

CST8276: Lab 6 – Flashback Recovery

PURPOSE:

This lab provides you with the opportunity to gain hands-on experience with an important DBMS features - **Flashback Recovery** – as implemented in Oracle.

BACKGROUND:

Flashback Recovery is a very useful feature that allows an authorized user to recover an object at a particular point in time. Conceptually, Flashback Recovery is similar to a substantially more powerful version of the *Recycle Bin* in MS Windows. We will look at one aspect of Flashback Recovery - recovering a dropped table and its Primary Key constraint implemented as an index. The feature also has much greater capabilities (for example, recovering data at a point in time, or recovering the entire database).

REQUIREMENTS:

Complete the following tasks, submit your solution via BrightSpace and demo the lab to your lab professor. You can earn a maximum of 2 marks towards your lab mark for a complete, correct and on-time submission.

TASKS:

1. Logon to SYS AS SYSDBA:

- a. To use the FLASHBACK RECOVERY feature you need to confirm a few parameters have been set to appropriate values. You will (hopefully) recall from an earlier lab that the SPFILE and PFILE files contain initialization parameters. These parameters are read at startup and remain in effect until altered or the instance is restarted – depending on the type of parameter being considered. Running the following query ***SELECT COUNT(*) FROM V\$PARAMETER;*** indicates there are 366 parameters – though there are more than another 900 parameters that aren't documented (i.e., only to be changed on instruction from Oracle support).
- b. Enter: ***SHOW PARAMETERS UNDO*** to determine the values for parameters UNDO_MANAGEMENT, UNDO_TABLESPACE, and UNDO_RETENTION:
 - i. List your values for:
 1. UNDO_MANAGEMENT: AUTO_____
 2. UNDO_TABLESPACE: UNDOTBS1_____
 3. UNDO_RETENTION: 900_____
 - ii. The SHOW PARAMETER command in SQLPlus in essence issues a ***SELECT NAME FROM V\$PARAMETER*** query. The V\$PARAMETER view however, contains much more valuable information. Let's take a look at some of this information.
https://docs.oracle.com/cd/B28359_01/server.111/b28320/dynviews_2085.htm#REFRN30176
 1. First, use SQLPlus column formatting to change the display format of V\$PARAMETER's NAME, VALUE and DESCRIPTION columns to A20 (e.g., COLUMN NAME FORMAT a20).
 2. Enter:
SELECT name, value, is_iss_modifiable, issys_modifiable, description FROM V\$PARAMETER

CST8276: Lab 6 – Flashback Recovery

WHERE lower(name) like 'nls_date%'

Paste the result below:

```
SQL> SELECT name, value, isses_modifiable, issys_modifiable, description
2 FROM V$PARAMETER
3 WHERE lower(name) like 'undo%';
```

NAME

VALUE

ISSES ISSYS_MOD

DESCRIPTION

undo_management

AUTO

FALSE FALSE

instance runs in SMU mode if TRUE, else in RBU mode

NAME

VALUE

ISSES ISSYS_MOD

DESCRIPTION

undo_tablespace

UNDOTBS1

FALSE IMMEDIATE

use/switch undo tablespace

3. Repeat the above command changing the *like 'undo%'* condition to *like 'nls_date%'*

Paste the result below:

CST8276: Lab 6 – Flashback Recovery

```
SQL> SELECT name, value, isses_modifiable, issys_modifiable, description
  2  FROM V$PARAMETER
  3  WHERE lower(name) like 'nls_date%'
  4  ;
```

NAME

VALUE

ISSES ISSYS_MOD

DESCRIPTION

nls_date_language

AMERICAN

TRUE FALSE

NLS date language name

NAME

VALUE

ISSES ISSYS_MOD

DESCRIPTION

nls_date_format

DD-MON-RR

TRUE FALSE

4. Enter: *SELECT SYSDATE FROM DUAL;*

Paste the result below:

```
SQL> SELECT SYSDATE FROM DUAL;
```

SYSDATE

25-JUN-21

SQL> _

iii. Complete the following table based on your recent results:

Name	Value	ISSES_ Modifiable	ISSYS_ Modifiable
UNDO_MANAGEMENT	Auto	False	False
UNDO_TABLESPACE	Undotbs1	False	Immediate
UNDO_RETENTION	900	False	Immediate
NLS_DATE_FORMAT	dd-mon-rr	True	False

CST8276: Lab 6 – Flashback Recovery

- iv. Even if your `UNDO_MANAGEMENT` parameter value is currently 'AUTO', enter the following commands (Note: some will fail):

ALTER SYSTEM SET UNDO_MANAGEMENT=AUTO;

ALTER SYSTEM SET UNDO_MANAGEMENT=AUTO SCOPE=SPFILE;

***ALTER SYSTEM SET UNDO_MANAGEMENT=AUTO
SCOPE=MEMORY;***

ALTER SYSTEM SET UNDO_MANAGEMENT=AUTO SCOPE=BOTH;

ALTER SYSTEM SET UNDO_RETENTION = 1200 SCOPE=BOTH;

***ALTER SESSION SET NLS_DATE_FORMAT ='DAY MONTH
DD,YYYY';***

(Note: If your `SHOW PARAMETERS UNDO_MANAGEMENT` command returned the value `MANUAL` instead of `AUTO`, then you will also need to "bounce" (i.e., shutdown then start) the database.

- v. Confirm your changes were effective by running the following query and copying the results below:

***SELECT name, value
FROM V\$PARAMETER
WHERE lower(name) like 'undo%' OR lower(name) like 'nls_date%';***

CST8276: Lab 6 – Flashback Recovery

```
SQL> SELECT name, value
  2  FROM V$PARAMETER
  3  WHERE lower(name) like 'undo%' OR lower(name) like 'nls_date%' ;

NAME
-----
VALUE
-----
nls_date_language
AMERICAN

nls_date_format
DAY MONTH DD,YYYY

undo_management
AUTO

NAME
-----
VALUE
-----
undo_tablespace
UNDOTBS1

undo_retention
1200
```

- vi. Confirm your SPFILE change was effective (i.e., from your ***ALTER SYSTEM SET UNDO_RETENTION = 1200 SCOPE=BOTH;*** command) by creating a text pfile from the spfile (i.e., ***CREATE PFILE FROM SPFILE***) and copying the UNDO_RETENTION parameter and value below:

```
SQL> ALTER SYSTEM SET UNDO_RETENTION = 1200 SCOPE=BOTH;

System altered.

SQL> CREATE PFILE FROM SPFILE;

File created.
```

2. Logon as a general user.

CST8276: Lab 6 – Flashback Recovery

- a. Create a basic sequence named Student_seq (i.e., that starts at 1 and increments by 1). Refer to this online Oracle document page for syntax help:

http://docs.oracle.com/database/121/SQLRF/statements_6017.htm#SQLRF01314

Note: if the account is lacking privileges, use sqlplus (sqlplus / as sysdba) to grant the account the necessary privileges.

Show your work below.

```
SQL> CREATE SEQUENCE STUDENT_SEQ START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;  
  
Sequence created.  
  
SQL> _
```

Enter: *Set Timing On*

Create a table Students with the following columns:

- i. Sid, with a datatype of number, -- this is the primary key, give the constraint an obvious name (e.g., St_sid_pk)
- ii. Fname, with a datatype of varchar2(12), and
- iii. Lname, with a datatype of varchar2(24)

Show your work below.

```
SQL> CREATE TABLE STUDENTS (  
2  SID NUMBER,  
3  FNAME VARCHAR2(12),  
4  LNAME VARCHAR2(24),  
5  PRIMARY KEY(SID));  
  
Table created.
```

Run the following PL/SQL block to insert 1000 records into the *Students* table:

```
DECLARE  
i NUMBER := 1;  
BEGIN  
  LOOP  
    INSERT INTO Students  
    VALUES ( Student_seq.nextval,  
              concat('FName_', to_char(Student_seq.currval)),  
              concat('LName_', to_char(Student_seq.currval))  
            );  
    Commit;  
    i := i+1;  
    EXIT WHEN i>1000;  
  END LOOP;  
END;  
/
```

CST8276: Lab 6 – Flashback Recovery

Show your work below.

```
SQL> DECLARE
  2  i  NUMBER := 1;
  3  BEGIN
  4      LOOP
  5          INSERT INTO Students
  6          VALUES ( Student_seq.nextval,
  7                  concat('FName_', to_char(Student_seq.currval)),
  8                  concat('LName_', to_char(Student_seq.currval))
  9                  );
 10      Commit;
 11      i := i+1;
 12      EXIT WHEN i>1000;
 13  END LOOP;
 14  END;
 15
 16 /

PL/SQL procedure successfully completed.
```

How long did it take to insert the 1000 records in the Students table (in seconds)?

0.23 seconds

- b. Create a sequence named ***Emp_seq*** that **starts at 1 and increments by 2**. Create an ***Employees*** table with the following columns:
- Ee#, with a datatype of number, -- this is the primary key. Give the constraint an obvious name (e.g., Emp_ee_pk).
 - First, with a datatype of varchar2(12), and
 - Last, with a datatype of varchar2(24)

Run the following PL/SQL block to insert 1000 records into the ***Employees*** table.

```
DECLARE
e NUMBER := 1;
BEGIN
  LOOP
    INSERT INTO Employees
    VALUES ( Emp_seq.nextval,
              concat('First_', to_char(Emp_seq.currval)),
              concat('Last_', to_char(Emp_seq.currval))
            );
    e := e+1;
    EXIT WHEN e>1000;
  END LOOP;
Commit;
```

CST8276: Lab 6 – Flashback Recovery

END;

/

How long did it take to insert the 1000 records in the Employees table?

0.14 seconds

How you would explain the difference in the time required to insert 1000 records in the Students and Employees tables?

It's because the commit statement is placed differently.

- c. Create an index on *Students.Lname* and an index on *Employees.Last*
- d. Enter: *purge recyclebin*
Enter: *show recyclebin*
- e. Delete all of the **data** in the Employees table
Enter: *SELECT count(*) FROM EMPLOYEES* to confirm the rows were deleted.
Next issue a **ROLLBACK**
Re-enter *SELECT count(*) FROM EMPLOYEES* to confirm the transaction was cancelled and the rows were recovered.
Copy your statements and results here:

```
SQL> PURGE RECYCLEBIN;

Recyclebin purged.

SQL> SHOW RECYCLEBIN;
SQL> SELECT COUNT(*) FROM EMPLOYEES;

  COUNT(*)
-----
       1000

SQL> ROLLBACK;

Rollback complete.

SQL> SELECT COUNT(*) FROM EMPLOYEES;

  COUNT(*)
-----
       1000
```

- f. Drop the Employees table,
Enter *SHOW RECYCLEBIN*

CST8276: Lab 6 – Flashback Recovery

Enter ***SELECT * FROM RECYCLEBIN;*** These two commands should show the same results.

Copy your statements and results here:

```
SQL> drop table employees;

Table dropped.

SQL> show recyclebin;
ORIGINAL NAME      RECYCLEBIN NAME      OBJECT TYPE  DROP TIME
-----
EMPLOYEES          BIN$4/uIBhasSCO/S52Yfdi2Zw==$0  TABLE      2021-06-22:10:50:40
SQL> select * from recyclebin;
```

- g. Enter ***FLASHBACK TABLE Employees TO BEFORE DROP;***
Enter ***SELECT count(*) FROM EMPLOYEES;***
Enter ***INSERT INTO Employees VALUES (1, 'PK', 'Oops');***
Enter ***SELECT * FROM Employees WHERE ee# = 1;***
Copy your statements and results here:

```
SQL> FLASHBACK TABLE EMPLOYEES TO BEFORE DROP;
FLASHBACK TABLE EMPLOYEES TO BEFORE DROP
*
ERROR at line 1:
ORA-38305: object not in RECYCLE BIN
```

Has the dbms permitted the above insert? What does this mean?

It wouldn't allow because of the constraint.

- h. Drop the Employees table again.
Create a new Employees table from the Students table using the CTAS (Create Table As Select) approach described here:
http://www.techonthenet.com/sql/tables/create_table2.php
Make sure rename the fields so that the Last -> Lname, First -> Fname, and Sid->EE#
Show your SQL and result below:

Enter ***FLASHBACK TABLE Employees TO BEFORE DROP;***
What resulted and why?

- i. Enter ***FLASHBACK TABLE Employees TO BEFORE DROP RENAME TO Employees2;***

CST8276: Lab 6 – Flashback Recovery

Confirm the table was recovered by either **selecting * from the recovered table** or listing user's tables created and recovered in this lab.

--

You're Done!