

Application Containers – an Introduction

Oracle Database 12c Release 2 - Multitenancy for Applications


Markus Flechtner

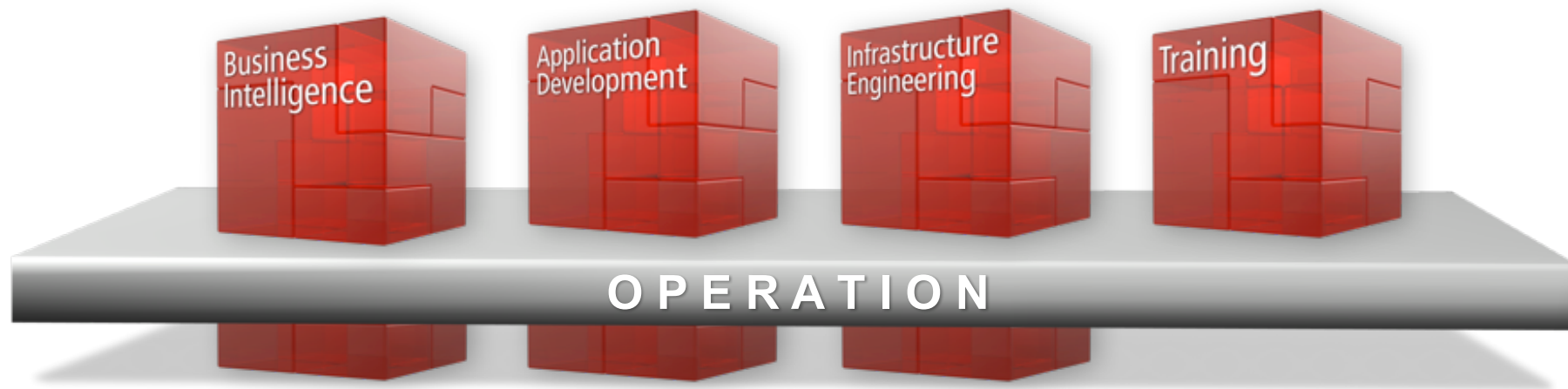


BASLE ▪ BERN ▪ BRUGG ▪ DÜSSELDORF ▪ FRANKFURT A.M. ▪ FREIBURG I.BR. ▪ GENEVA
HAMBURG ▪ COPENHAGEN ▪ LAUSANNE ▪ MUNICH ▪ STUTTGART ▪ VIENNA ▪ ZURICH

trivadis
makes IT easier. ■ ■ ■

■ Our company.

Trivadis is a **market leader in IT consulting, system integration, solution engineering** and the provision of **IT services** focusing on **ORACLE®** and  **Microsoft** technologies in Switzerland, Germany, Austria and Denmark. We offer our services in the following strategic business fields:



Trivadis Services takes over the interactive operation of your IT systems.

■ With over 600 specialists and IT experts in your region.



- 14 Trivadis branches and more than 600 employees
- 200 Service Level Agreements
- Over 4,000 training participants
- Research and development budget: CHF 5.0 / EUR 4 million
- Financially self-supporting and sustainably profitable
- Experience from more than 1,900 projects per year at over 800 customers

trivadis
makes IT easier. ■ ■ ■

■ About me .. Markus Flechtner

- Principal Consultant, Trivadis, Duesseldorf/Germany, since April 2008
- Discipline Manager Infrastructure Database @Trivadis
- Working with Oracle since the 1990's
 - Development (Forms, Reports, PL/SQL)
 - Support
 - Database Administration
- Focus
 - Oracle Real Application Clusters
 - Database Migration Projects
- Teacher
 - O-RAC – Oracle Real Application Clusters
 - O-NF12CDBA – Oracle 12c New Features for the DBA



Blog:

<https://www.markusdba.net/>



trivadis
makes IT easier. ■ ■ ■

■ Agenda

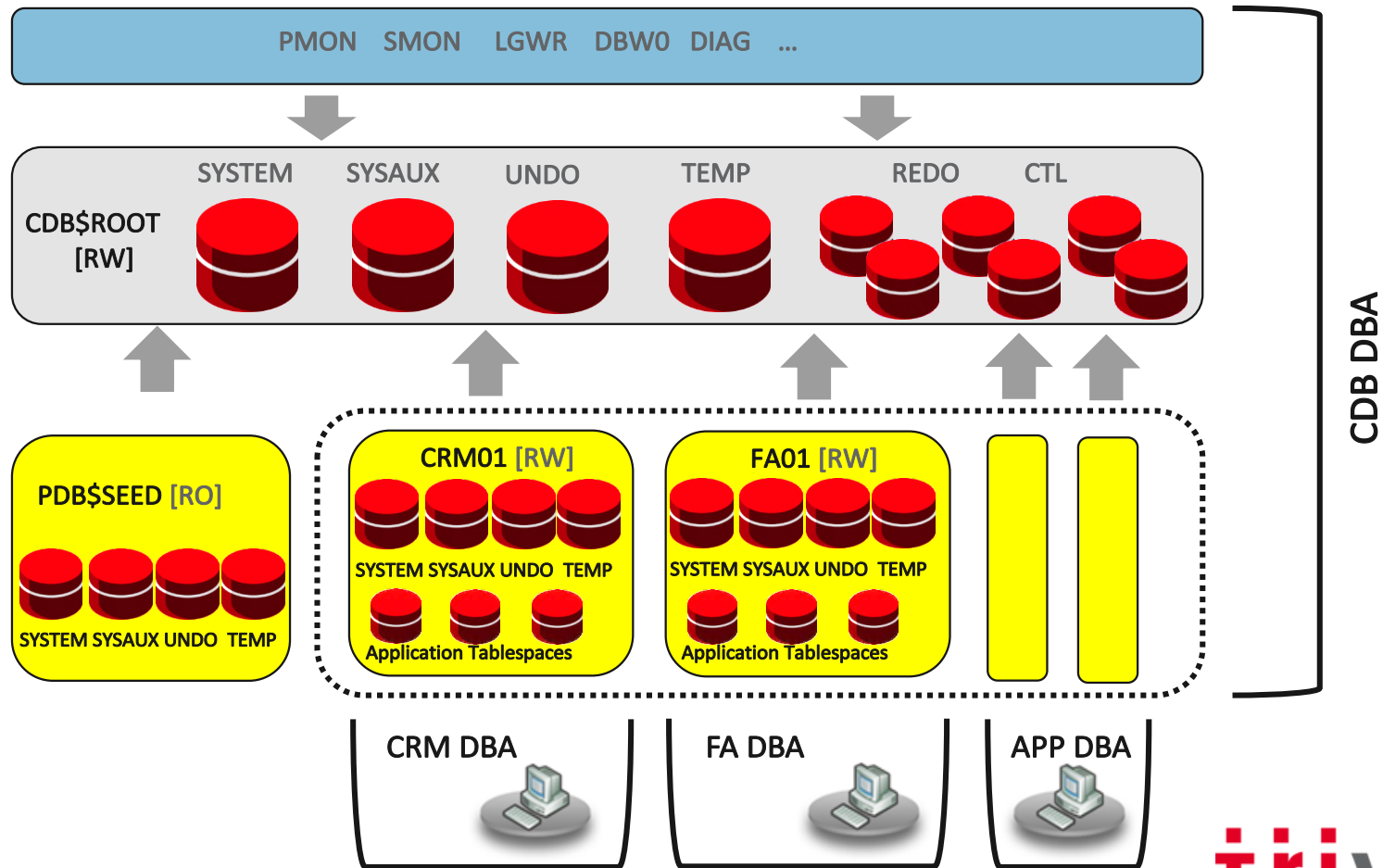
1. **Container Database Architecture**
2. **Application Containers – Overview**
3. **Application Common Objects**
4. **Installing Applications**
5. **Upgrading and Patching Applications**
6. **Administration & Daily Business**
7. **Summary**

Container Database Architecture

■ Oracle Multitenant in Oracle Database 12c

- The **container database architecture (multitenant architecture)** introduced in Oracle Database 12c Release 1 enables an Oracle database to work as a **container database (CDB)**
- A new database architecture designed for:
 - consolidation/database virtualization
 - fast and easy provisioning
 - separation of administrative duties
 - rapid movement of user data (unplug/plugin)
- Pluggable databases (PDBs) are **compatible** with traditional non-CDB (same behaviour from the application point of view)

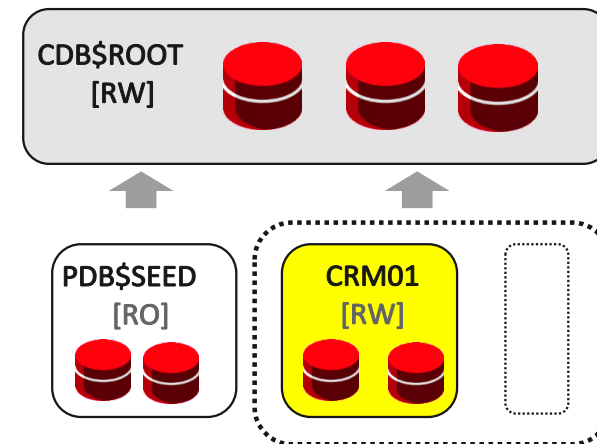
■ Container Database Architecture - overview



■ Container Database Architecture - Sharing

- To avoid the duplication of system metadata, a CDB uses a new object property called **SHARING**

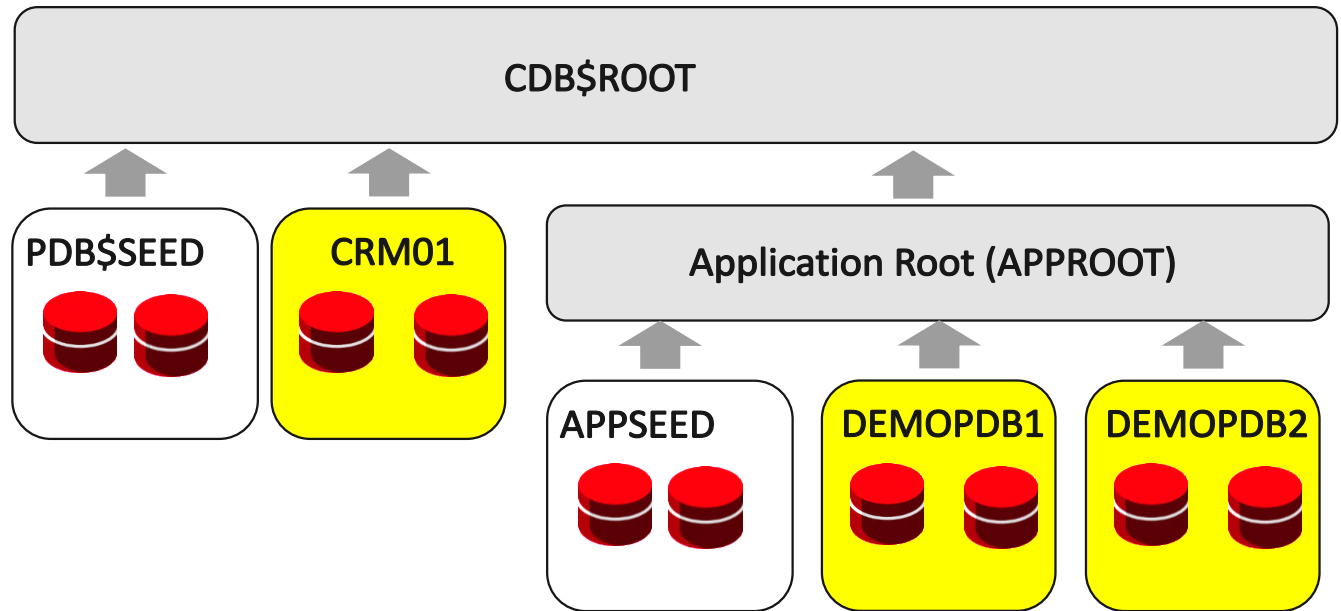
```
SQL> SELECT sharing, count(*)  
       2 FROM dba_object GROUP BY sharing;  
SHARING                COUNT (*)  
-----  
METADATA LINK          66457  
DATA LINK              214  
EXTENDED DATA LINK    56  
NONE                   5053
```



Application Containers Overview

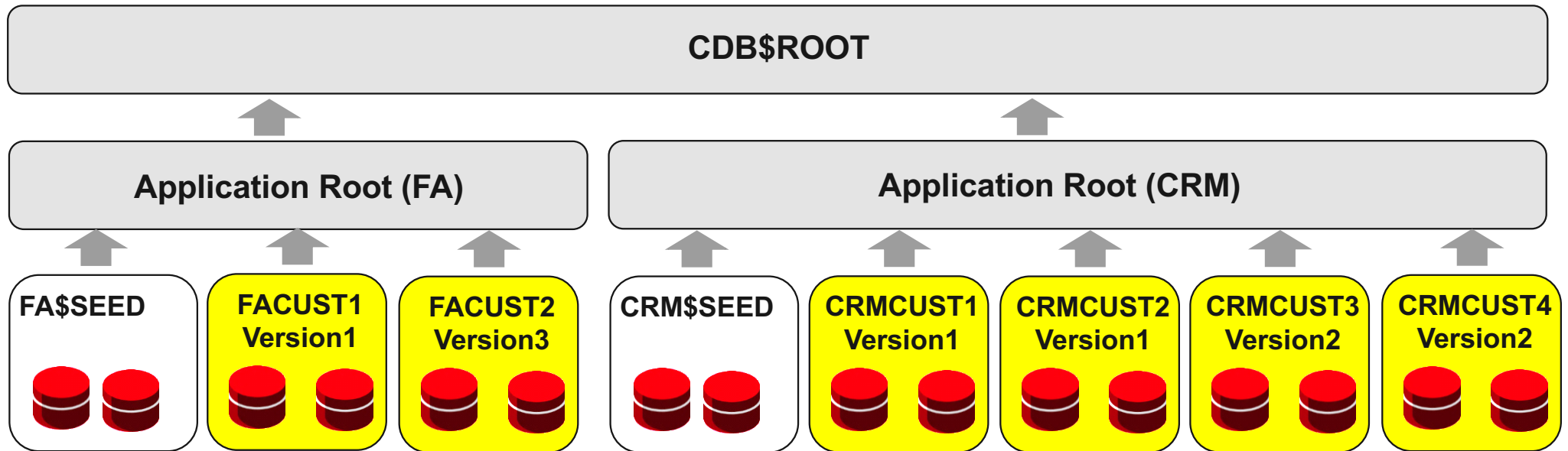
■ Application Containers - Overview

- Consist of
 - Application Root
 - Application Seed
 - Application PDBs
- Applications can share
 - Data model
 - Code
 - Metadata
 - Data



■ Application Containers – Use Cases

- Software as a Service (SaaS)
- Better master data management
- Coordinated and simplified upgrades
- „manage many as one“



Application Common Objects

■ Sharing - Application Common Objects

■ Application Common Objects (Sharing-Attribute)

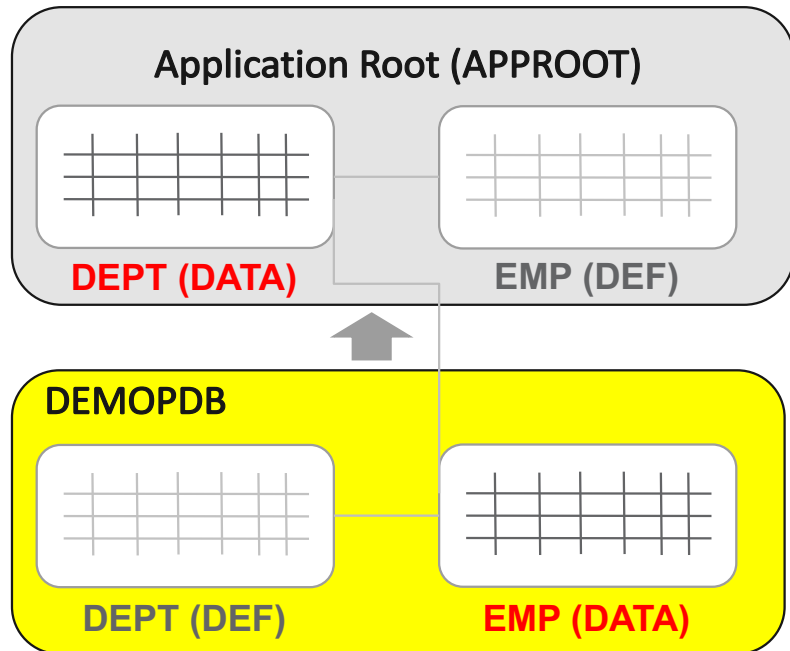
Sharing	Definition	Data	
METADATA	APPROOT	APPPDB	
DATA	APPROOT	APPROOT	Data accessible from Application PDB (RO)
EXTENDED DATA	APPROOT	APPROOT APPPDB	Data is stored in Application Root and can be retrieved from all Application PDBs (RO) Additional PDB-level data can be stored in Application PDB
NONE	local	local	

```
CREATE TABLE <table_name> SHARING=METADATA  
( col1 .. )
```

■ Which sharing methods are allowed?

Object Type	Metadata Link	Data Link	Extended Data Link	None
Table	YES	YES	YES	YES
Views	YES	YES	YES	YES
Sequences	YES	YES	NO	YES
All others	YES	NO	NO	YES

Foreign Keys to Data Linked Tables



```
APPROOT SQL> create table DEPT SHARING=DATA  
(DEPTNO number constraint PK_DEPT primary key,  
 DNAME varchar2(30));
```

```
APPROOT SQL> create table EMP SHARING=METADATA  
(EMPNO number constraint PK_EMP primary key,  
 ENAME varchar2(30) not null,  
 DEPTNO constraint FK_DEPT references DEPT);
```

```
APPROOT SQL> insert into DEPT ..
```

```
DEMOPDB SQL> insert into EMP ..
```

```
ORA-02291: integrity constraint (SCOTT.FK_DEPTNO) violated - parent key not found
```

- Bug 21955394 (Patches are available for 12.2 Base Release and RU 12.2.0.1.170718)

■ Be Careful with DML on Application Container Objects

■ DML from Application PDB on an a Data-Linked Table:

```
SQL> update scott.dept set loc='SEATTLE' where loc='CHICAGO';
update scott.dept set loc='SEATTLE' where loc='CHICAGO'
*ERROR at line 1:
ORA-65097: DML into a data link table is outside an application action
```

■ DML from Application PDB on a Extended Data-Linked Table (row from Application Root):

```
SQL> select * from scott.zip_codes where zip_code='40227';
CO ZIP_C CITY
-- -----
DE 40227 Duesseldorf

SQL> update scott.zip_codes set zip_code='44444' where zip_code='40227';
0 rows updated.
```

Installing Applications

■ Prepare the Application in the Application Root (1)

- Create Application Root (similar to CDB\$ROOT)

Not documented!

- Oracle Managed Files (OMF) are required when using Application Containers

```
create pluggable database DEMO AS APPLICATION CONTAINER
admin user admin identified by manager;
```

- Create the application in the Application Root

```
alter pluggable database application DEMOAPP begin install '1.0';
.. create tablespaces (OMF)
.. create users
.. create application objects
alter pluggable database application DEMOAPP end install '1.0';
```

- During application creation the statements are captured

■ Prepare the Application in the Application Root (2)

■ Create Application Common Objects

Works!

```
SQL> Alter session set container=DEMO;  
SQL> alter pluggable database application DEMOAPP begin install '1.0';  
  
SQL> show user  
USER is "SYS"  
  
SQL> create table SCOTT.EMP SHARING=METADATA [...]   
Table created.  
  
alter pluggable database application DEMOAPP end install '1.0';
```

■ Prepare the Application in the Application Root (3)

■ Create Application Common Objects (as object owner)

```
SQL> Alter session set container=DEMO;  
SQL> alter pluggable database application DEMOAPP begin install '1.0';  
SQL> create user scott ..
```

```
SQL> connect scott/tiger@DEMO
```

```
SQL> create table EMP SHARING=METADATA [..]
```

```
ORA-65021 illegal use of „SHARING“-clause
```

■ Prepare the Application in the Application Root (4)

Not documented!

- Create Application Common Objects (as object owner)

```
SQL> alter pluggable database application DEMOAPP begin install '1.0';
```

```
SQL> exec dbms_application_info.set_module('INSTALL_V1',null);
```

```
SQL> create user scott ..
```

```
SQL> connect scott/tiger@DEMO
```

```
SQL> exec dbms_application_info.set_module('INSTALL_V1',null);
```

```
SQL> create table EMP SHARING=METADATA [..]
```

Table created.

```
SQL> connect sys/manager@DEMO as sysdba
```

```
SQL> exec dbms_application_info.set_module('INSTALL_V1',null);
```

```
SQL> alter pluggable database application DEMOAPP end install '1.0';
```

■ Create an Application PDB

- Two ways to create an Application PDB
 - Via an Application Seed
 - Directly from Application Root
- During the PDB creation the statements are „replayed“

■ Application Seed

- Similar to PDB\$SEED
- Optional
- Only one Application seed per Application container allowed

```
SQL> CREATE PLUGGABLE DATABASE AS SEED
      2 ADMIN USER app_admin IDENTIFIED BY manager;
Pluggable database created.
```

```
SQL> show pdbs
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
3	DEMO	READ WRITE	NO
4	DEMO\$SEED	MOUNTED	

■ Create an Application PDB (1) – from Application Seed

■ Step 1: Install Application in Application Seed

```
ALTER PLUGGABLE DATABASE DEMO$SEED OPEN;  
ALTER SESSION SET CONTAINER=DEMO$SEED;  
ALTER PLUGGABLE DATABASE APPLICATION DEMO SYNC;  
ALTER PLUGGABLE DATABASE CLOSE IMMEDIATE;  
ALTER PLUGGABLE DATABASE OPEN READ ONLY;
```

■ Clone the Application Seed

```
create pluggable database DEMOPDB1 from DEMO$SEED;  
  
alter pluggable database DEMOPDB1 open;
```

■ Create an Application PDB (2) – from Application Root

```
ALTER SESSION SET CONTAINER=DEMO;  
  
create pluggable database DEMOPDB1  
admin user admin identified by admin;  
  
Alter pluggable database DEMOPDB1 open;  
  
ALTER SESSION SET CONTAINER=DEMOPDB1;  
Alter pluggable database application DEMOAPP sync;  
  
.. after that the application can be used in DEMOPDB1
```

Patching and Upgrading Applications

■ Application Containers – Upgrade an Application (1)

■ Upgrade the application in the Application Root

```
alter pluggable database application DEMOAPP  
begin upgrade from '1.0' to '2.0';
```

```
.. modify application objects
```

```
alter pluggable database application DEMOAPP end upgrade to '2.0';
```

■ Upgrade in Application PDB

```
Alter pluggable database application DEMOAPP sync;
```

■ Application Containers – Upgrade an Application (2)

- During an upgrade, a read-only clone of the source Application Root is created
- From the alert.log:

```
DEMO(5):alter pluggable database application DEMOAPP begin upgrade '1.0' to '2.0'  
CREATE PLUGGABLE DATABASE "F613214177_3_1" AS APPLICATION CONTAINER from "DEMO"  
CREATE_FILE_DEST='/u01/oradata'  
[...]  
Completed: CREATE PLUGGABLE DATABASE "F613214177_3_1" AS APPLICATION CONTAINER from  
"DEMO" CREATE_FILE_DEST='/u01/oradata' [...]  
ALTER PLUGGABLE DATABASE "F613214177_3_1" OPEN READ ONLY INSTANCES=ALL  
[...]  
Completed: ALTER PLUGGABLE DATABASE "F613214177_3_1" OPEN READ ONLY INSTANCES=ALL
```

- Purpose: Application Root for Application PDBs not upgraded to the new release

■ Patching an Application

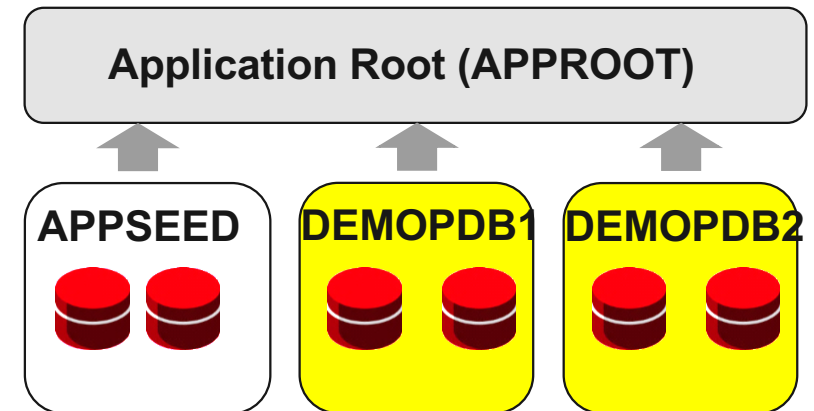
```
alter pluggable database application DEMOAPP  
begin patch <Patch#>  
  
.. modify application objects  
  
alter pluggable database application DEMOAPP end patch <Patch#>;
```

- Limited set of operations (e.g. no DROP commands, no ALTER TABLE)
- A minimum start version can be defined

Administration & Daily Business

■ Administration

- All administrative tasks are executed from the application root
- CREATE/OPEN/CLOSE/DROP work as with normal PDBs
 - But closing the Application Root will clause all Application PDBs
- Application Root can only be dropped if there are no Application PDBs
- A normal PDB can be converted to an Application Container



■ Backup & Recovery

- When connected to CDB\$ROOT
 - Same as with usual PDBs
- When connected to Application Root

```
RMAN> CONNECT TARGET sys/manager@oracle12c:1521/demoapp
RMAN> BACKUP DATABASE ROOT;           # backup application root
RMAN> BACKUP PLUGGABLE DATABASE DEMOPDB1; # backup application PDB
RMAN> BACKUP DATABASE;               # backup application root + PDBs
```

- For recovery the same principles apply

■ Execution Plans (1) – Data Linked Table

```
select  e.empno, e.ename, d.deptno, d.dname
  2    from scott.emp e  -- local in application PDB
  3          scott.dept d -- from application root (data linked)
  4    where d.deptno=e.deptno;
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		1	35	3 (0)
* 1	HASH JOIN		1	35	3 (0)
2	DATA LINK FULL	DEPT	1	22	
3	TABLE ACCESS FULL	EMP	14	182	3 (0)

■ Execution Plans (2) – Extended Data Linked Table

```
select * from scott.zip_codes;
```

Id	Operation	Name	Pstart	Pstop	TQ	IN-OUT	PQ	Distrib
0	SELECT STATEMENT							
1	PX COORDINATOR							
2	PX SEND QC (RANDOM)	:TQ10000			Q1,00	P->S		QC (RAND)
3	PX PARTITION LIST ALL		1	2	Q1,00	PCWC		
4	EXTENDED DATA LINK FULL	ZIP_CODES			Q1,00	PCWP		

■ Data Dictionary Views (CDB_APP%)

- CDB_APPLICATIONS
- CDB_APPLICATION_ROLES
- CDB_APP_ERRORS
- CDB_APP_PATCHES
- CDB_APP_PDB_STATUS
- CDB_APP_STATEMENTS
- CDB_APP_VERSIONS

Summary

■ Application Containers - Summary

- + Interesting concept for SaaS
- + Easy upgrade and patching of lots of Application PDBs
- It's a "release 1.0" of a new feature
- Some flaws
- Oracle Managed Files (OMF) are required
- Documentation is misleading in some places

Further Information



- Oracle 12.2 Concepts (Chapter 19)
- Oracle 12.2 Administrators Guide (Chapter 40 and 44)

Questions and Answers

Markus Flechtner
Principal Consultant

Phone +49 211 5866 64725

Markus.Flechtner@Trivadis.com

 @markusdba <https://www.markusdba.net>

