

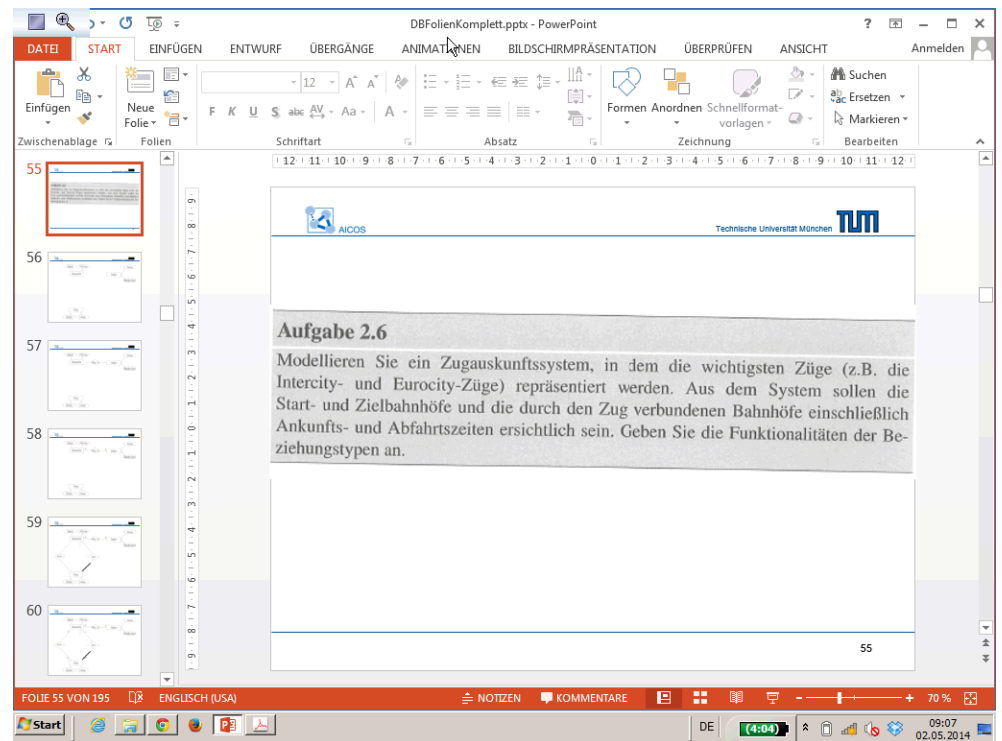
Script generated by TTT

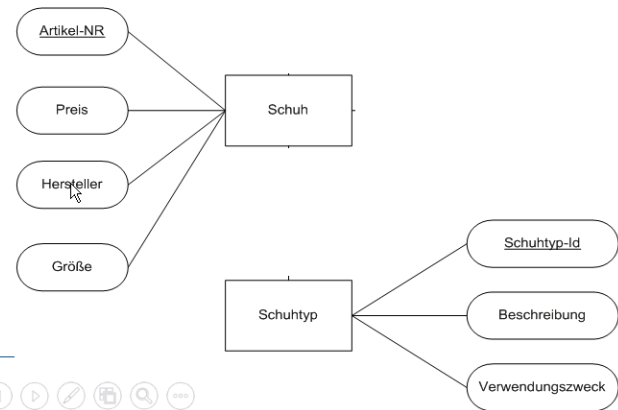
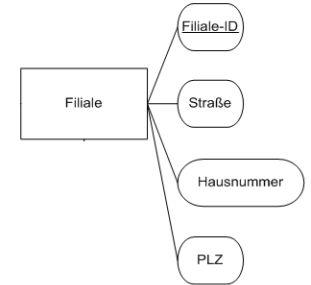
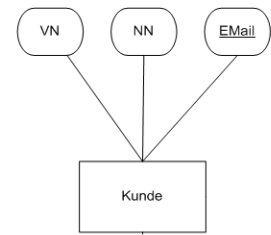
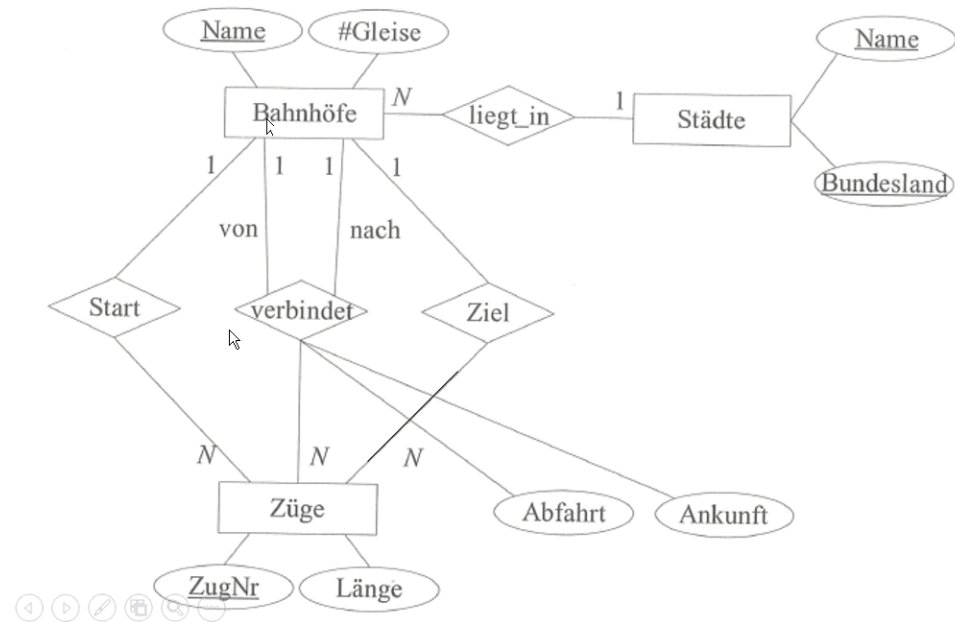
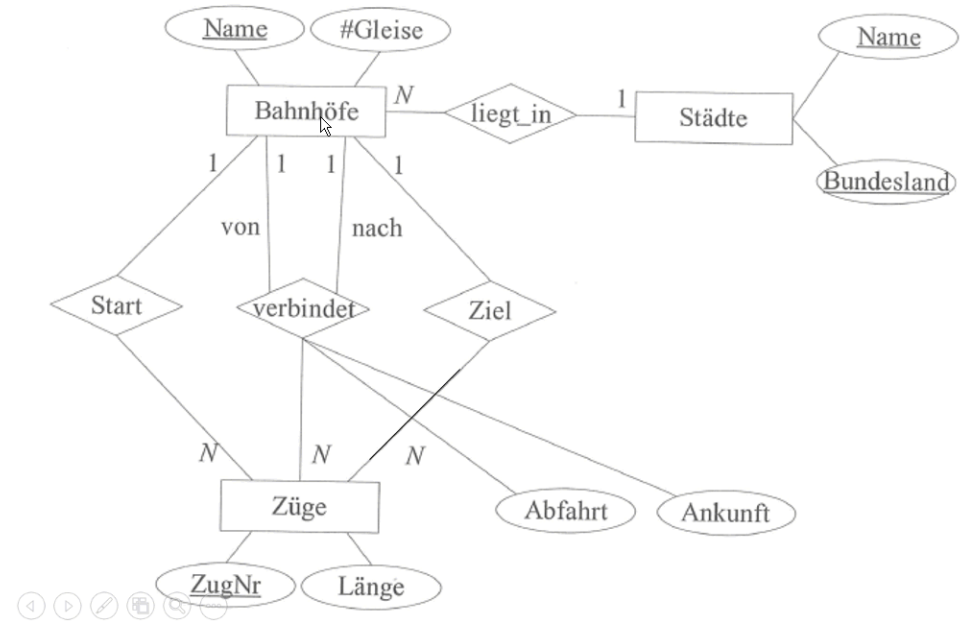
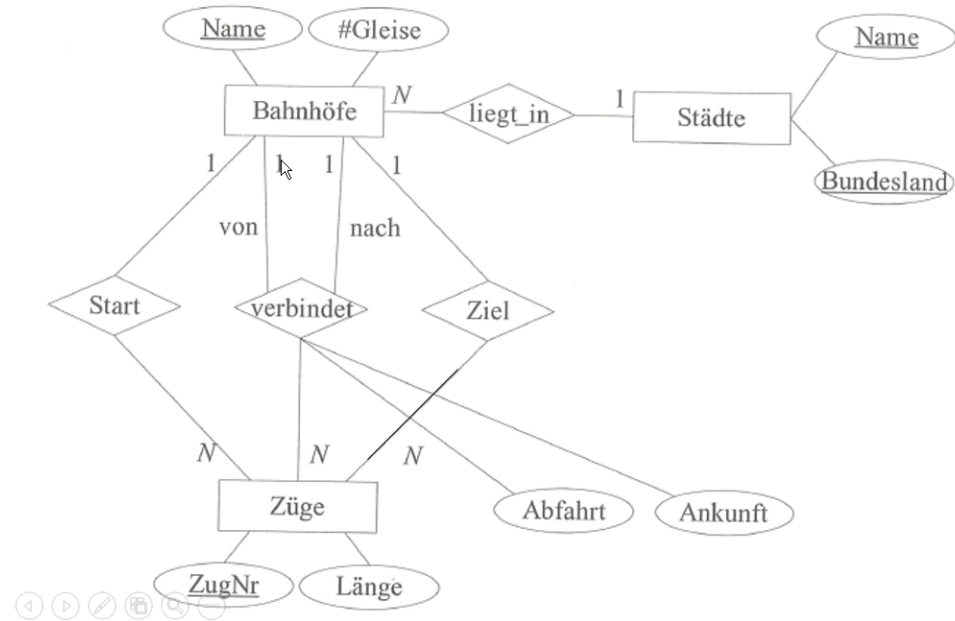
Title: groh: profile1 (02.05.2014)

Date: Fri May 02 09:07:57 CEST 2014

Duration: 94:15 min

Pages: 77





Grundlagen des relationalen Modells

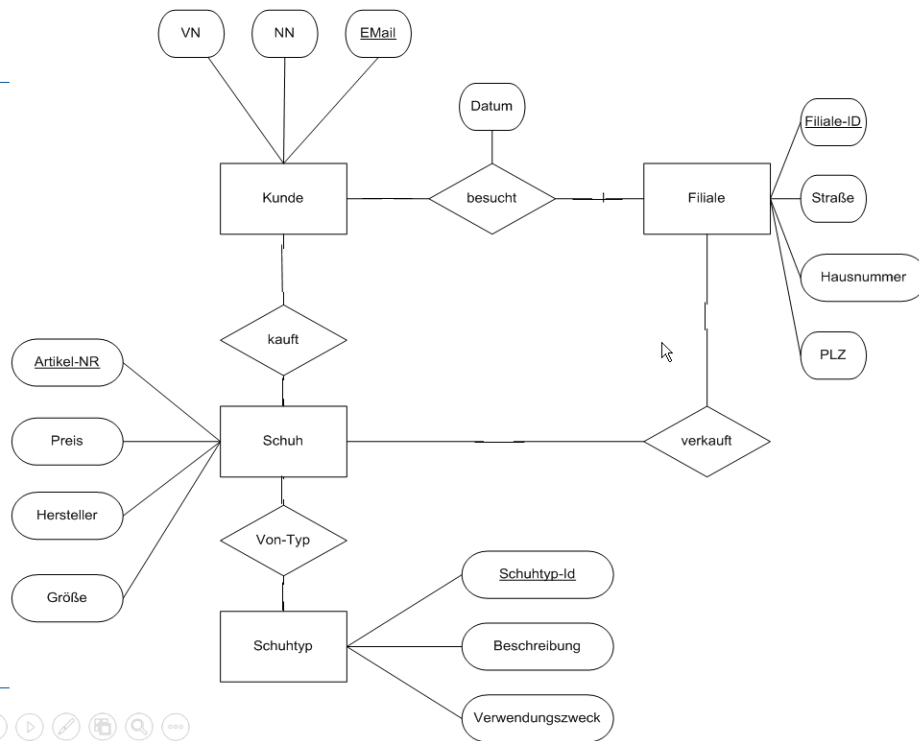
Seien D_1, D_2, \dots, D_n **Domänen** (~Wertebereiche)

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

Grundlagen des relationalen Modells

Seien D_1, D_2, \dots, D_n **Domänen** (~Wertebereiche)

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



Grundlagen des relationalen Modells

Seien D_1, D_2, \dots, D_n **Domänen** (~Wertebereiche)

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

Grundlagen des relationalen Modells

Seien D_1, D_2, \dots, D_n **Domänen** (\sim Wertebereiche)

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

Notationen:

- **Attribute:** A_1, A_2, \dots, A_n
- **Domänen :** D_1, D_2, \dots, D_n
 $== dom(A_1), dom(A_2), \dots, dom(A_n)$
- **Relation:** $R \subseteq D_1 \times D_2 \times \dots \times D_n$
 $= dom(A_1) \times dom(A_2) \times \dots \times dom(A_n)$
- **Schema:** $sch(R) == \mathcal{R} = \{ A_1, A_2, \dots, A_n \}$

Notationen:

- **Attribute:** A_1, A_2, \dots, A_n
- **Domänen :** D_1, D_2, \dots, D_n
 $== dom(A_1), dom(A_2), \dots, dom(A_n)$
- **Relation:** $R \subseteq D_1 \times D_2 \times \dots \times D_n$
 $= dom(A_1) \times dom(A_2) \times \dots \times dom(A_n)$
- **Schema:** $sch(R) == \mathcal{R} = \{ A_1, A_2, \dots, A_n \}$

Telefonbuch		
<u>Name:String</u>	<u>Straße:String</u>	<u>Telefon#:integer</u>
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

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

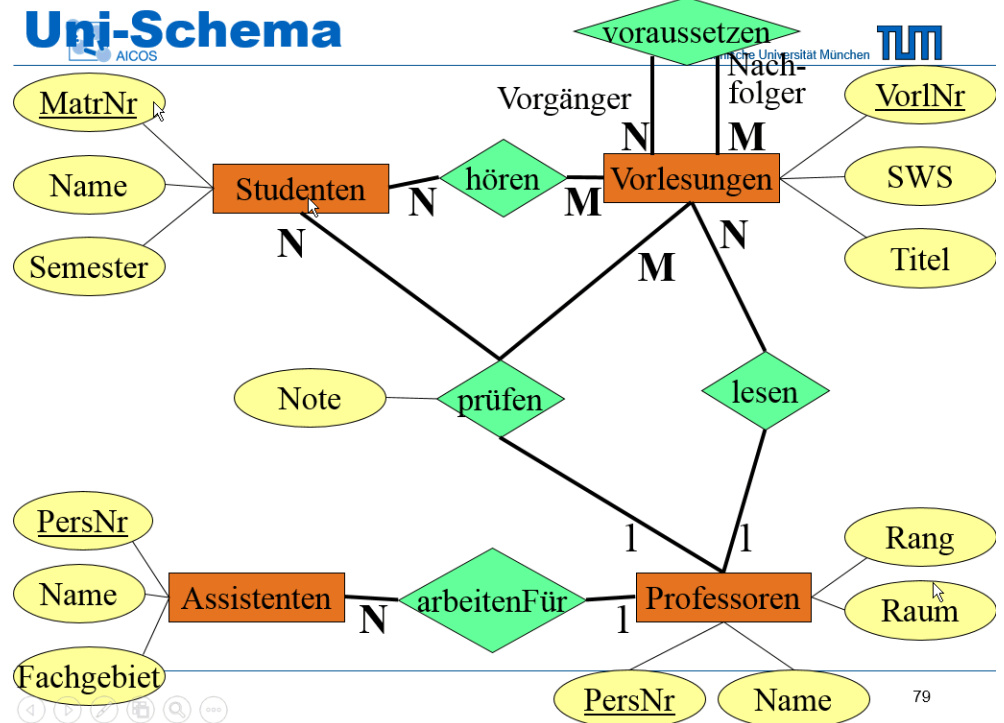
- **Ausprägung:** der aktuelle Zustand der Datenbasis
- **Schlüssel:** minimale Menge von Attributen, deren Werte ein Tupel eindeutig identifizieren
- **Primärschlüssel:** wird unterstrichen
 - Einer der Schlüsselkandidaten wird als Primärschlüssel ausgewählt
 - Hat eine besondere Bedeutung bei der Referenzierung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

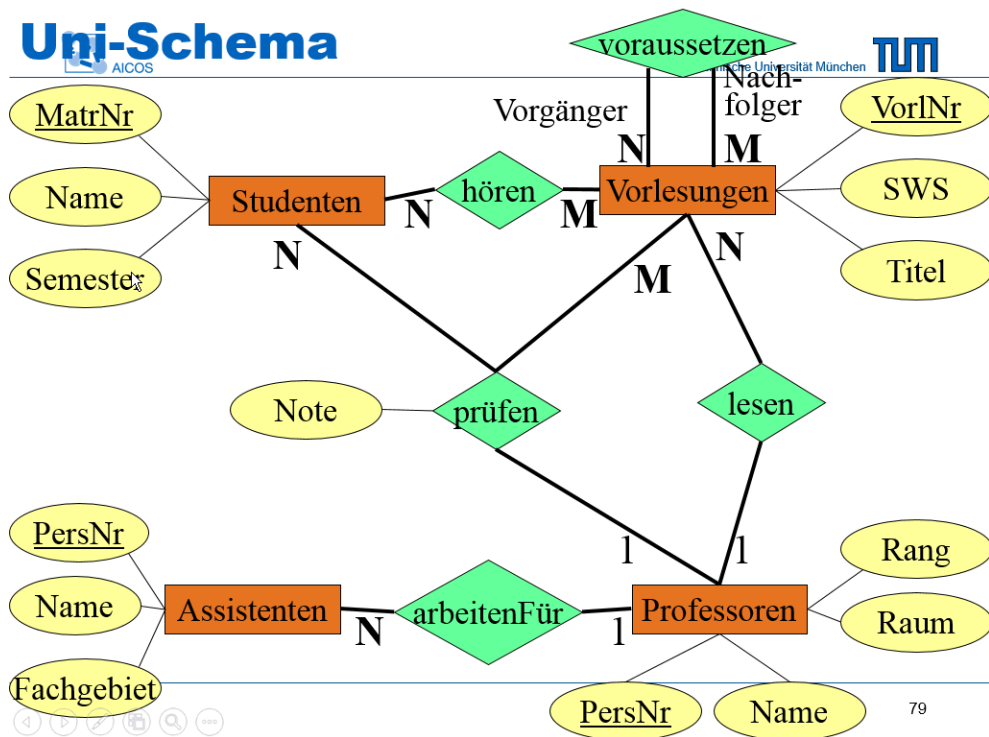
Telefonbuch		
Name:String	Straße:String	<u>Telefon#:integer</u>
Mickey Mouse	Main Street	4711
Donald Duck	Broadway	95672
...

- **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

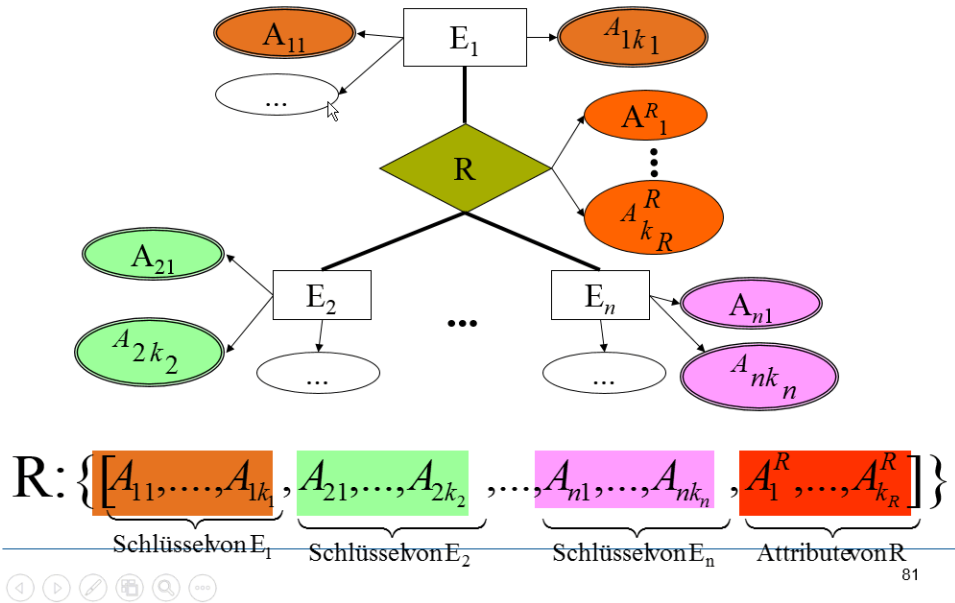
Telefonbuch		
Name:String	Straße:String	<u>Telefon#:integer</u>
Mickey Mouse	Main Street	4711
Donald Duck	Broadway	95672
...

- **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

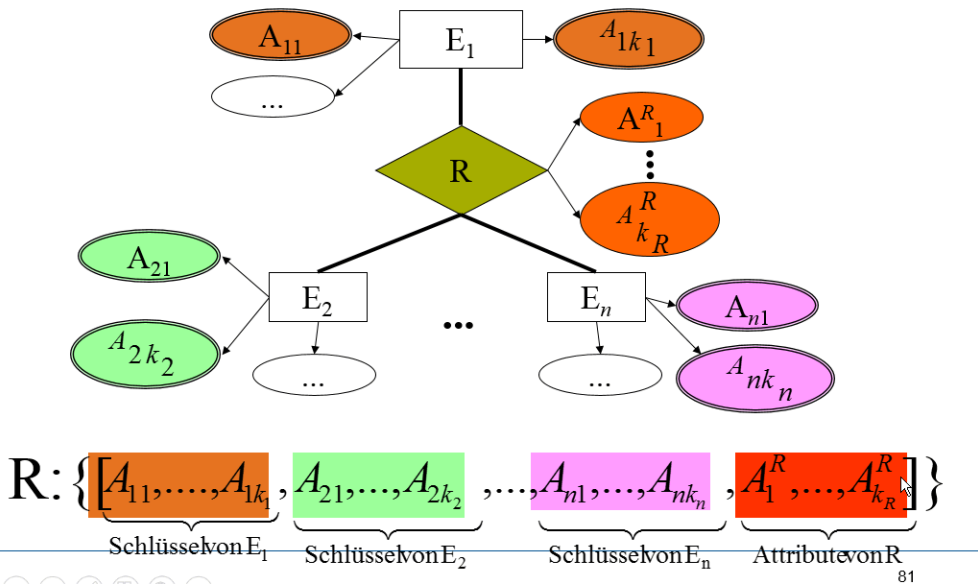




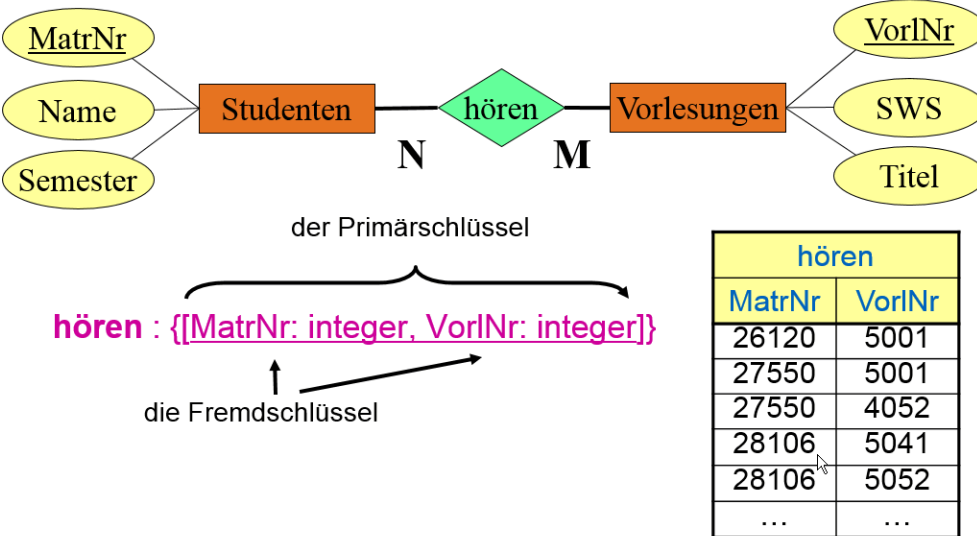
Relationale Darstellung von Beziehungen



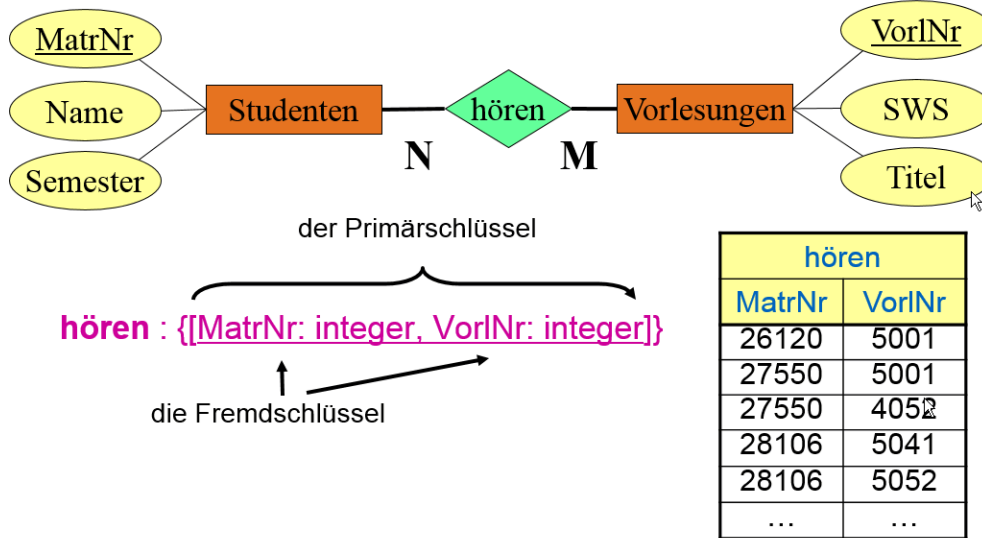
Relationale Darstellung von Beziehungen



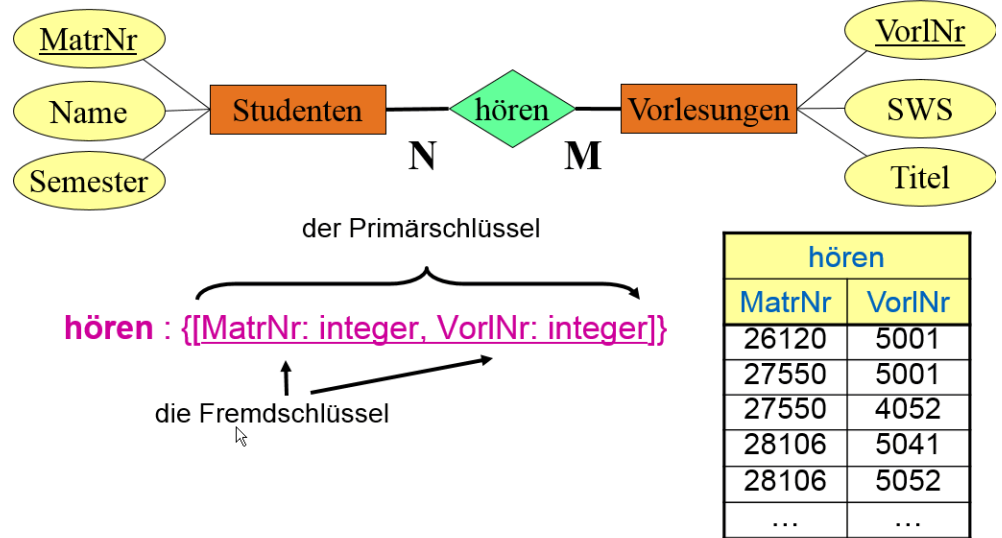
Beziehungen unseres Beispiel-Schemas



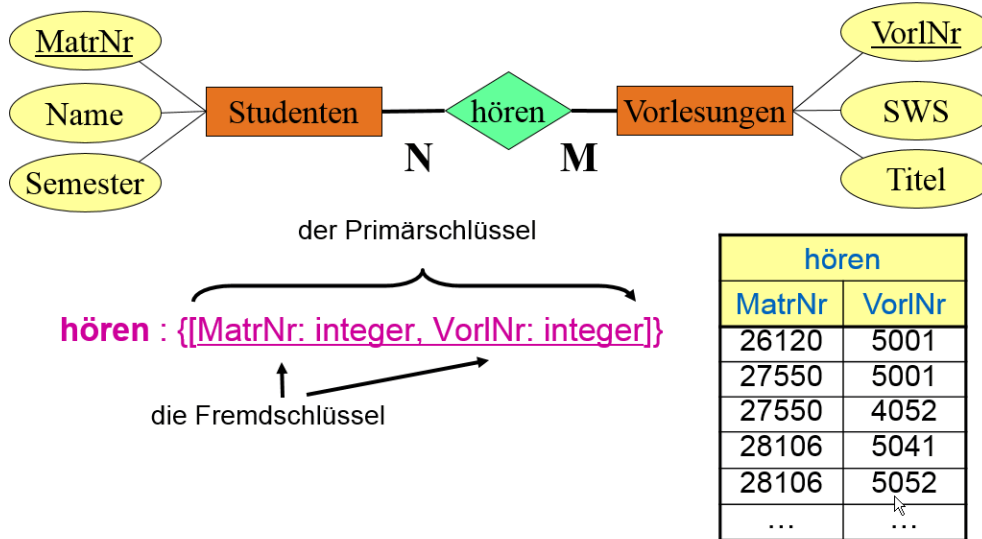
Beziehungen unseres Beispiel-Schemas



Beziehungen unseres Beispiel-Schemas



Beziehungen unseres Beispiel-Schemas



Beziehungen unseres Beispiel-Schemas



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

Nur das ist der Schlüssel !!! Der zugehörige Prof ist durch 1:N bereits eindeutig bestimmt

Beziehungen unseres Beispiel-Schemas



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

Nur das ist der Schlüssel !!! Der zugehörige Prof ist durch 1:N bereits eindeutig bestimmt

Beziehungen unseres Beispiel-Schemas



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

Nur das ist der Schlüssel !!! Der zugehörige Prof ist durch 1:N bereits eindeutig bestimmt

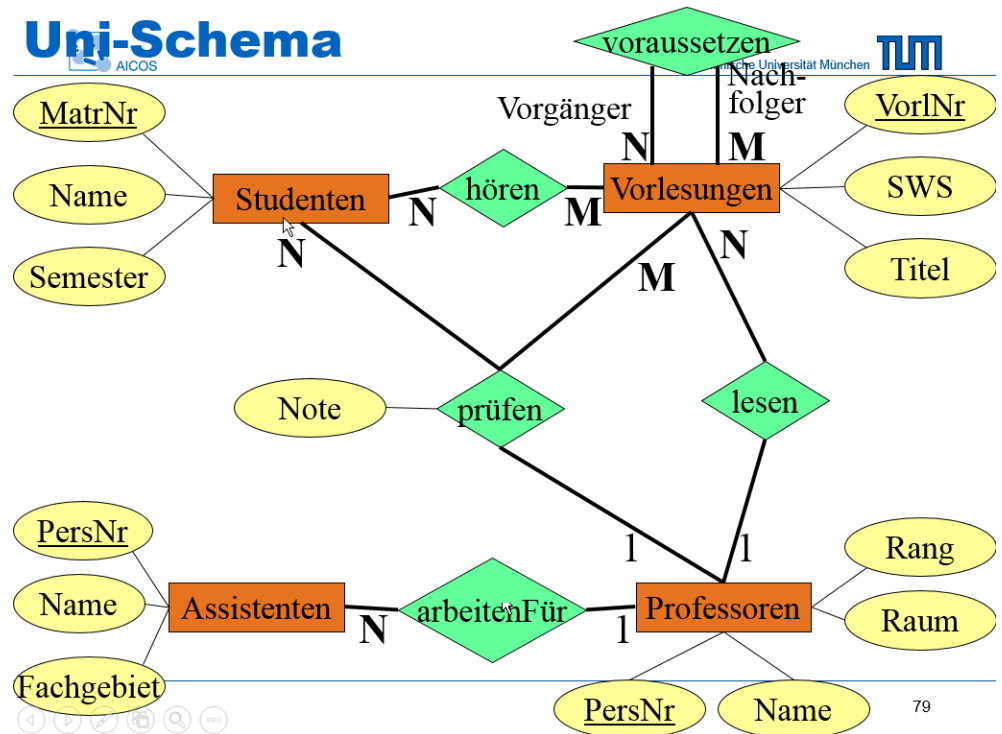
Beziehungen unseres Beispiel-Schemas



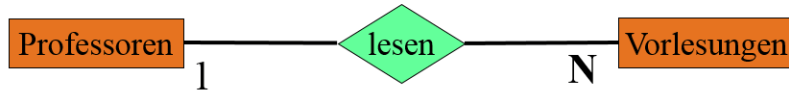
lesen : {[PersNr: integer, VorlNr: integer]}

Nur das ist der Schlüssel !!! Der zugehörige Prof ist durch 1:N bereits eindeutig bestimmt

Uni-Schema



Verfeinerung des relationalen Schemas



1:N-Beziehung sind verfeinerbar (gilt daher auch für 1:1)

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

Verfeinerung des relationalen Schemas



1:N-Beziehung sind verfeinerbar (gilt daher auch für 1:1)

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

Verfeinerung des relationalen Schemas

1:N-Beziehung (gilt somit auch für 1:1)

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

Regel

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

Verfeinerung des relationalen Schemas

1:N-Beziehung (gilt somit auch für 1:1)

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

Regel

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

Verfeinerung des relationalen Schemas

1:N-Beziehung (gilt somit auch für 1:1)

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

Regel

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

Vorsicht: So geht es NICHT !! → Anomalien

Professoren					Vorlesungen		
PersNr	Name	Rang	Raum	liest	VorlNr	Titel	SWS
2125	Sokrates	C4	226	5041	5041	Ethik	4
2125	Sokrates	C4	226	5049	5043	Erkenntnistheorie	3
2125	Sokrates	C4	226	4052	5049	Mäeutik	2
2134	Augustinus	C3	309	5022	4052	Logik	4
2136	Curie	C4	36	??	5052	Wissenschaftstheorie	3
...	5022	Glaube und Wissen	2
...

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

Vorsicht: So geht es NICHT !! → Anomalien

Professoren					Vorlesungen		
PersNr	Name	Rang	Raum	liest	VorlNr	Titel	SWS
2125	Sokrates	C4	226	5041	5041	Ethik	4
2125	Sokrates	C4	226	5049	5043	Erkenntnistheorie	3
2125	Sokrates	C4	226	4052	5049	Mäeutik	2
2134	Augustinus	C3	309	5022	4052	Logik	4
2136	Curie	C4	36	??	5052	Wissenschaftstheorie	3
...	5022	Glaube und Wissen	2
...

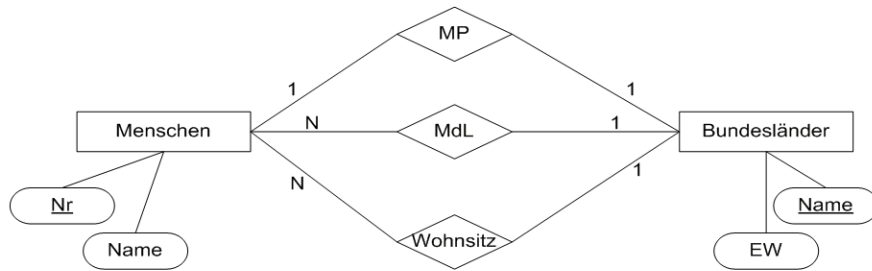
- Update-Anomalie: Was passiert wenn Sokrates umzieht
- Lösch-Anomalie: Was passiert wenn „Glaube und Wissen“ wegfällt
- Einfügeanomalie: Curie forscht nur und liest keine Vorlesung

Vorsicht: So geht es NICHT !! → Anomalien

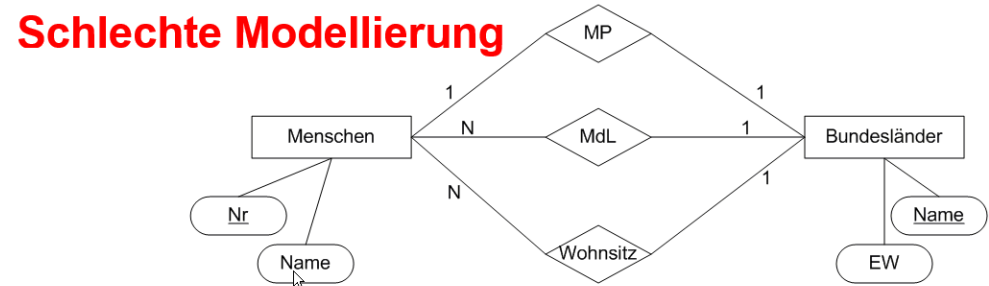
Professoren					Vorlesungen		
PersNr	Name	Rang	Raum	liest	VorlNr	Titel	SWS
2125	Sokrates	C4	226	5041	5041	Ethik	4
2125	Sokrates	C4	226	5049	5043	Erkenntnistheorie	3
2125	Sokrates	C4	226	4052	5049	Mäeutik	2
2134	Augustinus	C3	309	5022	4052	Logik	4
2136	Curie	C4	36	??	5052	Wissenschaftstheorie	3
...	5022	Glaube und Wissen	2
...

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

Vermeidung von Null-Werten

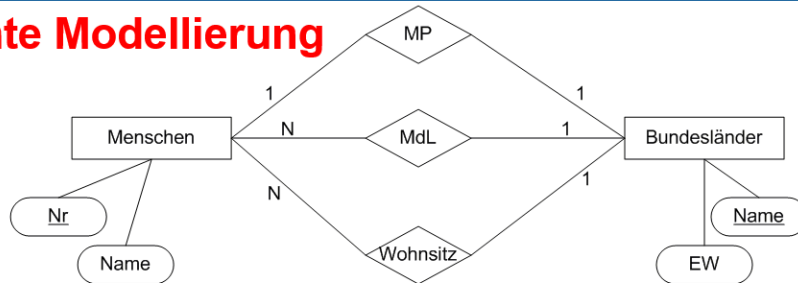


Schlechte Modellierung



Menschen				
Nr	Name	Wohnsitz	MPvon	MdLvon
4711	Kemper	Bayern	–	–
4813	Seehofer	Bayern	Bayern	Bayern
5833	Maget	Bayern	–	Bayern
6745	Platzeck	Brandenburg	Brandenburg	Brandenburg
8978	Wulff	Niedersachsen	–	–
...

Schlechte Modellierung



Bessere Modellierung

- der Wohnsitz kann als Fremdschlüssel in der Entity-Relation Mensch bleiben
- die Beziehung MP modelliert man am besten als Fremdschlüssel in Bundesländer, da alle BL einen MP haben
- die Beziehung MdL repräsentiert man als eigenständige Relation mit den Fremdschlüsseln Nr auf Mensch und Bundesland auf Bundesländer

Menschen				
Nr	Name	Wohnsitz	MPvon	MdLvon
4711	Kemper	Bayern	–	–
4813	Seehofer	Bayern	Bayern	Bayern
5833	Maget	Bayern	–	Bayern
6745	Platzeck	Brandenburg	Brandenburg	Brandenburg
8978	Wulff	Niedersachsen	–	–
...

Menschen			MdL		Bundesländer		
Nr	Name	Wohnsitz	Nr	Bundesland	Name	EW	MP
4711	Kemper	Bayern	4813	Bayern	Bayern	12443893	4813
4813	Seehofer	Bayern	5833	Bayern	Brandenburg	2562946	6745
5833	Maget	Bayern	6745	Brandenburg
6745	Platzeck	Brandenburg			
8978	Wulff	Niedersachsen					
...					

Bessere Modellierung

- der Wohnsitz kann als Fremdschlüssel in der Entity-Relation Mensch bleiben
- die Beziehung MP modelliert man am besten als Fremdschlüssel in Bundesländer, da alle BL einen MP haben
- die Beziehung MdL repräsentiert man als eigenständige Relation mit den Fremdschlüsseln Nr auf Mensch und Bundesland auf Bundesländer

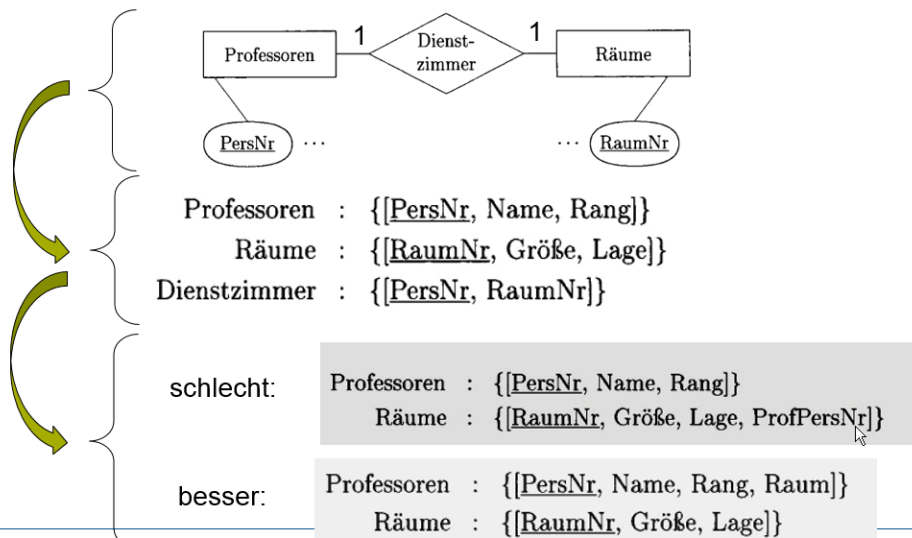
Menschen			MdL		Bundesländer		
Nr	Name	Wohnsitz	Nr	Bundesland	Name	EW	MP
4711	Kemper	Bayern	4813	Bayern	Bayern	12443893	4813
4813	Seehofer	Bayern	5833	Bayern	Brandenburg	2562946	6745
5833	Maget	Bayern	6745	Brandenburg
6745	Platzek	Brandenburg			
8978	Wulff	Niedersachsen					
...					

Bessere Modellierung

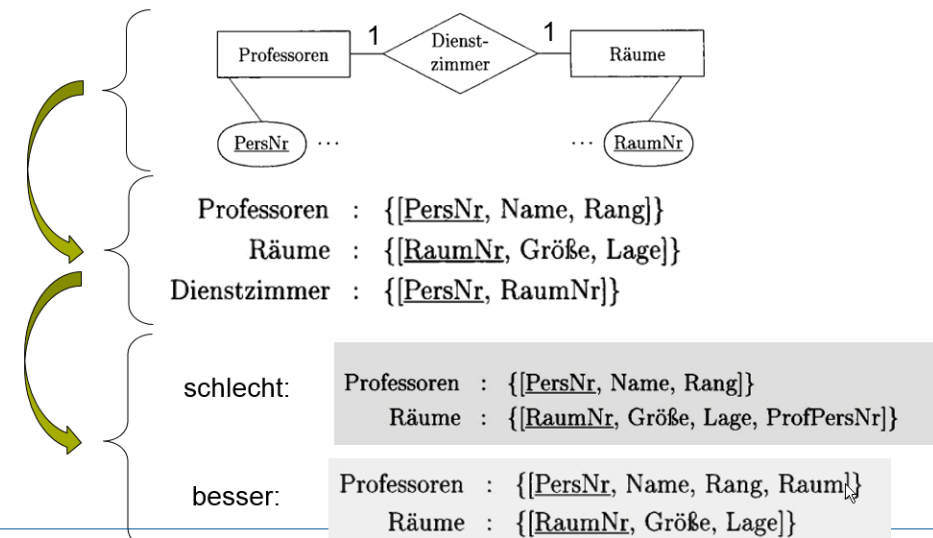
- der Wohnsitz kann als Fremdschlüssel in der Entity-Relation Mensch bleiben
- die Beziehung MP modelliert man am besten als Fremdschlüssel in Bundesländer, da alle BL einen MP haben
- die Beziehung MdL repräsentiert man als eigenständige Relation mit den Fremdschlüsseln Nr auf Mensch und Bundesland auf Bundesländer

Menschen			MdL		Bundesländer		
Nr	Name	Wohnsitz	Nr	Bundesland	Name	EW	MP
4711	Kemper	Bayern	4813	Bayern	Bayern	12443893	4813
4813	Seehofer	Bayern	5833	Bayern	Brandenburg	2562946	6745
5833	Maget	Bayern	6745	Brandenburg
6745	Platzek	Brandenburg			
8978	Wulff	Niedersachsen					
...					

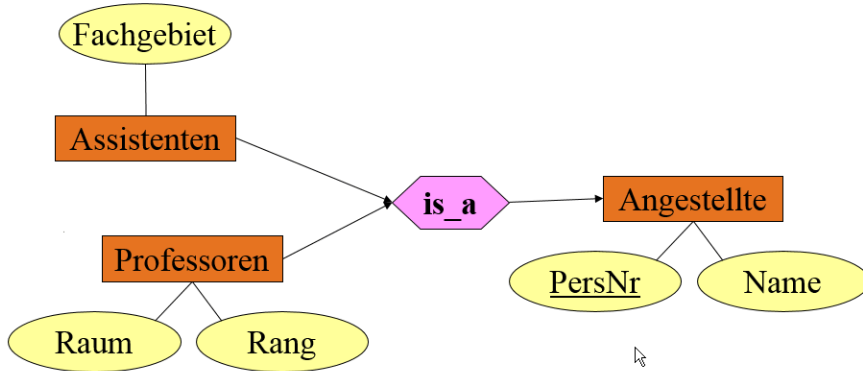
Vermeidung von Nullwerten Bsp 2



Vermeidung von Nullwerten Bsp 2



Relationale Modellierung der Generalisierung

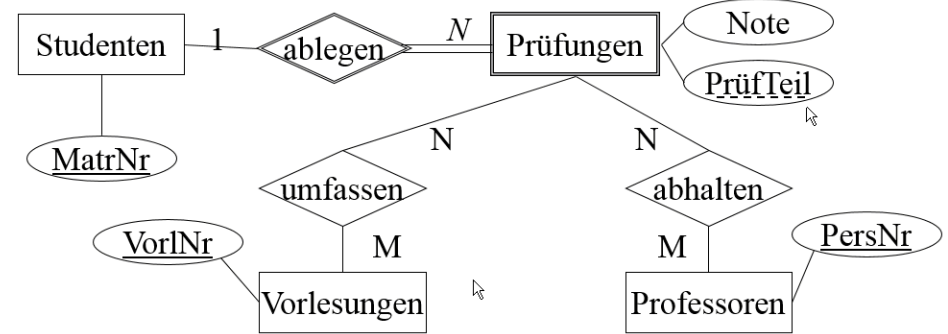


Angestellte: {[PersNr, Name]}

Professoren: {[PersNr, Rang, Raum]}

Assistenten: {[PersNr, Fachgebiet]}

Relationale Modellierung schwacher Entitytypen

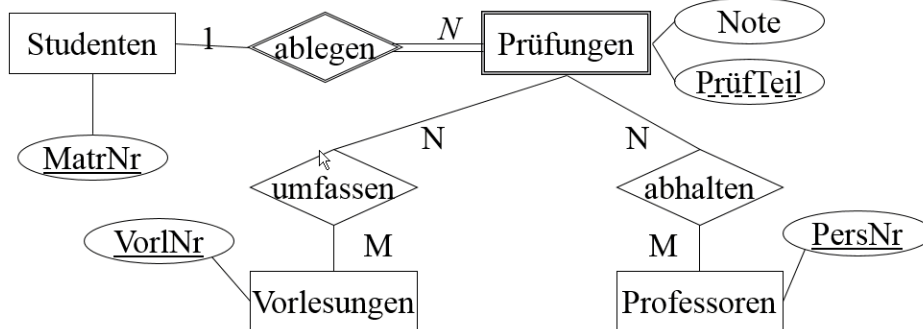


Prüfungen: {[MatrNr: integer, PrüfTeil: string, Note: integer]}

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

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

Relationale Modellierung schwacher Entitytypen

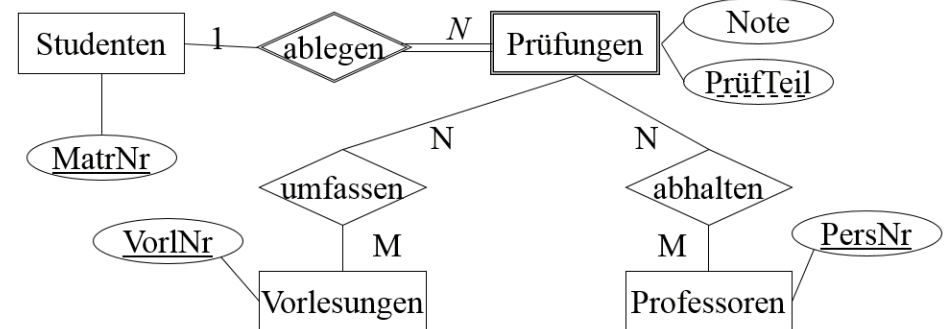


Prüfungen: {[MatrNr: integer, PrüfTeil: string, Note: integer]}

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

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

Relationale Modellierung schwacher Entitytypen

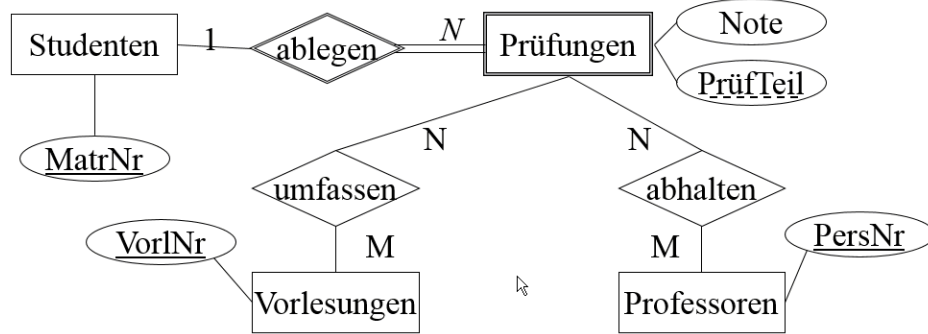


Prüfungen: {[MatrNr: integer, PrüfTeil: string, Note: integer]}

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

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

Relationale Modellierung schwacher Entitytypen



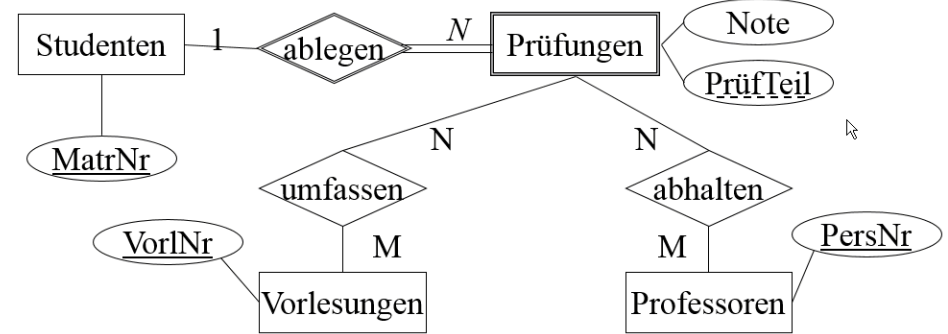
Prüfungen: {[MatrNr: integer, PrüfTeil: string, Note: integer]}

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

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



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]}



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	Feyerbach	2

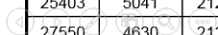
Vorlesungen			
VorlNr	Titel	SWS	gelesen von
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137

Assistenten			
PersNr	Name	Fachgebiet	Boss
3002	Platon	Ideenlehre	2125
3003	Aristoteles	Sylogistik	2125
3004	Wittgenstein	Sprachtheorie	2126
3005	Rhetikus	Planetarbewegung	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

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

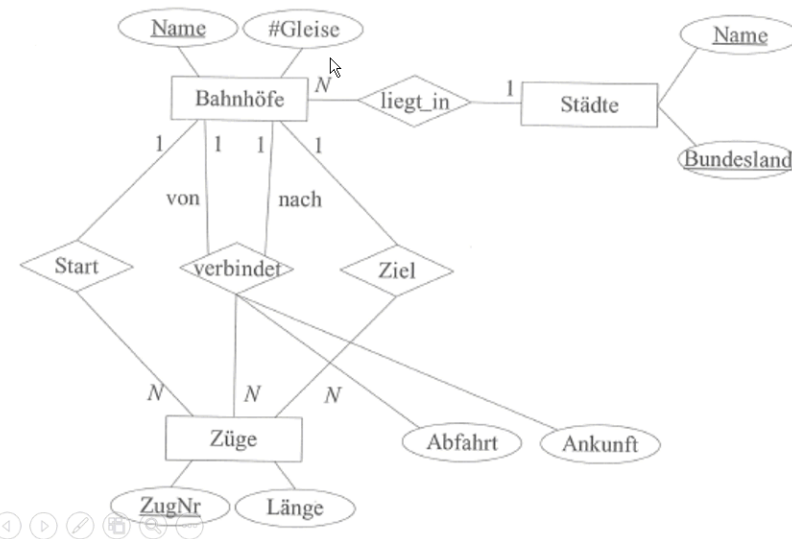
prüfen			
MatrNr	VorlNr	PersNr	Note
28106	5001	2126	1
25403	5041	2125	2
27550	4630	2137	2



Aufgabe 3.1

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

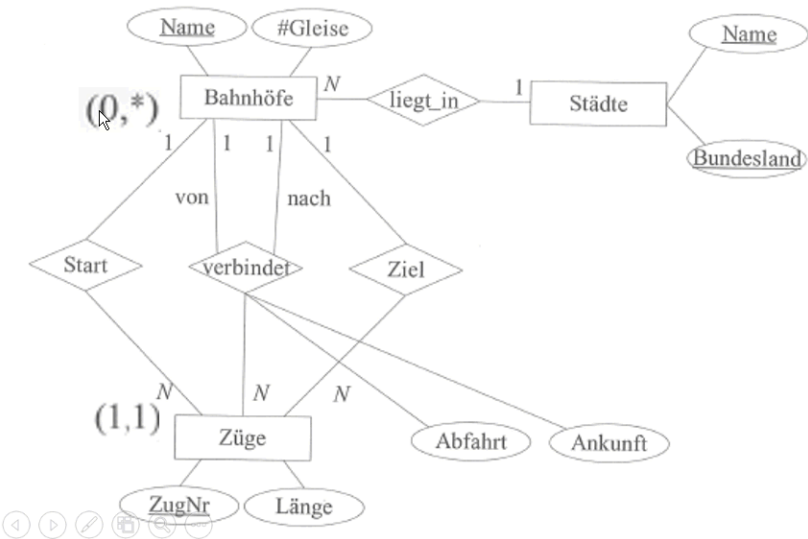
a) Fügen Sie bei den Beziehungen Kardinalitäten in der (min, max)-Notation hinzu.



Aufgabe 3.1

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

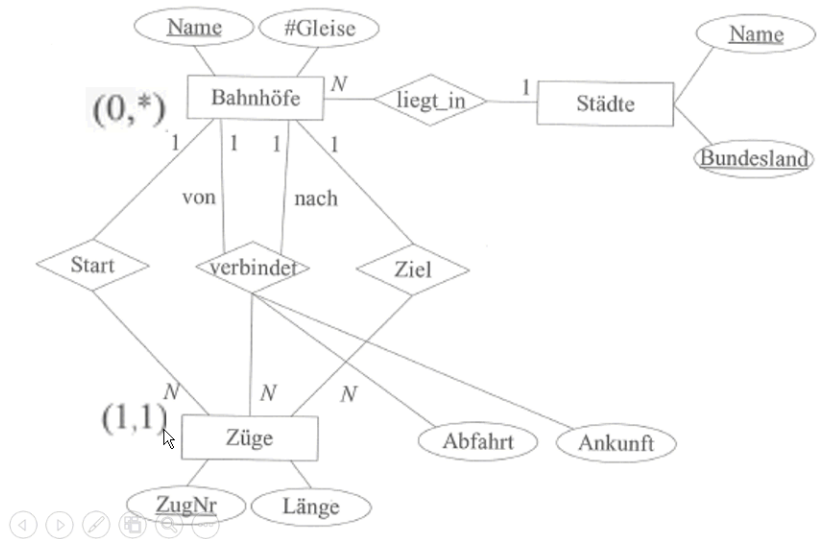
a) Fügen Sie bei den Beziehungen Kardinalitäten in der (min, max) -Notation hinzu.



Aufgabe 3.1

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

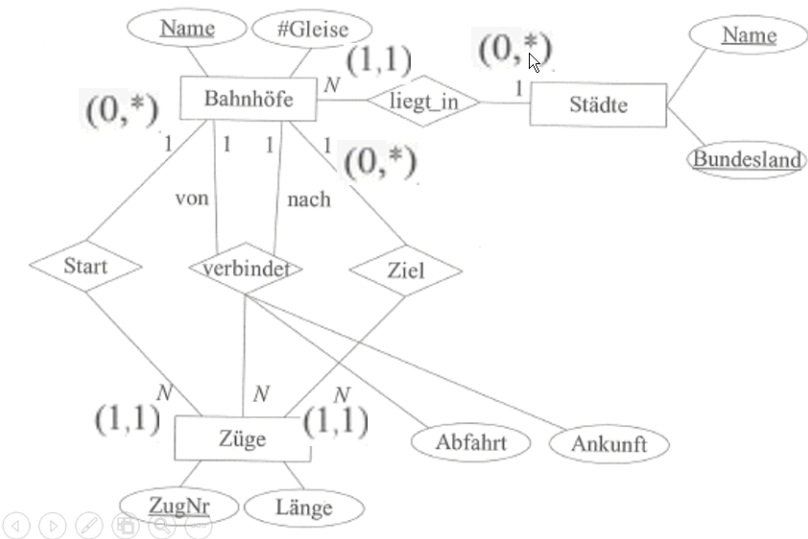
a) Fügen Sie bei den Beziehungen Kardinalitäten in der (min, max) -Notation hinzu.



Aufgabe 3.1

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

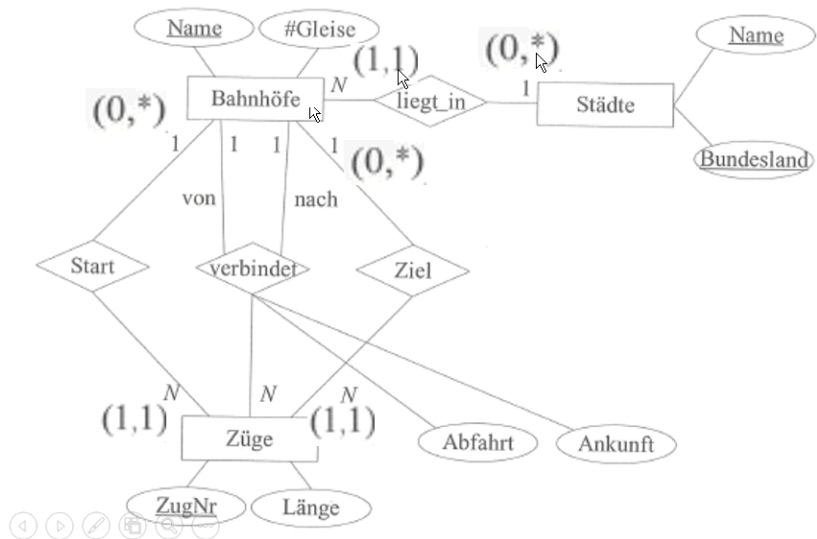
a) Fügen Sie bei den Beziehungen Kardinalitäten in der (min, max) -Notation hinzu.



Aufgabe 3.1

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

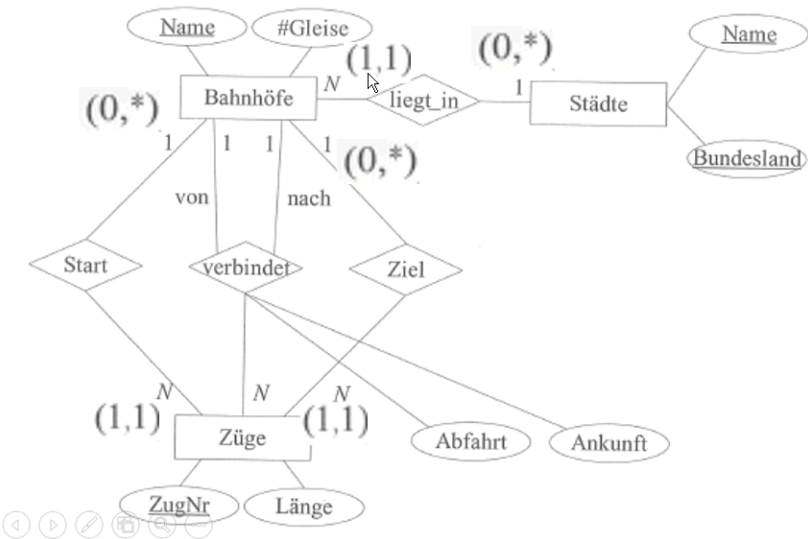
a) Fügen Sie bei den Beziehungen Kardinalitäten in der (min, max) -Notation hinzu.



Aufgabe 3.1

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

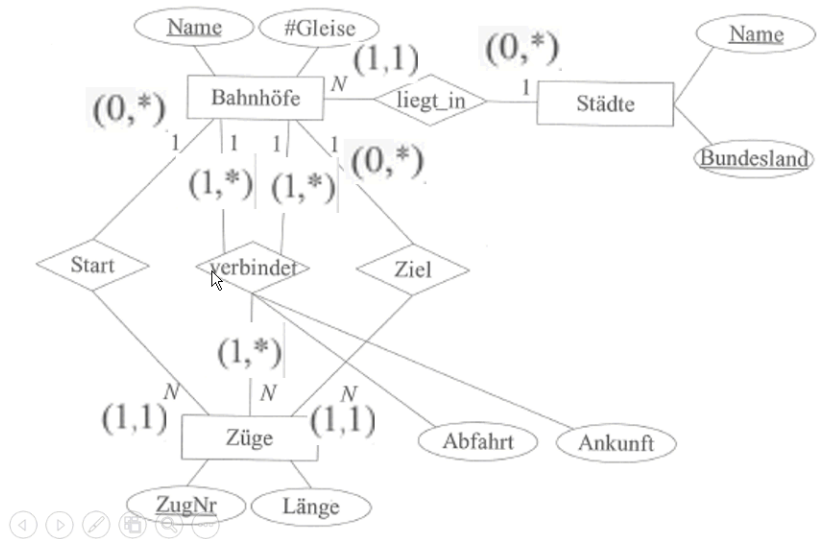
a) Fügen Sie bei den Beziehungen Kardinalitäten in der (min, max) -Notation hinzu.



Aufgabe 3.1

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

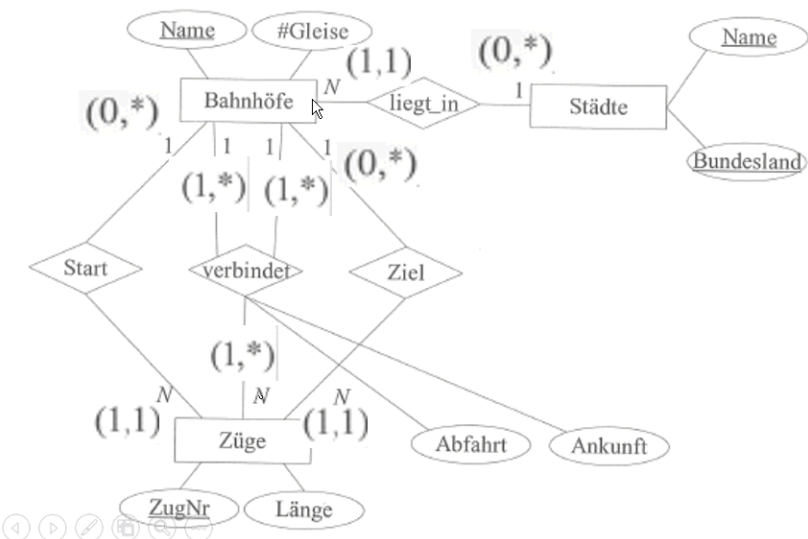
a) Fügen Sie bei den Beziehungen Kardinalitäten in der (min, max) -Notation hinzu.



Aufgabe 3.1

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

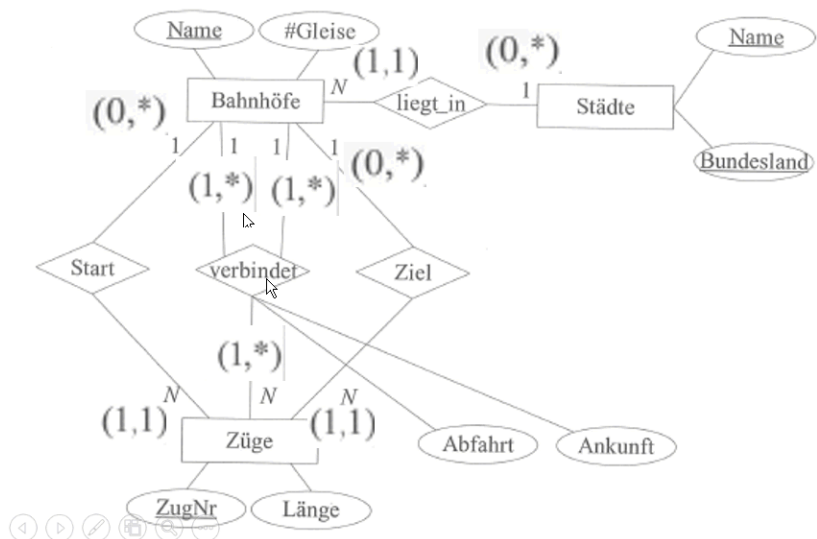
a) Fügen Sie bei den Beziehungen Kardinalitäten in der (min, max) -Notation hinzu.



Aufgabe 3.1

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

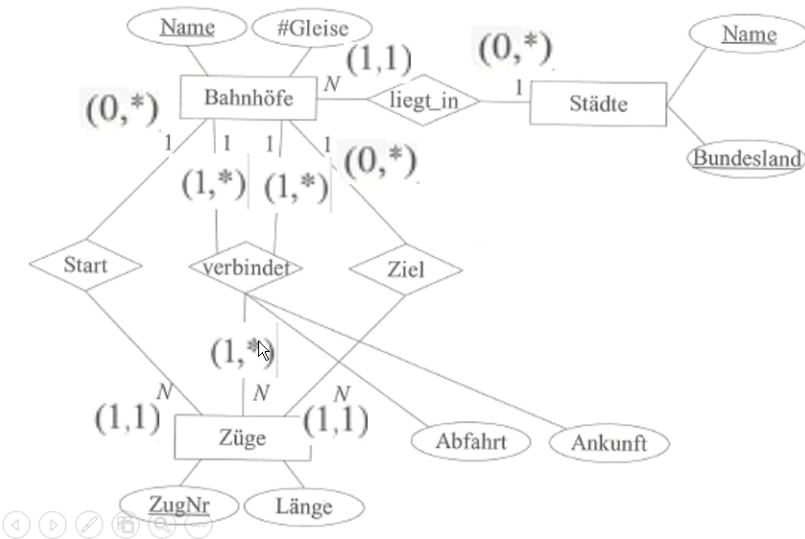
a) Fügen Sie bei den Beziehungen Kardinalitäten in der (min, max) -Notation hinzu.



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.



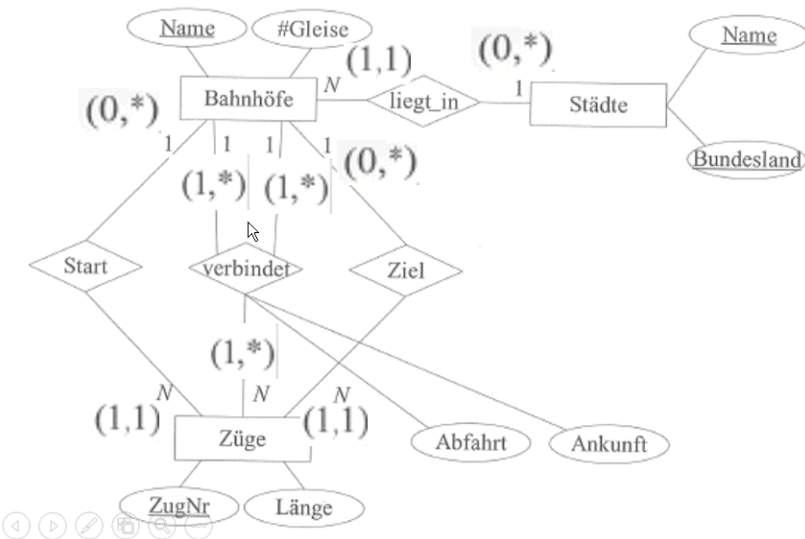
b) Übertragen Sie das ER-Modell in ein relationales Schema.

Die initiale Überführung ergibt folgende Relationen für die Entitytypen:

Aufgabe 3.1

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

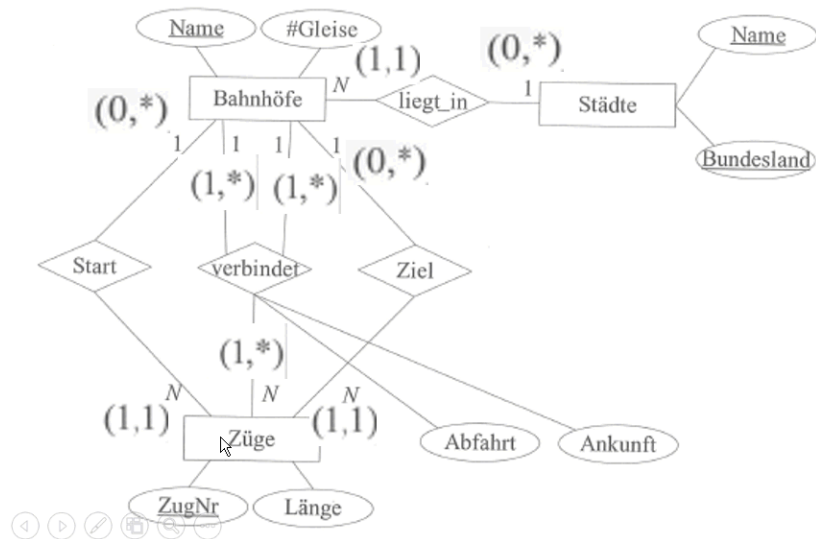
a) Fügen Sie bei den Beziehungen Kardinalitäten in der (min, max) -Notation hinzu.



Aufgabe 3.1

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

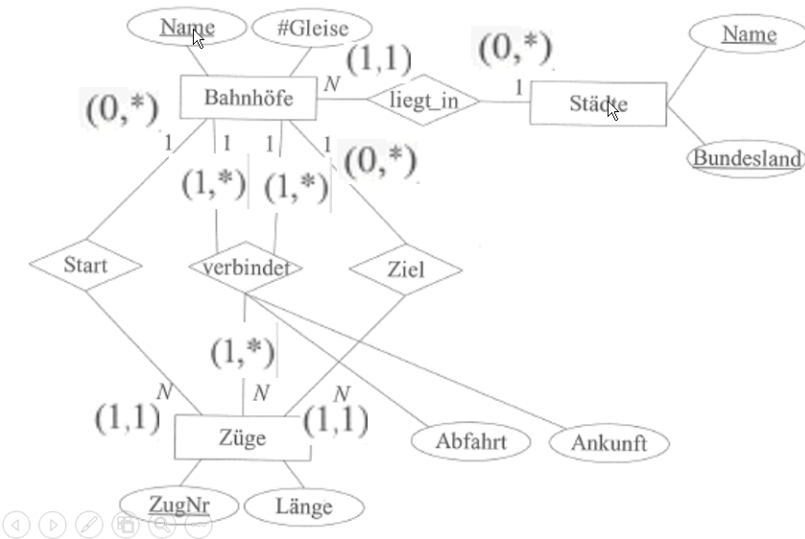
a) Fügen Sie bei den Beziehungen Kardinalitäten in der (min, max) -Notation hinzu.



Aufgabe 3.1

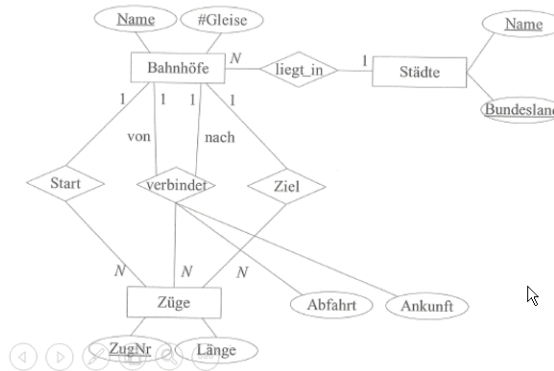
Gegeben sei die ER-Modellierung von Zugverbindungen in Abbildung 3.1.

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



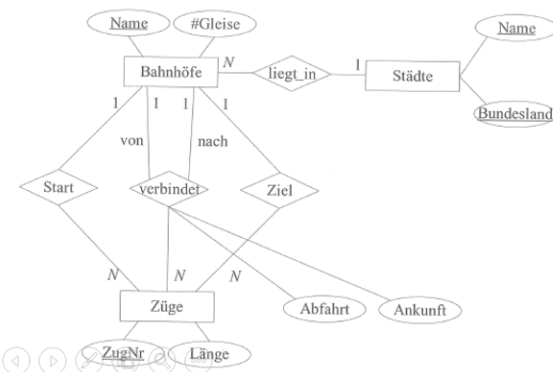
c) Verfeinern Sie das relationale Schema soweit möglich durch Eliminierung von Relationen.

- Städte : {[Name : string, Bundesland : string]} (1)
- Bahnhöfe : {[Name : string, #Gleise : integer]} (2)
- Züge : {[ZugNr : integer, Länge : integer]} (3)
- liegt_in : {[BName : string, SName : string, Bundesland : string]} (4)
- Start : {[ZugNr : integer, BName : string]} (5)
- Ziel : {[ZugNr : integer, BName : string]} (6)
- verbindet : {[VonBahnhof : string, NachBahnhof : string, ZugNr : integer, Abfahrt : date, Ankunft : date]} (7)



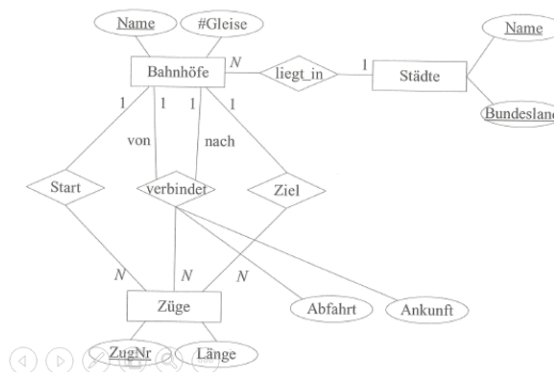
c) Verfeinern Sie das relationale Schema soweit möglich durch Eliminierung von Relationen.

- Städte : {[Name : string, Bundesland : string]} (1)
- Bahnhöfe : {[Name : string, #Gleise : integer]} (2)
- Züge : {[ZugNr : integer, Länge : integer]} (3)
- liegt_in : {[BName : string, SName : string, Bundesland : string]} (4)
- Start : {[ZugNr : integer, BName : string]} (5)
- Ziel : {[ZugNr : integer, BName : string]} (6)
- verbindet : {[VonBahnhof : string, NachBahnhof : string, ZugNr : integer, Abfahrt : date, Ankunft : date]} (7)



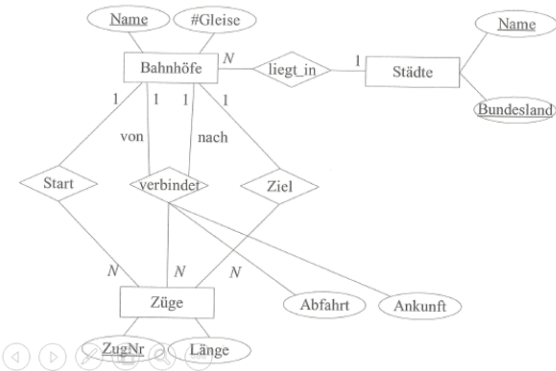
c) Verfeinern Sie das relationale Schema soweit möglich durch Eliminierung von Relationen.

- Städte : {[Name : string, Bundesland : string]} (1)
- Bahnhöfe : {[Name : string, #Gleise : integer]} (2)
- Züge : {[ZugNr : integer, Länge : integer]} (3)
- liegt_in : {[BName : string, SName : string, Bundesland : string]} (4)
- Start : {[ZugNr : integer, BName : string]} (5)
- Ziel : {[ZugNr : integer, BName : string]} (6)
- verbindet : {[VonBahnhof : string, NachBahnhof : string, ZugNr : integer, Abfahrt : date, Ankunft : date]} (7)



c) Verfeinern Sie das relationale Schema soweit möglich durch Eliminierung von Relationen.

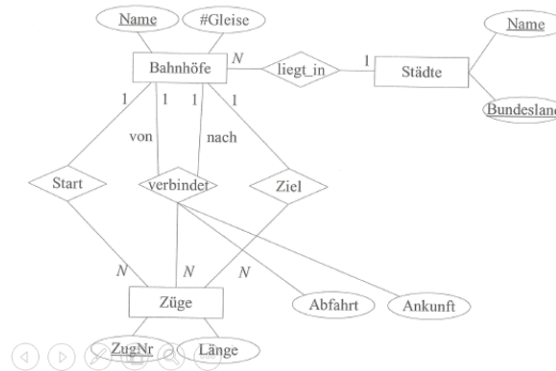
- Städte : {[Name : string, Bundesland : string]} (1)
- Bahnhöfe : {[Name : string, #Gleise : integer]} (2)
- Züge : {[ZugNr : integer, Länge : integer]} (3)
- liegt_in : {[BName : string, SName : string, Bundesland : string]} (4)
- Start : {[ZugNr : integer, BName : string]} (5)
- Ziel : {[ZugNr : integer, BName : string]} (6)
- verbindet : {[VonBahnhof : string, NachBahnhof : string, ZugNr : integer, Abfahrt : date, Ankunft : date]} (7)



111

c) Verfeinern Sie das relationale Schema soweit möglich durch Eliminierung von Relationen.

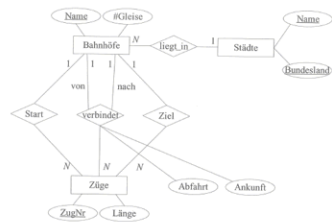
- Städte : {[Name : string, Bundesland : string]} (1)
- Bahnhöfe : {[Name : string, #Gleise : integer]} (2)
- Züge : {[ZugNr : integer, Länge : integer]} (3)
- liegt_in : {[BName : string, SName : string, Bundesland : string]} (4)
- Start : {[ZugNr : integer, BName : string]} (5)
- Ziel : {[ZugNr : integer, BName : string]} (6)
- verbindet : {[VonBahnhof : string, NachBahnhof : string, ZugNr : integer, Abfahrt : date, Ankunft : date]} (7)



111

c) Verfeinern Sie das relationale Schema soweit möglich durch Eliminierung von Relationen.

- Städte : {[Name : string, Bundesland : string]} (1)
- Bahnhöfe : {[Name : string, #Gleise : integer]} (2)
- Züge : {[ZugNr : integer, Länge : integer]} (3)
- liegt_in : {[BName : string, SName : string, Bundesland : string]} (4)
- Start : {[ZugNr : integer, BName : string]} (5)
- Ziel : {[ZugNr : integer, BName : string]} (6)
- verbindet : {[VonBahnhof : string, NachBahnhof : string, ZugNr : integer, Abfahrt : date, Ankunft : date]} (7)



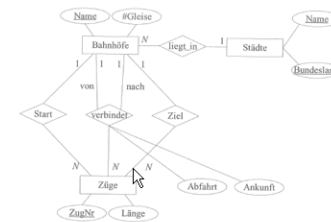
(4) \mapsto (2), (5) \mapsto (3), (6) \mapsto (3)

Damit ergibt sich folgendes Schema:

- Städte : {[Name : string, Bundesland : string]}
- Bahnhöfe : {[Name : string, #Gleise : integer, SName : string, Bundesland : string]}
- Züge : {[ZugNr : integer, Länge : integer, StartBahnhof : string, ZielBahnhof : string]}
- verbindet : {[VonBahnhof : string, NachBahnhof : string, ZugNr : integer, Abfahrt : date, Ankunft : date]}

c) Verfeinern Sie das relationale Schema soweit möglich durch Eliminierung von Relationen.

- Städte : {[Name : string, Bundesland : string]} (1)
- Bahnhöfe : {[Name : string, #Gleise : integer]} (2)
- Züge : {[ZugNr : integer, Länge : integer]} (3)
- liegt_in : {[BName : string, SName : string, Bundesland : string]} (4)
- Start : {[ZugNr : integer, BName : string]} (5)
- Ziel : {[ZugNr : integer, BName : string]} (6)
- verbindet : {[VonBahnhof : string, NachBahnhof : string, ZugNr : integer, Abfahrt : date, Ankunft : date]} (7)

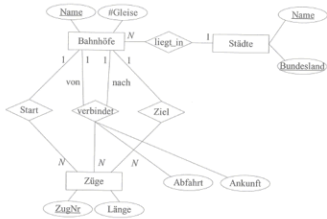


(4) \mapsto (2), (5) \mapsto (3), (6) \mapsto (3)

Damit ergibt sich folgendes Schema:

- Städte : {[Name : string, Bundesland : string]}
- Bahnhöfe : {[Name : string, #Gleise : integer, SName : string, Bundesland : string]}
- Züge : {[ZugNr : integer, Länge : integer, StartBahnhof : string, ZielBahnhof : string]}
- verbindet : {[VonBahnhof : string, NachBahnhof : string, ZugNr : integer, Abfahrt : date, Ankunft : date]}

c) Verfeinern Sie das relationale Schema soweit möglich durch Eliminierung von Relationen.



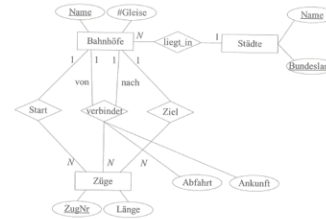
- (1) Städte : {[Name : string, Bundesland : string]}
- (2) Bahnhöfe : {[Name : string, #Gleise : integer]}
- (3) Züge : {[ZugNr : integer, Länge : integer]}
- (4) liegt_in : {[BName : string, SName : string, Bundesland : string]}
- (5) Start : {[ZugNr : integer, BName : string]}
- (6) Ziel : {[ZugNr : integer, BName : string]}
- (7) verbindet : {[VonBahnhof : string, NachBahnhof : string, ZugNr : integer, Abfahrt : date, Ankunft : date]}

(4) \mapsto (2), (5) \mapsto (3), (6) \mapsto (3)

Damit ergibt sich folgendes Schema:

- Städte : {[Name : string, Bundesland : string]}
- Bahnhöfe : {[Name : string, #Gleise : integer, SName : string, Bundesland : string]}
- Züge : {[ZugNr : integer, Länge : integer, StartBahnhof : string, ZielBahnhof : string]}
- verbindet : {[VonBahnhof : string, NachBahnhof : string, ZugNr : integer, Abfahrt : date, Ankunft : date]}

c) Verfeinern Sie das relationale Schema soweit möglich durch Eliminierung von Relationen.



- (1) Städte : {[Name : string, Bundesland : string]}
- (2) Bahnhöfe : {[Name : string, #Gleise : integer]}
- (3) Züge : {[ZugNr : integer, Länge : integer]}
- (4) liegt_in : {[BName : string, SName : string, Bundesland : string]}
- (5) Start : {[ZugNr : integer, BName : string]}
- (6) Ziel : {[ZugNr : integer, BName : string]}
- (7) verbindet : {[VonBahnhof : string, NachBahnhof : string, ZugNr : integer, Abfahrt : date, Ankunft : date]}

(4) \mapsto (2), (5) \mapsto (3), (6) \mapsto (3)

Damit ergibt sich folgendes Schema:

- Städte : {[Name : string, Bundesland : string]}
- Bahnhöfe : {[Name : string, #Gleise : integer, SName : string, Bundesland : string]}
- Züge : {[ZugNr : integer, Länge : integer, StartBahnhof : string, ZielBahnhof : string]}
- verbindet : {[VonBahnhof : string, NachBahnhof : string, ZugNr : integer, Abfahrt : date, Ankunft : date]}

