

Unit 6. Multitenant Database Architecture

- Understanding the Multitenant Architecture, Pluggable Architecture; Creating CDB;
- Administrating Root Container;
- Creating Pluggable Databases (PDBs) within a CDB;
- Administrating Pluggable Databases;
- Backup and Recovery in multitenant Environment;
- Databases in the Cloud

Multitenant Architecture

- The **multitenant architecture** enables an Oracle database to function as a multitenant container database (CDB).
- A **CDB** includes zero, one, or many customer-created pluggable databases (PDBs).
- A **PDB** is a portable collection of schemas, schema objects, and nonschema objects that appears to an Oracle Net client as a **non-CDB**. All Oracle databases before Oracle Database 12c were non-CDBs.

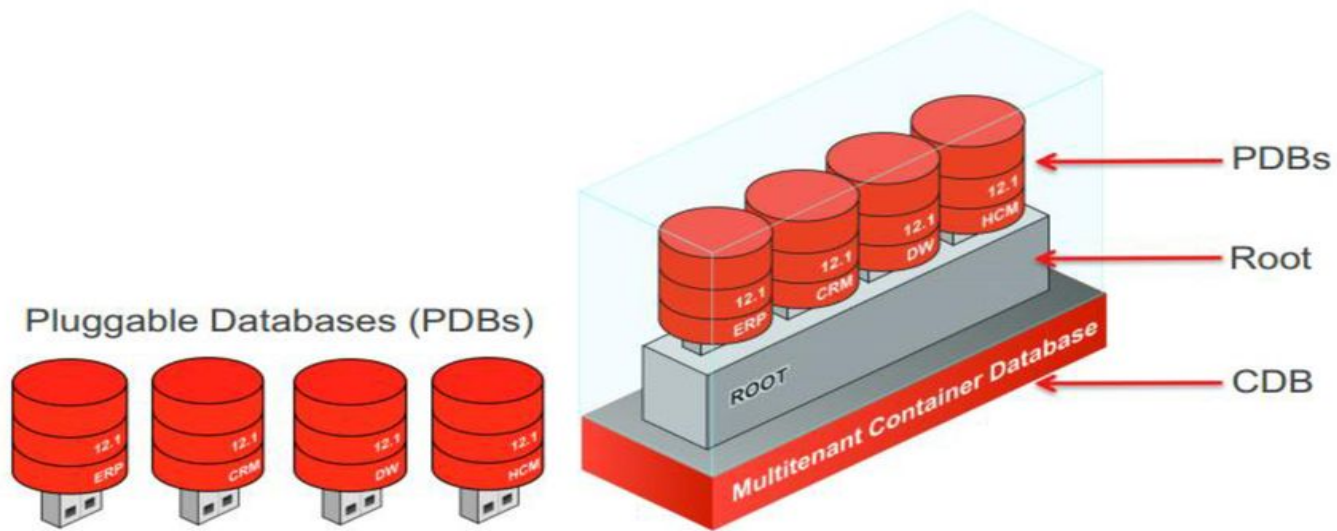
About Containers in a CDB

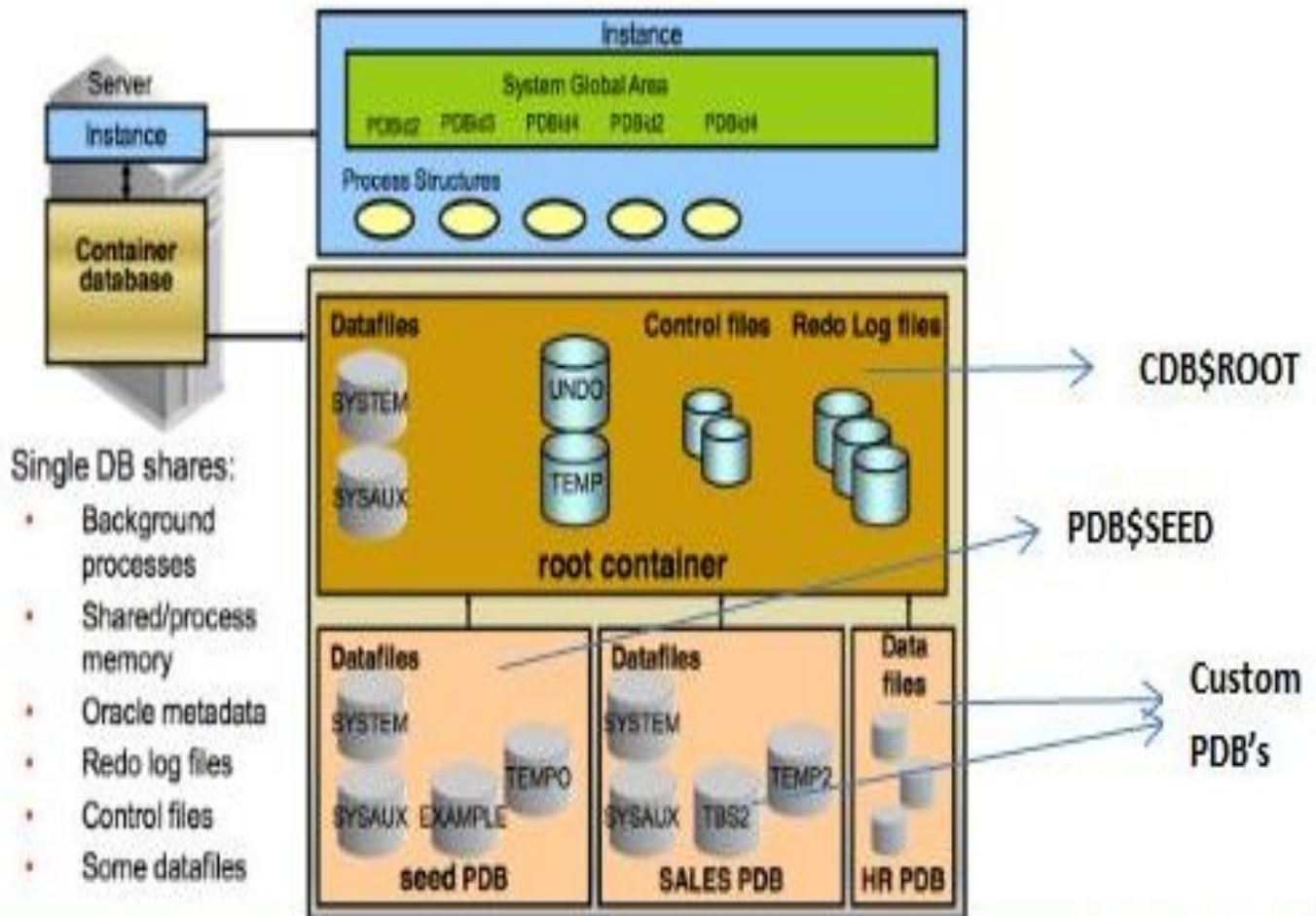
A **container** is either a PDB or the root. The **root container** is a collection of schemas, schema objects, and nonschema objects to which all PDBs belong.

- 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.
 - 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.

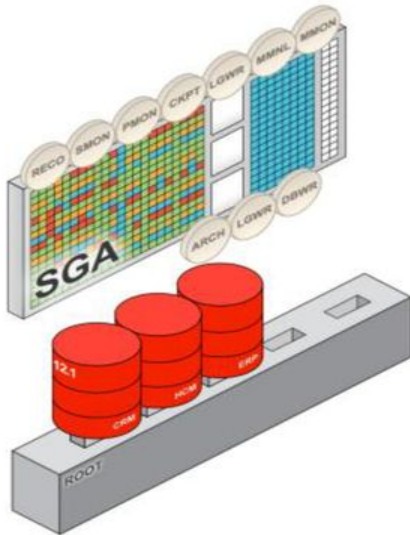
Overview of Multitenant : (CDB) & (PDB)

Components of a Multitenant Container Database (CDB)



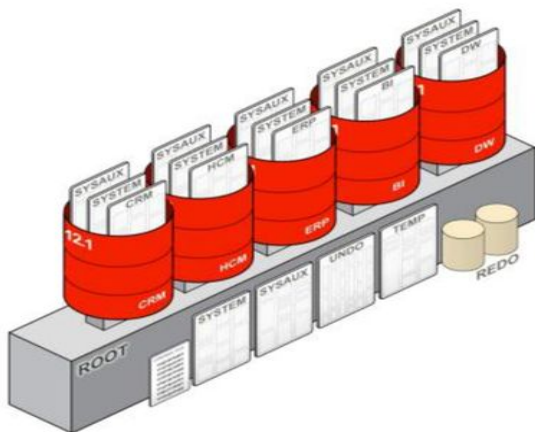


Overview of Multitenant : (CDB) & (PDB)



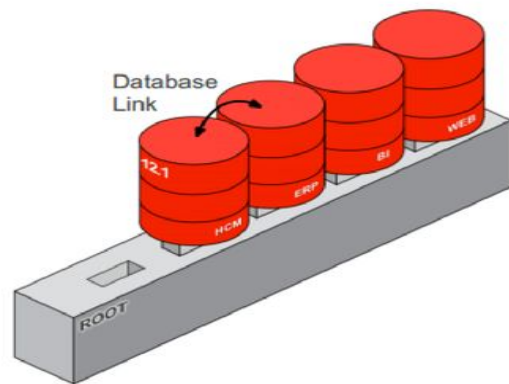
- PDBs share a common memory area
- There is only a single set of database processes serving all PDB's
- Also available on RAC and Exadata

Overview of Multitenant : (CDB) & (PDB)



- Each PDB has its own set of tablespaces including SYSTEM and SYSAUX
- PDBs share UNDO, REDO and control files, (s)pfile
- By default the CDB has a single TEMP tablespace but PDBs may create their own

Overview of Multitenant : (CDB) & (PDB)



- Multitenant architecture can currently support up to 252 PDBs
- A PDB feels and operates identically to a non-CDB
- You cannot tell, from the viewpoint of a connected client, if you're using a PDB or a non-CDB

Create and Configure a (CDB) & (PDB) Databases

Oracle Universal Installer (OUI)

Typical Install Configuration

Perform full database installation with basic configuration.

Oracle base:

Software location:

Storage type:

Database file location:

Database edition:

OSDBA group:

Global database name:

Administrative password:

Confirm password:


Service name:

☒ Create as Container database

Pluggable database name:

Create and Configure a (CDB) & (PDB) Databases

Advanced configuration options(OUI)



Specify Database Identifiers

ORACLE[®] 12^c DATABASE

Provide the identifier information required to access the database uniquely. An Oracle database is uniquely identified by a Global database name, typically of the form "name.domain". A database is referenced by at least one Oracle instance which is uniquely identified from any other instance on this computer by an Oracle system Identifier (SID).

Global database name:

Oracle system identifier (SID):

☒ Create as Container database

Creates a database container for consolidating multiple databases into a single database and enables database virtualization.

Pluggable database name:

Help < Back Next > Install Cancel

Create and Configure a (CDB) & (PDB) Databases

Database Configuration Assistant (DBCA)

Creation Mode

ORACLE DATABASE 12^c

Database Operation

Creation Mode

Pre Requisite Checks

Summary

Progress Page

Create a database with default configuration

Global Database Name:

Storage Type: **Browse...**

Database Files Location: **Browse...**

Fast Recovery Area: **Browse...**

Database Character Set:

Administrative Password:

Confirm Password:

☒ **Create As Container Database**

Pluggable Database Name:

☐ **Advanced Mode**

Help **< Back** **Next >** **Finish** **Cancel**

Create and Configure a (CDB) & (PDB) Databases

Database Configuration Assistant (DBCA)

Database Identification

Database Operation
Creation Mode
Database Template
Database Identification
Management Options
Database Credentials
Storage Locations
Database Options
Initialization Parameters
Creation Options
Pre Requisite Checks
Summary
Progress Page

Database Identification

Global Database Name:

SID:

☒ Create As Container Database
Creates a database container for consolidating multiple databases into a single database and enables database virtualization. A container database (CDB) can have zero or more pluggable databases (PDB).

☐ Create an Empty Container Database

☒ Create a Container Database with one or more PDBs

Number of PDBs:

PDB Name Prefix:

Help < Back Next > Finish Cancel

Create and Configure a (CDB) & (PDB) Databases

Manual Creation

Creating CDB using CREATE DATABASE

- ⇒ The **CREATE DATABASE** command is *almost* the same
- ⇒ **ENABLE PLUGGABLE DATABASE** clause must be used
- ⇒ **SEED FILE_NAME_CONVERT** clause (only if *not* using OMF)
- ⇒ **DB_CREATE_FILE_DEST** initialization parameter if using OMF
- or
- ⇒ **PDB_FILE_NAME_CONVERT** initialization parameter

Create and Configure a (CDB) & (PDB) Databases

CREATE DATABASE *without* OMF

```
SQL> CREATE DATABASE acdb  
USER SYS IDENTIFIED BY ORACLE  
USER SYSTEM IDENTIFIED BY ORACLE
```

....

```
ENABLE PLUGGABLE DATABASE SEED FILE_NAME_CONVERT =  
('/u01/app/oracle/oradata/acdb/', '/u01/app/oracle/oradata/pdbseed/') SYSTEM  
DATAFILES SIZE 300M AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED SYSAUX  
DATAFILES SIZE 200M USER_DATA TABLESPACE usertbs DATAFILE  
'/u01/app/oracle/oradata/pdbseed/usertbs01.dbf' SIZE 200M REUSE  
AUTOEXTEND ON MAXSIZE UNLIMITED;
```

```
SQL> @?/rdbms/admin/catcdb.sql
```


Create and Configure a (CDB) & (PDB) Databases

CREATE DATABASE *with OMF*

```
SQL> CREATE DATABASE acdb  
USER SYS IDENTIFIED BY ORCL  
USER SYSTEM IDENTIFIED BY ORCL  
EXTENT MANAGEMENT LOCAL  
DEFAULT TABLESPACE users  
DEFAULT TEMPORARY TABLESPACE temp  
UNDO TABLESPACE undotbs1
```

ENABLE PLUGGABLE DATABASE

```
SEED  
SYSTEM DATAFILES SIZE 300M AUTOEXTEND ON NEXT 10M MAXSIZE  
UNLIMITED SYSAUX DATAFILES SIZE 200M;
```

```
SQL> @?/rdbms/admin/catcdb.sql*
```


Multitenant : Backup and Recovery (CDB) & (PDB)

Container Database (CDB) Backup

```
$ rman target=/  
RMAN> BACKUP DATABASE PLUS ARCHIVELOG;
```

Root Container Backup

```
$ rman target=/  
RMAN> BACKUP DATABASE ROOT;
```

Pluggable Database (PDB) Backup

```
$ rman target=/  
RMAN> BACKUP PLUGGABLE DATABASE pdb1, pdb2;
```

Multitenant : Backup and Recovery (CDB) & (PDB)

Tablespace and Datafile Backups

⇒ \$ rman target=sys@pdb1
RMAN> BACKUP TABLESPACE system, sysaux, users;

⇒ \$ rman target=sys@pdb1
RMAN> BACKUP DATAFILE 8, 9, 10;

Complete Recovery

Container Database (CDB) Complete Recovery

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

Multitenant : Backup and Recovery (CDB) & (PDB)

Pluggable Database (PDB) Complete Recovery

```
$ rman target=/  
RUN  
{  
ALTER PLUGGABLE DATABASE pdb1, pdb2 CLOSE;  
RESTORE PLUGGABLE DATABASE pdb1, pdb2;  
RECOVER PLUGGABLE DATABASE pdb1, pdb2;  
ALTER PLUGGABLE DATABASE pdb1, pdb2 OPEN;  
}
```

Table Point In Time Recovery (PITR) in PDBs

```
RECOVER TABLE 'TEST'. 'T1' OF PLUGGABLE DATABASE pdb1  
UNTIL SCN 5695703 AUXILIARY DESTINATION '/u01/aux'  
REMAP TABLE 'TEST'. 'T1': 'T1_PREV';
```

Databases in the Cloud

- A cloud database is a database service built and accessed through a cloud platform. It serves many of the same functions as a traditional database with the added flexibility of cloud computing. Users install software on a cloud infrastructure to implement the database.
- Key features:
 - A database service built and accessed through a cloud platform
 - Enables enterprise users to host databases without buying dedicated hardware
 - Can be managed by the user or offered as a service and managed by a provider
 - Can support relational databases and NoSQL databases
 - Accessed through a web interface or vendor-provided API

Platform-as-a-Service (PaaS)

Database-as-a-Service

Self-service / on-demand database consumption, coupled with automation of operations

Compute Services

Virtual servers

Object Storage

Buckets

Block Storage

File Systems

Infrastructure-as-a-Service (IaaS)

Abstraction of Compute, Storage, Networking

Hardware

Database as a Service (DBaaS)

- DBaaS is becoming an increasingly used database solution. Organizations using DBaaS do not require any on-premises infrastructure or in-house database administrators. And DBaaS vendors can provide a fully managed database. It requires almost no investment and normally offers pay-per-use pricing.
- DBaaS reduces in-house labor requirements by making use of automation and machine learning in multiple data stores. Cloud-native API integrations allow organizations to consolidate databases and use only one app. Things like maintenance and security can all be handled in real-time and autonomously. This makes DBaaS easy to use and cost-effective.

Why cloud databases

- **Ease of access**

- Users can access cloud databases from virtually anywhere, using a vendor's API or web interface.

- **Scalability**

- Cloud databases can expand their storage capacities on run-time to accommodate changing needs. Organizations only pay for what they use.

- **Disaster recovery**

- In the event of a natural disaster, equipment failure or power outage, data is kept secure through backups on remote servers.

Thank You