

## Oracle Flashback

# Gute alte Zeiten

**Andrea Held**

Fehler sind unvermeidlich – im günstigsten Fall ist nur ein Datensatz betroffen, im schlimmsten der ganze Datenbestand korrumpiert. Oracle 10g bietet einige Möglichkeiten, das Malheur zu reparieren.



Irgendwann passiert es: Ein unkonzentrierter Moment, eine kleine Unachtsamkeit, und noch während der Zeigefinger auf die Return-Taste niedersaust, schießt es einem schon durch den Kopf „Das war vielleicht falsch!“. Die Wiederherstellung fälschlich geänderter Daten und gelöschter Tabellen ist – gerade bei großen Datenbanken – oft unverhältnismäßig aufwendig. Im schlimmsten Fall muss man ein Backup einspielen und die Datenbank mit einem Point-in-Time-Recovery bis knapp vor dem Fehlerzeitpunkt wiederherstellen. Während diese Arbeiten meist Stunden dauern, braucht Oracle 10g mit den Optionen rund um Flashback dafür wenige Minuten.

Beschädigungen am Datenbestand durch Fehlbedienung oder durch Fehler in einer Anwendung lassen sich damit durch ein Zurückspulen der Datenänderungen einfach bereinigen. Unterschiedliche Level sorgen für die notwendige Granularität. So setzen *flashback database* den gesamten Bestand einer Datenbank und *flashback table* den Inhalt einer einzelnen Tabelle zurück. Der Befehl *flashback drop* stellt eine einzelne gelöschte Tabelle wieder her. Zudem kann der Ad-

ministrator mit *flashback query* die Änderungsgeschichte eines Datensatzes prüfen. Die für diese Flashback-Level eingesetzten Verfahren unterscheiden sich in Architektur und interner Implementierung zum Teil deutlich voneinander.

## Schnell auf dem Stand von gestern

Bei *flashback database* handelt es sich um den mächtigsten der Flashback-Befehle. In Minutenschnelle setzt er den gesamten Inhalt einer Datenbank zurück und ersetzt damit ein aufwendiges Point-in-Time-Recovery. Um diese Option nutzen zu können, benötigt die Datenbank den Archive-Log-Mode, in dem sie alle ihre Transaktions-Logs archiviert, und der Flashback-Log-Modus muss aktiviert sein. Er veranlasst das Schreiben von Speicherblöcken in die Flashback-Logs vor einer Änderung. Diese Form des Logging ist noch recht neu und erst seit Oracle 10.1 verfügbar. Ist sie eingeschaltet, führt der Befehl *flashback database* einen schnellen Datenbank-Reset herbei (siehe Listing 1).

Die einmalige Aktivierung des Logging erfolgt im Mount-Status der Datenbank. Hat man die Archivierung der Online-Redo-Log-Files und das Schreiben der Flashback-Logs eingeschaltet, kann man die Datenbank wieder öffnen. Hierfür ist ein Wartungsfenster von meist nur wenigen Minuten erforderlich.

Flashback-Logs speichert Oracle in der Flash Recovery Area. Dieser Bereich auf der Festplatte dient seit Version 10g für Backup-relevante Dateien. Den Pfad dorthin legt der Datenbankparameter *db\_recovery\_file\_dest* fest,

### XTRACT

- Oracle Flashback gestattet eine einfache und schnelle Korrektur von Datenfehlern.
- Die Möglichkeiten reichen vom Zurücksetzen des gesamten Datenbestandes einer Datenbank bis zur Korrektur von Datenfehlern in Tabellen.
- Flashback lässt sich per grafischer Oberfläche und durch SQL-Befehle auf der Kommandozeile bedienen.

`db_recovery_file_dest_size` schränkt seine maximale Größe ein. Für das Schreiben der Flashback-Logs muss ausreichend Plattenplatz bereitstehen. Zudem kostet das zusätzliche Protokoll bei hoher Transaktionsrate etwas Geschwindigkeit.

Bei aktiviertem Flashback-Log-Modus setzt der Befehl `flashback database` im Mount-Mode der Datenbank Änderungen zurück (Listing 2). Anschließend öffnet man sie mit einem Reset der Redo-Logs. Sie befindet sich danach mit allen Daten auf dem Stand des im Flashback-Befehl angegebenen Zeitpunkts, im Beispiel der 15. August 2005 um 09:30 Uhr. Vor dem Öffnen der Datenbank mit `open resetlogs` empfiehlt es sich, im Read-Only-Betrieb den Datenbestand zu überprüfen. Nötigenfalls kann man ihn mit `recover database` wiederum nach vorn, mit einem weiteren Flashback zurück rollen.

Aus der Kombination von Standby-System und `flashback database` ergeben sich weitere Optionen: Kommt es in der Primärdatenbank zu einem Fehler, kann man das Standby-System mit `flashback database` auf einen Zeitpunkt vor diesem Eingriff zurücksetzen. Anschließend öffnet man die Datenbank read-only und prüft ihren Inhalt. Ist der gewünschte Zeitpunkt erreicht, kann entweder ein Failover auf die Standby-Datenbank oder ein Export der fehlenden Informationen durchgeführt werden, den man anschließend wieder in die Primärdatenbank importiert. Alternativ lassen sich die Daten über einen Datenbank-Link vom Standby- auf das Primärsystem übertragen. Danach stellt ein Neustart des Recovery die Standby-Datenbank bis zum aktuellen Zeitpunkt wieder her.

## Vorsicht ist die Mutter des Upgrade

Oracles kürzlich erschienene Release 10.2 erweitert den Befehlsumfang; so kann man nun Restore Points setzen. Eine gesicherte Verfügbarkeit der Flashback Logs bis zu diesem Zeitpunkt ist über eine optionale Flashback-Garantie möglich:

```
CREATE RESTORE POINT before_upgrade
  GUARANTEE FLASHBACK DATABASE;
```

Das Reset der Datenbank ist damit zielgenau, sicher und einfach:

```
FLASHBACK DATABASE TO
  RESTORE POINT "before_upgrade";
```

### Listing 1: Aktivieren des Flashback-Log-Mode

```
--Setzen der Größe der Flash Recovery Area
alter system set db_recovery_file_dest_size = 10G;
--Setzen des Pfades der Flash Recovery Area
alter system set db_recovery_file_dest =
  'C:\oracle\flash_recovery_area';
--Datenbank in Mount-Mode
shutdown immediate;
startup mount;
--Aktivieren der Logs
alter database archivelog;
alter database flashback on;
-- Starten der Datenbank
alter database open;
```

### Vor dem Einsatz von `flashback database` sind einige Einstellungen erforderlich.

`drop restore point` entfernt nicht mehr erforderliche Haltepunkte wieder. Eine Liste aller Restore Points liefert eine Abfrage der View `V$RESTORE_POINT`:

```
SELECT NAME, SCN, TIME,
  GUARANTEE_FLASHBACK_DATABASE
  FROM V$RESTORE_POINT
  WHERE
    GUARANTEE_FLASHBACK_DATABASE='YES';
```

Neben `flashback database` gibt es weitere Befehle, die einen Blick in die Vergangenheit des Datenbestandes gestatten: `flashback query` erlaubt dies, ohne die Daten zu ändern. Das Kommando existiert bereits seit Oracle 9i, doch war seine Handhabung umständlich. Weder gab es eine grafische Oberfläche, noch ließ sich mit einfachen Abfragen die Datenhistorie durchforsten. Stattdessen musste das PL/SQL-Package `dbms_flashback` für ein virtuelles Zurücksetzen des Abfragezeitpunkts herhalten. Oracle 10g vereinfacht die Bedienung durch Integration in den grafischen Oracle Enterprise Manager erheblich. Wer lieber mit der Kommandozeile arbeitet, kann seine SQL-Abfragen einfach um `as of timestamp` oder `versions between` erweitern.

Dazu ein Beispiel: Angenommen, die Gehälter aller Mitarbeiter wurden aufgrund eines Fehlers am 1. 8. 2005

### Listing 2: Zurücksetzen der Datenbank

```
-- Datenbank in Mount-Mode
shutdown immediate;
startup mount;
-- Flashback der Datenbank
FLASHBACK DATABASE TO TIMESTAMP
  to_timestamp ('15.08.2005 09:30:00',
    'dd.mm.yyyyhh24:mi:ss');
-- Öffnen der Datenbank mit Reset der Redo Logs
alter database open resetlogs;
```

### Mit einem Flashback-Kommando lässt sich der Zustand der Datenbank zu einem bestimmten Zeitpunkt wiederherstellen.

gegen 20 Uhr auf 0 gesetzt. Eine SQL-Abfrage, die den Datenbestand vor dem Fehler anzeigt, kann etwa so aussehen:

```
SELECT *
  FROM tab_mitarbeiter
  AS OF TIMESTAMP
    to_timestamp('01-08-2004 19:00:00',
      'dd-mm-yyyy hh24:mi:ss');
```

Möchte man die Änderungen eines Datensatzes über einen Zeitraum verfolgen, bietet sich die Anzeige der Satzhistorie an:

```
SELECT mit_no as no, mit_name as name,
  einstell_dat, gehalt,
  versions_operation, versions_starttime as time
  FROM tab_mitarbeiter
  VERSIONS BETWEEN TIMESTAMP MINVALUE
    AND MAXVALUE
  WHERE mit_name = 'Schmidt'
  ORDER BY mit_no, versions_starttime;
```

Oracle gibt daraufhin etwa Folgendes aus:

NO	NAME	EINSTELL	GEHALT	V	TIME
123	Schmidt	01.01.04	0	I	01.08.05 20:22:05
123	Schmidt	01.01.04	5200	U	01.08.05 20:22:32
123	Schmidt	01.01.04	5880	U	01.08.05 20:22:41
123	Schmidt	01.01.04	0	U	01.08.05 20:34:57

Dies zeigt, wann ein Datensatz eingefügt („I“), geändert („U“) oder gelöscht („D“) wurde.

## Ein Rewind-Knopf für Tabellen

`Flashback query` nutzt den Undo-Tablespace der Datenbank. Dieser hat originär eine andere Funktion: Damit es konsistentes Lesen und ein Rollback einer Transaktion gewährleisten kann, hinterlegt Oracle in ihm die Before Images, die Daten vor ihrer Änderung. Diese Informationen gibt es jedoch nach Abschluss einer Transaktion zum Überschreiben frei, sodass sie nur für einen begrenzten Zeitraum verfügbar bleiben. Dieser lässt sich jedoch verlängern, denn der Datenbankparameter `undo_retention` legt fest, wie viele Sekunden Oracle Before Images aufbewahrt. Belegt man den Undo-Tablespace mit der Option `retention guarantee`, hält der Server diese Aufbewahrungszeit zwingend ein und überschreibt keine Information vor Ablauf der Frist.

Allerdings hat diese Technik ihre Grenzen: Bei hohen Transaktionsraten muss der Undo-Tablespace so groß dimensioniert sein, dass er die Infor-

## Flashback-Funktionen

Typ	Architektur	Funktion
Flashback Query	Undo Management	Abfrage der Row History
Flashback Table	Undo Management	Zurücksetzen einer Tabelle auf einen historischen Zeitpunkt
Flashback Drop	Verwendung eines internen Recycle Bin	Wiederherstellen einer gelöschten Tabelle
Flashback Database	Flashback-Logs und Archive-Logs	Zurücksetzen der gesamten Datenbank auf einen historischen Zeitpunkt

mationen für den angegebenen Zeitraum aufnehmen kann.

Will man den Datenbestand einer Tabelle zurücksetzen, hilft *flashback table* weiter. Auch dieses Kommando nutzt den Undo-Tablespace. Damit ein Benutzer eine Tabelle zurücksetzen kann, benötigt er für sie zumindest die Privilegien *flashback table*, *insert*, *update* und *delete* oder das DBA-Privileg *flashback any table*. Letzteres gestattet, jede beliebige Tabelle zurückzusetzen. Außerdem muss die Option *Row Movement* für die Tabelle aktiviert sein:

```
ALTER TABLE tab_mitarbeiter
ENABLE ROW MOVEMENT;
```

Für das eigentliche Wiederherstellen der Daten benutzt man einen Befehl wie den folgenden:

```
FLASHBACK TABLE dept
TO TIMESTAMP
to_timestamp('02.03.2004 09:30,
              'dd.mm.yyyy hh24:mi');
```

Beim Zurücksetzen einer Tabelle ist es wichtig, auf Konsistenz zu achten: Nach dem Löschen aller Mitarbeiter der Abteilung 50 in der Mitarbeitertabelle sowie von Abteilung 50 in der Abteilungstabelle führt das alleinige Zurücksetzen der Mitarbeitertabelle zu inkonsistenten Daten. In solchen Fällen müssen immer alle voneinander abhängigen Tabellen zurückgesetzt werden. Primary-Key- und Foreign-Key-Constraints verhindern auch bei *flashback table* Inkonsistenzen im Datenbestand.

## Tabellen mit dem Grünen Punkt

In früheren Releases war eine Tabelle unwiederbringlich verloren, so

bald man sie mit *drop table* gelöscht hatte. Zwar ließ Oracle die Daten unberührt und gab sie nur zum Überschreiben frei. Jedoch entfernte der Löschbefehl die Metainformationen des Data Dictionary, die Datenblöcke ließen sich der ursprünglichen Tabelle nicht mehr zuordnen.

Oracle 10g brachte eine kleine Änderung: Statt die Tabelle „hart“ zu löschen, verschiebt es sie intern in einen Papierkorb (Recycle Bin). Das geht – unabhängig von der Größe der zu löschenden Tabelle – unglaublich schnell, da intern nur geringfügige Änderungen im Data Dictionary stattfinden. Zur Wiederherstellung erfolgt das Umbenennen einfach in umgekehrter Richtung: Das Objekt erhält wieder seinen alten Namen und Eigentümer – ein ähnlicher Vorgang wie die Nutzung des Papierkorbs in aktuellen GUIs. Ein Beispiel:

```
DROP TABLE dept CASCADE CONSTRAINTS;
```

Dieser Befehl löscht die Tabelle *dept*. Die Option *cascade constraints* bewirkt, dass Oracle dabei die referenzielle Integrität ignoriert und betroffene Foreign-Key-Constraints gleich mit löscht. Der folgende Befehl stellt die Tabelle wieder her:

```
FLASHBACK TABLE dept TO BEFORE DROP;
```

Eine Warnung: Obwohl das Wiederherstellen so leicht und schnell funktioniert, sollte man Experimente damit nur in Testumgebungen durchführen.

Objekte im Recycle Bin bleiben dort nicht beliebig lange liegen. Bei einer Reorganisation, einer Größenänderung des Tablespace und beim Überschreiten der Speicherbegrenzung des Benutzers werden Objekte aus dem Papierkorb entfernt. Darüber hinaus gibt es noch den *purge*-Befehl, der ein

Objekt gezielt endgültig entfernt. Zwei Beispiele:

```
DROP TABLE dept PURGE;
- Löscht die Tabelle endgültig
PURGE RECYCLEBIN;
- Leert den gesamten Recycle Bin
```

## Mülleimer mit Grenzen

Den Inhalt des Recycle Bin gibt der Befehl *show recyclebin* aus. Objekte im Papierkorb lassen sich weiterhin abfragen; DML- oder DDL-Statements (Data Manipulation Language, Data Definition Language) sind jedoch nicht gestattet. Eine Veränderung des Datenbestands ist deshalb ebenso ausgeschlossen wie ein Umbau der Datenstruktur. Möglich ist jedoch ein *flashback query* auf Objekte im Papierkorb, dabei muss man jedoch den etwas kryptischen Systemnamen statt desjenigen vor dem Löschen angeben.

Weitere Einschränkungen des Recycle Bin sind:

- Referenzielle Constraints, etwa Foreign Keys, bestehen nach der Wiederherstellung einer Tabelle nicht mehr. Diese muss man erneut anlegen.
  - Daten von Tabellen, die mit Fine Grained Auditing (FGA) beziehungsweise mit Policies einer Virtual Private Database (VPD) geschützt wurden, sind im Papierkorb nicht gesichert.
  - Partitionierte Index Organized Tables landen derzeit nicht im Recycle Bin.
- Beim Löschen einer Tabelle entfernt Oracle gleichfalls abhängige Objekte wie Indizes, Constraints, LOB-Segmente und Trigger. Bei der Wiederherstellung der Basistabelle sollte es diese zwar automatisch wiederherstellen, erfahrungsgemäß funktioniert das aber nicht immer. So können zum Beispiel Indizes aufgrund von Speicherplatzmangel bereits aus dem Recycle Bin verschwunden sein. Nach Wiederherstellung einer Tabelle sollte man daher die Vollständigkeit aller abhängigen Datenbankobjekte prüfen. (ck)

ANDREA HELD

ist als Systemarchitektin tätig. Sie ist Oracle Certified Professional und Autorin des Buches „Oracle 10g Hochverfügbarkeit“ (Addison-Wesley). Im Fachbereich Informatik der Fachhochschule Frankfurt/Main lehrt sie hochverfügbare und robuste Systeme.



## Flashback im Web

Einführung	<a href="http://www.nyoug.org/10gflashback.pdf">www.nyoug.org/10gflashback.pdf</a>
Kurzbeschreibung	<a href="http://www.oracle.com/technology/deploy/availability/htdocs/Flashback_Overview.htm">www.oracle.com/technology/deploy/availability/htdocs/Flashback_Overview.htm</a>
ausführliche Beschreibung	<a href="http://www.oracle-doku.de/down/Flashback.pdf">www.oracle-doku.de/down/Flashback.pdf</a>