**Create Recovery Catalog:**

A Recovery Catalog in Oracle's Recovery Manager (RMAN) acts as a central repository for crucial metadata about your database backups. It essentially creates a safety net for your backup information.

**Benefits of a Recovery Catalog:**

**Redundancy:**

It provides a secondary storage location for RMAN metadata, separate from the control file of each target database. This ensures that even if the control file and backups are lost, the recovery catalog might still hold the information needed to recover your database.

**Centralization:**

It centralizes metadata for all your target databases, simplifying tasks like reporting and administration. Imagine having backup details for all your databases in one place, rather than scattered across individual control files.

**Extended Backup History:**

The recovery catalog can store backup history for a much longer duration compared to the control file. This allows you to potentially recover from incidents that happened further back in time.

**Recovery Catalog Functionality:**

The recovery catalog stores various types of metadata related to RMAN operations, including:

* Backup Information: Details about data file and archived redo log backup sets and pieces.
* Data File Copies: Tracks information on data file copies.
* Archived Redo Logs: Maintains metadata on archived redo logs and their copies.
* Database Structure: Stores information about tablespaces and datafiles within the database.

RMAN Scripts and Configurations: Allows storing named user-created RMAN command sequences (scripts) and persistent RMAN configuration settings.

In essence, the recovery catalog acts as a vital insurance policy for your RMAN backups. It ensures that even in case of unforeseen circumstances, you have a high chance of retrieving your database to a functional state.

**LAB:**

sqlplus / as sysdba

select name, open\_mode from v$pdbs;

alter pluggable database orclpdb open;

alter session set container = orclpdb;

show con\_name

alter pluggable database orclpdb save state;

**---DISPLAY TABLESPACES**

select tablespace\_name from dba\_tablespaces;

**---------CREATE TABLESPACE--------------**

create tablespace tbs\_data

datafile '/u01/app/oracle/oradata/tbs\_data01.dbf' size 10m

autoextend on next 1m maxsize 100m;

alter tablespace tbs\_data

add datafile '/u01/app/oracle/oradata/tbs\_data02.dbf' size 10m

autoextend on next 1m maxsize 100m;

create tablespace tbs\_indx

datafile '/u01/app/oracle/oradata/tbs\_indx01.dbf' size 10m

autoextend on next 1m maxsize 100m;

**---DISPLAY USERS**

SELECT USERNAME FROM ALL\_USERS;

**-------CREATE USER--------------**

create user mytest

identified by rkp23

default tablespace tbs\_data

temporary tablespace temp;

grant connect, resource, dba to mytest;

grant unlimited tablespace to mytest;

**---ADD AN ENTRY TO TNSNAMES FOR ORCLPDB**

vi tnsnames.ora

ORCLPDB =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = orclpdb)

)

)

sqlplus mytest/rkp23@orclpdb;

**-----CREATE TABLE**

CREATE TABLE test\_tb1(id number, name varchar2(30), surname varchar2(30), birth\_date date) tablespace tbs\_data;

grant all on test\_tb1 to public;

INSERT INTO test\_tb1 (id, name, surname, birth\_date)

VALUES (1, 'John', 'Doe', TO\_DATE('1990-05-15', 'YYYY-MM-DD'));

INSERT INTO test\_tb1 (id, name, surname, birth\_date)

VALUES (2, 'Jane', 'Smith', TO\_DATE('1985-08-20', 'YYYY-MM-DD'));

INSERT INTO test\_tb1 (id, name, surname, birth\_date)

VALUES (3, 'Michael', 'Johnson', TO\_DATE('1978-12-10', 'YYYY-MM-DD'));

commit;

**----STARTING AND INTERACTING WITH RMAN CLIENT**

lsnrctl status;

lsnrctl start;

rman

**--connect to the db**

connect target/

**--connect to the db when running rman prompt**

rman target /

**--to display the configuration**

show all;

**--Configure Network**

enable network adapters on both machine

Adapter 1 to: Intrernal n/w

Name:pubnet

Adapter 2 to: Intrernal n/w

Name:privnet

Adapter1 to: NAT

**Restart both vms after this**

vi /etc/hosts

192.168.56.101 rmancat.localdomain rmancat

192.168.56.100 oracle linux.localdomain oracle linux

**--Check connection**

ping rmancat

from oracle linux machine

ping primary

from rmancat machine

**--Add to tnsnames.ora**

vi $ORACLE\_HOME/network/admin/tnsnames.ora

rmancat =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = rmancat.localhost)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = rmancat)

)

)

orclpdb =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = orclpdb.localhost)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = orclpdb)

)

)

lsnrctl start

sqlplus / as sysdba

startup

**------------CREATE RMAN CATALOG DATABASE-----------**

**--ON RMANCAT**

select name,open\_mode from v$database;

show pdbs

alter pluggable database orclpdb open;

alter pluggable database orclpdb save state;

**----Change session to pdb database**

alter session set container = orclpdb;

show con\_name

show parameter db\_create\_file\_dest

ALTER SYSTEM SET db\_create\_file\_dest = '/u01/app/oracle/oradata';

**----Create tablespace to store backup information of registered databases**

create tablespace tbs\_catalog

datafile '/u01/app/oracle/oradata/catalog01.dbf' size 100m

autoextend on next 10m maxsize 1024m;

**---Create user who will work as a recovery catalog owner**

create user rcatown

identified by rkp23

default tablespace tbs\_catalog

temporary tablespace temp

quota unlimited on tbs\_catalog;

grant connect, resource, recovery\_catalog\_owner TO rcatown;

**---CONNECT RMAN TO orclpdb DATABASE WITH CATALOG CLAUSE**

rman catalog rcatown/rkp23@orclpdb;

**--Apply to create catalog database command**

RMAN> create catalog

RMAN> exit

**--CONNECT RMAN WITH TARGET AND CATALOG CLAUSE**

rman target sys/oracle@primary catalog rcatown@orclpdb

**--REGISTER TARGET DATABASE**

RMAN> register database;

RMAN> report schema;

**Flashback database and Restore Points:**

Flashback Database and Restore Points are powerful features in Oracle that allow you to rewind your database to a previous state, essentially performing a point-in-time recovery. Here's a breakdown of both:

**Flashback Database**

* **Function**:

Let’s us revert the entire database to a specific point in time.

* **Benefits**:
* Faster than traditional point-in-time recovery as it doesn't require restoring data files from backups.
* Ideal for rectifying logical errors or data corruption.
* **Process**:
* Requires a configured Flash Recovery Area (FRA) to store historical database block images.
* Flashback Database utilizes these images to reconstruct the database state at the desired time.

**Restore Points**

* **Function:**

Act as named references to specific points in time within the Flashback Database timeline.

* **Benefits:**

Simplifies specifying the target time for a Flashback Database operation.

Two types:

**i. Regular Restore Point:**

A reference to a System Change Number (SCN), a unique identifier for a database change.

**ii. Guaranteed Restore Point:**

Ensures the database retains flashback logs up to that point, enabling rollback even if further changes occur.

**LAB:**

Using flashback query

sqlplus mytest/rkp23@orclpdb;

col name format a10

col surname format a10

select \* from test\_tb1;

update test\_tb1 set name='Rohit' where id=1;

commit;

alter session set nls\_date\_format='dd.mm.yyyy hh24:mi:ss';

**--Finding old version of updated value**

select \* from (select name from test\_tb1 where id=1) as of timestamp sysdate-1/24;

update test\_tb1 set name='John' where id=1;

commit;

**---Control archievelog mode**

archieve log list

select log\_mode from v$database;

**---Open Archivelog mode**

shutdown immediate;

startup mount;

alter database archivelog;

alter database open;

**---Enable force logging and flashback**

alter database force logging;

ALTER SYSTEM SET DB\_RECOVERY\_FILE\_DEST\_SIZE=5g scope=both;

ALTER SYSTEM SET DB\_RECOVERY\_FILE\_DEST='/u01/app/oracle/fast\_recovery\_area' SCOPE=BOTH;

alter database flashback on;

select flashback\_on from v$database;

show parameter db\_flashback\_retention\_target

**-----OPTIONS------**

--set the length of desired flashback window in minutes

alter system set db\_flashback\_retention+target=4320 scope=both;

--we can enable or disable flashback logging for a tablespace

alter tablespace tbs\_data flashback on;

alter tablespace tbs\_data flashback off;

**--Disabling flashback database logging**

alter database flashback off;

**---FLASHBACK TABLE**

sqlplus mytest/rkp23@orclpdb

drop table test\_tb1;

select object\_name, original\_name, operation, droptime from user\_recyclebin;

flashback table test\_tb1 to before drop;

**---FLASHBACK DATABASE**

**--Connect container database**

sqlplus / as sysdba

select current\_scn from v$database;

CURRENT\_SCN

-----------

2597256

**--Connect pdb**

sqlplus mytest/rkp23@orclpdb

drop table test\_tb1 purge;

**--Connect container database**

sqlplus / as sysdba

shutdown immediate;

startup mount;

flashback database to scn 2597256;

alter database open resetlogs;

**--Check the dropped table**

sqlplus mytest/rkp23@orclpdb

**========RESTORE POINTS========**

**--CREATING PDB RESTORE POINTS**

sqlplus / as sysdba

**--Display the state of PDB**

select name,open\_mode from v$pdbs;

**--Close the pdb**

alter pluggable database orclpdb close;

alter session set container=orclpdb;

**--Create a normal pdb restore point**

create restore point rp\_orclpdb;

**--Open pdb**

alter pluggable database orclpdb open;

**--Connect with test user**

sqlplus mytest/rkp23@orclpdb

drop table test\_tb1;

**--Flashback to restore point**

sqlplus sys/rkp23@orclpdb as sysdba

alter pluggable database orclpdb close;

flashback pluggable database to restore point rp\_orclpdb;

alter pluggable database orclpdb open resetlogs;

**--Run with test user**

conn mytest/rkp23@orclpdb

col name format a10

col surname format a10

select \* from test\_tb1;

**-------TO CREATE A PDB RESTORE POINT WHEN CONNECTED TO THE CDB----**

sqlplus / as sysdba

**--close orclpdb**

alter pluggable database orclpdb close;

**--Place cdb in mounted state**

shutdown immediate

startup mount;

**--Check the container we are copnnected to**

show con\_name

**--Select the current container to the root if it is not already**

alter session set container = CDB$ROOT;

**--Create a normal pdb restore point**

create restore point rp\_pdb\_from\_cdb for pluggable database orclpdb;

create restore point grp\_pdb\_from\_cdb for pluggable database orclpdb guarantee flashback database;

**--Create clean PDB restore point \*when pdb is closed and has no pending transactions)**

create clean restore point crp\_pdb\_from\_cdb for pluggable database orclpdb;

**--Creating CDB restore points**

sqlplus / as sysdba

create restore point rp\_cdb;

**--Open the cdb**

alter database open;

**--To create gauranteed restore point**

create restore point grp\_cdb guarantee flashback database;

**--Connect with test user**

sqlplus mytest/rkp23@orclpdb

drop table test\_tb1;

**--Connect cdb**

sqlplus / as sysdba

shutdown immediate

startup mount;

**--Flashback to restore point**

flashback database to restore point grp\_cdb;

alter database open resetlogs;

**--Connect to orclpdb with test user**

sqlplus mytest/rkp23@orclpdb

select \* from test\_tb1;

**--LISTING THE RESTORE POINTS**

rman target /

RMAN> list restore point all;

using target database control file instead of recovery catalog

SCN RSP Time Type Time Name

---------------- ------------------- ---------- ------------------- ----

2609797 2024-03-27:15:18:34 RP\_CDB

2611021 GUARANTEED 2024-03-27:15:21:12 GRP\_CDB

2599219 2024-03-27:14:00:26 RP\_ORCLPDB

2609797 2024-03-27:15:12:47 RP\_PDB\_FROM\_CDB

2609797 GUARANTEED 2024-03-27:15:14:47 GRP\_PDB\_FROM\_CDB

2609797 2024-03-27:15:16:54 CRP\_PDB\_FROM\_CDB

RMAN> list restore point rp\_orclpdb;

SCN RSP Time Type Time Name

---------------- ------------------- ---------- ------------------- ----

2599219 2024-03-27:14:00:26 RP\_ORCLPDB

**--Displaying restore points**

sqlplus / as sysdba

col name format a30

select name, guarantee\_flashback\_database, pdb\_restore\_point, clean\_pdb\_restore\_point,

pdb\_incarnation#, storage\_size from v$restore\_point;

**--DROP RESTORE POINTS**

drop restore point grp\_cdb;

drop restore point GRP\_PDB\_FROM\_CDB;

alter session set container=orclpdb;

drop restore point RP\_ORCLPDB;

**Backup Database (with RMAN)**

RMAN (Recovery Manager) is a powerful tool included with Oracle Database that allows you to perform backups and recoveries of your database. It offers various advantages over traditional methods, including:

* **Online Backups:**

RMAN can back up your database while it's still running, minimizing downtime and impact on users.

* **Different Backup Types:**

RMAN supports various backup options like full backups, which capture the entire database, and incremental backups, which focus on changes since the last full backup.

* **Archive Log Backups:**

RMAN can also back up archive logs, crucial for point-in-time recovery to a specific moment.

* **Automation:**

RMAN allows scripting and scheduling backups, ensuring a consistent and reliable backup strategy.

**LAB:**

**---BACKING UP THE DATABASE IN NOARCHIVELOG MODE**

**--Control archivelog mode**

sqlplus / as sysdba

archive log list

select log\_mode from v$database;

**--Close archivelog mode**

shutdown immediate;

startup mount;

alter database flashback off;

select name, guarantee\_flashback\_database, pdb\_restore\_point, clean\_pdb\_restore\_point,

pdb\_incarnation#, storage\_size from v$restore\_point;

drop restore point grp\_cdb;

alter database noarchivelog;

alter database open;

archive log list;

select log\_mode from v$database;

**--Shutdown the database before backing up**

shutdown immediate;

startup mount;

**--Start backup with RMAN**

rman target /

backup database tag 'backup\_norachivelog';

**--Open database**

alter database open;

**--Open archivelog mode**

sqlplus / as sysdba

select open\_mode from v$database;

**--Change log\_mode to archive log back**

shutdown immediate;

startup mount;

alter database archivelog;

alter database flashback on;

alter database open;

archive log list;

select log\_mode, flashback\_on from v$database;

**--Backup with catalog database**

**--Connect on primary server**

rman target sys/rkp23@primary catalog rcatown/rkp23@orclpdb;

show all;

**--Connect on rmancat server**

rman target sys/rkp23@primary catalog rcatown/rkp23@orclpdb;

show all;

backup database plus archivelog;

**--Specifying the device type for an RMAN backup**

show device type;

backup device type disk database;

**Reporting on RMAN operations:**

RMAN (Recovery Manager) offers several ways to generate reports on backup activities. These reports rely on the metadata stored in the RMAN repository, which can reside in the control file of the target database or a separate recovery catalog database.

**1. Using the LIST Command:**

* Provides detailed information on backups and copies.
* Common options include:
* SUMMARY: Generates a concise summary of all backups and copies.
* BY FILE: Lists backups categorized by the specific files backed up.
* VERBOSE (default): Offers extensive details on each backup or copy.

**2. Using the REPORT Command:**

* Performs more in-depth analysis compared to LIST.
* Key functionalities include:
* NEED BACKUP: Identifies files requiring backup based on the current retention policy.
* OBSOLETE (after running CROSSCHECK): Reports backups no longer necessary for recovery based on retention policies.

**3. CROSSCHECK Command:**

* Validates the status of backups listed in the repository against their physical existence on disk or tape.
* Use CROSSCHECK before running REPORT OBSOLETE to ensure accuracy.

**LAB:**

**BACKUP:**

Initiates the backup of one or more database files or backup sets.

BACKUP DATABASE;

BACKUP TABLESPACE users;

BACKUP ARCHIVELOG ALL;

**RESTORE:**

Specifies the files to be restored from backup sets, image copies, or archived redo logs.

RESTORE DATABASE;

RESTORE TABLESPACE users;

**RECOVER:**

Initiates the recovery of restored files, applying redo logs to bring them to a consistent state.

RECOVER DATABASE;

RECOVER TABLESPACE users;

**LIST:**

Displays details about backups, copies, or archived logs recorded in the RMAN repository.

LIST BACKUP;

LIST COPY;

LIST ARCHIVELOG ALL;

**DELETE:**

Removes backups, copies, or archived logs from disk or tape based on specified criteria.

DELETE BACKUP;

DELETE COPY;

DELETE ARCHIVELOG UNTIL TIME 'SYSDATE-7';

**CROSSCHECK:**

Verifies the existence of backups and copies recorded in the repository and updates their status.

CROSSCHECK BACKUP;

CROSSCHECK COPY;

**REPORT:**

Provides summary and detailed information about backups, copies, and archived logs.

REPORT SCHEMA;

REPORT NEED BACKUP;

REPORT OBSOLETE;

**CONFIGURE:**

Sets persistent configuration parameters for RMAN operations.

CONFIGURE BACKUP OPTIMIZATION ON;

CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;

**SHOW:**

Displays the current RMAN configuration settings.

SHOW ALL;

SHOW RETENTION POLICY;

**Maintaining RMAN Backups and Repository records**