

**Subcereri necorelate. Subcereri corelate.**

**[Scurtă recapitulare]**

- I. Să se afișeze salariul maxim al angajaților doar dacă acesta este mai mare decât 15000.
- II. Să se modifice exercițiul anterior astfel încât să se afișeze o listă cu id-ul și numele complet al angajaților/angajatului, denumirea departamentului și orașului în care lucrează aceștia pentru angajații care au salariul egal cu salariul maxim din firmă, doar dacă acest salariu maxim este mai mare decât 15000. Dați alias-uri corespunzătoare tuturor coloanelor.
- III. Să se afișeze lista salariaților (nume complet și data angajării) din departamentele în care lucrează mai mult de 8 salariați și care au fost angajați după ultimul salariat angajat în departamentul 30. Dați alias-uri corespunzătoare tuturor coloanelor.

**I. [Subcereri în clauza FROM]**

Subcererile pot apărea în clauza SELECT, WHERE sau FROM a unei cereri. O subcerere care apare în clauza FROM se mai numește **view in-line**.

1. Să se afișeze codul, numele departamentului și suma salariilor pe departamente.

```
SELECT d.department_id, department_name, a.suma
FROM departments d, (SELECT department_id ,SUM(salary) suma
                     FROM employees
                     GROUP BY department_id) a
WHERE d.department_id =a.department_id;
```

2. Să se afișeze numele, salariul, codul departamentului și salariul mediu din departamentul respectiv.

```
SELECT e.last_name, e.salary, d.department_id, d.medie
FROM employees e, (SELECT department_id, AVG(salary) medie
                  FROM employees
                  GROUP BY department_id) d
WHERE e.department_id = d.department_id ;
```

3. Modificați cererea anterioară, pentru a determina și listarea numărului de angajați din departamente.
4. Pentru fiecare departament, să se afișeze numele acestuia, numele și salariul celor mai prost plătiți angajați din cadrul său.

## II. [Subcereri corelate (sincronizate)]

O subcerere (cerere imbricată sau încuibărită) corelată poate avea 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_1 = extern.expresie_2);
```

Modul de execuție este următorul :

- 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:

- *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 de o linie.

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

- SELECT
- FROM
- WHERE
- HAVING (vezi laboratorul 3-4-5)

### 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.

### Exerciții:

5. 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 și media salariilor acestuia și numărul de angajați.

**Soluția 1** (subcerere necorelată în clauza FROM):

```
SELECT last_name, salary, e.department_id, department_name, sal_med, nr_sal
FROM   employees e, departments d,
        (SELECT department_id, AVG(salary) sal_med, COUNT(*) nr_sal
         FROM   employees
```

```

GROUP BY department_id) sm
WHERE e.department_id = d.department_id
AND d.department_id = sm.department_id
AND salary > (SELECT AVG(salary)
FROM employees
WHERE department_id = e.department_id);

```

**Soluția 2** (subcerere corelată în clauza SELECT):

```

SELECT last_name, salary, e. department_id, department_name,
(SELECT AVG(salary)
FROM employees
WHERE department_id = e. department_id) "salariu mediu",
(SELECT COUNT(*)
FROM employees
WHERE department_id = e. department_id) "Nr angajati"
FROM employees e, departments d
WHERE e.department_id = d.department_id
AND salary > (SELECT AVG(salary)
FROM employees
WHERE department_id = e.department_id);

```

6. 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 sau cu funcția MAX.
7. Sa se afiseze numele si salariul celor mai prost platiti angajati din fiecare departament.

**Soluția 1** (cu sincronizare):

```

SELECT last_name, salary, department_id
FROM employees e
WHERE salary = (SELECT MIN(salary)
FROM employees
WHERE department_id = e.department_id);

```

**Soluția 2** (fără sincronizare):

```

SELECT last_name, salary, department_id
FROM employees
WHERE (department_id, salary) IN (SELECT department_id, MIN(salary)
FROM employees
GROUP BY department_id);

```

**Soluția 3:** Subcerere în clauza FROM (individual)

8. Pentru fiecare departament, să se obtina numele salariatului avand cea mai mare vechime din departament. Să se ordoneze rezultatul după numele departamentului.
9. Sa se obtina numele salariatilor care lucreaza intr-un departament in care exista cel putin 1 angajat cu salariul peste 12500\$.

```

SELECT e.last_name, e.salary, e.department_id
FROM employees e
WHERE EXISTS (SELECT e2.employee_id
FROM employees e2
WHERE e2.department_id = e.department_id
AND e2.salary > 12500);

```

Alta varianta: cu self-join

```

SELECT DISTINCT e.last_name, e.salary, e.department_id
FROM employees e, employees e2

```

WHERE e2.department\_id = e.department\_id  
and e2.salary > 12500;

10. Sa se obtina numele salariatilor care lucreaza intr-un departament in care exista cel putin 1 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));
```

**Obs:** Deoarece nu este necesar ca instrucțiunea *SELECT* interioară să returneze o anumită valoare, se poate selecta o constantă ('x', '', 1 etc.). De altfel, din punct de vedere al performanței, selectarea unei constante asigură mai multă rapiditate decât selectarea unei coloane.

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

```
SELECT department_id, department_name
FROM departments d
WHERE NOT EXISTS (SELECT 'x'
                  FROM employees
                  WHERE department_id = d.department_id);
```

**Obs:** Acest exemplu poate fi rezolvat și printr-o subcerere necorelată, utilizând operatorul *NOT IN* (vezi și laboratorul 3-4-5). Atenție la valorile NULL! (fie puneți condiția *IS NOT NULL* în subcerere, fie utilizați funcția *NVL*). Scrieți și această variantă de rezolvare.

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

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

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

14. Titlurile joburilor care nu au fost niciodata ocupate in firma (nici acum, nici in trecut)