



Multitenant For Free

Johannes Ahrends (CarajanDB GmbH)

- Oracle Spezialist seit 1992
 - 1992: Presales bei Oracle in Düsseldorf
 - 1999: Projektleiter bei Herrmann & Lenz Services GmbH
 - 2005: Technischer Direktor ADM Presales bei Quest Software GmbH
 - 2011: Geschäftsführer CarajanDB GmbH
- 2011 → Ernennung zum Oracle ACE
- Autor der Bücher:
 - Oracle9i für den DBA, Oracle10g für den DBA, Oracle 11g Release 2 für den DBA
- DOAG Themenverantwortlicher Datenbankadministration, Standard Edition
- Hobbies:
 - Drachen steigen lassen (Kiting) draußen wie drinnen (Indoorkiting)
 - Motorradfahren (nur draußen)
 - Singen (überall)





Multitenant Database

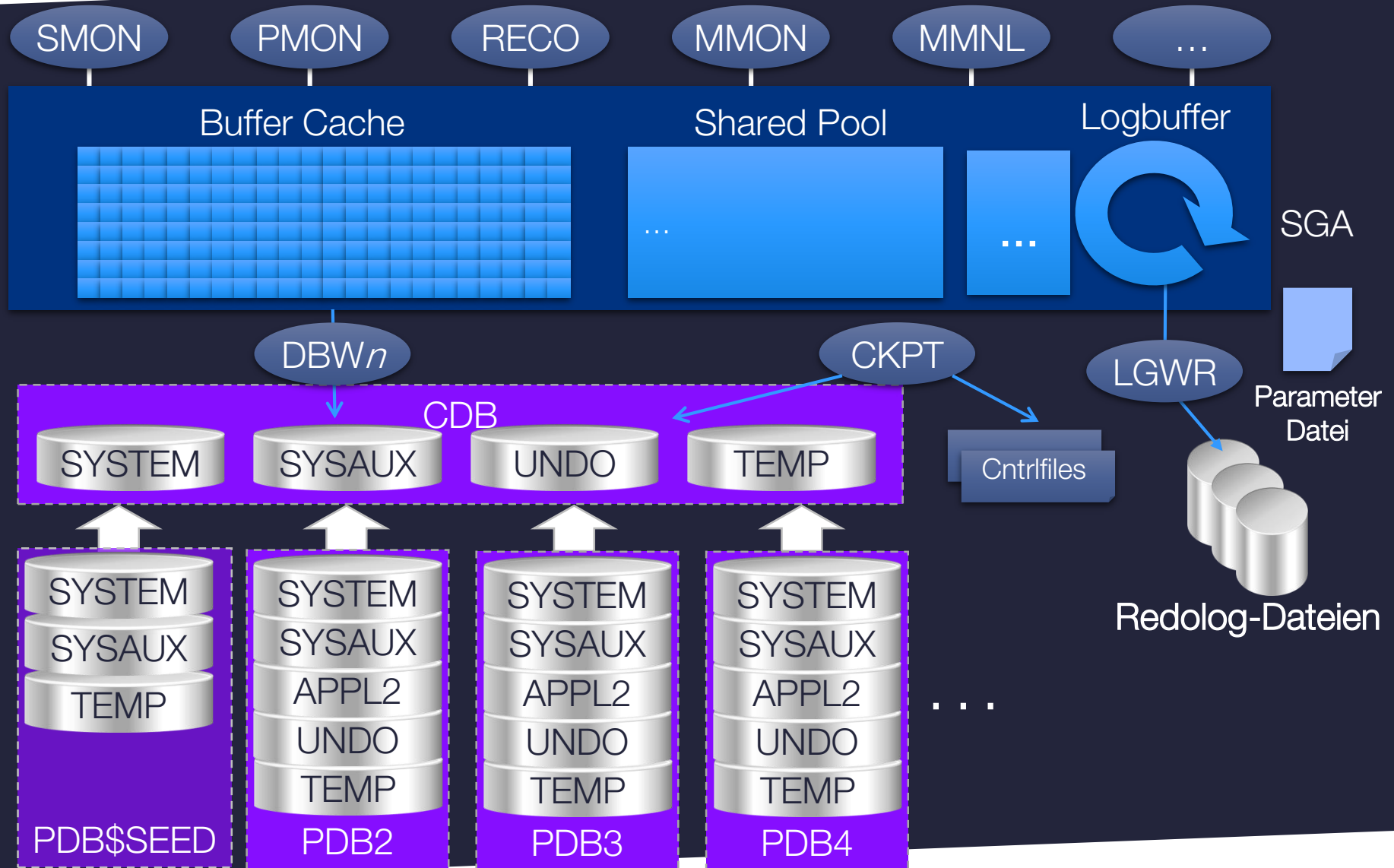
- non-CDB
 - Architektur bis Oracle 11.2
- Multitenant Architektur
 - Alternative Architektur ab Version 12.1.1 für alle Editionen
- Multitenant Option
 - Kostenpflichtig für die Enterprise Edition
 - Bis zu 253 PDBs pro CDB (Exadata ab 12.2 4096 PDBs)
- The non-CDB architecture was deprecated in Oracle Database 12c. It can be desupported and unavailable in a release after Oracle Database 12c Release Oracle recommends use of the CDB architecture.
- The non-CDB architecture will be desupported from Oracle 20c on (OOW 2019 Dominic Gilles)

3 PDBs in jeder Edition

Feature / Option / Pack	SE2	EE	EE-ES	DBCS SE	DBCS EE	DBCS EE-HP	DBCS EE-EP	ExaCS	Notes
Oracle Multitenant - # of PDBs	3	252	4096	3	3	4096	4096	4096	The number value in each column indicates the maximum number of pluggable databases (PDBs) that can be created for each offering. - For all offerings, if you are not licensed for Oracle Multitenant, then you may have up to 3 PDBs in a given container database at any time. EE: Extra cost option; if you are licensed for Oracle Multitenant, then you can create up to 252 PDBs. EE-ES: Extra cost option; if you are licensed for Oracle Multitenant, then you can create up to 4096 PDBs. DBCS EE-HP, DBCS EE-EP, and ExaCS: Included option; you can create up to 4096 PDBs.
CDB Fleet Management	N	N	Y	N	Y	Y	Y	Y	
PDB Snapshot Carousel	N	N	Y	N	Y	Y	Y	Y	
Refreshable PDB switchover	N	N	Y	Y	Y	Y	Y	Y	

Quelle: Oracle Database Database Licensing Information User Manual 19c E94254-10 September 2019

Multitenant Database 12.2



- Geht nicht in unserer Umgebung!
- Da kann ja eine PDB alle Ressourcen verbrauchen
- Wir haben unterschiedliche Zeichensätze in unseren Datenbanken
- Wir müssen die Datenbank im Fehlerfall (Batch) zurücksetzen können



Installation

Erstellen einer CDB - Einfach

Database Configuration Assistant - Create Database - Step 2 of 5

ORACLE 12c DATABASE

Creation Mode

- Database Operation
- Creation Mode**
- Pre Requisite Checks
- Summary
- Progress Page

☒ Create a database with default configuration

Global Database Name: CDBACFS

Storage Type: File System

Database Files Location: /u02/oradata

Fast Recovery Area: /u03/orabackup

Database Character Set: AL32UTF8 - Unicode UTF-8 Universal character set

Administrative Password:

Confirm Password:

☒ Create As Container Database

Pluggable Database Name: PLUGACFS1

☐ Advanced Mode

Erstellen einer CDB - Besser

Database Configuration Assistant - Create a database - Step 2 of 14@trzlora101.carajandb.intra

Select Database Creation Mode

ORACLE 12c DATABASE

Database Operation

Creation Mode

Deployment Type

Database Identification

Storage Option

Fast Recovery Option

Database Options

Configuration Options

Management Options

User Credentials

Creation Option

Summary

Progress Page

Finish

☐ Typical configuration

Global database name: orcl.carajandb.intra

Storage type: Automatic Storage Management (ASM)

Database files location: +OCR/{DB_UNIQUE_NAME} Browse...

Fast Recovery Area (FRA): +MCMT Browse...

Database character set: AL32UTF8 - Unicode UTF-8 Universal character set

Administrative password:

Confirm password:

☒ Create as Container database

Pluggable database name:

☒ Advanced configuration

Help < Back Next > Finish Cancel

Erstellen einer CDB – Ohne PDB

Database Configuration Assistant - Create a database - Step 5 of 16@trzlora101.carajandb.intra

Specify Database Identification Details

ORACLE 12c DATABASE

Database Operation
Creation Mode
Deployment Type
Nodes Selection
Database Identification
Storage Option
Fast Recovery Option
Database Options
Configuration Options
Management Options
User Credentials
Creation Option
Prerequisite Checks
Summary
Progress Page
Finish

Provide a unique database identifier information. An Oracle database is uniquely identified by a Global database name, typically of the form "name.domain".

Global database name: TC01.carajandb.intra

SID Prefix: TC01

Service name:

☒ Create as Container database

A Container database can be used for consolidating multiple databases into a single database, and it enables database virtualization. A Container database (CDB) can have zero or more pluggable databases (PDB).

☒ Use Local Undo tablespace for PDBs

☒ Create an empty Container database

☐ Create a Container database with one or more PDBs

Number of PDBs: 1

PDB name: TC01pdb

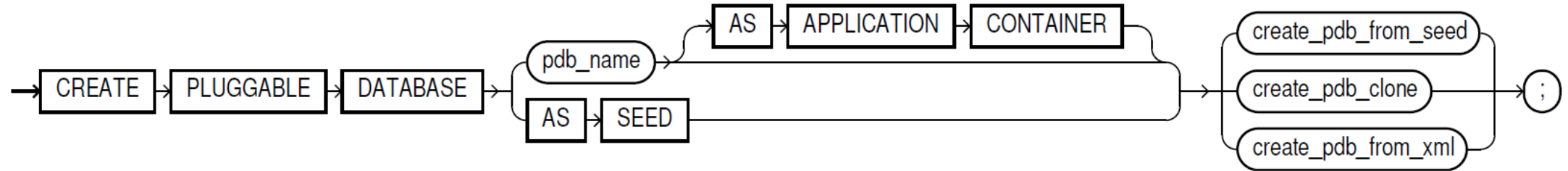
Help < Back Next > Finish Cancel

Arbeiten mit PDBs

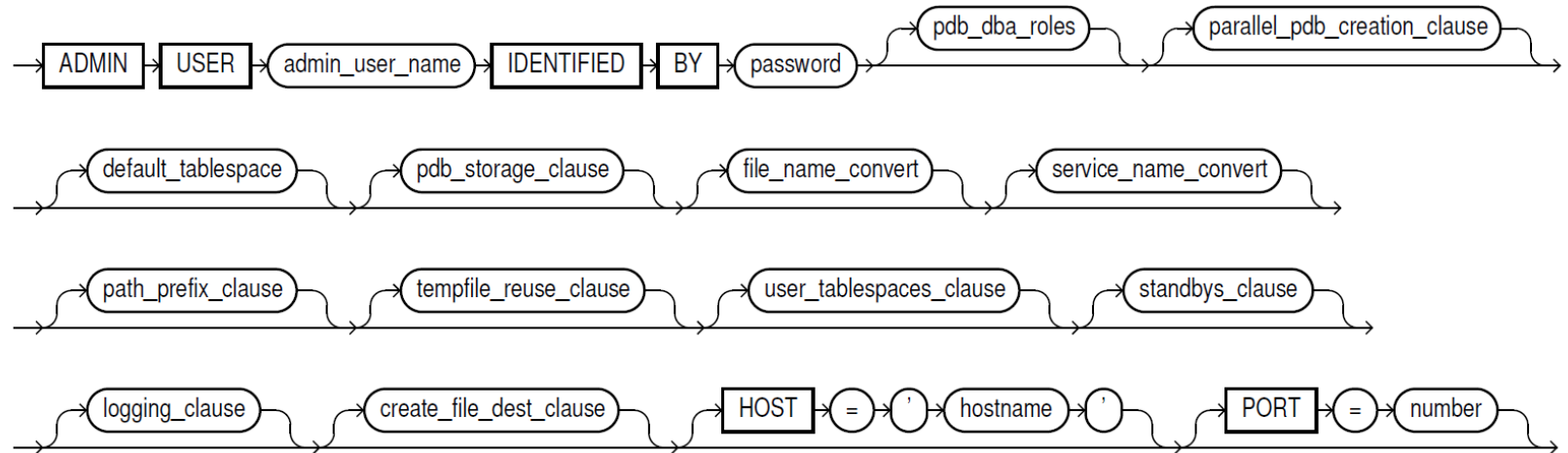
- CONTAINER → Eigenständiger Bereich einer Datenbank (kann PDB oder CDB sein)
Besteht mindestens aus SYSTEM und SYSAUX Tablespace
- CDB → Container Database (CDB\$ROOT)
- PDB → Pluggable Database
- CON_ID → Identifier für jede Datenbank
 - 0 → Gesamtheit aller Datenbanken
 - 1 → CDB
 - 2 → PDB\$SEED
 - >2 → PDB
- GUID → eindeutiger Bezeichner von CDB und PDB

Create Pluggable Database

create_pluggable_database::=



create_pdb_from_seed::=



- from SEED (Default)
 - Template PDB in jeder CDB
 - Ist Read-Only geöffnet und kann nicht (offiziell) geändert werden)
 - Bei OMF nichts weiter zu tun
 - Bei Non-OMF → FILE_NAME_CONVERT Parameter verwenden
- from Clone
 - Aus einer existierenden PDB
 - Identisch zu Seed, nur dass es „meine eigene“ PDB ist
 - Ab 12.2 auch bei geöffneter PDB möglich
 - Clone kann auch remote (CDB, Non-CDB) sein, dann DB-Link
- from XML (Manifest)
 - Datendateien werden kopiert oder angehängt
 - Konfiguration in der XML-Datei gespeichert
 - Z.B. bei NON-CDB nach PDB Migration

Erstellen einer PDB aus Seed

- Erstellen einer PDB aus der Seed-Datenbank:

```
SQL> CREATE PLUGGABLE DATABASE johann1 ADMIN USER joadmin  
      IDENTIFIED BY manager  
      FILE_NAME_CONVERT=('/u02/oradata/CJOHANN/pdbseed',  
                        '/u02/oradata/CJOHANN/johann1');
```

- Einfacher

```
SQL> CREATE PLUGGABLE DATABASE johann1 ADMIN USER joadmin  
      IDENTIFIED BY manager  
      FILE_NAME_CONVERT=('pdbseed','johann1');
```

- Mit Privilegien und OMF

```
SQL> CREATE PLUGGABLE DATABASE johann1 ADMIN USER joadmin  
      IDENTIFIED BY manager ROLES=(connect);
```


- Eine PDB wird beim Starten der Instanz nicht automatisch geöffnet
- Daher immer:

```
SQL> CREATE PLUGGABLE DATABASE <PDBNAME>
```

```
SQL> ALTER PLUGGABLE DATABASE <PDBNAME> OPEN INSTANCES=ALL;
```

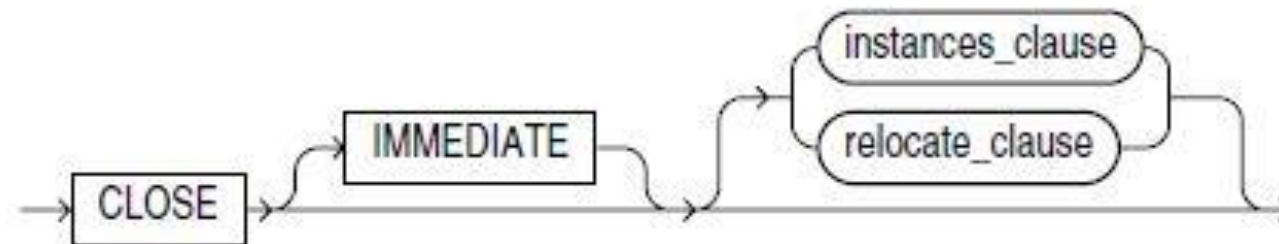
```
SQL> ALTER PLUGGABLE DATABASE <PDBNAME> SAVE STATE;
```

- SAVE STATE: sichert den derzeitigen Status der PDB

- Schließen einer oder mehrerer Pluggable Databases

```
ALTER PLUGGABLE DATABASE [<PDB>,<PDB>,ALL] [EXCEPT <PDB>] CLOSE ...
```

pdb_close ::=



- Es gibt zwar den Befehl „UNPLUG“ aber nicht „PLUG“
- „UNPLUG“ ist nur aus der CDB möglich
- Nur eine nicht geöffnete PDB kann „UNPLUGGED“ werden
- Die Metainformation wird in eine XML-Datei gespeichert (Endung .xml zwingend)

```
ALTER PLUGGABLE DATABASE <PDB> UNPLUG INTO '<filename>.xml';
```

- Eine „UNPLUGGED“ Database wird trotzdem noch mit gesichert

- Information über „UNPLUGGED“ PDB

```
SQL> SELECT pdb_id, pdb_name, guid, status, con_id  
        FROM   cdb_pdb;
```

PDB_ID	PDB_NAME	GUID	STATUS	CON_ID
3	ANTON1	EF89F348D7051BF3E043281E10ACCCD7	NORMAL	1
2	PDB\$SEED	EF899360B02E192DE043281E10ACBA7D	NORMAL	1
4	ANTON2	EF8AC4B702D10788E043281E10ACE619	NORMAL	1
5	ANTON_CLONE1	EFA00E301D553993E043281E10ACB128	UNPLUGGED	1

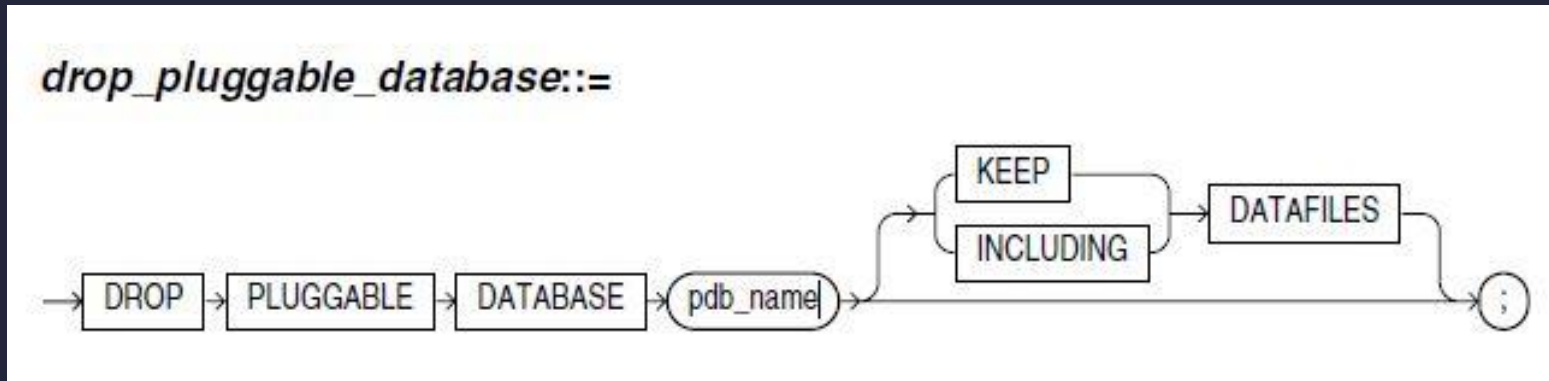
- Informationen über die „unplugged“ PDB bleibt im Controlfile vorhanden

```
SELECT p.pdb_name, t.name, p.status, p.pdb_id
FROM v$tablespace t, cdb_pdb$seeds p
WHERE t.con_id = p.pdb_id ORDER BY 4 NULLS FIRST;
```

PDB_NAME	NAME	STATUS	PDB_ID
-----	-----	-----	-----
PDB\$SEED	SYSAUX	NORMAL	2
PDB\$SEED	SYSTEM	NORMAL	2
PDB\$SEED	TEMP	NORMAL	2
JOHANN1	SYSAUX	NORMAL	3
JOHANN1	TEMP	NORMAL	3
JOHANN1	SYSTEM	NORMAL	3
JOHANN2	SYSTEM	UNPLUGGED	4
JOHANN2	SYSAUX	UNPLUGGED	4
JOHANN2	TEMP	UNPLUGGED	4

Unplug Pluggable Database

- Nach dem „UNPLUG“ nur noch ein „DROP“ möglich



- Default: KEEP DATAFILES

```
DROP PLUGGABLE DATABASE johann2;
```

- Löscht nur den Eintrag der Pluggable Database im Data Dictionary
- Datafiles bleiben bestehen

DROP PLUGGABLE DATABASE

- Datafiles werden nicht mehr mit gesichert
- Nach dem DROP PLUGGABLE DATABASE kann die PDB wieder „eingehängt“ werden (CREATE PLUGGABLE DATABASE)

Views und Tabellen

Anmelden an eine ein PDB

- Indirektes Anmelden über die Container Datenbank:

```
$ sqlplus / as sysdba                # CDB  
SQL> ALTER SESSION SET CONTAINER= ...;
```

- „Normale“ Anmeldung an die PDB nur über entsprechenden Alias oder Easy Connect

```
$ sqlplus username/password@<TNSALIAS>
```

```
$ sqlplus username/password@<hostname>:<port>/<SERVICE_NAME>
```

Service Erstellung für die PDB (1)

- Jede PDB hat automatisch einen eigenen Service, der aber nicht geändert werden kann
- Konfigurieren von Services mit `srvctl` (GI Installation):
 - Spezieller Parameter `-pdb` zum Anlegen bzw. Ändern von Services
 - Geänderte Optionen (z.B. `-db` statt `-d`, `-service` statt `-s`)

```
$ srvctl add service -db ja12c -pdb pdb1_ja12c  
    -service pdb1ja12c  
$ srvctl start service -db ja12c -service pdb1ja12c
```

- Die PDB wird danach automatisch gestartet, wenn der Service gestartet wird

Service Erstellung für die PDB (2)

- Ohne srvctl (d.h. keine GI Installation):
 - dbms_service Package
 - Muss in der entsprechenden PDB ausgeführt werden!

```
SQL> ALTER SESSION SET CONTAINER = pdb1_ja12c;
```

Session wurde geändert.

```
SQL> EXECUTE  
      dbms_service.create_service('PDB1JA12C','PDB1JA12C.DB.DE');
```

PL/SQL-Prozedur erfolgreich abgeschlossen.

```
SQL> EXECUTE dbms_service.start_service('PDB1JA12C');
```

PL/SQL-Prozedur erfolgreich abgeschlossen.

- V\$PDBS → welche PDBs gibt es (außer CDB)

```
SQL> SELECT con_id, name, open_mode  
        FROM v$pdb;
```

CON_ID	NAME	OPEN_MODE
2	PDB\$SEED	READ ONLY
3	ANTON1	READ WRITE
4	ANTON2	MOUNTED
5	ANTON_CLONE1	MOUNTED

- Keine Informationen über „UNPLUGGED“ PDBs

- V\$CONTAINERS → welche PDBs gibt es (inclusive CDB)

```
SQL> SELECT con_id, name, open_mode  
        FROM v$containers;
```

CON_ID	NAME	OPEN_MODE
1	CDB\$ROOT	READ WRITE
2	PDB\$SEED	READ ONLY
3	ANTON1	READ WRITE
4	ANTON2	MOUNTED
5	ANTON_CLONE1	MOUNTED

- Ebenfalls keine Informationen über „UNPLUGGED“ PDBs

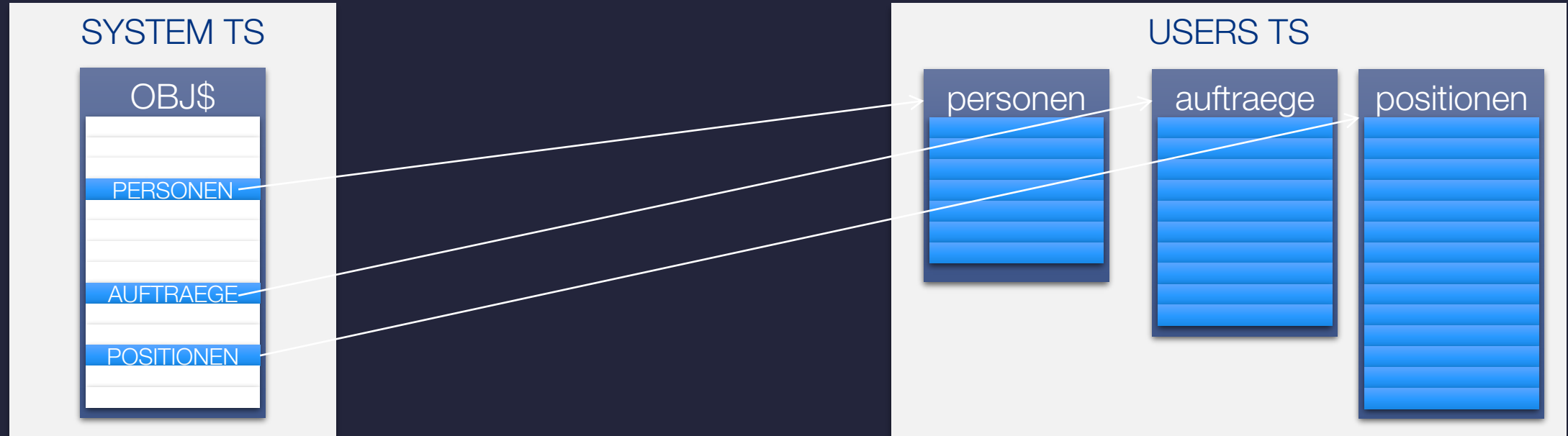
- CDB_PDBS → welche PDBs gibt es
 - VORSICHT: in dieser View ist die CON_ID immer 1!
 - Statt dessen PDB_ID nutzen

```
SQL> SELECT pdb_id, pdb_name, guid, status, con_id  
        FROM   cdb_pdb;
```

PDB_ID	PDB_NAME	GUID	STATUS	CON_ID
2	PDB\$SEED	41039489BBAE5C6BE0550A0027AEAD56	NORMAL	2
3	SUZANNE	410466F3114C1359E0550A0027AEAD56	NORMAL	3
4	ANGEL	410586AE2BE52E2EE0550A0027AEAD56	UNPLUGGED	4

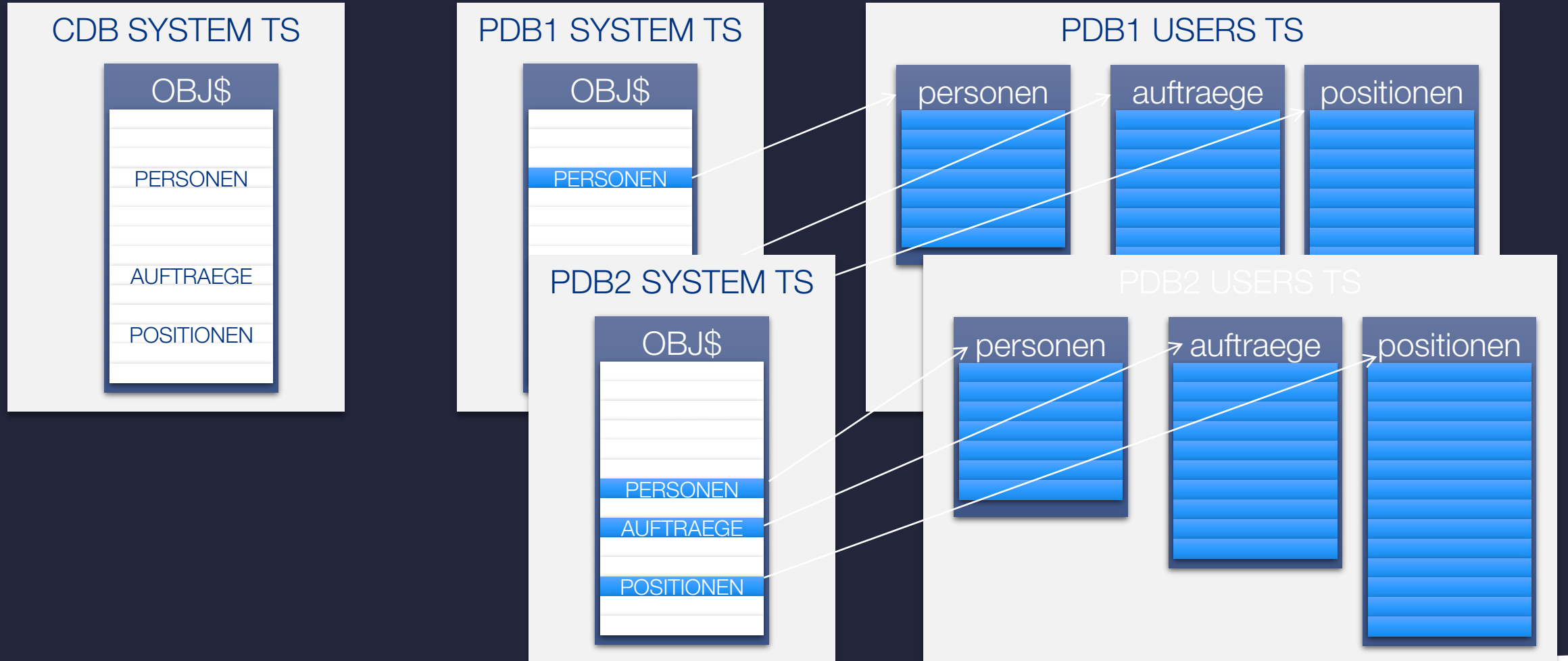
- Liefert Informationen, ob PDB „eingehängt“ oder „unplugged“ ist
- Liefert keine Informationen, ob PDB geöffnet bzw. wie geöffnet ist!

NON-CDB Data Dictionary





PDB Data Dictionary



- Data Dictionary Metadaten liegen in der CDB\$ROOT
 - Z.B. TAB\$
- PDBs haben einen Link in diese Root-Tabellen (Metadata Link)
- Data Dictionary Daten können sowohl im Root als auch in der PDB liegen
 - COMMON USERS → Root
 - Local Users → PDB
 - PDB bekommt die Root-Daten vererbt (Read Only) und hat seine eigenen Daten

- CDB_VIEWS
 - CDB: Informationen über die CDB und alle PDBs
 - PDB: Informationen über die PDB
- DBA_VIEWS
 - CDB: Informationen über die CDB
 - PDB: Informationen über die PDB
- Unterschied CDB_VIEWS zu DBA_VIEWS
 - DBA_VIEWS haben keine CON_ID

CDB_OBJECTS

OWNER	OBJECT_NAME	OBJECT_ID	OBJECT_TYPE	SHARING	CON_ID
SYS	TAB\$	4	TABLE	METADATA LINK	1
SYS	WRM\$_DATABASE_INSTANCE	7778	TABLE	METADATA LINK	1
SYS	DBA_HIST_DATABASE_INSTANCE	7799	VIEW	OBJECT LINK	1
SYS	TAB\$	4	TABLE	METADATA LINK	2
SYS	WRM\$_DATABASE_INSTANCE	7776	TABLE	METADATA LINK	2
SYS	DBA_HIST_DATABASE_INSTANCE	7797	VIEW	OBJECT LINK	2
SYS	TAB\$	4	TABLE	METADATA LINK	3
SYS	WRM\$_DATABASE_INSTANCE	7776	TABLE	METADATA LINK	3
SYS	DBA_HIST_DATABASE_INSTANCE	7797	VIEW	OBJECT LINK	3
DEMO	PERSONEN	19938	TABLE	NONE	3
SYS	TAB\$	4	TABLE	METADATA LINK	4
SYS	WRM\$_DATABASE_INSTANCE	7776	TABLE	METADATA LINK	4
SYS	DBA_HIST_DATABASE_INSTANCE	7797	VIEW	OBJECT LINK	4
DEMO	PERSONEN	19881	TABLE	NONE	4
SYS	TAB\$	4	TABLE	METADATA LINK	5
SYS	WRM\$_DATABASE_INSTANCE	7776	TABLE	METADATA LINK	5
SYS	DBA_HIST_DATABASE_INSTANCE	7797	VIEW	OBJECT LINK	5
PUBLIC	DBA_HIST_DATABASE_INSTANCE	7798	SYNONYM	METADATA LINK	5
DEMO	PERSONEN	19938	TABLE	NONE	5

- Abfrage 1

```
SELECT obj#, owner#, name FROM obj$ WHERE name = 'PERSONEN';
```

- CDB

```
no rows selected
```

- PDB1

OBJ#	OWNER#	NAME
19938	69	PERSONEN

- PDB2

OBJ#	OWNER#	NAME
19881	69	PERSONEN

- Abfrage 2

```
SELECT username, user_id, common FROM dba_users WHERE username IN ('DEMO','SYSTEM');
```

- CDB

USERNAME	USER_ID	COMMON
SYSTEM	8	YES

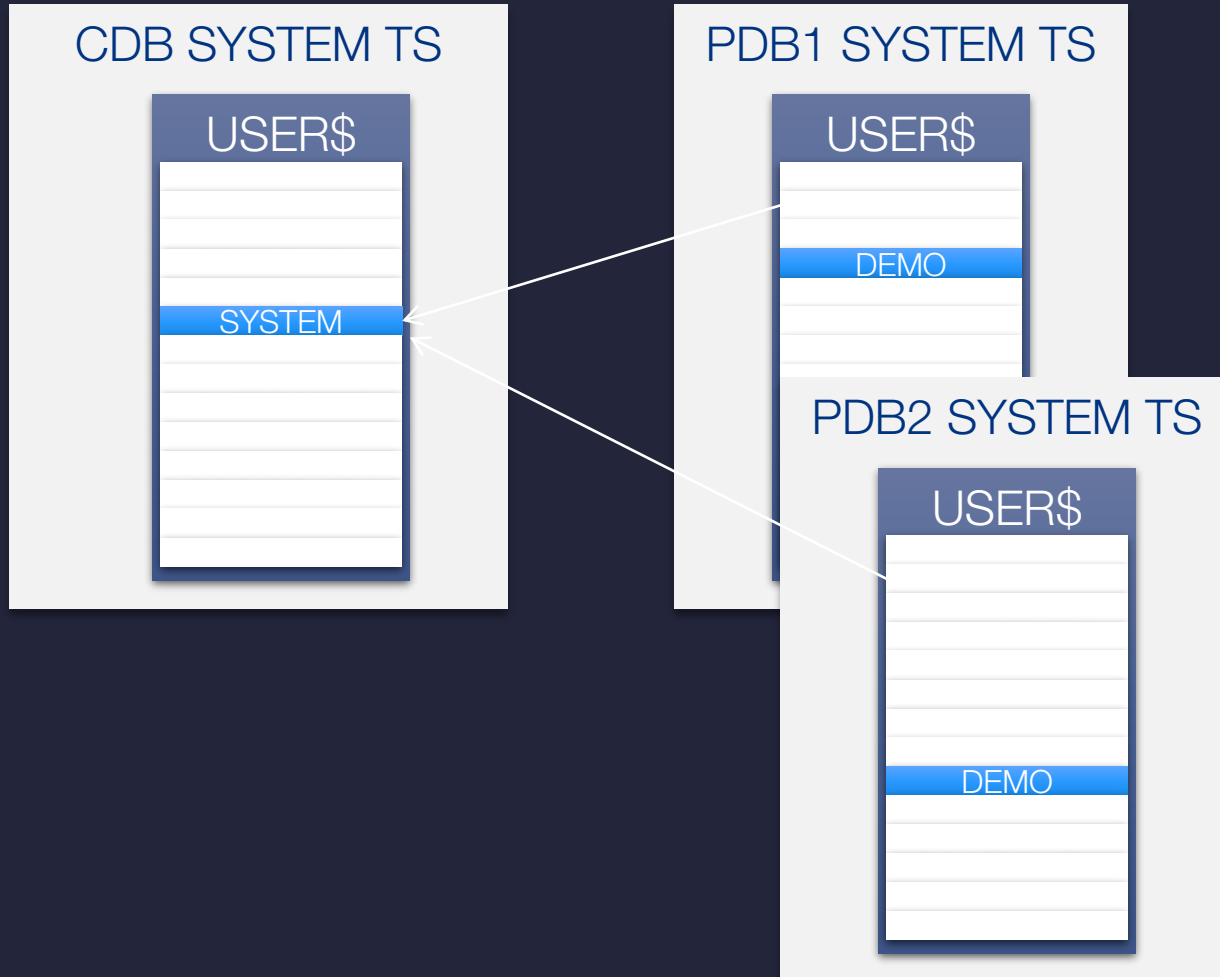
- PDB1

USERNAME	USER_ID	COMMON
SYSTEM	8	YES
DEMO	69	NO

- PDB2

USERNAME	USER_ID	COMMON
SYSTEM	8	YES
DEMO	69	NO

„COMMON“ Users



- Common User
 - Alle DBAs als Common User angelegt
 - Teilweise in PDBs als expired angezeigt – Login trotzdem möglich
 - Wichtig: CONTAINER_DATA=ALL
- Common Roles
 - Funktionieren gut
 - Nicht vergessen: CONTAINER=ALL (!)
- Common Profiles
 - Funktionieren gut
 - DEFAULT Profile muss in jeder PDB angepasst werden!

- Eigenständige Applikations-DBAs
- Mandantenfähigkeit
- Ressourcen besser genutzt (vor allem in den Testsystemen)

- Abfrage 3

```
SELECT username, user_id, common FROM cdb_users WHERE username IN ('DEMO','SYSTEM');
```

- CDB

USERNAME	USER_ID	COMMON
-----	-----	-----
SYSTEM	8	YES
DEMO	66	NO
SYSTEM	8	YES
SYSTEM	8	YES
DEMO	69	NO
SYSTEM	8	YES
SYSTEM	8	YES
DEMO	69	NO

- Abfrage 4

```
SELECT owner, object_name, object_id, con_id FROM cdb_objects  
WHERE object_name = 'PERSONEN';
```

- CDB

OWNER	OBJECT_NAME	OBJECT_ID	CON_ID
DEMO	PERSONEN	91897	3
DEMO	PERSONEN	92039	4

- PDB1

OWNER	OBJECT_NAME	OBJECT_ID	CON_ID
DEMO	PERSONEN	91897	3

- PDB2

OWNER	OBJECT_NAME	OBJECT_ID	CON_ID
DEMO	PERSONEN	92039	4

- SELECT COUNT(*) FROM

Tabelle	CDB	PLUGJA21	PLUGJA22	PJA12CD1	PDB\$SEED
USER\$	121	123	122	122	121
TAB\$	2362	2364	2363	2363	2362
IND\$	4240	4244	4244	4244	4240
OBJ\$	91372	90828	90803	91157	90773
COL\$	106677	106694	106692	106692	106678
CDB_USERS	179	37	36	36	35
CDB_TABLES	11579	2317	2316	2316	2315
CDB_INDEXES	21212	4244	4244	4244	4240
CDB_OBJECTS	454953	90874	90801	91155	90693
CDB_TAB_COLUMNS	521624	104332	104330	104330	104316

- Nachinstallation einer Funktion, z.B. Context (catctx.sql)
 - Keine Verwendung von SQL*Plus möglich
 - Statt dessen perl und catcon.pl

```
$ORACLE_HOME/perl/bin/perl -u SYS -n 8 -d $ORACLE_HOME/ctx/admin ... catctx.sql
```

- Sorgt dafür, dass das Package in allen PDBs ausgeführt wird (hier parallel 8)

- Zitat:
„The PDB name must be unique in the CDB, and it must be unique within the scope of all the CDBs whose instances are reached through a specific listener.“
- Erstellen von NetApp Snapshots
 - Snapshot PDBs haben den gleichen Namen wie die originalen
 - Standard Services können nicht zugeordnet werden
- IMMER MIT ANWENDUNGSSPEZIFISCHEN SERVICES ARBEITEN!

Example 19-7 Showing the Data Files for Each PDB in a CDB

This example queries the `DBA_PDBS` and `CDB_DATA_FILES` views to show the name and location of each data file for all of the PDBs in a CDB, including the PDB seed.

```
COLUMN PDB_ID FORMAT 999
COLUMN PDB_NAME FORMAT A8
COLUMN FILE_ID FORMAT 9999
COLUMN TABLESPACE_NAME FORMAT A10
COLUMN FILE_NAME FORMAT A45

SELECT p.PDB_ID, p.PDB_NAME, d.FILE_ID, d.TABLESPACE_NAME, d.FILE_NAME
FROM DBA_PDBS p, CDB_DATA_FILES d
WHERE p.PDB_ID = d.CON_ID
ORDER BY p.PDB_ID;
```

Sample output:

PDB_ID	PDB_NAME	FILE_ID	TABLESPACE_NAME	FILE_NAME
2	PDB\$SEED	6	SYSAUX	/disk1/oracle/dbs/pdbseed/cdbl_ax.f
2	PDB\$SEED	5	SYSTEM	/disk1/oracle/dbs/pdbseed/cdbl_db.f
3	HRPDB	9	SYSAUX	/disk1/oracle/dbs/hrpdb/hrpdb_ax.f
3	HRPDB	8	SYSTEM	/disk1/oracle/dbs/hrpdb/hrpdb_db.f
3	HRPDB	13	USER	/disk1/oracle/dbs/hrpdb/hrpdb_usr.dbf
4	SALESPDB	15	SYSTEM	/disk1/oracle/dbs/salespdb/salespdb_db.f
4	SALESPDB	16	SYSAUX	/disk1/oracle/dbs/salespdb/salespdb_ax.f
4	SALESPDB	18	USER	/disk1/oracle/dbs/salespdb/salespdb_usr.dbf

Oracle Database Oracle® Multitenant Administrator's Guide 19c E96136-04 June 2019

- Default: PDB\$SEED wird bei allen CDB-Views ignoriert
- Laut Oracle muss das so sein (warum auch immer)
- Ausschalten mit:
 - 12.1: `exclude_seed_cdb_view = FALSE`
 - 12.2: `_exclude_seed_cdb_view = FALSE`
- Setting this parameter to `FALSE` would return results for the seed database when querying against the CDB views


```
ALTER SYSTEM SET "_exclude_seed_cdb_view"=FALSE;
SELECT con_id, tablespace_name,
       trunc(sum(bytes)/1024/1024) Mbytes,
       trunc(sum(decode(maxbytes,0,bytes,maxbytes)/1024/1024)) maxmbytes
FROM   cdb_data_files
GROUP  BY con_id, tablespace_name
ORDER  BY con_id;
```

CON_ID	TABLESPACE_NAME	MBYTES	MAXMBYTES
1	USERS	5	32767
1	UNDOTBS1	920	32767
1	SYSTEM	800	32767
1	SYSAUX	870	32767
2	SYSTEM	250	32767
2	SYSAUX	550	32767
3	USERS	100	5120
3	SYSAUX	600	32767
3	SYSTEM	260	32767
4	USERS	300	1024
4	SYSTEM	260	32767
4	SYSAUX	600	32767

- Connect mit ORACLE_SID
- Kein Connect an die PDB möglich

Logon

Database

User ID

Password

Logon Exit

Configuration Name

Database Type Oracle

Host name or IP <HOST>

Database Port 1521

Database Name <SID>

Database Alias

☒ Operator Mode Table Owner

Test Save Remove Import/Exp... Help

- Listener Parameter

```
USE_SID_AS_SERVICE_listener=ON
```

- Der „SID“ Parameter in der Connect Beschreibung wird als Servicename interpretiert

Patches und Upgrades



New Paradigms for Rapid Patching and Upgrades

The investment of time and effort to patch one multitenant container database results in patching all of its many pluggable databases. To patch a single pluggable database, you simply unplug/plug to a multitenant container database at a different Oracle Database software version.

To upgrade all hosted pluggable databases in a container database, simply upgrade the container database and all hosted pluggable databases are upgraded 'in-place'. To upgrade a single PDB, you simply unplug/plug the pluggable database in to a container database at a higher version and upgrade the pluggable database as described in the [Database Upgrade Guide](#).

<http://www.oracle.com/technetwork/database/multitenant/overview/index.html>

ALTERNATIVE FAKTEN

- Die Investition von Zeit und Aufwand für das Patchen einer Multitenant-Containerdatenbank führt dazu, dass all die vielen zugehörigen plug-fähigen Datenbanken gepatcht werden. Um eine einzelne plug-fähige Datenbank zu patchen, trennen Sie einfach mit einer anderen Oracle Database-Softwareversion die Verbindung zu einer Multitenant-Containerdatenbank oder stellen sie her.

Quelle: <https://www.oracle.com/de/database/technologies/multitenant.html>

- Optimizer Fixes werden über das Package `dbms_optim_bundle` aktiviert
 - `GETBUGSFORBUNDLE` → Zeigt die Patchids an, die damit gefixt sind
 - `ENABLE_OPTIM_FIXES` → Aktiviert die Fixes
- Funktioniert auch für Oracle 12.1

```
SQL> execute dbms_optim_bundle.enable_optim_fixes('ON','BOTH','YES');
```

- Funktioniert nicht für Standby Datenbank!
 - Entweder nach Switchover / Failover explizit ausführen
 - Besser:

```
SQL> ALTER SYSTEM SET _fix_control='patch_nr1','patch_nr2',...
```

PDB Parameter

- Oracle 12.1 nur sehr eingeschränkt
- Oracle 12.2 für SGA Parameter und Lockdown Profiles

- CDB Resource Plan
 - CPU pro PDB (Shares, Limit)
 - Memory pro PDB (Shares, Limit, Minimum)
 - Parallelisierungsgrad (Shares, Limit)
- PDB Parameter
 - IOPS (I/O per Second)
 - MBPS (MB per Second)

- Jede PDB eigene SGA
 - SGA_TARGET
 - DB_CACHE_SIZE
 - SHARED_POOL_SIZE
 - PGA_AGGREGATE_TARGET
 - PGA_AGGREGATE_LIMIT

```
SQL> ALTER SESSION SET CONTAINER=suzanne;
```

```
SQL> show parameter sga
```

NAME	TYPE	VALUE
sga_max_size	big integer	2512M
sga_min_size	big integer	0
sga_target	big integer	0

```
SQL> ALTER SYSTEM SET sga_target=500M;
```

```
SQL> show parameter sga
```

NAME	TYPE	VALUE
sga_max_size	big integer	2512M
sga_min_size	big integer	0
sga_target	big integer	500M

```
SQL> ALTER SESSION SET CONTAINER=CDB$ROOT;
```

```
SQL> show parameter sga
```

NAME	TYPE	VALUE
sga_max_size	big integer	2512M
sga_min_size	big integer	0
sga_target	big integer	2512M

- Einschränkung von Datenbank Optionen
 - Partitioning
 - Database Queuing (warum auch immer)
- Datenbank Features
 - AWR
 - OS_ACCESS
 - usw.
- SQL Befehle
 - ALTER DATABASE
 - ALTER PLUGGABLE DATABASE
 - ALTER SYSTEM
 - ALTER SESSION

Flashback

- Erstellen Restore Point

```
SQL> ALTER SESSION SET CONTAINER=konstantin;  
SQL> CREATE RESTORE POINT insert30 GUARANTEE FLASHBACK DATABASE;
```

- Flashback PDB

```
SQL> ALTER PLUGGABLE DATABASE CLOSE IMMEDIATE;  
SQL> FLASHBACK PLUGGABLE DATABASE TO RESTORE POINT insert30;  
SQL> ALTER PLUGGABLE DATABASE OPEN RESETLOGS;
```

- Standby DB

```
(3):Recovery of pluggable database KONSTANTIN aborted due to pluggable database open  
resetlog marker.  
(3):To continue recovery, restore all data files for this PDB to checkpoint SCN lower  
than 948798, or timestamp before 09/08/2018 14:55:32, and restart recovery
```

- DOAG Konferenz und Ausstellung 19 bis 23. November
 - 19.11 14:00 Uhr Das Battle: On Premises vs. Cloud
 - 19.11 15:00 Uhr Multitenant: ein Review
 - 20.11 11:00 Uhr Oracle 18 XE „Mehr als eine Spieldatenbank“
 - 20.11 13:00 Uhr Geliebter Feind „Der DBA und DevOps“
 - 21.11 12:00 Uhr Datenbank-Monitoring für das nächste Jahrzehnt
 - 21.11 13:00 Uhr Alternativen zur Enterprise Edition der Oracle DB
 - 22.11 ganztägig PostgreSQL für Oracle DBAs
- Frankfurter IT Tage 9. bis 12. Dezember in Frankfurt
 - 12.12 09:00 Uhr Alternativen zur Oracle Enterprise Edition
- DOAG Noon2Noon „Oracle auf VMware“ 23. bis 24 Januar 2020 München

- Experten mit über 30 Jahren Datenbank Erfahrung
- Spezialisten für
 - Datenbank Administration (Oracle und PostgreSQL)
 - Hochverfügbarkeit (RAC, Data Guard, Replication, etc.)
 - Migrationen (Unicode, PostgreSQL)
 - Performance Optimierung
 - Monitoring (OEM, Foglight, CheckMK, PEM)
- Fernwartung
- Schulung und Workshops
 - PostgreSQL
 - Oracle
 - Toad



- E-Mail: johannes.ahrends@carajandb.com
- Homepage: www.carajandb.com
- Adresse:
 - CarajanDB GmbH
Siemensstraße 25
50374 Erftstadt
- Telefon:
 - +49 (22 35) 1 70 91 84
 - +49 (1 70) 4 05 69 36
- Twitter: carajandb
- Facebook: johannes.ahrends
- Blogs:
 - blog.carajandb.com
 - www.toadworld.com