# Practices for Lesson 20: Using Logical Flashback Features

Practices for Lesson 20: Overview

Overview

In these practices, you will configure your database for the use of flashback technologies. You will then use flashback technologies to restore a dropped table and reverse the actions of a transaction.

Practice 20-1: Preparing to Use Flashback Technologies

Overview

In this practice, you will configure the database to use the flashback transaction and flashback table features.

Assumptions

You have two terminal windows open in which you are logged in as the oracle OS user and

/home/oracle/labs/DBMod\_Flashback is the current directory. Each terminal window is connected to the orclpdb1 instance.

Tasks

Open a terminal window. Use SQL\*Plus to connect to ORCLPDB1 as the SYS user.

Determine how far the undo data will allow you to flash back transactions in the database. The V$UNDOSTAT view contains up to four days of statistics, over each 10-minute interval. The first row contains the current (partial) time period statistics. (*Your value may be different.*)

**Question:** What does the value of the TUNED\_UNDORETENTION time represent? **Answer:** The number of seconds the data is being retained in the database at any given time. Undo retention is not guaranteed by default. If the system needs more space, the

Oracle Database server can overwrite unexpired undo with more recently generated undo

data.

Set the UNDO\_RETENTION parameter and RETENTION GUARANTEE clause on the undo tablespace to guarantee retention for 24 hours. Change the properties of the tablespace so you will not run out of space in the tablespace.

Change the UNDO\_RETENTION parameter to 14400 seconds (4 hours).

**Note:** Increasing the value of UNDO\_RETENTION to more than a few days can cause unreasonable growth of the undo tablespace.

Determine the name of the undo tablespace.

Change the RETENTION GUARANTEE value of the undo tablespace.

Find the names of the data files associated with the UNDOTBS1 tablespace. Note the

FILE\_ID value.

Configure the undo tablespace data file to automatically extend if more space is needed to keep unexpired undo and active undo records. Use your own FILE\_ID value instead of <*n>*.

**Question:** What happens if undo retention is guaranteed and no more space is available for active undo records (either because the undo tablespace has filled, reached the maximum size, or there is no more space left on the storage device [disk])?

**Answer:** Transactions fail due to lack of space in the undo tablespace.

View the value of the RECYCLEBIN parameter, then exit SQL\*Plus.

**Note:** ON is the default value.

Practice 20-2: Restoring a Dropped Table

Overview

In this practice, you will recover a table that has been dropped.

Assumptions

The RECYCLEBIN parameter is set to ON (which you confirmed in Practice 2-1 Step 4).

You have two terminal windows open in which you are logged in as the oracle OS user and

/home/oracle/labs/DBMod\_Flashback is the current directory. Each terminal window is connected to the orclpdb1 instance.

Tasks

Execute the setup\_02\_02.sh script to create the practice environment. The script saves its output in the /home/oracle/labs/DBMod\_Flashback/setup.log file.

Execute the break\_02\_02.sh script to simulate the work done by a developer. The script saves its output in the /home/oracle/labs/DBMod\_Flashback/break.log file.

A developer, who has been using the BAR Oracle user account comes to you and asks you to restore a table that was dropped. This table has had several iterations but the one the developer needs was named BAR102 in the BAR schema. It should have 12 columns, one of which was named LOCATION\_ID. There is currently a BAR102 table in the BAR schema. Restore the requested table to BAR102A.

Attempt to use the SHOW RECYCLEBIN command to view the contents of the recycle bin for orclpdb1.

**Note:** The SHOW RECYCLEBIN command shows only those objects that belong to the *current* user. Because you are logged in as the SYS user, the SHOW RECYCLEBIN command does not show the dropped tables you are interested in restoring.

Examine the objects in the DBA\_RECYCLEBIN view. Optionally, change the SQL\*Plus page size to 99 lines.

**Note:** In the preceding output, you see the same object dropped twice at different points in time. With the time stamp, you determine which version of the table you really want to restore.

Determine which object contains the column of interest. *Your object names will be different*. *Use your own values from the previous step*.

**Hint:** Select the column from the tables, the one which does not report an error is the one you want.

**Note:** The object name from the recycle bin must be in double quotation marks because it may contain special characters.

Restore the object that has the correct columns. *Use the correct object name that you found in the previous step*.

Confirm that the BAR.BAR102A table has been restored by selecting the first row. Then exit SQL\*Plus.

**Note**: It does not matter which row is displayed from the query.

Clean up from this practice by executing the cleanup\_02\_02.sh script.

**Note:** This script uses the PURGE DBA\_RECYCLEBIN command to remove all objects from the recycle bin. The script saves its output in the

/home/oracle/labs/DBMod\_Flashback/cleanup.log file.

Practice 20-3: Using Flashback Table

Overview

In this practice, you will use flashback table to reverse a transaction.

Assumptions

Practice 2-1 has been completed.

You have two terminal windows open in which you are logged in as the oracle OS user and

/home/oracle/labs/DBMod\_Flashback is the current directory. Each terminal window is connected to the orclpdb1 instance.

Tasks

Execute the setup\_02\_03.sh script to create the user and tables used in this practice. The tables have a foreign-key relationship. The script saves its output in the

/home/oracle/labs/DBMod\_Flashback/setup.log file.

Determine the current time to the nearest second. Record this as T1.

**Note:** Your date will be different.

Open a new terminal window and execute the break\_02\_03.sh script. This simulates a transaction that scrambles the data in the BARCOPY and BARDEPT tables. There is a foreign key constraint between BARCOPY and BARDEPT. The script saves its output in the

/home/oracle/labs/DBMod\_Flashback/break.log file.

Return to the SQL\*Plus terminal session connected to ORCLPDB1 as the SYS user. The HR representative reports that an employee has incorrectly changed the names of the departments, and scrambled which employees are assigned to which departments. The tables were correct at time T1 and no authorized changes have been made since that time. The tables involved are BAR.BARCOPY and BAR.DEPT. Restore the tables to the state they were at T1 (determined in step 2).

Because there is a foreign-key relationship, both tables must be restored. Continue in your existing SQL\*Plus session.

Check that you have correctly restored the tables. The results of the following query should match yours, although the row order may be different. Then exit SQL\*Plus.

Clean up the practice environment by running the cleanup\_02\_03.sh script.