

# LABORATORUL

## SUBINTEROGARI

Obiectivele acestui capitol sunt, să învățăm cum:

- să construim o subinterogare;
- să folosim o subinterogare într-o expresie dintr-o instrucțiune SQL;
- să construim și să utilizăm subinterogări legate;
- să construim o subinterogare folosind următoarele cuvinte cheie într-o clauză WHERE a unei instrucțiuni SELECT: ALL, ANY, IN, EXISTS.

O instrucțiune SELECT imbricată în clauza WHERE a unei instrucțiuni SELECT ( INSERT, DELETE sau UPDATE ) se numește **subinterogare** (subquery). Fiecare interogare conține în mod obligatoriu o clauză SELECT și o clauză FROM. Fiecare subinterogare trebuie să fie închisă între paranteze pentru ca serverul BD să o execute mai întâi. Asta înseamnă că subinterogarea este o interogare ale cărei rezultate sunt transmise altei interogări. Subinterogarea este o modalitate de a face legături între două sau mai multe interogări. Subinterogările pot fi corelate sau necorelate. O subinterogare (sau o instrucțiune SELECT interior ) este corelată când valoarea produsă de ea depinde de o valoare produsă de o instrucțiune SELECT exterioară care o conține. Orice alt tip de subinterogare este necorelată.

O caracteristică importantă a unei subinterogări corelate este că, deoarece ea depinde de o valoare a unui SELECT exterior, ea trebuie să fie executată în mod repetat, o dată pentru fiecare valoare produsă de SELECT-ul exterior. O subinterogare necorelată este executată numai o dată.

Putem construi o instrucțiune SELECT cu o subinterogare care să înlocuie două instrucțiuni SELECT separate.

Folosind subinterogarea putem găsi informații din unul sau mai multe tabele fără a face unirea lor.

Subinterogările din instrucțiunile SELECT ne permit să realizăm acțiunile următoare:

- să comparăm o expresie cu rezultatul unei alte instrucțiuni SELECT;
  - să determinăm când rezultatul unei alte instrucțiuni SELECT include o expresie;
  - să determinăm când o altă instrucțiune SELECT selectează anumite linii.
- Clauza opțională WHERE dintr-o subinterogare este adesea utilizată ca să limiteze condiția de căutare. O subinterogare poate returna nimic, o singură valoare sau o mulțime de valori;
- dacă subîntrebarea nu returnează nici o valoare, ea este echivalentă cu valoarea null atunci întrebarea nu returnează nici o linie;
  - dacă subinterogarea returnează o singură valoare care este dată fie de o expresie agregat fie exact de o linie compusă dintr-o coloană atunci, ea este echivalentă cu un singur număr sau o valoare de caracter;
  - dacă o subîntrebare returnează o listă sau o mulțime de valori, valorile

reprezintă fie o linie, fie o coloană;

### **Construcția unei subinterogări**

Fie tabelele **r** și **s** care conțin coloanele **A** respectiv **B** care au același domeniu. Structura formală a unei subinterogări este:

```
SELECT *  
FROM r  
WHERE r.A=(SELECT B  
            FROM s  
            WHERE B='valoare')
```

Se execută mai întâi subinterogarea dintre paranteze și se selectează toate liniile din tabelul **s** care sunt egale cu **A** pentru liniile din **r**.

**Exemplul 1.** Subinterogarea determină toți clienții care sunt din același oraș cu **Albu**:

```
SELECT *  
FROM clienți  
WHERE oraș=  
      (SELECT oraș  
       FROM clienți  
       WHERE NUME='Albu')
```

**Exemplul 2.** Determinați toți clienții din orașul celui cu comanda 1015:

```
SELECT *  
FROM clienți  
WHERE oraș=  
      (SELECT oraș  
       FROM companie  
       WHERE NR_COM=1015)
```

**Exemplul 3.** Următoarea subinterogare furnizează fabricile care au termenul de livrare mai mic decât termenul de livrare a lui **DAEWO**.

```
SELECT *  
FROM fabrici  
WHERE termen_liv < (SELECT termen_liv  
                    FROM fabrici  
                    WHERE ID_FABRICA="DAEWO")
```

**Exemplul 4.** Utilizarea cuvântului cheie ALL înaintea unei subinterogare determină când comparația este adevărată pentru orice valoare returnată. Determinați comenzile și articolele și fabricile care conțin articole ale căror prețuri sunt mai mici decât prețul oricărui articol din comanda 1023.

```
SELECT NR_COM, CD_ART, ID_FABR, PREȚ
FROM articole
WHERE PREȚ < ALL (SELECT PREȚ FROM articole
                  WHERE NR_COM=1023)
```

**Exemplu 5.** Utilizarea cuvântului cheie înaintea unei subinterogări determină când comparația este adevărată pentru cel puțin o valoare returnată.

```
SELECT DISTINCT NR_COMANDA
FROM articole
WHERE PREȚ_TOTAL > ANY(SELECT PREȚ_TOTAL
                       FROM articole
                       WHERE NR_COMANDA=1005)
```

**Exemplu 6.** Fie tabelele:

```
articole (ID_ART, DENUMIRE, ID_FABR, PREȚ_UNIT)
fabrici (ID_FABR, NUME_FABR, ORAS, STRADA, NR)
```

Determinați articolele care sunt făcute de o fabrică din București.

```
SELECT ID_ART, DENUMIRE
FROM articole
WHERE ID_FABR IN (SELECT ID_FABR
                  FROM fabrici
                  WHERE ORAS = 'București')
```

**Exemplu 7.** Fie tabelele:

```
clienți (ID_CL, NUME_CL, PREȚ_CL, ORAS, STR, NR)
comenzi (NR_COMANDA, DATA_COM, ID_CL, DATA, LIVRĂRI)
```

Determinați clienții care nu au nici o comandă:

```
SELECT NUME_CL, PREȚ_CL
FROM clienți
WHERE ID_CL NOT IN (SELECT ID_CL
                   FROM comenzi)
```

**Exemplu 8.** Fie tabelele din exemplu 6. Determinați articolele care au prețul de 10 ori prețul unitar minim.

```

SELECT ID_ART, DENUMIRE, ID_FABR
FROM articole
WHERE PRET_UNIT > (SELECT 10*MIN(PRET_UNIT)
                   FROM articole)

```

**Exemplu 9.** Aceleași tabele de la exemplu 6

```

SELECT ID_ART, DENUMIRE, ID_FABR
FROM articole
WHERE PRET_UNIT < (SELECT AVG(PRET_UNIT)
                   FROM articole)

```

## Exemple de subinterogări corelate

**Exemplu 10.** Fie tabelul:

```

comenzi(NR_COM, DATA_COMENZII, DATA_LIVRARI, ID_CL)

```

Întrebarea este un exemplu de întrebare corectă ce returnează o listă a primelor 10 date de livrare din tabelul de comenzi.

```

SELECT DATA_LIVRĂRII
FROM comenzi a
WHERE 10>(SELECT COUNT(*)
          FROM comenzi b
          WHERE b.DATA_LIVRARI>a.DATA_LIVRARI)
AND DATA_LIVRARI IS NOT NULL
ORDER BY DATA_LIVRARI

```

**Exemplu 11.** Fie tabelul de articole din exemplul din exemplu 4 și tabelul de comenzi din exemplu 8. Determinați comenzile care au mai mult de 3 articole.

```

SELECT NR_COM
FROM comenzi c
WHERE 3<(SELECT COUNT(*)
        FROM articole a
        WHERE c.NR_COM = a.NR_COM)

```

Exercițiu. Determinați produsele care sunt fabricate de cel puțin 2 fabrici.

**Exemplu 12.** Fie tabelele de comenzi și de articole din exemplul 7. Determinați toate comenzile de CIELO

```

SELECT *
FROM comenzi c
WHERE "CIELO" = (SELECT DENUMIRE
                  FROM articole a
                  WHERE c.ID_AR=a.ID_ART)

```

EXISTS este o funcție predicat care acceptată ca paramentru unic a subinterogării. Dacă o interogarea returnează o mulțime de rezultate atunci EXISTS returnează TRUE în caz contrar FALSE:

**Exemplu 13.** Fie tabelele:

```

titlu (ID_TITLU, DENUMIRE_T, SAL_ORA)
salariati(ID_SALARIAT, NUME, PREN, ID_TITLU, ADR)

```

Determinați titlurile pentru care există salariați:

```

SELECT ID_TITLU, DENUMIRE_T
FROM titlu t
WHERE EXISTS (SELECT *
              FROM salariati s
              WHERE t.ID_TITLU=s.ID_TITLU )

```

**Exemplu 14.** Determinați titlurile pentru care nu există salariați.

```

SELECT ID_TITLU, DENUMIRE_T
FROM titlu t
WHERE NOT EXISTS (SELECT *
                  FROM salariati s
                  WHERE t.ID_TITLU=s.ID_TITLU )

```

**Exemplu 15.** Determinați fabricile și produsele pentru care nu a existat nici o comanda.

```

SELECT ID_ART, ID_FABR
FROM produse p
WHERE NOT EXISTS (SELECT ID_ART, ID_FABR
                  FROM articole a
                  WHERE p.ID_ART = a.ID_ART
                  AND p.ID_FABR=a.ID.FABR )

```

**Exemplu 16.** Aceasta este echivalență cu cea din exemplu 4.

```

SELECT NR_COM
FROM articole
WHERE PRET_TOTAL>(SELECT max(PRET_TOTAL))
FROM ARTICOLE
WHERE NRCOM=1023)

```

**Exemplu 17.** Această subinterogare determină comenzile care au valoarea mai mică decât valoarea medie:

```

SELECT c.NR_COM
FROM comenzi c, articole a
WHERE c.NR_COM = s.NR_COM
AND PRET_TOTAL >
(SELECT AVG(PRET_TOTAL))
FROM articole
WHERE c.NR_COM = a.NR_COM)

```

**Exemplu 18.** Această instrucțiune SELECT returnează numărul tuturor comenzilor care conțin un articol al cărui PRET\_TOTAL este mai mare decât pretul total al cel puțin unui articol din comanda cu numărul 1023. Ea poate fi scrisă cu ajutorul cuvântului cheie ANY sau cu ajutorul funcției MIN.

```

SELECT DISTINCT NR_COM
FROM articole
WHERE PRET_TOTAL > ANY (SELECT PRET_TOTAL
                        FROM articole
                        WHERE NR_COM=1023)

```

**Exemplu 19.** Aceasta este echivalentă cu 18.

```

SELECT DISTINCT NR_COM
FROM articole
WHERE PRET_TOTAL > (SELECT MIN(PRET_TOTAL))
                   FROM articole
                   WHERE NR_COM=1023)

```

## Utilizarea unei condiții de unire în cauza where

Când se unesc 2 tabele trebuie creată o legătură în clauza WHERE prin cel puțin a coloană dintr-un tabel și o coloană din alt tabel.

Efectul este un tabel compus care e temporar, în care fiecare pereche de linii satisface legătura, condițiile de unire formează o singură linie. Putem crea autouniri (unirea unui tabel cu el însuși) unirea a 2 tabele sau a unirea a mai multor tabele.

**Exemplu 20.** Exemplu de unire a doua tabele:

```

SELECT NUME, PREN, NR_COM
FROM clienți, comenzi
WHERE clienti.NR_CL=comenzi.NR_CL

```

**Exemplu 21.** Autounirea (SELF JOIN) crează o listă de articole din atributul stoc care sunt produse de 3 fabrici:

```

SELECT a.ID_FABR, b.ID_FABR, c.ID_FABR, c.ID_ART
FROM stoc a, stoc b, stoc c
WHERE a.ID_ART= b.ID_ART
      AND a.ID_ART= c.ID_ART

```

```

AND a. ID_FABR< b.ID_FABR
AND b. ID_FABR< c.ID_FABR)
ORDER BY ID_ART

```

**Exemplu 22.** Comenzile care au fost comandate în aceeași zi de 2 ori se consideră duplicate și se elimină una.

```

SELECT *.NUME_CL, *.DATA_COMENZII
FROM emitere-comenzi x
WHERE 1<(SELECT COUNT(x)
FROM emitere-comenzi y
WHERE x. NUME_CL =y. NUME_CL)

```

## Exercitii

Determinați rezultatul fiecărei secvențe de instrucțiuni

```

SELECT *
FROM Catalog
WHERE NrLeg=(SELECT NrLeg
FROM Student
WHERE Nume = 'Popescu' AND Grupa='222')

```

```

SELECT *
FROM Catalog
WHERE NrLeg>(SELECT NrLeg
FROM Student
WHERE Nume = 'Popescu' AND Grupa='222')

```

```

SELECT S.NrLeg, Nume +' '+Prenume Student,Grupa, cod_disciplina, Nota
FROM Student as S, Catalog as N
WHERE S.NrLeg=C.NrLeg
AND S.NrLeg=(SELECT N1.NrLeg
FROM Catalog N1, Discipline D
WHERE N1.cod_disciplina=D.cod_disciplina AND
D.cod_disciplina>3)

```

Eroare subinterogarea returnează mai mult de 1 înregistrare

```

SELECT S.NrLeg, Nume +' '+Prenume Student,Grupa, cod_disciplina, Nota
FROM Student as S, Catalog as C
WHERE S.NrLeg=C.NrLeg
AND S.NrLeg IN (SELECT N1.NrLeg
FROM Catalog N1, Discipline D
WHERE N1.cod_disciplina=D.cod_disciplina AND
D.cod_disciplina>3)

```

```

SELECT S.NrLeg, Nume +' '+Prenume Student,Grupa, cod_disciplina, Nota
FROM Student as S, Catalog as C
WHERE S.NrLeg=C.NrLeg
      AND S.NrLeg = ANY (SELECT N1.NrLeg
                        FROM Catalog N1, Discipline D
                        WHERE N1.cod_disciplina=D.cod_disciplina
                        AND
                        D.cod_disciplina>3)

```

```

SELECT S.NrLeg, Nume +' '+Prenume Student,Grupa, cod_disciplina, Nota
FROM Student as S, Catalog as N
WHERE S.NrLeg=C.NrLeg
AND S.NrLeg > any (SELECT N1.NrLeg
                  FROM Catalog N1, Discipline D
                  WHERE N1.cod_disciplina=D.cod_disciplina AND
                  D.cod_disciplina>3)

```

```

SELECT S.NrLeg, Nume +' '+Prenume Student,Grupa, cod_disciplina, Nota
FROM Student as S, Catalog as C
WHERE S.NrLeg=C.NrLeg
      AND EXISTS (SELECT N1.NrLeg
                  FROM Catalog N1, Discipline D
                  WHERE N1.cod_disciplina=D.cod_disciplina AND
                  D.cod_disciplina>3)

```

```

SELECT S.NrLeg, Nume +' '+Prenume Student,Grupa, cod_disciplina, Nota
FROM Student as S, Catalog as C
WHERE S.NrLeg=C.NrLeg
      AND EXISTS (SELECT N1.NrLeg FROM
                  Catalog N1, Discipline D
                  WHERE N1.cod_disciplina=D.cod_disciplina AND
                  D.cod_disciplina>8)

```

```

SELECT S.NrLeg, Nume +' '+Prenume Student,Grupa, cod_disciplina, Nota
FROM Student as S, Catalog as C
WHERE S.NrLeg=C.NrLeg
      AND S.NrLeg <> ALL (SELECT N1.NrLeg
                        FROM Catalog N1, Discipline D
                        WHERE N1.cod_disciplina=D.cod_disciplina
                        AND D.cod_disciplina>3)

```

```

SELECT S.NrLeg, Nume +' '+Prenume Student,Grupa, cod_disciplina, Nota
FROM Student as S, Catalog as C
WHERE S.NrLeg=C.NrLeg
      AND S.NrLeg >= all (SELECT N1.NrLeg
                        FROM Catalog N1, Discipline D
                        WHERE N1.cod_disciplina=D.cod_disciplina AND
                        D.cod_disciplina>3)

```

Notele studenților al căror nume nu se termină în 'escu'

```

SELECT S.NrLeg, Nume,Prenume ,Denumire, Nota,Data
FROM Student S,Catalog C, Discipline D
WHERE S.Nrleg=C.NrLeg
      AND D.cod_disciplina=C.cod_disciplina
      AND Nume in (SELECT Nume
                  FROM Student
                  WHERE Nume not like '%escu')

```

```

SELECT S.NrLeg, Nume,Prenume ,Denumire, Nota,Data
FROM Student S,Catalog C, Discipline D
WHERE S.Nrleg=C.NrLeg
      AND D.cod_disciplina=C.cod_disciplina
      AND Nume =any (SELECT Nume
                    FROM Student
                    WHERE Nume not like '%escu')

```

```

SELECT S.NrLeg, Nume,Prenume ,Denumire, Nota,Data
FROM Student S,Catalog C, Discipline D
WHERE S.Nrleg=C.NrLeg
      AND D.cod_disciplina=C.cod_disciplina
      AND Nume <> all (SELECT Nume
                    FROM Student
                    WHERE Nume like '%escu')

```

Notele studenților al căror nume se termină în 'escu'

```

SELECT S.NrLeg, Nume,Prenume ,Denumire, Nota,Data
FROM Student S,Catalog C, Discipline D
WHERE S.Nrleg=C.NrLeg
      AND D.cod_disciplina=C.cod_disciplina
      AND Nume <> all (SELECT Nume
                    FROM Student
                    WHERE Nume not like '%escu')

```

Mediile studenților al căror nume începe cu 'Pop'

```
SELECT S.NrLeg Legitimatie, Nume+' '+Prenume Student,
AVG(Nota)Media
FROM Student S,Catalog C, Discipline D
WHERE S.Nrleg=C.NrLeg
      AND D.cod_disciplina=C.cod_disciplina
      AND Nume in (SELECT Nume
                    FROM Student
                    WHERE Nume like 'Pop%')
GROUP by S.NrLeg,Nume+' '+Prenume
```

Studenții care au note mai mari decât media tuturor studenților

```
SELECT S.NrLeg Legitimatie, Nume+' '+Prenume Student, Nota
FROM Student S,Catalog C, Discipline D
WHERE S.Nrleg=C.NrLeg
      AND D.cod_disciplina=C.cod_disciplina
      AND Nota > (SELECT AVG(Nota)
                  FROM Catalog )
Order by S.NrLeg
```

Studentii care au media mai mare decât media tuturor studenților

```
SELECT S.NrLeg Legitimatie, Nume+' '+Prenume Student, AVG(Nota)
Media
FROM Student S,Catalog C, Discipline D
WHERE S.Nrleg=C.NrLeg
      AND D.cod_disciplina=C.cod_disciplina
GROUP BY S.NrLeg,Nume+' '+Prenume
HAVING AVG(Nota) > (SELECT AVG(Nota)
                    FROM Catalog )
Order by S.NrLeg
```

Subinterogari iImbricate

Studenții care au note mai mari decât media tuturor studentilor

```
SELECT *
FROM Student S
WHERE S.NrLeg in (SELECT S1.NrLeg
                  FROM Student S1, Catalog C
                  WHERE S1.NrLeg=C.NrLeg
                  AND Nota>(SELECT avg(Nota)
                           FROM Catalog ))
```

Studenții care au media mai mare decât media tuturor studenților

```
SELECT *
FROM Student S
WHERE S.NrLeg in (SELECT S1.NrLeg
```

```

FROM Student S1, Catalog C
WHERE S1.NrLeg=C.NrLeg
GROUP by S1.NrLeg
HAVING avg(C.Nota)>(SELECT avg(Nota)
FROM Catalog ) )

```

**Subinterogari corelate –ce permit folosirea unei referinte externe**

**Cataloge studenților care au numele 'Popescu'**

```

SELECT *
FROM Catalog C
WHERE 'Popescu'=(SELECT Nume
FROM Student S
WHERE S.NrLeg=C.NrLeg)

```

echivalent cu ..

```

SELECT C.*
FROM Student S, Catalog C
WHERE S.NrLeg=C.NrLeg
AND Nume='Popescu'

```

**Calculul unei medii a materiilorlor ce depasesc media pe fiecare student**

```

SELECT S.NrLeg, avg(C.Nota) Media, Count(S.NrLeg) NrMaterie
FROM Student S, Catalog C
WHERE S.NrLeg=C.NrLeg
AND C.Nota>=(SELECT avg(C1.Nota)
FROM Student S1,Catalog C1
WHERE S1.NrLeg=C1.NrLeg AND C1.NrLeg=S.NrLeg)
GROUP by S.NrLeg

```

**Afisarea pt fiecare student a materiei cu nota maxima mai mare decât media sa**

```

SELECT S.NrLeg, max(C.Nota) Nota_Max, Count(S.NrLeg) NrMaterie
FROM Student S, Catalog C
WHERE S.NrLeg=C.NrLeg
GROUP by S.NrLeg
HAVING max(C.Nota)>=(SELECT avg(C1.Nota)
FROM Student S1,Catalog C1
WHERE S1.NrLeg=C1.NrLeg AND C1.NrLeg=S.NrLeg)

```

**Afisarea mediei fiecarui student...pt comparare cu exemplul anterior**

```

SELECT S.NrLeg, avg(C.Nota) Media
FROM Student S,Catalog C
WHERE S.NrLeg=C.NrLeg
GROUP by S.NrLeg

```