



# XENIALAB<sup>TM</sup>

WEB AND MOBILE COMMUNICATIONS



## Abstract

- PDB PITR
- Flashback PDB
- PDB Hot Cloning e PDB Refresh

Come cambiano i “ribaltamenti” con la 12cR2

**Autore:** Umberto Signori

**Relatore:** Umberto Signori





## Pluggable Database (PDB) Point In Time Recovery (PITR)

Permette di recuperare un singolo PDB ad un SCN senza dare disservizio agli altri CDB/PDB

### Prerequisiti

- Modalità ARCHIVELOG
- Backup full di tutti i datafile del PDB terminato prima del target SCN
- Backup degli archivelog generati fra il backup full ed il target SCN

```
$ rman target /  
ALTER PLUGGABLE DATABASE XENIA01 CLOSE;  
RUN {  
    SET UNTIL TIME "sysdate-1/48";  
    RESTORE PLUGGABLE DATABASE XENIA01;  
    RECOVER PLUGGABLE DATABASE XENIA01 AUXILIARY  
    DESTINATION='/home/oracle/webinar/AUX';  
}  
ALTER PLUGGABLE DATABASE XENIA01 OPEN RESETLOGS;
```

Alla fine il PDB avrà una nuova incarnazione, si vede consultando la vista V\$PDB\_INCARNATION





## Flashback Pluggable Database

Annulla le modifiche fatte al database successive ad un determinato istante temporale – restore point

### *Preparazione*

Occorre abilitare il flashback ed eventualmente modificare il parametro

DB\_FLASHBACK\_RETENTION\_TARGET il cui valore rappresenta di quanti minuti si può tornare indietro

```
SQL> show parameter DB_FLASHBACK_RETENTION_TARGET
NAME                                TYPE          VALUE
-----                                -
db_flashback_retention_target      integer       1440

SQL> alter database flashback on;
Database altered.

SQL> select flashback_on from v$database;
FLASHBACK_ON
-----
YES
```





Il Flashback si abilita a livello di CDB

```
SQL> alter session set container=xenia01;
Session altered.

SQL> alter database flashback on;

Error starting at line : 1 in command -
alter database flashback on
Error report -
ORA-03001: unimplemented feature
03001. 00000 - "unimplemented feature"
*Cause:      This feature is not implemented.
*Action:     None.
```





## Restore Point

Un restore point è un semplice alias di SCN.

Quando è garantito evita che i flashback logs successivi vengano rimossi.

Ricordarsi di rimuovere sempre i Restore Point garantiti quando non servono più

```
create restore point cdb_before_patch;  
drop restore point cdb_before_patch;  
  
create restore point cdb_before_test guarantee  
flashback database;  
drop restore point cdb_before_test;
```





I Restore Point possono essere creati anche a livello di PDB

```
SQL> alter session set container=xenia01;  
Session altered.
```

```
SQL> create restore point xenia01_rp01;  
Restore point created.
```

```
SQL> alter session set container=CDB$ROOT;  
Session altered.
```

```
SQL> create restore point xenia01_rp02 for pluggable database xenia01;  
Restore point created.
```

```
SQL> create restore point cdb_rp01;  
Restore point created.
```



Flashback  
PDB





```
SQL> select scn, storage_size, name, pdb_restore_point from v$restore_point;
```

SCN	STORAGE_SIZE	NAME	PDB
3830414	0	XENIA01_RP01	YES
3830423	0	XENIA01_RP02	YES
3830437	0	CDB_RP01	NO

 Flashback  
PDB

```
SQL> drop restore point cdb_rp01;  
Restore point dropped.
```

```
SQL> drop restore point xenia01_rp01 for pluggable database xenia01;  
Restore point dropped.
```

```
SQL> drop restore point xenia01_rp02 for pluggable database xenia01;  
Restore point dropped.
```







Il Flashback di un PDB si implementa con un semplice comando dopo aver chiuso il PDB

```
alter pluggable database xenia01 close;  
flashback pluggable database xenia01 to restore point xenia01_rp02;  
alter pluggable database xenia01 open resetlogs;
```



Flashback  
PDB

Il Flashback a livello del database (ovvero del CDB) coinvolge tutti i PDB

```
shutdown immediate;  
startup mount;  
flashback database to restore point cdb_rp02;  
alter database open resetlogs;  
alter pluggable database all open resetlogs;
```





Può capitare che non si sappia a quale SCN tornare indietro e l'unico modo per capirlo è guardare il contenuto delle tabelle. In questi casi è possibile aprire il PDB in Read Only.

```
alter pluggable database xenia01 close;  
flashback pluggable database xenia01 to scn 4791907;  
alter pluggable database xenia01 open read only;
```



Flashback  
PDB

E poi si può scegliere di tornare avanti

```
alter pluggable database xenia01 close;  
recover pluggable database xenia01 until scn 4791945;  
alter pluggable database xenia01 open read only;
```

I 2 passi precedenti si alternano fino a quando non si trova l'SCN voluto, quindi si può aprire il PDB per il normale utilizzo

```
alter pluggable database xenia01 close;  
alter pluggable database xenia01 open resetlogs;
```





PDB  
Refresh e  
Cloning

Il **cloning** consiste nel replicare un database (PDB oppure un NON PDB) in un nuovo PDB di un CDB.

Con il **refresh** è possibile aggiornare la copia iniziale applicando solo le modifiche.

*Un inizio travagliato*

**12.1.0.1:** Remote Cloning presente fra le funzionalità ma non funzionava perfettamente

**12.1.0.2:** il source database deve essere messo in Read-Only prima di iniziare il processo di cloning. Inaccettabile per la maggior parte dei sistemi di produzione

**12.2:** funziona l'Hot Cloning di un DB (PDB o non-CDB)





**Requisiti:** i più significativi sono

- Modalità LOCAL UNDO per il CDB sorgente
- Modalità ARCHIVELOG per il DB sorgente
- Se sul DB sorgente è abilitata il TDE il CDB di destinazione richiede una configurazione particolare
- Sorgente e destinazione condividono: ENDIANESS, CHARACTER SET, opzioni



PDB  
Refresh e  
Cloning

Per il refresh si sono vincoli aggiuntivi

- PDB sorgente e PDB destinazione risiedono su 2 CDB separati
- Gli archive log necessari al refresh si possono leggere tramite db-link e tramite area condivisa
- Il PDB destinazione deve essere creato come refreshable PDB
- Il PDB destinazione è stato aperto solo in Read Only





## Implementazione

Nel DB destinazione occorre creare un DB LINK che punta al sorgente

```
create database link clone_link
connect to c##clone_user
identified by oracle
using '//192.xxx.1.1:1521/orcl12c'
;
```

Creare il PDB sulla destinazione come copia di un PDB sorgente Negli esempi che seguono orcl è un PDB presente sul sorgente

- Se il sorgente è un PDB

```
create pluggable database orcl_clone
from orcl@clone_link
file_name_convert=(
  '/u01/orcl12c/orcl/',
  '/u01/orcl_clone/'
)
;
```

Per il NON-PDB è necessario eseguire lo script di conversione

PDB  
Refresh e  
Cloning

- Se il sorgente è un NON-PDB

```
create pluggable database orcl_clone
from NON$CDB@clone_link;

alter session set container=orcl_clone;

@$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql
```





Per il refresh basta aggiungere la clausola di refresh

```
create pluggable database orcl_clone  
from orcl@clone_link  
refresh mode manual;
```

PDB  
Refresh e  
Cloning

Dopo la creazione sarà in stato MOUNTED, quindi deve essere aperto a mano

```
alter pluggable database orcl_clone open [read only]
```

L'opzione Read Only è un vincolo per il Refresh

Dopo il clone occorre allineare i COMMON USERS





## Refresh

### Refresh manuale

```
alter session set container=orcl_clone;  
alter pluggable database close immediate;  
alter session set pdb_file_name_convert = 'ORCL',  
'ORCL_CLONE';  
alter pluggable database refresh;  
alter pluggable database open read only;
```



In alternativa possiamo abilitare il refresh automatico che però è effettivo solo se il PDB è stato chiuso (mounted)

```
ALTER PLUGGABLE DATABASE pdb5_ro REFRESH MODE EVERY 60 MINUTES;
```

