Baze de date
Algebra relațională

Nicolae-Cosmin Vârlan

Elemente ale modelului relațional

- U mulțime de atribute: $U = \{A_1, A_2, \ldots, A_n\}$;
- $dom(A_i)$ - domeniul valorilor atributului A_i;

Definim *uplu* peste U ca fiind funcția:

$$\varphi : U \to \bigcup_{1 \leq i \leq n} dom(A_i) \quad \text{a.i. } \varphi(A_i) \in dom(A_i), 1 \leq i \leq n$$

Fie valorile v_i astfel încât $v_i = \varphi(A_i)$.

Notăm cu $\{A_1 : v_1,\ A_2 : v_2,\ \ldots,\ A_n : v_n\}$ asocierea dintre atributele existente în U și valorile acestora. În cazul în care sunt considerate mulțimi ordonate (de forma $(A_1, A_2, \ldots, A_n)$), notația va fi de forma: $(v_1, v_2, \ldots, v_n)$.

Elemente ale modelului relațional

Considerăm mulțimea ordonată $(A_1, A_2, \ldots A_n)$. Pentru orice uplu φ, există vectorul $(v_1, v_2, \ldots v_n)$ a.î. $\varphi(A_i) = v_i,\ 1 \leq i \leq n$.

Pentru un vector $(v_1, v_2, \ldots v_n)$ cu $v_i \in dom(A_i),\ 1 \leq i \leq n$ există un uplu φ a.î. $\varphi(A_i) = v_i$.

În practică este considerată o anumită ordonare a atributelor.

Elemente ale modelului relațional

O mulțime de uple peste U se numește *relație* și se notează cu r.

r poate varia în timp dar nu și în structură.

Exemplu:
$r = \{(v_{11}, v_{12}, \ldots v_{1n}), (v_{21}, v_{22}, \ldots v_{2n}), \ldots, (v_{m1}, v_{m2}, \ldots v_{mn})\}$.

Structura relației se va nota cu $R[U]$ unde R se numește *numele relației* iar U este mulțimea de *atribute* corespunzătoare.

Notații echivalente $R(U)$, $R(A_1, A_2, \ldots, A_n)$, $R[A_1, A_2, \ldots, A_n]$.

$R[U]$ se mai numește și *schemă de relație*.

Elemente ale modelului relațional

În practică, o relație r poate fi reprezentată printr-o matrice:

$$r : \begin{array}{cccc} A_1 & A_2 & \ldots & A_n \\ \hline v_{11} & v_{12} & \ldots & v_{1n} \\ \ldots & \ldots & \ldots & \ldots \\ v_{m1} & v_{m2} & \ldots & v_{mn} \\ \hline \end{array}$$

unde $(v_{i1}, v_{i2}, \ldots, v_{in})$ este un uplu din r, $1 \leq i \leq m$ și $v_{ij} \in dom(A_j),\ 1 \leq j \leq n, 1 \leq i \leq m$

Vom nota cu t_i linia (tuplul) cu numărul i din matrice: $t_i = (v_{i1}, v_{i2}, \ldots, v_{in}), \forall i \in [1, m]$

Elemente ale modelului relațional

O mulțime finită D de scheme de relație se numește *schemă de baze de date*. Formal, $D = \{R_1[U_1], \ldots, R_h[U_h]\}$ unde $R_i[U_i]$ este o schemă de relație, $1 \leq i \leq h$.

O *bază de date peste D* este o corespondență ce asociază fiecărei scheme de relație din D o relație.

Exemplu:

$r_1, r_2, \ldots r_h$ este o bază de date peste $D = \{R_1[U_1], \ldots, R_h[U_h]\}$.

Considerând D ca fiind ordonată $D = (R_1[U_1], \ldots, R_h[U_h])$, putem nota baza de date sub forma $(r_1, r_2, \ldots r_h)$

Corespondența cu terminologia din practică

- atribut (A_i) = denumirea unei coloane dintr-un tabel;
- valoarea atributului A_i ($\varphi(A_i)$ sau v_i) = valoarea dintr-o celulă a tabelului
- relație (r) = tabel
- schema de relație ($R[U]$) = schema tabelei
- tuplu (t_i) = linie din tabel

Operații

Asupra unei mulțimi de relații putem efectua o serie de operații. Există două categorii de operatori:

- Operatori din teoria mulțimilor: Reuniunea($\cup$), Intersecția ($\cap$), Diferența($-$), Produsul Cartezian($\times$)
- Operatori specifici algebrei relaționale: Proiecția (π), Selecția(σ), Redenumirea(ρ), Joinul Natural($\bowtie$), θ-Joinul, equijoinul, Semijoinul($\ltimes$ și $\rtimes$), Antijoinul($\triangleright$), Divizarea($\div$), Joinul la Stânga (⟕), Joinul la Dreapta(⟖), Joinul Exterior(⟗)

Operații pe mulțimi de tuple - *Reuniunea*: $\cup$

În cazul operațiilor pe mulțimi (cu excepția Produsului Cartezian), acestea se realizează între două relații r_1 și r_2 care sunt NEAPĂRAT construite peste aceeași mulțime de atribute.

Reuniunea a două relații r_1 și r_2, ambele peste o aceeași mulțime de atribute U (sau peste aceeași schemă de relație $R[U]$), este o relație notată cu $r_1 \cup r_2$ definită astfel:

$$r_1 \cup r_2 = \{t \mid t = \text{uplu},\ \ t \in r_1 \text{ sau } t \in r_2\}$$

În practică, acest lucru se realizează utilizând cuvântul cheie UNION. Studenții din anii 1 și 3 sunt selectați de interogarea:

```
SELECT * FROM studenti WHERE an=1
UNION
SELECT * FROM studenti WHERE an=3;
```

Operații pe mulțimi de tuple - *Diferența*: $-$

Diferența a două relații r_1 și r_2, ambele peste o aceeași mulțime de atribute U (sau peste aceeași schemă de relație $R[U]$), este o relație notată cu $r_1 - r_2$ definită astfel:

$$r_1 - r_2 = \{t \mid t = uplu, \;\; t \in r_1 \; si \; t \notin r_2\}$$

În practică, acest lucru se realizează utilizând cuvântul cheie MINUS. Pentru a-i selecta pe studenții din anul 2 fără bursă, putem să îi selectăm pe toți studenții din anul 2 și apoi să îi eliminăm pe cei cu bursă:

```
SELECT * FROM studenti WHERE an=2
MINUS
SELECT * FROM studenti WHERE bursa IS NOT NULL;
```

Operații pe mulțimi de tuple - *Intersecția*: $\cap$

Intersecția a două relații r_1 și r_2, ambele peste o aceeași mulțime de atribute U (sau peste aceeași schemă de relație $R[U]$), este o relație notată cu $r_1 \cap r_2$ definită astfel:

$$r_1 \cap r_2 = \{t \mid t = uplu, \;\; t \in r_1 \; si \; t \in r_2\}$$

În practică, acest lucru se realizează utilizând cuvântul cheie INTERSECT. Putem afla care studenți din anul 2 au bursă rulând:

```
SELECT * FROM studenti WHERE an=2
INTERSECT
SELECT * FROM studenti WHERE bursa IS NOT NULL;
```

Operatorul de intersecție poate fi obținut din ceilalți doi: $r_1 \cap r_2 = r_1 - (r_1 - r_2)$.

Operații pe mulțimi de tuple - *Produsul Cartezian*: $\times$

Produsul cartezian a două relații r_1 definită peste $R_1[U_1]$ și r_2 definită peste $R_2[U_2]$ cu $U_1 \cap U_2 = \emptyset$ este o relație notată cu $r_1 \times r_2$ definită astfel:

$$r_1 \times r_2 = \{t \mid t = \text{uplu peste } U_1 \cup U_2,\ t[U_1] \in r_1 \text{ și } t[U_2] \in r_2\}$$

De aceasta dată, cele două relații nu trebuie să fie peste aceeași mulțime de atribute. Rezultatul va fi o nouă relație peste o mulțime de atribute formată din atributele relațiilor inițiale.

Operații pe mulțimi de tuple - *Produsul Cartezian*: $\times$

Dacă un atribut s-ar repeta, el va fi identificat diferit. Spre exemplu, chiar dacă tabelele note și cursuri au un același atribut (id_curs), nu se face nici o sincronizare după acesta ci se vor crea două atribute diferite: note.id_curs respectiv cursuri.id_curs.

Produsul cartezian între aceste tabele, în practică, se obține executând interogarea:

```
SELECT * FROM cursuri, note;
```

Operații specifice algebrei relaționale

Operațiile pe mulțimi aveau ca elemente tuplele. Uneori aceste tuple nu sunt compatibile (de exemplu nu putem reuni o relație peste $R_1[U_1]$ cu una peste $R_2[U_2]$ dacă $U_1 \neq U_2$).

Pentru a opera asupra atributelor ce definesc tuplele din rezultat, avem nevoie de o serie de operatori specifici algebrei relaționale.

Operații în algebra relațională - *Proiecția:* π

Considerăm:

- $R[U]$ = schemă de relație;
- $X \subseteq U$;
- t = uplu peste $R[U]$ ($t \in r$).

Se numește *proiecția lui t relativă la X* și notată cu $\pi_X[t]$, restricția lui t la mulțimea de atribute X. (Uneori vom scrie t[X])

Exemplu:

Dacă $U = (A_1, A_2, \ldots, A_n)$ atunci $t = (v_1, v_2, \ldots, v_n)$.
Considerăm $X = (A_{i_1}, A_{i_2}, \ldots, A_{i_k})$, $1 \leq i_1 < i_2 < \ldots < i_k \leq n$.

atunci $\pi_X[t] = (v_{i_1}, v_{i_2}, \ldots, v_{i_k})$;

Operații în algebra relațională - *Proiecția:* π

Dacă r este o relație peste $R[U]$ și $X \subseteq U$, atunci *proiecția lui r relativă la X* este $\pi_X[r] = \{\pi_X[t] \mid t \in r\}$

Exemplu:

Dacă $U = (A_1, A_2, \ldots, A_n)$ atunci
$r = \{(v_{11}, v_{12}, \ldots v_{1n}), (v_{21}, v_{22}, \ldots v_{2n}), \ldots, (v_{m1}, v_{m2}, \ldots v_{mn})\}$.
Considerăm $X = (A_{i_1}, A_{i_2}, \ldots, A_{i_k})$, $1 \leq i_1 < i_2 < \ldots < i_k \leq n$.

atunci
$\pi_X[r] = \{(v_{1i_1}, v_{1i_2}, \ldots v_{1i_k}), (v_{2i_1}, \ldots v_{2i_k}), \ldots, (v_{mi_1}, \ldots v_{mi_k})\}$

În practică, proiecția se realizează selectând doar anumite câmpuri ale tabelei (anumite atribute):

```
SELECT nume, prenume FROM studenti;
```

Operații în algebra relațională - *Proiecția:* π

Ca și exemplu, vom scrie o interogare care să returneze toate persoanele care trec pragul Facultății (studenți și profesori):

```
SELECT nume, prenume FROM studenti
  UNION
  SELECT nume, prenume FROM profesori;
```

În cazul în care cele două câmpuri (nume, prenume) din cele două tabele au același tip (de exemplu nume este de tip VARCHAR2(10) în ambele tabele), interogarea va afișa toate persoanele ce "trec pragul Facultății".

Observație: Pentru a modifica tipul nume din tabela profesori la VARCHAR2(10) executați comanda:

```
ALTER TABLE profesori MODIFY nume VARCHAR2(10);
```

Operații în algebra relațională - *Selecția:* σ

Fie r o relație peste $R[U]$, $A, B \in U$ și c este o constantă

O expresie elementară de selecție este definită prin următoarea formulă (forma Backus-Naur):

$$e = A\,\varphi\,B \mid A\,\varphi\,c \mid c\,\varphi\,B$$

Unde φ este o relație booleană între operanzi.

Se numește expresie de selecție (forma Backus-Naur):

$$\theta = e \mid \theta \wedge \theta \mid \theta \vee \theta \mid (\theta)$$

Operații în algebra relațională - *Selecția:* σ

Fie θ o expresie de selecție. Atunci:

- când $\theta = A \varphi B$, t satisface θ dacă are loc $\pi_A[t] \varphi \pi_B[t]$,
- când $\theta = A \varphi c$, t satisface θ dacă are loc $\pi_A[t] \varphi c$,
- când $\theta = c \varphi B$, t satisface θ dacă are loc $c \varphi \pi_B[t]$,
- când $\theta = \theta_1 \wedge \theta_2$, t satisface θ dacă t satisface atât pe θ_1 cât și pe θ_2,
- când $\theta = \theta_1 \vee \theta_2$, t satisface θ dacă t satisface măcar pe unul dintre θ_1 și θ_2.

Dacă θ este o expresie de selecție atunci selecția se notează cu $\sigma_\theta(r)$ și este definită ca:
$\sigma_\theta(r) = \{t | t \in r, t \text{ satisface } \theta\}$

Operații în algebra relațională - *Selecția:* σ

În SQL, selecția se obține utilizând o formulă logică ce are rolul de a selecta doar anumite rânduri.

Exemplu:

```
SELECT * FROM studenti
WHERE ((an=2) AND (bursa IS NULL));
```

În acest exemplu, θ_1 este $an = 2$, θ_2 este $bursa\ IS\ NULL$, $\theta = \theta_1 \wedge \theta_2$ și r este mulțimea de rânduri din tabela studenți.

Rezultatul este mulțimea studenților din anul 2 care nu au bursă.

Operații în algebra relațională - *Redenumirea:* ρ

Operatorul de *redenumire* are rolul de a schimba numele unui atribut cu alt nume. Formal, dacă dorim să schimbăm atributul A_1 în A_1' vom utiliza scrierea $\rho_{A_1/A_1'}(r)$. Restul atributelor peste care a fost construit r vor rămâne neschimbate.

În SQL, redenumirea se realizează prin utilizarea cuvântului AS:

Exemplu:
SELECT bursa * 1.25 AS "BursaNoua" FROM studenti;
SELECT bursa + bursa/4 AS "BursaNoua" FROM studenti;
Dacă nu am redenumi atributul nou obținut, cele două relații ar fi considerate diferite (în prima numele atributului ar fi "bursa * 1.25", iar în a doua ar fi fost "bursa + bursa/4")

Operații în algebra relațională - *Join natural*: $\bowtie$

Considerăm:

- r_1 relație peste $R_1[U_1]$;
- r_2 relație peste $R_2[U_2]$;

Se numește *Join natural* a relațiilor r_1 și r_2, relația $r_1 \bowtie r_2$ peste $U_1 \cup U_2$ definită prin:

$$r_1 \bowtie r_2 = \{t \mid t \text{ uplu peste } U_1 \cup U_2,\ t[U_i] \in r_i,\ i = 1, 2\}$$

Dacă R este un nume pentru relația peste $U_1 \cup U_2$ atunci $r_1 \bowtie r_2$ este definită peste $R[U_1 \cup U_2]$
Pentru simplitate vom nota $U_1 \cup U_2$ cu U_1U_2.

Operații în algebra relațională - *Join natural*: $\bowtie$

Exemplu:

Fie $R_1[A, B, C, D]$, $R_2[C, D, E]$ și r_1, r_2 a.i.:

r_1 :

A	B	C	D
0	1	0	0
1	1	0	0
0	0	1	0
1	1	0	1
0	1	0	1

r_2 :

C	D	E
1	1	0
1	1	1
0	0	0
1	0	0
1	0	1

Atunci: $r_1 \bowtie r_2$:

A	B	C	D	E
0	1	0	0	0
1	1	0	0	0
0	0	1	0	0
0	0	1	0	1

Operații în algebra relațională - *Join natural*: ⋈

Următoarea interogare identifică cui aparține fiecare nota din tabelul note. Joinul se face după câmpul nr_matricol între tabelele studenti și note:

```
SELECT nume, valoare FROM studenti
  NATURAL JOIN note;
```

```
SELECT nume, valoare FROM studenti
  JOIN note ON studenti.nr_matricol = note.nr_matricol;
```

Se poate observa că dacă din produsul cartezian am elimina acele cazuri în care câmpul "nr_matricol" nu este identic în ambele tabele, am obține, de fapt, același rezultat. Din acest motiv, joinul de mai sus poate fi scris și sub forma:

```
SELECT nume, valoare FROM studenti,note
  WHERE studenti.nr_matricol = note.nr_matricol;
```

Proprietăți ale Joinului natural

- $(r_1 \bowtie r_2)[U_1] \subseteq r_1$
- $(r_2 \bowtie r_1)[U_2] \subseteq r_2$

Dacă $X = U_1 \cap U_2$ și:
$r_1' = \{t_1 | t_1 \in r_1, \exists t_2 \in r_2$ a.i. $t_1[X] = t_2[X]\}$ și $r_1" = r_1 - r_1'$,
$r_2' = \{t_2 | t_2 \in r_2, \exists t_1 \in r_1$ a.i. $t_1[X] = t_2[X]\}$ și $r_2" = r_2 - r_2'$,
atunci: $r_1 \bowtie r_2 = r_1' \bowtie r_2'$, $(r_1 \bowtie r_2)[U_1] = r_1'$, $(r_2 \bowtie r_1)[U_2] = r_2'$.

Dacă $\overline{r_1} \subseteq r_1, \overline{r_2} \subseteq r_2$ și $\overline{r_1} \bowtie \overline{r_2} = r_1 \bowtie r_2$ atunci $r_1' \subseteq \overline{r_1}$ si $r_2' \subseteq \overline{r_2}$

Dacă $U_1 \cap U_2 = \emptyset$ atunci $r_1 \bowtie r_2 = r_1 \times r_2$.

Extindere Join natural

Fie r_i relație peste $R_i[U_i], i = \overline{1, h}$ atunci:

$r_1 \bowtie r_2 \bowtie \ldots \bowtie r_h = \{t | t$ uplu peste $U_1, \ldots U_h,$ a.î. $t[U_i] \in r_i, i = \overline{1, h}\}$

Notații echivalente:

- $r_1 \bowtie r_2 \bowtie \ldots \bowtie r_h$
- $\bowtie \langle r_i, i = 1, h \rangle$
- $* \langle r_i, i = 1, h \rangle$

Operația join este asociativă.

Operații în algebra relațională - *θ-join, equijoin*

Fie r_i peste $R_i[U_i]$, $i = \overline{1,2}$ cu $A_{\alpha_1}, A_{\alpha_2}, \ldots A_{\alpha_k} \in U_1$ și $B_{\beta_1}, B_{\beta_2}, \ldots B_{\beta_k} \in U_2$ și
$\theta_i : dom(A_{\alpha_i}) \times dom(B_{\beta_i}) \rightarrow \{true, false\}, \forall i = \overline{1,k}$

θ-joinul a două relații r_1 și r_2, notat cu $r_1 \underset{\theta}{\bowtie} r_2$, este definit prin:

$$r_1 \underset{\theta}{\bowtie} r_2 = \{(t_1, t_2) | t_1 \in r_1, t_2 \in r_2, t_1[A_{\alpha_i}]\theta_i t_2[B_{\beta_i}], i = \overline{1,k}\}$$

unde $\theta = (A_{\alpha_1}\theta_1 B_{\beta_1}) \wedge (A_{\alpha_2}\theta_2 B_{\beta_2}) \wedge \ldots \wedge (A_{\alpha_k}\theta_k B_{\beta_k})$

Daca θ_i este operatorul de egalitate, atunci θ-joinul se mai numeste si equijoin.

Operații în algebra relațională - *θ-join, equijoin*

Observație 1: un join oarecare cu condiția TRUE pentru orice combinație de tuple este un produs cartezian: $r_1 \underset{true}{\bowtie} r_2 = r_1 \times r_2$

Observație 2: Joinul oarecare poate fi considerat ca fiind o filtrare după anumite criterii ale rezultatelor unui produs cartezian: $r_1 \underset{\theta}{\bowtie} r_2 = \sigma_\theta(r_1 \times r_2)$

Exemplu SQL:

```
SELECT s.nume, p.nume FROM studenti s, profesori p
  WHERE s.nume > p.nume;
```

Operații în algebra relațională - *Semijoin:* $\ltimes$ *și* $\rtimes$

Operația de semijoin stâng selectează acele rânduri din relația aflată în partea stângă ($\ltimes$) care au corespondent (în sensul joinului natural) în relația din partea dreapta.

Formal, definim semijoinul stâng a două relații r_1 peste $R_1[U_1]$ și r_2 peste $R_2[U_2]$ ca fiind:

$$r_1 \ltimes r_2 = \pi_{U_1}(r_1 \bowtie r_2)$$

Deja întâlnit la proprietațile Joinului natural sub denumirea r_1'.

Semijoinul drept este definit similar dar preluând liniile din relația aflată în dreapta (doar cele ce au corespondent în relația din stânga).

Operații în algebra relațională - *Antijoin:* $\triangleright$

Tuplele rămase din relația din stânga (care nu au fost preluate de semijoinul stâng), formează rezultatul operatorului Antijoin.

Formal, definim antijoinul stâng a două relații r_1 peste $R_1[U_1]$ și r_2 peste $R_2[U_2]$ ca fiind:

$$r_1 \triangleright r_2 = r_1 - \pi_{U_1}(r_1 \bowtie r_2)$$

$\ldots r_1$"

Operații în algebra relațională - *Joinul la Stânga:* $⟕$

Fie r_1 și r_2 două relații în care nu toate tuplele din r_1 au un corespondent în r_2.
Operația Join la Stanga a celor două relații r_1 și r_2 este reuniunea dintre tuplele existente în $r_1 \bowtie r_2$ și tuplele din r_1 ce nu sunt utilizate în join completate cu valoarea NULL pentru atributele din U_2.

$$r_1 ⟕ r_2 = r_1 \bowtie r_2 \text{ UNION } \pi_{U_1 U_2}(r_1 - \pi_{U_1}(r_1 \bowtie r_2))$$

Joinul la Dreapta este definit similar, de această dată preluând și liniile ce nu s-au folosit in Joinul natural din tabela din dreapta (r_2).

Operații în algebra relațională - *Joinul Extern:* $⟗$

Operația de Join exterior cuprinde toate liniile din Joinul la Stânga și din Joinul la Dreapta.

$$r_1 ⟗ r_2 = (r_1 ⟕ r_2) \cup (r_1 ⟖ r_2)$$

Operații în algebra relațională - *Joinul Extern:* ⟗

Exemple:

```
SELECT * FROM studenti LEFT JOIN profesori ON
  studenti.prenume = profesori.prenume;
```

(Toți studenții și asociați cu profesorii cu același prenume când este cazul)

```
SELECT * FROM studenti RIGHT JOIN profesori ON
  studenti.prenume = profesori.prenume;
```

(Unii studenți care sunt asociați cu profesorii având același prenume împreună cu restul profesorilor)

```
SELECT * FROM studenti FULL JOIN profesori ON
  studenti.prenume = profesori.prenume;
```

(Studenții și profesorii și asocierile între ei, dacă există)

Notații (alternative) pentru operatorii din alg. relațională

Proiecția ($\pi_U(r_1)$): $r_1[U]$
Join natural ($r_1 \bowtie r_2$): $r_1 * r_2$
Join oarecare (sau theta-join): $r_1 \underset{\theta}{\bowtie} r_2$
Selecția : $\sigma_\theta(r_1)$ [obs: $r_1 \underset{\theta}{\bowtie} r_2 = \sigma_\theta(r_1 \times r_2)$]
Join la stânga: $r_1 \rhd \circ \lhd_L r_2$
Join la dreapta: $r_1 \rhd \circ \lhd_R r_2$
Full outer join : $r_1 \rhd \circ \lhd r_2$
Redenumirea: Dacă r este definit peste $B_1, B_2, \ldots, B_n$ și vrem să redenumim numele atributelor, vom folosi operatorul de redenumire $\rho : r' = \rho(r_1)_{A_1, A_2, \ldots, A_n}$ - redenumirea atributelor lui r în $A_1, A_2, \ldots, A_n$

Exerciții:

1. Pentru r_1, r_2 exemplificate la Joinul natural, construiți restul tipurilor de Join studiate.
2. Utilizând schema de baze de date de la laborator, scrieți în algebra relațională expresii de selecție pentru următoarele:
   - Cursurile din facultate împreună cu numele prof. ce le țin.
   - Numele și prenumele studenților din anul 1 și care au bursă mai mare de 300 ron.
   - Prenumele studenților care au același nume de familie ca măcar unul din profesori.
   - Numele și prenumele studenților, cursurile pe care le-au urmat și notele pe care le-au obținut.

Scrieți interogările SQL asociate formulelor din algebra relațională scrise mai sus.

Bibliografie

- Baze de date relaționale. Dependențe - *Victor Felea*; Univ. Al. I. Cuza, 1996

Software

- Relational