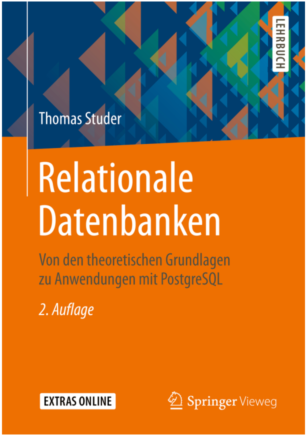

Thomas Studer
Relationale Datenbanken:
Von den theoretischen Grundlagen zu Anwendungen mit PostgreSQL
2. Auflage
Springer Vieweg, 2019
ISBN 978-3-662-58975-5

UNIVERSITÄT BERN

Datenbanken

—

Relationale Algebra

Thomas Studer

Institut für Informatik
Universität Bern

UNIVERSITÄT BERN

Input Relationen

Sei $\mathcal{S}$ ein Schema des gegebenen DB-Schemas und sei R eine Instanz von $\mathcal{S}$. Dann ist R eine Basisrelation der relationalen Algebra.

Konstante Relationen

Sei A ein Attribut mit Domäne D, welches im gegebenen DB-Schema vorkommt. Weiter sei a eine Element von D. Dann ist die konstante 1-stellige Relation $\{(a)\}$ eine Basisrelation der relationalen Algebra. Offensichtlich ist die Relation $\{(a)\}$ eine Instanz des Schemas (A).

Bemerkung

Wir erinnern uns, dass der Wert `Null` *zu jeder Domäne gehört. Somit gilt für jedes Attribut* A*, dass die konstante Relation* $\{(\texttt{Null})\}$ *eine Instanz von* (A) *ist.*

Projektion

Wir gehen aus von Attributen $A_1, \ldots, A_n$, dem Relationenschema

$$\mathcal{S} = (A_1, \ldots, A_n)$$

sowie einer Teilmenge $\{A_{i_1}, \ldots, A_{i_m}\}$ von $\{A_1, \ldots, A_n\}$. Ist R eine Instanz von $\mathcal{S}$, so setzen wir

$$\begin{aligned}\pi_{A_{i_1},\ldots,A_{i_m}}(R) :=\big\{(b_1, \ldots, b_m) \mid &\text{ es gibt ein } a \in R \text{ mit} \\ &b_1 \simeq \pi_{i_1}(a) \text{ und } \cdots \text{ und } b_m \simeq \pi_{i_m}(a)\big\} \ .\end{aligned}$$

Das Resultat dieser Projektion ist also eine Instanz des Schemas

$$(A_{i_1}, \ldots, A_{i_m}) \ .$$

Da die erhaltene Relation wieder eine Menge ist, werden etwa auftretende Duplikate selbstverständlich entfernt.

Beispiel

Betrachten wir die Relation

besucht

MatNr	Vorlesung
1	Datenbanken
1	Programmieren
2	Programmieren

Für die Projektion $\pi_{\texttt{Vorlesung}}(\texttt{besucht})$ erhalten wir folgende Tabelle:

$\pi_{\texttt{Vorlesung}}(\texttt{besucht})$

Vorlesung
Datenbanken
Programmieren

Beispiel

Betrachten wir die Relation

Studierende	
MatNr	Name
1	Ann
2	Tom

Es gilt dann:

$\pi_{\texttt{Name},\texttt{MatNr},\texttt{MatNr}}(\texttt{Studierende})$		
Name	MatNr	MatNr
Ann	1	1
Tom	2	2

Kartesisches Produkt

Wir gehen aus von Attributen $A_1, \ldots, A_m$ und $B_1, \ldots, B_n$, den Relationenschemata

$$\mathcal{R} = (A_1, \ldots, A_m) \text{ und } \mathcal{S} = (B_1, \ldots, B_n) \ ,$$

sowie einer Instanz R von $\mathcal{R}$ und einer Instanz S von $\mathcal{S}$. Wir setzen

$$R \times S := \\ \{(x_1, \ldots, x_{m+n}) \mid (x_1, \ldots, x_m) \in R \text{ und } (x_{m+1}, \ldots, x_{m+n}) \in S\} \ .$$

Das Relationenschema für das kartesische Produkt $R \times S$ hat die Form

$$(R.A_1, \ldots, R.A_m, S.B_1, \ldots, S.B_n) \ .$$

Für diejenigen Attribute C die nur in einem der beiden Schemata $\mathcal{R}$ und $\mathcal{S}$ vorkommen können wir

$$C \text{ anstelle von } R.C\text{, beziehungsweise } S.C,$$

schreiben.

Beispiel

Studierende

MatNr	Name
1	Ann
2	Tom

besucht

MatNr	Vorlesung
1	Datenbanken
1	Programmieren
2	Programmieren

Studierende $\times$ besucht

Studierende.MatNr	Name	besucht.MatNr	Vorlesung
1	Ann	1	Datenbanken
1	Ann	1	Programmieren
1	Ann	2	Programmieren
2	Tom	1	Datenbanken
2	Tom	1	Programmieren
2	Tom	2	Programmieren

n-stellige konstante Relationen

Mit Hilfe des kartesischen Produktes können wir nun konstante n-stellige Relationen für beliebige n definieren.

Insbesondere ist das kartesische Produkt

$$\{(\mathtt{Null})\} \times \cdots \times \{(\mathtt{Null})\} = \{(\mathtt{Null}, \ldots, \mathtt{Null})\}$$

eine Relation über $(A_1, \ldots, A_n)$ für beliebige Attribute A_i.

Prädikat

Ein Prädikat über Attributen $A_1, \ldots, A_n$ ist folgendermassen aufgebaut:

- Argumente sind konstante Werte und die Attributnamen $A_1, \ldots, A_n$;
- als Vergleichsoperatoren verwenden wir

$$<, \quad \leq, \quad >, \quad \geq, \quad =, \quad \neq;$$

- komplexe Prädikate werden aufgebaut durch die Junktoren

$$\neg \text{ (nicht)}, \quad \vee \text{ (oder)}, \quad \wedge \text{ (und)}.$$

Wahrheitswert

Bei der Auswertung von Prädikaten ist zu beachten, dass wir nicht nur die Wahrheitswerte `true` und `false` zur Verfügung haben, sondern noch einen dritten Wahrheitswert `unknown`. Dieser wird für das Resultat von Vergleichen verwendet, bei denen der Wert `Null` involviert ist.

Beispielsweise liefert der Vergleich $7 < \texttt{Null}$ das Ergebnis `unknown` und auch $\texttt{Null} = \texttt{Null}$ resultiert in `unknown`.

Negation

$\neg$ (nicht)	
`true`	`false`
`false`	`true`
`unknown`	`unknown`

Disjunktion

$\vee$ (oder)	`true`	`false`	`unknown`
`true`	`true`	`true`	`true`
`false`	`true`	`false`	`unknown`
`unknown`	`true`	`unknown`	`unknown`

Konjunktion

$\wedge$ (und)	true	false	unknown
true	true	false	unknown
false	false	false	false
unknown	unknown	false	unknown

Null Werte

Die Definition der logischen Negation ist verträglich mit der Semantik des `Null`-Wertes. Insbesondere gilt für alle a:

$$\neg(a = \texttt{Null}) \quad \text{ist gleich} \quad a \neq \texttt{Null} \ .$$

In der Tat, wir haben

$$\neg(a = \texttt{Null}) \quad \text{ist} \quad \neg(\texttt{unknown}) \quad \text{ist} \quad \texttt{unknown}$$

und

$$a \neq \texttt{Null} \quad \text{ist} \quad \texttt{unknown} \ .$$

Selektion

Gegeben seien Attribute $A_1, \ldots, A_n$, ein Prädikat Θ über $A_1, \ldots, A_n$ sowie das Relationenschema

$$\mathcal{S} = (A_1, \ldots, A_n) \ .$$

Ist R eine Instanz von $\mathcal{S}$, so ist die Selektion

$$\sigma_\Theta(R)$$

die Menge aller Tupel aus R, deren Werte dem Prädikat Θ den Wahrheitswert `true` geben.

Formal setzen wir

$$\sigma_\Theta(R) := \{t \mid t \in R \text{ und } \Theta(t)\} \ .$$

Damit ist $\sigma_\Theta(R)$ ebenfalls eine Instanz des Schemas $\mathcal{S}$.

Beispiel

Autos

Marke	Farbe	Baujahr	FahrerId
Opel	silber	2010	1
Opel	schwarz	2010	2
VW	rot	2014	2
Audi	schwarz	2014	3
VW	blau	2015	-

Sei $\Theta :\Longleftrightarrow \texttt{Marke} = \texttt{'Opel'} \vee \texttt{FahrerId} = 2$ dann

$\sigma_\Theta(\texttt{Autos})$

Marke	Farbe	Baujahr	FahrerId
Opel	silber	2010	1
Opel	schwarz	2010	2
VW	rot	2014	2

Selektionsprädikat

Oft geben wir das Prädikat explizit in der Selektionsoperation an. Das heisst, wir schreiben direkt

$$\sigma_{\texttt{Marke}='\texttt{Opel}' \vee \texttt{FahrerId}=2}(\texttt{Autos})$$

und führen keinen eigenen Namen für das Selektionsprädikat ein.

Null Werte

Wir betrachten nochmals die Tabelle `Autos`. Für das Selektionsprädikat

$$\Theta \quad := \quad \texttt{FahrerId} = 1 \;\vee\; \texttt{FahrerId} \neq 1$$

gilt

$$\texttt{Autos} \;\neq\; \sigma_{\Theta}(\texttt{Autos}) \;\;.$$

In der Tat erfüllt das Tupel

$$(\texttt{VW}, \texttt{blau}, 2015, \texttt{Null})$$

das Prädikat Θ nicht, da sowohl $\texttt{Null} = 1$ als auch $\texttt{Null} \neq 1$ zu `unknown` ausgewertet werden und damit auch Θ zu `unknown` ausgewertet wird. Dies ist korrekt, wenn wir beachten, dass `Null` für *keinen Wert* stehen kann.

Null Werte 2

Hätte `Null` nur die Bedeutung *unbekannter Wert*, so müsste dieser unbekannte Wert entweder $= 1$ oder $\neq 1$ sein.

Obwohl wir nicht wissen, welcher dieser beiden Fälle gilt, so wissen wir doch, dass einer von beiden gelten muss, und damit müsste Θ zu `true` evaluiert werden.

Umbenennung

Durch Verknüpfung der bisher eingeführten Operationen erhalten wir Ausdrücke der relationalen Algebra. Mit diesen können aus den Basisrelationen neue Relationen berechnet werden. Diesen berechneten Relationen haben wir implizit immer auch ein Schema zugeordnet. Mit Hilfe der Umbenennungsoperation ρ können wir nun die Attribute dieses Schemas umbenennen.

Ist E ein Ausdruck der relationalen Algebra, der für eine n-stellige Relation steht, so liefert der Ausdruck

$$\rho_{A_1,\ldots,A_n}(E)$$

das Ergebnis von E unter dem Schema $(A_1, \ldots, A_n)$.

Beispiel

Autos

Marke	Farbe	Baujahr	FahrerId
Opel	silber	2010	1
Opel	schwarz	2010	2
VW	rot	2014	2
Audi	schwarz	2014	3
VW	blau	2015	-

Sei $\quad \Theta \quad :\Longleftrightarrow \quad \texttt{Marke} = \texttt{'Opel'} \;\vee\; \texttt{FahrerId} = 2$

Der Ausdruck $\quad \rho_{\texttt{Automarke},\texttt{Jahrgang}}(\pi_{\texttt{Marke},\texttt{Baujahr}}(\sigma_{\Theta}(\texttt{Autos}))) \quad$ liefert:

Automarke	Jahrgang
Opel	2010
VW	2014

Mehrfache Attributnamen

Mit der Umbenennungsoperation können wir das Problem von mehrfach auftretenden Attributnamen lösen. Wir haben gesehen, dass

$$\pi_{\texttt{MatNr}}\big(\pi_{\texttt{Name},\texttt{MatNr},\texttt{MatNr}}(\texttt{Studierende})\big)$$

kein zulässiger relationaler Ausdruck ist.

Mit Hilfe der ρ-Operation können wir nun eines der `MatNr` Attribute umbennen und so einen zulässigen Ausdruck formulieren durch:

$$\pi_{\texttt{MatNr}}\Big(\rho_{\texttt{Name},\texttt{MatNr},\texttt{MatNr2}}\big(\pi_{\texttt{Name},\texttt{MatNr},\texttt{MatNr}}(\texttt{Studierende})\big)\Big)$$

.

Umbennenung und kartesisches Produkt

Betrachte eine Relation $\mathtt{S}$ über einem Schema $(\mathtt{A}, \mathtt{B})$. Das kartesische Produkt

$$\rho_{\mathtt{L.A},\mathtt{L.B}}(\mathtt{S}) \times \rho_{\mathtt{R.A},\mathtt{R.B}}(\mathtt{S})$$

ist dann eine Relation über dem Schema

$$(\ \mathtt{L.A},\ \mathtt{L.B},\ \mathtt{R.A},\ \mathtt{R.B}\ )\ .$$

In diesem Schema sind die Namen der Attribute eindeutig.

Wir werden diesen Ansatz später noch verwenden und führen dafür folgende abkürzende Schreibweise ein. Sei S eine Relation über einem Schema $(A_1, \ldots, A_n)$. Dann schreiben wir

$$\rho_L(\mathtt{S})$$

für

$$\rho_{L.A_1,\ldots,L.A_n}(\mathtt{S})\ .$$

Vereinigung und Mengendifferenz

Wir gehen aus von den Attributen $A_1, \ldots, A_n$ und dem Relationenschema

$$\mathcal{S} := (A_1, \ldots, A_n) \ ,$$

sowie Instanzen R und S von $\mathcal{S}$. Die Vereinigung $(R \cup S)$ und die Mengendifferenz $(R \setminus S)$ von R und S definieren wir dann durch

$$(R \cup S) := \{t \mid t \in R \text{ oder } t \in S\}$$

und

$$(R \setminus S) := \{t \mid t \in R \text{ und } t \notin S\} \ .$$

Offensichtlich sind $(R \cup S)$ und $(R \setminus S)$ ebenfalls Instanzen des Relationenschemas $\mathcal{S}$.

Bemerkungen

Wir beachten, dass Tupel, die sowohl in R als auch in S vorkommen in $(R \cup S)$ nicht mehrfach gezählt werden.

In der relationalen Algebra können nur *relative* Komplemente von Relationen S bezüglich von Relationen R eingeführt werden (mit Hilfe der Mengendifferenz $R \setminus S$). Dagegen ist es nicht möglich, das absolute Komplement einer Relation S, d.h. die Menge aller Tupel $\{t \mid t \notin S\}$, zu definieren.

Relationale Ausdrücke, Repetition

Basisrelationen

1. Inputrelationen
2. konstante Relationen

Grundoperationen

1. Projektion
2. kartesisches Produkt
3. Selektion
4. Umbenennung
5. Vereinigung
6. Differenz

Beispiel

Studierende

MatNr	Name
1	Ann
2	Tom

besucht

MatNr	Vorlesung
1	Datenbanken
1	Programmieren
2	Programmieren

Studierende $\times$ besucht

Studierende.MatNr	Name	besucht.MatNr	Vorlesung
1	Ann	1	Datenbanken
1	Ann	1	Programmieren
1	Ann	2	Programmieren
2	Tom	1	Datenbanken
2	Tom	1	Programmieren
2	Tom	2	Programmieren

Beispiel 2

Studierende $\times$ besucht

Studierende.MatNr	Name	besucht.MatNr	Vorlesung
1	Ann	1	Datenbanken
1	Ann	1	Programmieren
1	Ann	2	Programmieren
2	Tom	1	Datenbanken
2	Tom	1	Programmieren
2	Tom	2	Programmieren

$\sigma_{\texttt{Studierende.MatNr}=\texttt{besucht.MatNr}}(\texttt{Studierende} \times \texttt{besucht})$

Studierende.MatNr	Name	besucht.MatNr	Vorlesung
1	Ann	1	Datenbanken
1	Ann	1	Programmieren
2	Tom	2	Programmieren

Beispiel 3

$$\sigma_{\texttt{Studierende.MatNr}=\texttt{besucht.MatNr}}(\texttt{Studierende} \times \texttt{besucht})$$

Studierende.MatNr	Name	besucht.MatNr	Vorlesung
1	Ann	1	Datenbanken
1	Ann	1	Programmieren
2	Tom	2	Programmieren

$$\pi_{\texttt{Studierende.MatNr},\texttt{Name},\texttt{Vorlesung}}\left(\sigma_{\texttt{Studierende.MatNr}=\texttt{besucht.MatNr}}(\texttt{Studierende} \times \texttt{besucht})\right)$$

Studierende.MatNr	Name	Vorlesung
1	Ann	Datenbanken
1	Ann	Programmieren
2	Tom	Programmieren

Beispiel 4

Studierende.MatNr	Name	Vorlesung
1	Ann	Datenbanken
1	Ann	Programmieren
2	Tom	Programmieren

$$\rho_{\texttt{MatNr,Name,Vorlesung}} \\ (\pi_{\texttt{Studierende.MatNr},\texttt{Name},\texttt{Vorlesung}} \\ (\sigma_{\texttt{Studierende.MatNr}=\texttt{besucht.MatNr}}(\texttt{Studierende} \times \texttt{besucht})))$$

MatNr	Name	Vorlesung
1	Ann	Datenbanken
1	Ann	Programmieren
2	Tom	Programmieren

Definierte Operationen: Durchschnitt

Gegeben seien Attribute $A_1, \ldots, A_n$, das Relationenschema

$$\mathcal{S} := (A_1, \ldots, A_n) \ ,$$

sowie Instanzen R und S des Schemas $\mathcal{S}$. Der Durchschnitt $(R \cap S)$ von R und S ist dann gegeben durch

$$(R \cap S) := (R \setminus (R \setminus S)) \ .$$

Natürlicher Verbund

Gegeben

$$\mathcal{S} := (A_1, \ldots, A_m) \text{ und } \mathcal{T} := (B_1, \ldots, B_n)$$

sowie eine Instanz S von $\mathcal{S}$ und eine Instanz T von $\mathcal{T}$. Ausserdem gelte

$$\begin{aligned}
\{A_1, \ldots, A_m\} \cap \{B_1, \ldots, B_n\} &= \{A_{i_1}, \ldots, A_{i_p}\} && \text{mit } i_l < i_h \text{ für } l < h \\
\{A_1, \ldots, A_m\} \setminus \{A_{i_1}, \ldots, A_{i_p}\} &= \{A_{j_1}, \ldots, A_{j_q}\} && \text{mit } j_l < j_h \text{ für } l < h \\
\{B_1, \ldots, B_n\} \setminus \{A_{i_1}, \ldots, A_{i_p}\} &= \{B_{k_1}, \ldots, B_{k_r}\} && \text{mit } k_l < k_h \text{ für } l < h
\end{aligned}$$

Der natürliche Verbund $(S \bowtie T)$ ist nun definiert als

$$\begin{aligned}
S \bowtie T \quad := \quad & \rho_{A_{i_1},\ldots,A_{i_p},A_{j_1},\ldots,A_{j_q},B_{k_1},\ldots,B_{k_r}}(\\
& \qquad \pi_{L.A_{i_1},\ldots,L.A_{i_p},L.A_{j_1},\ldots,L.A_{j_q},R.B_{k_1},\ldots,R.B_{k_r}}(\\
& \qquad\qquad \sigma_{L.A_{i_1}=R.A_{i_1}\wedge\cdots\wedge L.A_{i_p}=R.A_{i_p}}(\rho_L(S) \times \rho_R(T)))) \ .
\end{aligned}$$

Natürlicher Verbund

Der natürliche Verbund ist also eine zweistellige Operation, die

1. die Attributnamen so umbenennt, dass sie mit $L.$, beziehungsweise mit $R.$, beginnen,
2. ein kartesisches Produkt bildet,
3. eine Selektion durchführt, welche Gleichheit auf den Attributen verlangt, die beiden Schemata gemeinsam sind,
4. mit einer Projektion die Duplikate dieser gemeinsamen Attribute entfernt und die gemeinsamen Attribute an den Anfang stellt,
5. mit einer Umbenennung den Attributen wieder ihre ursprünglichen Namen gibt.

Beispiel

Autos

Marke	Jahrgang	PersId
Opel	2010	1
VW	1990	1
Audi	2014	-
Skoda	2014	2

Personen

PersId	Name
1	Studer
2	Meier

Autos $\bowtie$ Personen

PersId	Marke	Jahrgang	Name
1	Opel	2010	Studer
1	VW	1990	Studer
2	Skoda	2014	Meier

Beispiel: unerwünschtes Ergebnis

Autos

Marke	Jahrgang	PersId
Opel	2010	1
VW	1990	1
Audi	2014	-
Skoda	2014	2

Personen

PersId	Name	Jahrgang
1	Studer	1990
2	Meier	1994

Autos $\bowtie$ Personen

Jahrgang	PersId	Marke	Name
1990	1	VW	Studer

Θ-Verbund

Der Θ-Verbund $R \bowtie_\Theta S$ von zwei Relationen R und S ist definiert durch

$$R \bowtie_\Theta S := \sigma_\Theta(R \times S) \ .$$

Beispiel

`Autos`

Marke	Jahrgang	PersId
Opel	2010	1
VW	1990	1
Audi	2014	-
Skoda	2014	2

`Personen`

PersId	Name	Jahrgang
1	Studer	1990
2	Meier	1994

Wir erhalten für

$$\texttt{Autos} \bowtie_{\texttt{Autos.PersId}=\texttt{Personen.PersId}} \texttt{Personen}$$

folgende Tabelle, wobei `A` für `Autos` und `P` für `Personen` steht:

A.Marke	A.Jahrgang	A.PersId	P.PersId	P.Name	P.Jahrgang
Opel	2010	1	1	Studer	1990
VW	1990	1	1	Studer	1990
Skoda	2014	2	2	Meier	1994

Beliebige Attribute

Ein Θ-Join bei dem im Prädikat Θ nur auf Gleichheit getestet wird heisst *Equi-Join*.

In einem allgemeinen Θ-Join können aber beliebige Prädikate verwendet werden. Beispielsweise liefert der Θ-Join

$$\texttt{Autos} \bowtie_{\texttt{Autos.PersId}=\texttt{Personen.PersId} \land \texttt{Personen.Jahrgang} \leq 1990} \texttt{Personen}$$

die Fahrer, welche 1990 oder früher geboren wurden, zusammen mit ihren Autos. Das heisst, wir erhalten folgende Tabelle für die Relationen `Autos` und `Personen` aus dem vorherigen Beispiel:

A.Marke	A.Jahrgang	A.PersId	P.PersId	P.Name	P.Jahrgang
Opel	2010	1	1	Studer	1990
VW	1990	1	1	Studer	1990

Linker äusserer Verbund

Autos

Marke	Jahrgang	PersId
Opel	2010	1
VW	1990	1
Audi	2014	-
Skoda	2014	2

Personen

PersId	Name
1	Studer
2	Meier

Der natürlichen Verbund `Autos` $\bowtie$ `Personen` enthält den `Audi` nicht.

Der linke äussere Verbund `Autos` $⟕$ `Personen` ist definiert durch:

$$R ⟕ S := (R \bowtie S) \cup \pi_{\mathcal{T}}\Big(\big(R \setminus \pi_{\mathcal{R}}(R \bowtie S)\big) \times \{(\mathtt{Null}, \ldots, \mathtt{Null})\}\Big)$$

Linker äusserer Verbund

Autos

Marke	Jahrgang	PersId
Opel	2010	1
VW	1990	1
Audi	2014	-
Skoda	2014	2

Personen

PersId	Name
1	Studer
2	Meier

Autos ⟕ Personen

PersId	Marke	Jahrgang	Name
1	Opel	2010	Studer
1	VW	1990	Studer
2	Skoda	2014	Meier
-	Audi	2014	-

Division Beispiel

Mechaniker

Name	Marke
Studer	Opel
Meier	Opel
Meier	VW
Meier	Audi

Garage

Marke
Opel
Audi

Gesucht: die Namen derjenigen Mechaniker, welche *alle* Automarken, die in der Garage vorkommen, reparieren können.

Mechaniker ÷ Garage

Name
Meier

Division: *-Operation

Gegeben seien die Schemata

$$\mathcal{S} = (A_1, \ldots, A_m) \qquad \text{und} \qquad \mathcal{T} = (A_{m+1}, \ldots, A_{m+n})$$

(mit paarweise verschiedenen Attributen). Weiter sei

$$\mathcal{R} = (A_{i_1}, \ldots, A_{i_{m+n}}) \qquad \text{mit } i_j \text{ und } i_k \text{ paarweise verschieden}$$

ein Schema mit den Attributen $A_1, \ldots, A_{m+n}$.
Ist S eine Instanz von $\mathcal{S}$ und T eine Instanz von $\mathcal{T}$, ist s ein m-Tupel aus S und ist t ein n-Tupel aus T, so schreiben wir

$$(s * t)$$

für das $(m+n)$-Tupel, das wir durch geeignete Konkatenation von s und t erhalten, so dass gilt:

$$\pi_j(s * t) \simeq s[A_{i_j}] \quad \text{falls } i_j \leq m$$
$$\pi_j(s * t) \simeq t[A_{i_j}] \quad \text{falls } i_j > m$$

Division: *-Operation, Beispiel

Gegeben seien Attribute `A`, `B`, `C` und `D`, die Schemata

$$\mathcal{S} = (A, B) \qquad \mathcal{T} = (C, D) \qquad \mathcal{R} = (C, B, D, A)$$

sowie eine Instanz S von $\mathcal{S}$ und eine Instanz T von $\mathcal{T}$. Es seien nun

$$s = (1, 2) \in S \qquad \text{und} \qquad t = (3, 4) \in T \ .$$

Damit gilt

$$s * t = (3, 2, 4, 1) \ .$$

Division

Sei $\{B_1, \dots, B_n\}$ eine echte Teilmenge der Attributmenge $\{A_1, \dots, A_m\}$. Ausserdem sei

$$\{A_{i_1}, \dots, A_{i_{m-n}}\} := \{A_1, \dots, A_m\} \setminus \{B_1, \dots, B_n\} \text{ mit } i_l < i_k \text{ für } l < k.$$

Schliesslich betrachten wir noch die Schemata

$$\begin{aligned}
\mathcal{R} &:= (A_1, \dots, A_m), \\
\mathcal{S} &:= (B_1, \dots, B_n), \\
\mathcal{T} &:= (A_{i_1}, \dots, A_{i_{m-n}}) \ .
\end{aligned}$$

Sind R eine Instanz von $\mathcal{R}$ und S eine Instanz von $\mathcal{S}$, so ist die Division $(R \div S)$ von R durch S eine Relation die zum Schema $\mathcal{T}$ gehört. Ein $(m-n)$-Tupel t aus $\pi_{A_{i_1}, \dots, A_{i_{m-n}}}(R)$ soll genau dann Element der Relation $(R \div S)$ sein, falls für alle Tupel s aus S das Tupel $(t * s)$ zu R gehört, d.h.

$$(R \div S) := \{t \in \pi_{A_{i_1}, \dots, A_{i_{m-n}}}(R) \mid (\forall s \in S)((t * s) \in R)\} \ .$$

Division als Ausdruck der relationalen Algebra

Es gilt

$$(R \div S) \quad = \quad \pi_{A_{i_1},\ldots,A_{i_{m-n}}}(R) \setminus \pi_{A_{i_1},\ldots,A_{i_{m-n}}}\Big(\pi_{A_1,\ldots,A_m}(\pi_{A_{i_1},\ldots,A_{i_{m-n}}}(R) \times S) \setminus R\Big)$$

Für unser Beispiel würde dieser relationale Ausdruck etwa Folgendes bedeuten:

Finde alle Mechaniker,

für die es keine Automarke in der Garage gibt,

die sie nicht reparieren können.

Division mit mehreren Attributen

R

A1	A2	B1	B2
1	2	3	4
1	2	5	6
2	3	5	6
5	4	3	4
5	4	5	6
1	2	4	5
1	2	7	8

S

B1	B2
3	4
5	6
7	8

R ÷ S

A1	A2
1	2

Division als Inverses des kartesischen Produkts

Lemma

Seien $\mathcal{R} = (A_1, \ldots, A_m)$ *und* $\mathcal{S} = (B_1, \ldots, B_n)$ *zwei Relationenschemata,* R *eine Instanz von* $\mathcal{R}$ *und* S *eine Instanz von* $\mathcal{S}$*. Wir nehmen an, dass* $\mathcal{R}$ *und* $\mathcal{S}$ *keine gemeinsamen Attributnamen enthalten. Somit können wir das kartesische Produkt* $R \times S$ *als Relation über dem Schema* $(A_1, \ldots, A_m, B_1 \ldots, B_n)$ *auffassen.*
Dann gilt, falls S *nicht-leer ist,*

$$(R \times S) \div S = R \ .$$

Die Einschränkung auf nicht-leere Relationen S ist keine Überraschung.
Über den rationalen Zahlen gilt $(a \cdot b)/b = a$ auch nur für $b \neq 0$.

Division mit Rest

Die Division der relationalen Algebra verhält sich wie eine Division mit Rest (d.h. wie eine Ganzzahl-Division). Damit meinen wir, dass zwar

$$(R \div S) \times S \subseteq R$$

gilt, aber $(R \div S) \times S = R$ nicht gelten muss.

Genau so gilt für die Ganzzahl-Division nur $(7/3) * 3 \leq 7$ aber nicht $(7/3) * 3 = 7$.

Queries 1

Wir betrachten eine Film-Datenbank. Dazu verwenden wir die Attribute `PId` (Personen-Id), `Fn` (Familienname), `Vn` (Vorname), `FId` (Film-Id), `Dt` (Datum), `Reg` (Regisseur), `Titel` und `Jahr` und betrachten die Relationenschemata

$$\begin{aligned} \mathcal{P} &= (\underline{\texttt{PId}}, \texttt{Fn}, \texttt{Vn}) \ , \\ \mathcal{PF} &= (\underline{\texttt{PId}}, \underline{\texttt{FId}}, \underline{\texttt{Dt}}) \ , \\ \mathcal{F} &= (\underline{\texttt{FId}}, \texttt{Reg}, \texttt{Titel}, \texttt{Jahr}) \ . \end{aligned}$$

Ferner sei die Relation `Personen` eine Instanz von $\mathcal{P}$, die Relation `Schaut` eine Instanz von $\mathcal{PF}$ und `Filme` eine Instanz von $\mathcal{F}$.

Queries 2

Query 1

Wie lauten die Titel der Filme aus dem Jahr 2002 bei denen `Spielberg` Regie geführt hat?

$$\pi_{\texttt{Titel}}(\sigma_{\texttt{Jahr}=\texttt{2002}\wedge\texttt{Reg}=\texttt{'Spielberg'}}(\texttt{Filme}))$$

Query 2

Ermittle Familien- und Vorname derjenigen Personen, die Filme geschaut haben mit `Spielberg` oder `Coppola` als Regisseur.

$$\pi_{\texttt{Fn,Vn}}(\texttt{Personen} \bowtie (\texttt{Schaut} \bowtie (\sigma_{\texttt{Reg}=\texttt{'Spielberg'}\vee\texttt{Reg}=\texttt{'Coppola'}}(\texttt{Filme}))))$$

Query 3

Nenne `Titel` und `Jahr` der Filme, welche `Eva Meier` vor dem 30. November 2009 geschaut hat.

$$\pi_{\texttt{Titel,Jahr}}(\sigma_{\texttt{Fn}=\texttt{'Meier'}\wedge\texttt{Vn}=\texttt{'Eva'}}(\texttt{Personen}) \bowtie (\sigma_{\texttt{Dt}<20091130}(\texttt{Schaut}) \bowtie \texttt{Filme}))$$

Queries 3

Query 4
Wie lauten die Personennummern der Personen, die alle Filme von `Spielberg` geschaut haben?

$$\pi_{\texttt{PId,FId}}(\texttt{Schaut}) \div \pi_{\texttt{FId}}(\sigma_{\texttt{Reg}='\texttt{Spielberg}'}(\texttt{Filme}))$$

In diesem Ausdruck ist es wichtig, dass in der Division die Projektion

$$\pi_{\texttt{PId,FId}}(\texttt{Schaut})$$

verwendet wird und nicht die ursprüngliche Relation `Schaut`. Die Abfrage

$$\pi_{\texttt{PId}}(\texttt{Schaut} \div \pi_{\texttt{FId}}(\sigma_{\texttt{Reg}='\texttt{Spielberg}'}(\texttt{Filme})))$$

liefert nämlich die Personennummer der Personen, die *an einem einzigen Tag* alle Filme von `Spielberg` geschaut haben.

Aggregatsfunktionen

- **count** (Zählen),
- **sum** (Summieren),
- **min**, **max** (Minimum bzw. Maximum),
- **avg** (Durchschnitt).

Group By

Gegeben seien eine Teilmenge $\{B_1, \ldots, B_n\}$ der Attributmenge $\{A_1, \ldots, A_m\}$, ein Element C von $\{A_1, \ldots, A_m\} \setminus \{B_1, \ldots, B_n\}$ und das Relationenschema

$$\mathcal{S} = (A_1, \ldots, A_m).$$

Ist $\mathbf{agg}$ eine Aggregatsfunktion und R eine Instanz von $\mathcal{S}$, so definieren wir die *GroupBy* Operation γ durch:

$$\begin{aligned}\gamma(R, (B_1, \ldots, B_n), \mathbf{agg}, C) :=& \\ &\{(b_1, \ldots, b_n, a) \mid (b_1, \ldots, b_n) \in \pi_{B_1,\ldots,B_n}(R) \text{ und} \\ &\qquad a = \mathbf{agg}(\{x \mid (b_1, \ldots, b_n, x) \in \pi_{B_1,\ldots,B_n,C}(R)\})\}.\end{aligned}$$

Als Schema dieser Relation verwenden wir:

$$(B_1, \ldots, B_n, \mathbf{agg}(C)) \ .$$

Queries 4

Query 5
Welcher Regisseur hat in welchem Jahr wie viele Filme gedreht?

$$\gamma(\texttt{Filme}, (\texttt{Reg}, \texttt{Jahr}), \mathbf{count}, \texttt{FId})$$

Query 6
In welchem Jahr hat `Spielberg` das letzte Mal Regie geführt?

$$\sigma_{\texttt{Reg}='\texttt{Spielberg}'}(\gamma(\texttt{Filme}, (\texttt{Reg}), \mathbf{max}, \texttt{Jahr}))$$

Aggregatsfunktion über Menge

Filme

FId	Jahr	Dauer
1	2010	120
2	2012	90
3	2012	120
4	2010	100
5	2012	120

Query 7
Welches ist die durchschnittliche Dauer der Filme pro Jahr?

$\gamma(\texttt{Filme}, (\texttt{Jahr}), \textbf{avg}, \texttt{Dauer})$ ergibt:

Jahr	$\textbf{avg}(\texttt{Dauer})$
2010	110
2012	105

Im Detail

Für das Jahr 2010 wird die durchschnittliche Dauer berechnet durch

$$\mathbf{avg}(\{100, 120\}) \ ,$$

was das korrekte Resultat liefert.

Für das Jahr 2012 wird die durchschnittliche Dauer jedoch berechnet durch

$$\mathbf{avg}(\{90, 120, 120\}) \ ,$$

was das (unerwünschte) Resultat 105 liefert.

Multimengen

Eine *Multimenge* ist eine Kollektion von Objekten, bei der mehrfache Vorkommnisse von Elementen registriert werden, aber ihre Reihenfolge unwichtig ist.

Wir können Multimengen also als Mengen auffassen, bei denen Elemente mehrfach vorkommen können, oder als Listen, bei denen die Reihenfolge nicht beachtet wird.

Multimengen Group By

Wir definieren die *Multimengen-GroupBy* Operation

$$\Gamma(R, (B_1, \ldots, B_n), \mathbf{agg}, C)$$

analog zu $\gamma(R, \{B_1, \ldots, B_n\}, \mathbf{agg}, C)$ mit dem Unterschied, dass wir den Input der Aggregatsfunktion $\mathbf{agg}$ als Multimenge behandeln.

Aggregatsfunktion über Multi-Menge

Filme

FId	Jahr	Dauer
1	2010	120
2	2012	90
3	2012	120
4	2010	100
5	2012	120

Query 7
Welches ist die durchschnittliche Dauer der Filme pro Jahr?

$\Gamma(\texttt{Filme}, (\texttt{Jahr}), \mathbf{avg}, \texttt{Dauer})$ ergibt:

Jahr	$\mathbf{avg}(\texttt{Dauer})$
2010	110
2012	110