



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)





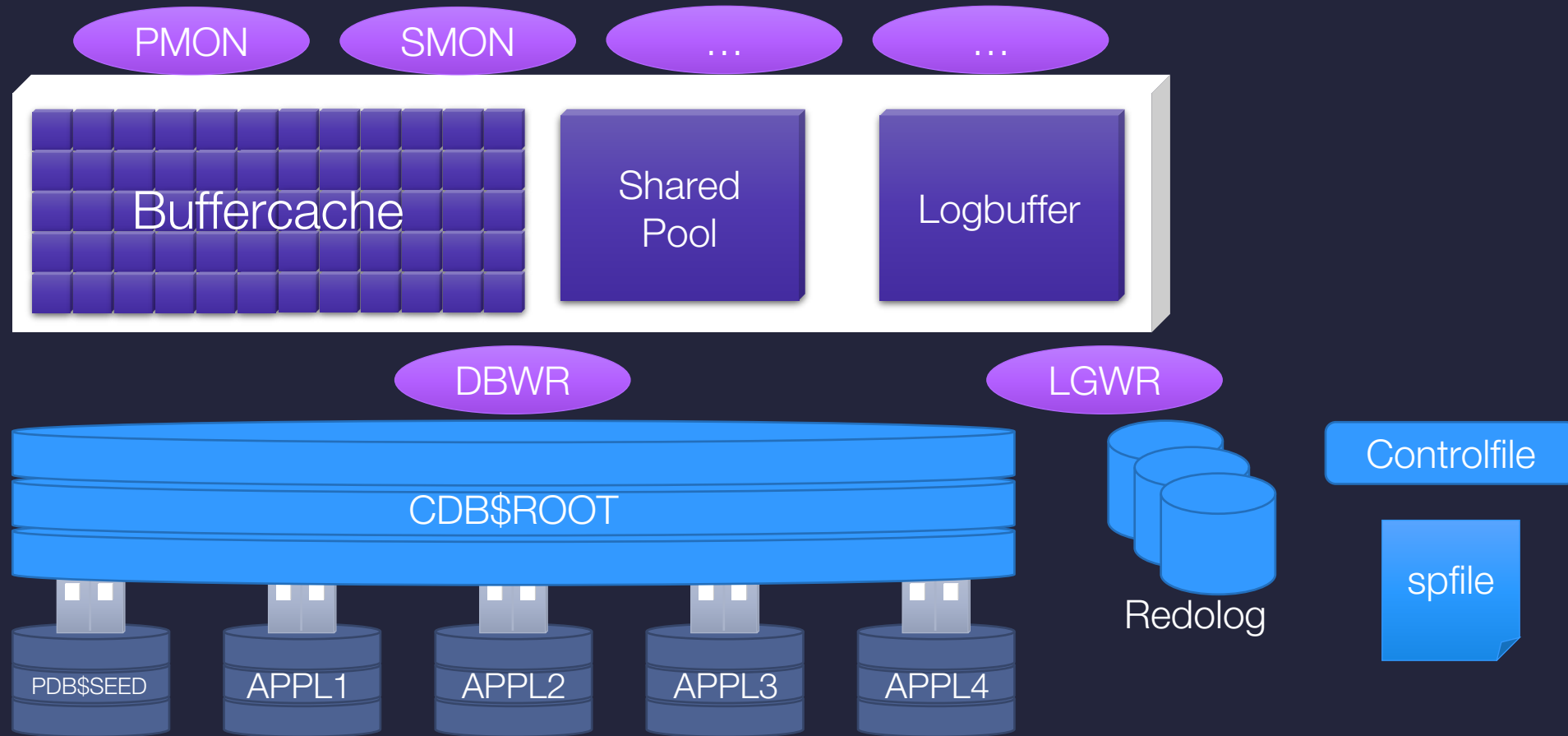
Multitenant Database

- 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



Root Container (CDB)

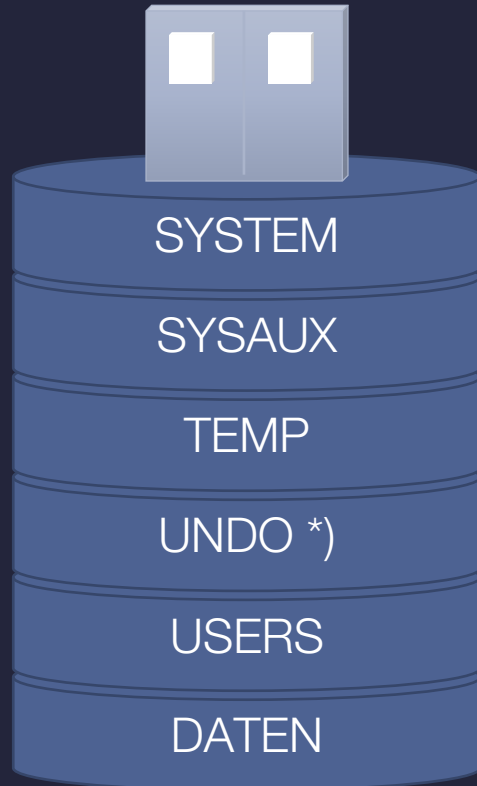


Controlfile

spfile

- Komponenten für den Aufbau der Instanz
 - Controlfiles
 - spfile
 - Connect an die Instanz nur über die CDB möglich
- Verwaltung der Redolog-Dateien
 - Vollständiges Recovery der Datenbank nur über die CDB möglich

Pluggable Database (PDB)



- In sich geschlossenes System
 - Data Dictionary
 - Tablespaces
 - Schemata
 - Objekte
 - Public Objekte
 - Workload Repository
 - Lokales Undo (nicht immer)
 - Parameter

- 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

Warum Pluggable Database?

- Konsolidierung von Datenbanken
- Eine gemeinsame Instanz
- Ein gemeinsames Redo Management
- Separate Tablespace Verwaltung
- Separate Schema Verwaltung
- Separate User / Rollen-Verwaltung

- 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

```
% cat /u01/app/oracle/admin/UDO_S1/scripts/init.ora
...
#####
# Database Identification
#####
db_name="UDO"
db_unique_name="UDO_S1"
#####
# File Configuration
#####
db_create_file_dest="/u02/oradata"
db_recovery_file_dest="/u03/fast_recovery_area/"
db_recovery_file_dest_size=20000m
#####
# Miscellaneous
#####
compatible=19.0.0
diagnostic_dest=/u01/app/oracle
enable_pluggable_database=true
...
```



```
% cat /u01/app/oracle/admin/UDO_S1/scripts/createDB.sql
...
CREATE DATABASE "UDO"
...
DATAFILE SIZE 700M AUTOEXTEND ON NEXT 100M MAXSIZE 2000M EXTENT MANAGEMENT LOCAL
SYSAUX DATAFILE SIZE 500M AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
DEFAULT TEMPORARY TABLESPACE TEMP TEMPFILE SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
UNDO TABLESPACE "UNDOTBS1" DATAFILE SIZE 200M AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
DEFAULT TABLESPACE "USERS" LOGGING DATAFILE SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
CHARACTER SET AL32UTF8
NATIONAL CHARACTER SET AL16UTF16
LOGFILE GROUP 1 SIZE 200M,
GROUP 2 SIZE 200M,
GROUP 3 SIZE 200M
USER SYS IDENTIFIED BY "&&sysPassword" USER SYSTEM IDENTIFIED BY "&&systemPassword"
enable pluggable database
LOCAL UNDO ON;
```

Create Database Skript - NONOMF

```
% cat /u01/app/oracle/admin/FRANZ_S1/scripts/createDB.sql
...
CREATE DATABASE "FRANZ"
...
DATAFILE '/u02/oradata/FRANZ/system01.dbf' SIZE 700M REUSE AUTOEXTEND ON NEXT 100M MAXSIZE
2000M EXTENT MANAGEMENT LOCAL
...
SMALLFILE TABLESPACE "USERS" LOGGING DATAFILE '/u02/oradata/FRANZ/users01.dbf' SIZE 100M
REUSE AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
...
enable pluggable database
seed
file_name_convert=('/u02/oradata/FRANZ/system01.dbf', '/u02/oradata/FRANZ/pdbseed/system01.db
f', '/u02/oradata/FRANZ/sysaux01.dbf', '/u02/oradata/FRANZ/pdbseed/sysaux01.dbf', '/u02/oradata
/FRANZ/temp01.dbf', '/u02/oradata/FRANZ/pdbseed/temp01.dbf', '/u02/oradata/FRANZ/undotbs01.dbf
', '/u02/oradata/FRANZ/pdbseed/undotbs01.dbf')
LOCAL UNDO ON;
spool off
```

- Besser:

```
% cat /u01/app/oracle/admin/FRANZ_S1/scripts/createDB.sql
...
CREATE DATABASE "FRANZ"
...
DATAFILE '/u02/oradata/FRANZ/system01.dbf' SIZE 700M REUSE AUTOEXTEND ON NEXT 100M MAXSIZE
2000M EXTENT MANAGEMENT LOCAL
...
SMALLFILE TABLESPACE "USERS" LOGGING DATAFILE '/u02/oradata/FRANZ/users01.dbf' SIZE 100M
REUSE AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
...
enable pluggable database
seed seed file_name_convert= ('FRANZ', 'FRANZ/pdbseed')
LOCAL UNDO ON;
spool off
```

Arbeiten mit PDBs

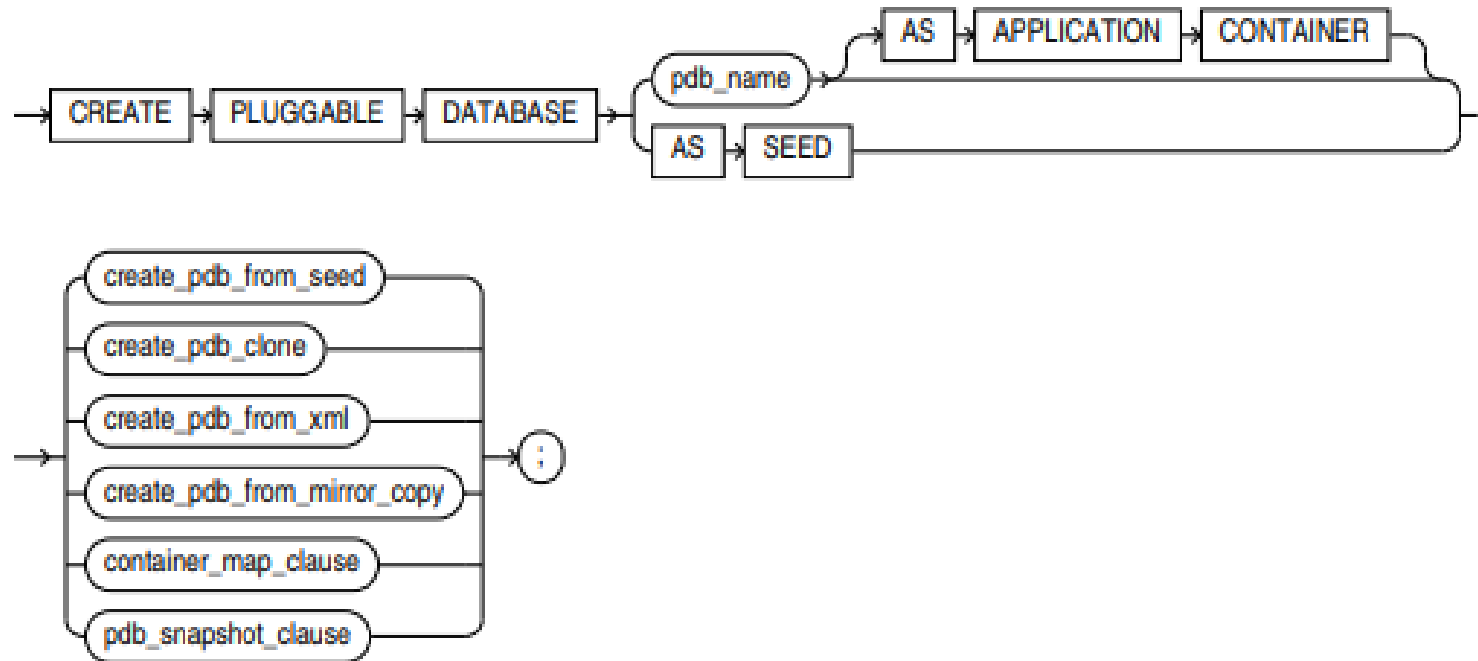
- **CONTAINER** → Eigenständiger Bereich einer Datenbank
(kann PDB oder CDB sein)
Besteht mindestens aus SYSTEM und SYSAUX Tablespace
- **CDB** → Container Database (**CDB\$ROOT**)
- **PDB** → Pluggable Database
- **CON_ID** → Identifier für jede Datenbank
 - 0 → Gesamtheit aller Datenbanken
 - 1 → **CDB**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

- Eine PDB kann nicht erstellt werden!
 - PDBs können ausschließlich kopiert / geklont werden
 - Ausnahme: Migration NON-CDB → PDB
- Den Befehl „Plugin“ gibt es nicht!
 - Eine PDB kann nur mit dem Befehl „CREATE“ in einer CDB angelegt werden
- Den Befehl „Unplug“ gibt es!
 - Mit dem Befehl wird die PDB aus der CDB herausgenommen
 - Der Befehl ist irreversibel

Create Pluggable Database

Syntax

create_pluggable_database::=



- Pluggable Database Template
- existiert immer
- kann nicht / eingeschränkt geändert werden
- Read Only geöffnet
- Für Patching read-write geöffnet
 - `ALTER SYSTEM SET "_ORACLE_SCRT"=TRUE;`
- Mit Oracle 19c kann die PDB\$SEED einfach schreibend geöffnet werden!
 - Bug oder Feature?

- Enthält ein Basis-Data Dictionary
- Enthält die notwendigen Tablespaces
 - SYSTEM
 - SYSAUX
 - UNDO
- Kann ein Default Tablespace enthalten
 - Muss beim Erstellen der CDB mit angelegt worden sein

Create PDB from PDB\$SEED - OMF

```
SQL> CREATE PLUGGABLE DATABASE cello  
      ADMIN USER pdb_admin IDENTIFIED BY manager;
```

- Admin User muss keine DBA Berechtigung haben
- Bei Oracle Managed Files keine weiteren Angaben notwendig

```
SQL> SELECT con_id, tablespace_name, file_name FROM cdb_data_files ORDER BY 1,2;
```

CON_ID	TABLESPACE_NAME	FILE_NAME
1	SYSAUX	/u02/oradata/UDO_S1/datafile/o1_mf_sysaux_h522h356_.dbf
1	SYSTEM	/u02/oradata/UDO_S1/datafile/o1_mf_system_h522gqm_.dbf
1	UNDOTBS1	/u02/oradata/UDO_S1/datafile/o1_mf_undotbs1_h522h7yv_.dbf
1	USERS	/u02/oradata/UDO_S1/datafile/o1_mf_users_h522h9fb_.dbf
2	SYSAUX	/u02/oradata/UDO_S1/9F290C149A1D0795E055FB909B80DF97/datafile/o1_mf_sysaux_h522h4rb_.dbf
2	SYSTEM	/u02/oradata/UDO_S1/9F290C149A1D0795E055FB909B80DF97/datafile/o1_mf_system_h522gtfk_.dbf
2	UNDOTBS1	/u02/oradata/UDO_S1/9F290C149A1D0795E055FB909B80DF97/datafile/o1_mf_undotbs1_h522h8w7_.dbf
2	USERS	/u02/oradata/UDO_S1/9F290C149A1D0795E055FB909B80DF97/datafile/o1_mf_users_h522h9x4_.dbf
3	SYSAUX	/u02/oradata/UDO_S1/9F2C4A9C6B704912E055FB909B80DF97/datafile/o1_mf_sysaux_h52j25f3_.dbf
3	SYSTEM	/u02/oradata/UDO_S1/9F2C4A9C6B704912E055FB909B80DF97/datafile/o1_mf_system_h52j25dc_.dbf
3	UNDOTBS1	/u02/oradata/UDO_S1/9F2C4A9C6B704912E055FB909B80DF97/datafile/o1_mf_undotbs1_h52j25f4_.dbf
3	USERS	/u02/oradata/UDO_S1/9F2C4A9C6B704912E055FB909B80DF97/datafile/o1_mf_users_h52j25f5_.dbf

- Schwierig, wenn PDBs und CDBs unterschiedliche Filesysteme bekommen sollen

Create PDB from PDB\$SEED - NONOMF

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

- Admin User muss keine DBA Berechtigung haben
- Bei Oracle Managed Files keine weiteren Angaben notwendig

Datafile Layout NON-OMF

```
SQL> SELECT con_id, tablespace_name, file_name FROM cdb_data_files ORDER BY 1,2;
```

CON_ID	TABLESPACE_NAME	FILE_NAME
1	SYSAUX	/u02/oradata/FRANZ/sysaux01.dbf
1	SYSTEM	/u02/oradata/FRANZ/system01.dbf
1	UNDOTBS1	/u02/oradata/FRANZ/undotbs01.dbf
1	USERS	/u02/oradata/FRANZ/users01.dbf
2	SYSAUX	/u02/oradata/FRANZ/pdbseed/sysaux01.dbf
2	SYSTEM	/u02/oradata/FRANZ/pdbseed/system01.dbf
2	UNDOTBS1	/u02/oradata/FRANZ/pdbseed/undotbs01.dbf
2	USERS	/u02/oradata/FRANZ/pdbseed/users01.dbf
4	SYSAUX	/u02/oradata/FRANZ/horsti/sysaux01.dbf
4	SYSTEM	/u02/oradata/FRANZ/horsti/system01.dbf
4	UNDOTBS1	/u02/oradata/FRANZ/horsti/undotbs01.dbf
4	USERS	/u02/oradata/FRANZ/horsti/users01.dbf

FALSCH

Datafile Layout NON-OMF

```
SQL> SELECT con_id, tablespace_name, file_name FROM cdb_data_files ORDER BY 1,2;
```

CON_ID	TABLESPACE_NAME	FILE_NAME
1	SYSAUX	/u02/oradata/FRANZ/sysaux01.dbf
1	SYSTEM	/u02/oradata/FRANZ/system01.dbf
1	UNDOTBS1	/u02/oradata/FRANZ/undotbs01.dbf
1	USERS	/u02/oradata/FRANZ/users01.dbf
4	SYSAUX	/u02/oradata/FRANZ/horsti/sysaux01.dbf
4	SYSTEM	/u02/oradata/FRANZ/horsti/system01.dbf
4	UNDOTBS	/u02/oradata/FRANZ/horsti/undotbs01.dbf
4	USERS	/u02/oradata/FRANZ/horsti/users01.dbf

RICHTIG / Falsch

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	FILE_NAME
2	PDB\$SEED	6	SYSTEM	/disk1/oracle/dbs/pdbseed/cdbl_ax.f
2	PDB\$SEED	7	SYSTEM	/disk1/oracle/dbs/pdbseed/cdbl_db.f
3	HRPDB	9	SYS_AUX	/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	SYS_AUX	/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

```
SQL> ALTER SYSTEM SET "_exclude_seed_cdb_view"=FALSE;
```

```
SQL> SELECT con_id, tablespace_name, file_name FROM cdb_data_files ORDER BY 1,2;
```

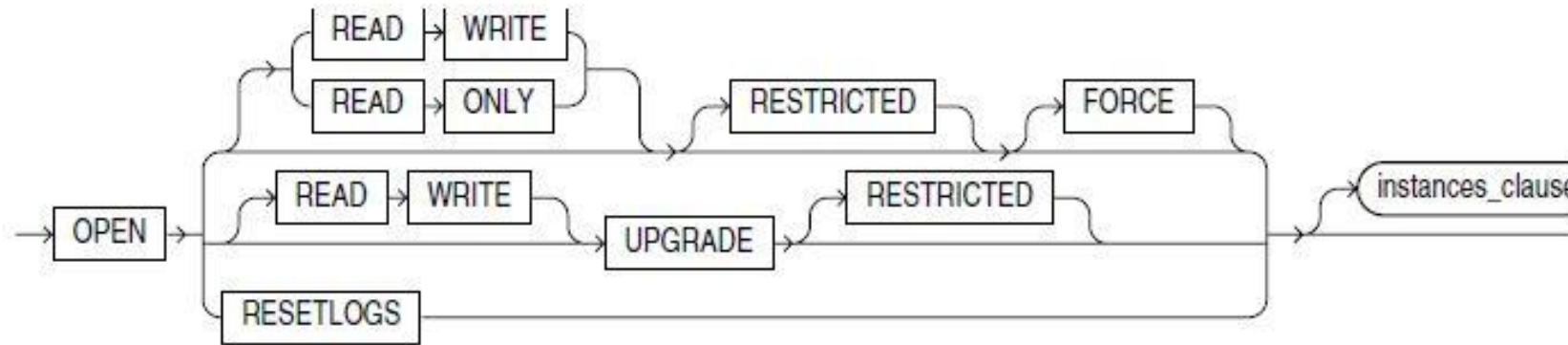
CON_ID	TABLESPACE_NAME	FILE_NAME
1	SYSAUX	/u02/oradata/FRANZ/sysaux01.dbf
1	SYSTEM	/u02/oradata/FRANZ/system01.dbf
1	UNDOTBS1	/u02/oradata/FRANZ/undotbs01.dbf
1	USERS	/u02/oradata/FRANZ/users01.dbf
2	SYSAUX	/u02/oradata/FRANZ/pdbseed/sysaux01.dbf
2	SYSTEM	/u02/oradata/FRANZ/pdbseed/system01.dbf
2	UNDOTBS1	/u02/oradata/FRANZ/pdbseed/undotbs01.dbf
2	USERS	/u02/oradata/FRANZ/pdbseed/users01.dbf
4	SYSAUX	/u02/oradata/FRANZ/horsti/sysaux01.dbf
4	SYSTEM	/u02/oradata/FRANZ/horsti/system01.dbf
4	UNDOTBS1	/u02/oradata/FRANZ/horsti/undotbs01.dbf
4	USERS	/u02/oradata/FRANZ/horsti/users01.dbf

- CDB_-Views ignorieren die SEED PDB
- Mike Dietrich
(https://blogs.oracle.com/UPGRADE/entry/new_undocumented_parameters_in_oracle):
 - `_EXCLUDE_SEED_CDB_VIEW`
 - Undocumented
 - Per feedback by the Multitenant team:
 - Default: TRUE
 - Setting this parameter to FALSE would return results for the seed database when querying against the CDB views

- Öffnen einer oder mehrerer Pluggable Databases

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

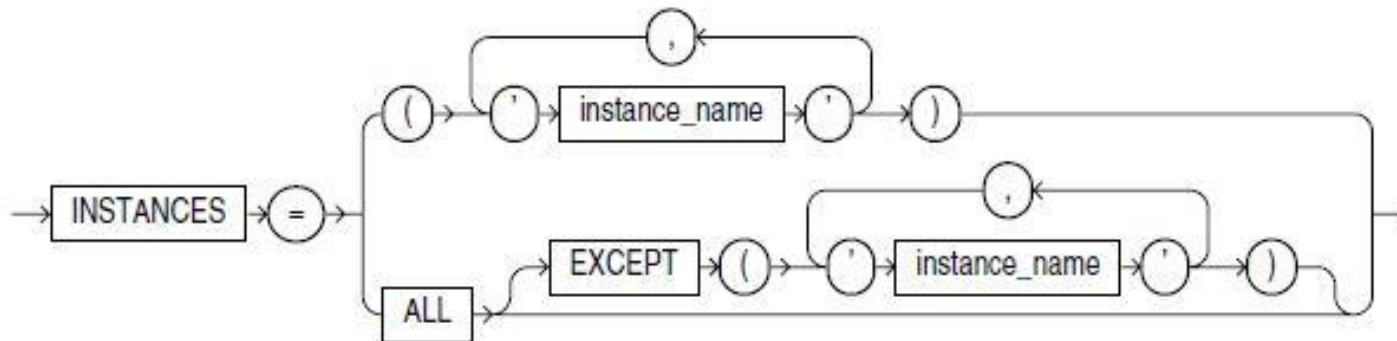
pdb_open::=



- Beim ersten Öffnen wird die Verbindung zum Data Dictionary der CDB hergestellt
→ dauert länger!

- Bei RAC Datenbanken
 - Angabe von Instanzen
 - ... außer

instances_clause::=



PDB öffnen – aus der PDB heraus

- <PLUGGABLE> kann weggelassen werden

```
ALTER PLUGGABLE DATABASE OPEN ...
```

- oder

```
ALTER DATABASE OPEN ...
```

- Öffnen einer einzelnen PDB

```
SQL> ALTER PLUGGABLE DATABASE cello OPEN INSTANCES=ALL;
```

- Öffnen aller PDBs

```
SQL> ALTER PLUGGABLE DATABASE all OPEN INSTANCES=ALL;
```

- Schließen und Öffnen einer PDBs READ ONLY

```
SQL> ALTER PLUGGABLE DATABASE horsti OPEN READ ONLY FORCE INSTANCES=ALL;
```

- Ab 12.1.0.2 → derzeitiger Status der PDB über Instanz-Restart sichern

```
SQL> ALTER PLUGGABLE DATABASE [<PDB>,<PDB>,ALL] [EXCEPT <PDB>] SAVE STATE INSTANCES=...
```

- Rückgängig machen mit

```
SQL> ALTER PLUGGABLE DATABASE [<PDB>,<PDB>,ALL] [EXCEPT <PDB>] DISCARD STATE
```

- Abfrage mit:

```
SQL> SELECT con_id, con_name, instance_name, state, restricted  
       FROM cdb_pdb_saved_states;
```

CON_ID	CON_NAME	INSTANCE_NAME	STATE	RES
3	MICHELLE1	JOHN1	OPEN	NO
3	MICHELLE1	JOHN2	OPEN	NO

- Aus der CDB nicht möglich!
 - SHUTDOWN PLUGGABLE DATABASE gibt es nicht
- Aus der Pluggable Database heraus:

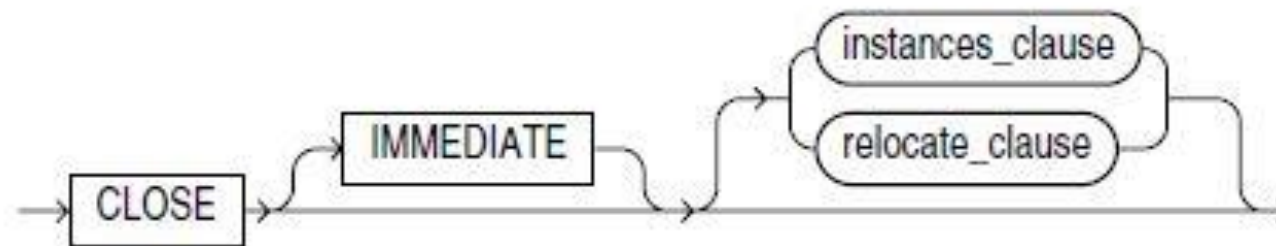
```
SHUTDOWN
```

BITTE NICHT VERWENDEN!

- Wesentlich umfangreicherer Befehlssatz
 - Schließen einer oder mehrerer Pluggable Databases

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

pdb_close::=



PDB schließen – aus der PDB heraus

- <PLUGGABLE> kann weggelassen werden

```
ALTER PLUGGABLE DATABASE CLOSE ...
```

- oder

```
ALTER DATABASE 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

- Schließen der PDB

```
SQL> ALTER PLUGGABLE DATABASE andrea CLOSE IMMEDIATE;
```

- Unplug der PDB

```
SQL> ALTER PLUGGABLE DATABASE andrea UNPLUG INTO '/u02/oradata/andrea.xml';
```

- 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
2	PDB\$SEED	9F290C149A1D0795E055FB909B80DF97	NORMAL	2
3	CELLO	9F2C4A9C6B704912E055FB909B80DF97	NORMAL	3
4	DORIA	9F3AC1DE46E50A89E0554D86E6126571	NORMAL	4
6	ANDREA	9F3AC1DE46ED0A89E0554D86E6126571	UNPLUGGED	6

- Keine Information über Status (Mount, Open, ...)

- 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	CELLO	READ WRITE
4	DORIA	READ WRITE
5	ANDREA	MOUNTED

- Keine Informationen über „UNPLUGGED“ PDBs

- v\$containers → welche PDBs gibt es (inclusive CDB)

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

CON_ID	NAME	GUID	OPEN_MODE	RES
1	CDB\$ROOT	9F290C149A1C0795E055FB909B80DF97	READ WRITE	NO
2	PDB\$SEED	9F290C149A1D0795E055FB909B80DF97	READ ONLY	NO
3	CELLO	9F2C4A9C6B704912E055FB909B80DF97	READ WRITE	NO
4	DORIA	9F3AC1DE46E50A89E0554D86E6126571	READ WRITE	NO
5	ANDREA	9F3AC1DE46ED0A89E0554D86E6126571	MOUNTED	

- Keine Informationen über „UNPLUGGED“ PDBs

- show pdbs

```
SQL> show pdbs
```

CON_ID	CON_NAME	OPEN	MODE	RESTRICTED
2	PDB\$SEED	READ ONLY		NO
3	CELLO	READ WRITE		NO
4	DORIA	READ WRITE		NO
5	ANDREA	MOUNTED		

- Keine Information über „UNPLUG“

- Unplug nur möglich, wenn die PDB einmal geöffnet (READ ONLY, READ WRITE) war

```
SQL> ALTER PLUGGABLE DATABASE andrea UNPLUG INTO '/u02/oradata/andrea.xml';  
ALTER PLUGGABLE DATABASE andrea UNPLUG INTO '/u02/oradata/andrea.xml'  
*  
ERROR at line 1:  
ORA-65085: cannot open pluggable database in read only mode
```


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

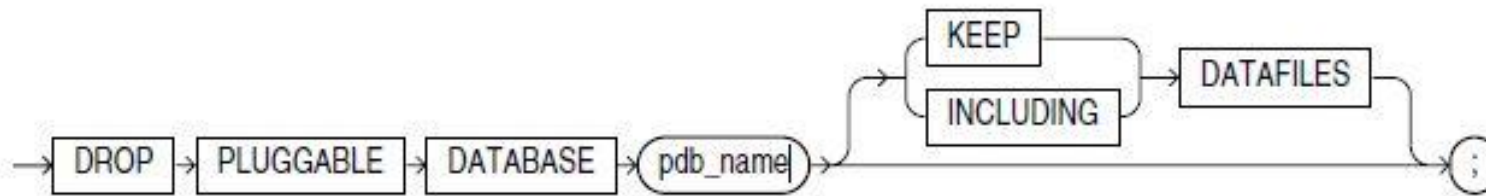
```
SELECT p.pdb_name, t.name, p.status, p.pdb_id
FROM v$tablespace t, cdb_pdb$seeds p
WHERE t.con_id = p.pdb_id ORDER BY 4;
```

PDB_NAME	NAME	STATUS	PDB_ID
-----	-----	-----	-----
PDB\$SEED	TEMP	NORMAL	2
PDB\$SEED	UNDOTBS1	NORMAL	2
...			
DORIA	TEMP	NORMAL	4
DORIA	USERS	NORMAL	4
ANDREA	SYSTEM	UNPLUGGED	6
ANDREA	SYSAUX	UNPLUGGED	6
ANDREA	UNDOTBS1	UNPLUGGED	6
ANDREA	TEMP	UNPLUGGED	6
ANDREA	USERS	UNPLUGGED	6

Drop Pluggable Database

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

drop_pluggable_database::=



- Default: KEEP DATAFILES

```
SQL> DROP PLUGGABLE DATABASE andrea;
```

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

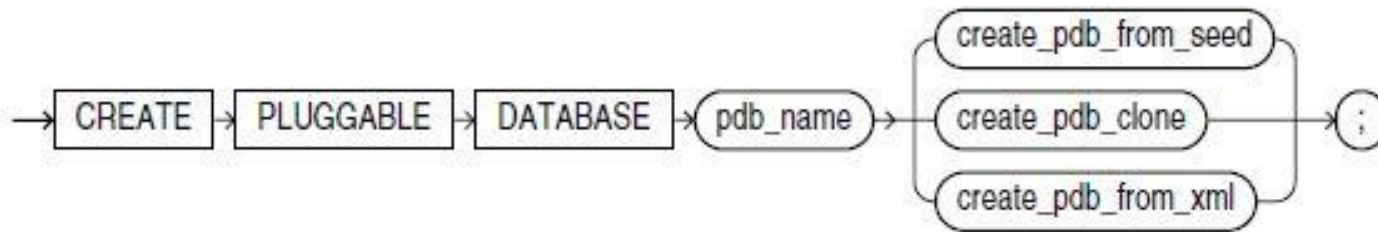
DROP PLUGGABLE DATABASE

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

... und was ist mit „PLUGIN“?

- ALTER PLUGGABLE DATABASE PLUG ... gibt es nicht
- Statt dessen: CREATE PLUGGABLE DATABASE

create_pluggable_database::=



```
SQL> CREATE PLUGGABLE DATABASE andrea USING '/u02/oradata/andrea.xml';
```

- VORSICHT: Damit werden alle Datafiles kopiert

```
SQL> CREATE PLUGGABLE DATABASE andrea USING '/u02/oradata/andrea.xml' NOCOPY;
```

Nach Recreate

- Alte Backups der PDB sind nicht mehr nutzbar!
- Status der PDB ist nicht mehr gesetzt (SAVE STATE)
- Aber: Parameter der PDB bleiben gesetzt

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>

- Multitenant für SE2 einsetzbar
- Neue Features nutzbar
 - Snapshot Standby (aber kein Switchover)
 - Cloning (speziell für Entwicklungsdatenbanken)
 - Point-in-Time Recovery einer PDB
- Konsolidierungsmöglichkeit
 - 16 Threads gelten für die CDB!
 - Ggf. eigene DB besser wegen der Ressourcenausnutzung!

- 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 Multitenant
 - 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