

Subcereri sincronizate. Clauza *WITH*. Analiza *top-n*.

I. [Obiective]

- Lucrul cu subcereri sincronizate (corelate).
- Definirea de blocuri de cerere înaintea instrucțiunii *SELECT* propriu-zise, cu scopul simplificării acestora.
- Aflarea primelor *n* înregistrări ordonate după un anumit criteriu.

II. [Subcereri corelate (sincronizate)]

O subcerere (cerere imbricată sau încuibărită) corelată poate fi regăsită în clauza *WHERE*, având forma următoare:

```
SELECT nume_coloană_1[, nume_coloană_2 ...]
FROM   nume_tabel_1 extern
WHERE  expresie operator
      (SELECT nume_coloană_1 [, nume_coloană_2 ...]
       FROM   nume_tabel_2
       WHERE  expresie_2 = extern.expresie_1);
```

Modul de execuție :

- 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.

Observație: *operator* poate fi:

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

Observație: O subcerere (corelată sau necorelată) poate apărea în clauzele:

- *SELECT* (vezi mai jos)
- *FROM* (vezi laboratorul 7)
- *WHERE*
- *HAVING* (vezi laboratorul 6)

Operatorul *EXISTS*

- În instrucțiunile *SELECT* imbricate, este permisă utilizarea oricărui operator logic.
- Pentru a testa dacă valoarea recuperată de cererea externă există în mulțimea valorilor regăsite de cererea internă corelată, se poate utiliza operatorul *EXISTS*. Dacă subcererea returnează cel puțin o linie, operatorul returnează valoarea *TRUE*. În caz contrar, va fi returnată valoarea *FALSE*.
- Operatorul *EXISTS* asigură că nu mai este continuată căutarea în cererea internă după ce aceasta regăsește o linie.

III. [Exerciții – subcereri sincronizate]

1. a) Să se afișeze informații despre angajații al căror salariu depășește valoarea medie a salariilor colegilor săi de departament.

```
SELECT last_name, salary, department_id
FROM   employees e
WHERE  salary > (SELECT AVG(salary)
                  FROM   employees
                  WHERE  department_id = e.department_id);
```

b) Analog cu cererea precedentă, afișându-se și numele departamentului, media salariilor acestuia și numărul de angajați. Se cer 2 soluții (cu subcerere nesincronizată în clauza *FROM* și cu subcerere sincronizată în clauza *SELECT*).

2. Să se afișeze numele și salariul angajaților al căror salariu este mai mare decât salariile medii din toate departamentele. Se cer 2 variante de rezolvare: cu operatorul *ALL* și cu funcția *MAX*.
3. Să se afișeze numele și salariul celor mai puțin plătiți angajați din fiecare departament (3 soluții: cu și fără sincronizare, subcerere în clauza *FROM*).
4. Pentru fiecare departament, să se obțină denumirea acestuia și numele salariatului având cea mai mare vechime din departament. Să se ordoneze rezultatul după numele departamentului.
5. Să se obțină numele salariaților care lucrează într-un departament în care există cel puțin un angajat cu salariul egal cu salariul maxim din departamentul 30.

```
SELECT last_name, salary
FROM   employees e
WHERE  EXISTS (SELECT 1
                FROM   employees
                WHERE  e.department_id = department_id
                AND    salary = (SELECT MAX(salary)
                                FROM   employees
```

```
WHERE department_id =30));
```

Observație: Deoarece nu este necesar ca instrucțiunea *SELECT* interioară (subcererea) să returneze o anumită valoare, se poate selecta o constantă ('x', 1 etc.). De altfel, din punct de vedere al performanței, selectarea unei constante este mai eficientă decât selectarea unei coloane, nemaifiind necesară accesarea datei respective.

6. Să se obțină numele primilor 3 angajați având cele mai mari salarii. Rezultatul se va afișa în ordine crescătoare a salariilor.

Soluția 1: subcerere sincronizată

Soluția 2: vezi analiza top-n (mai jos)

7. Să se afișeze codul, numele și prenumele angajaților care au cel puțin doi subalterni.

8. Să se determine locațiile în care se află cel puțin un departament.

Observație: Ca alternativă a lui *EXISTS*, poate fi utilizat operatorul *IN*. Scrieți și această variantă de rezolvare.

9. Să se determine departamentele în care nu există nici un angajat.

Observație: Se va utiliza *NOT EXISTS*. Acest exemplu poate fi rezolvat și printr-o subcerere necorelată, utilizând operatorul *NOT IN* (vezi laboratorul 3). Atenție la valorile *NULL*! (fie puneți condiția *IS NOT NULL* în subcerere, fie utilizați funcția *NVL*). Scrieți și această variantă de rezolvare.

IV. [Clauza *WITH*]

- Cu ajutorul clauzei *WITH* se poate defini un bloc de cerere înainte ca acesta să fie utilizat într-o interogare.
 - Clauza permite reutilizarea aceluiași bloc de cerere într-o instrucțiune *SELECT* complexă. Acest lucru este util atunci când o cerere face referință de mai multe ori la același bloc de cerere, care conține operații *join* și funcții agregat.
10. 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 valorilor totale ale salariilor tuturor angajaților.

```
WITH val_dep AS (
    select department_name, sum(salary) as total
    from departments d, employees e
    where e.department_id =d.department_id
    group by department_name),
val_medie AS (select sum(total)/count(*) as medie
    from val_dep )
```

```
SELECT  *
FROM    val_dep
WHERE    total > (SELECT  medie
                  FROM    val_medie)
ORDER BY department_name;
```

11. Să se afișeze codul, prenumele și numele (pe aceeași coloană), codul *job*-ului și data angajării, ale subalternilor subordonaților direcți ai lui Steven King care au cea mai mare vechime.

V. [Analiza *top-n*]

Pentru aflarea primelor *n* rezultate ale unei cereri, este utilă funcția *ROWNUM*. Aceasta returnează numărul de ordine al unei linii în rezultat. Condiția ce utilizează această funcție trebuie aplicată asupra unei mulțimi ordonate de înregistrări. Cum obținem acea mulțime?

12. Să se determine primii 10 cei mai bine plătiți angajați.

13. Să se determine cele mai slab plătite 3 *job*-uri, din punct de vedere al mediei salariilor acestora.

Exercițiu suplimentar

14. Să se afișeze:

- suma salariilor, pentru *job*-urile al căror cod începe cu litera S;
- media generală a salariilor, pentru *job*-ul având salariul maxim;
- salariul minim pe *job*, pentru fiecare dintre celelalte *job*-uri.