8

Multitenant Architecture

Oracle for Base

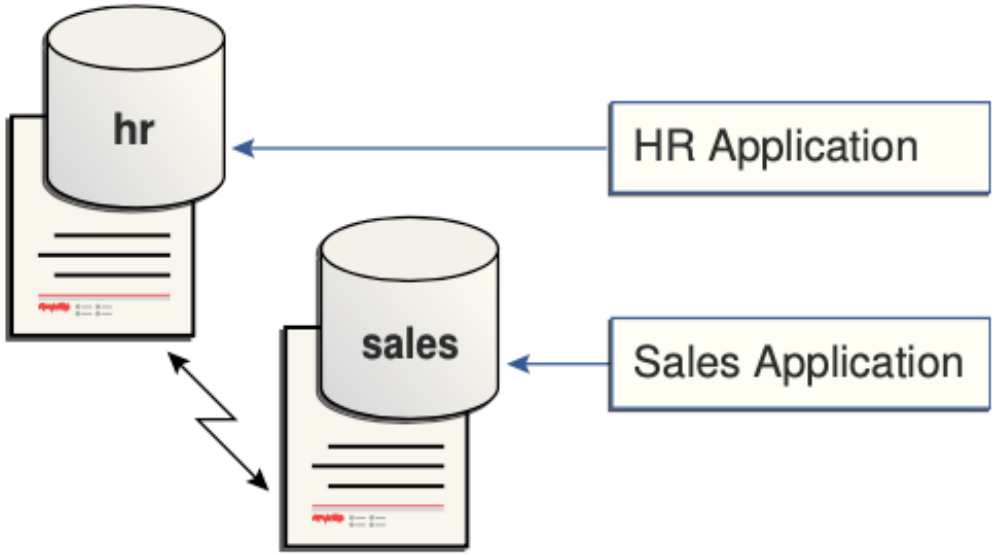
Multitenant Architecture

- Starting in Oracle Database 12*c*, the **multitenant architecture** enables an Oracle database to be a multitenant container database (CDB).
- A **CDB** is a single physical database that contains zero, one, or many user-created pluggable databases. A **pluggable database (PDB)** is a portable collection of schemas, schema objects, and nonschema objects that appears to an **Oracle Net** client as a non- CDB. A **non-CDB** is a traditional Oracle database that cannot contain PDBs.
- Starting in Oracle Database 12*c*, you must create a database as either a CDB or non- CDB. You can plug a non-CDB into a CDB as a PDB. To move a PDB to a non-CDB, you must use Oracle Data Pump.

Multitenant Architecture

- **Figure 1-2:** Shows two separate non-CDBs: hr and sales. Each non-CDB has its own memory and set of database files, and resides on its own computer. Each non-CDB has its own dedicated user application.

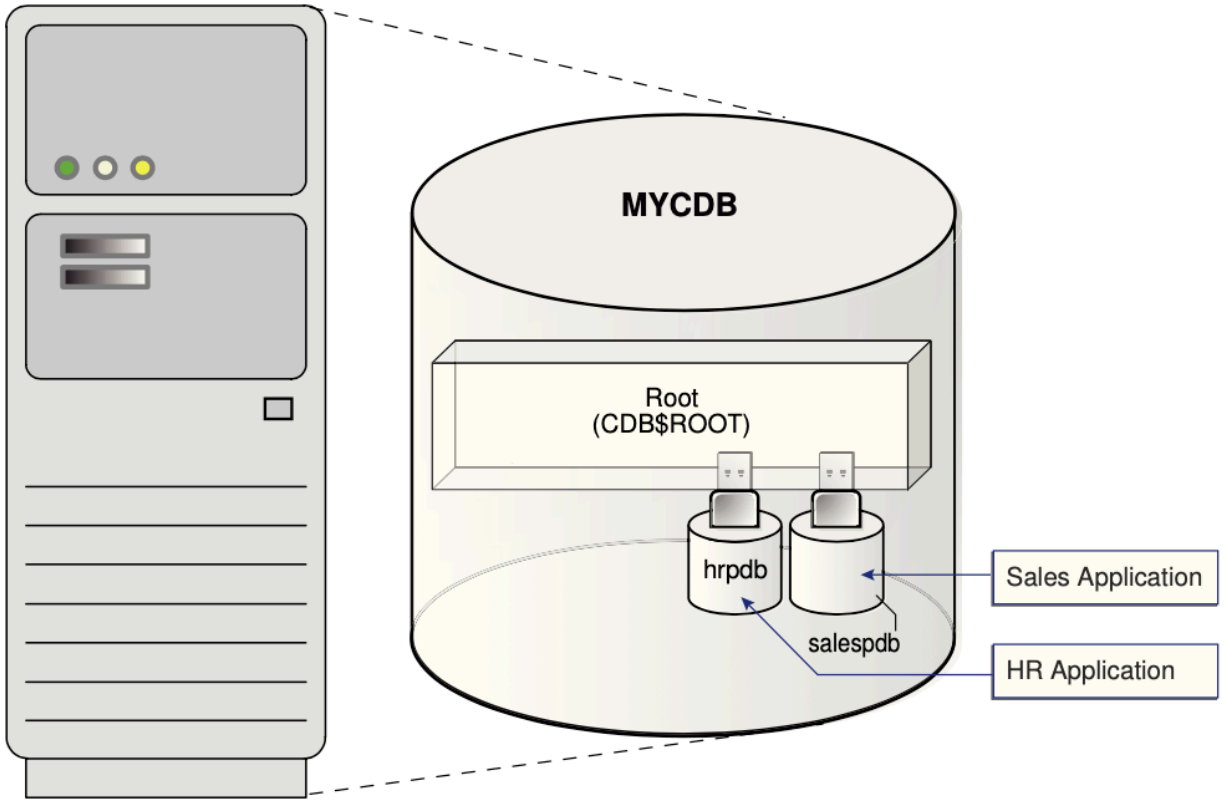
Figure 1-2 Non-CDBs



Multitenant Architecture

- **Figure 1-3**: Shows the same data after being consolidated into the CDB named MYCDB.

Figure 1-3 CDB



Multitenant Architecture

- Physically, MYCDB is an Oracle database. MYCDB has one database instance (although multiple instances are possible in Oracle Real Application Clusters) and one set of database files, just like a non-CDB.
- MYCDB contains two PDBs: hrpdb and salespdb. As shown in Figure 1-3 these PDBs appear to their respective applications just as they did before database consolidation. To administer the CDB itself or any PDB within it, a CDB administrator can connect to the **CDB root**, which is a collection of schemas, schema objects, and nonschema objects to which all PDBs belong.
- CDBs and non-CDBs have architectural differences. This manual assumes the architecture of a non-CDB unless otherwise indicated.

About Containers in a CDB

- A **container** is either a PDB or the root. The **root container** is a collection of schemas, schema objects, and nonschema objects to which all PDBs belong. Every CDB has the following containers:
  - **Exactly one root**: The root stores Oracle-supplied metadata and common users. An example of metadata is the source code for Oracle-supplied PL/SQL packages. A common user is a database user known in every container. The root container is named CDB $ROOT.
    - Every CDB has one and only one root container, named CDB$ROOT, which stores the system metadata required to manage PDBs. All PDBs belong to the root.
    - The root does not store user data.

About Containers in a CDB

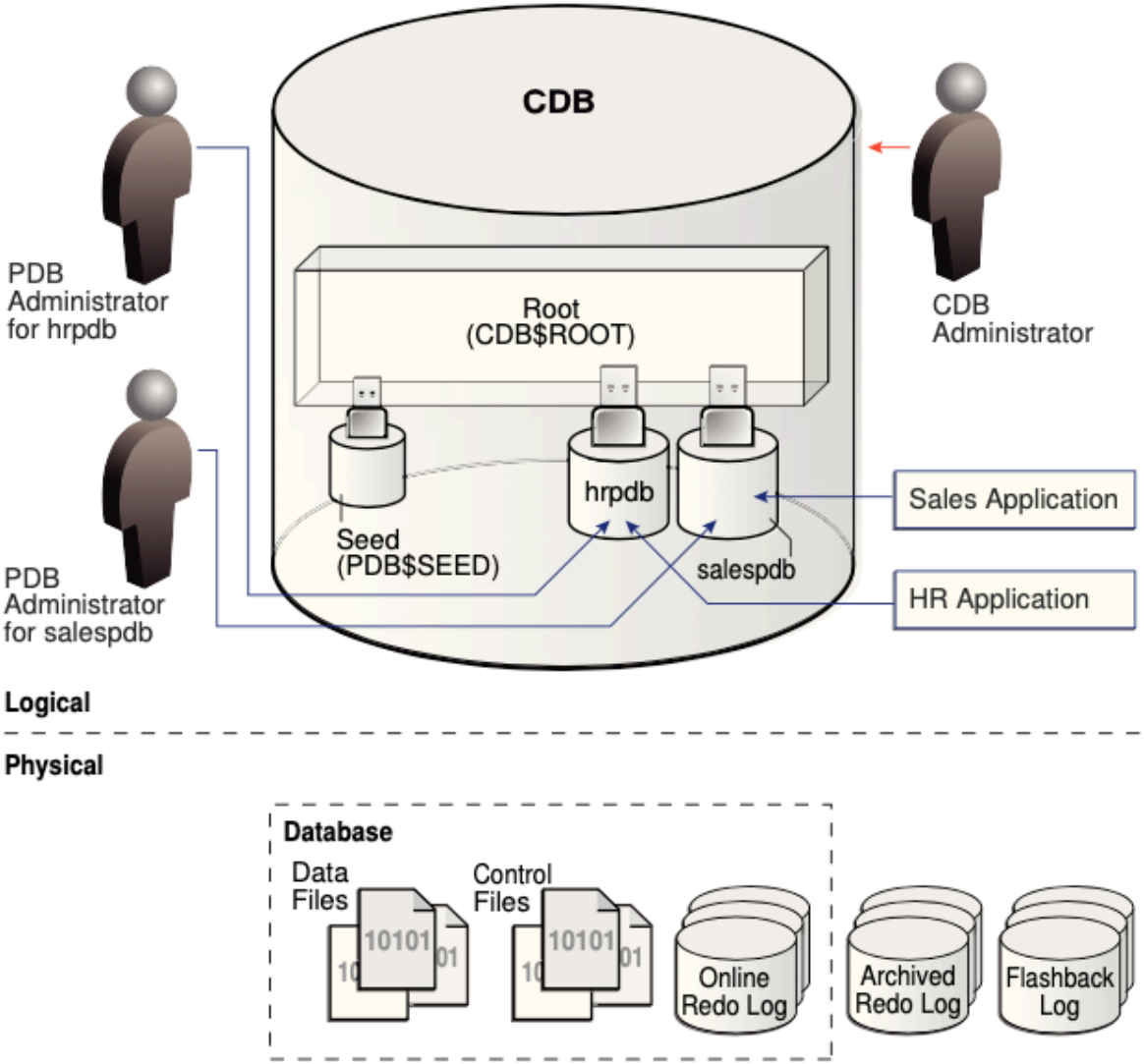
- A **container** is either a PDB or the root. The **root container** is a collection of schemas, schema objects, and nonschema objects to which all PDBs belong. Every CDB has the following containers:
  - Exactly one **seed PDB**: The seed PDB is a system-supplied template that the CDB can use to create new PDBs. The seed PDB is named PDB$SEED. You cannot add or modify objects in PDB$SEED.
  - Zero or more user-created PDBs : A PDB is a user-created entity that contains the data and code required for a specific set of features. For example, a PDB can support a specific application, such as a human resources or sales application. No PDBs exist at creation of the CDB. You add PDBs based on your business requirements.

About Containers in a CDB

- A **PDB** is a user-created set of schemas, objects, and related structures that appears logically to an application as a separate database. Every PDB is owned by SYS, which is a common user in the CDB

- The following figure shows a CDB with four containers: the root, seed, and two PDBs. Each PDB has its own dedicated application. A different PDB administrator manages each PDB. A **common user** exists across a CDB with a single identity. In this example, common user SYS can manage the root and every PDB. At the physical level, this CDB has a database instance and database files, just as a non-CDB does.

About Containers in a CDB

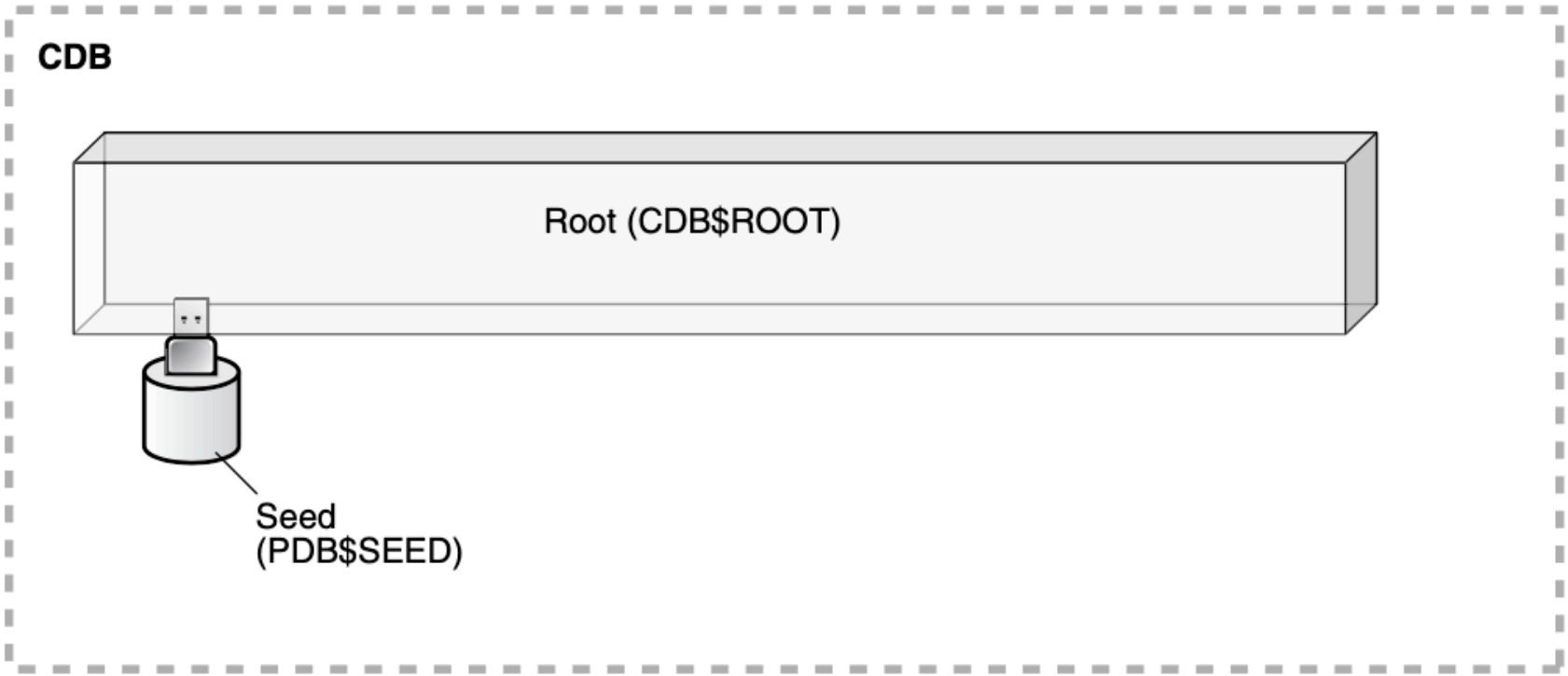


Creation of a CDB

- The CREATE DATABASE ... ENABLE PLUGGABLE DATABASE SQL statement creates a new CDB. If you do not specify the ENABLE PLUGGABLE DATABASE clause, then the newly created database is a non-CDB and can never contain PDBs.
- Along with the root container (CDB$ROOT), Oracle Database automatically creates a seed PDB (PDB$SEED). The following graphic shows a newly created CDB:

Creation of a CDB

Figure 17-4 CDB with Seed PDB



Creation of a CDB

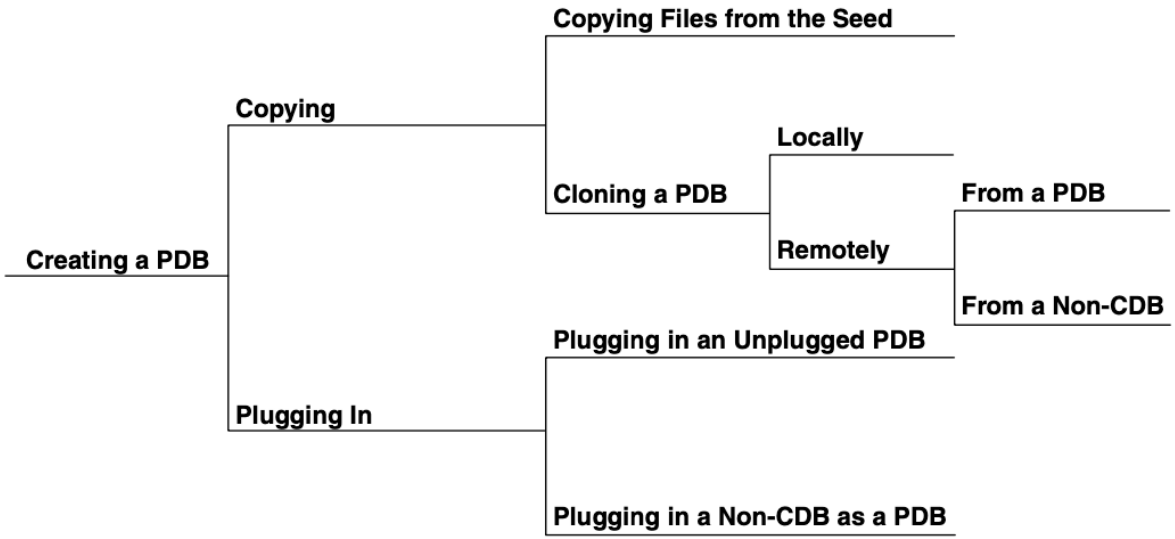
- The following simple query determines whether the database to which an administrative user is currently connected is a non-CDB, or a container in a CDB:

```
SELECT NAME, CDB, CON_ID FROM V$DATABASE;
```

```
NAME      CDB      CON_ID
--------- --- ----------
CDB1      YES           0
```

Creation of a CDB

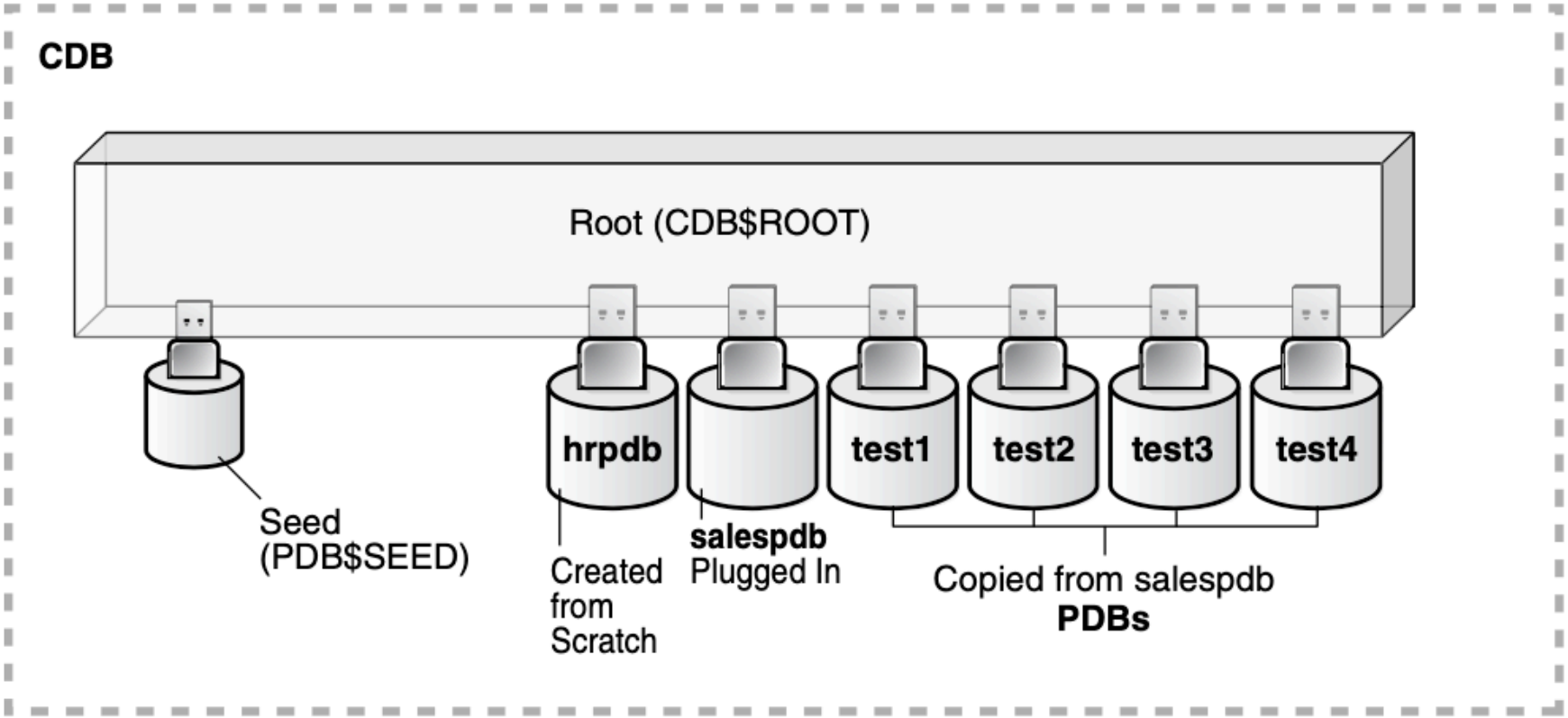
- The CREATE PLUGGABLE DATABASE SQL statement creates a PDB. This PDB automatically includes a full data dictionary including metadata and internal links to system-supplied objects in the root. You can only create a PDB in a CDB and not within another PDB.
- The following figure depicts the options for creating a PDB.



Creation of a CDB

- The following figure shows a CDB that contains six PDBs. hrpdb is a newly created PDB. salespdb was a pre-existing PDB that was unplugged from a different CDB and plugged into this CDB. The remaining four PDBs, each of whose names contains the prefix test, were copied from salespdb.

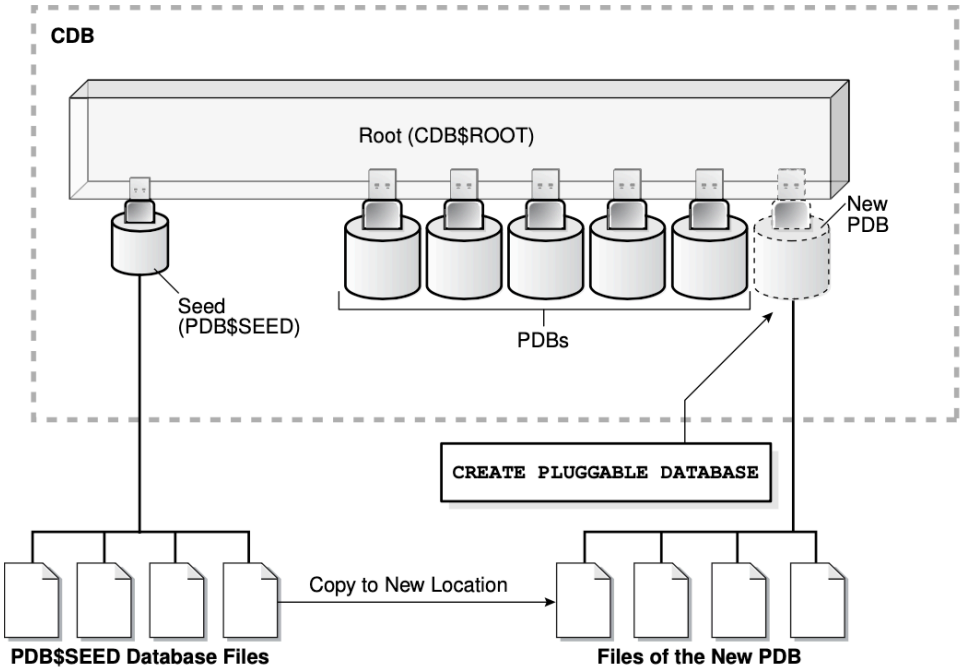
Creation of a CDB



- The following sections describe the different techniques for creating PDBs.

Creation of a PDB from Seed

- You can use the CREATE PLUGGABLE DATABASE statement to create a PDB by copying the files from PDB$SEED, which is a template for creating PDBs.
- The following figure illustrates creation from the seed.



Creation of a PDB from Seed

- The following SQL statement creates a PDB named hrpdb from the seed using Oracle Managed Files:

```
CREATE PLUGGABLE DATABASE hrpdb
ADMIN USER dba1 IDENTIFIED BY password
```

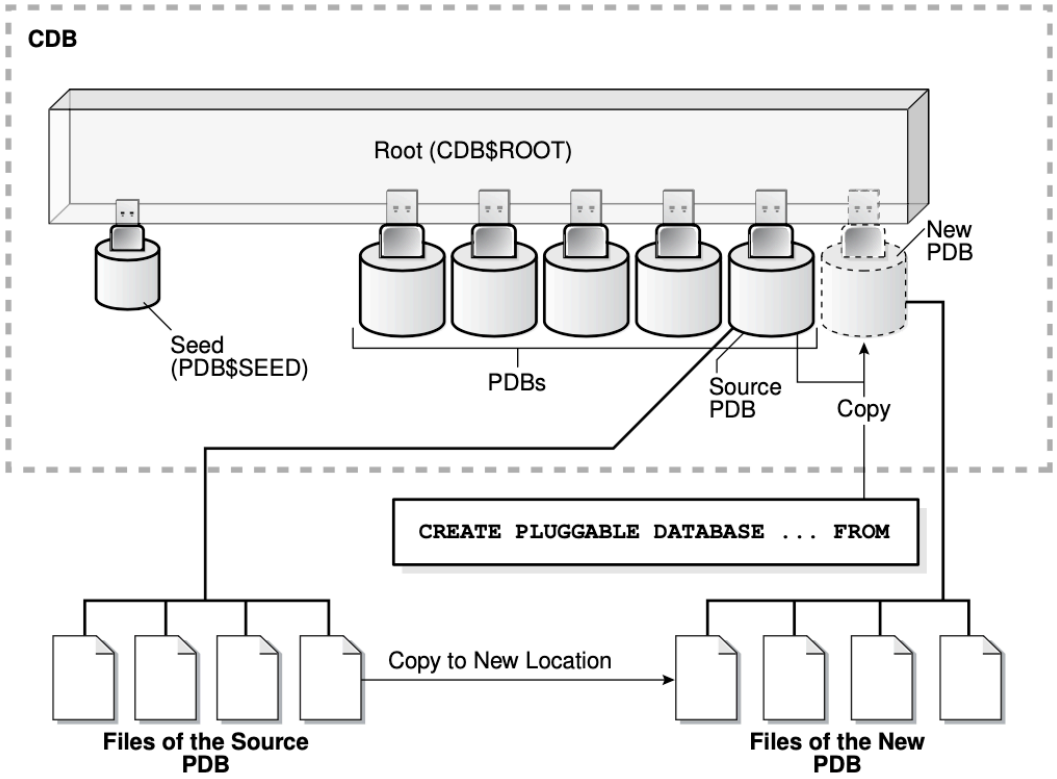
Creation of a PDB by Cloning a PDB or a Non-CDB

- You can use the CREATE PLUGGABLE DATABASE statement to clone a source PDB or non-CDB and plug the clone into the CDB.
- The source can be a PDB in a local or remote CDB, or starting in Oracle Database 12*c* Release 1 (12.1.0.2), it can also be a remote non-CDB. This technique copies the files associated with the source PDB or non-CDB to a new location and associates the copied files with the new PDB.

Creation of a PDB by Cloning a PDB or a Non-CDB

- The following graphic illustrates cloning a PDB from an existing PDB in the same CDB.

Figure 17-8 Cloning a PDB from a PDB in the Same CDB



Creation of a PDB by Cloning a PDB or a Non-CDB

- The following SQL statement clones a PDB named salespdb from the plugged-in PDB named hrpdb:

```
CREATE PLUGGABLE DATABASE salespdb FROM hrpdb
```

Common Users in a CDB

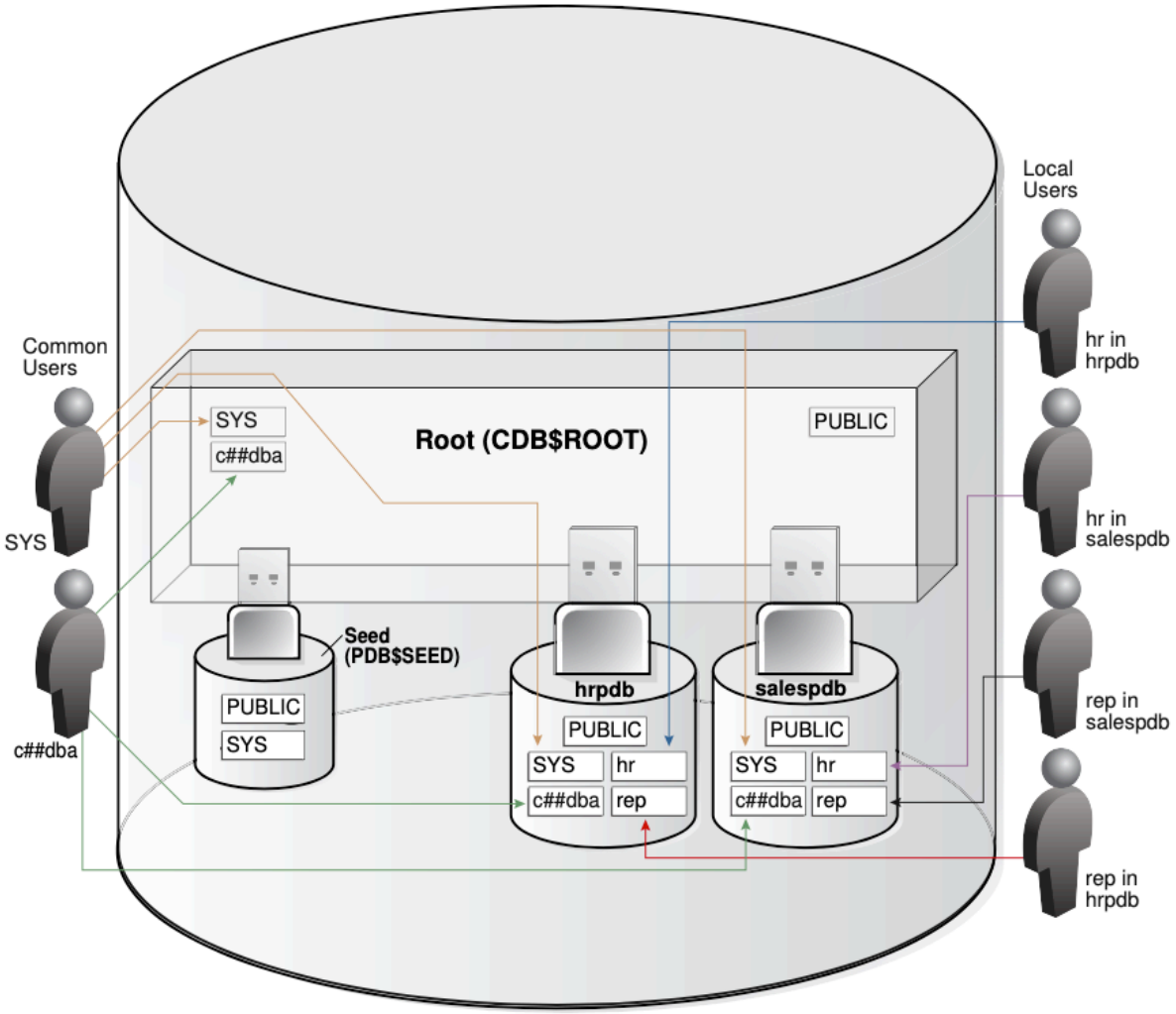
- A **common user** is a database user that has the same identity in the root and in every existing and future PDB. Every common user can connect to and perform operations within the root, and within any PDB in which it has privileges.
- Every common user is either Oracle-supplied or user-created. Examples of Oracle- supplied common users are SYS and SYSTEM.

Common Users in a CDB

- Figure 18-7: shows sample users and schemas in two PDBs: hrpdb and salespdb. SYS and c##dba are common users who have schemas in CDB$ROOT, hrpdb, and salespdb. Local users hr and rep exist in hrpdb. Local users hr and rep also exist in salespdb.

Common Users in a CDB

Figure 18-7 Users and Schemas in a CDB



Common Users in a CDB

- Common users have the following characteristics:
  - A common user can log in to any container (including CDB$ROOT) in which it has the CREATE SESSION privilege.
  - A common user need not have the same privileges in every container. For example, the c##dba user may have the privilege to create a session in hrpdb and in the root, but *not* to create a session in salespdb. Because a common user with the appropriate privileges can switch between containers, a common user in the root can administer PDBs.

Common Users in a CDB

- Common users have the following characteristics:
  - The name of every user-created common user must begin with the characters c## or C##. (Oracle-supplied common user names do not have this restriction.) No local user name may begin with the characters c## or C##.
  - The names of common users must contain only ASCII or EBCDIC characters.
  - Every common user is uniquely named across all containers. A common user resides in the root, but must be able to connect to every PDB with the same identity.
  - The schemas for a common user can differ in each container.

  For example, if c##dba is a common user that has privileges on multiple containers, then the c##dba schema in each of these containers may contain different objects.

Local Users in a CDB

- A **local user** is a database user that is not common and can operate only within a single PDB. Local users have the following characteristics:
  - A local user is specific to a particular PDB and may own a schema in this PDB. In Figure 18-7, local user hr on hrpdb owns the hr schema. On salespdb, local user rep owns the rep schema, and local user hr owns the hr schema.
  - A local user can administer a PDB, including opening and closing it. A common user with SYSDBA privileges can grant SYSDBA privileges to a local user. In this case, the privileged user remains local.

Local Users in a CDB

- A **local user** is a database user that is not common and can operate only within a single PDB. Local users have the following characteristics:
  - A local user in one PDB cannot log in to another PDB or to the CDB root. For example, when local user hr connects to hrpdb, hr cannot access objects in the sh schema that reside in the salespdb database without using a database link. In the same way, when local user sh connects to the salespdb PDB, sh cannot access objects in the hr schema that resides in hrpdb without using a database link.
  - The name of a local user must not begin with the characters c## or C##.

Local Users in a CDB

- A **local user** is a database user that is not common and can operate only within a single PDB. Local users have the following characteristics:
  - The name of a local user must only be unique within its PDB. The user name and the PDB in which that user schema is contained determine a unique local user. Figure 18-7 (page 18-13) shows that a local user and schema named rep exist on hrpdb. A completely independent local user and schema named rep exist on the salespdb PDB.

  The following table describes a scenario involving the CDB in Figure 18-7. Each row describes an action that occurs after the action in the preceding row. Common user SYSTEM creates local users in two PDBs.