


Datenbanken

Relationale Algebra


Thomas Studer

Institut für Informatik
Universität Bern

Repetition

Keine oder eine Beziehung 

Keine, eine oder mehrere Beziehungen 

Genau eine Beziehung 

Eine oder mehrere Beziehungen 

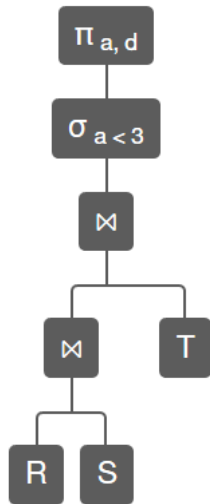
- $m : 1$ Beziehungen
- $m : n$ Beziehungen
- $1 : 1$ Beziehungen
- Ternäre Beziehungen
- Vererbung

Ausblick: Relationale Algebra und SQL

```
SELECT DISTINCT a, d  
FROM  
  R NATURAL JOIN S NATURAL JOIN T  
WHERE a < 3
```



$$\pi_{a,d}(\sigma_{a<3}(R \bowtie S \bowtie T))$$



<https://dbis-uibk.github.io/relax/calc.htm>

Ausblick: Relationale Ausdrücke

Basisrelationen

- ① Inputrelationen
- ② konstante Relationen

Grundoperationen

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

Input Relationen

Seien

- \mathcal{S} ein Schema des gegebenen DB-Schemas und
- R eine Instanz von \mathcal{S} .

Dann ist R eine Basisrelation der relationalen Algebra.

Konstante Relationen

Seien

- A ein Attribut mit Domäne D , welches im gegebenen DB-Schema vorkommt und
- a ein Element von D .

Dann ist die konstante 1-stellige Relation $\{(a)\}$ eine Basisrelation der relationalen Algebra.

Bemerkungen zu konstanten Relationen

Bemerkung

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 $\{(Null)\}$ eine Instanz von (A) ist.

Projektion

Seien

- $\mathcal{S} = (A_1, \dots, A_n)$ ein Relationenschema,
- $\{A_{i_1}, \dots, A_{i_m}\} \subseteq \{A_1, \dots, A_n\}$ und
- R eine Instanz von \mathcal{S} .

Dann definieren wir die *Projektion* auf A_{i_1}, \dots, A_{i_m} durch

$$\pi_{A_{i_1}, \dots, A_{i_m}}(R) := \{ (b_1, \dots, b_m) \mid \text{es gibt ein } a \in R \text{ mit} \\ b_1 \simeq \pi_{i_1}(a) \text{ und } \dots \text{ und } b_m \simeq \pi_{i_m}(a) \} \ .$$

Bemerkungen zur Projektion

Bemerkung

Das Resultat dieser Projektion ist also eine Instanz des Schemas

$$(A_{i_1}, \dots, A_{i_m}) \text{ .}$$

Bemerkung

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 $\pi_{\text{Vorlesung}}(\text{besucht})$ erhalten wir folgende Tabelle:

$\pi_{\text{Vorlesung}}(\text{besucht})$
Vorlesung
Datenbanken
Programmieren

Beispiel

Betrachten wir die Relation

Studierende	
MatNr	Name
1	Ann
2	Tom

Es gilt dann:

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

Kartesisches Produkt

Seien

- $\mathcal{R} = (A_1, \dots, A_m)$ und $\mathcal{S} = (B_1, \dots, B_n)$ Relationenschemata,
- R eine Instanz von \mathcal{R} und
- S eine Instanz von \mathcal{S} .

Dann definieren wir das *kartesische Produkt* von R und S durch

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

Bemerkungen zum kartesischen Produkt

Bemerkung

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

$$(R.A_1, \dots, R.A_m, S.B_1, \dots, S.B_n) \text{ .}$$

Konvention

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

C anstelle von $R.C$, 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

$$\{(\text{Null})\} \times \cdots \times \{(\text{Null})\} = \{(\text{Null}, \dots, \text{Null})\}$$

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

Prädikat

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

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

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

- komplexe Prädikate werden aufgebaut durch die Junktoren

\neg (nicht), \vee (oder), \wedge (und).

Beispiel

- $\text{Marke} = \text{'Opel'}$
- $\text{Baujahr} \geq 2015$
- $(\text{Farbe} = \text{'silber'} \vee \text{Farbe} = \text{'schwarz'}) \wedge \neg (\text{Baujahr} < 2010)$

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.

Beispiel

- Der Vergleich `7 < Null` liefert das Ergebnis `unknown`.
- Auch `Null = 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 :

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

In der Tat, wir haben

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

und

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

Selektion

Seien

- $\mathcal{S} = (A_1, \dots, A_n)$ ein Relationenschema,
- Θ ein Prädikat über A_1, \dots, A_n und
- R eine Instanz von \mathcal{S} .

Dann definieren wir die *Selektion* σ_Θ durch

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

Bemerkungen zur Selektion

Bemerkung

$\sigma_{\Theta}(R)$ ist die Menge aller Tupel aus R , deren Werte dem Prädikat Θ den Wahrheitswert *true* geben

Bemerkung

$\sigma_{\Theta}(R)$ ist ebenfalls eine Instanz des Schemas 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 $\Theta : \Longleftrightarrow \text{Marke} = \text{'Opel'} \vee \text{FahrerId} = 2$ dann

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

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

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

und führen keinen eigenen Namen für das Selektionsprädikat ein.

Null Werte

Wir betrachten nochmals die Tabelle Autos. Für das Selektionsprädikat

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

gilt

$$\text{Autos} \neq \sigma_{\Theta}(\text{Autos}) .$$

In der Tat erfüllt das Tupel

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

das Prädikat Θ nicht, da sowohl $\text{Null} = 1$ als auch $\text{Null} \neq 1$ 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

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

das Ergebnis von E unter dem Schema (A_1, \dots, 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 $\Theta : \Longleftrightarrow \text{Marke} = \text{'Opel'} \vee \text{FahrerId} = 2$

Der Ausdruck $\rho_{\text{Automarke, Jahrgang}}(\pi_{\text{Marke, Baujahr}}(\sigma_{\Theta}(\text{Autos})))$ 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

$$\pi_{\text{MatNr}}(\pi_{\text{Name}, \text{MatNr}, \text{MatNr}}(\text{Studierende}))$$

kein zulässiger relationaler Ausdruck ist.

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

$$\pi_{\text{MatNr}}(\rho_{\text{Name}, \text{MatNr}, \text{MatNr2}}(\pi_{\text{Name}, \text{MatNr}, \text{MatNr}}(\text{Studierende})))$$

.

Umbenennung und kartesisches Produkt

Betrachte eine Relation S über einem Schema (A, B) . Das kartesische Produkt

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

ist dann eine Relation über dem Schema

$$(L.A, L.B, R.A, R.B) .$$

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, \dots, A_n) . Dann schreiben wir

$$\rho_L(S)$$

für

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

Vereinigung und Mengendifferenz

Seien

- $\mathcal{S} := (A_1, \dots, A_n)$ ein Relationenschema und
- R und S Instanzen von \mathcal{S} .

Die *Vereinigung* $(R \cup S)$ und die *Mengendifferenz* $(R \setminus S)$ von R und S definieren wir durch

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

und

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

Bemerkungen zu Vereinigung und Mengendifferenz

Bemerkung

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

Bemerkung

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

Bemerkung

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

<u>MatNr</u>	<u>Name</u>
--------------	-------------

1	Ann
---	-----

2	Tom
---	-----

besucht

<u>MatNr</u>	<u>Vorlesung</u>
--------------	------------------

1	Datenbanken
---	-------------

1	Programmieren
---	---------------

2	Programmieren
---	---------------

Studierende \times besucht

<u>Studierende.MatNr</u>	<u>Name</u>	<u>besucht.MatNr</u>	<u>Vorlesung</u>
--------------------------	-------------	----------------------	------------------

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

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

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

Beispiel 3

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

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

$\pi_{\text{Studierende.MatNr,Name,Vorlesung}}$

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

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

$\rho_{\text{MatNr, Name, Vorlesung}}$

$(\pi_{\text{Studierende.MatNr, Name, Vorlesung}}$

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

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

Ausblick: Erweiterte Operationen

- 1 Durchschnitt
- 2 Natürlicher Verbund
- 3 Θ -Verbund
- 4 Linker äusserer Verbund
- 5 Division
- 6 Aggregatsfunktionen
 - 1 Group By
 - 2 Multimengen Group By

Definierte Operationen: Durchschnitt

Gegeben seien Attribute A_1, \dots, A_n , das Relationenschema

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

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

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

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

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

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

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

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 ⋈ 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 ⋈ 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

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

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

Equi-Join vs. Beliebige Prädikate

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.

Beispiel: Θ -Join mit beliebigen Prädikaten

$\text{Autos} \bowtie_{\text{Autos.PersId}=\text{Personen.PersId} \wedge \text{Personen.Jahrgang} \leq 1990} \text{Personen}$

liefert die Fahrer, welche 1990 oder früher geboren wurden, zusammen mit ihren Autos.

Mit dem vorherigen Beispiel erhalten wir:

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 $\text{Autos} \bowtie \text{Personen}$ enthält den Audi nicht.

Der linke äussere Verbund $\text{Autos} \ltimes \text{Personen}$ ist definiert durch:

$$R \ltimes S := (R \bowtie S) \cup \pi_{\mathcal{T}}\left(\left(R \setminus \pi_{\mathcal{R}}(R \bowtie S)\right) \times \{(\text{Null}, \dots, \text{Null})\}\right)$$

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 \bowtie Personen

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

Division Beispiel

Mechaniker		Garage
Name	Marke	Marke
Studer	Opel	Opel
Meier	Opel	Audi
Meier	VW	
Meier	Audi	

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

Mechaniker \div Garage
Name
Meier

Division: *-Operation

Gegeben seien die Schemata

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

(mit paarweise verschiedenen Attributen). Weiter sei

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

ein Schema mit den Attributen A_1, \dots, 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:

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

Division: *-Operation, Beispiel

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

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

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

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

Damit gilt

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

Division

Sei $\{B_1, \dots, B_n\} \subsetneq \{A_1, \dots, A_m\}$ und

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

Wir betrachten die Schemata

$$\mathcal{R} := (A_1, \dots, A_m),$$

$$\mathcal{S} := (B_1, \dots, B_n) .$$

und sei R eine Instanz von \mathcal{R} und S eine Instanz von \mathcal{S} .

Die Division $(R \div S)$ von R durch S ist definiert durch

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

Bemerkungen zur Division

Bemerkung

Die Division $(R \div S)$ ist eine Relation die zum Schema

$$\mathcal{T} := (A_{i_1}, \dots, A_{i_{m-n}})$$

gehört.

Bemerkung

Ein $(m - n)$ -Tupel t aus $\pi_{A_{i_1}, \dots, A_{i_{m-n}}}(R)$ ist genau dann Element der Relation $(R \div S)$, falls gilt:

*Für alle Tupel s aus S gehört das Tupel $(t * s)$ zu R .*

Division als Ausdruck der relationalen Algebra

Es gilt

$$(R \div S) = \pi_{A_{i_1}, \dots, A_{i_{m-n}}}(R) \setminus \pi_{A_{i_1}, \dots, A_{i_{m-n}}} \left(\pi_{A_1, \dots, A_m} (\pi_{A_{i_1}, \dots, A_{i_{m-n}}}(R) \times S) \setminus R \right)$$

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

$R \div S$	
A1	A2
1	2

Division als Inverses des kartesischen Produkts

Als Vereinfachung nehmen wir für das folgende Lemma 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 $(A_1, \dots, A_m, B_1, \dots, B_n)$ auffassen.

Lemma

Seien

- $\mathcal{R} = (A_1, \dots, A_m)$ und $\mathcal{S} = (B_1, \dots, B_n)$ zwei Relationenschemata ohne gemeinsame Attribute,
- R eine Instanz von \mathcal{R} und S eine Instanz von \mathcal{S} .

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 $(7/3) * 3 = 7$.

FilmDB

Wir betrachten eine Film-Datenbank. Dazu verwenden wir die Attribute

PId Personen-Id,	Dt Datum,
Fn Familienname	Reg Regisseur,
Vn Vorname,	Titel und
FId Film-Id,	Jahr

Wir betrachten die Relationenschemata und Instanzen

$\mathcal{P} = (\underline{\text{PId}}, \text{Fn}, \text{Vn})$ mit Instanz Personen ,
 $\mathcal{PF} = (\underline{\text{PId}}, \underline{\text{FId}}, \underline{\text{Dt}})$ mit Instanz Schaut ,
 $\mathcal{F} = (\underline{\text{FId}}, \text{Reg}, \text{Titel}, \text{Jahr})$ mit Instanz Filme .

FilmDB: Query 1

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

FilmDB: Query 1

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

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

FilmDB: Query 2

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

FilmDB: Query 2

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

$$\pi_{F_n, V_n}(\text{Personen} \bowtie (\text{Schaut} \bowtie (\sigma_{\text{Reg}='Spielberg' \vee \text{Reg}='Coppola'}(\text{Filme}))))$$

FilmDB: Query 3

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

FilmDB: Query 3

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

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

FilmDB: Query 4

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

FilmDB: Query 4

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

$$\pi_{\text{PIId,FIId}}(\text{Schaut}) \div \pi_{\text{FIId}}(\sigma_{\text{Reg}=\text{'Spielberg'}}(\text{Filme}))$$

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

$$\pi_{\text{PIId,FIId}}(\text{Schaut})$$

verwendet wird und nicht die ursprüngliche Relation Schaut. Die Abfrage

$$\pi_{\text{PIId}}(\text{Schaut} \div \pi_{\text{FIId}}(\sigma_{\text{Reg}=\text{'Spielberg'}}(\text{Filme})))$$

liefert nämlich die Personnummer 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

Seien

- $\{B_1, \dots, B_n\} \subseteq \{A_1, \dots, A_m\}$ Attributmengen,
- $C \in \{A_1, \dots, A_m\} \setminus \{B_1, \dots, B_n\}$,
- $\mathcal{S} = (A_1, \dots, A_m)$ ein Relationenschema,
- R eine Instanz von \mathcal{S} und
- **agg** eine Aggregatsfunktion.

Wir definieren die *GroupBy* Operation γ durch:

$$\gamma(R, (B_1, \dots, B_n), \mathbf{agg}, C) :=$$

$$\{(b_1, \dots, b_n, a) \mid (b_1, \dots, b_n) \in \pi_{B_1, \dots, B_n}(R) \text{ und} \\ a = \mathbf{agg}(\{x \mid (b_1, \dots, b_n, x) \in \pi_{B_1, \dots, B_n, C}(R)\})\}.$$

Bemerkung zu Group By

Bemerkung

Als Schema dieser Relation verwenden wir:

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

FilmDB: Query 5

Welcher Regisseur hat in welchem Jahr wie viele Filme gedreht?

FilmDB: Query 5

Welcher Regisseur hat in welchem Jahr wie viele Filme gedreht?

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

FilmDB: Query 6

In welchem Jahr hat Spielberg das letzte Mal Regie geführt?

FilmDB: Query 6

In welchem Jahr hat Spielberg das letzte Mal Regie geführt?

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

FilmDB: Query 7—Aggregatsfunktion über Menge

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

Welches ist die durchschnittliche Dauer der Filme pro Jahr?

FilmDB: Query 7—Aggregatsfunktion über Menge

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

Welches ist die durchschnittliche Dauer der Filme pro Jahr?

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

Jahr	avg(Dauer)
2010	110
2012	105

Im Detail

Für das Jahr 2010 wird die durchschnittliche Dauer berechnet durch

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

was das korrekte Resultat liefert.

Für das Jahr 2012 wird die durchschnittliche Dauer jedoch berechnet durch

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

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

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

analog zu

$$\gamma(R, \{B_1, \dots, B_n\}, \mathbf{agg}, C)$$

mit dem Unterschied, dass wir den Input der Aggregatsfunktion **agg** als Multimenge behandeln.

FilmDB: Query 7—Aggregatsfunktion über Multimenge

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

Welches ist die durchschnittliche Dauer der Filme pro Jahr?

FilmDB: Query 7—Aggregatsfunktion über Multimenge

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

Welches ist die durchschnittliche Dauer der Filme pro Jahr?

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

Jahr	avg(Dauer)
2010	110
2012	110