Kapitel DB:V

V. Grundlagen relationaler Anfragesprachen

Anfragen und Änderungen

Ausgangspunkt: Basisrelationen, die in der Datenbank gespeichert sind.

Ziel: abgeleitete Relationen, die aus Basisrelationen berechnet werden.

Ableitung von Relationen mit drei unterschiedlichen Mechanismen:

1. Anfrage

2. Sicht

3. Snapshot

Anfragen und Änderungen

Ausgangspunkt: Basisrelationen, die in der Datenbank gespeichert sind.

Ziel: abgeleitete Relationen, die aus Basisrelationen berechnet werden.

Ableitung von Relationen mit drei unterschiedlichen Mechanismen:

1. Anfrage

   Folge von Operationen, die aus Basisrelationen eine Ergebnisrelation berechnet. Die Ergebnisrelation kann angezeigt und interaktiv oder durch ein Programm weiterverarbeitet werden.

2. Sicht

   Folge von Operationen, die unter einem Sichtnamen *langfristig* gespeichert und unter diesem Namen wieder aufgerufen werden kann (Sichtrelation).

3. Snapshot

   Ergebnisrelation einer Anfrage, die unter einem Snapshot-Namen abgelegt wird, aber nie ein zweites Mal (mit geänderten Basisrelationen) berechnet wird. Beispiel: Erstellung einer Jahresbilanz.

Bemerkungen:

- Bei der Ableitung von Relationen bleiben die Basisrelationen unverändert.
- Update- und Änderungsoperationen verändern die Basisrelationen.
- Die Einbettung der Anfragesprache in eine Programmiersprache ermöglicht eine integrierte Weiterverarbeitung.

Anfragen und Änderungen

Eigenschaften relationaler Anfragesprachen [Heuer/Scholl 1991]

- ad-hoc-formulierbar
- deklarativ
- mengenbasiert
- abgeschlossen
- orthogonal
- adäquat

Anfragen und Änderungen

Eigenschaften relationaler Anfragesprachen [Heuer/Scholl 1991]

- ad-hoc-formulierbar

  Man kann Anfragen formulieren, ohne ein Programm dafür zu schreiben.

- deklarativ

  Man formuliert im deklarativen Stil „Was will ich haben?“ – nicht prozedural „Wie programmiere ich das, was ich haben will?“

- mengenbasiert

  Die Operationen arbeiten auf Datenmengen – nicht navigierend auf einzelnen Elementen.

- abgeschlossen

  Das Ergebnis einer Anfrage ist wieder vom Typ eines Operands (= Relation) und direkt als Eingabe für weitere Anfragen verwendbar.

- orthogonal

  Die Operationen sind ohne Einschränkung kombinierbar.

- adäquat

  Die Charakteristika des unterliegenden Datenmodells werden unterstützt.

Anfragen und Änderungen

Eigenschaften relationaler Anfragesprachen (Fortsetzung)

- vollständig (hinsichtlich eines Kalküls)
- optimierbar
- effizient
- sicher
- spezialisiert

Anfragen und Änderungen

Eigenschaften relationaler Anfragesprachen (Fortsetzung)

- vollständig (hinsichtlich eines Kalküls)

  Die Anfragesprache bildet (mindestens) die Relationenalgebra oder den sicheren Relationenkalkül ab.

- optimierbar

  Die Anfragesprache umfasst wenige Operationen, für die es leistungsfähige Optimierungsregeln gibt.

- effizient

  Die Anfragen sind effizient ausführbar.

- sicher

  Keine syntaktisch korrekte Anfrage gerät in eine Endlosschleife oder liefert ein unendliches Ergebnis.

- spezialisiert

  Die Anfragesprache ist keine vollständige Programmiersprache. Diese Eigenschaft folgt aus Optimierbarkeit, Effizienz, Sicherheit.

Bemerkungen:

- Beispiel für Orthogonalität: an jeder Stelle, an der ein Basisrelationenname stehen kann, darf auch eine Anfrage stehen (die wiederum eine Relation zurück liefert).
- Orthogonalität ist in SQL-89 u.a. deshalb nicht erfüllt, weil in der From-Klausel keine Anfrage stehen darf.
- Beispiel für Optimierbarkeit: Auswertung von Select-Operationen vor Join-Operationen.
- Beispiel für Effizienz: im Relationenmodell ist jede Operation in $O(n^2)$, mit n = Anzahl der Tupel einer Relation.

Relationale Algebra

Eine Algebra $\mathcal{A} = \langle M, \Omega \rangle$ besteht aus

1. einem Grundbereich M sowie
2. einer Menge von Operationen Ω mit $\circ : M^n \to M, \ \circ \in \Omega$.

Relationale Algebra

Eine Algebra $\mathcal{A} = \langle M, \Omega \rangle$ besteht aus

1. einem Grundbereich M sowie
2. einer Menge von Operationen Ω mit $\circ : M^n \to M$, $\circ \in \Omega$.

Bezogen auf die Relationenalgebra:

1. M ist die Menge aller Relationen über den Relationenschemata, die zu einer festen Menge von Attributen gebildet werden können.
2. Ω ist eine Menge von Operationen auf Relationen.

Folgende Operationen sind Teil der Relationenalgebra:

- einstellige Operatoren: Selektion, Projektion, Umbenennung
- Mengenoperationen: Vereinigung, Durchschnitt, Differenz
- Kartesisches Produkt
- Verbundoperationen: natürlicher-, allgemeiner-, äußerer-, semi-Verbund
- relationale Division

Relationale Algebra

Einstellige Operationen: Selektion

Syntax: $\sigma_{<\text{COND}>}(r)$

Semantik: Auswahl derjenigen Tupel in der Relation $r(\mathcal{R})$, die das Selektionsprädikat $<\text{COND}>$ erfüllen:

$$\{t \mid (t \in r) \ \wedge \ (<\text{COND}>(t) = \text{TRUE})\}$$

Relationale Algebra

Einstellige Operationen: Selektion

Syntax: $\sigma_{<\text{COND}>}(r)$

Semantik: Auswahl derjenigen Tupel in der Relation $r(\mathcal{R})$, die das Selektionsprädikat $<\text{COND}>$ erfüllen:

$$\{t \mid (t \in r) \ \land \ (<\text{COND}>(t) = \text{TRUE})\}$$

$<\text{COND}>$ ist aus folgenden Elementen aufgebaut:

1. Operanden: Attributnamen aus dem Schema $\mathcal{R}$, Konstanten
2. arithmetische Vergleichsoperatoren: $=, <, \leq, >, \geq, \neq$
3. logische Operatoren: $\land, \lor, \neg$

Relationale Algebra

Einstellige Operationen: Selektion

Syntax: $\sigma_{<\text{COND}>}(r)$

Semantik: Auswahl derjenigen Tupel in der Relation $r(\mathcal{R})$, die das Selektionsprädikat $<\text{COND}>$ erfüllen:

$$\{t \mid (t \in r) \;\wedge\; (<\text{COND}>(t) = \text{TRUE})\}$$

$<\text{COND}>$ ist aus folgenden Elementen aufgebaut:

1. Operanden: Attributnamen aus dem Schema $\mathcal{R}$, Konstanten
2. arithmetische Vergleichsoperatoren: $=, <, \leq, >, \geq, \neq$
3. logische Operatoren: $\wedge, \vee, \neg$

Beispiel:

Ausleihe	
InvNr	Name
4711	Meyer
1201	Schulz
0007	Müller
4712	Meyer

$\sigma_{\text{Name} \leq \text{'N'}}(\text{Ausleihe}) \quad \rightsquigarrow$

InvNr	Name
4711	Meyer
0007	Müller
4712	Meyer

Bemerkungen:

- Unmittelbar aufeinander folgende Selektionen lassen sich in ihrer Reihenfolge vertauschen, ohne dass sich die Ergebnisrelation ändert.

  Die Hintereinanderausführung von Selektionen besitzt eine konjunktive Semantik: es werden nur diejenigen Tupel berücksichtigt, die in der Schnittmenge aller selektierten Tupelmengen liegen. Die Schnittmengenbildung ist assoziativ.

- Zur Bezeichnung einer Relation können r und $r(\mathcal{R})$ gleichermaßen verwendet werden – abhängig davon, ob auf das Relationenschema $\mathcal{R}$ zu Unterscheidungszwecken Bezug genommen werden muss.

Relationale Algebra

Einstellige Operationen: Projektion

Syntax: $\pi_\alpha(r)$

Semantik: Projektion aller Tupel in r bzgl. der Attribute in α : $\{t(\alpha) \mid t \in r\}$

Relationale Algebra

Einstellige Operationen: Projektion

Syntax: $\pi_{\alpha}(r)$

Semantik: Projektion aller Tupel in r bzgl. der Attribute in α : $\{t(\alpha) \mid t \in r\}$

Beispiel:

Buch			
InvNr	Titel	ISBN	Autor
0007	Dr. No	3-125	James Bond
1201	Objektbanken	3-111	Heuer
4711	Datenbanken	3-765	Vossen
4712	Datenbanken	3-891	Ullman
4717	Pascal	3-999	Wirth

$\pi_{\text{InvNr,ISBN}}(\text{Buch}) \rightsquigarrow$

InvNr	ISBN
0007	3-125
1201	3-111
4711	3-765
4712	3-891
4717	3-999

Bemerkungen:

- Unmittelbar aufeinander folgende Projektionen lassen sich in ihrer Reihenfolge vertauschen, ohne dass sich die Ergebnisrelation ändert.

  Die Hintereinanderausführung von Projektionen besitzt eine konjunktive Semantik: es werden nur diejenigen Attribute berücksichtigt, die in der Schnittmenge aller projizierten Attributmengen liegen. Die Schnittmengenbildung ist assoziativ.

- Kombination von Selektion (<COND>) und Projektion (α). Bilden die Attribute in <COND> eine Teilmenge der Attribute einer *nachfolgenden* Projektion, lassen sich Selektion und Projektion vertauschen. Falls nicht, muss die Selektion zuerst ausgeführt werden, um sicherzustellen, dass <COND> definiert ist.

- Gilt $\alpha \subseteq \beta \subseteq \gamma$, so ist die unmittelbare Hintereinanderausführung der Projektionen bzgl. dieser Attributmengen äquivalent zu der alleinigen Anwendung der Projektion bzgl. α :
  $\pi_\alpha \pi_\beta \pi_\gamma(r) \equiv \pi_\alpha(r)$

- α ist eine Menge; entsprechend müsste man z.B. bei $\alpha = \{\text{InvNr}, \text{ISBN}\}$ die Selektion π_α als $\pi_{\{\text{InvNr,ISBN}\}}$ notieren. Die Notation von Attributnamen ohne Mengenklammern im Index ist formal unsauber, hat sich aber in der Datenbankliteratur durchgesetzt.

Relationale Algebra

Einstellige Operationen: Umbenennung

Syntax (a): $\rho_{\text{<Mapping>}}(r(\mathcal{R}))$, mit $\text{<Mapping>} = \{A_2 \leftarrow A_1 \mid A_1 \in \mathcal{R}, A_2 \notin \mathcal{R}\}$

Semantik: Umbenennung von Attribut A_1 zu A_2 in der Ergebnisrelation.

Relationale Algebra

Einstellige Operationen: Umbenennung

Syntax (a): $\rho_{\text{<Mapping>}}(r(\mathcal{R}))$, mit $\text{<Mapping>} = \{A_2 \leftarrow A_1 \mid A_1 \in \mathcal{R}, A_2 \notin \mathcal{R}\}$

Semantik: Umbenennung von Attribut A_1 zu A_2 in der Ergebnisrelation.

Syntax (b): $\rho_s(r)$

Semantik: Umbenennung der Relation r zu s

Relationale Algebra

Einstellige Operationen: Umbenennung

Syntax (a): $\rho_{\text{<Mapping>}}(r(\mathcal{R}))$, mit $\text{<Mapping>} = \{A_2 \leftarrow A_1 \mid A_1 \in \mathcal{R}, A_2 \notin \mathcal{R}\}$

Semantik: Umbenennung von Attribut A_1 zu A_2 in der Ergebnisrelation.

Syntax (b): $\rho_s(r)$

Semantik: Umbenennung der Relation r zu s

Beispiel:

zu (a) $\rho_{\text{Buchtitel} \leftarrow \text{Titel}}(\text{Buch})$

zu (b) $\rho_{\text{Dokument}}(\text{Buch})$

Bemerkungen:

- Mit der Umbenennung kann man fehlende Voraussetzungen zur Anwendung von Mengenoperationen schaffen.
- Die Umbenennung ermöglicht natürliche Verbünde, wo ansonsten kartesische Produkte entstünden: verschieden benannte Attribute werden gleich benannt.
- Die Umbenennung ermöglicht kartesische Produkte, wo ansonsten natürliche Verbünde entstünden: gleiche Attribute werden verschieden benannt.
- Beispiel: Auswertung von Abhängigkeiten zwischen Vorlesungen.

  $\mathcal{R} = \text{voraussetzen} = \{\text{Nachfolgervorlesung}, \text{Vorgaengervorlesung}\}$.

  Bestimmung der *Vor*vorgängervorlesungen von Vorlesung 4711:

$$
\begin{array}{l}
\pi_{\text{v2.Vorgaenger}}\big(\\
\quad \sigma_{\text{v1.Nachfolger}=4711\ \wedge\ \text{v1.Vorgaenger}=\text{v2.Nachfolger}}\big(\\
\quad\quad \rho_{\text{v1}}(\text{voraussetzen}) \times \rho_{\text{v2}}(\text{voraussetzen}) \\
\quad \big) \\
\big)
\end{array}
$$

Relationale Algebra

Mengenoperationen: Vereinigung, Durchschnitt, Differenz

Syntax: $r_1 \cup r_2, \; r_1 \cap r_2, \; r_1 - r_2$

Semantik:

$$= \{t \mid t \in r_1 \vee t \in r_2\}$$

$$= \{t \mid t \in r_1 \wedge t \in r_2\}$$

$$= \{t \mid t \in r_1 \wedge t \notin r_2\}$$

Mengenoperationen sind nur für Relationen mit gleichem Schema – d. h., Schemata mit gleichen Attributnamen und Domänen – definiert.

Relationale Algebra

Mengenoperationen: Vereinigung, Durchschnitt, Differenz

Syntax: $r_1 \cup r_2,\ \ r_1 \cap r_2,\ \ r_1 - r_2$

Semantik:
$$
\begin{aligned}
r_1 \ \cup \ r_2 &= \{t \mid t \in r_1 \vee t \in r_2\} \\
&= \{t \mid t \in r_1 \wedge t \in r_2\} \\
&= \{t \mid t \in r_1 \wedge t \notin r_2\}
\end{aligned}
$$

Mengenoperationen sind nur für Relationen mit gleichem Schema – d. h., Schemata mit gleichen Attributnamen und Domänen – definiert.

Beispiel:

Buch

Autor
James Bond
Heuer
Vossen
Ullman
Wirth

Book

Author
Witt
Vossen
Silberschatz
Meier
Wirth

Buch $\cup\ \rho_{\text{Autor}\leftarrow\text{Author}}$(Book) $\rightsquigarrow$

Autor
James Bond
Heuer
Vossen
Ullman
Wirth
Witt
Silberschatz
Meier

Relationale Algebra

Mengenoperationen: Vereinigung, Durchschnitt, Differenz

Syntax: $r_1 \cup r_2,\;\; r_1 \cap r_2,\;\; r_1 - r_2$

Semantik:

$$r_1 \;\cup\; r_2 = \{t \mid t \in r_1 \vee t \in r_2\}$$

$$r_1 \;\cap\; r_2 = \{t \mid t \in r_1 \wedge t \in r_2\}$$

$$= \{t \mid t \in r_1 \wedge t \notin r_2\}$$

Mengenoperationen sind nur für Relationen mit gleichem Schema – d. h., Schemata mit gleichen Attributnamen und Domänen – definiert.

Beispiel:

Buch
Autor
James Bond
Heuer
Vossen
Ullman
Wirth

Book
Author
Witt
Vossen
Silberschatz
Meier
Wirth

Buch $\cap\; \rho_{\text{Autor} \leftarrow \text{Author}}(\text{Book}) \;\leadsto$

Autor
Vossen
Wirth

Relationale Algebra

Mengenoperationen: Vereinigung, Durchschnitt, Differenz

Syntax: $r_1 \cup r_2,\ r_1 \cap r_2,\ r_1 - r_2$

Semantik:

$$r_1 \cup r_2 = \{t \mid t \in r_1 \vee t \in r_2\}$$

$$r_1 \cap r_2 = \{t \mid t \in r_1 \wedge t \in r_2\}$$

$$r_1 - r_2 = \{t \mid t \in r_1 \wedge t \notin r_2\}$$

Mengenoperationen sind nur für Relationen mit gleichem Schema – d. h., Schemata mit gleichen Attributnamen und Domänen – definiert.

Beispiel:

Buch

Autor
James Bond
Heuer
Vossen
Ullman
Wirth

Book

Author
Witt
Vossen
Silberschatz
Meier
Wirth

Buch $- \rho_{\text{Autor} \leftarrow \text{Author}}(\text{Book}) \rightsquigarrow$

Autor
James Bond
Heuer
Ullman

Bemerkungen:

- Die Syntax $A - B$ (anstelle von $A \setminus B$) zur Notation der Mengendifferenz ist in der Relationenalgebra üblich.

Relationale Algebra

Kartesisches Produkt [natürlicher Verbund]

Syntax: $r_1(\mathcal{R}_1) \times r_2(\mathcal{R}_2)$

Semantik: Sei $\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset$. Bildung aller $|r_1| \cdot |r_2|$ Tupel über $\mathcal{R}_1 \cup \mathcal{R}_2$:

$$\{t \mid t \in r(\mathcal{R}_1 \cup \mathcal{R}_2) \ \wedge \ t(\mathcal{R}_1) \in r_1 \ \wedge \ t(\mathcal{R}_2) \in r_2\}$$

Relationale Algebra

Kartesisches Produkt [natürlicher Verbund]

Syntax: $r_1(\mathcal{R}_1) \times r_2(\mathcal{R}_2)$

Semantik: Sei $\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset$. Bildung aller $|r_1| \cdot |r_2|$ Tupel über $\mathcal{R}_1 \cup \mathcal{R}_2$:

$$\{t \mid t \in r(\mathcal{R}_1 \cup \mathcal{R}_2) \ \wedge \ t(\mathcal{R}_1) \in r_1 \ \wedge \ t(\mathcal{R}_2) \in r_2\}$$

Beispiel:

Buch
Autor
James Bond
Heuer
Vossen
Ullman
Wirth

Book
Author
Witt
Vossen
Silberschatz
Meier
Wirth

Buch $\times$ Book $\rightsquigarrow$

Autor	Author
James Bond	Witt
James Bond	Vossen
James Bond	Silberschatz
James Bond	Meier
James Bond	Wirth
Heuer	Witt
Heuer	Vossen
...	...

Bemerkungen:

- Bei gleichen Attributnamen in den beteiligten Relationenschemata wird eine eindeutige Benennung dadurch erzwungen, dass ein qualifizierender Attributbezeichner $r.A$ aus dem Namen der Relation r und dem Attributnamen A konstruiert wird.

- Das kartesische Produkt ist eine Operation, die quadratischen Platz (und folglich auch mindestens quadratische Rechenzeit) benötigt.

Relationale Algebra

Konstruktion von Ausdrücken

1. Jede Basisrelation ist ein relationaler Algebra-Ausdruck.
2. Seien E_1 und E_2 relationale Algebra-Ausdrücke *(Expressions)*, dann sind auch folgende Ausdrücke relationale Algebra-Ausdrücke:

   (a) $E_1 \cup E_2, \quad E_1 \cap E_2, \quad E_1 - E_2,$

   (b) $E_1 \times E_2$

   (c) $\sigma_{<\text{COND}>}(E_1),$

   (d) $\pi_\alpha(E_1)$

   (e) $\rho_{A_2 \leftarrow A_1}(E_1), \quad \rho_s(E_1),$

Relationale Algebra

Konstruktion von Ausdrücken

1. Jede Basisrelation ist ein relationaler Algebra-Ausdruck.
2. Seien E_1 und E_2 relationale Algebra-Ausdrücke *(Expressions)*, dann sind auch folgende Ausdrücke relationale Algebra-Ausdrücke:

   (a) $E_1 \cup E_2$, $E_1 \cap E_2$, $E_1 - E_2$, wobei E_1 und E_2 das gleiche Relationenschema besitzen müssen.

   (b) $E_1 \times E_2$

   (c) $\sigma_{<\text{COND}>}(E_1)$, wobei $<\mathrm{COND}>$ ein Prädikat über den Attributen des Relationenschemas von E_1 ist.

   (d) $\pi_\alpha(E_1)$ mit einer Attributliste α, deren Attribute in dem Relationenschema von E_1 vorkommen.

   (e) $\rho_{A_2 \leftarrow A_1}(E_1)$, $\rho_s(E_1)$, wobei A_1 ein Attributname in dem Relationenschema von E_1 ist und A_2 dort nicht als Attributname vorkommt.

Relationale Algebra

Konstruktion von Ausdrücken (Fortsetzung)

(a) Formulierung komplexer Operationen mittels Schachtelung:

$$\pi_{\text{Nachname,Gehalt}}(\sigma_{\text{AbteilungsNr}=5}(\text{Angestellte}))$$

(a) Formulierung komplexer Operationen mittels Schachtelung:

$$\pi_{\text{Nachname,Gehalt}}(\sigma_{\text{AbteilungsNr}=5}(\text{Angestellte}))$$

Angestellte				
Vorname	Nachname	Abteilung	AbteilungsNr	Gehalt

(a) Formulierung komplexer Operationen mittels Schachtelung:

Angestellte

Angestellte				
Vorname	Nachname	Abteilung	AbteilungsNr	Gehalt
...	...	...	...	...
Derk	Smith	Research	5	6000
Peter	Sotelo	Research	5	5000
Pam	Brin	Accounting	3	5500
...	...	...	...	...

Relationale Algebra

Konstruktion von Ausdrücken (Fortsetzung)

(a) Formulierung komplexer Operationen mittels Schachtelung:

$$\sigma_{\text{AbteilungsNr}=5}(\text{Angestellte})$$

Angestellte				
Vorname	Nachname	Abteilung	AbteilungsNr	Gehalt
...	...	...	...	...
Derk	Smith	Research	5	6000
Peter	Sotelo	Research	5	5000
Pam	Brin	Accounting	3	5500
...	...	...	...	...

(a) Formulierung komplexer Operationen mittels Schachtelung:

$$\pi_{\text{Nachname,Gehalt}}(\sigma_{\text{AbteilungsNr}=5}(\text{Angestellte}))$$

Angestellte				
Vorname	Nachname	Abteilung	AbteilungsNr	Gehalt
...	...	...	...	...
Derk	Smith	Research	5	6000
Peter	Sotelo	Research	5	5000
Pam	Brin	Accounting	3	5500
...	...	...	...	...

Relationale Algebra

Konstruktion von Ausdrücken (Fortsetzung)

(b) Formulierung komplexer Operationen mittels benannten Ergebnisrelationen:

Abt5_Angestellte $\leftarrow$ $\sigma_{\text{AbteilungsNr}=5}(\text{Angestellte})$

Abt5_Angestellte				
Vorname	Nachname	Abteilung	AbteilungsNr	Gehalt
Derk	Smith	Research	5	6000
Peter	Sotelo	Research	5	5000

Relationale Algebra

Konstruktion von Ausdrücken (Fortsetzung)

(b) Formulierung komplexer Operationen mittels benannten Ergebnisrelationen:

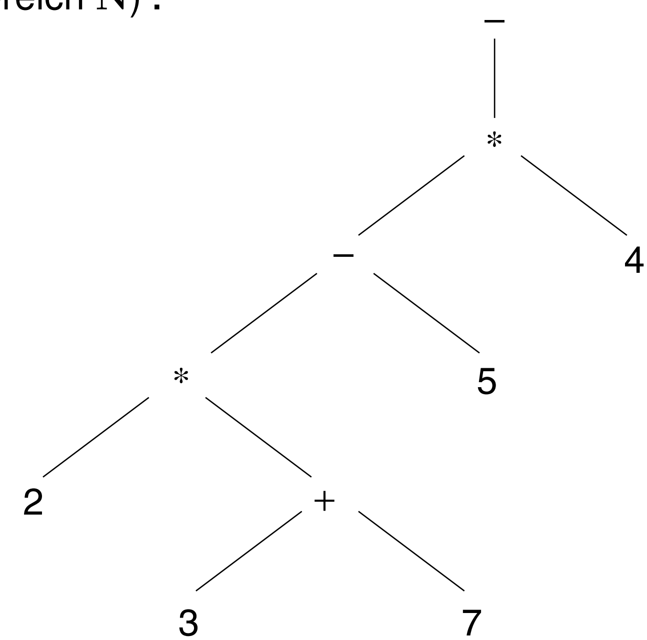
$$\text{Abt5\_Angestellte} \leftarrow \sigma_{\text{AbteilungsNr}=5}(\text{Angestellte})$$

Abt5_Angestellte				
Vorname	Nachname	Abteilung	AbteilungsNr	Gehalt
Derk	Smith	Research	5	6000
Peter	Sotelo	Research	5	5000

$$\text{Ergebnisrelation}(\text{Name}, \text{Einkommen}) \leftarrow \pi_{\text{Nachname},\text{Gehalt}}(\text{Abt5\_Angestellte})$$

Ergebnisrelation	
Name	Einkommen
Smith	6000
Sotelo	5000

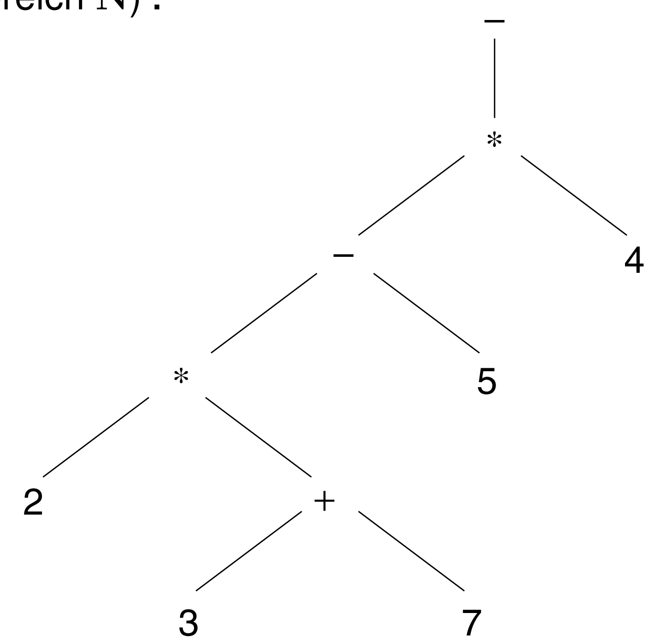
Baumdarstellung (Verknüpfungen über dem Grundbereich $\mathbf{N}$) :



Relationale Algebra

Konstruktion von Ausdrücken (Fortsetzung)

Baumdarstellung (Verknüpfungen über dem Grundbereich $\mathbf{N}$) :



Algebra:

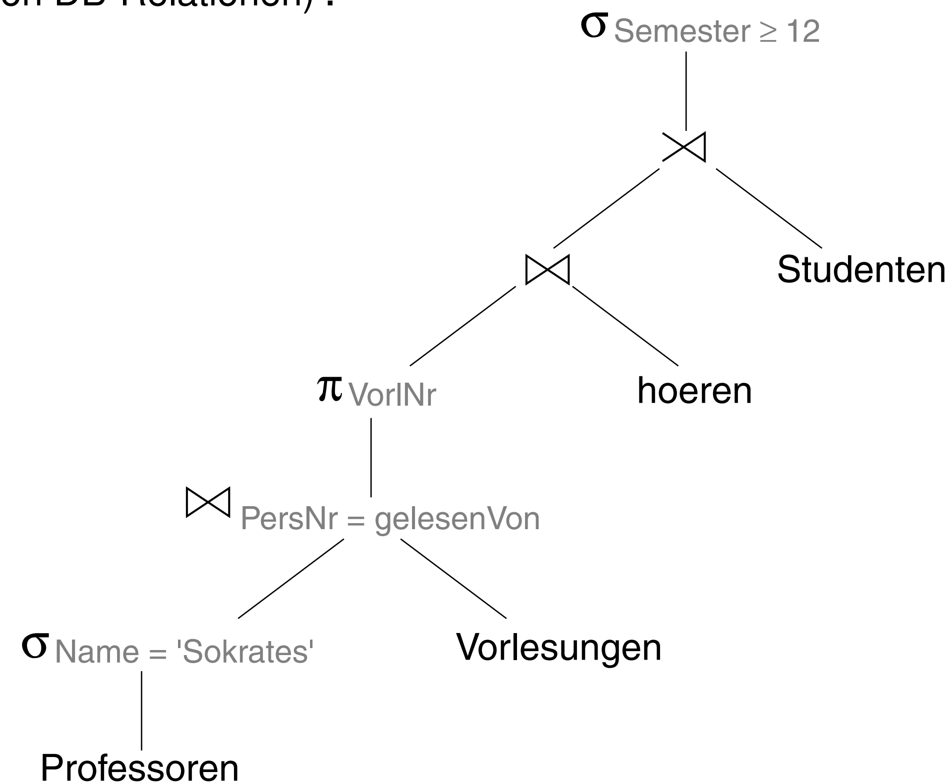
Infix-Schreibweise: `− ((2 * (3 + 7) − 5) * 4)`

Präfix-Schreibweise: `(− (* (− (* 2 (+ 3 7)) 5 ) 4))`

Relationale Algebra

Konstruktion von Ausdrücken (Fortsetzung)

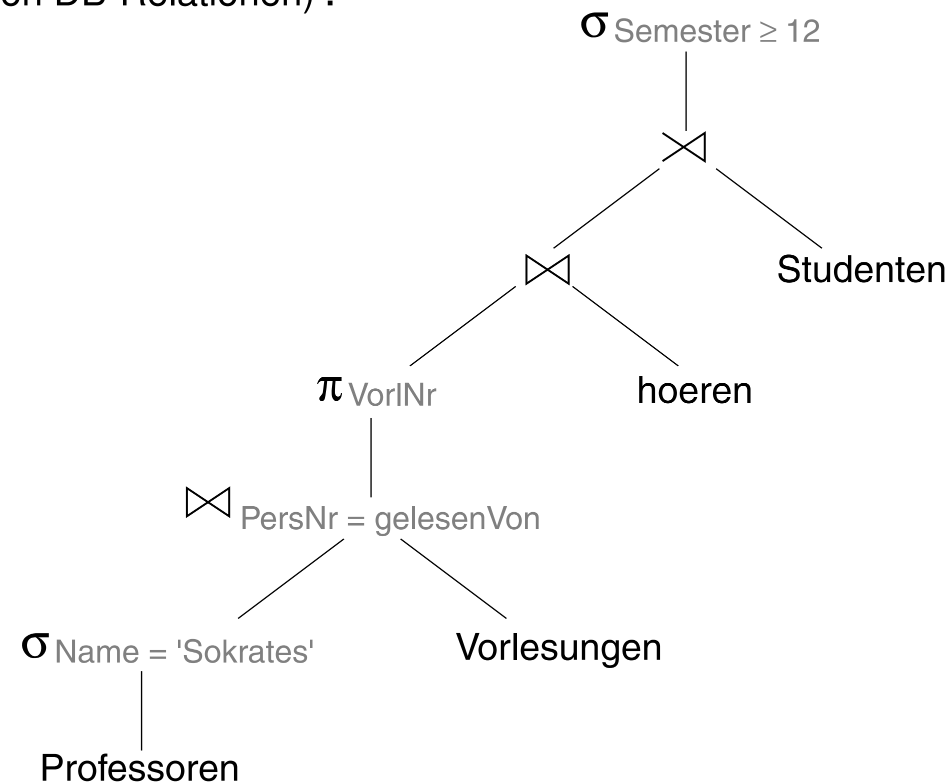
Baumdarstellung (Verknüpfungen von DB-Relationen) :



Relationale Algebra

Konstruktion von Ausdrücken (Fortsetzung)

Baumdarstellung (Verknüpfungen von DB-Relationen) :



Algebra:

$$\sigma_{\text{Semester}\geq 12}($$
$$((\pi_{\text{VorlNr}}(\sigma_{\text{Name}='\text{Sokrates}'}(\text{Professoren}) \bowtie_{\text{PersNr}=\text{gelesenVon}} \text{Vorlesungen})) \bowtie \text{hoeren})$$
$$\rtimes \text{Studenten})$$

Bemerkungen:

- Der relationale Algebra-Ausdruck spezifiziert die Dauerstudenten von Sokrates [Kemper/Eickler 2011] :
  „Diejenigen Studenten, die mindestens eine Vorlesung bei Sokrates gehört haben und sich im 12. Semester oder höher befinden.“

Relationale Algebra

Konstruktion von Ausdrücken (Fortsetzung)

Definition 1 (relational vollständig)

Eine Menge von Operationen Ω ist relational vollständig bezüglich einer anderen Menge von Operationen Ω', wenn mit Ω jede relationenalgebraische Operation ausgedrückt („simuliert“) werden kann, die sich mit Ω' ausdrücken lässt.

Konstruktion von Ausdrücken (Fortsetzung)

Definition 1 (relational vollständig)

Eine Menge von Operationen Ω ist relational vollständig bezüglich einer anderen Menge von Operationen Ω', wenn mit Ω jede relationenalgebraische Operation ausgedrückt („simuliert“) werden kann, die sich mit Ω' ausdrücken lässt.

Satz 2

Die Menge der Operationen $\Omega = \{\cup, -, \times, \sigma, \pi, \rho\}$ ist relational vollständig bezüglich der Menge $\Omega' = \{\cup, \cap, -, \times, \bowtie, \div, \sigma, \pi, \rho\}$ von Operationen der Relationenalgebra.

Insbesondere lassen sich ausdrücken:

- $r_1 \cap r_2$ durch $r_1 - (r_1 - r_2)$
- $\bowtie$ durch die Kombination von π, σ und $\times$
- $r_1 \div r_2$ durch $\pi_{(\mathcal{R}_1 - \mathcal{R}_2)}(r_1) - \pi_{(\mathcal{R}_1 - \mathcal{R}_2)}\Big(\big(\pi_{(\mathcal{R}_1 - \mathcal{R}_2)}(r_1) \times r_2\big) - r_1 \Big)$

Definition 1 (relational vollständig)

Eine Menge von Operationen Ω ist relational vollständig bezüglich einer anderen Menge von Operationen Ω', wenn mit Ω jede relationenalgebraische Operation ausgedrückt („simuliert“) werden kann, die sich mit Ω' ausdrücken lässt.

Satz 2

Die Menge der Operationen $\Omega = \{\cup, -, \times, \sigma, \pi, \rho\}$ ist relational vollständig bezüglich der Menge $\Omega' = \{\cup, \cap, -, \times, \bowtie, \div, \sigma, \pi, \rho\}$ von Operationen der Relationenalgebra.

Insbesondere lassen sich ausdrücken:

- $r_1 \cap r_2$ durch $r_1 - (r_1 - r_2)$
- $\bowtie$ durch die Kombination von π, σ und $\times$
- $r_1 \div r_2$ durch $\pi_{(\mathcal{R}_1 - \mathcal{R}_2)}(r_1) - \pi_{(\mathcal{R}_1 - \mathcal{R}_2)}\Big((\pi_{(\mathcal{R}_1 - \mathcal{R}_2)}(r_1) \times r_2) - r_1 \Big)$

Satz 3

Die Menge der Operationen Ω in Satz 2 ist unabhängig: keine Operation kann weggelassen werden, ohne die Vollständigkeit zu verlieren.

Bemerkungen:

- Die Beschränkung des Selektionsprädikates <COND> auf die einfache Attribut- und Konstantenselektion gefährdet nicht die Vollständigkeitseigenschaft von Ω in Satz 2. D.h., die booleschen Operatoren sind nicht notwendig.

Relationale Algebra

Weitere Operationen: natürlicher Verbund *(Natural-Join)* [kartesisches Produkt]

Syntax: $r_1(\mathcal{R}_1) \bowtie r_2(\mathcal{R}_2)$

Semantik: Alle Tupel, die bei Projektion auf $\mathcal{R}_i$ ein Tupel aus r_i, $i = 1, 2,$ liefern:

$$\{t \mid t \in r(\mathcal{R}_1 \cup \mathcal{R}_2) \;\wedge\; t(\mathcal{R}_1) \in r_1 \;\wedge\; t(\mathcal{R}_2) \in r_2\}$$

Relationale Algebra

Weitere Operationen: natürlicher Verbund *(Natural-Join)* [kartesisches Produkt]

Syntax: $r_1(\mathcal{R}_1) \bowtie r_2(\mathcal{R}_2)$

Semantik: Alle Tupel, die bei Projektion auf $\mathcal{R}_i$ ein Tupel aus r_i, $i = 1, 2$, liefern:

$$\{t \mid t \in r(\mathcal{R}_1 \cup \mathcal{R}_2) \ \wedge \ t(\mathcal{R}_1) \in r_1 \ \wedge \ t(\mathcal{R}_2) \in r_2\}$$

Beispiel:

Ausleihe	
InvNr	Name
4711	Meyer
1201	Schulz
0007	Müller
4712	Meyer

Buch			
InvNr	Titel	ISBN	Autor
0007	Dr. No	3-125	James Bond
1201	Objektbanken	3-111	Heuer
4711	Datenbanken	3-765	Vossen
4712	Datenbanken	3-891	Ullman
4717	Pascal	3-999	Wirth

Ausleihe $\bowtie$ Buch $\leadsto$

Name	InvNr	Titel	ISBN	Autor
Müller	0007	Dr. No	3-125	James Bond
Schulz	1201	Objektbanken	3-111	Heuer
Meyer	4711	Datenbanken	3-765	Vossen
Meyer	4712	Datenbanken	3-891	Ullman

Relationale Algebra

Weitere Operationen: natürlicher Verbund *(Natural-Join)* [kartesisches Produkt]

Syntax: $r_1(\mathcal{R}_1) \bowtie r_2(\mathcal{R}_2)$

Semantik: Alle Tupel, die bei Projektion auf $\mathcal{R}_i$ ein Tupel aus r_i, $i = 1, 2,$ liefern:

$$\{t \mid t \in r(\mathcal{R}_1 \cup \mathcal{R}_2) \ \wedge \ t(\mathcal{R}_1) \in r_1 \ \wedge \ t(\mathcal{R}_2) \in r_2\}$$

$$\overbrace{A_1, \ldots, A_n, B_1, \ldots, B_k}^{\mathcal{R}_1}$$

$$\underbrace{B_1, \ldots, B_k, C_1, \ldots, C_m}_{\mathcal{R}_2}$$

$$r_1(\mathcal{R}_1) \bowtie r_2(\mathcal{R}_2) \ = \ \underbrace{\pi_{A_1,\ldots,A_n,\ \mathcal{R}_1.B_1,\ldots,\mathcal{R}_1.B_k,\ C_1,\ldots,C_m}}_{\text{Projektion}} \ \big(\underbrace{\sigma_{\mathcal{R}_1.B_1=\mathcal{R}_2.B_1,\ldots,\mathcal{R}_1.B_k=\mathcal{R}_2.B_k}}_{\text{Selektion}} \ \underbrace{(r_1 \times r_2)}_{\text{kartesisches Produkt}}\big)$$

Relationale Algebra

Weitere Operationen: natürlicher Verbund *(Natural-Join)* [kartesisches Produkt]

Syntax: $r_1(\mathcal{R}_1) \bowtie r_2(\mathcal{R}_2)$

Semantik: Alle Tupel, die bei Projektion auf $\mathcal{R}_i$ ein Tupel aus r_i, $i = 1, 2$, liefern:

$$\{t \mid t \in r(\mathcal{R}_1 \cup \mathcal{R}_2) \ \wedge \ t(\mathcal{R}_1) \in r_1 \ \wedge \ t(\mathcal{R}_2) \in r_2\}$$

$$\overbrace{A_1, \ldots, A_n, B_1, \ldots, B_k}^{\mathcal{R}_1}$$

$$\underbrace{B_1, \ldots, B_k, C_1, \ldots, C_m}_{\mathcal{R}_2}$$

$$r_1(\mathcal{R}_1) \bowtie r_2(\mathcal{R}_2) \ = \ \underbrace{\pi_{A_1,\ldots,A_n,\ \mathcal{R}_1.B_1,\ldots,\mathcal{R}_1.B_k,\ C_1,\ldots,C_m}}_{\text{Projektion}} \big(\underbrace{\sigma_{\mathcal{R}_1.B_1=\mathcal{R}_2.B_1,\ldots,\mathcal{R}_1.B_k=\mathcal{R}_2.B_k}}_{\text{Selektion}} \ \underbrace{(r_1 \times r_2)}_{\text{kartesisches Produkt}}\big)$$

$r_1(\mathcal{R}_1) \bowtie r_2(\mathcal{R}_2)$		
$\mathcal{R}_1 - \mathcal{R}_2$	$\mathcal{R}_1 \cap \mathcal{R}_2$	$\mathcal{R}_2 - \mathcal{R}_1$

Relationale Algebra

Weitere Operationen: natürlicher Verbund (Fortsetzung)

Eigenschaften des natürlichen Verbunds:

- Kommutativität: $r_1 \bowtie r_2 \; = \; r_2 \bowtie r_1$
- Assoziativität: $(r_1 \bowtie r_2) \bowtie r_3 \; = \; r_1 \bowtie (r_2 \bowtie r_3)$
  Somit ist folgende Notation möglich: $r_1 \bowtie r_2 \bowtie \ldots \bowtie r_p \; \equiv \bowtie_{i=1}^{p} r_i$
- $\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset \quad \Rightarrow \quad r_1 \bowtie r_2 \; = \; r_1 \times r_2$

Relationale Algebra

Weitere Operationen: natürlicher Verbund (Fortsetzung)

Eigenschaften des natürlichen Verbunds:

- Kommutativität: $r_1 \bowtie r_2 \;=\; r_2 \bowtie r_1$
- Assoziativität: $(r_1 \bowtie r_2) \bowtie r_3 \;=\; r_1 \bowtie (r_2 \bowtie r_3)$
  Somit ist folgende Notation möglich: $r_1 \bowtie r_2 \bowtie \ldots \bowtie r_p \;\equiv \bowtie_{i=1}^{p} r_i$
- $\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset \quad \Rightarrow \quad r_1 \bowtie r_2 \;=\; r_1 \times r_2$

Beispiel „3-Wege-Join":

$$(\text{Ausleihe} \bowtie \text{Buch}) \bowtie \text{Verlag} \;=\; \text{Ausleihe} \bowtie (\text{Buch} \bowtie \text{Verlag})$$

Beispiel für die Entartung zum kartesischen Produkt:

$$\pi_{\text{InvNr}}(\text{Ausleihe}) \bowtie \pi_{\text{Autor}}(\text{Buch})$$

Bemerkungen:

- Die gemeinsamen Attribute der bei einem Join beteiligten Relationen werden auch als „Join-Attribute“ bezeichnet.
- Die Umbennungsoperation ρ ermöglicht es, Relationen über zwei Attribute zu verbinden, welche die gleiche Bedeutung (Semantik), aber einen unterschiedlichen Namen haben.
- Tupel, die keinen Join-Partner finden, sogenannte „Dangling Tuples“, werden eliminiert. Folglich ist die Projektion im Allgemeinen nicht die inverse Operation zum natürlichen Verbund. Es gilt: $\pi_{\mathcal{R}_1}(r_1 \bowtie r_2) \subseteq r_1$
- Der natürliche Verbund ist im Allgemeinen nicht die inverse Operation zu zwei Projektionen. Sei r eine Relation über $\mathcal{R}$ und mit $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$. Dann gilt der folgende Zusammenhang nur bei *Verbundtreue*: $\pi_{\mathcal{R}_1}(r) \bowtie \pi_{\mathcal{R}_2}(r) = r$

Relationale Algebra

Weitere Operationen: allgemeiner Verbund *(Theta-Join)*

Der allgemeine Theta-Join-Operator, $\bowtie_\theta$, erlaubt die Spezifikation eines beliebigen Join-Prädikates θ. Das Ergebnis des Theta-Joins enthält *alle* (bei Namensgleichheit: qualifizierten) Attribute der beteiligten Relationen:

$$r_1 \bowtie_\theta r_2 \ = \ \sigma_\theta(r_1 \times r_2)$$

Relationale Algebra

Weitere Operationen: allgemeiner Verbund *(Theta-Join)*

Der allgemeine Theta-Join-Operator, $\bowtie_\theta$, erlaubt die Spezifikation eines beliebigen Join-Prädikates θ. Das Ergebnis des Theta-Joins enthält *alle* (bei Namensgleichheit: qualifizierten) Attribute der beteiligten Relationen:

$$r_1 \bowtie_\theta r_2 \;=\; \sigma_\theta(r_1 \times r_2)$$

Beispiel:

$$r_1 \bowtie_{\underbrace{A>D \,\wedge\, B\leq E \,\wedge\, r_1.C=r_2.C}_{\theta}} r_2$$

r_1		
A	B	C
6	1	c_1
7	1	c_2
8	1	c_1

$\bowtie$

r_2		
C	D	E
c_1	3	2
c_2	4	2
c_2	5	2

$\rightsquigarrow$

$r_1 \bowtie_\theta r_2$					
A	B	$\mathcal{R}_1$.C	$\mathcal{R}_2$.C	D	E
6	1	c_1	c_1	3	2
7	1	c_2	c_2	4	2
7	1	c_2	c_2	5	2
8	1	c_1	c_1	3	2

Bemerkungen:

- Einen Theta-Join der Form $r_1 \bowtie_{A_1=B_1,\ldots,A_k=B_k} r_2$ nennt man auch Equi-Join. Im Unterschied zum Natural-Join werden beim Equi-Join alle Attribute übernommen.
- Die bislang eingeführten Join-Operatoren werden auch *innere* Joins genannt. Für sie gilt, dass diejenigen Tupel der Argumentrelationen verloren gehen, die keinen Join-Partner gefunden haben.
- Mit den *äußeren* Join-Operatoren können auch partnerlose Tupel der Argumentrelationen in die Ergebnisrelation übernommen werden: bei Anwendung des Left-Outer-Join bleiben die Tupel der linken Argumentrelation immer erhalten, bei Anwendung des Right-Outer-Join die Tupel der rechten Argumentrelation. Die nicht gegebenen Attributwerte der partnerlosen Tupel werden mit Nullwerten, in Zeichen: $\perp$, aufgefüllt.

Relationale Algebra

Weitere Operationen: äußerer Verbund *(Outer-Join)*

- Natural-Join:

r_1		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

$\bowtie$

r_2		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

$\rightsquigarrow$

$r_1 \bowtie r_2$				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1

- Left-Outer-Join:

r_1		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

$⟕$

r_2		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

$\rightsquigarrow$

$r_1 ⟕ r_2$				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
a_2	b_2	c_2	$\perp$	$\perp$

Relationale Algebra

Weitere Operationen: äußerer Verbund *(Outer-Join)*

- Natural-Join:

r_1		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

$\bowtie$

r_2		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

$\leadsto$

$r_1 \bowtie r_2$				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1

- Left-Outer-Join:

r_1		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

$⟕$

r_2		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

$\leadsto$

$r_1 ⟕ r_2$				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
a_2	b_2	c_2	$\perp$	$\perp$

- Right-Outer-Join:

r_1		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

$⟖$

r_2		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

$\leadsto$

$r_1 ⟖ r_2$				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
$\perp$	$\perp$	c_3	d_2	e_2

Relationale Algebra

Weitere Operationen: äußerer Verbund *(Outer-Join)*

- Natural-Join:

r_1		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

$\bowtie$

r_2		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

$\leadsto$

$r_1 \bowtie r_2$				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1

- Left-Outer-Join:

r_1		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

⟕

r_2		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

$\leadsto$

r_1 ⟕ r_2				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
a_2	b_2	c_2	$\perp$	$\perp$

- Right-Outer-Join:

r_1		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

⟖

r_2		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

$\leadsto$

r_1 ⟖ r_2				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
$\perp$	$\perp$	c_3	d_2	e_2

- Full-Outer-Join:

r_1		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

⟗

r_2		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

$\leadsto$

r_1 ⟗ r_2				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
a_2	b_2	c_2	$\perp$	$\perp$
$\perp$	$\perp$	c_3	d_2	e_2

Relationale Algebra

Weitere Operationen: Semi-Verbund *(Semi-Join)*

Die Semi-Verbundoperatoren projizieren die Tupel der Ergebnisrelation eines Natural-Join auf das Schema einer der Ausgangsrelationen:

$$r_1 \ltimes r_2 = \pi_{\mathcal{R}_1}(r_1 \bowtie r_2) \quad \text{bzw.} \quad r_1 \rtimes r_2 = \pi_{\mathcal{R}_2}(r_1 \bowtie r_2)$$

Weitere Operationen: Semi-Verbund *(Semi-Join)*

Die Semi-Verbundoperatoren projizieren die Tupel der Ergebnisrelation eines Natural-Join auf das Schema einer der Ausgangsrelationen:

$$r_1 \ltimes r_2 = \pi_{\mathcal{R}_1}(r_1 \bowtie r_2) \quad \text{bzw.} \quad r_1 \rtimes r_2 = \pi_{\mathcal{R}_2}(r_1 \bowtie r_2)$$

- Semi-Join von r_1 mit r_2:

r_1

A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

$\ltimes$

r_2

C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

$\leadsto$

$r_1 \ltimes r_2$

A	B	C
a_1	b_1	c_1

- Semi-Join von r_2 mit r_1:

r_1

A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

$\rtimes$

r_2

C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

$\leadsto$

$r_1 \rtimes r_2$

C	D	E
c_1	d_1	e_1

Es gilt folgende Identität: $r_1 \rtimes r_2 = r_2 \ltimes r_1$

Relationale Algebra

Weitere Operationen: relationale Division

Die bisher betrachteten Anfragebeispiele liefern diejenigen Tupel, die eine bestimmte Selektionsbedingung erfüllen.

Frage: Wie bestimmt man diejenigen Tupel, die *alle* Bedingungen – im Sinne von *gleichzeitig* – einer Menge von Selektionsbedingungen erfüllen?

Relationale Algebra

Weitere Operationen: relationale Division

Die bisher betrachteten Anfragebeispiele liefern diejenigen Tupel, die eine bestimmte Selektionsbedingung erfüllen.

Frage: Wie bestimmt man diejenigen Tupel, die *alle* Bedingungen – im Sinne von *gleichzeitig* – einer Menge von Selektionsbedingungen erfüllen?

Beispiel:

Buecher	
Titel	Verlag
Harry Potter	Princeton
Heuristics	Addison
Glücksformel	dpunkt
Datenbanken	Springer

Buchhaendler		
Name	Stadt	PLZ
Lehmann	Berlin	99011
Meiersche	Aachen	42100
Amazon	Köln	52100

Angebote	
Titel	Haendler
Harry Potter	Lehmann
Harry Potter	Meiersche
Harry Potter	Amazon
Datenbanken	Amazon
Glücksformel	Amazon
Glücksformel	Lehmann

Anfragen

1. „Welche Titel sind bei *allen* Buchhändlern im Angebot?“
2. Nicht zu verwechseln mit „Welche Titel befinden sich (alle) im Angebot?“

Relationale Algebra

Weitere Operationen: relationale Division (Fortsetzung)

Syntax: $\underbrace{r_1(\mathcal{R}_1)}_{\text{Dividend}} \div \underbrace{r_2(\mathcal{R}_2)}_{\text{Divisor}}$

Semantik: Sei $\mathcal{R}_2 \subseteq \mathcal{R}_1$. Dann ist $r_1 \div r_2$ definiert als:

$$\{t \mid \forall t_2 \in r_2 \;\; \exists t_1 \in r_1 \;\; : \;\; t \in t_1(\mathcal{R}_1 - \mathcal{R}_2) \;\wedge\; t_1(\mathcal{R}_2) = t_2\}$$

Beispiel:

r_1	
A	B
a_1	b_1
a_1	b_2
a_1	b_3
a_2	b_2
a_2	b_3

$\div$

r_2
B
b_1
b_2

$=$

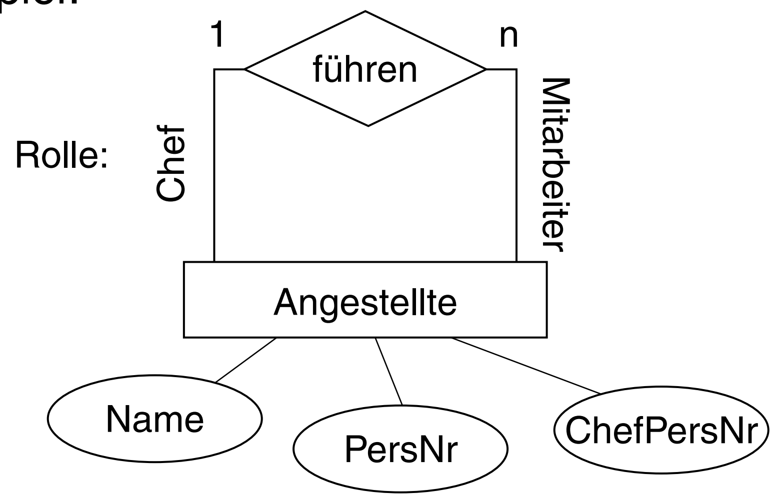
$r_1 \div r_2$
A
a_1

Relationale Algebra

Rekursiver Abschluss

Der rekursive Abschluss kann mit Mitteln der relationalen Algebra nicht ausgedrückt werden.

Beispiel:

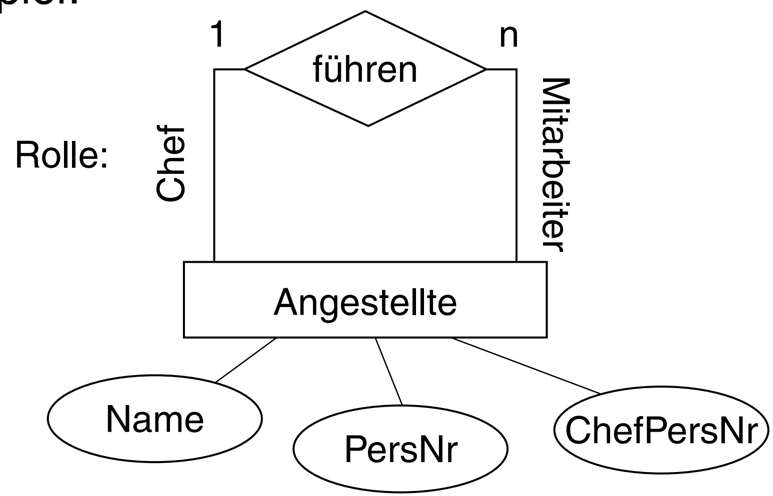


Relationale Algebra

Rekursiver Abschluss

Der rekursive Abschluss kann mit Mitteln der relationalen Algebra nicht ausgedrückt werden.

Beispiel:



Angestellte		
Name	PersNr	ChefPersNr
Franklin	333	888
Smith	123	333
Zelaja	999	987
Ramesh	666	333
Wallace	987	888
Borg	888	$\perp$
Jabbar	456	987

Anfrage (rekursiv)

„Liefere alle direkten und indirekten Untergebenen von Borg."

Relationenalgebra

$\rightsquigarrow$ $\mathcal{TAFEL}$

Bemerkungen:

- Ansatz zur Auflösung der Rekursion in der Beispielanfrage:
  1. Ausgehend von 'Borg', Bestimmung der Untergebenen in jeder Stufe – bzw. für soviel Stufen, für die angefragt ist.
  2. Vereinigung der Teilergebnisse aller Stufen.

Relationale Algebra

Übersicht über Operationen

Operation	Argumente	Notation
SELECT	Relation r, Auswahlbedingung <COND>	$\sigma_{\text{<COND>}}(r)$
PROJECT	Relation r, Attributliste α	$\pi_{\alpha}(r)$
RENAME	Relation r, Attributzuordnungen <Mapping> Relation r, Relationenname s	$\rho_{\text{<Mapping>}}(r)$ $\rho_s(r)$
UNION	Relationen r_1, r_2	$r_1 \cup r_2$
INTERSECTION	Relationen r_1, r_2	$r_1 \cap r_2$
DIFFERENCE	Relationen r_1, r_2	$r_1 - r_2$
CARTESIAN PRODUCT	Relationen r_1, r_2	$r_1 \times r_2$
NATURAL JOIN	Relationen r_1, r_2	$r_1 \bowtie r_2$
THETA JOIN	Relationen r_1, r_2, Verbundbedingung θ	$r_1 \bowtie_{\theta} r_2$
EQUI JOIN	Relationen r_1, r_2, $=$-Verbundbedingung θ	$r_1 \bowtie_{\theta} r_2$
OUTER JOIN (LEFT/RIGHT/FULL)	Relationen r_1, r_2	$r_1 \leftouterjoin r_2$, $r_1 \rightouterjoin r_2$, $r_1 \fullouterjoin r_2$
SEMI JOIN (LEFT/RIGHT)	Relationen r_1, r_2	$r_1 \ltimes r_2$, $r_1 \rtimes r_2$
DIVISION	Relationen r_1, r_2	$r_1 \div r_2$

Relationale Algebra

Übersicht über Operationen

Operation	Argumente	Notation
SELECT	Relation r, Auswahlbedingung <COND>	$\sigma_{\text{<COND>}}(r)$
PROJECT	Relation r, Attributliste α	$\pi_\alpha(r)$
RENAME	Relation r, Attributzuordnungen <Mapping> Relation r, Relationenname s	$\rho_{\text{<Mapping>}}(r)$ $\rho_s(r)$
UNION	Relationen r_1, r_2	$r_1 \cup r_2$
INTERSECTION	Relationen r_1, r_2	$r_1 \cap r_2$
DIFFERENCE	Relationen r_1, r_2	$r_1 - r_2$
CARTESIAN PRODUCT	Relationen r_1, r_2	$r_1 \times r_2$
NATURAL JOIN	Relationen r_1, r_2	$r_1 \bowtie r_2$
THETA JOIN	Relationen r_1, r_2, Verbundbedingung θ	$r_1 \bowtie_\theta r_2$
EQUI JOIN	Relationen r_1, r_2, $=$-Verbundbedingung θ	$r_1 \bowtie_\theta r_2$
OUTER JOIN (LEFT/RIGHT/FULL)	Relationen r_1, r_2	$r_1 ⟕ r_2$, $r_1 ⟖ r_2$, $r_1 ⟗ r_2$
SEMI JOIN (LEFT/RIGHT)	Relationen r_1, r_2	$r_1 \ltimes r_2$, $r_1 \rtimes r_2$
DIVISION	Relationen r_1, r_2	$r_1 \div r_2$

Relationale Algebra

Übersicht über Operationen

Operation	Argumente	Notation
SELECT	Relation r, Auswahlbedingung <COND>	$\sigma_{\text{<COND>}}(r)$
PROJECT	Relation r, Attributliste α	$\pi_{\alpha}(r)$
RENAME	Relation r, Attributzuordnungen <Mapping> Relation r, Relationenname s	$\rho_{\text{<Mapping>}}(r)$ $\rho_s(r)$
UNION	Relationen r_1, r_2	$r_1 \cup r_2$
INTERSECTION	Relationen r_1, r_2	$r_1 \cap r_2$
DIFFERENCE	Relationen r_1, r_2	$r_1 - r_2$
CARTESIAN PRODUCT	Relationen r_1, r_2	$r_1 \times r_2$
NATURAL JOIN	Relationen r_1, r_2	$r_1 \bowtie r_2$
THETA JOIN	Relationen r_1, r_2, Verbundbedingung θ	$r_1 \bowtie_\theta r_2$
EQUI JOIN	Relationen r_1, r_2, $=$-Verbundbedingung θ	$r_1 \bowtie_\theta r_2$
OUTER JOIN (LEFT/RIGHT/FULL)	Relationen r_1, r_2	$r_1 \mathbin{⟕} r_2$, $r_1 \mathbin{⟖} r_2$, $r_1 \mathbin{⟗} r_2$
SEMI JOIN (LEFT/RIGHT)	Relationen r_1, r_2	$r_1 \ltimes r_2$, $r_1 \rtimes r_2$
DIVISION	Relationen r_1, r_2	$r_1 \div r_2$

Relationale Algebra

Übersicht über Operationen

Operation	Argumente	Notation
SELECT	Relation r, Auswahlbedingung <COND>	$\sigma_{\text{<COND>}}(r)$
PROJECT	Relation r, Attributliste α	$\pi_\alpha(r)$
RENAME	Relation r, Attributzuordnungen <Mapping> Relation r, Relationenname s	$\rho_{\text{<Mapping>}}(r)$ $\rho_s(r)$
UNION	Relationen r_1, r_2	$r_1 \cup r_2$
INTERSECTION	Relationen r_1, r_2	$r_1 \cap r_2$
DIFFERENCE	Relationen r_1, r_2	$r_1 - r_2$
CARTESIAN PRODUCT	Relationen r_1, r_2	$r_1 \times r_2$
NATURAL JOIN	Relationen r_1, r_2	$r_1 \bowtie r_2$
THETA JOIN	Relationen r_1, r_2, Verbundbedingung θ	$r_1 \bowtie_\theta r_2$
EQUI JOIN	Relationen r_1, r_2, $=$-Verbundbedingung θ	$r_1 \bowtie_\theta r_2$
OUTER JOIN (LEFT/RIGHT/FULL)	Relationen r_1, r_2	$r_1 ⟕ r_2$, $r_1 ⟖ r_2$, $r_1 ⟗ r_2$
SEMI JOIN (LEFT/RIGHT)	Relationen r_1, r_2	$r_1 \ltimes r_2$, $r_1 \rtimes r_2$
DIVISION	Relationen r_1, r_2	$r_1 \div r_2$