

## LABORATOR 5: SUBCERERI OPERATORI PE MULȚIMI

### 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 șef direct coincide cu șeful 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 șeful 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 - în - line.

**Exercițiul 16:** Să se afișeze numele și jobul primilor angajați din companie, în ordinea descrescătoare a salariului

**Exercițiul 17:** Să se obțină media salariului noilor veniti (ultimii 10 angajati).

**Exercițiul 18:** Să se obțină diferența de vechime (numar de luni) dintre al 10 ultim angajat si primul angajat din companie.

## OPERATORI PE MULȚIMI

Operatorii pe mulțimi combină rezultatele obținute din două sau mai multe interogări. Cererile care conțin operatori pe mulțimi se numesc cereri compuse. Există patru operatori pe mulțimi:

UNION, UNION ALL, INTERSECT și MINUS.

Toți operatorii pe mulțimi au aceeași precedență. Dacă o instrucțiune SQL conține mai mulți operatori pe mulțimi, server-ul Oracle evaluează cererea de la stânga la dreapta (sau de sus în jos). Pentru a schimba această ordine de evaluare, se pot utiliza paranteze. În instrucțiunile SELECT asupra cărora se aplică operatori pe mulțimi, coloanele selectate trebuie să corespundă ca număr și tip de date. Nu este necesar ca numele coloanelor să fie identice. Numele coloanelor din rezultat sunt determinate de numele care apar în clauza SELECT a primei cereri.

Clauza ORDER BY poate apărea numai o singură dată într-o cerere compusă (la sfârșitul cererii). În mod implicit, pentru toți operatorii cu excepția lui UNION ALL, rezultatul este ordonat crescător după valorile primei coloane din clauza SELECT.

Operatorul **UNION** returnează toate liniile selectate de două cereri, eliminând duplicatele. Acest operator nu ignoră valorile null și are precedență mai mică decât operatorul IN.

Operatorul **UNION ALL** returnează toate liniile selectate de două cereri, fără a elimina duplicatele. Precizările făcute asupra operatorului UNION sunt valabile și în cazul operatorului UNION ALL. În cererile asupra cărora se aplică UNION ALL nu poate fi utilizat cuvântul cheie DISTINCT. Toți ceilalți operatori pe mulțimi elimină liniile duplicate.

Operatorul **INTERSECT** returnează toate liniile comune cererilor asupra cărora se aplică. Acest operator nu ignoră valorile null.

Operatorul **MINUS** determină liniile returnate de prima cerere care nu apar în rezultatul celei de-a doua cereri.

**Exemplul 19.** Să se obțină codurile departamentelor al căror nume conține sirul "re" sau în care lucrează angajați având codul job-ului "SA\_REP".

```
SELECT department_id "Cod departament"
FROM employees
WHERE UPPER(job_id) = 'SA_REP'
UNION
SELECT department_id
FROM departments
WHERE LOWER(department_name) LIKE '%re%';
```

**Exemplul 20.** Să se obțină codurile departamentelor în care nu lucrează nimeni (nu este introdus nici un salariat în tabelul employees).

```
SELECT department_id "Cod departament"
FROM departments
MINUS
SELECT DISTINCT department_id
FROM employees;
```

**Exemplul 21.** Se cer codurile departamentelor al căror nume conține sirul "re" și în care lucrează angajați având codul job-ului "HR\_REP".

```
SELECT department_id "Cod departament"
FROM employees
WHERE UPPER(job_id) = 'HR_REP'
INTERSECT
SELECT department_id
FROM departments
WHERE LOWER(department_name) LIKE '%re%'
```

**Exemplul 22.** Utilizând operatorul UNION, să se listeze codul salariaților, numele angajaților, codul și numele departamentelor. Să se ordoneze rezultatul după codul și numele departamentului.

```
SELECT employee_id, last_name, department_id, TO_CHAR(NULL) nume
FROM employees
UNION
SELECT TO_NUMBER(NULL), null, department_id, department_name
FROM departments
ORDER BY 3, nume;
```

**Exercițiul 23.** Să se așeze media venitului tuturor angajaților și media venitului salariaților angajați în anul 2000. Să se rotunjească mediile la două zecimale. Cele două linii rezultat vor include textele 'medie' respectiv 'medie 2000'.

Observați modul în care este ordonat rezultatul. Să se atribue unice coloane rezultat un titlu potrivit.

**Exercițiul 24.** Să se așeze numele departamentelor și numele angajaților. Se vor afișa și departamentele în care nu lucrează nimeni și angajații care nu lucrează în nici un departament (full outer join). Se va utiliza UNION.

**Exercițiul 25.** Folosind INTERSECT să se afișeze codul departamentului și numele departamentului pentru departamentele din orasul Seattle, coduse de un angajat al cărui salariu este mai mare decât 7000.

**Exercițiul 26.** Să se afișeze denumirile joburilor pe care nu le-a avut șeful departamentului „Administration”.

**Exercițiul 27.** Să se afișeze codurile și numele șefilor de departament care au mai avut cel puțin un job anterior.

**Exercițiul 28.** Să se obțină o listă cu istoricul funcțiilor avute de angajați (se va utiliza tabelul `job_history`). Tabelul va conține următoarele coloane: codul angajatului, numele angajatului, titlul jobului avut, numele departamentului în care a lucrat în timpul cât a deținut funcția respectivă, perioada în care a ocupat poziția respectivă exprimată în număr întreg de luni. Coloana "perioada" va conține null, în cazul funcțiilor deținute în prezent de angajați. Ordonăți rezultatul după numele angajatului.