3 Relationale DB-Sprachen

3.1 Klassifikation

(Aspekte von) DB-Sprachen, Teilsprachen:

- DDL (Data Definition Language); beinhaltet Schemadefinition

- DML (Data Manipulation Language); beinhaltet Anfragen und Änderungen

Klassifikation von Anfragesprachen:

(a) deskriptiv:

Auszuwählende Objekte werden als Gesamtheit beschrieben (Was?)

(b) prozedural:

Ergebnis wird durch Folge von Operationen konstruiert (Wie?)

(b1) Operationen auf Objektmengen, z.B. Relationen

(b2) Operationen auf einzelnen Objekten, z.B. Tupeln, Sätzen

3.2 Relationen-Algebra (RA)

$\mathcal{R}$ sei Menge aller endlichen Relationen (inklusive Schemata) mit Operationen $\omega : \mathcal{R}(\times ... \times \mathcal{R}) \rightarrow \mathcal{R}$

Grundlage für Anfragen der Form

τ (RelName$_1$, ..., RelName$_k$) : DB-Zustände $\rightarrow \mathcal{R}$

wobei τ relationaler Term, d.h. Operation oder korrekte Zusammensetzung von Operationen, ist

Grundoperationen:

Gegeben seien endliche Relationen $R, S, ...$ mit zugehörigen Schemata $R(A_1, ..., A_n), S(B_1, ..., B_m), ...$

Bezeichnung: $Rang(R) = n$

1. Vereinigung $R \cup S$

2. Differenz $R - S$

3. (Kartesisches) Produkt $R \times S$

4. Projektion $\pi_{\overline{A}}(R)$ mit $\overline{A} = A_{i_1}, ..., A_{i_k}$ und $1 \leq i_1, ..., i_k \leq Rang(R)$

5. Selektion $\sigma_\varphi(R)$ mit φ atomare Formel

6. Umbenennung $\delta_{C \leftarrow A_i}(R)$ mit $C \notin \{A_1, ..., A_n\}$

Abgeleitete Operationen:

1. Durchschnitt $R \cap S$

2. verallgemeinerte Selektion $\sigma_\varphi(R)$

   φ ist logische Verknüpfung mittels $\wedge$, $\vee$ und $\neg$ von atomaren Formeln

3. Verbund $R *_{A_i \Theta B_j} S$

   Equiverbund, falls $\Theta \equiv =$

4. Natürlicher Verbund $R * S$

5. Division $R \div S$

Relationen-Algebra und Anfragesprachen

Relationale Operationen lassen sich zu Termen über Relationennamen und konstanten Relationen zusammensetzen; diese relationalen Terme definieren eine prozedurale Anfragesprache

Relationen-Algebra ist ein Maß für die **Ausdrucksfähigkeit** von Datenmanipulationssprachen $\mathcal{L}$:

- $\mathcal{L}$ **relational vollständig** gdw. jeder Term der Relationen-Algebra kann in $\mathcal{L}$ simuliert werden

- $\mathcal{L}$ **streng relational vollständig** gdw. jeder Term der Relationen-Algebra kann durch eine einzige Anweisung in $\mathcal{L}$ simuliert werden

Grenzen der Relationen-Algebra

Es gibt keinen Term der Relationen-Algebra, der zu jeder zweistelligen Relation R deren transitive Hülle

$$R^* = R \cup \{(x, y) \mid \exists z_1, ..., z_k :$$

$$(x, z_1) \in R \wedge (z_1, z_2) \in R \wedge ... \wedge (z_k, y) \in R\}$$

berechnet

3.3 Relationenkalküle

Relationenkalküle sind Grundlage für deskriptive Anfragesprachen

3.3.1 Bereichskalkül (BK)

Syntax: (über gegebenen Datentypen und gegebenem relationalem DB-Schema)

Terme:

(i) Konstanten zu Datentypen

(ii) Bereichsvariablen über Datentypen
z.B. x:integer, y:string

(iii) Wenn $u_1, ..., u_n$ Terme (bzgl. $D_1, ..., D_n$) und f Datentyp-Operation (bzgl. $D_1 \times ... \times D_n \to D$), so ist $f(u_1, ..., u_n)$ ein Term (bzgl. D);

Atomare Formeln:

(i) $\Theta(u_1, ..., u_n)$, wobei Θ boolsche Operation (z.B. $n = 2$ und $\Theta \equiv <$, also $< (u_1, u_2)$ oder $u_1 < u_2$)

(ii) $R(u_1, ..., u_n)$, wobei R Relationenname mit Schema $R(A_1 : D_1, ..., A_n : D_n)$ und u_i Term bzgl. D_i $(i = 1..n)$

Formeln:

(i) Jede atomare Formel ist eine Formel

Alle Vorkommen von Variablen darin sind frei

(ii) Wenn φ_1, φ_2 Formeln sind, so auch $(\varphi_1 \wedge \varphi_2)$, $(\varphi_1 \vee \varphi_2)$, $\neg(\varphi_1)$

Freie bzw. gebundene Vorkommen von Variablen in φ_1, φ_2 bleiben frei bzw. gebunden in $(\varphi_1 \wedge \varphi_2)$, $(\varphi_1 \vee \varphi_2)$, $\neg(\varphi_1)$

(iii) Wenn φ Formel und x Variable, so sind auch $\exists x(\varphi)$ und $\forall x(\varphi)$ Formeln

Freie Vorkommen von x in φ sind in $\exists x(\varphi)$, $\forall x(\varphi)$ gebunden, alle anderen Variablenvorkommen bleiben frei bzw. gebunden

Quantifizierungen dürfen auch auf Datentypen hinweisen, der Typ der Variablen ist aber bereits durch ihre Deklaration gegeben: Erlaubt ist $\exists x : D(\varphi)$ und $\forall x : D(\varphi)$

(iv) Klammerung wie üblich

Semantik: Fest gegeben seien Wertebereiche $\mid D \mid$ zu Datentypen D und zugehörige Operationen; es sei σ ein Zustand und β eine Belegung, d.h.:

$$\begin{array}{ccccc} \sigma & : & \text{Relationennamen} & \rightarrow & \text{Relationen} \\ & & R & \mapsto & \sigma(R) \\ \beta & : & \text{Variablen} & \rightarrow & \text{Werte} \\ & & x : D & \mapsto & \beta(x) \in \mid D \mid \end{array}$$

$[\sigma, \beta](u)$ bezeichnet den **Wert** des Termes u im Zustand σ unter der Belegung β

$[\sigma, \beta] \models \varphi$ bedeutet: Formel φ **gilt** im Zustand σ unter der Belegung β

Definition:

$[\sigma, \beta] \models R(u_1, ..., u_n)$ **gdw.**
$([\sigma, \beta](u_1), ..., [\sigma, \beta](u_n)) \in \sigma(R)$

$[\sigma, \beta] \models \Theta(u_1, ..., u_n)$ **gdw.** $[\sigma, \beta](\Theta(u_1, ..., u_n))$ wahr ist

$[\sigma, \beta] \models \varphi_1 \wedge \varphi_2$ **gdw.** $[\sigma, \beta] \models \varphi_1$ und $[\sigma, \beta] \models \varphi_2$

$[\sigma, \beta] \models \varphi_1 \vee \varphi_2$ **gdw.** $[\sigma, \beta] \models \varphi_1$ oder $[\sigma, \beta] \models \varphi_2$

$[\sigma, \beta] \models \neg\varphi_1$ **gdw.** nicht $[\sigma, \beta] \models \varphi_1$

$[\sigma, \beta] \models \exists x(\varphi)$ **gdw.** es gibt Belegung β' mit
$\beta'(Y) = \beta(Y)$ für $Y \neq x$ und $[\sigma, \beta'] \models \varphi$ ist wahr

$[\sigma, \beta] \models \forall x(\varphi)$ **gdw.** für alle Belegungen β' mit
$\beta'(Y) = \beta(Y)$ für $Y \neq x$ ist $[\sigma, \beta'] \models \varphi$ wahr

Ausdrücke des Bereichskalküls und Anfragen

Syntax: $\{x_1[: D_1], ..., x_n[: D_n] \mid \varphi\}$ wobei φ Formel mit freien Variablen $x_1 : D_1, ..., x_n : D_n$; $x_1, ..., x_n$ sind die Ergebnisvariablen und φ nennt man Qualifikation

Semantik: (in gegebenem Zustand σ)
$\{x_1, ..., x_n \mid \varphi\}$ bestimmt folgende Relation über
$\mid D_1 \mid \times ... \times \mid D_n \mid$:
$\{(d_1, ..., d_n) \mid$ es gibt eine Belegung β mit
$\beta(x_i) = d_i (i = 1..n)$, so daß $[\sigma, \beta] \models \varphi\}$ gilt

Definition: Ein Ausdruck heißt sicher, wenn er bestimmte syntaktische Einschränkungen erfüllt (ein sicherer Ausdruck bestimmt in jedem Zustand eine endliche Relation)

Satz: Der sichere Anteil des BK (und damit der BK) ist streng relational vollständig, d.h. zu jedem Term τ der Relationen-Algebra gibt es einen äquivalenten sicheren Ausdruck α des BK

τ und α sind äquivalent gdw. sie in jedem Zustand die gleiche Relation bestimmen

Satz: Zu jedem sicheren Ausdruck des BK gibt es einen äquivalenten Term der Relationen-Algebra

3.3.2 Tupelkalkül (TK)

Syntax: Terme

(i) Konstanten zu Datentypen

(ii) Tupelvariablen r, s, t, ... über dem Kreuzprodukt von Datentypen zusammen mit Komponentennamen $r : (A_1 : D_1, ..., A_n : D_n)$

(iii) Bildung von Komponenten $r.A_i$

(iv) Anwendung von Datentyp-Operationen

Atomare Formeln

(i) $\Theta(u_1, ...u_n)$ boolscher Term

(ii) $r = s$, wobei r, s Tupelvariable mit passenden Datentypen

(iii) $R(r)$, wobei R Relationenname und r Tupelvariable mit passenden Datentypen

Formeln: analog zum Bereichskalkül

Semantik: (nur Abweichungen vom Bereichskalkül)

Belegung β:

$$\begin{array}{cccc} \beta & : & \text{Varia-blen} & \rightarrow & \text{Tupel} \\ & & r & \mapsto & \beta(r) = (d_1, .., d_n) \in \mid D_1 \mid \times ... \times \mid D_n \mid \end{array}$$

$[\sigma, \beta](r.A_i) \;=\; \beta(r)_i \quad (= d_i)$

$[\sigma, \beta] \models R(r)$ gdw. $\beta(r) \in \sigma(R)$

Abkürzung: Gegeben $R(A_1 : D_1, ..., A_n : D_n)$

$\exists\ r : R\ (\varphi) :\Leftrightarrow \exists\ r : (A_1 : D_1, ..., A_n : D_n)\ (R(r) \wedge \varphi)$

$\forall\ r : R\ (\varphi) :\Leftrightarrow \forall\ r : (A_1 : D_1, ..., A_n : D_n)\ (R(r) \Rightarrow \varphi)$

Tupelkalkülausdrücke, die nur solche Quantifizierungen verwenden, sind sicher

Ausdrücke des Tupelkalküls

$\{r : (A_1[: D_1], ..., A_n[: D_n]) \mid \varphi\}$

wobei r einzige freie Variable in φ; dies bestimmt:

$\{\beta(r) \mid \beta \text{ Belegung, so daß } [\sigma, \beta] \models \varphi\}$

Satz: Zu jedem (sicheren) Ausdruck des BK gibt es einen äquivalenten (sicheren) Ausdruck des TK

Satz: Zu jedem (sicheren) Ausdruck des TK gibt es einen äquivalenten (sicheren) Ausdruck des BK

Es sind also

1. Relationen-Algebra,

2. der sichere Anteil des Bereichskalküls und

3. der sichere Anteil des Tupelkalküls

in ihrer Ausdrucksfähigkeit äquivalent