

Practice 13

Performing Recovery Part III

Practice Target

In this practice you will perform further recovery scenarios on `ORADB` database.

Practice Overview

In this practice, you will implement the following recovery procedures:

- Perform tablespace point-in-time recovery (TSPITR) on `soetbs` tablespace.
- Recover specific tables using `RMAN RECOVER` command to a specific point-in-time in the past.

Assumptions

This practice assumes the `srv1` appliance is up and running and its database `ORADB` is running in `OPEN` state.

Pre-requisites

Take a **snapshot** of `srv1` appliance. Name the snapshot "**Practice 13 Start**".

Caution!

Do not implement the practice without creating the snapshot as instructed by the previous step.

Recovery Scenario 7:

Performing Tablespace Point-in-Time Recovery (TSPITR)

Scenario Target:

- We need to rewind `soetbs` tablespace to some point-in-time in the past without affecting the other tablespaces in the database. It is required to preserve the objects that were created after the incident.

Scenario assumptions:

- The database is running in `ARCHIVELOG` mode
- Backup of the database is available

Preparing for the scenario

1. Invoke RMAN with connecting to `ORADB` as target and take a full backup of the database as backupset.

```
rman target /  
BACKUP DATABASE TAG 'DB_FULL';  
ALTER SYSTEM SWITCH LOGFILE;
```

Simulating the data loss

2. Simulate data loss by performing the following steps:

- a. Create a testing user.

```
CREATE USER TUSER IDENTIFIED BY oracle DEFAULT TABLESPACE soetbs QUOTA UNLIMITED ON  
soetbs;  
GRANT CONNECT, CREATE TABLE TO TUSER;
```

- b. Create the following table.

```
CREATE TABLE TUSER.test1 TABLESPACE soetbs AS SELECT TABLE_NAME FROM DBA_TABLES;
```

- c. Take a note of the current time.

```
SELECT TO_CHAR(SYSDATE, 'YYYY-MM-DD:HH24:MI:SS') FROM DUAL;
```

- d. Issue the following command:

```
ALTER SYSTEM SWITCH LOGFILE;
```

- e. Drop the table.

This action is the destructive unintentional action that you want to recover from. We made the action simple because we concentrate on the recovery procedure.

```
DROP TABLE tuser.test1;
```

- f. Switch the redo logfile

```
ALTER SYSTEM SWITCH LOGFILE;
```

- g. Create another table and create an index on it. Those are the objects that were created after the incident.

```
CREATE TABLE TUSER.test2 TABLESPACE soetbs AS SELECT TABLE_NAME FROM DBA_TABLES;
CREATE INDEX TUSER.test2_ndx ON TUSER.test2 (TABLE_NAME) TABLESPACE soetbs;
```

Recovery Actions

The business noticed the data loss and reported the incident to you asking you to recover it. The database contains data for some other applications. It is required to recover the lost data without affecting the operations on the other applications. It is required to keep the changes on the tablespace that occurred after the incident.

Note: DBPITR is not an option because, it makes the other tablespaces unavailable during the recovery time.

3. Start SQL*Plus and connect as sysdba to ORADB.

```
sqlplus / as sysdba
```

4. Check if the tablespace set is self-contained.

```
exec DBMS_TTS.TRANSPORT_SET_CHECK('SOETBS', TRUE,TRUE);
```

5. Make sure no violation is reported.

If any violation is reported, you have to fix it before you proceed.

```
SELECT * FROM TRANSPORT_SET_VIOLATIONS;
```

6. Determine which objects created after the target recovery time.

You are expected to export those objects using Export Data Pump utility. We are not going to do that now just to concentrate on our target of this practice.

```
col OWNER for a10
col NAME for a10
col TABLESPACE_NAME for a10

SELECT OWNER, NAME, TABLESPACE_NAME
FROM TS_PITR_OBJECTS_TO_BE_DROPPED
WHERE TABLESPACE_NAME IN ('SOETBS')
AND CREATION_TIME > TO_DATE('<recovery point>','YYYY-MM-DD:HH24:MI:SS')
ORDER BY 1,2;
```

7. Create the following directory to save the auxiliary files in it.

You have to make sure that the auxiliary database destination has enough free space to accommodate the restored auxiliary database.

```
host mkdir /media/sf_extdisk/auxdest
```

8. Invoke RMAN, connect to ORADB as target and perform the TSPITR on soetbs tablespace.

After the recovery is finished, the tablespace will be in offline.

```
rman target /

RECOVER TABLESPACE soetbs UNTIL TIME "to_date('<recovery point>','YYYY-MM-DD:HH24:MI:SS')"
```

```
AUXILIARY DESTINATION '/media/sf_extdisk/auxdest';
```

Note: In my testing environment, I once received the following errors when running the command above.

```
EXPDP> ORA-39126: Worker unexpected fatal error in KUPW$WORKER.UNLOAD_DATA  
[TABLE_DATA:"SYS"."TSPITR_EXP_*"]  
  
ORA-20000: Unable to gather statistics concurrently: Resource manager plan is not  
active or is not managing CPU usage
```

To resolve the issue, I had to execute the following command and then try the `RECOVER TABLESPACE` command again.

```
exec DBMS_STATS.GATHER_FIXED_OBJECTS_STATS;
```

If it does not work, try the following:

```
exec DBMS_STATS.SET_GLOBAL_PREFS('CONCURRENT','OFF');
```

9. Back up the recovered tablespace.

```
BACKUP TABLESPACE soetbs TAG 'TBS SOETBS';
```

10. Bring the tablespace online.

```
ALTER TABLESPACE soetbs ONLINE;
```

11. Verify that the data has been successfully recovered.

```
SELECT COUNT(*) FROM tuser.test1;
```

Note: after recovering the lost data, import the objects that you exported earlier.

Clean up

12. Delete the subdirectories created in the shared folder.

```
host "rm -r /media/sf_extdisk/auxdest/ORADB";
```

13. Shutdown `srv1` and restore it to the snapshot "**Practice 13 Start**".

Recovery Scenario 8:

Recovering Tables using RMAN RECOVER Command

Scenario Target:

- This recovery scenario aims at recovering specific tables to specific point-in-time in the past. The application end user deleted a customer together with its orders and it is required to recover the deleted data.
- To keep the changes made on the tables after the incident, the data will be recovered into new tables.

Scenario assumptions:

- The database is running in ARCHIVELOG mode
- Backup of the database is available

Preparing for the scenario

14. Invoke RMAN with connecting to ORADB as target and take a full backup of the database as backupset.

```
rman target /  
BACKUP DATABASE TAG 'DB_FULL';  
ALTER SYSTEM SWITCH LOGFILE;
```

Simulating the data loss

15. Simulate data loss by performing the following steps:

- a. Start SQL*Plus and login as soe user

```
sqlplus soe/soe@oradb
```

- b. Take a note of the current time.

```
SELECT TO_CHAR(SYSDATE, 'YYYY-MM-DD:HH24:MI:SS') FROM DUAL;
```

- c. Run the following query and take a note of the returned CUSTOMER_ID. This is a customer who has at least one order registered under his account.

```
SELECT CUSTOMER_ID FROM SOE.ORDERS WHERE ROWNUM=1;
```

- d. Issue the following commands replacing the <customer_id> with the number obtained from the previous step.

Those are the destructive actions that you want to recover from.

```
DELETE ORDER_ITEMS WHERE ORDER_ID IN ( SELECT ORDER_ID FROM ORDERS WHERE  
CUSTOMER_ID=<customer_id>);  
DELETE FROM ORDERS WHERE CUSTOMER_ID=<customer_id>;  
DELETE FROM ADDRESSES WHERE CUSTOMER_ID=<customer_id>;  
DELETE CUSTOMERS WHERE CUSTOMER_ID=<customer_id>;  
COMMIT;
```

- e. Connect as sysdba and switch the redo logfile

```
conn / as sysdba
ALTER SYSTEM SWITCH LOGFILE;
```

Recovery Actions

You will recover the tables to the given recovery time. You do not want to lose the data entered in the original tables after the incident, therefore, you will restore the tables into new tables.

16. Create a staging user.

```
CREATE USER TUSER IDENTIFIED BY oracle DEFAULT TABLESPACE soetbs QUOTA UNLIMITED ON
soetbs;
GRANT CONNECT, CREATE TABLE TO TUSER;
```

17. Make sure the auxiliary database directory exists. If it does not, create it.

```
host ls -al /media/sf_extdisk/auxdest
```

18. Invoke RMAN, connect to ORADB as target and perform the PITR on the given tables. Replace the <recovery point> with the time noted earlier.

```
rman target /

RECOVER TABLE soe.ORDER_ITEMS, soe.ORDERS, soe.ADDRESSES, soe.CUSTOMERS
UNTIL TIME "to_date('<recovery point>','YYYY-MM-DD:HH24:MI:SS')"
```

AUXILIARY DESTINATION '/media/sf_extdisk/auxdest'

```
REMAP TABLE 'SOE'. 'ORDER_ITEMS': 'TUSER'. 'ORDER_ITEMS_B',
             'SOE'. 'ORDERS': 'TUSER'. 'ORDERS_B',
             'SOE'. 'ADDRESSES': 'TUSER'. 'ADDRESSES_B',
             'SOE'. 'CUSTOMERS': 'TUSER'. 'CUSTOMERS_B';
```

19. Verify that the data has been successfully recovered.

```
SELECT COUNT(*) FROM TUSER.ORDER_ITEMS_B WHERE ORDER_ID IN ( SELECT ORDER_ID FROM
TUSER.ORDERS_B WHERE CUSTOMER_ID=<customer_id>);
SELECT COUNT(*) FROM TUSER.ORDERS_B WHERE CUSTOMER_ID=<customer_id>;
SELECT COUNT(*) FROM TUSER.ADDRESSES_B WHERE CUSTOMER_ID=<customer_id>;
SELECT COUNT(*) FROM TUSER.CUSTOMERS_B WHERE CUSTOMER_ID=<customer_id>;
```

Clean up

20. Delete the subdirectories created in the shared folder.

```
host "rm -r /media/sf_extdisk/auxdest/ORADB";
```

21. Shutdown `srv1` and restore it to the snapshot "**Practice 13 Start**". Start `srv1`.

22. Delete "**Practice 13 Start**" snapshot.

Summary

In this practice, you implemented the following recovery procedures:

- Perform tablespace point-in-time recovery (TSPITR) on `soetbs` tablespace.
- Recover specific tables using RMAN `RECOVER` command to a specific point-in-time.

