



Kapitel 2 – Relationenmodell und Relationenalgebra

Vorlesung Datenbanken

Dr. Kai Höfig



Diskussion/Wiederholung

- ◆ Was ist eine Datenbank und welche Motivation sie einzusetzen gibt es?
- ◆ Was sind die Vorteile der Datenspeicherung in Relationen?
- ◆ Was versteht man unter der Schema-Architektur?
- ◆ Welche Architekturmuster im Zusammenhang mit Datenbanken gibt es und wozu sind sie gut?
- ◆ Was versteht man unter einem Datenmodell?



- ◆ 2.1 Relationales Datenmodell
- ◆ 2.2 Integritätsbedingungen
- ◆ 2.3 Relationale Algebra



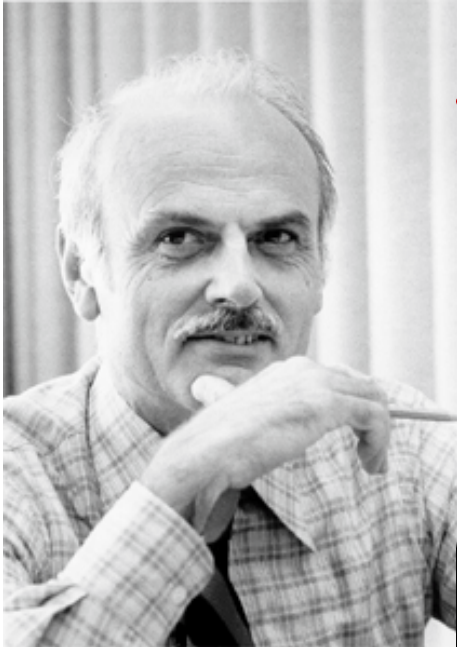
Das Relationale Datenmodell

- ◆ **Struktur** der Daten:
Daten werden in Relationen (Tabellen) abgelegt
- ◆ **Operationen** auf den Daten: 2 Alternativen
 1. Relationale Algebra: in Sprache SQL (Structured Query Language) praktisch umgesetzt
 2. Relationaler Kalkül: in Sprache QBE (Query by Example) / (QBC Query by Criteria) praktisch umgesetzt
- ◆ **Integritätsbedingungen**: Sehr viele! Am wichtigsten:
 1. **Schlüsselbedingungen** (key constraints)
 2. **Referentielle Integrität** = **Fremdschlüsselbedingungen**
 3. **Domain-Constraints** (Einschränkungen der erlaubten Werte für ein Attribut)
 - Integritätsbedingungen können als Bedingungen in der Relationalen Algebra, im Relationalen Kalkül oder in SQL formuliert werden

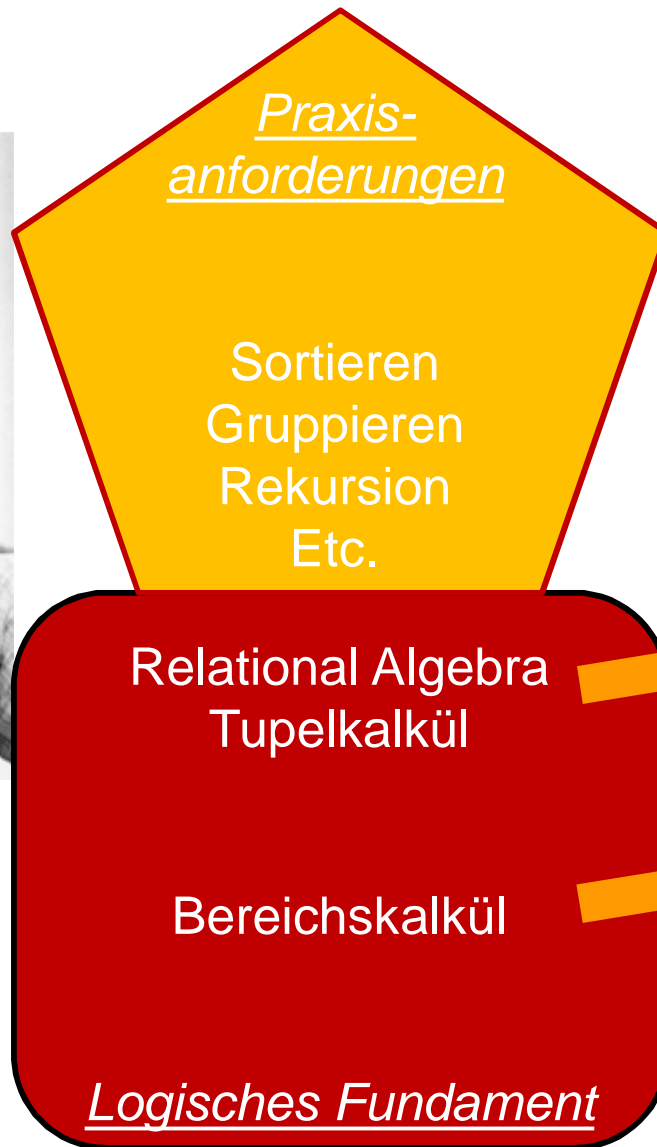


Relationale Algebra, Kalküle und Sprachen

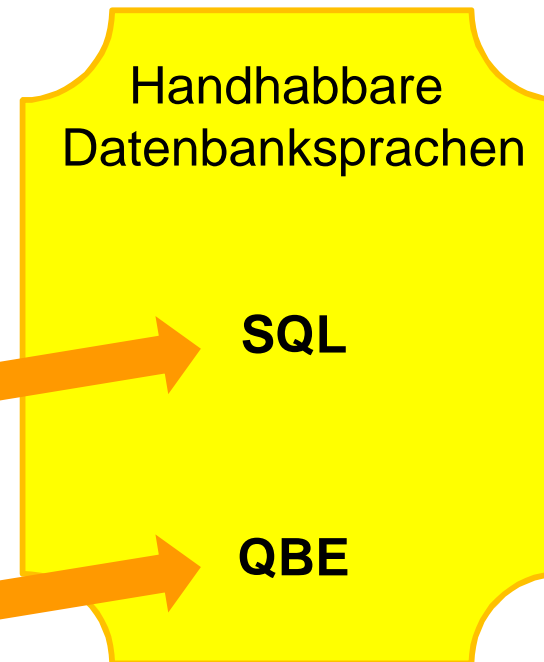
Theorie



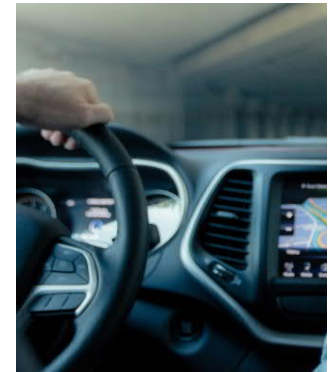
Codd schuf in den 1960er und 1970er Jahren das relationale Modell, das die Grundlage der relationalen Datenbanken ist, die bis heute einen Standard der Datenbanktechnik darstellen.



Ohne das Verständnis des logischen Fundaments interagieren wir mittels der Datenbanksprachen mit einem unbekannten Kommunikationspartner



Praxis





Relationale Algebra, Kalküle und Sprachen - Skript

- ◆ Ohne logisches Fundament ist es nicht möglich, gute Sprachen für die Praxis zu entwerfen.
- ◆ Warum brauchen wir die relationale Algebra und zwei Kalküle?
 - Tupelkalkül ist die Basis von SQL → wichtig für das Verständnis von SQL
 - Bereichskalkül ist die Basis von QBE (und auch QBC) → wichtig für das Verständnis von QBE
 - Kalküle sind deklarativ, d.h. Auswertungsreihenfolge ist nicht ersichtlich, d.h.
 - Einfacher zu benutzen als prozedurale Anweisungen – Nutzer muss nicht wissen, wie das DBMS das Ergebnis berechnet.
 - DBMS hat Freiheiten in der Abarbeitung, kann also eine möglichst effiziente wählen.
 - Algebra ist prozedural, d.h. gibt Abarbeitungsreihenfolge vor („von innen nach außen“).
 - Es gibt Gesetze zur Umformung von Algebra Ausdrücken.
 - DBMS, übersetzt die SQL Anfrage (=Tupelkalkül) in eine Algebra-Ausdruck, optimiert diesen (Query Optimizer) und führt ihn aus.
 - Diese „Ausführungsplan“ verwendet der DB Admin, um die DB zu optimieren (z.B. mittels Indexstrukturen).

WEINE

WeinID	Name	Farbe	Jahrgang	Weingut
1042	La Rose GrandCru	Rot	1998	Chateau La Rose
2168	Creek Shiraz	Rot	2003	Creek
3456	Zinfandel	Rot	2004	Helena
2171	Pinot Noir	Rot	2001	Creek
3478	Pinot Noir	Rot	1999	Helena
4711	Riesling Reserve	Weiß	1999	Müller
4961	Chardonnay	Weiß	2002	Bighorn

ERZEUGER

Weingut	Anbaugebiet	Region
Creek	Barossa Valley	South Australia
Helena	Napa Valley	Kalifornien
Chateau La Rose	Saint-Emilion	Bordeaux
Chateau La Pointe	Pomerol	Bordeaux
Müller	Rheingau	Hessen
Bighorn	Napa Valley	Kalifornien

EMPFEHLUNG

Wein
La Rose Grand Cru
Riesling Reserve
Merlot Selection
Sauvignon Blanc

WEINLISTE

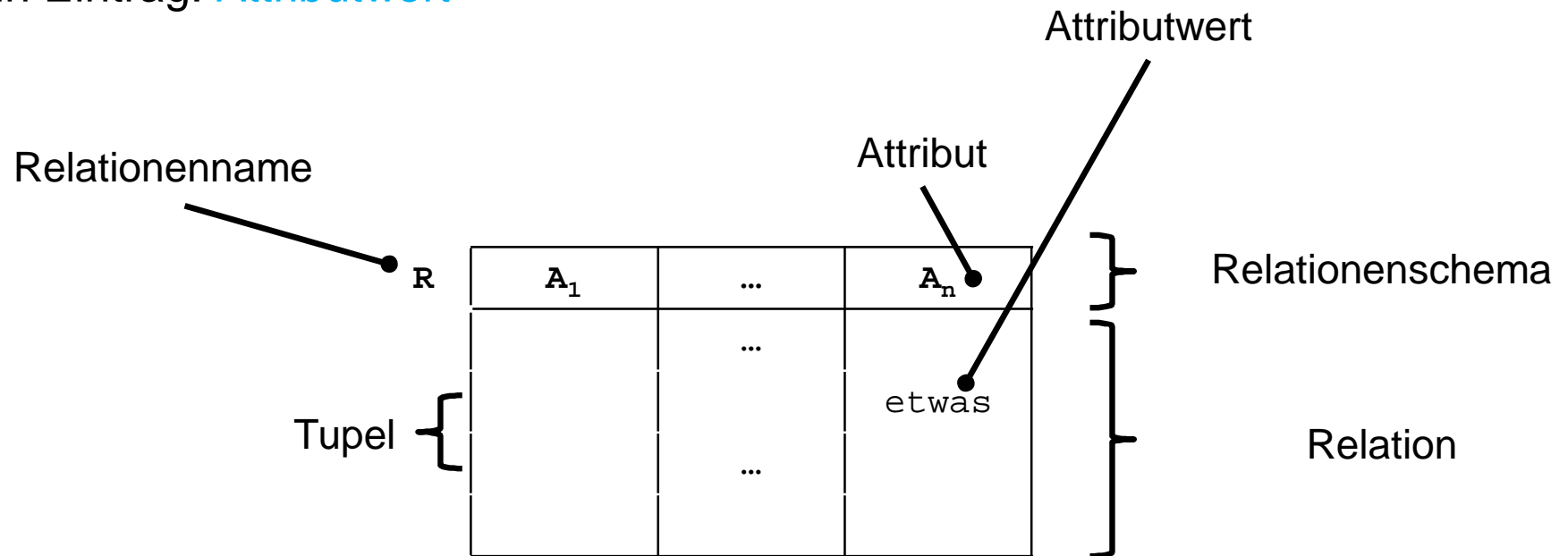
Name
La Rose Grand Cru
Creek Shiraz
Zinfandel
Pinot Noir
Riesling Reserve



Darstellung von Relationen und Begriffe

◆ Darstellung

- Erste Zeile: **Relationenschema**
- Weitere Einträge in der Tabelle: **Relation**
- Eine Zeile der Tabelle: **Tupel**
- Eine Spaltenüberschrift: **Attribut**
- Ein Eintrag: **Attributwert**





Definition des Relationalen Datenmodells (1)

- ♦ Im relationalen Datenmodell steht uns nur das Relationenschema zur strukturellen Modellierung zur Verfügung.
- ♦ Es sei $R = \{A_1, \dots, A_k\}$ ein **Relationenschema** mit dem Bezeichner R über den Attributen A_1, \dots, A_k auf die Wertebereiche D_1, \dots, D_s mit $dom: \{A_1, \dots, A_k\} \rightarrow \{D_1, \dots, D_s\}$, $s \geq 1$, der Wertebereichsfunktion.
 - ♦ Beispiel:
ERZEUGER = { Weingut, Anbaugebiet, Region } mit
 $dom(\text{Weingut}) = \text{string}$, $dom(\text{Anbaugebiet}) = \text{string}$, $dom(\text{Region}) = \text{string}$
- ♦ Ein **relationales Datenbankschema** ist eine endliche nicht leere Menge $S = \{ R_1(\alpha_1), \dots, R_m(\alpha_m) \}$ von Relationenschemata über Teilmengen einer gemeinsamen Attributmenge $\alpha = \alpha_1, \dots, \alpha_m$
- ♦ Dabei seien alle Bezeichner von Relationen paarweise verschieden und verschieden von allen Attributbezeichnern.



Definition des Relationalen Datenmodells (2)

- ♦ Eine **Relation** r über einem Relationenschema $R=\{A_1,\dots,A_k\}$ kurz $r(R)$, ist eine endliche Menge von Tupeln, die jedes Attribut A_j auf einen Wert aus $dom(A_j)$ abbilden.

- ♦ Beispiel: eine Mögliche Relation r über dem Relationenschema ERZEUGER ist mit

$t_2(\text{Weingut}) = \text{'Helena'}$	$t_2(\text{Anbaugebiet}) = \text{'Napa Valley'}$	$t_2(\text{Region}) = \text{'Kalifornien'}$
$t_3(\text{Weingut}) = \text{'Müller'}$	$t_3(\text{Anbaugebiet}) = \text{'Rheingau'}$	$t_3(\text{Region}) = \text{'Hessen'}$

- ♦ Eine **Datenbank** über einem Datenbankschema $S = R_1(\alpha_1), \dots, R_m(\alpha_m)$ ist eine Menge von Relationen $d := \{r_1, \dots, r_p\}$ wobei jede Relation r_i über dem Relationenschema R_i definiert ist: $r_i(R_i)$
- ♦ Eine Relation $r \in d$ bezeichnen wir als **Basisrelation**
 - ♦ Beispiel: Unser Datenbankschema WeinDB enthält 4 Basisrelationen, die wir formal mit $r1(\text{WEINE})$, $r2(\text{ERZEUGER})$, $r3(\text{EMPFEHLUNG})$, $r4(\text{WEINLISTE})$ bezeichnen



Wichtige Integritätsbedingungen im RM: Schlüsselbedingung

◆ Schlüsseleigenschaft:

Eine oder mehrerer Attribute **identifizieren** eindeutig gespeicherte Tupel

- Beispiel: Attribut **Weingut** für Tabelle **ERZEUGER**

ERZEUGER	<u>Weingut</u>	Anbaugebiet	Region
	Creek	Barossa Valley	South Australia
	Helena	Napa Valley	Kalifornien
	Chateau La Rose	Saint-Emilion	Bordeaux
	Chateau La Pointe	Pomerol	Bordeaux
	Müller	Rheingau	Hessen
	Bighorn	Napa Valley	Kalifornien

◆ Beachte

- Auch Attributkombinationen können Schlüssel sein
- Eine Tabelle kann mehrere Schlüssel haben
- Ein Schlüssel wird beim DB Entwurf als **Primärschlüssel** ausgewählt
- Schlüssel werden i.allg. durch Unterstreichen gekennzeichnet

Schlüsselbedingung: Es gibt keine 2 Tupel in ERZEUGER mit demselben Wert von Weingut



Wichtige Integritätsbedingungen im RM: Referentielle Integrität

WEINE

<u>WeinID</u>	Name	Farbe	Jahrgang	Weingut → ERZEUGER
1042	La Rose GrandCru	Rot	1998	Chateau La Rose
2168	Creek Shiraz	Rot	2003	Creek
3456	Zinfandel	Rot	2004	Helena
2171	Pinot Noir	Rot	2001	Creek
3478	Pinot Noir	Rot	1999	Helena
4711	Riesling Reserve	Weiß	1999	Müller
4961	Chardonnay	Weiß	2002	Bighorn

Weingut in WEINE
verwendet →
Fremdschlüssel

ERZEUGER

<u>Weingut</u>	Anbaugebiet	Region
Creek	Barossa Valley	South Australia
Helena	Napa Valley	Kalifornien
Chateau La Rose	Saint-Emilion	Bordeaux
Chateau La Pointe	Pomerol	Bordeaux
Müller	Rheingau	Hessen
Bighorn	Napa Valley	Kalifornien

Weingut ist
Schlüssel in
ERZEUGER

Referentielle Integrität:
Jeder Wert von
Weingut in WEINE
ist in ERZEUGER
vorhanden



Aber wie komme ich zu Sichten?

WEINE	WeinID	Name	Farbe	Jahrgang	Weingut → ERZEUGER
	1042	La Rose GrandCru	Rot	1998	Chateau La Rose
	2168	Creek Shiraz	Rot	2003	Creek

„Weine aus Kalifornien kann ich generell empfehlen, egal aus welcher Region“

EMPFEHLUNG	Wein	Farbe	Jahrgang	Region
	Zinfandel	Rot	2004	Napa Valley
	Chardonnay	Weiß	2002	Napa Valley
	Pinot Noir	rot	1999	Napa Valley

Chateau La Rose	Saint-Emilion	Bordeaux
Chateau La Pointe	Pomerol	Bordeaux
Müller	Rheingau	Hessen
Bighorn	Napa Valley	Kalifornien



Anfrageoperationen auf Tabellen

- ◆ **Relationale Algebra (Relationenalgebra):** Menge von **Basisoperationen** auf Relationen zur Berechnung von neuen (Ergebnis-)Relationen
 - beliebig kombinierbar
 - bilden dadurch eine Algebra zum „Rechnen mit Tabellen“
- ◆ Wiederholung aus der Mathematik:
Algebra = Wertebereich + darauf definierten Operationen
- ◆ Hier
 - Wertebereich = Inhalte der Datenbank = Tabellen
 - Operationen = Funktionen zum Berechnen von neuen Tabellen



Relationenalgebra: Übersicht

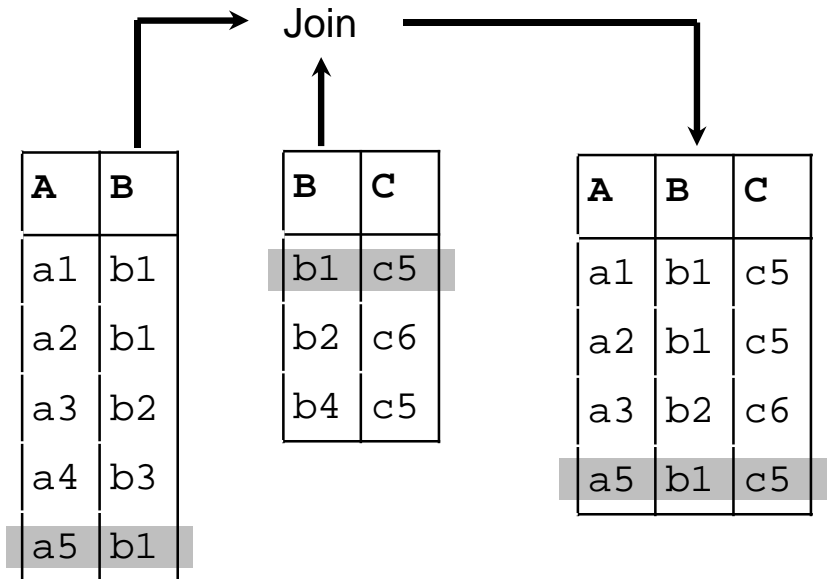
- ◆ Drei Hauptoperationen: Selektion, Projektion, Join

A	B	C
a1	b1	c1
a2	b1	c2
a3	b2	c1
a4	b3	c3
a5	b1	c1

Projektion

A	B	C
a1	b1	c1
a2	b1	c2
a3	b2	c1
a4	b3	c3
a5	b1	c1

Selektion



Verbund / Join



Projektion π Definition

- ◆ Projektion π (Pi): Auswahl von Spalten durch Angabe einer Attributliste

- ◆ Die Projektion entfernt doppelte Tupel (Mengen-semantik)

- ◆ Syntax:

$\pi_{\langle \text{Attributmenge} \rangle}(\text{Relation})$ oder $\langle \text{Relation} \rangle[\langle \text{Attributmenge} \rangle]$

- ◆ Beispiel:

$\pi_A(R)$ oder $R[A]$

R:	A	B	R[A]	A
	1	2		1
	1	3		3
	3	4		



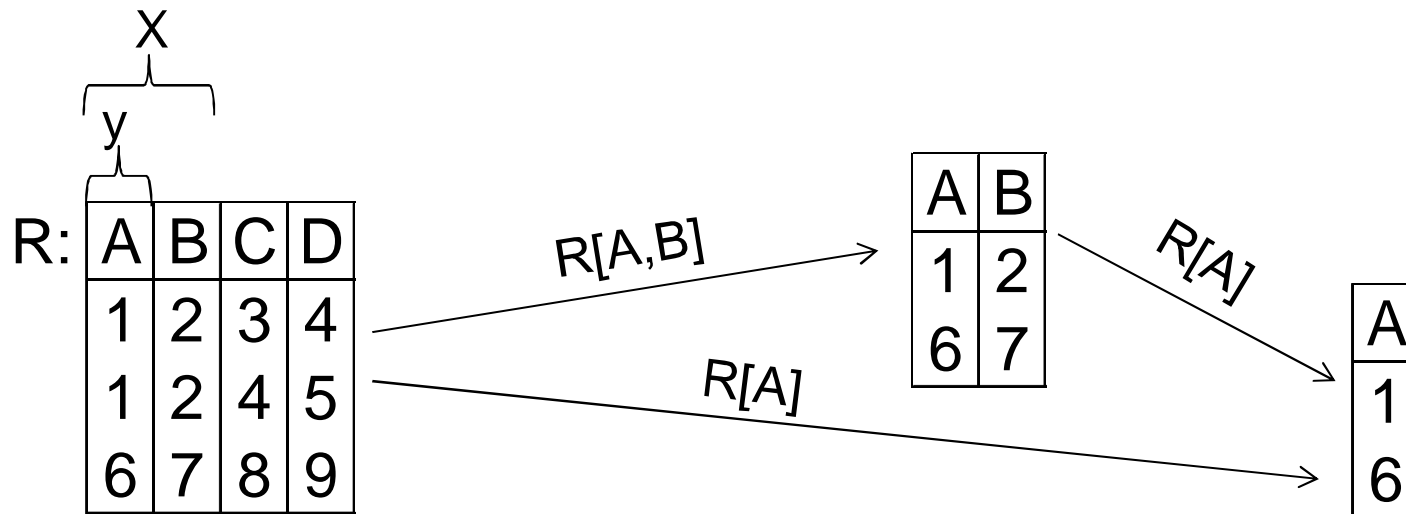
Projektion

- ◆ Semantik

$$\pi_X(r) := \{ t(X) \mid t \in r \}$$

für $r(R)$ und $X \subseteq R$ Attributmenge in R

- ◆ Eigenschaft für $Y \subseteq X \subseteq R$: $\pi_Y(\pi_X(r)) = \pi_Y(r)$



- ◆ Achtung: π entfernt Duplikate (Mengensemantik)



Projektion: weitere Beispiele

◆ Relation

ERZEUGER	Weingut	Anbaugebiet	Region
	Creek	Barossa Valley	South Australia
	Helena	Napa Valley	Kalifornien
	Chateau La Rose	Saint-Emilion	Bordeaux
	Chateau La Pointe	Pomerol	Bordeaux
	Müller	Rheingau	Hessen
	Bighorn	Napa Valley	Kalifornien

◆ $\pi_{\text{Region}}(\text{ERZEUGER})$

Region
South Australia
Kalifornien
Bordeaux
Hessen

◆ $\pi_{\text{Anbaugebiet}, \text{Region}}(\text{ERZEUGER})$

Anbaugebiet	Region
Barossa Valley	South Australia
Napa Valley	Kalifornien
Saint-Emilion	Bordeaux
Pomerol	Bordeaux
Rheingau	Hessen



Selektion σ Definition

- ◆ Selektion σ (Sigma): Auswahl von Zeilen einer Tabelle anhand eines Selektionsprädikats
- ◆ Syntax: $\sigma_{\langle \text{Bedingung} \rangle}(\langle \text{Relation} \rangle)$ oder $\langle \text{Relation} \rangle[\langle \text{Bedingung} \rangle]$
- ◆ Semantik (für $A \in R$)

$$\sigma_{A=a}(r) := \{ t \in r \mid t(A) = a \}$$

- ◆ Beispiel:

$\sigma_{A=1}(R)$ oder $R[A=1]$

R:	<table><tr><th>A</th><th>B</th></tr><tr><td>1</td><td>2</td></tr><tr><td>3</td><td>4</td></tr></table>	A	B	1	2	3	4	R[A=1]:	<table><tr><th>A</th><th>B</th></tr><tr><td>1</td><td>2</td></tr></table>	A	B	1	2
A	B												
1	2												
3	4												
A	B												
1	2												



♦ Konstantenselektion

Attribut θ Konstante

boolesches Prädikat θ ist = oder \neq , bei linear geordneten Wertebereichen auch \leq , $<$, \geq oder $>$

♦ Attributselektion

Attribut₁ θ Attribut₂

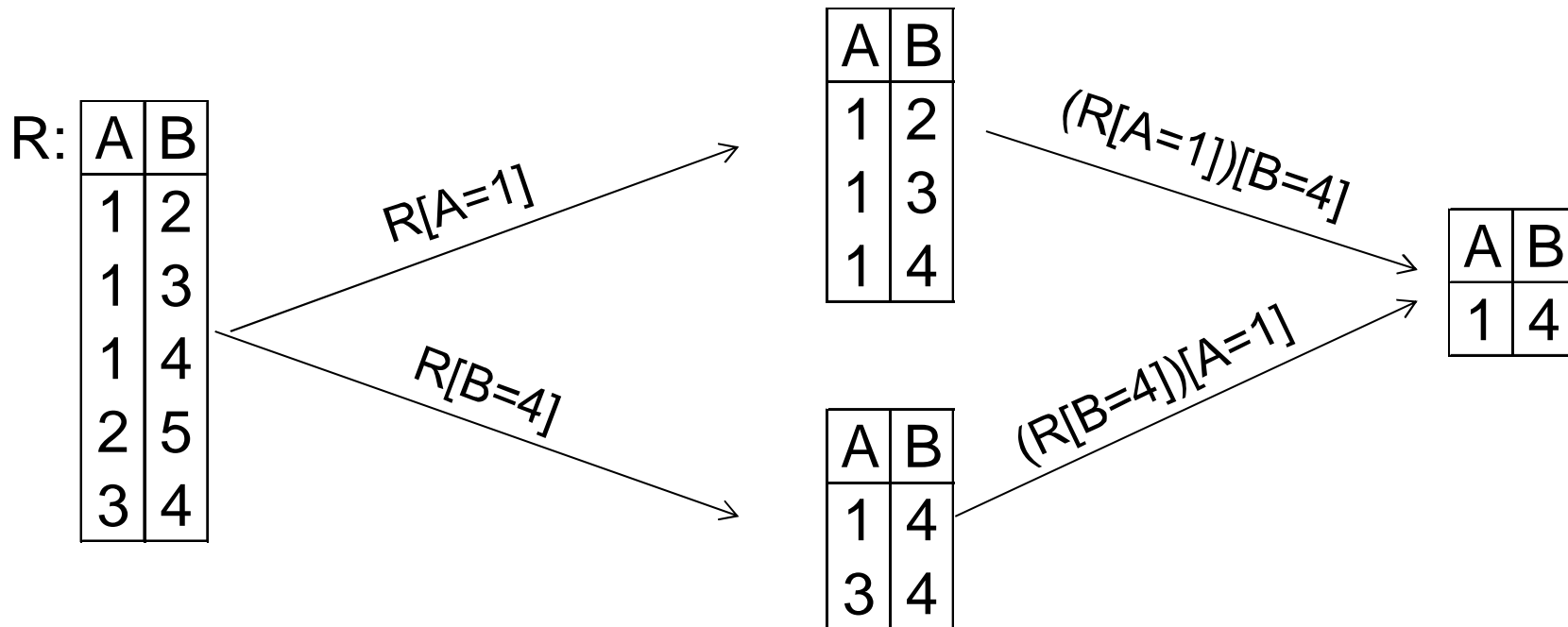
♦ Logische Verknüpfung mehrerer Konstanten- oder Attribut-Selektionen mit \wedge , \vee oder \neg .



Selektion: Gesetze zur Umformung - Kommutativität

♦ Kommutativität

$$\sigma_{A=a}(\sigma_{B=b}(r)) = \sigma_{B=b}(\sigma_{A=a}(r))$$

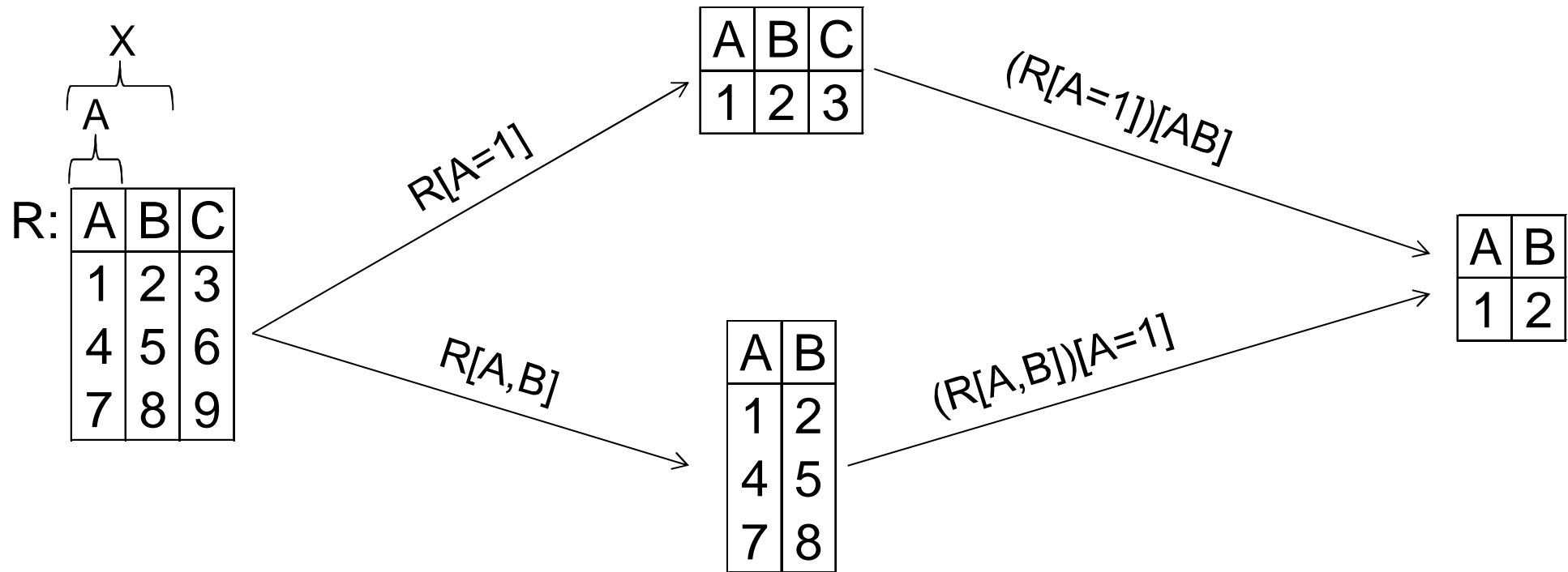




Selektion: Gesetze zur Umformung - Vertauschung

- ♦ Falls $A \in X, X \subseteq R$

$$\pi_X(\sigma_{A=a}(r)) = \sigma_{A=a}(\pi_X(r))$$





Kreuzprodukt \times

- ◆ **Kreuzprodukt \times (Kartesisches Produkt, cross join):** verknüpft zwei Tabellen, indem jedes Tupel der ersten mit jedem Tupel der zweiten kombiniert wird.
 - Vorsichtig: Ergebnis für Tabellen mit n bzw. m Tupel hat $n*m$ Tupel!
- ◆ **Syntax:** $\langle \text{Relation1} \rangle \times \langle \text{Relation2} \rangle$
- ◆ **Semantik:** $R \times S := \{ x_1 \dots x_n \dots x_{n+m} \mid R(x_1, \dots, x_n) \wedge S(x_{n+1}, \dots, x_{n+m}) \}$
- ◆ **Beispiel:**

R:

A	B
1	2
3	4

S:

C	D
5	6
7	8

$R \times S$:

A	B	C	D
1	2	5	6
1	2	7	8
3	4	5	6
3	4	7	8

* Gleichnamige Attribute werden umbenannt



Kreuzprodukt \times Beispiel

◆ Beispiel:

WEINE \times FLASCHE

FLASCHE	<u>Art</u>	Inhalt
	Normal	700
	Klein	375

WeinID	Name	Farbe	Jahrgang	Weingut	Art	Inhalt
1042	La Rose GrandCru	rot	1998	Chateau La Rose	Normal	700
1042	La Rose GrandCru	rot	1998	Chateau La Rose	Klein	375
2168	Creek Shiraz	rot	2003	Creek	Normal	700
2168	Creek Shiraz	rot	2003	Creek	Klein	375
3456	Zinfandel	rot	2004	Helena	Normal	700
3456	Zinfandel	rot	2004	Helena	Klein	375
...



Natürlicher Verbund \bowtie

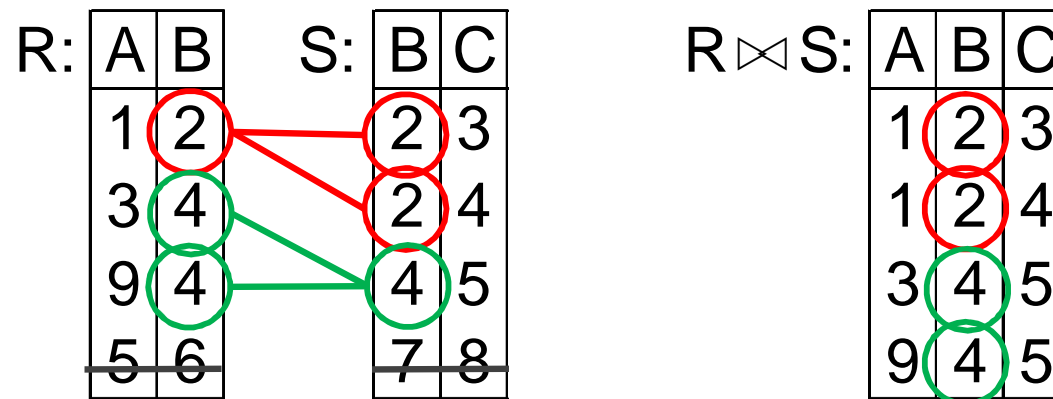
- ♦ **Natürlicher Verbund \bowtie (engl. natural join):** verknüpft Tabellen über **gleichbenannte Spalten**, indem er jeweils zwei Tupel verschmilzt, falls sie dort **gleiche Werte** haben
 - Tupel, die keinen Partner finden (dangling tuples), werden eliminiert.

♦ Syntax: $\langle \text{Relation1} \rangle \bowtie \langle \text{Relation2} \rangle$

- ♦ Semantik für A Attribute von R , C Attribute von S und B Attribute im Schnitt

$$R \bowtie S := \pi_{A1, \dots, Am, R.B1, \dots, R.Bk, S1, \dots, Sn} (\sigma_{R.B1=S.B1 \wedge \dots \wedge R.Bk=S.Bk} (R \times S))$$

- ♦ Beispiel:





◆ Beispiel:

WEINE ☒ ERZEUGER

<u>WeinID</u>	Name	...	Weingut	Anbaugebiet	Region
1042	La Rose GrandCru	...	Chateau La Rose	Saint-Emilion	Bordeaux
2168	Creek Shiraz	...	Creek	Barossa Valley	South Australia
3456	Zinfandel	...	Helena	Napa Valley	Kalifornien
2171	Pinot Noir	...	Creek	Barossa Valley	South Australia
3478	Pinot Noir	...	Helena	Napa Valley	Kalifornien
4711	Riesling Reserve	...	Müller	Rheingau	Hessen
4961	Chardonnay	...	Bighorn	Napa Valley	Kalifornien

- Das Weingut „Château La Pointe“ ist im Ergebnis verschwunden



Verbund: Gesetze zur Umformung

$$R \bowtie S := \pi_{A_1, \dots, A_m, R.B_1, \dots, R.B_k, S_1, \dots, S_n} (\sigma_{R.B_1=S.B_1 \wedge \dots \wedge R.B_k=S.B_k} (R \times S))$$

- ♦ Aus $R_1 \cap R_2 = \{\}$ folgt $r_1 \bowtie r_2 = r_1 \times r_2$.
Ohne Attribute im Schnitt, wird die Bedingung σ der Definition des natürlichen Verbundes aufgehoben.
- ♦ **Kommutativität:** $r_1 \bowtie r_2 = r_2 \bowtie r_1$
Die Bedingungen und Projektionen in der Definition sind kommutativ.
- ♦ **Assoziativität:** $(r_1 \bowtie r_2) \bowtie r_3 = r_1 \bowtie (r_2 \bowtie r_3)$

daher erlaubt: $\bowtie_{i=1}^p r_i$



Umbenennung β

- ◆ **Umbenennung β (Beta):** Anpassung und Umbenennung von Attributen

- ◆ **Syntax:**

$$\beta_{\langle \text{Zielattribut} \rangle \leftarrow \langle \text{Quellattribut} \rangle}(\langle \text{Relation} \rangle) \quad \text{oder} \\ \langle \text{Relation} \rangle[\text{Quellattribut} \rightarrow \text{Zielattribut}]$$

- ◆ **Semantik**

$$\beta_{B \leftarrow A}(r) := \{ t' \mid \exists t \in r : t'(R-A) = t(R-A) \wedge t'(B) = t(A) \}$$

- ◆ **Beispiel:**

R:	A	B	C	D	R[A \rightarrow X]:	X	B	C	D
	1	2	3	4		1	2	3	4
	6	7	8	9		6	7	8	9

- ◆ **durch Umbenennung nun möglich**

- Verbunde, wo bisher kartesische Produkte ausgeführt wurden (unterschiedliche Attribute werden gleich benannt),
- kartesische Produkte, wo bisher Verbunde ausgeführt wurden (gleiche Attribute werden unterschiedlich genannt),
- Mengenoperationen



Umbenennung Beispiel

- Wie kann man mit den bisher vorgestellten Operationen eine Tabelle mit den Personen, ihren Kindern und ihren Enkeln erzeugen?

R:

Person	Kind
Karl der Große	Ludwig der Fromme
Ludwig der Fromme	Lothar der I
Ludwig der Fromme	Karl II der Kahle
Lothar der I	Ludwig der II

R':

Person	Kind	Enkel
Karl der Große	Ludwig der Fromme	Lothar der I
Karl der Große	Ludwig der Fromme	Karl II der Kahle
Ludwig der Fromme	Lothar der I	Ludwig der II

- $R \bowtie (\beta_{\text{Kind} \leftarrow \text{Person}}(\beta_{\text{Enkel} \leftarrow \text{Kind}}(R)))$ oder
 $R \bowtie ((R[\text{Kind} \rightarrow \text{Enkel}])[\text{Person} \rightarrow \text{Kind}])$



Berechnung des Kreuzproduktes aus natürlichem Verbund

- ♦ Natürlicher Verbund entartet zum Kreuzprodukt, wenn keine gemeinsamen Attribute existieren
- ♦ Erzwingen durch Umbenennung: $R_1(A, B, C)$ und $R_2(C, D)$

$$R_1 \times R_2 \equiv R_1 \bowtie \beta_{E \leftarrow C}(R_2)$$

- ♦ Kreuzprodukt + Selektion simuliert natürlichen Verbund

$$R_1 \bowtie R_2 \equiv \sigma_{R_1.C=R_2.C}(R_1 \times R_2)$$



Diskussion

- ◆ Welche Strukturellen Elemente des relationalen Datenmodells kennen wir?
- ◆ Welche Operationen der relationalen Algebra gibt es?



Kombination von Operationen

- ◆ Kombinationen von Operationen sind möglich

- ◆ Beispiel

$\pi_{\text{Name, Farbe, Weingut}}(\sigma_{\text{Jahrgang} > 2000}(\text{WEINE}) \bowtie \sigma_{\text{Region} = \text{„Kalifornien“}}(\text{ERZEUGER}))$

$(\text{WEINE}[\text{Jahrgang} > 2000] \bowtie \text{ERZEUGER}[\text{Region} = \text{„Kalifornien“}])[\text{Name, Farbe, Weingut}]$

ergibt

Name	Farbe	Weingut
Zinfandel	Rot	Helena
Chardonnay	Weiß	Bighorn



Kombination von Operationen

- ♦ Klammerung des Ausdrucks ist wichtig!

- ♦ Beispiel

$\pi_{\text{Name, Farbe, Weingut}}(\sigma_{\text{Jahrgang} > 2000}(\text{WEINE})) \bowtie \sigma_{\text{Region} = \text{„Kalifornien“}}(\text{ERZEUGER})$

$(\text{WEINE}[\text{Jahrgang} > 2000])[\text{Name, Farbe, Weingut}] \bowtie \text{ERZEUGER}[\text{Region} = \text{„Kalifornien“}]$

ergibt

Name	Farbe	Weingut	Anbaugebiet	Region
Zinfandel	Rot	Helena	Napa Valley	Kalifornien
Chardonnay	Weiß	Bighorn	Napa Valley	Kalifornien



Mengenoperationen: Vereinigung

- ◆ Vereinigung $r_1 \cup r_2$ von zwei Relationen r_1 und r_2 : sammelt die Tupelmengen zweier Relationen unter einem gemeinsamen Schema auf
- ◆ Attributmengen beider Relationen müssen identisch sein
- ◆ Semantik: für $r_1(R)$ und $r_2(R)$

$$r_1 \cup r_2 := \{ t \mid t \in r_1 \vee t \in r_2 \}$$

- ◆ Beispiel:

R:

A	B	C
1	2	3
4	5	6

S:

A	B	C
4	5	6
7	8	9

$R \cup S$:

A	B	C
1	2	3
4	5	6
7	8	9



Beispiel Vereinigung

WEINLISTE

Name
La Rose Grand Cru
Creek Shiraz
Zinfandel
Pinot Noir
Riesling Reserve

EMPFEHLUNG

Wein
La Rose Grand Cru
Riesling Reserve
Merlot Selection
Sauvignon Blanc

◆ $WEINLISTE \cup \beta_{Name \leftarrow Wein}(EMPFEHLUNG)$

Name
La Rose Grand Cru
Creek Shiraz
Zinfandel
Pinot Noir
Riesling Reserve
Merlot Selection
Sauvignon Blanc



Mengenoperationen: Differenz

- ◆ Differenz $r_1 - r_2$ eliminiert die Tupel aus der ersten Relation, die auch in der zweiten Relation vorkommen
- ◆ Semantik: für $r_1(R)$ und $r_2(R)$

$$r_1 - r_2 := \{ t \mid t \in r_1 \wedge t \notin r_2 \}$$

- ◆ Beispiel:

R:

A	B	C
1	2	3
4	5	6

S:

A	B	C
4	5	6
7	8	9

R - S:

A	B	C
1	2	3



Beispiel Differenz

- ◆ WEINLISTE - $\beta_{\text{Name} \leftarrow \text{Wein}}(\text{EMPFEHLUNG})$
ergibt

Name
Creek Shiraz
Zinfandel
Pinot Noir



Mengenoperationen: Durchschnitt

- ◆ Durchschnitt $r_1 \cap r_2$: ergibt die Tupel, die in beiden Relationen gemeinsam vorkommen
- ◆ Semantik: für $r_1(R)$ und $r_2(R)$

$$r_1 \cap r_2 := \{ t \mid t \in r_1 \wedge t \in r_2 \}$$

- ◆ Beispiel:

R:

A	B	C
1	2	3
4	5	6

S:

A	B	C
4	5	6
7	8	9

R ∩ S:

A	B	C
4	5	6

- ◆ Durchschnitt \cap wegen $r_1 \cap r_2 = r_1 - (r_1 - r_2)$ überflüssig



Beispiel Durchschnitt

- ◆ $WEINLISTE \cap \beta_{Name \leftarrow Wein}(EMPFEHLUNG)$
ergibt

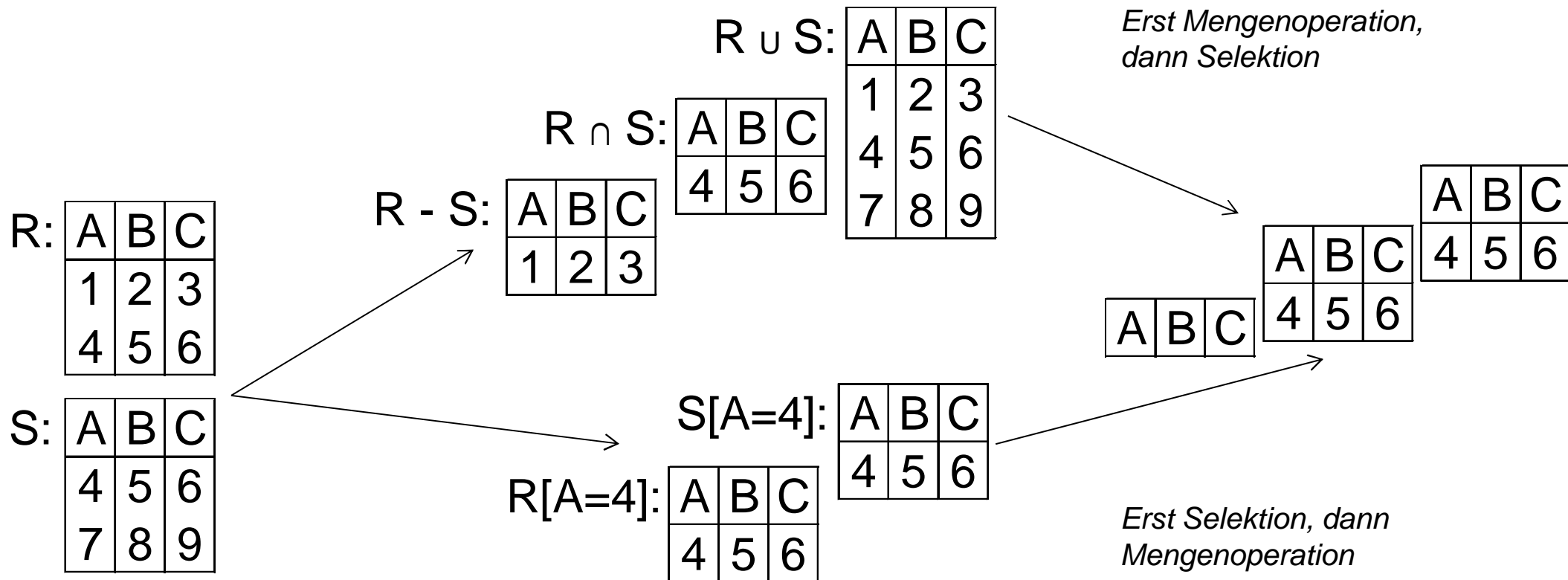
Name
La Rose Grand Cru
Riesling Reserve



Mengenoperationen : Gesetze zur Umformung

♦ Distributivität bzgl. \cap , \cup , $-$

$$\sigma_{A=a}(r \cup s) = \sigma_{A=a}(r) \cup \sigma_{A=a}(s)$$





- ◆ **Spalten ausblenden:** Projektion π
- ◆ **Zeilen heraussuchen:** Selektion σ
- ◆ **Tabellen verknüpfen:** Verbund (Join) \bowtie und Kreuzprodukt \times
- ◆ **Tabellen vereinigen:** Vereinigung \cup
- ◆ **Tabellen voneinander abziehen:** Differenz $-$ und Schnittmenge \cap
- ◆ **Spalten umbenennen:** Umbenennung β
(wichtig für \bowtie und $\cup, -$)



Unabhängigkeit und Vollständigkeit

- ◆ Eine Anfragesprache heißt **relational vollständig**, wenn jede Operation der relationalen Algebra in der Sprache durch (einen oder mehrere) Befehle umgesetzt werden kann
- ◆ Es gibt eine minimale Menge von Operationen innerhalb der relationalen Algebra, aus denen alle andere Operationen zusammengesetzt werden können: $\Omega = \pi, \sigma, \bowtie, \beta, \cup$ und -
 - Ω ist **unabhängig**: kein Operator kann weggelassen werden ohne Vollständigkeit zu verlieren
 - Andere unabhängige, vollständige Menge: \bowtie und β durch \times ersetzen
- ◆ Damit: es genügt, zu zeigen, dass alle Operationen aus Ω in einer Anfragesprache ausgedrückt werden können, um zu zeigen, dass diese relational vollständig ist
- ◆ SQL ist relational vollständig!



- ◆ Welche Strukturellen Elemente des relationalen Datenmodells kennen wir?
- ◆ Welche Operationen der relationalen Algebra gibt es?
- ◆ Welche Menge von Operationen der relationalen Algebra sind relational vollständig und unabhängig? Wieso ist das wichtig?
- ◆ Welche Integritätsbedingungen kennen wir aus dem relationalen Datenmodell?