

# Fiola de SQL (13)

Un pic de OLAP pentru (ceva mai) săraci – Marin Fotache

În ultima fiolă, promiteam că luna aceasta vom discuta câteva opțiuni SQL utile într-un domeniu pentru care nu avem prea multe motive să-i mulțumim minunatului stat român: impozitele și taxele. La drept vorbind, n-am văzut vreun contribuabil nici chiar din țările vestice care să fie bucuros din cale-afară pentru impozitele pe care le plătește, deși pe-acolo statul chiar își face treaba... Dar, întrucât executivul a amânat repetat aplicarea mai multor reglementări legate de impozite și taxe, mi s-a părut în firea lucrurilor, ba chiar mai mult, o dovadă de solidaritate cu guvernul pe altarul consecvenței și punctualității balcanice, să amân discuția până la o dată pe care nu o voi anunța din timp, astfel încât efectul surpriză să fie deplin.

Două materiale publicate în numerele PC Report-ului din noiembrie și decembrie ale anului trecut au fost dedicate opțiunilor OLAP ale noului standard SQL. Am lăsat să vină primăvara pentru a demonstra că analize ale datelor, chiar din cele complexe, sunt posibile și la case mai mici, în SGBD-uri cu mai puține pretenții sau versiuni mai vechi ale primadonelor bazelor de date (Oracle, DB2 etc.). Practic, ținând seama și de spațiu, variantele ce vor fi discutate în continuare sunt cele Visual Fox Pro și Oracle 8.

Bineînțeles că titlul nu onorează în întregime conținutul, și mă gândesc acum în primul rând la un corespondent din București, Marian Munteanu (nu cred că e...), care m-a îndemnat (și a fost convingător) să mă ocup și de SGBD-urile *free*: MySQL, PostgreSQL, Interbase etc., pentru că sunt la îndemâna unei mari părți a firmelor românești, pentru care o soluție de tip server BD reprezintă o fata morgana (financiară). Iar după ce am citit și materialul lui Mircea Sârbu din Net Report-ul pe martie, simt că dorința-i gata. Dar de aici până la performanță mai e drum lung.

## Listing 1. Ponderea fiecărui titlu în totalul vânzărilor – varianta rușinoasă 1

```
*** Pas 1. Salvarea totalului general într-un cursor
(tabela temporară)
SELECT SUM(ValTotala) AS TotalGeneral ;
FROM vinzari_carti ;
INTO CURSOR cTotalGeneral

*** Pas 2. Calculul vinzarilor pentru fiecare titlu
SELECT ISBN, SUM(ValTotala) AS Vinzari ;
FROM vinzari_carti ;
INTO CURSOR cVinzTitluri ;
GROUP BY ISBN

** Pas 3. Calculul ponderii
SELECT ISBN, Vinzari, Vinzari / TotalGeneral ;
FROM cVinzTitluri, cTotalGeneral
```

Pentru a diminua din critici, am schimbat titlul inițial, din *Un pic de OLAP pentru săraci* în *Un pic de OLAP pentru (ceva mai) săraci*. Sesiizați nuanța, nu-i așa ? Și, apoi, dacă m-aș fi referit la SGBD-uri *free*, fiola trebuia să se numească *OLAP pe gratis* sau *OLAP pe degeaba*. Poate peste câteva numere...

Fiola de astăzi este prima ce încalcă, oarecum, „spiritul” relațional. Asta nu înseamnă că face apel la „spiritul” obiectual, ci la extensii pro-

cedurale ale SQL. Ideea de bază este că, atunci când problema este prea pretențioasă pentru SGBD-ul de care dispunem, se poate recurge la „trucuri procedurale”, în cazul nostru funcții, stocate sau nu, funcții folosite în clauza SELECT a frazei cu același nume.

Funcțiile de genul cărora pe care le vom crea astăzi sunt o binecuvântare; cele mai frecvente pot fi incluse în schema bazei (stocate), însă nu trebuie exagerat în această privință, deoarece pot supraîncărca (nu în sens OO) dicționarul (catalogul) bazei.

Pentru comparabilitate cu materialul OLAP din lunile noiembrie-decembrie, folosim aceeași tabelă (denormalizată) ce conține date despre vânzările de cărți ale unei edituri, să zicem Polirom (aveți grijă la reclama mascată !).

```
CREATE TABLE vinzari_carti (
  NrFact DECIMAL(6) NOT NULL,
  DataFact DATE NOT NULL,
  Client VARCHAR(15),
  Loc VARCHAR(15),
  Jud CHAR(2),
  ISBN CHAR(9),
  Cantit DECIMAL (6),
  PretUn DECIMAL (8),
```

Începem cu o problemă aparent banală:

**Să se afișeze ponderea fiecărui titlu în totalul general al vânzărilor.** Reamintim că în Oracle 8i2 cu funcția OLAP `RATIO_TO_REPORT` soluția este banală:

```
SELECT ISBN, ROUND(RATIO_TO_REPORT(ValTotala) OVER (),2)
AS Pondere_Titlu
FROM
  (SELECT ISBN, SUM(ValTotala) AS ValTotala
  FROM vinzari_carti
  GROUP BY ISBN)
```

Ce ne facem cu Visual FoxPro-ul și alte SGBD-uri ale căror producători au grijă mai importante decât funcțiile de analiză a datelor din SQL-99 ? Firește, cea mai la îndemână soluție constă în folosirea unei succesiuni de fraze SELECT care să salveze rezultatele intermediare în tabele derivate, tabele temporare sau cursoare (în sens VFP).

Iată în listing 1 cum ar arăta o asemenea soluție rușinoasă prin proceduralitatea sa în VFP.

## Listing 2. Funcția stocată (VFP) TotalGeneral

```
PROCEDURE TotalGeneral
LOCAL v
DIME v(1,1)

SELECT SUM(ValTotala) ;
FROM vinzari_carti ;
INTO ARRAY v

RETURN v(1,1)
ENDPROC
```

**Listing 3. Script Oracle de creare a funcției TotalGeneral**

```
CREATE OR REPLACE FUNCTION TotalGeneral
RETURN vinzari_carti.valtotala%TYPE
AS
    ValTotala_ vinzari_carti.valtotala%TYPE ;
BEGIN
    SELECT SUM(ValTotala)
    INTO ValTotala_
    FROM vinzari_carti ;
    RETURN NVL(valTotala_, 0) ;
END ;
```

În VFP cursoarele, precum cTotalGeneral sau cVinzTitluri sunt tabele temporare, implicit *read-only* (dar pot fi declarate și actualizabile) care prezintă avantajul ștergerii automate atunci când sunt închise. Atenție, noțiunea de cursor din alte SGBD-uri, precum Oracle, este diferită !

Pentru generalizarea scriptului, în locul cursoarelor se pot folosi tabele derivate, operaționale în mai toate SGBD-urile. Astfel, în pașii 1 și 2 se poate folosi comanda CREATE VIEW AS SELECT..., bineînțeles fără opțiunea INTO CURSOR, scriptul obținut funcționând fără cosmetizări în Oracle, DB2 etc.

Soluțiile de genul celei pe care v-am propus-o sunt dezavuate de orice SQL-ist care se respectă, deși folosite pe din (ms)dos de mulți practicieni. Ceea ce vreau să vă propun în continuare este crearea unor funcții apelabile din interogări, funcții care sunt, de fapt, „prelungiri” ale funcțiilor agregat din SQL (SUM, COUNT, MIN, MAX etc.). Cele mai frecvente pot fi stocate în dicționarul bazei.

Iată, spre exemplu, în listing 2, corpul funcției stocate VFP TotalGeneral ce returnează valoarea totală a vânzărilor:

Această funcție poate fi apelată din orice frază SELECT executată atunci când baza de date este deschisă:

```
SELECT ISBN, SUM(ValTotala) AS Vinzari_Titlu, ;
        SUM(ValTotala) / TotalGeneral() AS Pondere_Titlu ;
FROM vinzari_carti ;
GROUP BY ISBN
```

Rețeta funcționează și la case mai mari. În Oracle, prin scriptul din Listing 3 se crează o funcție (inevitabil stocată) similară celei din VFP.

Fraza SELECT ce face apel la funcția TotalGeneral() este identică VFP. Acest stil de lucru este de mare utilitate, mai ales în versiunile premergătoare 8i2, dar chiar și la versiunile ulterioare (8i2, și apropiata 9i), deoarece rămân suficiente probleme nerezolvabile prin funcții OLAP.

**Să se afișeze ponderea fiecărui titlu în vânzările zilnice.** Soluția OLAP Oracle 8i2 pentru rezolvarea acestei probleme este:

```
SELECT DataFact, ISBN, TO_CHAR(Vinzari,'999999999999') AS
    Vinzari,
    ROUND(RATIO_TO_REPORT(Vinzari) OVER (PARTITION BY
    DataFact),2)
    * 100 || '%' AS Pondere_ISBN_Zi
FROM
    (SELECT DataFact, ISBN, SUM(ValTotala) AS Vinzari
    FROM vinzari_carti
    GROUP BY DataFact, ISBN)
```

După cum observați, ne îngrijim și de modul în care sunt afișate rezultatele (sub formă procentuală). Pentru VFP creăm o funcție stocată căreia i se transmite o dată calendaristică și care returnează totalul vânzărilor în ziua respectivă – vezi listing 4.

Astfel, fraza SELECT ce furnizează în VFP același răspuns precum interogarea OLAP este:

```
SELECT DataFact AS Zi, ISBN, ;
STR(ROUND(SUM(ValTotala) / TotalZi(DataFact)*100,0),2)+'%'
;
AS Pondere_ISBN_Zi ;
FROM vinzari_carti ;
GROUP BY dataFact, ISBN
```

Nu mai includem aici și soluția non-OLAP Oracle, lucrurile derulându-se aidoma primei probleme. Rămân de discutat situațiile în care prin variabile publice pot fi pasate valori de la linie la linie. Treceți această discuție la datoria internă.

Regretul care mă încearcă acum ține numai de faptul că variantele pe care vi le-am propus astăzi sunt extrem de tentante, în detrimentul sofisticărilor SQL ne-procedurale, și induc o lene SQL-istică (și intelectuală, în general) căreia puțini îi pot rezista...

**Listing 4. Funcție VFP care calculează totalul vânzărilor pentru o zi dată**

```
*****
PROCEDURE TotalZi
PARAMETER Zi_
LOCAL v
DIME v(1,1)

SELECT SUM(ValTotala) ;
FROM vinzari_carti ;
INTO ARRAY v ;
WHERE DataFact = Zi_

RETURN NVL(v(1,1),0)
ENDFUNC
```

*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*