

LABORATOR 5: SUBCERERI INTEROGĂRI IERARHICE

Subcereri

O subcerere este o comandă SELECT încapsulată într-o clauză a altei instrucțiuni SQL, numită instrucțiune părinte. Utilizând subcereri, se pot construi interogări complexe pe baza unor instrucțiuni simple. Subcererile mai sunt numite instrucțiuni SELECT imbricate sau interioare.

Subcererile pot fi utilizate în următoarele situații:

- pentru a furniza valori care intervin în condițiile din clauzele WHERE, HAVING și START WITH ale instrucțiunii SELECT.
- pentru a defini un tabel asupra căruia va opera cererea externă (plasarea subcererii în clauza FROM sau în instrucțiunile INSERT, UPDATE, DELETE).

Subcereri necorelate:

```
SELECT lista_select  
FROM nume_tabel  
WHERE expresie operator (SELECT lista_select  
                        FROM nume_tabel);
```

- cererea internă este executată prima și determină o valoare (sau o mulțime de valori);
- cererea externă se execută o singură dată, utilizând valorile returnate de cererea internă.

Subcereri corelate:

```
SELECT nume_coloana_1[, nume_coloana_2 ...]  
FROM nume_tabel1 extern  
WHERE expresie operator  
      (SELECT nume_coloana_1 [, nume_coloana_2 ...]  
      FROM nume_tabel2  
      WHERE expresie_1 = extern.expresie_2);
```

- cererea externă determină o linie candidat;
- cererea internă este executată utilizând valoarea liniei candidat;
- valorile rezultate din cererea internă sunt utilizate pentru calificarea sau descalificarea liniei candidat;
- pașii precedenți se repetă până când nu mai există linii candidat.

Obs: operator poate fi:

- operator single-row (>, =, >=, <, <=, <>), care poate fi utilizat dacă subcererea returnează o singură linie;
- operator multiple-row operator (IN, ANY, ALL), poate fi folosit dacă subcererea returnează mai mult de o linie.

Exemplu: > ANY <=> mai mare decât maximul.

Operatorul NOT poate fi utilizat în combinație cu IN, ANY și ALL.

Exercițiul 1: Să se determine codul angajaților, codul job-urilor și numele celor al căror salariu este mai mare decât 3000 sau este egal cu media dintre salariul minim și cel maxim pentru job-ul respectiv.

Exercițiul 2: Să se afișeze numele, numele departamentului, salariul și job-ul tuturor angajaților al căror salariu și comision coincid cu salariul și comisionul unui angajat din Oxford.

Exercițiul 3: Scrieți o cerere pentru a afișa numele, codul departamentului și salariul angajaților al căror număr de departament și salariu coincid cu numărul departamentului și salariul unui angajat care câștigă comision.

Exercițiul 4: Folosind subcereri, să se afișeze numele și salariul angajaților conduși direct de președintele companiei (acesta este considerat angajatul care nu are manager).

Exercițiul 5: Scrieți o cerere pentru a afișa numele, numele departamentului și salariul angajaților care nu câștigă comision, dar al căror sef direct coincide cu seful unui angajat care câștigă comision.

Exercițiul 6: Să se afișeze numele și salariul angajaților al căror salariu este maxim.
Folosiți `>=ALL`

Exercițiul 7: Să se afle dacă există angajați care nu lucrează în departamentul 'Sales' și al căror salariu și comision coincid cu salariul și comisionul unui angajat din departamentul 'Sales'.

Exercițiul 8: Să se afișeze salariații care au același manager ca și angajatul având codul 140.

Exercițiul 9: Să se afișeze numele departamentelor din America.

Exercițiul 10: Să se obțină numele salariaților având cea mai mare vechime din departamentul în care lucrează.

Exercițiul 11: Să se afișeze numele, job-ul și salariul celor mai prost plătiți angajați din fiecare departament.

Exercițiul 12: Să se afișeze numele, prenumele și salariul angajaților care lucrează în același departament în care lucrează seful lor direct.

Exemplul 13: Să se obțină codurile și numele departamentelor în care nu lucrează nimeni.

Exercițiul 14: Să se obțină salariații care nu au subordonați (care nu sunt manageri). (Utilizați NOT IN)

Exercițiul 15: Să se afișeze codul, numele, prenumele angajaților care îndeplinesc una dintre condițiile următoare

- în departamentul în care lucrează în prezent a ocupat o funcție diferită de cea actuală
- funcția pe care o ocupă în prezent a ocupat-o și în alt departament, diferit de cel actual.

Subcereri în clauza FROM (view inline-uri)

O subcerere care apare în clauza FROM se mai numește view - in - line.

Exercițiul 16: Să se afișeze codul și numele angajatului, numele departamentului și suma salariilor pe departamente.

Exercițiul 17: Să se afișeze pentru fiecare angajat numele, salariul și salariul mediu al celor care au același șef direct cu el.

Exercițiul 18: Să se afișeze numele și jobul primilor angajați din companie, în ordinea descrescătoare a salariului

Exercițiul 19: Să se obțină media salariului noilor veniti (ultimii 10 angajati).

Exercițiul 20: Să se obțină diferența de vechime (numar de luni) dintre al 10 ultim angajat și primul angajat din companie.

Interogări ierarhice

Interogările ierarhice permit regăsirea datelor pe baza unei relații ierarhice care există între liniile tabelului. Dacă există o relație ierarhică între liniile unui tabel, un proces de parcurgere a unui arbore (*tree walking*) permite construirea ierarhiei. O cerere ierarhică este o metodă de raportare, în ordine a ramurilor arborelui.

Clauzele instrucțiunii *SELECT* care intervin în formularea unei cereri ierarhice sunt *START WITH* și *CONNECT BY*.

START WITH specifică o condiție care identifică liniile ce urmează să fie considerate ca rădăcini ale cererii. Dacă se omite această clauză, sistemul *Oracle* utilizează toate liniile din tabel drept linii rădăcină.

Clauza **CONNECT BY** specifică o condiție care identifică relația dintre liniile „părinte” și „copil” ale ierarhiei.

Operatorul **PRIOR** face referință la linia părinte și poate fi plasat în fața oricărui membru al condiției specificate de clauza *CONNECT BY*. Plasarea acestui operator determină direcția interogării, dinspre „părinte” spre „copil” (*top-down*) sau invers (*bottom-up*).

Traversarea *topdown*, 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 cheie_copil = PRIOR cheie_parinte;

Liniile „părinte” ale interogării sunt identificate de 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”.

Pseudocoloana **LEVEL** determină lungimea drumului de la rădăcină la un nod.

Ordinea de parcurgere a arborelui, în adâncime, este ilustrată mai jos:

Clauza *ORDER BY* va înlocui parcurgerea în adâncime a arborelui cu o afișare a liniilor în ordinea specificată. Pentru a ordona doar fii unui anumit nod după expresiile de la *ORDER BY* se va utiliza **ORDER SIBLINGS BY**.

SYS_CONNECT_BY_PATH returnează calea de la rădăcina la un nod. Din fiecare linie nod se va alege coloana specificată de către primul parametru. Valorile coloanelor selectate sunt separate de caracterul specificat de al doilea parametru.

Exercițiul 21: Să se obțină ierarhia bazată pe relația angajat – manager considerând angajatul cu codul 100 ca rădăcină (clauza *START WITH*). Să se ordoneze după lungimea drumului de la rădăcină la fiecare nod (pseudocoloana *LEVEL*).

Exercițiul 22: Scrieți o cerere ierarhică pentru a afișa codul salariatului, codul managerului și numele salariatului, pentru angajații care sunt cu 2 niveluri sub De Haan. Afișați, de asemenea, nivelul angajatului în ierarhie.

Exercițiul 23: Să se afișeze codul, numele și salariul angajatului însoțit de codul managerului său, pentru angajații al căror salariu este mai mic decât 15000. Se va considera ca rădăcină salariatul având codul 206.

Exemplul 24: Să se afișeze pentru fiecare angajat toți superiorii săi pornind de la managerul general. Vârful ierarhiei este angajatul cu codul 101. Angajații subordonați aceluiași manager vor fi ordonați după salariu. Se vor afișa doar angajații cu salariul mai mare de 3000.

Exercițiul 25: Pentru fiecare linie din tabelul EMPLOYEES, se va afișa o structură arborescentă în care va apărea angajatul, managerul său, managerul managerului etc. (Fiecare linie va reprezenta rădăcina unui arbore). Să se indenteze rezultatul pentru a se sugera nivelul pe care se află în arbore fiecare linie.

Exercițiul 26: Să se afișeze codul, numele și salariul angajatului însoțit de codul managerului său, începând cu angajatul al cărui cod este maxim.

Exercițiul 27: Să se obțină ierarhia bazată pe relația angajat – manager considerând angajatul care nu are manager rădăcină.