

Tipps & Tricks: Verbessertes UNDO-Management

Bereich:	DBA	Erstellung:	01/2003 MM
Versionsinfo:	getestet mit 9.2.0.8, 10.2 und 11.1	Letzte Überarbeitung:	07/2009 MM

UNDO-Management

Ab Version 9i kann zwischen den Methoden "Manual UNDO-Management" und "Automatic UNDO-Management" gewählt werden. Der entsprechende Modus wird beim Starten der Instanz aktiviert und durch den Initialisierungs-Parameter UNDO_MANAGEMENT (= AUTO | MANUAL) festgelegt.

Es ist in einer Instanz nicht möglich, beide Methoden gleichzeitig zu benutzen. Zum Ändern der Methode muss die Datenbank heruntergefahren, der Parameter geändert und die Datenbank anschließend neu gestartet werden.

Unabhängig davon, welcher Modus aktiviert ist, gibt es weiterhin ein SYSTEM Rollback-Segment.

Manueller UNDO-Management Modus

Er wird immer dann gestartet, wenn der Parameter auf MANUAL gesetzt ist oder nicht angegeben wird. Im manuellen Modus werden die restlichen UNDO-Parameter ignoriert und das UNDO-Management wie gehabt über Rollback-Segmente realisiert. In diesem Fall können folgende Initialisierungs-Parameter gesetzt werden:

- MAX_ROLLBACK_SEGMENTS
- ROLLBACK_SEGMENTS
- TRANSACTIONS
- TRANSACTION_PER_ROLLBACK_SEGMENT

Automatischer UNDO-Management Modus

Der automatische Modus wird gewählt, wenn die Datenbank die UNDO-Segmente selbstständig und automatisch verwalten soll. In diesem Fall muss mindestens ein UNDO-Tablespace existieren. Über den Parameter UNDO_TABLESPACE kann dieser explizit angegeben und mit dem ALTER SYSTEM SET-Befehl geändert werden. Das folgende Beispiel gibt die Initialisierungs-Parameter für den automatischen UNDO-Management Modus wieder:

Beispiel:

```
SELECT name, value FROM v$parameter
WHERE name LIKE 'undo%';

NAME                                VALUE
-----
undo_management                     AUTO
undo_tablespace                     UNDOTBS
undo_suppress_errors                 FALSE   -- entfällt in Version 10g
undo_retention                       900
```

Erstellen eines UNDO-Tablespace

Das Erstellen eines UNDO-Tablespace kann entweder beim Erstellen einer neuen Datenbank erfolgen mit:

```
CREATE DATABASE db1 CONTROLFILE REUSE
...
UNDO TABLESPACE undo_01 DATAFILE '...\oradata\db1\undo_01a.dbf'
```

oder nachträglich bei einer bestehenden Datenbank durchgeführt werden mit:

```
CREATE UNDO TABLESPACE undo_01
DATAFILE '...\oradata\db1\undo_01a.dbf' SIZE 20M REUSE
AUTOEXTEND ON;
```

Hinweis:

Der Parameter UNDO_MANAGEMENT muss in der INIT.ORA bzw. im SPFILE auf AUTO gesetzt sein.

Wird bei der Datenbankerstellung im automatischen Modus die UNDO TABLESPACE-Klausel nicht angegeben, wird ein Default UNDO-Tablespace mit dem Namen SYS_UNDOTBS erstellt (Größe: 10MB; AUTOEXTENSIBLE)

Ändern eines UNDO-Tablespace

Mit dem Befehl ALTER TABLESPACE kann ein UNDO-Tablespace geändert werden. Dabei können Datenfiles hinzugefügt, online oder offline gesetzt, ein Datenfile-Name geändert oder das Backup eines Datenfiles begonnen oder beendet werden.

Beispiel:

```
ALTER TABLESPACE undo_01
ADD DATAFILE '...\oradata\db1\undo_01b.dbf' SIZE 10M
AUTOEXTEND ON NEXT 1M MAXSIZE UNLIMITED;
```

Switchen eines UNDO-Tablespace

Mit dem Befehl ALTER SYSTEM SET ist es möglich, dem Parameter UNDO_TABLESPACE einen anderen Wert zuzuweisen und somit den UNDO-Tablespace zu wechseln (switchen).

Beispiel:

```
ALTER SYSTEM SET UNDO_TABLESPACE=undo_01;
```

Durchführbar ist diese Änderung nur dann, wenn der neue Tablespace bereits existiert, vom Typ UNDO ist und von keiner anderen Instanz benutzt wird. Existieren im alten UNDO-Tablespace noch offene Transaktionen, wird dieser "PENDING OFFLINE" gesetzt und kann von einer anderen Instanz weder verwendet noch gelöscht werden.

Löschen eines UNDO-Tablespace

Mit dem Befehl DROP TABLESPACE kann ein UNDO-Tablespace gelöscht werden. Das ist jedoch nur möglich, wenn dieser nicht gerade benutzt wird, es also keine offenen Transaktionen bestehen.

Beispiel:

```
DROP TABLESPACE undo_01 INCLUDING CONTENTS AND DATAFILES ;
```

Hinweis:

Beim Löschen eines UNDO-Tablespace ist darauf zu achten, dass die UNDO-Daten, die dort noch vorgehalten werden, nicht mehr von lang laufenden Prozessen benötigt werden (SNAPSHOT TOO OLD).

User Quotas

Mit der Anweisung UNDO_POOL des Resource Managers können User Quotas auf UNDO-Tablespaces erstellt werden, also der UNDO-Speicher für Benutzergruppen eingeschränkt werden. Dabei kann jeder Gruppe ein eigener UNDO_POOL zugewiesen werden.

Ist die maximale Speichergrenze erreicht, wird die aktuelle Transaktion abgebrochen und erst nach Freigabe von neuem Speicher können neue Statements abgesetzt werden.

Wurde kein UNDO_POOL deklariert, bestehen keine UNDO-Speichereinschränkungen für die Benutzer.

Retention Period von UNDO-Informationen

UNDO-Informationen gehen üblicherweise verloren, sobald sie von einer neuen Transaktion überschrieben werden. Um zu vermeiden, dass bei lang laufenden Abfragen weiterhin benötigte Daten überschrieben werden, kann der Aufbewahrungszeitraum (Retention Period) von UNDO-Informationen explizit angegeben werden. Dies erfolgt durch den dynamischen Parameter UNDO_RETENTION, der im laufenden Betrieb geändert werden kann. Der Standard-Wert beträgt 900 Sekunden (15 Minuten).

Beispiel:

```
ALTER SYSTEM SET UNDO_RETENTION=1800 ;
```

Data Dictionary-Views zu UNDO-Management

Die View V\$UNDOSTAT enthält statistische Informationen zu Anzahl der Transaktionen, den Platzverbrauch und die maximale Anzahl gleichzeitiger Transaktionen. Diese Werte werden in einem Intervall von 10 Minuten gesammelt.

Beispiel:

```
SELECT to_char(begin_time, 'hh24:mi:ss') begin_time,
       to_char(end_time, 'hh24:mi:ss') end_time,
       undotsn, undoblks, txncount, maxconcurrency
FROM v$undostat
```

BEGIN_TI	END_TIME	UNDOTSN	UNDOBLKS	TXNCOUNT	MAXCONCURRENCY
10:44:00	10:50:47	10	1	1	1
10:34:00	10:44:00	10	2	2	1
10:24:00	10:34:00	10	2	2	1
10:14:00	10:24:00	10	2	2	1

In der View DBA_UNDO_EXTENTS werden Informationen über vorhandene UNDO-Extents gesammelt.

Beispiel:

```
SELECT segment_name, extent_id, block_id, commit_wtime, status
```

```
FROM dba_undo_extents
```

SEGMENT_NAME	EXTENT_ID	BLOCK_ID	COMMIT_WTIME	STATUS

_SYSSMU1\$	0	18	09/05/2001 02:20:17	ACTIVE
...				
_SYSSMU2\$	1	65	03/01/2002 10:09:52	EXPIRED
...				
_SYSSMU3\$	3	769	09/05/2001 01:54:31	UNEXPIRED

Die STATUS-Spalte kann dabei folgende Werte enthalten:

- ACTIVE - das Extent ist einer Transaktion zugewiesen und enthält aktive Rollback-Informationen.
- EXPIRED - das Extent enthält keine Daten, die noch vorgehalten werden müssen.
- UNEXPIRED - die Transaktion, die sich in dieses Extent eingetragen hat ist zwar bereits beendet, aber die Rollback-Informationen müssen noch für die UNDO_RETENTION-Zeit vorgehalten werden.