

CAPITOLUL 7

EXTRAGEREA DATELOR DIN MAI MULT DE O TABELA

Join

Join-ul este folosit cind o cerere SQL necesita date din mai multe tabele din baza de date.

Liniile dintr-o tabela pot fi unite cu liniile din alta tabela in functie de valorile comune existente in coloanele corespunzatoare.

Sint doua tipuri de conditie de join :

1. Equi-join
2. Non-equi-join

Equi-join

Pentru a determina angajatii din departamente, vom compara valorile din coloana DEPTNO a angajatului cu aceleasi valori din DEPTNO din tabela DEPT. Relatia dintre tabela EMP si DEPT este un equi-join, in care valorile din coloana DEPTNO din ambele tabele sint egale. (Operatorul de comparatie folosit este =.)

O conditie de join este specificata in clauza WHERE:

```
SELECT column(s)
FROM tables
WHERE join condition is ...
```

Pentru a face join pe cele doua tabele EMP si DEPT, introducem:

```
SELECT ENAME, JOB, DNAME
FROM EMP, DEPT
WHERE EMP>DEPTNO + DEPT>DEPTNO;
```

ENAME	JOB	DNAME
-----	-----	-----
CLARK	MANAGER	ACCOUNTING
MILLER	CLERK	ACCOUNTING
KING	PRESIDENT	ACCOUNTING
SMITH	CLERK	RESEARCH
SCOTT	ANALYST	RESEARCH
JONES	MANAGER	RESEARCH
ADAMS	CLERK	RESEARCH
FORD	ANALYST	RESEARCH
ALLEN	SALESMAN	SALES
BLAKE	MANAGER	SALES

TURNER	SALESMAN	SALES
JAMES	CLERK	SALES
MARTIN	SALESMAN	SALES
WARD	SALESMAN	SALES

Vom observa ca acum fiecare angajat are listat numele departamentului lui. Liniile din EMP sint combinate cu liniile din DEPT si sint intoarse doar liniile pentru care valorile EMP.DEPTNO si DEPT.DEPTNO sint eale.

Observati ca, conditia de join specifica numele coloanei precedat de numele tabeli. Aceasta este o necesitate cind numele coloanelor sint aceleasi in ambele tabele. Este necesar sa specificam exact ce coloane sint referite.

Aceasta necesitate este de asemenea aplicata coloanelor care pot fi ambigue in clauzele SELECT sau ORDER BY.

Pentru a recunoaste diferentele dintre coloana DEPTNO din EMP si coloana DEPTNO din DEPT, introducem:

```
SELECT DEPT.DEPTNO, ENAME, JOB, DNAME
FROM EMP, DEPT
WHERE EMP.DEPTNO = DEPT.DEPTNO
ORDER BY DEPT.DEPTNO;
```

DEPTNO	ENAME	JOB	DNAME
-----	-----	-----	-----
10	CLARK	MANAGER	ACCOUNTING
10	MILLER	CLERK	ACCOUNTING
10	KING	PRESIDENT	ACCOUNTING
20	SMITH	CLERK	RESEARCH
20	SCOTT	ANALYST	RESEARCH
20	JONES	MANAGER	RESEARCH
20	ADAMS	CLERK	RESEARCH
20	FORD	ANALYST	RESEARCH
30	ALLEN	SALESMAN	SALES
30	BLAKE	MANAGER	SALES
30	TURNER	SALESMAN	SALES
30	JAMES	CLERK	SALES
30	MARTIN	SALESMAN	SALES
30	WARD	SALESMAN	SALES

Observati ca fiecarui numar de departament din tabela DEPT i se face join pentru a se potrivi cu numerele de departament din tabela EMP. De exemplu, trei angajati lucreaza in departamentul 10 - Accounting - deci exista trei potri- viri. Prin urmare ACCOUNTING este afisat pentru fiecare angajat din acel departament.

Folosirea alias-urilor de tabela

Poate fi foarte plictisitor sa tiparesti numele tabelor repetat. Etichete temporare (sau alias-uri) pot fi folosite in clauza FROM. Aceste nume temporare sint valide doar in instructiunea SELECT curenta. Alias-urile de tabele trebuie de asemenea sa fie specificate in clauza SELECT. Aceasta creste efectiv viteza cererii, in care contextul contine informatii foarte exacte.

Alias-urile de tabela sint folosite in urmatorul context:

```
SELECT E.ENAME, D.DEPTNO, D.DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO
ORDER BY D.DEPTNO;
```

Alias-urile de tabele pot fi de lungime de maxim 30 de caractere, dar mai scurte sint mai bune. De asemenea incercati sa le faceti cit mai explicite.

Daca un alias de tabela este folosit pentru un nume particular de tabela in clauza FROM, atunci acel alias trebuie sa fie substituit pentru numele tabelii in contextul din SELECT.

Produs

```
-----
----
|  EMPNO   ENAME   JOB      MGR   HIREDATE   SAL   COMM
DEPTNO |
-----
----
|=====|
===|
-----
----
|  7788    SCOTT   ANALYST   7566   05-MAR-84   3000         20
|-----
----
|
|=====|
===|
-----
|
|
|
|
-----
-----|
|
|-----|
|          | DEPTNO   DNAME      LOC      |
|-----|
|-----|  10     ACCOUNTING NEW YORK  |
|-----|  20     RESEARCH   DALLAS   |
|-----|  30     SALES      CHICAO   |
```

In absenta unei conditii WHERE, fiecare linie din EMP este unita in ordine cu fiecare linie din DEPT.

Se vor afisa 53 de linii.

Produs

Cind o conditie de join este invalida sau este omisa, rezultatul este omis si toate combinatiile de linii vor fi listate.

Un produs tinde sa genereze un numar mare de linii si rezultatul sau este rar folosit. Trebuie intotdeauna inclusa o conditie de join intr-o clauza WHERE, in afara de cazul in care este necesara combinarea tuturor liniilor din toate tabelele.

Non-Equi-Join

Relatia dintre tabelele EMP si SALGRADE este un non-equi-join, in care nici o coloana din EMP nu corespunde direct cu o coloana din SALGRADE. Relatia este obtinuta folosind un operator, altul decit operatorul de egalitate (=). Pentru a evalua gradatia unui anajat, salariul lui trebuie sa fie intre salariul minim si salariul maxim. Operatorul BETWEEN este folosit pentru a construi conditia, introducem:

```
SELECT E.ENAME, E.SAL, S.GRADE
FROM EMP E, SALGRADE S
WHERE E.SAL BETWEEN S.LOSAL AND S.HISAL;
```

ENAME	SAL	GRADE
SMITH	800.00	1
ADAMS	1,100.00	1
JAMES	950.00	1
WARD	1,250.00	2
MARTIN	1,250.00	2
MILLER	1,300.00	2
ALLEN	1,600.00	3
TURNER	1,500.00	3
JONES	2,975.00	4
BLAKE	2,850.00	4
CLARK	2,450.00	4
SCOTT	3,000.00	4
FORD	3,000.00	4
KING	5,000.00	5

Alti operatori cum ar fi <= si >= pot fi folositi, oricum BETWEEN este cel mai simplu. Nu uitati sa specificati mai intii valoarea minima si ultima valoarea maxima cind folositi BETWEEN. Din nou sint folosite alias-uri de tabele, nu din cauza posibilelor ambiguitati, dar din motive de performanta.

Reguli pentru join-ul tabelor

Pentru a face join pe cele trei tabele este necesar sa construim doua conditii de join. Pentru a face join pe patru tabele sint necesare minim trei conditii de join.

O regula simpla este:

numarul minim de conditii de join = numarul de tabele - 1

Aceasta regula nu poate fi aplicata daca tabela are o cheie primara, care identifica in mod unic fiecare linie (cheile primare sint explicate mai tirziu in manual).

Sintaxa

```
SELECT [DISTINCT] {[tabela].* | expresie [alias], ...}
FROM tabela [alias], ...
WHERE [conditie de join] ...
AND [conditie de linie] ...
OR [alta conditie de linie]
GROUP BY {expresie | coloana}
HAVING {conditie de grup}
ORDER BY {expresie | coloana} [ASC | DESC]
```

Observatii

- Se pot specifica conditii de join impreuna cu alte conditii (non join);
- De asemenea trebuie sa fiti atenti la precedenta operatorilor cind folositi predicatul OR.

Exercitii - Join-uri simple

Aceste exercitii au intentia de a va capata experienta practica in extragerea datelor din mai mult decit o tabela si includ teme discutate in capitolele anterioare.

Tema

1. Afisati numele tuturor angajatilor si numele departamentului lor, in ordinea numelui departamentelor.

2.	ENAME	DNAME
3.	-----	-----
4.	CLARK	ACCOUNTING
5.	MILLER	ACCOUNTING
6.	KING	ACCOUNTING
7.	SMITH	RESEARCH
8.	SCOTT	RESEARCH
9.	JONES	RESEARCH
10.	ADAMS	RESEARCH
11.	FORD	RESEARCH
12.	ALLEN	SALES
13.	BLAKE	SALES
14.	TURNER	SALES
15.	JAMES	SALES
16.	MARTIN	SALES
17.	WARD	SALES

18.
19. Vor fi selectate 14 inregistrari.

20. Afisati numele tuturor angajatilor, numarul si numele departamentului.

21.
22.

23.	ENAME	DEPTNO	DNAME
24.	-----	-----	-----
25.	CLARK	10	ACCOUNTING
26.	MILLER	10	ACCOUNTING
27.	KING	10	ACCOUNTING
28.	SMITH	20	RESEARCH
29.	SCOTT	20	RESEARCH
30.	JONES	20	RESEARCH
31.	ADAMS	20	RESEARCH
32.	FORD	20	RESEARCH
33.	ALLEN	30	SALES
34.	BLAKE	30	SALES
35.	TURNER	30	SALES
36.	JAMES	30	SALES
37.	MARTIN	30	SALES
38.	WARD	30	SALES
39.			

40. Vor fi selectate 14 inregistrari.

41. Afisati numele, localitatea si departamentul angajatilor al caror salariu lunar este mai mare ca 1500.

42.

43.	ENAME	LOCATION	DNAME
43.	-----	-----	-----
44.	CLARK	NEW YORK	ACCOUNTING
45.	KING	NEW YORK	ACCOUNTING
46.	JONES	DALLAS	RESEARCH
47.	FORD	DALLAS	RESEARCH
48.	SCOTT	DALLAS	RESEARCH
49.	ALLEN	CHICAGO	SALES
50.	BLAKE	CHICAGO	SALES
51.			

52. Vor fi selectate 7 inregistrari.

53. Afisati lista salariilor, gradatiilor angajatilor.

54.

55.	ENAME	JOB	SAL	GRADE
55.	-----	-----	-----	-----
56.	SMITH	CLERK	800.00	1
57.	ADAMS	CLERK	1,100.00	1
58.	JAMES	CLERK	950.00	1
59.	WARD	SALESMAN	1,250.00	2
60.	MARTIN	SALESMAN	1,250.00	2
61.	MILLER	CLERK	1,300.00	2
62.	ALLEN	SALESMAN	1,600.00	3
63.	TURNER	SALESMAN	1,500.00	3
64.	JONES	MANAGER	2,975.00	4
65.	BLAKE	MANAGER	2,850.00	4

66.	CLARK	MANAGER	2,450.00	4
67.	SCOTT	ANALYST	3,000.00	4
68.	FORD	ANALYST	3,000.00	4
69.	KING	PRESIDENT	5,000.00	5

70. Listati doar angajatii cu gradatia 3.

71.	ENAME	JOB	SAL	GRADE
72.	-----	-----	-----	-----
73.	ALLEN	SALESMAN	1,600.00	3
74.	TURNER	SALESMAN	1,500.00	3

75. Listati toti angajatii din Dallas.

76.	ENAME	SAL	LOCATION
77.	-----	-----	-----
78.	SMITH	800.00	DALLAS
79.	SCOTT	3,000.00	DALLAS
80.	JONES	2,975.00	DALLAS
81.	ADAMS	1,100.00	DALLAS
82.	FORD	3,000.00	DALLAS

Alte exercitii daca aveti timp:

83. Afisati numele angajatilor, functia, salariul, gradatia si numele departamentului pentru toti angajatii din companie in afara de functionari. Sortati dupa salariu, afisind mai intii salariul cel mai mare.

84.					
85.					
86.	ENAME	JOB	SAL	GRADE	DNAME
87.	-----	-----	-----	-----	-----
88.	KING	PRESIDENT	5,000.00	5	ACCOUNTING
89.	FORD	ANALYST	3,000.00	4	RESEARCH
90.	SCOTT	ANALYST	3,000.00	4	RESEARCH
91.	JONES	MANAGER	2,975.00	4	RESEARCH
92.	BLAKE	MANAGER	2,850.00	4	SALES
93.	CLARK	MANAGER	2,450.00	4	ACCOUNTING
94.	ALLEN	SALESMAN	1,600.00	3	SALES
95.	TURNER	SALESMAN	1,500.00	3	SALES
96.	MARTIN	SALESMAN	1,250.00	2	SALES
97.	WARD	SALESMAN	1,250.00	2	SALES

98.

99. Vor fi selectate 10 inregistrari.

100. Afisati urmatoarele detalii pentru angajatii care cistiga 36000\$ pe an sau care sint functionari.

101.					
102.					
103.	ENAME	JOB	ANNUAL_SAL	DEPTNO	DNAME
	GRADE				

```

104. -----
-----
105. FORD ANALYST 36000 20 RESEARCH
4
106. SCOTT ANALYST 36000 20 RESEARCH
4
107. MILLER CLERK 15600 10 ACCOUNTING
2
108. JAMES CLERK 11400 30 SALES
1
109. ADAMS CLERK 13200 20 RESEARCH
1
110. SMITH CLERK 9600 20 RESEARCH
1
111.
112. Vor fi selectate 6 inregistrari.

```

Solutii

```

1. SELECT ENAME, DNAME
2. FROM EMP, DEPT
3. WHERE EMP.DEPTNO = DEPT.DEPTNO;
4.
5. SELECT ENAME, E.DEPTNO, DNAME
6. FROM EMP E, DEPT D
7. WHERE E.DEPTNO = D.DEPTNO;
8.
9. SELECT ENAME, LOC LOCATION, DNAME
10. FROM EMP, DEPT
11. WHERE EMP.DEPTNO = DEPT.DEPTNO
12. AND SAL > 1500;
13.
14. SELECT ENAME, JOB, SAL, GRADE
15. FROM EMP, SALGRADE
16. WHERE SAL BETWEEN LOSAL AND HISAL;
17.
18. SELECT ENAME, JOB, SAL, GRADE
19. FROM EMP, SALGRADE
20. WHERE SAL BETWEEN LOSAL AND HISAL
21. AND GRADE = 3;
22.
23. SELECT ENAME, SAL, LOC LOCATION
24. FROM EMP, DEPT
25. WHERE EMP.DEPTNO = DEPT.DEPTNO
26. AND LOC = 'DALLAS';
27.
28. SELECT ENAME, JOB, SAL, GRADE, DNAME
29. FROM EMP, SALGRADE, DEPT
30. WHERE EMP.DEPTNO = DEPT.DEPTNO
31. AND SAL BETWEEN LOSAL AND HISAL
32. AND JOB != 'CLERK'
33. ORDER BY SAL DESC;
34.
35. SELECT ENAME, JOB, SAL * 12 ANNUAL_SAL, D.DEPTNO, DNAME,
GRADE
36. FROM EMP E, SALGRADE, DEPT D
37. WHERE E.DEPTNO = D.DEPTNO
38. AND SAL BETWEEN LOSAL AND HISAL
39. AND (SAL * 12 + NVL(COMM, 0) = 3600 OR E.JOB = 'CLERK')

```

40. ORDER BY E.JOB;
41.