

# Fiola de SQL (26)

## Grup de discuții roSQL

– Marin Fotache

**P**e 2 Martie am primit un e-mail de la Adrian Tufă care anunța că s-a deschis pe yahoo, la adresa <http://groups.yahoo.com/group/rosql> un grup „românesc” de discuții dedicat SQL și problemelor colaterale ce țin de acesta. Între timp, numărul membrilor a crescut și, dincolo de ordinea ciudată în care unele mesajele sunt „postate”, exercițiile și soluțiile interesante nu au întârziat să apară. Firește, o parte din subiecte sunt cantonate pe chestiuni de genul „Care-i cel mai bun produs”, sau ingrediente procedurale valabile pe anumite platforme. În cele ce urmează, vă prezint însă o mostră curată SQL-istă, precum și două elegante soluții formulate de membri ai grupului.

Problema a fost propusă spre rezolvare în PostgreSQL de membrul cu numele *no\_hb* (*no* bănuiesc ce înseamnă, însă de *hb* nu vă pot spune nimic), care, pornind de la o tabelă denumită ACTIVITĂȚI, în care se stochează activitățile desfășurate între anumite intervale orare,

```
CREATE TABLE activitati (
    inceput TIMESTAMP PRIMARY KEY ,
    sfirsit TIMESTAMP UNIQUE,
    activitate VARCHAR(30),
    CHECK (inceput < sfirsit)
);
```

dorește să obțină o situația de tip agendă, ca în *figura 1*. Atât comanda de creare, cât și scriptul de populare (disponibile pe situl web al revistei) sunt redactate în PostgreSQL, după cum a dorit autorul.

Problema pare simplă, la prima vedere, și chiar este, numai că am complicat-o zdravăn. Mi-am zis că, pentru a ști dacă, după activitatea de pe linia curentă, urmează sau nu o perioadă „liberă”, tabela ar trebui auto-jonctionată, astfel încât, pe fiecare rând, să avem activitatea curentă, ora de început și de sfârșit a acesteia, precum și ora de început a activității următoare. Zis și făcut:

```
SELECT a1.inceput, a1.sfirsit, a2.inceput AS urmat,
       a1.activitate
FROM activitati a1, activitati a2
WHERE a1.sfirsit <= a2.inceput AND (a1.sfirsit, a2.inceput)
      IN (SELECT a1.sfirsit, MIN(a2.inceput)
          FROM activitati a1, activitati a2
          WHERE a1.sfirsit <= a2.inceput)
```

Figura 1. Rezultatul dorit, sub formă de agendă	
inceput	activitate
2002-05-14 06:30:00+03	masa
2002-05-14 06:45:00+03	LIBER
2002-05-14 08:00:00+03	curs1
2002-05-14 10:00:00+03	laborator1
2002-05-14 12:00:00+03	masa
2002-05-14 12:45:00+03	LIBER
2002-05-14 14:00:00+03	curs2
2002-05-14 15:45:00+03	studiu
2002-05-14 16:30:00+03	LIBER

```
GROUP BY a1.sfirsit)
UNION
SELECT inceput, sfirsit, sfirsit + 1, activitate
FROM activitati
WHERE inceput =
      (SELECT MAX(inceput)
       FROM activitati )
ORDER BY 1
```

SELECT-ul ce urmează UNION-ului are drept scop includerea în rezultat și a ultimei activități (ultima linie din tabelă). Rezultatul este cel din *figura 2*.

De aici încolo lucrurile sunt simple. Dacă, pentru o linie din *figura 2*, ora de început a următoarei activități este mai mare decât ora de sfârșit a activității curente, trebuie inserată, între cele două activități, o linie pentru a indica un interval liber, lucru posibil prin reuniune:

```
SELECT inceput, activitate FROM (
    SELECT a1.inceput, a1.sfirsit, a2.inceput AS urmat,
           a1.activitate
    FROM activitati a1, activitati a2
    WHERE a1.sfirsit <= a2.inceput AND (a1.sfirsit,
                                         a2.inceput) IN
          (SELECT a1.sfirsit, MIN(a2.inceput)
           FROM activitati a1, activitati a2
           WHERE a1.sfirsit <= a2.inceput
           GROUP BY a1.sfirsit)
    UNION
    SELECT inceput, sfirsit, sfirsit + 1, activitate
    FROM activitati
    WHERE inceput =
          (SELECT MAX(inceput)
           FROM activitati )
    ORDER BY 1
    ) x
UNION
SELECT sfirsit, 'LIBER'
FROM (
    SELECT a1.inceput, a1.sfirsit, a2.inceput AS urmat ,
           a1.activitate
    FROM activitati a1, activitati a2
    WHERE a1.sfirsit <= a2.inceput AND (a1.sfirsit,
                                         a2.inceput) IN
          (SELECT a1.sfirsit, MIN(a2.inceput)
           FROM activitati a1, activitati a2
           WHERE a1.sfirsit <= a2.inceput
           GROUP BY a1.sfirsit)
    UNION
    SELECT inceput, sfirsit, sfirsit + 1, activitate
    FROM activitati
    WHERE inceput =
          (SELECT MAX(inceput) FROM activitati )
    ORDER BY 1
    ) x
```

**Figura 2. Activitatea curentă și ora de început a activității următoare**

început	sfârșit	urmat	activitate
2002-05-14 06:30:00+03	2002-05-14 06:45:00+03	2002-05-14 08:00:00+03	masa
2002-05-14 08:00:00+03	2002-05-14 10:00:00+03	2002-05-14 10:00:00+03	curs1
2002-05-14 10:00:00+03	2002-05-14 12:00:00+03	2002-05-14 12:00:00+03	laborator1
2002-05-14 12:00:00+03	2002-05-14 12:45:00+03	2002-05-14 14:00:00+03	masa
2002-05-14 14:00:00+03	2002-05-14 15:45:00+03	2002-05-14 15:45:00+03	curs2
2002-05-14 15:45:00+03	2002-05-14 16:30:00+03	2002-05-15 07:00:00+03	studiu

```
WHERE sfirsit < urmat
ORDER BY inceput
```

Prin urmare, interogarea impresionează mai degrabă prin lungime, și mai puțin prin ingeniozitate.

### Soluții simple

Ei bine, există, ca de obicei, și soluții mult mai lejere decât logoreica interogare de mai sus. Iată una dintre ele:

```
SELECT inceput, activitate
FROM activitati
UNION
SELECT sfirsit, 'LIBER' FROM activitati
WHERE sfirsit NOT IN (SELECT inceput FROM activitati)
```

Dacă sfârșitul unei activități nu coincide cu începutul unei alte activități, atunci se adaugă în rezultat o linie pentru repaos (liber), repaos ce începe cu ora de sfârșit a activității cu pricina. Este ideea sugerată de Liviu Anca în varianta sa mai complexă, pe calapodul SQL Server, ce extrage activitățile pentru o zi dată.

Asemănător, Cătălin Lumezeanu formulează o elegantă soluție în Oracle, bazată pe diferență, ce funcționează, cu mici modificări (EXCEPT în loc de MINUS și alias pentru subconsultare), și în PostgreSQL:

```
SELECT per, 'LIBER' AS activitate
FROM (SELECT sfirsit AS per FROM activitati
EXCEPT SELECT inceput AS per FROM activitati) X
UNION (SELECT inceput, activitate FROM activitati)
```

Oarecum rușinat de complexitatea soluției mele, am găsit o problemă care să o pună în valoare: *în rezultat, începutul fiecărei zile trebuie marcat printr-o linie specială, care să indice ce dată urmează a fi detaliată*. Soluțiile lui Liviu și Cătălin trebuie modificate serios, în timp stufoasă creație proprie este mai nimerită:

```
SELECT inceput, activitate
FROM (
SELECT a1.inceput, a1.sfirsit, a2.inceput AS urmat,
a1.activitate
FROM activitati a1, activitati a2
WHERE a1.sfirsit <= a2.inceput AND (a1.sfirsit, a2.inceput)
IN
(SELECT a1.sfirsit, MIN(a2.inceput)
FROM activitati a1, activitati a2
WHERE a1.sfirsit <= a2.inceput
GROUP BY a1.sfirsit)
UNION
SELECT inceput, sfirsit, sfirsit + 1, activitate
FROM activitati
WHERE inceput =
(SELECT MAX(inceput)
FROM activitati)
ORDER BY 1
```

```
) x
UNION
SELECT sfirsit, 'LIBER'
FROM (
SELECT a1.inceput, a1.sfirsit, a2.inceput AS urmat,
a1.activitate
FROM activitati a1, activitati a2
WHERE a1.sfirsit <= a2.inceput AND (a1.sfirsit, a2.inceput)
IN
(SELECT a1.sfirsit, MIN(a2.inceput)
FROM activitati a1, activitati a2
WHERE a1.sfirsit <= a2.inceput
GROUP BY a1.sfirsit)
UNION
SELECT inceput, sfirsit, sfirsit + 1, activitate
FROM activitati
WHERE inceput = (SELECT MAX(inceput) FROM activitati)
ORDER BY 1x) x
WHERE sfirsit < urmat
UNION
SELECT sfirsit+1, 'Ziua: ' || CAST (urmat AS DATE)
FROM (
SELECT a1.inceput, a1.sfirsit, a2.inceput AS urmat,
a1.activitate
FROM activitati a1, activitati a2
WHERE a1.sfirsit <= a2.inceput AND (a1.sfirsit, a2.inceput)
IN
(SELECT a1.sfirsit, MIN(a2.inceput)
FROM activitati a1, activitati a2
WHERE a1.sfirsit <= a2.inceput
GROUP BY a1.sfirsit)
UNION
SELECT inceput, sfirsit, sfirsit + 1, activitate
FROM activitati
WHERE inceput =
(SELECT MAX(inceput)
FROM activitati)
ORDER BY 1
) x
WHERE CAST (sfirsit AS DATE) < CAST (urmat AS DATE)
UNION
SELECT CAST (inceput AS DATE), 'Ziua: ' || CAST (inceput AS DATE)
FROM activitati
WHERE inceput IN
(SELECT MIN(inceput) FROM activitati)
ORDER BY inceput
```

După cum spuneam, există și alte teme de discuție și soluții interesante. Așa că vă invit să subscrieți...

*Marin Fotache este conferențiar dr. la Catedra de Informatică Economică, UAIC Iași, Facultatea de Economie și Administrarea Afacerilor. Poate fi contactat pe email la: fotache@uaic.ro. ■ 98*