

# Laboratorul 10

De la BD

## Subinterogări corelate

Așa cum am văzut, valorile returnate de o subinterogare sunt în general utilizate de către interogarea principală pentru a filtra informațiile selectate. Din acest motiv, subinterogarea este executată o singură dată la început.

```
SELECT *
  FROM studenti NATURAL JOIN note
 WHERE valoare >=
        (SELECT ROUND(AVG(valoare))
         FROM note
        );
```

În acest caz, este suficient ca subinterogarea să fie executată o singură dată pentru a se afla media tuturor notelor. Odată aflată, această medie este utilizată în interogarea principală pentru a filtra notele și a le afișa doar pe cele ce depășesc această valoare.

În cazul subinterogărilor corelate, subinterogarea este evaluată pentru fiecare rând generat de interogarea principală. De această dată, la fiecare rulare, subinterogarea utilizează o valoare generată de interogarea principală.

Să considerăm interogarea următoare:

```
SELECT nume, prenume, bursa, an
  FROM studenti s1
 WHERE bursa >=
        (SELECT AVG(bursa)
         FROM studenti s2
          GROUP BY an HAVING s1.an = s2.an
        );
```

Această interogare va returna toți studenții care au bursă mai mare sau cel puțin egală cu media burselor date în anul lor (deci nu media burselor din toți anii ci doar media burselor studenților din același an cu ei).

Să considerăm prima linie returnată: Antonie Ioana are bursa egală cu 450 care este mai mare sau egală cu media burselor studenților din anul 3. Această medie a putut fi obținută numai din subinterogarea în care am ținut cont că media trebuie să fie calculată doar pentru anul 3 (în interogarea interioară am utilizat valoarea s1.an). Linia următoare este pentru un student din anul 2. Deoarece puțin mai înainte subinterogarea a întors media burselor studenților din anul 3, aceasta trebuie recalculată pentru studenții din anul 2 - deci subinterogarea va fi executată din nou. Procesul continuă și, pentru fiecare linie ce este returnată în cadrul interogării principale, este calculată media studenților din același an cu studentul din linia curentă. Linia curentă este apoi filtrată în funcție de criteriul stabilit (vor fi considerate doar liniile în care bursa este mai mare sau egală cu această medie).

În cazul interogărilor corelate, interogarea principală preia unul câte unul toate randurile și de fiecare dată le compară cu valorile ce sunt returnate de către subinterogare. Subinterogarea va returna (de obicei) date diferite în funcție de valoarea unui câmp din interogarea principală (în exemplul anterior, valoarea câmpului an).

Oracle tratează subinterogarea ca și corelată de fiecare dată când aceasta utilizează valoarea unei coloane returnată de interogarea principală.

## EXISTS

Uneori, nu ne interesează efectiv valoarea returnată de subinterogare ci doar dacă aceasta returnează măcar un rând. Spre exemplu, am putea fi interesați să afișăm toți studenții care au în grupă măcar un bursier. Deoarece grupa este identificată atât prin an cât și prin grupă, ambele câmpuri vor fi utilizate în subinterogare. Dacă subinterogarea returnează măcar un rând (de fapt, în subinterogare vom selecta count(\*) și vom ține cont ca valoarea să fie strict mai mare ca 0) atunci înseamnă că studentul din rândul curent are în grupă măcar un bursier:

```
SELECT nume, prenume, bursa, grupa, an
FROM studenti s1
WHERE (
    SELECT COUNT(*)
    FROM studenti s2
    WHERE bursa IS NOT NULL AND
          s1.an = s2.an AND
          s1.grupa=s2.grupa
    ) > 0;
```

Operația count în acest caz este destul de costisitoare: chiar dacă s-a găsit că există un student bursier, tabelul studenti este parcurs în continuare de către subinterogare pentru a-i găsi și pe alți eventuali studenți care sunt în aceeași grupă și au bursă). Cu alte cuvinte, nu ne interesează foarte tare dacă operația de numărare returnează 1,2 sau mai mulți bursieri aflați în aceeași grupă ci doar că acolo există măcar unul.

Un operator important în cazul interogărilor corelate este operatorul exists. Să adaptăm interogarea anterioară în felul următor:

```
SELECT nume, prenume, bursa, grupa, an
FROM studenti s1
WHERE EXISTS (
    SELECT *
    FROM studenti s2
    WHERE bursa IS NOT NULL AND
          s1.an = s2.an AND
          s1.grupa=s2.grupa
    );
```

De această dată, subinterogarea nu este executată în întregime ci numai până când se găsește primul student bursier. În acel moment, condiția este îndeplinită și nu se încearcă numărarea tuturor studenților ce au bursă în acea grupă ci se va trece la următoarea linie din interogarea principală.

În versiunile anterioare, subinterogările erau materializate chiar și în cazul lui EXISTS. Acest lucru înseamnă că

efectiv erau întoarse randuri de către subinterogare. Pentru a se evita accesul la HDD, ca și optimizare, se putea cere altceva decât randuri efective din tabel. Spre exemplu, în locul informației se putea trece o constantă:

```
SELECT nume, prenume, bursa, grupa, an
FROM studenti s1
WHERE EXISTS (
    SELECT 'bananaaaa (https://www.youtube.com/watch?v=sFukyIIM1XI)'
    FROM studenti s2
    WHERE bursa IS NOT NULL AND
          s1.an = s2.an AND
          s1.grupa=s2.grupa
);
```

Probabil că nu veți sesiza vreo diferență în contextul utilizării unor tabele ce conțin puține înregistrări, dar, în realitate, unde sunt milioane de înregistrări într-o tabelă, orice optimizare este binevenită.

Întrebare: ce se întâmplă atunci când este utilizat NOT EXISTS (pentru că ne interesează dacă nu avem înregistrări corelate) ? Pentru cât timp este executată subinterogarea ? Când se iese din subinterogare ?

## Exerciții

1. Afișați toți studenții care au media mai mare decât media tuturor studenților din ani cu ei. Pentru aceștia afișați numele, prenumele și media lor.
2. Afișați numele și prenumele celui mai bun student din fiecare în parte.
3. Găsiți toți studenții care au măcar un coleg în același an care să fi luat aceeași nota ca și el la măcar o materie.
4. Afișați toți studenții care au în an cu ei măcar un coleg care să fie mai mic ca ei (vezi data nașterii) și care au media notelor din toți mai mare decât a lor.

Adus de la „[http://85.122.23.37/BD/index.php?title=Laboratorul\\_10&oldid=447](http://85.122.23.37/BD/index.php?title=Laboratorul_10&oldid=447)”

- 
- Ultima modificare efectuată la 23:29, 8 decembrie 2015.
  - Conținutul este disponibil sub Creative Commons Atribuire, exceptând cazurile în care se specifică altfel.