

Fiola de SQL (7)

Clasamente

Ne ocupăm astăzi de o altă manie a contemporaneității: top-urile, adică, mai neaoș, clasamentele. De la Eistein la Hâncu, de la Puff Daddy la Nelu Vijelie, de la Camel și Yes la Enya și Loreena McKennit, lumea (muzicală, sportivă, politică, modei etc.) se împarte și desparte zilnic, săptămânal, lunar..., iar ierarhiile se schimbă după criterii bine stabilite (de obicei, acestea-s cantitative), după gusturi sau chiar ignoranță.

Țin minte că prin presa timpului am fost încântat să citesc că în unul dintre anii '80 (nu-mi amintesc care), în Rusia (URSS) cel mai ascultat disc al anului a fost *The Wall* de Pink Floyd. Un amator de muzică rock/pop ar putea fi interesat de solistul/formația care a vândut cele mai multe discuri pe o perioadă anume sau care sunt primele (ca vânzări) zece albume (CD-uri) ale unui an, cincinal, deceniu etc.

Lucrurile se petrec aidoma și la scară micro. Nu văd de ce o firmă nu ar dori să știe care îi sunt cele mai bine vândute zece produse, cele mai profitabile trei luni din anul calendaristic sau cei mai datornici cinci clienți. Ba, mai mult, chiar aș vedea utilă și o listă în care cele mai vândute 5 produse să fie determinate pentru fiecare județ în parte s.a.m.d

În fiola de astăzi folosim una două cele două tabele tocite de atâta întrebuințare: PERSONAL2{Marca, NumePren, DataNast, Compart, MarcaSef, SalTarifar}, al cărei conținut este prezentat în [figura 1](#).

MARCA	NUMEPREN	DATANAST	COMPART	MARCASEF	SALTARIFAR
1	ANGAJAT 1	01-JUL-62	DIRECTIUNE		6000000
2	ANGAJAT 2	11-OCT-77	FINANCIAR	1	4500000
3	ANGAJAT 3	22-AUG-62	MARKETING	1	4500000
4	ANGAJAT 4		FINANCIAR	2	3800000
5	ANGAJAT 5	30-APR-65	FINANCIAR	2	4200000
6	ANGAJAT 6	09-NOV-65	FINANCIAR	5	3500000
7	ANGAJAT 7		FINANCIAR	5	2800000
8	ANGAJAT 8	31-DEC-60	MARKETING	3	2900000
9	ANGAJAT 9	28-FEB-76	MARKETING	3	4100000
10	ANGAJAT 10	29-JAN-72	RESURSE UMANE	1	5500000

Figura 1. Tabela PERSONAL2

Problema căreia îi vom da târcoale este următoarea: *Cine sunt angajații care au cele mai mari trei salarii tarifare ?*

Dacă am dori să aflăm numai valorile celor trei salarii maxime, lucrurile ar fi simple, o soluție Oracle fiind:

```
select max(p1.saltarifar), max(p2.saltarifar), max(p3.saltarifar)
from (select marca, saltarifar from personal2) p1,
     (select marca, saltarifar from personal2) p2,
     (select marca, saltarifar from personal2) p3
where p1.saltarifar > p2.saltarifar and p2.saltarifar > p3.saltarifar
```

```
SELECT NumePren, SalTarifar  
FROM personal2  
WHERE Saltarifar >=  
    (SELECT MAX(SalTarifar)  
     FROM personal2  
     WHERE SalTarifar <  
         (SELECT MAX(SalTarifar)  
          FROM personal2  
          WHERE SalTarifar <  
              (SELECT MAX(SalTarifar)  
               FROM personal2  
               )  
            )  
        )  
ORDER BY SalTarifar DESC
```

NUMEPREN	SALTARIFAR
ANGAJAT 1	6000000
ANGAJAT 10	5500000
ANGAJAT 2	4500000
ANGAJAT 3	4500000

Din păcate, utilizatorii VFP-ului nu pot beneficia de această soluție, deoarece există mai multe nivele de sub-consultare, lucru netolerat de SGBD. Frustrarea nu ține însă prea mult, deoarece VFP prezintă o opțiune grozavă pentru acest gen de operațiuni – TOP. Fraza VFP următoare are aceeași finalitate ca și precedenta:

Să mai complicăm un pic lucrurile. Dorim ca fiecare linie din rezultat să fie prefixată de poziția efectivă în clasament. În VFP nu putem rezolva problema decât procedural, cu ajutorul unui cursor:

Cursorul c1 prezintă înregistrările dispune în ordine descrescătoare, poziția în top fiind indicată prin numărul extras de funcția RECNO() în cel de-al doilea SELECT. La valori egale ale salariului tarifar, ar trebui ca poziția să fie comună, iar următoarei valori să-i fie decalată numărul de ordine. Programul prezentat în [Listing 1](#) rezolvă situația oarecum onorabil.

Listing 1. Program VFP pentru alcătuirea clasamentului

```
PUBLIC pozitie_, increment_, saltarifar_

SELECT TOP 3 NumePren, SalTarifar ;
FROM personal2 ;
INTO CURSOR c1 ;
ORDER BY SalTarifar DESC

SELECT ordonare(SalTarifar) as Pozitie, * ;
FROM c1

FUNCTION ordonare
PARAMETER st_
IF RECNO() = 1
    increment_ = 0
    pozitie_ = 1
ELSE
    IF st_ = saltarifar_
        increment_ = increment_ + 1
    ELSE
        pozitie_ = pozitie_ + 1 + increment_
        increment_ = 0
    ENDIF
ENDIF
saltarifar_ = st_
RETURN pozitie_
ENDFUNC
```

Oracle face parte din lumea foarte bună a bazelor de date, astfel încât trebuie musai încercată a soluție neprocedurală. Cum RECNO()- ul din VFP este în Oracle ROWNUM, am considerat că avem șanse serioase cu interogarea:

```
SELECT ROWNUM, NumePren, SalTarifar
FROM
    (SELECT NumePren, SalTarifar
     FROM personal2
     ORDER BY SalTarifar DESC )
```

Ideea frazei de mai sus este ca ordonarea să fie efectuată de subconsultare, iar SELECT-ul principal să ia de-a gata înregistrările și să le prefixeze cu numărul care indică poziția în clasament. Ei bine, nu merge ! Motivul principal ține imposibilitatea includerii în subconsultare a opțiunii ORDER BY. Tocmai renunțasem la idee când un fost student (Florin Sârbu) mi-a vândut o idee grozavă, și anume că ordonarea subconsultării poate fi realizată prin GROUP BY. Nu mi-a mai rămas de făcut decât artificii pentru ordonarea descrescătoare:

```
SELECT ROWNUM, NumePren, SalTarifar
FROM
    (SELECT 1-SalTarifar , NumePren, SalTarifar
     FROM personal2
     GROUP BY 1-SalTarifar, NumePren, Saltarifar)
```

În [listing 2](#) este prezentată o altă soluție Oracle 8 mult mai stufoasă și neinteresantă, ce poate servi mai degrabă ca și contraexemplu.

Listing 2. Soluție Oracle 8 pentru obținerea primilor trei clasați

```
SELECT 1, NumePren, SalTarifar
FROM
    (SELECT NumePren, SalTarifar
     FROM personal2
     WHERE SalTarifar >=
         (SELECT MAX(SalTarifar)
          FROM personal2)
    )
UNION
SELECT 2, NumePren, SalTarifar
FROM
    (SELECT NumePren, SalTarifar
     FROM personal2
     WHERE SalTarifar =
         (SELECT MAX(SalTarifar)
          FROM personal2
          WHERE SalTarifar <
              (SELECT MAX(SalTarifar)
               FROM personal2)
    )
    )
UNION
SELECT 3, NumePren, SalTarifar
FROM
    (SELECT NumePren, SalTarifar
     FROM personal2
     WHERE SalTarifar =
         (SELECT MAX(SalTarifar)
          FROM personal2
          WHERE SalTarifar <
              (SELECT MAX(SalTarifar)
               FROM personal2
               WHERE SalTarifar <
                   (SELECT MAX(SalTarifar)
                    FROM personal2)
    )
    )
    )
    )
    )
```

Este adevărat că varianta funcționează corect, dar ce ne facem atunci când dorim să obținem clasamentul primilor 25 de angajați în funcție de salariul tarifar ? Se pot formula mai multe soluții, dar tare miroase a procedural (adică script-ural, în Oracle)...

În schimb, Oracle 8i versiunea 2 aduce o nouă boare în materie de funcții pentru rezolvarea problemelor de acest gen, la modul general, în materie de funcții analitice (vezi și articolul *Funcții OLAP în SQL99 și Oracle 8i2*). Astfel, pentru a ordona salariații după salariul tarifar, dar și pentru a le indica poziția în clasament se întrebuițează funcția **RANK** sau **DENSE_RANK**:

```
SELECT NumePren, SalTarifar,
       RANK() OVER (ORDER BY SalTarifar DESC) AS Pozitie
FROM personal2
```

NUMEPREN	SALTARIFAR	POZITIE
ANGAJAT 1	6000000	1
ANGAJAT 10	5500000	2
ANGAJAT 2	4500000	3
ANGAJAT 3	4500000	3
ANGAJAT 5	4200000	5
ANGAJAT 9	4100000	6
ANGAJAT 4	3800000	7
ANGAJAT 6	3500000	8
ANGAJAT 8	2900000	9
ANGAJAT 7	2800000	10

Figura 3. Utilizarea funcției RANK

Rezultatul din figura 3 numerotează continuu fiecare linie în ordinea dispunerii acestora în rezultat. Întrucât pe poziția 3 sunt doi angajați, numerotarea pozițiilor continuă după aceștia cu 5. Dacă se dorește ca după liniile poziției 3 să se continue cu 4, se folosește o funcție asemănătoare, DENSE_RANK:

```
SELECT NumePren, SalTarifar,
       DENSE_RANK() OVER (ORDER BY SalTarifar DESC) AS Pozitie
FROM personal2
```

Marele atu al funcțiilor RANK, respectiv DENSE_RANK ține de faptul că se poate determina simultan ordinea în clasament după mai multe criterii. De exemplu, dacă dorim să aflăm fiecărui angajat poziția în clasamentele după salariul tarifar și la nivel de organizație, și la nivel de compartiment, se poate formula fraza următoare, al cărui rezultat (în Oracle 8i) este prezentat în figura 4.

```
SELECT SUBSTR(NumePren,1,15) AS NumePren, Compart, SalTarifar,
       RANK() OVER (ORDER BY SalTarifar DESC) AS Pozitie_Generala,
       RANK() OVER (PARTITION BY compart ORDER BY SalTarifar DESC)
       AS Pozitie_Compart
FROM personal2
```

NUMEPREN	COMPART	SALTARIFAR	POZITIE_GENERALA	POZITIE_COMPART
ANGAJAT 1	DIRECTIUNE	6000000	1	1
ANGAJAT 10	RESURSE UMANE	5500000	2	1
ANGAJAT 2	FINANCIAR	4500000	3	1
ANGAJAT 3	MARKETING	4500000	3	1
ANGAJAT 5	FINANCIAR	4200000	5	2
ANGAJAT 9	MARKETING	4100000	6	2
ANGAJAT 4	FINANCIAR	3800000	7	3
ANGAJAT 6	FINANCIAR	3500000	8	4
ANGAJAT 8	MARKETING	2900000	9	3
ANGAJAT 7	FINANCIAR	2800000	10	5

Figura 4. Determinarea pozițiilor la nivel de organizație și de compartiment

O asemenea interogare, transpusă într-o variantă procedurală, fie VFP, fie Oracle (DB2 etc.) se lungește sensibil. Cei ce pot lucra cu ultimele versiuni de DB2, Oracle au motive de mulțumire. Ceilalți, motive de lucru...

Marin Fotache