

3. Logischer Datenbankentwurf

- Definition Relation
- Relationenschemata
- Transformation von ER-Modellen
- Normalisierungen
- Relationale Algebra

Datenbankentwurf

Entwurfsschritte

Informationsbedarfsanalyse

Sammlung aller für eine Miniwelt bedeutsamen Gegenstände, Eigenschaften, Beziehungen und Operationen

Konzeptueller Entwurf

Präzise Beschreibung einer Miniwelt durch relationale oder objektorientierte Modelle

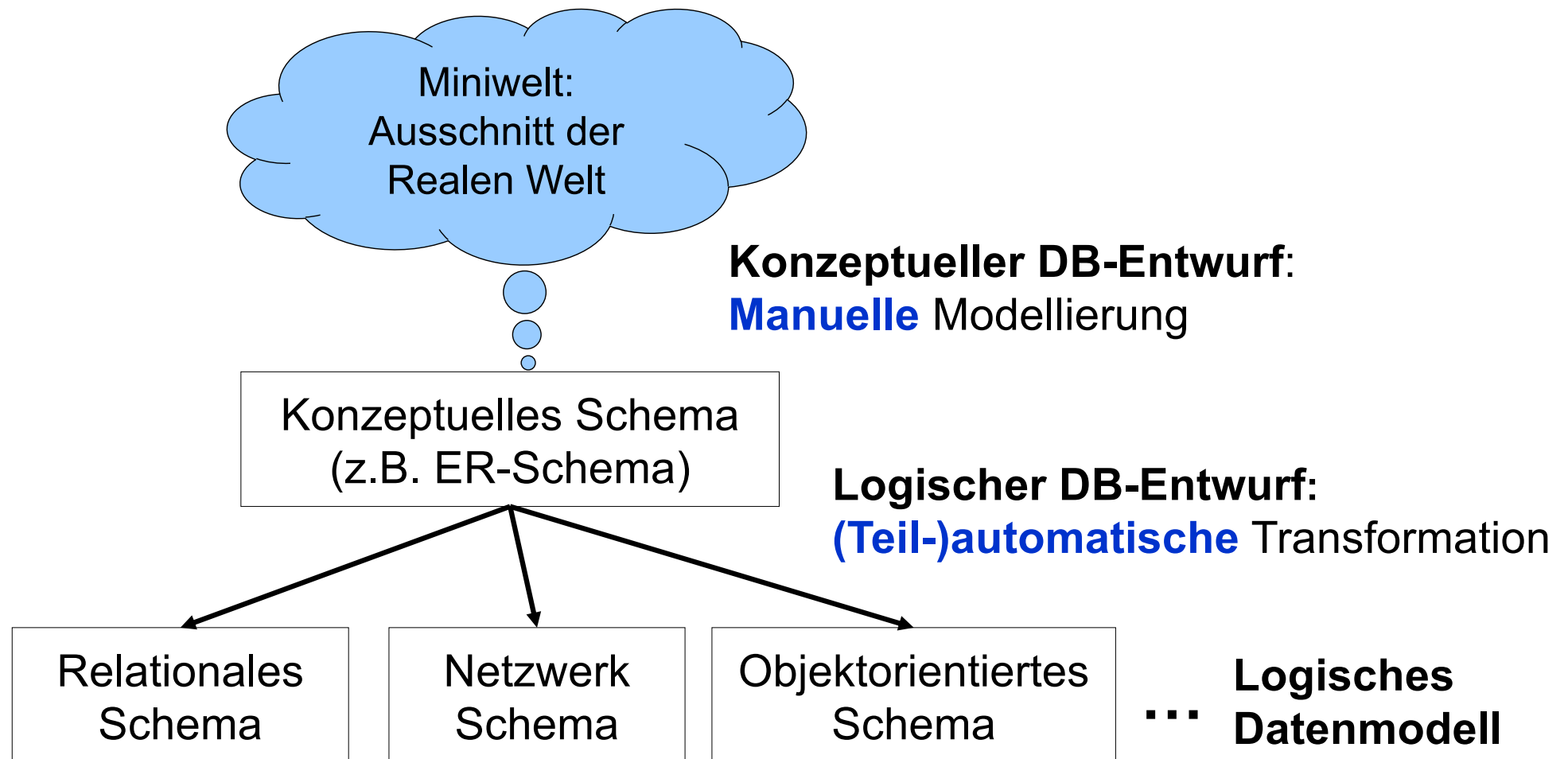
Logischer Entwurf

Abbildung auf ein rechnergestützt interpretierbares Schema, z.B. relationales Schema

Physischer Entwurf

Abbildung des logischen Datenbankschemas in eine effiziente physische Datenbasisstruktur

Logischer Datenbank-Entwurf



Das Relationenmodell

- Das Entity-Relationship-Modell von CHEN beschreibt die Miniwelt in einer sehr abstrakten Form
- Ziel: Beschreibung in "computerverständlicher" Form
- Bereitstellung einer Notation (Syntax) und eindeutigen Bedeutung (Semantik)
- Gebräuchliche Datenbankmodelle
 - Netzwerk-Datenbankmodell
 - Hierarchisches Datenbankmodell
 - Relationales Datenbankmodell
 - Objektorientiertes Datenbankmodell
 - XML-Datenmodell
 - Graph, JSON, etc.

Gliederung

- Strukturteil
 - Beschreibung von Objekttypen (Entity-Typen, Beziehungstypen) der Anwendungswelt
- Operationenteil
 - Bereitstellung von Operationen zur Anfrage oder Manipulation der diesen Objekttypen gehörenden Instanzen (Daten)

Definition Relation

A_1, A_2, \dots, A_n seien beliebige Mengen

1. Das kartesische Produkt $A_1 \times A_2 \times \dots \times A_n$ der Mengen A_i , $i=1, 2, \dots, n$ ist die Menge

$$A_1 \times A_2 \times \dots \times A_n := \{(a_1, a_2, \dots, a_n) \mid a_i \in A_i \text{ für } i = 1, 2, \dots, n\}$$

2. Eine Teilmenge $r \subseteq A_1 \times A_2 \times \dots \times A_n$ heißt (n-stellige) Relation über den Mengen (Attributen) A_1, A_2, \dots, A_n . n ist der Grad der Relation. Wir schreiben:

$$r(A_1, A_2, \dots, A_n)$$

3. Ein Element $t := (t_1, t_2, \dots, t_n) \in r$ wird als n-Tupel der Relation bezeichnet

Relationenschemata

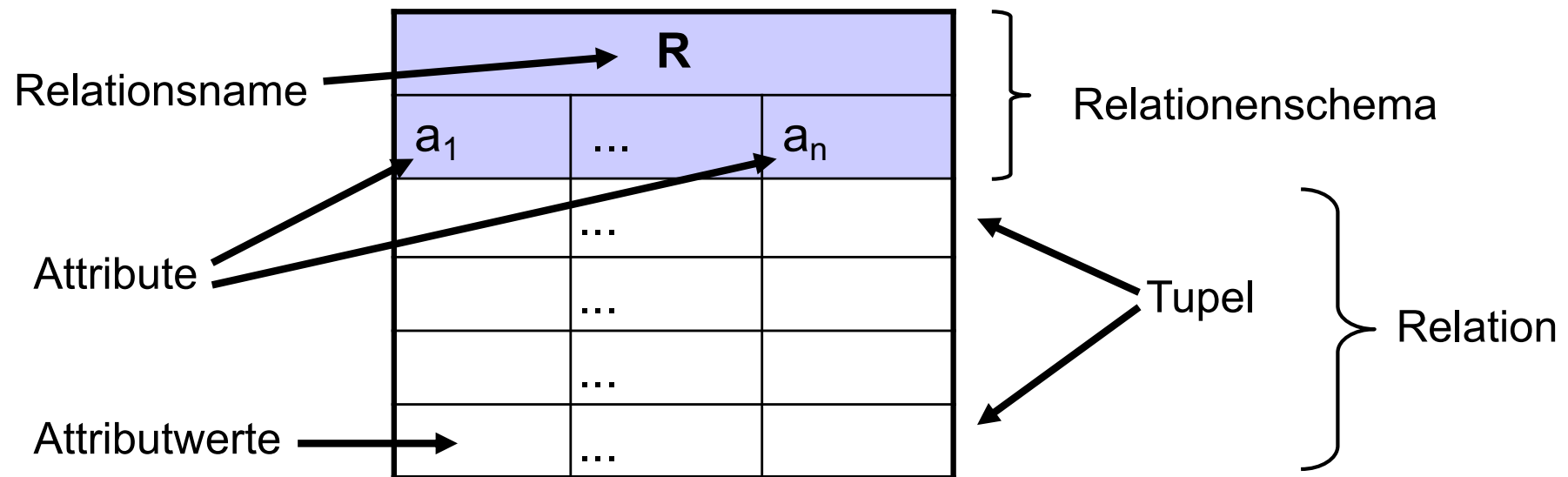
- Ein Relationenschema (Relationstyp) $R = (V, \Sigma)$ besteht aus
 - einem Namen R
 - einer Menge V von Attributen, $V = \{a_1, a_2, \dots, a_n\}$
 - einer Menge Σ von Integritätsbedingungen (Constraints)
- Attributen werden Wertebereiche (Domains) zugeordnet
 - In der Praxis Standard-Datentypen wie z.B. INTEGER, STRING, DATE, ...
 - Beispiel: $\text{dom}(\text{NAME}) = \text{STRING}$
 - $\text{dom}(V) = \text{dom}(a_1) \times \text{dom}(a_2) \times \dots \times \text{dom}(a_n)$

Instanzen von Relationenschemata

- Eine **Relation** ist eine Instanz des zugehörigen Relationenschemas $R: (V, \Sigma) \Leftrightarrow$
 - 1) r ist Relation vom Format V d.h. $r \subseteq \text{dom}(V)$
 - 2) r genügt allen Integritätsbedingungen von Σ
- Sei $\mathcal{R} = \{R_1, \dots, R_k\}$, $\Sigma_{\mathcal{R}}$ eine Menge von interrelationalen Integritätsbedingungen. Ein **Datenbankschema** wird definiert durch
 - $D = (\mathcal{R}, \Sigma_{\mathcal{R}})$
- Eine **relationale Datenbank** $d = \{r_1, \dots, r_k\}$ ist eine Menge von Relationen r_i vom Typ R_i $1 \leq i \leq k$, die $\Sigma_{\mathcal{R}}$ erfüllen
 - Eine relationale Datenbank ist also eine Zusammenfassung mehrerer Relationen mit Integritätsbedingungen

Tabellarische Darstellung von Relationen

- Spaltenüberschriften: Attribute des Relationsschemas
- Tupel: Zeile in Tabelle
- Relation: Einträge in Tabelle
- Einträge in Tabellen gehören zu den entsprechenden Domains



Tabellarische Darstellung von Relationen

Schlüsselattribute

Student = ({matrikelnr, name, wohnort}, {matrikelnr → name, wohnort ist vom Typ String})

Vereinfacht:

Student = ({matrikelnr, name, wohnort})

Mit Darstellung der Wertebereichen:

Student = ({matrikelnr: Integer, name: String, wohnort: String})

Student		
<u>matrikelnr</u>	name	wohnort
132004	Müller	Singen
131208	Zimmer	Lindau
131001	Abel	
131013	Jung	Konstanz
132740	Moser	Singen

Eigenschaften einer Relation / Tabelle

- Eindeutiger Name
- Reihenfolge der Tupel (Zeilen) ist beliebig
- Attributnamen sind eindeutig innerhalb einer Relation
- Die Tupel der Relation (Zeilen der Tabelle) sind paarweise verschieden
- Für das Einbringen eines Tupels in eine Relation ist mindestens der Primärschlüssel vorzugeben
- Nicht-Schlüsselattribute können durch „Nullwerte“ belegt werden
 - Bedeutung: „Wert ist nicht existent“

Primärschlüssel

- Primärschlüssel
 - Attribut (Attributmenge), die ein Tupel eindeutig identifiziert
 - Ein Primärschlüssel existiert nur einmal in einer Relation
 - Bei der Tupel-Suche reicht die Suche nach Primärschlüssels aus
 - Darstellung: Name unterstrichen
- Schlüsselkandidat
 - Es können mehrere potentielle Schlüssel vorhanden sein ("Schlüsselkandidaten")
 - Auszeichnung eines Schlüsselkandidaten als Primärschlüssel
 - Sekundärschlüssel: nicht als Primärschlüssel ausgezeichnet

Studierender			
<u>matrikelnr</u>	personalausweisnr	name	wohnort
132004	1252345432	Müller	Singen
131001	5432534234	Abel	

Primärschlüssel

- Ein Primärschlüssel kann aus mehreren Attributen bestehen

Benotung		
<u>matrikelnr</u>	<u>klausur</u>	note
135745	Datenbanksysteme	2,3
135745	Systemmodellierung	4,0
135663	Datenbanksysteme	1,7

- Einführung "künstlicher" Primärschlüssel (Pseudokey)
 - Falls kein Schlüsselkandidat existiert
 - Falls Schlüsselkandidaten aus zu vielen Attributen besteht

Kunde			
<u>kunde-ID</u>	name	vorname	wohnort
11200	Kunz	Stefan	Konstanz
11210	Maier	Andreas	Konstanz

Primärschlüssel

- Wahl des Primärschlüssels
 - Konstanz: Primärschlüssel sollten sich nicht ändern
 - Wertepflicht: Primärschlüssel muss ein Mußattribut sein
 - Minimalität: Bei zusammengesetzten Schlüsseln sollte Primärschlüssel so gewählt werden, dass kein Attribut entfernt werden kann, ohne dass Identifikationsvermögen verloren geht
 - Design-Entscheidung: Pseudokey / zusammengesetzter Key
- Tabellen mit gleichem Primärschlüssel können zusammengefasst werden

Fremdschlüssel

- Fremdschlüssel
 - Attribute zur Identifikation von Tupel aus anderen Relationen
 - Modellierung einer Zuordnung (Beziehung)
 - Darstellung: Name gestrichelt unterstrichen

Kunde			
<u>kundennr</u>	name	vorname	wohnort
11200	Kunz	Stefan	Konstanz
11210	Maier	Andreas	Konstanz

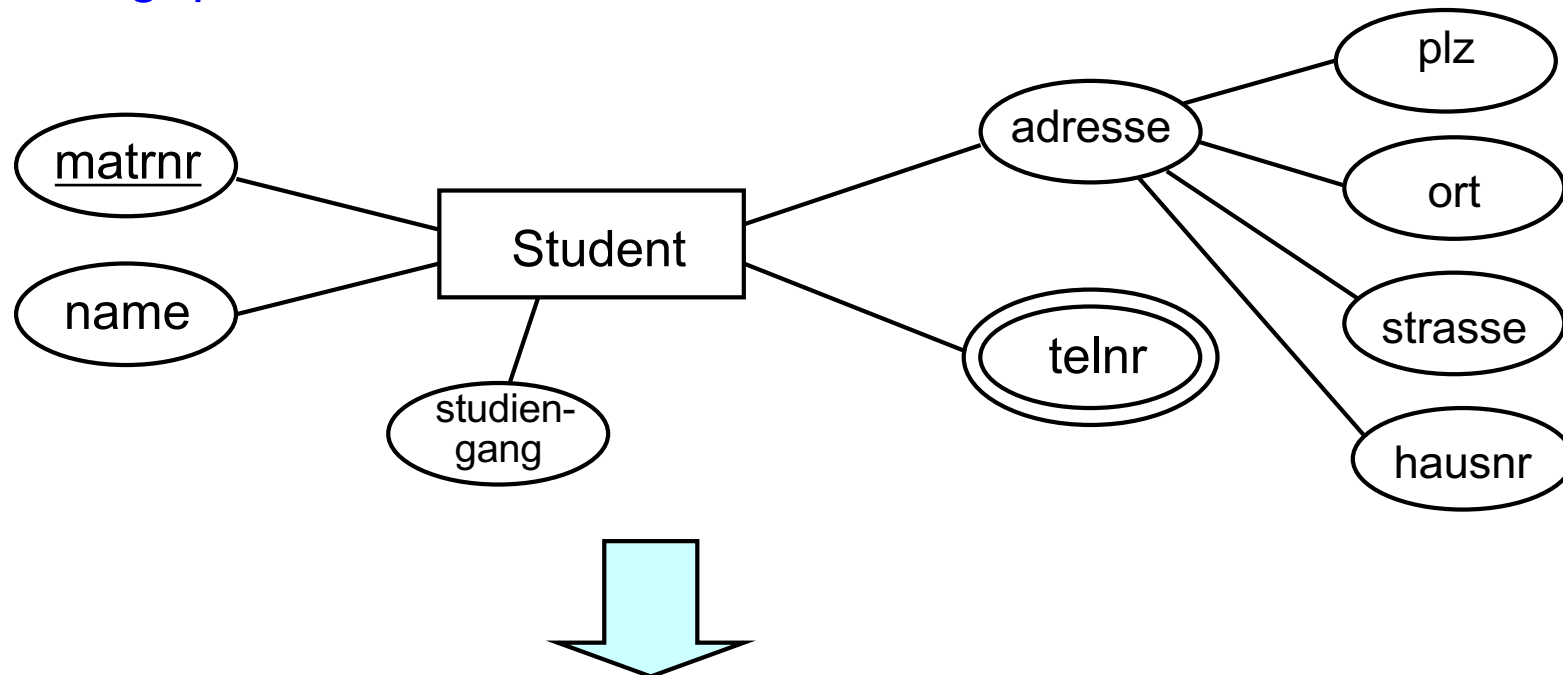
Artikel		
<u>artikelnr</u>	bezeichnung	preis
224	Fernseher	600
116	DVD-Player	200

Bestellung			
<u>bestellnr</u>	<u>kundennr</u>	<u>artikelnr</u>	datum
224533	11200	224	21.03.2005
226522	11210	116	17.02.2005

Transformation von ER-Modellen

Entity-Typen

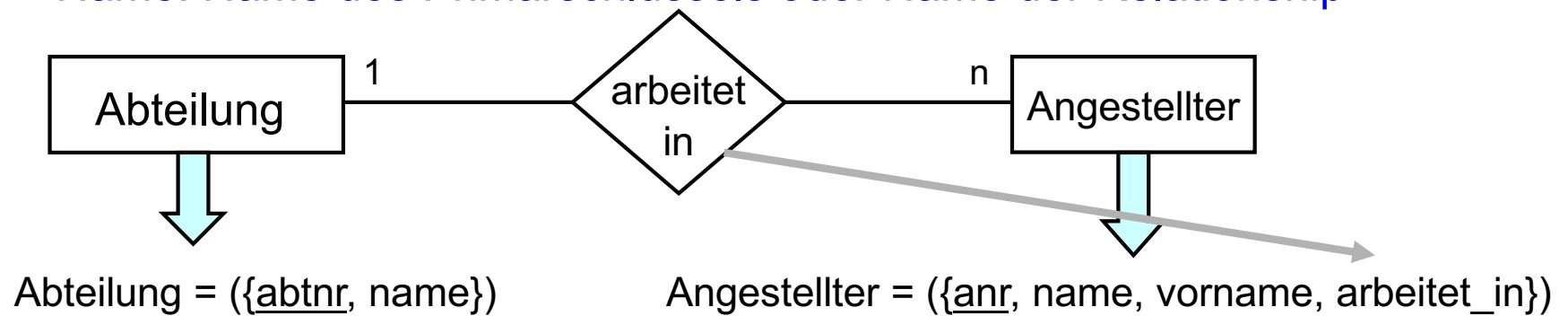
- Abbildung Entity-Typen auf Relationen
 - Abbildung Entity-Attribute auf Attribute der Relation
 - Mehrwertige und zusammengesetzte Attribute müssen noch angepasst werden !



Student = ({matrnr, name, studiengang, (plz, stadt, strasse, hnr), {telnr}})

Transformation von 1:n-Relationships

- Abbildung 1:n-Relationship
 - Anhängen der Attribute an die Relation, die dem Entity-Typ mit der mit „n“ bezeichneten Kante entspricht
- Fremdschlüssel
 - Attributmenge, die in einer anderen Relation Primärschlüssel ist
 - Name: Name des Primärschlüssels oder Name der Relationship



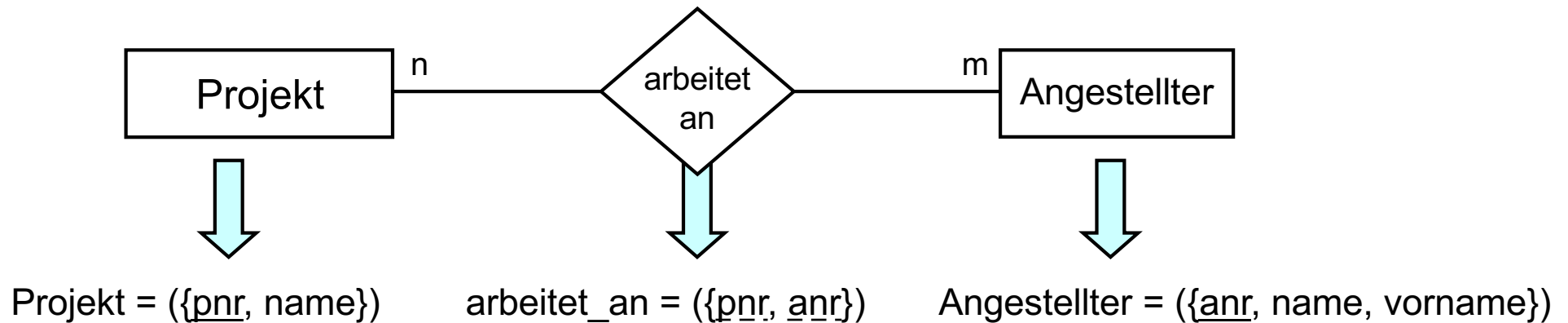
Beispiel:

Abteilung	
<u>abtnr</u>	name
448	HW
122	SW
200	Personal

Angestellter			
<u>anr</u>	name	vorname	<u>arbeitet_in</u>
2004	Müller	Hans	448
1208	Zimmer	Jochen	122
1001	Abel	Kai	122

Transformation von n:m-Relationships

- Realisierung einer eigenen Relation
 - Schlüssel sind die Schlüssel der Relationen der beteiligten Entity-Typen
 - Attribute des Beziehungstyps als zusätzliche Attribute



Beispiel:

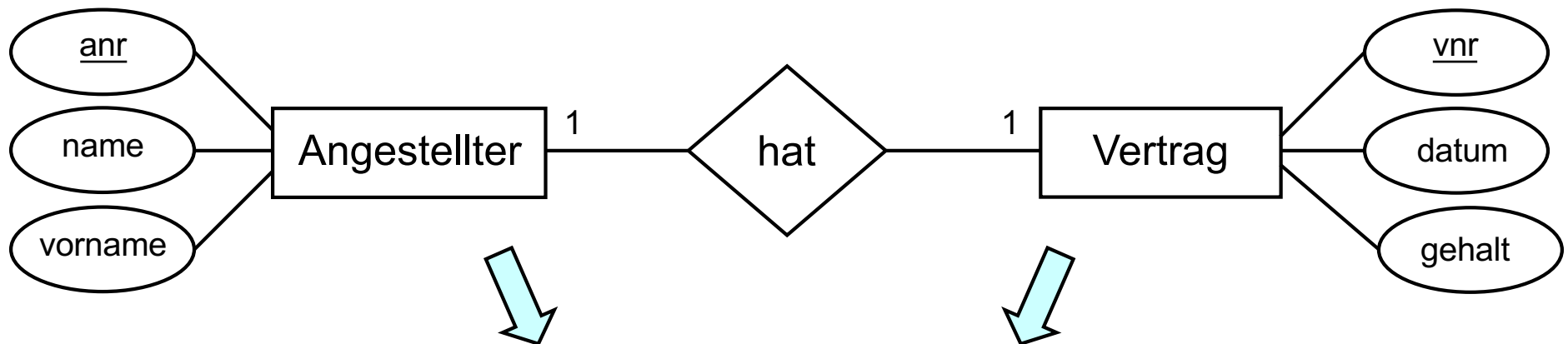
Projekt	
<u>pnr</u>	name
56	PS1
77	QEM
12	PN

arbeitet an	
<u>pnr</u>	<u>anr</u>
56	2004
77	2004
77	1001

Angestellter		
<u>anr</u>	name	vorname
2004	Müller	Hans
1208	Zimmer	Jochen
1001	Abel	Kai

Transformation von 1:1-Relationships

- Möglichkeit 1
 - Zusammenfassen zu einer Relation



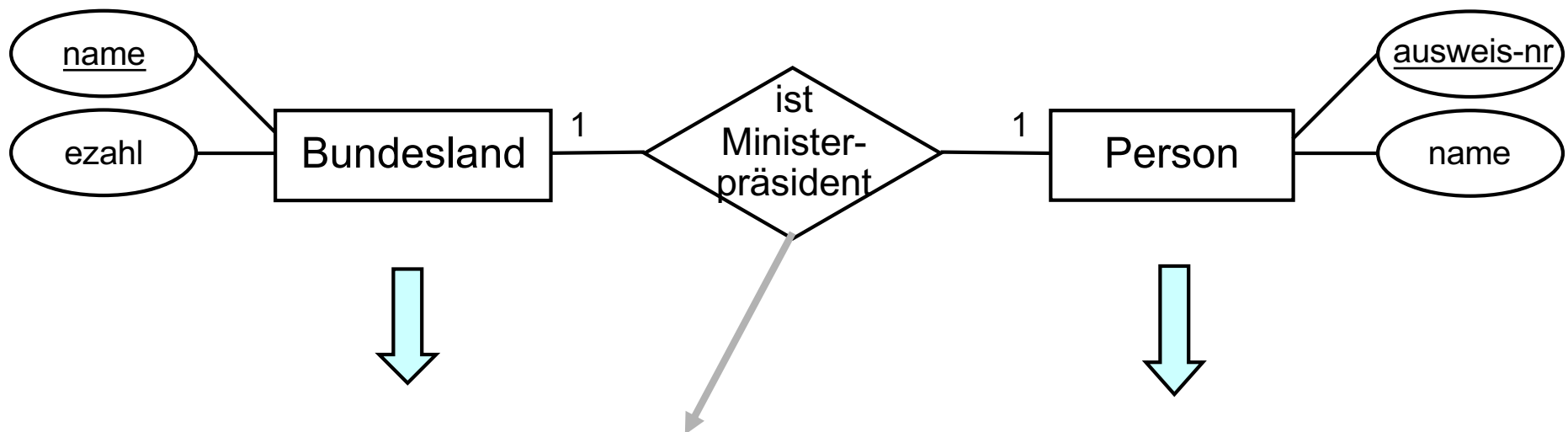
Angestellter = ({ANR, Name, Vorname, VNR, Datum, Gehalt})

Beispiel:

Angestellter					
<u>anr</u>	name	vorname	vnr	datum	gehalt
2004	Müller	Hans	v646	1.10.1999	3500
1208	Zimmer	Jochen	v83	1.1.2002	4000
1001	Abel	Kai	v143	1.3.1990	5500

Transformation von 1:1-Relationships

- Möglichkeit 2
 - Übernahme des Primärschlüssels



Bundesland = ({name, ezahl, ausweisnr})

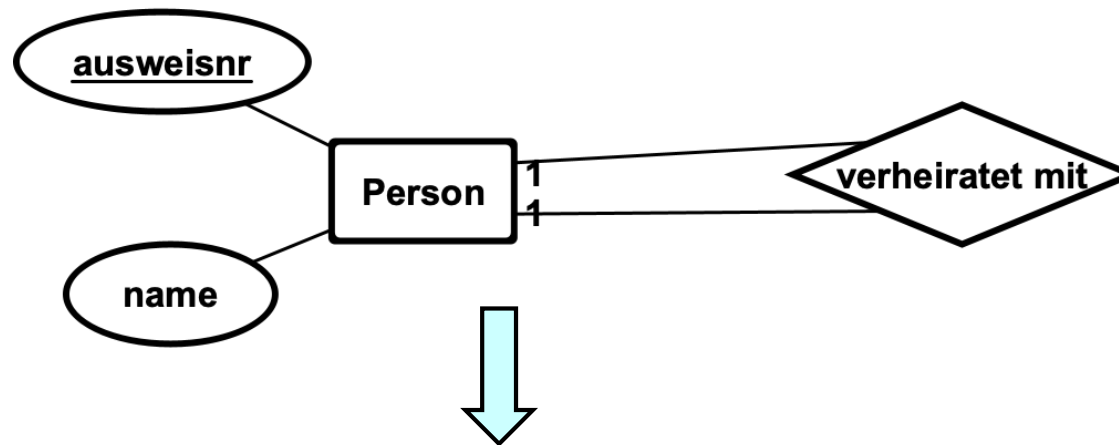
Bundesland		
<u>name</u>	ezahl	<u>ausweis-nr</u>
BW	10.000.000	20045566
Bayern	12.000.000	12087744

Person = ({ausweis-nr, name})

Person77	
<u>ausweis-nr</u>	name
20045566	Kretschmann
12087744	Söder

Transformation von 1:1-Relationships

- Möglichkeit 1
 - Schlechte Umsetzung !!!



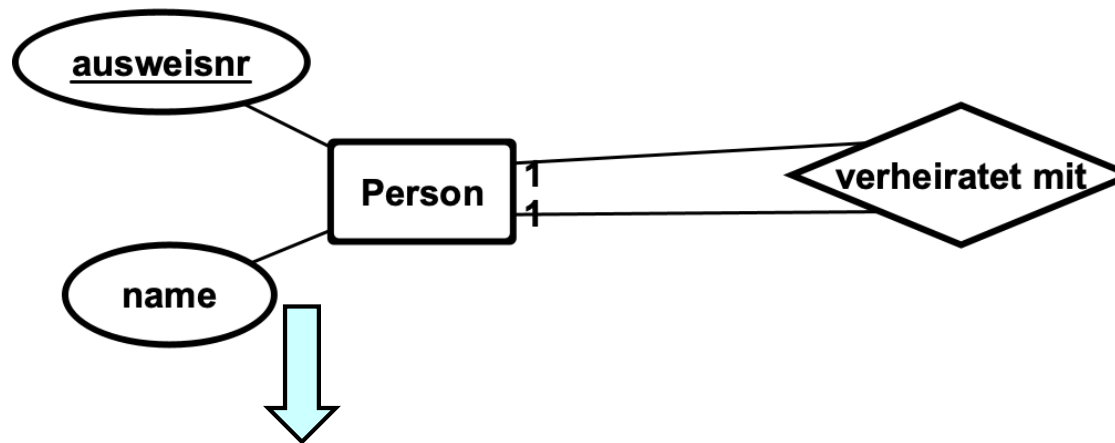
Person = ({ausweisnr1, name1, ausweisnr2, name2})

Person			
ausweisnr1	name1	ausweisnr2	name2
3333212	Müller		
		4444222	Maier
5555313	Kunz	7777636	Kunz

Probleme?

Transformation von 1:1-Relationships

- Möglichkeit 2
 - Realisierung als eigene Relation

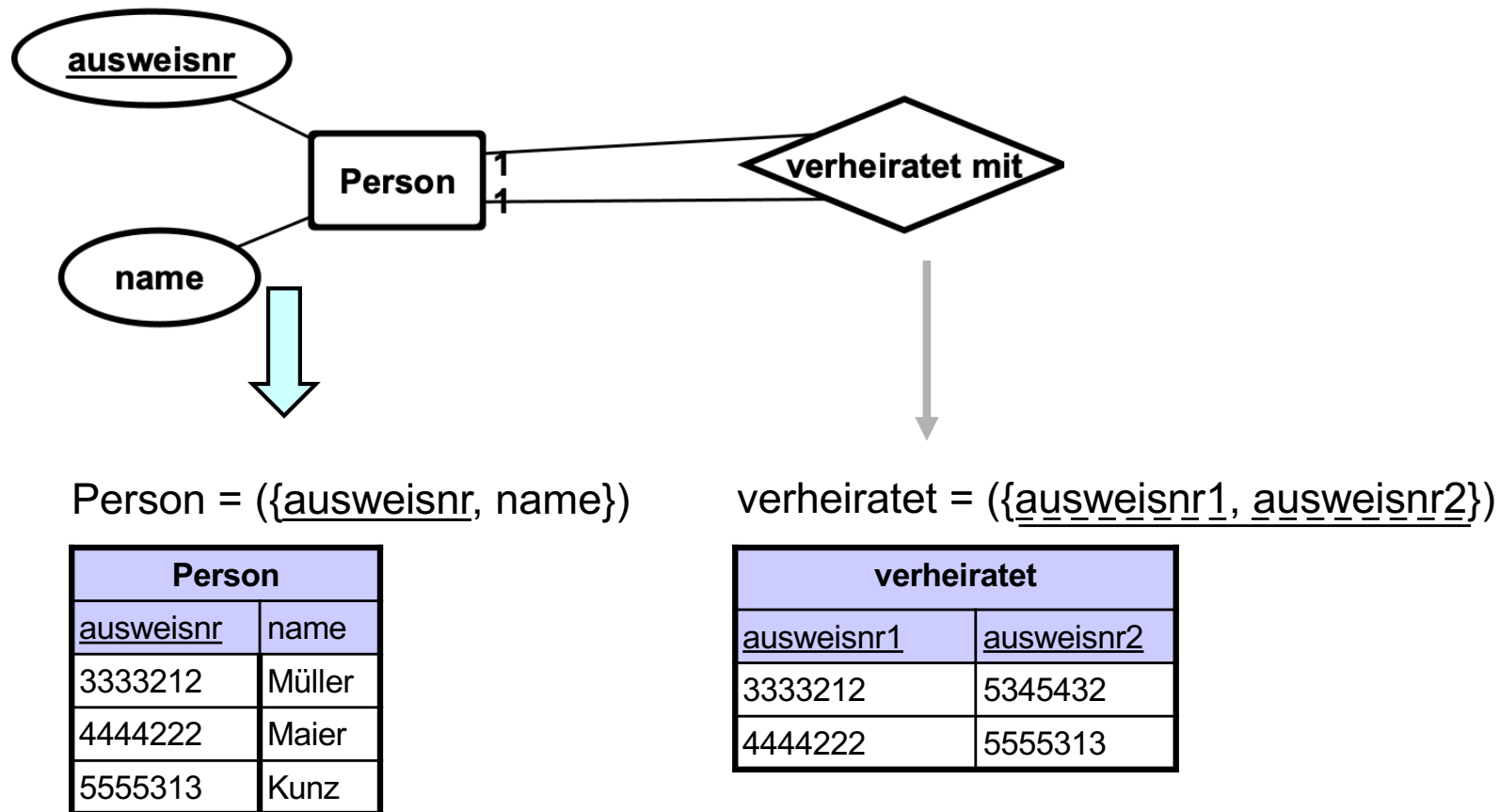


Person = ({ausweisnr, name, verheiratetMit})

Person		
<u>ausweisnr</u>	name	<u>verheiratetMit</u>
3333212	Müller	
5555313	Kunz	7777636
7777636	Kunz	???

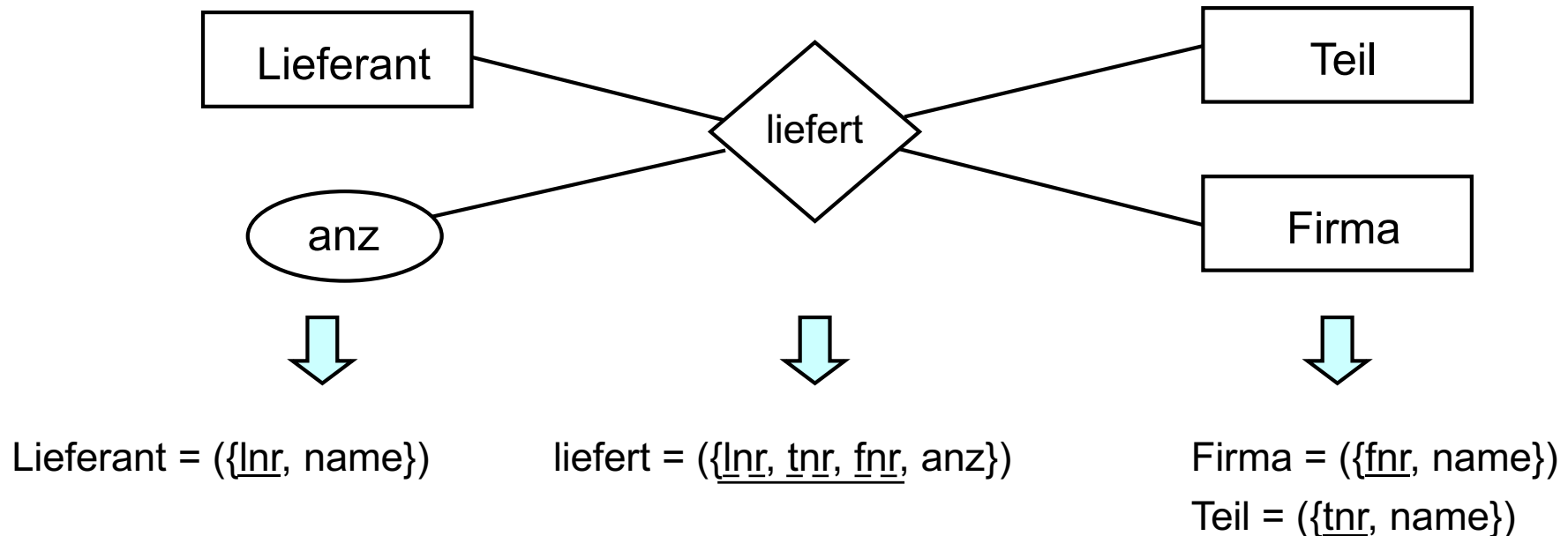
Transformation von 1:1-Relationships

- Möglichkeit 3 – Vermeidung von NULL-Werten
 - Realisierung als eigene Relation



Transformation von mehrstelligen Relationships

- Abbildung einer mehrstelligen Relationship analog zu n:m-Relationship



Beispiel:

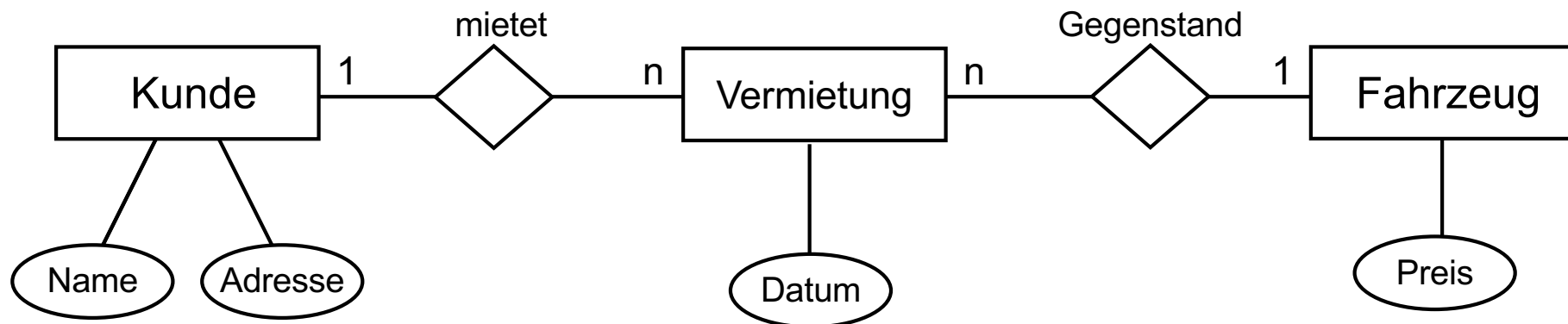
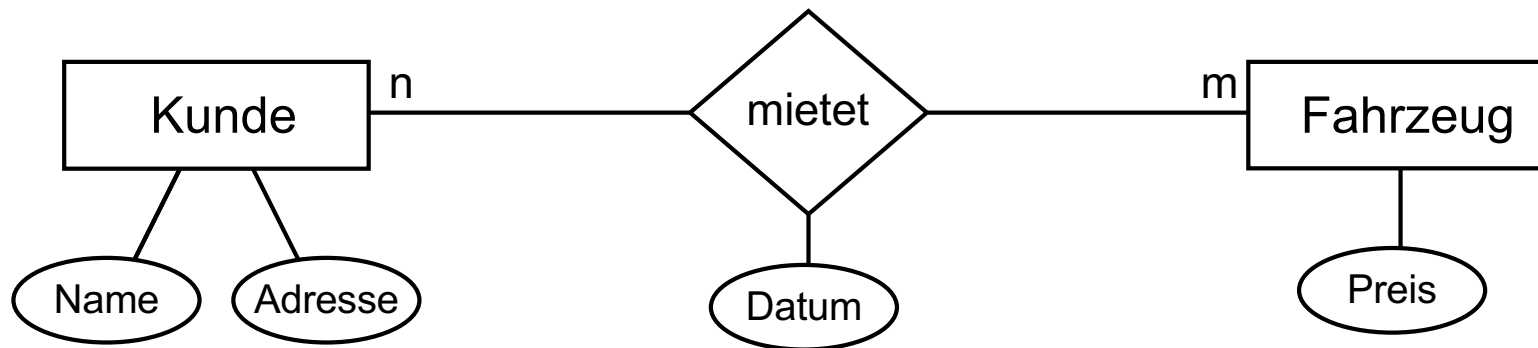
Lieferant	
<u>lnr</u>	name
346	XY GmbH
624	KL AG

Teil	
<u>tnr</u>	name
4211	Getriebe
6344	Kupplung

Firma	
<u>fnr</u>	name
442	AB GmbH
613	DE AG

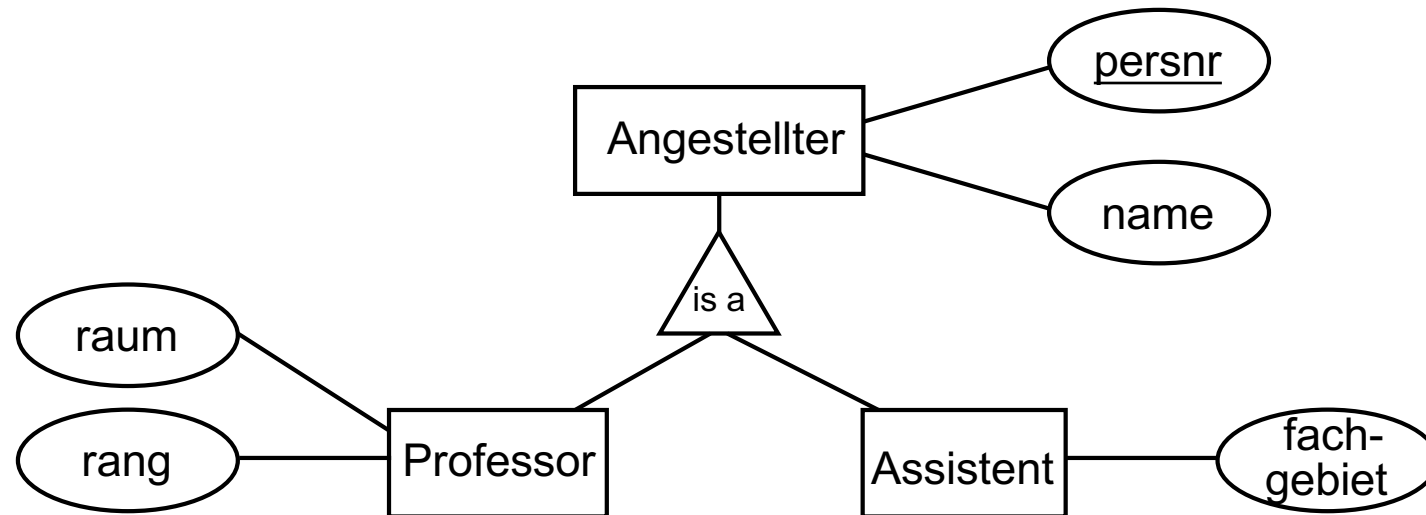
liefert			
<u>lnr</u>	<u>tnr</u>	<u>fnr</u>	anz
346	4211	442	1000
624	6344	613	500

Unterschiedliche Abbildung ?



Transformation von Spezialisierung

- Keine direkte Abbildung in relationales Modell möglich



1. Table per Subclass / Vertical

Angestellter = ({persnr, name})

Professor = ({persnr, raum, Rang})

Assistent = ({persnr, fachgebiet})

2. Table per Class Hierarchy / Flat

Angestellter = ({persnr, name, raum, rang, fachgebiet, is_a})

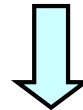
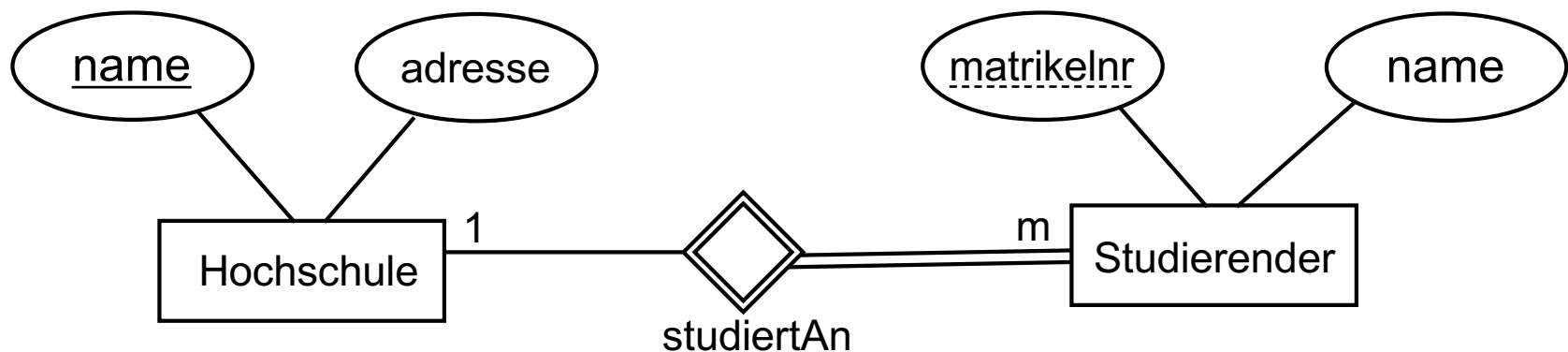
3. Table per Concrete Class / Horizontal

Professor = ({persnr, name, raum, rang})

Assistent = ({persnr, name, fachgebiet})

Transformation schwacher Entities

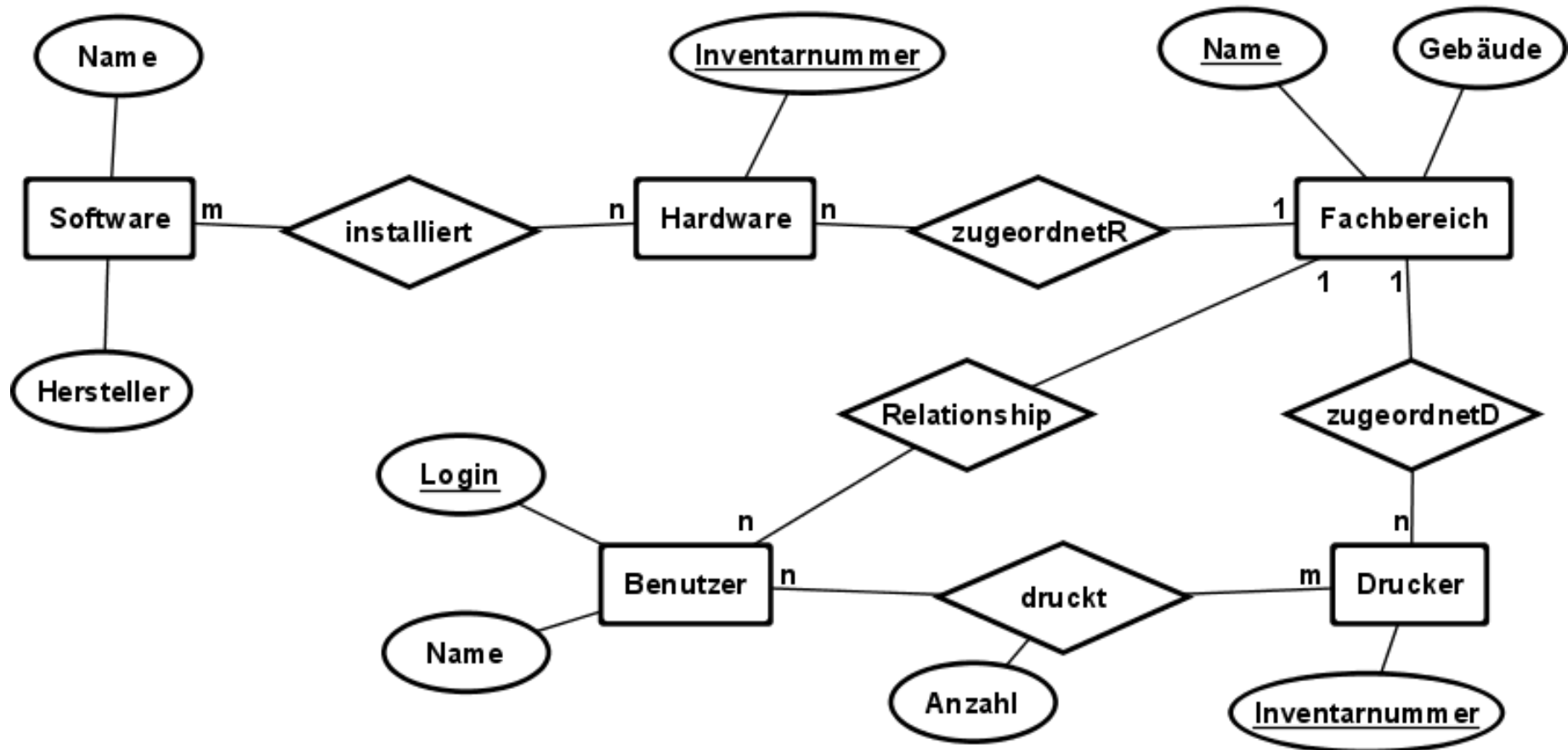
- Partieller Schlüssel bildet zusammen mit Fremdschlüssel den Primärschlüssel der Relation



Hochschule = ({name, adresse})
Studierender = ({matrikelnr, studiertAn, name})

Beispiel Abbildung Relationenmodell

- Bilden Sie folgende ER-Modell in das Relationenmodell ab:



Zusammenfassung ER → Relationenmodell

ER-Modell	Relationenmodell
Entity-Typ	Relation
1:1 – Relationship	Zu einer Relation zusammenfassen
1:n – Relationship	Fremdschlüssel zu eindeutiger Relation
n:m – Relationship	Relation aus Relationship mit zwei Fremdschlüssel
n-äre Relation	Relation aus Relationship mit n Fremdschlüssel
Einfaches Attribut	Attribut
Zusammengesetztes Attribut	Menge von einfachen Attributen
Mehrwertiges Attribut	Relation mit Fremdschlüssel
Schlüsselattribut	Primärschlüssel (oder Sekundärschlüssel)

Motivation Normalformen

Artikel					
<u>artikelnr</u>	bezeichnung	preis	lagernr	lagerort	lagerstrasse
211	Radkappe	25	20	Konstanz	Seestrasse
333	Pumpe	100	15	Stuttgart	Hauptstrasse
655	Federbein	250	15	Stuttgart	Hauptstrasse
225	Kugellager	190	15	Stuttgart	Hauptstrasse

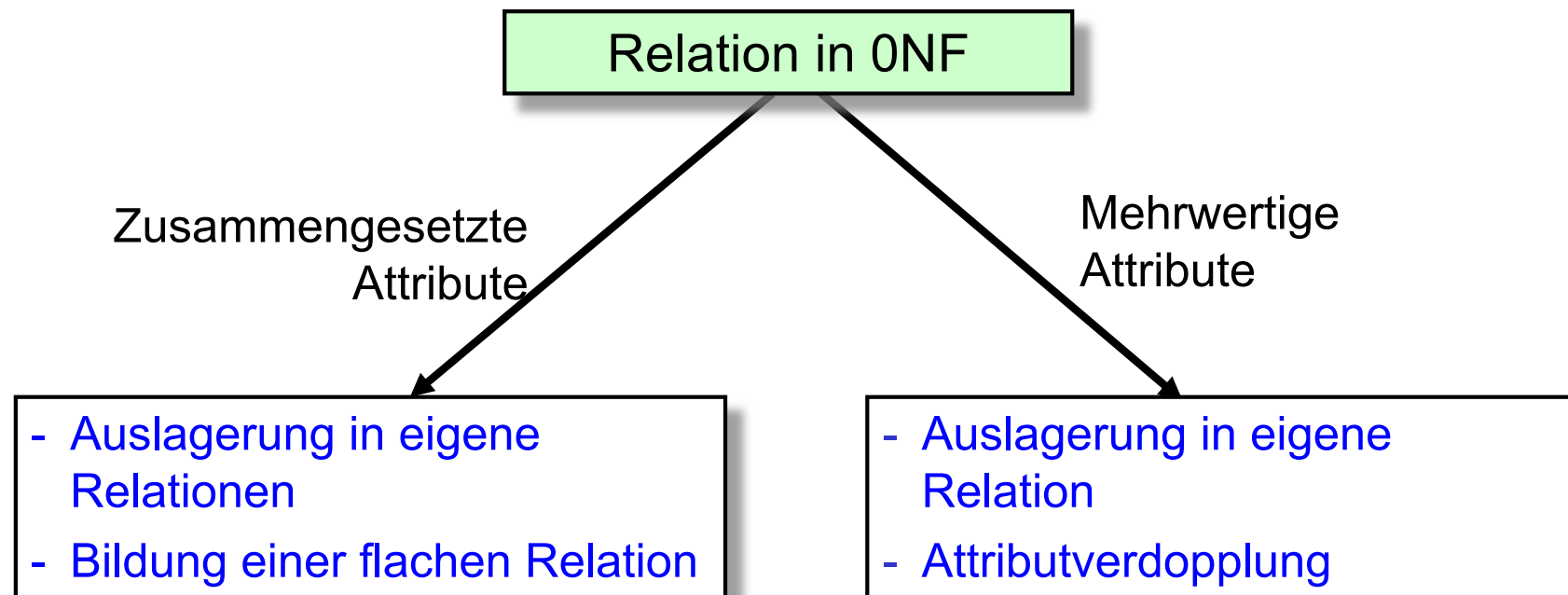
- Einfügeanomalie
 - Ein neues Lager kann nur dann aufgenommen werden, wenn bereits ein Artikel zugeordnet ist
- Änderungsanomalie
 - Wenn die Strasse des Lagers in Stuttgart geändert wird, müssen mehrere Tupel geändert werden
- Löschanomalie
 - Wenn die Radkappe aus der Artikelliste gelöscht wird, wird auch das Lager in Konstanz gelöscht

Flache Relationen

- Flache Relationen sind Relationen, die
 - Keine mehrwertigen Attribute und
 - Keine zusammengesetzte Attribute enthalten
- Beispiel für mehrwertiges Attribut
 - Student = ({ matrnr, name, {telnr} })
- Beispiel für zusammengesetztes Attribut
 - Person = ({ pnr, name, (plz, stadt, strasse, hnr) })

1. Normalform.

- Eine Relation ist in erster Normalform (1NF), wenn jeder Attributwert elementar ist
 - d.h. es existieren keine matrix-, listen-, mengenwertige Attribute
- Normalisierung:



Redundanz und Konsistenz

- Redundanz (redundancy)
 - Maß für „Überflüssigkeit“ von Informationen
 - Redundante Information: mehrfaches Vorhandensein gleicher Information
- Änderungen von redundanten Informationen führen häufig zu Inkonsistenzen

Funktionale Abhängigkeiten

- X, Y sind Attributmengen, R ein Relationsschema
- Y heißt *funktional abhängig* von X in R , geschrieben $X \rightarrow Y$, wenn es in jeder Relation zu R keine zwei Tupel gibt, die in ihrem Wert unter X , aber nicht in ihrem Wert unter Y übereinstimmen
- Y heißt *voll funktional abhängig* von X in R , geschrieben $X \twoheadrightarrow Y$, wenn X minimal ist, d.h. wenn es keine Teilmenge aus X gibt, die bereits Y funktional bestimmt
- Triviale funktionale Abhängigkeiten: $X \rightarrow Y$ mit $Y \subseteq X$

2. Normalform

- Eine Relation ist in zweiter Normalform (2NF), wenn sie
 - in 1NF ist und
 - jedes Nichtschlüsselattribut von jedem Schlüsselkandidaten voll funktional abhängig ist
- Informale Beschreibung
 - Ein Relationenschema verletzt die 2NF, wenn Informationen über mehr als ein einziges Konzept modelliert werden
- Normalisierung durch
 - Auslagerung nicht abhängiger Schlüsselattribute in eigene Tabellen

arbeitetIn					
<u>projektnr</u>	<u>personalnr</u>	projektname	name	vorname	seit
P01	2004	Carrera	Müller	Hans	01.10.2000
P02	1208	Coyote	Zimmer	Jochen	01.01.1999
P02	1001	Coyote	Abel	Kai	01.09.1995
P02	2004	Coyote	Müller	Hans	01.12.2000

3. Normalform

- Eine Relation ist in dritter Normalform (3NF), wenn sie
 - in 2NF ist und
 - kein Nichtschlüsselattribut transitiv von einem Schlüsselattribut abhängt
- Anschaulicher: In der Menge der Nichtschlüsselattribute darf es keine nicht trivialen funktionale Abhängigkeiten geben

Artikel					
<u>artikelnr</u>	bezeichnung	preis	lagernr	lagerort	lagerstrasse
211	Radkappe	25	20	Konstanz	Seestrasse
333	Pumpe	100	15	Stuttgart	Hauptstrasse
655	Federbein	250	15	Stuttgart	Hauptstrasse
225	Kugellager	190	15	Stuttgart	Hauptstrasse



Boyce-Codd Normalform

- Eine Relation ist in Boyce-Codd Normalform (BCNF), wenn sie
 - in 3NF ist und
 - keine transitiven Abhängigkeiten der Schlüsselattribute existieren
- Anschaulicher: In der Menge der Schlüsselattribute darf es keine funktionale Abhängigkeiten geben
- Beispiel Relation Adresse:
 - Adresse = {plz, ort, bundesland, hauptstadt}
 - Schlüsselkandidaten: plz, {ort, bundesland}, {ort, hauptstadt}
 - Transitive Abhängigkeiten: plz \twoheadrightarrow bundesland \twoheadrightarrow hauptstadt

*A table is based on
the key,
the whole key,
and nothing but the key (so help me Codd)*

(William Kent, "A Simple Guide to Five Normal Forms in Relational Database Theory")

Normalformen

Beispiel

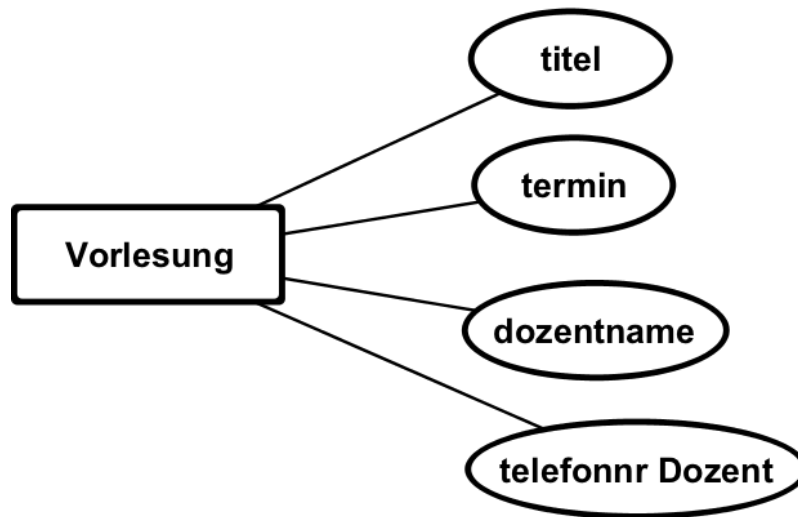
Gegeben ist ein Realweltausschnitt „Reisebüro“ mit folgender Relation:

Reisebuchung = ({buchungs-nr, reisebuero-nr, reisebuero-name, kunden-nr, kunden-nachname, kunden-vorname, kunden-wohnt, {kunden-tel-nr}, buchungs-datum, reise-nr, reise-beschreibung})

- (a) In welcher Normalform befindet sich diese Relation? Begründen Sie kurz Ihre Antwort.
- (b) Überführen Sie die Relation in die BCN Normalform. Legen Sie dabei auch die Primärschlüssel sinnvoll durch Unterstreichen der jeweiligen Attribute fest.

Kurzwiederholung ER-Modell

- Durch eine korrekte ER-Modellierung Abbildung ins Relationenmodell erhält man gleich Relationen in BCNF
- Relationen welcher Normalform werden aus folgendem Modell gebildet?



Relationale Algebra

Übersicht

- Selektion σ
 - Auswahl von Tupel aus einer Relation
- Projektion π
 - Auswahl auf ausgewählte Attribute einer Relation
- Verbund \bowtie
 - Verbindung mehrerer Relationen über gemeinsame Attribute

Selektion und Projektion

- Selektion σ : Auswahl von Tupel aus einer Relation

– $\sigma_{\text{name} = \text{„Müller“}}(\text{Student})$

Student		
<u>matrikelnr</u>	name	wohnort
132004	Müller	Singen

- Projektion π : Auswahl von Spalten (Attribute) aus einer Relation

– $\pi_{\text{name, wohnort}}(\text{Student})$

name	wohnort
Müller	Singen
Zimmer	Lindau

Beispiel zu relationale Operationen

Student = ({ matrikelnr, name, sg})

Studium = ({ sg, abschluss})

Student		
<u>matrikelnr</u>	name	<u>sg</u>
134711	Müller	AIN
138801	Kunz	WIN
131294	Maier	MSI

Studium	
<u>sg</u>	abschluss
AIN	Bachelor of Science
WIN	Bachelor of Science
MSI	Master of Science

Welche Matrikelnummer hat der Student mit Namen Müller?

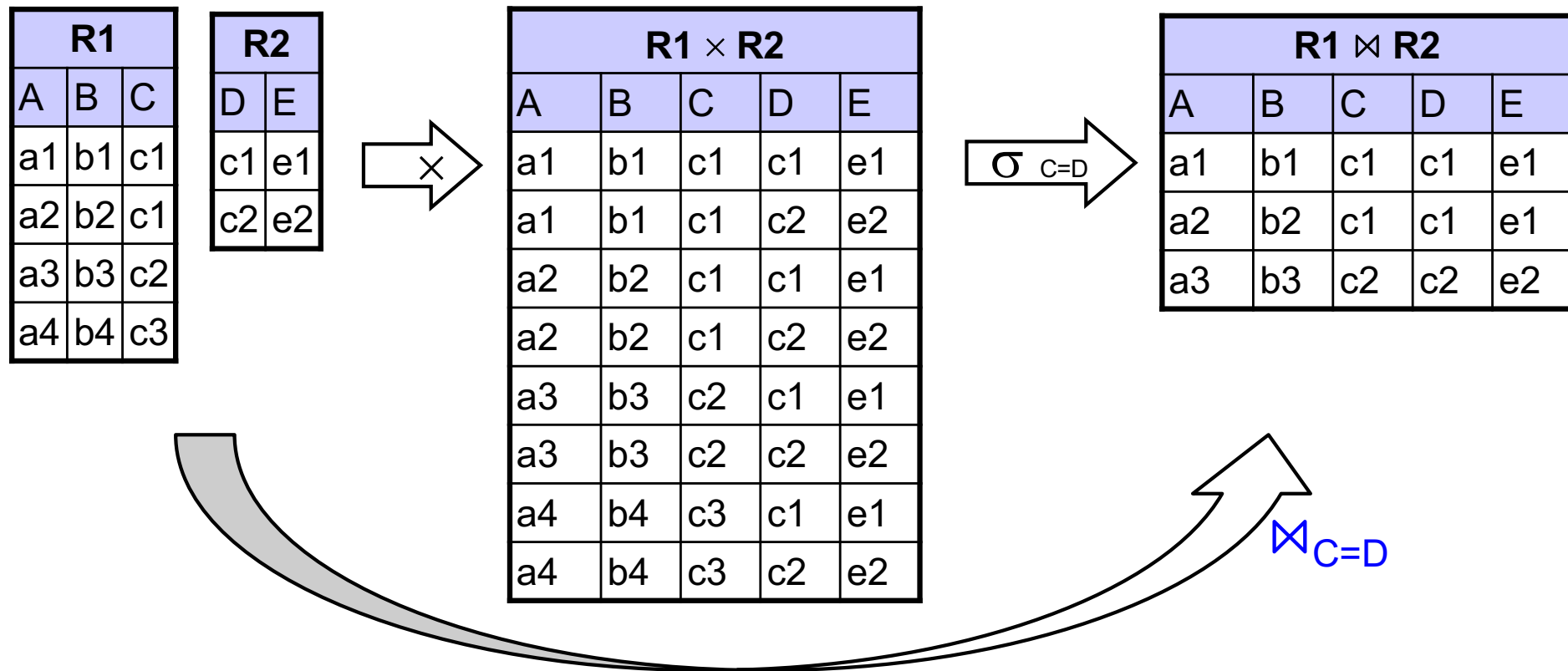
Join-Operation

- Join (Verbund)
 - Kartesisches Produkt zw. Relationen, eingeschränkt durch gemeinsame Attribute
 - Inverse ist Projektion
 - Symbol \bowtie
- Definition Theta-Join (Θ -Join, Theta-Verbund)
 - $R \underset{A \Theta B}{\bowtie} S = \sigma_{A \Theta B} (R \times S), \quad \Theta = \{ <, =, >, \leq, \neq, \geq \}$
- Andere Schreibweise
 - $R (A \Theta B) S$

Equi-Join (Gleich-Verbund)

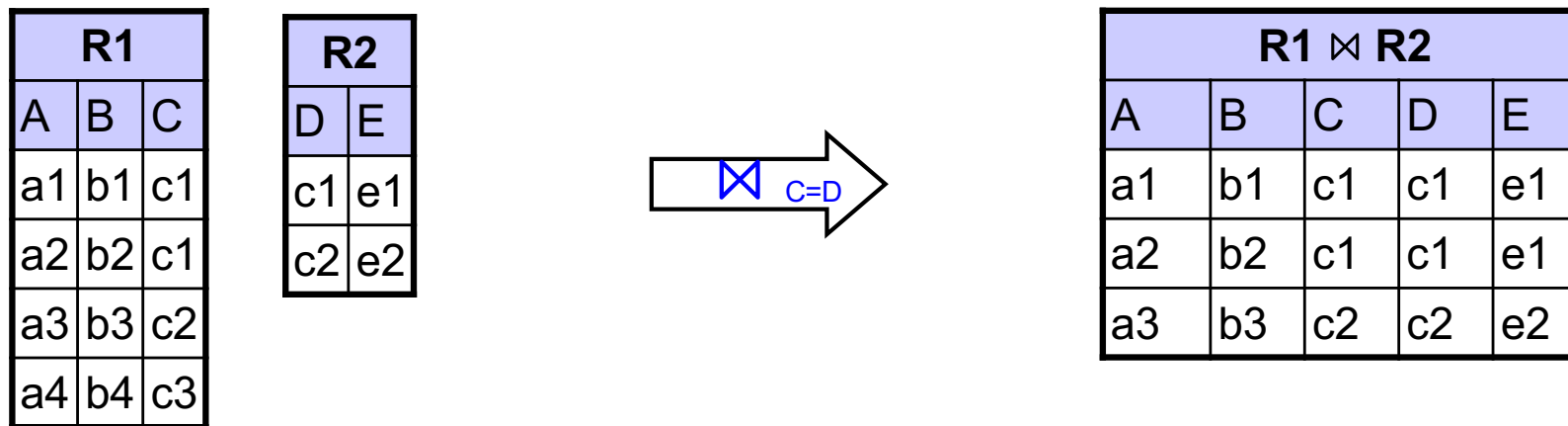
- Join = Kartesisches Produkt zw. Relationen, eingeschränkt durch Gleichheit zwischen Attributen

$$- R1 \bowtie_{C=D} R2 = \sigma_{C=D} (R1 \times R2)$$



Eigenschaften Join

- Eigenschaften
 - Assoziativ: $(A \bowtie B) \bowtie C = A \bowtie (B \bowtie C)$
 - Kommutativ: $A \bowtie B = B \bowtie A$
- Equi-Join (Gleich-Verbund)
 - Kartesisches Produkt zw. Relationen, eingeschränkt durch Gleichheit zwischen Attributen

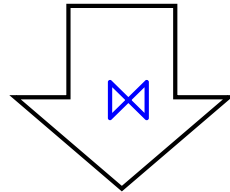


Beispiel Equi-Join

Student	
<u>matrikel-nr</u>	name
260111	Hans Müller
260112	Kai Maier
260113	Egon Schneider

besucht	
<u>matrikel-nr</u>	<u>vorl-nr</u>
260111	246
260111	241

Vorlesung	
<u>vorl-nr</u>	name
246	Datenbanksysteme
241	Systemmodellierung

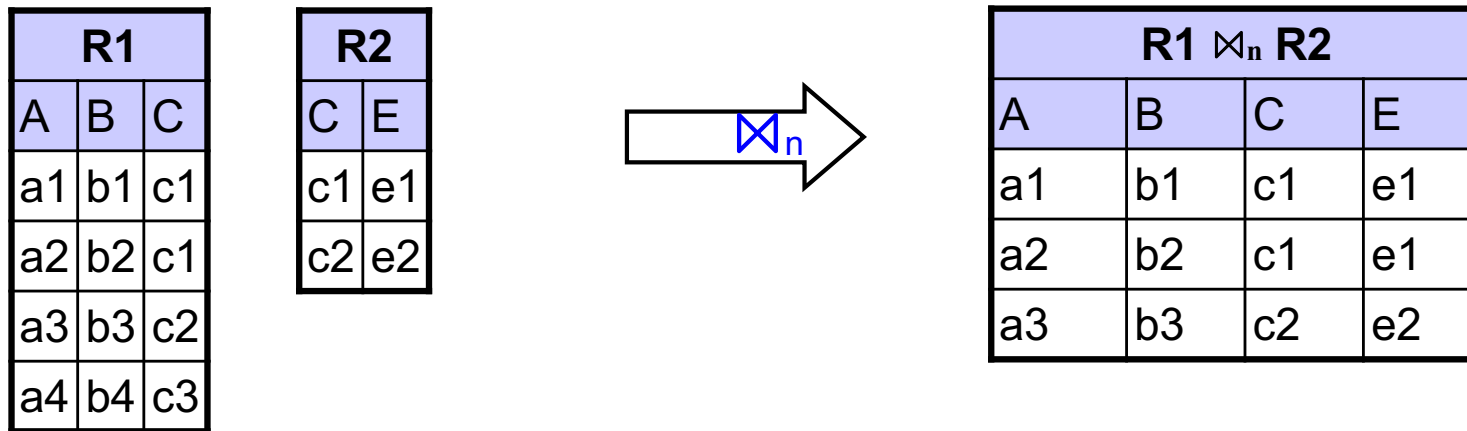


Student.Matrikel-Nr = besucht.Matrikel-Nr
 besucht.Vorl-Nr = Vorlesung.Vorl-Nr

Student ⋈ besucht ⋈ Vorlesung					
Student		besucht		Vorlesung	
matrikel-nr	name	matrikel-nr	vorl-nr	vorl-nr	name
260111	Hans Müller	260111	246	246	Datenbanksysteme
260111	Hans Müller	260111	241	241	Systemmodellierung

Natural Join (natürlicher Verbund)

- Gleichsetzen aller Attribute, die gleich heißen
- Entfernung der redundanten Attribute

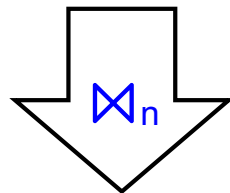


Beispiel Natural Join

Student	
<u>matrikelnr</u>	name
260111	Hans Müller
260112	Kai Maier
260113	Egon Schneider

besucht	
<u>matrikelnr</u>	<u>vorl-nr</u>
260111	246
260111	241

Vorlesung	
<u>vorl-nr</u>	name
246	Datenbanksysteme
241	Systemmodellierung



Probleme?

Student ⋈ₙ besucht ⋈ₙ Vorlesung			
matrikelnr	name	vorl-nr	name
260111	Hans Müller	246	Datenbanksysteme
260111	Hans Müller	241	Systemmodellierung

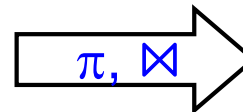
Beispiel Join-Operation

- Beispiel: Welche Studenten machen welchen Abschluss?

— $\pi_{\text{name, abschluss}} \bowtie_{\text{sg} = \text{sg}} (\text{Student}, \text{Studium})$

Student		
<u>matrikelnr</u>	name	<u>sg</u>
134711	Müller	AIN
138801	Kunz	WIN
131294	Maier	MSI

Studium	
<u>sg</u>	abschluss
AIN	Bachelor of Science
WIN	Bachelor of Science
MSI	Master of Science



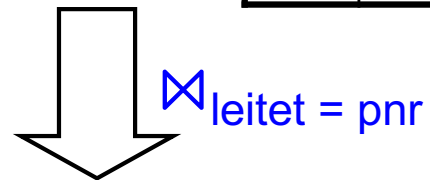
name	abschluss
Müller	Bachelor of Science
Kunz	Bachelor of Science
Maier	Master of Science

Eigenschaften Inner-Joins

Ziel: Auflistung aller Mitarbeiter und der Projekte, die sie leiten:

Mitarbeiter		
<u>mnr</u>	name	<u>leitet</u>
4711	Müller	44
8801	Schmidt	33
1288	Huber	
1294	Maier	42

Projekt	
<u>pnr</u>	projektname
33	Projekt 1
44	Projekt 2
55	Projekt 3
42	Projekt 4

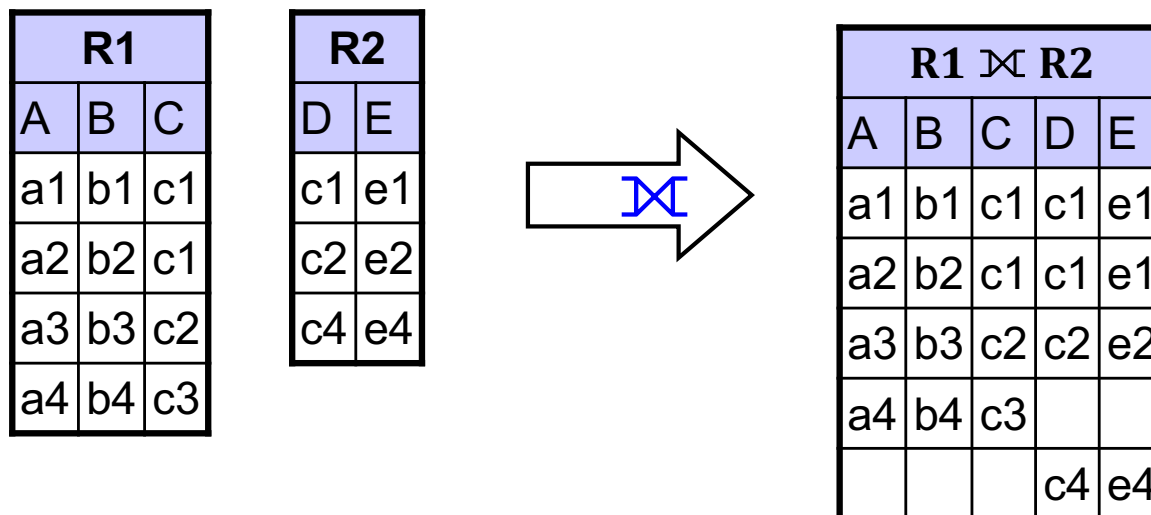


Mitarbeiter ⋈ Projekt				
<u>mnr</u>	name	leitet	pnr	projektname
4711	Müller	44	44	Projekt 2
8801	Schmidt	33	33	Projekt 1
1294	Maier	42	42	Projekt 4

Mitarbeiter
Huber fehlt !

Outer Joins (1)

- Full Outer Join
 - Bei einem full outer join zwischen Relationen R1 und R2 werden auch dann Tupel aus R1 (bzw. R2) in das Ergebnis aufgenommen, für die es kein passendes Tupel in R2 (bzw. R1) gibt
 - Fehlende Attributwerte werden mit Nullwerten aufgefüllt
 - Symbol \bowtie



Outer Joins (2)

- Left Outer Join von A, B
 - Der left outer join ist ein Verbund vereinigt mit den Tupeln aus A
 - Symbol \bowtie
- Right Outer Join von A, B
 - Der right outer join ist ein Verbund vereinigt mit den Tupeln aus B
 - Symbol \bowtie

Aufgabe zu Outer Join

Gegeben seien die folgenden Tabellen L und R. Geben Sie für die folgenden Join-Verknüpfungen die Ergebnistabellen an.

a) $L \bowtie_{L.a = R.d} R$

b) $L \bowtie_{L.b = R.d} R$

c) $L \bowtie_{L.a = R.c} R$

L	
a	b
8	2
15	16
40	16

R	
c	d
2	2
8	40
8	16

Beispiel zu relationale Operationen

Student = ({ matrikelnr, name, sg})

Studium = ({ sg, abschluss})

Student		
<u>matrikelnr</u>	name	<u>sg</u>
134711	Müller	AIN
138801	Kunz	WIN
131294	Maier	MSI

Studium	
<u>sg</u>	abschluss
AIN	Bachelor of Science
WIN	Bachelor of Science
MSI	Master of Science

Welchen Abschluss macht der Student Müller?

Beispiel zu relationale Operationen

Student = ({ matrikelnr, name, sg})

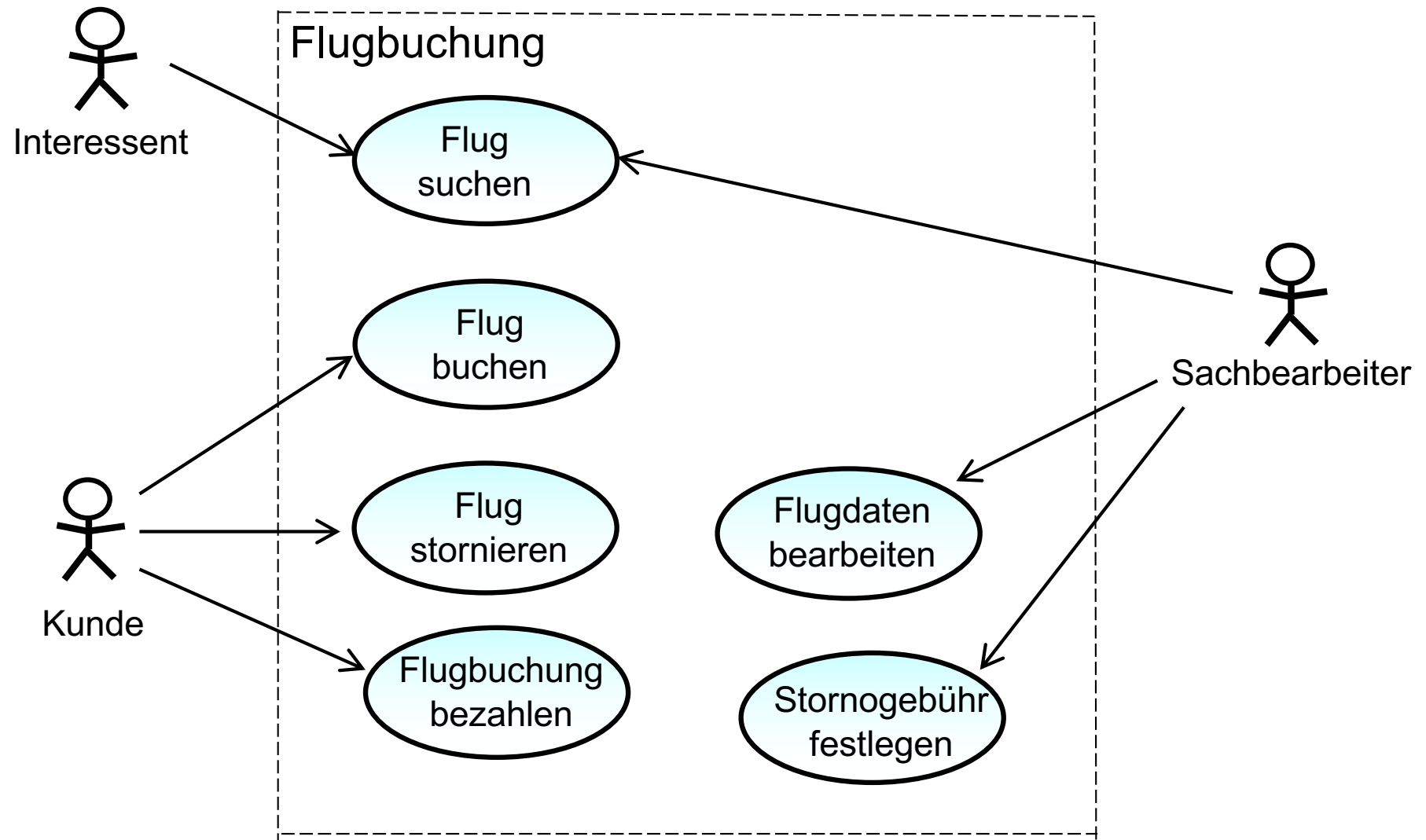
Studium = ({ sg, abschluss})

Student		
<u>matrikelnr</u>	name	<u>sg</u>
134711	Müller	AIN
138801	Kunz	WIN
131294	Maier	MSI

Studium	
<u>sg</u>	abschluss
AIN	Bachelor of Science
WIN	Bachelor of Science
MSI	Master of Science

Welche Studiengänge haben keinen einzigen Studenten?

Beispiel für Use-Case-Diagramm



Modellierung der Zugriffsrechte

- Darstellung als Zugriffsmatrix
 - Zugriffsrechte können aus Use Case-Diagramm abgeleitet werden
 - Subjekte sind einzelne Benutzer bzw. Benutzergruppen
 - Objekte können Entitytypen, Entities oder Attribute sein
 - Prinzip des kleinstmöglichen Privilegs:
Welche Rechte benötigt ein Benutzer mindestens, um einen Use Case auszuführen?
 - Zugriffsrechte = {read, write, delete}

Objekt Subjekt	Flug	Flug- hafen	Flugbuchung(flugNr, datum kundenNr, von, bis)	Flugbuchung (stornogebühr)
Kunde	r	r	r, w	r
Sachbearbeiter	r, w, d	r, w, d	r, w, d	r, w

User Interface

- User Interface für Datenfelder von Tabellen
 - Eingabefelder für Attribute
 - Auswahlliste (Listbox) zur Spezifikation von Relationships
 - Berücksichtigung der Kardinalität von Relationships (Combobox)

Student Enrollment

Add Enrollment

StudentID: 17625

Last Name: King

First Name: Steven

<input type="checkbox"/>	C101	Intro Computing	2012
<input checked="" type="checkbox"/>	C102	Intro Programming	2012
<input type="checkbox"/>	C101	Fundamentals	2012
<input type="checkbox"/>	C108	Databases	2012

Record: C303

No Filter Search

Quelle: C. Churcher: Beginning Database Design