Experience as a Multitenant DBA

Arup Nanda
Longtime Oracle DBA

3. Big Questions

1. What do I need to learn extra?
2. Do I need to change anything?
3. May something break?
Agenda

- Quick primer on Oracle Multitenant
- Developer Activities Affected
- DBA Activities Affected
- Tips and Tricks

Vexing Problem of Database Consolidation

Application 1
  - DB1
  - User SIEBEL

Application 2
  - DB2
  - User SIEBEL

Application 3
  - DB3
  - User SIEBEL

Application 1
  - DBX
  - User SIEBEL
Enter: Pluggable Database

SGA

DataFiles

Redologs

Virtual DB

Virtual DB

Container Database

Pluggable Database

(Containers)

Arup Nanda

Experience as a Multitenant DBA

```
SELECT 1 AS CON_ID, NAME
FROM USER$@cdb
UNION ALL
SELECT 2 AS CON_ID, NAME
FROM USER$@pdb1
UNION ALL
SELECT 3 AS CON_ID, NAME
FROM USER$@pdb2
```

Root
CON_ID=1

PDB2
CON_ID=3

PDB3
CON_ID=4

Arup Nanda

Experience as a Multitenant DBA
• CDB specific:
  – Alert Log
  – Redo Logs
  – Undo Tablespaces
  – SGA
  – ADR (Automatic Diagnostic Repository)
  – Characterset
  – Block size
  – Most pfile parameters

• PDB specific:
  – Additional datafiles (including system)
  – Some PDB-specific parameters

Parameters can be different

• View V$PARAMETER column ISPDB_MODIFIABLE shows if a parameter is modifiable
  select name, value from v$parameter where ispdb_modifiable = 'TRUE'

• Example:
  – parallel_degree_policy is PDB modifiable
  – result_cache_max_size is not PDB modifiable

• PDB does not have a SPFILE
• These parameters are stored in PDB_SPFILE$
Creating PDBs

- (In CDB) Logon to SQL*Plus as SYSDBA

```sql
SQL> create pluggable database PLUG1 admin user plug1admin identified by plug1admin;
```

Basics of PDBs

Is this a CDB? Or a non-CDB?

```sql
SQL> select cdb from v$database;
```

- CDB
- YES

How many PDBs?

```sql
SQL> select name from v$pdb;
```

- NAME
- PDB$SEED
- PDBORCL
- PLUG1

```sql
SQL> desc v$pdb
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Null?</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CON_ID</td>
<td>NUMBER</td>
<td></td>
</tr>
<tr>
<td>DBID</td>
<td>NUMBER</td>
<td></td>
</tr>
<tr>
<td>CON_UID</td>
<td>NUMBER</td>
<td></td>
</tr>
<tr>
<td>GUID</td>
<td>RAW(16)</td>
<td></td>
</tr>
<tr>
<td>NAME</td>
<td>VARCHAR2(30)</td>
<td></td>
</tr>
<tr>
<td>OPEN_MODE</td>
<td>VARCHAR2(10)</td>
<td></td>
</tr>
<tr>
<td>RESTRICTED</td>
<td>VARCHAR2(3)</td>
<td></td>
</tr>
<tr>
<td>OPEN_TIME</td>
<td>TIMESTAMP(3) WITH TIME ZONE</td>
<td></td>
</tr>
<tr>
<td>CREATE_SCN</td>
<td>NUMBER</td>
<td></td>
</tr>
<tr>
<td>TOTAL_SIZE</td>
<td>NUMBER</td>
<td></td>
</tr>
<tr>
<td>BLOCK_SIZE</td>
<td>NUMBER</td>
<td></td>
</tr>
<tr>
<td>RECOVERY_STATUS</td>
<td>VARCHAR2(8)</td>
<td></td>
</tr>
<tr>
<td>SNAPSHOT_PARENT_CON_ID</td>
<td>NUMBER</td>
<td></td>
</tr>
</tbody>
</table>
Service Name

- Creates a default service in the database (CDB) with the same name as PDB
- Listener listens to this service:
  
  `$ lsnrctl status`

  ...  
  Service "plug1" has 1 instance(s).
  Instance "cdborcl1", status READY, has 1 handler(s) for this service...

- But that service is not defined in the database
  
  SQL> show parameter service
  NAME TYPE VALUE
  -- -------- ----- -------
  service_names string CDBORCL

How do you Connect to a PDB?

- By default it connects to the “Root” container.
  
  `$ sqlplus / as sysdba`

- Three ways to connect. First approach:
  
  The SET CONTAINER Clause
  
  SQL> alter session set container = PLUG1;
Connect to Service Name

Put in TNSNAMES.ORA file, SERVICE_NAME = PLUG1

```
PLUG1 =
    (DESCRIPTION =
        (ADDRESS =
            (PROTOCOL = TCP)(HOST = host1)(PORT = 1521)
        )
        (CONNECT_DATA =
            (SERVER = DEDICATED)
            (SERVICE_NAME = PLUG1)
        )
    )

$ sqlplus scott/tiger@plug1
```

Connecting from Applications

- Connect through a connect string:
  ```
  $ sqlplus scott/tiger@mydb
  ```
  where `mydb` is a connect string in TNSNAMES.ORA

```
mydb =
    (DESCRIPTION =
        (ADDRESS =
            (PROTOCOL = TCP)
            (HOST = host1)
            (PORT = 1521)
        )
        (CONNECT_DATA =
            (SID = MYSID)
        )
    )

before

mydb =
    (DESCRIPTION =
        (ADDRESS =
            (PROTOCOL = TCP)
            (HOST = host1)
            (PORT = 1521)
        )
        (CONNECT_DATA =
            (SERVICE_NAME = PLUG1)
        )
    )

after
```
When Connect String is not Present

What to do when client connect on the server directly?
sqlplus scott/tiger

Put in TNSNAMES.ORA file, SERVICE_NAME = PLUG1

Set TWO_TASK environment variable
$ export TWO_TASK=PLUG1
$ sqlplus scott/tiger

Which PDB am I in?

• From SQL*Plus
  SQL> show con_name
• From any session
  sys_context('USERENV','CON_NAME')
**DBA views**

**At root**

```sql
SQL> select tablespace_name from dba_tablespaces;
```

<table>
<thead>
<tr>
<th>TABLESPACE_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM</td>
</tr>
<tr>
<td>SYSAUX</td>
</tr>
<tr>
<td>UNDOTBS1</td>
</tr>
<tr>
<td>TEMP</td>
</tr>
<tr>
<td>USERS</td>
</tr>
<tr>
<td>UNDOTBS2</td>
</tr>
<tr>
<td>UNDOTBS3</td>
</tr>
<tr>
<td>UNDOTBS4</td>
</tr>
</tbody>
</table>

**At PLUG1**

```sql
SQL> select tablespace_name from dba_tablespaces;
```

<table>
<thead>
<tr>
<th>TABLESPACE_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM</td>
</tr>
<tr>
<td>SYSAUX</td>
</tr>
<tr>
<td>TEMP</td>
</tr>
<tr>
<td>URBANCODE</td>
</tr>
<tr>
<td>URBANCODE1</td>
</tr>
</tbody>
</table>

**NEWURBANCODE**

**All PDB Views**

- Prefixed with CDB_ instead of DBA_

```sql
select tablespace_name from DBA_TABLESPACES
```

<table>
<thead>
<tr>
<th>TABLESPACE_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEWURBANCODE</td>
</tr>
<tr>
<td>SYSAUX</td>
</tr>
<tr>
<td>SYSTEM</td>
</tr>
<tr>
<td>TEMP</td>
</tr>
<tr>
<td>UNDOTBS1</td>
</tr>
<tr>
<td>UNDOTBS2</td>
</tr>
<tr>
<td>UNDOTBS3</td>
</tr>
<tr>
<td>UNDOTBS4</td>
</tr>
<tr>
<td>USERS</td>
</tr>
</tbody>
</table>

```sql
select tablespace_name from CDB_TABLESPACES
```

<table>
<thead>
<tr>
<th>TABLESPACE_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEWURBANCODE</td>
</tr>
<tr>
<td>SYSAUX</td>
</tr>
<tr>
<td>SYSAUX</td>
</tr>
<tr>
<td>SYSTEM</td>
</tr>
<tr>
<td>SYSTEM</td>
</tr>
<tr>
<td>TEMP</td>
</tr>
<tr>
<td>TEMP</td>
</tr>
<tr>
<td>TEMP</td>
</tr>
<tr>
<td>UNDOTBS1</td>
</tr>
<tr>
<td>UNDOTBS2</td>
</tr>
<tr>
<td>UNDOTBS3</td>
</tr>
<tr>
<td>UNDOTBS4</td>
</tr>
<tr>
<td>URBANCODE</td>
</tr>
<tr>
<td>URBANCODE</td>
</tr>
<tr>
<td>URBANCODE1</td>
</tr>
<tr>
<td>URBANCODE1</td>
</tr>
<tr>
<td>USERS</td>
</tr>
</tbody>
</table>

CON_ID column shows the container the data belongs to.
More Space

SQL> show pdbs

<table>
<thead>
<tr>
<th>CON_ID</th>
<th>CON_NAME</th>
<th>OPEN MODE</th>
<th>RESTRICTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>PDB$SEED</td>
<td>READ ONLY</td>
<td>NO</td>
</tr>
<tr>
<td>3</td>
<td>PLUG1</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
</tbody>
</table>

- Two additional containers

V$ Views

- Some V$ Views show the same data regardless of where you are connected
  - Examples:
    - V$DATABASE
    - V$LOGFILE
- But most show values specific for the PDB ➔ IMPORTANT

At root

```sql
select value from v$sysstat s, v$statname n
where n.name = 'parse time cpu'
and n.statistic# = s.statistic#;
```

<table>
<thead>
<tr>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>86637</td>
</tr>
</tbody>
</table>

At plug1

```sql
select value from v$sysstat s, v$statname n
where n.name = 'parse time cpu'
and n.statistic# = s.statistic#;
```

<table>
<thead>
<tr>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
</tr>
</tbody>
</table>
Opening and closing PDBs

- Open in current instance only:
  SQL> alter pluggable database PLUG1 open;
- Open in current instance read only:
  SQL> alter pluggable database PLUG1 open read only;
- Open all PDBs in current instance only:
  SQL> alter pluggable database all open;
- Open in all instances:
  SQL> alter pluggable database PLUG1 open instances=all;
- Or, you can be in the PDB
  SQL> alter session set container = plug1;
  SQL> startup

Shutdown PDBs

- Shutting down PDBs do not shut down the CDB
- Close in current instance only:
  SQL> alter pluggable database PLUG1 close;
- Close in current instance immediately:
  SQL> alter pluggable database PLUG1 close immediate;
- Close all PDBs in current instance only:
  SQL> alter pluggable database all close;
- Close in all instances:
  SQL> alter pluggable database PLUG1 close instances=all;
- Or, you can be in the PDB
  SQL> alter session set container = plug1;
  SQL> shutdown [immediate]
**Individualized Instances of PDBs**

- PDBs can be opened on selected instances

```sql
SELECT name, inst_id, OPEN_MODE
FROM gv$pdbs;
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>INST_ID</th>
<th>OPEN_MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDB$SEED</td>
<td>1</td>
<td>READ ONLY</td>
</tr>
<tr>
<td>PDB$SEED</td>
<td>4</td>
<td>READ ONLY</td>
</tr>
<tr>
<td>PLUG1</td>
<td>1</td>
<td>READ WRITE</td>
</tr>
<tr>
<td>PLUG1</td>
<td>4</td>
<td>MOUNTED</td>
</tr>
<tr>
<td>PLUG2</td>
<td>1</td>
<td>READ WRITE</td>
</tr>
<tr>
<td>PLUG2</td>
<td>4</td>
<td>MOUNTED</td>
</tr>
<tr>
<td>SARPRD</td>
<td>1</td>
<td>READ WRITE</td>
</tr>
<tr>
<td>SARPRD</td>
<td>4</td>
<td>READ WRITE</td>
</tr>
</tbody>
</table>

Note: PLUG1 is opened on one instance and just mounted on the other.

**Service Name**

- Default service created: PDB name
- You can create additional services

```sql
SELECT name, con_name
FROM v$active_services;
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>CON_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>plug1</td>
<td>PLUG1</td>
</tr>
<tr>
<td>pdborcl</td>
<td>PDBORCL</td>
</tr>
<tr>
<td>cdborclXDB</td>
<td>CDB$ROOT</td>
</tr>
<tr>
<td>cdborcl</td>
<td>CDB$ROOT</td>
</tr>
<tr>
<td>SYS$BACKGROUND</td>
<td>CDB$ROOT</td>
</tr>
<tr>
<td>SYS$USERS</td>
<td>CDB$ROOT</td>
</tr>
</tbody>
</table>
Tip: Do NOT Use Default Service

- Default service
  - is not managed by srvctl
  - can't be brought down
  - starts as soon as the PDB comes up
  - If you move the PDB to a different CDB (with a different name), you can use the same service name; so apps do not need to change

- How to create a service for that specific PDB
  $ srvctl add service -d cdborcl -s newplug3 -pdb plug3 -preferred "cdborcl1" -available "cdborcl2"

Bug

- Default service pointed to root container

<table>
<thead>
<tr>
<th>NAME</th>
<th>CON_NAME</th>
<th>NAME</th>
<th>CON_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>plug1</td>
<td>PLUG1</td>
<td>plug1</td>
<td>CDB$ROOT</td>
</tr>
<tr>
<td>pdborcl</td>
<td>PDBORCL</td>
<td>pdborcl</td>
<td>PDBORCL</td>
</tr>
<tr>
<td>cdborclXDB</td>
<td>CDB$ROOT</td>
<td>cdborclXDB</td>
<td>CDB$ROOT</td>
</tr>
<tr>
<td>cdborcl</td>
<td>CDB$ROOT</td>
<td>cdborcl</td>
<td>CDB$ROOT</td>
</tr>
<tr>
<td>SYS$BACKGROUND</td>
<td>CDB$ROOT</td>
<td>SYS$BACKGROUND</td>
<td>CDB$ROOT</td>
</tr>
<tr>
<td>SYS$USERS</td>
<td>CDB$ROOT</td>
<td>SYS$USERS</td>
<td>CDB$ROOT</td>
</tr>
</tbody>
</table>

This can't be altered. So all apps pointing to the PLUG1 service will point at the root container, resulting in not finding the right data, users, etc.
Local/Common User

Database Parameter common_user_prefix = C##

Cloning

- Clone from an existing PDB:
  1. Bring up the source PDB in read only mode
  2. SQL> create pluggable database plug2 from plug1;
  3. Creates the default service plug2 automatically.
  4. Create a non-default service name for this PDB
- Can clone a remote PDB too
  SQL> create pluggable database plug3 from plug1@mylink;
- Can clone only metadata
  SQL> create pluggable database plug4 from plug1 no data;
Subsetting

- You can clone only a few user tablespaces
  SQL> create pluggable database plug3 from plug1 user_tablespaces = ('URBANCODE1');
- Very useful in creating subsets for prod-to-non-prod moves, or dividing too large PDBs into smaller ones.

Tip for Creation of PDBs
Transporting

1. If the PDB is open, you should close it.
   SQL> alter pluggable database plug1 close;
2. Create the meta-information on the PDB in an XML file.
   SQL> alter pluggable database plug1 unplug into 'plug1_meta.xml';
3. Copy this file and all the datafiles of plug1 to the target server.
4. On the target server, connect to the CDB with SYSDBA privilege
   $ sqlplus sys/oracle as sysdba
5. Execute this:
   SQL> create pluggable database newplug1 using 'plug1_meta.xml';

History of PDBs

- View CDB_PDB_HISTORY shows all operations
  select pdb_name, operation, op_timestamp, cloned_from_pdb_name
  from cdb_pdb_history;

<table>
<thead>
<tr>
<th>PDB_NAME</th>
<th>OPERATION</th>
<th>OP_TIMESTAMP</th>
<th>CLONED_FROM_PDB_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDB$SEED</td>
<td>UNPLUG</td>
<td>07-JUL-14</td>
<td></td>
</tr>
<tr>
<td>PDB$SEED</td>
<td>PLUG</td>
<td>11-OCT-14</td>
<td>PDB$SEED</td>
</tr>
<tr>
<td>PLUG1</td>
<td>CREATE</td>
<td>01-JUN-16</td>
<td>PDB$SEED</td>
</tr>
<tr>
<td>PDB$SEED</td>
<td>UNPLUG</td>
<td>07-JUL-14</td>
<td></td>
</tr>
<tr>
<td>PDB$SEED</td>
<td>PLUG</td>
<td>11-OCT-14</td>
<td>PDB$SEED</td>
</tr>
<tr>
<td>PLUG1</td>
<td>CREATE</td>
<td>01-JUN-16</td>
<td>PDB$SEED</td>
</tr>
<tr>
<td>PLUG2</td>
<td>CLONE</td>
<td>10-JUN-16</td>
<td>PLUG1</td>
</tr>
</tbody>
</table>
**DB Links**

- No Change. SQL> create database link plug1 using ‘plug1’
- Can connect to
  - Root container
  - PDB
- Can't do “alter session set container ...” so root-link is not practical
- But useful for V$ views

**Transportable Read Only Database**
Will they all play along nicely?

Resource Manager

```sql
dbms_resource_manager.create_cdb_plan_directive (
  plan => 'dayplan1',
  pluggable_database => 'plug1',
  shares => 2,
  utilization_limit => 100,
  parallel_server_limit => 100
);

dbms_resource_manager.create_cdb_plan_directive (
  plan => 'dayplan1',
  pluggable_database => 'plug2',
  shares => 1,
  utilization_limit => 50,
  parallel_server_limit => 50
);
```
Data Pump

- Remember, directories are visible only in a PDB
  - So, in expdp or impdp, use user/pw@plug1
- SERVICE_NAME parameter doesn't help
  
  $ expdp u/p directory=tmp_dir cluster=no service_name=plug1
  
  schemas=UCRELEASE
- OR, use TWO_TASK
  
  $ export TWO_TASK=plug11
  
  $ expdp u/p ...

Backup/Recovery

- You can backup the entire CDB
  
  - RMAN> connect target /
  
  - RMAN> backup database;
- OR, individual PDBs
  
  - RMAN> backup pluggable database plug1;
- OR, use RMAN directly
  
  - RMAN> connect target=sys/oracle@plug1
  
  - RMAN> backup database;
- You can restore entire CDB or individual PDBs
Point in Time Recovery of PDB

- You can do PITs of individual PDBs leaving the rest in their place.
  ```sql
  RMAN> run {
    2>      set until time '08-MAR-16';
    3>      restore pluggable database plug1;
    4>      recover pluggable database plug1;
    5>  }
  ```
- Creates an auxiliary instance, creates that PDB only and plugs it in.

Upgrade

- Update the entire CDB. All individual PDBs will be upgraded
- Transport a specific PDB from one CDB to another at a higher version
Golden Gate

Golden Gate Setup

- In CDB
  - Create a common user
    
    SQL> create user C##GGADMIN identified by ggadmin;
    
    SQL> exec dbms_goldengate_auth.grant_admin_privilege ('C##GGADMIN', container => 'ALL')
    
    SQL> grant dba to C##GGADMIN container = ALL;
  
- Repeat for all PDBs.
  
    SQL> alter session set container = plug1;
    
    SQL> alter pluggable database add supplemental log data;
Golden Gate Configuration (ggsci)

GGSCI (source) 1> dblogin userid acme@plug1, password acme
GGSCI (source as acme@CDB1/PLUG1) 2> add schematrandata plug1.acme
GGSCI (source) 3> edit params extora
   EXTRACT EXTORA
   USERID C##GGADMIN@CDB1, password ggadmin
   RMTHOST remotehost1, MGRPORT 7809
   RMTTRAIL ./trails
   DDL INCLUDE MAPPED
   LOGALLSUPCOLS
   UPDATERECORDFORMAT COMPACT
   TABLE PLUG1.ACME.*;
   TABLE PLUG2.ACME.*

Naming Convention Tip

- CDB Names
  - C<Seq#><AppName>
  - C2SAR
- PDB Name
  - Make it unique across the enterprise
  - P1SAR
  - Makes it easy to plug in to any CDB without renaming
  - Allows you to open multiple PDBs read only
Caveats

- Non-CDBs may be deprecated
- Some features not supported on CDBs (as of 12.1)
  - Heat Map, Automatic Data Optimization
  - Change Notification
  - Client Side Result Cache
- Real Application Testing only for CDB; not PDB

Summary

- CDB/PDBs are transparent to the applications
- Most functions for the DBA stay the same without change
  - The scripts will work
  - Backup/recovery works
- Pay close attention to dynamic performance views as their meanings could change
- DBA_views shows PDB specific data. CDB_views show all PDBs
- Create a non-default service name for the PDB; do not use the default one.
Thank You!

Blog: arup.blogspot.com
Tweeter: @ArupNanda
Facebook.com/ArupKNanda