

Easier Hierarchical Queries

Conf.dr. Mihaela MUNTEAN
Catedra de Informatică Economică, ASE București

If a table contains hierarchical data, I can select rows in a hierarchical order using SELECT statement with START WITH clause and CONNECT BY clause. Oracle Database 10g introduces new features for hierarchical queries. In this article I discussed about these new features using examples.

Key words: relational databases, hierarchical queries, root rows, parent rows, child rows, SQL.

Oracle 10g oferă o serie de facilități care simplifică realizarea cererilor ierarhice și permite utilizatorilor să construiască cereri ierarhice foarte ușor și rapid. Într-o cerere ierarhică se poate specifica: direcția de parcur-

gere a ierarhiei, nodul rădăcină al ierarhiei, se pot elimina noduri și ramuri din ierarhie etc. Pentru exemplificare, vom utiliza tabela *Angajati* cu următoarea structură:

```
SQL> create table angajati(marca number(3) primary key, nume varchar2(30),
2   functie varchar2(25), codsef number(3) references angajati(marca));
SQL> select marca, nume, codsef from angajati;
MARCA NUME          CODSEF
```

```
-----
1          Popescu
2          Necula          1
3          Niculescu      1
4          Vinci          1
5          Grigore        1
6          Ionescu        2
7          Dobre          2
8          Dumitru        3
9          Alter          6
10         Muntean        6
11         Constantin     8
12         Nicolae        8
13         Negoe          8
```

➤ Pseudocoloana **CONNECT_BY_ISLEAF** returnează 1 dacă tuplul curent nu are copii, altfel returnează 0. De exemplu, se poate ve-

rifica dacă angajatul Muntean are subordonați:

```
SQL> select nume, connect_by_isleaf from angajati
2   start with nume='Muntean' connect by prior marca=codsef;
NUME          CONNECT_BY_ISLEAF
```

```
-----
Muntean          1
```

Următorul exemplu afișează primele două niveluri din ierarhie, indicând pentru fiecare nivel dacă are sau nu copii:

```
SQL> select lpad(' ',2*(level-1)) || nume nume, connect_by_isleaf, level
2   from angajati where level<=2
3   start with codsef is null connect by prior marca=codsef ;
```

NUME	CONNECT_BY_ISLEAF	LEVEL
Popescu	0	1
Necula	0	2
Niculescu	0	2
Vinciu	1	2
Grigore	1	2

Se observă că nodurile corespunzătoare angajaților Vinciu și Grigore sunt noduri frunză sau cei doi angajați nu au subordonați.

➤ Clauza **ORDER SIBLINGS BY**. Într-o cerere ierarhică nu se specifică clauza **ORDER BY** sau **GROUP BY**, deoarece aceste clauze distrug ordinea ierarhică specificată de clauza

CONNECT BY. Dacă dorim totuși să ordonăm tuplurile care au același părinte, se utilizează clauza **ORDER SIBLINGS BY**. De exemplu, se vor afișa toți subordonații lui Dumitru, inclusiv Dumitru, într-o structură ierarhică:

```
SQL>select lpad(' ',2*(level-1)) || nume "nume" from angajati
2      start with nume=Dumitru' connect by prior marca=codsef;
nume
-----
Dumitru
  Constantin
  Nicolae
  Negoe
```

Observăm că subordonații lui Dumitru nu sunt ordonați alfabetic. Pentru a-i ordona al-

fabetic se utilizează clauza **ORDER SIBLINGS BY**:

```
SQL>select lpad(' ',2*(level-1)) || nume "nume" from angajati
2      start with nume='Dumitru' connect by prior marca=codsef
3      order siblings by nume;
nume
-----
Dumitru
  Constantin
  Negoe
  Nicolae
```

➤ Operatorul **CONNECT_BY_ROOT** returnează rădăcina ierarhiei pentru tuplul curent. Acest operator nu se poate utiliza în clauza

START WITH sau în clauza **CONNECT BY**. De exemplu, se vor afișa pentru angajatul Muntean, șefii în ordine ierarhică:

```
SQL> select nume, connect_by_root nume sef from angajati
2      where nume='Muntean' and level>1 connect by prior marca=codsef;
NUME          SEF
-----
Muntean      Necula
Muntean      Ionescu
Muntean      Popescu
```

În exemplul anterior, atributul *nume* este precedat de operatorul **CONNECT_BY_ROOT**. Oracle va returna acele valori ale atributului *nume* corespunzătoare șefilor angajatului Muntean (șeful direct Necula, apoi Ionescu șeful lui Necula și indirect al lui Muntean, apoi Popescu șeful lui Ionescu și indirect al

lui Muntean).

➤ Funcția **SYS_CONNECT_BY_PATH (atribut, separator)** returnează calea de la nodul specificat (în exemplul anterior nodul corespunzător angajatului Muntean) la fiecare nod părinte corespunzător. Pentru a construi calea, se utilizează atributul *nume*, iar ca

separator (/).

```
SQL> select nume, connect_by_root nume sef,
2      sys_connect_by_path (nume, '/') calea from angajati
3      where nume='Muntean' and level>1 connect by prior marca=codsef;
```

NUME	SEF	CALEA
Muntean	Necula	/Necula/Ionescu/Muntean
Muntean	Ionescu	/Ionescu/Muntean
Muntean	Popescu	/Popescu/Necula/Ionescu/Muntean

sau:

```
SQL> select nume, sys_connect_by_path (nume, '/') calea from angajati
2      start with nume='Muntean' connect by prior codsef=marca;
```

NUME	CALEA
Muntean	/Muntean
Ionescu	/Muntean/Ionescu
Necula	/Muntean/Ionescu/Necula
Popescu	/Muntean/Ionescu/Necula/Popescu

➤ **CONNECT_BY_ISCYCLE** Pseudocoloana **CONNECT_BY_ISCYCLE** returnează 1 dacă tuplul curent are un copil care este la rândul său părinte pentru tuplul curent (se formează o “buclă” sau *loop*). Altfel returnează 0. De exemplu, Popescu este director și nu are șef (atributul *codsef* are valoarea *null*).

Necula îl are șef pe Popescu. Se face următoarea modificare: Necula devine șeful lui Popescu:

```
SQL>update angajati set codsef=2
where marca=1;
```

Se execută apoi următoarea cerere:

```
SQL>select nume, level from angajati
2      start with nume='Popescu' connect by prior marca=codsef;
```

Apare eroarea:

```
ERROR:
ORA-01436: CONNECT BY loop in user data
no rows selected
```

Apare o “buclă” (Popescu este șef pentru Necula și Necula este șef pentru Popescu) și cererea nu returnează nimic. Se poate utiliza pseudocoloana **CONNECT_BY_ISCYCLE** ca-

re va specifica tuplurile ce formează o “buclă” (pseudocoloana *connect_by_iscycle* returnează valoarea 1):

```
SQL> select nume, connect_by_iscycle "bucla" from angajati
2      where connect_by_iscycle=1 start with nume='Popescu'
3      connect by nocycle prior marca=codsef;
```

NUME	bucla
Necula	1

Se utilizează parametrul **NOCYCLE** în clauza **CONNECT BY** pentru a afișa tuplurile chiar dacă există o “buclă”.

2005
[***], Oracle Corporation, *Oracle Database SQL Reference 10g Release 1 (10.1)*, part number B10759-01, www.oracle.com

Bibliografie

[MM05], Mihaela Muntean, *Culegere de probleme în Oracle*, editura ASE, București,