

# Modelul Relațional

Algebra Relațională – partea 2

Calcul relațional

Query By Example - QBE

# Algebra Relațională – partea 2

# Semijoin

- Se utilizează de către SGBD-uri pe baze de date distribuite și se notează:

$$A \bowtie B$$

- Se presupune că cele două relații A și B ce trebuie cuplate se găsesc la locații diferite.
- Pentru optimizarea traficului în rețea este mai convenabil să se transporte doar mulțimea de valori corespunzătoare atributului de join de la un site la altul, se efectuează join și la urmă probabil la o a treia locație se transportă fracțiunile din cele două relații pentru a obține rezultatul final.

# Semijoin

$$A \bowtie B$$

- După notația de mai sus se transportă mai întâi de la locația de reședință a relației B la locația de reședință a locației A.
- De exemplu A este **Student(nr\_matr, nume, gen, coda)** și B este **Nota(nr\_matr, code, nota)**. Se transportă doar *nr\_matr* din **Nota** la locația unde se găsește **Student**.

# Diviziunea

- **Definiție:** Diviziunea relației  $A$  de grad  $m$  prin relația  $B$  de grad  $n$ , notată  $A \div B$ , este o relație  $R$  de grad  $m-n$ , formată din mulțimea tupelor  $r$  cu proprietatea că pentru orice tuplă  $b$  din  $B$  există o tuplă  $a$  în  $A$  egală cu rezultatul concatenării tupelor  $r$  și  $b$ .
- Mulțimea atributelor relației  $B$  trebuie să fie o submulțime a mulțimii atributelor relației  $A$ .
- Relația  $R$  conține doar acele atribute din  $A$  ce nu apar în  $B$ .
- O tuplă din relația  $A$  este reținută pentru rezultat dacă și numai dacă este legată de fiecare tuplă din relația  $B$ .

# Exemplu: Diviziunea

**A**

| Nr_matr | Nume   | Prenume   | Data_n     | Nota | Disciplina         |
|---------|--------|-----------|------------|------|--------------------|
| 1       | Pop    | Victor    | 31.12.1980 | 8    | Baze de Date       |
| 1       | Pop    | Victor    | 31.12.1980 | 9    | Programare         |
| 2       | Albert | Alexandru | 04.01.1979 | 7    | Baze de Date       |
| 2       | Albert | Alexandru | 04.01.1979 | 9    | Programare         |
| 2       | Albert | Alexandru | 04.01.1979 | 8    | Sisteme de Operare |
| 3       | Pop    | Mariana   | 12.03.1982 | 6    | Baze de Date       |
| 3       | Pop    | Mariana   | 12.03.1982 | 8    | Programare         |
| 3       | Pop    | Mariana   | 12.03.1982 | 10   | Sisteme de Operare |
| 3       | Pop    | Mariana   | 12.03.1982 | 7    | Analiza Matematica |
| 4       | Marcu  | Ioana     | 07.04.1983 | 8    | Baze de Date       |
| 4       | Marcu  | Ioana     | 07.04.1983 | 9    | Programare         |
| 5       | Vasile | Mircea    | 27.03.1981 | 7    | Baze de Date       |
| 5       | Vasile | Mircea    | 27.03.1981 | 8    | Programare         |
| 5       | Vasile | Mircea    | 27.03.1981 | 9    | Sisteme de Operare |

**B**

| Nota | Disciplina   |
|------|--------------|
| 8    | Baze de Date |
| 9    | Programare   |

$$R = A \div B$$

**R**

| Nr_matr | Nume  | Prenume | Data_n     |
|---------|-------|---------|------------|
| 1       | Pop   | Victor  | 31.12.1980 |
| 4       | Marcu | Ioana   | 07.04.1983 |

$$A \div B = \pi_X(A) - \pi_X(((\pi_X(A)) \times B) - A)$$

| A      |    | B  |  |
|--------|----|----|--|
| X      | Y  | Z  |  |
| Pop    | 8  | 8  |  |
| Pop    | 7  | 7  |  |
| Pop    | 10 | 10 |  |
| Man    | 9  |    |  |
| Man    | 9  |    |  |
| Man    | 9  |    |  |
| Man    | 7  |    |  |
| Vesa   | 8  |    |  |
| Vesa   | 9  |    |  |
| Vesa   | 10 |    |  |
| Mircea | 7  |    |  |
| Mircea | 8  |    |  |
| Mircea | 10 |    |  |

$$A \div B = \pi_X(A) - \pi_X(((\pi_X(A)) \times B) - A)$$

| $\pi_X(A)$ |
|------------|
| X          |
| Pop        |
| Man        |
| Vesa       |
| Mircea     |

| $(\pi_X(A)) \times B$ |    |
|-----------------------|----|
| X                     | Z  |
| Pop                   | 8  |
| Pop                   | 7  |
| Pop                   | 10 |
| Man                   | 8  |
| Man                   | 7  |
| Man                   | 10 |
| Vesa                  | 8  |
| Vesa                  | 7  |
| Vesa                  | 10 |
| Mircea                | 8  |
| Mircea                | 7  |
| Mircea                | 10 |

| $((\pi_X(A)) \times B) - A$ |      |
|-----------------------------|------|
| X                           | Y(Z) |
| Man                         | 8    |
| Man                         | 10   |
| Vesa                        | 7    |



$$A \div B = \pi_X(A) - \pi_X(((\pi_X(A)) \times B) - A)$$

$$\pi_X(((\pi_X(A)) \times B) - A)$$

|      |
|------|
| X    |
| Man  |
| Vesa |

$$\pi_X(A) - \pi_X(((\pi_X(A)) \times B) - A)$$

|        |
|--------|
| X      |
| Pop    |
| Mircea |

# Limitări ale Algebrei Relaționale

## □ Tratarea recursivității

- De exemplu o tuplă în relația *Traseu(plecare, sosire, nr\_tren)* conține date despre două stații adiacente din mersul trenurilor
- Pentru a răspunde la întrebarea “Care este traseul între două localități pentru un tren cu *nr\_tren dat?*” este nevoie de o succesiune nedeterminabilă inițial de pași
- La un pas se cuplează după predicatul  $plecare = sosire \wedge nr\_tren = valoare$  relația *Traseu* cu ea însăși

# Limitari ale Algebrei Relationale

## TRASEU

| Plecare   | Sosire      | Nr_tren |
|-----------|-------------|---------|
| Bucuresti | Ploiesti    | 311     |
| Ploiesti  | Sinaia      | 311     |
| Sinaia    | Predeal     | 311     |
| Predeal   | Brasov      | 311     |
| Brasov    | Cluj-Napoca | 311     |

## R1

| Plecare   | Sosire   |
|-----------|----------|
| Bucuresti | Ploiesti |

## R2

| Plecare1  | Sosire1  | Sosire2 |
|-----------|----------|---------|
| Bucuresti | Ploiesti | Sinaia  |

## R3

| Plecare1  | Sosire1  | Sosire2 | Sosire3 |
|-----------|----------|---------|---------|
| Bucuresti | Ploiesti | Sinaia  | Predeal |

## R4

| Plecare1  | Sosire1  | Sosire2 | Sosire3 | Sosire4 |
|-----------|----------|---------|---------|---------|
| Bucuresti | Ploiesti | Sinaia  | Predeal | Brasov  |

## R5

| Plecare1  | Sosire1  | Sosire2 | Sosire3 | Sosire4 | Sosire5     |
|-----------|----------|---------|---------|---------|-------------|
| Bucuresti | Ploiesti | Sinaia  | Predeal | Brasov  | Cluj-Napoca |

# Calcul relational

# Calcul Relațional - Caracteristici

- neprocedural sau *declarativ*
- descrie răspunsul la interogare fără a da detalii explicite despre cum va fi executată interogarea

# Calcul Relațional - Variante

- Calcul Relațional al Tuplelor (CRT)
  - Variabilele din CRT sunt variabile de tuplă.
  - Din acest punct de vedere SQL se aseamănă cu CRT.
- Calcul Relațional al Domeniilor (CRD)
  - Variabilele din CRD sunt variabile de domeniu.
  - Din acest punct de vedere QBE se aseamănă cu CRD.

# CRT

- *Variabila de tuplă* este o variabilă ale cărei valori sunt tuple ale unei scheme particulare de relație.
- Fiecare valoare atribuită unei variabile de tuplă are același număr și același tip de attribute.

# Interogare CRT

- Are forma  $\{ T \mid p(T) \}$ , unde  $T$  este o variabilă de tuplă, iar  $p(T)$  denotă o *formulă* ce descrie  $T$ .
  - $T$  este singura variabilă liberă din formula  $p$
- Rezultatul interogării este setul tuplelor  $t$  pentru care formula  $p(T)$  este evaluată "true" prin înlocuirea  $T = t$ .
- Limbajul ce descrie formulele  $p(T)$  este esența CRT și este un subset simplu al "*first-order logic*".



# Exemplu: Interogare CRT ce folosește o singură relație

- Folosind **Beers(name, manf)**, care sunt mărcile de bere produse de Anheuser-Busch?

$$\{ B.name \mid B \in \text{Beers} \wedge B.manf = 'Anheuser-Busch' \}$$

- Răspunsul conține componenta “name” pentru acele instanțe ale lui  $B$  ce trec testul.

# CRT – Sintaxa Interogării

- *Rel* - nume relație
- *R și S* - variabile de tuplă
  - *a* - un atribut din *R*
  - *b* - un atribut din *S*
- *op* - un operator din mulțimea  $\{<, \leq, \neq, \geq, >\}$

# Formulă atomică

- $R \in Rel$
- $R.a \text{ op } S.b$
- $R.a \text{ op } constantă$
- $constantă \text{ op } R.a$

# Formulă CRT

- Se definește recursiv ca fiind una din următoarele, unde  $p$  și  $q$  sunt formule și  $p(R)$  denotă o formulă în care apare variabila  $R$ :
  - orice formulă atomică
  - $\neg p, p \vee q, p \wedge q, p \Rightarrow q$
  - $\exists R( p(R))$ , unde  $R$  este o variabilă de tuplă
  - $\forall R( p(R))$ , unde  $R$  este o variabilă de tuplă

# Formulă CRT

- În ultimele două clauze, cuantificatorii  $\forall$  și  $\exists$  se spune că "*leagă*" variabila  $R$ .
- O variabilă se spune că este "*liberă*" într-o formulă sau subformulă (o formulă conținută într-o formulă mai mare) dacă (sub)formula nu conține un cuantificator care să o lege.

# Formulă CRT

- Se observă că fiecare variabilă într-o formulă CRT apare într-o subformulă ce este atomică, și fiecare schemă de relație specifică un domeniu pentru fiecare atribut.
- Asigură că fiecare variabilă într-o formulă CRT are un domeniu bine definit ("well-defined") din care sunt luate valorile variabilei.
  - Fiecare variabilă are un *tip* bine definit ("well-defined").

# Exemplu: Interogare CRT ce folosește două relații

- Se folosesc relațiile *Likes(drinker, beer)* și *Frequents(drinker, bar)*, pentru a găsi băuturile preferate de cel puțin o persoană ce frecventează "Joe's Bar".

$$\{P \mid \exists L \in \text{Likes} \wedge \exists F \in \text{Frequents} \\ (F.\text{bar} = \text{'Joe's Bar'} \wedge L.\text{drinker} \\ = F.\text{drinker} \wedge P.\text{beer} = L.\text{beer}) \}$$

# CRD

- *O variabilă de domeniu* ia valori din domeniul unui anumit atribut.
- O interogare CRD este de forma  $\{ \langle x_1, x_2, \dots, x_n \rangle \mid p(\langle x_1, x_2, \dots, x_n \rangle) \}$ .
- Unde fiecare  $x_i$  este sau o *variabilă de domeniu* sau o constantă și  $p(\langle x_1, x_2, \dots, x_n \rangle)$  denotă o *formulă CRD* cu unicele variabile libere  $x_i, 1 \leq i \leq n$ .
- Rezultatul interogării este mulțimea tuplelor  $\{ \langle x_1, x_2, \dots, x_n \rangle$  pentru care formula este evaluată "true".



# Formulă CRD

- Este definită asemănător cu formula CRT.
- Diferența constă în faptul că variabilele sunt variabile de domeniu.
- $op$  - un operator din mulțimea  $\{<, \leq, \neq, \geq, >\}$
- $X$  și  $Y$  sunt variabile de domeniu

# Formulă atomică CRD

- $\langle x_1, x_2, \dots, x_n \rangle \in Rel$ 
  - unde  $Rel$  este o relație cu  $n$  attribute
  - fiecare  $x_i$ ,  $1 \leq i \leq n$  este fie o variabilă fie o constantă
- $X \text{ op } Y$
- $X \text{ op } constantă$
- $constantă \text{ op } X$

# Formulă CRD

- Se definește recursiv una din următoarele, unde  $p$  și  $q$  sunt formule și  $p(X)$  denotă o formulă în care apare variabila  $X$ .
  - Orice formulă atomică
  - $\neg p, p \vee q, p \wedge q, p \Rightarrow q$
  - $\exists R( p(R))$ , unde  $R$  este o variabilă de domeniu
  - $\forall R( p(R))$ , unde  $R$  este o variabilă de domeniu

# Exemplu: Interogare CRD ce folosește o singură relație

- Folosind **Beers(name, manf)**, care sunt mărcile de bere produse de Anheuser-Busch?

$$\{ \langle N \rangle \mid \langle N, M \rangle \in \text{Beers} \wedge M = \text{'Anheuser-Busch'} \}$$

# Exemplu: Interogare CRD ce folosește două relații

- Se folosesc relațiile **Likes(drinker, beer)** și **Frequents(drinker, bar)**, pentru a găsi băuturile preferate de cel puțin o persoană ce frecventează "Joe's Bar".

$$\{ \langle B_1 \rangle \mid \exists \langle D_1, B_1 \rangle \in \text{Likes} \wedge \\ \exists \langle D_2, B_2 \rangle \in \text{Frequents} \\ (B_2 = \text{'Joe's Bar'} \wedge D_1 = D_2) \}$$

# Interogări nesigure

- Se consideră interogarea:

$$\{S \mid \neg(S \in Sells)\}$$

- Sintactic este corectă.
- Mulțimea tuplelor  $S$  este evident infinită, în contextul domeniilor infinite.
- Ideea este de a restricționa calculul relațional pentru a nu permite interogări nesigure.

# Formule independente de domeniu

□  $\text{DOM}(F)$  unde  $F$  este o formulă din calculul relațional, este reuniunea dintre mulțimea constantelor ce apar în  $F$  și mulțimea tuturor valorilor de attribute ce apar în relațiile specificate ca parametri în  $F$ .

□ Fie  $F(X_1, X_2, \dots, X_n)$  o formulă și  $D$  o mulțime de valori astfel încât  $\text{DOM}(F) \subseteq D$

Relația corespunzătoare formulei  $F$  în raport cu mulțimea de valori  $D$  este mulțimea tuplelor

$$(x_1, x_2, \dots, x_n) \in D \times D \times \dots \times D$$

astfel încât prin substituirea fiecărei variabile de domeniu  $X_i$  cu valoarea corespunzătoare  $x_i$ ,  $F(X_1, X_2, \dots, X_n)$  devine adevărată.

# Formule independente de domeniu

□ Formula  $F$  este independentă de domeniu dacă relația corespunzătoare acesteia în raport cu orice  $D \supseteq \text{DOM}(F)$  nu depinde de mulțimea de valori  $D$ .

□ Exemplu: Fie  $F(X,Y) = \neg R(X,Y)$  și fie  $R(X,Y) = \{(a,a), (b,b)\} \rightarrow \text{DOM}(F) = \{a,b\}$

Fie  $D_1 = \{a,b\}$  și  $D_2 = \{a,b,c\}$

$R_1(X,Y) = \{(a,b), (b,a)\}$

$R_2(X,Y) = \{(a,b), (b,a), (a,c), (c,a), (b,c), (c,b)\}$

$R_1 \neq R_2 \Rightarrow F(X,Y) = \neg R(X,Y)$  este dependentă de domeniu



# Formule sigure

- Independența de domeniu a unei formule  $F$  este o problemă indecidabilă.
- Formule sigure  $\subseteq$  Formule independente de domeniu.
- Proprietăți:
  - Orice formulă sigură este formulă independentă de domeniu.
  - Fiind dată o formulă  $F$ , se poate spune dacă este sau nu sigură.
  - Orice interogare din algebra relațională poate fi exprimată cu ajutorul formulelor sigure.

# Formule sigure

## □ Restricții:

- formula NU conține  $\forall$  (transformarea de echivalență  $(\forall X)F \equiv \neg(\exists X)\neg F$ )
- $F_1 \vee F_2 \Rightarrow F_1(X_1, X_2, \dots, X_n) \wedge F_2(X_1, X_2, \dots, X_n)$   
unde  $(X_1, X_2, \dots, X_n)$  sunt variabile libere
- orice variabilă liberă din  $F_1 \wedge F_2 \wedge \dots \wedge F_m$  trebuie să fie variabilă limitată:
  - variabilă liberă în cel puțin o  $F$  ce nu este comparație aritmetică sau negație
  - $F$  este de forma  $X = a$  sau  $a = X$ , unde  $a$  este o constantă
  - $F$  este de forma  $X = Y$  sau  $Y = X$ ,  $Y$  este limitată
- în  $F_1 \wedge F_2 \wedge \dots \wedge F_m$  cel puțin o  $F_i$  trebuie să fie pozitivă

# Query By Example (QBE)

# QBE

- M. Zloof. Query by Example. AFIPS, 44, 1975
- IBM
- editor grafic, mod ecran, interactiv

# QBE în SGBD actuale

- Microsoft Access - "Visual Query by Example,,
- Microsoft SQL Server Enterprise Manager
- SGBD object-oriented (db4o)

# QBE - caracteristici

- Derivă din CRD
- Permite toată suita de operații:
  - adăugare (I)
  - modificare (U)
  - ștergere (D)
  - interogare (P)

# Formularea interogărilor în QBE

- Pe ecran apar capetele de tabele asociate relațiilor.
- Pentru a exprima operația se utilizează litera în prima coloană (ce conține sus numele relației), excepție operatorul P.
- Un predicat simplu este exprimat prin prezența unei variabile sau a unei constante într-o coloană.

# Interogare cu predicat simplu

| STUDENT | Nr_matr | Nume | Prenume | Sex_f | Data_n | Obs |
|---------|---------|------|---------|-------|--------|-----|
| P.      |         |      |         | .T.   |        |     |

| STUDENT | Nr_matr | Nume | Prenume | Sex_f | Data_n | Obs |
|---------|---------|------|---------|-------|--------|-----|
|         | P.      | P.   | P.      | .T.   |        |     |

- Afișează detaliile studentelor (prima variantă)
- Afișează numărul matricol, numele și prenumele studentelor (varianta a doua)



# Interogare cu predicat complex (AND)

| STUDENT | Nr_matr | Nume | Prenume | Sex_f | Data_n       | Obs |
|---------|---------|------|---------|-------|--------------|-----|
| P.      |         |      |         | .T.   | {01/01/1985} |     |

| STUDENT | Nr_matr    | Nume | Prenume | Sex_f | Data_n       | Obs |
|---------|------------|------|---------|-------|--------------|-----|
|         | P._X<br>_X | P.   | P.      | .T.   | {01/01/1985} |     |

- Afișează detaliile studentelor născute înainte de anul 1985 (prima variantă)
- Afișează număr matricol, nume, prenume pentru studentele născute înainte de anul 1985 (varianta a doua)

# Interogare cu predicat complex (OR)

| STUDENT | Nr_matr      | Nume | Prenume | Sex_f | Data_n         | Obs |
|---------|--------------|------|---------|-------|----------------|-----|
|         | P._X<br>P._Y | P.   | P.      | .T.   | < {01/01/1985} |     |

- Afișează număr matricol, nume, prenume, cu condiția fie este studentă, fie data nașterii este înainte de anul 1985

# Join

| STUDENT | Nr_matr | Nume | Prenume | Sex_f | Data_n | Obs |
|---------|---------|------|---------|-------|--------|-----|
|         | P._X    | P.   | P.      | .T.   |        |     |

| NOTA | Nr_matr | Code | Nota |
|------|---------|------|------|
|      | _X      | _Y   | P.>8 |

| DISCIPLINA | Code | Titlu_mat | Profesor |
|------------|------|-----------|----------|
|            | _Y   | P.        |          |

- Afișează număr matricol, nume, prenume, titlu materie, nota pentru studentele cu note peste 8
- La Join participă trei relații

# Folosirea operatorului ALL

| EXAMEN | Code | Titlu_mat | Nume_prof |
|--------|------|-----------|-----------|
|        | _Y   |           | P.        |

| NOTA | Nr_matr | Code | Nota |
|------|---------|------|------|
|      | ALL._X  | _Y   |      |

| STUDENT | Nr_matr | Nume | Prenume | Sex_f | Data_n |
|---------|---------|------|---------|-------|--------|
|         | ALL._X  |      |         |       |        |

- Afișează numele profesorilor care au acordat note la toți studenții

# Operatorul de grupare G

| NOTA | Nr_matr | Code | Nota         |
|------|---------|------|--------------|
|      | G._X    |      | P.AVG.ALL._Y |

| STUDENT | Nr_matr | Nume | Prenume | Sex_f | Data_n |
|---------|---------|------|---------|-------|--------|
|         | P._X    | P.   | P.      |       |        |

| Conditions       |
|------------------|
| P.AVG.ALL._Y > 8 |

- AVG, CNT, MIN, MAX, SUM
- Afișează numărul matricol, numele, prenumele și media pentru studenții cu media peste 8
- Caseta de condiție acționează asemănător HAVING

# Adăugare I

| STUDENT | Nr_matr | Nume      | Prenume | Data_n       | Sex_f | Coda |
|---------|---------|-----------|---------|--------------|-------|------|
| I.      | 2174    | "Popescu" | "Radu"  | {12.02.1985} | .T.   | 4    |

- Se adaugă tupla nouă cu valorile precizate pentru attributele relației Student

# Modificare U

|      |         |      |      |
|------|---------|------|------|
| NOTA | Nr_matr | Code | Nota |
| U.   | _S      | _V   | _X+1 |
|      | _S      | _V   | _X   |

  

|            |      |              |           |
|------------|------|--------------|-----------|
| DISCIPLINA | Code | Titlu_mat    | Nume_prof |
|            | _V   | "Programare" |           |

  

|            |
|------------|
| Conditions |
| _X < 7     |

- Se modifică nota crescând cu 1 pentru studenții care au obținut note sub 7 indiferent de materie

# Ștergere D

| STUDENT | Nr_matr | Nume | Prenume | Data_n | Sex_f | Coda |
|---------|---------|------|---------|--------|-------|------|
| D.      | 3018    |      |         |        |       |      |
| STUDENT | Nr_matr | Nume | Prenume | Data_n | Sex_f | Coda |
| D.      | _X      |      |         |        |       | 1    |

- În prima variantă, se șterge din tabela Student studentul cu numărul matricol 3018
- În varianta a doua, se șterg studenții ce locuiesc la adresa cu codul adresei 1