

Tipps & Tricks: Neuerungen Joins

Bereich:	SQL	Erstellung:	07/2004 HA
Versionsinfo:	10.1, 10.2, 11.1, 11.2	Letzte Überarbeitung:	06/2009 MA

Neuerungen zu Joins

Ab Version 9i sind alle dem SQL:1999-Standard entsprechenden Joins implementiert.

CROSS JOIN

Entspricht dem kartesischen Produkt von zwei oder mehr selektierten Tabellen ohne Join-Bedingung.

Beispiel:

Der bisher verwendeten SELECT-Anweisung

```
SQL> SELECT ename, loc, grade
      FROM emp, dept, salgrade;
```

entspricht ab Version 9i die Syntax

```
SQL> SELECT ename, loc, grade
      FROM emp CROSS JOIN dept CROSS JOIN salgrade;
```

ENAME	LOC	GRADE
SMITH	NEW YORK	1
ALLEN	NEW YORK	1
WARD	NEW YORK	1
JONES	NEW YORK	1
MARTIN	NEW YORK	1
BLAKE	NEW YORK	1
CLARK	NEW YORK	1
SCOTT	NEW YORK	1
KING	NEW YORK	1
TURNER	NEW YORK	1
ADAMS	NEW YORK	1
JAMES	NEW YORK	1
FORD	NEW YORK	1
MILLER	NEW YORK	1
SMITH	DALLAS	1
ALLEN	DALLAS	1
WARD	DALLAS	1
JONES	DALLAS	1
...		

280 Zeilen ausgewählt.

NATURAL JOIN

Bei Verknüpfung zweier Tabellen über einen Natural Join werden alle Spalten, die in beiden Tabellen den gleichen Namen und den gleichen Datentyp besitzen automatisch in die Join-Bedingung eingebunden. Der Einsatz eines Präfix vor den zur Verknüpfung verwendeten Spaltennamen ist nicht zulässig und erzeugt die Fehlermeldung ORA-25155.

Beispiel:

```
SQL> SELECT deptno, ename, dname
      FROM emp NATURAL JOIN dept ;
```

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

14 Zeilen ausgewählt.

```
SQL> SELECT emp.deptno, ename, dname
      2  FROM emp NATURAL JOIN dept ;
SELECT emp.deptno, ename, dname
      *
FEHLER in Zeile 1:
ORA-25155: Spalte, die in NATURAL-Join verwendet wird,
          darf keinen Bezeichner haben
```

EQUIJOIN mit USING-Klausel

Während beim Natural Join alle übereinstimmenden Spalten zweier Tabellen zur Verknüpfung verwendet werden, kann beim Equijoin über die USING-Klausel explizit eine bestimmte Spalte für die Join-Bedingung angegeben werden.

Es ist allerdings darauf zu achten, dass diese Spalte im gesamten Statement keinen Tabellennamen oder Alias als Präfix haben darf.

Beispiel:

```
SQL> SELECT deptno, ename, dname
       FROM emp JOIN dept USING (deptno)
       WHERE deptno IN (10, 30);
```

DEPTNO	ENAME	DNAME
30	ALLEN	SALES
30	WARD	SALES
30	MARTIN	SALES
30	BLAKE	SALES
10	CLARK	ACCOUNTING
10	KING	ACCOUNTING
30	TURNER	SALES
30	JAMES	SALES
10	MILLER	ACCOUNTING

9 Zeilen ausgewählt.

Hinweis: LOB`s und Collections sind als Verknüpfungs-Spalten nicht erlaubt.

ON-Klausel

Statt wie bisher in der WHERE-Klausel, wird die Join-Bedingung nun in der ON-Klausel definiert. Somit kann jede Join-Bedingung explizit angegeben und von den übrigen Einschränkungen in der WHERE-Klausel getrennt werden, wodurch die Lesbarkeit des gesamten Statements erhöht wird.

Beispiel:

```
SQL> SELECT e.ename, d.dname
       FROM emp e JOIN dept d ON (e.deptno = d.deptno)
       WHERE e.deptno = 20;
```

ENAME	DNAME
SMITH	RESEARCH
JONES	RESEARCH
SCOTT	RESEARCH
ADAMS	RESEARCH
FORD	RESEARCH

OUTER JOIN

In der Version 9i entfällt die Verwendung des (+)-Zeichens in der Join-Bedingung und wird durch die Schlüsselwörter LEFT bzw. RIGHT OUTER JOIN ersetzt. Zu beachten ist, dass ein (+) auf der rechten Seite durch einen LEFT OUTER JOIN und ein (+) auf der linken Seite durch einen RIGHT OUTER JOIN ersetzt wird. Neu hinzugekommen ist der FULL OUTER JOIN, bei dem nun alle angesprochenen Datensätze beider Tabellen angezeigt werden.

Beispiele:

```
SQL> SELECT e.ename, e.deptno, d.dname
```

```
FROM emp e LEFT OUTER JOIN dept d
ON (e.deptno = d.deptno) ;
```

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

15 Zeilen ausgewählt.

```
SQL> SELECT e.ename, d.deptno, d.dname
FROM emp e RIGHT OUTER JOIN dept d
ON (e.deptno = d.deptno) ;
```

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

15 Zeilen ausgewählt.

```
SQL> SELECT e.ename, d.deptno, d.dname
FROM emp e FULL OUTER JOIN dept d
ON (e.deptno = d.deptno) ;
```

```
ENAME          DEPTNO DNAME
-----
MILLER          10 ACCOUNTING
KING            10 ACCOUNTING
CLARK           10 ACCOUNTING
FORD            20 RESEARCH
ADAMS           20 RESEARCH
SCOTT           20 RESEARCH
JONES           20 RESEARCH
SMITH           20 RESEARCH
JAMES           30 SALES
TURNER          30 SALES
BLAKE           30 SALES
MARTIN          30 SALES
WARD            30 SALES
ALLEN           30 SALES
HUBER           40 OPERATIONS
16 Zeilen ausgewählt.
```