



XENIALABTM

WEB AND MOBILE COMMUNICATIONS

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Oracle 12c Release2



Tutto quello che un DBA vorrebbe sapere su Oracle 12c R2...e non ha mai osato chiedere!

Agenda

Introduzione

Evoluzione di Oracle alla 12c Release 2

Multitenant Option

RMAN

Tips&Tricks

PDB PITR, Flashback, Hot Clone, Refresh

Data Guard

Application Containers

Cloud



Webinar 11 Aprile 2017





XENIALABTM

WEB AND MOBILE COMMUNICATIONS



Autore: meo bogliolo

Relatore: meo bogliolo

Abstract

Dopo una breve introduzione sui contenuti delle presentazioni tecniche vengono presentate le nuove funzionalita' della versione 12c e le ultime evoluzioni della versione 12c R2.





Il 6 marzo 2017 e' stata pubblicata la versione Oracle 12c R2 su ambienti on-premise (Linux e Solaris).

E' una "R2". Chi *conosce da tempo* Oracle sa che e' la versione da utilizzare.

La quantita' di innovazione presente in questa nuova release e' notevole. Abbiamo fatto quindi una serie di scelte sui contenuti e sulla tipologia di questa breve presentazione... conoscere e presentare ogni cosa non sarebbe possibile.

Immodestamente:

Questo webinar presenta tutto quello che un DBA Oracle deve sapere sulla 12c R2!

Il taglio e' "pratico" nel senso che sono presentati prima i concetti... ma soprattutto i comandi o gli esempi di utilizzo o le modalita' di configurazione.





Ma oltre ai DBA ci sono anche altri **stakeholders** interessati ad Oracle 12cR2...

Il livello di consolidamento ottenibile con questa versione e' cresciuto in modo molto significativo, quasi non immaginabile con le release precedenti. In particolare sul Multitenant sono state introdotte molteplici, nuove ed importanti funzionalita'.

La flessibilita' delle architetture disponibili e la semplicita' della gestione delle "solite" istanze Oracle sono i due punti di maggiore importanza per sfruttare in modo ottimale l'asset database.

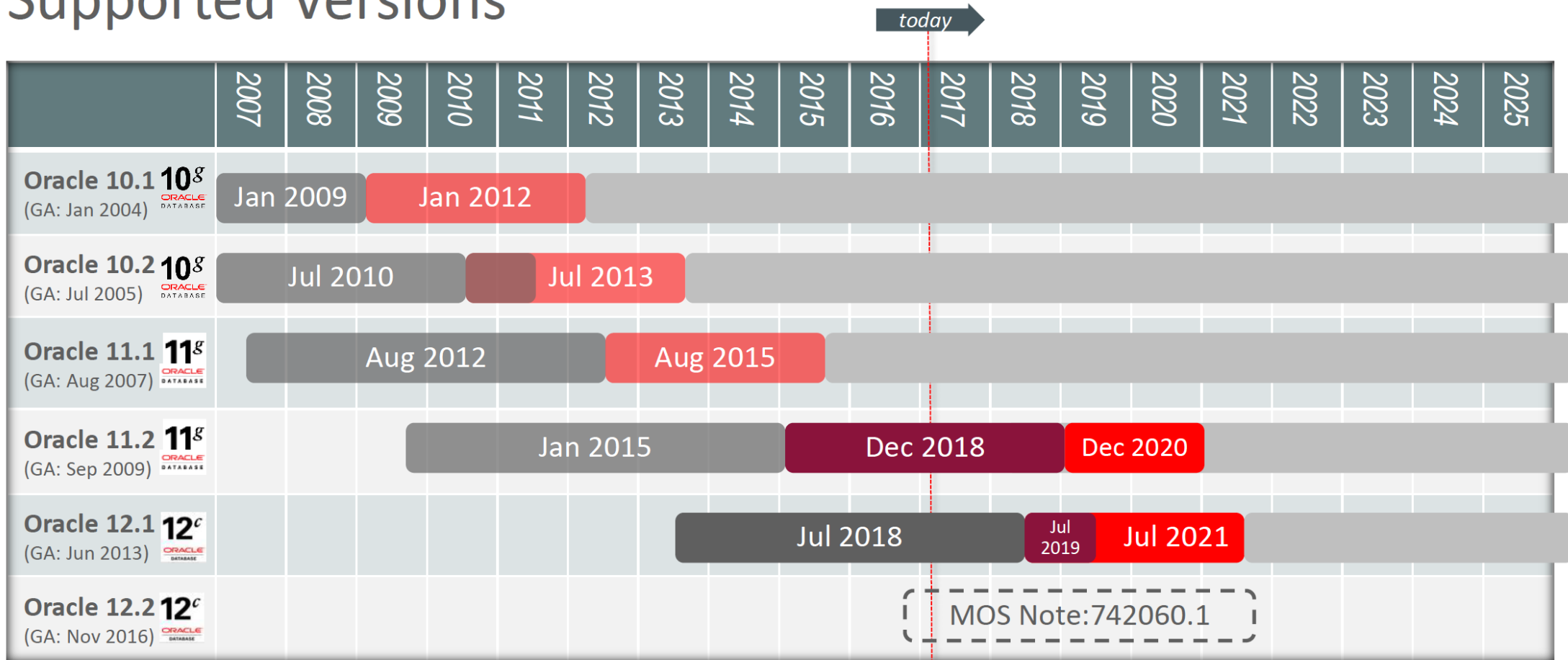
A questo si aggiungono nuove funzionalita' e, non ultimo, un ponte sicuro ed affidabile verso le evoluzioni in Cloud.

Cercheremo quindi di presentare il valore aggiunto della 12cR2 in ogni funzionalita'.





Supported Versions



today →

Premier Support
Waived Extended Support
Paid Extended Support
Sustaining Support





Il 17 marzo per Windows ed il 6 marzo 2017 per Linux e Solaris e' stata rilasciata la 12cR2 on premises.

Dal 5 novembre 2016 e' disponibile la 12cR2 sul Oracle Public Cloud.

Il Premier Support per la 11.2 e' terminato nel gennaio 2015, il waived ES terminera' a fine 2018 e l'ES a fine 2020.

La 12.1.0.2 e' disponibile da Luglio 2014; la 12c R1 e' uscita l'anno precedente.

Le ultime PSU rilasciate erano per la 12c e la 11g R2 [12.1.0.2.170117, 11.2.0.4.161018]. Per le altre releases non vengono piu' rilasciate PSU.

Abbiamo usato l'ordine cronologico inverso...
perche' si farebbe troppo tardi a partire dall'inizio!





Version	Status	Features	Last release	Date (from)	Date (to)
Oracle 12c R2	Production	Online partitioning, Online encryption, DBVault simulation, DBA_INDEX_USAGE, Data Guard automatic deploy, CREATE ANALYTIC VIEW, external and partitioned tables on HDFS, ... For Multitenant: up to 4096 PDB, MAX_PDBS, different character sets, active DG and DG Broker, flashback, hot clone, max_iops, proxy PDB, ... For Cloud: listener multiple redirects, SQLNET.COMPRESSION, sharding option, RMAN cross-platform support, ...	12.2.0.1	2016-11	2021-10 (2024 ES)
Oracle 12c	Production	Oracle Multitenant Option (aka Pluggable DBs); Heat Map Optimizations; Adaptive Query Optimization; RMAN improvements (table restore, standby restore, ...); Availability: transparent failover, Global Data Service; Infrastructure: light-weight grid stack, EM express; ILM (information lifecycle); SQL: default with sequence, varchar up to 32k, with-PLSQL functions, ... Desupport: <= 9.2 clients, raw devices 12.1.0.2: InMemory Option, SE2, ... Deprecation: Non-CDB architecture	12.1.0.1 12.1.0.2	2013-07 2014-07	2016-08 2018-07 (2021 ES)
Oracle 11g R2	Production	Stable, fast, low TCO; ADVM (ASM Dynamic Volume Manager) and ACFS (ASM Cluster File System), Oracle restart (srvctl), Caging (CPU_COUNT), Hybrid Compression; RAC One Node option, SCAN listener, Omotion; 11.2.0.2 Patch Set is a full release Desupport: raw devices in DBCA, Oracle Ultra Search	11.2.0.1 11.2.0.2 11.2.0.3 11.2.0.4	2009-09 2010-09 2011-09 2013-04	2015-01 (2020 ES)





Oracle 11g	Sust.Supp.	Availability enhancements, performance optimization (eg. client cache), native PL/SQL compilation, binary XML Options: Total Recall, Advanced Compression, real application testing (RAT) Desupport: exp, iSQLPlus, <= 8.1.7 clients	11.1.0.7	2007-08 (Linux)
Oracle 10g R2 (10.2)	Sust.Supp.	Stable, COMMIT NOWAIT, fully featured Enterprise Manager, Data Pump compression	10.2.0.5	2005
Oracle 10g (10.1)	Sust.Supp.	sqlplus / as sysdba, Regexp, Flashback tables, "Grid" Desupport: Rule based Optimizer, <= 7.3.4 clients , init parameters	10.1.0.5	2003
Oracle 9i R2 (9.2)	Sust.Supp.	Very stable, JDK 1.3 in JVM, XML	9.2.0.8	2002
Oracle 9i (9.0.1)	Unsupp.	ETL, RMAN, 9i Data Guard, Parallel DB->RAC, MTS->Shared Server Desupport: CONNECT Internal, SQL*Net, Server Manager, ESTAT/BSTAT more...	9.0.1.5	2001
Oracle 8i	Sust.Supp.	8.1.7: Very stable, Enterprise Manager, Apache, JSP 8.1.6: Statspack, DBA Studio, PL/SQL SP 8.1.5: DROP COLUMN, temporary table, <i>Internet</i> , iFS, Java Stored Procedures	8.1.7.4	1999
Oracle 8.0	Unsupp.	Partitioning, <i>Object option</i> , object types, Full NLS, SQL3, SQL*Net -> Net8	8.0.6.3	1997
Oracle 7.3	Unsupp.	Very stable, Temporary tablespaces, Bitmap indexes, UTL_FILE package, multithreading, Spatial Option Desupport: SQL*Net v.1, sqldb	7.3.4.5	1996
Oracle 7.2	Unsupp.	Subquery in FROM clause, features x DBA (resizeable datafile, rollback segment shrink)		1995
Oracle 7.1	Unsupp.	Parallel query, Server Manager, Advanced replication		1994
Oracle 7.0	Unsupp.	Stored procedures and triggers, Integrity constraints (enforcement), cost-based optimizer, SQL*Net v.2 with 2PC and MTS Desupport: SQL*Forms v.2.3		1992
Oracle 6.0	Unsupp.	Row level locking, PL-SQL, Integrity constraints (declaration), new architecture (B*Tree, tablespaces, rollback segments, redo logs) (6.2): Parallel Server Option		1988
Oracle 5.1	Unsupp.	Stable, distributed query		1986





Client Version	Server Version													
	12.2.0	12.1.0	11.2.0	11.1.0	10.2.0	10.1.0	9.2.0	9.0.1	8.1.7	8.1.6	8.1.5	8.0.6	8.0.5	7.3.4
12.2.0	Yes	Yes	Yes	No	No	No	No	No	No	No	No	No	No	No
12.1.0	Yes	Yes	Yes	Was	Was	No	No	No	No	No	No	No	No	No
11.2.0	Yes	Yes	Yes	Was	Was	No	Was	No	No	No	No	No	No	No
11.1.0	No	Was	Was	Was	Was	Was	Was	No	No	No	No	No	No	No
10.2.0	No	Was	Was	Was	Was	Was	Was	No	Was	No	No	No	No	No
10.1.0	No	No	Was	Was	Was	Was	Was	Was	Was	Was	No	No	No	No
9.2.0	No	No	Was	Was	Was	Was	Was	Was	Was	Was	No	No	Was	No
9.0.1	No	No	No	No	No	Was	Was	Was	Was	Was	Was	No	Was	No
8.1.7	No	No	No	No	Was	Was	Was	Was	Was	Was	Was	Was	Was	Was
8.1.6	No	No	No	No	No	No	No	Was	Was	Was	Was	Was	Was	Was
8.1.5	No	No	No	No	No	No	No	No	Was	Was	Was	Was	Was	Was
8.0.6	No	No	No	No	No	No	Was	Was	Was	Was	Was	Was	Was	Was
8.0.5	No	No	No	No	No	No	No	No	Was	Was	Was	Was	Was	Was
7.3.4	No	No	No	No	No	No	Was	Was	Was	Was	Was	Was	Was	Was

Legenda:

Yes – Supportato

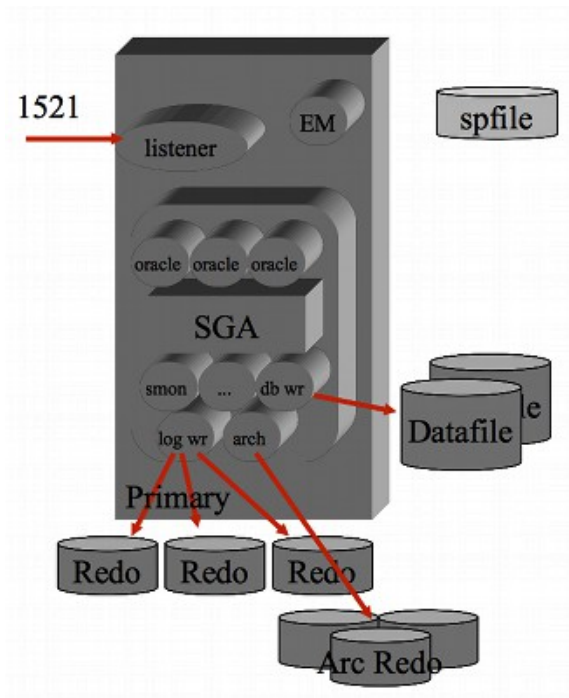
Was – Configurazione funzionante ma non piu' supportata

NO – Configurazione non funzionante

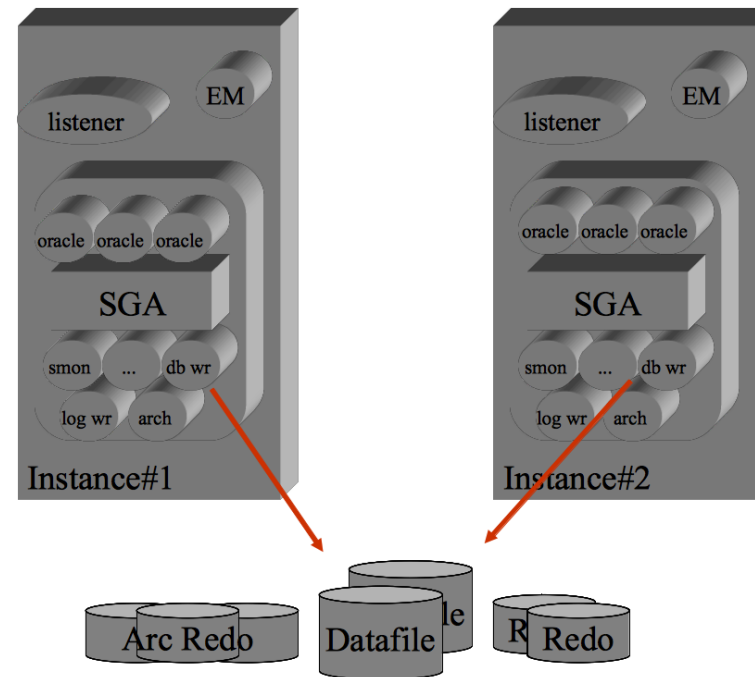




L'architettura di Oracle si e' *solo evoluta* nel tempo (almeno a partire dalla 6.0)...



Oracle RAC





Dal punto di vista dell'architettura il numero di processi e' cresciuto da 6 (v. 6.0) a 64 (v. 12.2), ma le logiche di base sono le stesse!

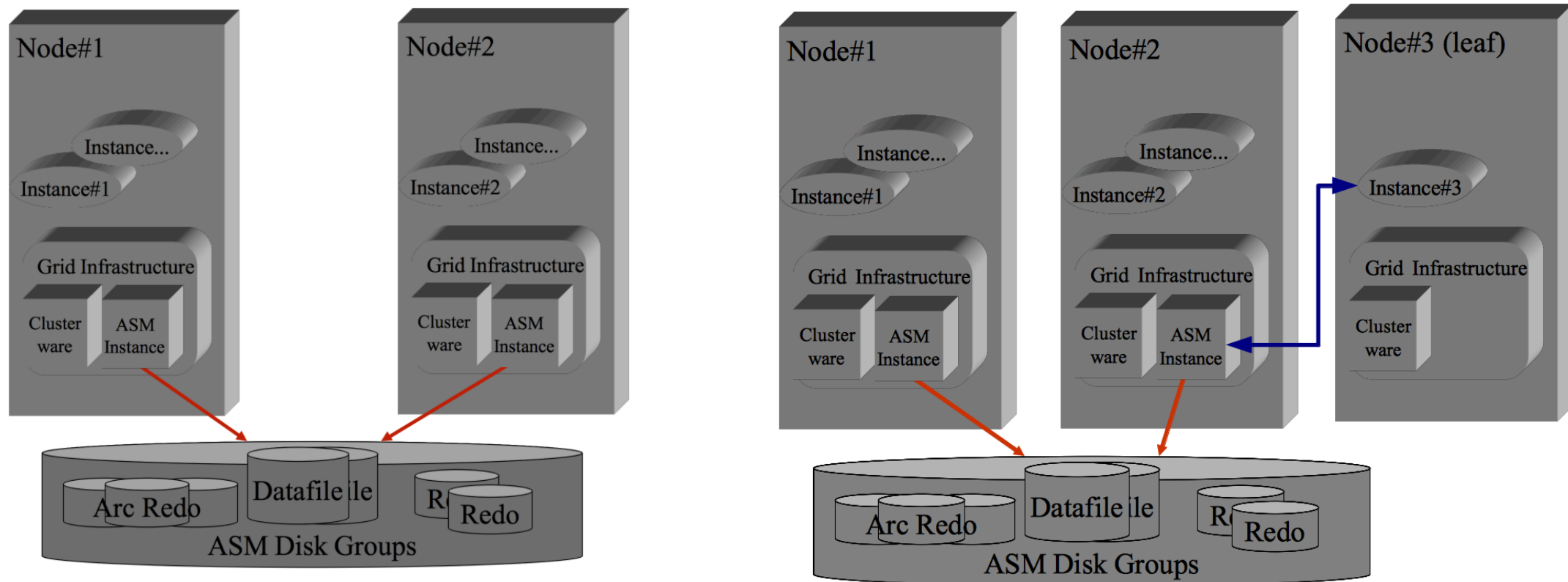
Ma sta per verificarsi un nuovo salto generazionale: l'architettura Multitenant.

E' cosi' importante che merita un capitolo a parte...
Quindi ora vediamo le altre innovazioni ed evoluzioni della 12c!





La componente Grid in 12c puo' utilizzare la configurazione **Flex**





Il vecchio, caro, ROWNUM, sostituito con il **fetch**:

```
select *  
  from scott.emp  
 order by sal desc  
 fetch first 5 rows only;
```

```
select *  
  from orders  
 order by id  
 offset 10 rows fetch next 5 rows only;
```





L'**Heat Map** e' semplice da utilizzare.

Ma la sua potenza sara' sfruttarla per una politica di ILM (Information Lifecycle Management) integrata con l'ADO (Automatic Data Optimization) e l'Advanced Compression Option

```
ALTER SYSTEM SET HEAT_MAP = ON;
```

```
select *  
  from DBA_HEATMAP_TOP_OBJECTS;
```

```
ALTER TABLE fatture_attive ILM ADD POLICY  
  COMPRESS FOR ARCHIVE HIGH SEGMENT  
  AFTER 12 MONTHS OF NO ACCESS;  
ALTER TABLE fatture_passive ILM ADD POLICY  
  TIER TO tbs_tier3 SEGMENT  
  AFTER 6 MONTHS OF LOW ACCESS;
```





In-Memory Option: come attivarla. Usarla e' banale, lo fa qualsiasi SELECT!

```
SQL> alter system set inmemory_size=8G scope=spfile;
```

```
SQL> shutdown immediate
```

```
SQL> startup
```

```
SQL> alter table MyImportantTable inmemory;
```





Diverse nuove funzionalita' sul partitioning tra cui: **Online partitioning**

```
ALTER TABLE t1 MODIFY
PARTITION BY RANGE (amount) INTERVAL (100)
( PARTITION P1 VALUES LESS THAN (300),
  PARTITION P2 VALUES LESS THAN (700) )
ONLINE
UPDATE INDEXES
( IDX01_AMOUNT LOCAL
( PARTITION IP1 VALUES LESS THAN (MAXVALUE) ) );
```





L'Encryption (**TDE**) puo' essere utilizzato sul singolo campo o su un intero tablespace.
La conversione di un tablespace pero' richiedeva fosse messo in READ ONLY, ora con la 12cR2:

```
alter tablespace USERS  
  ENCRYPTION online using 'AES256' ENCRYPT  
  FILE_NAME_CONVERT = ('users.dbf', 'users_enc.dbf');
```





Il Database Vault puo' essere utilizzato in Simulation Mode:

```
BEGIN
  DBMS_MACADM.CREATE_REALM(
    realm_name      => 'HR Apps',
    description     => 'Realm to protect the HR realm',
    enabled         => DBMS_MACUTL.G_SIMULATION,
    audit_options  => DBMS_MACUTL.G_REALM_AUDIT_FAIL,
    realm_type     => 1,
    realm_scope    => DBMS_MACUTL.G_SCOPE_LOCAL);
END;
/

SELECT USERNAME, COMMAND, SQLTEXT, VIOLATION_TYPE
   FROM DBA_DV_SIMULATION_LOG
  WHERE REALM_NAME = "HR APPS";
```





Utilizzo degli indici (vista **DBA_INDEX_USAGE**):

```
select *  
  from DBA_INDEX_USAGE  
 where TOTAL_ACCESS_COUNT=0;
```





Sharding Option: architettura shared nothing con il supporto completo dell'SQL e delle transazioni

```
ALTER SESSION ENABLE SHARD DDL;  
  
CREATE SHARDED TABLE Customers  
( CustNo          NUMBER NOT NULL  
, Name           VARCHAR2(50)  
, Address        VARCHAR2(250)  
, CONSTRAINT RootPK PRIMARY KEY(CustNo) )  
PARTITION BY CONSISTENT HASH (CustNo)  
PARTITIONS AUTO  
TABLESPACE SET ts1;
```

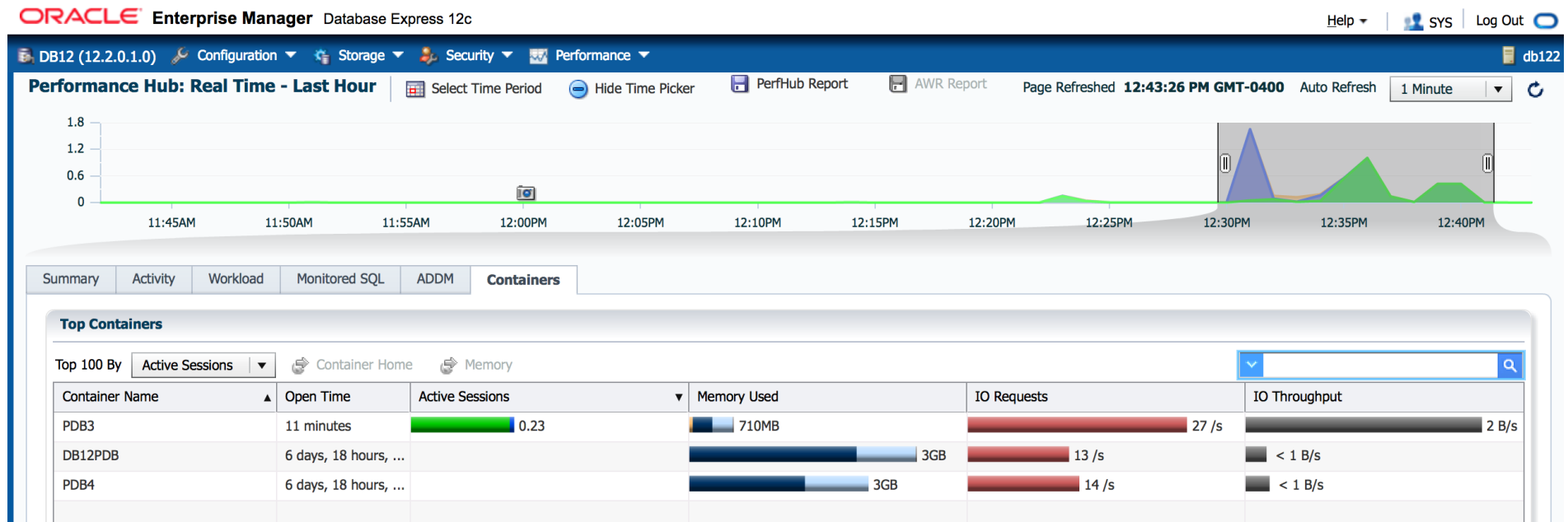




Molte altre evoluzioni della 12c (vale per tutti e tre i rilasci della 12c) non si vedono ma ci sono!

Ad esempio l'**ottimizzatore** e' ancora migliorato, le statistiche vengono raccolte con maggior frequenza e con snapshot "piu' leggeri", molti limiti sono stati aumentati (eg. varchar fino a 32K), la **compressione** e' sempre piu' evoluta, ...

Per chiudere con un'immagine del nuovo **Enterprise Manager**:





XENIALABTM

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 evoluzioni del 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)! Ma l'evoluzione inciampa...

Sta per verificarsi un nuovo salto generazionale: l'architettura **Multitenant**.

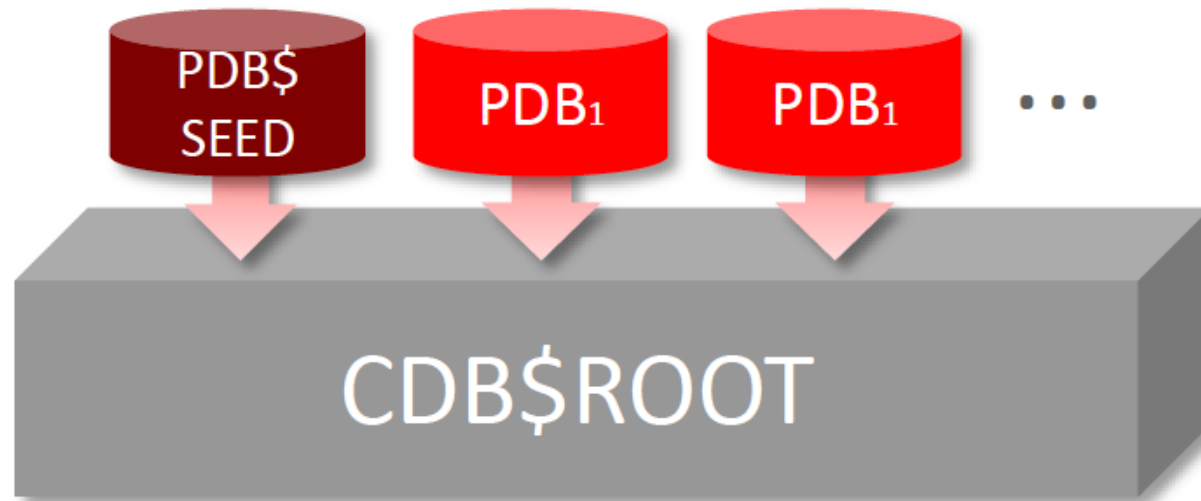
Con il Multitenant vengono ospitate piu' "istanze" nello stesso contenitore.

La terminologia corretta e' PDB (Pluggable Database) mentre i processi, la SGA, la base del Data Dictionary sono fanno parte del CDB (Container Database).

Pero' dal punto di servizio e di visibilita' per le applicazioni o per gli utenti i PDB corrispondono alle "istanze".



Multitenant Option in Oracle 12cR2





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 tablespace TEMP e di Undo possono essere comuni o separati per PDB mentre per i dati si utilizzano datafile differenti.

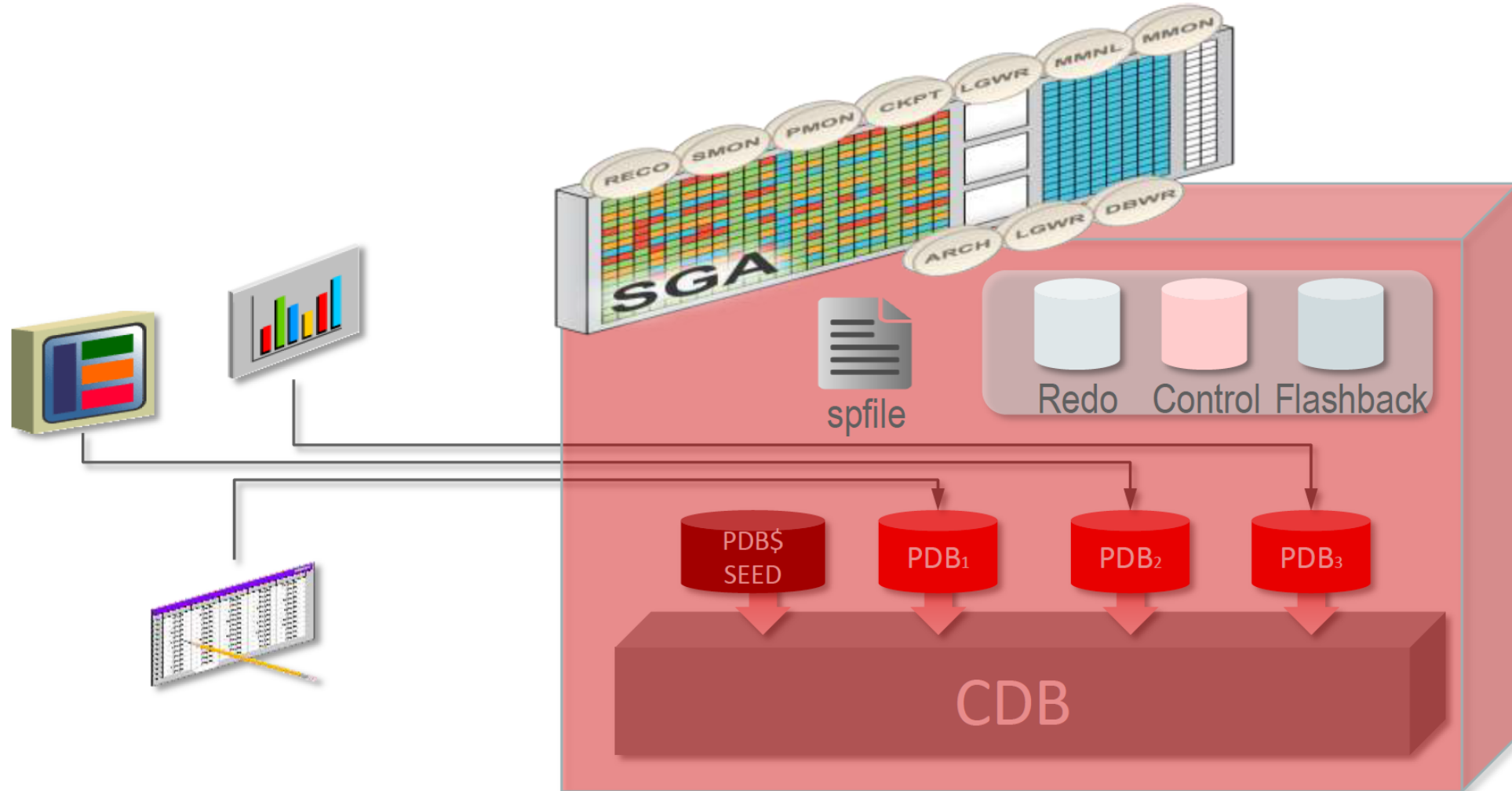
Dal punto di vista logico invece i PDB sono completamente separati e vedono data dictionary distinti, utenti diversi, ...

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.



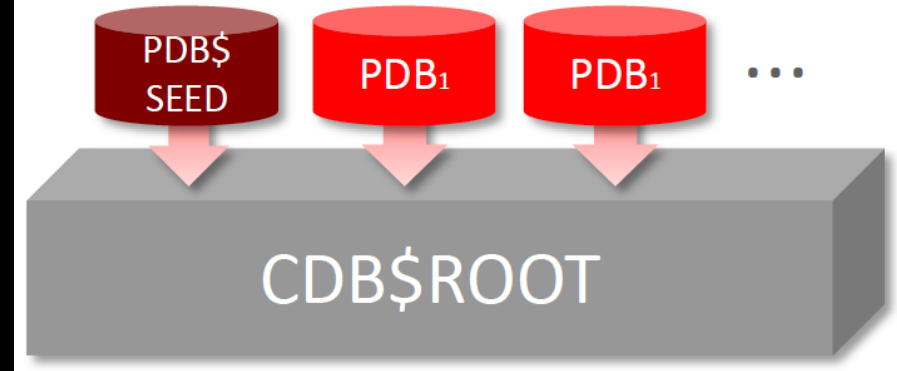
Multitenant Option in Oracle 12cR2





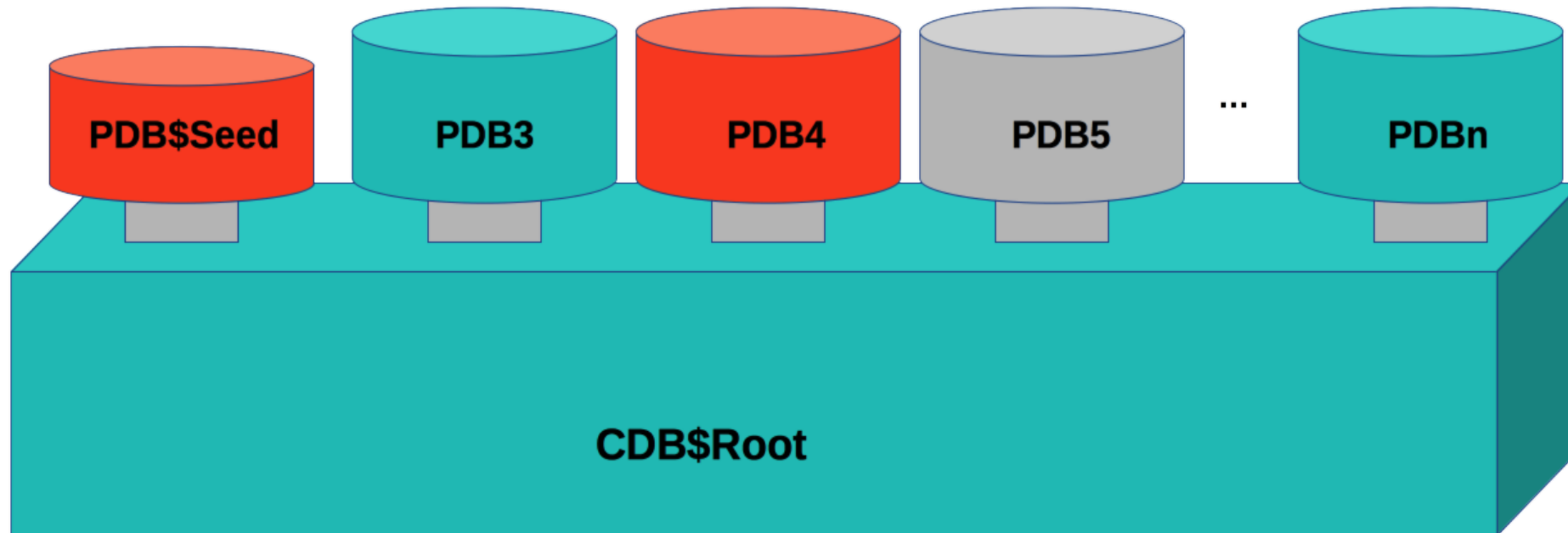
Creiamo i primi PDB:

```
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;
```





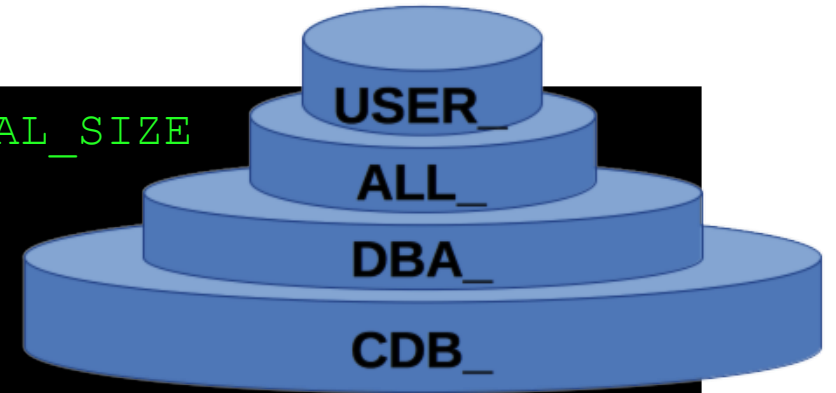
Informazioni sui PDB:

```
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;
```





XENIALABTM

WEB AND MOBILE COMMUNICATIONS



Abstract

- RMAN e il Multitenant
- Table Point In Time Recovery (PITR)
- RMAN e il Cloud

Autore: Umberto Signori

Relatore: Umberto Signori

... tante novita' per rendere eccitante una parte spesso considerata noiosa ma sempre estremamente importante per la sicurezza dei dati





Il seguente comando salva tutto

```
RMAN> BACKUP DATABASE PLUS ARCHIVELOG;
```

- CDB
- PDB\$SEED
- Tutti i PDB
- Controlfile
- SPFILE
- Archivelog

Occhio alla modalità di
connessione al target

RMAN
e il
Multitenant

```
RMAN> BACKUP PLUGGABLE DATABASE orcl, xenia01;
```

```
RMAN> BACKUP DATABASE ROOT;
```





I risultato dei 2 comandi che seguono è lo stesso

```
rman target sys/oracle@vbgeneric:1521/orcl12c <<EOF  
  BACKUP PLUGGABLE DATABASE XENIA01;  
EOF
```

```
connected to target database: ORCL12C (DBID=768045447)
```

```
rman target sys/oracle@vbgeneric:1521/xenia01 <<EOF  
  BACKUP DATABASE;  
EOF
```

```
connected to target database: ORCL12C:XENIA01 (DBID=212407603)
```



**RMAN
e il
Multitenant**

Entrambi salvano solo il PDB chiamato XENIA01 e nulla più





```
RMAN> report schema;
```

```
Report of database schema for database with db_unique_name ORCL12C
```

```
List of Permanent Datafiles
```

```
=====
```

File	Size (MB)	Tablespace	RB segs	Datafile Name
1	810	SYSTEM	YES	/u01/app/oracle/oradata/orcl12c/system01.dbf
3	490	SYSAUX	NO	/u01/app/oracle/oradata/orcl12c/sysaux01.dbf
5	250	PDB\$SEED:SYSTEM	NO	/u01/app/oracle/oradata/orcl12c/pdbseed/system01.dbf
6	330	PDB\$SEED:SYSAUX	NO	/u01/app/oracle/oradata/orcl12c/pdbseed/sysaux01.dbf
7	5	USERS	NO	/u01/app/oracle/oradata/orcl12c/users01.dbf
8	100	PDB\$SEED:UNDOTBS1	NO	/u01/app/oracle/oradata/orcl12c/pdbseed/undotbs01.dbf
9	350	ORCL:SYSTEM	YES	/u01/app/oracle/oradata/orcl12c/orcl/system01.dbf
10	1170	ORCL:SYSAUX	NO	/u01/app/oracle/oradata/orcl12c/orcl/sysaux01.dbf
11	460	ORCL:UNDOTBS1	YES	/u01/app/oracle/oradata/orcl12c/orcl/undotbs01.dbf
12	73	ORCL:USERS	NO	/u01/app/oracle/oradata/orcl12c/orcl/users01.dbf
...				

```
List of Temporary Files
```

```
=====
```

File	Size (MB)	Tablespace	Maxsize (MB)	Tempfile Name
1	33	TEMP	32767	/u01/app/oracle/oradata/orcl12c/temp01.dbf
2	64	PDB\$SEED:TEMP	32767	/u01/app/oracle/oradata/orcl12c/pdbseed/temp11.dbf
3	64	ORCL:TEMP	32767	/u01/app/oracle/oradata/orcl12c/orcl/temp01.dbf
4	64	XENIA01:TEMP	32767	/u01/app/oracle/oradata/orcl12c/xenia01/temp12.dbf

In report schema si nota la modalita per identificare i tablespace all'interno de PDB

RMAN
e il
Multitenant





Per salvare i comandi del CDB/PDB al quale si è connessi il comando non è

```
BACKUP TABLESPACE system, sysaux, users;
```

Dal CDB posso anche salvare i tablespace di un PDB

```
backup tablespace XENIA01:SYSTEM;
```

Dal PDB invece non posso salvare ciò che non appartiene al PDB

```
connected to target database: ORCL12C:XENIA01 (DBID=212407603)
```

```
RMAN> BACKUP DATAFILE 1;  
channel ORA_DISK_1: SID=15 device type=DISK  
RMAN-00571: =====  
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====  
RMAN-00571: =====  
RMAN-03002: failure of backup command at 04/01/2017 08:27:12  
RMAN-20201: datafile not found in the recovery catalog  
RMAN-06010: error while looking up datafile: 1
```



Anche i comandi
LIST BACKUP e
REPORT SCHEMA
hanno risultati
diversi se connessi
al CDB o al PDB





Per il restore dell'intero database non è cambiato nulla

```
$ rman target /  
  
RUN {  
  SHUTDOWN IMMEDIATE;  
  STARTUP MOUNT;  
  RESTORE DATABASE;  
  RECOVER DATABASE;  
  ALTER DATABASE OPEN;  
}
```

Con il multitenant posso ripristinare i singoli PDB

```
$ rman target=/  
RUN {  
  ALTER PLUGGABLE DATABASE XENIA01 CLOSE;  
  RESTORE PLUGGABLE DATABASE XENIA01;  
  RECOVER PLUGGABLE DATABASE XENIA01;  
  ALTER PLUGGABLE DATABASE XENIA01 OPEN;  
}
```

Restore della ROOT.

```
$ rman target=/  
  
RUN {  
  SHUTDOWN IMMEDIATE;  
  STARTUP MOUNT;  
  RESTORE DATABASE ROOT;  
  RECOVER DATABASE ROOT;  
  # Consider recovering PDBs before opening.  
  ALTER DATABASE OPEN;  
}
```

RMAN
e il
Multitenant





Table PITR nelle release precedenti

Il Table Point In Time Recovery era un'operazione lunga ed elaborata.

La complessità nasce dal dover estrarre informazioni logiche da un backup fisico.

Erano necessari i seguenti passi

- Creare un'istanza temporanea
- Restore del set di tablespaces
- Point In Time Recovery
- Export della tabella dal database temporanea
- Import nel database originale
- Rimozione del database temporaneo



Table
PITR





Dalla versione 12 basta un comando di poche righe

```
rman target /  
  
RECOVER TABLE ubi.tabella  
  UNTIL SCN 3565401  
  AUXILIARY DESTINATION '/u01/aux'  
  REMAP TABLE  
ubi.tabella:tabella_copy;
```



Table
PITR

**Si può scegliere se
creare una copia della
tabella oppure fermarsi
al file di export**

```
RECOVER TABLE ubi.tabella  
  UNTIL SCN 1853267  
  AUXILIARY DESTINATION '/u01/aux'  
  DATAPUMP DESTINATION  
  '/u01/export'  
  DUMP FILE 'tabella.dmp'  
  NOTABLEIMPORT;
```





Con il Multitenant c'è la possibilità di specificare il nome del PDB

```
RECOVER TABLE HR.EMPLOYEES
OF PLUGGABLE DATABASE ORCL
UNTIL SCN 4635828
AUXILIARY DESTINATION '/u01/aux'
REMAP TABLE HR.EMPLOYEES:XENIA.EMPLOYEES
#NOTABLEIMPORT
DATAPUMP DESTINATION '/u01/export'
DUMP FILE 'EMPLOYEES-03.dmp';
```



Table
PITR

Le opzioni REMAP TABLE e NOTABLEIMPORT sono mutuamente esclusive

Se la tabella destinazione esiste lo script termina subito in errore
RMAN-05112: table "UBI"."TAB01_PITR" already exists





Quali oggetti vengono creati dall'import?

```
SQL> select object_name, object_type from cdb_objects where owner = 'XENIA';
```

OBJECT_NAME	OBJECT_TYPE
EMPLOYEES	TABLE
SECURE_EMPLOYEES	TRIGGER
UPDATE_JOB_HISTORY	TRIGGER
EMPLOYEES_EMPLOYEE_ID_TRG	TRIGGER

Quando si usa l'opzione REMAP i constraint e gli indici con nome non vengono importati per evitare conflitti con gli oggetti esistenti

I constraint di tipo CHECK vengono importati e potrebbero generare errore.
Conviene creare la tabella sotto uno schema creato ad hoc.





RECOVER TABLE esegue una lunga serie di operazioni.

1. Verifica subito che ci sia abbastanza spazio nell'AUXILIARY DESTINATION.
In caso contrario interrompe subito lo script
2. Crea un database ausiliario



```
Creating automatic instance, with SID='CctE'
```

```
initialization parameters used for automatic instance:
```

```
db_name=ORCL12C
```

```
db_unique_name=CctE_pitr_XENIA01_ORCL12C
```

3. Ripristina il recovery set

```
channel ORA_AUX_DISK_1: starting datafile backup set restore
```

```
channel ORA_AUX_DISK_1: specifying datafile(s) to restore from backup set
```

```
channel ORA_AUX_DISK_1: restoring datafile 00012 to /u01/aux/mf_users_%u...
```





4. Point In Time Recovery

starting media recovery

```
archived log for thread 1 with sequence 10 is already on disk as file  
/u01/app/oracle/fast_recovery_area/.../o1_mf_1_10_dglbopln_.arc
```

5. Esporta la tabella dal DB ausiliario

```
EXPDP> . . exported "HR"."EMPLOYEES"          17.07 KB          107 rows
```

6. Importa la tabella nel DB target

```
IMPDP> Processing object type TABLE_EXPORT/TABLE/TABLE_DATA  
IMPDP> . . imported "XENIA"."EMPLOYEES"      17.07 KB          107 rows
```

7. Rimuove dump e DB ausiliario

```
Removing automatic instance  
Automatic instance removed  
auxiliary instance file /u01/aux/ORCL12C/.../o1_mf_temp_dglf9q04_.tmp deleted
```





Sono gli stessi passi che il DBA deve fare manualmente nelle versioni precedenti

```
starting media recovery
```

```
archived log for thread 1 with sequence 10 is already on disk as file  
/u01/app/oracle/fast_recovery_area/.../o1_mf_1_10_dg1bopln_.arc
```

L'istanza ausiliaria ha lo stesso db_name dell'istanza target, non deve ingannare il messaggio di riavvio dell'istanza. I riavvii riguardano sempre l'istanza ausiliaria, mai l'istanza target

```
Creating automatic instance, with SID='AxEA'
```

```
initialization parameters used for automatic instance:
```

```
db_name=ORCL12C
```

```
db_unique_name=AxEA_pitr_ORCL_ORCL12C
```

```
...
```

```
starting up automatic instance ORCL12C
```



Table
PITR





Vantaggi

- Sicuro
- Scalabile
- Tempi di implementazione ridotti
- I dati viaggiano criptati
- Compression



Requisiti

- Connessione a Internet
- Sistemi operativo: Linux, Solaris x86-64, SPARC, Windows, AIX, HP-UX, zLinux





Implementazione

1. Sottoscrizione: occorre aderire al servizio, volendo si può provare con una licenza trial
2. Installazione dell'Oracle Database Cloud Backup Module

```
java -jar opc_install.jar -serviceName myService  
-identityDomain myDomain -opcId 'myAccount@myCompany.com'  
-opcPass 'myPassword' -walletDir /walletDirectory  
-libDir /libraryDirectory
```



RMAN
e il
Cloud

3. Configurazione

```
RMAN> CONFIGURE CHANNEL DEVICE TYPE sbt  
PARMS='SBT_LIBRARY=location-of-the-SBT-library,  
SBT_PARMS=(OPC_PFILE=location-of-the-configuration file)';  
  
## esempio  
RMAN> CONFIGURE CHANNEL DEVICE TYPE sbt  
PARMS='SBT_LIBRARY=/orclhome/lib/libopc.so,  
SBT_PARMS=(OPC_PFILE=/orclhome/dbs/opctl.ora)';
```





4. Configurazione.

E' obbligatorio che i backup siano criptati con una delle seguenti modalità

- Password encryption

```
RMAN> SET ENCRYPTION ON IDENTIFIED BY 'xxx' ONLY;  
RMAN> BACKUP DEVICE TYPE sbt DATABASE;
```

- Transparent Data Encryption (TDE): occorre configurare il wallet in modo che non richieda la password ad ogni backup/restore

```
RMAN> SET ENCRYPTION ON;  
RMAN> BACKUP DEVICE TYPE sbt DATABASE;
```

- Dual-mode encryption (combination of password and TDE)

```
RMAN> SET ENCRYPTION ON IDENTIFIED BY 'my_pswd';  
RMAN> BACKUP DEVICE TYPE sbt DATABASE;
```

5. Abilitare la compressione

```
RMAN> CONFIGURE COMPRESSION ALGORITHM 'MEDIUM';  
RMAN> CONFIGURE DEVICE TYPE sbt BACKUP TYPE TO COMPRESSED BACKUPSET;
```





Restore

Niente di particolare oltre a ciò che è necessario per i restore criptati

```
RMAN> SET DECRYPTION [ON|IDENTIFIED BY 'my_pswd'];  
RMAN> RESTORE DATABASE;  
RMAN> RECOVER DATABASE;
```



RMAN
e il
Cloud





XENIALABTM

WEB AND MOBILE COMMUNICATIONS



Autore: meo bogliolo

Relatore: meo bogliolo

Abstract

In ogni software ci sono trucchi che offrono funzionalità aggiuntive e tranelli che possono fare perdere tempo...





E' possibile, anzi e' un'architettura consigliata, utilizzare un PDB senza *pagare* l'opzione Multitenant.
E' la configurazione Singletenant:

```
alter session set container=cdb$root;  
alter system set max_pdb=1 scope=both;
```





Dal CDB e' possibile accedere ad ogni oggetto sui PDB con una query:

```
alter session set container=cdb$root;

select *
  from containers (dual)
 where con_id<5;

select con_id, name, value
  from V$CON_SYSSTAT
 where name like 'physical % total bytes'
 order by con_id, name;
```





E' possibile lanciare con un comando lo stesso script su tutti i PDB con **catcon.pl**.
[NdA nell'esempio la procedura di pre-upgrade]:

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl \  
-n 1 -d $ORACLE_HOME/rdbms/admin -l /home/oracle/upgrade \  
-b preupgrd preupgrd.sql
```





Utilizzate assolutamente il Local Undo Mode!
Consigliabile anche l'Oracle Managed Files.

```
shutdown immediate
startup upgrade
alter database local undo on;
shutdown immediate
startup
```

```
alter system
  set DB_CREATE_FILE_DEST='/u02/app/oracle/oradata/db12'
  scope=both;
```





La 12cR2 e' progettata per il consolidamento delle istanze: definite i limiti delle risorse dei PDB.

```
dbms_resource_manager.create_cdb_plan(plan =>'base_plan',  
  comment =>'Resource Management Plan');  
  
dbms_resource_manager.CREATE_CDB_PROFILE_DIRECTIVE(  
  plan => 'newcdb_plan',  
  profile => 'gold',  
  shares => 10,  
  utilization_limit => 100,  
  parallel_server_limit => 100);
```





Definite i limiti delle risorse!
[NdA repetita juvant]

```
alter system set max_pdb=200 scope=both;
```

```
alter session set container=PDB3;
```

```
ALTER SYSTEM SET SGA_TARGET=1G SCOPE=BOTH;
```

```
ALTER SYSTEM SET SGA_MIN_SIZE=500M SCOPE=BOTH;
```

```
ALTER SYSTEM SET max_iops=1000 SCOPE=BOTH;
```

```
ALTER SYSTEM SET max_mbps=400 SCOPE=BOTH;
```





Hybrid Cloud? Occhio al TDE!

[NdA encrypt_new_tablespaces=CLOUD_ONLY]

```
select con_id, TABLESPACE_NAME, ENCRYPTED
       from cdb_tablespaces order by 1,2;
```

CON_ID	TABLESPACE_NAME	ENC
1	SYSAUX	NO
1	SYSTEM	NO
1	TEMP	NO
1	UNDOTBS1	NO
1	USERS	YES
3	SYSAUX	NO
3	SYSTEM	NO
3	TEMP	NO
3	UNDOTBS1	NO
3	USERS	YES





XENIALABTM

WEB AND MOBILE COMMUNICATIONS



Abstract

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

Autore: Umberto Signori

Relatore: Umberto Signori

Come cambiano i “ribaltamenti” con la 12cR2





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.
```





```
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

```
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.
```



Flashback
PDB





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





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.

Funzionalità crescenti:

12.1.0.1: Remote Cloning presente fra le funzionalità ma con molti limiti

12.1.0.2: Il source database deve essere messo in Read-Only prima di iniziare il processo di cloning. Un vincolo importante 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



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;

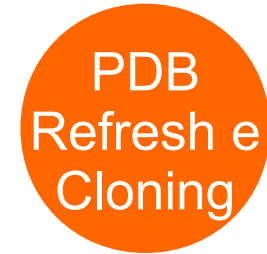
@$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql
```





Per il refresh basta aggiungere la clausola di

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



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

```
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

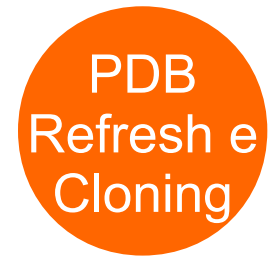




Refresh

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;
```





XENIALABTM

WEB AND MOBILE COMMUNICATIONS



Autore: luigi mendicino

Relatore: meo bogliolo

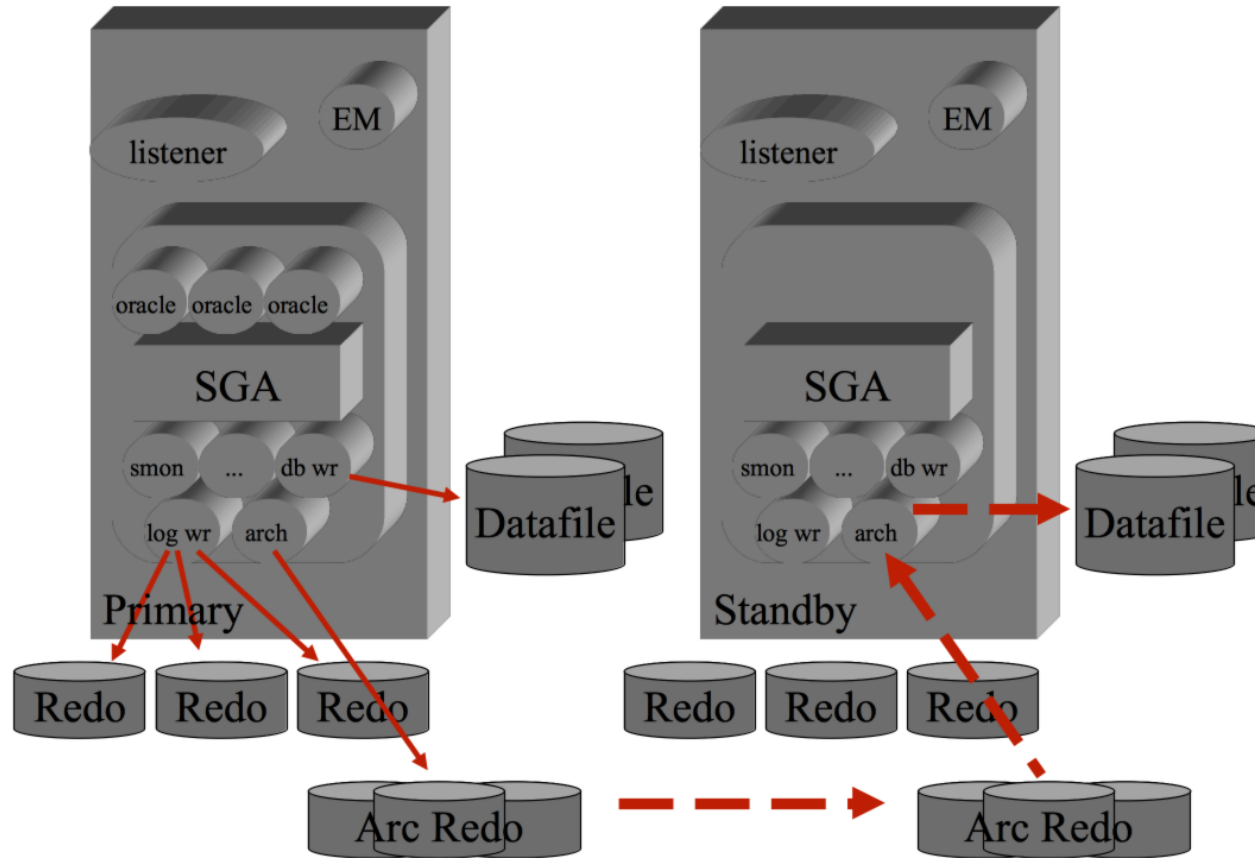
Abstract

Il titolo in realta' e' riduttivo... L'obiettivo e' quello di vedere come si evolvono le architetture di Business Continuity con la versione 12cR2. Ma certamente DataGuard ha un ruolo centrale su questa problematica.





Oracle Data Guard





DATA GUARD

Cosa c'è di nuovo

- Multi-Instance Redo apply
- IN-MEMORY column store
- Recover di operazioni in NO-LOGGING
- Data Blocks LOST-WRITES - Shadow Lost Write Protection
- Diagnostic Pack supportato sullo standby con Active Data Guard
- Password File gestiti e trasportati via Redo
- Possibilità di replicare solo un sotto-insieme di PDBs

Miglioramenti

- DGBROKER
- FAST FAILOVER





MULTI-INSTANCE REDO APPLY

Dalla versione 12cRelease 2 (12.2.0.1), e' disponibile una nuova clausola INSTANCES [ALL | integer] per il comando SQL ALTER DATABASE RECOVER MANAGED STANDBY DATABASE:

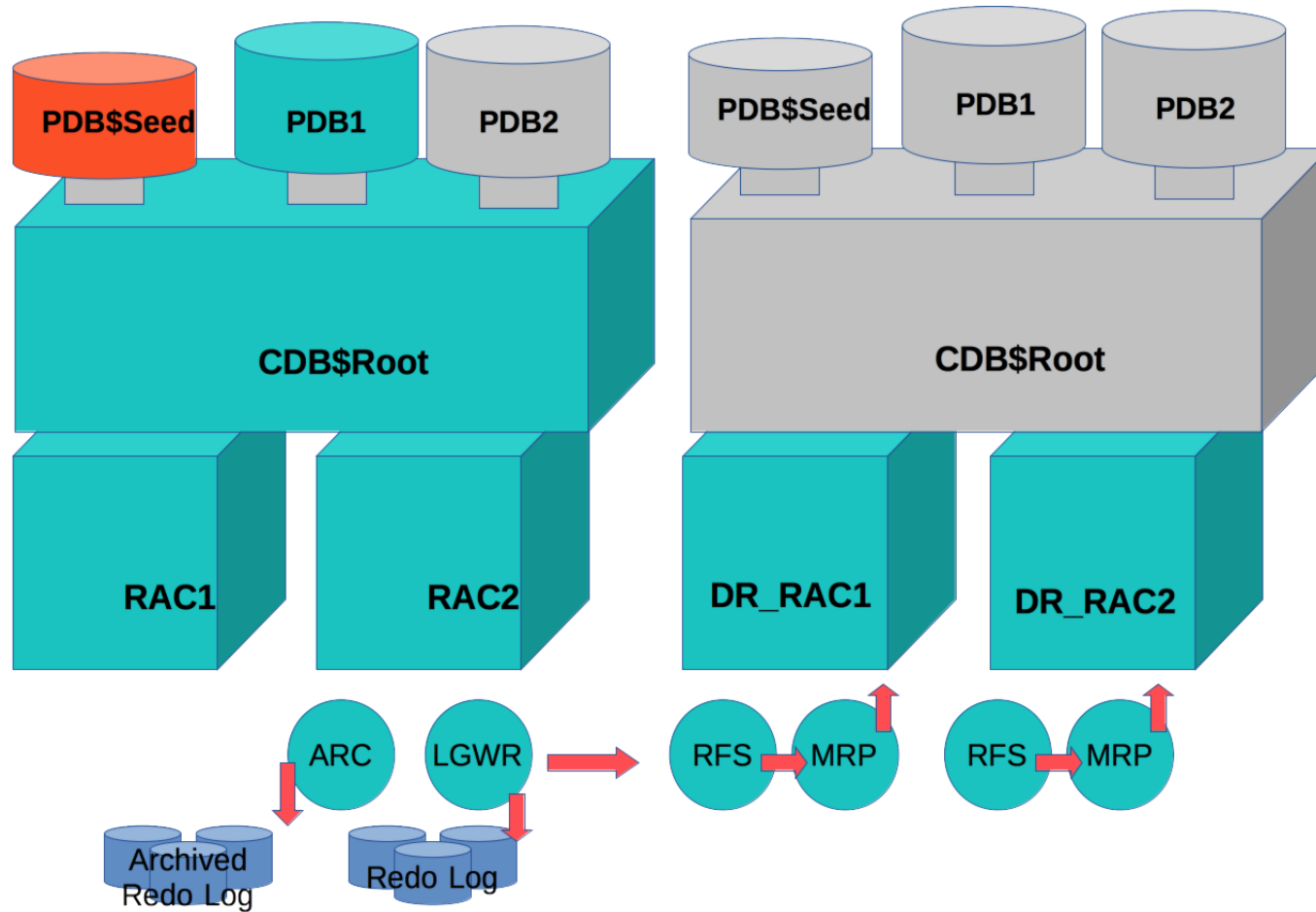
```
SQL> Recover managed standby database disconnect using instances 3;
```

Restrizioni:

- Applicabile solo a configurazioni di tipo Oracle Real Application Clusters (Oracle RAC) o Oracle RAC One Node.
- Block Change tracking non supportato.



Oracle 12cR2 DataGuard





ACTIVE DATA GUARD

Cos'è

Oracle Active Data Guard è la soluzione Oracle per la protezione dei dati in tempo reale e di Disaster Recovery, utilizzando un processo di replica fisica.

Oracle ADG fornisce anche un elevato ROI (Return On Investment) nei sistemi di DR, consentendo ad uno standby database di essere aperto in READ-ONLY, mentre si applicano le modifiche ricevute dal database primario.

Licensing

Oracle Active Data Guard è un prodotto con licenza separata che fornisce funzionalità avanzate che ampliano notevolmente le Data Guard capabilities incluse nella Oracle Enterprise Edition.





ACTIVE DATA GUARD Perchè adottarlo

- Production-offload per applicazioni di tipo read-only, reporting (ad esempio OBIEE), e ad-hoc queries.
- Oracle Active Data Guard 12c permette operazioni di tipo **DML su global temporary tables** e l'uso di **sequence sullo standby database**;
- **Connection preservation** sull' Active Data Guard standby durante un 'role change';
- Automatic Workload Repository (**AWR**)/ Automatic Database Diagnostic Monitor (**ADDM**);
- Active Data Guard supporta le opzioni di **Oracle Database In-Memory column store**.





IN-MEMORY Column store con Active Data Guard

Possibilità di definire due set differenti di dati per il primario e lo standby da storicizzare in memoria.

Restrizioni

- Le In-Memory Expression sono basate sulle query eseguite sul primario
- La In-Memory column store non è supportato in una configurazione multi-instance redo apply.
- Necessario l'Active Data Guard.





DATA BLOCK LOST WRITES

Si verifica quando un sottosistema di I/O riconosce il completamento della scrittura di un blocco, mentre in realtà la scrittura non avviene a livello storage, ad esempio per un fault di un disco, di un controller, di una componente di rete, etc etc!

Il sintomo più comune in queste circostanze è un blocco corrotto.

New in 12cR2: DBMS_DBCOMP.DBCOMP

```
dbms_dbcomp.dbcomp (  
  datafile      IN VARCHAR2,  
  outputfile   IN VARCHAR2,  
  block_dump   IN BOOLEAN := FALSE);
```

Confronta blocco per blocco i datafiles tra il db primario e lo standby.





DBMS_DBCOMP.DBCOMP

```
exec dbms_dbcomp.dbcomp('1', '/home/oracle/lost_wr_example.out', TRUE);
```

Se eseguita dal db primario, il datafile interessato è confrontato block-by-block con tutti gli standby; Al contrario, se la procedura viene eseguita dallo standby, la compare avviene tra il db primario e quello specifico standby database.

Si può monitorare attraverso la vista V\$SESSION_LONGOPS

```
SELECT target_desc, sofar, totalwork  
FROM v$session_longops  
WHERE opname = 'BlockCompare';
```

TARGET_DESC	SO FAR	TOTALWORK
-----	-----	-----
Compared Blocks	367104	403142
Lost Writes	0	0





SHADOW LOST WRITES PROTECTION

Prima della 12cR2 era necessario uno standby database con il parametro DB_LOST_WRITE_PROTECT settato sia sul primario che sullo standby.

New in 12cR2

12.2 Shadow Lost Write Protection - Detect Lost Writes Without Needing Standby (Doc ID 2159248.1)

va abilitato prima a livello di database e poi sul tablespace:

```
SQL>  
SQL> alter database enable lost write protection;  
  
SQL> alter tablespace USERS enable lost write protection;
```





OPERAZIONI IN NOLOGGING SUL PRIMARIO

New in 12cR2:

Viene spedita la lista dei blocchi non-logged allo standby database, e vengono memorizzati nel controlfile.

Quali blocchi sono stati spediti allo standby?

```
SQL> select * from V$nonlogged_block;
```

Non è necessario recuperare l'intero datafile. Basta un singolo comando (via RMAN):

```
RMAN> recover database nonlogged block;
```





Il 'database role' è definito a livello di CDB, ma è possibile replicare solo un **set** di PDBs? Yes, we can!

•ENABLED_PDBS_ON_STANDBY

Il nuovo parametro ENABLED_PDBS_ON_STANDBY permette di specificare un sotto insieme di PDBs per la replica su un physical standby di un CDB.

Nella 12cR1 era possibile specificare solo ALL o NONE.

Con la 12.2 il DG Broker supporta con lo ZDLRA piattaforme diverse (eg. Sparc vs Intel vs AIX).





XENIALABTM

WEB AND MOBILE COMMUNICATIONS



Autore: giorgio costamagna

Relatore: umberto signori

Abstract

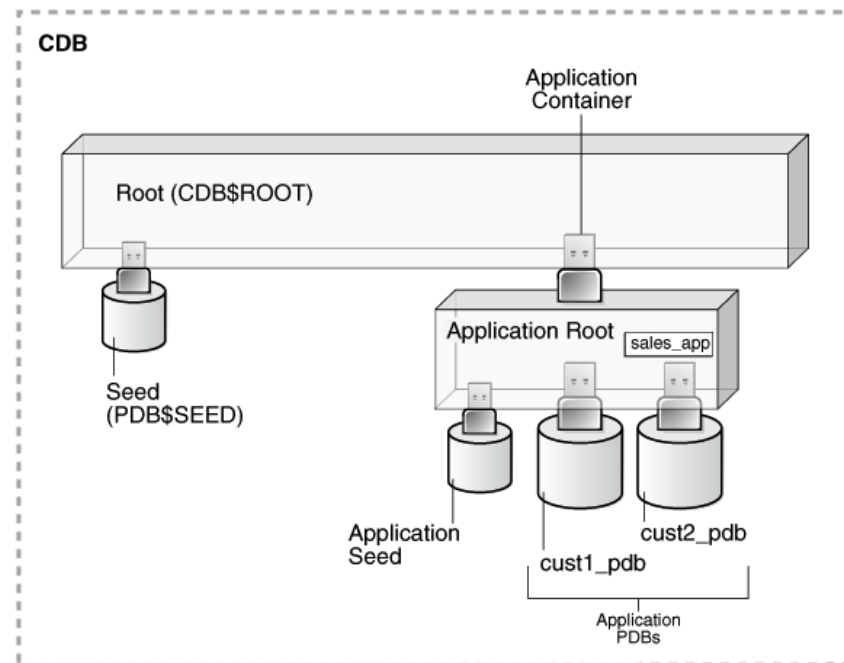
Tra le diverse innovazioni delle release 12cR2 sul Multitenant quella degli Application Containers e' la piu' significativa per i gruppi di sviluppo delle applicazioni.





La Multitenant Option cambierà il modo di operare su Oracle in modo significativo per tutti i DBA.

Un significativo cambiamento è introdotto per gli sviluppi applicativi dalla funzionalità degli Application Containers della 12cR2.





Nel Multitenant i PDB fanno riferimento al Data Dictionary del CDB per le parti comuni.

Nello stesso modo i PDB Applicativi fanno riferimento alla struttura del database definito nell'Application Container.

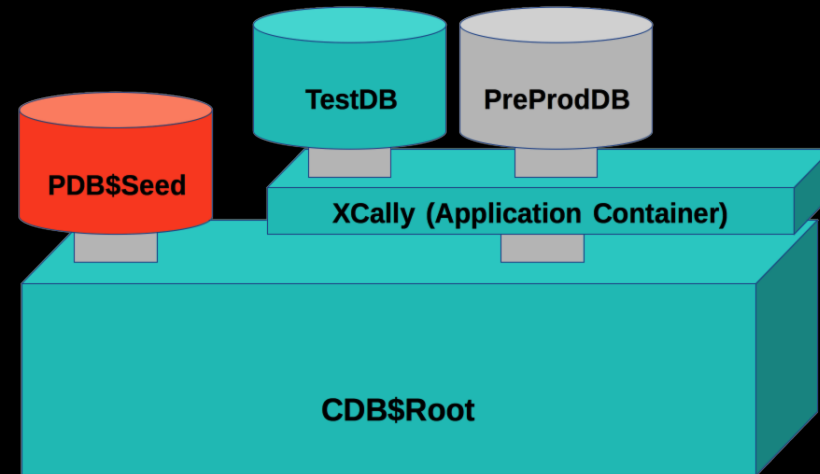
Quando un PDB viene inserito su un CDB e si sincronizza sull'applicazione eredita tutte le strutture dati ed, eventualmente, i dati definiti nell'Application Container.





La prima cosa da creare e' l'Application Container
quindi, dall'Application Container, vanno creati i PDB:

```
create pluggable database xCally as application container  
admin user db00 identified by xxx;  
alter pluggable database xCally open;  
  
alter session set container=xCally;  
create pluggable database TestDB;  
create pluggable database PreProdDB;  
  
alter pluggable database TestDB open;
```





Viene ora creata la struttura applicativa.

Vi sono tre differenti modalita' di condivisione delle tabelle:

```
alter pluggable database application Motion begin install '1.0';
create user xc identified by xxx;
grant connect, resource, unlimited tablespace to xc;

create table xc.emp7A SHARING=METADATA
  (empno integer not null, ename varchar2(20));

create table xc.emp7B SHARING=DATA
  (empno integer not null, ename varchar2(20));

create table xc.emp7C SHARING=EXTENDED DATA
  (empno integer not null, ename varchar2(20));
```





Vengono creati tutti gli oggetti applicativi e popolati i dati:

```
create unique index pkemp7a on xc.emp7A (EMPNO);
create unique index pkemp7b on xc.emp7B (EMPNO);
create unique index pkemp7c on xc.emp7C (EMPNO);

insert into xc.emp7A(empno, ename)
  select level, 'SMITH' from dual connect by level <= 7;
insert into xc.emp7B(empno, ename)
  select level, 'SMITH' from dual connect by level <= 14;
insert into xc.emp7C(empno, ename)
  select level, 'SMITH' from dual connect by level <= 7;

alter pluggable database application Motion end install '1.0';
```





Effettuato il sync dal PDB si accede alla struttura/dati applicativi:

```
alter session set container=TestDB;
alter pluggable database application Motion sync;

insert into xc.emp7A(empno, ename)
  select level+7, 'SMITH' from dual connect by level <= 7;
select count(*)
  from xc.emp7A;

select count(*)
  from xc.emp7B;

select count(*)
  from xc.emp7C;
```





La definizione a livello di Application Container delle strutture dati applicative cambia in modo significativo le modalita' di gestione del ciclo di vita delle applicazioni.

Un PDB puo' migrare tra ambienti diversi e consentire tutte le verifiche applicative senza richiedere passaggi di DDL o il caricamento iniziale dei dati.

Per sfruttare appieno questa possibilita' vanno cambiate le modalita' di gestione del ciclo di vita delle applicazioni. Non e' un processo breve ma una volta utilizzata la nuova modalita' i vantaggi saranno notevoli.





XENIALABTM

WEB AND MOBILE COMMUNICATIONS



Autore: XeniaLAB

Relatore: Meo Bogliolo

Abstract

Oracle 12c R2 e' vicina al Cloud ed avvicina al Cloud per diversi ottimi motivi. Cercheremo di vedere i principali punti salienti sia dal punto di vista tecnico che dal punto di vista di vantaggio competitivo.

Curiosamente le prime diapositive presenteranno come e' tutto uguale tra on-premise e cloud... ma quelle successive saranno su cosa invece e' diverso!





Le versioni Oracle disponibili in Cloud sono le stesse disponibili on-premises, anzi la 12cR2 e' stata resa disponibile prima in Cloud che on-premises.

L'offerta e' molto completa ed elastica: si puo' utilizzare una metrica per ore di utilizzo e si arriva ai modelli Engineered piu' performanti come gli Exadata.

Le prestazioni sono Enterprise Level e la latenza molto bassa.
I datacenter sono in tutto il mondo ed ovviamente anche in Europa.

Dal punto di vista applicativo e di gestione... **con il Cloud non cambia nulla!**





Creare un clone di un PDB da o verso Oracle Public Cloud e' identico a farlo on-premises!

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;
```





I Proxy, i Tombstone ed i Clone PDB saranno i protagonisti della nuova serie:
The Walking D... atabases

```
CREATE PLUGGABLE DATABASE pdb7  
  FROM Remote1PDB@clink1;  
ALTER PLUGGABLE DATABASE pdb7 OPEN;
```

```
CREATE PLUGGABLE DATABASE pdb8  
  FROM Remote1PDB@clink1 RELOCATE  
  [AVAILABILITY MAX];  
ALTER PLUGGABLE DATABASE pdb8 OPEN;
```

THE
WALKING D...





Encryption, TDE, Compression, listener redirect, ...
Licenze, option, versioni, ...

Ci sono oggettivi vantaggi nel Cloud ma per sfruttarlo al meglio non basta soddisfare i prerequisiti tecnici ed un gruppo DBA competente: occorre conoscere le proprie applicazioni, gestire al meglio gli spazi, mantenersi aggiornati, definire e gestire gli SLA, ... e da ultimo, ma non per ultimo: gestire la sicurezza!

Insomma dal punto di vista applicativo e di gestione... **con il Cloud cambia tutto!**

DBA, sei pronto per il Cloud? Sono nato pronto!





Grazie!

Per informazioni

Info@Xenialab.it

Guido.Marino@Xenialab.it

WWW.XENIALAB.IT



www.xenialab.com

