

Clauze

WHERE
ORDER BY
GROUP BY
HAVING

Subinterogări. Reguli

Subinterogările oferă un mijloc de a baza o interogare pe o alta (imbricarea interogărilor). Trebuie respectate regulile:

- lista de selecție a unei subinterogări create cu ajutorul unui operator de comparare poate conține doar o singură expresie sau nume de coloană;
- dacă clauza WHERE a unei interogări exterioare conține numele unei coloane, aceasta trebuie să fie compatibilă din punct de vedere al joncțiunii cu coloana din subinterogare;
- tipurile de date *ntext*, *text* și *image* nu sunt permise într-o subinterogare;
- într-o subinterogare nu poate fi utilizată clauza ORDER BY decât dacă s-a utilizat și restricția TOP;
- o vedere creată pe baza unei subinterogări nu poate fi actualizată;

Obs. Construcțiile SELECT pot fi imbricate pe max. 16 nivele. O subinterogare diferă de operatorul de cuplare prin aceea că rezultatul final conține date doar de la ultimul tabel.

Subinterogări

Limbajul SQL permite folosirea clauzei SELECT care să conțină în interior una dintre următoarele structuri:

- alte clauze SELECT
- comenzi SELECT...INTO
- comenzi INSERT...INTO
- comenzi DELETE
- comenzi UPDATE

O clauză SELECT care este inserată în cadrul altei clauze SELECT este numită subinterogare. O subinterogare se folosește, de obicei, în cadrul unei clauze WHERE. Sintaxa unei subinterogări poate avea una dintre următoarele variante prezentate în cele ce urmează.

Varianta 1

Predicatele [ANY | SOME | ALL]

Această variantă este folosită atunci când se compară o valoare cu o altă valoare obținută în urma folosirii unei subinterogări. De exemplu, următoarea construcție întoarce toți studenții (și notele corespunzătoare acestora) care au cele mai mari note.

```
SELECT Nume, Prenume, Nota
FROM Student INNER JOIN Note ON Student.Cods=Note.Cods
WHERE Nota = (SELECT Max(Nota) FROM Note);
```

ceea ce conduce la un rezultat diferit de varianta:

```
SELECT Nume, Prenume, MAX(Nota)
FROM Student INNER JOIN Note ON Student.Cods=Note.Cods
GROUP BY Nume, Prenume;
```

în care răspunsul corespunde întrebării: "Care este cea mai mare notă obținută de fiecare student?"

Se observă faptul că, deoarece subinterogarea întoarce o singură valoare, nu este nevoie să se folosească nici unul dintre predicatele ANY, SOME sau ALL. Următoarea construcție prezintă toți studenții (și notele corespunzătoare acestora) care au note mai mari decât notele studentului Radu Tiberiu:

```
SELECT Nume, Prenume, Nota
FROM Student INNER JOIN Note ON Student.Cods=Note.Cods
WHERE Nota > All(
    SELECT Nota
    FROM Note INNER JOIN Student ON Student.Cods =Note.Cods
    WHERE Nume = 'Radu' And Prenume='Tiberiu');
```

Se observă faptul că predicatele ANY și SOME conduc la obținerea aceluiași rezultat, returnând toate variantele ce respectă condiția de comparare pentru cel puțin una dintre valorile returnate de către subinterogare. De exemplu, dacă se înlocuiește ALL cu SOME în interogarea anterioară, rezultatul obținut va conține toți studenții (și notele corespunzătoare acestora) care sunt mai mari decât cea mai mică notă acordată studentului Radu Tiberiu.

Varianta 2

Predicatul [NOT] IN

Această variantă se folosește la căutarea unei valori a unei coloane dintr-un tabel rezultat în urma alteri interogări. De exemplu, următoarea construcție returnează toți studenții (și notele corespunzătoare) din tabelul "Student" care nu apar în tabelul "Note" (studenții care nu au note):

```
SELECT Cods, Nume, Prenume  
FROM Student  
WHERE Cods NOT IN (SELECT Cods FROM Note);
```

Varianta 3

Predicatul [NOT] EXISTS

Această variantă se folosește atunci când se verifică dacă o anumită valoare există (este returnată) în urma executării subinterogării. De exemplu, următoarea construcție afișează toți studenții care nu au note:

```
SELECT Cods, Nume, Prenume  
FROM Student  
WHERE NOT EXISTS  
(SELECT * FROM Note WHERE Note.Cods =Student.Cods);
```

Se observă faptul că tabelul "Note" este referit în subinterogare, ceea ce face ca SQL Server să evalueze subinterogarea câte o dată pentru fiecare valoare a identificatorului "CodS" din tabelul "Note".

Observații importante

1. Atunci când subinterogarea are ca rezultat un singur rând, se pot folosi operatori de comparare. În caz contrar trebuie folosite predicate.
2. Atunci când se folosește varianta 1 sau 2, subinterogarea trebuie să returneze o singură coloană, altfel va apare un mesaj de eroare.
3. Construcția SELECT care conține subinterogarea păstrează același format și reguli ca oricare altă construcție SELECT, dar trebuie în mod obligatoriu introdusă între paranteze.

Să se răspundă la întrebări în felul următor:

1. La întrebările de la 1 la 7 folosind clauzele: WHERE, ORDER BY, GROUP BY, HAVING
2. La celelalte întrebări folosind subinterogări cu una dintre variantele 1, 2, sau 3:

1. Să se afișeze ce cursuri are studentul Radu Tiberiu
2. Numele studentelor de la zi care sunt bursiere
3. Numele studenților bursieri și a studentelor ce au început studiile la data de 1 octombrie 2001
4. Numele studenților din ce au început studiile la data de 1 octombrie 2001 sau bursieri
5. Numărul studenților din fiecare an de studiu
6. Să se afișeze media notelor obținute de studenți
7. Să se afișeze studenții a căror medie este mai mare de 8
8. Să se utilizeze o subinterogare prin care să se afle notele studenților care nu sunt din județul Iași
9. Să se afle numele studenților care au o notă mai mare decât cea mai mică notă acordată
10. Cine a plătit cea mai mare taxă?
11. Să se determine numele și prenumele studenților care au platit taxa de 35.50 lei
12. Să se afișeze cursurile și notele obținute la aceste cursuri de către studentul Radu Tiberiu

- 13. Să se afișeze cursurile și notele obținute la aceste cursuri de către studentul ce are codul 005 care sunt mai mici decât cea mai mare notă obținută de către acesta.**
- 14. Să se afișeze numele, prenumele, cursurile și notele cele mai mici obținute la aceste cursuri de către fiecare student.**
- 15. Folosind o variantă cu rezultate multiple, să se afișeze numele, prenumele, cursurile și notele cele mai mici obținute la aceste cursuri.**