

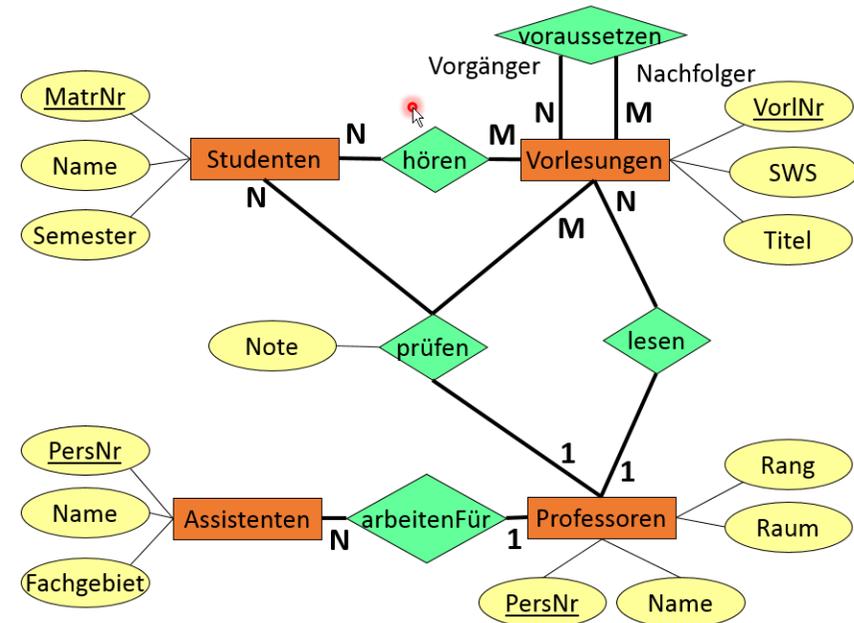
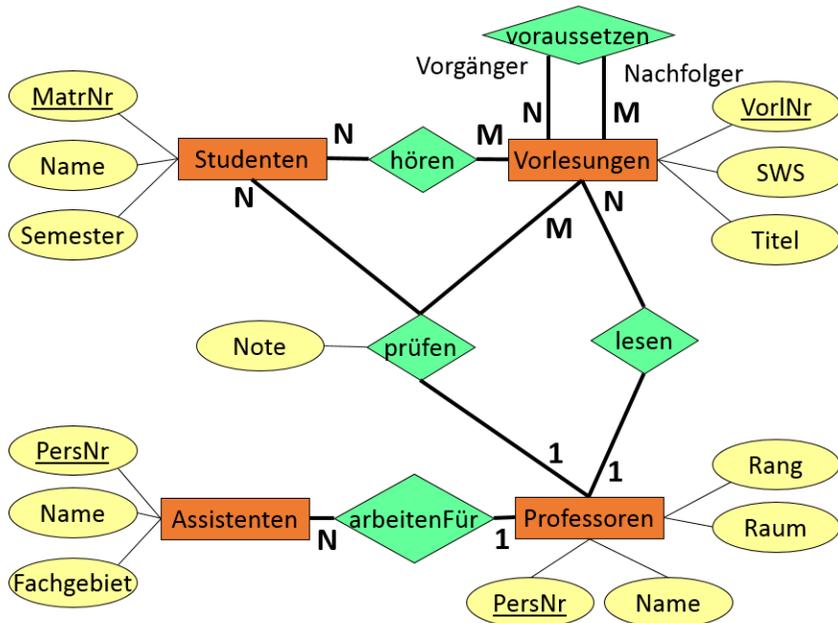
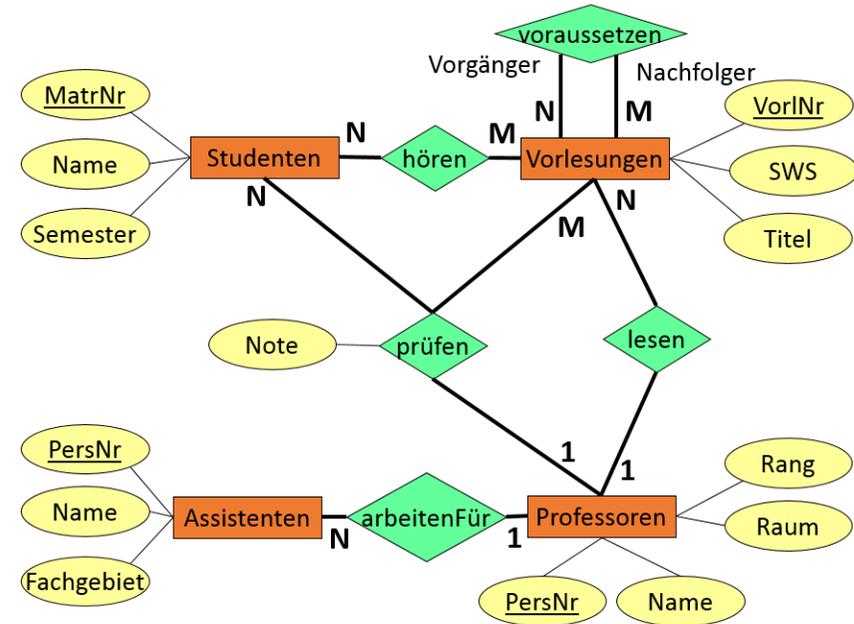
Script generated by TTT

Title: groh: profile1 (29.04.2016)

Date: Fri Apr 29 09:21:04 CEST 2016

Duration: 84:49 min

Pages: 61



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]}

99

Relationale Darstellung von Entities

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

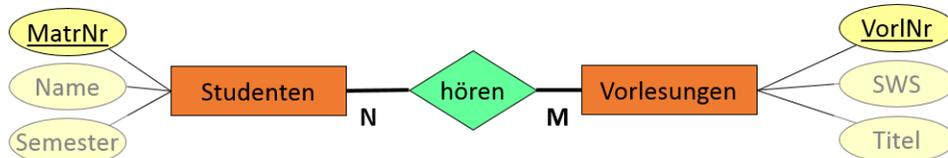
Vorlesungen: {[VorlNr:integer, Titel: string, SWS: integer]}

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

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

99

beziehungen unseres Beispiel-Schemas



Studenten		hören		Vorlesungen	
<u>MatrNr</u>	...	<u>MatrNr</u>	<u>VorlNr</u>	<u>VorlNr</u>	...
...	...	26120	5001	5001	...
26120	...	27550	5001	5001	...
...	...	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	
<u>PersNr</u>	...	<u>PersNr</u>	<u>VorlNr</u>	<u>VorlNr</u>	...
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* : {[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]}

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

104

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

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 : {[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]}

Regel:

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

104

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]}

Regel:

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

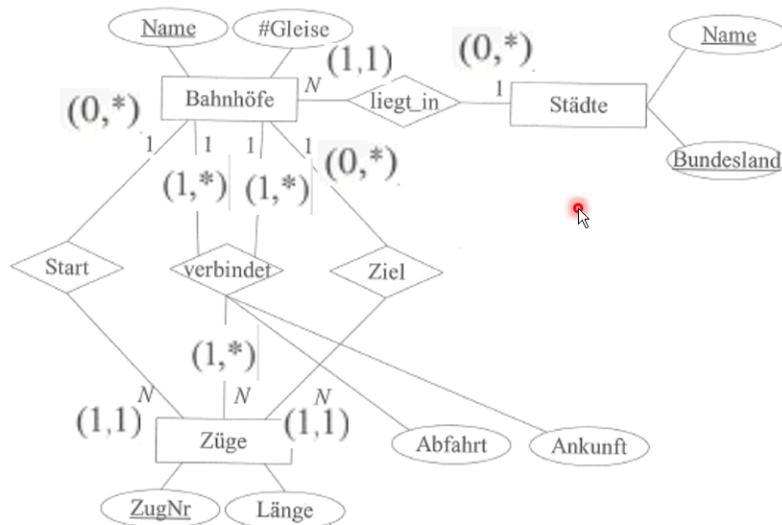
104



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.



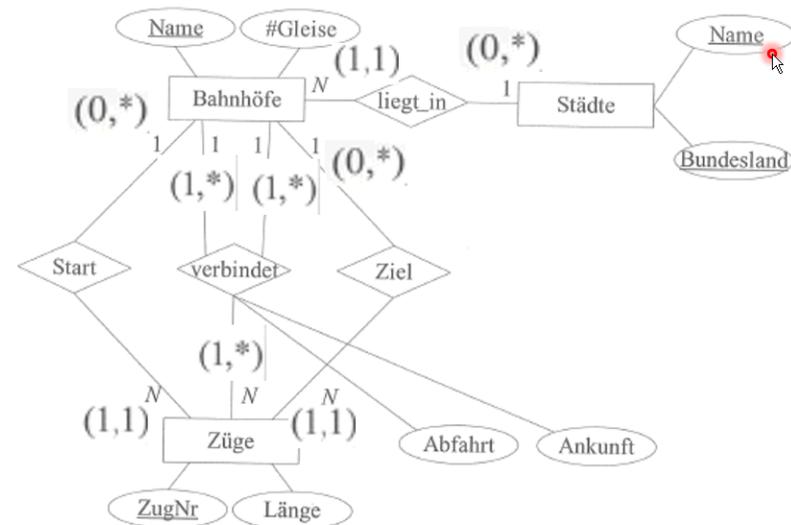
116



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.



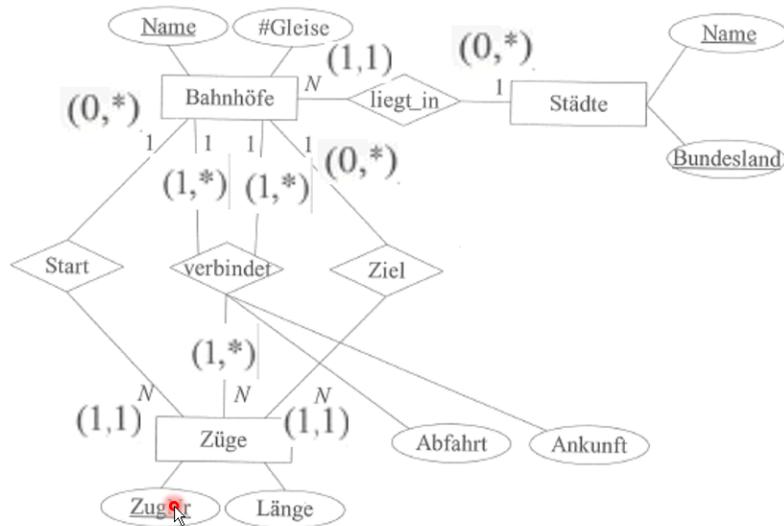
116



Aufgabe 3.1

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

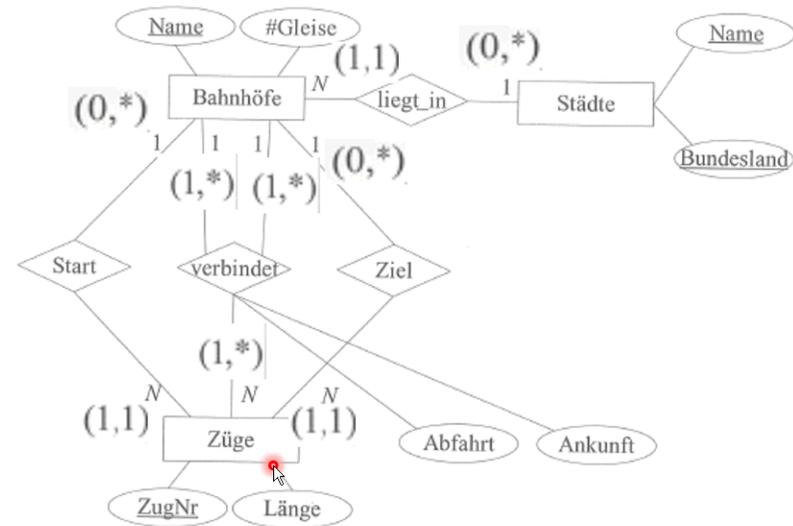
a) Fügen Sie bei den Beziehungen Kardinalitäten in der (min, max) -Notation hinzu.



Aufgabe 3.1

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

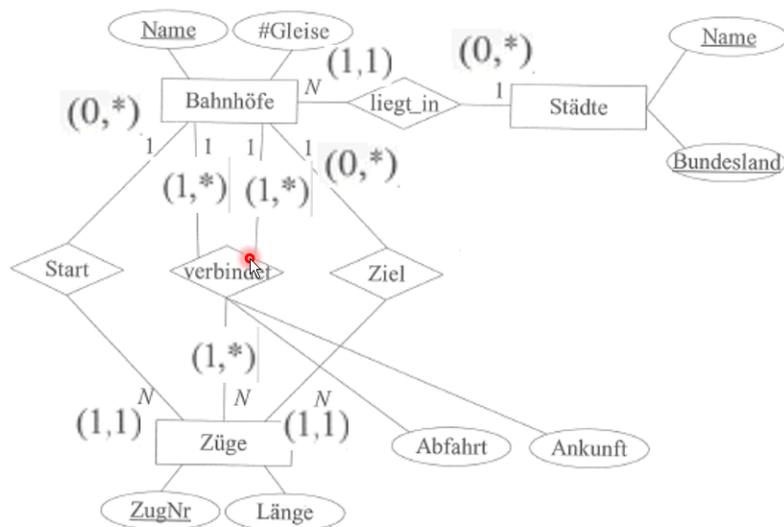
a) Fügen Sie bei den Beziehungen Kardinalitäten in der (min, max) -Notation hinzu.



Aufgabe 3.1

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

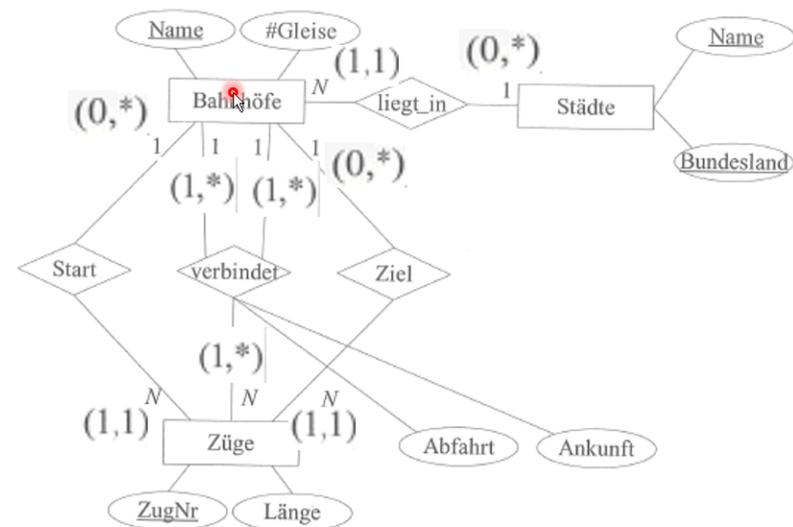
a) Fügen Sie bei den Beziehungen Kardinalitäten in der (min, max) -Notation hinzu.



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:

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



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

Die initiale Überführung ergibt folgende Relationen für die Entitytypen:

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

Bahnhöfe : {[Name : string, #Gleise : integer]}



Züge : {[ZugNr : integer, Länge : integer]}

Für die Beziehungstypen werden folgende Relationen erstellt:

118

119



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

Die initiale Überführung ergibt folgende Relationen für die Entitytypen:

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

Bahnhöfe : {[Name : string, #Gleise : integer]}

Züge : {[ZugNr : integer, Länge : integer]}

Für die Beziehungstypen werden folgende Relationen erstellt:

liegt_in : {[BName : string, SName : string, Bundesland : string]}

Start : {[ZugNr : integer, BName : string]}

Ziel : {[ZugNr : integer, BName : string]}

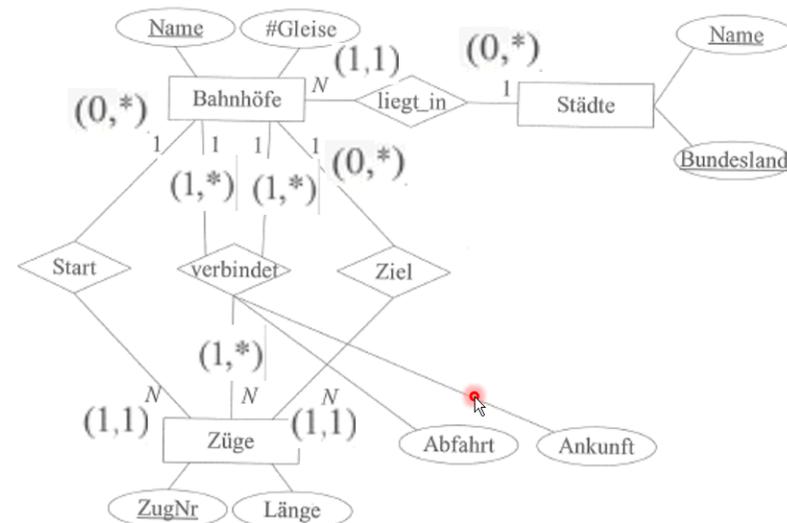
verbindet : {[VonBahnhof : string, NachBahnhof : string,
ZugNr : integer, Abfahrt : date, Ankunft : date]}



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.



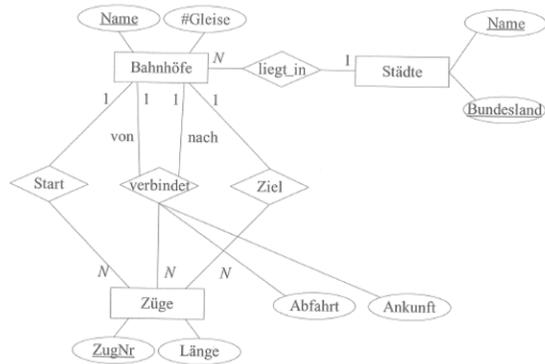
122

116



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]}

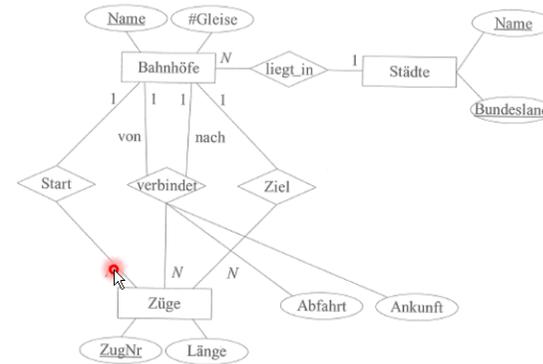


123



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]}

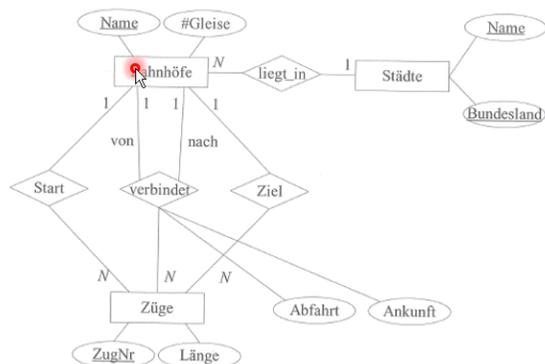


123



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]}

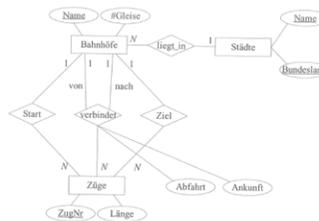


123



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)

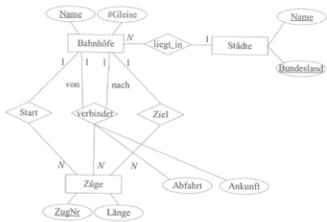
Damit ergibt sich folgendes Schema:

- 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]}

24



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]}

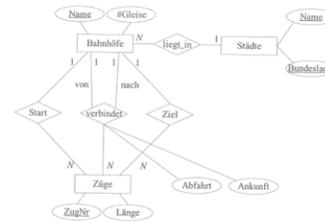
(4) \mapsto (2), (5) \mapsto (3), (6) \mapsto (3)

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]}

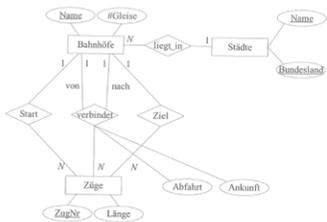
(4) \mapsto (2), (5) \mapsto (3), (6) \mapsto (3)

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]}



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]}

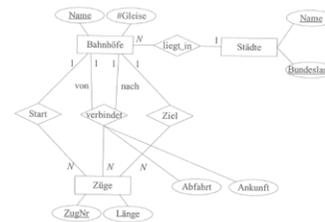
(4) \mapsto (2), (5) \mapsto (3), (6) \mapsto (3)

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.



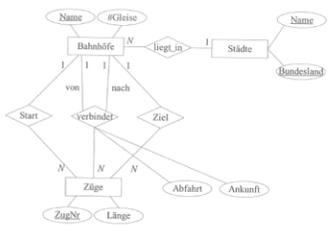
- (1) Städte : {[Name : string, Bundesland : string]}
- (2) Bahnhöfe : {[Name : string, #Gleise : integer]}
- (3) Züge : {[ZugNr : integer, Länge : integer]}
- (4) liegt_in : {[BName : string, SName : string, Bundesland : string]}
- (5) Start : {[ZugNr : integer, BName : string]}
- (6) Ziel : {[ZugNr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

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



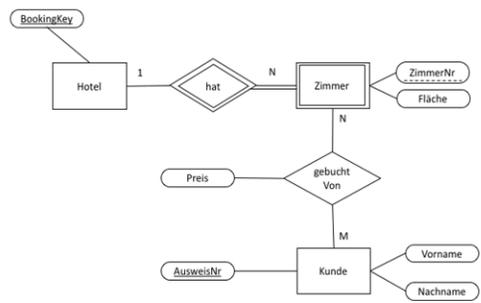
- (1) Städte : {[Name : string, Bundesland : string]}
- (2) Bahnhöfe : {[Name : string, #Gleise : integer]}
- (3) Züge : {[ZugNr : integer, Länge : integer]}
- (4) liegt_in : {[BName : string, SName : string, Bundesland : string]}
- (5) Start : {[ZugNr : integer, BName : string]}
- (6) Ziel : {[ZugNr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

(4) ↔ (2), (5) ↔ (3), (6) ↔ (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]}

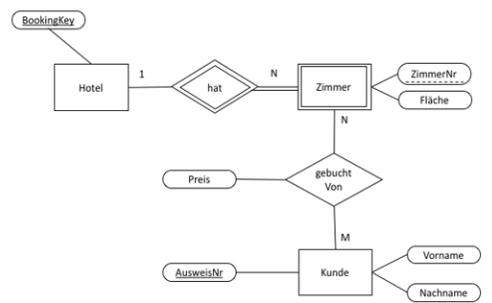
Gegeben sei folgendes ER Diagramm:



Geben Sie das relationale Schema für die Entity "Kunde" und die Beziehung "gebuchtVon" unter Berücksichtigung einer möglichen Verfeinerung (d.h. NACH einer möglichen Verfeinerung) an! (Typen nicht vergessen!)

A	Kunde: {[AusweisNr:string, Vorname:string, Nachname:string]} gebuchtVon: {[ZimmerNr:integer, BookingKey:integer, AusweisNr: integer, Preis:integer]}
B	Kunde: {[AusweisNr:integer, Vorname:string, Nachname:string, Preis:integer]}
C	Kunde: {[AusweisNr:integer, Vorname:string, Nachname:string, Preis:integer, ZimmerNr:integer, BookingKey:integer]}

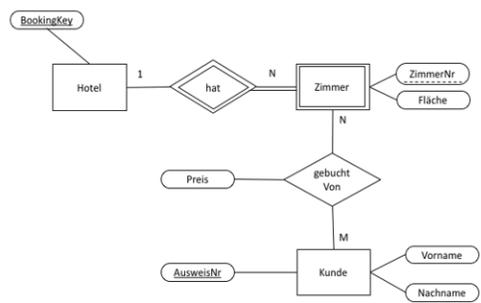
Gegeben sei folgendes ER Diagramm:



Geben Sie das relationale Schema für die Entity "Kunde" und die Beziehung "gebuchtVon" unter Berücksichtigung einer möglichen Verfeinerung (d.h. NACH einer möglichen Verfeinerung) an! (Typen nicht vergessen!)

A	Kunde: {[AusweisNr:string, Vorname:string, Nachname:string]} gebuchtVon: {[ZimmerNr:integer, BookingKey:integer, AusweisNr: integer, Preis:integer]}
B	Kunde: {[AusweisNr:integer, Vorname:string, Nachname:string, Preis:integer]}
C	Kunde: {[AusweisNr:integer, Vorname:string, Nachname:string, Preis:integer, ZimmerNr:integer, BookingKey:integer]}

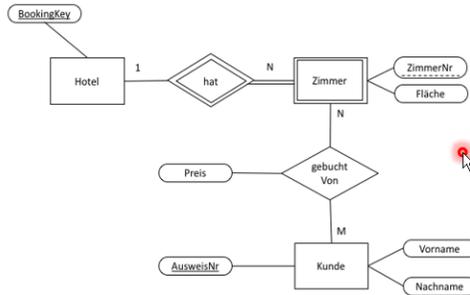
Gegeben sei folgendes ER Diagramm:



Geben Sie das relationale Schema für die Entity "Kunde" und die Beziehung "gebuchtVon" unter Berücksichtigung einer möglichen Verfeinerung (d.h. NACH einer möglichen Verfeinerung) an! (Typen nicht vergessen!)

A	Kunde: {[AusweisNr:string, Vorname:string, Nachname:string]} gebuchtVon: {[ZimmerNr:integer, BookingKey:integer, AusweisNr: integer, Preis:integer]}
B	Kunde: {[AusweisNr:integer, Vorname:string, Nachname:string, Preis:integer]}
C	Kunde: {[AusweisNr:integer, Vorname:string, Nachname:string, Preis:integer, ZimmerNr:integer, BookingKey:integer]}

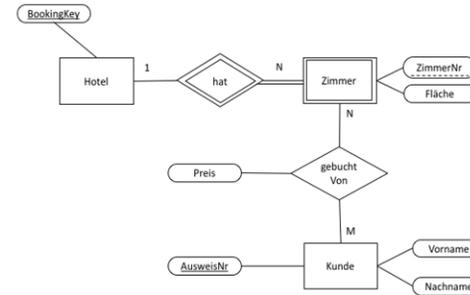
Gegeben sei folgendes ER Diagramm:



Geben Sie das relationale Schema für die Entity "Kunde" und die Beziehung "gebuchtVon" unter Berücksichtigung einer möglichen Verfeinerung (d.h. NACH einer möglichen Verfeinerung) an! (Typen nicht vergessen!)

A	Kunde: {[AusweisNr:string, Vorname:string, Nachname:string]} gebuchtVon: {[ZimmerNr:integer, BookingKey:integer, AusweisNr: integer, Preis:integer]}
B	Kunde: {[AusweisNr:integer, Vorname:string, Nachname:string, Preis:integer]}
C	Kunde: {[AusweisNr:integer, Vorname:string, Nachname:string, Preis:integer, ZimmerNr:integer, BookingKey:integer]}

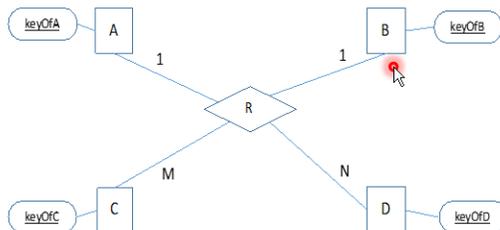
Gegeben sei folgendes ER Diagramm:



Geben Sie das relationale Schema für die Entity "Kunde" und die Beziehung "gebuchtVon" unter Berücksichtigung einer möglichen Verfeinerung (d.h. NACH einer möglichen Verfeinerung) an! (Typen nicht vergessen!)

A	Kunde: {[AusweisNr:string, Vorname:string, Nachname:string]} gebuchtVon: {[ZimmerNr:integer, BookingKey:integer, AusweisNr: integer, Preis:integer]}
B	Kunde: {[AusweisNr:integer, Vorname:string, Nachname:string, Preis:integer]}
C	Kunde: {[AusweisNr:integer, Vorname:string, Nachname:string, Preis:integer, ZimmerNr:integer, BookingKey:integer]}

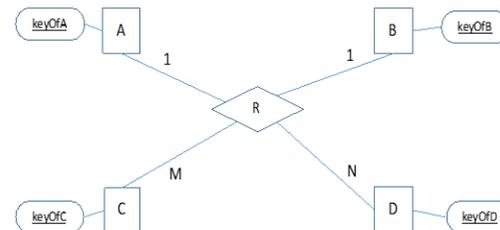
Gegeben sei folgendes ER Diagramm:



Geben Sie zwei mögliche relationale Modellierungen der Relation R an! (Der Typ der vier Schlüssel seien jeweils integer)

A	R: {[keyOfA:integer, keyOfB:integer, keyOfC:integer, keyOfD:integer]} R: {[keyOfA:integer, keyOfB:integer, keyOfC:integer, keyOfD:integer]}
B	R: {[keyOfA:integer, keyOfB:integer, keyOfC:integer, keyOfD:integer]} R: {[keyOfA:integer, keyOfB:integer, keyOfC:integer, keyOfD:integer]}
C	R: {[keyOfA:integer, keyOfB:integer, keyOfC:integer, keyOfD:integer]} R: {[keyOfA:integer, keyOfB:integer, keyOfC:integer, keyOfD:integer]}

Gegeben sei folgendes ER Diagramm:



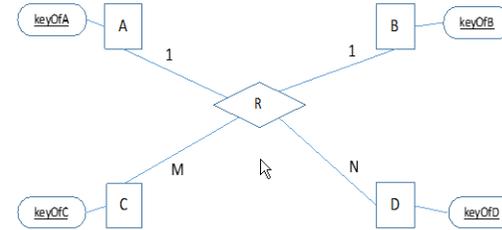
Geben Sie zwei mögliche relationale Modellierungen der Relation R an! (Der Typ der vier Schlüssel seien jeweils integer)

A	R: {[keyOfA:integer, keyOfB:integer, keyOfC:integer, keyOfD:integer]} R: {[keyOfA:integer, keyOfB:integer, keyOfC:integer, keyOfD:integer]}
B	R: {[keyOfA:integer, keyOfB:integer, keyOfC:integer, keyOfD:integer]} R: {[keyOfA:integer, keyOfB:integer, keyOfC:integer, keyOfD:integer]}
C	R: {[keyOfA:integer, keyOfB:integer, keyOfC:integer, keyOfD:integer]} R: {[keyOfA:integer, keyOfB:integer, keyOfC:integer, keyOfD:integer]}

Gegeben sei folgendes ER Diagramm:

A	R:{{ keyOfA:integer, keyOfB:integer, keyOfC:integer, keyOfD:integer }} R:{{ <u>keyOfA:integer</u> , keyOfB:integer, keyOfC:integer, keyOfD:integer }}
B	R:{{ keyOfA:integer, keyOfB:integer, keyOfC:integer, keyOfD:integer }} R:{{ <u>keyOfA:integer</u> , keyOfB:integer, keyOfC:integer, keyOfD:integer }}
C	R:{{ keyOfA:integer, keyOfB:integer, keyOfC:integer, keyOfD:integer }} R:{{ keyOfA:integer, <u>keyOfB:integer</u> , keyOfC:integer, keyOfD:integer }}

• Gegeben sei folgendes ER Diagramm:



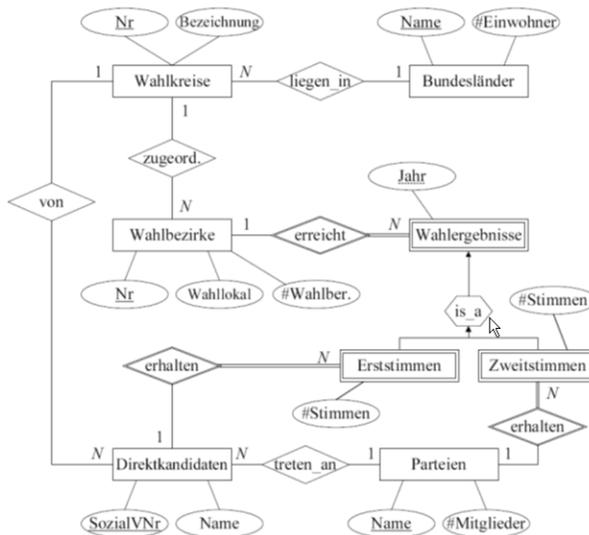
Geben Sie zwei mögliche **relationale Modellierungen** der Relation R an! (Der Typ der vier Schlüssel seien jeweils *integer*)

A	R:{{ keyOfA:integer, keyOfB:integer, keyOfC:integer, keyOfD:integer }} R:{{ <u>keyOfA:integer</u> , keyOfB:integer, keyOfC:integer, keyOfD:integer }}
B	R:{{ <u>keyOfA:integer</u> , keyOfB:integer, keyOfC:integer, keyOfD:integer }} R:{{ keyOfA:integer, <u>keyOfB:integer</u> , keyOfC:integer, keyOfD:integer }}
C	R:{{ keyOfA:integer, keyOfB:integer, keyOfC:integer, keyOfD:integer }} R:{{ keyOfA:integer, keyOfB:integer, <u>keyOfC:integer</u> , keyOfD:integer }}

Gegeben sei das folgende ER-Modell eines Wahlinformationssystems.

(a) Überführen Sie die Entitäten ‚Wahlkreise‘, ‚Wahlbezirke‘ und ‚Bundesländer‘ und die Beziehungen ‚liegen_in‘ und ‚zugeord.‘ (initial) in ein relationales Schema! Geben Sie dabei auch sinnvolle Typen für die Attribute an!

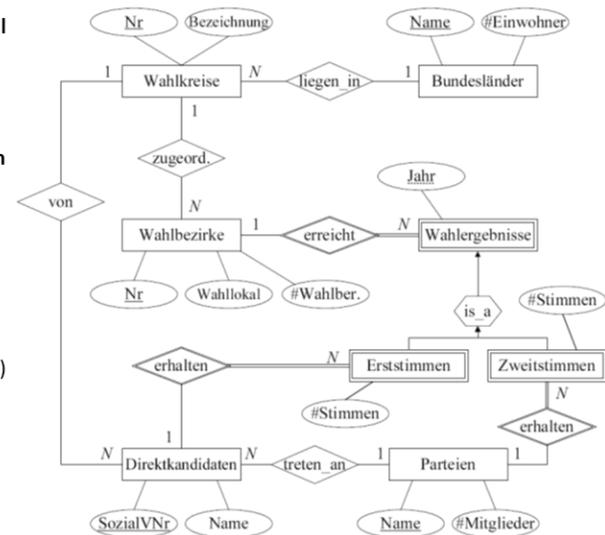
(b) Verfeinern Sie ihr Schema, wenn es möglich und sinnvoll ist! Geben Sie das endgültige (ggf. vereinfachte) Schema an!

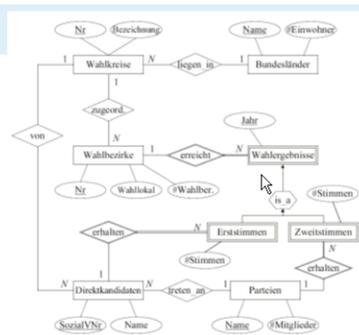


Gegeben sei das folgende ER-Modell eines Wahlinformationssystems.

(a) Überführen Sie die Entitäten ‚Wahlkreise‘, ‚Wahlbezirke‘ und ‚Bundesländer‘ und die Beziehungen ‚liegen_in‘ und ‚zugeord.‘ (initial) in ein relationales Schema! Geben Sie dabei auch sinnvolle Typen für die Attribute an!

(b) Verfeinern Sie ihr Schema, wenn es möglich und sinnvoll ist! Geben Sie das endgültige (ggf. vereinfachte) Schema an!

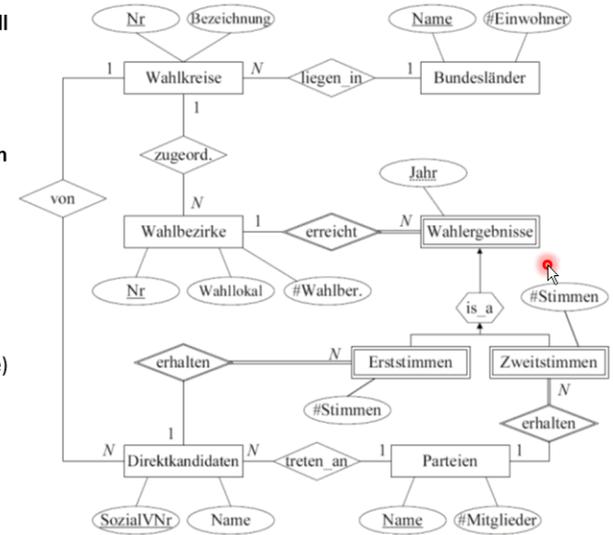




- (a)
- Wahlbezirke : {[Nr : integer, #Wahlberechtigte : integer, Wahllokal : string]} (1)
 - Wahlkreise : {[Nr : integer, Bezeichnung : string]} (2)
 - Bundesländer : {[Name : string, #Einwohner : integer]} (3)
 - liegen_in : {[Wahlkreis : integer, Bundesland : string]} (4)
 - zugeordnet : {[Wahlbezirk : integer, Wahlkreis : integer]} (5)
- (b)
- Verfeinerung: (4) in (2) und (5) in (1) →
- Wahlbezirke : {[Nr : integer, #Wahlberechtigte : integer, Wahllokal : string, Wahlkreis : integer]} (1')
 - Wahlkreise : {[Nr : integer, Bezeichnung : string, Bundesland : string]} (2')
 - Bundesländer : {[Name : string, #Einwohner : integer]} (3)

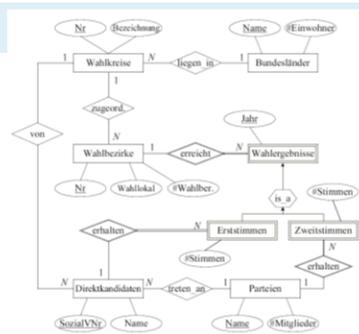
(Quelle (ebenso wie ER-Diagramm): Kemper, Wimmer: Übungsbuch Datenbanksysteme, Oldenbourg 2012)

Gegeben sei das folgende ER-Modell eines Wahlinformationssystems.



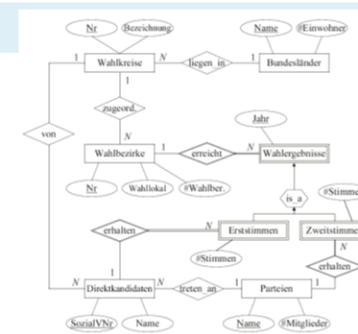
(a) Überführen Sie die Entitäten ‚Wahlkreise‘, ‚Wahlbezirke‘ und ‚Bundesländer‘ und die Beziehungen ‚liegen_in‘ und ‚zugeord.‘ (initial) in ein relationales Schema! Geben Sie dabei auch sinnvolle Typen für die Attribute an!

(b) Verfeinern Sie ihr Schema, wenn es möglich und sinnvoll ist! Geben Sie das endgültige (ggf. vereinfachte) Schema an!



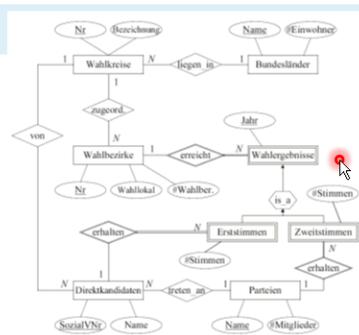
- (a)
- Wahlbezirke : {[Nr : integer, #Wahlberechtigte : integer, Wahllokal : string]} (1)
 - Wahlkreise : {[Nr : integer, Bezeichnung : string]} (2)
 - Bundesländer : {[Name : string, #Einwohner : integer]} (3)
 - liegen_in : {[Wahlkreis : integer, Bundesland : string]} (4)
 - zugeordnet : {[Wahlbezirk : integer, Wahlkreis : integer]} (5)
- (b)
- Verfeinerung: (4) in (2) und (5) in (1) →
- Wahlbezirke : {[Nr : integer, #Wahlberechtigte : integer, Wahllokal : string, Wahlkreis : integer]} (1')
 - Wahlkreise : {[Nr : integer, Bezeichnung : string, Bundesland : string]} (2')
 - Bundesländer : {[Name : string, #Einwohner : integer]} (3)

(Quelle (ebenso wie ER-Diagramm): Kemper, Wimmer: Übungsbuch Datenbanksysteme, Oldenbourg 2012)



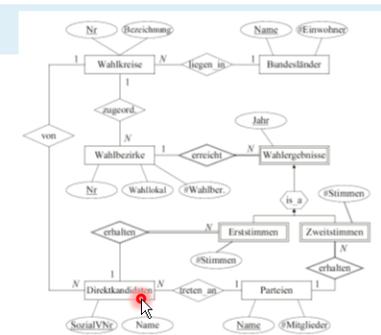
- (a)
- Wahlbezirke : {[Nr : integer, #Wahlberechtigte : integer, Wahllokal : string]} (1)
 - Wahlkreise : {[Nr : integer, Bezeichnung : string]} (2)
 - Bundesländer : {[Name : string, #Einwohner : integer]} (3)
 - liegen_in : {[Wahlkreis : integer, Bundesland : string]} (4)
 - zugeordnet : {[Wahlbezirk : integer, Wahlkreis : integer]} (5)
- (b)
- Verfeinerung: (4) in (2) und (5) in (1) →
- Wahlbezirke : {[Nr : integer, #Wahlberechtigte : integer, Wahllokal : string, Wahlkreis : integer]} (1')
 - Wahlkreise : {[Nr : integer, Bezeichnung : string, Bundesland : string]} (2')
 - Bundesländer : {[Name : string, #Einwohner : integer]} (3)

(Quelle (ebenso wie ER-Diagramm): Kemper, Wimmer: Übungsbuch Datenbanksysteme, Oldenbourg 2012)



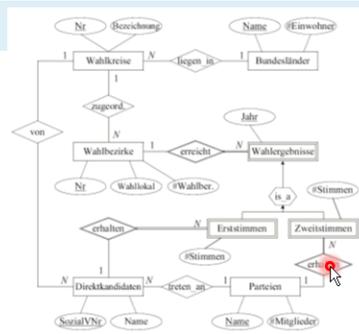
- (a)
- Wahlbezirke : {[Nr : integer, #Wahlberechtigte : integer, Wahllokal : string]} (1)
 - Wahlkreise : {[Nr : integer, Bezeichnung : string]} (2)
 - Bundesländer : {[Name : string, #Einwohner : integer]} (3)
 - liegen_in : {[Wahlkreis : integer, Bundesland : string]} (4)
 - zugeordnet : {[Wahlbezirk : integer, Wahlkreis : integer]} (5)
- (b)
- Verfeinerung: (4) in (2) und (5) in (1) →
- Wahlbezirke : {[Nr : integer, #Wahlberechtigte : integer, Wahllokal : string, Wahlkreis : integer]} (1')
 - Wahlkreise : {[Nr : integer, Bezeichnung : string, Bundesland : string]} (2')
 - Bundesländer : {[Name : string, #Einwohner : integer]} (3)

(Quelle (ebenso wie ER-Diagramm): Kemper, Wimmer: Übungsbuch Datenbanksysteme, Oldenbourg 2012)



- (a)
- Wahlbezirke : {[Nr : integer, #Wahlberechtigte : integer, Wahllokal : string]} (1)
 - Wahlkreise : {[Nr : integer, Bezeichnung : string]} (2)
 - Bundesländer : {[Name : string, #Einwohner : integer]} (3)
 - liegen_in : {[Wahlkreis : integer, Bundesland : string]} (4)
 - zugeordnet : {[Wahlbezirk : integer, Wahlkreis : integer]} (5)
- (b)
- Verfeinerung: (4) in (2) und (5) in (1) →
- Wahlbezirke : {[Nr : integer, #Wahlberechtigte : integer, Wahllokal : string, Wahlkreis : integer]} (1')
 - Wahlkreise : {[Nr : integer, Bezeichnung : string, Bundesland : string]} (2')
 - Bundesländer : {[Name : string, #Einwohner : integer]} (3)

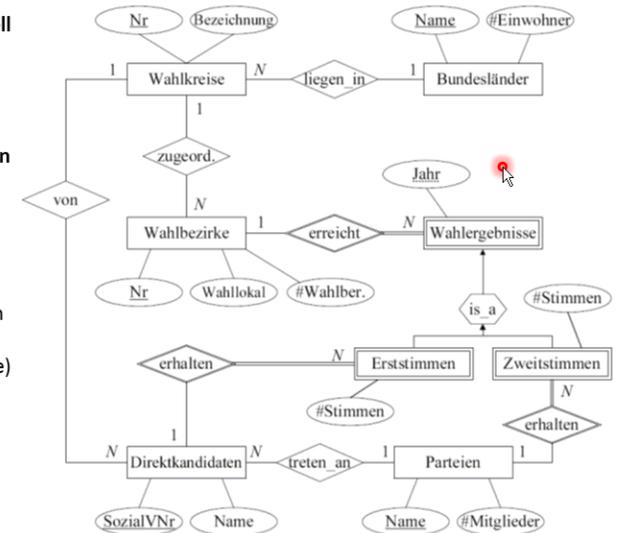
(Quelle (ebenso wie ER-Diagramm): Kemper, Wimmer: Übungsbuch Datenbanksysteme, Oldenbourg 2012)



- (a)
- Wahlbezirke : {[Nr : integer, #Wahlberechtigte : integer, Wahllokal : string]} (1)
 - Wahlkreise : {[Nr : integer, Bezeichnung : string]} (2)
 - Bundesländer : {[Name : string, #Einwohner : integer]} (3)
 - liegen_in : {[Wahlkreis : integer, Bundesland : string]} (4)
 - zugeordnet : {[Wahlbezirk : integer, Wahlkreis : integer]} (5)
- (b)
- Verfeinerung: (4) in (2) und (5) in (1) →
- Wahlbezirke : {[Nr : integer, #Wahlberechtigte : integer, Wahllokal : string, Wahlkreis : integer]} (1')
 - Wahlkreise : {[Nr : integer, Bezeichnung : string, Bundesland : string]} (2')
 - Bundesländer : {[Name : string, #Einwohner : integer]} (3)

(Quelle (ebenso wie ER-Diagramm): Kemper, Wimmer: Übungsbuch Datenbanksysteme, Oldenbourg 2012)

Gegeben sei das folgende ER-Modell eines Wahlinformationssystems.



- (a) Überführen Sie die Entitäten ‚Wahlkreise‘, ‚Wahlbezirke‘ und ‚Bundesländer‘ und die Beziehungen ‚liegen_in‘ und ‚zugeord.‘ (initial) in ein relationales Schema! Geben Sie dabei auch sinnvolle Typen für die Attribute an!
- (b) Verfeinern Sie ihr Schema, wenn es möglich und sinnvoll ist! Geben Sie das endgültige (ggf. vereinfachte) Schema an!

minimale Algebra

- σ Selektion
- π Projektion
- \times Kreuzprodukt
- ρ Umbenennung
- \cup Vereinigung
- $-$ Mengendifferenz

syntaktischer Zucker

- \div Division
- \cap Mengendurchschnitt
- \bowtie Join (Verbund)
- \bowtie äußerer Join
- \ltimes Semi-Join (rechter)
- \rtimes Semi-Join (linker)
- \bowtie linker äußerer Join
- \ltimes rechter äußerer Join
- ...

minimale Algebra

- σ Selektion
- π Projektion
- \times Kreuzprodukt
- ρ Umbenennung
- \cup Vereinigung
- $-$ Mengendifferenz

syntaktischer Zucker

- \div Division
- \cap Mengendurchschnitt
- \bowtie Join (Verbund)
- \bowtie äußerer Join
- \ltimes Semi-Join (rechter)
- \rtimes Semi-Join (linker)
- \bowtie linker äußerer Join
- \ltimes rechter äußerer Join
- ...

$\sigma_p: E \rightarrow E'$ mit P : Prädikat, E : Relation (Tabelle),
 E' : Relation (Tabelle):

$\sigma_p(E)$ wählt **alle Tupel** (Zeilen) aus E aus, **die P erfüllen**.
 Ergebnis: Relation (Tabelle) E'

Beispiel: $\sigma_{\text{Semester} > 10}$ (Studenten)

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

$\xrightarrow{\sigma_{\text{Semester} > 10}}$

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

Das **Selektionsprädikat** P in $\sigma_p(E)$ ist eine Formel, die zu **wahr** oder **falsch** ausgewertet werden kann und die aufgebaut ist aus:

- **Attributnamen** der Argumentrelation E
- **Konstanten**
- den arithmetischen **Vergleichsoperatoren** $=, <, \leq, >, \geq, \neq$
- den **logischen Operatoren** \wedge (und), \vee (oder) und \neg (nicht)

P wird auf **allen** $e \in E$ ausgewertet. Wenn $P(e) = \text{wahr}$ dann wird e in die Ergebnisrelation E' **aufgenommen**.

Das **Selektionsprädikat** P in $\sigma_P(E)$ ist eine Formel, die zu **wahr** oder **falsch** ausgewertet werden kann und die aufgebaut ist aus:

- **Attributnamen** der Argumentrelation E
- **Konstanten**
- den arithmetischen **Vergleichsoperatoren** $=, <, \leq, >, \geq, \neq$
- den **logischen** Operatoren \wedge (und), \vee (oder) und \neg (nicht)

P wird auf **allen** $e \in E$ ausgewertet. Wenn $P(e) = \text{wahr}$ dann wird e in die Ergebnisrelation E' **aufgenommen**.

Das **Selektionsprädikat** P in $\sigma_P(E)$ ist eine Formel, die zu **wahr** oder **falsch** ausgewertet werden kann und die aufgebaut ist aus:

- **Attributnamen** der Argumentrelation E
- **Konstanten**
- den arithmetischen **Vergleichsoperatoren** $=, <, \leq, >, \geq, \neq$
- den **logischen** Operatoren \wedge (und), \vee (oder) und \neg (nicht)

P wird auf **allen** $e \in E$ ausgewertet. Wenn $P(e) = \text{wahr}$ dann wird e in die Ergebnisrelation E' **aufgenommen**.

Das **Selektionsprädikat** P in $\sigma_P(E)$ ist eine Formel, die zu **wahr** oder **falsch** ausgewertet werden kann und die aufgebaut ist aus:

- **Attributnamen** der Argumentrelation E
- **Konstanten**
- den arithmetischen **Vergleichsoperatoren** $=, <, \leq, >, \geq, \neq$
- den **logischen** Operatoren \wedge (und), \vee (oder) und \neg (nicht)

P wird auf **allen** $e \in E$ ausgewertet. Wenn $P(e) = \text{wahr}$ dann wird e in die Ergebnisrelation E' **aufgenommen**.

$\Pi_A : E \rightarrow E'$ mit A : Teilmenge von Attributen von E : „wählt Spalten aus“

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

Π_{Rang}

$\Pi_{\text{Rang}}(\text{Professoren})$
Rang
C4
C3

Beispiel: E' = Professoren x hören

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
...
...
2137	Kant	C4	7

X

hören	
MatrNr	VorlNr
26120	5001
27550	5001
...	...
...	...
...	...
...	...
25403	5022

=

E' = Professoren x hören					
PersNr	Name	Rang	Raum	MatrNr	VorlNr
2125	Sokrates	C4	226	26120	5001
2125	Sokrates	C4	226	27550	5001
...
...
2125	Sokrates	C4	226	25403	5022
2126	Russel	C4	232	26120	5001
...
...
2126	Russel	C4	232	25403	5022
...
2137	Kant	C4	7	25403	5022

- Problem: riesige Ergebnisse (im Bsp. $|E'| = |\text{Prof.}| * |\text{hören}| = 7 * 12$)
- "bessere" Operation: Join (siehe weiter unten)

Beispiel: E' = Professoren x hören

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
...
...
2137	Kant	C4	7

X

hören	
MatrNr	VorlNr
26120	5001
27550	5001
...	...
...	...
...	...
...	...
25403	5022

=

E' = Professoren x hören					
PersNr	Name	Rang	Raum	MatrNr	VorlNr
2125	Sokrates	C4	226	26120	5001
2125	Sokrates	C4	226	27550	5001
...
...
2125	Sokrates	C4	226	25403	5022
2126	Russel	C4	232	26120	5001
...
...
2126	Russel	C4	232	25403	5022
...
2137	Kant	C4	7	25403	5022

- Problem: riesige Ergebnisse (im Bsp. $|E'| = |\text{Prof.}| * |\text{hören}| = 7 * 12$)
- "bessere" Operation: Join (siehe weiter unten)