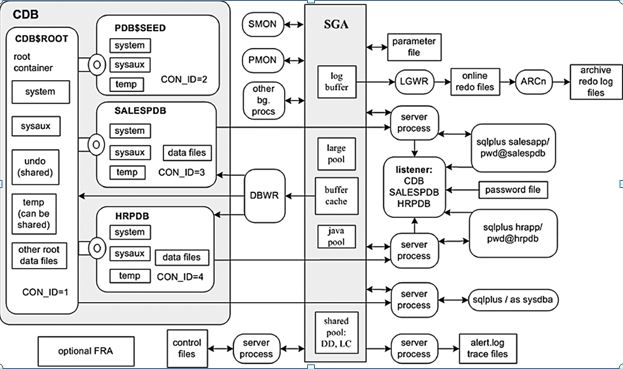
## **2 Oracle Multitenant Architecture**



**CDB$ROOT** 🡺 **PDP$SEED**

🡺 **PLUGGABLE DB1**

🡺**PLUGGABLE DB2**

* In multitenant architecture **USER-DATA** and **META-DATA** is separated.
* **CBD$ROOT** – is a root container **META-DATA** is stored. This is always in read write mode.
* **PDB$SEED** – is a seed and **TEMPLATE** database for creating new pluggable database. This is always in **read only** mode.
* **PDB1** – is a pluggable database stores **USER-DATA**.
* We can un-plug from one container and plug in another container.
* Every pluggable database will have separate tablespaces.
* In **12c** only one **UNDO** tablespace is shared between all the pluggable databases. From **19c** every pluggable database can have separate **UNDO** tablespace.
* The character set defined for the CDB is also used for all associated PDBs.
* We can use same time zone for CDB and all associated PDBs or you can set different time zone for individual for PDB.
* Application team can connect only pluggable database via networking.
* There is only one set of **control files** for CDB.
* There is only one set of **parameter files** for CDB.
* There is only one set **redolog files** for CDB.
* There is one **FRA** for CDB. Flashback database is turned on and off from root container. We can also enable flashback at pluggable database level.
* There is one **Alert log** and set of trace files.
* Each pluggable database is assigned with a unique container ID.
* A pluggable database can be cloned only if it is in read-only mode.
* If the directories not created. It will create automatically as part of the cloning operation.

##### **Enable DDL Logging in Oracle from 12c**

* We already have the alert log file which stores all the logs happening in the database along with some DDLs like creating a database, and altering the database but does not contain DDLs like creating a table, creating columns and dropping tables.
* We can enable it using database triggers but we have another option to enable it using the parameters.
* Show parameter ddl;
* **alter session set enable\_ddl\_logging=True;**
* **xml file for this will be created in ‘/SSD/oracle/app/oracle/diag/rdbms/mouli/mouli/log/ddl’**

**Networking For Container Database:**

* **LISTENER** and **TNS** are required.
* There will be only one listener in container database.
* TNS should be add for **CONTAINER** and **PLUGGABLE** database separately.

**User Management in Container Database:**

1. **Common User:**

* Common user is created under container and it is accessible form any pluggable database in that container.
* Comman user creation must contain **c##** in the beginning of username.
* Create user c##mouli identified by mouli;

1. **Local User:**

* It is accessible only from particular pluggable database where it is created.
* Create user mouli identified by mouli;
* To connect local user we should mention tns of the pluggable database.
* Conn u1/u1@priya

**To check database container or not:**

* Select CDB from v$database;

**To connect to particular pdb:**

* alter session set container=pdb1;

**To open pluggable database from container:**

* alter pluggable database EIS open:

**To close pdb:**

* alter pluggable database EIS close;

**To keep in read only mode:**

* alter pluggable database EIS open read only;

**To check connected pdb or pdbs:**

* show pdbs
* Show con\_name;
* select name,open\_mode from v$containers;

**To drop pdb:**

* dbca

**Or**

* Close the pluggable database and drop.

SQL>**Alter pluggable database pdb2 close.**

SQL>**drop pluggable database pdb2 including datafiles.**

**To create new pluggable database from seed:**

* Create pluggable database SUDHI admin user mouli identified by mouli123 DEFAULT TABLESPACE users datafile ‘/SSD/PDP/MOULI/sudhi/users01.dbf’ size 20m FILE\_NAME\_CONVERT=('/SSD/PDB/MOULI/pdbseed/','/SSD/PDB/MOULI/sudhi/');

**Or**

* Dbca
* Manage pluggable database
* Create pluggable database
* Select the container from which you want to create the pluggable database.
* Select if you want new pluggable database. **( OR )**
* Select existing pluggable database to create another pluggable database.
* Mention new pluggable db name.
* Set administrator user & user.
* Select datafiles location
* Finish.

**To unplug pluggable database:**

* ./dbca
* Manage pluggable database
* Unplug a pluggable database
* Select container database within which pluggable database needs to unplug.
* Select pluggable database name
* Select location for generating **METADATA** file & **DATAFILE** backup.
* generate pluggable database file set
* Finish.

**Pluggable Database Refresh from another PDB:**

* create pluggable database ABC from priya file\_name\_convert=('priya','abc');

**To plug unplugged pluggable database:**

* ./dbca
* Manage pluggable database
* Create a pluggable database
* Select container database within which pluggable database needs to create.
* Create pluggable database from an unplugged PDB.
* Create using PDB file set.
* Select **METADATA** file & **DATAFILE** backup files location.
* Pluggable database name.
* Specify location for pdb datafiles.
* Finish.

**To Create Container Database:**

* ./dbca
* Create a database
* Advanced configuration
* General purpose or transaction processing
* Database name🡺create as container database
* Use following for the database attributes
* Use the same administrative password for all accounts
* Create database
* Finish.