CDB and Regular PDBs

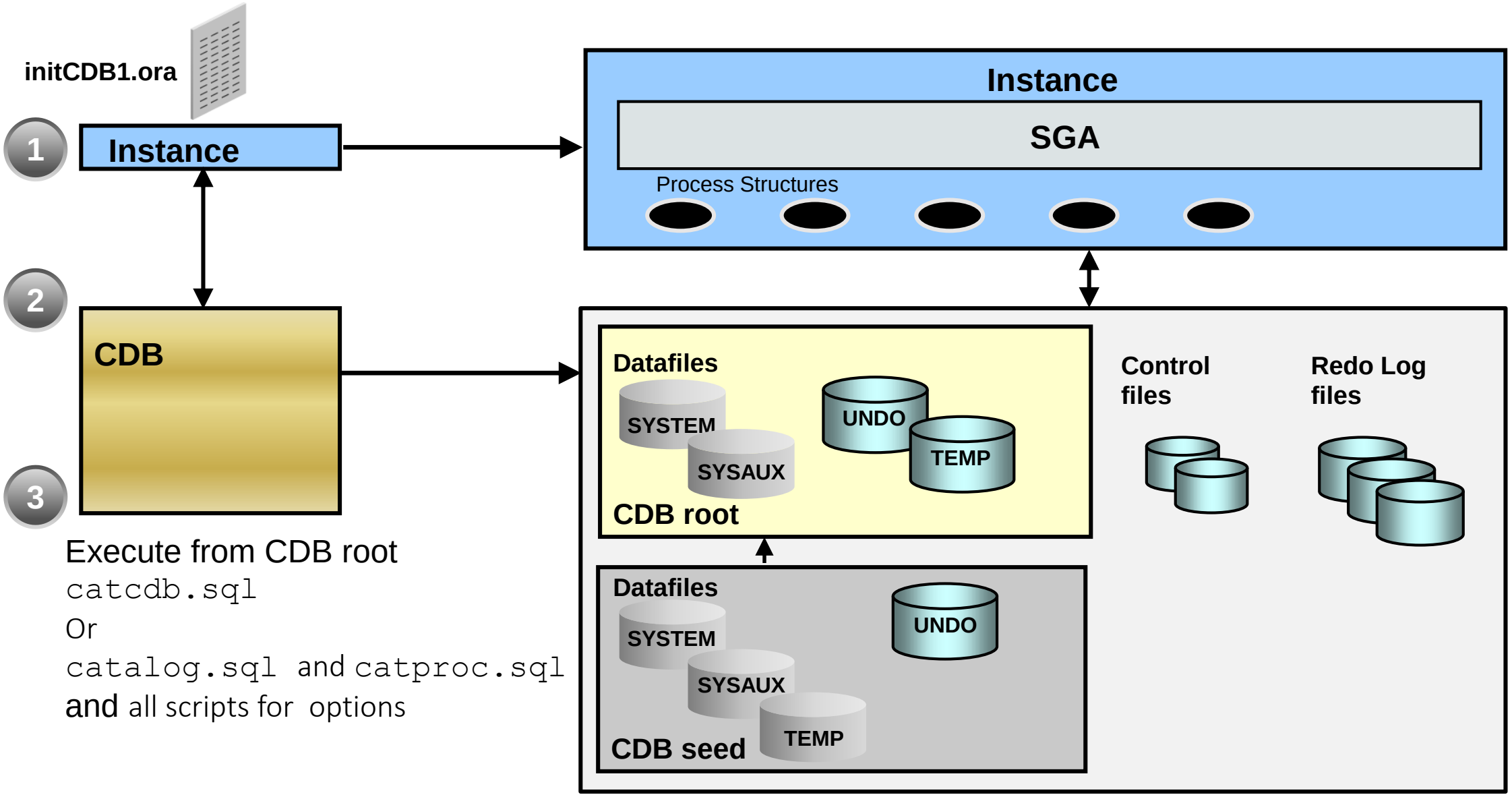
Objectives

- After completing this lesson, you should be able to:
  - Configure and create a CDB
  - Create a new PDB from the CDB seed
  - Explore the instance
  - Explore the structure of PDBs
  - Explore the ADR

Goals

- Create a multitenant container database:
  - To consolidate many pre-12.1, 12c, and 19c non-CDBs into a single, larger database
  - To prepare a container:
    - For plugging any future new application
    - For testing applications
    - For diagnosing application performance
  - To simplify and reduce time for patching and upgrade

Creating a CDB



Execute from CDB root

`catcdb.sql`

Or

`catalog.sql` and `catproc.sql`

and all scripts for options

Creating a CDB: Using SQL*Plus

1. Start up the instance :
   a. Set `ORACLE_SID=CDB1`.
   b. Create the `initCDB1.ora` file and set parameters:
      - `CONTROL_FILES` to CDB control file names
      - `DB_NAME` to a CDB name
      - `ENABLE_PLUGGABLE_DATABASE` to `TRUE`

```
SQL> CONNECT / AS SYSDBA
SQL> STARTUP NOMOUNT
```

2. Create the database:

```
SQL> CREATE DATABASE cdb1 ENABLE PLUGGABLE DATABASE …
             SEED FILE_NAME_CONVERT = ('/oracle/dbs','/oracle/seed');
```

➔ `CDB$ROOT` + `PDB$SEED` created

3. Execute the `$ORACLE_HOME/rdbms/admin/catcdb.sql` SQL script.

New Clause: SEED FILE_NAME_CONVERT

```
SQL> CREATE DATABASE cdb1
     USER SYS IDENTIFIED BY p1 USER SYSTEM IDENTIFIED BY p2
     LOGFILE GROUP 1 ('/u01/app/oradata/CDB1/redo1a.log',
                      '/u02/app/oradata/CDB1/redo1b.log') SIZE 100M,
             GROUP 2 ('/u01/app/oradata/CDB1/redo2a.log',
                      '/u02/app/oradata/CDB1/redo2b.log') SIZE 100M
     CHARACTER SET AL32UTF8 NATIONAL CHARACTER SET AL16UTF16
     EXTENT MANAGEMENT LOCAL DATAFILE
                     '/u01/app/oradata/CDB1/system01.dbf' SIZE 325M
     SYSAUX DATAFILE  '/u01/app/oradata/CDB1/sysaux01.dbf' SIZE 325M
     DEFAULT TEMPORARY TABLESPACE tempts1
            TEMPFILE '/u01/app/oradata/CDB1/temp01.dbf' SIZE 20M
     UNDO TABLESPACE undotbs
             DATAFILE '/u01/app/oradata/CDB1/undotbs01.dbf' SIZE 200M
     ENABLE PLUGGABLE DATABASE
     SEED FILE_NAME_CONVERT =('/u01/app/oradata/CDB1','/u01/app/oradata/CDB1/seed');
```

New Clause: ENABLE PLUGGABLE DATABASE

- Without **SEED FILE_NAME_CONVERT**:
  - OMF: **DB_CREATE_FILE_DEST=**'/u02/app/oradata'

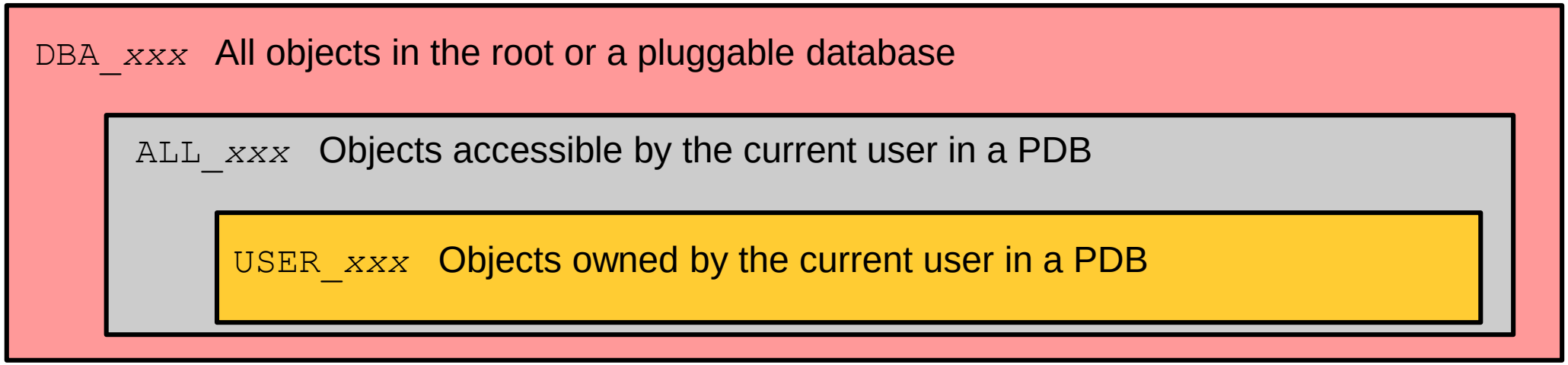
```
SQL> CONNECT / AS SYSDBA
SQL> STARTUP NOMOUNT
SQL> CREATE DATABASE cdb2
       USER SYS IDENTIFIED BY p1 USER SYSTEM IDENTIFIED BY p2
       EXTENT MANAGEMENT LOCAL
       DEFAULT TEMPORARY TABLESPACE temp
       UNDO TABLESPACE undotbs
       DEFAULT TABLESPACE users
       ENABLE PLUGGABLE DATABASE;
```

  - Or new instance parameter: **PDB_FILE_NAME_CONVERT =** '/u02/app/oradata/CDB2','/u02/app/oradata/seed'

After CDB Creation: What's New in CDB

- A CDB has new characteristics compared to non-CDBs:
  - **Two containers:**
    - The CDB **root** (`CDB$ROOT`)
    - The CDB **seed** (`PDB$SEED`)
  - **Several services**: One per container
    - Name of CDB root service = name of the CDB (`cdb2`)
      - Maximum number of services: 10000
      - Max nb of services per PDB<= max nb of services in CDB
  - **Common users** in CDB root and CDB seed: `SYS`, `SYSTEM` …
  - **Common privileges** granted to common users
  - **Predefined common roles**
  - Tablespaces and datafiles associated with each container:
    - `SYSTEM`, `SYSAUX`, and `UNDO`

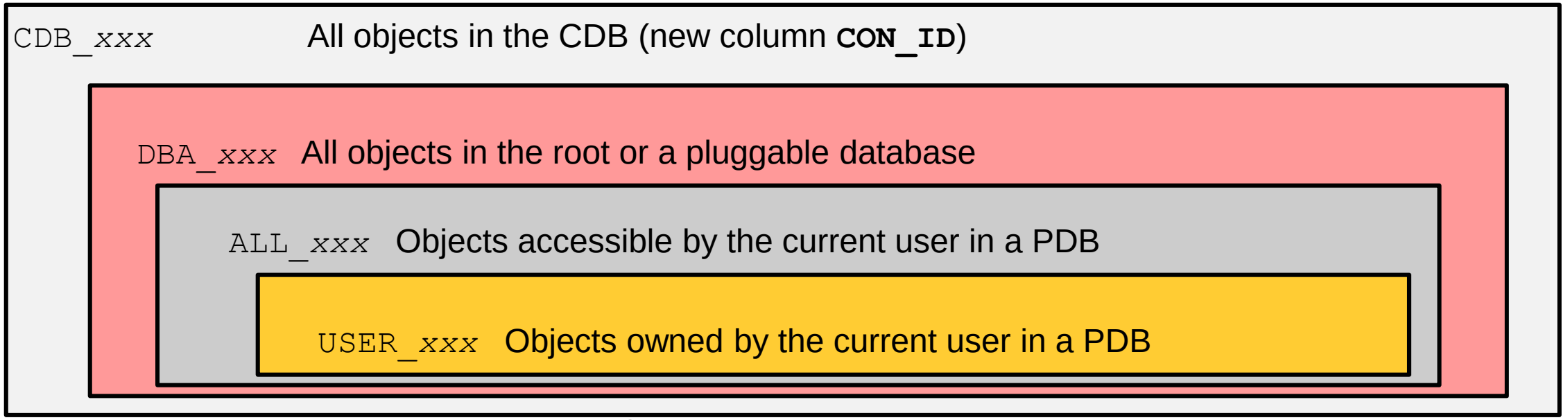
Data Dictionary Views: DBA_*xxx*




```
SQL> SELECT table_name FROM dict WHERE table_name like 'DBA%';
```

- DBA_tablespaces: All tablespaces of the PDB
- DBA_data_files: All datafiles of the PDB
- DBA_tables: All tables in the PDB
- DBA_users: All common and local users of the PDB

Data Dictionary Views: CDB_xxx



- CDB dictionary views provide information across PDBs:

```
SQL> SELECT view_name FROM dba_views WHERE view_name like 'CDB%';
```

- `CDB_pdbs`: All PDBs within the CDB
- `CDB_tablespaces`: All tablespaces within the CDB
- `CDB_users`: All users within the CDB (common and local)

Data Dictionary Views: Examples

- Comparisons:

```
SQL> CONNECT / AS SYSDBA
SQL> SELECT role, common, con_id FROM cdb_roles;
SQL> SELECT role, common FROM dba_roles;
```

```
SQL> CONNECT sys@PDB1 AS SYSDBA
SQL> SELECT role, common, con_id FROM cdb_roles;
SQL> SELECT role, common FROM dba_roles;
```

- Access to V$ views showing data from PDBs can be secured using privilege.

```
SQL> SELECT name,open_mode FROM v$pdbs;

NAME                 OPEN_MODE
---------------- ----------
PDB$SEED             READ ONLY
PDB1                 READ WRITE
```

Data Dictionary Views: V$xxx Views

- SGA accessed by all containers: V$ views and `CON_ID` column

```
SQL> SELECT distinct status, con_id FROM v$bh order by 2;

STATUS      CON_ID
---------- ------
cr               1 ──────────► CDB root
free             1
xcur             1
xcur             2 ──────────► CDB seed
cr               3 ──────────► PDB1 PDB
```

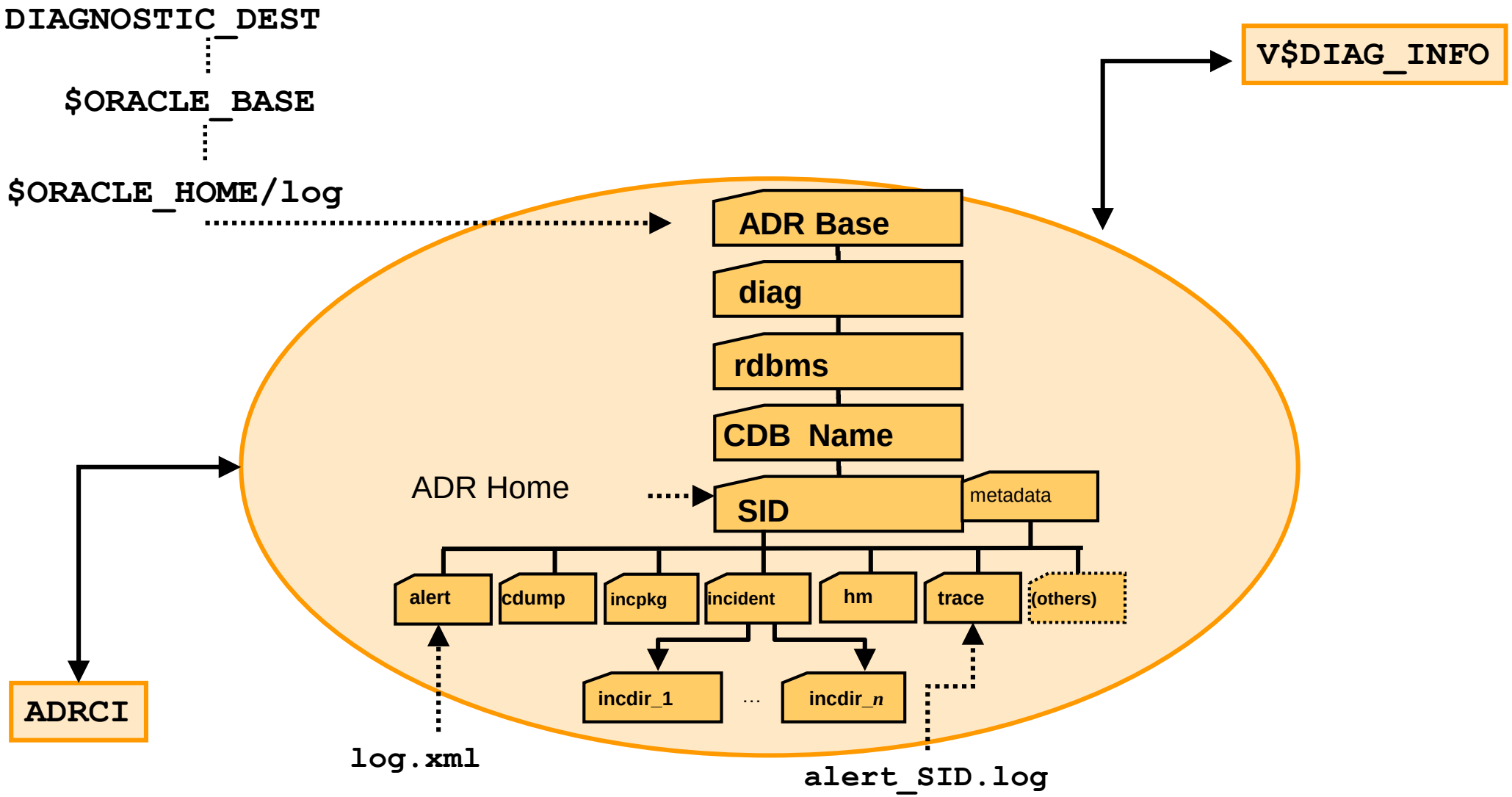
```
SQL> select OBJECT_ID, ORACLE_USERNAME, LOCKED_MODE, CON_ID from V$LOCKED_OBJECT;

OBJECT_ID  ORACLE_USERNAME  LOCKED_MODE  CON_ID
---------- ---------------- ----------- -------
     83711 SYS                       3       3 ◄────────── PDB1 PDB
     83710 DOM                       3       4 ◄────────── PDB2 PDB
```

After CDB Creation: To do List

- After CDB creation, the CDBA has to:
  - Create the SPFILE from the PFILE.
  - Execute the `$ORACLE_HOME/rdbms/admin/utlrp.sql` script.
  - Optionally plug non-CDBs and create new PDBs.
  - Test startup/shutdown procedures.
  - Automate PDBs opening.
  - Create backup and recovery procedures.
- After PDB creation, each PDBA in its own PDB has to:
  - Set a default tablespace.
  - Optionally create additional temporary tablespaces.

Automatic Diagnostic Repository



Automatic Diagnostic Repository: `alert.log` File

- The `alert_CBD1.log` shows new DDL statements.

```
CREATE DATABASE cdb1
  …
ENABLE PLUGGABLE DATABASE
SEED   FILE_NAME_CONVERT=('/u01/app/oradata/CDB1','/u01/app/oradata/seed');

CREATE PLUGGABLE DATABASE PDB$SEED AS CLONE USING …
CREATE PLUGGABLE DATABASE pdb1 … ;
ALTER PLUGGABLE DATABASE pdb1 UNPLUG INTO … ;
ALTER PLUGGABLE DATABASE ALL OPEN ;
ALTER PLUGGABLE DATABASE pdb2 CLOSE IMMEDIATE ;
```

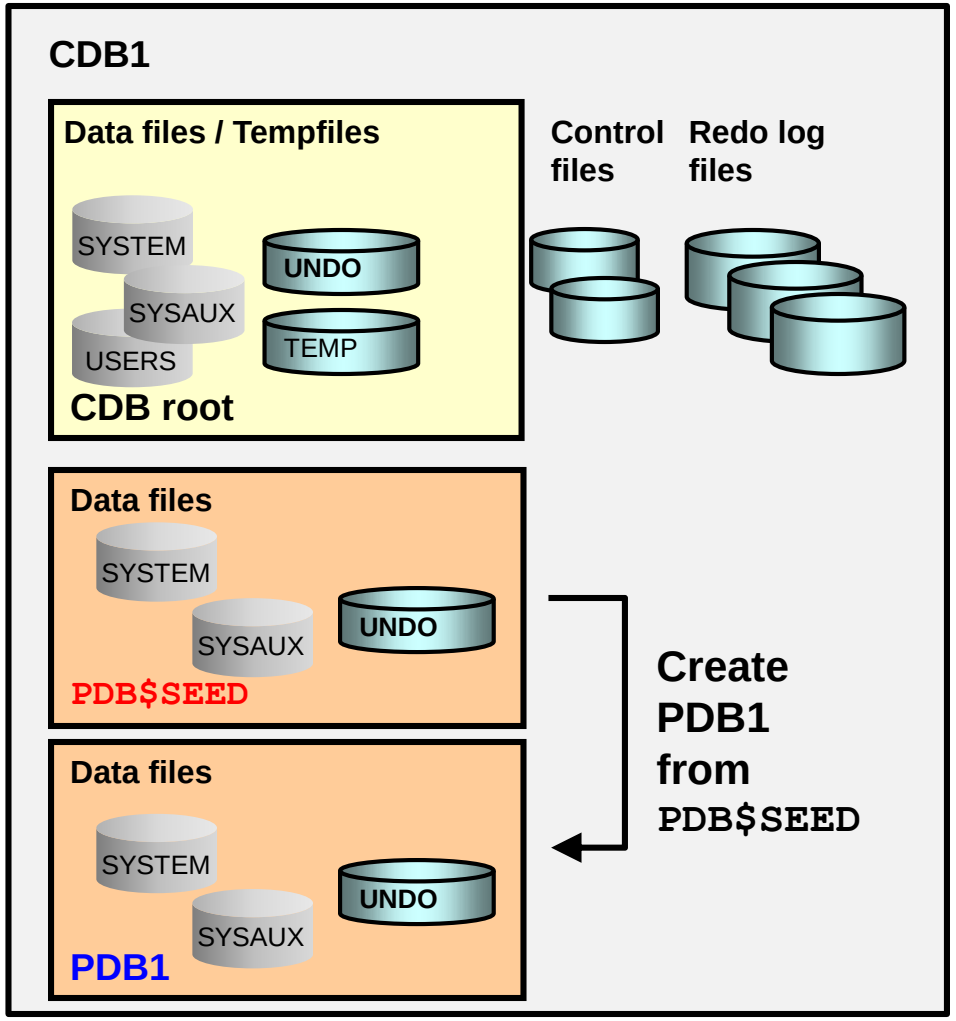
Provisioning New Pluggable Databases

- Create a new PDB from the CDB seed.
- Plug an unplugged PDB into the same CDB or into another CDB.
- Plug a non-CDB in a CDB.
- Clone a PDB from another PDB (local or remote CDB, hot or cold).
- Relocate a PDB from a CDB into another CDB.
- Proxy a PDB from another PDB.

Tools

- To provision new PDBs, you can use:
  - SQL*Plus
  - SQL Developer
  - Enterprise Manager Cloud Control
  - Enterprise Manager Database Express
  - Database Configuration Assistant (DBCA)
    - Clone from CDB seed
    - Clone from an existing PDB
    - Plug an unplugged PDB

Create New PDB from PDB$SEED



- Copies the datafiles from PDB$SEED datafiles
- Creates tablespaces SYSTEM, SYSAUX, UNDO
- 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

Steps: With **FILE_NAME_CONVERT**

- Create a new PDB from the seed using **FILE_NAME_CONVERT**:

1. Connect to the CDB root as a common user with the `CREATE PLUGGABLE DATABASE` system privilege:

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

2. Use views to verify:

```
SQL> CONNECT / AS SYSDBA
SQL> SELECT * FROM cdb_pdbs;
SQL> SELECT * FROM cdb_tablespaces;
SQL> SELECT * FROM cdb_data_files;
SQL> ALTER PLUGGABLE DATABASE pdb1 OPEN RESTRICTED;
SQL> CONNECT sys@pdb1 AS SYSDBA
SQL> CONNECT admin1@pdb1
```

Steps: Without FILE_NAME_CONVERT

- Create a new PDB from seed without **FILE_NAME_CONVERT**:
  - Use OMF: **DB_CREATE_FILE_DEST =** '/u01/app/oradata/CDB1/**pdb1**'

Or

  - Use the instance parameter: **PDB_FILE_NAME_CONVERT =** '/u01/app/oradata/CDB1/seed','/u01/app/oradata/CDB1/**pdb1**'

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

  - Use the clause in the CREATE PLUGGABLE DATABASE command: **CREATE_FILE_DEST =** '/u01/app/oradata/CDB1/**pdb1**'

Summary

- In this lesson, you should have learned how to:
  - Configure and create a CDB
  - Create a new PDB from the CDB seed
  - Explore the instance
  - Explore the structure of PDBs
  - Explore the ADR

Practice 2: Overview

- 2-1: Exploring CDB architecture and structures
- 2-2: Creating a new CDB
- 2-3: Creating a new PDB