

Practice

Performing Recovery Part I

Practice Target

In this practice you will perform specific recovery scenarios in the non-CDB database `ORADB` in `srv1`.

Practice Overview

In this practice, you will perform the following recovery scenarios:

- Complete recovery of the entire database in `NOARCHIVELOG` Mode
- Complete recovery of the entire database in `ARCHIVELOG` Mode
- Complete recovery of a user tablespace datafile loss

Assumptions

- This practice assumes that you have `srv1` up and running from the **non-CDB** snapshot.



Recovery Scenario 1:

Complete recovery of the entire database in NOARCHIVELOG Mode

Scenario assumptions:

- The database is running in NOARCHIVELOG mode
- One or more critical datafiles are lost

Preparing for the scenario

To prepare for this recovery scenario, you will first switch the database to run in NOARCHIVELOG mode then you will take a full cold backup of the database.

1. Invoke SQL*Plus and login to the database as SYS

```
sqlplus / as sysdba
```

2. Switch the database to run in NOARCHIVELOG mode.

```
# mount the database
SHUTDOWN IMMEDIATE
STARTUP MOUNT

# disable the archivelog mode
ALTER DATABASE NOARCHIVELOG;

# restart the database
ALTER DATABASE OPEN;

# exit from SQL*Plus
exit
```

3. Invoke RMAN with connecting to ORADB as target, mount the database and take a full backup of the database.

Because the database is running in NOARCHIVELOG mode, we can take a backup of the database only when it is in MOUNT state.

```
rman target /

SHUTDOWN IMMEDIATE
STARTUP MOUNT
BACKUP DATABASE TAG 'FULL_DB';
ALTER DATABASE OPEN;
```

4. Issue the following command to list all the database datafiles. Highlight the full path name of SYSTEM tablespace datafile (to copy it to the clipboard).

```
REPORT SCHEMA;
```

5. Exit from RMAN

```
exit
```

Simulating the Crash

In the following steps, you will simulate losing `SYSTEM` tablespace user datafiles.

6. Delete the `SYSTEM` tablespace datafile. This is a simulation to losing a database critical datafile.

```
rm /u01/app/oracle/oradata/ORADB/datafile/*_system_*.dbf
```

Recovery Actions

As a result of losing the critical datafile, the database instance might crash. In all cases, the application users will receive errors as a result of the destruction action. As a DBA, you are contacted to look into the issue.

7. Try logging on to the database.

You could or could not login to the database depending on the instance status.

```
sqlplus / as sysdba
```

8. If you could login to the database, issue the following statement to try querying some data dictionary views from the `SYSTEM` tablespace.

The statement should return errors like the following:

```
ERROR at line 1:
ORA-00604: error occurred at recursive SQL level 1
ORA-01116: error in opening database file 1
ORA-01110: data file 1: '/u01/app/oracle/oradata/ORADB/system01.dbf'
ORA-27041: unable to open file
Linux-x86_64 Error: 2: No such file or directory
```

```
SELECT * FROM DBA_OBJECTS ;
```

9. Exit from SQL*Plus

```
exit
```

10. As what we normally do in such cases, the first step is to look into the alertlog file. Open the alert log file and try searching for the issue error message.

vi editor tip: when you open the file the first time, the current location will be the first line of the file. To go to the last line in the file, press on [G] (Upper case 'G' letter).

```
vi /u01/app/oracle/diag/rdbms/oradb/oradb/trace/alert_oradb.log
```

You will observe a message like the following:

```
ORA-01565: error in identifying file '/u01/app/.._system_..dbf'
ORA-27037: unable to obtain file status
```

You reported the issue to the management and decided to restore the database. They understand that the data entered since last time the backup was taken will be lost.

11. Abort the database instance (if it is up and running) and start it up in `MOUNT` state.

We cannot cleanly shutdown the database in such scenarios because the instance is unable to access a critical tablespace.

```
sqlplus / as sysdba

SHUTDOWN ABORT;
STARTUP MOUNT;

exit
```

12. Invoke RMAN, login as target to `ORADB` and then restore it.

Observe that RMAN rebuilds all the datafiles. Including that one that are already there.

Of course, we can restore the lost datafile only. For this example, we want to experience restoring the entire database.

We must open the database in `RESETLOGS` in this scenario.

```
rman target /
RESTORE DATABASE;
RECOVER DATABASE;
ALTER DATABASE OPEN RESETLOGS;
```

13. Verify that the data dictionary is accessible.

```
SELECT COUNT(*) FROM DBA_TABLES;
```

14. (optional) Retrieve the incarnation history.

The `INCARNATION#` is incremented every time we open the database using `RESETLOGS` mode.

```
SELECT INCARNATION#, RESETLOGS_CHANGE#, PRIOR_RESETLOGS_CHANGE#, STATUS
FROM V$DATABASE_INCARNATION
ORDER BY 1;
```

15. As a cleanup, delete the backup files.

```
DELETE BACKUPSET;
```

16. Exit from RMAN.

```
quit
```

Note: In real life scenarios, after restoring a database, it is highly recommended to take a full backup of the entire database.

Recovery Scenario 2:

Complete recovery of the entire database in ARCHIVELOG Mode

Scenario assumptions:

- The database is running in ARCHIVELOG mode
- All or most datafiles are lost

Preparing for the scenario

To prepare for this recovery scenario, take a full online backup of the database.

17. Perform the steps below to enable the ARCHIVELOG mode in ORADB database.

```
sqlplus / as sysdba

SHUTDOWN IMMEDIATE
STARTUP MOUNT
ALTER SYSTEM SET LOG_ARCHIVE_DEST_1='LOCATION=USE_DB_RECOVERY_FILE_DEST' SCOPE=SPFILE;
ALTER DATABASE ARCHIVELOG;
ALTER DATABASE OPEN;
ALTER SYSTEM SWITCH LOGFILE;
SELECT NAME FROM V$ARCHIVED_LOG;
```

18. Exit from SQL*Plus

```
exit
```

19. Invoke RMAN and connect to oradb as target then take full backup of the database.

```
rman target /

BACKUP DATABASE TAG 'FULL_DB';
```

Simulating the Crash

20. As a simulation to losing a critical database file, delete the `SYSTEM` tablespace datafile.

```
rm /u01/app/oracle/oradata/ORADB/datafile/*_system_*.dbf
```

Recovery Actions

21. Try logging on to the database using SQL*Plus. It may or may not succeed.

```
sqlplus / as sysdba
```

22. If you could login to the database, issue the following statement to try querying some data dictionary views from the `SYSTEM` tablespace. The statement must return errors.

```
SELECT * FROM DBA_OBJECTS ;
```

23. Mount the database then exit from SQL*Plus.

```
SHUTDOWN ABORT  
STARTUP MOUNT  
  
EXIT
```

24. Invoke RMAN, login as target to `ORADB`.

```
rman target /
```

25. Before restoring the database, you want to make sure that the backup files are valid. Bear in mind that this command may take long time, if it is run against large databases.

```
RESTORE DATABASE VALIDATE;
```

26. Restore the database, recover it, then open it in read/write mode.

Observe that we do not open the database in `RESETLOGS` option because it is running in `ARCHIVELOG` mode.

```
RESTORE DATABASE;  
RECOVER DATABASE;  
ALTER DATABASE OPEN;
```

27. (optional) Retrieve the incarnation history.

The incarnation is still the same.

```
SELECT INCARNATION#, RESETLOGS_CHANGE#, PRIOR_RESETLOGS_CHANGE#, STATUS  
FROM V$DATABASE_INCARNATION  
ORDER BY 1;
```

Recovery Scenario 3:

Complete recovery of a user tablespace loss

Scenario assumptions:

- The database is running in `ARCHIVELOG` mode
- One or some datafiles of a user tablespace are lost

Preparing for the scenario

To prepare for this recovery scenario, take a full online backup of the tablespace.

28. Take full backup of the users tablespace.

```
BACKUP TABLESPACE users TAG 'FULL_USERS';
```

Simulating the Crash

29. As a simulation to losing a user tablespace datafile, delete the `USERS` tablespace datafile.

```
host 'rm /u01/app/oracle/oradata/ORADB/datafile/*_users*.dbf';
```

Recovery Actions

30. In RMAN session, to figure out which datafile is lost, issue the following command.

You should receive an error like the following:

```
ORA-01122: database file 4 failed verification check
```

```
VALIDATE TABLESPACE USERS;
```

31. Issue the following commands on the tablespace to recover it.

Observe that you could recover the tablespace while the database is still up and running. This is not possible in `NOARCHIVELOG` mode.

```
ALTER TABLESPACE users OFFLINE IMMEDIATE;  
RESTORE TABLESPACE users;  
RECOVER TABLESPACE users;  
ALTER TABLESPACE users ONLINE;
```

Clean up

32. Delete the backup files.

```
DELETE BACKUPSET;
```

33. Shutdown `srv1` and restore it from its **non-CDB** snapshot.

Summary

In this practice, you performed the full recovery procedure to the following scenarios:

- Complete recovery of the entire database in `NOARCHIVELOG` Mode
- Complete recovery of the entire database in `ARCHIVELOG` Mode
- Complete recovery of a user tablespace datafile loss

