

# **Best Features of Oracle Multitenant**

By Deiby Gómez

OTN Tour 2016 Uruguay, Argentina, Perú, Guatemala, Colombia

# DEIBY GOMEZ

- Oracle ACE - *Youngest in the world.*
- Oracle ACE Director - *Youngest in the world 2015*
- Presidente del Grupo de Usuarios de Oracle de Guatemala
- Speaker
  - OTN '13, '14, '15, '16
  - Oracle Open World USA, Brasil
  - Collaborate, Las Vegas, USA
- Director of Support Quality en LAOUC (*Latin American Oracle Users Group Community*)
- Co-fundador de OraWorld Team (*Suiza, India, Brasil, Francia, Guatemala*)
- Oracle Certified Master 11g (**OCM 11g**) – *Youngest in Latin America*
- Oracle Certified Master 12c (**OCM 12c**) – *Youngest in Latin America*
- SELECT Journal Editor's Choice Award 2016 (*Las Vegas, IOUG*)



# DEIBY GOMEZ



*Pre-order in Amazon!*

## Autores

- Anton Els (Nueva Zelanda)
- Vit Špinka (Republica Checa)
- Franck Pachot (Suiza)

## Revisores Técnicos:

- Deiby Gómez (Guatemala)
- Arup Nanda (India)



- Experiencia del equipo:
  - Oracle Linux Certified Implementation Specialist
  - Oracle Database 11g Administrator Certified Professional (**OCP 11g**)
  - Oracle Database 11g Administrator Certified Master (**OCM 11g**)
  - Oracle Database 12c Administrator Certified Professional (**OCP 12c**)
  - Oracle Service Oriented Architecture Infrastructure Implementation Certified Expert
  - Oracle Certified Expert, Oracle **Exadata** X3 and X4 Administrator
  - Oracle **RAC** 11g and Grid Infrastructure Administrator
  - Oracle Real Application Clusters 12c Certified Implementation Specialist
  - Oracle Database 12c Administrator Certified Master (**OCM 12c**)
- Soporte y Consultoría:
  - Bases de Datos Oracle.
  - Exadata
  - Middleware
- Cursos de Administración de Oracle



# Full Database Caching

## DB size > Buffer Cache

### Little tables:

- Allocated in memory If the table size < **2%** of the Buffer Cache size

### Medium tables:

- Allocated in memory if table size [**2%,10%**] of Buffer Cache size (No official information)
- Oracle analyze the date when the table was used last time
- Free space in Buffer Cache
- Etc

### Big Tables

- Big tables are not allocated in Buffer Cache
- Use "KEEP" to advice Oracle to allocate a big table in Buffer Cache



# Full Database Caching

## Buffer Cache > DB size

### Little tables:

- Allocated in memory If the table size < 2% of the Buffer Cache size

### Medium tables:

- Allocated in Buffer Cache if the table size < 5% of Buffer Cache size (No official information)
- Oracle analyze the table if the table was used last time
- Free space in Buffer Cache
- Etc

### Big Tables

- Big tables are not allocated in Buffer Cache
- Use "KEEP" to advice Oracle to allocate a big table in Buffer Cache



# Full Database Caching

## Buffer Cache > DB size

- Tables are not allocated in Buffer Cache **immediately**
- Buffer Cache must be bigger than DB **(Oracle doesn't verify this)**

## Is Full Database Caching enabled?

```
SQL> SELECT FORCE_FULL_DB_CACHING FROM V$DATABASE;
```

```
FORCE_FULL_DB_CACHING
```

```
-----
```

```
YES
```



# Full Database Caching

Buffer Cache **!>** DB size

What about if I enable it but Buffer Cache **!>** DB Size?

- Algorithms to allocate small, medium and large tables are **not used**.
- More Objects Deallocation
- New object scanned, blocks deallocated.





# Full Database Caching

## How to enabled it?

```
SQL> shutdown immediate;  
SQL> startup mount;  
SQL> ALTER DATABASE FORCE FULL DATABASE CACHING;  
SQL> alter database open;
```

## How to disable it?

```
SQL> shutdown immediate;.  
SQL> startup ;  
SQL> ALTER DATABASE NO FORCE FULL DATABASE CACHING;  
SQL> alter database open;
```



# Automatic Big Table Caching

- Big table cache is used for caching data for table scans.
- Automatic big table caching uses temperature and object based algorithms to track medium and big tables.

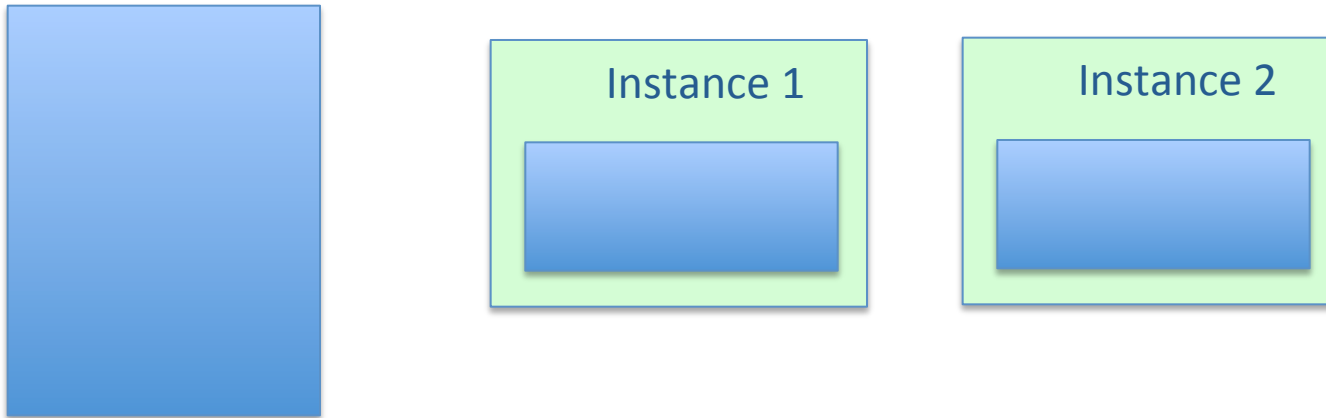
## How to enable it?

- DB\_BIG\_TABLE\_CACHE\_PERCENT\_TARGET
  - percentage **[0%,90%]**
  - 10% of cache for No table scans
- Set PARALLEL\_DEGREE\_POLICY for RAC



# Automatic Big Table Caching

If a large table is approximately the size of the combined size of the big table cache of all instances, then the table is partitioned and cached, or mostly cached, on all instances.

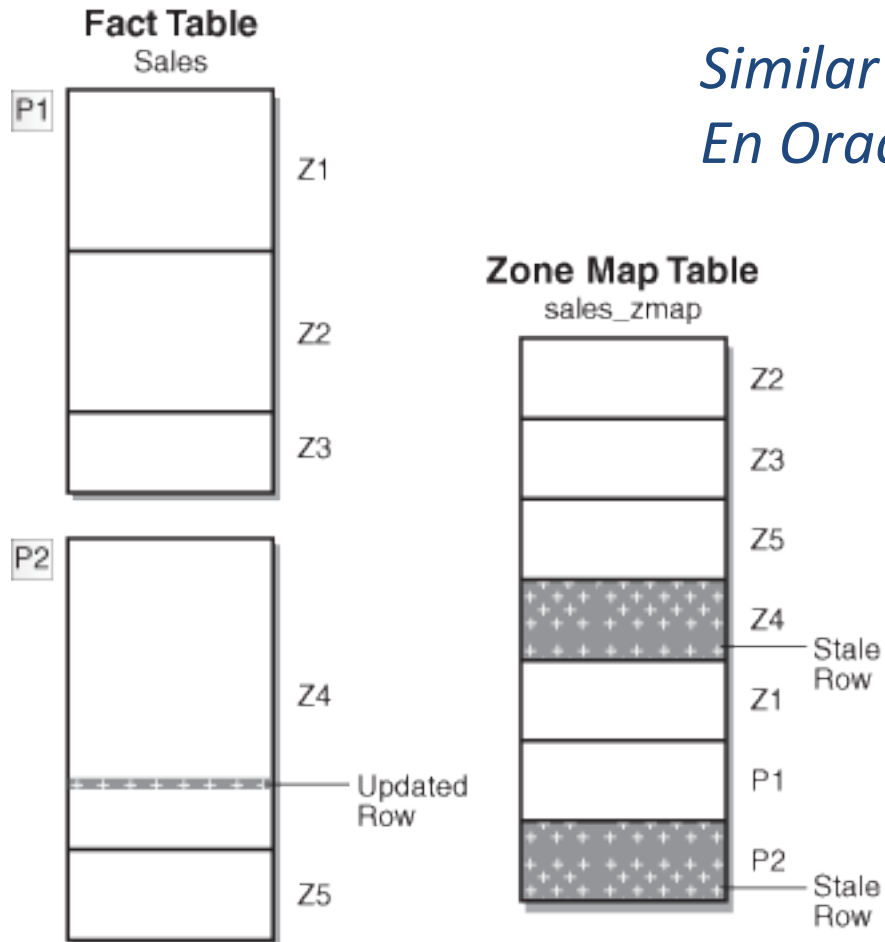


**Where to check information about big table caching?**

- V\$BT\_SCAN\_CACHE
- V\$BT\_SCAN\_OBJ\_TEMPS



# Zone Maps



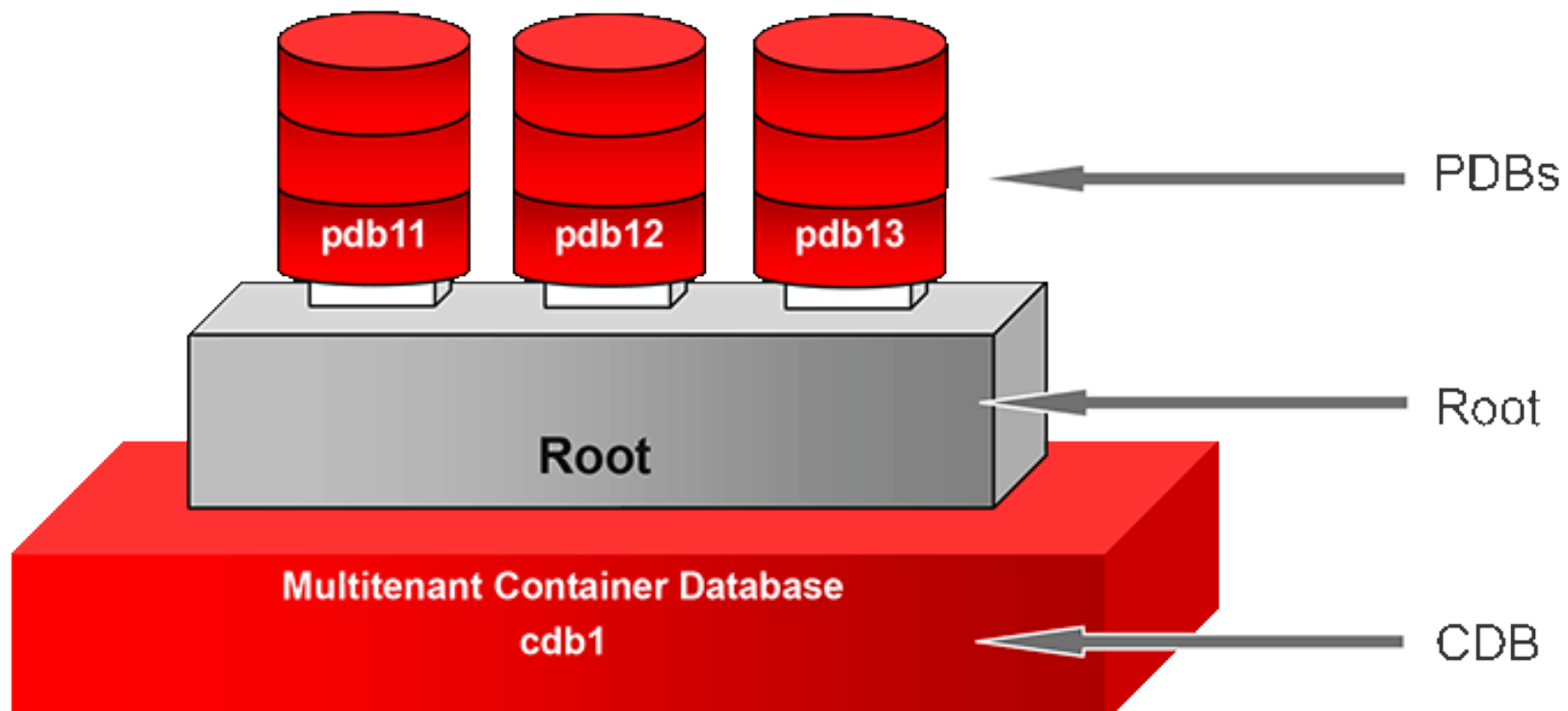
*Similar a Storage Indexes  
En Oracle Exadata pero en Disco*

*Disponible solo en  
Exadata*



# PDB CONTAINERS Clause

```
SELECT ename  
FROM CONTAINERS (scott.emp)  
WHERE CON_ID IN (11, 13);
```



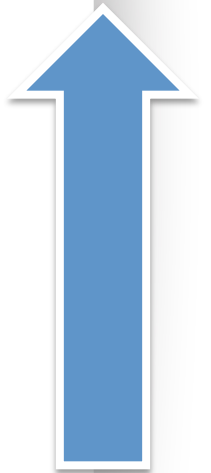
# PDB CONTAINERS Clause

- The tables and views, or synonyms of them, specified in the CONTAINERS clause **must exist in the root and in all PDBs**.
- Each table and view specified in the CONTAINERS clause **must be owned by the common user**.
  - When a synonym is specified in the CONTAINERS clause, the synonym **must resolve** to a table or a view owned by the common user issuing the statement.



# PDB File Placement in OMF

- FILE\_NAME\_CONVERT clause of CREATE PLUGGABLE DATABASE
- **CREATE\_FILE\_DEST** clause of CREATE PLUGGABLE DATABASE
- DB\_CREATE\_FILE\_DEST (root level)
- PDB\_FILE\_NAME\_CONVERT initialization parameter



# PDB Logging Clause

- Table Level – Logging
- Index Level – Logging
- Tablespace Level – Logging
- Tablespace Level - Force Logging
- **PDB Level – Logging**
- **PDB Level - Force No logging**
- CDB - Force Logging





# PDB Logging Clause

```
SQL> alter session set container=pdb1;  
SQL> startup restrict;  
SQL> alter pluggable database pdb1 nologging;  
  
SQL> alter pluggable database pdb1 logging;  
  
SQL> select pdb_name, logging, force_logging,  
force_nologging from cdb_pdbs;
```

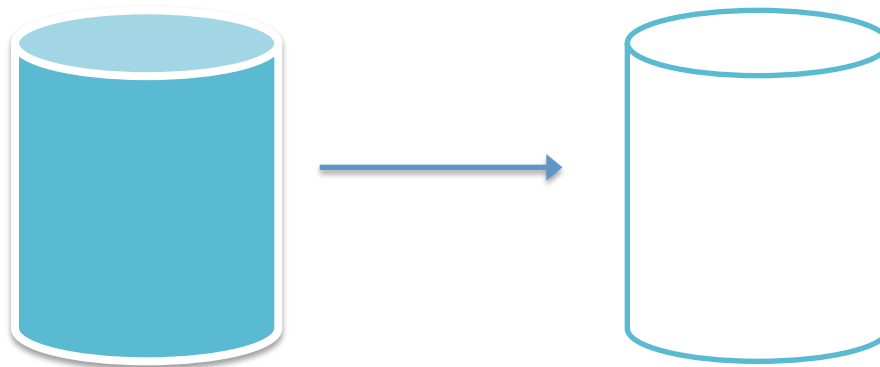
| PDB_NAME | LOGGING        | FORCE_LOGGING | FORCE_NOLOGGING |
|----------|----------------|---------------|-----------------|
| PDB1     | <b>LOGGING</b> | NO            | NO              |



# PDB Metadata Clone

**NO DATA** clause of the **CREATE PLUGGABLE DATABASE**

*“This clause is useful for quickly creating clones of a PDB with only the object definitions and no data.”*



# PDB Remote Clone

## IN SOURCE DB:

```
**alter pluggable database pdbsource open read only;  
  
create user C##dgomez identified by manager1;  
  
grant create session to C##dgomez;
```



# PDB Remote Clone

## IN TARGET DB:

```
create user C##dgomez identified by manager1;
```

```
grant create session, create database link, create pluggable database to C##dgomez;
```

```
PDBSOURCE =  
(DESCRIPTION =  
(ADDRESS = (PROTOCOL = TCP) (HOST = db12102.oraworld.com) (PORT =  
1521))  
(CONNECT_DATA =  
(SERVER = DEDICATED) (SERVICE_NAME = pdbsource)  
))
```

```
create database link dblinktosource connect to C##dgomez identified  
by manager1 using 'PDBSOURCE';
```

```
create pluggable database pdbtarget3 from pdbsource@dblinktosource;
```



# PDB Remote Clone

**Target:** 12.1.0.2

**Source:** 12.1.0.1

```
SQL> create pluggable database pdbtarget from  
pdbsource@dblinktosource;  
*
```

ERROR at line 1:

```
ORA-17627: ORA-00600: internal error code, arguments:  
[ksrpcsexec_1], [158],  
[20], [], [], [], [], [], [], [], [], []  
ORA-17627: ORA-00600: internal error code, arguments:  
[ksrpcsexec_1], [158],  
[20], [], [], [], [], [], [], [], [], []
```



# PDB Remote Clone

**Target: 12.1.0.1**

**Source: 12.1.0.1**

```
SQL> create pluggable database pdbtarget from  
sourcepdb@dblinktosource
```

```
*
```

**ERROR at line 1:**

**ORA-17628: Oracle error 19505 returned by remote  
Oracle server**

**ORA-19505: failed to identify file ""**



# PDB Remote Clone

**Target: 12.1.0.2**

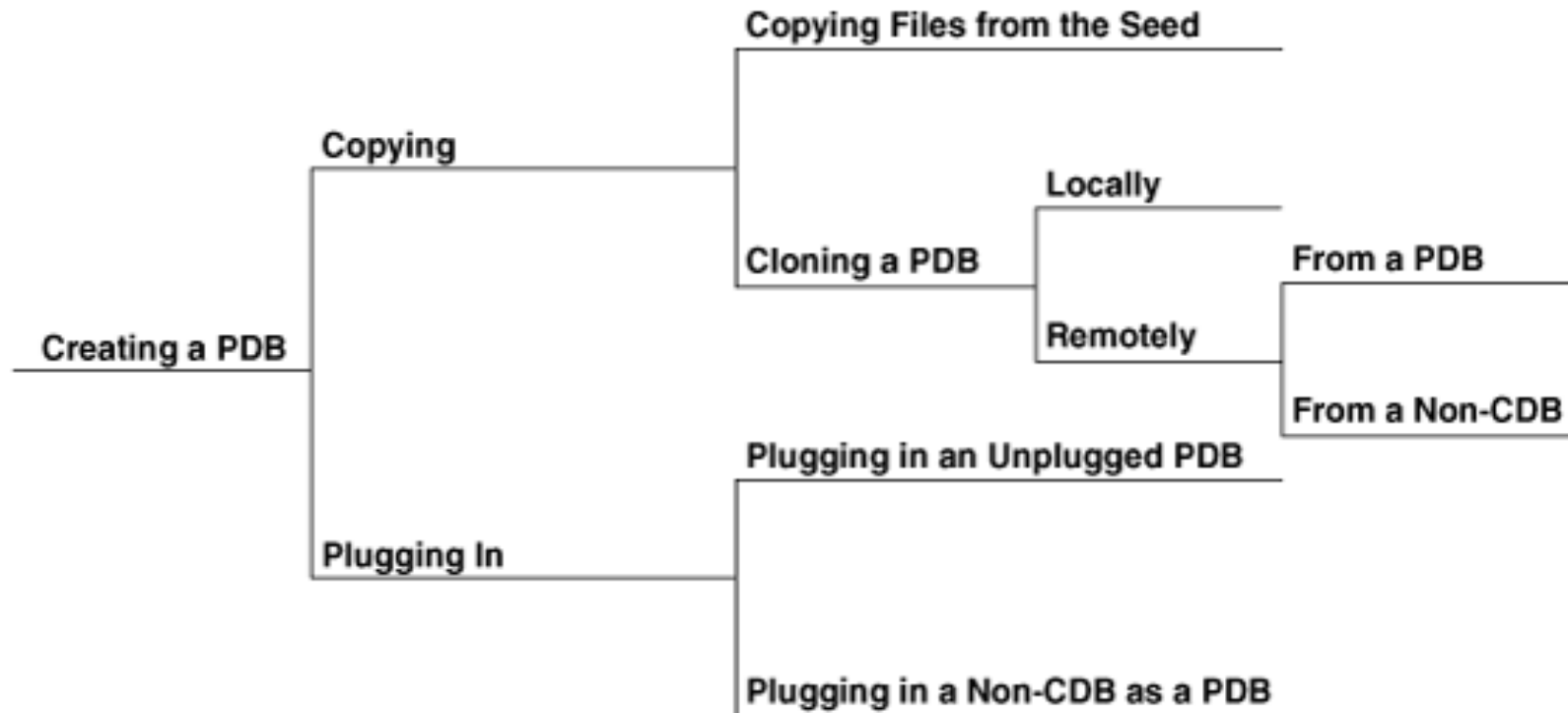
**Source: 12.1.0.2**

```
SQL> create pluggable database pdbtarget from  
pdbname@dblinktosource;
```

**Pluggable database created.**



# PDB Cloning flavors





# Create PDB from Non-CDB

DBMS\_PDB.DESCRIBE

**CREATE PLUGGABLE DATABASE** salespdb **USING** '/disk1/  
usr/salespdb.xml'  
**NOCOPY;**

- NOCOPY
- COPY
- MOVE

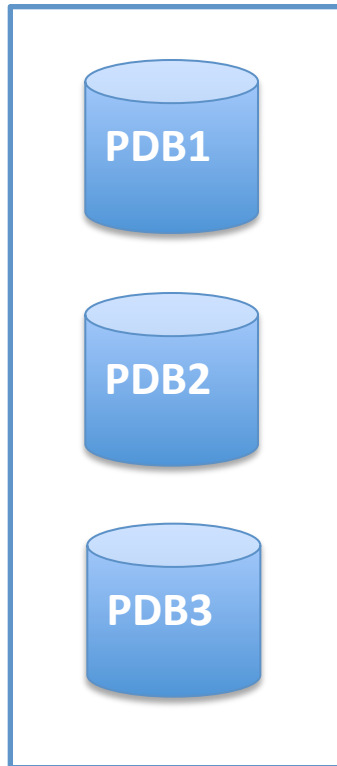
@\$ORACLE\_HOME/rdbms/admin/**noncdb\_to\_pdb.sql**

**noncdb\_to\_pdb.sql not re-runable!**

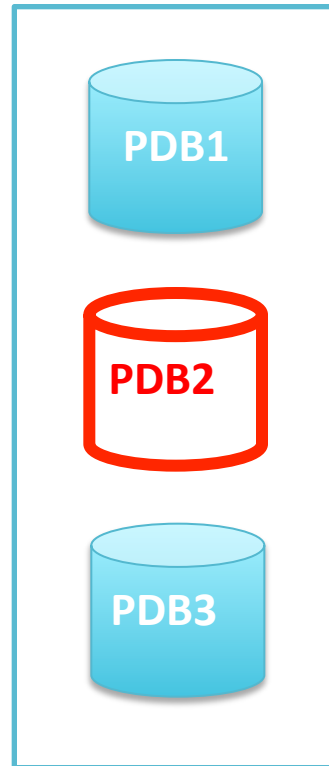


# PDB STANDBYS Clause

## Primary



## Standby



- Datafiles Offline
- Unnamed



# PDB STANDBYS Clause

*“The **STANDBYS** clause of the **CREATE PLUGGABLE DATABASE** statement specifies whether the new PDB is included in standby CDBs”*

- **STANDBYS=ALL** includes the new PDB in all of the standby CDBs
- .
- **STANDBYS=NONE** excludes the new PDB from all of the standby CDBs.



# PDB STANDBYS Clause

*“It is possible to enable a PDB on a standby CDB after it was excluded on that standby CDB.”*

- PDB data files need to be **copied to the standby** database from the primary database
- **The control file** needs to be updated to reflect their paths
- Execute `ALTER PLUGGABLE DATABASE ENABLE RECOVERY`
  - ◆ It automatically online all of the data files belonging to the PDB.



# PDB State Management Across CDB Restart

## 12.1.0.1

After opening a CDB, the PDBs are by default kept in mounted mode. If you want the PDBs opened automatically AFTER STARTUP ON DATABASE, you can create an database event trigger that opens all PDBs after STARTUP.

```
CREATE TRIGGER Open_All_PDBs
  after startup on database
begin
  execute immediate 'alter pluggable database all open';
end Open_All_PDBs;
/
```

Source: "Oracle Database 12c: New Features for Administrators" by Oracle University



# PDB State Management Across CDB Restart

## 12.1.0.2

1. Set the PDB at the State you want to save.

2. SQL> alter pluggable database pdb1 **save state**;

Pluggable database altered.

**or**

SQL> alter pluggable database pdb1 **discard state**;

Pluggable database altered.



# PDB State Management Across CDB Restart

## 12.1.0.2

```
SQL> ALTER PLUGGABLE DATABASE ALL EXCEPT PDB1  
SAVE STATE;
```

Pluggable database altered.

```
SQL> ALTER PLUGGABLE DATABASE ALL SAVE STATE;
```

Pluggable database altered.



# PDB Subset Cloning

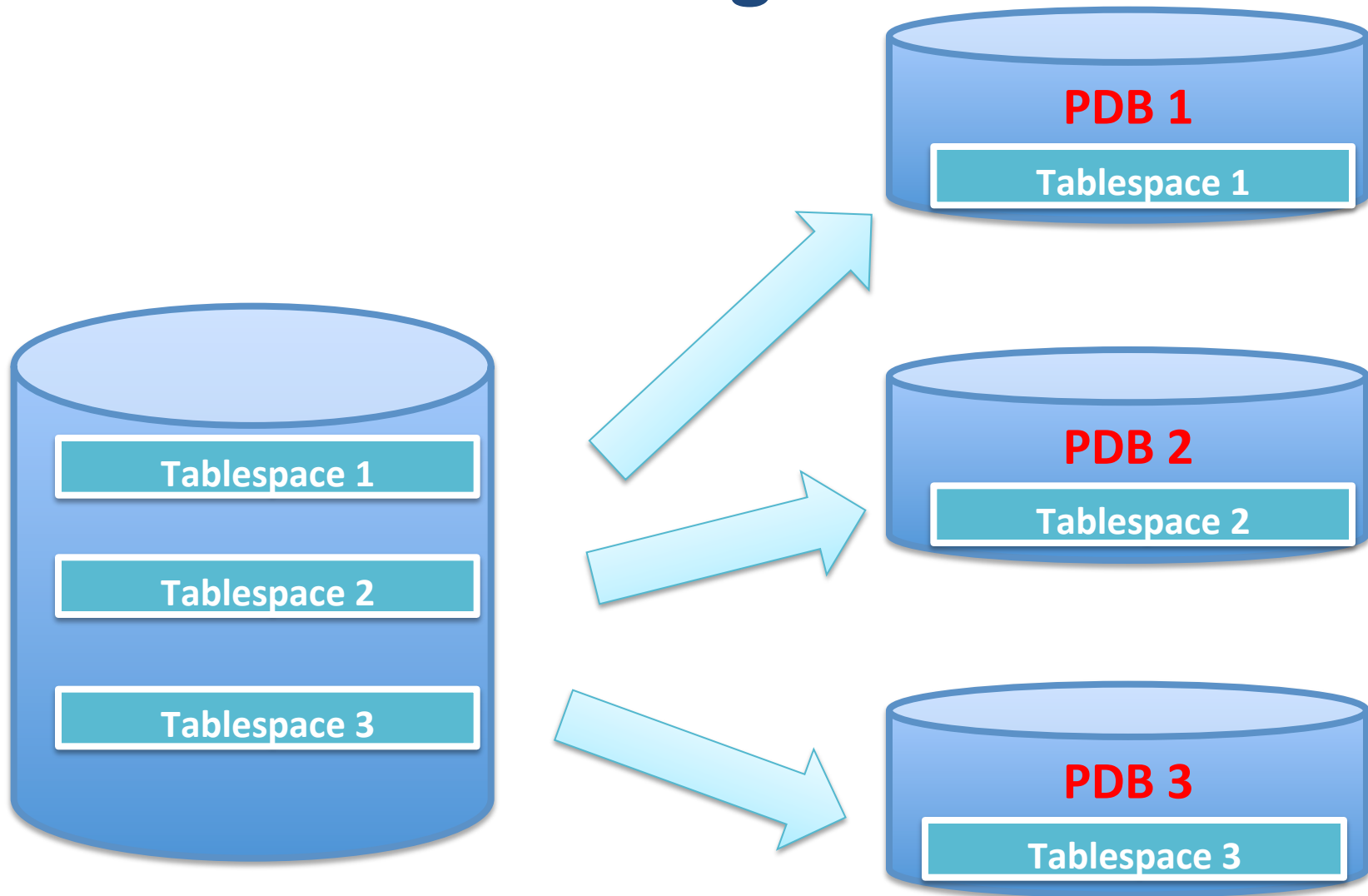
*“The **USER\_TABLESPACES** clause of the CREATE PLUGGABLE DATABASE statement specifies which tablespaces are available in the new PDB”*

- **List** one or more tablespaces to include.
- Specify **ALL**, the default, to include all of the tablespaces.
- Specify **ALL EXCEPT** to include all of the tablespaces, except for the tablespaces listed.
- Specify **NONE** to exclude all of the tablespaces.





# PDB Subset Cloning



# Duplicate with Backupsets

```
RMAN> DUPLICATE TARGET DATABASE TO 'DB2'  
FROM ACTIVE DATABASE  
USING BACKUPSET;
```

```
RMAN> DUPLICATE TARGET DATABASE  
TO 'DB2'  
FROM ACTIVE DATABASE  
USING COMPRESSED BACKUPSET;
```





**Deiby Gómez**

[dgomez@nuvolacg.com](mailto:dgomez@nuvolacg.com)

Twitter: @hdeiby

[www.nuvolacg.com](http://www.nuvolacg.com)