UNIUNEA EUROPEANĂ

GUVERNUL ROMÂNIEI

Instrumente Structurale
2007-2013

Platformă de e-learning și curriculă e-content pentru învățământul superior tehnic

Baze de date 1

8. Subcereri în SQL (2)

SUBCERERI PE ALTE CLAUZE DECÂT SELECT

- Expresiile logice conţinând subcereri se pot folosi şi pe clauzele HAVING, FROM, SELECT, ORDER BY în acelaşi mod ca în modulul anterior.
- Tot în acest modul vom prezenta şi subcererile corelate.

STUD

MATR	NUME	AN	GRUPA	DATAN	LOC	TUTOR	PUNCTAJ	CODS
1456	GEORGE	4	1141A	12-MAR-82	BUCURESTI		2890	11
1325	VASILE	2	1122A	05-OCT-84	PITESTI	1456	390	11
1645	MARIA	3	1131B	17-JUN-83	PLOIESTI		1400	11
3145	ION	1	2112B	24-JAN-85	PLOIESTI	3251	1670	21
2146	STANCA	4	2141A	15-MAY-82	BUCURESTI		620	21
3251	ALEX	5	2153B	07-NOV-81	BRASOV		1570	21
2215	ELENA	2	2122A	29-AUG-84	BUCURESTI	2146	890	21
4311	ADRIAN	3	2431A	31-JUL-83	BUCURESTI		450	24
3514	FLOREA	5	2452B	03-FEB-81	BRASOV		3230	24
1925	OANA	2	2421A	20-DEC-84	BUCURESTI	4311	760	24
2101	MARIUS	1	2412B	02-SEP-85	PITESTI	3514	310	24
4705	VOICU	2	2421B	19-APR-84	BRASOV	4311	1290	24

SPEC si BURSA

CODS	NUME	DOMENIU
11	MATEMATICA	STIINTE EXACTE
21	GEOGRAFIE	UMANIST
24	ISTORIE	UMANIST

TIP	PMIN	PMAX	SUMA
FARA BURSA	0	399	
BURSA SOCIALA	400	899	100
BURSA DE STUDIU	900	1799	150
BURSA DE MERIT	1800	2499	200
BURSA DE EXCEPTIE	2500	9999	300

SUBCERERI PE HAVING

- Expresiile logice conţinând subcereri se pot folosi şi pe clauza HAVING în acelaşi mod ca mai sus.
- În acest caz însă condiţiile vor conţine doar elementele care pot să apară într-o astfel de clauză: constante, expresiile după care se face gruparea şi funcţii statistice.
- În continuare sunt prezentate câteva exemple:

EXEMPLE (1)

- Afişarea codului numeric şi a punctajulelor minim şi maxim doar pentru specializările care au un punctaj MEDIU peste media calculată la nivelul întregii tabele STUD:

```
SELECT CODS, MIN(PUNCTAJ), MAX(PUNCTAJ)
FROM STUD
GROUP BY CODS
HAVING AVG(PUNCTAJ) >
       (SELECT AVG(PUNCTAJ) FROM STUD);
```

EXEMPLE (2)

- Afişarea codului şi a punctajului mediu pentru specializarea cu cel mai mare punctaj mediu (functii statistice imbricate in subcerere):

```
SELECT CODS, AVG(PUNCTAJ)
FROM STUD
GROUP BY CODS
HAVING AVG(PUNCTAJ) =
  (SELECT MAX(AVG(PUNCTAJ))
   FROM STUD
   GROUP BY CODS);
```

EXEMPLE (3)

- Afişarea codului şi a punctajului maxim pentru toate specializările în afara celei/celor cu cel mai mic punctaj maxim. Pentru ca o specializare să apară în rezultat punctajul său maxim trebuie să fie mai mare decât al vreunei alte specializări. Subcererea întoarce lista punctajelor maxime iar comparaţia dorită se face folosind operatorul ANY:

```
SELECT CODS, MAX(PUNCTAJ)
FROM STUD
GROUP BY CODS
HAVING AVG(PUNCTAJ) >
   ANY (SELECT AVG(PUNCTAJ)
        FROM STUD
        GROUP BY CODS);
```

SUBCERERI PE FROM

- În cazul în care o subcerere apare pe clauza FROM ea va fi tratată ca o tabelă temporară. Cererea următoare afişează numele studenţilor bursieri de la specializarea cu codul 11.

```
SELECT NUME, SUMA
FROM (SELECT *
      FROM STUD, BURSA
      WHERE PUNCTAJ BETWEEN PMIN AND PMAX
       AND CODS = 11
       AND SUMA IS NOT NULL);
```

JOIN DE SUBCERERI

- În cazul în care subcererea este implicată într-un join se pot folsi aliasuri de tabelă pentru a dezambigua numele coloanelor rezultatului său. Cererea anterioară se poate rescrie şi astfel:

```
SELECT NUME, SUMA
FROM (SELECT * FROM STUD WHERE CODS = 11) S11,
(SELECT * FROM BURSA WHERE SUMA IS NOT NULL) B
WHERE S11.PUNCTAJ BETWEEN B.PMIN AND B.PMAX;
```

JOIN CU SUBCERERI

- Folosirea aliasurilor de tabelă este obligatorie în cazul în care tabelele implicate în join au coloane cu acelaşi nume. În exemplul se afişează numele studenţilor şi numele specializării pentru specializarea cu codul 11:

```
SELECT ST.NUME, SP.NUME
FROM STUD ST,
(SELECT * FROM SPEC WHERE CODS = 11) SP
WHERE ST.CODS = SP.CODS;
```

REZULTAT VID

- În cazul în care o subcerere aflată pe clauza FROM nu întoarce nici o linie nu se semnalează o eroare dar cererea care o include va returna un rezultat vid, inclusiv în cazul unui produs cartezian.
- Subcererea din exemplul următor nu returnează linii deoarece specializarea cu codul 100 nu există:

```
SELECT ST.NUME, SP.NUME
FROM STUD ST,
(SELECT * FROM SPEC WHERE CODS = 100) SP
WHERE ST.CODS = SP.CODS;
```

REZULTAT VID (2)

- O astfel de subcerere poate participa însă la un join extern:

```
SELECT ST.NUME, SP.NUME
FROM STUD ST, (SELECT * FROM SPEC WHERE
  CODS = 100) SP
WHERE ST.CODS = SP.CODS(+);
```

- sau:

```
SELECT ST.NUME, SP.NUME
FROM STUD ST
FULL OUTER JOIN (SELECT * FROM SPEC
                 WHERE CODS = 100) SP
ON (ST.CODS = SP.CODS);
```

SUBCERERI CORELATE

- În exemplele anterioare subcererea se execută o singură dată după care rezultatul său este folosit pentru evaluarea care o include.
- Există însă posibilitatea ca rezultatul subcererii să fie dependent de valorile de pe linia curentă a parcurgerii definite de cerere.
- În acest caz se spune că avem o *subcerere corelată*.

SUBCERERI CORELATE (2)

- Descrierea modului de evaluare în astfel de situaţii va fi făcută pe baza următorului exemplu care afişează date despre studenţii care au un punctaj peste media celor din specializarea lor.
- Deoarece şi cererea şi subcererea lucrează pe aceeaşi tabelă, se foloseşte aliasul de tabelă S pentru a specifica parcurgerea principală (cea din cerere).

```
SELECT NUME, CODS, PUNCTAJ
FROM STUD S
WHERE PUNCTAJ > (SELECT AVG(PUNCTAJ) FROM
  STUD WHERE CODS = S.CODS);
```

```
SELECT NUME, CODS, PUNCTAJ  -- CALCUL REZULTAT
FROM STUD S
WHERE PUNCTAJ > (SELECT AVG(PUNCTAJ) FROM STUD WHERE CODS = S.CODS);
```

- Pentru fiecare linie a tabelei STUD valoarea de pe coloana CODS - codul specializării - va fi transmisă subcererii (S.CODS).
- Se execută subcererea care calculează punctajul mediu pentru studenţii de la specializarea respectivă (AVG şi condiţia CODS = S.CODS).
- Rezultatul subcererii este folosit în cerere pentru a filtra sau nu linia curentă (condiţia PUNCTAJ > rezultat subcerere).
- Teoretic subcererea nu se mai execută o singură dată ci pentru fiecare linie din parcurgerea specificată de cererea în care este inclusă.

EXEMPLU

- Numele, codul şi numărul total de studenţi pentru specializările la care este înmatriculat cel puţin un student de anul 1. :

```
SELECT SP.NUME, ST.CODS, COUNT(*)
FROM STUD ST, SPEC SP
WHERE ST.CODS = SP.CODS
GROUP BY SP.NUME, ST.CODS
HAVING 1 <= (SELECT COUNT(*)
             FROM STUD
             WHERE CODS = ST.CODS AND
             AN = 1);
```

OBSERVATII

- În cazul în care subcererea corelată se află pe clauza WHERE ea poate primi valoarea de pe orice coloană a tabelei/produsului cartezian din cererea înconjurătoare.
- În cazul în care subcererea corelată este pe clauza HAVING poate primi doar coloanele/expresiile după care s-a facut gruparea.
- Subcererile corelate pot returna o valoare, o coloană sau o tabelă, ca şi cele necorelate. În cazul în care returnează o coloană sau o tabelă se poate folosi operatorul IN. Sunt permişi de asemenea operatorii SOME/ANY şi ALL.

OPERATORUL EXISTS

- Acest operator testează dacă subcererea primită ca argument întoarce un rezultat nevid. Sintaxa sa este:

```
EXISTS(subcerere)
```

- Returnează valorea logică ADEVARAT dacă subcererea are un rezultat nevid şi FALS dacă rezultatul e vid (nici o linie).
- În cazul acestui operator nu este important conţinutul rezultatului subcererii ci doar existenţa sau absenţa sa.

EXEMPLU

- Lista specializărilor unde există cel puţin un student de anul 1:

```
SELECT CODS, NUME
FROM SPEC
WHERE EXISTS (SELECT 1
      FROM STUD
      WHERE SPEC.CODS = CODS
      AND AN = 1);
```

SUBCERERI IN ORDER BY

- În actualele versiuni ale sistemului Oracle clauza ORDER BY poate conţine o subcerere Aceasta trebuie să returneze o singură valoare şi să fie o subcerere corelată.
- Prezenţa unei cereri necorelate pe această clauză nu duce la ordonarea rezultatului deoarece rezultatul fiind o constantă are aceeaşi valoare pentru fiecare dintre liniile supuse sortării.

EXEMPLU

- Ordonarea se face după numărul de studenţi îndrumaţi de fiecare tutor:

```
SELECT NUME, CODS, AN
FROM STUD S
WHERE CODS = 24
ORDER BY (SELECT COUNT(*) FROM STUD
WHERE TUTOR = S.MATR) DESC;
```

SUBCERERI PE SELECT

- Ca şi în cazul celor din ORDER BY aceste subcereri trebuie să întoarcă o singură valoare.
- Subcererea poate fi necorelată (în acest caz pe coloana respectivă din rezultat vom avea aceeaşi valoare) sau corelată.

EXEMPLU

```
SELECT NUME, (SELECT COUNT(*)
              FROM STUD
              WHERE TUTOR = S.MATR) NUMAR
FROM STUD S
WHERE CODS = 24
      AND 1 <= (SELECT COUNT(*)
                FROM STUD
                WHERE TUTOR = S.MATR)
ORDER BY (SELECT COUNT(*)
          FROM STUD
          WHERE TUTOR = S.MATR) DESC
```

UNION INTERSET si MINUS

- Sintaxa:

```
Subcerere
UNION | INTERSECT | MINUS
Subcerere
```

REGULI

- Ambele cereri întorc acelaşi număr de coloane.
- Coloanele corespondente au valori de acelaşi tip sau de tipuri pentru care sistemul poate face automat conversia.
- Operatorii pot fi folosiţi repetat şi succesiv pentru a forma expresii. În acest caz, dacă nu se folosesc paranteze pentru a schimba ordinea de evaluare, ei se execută în ordinea în care apar în expresie.
- Capul de tabel al rezultatului este cel dat de prima cerere din expresie.
- Cererile implicate în aceste operaţii nu pot conţine clauza ORDER BY. Aceasta poate să fie pusă doar la sfârşitul expresiei şi poate conţine nume de coloane din rezultat sau numerele de ordine ale acestora.
- Rezultatul nu conţine linii duplicat. Acestea sunt eliminate chiar dacă provin din aceeaşi cerere - operand al expresiei.

EXEMPLU

```
SELECT NUME, CODS, LOC, PUNCTAJ
FROM STUD
WHERE CODS = 11 AND PUNCTAJ > 2000
  UNION
SELECT NUME, CODS, LOC, PUNCTAJ
FROM STUD
WHERE CODS = 21 AND LOC = 'BUCURESTI'
  UNION
SELECT NUME, CODS, LOC, PUNCTAJ
FROM STUD
WHERE CODS = 24 AND PUNCTAJ >1500
  INTERSECT
SELECT NUME, CODS, LOC, PUNCTAJ
FROM STUD
WHERE PUNCTAJ >= 700
ORDER BY LOC DESC, 4;
```

Bibliografie

1. **Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer D. Widom**: *Database Systems: The Complete Book*, Prentice-Hall, Englewood Cliffs, NJ, 2002.
2. **F. Rădulescu** : *Oracle SQL, PL/SQL*, Editura Printech, ISBN 973-718-203-02005