Universitatea Constantin Brâncuși din Târgu-Jiu
Automatică și Informatică Aplicată

Baze de date

Limbajul SQL

Adrian Runceanu

copyright@www.adrian.runceanu.ro

THE INFORMATION COMPANY

Curs 6

Limbajul SQL

copyright@www.adrian.runceanu.ro

Limbajul SQL

6.1. Subinterogări (Subqueries)

6.1.1. SINGLE ROW SUBQUERIES

6.1.2. MULTIPLE ROW SUBQUERIES

copyright@www.adrian.runceanu.ro

6. 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**.

copyright@www.adrian.runceanu.ro

6. SUBQUERIES (Subinterogari)

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

copyright@www.adrian.runceanu.ro

6. SUBQUERIES (Subinterogari)

Sintaxa generală:

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

copyright@www.adrian.runceanu.ro

6. SUBQUERIES (Subinterogari)

Reguli de folosire a subinterogarilor

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

copyright@www.adrian.runceanu.ro

6. 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.

- Un subquery 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.

copyright@www.adrian.runceanu.ro

6. 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.

copyright@www.adrian.runceanu.ro

Limbajul SQL

6. SUBQUERIES (Subinterogări)

6.1. SINGLE ROW SUBQUERIES

6.2. MULTIPLE ROW SUBQUERIES

copyright@www.adrian.runceanu.ro

6.1. SINGLE ROW SUBQUERIES

Single row-subquery

```
SELECT ename
FROM EMP
WHERE sal  >
              ( SELECT sal
                FROM EMP
                WHERE ename = 'SMITH' );
```

Aflati numele angajatilor care au salariul mai mare decat angajatul care se numeste SMITH.

copyright@www.adrian.runceanu.ro

6.1. SINGLE ROW SUBQUERIES

Single row-subquery

copyright@www.adrian.runceanu.ro

6.1. 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.

copyright@www.adrian.runceanu.ro

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 ename, job
FROM  EMP
WHERE job =
        (SELECT job
         FROM EMP
         WHERE empno = 7369)
        AND  sal >
                        ( SELECT  sal
                          FROM    EMP
                          WHERE  empno = 7876 );
```

ENAME	JOB
MILLER	CLERK

copyright@www.adrian.runceanu.ro

6.1. 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 1300.
- 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 1300), astfel ca aceasta instructiune SQL este denumita o *subinterogare single-row*.

copyright@www.adrian.runceanu.ro

6.1. SINGLE ROW SUBQUERIES

Language SQL Rows 15 Clear Command Find Tables

```
1  SELECT ename, job
2  FROM  EMP
3  WHERE job =
4             (SELECT job
5              FROM EMP
6              WHERE empno = 7369)
7             AND  sal >
8                ( SELECT  sal
9                  FROM  EMP
10                 WHERE empno = 7876 );
11
```

Results Explain Describe Saved SQL History

ENAME	JOB
MILLER	CLERK

1 rows returned in 0.01 seconds Download

copyright@www.adrian.runceanu.ro

6.1. SINGLE ROW SUBQUERIES

```
SELECT ename, job, sal, deptno
FROM EMP
WHERE job =
                    (SELECT job
                     FROM EMP
                     WHERE empno = 7934 )
AND deptno =
                    (SELECT deptno
                     FROM DEPT
                     WHERE dname = 'ACCOUNTING');
```

copyright@www.adrian.runceanu.ro

6.1. SINGLE ROW SUBQUERIES

```
1  SELECT ename, job, sal, deptno
2  FROM EMP
3  WHERE job =
4                (SELECT job
5                 FROM EMP
6                 WHERE empno = 7934 )
7  AND deptno =
8                (SELECT deptno
9                 FROM dept
10                WHERE dname = 'ACCOUNTING');
11
```

Results | Explain | Describe | Saved SQL | History

ENAME	JOB	SAL	DEPTNO
MILLER	CLERK	1300	10

copyright@www.adrian.runceanu.ro

6.1. 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);
```

copyright@www.adrian.runceanu.ro

6.1. 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.

Language SQL Rows 15 Clear Command Find Tables

```
1  SELECT ename, sal, job
2  FROM EMP
3  WHERE sal <
4              ( SELECT MAX(sal)
5                FROM EMP );
6
```

Results Explain Describe Saved SQL History

ENAME	SAL	JOB
BLAKE	2850	MANAGER
CLARK	2450	MANAGER
JONES	2975	MANAGER
SCOTT	3000	ANALYST
FORD	3000	ANALYST
SMITH	800	CLERK
ALLEN	1600	SALESMAN
WARD	1250	SALESMAN
MARTIN	1250	SALESMAN
TURNER	1500	SALESMAN
ADAMS	1100	CLERK
JAMES	950	CLERK
MILLER	1300	CLERK

13 rows returned in 0.00 seconds Download

copyright@www.adrian.runceanu.ro

6.1. 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 deptno, MIN(sal)
FROM EMP
GROUP BY deptno
HAVING MIN(sal) >
                    ( SELECT MIN(sal)
                      FROM EMP
                      WHERE deptno = 20 );
```

copyright@www.adrian.runceanu.ro

6.1. 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.

Language SQL Rows 15 Clear Command Find Tables

```
1  SELECT deptno, MIN(sal)
2  FROM EMP
3  GROUP BY deptno
4  HAVING MIN(sal) >
5                 (SELECT MIN(sal)
6                  FROM EMP
7                  WHERE deptno = 20);
8
```

Results Explain Describe Saved SQL History

DEPTNO	MIN(SAL)
30	950
10	1300

2 rows returned in 0.01 seconds Download

copyright@www.adrian.runceanu.ro

6.1. SINGLE ROW SUBQUERIES

Aplicatii rezolvate

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 7698?

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

copyright@www.adrian.runceanu.ro

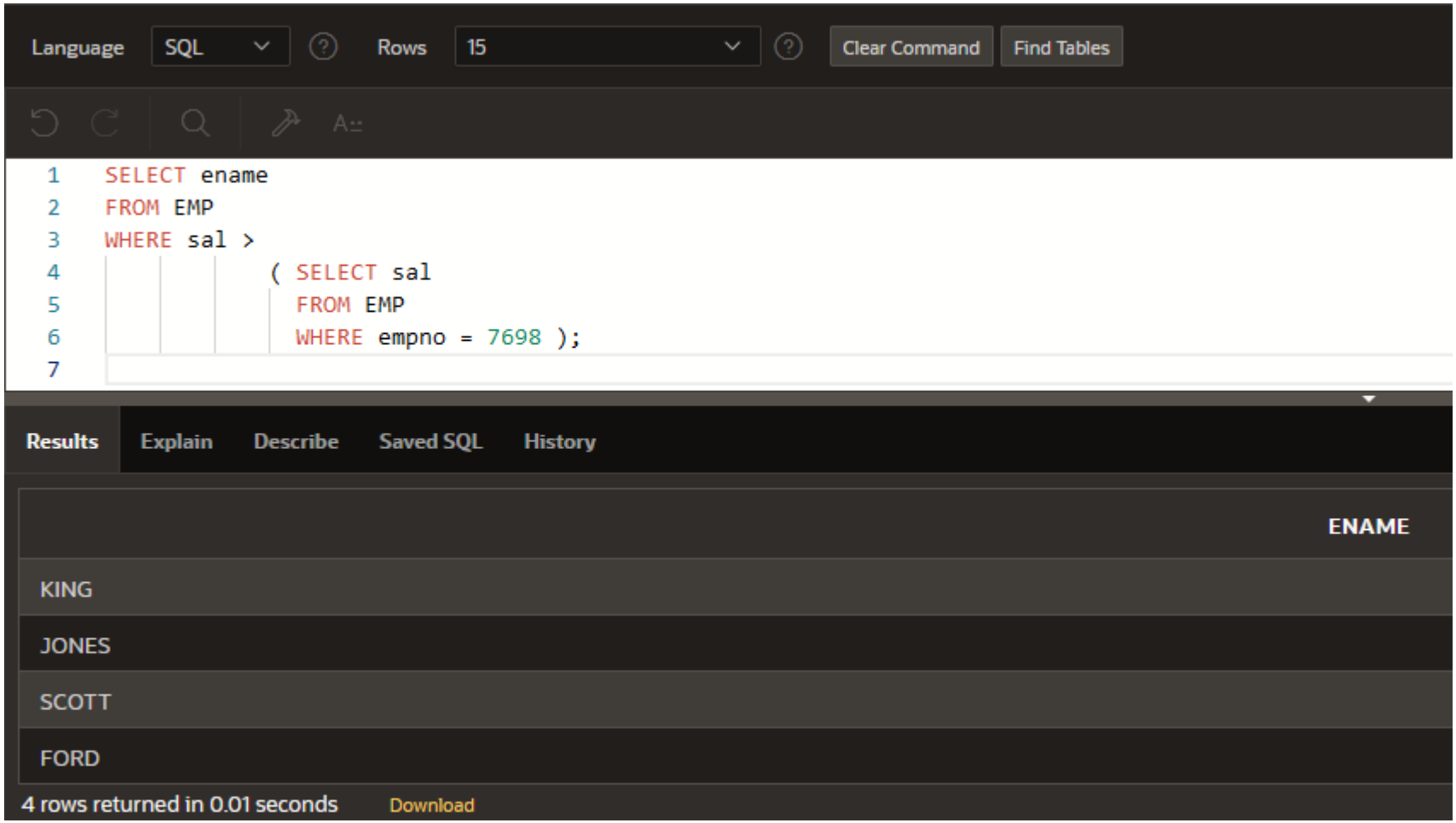
6.1. SINGLE ROW SUBQUERIES

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 7698?

```
SELECT ename
FROM EMP
WHERE sal >
                ( SELECT sal
                  FROM EMP
                  WHERE empno = 7698 );
```

copyright@www.adrian.runceanu.ro

6.1. SINGLE ROW SUBQUERIES



copyright@www.adrian.runceanu.ro

6.1. SINGLE ROW SUBQUERIES

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

```
SELECT ename, sal
FROM EMP
WHERE deptno =
                    ( SELECT deptno
                      FROM DEPT
                      WHERE dname = 'RESEARCH' );
```

copyright@www.adrian.runceanu.ro

6.1. SINGLE ROW SUBQUERIES

Language SQL Rows 15 Clear Command Find Tables

```
2  FROM EMP
3  WHERE deptno =
4                ( SELECT deptno
5                  FROM DEPT
6                  WHERE dname = 'RESEARCH' );
7
```

Results Explain Describe Saved SQL History

ENAME	SAL
JONES	2975
SCOTT	3000
FORD	3000
SMITH	800
ADAMS	1100

5 rows returned in 0.00 seconds Download

copyright@www.adrian.runceanu.ro

Limbajul SQL

6. SUBQUERIES (Subinterogări)

6.1. SINGLE ROW SUBQUERIES

6.2. MULTIPLE ROW SUBQUERIES

copyright@www.adrian.runceanu.ro

6.2. 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.

copyright@www.adrian.runceanu.ro

6.2. 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 ename, sal, deptno
FROM EMP
WHERE sal IN
        ( SELECT MIN(sal)
          FROM EMP
          GROUP BY deptno );
```

copyright@www.adrian.runceanu.ro

6.2. 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).

Language SQL Rows 15 Clear Command Find Tables

```
1  SELECT ename, sal, deptno
2  FROM EMP
3  WHERE sal IN
4            ( SELECT MIN(sal)
5              FROM EMP
6              GROUP BY deptno );
7
```

Results Explain Describe Saved SQL History

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

3 rows returned in 0.01 seconds Download

copyright@www.adrian.runceanu.ro

6.2. 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 ename, sal, hiredate
FROM EMP
WHERE hiredate < ANY
                 ( SELECT hiredate
                   FROM EMP
                   GROUP BY hiredate );
```

copyright@www.adrian.runceanu.ro

6.2. 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.

SQL Commands

Language SQL Rows 15

```
1  SELECT ename, sal, hiredate
2  FROM EMP
3  WHERE hiredate < ANY
4                 ( SELECT hiredate
5                   FROM EMP
6                   GROUP BY hiredate );
```

Results | Explain | Describe | Saved SQL | History

ENAME	SAL	HIREDATE
SMITH	800	12/17/1980
ALLEN	1600	02/20/1981
WARD	1250	02/22/1981
JONES	2975	04/02/1981
BLAKE	2850	05/01/1981
CLARK	2450	06/09/1981
TURNER	1500	09/08/1981
MARTIN	1250	09/28/1981
KING	5000	11/17/1981
FORD	3000	12/03/1981
JAMES	950	12/03/1981
MILLER	1300	01/23/1982
SCOTT	3000	12/09/1982

13 rows returned in 0.01 seconds Download

copyright@www.adrian.runceanu.ro

6.2. 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 ename, sal, hiredate
FROM EMP
WHERE hiredate > ALL
        ( SELECT hiredate
          FROM EMP
          GROUP BY hiredate );
```

copyright@www.adrian.runceanu.ro

6.2. MULTIPLE ROW SUBQUERIES

VALORI NULL

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

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

copyright@www.adrian.runceanu.ro

6.2. MULTIPLE ROW SUBQUERIES

Clauzele GROUP BY şi HAVING

- Pot fi folosite cu subinterogările de tip multiple row.

```
SELECT deptno, MIN(sal)
FROM EMP
GROUP BY deptno
HAVING MIN(sal) < ANY
                        ( SELECT sal
                          FROM EMP
                          WHERE deptno IN (10,20) );
```

copyright@www.adrian.runceanu.ro

6.2. MULTIPLE ROW SUBQUERIES

Clauzele GROUP BY şi HAVING

- Pot fi folosite cu subinterogările de tip multiple row.

SQL Commands

Language SQL Rows 15

```
1  SELECT deptno, MIN(sal)
2  FROM EMP
3  GROUP BY deptno
4  HAVING MIN(sal) < ANY
5                      ( SELECT sal
6                        FROM EMP
7                        WHERE deptno IN (10,20) );
```

Results | Explain | Describe | Saved SQL | History

DEPTNO	MIN(SAL)
30	950
10	1300
20	800

3 rows returned in 0.00 seconds Download

copyright@www.adrian.runceanu.ro

6.2. MULTIPLE ROW SUBQUERIES

Clauzele GROUP BY si HAVING

De asemenea, se poate folosi clauza **GROUP BY** intr-o subinterogare

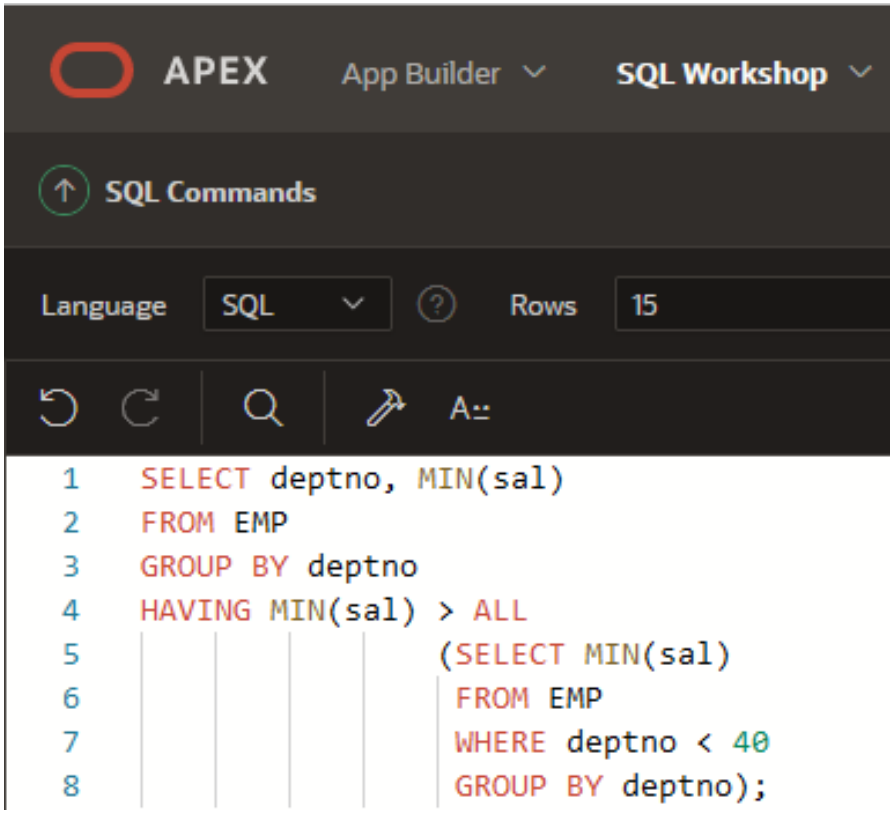
```
SELECT deptno, MIN(sal)
FROM EMP
GROUP BY deptno
HAVING MIN(sal) > ALL
                  (SELECT MIN(sal)
                   FROM EMP
                   WHERE deptno < 40
                   GROUP BY deptno);
```

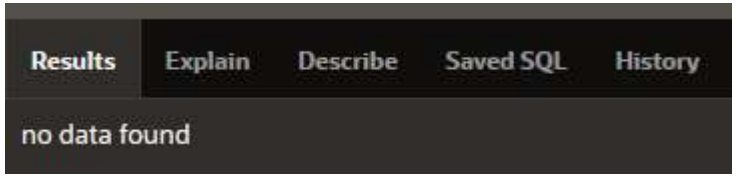
copyright@www.adrian.runceanu.ro

6.2. MULTIPLE ROW SUBQUERIES

Clauzele GROUP BY si HAVING

De asemenea, se poate folosi clauza **GROUP BY** intr-o subinterogare





copyright@www.adrian.runceanu.ro

6.2. MULTIPLE ROW SUBQUERIES

Aplicatii rezolvate:

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

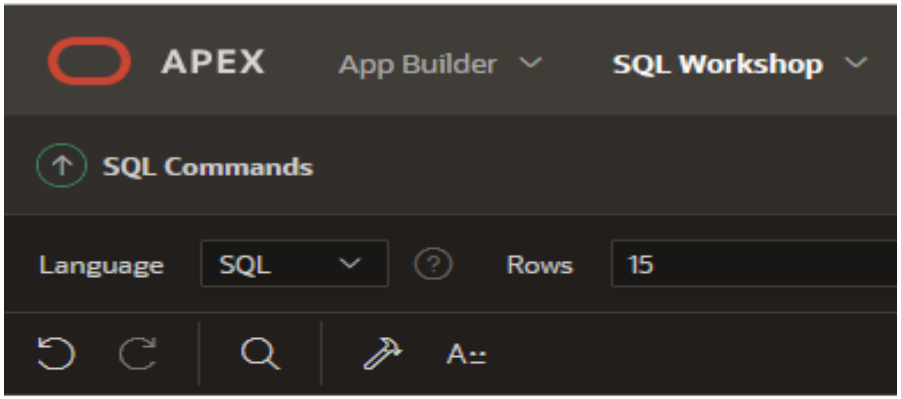
```
SELECT ename
FROM EMP
WHERE sal = ANY
                    (SELECT MIN(sal)
                     FROM EMP
                     GROUP BY deptno);
```

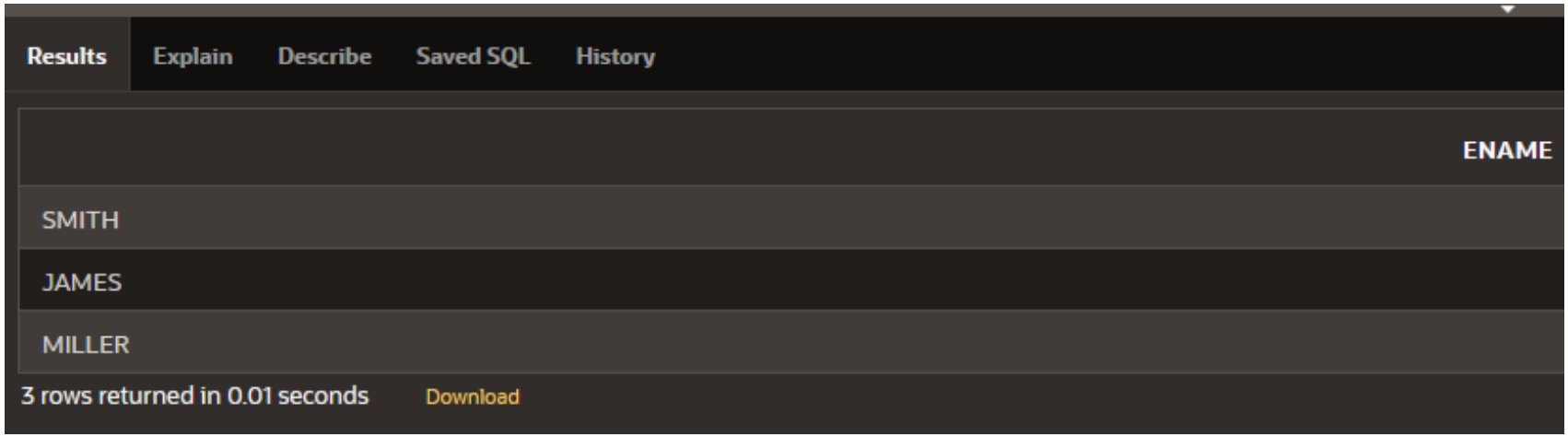
copyright@www.adrian.runceanu.ro

6.2. MULTIPLE ROW SUBQUERIES

Aplicatii rezolvate:

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



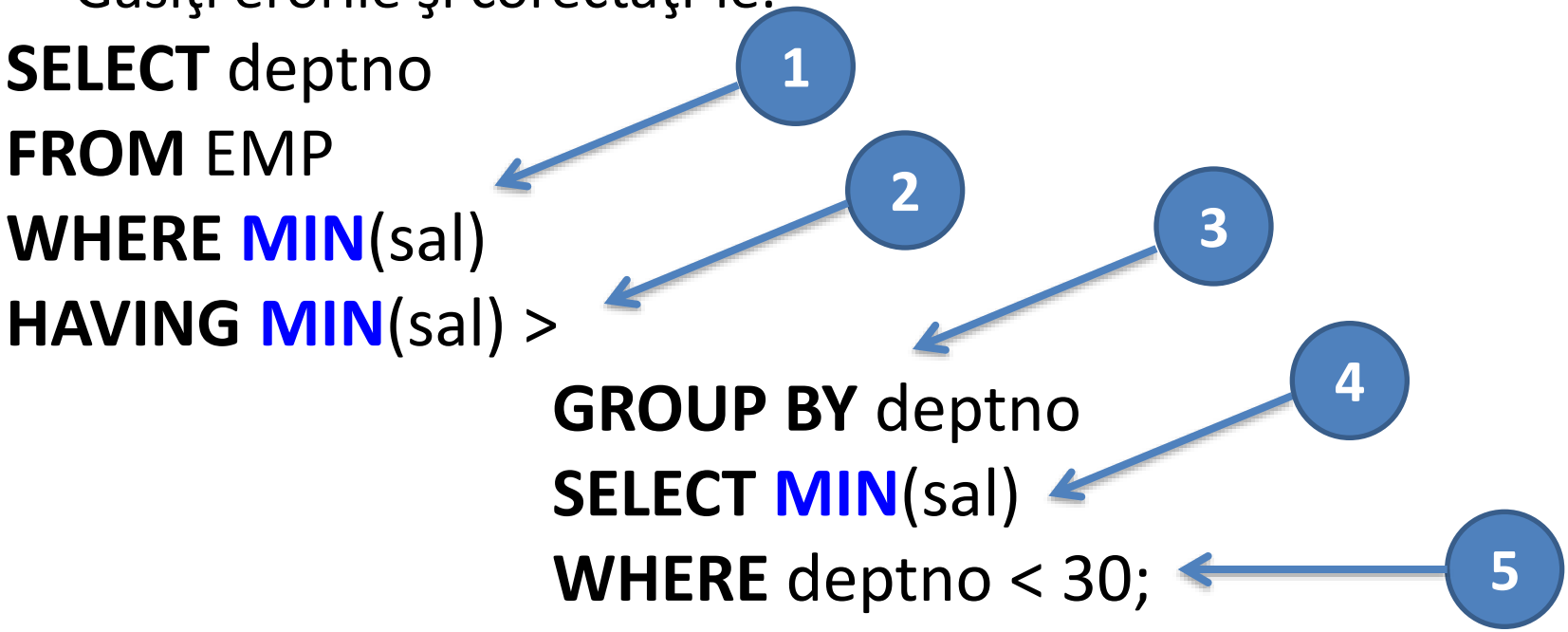


copyright@www.adrian.runceanu.ro

6.2. 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 30.

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



copyright@www.adrian.runceanu.ro

6.2. MULTIPLE ROW SUBQUERIES

Soluţia corectă este următoarea:

```
SELECT   deptno, MIN(sal)
FROM     EMP
GROUP BY deptno
HAVING   MIN(sal) <
                    ( SELECT MIN(sal)
                      FROM   EMP
                      WHERE  deptno = 30);
```

copyright@www.adrian.runceanu.ro

Subcereri multilinie

- Subcererile multilinie returneaza mai mult decat o linie.
- Cu astfel de subinterogari 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 subinterogare
3. ALL - compara o valoare cu oricare (toate) din valorile returnate de subinterogare

copyright@www.adrian.runceanu.ro

Exemplu **IN**

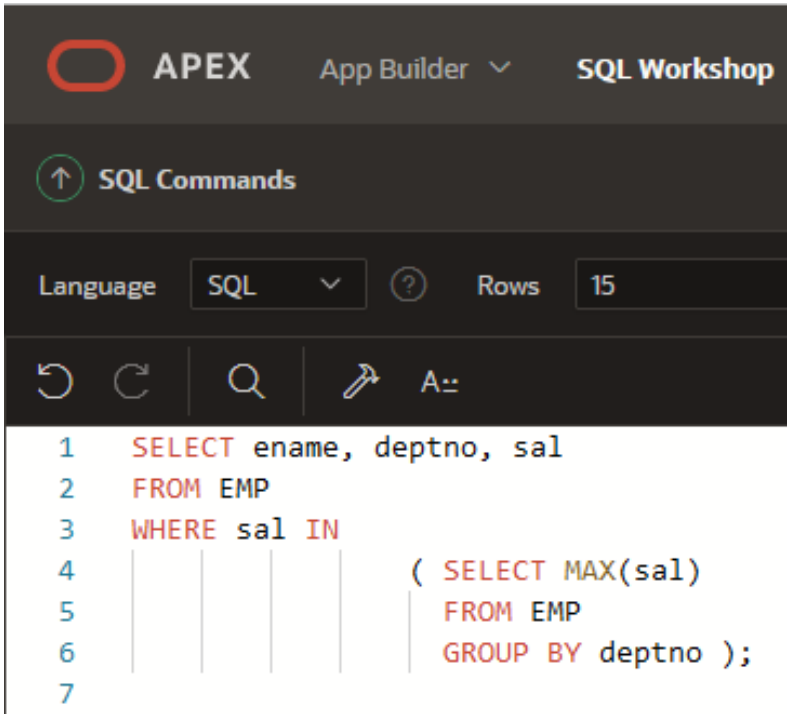
Aflati angajatii care au salariul egal cu salariul cel mai mare din fiecare departament

```
SELECT ename, deptno, sal
FROM EMP
WHERE sal IN
            ( SELECT MAX(sal)
              FROM EMP
              GROUP BY deptno )
```

subinterogarea ofera salariile maxime din fiecare departament si prin interogarea principala se afla angajatii cu aceste salarii.

copyright@www.adrian.runceanu.ro

Exemplu **IN** - Aflati angajatii care au salariul egal cu salariul cel mai mare din fiecare departament



Results | Explain | Describe | Saved SQL | History

ENAME	DEPTNO	SAL
KING	10	5000
BLAKE	30	2850
SCOTT	20	3000
FORD	20	3000

4 rows returned in 0.01 seconds Download

copyright@www.adrian.runceanu.ro

Exemplu **ANY**

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

```
SELECT ename, deptno, sal
FROM EMP
WHERE sal > ANY
                    ( SELECT sal
                    FROM EMP
                    WHERE deptno = 20 )
AND deptno <> 20;
```

copyright@www.adrian.runceanu.ro

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

SQL Commands

Language SQL Rows 15

```
1  SELECT ename, deptno, sal
2  FROM EMP
3  WHERE sal > ANY
4                  ( SELECT sal
5                    FROM EMP
6                    WHERE deptno = 20 )
7  AND deptno <> 20;
```

Results | Explain | Describe | Saved SQL | History

ENAME	DEPTNO	SAL
KING	10	5000
BLAKE	30	2850
CLARK	10	2450
ALLEN	30	1600
TURNER	30	1500
MILLER	10	1300
MARTIN	30	1250
WARD	30	1250
JAMES	30	950

9 rows returned in 0.01 seconds Download

copyright@www.adrian.runceanu.ro

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

Astfel,

< ANY inseamna mai mic decat maximul

ANY inseamna mai mare decat minimul

= ANY este echivalent cu **IN**

copyright@www.adrian.runceanu.ro

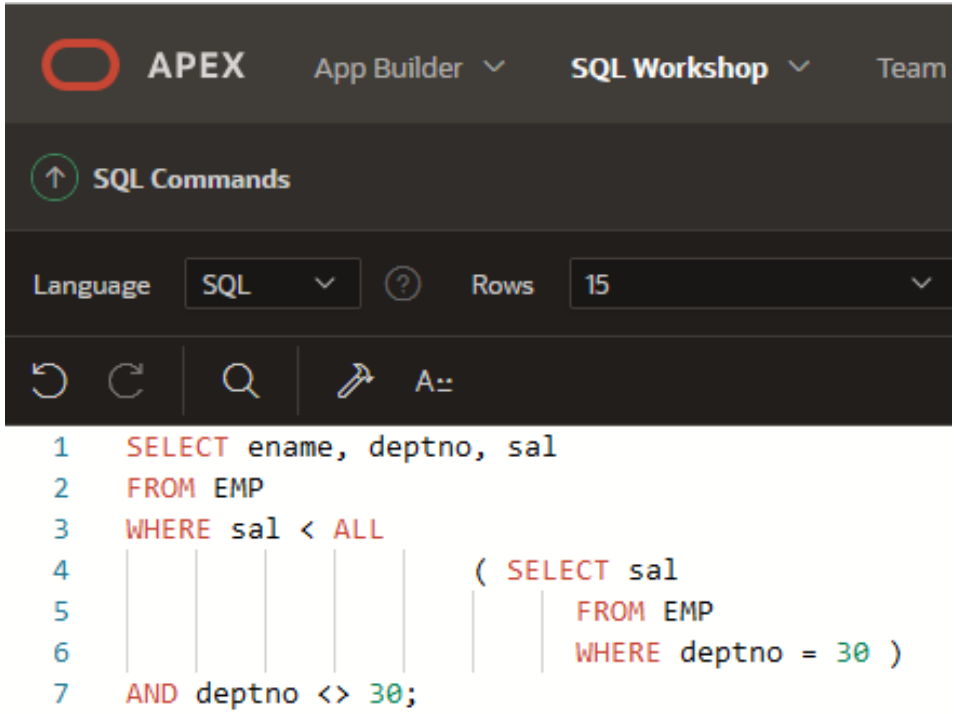
Exemplu **ALL**

Gasiti angajatii care au salariul mai mic decat oricare (toti) angajatii de la departamentul 30.

```
SELECT ename, deptno, sal
FROM EMP
WHERE sal < ALL
                    ( SELECT sal
                      FROM EMP
                      WHERE deptno = 30 )
AND deptno <> 30;
```

copyright@www.adrian.runceanu.ro

Exemplu **ALL** - Gasiti angajatii care au salariul mai mic decat oricare (toti) angajatii de la departamentul 30.



copyright@www.adrian.runceanu.ro

- Operatorul **ALL** din interogarea principala compara o valoare cu oricare valoare returnata de subinterogare.

Astfel:

> ALL inseamna mai mare decat maximul

< ALL inseamna mai mic decat minimul

copyright@www.adrian.runceanu.ro

Imbricarea subcererilor

Subcererile pot fi folosite si in interiorul altor subinterogari.

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 12/09/1982.

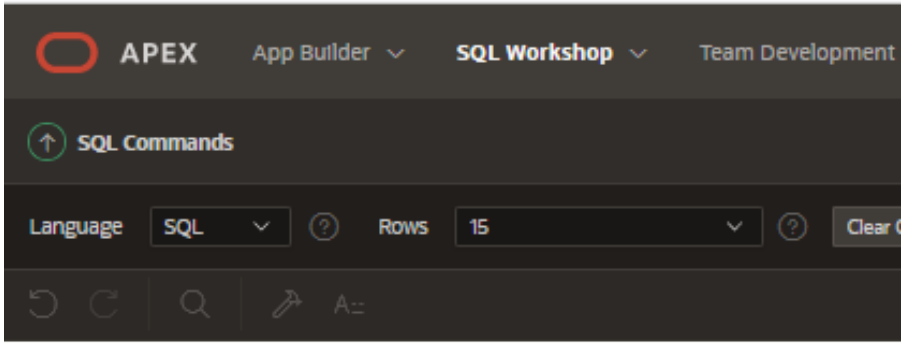
copyright@www.adrian.runceanu.ro

```
SELECT ename, job, hiredate, sal
FROM EMP
WHERE sal >
        ( SELECT MAX(sal)
        FROM EMP
        WHERE hiredate IN
                ( SELECT hiredate
                FROM EMP
                WHERE hiredate > '12/09/1982' ) );
```

Numarul maxim de imbricari pentru o subinterogare este de 255.

copyright@www.adrian.runceanu.ro

Numarul maxim de imbricari pentru o subinterogare este de 255.




```
1  SELECT ename, job, hiredate, sal
2  FROM EMP
3  WHERE sal >
4            ( SELECT MAX(sal)
5            FROM EMP
6            WHERE hiredate IN
7                        ( SELECT hiredate
8                        FROM EMP
9                        WHERE hiredate > '12/09/1982' ) );
10
```

Results | Explain | Describe | Saved SQL | History

ENAME	JOB	HIREDATE	SAL
KING	PRESIDENT	11/17/1981	5000
BLAKE	MANAGER	05/01/1981	2850
CLARK	MANAGER	06/09/1981	2450
JONES	MANAGER	04/02/1981	2975
SCOTT	ANALYST	12/09/1982	3000
FORD	ANALYST	12/03/1981	3000
ALLEN	SALESMAN	02/20/1981	1600
WARD	SALESMAN	02/22/1981	1250
MARTIN	SALESMAN	09/28/1981	1250
TURNER	SALESMAN	09/08/1981	1500
MILLER	CLERK	01/23/1982	1300

11 rows returned in 0.01 seconds Download

copyright@www.adrian.runceanu.ro

Subcereri corelate

- O subinterogare corelata este o subinterogare care se executa o data pentru fiecare linie considerata de interogarea principala si care la executie foloseste o valoare dintr-o coloana din interogarea exterioara.
- Ea se poate identifica prin folosirea unei coloane a interogarii exterioare in clauza operatorului interogarii interioare.

copyright@www.adrian.runceanu.ro

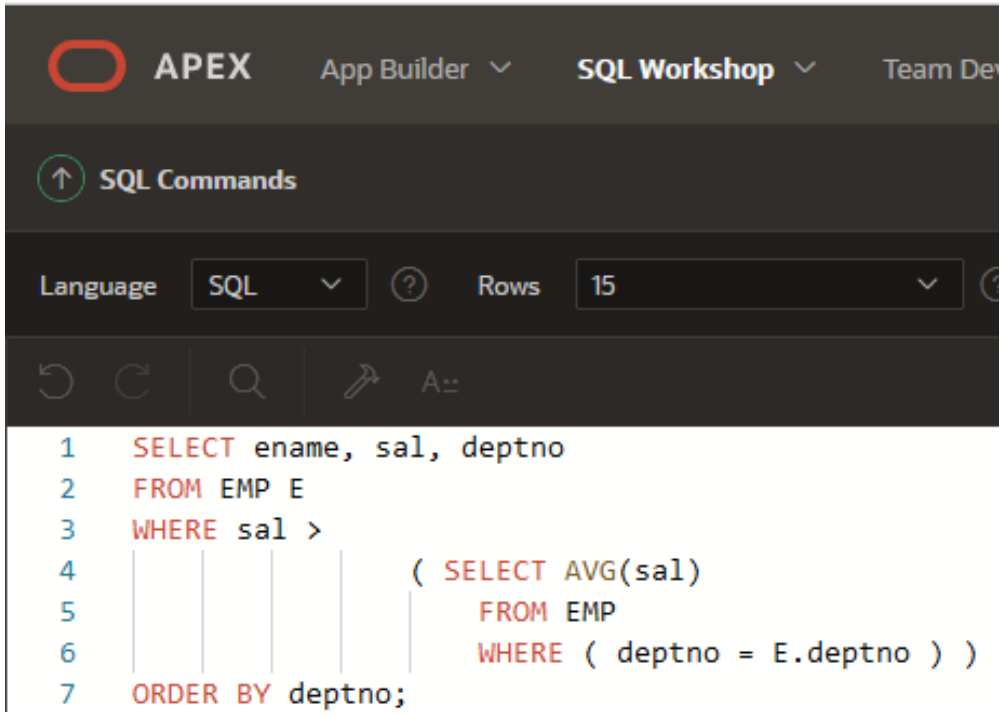
Exemplu

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

```
SELECT ename, sal, deptno
FROM EMP E
WHERE sal >
                ( SELECT AVG(sal)
                        FROM EMP
                        WHERE ( deptno = E.deptno ) )
ORDER BY deptno;
```

copyright@www.adrian.runceanu.ro

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



Results | Explain | Describe | Saved SQL | History

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

6 rows returned in 0.03 seconds Download

copyright@www.adrian.runceanu.ro

Valori de NULL intr-o subinterogare

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

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

copyright@www.adrian.runceanu.ro

Exemplu

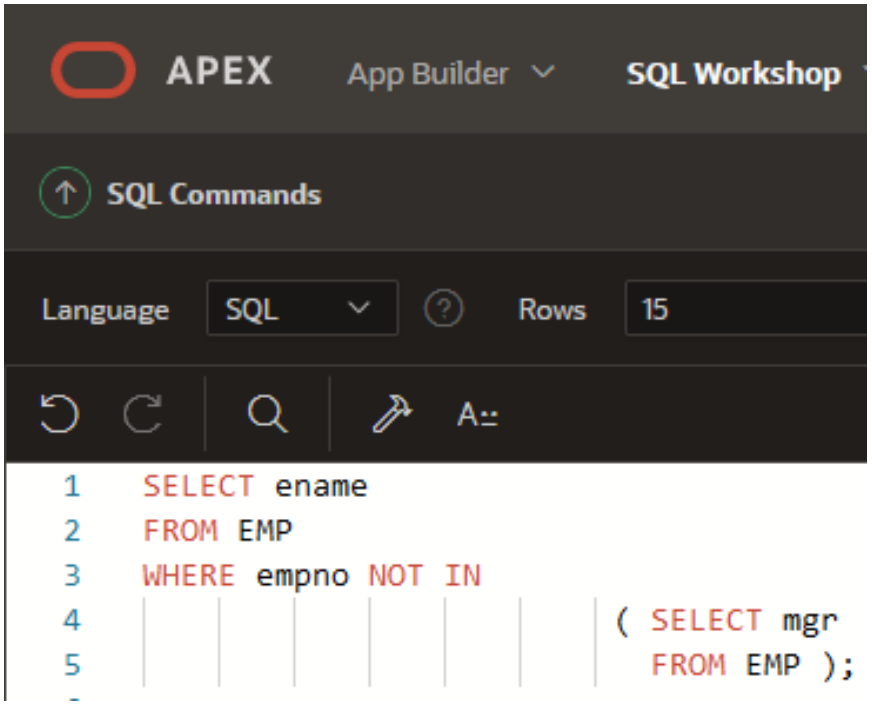
Gasiti angajatii care nu au subordonati.

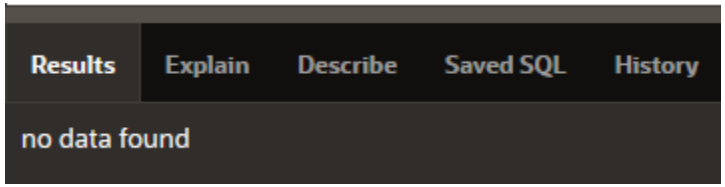
```
SELECT ename
FROM EMP
WHERE empno NOT IN
                    ( SELECT mgr
                      FROM EMP );
```

copyright@www.adrian.runceanu.ro

Exemplu

Gasiti angajatii care nu au subordonati.





copyright@www.adrian.runceanu.ro

- 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 subinterogare nu prezinta nici o problema in cazul operatorului IN in interogarea principala (in echivalent cu = ALL).

copyright@www.adrian.runceanu.ro

Exemplu

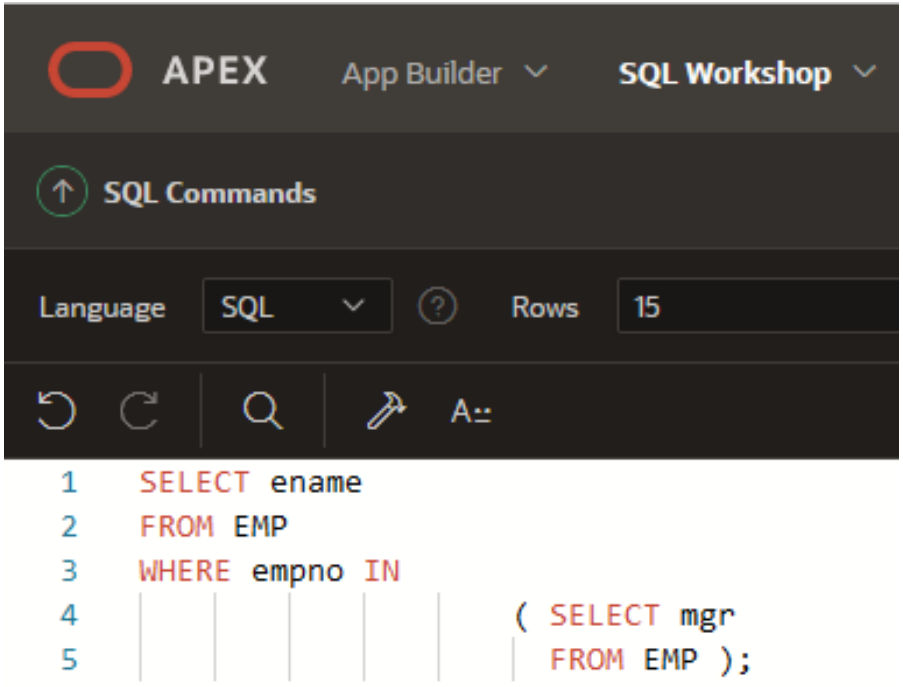
Gasiti angajatii care au subordonati.

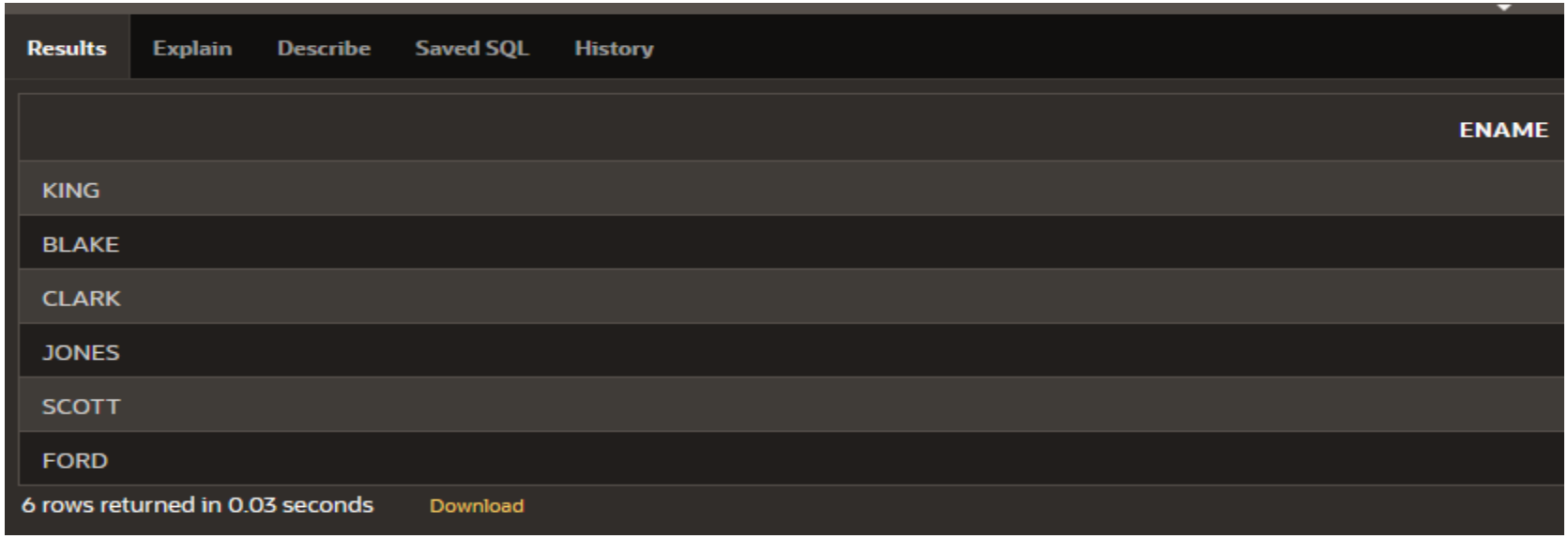
```
SELECT ename
FROM EMP
WHERE empno IN
                    ( SELECT mgr
                      FROM EMP );
```

copyright@www.adrian.runceanu.ro

Exemplu

Gasiti angajatii care au subordonati.





copyright@www.adrian.runceanu.ro

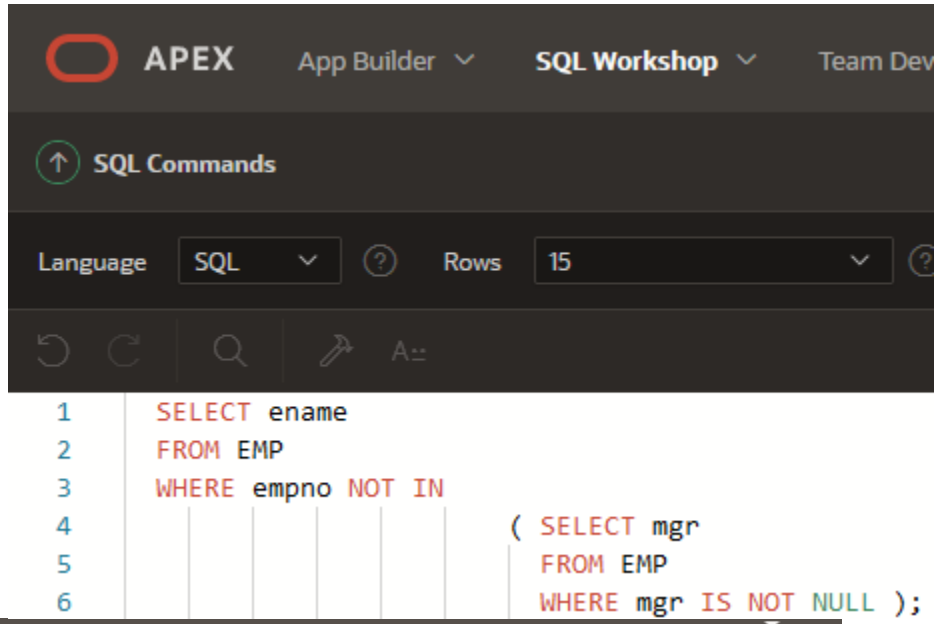
- In cazul utilizarii operatorului NOT IN in interogarea principala trebuie avut grija sa se excluda valorile NULL din raspunsurile subcererii.

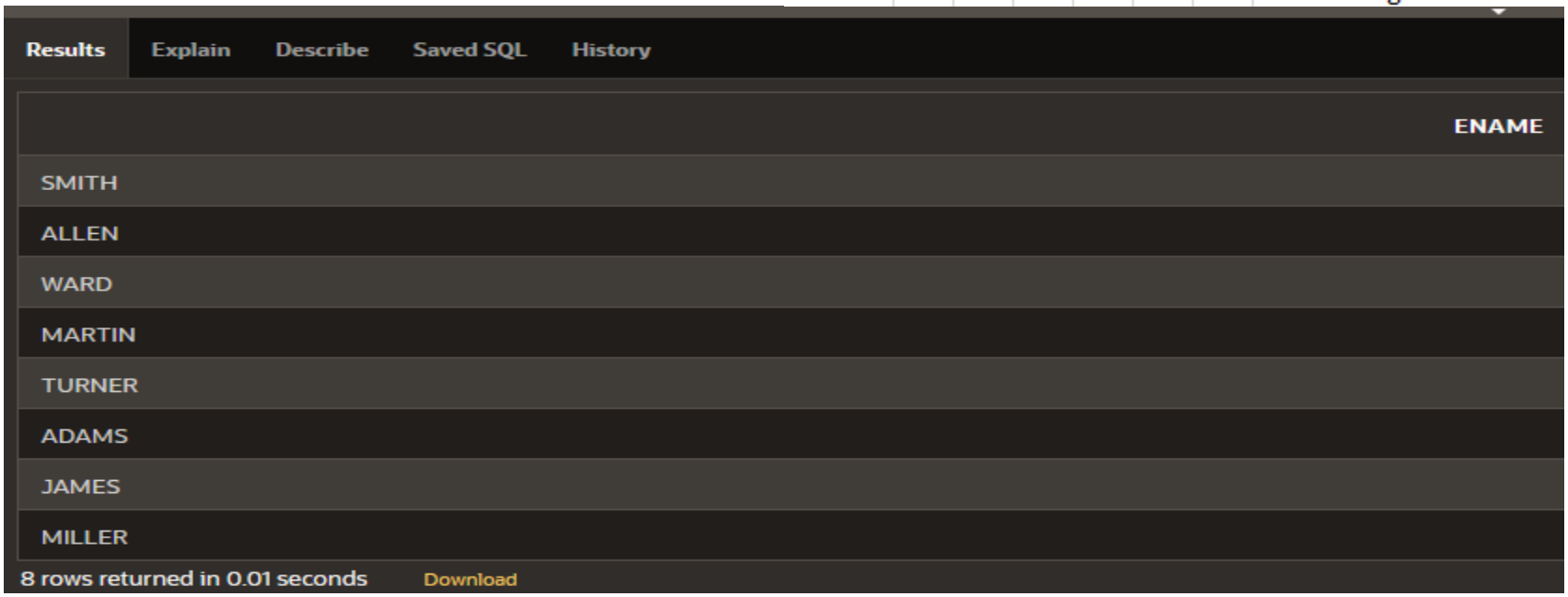
Exemplu

Gasiti angajatii care nu au subordonati.

copyright@www.adrian.runceanu.ro

```
SELECT ename
FROM EMP
WHERE empno NOT IN
        ( SELECT mgr
          FROM EMP
          WHERE mgr IS NOT NULL );
```





copyright@www.adrian.runceanu.ro

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

copyright@www.adrian.runceanu.ro

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.

copyright@www.adrian.runceanu.ro

Limbajul SQL

6.2. Cereri din mai multe tabele (JOIN-uri)

6.2.1. JOIN-urile proprietatea ORACLE

6.2.1.1. Cartesian Product

6.2.1.2. Equijoin

6.2.1.3. Non-equijoin

6.2.1.4. Outer join

6.2.1.5. Self join

6.2.2. JOIN-urile ANSI/ISO SQL99

6.2.2.1. Cross join

6.2.2.2. Natural join

6.2.2.3. Using clause

6.2.2.4. Full (two sided) outer joins

6.2.2.5. Arbitrary join conditions for outer joins

6.2.3. Operatorii pe mulţimi

copyright@www.adrian.runceanu.ro

6.2. Cereri din mai multe tabele (JOIN-uri)

Scopul acestui curs este de a studia modalitatea preluării de date din mai multe tabele.

Pentru a putea realiza acest lucru, făcând legături între tabelele unei baze de date, putem folosi:

1. **JOIN**-urile proprietate **ORACLE**

2. **JOIN**-urile **ANSI/ISO SQL99**

6.2. Cereri din mai multe tabele (JOIN-uri)

Tipuri de **JOIN**-uri:

Join-uri proprietate ORACLE	Join-uri SQL 1999
Cartesian Product	Cross join
Equijoin	Natural join
Non-equijoin	Using clause
Outer join	Full (two sided) outer joins
Self join	Arbitrary join conditions for outer joins

copyright@www.adrian.runceanu.ro

Limbajul SQL

6.2. Cereri din mai multe tabele (JOIN-uri)

6.2.1. JOIN-urile proprietatea ORACLE

6.2.1.1. Cartesian Product

6.2.1.2. Equijoin

6.2.1.3. Non-equijoin

6.2.1.4. Outer join

6.2.1.5. Self join

6.2.2. JOIN-urile ANSI/ISO SQL99

6.2.2.1. Cross join

6.2.2.2. Natural join

6.2.2.3. Using clause

6.2.2.4. Full (two sided) outer joins

6.2.2.5. Arbitrary join conditions for outer joins

6.2.3. Operatorii pe mulţimi

copyright@www.adrian.runceanu.ro

Cartesian Product and the Join Operations

JOIN-uri proprietatea **Oracle**

Pentru a prelua date din mai multe tabele, forma de bază a unei instrucţiuni **SELECT** constă în adăugarea unei **condiţii de legătură (join)** în clauza **WHERE**

Numele coloanei trebuie prefixat de numele tabelei în situaţiile când acelaşi nume de coloană apare în mai multe tabele.

copyright@www.adrian.runceanu.ro

Cartesian Product and the Join Operations

```
SELECT table1.column, table2.column
FROM table1, table2
WHERE table1.column1 = table2.column2;
```

copyright@www.adrian.runceanu.ro

Cartesian Product and the Join Operations

PRODUSUL CARTEZIAN (CARTESIAN PRODUCT)

Presupune că *toate liniile din prima tabelă să fie unite (legate) cu toate liniile din tabela a doua*.

Se produce atunci când:

1) condiţia de join este omisă
2) condiţia de join nu este validă

Pentru a evita produsul cartezian trebuie adăugată o condiţie de join validă.

Produsul cartezian generează foarte multe linii şi este folosit foarte rar.

copyright@www.adrian.runceanu.ro

Limbajul SQL

6.2. Cereri din mai multe tabele (JOIN-uri)

6.2.1. JOIN-urile proprietatea ORACLE

6.2.1.1. Cartesian Product

6.2.1.2. Equijoin

6.2.1.3. Non-equijoin

6.2.1.4. Outer join

6.2.1.5. Self join

6.2.2. JOIN-urile ANSI/ISO SQL99

6.2.2.1. Cross join

6.2.2.2. Natural join

6.2.2.3. Using clause

6.2.2.4. Full (two sided) outer joins

6.2.2.5. Arbitrary join conditions for outer joins

6.2.3. Operatorii pe mulţimi

copyright@www.adrian.runceanu.ro

Equijoin

EQUIJOIN

Uneori denumit **simple join** sau **inner join**, *un equijoin este o legătură între tabele care combină linii ce au valori echivalente pentru coloanele specificate.*

Exemplu:

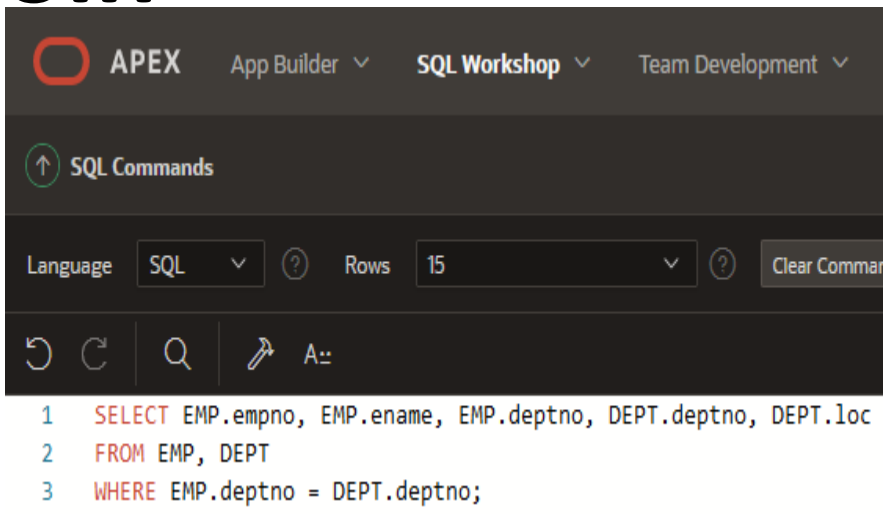
```
SELECT EMP.empno, EMP.ename,
       EMP.deptno, DEPT.deptno,
       DEPT.loc
FROM EMP, DEPT
WHERE EMP.deptno = DEPT.deptno;
```

copyright@www.adrian.runceanu.ro

Equijoin

EQUIJOIN

Uneori denumit **simple join** sau **inner join**, *un equijoin este o legătură între tabele care combină linii ce au valori echivalente pentru coloanele specificate.*



Results | Explain | Describe | Saved SQL | History

EMPNO	ENAME	DEPTNO	DEPTNO	LOC
7782	CLARK	10	10	NEW YORK
7934	MILLER	10	10	NEW YORK
7839	KING	10	10	NEW YORK
7902	FORD	20	20	DALLAS
7788	SCOTT	20	20	DALLAS
7566	JONES	20	20	DALLAS
7369	SMITH	20	20	DALLAS
7876	ADAMS	20	20	DALLAS
7521	WARD	30	30	CHICAGO
7654	MARTIN	30	30	CHICAGO
7844	TURNER	30	30	CHICAGO
7900	JAMES	30	30	CHICAGO
7499	ALLEN	30	30	CHICAGO
7698	BLAKE	30	30	CHICAGO

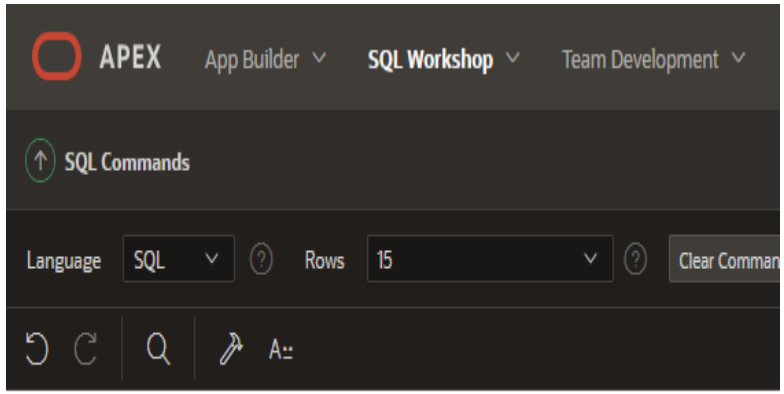
14 rows returned in 0.02 seconds Download

copyright@www.adrian.runceanu.ro

Equijoin

Exemplu:

SELECT EMP.empno, EMP.ename,
EMP.deptno, DEPT.deptno,
DEPT.loc
FROM EMP, DEPT
WHERE EMP.deptno = DEPT.deptno;



EMPNO	ENAME	DEPTNO	DEPTNO	LOC
7782	CLARK	10	10	NEW YORK
7934	MILLER	10	10	NEW YORK
7839	KING	10	10	NEW YORK
7902	FORD	20	20	DALLAS
7788	SCOTT	20	20	DALLAS
7566	JONES	20	20	DALLAS
7369	SMITH	20	20	DALLAS
7876	ADAMS	20	20	DALLAS
7521	WARD	30	30	CHICAGO
7654	MARTIN	30	30	CHICAGO
7844	TURNER	30	30	CHICAGO
7900	JAMES	30	30	CHICAGO
7499	ALLEN	30	30	CHICAGO
7698	BLAKE	30	30	CHICAGO

copyright@www.adrian.runceanu.ro

Equijoin

DEPT

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST | Sample Queries

Query | Count Rows | Insert Row | Load Data

EDIT	DEPTNO	DNAME	LOC
	10	ACCOUNTING	NEW YORK
	20	RESEARCH	DALLAS
	30	SALES	CHICAGO
	40	OPERATIONS	BOSTON

Cheie străină (Foreign key)

Cheie primară (Primary key)

EMP

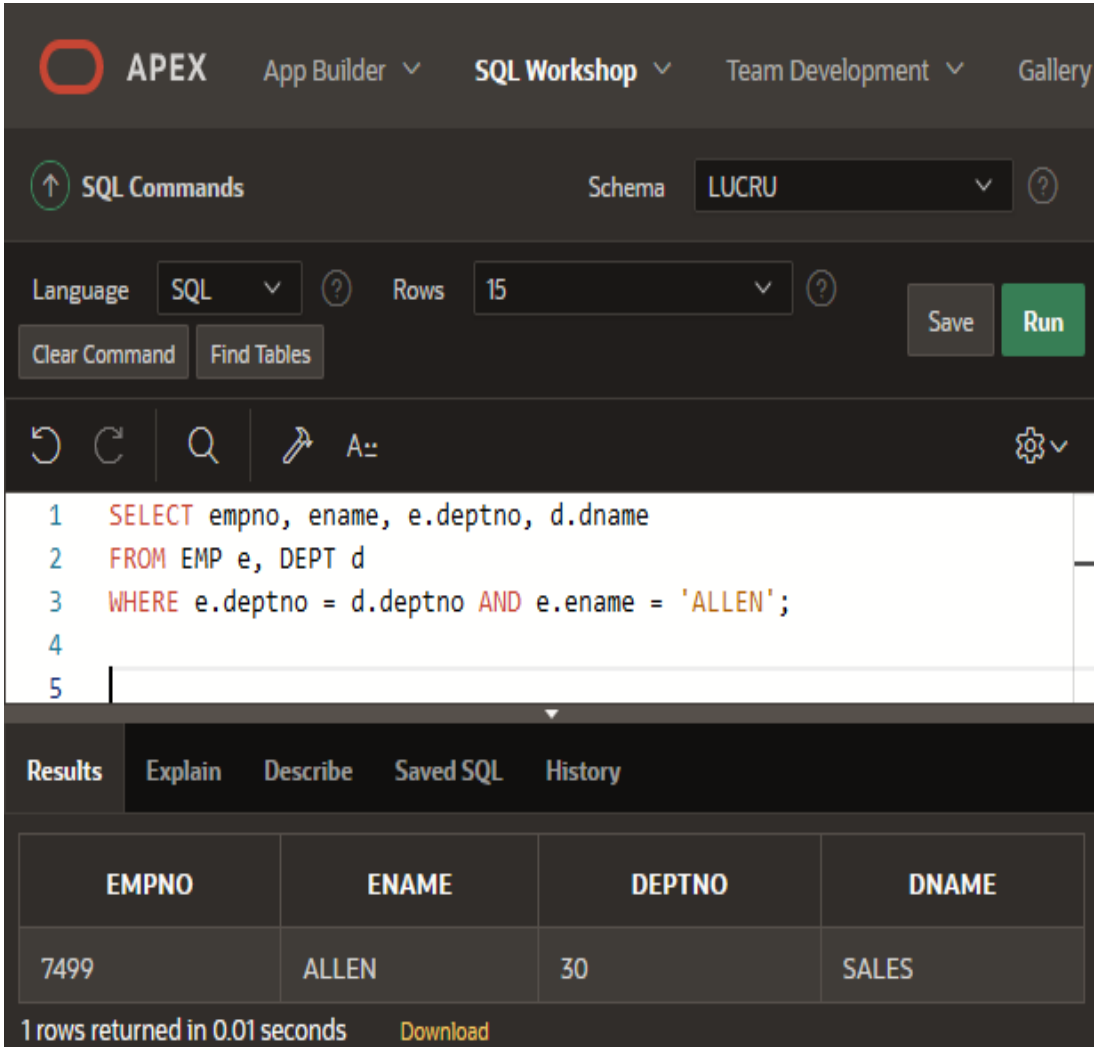
EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	11/17/1981	5000	-	10
7698	BLAKE	MANAGER	7839	05/01/1981	2850	-	30
7782	CLARK	MANAGER	7839	06/09/1981	2450	-	10
7566	JONES	MANAGER	7839	04/02/1981	2975	-	20
7788	SCOTT	ANALYST	7566	12/09/1982	3000	-	20
7902	FORD	ANALYST	7566	12/03/1981	3000	-	20
7369	SMITH	CLERK	7902	12/17/1980	800	-	20
7499	ALLEN	SALESMAN	7698	02/20/1981	1600	300	30
7521	WARD	SALESMAN	7698	02/22/1981	1250	500	30
7654	MARTIN	SALESMAN	7698	09/28/1981	1250	1400	30
7844	TURNER	SALESMAN	7698	09/08/1981	1500	0	30
7876	ADAMS	CLERK	7788	01/12/1983	1100	-	20
7900	JAMES	CLERK	7698	12/03/1981	950	-	30
7934	MILLER	CLERK	7782	01/23/1982	1300	-	10

copyright@www.adrian.runceanu.ro

Equijoin

FOLOSIREA OPERATORULUI AND

Ca şi la interogările care folosesc o singură tabelă, se poate folosi operatorul **AND** pentru a restricţiona liniile selectate.



copyright@www.adrian.runceanu.ro

Equijoin

ALIAS-URI

Atunci când denumirile coloanelor şi tabelelor sunt mari, devine incomod de lucrat cu acestea.

Pentru a scurta denumirile respective, se folosesc alias-urile.

Se pot folosi alias-uri atât pentru coloane cât şi pentru tabele.

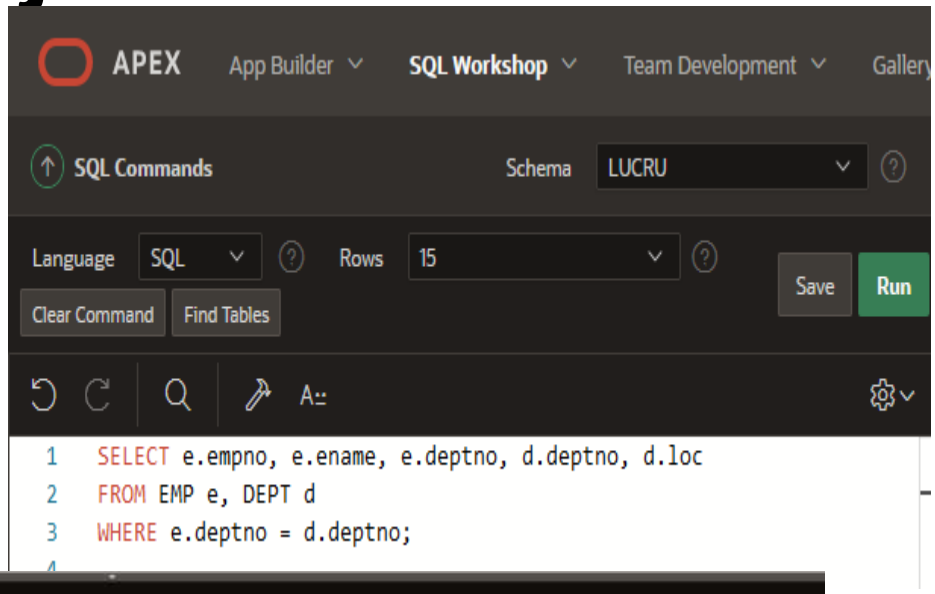
Dacă este precizat un alias pentru o tabelă în clauza **FROM**, atunci alias-ul respectiv trebuie să înlocuiască numele tabelei în clauza **SELECT.**

copyright@www.adrian.runceanu.ro

Equijoin

Exemplu:

```
SELECT e.empno, e.ename,
       a.deptno, d.deptno, d.loc
FROM EMP e, DEPT d
WHERE e.deptno = d.deptno;
```



Results | Explain | Describe | Saved SQL | History

EMPNO	ENAME	DEPTNO	DEPTNO	LOC
7782	CLARK	10	10	NEW YORK
7934	MILLER	10	10	NEW YORK
7839	KING	10	10	NEW YORK
7902	FORD	20	20	DALLAS
7788	SCOTT	20	20	DALLAS
7566	JONES	20	20	DALLAS
7369	SMITH	20	20	DALLAS
7876	ADAMS	20	20	DALLAS
7521	WARD	30	30	CHICAGO
7654	MARTIN	30	30	CHICAGO
7844	TURNER	30	30	CHICAGO
7900	JAMES	30	30	CHICAGO
7499	ALLEN	30	30	CHICAGO
7698	BLAKE	30	30	CHICAGO

14 rows returned in 0.01 seconds | Download

copyright@www.adrian.runceanu.ro

Limbajul SQL

6.2. Cereri din mai multe tabele (JOIN-uri)

6.2.1. JOIN-urile proprietatea ORACLE

6.2.1.1. Cartesian Product

6.2.1.2. Equijoin

6.2.1.3. Non-equijoin

6.2.1.4. Outer join

6.2.1.5. Self join

6.2.2. JOIN-urile ANSI/ISO SQL99

6.2.2.1. Cross join

6.2.2.2. Natural join

6.2.2.3. Using clause

6.2.2.4. Full (two sided) outer joins

6.2.2.5. Arbitrary join conditions for outer joins

6.2.3. Operatorii pe mulţimi

copyright@www.adrian.runceanu.ro

NONEQUIJOINS

Este posibil să dorim *să extragem date dintr-o tabelă ce nu are coloană corespondentă în cealaltă tabelă* (exemplu – când datele se înregistrează ca domenii de valori).

În această situaţie se foloseşte **nonequijoin**-ul.

În acest tip de join, deoarece nu există o potrivire exactă între 2 coloane din fiecare tabelă, nu se foloseşte operatorul de egalitate.

Se pot folosi operatorii:

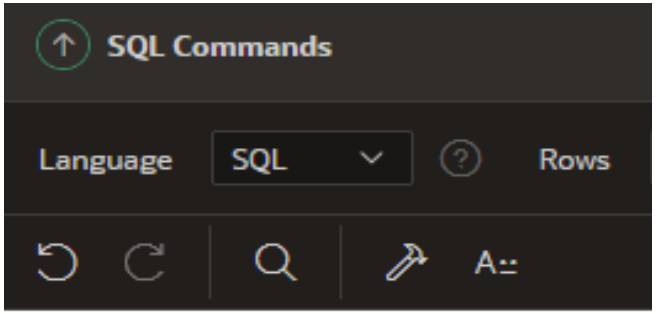
<=, >=, BETWEEN…AND

copyright@www.adrian.runceanu.ro

NONEQUIJOINS

Exemplu:

SELECT *
FROM EMP **E,** DEPT **D**
WHERE E.deptno **>** D.deptno
AND E.deptno **IN** (10, 30)**;**



Results | Explain | Describe | Saved SQL | History

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO	DNAME	LOC
7654	MARTIN	SALESMAN	7698	09/28/1981	1250	1400	30	10	ACCOUNTING	NEW YORK
7844	TURNER	SALESMAN	7698	09/08/1981	1500	0	30	10	ACCOUNTING	NEW YORK
7521	WARD	SALESMAN	7698	02/22/1981	1250	500	30	10	ACCOUNTING	NEW YORK
7900	JAMES	CLERK	7698	12/03/1981	950	-	30	10	ACCOUNTING	NEW YORK
7698	BLAKE	MANAGER	7839	05/01/1981	2850	-	30	10	ACCOUNTING	NEW YORK
7499	ALLEN	SALESMAN	7698	02/20/1981	1600	300	30	10	ACCOUNTING	NEW YORK
7654	MARTIN	SALESMAN	7698	09/28/1981	1250	1400	30	20	RESEARCH	DALLAS
7844	TURNER	SALESMAN	7698	09/08/1981	1500	0	30	20	RESEARCH	DALLAS
7521	WARD	SALESMAN	7698	02/22/1981	1250	500	30	20	RESEARCH	DALLAS
7900	JAMES	CLERK	7698	12/03/1981	950	-	30	20	RESEARCH	DALLAS
7698	BLAKE	MANAGER	7839	05/01/1981	2850	-	30	20	RESEARCH	DALLAS
7499	ALLEN	SALESMAN	7698	02/20/1981	1600	300	30	20	RESEARCH	DALLAS

12 rows returned in 0.01 seconds Download

copyright@www.adrian.runceanu.ro

Limbajul SQL

6.2. Cereri din mai multe tabele (JOIN-uri)

6.2.1. JOIN-urile proprietatea ORACLE

6.2.1.1. Cartesian Product

6.2.1.2. Equijoin

6.2.1.3. Non-equijoin

6.2.1.4. Outer join

6.2.1.5. Self join

6.2.2. JOIN-urile ANSI/ISO SQL99

6.2.2.1. Cross join

6.2.2.2. Natural join

6.2.2.3. Using clause

6.2.2.4. Full (two sided) outer joins

6.2.2.5. Arbitrary join conditions for outer joins

6.2.3. Operatorii pe mulţimi

copyright@www.adrian.runceanu.ro

Outer Joins

Join-urile studiate până în acest moment au avut ca rezultat linii care:

1. *fie au avut o valoare care să corespundă în ambele tabele*
2. *fie o valoare într-o tabelă se regăsea în intervalul dintre 2 valori ale celeilalte tabele*

Liniile care nu îndeplineau condiţiile nu erau selectate.

copyright@www.adrian.runceanu.ro

Outer Joins

Uneori, *dorim să extragem toate datele dintr-o tabelă, chiar dacă nu au valori care să se potrivească în cealaltă tabelă (“missing data”).*

Acest lucru se realizează folosind **outer join-ul**.

Operatorul pentru outer join este semnul plus pus între paranteze rotunde – **(+)**

copyright@www.adrian.runceanu.ro

Outer Joins

Un outer join este folosit *pentru a vizualiza toate liniile care au valoare corespondentă în cealaltă tabelă şi liniile dintr-o tabelă care nu au valoare corespondentă în cealaltă tabelă.*

*Pentru a indica tabela deficitară (care poate avea date lipsă – "missing data"), se pune operatorul **(+)** după numele coloanei din tabelă, în clauza **WHERE**.*

copyright@www.adrian.runceanu.ro

Outer Joins

OBSERVAŢIE:

Un **outer join** nu poate folosi operatorul **IN** şi nu poate fi legat la altă condiţie prin operatorul **OR**.

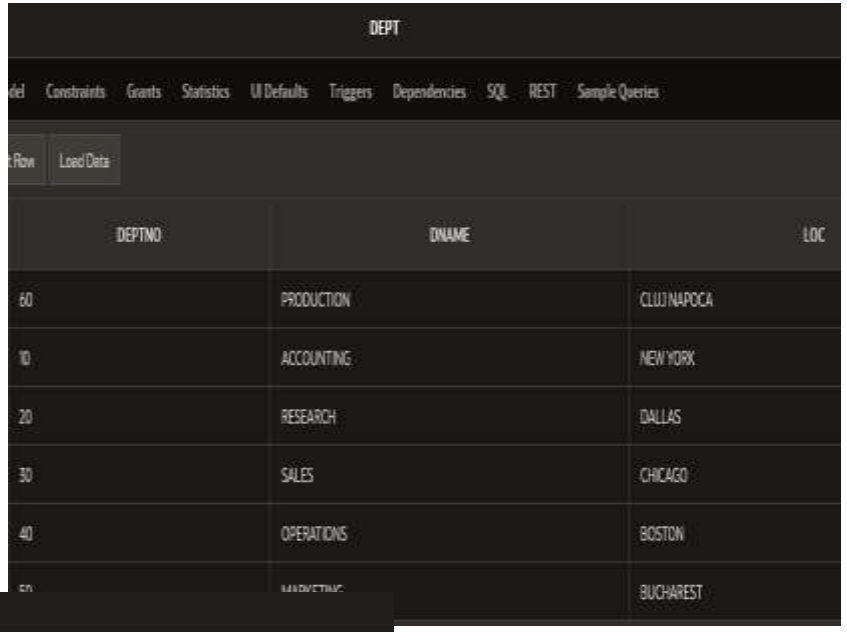
```
SELECT table1.column, table2.column
FROM table1, table2
WHERE table1.column(+) = table2.column
```

```
SELECT table1.column, table2.column
FROM table1, table2
WHERE table1.column = table2.column(+)
```

copyright@www.adrian.runceanu.ro

Outer Joins

Nu există angajaţi in departamentele 40, 50 şi 60.



DEPT

del Constraints Grants Statistics UI Defaults Triggers Dependencies SQL REST Sample Queries

t Row Load Data

DEPTNO	DNAME	LOC
60	PRODUCTION	CLUJ NAPOCA
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON
50	MARKETING	BUCHAREST

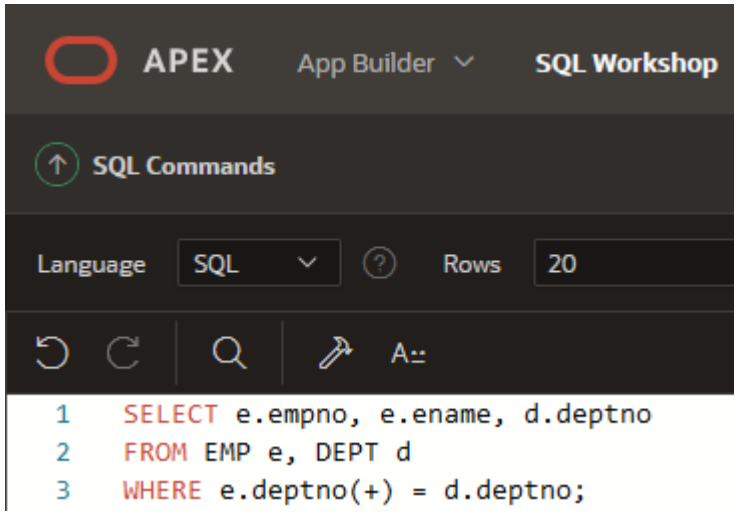
EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	11/17/1981	5000	-	10
7698	BLAKE	MANAGER	7839	05/01/1981	2850	-	30
7782	CLARK	MANAGER	7839	06/09/1981	2450	-	10
7566	JONES	MANAGER	7839	04/02/1981	2975	-	20
7788	SCOTT	ANALYST	7566	12/09/1982	3000	-	20
7902	FORD	ANALYST	7566	12/03/1981	3000	-	20
7369	SMITH	CLERK	7902	12/17/1980	800	-	20
7499	ALLEN	SALESMAN	7698	02/20/1981	1600	300	30
7521	WARD	SALESMAN	7698	02/22/1981	1250	500	30
7654	MARTIN	SALESMAN	7698	09/28/1981	1250	1400	30
7844	TURNER	SALESMAN	7698	09/08/1981	1500	0	30
7876	ADAMS	CLERK	7788	01/12/1983	1100	-	20
7900	JAMES	CLERK	7698	12/03/1981	950	-	30
7934	MILLER	CLERK	7782	01/23/1982	1300	-	10

copyright@www.adrian.runceanu.ro

Outer Joins

```
SELECT e.empno, a.ename,
    d.deptno
FROM EMP e, DEPT d
WHERE a.deptno(+) = d.deptno;
```



Results | Explain | Describe | Saved SQL | History

EMPNO	ENAME	DEPTNO
7839	KING	10
7698	BLAKE	30
7782	CLARK	10
7566	JONES	20
7788	SCOTT	20
7902	FORD	20
7369	SMITH	20
7499	ALLEN	30
7521	WARD	30
7654	MARTIN	30
7844	TURNER	30
7876	ADAMS	20
7900	JAMES	30
7934	MILLER	10
-	-	50
-	-	40
-	-	60

17 rows returned in 0.01 seconds Download

copyright@www.adrian.runceanu.ro

Outer Joins

APLICAŢII:

1) Creaţi o interogare care are ca rezultat afişarea numelui (nume) şi id-ul şi numele departamentului pentru angajaţi. Includeţi toţi angajaţii, chiar dacă nu sunt asignaţi unui departament.

2) Modificaţi interogarea din problema anterioară pentru a afişa toate id-urile departamentelor, chiar dacă nu au angajaţi asociaţi lor.

copyright@www.adrian.runceanu.ro

Outer Joins

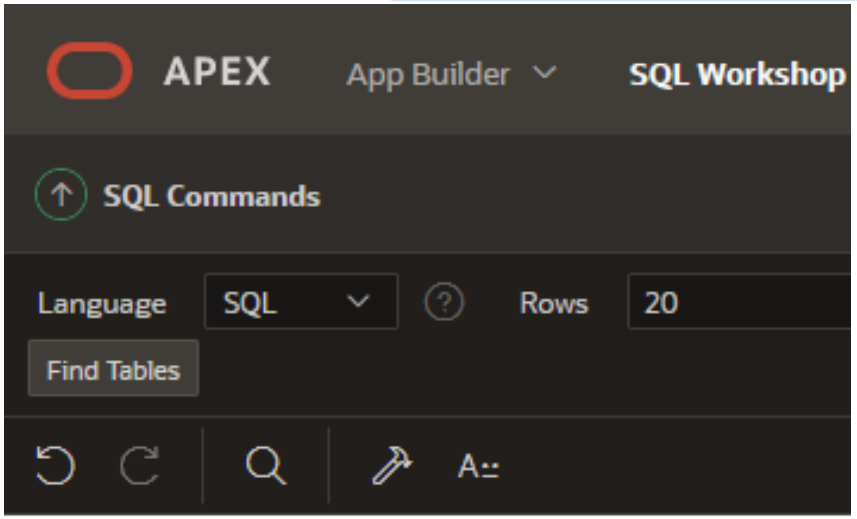
1) Creaţi o interogare care are ca rezultat afişarea numelui (nume) şi id-ul şi numele departamentului pentru angajaţi. Includeţi toţi angajaţii, chiar dacă nu sunt asignaţi unui departament.

```
SELECT e.name, e.deptno, d.dname
FROM EMP e, DEPT d
WHERE e.deptno = d.deptno(+);
```

copyright@www.adrian.runceanu.ro

Outer Joins

1) Creaţi o interogare care are ca rezultat afişarea numelui (nume) şi id-ul şi numele departamentului pentru angajaţi. Includeţi toţi angajaţii, chiar dacă nu sunt asignaţi unui departament.




```
1   SELECT e.ename, e.deptno, d.dname
2   FROM EMP e, DEPT d
3   WHERE e.deptno = d.deptno(+);
```

Results | Explain | Describe | Saved SQL | History

ENAME	DEPTNO	DNAME
KING	10	ACCOUNTING
CLARK	10	ACCOUNTING
MILLER	10	ACCOUNTING
JONES	20	RESEARCH
SCOTT	20	RESEARCH
FORD	20	RESEARCH
SMITH	20	RESEARCH
ADAMS	20	RESEARCH
BLAKE	30	SALES
ALLEN	30	SALES
WARD	30	SALES
MARTIN	30	SALES
TURNER	30	SALES
JAMES	30	SALES

copyright@www.adrian.runceanu.ro

Outer Joins

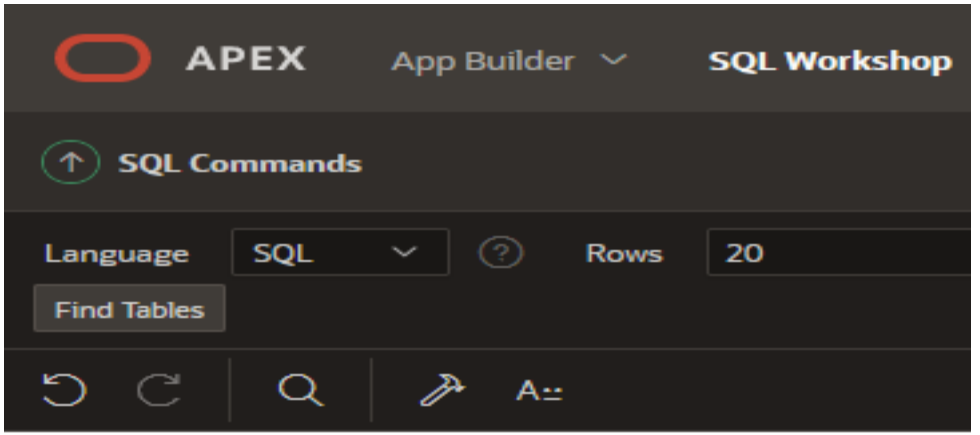
2) Modificaţi interogarea din problema anterioară pentru a afişa toate id-urile departamentelor, chiar dacă nu au angajaţi asociaţi lor.

```
SELECT e.ename, e.deptno, d.dname
FROM EMP e, DEPT d
WHERE e.deptno(+) = d.deptno;
```

copyright@www.adrian.runceanu.ro

Outer Joins

2)Modificaţi interogarea din problema anterioară pentru a afişa toate id-urile departamentelor, chiar dacă nu au angajaţi asociaţi lor.




```
1   SELECT e.ename, e.deptno, d.dname
2   FROM EMP e, DEPT d
3   WHERE e.deptno(+) = d.deptno;
```

Results | Explain | Describe | Saved SQL | History

ENAME	DEPTNO	DNAME
CLARK	10	ACCOUNTING
MILLER	10	ACCOUNTING
KING	10	ACCOUNTING
FORD	20	RESEARCH
SCOTT	20	RESEARCH
JONES	20	RESEARCH
SMITH	20	RESEARCH
ADAMS	20	RESEARCH
WARD	30	SALES
MARTIN	30	SALES
TURNER	30	SALES
JAMES	30	SALES
ALLEN	30	SALES
BLAKE	30	SALES
-	-	OPERATIONS
-	-	MARKETING
-	-	PRODUCTION

17 rows returned in 0.01 seconds Download

copyright@www.adrian.runceanu.ro

Limbajul SQL

6.2. Cereri din mai multe tabele (JOIN-uri)

6.2.1. JOIN-urile proprietatea ORACLE

6.2.1.1. Cartesian Product

6.2.1.2. Equijoin

6.2.1.3. Non-equijoin

6.2.1.4. Outer join

6.2.1.5. Self join

6.2.2. JOIN-urile ANSI/ISO SQL99

6.2.2.1. Cross join

6.2.2.2. Natural join

6.2.2.3. Using clause

6.2.2.4. Full (two sided) outer joins

6.2.2.5. Arbitrary join conditions for outer joins

6.2.3. Operatorii pe mulţimi

copyright@www.adrian.runceanu.ro

Self Joins

- *În modelarea de date, uneori este necesar să punem în evidenţă o entitate în relaţie cu ea însăşi.*
- Un exemplu este entitatea **angajat**.
- Un angajat poate fi şi manager.
- Odată ce avem tabela EMP, devine necesară o relaţie specială numită **self join** (un join de la tabela EMP la ea însăşi), pentru a afla numele managerului pentru fiecare angajat.

copyright@www.adrian.runceanu.ro

Self Joins

- *Pentru a face join de la o tabelă la ea însăşi, tabelei îi sunt asociate 2 alias-uri.*

- Astfel, pentru baza de date, există în aparenţă 2 tabele.

copyright@www.adrian.runceanu.ro

Self Joins

EMP

EMPNO	ENAME	MGR
7839	KING	-
7698	BLAKE	7839
7782	CLARK	7839
7566	JONES	7839
7788	SCOTT	7566
7902	FORD	7566
7369	SMITH	7902
7499	ALLEN	7698
7521	WARD	7698
7654	MARTIN	7698
7844	TURNER	7698
7876	ADAMS	7788
7900	JAMES	7698
7934	MILLER	7782

MGR in tabela angajatilor are aceeasi valoare in EMPNO din tabela managerilor

EMP

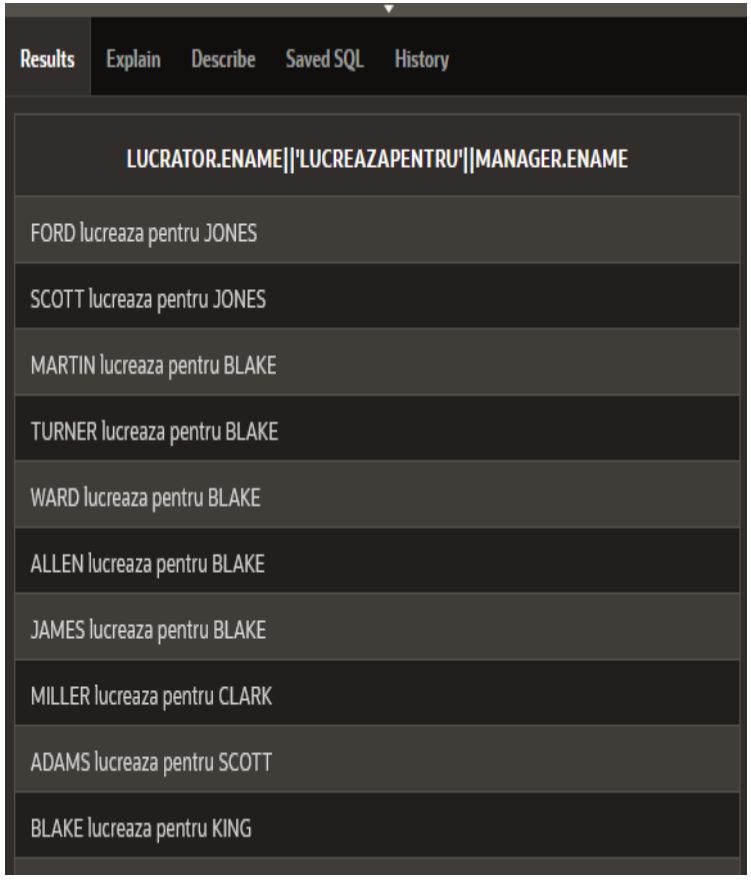
EMPNO	ENAME
7839	KING
7698	BLAKE
7782	CLARK
7566	JONES
7788	SCOTT
7902	FORD
7369	SMITH
7499	ALLEN
7521	WARD
7654	MARTIN
7844	TURNER
7876	ADAMS
7900	JAMES
7934	MILLER

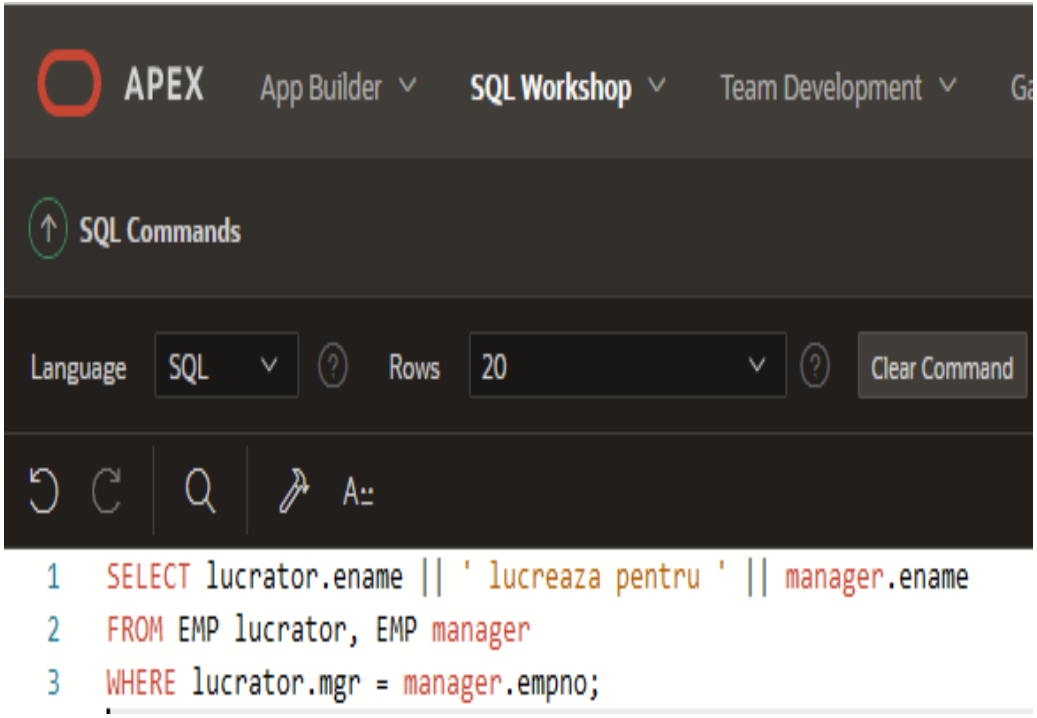
copyright@www.adrian.runceanu.ro

Self Joins

Exemplu:

```
SELECT lucrator.ename || ' lucreaza pentru ' || manager.ename
FROM EMP lucrator, EMP manager
WHERE lucrator.mgr = manager.empno;
```





copyright@www.adrian.runceanu.ro

Self Joins

APLICAŢII

1) Afişaţi numele şi numărul pentru fiecare angajat împreună cu numele şi numărul managerului. Denumiţi coloanele: Angajat, Ang#, Manager şi Mgr#.

2) Modificaţi interogarea 1 pentru a afişa toţi angajaţii şi managerii lor chiar dacă un angajat nu are un manager. Ordonaţi lista obţinută alfabetic, după numele angajaţilor.

copyright@www.adrian.runceanu.ro

Self Joins

1) Afişaţi numele şi numărul pentru fiecare angajat împreună cu numele şi numărul managerului.
Denumiţi coloanele: Angajat, Ang#, Manager şi Mgr#.

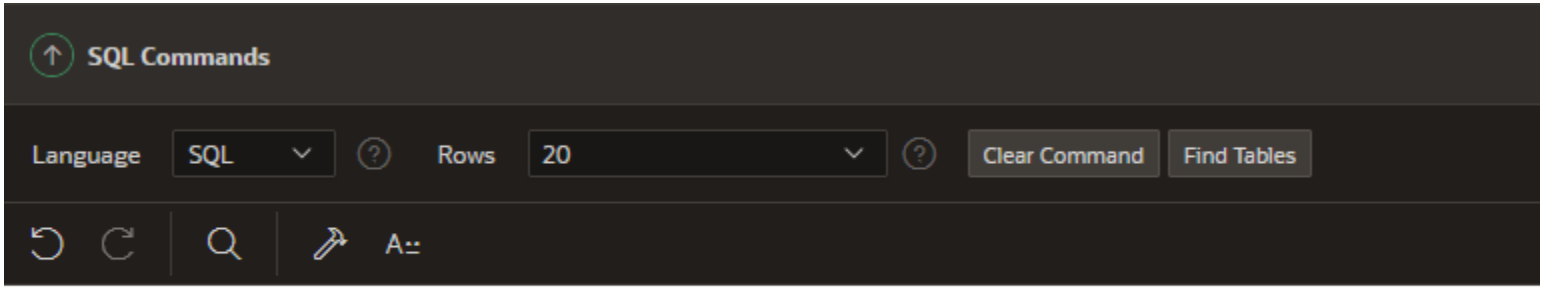
```
SELECT      a.ename AS "Angajat", a.empno AS
   "Ang#", m.ename AS "Manager", m.empno AS
   "Mgr#"
FROM EMP a, EMP m
WHERE a.empno = m.empno;
```

copyright@www.adrian.runceanu.ro

Self Joins

1) Afişaţi numele şi numărul pentru fiecare angajat împreună cu numele şi numărul managerului.

Denumiţi coloanele: Angajat, Ang#, Manager şi Mgr#.




```
1   SELECT  a.ename AS "Angajat", a.empno AS "Ang#", m.ename AS "Manager", m.empno AS "Mgr#"
2   FROM EMP a, EMP m
3   WHERE a.empno = m.empno;
```

Results | Explain | Describe | Saved SQL | History

Angajat	Ang#	Manager	Mgr#
KING	7839	KING	7839
BLAKE	7698	BLAKE	7698
CLARK	7782	CLARK	7782
JONES	7566	JONES	7566
SCOTT	7788	SCOTT	7788
FORD	7902	FORD	7902
SMITH	7369	SMITH	7369
ALLEN	7499	ALLEN	7499
WARD	7521	WARD	7521
MARTIN	7654	MARTIN	7654
TURNER	7844	TURNER	7844
ADAMS	7876	ADAMS	7876
JAMES	7900	JAMES	7900
MILLER	7934	MILLER	7934

14 rows returned in 0.00 seconds Download

copyright@www.adrian.runceanu.ro

Self Joins

2) Modificaţi interogarea 1 pentru a afişa toţi angajaţii şi managerii lor chiar dacă un angajat nu are un manager.

Ordonaţi lista obţinută alfabetic, după numele angajaţilor.

```
SELECT a.ename AS "Angajat", a.empno AS
   "Ang#", m.ename AS "Manager", m.empno AS
   "Mgr#"
FROM EMP a, EMP m
WHERE a.empno = m.empno
ORDER BY a.ename;
```

copyright@www.adrian.runceanu.ro

Self Joins

SQL Commands

Language SQL Rows 20 Clear Command Find Tables

```
1  SELECT  a.ename AS "Angajat", a.empno AS "Ang#", m.ename AS "Manager", m.empno AS "Mgr#"
2  FROM EMP a, EMP m
3  WHERE a.empno = m.empno
4  ORDER BY a.ename;
```

Results | Explain | Describe | Saved SQL | History

Angajat	Ang#	Manager	Mgr#
ADAMS	7876	ADAMS	7876
ALLEN	7499	ALLEN	7499
BLAKE	7698	BLAKE	7698
CLARK	7782	CLARK	7782
FORD	7902	FORD	7902
JAMES	7900	JAMES	7900
JONES	7566	JONES	7566
KING	7839	KING	7839
MARTIN	7654	MARTIN	7654
MILLER	7934	MILLER	7934
SCOTT	7788	SCOTT	7788
SMITH	7369	SMITH	7369
TURNER	7844	TURNER	7844
WARD	7521	WARD	7521

14 rows returned in 0.01 seconds Download

copyright@www.adrian.runceanu.ro

Limbajul SQL

6.2. Cereri din mai multe tabele (JOIN-uri)

6.2.1. JOIN-urile proprietatea ORACLE

6.2.1.1. Cartesian Product

6.2.1.2. Equijoin

6.2.1.3. Non-equijoin

6.2.1.4. Outer join

6.2.1.5. Self join

6.2.2. JOIN-urile ANSI/ISO SQL99

6.2.2.1. Cross join

6.2.2.2. Natural join

6.2.2.3. Using clause

6.2.2.4. Full (two sided) outer joins

6.2.2.5. Arbitrary join conditions for outer joins

6.2.3. Operatorii pe mulţimi

copyright@www.adrian.runceanu.ro

Cross Joins and Natural Joins

NATURAL JOIN

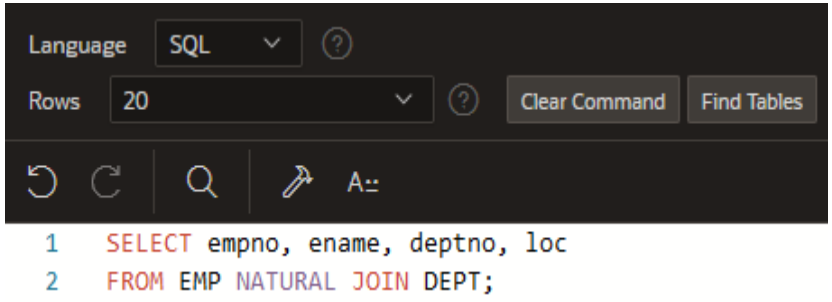
- Este un ANSI/ISO SQL:1999 join echivalent cu equijoin-ul.
- *Se face pe baza tuturor coloanelor care au acelaşi nume din 2 tabele; coloanele respective trebuie să aibă acelaşi tip de date.*
- Sunt selectate liniile care au valori egale în toate coloanele corespondente.

copyright@www.adrian.runceanu.ro

Cross Joins and Natural Joins

Exemplu:

SELECT empno, ename, deptno, loc
FROM EMP **NATURAL JOIN** DEPT;



Results | Explain | Describe | Saved SQL | History

7782	CLARK	10	NEW YORK
7934	MILLER	10	NEW YORK
7839	KING	10	NEW YORK
7902	FORD	20	DALLAS
7788	SCOTT	20	DALLAS
7566	JONES	20	DALLAS
7369	SMITH	20	DALLAS
7876	ADAMS	20	DALLAS
7521	WARD	30	CHICAGO
7654	MARTIN	30	CHICAGO
7844	TURNER	30	CHICAGO
7900	JAMES	30	CHICAGO
7499	ALLEN	30	CHICAGO
7698	BLAKE	30	CHICAGO

14 rows returned in 0.03 seconds Download

copyright@www.adrian.runceanu.ro

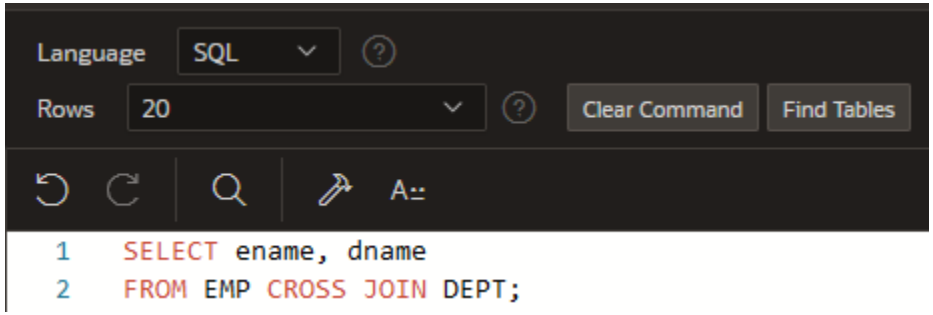
Cross Joins and Natural Joins

CROSS JOIN

Realizează produsul cartezian pentru două tabele.

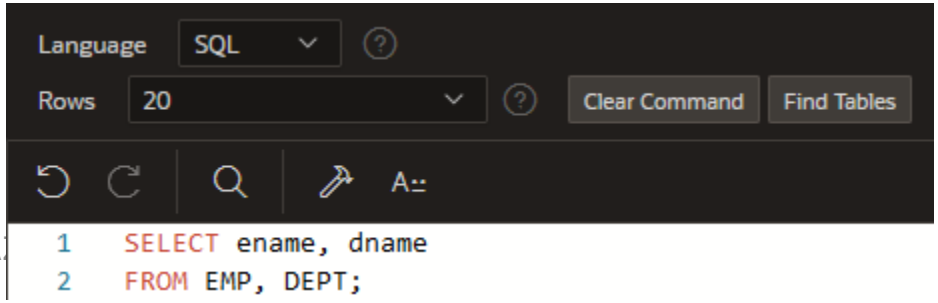
Exemplu:

SELECT ename, dname
FROM EMP **CROSS JOIN** DEPT;



este echivalentă cu instrucţiunea:

SELECT ename, dname
FROM EMP, DEPT;



copyright@www.adrian.runceanu.ro

Cross Joins and Natural Joins

APLICAŢII

1) Creaţi un cross-join care afişează numele şi denumirea departmentului din tabelele **EMP** şi **DEPT**.

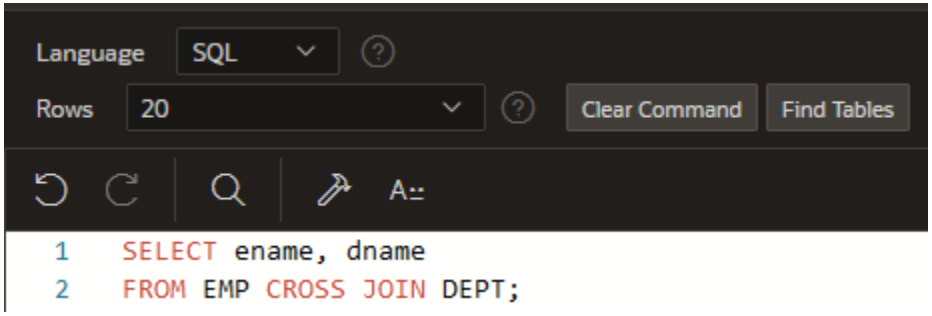
2) Creaţi o interogare care foloseşte un natural join pentru a pune în legătură tabelele **DEPT** şi **EMP**. Afişaţi id-ul, denumirea departamentului şi oraşul.

copyright@www.adrian.runceanu.ro

Cross Joins and Natural Joins

1) Creaţi un cross-join care afişează numele şi denumirea departmentului din tabelele **EMP** şi **DEPT**.

SELECT ename, dname
FROM EMP **CROSS JOIN** DEPT;

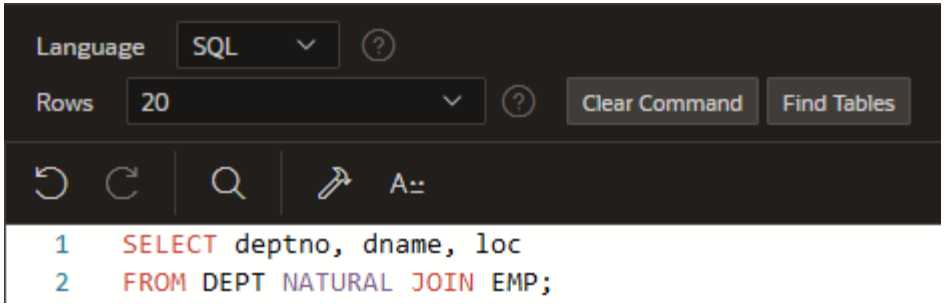


copyright@www.adrian.runceanu.ro

Cross Joins and Natural Joins

2) Creaţi o interogare care foloseşte un natural join pentru a pune în legătură tabelele **DEPT** şi **EMP**. Afişaţi id-ul şi denumirea departamentului, oraşul.

SELECT deptno, dname, loc
FROM DEPT
NATURAL JOIN EMP;



copyright@www.adrian.runceanu.ro

Limbajul SQL

6.2. Cereri din mai multe tabele (JOIN-uri)

6.2.1. JOIN-urile proprietatea ORACLE

6.2.1.1. Cartesian Product

6.2.1.2. Equijoin

6.2.1.3. Non-equijoin

6.2.1.4. Outer join

6.2.1.5. Self join

6.2.2. JOIN-urile ANSI/ISO SQL99

6.2.2.1. Cross join

6.2.2.2. Natural join

6.2.2.3. Using clause

6.2.2.4. Full (two sided) outer joins

6.2.2.5. Arbitrary join conditions for outer joins

6.2.3. Operatorii pe mulţimi

copyright@www.adrian.runceanu.ro

Join Clauses

În această parte vom studia:

1. folosirea clauzelor **USING** şi **ON**
2. realizarea unui join cu 3 tabele

1. Clauza USING

Într-un natural join, *dacă tabelele au coloane cu acelaşi nume dar tipuri diferite*, se produce eroare.

Pentru a evita această situaţie, clauza **JOIN** *se înlocuieşte* cu clauza **USING**.

copyright@www.adrian.runceanu.ro

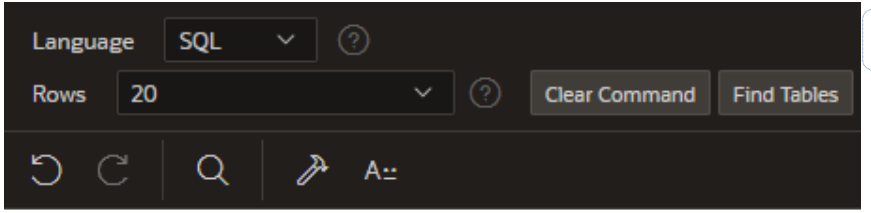
Join Clauses

Clauza **USING** specifică coloanele care ar trebui folosite pentru pentru equijoin.

Coloana specificată în clauza **USING** nu trebuie să aibă nici un specificator (nume de tabela sau alias), în nici o parte din instrucţiunea SELECT.

copyright@www.adrian.runceanu.ro

Join Clauses




```
1  SELECT e.empno, e.ename, d.loc
2  FROM EMP e
3  JOIN DEPT d
4  USING (deptno);
```

Exemplu:

SELECT e.empno, e.ename, d.loc
FROM EMP e
JOIN DEPT d
USING (deptno);

Results | Explain | Describe | Saved SQL | History

EMPNO	ENAME	LOC
7782	CLARK	NEW YORK
7934	MILLER	NEW YORK
7839	KING	NEW YORK
7902	FORD	DALLAS
7788	SCOTT	DALLAS
7566	JONES	DALLAS
7369	SMITH	DALLAS
7876	ADAMS	DALLAS
7521	WARD	CHICAGO
7654	MARTIN	CHICAGO
7844	TURNER	CHICAGO
7900	JAMES	CHICAGO
7499	ALLEN	CHICAGO
7698	BLAKE	CHICAGO

14 rows returned in 0.01 seconds Download

copyright@www.adrian.runceanu.ro

Join Clauses

Clauza ON

Dacă coloanele folosite pentru join au *denumiri diferite* sau *dacă sunt folosiţi operatorii*: **<, >** sau BETWEEN, atunci nu putem folosi clauza USING.

În această situaţie se foloseşte clauza **ON**.

Aceasta permite specificarea unei game variate de condiţii pentru join-uri.

De asemenea, clauza **ON** ne permite să folosim **WHERE** pentru a restricţiona linii dintr-o tabelă sau din ambele tabele.

copyright@www.adrian.runceanu.ro

Join Clauses

Exemple:

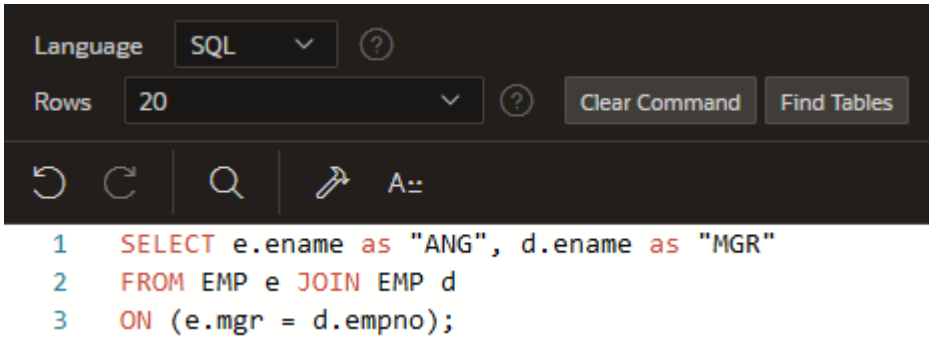
1) **SELECT** e.ename as "ANG", d.ename as "MGR"
FROM EMP e **JOIN** EMP d
ON (e.mgr = d.empno);

Se realizează un **self-join** pentru a selecta acei angajaţi care sunt şi manageri.

copyright@www.adrian.runceanu.ro

Join Clauses

1) **SELECT** e.ename as "ANG",
d.ename as "MGR"
FROM EMP e **JOIN** EMP d
ON (e.mgr = d.empno);



Results | Explain | Describe | Saved SQL | History

ANG	MGR
FORD	JONES
SCOTT	JONES
MARTIN	BLAKE
TURNER	BLAKE
WARD	BLAKE
ALLEN	BLAKE
JAMES	BLAKE
MILLER	CLARK
ADAMS	SCOTT
BLAKE	KING
CLARK	KING
JONES	KING
SMITH	FORD

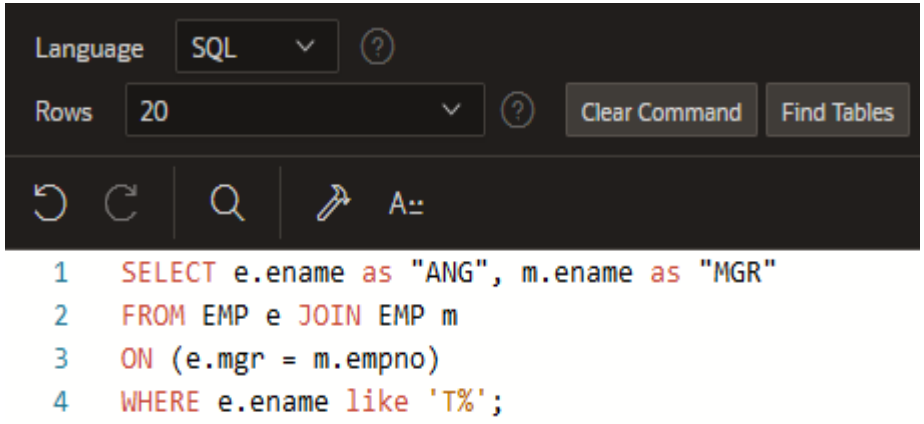
13 rows returned in 0.01 seconds Download

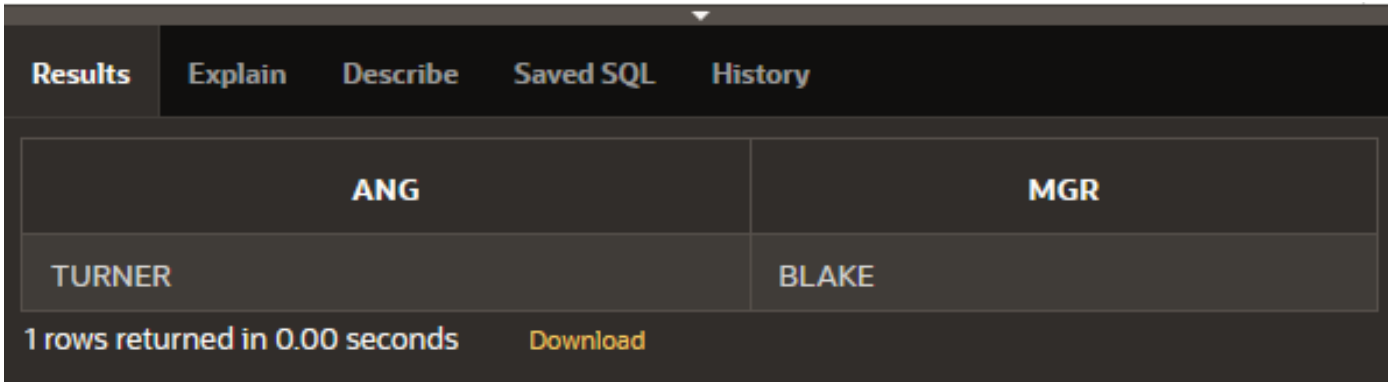
copyright@www.adrian.runceanu.ro

Join Clauses

2) – folosirea clauzei **WHERE**

```
SELECT e.ename as "ANG", m.ename as "MGR"
FROM EMP e JOIN EMP m
ON (e.mgr = m.empno)
WHERE e.ename like 'T%';
```





copyright@www.adrian.runceanu.ro

Limbajul SQL

6.2. Cereri din mai multe tabele (JOIN-uri)

6.2.1. JOIN-urile proprietatea ORACLE

6.2.1.1. Cartesian Product

6.2.1.2. Equijoin

6.2.1.3. Non-equijoin

6.2.1.4. Outer join

6.2.1.5. Self join

6.2.2. JOIN-urile ANSI/ISO SQL99

6.2.2.1. Cross join

6.2.2.2. Natural join

6.2.2.3. Using clause

6.2.2.4. Full (two sided) outer joins

6.2.2.5. Arbitrary join conditions for outer joins

6.2.3. Operatorii pe mulţimi

copyright@www.adrian.runceanu.ro

Join Clauses

Realizarea unui join cu 3 tabele

Ambele clauze (**ON** si **USING**) se pot folosi pentru un astfel de join.

```
SELECT empno, loc, dname, city
FROM EMP e JOIN DEPT d
ON (d.deptno = e.deptno)
JOIN locations l
ON (d.loc = l.loc)
```

copyright@www.adrian.runceanu.ro

Join Clauses

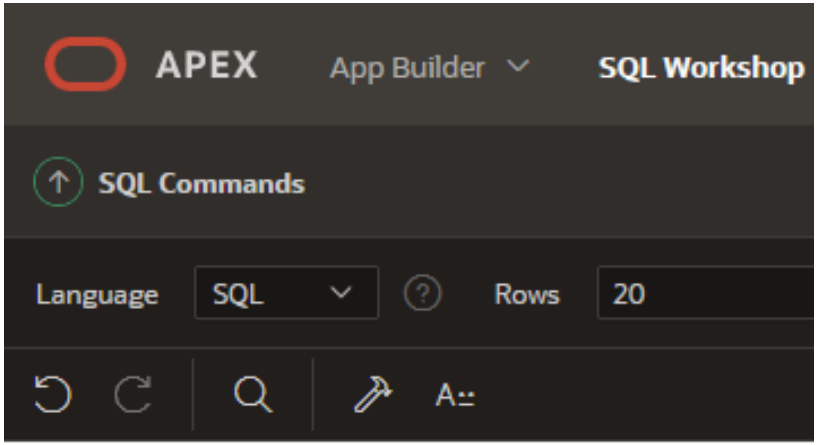
Comparatie intre join-urile proprietate ORACLE si join-urile ANSI/ISO SQL 1999

Join-uri proprietate ORACLE	Join-uri ANSI/ISO SQL 1999
Produs cartezian	Cross Join
Equijoin	Natural Join (daca coloanele de join au acelasi nume si acelasi tip de date)
	Clauza USING (daca coloanele de join au acelasi nume dar tipuri de date diferite)
	Clauza ON (daca coloanele au nume diferite)
Non-equijoin	Clauza ON

copyright@www.adrian.runceanu.ro

Join Clauses

SELECT e.empno, e.ename, d.loc
FROM EMP e **JOIN** DEPT d
USING(deptno);




```
1   SELECT e.empno, e.ename, d.loc
2   FROM EMP e JOIN DEPT d
3   USING(deptno);
```

Results | Explain | Describe | Saved SQL | History

EMPNO	ENAME	LOC
7782	CLARK	NEW YORK
7934	MILLER	NEW YORK
7839	KING	NEW YORK
7902	FORD	DALLAS
7788	SCOTT	DALLAS
7566	JONES	DALLAS
7369	SMITH	DALLAS
7876	ADAMS	DALLAS
7521	WARD	CHICAGO
7654	MARTIN	CHICAGO
7844	TURNER	CHICAGO
7900	JAMES	CHICAGO
7499	ALLEN	CHICAGO
7698	BLAKE	CHICAGO

14 rows returned in 0.00 seconds Download

copyright@www.adrian.runceanu.ro

Limbajul SQL

6.2. Cereri din mai multe tabele (JOIN-uri)

6.2.1. JOIN-urile proprietatea ORACLE

6.2.1.1. Cartesian Product

6.2.1.2. Equijoin

6.2.1.3. Non-equijoin

6.2.1.4. Outer join

6.2.1.5. Self join

6.2.2. JOIN-urile ANSI/ISO SQL99

6.2.2.1. Cross join

6.2.2.2. Natural join

6.2.2.3. Using clause

6.2.2.4. Full (two sided) outer joins

6.2.2.5. Arbitrary join conditions for outer joins

6.2.3. Operatorii pe mulţimi

copyright@www.adrian.runceanu.ro

Inner vs. Outer Joins

În ANSI-99 SQL, *un join cu 2 sau mai multe tabele care returnează doar liniile care se potrivesc* se numeşte **inner join**.

Atunci când *un join returnează atât liniile care se potrivesc cât şi cele care nu se potrivesc*, acesta se numeşte **outer join**.

Sunt trei tipuri de outer join în ANSI/ISO SQL:

1. **LEFT OUTER JOIN**
2. **RIGHT OUTER JOIN**
3. **FULL OUTER JOIN**

copyright@www.adrian.runceanu.ro

Inner vs. Outer Joins

1. LEFT OUTER JOIN

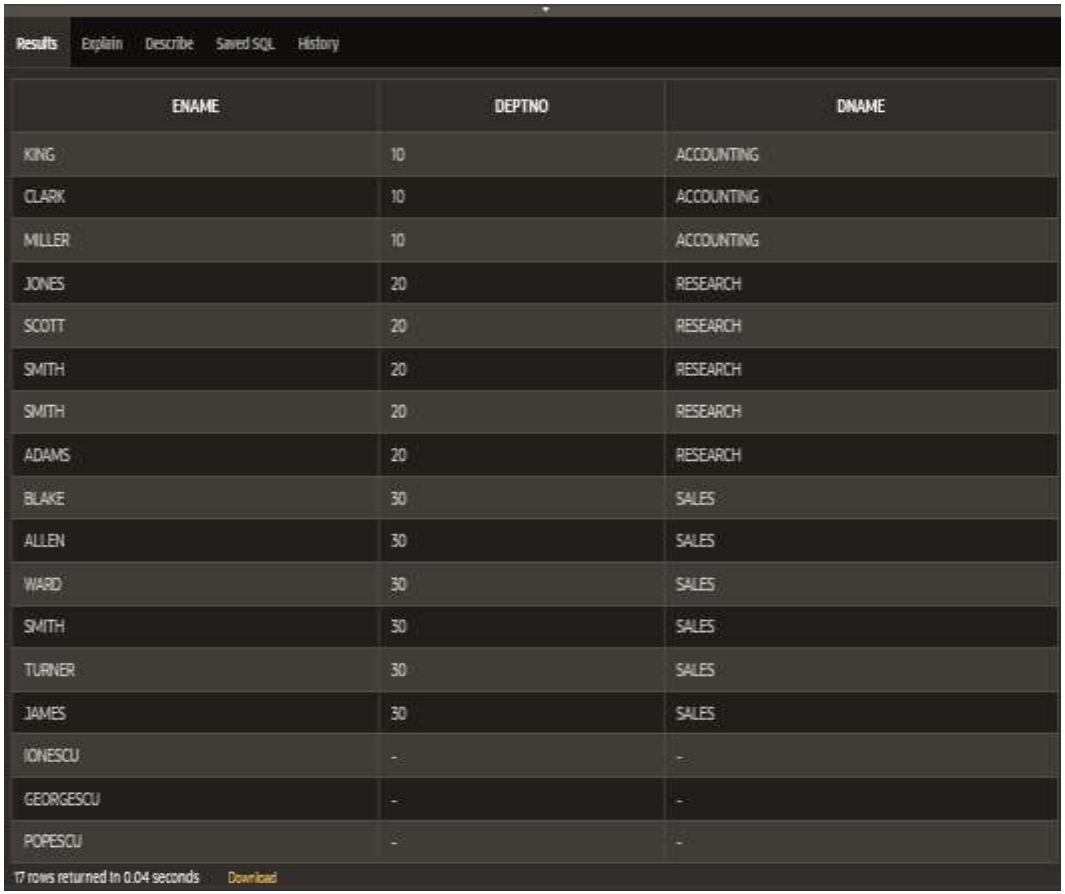
Sunt afişaţi şi acei angajaţi care nu au desemnat un id_dept (tabela DEPT este cea deficitară).

```
SELECT e.ename, e.deptno, d.dname
FROM EMP e
LEFT OUTER JOIN DEPT d
ON (e.deptno = d.deptno);
```

copyright@www.adrian.runceanu.ro

Inner vs. Outer Joins

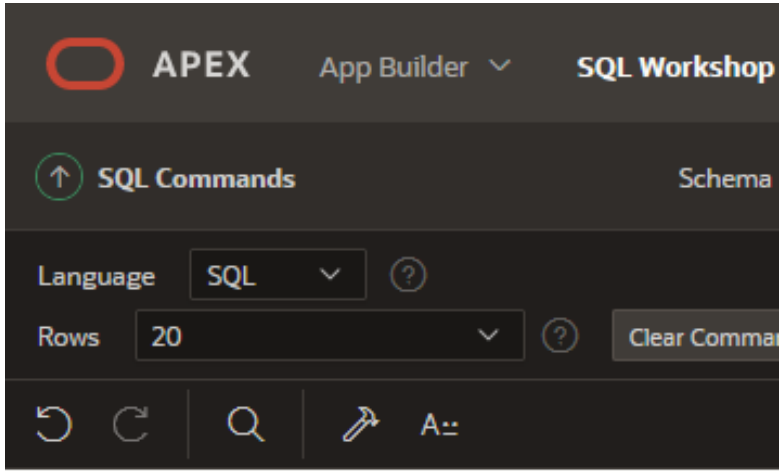
1. LEFT OUTER JOIN



Results | Explain | Describe | Saved SQL | History

ENAME	DEPTNO	DNAME
KING	10	ACCOUNTING
CLARK	10	ACCOUNTING
MILLER	10	ACCOUNTING
JONES	20	RESEARCH
SCOTT	20	RESEARCH
SMITH	20	RESEARCH
SMITH	20	RESEARCH
ADAMS	20	RESEARCH
BLAKE	30	SALES
ALLEN	30	SALES
WARD	30	SALES
SMITH	30	SALES
TURNER	30	SALES
JAMES	30	SALES
IONESCU	-	-
GEORGESCU	-	-
POPESCU	-	-

17 rows returned in 0.04 seconds Download




```
1   SELECT e.ename, e.deptno, d.dname
2   FROM EMP e
3   LEFT OUTER JOIN DEPT d
4   ON (e.deptno = d.deptno);
```

copyright@www.adrian.runceanu.ro

Inner vs. Outer Joins

2. RIGHT OUTER JOIN

Sunt afişate şi acele DEPT care nu au angajaţi.

```
SELECT e.ename, e.deptno, d.dname
FROM EMP e
RIGHT OUTER JOIN DEPT d
ON (e.deptno = d.deptno);
```

copyright@www.adrian.runceanu.ro

Inner vs. Outer Joins

2. RIGHT OUTER JOIN

Sunt afişate şi acele DEPT care nu au angajaţi.

```
SELECT e.ename, e.deptno, d.dname
FROM EMP e
RIGHT OUTER JOIN DEPT d
ON (e.deptno = d.deptno);
```

copyright@www.adrian.runceanu.ro

Inner vs. Outer Joins

2. RIGHT OUTER JOIN

Sunt afişate şi acele DEPT care nu au angajaţi.

Results | Explain | Describe | Saved SQL | History

ENAME	DEPTNO	DNAME
CLARK	10	ACCOUNTING
MILLER	10	ACCOUNTING
KING	10	ACCOUNTING
FORD	20	RESEARCH
SCOTT	20	RESEARCH
JONES	20	RESEARCH
SMITH	20	RESEARCH
ADAMS	20	RESEARCH
WARD	30	SALES
MARTIN	30	SALES
TURNER	30	SALES
JAMES	30	SALES
ALLEN	30	SALES
BLAKE	30	SALES
-	-	OPERATIONS
-	-	MARKETING
-	-	PRODUCTION

SQL Commands

Language SQL

Rows 20

Clear Co

```
1  SELECT e.ename, e.deptno, d.dname
2  FROM EMP e
3  RIGHT OUTER JOIN DEPT d
4  ON (e.deptno = d.deptno);
```

copyright@www.adrian.runceanu.ro

Inner vs. Outer Joins

3. FULL OUTER JOIN

Returnează atât liniile care se potrivesc cât şi cele care nu se potrivesc din ambele tabele.

Spre deosebire de **outer join-ul** proprietatea Oracle, care nu permitea folosirea operatorului **(+)** în ambele părţi ale clauzei WHERE, **full outer join-ul** permite acest lucru.

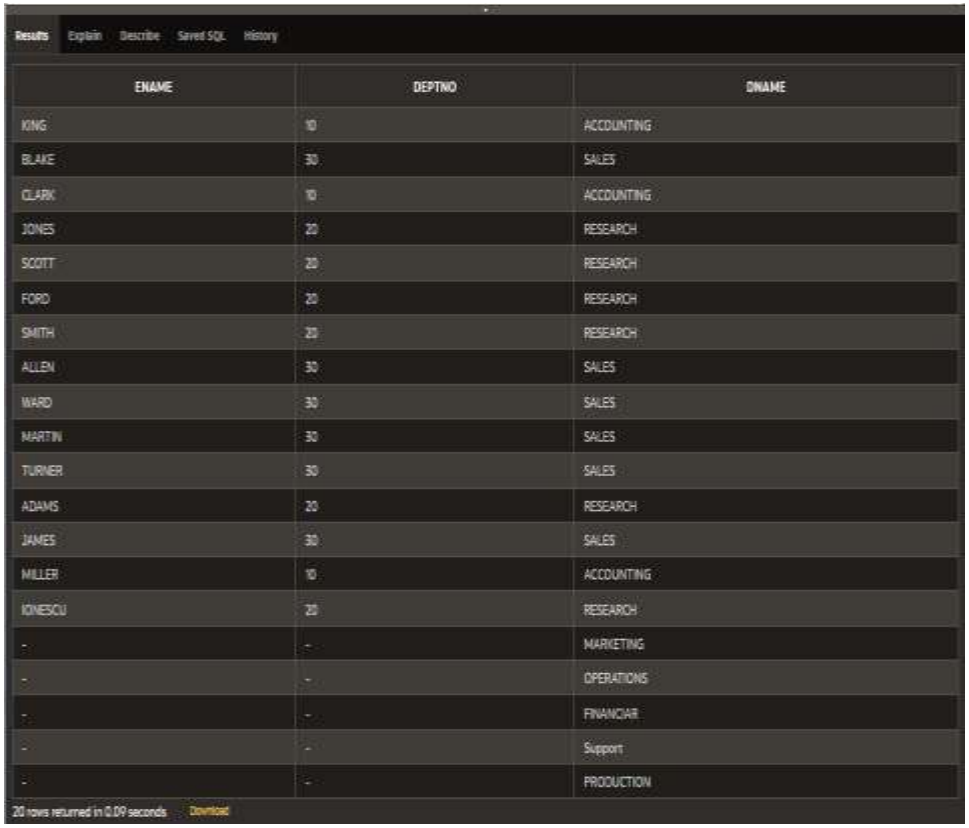
```
SELECT e.ename, e.deptno, d.dname
FROM EMP e
FULL OUTER JOIN DEPT d
ON (e.deptno = d.deptno);
```

copyright@www.adrian.runceanu.ro

Inner vs. Outer Joins

3. FULL OUTER JOIN

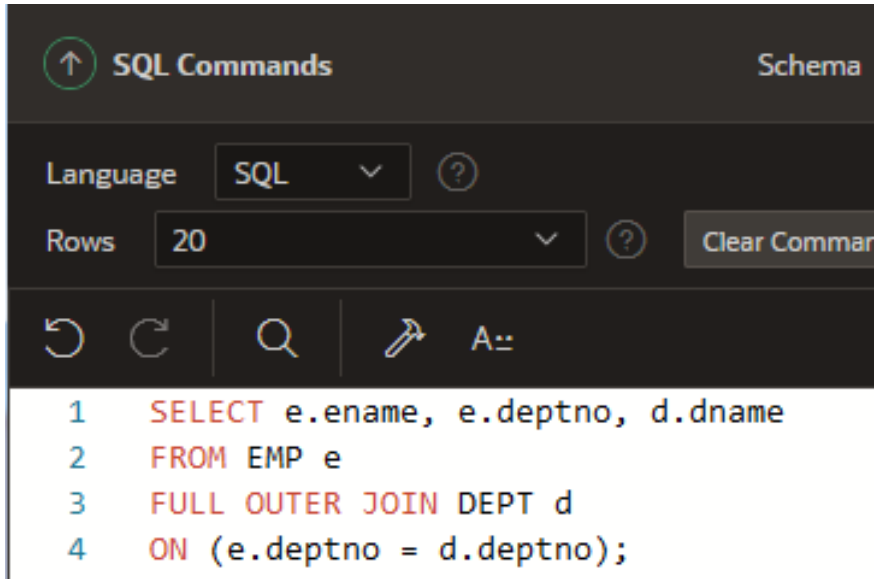
Returnează atât liniile care se potrivesc cât şi cele care nu se potrivesc din ambele tabele.



Results | Explain | Describe | Saved SQL | History

ENAME	DEPTNO	DNAME
KING	10	ACCOUNTING
BLAKE	30	SALES
CLARK	10	ACCOUNTING
JONES	20	RESEARCH
SCOTT	20	RESEARCH
FORD	20	RESEARCH
SMITH	20	RESEARCH
ALLEN	30	SALES
WARD	30	SALES
MARTIN	30	SALES
TURNER	30	SALES
ADAMS	20	RESEARCH
JAMES	30	SALES
MILLER	10	ACCOUNTING
IONESCU	20	RESEARCH
-	-	MARKETING
-	-	OPERATIONS
-	-	FINANCIAR
-	-	Support
-	-	PRODUCTION

20 rows returned in 0.09 seconds Download



copyright@www.adrian.runceanu.ro

Limbajul SQL

6.2. Cereri din mai multe tabele (JOIN-uri)

6.2.1. JOIN-urile proprietatea ORACLE

6.2.1.1. Cartesian Product

6.2.1.2. Equijoin

6.2.1.3. Non-equijoin

6.2.1.4. Outer join

6.2.1.5. Self join

6.2.2. JOIN-urile ANSI/ISO SQL99

6.2.2.1. Cross join

6.2.2.2. Natural join

6.2.2.3. Using clause

6.2.2.4. Full (two sided) outer joins

6.2.2.5. Arbitrary join conditions for outer joins

6.2.3. Operatorii pe mulţimi

copyright@www.adrian.runceanu.ro

Inner vs. Outer Joins

APLICAŢII

1) Afişaţi numele şi denumirea departamentului pentru toţi angajaţii, inclusiv pentru cei care nu sunt desemnaţi la nici un departament.
2) Afişaţi numele şi denumirea departamentului pentru toţi angajaţii, inclusiv acele departamente care nu au nici un angajat asociat.
3) Afişaţi numele şi denumirea departamentului pentru toţi EMP, inclusiv acele departamente care nu au nici un angajat asociat şi acei angajaţi care nu sunt desemnaţi nici unui departament.

copyright@www.adrian.runceanu.ro

Inner vs. Outer Joins

1) Afişaţi numele şi denumirea departamentului pentru toţi angajaţii, inclusiv pentru cei care nu sunt desemnaţi la nici un departament.

```
SELECT e.ename, e.job, d.dname
FROM EMP e
LEFT OUTER JOIN DEPT d
ON (e.deptno = d.deptno);
```

copyright@www.adrian.runceanu.ro

Inner vs. Outer Joins

SQL Commands

Language SQL

Rows 20

```
1  SELECT e.ename, e.job, d.dname
2  FROM EMP e
3  LEFT OUTER JOIN DEPT d
4  ON (e.deptno = d.deptno);
```

Results | Explain | Describe | Saved SQL | History

SCOTT	ANALYST	RESEARCH
FORD	ANALYST	RESEARCH
SMITH	CLERK	RESEARCH
ADAMS	CLERK	RESEARCH
IONESCU	SALESMAN	RESEARCH
BLAKE	MANAGER	SALES
ALLEN	SALESMAN	SALES
WARD	SALESMAN	SALES
MARTIN	SALESMAN	SALES
TURNER	SALESMAN	SALES
JAMES	CLERK	SALES

15 rows returned in 0.09 seconds Download

copyright@www.adrian.runceanu.ro

Inner vs. Outer Joins

2) Afişaţi numele şi denumirea departamentului pentru toţi angajaţii, inclusiv acele departamente care nu au nici un angajat asociat.

```
SELECT e.ename, e.sal, d.dname
FROM EMP e
RIGHT OUTER JOIN DEPT d
ON (e.deptno = d.deptno);
```

copyright@www.adrian.runceanu.ro

Inner vs. Outer Joins

SQL Commands

Language SQL

Rows 20

```
1   SELECT e.ename, e.sal, d.dname
2   FROM EMP e
3   RIGHT OUTER JOIN DEPT d
4   ON (e.deptno = d.deptno);
```

Results | Explain | Describe | Saved SQL | History

MARTIN	1250	SALES
TURNER	1500	SALES
ADAMS	1100	RESEARCH
JAMES	950	SALES
MILLER	1300	ACCOUNTING
IONESCU	1000	RESEARCH
-	-	MARKETING
-	-	OPERATIONS
-	-	FINANCIAR
-	-	Support
-	-	PRODUCTION

20 rows returned in 0.01 seconds Download

copyright@www.adrian.runceanu.ro

Inner vs. Outer Joins

3) Afişaţi numele şi denumirea departamentului pentru toţi EMP, inclusiv acele departamente care nu au nici un angajat asociat şi acei angajaţi care nu sunt desemnaţi nici unui departament.

```
SELECT e.ename, e.job, d.dname
FROM EMP e
FULL OUTER JOIN DEPT d
ON (e.deptno = d.deptno);
```

copyright@www.adrian.runceanu.ro

Inner vs. Outer Joins

SQL Commands

Language SQL

Rows 20

```
1  SELECT e.ename, e.job, d.dname
2  FROM EMP e
3  FULL OUTER JOIN DEPT d
4  ON (e.deptno = d.deptno);
```

Results | Explain | Describe | Saved SQL | History

MARTIN	SALESMAN	SALES
TURNER	SALESMAN	SALES
ADAMS	CLERK	RESEARCH
JAMES	CLERK	SALES
MILLER	CLERK	ACCOUNTING
IONESCU	SALESMAN	RESEARCH
-	-	MARKETING
-	-	OPERATIONS
-	-	FINANCIAR
-	-	Support
-	-	PRODUCTION

20 rows returned in 0.01 seconds Download

copyright@www.adrian.runceanu.ro

Inner vs. Outer Joins

ALTE APLICAŢII:

1. Afişaţi numele şi numărul pentru angajaţi împreună cu numele şi numărul managerului. Denumiţi coloanele astfel: Angajat, Ang#, Manager, şi Mgr#.

2. Modificaţi rezultatul de la problema 1 astfel încât să fie afişaţi toţi angajaţii, inclusiv aceia care nu au manager. Ordonaţi rezultatele după numărul angajatului.

copyright@www.adrian.runceanu.ro

Inner vs. Outer Joins

1. Afişaţi numele şi numărul pentru angajaţi împreună cu numele şi numărul managerului. Denumiţi coloanele astfel: Angajat, Ang#, Manager, şi Mgr#.

```
SELECT      w.ename AS "Angajat",
            w.empno AS "Ang#",
            m.ename AS "Manager",
            m.empno AS "Mgr#"
FROM EMP w JOIN EMP m
ON (w.mgr = m.empno );
```

copyright@www.adrian.runceanu.ro

Inner vs. Outer Joins

SQL Commands Schema: LUCRU

Language: SQL Rows: 20 Clear Command Find Tables Save

```
1  SELECT  w.ename AS "Angajat", w.empno AS "Ang#", m.ename AS "Manager", m.empno AS "Mgr#"
2  FROM EMP w JOIN EMP m
3  ON (w.mgr = m.empno );
```

Results | Explain | Describe | Saved SQL | History

Angajat	Ang#	Manager	Mgr#
FORD	7902	JONES	7566
SCOTT	7788	JONES	7566
MARTIN	7654	BLAKE	7698
TURNER	7844	BLAKE	7698
WARD	7521	BLAKE	7698
ALLEN	7499	BLAKE	7698
JAMES	7900	BLAKE	7698
MILLER	7934	CLARK	7782
ADAMS	7876	SCOTT	7788
BLAKE	7698	KING	7839
CLARK	7782	KING	7839
JONES	7566	KING	7839
SMITH	7369	FORD	7902

13 rows returned in 0.01 seconds Download

copyright@www.adrian.runceanu.ro

Inner vs. Outer Joins

2. Modificaţi rezultatul de la problema 1 astfel încât să fie afişaţi toţi angajaţii, inclusiv aceia care nu au manager. Ordonaţi rezultatele după numărul angajatului.

```
SELECT      w.ename AS "Angajat",
            w.empno AS "Ang#",
            m.ename AS "Manager",
            m.empno AS "Mgr#"
FROM EMP w LEFT OUTER JOIN EMP m
ON (w.mgr = m.empno );
ORDER BY w.empno;
```

copyright@www.adrian.runceanu.ro

Inner vs. Outer Joins

SQL Commands Schema LUCRU

Language SQL Rows 20 Clear Command Find Tables Save

```
1  SELECT  w.ename AS "Angajat", w.empno AS "Ang#", m.ename AS "Manager", m.empno AS "Mgr#"
2  FROM EMP w LEFT OUTER JOIN EMP m
3  ON (w.mgr = m.empno )
4  ORDER BY w.empno;
```

Results | Explain | Describe | Saved SQL | History

Angajat	Ang#	Manager	Mgr#
SMITH	7369	FORD	7902
ALLEN	7499	BLAKE	7698
WARD	7521	BLAKE	7698
JONES	7566	KING	7839
MARTIN	7654	BLAKE	7698
BLAKE	7698	KING	7839
CLARK	7782	KING	7839
SCOTT	7788	JONES	7566
KING	7839	-	-
TURNER	7844	BLAKE	7698
ADAMS	7876	SCOTT	7788
JAMES	7900	BLAKE	7698
FORD	7902	JONES	7566
MILLER	7934	CLARK	7782

14 rows returned in 0.02 seconds Download

copyright@www.adrian.runceanu.ro

Limbajul SQL

6.2. Cereri din mai multe tabele (JOIN-uri)

6.2.1. JOIN-urile proprietatea ORACLE

6.2.1.1. Cartesian Product

6.2.1.2. Equijoin

6.2.1.3. Non-equijoin

6.2.1.4. Outer join

6.2.1.5. Self join

6.2.2. JOIN-urile ANSI/ISO SQL99

6.2.2.1. Cross join

6.2.2.2. Natural join

6.2.2.3. Using clause

6.2.2.4. Full (two sided) outer joins

6.2.2.5. Arbitrary join conditions for outer joins

6.2.3. Operatorii pe mulţimi

copyright@www.adrian.runceanu.ro

Operatorii pe mulţimi

Operatorii pe mulţimi combină rezultatele obţinute din două sau mai multe interogări.

Cererile care conţin operatori pe mulţimi se numesc ***cereri compuse***.

Există patru operatori pe mulţimi:

1. **UNION**
2. **UNION ALL**
3. **INTERSECT**
4. **MINUS**

copyright@www.adrian.runceanu.ro

Operatorii pe mulţimi

- Toţi operatorii pe mulţimi au aceeaşi precedenţă.
- Dacă o instrucţiune ***SQL*** conţine mai mulţi operatori pe mulţimi, *server*-ul ***Oracle*** evaluează cererea de la stânga la dreapta (sau de sus în jos).
- Pentru a schimba această ordine de evaluare, se pot utiliza paranteze.

copyright@www.adrian.runceanu.ro

Operatorii pe mulţimi

- *Operatorul **UNION*** returnează toate liniile selectate de două cereri, eliminând duplicatele.

- Acest operator nu ignoră valorile *null* şi are precedenţă mai mică decât operatorul ***IN***.

copyright@www.adrian.runceanu.ro

Operatorii pe mulţimi

- Operatorul ***UNION ALL*** returnează toate liniile selectate de două cereri, fără a elimina duplicatele.
- Precizările făcute asupra operatorului *UNION* sunt valabile şi în cazul operatorului *UNION ALL*.
- În cererile asupra cărora se aplică *UNION ALL* nu poate fi utilizat cuvântul cheie *DISTINCT*.

copyright@www.adrian.runceanu.ro

Operatorii pe mulţimi

- Operatorul ***INTERSECT*** returnează toate liniile comune cererilor asupra cărora se aplică.

- Acest operator nu ignoră valorile *null*.

copyright@www.adrian.runceanu.ro

Operatorii pe mulţimi

- Operatorul ***MINUS*** determină liniile returnate de prima cerere care nu apar în rezultatul celei de-a doua cereri.

- Pentru ca operatorul *MINUS* să funcţioneze, este necesar ca toate coloanele din clauza *WHERE* să se afle şi în clauza *SELECT*.

copyright@www.adrian.runceanu.ro

Operatorii pe mulţimi

Observaţii:

1. În mod implicit, pentru toţi operatorii cu excepţia lui *UNION ALL*, rezultatul este ordonat crescător după valorile primei coloane din clauza *SELECT*.

2. Pentru o cerere care utilizează operatori pe mulţimi, cu excepţia lui *UNION ALL*, *server*-ul ***Oracle*** elimină liniile duplicat.

copyright@www.adrian.runceanu.ro

Operatorii pe mulţimi

1. În instrucţiunile *SELECT* asupra cărora se aplică operatori pe mulţimi, coloanele selectate trebuie să corespundă ca număr şi tip de date.
2. Nu este necesar ca numele coloanelor să fie identice.
3. Numele coloanelor din rezultat sunt determinate de numele care apar în clauza *SELECT* a primei cereri.

copyright@www.adrian.runceanu.ro

Operatorii pe mulţimi

APLICAŢII:

1. Să se afişeze: codurile departamentelor al căror nume conţine şirul “CC” sau în care lucrează angajaţi având codul job-ului “CLERK”.

```
SELECT deptno "Cod departament"
FROM    EMP
WHERE  UPPER(job) = 'CLERK'
UNION
SELECT deptno
FROM    DEPT
WHERE LOWER(dname) LIKE '%CC%';
```

copyright@www.adrian.runceanu.ro

Operatorii pe mulţimi

APLICAŢII:

1. Să se afişeze: codurile departamentelor al căror nume conţine şirul “CC” sau în care lucrează angajaţi având codul job-ului “CLERK”.

SQL Commands

Language SQL Rows 20

```
1   SELECT deptno "Cod departament"
2   FROM     EMP
3   WHERE  UPPER(job) = 'CLERK'
4   UNION
5   SELECT deptno
6   FROM    DEPT
7   WHERE LOWER(dname) LIKE '%CC%';
```

Results | Explain | Describe | Saved SQL | History

Cod departament
10
20
30

3 rows returned in 0.03 seconds Download

copyright@www.adrian.runceanu.ro

Operatorii pe mulţimi

2. Să se obţină codurile departamentelor în care nu lucrează nimeni (nu este introdus nici un salariat în tabelul EMP).
   Se cer două soluţii.

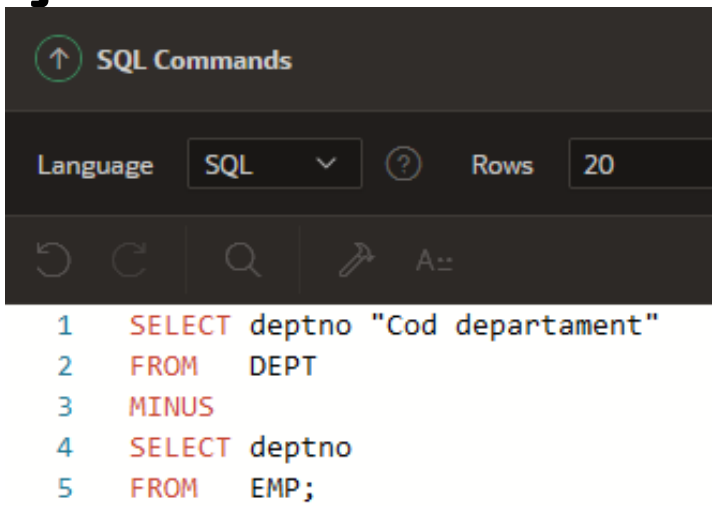
Obs: Operatorii pe mulţimi pot fi utilizaţi în subcereri.
Coloanele care apar în clauza WHERE a interogării trebuie să corespundă, ca număr şi tip de date, celor din clauza SELECT a subcererii.

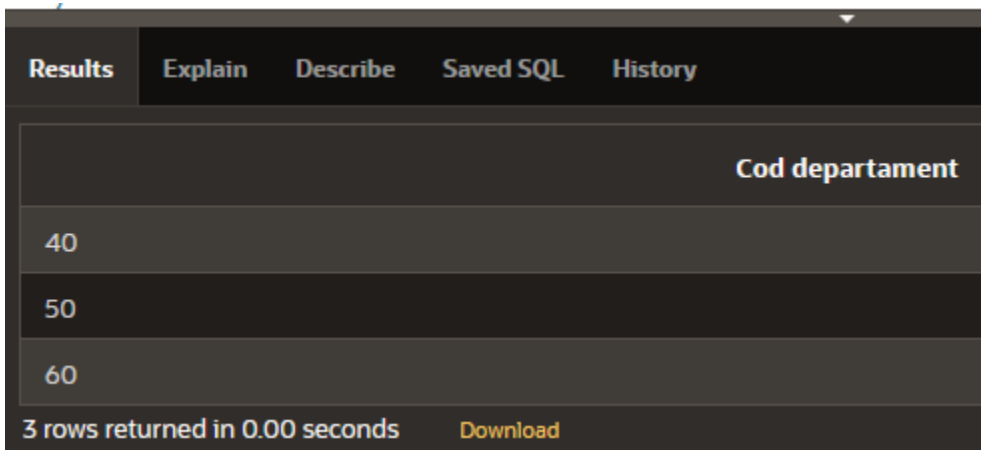
copyright@www.adrian.runceanu.ro

Operatorii pe mulţimi

Soluţia 1

SELECT deptno "Cod departament"
FROM DEPT
MINUS
SELECT deptno
FROM EMP;

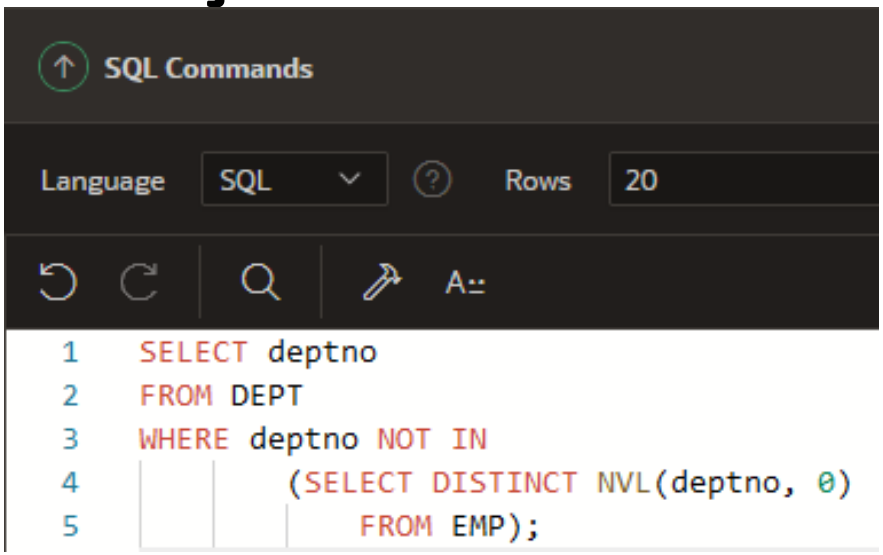




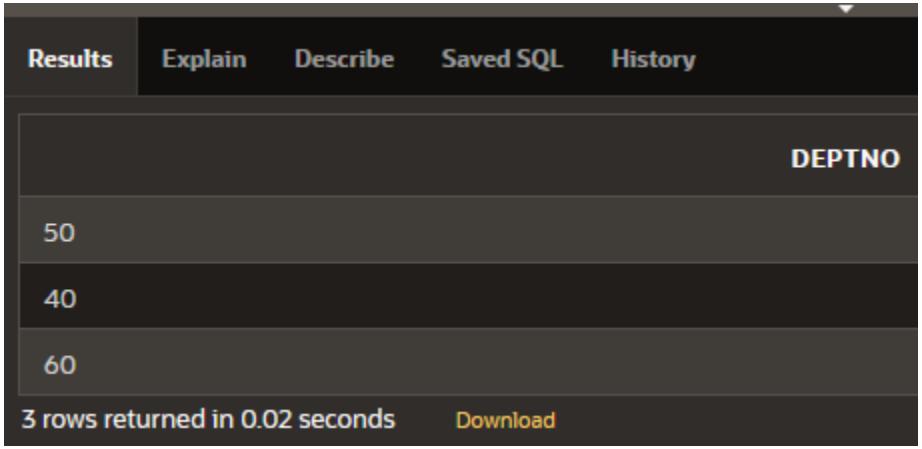
copyright@www.adrian.runceanu.ro

Operatorii pe mulţimi

Soluţia 2



SELECT deptno
FROM DEPT
WHERE deptno **NOT IN**
(**SELECT DISTINCT NVL**(deptno, 0)
FROM EMP);



copyright@www.adrian.runceanu.ro

Întrebări?