



BAZE DE DATE

CURS 7

Evaluarea și optimizarea interogărilor

Procesarea interogărilor

- O **expresie a algebrei relaționale** este constituită din **relații** legate prin **operații** din algebra relațională.
- O expresie se poate reprezenta **grafic** cu ajutorul unui **arbore**, numit **arbore algebric**, în care nodurile corespund operatorilor din cadrul expresiei respective.
- Evaluarea unei expresii presupune efectuarea prelucrărilor indicate de operatorii din expresie în ordinea aparițiilor sau în ordinea fixată prin paranteze.
- **Rezultatul** evaluării unei expresii este o **relație derivată** din relațiile menționate ca operanzi în cadrul expresiei.
- Două expresii sunt **echivalente**, dacă în urma evaluării lor se obține ca rezultat aceeași relație.

Procesarea interogărilor

Exemple referitoare la moduri echivalente de exprimare a unei cereri.

1. Informații despre salariații care nu contribuie la machetarea nici unei publicații, dar au retribuirea mai mare decât o valoare dată.
2. Codul și numele subantreprenorilor care au realizat lucrări specializate la obiective case de vacanță sau cabane.
3. Codurile și telefoanele investitorilor, valoarea investiției, durata de execuție a lucrărilor pentru pentru care valoarea investiției este între două limite specificate.
4. Perioada de desfășurare și prețul ofertelor care încep după 1 ianuarie 2015 și sunt:
 - sejururi la munte;
 - excursii în care autocarele sunt conduse de șoferi angajați după 1 mai 1997 și supravegheate de ghizi ce cunosc limba engleză care au făcut specializare în Suedia.

Procesarea interogărilor

- În majoritatea sistemelor de gestiune, și în special în cele relaționale, interfața cu utilizatorul este de tip **neprocedural**.
 - Utilizatorul definește datele pe care dorește să le vizualizeze fără a da algoritmi de acces (**CE**, nu **CUM**).
 - Sistemul trebuie să convertească cererea utilizatorului:
 - într-o cerere optimală;
 - în proceduri de acces optimal la datele fizice.
- *Garantarea absolută a performanțelor optime pentru procesorul limbajului relațional este imposibilă.*
- *Corectă ar fi utilizarea cuvântului „**ameliorare**” în locul cuvântului „**optimizare**”.*

Procesarea interogărilor

Evaluarea unei interogări se efectuează în trei etape:

- ❑ **Analiza cererii** – presupune **studierea sintactică și semantică** a cererii pentru a verifica corectitudinea sa și a simplifica criteriul de căutare.
- ❑ **Ordonanțarea** – presupune **descompunerea** cererii **într-o mulțime de operații elementare** și determinarea unei **ordini optimale** a acestor operații. Operațiile sunt, în general, cele ale algebrei relaționale. La sfârșitul etapei se obține un **plan de execuție al cererii**.
- ❑ **Execuția** – presupune **efectuarea** (paralelă și/sau secvențială) operațiilor elementare furnizate de planul de execuție pentru a obține **rezultatul cererii**.

Procesarea interogărilor

- Presupunem că utilizatorul transmite sistemului de gestiune o **cerere** exprimată prin **comenzi SQL**.
 - Pentru a răspunde cererii, SGBD-ul trebuie să înțeleagă cererea utilizatorului.
 - Cererea trebuie să fie:
 - **corectă sintactic**,
 - datele trebuie să fie **disponibile** utilizatorului și
 - trebuie localizate analizând diferite **drumuri de acces** la ele.

Procesarea interogărilor

- Aceste funcții sunt realizate de SGBD cu ajutorul a **două module funcționale** care comunică permanent:
 - **analizorul cererilor**, care asigură:
 - verificarea sintactică și semantică a cererii,
 - localizarea datelor implicate în cerere (găsirea adresei blocurilor ce conțin datele),
 - furnizarea planului de execuție.
 - **administratorul datelor** (executorul), care execută efectiv cererea (primește planurile de execuție furnizate de modulul de optimizare și le execută).
 - Execuția presupune căutarea blocurilor ce conțin datele și transferul blocurilor în memoria *cache*.

Procesarea interogărilor

Ideea generală:

cerere → arbore algebric (nu este unic) → plan de execuție → optimizare

► Un plan de execuție implică:

- **o secvență de pași pentru evaluarea cererii** - în mod obișnuit, fiecare pas din planul de execuție corespunde unei operații relaționale
- **metoda care va fi folosită pentru evaluarea operației** - de obicei, pentru o operație relațională dată, există mai multe metode ce pot fi folosite pentru evaluarea acesteia.

Procesarea interogărilor

- ▶ Două planuri de execuție **diferite care au întotdeauna același rezultat se numesc echivalente. Planuri de execuție echivalente pot avea diferite costuri.**
- ▶ Scopul optimizării cererilor este de a găsi, printre diversele planuri de execuție echivalente, pe acela de **cost minim**.
- ▶ Într-un sistem centralizat, costul evaluării unei cereri este suma a două componente:

costul I/O (transferuri de date) + costul CPU (verificare de condiții, operații join etc.).

Ordinea de execuție a operațiilor

- O interogare constă dintr-un număr de operații.
 - Ordinea în care se efectuează operațiile are un rol important în evaluarea costului necesar realizării interogării.
- Există **două modalități** de abordare pentru a determina ordinea de execuție a operațiilor:
 - algebric
 - bazat pe estimarea costului.
- Ambele folosesc o mulțime de **reguli** care permit **transformarea unui plan de execuție** (reprezentat ca o expresie scrisă în termenii algebrei relaționale) în altul, echivalent.

Ordinea de execuție a operațiilor

- **Optimizarea cererilor bazată pe algebra relațională** se realizează în două etape:
 - se exprimă cererile sub forma unor **expresii algebrice** relaționale;
 - se aplică acestor expresii **transformări algebrice** care conduc la **expresii echivalente**, dar care vor fi **executate mai eficient**.
- Procesul de transformare a cererilor se realizează conform unei **strategii de optimizare** care poate să fie:
 - independentă de modul de memorare a datelor (strategie generală) sau
 - dependentă de modul de memorare (strategie specifică unui anumit SGBD).

Proprietățile operatorilor algebrei relaționale

Proprietatea 1. Comutativitatea operațiilor de *join* și produs cartezian:

$$\text{JOIN}(R1, R2) = \text{JOIN}(R2, R1)$$

$$R1 \times R2 = R2 \times R1$$

Proprietatea 2. Asociativitatea operațiilor de *join* și produs cartezian:

$$\text{JOIN}(\text{JOIN}(R1, R2), R3) = \text{JOIN}(R1, \text{JOIN}(R2, R3))$$

$$(R1 \times R2) \times R3 = R1 \times (R2 \times R3)$$

Proprietatea 3. Compunerea proiecțiilor:

$$\Pi_{A1, \dots, Am} (\Pi_{B1, \dots, Bn} (R)) = \Pi_{A1, \dots, Am} (R),$$

unde $\{A_1, A_2, \dots, A_m\} \subseteq \{B_1, B_2, \dots, B_n\}$.

Proprietatea 4. Compunerea selecțiilor:

$$\sigma_{cond1} (\sigma_{cond2} (R)) = \sigma_{cond1 \wedge cond2} (R) = \sigma_{cond2} (\sigma_{cond1} (R)),$$

unde am notat prin *cond* condiția după care se face selecția.

Proprietățile operatorilor algebrei relaționale

Proprietatea 5. Comutarea selecției cu proiecția:

$$\Pi_{A_1, \dots, A_m} (\sigma_{cond} (R)) = \sigma_{cond} (\Pi_{A_1, \dots, A_m} (R)),$$

unde condiția după care se face selecția implică numai attributele A_1, \dots, A_m .

Dacă condiția implică și attributele B_1, \dots, B_n , care nu aparțin mulțimii $\{A_1, \dots, A_m\}$, atunci:

$$\Pi_{A_1, \dots, A_m} (\sigma_{cond} (R)) = \Pi_{A_1, \dots, A_m} (\sigma_{cond} (\Pi_{A_1, \dots, A_m, B_1, \dots, B_n} (R)))$$

Proprietățile operatorilor algebrei relaționale

Proprietatea 6. Comutarea selecției cu produsul cartezian:

Dacă toate atributele menționate în condiția după care se face selecția sunt atribute ale relației $R1$, atunci:

$$\sigma_{cond} (R1 \times R2) = \sigma_{cond} (R1) \times R2$$

Dacă condiția este de forma $cond1 \wedge cond2$ și dacă $cond1$ implică numai atribute din $R1$, iar $cond2$ implică numai atribute din $R2$, atunci

$$\sigma_{cond} (R1 \times R2) = \sigma_{cond1} (R1) \times \sigma_{cond2} (R2)$$

Dacă $cond1$ implică numai atribute din $R1$, iar $cond2$ implică atribute atât din $R1$ cât și din $R2$, atunci:

$$\sigma_{cond} (R1 \times R2) = \sigma_{cond2} (\sigma_{cond1} (R1) \times R2)$$

Proprietățile operatorilor algebrei relaționale

Proprietatea 7. Comutarea selecției cu reuniunea:

$$\sigma_{cond} (R1 \cup R2) = \sigma_{cond} (R1) \cup \sigma_{cond} (R2)$$

Proprietatea 8. Comutarea selecției cu diferența:

$$\sigma_{cond} (R1 - R2) = \sigma_{cond} (R1) - \sigma_{cond} (R2)$$

Proprietatea 9. Comutarea proiecției cu produsul cartezian:

Dacă A_1, \dots, A_m este o listă de attribute ce apar în schemele relaționale $R1$ și $R2$ și dacă lista este formată din attribute aparținând lui $R1$ (notate prin B_1, \dots, B_n) și din attribute aparținând lui $R2$ (notate prin C_1, \dots, C_k) atunci:

$$\Pi_{A_1, \dots, A_m} (R1 \times R2) = \Pi_{B_1, \dots, B_n} (R1) \times \Pi_{C_1, \dots, C_k} (R2)$$

Proprietatea 10. Comutarea proiecției cu reuniunea:

$$\Pi_{A_1, \dots, A_m} (R1 \cup R2) = \Pi_{A_1, \dots, A_m} (R1) \cup \Pi_{A_1, \dots, A_m} (R2)$$

Proprietățile operatorilor algebrei relaționale

Proprietatea 11. Compunerea proiecției cu operația *join*:

Dacă A_1, \dots, A_m este o listă de atribute ce apar în schemele relaționale $R1$ și $R2$ și dacă lista este formată din atribute aparținând lui $R1$ (notate prin B_1, \dots, B_n) și din atribute aparținând lui $R2$ (notate prin C_1, \dots, C_k) atunci:

$$\Pi_{A_1, \dots, A_m} (\text{JOIN}(R1, R2, D)) = \Pi_{A_1, \dots, A_m} (\text{JOIN}(\Pi_{D, B_1, \dots, B_n}(R1), \Pi_{D, C_1, \dots, C_k}(R2), D)),$$

unde am notat prin $\text{JOIN}(R1, R2, D)$ operația de compunere naturală între $R1$ și $R2$ după atributul comun D .

Proprietatea 12. Compunerea selecției cu operația *join*:

$$\sigma_{\text{cond}} (\text{JOIN} (R1, R2, D)) = \sigma_{\text{cond}} (\text{JOIN} (\Pi_{D, A} (R1), \Pi_{D, A} (R2), D)),$$

unde A reprezintă atributele care apar în condiția după care se face selecția.

Proprietățile operatorilor algebrei relaționale

Reguli de optimizare frecvent folosite:

- **Regula de optimizare 1. Selecțiile se execută cât mai devreme posibil.** Motivația acestei reguli este că selecțiile reduc substanțial dimensiunea relațiilor. Regula de transformare 4 poate fi folosită pentru a separa două sau mai multe selecții în selecții individuale care pot fi distribuite *join*-ului sau produsului cartezian folosind comutarea selecției cu *join*-ul.
- **Regula de optimizare 2. Produsurile carteziene se înlocuiesc cu *join*-uri, ori de câte ori este posibil.** Un produs cartezian între două relații este de obicei mult mai scump (ca și cost) decât un *join* între cele două relații, deoarece primul generează concatenarea tuplurilor în mod exhaustiv și poate genera un rezultat foarte mare. Această transformare se poate realiza folosind legătura dintre produs cartezian, *join* și selecție.

Proprietățile operatorilor algebrei relaționale

- **Regula de optimizare 3. Dacă sunt mai multe *join*-uri atunci cel care se execută primul este cel mai restrictiv.**
 - Un *join* este mai restrictiv decât altul dacă produce o relație mai mică.
 - Se poate determina care *join* este mai restrictiv pe baza selectivității sau cu ajutorul informațiilor statistice.
 - Algebric, acest lucru se poate realiza folosind regula de transformare 2.
- **Regula de optimizare 4. Proiecțiile se execută la început pentru a îndepărta attributele nefolositoare.**
 - Dacă un atribut al unei relații nu este folosit în operațiile ulterioare atunci trebuie îndepărtat. În felul acesta se va folosi o relație mai mică în operațiile ulterioare.
 - Aceasta se poate realiza folosind comutarea proiecției cu *join*-ul.