



Multitenant Database – pro and con!

Johannes Ahrends
CarajanDB GmbH



- **Oracle Professional since 1992**
 - 1992: Presales at Oracle
 - 1999: Project Manager at Herrmann & Lenz Services
 - 2005: Technical Director ADM Presales at Quest Software
 - 2011: Managing Director CarajanDB GmbH
- **2011 → Oracle ACE Award**
- **Author of several well known textbooks (in German):**
 - “Oracle9i für den DBA”, “Oracle10g für den DBA”, “Oracle 11g Release 2 für den DBA”
- **Responsible for Database Administration related topics at DOAG**
- **Hobbies:**
 - Kiting and esp. Indoorkiting
 - Motorbike



... and my company

- Oracle professionals with more than 25 years of experience
- Located near Cologne
- Specialized in
 - Oracle Database Administration
 - High Availability (RAC, Data Guard, Failsafe, etc.)
 - Oracle Standard Edition
 - Oracle Migrations (i.e. Unicode, Standard Edition)
 - Replication (Goldengate, SharePlex, Dbvisit)
 - Performance Optimization
 - Database Cloning (Actifio, Delphix, CloneDB)
- Trainings and Workshops (Oracle, Toad)



Contact Information



- E-Mail: johannes.ahrends@carajandb.com
- Homepage: www.carajandb.com
- Address:
 - CarajanDB GmbH
Siemensstraße 25
50374 Erftstadt
- Phone:
 - +49 (22 35) 1 70 91 84
 - +49 (1 70) 4 05 69 36
- Twitter: [@carajandb](https://twitter.com/carajandb)
- Facebook: <https://www.facebook.com/johannes.ahrends>
- Blogs:
 - blog.carajandb.com
 - www.toadworld.com

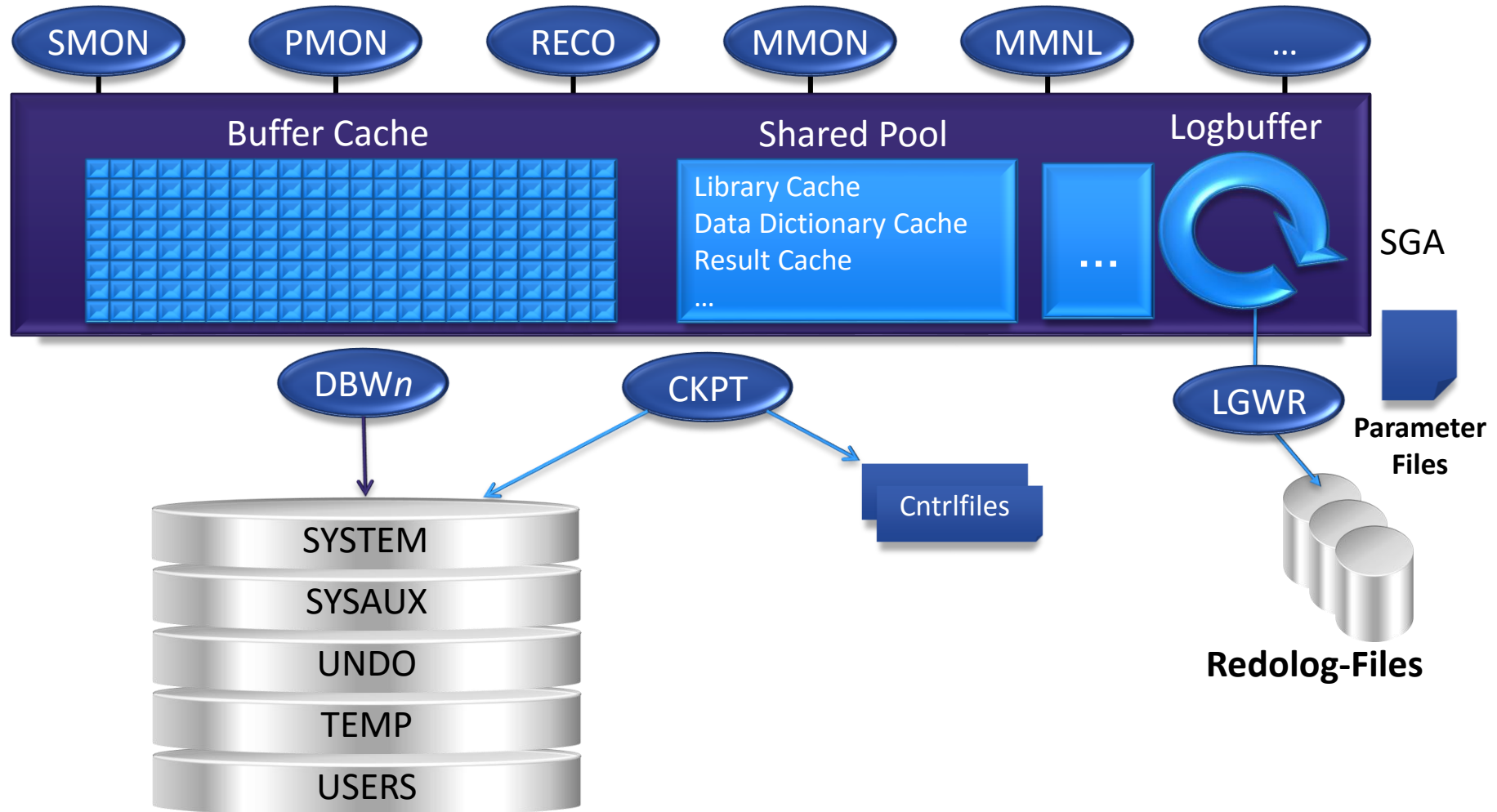
Agenda

- **Architecture**
- **Database Administration (segregation of duties)**
- **Single Tenant Database**
- **Data Cloning**
- **Advantages in Multitenant Database Environments**
- **Conclusion**

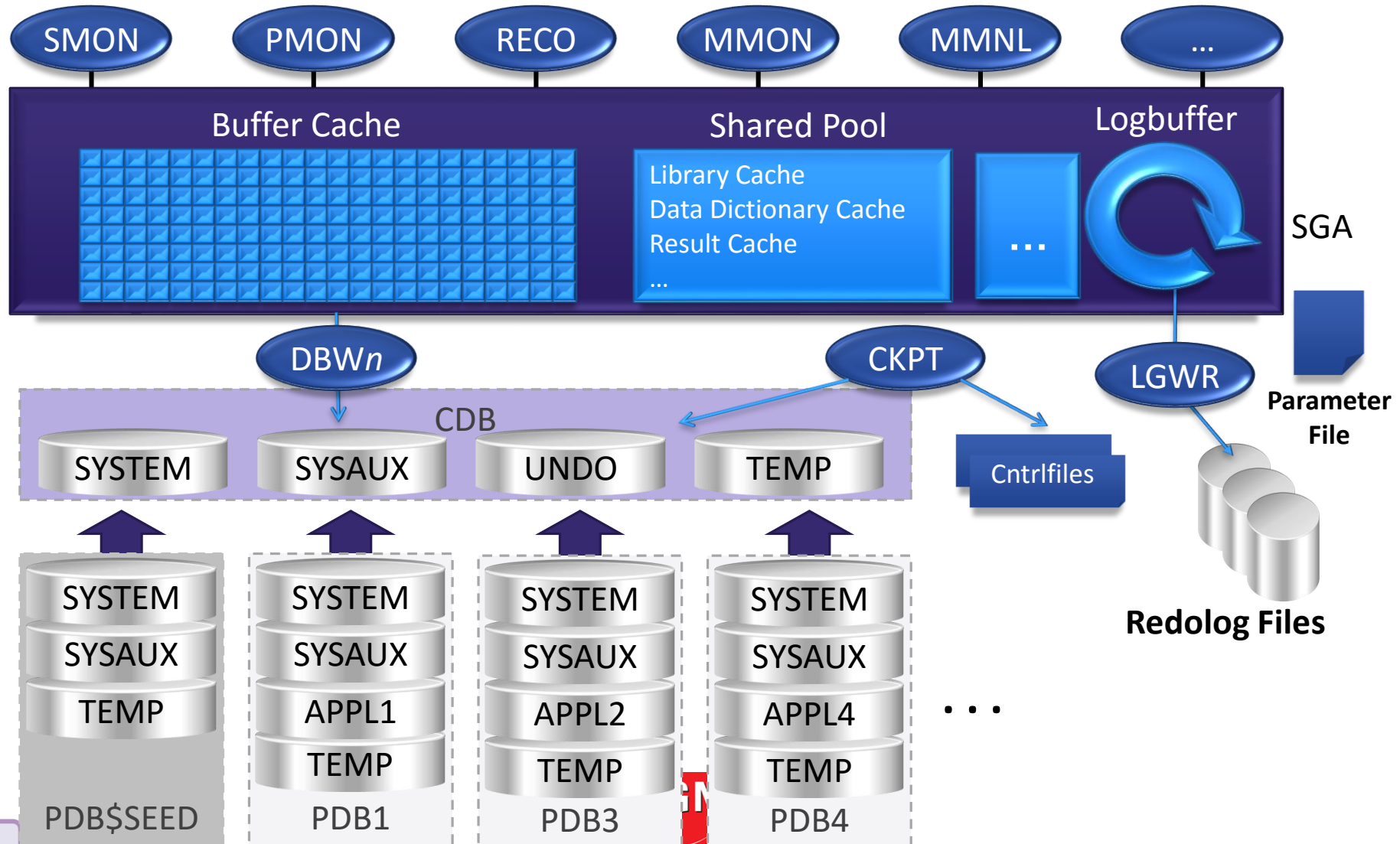
Non-CDB is Deprecated!

- *“The non-CDB architecture is deprecated in Oracle Database 12c, and may be desupported and unavailable in a later Oracle Database release. Oracle recommends use of the CDB architecture.”* (Oracle 12c Database Upgrade Guide, Chapter 8.1.1)
- *“By deprecate, we mean that the feature is no longer being enhanced but is still supported for the full life of the 12.1 release”* (Oracle 12c Database Upgrade Guide, Chapter 8)

Non-CDB Architecture



Multitenant Database Architecture



Multitenant Database

- Database Option for Enterprise Edition Only

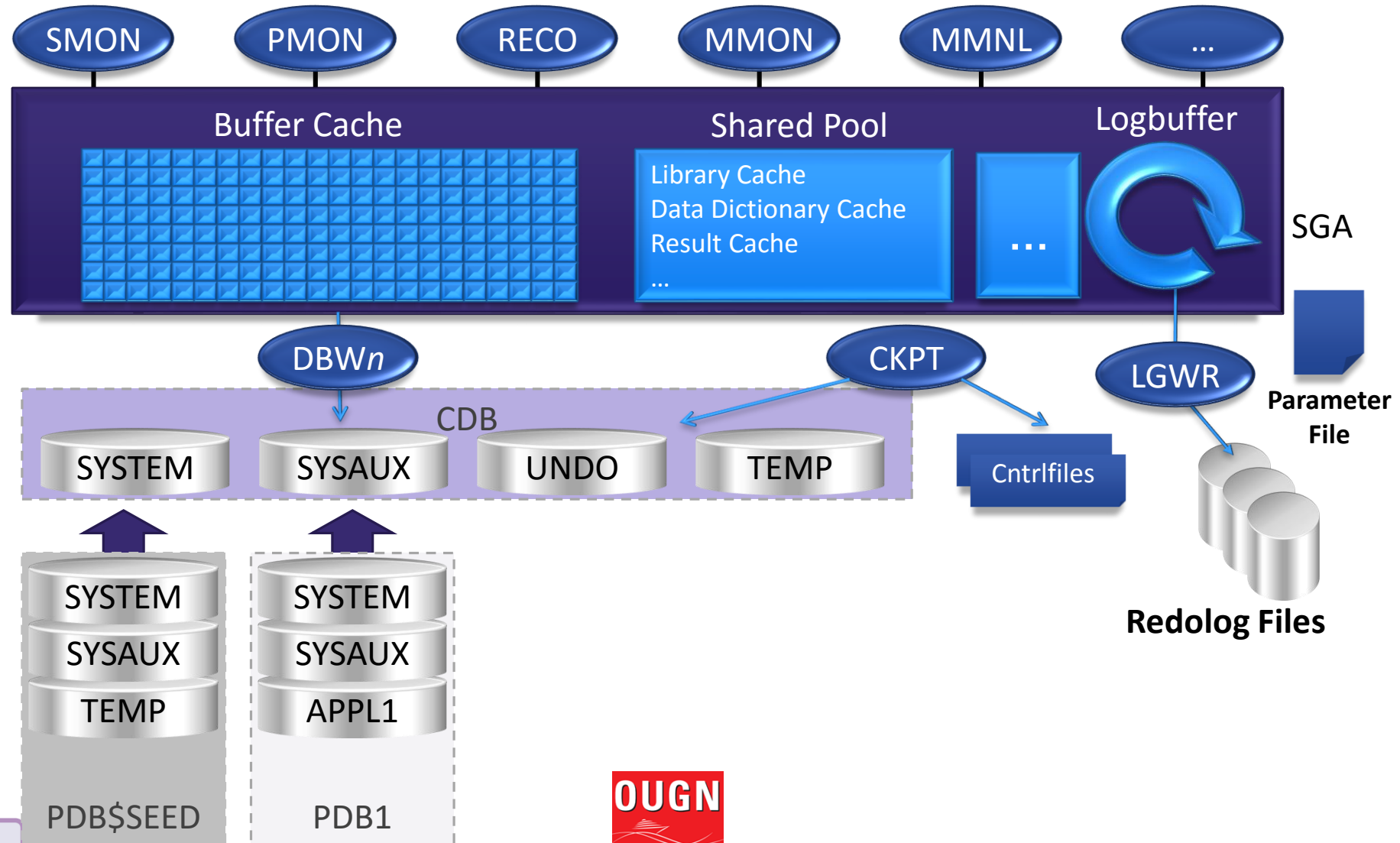
	Named User Plus	Software Update License & Support	Processor License	Software Update License & Support
Database Products				
Oracle Database				
Standard Edition One	180	39.60	5,800	1,276.00
Standard Edition	350	77.00	17,500	3,850.00
Standard Edition 2	350	77.00	17,500	3,850.00
Enterprise Edition	950	209.00	47,500	10,450.00
Personal Edition	460	101.20	-	-
Mobile Server	-	-	23,000	5,060.00
NoSQL Database Enterprise Edition	200	44	10,000	2,200.00
Enterprise Edition Options:				
Multitenant	350	77.00	17,500	3,850.00

Source: Oracle Technology Global Price List September 1, 2015 (in USD)

... and without any license fee

- **CDB with one user defined PDB does not require Multitenant Option**
 - PDB\$SEED is always included and does not count
- **This configuration = Single Tenant Database**
 - Also available for Standard Edition 2
- **WARNING: there is no chance to limit the number of PDBs in the Enterprise Edition**

Single Tenant Database Architecture



Instances per Server

- Tom Kyte:
„the only reasonable, correct number of instances on a production machine is one”
- Reality:
 - Virtualization: One Database per Guest
 - ODA: 10 – 20 Databases per Server
 - Exadata: > 50 Databases per Server

Scalability

Instance Memory	N# of Databases	ODA	Exadata	Multitenant
10 GB	1	10 GB	10 GB	10 GB
	2	20 GB	20 GB	10 GB
	10	100 GB	100 GB	10 GB
	50	500 GB	500 GB	10 GB
200 GB	1	200 GB	200 GB	200 GB
	2	400 GB	400 GB	200 GB
	10	N/A	2.000 GB	200 GB
	50	N/A	10.000 GB	200 GB

The Perfect World



- **One CDB per Server**
- **Every new PDB inherits the HA and DR configuration automatically**
 - High Availability through Real Application Clusters
 - Disaster Recovery with Dataguard
- **No more „create database“ scripts as one command line is sufficient**
- **No need for virtualization as Multitenant incorporates virtualization**
- **Patching without downtime**
- **Online PDB Cloning**
- **Entire Resource Management**
- **Flashback PDB**

... but today

- Multiple CDBs per server due to character set restrictions
- Every new PDB inherits the HA and DR configuration automatically only for Active Dataguard
- No more „create database“ scripts as one command line is sufficient but still multiple CDBs required
- CDB is “responsible for Backup and Recovery”
 - Flashback PDB only via Auxiliary Database
- Every PDB needs to be patched individually
- Only Read Only PDB Cloning
- Resource Management only for CPU and Parallelism
- All your scripts are useless!

Separation of Duty

- **CDB Administrator**
 - Software Installation
 - Database Configuration
- **Application Administrator**
 - Tablespace Management
 - User Management
 - Application Management
- **Backup and Recovery Manager**
- **Dataguard Manager**



Create a CDB Database

Database Configuration Assistant (1)

Database Configuration Assistant - Create Database - Step 2 of 5

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 Browse...

Fast Recovery Area: /u03/orabackup Browse...

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

Administrative Password:

Confirm Password:

☒ Create As Container Database

Pluggable Database Name: PLUGACFS1

☐ Advanced Mode

Help < Back Next > Finish Cancel

Database Configuration Assistant (2)

- All options must be installed in the CDB

Database Configuration Assistant - Create Database - Step 4 of 5

Summary

Database Configuration Assistant: Summary

Memory Configuration Type: Automatic Memory Management
Template Name: General Purpose or Transaction Processing

Database Configuration Details

Database Components

Component	Selected
Oracle JVM	true
Oracle Text	true
Oracle Multimedia	true
Oracle OLAP	true
Oracle Spatial	true
Oracle Label Security	true
Oracle Application Express	true
Oracle Database Vault	true

Initialization Parameters

Name	Value
audit_file_dest	{ORACLE_BASE}/admin/(DB_UNIQUE_NAME)/adump
audit_trail	db
compatible	12.0.0.0.0
db_block_size	8KB
db_create_file_dest	/u02/oradata
db_domain	
db_name	CDBACFS
db_recovery_file_dest	/u03/orabackup

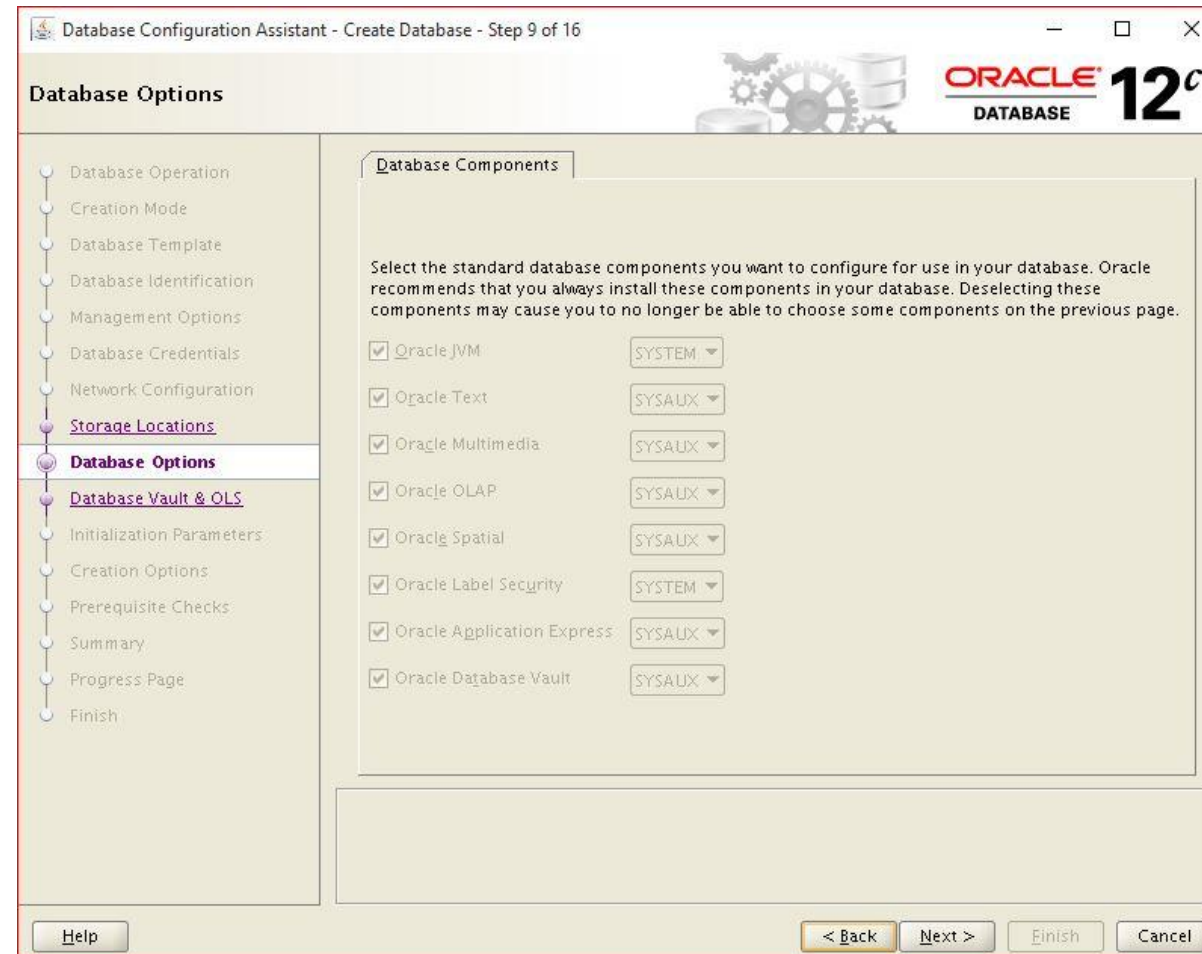
Help < Back Next > Finish Cancel

Database Configuration Assistant (3)



- **DBCA Default installation will install all options (e.g. APEX, JAVA, OLAP, Text,...)**
- **DBCA customized installation will install all options ...**
- **DBCA script will install all options**
 - BUT: you can simple comment out those options you don't want to install
 - (for details see Mike Dietrich

No chance to deselect any Option?



Generate Database Creation Scripts

Database Configuration Assistant - Create Database - Step 12 of 16

Creation Options

Select the database creation options.

☐ Create Database

☐ Save as a Database Template

Name: CDBHERB

Description: This is a template created from an existing template - NewDatabase.

☒ Generate Database Creation Scripts

Destination Directory: /u01/app/oracle/admin/CDBHERB/scripts [Browse...](#)

[Customize Storage Locations...](#)

[Help](#) [< Back](#) [Next >](#) [Finish](#) [Cancel](#)

Database Creation Script

- My Oracle Support DOC-ID 2001512.1 “Customization of Database Options in a Multitenant Setup”
- Mike Dietrich

```
oracle@Groenemeyer[HERBERT]% more CDBHERB.sql
set verify off
ACCEPT sysPassword CHAR PROMPT 'Enter new password for SYS: ' HIDE
ACCEPT systemPassword CHAR PROMPT 'Enter new password for SYSTEM: ' HIDE
ACCEPT dvOwnerPassword CHAR PROMPT 'Enter user password: ' HIDE
ACCEPT dvAccountManagerPassword CHAR PROMPT 'Enter user password: ' HIDE
host /u01/app/oracle/product/12.1.0.2/dbhome_1/bin/orapwd file=/u01/app/oracle/pr
@/u01/app/oracle/admin/CDBHERB/scripts/CreateDB.sql
@/u01/app/oracle/admin/CDBHERB/scripts/CreateDBFiles.sql
@/u01/app/oracle/admin/CDBHERB/scripts/CreateDBCatalog.sql
-- @/u01/app/oracle/admin/CDBHERB/scripts/JServer.sql
-- @/u01/app/oracle/admin/CDBHERB/scripts/context.sql
-- @/u01/app/oracle/admin/CDBHERB/scripts/ordinst.sql
-- @/u01/app/oracle/admin/CDBHERB/scripts/interMedia.sql
-- @/u01/app/oracle/admin/CDBHERB/scripts/cwmlite.sql
-- @/u01/app/oracle/admin/CDBHERB/scripts/spatial.sql
-- @/u01/app/oracle/admin/CDBHERB/scripts/labelSecurity.sql
-- @/u01/app/oracle/admin/CDBHERB/scripts/apex.sql
-- @/u01/app/oracle/admin/CDBHERB/scripts/datavault.sql
@/u01/app/oracle/admin/CDBHERB/scripts/CreateClustDBViews.sql
@/u01/app/oracle/admin/CDBHERB/scripts/lockAccount.sql
@/u01/app/oracle/admin/CDBHERB/scripts/postDBCcreation.sql
```




Database Administration

- The command „UNPLUG“ exists – but not „PLUGIN“
- You can only „UNPLUG“ a mounted but not opened PDB
- The Metainformation – called manifest – is stored in an XML-File (postfix .xml mandatory)

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

- An „UNPLUGGED“ Database will still be backed up

- Information about „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

- **V\$CONTAINERS**

```
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

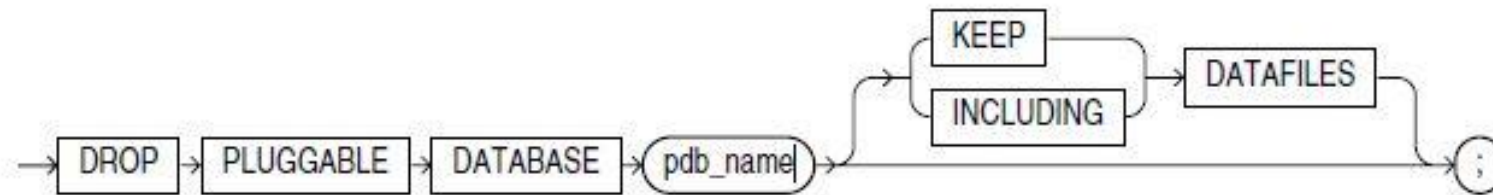
- No information about „UNPLUGGED“ PDBs

- **What if you accidentally unplugged a PDB?**
 - Or unplugged it and you later want to plug it in again?
- **There is no command „Plugin“!**

Unplug Pluggable Database

- After „UNPLUG“ the only possible comand is: „DROP“

drop_pluggable_database::=



- Default: KEEP DATAFILES

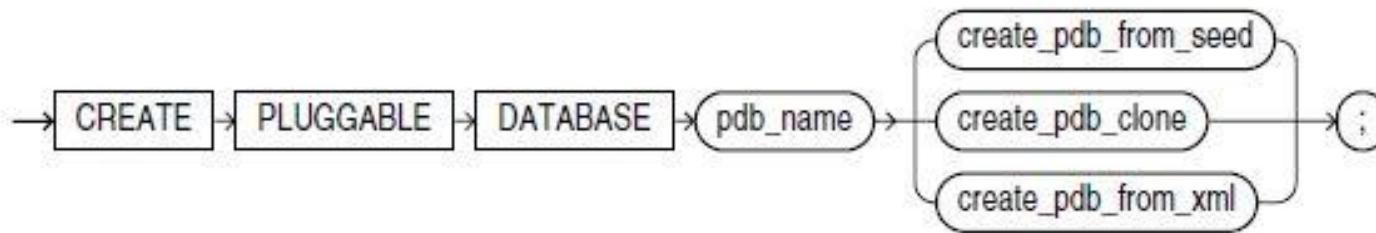
```
DROP PLUGGABLE DATABASE johann2;
```

- Deletes all PDB related entries in the data dictionary and control files
- Default will not remove the files!

... so what about „PLUGIN“?

- ALTER PLUGGABLE DATABASE PLUGIN ... does not exists
- Instead use: CREATE PLUGGABLE DATABASE

create_pluggable_database::=



```
CREATE PLUGGABLE DATABASE johann2 USING '/tmp/pdb_johann2.xml';
```

- **WARNING:** in this case the original files will be copied. Instead use:

```
CREATE PLUGGABLE DATABASE johann2 USING '/tmp/pdb_johann2.xml' NOCOPY;
```

- **Useful for pluggable Databases**

- `db_create_file_dir` → Base directory for the CDB files
- `db_create_file_dir/GUID` → Base directory for the PDB files

PDB_ID	PDB_NAME	DBID	GUID
3	ANTON1	2660983207	EF89F348D7051BF3E043281E10ACCCD7
2	PDB\$SEED	4078985758	EF899360B02E192DE043281E10ACBA7D
4	ANTON2	2965167653	EF8AC4B702D10788E043281E10ACE619

```
SQL> !ls -l /u02/oradata/CANTON
```

```
total 20
```

```
drwxr-x--- 2 oracle oinstall 4096 Jan  9 13:13 controlfile
drwxr-x--- 2 oracle oinstall 4096 Jan  9 13:16 datafile
drwxr-x--- 3 oracle oinstall 4096 Jan  9 13:43 EF89F348D7051BF3E043281E10ACCCD7
drwxr-x--- 3 oracle oinstall 4096 Jan  9 14:41 EF8AC4B702D10788E043281E10ACE619
drwxr-x--- 2 oracle oinstall 4096 Jan  9 13:13 onlineolog
```


- **Directory must exist!**

```
SQL> CREATE PLUGGABLE DATABASE theo ADMIN USER pdb_admin IDENTIFIED BY manager  
      CREATE_FILE_DEST='/u02/oradata/HANNES/THEO';
```

```
SQL> SELECT d.file_name  
      FROM   cdb_data_files d, v$pdb p  
      WHERE  p.con_id = d.con_id  
      AND    p.name = 'THEO';
```

FILE_NAME

```
-----  
/u02/oradata/HANNES/THEO/HANNES/040F34E2A2CF2025E0530B6E10ACD6DB/datafile/o1_mf_system_b2fsvkqd_.dbf  
/u02/oradata/HANNES/THEO/HANNES/040F34E2A2CF2025E0530B6E10ACD6DB/datafile/o1_mf_sysaux_b2fsvkqg_.dbf
```



DBA Scripts

- DBAs have ton's of scripts to manage and monitor the database
- Simple example:

```
SELECT tablespace_name,  
       trunc(sum(bytes)/1024/1024) Mbytes,  
       trunc(sum(decode(maxbytes,0,bytes,maxbytes)/1024/1024)) maxmbytes  
FROM dba_data_files  
GROUP BY tablespace_name;
```

TABLESPACE_NAME	MBYTES	MAXMBYTES
-----	-----	-----
SYSAUX	870	32767
UNDOTBS1	920	32767
USERS	5	32767
SYSTEM	800	32767

- Is this the correct result?

- It depends!
- The result is probably for a CDB (as it includes the UNDO-Tablespace)
- But what is the correct result?

```
SELECT tablespace_name,  
       trunc(sum(bytes)/1024/1024) Mbytes,  
       trunc(sum(decode(maxbytes,0,bytes,maxbytes)/1024/1024)) maxmbytes  
FROM cdb_data_files  
GROUP BY tablespace_name;
```

TABLESPACE_NAME	MBYTES	MAXMBYTES
USERS	405	38911
UNDOTBS1	920	32767
SYSTEM	1320	98303
SYSAUX	2070	98303

And what you ment...

```
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
3	USERS	100	5120
3	SYSAUX	600	32767
3	SYSTEM	260	32767
4	USERS	300	1024
4	SYSTEM	260	32767
4	SYSAUX	600	32767



Application Management

- **Predefined Service for every PDB exists**
- **Recommendation:**
 - Create your own service, which has an association with the application
- **Grid Infrastructure:**

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

- PDB will be opened whenever the service is started
- PDB will not close when the service is terminated

- **Without Grid Infrastructure:**

- dbms_service Package
- Executed from within the PDB

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

```
SQL> BEGIN
      dbms_service.create_service(
        service_name => 'CUSTOMERAPPLICATION',
        network_name => 'CUSTOMERAPPLICATION.CARAJANDB.DE');
END;
```

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


- **tnsnames.ora**

```
CUSTOMERAPPLICATION =  
  (DESCRIPTION =  
    (ADDRESS_LIST =  
      (ADDRESS = (PROTOCOL = TCP) (HOST = CDB-JA2) (PORT = 1521))  
    )  
    (CONNECT_DATA =  
      (SERVICE_NAME = CUSTOMERAPPLICATION)  
    )  
  )
```

- **No indication that this is a pluggable database!**



Resource Management

- Only one SGA → No limitation for Buffer cache, Shared Pool, etc.
- Only one set of Redologs and one Undo-Tablespace
- No limitation for I/O bandwidth
- Limitation for PDB size (esp. useful in test environments)
- Resource limitation and shares for CPU and parallelism
- Limitation for PDB size (esp. useful in test environments)
 - `ALTER PLUGGABLE DATABASE
STORAGE (MAXSIZE 10G) ;`

- **CDB Resource Plans**
 - Assignment of resources for individual PDBs
- **PDB Resource Plans**
 - Assignment of resources within one individual PDB

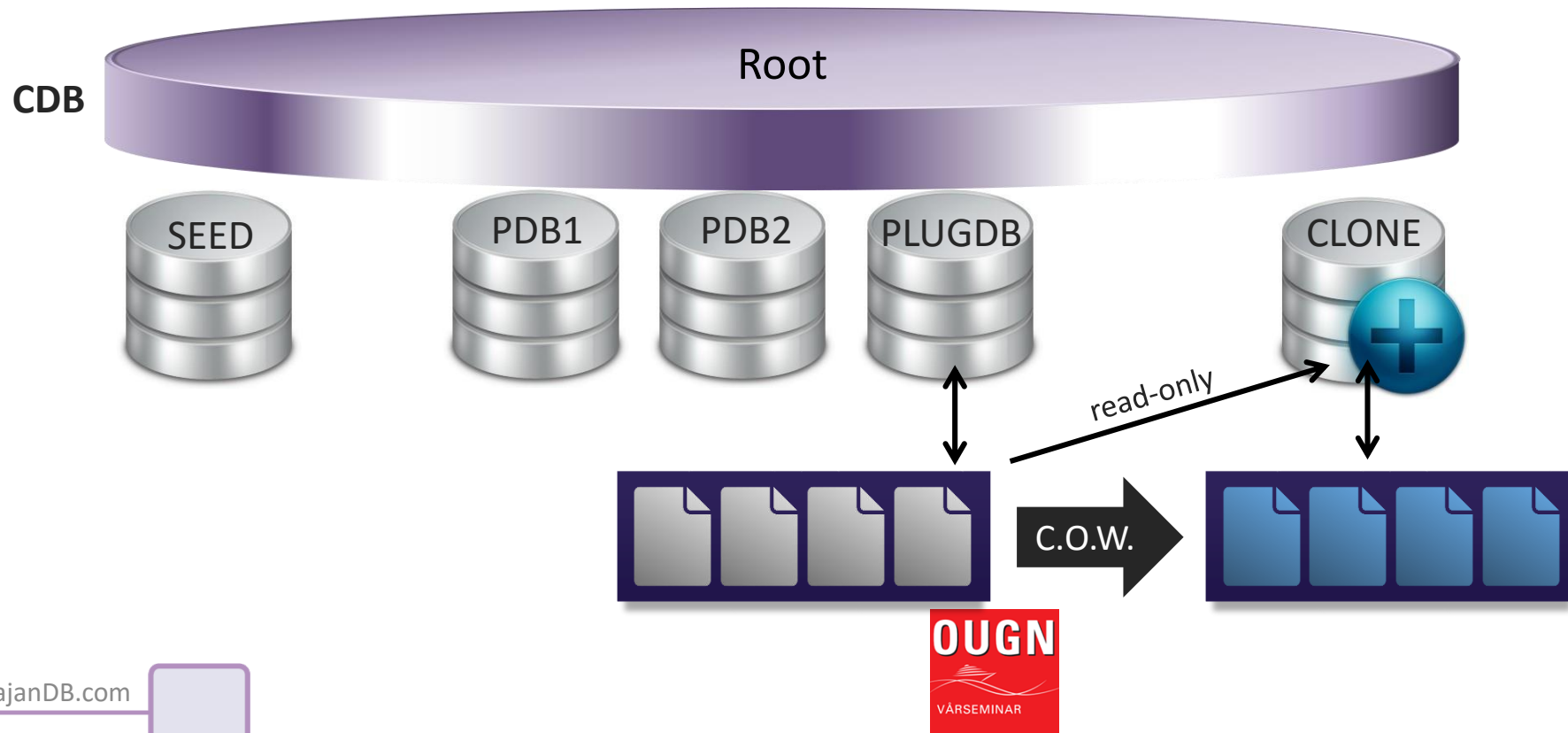
- **Resource Shares**
 - Ratio between PDBs
- **Currently only for**
 - CPU
 - Parallelism
- **Controlled via directives**



Database Cloning

Pluggable Database Snapshot Clones

- **Copy On Write (C.O.W)**
 - Create PDBs within seconds
 - Snapshot clone: read remote - write local



- **Prerequisites**
 - Multitenant-Architecture
- **Advantages**
 - Cloning instantly
 - Minimized storage footprint
 - Ideal solution for test and development
- **Restrictions**
 - Source PDB open read only while cloning (few seconds)
 - Source and target PDB in the same CDB
 - Source PDB cannot be unplugged or dropped as long as one clone exists

- **CLONEDB Parameter**
 - Set Parameter cloneDB (spfile)

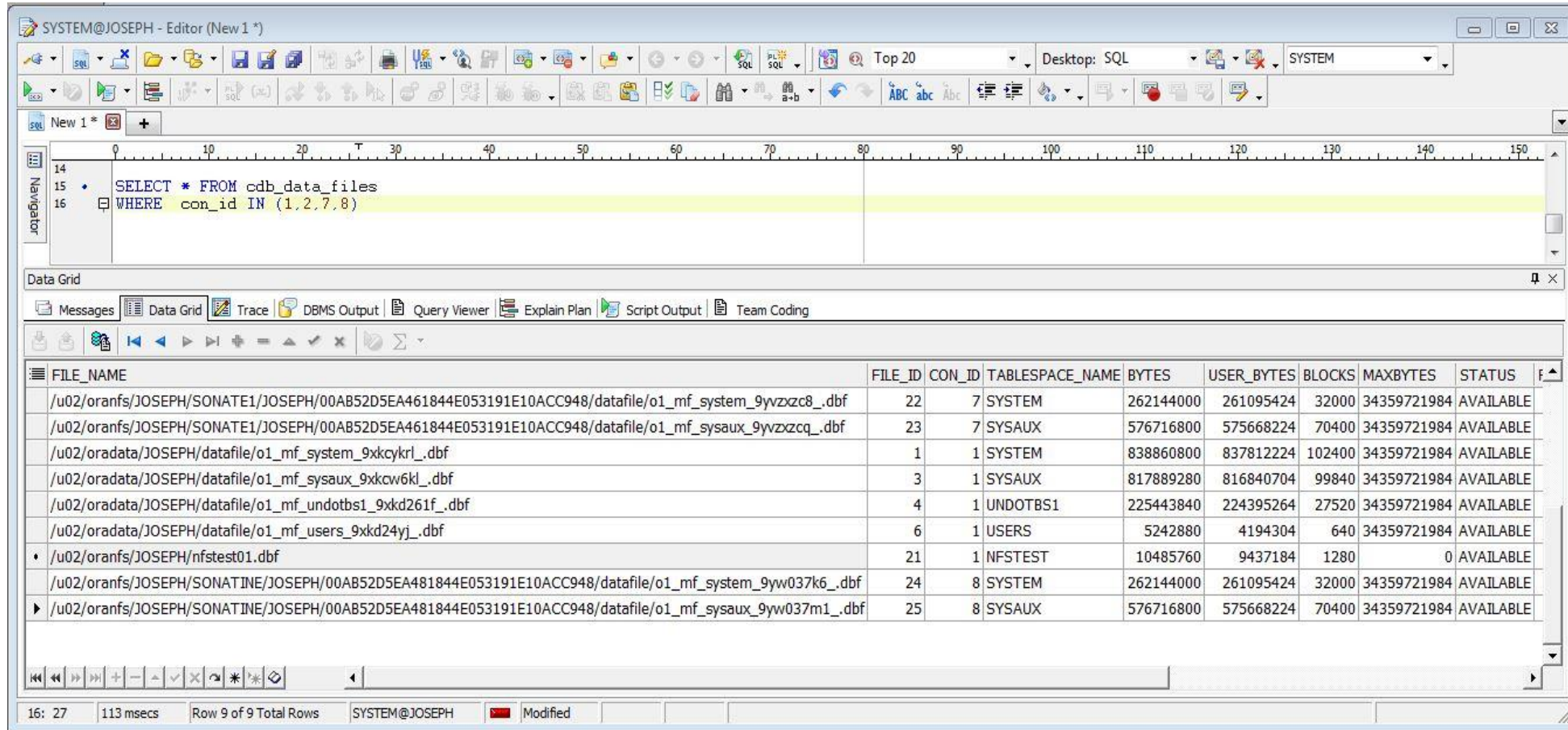
```
SQL> ALTER SYSTEM SET clonedb=true SCOPE=spfile;  
SQL> SHUTDOWN IMMEDIATE;  
SQL> STARTUP;
```

- Use Parameter **CREATE_FILE_DEST** for individual file locations
- Directory must exists (e.g. SONATINE).

```
SQL> CREATE PLUGGABLE DATABASE sonate1  
      ADMIN USER pdb_admin IDENTIFIED BY manager  
      CREATE_FILE_DEST='/u02/oranfs/JOSEPH/SONATE1';  
SQL> ALTER PLUGGABLE DATABASE sonate1 OPEN;  
SQL> ALTER PLUGGABLE DATABASE sonate1 OPEN READ ONLY FORCE;  
SQL> ALTER PLUGGABLE DATABASE sonate1 SAVE STATE;  
SQL> CREATE PLUGGABLE DATABASE sonatine FROM sonate1  
      CREATE_FILE_DEST='/u02/oranfs/JOSEPH/SONATINE' SNAPSHOT COPY;
```

Snapshot Copy PDB

- Validate storage (CON ID 7 + 8)



The screenshot shows the SQL Developer interface with a query executed in the SQL Editor. The query is:

```
SELECT * FROM cdb_data_files  
WHERE con_id IN (1,2,7,8)
```

The results are displayed in the Data Grid, showing a table with columns: FILE_NAME, FILE_ID, CON_ID, TABLESPACE_NAME, BYTES, USER_BYTES, BLOCKS, MAXBYTES, STATUS, and F. The table contains 10 rows of data, including files for CON_ID 7 and 8.

FILE_NAME	FILE_ID	CON_ID	TABLESPACE_NAME	BYTES	USER_BYTES	BLOCKS	MAXBYTES	STATUS	F
/u02/oranfs/JOSEPH/SONATE1/JOSEPH/00AB52D5EA461844E053191E10ACC948/datafile/o1_mf_system_9yvzxc8_.dbf	22	7	SYSTEM	262144000	261095424	32000	34359721984	AVAILABLE	
/u02/oranfs/JOSEPH/SONATE1/JOSEPH/00AB52D5EA461844E053191E10ACC948/datafile/o1_mf_sysaux_9yvzxcq_.dbf	23	7	SYSAUX	576716800	575668224	70400	34359721984	AVAILABLE	
/u02/oradata/JOSEPH/datafile/o1_mf_system_9xkcykrl_.dbf	1	1	SYSTEM	838860800	837812224	102400	34359721984	AVAILABLE	
/u02/oradata/JOSEPH/datafile/o1_mf_sysaux_9xkcw6kl_.dbf	3	1	SYSAUX	817889280	816840704	99840	34359721984	AVAILABLE	
/u02/oradata/JOSEPH/datafile/o1_mf_undotbs1_9xkd261f_.dbf	4	1	UNDOTBS1	225443840	224395264	27520	34359721984	AVAILABLE	
/u02/oradata/JOSEPH/datafile/o1_mf_users_9xkd24yj_.dbf	6	1	USERS	5242880	4194304	640	34359721984	AVAILABLE	
• /u02/oranfs/JOSEPH/nfstest01.dbf	21	1	NFSTEST	10485760	9437184	1280	0	AVAILABLE	
/u02/oranfs/JOSEPH/SONATINE/JOSEPH/00AB52D5EA481844E053191E10ACC948/datafile/o1_mf_system_9yw037k6_.dbf	24	8	SYSTEM	262144000	261095424	32000	34359721984	AVAILABLE	
▶ /u02/oranfs/JOSEPH/SONATINE/JOSEPH/00AB52D5EA481844E053191E10ACC948/datafile/o1_mf_sysaux_9yw037m1_.dbf	25	8	SYSAUX	576716800	575668224	70400	34359721984	AVAILABLE	

- **But do we really use snapshot copy?**
 - Use the Linux command „du“

```
$ du -sh *
```

```
11M      nfstest01.dbf
801M     SONATE1
952K     SONATINE
```

Conclusion

- YES – Pluggable database are different to our past behavior
- YES – Pluggable database has some limitations (12.2 ?)
- YES – Single Tenant has no advantages over Non-CDB (12.2 ?)
- YES – Multitenant Database scales even better than Exadata
- YES – Multitenant Database is easy to implement
- YES – Multitenant Database enables fast provisioning
- YES – Multitenant Database enables snapshot cloning
- YES – Multitenant Database option is expensive
- YES – Only CPU and Parallelism can be limited (12.2 ?)



Questions?

Johannes Ahrends

Johannes.ahrends@carajandb.com

blog.carajandb.com

Twitter: @carajandb

