

Kapitel 5: Relationenmodell und algebraorientierte Anfragesprachen

5.1 Grundbegriffe

Das Relationenmodell beruht auf Arbeiten von E.F. Codd und anderen um 1970, für die Codd 1981 mit dem Turing Award ausgezeichnet wurde.

Grundprinzip:

Alle Daten werden in Form von mathematischen Relationen (Teilmengen eines kartesischen Produkts von Wertemengen) repräsentiert.

Definition:

Gegeben sei eine Menge von Wertebereichen primitiver Datentypen $\{D_1, \dots, D_m\}$, die als "Domains" bezeichnet werden.

Eine *Relation* R ist ein Paar $R = (s, v)$ mit

- einem *Schema* $s = \{A_1, \dots, A_n\}$, das aus einer Menge von *Attributen* (genauer: Attributnamen) besteht und für jedes Attribut A_i einen Domain $\text{dom}(A_i) \in \{D_1, \dots, D_m\}$ festlegt, und
- einer *Ausprägung* (auch *Wert* genannt) $v \subseteq \text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n)$.

Schema und Ausprägung von R werden auch mit $\text{sch}(R)$ und $\text{val}(R)$ bezeichnet.

Bemerkungen:

- Die Elemente der Ausprägung einer Relation heißen *Tupel*. Für den Wert eines Tupels t bezüglich eines Attributs A_i schreibt man $t.A_i$ (seltener auch $t[A_i]$ oder $A_i(t)$).
- Die Tupel einer Relation bilden eine (ungeordnete) Menge. Es gibt also keine Reihenfolge unter den Tupeln, und je zwei Tupel müssen sich in mindestens einem Attribut unterscheiden.
- Die Attribute einer Relation bilden eine (ungeordnete) Menge; es gibt also keine Reihenfolge unter den Attributen.
- Als Domains von Attributen sind nur primitive Wertebereiche zugelassen, d.h. Wertebereiche elementarer Datentypen (im wesentlichen INTEGER, REAL, BOOLEAN plus Spezialtypen wie DATE, MONEY, etc.), die Wertebereiche von Bereichstypen und Aufzählungstypen sowie der Wertebereich der Zeichenketten (ARRAY OF CHAR). Diese Einschränkung wird auch als *1. Normalform* des Relationenmodells bezeichnet.
- Eine relationale Datenbank ist eine Menge von Relationen. Die Menge der betreffenden Relationenschemata wird als Datenbankschema bezeichnet.

Häufige Schreibweise für Relationenschemata:

$R(A_1, \dots, A_n)$

Kunden (KNr, Name, Stadt, Saldo, Rabatt)

Produkte (PNr, Bez, Gewicht, Preis, Lagerort, Vorrat)

Bestellungen (BestNr, Monat, Tag, KNr, PNr, Menge, Summe, Status)

Informelle Sprechweise:

Relation = Tabelle

Schema = Tabellenkopf (Überschrift der Tabelle)

Attribut = Spalte der Tabelle (engl.: column, field; Feld)

Tupel = Zeile der Tabelle (engl.: row, record; Datensatz)

Beispieldatenbank:

Kunden

KNr	Name	Stadt	Saldo	Rabatt
1	Lauer	Merzig	- 1080.00	0.10
2	Schneider	Homburg	- 800.00	0.20
3	Kirsch	Homburg	0.00	0.10
4	Schulz	Merzig	0.00	0.10
5	Becker	Dillingen	0.00	0.05
6	Meier	Saarlouis	- 3800.00	0.05

Produkte

PNr	Bez	Gewicht	Preis	Lagerort	Vorrat
1	Papier	2.000	20.00	Homburg	10000
2	Platte	1.000	2500.00	Saarbrücken	400
3	Drucker	5.000	2000.00	Merzig	200
4	Bildschirm	5.000	3000.00	Merzig	80
5	Disketten	0.500	20.00	Homburg	5000
6	Maus	0.250	100.00	Homburg	200
7	Speicher	0.100	200.00	Saarbrücken	2000

Bestellungen

BestNr	Monat	Tag	KNr	PNr	Menge	Summe	Status
1	7	16	1	1	100	1800.00	bezahlt
2	7	21	1	1	100	1800.00	bezahlt
3	9	30	1	2	4	9000.00	bezahlt
4	9	30	1	3	1	1800.00	bezahlt
5	9	30	1	4	10	27000.00	bezahlt
6	10	15	1	5	50	900.00	bezahlt
7	10	28	1	6	2	180.00	geliefert
8	11	2	1	7	5	900.00	neu
9	10	26	2	1	100	1600.00	bezahlt
10	11	2	2	5	50	800.00	neu
11	9	28	3	5	50	900.00	bezahlt
12	10	28	3	7	10	1800.00	bezahlt
13	4	15	4	1	50	900.00	bezahlt
14	5	31	6	1	200	3800.00	bezahlt
15	6	30	6	7	10	1900.00	geliefert
16	7	31	6	1	100	1900.00	geliefert

Konsequenz der 1. Normalform

Studenten

Name	Fach	...	Systemkenntnisse
Meier	Informatik		{Oracle, mySQL, PHP}
Schmidt	Informatik		{Java, Oracle}
Kunz	Informatik		{Oracle}
Müller	Mathematik		∅

ist nicht zulässig!

Korrekte Repräsentation im Relationenmodell:

Studenten

Name	Fach	...	Systemkenntnis
Meier	Informatik		Oracle
Meier	Informatik		mySQL
Meier	Informatik		PHP
Schmidt	Informatik		Java
Schmidt	Informatik		Oracle
Kunz	Informatik		Oracle
Müller	Mathematik		---

oder besser:

Studenten

Name	Fachbereich	...
Meier	Informatik	
Schmidt	Informatik	
Kunz	Informatik	
Müller	Mathematik	

Kenntnisse

Name	Systemkenntnis
Meier	Oracle
Meier	mySQL
Meier	PHP
Schmidt	Java
Schmidt	Oracle
Kunz	Oracle

Inhärente Integritätsbedingungen des Relationenmodells

Definitionen:

Sei R eine Relation. Eine Attributmenge $K \subseteq \text{sch}(R)$ heißt *Schlüsselkandidat*, wenn zu jedem Zeitpunkt für je zwei Tupel $t_1, t_2 \in \text{val}(R)$ gelten muß:

$$t_1.K = t_2.K \Rightarrow t_1 = t_2$$

und wenn es keine echte Teilmenge von K gibt, die diese Eigenschaft hat.

(Dabei bedeutet $t_1.K = t_2.K$ präzise: $\forall A \in K: t_1.A = t_2.A$.)

Bemerkung: Jede Relation hat mindestens einen Schlüsselkandidaten.

Ein Attribut einer Relation R , das in mindestens einem Schlüsselkandidaten vorkommt, heißt *Schlüsselattribut*.

Der *Primärschlüssel* (engl.: primary key) einer Relation ist ein Schlüsselkandidat, der explizit ausgewählt wird.

Eine Attributmenge $F \subseteq \text{sch}(S)$ einer Relation S ist ein *Fremdschlüssel* (engl.: foreign key) in S , wenn es eine Relation R gibt, in der F Primärschlüssel ist.

Ein Attribut A eines Tupels t hat einen *Nullwert*, wenn der Wert $t.A$ undefiniert oder unbekannt ist. (Gedanklich werden alle Domains um einen solchen Nullwert erweitert.)

Beispiele:

- Vorrat für Produkte, die nicht gelagert werden, sondern bei Bedarf extern gekauft werden.
- Note von Studenten, die ihre Prüfung noch nicht abgelegt haben.
- Informatikkenntnisse von Studenten zu Beginn des Studiums.

Schreibweise:

In einem Relationenschema wird der Primärschlüssel unterstrichen; weitere Schlüsselkandidaten werden oft gestrichelt unterstrichen.

Das Relationenmodell (genauer: eine Implementierung des Relationenmodells) garantiert die folgenden Integritätsbedingungen:

Primärschlüsselbedingung ("Entity Integrity"):

Für jede Relation muß ein Primärschlüssel festgelegt werden. Der Primärschlüssel eines Tupels darf nie den Nullwert annehmen (auf keinem einzigen der zum Primärschlüssel gehörigen Attribute).

Beispiel:

Kunden (KNr, Name, Stadt, Saldo, Rabatt)

Produkte (PNr, Bez, Gewicht, Preis, Lagerort, Vorrat)

Bestellungen (BestNr, Monat, Tag, KNr, PNr, Menge, Summe, Status)

Fremdschlüsselbedingung ("Referential Integrity"):

Für jeden Wert eines Fremdschlüssels in einer Relation R muß in den referenzierten Relationen jeweils ein Tupel mit demselben Wert als Primärschlüssel existieren, oder der Wert des Fremdschlüssels muß der Nullwert sein.

Beispiel für Nullwert als Fremdschlüssel:

Produkte (PNr, ..., Lagerort)

Lager (Lagerort, Adresse, Verwalter, ...)

5.2 Relationalalgebra

Die Relationalalgebra bildet ein Fundament für Datenbankanfragesprachen (Query Languages), indem sie Grundoperationen für Relationen definiert.

- Eine Operation hat eine oder mehrere Relationen als Operanden und stets wiederum eine Relation als Ergebnis.
→ Abgeschlossenheit der Relationalalgebra.
- Das Schema eines Operationsresultats kann von den Schemata der Operanden verschieden sein.

Mengenoperationen:

Für zwei Relationen R, S mit $\text{sch}(R) = \text{sch}(S)$ sind die üblichen Mengenoperationen definiert:

- Vereinigung (Union) $R \cup S$:
 $\text{sch}(R \cup S) = \text{sch}(R)$
 $\text{val}(R \cup S) = \{t \mid t \in \text{val}(R) \vee t \in \text{val}(S)\}$
- Durchschnitt (Intersection) $R \cap S$:
 $\text{sch}(R \cap S) = \text{sch}(R)$
 $\text{val}(R \cap S) = \{t \mid t \in \text{val}(R) \wedge t \in \text{val}(S)\}$
- Differenz (Difference) $R - S$:
 $\text{sch}(R - S) = \text{sch}(R)$
 $\text{val}(R - S) = \{t \mid t \in \text{val}(R) \wedge t \notin \text{val}(S)\}$

Selektion σ (Filterung, Auswahl von Zeilen einer Tabelle):

Sei F eine Boolesche Formel über einfachen Vergleichsbedingungen zwischen zwei Attributen einer Relation oder einem Attribut und einer Konstanten. Das Resultat einer Selektion $\sigma[F](R)$ auf einer Relation R (auch $\sigma_F(R)$ geschrieben) ist wie folgt definiert:

$$\text{sch}(\sigma[F](R)) = \text{sch}(R)$$
$$\text{val}(\sigma[F](R)) = \{t \mid t \in R \wedge F(t)\} \text{ wobei } F(t) \text{ bedeutet, daß } t \text{ die Bedingung } F \text{ erfüllt.}$$

Die Menge der möglichen Filterformeln F ist wie folgt präzise definiert:

- 1) Für Attribute A, B von R mit $\text{dom}(A) = \text{dom}(B)$, Konstanten $c \in \text{dom}(A)$ und Vergleichsoperationen $\theta \in \{=, \neq, <, >, \leq, \geq\}$ sind $A \theta B$ und $A \theta c$ zulässige Filterbedingungen.
- 2) Falls F_1 und F_2 zulässige Filterbedingungen sind, dann sind auch $F_1 \wedge F_2$, $F_1 \vee F_2$, $\neg F_1$ und (F_1) zulässig.
- 3) Nur die aufgrund von 1) und 2) erzeugten Filterbedingungen sind zulässig.

Projektion π (Auswahl von Spalten einer Tabelle):

Sei $A \subseteq \text{sch}(R)$ eine Teilmenge der Attribute einer Relation R . Das Resultat einer Projektion $\pi[A](R)$ auf der Relation R (auch $\pi_A(R)$ geschrieben) ist wie folgt definiert:

$$\text{sch}(\pi[A](R)) = A$$
$$\text{val}(\pi[A](R)) = \{t \mid \exists r \in \text{val}(R): t.A = r.A\} \text{ bzw. ausführlicher}$$
$$\text{val}(\pi[A](R)) = \{t \mid \exists r \in \text{val}(R): t.A_1 = r.A_1 \wedge \dots \wedge t.A_n = r.A_n \text{ für } A = \{A_1, \dots, A_n\}\}$$

Achtung: Die Projektion beinhaltet eine Duplikateliminierung.

Beispiele für die Selektion:

- 1) Finde alle Homburger Kunden.
→ $\sigma[\text{Stadt}='Homburg']$ (Kunden)
→ Resultat:

KNr	Name	Stadt	Saldo	Rabatt
2	Schneider	Homburg	- 800.00	0.20
3	Kirsch	Homburg	0.00	0.10

- 2) Finde alle Homburger Kunden, die einen Rabatt von mindestens 15 % haben.
→ $\sigma[\text{Stadt}='Homburg' \wedge \text{Rabatt} \geq 0.15]$ (Kunden)
→ Resultat:

KNr	Name	Stadt	Saldo	Rabatt
2	Schneider	Homburg	- 800.00	0.20

Beispiele für die Projektion:

- 3) Gib alle Produktbezeichnungen aus.
→ $\pi[\text{Bez}]$ (Produkte)
→ Resultat:

Bez
Papier
Platte
Drucker
Bildschirm
Disketten
Maus
Speicher

- 4) Gib alle Lagerorte von Produkten aus.
→ $\pi[\text{Lagerort}]$ (Produkte)
→ Resultat:

Lagerort
Homburg
Saarbrücken
Merzig

(Natural) Join \bowtie

(Natürlicher Verbund, Verbindung zweier Relationen über gleiche Attributnamen und gleiche Attributwerte der Tupel):

Seien R, S Relationen mit Schemata $A = \text{sch}(R)$ und $B = \text{sch}(S)$. Das Resultat des Joins

$R \bowtie S$ ist wie folgt definiert:

$$\text{sch}(R \bowtie S) = \text{sch}(R) \cup \text{sch}(S)$$

$$\text{val}(R \bowtie S) = \{t \mid \exists r \in \text{val}(R) \exists s \in \text{val}(S): t.A = r.A \wedge t.B = s.B\}$$

Ein Sonderfall des Joins, nämlich für $\text{sch}(R) \cap \text{sch}(S) = \emptyset$, erhält man das kartesische Produkt $R \times S$.

Beispiele für den Join:

1) Führe alle Bestellungen zusammen mit den dazugehörigen Produktdaten auf.

→ Bestellungen \bowtie Produkt

→ Resultat:

Best Nr	Monat	Tag	KNr	PNr	Menge	Summe	Status	Bez	Gewicht	Preis	Lagerort	Vorrat
1	7	16	1	1	100	1800.00	bezahlt	Papier	2.000	20.00	Homburg	10000
2	7	21	1	1	100	1800.00	bezahlt	Papier	2.000	20.00	Homburg	10000
3	9	30	1	2	4	9000.00	bezahlt	Platte	1.000	2500.00	Saar-brücken	400
.												
.												
.												

2) $R \bowtie S$ mit:

R

S

R1	J
a	1
b	2
c	2
d	3

S1	J
w	2
x	2
y	3
z	4

→ Resultat:

R1	J	S1
b	2	w
b	2	x
c	2	w
c	2	x
d	3	y

Achtung: Im allgemeinen kann $\pi[\text{sch}(R)](R \bowtie S) \neq R$ sein.

Zuweisung:

Seien R und S zwei Relationen mit $\text{sch}(R)=\{A_1, \dots, A_n\}$ und $\text{sch}(S)=\{B_1, \dots, B_n\}$, so daß für alle i gilt: $\text{dom}(A_i)=\text{dom}(B_i)$. Die Zuweisung $R := S$ bedeutet, daß sich die Ausprägung von R wie folgt ändert: $\text{val}(R) = \text{val}(S)$. Ausführlicher schreibt man ggf. auch $R(A_1, \dots, A_n) := S(B_1, \dots, B_n)$. Allgemeiner kann man für Ausdrücke E_1, \dots, E_n , die über B_1, \dots, B_n und k -stelligen skalaren Operatoren $\psi: \text{dom}(B_{i1}) \times \dots \times \text{dom}(B_{ik}) \rightarrow D$ mit skalarem Resultat im Wertebereich W Zuweisungen der Form $R(A_1, \dots, A_n) := S(E_1, \dots, E_n)$ vornehmen, wenn der Wertebereich von E_i mit $\text{dom}(A_i)$ übereinstimmt. Dabei ist $\text{val}(R) = \{t \mid \text{es gibt } s \in \text{val}(S) \text{ und } t.A_i = E_i(s) \text{ für alle } i\}$. Die Zuweisung ist nur eine Hilfsoperation der Relationenalgebra. Sie ist nützlich als Notation für Zwischenresultate, für Umbenennungen von Attributnamen und um Änderungsoperationen ausdrücken zu können.

Beispiele für komplexere Anfragen:

- 1) Finden Sie die Namen der Kunden mit negativem Saldo.
 $\rightarrow \pi[\text{Name}] (\sigma[\text{Saldo} < 0.0] (\text{Kunden}))$
- 2) Finden Sie die Namen der Kunden, die eine unbezahlte Bestellung haben, die vor Anfang Oktober erfolgte.
 $\rightarrow \pi[\text{Name}] (\sigma[\text{Status} \neq \text{'bezahlt'} \wedge \text{Monat} < 10] (\text{Bestellungen}) \mid \times \mid \text{Kunden})$
oder mit Zwischenresultaten:
 $B := \sigma[\text{Status} \neq \text{'bezahlt'} \wedge \text{Monat} < 10] (\text{Bestellungen})$
 $BK := B \mid \times \mid \text{Kunden}$
 $\pi[\text{Name}] (BK)$
- 3) Finden Sie die Namen der Homburger Kunden, die seit Anfang September ein Produkt aus Homburg geliefert bekommen haben.
 $\rightarrow \pi[\text{Name}] (\sigma[\text{Monat} \geq 9 \wedge \text{Status} \neq \text{'neu'}] (\text{Bestellungen})$
 $\quad \mid \times \mid \sigma[\text{Lagerort} = \text{'Homburg'}] (\text{Produkte})$
 $\quad \mid \times \mid \sigma[\text{Stadt} = \text{'Homburg'}] (\text{Kunden}))$
oder mit Zwischenresultaten:
 $B := \sigma[\text{Monat} \geq 9 \wedge \text{Status} \neq \text{'neu'}] (\text{Bestellungen})$
 $P := \sigma[\text{Lagerort} = \text{'Homburg'}] (\text{Produkte})$
 $K := \sigma[\text{Stadt} = \text{'Homburg'}] (\text{Kunden})$
 $BP := B \mid \times \mid P$
 $BPK := BP \mid \times \mid K$
 $\pi[\text{Name}] (BPK)$
- 4) Finden Sie (alle Attribute bzw. die Kundennummern der) Kunden, von denen mindestens eine Bestellung registriert ist.
 $\rightarrow \pi[\text{sch}(\text{Kunden})] (\text{Kunden} \mid \times \mid \text{Bestellungen})$ bzw.
 $\pi[\text{KNr}] (\text{Kunden} \mid \times \mid \text{Bestellungen})$ oder $\pi[\text{KNr}] (\text{Bestellungen})$
- 5) Finden Sie die Kundennummern der Kunden, von denen keine Bestellung registriert ist.
 $\rightarrow \pi[\text{KNr}] (\text{Kunden}) - \pi[\text{KNr}] (\text{Kunden} \mid \times \mid \text{Bestellungen})$
oder:
 $\pi[\text{KNr}] (\text{Kunden}) - \pi[\text{KNr}] (\text{Bestellungen})$
- 6) Finden Sie die Namen der Kunden, von denen keine Bestellung registriert ist.
 $\rightarrow \pi[\text{Name}] (\pi[\text{KNr}, \text{Name}] (\text{Kunden}) - \pi[\text{KNr}, \text{Name}] (\text{Kunden} \mid \times \mid \text{Bestellungen}))$
oder:
 $\pi[\text{Name}] (\text{Kunden} \mid \times \mid (\pi[\text{KNr}] (\text{Kunden}) - \pi[\text{KNr}] (\text{Bestellungen})))$

Division \div :

Seien R, S Relationen mit Schemata $A = \text{sch}(R)$ und $B = \text{sch}(S)$, so daß $B \subset A$. Mit $Q = A - B$ sei die Menge der Attribute bezeichnet, die in R vorkommen, nicht aber in S. Das Resultat der Division $R \div S$ ist wie folgt definiert:

$$\text{sch}(R \div S) = A - B$$

$$\text{val}(R \div S) = \{t \mid \forall s \in \text{val}(S) \exists r \in \text{val}(R): r.B = s.B \wedge t = r.Q\}$$

Intuitive Bedeutung:

Ein Tupel t ist in $R \div S$ genau dann, wenn für alle S-Tupel s ein zusammengesetztes Tupel $\langle t, s \rangle$ in R enthalten ist.

Bemerkungen:

- Die Division ist die einfachste Art, Anfragen der Form "... für alle ..." auszudrücken.
- Ein Sonderfall der Division ist $R \div \emptyset = \pi[\text{sch}(R) - \text{sch}(\emptyset)](R)$.

Beispiel für die Division:

Finden Sie die Kundennummern derjenigen Kunden, die alle überhaupt lieferbaren Produkte irgendwann bestellt haben.

→ $\pi[\text{KNr}, \text{PNr}]$ (Bestellungen) \div $\pi[\text{PNr}]$ (Produkte)

$\pi[\text{KNr}, \text{PNr}]$ (Bestellungen)

$\pi[\text{PNr}]$ (Produkte)

KNr	PNr
1	1
1	2
1	3
1	4
1	5
1	6
1	7
2	1
2	5
3	5
3	7
4	1
6	1
6	7

PNr
1
2
3
4
5
6
7

→ Resultat:

KNr
1

Satz:

Seien $R(A,B,C)$, $T(A,B)$ und $S(C)$ Relationen, so daß $R = T \times S$. Dann gilt: $T = R \div S$.

Kartesisches Produkt \times :

Seien R, S zwei Relationen mit Schemata $A = \text{sch}(R)$ und $B = \text{sch}(S)$. Sei A' ein Schema, bei dem alle Attributnamen A_i , die auch in $\text{sch}(S)$ vorkommen, unbenannt sind in $R.A_i$, und sei B' ein Schema, bei dem alle Attributnamen A_i , die auch in $\text{sch}(R)$ vorkommen, umbenannt sind in $S.A_i$. Das Resultat des kartesischen Produkts $R \times S$ ist wie folgt definiert:

$$\text{sch}(R \times S) = A' \cup B'$$

$$\text{val}(R \times S) = \{t \mid \exists r \in \text{val}(R) \exists s \in \text{val}(S): t.A' = r.A \text{ und } t.B' = s.B\}$$

Intuitive Bedeutung:

$R \times S$ enthält alle möglichen Kombinationen von R -Tupeln und S -Tupeln.

Beispiel für das kartesische Produkt:

$R \times S$ mit:

R

S

R1	J
a	1
b	2
c	2

S1	J
w	2
x	2

→ Resultat:

R1	R.J	S1	S.J
a	1	w	2
b	2	w	2
c	2	w	2
a	1	x	2
b	2	x	2
c	2	x	2

Rückführung des Join auf das kartesische Produkt:

Seien R und S zwei Relationen mit Schemata $A = \text{sch}(R)$ und $B = \text{sch}(S)$, und sei $J = \text{sch}(R) \cap \text{sch}(S)$. Es gilt (bis auf Umbenennungen von Attributen):

$$R \bowtie S = \pi[A \cup B - \{S.J\}] (\sigma[R.J=S.J] (R \times S))$$

Rückführung der Division auf das kartesische Produkt:

Satz:

Seien R und S zwei Relationen mit Schemata $A = \text{sch}(R)$ und $B = \text{sch}(S)$ mit $A \supset B$, und sei $Q = \text{sch}(R) - \text{sch}(S)$. Es gilt (bis auf Umbenennungen von Attributen):

$$R \div S = \pi[Q](R) - \pi[Q] ((\pi[Q](R) \times S) - R)$$

Beispiel:

$R := \pi[\text{KNr}, \text{PNr}]$ (Bestellungen)

$S := \pi[\text{PNr}]$ (Produkte)

$T1 := \pi[\text{KNr}](R) \times S$

alle überhaupt möglichen Bestellungen

$T2 := T1 - R$

potentiell mögliche, aber nicht erfolgte Bestellungen

$T3 := \pi[\text{KNr}](T2)$

Kunden, die nicht alle Produkte bestellt haben

$T4 := \pi[\text{KNr}](R) - T3$

Kunden, die alle Produkte bestellt haben

θ -Join:

Seien R, S zwei Relationen mit Schemata $\text{sch}(R)$ und $\text{sch}(S)$, so daß $\text{sch}(R) \cap \text{sch}(S) = \emptyset$. Seien $A \subseteq \text{sch}(R)$ und $B \subseteq \text{sch}(S)$. Sei ferner θ eine der Vergleichsoperationen $=, \neq, <, >, \leq, \geq$. Das Resultat des θ -Joins $R \times |[A \theta B] S$ (auch $R \times |_{A\theta B} S$ geschrieben) ist wie folgt definiert:

$$\text{sch}(R \times |[A \theta B] S) = \text{sch}(R \times S) = \text{sch}(R) \cup \text{sch}(S)$$

$$\text{val}(R \times |[A \theta B] S) = \text{val}(\sigma[A \theta B](R \times S))$$

Wenn θ die Vergleichsoperation $=$ ist, dann heißt der θ -Join auch Equijoin.

In analoger Weise läßt sich der θ -Join zu einem Join mit allgemeiner Boolescher Filterformel erweitern, wobei dieselben Formeln wie bei der Selektion zulässig sind.

Beispiele für den θ -Join:

1) Finde alle Kunden, die an einem Lagerort wohnen.

→ $\pi[\text{KNr}, \text{Name}, \text{Stadt}, \dots] (\text{Kunden} \times |_{[\text{Stadt}=\text{Lagerort}]} \text{Produkte})$

2) Finde alle bisherigen Bestellungen, die den momentanen Vorrat erschöpfen würden.

→ $B(\text{BestNr}, \text{Monat}, \text{Tag}, \text{KNr}, \text{B.PNr}, \dots) := \text{Bestellungen}(\text{BestNr}, \text{Monat}, \text{Tag}, \text{KNr}, \text{PNr}, \dots)$

$\pi[\text{BestNr}, \dots] (B \times |[B.\text{PNr}=\text{PNr} \wedge \text{Menge} \geq \text{Vorrat}] \text{Produkte})$

3) Finde alle Paare von Kunden, die in derselben Stadt wohnen.

→ $K1(\text{K1.KNr}, \text{K1.Name}, \text{K1.Stadt}, \dots) := \text{Kunden}(\text{KNr}, \text{Name}, \text{Stadt}, \dots)$

$K2(\text{K2.KNr}, \text{K2.Name}, \text{K2.Stadt}, \dots) := \text{Kunden}(\text{KNr}, \text{Name}, \text{Stadt}, \dots)$

$\pi[\text{K1.KNr}, \text{K2.KNr}] (K1 \times |[K1.\text{Stadt} = K2.\text{Stadt}] K2)$

bzw. besser:

$\pi[\text{K1.KNr}, \text{K2.KNr}] (K1 \times |[K1.\text{Stadt} = K2.\text{Stadt} \wedge K1.\text{KNr} < K2.\text{KNr}] K2)$

um jedes (echte) Kundenpaar nur einmal aufzulisten

saloppere Schreibweise häufig:

$\pi[\text{KNr1}, \text{KNr2}] (\text{Kunden} \times |_{[\text{Stadt1} = \text{Stadt2} \wedge \text{KNr1} < \text{KNr2}]} \text{Kunden})$

Semi-Join und Anti-Join:

Seien R, S zwei Relationen mit Schemata $\text{sch}(R)$ und $\text{sch}(S)$, so daß $\text{sch}(R) \cap \text{sch}(S) = J$.

Das Resultat des Semi-Joins $R \times S$ und das Resultat des Anti-Joins $R \mid- S$ sind wie folgt definiert:

$$\text{sch}(R \times S) = \text{sch}(R)$$

$$\text{val}(R \times S) = \{r \mid r \in \text{val}(R) \wedge \exists s \in \text{val}(S): r.J = s.J\}$$

$$\text{sch}(R \mid- S) = \text{sch}(R)$$

$$\text{val}(R \mid- S) = \{r \mid r \in \text{val}(R) \wedge \neg (\exists s \in \text{val}(S): r.J = s.J)\}$$

Satz:

$R \times S = \pi[R] (R \times | S)$ und

$R \mid- S = R - \pi[R] (R \times | S)$

Outer Join

Seien R und S zwei Relationen mit Schemata $A = \text{sch}(R)$ und $B = \text{sch}(S)$, und sei $J = \text{sch}(R) \cap \text{sch}(S)$.

Bezeichne ferner ω den Nullwert.

Das Resultat des Outer Joins $R \mid^* S$ ist wie folgt definiert:

$$\text{sch}(R \mid^* S) = \text{sch}(R) \cup \text{sch}(S)$$

$$\begin{aligned} \text{val}(R \mid^* S) = & \text{val}(R \mid \times S) \cup \\ & \{t \mid \exists r \in \text{val}(R) : t.A=r.A \wedge \neg(\exists s \in \text{val}(S) : t.J=s.J) \wedge t.(B-J)=\omega\} \cup \\ & \{t \mid \exists s \in \text{val}(S) : t.B=s.B \wedge \neg(\exists r \in \text{val}(R) : t.J=r.J) \wedge t.(A-J)=\omega\}. \end{aligned}$$

Beispiel für den Outer Join:

Gib alle Kunden mit 5 % Rabatt zusammen mit ihren Bestellungen aus, und zwar auch, wenn für einen Kunden gar keine Bestellungen vorliegen.

$\sigma[\text{Rabatt} = 0.05](\text{Kunden} \mid^* \text{Bestellungen})$

KNr	Name	Stadt	Saldo	Rabatt	BestNr	...
5	Becker	Dillingen	0.00	0.05	ω	...
6	Meier	Saarlouis	-3800.00	0.05	14	...
6	Meier	Saarlouis	-3800.00	0.05	15	...
6	Meier	Saarlouis	-3800.00	0.05	16	...

Rückführung des Outer Join auf Join und kartesisches Produkt:

Seien NR und NS Relationen mit $\text{sch}(NR) = \text{sch}(R) - \text{sch}(S)$ und $\text{sch}(NS) = \text{sch}(S) - \text{sch}(R)$, die jeweils genau ein Tupel enthalten, dessen Attribute alle ω als Wert haben.

$$\begin{aligned} \text{Dann ist: } R \mid^* S = & (R \mid \times S) \cup \\ & ((R - \pi[\text{sch}(R)](R \mid \times S)) \times NS) \cup \\ & (NR \times (S - \pi[\text{sch}(S)](R \mid \times S))) \end{aligned}$$

Left Outer Join:

$$\text{sch}(R \parallel^* S) = \text{sch}(R) \cup \text{sch}(S)$$

$$\begin{aligned} \text{val}(R \parallel^* S) = & \text{val}(R \mid \times S) \cup \\ & \{t \mid \exists r \in \text{val}(R) : t.A=r.A \wedge \neg(\exists s \in \text{val}(S) : t.J=s.J) \wedge t.(B-J)=\omega\}. \end{aligned}$$

Right Outer Join:

$$\text{sch}(R \mid^* S) = \text{sch}(R) \cup \text{sch}(S)$$

$$\begin{aligned} \text{val}(R \mid^* S) = & \text{val}(R \mid \times S) \cup \\ & \{t \mid \exists s \in \text{val}(S) : t.B=s.B \wedge \neg(\exists r \in \text{val}(R) : t.J=r.J) \wedge t.(A-J)=\omega\}. \end{aligned}$$

Äquivalenzregeln, Minimalität und Vollständigkeit der Relationenalgebra

Seien R, S, T Relationen, P, P_1, P_2 Prädikate, R_1, R_2, S_1 Teilmengen der Attribute von R bzw. S . Es gelten u.a. die folgenden Äquivalenzregeln:

Kommutativitätsregeln:

- 1) $\pi[R_1] \sigma[P] (R) = \sigma[P] \pi[R_1] (R)$
falls P nur R_1 -Attribute enthält
- 2) $R \mid x \mid S = S \mid x \mid R$

Assoziativitätsregeln:

- 3) $R \mid x \mid (S \mid x \mid T) = (R \mid x \mid S) \mid x \mid T$

Idempotenzregeln:

- 4) $\pi[R_1] (\pi[R_2] (R)) = \pi[R_1] (R)$
falls $R_1 \subseteq R_2$
- 5) $\sigma[P_1] (\sigma[P_2] (R)) = \sigma[P_1 \wedge P_2] (R)$

Distributivitätsregeln:

- 6) $\pi[R_1] (R \cup S) = \pi[R_1](R) \cup \pi[R_1](S)$
- 7) $\sigma[P] (R \cup S) = \sigma[P](R) \cup \sigma[P](S)$
- 8) $\sigma[P] (R \mid x \mid S) = \sigma[P](R) \mid x \mid S$
falls P nur R -Attribute enthält
- 9) $\pi[R_1, S_1] (R \mid x \mid S) = \pi[R_1](R) \mid x \mid \pi[S_1](S)$
falls Joinattribute $\subseteq R_1 \cup S_1$
- 10) $R \mid x \mid (S \cup T) = (R \mid x \mid S) \cup (R \mid x \mid T)$

Invertierungsregeln:

- 11) $\pi[\text{sch}(R)] (R \parallel * \mid S) = R$

Diese und weitere Äquivalenzregeln bilden die Grundlage für eine automatische Optimierung von deklarativen Anfragen in Form von Ausdrücken der Relationenalgebra.

Definition:

Die Menge der relationalalgebraischen Ausdrücke über einer Menge von Relationen R_1, \dots, R_n ist wie folgt definiert:

- (i) R_1, \dots, R_n sind Ausdrücke.
- (ii) Wenn R, S, T, Q Ausdrücke sind,
F eine Filterformel über dem Resultatschema von R ist,
A eine Teilmenge des Resultatschemas von R ist,
S und T dasselbe Resultatschema haben und $\text{sch}(R) \supseteq \text{sch}(Q)$ gilt,
dann sind
 $\sigma[F](R), \pi[A](R), R \bowtie S, R \times S, R \Join S, S \cap T, S \cup T, S - T, R \div Q$
auch Ausdrücke.
- (iii) Nur die gemäß (i) und (ii) erzeugten Ausdrücke sind relationalalgebraische Ausdrücke.

Satz:

$\times, \pi, \sigma, \cup$ und $-$ bilden eine minimale Menge von Operationen, mit denen sich alle Operationen der Relationalalgebra ausdrücken lassen.

Beweis: siehe Vorlesung

Definition:

Eine Anfragesprache heißt *relational vollständig*, wenn sich damit alle Anfragen der (minimalen) Relationalalgebra ausdrücken lassen.

5.3 Erweiterte Relationenalgebra

Die "reine" Relationenalgebra ist auf Mengen von Tupeln definiert. In der Einsatzpraxis von Datenbanken werden aber auch Anfragen auf *Multimengen* (und u.U. auch - sortierten - Listen) benötigt.

Definition:

Eine *Multimenge* (engl. multiset, bag) M über einer Grundmenge G ist eine Abbildung $M: G \rightarrow \mathbb{N}_0$ in die natürlichen Zahlen $\mathbb{N}_0 = \{0, 1, 2, \dots\}$.

$M(x)$ wird als die Häufigkeit von x in M bezeichnet oder auch als die "Anzahl der Duplikate von x ". Multimengen werden häufig in einer Pseudomengennotation geschrieben, bei der die Häufigkeit eines Elements durch entsprechend wiederholtes Aufführen des Elements angegeben wird. Beispielsweise ist $M = \{\clubsuit, \clubsuit, \heartsuit, \spadesuit, \spadesuit, \spadesuit\}$ eine Multimenge über $G = \{\clubsuit, \spadesuit, \heartsuit, \diamondsuit\}$ mit $M(\clubsuit)=2$, $M(\heartsuit)=1$, $M(\spadesuit)=3$ und $M(x)=0$ für alle anderen $x \in G$.

Für Multimengen M und M' über derselben Grundmenge G gilt $M \subseteq M'$ genau dann, wenn für alle $x \in G$: $M(x) \leq M'(x)$.

Definition:

Eine *Multirelation* R ist ein Paar $R = (s, v)$ mit

- einem Schema $s = \{A_1, \dots, A_n\}$, das aus einer Menge von *Attributen* (genauer: Attributnamen) besteht und für jedes Attribut A_i einen Domain $\text{dom}(A_i) \in \{D_1, \dots, D_m\}$ festlegt, und
- einer Ausprägung (auch Wert genannt) v , die eine Multimenge über der Grundmenge $\text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n)$ ist.

Schema und Ausprägung von R werden - wie bei Relationen - mit $\text{sch}(R)$ und $\text{val}(R)$ bezeichnet. Eine Multirelation hat in der Regel *keinen* Primärschlüssel.

Beispiel einer Multirelation:

Name	Stadt	Rabatt
Lauer	Merzig	0.10
Schneider	Homburg	0.20
Schneider	Homburg	0.20
Schulz	Merzig	0.10
Schulz	Merzig	0.05
Meier	Saarlouis	0.05

Jede Relation ist zugleich auch eine Multirelation (mit Häufigkeit 1 für alle Tupel). Umgekehrt kann jede Multirelation R durch eine Funktion χ wie folgt in eine Relation $\chi(R)$ konvertiert werden: $\text{val}(\chi(R)) = \{t \mid R(t) > 0\}$.

Multimengenoperationen:

Für zwei Multirelationen R, S mit $\text{sch}(R) = \text{sch}(S)$ sind die folgenden Multimengenoperationen definiert:

- Vereinigung $R \cup_+ S$:
 $\text{sch}(R \cup_+ S) = \text{sch}(R)$
 $\text{val}(R \cup_+ S) = t \mapsto R(t) + S(t)$
bzw. $\{t \mid t \in \text{val}(R) \vee t \in \text{val}(S) \text{ und die Häufigkeit von } t \text{ ist } R(t)+S(t)\}$
- Durchschnitt $R \cap_+ S$:
 $\text{sch}(R \cap_+ S) = \text{sch}(R)$
 $\text{val}(R \cap_+ S) = t \mapsto \min(R(t), S(t))$
bzw. $\{t \mid t \in \text{val}(R) \wedge t \in \text{val}(S) \text{ und die Häufigkeit von } t \text{ ist } \min(R(t), S(t))\}$
- Differenz $R -_+ S$:
 $\text{sch}(R -_+ S) = \text{sch}(R)$
 $\text{val}(R -_+ S) = t \mapsto R(t) - S(t) \text{ falls } R(t) \geq S(t)$
 $0 \quad \text{sonst}$
bzw. $\{t \mid t \in \text{val}(R) \text{ und die Häufigkeit von } t \text{ ist } R(t)-S(t) \geq 0\}$

Selektion σ_+ (Filterung, Auswahl von Zeilen einer Tabelle):

Sei F eine Boolesche Formel über einfachen Vergleichsbedingungen zwischen zwei Attributen einer Multirelation oder einem Attribut und einer Konstanten. Das Resultat einer Selektion $\sigma_+[F](R)$ auf einer Multirelation R (auch $\sigma_{+F}(R)$ geschrieben) ist wie folgt definiert:

$$\begin{aligned}\text{sch}(\sigma_+[F](R)) &= \text{sch}(R) \\ \text{val}(\sigma_+[F](R)) &= t \mapsto R(t) \text{ falls } R(t) \wedge F(t) \\ &0 \quad \text{sonst} \\ &\text{bzw. } \{t \mid t \in R \wedge F(t) \text{ und die Häufigkeit von } t \text{ ist } R(t)\} \\ &\text{wobei } F(t) \text{ bedeutet, daß } t \text{ die Bedingung } F \text{ erfüllt.}\end{aligned}$$

Die Menge der möglichen Filterformeln F ist wie bei der Selektion auf Relationen definiert.

Projektion π_+ (Auswahl von Spalten einer Tabelle):

Sei $A \subseteq \text{sch}(R)$ eine Teilmenge der Attribute einer Multirelation R . Das Resultat einer Projektion $\pi_+[A](R)$ auf der Multirelation R (auch $\pi_{+A}(R)$ geschrieben) ist wie folgt definiert:

$$\begin{aligned}\text{sch}(\pi_+[A](R)) &= A \\ \text{val}(\pi_+[A](R)) &= t \mapsto \Sigma \{R(r) \mid r \in \text{val}(R) \text{ und } r.A = t.A\} \\ &\text{bzw. } \{t \mid \exists r \in \text{val}(R): t.A = r.A \text{ und die Häufigkeit von } t \text{ ist } |\{d \in \text{val}(R) \mid d.A = t.A\}| \}\end{aligned}$$

Achtung: Die Projektion auf Multirelationen beinhaltet *keine* Duplikateliminierung.

Zuweisung (Umbenennung von Attributen):

Seien R und S zwei Multirelationen mit $\text{sch}(R) = \{A_1, \dots, A_n\}$ und $\text{sch}(S) = \{B_1, \dots, B_n\}$, so daß für alle i gilt: $\text{dom}(A_i) = \text{dom}(B_i)$. Die Zuweisung $R := S$ bedeutet, daß sich die Ausprägung von R wie folgt ändert: $\text{val}(R) = \text{val}(S)$. Ausführlicher schreibt man auch $R(A_1, \dots, A_n) := S(B_1, \dots, B_n)$ bzw. $R(A_1, \dots, A_n) := S(E_1, \dots, E_n)$ mit Ausdrücken E_1, \dots, E_n , die wie bei der Zuweisung für Relationen gebildet werden. Im letzteren Fall ist $\text{val}(R) = \{t \mid \text{es gibt } s \in S \text{ und } t.A_i = E_i(s) \text{ für alle } i\}$.

Kartesisches Produkt \times_+ (Kombination von Zeilen zweier Tabellen):

Seien R, S zwei Multirelationen mit Schemata $A = \text{sch}(R)$ und $B = \text{sch}(S)$. Sei A' ein Schema, bei dem alle Attributnamen A_i , die auch in $\text{sch}(S)$ vorkommen, unbenannt sind in $R.A_i$, und sei B' ein Schema, bei dem alle Attributnamen A_i , die auch in $\text{sch}(R)$ vorkommen, umbenannt sind in $S.A_i$. Das Resultat des kartesischen Produkts $R \times_+ S$ ist wie folgt definiert:

$$\begin{aligned} \text{sch}(R \times_+ S) &= A' \cup B' \\ \text{val}(R \times_+ S) &= t \mapsto \begin{cases} R(s) * S(s) & \text{falls } t.A' = r.A \text{ und } t.B' = s.B \\ 0 & \text{sonst} \end{cases} \\ &\text{bzw. } \{t \mid \exists r \in \text{val}(R) \exists s \in \text{val}(S): t.A' = r.A \text{ und } t.B' = s.B \\ &\quad \text{und die H\u00e4ufigkeit von } t \text{ ist } R(r) * S(s)\} \end{aligned}$$

Aggregation α_+ (Zusammenfassen von Zeilen einer Tabelle):

Sei $A \in \text{sch}(R)$ und sei f eine Funktion, die Multimengen \u00fcber $\text{dom}(A)$ in einen Wertebereich W abbildet (z.B. max, min, sum, count, median). Das Resultat einer Aggregation $\alpha_+ [A, f](R)$ auf der Multirelation R (auch $\alpha_{+, A, f}(R)$ geschrieben) ist wie folgt definiert:

$$\begin{aligned} \text{sch}(\alpha_+ [A, f](R)) &= A' \text{ mit } \text{dom}(A') = W \\ \text{val}(\alpha_+ [A, f](R)) &= f(\pi_{+, A}(R)) \end{aligned}$$

Aggregationsfunktionen f bilden eine Multimenge auf einen einzelnen Wert ab.

Gruppierung γ_+ (Zusammenfassen von \u00c4quivalenzklassen der Zeilen einer Tabelle):

Die Wertgleichheit auf einer Multimenge M ist eine \u00c4quivalenzrelation \sim auf den Elementen von M mit H\u00e4ufigkeit ≥ 1 , f\u00fcr die gilt $x \sim y$ genau dann, wenn $x = y$.

Sei $X \subseteq \text{sch}(R)$, $A \in \text{sch}(R)$, und sei f eine Funktion, die Multimengen \u00fcber $\text{dom}(A)$ in einen Wertebereich W abbildet. Das Resultat einer Gruppierung $\gamma_+ [X, A, f](R)$ auf der Multirelation R (auch $\gamma_{+, X, A, f}(R)$ geschrieben) ist eine Multimenge, die wie folgt definiert ist:

$$\begin{aligned} \text{sch}(\gamma_+ [X, A, f](R)) &= X \cup \{A'\} \text{ mit } \text{dom}(A') = W \\ \text{val}(\gamma_+ [X, A, f](R)) &= \{t \mid \text{es gibt eine \u00c4quivalenzklasse } G \text{ von } \pi_{+, X}(R) \text{ unter der} \\ &\quad \text{Wertgleichheit und } t.X \text{ ist der Wert der Tupel in } G \\ &\quad \text{und } t.A' = f(\pi_{+, A}(G)) \} \end{aligned}$$

Die hier zugrundeliegenden \u00c4quivalenzklassen hei\u00dfen auch "(Wertgleichheits-) Gruppen".

Beispielanfragen auf Multirelationen:

Produkte:

PNr	Bez	Gewicht	Preis	Lagerort	Vorrat
1	Papier	2.000	20.00	Homburg	10000
2	Platte	1.000	2500.00	Saarbr\u00fccken	400
3	Drucker	5.000	2000.00	Merzig	200
4	Bildschirm	5.000	3000.00	Merzig	80
5	Disketten	0.500	20.00	Homburg	5000
6	Maus	0.250	100.00	Homburg	200
7	Speicher	0.100	200.00	Saarbr\u00fccken	2000

- 1) Bestimme die (Anzahl der) Produkte unter 50 DM sowie deren Lagerorte und Preise:
 $\sigma_+ [\text{Preis} < 50.00] (\pi_+ [\text{Lagerort, Preis}] (\text{Produkte}))$

Resultat:

Lagerort	Preis
Homburg	20.00
Homburg	20.00

- 2) Bestimme die Gesamtstückzahl (aller Produkte) über alle Lager:
 Resultat (Gesamtvorrat) := $\alpha_+ [\text{Vorrat, sum}] (\text{Produkte})$

Resultat:

Gesamtvorrat
17880

- 3) Bestimme für jedes Lager die Gesamtstückzahl aller dort gelagerten Produkte:
 Resultat (Lagerort, Gesamtvorrat) := $\gamma_+ [\{\text{Lagerort}\}, \text{Vorrat, sum}] (\text{Produkte})$

Resultat:

Lagerort	Gesamtvorrat
Homburg	15200
Saarbrücken	2400
Merzig	280

- 4) Bestimme für jedes Lager die Gesamtkapitalbindung (Stückzahl * Preis) aller dort gelagerten Produkte:
 $P1 := \pi_+ [\text{Lagerort, Preis, Vorrat}] (\text{Produkte})$
 $P2 (\text{Lagerort, Wert}) := P1(\text{Lagerort, Preis*Vorrat})$
 Resultat (Lagerort, Kapitalbindung) := $\gamma_+ [\{\text{Lagerort}\}, \text{Wert, sum}] (P2)$

Resultat:

Lagerort	Kapitalbindung
Homburg	320 000
Saarbrücken	1 400 000
Merzig	640 000

Weitere Erweiterungen der Relationenalgebra

Orthogonal zu der Erweiterung auf Multimengen und Listen oder auch auf geschachtelte Relationen (sog. NF²-Relationen: Non First Normal Form), bei denen Tupelkomponenten selbst ganze Relationen sein dürfen, kann man die Relationenalgebra auch noch um mächtigere Operatoren ergänzen, die nicht auf die Standardoperatoren zurückführbar sind. Die wichtigste Operation dieser Kategorie ist die transitive Hülle von binären (d.h. zweistelligen) Relationen.

Transitive Hülle R^+ einer binären Relation R :

Sei $R(A, B)$ eine binäre Relation mit $\text{dom}(A)=\text{dom}(B)$. Das Resultat der transitiven Hülle R^+ ist wie folgt definiert:

$$\text{sch}(R^+) = \text{sch}(R)$$

$\text{val}(R^+)$ ist die kleinste Menge, für die gilt:

- (i) für jedes $r \in R$ gilt $r \in R^+$ und
- (ii) für $t \in R^+$ und $r \in R$ mit $t.B=r.A$ gilt $(t.A, r.B) \in R^+$

R^+ ist die kleinste transitiv abgeschlossene Menge, die R enthält.

R^+ ist die (bzgl. \subseteq) kleinste Lösung V der Fixpunktgleichung $V = R \cup (V \mid \times \mid [B=A] R)$.

Wenn man R als Kanten eines gerichteten Graphen interpretiert, $\text{dom}(A)=\text{dom}(B)$ also Knotenmengen sind, ist R^+ die Menge aller Knotenpaare, die über einen Weg verbunden sind.

Beispiel:

Flugverbindungen := $\pi[\text{Abflugort}, \text{Zielort}]$ (Flüge)

Flüge

FlugNr	Abflugort	Zielort	...
LH58	Frankfurt	Chicago	
AA371	Chicago	Phoenix	
DA77	Phoenix	Yuma	
AA70	Frankfurt	Dallas	
AA351	Dallas	Phoenix	
UA111	Chicago	Dallas	

Flugverbindungen

Abflugort	Zielort
Frankfurt	Chicago
Frankfurt	Dallas
Frankfurt	Phoenix
Frankfurt	Yuma
...	...

Ergänzende Literatur zu Kapitel 5

Jeffrey D. Ullman: Principles of Database and Knowledge Base Systems Vol. 1, Computer Science Press, 1988

David Maier: The Theory of Relational Databases, Computer Science Press, 1983

Serge Abiteboul, Richard Hull, Victor Vianu: Foundations of Databases, Addison-Wesley, 1995

Giedrius Slivinskas, Christian S. Jensen, Richard T. Snodgrass: A Foundation for Conventional and Temporal Query Optimization Addressing Duplicates and Ordering. IEEE Transactions on Knowledge and Data Engineering Vol. 13 No.1, 2001