

Fiola de SQL (14)

Șefi și copaci – Marin Fotache

După cum știți, casta informaticienilor, chiar și a celor fără sânge albastru (ca mine, de exemplu), este privită de multe ori ca principala sursă de contaminare a limbii române contemporane. Drept e că există sintagme, precum *customizare*, *downloadare* etc. căroră li se pot găsi un echivalent românesc rezonabil. Dar ce te faci cu *middleware*, *server*, *cluster*, *backbone*, *hub* și alte câte și mai câte? Traducerea în trei patru vorbe românești a unui cuvânt englezesc e greu de generalizat, pe alocuri chiar caraghioasă.

Regimentul apărătorilor limbii române străbune în fața împetritării tehnocrate se îngroașă și agresivizează de la an la an. Singura mea mulțumire este că mulți dintre anti-bonjour-iști (de fapt, anti good-afternoon-iști, că engleza e acu la modă) au contribuții majore la actuala esență lemnoasă a graiului mioritic și, pe vremuri, au făcut din obediență (inclusiv lingvistică) o înfloritoare (chira dacă urât mirositoare) meserie.

În orice caz, aveți grijă, suntem urmăriți! Așa încât, în loc de titlul *manageri și structuri arborescente*, cum era mai nimerit, am preferat, precaut și anxios, *șefi și copaci*. E mioritic & patriotic.

Există mai multe moduri în care se pot reprezenta structurile ierarhice în bazele de date relaționale. Cine nu mă crede poate consulta cărțile sau articolele din *DBMS Magazine* și *Intelligent Enterprise* ale

Figura 1. Conținutul tabelii PERSONAL2

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-6-	MARKETING	3	2900000
9	ANGAJAT 9	28-FEB-76	MARKETING	3	4100000
10	ANGAJAT 10	29-JAN72	RESURSE UMANE	1	3700000

lui Joe Celko. Materialul de față vizează doar una din posibilități, cea încorporată în tabela PERSONAL2 pe care am folosit-o și cu alte date - (figura 1).

Tabela conține câteva date despre angajații unei firme/organizații, însă reflectă și stuctura ierarhică a firmei, deoarece, pentru fiecare angajat, este indicată marca șefului său direct (angajatul cu marca 1 este chiar directorul general, pentru linia corespunzătoare acestuia valoarea atributului MarcaSef fiind NULL.). Pornind de aici, vom încerca să formulăm câteva întrebări la care vom răspunde tot noi.

Soluția clasică - autojuncțiunea

Pentru aflarea majorității informațiilor privitoare la ierarhia firmei, soluția obișnuită în SQL constă în joncționarea a două instanțe ale tabelii PERSONAL2.

Cum se numește șeful Angajatului 7?

Soluție SQL92/DB2/VFP:

```
SELECT SEFI.NumePren
FROM PERSONAL2 SUBORDONATI INNER JOIN PERSONAL2 SEFI
    ON SUBORDONATI.MarcaSef = SEFI.Marca
WHERE SUBORDONATI.NumePren = 'ANGAJAT 7'
```

Care sunt subordonații direcți ai Angajatului 2?

Soluție SQL92/DB2/VFP:

```
SELECT SUBORDONATI.NumePren
FROM PERSONAL2 SUBORDONATI INNER JOIN PERSONAL2 SEFI
    ON SUBORDONATI.MarcaSef = SEFI.Marca
WHERE SEFI.NumePren = 'ANGAJAT 2'
```

Firește, cele două probleme pot fi rezolvate în aproape toate SGBD-urile și prin subconsultări, astfel:

```
SELECT NumePren
FROM PERSONAL2
WHERE Marca IN
    (SELECT MarcaSef
     FROM PERSONAL2
     WHERE NumePren =
       'ANGAJAT 7')
```

respectiv:

```
SELECT NumePren
FROM PERSONAL2
WHERE MarcaSef IN
    (SELECT Marca
     FROM PERSONAL2
     WHERE NumePren = 'ANGAJAT 2')
```

Figura 2. Numărul subordonaților pentru fiecare salariat

NUMEPREN	NR_SUBORDONAȚI
ANGAJAT 1	3
ANGAJAT 10	0
ANGAJAT 2	2
ANGAJAT 3	2
ANGAJAT 4	0
ANGAJAT 5	2
ANGAJAT 6	0
ANGAJAT 7	0
ANGAJAT 8	0
ANGAJAT 9	0

Câți subordonați are fiecare angajat al firmei?

Interogarea:

```
SELECT SEFI.NumePren, COUNT(*) AS Nr_Subordonati
FROM PERSONAL2 SUBORDONATI INNER JOIN PERSONAL2 SEFI
    ON SUBORDONATI.MarcaSef = SEFI.Marca
GROUP BY SEFI.NumePren
```

extrage numai pe cei care au măcar un subordonat. Dacă dorim ca rezultatul să conțină toți angajații în ordine alfabetică, trebuie folosită joncțiunea externă la dreapta și modificat argumentul funcției COUNT:

```
SELECT SEFI.NumePren, COUNT(SUBORDONATI.MarcaSef) AS
    Nr_Subordonati
FROM PERSONAL2 SUBORDONATI RIGHT OUTER JOIN PERSONAL2 SEFI
    ON SUBORDONATI.MarcaSef = SEFI.Marca
GROUP BY SEFI.NumePren
```

Figura 2 conține urmările acestei interogări.

Care este nivelul ierarhic al fiecărui salariat ?

De la început, știm că ierarhia reflectată în tabela PERSONAL2 se derulează pe patru nivele, așa încât o interogare pentru rezolvarea problemei poate fi:

```
SELECT 'Nivel 1' AS Nivel, NumePren, Compart
FROM PERSONAL2
WHERE MarcaSef IS NULL
UNION
SELECT 'Nivel 2' AS Nivel, NIVEL2.NumePren, NIVEL2.Compart
FROM PERSONAL2 NIVEL1
INNER JOIN PERSONAL2 NIVEL2
ON NIVEL1.Marca = NIVEL2.MarcaSef AND
Nivel1.MarcaSef IS NULL
UNION
SELECT 'Nivel 3' AS Nivel, NIVEL3.NumePren, NIVEL3.Compart
FROM PERSONAL2 NIVEL1
INNER JOIN PERSONAL2 NIVEL2
ON NIVEL1.Marca = NIVEL2.MarcaSef AND
Nivel1.MarcaSef IS NULL
INNER JOIN PERSONAL2 NIVEL3
ON NIVEL2.Marca = NIVEL3.MarcaSef
UNION
SELECT 'Nivel 4' AS Nivel, NIVEL4.NumePren, NIVEL4.Compart
FROM PERSONAL2 NIVEL1
INNER JOIN PERSONAL2 NIVEL2
ON NIVEL1.Marca = NIVEL2.MarcaSef AND
Nivel1.MarcaSef IS NULL
INNER JOIN PERSONAL2 NIVEL3
ON NIVEL2.Marca = NIVEL3.MarcaSef
INNER JOIN PERSONAL2 NIVEL4
ON NIVEL3.Marca = NIVEL4.MarcaSef
ORDER BY NumePren
```

Iată și rezultatul în figura 3.

Figura 3. Nivelul ierarhic al fiecărui angajat

NUMEPREN	COMPART	NIVEL
ANGAJAT 1	DIRECTIUNE	Nivel 1
ANGAJAT 10	RESURSE UMANE	Nivel 2
ANGAJAT 2	FINANCIAR	Nivel 2
ANGAJAT 3	MARKETING	Nivel 2
ANGAJAT 4	FINANCIAR	Nivel 3
ANGAJAT 5	FINANCIAR	Nivel 3
ANGAJAT 6	FINANCIAR	Nivel 4
ANGAJAT 7	FINANCIAR	Nivel 4
ANGAJAT 8	MARKETING	Nivel 3
ANGAJAT 9	MARKETING	Nivel 3

trebuie să știm, aprioric, numărul nivelelor ierarhice, iar în al doilea rând, la un număr mult mai mare de nivele, întinderea consultării crește sensibil.

O variantă mai elegantă, ca volum de scris, nu și ca resurse consumate este:

```
SELECT NIVEL4.*,
CASE
WHEN NIVEL4.MarcaSef IS NULL THEN 1
WHEN NIVEL3.MarcaSef IS NULL THEN 2
WHEN NIVEL2.MarcaSef IS NULL THEN 3
WHEN NIVEL1.MarcaSef IS NULL THEN 4
```

```
END AS Nivel
FROM PERSONAL2 NIVEL1
RIGHT OUTER JOIN PERSONAL2 NIVEL2
ON NIVEL1.Marca = NIVEL2.MarcaSef AND
Nivel1.MarcaSef IS NULL
RIGHT OUTER JOIN PERSONAL2 NIVEL3
ON NIVEL2.Marca = NIVEL3.MarcaSef
RIGHT OUTER JOIN PERSONAL2 NIVEL4
ON NIVEL3.Marca = NIVEL4.MarcaSef
```

Se jonctionează extern patru instanțe ale tabelului PERSONAL2, câte una pentru fiecare nivel ierarhic, astfel încât au și fost denumite NIVEL1... NIVEL4.

Fraza SELECT compune ierarhia pornind de la NIVEL1 care reprezintă o instanță a PERSONAL2 cu o singură linie, cea a directorului general, instanță jonctiionată extern la dreapta cu NIVEL2 care va furniza subordonații direcți ai înregistrării rădăcină s.a.m.d.

Să se afișeze structura ierarhică a firmei.

Practic, dorim o formă de vizualizare a angajaților care să țină cont de modul lor de subordonare – ca în figura 4.

Dintre variantele DB2 operaționale, începem cu una destul de stufoasă bazată pe expresia tabelă NIVELE.

Figura 4. Vizualizarea ierarhiei

NUME	COMPART	SEF1	SEF2	SEF3	SEF4
ANGAJAT 1	DIRECTIUNE	1	1	1	1
- ANGAJAT 2	FINANCIAR	1	2	2	2
- - ANGAJAT 4	FINANCIAR	1	2	4	4
- - ANGAJAT 5	FINANCIAR	1	2	5	5
- - - ANGAJAT 6	FINANCIAR	1	2	5	6
- - - ANGAJAT 7	FINANCIAR	1	2	5	7
- ANGAJAT 3	MARKETING	1	3	3	3
- - ANGAJAT 8	MARKETING	1	3	8	8
- - ANGAJAT 9	MARKETING	1	3	9	9
- ANGAJAT 10	RESURSE UMANE	1	10	10	10

```
WITH NIVELE AS
(SELECT NIVEL4.*,
CASE
WHEN NIVEL4.MarcaSef IS NULL THEN 1
WHEN NIVEL3.MarcaSef IS NULL THEN 2
WHEN NIVEL2.MarcaSef IS NULL THEN 3
WHEN NIVEL1.MarcaSef IS NULL THEN 4
END AS Nivel
FROM PERSONAL2 NIVEL1
RIGHT OUTER JOIN PERSONAL2 NIVEL2
ON NIVEL1.Marca = NIVEL2.MarcaSef AND
Nivel1.MarcaSef
IS NULL
RIGHT OUTER JOIN PERSONAL2 NIVEL3
ON NIVEL2.Marca = NIVEL3.MarcaSef
RIGHT OUTER JOIN PERSONAL2 NIVEL4
ON NIVEL3.Marca = NIVEL4.MarcaSef )
```

```
SELECT NIVELE.NumePren AS Nume, NIVELE.Compart, NIVELE.Marca
AS Sef1,
NIVELE.Marca AS Sef2, NIVELE.Marca AS Sef3, NIVELE.Marca AS
Sef4
FROM NIVELE
WHERE Nivel = 1
UNION
```

```

SELECT '——' || N2.NumePren AS Nume, N2.Compart, N1.Marca AS
    Sef1,
N2.Marca AS Sef2, N2.Marca AS Sef3, N2.Marca AS Sef4
FROM NIVELE N2 INNER JOIN NIVELE N1 ON
N2.Nivel=2 AND N2.MarcaSef=N1.Marca AND N1.Nivel=1
UNION
SELECT '———' || N3.NumePren AS Nume, N3.Compart, N1.Marca
    AS Sef1,
N2.Marca AS Sef2, N3.Marca AS Sef3, N3.Marca AS Sef4
FROM NIVELE N3 INNER JOIN NIVELE N2
ON N3.Nivel=3 AND N3.MarcaSef=N2.Marca AND N2.Nivel=2
    INNER JOIN NIVELE N1
ON N2.Nivel=2 AND N2.MarcaSef=N1.Marca AND N1.Nivel=1
UNION
SELECT '————' || N4.NumePren AS Nume, N4.Compart,
    N1.Marca AS Sef1,
N2.Marca AS Sef2, N3.Marca AS Sef3, N4.Marca AS Sef4
FROM NIVELE N4 INNER JOIN NIVELE N3
ON N4.Nivel=4 AND N4.MarcaSef=N3.Marca AND N3.Nivel=3
    INNER JOIN NIVELE N2
ON N3.Nivel=3 AND N3.MarcaSef=N2.Marca AND N2.Nivel=2
    INNER JOIN NIVELE N1
ON N2.Nivel=2 AND N2.MarcaSef=N1.Marca AND N1.Nivel=1
ORDER BY Sef1, Sef2, Sef3, Sef4

```

Lucrurile pot fi simplificate sensibil utilizând o structură de tip CASE prin care determinăm nivelul de subordonare, nivel care va determina de câte ori se repetă (funcția DB2 REPEAT) grupul de șase liniiuțe, astfel încât listarea primei coloane este asemănătoare celei din figura 4.

```

SELECT RTRIM ( CHAR ( REPEAT ( '-', 6 * ( (
CASE
    WHEN NIVEL4.MarcaSef IS NULL THEN 1
    WHEN NIVEL3.MarcaSef IS NULL THEN 2
    WHEN NIVEL2.MarcaSef IS NULL THEN 3
    WHEN NIVEL1.MarcaSef IS NULL THEN 4
END ) - 1 ) ) ) ) || NIVEL4.NumePren AS Nume,
    NIVEL4.Compart
FROM PERSONAL2 NIVEL1
    RIGHT OUTER JOIN PERSONAL2 NIVEL2

```

```

ON NIVEL1.Marca = NIVEL2.MarcaSef AND
    Nive11.MarcaSef IS NULL
RIGHT OUTER JOIN PERSONAL2 NIVEL3
    ON NIVEL2.Marca = NIVEL3.MarcaSef
RIGHT OUTER JOIN PERSONAL2 NIVEL4
    ON NIVEL3.Marca = NIVEL4.MarcaSef

```

În VFP, același rezultat se obține printr-un lanț de IF-uri imediate (IIF-uri) :

```

SELECT REPLICATE('-', (IIF(ISNULL(NIVEL4.MarcaSef), 1, ;
    IIF(ISNULL(NIVEL3.MarcaSef), 2, ;
        IIF(ISNULL(NIVEL2.MarcaSef), 3, 4) ) ) - 1)
;
* 7)+NIVEL4.NumePren AS Nume, NIVEL4.Compart ;
FROM PERSONAL2 NIVEL1 ;
    RIGHT OUTER JOIN PERSONAL2 NIVEL2 ;
    ON NIVEL1.Marca = NIVEL2.MarcaSef AND Nive11.MarcaSef IS
    NULL ;
    RIGHT OUTER JOIN PERSONAL2 NIVEL3 ;
        ON NIVEL2.Marca = NIVEL3.MarcaSef ;
    RIGHT OUTER JOIN PERSONAL2 NIVEL4 ;
        ON NIVEL3.Marca = NIVEL4.MarcaSef

```

Interogări arborescente în Oracle

Până acum au fost formulate numai soluții DB2 și VFP, lăsându-se să se înțeleagă că redactarea variantelor echivalente în Oracle ar fi o temă pentru acasă/la birou. De fapt, ne-am rezervat pentru cele ce urmează. Dialectul SQL al Oracle are câteva opțiuni speciale pentru parcurgerea structurilor ierarhice, cele mai importante fiind START WITH și CONNECT BY.

Care este nivelul ierarhic al fiecărui salariat ?

Soluția următoare conduce la rezultatul din figura 5.

```

SELECT PERSONAL2.*, LEVEL
FROM PERSONAL2
START WITH MarcaSef IS NULL
CONNECT BY PRIOR Marca=MarcaSef

```

Figura 5. Interogare ierarhică

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

Construirea structurii ierarhice începe cu înregistrarea (înregistrările) care îndeplinesc condiția din clauza **START WITH**. Această înregistrare-părinte va fi legată de înregistrarea sau înregistrările copil prin condiția **Marca = MarcaSef**. Clauza **PRIOR** plasată în stânga condiției semnifică: valoarea atributului **Marca** din părinte trebuie să fie egală cu valoarea **MarcaSef** din înregistrările copil.

Prin **CONNECT BY** sunt selectate toate generațiile succesive de linii-copil (copii, nepoți, strănepoți etc.). După construirea ierarhiei, se elimină tuplurile ce nu îndeplinesc condiția formulată în clauza **WHERE**. Este important de notat că selecția se aplică linie cu linie, iar eliminarea unei linii-părinte nu atrage automat eliminarea copiilor, nepoților s.a.m.d. Dacă există clauza **ORDER BY**, aceasta va determina dispunerea înregistrărilor în rezultat. În lipsa clauzei de ordonare, înregistrările sunt dispuse în funcție de ordinea parcurgerii arborelui.

Un avantaj major a interogărilor ierarhice ține de folosirea pseudo-coloanei **LEVEL** ce semnifică tocmai nivelul ierarhiei, relativ la înregistrarea/înregistrările „rădăcină” care îndeplinește/îndeplinesc condiția din **START WITH**.

Ca principale restricții trebuie amintit că **SELECT**-ul care execută o interogare ierarhică nu poate efectua o joncțiune și nici extrage date dintr-o tabelă virtuală creată printr-o joncțiune.

Cum se numește șeful Angajatului 7 ?

Dacă folosim interogarea:

```
SELECT PERSONAL2.*, LEVEL
FROM PERSONAL2
START WITH NumePren = 'ANGAJAT 7'
CONNECT BY PRIOR MarcaSef = Marca
```

se obține raportul din figura 6.

Răspunsul exact la întrebare (figura 7) presupune următoarea soluție:

```
SELECT PERSONAL2.*, LEVEL
FROM PERSONAL2
WHERE LEVEL - 1 =
( SELECT LEVEL
  FROM PERSONAL2
  WHERE NumePren = 'ANGAJAT 7'
  START WITH NumePren = 'ANGAJAT 7'
CONNECT BY PRIOR MarcaSef = Marca )
START WITH NumePren = 'ANGAJAT 7'
CONNECT BY PRIOR MarcaSef = Marca
```

Să se afișeze structura ierarhică a firmei.

Nu putem să reprezentăm arborele ierarhic cu „verdele în jos”, ci cu rădăcina la stânga, la fel ca în soluția DB2 de acum câteva figuri. Indentarea subordonaților se obține cu ajutorul funcției **LPAD** și pseudo-atributului **LEVEL** după cum urmează:

```
SELECT LPAD(' ', 5*(LEVEL - 1), '-') || numepren AS Nume,
       Compart
FROM PERSONAL2
START WITH MarcaSef IS NULL
CONNECT BY PRIOR Marca = MarcaSef
```

Închei, (aproape) ca de obicei, cu promisiuni evazive. Ne-ar rămâne de discutat prin fiole viitoare despre alte modalități de reflectare a structurilor arborescente, a avantajelor și dezavantajelor fiecăreia și, firește, a interogărilor necesare obținerii diverselor informații.

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

Figura 6. Toți șefii Angajatului 7

MARCA	NUMEPREN	DATANAST	COMPART	MARCASEF	SALTARIFAR	LEVEL
7	ANGAJAT 7		FINANCIAR	5	2800000	1
5	ANGAJAT 5	30-APR-65	FINANCIAR	2	4200000	2
2	ANGAJAT 2	11-OCT-77	FINANCIAR	1	4500000	3
1	ANGAJAT 1	01-JUL-62	DIRECTIUNE		6000000	4

Figura 7. Șeful direct al Angajatului 7

MARCA	NUMEPREN	DATANAST	COMPART	MARCASEF	SALTARIFAR	LEVEL
5	ANGAJAT 5	30-APR-65	FINANCIAR	2	4200000	2