Kapitel 2: Grundlagen von Anfragesprachen

Sprachparadigmen

- Relationenalgebra (in der Vorlesung)
- Relationenkalkül (siehe Literatur)

SQL - vorgestellt in den folgenden Kapiteln - basiert auf der Algebra und dem Kalkül gleichermaßen und enthält weitere für die praktische Anwendung sehr nützliche Sprachkonzepte.

Relationen dargestellt als Tabellen

Student

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

Kurs

<u>KursNr</u>	Institut	Name	Beschreibung
K010	DBIS	Datenbanken	Grundlagen von Datenbanken
K011	DBIS	Informationssysteme	Grundlagen von Informationssystemen

Belegung

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

2.1 Das relationale Datenmodell

Attribute und Tupel

- Sei $X = \{A_1, \dots, A_k\}$ 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 $dom(X) = \cup_{A \in X} dom(A)$.
- 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.

Student

<u>MatrNr</u>	Name	Adresse	Semester
1223	Hans Eifrig	Seeweg 20	2
⋮	⋮	⋮	⋮

Tupel als Abbildungen versus Tupel als Vektoren

$$\mu = \{\texttt{MatrNr} \rightarrow 1223,\ \texttt{Name} \rightarrow \text{Hans Eifrig},\ \texttt{Adresse} \rightarrow \text{Seeweg 20},\ \texttt{Semester} \rightarrow 2\}$$

$$\mu' = \{\texttt{MatrNr} \rightarrow 1223,\ \texttt{Adresse} \rightarrow \text{Seeweg 20},\ \texttt{Semester} \rightarrow 2,\ \texttt{Name} \rightarrow \text{Hans Eifrig}\}$$

μ_1 = (1223,Hans Eifrig,Seeweg 20,2)

μ_1' = (1223,Seeweg 20,2,Hans Eifrig)

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

Relation

- Eine *Relation* r über einer Attributmenge X ist eine *endliche* Menge $r \subseteq \mathrm{Tup}(X)$.
- 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 $r_i \subseteq \mathrm{Tup}(X_i)$ *Instanz* zu R_i für $1 \leq i \leq m$.

2.2 Relationenalgebra

Warum?

was berechnet dieser SQL-Ausdruck:

```
SELECT DISTINCT MatrNr FROM Belegung
MINUS
SELECT MatrNr FROM (
   SELECT MatrNr, KursNr FROM (
      (SELECT MatrNr FROM Belegung)
      CROSS JOIN
      (SELECT KursNr FROM Kurs) )
   MINUS
   SELECT MatrNr, KursNr FROM Belegung
)
```

...Notwendigkeit von Auswertungsoperatoren mit formal definierter Semantik!

Basisoperatoren der Relationenalgebra

- Attribute aus Relationen herausstreichen: *Projektion* π,
- Tupel aus Relationen auswählen: *Selektion* σ,
- Relationen miteinander verknüpfen: *Verbund* $\bowtie$,
- Relationen wie Mengen verarbeiten: *Vereinigung* $\cup$, *Differenz* $-$,
- Attribute umbenennen.

Beispiel Projektion einer Relation

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

Projektion einer Relation

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

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

Beispiel Selektion

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

Selektion

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

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

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 $\mu \in \mathrm{Tup}(X)$ *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

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

Welche Tupel erfüllen welche Selektionsbedingungen?

Beispiel Vereinigung und Differenz

$r =$

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

Vereinigung $\cup$ und Differenz $-$

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

Beispiel Verbund

$$r = \begin{array}{ccc} A & B & C \\ \hline 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 6 \end{array} \qquad s = \begin{array}{cc} C & D \\ \hline 3 & 1 \\ 6 & 2 \\ 4 & 5 \end{array} \qquad r \bowtie s = \begin{array}{cccc} A & B & C & D \\ \hline 1 & 2 & 3 & 1 \\ 4 & 5 & 6 & 2 \\ 7 & 8 & 6 & 2 \end{array}$$

Sei $\mu \in \mathrm{Tup}(X)$ ein Tupel über X, $Y \subseteq X$.

$$\mu = \{A \to 1, B \to 2, C \to 3, D \to 1\}$$

$Y = \{B, D\}$.

$$\mu[Y] = \{B \to 2, D \to 1\}$$

Der Ausdruck $\mu[Y]$ heißt *Projektion* des Tupels μ auf Y. Es gilt:

$$\mu[Y] \in \mathrm{Tup}(Y),$$

wobei $\mu[Y](A) = \mu(A), A \in Y$.

Beispiel Verbund

$$r = \begin{array}{ccc} A & B & C \\ \hline 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 6 \end{array} \qquad s = \begin{array}{cc} C & D \\ \hline 3 & 1 \\ 6 & 2 \\ 4 & 5 \end{array} \qquad r \bowtie s = \begin{array}{cccc} A & B & C & D \\ \hline 1 & 2 & 3 & 1 \\ 4 & 5 & 6 & 2 \\ 7 & 8 & 6 & 2 \end{array}$$

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:

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

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.

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

- Sei $X_1 \cap X_2 = \emptyset$ und bezeichne $\times$ das kartesische Produkt.

  $r_1 \times r_2 = r_1 \bowtie r_2$.

Beispiel Umbenennung

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

$r =$

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

Umbenennung

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

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

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 $r_i \subseteq \text{Tup}(X_i)$, $1 \leq i \leq n$, Relationen.

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

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

  Bemerkung: Da $X_1 = X_2$ gilt $r_1 \cap r_2 = r_1 \bowtie r_2$.
- 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*.

Beispiel Division

$r_1 =$

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

Division

Seien X_1, X_2 Formate, $X_2 \subset X_1$, $Z = X_1 - X_2$ und weiter $r_1 \subseteq Tup(X_1)$ (Dividend), $r_2 \subseteq Tup(X_2)$ (Divisor), wobei $r_2 \neq \emptyset$.

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

Beispiel: Welche Studierenden belegen alle Kurse?

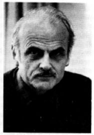
Kurs(<u>KursNr</u>, Institut, Name, Beschreibung)
Belegung(<u>MatrNr</u>, <u>KursNr</u>, Semester, Note)

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

empfohlene Lektüre

The 1981 ACM Turing Award Lecture

Delivered at ACM '81, Los Angeles, California, November 9, 1981



The 1981 ACM Turing Award was presented to Edgar F. Codd, an IBM Fellow of the San Jose Research Laboratory, by President Peter Denning on November 9, 1981 at the ACM Annual Conference in Los Angeles, California. It is the Association's foremost award for technical contributions to the computing community.

Codd was selected by the ACM General Technical Achievement Award Committee for his "fundamental and continuing contributions to the theory and practice of database management systems." The originator of the relational model for databases, Codd has made further important contributions in the development of relational algebra, relational calculus, and normalization of relations.

Edgar F. Codd joined IBM in 1949 to prepare programs for the Selective Sequence Electronic Calculator. Since then, his work in computing has encompassed logical design of computers (IBM 701 and Stretch), managing a computer center in Canada, heading the development of one of the first operating systems with a general multiprogramming capability, contributing to the logic of self-reproducing automata, developing high level techniques for software specification, creating and extending the relational approach to database management, and developing an English analyzing and synthesizing subsystem for casual users of relational databases. He is also the author of *Cellular Automata*, an early volume in the ACM Monograph Series.

Codd received his B.A. and M.A. in Mathematics from Oxford University in England, and his M.Sc. and Ph.D. in Computer and Communication Sciences from the University of Michigan. He is a Member of the National Academy of Engineering (USA) and a Fellow of the British Computer Society.

The ACM Turing Award is presented each year in commemoration of A. M. Turing, the English mathematician who made major contributions to the computing sciences.

Relational Database: A Practical Foundation for Productivity

E. F. Codd
IBM San Jose Research Laboratory

It is well known that the growth in demands from end users for new applications is outstripping the capability of data processing departments to implement the corresponding application programs. There are two complementary approaches to attacking this problem (and both approaches are needed): one is to put end users into direct touch with the information stored in computers; the other is to increase the productivity of data processing professionals in the development of application programs. It is less well known that a single technology, relational database management, provides a practical foundation for both approaches. It is explained why this is so.

While developing this productivity theme, it is noted that the time has come to draw a very sharp line between relational and non-relational database systems, so that the label "relational" will not be used in misleading ways. The key to drawing this line is something called a "relational processing capability."

CR Categories and Subject Descriptors: H.2.0 [**Database Management**]: General; H.2.1 [**Database Management**]: Logical Design–*data models*; H.2.4 [**Database Management**]: Systems

General Terms: Human Factors, Languages

Additional Key Words and Phrases: database, relational database, relational model, data structure, data manipulation, data integrity, productivity [1]

Author's Present Address: E. F. Codd, IBM Research Laboratory, 5600 Cottle Road, San Jose, CA 95193.

[1] In: Communications of the ACM, Volume 25 , Issue 2.