# Creating PDBs

# Objectives

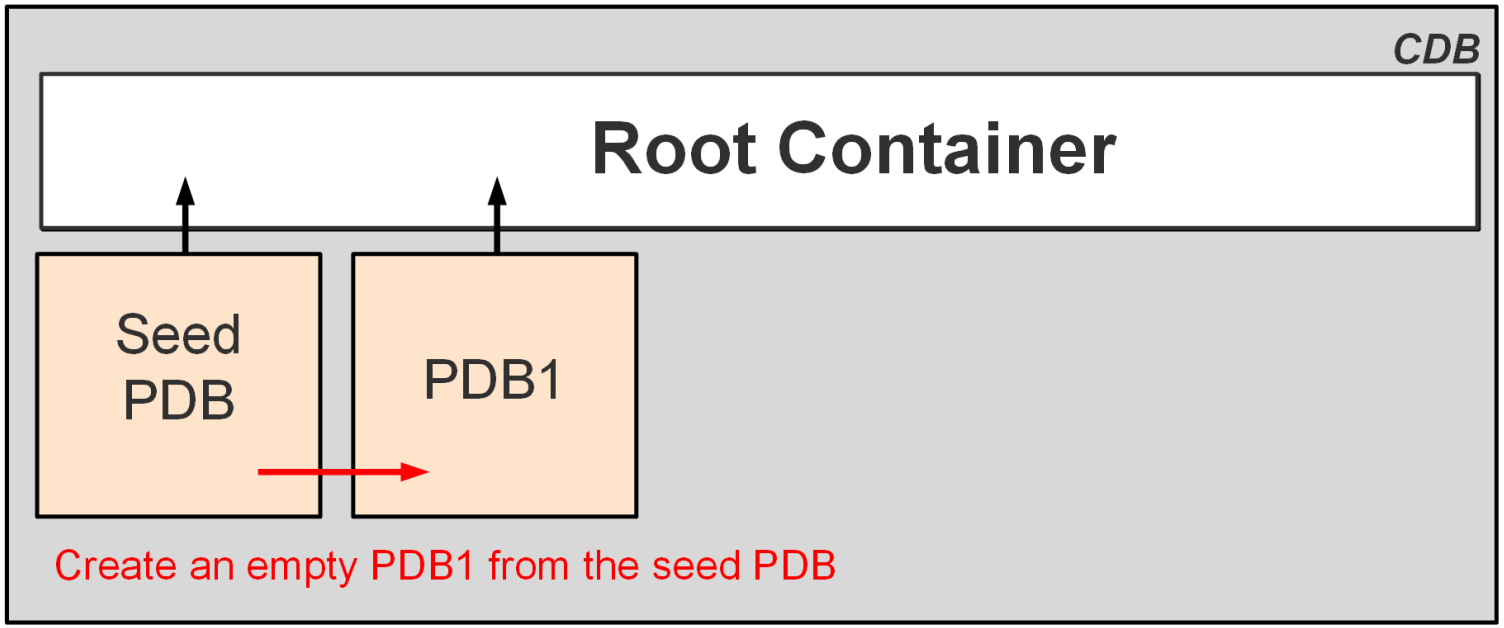
- After completing this lesson, you should be able to:
  - Describe the methods and tools used to create PDBs
  - Create PDBs from seed by using SQL*Plus
  - Clone PDBs by using SQL*Plus
  - Unplug and plug in PDBs by using SQL*Plus
  - Drop PDBs by using SQL*Plus

# Methods and Tools to Create PDBs

- Methods to create PDBs:
  - Create a PDB by using the seed
  - Create a PDB from a non-CDB
  - Clone an existing PDB or non-CDB
  - Plug an unplugged PDB into a different CDB
  - Relocate a PDB to a different CDB
  - Create a PDB as a proxy PDB
- Tools to create PDBs:
  - SQL*Plus
  - SQL Developer
  - Enterprise Manager Cloud Control
  - DBCA—Create a PDB from seed or by using the unplug/plug method.

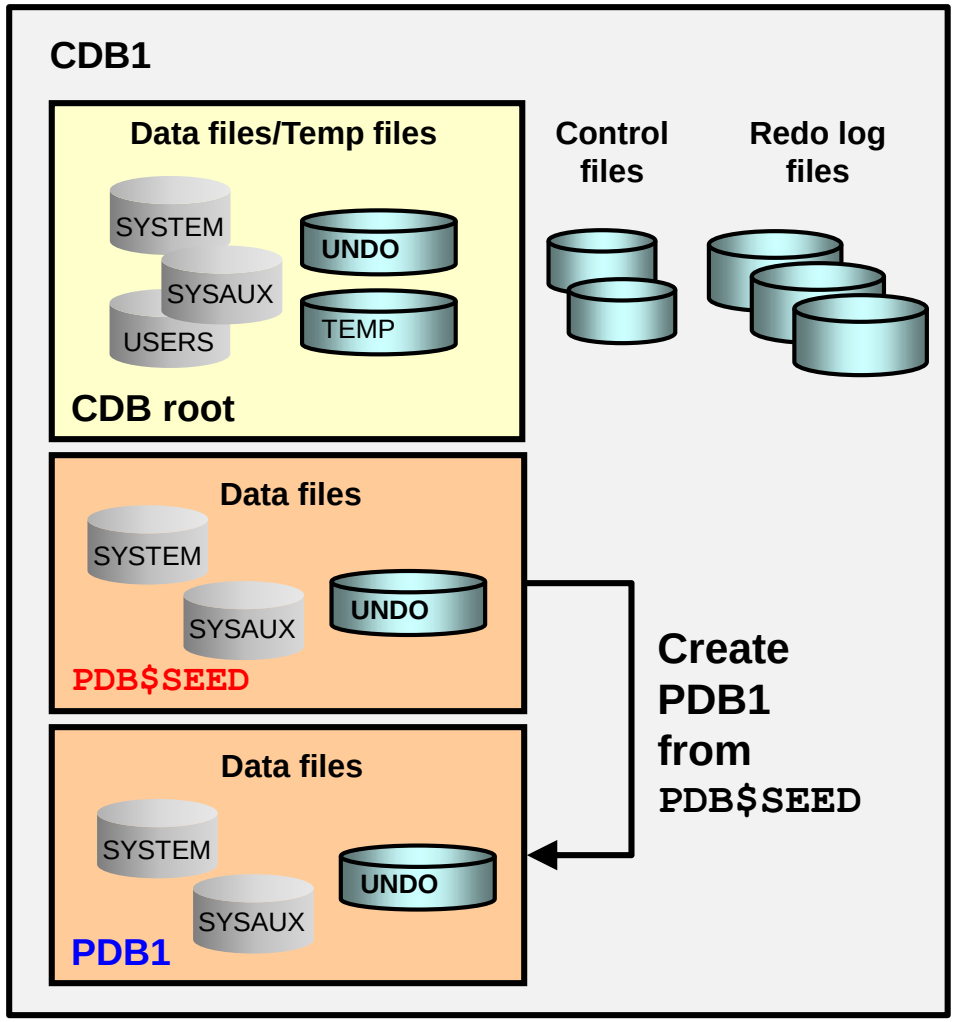
# Creating PDBs from Seed

- You can create a new empty PDB by using the seed PDB as a template.
- Every CDB has a seed PDB.
- To create a PDB from seed with SQL*Plus, use the `CREATE PLUGGABLE DATABASE` statement.



# Creating a New PDB from PDB$SEED

- Copies the data files from `PDB$SEED` data files
- Creates the `SYSTEM`, `SYSAUX`, and `UNDO` tablespaces
- Creates a full catalog including metadata pointing to Oracle-supplied objects
- Creates common users:
  - `SYS`
  - `SYSTEM`
- Creates a local user (PDBA), granted local `PDB_DBA` role
- Creates a new default service



# Examples: Creating a PDB from Seed

- Create a new PDB from the seed by using `FILE_NAME_CONVERT`:

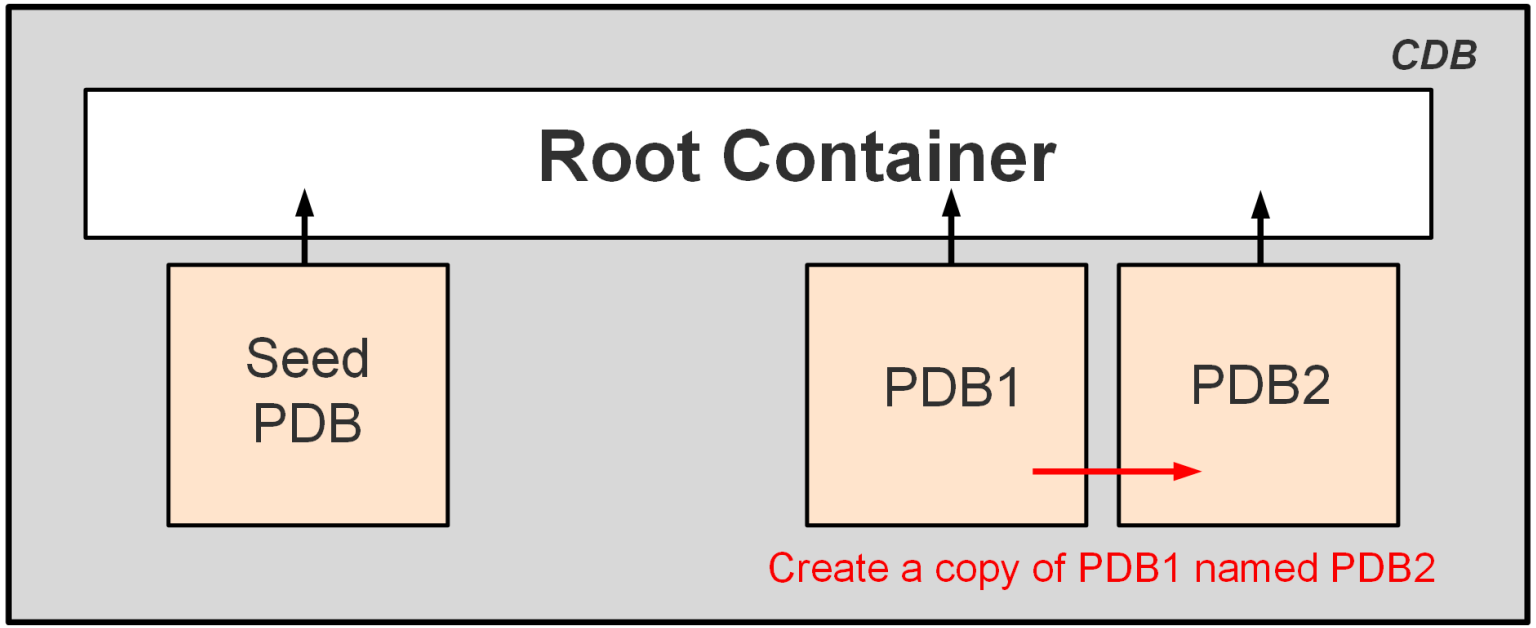
```
SQL> CREATE PLUGGABLE DATABASE pdb1
     ADMIN USER admin1 IDENTIFIED BY p1 ROLES=(CONNECT)
     FILE_NAME_CONVERT = ('PDB$SEEDdir', 'PDB1dir');
```

- Set the `PDB_FILE_NAME_CONVERT` initialization parameter and then create the PDB:

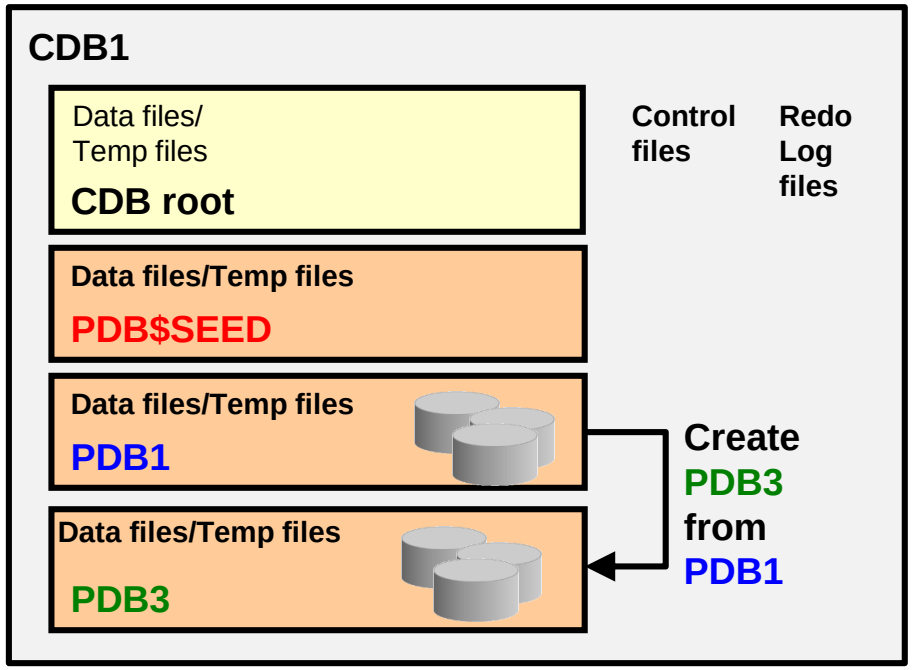
```
SQL> CREATE PLUGGABLE DATABASE pdb1
     ADMIN USER pdb1_admin IDENTIFIED BY p1 ROLES=(CONNECT);
```

# Cloning PDBs

- Cloning is copying a source PDB from a CDB and plugging the copy into the same CDB or another CDB.
- Example: PDB1 is cloned as PDB2 in the same CDB. The seed PDB, while present in the CDB, is not used.



# Cloning Regular PDBs




**PDB3** owns:

- SYSTEM, SYSAUX, UNDO tablespaces
- Full catalog
- SYS, SYSTEM common users
- Same local administrator name
- New service name

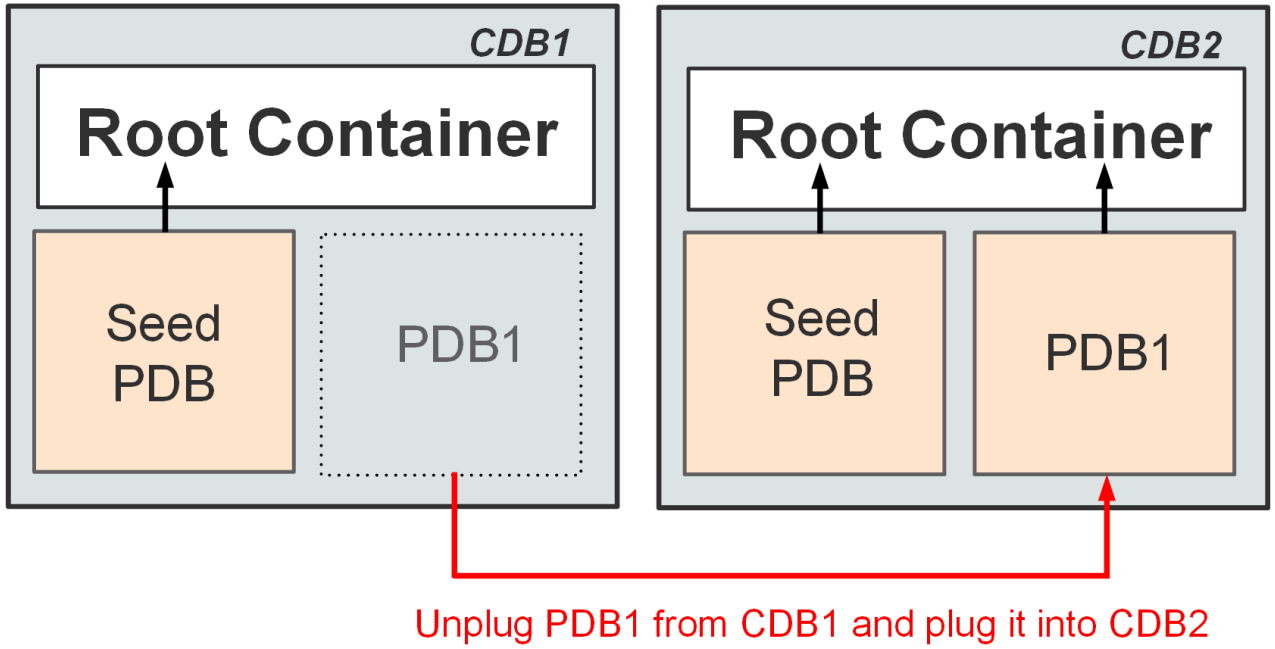
1. Define how Oracle will find the location of the data files:
   - In `init.ora`, set `DB_CREATE_FILE_DEST= 'PDB3dir'`
   - In `init.ora`, set `PDB_FILE_NAME_CONVERT='PDB1dir', 'PDB3dir'`
   - Using the `CREATE_FILE_DEST= 'PDB3dir'` clause
2. Connect to the CDB root to close **PDB1**.
3. Clone **PDB3** from **PDB1**.
4.
```
SQL> CREATE PLUGGABLE DATABASE pdb3 FROM pdb1
          CREATE_FILE_DEST = 'PDB3dir';
```

```
SQL> ALTER PLUGGABLE DATABASE pdb3 OPEN;
```

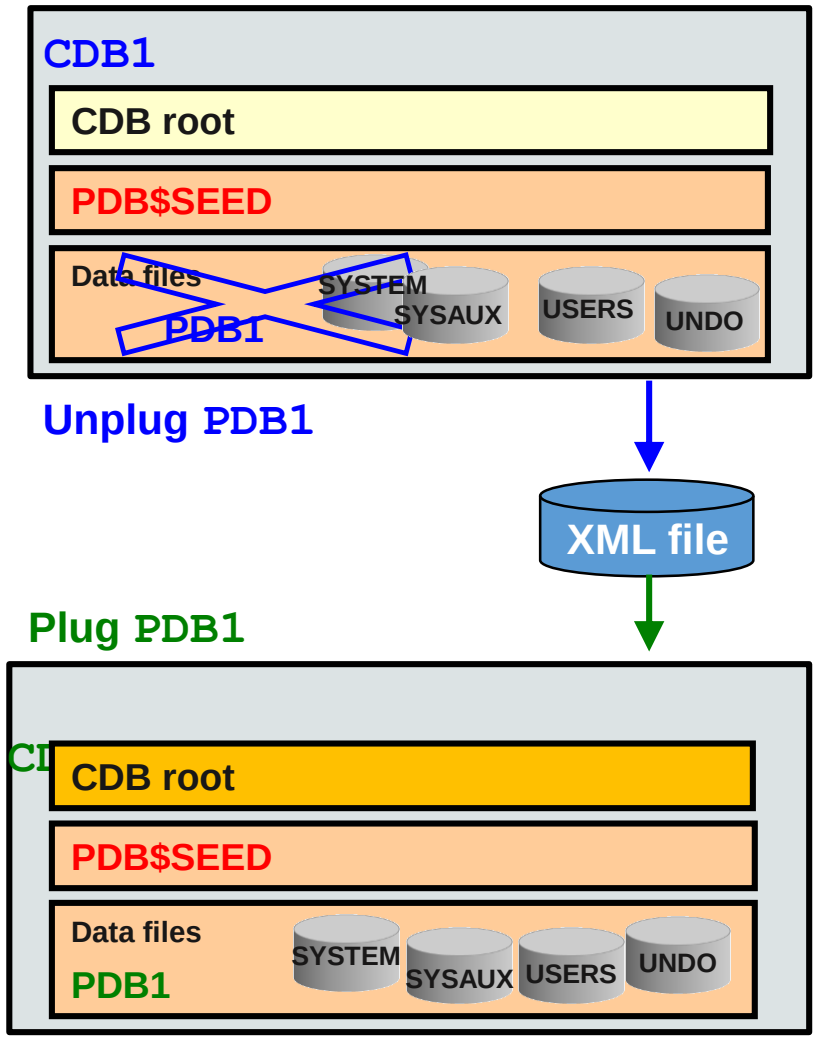
**Note**: Cloning metadata only with `NO DATA`

# Unplugging and Plugging in PDBs

- Unplugging a PDB is disassociating the PDB from its CDB.
- Plugging in a PDB is associating a PDB with a CDB.
- You can plug a PDB into the same or another CDB.
- Example: PDB1 is unplugged from CDB1 and plugged into CDB2.



# Plugging an Unplugged Regular PDB into a CDB




Unplug **PDB1** from **CDB1**:

1. Connect to **CDB1** as a common user.
2. Verify that **PDB1** is closed.

```
SQL> ALTER PLUGGABLE DATABASE pdb1
             UNPLUG INTO 'xmlfile1';
```

3. Drop **PDB1** from **CDB1** .

Plug **PDB1** into **CDB2**:

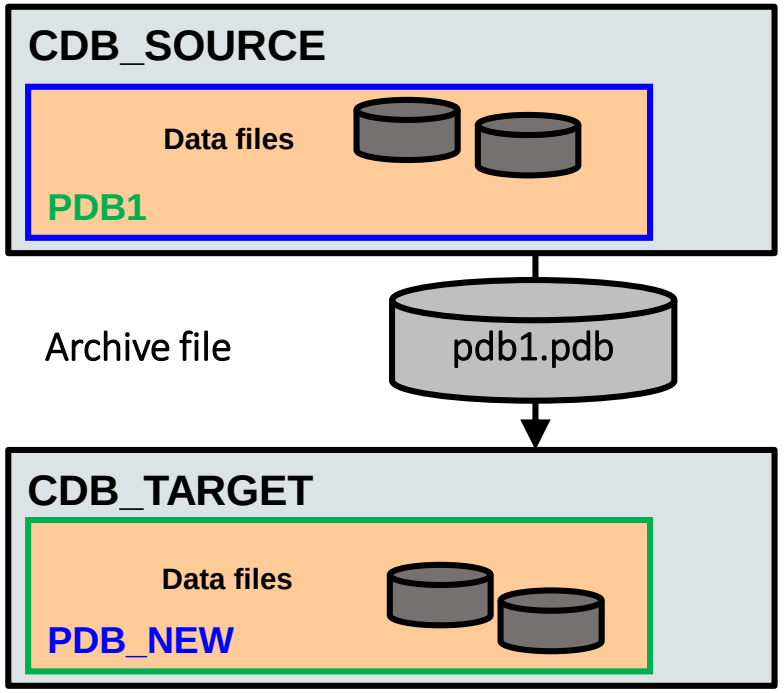
1. Connect to **CDB2** as a common user.
2. Use the `DBMS_PDB` package to check the compatibility of **PDB1** with **CDB2**.

```
SQL> CREATE PLUGGABLE DATABASE pdb1
             USING 'xmlfile1' NOCOPY;
```

3. Open **PDB1** in read/write mode.

# Plugging Using an Archive File

- Unplugging a PDB into a single archive file includes:
  - XML file
  - Data files
- Plugging the PDB requires only the archive file.




```
SQL> ALTER PLUGGABLE DATABASE pdb1
         UNPLUG INTO '/tmp/pdb1.pdb';
```

```
SQL> CREATE PLUGGABLE DATABASE pdb_new
         USING '/tmp/pdb1.pdb';
```

# Dropping PDBs

- When you drop a PDB, you remove all references to it and its data files in the control file of the CDB.
- Archived logs and backups associated with the dropped PDB are not deleted in case you later want to recover the PDB.
- You can also use Oracle Recovery Manager (RMAN) to delete archived logs and backups.
- You use the `DROP PLUGGABLE DATABASE` statement to drop a pluggable database (PDB).
  - Example:

```
SQL> DROP PLUGGABLE DATABASE SALESPDB INCLUDING DATAFILES;
```

# Summary

- In this lesson, you should have learned how to:
  - Describe the methods and tools used to create PDBs
  - Create PDBs from seed by using SQL*Plus
  - Clone PDBs by using SQL*Plus
  - Unplug and plug in PDBs by using SQL*Plus
  - Drop PDBs by using SQL*Plus

# Practice 10: Overview

- 10-1: Creating a PDB from Seed
- 10-2: Cloning a PDB
- 10-3: Unplugging and Plugging in a PDB
- 10-4: Dropping a PDB