Teil VIII

Grundlagen von Anfragen: Algebra & Kalkül

Grundlagen von Anfragen: Algebra & Kalkül

Lernziele für heute ...

- Verständnis der formalen Grundlagen relationaler Anfragesprachen
- Kenntnisse zur Formulierung von Anfragen in der relationalen Algebra
- Kenntnisse zur Formulierung von Kalkülanfragen

Einführung

- bisher:
  - Relationenschemata mit Basisrelationen, die in der Datenbank gespeichert sind
- jetzt:
  - „abgeleitete“ Relationenschemata mit virtuellen Relationen, die aus den Basisrelationen berechnet werden (Basisrelationen bleiben unverändert)

Begriffe

- Anfrage: Folge von Operationen, die aus den Basisrelationen eine Ergebnisrelation berechnet
  - Ergebnisrelation interaktiv auf dem Bildschirm anzeigen oder
  - per Programm weiterverarbeiten („Einbettung“)
- Sicht: Folge von Operationen, die unter einem Sichtnamen langfristig abgespeichert wird und unter diesem Namen wieder aufgerufen werden kann; ergibt eine Sichtrelation
- **Snapshot:** Ergebnisrelation einer Anfrage, die unter einem Snapshot-Namen abgelegt wird, aber nie ein zweites Mal (mit geänderten Basisrelationen) berechnet wird (etwa Jahresbilanzen)

Kriterien für Anfragesprachen

- **Ad-Hoc-Formulierung**: Benutzer soll eine Anfrage formulieren können, ohne ein vollständiges Programm schreiben zu müssen
- **Deskriptivität**: Benutzer soll formulieren „Was will ich haben?“ und nicht „Wie komme ich an das, was ich haben will?“
- **Mengenorientiertheit**: jede Operation soll auf Mengen von Daten gleichzeitig arbeiten, nicht navigierend nur auf einzelnen Elementen („one-tuple-at-a-time“)
- **Abgeschlossenheit**: Ergebnis ist wieder eine Relation und kann wieder als Eingabe für die nächste Anfrage verwendet werden

Kriterien für Anfragesprachen /2

- **Adäquatheit**: alle Konstrukte des zugrundeliegenden Datenmodells werden unterstützt
- **Orthogonalität**: Sprachkonstrukte sind in ähnlichen Situationen auch ähnlich anwendbar
- **Optimierbarkeit**: Sprache besteht aus wenigen Operationen, für die es Optimierungsregeln gibt
- **Effizienz**: jede Operation ist effizient ausführbar (im Relationenmodell hat jede Operation eine Komplexität $\leq O(n^2), n$ Anzahl der Tupel einer Relation).

Kriterien für Anfragesprachen /3

- **Sicherheit**: keine Anfrage, die syntaktisch korrekt ist, darf in eine Endlosschleife geraten oder ein unendliches Ergebnis liefern
- **Eingeschränktheit**: (folgt aus Sicherheit, Optimierbarkeit, Effizienz) Anfragesprache darf keine komplette Programmiersprache sein
- **Vollständigkeit**: Sprache muss mindestens die Anfragen einer Standardsprache (wie etwa die in diesem Kapitel einzuführende Relationenalgebra oder den sicheren Relationenkalkül) ausdrücken können

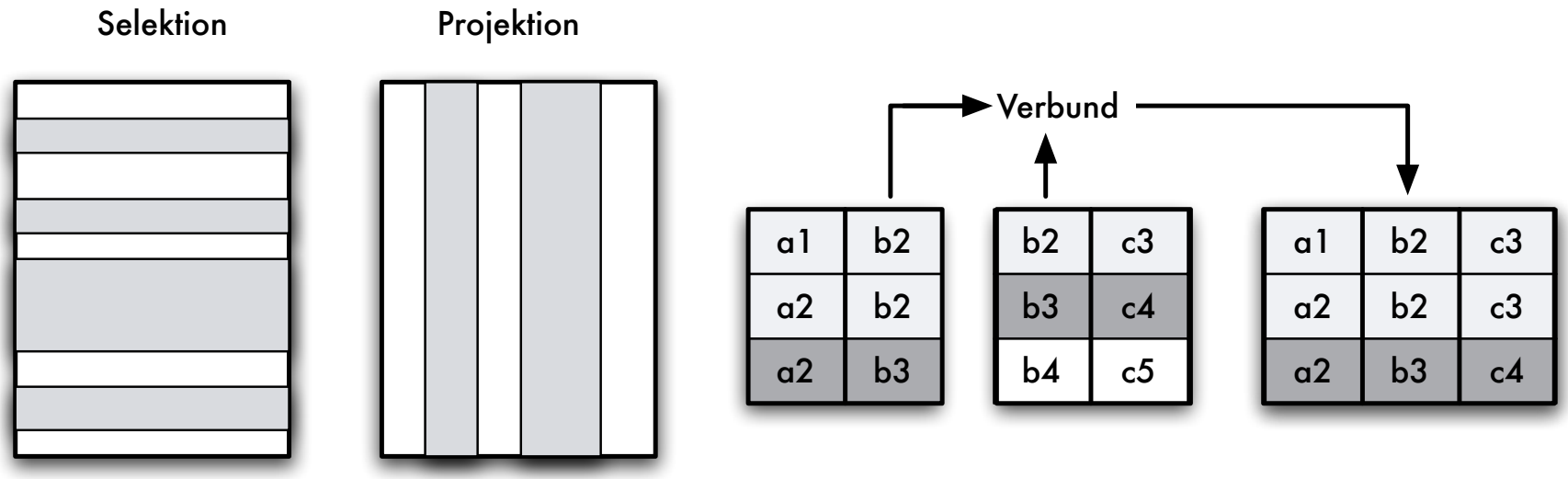
Anfragealgebren

- Mathematik: Algebra definiert durch Wertebereich und auf diesem definierte Operatoren
- für Datenbankanfragen: Inhalte der Datenbank sind Werte, und Operatoren definieren Funktionen zum Berechnen von Anfrageergebnissen
  - Relationenalgebra
  - Algebra-Erweiterungen

Relationenalgebra

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

Relationenalgebra: Übersicht



Projektion

- Syntax

$$\pi_{\textit{Attributmenge}}\,(\textit{Relation})$$

- 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: Beispiel

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

Region
South Australia
Kalifornien
Bordeaux
Hessen

Projektion: Beispiel 2

$$\pi_{\texttt{Anbaugebiet},\texttt{Region}}(\texttt{ERZEUGER})$$

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

Selektion

- Syntax

$$\sigma_{\mathit{Bedingung}}(\mathit{Relation})$$

- Semantik (für $A \in R$)

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

Selektionsbedingungen

- **Konstantenselektion**

  $$\texttt{Attribut}\ \theta\ \texttt{Konstante}$$

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

  $$\texttt{Attribut1}\ \theta\ \texttt{Attribut2}$$
- logische Verknüpfung mehrerer Konstanten- oder Attribut-Selektionen mit $\wedge$, $\vee$ oder $\neg$

Selektion: Eigenschaften

- Kommutativität

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

- falls $A \in X$, $X \subseteq R$

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

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

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

Selektion: Beispiel

$$\sigma_{\texttt{Jahrgang}>2000}(\texttt{WEINE})$$

WeinID	Name	Farbe	Jahrgang	Weingut
2168	Creek Shiraz	Rot	2003	Creek
3456	Zinfandel	Rot	2004	Helena
2171	Pinot Noir	Rot	2001	Creek
4961	Chardonnay	Weiß	2002	Bighorn

Verbund

- Syntax des (natürlichen) Verbundes (engl.: natural join)

$$\texttt{Relation1} \bowtie \texttt{Relation2}$$

- Semantik

$$\begin{aligned} r_1 \bowtie r_2 \quad := \quad & \{t \mid t(R_1 \cup R_2) \wedge \\ & [\forall i \in \{1,2\} \exists t_i \in r_i : t_i = t(R_i)]\} \end{aligned}$$

- Verbund verknüpft Tabellen über gleichbenannten Spalten bei gleichen Attributwerten

Verbund: Eigenschaften

- Schema für $r(R) \bowtie r(S)$ ist Vereinigung der Attributmengen $RS = R \cup S$
- aus $R_1 \cap R_2 = \{\}$ folgt $r_1 \bowtie r_2 = r_1 \times r_2$
- Kommutativität: $r_1 \bowtie r_2 = r_2 \bowtie r_1$
- Assoziativität: $(r_1 \bowtie r_2) \bowtie r_3 = r_1 \bowtie (r_2 \bowtie r_3)$
- daher erlaubt:

$$\bowtie_{i=1}^{p} r_i$$

Verbund: Beispiel

$$\texttt{WEINE} \bowtie \texttt{ERZEUGER}$$

WeinID	**Name**	...	**Weingut**	**Anbaugebiet**	**Region**
1042	La Rose Grand Cru	...	Ch. 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

Umbenennung

- Syntax

$$\beta_{neu \leftarrow alt}(Relation)$$

- Semantik

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

- ändert Attributnamen von *alt* in *neu*

$$\beta_{\text{Name} \leftarrow \text{Nachname}}\ (\texttt{KRITIKER})$$

- 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

Berechnung des Kreuzproduktes

- natürlicher Verbund entartet zum Kreuzprodukt, wenn keine gemeinsamen Attribute existieren
- Erzwingen durch Umbenennung
  - Beispiel: $R1(A, B, C)$ und $R2(C, D)$

$$R1 \times R2 \equiv R1 \bowtie \beta_{E \leftarrow C}(R2)$$

- Kreuzprodukt + Selektion simuliert natürlichen Verbund

$$R1 \bowtie R2 \equiv \sigma_{R1.C=R2.C}(R1 \times R2)$$

Mengenoperationen: Semantik

- formal für $r_1(R)$ und $r_2(R)$
  - Vereinigung $r_1 \cup r_2 := \{t \mid t \in r_1 \vee t \in r_2\}$
  - Durchschnitt $r_1 \cap r_2 := \{t \mid t \in r_1 \wedge t \in r_2\}$
  - Differenz $r_1 - r_2 := \{t \mid t \in r_1 \wedge t \notin r_2\}$
- Durchschnitt $\cap$ wegen $r_1 \cap r_2 = r_1 - (r_1 - r_2)$ überflüssig

Unabhängigkeit und Vollständigkeit

- Minimale Relationenalgebra:

$$\Omega = \pi, \sigma, \bowtie, \beta, \cup \text{ und } -$$

- unabhängig: kein Operator kann weggelassen werden ohne Vollständigkeit zu verlieren
- andere unabhängige Menge: $\bowtie$ und β durch $\times$ ersetzen
- Relationale Vollständigkeit: jede andere Menge von Operationen genauso mächtig wie Ω
- strenge relationale Vollständigkeit: zu jedem Ausdruck mit Operatoren aus Ω gibt es einen Ausdruck auch mit der anderen Menge von Operationen

Erweiterungen der Relationenalgebra

- weitere Verbundoperationen
- Division
- Gruppierung und geschachtelte Relationen
- ...

Verbundvarianten

- für $L(AB)$, $R(BC)$, $S(DE)$
- Gleichverbund (engl. *equi-join*): Gleichheitsbedingung über explizit angegebene und evtl. verschiedene Attribute

$$r(R) \bowtie_{C=D} r(S)$$

- Theta-Verbund (engl. θ*-join*): beliebige Verbundbedingung

$$r(R) \bowtie_{C>D} r(S)$$

- Semi-Verbund: nur Attribute eines Operanden erscheinen im Ergebnis

$$r(L) \ltimes r(R) = \pi_L(r(L) \bowtie r(R))$$

- äußere Verbunde (engl. *outer join*)

Äußere Verbunde

- Übernahme von „dangling tuples“ in das Ergebnis und Auffüllen mit Nullwerten
- **voller äußerer Verbund** übernimmt alle Tupel beider Operanden

$$r \mathbin{⟗} s$$

- **linker äußerer Verbund** übernimmt alle Tupel des linken Operanden

$$r \mathbin{⟕} s$$

- **rechter äußerer Verbund** übernimmt alle Tupel des rechten Operanden

$$r \mathbin{⟖} s$$

Äußere Verbunde /2

LINKS

A	B
1	2
2	3

RECHTS

B	C
3	4
4	5

$\bowtie$

A	B	C
2	3	4

$\supset\!\bowtie\!\subset$

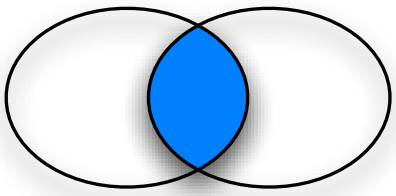
A	B	C
1	2	$\perp$
2	3	4
$\perp$	4	5

$\supset\!\bowtie$

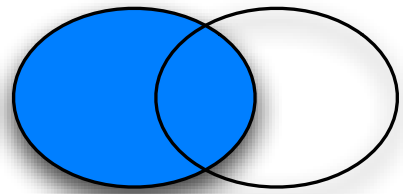
A	B	C
1	2	$\perp$
2	3	4

$\bowtie\!\subset$

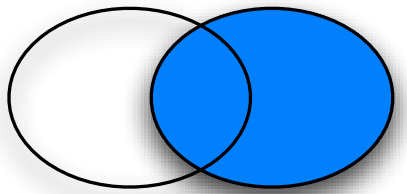
A	B	C
2	3	4
$\perp$	4	5



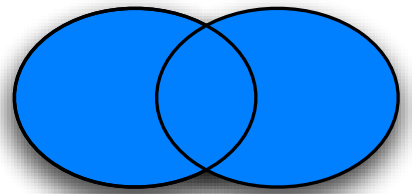
natural join



left outer join



right outer join



full outer join

Problem: Quantoren

- Allquantor in Relationenalgebra ausdrücken, obwohl in Selektionsbedingungen nicht erlaubt
- Division (kann aus Ω hergeleitet werden)
- $r_1(R_1)$ und $r_2(R_2)$ gegeben mit $R_2 \subseteq R_1$, $R' = R_1 - R_2$. Dann ist

$$\begin{aligned} r'(R') &= \{t \mid \forall t_2 \in r_2 \exists t_1 \in r_1 : t_1(R') = t \wedge t_1(R_2) = t_2\} \\ &= r_1 \div r_2 \end{aligned}$$

- Division von r_1 durch r_2

$$r_1 \div r_2 = \pi_{R'}(r_1) - \pi_{R'}((\pi_{R'}(r_1) \bowtie r_2) - r_1)$$

Division: Beispiel

WEIN_EMPFEHLUNG

Wein	Kritiker
La Rose Grand Cru	Parker
Pinot Noir	Parker
Riesling Reserve	Parker
La Rose Grand Cru	Clarke
Pinot Noir	Clarke
Riesling Reserve	Gault-Millau

GUIDES1

Kritiker
Parker
Clarke

GUIDES2

Kritiker
Parker
Gault-Millau

Divisionsbeispiele

- Division mit erster Tabelle

$$\texttt{WEIN\_EMPFEHLUNG} \div \texttt{GUIDES1}$$

liefert

Wein
La Rose Grand Cru
Pinot Noir

- Division mit zweiter Kritikerliste

$$\texttt{WEIN\_EMPFEHLUNG} \div \texttt{GUIDES2}$$

liefert

Wein
Riesling Reserve

Begriff Division

- Analogie zur arithmetischen Operation der ganzzahligen Division

Die ganzzahlige Division ist in dem Sinne die Inverse zur Multiplikation, indem sie als Ergebnis die größte Zahl liefert, für die die Multiplikation mit dem Divisor kleiner ist als der Dividend.
Analog gilt: $r = r_1 \div r_2$ ist die größte Relation, für die $r \bowtie r_2 \subseteq r_1$ ist.

Division in SQL

- Simulation des Allquantors (Division)
- mit doppelter Negation:

```
select distinct Wein
from WEIN_EMPFEHLUNG w1
where not exists (
   select * from GUIDES2 g
   where not exists (
      select * from WEIN_EMPFEHLUNG w2
      where g.Kritiker = w2.Kritiker and w1.Wein = w2.Wein))
```

  - „Gib alle Weine aus, so dass kein Wein existiert, der nicht von allen Kritikern in der Relation `GUIDES2` empfohlen wird“.
  - (Wir verwenden die Relation `GUIDES2`, weil die im Lehrbuch angegebene Ergebnisrelation sich auf diese Vergleichsrelation bezieht.)

Gruppierung

- Gruppierungsoperator γ:

$$\gamma_{f_1(x_1),f_2(x_2),\ldots,f_n(x_n);A}(r(R))$$

  - erweitert Attributschema von $r(R)$ um neue Attribute, die mit den Funktionsanwendungen $f_1(x_1), f_2(x_2), \ldots, f_n(x_n)$ korrespondieren
  - Anwendung der Funktionen $f_i(x_i)$ auf die Teilmenge derjenigen Tupel von $r(R)$ die gleiche Attributwerte für die Attribute A haben

```
select f1(x1), f2(x2), ..., fn(xn), A
from R
group by A
```

Semantik des Gruppierungsoperators

- leere Attributmenge $A = \emptyset$:

$$\gamma_{F(X);\emptyset}(r(R)) = r(R) \times r(R)^{F(X)}$$

  mit $r(R)^{F(X)}$ ist Relation mit Attribut $F(X)$ und einem Tupel als Wert von $F(X)$ auf $r(R)$

- ohne Funktion:

$$\gamma_{\emptyset;\emptyset}(r(R)) = r(R)$$

- allgemeiner Fall:

$$\gamma_{F(X);A}(r(R)) = \bigcup_{t \in R} \gamma_{F(X);\emptyset}(\sigma_{A=t.A}(r(R)))$$

Anfragekalküle

- Kalkül: eine formale logische Sprache zur Formulierung von Aussagen
- Ziel: Einsatz eines derartigen Kalküls zur Formulierung von Datenbank-Anfragen
- Logikbasierter Ansatz:
  - Datenbankinhalte entsprechen Belegungen von Prädikaten einer Logik
  - Anfragen abgeleiteten Prädikaten

Ein allgemeiner Kalkül

- Motivation: mathematische Notation

$$\{x^2 \mid x \in \mathbb{N} \wedge x^3 > 0 \wedge x^3 < 1000\}$$

- Anfrage hat die Form

$$\{f(\bar{x}) \mid p(\bar{x})\}$$

  - $\bar{x}$ bezeichnet Menge von freien Variablen

$$\bar{x} = \{x_1 : D_1, \ldots, x_n : D_n\}$$

Ein allgemeiner Kalkül /2

- Funktion f bezeichnet Ergebnisfunktion über $\bar{x}$
  - wichtige Spezialfälle: Angabe einer Variable selber (f ist hier die Identitätsfunktion) und Tupelkonstruktion (Ergebnis vom Typ **tuple of**)
- p Selektionsprädikat über freien Variablen $\bar{x}$
  - Terme aus Variablen, Konstanten und Funktionsanwendungen
  - Prädikate der Datentypen, etwa $\leq$, $<$, $>$, $\geq$, ...
    $\rightarrow$ atomare Formeln über Termen
  - Bezug zur aktuellen Datenbank $\rightarrow$ Datenbankprädikate, z.B. Relationennamen im RM
  - prädikatenlogischen Operatoren $\wedge$, $\vee$, $\neg$, $\forall$, $\exists$
    $\rightarrow$ Formeln

Ergebnisbestimmung einer Anfrage

$$\bar{x} = \{x_1 : D_1, \ldots, x_n : D_n\}$$

1. Bestimme aller Belegungen der freien Variablen in $\bar{x}$, für die das Prädikat p wahr wird.
2. Wende Funktion f auf die durch diese Belegungen gegebenen Werte an.

Unter welchen Umständen liefern Kalkülanfragen endliche Ergebnisse?

$\rightarrow$ Sicherheit von Anfragen

Relationale Kalküle

- **Bereichskalkül**: Variablen nehmen Werte elementarer Datentypen (*Bereiche*) an
- **Tupelkalkül**: Variablen variieren über Tupelwerte (entsprechend den Zeilen einer Relation)

Tupelkalkül

- Grundlage von SFW-Anfragen in SQL
- Variablen sind tupelwertig
- Beispiel:

$$\{w \mid w \in \texttt{WEINE} \wedge w.\texttt{Farbe} = \texttt{'Rot'}\}$$

Tupelkalkül: Beispiele

- konstruierte Tupel

$$\{\langle w.\texttt{Name}, w.\texttt{Weingut}\rangle \mid w \in \texttt{WEINE} \wedge w.\texttt{Farbe} = \texttt{'Rot'}\}$$

- Verbund

$$\{\langle e.\texttt{Weingut}\rangle \mid e \in \texttt{ERZEUGER} \wedge w \in \texttt{WEINE} \wedge e.\texttt{Weingut} = w.\texttt{Weingut}\}$$

- Schachtelung

$$\{\langle w.\texttt{Name}, w.\texttt{Weingut}\rangle \mid w \in \texttt{WEINE} \wedge$$
$$\exists e \in \texttt{ERZEUGER}(w.\texttt{Weingut} = e.\texttt{Weingut} \wedge e.\texttt{Region} = \texttt{'Bordeaux'})\}$$

Motivation: Die Sprache QBE

- „Query by Example“
- Anfragen in QBE: Einträge in Tabellengerüsten
- Intuition: Beispieleinträge in Tabellen
- Vorläufer verschiedener tabellenbasierter Anfrageschnittstellen kommerzieller Systeme
- basiert auf logischem Kalkül mit Bereichsvariablen

Anfragen in QBE: Selektion und Projektion

- Anfrage: „Alle Rock-Alben aus den Jahren vor 2006“

Album	ANr	Titel	Jahr	Genre	Preis	MNr
		P.	<2006	Rock		

$$\{t \mid \texttt{Album}(\_, t, j, \texttt{'Rock'}, \_, \_) \wedge j < 2006\}$$

Anfragen in QBE: Verbund

- Anfrage: „Alle Rock-Alben von deutschen Musikern“

Album	ANr	Titel	Jahr	Genre	Preis	MNr
		P.		Rock		_musiker

Musiker	MNr	Name	Land
	_musiker		Deutschland

$$\{t \mid \texttt{Album}(\_, t, \_, \texttt{'Rock'}, \_, m) \wedge \texttt{Musiker}(m, \_, \texttt{'Deutschland'})\}$$

Anfragen in QBE: Selbstverbund

- Anfrage: „Länder mit zwei oder mehr Musikern“

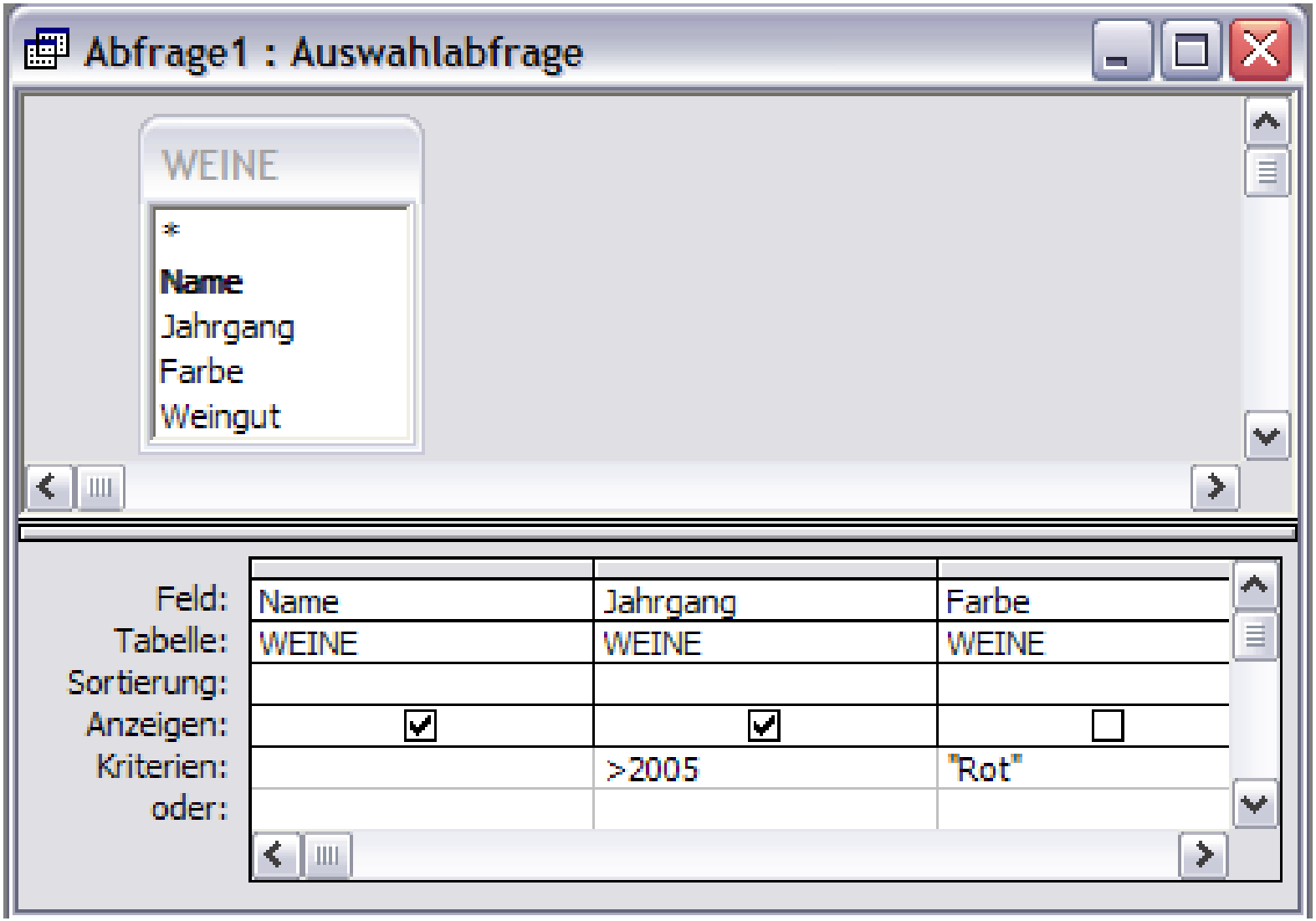
Musiker	MNr	Name	Land
	_eins ¬ _eins		**P.** _land _land

$$\{l \mid \texttt{Musiker}(x, \_, l) \wedge \texttt{Musiker}(y, \_, l) \wedge x \neq y\}$$

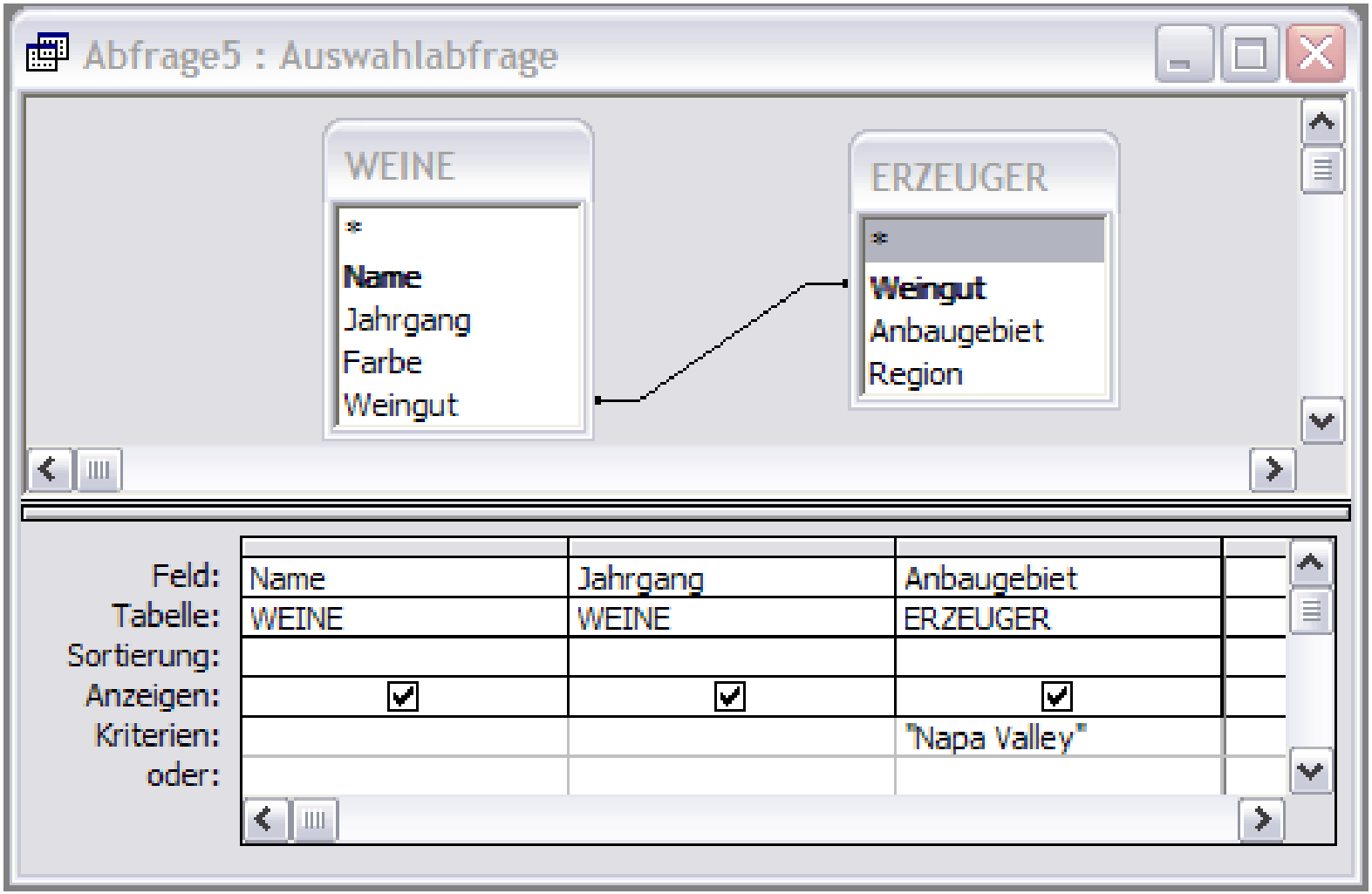
QBE in MS-Access

- MS-Access: Datenbankprogramm für Windows
  - Basisrelationen mit Schlüsseln
  - Fremdschlüssel über graphische Angabe von Beziehungen
  - graphische Definition von Anfragen (SQL-ähnlich)
  - interaktive Definition von Formularen und Berichten
- Unterstützung von QBE

Access: Projektion und Selektion



Access: Verbund



Bereichskalkül

- Terme:
  - Konstanten, etwa 42 oder `'MZ-4'`
  - Variablen zu Datentypen, etwa x
    Datentypangabe erfolgt in der Regel implizit und wird nicht explizit deklariert!
  - Funktionsanwendung $f(t_1, \ldots, t_n)$: Funktion f, Terme t_i, etwa $plus(12, x)$ bzw. in Infixnotation $12 + x$
- Atomare Formeln:
  - Prädikatanwendung $\Theta(t_1, \ldots, t_n)$, $\Theta \in \{<, >, \leq, \geq, \neq, =, \ldots\}$
    Datentypprädikat, Terme t_i
    Zweistellige Prädikate wie üblich in Infix-Notation.
    Beispiele: $x = y$, $42 > x$ oder $3 + 7 = 11$.

Bereichskalkül /2

- Atomare Formeln (fortg.):
  - Prädikatanwendungen für Datenbankprädikate, notiert als $R(t_1, \ldots, t_n)$ für einen Relationennamen R
    Voraussetzung: n muss die Stelligkeit der Relation R sein und alle t_i müssen vom passenden Typ sein
    Beispiel: ERZEUGER$(x, \text{'Hessen'}, z)$
- Formeln wie üblich mit $\wedge$, $\vee$, $\neg$, $\forall$ und $\exists$

Bereichskalkül /3

- *Anfragen*: $\{x_1, \ldots, x_n \mid \phi(x_1, \ldots, x_n)\}$
  - ϕ ist Formel über den in der Ergebnisliste aufgeführten Variablen x_1 bis x_n
  - Ergebnis ist eine Menge von Tupeln
  - Tupelkonstruktion erfolgt implizit aus den Werten der Variablen in der Ergebnisliste
- Beispiel

$$\{x \mid \text{ERZEUGER}(x, y, z) \wedge z = \text{'Hessen'}\}$$

Basiskalkül

- Einschränkung des Bereichskalküls:
  - **Wertebereich**: ganze Zahlen
  - **Datentypprädikate** werden wie bei der Relationenalgebra auf Gleichheit und elementare Vergleichsoperatoren eingeschränkt
  - **Funktionsanwendungen** sind nicht erlaubt; nur Konstanten dürfen neben Bereichsvariablen als Terme verwendet werden

Sichere Anfragen

- Sichere Anfragen (auch **semantisch sichere Anfragen**):

Anfragen, die für jeden Datenbankzustand $\sigma(\mathcal{R})$ ein endliches Ergebnis liefern

- Beispiel für nicht sichere Anfrage:

$$\{x, y \mid \neg R(x, y)\}$$

- Beispiel für sichere Anfrage:

$$\{x, y \mid R(x, y)\}$$

Sichere Anfragen /2

- Weiteres Beispiel für sichere Anfrage:

$$\{x, y \mid y = 10 \wedge x > 0 \wedge x < 10\}$$

Sicherheit folgt direkt aus den Regeln der Arithmetik.

Semantische Sicherheit ist im Allgemeinen nicht entscheidbar!

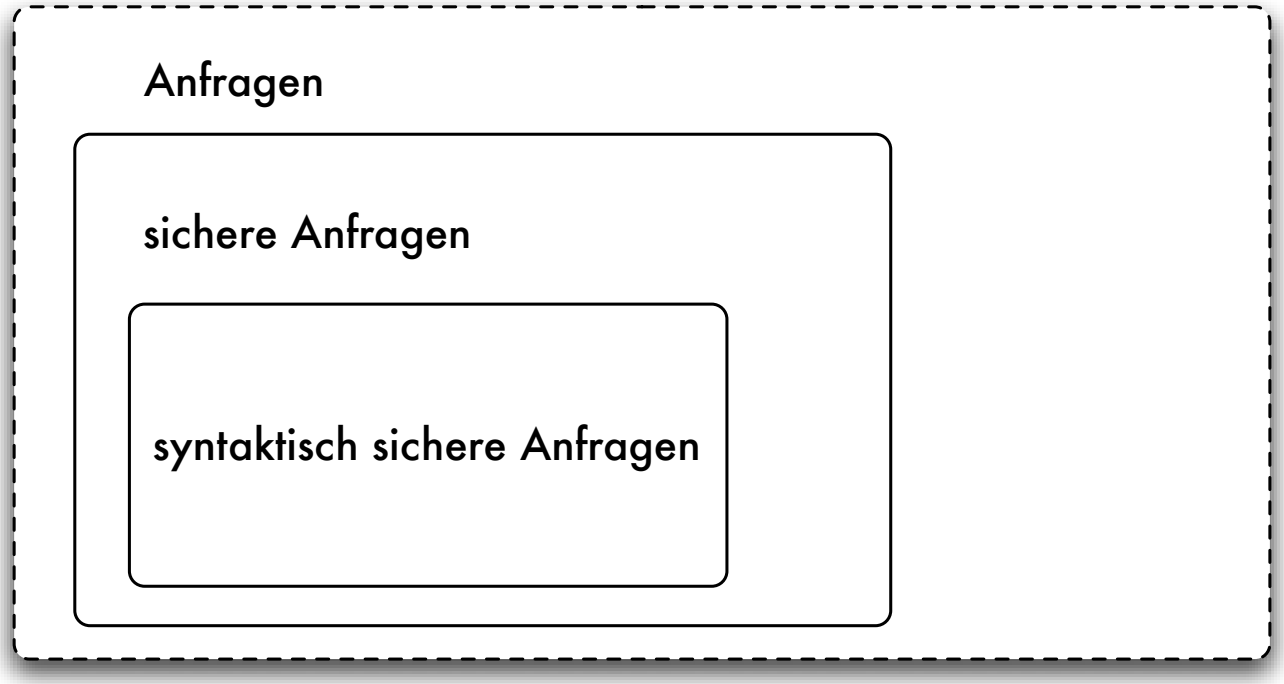
Syntaktisch sichere Anfragen

- Syntaktisch sichere Anfragen: Anfragen, die syntaktischen Einschränkungen unterliegen, um die semantische Sicherheit zu erzwingen.
- Grundidee:

Jede freie Variable x_i muss überall in $\phi(x_1, \dots)$ durch positives Auftreten $x_i = t$ oder $R(\dots, x_i, \dots)$ an endliche Bereiche gebunden werden.

- Bindung an endliche Bereiche muss für die ganze Bedingung gelten, also insbesondere für alle Zweige einer Disjunktion

Sichere Anfragen im Überblick



Beispiele Bereichskalkül

- Anfrage: „Alle Weingüter von Erzeugern in Hessen.“

$$\{x \mid \text{ERZEUGER}(x, y, z) \wedge z = \text{'Hessen'}\}$$

- Vereinfachte Notation: Ansonsten ungebundene Variablen (hier y und z) im Bedingungsteil existentiell mit $\exists$ gebunden
- Vollständige Version:

$$\{x \mid \exists y \exists z \text{ERZEUGER}(x, y, z) \wedge z = \text{'Hessen'}\}$$

- Einsparung von Bereichsvariablen, indem Konstanten als Parameter des Prädikats eingesetzt werden:

$$\{x \mid \text{ERZEUGER}(x, y, \text{'Hessen'})\}$$

Beispiele Bereichskalkül /2

- Abkürzung für beliebige, unterschiedliche existentiell gebundene Variablen ist _ Symbol:

$$\{x \mid \text{ERZEUGER}(x, \_, z) \wedge z = \text{'Hessen'}\}$$

- Verschiedene Auftreten des Symbols _ stehen hierbei für paarweise verschiedene Variablen

Beispiele Bereichskalkül /3

- Anfrage: „Regionen mit mehr als zwei Weingütern.“

$$\{z \mid \text{ERZEUGER}(x, y, z) \wedge \text{ERZEUGER}(x', y', z) \wedge x \neq x'\}$$

- Anfrage zeigt eine Verbundbildung über das dritte Attribut der ERZEUGER-Relation
- Verbundbildung kann einfach durch die Verwendung der selben Bereichsvariablen als Parameter in verschiedenen Relationsprädikaten erfolgen

Beispiele Bereichskalkül /4

- Anfrage: „Aus welcher Region sind welche Weine mit Jahrgang vor 1970 im Angebot?“

$$\{y, r \mid \texttt{WEINE}(x, y, z, j, w) \wedge \texttt{ERZEUGER}(w, a, r) \wedge j < 1970\}$$

- Verbund über zwei Relationen

Beispiele Bereichskalkül /5

- Anfrage: „Aus welchen Regionen gibt es Rotweine?“

$$\{z \mid \texttt{ERZEUGER}(x, y, z) \wedge \exists a \exists b \exists c \exists d(\texttt{WEIN}(a, b, c, d, x) \wedge c = \text{'Rot'})\}$$

- Einsatz einer existentiell gebundenen Unteranfrage
- derartige Unteranfragen können aufgrund der Regeln der Prädikatenlogik wie folgt aufgelöst werden:

$$\{z \mid \texttt{ERZEUGER}(x, y, z) \wedge (\texttt{WEIN}(a, b, c, d, x) \wedge c = \text{'Rot'})\}$$

Beispiele Bereichskalkül /6

- Anfrage: „Welches Weingut hat nur Weine mit Jahrgang nach 1995 im Angebot?“

$$\{x \mid \texttt{ERZEUGER}(x,y,z) \wedge \forall a \forall b \forall c \forall d(\texttt{WEIN}(a,b,c,d,x) \implies d > 1995)\}$$

- universell gebundene Teilformeln können nicht aufgelöst werden

Ausdrucksfähigkeit Bereichskalkül

Bereichskalkül ist streng relational vollständig, d.h. zu jedem Term τ der Relationenalgebra gibt es einen äquivalenten (sicheren) Ausdruck η des Bereichskalküls.

Umsetzung von Relationenoperationen

Geg.: Relationenschemata $R(A_1, \ldots, A_n)$ und $S(B_1, \ldots, B_m)$

- Vereinigung (für $n = m$)

$$R \cup S \hat{=} \{x_1 \ldots x_n \mid R(x_1, \ldots, x_n) \vee S(x_1, \ldots, x_n)\}$$

- Differenz (für $n = m$)

$$R - S \hat{=} \{x_1 \ldots x_n \mid R(x_1, \ldots, x_n) \wedge \neg S(x_1, \ldots, x_n)\}$$

- Natürlicher Verbund

$$R \bowtie S \hat{=} \quad \{x_1 \ldots x_n x_{n+1} \ldots x_{n+m-i} \mid R(x_1, \ldots, x_n) \wedge S(x_1, \ldots, x_i, x_{n+1}, \ldots, x_{n+m-i})\}$$

Annahme: die ersten i Attribute von R und S sind die Verbundattribute, also $A_j = B_j$ für $j = 1 \ldots i$

Umsetzung von Relationenoperationen /2

- Projektion

$$\pi_{\overline{A}}(R) \hat{=} \{y_1 \ldots y_k \mid \exists x_1 \ldots \exists x_n (R(x_1, \ldots, x_n) \wedge y_1 = x_{i_1} \wedge \cdots \wedge y_k = x_{i_k})\}$$

  Attributliste der Projektion: $\overline{A} = (A_{i_1}, \ldots, A_{i_k})$

- Selektion

$$\sigma_\phi(R) \hat{=} \{x_1 \ldots x_n \mid R(x_1, \ldots, x_n) \wedge \phi'\}$$

  ϕ' wird aus ϕ gewonnen, indem Variable x_i an Stelle der Attributnamen A_i eingesetzt werden

Zusammenfassung

- formale Modelle für Anfragen in Datenbanksystemen
- Relationenalgebra
  - operationaler Ansatz
  - Anfrage als Schachtelung von Operatoren auf Relationen
- Anfragekalküle
  - logikbasierter Ansatz
  - Anfragen als abgeleitete Prädikate
  - *im Buch: Abschnitte 4.2.3, 4.2.4 und 9.3*

Kontrollfragen

- Welche Bedeutung haben Äquivalenz, Unabhängigkeit und Vollständigkeit in der Relationenalgebra?
- Wie lässt sich die Semantik von erweiterten SQL-Operationen in der Relationenalgebra ausdrücken?
- Was unterscheidet Relationenalgebra und relationale Anfragekalküle?
- Welche Rolle spielt die Sicherheit von Anfragen?