ORACLE

Oracle Database 12*c* –
Pluggable Databases – Part II

ORACLE

Using the Player

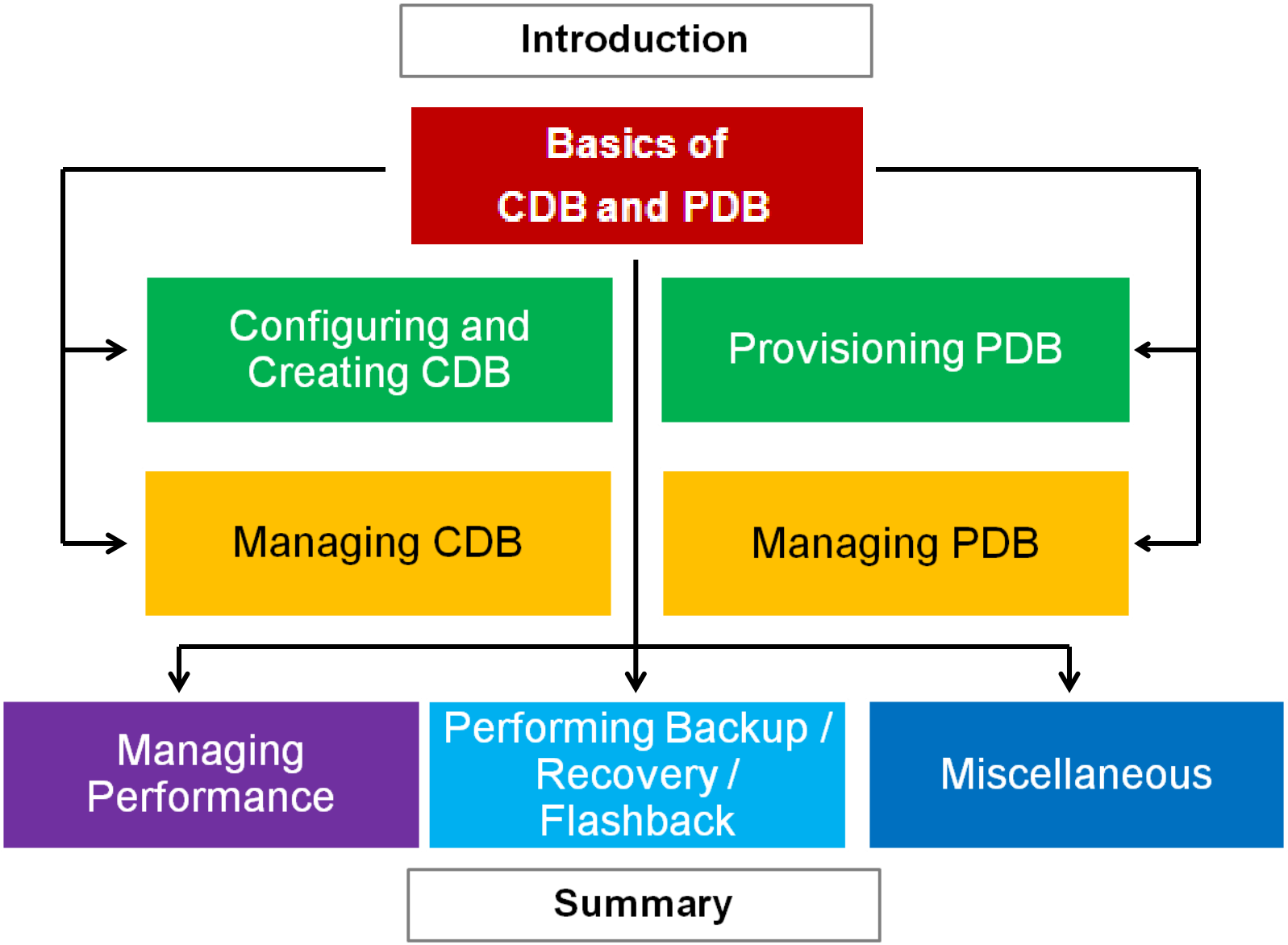
Print Version

Course Outline

Player Controls

Change View

Road Map



Road Map

PART – I

Basics of CDB and PDB

Configuring and Creating CDB

Provisioning PDB

PART – II

Managing CDB

Managing PDB

PART – III

Managing Performance

Performing Backup / Recovery / Flashback

Miscellaneous

About This Course

Search

What skills will I learn?

At the end of this course, you should be able to:

- Connect to a container database
- Connect to a pluggable database
- Start up and shut down a container database
- Open and close a pluggable database
- Change the open mode of a pluggable database
- Change instance parameter values
- Manage permanent and temporary tablespaces in CDB and PBD
- Manage common and local users
- Manage common and local roles
- Manage common and local privileges
- Manage common and local objects

Who is the target audience?

What are the prerequisites?

PROPERTIES

Allow user to leave interaction:	Anytime
Show 'Next Slide' Button:	Show upon completion
Completion Button Label:	Next Slide

Properties...

Edit in Engage

Why Take This Course?

- What's in it for me?
- What are challenges I face on the job?
- How can this help me do my job better?

ORACLE

Oracle Database 12*c* –
Managing Pluggable Databases

ORACLE

Road Map

PART – II

Managing CDB

Managing PDB

Goals

Manage applications independently:

- Perform application patching

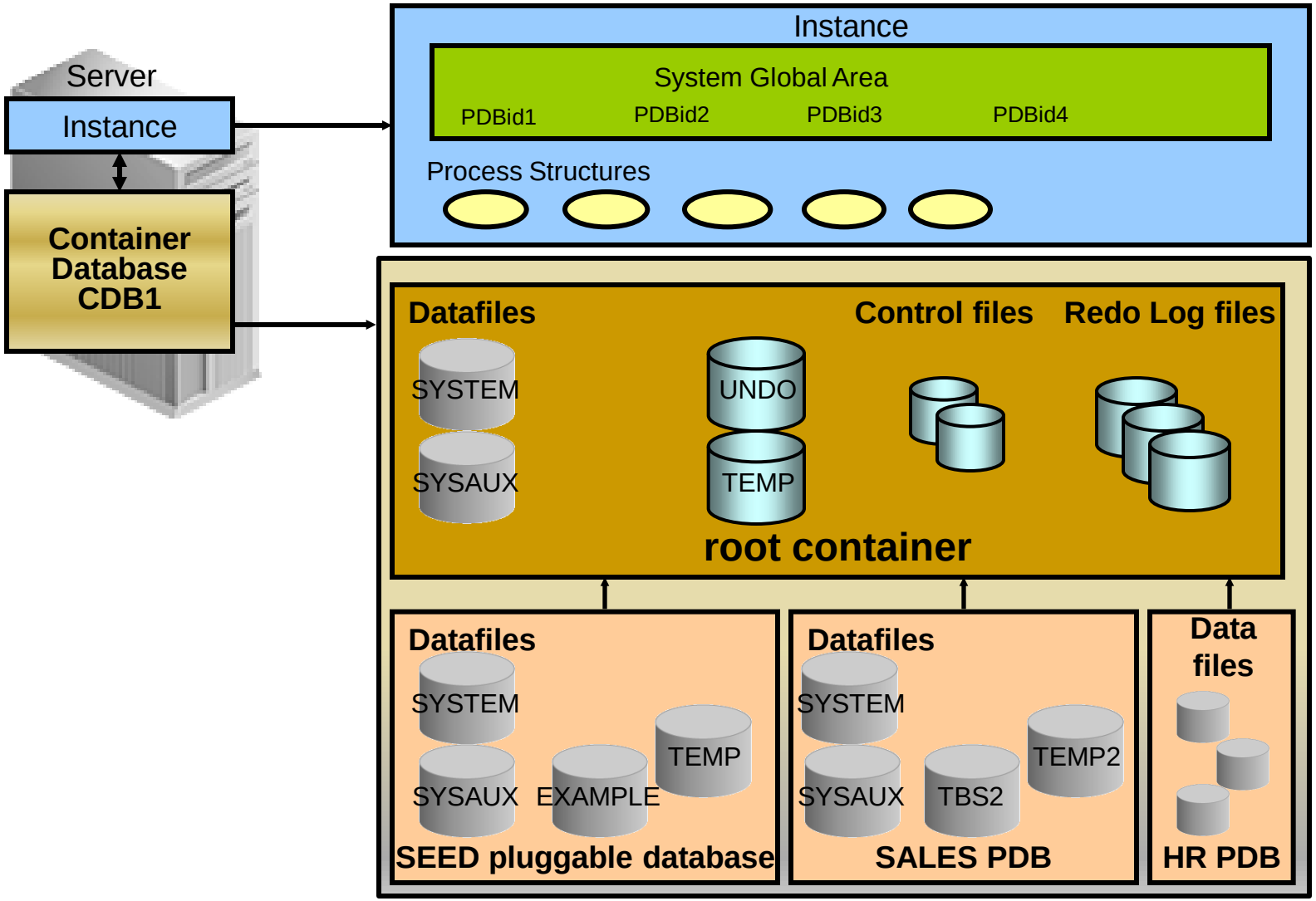
Manage pluggable databases independently

- Close PDBs in a CDB while others are kept opened
- Change the state of one PDB and not others
  - Open one PDB in restrict mode for maintenance
  - Open one PDB in read only mode for cloning
  - Perform PDB files maintenance
  - Recover a PDB while others are kept available
- Changing the default temporary tablespace of a PDB
- Setting the storage limit for a PDB

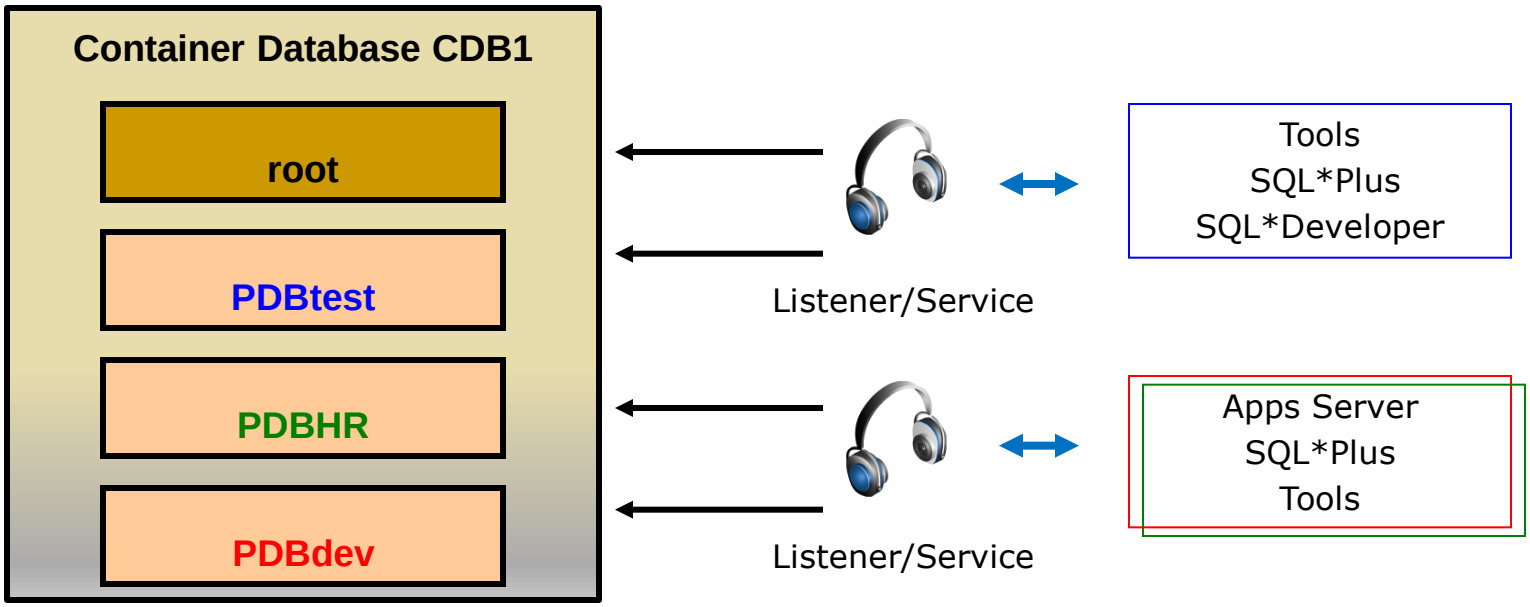
Secure navigation across PDBs

- Use new system privileges

Oracle Database Server Architecture in 12*c*



Connection



- Every PDB has a default service with its name.

```
SQL> SELECT name, pdb FROM cdb_services;
```

- Service name has to be unique across CDB.

```
SQL> CONNECT / AS SYSDBA
SQL> CONNECT sys/pass2@PDBtest AS SYSDBA
SQL> CONNECT local_user1/pass1@hostname1:1525/PDBHR
SQL> CONNECT common_user2/pass2@PDBdev
SQL> SHOW CON_NAME
SQL> SHOW CON_ID
```

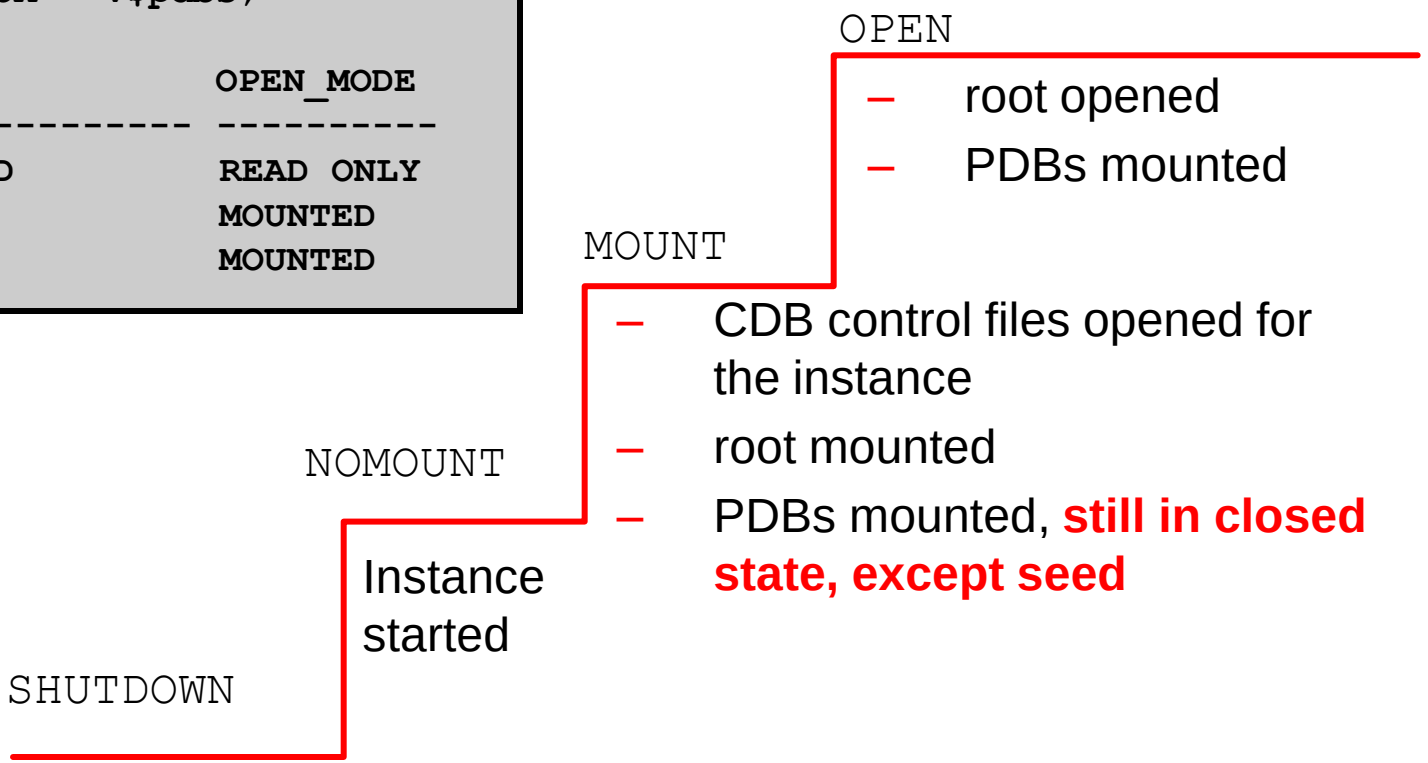
Opening a CDB

```
SQL> STARTUP
```

Or

```
SQL> ALTER DATABASE cdb1 OPEN;
```

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

NAME                 OPEN_MODE
---------------- ----------
PDB$SEED             READ ONLY
PDB1                 MOUNTED
PDB2                 MOUNTED
```



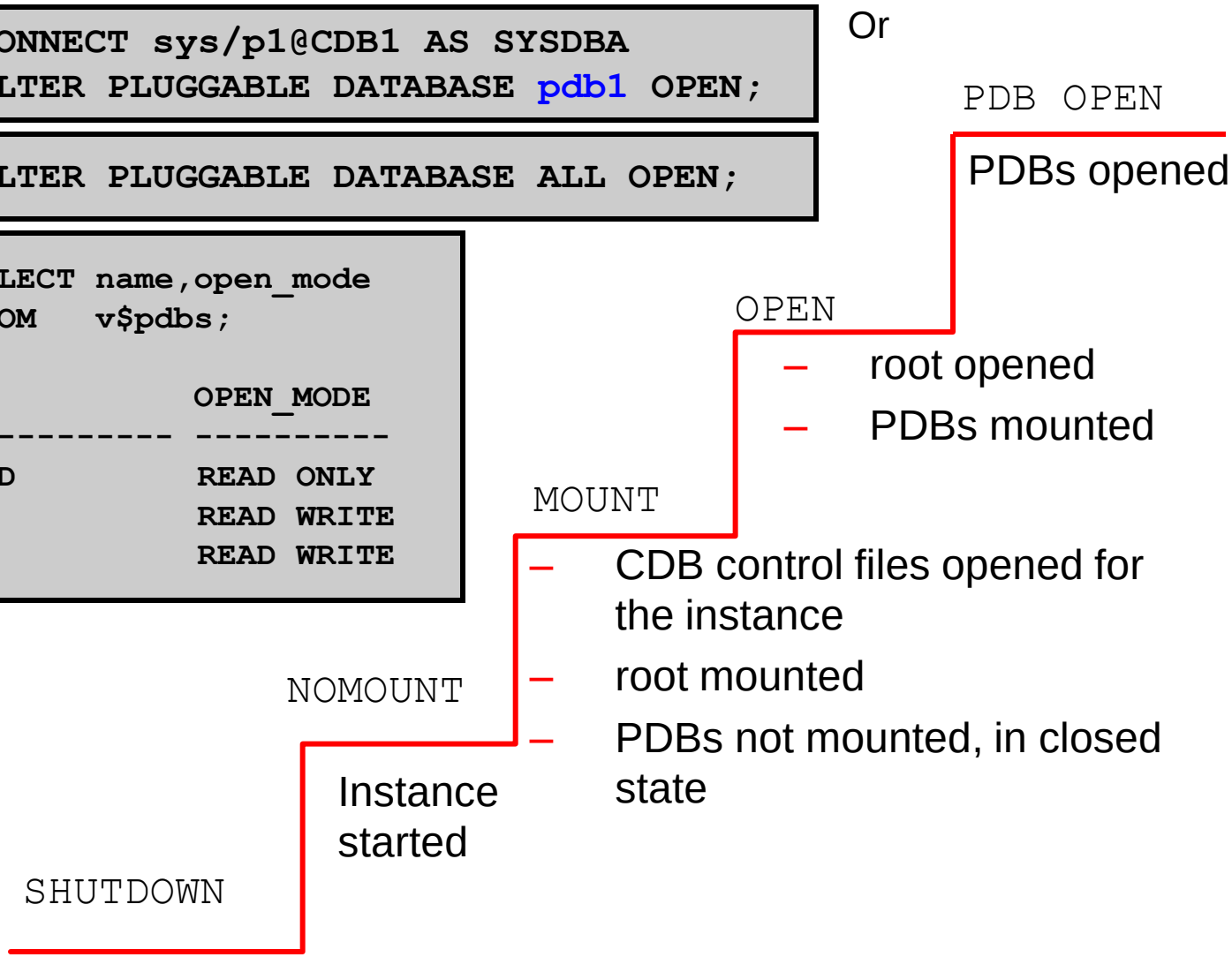
Opening a PDB

```
SQL> CONNECT sys/p1@CDB1 AS SYSDBA
SQL> ALTER PLUGGABLE DATABASE pdb1 OPEN;
```

Or

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

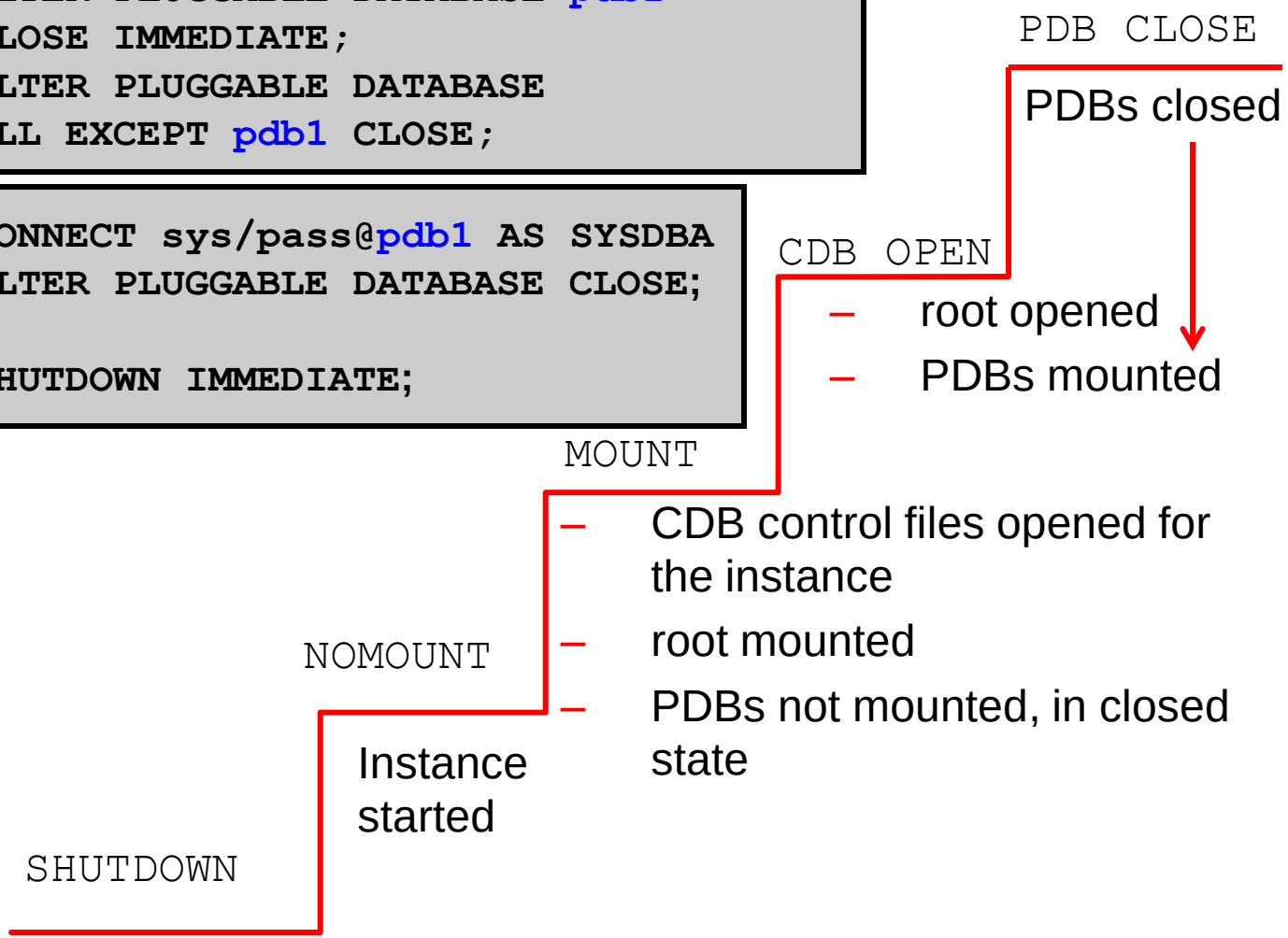
```
SQL> SELECT name,open_mode
  2  FROM   v$pdbs;

NAME                 OPEN_MODE
-------------------- ----------
PDB$SEED             READ ONLY
PDB1                 READ WRITE
PDB2                 READ WRITE
```



Closing a PDB

```
SQL> CONNECT / AS SYSDBA
SQL> ALTER PLUGGABLE DATABASE pdb1
  2  CLOSE IMMEDIATE;
SQL> ALTER PLUGGABLE DATABASE
  2  ALL EXCEPT pdb1 CLOSE;
```

```
SQL> CONNECT sys/pass@pdb1 AS SYSDBA
SQL> ALTER PLUGGABLE DATABASE CLOSE;
Or
SQL> SHUTDOWN IMMEDIATE;
```



Changing a PDB Open Mode

After closing a PDB, open in:

- Restricted mode
- Read only mode
- Read write

```
SQL> CONNECT sys/pass@pdb1 AS SYSDBA
SQL> ALTER PLUGGABLE DATABASE CLOSE;
```

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

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

NAME                           OPEN_MODE
------------------------------ ----------
PDB1                           RESTRICTED
```

```
SQL> CONNECT / AS SYSDBA
SQL> ALTER PLUGGABLE DATABASE ALL OPEN READ ONLY;
```

Instance Parameter Change Impact

- A single SPFILE per CDB
- Different values per PDB in memory and SPFILE
- Only for parameters `ISPDB_MODIFIABLE=TRUE`
- PDB values changes stored in SPFILE after:
  - CDB shutdown / startup
  - PDB close / open

Demo: Opening and Closing CDBs and PDBs

PROPERTIES

Allow user to leave interaction:	Anytime
Show ‘Next Slide’ Button:	Show upon completion
Completion Button Label:	Next Slide

Properties...

Edit in Engage

Demo: Managing PDBs Open Modes

PROPERTIES

Allow user to leave interaction:	Anytime
Show ‘Next Slide’ Button:	Show always
Completion Button Label:	Next Slide

Properties...

Edit in Engage

Lesson Summary

In this first lesson of the second part of the course, we discussed:

PART – II

Managing CDB

Managing PDB

- The connections to CDB / PDB
- How to start up and shut down a CDB
- How to open and close PDBs
- The different open modes of PDBs
- The impact of parameter value changes

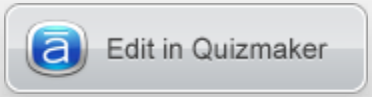
Lesson Review

Which of the following is true when switching connection to another PDB?

- ○ Always possible as long as you are granted SET CONTAINER privilege.
- ○ Always possible if initially connected under a common user.
- ○ Requires SYSDBA privilege.
- ○ Only possible if initially connected under a common user with granted SET CONTAINER privilege in both PDBs.

PROPERTIES

On passing, 'Finish' button: **Goes to Next Slide**
On failing, 'Finish' button: **Goes to Next Slide**
Allow user to leave quiz: **At any time**
User may view slides after quiz: **At any time**
User may attempt quiz: **Unlimited times**



ORACLE

ORACLE THIRTY YEARS 30

Oracle Database 12*c* –
Managing the Container Database

ORACLE

Road Map

PART – II

Managing CDB

Managing PDB

Data Dictionary Views

CDB dictionary views provide information across PDBs:

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

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

PDB dictionary views providing information within PDB:

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

- `DBA_tablespaces` : All tablespaces within the PDB
- `DBA_tables`

Access to data in views containing data pertaining to multiple PDBs (such as V$ or GV$) can be secured using privilege.

Cross Container Operations

Are operations that affect:

- The entire CDB including all PDBs
- Multiple containers or objects (common users) in multiple containers
- A container different than the one the user is connect to.

Cross-container operations are limited to:

- Common users connected to root container

Examples:

- Grant of common privilege to a common user
- `ALTER DATABASE` specifying a recovery clause
- `ALTER PLUGGABLE DATABASE` that changes the state of a PDB

Tablespaces in Pluggable Databases

- A tablespace in a PDB can contain objects associated with exactly one PDB
- In `CREATE PLUGGABLE DATABASE`:
  - The amount of space allowed to all tablespaces in a PDB can be limited with `STORAGE (MAXSIZE , <size>)`
  - The amount of space allowed for a PDB in a shared temporary tablespace can be limited by `STORAGE(MAX_SHARED_TEMP_SIZE, <size>)`
- In `CREATE DATABASE`:
  - `USER_DATA TABLESPACE` replaces automatic creation of `USERS` tablespace by DBCA.
  - `DATAFILES` clause sets sizes for `SYSTEM` and `SYSAUX` datafiles in seed PDB
- Only one active `UNDO` tablespace per CDB

Creating Permanent Tablespaces in a CDB

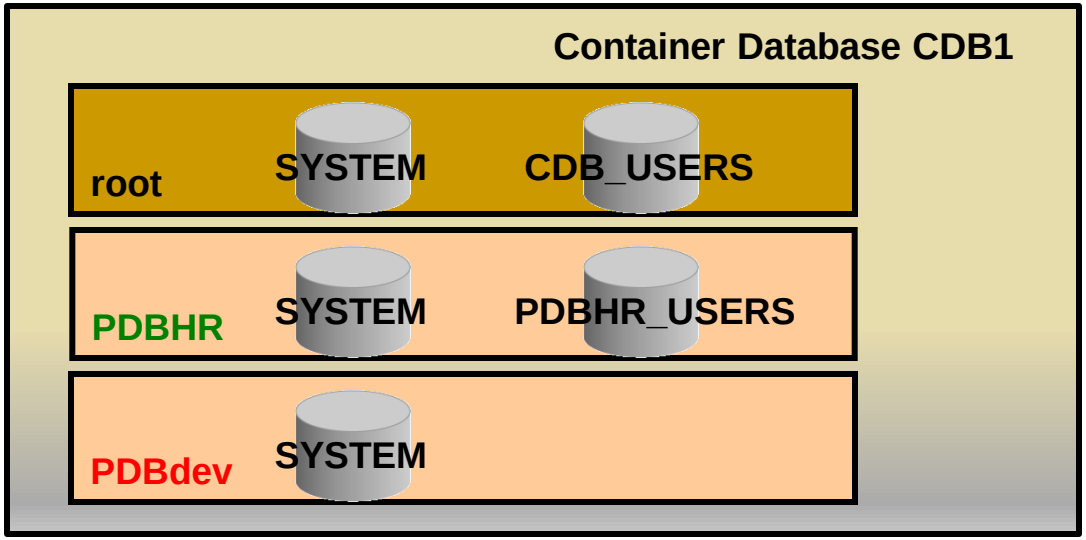
Create a permanent tablespace in the root container:

```
SQL> CONNECT system/pwd@ROOT
SQL> CREATE TABLESPACE CDB_USERS
  2> DATAFILE
  3> '/u1/app/oracle/oradata/cdb/cdb_users01'
  4>  SIZE 100M;
```

Create a permanent tablespace in a PDB:

```
SQL> CONNECT system/pwd@PDB1
SQL> CREATE TABLESPACE PDB1_users
  2> DATAFILE
  3> '/u1/app/oracle/oradata/cdb/pdb1/users01.dbf'
  4> SIZE 100M;
```

Assigning Default Tablespaces



In the CDB:

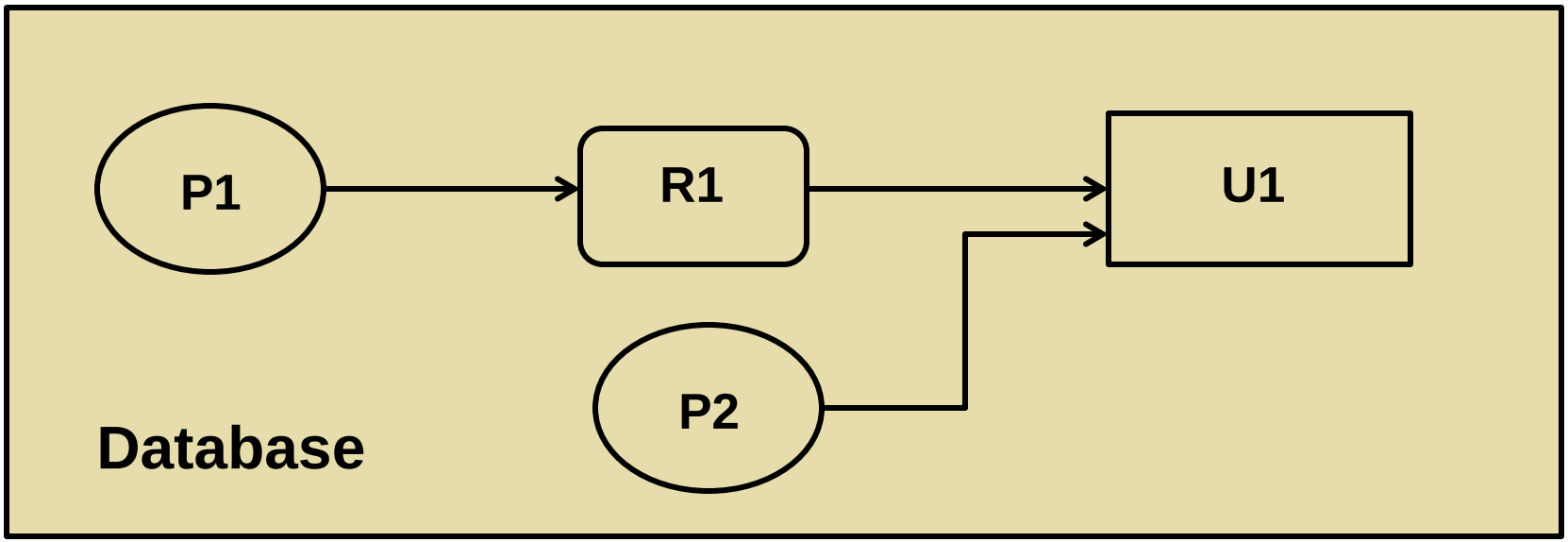
```
SQL> CONNECT system/pwd@ROOT
SQL> ALTER DATABASE
  2> DEFAULT TABLESPACE CDB_USERS;
```

In the PBD:

```
SQL> CONNECT PDB1_ADMIN/pwd@PDBHR
SQL> ALTER PLUGGABLE DATABASE
  2> DEFAULT TABLESPACE PDBHR_USERS;
```

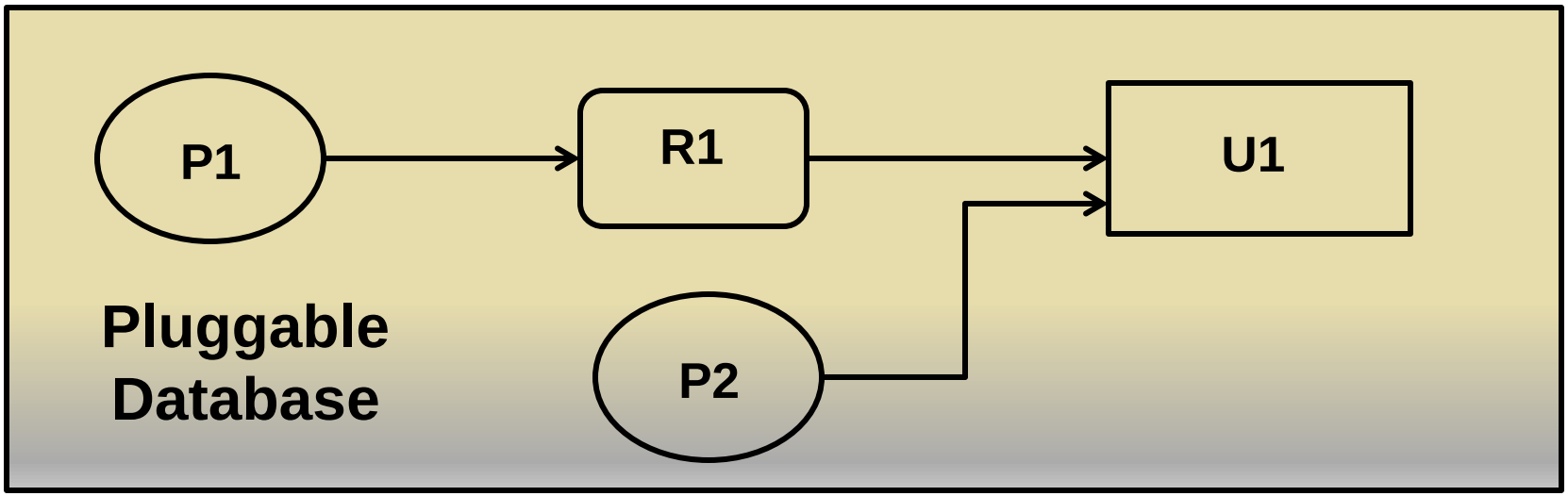
Users, Roles and Privileges

- Each user can exercise granted privileges in the context of a single database.
- A role is a collection of privileges.

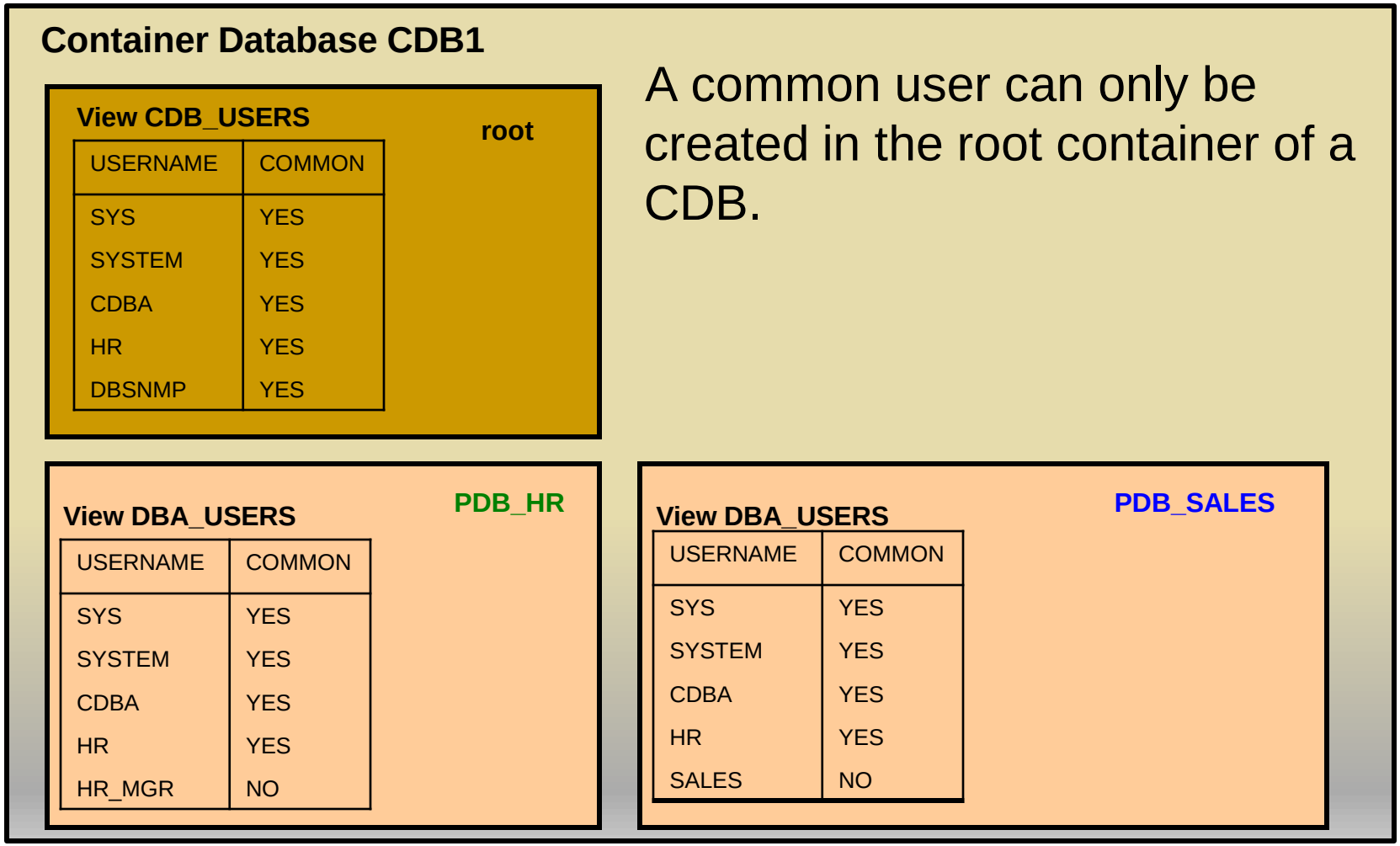


Local Users, Roles and Privileges

- Each **local** user can exercise granted privileges in the context of a single **pluggable** database.
- A **local** role is a collection of privileges that are assigned at user login to a specific **pluggable** database.
- A **local** privilege is one that is granted in the context of a single **pluggable** database.



Common Users



Creating a User by a Common User

- A common user can create a common user or local user.
- The CONTAINER clause determines type of user created.

Create a **common** user in the root container:

```
SQL> CREATE USER CU_GEORGE IDENTIFIED BY x
  2> CONTAINER=ALL;
```

Create a **local** user in a PDB:

```
SQL> CREATE USER LU_FRED IDENTIFIED BY y
  2> CONTAINER=CURRENT;
```

Common and Local Privileges

- A privilege granted across all containers is a common privilege.
- A privilege granted in the context of a single PDB is a local privilege.
- Local users can only exercise privileges locally in the context of the PDB.
- Common users can only exercise privileges in the context of the PDB to which they are connected.
- Common users connected to the root container can exercise cross-container privileges, such as creating a common user.

Common and Local Roles

- A local user can create local roles.

```
SQL> CREATE ROLE LR_HR CONTAINER=CURRENT;
```

- A common user can create common roles

```
SQL> CREATE ROLE CR_CONNECT CONTAINER=ALL;
```

or local roles.

- Local roles can be granted to local or common users.
- Common roles can be granted to local or common users.

- Local roles can be granted to common roles.
- Common roles can be granted to local roles.

Grant and Revoke Privileges and Roles

Grant common privilege by common user

```
SQL> GRANT P1 TO common_user CONTAINER=ALL;
```

Grant local privilege by common user

```
SQL> GRANT P1 TO local_user CONTAINER=CURRENT;
```

Grant a local privilege by a local user

```
SQL> GRANT P1 TO local_user;
```

Revoke common privilege by common user

```
SQL> REVOKE P1 FROM common_user CONTAINER=ALL;
```

Revoke local privilege by common user

```
SQL> REVOKE P1 FROM local_user
  2  CONTAINER=CURRENT;
```

Revoke a local privilege by a local user

```
SQL> REVOKE P1 FROM local_user;
```

Creating Common and Local Objects

- Local tables can be created by common and local users
- Common tables can NOT be created by common users created by the customer.
- Common tables can be created by Oracle supplied users

Demo 1: Managing Common and Local Users

PROPERTIES

Allow user to leave interaction:	After viewing all the steps
Show ‘Next Slide’ Button:	Show upon completion
Completion Button Label:	Next Slide

Properties...

Edit in Engage

Demo 2: Managing Common and Local Roles and Privileges

PROPERTIES

Allow user to leave interaction:	Anytime
Show ‘Next Slide’ Button:	Show always
Completion Button Label:	Next Slide

Properties...

Edit in Engage

Lesson Summary

In this second lesson of the second part of the course, we discussed the management of the following entities in CDBs:

PART – II

Managing CDB

Managing PDB

- Permanent and temporary tablespaces in CDB and PBD
- Common and local users
- Common and local roles
- Common and local privileges
- Common and local objects

Lesson Review

You manage the tablespaces in a CDB and its PDBs. Choose the correct choice.

If you change the default tablespace for the CDB, it applies to all PDBs.

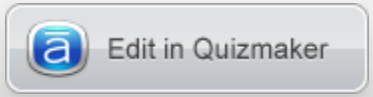
You can change the default tablespace of a PDB independently of its CDB default tablespace and all other PDBs.

The default temporary tablespace is necessarily the same for all containers.

If you create a common user with a default tablespace, the tablespace is not required in all PDBs, only in the root.

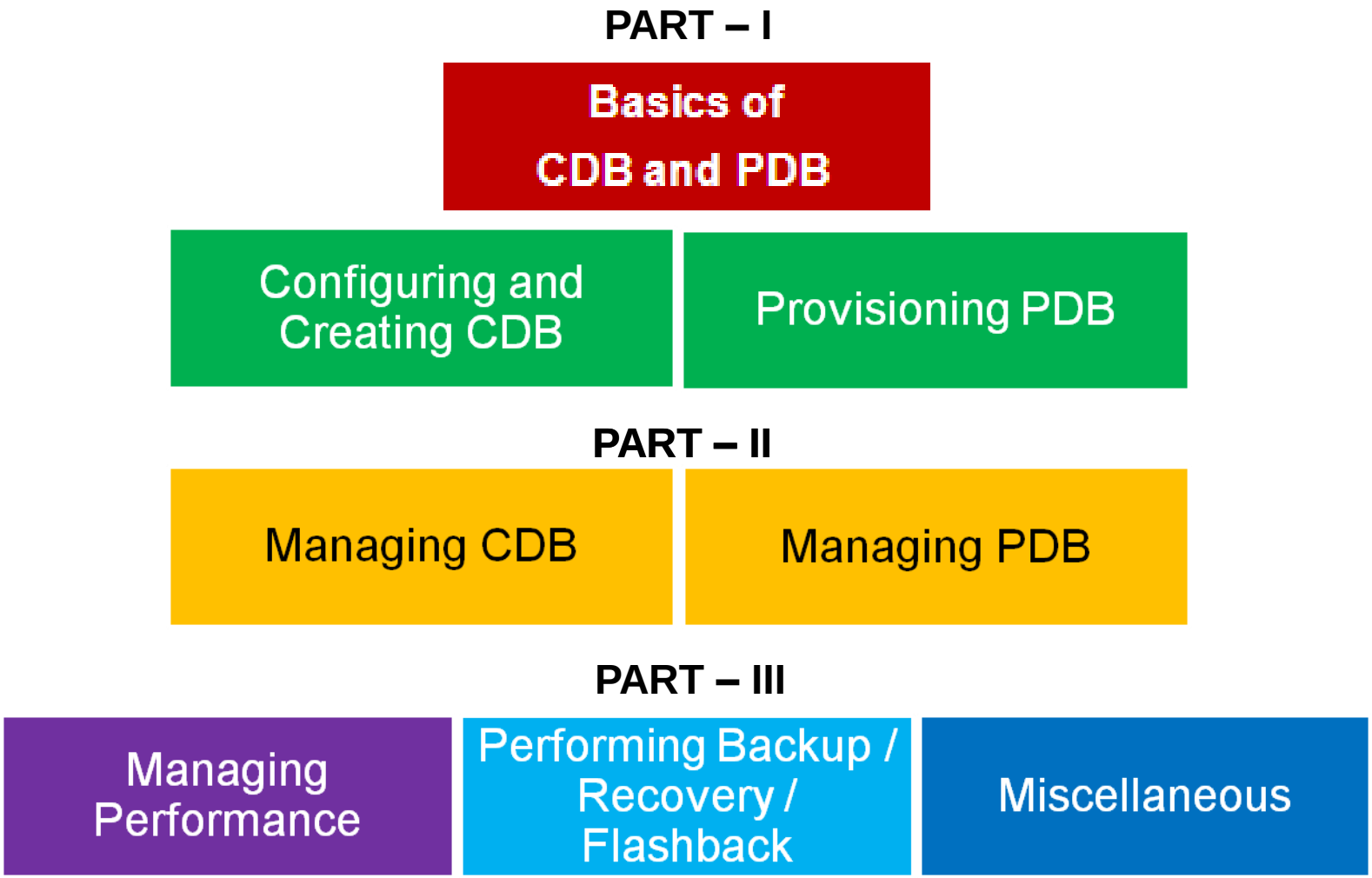
PROPERTIES

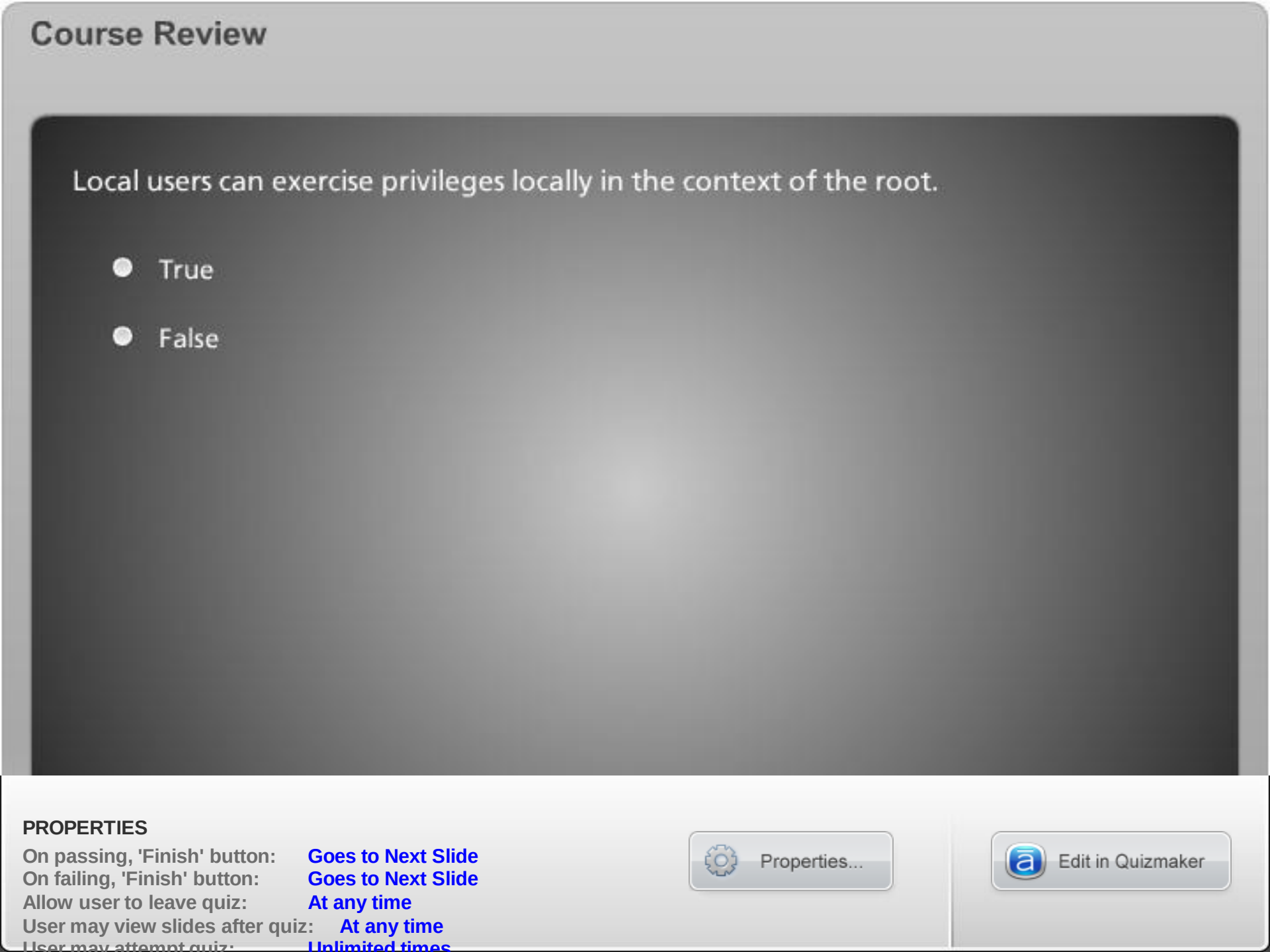
On passing, 'Finish' button: **Goes to Next Slide**
On failing, 'Finish' button: **Goes to Next Slide**
Allow user to leave quiz: **At any time**
User may view slides after quiz: **At any time**
User may attempt quiz: **Unlimited times**



Course Summary

In this second part of the course, we discussed:




Course Review
Local users can exercise privileges locally in the context of the root.
True
False
PROPERTIES
On passing, 'Finish' button: Goes to Next Slide
On failing, 'Finish' button: Goes to Next Slide
Allow user to leave quiz: At any time
User may view slides after quiz: At any time
Properties...
Edit in Quizmaker

How Can I Learn More?

Self-Study Courses

Instructor-led Courses

Other Resources

More Information

There are a variety of channels from which you can learn more about Oracle Database 12*c* New Features. Click on a tab on the left for more information about just a few of the possibilities.

We hope you found this self-study course informative and useful.

PROPERTIES

Allow user to leave interaction:	Anytime
Show 'Next Slide' Button:	Show upon completion
Completion Button Label:	Next Slide

Properties...

Edit in Engage