

Kapitel DB: V (Fortsetzung)

- I. Einführung und grundlegende Konzepte von Datenbanken
- II. Datenbankentwurf und Datenbankmodelle
- III. Konzeptueller Datenbankentwurf
- IV. Logischer Datenbankentwurf mit dem relationalen Modell
- V. Grundlagen relationaler Anfragesprachen
 - Anfragen und Änderungen
 - Relationale Algebra
 - Anfragekalküle
 - Relationaler Tupelkalkül
 - Relationaler Domänenkalkül
- VI. Die relationale Datenbanksprache SQL
- VII. Entwurfstheorie relationaler Datenbanken

Anfragekalküle

Paradigmen

- Anfrage**algebren** spiegeln das Konzept von abstrakten Datenstrukturen wider; der Datentyp ist die Relation mit entsprechenden Operationen hierauf. Ein relationaler Ausdruck ist eine *prozedurale* Beschreibung, also eine genau festgelegte Folge von Operationen zur Berechnung einer Anfrage.
- Anfrage**kalküle** sind ein logikbasierter Ansatz zur Beschreibung der Ergebnismenge einer Anfrage.
Sie können als *deklarative* bzw. *nicht-prozedurale* Sprache aufgefasst werden. Insbesondere enthält eine Formel des Kalküls keine Information darüber, wie sie auszuwerten ist.

Für das relationale Modell betrachtet man folgende Kalküle:

1. relationaler Tupelkalkül
2. relationaler Domänenkalkül, auch Bereichskalkül genannt

Anfragekalküle

Aufbau einer Formel / Sprache / Syntax (Teil I)

Sei Σ eine Menge von Atomen aus einem Anfragekalkül, dann sind folgende Ausdrücke Formeln in diesem Kalkül:

1. Jedes Atom in Σ ist eine Formel.
2. Sind α und β Formeln, so sind es auch (α) , $\neg\alpha$, $\alpha \wedge \beta$ und $\alpha \vee \beta$.

Bewertung einer Formel / Interpretation der Sprache / Semantik (Teil I)

1. Hinsichtlich eines Datenbankzustandes $d(\mathcal{R}) = \{r_1, \dots, r_p\}$ kann jedem Atom in Σ ein Wahrheitswert zugewiesen werden.
2. Auf Basis der Wahrheitswerte der Atome lässt sich rekursiv gemäß der üblichen Semantik für \neg, \wedge, \vee jeder Formel α ein Wahrheitswert zuordnen.

Bemerkungen:

- ❑ Ein Atom ist die einfachste Aussage, die formuliert werden kann. Von einer Aussage lässt sich feststellen, ob sie wahr oder falsch ist.
- ❑ Ein Atom in einem Anfragekalkül entspricht einem Test, ob ein Wertetupel oder ein einzelner Wert ein Element in einer Relation r ist. Das heißt, auf Basis von r wird der Wahrheitswert eines Atoms bestimmt.
- ❑ Der Wahrheitswert einer komplexen Formel leitet sich in eindeutiger Weise von den Wahrheitswerten der Atome der Formel ab.
- ❑ Mit Formeln werden die Bedingungen in einer Anfrage nachgebildet.

Anfragekalküle

Freie und gebundene Variablen

Sei α eine Formel, die eine Variable x enthält. Dann sei vereinbart:

- (a) Ist α ein Atom, so ist x eine *freie* Variable.
- (b) Das Vorkommen von x in (α) , $\neg\alpha$, $\alpha \wedge \beta$ und $\alpha \vee \beta$ ist *frei* oder *gebunden* – abhängig davon, ob es in α frei oder gebunden ist.
- (c) Alle freien Vorkommen von x in α sind gebunden in $\exists x\alpha$ und $\forall x\alpha$.
- (d) In keiner Formel darf eine Variable sowohl frei als auch gebunden auftreten.

Anfragekalküle

Freie und gebundene Variablen

Sei α eine Formel, die eine Variable x enthält. Dann sei vereinbart:

- (a) Ist α ein Atom, so ist x eine *freie* Variable.
- (b) Das Vorkommen von x in (α) , $\neg\alpha$, $\alpha \wedge \beta$ und $\alpha \vee \beta$ ist *frei* oder *gebunden* – abhängig davon, ob es in α frei oder gebunden ist.
- (c) Alle freien Vorkommen von x in α sind gebunden in $\exists x\alpha$ und $\forall x\alpha$.
- (d) In keiner Formel darf eine Variable sowohl frei als auch gebunden auftreten.

Aufbau einer Formel / Sprache / Syntax (Teil II)

- 3. Ist α eine Formel, so sind es auch $\exists x\alpha$ und $\forall x\alpha$ – wobei x eine Variable ist, die in α frei vorkommt.

Bewertung einer Formel / Interpretation der Sprache / Semantik (Teil II)

- 3. Eine Formel $\exists x\alpha$ ist wahr, falls α bzgl. einer Instanziierung von x wahr wird.
Eine Formel $\forall x\alpha$ ist wahr, falls α bzgl. aller möglichen Instanziierungen von x wahr wird.

Anfragekalküle

Auswertung einer Anfrage

Gegeben:

- Anfrage $\{(_)\mid\alpha\}$ mit freien Variablen $(_)$ und Formel α
- Datenbankzustand $d(\mathcal{R}) = \{r_1, \dots, r_p\}$

Konstruktion der Ergebnisrelation res für Anfrage $\{(_)\mid\alpha\}$ unter $d(\mathcal{R})$:

1. $res = \emptyset$
2. Die freien Variablen $(_)$ werden hinsichtlich aller Tupel bzw. Attributwerte der in der Datenbank befindlichen Relationen $\{r_1, \dots, r_p\}$ instanziiert.
3. Für jede Instanziierung von $(_)$ wird geprüft, ob die Formel α erfüllbar ist. Falls ja, setze $res = res \cup \{(_)\}$
4. res enthält keine weiteren Elemente.

Anfragekalküle

Sichere Anfragen

- Unter (semantisch) sicheren Anfragen versteht man Formeln eines Anfragekalküls, die für jeden Datenbankzustand $d(\mathcal{R}) = \{r_1, \dots, r_p\}$ nur für eine endliche Menge von Variableninstanziierungen erfüllt sind.
- Durch die Forderung bestimmter syntaktischer Einschränkungen kann man die semantische Sicherheit für eine *Teilmenge* der semantisch sicheren Anfragen auf einfache Art bestimmen.

Bemerkungen:

- Syntaktische Einschränkungen sind z. B. endliche Wertebereiche von Variablen oder die eingeschränkte Verwendung von Quantoren.
- Semantische Sicherheit ist eine Eigenschaft, die im Einzelfall leicht zu zeigen sein kann, die aber in der Allgemeinheit nicht automatisch nachprüfbar ist. Die Ursache dafür liegt in der Unentscheidbarkeit der Prädikatenlogik erster Stufe mit Arithmetik.
- Beispiel für eine nicht-sichere Anfrage:
„Alle Tupel des Universums (= Datenbank), die nicht in r_1 sind.“ bzw. $\neg(t_1 \in r_1)$

Relationaler Tupelkalkül

Konzepte (vgl. Konzepte im Domänenkalkül)

1. Tupelvariablen, die sich auf **Relationen** $r_i \in d(\mathcal{R})$, $d(\mathcal{R}) = \{r_1, \dots, r_p\}$, beziehen und mit jedem Tupel aus r_i instanziiert werden können.
2. Formeln, mit denen auf Basis der Tupelvariablen Zusammenhänge zwischen Attributen formuliert werden können.

Relationaler Tupelkalkül

Anfragen (vgl. Anfragen im Domänenkalkül)

Anfrage im relationalen Tupelkalkül mit Variablen $t_1, t_2, \dots, t_n, t_{n+1}, \dots, t_{n+m}$:

$$\{t \mid \alpha\} \quad \text{allgemein : } \{(t_1.A_1, t_2.A_2, \dots, t_n.A_n) \mid \alpha\}$$

- t_1, \dots, t_n sind *freie*, t_{n+1}, \dots, t_{n+m} sind *gebundene* Tupelvariablen.
- A_1, \dots, A_n sind Attribute der Relationen bzgl. derer die t_i instanziiert sind.
- α ist eine logische Formel, wobei die Menge der Atome Σ , aus denen α besteht, wie folgt definiert ist:

Relationaler Tupelkalkül

Anfragen (vgl. Anfragen im Domänenkalkül)

Anfrage im relationalen Tupelkalkül mit Variablen $t_1, t_2, \dots, t_n, t_{n+1}, \dots, t_{n+m}$:

$$\{t \mid \alpha\} \quad \text{allgemein : } \{(t_1.A_1, t_2.A_2, \dots, t_n.A_n) \mid \alpha\}$$

- t_1, \dots, t_n sind *freie*, t_{n+1}, \dots, t_{n+m} sind *gebundene* Tupelvariablen.
- A_1, \dots, A_n sind Attribute der Relationen bzgl. derer die t_i instanziiert sind.
- α ist eine logische Formel, wobei die Menge der Atome Σ , aus denen α besteht, wie folgt definiert ist:
 1. „ $t_i \in r_i$ “ alternativ: „ $r_i(t_i)$ “
ist ein Atom, wobei t_i eine Tupelvariable und r_i eine Relation bezeichnet. r_i definiert die Werte, die t_i annehmen muss, damit $r_i(t_i)$ wahr ist.

Relationaler Tupelkalkül

Anfragen (vgl. Anfragen im Domänenkalkül)

Anfrage im relationalen Tupelkalkül mit Variablen $t_1, t_2, \dots, t_n, t_{n+1}, \dots, t_{n+m}$:

$$\{t \mid \alpha\} \quad \text{allgemein : } \{(t_1.A_1, t_2.A_2, \dots, t_n.A_n) \mid \alpha\}$$

- t_1, \dots, t_n sind *freie*, t_{n+1}, \dots, t_{n+m} sind *gebundene* Tupelvariablen.
- A_1, \dots, A_n sind Attribute der Relationen bzgl. derer die t_i instanziiert sind.
- α ist eine logische Formel, wobei die Menge der Atome Σ , aus denen α besteht, wie folgt definiert ist:
 1. „ $t_i \in r_i$ “ alternativ: „ $r_i(t_i)$ “
ist ein Atom, wobei t_i eine Tupelvariable und r_i eine Relation bezeichnet. r_i definiert die Werte, die t_i annehmen muss, damit $r_i(t_i)$ wahr ist.
 2. „ $t_i.A_i \text{ op } t_j.A_j$ “
ist ein Atom mit $op \in \{=, <, \leq, >, \geq, \neq\}$. t_i, t_j bezeichnen Tupelvariablen und A_i, A_j bezeichnen Attribute aus den Relationen hinsichtlich derer t_i bzw. t_j instanziiert sind.
 3. „ $t_i.A_i \text{ op } c$ “
ist ein Atom mit $op \in \{=, <, \leq, >, \geq, \neq\}$. t_i bezeichnet eine Tupelvariable, A_i ein Attribut aus der Relation hinsichtlich der t_i instanziiert ist und $c \in \text{dom}(A_i)$ ist eine Konstante.

Relationaler Tupelkalkül

Beispiel 1

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

Anfrage

„Liefere Name und Wohnort der Mitarbeiter, die in der Forschung arbeiten.“

Relationaler Tupelkalkül

Beispiel 1

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

Anfrage

„Liefere Name und Wohnort der Mitarbeiter, die in der Forschung arbeiten.“

Relationenalgebra

$\pi_{\text{Name, Wohnort}}(\text{Mitarbeiter} \bowtie_{\text{AbtNr=Nr}} (\sigma_{\text{AbtName='Forschung'}}(\text{Abteilung})))$

Tupelkalkül

$\{(t_1.\text{Name}, t_1.\text{Wohnort}) \mid$

$\text{Mitarbeiter}(t_1) \wedge \exists t_2(\text{Abteilung}(t_2) \wedge t_2.\text{AbtName} = \text{'Forschung'} \wedge t_2.\text{Nr} = t_1.\text{AbtNr})\}$

Bemerkungen:

- Eine Bedingung, die sich auf ein Attribut und eine Konstante bezieht, entspricht einer Selektion, σ , in der relationalen Algebra.
Beispiel: $t_2.\text{AbtName} = \text{'Forschung'}$
- Eine Bedingung bzgl. zweier Attribute, die sich auf Tupel aus verschiedenen Relationen beziehen, entspricht einem Verbund (Join), \bowtie , in der relationalen Algebra.
Beispiel: $t_2.\text{Nr} = t_1.\text{AbtNr}$

Relationaler Tupelkalkül

Beispiel 2

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

Anfrage

„Liefere für jedes Projekt in Weimar dessen Nummer, die Nummer der durchführenden Abteilung sowie Name und Wohnort des Abteilungsmanagers.“

Relationaler Tupelkalkül

Beispiel 2

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

Anfrage

„Liefere für jedes Projekt in Weimar dessen Nummer, die Nummer der durchführenden Abteilung sowie Name und Wohnort des Abteilungsmanagers.“

Relationenalgebra

$$\pi_{\text{PNr, AbtNr, Name, Wohnort}} \left(\left(\rho_{\text{PNr} \leftarrow \text{Nr}, \text{PName} \leftarrow \text{Name}} \left(\sigma_{\text{Ort} = \text{'Weimar'}}(\text{Projekt}) \right) \bowtie_{\text{AbtNr} = \text{Nr}} \text{Abteilung} \bowtie_{\text{Manager} = \text{PersNr}} \text{Mitarbeiter} \right) \right)$$

Tupelkalkül

$$\{ (t_1.\text{Name}, t_1.\text{Wohnort}, t_3.\text{Nr}, t_3.\text{AbtNr}) \mid \text{Mitarbeiter}(t_1) \wedge \text{Projekt}(t_3) \wedge t_3.\text{Ort} = \text{'Weimar'} \wedge \exists t_2 (\text{Abteilung}(t_2) \wedge t_2.\text{Nr} = t_3.\text{AbtNr} \wedge t_2.\text{Manager} = t_1.\text{PersNr}) \}$$

Relationaler Tupelkalkül

Beispiel 3

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

Anfrage

„Liefere die Namen der Mitarbeiter, die in *allen* Projekten der Abteilung 5 arbeiten.“

Relationaler Tupelkalkül

Beispiel 3

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

Anfrage

„Liefere die Namen der Mitarbeiter, die in *allen* Projekten der Abteilung 5 arbeiten.“

Relationenalgebra

$$\pi_{\text{Name}} ((\text{ArbeitetInProjekt} \div \rho_{\text{ProjektNr} \leftarrow \text{Nr}}(\pi_{\text{Nr}}(\sigma_{\text{AbtNr}=5}(\text{Projekt})))) \bowtie \text{Mitarbeiter})$$

Tupelkalkül

$$\{(t_1.\text{Name}) \mid \text{Mitarbeiter}(t_1) \wedge \\ \forall t_3 (\neg(\text{Projekt}(t_3) \wedge (t_3.\text{AbtNr} = 5))) \vee \text{oder: } \forall t_3 ((\text{Projekt}(t_3) \wedge (t_3.\text{AbtNr} = 5)) \rightarrow \\ \exists t_4 (\text{ArbeitetInProjekt}(t_4) \wedge t_4.\text{ProjektNr} = t_3.\text{Nr} \wedge t_4.\text{PersNr} = t_1.\text{PersNr}))\}$$

Bemerkungen:

- Zur Semantik des \forall -Quantors in der Formel $\alpha = \text{Mitarbeiter}(t_1) \wedge \forall t_3 \beta$: „ α ist erfüllt für diejenigen Mitarbeiter (Tupel t_1), bei denen für *alle* Tupel t_3 die Teilformel β erfüllt ist.“
Beachte, dass t_3 an *alle* Tupel des Universums bzw. der Datenbank $d(\mathcal{R})$ gebunden wird und bzgl. *aller* möglichen Instanziierungen die Formel β erfüllen muss.
- Die Quantoren können verschoben werden, solange sich keine Variablenbindungen ändern und die Ordnung zwischen den \exists - und \forall -Quantoren erhalten bleibt:

$$\{(t_1.\text{Name}) \mid \text{Mitarbeiter}(t_1) \wedge \forall t_3 \exists t_4 (\neg(\text{Projekt}(t_3) \wedge (t_3.\text{AbtNr} = 5)) \vee (\text{ArbeitetInProjekt}(t_4) \wedge t_4.\text{ProjektNr} = t_3.\text{Nr} \wedge t_4.\text{PersNr} = t_1.\text{PersNr}))\}$$

- Bei (formalen, logischen, natürlichen) Sprachen unterscheidet man zwischen Sätzen aus der Sprache selbst und der Formulierung von Zusammenhängen *über* solche Sätze.
- Sätze aus der Sprache selbst dienen uns zur Kommunikation mittels dieser Sprache; die Symbole, die verwendet werden, um solche Sätze zu formulieren, gehören zur Objektsprache. Symbole, die verwendet werden, um *über* Sätze zu sprechen, die in der Objektsprache formuliert sind, gehören zur Metasprache.
- Die Formelbezeichner α , β und γ sowie die Junktoren \wedge , \vee und \neg gehören zur Objektsprache. Das \approx -Zeichen ist ein Zeichen der Metasprache und steht für „ist logisch äquivalent mit“. Es gelten folgende Zusammenhänge:

$$\alpha \rightarrow \beta \approx \neg \alpha \vee \beta$$

$$\neg(\alpha \wedge \beta) \approx \neg \alpha \vee \neg \beta$$

$$\Rightarrow (\alpha \wedge \beta) \rightarrow \gamma \approx \neg \alpha \vee \neg \beta \vee \gamma$$

Relationaler Tupelkalkül

Sichere Ausdrücke im Tupelkalkül

Folgende Anfrage liefert eine unendliche Zahl von Ergebnissen:

$$\{t \mid \neg \text{Mitarbeiter}(t)\}$$

Definition 14 (Domäne)

Der Bereich bzw. die Domäne einer Formel α ist die Menge aller Konstanten in α vereinigt mit der Menge aller Attributwerte der Relationen $r, r \in \alpha$.

Relationaler Tupelkalkül

Sichere Ausdrücke im Tupelkalkül

Folgende Anfrage liefert eine unendliche Zahl von Ergebnissen:

$$\{t \mid \neg \text{Mitarbeiter}(t)\}$$

Definition 14 (Domäne)

Der Bereich bzw. die Domäne einer Formel α ist die Menge aller Konstanten in α vereinigt mit der Menge aller Attributwerte der Relationen r , $r \in \alpha$.

Ein Ausdruck des Tupelkalküls ist sicher, wenn das Ergebnis des Ausdrucks eine Teilmenge der Domäne ist. Für sichere Kalkülausdrücke ist garantiert, dass ihr Ergebnis endlich ist.

Beispiel:

Die Domäne von „ $\neg \text{Mitarbeiter}(t)$ “ :

{Smith, Wong, Zelaya, Weimar, Köln, Erfurt, 4, 5, 1234, 3334, 8886, 9876, 9998}

Relationaler Domänenkalkül

Konzepte ([vgl. Konzepte im Tupelkalkül](#))

1. Domänenvariablen, die sich auf die **Attribute**, A , in den Relationen beziehen und mit jedem Wert aus dem Wertebereich $dom(A)$ von A instanziiert werden können.
2. Formeln, mit denen auf Basis der Domänenvariablen Zusammenhänge zwischen Attributen formuliert werden können.

Bemerkungen:

- ❑ SQL basiert auf dem relationalen Tupelkalkül und wurde von IBM-Research, San Jose, Kalifornien, entwickelt.
- ❑ QBE (Query-By-Example) basiert auf dem relationalen Domänenkalkül und wurde von IBM-Research, Yorktown Heights, New York, entwickelt. Diese Entwicklung fand fast zeitgleich mit der Entwicklung von SQL in San Jose statt.
- ❑ QBE war eine der ersten graphischen Anfragesprachen für Datenbanksysteme und ist bei IBM als Interface-Option für DB2 erhältlich.

Relationaler Domänenkalkül

Anfragen (vgl. Anfragen im Tupelkalkül)

Anfrage im relationalen Domänenkalkül mit Variablen $x_1, \dots, x_n, x_{n+1}, \dots, x_{n+m}$:

$$\{(x_1, x_2, \dots, x_n) \mid \alpha\}$$

- x_1, \dots, x_n sind *freie*, x_{n+1}, \dots, x_{n+m} sind *gebundene* Domänenvariablen.
- α ist eine logische Formel, wobei die Menge der Atome Σ , aus denen α besteht, wie folgt definiert ist:

Relationaler Domänenkalkül

Anfragen (vgl. Anfragen im Tupelkalkül)

Anfrage im relationalen Domänenkalkül mit Variablen $x_1, \dots, x_n, x_{n+1}, \dots, x_{n+m}$:

$$\{(x_1, x_2, \dots, x_n) \mid \alpha\}$$

- x_1, \dots, x_n sind *freie*, x_{n+1}, \dots, x_{n+m} sind *gebundene* Domänenvariablen.
- α ist eine logische Formel, wobei die Menge der Atome Σ , aus denen α besteht, wie folgt definiert ist:
 1. „ $(x_{r_1}, x_{r_2}, \dots, x_{r_k}) \in r$ “ alternativ: „ $r(x_{r_1}, x_{r_2}, \dots, x_{r_k})$ “
ist ein Atom, wobei die x_{r_i} Domänenvariablen und r eine Relation über k Attribute bezeichnet. $r(x_{r_1}, x_{r_2}, \dots, x_{r_k})$ ist wahr für eine Instanziierung von $(x_{r_1}, x_{r_2}, \dots, x_{r_k})$, falls diese Instanziierung ein Tupel in r ist.

Relationaler Domänenkalkül

Anfragen (vgl. Anfragen im Tupelkalkül)

Anfrage im relationalen Domänenkalkül mit Variablen $x_1, \dots, x_n, x_{n+1}, \dots, x_{n+m}$:

$$\{(x_1, x_2, \dots, x_n) \mid \alpha\}$$

- x_1, \dots, x_n sind *freie*, x_{n+1}, \dots, x_{n+m} sind *gebundene* Domänenvariablen.
- α ist eine logische Formel, wobei die Menge der Atome Σ , aus denen α besteht, wie folgt definiert ist:
 1. „ $(x_{r_1}, x_{r_2}, \dots, x_{r_k}) \in r$ “ alternativ: „ $r(x_{r_1}, x_{r_2}, \dots, x_{r_k})$ “
ist ein Atom, wobei die x_{r_i} Domänenvariablen und r eine Relation über k Attribute bezeichnet. $r(x_{r_1}, x_{r_2}, \dots, x_{r_k})$ ist wahr für eine Instanziierung von $(x_{r_1}, x_{r_2}, \dots, x_{r_k})$, falls diese Instanziierung ein Tupel in r ist.
 2. „ $x_i \text{ op } x_j$ “
ist ein Atom mit $\text{op} \in \{=, <, \leq, >, \geq, \neq\}$. x_i, x_j bezeichnen Domänenvariablen, die über den Wertebereichen der zugeordneten Attribute instanziiert sind.
 3. „ $x_i \text{ op } c$ “
ist ein Atom mit $\text{op} \in \{=, <, \leq, >, \geq, \neq\}$. x_i bezeichnet eine Domänenvariable, die über dem Wertebereich des zugeordneten Attributes instanziiert ist, und $c \in \text{dom}(A_i)$ ist eine Konstante aus dem gleichen Wertebereich.

Relationaler Domänenkalkül

Beispiel 1

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

Anfrage

„Liefere Name und Wohnort der Mitarbeiter, die in der Forschung arbeiten.“

Relationaler Domänenkalkül

Beispiel 1

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

Anfrage

„Liefere Name und Wohnort der Mitarbeiter, die in der Forschung arbeiten.“

Relationenalgebra

$\pi_{\text{Name, Wohnort}}(\text{Mitarbeiter} \bowtie_{\text{AbtNr=Nr}} (\sigma_{\text{AbtName='Forschung'}}(\text{Abteilung})))$

Domänenkalkül

$\{(x_1, x_3) \mid \exists x_2 \exists x_4 \exists x_5 \exists y_1 \exists y_2 \exists y_3$

$(\text{Mitarbeiter}(x_1, x_2, x_3, x_4, x_5) \wedge \text{Abteilung}(y_1, y_2, y_3) \wedge y_1 = \text{'Forschung'} \wedge y_2 = x_5)\}$

Relationaler Domänenkalkül

Beispiel 1

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

Anfrage

„Liefere Name und Wohnort der Mitarbeiter, die in der Forschung arbeiten.“

Relationenalgebra

$\pi_{\text{Name, Wohnort}}(\text{Mitarbeiter} \bowtie_{\text{AbtNr=Nr}} (\sigma_{\text{AbtName='Forschung'}}(\text{Abteilung})))$

Domänenkalkül Konvention: gebundene Variablen sind per Default \exists -quantifiziert.

$\{(x_1, x_3) \mid \exists x_5 \exists y_1 \exists y_2$

$(\text{Mitarbeiter}(x_1, x_2, x_3, x_4, x_5) \wedge \text{Abteilung}(y_1, y_2, y_3) \wedge y_1 = \text{'Forschung'} \wedge y_2 = x_5)\}$

Relationaler Domänenkalkül

Beispiel 1

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

Anfrage

„Liefere Name und Wohnort der Mitarbeiter, die in der Forschung arbeiten.“

Relationenalgebra

$\pi_{\text{Name, Wohnort}}(\text{Mitarbeiter} \bowtie_{\text{AbtNr=Nr}} (\sigma_{\text{AbtName='Forschung'}}(\text{Abteilung})))$

Domänenkalkül Abkürzung: Konstanten als Parameter.

$\{(x_1, x_3) \mid \exists x_5 \exists y_2$

$(\text{Mitarbeiter}(x_1, x_2, x_3, x_4, x_5) \wedge \text{Abteilung}('Forschung', y_2, y_3) \wedge y_2 = x_5)\}$

Relationaler Domänenkalkül

Beispiel 1

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

Anfrage

„Liefere Name und Wohnort der Mitarbeiter, die in der Forschung arbeiten.“

Relationenalgebra

$\pi_{\text{Name, Wohnort}}(\text{Mitarbeiter} \bowtie_{\text{AbtNr=Nr}} (\sigma_{\text{AbtName='Forschung'}}(\text{Abteilung})))$

Domänenkalkül Abkürzung: Unifikation von Domänenvariablen

$\{(x_1, x_3) \mid \exists x_5$

$(\text{Mitarbeiter}(x_1, x_2, x_3, x_4, x_5) \wedge \text{Abteilung}('Forschung', x_5, y_3))\}$

Bemerkungen:

- Eine Bedingung, die sich auf eine Domänenvariable und eine Konstante bezieht, entspricht einer Selektion, σ , in der relationalen Algebra.

Beispiel: $y_1 = \text{'Forschung'}$

- Eine Bedingung bzgl. zweier Domänenvariablen, die sich auf zwei verschiedene Relationen beziehen, entspricht einem Verbund (Join), \bowtie , in der relationalen Algebra.

Beispiel: $x_5 = y_2$

Relationaler Domänenkalkül

Beispiel 2

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

Anfrage

„Liefere für jedes Projekt in Weimar dessen Nummer, die Nummer der durchführenden Abteilung sowie Name und Wohnort des Abteilungsmanagers.“

Relationaler Domänenkalkül

Beispiel 2

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

Anfrage

„Liefere für jedes Projekt in Weimar dessen Nummer, die Nummer der durchführenden Abteilung sowie Name und Wohnort des Abteilungsmanagers.“

Relationenalgebra

$$\pi_{\text{PNr,AbtNr, Name,Wohnort}} \left(\left(\rho_{\text{PNr} \leftarrow \text{Nr}, \text{PName} \leftarrow \text{Name}} \left(\sigma_{\text{Ort} = \text{'Weimar'}}(\text{Projekt}) \right) \bowtie_{\text{AbtNr} = \text{Nr}} \text{Abteilung} \bowtie_{\text{Manager} = \text{PersNr}} \text{Mitarbeiter} \right) \right)$$

Domänenkalkül

$$\{(x_1, x_3, z_2, z_4) \mid \exists x_2 \exists x_4 \exists x_5 \exists y_1 \exists y_2 \exists y_3 \exists z_1 \exists z_3 \\ (\text{Mitarbeiter}(x_1, x_2, x_3, x_4, x_5) \wedge \text{Abteilung}(y_1, y_2, y_3) \wedge \text{Projekt}(z_1, z_2, z_3, z_4) \wedge \\ z_3 = \text{'Weimar'} \wedge z_4 = y_2 \wedge y_3 = x_2)\}$$

Relationaler Domänenkalkül

Beispiel 3

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

Anfrage

„Liefere die Namen der Mitarbeiter, die in *allen* Projekten der Abteilung 5 arbeiten.“

Relationaler Domänenkalkül

Beispiel 3

Mitarbeiter				
Name	PersNr	Wohnort	ChefPersNr	AbtNr
Smith	1234	Weimar	3334	5
Wong	3334	Köln	8886	5
Zelaya	9998	Erfurt	9876	4

Abteilung		
AbtName	Nr	Manager
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

AbtStandort	
AbtNr	Ort
1	Berlin
4	Weimar
5	Hamburg
5	Köln

ArbeitetInProjekt	
PersNr	ProjektNr
1234	1
1234	2
6668	3
4534	1

Projekt			
Name	Nr	Ort	AbtNr
X	1	Köln	5
Y	2	Hamburg	5
Z	3	Weimar	4
New	8	Weimar	4

Anfrage

„Liefere die Namen der Mitarbeiter, die in *allen* Projekten der Abteilung 5 arbeiten.“

Relationenalgebra

$$\pi_{\text{Name}} ((\text{ArbeitetInProjekt} \div \rho_{\text{ProjektNr} \leftarrow \text{Nr}}(\pi_{\text{Nr}}(\sigma_{\text{AbtNr}=5}(\text{Projekt})))) \bowtie \text{Mitarbeiter})$$

Domänenkalkül

$$\{(x_1) \mid \exists x_2 \exists x_3 \exists x_4 \exists x_5 \\ (\text{Mitarbeiter}(x_1, x_2, x_3, x_4, x_5) \wedge \forall z_1 \forall z_2 \forall z_3 \forall z_4 \\ (\neg \text{Projekt}(z_1, z_2, z_3, z_4) \vee \neg(z_4 = 5) \vee \text{oder: } ((\text{Projekt}(z_1, z_2, z_3, z_4) \wedge (z_4 = 5)) \rightarrow \\ \exists y_1 \exists y_2 (\text{ArbeitetInProjekt}(y_1, y_2) \wedge y_2 = z_2 \wedge y_1 = x_2))))))\}$$

Bemerkungen:

- Die Quantoren können verschoben werden, solange sich keine Variablenbindungen ändern und die Reihenfolge unter den \exists - und \forall -Quantoren erhalten bleibt:

$$\{(x_1) \mid \exists x_2 \exists x_3 \exists x_4 \exists x_5 \forall z_1 \forall z_2 \forall z_3 \forall z_4 \exists y_1 \exists y_2 \\ (\text{Mitarbeiter}(x_1, x_2, x_3, x_4, x_5) \wedge (\neg \text{Projekt}(z_1, z_2, z_3, z_4) \vee \neg(z_4 = 5) \vee \\ (\text{ArbeitetInProjekt}(y_1, y_2) \wedge y_2 = z_2 \wedge y_1 = x_2)))\}$$

Relationaler Domänenkalkül

Sichere Ausdrücke im Domänenkalkül

Folgende Anfrage liefert eine unendliche Zahl von Ergebnissen:

$$\{(x_1, x_2, x_3, x_4, x_5) \mid \neg \text{Mitarbeiter}(x_1, x_2, x_3, x_4, x_5)\}$$

Sei der Bereich bzw. die Domäne einer Formel α wie zuvor definiert. Dann ist ein Ausdruck $\{(x_1, x_2, \dots, x_n) \mid \alpha\}$ des Domänenkalküls sicher, falls folgende Bedingungen erfüllt sind:

1. Ist (c_1, c_2, \dots, c_n) im Anfrageergebnis enthalten, so muss $\{c_1, c_2, \dots, c_n\}$ Teilmenge der Domäne von α sein.
2. Für jede Teilformel $\exists x\beta$ muss gelten, dass β höchstens für Elemente aus seiner Domäne erfüllbar sein kann.
 $\Rightarrow \beta$ ist für alle Elemente, die nicht in seiner Domäne sind, unerfüllbar.
3. Für jede Teilformel $\forall x\beta$ muss gelten, dass $\forall x\beta$ dann und nur dann erfüllt ist, wenn β für alle Elemente aus seiner Domäne erfüllt ist.
 $\Rightarrow \beta$ ist für alle Elemente, die nicht in seiner Domäne sind, immer erfüllt.

Bemerkungen:

- Die Bedingungen 2. und 3. machen Vorschriften für die Struktur einer Formel. Sie konnten bei der Definition sicherer Ausdrücke im Tupelkalkül entfallen, da alle quantifizierten Tupelvariablen immer an eine (endliche) Relation gebunden sind. Im Domänenkalkül werden die Variablen an die Domänen der Attribute der Relationen gebunden; diese können unendlich viele Elemente enthalten.
- Die Bedingungen 2. und 3. verhindern, dass unendlich viele Variableninstanziierungen evaluiert werden müssen: Es sind Anfragen denkbar, die ein endliches Ergebnis liefern, aber die Evaluierung unendlich vieler Variableninstanziierungen erfordern. Dann läge die „Unendlichkeit“ in der Zeit und nicht in der Größe der Ergebnismenge.

Anfragekalküle

Ausdrucksstärke der Kalküle

Folgende drei Sprachen besitzen die gleiche Ausdruckskraft:

1. die relationale Algebra
2. der relationale Tupelkalkül, eingeschränkt auf sichere Ausdrücke
3. der relationale Domänenkalkül, eingeschränkt auf sichere Ausdrücke

Bemerkungen:

- ❑ Der Beweis erfolgt induktiv über den Aufbau der Ausdrücke in der jeweiligen Sprache. Unter anderem spezifiziert man äquivalente Ausdrücke des Tupelkalküls zu den Basisoperatoren der relationalen Algebra.
- ❑ Weil der (sichere) relationale Tupelkalkül und der (sichere) relationale Domänenkalkül die gleiche Ausdruckskraft wie die relationale Algebra besitzen, sind sie auch relational vollständig.
- ❑ Die Aussage, dass der relationale Tupelkalkül und der relationale Domänenkalkül relational vollständig sind, bedarf nicht der Einschränkung auf sichere Ausdrücke.