

## Orizontalizarea verticalității. Caz practic nr.1

Cer scuze limbii române pentru așa un afront, dar încă nu mi-am revenit de pe urma “verticalizării jocului” de la Euro 2000.

Luna trecută am prezentat o situație în care era nimerită folosirea joncțiunii externe. Pornind de la cele două tabele, PERSONAL{Marca, NumePren, DataNast, Compart, MarcaSef, SalTarifar} și SPORURI {An, Luna, Marca, SporVechime, SporNoapte, SporCD, AlteSpor}, ne interesau (printre altele) sporurile personalului pe primele trei luni ale anului (chiar și pentru cei angajați pe parcursul acelei perioade sau după). Încercăm acum să generalizăm problema la nivelul întregului an.

**Problemă. Care a fost cuantumul sporului de noapte acordat fiecărui angajat în fiecare lună a anului 2000 ?**

Diferența supărătoare față de problema anterioară ține de faptul că trebuie luate în considerare toate lunile din anul 2000 pentru care există înregistrări ? Luna de început este, de obicei, ianuarie, dar dacă implementarea BD s-a făcut în altă lună, aceasta va reprezenta limita (din stânga) a intervalului care interesează. Luna de sfârșit este ultima pentru care s-au introdus datele legate de sporuri (în cazul nostru august).

Cu alte cuvinte, pe baza unei tabeli de forma:

An	Luna	Marca	...	SporNoapte	...
2000	1	1		150 000	
2000	1	2		235 000	
2000	1	3		0	
2000	2	1		0	
2000	2	3		189 000	
2000	2	4		0	
...	...	...		...	

dorim să obținem o situație de genul:

Marca	NumePren	SalTarifar	SporNoapte1	SporNoapte2	...	SporNoapte8
1	ANGAJAT 1	4550000	150000	0		257450
2	ANGAJAT 2	3589000	235000	189000		0
...	...	...	...	....		...

De aici și mult-prea-generosul titlu al materialului de față. Ultima coloană este SporNoapte8 deoarece ultima lună pentru care au fost introduse datele este, așa cum am convenit, august.

### ***Tentativa nr.1 (nereușită) VFP***

Mergând pe “mâna” soluției de luna trecută am fi tentați să scriem o frază SELECT de genul:

```

SELECT p.marca, numepren, saltarif, ;
      NVL(s1.spornoapte,0) as SporurNoapte1, ;
...
      NVL(s12.spornoapte,0) as SporurNoapte12 ;
FROM personal2 p ;
      LEFT OUTER JOIN sporuri s1 ON p.Marca=s1.Marca AND s1.An=2000 AND s1.Luna=1 ;
...
      LEFT OUTER JOIN sporuri s12 ON p.Marca=s12.Marca AND s12.An=2000 AND s12.Luna=12

```

Varianta prezentată, dincolo de faptul că e cât se poate de primitivă, necesită și un cuvânt de mulțumire cerului că anul are numai 12 luni (în loc de vreo 50).

Spre lauda sa, VFP nu execută o asemenea nenorocire de interogare, e adevărat, dintr-un alt motiv: Too many subqueries.

### ***O soluție VFP***

Varianta pe care o discutăm (prezentată în [listing 1](#)) este una curat... procedurală. PRGul următor se bazează pe una din cele mai grozave găselnițe ale FoxPro – macrosubstituția .

#### **Listing 1. Soluția VFP**

```

SELECT DISTINCT An, Luna ;
FROM sporuri ;
INTO ARRAY vLuni ;
WHERE An = 2000

SELECT Marca, NumePren AS NumePrenume, SalTarifar ;
FROM personal2 p ;
INTO CURSOR cursor0

nPrecedent = 0
cPrecedent = ALLTRIM(STR(nPrecedent,2))

FOR i = 1 TO ALEN(vLuni) / 2
    cAn = ALLT(STR(vLuni(i,1),3))
    cLuna = ALLT(STR(vLuni(i,2),2))
    SELECT cursor&cPrecedent..*, NVL(s.spornoapte,0) AS SporNoapte&cLuna ;
    FROM cursor&cPrecedent LEFT OUTER JOIN sporuri s ;
        ON cursor&cPrecedent..Marca = s.Marca AND s.An=vLuni(i,1) AND s.Luna=vLuni(i,2) ;
        INTO CURSOR cursor&cLuna
    nPrecedent = vLuni(i,2)
    cPrecedent = ALLTRIM(STR(nPrecedent,2))
ENDFOR

BROWSE
RETURN

```

Masivul vLuni creat prin primul **SELECT** are două coloane în care sunt stocate anul și luna pentru care există sporuri introduse, servind drept element esențial în crearea unor cursoare intermediare (câte unul pentru fiecare lună). Apoi am creat primul cursor, cursor0, pentru a pregăti structura ciclică de creare a cursoarelor intermediare.

Secvența FOR... NEXT se repetă în funcție de numărul de luni conținut în masiv. la fiecare execuție se crează un cursor nou prin joncțiunea externă a cursorului precedent cu tabela SPORURI pentru luna preluată din linia curentă a masivului.

Frumos, nu ?

### ***Soluția Oracle:***

În Oracle 8 ne izbim frontal de o altă problemă. Printr-un singură frază SQL nu o scoatem la capăt, iar structuri de program (alternative, repetitive) înseamnă PL/SQL. În PL/SQL, crearea “din mers” a tabelelor derivate (view-urilor) prin secvențe ciclice presupune Dynamic SQL. Ori, n-am ajuns cu materia (încă) la asemenea minunății. Ambulanța (de fapt, salvarea) vine de la un mod de lucru cu script-urile care pe mine m-a entuziasmat prin simplitate, așa că mă grăbesc să vi-l prezint în [listing 2](#).

## Listing 2. Soluția Oracle 8

```
DROP VIEW luni ;
DROP VIEW luni2 ;
DROP VIEW interog ;
DROP VIEW inter0 ;
DROP VIEW inter1 ;
...se continua DROPurile până la inter12
DROP VIEW selectu ;

CREATE VIEW luni as
    SELECT DISTINCT luna FROM sporuri WHERE An=2000;

CREATE VIEW luni2 as
    SELECT 0 as luna1, MIN(luna) as luna2
    FROM luni
    UNION
    SELECT I1.luna, I2.luna
    FROM luni I1, luni I2
    WHERE I1.luna < I2.luna AND (I1.luna, I2.luna) IN
        (SELECT I3.luna, MIN(I4.luna)
        FROM luni I3, luni I4
        WHERE I3.luna < I4.luna
        GROUP BY I3.luna) ;

CREATE VIEW inter0 AS
    SELECT marca, numepren, saltarif
    FROM personal2 ;

CREATE VIEW interog AS
    SELECT ' inter'||LTRIM(luni2.luna2)||
        ' AS SELECT inter'||LTRIM(luni2.luna1)||'.*, spornoapte AS SporNoapte'||
        LTRIM(luni2.luna2)||
        ' FROM inter'||LTRIM(luni2.luna1)||
        ', sporuri WHERE inter'||LTRIM(luni2.luna1)||'.marca=sporuri.marca (+)
        AND sporuri.luna (+)=||luni2.luna2||' ;
    AS TEXT
    FROM luni2 ;

CREATE VIEW selectu AS
    SELECT '* FROM inter'||LTRIM(MAX(luna))||' ;' AS text FROM luni ;

spool final.lst

SELECT 'CREATE VIEW '||text
FROM interog
UNION
    SELECT 'SELECT '||text
    FROM selectu ;

spool off

start final.lst
```

Să presupunem că în tabela **sporuri**, pentru anul 2000 sunt introduse date pentru lunile ianuarie, februarie, martie și iunie. Tabela derivată **luni** are musai patru linii (cu

valorile 1,2,3,6). A doua tabelă derivată, luni2, stochează toate succesiunile de luni pentru care există date, și anume: (0,1), (1,2), (2,3) și (3,6).

Toate acestea pregătesc joncțiunea externă a tabelii derivată inițială (botezată inter0) cu datele de pe luna 1 din tabela sporuri, obținându-se tot o tabelă derivată, inter1 (de la *intermediară pentru luna a 1-a*). În continuare, inter1 va fi joncționată (extern) cu datele de pe luna a 2-a din sporuri, rezultând inter2. Procesul se repetă în funcție de numărul de linii din luni2.

Ideea nu ar fi valorat prea mult, dacă nu exista mecanismul de lucru cu script-uri din SQL Plus pe care-l pomeneam mai sus. Practic, toate frazele CREATE VIEW sunt stocate într-o tabelă derivată denumită interog al cărei linii sunt prezentate în [tabelul 1](#).

**Tabel 1. Conținutul tabelii derivate interog**

Linie	Conținut (atributul TEXT)
1	inter1 AS SELECT inter0.*, spornoapte AS SporNoapte1 FROM inter0 , sporuri WHERE inter0.marca=sporuri.marca (+) AND sporuri.luna (+)=1 ;
2	inter2 AS SELECT inter1.*, spornoapte AS SporNoapte2 FROM inter1 , sporuri WHERE inter1.marca=sporuri.marca (+) AND sporuri.luna (+)=2 ;
3	inter3 AS SELECT inter2.*, spornoapte AS SporNoapte3 FROM inter2 , sporuri WHERE inter2.marca=sporuri.marca (+) AND sporuri.luna (+)=3 ;
4	inter6 AS SELECT inter3.*, spornoapte AS SporNoapte6 FROM inter3 , sporuri WHERE inter3.marca=sporuri.marca (+) AND sporuri.luna (+)=6 ;

Tabela derivată selectu are o linie și o coloană; mai exact, conținutul acesteia este \* FROM inter6 ; Scopul acesteia este de a determina ultima tabelă derivată calculată (inter<sub>i</sub>) și de a o afișa prin SELECT \*... .

Execuția efectivă a frazelor CREATE VIEW și SELECT memorate în interog și selectu se realizează prin redirectarea rezultatelor execuției într-un fișier denumit final.lst. Comanda SQL\*Plus pentru redirectare este SPOOL. Pentru curiozitate, verificați conținutul “noului” script - final.lst.

## Concluzii

Luna aceasta am îngroșat puțin gluma cu joncțiunea externă.

**Marin Fotache**