

Interogarea bazelor de date relationale cu operatori din algebra relationala

Ioana Ciuciu

oana@cs.ubbcluj.ro

<http://www.cs.ubbcluj.ro/~oana/>

Planificare

Saptamana	Curs	Seminar	Laborator
S1	1. Concepte fundamentale ale bazelor de date	1. Modelul Entitate-Relatie. Modelul relational	1. Modelarea unei BD in modelul ER si implementarea ei in SQL Server
S2	2. Modelare conceptuala		
S3	3. Modelul relational de organizare a bazelor de date	2. Limbajul SQL – definirea si actualizarea datelor	2. Interogari SQL
S4	4. Gestiunea bazelor de date relationale cu limbajul SQL		
S5-6	5-6. Dependente functionale, forme normale	3. Limbajul SQL – regasirea datelor	3. Interogari SQL avansate
S7	7. Interogarea bazelor de date relationale cu operatori din algebra relationala	4. Proceduri stocate	
S8	8. Structura fizica a bazelor de date relationale		
S9-10-11	9-10-11. Indeksi. Arbori B. Fisiere cu acces direct	5. View-uri. Functii definite de utilizator. Executie dinamica	4. Proceduri stocate. View. Trigger
S12	12. Evaluarea interogarilor in bazele de date relationale	6. Formele normale ale unei relatii. Indeksi	
S13	13. Extensii ale modelului relational si baze de date NoSQL	7. Probleme	Examen practic
S14	14. Aplicatii		

Planul cursului

- ▶ Ce este algebra relationala
- ▶ Tipurile de conditii
- ▶ Operatori unari
- ▶ Operatori binari
- ▶ Exemple



Ce este algebra relationala

- ▶ Algebra relationala definește **operatii asupra relatiilor**
- ▶ În majoritatea sistemelor relationale, răspunsul la o interogare se obține prin utilizarea unuia sau a mai multor **operatori relationali**



Operatorii relationali

- ▶ Operatorii folositi in algebra relationala se impart in doua mari categorii:
 - ▶ Operatori care se aplica **unei relatii**: operatori unari
 - ▶ Operatori care se aplica asupra a **doua relatii**: operatori binari



Operatorii relationali

▶ Operatori unari:

- ▶ Selectia
- ▶ Proiectia

▶ Operatori binari:

- ▶ Reuniunea
 - ▶ Intersectia
 - ▶ Diferenta
 - ▶ Produsul
 - ▶ Catul
- } Teoria multimilor



Tipurile de conditii

- ▶ **Conditiiile se evalueaza pentru fiecare inregistrare (linie) din relatie (tabel)**
- ▶ Sunt asemanatoare cu conditiile de filtrare din instructiunile **select-sql**
- ▶ **Tipurile de conditii** ce apar in cadrul diferitelor **operatori relationali**:
 - ▶ I. Pentru a verifica daca un atribut îndeplinește o conditie simpla se face compararea acestuia cu o anumita valoare, sub forma:
`nume_atribut operator_relational valoare`



Tipurile de conditii

- ▶ **Tipurile de conditii** ce apar in cadrul diferitelor operatori relationali:
- ▶ 2. O relatie cu o singura coloana poate fi considerata ca o **multime**. Urmatoarea conditie testeaza daca o anumita valoare **apartine** sau nu unei multimi:

nume_atribut $\left\{ \begin{array}{l} \text{IS IN} \\ \text{IS NOT IN} \end{array} \right\}$ relatie_cu_o_coloana



Tipurile de conditii

- ▶ **Tipurile de conditii** ce apar in cadrul diferitelor **operatori relationali**:
- ▶ 3. Doua relatii (considerate ca multimi de inregistrari) se pot compara prin **operatiile de egalitate, diferit, incluziune, neincluziune**.
- ▶ Intre doua relatii cu acelasi numar de coloane si cu aceleasi tipuri de date pentru coloane (deci intre doua multimi comparabile) putem avea conditii de tipul urmator:

$$\text{relatie} \left[\begin{array}{c} \text{IS IN} \\ \text{IS NOT IN} \\ = \\ <> \end{array} \right] \text{relatie}$$



Tipurile de conditii

- ▶ **Tipurile de conditii** ce apar in cadrul diferitelor **operatori relationali**:
- ▶ Conditie este si oricare din constructiile urmatoare:

(conditie)
NOT conditie
conditie1 AND conditie2
conditie1 OR conditie2



Tipurile de conditii

Observatii:

- ▶ In primul tip de conditie apare constructia '*valoare*', care poate fi una din tipurile urmatoare:
 - ▶ Pentru fiecare constructie se ia pentru *valoare* o anumita relatie curenta, care rezulta din contextul in care apare aceasta
- ▶ **nume_atribut** - care precizeaza valoarea atributului dintr-o inregistrare curenta. Daca precizarea numai a numelui atributului creeaza ambiguitate (exista mai multe relatii curente care contin cate un atribut cu acest nume), atunci se va face o calificare a atributului cu numele relatiei sub forma: **relatie.nume_atribut**
- ▶ **expresie** - se evalueaza expresia, iar daca apar si denumiri de attribute, atunci acestea se iau dintr-o inregistrare curenta



Tipurile de conditii

Observatii:

- ▶ **COUNT(*) FROM relatie** - precizeaza numarul de inregistrari din relatia specificata

▶ $\left\{ \begin{array}{c} \text{IS IN} \\ \text{IS NOT IN} \\ = \\ <> \end{array} \right\} ([\text{DISTINCT}] \text{ expresie}) \text{ FROM relatie}$

care determina o valoare folosind toate inregistrările din relatia curenta.

La determinarea acestei valori se iau toate valorile atributului precizat ca argument (din toate inregistrările), sau numai valorile distincte, dupa cum lipseste sau apare cuvantul DISTINCT. Valorile astfel determinate sunt: numarul de valori (pentru COUNT), suma acestor valori (apare SUM, valorile trebuie sa fie numerice), valoarea medie (apare AVG, valorile trebuie sa fie numerice), valoarea maxima (apare MAX), respectiv valoarea minima (apare MIN).



Operatori unari: Selectia

- ▶ **Selectia** unei relatii **R** – determina o noua relatie ce are aceeaasi schema cu a relatiei **R**
- ▶ Din relatia **R** se iau numai inregistrarile care indeplinesc o conditie **c**
- ▶ Notatie:

$$\sigma_c(R)$$

- ▶ Echivalent SQL: `select * from R where c`



Operatori unari: Selectia

Exemplu

- Fie relatia Comanda

nrComanda	Data	Suma
1013	Octombrie	685
1089	Octombrie	1300
1130	Noiembrie	115
1200	Decembrie	512

- Selectia permite raspunsul la intrebari cum sunt:

- care sunt comenzile plasate dupa luna octombrie?

$$\sigma_{(Data > Octombrie)}(Comanda)$$

nrComanda	Data	Suma
1130	Noiembrie	115
1200	Decembrie	512

- care sunt comenzile plasate dupa luna octombrie, mai mari de 500 lei?

$$\sigma_{(Data > Octombrie \text{ and } Suma > 500)}(Comanda)$$

nrComanda	Data	Suma
1200	Decembrie	512

Operatori unari: Proiectia

- ▶ **Proiectia** unei relatii **R** – determina o relatie noua ce are attributele precizate printr-o multime α de attribute
- ▶ Din fiecare inregistrare a unei relatii R se determina numai valorile atributelor incluse in mulimea α
- ▶ Multimea α de attribute **se poate extinde** la o **multime de expresii** (in loc de o multime de attribute), care precizeaza coloanele relatiei care se construiesc
- ▶ Notatie:

$$\Pi_{\alpha} (R)$$

- ▶ Echivalent SQL: `select all α from R`



Operatori unari: Proiectia

Exemplu

- Proiectia relatiei Comanda pe nrComanda, Data:

nrComanda	Data
1013	Octombrie
1089	Octombrie
1130	Noiembrie
1200	Decembrie

$\pi_{nrComanda, Data}(Comanda)$

- Proiectia relatiei Comanda pe nrComanda, Suma:

nrComanda	Suma
1013	685
1089	1300
1130	115
1200	512

$\pi_{nrComanda, Suma}(Comanda)$



Operatori binari: Reuniunea, diferenta si intersectia

► Reuniunea, diferenta si intersectia a doua relatii:

► Notatie: $R_1 \cup R_2$ $R_1 - R_2$ $R_1 \cap R_2$

► Cele doua relatii trebuie sa aibe aceeasi schema (sau scheme 'compatibile')

► Echivalent SQL:

`select * from R1 union all select * from R2`

`select * from R1 except select * from R2`

`select * from R1 intersect select * from R2`



Operatori binari: Produsul cartezian

- ▶ **Produsul cartezian** a doua relatii – este o noua relatie continand attributele ambelor relatii implicate
- ▶ Se construiește facand toate combinatiile posibile intre inregistrarile celor doua relatii
- ▶ Notatie
 - ▶ Fie $R(X)$ si $S(Y)$ doua relatii unde X si Y sunt multimi disjuncte
 - ▶ Produsul cartezian al lui $R(X)$ si $S(Y)$ se noteaza:

$$R \times S$$



Operatori binari: Produsul cartezian

Exemplu

Fie relatia Carte:

Titlu	Autor
X	Toto
Y	Lulu

Fie relatia Editie:

Culoare	Editie
Rosu	Lux
Alb	Paperback
Verde	Cartonata

Produsul cartezian permite asocierea titlurilor, autorilor, culorilor si editiilor:

Titlu	Autor	Culoare	Editie
X	Toto	Rosu	Lux
X	Toto	Alb	Paperback
X	Toto	Verde	Cartonata
Y	Lulu	Rosu	Lux
Y	Lulu	Alb	Paperback
Y	Lulu	Verde	Cartonata

Operatori binari: Operatori Join

Exista mai multi operatori join

- ▶ **Joinul conditional** sau **theta join** - care determina acele inregistrari din produsul cartezian al celor doua relatii care indeplinesc o anumita conditie

- ▶ Notatie: $R_1 \otimes_{\theta} R_2$

- ▶ Din definitie se observa ca avem: $R_1 \otimes_{\theta} R_2 = \sigma_{\theta}(R_1 \times R_2)$

- ▶ Echivalent:

```
select * from R1 inner join R2 on  $\theta$ 
```



Operatori binari: Operatori Join

Exista mai multi operatori join

- ▶ **Joinul natural** - care determina o relatie noua ce are ca attribute reuniunea atributelor din cele doua relatii, iar inregistrarile se obtin din toate perechile de inregistrari ale celor doua relatii care au aceleasi valori pentru attributele comune
- ▶ Notatie: $R_1 * R_2$
- ▶ Daca cele doua relatii au schemele

$$R_1[\alpha], R_2[\beta], \text{ si } \alpha \cap \beta = \{\alpha_1, \alpha_2, \dots, \alpha_m\}$$

, atunci joinul natural se poate calcula prin constructia urmatoare:

$$R_1 * R_2 = \Pi_{\alpha \cup \beta} \left(R_1 \otimes_{R_1.A_1=R_2.A_1 \text{ and } \dots \text{ and } R_1.A_m=R_2.A_m} R_2 \right)$$

- ▶ Echivalent:

```
select * from R1 natural join R2
```



Operatori binari: Operatori Join

Exista mai multi operatori join

- ▶ **Joinul extern stanga** - determina o relatie noua ce are ca attribute concatenarea^R atributelor din cele doua relatii, iar inregistrarile se obtin astfel: se iau inregistrarile care se obtin prin joinul conditional $R_1 \otimes_C R_2$ la care se adauga inregistrarile din R1 care nu s-au folosit la acest join conditional combinate cu valoarea *null* pentru toate attributele corespunzatoare relatiei R2

- ▶ Notatie: $R1 \bowtie_C R2$

- ▶ Echivalent:

```
select * from R1 left outer join R2 on Ⓜ
```



Operatori binari: Operatori Join

Exista mai multi operatori join

- ▶ **Joinul extern dreapta** - se obtine asemenea ca joinul extern stanga, dar la inregistrarile din $R_1 \otimes_C R_2$ se adauga inregistrarile din R2 care nu s-au folosit la acest join conditional combinate cu valoarea *null* pentru toate attributele corespunzatoare relatiei R1
- ▶ Notatie: $R1 \bowtie_C R2$
- ▶ Echivalent:
`select * from R1 right outer join R2 on (H)`



Operatori binari: Operatori Join

Exista mai multi operatori join

► **Joinul extern complet** - este reuniunea rezultatelor obinute cu join extern stanga si join extern dreapta

► Notatie: $R1 \bowtie_C R2$

► Echivalent:

`select * from R1 full outer join R2 on Θ`



Operatori binari: Operatori Join

Exista mai multi operatori join

- ▶ **Semi joinul stanga** - determina o relatie noua ce
contine inregistrarile din R_1 utile pentru joinul natural $R_1 * R_2$
- ▶ Notatie: $R_1 \bowtie R_2$
- ▶ **Semi joinul dreapta** - contine inregistrarile din R_2 utile
pentru joinul natural $R_1 * R_2$
- ▶ Notatie: $R_1 \ltimes R_2$



Operatori binari: Catul

- ▶ **Catul** – pleaca de la doua relatii $R_1[\alpha]$ si $R_2[\beta]$, cu $\beta \subset \alpha$,
- ▶ Deducem ca attributele din cat sunt date de multimea $\alpha - \beta$
- ▶ Notatie: $R_1 \div R_2 [\alpha - \beta]$
- ▶ O inregistrare r apartine $R_1 \div R_2$ daca $\forall r_2 \in R_2, \exists r_1 \in R_1$ ce indeplineste conditiile:

1. $\Pi_{\alpha - \beta}(r_1) = r$
2. $\Pi_{\beta}(r_1) = r_2$

Obs. O inregistrare **r** apartine catului
daca in relatia **R1** apar toate
concatenarile dintre aceasta inregistrare
si fiecare inregistrare din **R2**

Operatori binari: Catul

Exemplu

Fie relatia urmatoare:

Nume_stud	Nume_prof
Toto	Paul
Fifi	Petru
Lulu	Paul
Riri	Florin
Toto	Petru
Fifi	Paul
Lulu	Florin

Fie relatia urmatoare:

Nume_stud
Toto
Fifi

Catul permite raspunsul la intrebarea: Care este numele profesorilor care predau tuturor elevilor din a doua relatie?

Rezultatul:

Nume_prof
Paul
Petru

Obs.:

- Catul permite obtinerea inregistrarilor din relatia 1 care sunt asociate tuturor inregistrarilor din relatia 2
- O relatie se divide prin a doua relatie, continand exclusiv attribute din prima relatie

Exemple

- ▶ Pentru a putea da cateva exemple de utilizare a acestui limbaj de interogare vom introduce urmatoarele instructiuni:
- ▶ **Atribuirea:** unei variabile **R[lista]** (care va desemna o relatie si eventuala denumire a coloanelor) ii vom atribui o relatie data printr-o expresie construita cu operatorii de mai sus:

R[lista] := expresie

- ▶ **Eliminarea duplicarilor** unei relatii: $\partial (R)$
- ▶ **Sortarea** inregistrarilor dintr-o relatie: $S_{\{lista\}}(R)$
- ▶ **Gruparea** : $G_{\{lista1\} \text{ group by } \{lista2\}}(R)$, care este o **extensie** pentru proiectie
 - ▶ Inregistrările din R sunt grupate dupa coloanele din *lista2*, iar pentru un grup de inregistrari cu aceleasi valori pentru *lista2* se evalueaza *lista1* (unde pot aparea si *functii de grupare*).



Exemple

► Fie urmatoarea baza de date relationala:

studenti [cod, nume, grupa, media, datanasterii]

grupe [cod, an, sectia]

orar [zi, orainc, orasf, tipactiv, sala, grupa, cod_cadrudid]

cadredid [cod, nume]

1. Numele studentilor dintr-o grupa data

$$R := \Pi_{\{nume\}} \left(\sigma_{grupa='281'} (studenti) \right)$$

select nume from studenti where grupa='281'



Exemple

2. Studentii de la o anumita sectie (lista alfabetica pe grupe)

$$G := \Pi_{\{cod\}} \left(\sigma_{sectia='S'} (grupe) \right)$$

$$R := S_{\{grupa, nume\}} \left(\sigma_{grupa-is-in-G} (studenti) \right)$$

```
select grupa,nume
from studenti
where grupa is in (select cod from grupe where sectia='S')
```



Exemple

3. Numărul de studenți din fiecare grupă a unei secții date:

$$ST := \sigma_{\text{grupa is in } \left[\Pi_{\{cod\}} \left(\sigma_{\text{sectia}='S'}(\text{grupe}) \right) \right]}(\text{studenti})$$

$$NR := \gamma_{\{grupa, count(*)\} \text{ group by } \{grupa\}}(ST)$$

```
select grupa, count(*)
from
    (select * from studenti
     where grupa is in (select cod from grupe where sectia='S')
    )
group by grupa
```

Explicatie pas cu pas (ex.3):

$$R_1 = \sigma_{\text{sectia}='S'}(\text{grupe}) \quad \text{- Toate grupele din sectia 'S'}$$

$$R_2 = \Pi_{\{cod\}} R_1 \quad \text{- Codurile corespunzatoare grupelor din sectia 'S'}$$

$$R_3 = \sigma_{\{grupa \text{ is in } R_2\}}(\text{studenti}) \quad \text{- Studentii din sectia 'S'}$$

$$NR = G_{\{grupa, count(*)\} \text{ group-by } \{grupa\}} R_3 \quad \text{- Numarul de studenti din fiecare grupa a sectiei 'S'}$$

Exemple

4. Orarul unui student dat

$$T := \sigma_{\text{grupa is in } \left(\Pi_{\{grupa\}} \left(\sigma_{\text{nume} = \text{'nume_dat'}}(studenti) \right) \right)}(orar)$$

5. Numărul orelor, dintr-o săptămână, pentru fiecare grupă

$$F(nr, grupa) := \Pi_{\{orasf - orainc, grupa\}}(orar)$$

$$NrOre(grupa, nr) := \gamma_{\{grupa, sum(nr)\}} \text{ group by } \{grupa\} (F)$$



Example

6. Cadrele didactice (numele acestora) care au ore cu un anumit student

$$A := \left(\sigma_{\text{nume} = \text{'nume_dat'}}(\text{studenti}) \right) \otimes \text{studenti.grupa} = \text{orar.grupa} (\text{orar})$$

$$B := \Pi_{\{\text{cadrudid}\}}(A)$$

S')

$$C := \text{cadredid} \otimes_{\text{cadredid.cod} = B.\text{cadrudid}} (B)$$

$$D := \Pi_{\{\text{nume}\}}(C)$$

7. Cadrele didactice fără ore trecute în orar

$$C := \Pi_{\{\text{nume}\}}(\text{cadredid}) - \Pi_{\{\text{nume}\}}(\text{orar} \otimes_{\text{orar.cadrudid} = \text{cadredid.cod}} \text{cadredid})$$



Example

8. Studenții care au ore în fiecare zi din săptămână (în toate zilele în care sunt activități didactice)

$$A := \delta\left(\Pi_{\{zi\}}(orar)\right), \text{ zilele cu ore}$$

$$B := \text{studenti} \otimes_{\text{studenti.grupa} = \text{orar.grupa}} \text{orar}$$

$$C := \Pi_{\{nume, zi\}}(B)$$

$$D := C \div A$$



Cursul urmator

- ▶ Curs 8 - Structura fizica a bazelor de date relationale



Bibliografie

- ▶ [Da04] DATE, C.J., An Introduction to Database Systems (8th Edition), Addison-Wesley, 2004, *cap. 7*.
- ▶ [Ga08] GARCIA-MOLINA, H., ULLMAN, J., WIDOM, J., *Database Systems: The Complete Book*, Pearson Prentice Hall, 2008], *cap. 5*.
- ▶ [Ra07] RAMAKRISHNAN, R., Database Management Systems. McGraw-Hill, 2007, *cap. 4*,

<http://pages.cs.wisc.edu/~dbbook/openAccess/thirdEdition/slides/slides3ed.html>

<http://infolab.stanford.edu/~ullman/fcdb/slides/slides5.pdf>

- ▶ Leon Tambulea, UBB, curs de baze de date
- ▶ Christine Verdier, Sarah Bouchenak , UJF Grenoble, curs Systèmes d'Information et Bases de données

