Kapitel 3: Grundlagen von Anfragesprachen

Sprachparadigmen

- Relationenalgebra
- Relationenkalkül

3.1 Das relationale Datenmodell

Überblick

- Relationale Datenbanken repräsentieren Zustände einer Miniwelt durch Relationen.
- Wir unterscheiden die *Definition* der Relationen von den konkreten zeitabhängigen *Inhalten* (Zuständen).
- Wir reden von einem *Schema* einer Relation, wenn wir die Definition meinen, und von einer *Instanz* einer Relation, wenn wir eine entsprechende Menge von Tupeln meinen.

einige bekannte Relationen

Student

MatrNr	Name	Adresse	Semester
1223	Hans Eifrig	Seeweg 20	2
3434	Lisa Lustig	Bergstraße 11	4
1234	Maria Gut	Am Bächle 1	2

Kurs

KursNr	Institut	Name	Beschreibung
K010	DBIS	Datenbanken	Grundlagen von Datenbanken
K011	DBIS	Informationssysteme	Grundlagen von Informationssystemen

Belegung

MatrNr	KursNr	Semester	Note
1223	K010	WS2003/2004	2.3
1234	K010	SS2004	1.0

Attribute

- Objekte und Beziehungen werden durch ihre Eigenschaften repräsentiert. Eigenschaften werden im relationalen Modell als *Attribute* bezeichnet.
- Sei $X = \{A_1, \ldots, A_k\}$ eine endliche *Attributmenge*, wobei $k \geq 1$.
- Jedes Attribut $A \in X$ besitzt einen nicht-leeren *Wertebereich* $dom(A)$.
- Die Vereinigung aller Wertebereiche ergibt sich dann zu $dom(X) = \cup_{A \in X} dom(A)$.

Tupel

- Die Eigenschaften der Objekte und Beziehungen werden zu *Tupeln* zusammengefasst.
- Ein *Tupel* μ über Attributmenge X ist eine Abbildung

$$\mu : X \longrightarrow dom(X),$$

wobei $(\forall A \in X)\mu(A) \in dom(A)$.
- Sei $\mathrm{Tup}(X)$ im folgenden die Menge aller Tupel über X.

Beispiel

$\mu_1 =$ {`MatrNr` → 1223, `Name` → Hans Eifrig, `Adresse` → Seeweg 20, `Semester` → 2}

$\mu_2 =$ {`MatrNr` → 3434, `Name` → Lisa Lustig, `Adresse` → Bergstraße 11, `Semester` → 4}

$\mu_3 =$ {`MatrNr` → 1234, `Name` → Maria Gut, `Adresse` → Am Bächle 1, `Semester` → 2}

Tupel als Abbildungen versus Tupel als Vektoren

$\mu_1 =$ {`MatrNr` → 1223, `Name` → Hans Eifrig, `Adresse` → Seeweg 20, `Semester` → 2}

$\mu_2 =$ {`MatrNr` → 1223, `Adresse` → Seeweg 20, `Semester` → 2, `Name` → Hans Eifrig}

μ_1' `= (1223,Hans Eifrig,Seeweg 20,2)`
μ_2' `= (1223,Seeweg 20,2,Hans Eifrig)`

$\mu_1 = \mu_2$, aber $\mu_1' \neq \mu_2'$.

Relation

- Eine *Relation* r über einer Attributmenge X ist eine *endliche* Menge $r \subseteq \text{Tup}(X)$.
- Sei R ein *Relationsbezeichner*.
  Ein *(Relations)-Schema* zu R hat die Form $R(X)$. X ist hier eine endliche Attributmenge, das so genannte *Format* des Schemas.
  Anstatt $R(\{A_1, \ldots, A_k\})$ schreiben wir auch $R(A_1, \ldots, A_k)$.
  k ist die *Stelligkeit* des Relationsbezeichners.
  Auch:
  $$R(A_1 : dom(A_1), \ldots, A_k : dom(A_k))$$

Datenbank

- Ein *(relationales) Datenbank-Schema* $\mathcal{R}$ ist gegeben durch eine Menge von (Relations-) Schemata,
  $$\mathcal{R} := \{R_1(X_1), \ldots, R_m(X_m)\},$$
  bzw. $\mathcal{R} = \{R_1, \ldots, R_m\}$.
- Eine *Instanz* $\mathcal{I}$ zu einem relationalen Datenbankschema $\mathcal{R} = \{R_1, \ldots, R_m\}$ ist eine Menge von endlichen Relationen $\mathcal{I} := \{r_1, \ldots, r_m\}$, wobei $r_i \subseteq \text{Tup}(X_i)$ *Instanz* zu R_i für $1 \leq i \leq m$.

3.2 Relationenalgebra

Basisoperatoren

- Attribute aus Relationen herausstreichen: *Projektion*.
- Tupel aus Relationen auswählen: *Selektion*.
- Relationen miteinander verknüpfen: *Verbund*.
- Relationen wie Mengen verarbeiten: *Vereinigung, Differenz*.

Projektion

Student

MatrNr	Name	Adresse	Semester
1223	Hans Eifrig	Seeweg 20	2
3434	Lisa Lustig	Bergstraße 11	4
1234	Maria Gut	Am Bächle 1	2

$\Downarrow$

Student'

MatrNr	Name
1223	Hans Eifrig
3434	Lisa Lustig
1234	Maria Gut

Projektion einer Relation

- Sei $r \subseteq \mathrm{Tup}(X)$ eine Relation und $\emptyset \subset Y \subseteq X$.
- Der Ausdruck $\pi[Y]r$ heißt *Projektion* der Relation r auf Y. Es gilt:

$$\pi[Y]r := \{\mu \in \mathrm{Tup}(Y) \mid \exists \mu' \in r, \text{ so dass } \mu(A) = \mu'(A), A \in Y.\}.$$

Beispiel

$$r = \begin{array}{ccc} A & B & C \\ \hline a & b & c \\ a & a & c \\ c & b & d \end{array} \qquad \pi[A, C]r = \begin{array}{cc} A & C \\ \hline a & c \\ c & d \end{array}$$

Selektion

Kurs

KursNr	Institut	Name	Beschreibung
K010	DBIS	Datenbanken	Grundlagen von Datenbanken
K011	DBIS	Informationssysteme	Grundlagen von Informationssystemen
K100	MST	Mikrosystemtechnik	Grundlagen der Mikrosystemtechnik

$\Downarrow$

Kurs'

KursNr	Institut	Name	Beschreibung
K100	MST	Mikrosystemtechnik	Grundlagen der Mikrosystemtechnik

Selektionsbedingung

- Seien $A, B \in X$, $a \in dom(A)$ und sei $\theta \in \{=, \neq, <, \leq, >, \geq\}$ ein arithmetischer Vergleichsoperator.
- Eine (atomare) *Selektionsbedingung* α (bezüglich X) ist ein Ausdruck der Form $A\,\theta\,B$, bzw. $A\,\theta\,a$, bzw. $a\,\theta\,A$.
- Ein Tupel $\mu \in \text{Tup}(X)$ *erfüllt* eine Selektionsbedingung α, wenn gerade $\mu(A)\,\theta\,\mu(B)$, bzw. $\mu(A)\,\theta\,a$, bzw. $a\,\theta\,\mu(A)$.
- Atomare Selektionsbedingungen können mittels $\wedge$, $\vee$, $\neg$, (,) zu Formeln verallgemeinert werden.

Beispiel

$$X = \{A, B, C\}.$$
$$\mu_1 = (A \to 2, B \to 2, C \to 1),\ \mu_2 = (A \to 2, B \to 3, C \to 2)$$
$$\alpha_1 = (A = B),\ \alpha_2 = ((B > 1) \wedge (C > 1))$$

Welche Tupel erfüllen welche Selektionsbedingungen?

Selektion

- Sei $r \subseteq \text{Tup}(X)$ eine Relation und α eine Selektionsbedingung zu X.
- Der Ausdruck $\sigma[\alpha]r$ heißt *Selektion* der Relation r bezüglich α. Es gilt:

$$\sigma[\alpha]r := \{\mu \in \text{Tup}(X) \mid \mu \in r \wedge \mu \text{ erfüllt } \alpha\}.$$

Beispiel

$r =$

A	B	C
a	b	c
d	a	f
c	b	d

$\sigma[B = b]r =$

A	B	C
a	b	c
c	b	d

Vereinigung und Differenz

- Seien X, Y Attributmengen, wobei $X = Y$ und seien weiter $r \subseteq \text{Tup}(X), s \subseteq \text{Tup}(Y)$ zwei entsprechende Relationen.
- $$r \cup s = \{\mu \in \text{Tup}(X) \mid \mu \in r \vee \mu \in s\}.$$
  $$r - s = \{\mu \in \text{Tup}(X) \mid \mu \in r, \text{ wobei } \mu \notin s\}.$$

Beispiel

$r =$

A	B	C
a	b	c
d	a	f
c	b	d

$s =$

A	B	C
b	g	a
d	a	f

$r \cup s =$

A	B	C
a	b	c
d	a	f
c	b	d
b	g	a

$r =$

A	B	C
a	b	c
d	a	f
c	b	d

$s =$

A	B	C
b	g	a
d	a	f

$r - s =$

A	B	C
a	b	c
c	b	d

Verbund

Student

MatrNr	Name
1223	Hans Eifrig
3434	Lisa Lustig
1234	Maria Gut

Belegung

MatrNr	KursNr
1223	K010
1234	K010

Kurs

KursNr	K-Name
K010	Datenbanken
K011	Informations-systeme
K100	Mikrosystem-technik

$\Downarrow$

Student'

MatrNr	Name	KursNr	K-Name
1223	Hans Eifrig	K010	Datenbanken
1234	Maria Gut	K010	Datenbanken

Verbund

- Seien X, Y Attributmengen; XY sei im Folgenden eine Kurzschreibweise für $X \cup Y$.
- Seien weiter $r \subseteq \mathrm{Tup}(X), s \subseteq \mathrm{Tup}(Y)$ zugehörige Relationen.
- Der *(natürliche) Verbund* $\bowtie$ von r und s ist dann definiert:

$$r \bowtie s := \{\mu \in \mathrm{Tup}(XY) \mid \mu[X] \in r \wedge \mu[Y] \in s\}.$$

Beispiel

$r =$

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

$s =$

C	D
3	1
6	2
4	5

$r \bowtie s =$

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

Der *(natürliche) Verbund* $\bowtie$ von r und s ist definiert:

$$r \bowtie s := \{\mu \in \mathrm{Tup}(XY) \mid \mu[X] \in r \wedge \mu[Y] \in s\}.$$

Verbund fortgesetzt

Seien X_i, $1 \leq i \leq n$ Formate.

- Sei $X_1 \cap X_2 = \emptyset$

  $r_1 \times r_2 := r_1 \bowtie r_2$.
- $\bowtie_{i=1}^{n} r_i := \{\mu \in \mathrm{Tup}(\cup_{i=1}^{n} X_i) \mid \mu[X_i] \in r_i, 1 \leq i \leq n\}$.

Umbenennung

- Seien $X = \{A_1, \dots, A_k\}$, $Y = \{B_1, \dots, B_k\}$ Formate.
- Sei δ eine umkehrbar eindeutige Abbildung von X nach Y, wobei $dom(A) = dom(\delta(A))$. Gilt $\delta(A) = B$, so schreiben wir $A \to B$.
- Sei $r \subseteq \text{Tup}(X)$ eine Relation zu X.
- Die Umbenennung $\delta[X, Y]$ bezüglich r ist wie folgt:

$$\delta[X, Y]r := \{\mu \in \text{Tup}(Y) \mid \quad \exists \mu' \in r, \text{so dass} \quad \mu'(A_i) = \mu(\delta(A_i)), 1 \leq i \leq k\}$$

Beispiel

$X = \{A, B, C\}$, $Y = \{D, E, C\}$ und $\delta = \{A \to D, B \to E, C \to C\}$.

$r =$

A	B	C
a	b	c
d	a	f
c	b	d

$\delta[X, Y]r =$

D	E	C
a	b	c
d	a	f
c	b	d

Basisoperatoren

- Selektion, Projektion, Vereinigung, Differenz, Verbund und Umbenennung sind die Basisoperatoren der Relationenalgebra.
- Die Anwendung dieser Operatoren auf Relationen liefert als Ergebnis wiederum eine Relation.
- Die zulässigen Ausdrücke der Relationenalgebra können ausgehend von den Basisoperatoren induktiv definiert werden.
- Wir können andere nützliche Operatoren definiern.

weitere Operatoren

Seien X_i, $1 \leq i \leq n$, Formate und seien $r_i \subseteq \mathrm{Tup}(X_i)$, $1 \leq i \leq n$, Relationen.

- *Durchschnitt*. Sei $X_1 = X_2$.

$$r_1 \cap r_2 := r_1 - (r_1 - r_2).$$

- *θ-Verbund*. Sei $X_1 \cap X_2 = \emptyset$ und sei α eine beliebige Selektionsbedingung über $X_1 \cup X_2$.

$$r \bowtie_\alpha s := \sigma[\alpha](r \bowtie s).$$

Enthält α ausschließlich Gleichheitsvergleiche, dann redet man von einem *Equi-Verbund*.

Division

Seien X_1, X_2 Formate, $X_2 \subset X_1$, $Z = X_1 - X_2$ und weiter $r_2 \neq \emptyset$.

$$\begin{aligned} r_1 \div r_2 \quad &:= \quad \{\mu \in \mathrm{Tup}(Z) \mid \{\mu\} \times r_2 \subseteq r_1\} \\ &= \quad \pi[Z]r_1 - \pi[Z](((\pi[Z]r_1) \times r_2) - r_1). \end{aligned}$$

Beispiel

$r_1 =$

A	B	C	D
a	b	c	d
a	b	e	f
b	c	e	f
e	d	c	d
e	d	e	f
a	b	d	d

$r_2 =$

C	D
c	d
e	f

$r_1 \div r_2 =$

A	B
a	b
e	d

Beispiel: Welche Studierenden belegen alle Kurse?

Kurs(<u>KursNr</u>, Institut, Name, Beschreibung)
Belegung(<u>MatrNr</u>, <u>KursNr</u>, Semester, Note)

$\pi[\texttt{MatrNr}](\pi[\texttt{MatrNr,KursNr}]\texttt{Belegung} \div \pi[\texttt{KursNr}]\texttt{Kurs})$

Algebra als Anfragesprache

- Betrachte Ausdrücke Q über den *Relationsbezeichnern* eines Datenbankschemas, die dann bzgl. einer konkreten Instanz $\mathcal{I}$ ausgewertet werden können.

  Das Resultat der Auswertung von Q über $\mathcal{I}$ sei eine Relation $\mathcal{I}(Q)$, die *Antwort*.

  Das Schema von $\mathcal{I}(Q)$ ergibt sich aus dem Schema der Datenbank und der Struktur von Q.
- Jede Anfrage kann somit als Transformation über den Mengen der Instanzen zweier Datenbank-Schemata betrachtet werden.
- Es können nicht alle berechenbaren Transformationen über den Mengen der Instanzen zweier Datenbankschemata mittels der Algebra ausgedrückt werden.

  Das bekannteste Beispiel für diese eingeschränkte Mächtigkeit ist das Problem, zu einer binären Relation die transitive Hülle zu berechnen.

  Kein Ausdruck der relationalen Algebra kann dies für beliebige binäre Relationen bewerkstelligen!

Äquivalenz

Zwei Ausdrücke der Algebra Q, Q' heißen *äquivalent*, $Q \equiv Q'$, genau dann, wenn für jede Instanz $\mathcal{I}$ der Datenbank für die Antworten $\mathcal{I}(Q)$ und $\mathcal{I}(Q')$ der Anfragen Q und Q' gilt:

$$\mathcal{I}(Q) = \mathcal{I}(Q').$$

Beispiel: Welche Kurse belegt Lisa Lustig?

```
Student(MatrNr, SName, Adresse, Semester)
Kurs(KursNr, Institut, KName, Beschreibung)
Belegung(MatrNr, KursNr, Note)
```

- $\sigma[SName = LisaLustig](Student \bowtie (Kurs \bowtie Belegung))$
- $\sigma[SName = LisaLustig]Student \bowtie (Kurs \bowtie Belegung)$
- $Kurs \bowtie (\sigma[SName = LisaLustig]Student \bowtie Belegung))$

Äquivalenz

Zwei Ausdrücke der Algebra Q, Q' heißen *äquivalent*, $Q \equiv Q'$, genau dann, wenn für jede Instanz $\mathcal{I}$ der Datenbank für die Antworten $\mathcal{I}(Q)$ und $\mathcal{I}(Q')$ der Anfragen Q und Q' gilt:

$$\mathcal{I}(Q) = \mathcal{I}(Q').$$

Beispiele

Sei $\mathrm{attr}(\alpha)$ die in einer Selektionsbedingung α verwendete Menge von Attributen und seien $R, S, T \ldots$ Relationsbezeichner mit Formaten X, Y, Z.

- $Z \subseteq Y \subseteq X \Longrightarrow \pi[Z](\pi[Y]R) \equiv \pi[Z]R$.
- $\mathrm{attr}(\alpha) \subseteq Y \subseteq X \Longrightarrow \pi[Y](\sigma[\alpha]R) \equiv \sigma[\alpha](\pi[Y]R)$.
- $R \bowtie R \equiv R$.
  Betrachte die Definition des Verbundes $R_1 \bowtie R_2$. Da hier $R = R_1 = R_2$, mit X Format von R, folgt $R \bowtie R = \{\mu \in Tup(X) | \mu[X] \in \mathcal{I}(R)\}$ für eine beliebige Instanz $\mathcal{I}(R)$.
  Damit: $R \bowtie R = \{\mu \in Tup(X) | \mu \in \mathcal{I}(R)\} = \mathcal{I}(R)$.
- $X = Y \Longrightarrow R \cap S \equiv R \bowtie S$.
- $\mathrm{attr}(\alpha) \subseteq X, \mathrm{attr}(\alpha) \cap Y = \emptyset \Longrightarrow \sigma[\alpha](R \bowtie S) \equiv (\sigma[\alpha]R) \bowtie S$.

3.3 Relationenkalkül

Syntax

Die Formeln des Relationenkalküls (*R-Formeln*) werden aus Konstanten, Variablen, Relationsbezeichnern, Junktoren $\neg, \wedge, \vee$, Quantoren $\forall, \exists$ und Hilfszeichen '(', ')', ',' gebildet.

- Sei R ein Relationsbezeichner der Stelligkeit k. Seien weiter $a_1, \ldots, a_k$ Konstanten oder Variablen.

  $R(a_1, \ldots, a_k)$ ist eine (atomare) R-Formel.
- Eine *Selektionsbedingung* α ist von der Form $X\theta Y$, oder $X\theta a$, oder $a\theta X$, wobei X, Y Variablen, a eine Konstante und $\theta \in \{=, \neq, \leq, <, \geq, >\}$ ein Vergleichsoperator.

  α ist eine (atomare) R-Formel.
- Sei F eine R-Formel.

  $\neg F$ ist eine R-Formel.

- Sei X eine Variable und F eine R-Formel, die die Variable X enthält, jedoch keinen Ausdruck der Form $\exists X$, bzw. $\forall X$. X heißt in diesem Fall *frei* in der R-Formel F und anderenfalls *gebunden* in F.

  $\exists X\ F$ ist eine ($\exists$-*quantifizierte*) R-Formel.
  $\forall X\ F$ ist eine ($\forall$-*quantifizierte*) R-Formel.

  F ist der *Wirkungsbereich* des $\exists$-, bzw. $\forall$-Quantors.
- Seien F und G R-Formeln und sei $\mathcal{V}_F$, bzw. $\mathcal{V}_G$ die Menge der in F, bzw. G vorhandenen Variablen, wobei die Variablen in $\mathcal{V}_F \cap \mathcal{V}_G$ sowohl in F als auch in G frei.

  Die *Konjunktion* $(F \wedge G)$ ist eine R-Formel.
  Die *Disjunktion* $(F \vee G)$ ist eine R-Formel.

- Für Konjunktion und Disjunktion gilt das Assoziativgesetz; überflüssige Klammerungen lassen wir im Folgenden weg.
- Eine Anfrage Q über einem Datenbank-Schema $\mathcal{R}$ im Relationenkalkül, kurz *Kalkülanfrage*, hat die Form

$$\{(a_1, \ldots, a_n) \mid F\},$$

wobei F eine R-Formel zu $\mathcal{R}$ und $a_1, \ldots, a_n$ Variablen und Konstanten.
- Die Menge der Variablen unter den a_i muss hierbei gerade die Menge der freien Variablen in F sein.
- Soll ein Format $\{A_1, \ldots, A_n\}$ für die Menge der Antworten definiert werden, so schreiben wir

$$\{(a_1 : A_1, \ldots, a_n : A_n) \mid F\}.$$

Beispiel Projektion $\pi[A]R$

Sei ein Relationsschemata $R(A, B)$ gegeben.

$$r = \begin{array}{cc} A & B \\ \hline 1 & 2 \\ 2 & 2 \\ 2 & 1 \end{array}$$

$\{(X : A) \mid \exists Y\ R(X, Y)\}$

$$\begin{array}{c} A \\ \hline 1 \\ 2 \end{array}$$

Beispiel Projektion und Selektion $\pi[A](\sigma[B=2]R)$

Sei ein Relationsschemata $R(A,B)$ gegeben.

$$r = \begin{array}{cc} A & B \\ \hline 1 & 2 \\ 2 & 2 \\ 2 & 1 \end{array}$$

$\{(X:A) \mid \exists Y \; R(X,Y) \wedge Y = 2\}$

$\{(X:A) \mid R(X,2)\}$

$$\begin{array}{c} A \\ \hline 1 \\ 2 \end{array}$$

Beispiel Selektion $\sigma[A=B]R$

Sei ein Relationsschemata $R(A,B)$ gegeben.

$$r = \begin{array}{cc} A & B \\ \hline 1 & 2 \\ 2 & 2 \\ 2 & 1 \end{array}$$

$\{(X:A, Y:B) \mid R(X,Y) \wedge X = Y\}$

$\{(X:A, X:B) \mid R(X,X)\}$

$$\begin{array}{cc} A & B \\ \hline 2 & 2 \end{array}$$

Beispiel Verbund $R \bowtie S$

Seien Relationsschemata $R(A, B)$ und $S(B, C)$ gegeben.

$r =$

A	B
1	2
2	2
2	1

$s =$

B	C
1	1
1	2
3	1

$\{(X : A, Y : B, Z : C) \mid \exists U\ R(X, Y) \land S(U, Z) \land Y = U\}$

$\{(X : A, Y : B, Z : C) \mid R(X, Y) \land S(Y, Z)\}$

A	B	C
2	1	1
2	1	2

Beispiel Vereinigung $R \cup S$

Seien Relationsschemata $R(A, B)$ und $S(A, B)$ gegeben.

$r =$

A	B
1	2
2	2
2	1

$s =$

A	B
1	1
1	2
3	1

$\{(X : A, Y : B) \mid R(X, Y) \lor S(X, Y)\}$

A	B
1	2
2	2
2	1
1	1
3	1

Beispiel Differenz $R - S$

Seien Relationsschemata $R(A, B)$ und $S(A, B)$ gegeben.

$r =$

A	B
1	2
2	2
2	1

$s =$

A	B
1	1
1	2
3	1

$\{(X : A, Y : B) \mid R(X, Y) \land \neg S(X, Y)\}$

A	B
2	2
2	1

Beispiel Division $R \div S$

Seien Relationsschemata $R(A, B)$ und $S(B)$ gegeben.

$r =$

A	B
1	2
2	2
2	1

$s =$

B
1
2
1

mit Implikation:
$\{(X : A) \mid \forall Y \ (S(Y) \Rightarrow R(X, Y))\}$

A
2

logisch äquivalente Schreibweisen:
$\{(X : A) \mid \forall Y \ (\neg S(Y) \lor R(X, Y))\}$, oder

$\{(X : A) \mid \neg \exists Y \ (S(Y) \land \neg R(X, Y))\}$

Eine Antwort zu einer Anfrage Q darf nur aus solchen Konstanten gebildet werden, die in Q selbst oder in (irgendeinem Tupel) der betrachteten Datenbank-Instanz vorhanden sind.

Wertebereichsunabhängigkeit

- Sei $Q := \{(a_1, \ldots, a_n) \mid F\}$. Sei $\mathcal{I}$ eine Instanz zu $\mathcal{R}$ und *adom* diejenige Menge, die gerade alle Konstanten in Q und alle Konstanten aus $\mathcal{I}$ enthält. *adom* ist der *aktive* Wertebereich von Q; *adom* ist endlich.
- Q, bzw. F, heißen *wertebereichsunabhängig*, wenn für jede beliebige Menge $D \supset adom$ gilt:

$$Q(\mathcal{I}, adom) = Q(\mathcal{I}, D).$$

Beispiel: nicht wertebereichsunabhängige Anfragen

- Sei $R(A)$ ein Relationsschema und Q eine Anfrage der Form

$$\{X \mid \neg R(X)\},$$

  wobei $\mathcal{I}(R) = \{1\}$.
- Seien $R(A)$ und $S(B)$ Relationsschemata. Sei Q eine Anfrage der Form

$$\{(X, Y) \mid R(X) \vee S(Y)\},$$

  wobei $\mathcal{I}(R) = \{(1)\}$, bzw. $\mathcal{I}(S) = \emptyset$.

Sicherheit

Ist eine R-Formel F *sicher*, dann ist sie auch wertebereichsunabhängig.

Beispiel: nicht sichere und sichere Anfragen

- $\{X \mid \neg R(X)\}$ ist nicht sicher, jedoch
$$\{X \mid \neg R(X) \wedge S(X)\}$$
ist sicher.
- $\{(X, Z) \mid \exists Y(R(X, Y) \vee S(Y, Z))\}$ ist nicht sicher, jedoch
$$\{(X, Z) \mid \exists Y(R(X, Y) \wedge S(Y, Z))\}$$
ist sicher.
- $\{(X, Y) \mid R(X, Y) \wedge (S(X) \vee T(Y))\}$ ist nicht sicher, jedoch in der äquivalenten Schreibweise
$$\{(X, Y) \mid (R(X, Y) \wedge S(X)) \vee (R(X, Y) \wedge T(Y))$$
- $\{(X, Y, Z) \mid R(X, Y, Z) \wedge \neg(S(X, Y) \vee T(Y, Z))\}$ ist nicht sicher, jedoch in der äquivalenten Schreibweise
$$\{(X, Y, Z) \mid R(X, Y, Z) \wedge \neg S(X, Y) \wedge \neg T(Y, Z)$$

klassisches Resultat (Codd 1972)

Die Relationenalgebra und der sichere Relationenkalkül haben dieselbe Ausdrucksstärke:

> Zu jedem Ausdruck der Algebra existiert ein äquivalenter Ausdruck des sicheren Kalküls, und umgekehrt.

Eine Anfragesprache heißt *relational vollständig*, wenn sie mindestens die Ausdrucksstärke der relationalen Algebra besitzt.

3.4 empfohlene Lektüre

A Relational Model of Data for Large Shared Data Banks

E. F. Codd
IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on *n*-ary relations, a normal form for data base relations, and the concept of a universal data sublanguage are introduced. In Section 2, certain operations on relations (other than logical inference) are discussed and applied to the problems of redundancy and consistency in the user's model. [1]

[1] In: Communications of the ACM, Volume 13 , Issue 6 (June 1970). Kann aus dem Institutsnetz heraus vom ACM-Portal heruntergeladan werden.