



Multitenant Database Fluch oder Segen?

Johannes Ahrends
CarajanDB GmbH



ORACLE
ACE

- Experten mit über 30 Jahren Oracle Erfahrung
- Firmensitz in Erftstadt bei Köln
- Spezialisten für
 - Oracle Datenbank Administration
 - Hochverfügbarkeit (RAC, Data Guard, Failsafe, etc.)
 - Einsatz der Oracle Standard Edition
 - Oracle Migrationen (HW, Unicode, Standard Edition)
 - Replikation
 - Performance Tuning
 - Datenbank Cloning (Delphix, Actifio, CloneDB)
- Fernwartung
- Schulung und Workshops (Oracle, Toad)



- **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)



- 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](https://twitter.com/carajandb)
- Facebook: [johannes.ahrends](https://www.facebook.com/johannes.ahrends)
- Blogs:
 - blog.carajandb.com
 - www.toadworld.com

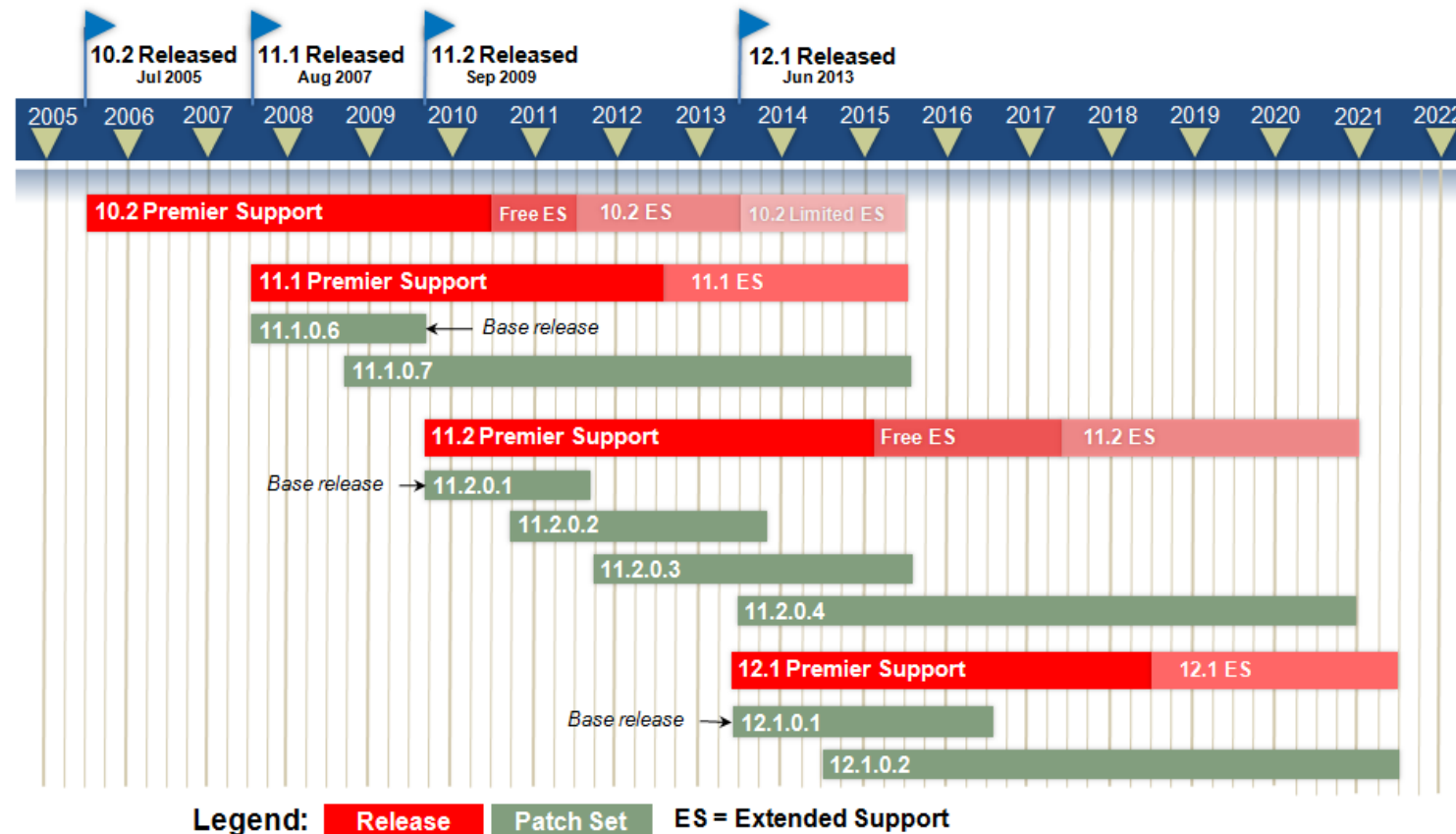


Multitenant Database

“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, Kapitel 8.1.1)

Release Schedule

- Quelle: MOS Release Schedule of Current Database Releases (Doc ID 742060.1)



- 12.1.0.1 → Seit Juni 2013
- 12.1.0.2 → Seit Juli 2014
- 12.2.0.1 → ???

- Datenbank:
 - SYSTEM, SYSAUX, UNDO, TEMP Tablespace
 - Redolog Dateien
 - Archivierte Redolog Dateien
 - Controlfile
- Instanz:
 - SGA
 - PGA
 - Background Prozesse (einfache DB → 33 Prozesse)

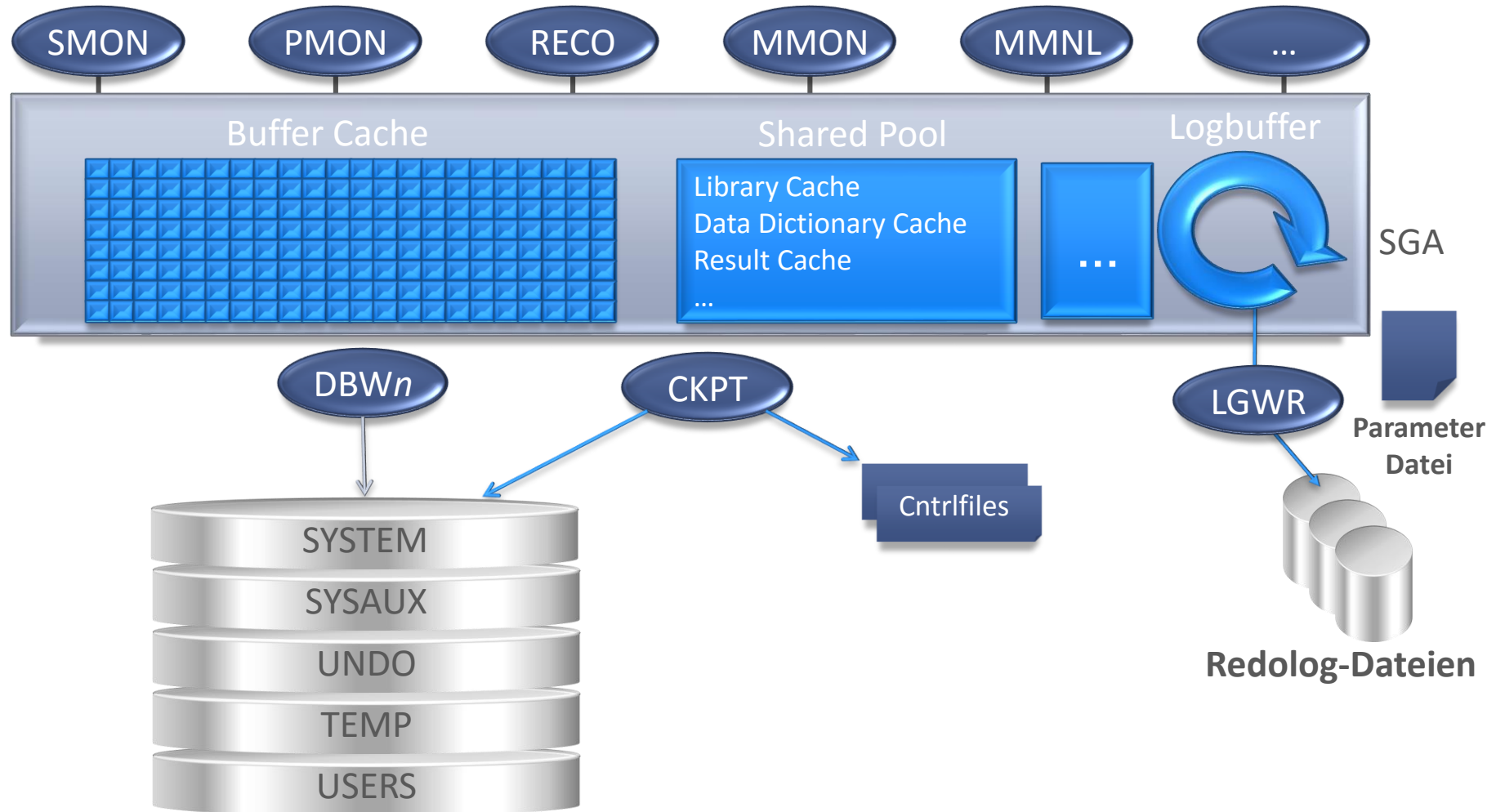
- Minimum 2 GB Memory pro Instanz
- Minimum 2 CPUs pro Instanz
- Tom Kyte:
„the only reasonable, correct number of instances on a production machine is one”

Pluggable Database

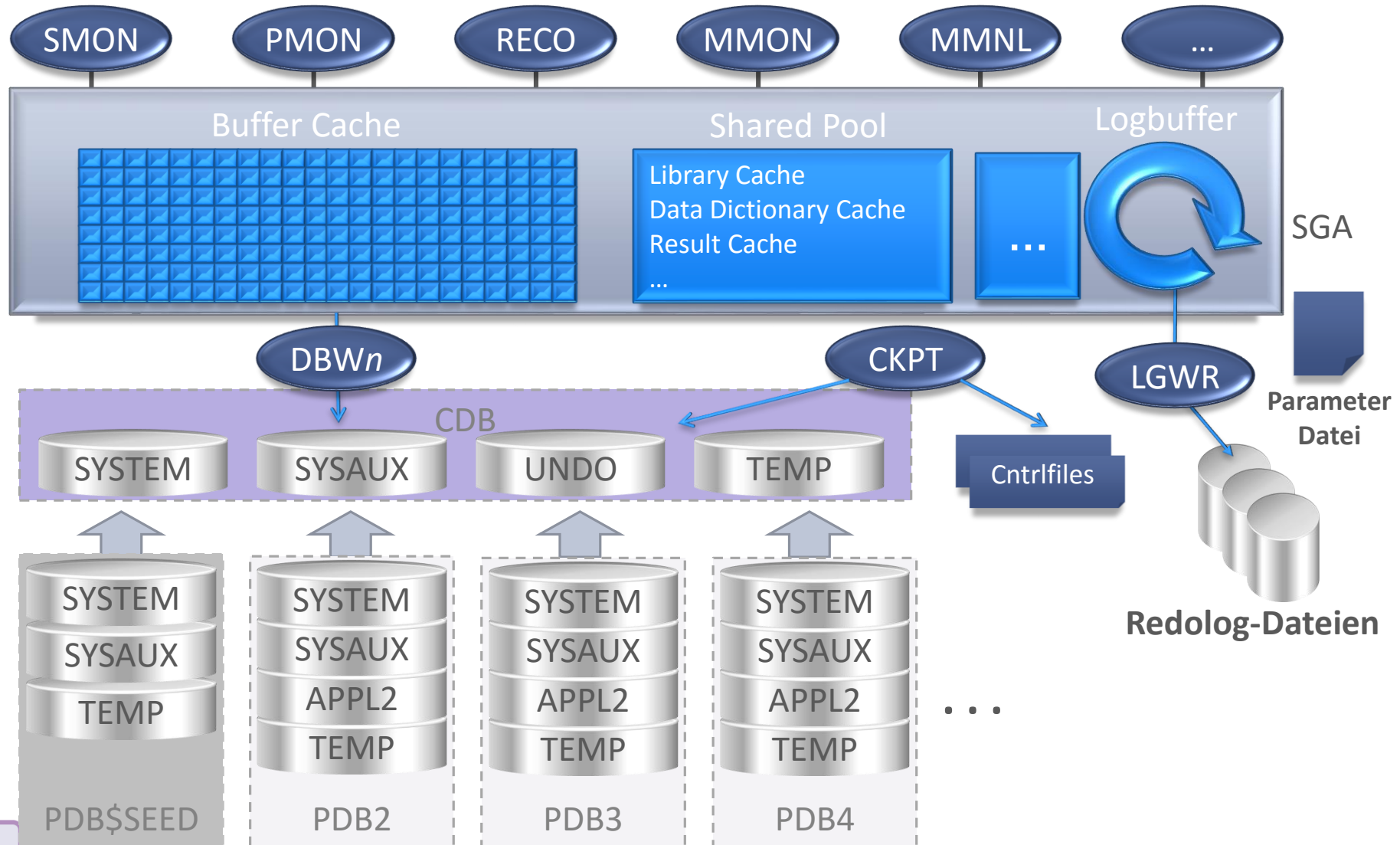


- Konsolidierung von Datenbanken
- Eine gemeinsame Instanz
- Ein gemeinsames Undo Management
- Ein gemeinsames Redo Management
- Separate Tablespace Verwaltung
- Separate Schema Verwaltung

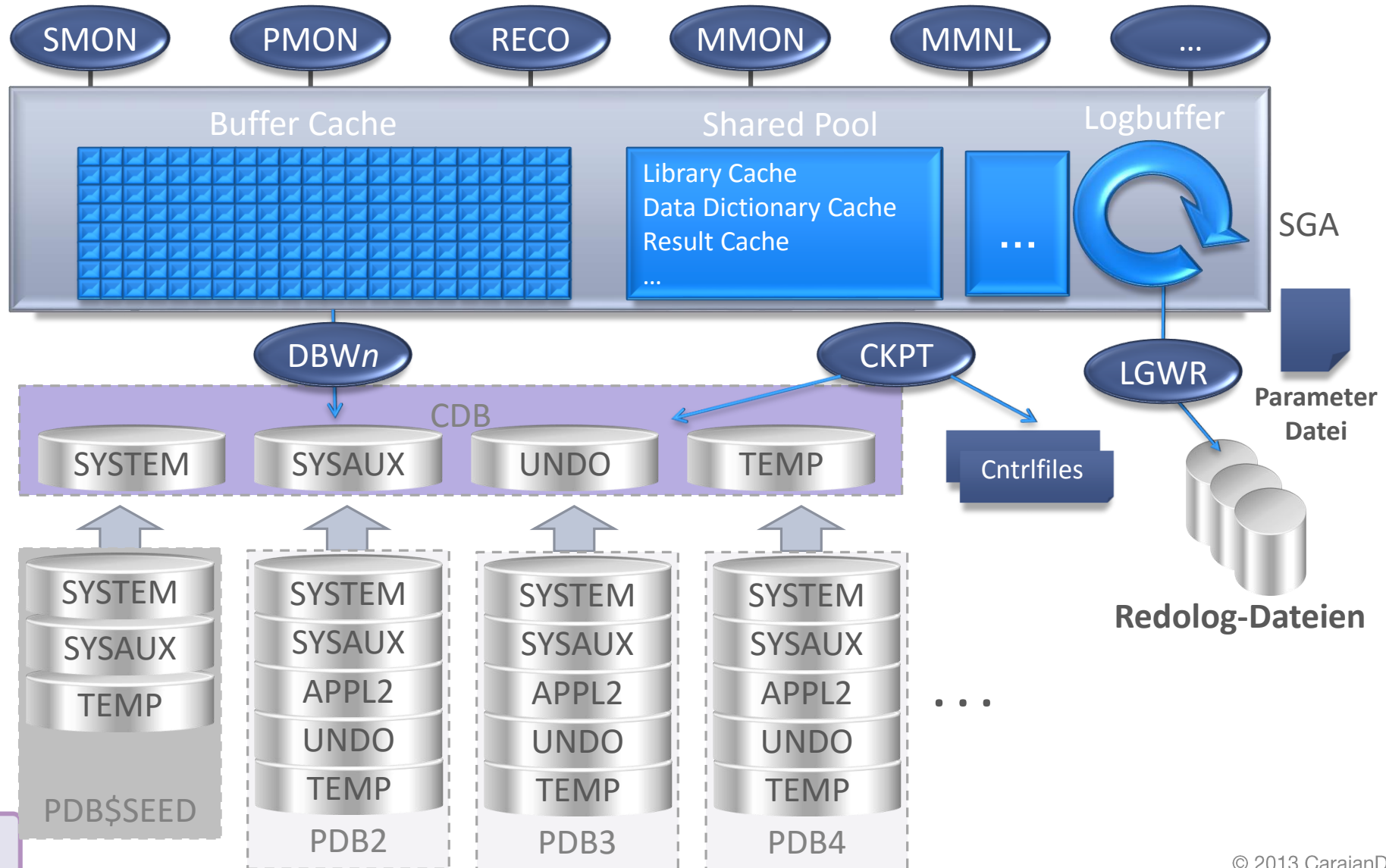
„Klassische“ Datenbank



Multitenant Database 12.1



Multitenant Database 12.2



- Multitenant Database Option verfügbar seit 2013 (12.1.0.1)
- Implementierungsrate gering
- Kostenpflichtige Option (17.500,00 \$ pro Prozessor)
- NON-CDB Architektur deprecated mit Version 12.1.0.2 (seit Mitte 2014)
 - Standardsupport bis Juli 2018
 - Extended Support bis Juli 2021
- Alle PDBs teilen sich eine Instanz (außer RAC)
- Alle PDBs teilen sich ein Undo (bis 12.2)
- Alle PDBs teilen sich ein Redo
- DBAs müssen geschult werden
- Skripte funktionieren nicht mehr



Skripte

- Beispiel:

```
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

- Ist das richtig? – Es kommt darauf an!
- Es handelt sich hier wahrscheinlich um eine CDB, weil der UNDO-Tablespace angezeigt wird

- Ist das hier besser?

```
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

- Die MBYTES scheinen okay aber MAXMBYTES sehen merkwürdig aus!

Vielleicht passt das hier?

```
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

Example 43-7 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 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	SYS_AUX	/disk1/oracle/dbs/pdbseed/cdb1_ax.f
2	PDB\$SEED	5	SYSTEM	/disk1/oracle/dbs/pdbseed/cdb1_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

- 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

Das hier sollte „richtig“ sein!

```
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



Flashback Database

- Selten benutzt – beruhigt aber das Gewissen
- Praxis: regelmäßige DataPump exports „Falls etwas schief geht“
 - Täglich GB bis TB für die Dumpfiles
 - Ggf mehrere Stunden für einen Export
- Alternative → Flashback Database
- Keine Alternative wenn
 - Schemakonsolidierung
 - Pluggable Database

- In Zukunft kann eine PDB ein eigenes UNDO Management haben
„A CDB can run in different undo modes. You can configure a CDB to have one active undo tablespace for the entire CDB or a separate undo tablespace for each container in the CDB. You can specify the undo mode during CDB creation, and you change the undo mode after the CDB is created.“
- Damit ist ein Flashback Pluggable Database möglich!



Multiple Character Set

- Ein gemeinsamer Zeichensatz für CDB und alle zugehörigen PDBs
 - Daher oftmals mehr als eine CDB erforderlich (WE8ISO8859P15 vs. AL32UTF8)
- Zukünftig
 - Support für unterschiedliche Zeichensätze in einer CDB
 - Voraussetzung: CDB mit AL32UTF8 angelegt

„When the character set of the root is AL32UTF8, PDBs that are plugged into the CDB or cloned can have a different character set than the root. PDBs that are created from the seed inherit the AL32UTF8 from it, but you can migrate the PDB to a different character set.”
- Kein direktes Anlegen einer PDB mit anderem Zeichensatz möglich
- Praktisch: Eigenen Template bauen und dann einhängen

Fehlermeldung (derzeit)

```
SQL> SELECT * from PDB_PLUG_IN_VIOLATIONS;
```

```
TIME
```

```
-----
```

```
NAME
```

```
CAUSE
```

```
-----
```

```
TYPE
```

```
ERROR_NUMBER
```

```
LINE
```

```
-----
```

```
MESSAGE
```

```
-----
```

```
-----STATUS
```

```
-----
```

```
ACTION
```

```
-----
```

```
-----
```

```
08-MAR-16 12.06.32.938789 PM
```

```
KARELIA_WE8
```

```
PDB not Unicode
```

```
WARNING
```

```
0
```

```
1
```

```
Character set mismatch: PDB character set WE8ISO8859P15. CDB character set AL32UTF8.
```

```
PENDING
```

```
Oracle recommends using Unicode (AL32UTF8) character set for the database. Consider migrating the database to Unicode.
```



Ressourcennutzung

- Nur eine SGA → Keine Einschränkung hinsichtlich Buffer-Cache, Shared-Pool, etc.
- Nur ein Set an Redologs und Undo-Tablespace
- Keine Einschränkung bezüglich I/O
- Ressourceneinschränkungen nur über den Resource Manager (z.B: CPU, Sessions, execution Time)
- Einschränkung der Gesamtgröße einer PDB, z.B:
 - `ALTER PLUGGABLE DATABASE
STORAGE (MAXSIZE xG) ;`
- **Eigener TEMP-TABLESPACE**

- Instance Caging
 - CPU Zuweisung über `cpu_count`
- Voraussetzung
 - Resource Manager eingeschaltet
- `cpu_count` ist Instanz-spezifisch, d.h. setzen pro PDB ist nicht möglich

- **CDB Resource Plans**
 - Aufteilung von Ressourcen auf einzelne PDBs
 - Verteilung der gesamten Ressourcen über Shares
- **PDB Resource Plans**
 - Aufteilung von Ressourcen innerhalb einer PDB

- Beschränkung über Shares
 - Verhältnis der PDBs zueinander
- Derzeit nur
 - CPU
 - Parallelisierungsgrad
- Kontrolliert über „Directives“

Erstellung von Plan Directives

```
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;
```

- Wie kann man verhindern, dass eine PDB alle Ressourcen verschwendet?
 - Oracle 12.1 → CDB Resource Manager
 - Gilt nur für CPU und Parallelisierung
 - Oracle 12.2 → Oracle Parameter auf PDB Ebene
 - SGA_TARGET
 - SGA_MIN_SIZE
 - SGA_MAX_SIZE
 - DB_CACHE_SIZE
 - SHARED_POOL_SIZE
 - PGA_AGGREGATE_TARGET
 - PGA_AGGREGATE_LIMIT

- **Multitenant Database ist ein Fluch!**
 - Administratoren müssen neu geschult werden
 - Skripte funktionieren nicht mehr
 - Hohe Kosten
- **Multitenant Database ist ein Segen!**
 - Kosten für die Option können sich schnell amortisieren
 - Bessere Ausnutzung der Ressourcen
 - Schnelle Erstellung von Datenbanken
 - Segregation of Duties

Weitere Informationen

- 25. Jan. 16:30 Uhr: Toad User Forum Webseminar: Tuning mit Toad
- 02. bis 03. Feb. 2017: Noon2Noon (Mainz) „Upgrade nach Oracle 12c“
- 30. bis 31. Mai 2017: DOAG Datenbank Konferenz in Düsseldorf (CfP jetzt offen)

Eine kleine Weihnachtsgeschichte: Weihnachten 12.2 (=24): blog.carajandb.com





Fragen?