

## 1. TEORIE

### Cereri ierarhice

→ dacă un tabel conține date ierarhice, ele pot fi selectate pe baza acestei ierarhii

```
SELECT nume_coloane  
FROM nume_tabel  
START WITH conditie  
CONNECT BY PRIOR nume_coloana_1 = nume_coloana_2;
```

#### *START WITH*

- specifică o condiție care identifică liniile ce urmează să fie considerate ca rădăcini
- dacă se omite această clauză, sistemul Oracle utilizează toate liniile din tabel drept linii rădăcină

#### *CONNECT BY*

- specifică o condiție care identifică relația dintre liniile „părinte” și „copil” ale ierarhiei
- condiția trebuie să conțină operatorul PRIOR pentru a face referință la linia „părinte”

#### Operatorul *PRIOR*

- face referință la linia „părinte”. Plasarea acestui operator determină direcția interogării, dinspre „părinte” spre „copil” (top-down) sau invers (bottom-up)
- Traversarea top-down, respectiv bottom-up a arborelui se realizează prin specificări de forma următoare:

Top-down: **CONNECT BY PRIOR** cheie\_parinte = cheie\_copil;

Bottom-up: **CONNECT BY PRIOR** cheie\_copil = cheie\_parinte;

**OBS 1:** Operatorul PRIOR poate fi plasat în fața oricărui membru al condiției specificate în clauza CONNECT BY.

**OBS 2:** Liniile „părinte” ale interogării sunt identificate prin clauza START WITH. Pentru a găsi liniile „copil”, server-ul evaluează expresia din dreptul operatorului PRIOR pentru linia „părinte”, și cealaltă expresie pentru fiecare linie a tabelului. Înregistrările pentru care condiția este adevărată vor fi liniile „copil”. Spre deosebire de START WITH, în clauza CONNECT BY nu pot fi utilizate subcereri.

Pseudocoloana **LEVEL** poate fi utilă într-o cerere ierarhică. Aceasta determină lungimea drumului de la rădăcină la un nod.

**Exemplu:**

Dorim să afișăm angajații în ordine ierarhică (managerii urmați de subordonații lor), pornind de la angajatul cu id-ul 100:

```
SELECT last_name, employee_id, manager_id, LEVEL
FROM employees
START WITH employee_id = 100
CONNECT BY PRIOR employee_id = manager_id
ORDER BY 4;
```

Rezultatul va fi următorul:

SQL | Fetched 50 rows in 0 seconds

	LAST_NAME	EMPLOYEE_ID	MANAGER_ID	LEVEL
1	King	100	(null)	1
2	De Haan	102	100	2
3	Raphaely	114	100	2
4	Weiss	120	100	2
5	Fripp	121	100	2
6	Kaufling	122	100	2
7	Vollman	123	100	2
8	Mourgos	124	100	2
9	Russell	145	100	2
10	Partners	146	100	2
11	Errazuriz	147	100	2
12	Cambrault	148	100	2
13	Zlotkey	149	100	2
14	Hartstein	201	100	2
15	Kochhar	101	100	2
16	Greenberg	108	101	3
17	Whalen	200	101	3
18	Mavris	203	101	3
19	Baer	204	101	3
20	Higgins	205	101	3
21	Hunold	103	102	3
22	Khoo	115	114	3

**Analiza top-n**

Pentru aflarea primelor n rezultate ale unei cereri, este utilă pseudocoloana **ROWNUM**. Aceasta returnează numărul de ordine al unei linii în rezultat.

**Exemplu:**

Dorim să aflăm cei mai vechi 3 angajați:

```
SELECT *
FROM (SELECT first_name, last_name, hire_date
      FROM employees
      ORDER BY hire_date)
WHERE ROWNUM < 4;
```

Rezultatul va fi următorul:

	FIRST_NAME	LAST_NAME	HIRE_DATE
1	Steven	King	17-JUN-87
2	Jennifer	Whalen	17-SEP-87
3	Neena	Kochhar	21-SEP-89

### Clauza WITH

- prin intermediul ei se poate da un nume unei subcereri, înainte ca aceasta să fie utilizată într-o interogare
- permite reutilizarea aceleiași subcereri într-o instrucțiune SELECT complexă. Acest lucru este util atunci când o cerere face referire de mai multe ori la aceeași subcerere, care conține operații join și funcții agregat.

### Exemplu:

Pentru fiecare angajat dorim să afișăm salariul mediu din departamentul său:

```
WITH salaries (department_id, SalMed) AS
    (SELECT department_id, round(avg(salary))
     FROM employees
     GROUP BY department_id)
SELECT employee_id, department_id, SalMed
FROM salaries JOIN employees USING (department_id);
```

Rezultatul va fi următorul:

	EMPLOYEE_ID	DEPARTMENT_ID	SALMED
1	100	90	19333
2	101	90	19333
3	102	90	19333
4	103	60	5760
5	104	60	5760
6	105	60	5760
7	106	60	5760
8	107	60	5760
9	108	100	8600
10	109	100	8600
11	110	100	8600
12	111	100	8600
13	112	100	8600
14	113	100	8600
15	114	30	4150
16	115	30	4150
17	116	30	4150
18	117	30	4150
19	118	30	4150
20	119	30	4150
21	120	50	3475

## 2. EXERCIIȚII

### [Cereri ierarhice]

1. Să se obțină ierarhia șef-subaltern, considerând ca rădăcină angajatul având codul 114.
2. Să se afișeze codul, numele, data angajării, salariul și managerul pentru:
  - a. subalternii directi ai lui De Haan;
  - b. ierarhia arborescenta de sub De Haan.

**Obs:** Traversarea precedentă este top-down. Faceți modificarea necesară obtinerii unei traversari bottom-up. Interpretați rezultatul.

3. Scrieti o cerere ierarhica pentru a afisa codul salariatului, codul managerului si numele salariatului, pentru angajatii care sunt cu 2 niveluri sub De Haan. Afișați, de asemenea, nivelul angajatului în ierarhie.
4. Pentru fiecare linie din tabelul EMPLOYEES, se va afisa o structura arborescenta in care va apărea angajatul, managerul său, managerul managerului etc. Coloanele afișate vor fi: codul angajatului, codul managerului, nivelul în ierarhie (LEVEL) și numele angajatului.
5. Să se afișeze ierarhia de sub angajatul având salariul maxim, reținând numai angajații al căror salariu este mai mare de 5000. Se vor afișa codul, numele, salariul, nivelul din ierarhie și codul managerului.

**Obs:** În clauza CONNECT BY, coloana employee\_id este evaluată pentru linia „părinte”, iar coloanele manager\_id și salary sunt evaluate pentru linia „copil”. Pentru a introduce, de exemplu, conditia ca salariul managerilor sa fie mai mare decât 15000, se scrie: PRIOR salary > 15000.

### [Analiza top-n]

6. Să se determine primii 10 cei mai bine plătiți angajați.
7. Să se determine cele mai prost plătite 3 job-uri, din punct de vedere al mediei salariilor.

### [Clauza WITH]

8. Utilizând clauza WITH, să se scrie o cerere care afișează numele departamentelor și valoarea totală a salariilor din cadrul acestora. Se vor considera departamentele a căror valoare totală a salariilor este mai mare decât media salariilor pe departamente.
9. Să se afișeze ierarhic codul, prenumele și numele (pe aceeași coloană), codul job-ului și data angajării, pornind de la subordonații direcți ai lui Steven King care au cea mai mare vechime. Rezultatul nu va conține angajații în anul 1970.