

Fiola de SQL (15)

NULii se întorc

– Marin Fotache

Această parte, a doua după socotelile mele (prima a fost publicată în PC Report nr. 93/2000), era programată pentru luna decembrie a anului trecut. Plasată imediat după alegeri, ar fi avut o conotație politică dificil de mascat. Chiar dacă și acum aluzia politică poate fi anatemitată de bănuitori, apariția a două fiole SQL dedicate NULL-ităților în două legislaturi diferite îmi oferă o confortabilă senzație de echidistanță.

Însă, dincolo de insinuările răutăcioase de care sunt în stare, nu fac decât să-mi respect una din numeroasele promisiuni presărate de-a lungul și de-a curmezișul tratamentului cu fiole de SQL la care supun câțiva dintre numeroșii cititori ai revistei.

După cum reaminteam și în fiola din luna iunie a anului trecut, valoarea (sau metavalorarea) NULL a fost introdusă în modelul relațional pentru a pune în evidență ceva necunoscut, nespecificat sau ne-aplicabil. Raportul Interim 75-02-09 înaintat ANSI X3 (SPARC Study Group) în 1975, delimita 14 tipuri de date incomplete ce ar putea fi valori de atribute sau rezultate ale unor operații precum: depășiri ale capacității de stocare, diviziune la zero, trunchierea șirurilor de caractere, alte erori computaționale, valori necunoscute etc.

Prima chestiune discutată cu privire la valorile NULL a fost legată de modul lor de propagare în expresii numerice și necesitatea folosirii funcțiilor VALUE, COALESCE sau NVL. O altă problemă ține de faptul că o (meta)valoare NULL nu poate fi comparată cu alt NULL utilizând operatorii >, <, >=, <=, =, <>, deoarece NULL are la bază o logică trivalentă. Logica operatorilor NOT, AND și OR, în condițiile apariției valorilor necunoscute, este ilustrată în *tabelul 1*.

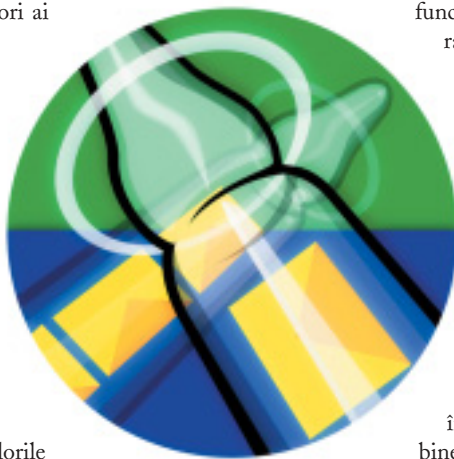
Valoarea NECUNOSCUȚ rezultă prin utilizarea lui NULL în comparații/predicate. NECUNOSCUȚ este însă o valoare logică cu regim diferit de

NULL, deoarece NULL reprezintă valoarea unor date (attribute). Diferența este destul de subtilă. În orice caz, acesta este și motivul pentru care în SQL se formulează (atribut IS [NOT] NULL), și nu (atribut = NULL).

Astăzi ne vom concentra pe o altă discuție cu importante implicații: subconsultări în care pot fi extrase (și) valori nule. Tabela utilizată va fi (tot) PERSONAL2 al cărei conținut este prezentat în figura 1.

Problema și explicații

Să începem cu o întrebare banală: Care sunt angajații fără funcții de conducere, altfel spus, cei de la baza piramidei organizaționale?



```
SELECT *
FROM PERSONAL2
WHERE marca NOT IN
  (SELECT marcasef
   FROM PERSONAL2)
```

Rezultatul interogării ar trebui să fie alcătuit din șase linii, corespunzătoare angajaților 4, 6, 7, 8, 9 și 10. Aceasta deoarece subconsultarea extrage mărcile angajaților care au măcar un subordonat, iar predicatul de legătură între cele două nivele de consultare este NOT IN. Ei bine, rezultatul acestei consultări este de-a dreptul vid!

Explicația pentru aceste situații este foarte bine formulată de Joe Celko. Astfel, fraza SELECT de mai sus este echivalentă cu:

```
SELECT *
FROM PERSONAL2
WHERE NOT (marca IN (SELECT marcasef FROM PERSONAL2))
```

Logica operatorului IN și conținutul tabelului PERSONAL2 determină ca predicatul interogării să se prezinte astfel:

```
SELECT *
FROM PERSONAL2
WHERE NOT ( (marca = NULL) OR (marca = 1) OR (marca = 2) OR
  (marca = 5) OR
  (marca = 3) )
```

În continuare aplicăm legile lui DeMorgan și obținem forma:

```
SELECT *
FROM PERSONAL2
WHERE (NECUNOSCUȚ) AND (marca <> 1) AND (marca <> 2)
AND (marca <> 5) AND (marca <> 3)
```

Deoarece primul termen al predicatului este NECUNOSCUȚ, interogarea nu va extrage nici o linie din tabela PERSONAL2.

Figura 1. Tabela PERSONAL2

Marca	Numepren	Datanast	Compart	Marcasef	Saltarifar
1	ANGAJAT 1	01/07/1962	DIRECTIUNE	.NULL.	6000000
2	ANGAJAT 2	11/10/1977	FINANCIAR	1	4500000
3	ANGAJAT 3	22/10/1962	MARKETING	1	4500000
4	ANGAJAT 4	.NULL.	FINANCIAR	2	3800000
5	ANGAJAT 5	30/04/1965	FINANCIAR	2	4200000
6	ANGAJAT 6	09/11/1965	FINANCIAR	5	3500000
7	ANGAJAT 7	.NULL.	FINANCIAR	5	2800000
8	ANGAJAT 8	31/12/1960	MARKETING	3	2900000
9	ANGAJAT 9	28/02/1976	MARKETING	3	4100000
10	ANGAJAT 10	29/01/1972	RESURSE UMANE	1	5500000

Soluții

În vederea eliminării problemelor determinate de valorile NULL și conversiei rezultatelor logicii trivalente (ADEVĂRAT/FALS/NECUNOSCUȚ) pe calapodul logicii obișnuite, bivalente (ADEVĂRAT/FALS), standardul SQL-92 este înzestrat cu un predicat de forma:

```
<predicat> IS [NOT] TRUE | FALSE | UNKNOWN
```

Aplicabilitatea sa este însă limitată, iar multe SGBD-uri încă n-au implementat această opțiune. Prin urmare, cel mai sigur mod de a scăpa de griji este folosirea funcțiilor de conversie a valorilor nule: COALESCE, VALUE, NVL.

Astfel, pentru a răspunde la problema formulată (*care sunt angajații fără nici un subordonat ?*), o soluție comună Oracle/VFP este:

```
SELECT *
FROM PERSONAL2
WHERE NVL(marca,0) NOT IN
      (SELECT NVL(marcasef,0)
       FROM PERSONAL2)
```

Funcția NVL va converti toate eventualele valori nule ale atributelor marca și marcasef în valori 0. Eludăm, astfel, logica trivalentă și obținem rezultatul corect.

A doua problemă

Care sunt angajații cu funcții de conducere (cei care au măcar un subordonat)?

Interesant este că soluția:

```
SELECT *
FROM PERSONAL2
WHERE marca IN
      (SELECT marcasef
       FROM PERSONAL2)
```

Tabel 1. Rezultatele expresiilor în care apar operatorii NOT, AND și OR și valori necunoscute

Aplicarea operatorului NOT unei condiții			
NOT	ADEVĂRAT	FALS	NECUNOSCUȚ
	FALS	ADEVĂRAT	NECUNOSCUȚ
Combinarea a două expresii utilizând operatorul AND			
AND	ADEVĂRAT	FALS	NECUNOSCUȚ
ADEVĂRAT	ADEVĂRAT	FALS	NECUNOSCUȚ
FALS	FALS	FALS	FALS
NECUNOSCUȚ	NECUNOSCUȚ	FALS	NECUNOSCUȚ
Combinarea a două expresii utilizând operatorul OR			
OR	ADEVĂRAT	FALS	NECUNOSCUȚ
ADEVĂRAT	ADEVĂRAT	ADEVĂRAT	ADEVĂRAT
FALS	ADEVĂRAT	FALS	NECUNOSCUȚ
NECUNOSCUȚ	ADEVĂRAT	NECUNOSCUȚ	NECUNOSCUȚ

funcționează ! Prin analogie cu prima problemă, explicităm logica predicatului din clauza WHERE:

```
SELECT *
FROM PERSONAL2
WHERE ( (marca = NULL) OR (marca = 1) OR
```

```
(marca = 2) OR (marca = 5) OR (marca = 3) )
```

Rezultatul expresiei marca = NULL este NECUNOSCUȚ, iar, așa cum am văzut în tabelul 1, rezultatul expresiei logice NECUNOSCUȚ OR ADEVĂRAT este ADEVĂRAT.

Prin urmare, nu orice subconsultare ce extrage valori nule generează probleme, ci, în principiu, numai cele conectate prin NOT IN.

Operatorii ALL/ANY și valorile nule

Dacă dorim să aflăm răspunsul la întrebarea *Care sunt angajații mai tineri decât cel puțin unul dintre cei angajați la compartimentul Financiar*, soluția următoare bazată pe operatorul ANY este funcțională.

```
SELECT *
FROM PERSONAL2
WHERE DataNast > ANY
      (SELECT DataNast
       FROM PERSONAL2
       WHERE compart = 'FINANCIAR')
```

În schimb, dacă intenția este de a afla *angajații mai maturi (mai în vârstă) decât toți cei angajați la compartimentul Financiar*, interogarea următoare nu extrage nici o linie.

```
SELECT *
FROM PERSONAL2
WHERE DataNast < ALL
      (SELECT DataNast
       FROM PERSONAL2
       WHERE compart = 'FINANCIAR')
```

La mijloc sunt cele două valori nule pentru angajații cărora nu li se cunoaște data nașterii. Necazul nu are neapărat legătură cu SQL-ul. Ce dată să luăm în calcul, implicit, pentru valorile nule ale celor doi angajați ? Apelând la o improvizație, și ținând seama de specificul problemei, am putea înlocui valorile nule ale DataNast cu data curentă a sistemului. Iată varianta Oracle:

```
SELECT *
FROM PERSONAL2
WHERE NVL(DataNast,SYSDATE) < ALL
      (SELECT NVL(DataNast, SYSDATE)
       FROM PERSONAL2
       WHERE compart = 'FINANCIAR')
```

Toate bune și vacanță plăcută !

Moment publicitar

Beneficiind de degetul întins de redactorii NetReport, îmi face plăcere să vă informez, absolut dezinteresat, că la editura Polirom din Iași a apărut o carte despre SQL, dedicată atât standardului SQL-92, cât și găselnițelor din DB2, Oracle și Visual FoxPro. Cartea nu-i nici prea ieftină, nici prea ușoară, dar strașnică (vă spune ceva cuvântul *paranoia*?)...

Marin Fotache este conferențiar la Catedra de Informatică Economică, UAIC Iași, Facultatea de Economie și Administrarea Afacerilor. Poate fi contactat pe email la: fotache@uaic.ro. ■ 76

Bibliografie

- ✓ Celko, J. - SQL for Smarties: Advanced SQL Programming, Morgan Kaufmann, San Francisco, 2000