

Tipps & Tricks: Mengenoperatoren

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

 [Als PDF Downloaden!](#)

Mengenoperatoren

Mengen- oder SET-Operatoren fassen das Ergebnis von zwei oder mehreren Teilabfragen zu einem Ergebnis zusammen.

Die folgenden SET-Operatoren werden an dieser Stelle besprochen:

- UNION
- UNION ALL
- INTERSECT
- MINUS

Alle Mengenoperatoren haben die gleiche Priorität. Bei mehr als einem Operator in einer SQL-Anweisung werden diese standardmäßig von links (oben) nach rechts (unten) ausgewertet. Durch Klammersetzung kann jedoch eine andere Reihenfolge festgelegt werden.

UNION

UNION ist die Kombination zweier Tabellen. Durch den UNION-Operator werden alle Zeilen aus Mehrfachabfragen zurückgegeben und doppelte Ausgabezeilen ausgeblendet. Mathematisch gesehen entspricht UNION der Vereinigungsmenge.

Beispiel:

Es werden die EMP- und DEPT-Tabellen über einen OUTER JOIN miteinander verknüpft, wobei zusätzlich sowohl Mitarbeiter ausgegeben werden sollen, die in keiner Abteilung sitzen, als auch Abteilungen, denen noch kein Mitarbeiter zugeordnet worden ist.

Hinweis: Für diese Beispiele werden nicht die Standard Tabellen verwendet, es wurden zusätzliche Eintragungen vorgenommen.

```
SQL> SELECT e.empno, e.ename, d.dname, d.loc, d.deptno
      FROM emp e, dept d
      WHERE e.deptno=d.deptno(+)
      UNION
      SELECT e.empno, e.ename, d.dname, d.loc, d.deptno
      FROM emp e, dept d
      WHERE e.deptno(+) =d.deptno;
```

EMPNO	ENAME	DNAME	LOC	DEPTNO
7369	SMITH	RESEARCH	DALLAS	20
7499	ALLEN	SALES	CHICAGO	30

7521 WARD	SALES	CHICAGO	30
7566 JONES	RESEARCH	DALLAS	20
7654 MARTIN	SALES	CHICAGO	30
7698 BLAKE	SALES	CHICAGO	30
7782 CLARK	ACCOUNTING	NEW YORK	10
7788 SCOTT	RESEARCH	DALLAS	20
7839 KING	ACCOUNTING	NEW YORK	10
7844 TURNER	SALES	CHICAGO	30
7876 ADAMS	RESEARCH	DALLAS	20
7900 JAMES	SALES	CHICAGO	30
7902 FORD	RESEARCH	DALLAS	20
7934 MILLER	ACCOUNTING	NEW YORK	10
8000 MEYER			
8001 HUBER			
	HEAD_QUARTER	LAKE TAHOE	50
	OPERATIONS	BOSTON	40

Anmerkung:

Ab Version 9i kann dieselbe Ausgabemenge auch mittels FULL OUTER JOIN erzielt werden:

```
SQL> SELECT e.empno, e.ename, d.dname, d.loc, d.deptno
FROM emp e FULL OUTER JOIN dept d
ON e.deptno=d.deptno
ORDER BY e.empno;
```

Regeln für die Verwendung von UNION:

- Die Anzahl der Spalten und ihre Datentypen müssen zwischen den SELECT-Anweisungen übereinstimmen.
- Die Spaltennamen können verschieden sein.
- Standardmäßig wird die Ausgabe in aufsteigender Reihenfolge sortiert.
- Zeilen werden nur dann ausgeblendet, wenn sie in allen ausgewählten Spalten übereinstimmen.

UNION ALL

UNION ALL ist die Erweiterung des UNION-Operators (es gelten die Regeln 1-3). Er wird verwendet, wenn alle (auch doppelte) Zeilen aus Mehrfachabfragen zurückgegeben werden sollen.

Beispiel:

```
SQL> SELECT e.empno, e.ename, d.dname, d.loc, d.deptno
FROM emp e, dept d
WHERE e.deptno=d.deptno(+)
UNION ALL
SELECT e.empno, e.ename, d.dname, d.loc, d.deptno
FROM emp e, dept d
WHERE e.deptno(+) =d.deptno;
```

EMPNO	ENAME	DEPTNO	DNAME	LOC
-------	-------	--------	-------	-----

```

-----
7369 SMITH          20 RESEARCH      DALLAS
7499 ALLEN          30 SALES         CHICAGO
7521 WARD           30 SALES         CHICAGO
7566 JONES          20 RESEARCH      DALLAS
7654 MARTIN         30 SALES         CHICAGO
7698 BLAKE          30 SALES         CHICAGO
7782 CLARK          10 ACCOUNTING    NEW YORK
7788 SCOTT          20 RESEARCH      DALLAS
7839 KING           10 ACCOUNTING    NEW YORK
7844 TURNER         30 SALES         CHICAGO
7876 ADAMS          20 RESEARCH      DALLAS
7900 JAMES          30 SALES         CHICAGO
7902 FORD           20 RESEARCH      DALLAS
7934 MILLER         10 ACCOUNTING    NEW YORK
8000 MEYER
8001 HUBER
7782 CLARK          10 ACCOUNTING    NEW YORK
7839 KING           10 ACCOUNTING    NEW YORK
7934 MILLER         10 ACCOUNTING    NEW YORK
7369 SMITH          20 RESEARCH      DALLAS
7876 ADAMS          20 RESEARCH      DALLAS
7902 FORD           20 RESEARCH      DALLAS
7788 SCOTT          20 RESEARCH      DALLAS
7566 JONES          20 RESEARCH      DALLAS
7499 ALLEN          30 SALES         CHICAGO
7698 BLAKE          30 SALES         CHICAGO
7654 MARTIN         30 SALES         CHICAGO
7900 JAMES          30 SALES         CHICAGO
7844 TURNER         30 SALES         CHICAGO
7521 WARD           30 SALES         CHICAGO
                    40 OPERATIONS    BOSTON
                    50 HEAD_QUARTER  LAKE TAHOE
  
```

INTERSECT

Durch die Verwendung des Operators INTERSECT werden ausschließlich Zeilen ausgegeben, die in beiden Abfragen zurückgegeben werden. Mathematisch gesehen entspricht INTERSECT der Schnittmenge.

Beispiel:

```

SQL> SELECT e.empno, e.ename, d.dname, d.loc, d.deptno
      FROM emp e, dept d
      WHERE e.deptno=d.deptno(+)
      INTERSECT
      SELECT e.empno, e.ename, d.dname, d.loc, d.deptno
      FROM emp e, dept d
      WHERE e.deptno(+) = d.deptno;
  
```

```

EMPNO ENAME      DNAME      LOC      DEPTNO
-----
  
```

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

MINUS

Durch den Operator MINUS werden lediglich die Zeilen ausgegeben, die von der ersten Abfrage, nicht jedoch von der zweiten Abfrage zurückgegeben werden. Je nach Reihenfolge der Abfragen, ergibt sich eine andere Ausgabemenge.

Beispiel:

```
SQL> SELECT e.empno, e.ename, d.dname, d.loc, d.deptno
      FROM emp e, dept d
      WHERE e.deptno=d.deptno(+)
      MINUS
      SELECT e.empno, e.ename, d.dname, d.loc, d.deptno
      FROM emp e, dept d
      WHERE e.deptno(+) =d.deptno;
```

EMPNO	ENAME	DNAME	LOC	DEPTNO
8000	MEYER			
8001	HUBER			

Hinweis:

Ab 9i erreicht man den gleichen Effekt mit folgendem SELECT-Befehl:

```
SQL> SELECT e.empno, e.ename, d.dname, d.loc, d.deptno
      FROM emp e left join dept d on (e.deptno=d.deptno)
      MINUS
      SELECT e.empno, e.ename, d.dname, d.loc, d.deptno
      FROM emp e right join dept d on (e.deptno=d.deptno);
```

Wird die Reihenfolge der beiden SELECT-Abfragen vertauscht (Abfrage 2 minus Abfrage 1), ändert sich auch die Ausgabemenge:

EMPNO	ENAME	DNAME	LOC	DEPTNO
8000	MEYER			
8001	HUBER			

HEAD_QUARTER	LAKE TAHOE	50
OPERATIONS	BOSTON	40

Kombination verschiedener Mengenoperatoren

Um eine entsprechende Ausgabe zu erzielen, ist auch eine Kombination mehrerer Mengenoperatoren möglich.

Beispiel:

Es sollen nur die Datensätze ausgegeben werden, die keine Entsprechung in der jeweils anderen Tabelle besitzen.

```
SQL> (SELECT e.empno, e.ename, d.dname, d.loc, d.deptno
      FROM emp e, dept d
      WHERE e.deptno=d.deptno(+))
MINUS
      SELECT e.empno, e.ename, d.dname, d.loc, d.deptno
      FROM emp e, dept d
      WHERE e.deptno(+)=d.deptno)
UNION
      (SELECT e.empno, e.ename, d.dname, d.loc, d.deptno
      FROM emp e, dept d
      WHERE e.deptno(+)=d.deptno)
MINUS
      SELECT e.empno, e.ename, d.dname, d.loc, d.deptno
      FROM emp e, dept d
      WHERE e.deptno=d.deptno(+));
```

EMPNO	ENAME	DNAME	LOC	DEPTNO
8000	MEYER			
8001	HUBER			
		HEAD_QUARTER	LAKE TAHOE	50
		OPERATIONS	BOSTON	40

Die neue Syntax ab 9i lautet für diesen Befehl wie folgt:

```
SQL> (SELECT e.empno, e.ename, d.dname, d.loc, d.deptno
      FROM emp e left outer join dept d on (e.deptno=d.deptno)
      MINUS
      SELECT e.empno, e.ename, d.dname, d.loc, d.deptno
      FROM emp e right outer join dept d on (e.deptno=d.deptno)
      UNION
      (SELECT e.empno, e.ename, d.dname, d.loc, d.deptno
      FROM emp e right outer join dept d on (e.deptno=d.deptno)
      MINUS
      SELECT e.empno, e.ename, d.dname, d.loc, d.deptno
      FROM emp e left outer join dept d on (e.deptno=d.deptno)));
```