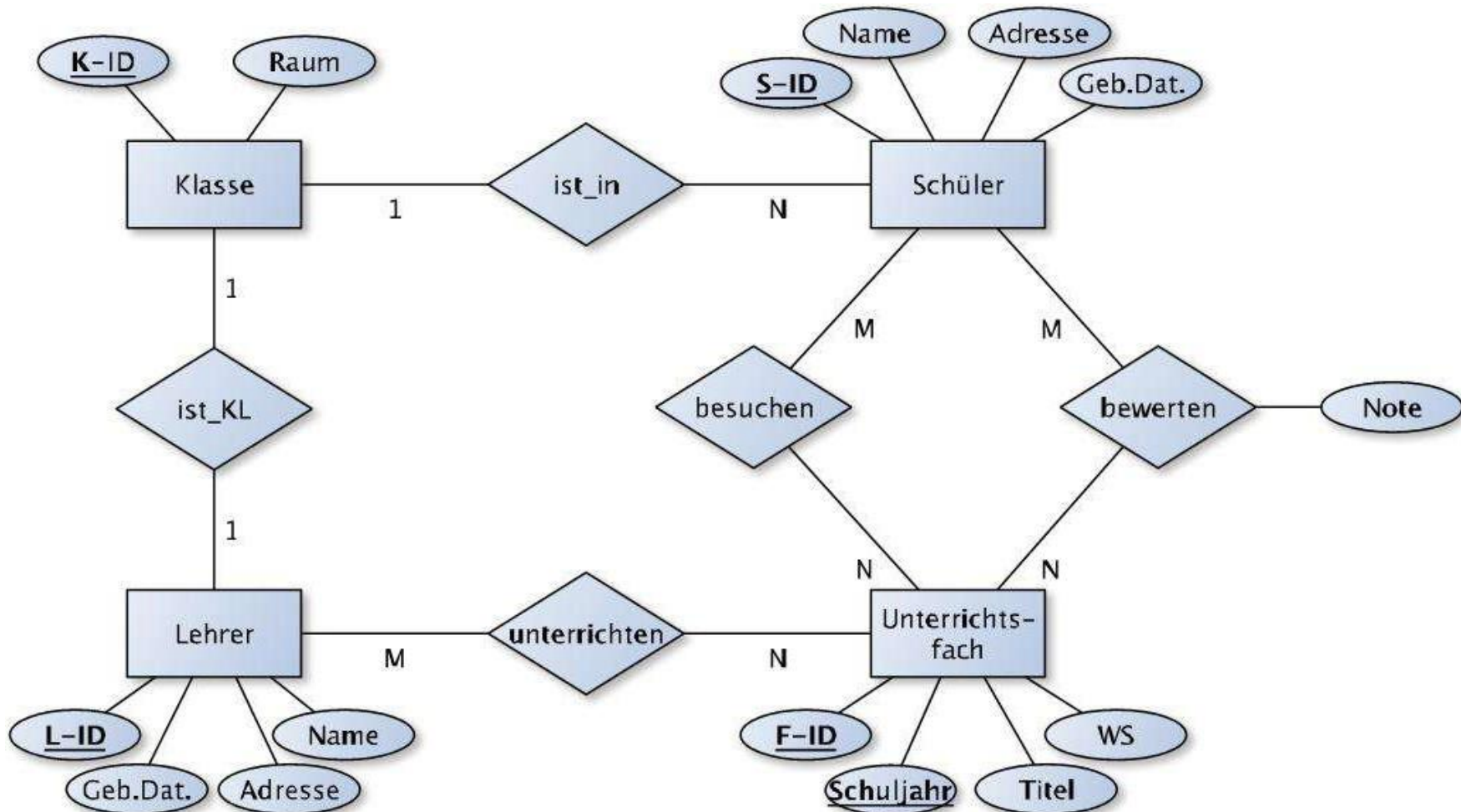
Überführung vom ER-Modell :: Zusammenfassung

Transformationsregeln:

- Jeder Entity-Typ wird zu einem Relationenschema
 - **Straight forward**
- Mehrwertige Attribute werden in ein Relationenschema überführt
 - **Mehrwertige Attribute wie 1:N-Beziehungen behandeln, und weiter**
- Jeder Relationshiptyp wird zu einem Relationenschema
 - **PKs aller Teilnehmer werden zu FKs in der neuen Relation, PK der neuen Relation ist Teilmenge der FKs**
- Relationen mit gleichem PK werden, falls kein Grund dagegen spricht, zusammengefasst
 - **1:N-Beziehungen allgemein und 1:c-Beziehungen speziell benötigen keine zusätzliche Relation für den Relationship-Typen. Wenn Sie das verinnerlicht haben, überspringen Sie Schritt 3 meist intuitiv**
 - **Die modifizierte Chen-Notation ist hier recht hilfreich (c=„can“)**
- Generalisierungen werden modelliert
 - **Wird wie eine 1:c-Beziehung behandelt, jedoch i.d.R. nicht gemerged**

Relationales Modell

Übung



Relationales Modell

Übung



?

Überführen Sie das ER-Diagramm von der letzten Seite in ein logisches DB-Schema in Form des Relationenmodells.

- (1) Wenden Sie im ersten Schritt die ersten 3 bekannten Transformationsregeln an. Ergänzen Sie das Modell um sinnvolle Datentypen. Es wurde die einfache Chen-Notation verwendet... definieren Sie ggf. selbst „can“-Beziehungen.
- (2) Verfeinern Sie das Modell im Anschluss mit der 4. Transformationsregel
- (3) Hinterfragen Sie die fachliche Korrektheit des ER-Modells. Fällt Ihnen etwas bei diesem ER-Modell auf? Welche Probleme könnten sich ergeben?

Bearbeitungszeit:
20 Minuten