

Script generated by TTT

Title: groh: profile1 (22.04.2016)

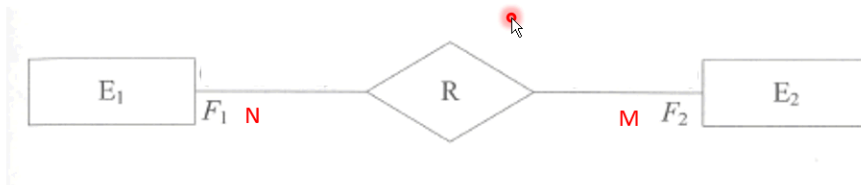
Date: Fri Apr 22 13:59:58 CEST 2016

Duration: 83:17 min

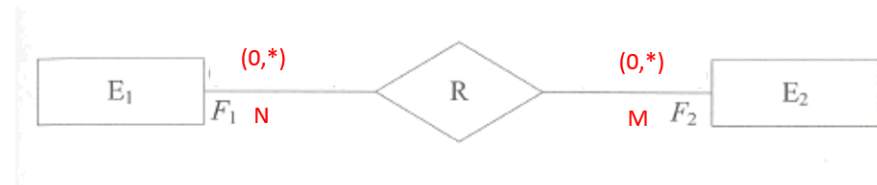
Pages: 58

Aufgabe 2.2

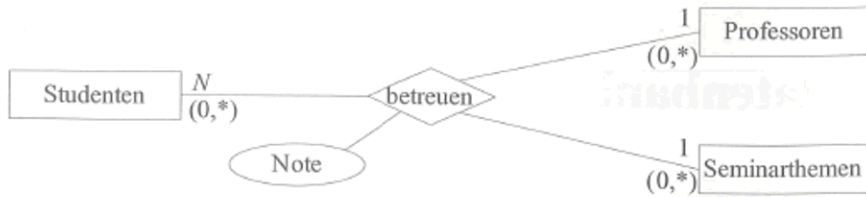
Zeigen Sie, dass die Ausdruckskraft der Funktionalitätsangaben und der (min, max) -Angaben bei n -stelligen Beziehungen mit $n > 2$ unvergleichbar ist: Finden Sie realistische Beispiele von Konsistenzbedingungen, die mit Funktionalitätsangaben, aber nicht mit (min, max) -Angaben ausdrückbar sind, und wiederum andere Konsistenzbedingungen, die mit der (min, max) -Angabe formulierbar sind aber nicht durch Funktionalitätseinschränkungen.



$F_1 : F_2$	(min_1, max_1)	(min_2, max_2)
1 : 1	(0, 1)	(0, 1)
1 : N	(0, *)	(0, 1)
N : 1	(0, 1)	(0, *)
N : M		



$F_1 : F_2$	(min_1, max_1)	(min_2, max_2)
1 : 1	(0, 1)	(0, 1)
1 : N	(0, *)	(0, 1)
N : 1	(0, 1)	(0, *)
N : M	(0, *)	(0, *)



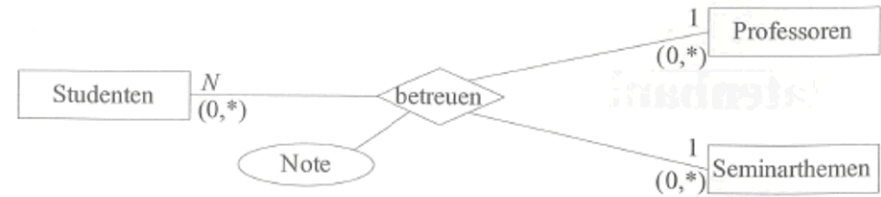
partielle Funktionen:

$\text{betreuen} : \text{Professoren} \times \text{Studenten} \rightarrow \text{Seminarthemen}$
 $\text{betreuen} : \text{Seminarthemen} \times \text{Studenten} \rightarrow \text{Professoren}$

Konsistenzbedingungen:

- Studenten dürfen bei einem Professor nur **ein** Seminarthema bearbeiten.
- Studenten dürfen dasselbe Thema nur bei **einem** Professor bearbeiten.

78



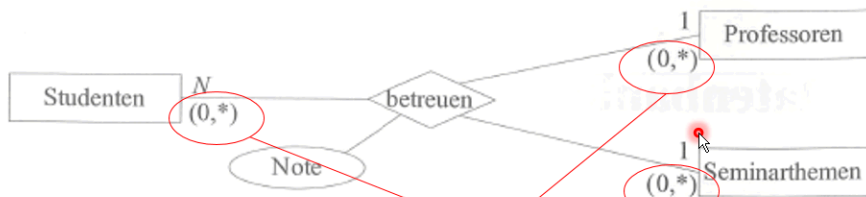
partielle Funktionen:

$\text{betreuen} : \text{Professoren} \times \text{Studenten} \rightarrow \text{Seminarthemen}$
 $\text{betreuen} : \text{Seminarthemen} \times \text{Studenten} \rightarrow \text{Professoren}$

Konsistenzbedingungen:

- Studenten dürfen bei einem Professor nur **ein** Seminarthema bearbeiten.
- Studenten dürfen dasselbe Thema nur bei **einem** Professor bearbeiten.

78



partielle Funktionen:

$\text{betreuen} : \text{Professoren} \times \text{Studenten} \rightarrow \text{Seminarthemen}$
 $\text{betreuen} : \text{Seminarthemen} \times \text{Studenten} \rightarrow \text{Professoren}$

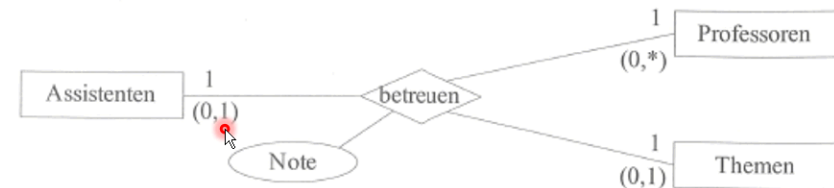
Müssen aus Modellierungsgründen so sein (bspw: Profs verwenden Seminarthemen erneut), aber können, ohne 1:N:1 dargestellt, die Konsistenzbedingungen, die durch 1:N:1 modelliert werden, verletzen

Konsistenzbedingungen:

- Studenten dürfen bei einem Professor nur **ein** Seminarthema bearbeiten.
- Studenten dürfen dasselbe Thema nur bei **einem** Professor bearbeiten.

79

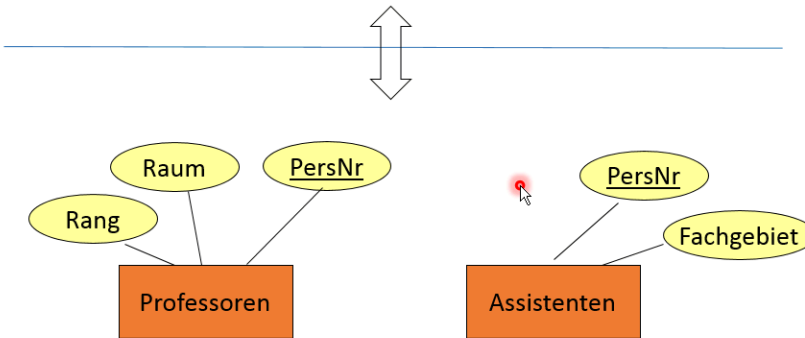
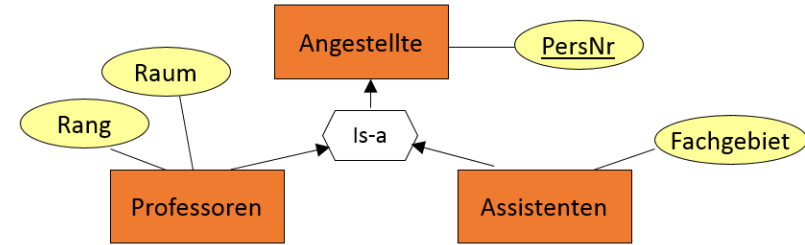
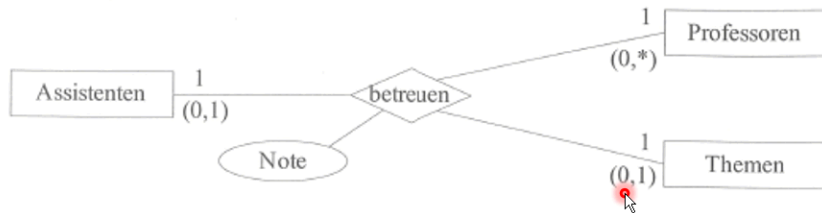
weiteres Bsp:



80

Generalisierung

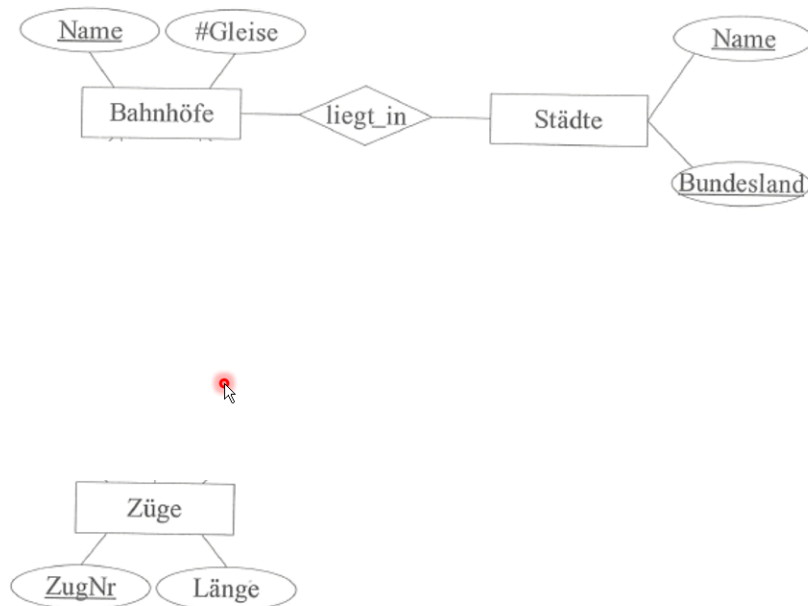
weiteres Bsp:



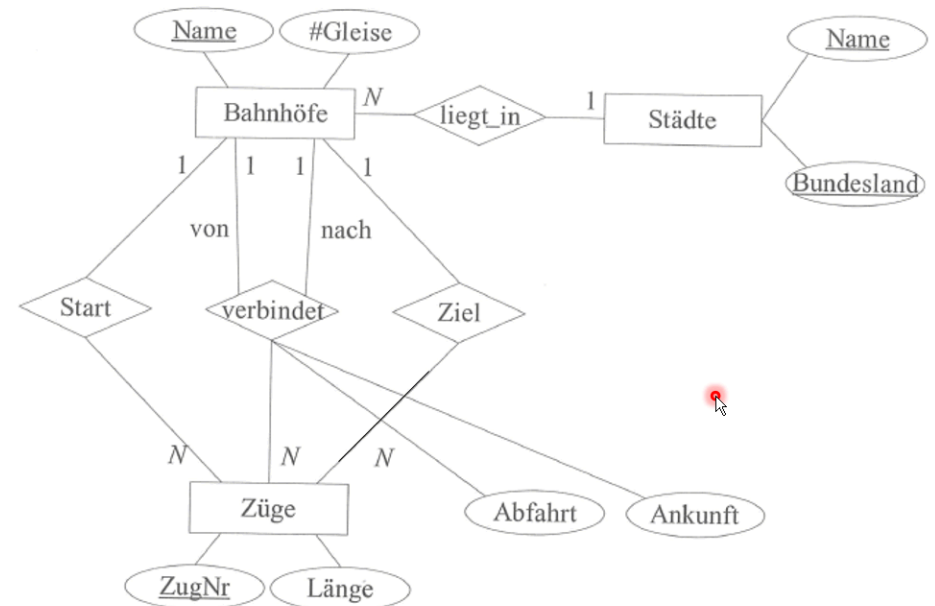
80

65

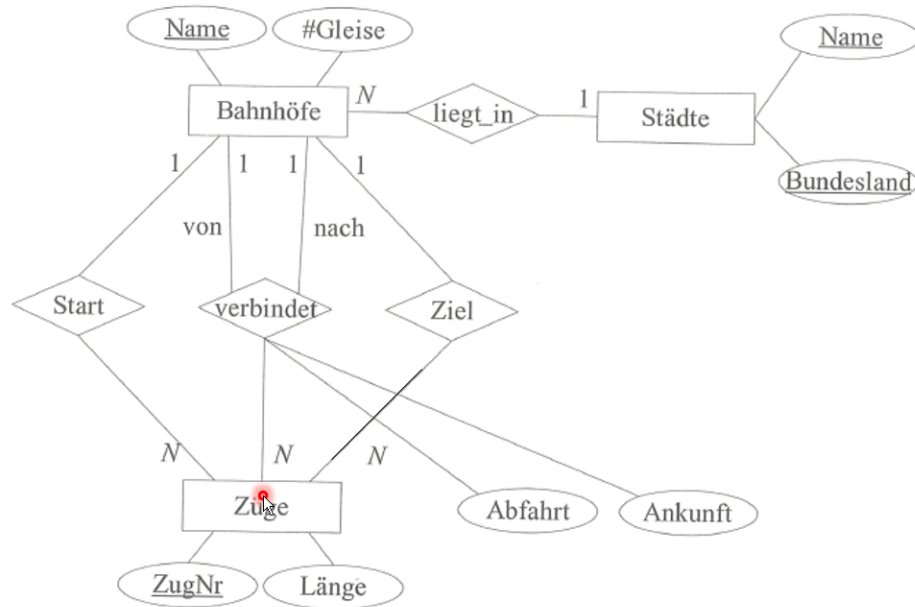
Aufgabe: Zugauskunftssystem



Aufgabe: Zugauskunftssystem

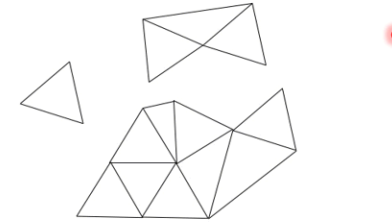


Aufgabe: Zugauskunftssystem

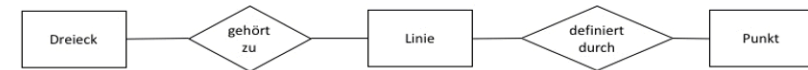


Aufgabe Triangulierung

In der Computergrafik werden Flächen in der Regel durch Dreiecke approximiert (**Triangulierung**).



Gegeben sei folgendes unvollständiges **ER-Diagramm** für eine solche Triangulierung

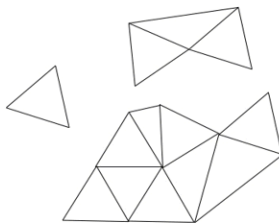


Tragen Sie für die Beziehungen **sinnvolle Funktionalitäten** (also **1:1**, **1:N**, **N:M**,...) **UND Angaben** in (**min**, **max**) Notation ein!

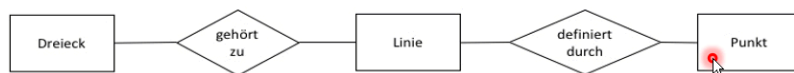
92

Aufgabe Triangulierung

In der Computergrafik werden Flächen in der Regel durch Dreiecke approximiert (**Triangulierung**).



Gegeben sei folgendes unvollständiges **ER-Diagramm** für eine solche Triangulierung

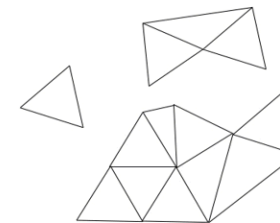


Tragen Sie für die Beziehungen **sinnvolle Funktionalitäten** (also **1:1**, **1:N**, **N:M**,...) **UND Angaben** in (**min**, **max**) Notation ein!

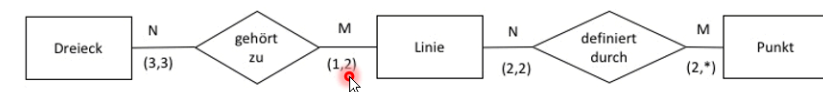
92

Aufgabe Triangulierung

In der Computergrafik werden Flächen in der Regel durch Dreiecke approximiert (**Triangulierung**).



Gegeben sei folgendes unvollständiges **ER-Diagramm** für eine solche Triangulierung

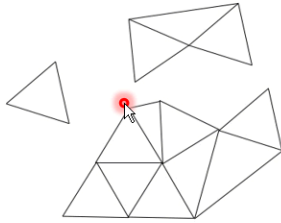


Tragen Sie für die Beziehungen **sinnvolle Funktionalitäten** (also **1:1**, **1:N**, **N:M**,...) **UND Angaben** in (**min**, **max**) Notation ein!

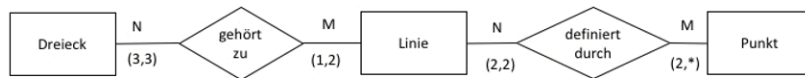
93

Aufgabe Triangulierung

In der Computergrafik werden Flächen in der Regel durch Dreiecke approximiert (**Triangulierung**).



Gegeben sei folgendes unvollständiges **ER-Diagramm** für eine solche Triangulierung

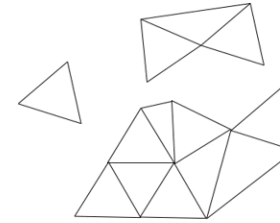


Tragen Sie für die Beziehungen **sinnvolle Funktionalitäten** (also **1:1, 1:N, N:M,...**) **UND Angaben** in (**min, max**) Notation ein!

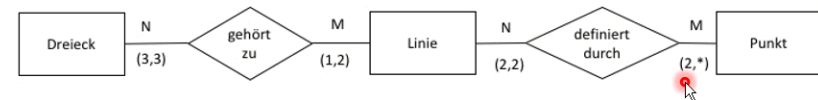
93

Aufgabe Triangulierung

In der Computergrafik werden Flächen in der Regel durch Dreiecke approximiert (**Triangulierung**).



Gegeben sei folgendes unvollständiges **ER-Diagramm** für eine solche Triangulierung

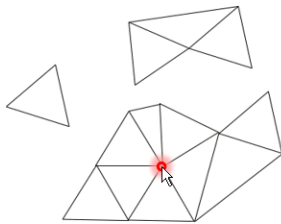


Tragen Sie für die Beziehungen **sinnvolle Funktionalitäten** (also **1:1, 1:N, N:M,...**) **UND Angaben** in (**min, max**) Notation ein!

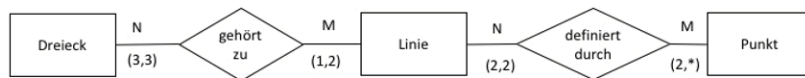
93

Aufgabe Triangulierung

In der Computergrafik werden Flächen in der Regel durch Dreiecke approximiert (**Triangulierung**).



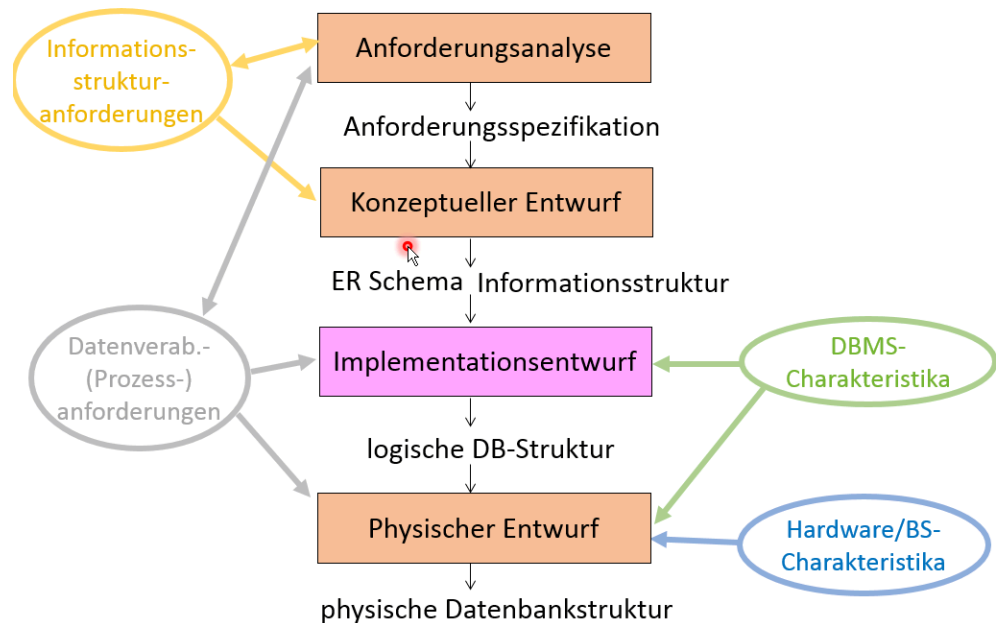
Gegeben sei folgendes unvollständiges **ER-Diagramm** für eine solche Triangulierung



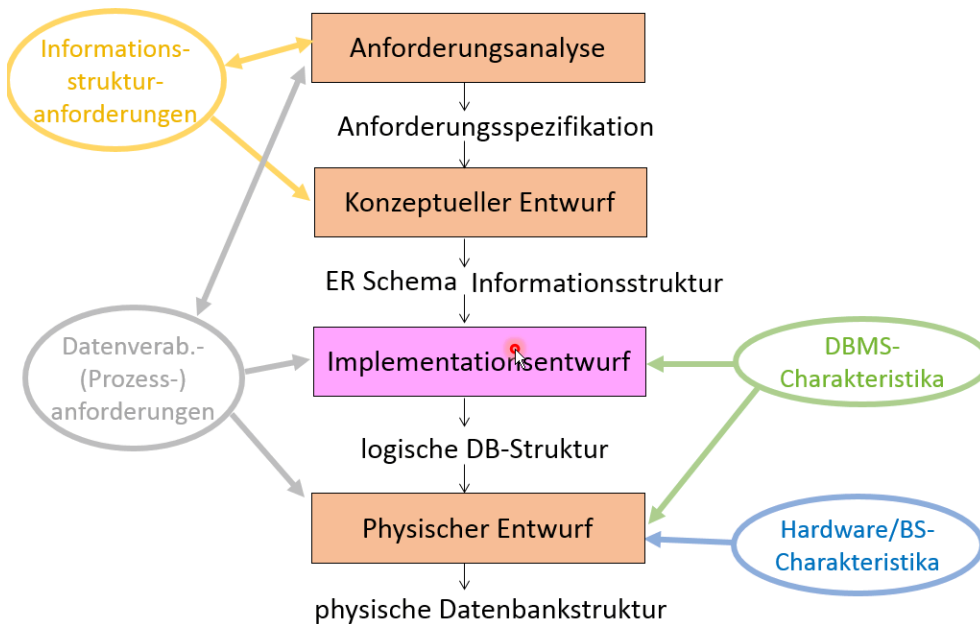
Tragen Sie für die Beziehungen **sinnvolle Funktionalitäten** (also **1:1, 1:N, N:M,...**) **UND Angaben** in (**min, max**) Notation ein!

93

Phasen des Datenbankentwurfs



95



95

Seien A_1, A_2, \dots, A_n **Attribute** und D_1, D_2, \dots, D_n ihre **Domänen** (Wertebereiche)

- **Relation:** $R \subseteq D_1 \times \dots \times D_n$
Bsp.: Telefonbuch \subseteq string x string x integer
- **Tupel:** $t \in R$
Bsp.: $t = (\text{„Mickey Mouse“}, \text{„Main Street“}, 8124711)$
- **Schema:** legt die Struktur der gespeicherten Daten fest
Bsp.: Telefonbuch: $\{[\text{Name: string, Straße: string, Telefon\#:integer}]\}$
Notation: sch(R)

96

Seien A_1, A_2, \dots, A_n **Attribute** und D_1, D_2, \dots, D_n ihre **Domänen** (Wertebereiche)

- **Relation:** $R \subseteq D_1 \times \dots \times D_n$
Bsp.: Telefonbuch \subseteq string x string x integer
- **Tupel:** $t \in R$
Bsp.: $t = (\text{„Mickey Mouse“}, \text{„Main Street“}, 8124711)$
- **Schema:** legt die Struktur der gespeicherten Daten fest
Bsp.: Telefonbuch: $\{[\text{Name: string, Straße: string, Telefon\#:integer}]\}$
Notation: sch(R)

96

Seien A_1, A_2, \dots, A_n **Attribute** und D_1, D_2, \dots, D_n ihre **Domänen** (Wertebereiche)

- **Relation:** $R \subseteq D_1 \times \dots \times D_n$
Bsp.: Telefonbuch \subseteq string x string x integer
- **Tupel:** $t \in R$
Bsp.: $t = (\text{„Mickey Mouse“}, \text{„Main Street“}, 8124711)$
- **Schema:** legt die Struktur der gespeicherten Daten fest
Bsp.: Telefonbuch: $\{[\text{Name: string, Straße: string, Telefon\#:integer}]\}$
Notation: sch(R)

96

Seien A_1, A_2, \dots, A_n **Attribute** und
 D_1, D_2, \dots, D_n ihre **Domänen** (Wertebereiche)

- **Relation:** $R \subseteq D_1 \times \dots \times D_n$
Bsp.: Telefonbuch \subseteq string x string x integer
- **Tupel:** $t \in R$
Bsp.: $t = (\text{„Mickey Mouse“}, \text{„Main Street“}, 8124711)$
- **Schema:** legt die Struktur der gespeicherten Daten fest
Bsp.: Telefonbuch: {[Name: string, Straße: string, Telefon#:integer]}
Notation: sch(R)

96

Seien A_1, A_2, \dots, A_n **Attribute** und
 D_1, D_2, \dots, D_n ihre **Domänen** (Wertebereiche)

- **Relation:** $R \subseteq D_1 \times \dots \times D_n$
Bsp.: Telefonbuch \subseteq string x string x integer
- **Tupel:** $t \in R$
Bsp.: $t = (\text{„Mickey Mouse“}, \text{„Main Street“}, 8124711)$
- **Schema:** legt die Struktur der gespeicherten Daten fest
Bsp.: Telefonbuch: {[Name: string, Straße: string, Telefon#:integer]}
Notation: sch(R)

96

Seien A_1, A_2, \dots, A_n **Attribute** und
 D_1, D_2, \dots, D_n ihre **Domänen** (Wertebereiche)

- **Relation:** $R \subseteq D_1 \times \dots \times D_n$
Bsp.: Telefonbuch \subseteq string x string x integer
- **Tupel:** $t \in R$
Bsp.: $t = (\text{„Mickey Mouse“}, \text{„Main Street“}, 8124711)$
- **Schema:** legt die Struktur der gespeicherten Daten fest
Bsp.: Telefonbuch: {[Name: string, Straße: string, Telefon#:integer]}
Notation: sch(R)

96

Seien A_1, A_2, \dots, A_n **Attribute** und
 D_1, D_2, \dots, D_n ihre **Domänen** (Wertebereiche)

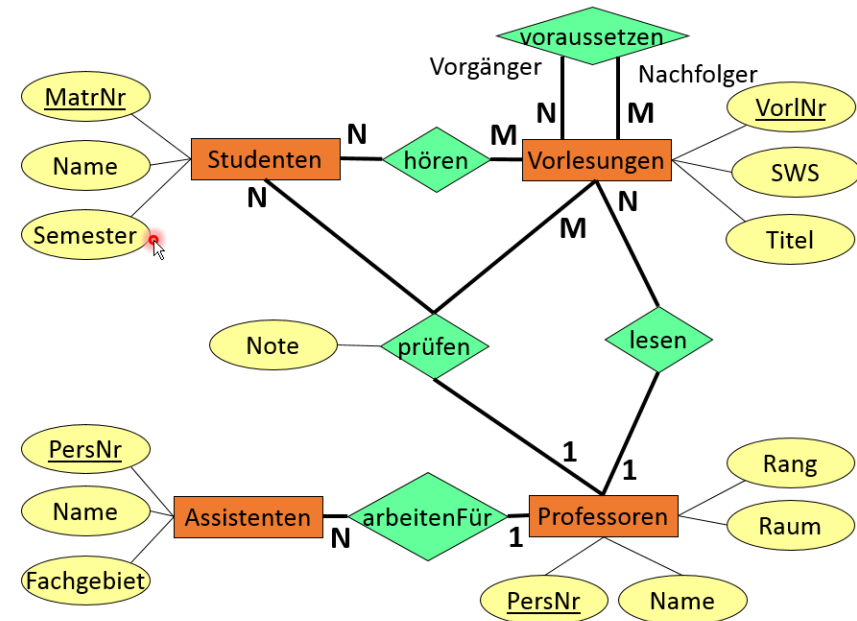
- **Relation:** $R \subseteq D_1 \times \dots \times D_n$
Bsp.: Telefonbuch \subseteq string x string x integer
- **Tupel:** $t \in R$
Bsp.: $t = (\text{„Mickey Mouse“}, \text{„Main Street“}, 8124711)$
- **Schema:** legt die Struktur der gespeicherten Daten fest
Bsp.: Telefonbuch: {[Name: string, Straße: string, Telefon#:integer]}
Notation: sch(R)

96

Telefonbuch		
Name:String	Straße:String	Telefon#:integer
Mickey Mouse	Main Street	4711
Donald Duck	Broadway	95672
...

- **Ausprägung:** der aktuelle Zustand der Datenbasis
- **Schlüssel:** minimale Menge von Attributen, deren Werte ein Tupel eindeutig identifizieren
- **Primärschlüssel:** wird unterstrichen
 - Einer der Schlüsselkandidaten wird als Primärschlüssel ausgewählt
 - Hat eine besondere Bedeutung bei der Referenzierung von Tupeln

97



98

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]}



99

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]}



99

Relationale Darstellung von Entities

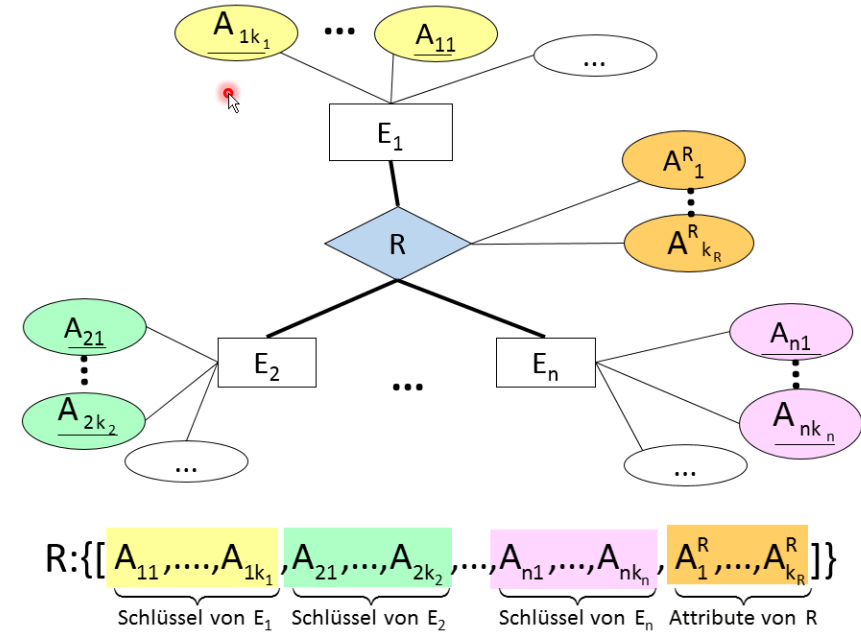
Studenten: {[MatrNr:integer, Name: string, Semester: integer]}

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

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

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

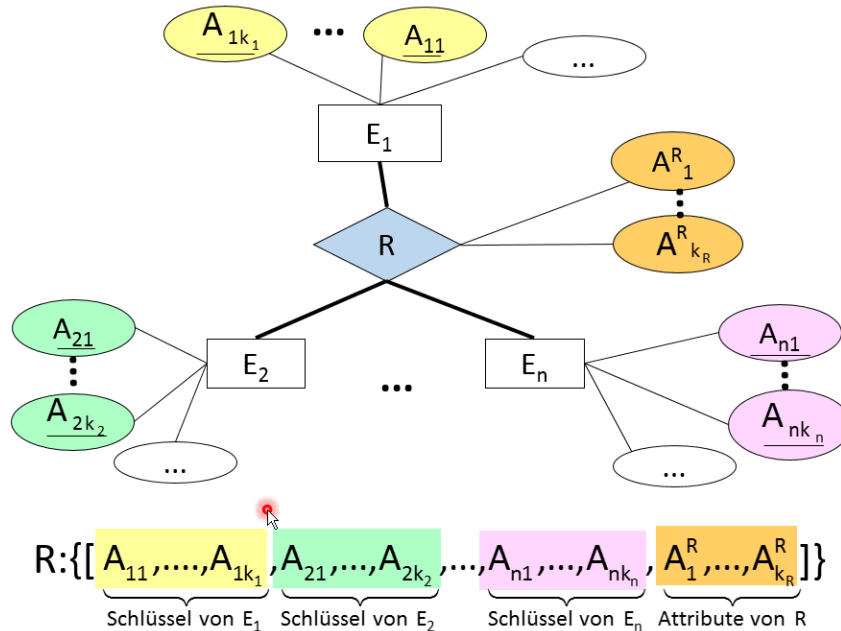
Relationale Darstellung von Beziehungen



99

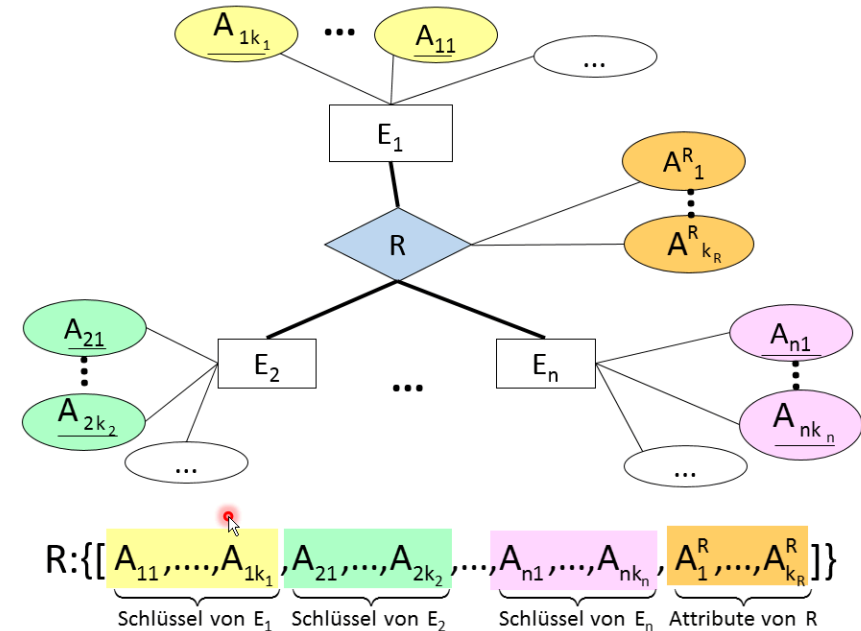
100

Relationale Darstellung von Beziehungen



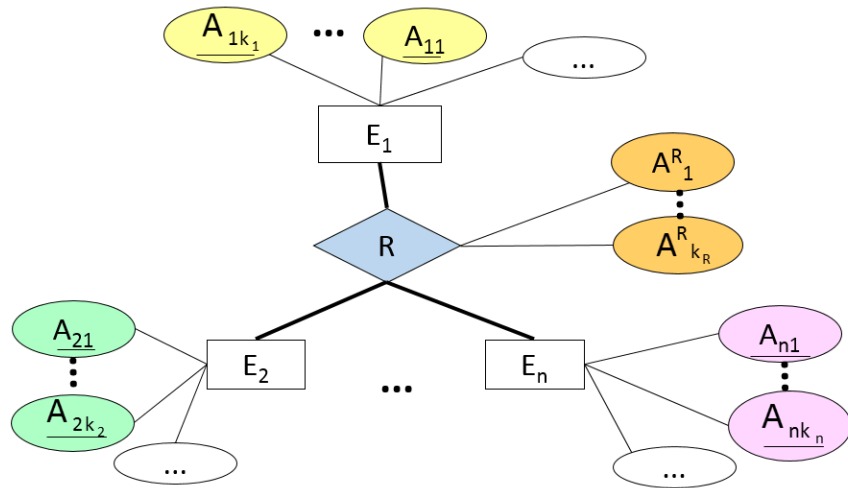
100

Relationale Darstellung von Beziehungen



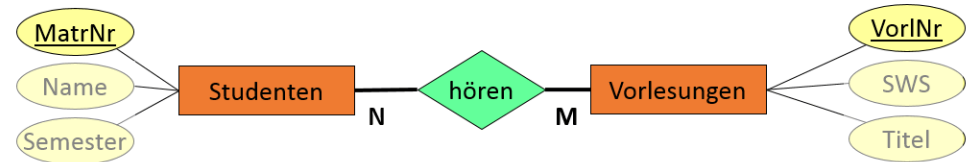
100

Relationale Darstellung von Beziehungen



$$R: \{ \underbrace{A_{11}, \dots, A_{1k_1}}_{\text{Schlüssel von } E_1}, \underbrace{A_{21}, \dots, A_{2k_2}}_{\text{Schlüssel von } E_2}, \dots, \underbrace{A_{n1}, \dots, A_{nk_n}}_{\text{Schlüssel von } E_n}, \underbrace{A_{1R}, \dots, A_{kR}}_{\text{Attribute von R}} \}$$

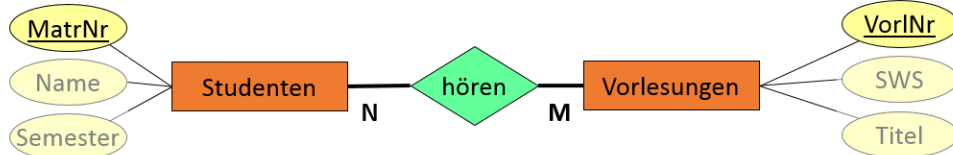
beziehungen unseres Beispiel-Schemas



Studenten		hören		Vorlesungen	
MatrNr	...	MatrNr	VorlNr	VorlNr	...
...	...	26120	5001	5001	...
26120	...	27550	5001
...	...	27550	4052
27550	...	28106	5041	4052	...
...	...	28106	5052
...

Primärschlüssel von hören
 hören : {[MatrNr: integer, VorlNr: integer]}
 Fremdschlüssel

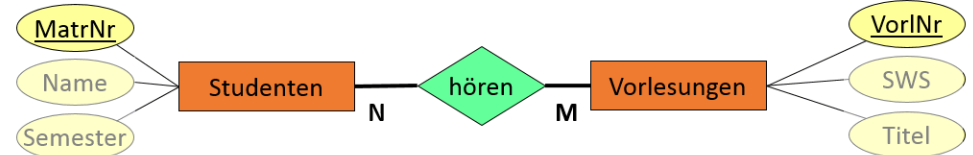
beziehungen unseres Beispiel-Schemas



Studenten		hören		Vorlesungen	
MatrNr	...	MatrNr	VorlNr	VorlNr	...
...	...	26120	5001	5001	...
26120	...	27550	5001
...	...	27550	4052
27550	...	28106	5041	4052	...
...	...	28106	5052
...

Primärschlüssel von hören
 hören : {[MatrNr: integer, VorlNr: integer]}
 Fremdschlüssel

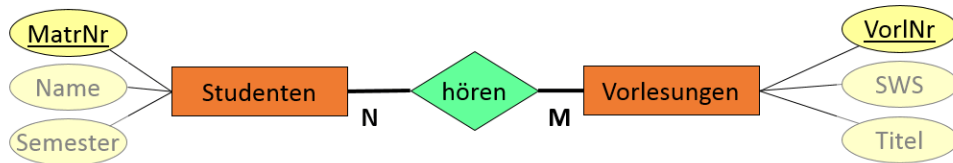
beziehungen unseres Beispiel-Schemas



Studenten		hören		Vorlesungen	
MatrNr	...	MatrNr	VorlNr	VorlNr	...
...	...	26120	5001	5001	...
26120	...	27550	5001
...	...	27550	4052
27550	...	28106	5041	4052	...
...	...	28106	5052
...

Primärschlüssel von hören
 hören : {[MatrNr: integer, VorlNr: integer]}
 Fremdschlüssel

beziehungen unseres Beispiel-Schemas



Studenten		hören		Vorlesungen	
MatrNr	...	MatrNr	VorlNr	VorlNr	...
...	...	26120	5001	5001	...
26120	...	27550	5001	4052	...
...	...	27550	4052
...	...	28106	5041
27550	...	28106	5052	4052	...
...

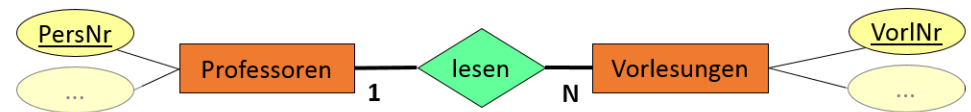
Primärschlüssel von hören

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

Fremdschlüssel

101

beziehungen unseres Beispiel-Schemas



Professoren		lesen		Vorlesungen	
PersNr	...	PersNr	VorlNr	VorlNr	...
2125	...	2137	5001	5001	...
...	...	2125	5041	5041	...
...
...	...	2125	5049	5049	...
2137

Primärschlüssel von lesen

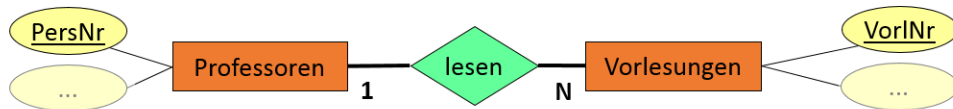
lesen : {[PersNr: integer, VorlNr: integer]}

Fremdschlüssel

Nur das ist der Primärschlüssel von lesen!!!
Der zugehörige Prof ist durch 1:N bereits eindeutig bestimmt

102

beziehungen unseres Beispiel-Schemas



Professoren		lesen		Vorlesungen	
PersNr	...	PersNr	VorlNr	VorlNr	...
2125	...	2137	5001	5001	...
...	...	2125	5041	5041	...
...
...	...	2125	5049	5049	...
2137

Primärschlüssel von lesen

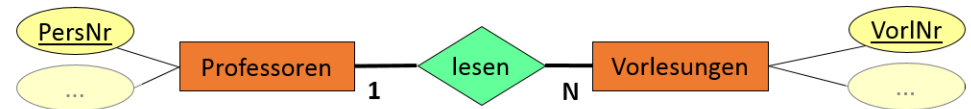
lesen : {[PersNr: integer, VorlNr: integer]}

Fremdschlüssel

Nur das ist der Primärschlüssel von lesen!!!
Der zugehörige Prof ist durch 1:N bereits eindeutig bestimmt

102

verfeinerung des relationalen Schemas

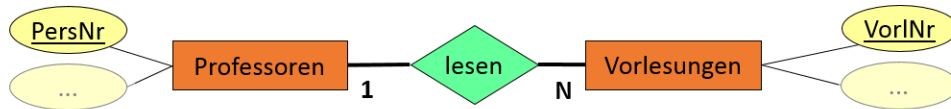


⇒ 1:N-Beziehungen (auch 1:1-Bez.) sind verfeinerbar:

- Initial-Entwurf:
 - Vorlesungen : {[VorlNr, Titel, SWS]}
 - Professoren : {[PersNr, Name, Rang, Raum]}
 - lesen : {[PersNr, VorlNr]} ≡ lesen : {[VorlNr, PersNr]}
- Verfeinerung durch Zusammenfassung
 - Vorlesungen : {[VorlNr, Titel, SWS, gelesenVon]}
 - Professoren : {[PersNr, Name, Rang, Raum]}

103

verfeinerung des relationalen Schemas



⇒ 1:N-Beziehungen (auch 1:1-Bez.) sind **verfeinerbar**:

- Initial-Entwurf:

Vorlesungen : {[VorNr], Titel, SWS}}

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

lesen : {[PersNr, VorNr]} ≡ *lesen* : {[VorNr, PersNr]}

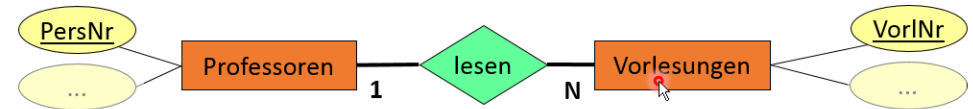
- **Verfeinerung** durch Zusammenfassung

Vorlesungen : {[VorNr], Titel, SWS, *gelesenVon*}

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

103

verfeinerung des relationalen Schemas



⇒ 1:N-Beziehungen (auch 1:1-Bez.) sind **verfeinerbar**:

- Initial-Entwurf:

Vorlesungen : {[VorNr], Titel, SWS}}

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

lesen : {[PersNr, VorNr]} ≡ *lesen* : {[VorNr, PersNr]}

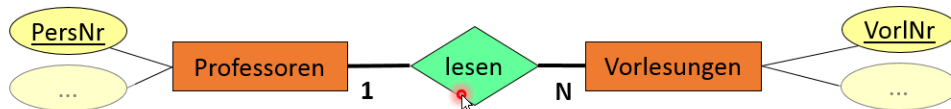
- **Verfeinerung** durch Zusammenfassung

Vorlesungen : {[VorNr], Titel, SWS, *gelesenVon*}

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

103

verfeinerung des relationalen Schemas



⇒ 1:N-Beziehungen (auch 1:1-Bez.) sind **verfeinerbar**:

- Initial-Entwurf:

Vorlesungen : {[VorNr], Titel, SWS}}

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

lesen : {[PersNr, VorNr]} ≡ *lesen* : {[VorNr, PersNr]}

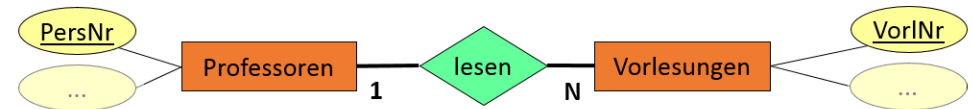
- **Verfeinerung** durch Zusammenfassung

Vorlesungen : {[VorNr], Titel, SWS, *gelesenVon*}

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

103

verfeinerung des relationalen Schemas



⇒ 1:N-Beziehungen (auch 1:1-Bez.) sind **verfeinerbar**:

- Initial-Entwurf:

Vorlesungen : {[VorNr], Titel, SWS}}

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

lesen : {[PersNr, VorNr]} ≡ *lesen* : {[VorNr, PersNr]}

- **Verfeinerung** durch Zusammenfassung

Vorlesungen : {[VorNr], Titel, SWS, *gelesenVon*}

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

104

Regel:
Relationen mit gleichem Schlüssel kann man zusammenfassen
aber **nur diese** und keine anderen!

verfeinerung des relationalen Schemas

Ausprägung von Professoren und Vorlesungen nach Verfeinerung

PersNr	Name	Rang	Raum	VorNr	Titel	SWS	gelesen
2125	Sokrates	C4	226	5001	Grundzüge	4	2127
2125	Sokrates	C4	226	5041	Ethik	4	2129
2125	Sokrates	C4	226	5043	Erkenntnistheorie	3	2128
2125	Sokrates	C4	226	5049	Mäeutik	2	2125
2134	Augustinus	C3	309	4052	Logik	4	2125
2136	Curie	C4	36	5022	Wissenschaftstheorie	3	2129
2137	Karl	C4	2

Vorsicht: So geht es NICHT !! --> Anomalien

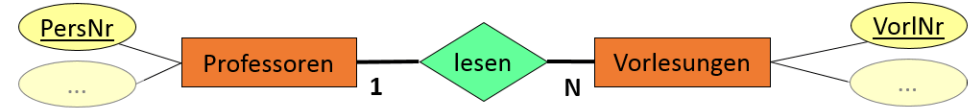
Professoren				
PersNr	Name	Rang	Raum	liest
2125	Sokrates	C4	226	5041
2125	Sokrates	C4	226	5049
2125	Sokrates	C4	226	4052
...
2134	Augustinus	C3	309	5022
2136	Curie	C4	36	???
...

Vorlesungen		
VorNr	Titel	SWS
5001	Grundzüge	4
5041	Ethik	4
5043	Erkenntnistheorie	3
5049	Mäeutik	2
4052	Logik	4
5052	Wissenschaftstheorie	3
5022	Glaube und Wissen	2
...

- **Update**-Anomalie: Was passiert wenn Sokrates umzieht?
- **Lösch**-Anomalie: Was passiert wenn „Glaube und Wissen“ wegfällt?
- **Einfüge**-Anomalie: Curie forscht nur und liest keine Vorlesung.

106

verfeinerung des relationalen Schemas



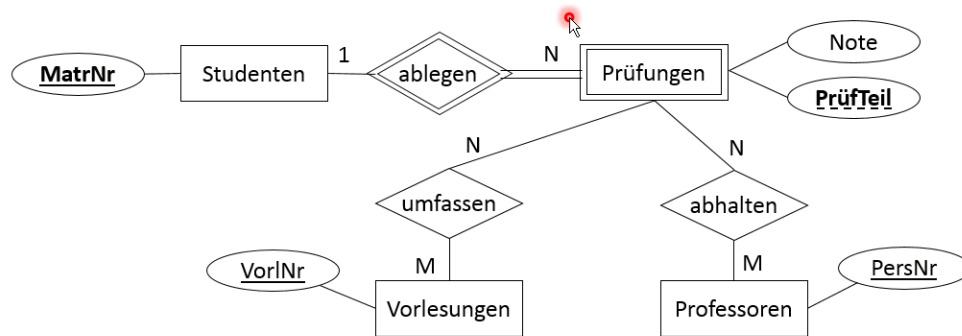
⇒ 1:N-Beziehungen (auch 1:1-Bez.) sind verfeinerbar:

- Initial-Entwurf:
 - Vorlesungen** : {[VorNr, Titel, SWS]}
 - Professoren** : {[PersNr, Name, Rang, Raum]}
 - lesen** : {[PersNr, VorNr]} ≡ lesen : {[VorNr, PersNr]}
- **Verfeinerung** durch Zusammenfassung
 - Vorlesungen** : {[VorNr, Titel, SWS, *gelesenVon*]}
 - Professoren** : {[PersNr, Name, Rang, Raum]}

Regel: Relationen mit gleichem Schlüssel kann man zusammenfassen aber **nur diese** und keine anderen!

104

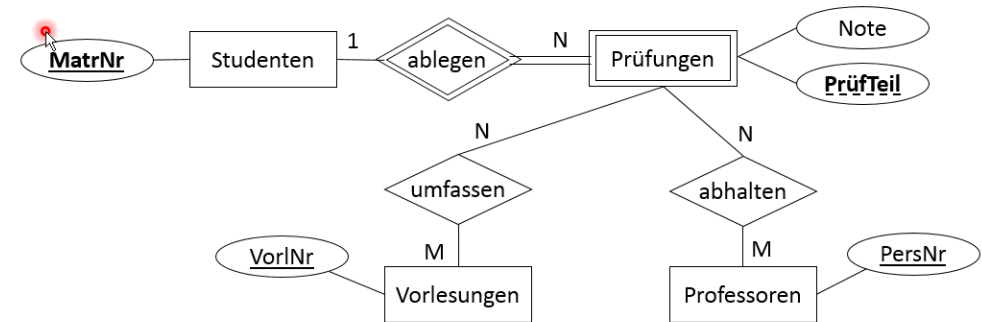
Relationale Modellierung schwacher Entitytypen



Prüfungen: {[**MatrNr: integer**, PrüfTeil: string, Note: integer]}

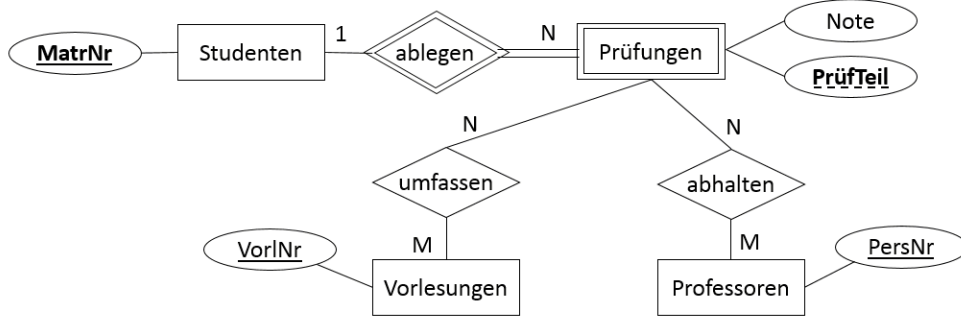
107

Relationale Modellierung schwacher Entitytypen



Prüfungen: {[**MatrNr: integer**, PrüfTeil: string, Note: integer]}

107



Prüfungen: {[**MatrNr: integer, PrüfTeil: string**, Note: integer]}

umfassen: {[**MatrNr: integer, PrüfTeil: string, VorlNr: integer**]}

abhalten: {[**MatrNr: integer, PrüfTeil: string, PersNr: integer**]}

Fremdschlüssel auf ein schwaches Entity: Es muss der (global eindeutige) Schlüssel der Relation *Prüfung* nämlich *MatrNr* und *PrüfTeil* als Fremdschlüssel in die Relationen *umfassen* und *abhalten* übernommen werden.

Beispiel-Aufgaben



Aufgabe 3.1

Gegeben sei die ER-Modellierung von Zugverbindungen in Abbildung 3.1.

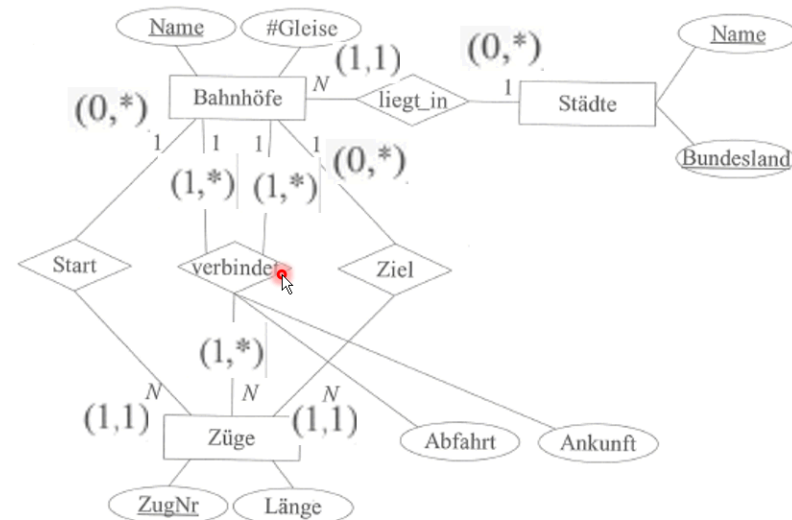
- Fügen Sie bei den Beziehungen Kardinalitäten in der (min, max) -Notation hinzu.
- Übertragen Sie das ER-Modell in ein relationales Schema.
- Verfeinern Sie das relationale Schema soweit möglich durch Eliminierung von Relationen.



Aufgabe 3.1

Gegeben sei die ER-Modellierung von Zugverbindungen in Abbildung 3.1.

- Fügen Sie bei den Beziehungen Kardinalitäten in der (min, max) -Notation hinzu.

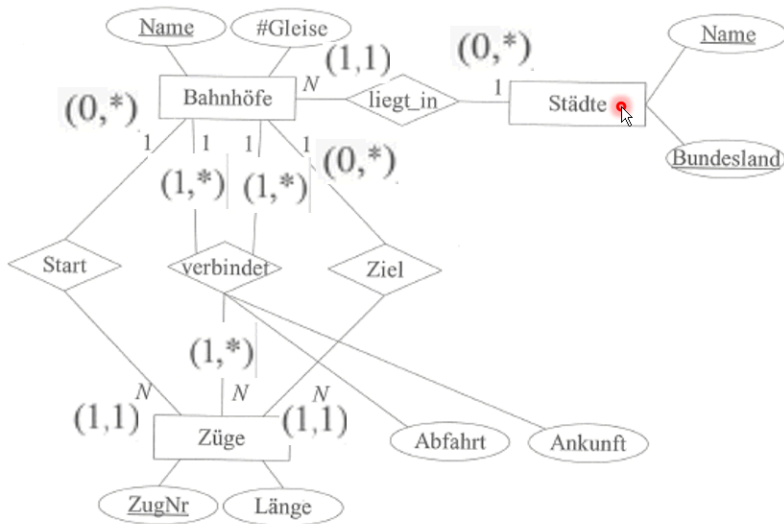




Aufgabe 3.1

Gegeben sei die ER-Modellierung von Zugverbindungen in Abbildung 3.1.

a) Fügen Sie bei den Beziehungen Kardinalitäten in der (min, max) -Notation hinzu.



b) Übertragen Sie das ER-Modell in ein relationales Schema.

Die initiale Überführung ergibt folgende Relationen für die Entitytypen:

Städte : {[Name : string, Bundesland : string]}



Aufgabe 3.1

Gegeben sei die ER-Modellierung von Zugverbindungen in Abbildung 3.1.

a) Fügen Sie bei den Beziehungen Kardinalitäten in der (min, max) -Notation hinzu.

