



Don't fear Pluggable Databases!

Johannes Ahrends
CarajanDB GmbH



DOAG
Deutsche ORACLE-Anwendergruppe e.V.

- **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

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

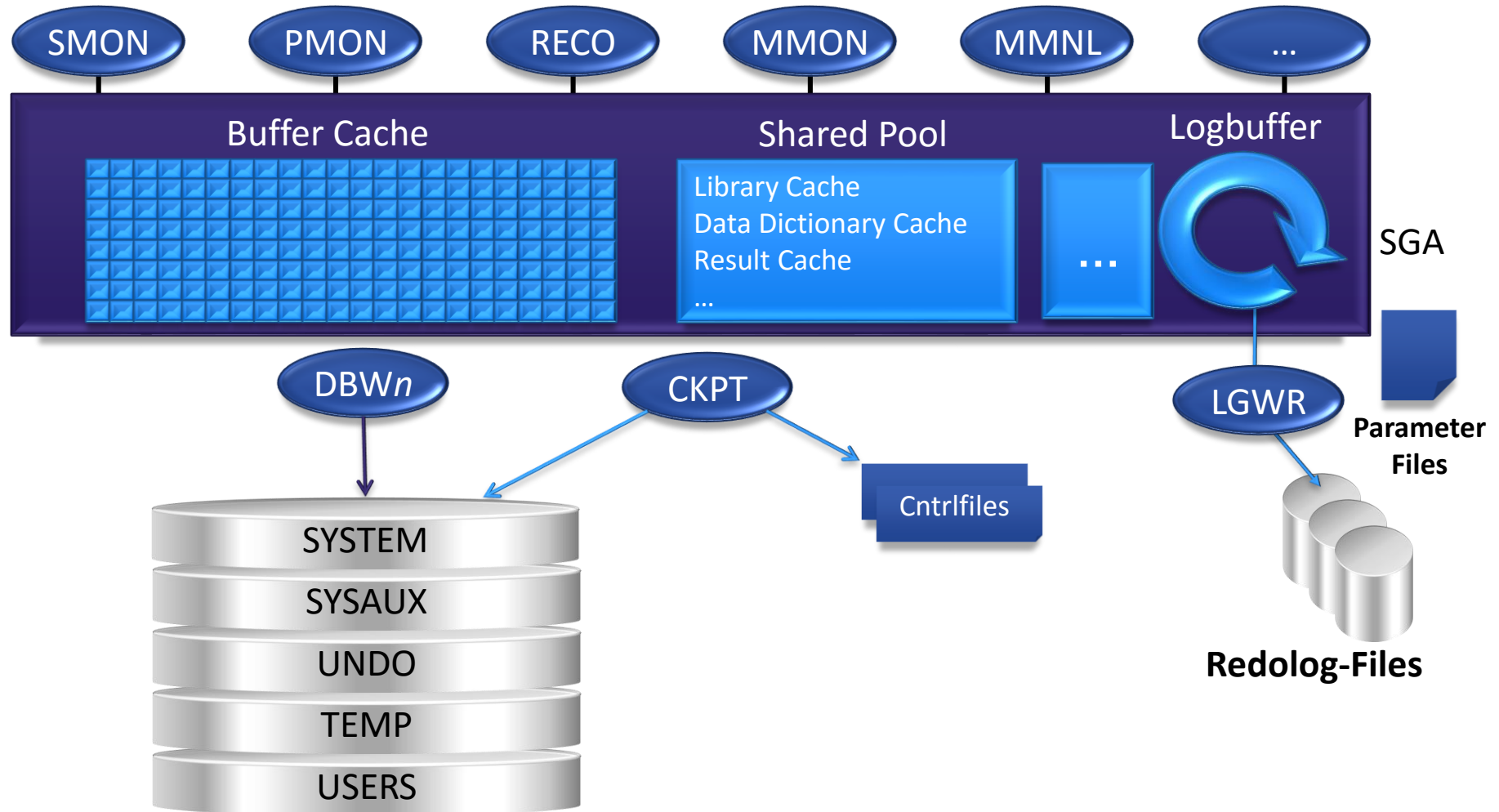
- **What do Multitenant Database and Indoorkiting have in common?**
 - Used in rare cases ...
 - Start slow ...



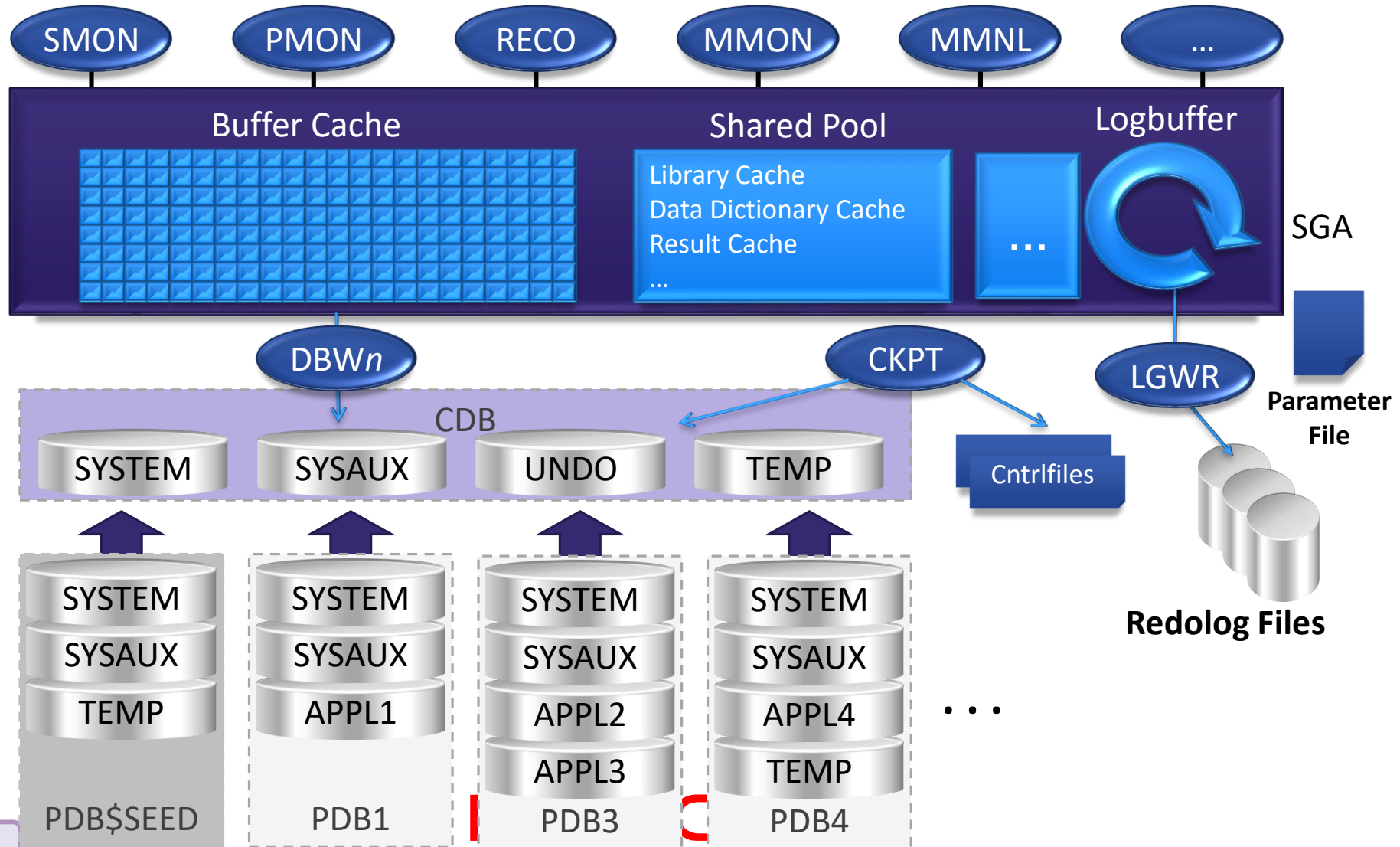
Non-CDB is Deprecated!

- ***“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



- Database Option for Enterprise Edition Only

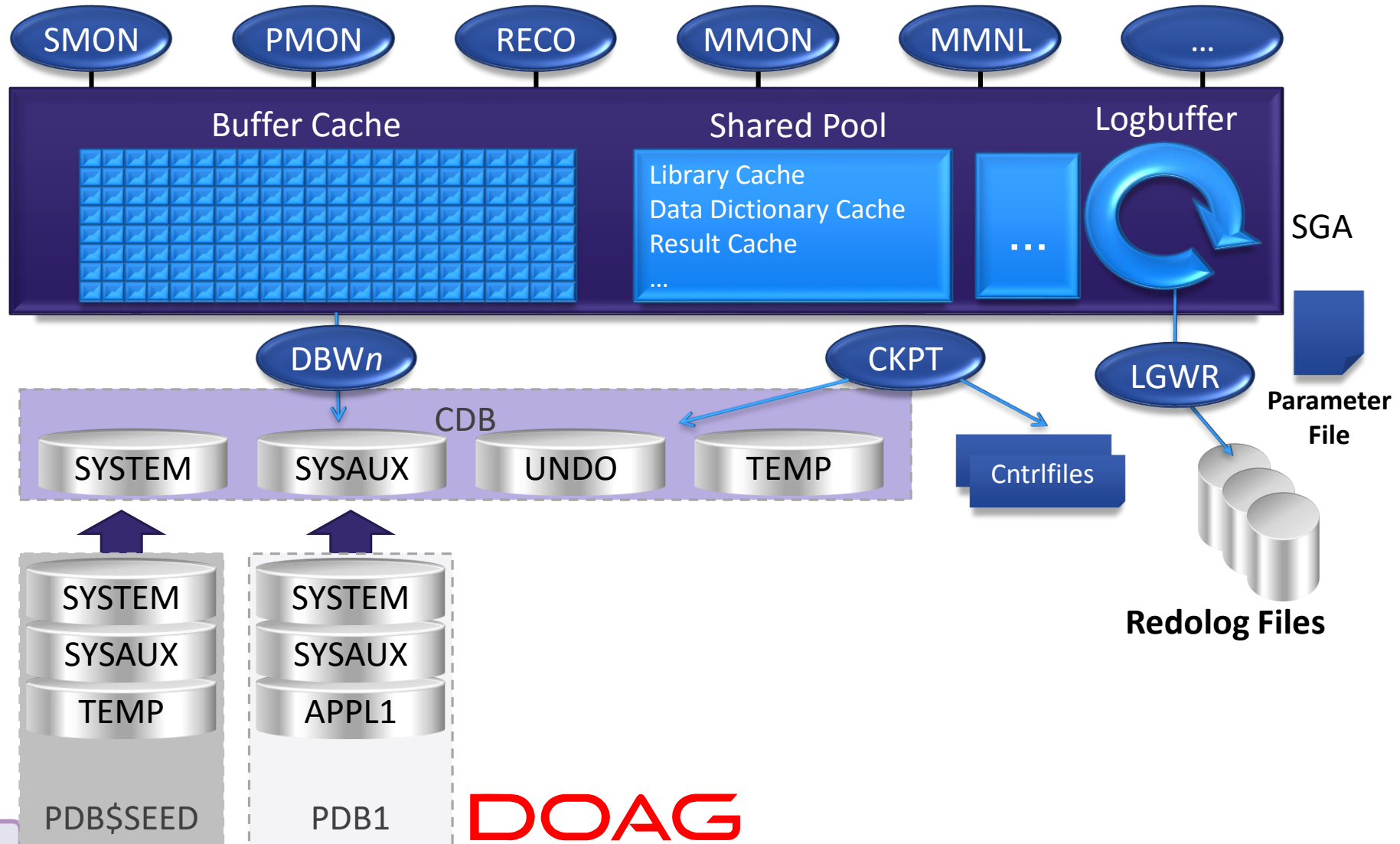
	Named User Plus	Software Update License & Support	Processor License	Software Update License & Support
Database Products				
Oracle Database				
Standard Edition One	180	39.60	5,800	1,276.00
Standard Edition	350	77.00	17,500	3,850.00
Standard Edition 2	350	77.00	17,500	3,850.00
Enterprise Edition	950	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 Standard Edition 1 2 3

- **Standard Edition and Standard Edition One:**
 - Version 12.1.0.1 → 1 Pluggable Database per CDB → Single Tenant Database
- **Standard Edition 2**
 - Version 12.1.0.2 → 1 Pluggable Database per CDB → Single Tenant Database

Singletenant 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

- **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**

- 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
- No need for virtualization as Multitenant incorporates virtualization
- Every PDB needs to be patched individually
- Read Only PDB Cloning
- Resource Management for CPU and Parallelism

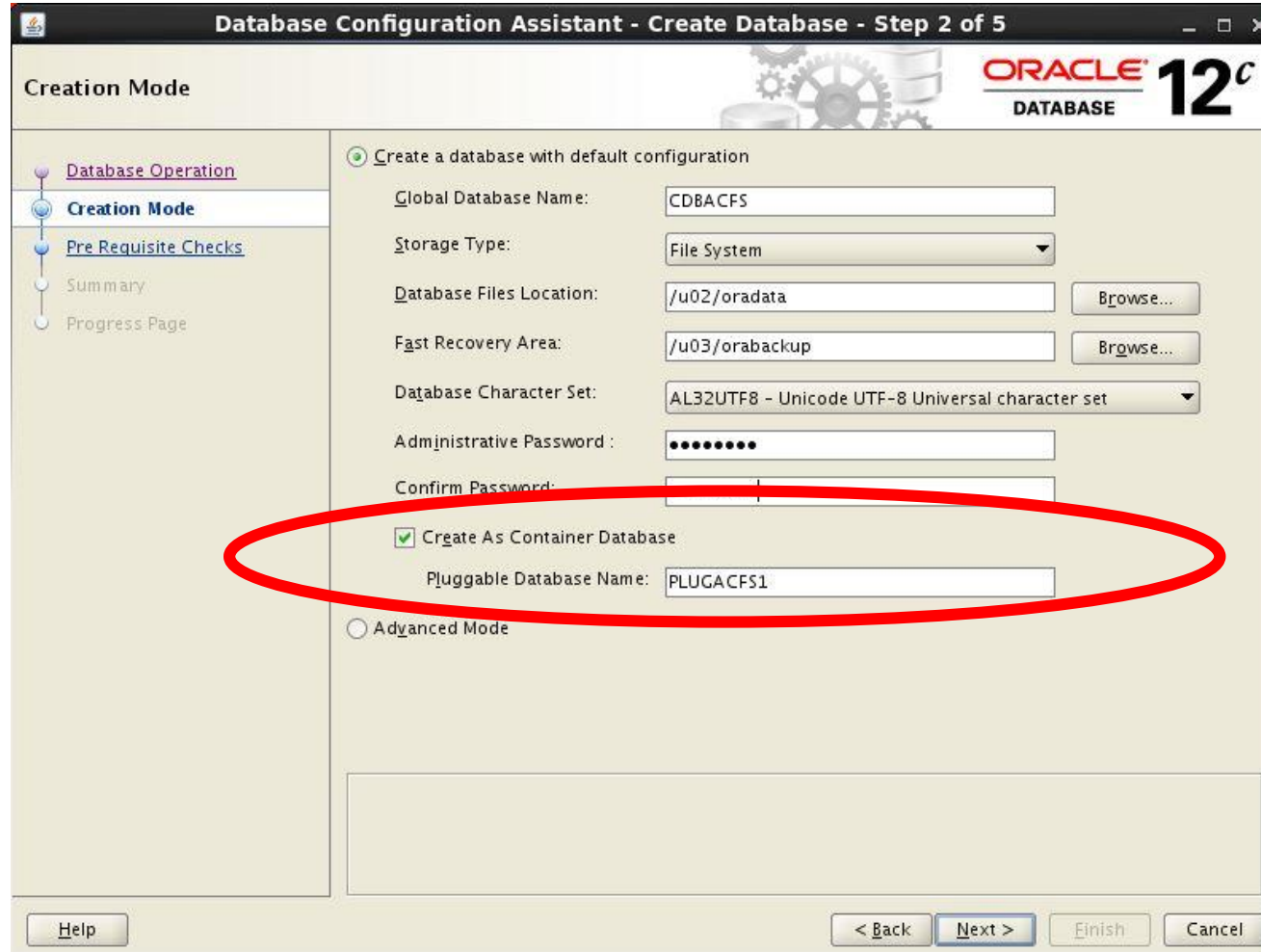
Segregation of Duty

- **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)



The screenshot shows the Oracle Database Configuration Assistant (DBCA) window for creating a database. The window title is "Database Configuration Assistant - Create Database - Step 2 of 5". The "Creation Mode" section is active, and the "Create a database with default configuration" radio button is selected. The "Create As Container Database" checkbox is checked and circled in red. The "Pluggable Database Name" field contains "PLUGACFS1". Other fields include "Global Database Name" (CDBACFS), "Storage Type" (File System), "Database Files Location" (/u02/oradata), "Fast Recovery Area" (/u03/orabackup), and "Database Character Set" (AL32UTF8 - Unicode UTF-8 Universal character set). The "Administrative Password" and "Confirm Password" fields are masked with dots. The "Advanced Mode" radio button is unselected. The "Next >" button is highlighted.

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

Fast Recovery Area: /u03/orabackup

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

Administrative Password:

Confirm Password:

Create As Container Database

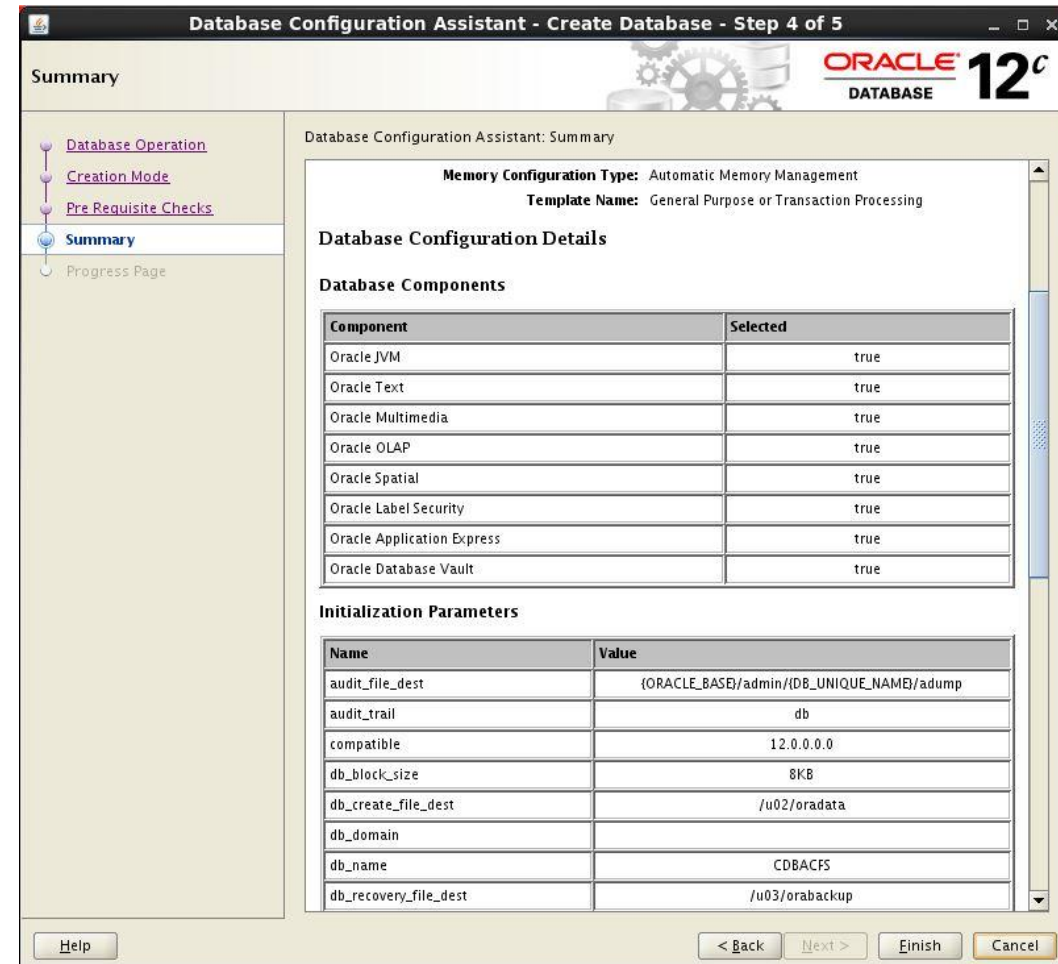
Pluggable Database Name: PLUGACFS1

Advanced Mode

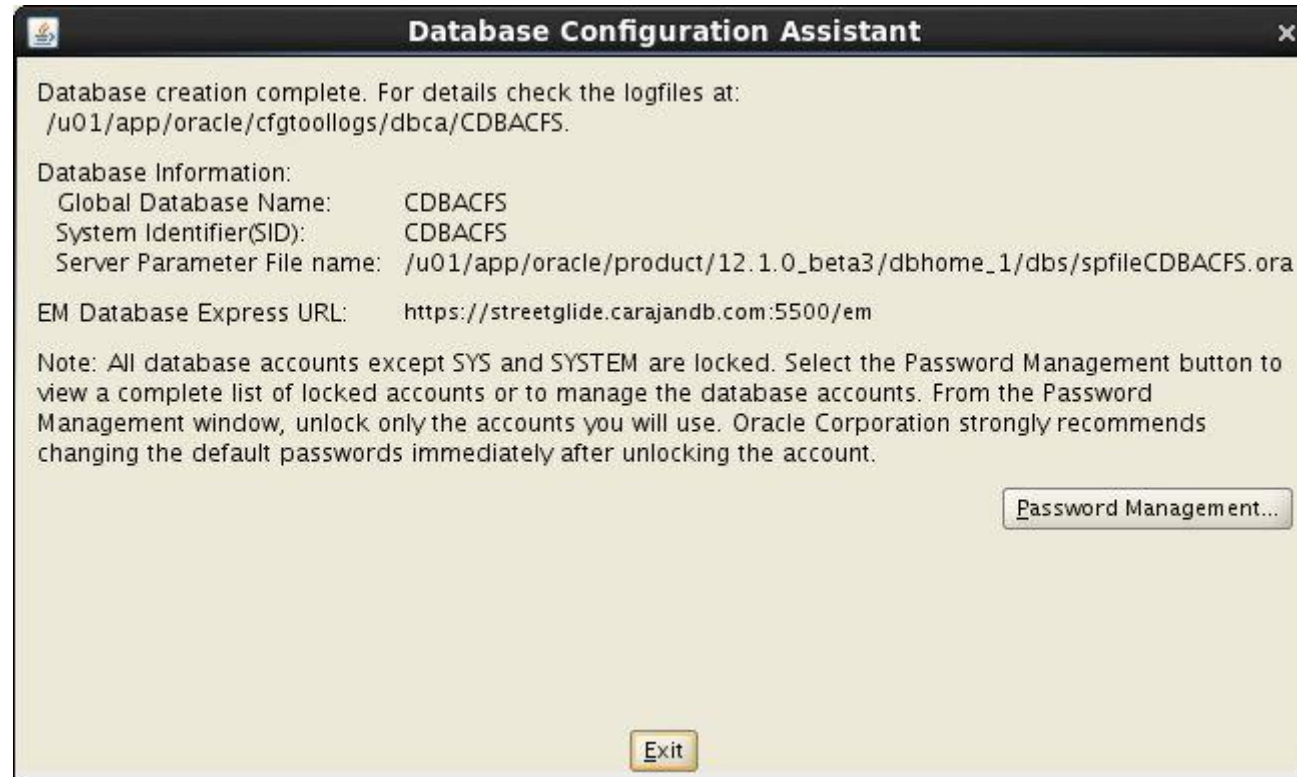
Help

Database Configuration Assistant (2)

- All options must be installed in the CDB



Database Configuration Assistant (3)





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) ;`

- **Instance Caging**
 - Using `cpu_count` for max. number of threads
 - Resource Manager must be enabled
- **But `cpu_count` is responsible for the entire instance and cannot be set on PDB level**

- **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**

- **DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE**
- **Similar to ordinary resource plans**
 1. Create a pending area
 2. Create a resource plan
 3. Create directives
 4. Validate pending area
 5. Submit pending area

- **Pending area**

```
BEGIN
    dbms_resource_manager.create_pending_area();
END;
```

- **Resource plans**

```
BEGIN
    dbms_resource_manager.create_cdb_plan(
        PLAN                => 'johanns_plan',
        COMMENT              => 'CDB Resource Plan for Database CJOHANN');
END;
```


- **create_cdb_plan_directive**

- Plan for individual Pluggable Database

- Three Parameters:

- shares → ratio between CPU consumption (total of all shares = 100%)
- utilization_limit → max. percentage of CPU consumption
- parallel_server_limit → maximal percentage of parallelism

```
dbms_resource_manager.create_cdb_plan_directive (  
    PLAN                => <planname>,  
    pluggable_database  => <PDB>,  
    shares              => <number>,  
    utilization_limit   => <Number>,  
    parallel_server_limit => <Number>);
```

Directives Example

```
BEGIN
  dbms_resource_manager.create_cdb_plan_directive(
    PLAN                => 'johanns_plan',
    pluggable_database  => 'JOHANN1',
    shares              => 100,
    utilization_limit   => 100,
    parallel_server_limit => 100);
  dbms_resource_manager.create_cdb_plan_directive(
    PLAN                => 'johanns_plan',
    pluggable_database  => 'JOHANN2',
    shares              => 100,
    utilization_limit   => 100,
    parallel_server_limit => 100);
  dbms_resource_manager.create_cdb_plan_directive(
    PLAN                => 'johanns_plan',
    pluggable_database  => 'JOHANN1CLONE',
    shares              => 10,
    utilization_limit   => 50,
    parallel_server_limit => 0);
END;
```

- **Validation of pending area**

```
BEGIN
    dbms_resource_manager.validate_pending_area();
END;
```

- **Submit**

```
BEGIN
    dbms_resource_manager.submit_pending_area();
END;
```

Resource Plan Activation

- Enable Resource Plan
- Called from the CDB

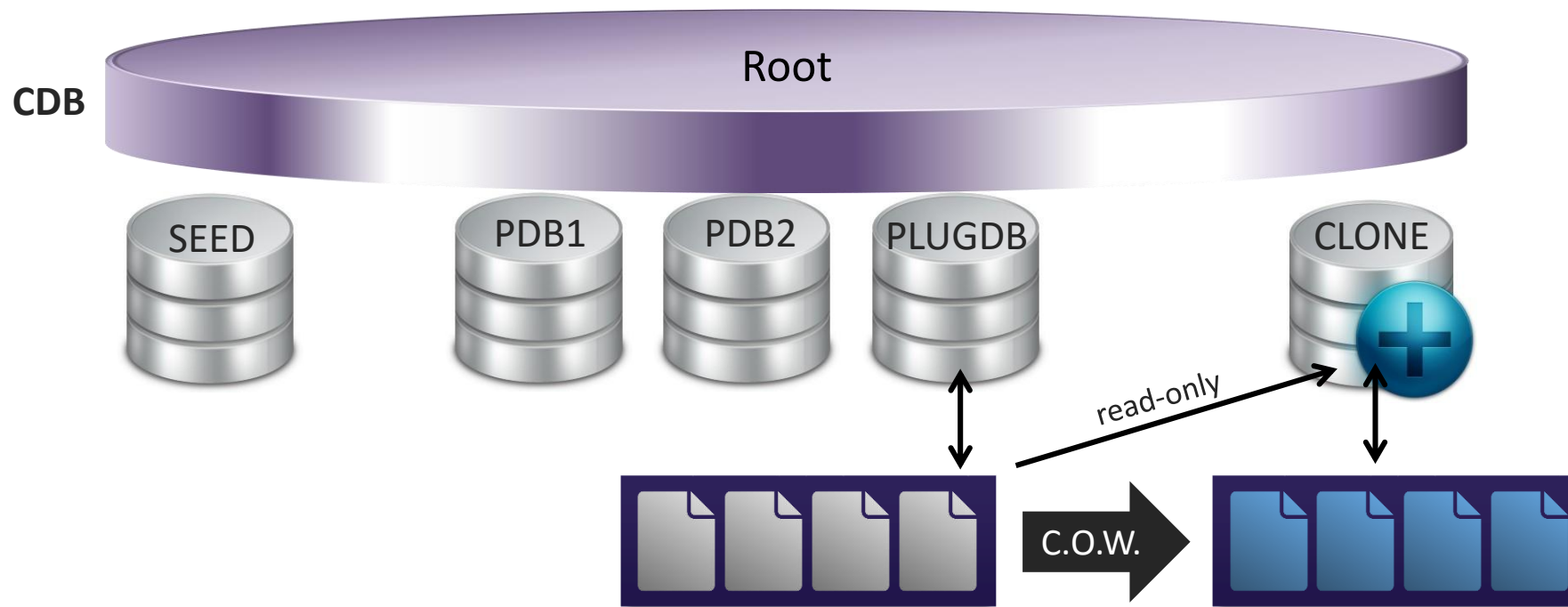
```
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = 'johanns_plan';
```



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)

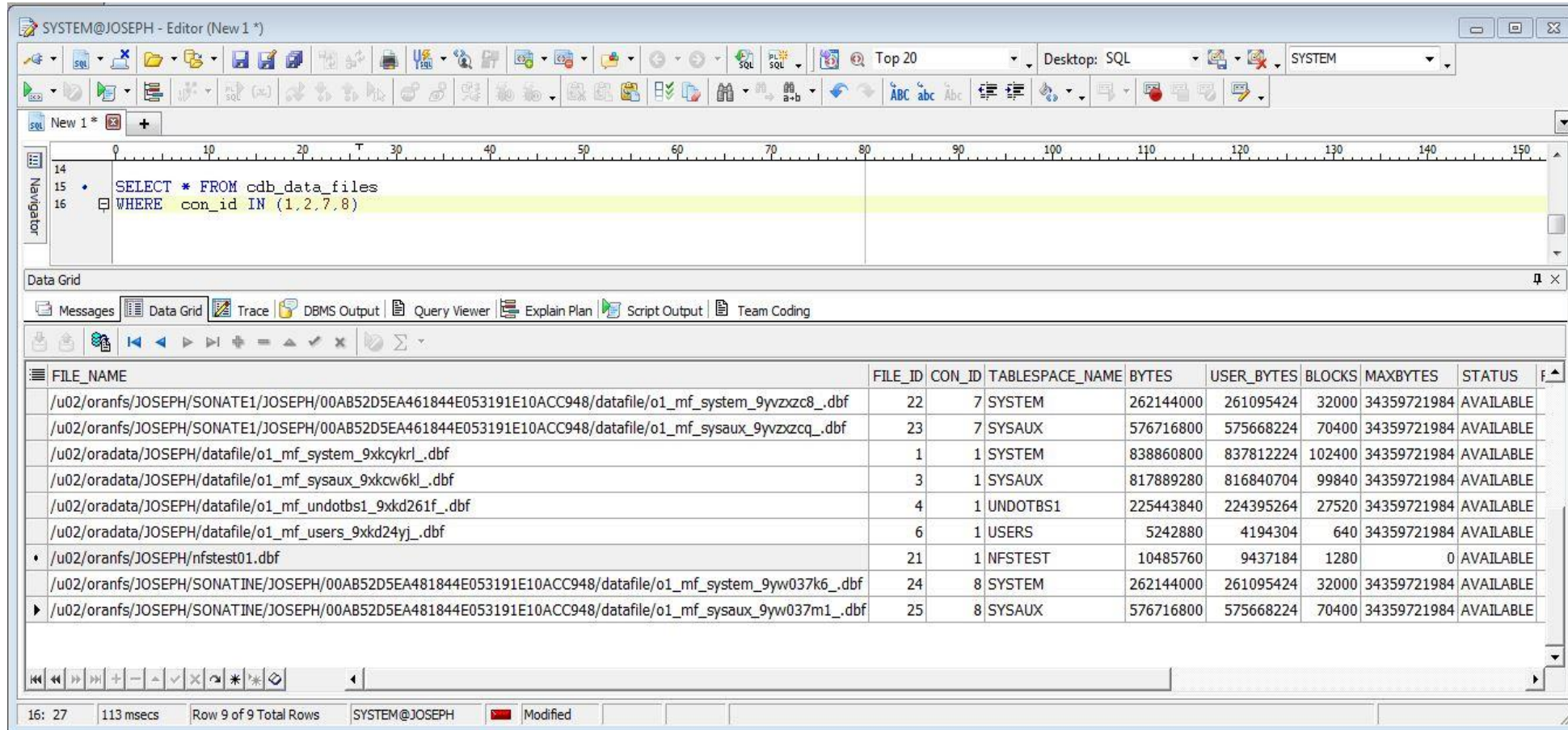
```
SQL> ALTER SYSTEM SET clonedb=true SCOPE=spfile;  
SQL> SHUTDOWN IMMEDIATE;  
SQL> STARTUP;
```


- Use Parameter `CREATE_FILE_DEST` for individual file locations
- Directory must exist (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 SYSTEM user. The query is:

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

The results are displayed in a Data Grid with the following columns: FILE_NAME, FILE_ID, CON_ID, TABLESPACE_NAME, BYTES, USER_BYTES, BLOCKS, MAXBYTES, STATUS, and F. The data shows files for CON_ID 7 and 8, all with a status of AVAILABLE.

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
```

- **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 ?)**

Next presentations

- 18.11. 10:00h Unconference “Oracle Standard Edition” (M. Paege, J. Ahrends)
- 18.11. 11:00h „RAC Battle“ (M. Klier, B. Rost, J. Ahrends)
- 18.11. 12:00h Database Community „Replication“ (J. Ahrends)
- 18.11. 13:00h „Partitioning für mehr Performance“ (S. Winkler)
- 20.11. DOAG Training
Oracle 12c Datenbank Installation und Konfiguration
(J. Ahrends, S. Winkler)



Questions?

Johannes Ahrends

Johannes.ahrends@carajandb.com

blog.carajandb.com

Twitter: @carajandb

DOAG
Deutsche ORACLE-Anwendergruppe e.V.