Oracle's 'Undo' button

10 Life Saver Flashback scenarios

By Craig Moir of MyDBA

January 2011

Introducing Flashback technology

To be honest, it is not quite as simple as an 'UNDO' button but is as effective.

Instantly undo, rewind, recover, reverse, restore any of the following with Flashback:

- **Dropped table**
- **Truncated table**
- **Blocks**
- **Standby databases**
- **Deleted rows**
- **Entire Database**
- **Restore Points**
- **Undo SQL**

10 Life Saver Flashback scenarios

# 1 – When a Batch job goes bad

# 2 – Help, I truncated a table on production!

# 3 – Implementations that fail

# 4 – 'WHERE' clause goes wrong

# 5 – Dropped tables

# 6 – Find and Fix

# 7 – Catch that crook!

# 8 – Open 'resetlogs' too soon or too late?

# 9 – Flashback your Data Guard database

# 10 – Corrupt database blocks

MyDBA
Your data in professional hands

1 – When a Batch job goes bad

There are countless reasons for batch processing to go wrong. Normally this requires a restore of tables or databases, which is time consuming.

Not anymore with Flashback!

1. Simply rewind your database using Flashback.
2. Rectify the problem.
3. Restart the batch processing.

Significantly reducing MTTR.

2 – Help, I truncated a table on production!

A DBA's worst nightmare. A developer thought they were on DEV, *truncated* a table, but were actually on PROD!

Flashback will drastically reduce MTTR* for a truncate scenario:

1. **Flashback the database to before the 'truncate'.**
2. **Export the table that was affected.**
3. **Recover the database.**
4. **Import the table.**

** A Standby database can be used instead, therefore eliminating any production downtime!*

3 – Implementations that fail

Implementations, upgrades, data fixes and the like can go wrong.

No worries, and no restore or recovery required either with Flashback.

1. **Simply rewind your database to the starting point or a Restore Point.**
2. **Retry whatever you were doing.**
3. **Easily repeatable.**

Significantly reducing maintenance downtime.

4 – ‘WHERE’ clause goes wrong

Data fixes are sometimes necessary but if you get the ‘where’ clause wrong or forget it completely, then they can be disastrous.

However, with FLASHBACK TABLE a DBA can recover from mistakes even after the ‘commit’ was issued.

Significantly reducing MTTR.

5 – Dropped tables

Dropped a table accidently?

Dropped a table without backing it up first?

Don't stress, just bring it back in a flash with Flashback!

Example: **SQL> flashback table emp to before drop;**

Significantly reducing MTTR.

6 – Find and Fix

Database transactions can be large and complex and will often effect multiple objects within the boundaries of a single transaction. Even though a transaction completes successfully, it does not mean it was correct. Finding where the problem was and fixing it is usually more complicated than a simple Flashback database or Flashback table.

Using Flashback Transaction Query a DBA can do the following:

- ✓ View the sequence of commands for any given transaction/s.
- ✓ View the change made by any particular command for any given transaction/s.
- ✓ Extract the exact Undo SQL command for any of the changes.
- ✓ Manually apply any one or more of the Undo SQL to fix the problem.

7 – Catch that crook!

Flashback is not one of your typical security tools, but if applied cleverly, it could easily be used for auditing purposes to detect fraud and other unusual activity.

Use in conjunction with Flashback Row History, Flashback Transaction History & Flashback Transaction Query.

DBA's can do the following:

- ✓ **View the different versions (changes) of data through time.**
- ✓ **View changes at Transactional level.**
- ✓ **Reconstruct the SQL statements.**
- ✓ **Extract *Undo SQL.***
- ✓ **See who did what and when, and undo it if necessary.**

8 – Open 'resetlogs' too soon or too late?

Database point-in-time recoveries (DBPITR) are often delicate tasks, especially if you need to stop recovery at a very specific point in time. If you stop recovery too soon you may loose data, but if you stop recovery too late you may not fix the problem. DBPITR requires the database to be opened with the 'resetlogs' command. Previously if you did an open 'resetlogs' at the ***wrong*** point in time you needed to restore the whole database from scratch and try again.

Using Flashback you can now:

- ✓ Easily move backwards and forwards in time, **even through a 'resetlogs'.**
- ✓ Quickly and accurately determine the correct PITR to open your database.
- ✓ No time consuming restores if you miscalculate!

9 – Flashback your Data Guard database

Make use of your Data Guard database as a test environment, somewhere to measure the impact changes will have on production, for perform tuning exercises or a suitable environment to run Database Replay on.

Problem solving using Flashback and a Data Guard database:

1. **Simply open your Data Guard database for read/write, test your changes, try out your tuning or run your Database Replay session etc.**
2. **When done just *Flashback* your Data Guard database to before your testing.**
3. **Reinstate your database as Data Guard.**
4. **Resync with production.**

10 – Corrupt database blocks

RMAN does not need to recover the corrupt block from the most recent backup anymore. RMAN simply flashes back the block *instantly* from the Flashback area.

- ✓ **Eliminates relatively slow restores from tape, Tivoli, NetBackup, Legato etc.**
- ✓ **Dramatically reduces database *MTTR.***

MyDBA Consulting Services

For more information on developing and implementing a Flashback strategy or to enquire about MyDBA’s consulting services please contact us on:

info@mydba.co.za

0861 911 DBA

+27 11 431 0930