

1. Die Bedeutung des Datenmanagements
2. Datenbank-Architektur
3. Modellierung und Entwurf von DB-Systemen
- 4. Relationale Algebra und Normalisierung**
5. Definition und Abfrage von Datenbank-Systemen
6. Dateiorganisation und Zugriffsstrukturen
7. Optimierung von Anfragen
8. Transaktionen

1. Die Relationale Algebra

1. *Übersicht*
2. *Mengenorientierte Operatoren*
3. *Relationenorientierte Operatoren*
4. *Änderungsoperationen*

2. Die Normalisierung relationaler Datenbanken

1. *Einführung*
2. *Mutationsanomalien*
3. *Was ist Normalisierung?*
4. *Gründe für die Normalisierung*
5. *Übersicht über die Normalformen*
6. *Nullte Normalform (0 NF)*

7. *Erste Normalform (1 NF)*
8. *Klassifizierung der Attribute nach Abhängigkeiten*
9. *Zweite Normalform (2 NF)*
10. *Dritte Normalform (3 NF)*
11. *Zusammenfassung*
12. *Beurteilung der Normalisierung*

- TODO: Erst Normalisierung, dann rel. Algebra

Die Relationale Algebra (Relationenalgebra)

- bildet den formalen Rahmen für relationale Datenbanksprachen
- definiert einen Satz von algebraischen Operationen, die auf Tabellen wirken
- wird eingeteilt in:
 - Mengenoperationen (Mengenorientierte Operatoren)
 - Relationsoperationen (Relationenorientierte Operatoren)
 - Änderungsoperationen (kein echter Bestandteil der RA)

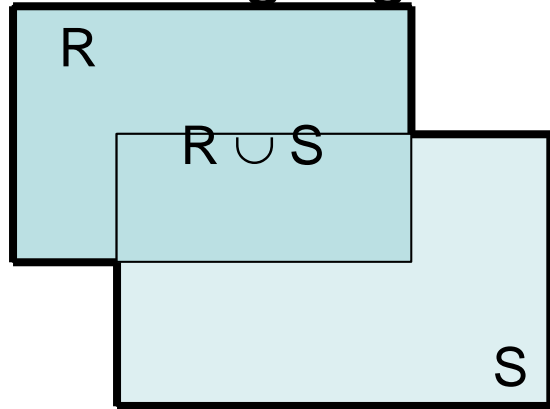
- Relationale Datenbanksprachen, die die Operationen der Relationenalgebra sinngemäß umsetzen, heißen **relational vollständige Sprachen**.
- Die relationale Algebra (Relationenalgebra) setzt Relationen (Tabellen) in der ersten Normalform voraus (nur atomare Werte für Attribute, siehe später).

- Vereinigung $R \cup S$
- Durchschnitt $R \cap S$
- Differenz $R \setminus S$ (oder auch $R-S$)
- Kartesisches Produkt $R \times S$

Voraussetzungen:

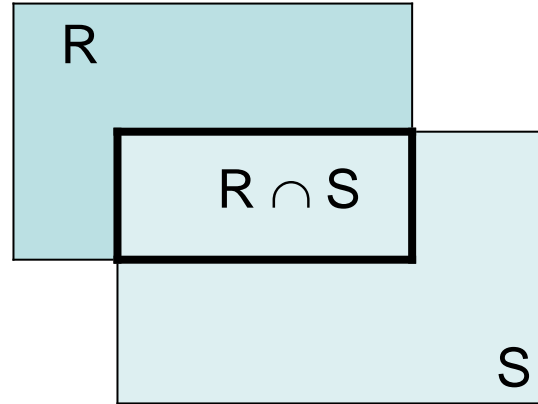
- Jede Tabelle entspricht einer Menge von Datensätzen, somit sind mengentheoretische Verknüpfungen von Tabellen möglich.
- Bedingung für Verknüpfungen ist das Verträglichkeitskriterium, d.h. Tabellen müssen die gleiche Anzahl von Attributen aufweisen und die Datenformate der korrespondierenden Merkmalskategorien müssen identisch sein.
- Die Ergebnistabelle hat dann auch die gleiche Anzahl Attribute mit den gleichen Domänen.

Vereinigung



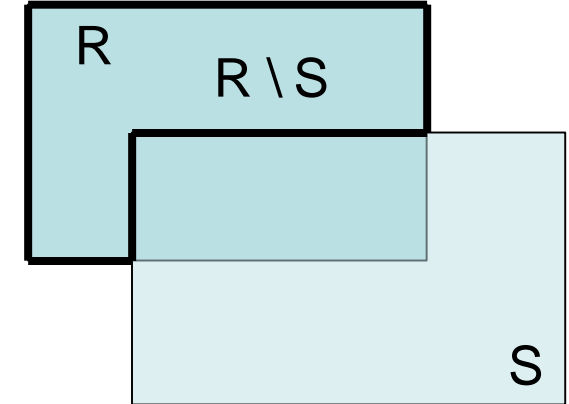
$$\{t \mid t \in R \text{ or } t \in S\}$$

Durchschnitt

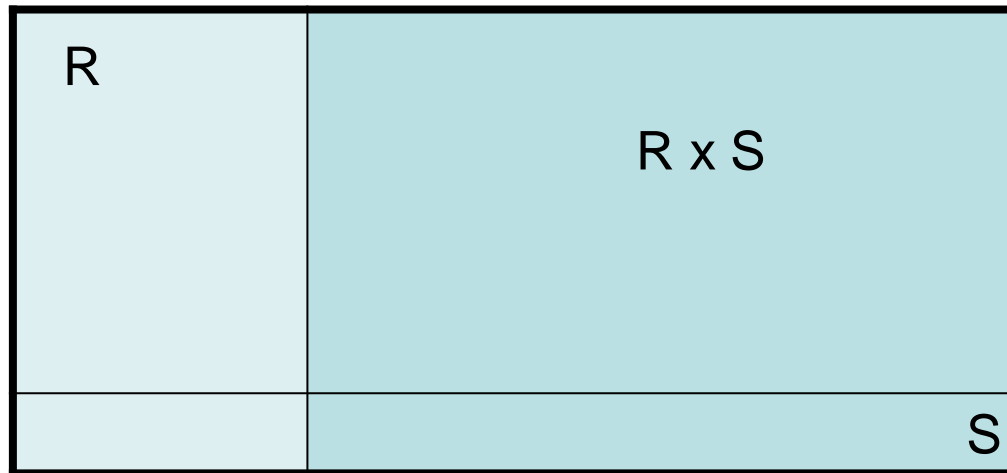


$$\{t \mid t \in R \text{ and } t \in S\}$$

Differenz



$$\{t \mid t \in R \text{ and } t \notin S\}$$



Kartesisches Produkt

$$\{(x,y) : R(x) \wedge S(y)\}$$

Vereinigung ($R \cup S$):

- Die Vereinigungstabelle enthält alle Datensätze, die in R **oder** in S enthalten sind.
- Da es sich um die mengentheoretische Vereinigung handelt, werden Duplikate dabei eliminiert.
- Zur Anwendung der Vereinigung müssen R und S gleiche Stelligkeit und verträgliche Domänen haben.

Durchschnitt ($R \cap S$):

- Die Durchschnittstabelle von R und S enthält alle Datensätze (Tupel), die in R **und** in S enthalten sind.
- Alle Datensätze (Tupel) sind nur einmal enthalten.
- Zur Anwendung des Durchschnitts müssen R und S gleiche Stelligkeit und verträgliche Domänen haben.

Differenz ($R \setminus S$):

- Die Ergebnistabelle der Differenzoperation zwischen R und S enthält alle Datensätze (Tupel), die in R **und** nicht in S enthalten sind.
- Zur Anwendung des Durchschnitts müssen R und S gleiche Stelligkeit und verträgliche Domänen haben.

4.1.2 Mengenorientierte Operatoren

Tab. Fußball

<u>MNr</u>	Name	Ort
1	Meier	WÜ
2	Müller	N
3	Huber	B

Tab. Volleyball

<u>MNr</u>	Name	Ort
8	Amsel	M
9	Wicht	HH
3	Huber	B

Ausgangstabellen

Vereinigung $F \cup V$

<u>MNr</u>	Name	Ort
1	Meier	WÜ
2	Müller	N
3	Huber	B
8	Amsel	M
9	Wicht	HH

Mengendifferenz $F - V$

<u>MNr</u>	Name	Ort
1	Meier	WÜ
2	Müller	N

Kartesisches Produkt ($R \times S$):

- Das Kartesische Produkt $R \times S$ zwischen R und S ist die Menge aller möglichen Kombinationen (x,y) , die sich aus den Tupeln x aus R und y aus S bilden lassen.
- Zur Anwendung des Kartesischen Produktes muss das Verträglichkeitskriterium nicht gelten, d.h. die Attribute der beteiligten Relationen müssen nicht gleiche Stelligkeit und verträgliche Domänen haben.

Kartesisches Produkt: $R \times S$

Beispiel:

Gegeben seien die Relationen

$$R = \{(1, \text{Miriam}), (2, \text{Michael}), (3, \text{Katrin})\}$$

und

$$S = \{(1, \text{Julia}), (3, \text{Toni})\}$$

$$R \times S = \{(1, \text{Miriam}, 1, \text{Julia}), (1, \text{Miriam}, 3, \text{Toni}), (2, \text{Michael}, 1, \text{Julia}), \\ (2, \text{Michael}, 3, \text{Toni}), (3, \text{Katrin}, 1, \text{Julia}), (3, \text{Katrin}, 3, \text{Toni})\}$$

- Sportclub

<u>MNr</u>	Name	Straße	Ort
M1	Meier	Lindstraße	Liestal
M7	Huber	Mattenweg	Basel
M19	Schweizer	Hauptstraße	Zürich

- Fotoclub

<u>MNr</u>	Mitglied	Straße	Ort
M4	Becker	Wasserweg	Liestal
M7	Huber	Mattenweg	Basel

- Clubmitglieder = Sportclub \cup Fotoclub

<u>MNr</u>	Name	Straße	Ort
M1	Meier	Lindstraße	Liestal
M7	Huber	Mattenweg	Basel
M19	Schweizer	Hauptstraße	Zürich
M4	Becker	Wasserweg	Liestal

- Clubmitglieder = Sportclub \cap Fotoclub

<u>MNr</u>	Name	Straße	Ort
M7	Huber	Mattenweg	Basel

- Clubmitglieder = Sportclub \ Fotoclub

<u>MNr</u>	Name	Straße	Ort
M1	Meier	Lindstraße	Liestal
M19	Schweizer	Hauptstraße	Zürich

- Kartesisches Produkt:
- Wettkampfpaarungen: (Sportclub \ Fotoclub) x Fotoclub

<u>MNr</u>	Name	Straße	Ort	<u>MNr</u>	Mitglied	Straße	Ort
M1	Meier	Lindstraße	Liestal	M4	Becker	Wasserweg	Liestal
M1	Meier	Lindstraße	Liestal	M7	Huber	Mattenweg	Basel
M19	Schweizer	Hauptstraße	Zürich	M4	Becker	Wasserweg	Liestal
M19	Schweizer	Hauptstraße	Zürich	M7	Huber	Mattenweg	Basel

Führen Sie folgende Mengenoperation durch:
 $(\text{STUDENT} \cap \text{INSTRUCTOR}) \times (\text{INSTRUCTOR} - \text{STUDENT})$

STUDENT	FN	LN
	Susan	Yao
	Ramesh	Shah
	Johnny	Kohler
	Barbara	Jones
	Amy	Ford
	Jimmy	Wang
	Ernest	Gilbert

INSTRUCTOR	FNAME	LNAME
	John	Smith
	Ricardo	Browne
	Susan	Yao
	Francis	Johnson
	Ramesh	Shah

- Selektion von Zeilen aus einer Tabelle R mittels einer Formel F: $\sigma_F(R)$
- Projektion der Tabelle R auf eine Menge von Merkmalen M: $\pi_M(R)$
(Ein Merkmal entspricht einem Attribute)
- Verbund zweier Tabellen R und S durch Prädikat P:
 $R \bowtie_P S$
- Division der Tabelle R durch die Teiltabelle S: $R \div S$

Selektion (Select):

- Auswahl von Zeilen
- Auslesen von Datensätzen aus einer Tabelle, die bestimmten Kriterien entsprechen
- Extrahiert aufgrund einer Bedingung Tupel aus einer Relation.

Selektion (Select):

Es sollen nur die Verkäufer angezeigt werden, deren Umsatz größer als 300 ist.

Tab. Verkäufer

<u>VNr</u>	Name	Umsatz
1	Meier	210
2	Müller	310
3	Huber	180
8	Amsel	100
9	Wicht	350



<u>VNr</u>	Name	Umsatz
2	Müller	310
9	Wicht	350

Select **VNr**, **Name**, **Umsatz** from Verkäufer **where Umsatz > 300**

Selektion (Select): Beispiele

$\sigma_{(\text{Umsatz} < 1000 \text{ AND } \text{Umsatz} > 200)} (\text{Verkäufer})$

$\sigma_{(\text{Umsatz} \geq 100 \text{ OR } \text{Name} = \text{"Meier"})} (\text{Verkäufer})$

Projektion:

- Selektion von Spalten
- Auswahl bestimmter Spalten aus der Originaltabelle zur Bildung einer neuen Tabelle
- Extrahiert Attribute aus einer Relation

Projektion:

Es sollen nur die Verkäufernummer und der Name angezeigt werden.



<u>VNr</u>	Name	Umsatz
1	Meier	210
2	Müller	310
3	Huber	180
8	Amsel	100
9	Wicht	350

Tab. Verkäufer

<u>VNr</u>	Name
1	Meier
2	Müller
3	Huber
8	Amsel
9	Wicht

Select **VNr**, **Name** from Verkäufer

Beispiel: Projektion

Mitarbeiter

<u>MNr</u>	Name	Strasse	Ort	<u>ANr-Unt</u>
M1	Meier	Lindstrasse	Liestal	A3
M7	Huber	Mattenweg	Basel	A5
M19	Schweizer	Hauptstrasse	Zürich	A6
M4	Becker	Wasserweg	Liestal	A6

Beispiel: Projektion

π_{Ort} (Mitarbeiter)

Ort
Liestal
Basel
Zürich
Liestal

Beispiel: Projektion

$\pi_{\text{Ort, ANr-Unt}}(\text{Mitarbeiter})$

Ort	<u>ANr-Unt</u>
Liestal	A3
Basel	A5
Zürich	A6
Liestal	A6

Verbund (Join):

- Verknüpfung verschiedener Tabellen, die ein gleiches oder ähnliches Attribut besitzen
- Hier werden zwei Relationen anhand von Vergleichen zwischen Attributwerten der Tupel miteinander verknüpft. Dabei ist einzige Bedingung, dass die JOIN-Attribute aus dem selben Wertebereich sein müssen.

Natural Join (Natürlicher Verbund):

Tab. Mitarbeiter

<u>MNr</u>	Name	ProjNr
1	Meier	1
2	Müller	1
3	Huber	2
8	Amsel	3
9	Wicht	2

Tab. Projekte

<u>PNr</u>	PName
1	Planung
2	Kurs
3	Fete

<u>MNr</u>	Name	PNr	PName
1	Meier	1	Planung
2	Müller	1	Planung
3	Huber	2	Kurs
8	Amsel	3	Fete
9	Wicht	2	Kurs

```
Select MNr, Name, PNr, PName  
from Mitarbeiter  
Join Projekte  
on ProjNr=PNr;
```

Es sind auch beliebige Kombinationen aus Selektion, Projektion und Join möglich.

Beispiel-Daten für allgemeinen Verbund:

Mitarbeiter

<u>MNr</u>	Name	Straße	Ort	Unt
M19	Schweizer	Hauptstraße	Zürich	A6
M1	Meier	Lindstraße	Liestal	A3
M7	Huber	Mattenweg	Basel	A5
M4	Becker	Wasserweg	Liestal	A6

Abteilung

<u>ANr</u>	Bezeichnung
A3	Informatik
A5	Personal
A6	Finanz

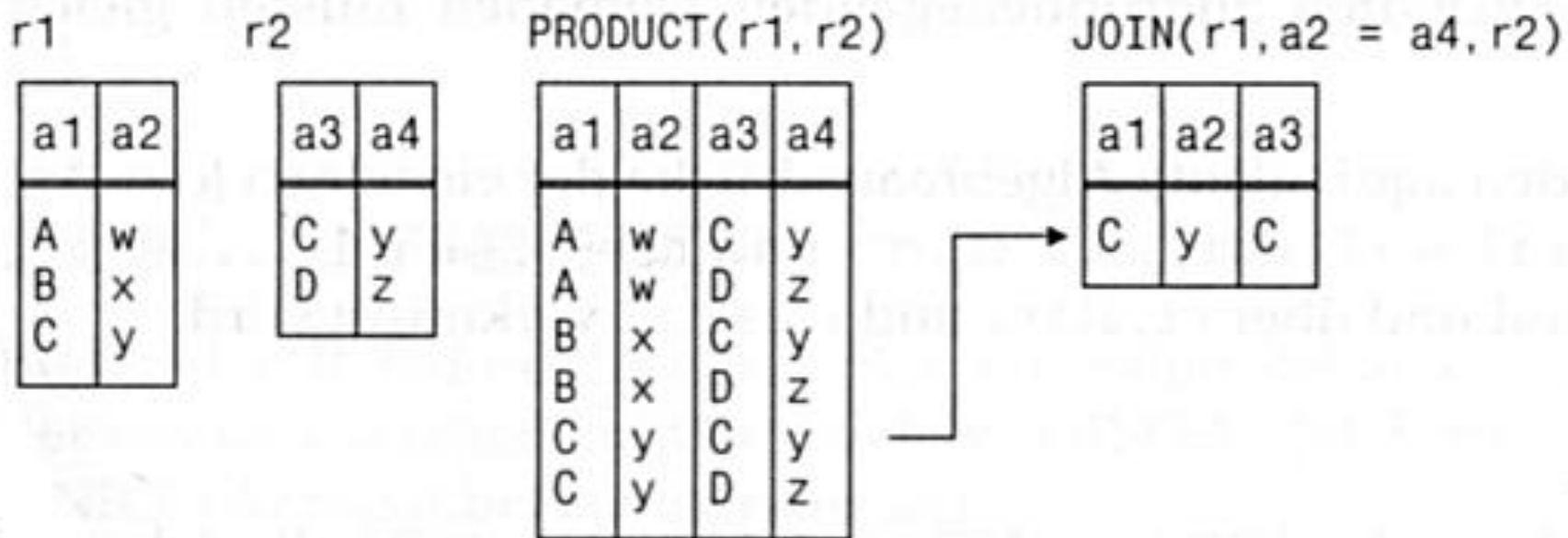
Verbund (Join):

Mitarbeiter ⋈_{Unt=ANr} **Abteilung**

<u>MNr</u>	Name	Straße	Ort	Unt	<u>ANr</u>	Bezeichnung
M19	Schweizer	Hauptstraße	Zürich	A6	A6	Finanz
M1	Meier	Lindstraße	Liestal	A3	A3	Informatik
M7	Huber	Mattenweg	Basel	A5	A5	Personal
M4	Becker	Wasserweg	Liestal	A6	A6	Finanz

Eine Join-Operation mit dem Gleichheitsoperator (=) wird als *Equijoin* bezeichnet.

Equijoin (Verbund mit Gleichheitsoperator):



Verbund (Join)

Es gilt folgende Formel:

$$\mathbf{R} \bowtie_P \mathbf{S} = \sigma_P (\mathbf{R} \times \mathbf{S})$$

Dies bedeutet, dass der Verbund von R und S mittels P durch das Kartesische Produkt von R und S plus anschließender Selektion mittels P definiert werden kann.

Beispiel-Daten für natürlichen Verbund:

Mitarbeiter

<u>MNr</u>	Name	Straße	Ort	<u>ANr</u>
M19	Schweizer	Hauptstraße	Zürich	A6
M1	Meier	Lindstraße	Liestal	A3
M7	Huber	Mattenweg	Basel	A5
M4	Becker	Wasserweg	Liestal	A6

Unt wurde in ANr umbenannt!

Abteilung

<u>ANr</u>	Bezeichnung
A3	Informatik
A5	Personal
A6	Finanz

Natürlicher Verbund (Natural Join):

Mitarbeiter * Abteilung

<u>MNr</u>	Name	Straße	Ort	<u>ANr</u>	Bezeichnung
M19	Schweizer	Hauptstraße	Zürich	A6	Finanz
M1	Meier	Lindstraße	Liestal	A3	Informatik
M7	Huber	Mattenweg	Basel	A5	Personal
M4	Becker	Wasserweg	Liestal	A6	Finanz

Der natürliche Verbund verknüpft die Tabellen über gleiche Werte in gleich benannten Spalten und entfernt doppelte Spalten in der Ergebnisrelation.

Drei-Wege-Join:

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

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
25403	5022

Vorlesungen			
VorlNr	Titel	SWS	gelesenVon
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

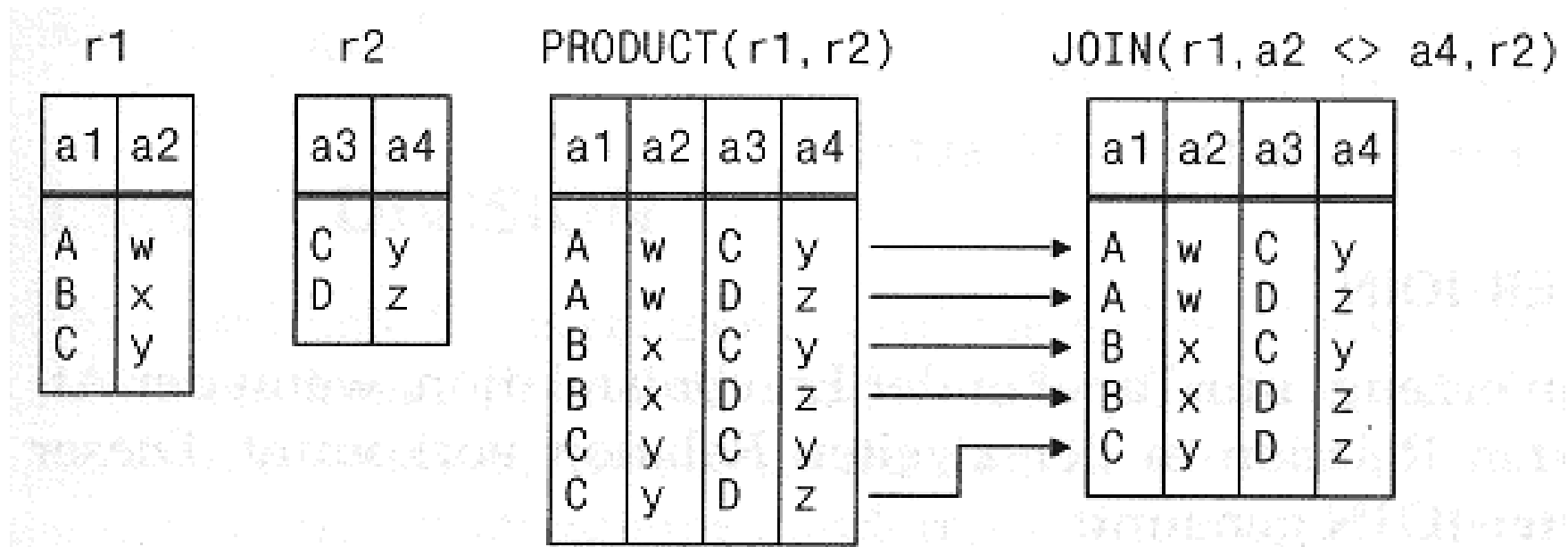
Drei-Wege-Join (hier mit natural joins):

(Studenten * hören) * Vorlesungen						
MatrNr	Name	Semester	VorlNr	Titel	SWS	gelesenVon
26120	Fichte	10	5001	Grundzüge	4	2137
25403	Jonas	12	5022	Glaube und Wissen	2	2134
28106	Carnap	3	5052	Wissenschaftstheorie	3	2126
...

Theta Join (Allgemeiner Verbund):

- Im Gegensatz zum natürlichen Verbund, bei dem die Werte gleicher Attribute übereinstimmen, werden beim Theta Join $>$, $<$, $<>$, $>=$, $<=$ als Operatoren eingesetzt.

Theta Join (Allgemeiner Verbund):



Theta Join (Allgemeiner Verbund):

Tabelle T1

P	Q	R
10	A	5
15	B	8
25	A	6

Tabelle T2

A	B	C
10	B	6
25	C	3
10	B	5

Lösen Sie folgende Operationen, d.h. geben Sie die resultierende Relation basierend auf obigen Daten an.

- $T1 \bowtie_{(T1.P = T2.A \text{ and } T1.R = T2.C)} T2$
- $T1 \bowtie_{(T1.R < T2.C)} T2$

Inner Join (Innerer Verbund):

- Alle bisher genannten Join-Operatoren werden auch Inner Joins genannt.
- Bei ihnen gehen die Tupel der Relation verloren, die keinen Join-Partner gefunden haben.

Outer Join (Äußerer Verbund):

- Bei den äußeren Join-Operatoren bleiben auch partnerlose Tupel erhalten und werden mit Null-Werten „aufgefüllt“.

Left Outer Join (Linker äußerer Verbund):

- Die Tupel der linken Relation bleiben erhalten.

Right Outer Join (Rechter äußerer Verbund):

- Die Tupel der rechten Relation bleiben erhalten.

Full Outer Join (Vollständiger äußerer Verbund):

- Die Tupel beider Relationen bleiben erhalten.

4.1.3 Relationenorientierte Operatoren

L		
A	B	C
a ₁	b ₁	c ₁
a ₂	b ₂	c ₂

R		
C	D	E
c ₁	d ₁	e ₁
c ₃	d ₂	e ₂

Inner Join				
A	B	C	D	E
a ₁	b ₁	c ₁	d ₁	e ₁

Left Outer Join				
A	B	C	D	E
a ₁	b ₁	c ₁	d ₁	e ₁
a ₂	b ₂	c ₂	-	-

Full Outer Join				
A	B	C	D	E
a ₁	b ₁	c ₁	d ₁	e ₁
a ₂	b ₂	c ₂	-	-
-	-	c ₃	d ₂	e ₂

Right Outer Join				
A	B	C	D	E
a ₁	b ₁	c ₁	d ₁	e ₁
-	-	c ₃	d ₂	e ₂

(basierend auf einem natural Join)

Division:

- Werden zwei Relationen miteinander dividiert, ergibt die Ergebnisrelation alle Tupel, für die es eine dazugehörige Wertmenge in der anderen Relation gibt.
- Es treten alle Attribute im Ergebnis auf, außer dem Attribut durch das geteilt wurde.

Division:

R	
M	V
m ₁	v ₁
m ₁	v ₂
m ₁	v ₃
m ₂	v ₂
m ₂	v ₃

÷

S
V
v ₁
v ₂

=

R ÷ S
M
m ₁

Gibt es die Möglichkeit, das in deutscher Sprache zu lesen?

Division:

R	
M	V
m ₁	v ₁
m ₁	v ₂
m ₁	v ₃
m ₂	v ₂
m ₂	v ₃

÷

S
V
v ₁
v ₂

=

R ÷ S
M
m ₁

Frage: „Für welches M (in R) existiert als V (in S) v₁ und v₂?“

Antwort: „Das existiert für m₁ und für kein anderes M.“

Division:

MNr	PNr
M1	P1
M1	P2
M1	P4
M2	P1
M2	P2
M4	P2
M4	P4

R

Tabelle der den Projekten zugeordneten Mitarbeiter.

$$R' = R \div S$$

MNr
M1
M4

S

PNr
P2
P4

Mitarbeiter, die gleichzeitig an Projekten P2 und P4 arbeiten.

4.1.3 Relationenorientierte Operatoren

Abbildung 5.6: Ein möglicher relationaler Datenbankzustand entsprechend dem Schema FIRMA.

ANGESTELLTER	VNAME	INITIAL	NNAME	SSN	GDATE	ADRESSE	GESCHLECHT	GEHALT	SUPERSSN	ABT
	John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia	J	Zelaya	999887777	1968-07-19	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
	James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1

ABT_STANDORT	ABTNUMMER	ASTANDORT
	1	Houston
	4	Stafford
	5	Bellaire
	5	Sugarland
	5	Houston

ABTEILUNG	ANAME	ABTNUMMER	MGRSSN	MGR_ANFANGSDATUM
	Research	5	333445555	1988-05-22
	Administration	4	987654321	1995-01-01
	Headquarters	1	888665555	1981-06-19

ARBEITET_AN	ESSN	PNR	STUNDEN
	123456789	1	32.5
	123456789	2	7.5
	666884444	3	40.0
	453453453	1	20.0
	453453453	2	20.0
	333445555	2	10.0
	333445555	3	10.0
	333445555	10	10.0
	333445555	20	10.0
	999887777	30	30.0
	999887777	10	10.0
	987987987	10	35.0
	987987987	30	5.0
	987654321	30	20.0
	987654321	20	15.0
	888665555	20	null

PROJEKT	PNAME	PNUMMER	PSTANDORT	ABTNR
	ProductX	1	Bellaire	5
	ProductY	2	Sugarland	5
	ProductZ	3	Houston	5
	Computerization	10	Stafford	4
	Reorganization	20	Houston	1
	Newbenefits	30	Stafford	4

ANGEHÖRIGER	ESSN	ANGEHÖRIGER_NAME	GESCHLECHT	GDATE	GRAD
	333445555	Alice	F	1986-04-05	DAUGHTER
	333445555	Theodore	M	1983-10-25	SON
	333445555	Joy	F	1958-05-03	SPOUSE
	987654321	Abner	M	1942-02-28	SPOUSE
	123456789	Michael	M	1988-01-04	SON
	123456789	Alice	F	1988-12-30	DAUGHTER
	123456789	Elizabeth	F	1967-05-05	SPOUSE

Beispieldaten
für die Division

Siehe Elmasri, Navathe (S. 144)

Wie heißen die Angestellten, die an **allen** Projekten arbeiten, an denen John Smith arbeitet?

(wird gemeinsam am Whiteboard entwickelt)

(siehe Elmasri, Navathe (S. 162))

- Sind keine echten Bestandteile der Relationenalgebra.
- Bei echter RA werden durch die Kombination neue Relationen erstellt aber keine bestehenden verändert.
- Änderungsoperationen sind erfolgreich, um Werte in Tabellen zu ändern.

Allgemein (unabhängig vom Datenmodell):

- Hinzufügen (Insert)
- Löschen (Delete)

Weitere Möglichkeiten der Datenmanipulation:

- Aktualisieren (Update)

Von allen genannten Operationen wird verlangt, dass ihre Ausführung die Integritätsbedingungen nicht verletzt!

Normalisierung

Gute Datenbankschemata

- ermöglichen die Herleitung aller benötigten Daten aus den Basisrelationen
- schränken die Möglichkeit inkonsistente Daten darzustellen weitgehend ein
- stellen die Daten möglichst redundanzfrei dar
- ermöglichen gute Performance bei Zugriffen
- verhindern Mutationsanomalien

- **Einfügeanomalien** (Insert-Anomalie)
- **Änderungsanomalien** (Update-Anomalie)
- **Löschanomalien** (Delete-Anomalie)

Mutationsanomalien (Beispiel):

Abteilungsmitarbeiter

MNr	Name	Straße	Ort	AbtNr	Bezeichnung
M19	Schweizer	Hauptstraße	Zürich	A6	Finanz
M1	Meier	Lindstraße	Liestal	A3	Informatik
M7	Huber	Mattenweg	Basel	A5	Personal
M4	Becker	Wasserweg	Liestal	A6	Finanz

Einfügeanomalie:

- Falls man eine neue Abteilung z.B. A9 mit dem Namen „Marketing“ definieren will, ist dies nur möglich, wenn man auch einen Mitarbeiter dafür erfasst, auch wenn es noch keinen gibt.
- Ein NULL-Wert für Mitarbeiter ist nicht möglich, da MNr Teil des Schlüssels ist.

Änderungsanomalie:

- Soll z.B. die Bezeichnung der Abteilung A6 von „Finanz“ in „Finance“ geändert werden, muss die Änderung bei sämtlichen Mitarbeitertupeln der Abteilung vollzogen werden, obwohl sich nur ein einziger Sachverhalt ändert.

Löschanomalie:

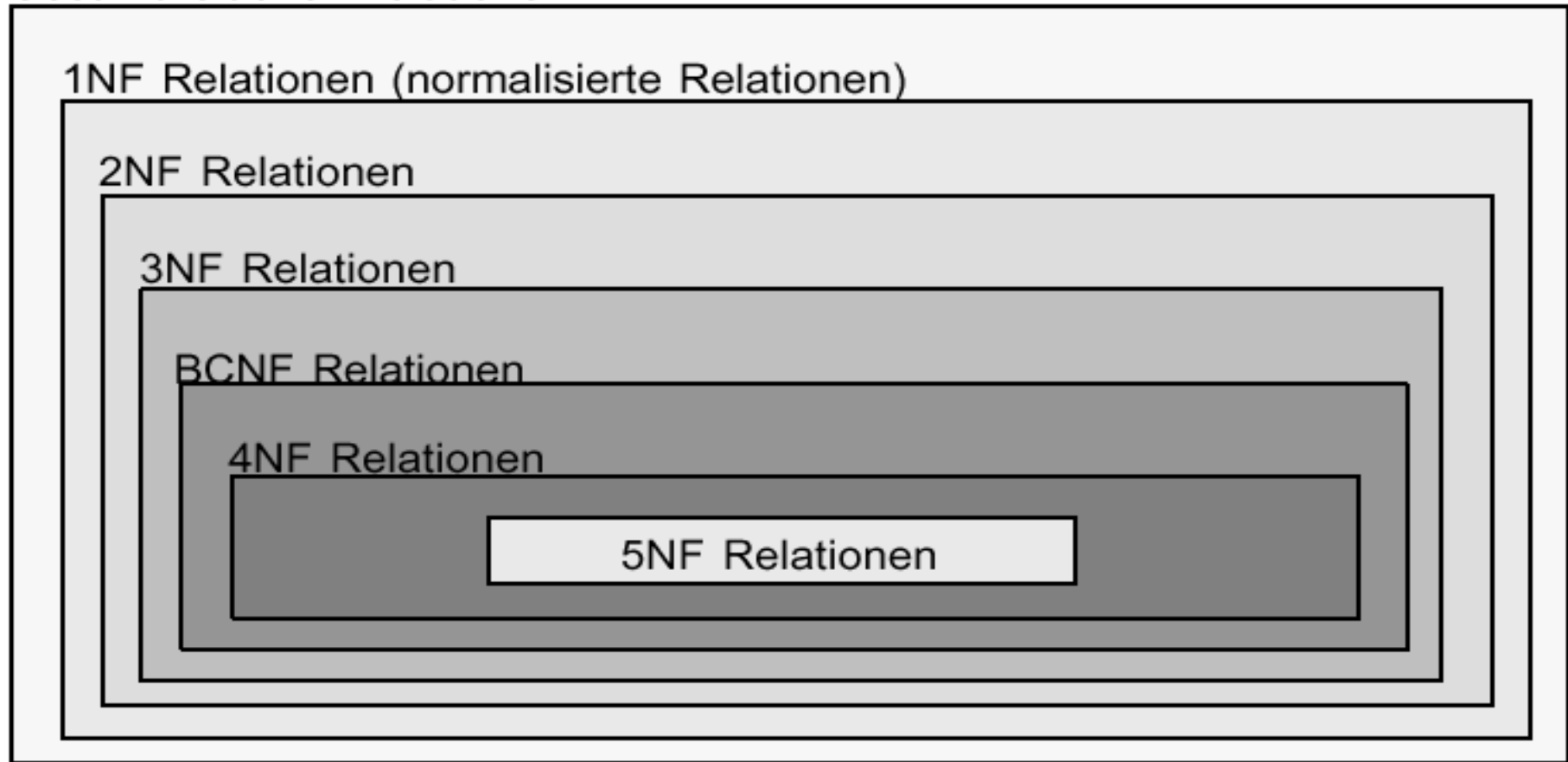
- Eine Löschanomalie liegt vor, wenn eine Information ungewollt verloren geht.
- Wenn man z.B: in der Tabelle „Abteilungsmitarbeiter“ Mitarbeiter „M1“ löscht, geht die Abteilung „Informatik“ mit der Abteilungsnummer „A3“ verloren.

4.2.3 Was ist Normalisierung?

- Grundlage ist das Konzept der Normalformen: Eine Relation befindet sich in einer Normalform, wenn es jeweils gewissen Bedingungen genügt.
- Streng genommen bedeutet „normalisiert“, dass sich eine Relation in der ersten Normalform (1NF) befindet. Allerdings wird mit diesem Begriff oft auch (ungenau) eines der höheren Levels bezeichnet (insbesondere die dritte Normalform bzw. 3NF).
- Der Prozess der Normalisierung kann beschrieben werden als die schrittweise Reduktion (Zerlegung) einer Menge komplexer Relationen in einfachere Beziehungen durch Aufteilung der Attribute einer Tabelle auf mehreren Tabellen.
- Normalisierung ist stets umkehrbar, d.h. durch Normalisierung gehen keine Informationen verloren.

- Die Eliminierung von Redundanzen
- Das Verhindern von Anomalien, d.h. Modifikationen der Daten, die zur Inkonsistenz führen
- Integritätssicherung
- Generierung von „lesbaren Relationen“, d.h. stabile und flexible Datenstrukturen, die bei Erweiterungen möglichst wenig geändert werden müssen
- Verringerung der Notwendigkeit zur Umstrukturierung, dadurch müssen die Anwendungsprogramme nicht so oft geändert werden
- verständliches Datenmodell für Benutzer und Programmierer
- Datenbankdesigner könnten auf Relationen in der „höchstmöglichen“ Normalform abzielen. Aber oft genügt die dritte Normalform (oder weniger).

Gesamtheit aller Relationen



4.2.6 Nullte Normalform (0 NF)

<u>PerNr</u>	Name	Wohnort	AbtNr	AbtName	ProjNr	ProjName	Zeit
101	Hans	Zürich	1	Physik	11, 12	A, B	60, 40
102	Rolf	Basel	2	Chemie	13	C	100
103	Urs	Genf	2	Chemie	11, 12, 13	A, B, C	20, 50, 30
104	Paul	Zürich	1	Physik	11, 13	A, C	80, 20

- Die Attributwerte sind nicht atomar, sondern zusammengesetzt oder ein Feld enthält mehrere verschiedene Werte eines Attribut-Typs.
- Die Relation lässt Anomalien zu.
- Abhilfe: Attribute mit nicht-atomaren Attributwerten werden in mehrere Attribute aufgeteilt, d.h. alle Wiederholungsgruppen werden entfernt und in eigenen Zeilen untergebracht.

Erste Normalform (1NF)

Eine Relation ist in der **ersten Normalform (1NF)**, wenn alle ihre Attribute nur atomare Attributwerte besitzen.

(Streng betrachtet wird dies eigentlich bereits durch das relationale Modell erzwungen.)

4.2.7 Erste Normalform (1NF)

<u>PerNr</u>	Name	Wohnort	AbtNr	AbtName	<u>ProjNr</u>	ProjName	Zeit
101	Hans	Zürich	1	Physik	11	A	60
101	Hans	Zürich	1	Physik	12	B	40
102	Rolf	Basel	2	Chemie	13	C	100
103	Urs	Genf	2	Chemie	11	A	20
103	Urs	Genf	2	Chemie	12	B	50
103	Urs	Genf	2	Chemie	13	C	30
104	Paul	Zürich	1	Physik	11	A	80
104	Paul	Zürich	1	Physik	13	C	20

- **Funktionale Abhängigkeit**
- **Voll funktionale Abhängigkeit**
- **Transitive Abhängigkeit**

Funktionale Abhängigkeit

(functional dependency, abgekürzt als „FD“)

- In einer Relation $R(A, B)$ ist das Attribut B von dem Attribut A funktional abhängig, falls zu jedem **möglichen** Wert des Attributs A genau ein Wert des Attributs B gehört, d.h. gleiche Attributwerte für A erzwingen gleiche Attributwerte für B .
- Bezeichnung: $A \rightarrow B$

4.2.8 Klassifizierung der Attribute nach Abhängigkeiten

Beispiel: Projekte haben einen Ansprechpartner

PersNr	Name	Wohnort	<u>ProjektNr</u>	ProjektName	Priorität
4	John	Boston	2	TopSecret	1
3	Bob	Los Angeles	5	Innovation	3
3	Bob	Los Angeles	8	Cobra	5
3	Bob	Los Angeles	4	Expansion	8
2	Alice	Portland	1	Optimization	2
2	Alice	Portland	3	Firewall	4

- Funktionale Abhängigkeit: $\text{PersNr} \rightarrow \text{Name}$...gilt auch: $\text{Name} \rightarrow \text{PersNr}$?
- Weitere Beispiele für FDs: $\text{PersNr} \rightarrow \text{Name}, \text{Wohnort}$
 $\text{ProjektNr} \rightarrow \text{ProjektName}, \text{Priorität}$
 $\text{ProjektNr} \rightarrow \text{PersNr}, \text{Name}, \text{Wohnort}$

Funktionale Abhängigkeiten

- beziehen sich auf die Semantik von Attributen für alle **möglichen** Instanzen (Datenbelegungen) einer Relation
- sie können also nur vom Datenbankdesigner aus dem Wissen über den Anwendungsbereich abgeleitet werden!
- Beispiel:
Eine weitere Person namens „John“ ist Ansprechpartner für ein neues Projekt, das später eingefügt wird.

Schlüssel als Spezialfall

- Funktionale Abhängigkeit:

`ProjektNr →`

`PersNr, Name, Wohnort, ProjektNr, ProjektName, Priorität`

- X ist **Schlüssel**, wenn für ein Relationenschema R die funktionale Abhängigkeit $X \rightarrow R$ gilt und X **minimal** ist
- Ziel: alle gegebenen funktionalen Abhängigkeiten sollen in diese „Schlüsselabhängigkeiten“ umgeformt werden, ohne dabei semantische Information zu verlieren.

Vollständige funktionale Abhängigkeit

- Vollständig funktional abhängig bedeutet, dass das **Nicht-Schlüsselattribut** nicht nur von einem Teil der Attribute eines zusammengesetzten Schlüsselkandidaten funktional abhängig ist, sondern **von allen Teilen**.
- In einer Relation $R (S1 / S2, A)$ ist das Attribut A von den Attributen (Schlüsseln) $S1, S2$ voll funktional abhängig, wenn A von den zusammengesetzten Attributen $(S1, S2)$ funktional abhängig ist, nicht aber von einem einzelnen Attribut $S1$ oder $S2$.

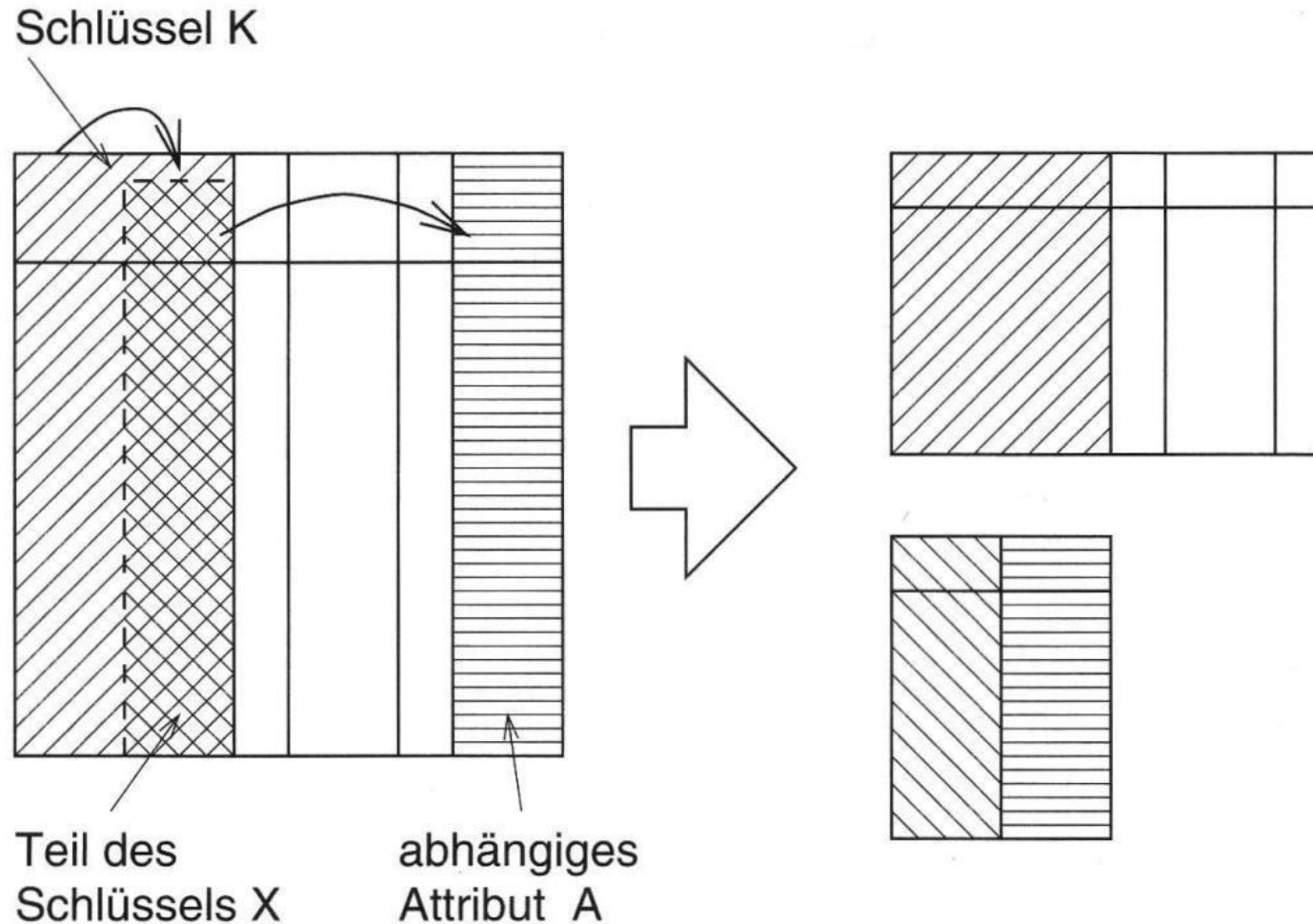
Transitive Abhängigkeit

- Transitive Abhängigkeit ist die funktionale Abhängigkeit eines **Nicht-Schlüsselattributes** einer Relation R **von einem anderen Nicht-Schlüsselattribut** in R.
- In einer Relation R (S, A, B) ist das Attribut B vom Attribut (Schlüssel) S (der auch ein zusammengesetzter Schlüssel sein kann) transitiv abhängig, wenn A von S funktional abhängig ist, S jedoch nicht von A, und B von A funktional abhängig ist.

Zweite Normalform (2NF)

- Eine Relation befindet sich in der Zweiten Normalform, wenn sie sich in der Ersten Normalform befindet und jedes Nicht-Schlüsselattribut vom gesamten Primärschlüssel (und nicht nur von Teilen des Schlüssels) **voll funktional abhängig** ist.
- Ein Attribut ist **voll funktional abhängig** von einem zusammengesetzten Schlüssel, falls es nicht funktional abhängig ist von einem Teil des Schlüssel (**partielle Abhängigkeit**).
- Hinweis: die 2NF ist nur dann zu verletzen, wenn eine Relation einen zusammengesetzten Schlüssel hat und mindestens ein Attribut besitzt, das nicht zum Gesamt-Schlüssel gehört.

Partielle Abhängigkeit und ihre Elimination



Ausgangssituation: Relation in der ersten Normalform

<u>PerNr</u>	Name	Wohnort	AbtNr	AbtName	<u>ProjNr</u>	ProjName	Zeit
101	Hans	Zürich	1	Physik	11	A	60
101	Hans	Zürich	1	Physik	12	B	40
102	Rolf	Basel	2	Chemie	13	C	100
103	Urs	Genf	2	Chemie	11	A	20
103	Urs	Genf	2	Chemie	12	B	50
103	Urs	Genf	2	Chemie	13	C	30
104	Paul	Zürich	1	Physik	11	A	80
104	Paul	Zürich	1	Physik	13	C	20

Warum ist diese Relation nicht in zweiter Normalform?

Zerlegung der Relation führt zur zweiten Normalform.

<u>PerNr</u>	<u>ProjNr</u>	Zeit
101	11	60
101	12	40
102	13	100
103	11	20
103	12	50
103	13	30
104	11	80
104	13	20

<u>PerNr</u>	Name	Wohnort	AbtNr	AbtName
101	Hans	Zürich	1	Physik
102	Rolf	Basel	2	Chemie
103	Urs	Genf	2	Chemie
104	Paul	Zürich	1	Physik

<u>ProjNr</u>	ProjName
11	A
12	B
13	C

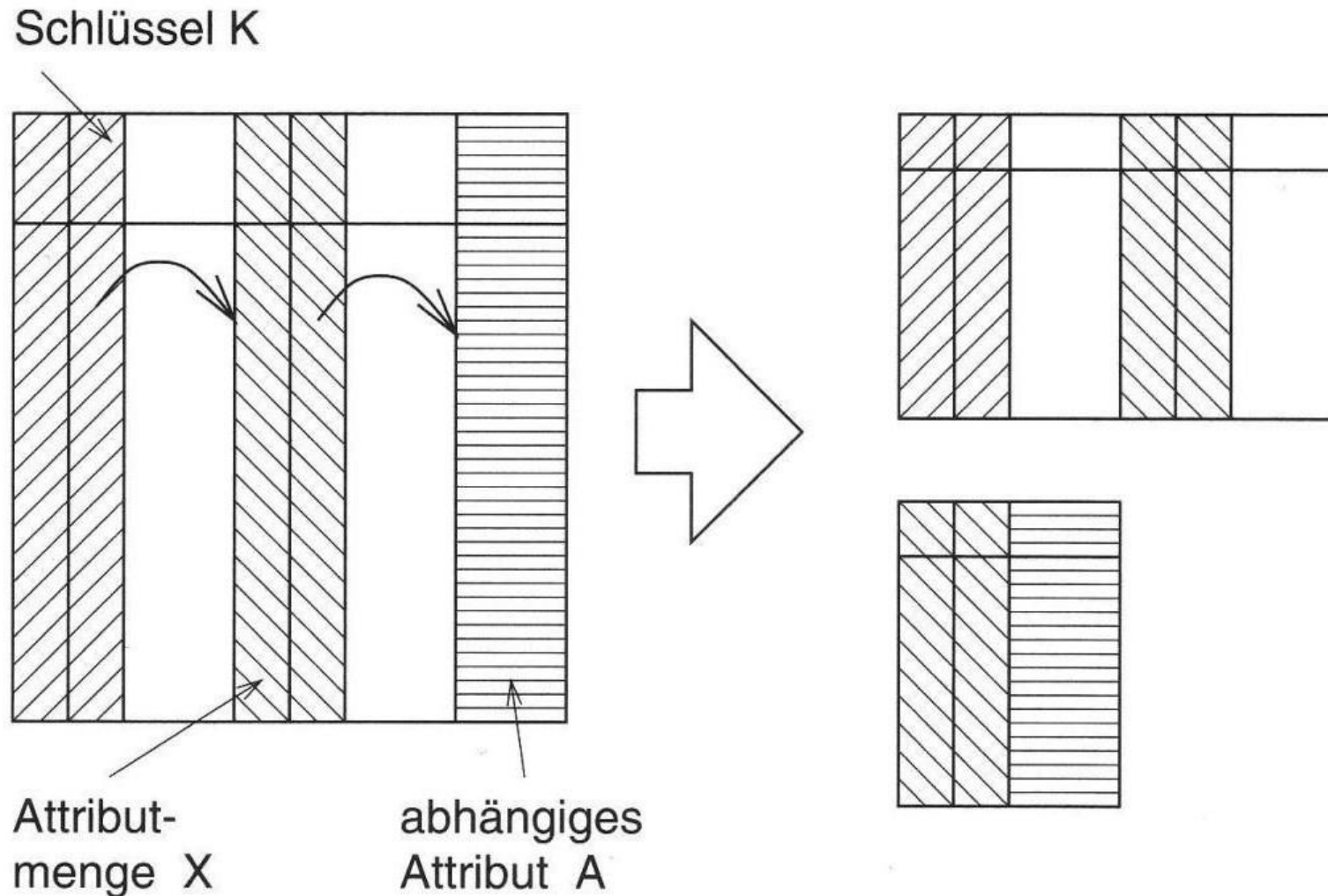
Entstanden durch Zerlegung aufgrund
der Abhängigkeit $\text{PerNr} \rightarrow \text{Name}$

Entstanden durch Zerlegung aufgrund
der Abhängigkeit $\text{ProjNr} \rightarrow \text{ProjName}$

Dritte Normalform (3NF)

- Eine Tabelle ist in der **dritten Normalform**, falls sie in 2NF ist und kein Nichtschlüsselattribut von irgendeinem Schlüssel **transitiv abhängig** ist.
- Eine Tabelle ist in der dritten Normalform, wenn sie in der zweiten Normalform ist und keine funktionalen Abhängigkeiten zwischen Attributen erlaubt, die nicht als Schlüsselkandidaten in Frage kommen.

Transitive Abhängigkeit und ihre Elimination



Ausgangssituation: Relation in der zweiten Normalform

<u>PerNr</u>	<u>ProjNr</u>	Zeit
101	11	60
101	12	40
102	13	100
103	11	20
103	12	50
103	13	30
104	11	80
104	13	20

<u>PerNr</u>	Name	Wohnort	AbtNr	AbtName
101	Hans	Zürich	1	Physik
102	Rolf	Basel	2	Chemie
103	Urs	Genf	2	Chemie
104	Paul	Zürich	1	Physik

<u>ProjNr</u>	ProjName
11	A
12	B
13	C

Warum ist diese Relation nicht in dritter Normalform?


Zerlegung der Relation führt zur dritten Normalform.

<u>PerNr</u>	<u>ProjNr</u>	Zeit
101	11	60
101	12	40
102	13	100
103	11	20
103	12	50
103	13	30
104	11	80
104	13	20

<u>PerNr</u>	Name	Wohnort	<u>AbtNr</u>
101	Hans	Zürich	1
102	Rolf	Basel	2
103	Urs	Genf	2
104	Paul	Zürich	1

<u>ProjNr</u>	ProjName
11	A
12	B
13	C

Entstanden durch Zerlegung aufgrund der transitiven Abhängigkeit von $\text{PerNr} \rightarrow \text{AbtNr}$ und $\text{AbtNr} \rightarrow \text{AbtName}$



<u>AbtNr</u>	AbtName
1	Physik
2	Chemie

Erste Normalform

Zweite Normalform

Dritte Normalform

weitere Normalformen...

Keine transitiven Abhängigkeiten

**Nichtschlüsselmerkmale sind voll vom
Schlüssel abhängig**

Keine Wiederholungsgruppen zugelassen

- Durch die **Zerlegung von Tabellen** wird für manche Queries dadurch ein zusätzlicher **Join** notwendig.
- Dies beeinträchtigt die **Performance!!!**
- Folglich müssen bei sehr großen Datenbeständen die Gründe für eine Zerlegung genau betrachtet werden, um die potenziellen **Performanceverluste** aufzuwiegen.

Folge:

- Unvermeidbarer **Zielkonflikt** zwischen der Forderung nach guter **Performance** und **Redundanzfreiheit**

- Nachteile der Normalisierung
 - Normalisierungsaufwand
 - Schlechtere Laufzeiten bei der Zusammenfassung (Join) mehrerer Relationen
- Beim Datenbankenentwurf sind somit die Kosten und Nutzen der Normalisierung abzuwägen
- Normalisierung bis zur 3NF ist in der Regel ausreichend