

# Arbeiten mit Oracle Multitenant in der Praxis

Mike Dietrich  
Oracle Deutschland BV  
München

## Schlüsselworte

Oracle Database 12c Multitenant Pluggable Databases

## Einleitung

Oracle Multitenant Option (vormals bekannt als "Pluggable Databases" oder Container-Datenbanken) erschliesst ein völlig neues Gebiet für Konsolidierung und die Reduktion des Wartungsaufwands. Aber welche Veränderungen werden notwendig, wenn man diese neue Option in Oracle Database 12c einsetzt? Welche praktischen Aspekte ergeben sich daraus? Und wo muss der DBA schlichtweg umdenken?

## Die Architektur

Die Idee hinter Oracle Multitenant ist einfaches Provisioning, schnelles Cloning sowie Patching und Upgrades ohne grossen Aufwand für eine Vielzahl von Datenbanken in einem Arbeitsgang. Dazu werden in eine CDB sogenannte Pluggable Databases eingehängt, die für sich genommen nach aussen wie eine eigenständige Datenbank mit Applikationsschemata, Daten, Prozeduren etc. erscheinen, aus Datenbank-Sicht aber vom Root-Container, der CDB\$ROOT, administriert werden.

- Multitenant container database
  - Administration from CDB
- A CDB can contain one or many pluggable databases
  - Data and code in PDBs
  - Applications connect to PDBs
- Simple and fast provisioning, cloning, patching and upgrade

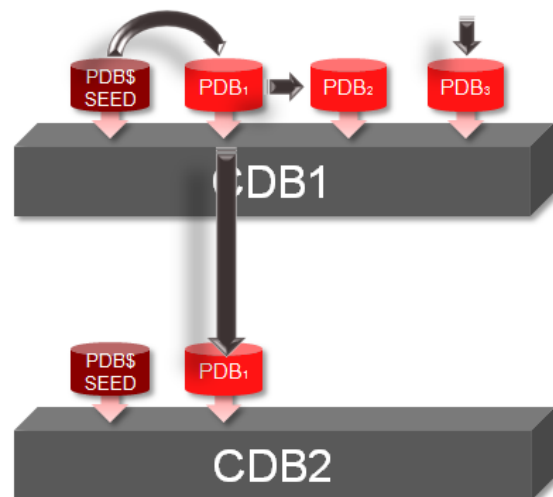


Abb. 1: Die Oracle Multitenant Architektur

Die PDB\$SEED (siehe Bild) existiert nach Anlegen einer CDB immer. Sie agiert als eine Art Blaupause, mit Hilfe deren das schnelle Provisioning ermöglicht wird.

Desweiteren gewährleistet diese Architektur eine effizientere Nutzung von System-Ressourcen, da nur die SGA für die CDB existiert, die alle PDBs umfasst, genauso, wie es nur ein Set von Hintergrundprozessen für die CDB gibt, ein Set von Controlfiles, sowie je einen Stream für Redo- und für Flashbacklogs. Im SPFILE finden sich die Parameter, die für die CDB und alle PDBs gemeinsam gelten.

- One SGA
- One set of background processes
- One SPFILE

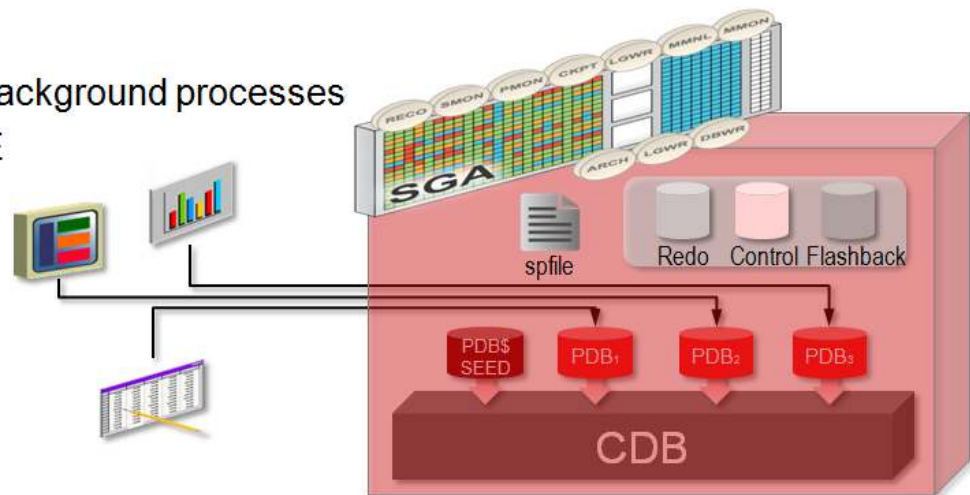


Abb.2: SGA und Hintergrundprozesse

Applikationen verbinden sich über den Service-Namen wie gehabt gegen die Datenbank, allerdings nun gegen die PDB. In der CDB existieren keine Nutzerdaten. Es handelt sich architekturell betrachtet um eine reine Administrationsschicht.

Aus dieser Architektur leitet sich auch ab, dass die Hardware-Ressourcen besser genutzt werden können, da eben keine 10x Sets von Hintergrundprozessen für 10 separate Datenbanken vorhanden sind, keine 10 Shared Memory Segmente allokiert und verwaltet werden müssen. Auf der anderen Seite sollte aber nur PDBs mit ähnlichem Last- und Anforderungsprofil zusammen konsolidiert werden. Um das Lastverhalten genauer zu untersuchen bietet übrigens Oracle Real Application Testing eine sehr gute Unterstützung, da es das Aggregieren von Workloads aus unterschiedlichen Systemen zur Wiedergabe gegen einen Multitenant Architektur erlaubt.

### Wege nach Oracle Multitenant

4 unterschiedliche Ansätze führen in die Oracle Multitenant-Architektur:

- Provisionierung einer neuen PDB und anschliessendes Befüllen
- Clonen einer existieren PDB – lokal oder remote
- Upgrade einer stand-alone Datenbank nach Oracle Database 12c und plugin in eine CDB
- Full Transportable Export-Import

### Provisierung einer neuen PDB

Der Parameter `PDB_FILE_NAME_CONVERT` gibt an, wo die neue PDB entstehen soll. Dazu muss zuerst das Directory angelegt worden sein. Dann wird die neue PDB mit dem Kommando `CREATE PLUGGABLE DATABASE PDB1 ADMIN USER ADM IDENTIFIED BY ADM;` angelegt.

- **Fast provisioning from PDB\$SEED**

```
create pluggable database PDB1  
admin user adm1 identified by pwd;
```

- `PDB_FILE_NAME_CONVERT`

- **Import data with `impdp`**

- Dump file or `NETWORK_LINK`
- `imp` for  $\leq$  Oracle 9i

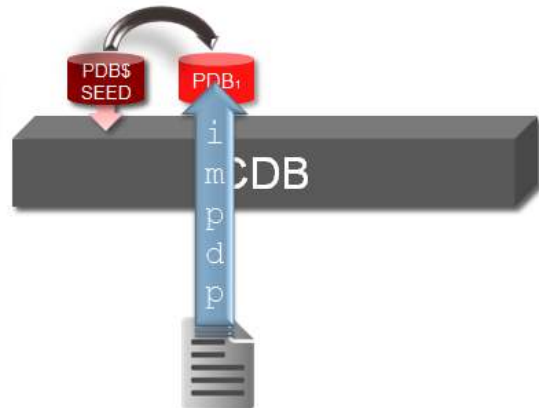


Abb3: Schnelle Provisionierung

Anschliessend kann die PDB mit den Oracle-üblichen Bordmitteln wie zum Beispiel Data Pump oder dem Import befüllt werden.

### Cloning einer existierenden PDB

Schnelles und einfaches Cloning einer existierenden PDB ist ein weiteres, wichtiges Merkmal des Konzepts. Das Remote-Cloning wird erst mit dem ersten Patch Set 12.1.0.2 funktionieren, das voraussichtlich auch die Beschränkung, eine PDB für das Cloning in den Read-Only-Modus versetzen zu müssen, aufheben könnte.

- **Fast cloning of a PDB**

- Local:

```
create pluggable database  
PDB2 from PDB1;
```

- Remote:

```
create pluggable database  
PDB1 from PDB1@CDB1;
```

- Will work with Oracle 12.1.0.2 onwards

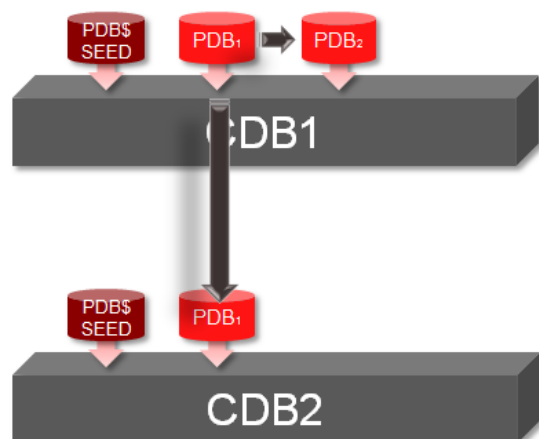


Abb4: Einfaches Cloning – lokal und remote

### Upgrade und Plugin einer existierenden Datenbank

Eine weitere Technik, die sicherlich häufig zum Einsatz kommen wird, ist das Upgrade einer bestehenden Datenbank nach Oracle Database 12c mit anschließendem Plugin in eine existierende CDB. Dazu wird nach dem Upgrade ein XML File erzeugt, das das physikalische Layout der Datenbank beschreibt. Anhand dieses XML Files wird die Datenbank dann in die CDB eingehängt. Schliesslich sind noch Schritte notwendig, um Einstellungen und das Dictionary der neuen PDB anzupassen, was durch die Ausführung des Skripts `noncdb_to_pdb.sql` durchgeführt wird.

- Database upgrade
- Start database read-only
- Create XML description file

```
exec DBMS_PDB.DESCRIBE ('PDB1.xml');
```
- Shutdown database
- Plugin database

```
create pluggable database PDB1
using ('PDB1.xml') nocopy tempfile reuse;
```
- Sanity operations

```
start ?/rdbms/admin/noncdb_to_pdb.sql
```

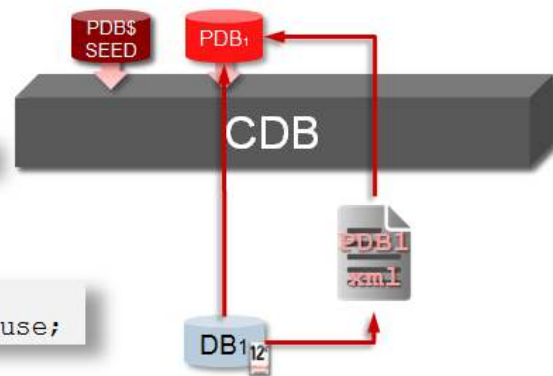


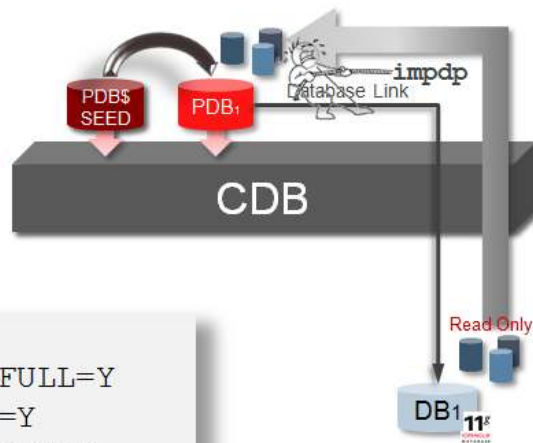
Abb5: Upgrade und Plugin

### Full Transportable Export-Import in eine PDB

Neu in Oracle Database 12c hinzugekommen ist die Möglichkeit, eine PDB auch über alle Plattformgrenzen hinweg mit einer Technik zu befüllen, die eine Kombination aus Transportable Tablespace mit der Oracle Data Pump darstellt. Dabei übernimmt die Data Pump alle bisherigen manuellen Schritte von TTS. Zusätzlich kann *minimal downtime* mit Hilfe von *RMAN Incremental Backups* erreicht werden, die RMAN nun auch über Plattformgrenzen hinweg konvertieren und anwenden kann.

Spannend an diesem Feature ist, dass es bereits mit Oracle 11.2.0.3 und Oracle 11.2.0.4 Datenbanken als Quelle funktioniert. Die Data Pump kann dazu Informationen im Oracle 12c Format liefern (VERSION=12). Ziel muss aber mindestens eine Oracle Database 12c Umgebung mit COMPATIBLE ≥ 12.0.0 sein.

- Create a fresh database/PDB
- Create database link to source
- Tablespaces read-only – *downtime!*
- Copy datafiles to destination
- Run impdp on NETWORK\_LINK



```
impdp oow/passwd@PDB1
NETWORK_LINK=DB1 VERSION=12 FULL=Y
TRANSPORTABLE=ALWAYS METRICS=Y
LOGFILE=oow_dir:src112fullimp.log
TRANSPORT_DATAFILE='/oradata/ts1.dbf' ...
```

Abb6: Full Transportable Export-Import

**Kontaktadresse:**

Mike Dietrich  
 Oracle Deutschland BV  
 Riesstr. 25  
 D-80992 München

Telefon: +49 89 1430-0  
 E-Mail: Mike.Dietrich@oracle.com  
 Internet: <http://blogs.oracle.com/UPGRADE>