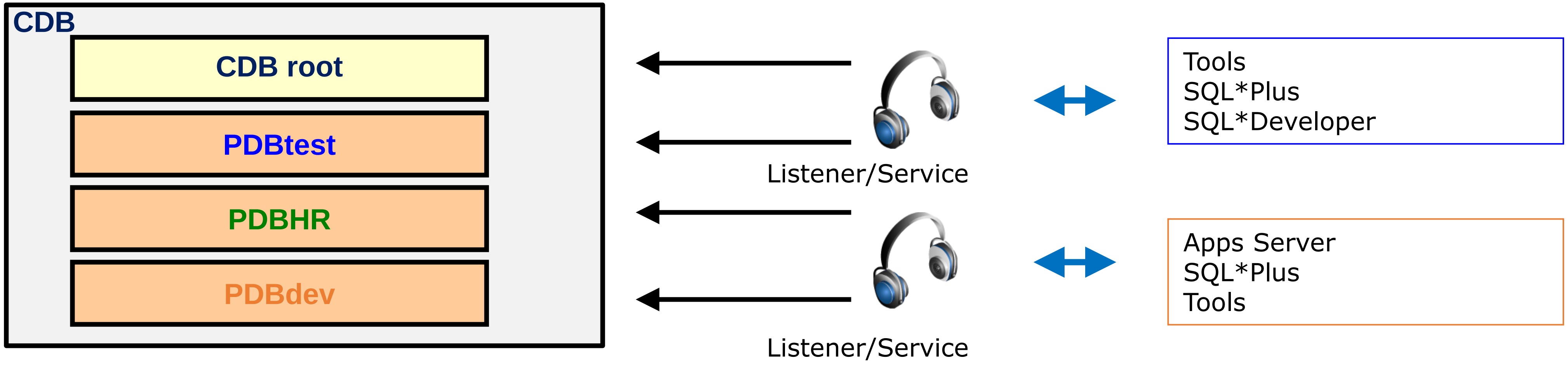
CDB and PDB Management

Objectives

- After completing this lesson, you should be able to:
  - Establish connections to a CDB / PDB
  - Avoid service name conflicts
  - Start PDB service
  - Start up and shut down a CDB
  - Open and close PDBs
  - Change the different modes and settings of PDBs
  - Evaluate the impact of parameter value changes
  - Configure host name and port number per PDB

Connection



CDB

CDB root

PDBtest

PDBHR

PDBdev

Listener/Service

Listener/Service

Tools
SQL*Plus
SQL*Developer

Apps Server
SQL*Plus
Tools

1.Every PDB has a default service.

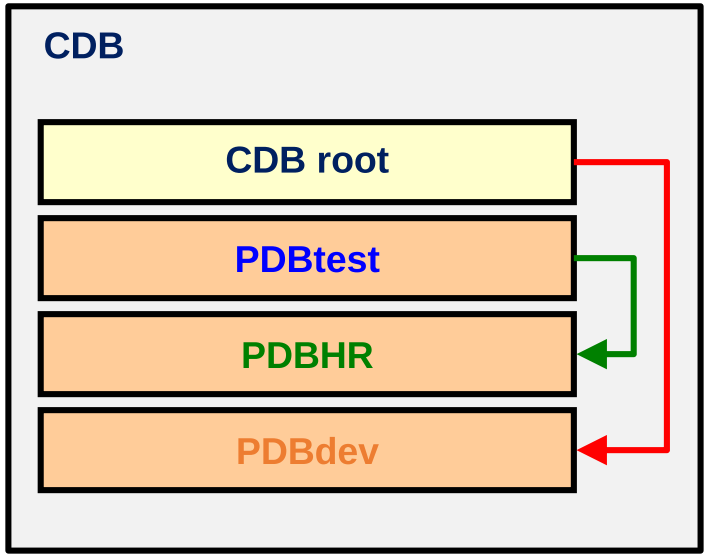
```
SQL> SELECT name, pdb FROM cdb_services;
```

2.Service name has to be unique across CDBs.

```
SQL> CONNECT / AS SYSDBA
SQL> CONNECT sys@CDB1 AS SYSDBA
SQL> CONNECT sys@PDBtest AS SYSDBA
SQL> CONNECT local_user1@hostname1:1525/PDBHR
SQL> CONNECT common_user2@PDBdev
SQL> SHOW CON_NAME
```

Switching Connection

- Two possible ways to switch connection between containers within a CDB:



CDB

CDB root

PDBtest

PDBHR

PDBdev

- Reconnect: Allows connection under common or local user

```
SQL> CONNECT / AS SYSDBA
SQL> CONNECT local_user1@PDBdev
```

- Use `ALTER SESSION SET CONTAINER` statement:

```
SQL> CONNECT sys@PDBtest AS SYSDBA
SQL> ALTER SESSION SET CONTAINER=PDBHR;
SQL> SHOW CON_NAME
SQL> ALTER SESSION SET CONTAINER=CDB$ROOT;
```

- Allows connection under common user only who is granted new system privilege `SET CONTAINER`.
  - `AFTER LOGON` triggers do not fire.
  - Transactions are still pending after switching containers.

Creating Services

- Using the **DBMS_SERVICE** package in an environment without Oracle Restart:

```
SQL> EXEC DBMS_SERVICE.CREATE_SERVICE('hrpdb','hrpdb')
```

```
SQL> EXEC DBMS_SERVICE.START_SERVICE('hrpdb')
```

- Using the **SRVCTL** utility in a Grid Infrastructure environment with Oracle Restart:

```
$ srvctl add service -db mycdb -service hrpdb -pdb hrpdb
```

```
$ srvctl start service -db mycdb -service hrpdb
```

- Oracle Restart configuration automatically updated:

Create operations and the Oracle Restart configuration	Automatically added to configuration?
Create a database service with SRVCTL	YES
Create a database service with DBMS_SERVICE.CREATE_SERVICE	NO

Renaming Services

- Renaming PDB service to avoid name conflicts

```
SQL> CREATE PLUGGABLE DATABASE pdb1 … FROM pdb1@link_node1
        SERVICE_NAME_CONVERT = ('pdb1_node1', 'pdb1_node2');
```

- Starting a PDB service at PDB opening

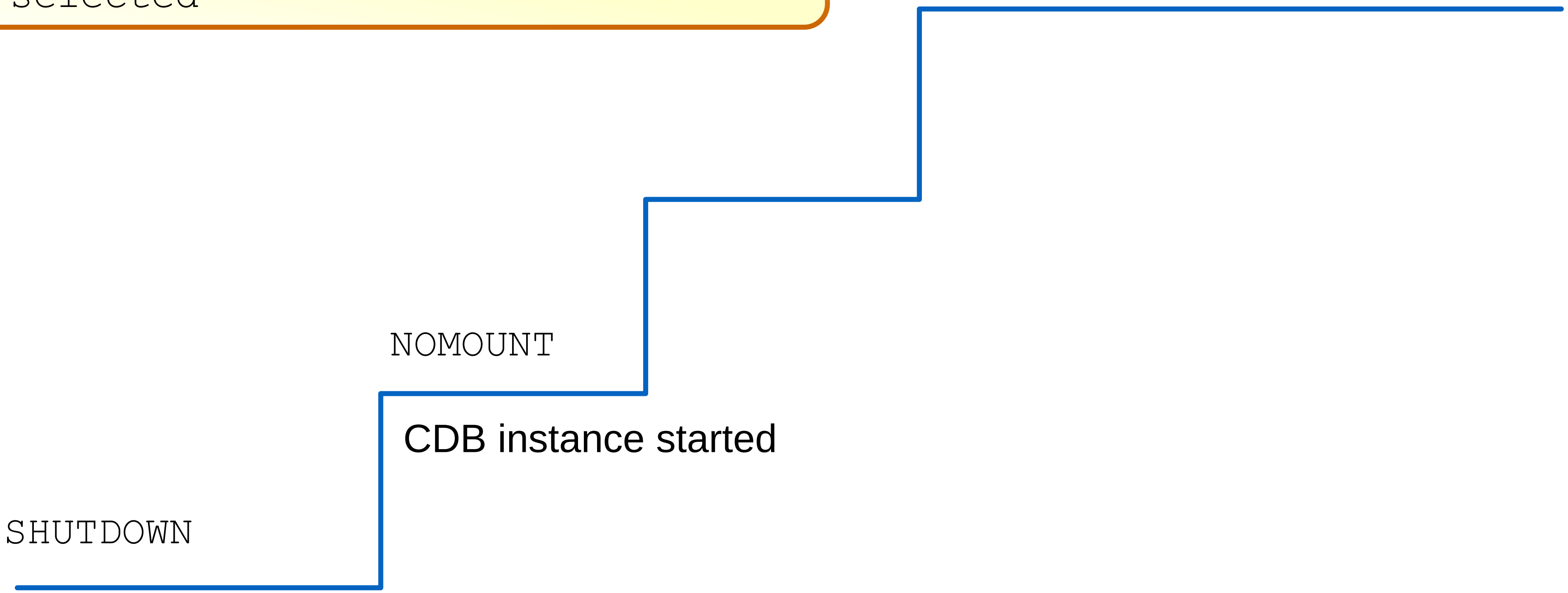
```
SQL> ALTER PLUGGABLE DATABASE pdb1 OPEN
        SERVICES = ('pdb1_node2');
```

Starting Up a CDB Instance

```
SQL> CONNECT sys@CDB1 AS SYSDBA
SQL> STARTUP NOMOUNT
```

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

no rows selected
```

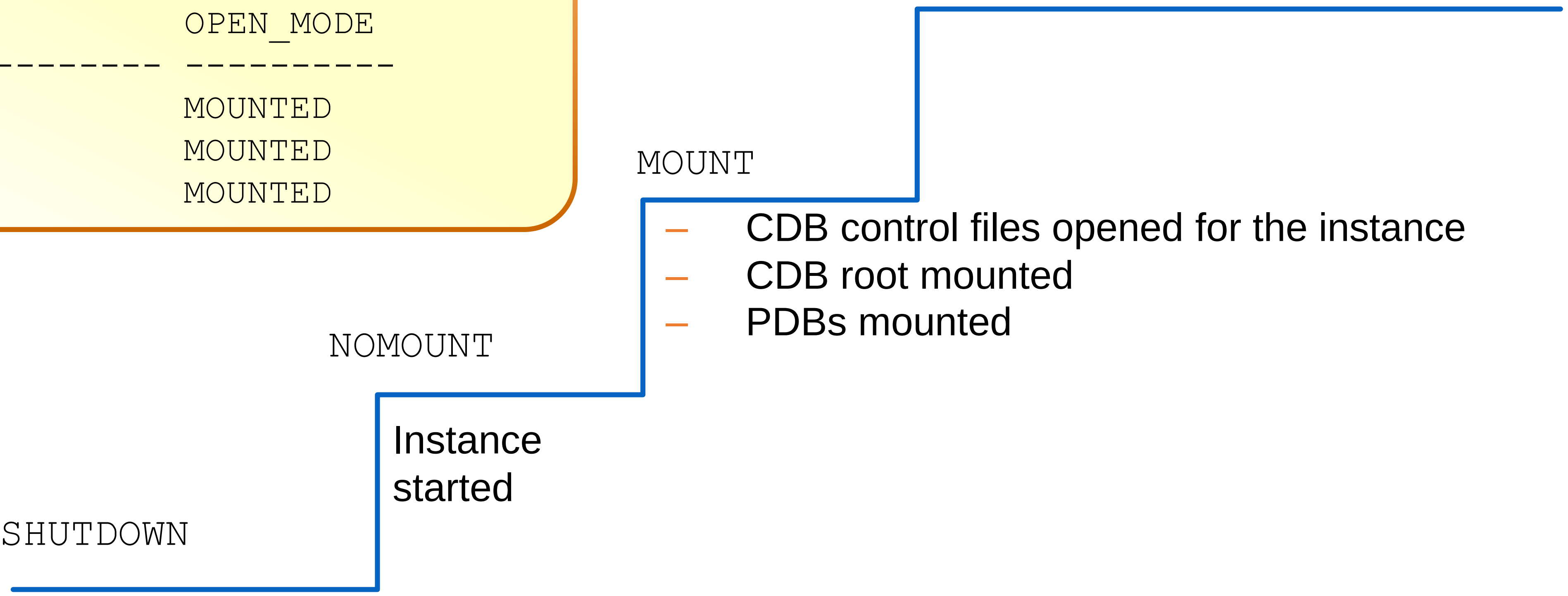


Mounting a CDB

```
SQL> CONNECT sys@CDB1 AS SYSDBA
SQL> STARTUP MOUNT
```

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

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



MOUNT

- CDB control files opened for the instance
- CDB root mounted
- PDBs mounted

NOMOUNT

Instance started

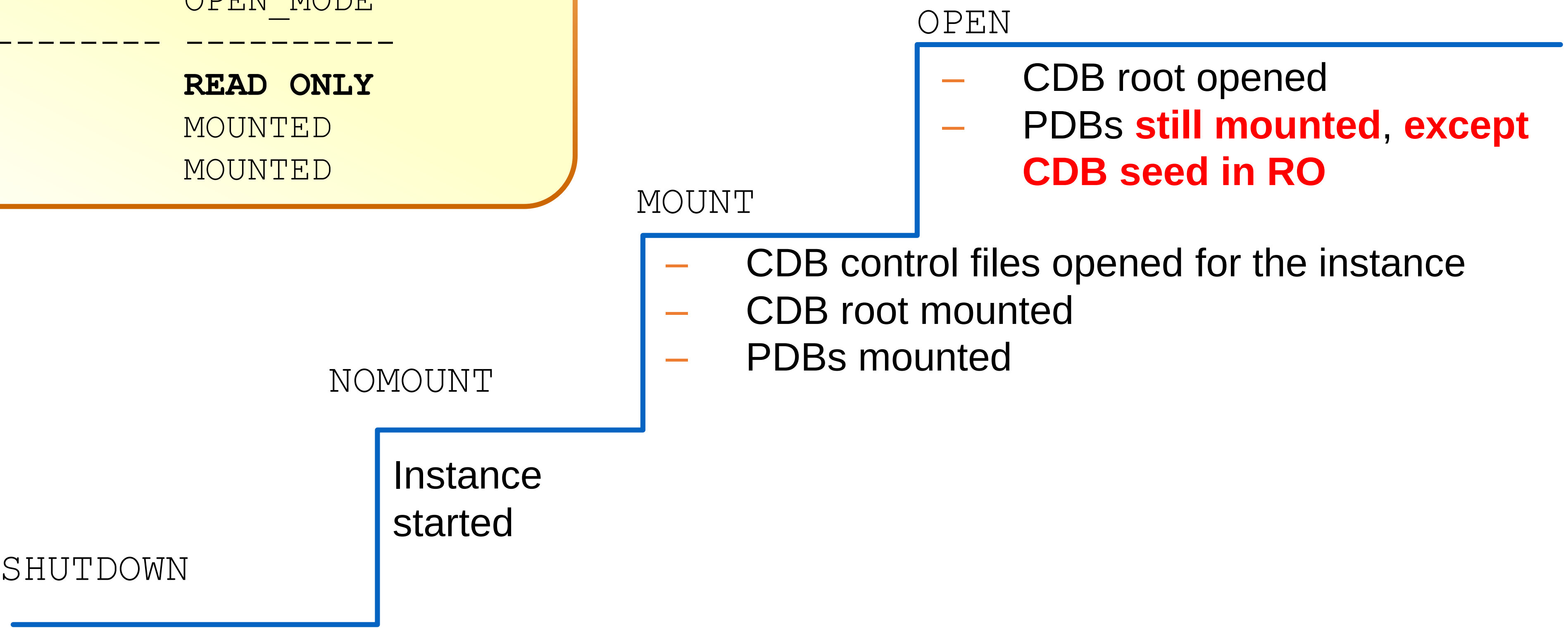
SHUTDOWN

Opening a CDB

```
SQL> STARTUP
```

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

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



SHUTDOWN

NOMOUNT

Instance started

MOUNT

- CDB control files opened for the instance
- CDB root mounted
- PDBs mounted

OPEN

- CDB root opened
- PDBs **still mounted**, **except CDB seed in RO**

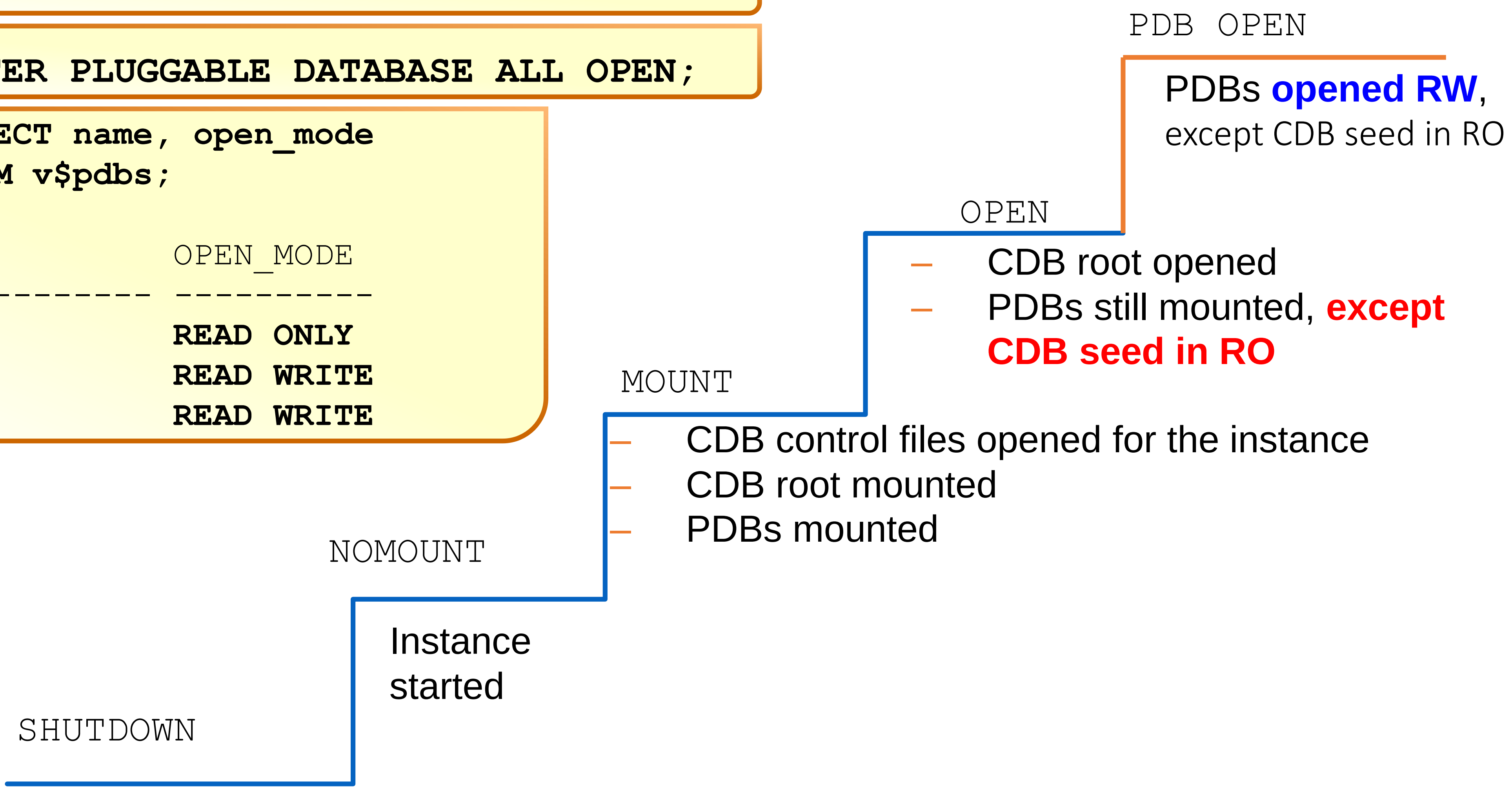
Opening a PDB

```
SQL> ALTER PLUGGABLE DATABASE pdb2 OPEN;
```

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

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

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



SHUTDOWN

NOMOUNT

Instance started

MOUNT

- CDB control files opened for the instance
- CDB root mounted
- PDBs mounted

OPEN

- CDB root opened
- PDBs still mounted, **except CDB seed in RO**

PDB OPEN

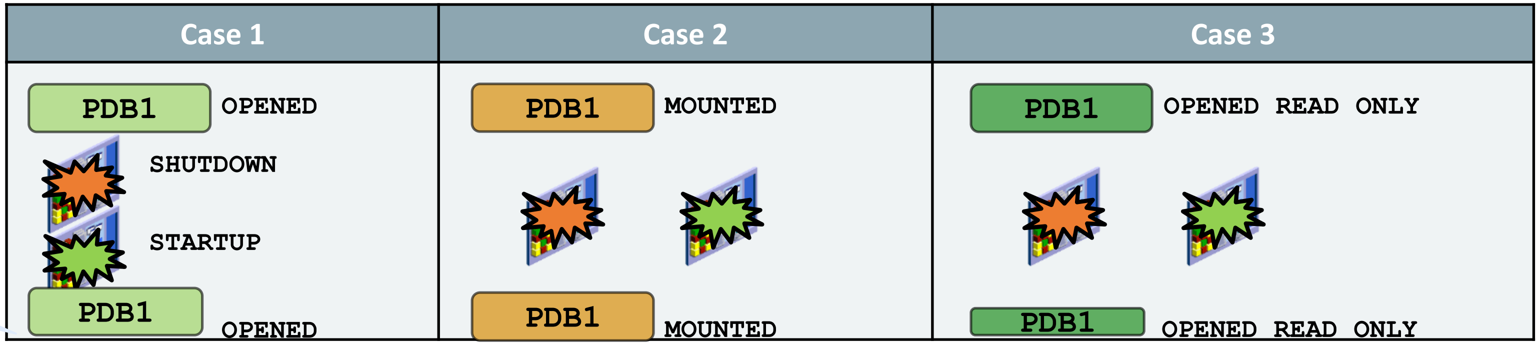
PDBs **opened RW**, except CDB seed in RO

Automatic PDB Opening

- Automatically keep PDBs state after `CDB STARTUP`:

```
SQL> ALTER PLUGGABLE DATABASE pdb1 SAVE STATE;
```

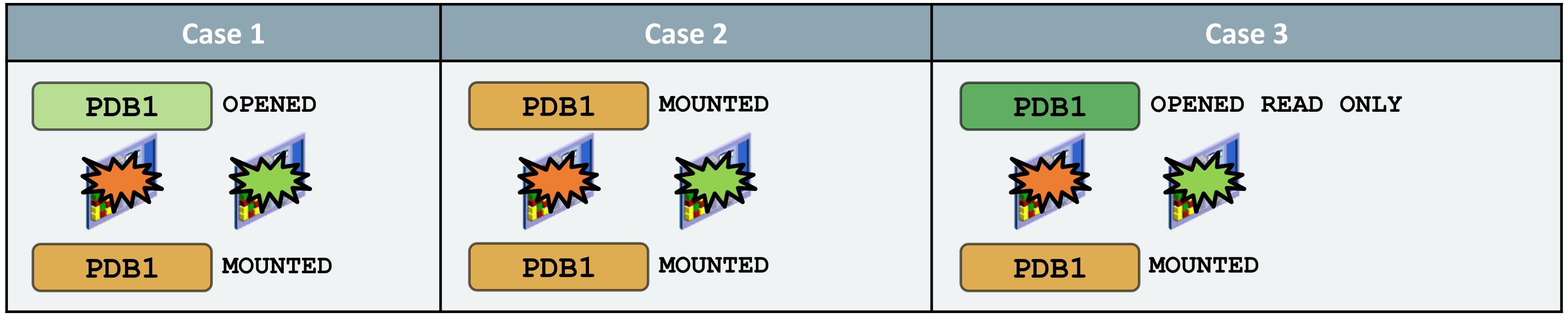
Automatic PDB opening



Case 1	Case 2	Case 3
PDB1 OPENED	PDB1 MOUNTED	PDB1 OPENED READ ONLY
SHUTDOWN STARTUP		
PDB1 OPENED	PDB1 MOUNTED	PDB1 OPENED READ ONLY

- Automatically discard PDBs state after `CDB STARTUP`:

```
SQL> ALTER PLUGGABLE DATABASE pdb1 DISCARD STATE;
```

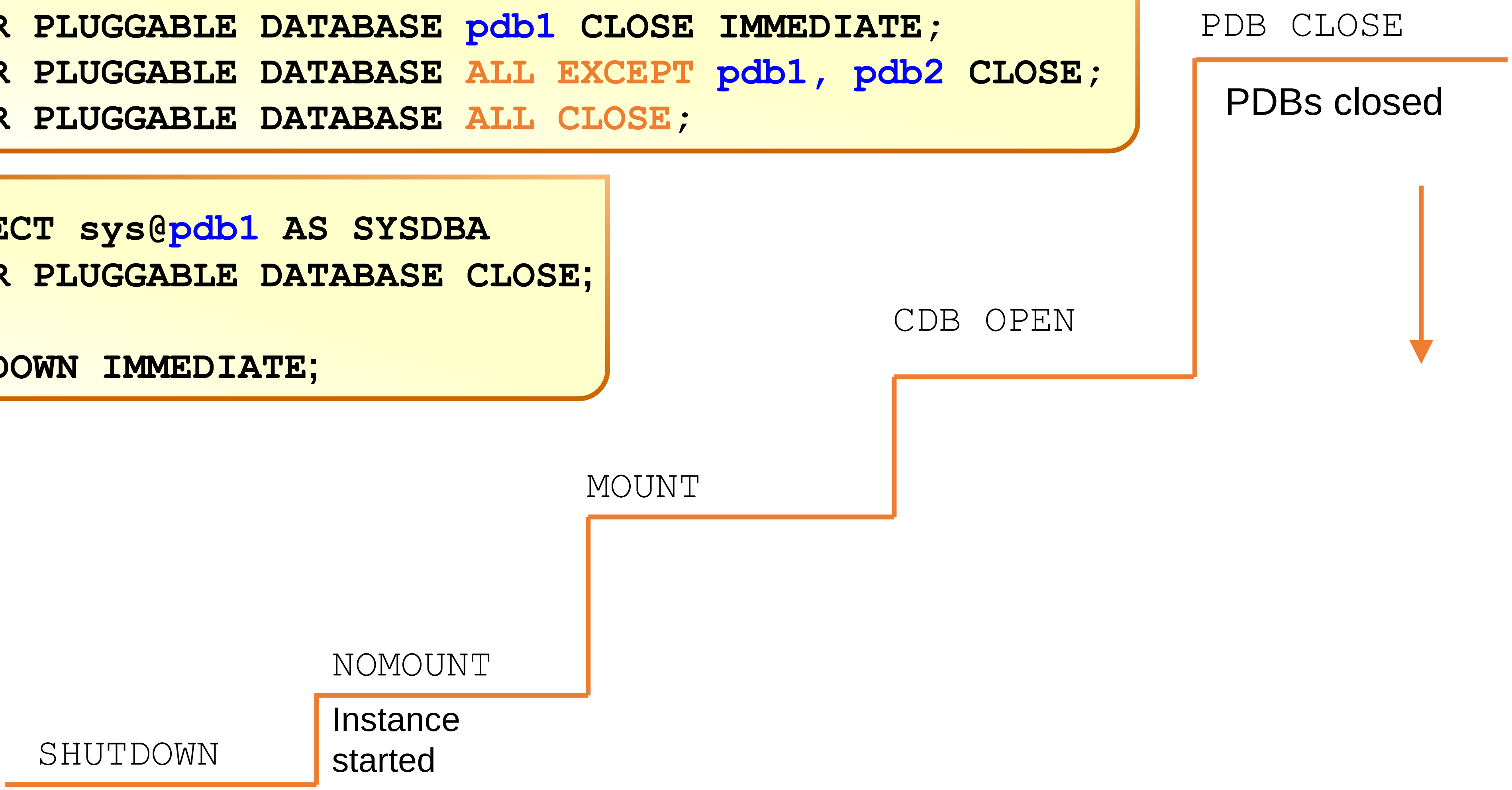


Case 1	Case 2	Case 3
PDB1 OPENED	PDB1 MOUNTED	PDB1 OPENED READ ONLY
PDB1 MOUNTED	PDB1 MOUNTED	PDB1 MOUNTED

Closing a PDB

```
SQL> CONNECT / AS SYSDBA
SQL> ALTER PLUGGABLE DATABASE pdb1 CLOSE IMMEDIATE;
SQL> ALTER PLUGGABLE DATABASE ALL EXCEPT pdb1, pdb2 CLOSE;
SQL> ALTER PLUGGABLE DATABASE ALL CLOSE;
```

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



Shutting Down a CDB Instance

```
SQL> CONNECT sys@CDB1 AS SYSDBA
SQL> SHUTDOWN IMMEDIATE
```

- All PDBs closed (no new specific message)
- CDB closed
- CDB dismounted
- Instance shut down

```
SQL> CONNECT sys@PDB1 AS SYSDBA
SQL> SHUTDOWN IMMEDIATE
```

- PDB closed

Changing PDB Mode

- After closing a PDB, open in:
  - Restricted read-write

```
SQL> CONNECT sys@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  RES
-------------------------- ---------- ---
PDB1                       READ WRITE YES
```

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

Modifying PDB Settings

- Bring a PDB datafile online
- Change the PDB default tablespace
- Change the PDB default temporary tablespace
- Set the PDB storage limit
- Change the global name

```
SQL> CONNECT sys@pdb1 AS SYSDBA
SQL> ALTER PLUGGABLE DATABASE DATAFILE '/u03/pdb1_01.dbf' ONLINE;
```

```
SQL> ALTER PLUGGABLE DATABASE DEFAULT TABLESPACE pdb1_tbs;
```

```
SQL> ALTER PLUGGABLE DATABASE DEFAULT TEMPORARY TABLESPACE temp_tbs;
```

```
SQL> ALTER PLUGGABLE DATABASE STORAGE (MAXSIZE 2G);
```

```
SQL> ALTER PLUGGABLE DATABASE RENAME GLOBAL_NAME TO pdbAPP1;
```

Instance Parameter Change Impact

- A single SPFILE per CDB
- PDB values change:
  - Loaded in memory after PDB close
  - Stored in dictionary after CDB shutdown
  - Only for parameter `ISPDB_MODIFIABLE=TRUE`

```
SQL> CONNECT sys@pdb1 AS SYSDBA
Connected.
SQL> ALTER SYSTEM SET ddl_lock_timeout=10;
System altered.
SQL> SHOW PARAMETER ddl_lock_timeout

NAME                                 TYPE        VALUE
------------------------------------ ----------- ---------
ddl_lock_timeout                     boolean     10
```

Instance Parameter Change Impact: Example

```
SQL> CONNECT sys@pdb2 AS SYSDBA

SQL> ALTER SYSTEM SET ddl_lock_timeout=20 SCOPE=BOTH;

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

```
SQL> CONNECT / AS SYSDBA
SQL> SELECT value, ispdb_modifiable, con_id FROM v$system_parameter
     WHERE  name = 'ddl_lock_timeout';

VALUE                ISPDB     CON_ID
-------------------- ----- ----------
0                    TRUE           0
10                   TRUE           3
20                   TRUE           4
```

Using ALTER SYSTEM Statement on PDB

- Some statements change the way a PDB operates:

ALTER SYSTEM Affecting the PDB only	Objects Impacted
ALTER SYSTEM FLUSH SHARED_POOL	Only for objects of the PDB
ALTER SYSTEM FLUSH BUFFER_CACHE	Only for buffers of the PDB
ALTER SYSTEM ENABLE/DISABLE RESTRICTED SESSION	Only for sessions of the PDB
ALTER SYSTEM KILL SESSION	Only for sessions of the PDB
ALTER SYSTEM SET *parameter*	Only for parameter of the PDB

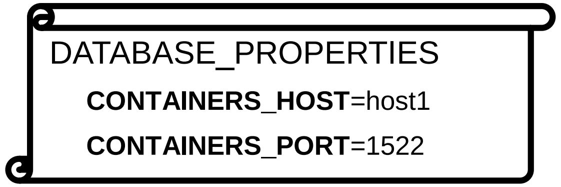
- Some ALTER SYSTEM statements can be executed in a PDB but affect the whole CDB:

ALTER SYSTEM CHECKPOINT	Affects all datafiles except those in read only or offline

- All other ALTER SYSTEM statements affect the entire CDB and must be run by a common user in the CDB root.

ALTER SYSTEM SWITCH LOGFILE	Operation not allowed from within a pluggable database

Configuring Host Name and Port Number per PDB



DATABASE_PROPERTIES

CONTAINERS_HOST=host1

CONTAINERS_PORT=1522

- The host name and port number settings for a PDB are important only if proxy PDBs will reference the PDB.

```
SQL> ALTER PLUGGABLE DATABASE CONTAINERS HOST = <host_name>;
SQL> ALTER PLUGGABLE DATABASE CONTAINERS PORT = <port_nb>;
```

- The host name and port number can be reset to their default:

```
SQL> ALTER PLUGGABLE DATABASE CONTAINERS HOST RESET;
SQL> ALTER PLUGGABLE DATABASE CONTAINERS PORT RESET;
```

Summary

- In this lesson, you should have learned how to:
  - Establish connections to a CDB / PDB
  - Avoid service name conflicts
  - Start PDB service
  - Start up and shut down a CDB
  - Open and close PDBs
  - Change the different modes and settings of PDBs
  - Evaluate the impact of parameter value changes
  - Configure host name and port number per PDB

Practice 5: Overview

- 5-1: Starting up and shutting down a CDB
- 5-2: Opening and closing PDBs
- 5-3: Renaming a PDB
- 5-4: Setting parameter values for PDBs
- 5-5: Renaming PDB services