

Relationen-Algebra

Allgemeines
Schnitt, Vereinigung, Differenz
Projektion
Selektion
Verbund
Komplexer Algebra-Ausdruck

Aufsetzen und Füllen einer DB ist eine Aufgabe. (Und haben wir schon erledigt.)

Befragen der DB ist eine andere Aufgabe. Und das machen wir jetzt.

Tabellen sind Relationen. Befragen der Tabellen kann man als Operatoren auf Relationen betrachten, welche wieder Relationen liefern. Und damit Tabellen.

Basisoperationen:

- ▷ Mengenoperationen,
- ▷ Projektion,
- ▷ Selektion,
- ▷ Verbund.

Relationen-Algebra

Allgemeines

„Binäre Relation“: Seien A, B zwei Mengen. Eine **„binäre Relation“** R von A und B ist eine Teilmenge des kartesischen Produktes dieser Mengen, also

$$R \subseteq A \times B = \{(a, b) : a \in A, b \in B\}.$$

Allgemeiner ist eine **„ n -stellige Relation“** eine Teilmenge des kartesischen Produktes von n Mengen A_1, \dots, A_n .

Das hatten wir schon.

Bsp.: Mit den Mengen

$$A = \{1, 2\} \quad , \quad B = \{\alpha, \beta, \gamma\} \quad \text{und} \quad C = \{J, N\}$$

erhalten wir als ein mögliches Kreuzprodukt die Menge

$$A \times B \times C = \{(1, \alpha, J), \dots, (2, \gamma, N)\}$$

mit insgesamt 12 Elementen. Dreistellige Relation zu den gegebenen Grundmengen ist bspw.

$$R = \{(1, \beta, J), (1, \gamma, J), (2, \alpha, J), (2, \beta, N)\}.$$

Schnitt, Vereinigung, Differenz

Operationen aus der Mengenlehre bekannt. Sinnvoll auf Daten-Relationen, wenn diese gleiche Struktur besitzen.

„Vereinigungsverträglich“: Zwei Relationen $R \subseteq A_1 \times A_2 \times \dots \times A_n$ und $S \subseteq B_1 \times B_2 \times \dots \times B_n$ heißen **„vereinigungsverträglich“**, wenn für alle $i = 1, 2, \dots, n$ die Wertemengen A_i und B_i verträglich sind. D.h. den gleichen Wertetyp besitzen oder einen gemeinsamen Obertyp besitzen.

Bsp.:

$A \subseteq \text{INTEGER}$ und $B \subseteq \text{INTEGER}$ sind vereinigungsverträglich,
da *gleicher* Wertyp.

$A \subseteq \text{INTEGER}$ und $B \subseteq \text{REAL}$ sind vereinigungsverträglich,
da *gemeinsamer* Obertyp REAL.

$A \subseteq \text{VARCHAR}(100)$ und $B \subseteq \text{VARCHAR}(50)$ sind vereinigungsverträglich,
da *gemeinsamer* Obertyp TEXT.

Jetzt

„Mengenoperationen“: Seien die beiden Relationen R und S vereinigungsverträglich, dann bilden wir daraus neue Relationen über

$$R \cap S = \{t : t \in R \wedge t \in S\} \quad \text{„Schnittmenge“}$$

$$R \cup S = \{t : t \in R \vee t \in S\} \quad \text{„Vereinigungsmenge“}$$

$$R \setminus S = \{t : t \in R \wedge t \notin S\} \quad \text{„Differenzmenge“}$$

Anfrage an eine Datenbank mit zwei Tabellen

- ▷ **Schnittmenge**, Intersect: $R \cap S$, $R \text{ IN } S$
Suche nach Datensätzen, die in beiden Tabellen gemeinsam vorkommen.
- ▷ **Vereinigungsmenge**, Union: $R \cup S$, $R \text{ UN } S$
Betrachte alle Datensätze beider Tabellen gemeinsam.
- ▷ **Differenzmenge**, Difference: $R \setminus S$, $R \text{ DF } S$
Suchen nach Datensätzen, die nur in einer Tabelle vorkommen.

Bsp.: Zwei Ausgangstabellen

Verkäufer1

Verkäufer	Produkt	Käufer
Meier	Hose	Schmidt
Müller	Rock	Schmidt
Meier	Hose	Schulz

Verkäufer2

Verkäufer	Produkt	Käufer
Müller	Hemd	Schmidt
Müller	Rock	Schmidt
Meier	Rock	Schulz

Ergebnisse der Mengenoperationen

Verkäufer1 \cup Verkäufer2

Verkäufer	Produkt	Käufer
Meier	Hose	Schmidt
Müller	Rock	Schmidt
Meier	Hose	Schulz
Müller	Hemd	Schmidt
Meier	Rock	Schulz

Verkäufer1 \cap Verkäufer2

Verkäufer	Produkt	Käufer
Müller	Rock	Schmidt

Verkäufer1 \setminus Verkäufer2

Verkäufer	Produkt	Käufer
Meier	Hose	Schmidt
Meier	Hose	Schulz

Ergebnisse jeweils wieder eine Tabelle/Relation.

Projektion

Auswählen weniger Spalten aus einer Tabelle.

„Projektion“: Sei $R \subseteq A_1 \times A_2 \times \dots \times A_n$ eine Relation und $B_1, \dots, B_i \in \{A_1, \dots, A_n\}$. Eine **„Projektion“** von R auf B_1, \dots, B_i , $\text{Proj}(R, [B_1, \dots, B_i])$ oder $\Pi_{B_1, \dots, B_i} R$ oder $\text{PJ}_{B_1, \dots, B_i} R$, ist die Relation, die man erhält, wenn man in R alle Spalten löscht, die nicht in B_1, \dots, B_i vorkommen. Die Reihenfolge der Attribute B ist dabei egal.

Zeige Spalten einer Tabelle, die von Interesse sind.

Bsp.: $R = \text{Proj}(\text{Verkäufer1}, [\text{Käufer}, \text{Produkt}])$

Verkäufer1

Verkäufer	Produkt	Käufer
Meier	Hose	Schmidt
Müller	Rock	Schmidt
Meier	Hose	Schulz

R

Käufer	Produkt
Schmidt	Hose
Schmidt	Rock
Schulz	Hose

Umgang mit doppelten Einträgen. Etwa

$$R = \text{Proj}(\text{Verkäufer1}, [\text{Käufer}])$$

Verkäufer1			R
Verkäufer	Produkt	Käufer	Käufer
Meier	Hose	Schmidt	Schmidt
Müller	Rock	Schmidt	Schmidt
Meier	Hose	Schulz	Schulz

- ▷ **Mathematik**: doppelte herauswerfen.
- ▷ **SQL**: doppelte bleiben drin bzw. `SELECT DISTINCT`. Dann aber teuer!

Ergebnis ist wieder eine Relation/Tabelle.

Selektion

Auswahl **bestimmter** Datensätze.

„Selektion“: Sei R eine Relation, dann bezeichnet $Sel(R, Bed)$ oder $\sigma_{Bed}R$ oder $SL_{Bed}R$ eine Relation, die alle Zeilen von R enthält, die die Bedingung Bed erfüllen.

Einfache Bedingungen werden wie folgt zusammengebaut:

- ▷ $Att <op> \text{Konstante}$ **„Konstantenselektion“**
- ▷ $Att_i <op> Att_j$ **„Attributselektion“**

wobei Datentyp von Att und Konstante kompatibel. Bzw. Datentyp von Att_i und Att_j kompatibel.

Operatoren $<op>$ sind: $=, !=, <, <=, >, >=$. Es gehen auch $=$ und $<>$.

Auswertung auf einem Tupel. Kein Vergleich über mehrere Datensätze.

Bsp.:

$$R = \text{Sel}(\text{Verkäufer1}, \text{Verkäufer} == \text{'Meier'})$$

Verkäufer1

Verkäufer	Produkt	Käufer
Meier	Hose	Schmidt
Müller	Rock	Schmidt
Meier	Hose	Schulz

R

Verkäufer	Produkt	Käufer
Meier	Hose	Schmidt
Meier	Hose	Schulz

Wer hat bei Meier gekauft?

$$\text{Proj}(\text{Sel}(\text{Verkäufer1}, \text{Verkäufer} == \text{'Meier'}), [\text{Käufer}])$$

Die Anfrage

$$\text{Sel}(\text{Verkäufer1}, \text{Verkäufer} == \text{Produkt})$$

liefert eine leere Tabelle.

Bedingungen können zusammengesetzt werden: *„Logische Verknüpfung“*.

$$\text{Bed}_1 \text{ and } \text{Bed}_2, \quad \text{Bed}_1 \text{ or } \text{Bed}_2, \quad \text{not } \text{Bed}$$

Vorsicht! **Dreiwertige Logik** wegen NULL.

and	TRUE	FALSE	NULL	or	TRUE	FALSE	NULL
TRUE	TRUE	FALSE	NULL	TRUE	TRUE	TRUE	TRUE
FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	NULL
NULL	NULL	FALSE	NULL	NULL	TRUE	NULL	NULL

	not
TRUE	FALSE
FALSE	TRUE
NULL	NULL

Was hat Schmidt bei Meier gekauft?

$$\text{Proj}(\text{Sel}(\text{Verkäufer1}, (\text{Verkäufer} == \text{'Meier'}) \text{ AND } (\text{Käufer} == \text{'Schmidt'})), [\text{Produkt}])$$

Verbund

Bisher nur *eine* Relation betrachtet und manipuliert: Projektion, Selektion.

Bzw. mit *vereinigungsverträglichen* Relationen gearbeitet: Vereinigung, Schnitt, Differenz.

Wollen nun strukturell verschiedene Tabellen verknüpfen: *„Verbund“*, *„Join“*.

Bsp.:

Verkäufer1

Verkäufer	Produkt	Käufer
Meier	Hose	Schmidt
Müller	Rock	Schmidt
Meier	Hose	Schulz

Produkte

Produkt	Preis	Klasse
Hose	100	B
Rock	200	A

Kreuzprodukt, Cross Join

„Kreuzprodukt“: Seien R und S zwei Relationen, dann ist das *„Kreuzprodukt“* von R und S definiert über

$$R \times S = \{r \circ s : r \in R \text{ und } s \in S\}.$$

Darin ist $r \circ s = (r_1, \dots, r_n, s_1, \dots, s_m)$ die *„Konkatenation“* von r und s .

Bsp.:

$R = \text{Verkäufer1} \times \text{Produkte}$

Verkäufer1

Verkäufer	Produkt	Käufer
Meier	Hose	Schmidt
Müller	Rock	Schmidt
Meier	Hose	Schulz

Produkte

Produkt	Preis	Klasse
Hose	100	B
Rock	200	A

R

Verkäufer	V1.Produkt	Käufer	P.Produkt	Preis	Klasse
Meier	Hose	Schmidt	Hose	100	B
Meier	Hose	Schmidt	Rock	200	A
Müller	Rock	Schmidt	Hose	100	B
Müller	Rock	Schmidt	Rock	200	A
Meier	Hose	Schulz	Hose	100	B
Meier	Hose	Schulz	Rock	200	A

Viele (sinnlose?) Datensätze. Etwa (Hose, Rock)–Kombinationen.

„Verbund, Join“: Seien R und S zwei Relationen. Dann bezeichnet $Join(R, S, Bed)$ oder $R \bowtie_{Bed} S$ oder $R \Join_{Bed} S$ eine Relation, die alle Tupel in R und S konkateniert, die die Bedingung Bed erfüllen.

Hintereinanderausführung des Kreuzprodukts und einer Selektion und Projektion.

Man unterscheidet noch

- ▷ **„Equi-Join“:** Verwendung von $=$ in der Bed . (Häufigster Fall)
- ▷ **„Natural Join“:** Equi-Join auf *allen* gleichnamigen and-verknüpften Attributen von R und S . Dann kann man Bed weglassen.
- ▷ **„Inner Join“, „Outer Join“** (Machen wir nicht)

Bsp.:

Verkäufer1			Produkte		
Verkäufer	Produkt	Käufer	Produkt	Preis	Klasse
Meier	Hose	Schmidt	Hose	100	B
Müller	Rock	Schmidt	Rock	200	A
Meier	Hose	Schulz			

$Join(Verkäufer1, Produkte, Verkäufer1.Verkäufer==Produkte.Produkt)$

Liefert eine leere Relation.

Bsp.:

Verkäufer1			Produkte		
Verkäufer	Produkt	Käufer	Produkt	Preis	Klasse
Meier	Hose	Schmidt	Hose	100	B
Müller	Rock	Schmidt	Rock	200	A
Meier	Hose	Schulz			

$R = Join(Verkäufer1, Produkte, Verkäufer1.Produkt==Produkte.Produkt)$

$Verkäufer1 \Join_{N} Produkte$ oder $Verkäufer1 \bowtie_{N} Produkte$

R

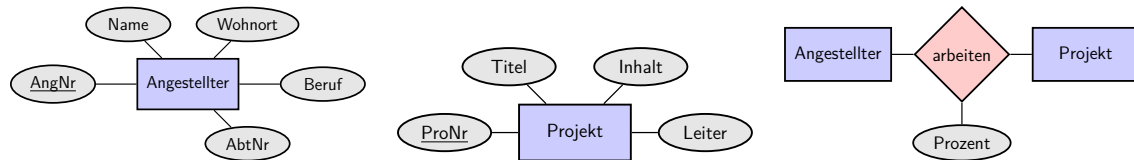
Verkäufer	Produkt	Käufer	Preis	Klasse
Meier	Hose	Schmidt	100	B
Müller	Rock	Schmidt	200	A
Meier	Hose	Schulz	100	B

Ergebnis ist eine Relation/Tabelle.

Komplexer Algebra-Ausdruck

Betrachten komplexeren Algebra-Ausdruck.

ER-Diagramm:



Relationen:

Angestellter = { AngNr : integer, Name : text, Wohnort : text, Beruf : text, AbtNr : integer }

Projekt = { ProNr : integer, Titel : text, Inhalt : text, Leiter : text }

Mitarbeit = { ProNr : integer, AngNr : integer, Prozent : real }

Tabellen:

Angestellter

AngNr	Name	Wohnort	Beruf	AbtNr
112	Müller	Erfurt	Ingenieur	3
205	Winter	Zwickau	Programmierer	3
117	Rüllich	Weimar	Hundezüchter	5
198	Schumann	Jena	Kaufmann	4

Projekt

ProNr	Titel	Inhalt	Leiter
27	Pkw2000	blabla...	205
16	Wankel99	blabla...	117
84	Trabi00	blabla...	117

Mitarbeit

ProNr	AngNr	Prozent
27	112	100
27	198	70
16	198	20

Anfrage: „Finde alle Namen der Angestellten, die im Projekt 27 mitarbeiten“

Umsetzen als Algebra-Ausdruck

Angestellter

AngNr	Name	Wohnort	Beruf	AbtNr
112	Müller	Erfurt	Ingenieur	3
205	Winter	Zwickau	Programmierer	3
117	Rüllich	Weimar	Hundezüchter	5
198	Schumann	Jena	Kaufmann	4

Projekt

ProNr	Titel	Inhalt	Leiter
27	Pkw2000	blabla...	205
16	Wankel99	blabla...	117
84	Trabi00	blabla...	117

Mitarbeit

ProNr	AngNr	Prozent
27	112	100
27	198	70
16	198	20

- ▷ $PJ_{(Name)}(Angestellter \Join_{AngNr=AngNr} (SL_{ProNr=27} \text{Mitarbeit}))$
- ▷ $PJ_{(Name)}(Angestellter \Join_{AngNr=AngNr} (PJ_{(AngNr)} (SL_{ProNr=27} \text{Mitarbeit})))$
- ▷ $PJ_{(Name)}(SL_{ProNr=27}(Angestellter \Join_{AngNr=AngNr} \text{Mitarbeit}))$
- ▷ $SL_{ProNr=27}(PJ_{(Name)}(Angestellter \Join_{AngNr=AngNr} \text{Mitarbeit}))$

Vorsicht! Letzter Ausdruck syntaktisch falsch.

SQL — Teil 2

SELECT
 Projektion
 Selektion
 Vereinigung, Schnitt, Differenz
 Verbund
 Komplexer SELECT-Ausdruck

Fahren fort mit SQL Befehlen. Bilden Relationenalgebra auf SQL ab.

So Umsetzung von Anfragen an die DB (bzw. Tabellen) möglich.