

Despre NULLi, dar numai în SQL

Discuția de astăzi este rezervată valorilor nule. Nu, nu e vorba, cum v-ați fi așteptat, de politicieni sau guvernanți, ci pur și simplu de valoarea relațională NULL. De fapt, un soi de ...*metavaloare*, deși am vaga senzație că nu am lămurit nimic cu acest termen.

În limbajul curent, când spunem despre cineva că este nul (și nulă (așa, pentru *sexual correctness*)), înseamnă că e un/o zero, un/o cvasi-inexistent(ă). Sensul relațional este mult mai aproape de această percepție, decât de, spre exemplu, semnificația fotbalistică – *meci nul* deseori se folosește și pentru scoruri precum 3-3, deși numai nulă nu este o asemenea partidă.

Valoarea (sau metavaloarea) NULL a fost introdusă în modelul relațional pentru a pune în evidență ceva necunoscut, nespecificat sau ne-aplicabil. De exemplu, într-o tabelă CLIEȚI, trebuie inserată o înregistrare aferentă unui client căruia îi cunoaștem codul fiscal, denumirea, localitatea în care-și are sediul, dar nu și adresa (strada, nr. ...). Din punctul de vedere al teoriei relaționale, cel mai sănătos ar fi să îi atribuim câmpului Adresa aferent acestei linii valoarea NULL.

Una dintre “poruncile” lui Codd pe care ar trebui să le urmeze un SGBD pentru a fi “uns” ca relațional se referă la prelucrarea sistematică a valorilor nule. La casele mari (Oracle, IBM, Informix etc.), lucrurile sunt puse la punct de mai multă vreme, în timp ce la SGBD-urile din categoria celor pentru PC-uri (cum este FoxPro), doar de câțiva ani.

NULL nu este echivalent cu zero/spațiu

Valorile NULLe pun în evidență locul în care informațiile lipsesc, pentru a nu se confunda cu valorile respective care sunt chiar spații sau zero (nici un atribut component al cheii primare unei tabele nu poate avea valori nule).

În baza de date prezentată în numărul trecut, tabelii PERSONAL îi este adăugată tabela SPORURI. Pentru VFP comanda este:

```
CREATE TABLE sporuri ( ;  
    an INTEGER CHECK BETWEEN(an,2000,2005) ERROR 'An incorect' ;  
    luna INTEGER CHECK BETWEEN(luna,1,12) ERROR 'Luna incorecta' ;  
    marca INTEGER ;  
    sporvechime INTEGER NULL ;  
    spornoapte INTEGER NULL ;  
    sporcd INTEGER NULL ;  
    altespor INTEGER NULL ;  
    PRIMARY KEY STR(an,4)+STR(luna,2)+STR(marca,5) TAG primaru ;  
    FOREIGN KEY marca TAG marca REFERENCES personal TAG primaru)
```

Fără a mai prezenta formatul comenzii CREATE TABLE în Oracle, trebuie amintit faptul că, la creare, în VFP se specifică attributele ce *pot* avea valori nule, în timp ce în Oracle trebuie declarate explicit cele care *nu pot* avea asemenea valori.

După cum se observă, obiectivul tabelii SPORURI este de a evidenția sporurile lunare primite de fiecare angajat. Sporcd se referă la sporurile pentru condiții deosebite, iar altespor la sporurile diverse. Cu ajutorul valorii NULL se poate face diferența între persoanele care au un anume spor zero (de ex. directorul general nu prea primește spor de noapte (poate că am avea, totuși, nevoie de un alt câmp - spor_de_club_de_noapte...))

și cele pentru care nu fost operate încă sporurile pentru luna respectivă. Altfel spus, angajații pentru care nu s-a introdus în bază valoarea sporului pe luna curentă vor avea în câmpul corespunzător valoarea NULL, în timp ce pentru cei care nu au dreptul la un asemenea spor atributul va avea valoarea 0.

Spre deosebire de Oracle, până la versiunea 3 (VFP), FoxPro nu “suporta” valorile NULL. Ca alternativă, se folosea funcția EMPTY(sporcd) sau, pur și simplu, sporcd=0. Începând cu această versiune, orice atribut pentru care sunt permise valori nule (vezi și clauza NULL din comanda CREATE TABLE) le poate “primi”, fie prin combinația de taste Ctrl/0 într-o fereastră BROWSE, fie printr-o comandă INSERT INTO, REPLACE sau UPDATE.

Utilizand NULL, se pot afla **angajații pentru care nu s-au operat sporul pentru condiții deosebite pe luna curentă** (iunie 2000) :

```
SELECT *
FROM sporuri
WHERE an=2000 AND luna=6 AND sporcd IS NULL
```

Angajații pentru care s-a introdus acest spor, dar a cărui valoare este zero (angajații despre care se știe sigur că nu primesc spor de condiții deosebite) pot fi obținuți prin

```
SELECT *
FROM sporuri
WHERE an=2000 AND luna=6 AND sporcd = 0
```

Este important de notat că pentru aflarea valorilor nule, operatorul este IS NULL și nu =NULL. Rezultatul interogării

```
SELECT *
FROM sporuri
WHERE an=2000 AND luna=6 AND sporcd = NULL
```

Este o tabelă cu 0 (zero) linii. În general, rezultatul evaluării sporcd = NULL va fi întotdeauna FALSE.

Valorile NULL ajută la aflarea răspunsului corect la întregări de genul: **care este media sporului de condiții deosebite pentru luna iunie 2000 ?**

```
SELECT AVG(sporcd)
FROM sporuri
WHERE an=2000 AND luna=6
```

Probleme privind NULL

O primă observație privind valorile NULL este că operează după o logică tri-valentă. De exemplu dacă facem **reuniunea dintre mulțimea angajaților cu sporcd sub 2 000 000 cu mulțimea celor ce au sporcd peste 2 000 000** nu obținem tabela sporuri, cu numai liniile acesteia pentru care sporcd IS NOT NULL.

```
SELECT *
FROM sporuri
WHERE an=2000 AND luna=5 AND sporcd >= 2000000
UNION
SELECT *
FROM sporuri
WHERE an=2000 AND luna=6 AND sporcd < 2000000
```

Prin urmare, valorile nule sunt imposibil de încadrat, nefiind nici mai mari, dar nici mai mici de 2 000 000.

O altă problemă a valorilor NULL ale unui atribut este că antrenează nulitatea rezultatului unei expresii în care apare. Excepție fac funcțiile statistice precum AVG. Răspunsul la întrebarea: **Care este valoarea totală a sporurilor fiecărui angajat pentru luna iunie 2000 ?** ar fi obținut prin fraza SQL.

```
SELECT *, (sporvechime+spornoapte+sporcd+altespor) AS totalsporuri
FROM sporuri
WHERE an=2000 AND luna=1
```

Pentru liniile extrase din tabela SPORURI (cele care satisfac condiția din clauza WHERE) în care măcar unui din attributele sporvechime, spornoapte, sporcd, altespor este NULL, totalsporuri va fi, implicit, nulă.

Există diverse situații în care NULL se dorește a fi convertit în spațiu sau zerouri, în funcție de tipul atributului. Situațiile la care mă refer sunt cele de joncțiune externă (OUTER JOIN) tratate într-o fiolă viitoare. Pentru a nu aștepta atât, **ne propunem să obținem totalul sporurilor fiecărui angajat, substituind în calcule, fiecărei valorilor NULL valoarea 0.** În acest scop se folosește funcția NVL.

```
SELECT *, NVL(sporvechime,0)+NVL(spornoapte,0)+NVL(sporcd,0)+NVL(altespor,0)
      AS totalsporuri
FROM sporuri ;
WHERE an=2000 AND luna=6
```

Reținem, însă: funcția *NVL* nu se aplică expresiei, ci fiecărui operand susceptibil de nulitate.

Concluzii

Pentru reprezentarea informațiilor necunoscute sau ne-aplicabile, modelul relațional dispune de (meta)valoarea NULL, aplicabilă identic tuturor tipurilor de attribute (numeric, șir de caractere, dată calendaristică etc.). Operatorul utilizat în SQL este IS NULL iar conversia valorilor nule în alte valori presupune utilizarea funcției NVL. Pentru numerele viitoare rămâne de discutat despre NULL în joncțiunile externe și mai ales problemele subconsultărilor (operatorii IN și EXISTS) în condițiile prezenței unor valori nule.

Ar mai fi meritat măcar câteva rânduri și dezbateră în jurul logicii tri-valente specifice valorilor NULL, precum și păreri pro și contra privind înlocuirea sa în modelul relațional printr-o logică bivalentă (C.J. Date, E.F.Codd, D. McGoveran, Tom Johnson etc.). Dar credeți că aceasta e singura datorie pe care o am de achitat către dvs...

Marin Fotache