BAZE DE DATE

Bibliografie

1. Abiteboul S., R.Hull, V.Vianu – *“Foundations of Databases”*, Addison-Wesley, Reading, MA, 1995.
2. Chandra, A. K., D. Harel – *“Structure and complexity of relational queries”*, J. Computer and Systems Sciences 25:1, pp.99-128.
3. Date, C.J. – *“Baze de date”*, traducere de Simona Preda si Titi Preda, editura PLUS, 2005.
4. Date, C.J., H. Darwen – *“A Guide to the SQL standard”*, Addison-Wesley, Reading, MA, 1997.

5. Date, C.J., Hugh Darwen – *“A Guide to the SQL Standard”*, Reading Mass., Addison-Wesley, 1997.
6. Date, C.J. – *“A Critique of the SQL Database Language”*, ACM SIGMOD Record 14, N.3, 1984.
7. Dubois, Paul – *“MySQL”* , traducere în limba română, Editura Teora, 2001.
8. England K. – *“Microsoft SQL Server 200 Performance Optimization and Tuning Handbook”*, Digital Press, 2001.
9. Fagin, R. – *“On an authorization mechanism”*, ACM Transactions on Database systems, 3:3, pp.310-319, 1978.
10. Fehily, C. – *“SQL visual quickstart guide”*, traducere în limba română, Editura B.I.C. ALL, 2004.

11. V.Felea: Baze de date relationale. Dependente. Editura Universitatii “Al.I.Cuza” Iasi, 1996
12. V.Felea, C.Matei si M.Balta: Interogarea bazelor de date, Aplicatii in Oracle si SQL Server, Editura MATRIXROM Bucuresti 2005.
13. V.Felea: Elemente ale implementarii modelului relational in Sisteme de Gestiune de Baze de Date”, Editura MATRIX-ROM, Bucuresti 2007.
14. Fotache Marin, Catalin Strimbei, Liviu Cretu – *“ORACLE 9i2, Ghidul dezvoltarii aplicatiilor profesionale”*, Editura Polirom, 2003.
15. Garcia-Molina Hector, Jeffrey D.Ullman, Jennifer Widom – *“Database Sistems: The complete book”*, Pearson Education International, 2002.

16. Gulutzan, P., T.Pelzer – *“SQL-99 Complete, Really, R&D Books”*, Lawrence, KA, 1999.
17. Gulutzan P., Peltzer T. – *“SQL Performance Tuning”*, Addison Wesley, 2002
18. Lorie, R.A., J.F. Nilsson – *“An Access Specification Language for a Relational Data Base System”*, IMB J. R&D, 23, 3, 1979.
19. Lorie, R.A., J.J. Daudenarde – *“SQL and its Applications”*, Englewoods Cliffs, N.J., Prentice_Hall,1991.
20. Melton, J., A.R.Simon – *“SQL-1999-Understanding Relational Components”*, San Francisco, Calif., Morgan-Kaufmann, 2002.

21. Melton, J., A.R.Simon – *“Understanding the New SQL: A Complete Guide”*, Morgan-Kaufmann, San Francisco, 1993.
22. Palinski, J.A. – *“Oracle SQL and PL/SQL Handbook”*, Addison-Wesley, 2003.
23. Perkins, J. şi Bryan Morgan – *“SQL fără profesor în 14 zile”*, traducere în limba română, Editura Teora, 1997.
24. Popescu, Ileana, *“Oracle8. – “Prelucrarea avansata a informatiei”*, Editura Tehnica, Bucuresti, 1999.
25. Riordan R. – *“Designing Relational Database Systems”*, Microsoft Press, 1999.

26. Rozenshtein, D.A., Abramovich and E.Birger – *“Optimizing Transact-SQL: Advanced Programming Techniques”*, Fremont, Calif., SQL Forum Press, 1995.
27. Sheldon R., Wilansky E. – *“MCSE Training Kit : Microsoft SQL Server 2000 Database Design and Implemention”*, Microsoft Press, 2001
28. Smith, J.M., P.Y.Chang – *“Optimizing the performance of a relational algebra database interface”*, Comm. ACM 18:10 (1975), pp.568-579.
29. Stonebraker M., J.M. Hellerstein (eds) – ”Reading in Databases Systems”, Morgan-Kaufmann, Sab Francisco, 1998.

30. Ullman J.D. – *“Principles of Database and Knowledge-Base Systems”*, Volume I, Computer Science Press, New York, 1988.
31. Ullman J.D. – *“Principles of Database and Knowledge-Base Systems”*, Volume II, Computer Science Press, New York, 1989.
32. Urman, S. – *“Oracle 9i PL/SQL Programming”, Osborne/McGraw-Hill, 2002*.
33. Welling, L., Laura Thomson – “Dezvoltarea de aplicații WEB cu PHP si MySQL”, traducere în limba română, Editura Teora, 2004.

CAPITOLUL I
ELEMENTE ALE MODELULUI RELAȚIONAL

Considerăm o mulțime nevidă, finită de simboluri, notată cu **U**. Numim elementele lui **U** atribute.

Fie **U** = $\{A_1, A_2, \ldots, A_n\}$. Fiecărui atribut A_i îi vom asocia o mulțime nevidă de valori notată dom(A_i), care va fi numită domeniul valorilor atributului A_i. Această mulțime dom(A_i) este numită şi mulțime valorilor posibile pentru A_i, $1 \leq i \leq n$.

Vom numi uplu peste **U** o aplicație, notată φ, φ : **U** $\rightarrow \bigcup_{1 \leq i \leq n} \mathrm{dom}(A_i)$, astfel încât $\varphi(A_i) \in \mathrm{dom}(A_i)$, $1 \leq i \leq n$.

Pentru fiecare uplu φ peste **U** putem să-i asociem mulțimea $\{A_1:V_1, A_2 : V_2, \ldots, A_n : V_n\}$, unde $V_i = \varphi(A_i)$, $1 \leq i \leq n$.

Invers, dacă este dată mulțimea $\{A_1:V_1, A_2 : V_2, \ldots, A_n : V_n\}$, cu $V_i \in dom(A_i)$, $1 \leq i \leq n$, atunci putem defini uplul φ peste **U** prin $\varphi(A_i) = V_i$, $1 \leq i \leq n$. Deci există o corespondență biunivocă între mulțimea de uple peste **U** şi mulțimea mulțimilor de forma considerată. Mai mult, dacă ordonăm mulțimea de atribute ale lui **U** sub forma: **U** $= (A_1, A_2, \ldots, A_n)$, atunci în locul mulțimii $\{A_1:V_1, A_2 : V_2, \ldots, A_n : V_n\}$ se poate considera numai vectorul $(V_1, V_2, \ldots,V_n)$.

În sisteme de gestiune de baze de date (FOX, DBASE, ORACLE, etc.) se lucrează cu astfel de ordonări. Vom privi uplele atât ca aplicații, cât şi ca vectori, considerând **U** ordonată.

(i) Relatia ca set.

O relație peste **U**, notată r, este o mulțime de uple peste **U**. În cazul în care mulțimea de uple este vidă, spunem că relația este vidă. În cazul în care mulțimea de uple este finită, spunem că relația este finită.

Rezultă că într-o astfel de relație nu putem avea două uple identice.

Ca operații de actualizare a unei relații putem avea adăugarea de noi uple, ştergerea unor uple, modificarea de uple.

În prelucrarea unei relații, mulțimea de uple variază în timp, ceea ce este constant în timp este structura sa, adică numele relației, împreună cu mulțimea de atribute. Să notăm această structură prin R(**U**) sau R[**U**], sau R(A_1, A_2, …, A_n) sau R[A_1, A_2, …, A_n], unde R este un simbol numit numele relației, iar **U** sau {A_1, A_2, …, A_n} este mulțimea de atribute corespunzătoare.

R(**U**) poartă denumirea şi de schemă de relație.

În cazul în care r este o relație finită peste **U** şi **U** se ordonează sub forma (A_1, A_2, …, A_n), atunci rezultă o reprezentare a lui r sub forma unui tablou:

	A_1	A_2	…	A_n
	V_1	V_{12}	…	V_{1n}
	1			
r	……….			
	V_{h1}	V_{h2}	…	V_{hn}

unde $(V_{i1}, V_{i2}, \ldots, V_{in})$ constituie un uplu din r, $1 \leq i \leq h$. Avem $V_{ij} \in dom(A_j)$, $1 \leq j \leq n$, $1 \leq i \leq h$.

Din acest motiv, în majoritatea sistemelor de gestiune de baze de date, structurile de date ce memorează aceste relații se numesc tabele, iar uplele se numesc linii.

O schemă de baze de date, notată D, este o mulțime finită de scheme de relație: $D = \{R_1[U_1], \ldots, R_p[U_p]\}$, unde $R_j[U_j]$ este o schemă de relație, $1 \le j \le p$.

O bază de date peste schema de baze de date D, este o aplicație ce asociază fiecărei scheme de relație $R_j[U_j]$ o relație r_j peste U_j, $1 \le j \le p$. Dacă D se consideră ordonată sub forma $(R_1[U_1], \ldots, R_p[U_p])$, atunci baza de date este un vector de relații, notat $\bar{r} = (r_1, r_2, .., r_p)$.

Structura unei relații se numeşte partea sa intensională, iar uplele relației poartă denumirea de partea sa extensională.

Operații referitoare la relații:

a) Proiecția unei relații.

Fie t un uplu peste R[U] şi X o submulțime a lui U. Proiecția lui t relativ la X, notată prin t[X] este restricția lui t (ca uplu) la submulțimea X. Aici t este considerat ca aplicație. Dacă X este mulțimea vidă, atunci t[X] il vom considera uplul vid. Dacă U este ordonată sub forma $(A_1,A_2,\ldots,A_n)$, iar în această ordonare $X=(A_{i1},A_{i2},\ldots,A_{ik})$, cu $i_1<i_2<\ldots<i_k$ şi $X=(V_1,V_2,\ldots,V_n)$, atunci $t[X] =(V_{i1},V_{i2},\ldots,V_{ik})$. Dacă r este o relație peste U, atunci proiecția lui r relativă la X, notată r[X] va fi definita prin: $r[X]=\{t[X] \ / \ t\in r\}$.

b)Reuniunea a două relații.

Fie r_1 şi r_2 relații peste R[U]. Atunci $r_1\cup r_2$ notează reuniunea celor două şi se defineşte ca:

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

c) Diferența a două relații.

Fie r_1 şi r_2 relații peste R[U]. Diferența lor, notată r_1-r_2 se defineşte astfel:

$r_1-r_2=\{t \,/\, t \text{ uplu}, t \in r_1 \text{ şi } t \notin r_2\}$

d) Unire (join) oarecare.

Fie relațiile r_i definite peste $R_i[U_i]$, i=1,2 $U_1 \cap U_2=\emptyset$.
Fie $A_{\alpha 1}$, $A_{\alpha 2}$,…, $A_{\alpha q} \in U_1$, $B_{\beta 1}$, $B_{\beta 2}$,…, $B_{\beta q} \in U_2$ şi Θ_i un operator de comparare între dom($A_{\alpha i}$) şi dom($B_{\beta i}$), $1 \le i \le q$. Sintactic Θ_i este unul din operatorii =, <, >, <=, >=, <>. Deci Θ_i este o relație binară pe dom($A_{\alpha i}$)×dom($B_{\beta i}$).

Rezultă că cele două domenii conțin valori comparabile prin Θ_i. Considerăm $\Theta=(A_{\alpha 1}\Theta_1 B_{\beta 1})\wedge\ldots\wedge (A_{\alpha q}\Theta_q B_{\beta q})$, unde semnul "$\wedge$" reprezintă conjuncția.

Join-ul oarecare între r_1 şi r_2 prin expresia Θ se notează prin $r_1 \overset{*}{\underset{\Theta}{-}} r_2$ şi se defineşte prin:

$r_1 \overset{*}{\underset{\Theta}{-}} r_2=\{t$ / t uplu peste $U_1\cup U_2$, $t[U_i]\in r_i$, $i=1,2$ şi $t[A_{\alpha j}]\Theta_j t[B_{\beta j}]$, $j=1,q\}$

Se pot defini expresii de tip Θ mai generale decit cea de sus, folosind parantezele şi operatorii de conjuncție şi disjuncție ($\wedge$, respectiv $\vee$) astfel:

- o expresie elementară este de forma $A\Theta B$, unde A şi B sunt atribute, iar Θ este operator de comparare.
- o expresie join se defineşte astfel:

1. Dacă e este o expresie elementară, atunci e şi (e) sunt expresii join.
2. Dacă e_1 şi e_2 sunt expresii join, atunci $e_1 \wedge e_2$, $e_1 \vee e_2$, $(e_1 \wedge e_2)$, $(e_1 \vee e_2)$ sunt expresii join.
3. Orice expresie join se obține numai prin regulile 1 şi 2.

Dacă Θ este o expresie join, atunci se defineşte faptul că (t_1,t_2) satisface Θ în mod recursiv, $t_1 \in r_1$, $t_2 \in r_2$.

1. (t_1,t_2) satisface $A\Theta B$ dacă $t_1[A]\Theta t_2[B]$.
2. (t_1,t_2) satisface $e_1 \wedge e_2$ şi $(e_1 \wedge e_2)$ dacă (t_1,t_2) satisface e_1 şi e_2.
3. (t_1,t_2) satisface $e_1 \vee e_2$ şi $(e_1 \vee e_2)$, dacă (t_1,t_2) satisface e_1 sau e_2.

În acest caz joinul oarecare se defineşte prin:
$r_1 \underset{\Theta}{\overset{*}{-}} r_2=\{(t_1,t_2)$ / (t_1,t_2) uplu peste $U_1 \cup U_2$, $t_i \in r_i$, $i=1,2$ şi (t_1,t_2) satisface $\Theta\}$

e) Produsul cartezian.
Fie r_i relații definite peste $R_i[U_i]$, $i=1,2$ şi $U_1 \cap U_2=\varnothing$.
Produsul cartezian al relațiilor r_1 şi r_2 se noteaza prin $r_1 \times r_2$ şi se defineşte prin: $r_1 \times r_2=\{t$ / t uplu peste $U_1 \cup U_2$, $t[U_i] \in r_i$, $i=1,2\}$

Obs. Produsul cartezian este un join oarecare cu expresia join Θ=TRUE.

f) Unirea naturală (join natural)

Fie r_i relații peste $R_i[U_i]$, i=1,2. Se numeşte join natural sau unire a celor două relații, notat r_1*r_2 o relație peste $U_1 \cup U_2$ definită prin:

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

Fie R un nume pentru relația peste $U_1 \cup U_2$. Deci relația r_1*r_2 va fi considerată peste schema $R[U_1 \cup U_2]$. Considerăm un exemplu pentru operația de join natural. Fie $R_1[ABCD]$, $R_2[CDE]$ şi relațiile r_1 şi r_2 date astfel:

	A	B	C	D	
r_1	1	0	0	0	t_1
	1	1	0	0	t_2
	0	1	0	1	t_3
	0	0	0	1	t_4
	1	1	1	1	t_5

	C	D	E	
r_2	0	0	0	V_1
	1	1	1	V_2
	1	1	0	V_3
	1	0	0	V_4
	1	0	1	V_5

Joinul natural va fi:

$r_1*r_2=$

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

Se observă că uplele t_3 şi t_4 din r_1 nu contribuie la construcția joinului, deoarece nu există uple $V_j \in r_2$, astfel încît $t_3[CD]=V_j[CD]$, şi nu există uple $V_j \in r_2$, astfel încît $t_4[CD]=V_j[CD]$. La fel uplele V_4 şi V_5 din r_2 nu contribuie la formarea joinului, deoarece $V_4[CD]=V_5[CD] \notin r_1[CD]$.
Să remarcăm faptul că perechea (t_i,V_j), $t_i \in r_1$, $V_j \in r_2$ contribuie la formarea joinului, dacă $t_i[CD]=V_j[CD]$.
În acest caz, va rezulta un uplu în joinul natural, notat w, definit prin: $w[U_1]=t_i[U_1]$, $w[U_2-U_1]=V_j[U_2-U_1]$.

Se observă uşor următoarele relații:

1. $(r_1*r_2)[U_1] \subseteq r_1$, $(r_1*r_2)[U_2] \subseteq r_2$.
2. Dacă $r_1'=\{t_1 \ / \ t_1 \in r_1, \exists t_2 \in r_2, \text{ a.î., } t_1[U_1 \cap U_2]=t_2[U_1 \cap U_2]\}$
   şi $r_2'=\{t_2 \ / \ t_2 \in r_2, \exists t_1 \in r_1, \text{ a.î., } t_1[U_1 \cap U_2]=t_2[U_1 \cap U_2]\}$

şi $r_1'' = r_1 - r_1'$, $r_2'' = r_2 - r_2'$, atunci

$$r_1 * r_2 = r_1' * r_2',\ (r_1 * r_2)[U_1] = r_1',\ (r_1 * r_2)[U_2] = r_2'.$$

3. Dacă $\overline{r_1} \subseteq r_1$ şi $\overline{r_2} \subseteq r_2$ satisfac relația $\overline{r_1} * \overline{r_2} = r_1 * r_2$, atunci $r_1' \subseteq \overline{r_1}$ şi $r_1' \subseteq \overline{r_2}$, adică relațiile r_1', r_2' sunt minimale în r_1 respectiv r_2 cu proprietatea $r_1' * r_2' = r_1 * r_2$. În cazul în care $U_1 \cap U_2 = \emptyset$, atunci este clar că joinul natural este tocmai produsul cartezian $r_1 \times r_2$.

Operația de join natural se poate extinde la mai multe relații.

Fie r_i peste $R_i[U_i]$, i=1,h. Joinul natural al acestora se notează prin

$r_1 * r_2 * \ldots * r_h$ sau $*<r_i, i=1,h>$, sau $* <r_i, i=1,h>$ şi se definește prin:

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

Este clar că operația de join natural nu depinde de ordinea considerării relațiilor r_i, $i=1,h$.

Dacă notăm cu $*_2$ operația de join natural între două relații şi cu $*_h$ operația de join natural între h relații, apare următoarea problemă:

Problema 1. Operația $*_h$ se poate exprima cu ajutorul operației $*_2$?

Răspunsul la problema 1 este NU. Demonstrați.

g) Selecția.

Fie r o relație peste R[U].

Definim întâi expresia elementară de selecție prin:

> $A\Theta B$ sau $A\Theta c$ sau $c\Theta B$, unde $A,B\in U$, iar c este o constantă, comparabilă cu elementele din dom(A) şi dom(B).

Dacă e_1 şi e_2 sunt expresii elementare de selecție, atunci (e_1), $e_1\wedge e_2$, $e_1\vee e_2$, $(e_1\wedge e_2)$, $(e_1\vee e_2)$ este o expresie de selecție. Orice expresie de selecție se obține numai prin regulile de mai sus. Fie E o expresie de selecție.

Precizăm cazul în care uplul t satisface E.

Dacă $E\equiv A\Theta B$, atunci t satisface E, dacă $t[A]\Theta t[B]$.

Dacă $E\equiv A\Theta c$, atunci t satisface E, dacă $t[A]\Theta c$.

Dacă $E\equiv c\Theta B$, atunci t satisface E, dacă t satisface $c\Theta t[B]$.

Dacă $E \equiv e_1 \wedge e_2$, atunci t satisface E, dacă t satisface e_1 şi e_2.
Dacă $E \equiv e_1 \vee e_2$, atunci t satisface E, dacă t satisface e_1 sau t satisface e_2.
Fie F o expresie de selecție.
Selecția se notează prin $\sigma_F(\underline{r})$ şi se defineşte prin:
$\sigma_F(\underline{r})=\{t$ / t uplu peste R[U], t satisface F$\}$.

h) Intersecția relațiilor.

Fie r_i relații peste $R_i[U_i]$, $i=1,2$ şi $U_1=U_2$.
Atunci intersecția se defineşte prin:

$r_1 \cap r_2=\{t$ / t uplu peste U_1, a.î. $t \in r_1$ şi $t \in r_2\}$

Se arată că o parte din operațiile cu relații se exprimă cu ajutorul altor operații. Mulțimea de operații: reuniune, diferență, produs cartezian, proiecție şi selecție se arată a fi minimală cu proprietatea că celelalte operații se exprimă cu ajutorul acestora.

(ii) Relatia ca multiset

Fie schema R(U). O relatie ca multiset peste R(U) este un multiset de uple peste U.

$r=\{u_1:m_1,\ldots, u_h:m_h\}$.

Orice relatie de tip set este multiset. Operatii:

a) Proiectia. $r=\{u_1:m_1,\ldots, u_h:m_h\}$ peste R(U) si X <=U, r[X]

   $v_i=u_i[X]$, consideram $v_1:m_1,\ldots, v_h:m_h$, v_i pot fie egale. Se grupeaza toate egale si se aduna multiplicitatile pentru cele egale.

A	B	C	D	m_i
0	0	0	0	3
0	0	1	0	2
0	0	0	1	4
1	1	0	1	2
1	1	1	2	1

X=AB,

$v_1[X]=(0 \quad 0) \; : \; 3$

$v_2[X]=(0 \quad 0) \; : \; 2$

$v_3[X]=(0 \quad 0) \; : \; 4$ $\qquad r[X]=\{(0 \quad 0):9, (1 \quad 1):3\}$.

$v_4[X]=(1 \quad 1) \; : \; 2$
$v_5[X]=(1 \quad 1) \; : \; 1$
----

b) Reuniunea a doua relatii:

$r_1=\{u_1:m_1, \ldots, u_h:m_h\}$, $r_2=\{v_1:n_1,\ldots, v_p:n_p\}$, peste R(U).

$r_1 \cup r_2 =\{w_1 : q_1,\ldots, w_l: q_l \}$, $u_1,\ldots, u_h$, $v_1,\ldots, v_p$,

For i=1 to h

1) If exista v_j astfel incat $u_i= v_j$, atunci $w_\alpha= u_i$, $q_\alpha=m_i+n_j$,
2) altfel $w_\alpha= u_i$, $q_\alpha=m_i$

   endfor

For j=1 to p

3) If $v_j \neq u_i$, for all i=1,h, atunci $w_\alpha = v_j$, $q_\alpha = n_j$

Endfor

Exemplu: r_1 = {(0 0):2,(0 1):3, (1 0):2},

r_2 = {(0 0):1,(1 1):3},

$r_1 \cup r_2$ ={(0 0):3, (0 1):3, (1 0):2, (1 1):3}

c) Diferenta a 2 relatii multiset.

$r_1=\{u_1:m_1, \ldots, u_h:m_h\}$, $r_2=\{v_1:n_1,\ldots, v_p:n_p\}$, peste R(U).

$r_1 - r_2 = \{w_1 : q_1,\ldots, w_l: q_l\}$,

For i=1 to h

4) If exista v_j astfel incat $u_i= v_j$, si $m_i > n_j$, atunci $w_\alpha= u_i$, $q_\alpha=m_i-n_j$, altfel

5) If exista v_j astfel incat $u_i = v_j$, si $m_i <= n_j$, atunci continue,

6) altfel $w_\alpha = u_i$, $q_\alpha = m_i$

endfor

d) Joinul oarecare.

$r_1=\{u_1:m_1, \ldots, u_h:m_h\}$, peste $R_1(U_1)$, $r_2=\{v_1:n_1,\ldots, v_p:n_p\}$, peste $R_2(U_2)$, $U_1 \cap U_2 =\Phi$,

$r_1 \theta r_2= \{w : q \mid w$ peste $U_1 \cup U_2$, exista i si j, $w[U_1]= u_i$, $w[U_2]= v_j$, w satisface $\theta\}$.

e) Produsul cartezian.

$r_1=\{u_1:m_1, \ldots, u_h:m_h\}$, peste $R_1(U_1)$, $r_2=\{v_1:n_1,\ldots, v_p:n_p\}$, peste $R_2(U_2)$, $U_1 \cap U_2 =\Phi$,

$r_1 \times r_2 = \{w : q \mid w$ peste $U_1 \cup U_2$, exista i si j, $w[U_1] = u_i$, $w[U_2] = v_j\}$.

f) Join natural.

(C1) $u_i [U_1 \cap U_2] = v_j[U_1 \cap U_2]$,

(C2) $w[U_1] = u_i [U_1]$, $w[U_2 - U_1] = v_j[U_2 - U_1]$, $q = m_i * n_j$.

```
r= Φ
For i=1,h
For j=1,p
 If perechea (u_i, v_j) satisface (C1), then
 calculeaza  w:q  conform  lui (C2) si adauga w:q la r,
endif
endfor
```

endfor

Exemplu:

A	B	C	D	m_i	
0	1	0	0	3	u_1
0	0	1	0	2	u_2
0	1	1	1	2	u_3

C	D	E	n_j	
0	0	1	2	v_1
1	0	0	3	v_2
0	1	1	1	v_3
0	0	0	3	v_4

	A	B	C	D	E	q_l
(u_1,v_1)->	0	1	0	0	1	6
$(u_1,v4)$->	0	1	0	0	0	9
(u_2,v_2)->	0	0	1	0	0	6

Sunt valabile proprietatile joinului natural pentru cazul set?