

Tipps & Tricks: Constraints

Bereich:	DBA, SQL	Erstellung:	02/2005 HA
Versionsinfo:	10.1, 10.2, 11.1, 11.2	Letzte Überarbeitung:	06/2009 MA

Constraints

Constraints werden verwendet, um ungültige Dateneinträge in Tabellen zu verhindern. Mit dem Anlegen von Constraints werden Regeln auf Tabellenebene erzwungen, sobald eine DML-Anweisung ausgeführt wird. Das Löschen einer Tabelle oder Tabellenzeile wird verhindert, falls Abhängigkeiten zu anderen Tabellen bestehen. Oberstes Ziel ist demnach die Gewährleistung der Datenintegrität.

Folgende Constraints sind in Oracle zulässig (in Klammern ist der Constraint-Typ angegeben, unter dem das Constraint in der View USER_CONSTRAINTS angezeigt wird):

- NOT NULL (C): erlaubt keine NULL-Werte
- UNIQUE (U): erlaubt nur eindeutige oder NULL-Werte
- PRIMARY KEY (P): Kombination aus NOT NULL und UNIQUE
- FOREIGN KEY (R): legt eine Fremdschlüsselbeziehung zu einer referenzierten Tabelle fest
- CHECK (C): gibt eine (oder mehrere) Bedingung(en) an, die erfüllt sein muss (müssen)

Constraints können entweder beim Anlegen der Tabelle (CREATE TABLE) angegeben werden oder nachträglich über ALTER TABLE.

Beispiele:

1.) Erzeugt ein NOT NULL-Constraint (ohne Namensangabe):

```
SQL> CREATE TABLE test (nr number NOT NULL, name varchar2(10));
```

2.) Erzeugt einen Primärschlüssel (Angabe auf Spaltenebene):

```
SQL> CREATE TABLE dept (deptno number(3) CONSTRAINT pk_dept PRIMARY KEY,
                        dname varchar2(10), loc varchar2(10));
```

3.) Erzeugt einen Primär- und einen Fremdschlüssel (beide auf Tabellenebene):

```
SQL> CREATE TABLE emp (empno number(3), ename varchar2(10), deptno number(3),
                        CONSTRAINT pk_emp PRIMARY KEY(empno),
                        CONSTRAINT fk_deptno FOREIGN KEY(deptno) REFERENCES
                        dept(deptno));
```

4.) Erzeugt nachträglich ein NOT NULL-Constraint (mit Namensangabe):

```
SQL> ALTER TABLE test MODIFY (name CONSTRAINT nn_name NOT NULL);
```

- 5.) Erzeugt nachträglich ein CHECK-Constraint mit den zwei Bedingungen:
alle Namen müssen in Großbuchstaben angegeben werden und mind. 5 Zeichen lang sein.

```
SQL> ALTER TABLE test ADD (CONSTRAINT ck_name  
                           CHECK (upper(name)=name) AND length(name)>=5));
```

- 6.) Erzeugt einen Fremdschlüssel, der Löschvorgänge der übergeordneten Tabelle weiterreicht:

```
SQL> ALTER TABLE emp_copy ADD (CONSTRAINT fk_copy FOREIGN KEY(deptno)  
                               REFERENCES dept(deptno) ON DELETE CASCADE);  
[alternativ: ON DELETE SET NULL]
```

- 7.) Löscht ein bestehendes Constraint:

```
SQL> ALTER TABLE test DROP CONSTRAINT ck_name;
```

Informationen zu Constraints im Data Dictionary:

Eine Übersicht über die eigenen Constraints erhält man über die View USER_CONSTRAINTS. Auf welche Spalten die Constraints gelegt sind, sehen Sie in USER_CONS_COLUMNS.

Folgender SELECT joint die beiden Views und gibt die wichtigsten Informationen zurück:

```
SQL> SELECT a.constraint_name, a.constraint_type, a.table_name, b.column_name,  
           a.search_condition, a.r_constraint_name  
FROM user_constraints a, user_cons_columns b  
WHERE a.constraint_name = b.constraint_name;
```

Tipps zu Constraints:

- Benennen Sie ihre Constraints beim Anlegen selbst, ansonsten generiert Oracle einen in der Art SYS_C00xxxx. Im Falle von Fehlermeldungen ist dies ganz hilfreich.
- Pro Tabelle kann es nur einen Primärschlüssel geben. Dieser kann jedoch über mehrere Spalten gehen.
- Alle Constraints werden nachträglich über die Option ADD in der ALTER TABLE-Anweisung angelegt. Einzige Ausnahme: ein NOT NULL-Constraint wird über MODIFY erzeugt.
- Vor der Erzeugung eines Fremdschlüssels muss ein PRIMARY KEY- oder UNIQUE-Constraint erzeugt worden sein.
- Wird ein Fremdschlüssel auf eine Spaltenkombination gelegt, muss der übergeordnete Schlüssel auf die gleiche Anzahl an Spalten gelegt worden sein.
- Lässt sich ein Constraint nachträglich über ALTER TABLE nicht mehr anlegen bzw. einschalten, können Sie über den Zusatz EXCEPTIONS INTO EXCEPTIONS herausfinden, welche Zeilen dem Constraint widersprechen. Dazu muss vorher eine EXCEPTIONS-Tabelle erzeugt worden sein (über das Skript utlexcpt.sql im rdbms\admin-Verzeichnis).
- Beachten Sie, dass beim Erzeugen eines Primärschlüssels bzw. UNIQUE-Constraints automatisch ein Index mit dem selben Namen angelegt wird.
- Weitere Tipps und Informationen zu Constraints hören Sie in unserem SQL Grundlagen-Kurs