BAZE DE DATE - 4

Autor

id_autor	nume_autor
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

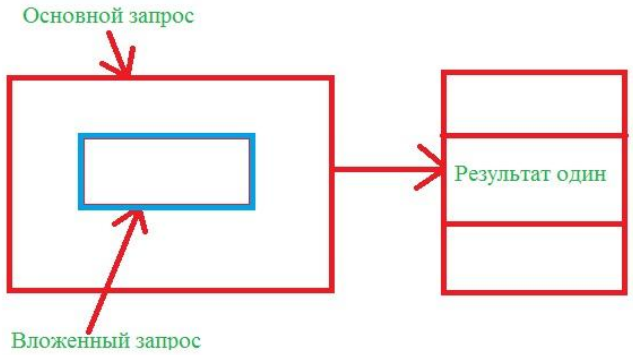
Carti

idcarte	autor	titlu	pret	cantitatea	id_autor
2-2222-222-10	Petru GFGF	Laborator Mysql-Php vbvbvb	950.00	23	1
2-2222-222-13	Vieru	MAMA	2000.00	200	3
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40	4
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3	6
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3	2

Limbajul SQL

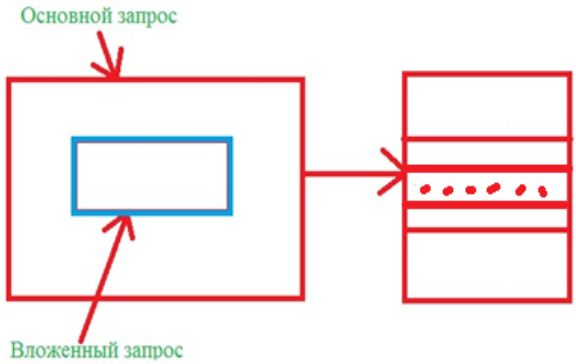
Limbajul SQL

Subinterogări (Subqueries)



1. **SINGLE ROW SUBQUERIES**

2. **MULTIPLE ROW SUBQUERIES**



EN
https://www.w3resource.com/sql/subqueries/understanding-sql-subqueries.php
RU
https://www.internet-technologies.ru/articles/podzaprosy-sql.html

SUBQUERIES (Subinterogari)

În **SQL**, subinterogările ne permit să aflăm **o informaţie care ne este necesară pentru a obţine informaţia *pe care o vrem***.

- O **subinterogare (subquery)** este o instrucţiune **SELECT** care este inclusă în clauza unei alte instrucţiuni **SELECT**.

SUBQUERIES (Subinterogari)

- Subinterogarea poate fi plasata în una din următoarele clauze:
  1. **WHERE**
  2. **HAVING**
  3. **FROM**
- **Subinterogarea** se execută prima dată, iar rezultatul este folosit pentru obţinerea rezultatului de către interogarea principală (**outer query**).

SUBQUERIES (Subinterogari)

Sintaxa generală:

```
SELECT select_list
FROM table
WHERE expression operator
                    (SELECT select_list
                     FROM table);
```

SUBQUERIES (Subinterogari)

Reguli de folosire a subinterogarilor

- O **subinterogare** se pune între paranteze rotunde
- O **subinterogare** este plasata în partea dreaptă a unei condiţii de comparare
- Interogarea exterioară şi **subinterogarea**-ul pot prelua date din tabele diferite

SUBQUERIES (Subinterogari)

- Într-o instrucţiune **SELECT** se poate folosi o singură clauză **ORDER BY** şi, dacă se foloseşte, trebuie să fie ultima clauza a interogării principale.
- O subinterogare nu poate avea propria clauză **ORDER BY**.
- Singura limită a numărului de interogări este dimensiunea buffer-ului folosit de interogare.
- Dacă subinterogarea returnează null sau nu returneaza nici o linie, atunci interogarea exterioară nu va returna nimic.

SUBQUERIES (Subinterogari)

Sunt două tipuri de subinterogări(subqueries):

1) **single-row subqueries** – care folosesc operatorii single-row: **>, =, >=, <, <=** şi dau ca rezultat o singură linie.

2) **multiple-row subqueries** – care folosesc operatorii multiple-row: **IN, ANY, ALL** şi dau ca rezultat mai multe linii.

Limbajul SQL

SUBQUERIES (Subinterogări)

1. **SINGLE ROW SUBQUERIES**
2. **MULTIPLE ROW SUBQUERIES**

5.1. SINGLE ROW SUBQUERIES

Single row-subquery

Aflati prenumele angajatilor care au salariul mai mare decat angajatul care se numeste Costica.

```
SELECT prenume
FROM angajati
WHERE salariu  >
        ( SELECT salariu
          FROM angajati
          WHERE prenume='Costica' );
```

5.1. SINGLE ROW SUBQUERIES

Subcereri din mai multe tabele

- Subcererile (subinterogările) nu sunt limitate la o singură interogare (cerere).
- Aşa cum se poate observa în exemplul următor, pot fi mai mult de o singură interogare.
- De asemenea se pot face interogări din tabele diferite.

Exemplul urmator afiseaza angajatii a caror functie este acelasi cu cel al angajatului cu numarul 7369 si a caror salariu este mai mare decat cel al angajatului 7875.

Executarea unei subinterogari single-row

```
SELECT   nume, functie
FROM     angajati
WHERE    functie =                          FUNCTIONAR
               (SELECT  functie
                FROM    angajati
                WHERE   id_angajat = 7369)
               AND      salariu >
                        (SELECT  salariu                  1100
                         FROM            angajati
                         WHERE           id_angajat = 7876);
```

NUME	FUNCTIE
MILLER	CLERK

5.1. SINGLE ROW SUBQUERIES

- Exemplul este format din 3 blocuri de cereri:
  - o cerere exterioara
  - doua cereri interne
- Blocurile de cereri interne sunt primele executate, producand rezultatele cererii: FUNCTIONAR (CLERK), respectiv 1100.
- Blocul exterior de cereri este apoi procesat si foloseste valorile returnate de catre cererile interne pentru a finaliza propriile conditii de cautare.
- Ambele cereri interne returneaza valori singulare (FUNCTIONAR si 1100), astfel ca aceasta instructiune SQL este denumita o *subinterogare single-row*.

5.1. SINGLE ROW SUBQUERIES

```
SELECT nume, id_functie, salariu, id_dept
FROM angajati
WHERE id_functie =
                    (SELECT id_functie
                     FROM angajati
                     WHERE id_angajat=141)
AND id_dept =
                    (SELECT id_dept
                     FROM departamente
                     WHERE id_locatie=1500);
```

5.1. SINGLE ROW SUBQUERIES

Se pot folosi funcţiile de grup în subinterogări.

O funcţie de grup utilizată în subquery fără clauza GROUP BY, *returnează o singură linie*.

```
SELECT nume, prenume, salariu
FROM angajati
WHERE salariu <
                (SELECT MAX(salariu)
                 FROM angajati);
```

5.1. SINGLE ROW SUBQUERIES

- Subinterogările pot fi plasate şi în clauza **HAVING.**
- Deoarece clauza **HAVING** are întotdeauna o condiţie de grup, şi subinterogarea va avea aproape întotdeauna o condiţie de grup.

```
SELECT id_dept, MIN(salariu)
FROM angajati
GROUP BY id_dept
HAVING MIN(salariu) >
                        (SELECT MIN(salariu)
                         FROM angajati
                         WHERE id_dept = 50);
```

5.1. SINGLE ROW SUBQUERIES

APLICAŢII

1) Care este numele membrilor din personalul de la firma “COSTICA S.R.L.”, al căror salariu este mai mare decât angajatul cu ID-ul 12?

2) Care dintre angajaţii Oracle au acelaşi id al departamentului ca şi cel corespunzător cu departamentul IT?

5.1. SINGLE ROW SUBQUERIES

1) Care este numele membrilor din personalul de la firma "COSTICA S.R.L.", al căror salariu este mai mare decât angajatul cu ID-ul 12?

```
SELECT nume
FROM angajati
WHERE salariu >
                    (SELECT salariu
                     FROM angajati
                     WHERE id_angajat = 12);
```

5.1. SINGLE ROW SUBQUERIES

2) Care dintre angajatii Oracle au acelasi id al departamentului ca si cel corespunzator cu departamentul IT?

```
SELECT nume, prenume
FROM angajati
WHERE id_dept =
                    ( SELECT id_dept
                      FROM departamente
                      WHERE nume_department = 'IT' );
```

Limbajul SQL

5. SUBQUERIES (Subinterogări)

1. SINGLE ROW SUBQUERIES

2. MULTIPLE ROW SUBQUERIES

5.2. MULTIPLE ROW SUBQUERIES

Sunt acele subinterogări care dau ca rezultat mai multe valori.

Folosesc operatorii **multiple row**:

1. **IN**
2. **ANY**
3. **ALL**

Operatorul **NOT** poate fi folosit în combinaţie cu oricare dintre aceştia.

5.2. MULTIPLE ROW SUBQUERIES

1. Operatorul IN

Operatorul **IN** este folosit dacă în interogarea exterioară clauza **WHERE** este folosită pentru a selecta acele valori care sunt egale cu una dintre valorile din lista returnată de subinterogare (inner query).

```
SELECT nume, salariu, id_dept
FROM angajati
WHERE salariu IN
                ( SELECT MIN(salariu)
                  FROM angajati
                  GROUP BY id_dept );
```

Funcţia de grup utilizată în subquery in care se afla clauza GROUP BY, returnează mai multe linii (inregistrari).

5.2. MULTIPLE ROW SUBQUERIES

2. Operatorul ANY

Acest operator este folosit atunci când dorim ca interogarea exterioară să selecteze valori egale, mai mici sau mai mari decât cel puţin o valoare dintre cele extrase de subquery.

```
SELECT titlu, producator
FROM albume
WHERE an < ANY
                ( SELECT an
                  FROM albume
                  WHERE producator = 'Ice T' );
```

5.2. MULTIPLE ROW SUBQUERIES

3. Operatorul ALL

Acest operator este folosit atunci când dorim ca interogarea exterioară să selecteze valori egale, mai mici sau mai mari decât toate valorile extrase de subquery.

```
SELECT titlu, producator
FROM albume
WHERE an > ALL
                ( SELECT an
                  FROM albume
                  WHERE producator = 'Ice T' );
```

5.2. MULTIPLE ROW SUBQUERIES

VALORI NULL

Dacă una dintre valorile returnate de subinterogarea multiple row este null, dar celelalte valori nu sunt null, atunci:

1. Dacă sunt folosiţi operatorii **IN** sau **ANY**, interogarea exterioară (outer query) va returna liniile care se potrivesc cu valorile non-null.
2. Dacă este folosit operatorul **ALL**, interogarea exterioară (outer query) nu va returna nimic.

5.2. MULTIPLE ROW SUBQUERIES

Clauzele GROUP BY şi HAVING

- Pot fi folosite cu subinterogările de tip multiple row.

```
SELECT id_dept, MIN(salariu)
FROM angajati
GROUP BY id_dept
HAVING MIN(salariu) < ANY
                        ( SELECT salariu
                          FROM angajati
                          WHERE id_dept IN (10,20) );
```

5.2. MULTIPLE ROW SUBQUERIES

Clauzele GROUP BY si **HAVING**

De asemenea, se poate folosi clauza **GROUP BY** intr-o subinterogare (inner query)

```
SELECT id_dept, MIN(salariu)
FROM angajati
GROUP BY id_dept
HAVING MIN(salariu) > ALL
                      (SELECT MIN(salariu)
                       FROM angajati
                       WHERE id_dept < 50
                       GROUP BY id_dept);
```

5.2. MULTIPLE ROW SUBQUERIES

APLICATII

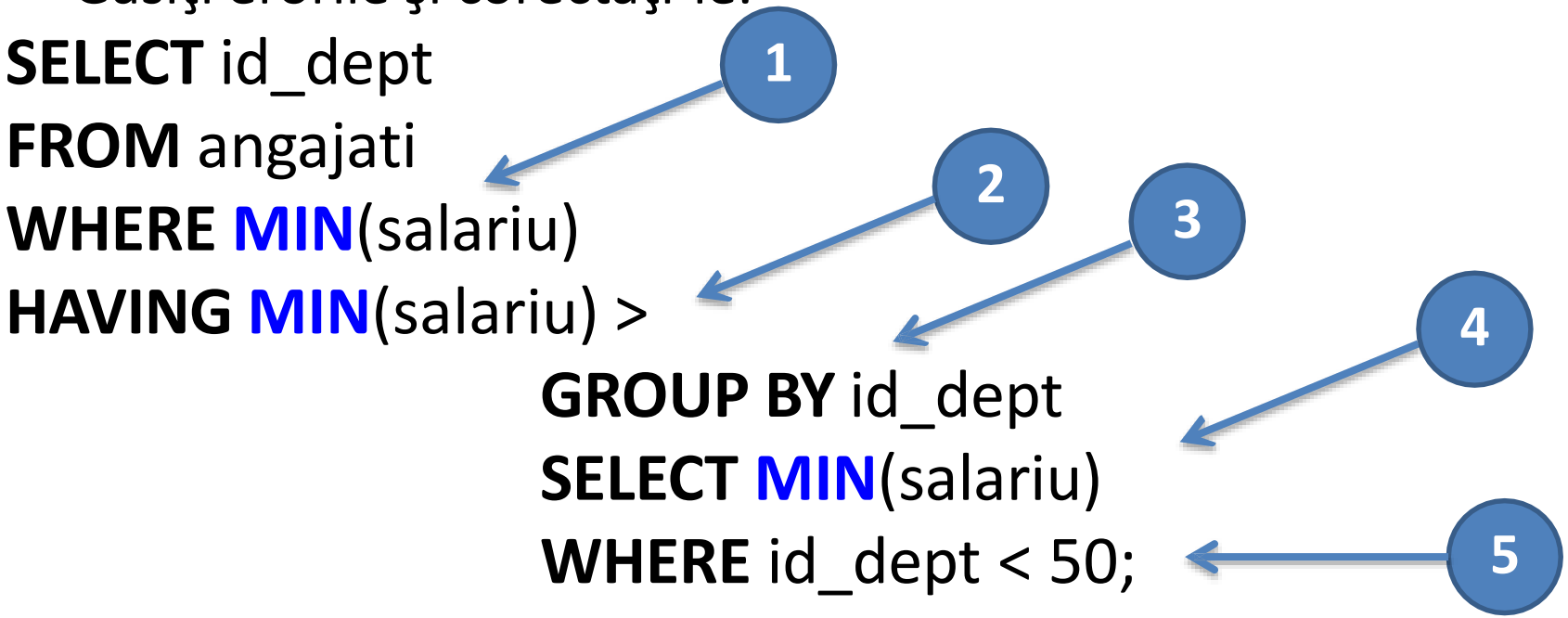
1) Găsiţi numele pentru toţi angajaţii ale căror salarii sunt aceleaşi cu salariul minim din oricare (any) departament.

```
SELECT nume
FROM angajati
WHERE salariu = ANY
                    (SELECT MIN(salariu)
                     FROM angajati
                     GROUP BY id_dept);
```

5.2. MULTIPLE ROW SUBQUERIES

2) Scopul interogării următoare este de a afişa salariul minim pentru fiecare departament al cărui salariu minim este mai mic decât cel mai mic salariu al angajaţilor din departamentul 50.

Oricum, subinterogarea nu se execută deoarece are 5 erori. Găsiţi erorile şi corectaţi-le.



5.2. MULTIPLE ROW SUBQUERIES

Soluţia corectă este următoarea:

```
SELECT   id_dept, MIN(salariu)
FROM     angajati
GROUP BY id_dept
HAVING   MIN(salariu) <
                              (SELECT MIN(salariu)
                               FROM   angajati
                               WHERE  id_dept = 50);
```

Subcereri multilinie (multiple-row queries)

- Subcererile multilinie returneaza mai mult decat o linie.
- Cu astfel de subcereri trebuie folositi operatori multilinie care pot prelucra una sau mai multe valori.

Operatorii utilizati sunt:

1. IN - egal cu oricare dintre membrii unei liste
2. ANY/SOME - compara o valoare cu fiecare (vreo) valoare returnata de subcerere
3. ALL - compara o valoare cu oricare (toate) din valorile returnate de subcerere

Exemplu IN

- Aflati angajatii care au salariul egal cu salariul cel mai mare din fiecare departament
- SELECT nume, id_dept, salariu FROM angajati
- WHERE salariu **IN**
  - (SELECT MAX(salariu) FROM angajati GROUP BY id_dept)
- Subcererea ofera salariile maxime din fiecare departament si prin cererea principala se afla angajatii

Exemplu **ANY**

Aflati angajatii care au salariul mai mare decat vreun angajat al departamentului 20 si nu fac parte din acest departament.

```
SELECT nume, id_dept, salariu
        FROM angajati
    WHERE salariu > ANY
                ( SELECT salariu
                FROM angajati
                WHERE id_dept=20 )
AND id_dept<>20
```

- Operatorul **ANY** (sinonim operatorului SOME) compara o valoare cu fiecare valoare din cele returnate de subcerere.

Astfel,

< ANY inseamna mai mic decat maximul

ANY inseamna mai mare decat minimul

= ANY este echivalent cu **IN**

Exemplu ALL

Gasiti angajatii care au salariul mai mic decat oricare (toti) angajatii de la departamentul 5.

```
SELECT nume, id_dept, salariu
FROM angajati
WHERE salariu < ALL
                    ( SELECT salariu
                      FROM angajati
                      WHERE id_dept=5 )
AND id_dept<>5
```

- Operatorul **ALL** din cererea principala compara o valoare cu oricare valoare returnata de subcerere.

Astfel:

> ALL inseamna mai mare decat maximul

< ALL inseamna mai mic decat minimul

Imbricarea subcererilor

Subcererile pot fi folosite si in interiorul altor subcereri.

Exemplu

Gasiti numele, functia, data angajarii si salariul angajatilor al caror salariu este superior celui mai mare salariu al vreunei persoane angajate dupa data de 05-JUN-1982

```
SELECT nume, functie, data_ang, salariu
                FROM angajati
                WHERE salariu >
        (SELECT MAX(salariu)
        FROM angajati
        WHERE data_ang IN
                (SELECT data_ang
                FROM angajati
                WHERE data_ang>'05-JUN-1982'))
```

Numarul maxim de imbricari pentru o subcerere este de 255.

Subcereri corelate

- *O subcerere corelata este o subcerere care se executa o data pentru fiecare linie considerata de cererea principala si care la executie foloseste o valoare dintr-o coloana din cererea exterioara.*

- Ea se poate identifica prin folosirea unei coloane a cererii exterioare in clauza operatorului cererii interioare.

Exemplu

Gasiti angajatii care au un salariu superior salariului mediu al departamentului lor.

```
SELECT nume, salariu, id_dept
FROM angajati A
WHERE salariu >
                    (SELECT AVG(salariu)
                     FROM angajati
                     WHERE (id_dept=A.id_dept))
ORDER BY id_dept
```

Valori de NULL intr-o subcerere

- In cazul in care subcererea returneaza vreuna din valori NULL si cererea principala are operator NOT IN, atunci cererea principala nu va returna niciun rand.

- Motivul este ca *o comparatie cu NULL conduce la un rezultat NULL*.

Exemplu

Gasiti angajatii care nu au subordonati.

```
SELECT nume
FROM angajati
WHERE id_angajat NOT IN
                    ( SELECT manager
                      FROM angajati );
```

- Astfel ori de cate ori valoarea NULL face parte din raspunsurile subcererii nu trebuie folosit operatorul NOT IN.

- De fapt operatorul NOT IN este echivalent cu <> ALL.

- Returnarea de valori NULL de catre subcerere nu prezinta nici o problema in cazul operatorului IN in cererea principala (in echivalent cu = ALL).

Exemplu

Gasiti angajatii care au subordonati.

```
SELECT nume
FROM angajati
WHERE id_angajat IN
                    ( SELECT manager
                      FROM angajati );
```

- In cazul utilizarii operatorului NOT IN in cererea principala trebuie avut grija sa se excluda valorile NULL din raspunsurile subcererii.

Exemplu

Gasiti angajatii care nu au subordonati.

```
SELECT nume
FROM angajati
WHERE id_angajat NOT IN
                    ( SELECT manager
                      FROM angajati
                      WHERE manager IS NOT NULL );
```

Sfaturi în utilizarea subinterogarilor

1. Includerea subinterogarilor în paranteze
2. Plasarea subinterogarilor în partea dreapta a operatorului de comparare
3. A nu se adauga clauza ORDER BY într-o subinterogare
4. Folosirea operatorilor single-row în subinterogari single-row
5. Folosirea operatorilor multiple-row în subinterogari multiple-row

Concluzii

1. O subinterogare este o instructiune SELECT inclusa într-o clauza a altei instructiuni SQL.
2. Subinterogarile sunt folositoare atunci cînd interogarea se bazeaza pe criterii necunoscute.
3. Subinterogarile au urmatoarele caracteristici:
   a) Pot transmite un rand de date instructiunii principale care contine un operator single-row, precum: =, <>, >, >=, < sau <=;
   b) Pot transmite rînduri multiple de date instructiunii principale care contine un operator multiple-row, precum: IN, ANY sau ALL;
   c) Sunt primele procesate de catre server-ul Oracle, iar clauzele WHERE si HAVING folosesc rezultatele;
   d) Pot contine functii de grup.

Întrebări?