CAPITOLUL 9

SUBCERERI

In acest capitol se vor prezenta alte caracteristici ale declaratiei SELECT, cum ar fi cereri continute in clauza WHERE sau HAVING ale altei decla- ratii SQL.

Subcereri ascunse

O subcerere este o declaratie SELECT care este ascunsa in interiorul altei declaratii SELECT si care intoarce rezultate intermediare.

De exemplu:

```
SELECT column1, column2, ...
FROM table
WHERE column =
               (SELECT column
                FROM table
                WHERE condition)
```

Subcererea este adesea referita ca un subselect sau ca un SELECT interior; in general, subcererea se executa prima si rezultatul sau este folosit pentru a completa conditia cererii principale sau a cererii externe.Folosirea sub- cererilor permite consruirea de comenzi puternice pornind de la unele simple. Cererea ascunsa poate fi foarte folositoare cind este nevoie sa fie selectate linii dintr-o tabela printr-o conditie ce depinde de date din propria tabela.

Subcereri ce intorc o linie

Pentru a gasi angajatul care cistiga salariul minim din companie (salariul minim este o cantitate necunoscuta), trebuie parcursi doi pasi:

1. Gasirea salariului minim:

```
2.
3.        SELECT MIN(SAL)
4.        FROM EMP;
5.
6.        MIN(SAL)
7.       ----------
8.          800
```

9. Gasirea angajatului care cistiga salariul minim:

```
10.
11.
12.       SELECT ENAME, JOB, SAL
```

```
13.       FROM EMP
14.       WHERE SAL = (cel mai mic salariu care este cunoscut)
```

Putem combina cele doua cereri ca o subcerere ascunsa:

```
SELECT ENAME, JOB, SAL
FROM EMP
WHERE SAL = (SELECT MIN(SAL)
             FROM EMP);

ENAME     JOB       SAL
-------   ------    -----
SMITH     CLERK      800
```

Cum sint prelucrate subcererile ascunse ?

O declarartie SELECT poate fi considerata ca un bloc de cerere.Exemplul de mai sus consta din doua blocuri de cerere - cererea principala si cererea in- terioara.

Declaratia SELECT interioara sau blocul de cerere este executata prima, producind un rezultat al cererii: 800.Blocul de cerere principal este apoi prelucrat si foloseste valoarea intoarsa de cererea interioara pentru a com- pleta conditia de cautare.Deci, cererea principala in final va arata in felul urmator:

```
SELECT ENAME, SAL, DEPTNO
FROM EMP
WHERE SAL = 800;
```

In exemplul de mai sus, 800 este o valoare unica.Subcererea care a intors valoarea 800 se numeste subcerere ce intoarce o singura linie.Cind o subcerere intoarce doar o linie, va fi folosit un operator logic sau un operator de com- paratie.De exemplu: =, <, >, <=, etc.

Pentru a gasi toti angajatii ce au aceeasi functie ca BLAKE, vom introduce:

```
SELECT ENAME, JOB
FROM EMP
WHERE JOB = (SELECT JOB
             FROM EMP
             WHERE ENAME = 'BLAKE');

ENAME       JOB
-------     -----
JONES     MANAGER
BLAKE     MANAGER
CLARK     MANAGER
```

Cererea interioara intoarce functia lui BLAKE, care este folosita in partea dreapta a clauzei WHERE di cererea principala (dupa operatorul de comparatie).

Subcereri ce intorc mai mult de o linie

Urmatoarea cerere gaseste angajatii care au salariul egal cu salariul minim din fiecare departament:

```
SELECT ENAME, SAL, DEPTNO
FROM EMP
WHERE SAL IN
             (SELECT MIN(SAL)
              FROM EMP
              GROUP BY DEPTNO);
```

```
 ENAME        SAL        DEPTNO
-------      -----      ---------
SMITH         800          20
JAMES         950          30
MILLER       1300          10
```

Observati ca, cererea interioara are clauza GROUP BY.Aceasta inseamna ca va intoarce mai mult decit o valoare.Deci este nevoie sa folosim un operator de comparatie multi linie.In acest caz, operatorul IN trebuie sa fie folosit, deoarece rezulta o lista de valori.

Rezultatul obtinut nu arata si departamentul in care lucreaza angajatul.
Deoarece se compara doar valorile salariilor, cererea interioara poate intoar- ce o valoare simpla, pentru ca el cauta cel mai mic salariu pentru unul din departamente, nu in mod necesar departamentul propriu al angajatului.Prin urmare, cererea poate fi rescrisa pentru a gasi combinatia salariul angaja- tului si numarul departamentului cu salariul minim si numarul departamentului:

Compararea a mai multor valori

Urmatoarea cerere ggaseste angajatii care cistiga salariul cel mai mic din departamentul lor:

```
SELECT ENAME, SAL, DEPTNO
FROM EMP
WHERE (SAL, DEPTNO) IN
                         (SELECT MIN(SAL), DEPTNO
                          FROM EMP
                          GROUP BY DEPTNO);
```

```
 ENAME       SAL      DEPTNO
-------     -----    ---------
 SMITH       800        20
 JAMES       950        30
 MILLER     1300        10
```

Cererea de mai sus compara o pereche de coloane.

Observatie : coloanele din partea stinga a conditiei de cautare sint in pa- ranteze si fiecare coloana este separata printr-o virggula.

Coloanele listate in clauza SELECT a subcererii trebuie sa fie in aceeasi ordine ca si lista coloanelor dintre paranteze din clauza WHERE a cererii externe.

Coloanele intoarse de cererea interioara trebuie, de asemenea sa se potri- veasca in numar si tip de date cu coloanele cu care ele sint comparate in cererea externa.

De exemplu :

```
...WHERE (numcolumn, charcolumn) =
         (SELECT datecolumn, numcolumn, charcolumn ...
```

nu este permis.

Erori intilnite

Cind o subcerere intoarce mai mult decit o linie si este folosit un opera- tor de comparatie pt o singura linie, SQL*Plus da urmatorul mesaj de eroare:

```
SELECT ENAME, SAL, DEPTNO
FROM EMP
WHERE SAL = (SELECT MIN(SAL)
             FROM EMP
             GROUP BY DEPTNO);
```

Eroare : subcerere ce trebuie sa intoarca o singura linie intoarce mai mult decit o linie.

Nu este selectata nici o inregistrare.

Daca cererea interioara nu intoarce nici o linie, va fi dat urmatorul mesaj de eroare :

```
SELECT ENAME, JOB
FROM EMP
WHERE JOB = (SELECT JOB
             FROM EMP
             WHERE ENAME = 'SMITHE');
```

Eroare : subcerere ce trebuie sa intoarca o singura linie nu intoarce nici o linie.

Nu este selectata nici o inregistrare.

Operatorii SOME/ANY, ALL

Operatorii ANY sau ALL pot fi folositi pentru subcererile care intorc mai mult de o linie.Ei sint folositi in clauzele WHERE sau HAVING in legatura cu operatorii logici (=, !=, <, >, >=, <+).
ANY (sau sinonimul sau SOME) compara o valoare cu fiecare valoare intoarsa de o subcerere.

Pentru a afisa angajatii care cistiga mai mult decit cel mai mic salariu din departamentul 30, introducem :

```
SELECT ENAME, SAL, JOB, DEPTNO
FROM EMP
WHERE SAL > SOME (SELECT DISTINCT SAL
                  FROM EMP
                  WHERE DEPTNO = 30)
ORDER BY SAL DESC;

ENAME       SAL        JOB        DEPTNO
-------     -----      -----      --------
KING        5000     PRESIDENT       10
SCOTT       3000     ANALYST         20
FORD        3000     ANALYST         20
JONES       2975     MANAGER         20
BLAKE       2850     MANAGER         30
CLARK       2450     MANAGER         10
ALLEN       1600     SALESMAN        30
MILLER      1300     CLERK           10
WARD        1250     SALESMAN        30
ADAMS       1100     CLERK           20
```

Cel mai mic salariu din departamentul 30 este 950$ (a lui James).Cererea principala intoarce angajatii care cistiga un salariu mai mare ca salariul minim din departamentul 30.Asa ca '> ANY' inseamna mai mare ca minim.'=ANY' este echivalent cu IN.

Cind se foloseste SOME/ANY, DISTINCT este frecvent folosit pentru a impie- dica sa se selecteze linniile de mai multe ori.
ALL compara o valoare cu fiecare valoare intoarsa de o subcerere.
Urmatoarea cerere gaseste angajatii care cistiga mai mult ca fiecare angajat din departamentul 30 :

```
SELECT ENAME, SAL, JOB, DEPTNO
FROM EMP
WHERE SAL > ALL (SELECT DISTINCT SAL
                 FROM EMP
                 WHERE  DEPTNO = 30)
ORDER BY SAL DESC;

ENAME       SAL          JOB        DEPTNO
-------     -----        -----      ---------
KING        5000       PRESIDENT       10
SCOTT       3000       ANALYST         20
FORD        3000       ANALYST         20
JONES       2975       MANAGER         20
```

Cel mai mare salariu din departamentul 30 este 250$ (a lui Blake), asa ca cererea intoarce acei angajati ai caror salariu este mai mare ca salariul maxim din departamentul 30, prin urmare mai mare ca fiecare salariu din de- partament.

Operatorul NOT poate fi folosit cu IN, ANY sau ALL.

Clauza HAVING cu subcereri ascunse

Subcererile ascunse pot fi folosite de asemenea in clauza HAVING.

(Observatie : clauza WHERE se refera la o singura linie si clauza HAVING la grupuri de linii specificate in clauza GROUP BY.)

De exemplu, pentru a afisa departamentul(ele) care au un salariu mediu mai mare ca departamentul 30, introducem :

```
SELECT DEPTNO, AVG(SAL)
FROM EMP
HAVING AVG(SAL) > (SELECT AVG(SAL)
                   FROM EMP
                   WHERE DEPTNO = 30)
GROUP BY DEPTNO;

 DEPTNO      AVG(SAL)
--------    ----------
  10         2916.66667
  20         2175
```

Pentru a construi o cerere care gaseste functia cu cel mai mare salariu mediu, introducem :

```
SELECT JOB, AVG(SAL)
FROM EMP
GROUP BY JOB
HAVING AVG(SAL) = (SELECT MAX(AVG(SAL))
                   FROM EMP
                   GROUP BY JOB);

 JOB          AVG(SAL)
-----        ----------
 PRESIDENT     5000
```

Mai intii cererea interioara gaseste salariul mediu pentru fiecare grup de functii diferit si functia MAX alege cel mai mare salariu mediu.Acea valoare (5000) este folosita in clauza HAVING.Clauza GROUP BY din cererea principala este necesara pentru ca lista ce urmeaza dupa SELECT-ul din cererea principala contine atit o coloana agregat cit si o coloana non-agragat.

Ordonarea datelor cu subcereri

Nu poate exista o clauza ORDER BY intr-o subcerere.

Regula este ca poate exista doar o singura clauza ORDER BY pentru o decla- ratie SELECT si, daca este specificata, trebuie sa fie ultima clauza din co- manda SELECT.

Subcereri ascunse

Subcererile pot fi ascunse (folosite in interiorul unei subcereri) :

Afisati numele, functia si data angajarii pentru angajatii al caror salariu este mai mare ca cel mai mare salariu din orice departament de vinzari.

```
SELECT ENAME, JOB, HIREDATE, SAL
FROM EMP
WHERE SAL > (SELECT MAX(SAL)
             FROM EMP
             WHERE DEPTNO = (SELECT DEPTNO
                             FROM DEPT
                             WHERE DNAME = 'SALES'));
```

ENAME	JOB	HIREDATE	SAL
JONES	MANAGER	02-APR-81	2975
SCOTT	ANALYST	09-DEC-82	3000
KING	PRESIDENT	17-NOV-81	5000
FORD	ANALYST	03-DEC-81	3000

Limitele de imbricare

Limita nivelelor de imbricare pentru o subcerere este 255.

Reguli de scriere a cererilor

- cererea interioara trebuie sa fie inclusa intre paranteze si trebuie sa fie in partea dreapta a conditiei.
- subcererile nu pot avea clauza ORDER BY.
- clauza ORDER BY apare la sfirsitul declaratiei SELECT principale.
- coloanele multiple din lista din SELECT a cererii interioare trebuie sa fie in aceeasi ordine ca si coloanele ce apar in conditia din clauza cere- rii principale.De asemenea mai trebuie sa corespunda si tipul si numarul coloanelor listate.
- subcererile sint intotdeauna executate de la cea mai adinca imbricare pina la nivelul principal de imbricare, daca nu sint subcereri corelate (acestea vor fi discutate mai tirziu).
- pot fi folositi operatorii logici si SQL la fel de bine ca si ANY si ALL.
- subcererile pot
    - intoarce una sau mai multe linii;
    - intoarce una sau mai multe coloane;
    - folosi GROUP BY sau functii de grup;
    - fi folosite inlantuite cu predicate multiple AND sau OR in aceesi cerere externa.
    - uni tabele.
    - recupera dintr-o tabela diferita de cea a cererii exterioare.
    - apare in declaratii SELECT, UPDATE, DELETE, INSERT, CREATE TABLE.
    - fi corelate cu o cerere exterioara.
    - folosi operatori de multimi.

Subcereri corelate

O subcerere corelata este o subcerere care este executata o data pentru fiecare linie candidat considerata de cererea principala si care la executie foloseste o valoare dintr-o coloana din cererea exterioara.Aceasta determina ca subcererea corelata sa fie prelucrata intr-un mod diferit de subcererea ascunsa obisnuita.

O subcerere corelata este identificata prin folosirea unei coloane a cererii exterioare in clauza predicatului cererii interioare.

Cu o subcerere ascunsa obisnuita, SELECT-ul interior ruleaza primul si se executa o data, intorcind valori ce vor fi folosite de cererea principala.Pe de alta parte, o subcerere corelata se executa o data pentru fiecare linie candidat considerata de cererea externa.Cererea interioara este dirijata de cererea externa.

Pasii de executie ai unei subcereri corelate :

1. Se obtine linia candidat. (obtinuta de cererea exterioara)
2. Se executa cererea interioara folosind valoarea liniei candidat.
3. Se folosesc valorile rezultate din cererea interioara pentru a pastra sau pentru a inlatura linia candidat.
4. Se repeta pina nu mai ramine nici o linie candidat.

Desi subcererea corelata se executa repetat, o data pentru fiecare linie in cererea principala, aceasta nu inseamna ca subcererile corelate sint mai putin eficiente ca subcererile necorelate obisnuite.Se va vorbi despre eficienta mai tirziu in acest capitol.

Putem folosi o subcerere corelata pentru a gasi angajatii care cistiga un salariu mai mare ca salariul mediu al departamentului lor :

```
SELECT EMPNO, ENAME, SAL, DEPTNO
FROM EMP E
WHERE SAL > (SELECT AVG(SAL)
             FROM EMP
             WHERE DEPTNO = E.DEPTNO)
ORDER BY DEPTNO;
```

EMPNO	ENAME	SAL	DEPTNO
7839	KING	5000	10
7566	JONES	2975	20
7788	SCOTT	3000	20
7902	FORD	3000	20
7499	ALLEN	1600	30
7698	BLAKE	2850	30

Putem observa imediat ca este o cerere corelata pentru ca am folosit o co- loana din SELECT-ul extern in clauza WHERE din SELECT-ul interior.

Observati ca alias-ul este necesar doar pentru a indeparta ambiguitatea pentru numele coloanelor.

Sa analizam exemplul de mai sus folosind tabela EMP :

Cererea principala

1. Se selecteaza prima linie candidat - Smith din departamentul 20 cistiga 800 œ.
2. EMP in clauza FROM are alias-ul E care obtine coloana DEPTNO referita in clauza WHERE a cererii interioare.
3. Clauza WHERE compara 800 cu valoarea intoarsa de cererea interioara.

Cererea interioara

4. Calculeaza AVG(SAL) pentru departamentul angajatului.
5. Valoarea departamentului din clauza WHERE este departamentul candidatului (E.DEPTNO), valoare transmisa cererii interioare din coloana DEPTNO a cererii exterioare.
6. AVG(SAL) pentru departamentul lui Smith - 20 - este 2175 œ.
7. Linia candidat nu indeplineste conditia, asa ca este indepartata.
8. Se repeta de la pasul 1 pentru urmatoarea linie candidat; ALLEN din departamentul 30 cistiga 1600 œ.

Selectia liniilor candidat continua cu verificarea conditiei ce apare in re- zultatul cererii.

Observatie : o subcerere corelata este semnalata de un nume de coloana, un nume de tabela sau un alias de tabela in clauza WHERE, care se refera la va- loarea unei coloane in fiecare linie candidat din SELECT-ul exterior.De aseme- nea subcererea corelata se executa repetat pentru fiecare linie candidat in cererea principala.

Comenzile UPDATE pot contine subcereri corelate :

```
UPDATE EMP E
SET (SAL, COMM) = (SELECT AVG(SAL) * 1.1, AVG(COMM)
                   FROM EMP
                   WHERE DEPTNO = E.DEPTNO)
    HIREDATE = '11-JUN-85';
```

Operatori

Cind se folosesc declaratii SELECT ascunse, operatorii logici, precum si ANY si ALL sint toti valizi.In plus poate fi folosit operatorul EXISTS.

Operatorul EXISTS

Operatorul EXISTS este frecvent folosit cu subcererile corelate.El testeaza daca o valoare exista (NOT EXISTS specifica daca ceva nu exista).Daca valoarea exista se intoarce TRUE; daca valoarea nu exista se intoarce FALSE.

Pentru a gasi angajatii ce au cel putin un subordonat, introducem :

```
SELECT EMPNO, ENAME, JOB, DEPTNO
```

```
FROM EMP E
WHERE EXISTS (SELECT EMPNO
              FROM EMP
              WHERE EMP.MGR = E.EMPNO)
ORDER BY EMPNO;
```

EMPNO	ENAME	JOB	DEPTNO
7566	JONES	MANAGER	20
7698	BLAKE	MANAGER	30
7782	CLARK	MANAGER	10
7788	SCOTT	ANALYST	20
7839	KING	PRESIDENT	10
7902	FORD	ANALYST	20

Sa gasim toti angajatii al caror departament nu este in tabela DEPT :

```
SELECT EMPNO, ENAME, DEPTNO
FROM EMP
WHERE NOT EXISTS (SELECT DEPTNO
                  FROM DEPT
                  WHERE DEPT.DEPTNO = EMP.DEPTNO);
```

Nu va fi selectata nici o inregistrare.

Alt mod de a gasi departamentul care nu are nici un angajat este :

```
SELECT DEPTNO, DNAME
FROM DEPT D
WHERE NOT EXISTS (SELECT 1
                  FROM EMP E
                  WHERE E.DEPTNO = D.DEPTNO);
```

DEPTNO	DNAME
40	OPERATIONS

Observati ca SELECT-ul interior nu este necesar sa intoarca o valoare spe- cifica, asa ca poate fi selectata o cifra.

De ce sa folosim o subcerere corelata ?

Subcererea corelata este un mod de a 'citi' fiecare linie din tabela si de a compara valorile din fiecare linie cu datele inrudite.Aceasta este folosita oricind o subcerere trebuie sa intoarca un rezultat diferit sau o multime de rezultate pentru fiecare linie candidat considerata de cererea principala.Cu alte cuvinte, o subcerere corelata este folosita pentru a raspunde la intre- bari cu mai multe subpuncte al caror raspuns depinde de valoarea din fiecare linie din cererea parinte.

SELECT-ul interior este executat normal o data pentru fiecare linie candi- dat.

Considerente de eficienta

Vom examina cele doua tipuri de subcereri.Trebuie mentionat ca subcererea corelata (cu EXISTS) poate fi cel mai performant mod pentru unele cereri.

Performanta va depinde de folosirea indexarilor, de numarul liniilor in- toarse de cereri, de dimensiunea tabelelor si daca sint necesare tabelele temporare pentru a evalua rezultatele temporare.Tabelele temporare generate de ORACLE nu sint indexate si acest lucru conduce la degradarea performantelor pentru subcererile ce folosesc IN, ANY si ALL.

Cele de mai sus sint puncte de vedere generale.Performantele sint discutate mai in detaliu in alte cursuri.

NOT EXISTS contra NOT IN

Desi intr-o subcerere o operatie NOT IN poate fi la fel de eficienta ca si NOT EXISTS, NOT EXISTS este mult mai sigur daca subcererea intoarce niste valori NULL, fata de de NOT IN pentru care conditia se evalueaza la FALSE cind in lista de comparatii sint incluse valori NULL.

Considerind urmatoarea cerere care gaseste angajatii ce nu au nici un su- bordonat :

```
SELECT ENAME, JOB
FROM EMP
WHERE EMPNO NOT IN (SELECT MGR
                    FROM EMP);
```

Nici o linie nu va fi intoarsa de cererea de mai sus, deoarece coloana MGR contine o valoare NULL.

Cererea corecta este :

```
SELECT ENAME, JOB
FROM EMP E
WHERE NOT EXISTS (SELECT MGR
                  FROM EMP
                  WHERE MGR = E.EMPNO);

ENAME       JOB
-------     -----
SMITH       CLERK
ADAMS       CLERK
ALLEN       SALESMAN
WARD        SALESMAN
MARTIN      SALESMAN
TURNER      SALESMAN
JAMES       CLERK
MILLER      CLERK
```

Exercitii - Subcereri

Aceste exercitii va vor permite sa scrieti cereri complete folosind SELECT- uri ascunse si SELECT-uri corelate.

Tema

1. Gasiti angajatii care cistiga cell mai mare salariu pentru fiecare tip de functie.Sortati in ordinea descrescatoare a salariului.

```
        JOB         ENAME          SAL
       -----       -------        ------
       PRESIDENT    KING         5,000.00
       ANALYST      SCOTT        3,000.00
       ANALYST      FORD         3,000.00
       MANAGER      JONES        2,975.00
       SALESMAN     ALLEN        1,600.00
       CLERK        MILLER       1,300.00

       Vor fi 6 inregistarri selectate.
```

12. Gasiti angajatii care cistiga salariul minim pentru functia lor.Afisati rezultatul in ordine crescatoare a salariului.

```
       ENAME        JOB           SAL
      -------      -----         -----
       SMITH       CLERK         800.00
       WARD        SALESMAN     1,250.00
       MARTIN      SALESMAN     1,250.00
       CLARK       MANAGER      2,450.00
       SCOTT       ANALYST      3,000.00
       FORD        ANALYST      3,000.00
       KING        PRESIDENT    5,000.00

       Vor fi 7 inregistari selectate.
```

24. Gasiti cei mai recenti angajati din fiecare departament.Ordonati dupa data angajarii.

```
       DEPTNO     ENAME     HIREDATE
      --------   -------   ----------
         20       ADAMS     04-JUN-84
         10       KING      09-JUL-84
         30       JAMES     23-JUL-B4
```

31. Afisati urmatoarele detalii pentru orice angajat care cistiga un salariu mai mare ca media pentru departamentul lor.Sortati dupa numarul departa- mentului.

```
       ENAME      SALARY      DEPTNO
      -------    --------    --------
       KING        5000         10
       JONES       2975         20
       SCOTT       3000         20
       FORD        3000         20
       ALLEN       1600         30
       BLAKE       2850         30
```

```
40.
41.       Vor fi selectate 6 inregistrari.
```

42. Listati toate departamentele care nu au angajati (folositi o subcerere).

```
43.       DEPTNO       DNAME
44.      --------     -------
45.         40       OPERATIONS
```

46. Afisati urmatoarele informatii pentru departamentul cu cea mai mare nota de plata anuala.

```
47.       DEPTNO       COMPENSATION
48.      --------     ---------------
49.         20           130500
```

50. Care sint primii trei angajati in functie de salariul cistigat ? Afisati numele lor si salariul.

```
51.
52.
53.        ENAME          SAL
54.       -------        -----
55.        SCOTT        3,000.00
56.        KING         5,000.00
57.        FORD         3,000.00
```

58. In ce an s-au angajat cei mai multi in companie ? Afisati anul si numarul angajatilor.

```
59.       YEAR        NUMBER OF EMPS
60.      ------      ----------------
61.       1984               8
```

62. Modificati intrebarea 4 pentru a afisa si salariul mediu pentru departa- ment.

```
63.       ENAME         SAL       DEPTNO       DEPTAVG
64.      -------       -----     --------     ----------
65.       ALLEN         1600        30        1566.66667
66.       BLAKE         2850        30        1566.66667
67.       JONES         2975        20              2175
68.       FORD          3000        20              2175
69.       SCOTT         3000        20              2175
70.       KING          5000        10        2916.66667
71.
72.       Vor fi selectate 6 inregistrari.
```

73. Scrieti o subcerere care afiseaza o '*' linga linia celui mai recent angajat.Afisati ENAME, HIREDATE si coloana (maxdate) completata cu '*'.

```
74.
75.        ENAME        HIREDATE      MAXDATE
76.       -------      ----------    ---------
77.        SMITH        13-JUN-83
78.        ADAMS        04-JUN-84
79.        ALLEN        15-AUG-83
80.        BLAKE        11-JUN-84
81.        CLARK        14-MAY-84
82.        FORD         05-DEC-83
83.        JAMES        23-JUL-84        *
84.        JONES        31-OCT-83
85.        KING         09-JUL-84
86.        MARTIN       05-DEC-83
87.        MILLER       21-NOV-83
88.        SCOTT        05-MAR-84
89.        TURNER       04-JUN-84
90.        WARD         26-MAR-84
91.
92.        Vor fi selectate 14 inregistrari.
```

Subcereri

```
1.        SELECT JOB, ENAME, SAL
2.        FROM EMP
3.        WHERE (SAL, JOB) IN (SELECT MAX(SAL), JOB
4.                             FROM EMP
5.                             GROUP BY JOB)
6.        ORDER BY SAL DESC;
7.
8.        SELECT ENAME, JOB, SAL
9.        FROM EMP
10.       WHERE (SAL, JOB) IN (SELECT MIN(SAL), JOB
11.                            FROM EMP
12.                            GROUP BY JOB)
13.       ORDER BY SAL;
14.
15.       SELECT DEPTNO, ENAME, HIREDATE
16.       FROM EMP
17.       WHERE (HIREDATE, DEPTNO) IN (SELECT MAX(HIREDATE),
   DEPTNO
18.                                    FROM EMP
19.                                    GROUP BY DEPTNO)
20.       ORDER BY HIREDATE;
21.

22.       SELECT ENAME, SAL SALARY, DEPTNO
23.       FROM EMP E
24.       WHERE SAL > (SELECT AVG(SAL)
25.                    FROM EMP
26.                    WHERE DEPTNO = E.DEPTNO)
27.       ORDER BY DEPTNO;
28.
29.       SELECT DEPTNO, DNAME
30.       FROM DEPT D
31.       WHERE NOT EXISTS (SELECT 'anything'
32.                         FROM EMP
33.                         WHERE DEPTNO = D.DEPTNO);
```

```
34.
35.       DEFINE REM = SAL * 12 + NVL(COMM, 0)
36.       SELECT DEPTNO, SUM(&REM) COMPENSATION
37.       FROM EMP
38.       GROUP BY DEPTNO
39.       HAVING SUM(&REM) = (SELECT MAX(SUM(&REM))
40.                           FROM EMP
41.                           GROUP BY DEPTNO);
42.
43.       SELECT ENAME, SAL
44.       FROM EMP E
45.       WHERE 3 > (SELECT COUNT(*)
46.                  FROM EMP
47.                  WHERE E.SAL < SAL);
48.
49.       SELECT TO_CHAR(HIREDATE, 'YYYY') YEAR,
50.              COUNT(EMPNO) NUMBER_OF_EMPS
51.       FROM EMP
52.       GROUP BY TO_CHAR(HIREDATE, 'YYYY')
53.       HAVING COUNT(EMPNO) = (SELECT MAX(COUNT(EMPNO))
54.                              FROM EMP
55.                              GROUP BY TO_CHAR(HIREDATE,
   'YYYY'));
56.
57.       SELECT E.ENAME ENAME, E.SAL, E.DEPTNO, AVG(A.SAL)
   DEPT_AVG
58.       FROM EMP A, EMP E
59.       WHERE E.DEPTNO = A.DEPTNO
60.       AND E.SAL > (SELECT AVG(SAL)
61.                    FROM EMP
62.                    WHERE DEPTNO = E.DEPTNO)
63.       GROUP BY E.ENAME, E.SAL, E.DEPTNO
64.       ORDER BY AVG(A.SAL);
65.
66.        SELECT ENAME, HIREDATE, '*' MAXDATE
67.        FROM EMP
68.        WHERE HIREDATE = (SELECT MAX(HIREDATE) FROM EMP)
69.        UNION
70.        SELECT ENAME, HIREDATE, ' '
71.        FROM EMP
72.        WHERE HIREDATE <> (SELECT MAX(HIREDATE) FROM EMP)
73.
```