



# XENIALAB<sup>TM</sup>

WEB AND MOBILE COMMUNICATIONS



**Autore:** meo bogliolo

**Relatore:** meo bogliolo

## Abstract

L'opzione piu' significativa della 12c e' il Multitenant. Presentati i concetti di base si vedono i principali comandi per la gestione di PDB. Quindi vengono riportate le significative evoluzioni sul multitenant della Release 2.





Dal punto di vista dell'architettura il numero di processi e' cresciuto da 6 (v. 6.0) a 64 (v. 12.2)!

L'evoluzione inciampa...  
sta per verificarsi un nuovo salto generazionale: l'architettura Multitenant.

Con il Multitenant vengono ospitate piu' "istanze" nello stesso contenitore.

La terminologia corretta e' quella del PDB (Pluggable Database) ed i processi, la SGA, la base del Data Dictionary sono mantenuti nel CDB (Container Database).  
Pero' dal punto di servizio e di visibilita' per le applicazioni o per gli utenti i PDB corrispondono alle "istanze".





Nell'architettura Multitenant l'istanza CDB mantiene al suo interno piu' PDB utilizzando lo stesso set di processi, la stessa SGA (System Global Area), un unico listener. In pratica tutte le componenti di sistema sono condivise. I tablespaces TEMP e di Undo possono essere comuni o separati per PDB mentre per i dati si utilizzano datafile differenti.

Dal punto di vista logico invece i PDB sono completamente separati e vedono data dictionary distinti, utenti diversi, ...

La creazione di un'utenza e' locale al DB cui si e' connessi ma e' possibile creare nel CDB utenze comuni (c##utente) per tutti i PDB. Le utenze comuni sono tipicamente utenze amministrative o di controllo.

Un'istanza CDB contiene inizialmente il PDB\$SEED in READ ONLY che viene utilizzato come base per la creazione dei successivi PDB.





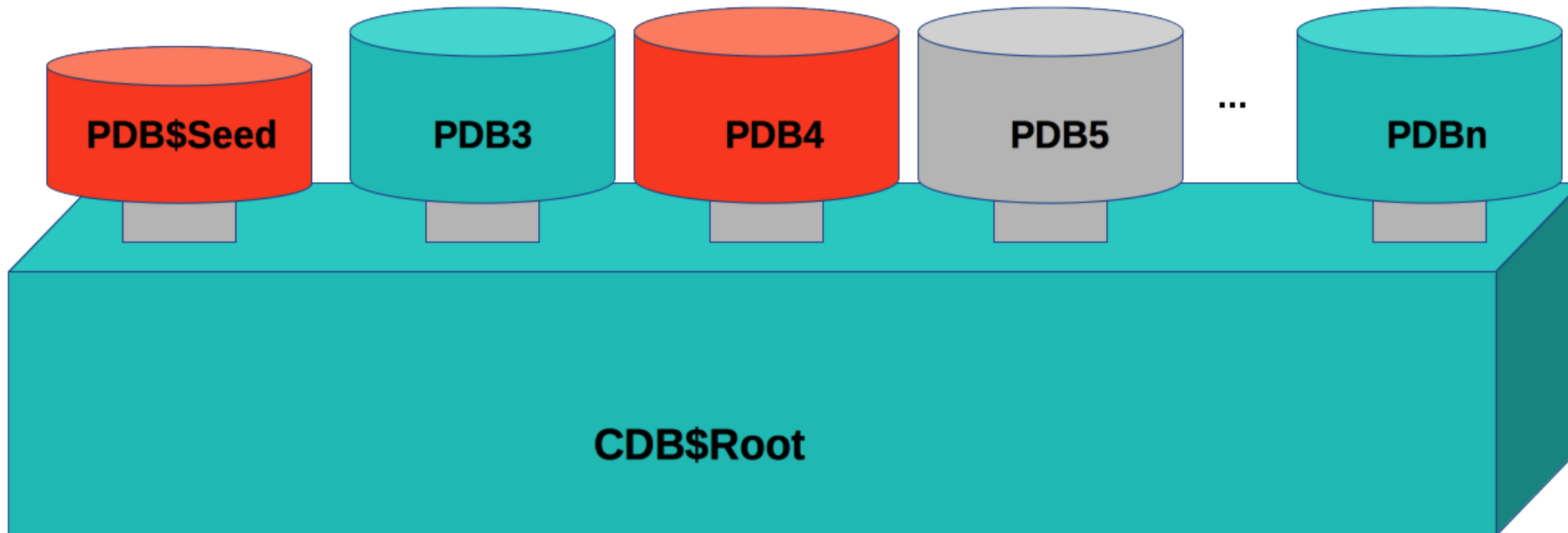
Creiamo i primi PDB:

```
CREATE PLUGGABLE DATABASE pdb3  
  ADMIN USER sysp3 IDENTIFIED BY xxx;  
CREATE PLUGGABLE DATABASE pdb4  
  ADMIN USER sysp4 IDENTIFIED BY xxx;  
CREATE PLUGGABLE DATABASE pdb5  
  ADMIN USER sysp5 IDENTIFIED BY xxx;  
  
alter pluggable database pdb3 open;  
  
alter pluggable database pdb4 open read only;
```





I PDB hanno stati diversi:





Come collegarsi ai PDB (e capire dove si e'):

```
alter session set container=pdb3;  
alter session set container=cdb$root;  
  
conn sys/xxx@pdb3 as sysdba  
  
export TWO_TASK=PDB3  
  
/ as sysdba ed OPS$ --> solo per il CDB!  
  
show con_name  
select Sys_Context('Userenv', 'Con_Name') con_name from dual;
```





Informazioni sui PDB:

```
SELECT NAME, OPEN_MODE, RESTRICTED, OPEN_TIME, TOTAL_SIZE  
FROM V$PDBS;
```

```
SELECT PDB_ID, PDB_NAME, STATUS  
FROM CDB_PDBS;
```

```
select CON_ID, NAME, OPEN_MODE, RESTRICTED, OPEN_TIME, CREATION_TIME,  
TOTAL_SIZE, LOCAL_UNDO, PROXY_PDB, PDB_COUNT, MAX_SIZE, PDB_COUNT  
from V$CONTAINERS;
```

```
select NAME, CDB, CON_ID from V$DATABASE;
```







Ecco il risultato della query (eseguita sul CDB).  
E' chiaro che il conteggio del PDB utente... ricomincia da tre!

```
SELECT NAME, OPEN_MODE, RESTRICTED, OPEN_TIME, TOTAL_SIZE  
FROM V$PDBS;
```

CON_ID	NAME	OPEN_MODE	RES	CREATION_	TOTAL_SIZE
2	PDB\$SEED	READ ONLY	NO	06-MAR-17	780140544
3	PDB3	READ WRITE	NO	06-MAR-17	995098624
4	PDB4	READ ONLY	NO	17-MAR-17	780140544
5	PDB5	MOUNTED		17-MAR-17	0





Cancellare un PDB e migrare un PDB.

```
drop pluggable database pdb5 including datafiles;  
  
alter pluggable database pdb4 close;  
alter pluggable database pdb4 unplug  
  into '/stage/pdb4.xml';  
drop pluggable database pdb4 keep datafiles;  
  
...  
  
CREATE PLUGGABLE DATABASE pdb6  
  USING '/stage/pdb4.xml';
```





Clone di un PDB. Online su 12cR2!

**Source:**

```
CREATE USER c##rclone IDENTIFIED BY xxx CONTAINER=ALL;  
  
GRANT CREATE SESSION, CREATE PLUGGABLE DATABASE  
  TO c##rclone CONTAINER=ALL;
```





Clone di un PDB. Online su 12cR2!

***Destination:***

```
CREATE DATABASE LINK clink1
  CONNECT TO c##rclone IDENTIFIED BY xxx
  USING '@SourceHost:1521/RemoteCDB';

CREATE PLUGGABLE DATABASE pdb7 FROM RemotePDB@clink1;

ALTER PLUGGABLE DATABASE pdb7 OPEN;
```

