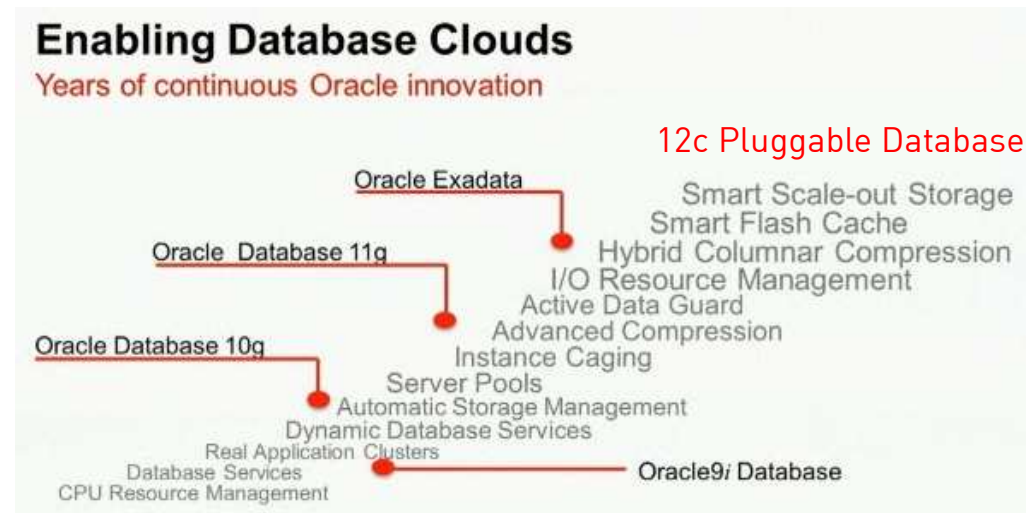


ORACLE 12C PLUGGABLE DATABASE FEATURE INSIGHTS

Kirill Loifman, 21/02/2013
Oracle Certified Professional DBA
email: loifmkir@gmail.com
www: dadbm.com
Twitter: [@loifmkir](https://twitter.com/loifmkir)

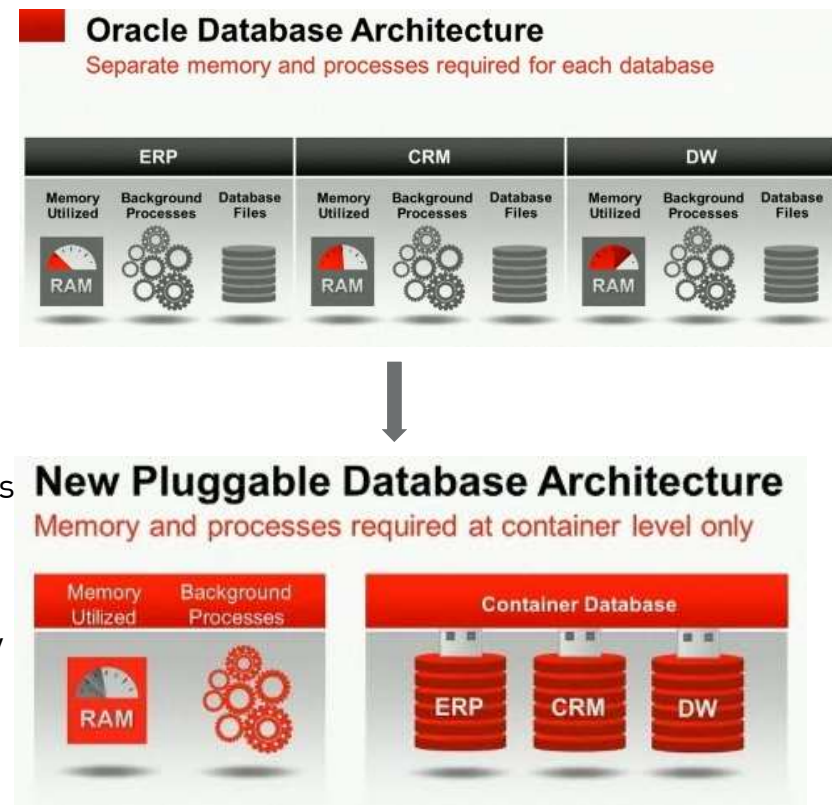
ORACLE 12C PLUGGABLE DATABASE (PDB)

- Native support for Cloud Computing
=> great help in database consolidation
- Fundamental architectural change
=> true for Oracle DB
- Feature that I've been waiting a decade



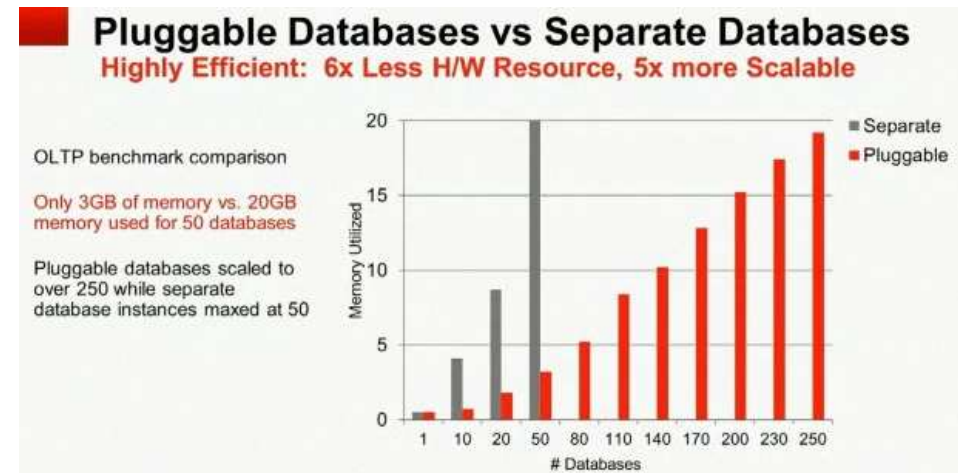
12C PDB ARCHITECTURE & FEATURES

- Many Pluggable Databases (PDBs) inside a single container database (CDB)
- PDB is backwards compatible with an ordinary pre-12.1 database
- PDB namespaces are entirely independent of one another
- A session sees only the single PDB it connects to
- Unplug a PDB from one CDB and plug it into another CDB
- Clone a PDB, both within the same CDB or from one CDB to another one
- The operations on PDBs are implemented as SQL statements
- CDB's administrator executes these operations when connected to a CDB
- All PDBs can be backed up at once, but recovered separately
- PDB is transparent to applications



12C PLUGGABLE DATABASE BENEFITS

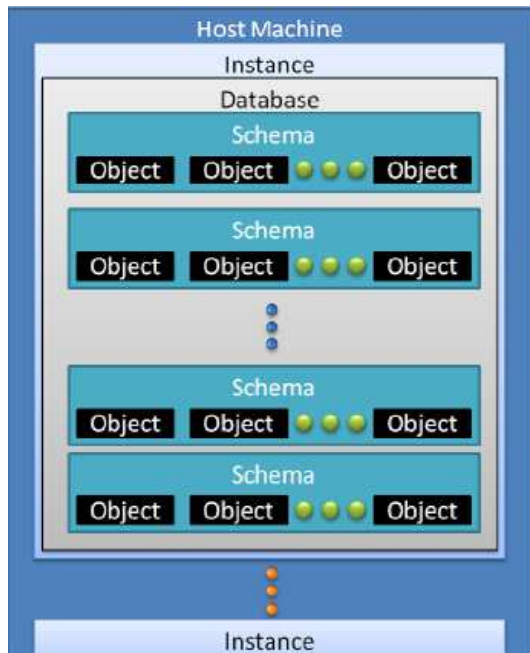
- Consolidate many PDBs onto a single platform
- Fast provisioning of a new PDB or of a clone of an existing PDB
- Fast redeployment, by unplug and plug, of an existing database to a new platform
- Patch or upgrade the Oracle Database version for many PDBs quickly by doing it just once
- Patch or upgrade a single PDB by unplugging it and plugging it into a different CDB at a later version
- Separation of the content of one PDB from that of peer PDBs in the same CDB
- Separation of the duties of the application administrators of these PDBs



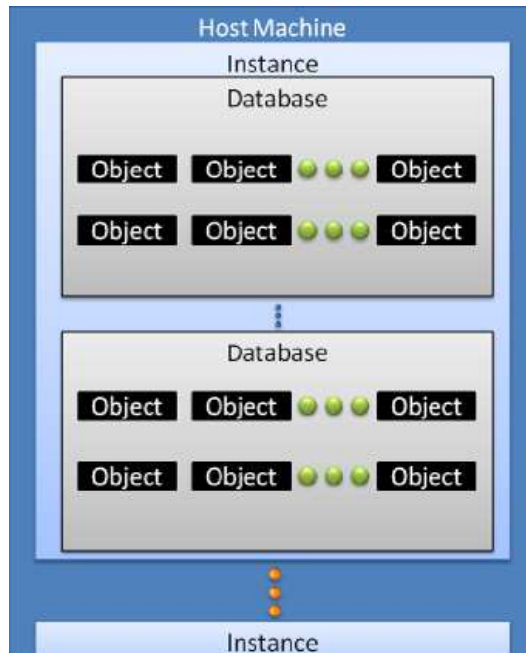
- Conventional non-CDB database mode is still available in Oracle 12c
- More efficient in terms of resource consumption, scalability and manageability
- Based on Oracle show case, PDBs consume 6 times less hardware resources and are 5 times more scalable

ORACLE VS OTHERS

Oracle before-12c database



SQL Server 2000



Oracle 12c & SQL Server 2005



* Encapsulation of instances, databases, schemas and objects

12C PLUGGABLE DATABASE DETAILS

- Each PDB has its own private data dictionary for customer-created database objects; CDB has the data dictionary for the Oracle-supplied system; Each data dictionary defines its own namespace.
- There is a new split data dictionary architecture that allows a PDB to be quickly unplugged from one CDB and plugged into a different CDB
- Each PDB sees a read-only definition of the Oracle-supplied system
- There are global database initialization parameters on CDB level and local PDB ones. PDB parameters belong only to a particular PDB and will be persistent even after you unplug you PDB.
- Database users can be global (CDB) or local (PDB only).
If a new user created in CDB it's seen also in PDB.
In case of creation of a user on PDB level it will stay local.

12C PLUGGABLE DATABASE DETAILS (CONTINUE)

- Temporary tablespaces can be global or local
- Redo logs and Undo tablespace are only global (on CDB level)
- Data Guard acts on the CDB as a whole
- RMAN scheduled backups are done for the CDB as a whole; you can back up and recover a selected PDB whenever you want to
- An application connects, with no code changes, to a PDB; the system administrator connects to the CDB; the service named in the connect string specifies the destination DB
- One PDB knows nothing of the other PDBs within the same CDB; each PDB is a hermetically sealed container. That ensures new level of DB independence and robust security

HANDS-ON LAB : PREPARATION

- **Hands-on Lab databases (12.1 Beta2)**

12c Container DB 1 (CDB1): /u01/app/oracle/oradata/cdb1

12c Container DB 2 (CDB2): /u01/app/oracle/oradata/cdb2

12c Non-Container DB (nonCDB): /u01/app/oracle/oradata/noncdb

- **SQL*Plus Database Connection examples**

-- connect to a CDB (container called CDB\$Root) :
connect sys/pass@localhost:1521/cdb1 as sysdba

-- connect to a PDB (container My_PDB):
connect scott/pass@localhost:1521/My_PDB

- **Discover the current container**

```
select Sys_Context('Userenv', 'Con_Name') "current container" from dual;
```

```
SQL+> show Con_Name
```


LAB 1: CREATE AND OPEN A PDB

- **Connect to a Container Database (CDB)**

```
sqlplus sys/pass@localhost:1521/cdb1 as sysdba
```

- **Create a new Oracle 12c Pluggable Database (PDB)** (by cloning PDB\$Seed template)

```
create pluggable database My_PDB  
admin user App_Admin identified by password  
file_name_convert = ('/pdbseed/', '/my_pdb/');
```

- **Open PDB**

```
alter pluggable database My_PDB open;
```

- **Verify the PDB datafiles**

```
select Con_ID, Tablespace_Name, File_Name from CDB_Data_Files  
where File_Name like '%/cdb1/pdbseed/%' or File_Name like '%/cdb1/my_pdb/%';
```

CON_ID	Tablespace_Name	File_Name
2	SYSAUX	/u01/app/oracle/oradata/cdb1/pdbseed/sysaux01.dbf
2	SYSTEM	/u01/app/oracle/oradata/cdb1/pdbseed/system01.dbf
3	SYSAUX	/u01/app/oracle/oradata/cdb1/My_PDB/sysaux01.dbf
3	SYSTEM	/u01/app/oracle/oradata/cdb1/My_PDB/system01.dbf

LAB 2: SETTING OPEN_MODE FOR A PDB / DROP PDB

- **Each PDB within a CDB has its own Open_Mode and Restricted status**

```
select Name, Open_Mode, Restricted, Inst_ID from gv$PDBs;
```

```
Open_Mode= MOUNTED | READ ONLY | READ WRITE / Restricted = Yes | No | Null
```

- **Starting an instance (which opens the entire CDB) does not cause PDBs to be opened**

```
sqlplus sys/pass@localhost:1521/cdb1 as sysdba
```

```
startup
```

```
alter pluggable database all open;
```

```
...
```

```
alter pluggable database all close;
```

- **Drop a PDB**

```
alter pluggable database MY_PDB close immediate;
```

```
drop pluggable database MY_PDB including datafiles';
```

LAB 3: CLONE AN EXISTING PDB

- **Switch a PDB to READ ONLY mode before cloning (online mode will come)**

```
alter pluggable database My_PDB close;  
alter pluggable database My_PDB open read only;
```

```
create pluggable database My_Clone from My_PDB  
file_name_convert = ('/my_pdb/', '/my_clone/');
```

```
alter pluggable database My_PDB close;  
alter pluggable database My_PDB open;  
alter pluggable database My_Clone open;
```

- **Verify CDB data files**

```
select Con_ID, File_Name from CDB_Data_Files;
```

LAB 4: UNPLUG PDB FROM CDB1

- **Unplug MY_PDB**

```
alter pluggable database My_PDB close;
```

```
alter pluggable database My_PDB
```

```
unplug into '/u01/app/oracle/oradata/cdb1/my_pdb/my_pdb.xml';
```

- **Backup UNPLUGGED MY_PDB with RMAN (if required)**

- **Drop PDB – remove it from CDB catalog**

```
drop pluggable database My_PDB;
```

-- or preserve data files for later PDB plug-in operation:

```
drop pluggable database My_PDB keep datafiles;
```

LAB 5: PLUG PDB INTO CDB2

- **Connect to cdb2** (/u01/app/oracle/oradata/cdb2)
sqlplus sys/pass@localhost:1521/cdb2 as sysdba;
- **Compatibility Check** (error if NOT compatible)
exec DBMS_PDB.Check_Plug_Compatibility(
PDB_Descr_File =>'/u01/app/oracle/oradata/cdb1/my_pdb/my_pdb.xml')
- **Plug In My_PDB**

```
create pluggable database My_PDB  
using '/u01/app/oracle/oradata/cdb1/my_pdb/my_pdb.xml'  
move file_name_convert = ('/cdb1/', '/cdb2/');
```

```
alter pluggable database My_PDB open;
```

LAB 6: CREATE PDB AS A CLONE OF AN UNPLUGGED PDB

- **Clone From Sample_Schemas “Gold Image”**

```
create pluggable database Sample_Schemas_1 as clone
using
'/u01/app/oracle/oradata/gold_pdb/sample_schemas/sample_schemas.xml'
copy
file_name_convert =
('/gold_pdb/sample_schemas/', '/cdb1/sample_schemas_1/');
```

```
alter pluggable database Sample_Schemas_1 open;
```

- **Show PDB GUID**

```
select PDB_Name, GUID
from DBA_PDBs
order by Creation_scn;
```

LAB 7: ADOPTING NON-CDB AS A PDB INTO EXISTING CDB

- **Upgrade your existing database into a 12.1 non-CDB**

- **Generate the xml manifest file of nonCDB**

```
shutdown immediate
```

```
startup mount
```

```
alter database open read only;
```

```
begin
```

```
DBMS_PDB.Describe(PDB_Descr_File => '/u01/app/oracle/oradata/noncdb/noncdb.xml');
```

```
end;
```

```
/
```

```
shutdown immediate
```

- **Connect to CDB1**

```
sqlplus sys/pass@localhost:1521/cdb1 as sysdba
```

LAB 7: CONTINUE

- **Plug in the datafiles of nonCDB into CDB1**

```
create pluggable database ExNonCDB as clone
using '/u01/app/oracle/oradata/noncdb/noncdb.xml'
source_file_name_convert = none
copy file_name_convert = ('/noncdb/', '/cdb1/exnoncdb/')
storage unlimited;
```

- **Open it, to finalize the plug-in, close it, and re-open it Restricted mode**

```
alter pluggable database ExNonCDB open;
alter pluggable database ExNonCDB close;
alter pluggable database ExNonCDB open restricted;
```

- **Remove not required data from the PDB local data dictionary**

```
alter session set container = ExNonCDB;
@?/rdbms/admin/noncdb_to_pdb.sql
```

- **Open the newly-adopted PDB (former non-CDB)**

```
alter pluggable database ExNonCDB open;
```


ORACLE 12C CDB / PDB CHARACTERISTICS

	CDB	PDB
DB name example	cdb1	MY_PDB
Container name	CDB\$Root	MY_PDB
Service name	cdb1	MY_PDB
Tablespaces	system, sysaux, temp, undo	system, sysaux, (temp is possible)
Redo logs	Yes	No
Data Dictionary	Own global	Own local
PDB Template	PDB\$Seed	
PDB manifest		XML file

FAQ?

THANK YOU!

Kirill Loifman

www.dadbm.com