

## Tipps & Tricks: Januar 2006 - Bonus

Bereich:	DBA, PL/SQL	Erstellung:	01/2006 MP
Versionsinfo:	9.2, 10.2, 11.1, 11.2	Letzte Überarbeitung:	06/2009 MP

 [Als PDF Downloaden!](#)

### Online Table Redefinition

Bis zur Version 9i waren nur folgende Operationen zum Reorganisieren von Tabellen möglich:

- Tabelle exportieren (evtl. mit COMPRESS=Y)
- Tabelle löschen
- Tabelle evtl. mit neuen Speicherparametern anlegen
- Tabelle importieren
- Ab Version 8i wurde folgende Operation ermöglicht:
  - ALTER TABLE <tab> STORAGE (...) MOVE TABLESPACE <tbs>;
  - Hierbei konnte die Tabelle im laufenden Betrieb mit neuen Speicherparametern angelegt werden.
  - Leider war sie während der Reorganisation nicht für Schreiboperationen verfügbar

Ab Version 9i kann nun die Tabelle online reorganisiert werden, auch Schreibzugriffe während der Reorganisation sind hier möglich

Mit dem neuen Package DBMS\_REDEFINITION können Tabellen mit einer minimalen Sperrzeit online reorganisiert werden.

#### Sie können mit diesem Package:

- die Unterstützung für parallele Abfragen hinzufügen oder entfernen,
- Partitionierungs-Unterstützung entfernen oder hinzufügen,
- die Tabelle neu aufbauen, um Fragmentierung zu vermindern,
- Spalten löschen oder hinzufügen,
- Tabellen Organisation ändern (IOT zu normal oder umgekehrt),
- die Speicherparameter einer Tabelle ändern,
- Löschen/Hinzufügen von Non-Primary Spalten
- die Tabelle in einen anderen Tablespace (im gleichen Schema) verschieben.

#### Einschränkungen beim Reorg:

- Der doppelte Speicherplatz für die Tabelle muss vorhanden sein
- Primärschlüsselspalten können nicht verändert werden
- Die zu ändernden Tabellen müssen einen Primärschlüssel besitzen (ab 9.2.0.5 geht auch ohne)
- Die neue Tabelle muss im gleichen Schema liegen
- Neu hinzugefügte Spalten können erst nach der Redefinition auf NOT NULL gesetzt werden
- Die Tabelle darf keine LONGs, BFILEs oder User Defined Types enthalten.
- Cluster Tabellen und Tabellen im Schema SYS und SYSTEM sind ausgeschlossen
- Tabellen mit Materialized View Logs oder Materialized Views dürfen nicht benutzt werden. Ab 11.1 ist diese Einschränkung entfallen.

#### Vorgehensweise:

Folgende Änderungen sollen an einer Tabelle emp durchgeführt werden:

Spalte ENAME soll umbenannt werden in NAME,  
 Spalte SAL soll umbenannt werden in VERDIENST und um den Faktor 1,5 erhöht,  
 die Spalten MGR, HIREDATE, COMM sollen gelöscht werden.

Tabelle EMP2

Name	Null?	Typ
-----	-----	-----
EMPNO	NOT NULL	NUMBER ( 4 )
ENAME		VARCHAR2 ( 10 )
JOB		VARCHAR2 ( 9 )
MGR		NUMBER ( 4 )
HIREDATE		DATE
SAL		NUMBER ( 7 , 2 )
COMM		NUMBER ( 7 , 2 )
DEPTNO		NUMBER ( 2 )

Überprüfen, ob EMP zu reorganisieren ist:

```
SQL> exec dbms_redefinition.can_redef_table('SCOTT', 'EMP');
PL/SQL-Prozedur wurde erfolgreich abgeschlossen.
```

Erstellen Sie eine vorläufige Tabelle mit den gewünschten Änderungen

```
SQL> CREATE TABLE scott.vor_emp (
empno, name, verdienst, deptno)
AS SELECT empno, ename, sal, deptno FROM scott.emp
WHERE 1=2;
```

Start der Reorganisation:

```
SQL> BEGIN
dbms_redefinition.start_redef_table(
'SCOTT', 'EMP', 'VOR_EMP',
'EMPNO EMPNO, ENAME NAME, SAL*1.5 VERDIENST');
END;
/
```

Erstellen Sie den Primärschlüssel auf empno:

```
SQL> ALTER TABLE scott.vor_emp ADD
CONSTRAINT pk_voremp2
PRIMARY KEY (empno);
```

Erstellen Sie einen Fremdschlüssel auf der Spalte deptno der auf die Spalte deptno der Tabelle DEPT verweist:

```
SQL> ALTER TABLE scott.vor_emp ADD(  
  CONSTRAINT fk_emp  
  FOREIGN KEY (deptno)  
  REFERENCES scott.dept (deptno));
```

Der FK muss ausgeschaltet werden. Dieser wird dann am Ende der Reorganisation automatisch aktiviert.

```
SQL> ALTER TABLE scott.vor_emp DISABLE CONSTRAINT fk_emp;
```

Wenn nötig, synchronisieren Sie die Tabellen zwischendurch:

```
SQL> BEGIN  
  dbms_redefinition.sync_interim_table(  
  'SCOTT', 'EMP', 'VOR_EMP');  
END;  
/
```

PL/SQL-Prozedur wurde erfolgreich abgeschlossen.

Beenden Sie die Redefinition:

```
SQL> BEGIN  
  dbms_redefinition.finish_redef_table(  
  'SCOTT', 'EMP', 'VOR_EMP');  
END;  
/
```

Löschen der alten Tabelle:

```
SQL> DROP TABLE vor_emp;
```

Der Redefinitions Vorgang kann abgebrochen werden mit:

```
exec dbms_redefinition.abort_redef_table('SCOTT', 'EMP', 'EMP');
```

Neuerungen ab Version 10.2:

Bevor die Reorg abgeschlossen ist, können noch diverse Objekte (wie Constraints, Trigger, Indizes, u.w.) mit übernommen werden.

Neuerungen ab Version 11.x:

Nun können auch abhängige Materialized Views beim Reorg-Prozess mit übernommen werden.

```
VARIABLE num_err NUMBER  
SET SERVEROUTPUT ON SIZE 200000
```

```
BEGIN
DBMS_REDEFINITION.COPY_TABLE_DEPENDENTS(
  uname=>'SCOTT',
  orig_table=>'EMP',
  int_table=>'VOR_EMP',
  copy_indexes=>dbms_redefinition.cons_orig_params,
  copy_triggers=>TRUE,
  copy_constraints=>TRUE,
  copy_privileges=>TRUE,
  copy_statistics=>TRUE,
  --copy_mvlog=>TRUE, /* Ab 11g*/
  ignore_errors=>FALSE,
  num_errors=>:num_err);
dbms_output.put_line('Anzahl Fehler bei Übernahme:' || :num_err);
END;
/
```

Weitere Informationen zum Thema Reorganisation erhalten Sie in unseren [Tuning](#) und [Monitoring](#) Kursen. Unser spezialisiertes Consulting-Team hilft Ihnen gerne beim Reorganisieren Ihrer Objekte.