# PDB Cloning and Relocation using DBCA

By Ahmed Baraka

# Objectives

By the end of this lecture, you should be able to:

- Use dbca utility to create PDBs with the following methods:
  - Cloning from the Seed container
  - Cloning from a local PDB
  - Cloning from a remote PDB
  - Plugging in an unplugged PDB
  - Restoring from RMAN Backupset and PDB Metadata xml file

# Options for Creating PDBs using CREATE PLUGGABLE DATABASE command

| Category | Option | Description |
|---|---|---|
| Copying | **From the seed** | By using the files of the CDB seed or application seed. |
| | **Clone a local PDB** | By cloning a source local PDB. |
| | **Clone a remote PDB** | By cloning a remote PDB. |
| | **Clone a non-CDB** | By cloning a non-CDB. |
| Plugging In | **Plugging in an Unplugged PDB** | By plugging in an unplugged PDB |
| | **Adopting a Non-CDB as a PDB (Using the DBMS_PDB)** | By creating an unplugged PDB from a non-CDB and then plug the unplugged PDB into the CDB. |
| Relocating | **Relocating a PDB** | By moving the files associated with the PDB to a new location. |
| Proxy PDB | **Creating a PDB as a proxy PDB** | By referencing a different PDB with a database link |

# Options for Creating PDBs using DBCA

| Category | Option | Description |
|---|---|---|
| Copying | **From the seed** | By using the files of the CDB seed or application seed. |
| | **Clone a local PDB** | By cloning a source local PDB. |
| | **Clone a remote PDB** | By cloning a remote PDB. |
| | ~~**Clone a non-CDB**~~ | ~~By cloning a non-CDB.~~ |
| Plugging In | **Plugging in an Unplugged PDB** | By plugging in an unplugged PDB |
| | **Adopting a Non-CDB as a PDB (Using the DBMS_PDB)** | By creating an unplugged PDB from a non-CDB and then plug the unplugged PDB into the CDB. |
| Relocating | **Relocating a PDB** | By moving the files associated with the PDB to a new location. |
| ~~Proxy PDB~~ | ~~**Creating a PDB as a proxy PDB**~~ | ~~By referencing a different PDB with a database link~~ |

# About Creating PDBs using DBCA

- Implemented by issuing the command:

```
dbca -silent -createpluggabledatabase ...
```

- The other parameters set which option dbca uses
- Implemented in silent mode
- The cloned PDB will automatically open in read/write
- Reference:
  - Oracle Database, Database Administrator's Guide

# Cloned PDB Datafiles Destination

- If the OMF is enabled, the datafiles will be saved under the OMF (subdirectories will automatically be created)
- If the OMF is not enabled (based on personal testing):
  - the datafiles will automatically be created in $ORACLE_BASE/oradata/$ORACLE_SID/<pdb>
  - To save the datafiles in specific location use the parameter **`pdbDatafileDestination`**
  - Alternatively, use the parameter **`-fileNameConvert`** (not working)

# dbca Impact on the To-be-cloned PDB State

- When cloning from a local PDB or from a remote PDB, the to-be-cloned PDB remains open during cloning if:
  - The CDB is running in archivelog mode
  - The Local Undo option is enabled

  otherwise, dbca closes the source PDB during the clone operation, and after receiving confirmation, opens the source PDB in read-only mode.

# Cloning from the Seed PDB using DBCA

- Example:

```
dbca -silent -createpluggabledatabase
-sourcedb oradb -createpdbfrom DEFAULT -pdbName pdb2
-pdbAdminUserName pdb2admin -pdbAdminPassword ***
```

| Parameter | Description |
|---|---|
| **sourcedb** | Oracle database SID |
| **createpdbfrom** | For this option is must be **DEFAULT** |
| **pdbName** | Name of the cloned PDB |
| **pdbAdminUserName** | Cloned PDB admin name |
| **pdbAdminPassword** | Cloned PDB admin password |

# Cloning from a Local PDB using DBCA

- Syntax:

```
dbca -silent -createpluggabledatabase
-sourcedb oradb -createpdbfrom PDB -pdbName pdb2 -sourcepdb pdb1
```

| Parameter | Description |
|---|---|
| **sourcedb** | Oracle database SID |
| **createpdbfrom** | For this option is must be **PDB** |
| **pdbName** | Name of the cloned PDB |
| **sourcepdb** | Name of the source PDB |

# Cloning from a Remote PDB using DBCA

- Syntax:

```
dbca -silent -createPluggableDatabase -createFromRemotePDB
-sourceDB oradb2
-remotePDBName pdb1
-remoteDBConnString srv1:1521/oradb.localdomain
-remoteDBSYSDBAUserName SYS
-remoteDBSYSDBAUserPassword oracle
-dbLinkUsername c##ruser
-dbLinkUserPassword ****
-sysDBAUserName SYS
-sysDBAPassword ****
-pdbName pdb2
```

# Cloning from a Remote PDB using DBCA

| Parameter | Description |
|---|---|
| `createFromRemotePDB` | Create a PDB by cloning a remote PDB. |
| `sourceDB` | Database name of the local PDB |
| `remotePDBName` | To-be-cloned PDB name in the remote database |
| `remoteDBConnString` | Database connection string of the remote PDB |
| `remoteDBSYSDBAUserName` | Username in the remote database with sysdba privilege |
| `remoteDBSYSDBAUserPassword` | Password of the username in the preceding line |
| `dbLinkUsername` | Username with the required privileges in the remote database |
| `dbLinkUserPassword` | Password of the username in the preceding line |
| `sysDBAUserName` | Username in the local database with sysdba privilege |
| `sysDBAPassword` | Password of the username in the preceding line |
| `pdbName` | PDB name of the new PDB |

# Plugging in a PDB using DBCA

1. Close the PDB
2. Unplug the PDB:

```
dbca -silent -unplugDatabase -sourceDB oradb -pdbName pdb1
-archiveType TAR -pdbArchiveFile '/u02/staging/pdb1.arc'
```

3. Plug in the PDB (into the same or different CDB):

```
dbca -silent -createPluggableDatabase -createPDBFrom FILEARCHIVE
-sourceDB oradb2 -pdbName pdb2 -pdbArchiveFile
'/u02/staging/pdb1.arc' -workArea '/media/sf_staging/'
```

# Plugging in a PDB using DBCA

| Parameter | Description |
|---|---|
| **createPDBFrom** | Must be set to **FILEARCHIVE** |
| **sourceDB** | Database name of the **local** PDB |
| **pdbName** | PDB name of the new PDB |
| **pdbArchiveFile** | The fullpath name of the archive file |
| **workArea** | The fullpath directory which will be used for extracting the archive file |

# Using DBCA to Create a PDB from RMAN Backup

1. Generate the PDB metadata XML file:

```
ALTER SESSION SET CONTAINER=PDB1;
EXEC DBMS_PDB.DESCRIBE( PDB_DESCR_FILE => '/.../pdb1.xml')
```

2. Create the PDB from the RMAN backupset file:

```
dbca -silent -createPluggableDatabase  -createPDBFrom RMANBACKUP
-pdbBackUpfile '/media/sf_staging/pdb1_0309cpdp.bak'
-pdbMetadataFile '/media/sf_staging/pdb1.xml'
-pdbName pdb2
-sourceDB oradb2
```

# Summary

By the end of this lecture, you should have learnt how to:

- Use dbca utility to create PDBs with the following methods:
  - Cloning from the Seed container
  - Cloning from a local PDB
  - Cloning from a remote PDB
  - Plugging in an unplugged PDB
  - Restoring from RMAN Backupset and PDB Metadata xml file