Oracle 12c Container Databases for Beginners to Advanced

20th Nov , 2013

Anuj Mohan

Database enthusiast and Public speaker

Sr. Managing DBA
dbaDIRECT .
Oracle certified Exadata Implementer
Oracle Certified RAC Expert
Oracle 11g Certified Professional (OCP)

About dbaDIRECT

dbaDIRECT provides a complete range of Oracle DBA services, no matter what your organization demands. Whether you're looking for money-saving services or more comprehensive third-party support, They include:

- Certified Oracle database consultants
- 24x7 Remote Oracle DBA services
- Oracle database upgrade services- including Oracle 9i, 9.2, 10g, 10.2, and 11g
- Oracle database tuning, migration services
- ONdemand - Oracle experts on call
- Technologies – RAC, RMAN, CRS, OEM, GRID, STATSPACK

ORACLE Gold Partner Specialized Oracle Database

MySQL

IBM

SYBASE

Microsoft SQL Server 2012

ORACLE e-Business Suite

AGENDA

- Oracle's Previous Releases 
- CDB/PDB Introduction and Concepts
- Shared / Exclusive components
- Accessing CDB's/PDB's
- Backup and Recovery
- Quick overview: migration options to 12c from previous versions.
- PDB - Explanation of Value (EOV)

*** In this presentation I am representing my understanding of 12c. I am not representing any organization or entity.

Oracle's Evolution

- 1982: RSI became Oracle Corporation
- 1983: version 3, supported COMMIT and ROLLBACK functionality for transactions. extended platform support to include Unix environments
- 1984: version 4, which supported read-consistency
- 1985: version 5, which supported the client–server
- 1986: Oracle version 5.1 started supporting distributed queries
- 1988: version 6 supported PL/SQL embedded within Oracle Forms v3 (version 6 could not store PL/SQL in the database proper), row-level locking and hot backups

1992: Oracle version 7

- Referential integrity
- Stored procedures
- Triggers.

1997: Oracle version 8

- Object-oriented
- Multimedia applications

1999: Oracle8*i*

- Internet
- Java virtual machine

2001: Oracle9*i*

- 400+ New features
- Oracle RAC replace OPS

2003: Oracle Database 10g

- Automatic Storage Management
- Oracle Data Pump
- Virtual Private Database
- Automatic Shared Memory Management
- ADDM and SQL Tuning Advisor
- Automatic Workload Repository
- Automatic Segment Management
- Flashback Table

2007: Oracle Database 11g

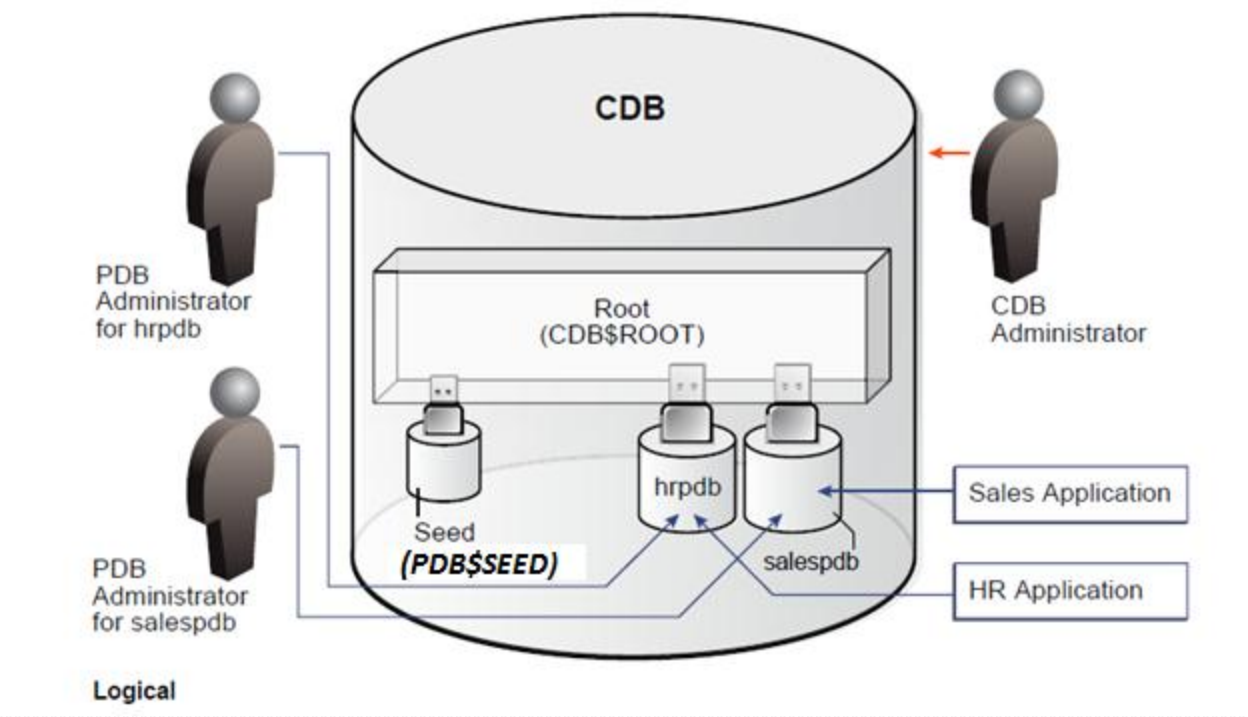
- Database Replay
- SQL Performance Analyzer
- Active Data Guard
- Snapshot Standby
- Flashback Data Archives
- Edition-Based Redefinition
- RAC One Node, and Clusterware
- Grid Ifrastructure (R2)
- Data Recovery Advisor
- Few years later EXADATA

2013: Oracle 12c

- Container / Pluggable databases
- Online Datafile and Partition Movement
- Policy-Based Automatic Redaction
- Flex ASM
- SQL Plan Management enhancements
- Information Lifecycle Management (ILM)
- Automatic Data Optimization (ADO)

AGENDA

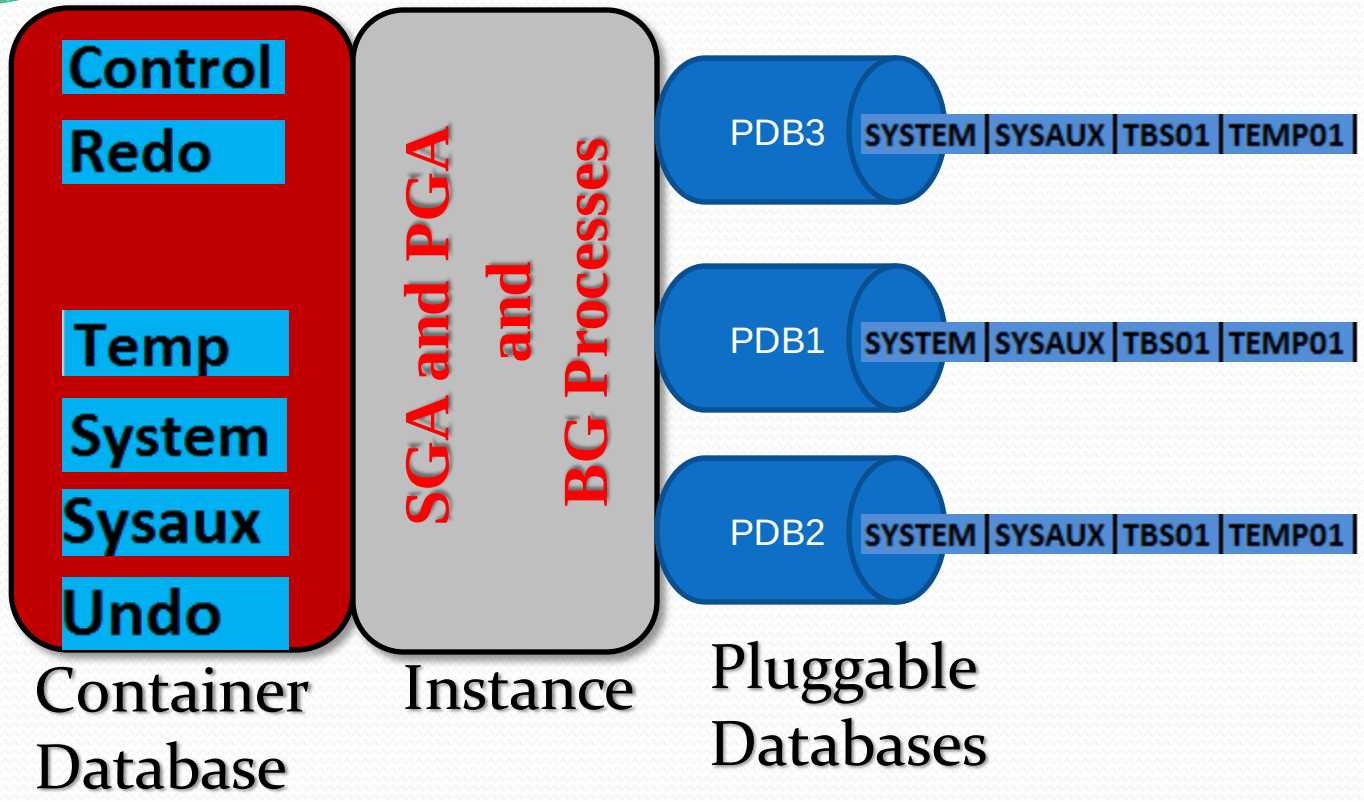
- Oracle's Previous Releases
- CDB/PDB Introduction and Concepts 
- Shared / Exclusive components
- Accessing CDB's/PDB's
- Backup and Recovery
- Quick overview: migration options to 12c from previous versions.
- PDB - Explanation of Value (EOV)


CDB
PDB
Administrator
for hrpdb
Root
(CDB$ROOT)
CDB
Administrator
hrpdb
Sales Application
Seed
(PDB$SEED)
salespdb
PDB
Administrator
for salespdb
HR Application
Logical

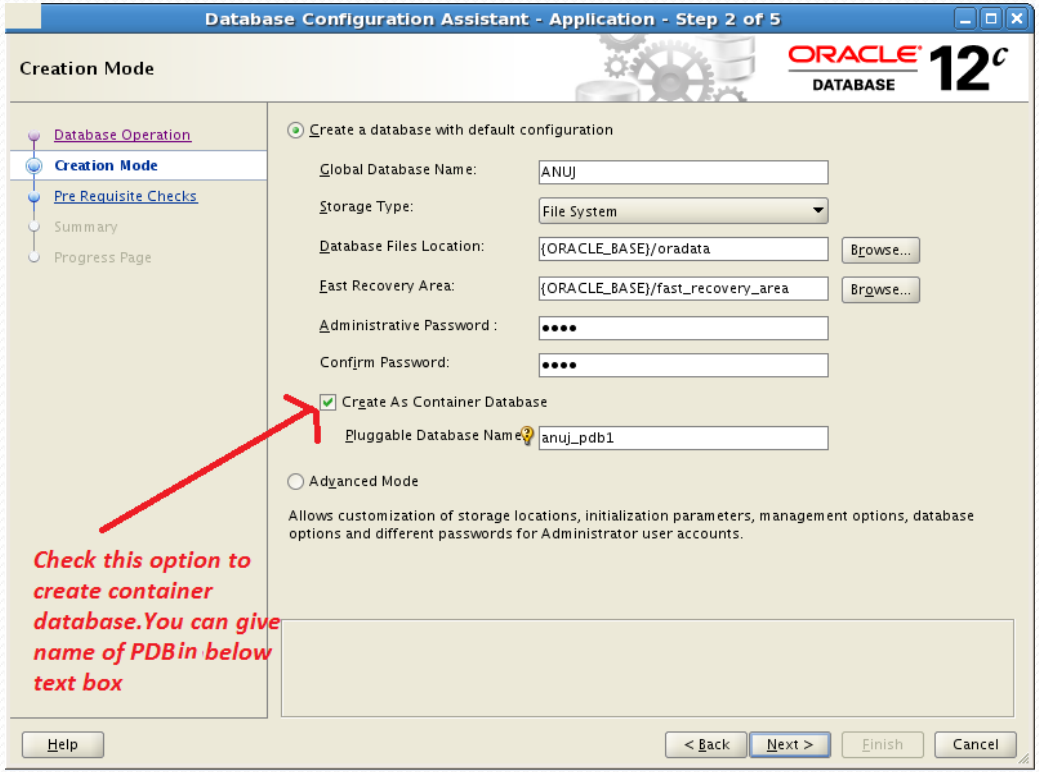
Understanding in simple words

- CDB is a normal database with additional capability of storing one or more non CDB
- These non CDB's within CDB are called PDB
- PDB's can be plugged and unplugged easily from one container to another container. In above example we have two PDBs hrpdb and salespdb

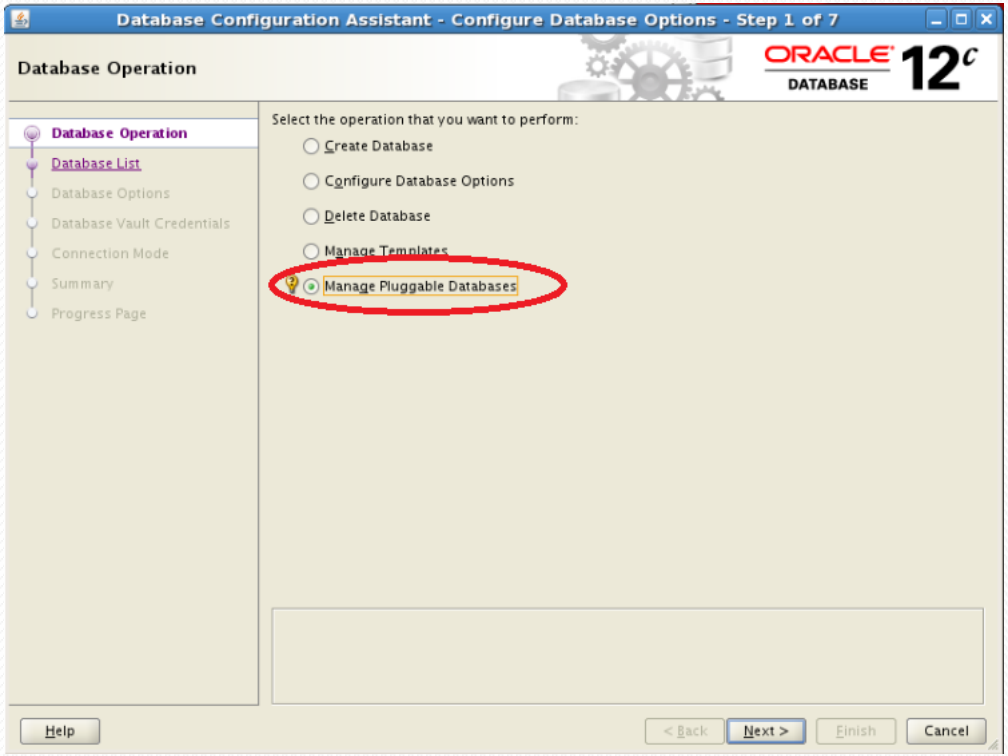
- ***CDB Container database
- ***PDB pluggable database


Control
Redo
Temp
System
Sysaux
Undo
SGA and PGA
and
BG Processes
PDB3
SYSTEM
SYSAUX
TBS01
TEMP01
PDB1
SYSTEM
SYSAUX
TBS01
TEMP01
PDB2
SYSTEM
SYSAUX
TBS01
TEMP01
Container Database
Instance
Pluggable Databases

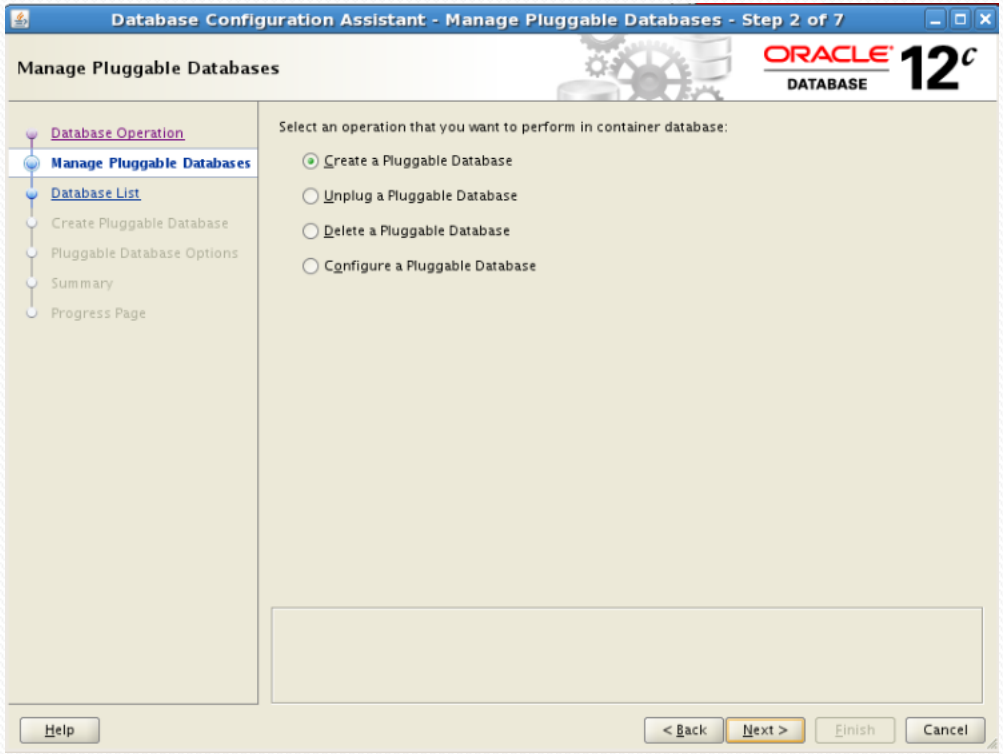
Option for creating Container Database



PDB using DBCA



PDB using DBCA



Ways to get a PDB inside CDB

- From an existing remote PDB
- From PDB$SEED
- From an existing local PDB
- Unplug from a CDB and plug into another CDB

Ways to get a PDB inside CDB

- **Cloning a remote PDB (Clone anuj@CDB1 as mohan@CDB2)**
- Connect to CDB1 and place Anuj in read only mode
- Connect to CDB2 as sys and run clone command

(if using a common user you need to grant "create pluggable database" container=all)

- create pluggable database mohan from anuj@CDB2_dbl** admin user sys identified by sys_pwd file_name_convert = ('/u01/pdbs/anuj','/u01/pdbs/mohan');
- alter pluggable database Mohan open;

* Endian format,character set, etc must be compatible between CDBs

** CDB2_dbl is database link used for communication between CDB1 and CDB2

Ways to get a PDB inside CDB

- **Creating an PDB (Anuj) from PDB$SEED**

- Set db_file_name_convert
  - SQL> alter system set PDB_FILE_NAME_CONVERT ='/u01/datafiles/pdbseed/','/u01/datafiles/Anuj/' scope=both;

    System altered.
- Run create pdb command
  - SQL> CREATE PLUGGABLE DATABASE Anuj ADMIN USER PDB_Anj IDENTIFIED BY PDB_Anj default tablespace users datafile '/u01/datafiles/Anuj/users_01.dbf' size 1000M;ROLES=(DBA);

    Pluggable database created.
- Open PDB in read write mode
  - SQL> alter pluggable database Anuj open;

Ways to get a PDB inside CDB

- **Cloning a PDB (Clone Anuj as Mohan)**
- Connect to CDB root as sys
- Open Anuj in read-only mode and run create pdb
  - alter pluggable database Anuj close;
  - alter pluggable database Anuj open read only;
  - create pluggable database Mohan from Anuj admin user sys identified by sys_pwd file_name_convert = (‘/u01/pdbs/anuj’,’/u01/pdbs/mohan’);
- Open the database
  - alter pluggable database mohan open;

Ways to get a PDB inside CDB

- **Unplug and plug**
- Connect to source(CDB1) as sys
- Shut down Anuj
  - alter pluggable database Anuj close;
  - alter pluggable database Anuj unplug into '/u01/datafiles/Anuj/Anuj.xml';
- If you do not want to keep this PDB anymore on Current CDB you can drop it but make a copy of data files before drop.
- drop pluggable database Anuj

Ways to get a PDB inside CDB

- **Unplug and plug continue…**
- Connect to Destination(CDB2) as sys
  - select dbms_pdb.check_plug_compatibility(pdb_descr_file=>'/u01/datafiles/Anuj/Anuj.xml', store_report=>true) from dual;
- Check for errors in pdb_plug_in_violations table
- move PDB files and use nocopy
  - create pluggable database Anuj using '/u01/datafiles/Anuj/Anuj.xml' nocopy;

** Similar to transportable databases

** XML file has PDB metadata

AGENDA

- Oracle's Previous Releases
- CDB/PDB Introduction and Concepts
- Shared / Exclusive components 
- Accessing CDB's/PDB's
- Backup and Recovery
- Quick overview: migration options to 12c from previous versions.
- PDB - Explanation of Value (EOV)

Common Component in Container Databases (CDB's)

- Background processes –
- Memory areas – buffer cache, log buffer, etc.
- Datafiles (Undo / Redo /system /
- Undo tablespace
- Single ADR location

- PDB's may have their own datafiles

Common vs. Local Concept

- Common vs Local Concept
  - Can be defined on CDB or PDB Level
    - Users →Defined on root or local PDB
    - Roles →Defined on root or local PDB
    - Privileges →Defined on root or local PDB
    - Objects →Defined on root or local PDB

Common and Local Privileges

- A common privilege is privilege granted across all containers
- A privilege granted to a single PDB is a local privilege
- Local users can only utilize privileges locally in the current PDB
- Common users can only utilize privileges locally in the current PDB
- Common users connected to the root container can utilize privileges across container, such as creating a common user

Common and Local Roles

- local user can only create local roles. Local roles can be granted to local or common users. Local roles can be granted to common roles.
- common user can create common roles and/or local roles. Common roles can be granted to local or common users. Common roles can be granted to local roles
  - SQL> create role bill_master CONTAINER=ALL;
  - SQL> create role bill_local CONTAINER=CURRENT;

Grant and Revoke Privileges and Roles

- Grant common privilege by common user
- SQL> grant prv1 to cm_user CONTAINER=ALL;

- Grant local privilege by common user
- SQL> grant prv2 to pdb01_user CONTAINER=CURRENT;

- Grant a local privilege by a local user
- SQL> grant prv2 to pdb01_user;
- Revoke common privilege by common user
- SQL> revoke prv1 from cm_user CONTAINER=ALL;

Create Common User

- Connect to CDB and Create Common User

```
SQL> show con_id
    CON_ID
    ----------
         1
SQL> show con_name
CON_NAME
----------------
CDB$ROOT
```

- SQL> create user C##_main identified by qweasdzxc
  2 container=ALL;

  User created.

Grant some role and privilege to common user

- SQL> grant dba to C##_main container=ALL;
- Grant succeeded.
- SQL> grant select any table to C##_main container=ALL;
- Grant succeeded.
- select grantee, granted_role from dba_role_privs where grantee ='C##_MAIN';

GRANTEE	GRANTED_ROLE
C##_main	DBA
C##_main	DBA

- select grantee, privilege from dba_sys_privs where grantee ='C##_MAIN';

GRANTEE	PRIVILEGE
C##_main	UNLIMITED TABLESPACE
C##_main	SELECT ANY TABLE
C##_main	UNLIMITED TABLESPACE
C##_main	SELECT ANY TABLE

Common User Information

- Connect to PDB using just created common user
  - SQL> show con_name
  - CON_NAME

    ANUJPDB
    SQL> show con_id
  - CON_ID

    3

  - SQL> select grantee, granted_role from dba_role_privs where grantee ='C##_MAIN';
  - GRANTEE GRANTED_ROLE
    --- ---
    C##_main DBA
  - SQL> select grantee, privilege from dba_sys_privs where grantee ='C##_MAIN';
  - GRANTEE PRIVILEGE
    --- ---
    C##_main UNLIMITED TABLESPACE
    C##_main SELECT ANY TABLE

While connected to PDB try to create common user

- create user C##_main1 identified by qweasdzxc container=ALL;
  create user C##_main1 identified by qweasdzxc container=ALL
  *
  ERROR at line 1:
  ORA-65050: Common DDLs only allowed in CDB$ROOT
- create user C##_main1 identified by qweasdzxc
  *
  ERROR at line 1:
  ORA-65094: invalid local user or role name
- SQL> create user main1 identified by qweasdzxc; /* User local to PDB */
  User created.
- SQL> grant dba to main1;
  Grant succeeded.
- SQL> grant create any table to main1;
  Grant succeeded.

Check roles/privileges you just granted

- SQL> select grantee, privilege from dba_sys_privs where grantee ='MAIN1';
- GRANTEE PRIVILEGE
  ---------- ----------
  main1 CREATE ANY TABLE
  main1 UNLIMITED TABLESPACE
- SQL> select grantee, granted_role from dba_role_privs where grantee ='MAIN1';
- GRANTEE GRANTED_ROLE
  ---------- ----------
  main1 DBA

Connect to CDB and check privileges of local user we just granted

- SQL> show con_name
  - CON_NAME

    CDB$ROOT

- SQL> select grantee, privilege from dba_sys_privs where grantee =‘MAIN1';
  - no rows selected
- SQL> select grantee, granted_role from dba_role_privs where grantee =MAIN1';
  - no rows selected

- SQL> select grantee, granted_role from cdb_role_privs where grantee =‘MAIN1';

```
GRANTEE              GRANTED_ROLE
--------------------------------------------------------------------------------
main1       DBA
```

- SQL> select grantee, privilege from cdb_sys_privs where grantee =‘MAIN1';

```
GRANTEE              PRIVILEGE
----------------------------------------
main1       CREATE ANY TABLE
main1       UNLIMITED TABLESPACE
```

- USER_%% Show list of all the Objects owned by the current user in a PDB
- ALL_%% Show list of all the Objects accessible by the current user in a PDB
- DBA_%% Show list of all the Objects in the root or a pluggable database
- CDB_%% Show list of all the Objects in the container database. CDB_%% use new column CON_ID

Creating Objects

- Local object can be created by common and local users
- Common objects can NOT be created by user defined common users.To create Common objects you need to use by Oracle supplied common user (sys/system)

DBA_USERS

- Name Null? Type

USERNAME	NOT NULL	VARCHAR2(128)
USER_ID	NOT NULL	NUMBER
PASSWORD		VARCHAR2(4000)
ACCOUNT_STATUS	NOT NULL	VARCHAR2(32)
LOCK_DATE		DATE
EXPIRY_DATE		DATE
DEFAULT_TABLESPACE	NOT NULL	VARCHAR2(30)
TEMPORARY_TABLESPACE	NOT NULL	VARCHAR2(30)
CREATED	NOT NULL	DATE
PROFILE	NOT NULL	VARCHAR2(128)
INITIAL_RSRC_CONSUMER_GROUP		VARCHAR2(128)
EXTERNAL_NAME		VARCHAR2(4000)
PASSWORD_VERSIONS		VARCHAR2(12)
EDITIONS_ENABLED		VARCHAR2(1)
AUTHENTICATION_TYPE		VARCHAR2(8)
PROXY_ONLY_CONNECT		VARCHAR2(1)
COMMON		VARCHAR2(3)
LAST_LOGIN		TIMESTAMP(9) WITH TIME ZONE

CDB_USERS

- Name Null? Type

```
----------------------------- -------- ----------------------------
USERNAME                      NOT NULL VARCHAR2(128)
USER_ID                       NOT NULL NUMBER
PASSWORD                               VARCHAR2(4000)
ACCOUNT_STATUS                NOT NULL VARCHAR2(32)
LOCK_DATE                              DATE
EXPIRY_DATE                            DATE
DEFAULT_TABLESPACE            NOT NULL VARCHAR2(30)
TEMPORARY_TABLESPACE          NOT NULL VARCHAR2(30)
CREATED                       NOT NULL DATE
PROFILE                       NOT NULL VARCHAR2(128)
INITIAL_RSRC_CONSUMER_GROUP            VARCHAR2(128)
EXTERNAL_NAME                          VARCHAR2(4000)
PASSWORD_VERSIONS                                VARCHAR2(12)
EDITIONS_ENABLED                       VARCHAR2(1)
AUTHENTICATION_TYPE                    VARCHAR2(8)
PROXY_ONLY_CONNECT                     VARCHAR2(1)
COMMON                                 VARCHAR2(3)
LAST_LOGIN                             TIMESTAMP(9) WITH TIME ZONE
CON_ID                                 NUMBER
```

AGENDA

- Oracle's Previous Releases
- CDB/PDB Introduction and Concepts
- Shared / Exclusive components
- Accessing CDB's/PDB's 
- Backup and Recovery
- Quick overview: migration options to 12c from previous versions.
- PDB - Explanation of Value (EOV)

Tools to manage CDBs and PDBs

- Your favorite SQL *Plus
- 12c EM gives dropdown option to select PDB
- Sql developer
- Any third party software you prefer
  - Toad
  - Dbartisan

Connecting CDB and PDB

- [oracle@test121 ~]$ export ORACLE_SID=AnujC
  [oracle@test121 ~]$ sqlplus / as sysdba

```
SQL> select name,open_mode  from v$database;

NAME      OPEN_MODE
--------- --------------------
ANUJC     READ WRITE

SQL> show con_id

CON_ID
------------------------------
1

SQL> show user
USER is "SYS"
```

-

Connecting CDB and PDB continued…

- select pdb_id,pdb_name from cdb_pdbs;

```
   PDB_ID PDB_NAME
---------- ------------------------------------------
         2 PDB$SEED
         3 ANUJPDB
         4 ANUJP02
```

- There are ways you can access PDBs
- Using tns

```
sqlplus C##_MAIN/qweasdzxc@ANUJPDB  /* Tnsnames.ora */
SQL> select NAME, OPEN_MODE, CON_ID from V$PDBS;
NAME                           OPEN_MODE      CON_ID
------------------------------ ---------- ----------
ANUJP DB                       READ WRITE          3
```

Connecting CDB and PDB continued…

- Connect using service name
  - Sqlplus C##_MAIN/qweasdzxc@anuj-w530:1521/ANUJPDB /* Service Name */
- Connect using Local/TWO_TASK
  - set LOCAL=ANUJPDB /* on windows */
  - set TWO_TASK=ANUJPDB /* on unix */
  - sqlplus C##_MAIN/qweasdzxc
- How to check which PDB you are connected
  - Show CON_ID
  - Show CON_NAME
  - Using SYS_CONTEXT

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

CDB Startup stages

- CDB Startup stages
  - Shutdown
    - Instance and database both down
  - Nomount
    - Instance is started. V$ views are accessible
  - Mount
    - CDB control file is opened
    - Root is mounted
    - PDBs are also in mount stage
  - Open
    - CDB is open for read /write
    - PDBs will be still in mount stage by default

PDBs Startup

- PDBs Startup
  - When you open your CDB , that will bring your PDBs to mount stage. At this stage you can use below options to open your PDBs
    - Connect to CDB
    - Alter pluggable database pdb_name open;
    - Alter pluggable database all open ; /* open all PDBs */
    - Select name,open_mode from v$pdbs;

Shutting CDBs and PDBs

- Shutting CDBs and PDBs
  - To shutdown CDBs you use “shutdown *****” commands
    - Shutdown immediate
    - Shutdown transactional
    - Shutdown abort
  - Their possible states are MOUNT/OPEN/CLOSE

Closing PDBs

- Closing PDBs
  - Connect to CDB
    - Alter pluggable database all except pdb_name close/open;
    - Alter pluggable database pdb_name close immediate;
  - Connect to PDB
    - Username/password@PDB as sysdba
    - Shutdown immediate
    - OR
    - Alter pluggable database close
  - Connect to CDB and check
    - Select name,open_mode from v$pdbs;

Open restrict

- You can Open your PDBs in restricted /read only and read write based on your requirement

  - Alter pluggable database open restricted
  - Alter pluggable database open read only
  - Alter pluggable database open

Init Parameters

- Parameter file is defined at CDB level i.e one parameter file per CDB
- There are some parameter which can be defined for PDB but it will be defined in init.ora of CDB hosting that pdb
- v$parameter (ISPDB_MODIFIABLE='TRUE')
  - select ispdb_modifiable,count(*) from v$parameter group by ispdb_modifiable
    ;

```
ISPDB   COUNT(*)
----- ----------
TRUE         151
FALSE        213
SQL> select name from v$parameter where ispdb_modifiable='TRUE' and rownum<4;
NAME
--------------------------------------------------------------------------------
timed_statistics
timed_os_statistics
resource_limit
```

- Change in PDB parameter value becomes effective after CDB restart or close and open of PDB

AGENDA

- Oracle's Previous Releases
- CDB/PDB Introduction and Concepts
- Shared / Exclusive components
- Accessing CDB's/PDB's
- Backup and Recovery 
- Quick overview: migration options to 12c from previous versions.
- PDB - Explanation of Value (EOV)

Backup and Recovery

Backups

- RMAN Backup Levels
  - CDB Level backup: All CDB datafiles, PDBs datafiles and controlfile and spfile if control file autobackup is defined
  - PDB backup: All PDBs datafiles and controlfile and spfile if control file autobackup is defined
  - Tablespace level backup, so you can backup individual tablespace of any pdb
  - Datafile level backups

Backup…

- You can connect to CDB/PDB to perform backup

- sysdba or sysbackup privilege required to run backup
  - RMAN> BACKUP DATABASE;
  - BACKUP PLUGGABLE DATABASE pdb1prd, pdb2prd;
  - BACKUP TABLESPACE pdb1prd:tbs2;
  - BACKUP PLUGGABLE DATABASE "CDB$ROOT";

Recovery options

- Media recovery : File loss or damage
  - CDB level –Entire CDB
  - PDB level – Entire PDB
  - Tablespace level: same as for non-CDB.Any tablespace of CDB or PDB
  - Datafile level recovery option
  - Data block level recovery
  - TSPITR for root tablespaces ONLY except SYSTEM, UNDO, SYSAUX

  - Flashback database
    - CDB level
    - PDB level
  - Block recovery: Same as Oracle 11g

***** Instance recovery is not possible on PDB level**

Recovery Commands

- Loss of system datafile from PDB
  - In order to recover from loss of PDB's system datafile we need to take CDB on mount stage(No CDB/PDB available to users)
  - RMAN> startup mount; (cdb mounted)
  - RMAN> restore tablespace pdb1prd :system;
  - RMAN> recover tablespace pdb1prd :system;
  - RMAN> alter database open; (cdb open)
  - RMAN> alter pluggable database all open;

Recovery Commands

- Loss of non system datafile of PDB or CDB
  - We do not need to take PDB/CDB offline (CDB/PDB available to users except offline tablespace)
  - SQL> connect system/12cmanager@ pdb1prd
  - SQL> alter tablespace tbs01 offline immediate;
  - RMAN> restore tablespace pdb1prd :tbs01;
  - RMAN> recover tablespace pdb1prd : tbs01;
  - SQL> alter tablespace tbs01 online

Enable Flashback

- Database must be in archivelog mode
- Set Init Parameters
  - db_recovery_file_dest_size=xxgb
  - db_recovery_file_dest=recovery file location
- Enable on sqlplus
  - sql> alter system set db_flashback_retention_target=1440 scope=both;
  - sql> alter database flashback on;

Flashback database

- Create Restore points
  - sql> create restore point before_upgrade;
  - sql>create restore point before_upgrade guarantee flashback database;

- Find out SCN you want to flashback or restore points

  - sql> select name, scn, time, database_incarnation#, guarantee_flashback_database,storage_size from v$restore_point;

  OR

  - rman> list restore point all;

Flashback database

- sql>startup mount
- flashback database to scn 1809;

OR

- Rman> flashback database to restore point 'before_upgrade';
- rman> alter database open read only;
- Validate recovered information

- rman> shutdown immediate
- rman> startup mount
- flashback database to scn 1809;

OR

- flashback database to restore point 'before_upgrade';

- rman> alter database open resetlogs;
- rman> alter pluggable database all open;

AGENDA

- Oracle's Previous Releases
- CDB/PDB Introduction and Concepts
- Shared / Exclusive components
- Accessing CDB's/PDB's
- Backup and Recovery
- Quick overview: migration options to 12c from previous versions. 
- PDB - Explanation of Value (EOV)

Migration to 12c : Method 1

- Create a Container database
- Create a pluggable database inside the container
- Take export of source db (10g/11g expdp)
- Import in pluggable or container database (12c impdp)

Migration to 12c : Method 2

- Use of some sort of replication tool
- Source : 10g/11g
- Destination : 12c

- E.g. GoldenGate , transient logical standby

Migration to 12c : Method 3

- Upgrade a pre-12c database to 12c (non cdb)
- Plug-in the non-CDB into a CDB
- Helpful features :
  - parallel processing to regular upgrades
  - Manual / dbua generates automatic fixups before upgrade
  - dbua can use existing rman backups for fall back strategy
  - you can create restore points and do a flashback database
  - datapump full transportable (combination of datapump and transportable tablespace)

Direct upgrade path

Source DB Version	Target DB Version
10.2.0.5 (or higher)	12.1.x
11.1.0.7 (or higher)	12.1.x
11.2.0.2 (or higher)	12.1.x

Indirect upgrade path

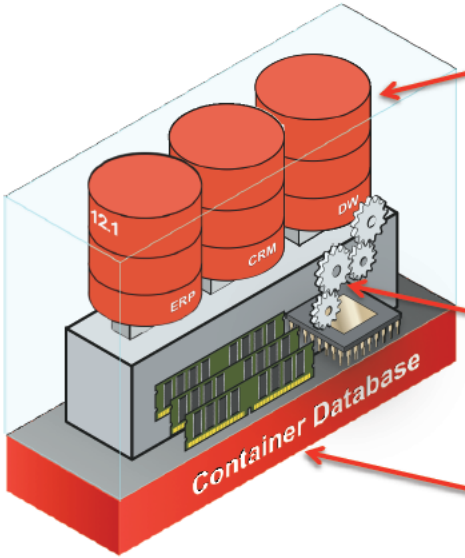
Lower Versions → Intermediate version	→	Final Version
7.3.4.x (or lower) → 9.2.0.8 →	11.2.x →	12.1.x
8.0.6.x (or lower) → 9.2.0.8 →	11.2.x →	12.1.x
8.1.7.4 (or lower) → 9.2.0.8 →	11.2.x →	12.1.x
9.0.1.4 (or lower) → 9.2.0.8 →	11.2.x →	12.1.x

AGENDA

- Oracle's Previous Releases
- CDB/PDB Introduction and Concepts
- Shared / Exclusive components
- Accessing CDB's/PDB's
- Backup and Recovery
- Quick overview: migration options to 12c from previous versions.
- PDB - Explanation of Value (EOV)

Oracle Multitenant

New architecture for consolidating databases and simplifying operations



Self-contained PDB for each application
- Applications run unchanged
- Rapid provisioning (via clones)
- Portability (via pluggability)

Shared memory and background processes
- More applications per server

Common operations performed at CDB level
- Manage many as one (upgrade, HA, backup)
- Granular control when appropriate

ORACLE

Source: Oracle Corporation

Key Benefits

Benefit	Capability Enabled
Minimize CapEx	• More applications per server
Minimize OpEx	• Manage many as one • Standardized procedures & service levels • Cloning for development / testing
Maximize Agility	• Rapid provisioning • Portability through “pluggability” • Scalability with RAC
Ease of Adoption	• Applications run unchanged

ORACLE

Source: Oracle Corporation

252 open PDBs with 8GB SGA in container

- SQL> show parameter sga
- NAME TYPE VALUE
  ---
  lock_sga boolean FALSE
  pre_page_sga boolean TRUE
  sga_max_size big integer 8000M
  sga_target big integer 8000M
  SQL> select open_mode,count(*) from v$pdbs group by open_mode;

- OPEN_MODE COUNT(*)

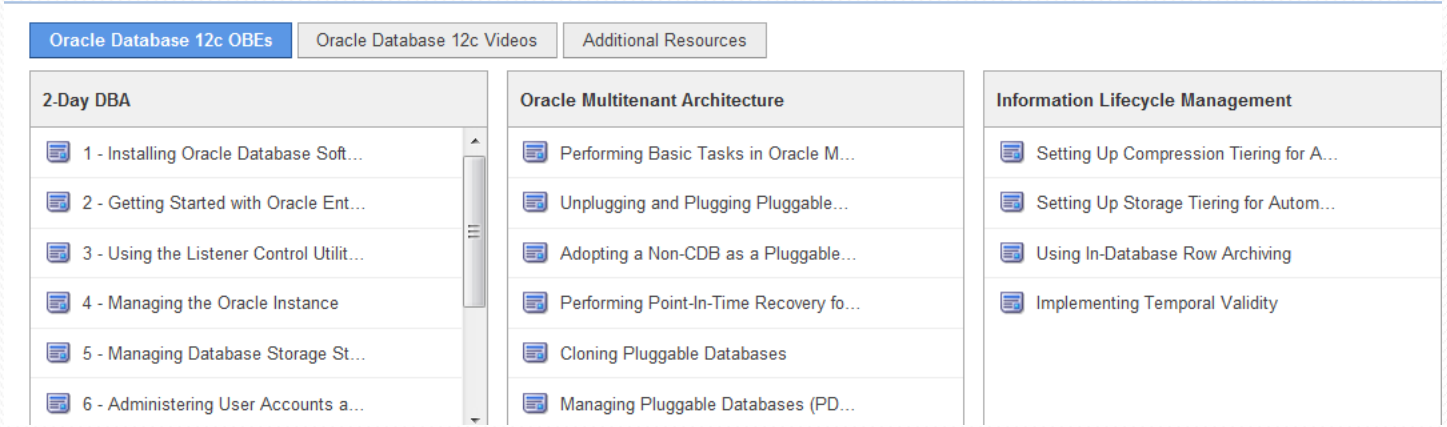
  READ ONLY 1
  READ WRITE 252

Summary and Conclusions

- There is no application change required for using PDBs
- Cloning is very easy . Perfect for creating test/development system quickly
- If file system supports copy-on-change then clones can be done in sub seconds
- PDBs need to be opened explicitly
- One physical database with multiple PDBs - Less DBA Effort to manage
- One set of memory / processes - Less usage means you can deploy more applications per physical server
- Adding more PDB only requires space
- Upgrade to CDB and plug in PDBs - PDBs are upgraded
- Cloning to remote servers
- Enhanced Resource manager
- No change required in Scripts and applications
- More than One PDB inside a CDB is Extra cost

Explore Oracle Database 12c Learning Resources

- Oracle Database 12c Learning Library:
  http://www.oracle.com/oll/database



Thank you …

- If you have further questions or need database assistance, Send me email with subject “12cSIG”
- Anuj.Mohan@dbadirect.com