

Sfinți și corespondenți

Fiola de SQL (10) – Marin Fotache

La momentul în care redactez acest material, stomacul și ficatul sunt în plin avânt și abuz culinaro-bahic. Atunci însă când revista o să apară pe piață, toate acestea vor fi o dureroasă amintire. Prin comparație atât cu firolele SQL anterioare, dar și cu carnea de porc și cârnații, materialul de astăzi este ceva mai digerabil, așa, ca pentru dres după sărbători.

Luăm în discuție o problemă care își are sezonul mai tot timpul anului. Se obișnuiește ca, atunci când este ziua de naștere sau onomastica unei rude, prieten, apropiat, partener de afaceri, șefului (care, de obicei, nu intră în nici una din categoriile precedente), să-i trimitem un mesaj de felicitare, o carte poștală sau „doar” un e-mail. Sigur, un cadou ar fi și mai binevenit, dar nu întotdeauna ne permitem sau „andrisantul” merită cheltuiala.

Un manager populist ar putea folosi baza de date ce conține informații despre personalul firmei pentru a afla din timp angajații ce urmează a-și serba, în perioada următoare, ziua de naștere/onomastica pentru a-și felicita subordonații sau, mă rog, măcar pentru a evita/amăna sancțiunile.

Iar dacă exemplul cu managerul vi se pare exagerat, vă prezint un altul garantat de un bun amic de la catedra de Management-Marketing, legat de ceea ce în marketing se numește *fidelizarea clientului*. Sigur, ca orice român, am avut o oarecare doză de cartezianism, mai ales în ceea ce privește îndoitul; problema fidelizării clientului mi s-a părut la început la fel anxioasă ca și cea a fidelizării soției/soțului... Argumentele sale m-au convins însă rapid că este o practică răspândită în țările occidentale și multe firme importante de la noi au început să culeagă informații despre partenerii de afaceri (în primul rând clienți): vârstă, preferințe, hobby-uri, slăbiciuni etc. și folosesc toate oportunitățile pentru a și-i apropia, astfel încât comunicarea este considerabil netezită. Din această perspectivă, felicitarea cu ocazia onomasticii este obligatorie.

De Sfântul Ion sau Sfântul Dumitru sau orice altă onomastică, SQL are pregătit un operator special – LIKE. Pentru exemplele ce urmează, utilizăm tabela PERSOANE care se poate referi la angajații unei firme, la clienții-persoane fizice sau persoanele cheie de la firmele client sau de la bănci, furnizori etc.

```
CREATE TABLE persoane (
  cnp CHAR(14) CONSTRAINT pk_persoane PRIMARY KEY,
  nume VARCHAR(20),
  prenume VARCHAR(20),
  adresa VARCHAR(40),
  sex CHAR(1) DEFAULT 'B' CONSTRAINT ck_sex CHECK (sex
  IN ('F','B')),
  localitate VARCHAR(25),
  telbirou VARCHAR(10),
  telmobil VARCHAR(10),
  email VARCHAR(30)
);
```



Operatorul LIKE

Ar fi nedrept să legăm operatorul LIKE exclusiv de sindrofii. De multe ori, când se dorește obținerea unor informații din bază, suntem în postura de a nu ști cu exactitate cum se numește un client, produs... sau, pur și simplu, pentru o persoană cunoaștem numai unul dintre prenume etc.

Operatorul LIKE permite compararea unui atribut (expresii) cu un literal utilizând o „mască” ce folosește specificatorii multipli % și _. Simbolurile *procent* și *liniuță_de_jos* (underscore, diferită de cratimă sau linia-de-unire) sunt denumite și *jokeri*. Procentul substituie un șir de lungime variabilă, între 0 și n caractere, în timp ce linia de un singur caracter.

Ce persoane au numele ce conține litera S pe a treia poziție ?

```
SELECT *
FROM PERSOANE
WHERE UPPER(Nume) LIKE '___S%'
```

În mască sunt două underscor-uri ce preced litera S și un procent care-i succede. Funcția UPPER face conversia valorilor atributului Nume în majuscule.

Care sunt persoanele ce trebuie felicitate de Sfântul Ion ?

Aici voiam să ajungem ! Firește, am fi tentați să redactăm varianta:

```
SELECT *
FROM PERSOANE
WHERE UPPER(Prenume) LIKE '%ION%'
```

Figura 1. Tentativă ratată de a extrage „Sfinții Ioni”

CNP	Nume	Prenume	Adresa	Sex	Localitate	TelBirou	TelMobil	EEmail
CNP2	Vasile	Ion		B	Iasi	876543	094222223	lon@a.ro
CNP4	Lazar	Caraion	M.Eminescu, 42	B	Suceava	211888	092222225	laz@b.ro
CNP5	Iurea	Simion	I.Creanga, 44 bis	B	Iasi	543210	NULL	NULL

După cum se vede în rezultatul din figura 1, interogarea nu și-a atins ținta, deoarece, pe de o parte, nu au fost extrase persoanele cu prenume ca Ioan, Ioana, Ioanid, iar, pe de altă parte, au fost eronat extrase și persoanele cu prenumele Caraion și Simion.

Prin urmare, cele trei litere - ION - trebuie să fie plasate la începutul prenumelui. La acest șablon se adaugă și IOAN. Bun, dar dacă pe omul nostru îl cheamă Marius Ion (are două prenume)? Cele două șabloane trebuie să se găsească la începutul fiecărui cuvânt din atributul Prenume. Iar dacă avem în vedere că uneori cele două prenume se despart prin cratimă (Marius-Ion), rezultă o variantă de genul:

```
SELECT *
FROM PERSOANE
WHERE UPPER(Prenume) LIKE 'ION%' OR UPPER(Prenume) LIKE
      'IOAN%'
      OR UPPER(Prenume) LIKE '% ION%' OR UPPER(Prenume) LIKE
      '% IOAN%'
      OR UPPER(Prenume) LIKE '%-ION%' OR UPPER(Prenume) LIKE
      '%-IOAN%'
```

Varianta funcționează în această formă deopotrivă în DB2, Oracle și Visual FoxPro. Deși rare, există cazuri când printre caracterele căutate în valorile unui atribut de tip șir de caractere se găsește chiar unul dintre cele două șabloane, _ sau %. Dacă, spre exemplu, interesează toate firmele client care conțin simbolul % în denumirea lor (s-ar putea ca, la un moment dat, să existe în baza de date o firmă cu numele Procentul vesel % SRL), soluția ar fi:

```
SELECT *
FROM CLIENTI
WHERE DenCl LIKE '%\%' ESCAPE '\'
```

Datorită primului și ultimului simbol procent, poziția caracterului căutat (în cazul nostru, chiar %) nu prezintă importanță. Însă rezultatul corect este obținut grație unui caracter declarat prin clauza ESCAPE. Acesta este backslash-ul, dar poate fi oricare altul. Prin clauza ESCAPE s-a indicat SQL-ului că procentul ce urmează simbolului \ nu este joker, ci are regim de caracter oarecare, ce trebuie căutat în tabelă ca atare.

Problemă de la un corespondent

Pentru problema următoare copyright-ul îl are Cristian Mătase. V-o prezint deoarece reprezintă o bună oportunitate pentru folosirea în șablon a variabilelor/atributelor unei table. Pe scurt, Cristian are o tabelă MESAJE ce conține mesajele trimise unor destinatari din diferite orașe ale țării și, pe baza acesteia, dorește obținerea unei situații de genul celei din figura 2.

Mai întâi se cuvine de remarcat că se impune folosirea și a unei alte table, DESTINATARI, care reprezintă nomenclatorul destinatarilor. Fără această tabelă nu putem obține în statistică linia (Ploiești, 0). La modul cel mai simplist, structura acestei table este: DESTINATARI {Destinatar}.

Soluția este, în aceste condiții, una relativ simplă. Iată varianta Oracle:

```
SELECT d.destinatar, COUNT(DISTINCT Mesaj) AS NrMesaje
FROM destinatari d, mesaje m
WHERE m.destinatari LIKE '%'||d.destinatar||',' OR
m.destinatari LIKE '%, '||d.destinatar||'%' OR
m.destinatari LIKE '%, '||d.destinatar||'%' OR
m.destinatari LIKE d.destinatar
GROUP BY d.destinatar
```

Deoarece pot exista destinatari cu nume foarte apropiate, sunt necesare patru condiții în clauza WHERE separate prin OR:

- prima extrage din MESAJE toți destinatarii în afară de ultimul (atunci când sunt mai mulți), datorită faptului că în șablon, înaintea celui de-al doilea %, apare virgula;
- a doua și a treia extrag din MESAJE toți destinatarii cu excepția primului (atunci când sunt mai mulți), deoarece după primul %, apare fie virgulă și spațiu, fie numai virgulă (deci, se poate scrie: „Iași,Vaslui”, dar și „Iași, Vaslui” (cu spațiu după virgulă));
- a patra extrage destinatarii care sunt singuri la un mesaj.

În rezultatul din figura 2 s-au extras și destinatarii fără nici un mesaj, așa încât ameliorăm soluția anterioară (Oracle):

```
SELECT d.destinatar, COUNT(DISTINCT Mesaj) AS NrMesaje
FROM destinatari d, mesaje m
WHERE m.destinatari LIKE '%'||d.destinatar||',' OR
m.destinatari LIKE '%, '||d.destinatar||'%' OR
m.destinatari LIKE '%, '||d.destinatar||'%' OR
m.destinatari LIKE d.destinatar
GROUP BY d.destinatar
UNION
SELECT d.destinatar, 0 AS NrMesaje
FROM destinatari d
WHERE destinatar NOT IN
(SELECT d.destinatar
FROM destinatari d, mesaje m
WHERE m.destinatari LIKE '%'||d.destinatar||',' OR
m.destinatari LIKE '%, '||d.destinatar||'%' OR
m.destinatari LIKE '%, '||d.destinatar||'%' OR
m.destinatari LIKE d.destinatar)
```

Pentru alte SGBD-uri, trebuie avut în vedere operatorul de concatenare și, în plus, dacă produsul respectiv nu prezintă tipul de date VARCHAR - așa cum este cazul Visual FoxPro - se folosesc funcții de eliminare a spațiilor din valorile atributului d.destinatar sau se controlează modul de comparare a celor doi termeni (SET EXACT).

Figura 2. Situație statistică dintr-o tabelă denormalizată

MESAJE		STATISTICA	
Mesaj	Destinatari	Destinatar	NrMesaje
Mesaj1	Arad, Pitesti	Arad	3
Mesaj2	Arad	Iasi	2
Mesaj3	Iasi, Pitesti	Pitesti	3
Mesaj4	Arad, Iasi, Pitesti	Ploiesti	0
Mesaj5	Vaslui	Vaslui	1

Soluția „normalizată”

„Talibanii” modelului relațional ar putea reproșa tablei MESAJE de-normalitatea, deoarece atributul Destinatar conține valori non-atomice. În plus, suntem în imposibilitatea de a institui o restricție referențială între MESAJE și DESTINATII, astfel încât există pericolul introducerii eronate a numelui unui destinatar.

Este drept că problema se poate rezolva printr-un declanșator pentru tabela MESAJE sau regulă de validare pentru atributul MESAJE.Destinatar, declanșator/regulă în care se extrag cuvintele dintre două virgule și se compară cu valorile atributului DESTINATII.Destinatar.

Structura „curat relațională” a tablei MESAJE ar fi:

```
CREATE TABLE mesaje (
mesaj VARCHAR(20),
destinatar VARCHAR(25) REFERENCES destinatari (destinatar),
CONSTRAINT pk_mesaje PRIMARY KEY (mesaj, destinatar)
);
```

În noua variantă, un mesaj trimis la cinci destinatari va presupune inserarea a cinci linii în tabela MESAJE. Dezavantajul este că statistica dorită se obține mult prea simplu și că trebuie să renunțăm la bunătatea de operator LIKE...

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. ■ 33