

Fiola de SQL (17)

Reduceri plăcute

– Marin Fotache

Titul prezentului episod este, oarecum, oximoronic, ca să folosesc un termen pe care-l mai plasez uneori în discuțiile intelectuale pentru a câștiga puncte sau ieși din dificultate (aduc un omagiu pe această cale lui Radu Cosașu, cel mai de soi oximoronaut/oximoronist (dilema(tic)) pe care-l cunosc (din cetite).

De obicei, cuvântul *reducere* semnifică *ciuntire*, *indepartare*, chiar *renunțare*, și este asociat cu restructurare și, implicit, criză. Deci, nici urmă de ceva pozitiv, plăcut. Pe vremuri, reducerea se apropiase periculos, din punct de vedere semantic, de alte noțiuni – *reformă* și mai ales *privatizare*. Între timp, reforma a făcut, deseori, casă bună cu *clientela politică* și acum dispunem de câțiva români oacheși (nu mă refer numai la simbolistica de partid) care au trudit din greu pe altarul înfrățirii comisionului cu privatizarea.

Ei bine, în fiola de astăzi ne ocupăm de un alt gen de reduceri, mult mai reconfortant, chiar dorit de cele mai multe firme. În țările cu economie un pic mai dezvoltată – zic *un pic* deoarece folosesc o scală foarte mare – se obișnuiește ca unui client (de fapt, oricărui client) care a plătit facturile în termen scurt să i se ofere o reducere cu atât mai semnificativă cu cât plata este mai iute. Aceasta deoarece firma-încasatoare își ameliorează lichiditatea, un indicator pe care nu-l poate eluda nici un analist financiar sau manager.

Calculul – în SQL și în afara lui – reducerilor ce trebuie acordate pentru fiecare factură încasată, deși plăcut, nu este întotdeauna o treabă tocmai ușoară. O factură poate fi încasată în două sau mai multe tranșe, la anumite distanțe în timp; fiecărei tranșe i se poate aplica un anumit procent de reduceri, sau chiar penalizări (penalizări pe care astăzi le trecem aproape cu vederea).

Să presupunem că datele legate de vânzări și încasări sunt organizate în trei tabele – FACTURI, INCASĂRI și INCASFACT ale căror structuri se poate deduce din comenzile de creare PostgreSQL reunite în listing 1. În mod normal, în afara acestora, baza de date cuprinde și alte tabele

Listing 1. Crearea în PostgreSQL a tabelor ce reflectă vânzările și încasările

```
CREATE TABLE facturi (
    nrfact NUMERIC(8) CONSTRAINT pk_facturi PRIMARY KEY,
    datafact DATE DEFAULT CURRENT_DATE,
    gestiune CHAR(4) NOT NULL,
    codcl NUMERIC(6)
    CONSTRAINT fk_facturi_clienti REFERENCES clienti(codcl)
    , valtotala NUMERIC(16)
);

CREATE TABLE incasari (
    codinc NUMERIC(8)
    CONSTRAINT pk_incasari PRIMARY KEY,
    datainc DATE DEFAULT CURRENT_DATE,
    coddoc CHAR(4),
    nrdoc VARCHAR(16),
    datadoc DATE DEFAULT CURRENT_DATE - 7
);

CREATE TABLE incasfact (
    codinc NUMERIC(8) CONSTRAINT fk_incasfact_incasari REFERENCES
    incasari(codinc) ,
    nrfact NUMERIC(8) CONSTRAINT fk_incasfact_facturi REFERENCES
    facturi(nrfact),
    transa NUMERIC(16) NOT NULL,
    CONSTRAINT pk_incasfact PRIMARY KEY (codinc, nrfact)
);
```

Listing 2. Popularea tabelor FACTURI, INCASARI și INCASFACT (sintaxa Oracle/PostgreSQL)

```
INSERT INTO facturi VALUES (1111, TO_DATE('01/08/2001',
    'DD/MM/YYYY'), '001', 1001, 5399625);
INSERT INTO facturi VALUES (1112, TO_DATE('01/08/2001',
    'DD/MM/YYYY'), '001', 1005, 1337560);
INSERT INTO facturi VALUES (1113, TO_DATE('01/08/2001',
    'DD/MM/YYYY'), '001', 1002, 1160250);
INSERT INTO facturi VALUES (1114, TO_DATE('01/08/2001',
    'DD/MM/YYYY'), '002', 1006, 6786570);
INSERT INTO facturi VALUES (1115, TO_DATE('02/08/2001',
    'DD/MM/YYYY'), '002', 1001, 1651125);
INSERT INTO facturi VALUES (1116, TO_DATE('02/08/2001',
    'DD/MM/YYYY'), '001', 1007, 1383375);
INSERT INTO facturi VALUES (1117, TO_DATE('03/08/2001',
    'DD/MM/YYYY'), '002', 1001, 2320500);
INSERT INTO facturi VALUES (1118, TO_DATE('04/08/2001',
    'DD/MM/YYYY'), '003', 1001, 2052750);
INSERT INTO facturi VALUES (1119, TO_DATE('07/08/2001',
    'DD/MM/YYYY'), '003', 1003, 7242935);
INSERT INTO facturi VALUES (1120, TO_DATE('07/08/2001',
    'DD/MM/YYYY'), '003', 1001, 1066240);
INSERT INTO facturi VALUES (1121, TO_DATE('07/08/2001',
    'DD/MM/YYYY'), '001', 1004, 5438300);

INSERT INTO incasari VALUES (1234, TO_DATE('2001/08/15',
    'YYYY/MM/DD'), 'OP', '111', TO_DATE('2000/08/10', 'YYYY/MM/DD'))
;
INSERT INTO incasari VALUES (1235, TO_DATE('2001/08/15',
    'YYYY/MM/DD'), 'CHIT', '222', TO_DATE('2000/08/15',
    'YYYY/MM/DD'))
;
INSERT INTO incasari VALUES (1236, TO_DATE('2001/08/16',
    'YYYY/MM/DD'), 'OP', '333', TO_DATE('2000/08/09', 'YYYY/MM/DD'))
;
INSERT INTO incasari VALUES (1237, TO_DATE('2001/08/17',
    'YYYY/MM/DD'), 'CEC', '444', TO_DATE('2000/08/10', 'YYYY/MM/DD'))
;
INSERT INTO incasari VALUES (1238, TO_DATE('2001/08/17',
    'YYYY/MM/DD'), 'OP', '555', TO_DATE('2000/08/10', 'YYYY/MM/DD'))
;
INSERT INTO incasari VALUES (1239, TO_DATE('2001/08/18',
    'YYYY/MM/DD'), 'OP', '666', TO_DATE('2000/08/11', 'YYYY/MM/DD'))
;

INSERT INTO incasfact VALUES (1234, 1111, 5399625) ;
INSERT INTO incasfact VALUES (1234, 1118, 1026375) ;
INSERT INTO incasfact VALUES (1235, 1112, 487705) ;
INSERT INTO incasfact VALUES (1236, 1117, 975410) ;
INSERT INTO incasfact VALUES (1236, 1118, 1026375) ;
INSERT INTO incasfact VALUES (1236, 1120, 731557) ;
INSERT INTO incasfact VALUES (1237, 1117, 975410) ;
INSERT INTO incasfact VALUES (1238, 1113, 1160250) ;
INSERT INTO incasfact VALUES (1239, 1117, 369680) ;
```

– precum CLIEȚI, DOCUMENTE – dar nu îndrăzesc să complic situația, așa că opțiunile ce fac referință la alte tabele au fost comentate.

Comenzile INSERT pentru popularea celor trei tabele sunt prezentate în listing 2.

Să presupunem că reducerile se acordă după următorul sistem:

- pentru tranșele plătite în termen de până la 10 zile (INCASARI.DataInc <= FACTURI.DataFact + 10), se acordă o reducere de 10%;
- între 11 și 12 zile procentul este de 9%;
- între 13 și 15 zile – 8%;
- peste 16 zile – nu se acordă reduceri.

Intervalele și procentele de calcul a reducerilor constituie obiectul unei noi tabele pe care denumim TRANȘE_RED – vezi listing 3.

Varianta Visual FoxPro

Începem prezentarea soluțiilor cu o variantă mai complicată redactată în VFP care utilizează cursoare și se prezintă sub forma programului din listing 4.

Figura 1. Cursorul FACT_TRANSE

NrFact	DataFact	ValTotala	LimInf	LimSup	Procent
1111	01/08/2001	539625	01/08/2001	11/08/2001	30
1112	01/08/2001	1337560	01/08/2001	11/08/2001	30
1113	01/08/2001	1166250	01/08/2001	11/08/2001	30
1114	01/08/2001	6766250	01/08/2001	11/08/2001	30
1115	02/08/2001	1851125	02/08/2001	12/08/2001	30
1116	02/08/2001	1363375	02/08/2001	12/08/2001	30
1117	03/08/2001	2320500	03/08/2001	13/08/2001	30
1118	04/08/2001	2052750	04/08/2001	14/08/2001	30
1119	07/08/2001	7342638	07/08/2001	17/08/2001	30
1120	07/08/2001	1866240	07/08/2001	17/08/2001	30
1121	07/08/2001	5438308	07/08/2001	17/08/2001	30
1111	01/08/2001	539625	12/08/2001	13/08/2001	30
1112	01/08/2001	1337560	12/08/2001	13/08/2001	30
1113	01/08/2001	1166250	12/08/2001	13/08/2001	30
1114	01/08/2001	6766250	12/08/2001	13/08/2001	30
1115	02/08/2001	1851125	13/08/2001	14/08/2001	30
1116	02/08/2001	1363375	13/08/2001	14/08/2001	30
1117	03/08/2001	2320500	14/08/2001	15/08/2001	30
1118	04/08/2001	2052750	15/08/2001	16/08/2001	30
1119	07/08/2001	7342638	16/08/2001	17/08/2001	30
1120	07/08/2001	1866240	16/08/2001	17/08/2001	30
1121	07/08/2001	5438308	16/08/2001	17/08/2001	30
1111	01/08/2001	539625	14/08/2001	15/08/2001	30
1112	01/08/2001	1337560	14/08/2001	15/08/2001	30
1113	01/08/2001	1166250	14/08/2001	15/08/2001	30
1114	01/08/2001	6766250	14/08/2001	15/08/2001	30
1115	02/08/2001	1851125	15/08/2001	16/08/2001	30
1116	02/08/2001	1363375	15/08/2001	16/08/2001	30
1117	03/08/2001	2320500	16/08/2001	17/08/2001	30
1118	04/08/2001	2052750	17/08/2001	18/08/2001	30
1119	07/08/2001	7342638	20/08/2001	22/08/2001	30
1120	07/08/2001	1866240	20/08/2001	22/08/2001	30
1121	07/08/2001	5438308	20/08/2001	22/08/2001	30

Figura 2. Cursorul REDUCERI_TRANSE

NrFact	DataFact	ValTotala	LimInf	LimSup	Procent	Reduceri
1111	01/08/2001	539625	01/08/2001	11/08/2001	30	1730
1112	01/08/2001	1337560	01/08/2001	11/08/2001	30	4016
1113	01/08/2001	1166250	01/08/2001	11/08/2001	30	3701
1114	01/08/2001	6766250	01/08/2001	11/08/2001	30	1730
1115	02/08/2001	1851125	02/08/2001	12/08/2001	30	4016
1116	02/08/2001	1363375	02/08/2001	12/08/2001	30	3701
1117	03/08/2001	2320500	03/08/2001	13/08/2001	30	1730
1118	04/08/2001	2052750	04/08/2001	14/08/2001	30	4016
1119	07/08/2001	7342638	07/08/2001	17/08/2001	30	1730
1120	07/08/2001	1866240	07/08/2001	17/08/2001	30	4016
1121	07/08/2001	5438308	07/08/2001	17/08/2001	30	1730

Figura 3. Rezultatul final – suma reducerilor pe facturi

NrFact	DataFact	ValTotala	Reduceri
1111	01/08/2001	539625	431970
1112	01/08/2001	1337560	39016
1117	03/08/2001	2320500	185638
1118	04/08/2001	2052750	184746
1120	07/08/2001	1066240	73155

Varianta Oracle/DB2/PostgreSQL

Am început cu varianta VFP și pentru a ilustra mai bine logica interogării. În Oracle, DB2 sau PostgreSQL 7.1 (atenție la funcția de extragere a întregilor din expresia numerică, funcție care în sintaxa Oracle și PostgreSQL este FLOOR) rezultatul final poate fi obținut printr-o singură interogare, prin folosirea subconsultărilor în clauza FROM:

```
SELECT NrFact, DataFact, ValTotala, SUM(Reduceri) AS
Reduceri
FROM (
(
SELECT FACT_TRANSE.*, DataInc, Transa,
FLOOR(Procent * Transa / 100) AS Reducere
FROM INCASFACT, INCASARI,
(
SELECT NrFact, DataFact, ValTotala, DataFact +
```

Soluția e destul de scumpă, dar operațională. Prima frază SELECT constituie unul dintre puținele exemple de folosire a produsului cartezian urmat de o selecție care nu reprezintă o joncțiune. Cursorul (tabela temporară) FACT_TRANSE conține, pentru fiecare factură cu valoare mai mare ca zero, toate intervalele calendaristice pentru care se pot acorda reduceri. Ținând cont de valorile curente din baza de date, iată, în figura 1, conținutul cursorului.

A doua frază SELECT determină reducerile efective, pe tranșe, conținutul cursorului creat - REDUCERI_TRANSE - fiind prezentat în figura 2.

În final, se însumează toate reducerile pe facturi, rezultatul fiind cel din figura 3.

```
zile_min AS LimInf,
DataFact + zile_max AS LimSup, procent
FROM FACTURI, TRANSE_RED
WHERE procent > 0 AND ValTotala > 0
) FACT_TRANSE
WHERE
INCASFACT.CodInc=INCASARI.CodInc AND
INCASFACT.NrFact = FACT_TRANSE.NrFact AND
INCASARI.DataInc >= FACT_TRANSE.LimInf AND
INCASARI.DataInc <= FACT_TRANSE.LimSup
)
) reduceri_transe
GROUP BY NrFact, DataFact, ValTotala
```

„Posesorii” de DB2 pot redacta și o soluție în care să folosească avantajele expresiilor tabelă. În schimb, pentru SGBD-urile sau versiunile care nu suportă subconsultări în clauza FROM, se poate recurge, în locul script-ului „pedagogic” prezentat în listing 4, la o variantă mai simplă, a cărei sintaxă PostgreSQL 7.0.3 / VFP 6 / Oracle / DB2 este:

```
SELECT INCASFACT.NrFact, DataFact, ValTotala,
SUM (FLOOR(Procent * Transa / 100)) AS Reduceri
FROM FACTURI, INCASFACT, INCASARI, TRANSE_RED
WHERE FACTURI.NrFact=INCASFACT.NrFact AND
INCASFACT.CodInc=INCASARI.CodInc AND
INCASARI.DataInc >= DataFact + zile_min AND
INCASARI.DataInc <= DataFact + zile_max AND
procent > 0 AND ValTotala > 0
GROUP BY INCASFACT.NrFact, DataFact, ValTotala
```

Listing 4. Program VFP de calculare a reducerilor acordate pentru încasări

```
* pentru fiecare factura cu valoare peste zero se insereaza
* in cursorul FACT_TRANSE cite o linie pentru fiecare tranșă
* posibila de reducere
SELECT NrFact, DataFact, ValTotala, DataFact + zile_min AS LimInf, ;
DataFact + zile_max AS LimSup, procent ;
FROM FACTURI, TRANSE_RED ;
INTO CURSOR FACT_TRANSE ;
WHERE procent > 0 AND ValTotala > 0

* se calculeaza reducerile pe fiecare tranșă
SELECT FACT_TRANSE.*, DataInc, Transa, ;
INT(Procent * Transa / 100) AS Reducere ;
INTO CURSOR reduceri_transe ;
FROM INCASFACT INNER JOIN INCASARI ON
INCASFACT.CodInc=INCASARI.CodInc ;
INNER JOIN FACT_TRANSE ;
ON INCASFACT.NrFact = FACT_TRANSE.NrFact ;
WHERE INCASARI.DataInc >= FACT_TRANSE.LimInf AND ;
INCASARI.DataInc <= FACT_TRANSE.LimSup

SELECT NrFact, DataFact, ValTotala, SUM(Reduceri) AS Reduceri ;
FROM reduceri_transe ;
GROUP BY NrFact
```

Listing 3. Crearea și popularea tabelii TRANSE_RED

```
CREATE TABLE transe_red (
zile_min INTEGER,
zile_max INTEGER,
procent INTEGER
);

INSERT INTO transe_red VALUES ( 0, 10, 10 );
INSERT INTO transe_red VALUES ( 11, 12, 9 );
INSERT INTO transe_red VALUES ( 13, 15, 8 );
INSERT INTO transe_red VALUES ( 16, 99999, 0 );
```

Porniți vijelios, cu un exercițiu ce părea inițial destul de complex, am ajuns la o soluție destul de banală. În plus, deși am abordat o problemă cât se poate de coerentă și, pentru unii, chiar interesantă, am o vagă senzație a secundarului. La drept vorbind, plata facturilor este o mare problemă a economiei noastre, dar nu rezezițuina plăților ne chinuie pe noi, ci lentoarea lor sau amânarea până la termene ce se apropie de, dacă-mi permiteți creația, *Horse-Easter*. Prin urmare, trebuie să recunoaștem că este necesar să purtăm o discuție și pe marginea (și chiar în interiorul) *penalizărilor* datorate întârzierii plăților.

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. ■ 87