

## Tipps & Tricks: Indizes

Bereich:	SQL	Erstellung:	2003 MP
Versionsinfo:	8.2 9.2 10.2 11.1	Letzte Überarbeitung:	05/2009 BK

## Indizes

Ein Index ist ein Datenbankobjekt, das durch Nutzung einer effektiven Speicherstruktur den Abruf von Zeilen beschleunigen kann.

Indizes können vom Benutzer explizit (CREATE INDEX) oder bei der Definition eines PRIMARY KEY- bzw. UNIQUE KEY-Constraints automatisch erstellt werden.

### Erstellen eines Index

Die allgemeine Syntax lautet:

```
CREATE [UNIQUE] INDEX <index_name>
  ON <tabellen_name> (<spalten_liste>)
  [TABLESPACE <index_tbs>];
```

wobei:

- UNIQUE einen eindeutigen Index auf die <spalten\_liste> legt und somit doppelte Spaltenwerte verhindert.
- TABLESPACE den Bereich auf der Platte angibt, in dem der Index gespeichert werden soll.

Beispiele:

```
CREATE UNIQUE INDEX emp_ename_idx
  ON scott.emp(ename)
  TABLESPACE indx;
```

Im zweiten Beispiel werden beim Anlegen der Tabelle test automatisch Indizes auf die Spalten col1 und col2 gelegt.

```
CREATE TABLE test (
  col1 number(3) primary key,
  col2 varchar2(15) unique);
```

### Richtlinien zur Indexerstellung

Unter folgenden Voraussetzungen erscheint eine Indizierung sinnvoll:

- Es handelt sich um eine sehr große Tabelle und die meisten Abfragen rufen zwischen 2-5% der Zeilen ab.
- In der Index-Spalte befinden sich viele NULL-Werte.
- Die Index-Spalte wird oft in der WHERE-Klausel oder einer JOIN-Bedingung verwendet.
- Die Index-Spalte enthält einen großen Wertebereich.
- Die Tabelle wird nicht häufig aktualisiert.

## Funktionsbasierter Index

Enthält die WHERE-Klausel statt der Spalte eine Funktion auf die Spalte, wird ein normaler Index nicht verwendet. Ab Version 8i gibt es die Möglichkeit einen sog. funktionsbasierten Index zu erstellen, der dann für die angegebene Funktion wirksam wird.

Der oben erstellte Index emp\_ename\_idx wird bei folgender Abfrage nicht verwendet:

```
SELECT ename, empno, sal
FROM emp
WHERE UPPER(ename) = 'KING';
```

Um dieses Problem zu umgehen, erstellt man einen entsprechenden funktionsbasierten Index:

```
CREATE INDEX emp_upper_idx
ON scott.emp(UPPER(ename));
```

Hinweis: Bis zur Version 9.2 müssen folgende Voraussetzungen gegeben sein damit ein funktionsbasierter Index erstellt werden kann.

Der Parameter QUERY\_REWRITE\_ENABLED muss auf TRUE gesetzt sein.

Der Benutzer braucht das Recht QUERY REWRITE.

## Löschen eines Index

Manuell erzeugte Indizes können durch folgende allgemeine Syntax entfernt werden:

```
DROP INDEX <index_name>;
```

Indizes, die beim Anlegen eines PRIMARY KEY- oder UNIQUE KEY-Constraints automatisch erstellt worden sind, können nur durch Ausschalten oder Löschen des jeweiligen Constraints entfernt werden.

## Data Dictionary-Views

Alle Informationen zu den Indizes eines Benutzers lassen sich über die DD-Views

- USER\_INDEXES
- USER\_IND\_COLUMNS

anzeigen.