# **Part 2 - Backup and Recovery – Checklist 8-12**

## **Create Recovery Table – Checklist 8**

SQL>

SQL> SET SERVEROUTPUT ON;

SQL>

SQL> /\*---------------------------------------------------\*/

SQL> /\* DROP OBJECTS \*/

SQL> /\*---------------------------------------------------\*/

SQL>

SQL> SHOW USER;

USER is "DATADESIGNLEADUSER"

SQL>

SQL> CONNECT DataDesignLeadUser/abc123;

Connected.

SQL>

SQL> SHOW USER;

USER is "DATADESIGNLEADUSER"

SQL>

SQL> /\*Drop Table\*/

SQL>

SQL> DROP TABLE product\_cat\_backup;

Table PRODUCT\_CAT\_BACKUP dropped.

SQL>

SQL>

SQL> /\*-------------------------------------------------------------------\*/

SQL> /\*A script to create a small table with 3 rows for recovery testing.\*/

SQL> /\*-----------------------------------------------------------------\*/

SQL>

SQL> /\*Create Table\*/

SQL>

SQL> CREATE TABLE product\_cat\_backup (

2 prodcatid NUMBER(10) NOT NULL,

3 prodcattype VARCHAR(25),

4 description VARCHAR(150),

5 CONSTRAINT pk\_prodcat\_backup PRIMARY KEY ( prodcatid )

6 USING INDEX TABLESPACE user\_indx1

7 STORAGE ( INITIAL 50 K NEXT 10 K )

8 )

9 PCTFREE 5 PCTUSED 90 TABLESPACE user\_data1

10 STORAGE ( INITIAL 192 k NEXT 20 k MAXEXTENTS UNLIMITED PCTINCREASE 0 );

Table PRODUCT\_CAT\_BACKUP created.

SQL>

SQL> /\*Insert 3 rows of data from original product category table\*/

SQL>

SQL> INSERT INTO product\_cat\_backup SELECT \* FROM product\_cat WHERE PRODCATID <= 3;

3 rows inserted.

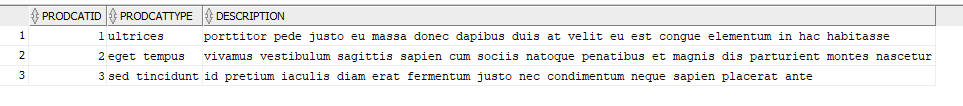
SQL>

SQL> commit;

Commit complete.

SQL> /\*Verify data is there\*/

SQL> SELECT \* FROM product\_cat\_backup;



Connection created by CONNECT script command disconnected

## **Export/Import – Checklist 9 & 10**

First run the export command: “exp owner=DataDesignLeadUser file=DataDesignLeadUser.exp statistics=none*,”* see figure 1.



Figure 1. Example of exporting data from the database.

Return to the script and drop the table, and then confirm it is dropped:

SQL> /\*------------------------------------------------------------------------------------------\*/

SQL> /\*Backup and recovery commands for Recovery testing using export/import and RMAN utilities.\*/

SQL> /\*----------------------------------------------------------------------------------------\*/

SQL>

SQL> /\*Run exp command from Unix command line then return to drop table\*/

SQL> DROP TABLE product\_cat\_backup;

Table PRODUCT\_CAT\_BACKUP dropped.

SQL> SELECT \* FROM product\_cat\_backup;

Error starting at line : 53 in command -

SELECT \* FROM product\_cat\_backup

Error at Command Line : 53 Column : 15

Error report -

SQL Error: ORA-00942: table or view does not exist

00942. 00000 - "table or view does not exist"

\*Cause:

\*Action:

SQL>

SQL> /\*Drop other objects in schema for cleaner import output\*/

SQL> /\*Drop Indexes\*/

SQL>

SQL> DROP INDEX fk\_zip;

Index FK\_ZIP dropped.

SQL>

SQL> DROP INDEX fk\_customerid;

Index FK\_CUSTOMERID dropped.

SQL>

SQL> DROP INDEX fk\_order\_id1;

Index FK\_ORDER\_ID1 dropped.

SQL>

SQL> DROP INDEX fk\_product\_id;

Index FK\_PRODUCT\_ID dropped.

SQL>

SQL> DROP INDEX fk\_order\_id2;

Index FK\_ORDER\_ID2 dropped.

SQL>

SQL> DROP INDEX fk\_emp\_storeid;

Index FK\_EMP\_STOREID dropped.

SQL>

SQL> DROP INDEX fk\_emp\_zip;

Index FK\_EMP\_ZIP dropped.

SQL>

SQL> DROP INDEX fk\_store\_region;

Index FK\_STORE\_REGION dropped.

SQL>

SQL> DROP INDEX fk\_store\_zip;

Index FK\_STORE\_ZIP dropped.

SQL>

SQL> DROP INDEX fk\_prodcatid;

Index FK\_PRODCATID dropped.

SQL>

SQL> DROP INDEX fk\_emp\_ssn;

Index FK\_EMP\_SSN dropped.

SQL>

SQL> /\*Drop Tables\*/

SQL>

SQL> DROP TABLE payment;

Table PAYMENT dropped.

SQL>

SQL> DROP TABLE order\_item;

Table ORDER\_ITEM dropped.

SQL>

SQL> DROP TABLE orders;

Table ORDERS dropped.

SQL>

SQL> DROP TABLE product;

Table PRODUCT dropped.

SQL>

SQL> DROP TABLE customer;

Table CUSTOMER dropped.

SQL>

SQL> DROP TABLE employee;

Table EMPLOYEE dropped.

SQL>

SQL> DROP TABLE stores;

Table STORES dropped.

SQL>

SQL> DROP TABLE zip;

Table ZIP dropped.

SQL>

SQL> DROP TABLE region;

Table REGION dropped.

SQL>

SQL> DROP TABLE product\_cat;

Table PRODUCT\_CAT dropped.

Now, we can run the import command: “imp file=DataDesignLeadUser.exp fromuser=DataDesignLeadUser touser=DataDesignLeadUser,” see figure 2.

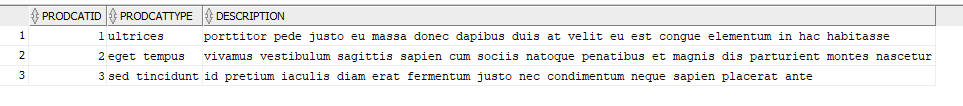


Figure 2. Example of importing data to the database.

From the SQL Script verify the table has returned:

SQL> /\*Run imp command from Unix command line then return to view restored table\*/

SQL> SELECT \* FROM product\_cat\_backup;



## **RMAN – Checklist 9 & 10**

From the command line, we can use RMAN to back up the database, see figure 3.

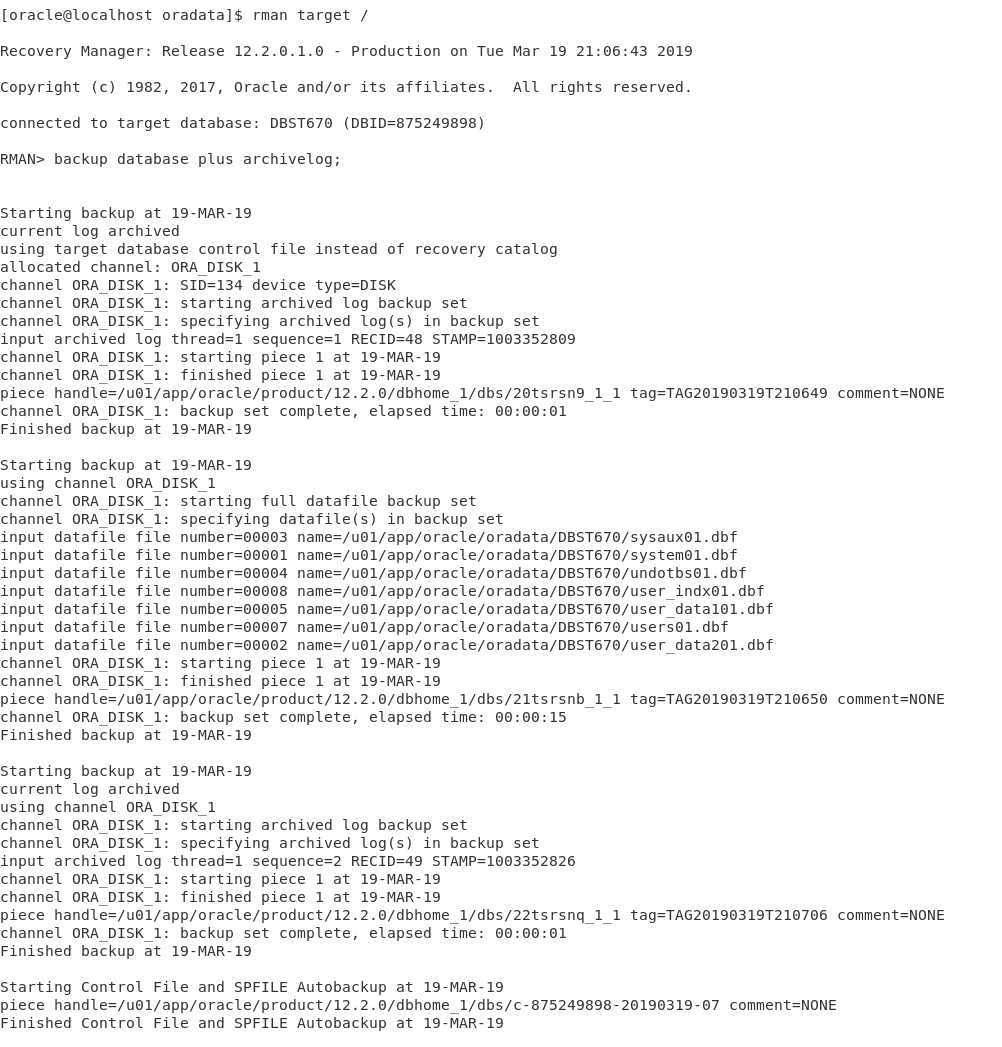


Figure 3. Creating a database back up with RMAN.

We can then list our backups to see what was accomplished, see figure 4.

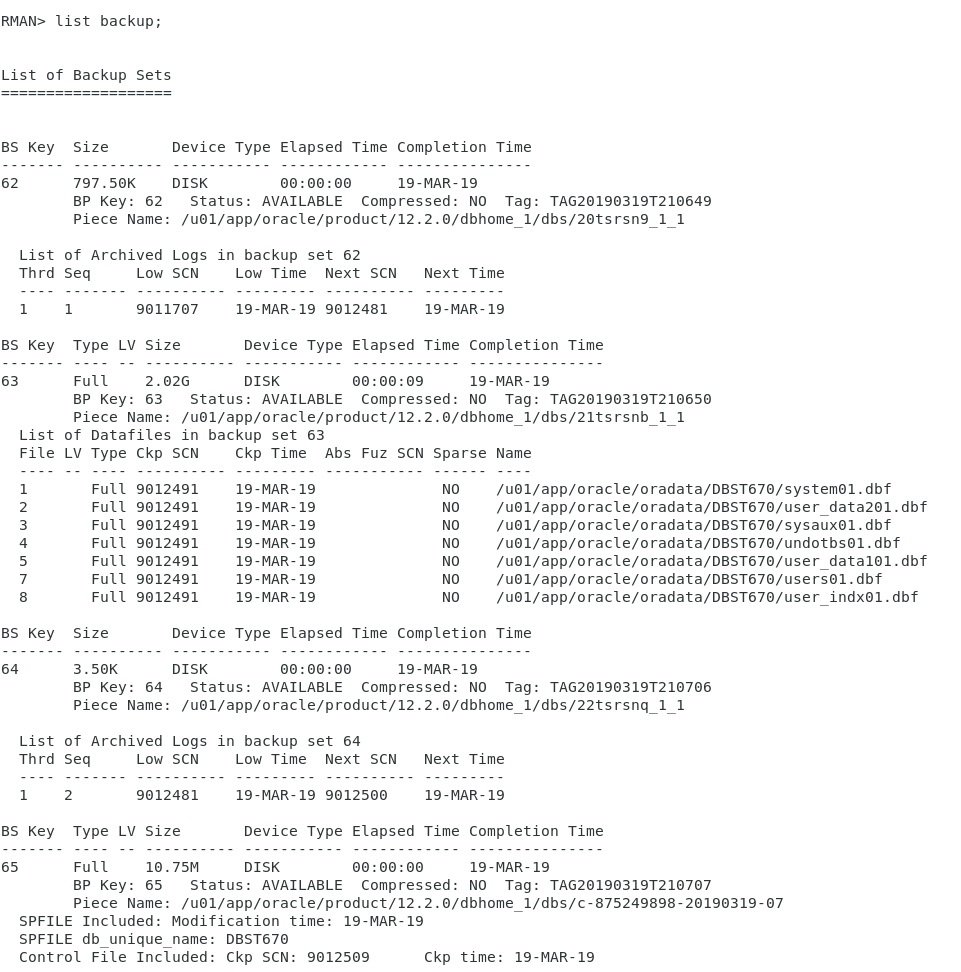


Figure 4. List the existing backups.

We drop the recovery table and check the scn number to ensure it moved past the backup scn number we made before.

SQL> /\*Connect to RMAN from Unix Command Line. Create backup. Return and drop table\*/

SQL> DROP TABLE product\_cat\_backup;

Table PRODUCT\_CAT\_BACKUP dropped.

SQL> /\*Check the current scn, it should be greater than the backup scn we made at this point\*/

SQL> SELECT current\_scn FROM v$database; /\*Run as system\*/



To begin the recovery process, we start the database in a force mount condition, restore the database, recover the database up to the scn number of our original backup, and then open the database with reset logs, see figure 5.

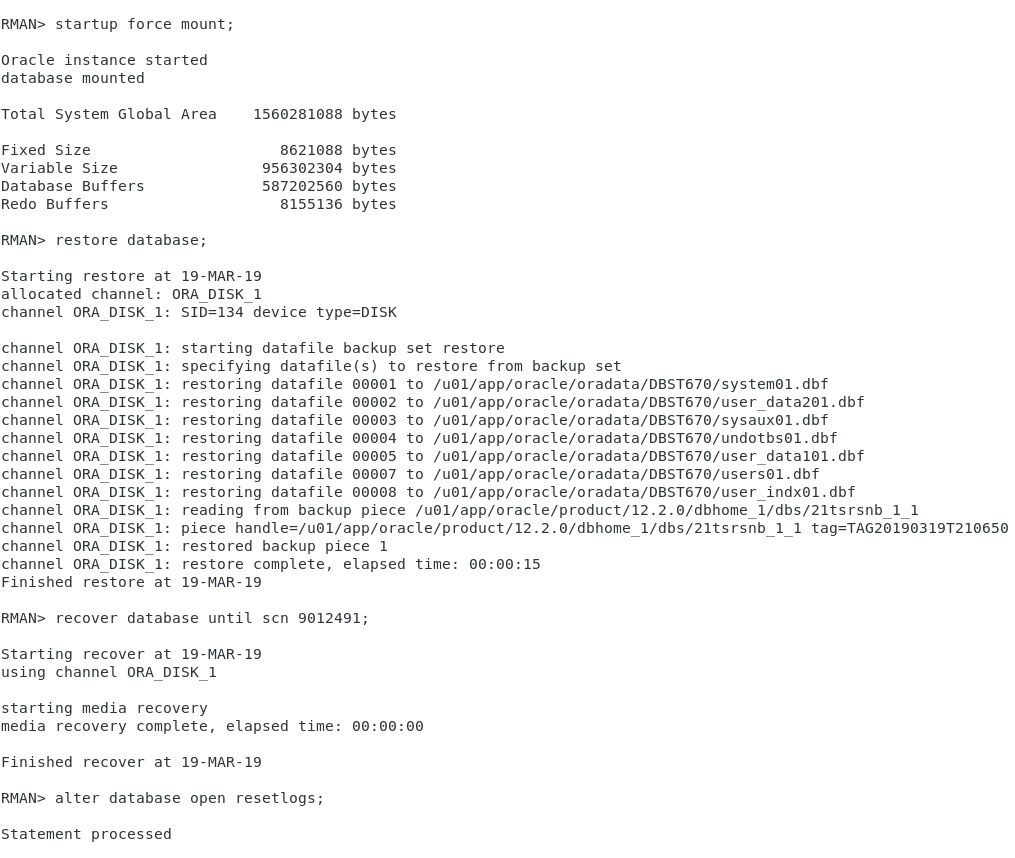
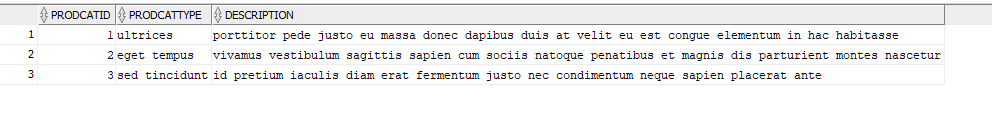


Figure 5. Run a database restore to a specific SCN number in RMAN.

From the SQL script, verify the table and data has returned.

SQL> /\*Restore to point in time and check Status with the select statement.\*/

SQL> SELECT \* FROM product\_cat\_backup;



## **RMAN Versus Import/Export – Checklist 11**

There are distinct differences between how RMAN works and the Import and Export Commands work. The best way to differentiate between the two is to consider RMAN a physical backup and Import/Export a logical backup. When we run an RMAN backup we are creating a complete backup of all the database physical components, i.e. data and control files. This method provides many benefits, such as providing either consistent or inconsistent backup options, backups of the archived redo logs, and automatic back up features. With this method, we can do a complete or incremental backup and restores from a point-in-time based off actual time or a System Change Number (SCN). If there is a complete database failure, this would be the preferred backup solution as it has everything you would need to fully restore your data (15 Backup and Recovery, 2019).

The Import/Export functions are considered a logical backup. When we use the export command, we are taking a copy at that time of specific database objects, such as tables and indexes, and placing them in a binary-formatted dump file. This is a one time issued command and does not provide the automated incremental backup features seen with RMAN. Additionally, while we may have a copy of critical table data at this point, we do not have a complete back up of the database. This form of back up is typically considered supplemental to another. If we relied on exported data files, we would have to rebuild the database manually and then import the data (20 Original Export and Import, 2019).

## **Flashback – Checklist 12**

For this project, we will use restore points to flashback the database to. First, we connect to RMAN and create a restore point, see figure 6.

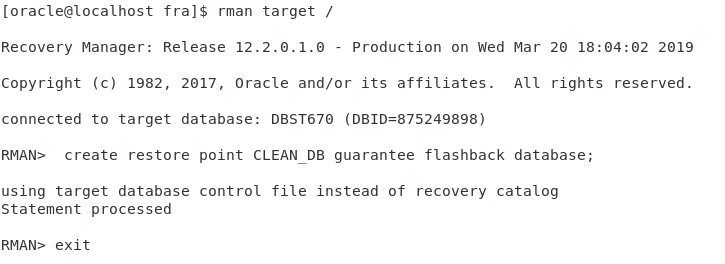


Figure 6. Create a restore point in RMAN.

Then, verify the restore point was created.

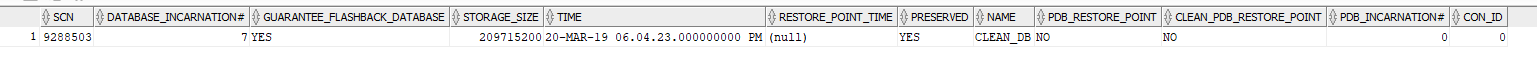
SQL> /\*-----------------------------\*/

SQL> /\*Flashback recovery commands.\*/

SQL> /\*---------------------------\*/

SQL> /\*Create restore point in RMAN, then view it with this select statement\*/

SQL> SELECT \* FROM v$restore\_point;



Drop the table:

SQL> /\*Drop table\*/

SQL> DROP TABLE product\_cat\_backup;

Table PRODUCT\_CAT\_BACKUP dropped.

We can now run the restoration steps, see figure 7.

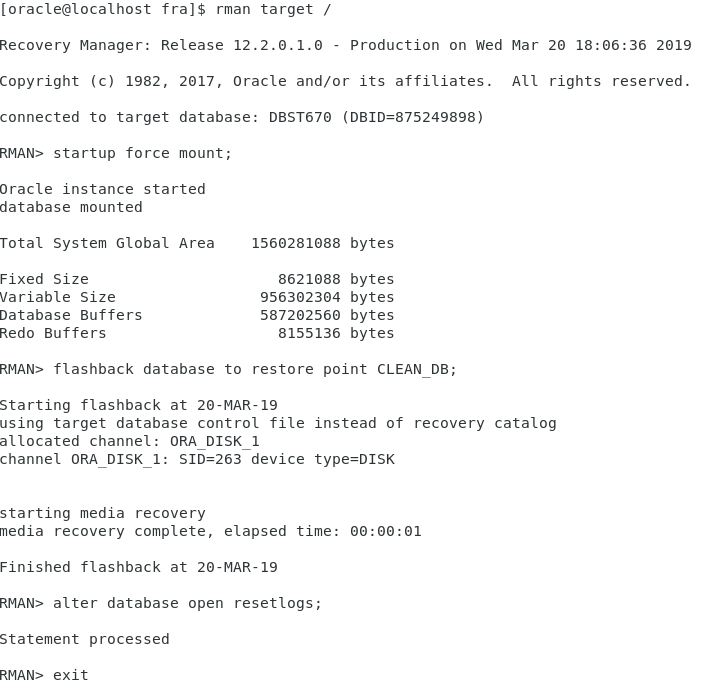


Figure 7. Restoring database to restore point with flashback in RMAN.

Now, we run our select statement again to verify the dropped table has returned:

SQL> /\*Use flashback to restore point then verify table has returned\*/

SQL> SELECT \* FROM product\_cat\_backup;

