

A white starburst graphic with a black outline, containing the text "Updated for R2!".

Updated
for R2!

Oracle 10g Flashback Enhancements

Presented at SEOUC
March 2, 2007
Dave Anderson

SKILLBUILDERS



Agenda

- Flashback Evolution
- Flashback Setup
- Flashback Version Query
- FLASHBACK_TRANSACTION_QUERY View
- Flashback Table
- Flashback Database
 - Including Flash Recovery Area



Flashback Evolution

- Oracle9i provided
 - Session-level flashback
 - DBMS_FLASHBACK
 - Statement-level flashback
 - SQL "AS OF" clause

9i R1

9i R2

Can join a table to a previous version of *itself*

```
SQL> select a.lastname, a.total_purchase, nvl(b.total_purchas
2  from sales a , sales AS OF timestamp(sysdate - 1) b
3  where a.cust_no = b.cust_no
4  and a.total_purchase != nvl(b.total_purchase, 0);
```

LASTNAME	TOTAL_PURCHASE	TOTAL_PURCHASE
ANDERSON	55000	0
DASWANI	55000	0

Amount today

Amount yesterday



Flashback Setup

- Flashback query and flashback table* require AUM and sufficient UNDO

```
LINUX> show parameter undo_
```

NAME	TYPE	VALUE
undo_management	string	AUTO
undo_retention	integer	172800
undo_tablespace	string	UNDOTBS1

```

1  select file_name, bytes
2  from dba_data_files
3* where tablespace_name = 'UNDOTBS1'
```

FILE_NAME	BYTES
+ASM_DISK_GROUP1/orcl/datafile/undotbs1.265.1	209715200
+ASM_DISK_GROUP1/orcl/datafile/undotbs1.295.17	104857600

Try to retain 48 hours of Undo



Flashback Version Query...

- Show all “versions” of data between two SCNs or timestamps
- Version is transaction-based
 - Commits create a version
 - Rollbacks do not
- Uses UNDO
- Introduced with Oracle 10g Release 1



...Flashback Version Query...

- Query table with new SQL operator
 - VERSIONS BETWEEN
 - Part of the FROM clause
- Optionally include new pseudo-columns
 - VERSIONS_STARTTIME / STARTSCN
 - Time or SCN when row version was created with commit
 - Can use in Flashback Table to undo changes
 - VERSIONS_OPERATION
 - Type of operation that created row version

Example next...



...Flashback Version Query...

New pseudocolumns

```
DAVE@LINUX> select c2, versions_starttime, versions_endtime,  
2             versions_startscn , versions_endscn,  
3             versions_operation, versions_xid  
4 from system.test  
5     versions between timestamp  
6             to_timestamp('11-SEP-04 12.02.00.000000000 PM',  
7             'dd-mon-yy hh.mi.ss.ff PM')  
8             and systimestamp  
9 where c1 = '1'  
10 order by versions_startscn nulls first;
```

MINVALUE and
MAXVALUE supported

C2	VERSIONS_STARTTIME	VERSIONS_ENDTIME	VERSIONS_STARTSCN
a		11-SEP-04 12.01.55 PM	
b	11-SEP-04 12.01.55 PM	11-SEP-04 12.01.55 PM	988714
c	11-SEP-04 12.01.55 PM	11-SEP-04 12.02.04 PM	988716
d	11-SEP-04 12.02.04 PM	11-SEP-04 12.02.17 PM	988719
x	11-SEP-04 12.02.17 PM		

NULL in ENDTIME means
"x" is the current value



...Flashback Version Query

- Users need privileges to flashback

Privilege to flashback query, flashback version query and flashback table

```
LINUX> grant flashback, select on system.test to dave;  
Grant succeeded.
```



FLASHBACK_TRANSACTION_ QUERY View

- R1 EE feature
- Access undo records for details on transactions
- Not as robust, but easier than logminer utility

```
LINUX> exec print_table('select logon_user, undo_sql -  
> from flashback_transaction_query -  
> where xid = HEXTORAW(''0700010059020000'') ')  
LOGON_USER      : SYSTEM  
UNDO_SQL        : delete from "SYSTEM"."TEST"  
                 where ROWID = 'AAALquAABAAAaIiAAA';
```

Get XID from
flashback version
query

```
LINUX> grant select any transaction to dave;
```

```
Grant succeeded.
```

Grant this or SELECT ANY
DICTIONARY privilege



Flashback Table Concepts

- R1 EE feature
- Turn back hands of time for one or more tables
- Two distinct uses
 - Recover from DROP TABLE
 - Uses “recycle bin” concept
 - See “TO BEFORE DROP” option
 - Recover from application or user changes
 - Uses available UNDO
 - See “TO SCN” and “TO TIMESTAMP” options
 - Deletes and re-inserts rows



Flashback Table Prep

- Enable row movement on table
- Grant privileges to desired user(s)

```
SYSTEM@LINUX> alter table test enable row movement;
```

```
Table altered.
```

```
SYSTEM@LINUX> grant flashback on test to dave;
```

```
Grant succeeded.
```

```
SYSTEM@LINUX> grant alter, select, update, delete,  
2             insert on system.test to dave;
```

```
Grant succeeded.
```



Flashback Table: Undo Application Changes

```
DAVE@LINUX> select dbms_flashback.get_system_change_number  
from dual;
```

X

1057181

Undo the last 30
minutes of changes

Optional, but can
undo flashback if
you know SCN
prior to flashback

```
DAVE@LINUX> flashback table system.test to timestamp  
2          systimestamp - interval '30' minute;
```

```
DAVE@linux3> exec dbms_stats.gather_table_stats  
              (ownname=>'system', tabname=>'test')
```

```
PL/SQL procedure successfully completed.
```

Update statistics after
flashback table



Flashback Table Notes

- Honors RI constraints
- No active transactions
- Cannot flashback across DDL, including TRUNCATE
- 10.1 bug leaves `SYS_TEMP_FBT`
- Default – Triggers disabled
- Indexes kept in sync
- Object restrictions



Flashback Table: Recover Dropped TABLE...

- Dropped tables and dependent objects aren't really dropped
- Just renamed
 - “BINuniquestring”
- This is the “recycle bin” concept
- Space for dropped objects shows as free space
- R1 EE feature



...Flashback Table: Recover Dropped TABLE...

- Objects actually dropped when:
 - Space is needed
 - Could be right away, might be there “forever”
 - Tablespace dropped
 - Recycle bin purged
 - PURGE clause used on DROP
- Recycle bin object is dropped before an autoextend operation



...Flashback Table: Recover Dropped TABLE

- Limitations
 - No External tables, MViews, bitmap join indexes
 - No dictionary managed tablespaces
 - No SYSTEM objects
 - Constraints are not recovered
 - Must manually rebuild
- Can disable with “_recyclebin” parameter



Example: Recover from DROP TABLE

➤ Recover from DROP TABLE

```
SQL> drop table t purge;
```

```
SQL> drop table big;
```

Table dropped.

Table segment remains unless PURGE used

```
SQL> show recyclebin
```

ORIGINAL NAME	RECYCLEBIN NAME
BIG	BIN\$Kq3scQsDQP+jkI0atdsqqQ==\$0

Table is recorded in "recycle bin"

OBJECT TYPE	DROP TIME
TABLE	2004-08-19 11:40:00

Easy and fast to recover table

```
SQL> flashback table big to before drop;
```

Flashback complete.

See also the optional "RENAME TO" clause



LIFO to the Recycle Bin

```
SQL> show recyclebin
```

ORIGINAL NAME	RECYCLEBIN NAME	OBJECT TYPE	DROP TIME
T	BIN\$KbsjpLR0Ts6+eG0GqifipQ==\$0	TABLE	2004-09-19:14:59:30
T	BIN\$y0AR2dYXSnC0bRn+vJTf0Q==\$0	TABLE	2004-08-19:16:52:54

```
SQL> flashback table t to before drop;
```

LIFO

Flashback complete.

```
SQL> show recyclebin
```

ORIGINAL NAME	RECYCLEBIN NAME	OBJECT TYPE	DROP TIME
T	BIN\$y0AR2dYXSnC0bRn+vJTf0Q==\$0	TABLE	2004-08-19:16:52:54

```
SQL> rename t to t_old;
```

Table renamed.

```
SQL> flashback table t to before drop;
```

Flashback complete.

```
SQL> show recyclebin
```

```
SQL>
```



Purging Recycle Bin

```
SQL> show recyclebin
```

ORIGINAL NAME	RECYCLEBIN NAME	OBJECT TYPE	DROP TIME
T2	BIN\$cfmTndAnQk+junRG6hCngg==\$0	TABLE	2004-09-19:14:59:28
T_AUDIT	BIN\$Fx+nQHXC4qgr4fSc18QiQ==\$0	TABLE	2004-08-23:15:52:29
T_AUDIT	BIN\$mPxGfYd8QJ+sLvSNPLszZg==\$0	TABLE	2004-08-23:15:51:13
T_AUDIT	BIN\$7sKuyR6YR7KM4Bc7mzewiQ==\$0	TABLE	2004-08-23:15:19:19
T_AUDIT	BIN\$ZD1sowFXTt6T7uFsmDt4aw==\$0	TABLE	2004-08-23:13:50:33
T_AUDIT	BIN\$STuYtaPCQcC0eEhgZd+HRw==\$0	TABLE	2004-08-23:13:49:26
T_AUDIT	BIN\$Em072EijQIilk8I5QhHFrA==\$0	TABLE	2004-08-23:13:48:08
T_AUDIT	BIN\$nuPfG+vCT8OsZDi4lJ0jdA==\$0	TABLE	2004-08-23:13:05:05

```
SQL> purge recyclebin;
```

```
Recyclebin purged.
```

```
SQL> show recyclebin
SQL>
```

PURGE clears all
objects from the bin



Flashback Database...

- Alternative to point-in-time recovery
- Undo transactions
 - *Not* for lost or corrupt datafiles
- Easier than traditional PIT recovery
- Better performance than point-in-time recovery
 - Write changed blocks versus *restore* then apply transactions
- Can easily re-flashback more than once
- R1 EE feature



...Flashback Database

- Flashback logs used to recover database
- Logs contain changed blocks
- Automatic and ongoing creation of new logs to capture changes
 - Oracle claims 2% overhead
- Written to “Flash Recovery Area”
- Automatic deletion of obsolete logs

See tuning tips in notes and undoc'd parameters such as
`_flashback_log_min_size`
`_flashback_log_size`



Configure Database

- Archivelog mode
- Configure flash recovery area
 - Destination, size limit, retention
- Flashback mode

```
SQL> alter database flashback on;
```

```
Database altered.
```

```
SQL> select flashback_on from v$database;
```

```
FLA
```

```
---
```

```
YES
```

Starts recovery writer
background process (RVWR)



Flash Recovery Area: Concepts

- Storage area for all recovery-related files
 - Redo logs , Archive logs
 - Control file backups
 - RMAN backups
 - Flashback logs
- Optional, but Oracle recommends
 - Required for Flashback Database and Guaranteed Restore Points
- Use directory, file system or ASM
- Put on separate disks from data and log disks



Flash Recovery Area: Benefits

- Simplify management
 - Auto-delete files when space needed
 - Obsolete files
 - Files backed up to tape
 - Issues warnings when space constrained
- Easy to backup all recovery-related files to tape

```
RMAN> BACKUP RECOVERY AREA;
```

- Fast recovery with

```
RMAN> SWITCH DATABASE TO COPY;
```



Flash Recovery Area: Configuration

- Set location
 - Separate from datafiles
- Set space limit
 - Maximum space dedicated to flashback area
 - database size + size of incremental backups + size of all archive logs not backed up to tape
- If Flashback Database desired, set retention limit
 - How far back can we flashback database?

NAME	VALUE
-----	-----
db_recovery_file_dest	/mnt/mickeymantle
db_recovery_file_dest_size	10G
db_flashback_retention_target	1440

Location points to db_recovery_file_dest

Size Limit points to db_recovery_file_dest_size

How long to keep flash logs points to db_flashback_retention_target



Flash Recovery Area: Monitoring...

➤ V\$RECOVERY_FILE_DEST

```
SQL> exec print_table('select * from v$recovery_file_dest')
NAME                               : /mnt/mickeymantle/
SPACE_LIMIT                         : 10737418240
SPACE_USED                          : 2485816832
SPACE_RECLAIMABLE                   : 19995648
NUMBER_OF_FILES                     : 35
-----
```

How much space can be made available through delete of “obsolete, redundant or low priority files”

Total number of archive and flashback logs in the recovery area



...Flash Recovery Area: Monitoring

➤ V\$FLASH_RECOVERY_AREA_USAGE



```
SQL> exec system.print_table -
>      ('select * from v$flash_recovery_area_usage')
FILE_TYPE           : ARCHIVELOG
PERCENT_SPACE_USED  : 6.02
PERCENT_SPACE_RECLAIMABLE : 0
NUMBER_OF_FILES     : 5
-----
FILE_TYPE           : BACKUPPIECE
PERCENT_SPACE_USED  : .21
PERCENT_SPACE_RECLAIMABLE : .1
NUMBER_OF_FILES     : 2
-----
```



Flashback Database: Example

```
SQL> startup mount
```

Flashback to SCN or timestamp.
Need SYSDBA privilege.

```
SQL> flashback database to scn 1427369;
```

```
Flashback complete.
```

Can open READ ONLY to
check things out

```
SQL> alter database open read only;
```

```
Database altered.
```

If necessary, re-mount, flashback to
different SCN or time

```
SQL> alter database open resetlogs;
```

```
Database altered.
```

Resetlogs deletes old FB logs



R2 Flashback Database



- “TO BEFORE RESETLOGS” option
 - Undo an OPEN RESETLOGS
 - Restore to the previous incarnation.
- Restore points
 - Convenient restore marker

V\$RESTORE_POINT records details about restore points

```
CREATE RESTORE POINT x GUARANTEE FLASHBACK DATABASE;  
.   
. (run updates)   
.   
FLASHBACK DATABASE TO RESTORE POINT x;
```



Summary...

- Flashback Query (Oracle9i)
 - Session or sub-statement level
 - Query data at a previous point-in-time
- Flashback Version Query
 - Show versions of data
 - “VERSIONS BETWEEN” clause
- Flashback_Transaction_Query View
 - Access transaction details, including undo-SQL
 - EE feature



...Summary

- Flashback Table
 - “Reset” table to previous point-in-time
 - Recover from dropped table
 - EE/PE feature
- Flashback Database
 - Point-in-time recovery for entire database
 - Must configure database to create flashback logs
 - R2 supports
 - Restore points
 - “TO BEFORE RESETLOGS”
 - EE/PE feature



The End

- Thanks for listening!
- Please complete your evaluation
 - Dave Anderson
 - Oracle 10g New Features: Flashback Enhancements
- For more information
 - dave@skillbuilders.com
 - www.skillbuilders.com

FREE Tech papers and
Presentations

FREE Scripts

FREE Oracle Tutorials

FREE Practice Exams