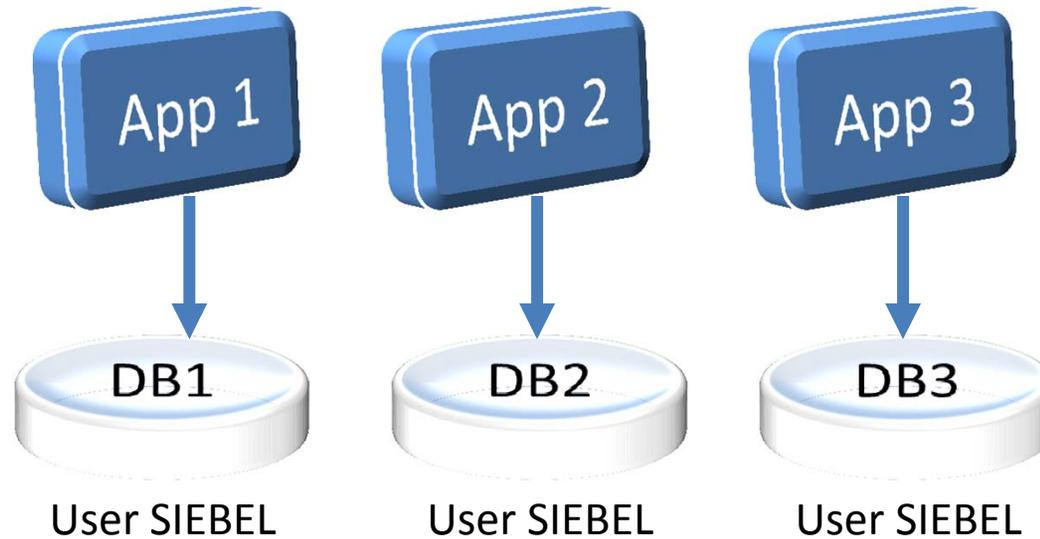


# Multitenant Databases

**Arup Nanda**

*Longtime Oracle DBA*

# One App: One DB



# Database User Issue

*Application 1*



User SIEBEL

*Application 2*



User SIEBEL

*Application 3*



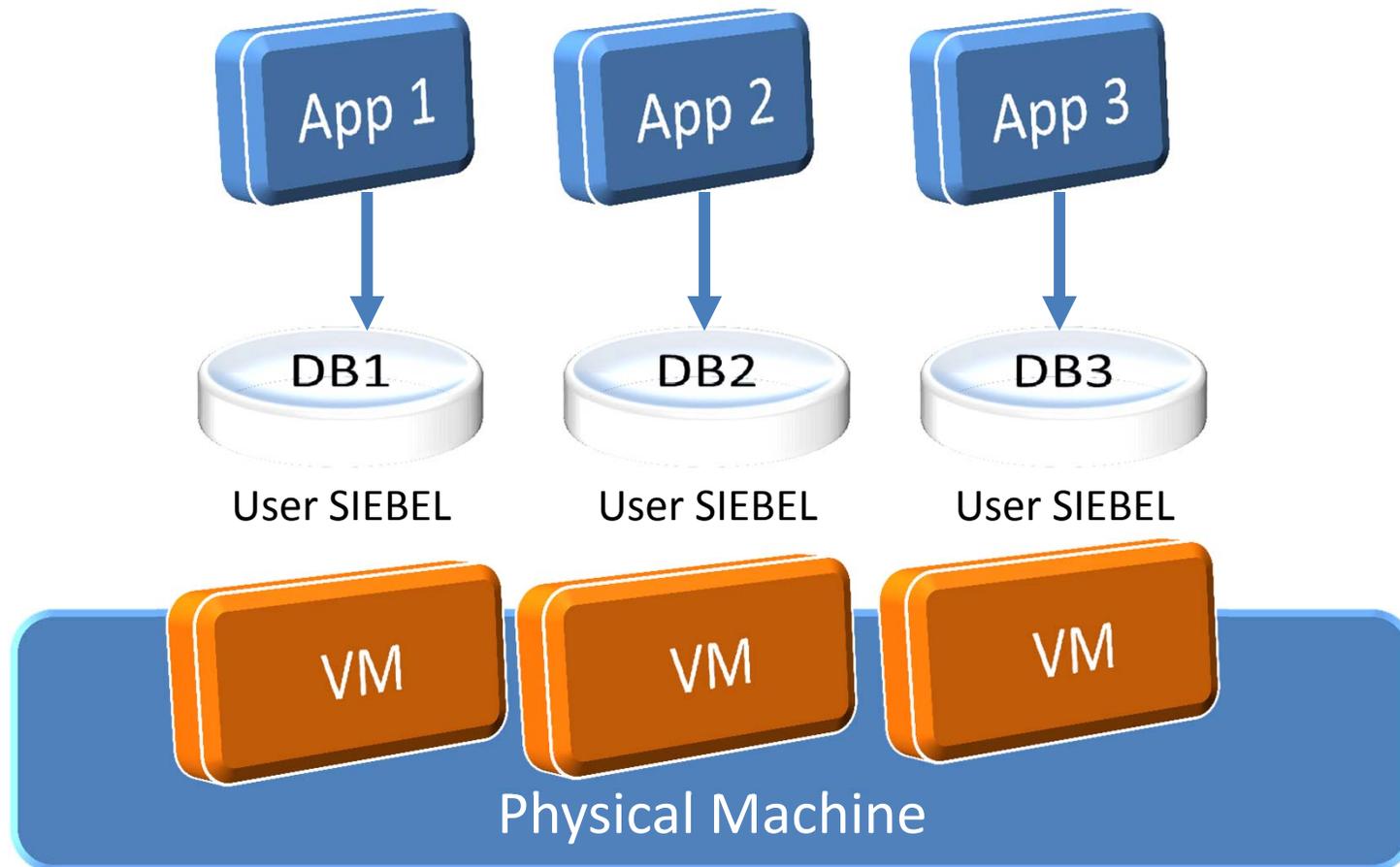
User SIEBEL

*Application 1*  
*Application 2*  
*Application 3*

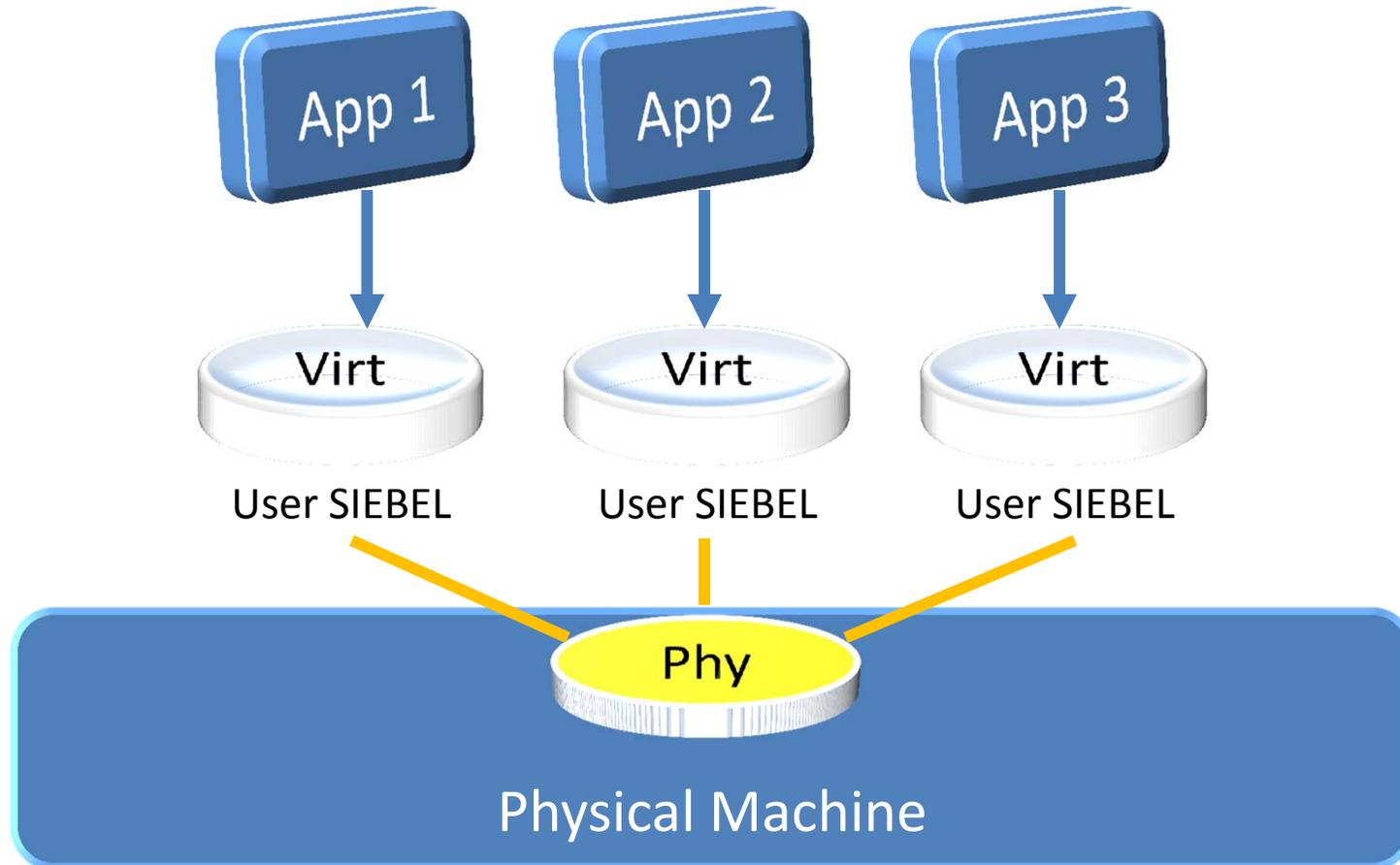


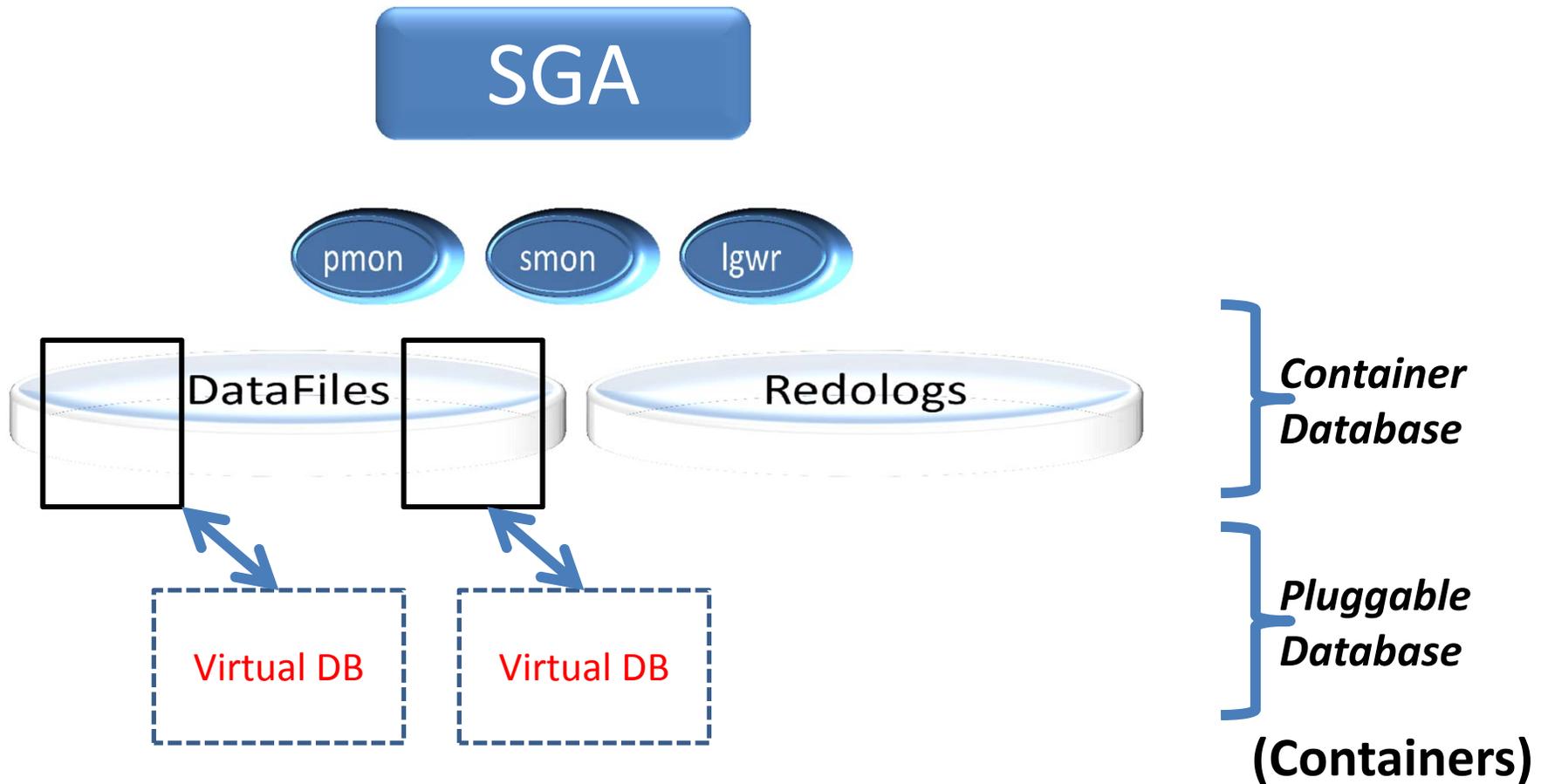
User SIEBEL

# Host Virtualization



# Database Virtualization





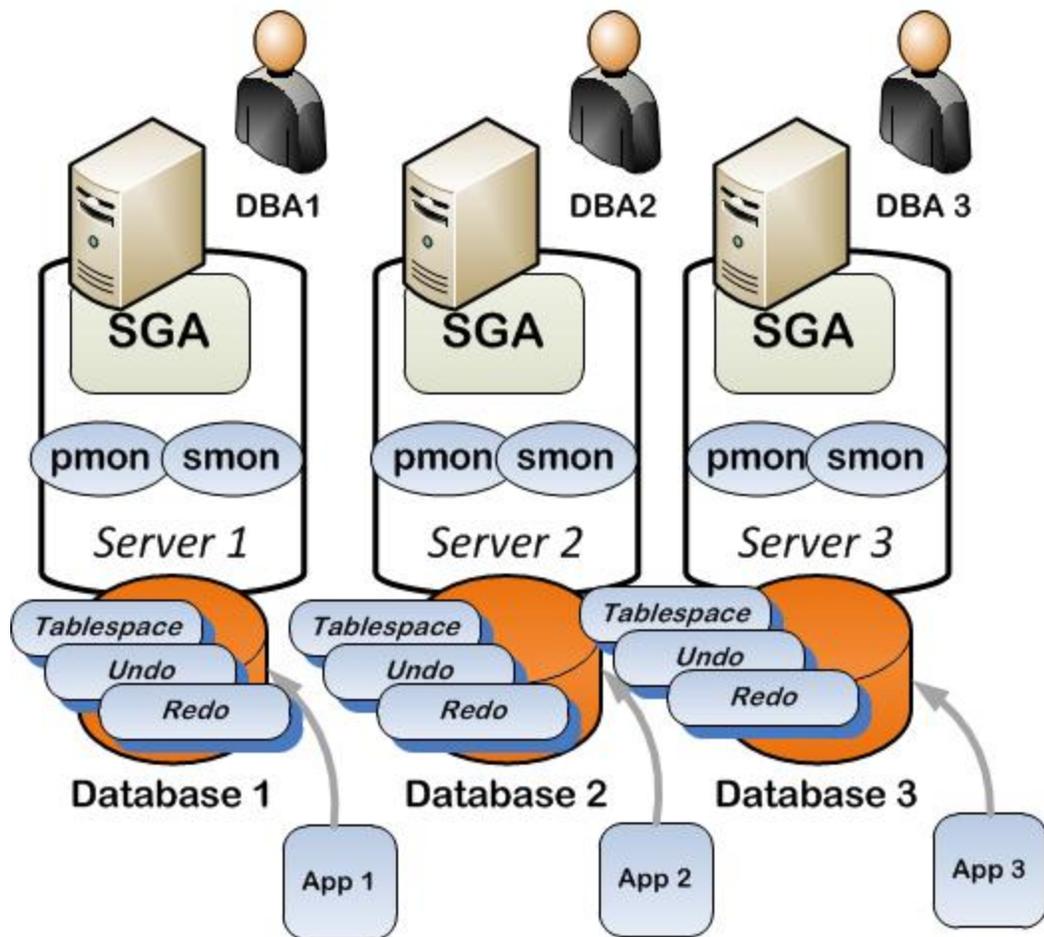
## DBA\_USERS

```
SELECT NAME  
FROM USER$  
WHERE CON_ID = ...
```

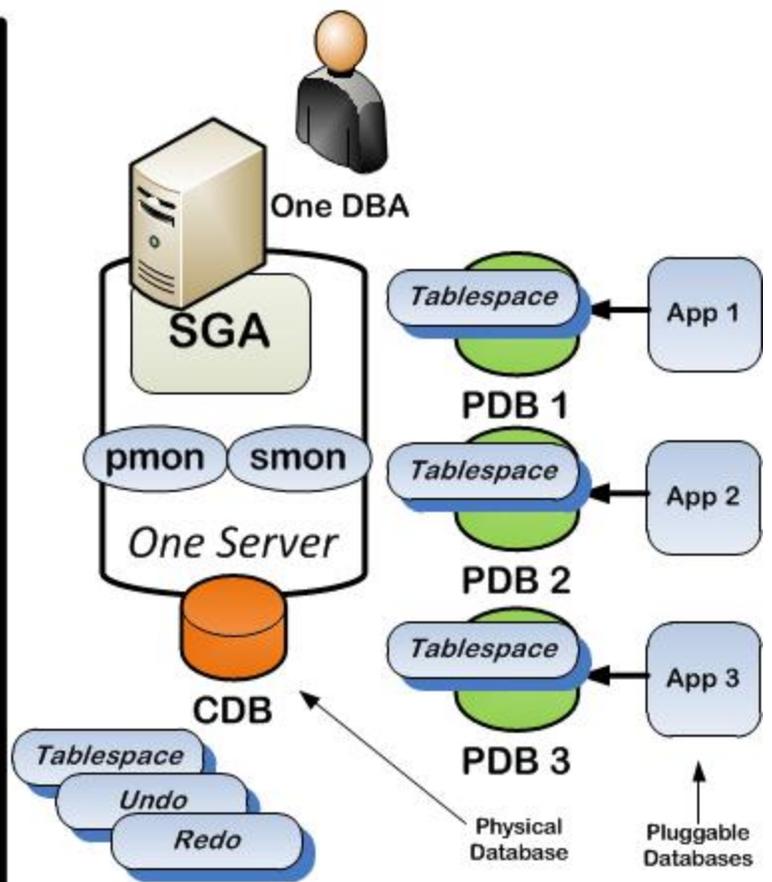
PDB1  
CON\_ID=2

PDB2  
CON\_ID=3

PDB3  
CON\_ID=4



*Before Consolidation*



*After Consolidation*

# DBCA can Create it

The screenshot displays the Oracle Database Configuration Assistant (DBCA) interface for Oracle Database 12c. The window title is "Database Configuration Assistant - Manage Pluggable Databases - Step 2 of 7".

**Manage Pluggable Databases**

Select an operation that you want to perform in container database:

- Create a Pluggable Database
- Unplug a Pluggable Database
- Delete a Pluggable Database
- Configure a Pluggable Database

**Pluggable Database Options**

The "Pluggable Database Options" screen shows the following configuration:

- Identification** (selected tab)
- Pluggable Database Name:** PDB2
- PDB Storage**
  - Use Oracle Managed Files
  - Specify Common Location
  - Create Default User Tablespace
- PDB User**
  - Administrator Username:** syspdb2
  - Administrator Password:** [masked]
  - Confirm Administrator Password:** [masked]

# CDB –vs- PDB

CDB  
(Physical Database)

Background processes –  
pmon, smon, etc.

Memory areas – buffer  
cache, log buffer, etc.

Datafiles

Undo tablespace

Single ADR location

PDB1  
(Virtual Database)

Some datafiles

PDB2  
(Virtual Database)

Some datafiles

PDB3  
(Virtual Database)

Some datafiles

# Checking for CDB

- V\$DATABASE has a new column: CDB
- `select cdb from v$database;`
- YES – is a CDB
- NO – is not

`iscdb.sql`

# How many PDBs

- V\$PDBS
  - CON\_ID: container ID
  - DBID: DBID of the PDB
  - NAME: name of the PDB
  - OPEN\_MODE: how it is open
  - OPEN\_TIME: when it was opened
  - CREATE\_SCN: the SCN number it was created at
  - TOTAL\_SIZE – the incremental size of the PDB
    - (zero if not mounted)

pdb.sql

# Basic PDB Operations

- While connected to CDB

- To open a PDB:

```
alter pluggable database pdba open;
```

open.sql

- To close a PDB

```
alter pluggable database pdba close;
```

Close.sql

- To create a PDB

```
create pluggable database pdbc  
admin user sysc identified by pdbc
```

# Clone

- The source PDB must be in read only mode

- Clone

Clone.sql

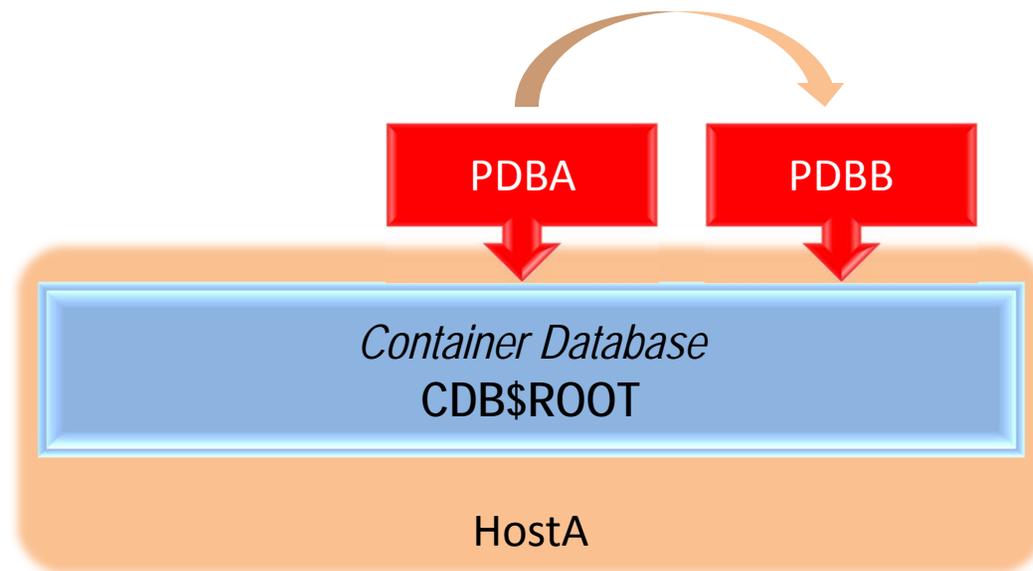
```
create pluggable database pdbb
```

```
from pdba
```

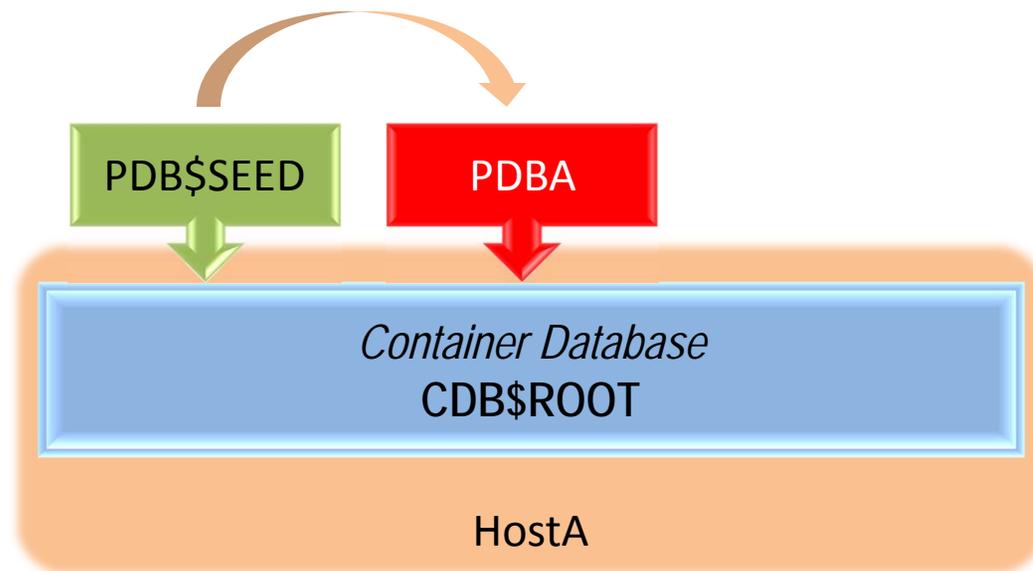
```
file_name_convert=('pdba', 'pddb');
```

- The new PDB will be mounted. You need to open it.

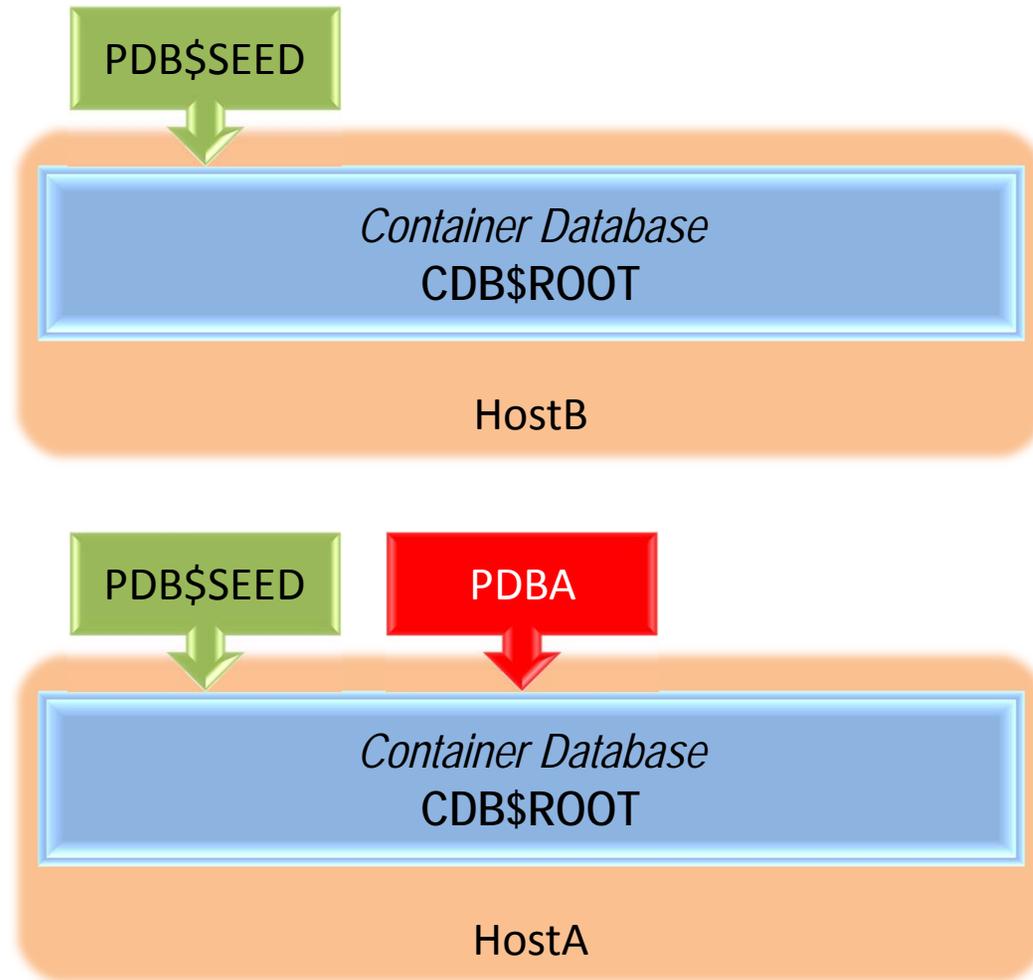
# Cloning



# Seed PDB



# Clone from Foreign Host



# Clone from Foreign Host

# Connection

- Three choices:

1. Alter Session

```
SQL> connect / as sysdba (CDB)
```

```
SQL> alter session set container = pdba;
```

2. Service Name

```
sqlplus sysc/pdbc@host:1522/PDBC
```

```
sqlplus sysc/pdbc@pdbc (in tnsnames.ora)
```

3. Two Task

```
set TWO_TASK=PDBC
```

```
sqlplus user/pass
```

# Checking the PDB

- Two SQL\*Plus parameters

Show CON\_ID

Show CON\_NAME

- Using SYS\_CONTEXT

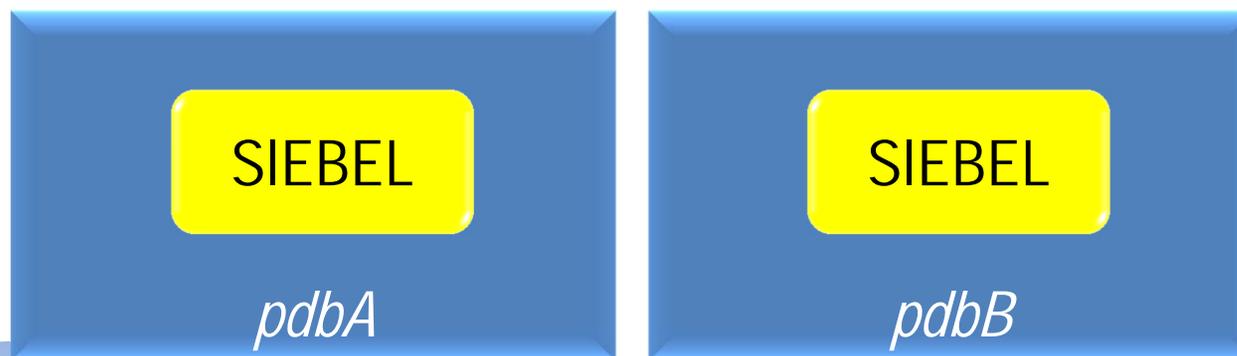
Context.sql

```
select sys_context('userenv','CON_ID') from  
dual;
```

# Local Users

- Created in a specific PDB
- Only in that particular PDB
  - Users with the same name could exist in many PDBs; but they are all different users

```
Cr_siebela.sql  
Cr_siebelb.sql  
Cr1.sql  
Cr2.sql
```

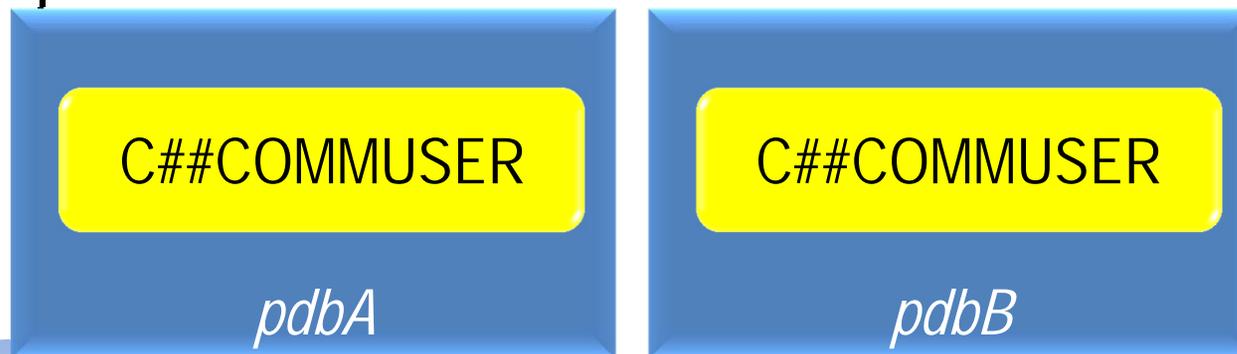


# Common Users

- Defined in all PDBs

```
create user c##commuser identified by mypass  
container=all;  
grant create session to c##commuser  
container=all;
```

- Need the C## prefix. They are all different users Commuser.sql
- The password must be same.



# Backup

- You can backup all PDBs at once  
\$ rman target=/ connected to CDB
- Or, one specific PDB  
\$ rman target=sys/oracle@pdba
- Issue the following to check:  
RMAN> report schema;

# Different Parameters

- Between PDBs
  - Different values of some system parameters
    - e.g. optimizer\_index\_cost\_adj
  - Some parameters are not changeable
    - e.g. audit\_trail

Altsess1.sql  
Altsess2.sql

- Check:

```
select name, ispdb_modifiable  
from v$parameter  
where name in (  
'optimizer_index_cost_adj',  
'audit_trail');
```

Pdbparms.sql

# Same, yet Different

- Same
  - Oracle Version
  - Characterset
  - Redo, Undo TS
  - Blocksize
- Different
  - Timezones
  - Data, System Tablespaces
  - Temporary Tablespaces

# Services

- A service in the name of the PDB is created by default
- If you want to add a new service, you have to add that to the PDB using `srvctl`.

```
$ srvctl add service -db CONA -s SERV1 -pdb PDBA
```

- Services are unique in a CDB
  - You can't create another service called `SERV1` in a different PDB

# Checking for Service

```
[oracle@prosrv1 ~]$ srvctl config service -db CONA -s SERV1
```

Service name: SERV1Service is enabled

Cardinality: SINGLETON

Disconnect: false

Service role: PRIMARY

Management policy: AUTOMATIC

DTP transaction: false

AQ HA notifications: false

Global: false

Commit Outcome: false

Failover type:

Failover method: TAF

failover retries:TAF

failover delay:

Connection Load Balancing Goal: LONG

Runtime Load Balancing Goal: NONE

TAF policy specification: NONE

Edition:

Pluggable database name: PDB1

Maximum lag time: ANY

SQL Translation Profile:

Retention: 86400 seconds

Replay Initiation Time: 300 seconds

Session State Consistency:

# Resource Manager

```
begin

dbms_resource_manager.clear_pending_area();
dbms_resource_manager.create_pending_area();
  -- create the CDB resource plan
  dbms_resource_manager.create_cdb_plan(
    plan    => 'dayshift_cona_plan',
    comment => 'cdb plan for cona'
  );

-- give the limits in the plan for PDB1
dbms_resource_manager.create_cdb_plan_directive(
  plan                => 'dayshift_cona_plan',
  pluggable_database => 'pdb1',
  shares              => 2,
  utilization_limit   => 100,
  parallel_server_limit => 100
);

  -- and, now the same for PDB2
  dbms_resource_manager.create_cdb_plan_directive(
    plan                => 'dayshift_cona_plan',
    pluggable_database => 'pdb2',
    shares              => 1,
    utilization_limit   => 50,
    parallel_server_limit => 50
  );

  -- and now, PDB3
  dbms_resource_manager.create_cdb_plan_directive(
    plan                => 'dayshift_cona_plan',
    pluggable_database => 'pdb3',
    shares              => 1,
    utilization_limit   => 70,
    parallel_server_limit => 70
  );

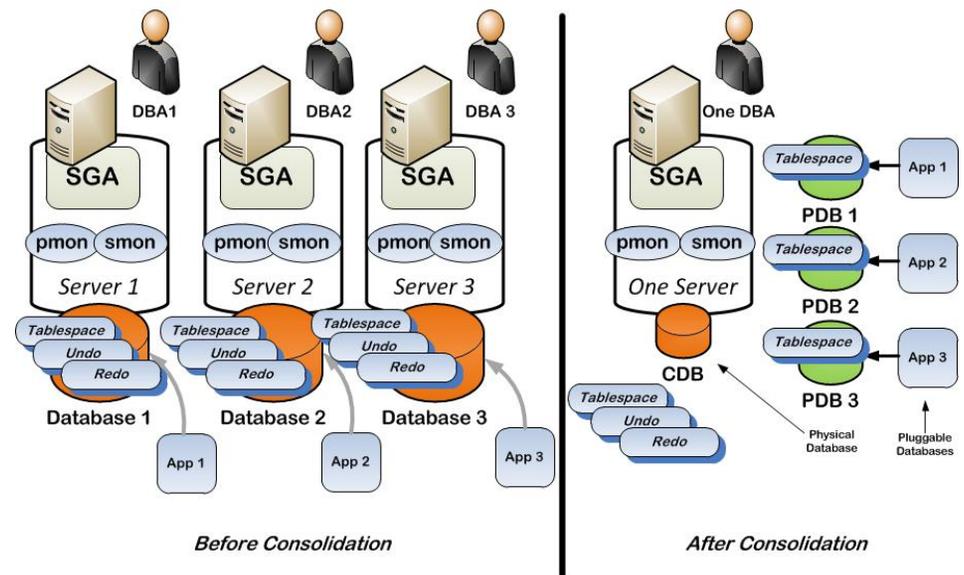
  dbms_resource_manager.validate_pending_area();
  dbms_resource_manager.submit_pending_area();
end;
/
```

# Is it Difficult?

Activity	Traditional Approach	PDB Approach
Connection from App Server	<code>sqlplus u/p@serv1</code>	<code>sqlplus u/p@serv1</code>
Connection on the DB Server	<code>sqlplus u/p</code>	<code>sqlplus u/p@serv1</code> <code>set two_task=serv1</code> <code>sqlplus u/p</code>
Connection by DBA	<code>Sqlplus / as sysdba</code>	<code>Sqlplus as sysdba for CDB</code> <code>Sqlplus sys/p@serv1 as sysdba</code>
Backup	<code>Rman target=/</code>	<code>Rman target=/ (CDB)</code> <code>Rman target=sys/p@serv1 (PDB)</code>
Recovery	<code>Restore database</code>	<code>Restore database; (CDB)</code> <code>Restore pluggable database; (PDB)</code>

# Why do it?

- One physical database
  - One set of memory
  - One set of processes
  - Less number of DBAs
- PDBs don't take anything other than space
- Upgrade to CDB upgrades all PDBs



# Why do it, contd.?

- Creation is super fast
- Cloning is fast and easy
- Cloning with storage snapshots is even faster and easier
- Cloning to different servers
- Resource manager really works
- Dictionary tables and views don't need to change
- Scripts and apps don't need to change
- Extra cost
  - One PDB inside a CDB is free

# Lesson learned

1. Apps didn't see any different
2. Clones are perfect to ramp up test envs
3. Clones with storage snapshots are perfect for refreshes of non-prod from prod
4. Point in time Recovery for PDBs perfect for rewinding
5. Confusion about connecting to CDB, or PDB by DBAs
6. PDBs need to be opened explicitly



# *Thank You!*

**My Blog: [arup.blogspot.com](http://arup.blogspot.com)**

**My Tweeter: [arupnanda](#)**