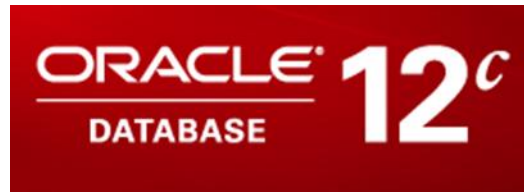




# Top 5 Features in



# 15 Jahre MuniQSoft GmbH

## ◆ Tätigkeitsbereiche:

- ▶ Oracle Support  
Hotline: Mo-Fr 8.00 – 18.00 Uhr  
Erweiterung um Rufbereitschaft auch am Wochenende möglich
- ▶ Oracle IT-Consulting & Services, Schwerpunkt auch RAC
- ▶ Oracle Schulungen (SQL, PL/SQL, DBA, APEX, B&R, MySQL...51  
verschiedene Schulungen, gerne auch Inhouse )  
**Seit Sommer 2013: Pimp up my Standard Edition**
- ▶ Software-Lösungen
- ▶ Oracle Lizenzen

MuniQSoft GmbH  
**Schulungszentrum**  
Grünwalder Weg 13a  
82008 Unterhaching  
Tel.: 089 / 679090 40



MuniQSoft GmbH  
**IT-Consulting, Support,  
& Software-Lösungen**  
Witneystr. 1  
82008 Unterhaching  
Tel.: 089 / 6228 6789 0

# MuniQSoft Tipps & Tricks

◆ <http://www.muniqsoft.de/tipps.htm>



## Oracle Tipps und Tricks

Wir haben für Sie eine kleine Auswahl an interessanten Oracle-Informationen zusammengestellt.

[Neuerungen von Oracle 10g Rel 2](#)

[Neuerungen von Oracle 11g Rel 1](#)

[Neuerungen von Oracle 11g Rel 2](#)

[Neuerungen von Oracle 12c](#)

Sie möchten mehr über die New Features von Oracle erfahren? Wir haben hier eine Auswahl für Sie bereit gestellt.

[Oracle DBA](#)

[Oracle SQL](#)

[Oracle PL/SQL](#)

[Oracle APEX](#)

[Oracle Skripte](#)

Hier finden Sie Infos zu den wichtigsten Oracle Themengebieten.

[Tipps des Monats](#)

Wir bieten Ihnen jeden Monat interessante Tipps zum Thema Oracle.

[Tipps & Tricks von A - Z](#)

Hier finden Sie alle unsere Tipps & Tricks auf einen Blick.

[Glossar](#)

Wir haben hier eine Sammlung von typischen Oracle-Begriffen mit einer entsprechenden Erklärung auf deutsch. Die Sammlung wird ständig erweitert.

# Platz 5: Neue VARCHAR2 Spaltenlängen

- ◆ **Ab Version 12.1 kann für VARCHAR2 und Raw die Länge 32767 Bytes verwendet werden**
- ◆ **Dazu muss die DB einmalig umgestellt werden**
- ◆ **Hinweis:**
  - ▶ **Function Based Indizes werden ungültig, wenn Sie auf einer Spalte basieren, die länger sein kann als 87% eines Index Blocks**

# Umstellung einer DB auf neue Spaltenlänge

- ◆ **1. Schließen Sie die Datenbank**
  - ▶ `shutdown immediate`
- ◆ **2. DB starten im Upgrade Modus**
  - ▶ `startup upgrade`
- ◆ **3. Einstellung von MAX\_STRING\_SIZE ändern**
  - ▶ `ALTER SYSTEM SET MAX_STRING_SIZE =extended;`
- ◆ **4. Starten Sie das Skript**
  - ▶ `@?/rdbms/admin/ut132k.sql`
- ◆ **5. DB neustarten**
  - ▶ `shutdown immediate`
  - ▶ `startup`

# VARCHAR2(32767)

◆ Nun können Sie die neue Spaltenlänge nutzen:

◆ Beispiel:

- ▶ `CREATE TABLE t (  
id NUMBER,  
text VARCHAR2(32767));`
- ▶ `INSERT INTO t (text)  
VALUES (lpad('#', 32000, 'X'));`
- ▶ `SELECT length(text) FROM t;`
- ▶ => 32000

# Nebenwirkungen der neuen Länge

- ◆ Intern verwendet Oracle die CLOB Technik um die Varchar2 Länge von 4k-32K darzustellen.
- ◆ Folgende Einschränkungen sind uns aufgefallen:
  - ▶ `SELECT * FROM t_with_clob_col  
UNION  
SELECT * FROM t_with_clob_col;`
  - ▶ SQL\*Plus 11.2 / SQL Developer 4.0 (Beta)
  - ▶ => ORA-24920: Spaltengröße für Client zu groß
  - ▶ SQL\*Plus 12.1 kann mit dem Befehl umgehen !
  - ▶ Varchar2(32k) Spalten via Database Links: gleiches Problem

# Nebenwirkungen der neuen Länge (f)

## ◆ Ein Index auf einer 32K Spalte

▶ `CREATE INDEX i_text_ix ON t(text);`

## ◆ Endet mit einem

▶ `ORA-01450: Maximale Schlüssellänge (6398) überschritten`



# Platz 4: IDENTITY Spalte

- ◆ Sie können auch einen automatischen Sequenz-Generator zum Füllen von PK/UK Spalten verwenden

- ▶ 

```
CREATE TABLE t (id NUMBER  
GENERATED AS IDENTITY,  
text VARCHAR2(10));  
INSERT INTO t (text) VALUES ('X');
```

ID	TEXT
1	III

- ▶ 

```
CREATE TABLE t (id NUMBER  
GENERATED BY DEFAULT AS IDENTITY  
(START WITH 100 INCREMENT BY 10));
```

# IDENTITY Optionen

- ◆ Im Prinzip können die gleichen Optionen, wie bei einer Sequenz verwendet werden:
- ◆ ... ( START WITH ( <int> | LIMIT VALUE ) | INCREMENT BY <int> |  
( MAXVALUE integer | NOMAXVALUE ) |  
( MINVALUE integer | NOMINVALUE ) |  
( CYCLE | NOCYCLE ) |  
( CACHE integer | NOCACHE ) |  
( ORDER | NOORDER ) }
- ◆ Zusätzlich gibt es noch:
  - ▶ ALWAYS: Spalte darf nur durch IDENTITY gefüllt werden. Manuelles Füllen bei Insert oder Update ist nicht erlaubt
  - ▶ BY DEFAULT [ON NULL]: Spalte wird per Default durch IDENTITY gefüllt, manuelles Füllen ist jedoch möglich.  
ON NULL: Wenn ein Insert für die Spalte Null ergibt, wird durch IDENTITY gefüllt.

# Tabellenänderungen: IDENTITY Spalte

## ◆ Wie funktioniert's?

- ▶ Oracle legt eine Sequenz im Schema der Tabelle an (z.B. mit Namen ISEQ\$\$\_12345 und als Default mit Cache=20)
- ▶ Welche Tabelle verwendet IDENTITY Spalten :

- ```
SELECT owner, table_name, has_identity
FROM all_tables WHERE owner='MARCO';
```

| OWNER | TABLE_NAME | HAS_IDENTITY |
|-------|------------|--------------|
| MARCO | T          | YES          |

- ```
SELECT owner, table_name, column_name,
identity_column
FROM all_tab_columns
WHERE owner='MARCO';
```

OWNER	TABLE_NAME	COLUMN_NAME	IDENTITY_COLUMN
MARCO	T	ID	YES
MARCO	T	TEXT	NO

# IDENTITY Wartung

## ◆ Bestehende IDENTITY löschen (Spalte bleibt erhalten):

- ▶ `ALTER TABLE <schema>.<TABLE> MODIFY  
(<ident_col> DROP IDENTITY);`

## ◆ Beispiel:

- ▶ `ALTER TABLE t MODIFY  
(id DROP IDENTITY);`

## ◆ IDENTITY nachträglich (inkl. Spalte) hinzufügen:

- ▶ `ALTER TABLE T ADD  
(ID_NEU NUMBER GENERATED AS IDENTITY  
(START WITH 100));`

Hinweis: Eine IDENTITY nachträglich zu einer existierenden Spalte hinzuzufügen, ist nicht vorgesehen !

# Platz 3: Unified Auditing

- ◆ Das Feature **UNIFIED AUDITING** fasst sämtliche Audit-Informationen an einer einzigen Stelle zusammen
- ◆ Dadurch ergeben sich einige Vorteile
- ◆ Keine Parametereinstellung mehr notwendig
  - ▶ Bessere Übersicht und einfachere Sichtung der Audit-Informationen
  - ▶ Bessere Performance, was die Überwachung betrifft
- ◆ Ermittlung, ob Option bereits eingeschaltet (TRUE oder FALSE):
  - ▶ 

```
SELECT * FROM v$option  
WHERE parameter = 'Unified Auditing';
```

# Unified Auditing (f)

- ◆ Bei FALSE muss die Nutzung explizit noch aktiviert werden.
  - ▶ Dazu Instanz und Listener herunterfahren
  - ▶ Unter Unix Wechsel in `$ORACLE_HOME/rdbms/lib`, dann
    - `make -f ins_rdbms.mk uniaud_on ioracle ORACLE_HOME=$ORACLE_HOME`
  - ▶ Unter Windows in `%ORACLE_HOME%\bin` folgende Datei umbenennen
    - `orauniaud12.dll.dbl` in `orauniaud12.dll`
  - ▶ Listener und Instanz wieder starten

# Unified Auditing Speichermethoden

- ◆ Der Initialisierungsparameter **UNIFIED\_AUDIT\_SGA\_QUEUE\_SIZE** regelt die Speichermenge, die der Hauptspeicher für die Audit Einträge zur Verfügung stellt.
  - ▶ Speichergröße 1-30MB
- ◆ Benutzer die Unified Auditing administrieren wollen, benötigen das Recht **AUDIT SYSTEM** oder die Rolle **AUDIT ADMIN**.
- ◆ Die Auditdaten werden zuerst gecached und periodisch in Tabellen gespeichert.
- ◆ Bei einem Shutdown abort könnten jedoch Einträge verloren gehen, deswegen unterstützt Oracle zwei Modi:
  - ▶ Immediate (sofort schreiben)
  - ▶ Queued Write (verzögert schreiben)

# Unified Auditing Syntax

## ◆ Syntax

```
▶ CREATE AUDIT POLICY policy
  [ privilege_audit_clause ]
  [ standard_or_component_clause ]
  [ role_audit_clause ]
  [ WHEN 'audit_condition' EVALUATE PER
    { STATEMENT | SESSION | INSTANCE } ]
  [ CONTAINER = { ALL | CURRENT } ] ;
```



# Unified Auditing Syntax (f)

## ◆ EVALUATE PER ...

### ▶ STATEMENT

- Für jedes ausgeführte Statement wird ein Auditeintrag erstellt

### ▶ SESSION

- Innerhalb wird nur einmal für ein ausgeführtes Statement ein Auditeintrag erzeugt

### ▶ INSTANCE

- Solange die Instanz läuft, wird das Vorgehen nur einmal aufgezeichnet

## ◆ CONTAINER =

### ▶ ALL (alle container einer Plugable Database)

### ▶ CURRENT (nur der aktuelle Container)

# Beispiele zur WHEN-Klausel

## ◆ SELECT ANY TABLE und CREATE VIEW für zwei Benutzer überwachen:

```
▶ CREATE AUDIT POLICY osusers_tab_view_pol
  PRIVILEGES SELECT ANY TABLE, CREATE VIEW
  WHEN q'!SYS_CONTEXT ('USERENV', 'OS_USER')
  IN ('MARCO', 'HANS')!'
  EVALUATE PER SESSION;
```

## ◆ Alle Anmeldungen durch SQL\*Plus überwachen:

```
▶ CREATE AUDIT POLICY sqlplus_logon_pol
  ACTIONS LOGON
  WHEN q'!INSTR(UPPER(SYS_CONTEXT('USERENV',
  'CLIENT_PROGRAM_NAME')), 'SQLPLUS') > 0!'
  EVALUATE PER SESSION;
```

# Unified Auditing Beispiele

## ◆ Beispiele für Objekt-Privilegien

- ▶ **CREATE AUDIT POLICY** `dml_poli` ACTIONS  
DELETE on scott.emp,  
INSERT on scott.emp,  
UPDATE on scott.emp,  
ALL on scott.dept;
- ▶ **AUDIT POLICY** `dml_poli` BY scott;

# Unified Auditing Beispiele (f)

## ◆ Beispiele für System-Privilegien

- ▶ `CREATE AUDIT POLICY table_poli PRIVILEGES  
CREATE ANY TABLE,  
DROP ANY TABLE;`
- ▶ `AUDIT POLICY table_poli BY scott;`

# Interessante Privilegien für das Audit

- ◆ CREATE USER
- ◆ ALTER USER
- ◆ CREATE DATABASE LINK
- ◆ ALTER DATABASE LINK
- ◆ ALTER SESSION
- ◆ ALTER SYSTEM
- ◆ AUDIT ANY
- ◆ AUDIT SYSTEM
- ◆ BECOME USER
- ◆ CREATE ANY DIRECTORY
- ◆ CREATE ANY JOB
- ◆ CREATE EXTERNAL JOB
- ◆ CREATE JOB
- ◆ CREATE LIBRARY
- ◆ CREATE PUBLIC SYNONYM
- ◆ DROP TABLESPACE
- ◆ DROP ANY TABLE
- ◆ DROP USER
- ◆ EXEMPT DDL REDACTION POLICY
- ◆ EXEMPT DML REDACTION POLICY
- ◆ EXEMPT IDENTITY POLICY
- ◆ EXEMPT REDACTION POLICY
- ◆ EXPORT FULL DATABASE
- ◆ GRANT ANY OBJECT PRIVILEGE
- ◆ GRANT ANY PRIVILEGE

# Unified Auditing Beispiele (ff)

## ◆ Beispiele für Packages

- ▶ Überwachen, ob SYS Rechte an UTL\_FILE/UTL\_TCP/UTL\_SMTP vergibt:
- ▶ 

```
CREATE AUDIT POLICY dbms_utl_grants ACTIONS  
GRANT ON UTL_FILE,  
GRANT ON UTL_TCP  
GRANT ON UTL_SMTP;
```
- ▶ 

```
AUDIT POLICY dbms_utl_grants BY SYS;
```

# Audit auswerten

```
◆ SELECT TO_CHAR(EVENT_TIMESTAMP, 'DD.MM.YY Hh24:MI:SS') AS  
EVENT_TIMESTAMP, OS_USERNAME, DBUSERNAME,  
substr(client_program_name,1,30) client_prg,  
action_name, object_name, sql_text,  
system_privilege_used  
FROM UNIFIED_AUDIT_TRAIL  
ORDER BY 1 desc;
```

EVENT_TIMESTAMP	OS_USERNAME	DBUSERNAME	CLIENT_PRG	ACTION_NAME	OBJECT_NAME	SQL_TEXT
03.07.13 02:05:53	marco	SYS	SQL Developer	AUDIT	TABLE_POLI	AUDIT POLICY TABLE
03.07.13 02:03:55	marco	SCOTT	SQL Developer	LOGON	(null)	(null)
03.07.13 02:03:46	marco	SYS	SQL Developer	CREATE AUDIT POLICY	TABLE_POLI	CREATE AUDIT POLICY
03.07.13 02:02:56	marco	SYS	SQL Developer	LOGON	(null)	(null)
03.07.13 01:53:39	oracle	SYS	sqlplus@localhost.localdomain	ALTER DATABASE	(null)	ALTER DATABASE OPEN
03.07.13 01:53:29	oracle	SYS	sqlplus@localhost.localdomain	ALTER SESSION	(null)	ALTER SESSION SET M
03.07.13 01:53:29	oracle	SYS	sqlplus@localhost.localdomain	LOGON	(null)	(null)
03.07.13 01:53:16	oracle	SYS	(null)	LOGON	(null)	CONNECT

# Platz 2: Datendateien verschieben

◆ Datendateien können online verschoben werden

◆ Syntax:

```
▶ ALTER DATABASE MOVE DATAFILE ( 'filename' |  
  'ASM_filename' | file_number )  
  [ TO ( 'filename' | 'ASM_filename' ) ]  
  [ REUSE ] /* vorhandene Datei überschreiben */  
  [ KEEP ]; /* alte Version behalten */
```

◆ Beispiel:

```
▶ ALTER DATABASE MOVE DATAFILE 4  
  TO '/u01/app/oracle/oradata/user02.dbf'  
  KEEP;
```

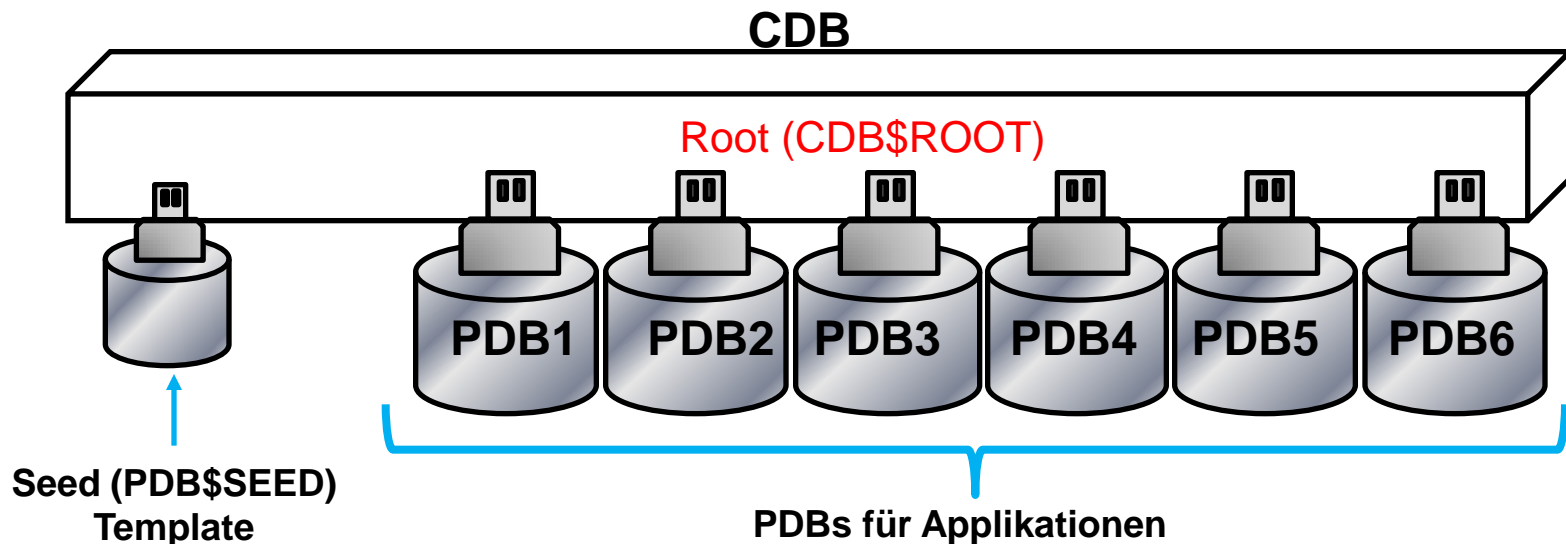


# Platz 1: Pluggable Database

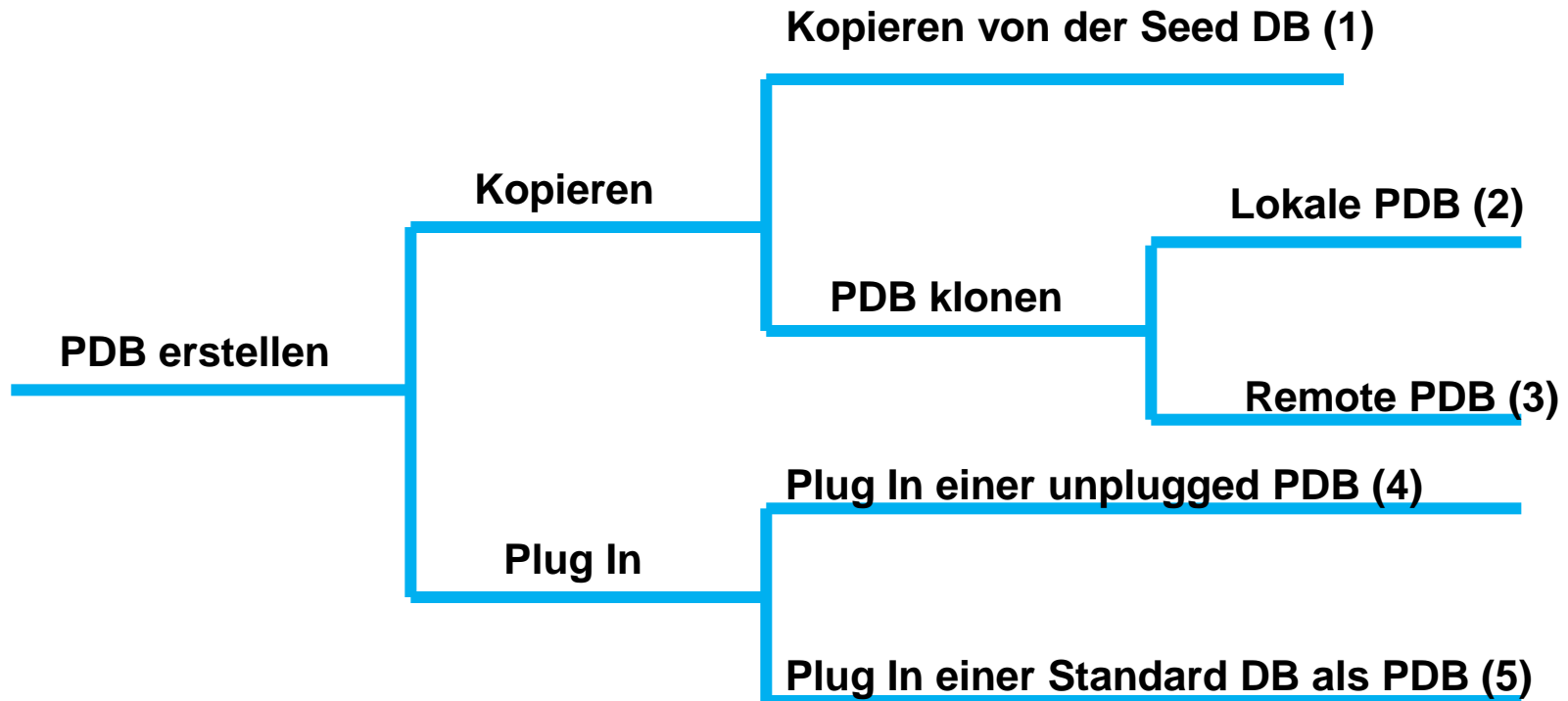
- ◆ **Pluggable Database ist eine Containerdatenbank (CDB) für 0, 1 oder mehrere Pluggable Databases (PDB)**
- ◆ **Jede CDB besteht aus**
  - ▶ **Genau einem ROOT (CDB\$ROOT)**
    - **Standard Benutzer (der in jedem Container bekannt ist) Hier stehen die Metadaten und**
  - ▶ **Genau eine Seed PDB (PDB\$SEED)**
    - **Template zur Erstellung von neuen PDBs. Objekte in PDB\$SEED dürfen NICHT verändert werden.**
  - ▶ **Null-n PDBs**
    - **Benutzerdefinierter und erstellter Container für Daten oder Code (evtl. eigens angelegt für versch. Applikationen)**

# Generelles zu Pluggable Database (f)

- ◆ Mehrere PDBs können in einer einzelnen CDB liegen
- ◆ Sie können PDBs aus einer CDB entfernen und in eine andere verschieben
- ◆ Sie können PDBs klonen
- ◆ PDB ist kompatibel zu allen Oracle Optionen und RAC



# Erzeugung einer PDB



# Infos zu CDB

Bereich	CDB\$ROOT	PDB
OS Prozesse	CDB verwendet Background Prozesse für Root und alle PDB	-
Speicher	CDB verwendet eine SGA + PGA. Speicherverbrauch richtet sich nach der Summe der PDB's	-
Alert.log	Eine Alert.log für alle	Keine eigene
Controldatei	Eine Controldatei (optional gespiegelt) für die CDB	Keine eigenen Controldateien
Redologs	Es gibt die Redologs nur einmal	Keine eigenen
Archivelogs	Archivelogs nur einmal	Keine eigenen
SPFILE	Eine	Eigene Parameter pro PDB möglich

# Infos zu CDB

Bereich	CDB\$ROOT	PDB
Tablespaces	SYSTEM SYSAUX TEMP	SYSTEM SYSAUX TEMP <b>USERS</b> ...
Datendateien	Je TBS vorhanden	PDB können zusätzlich eigene Datendateien besitzen
Undo	Nur ein Undo Tablespace	

# PDB VerwaltungsvIEWS

- ◆ Neue Data Dictionary Views (z.B. *CDB\_PDBs*)
- ◆ Viele Views haben neue Spalten: *Con\_ID* (Container ID)
- ◆ Alte Data Dictionary Views (*DBA\_*, *All\_*, and *User\_*) wurden ergänzt mit dem *CDB\_* Prefix.
- ◆ Die *CDB\_VIEWS* besitzen die neue Spalte *Con\_ID*.
- ◆ Test, ob die angebundene DB eine CDB ist:
  - ▶ `SELECT count(*) FROM v$pdbs;`
  - ▶ 0 => Nein keine PDB, sondern normale DB
  - ▶ 1-n => CDB mit n PDB Datenbanken oder ...
  - ▶ `SELECT cdb FROM v$database;`
  - ▶ => YES

# V\$PDB und CDB\_PDBS

## ◆ Welche Container gibt es ?

▶ `SELECT con_id,name,open_mode,  
restricted, open_time  
FROM v$pdb;`

CON_ID	Beschreibung
0	Komplette CDB
1	CDB\$ROOT
2	PDB\$SEED
3-n	Weitere PDB's

CON_ID	NAME	OPEN_MODE	RESTRICTED	OPEN_TIME
2	PDB\$SEED	READ ONLY	NO	29.08.12 00:31:56,447
3	PDBMUSO1	READ WRITE	NO	29.08.12 00:33:30,628
4	PDBMUSO2	READ WRITE	NO	29.08.12 00:34:15,190

▶ `SELECT pdb_id,con_id,pdb_name,dbid,status  
FROM CDB PDBS;`

PDB_ID	CON_ID	PDB_NAME	DBID	STATUS
2	1	PDB\$SEED	4035195543	NORMAL
3	1	PDBMUSO1	1729526371	NORMAL
4	1	PDBMUSO2	2298948960	NORMAL

# V\$SESSION

## ◆ Wer ist in welchem Container angemeldet ?

▶ `SELECT con_id,sid,serial#,username,program  
FROM v$session  
ORDER BY 1 DESC;`

CON_ID	SID	SERIAL#	USERNAME	PROGRAM
1	73	205	SYS	SQL Developer
1	60	581	(null)	oracle@localhost.localdomain (J001)
1	55	265	(null)	oracle@localhost.localdomain (J002)
1	41	83	(null)	(null)
1	39	189	(null)	oracle@localhost.localdomain (J000)
0	7	3	(null)	oracle@localhost.localdomain (DIAG)
0	8	1	(null)	oracle@localhost.localdomain (OFSD)



# V\$DATAFILE

## ◆ Zu welchen Tablespace gehören Dateien im Container 2?

```
▶ SELECT tablespace_name, file_name  
FROM CDB_DATA_FILES WHERE con_id=2;
```

TABLESPACE_NAME	FILE_NAME
SYSTEM	/u01/app/oracle/oradata/orapdb/pdbseed/system01.dbf
SYSAUX	/u01/app/oracle/oradata/orapdb/pdbseed/sysaux01.dbf

## ◆ Das gleiche für Container 1:

TABLESPACE_NAME	FILE_NAME
USERS	/u01/app/oracle/oradata/O12PLUG/datafile/o1_mf_users_8o7wof2b_.dbf
UNDOTBS1	/u01/app/oracle/oradata/O12PLUG/datafile/o1_mf_undotbs1_8o7wog1c_.dbf
SYSAUX	/u01/app/oracle/oradata/O12PLUG/datafile/o1_mf_sysaux_8o7wkn6z_.dbf
SYSTEM	/u01/app/oracle/oradata/O12PLUG/datafile/o1_mf_system_8o7wmlfq_.dbf

# Aktuelle PDB anzeigen/wechseln

## ◆ Container in der Session wechseln:

- ▶ `ALTER SESSION SET CONTAINER=pdb1;`
- ▶ `CDB$ROOT` = Root Container
- ▶ `PDB$SEED` = Seed Container

## ◆ In welchem Container befindet sich die Session?

- ▶ `SELECT Sys_Context('Userenv', 'Con_Name') "current container" FROM dual;`

## ◆ Alle Container in SQL\*Plus anzeigen:

```
show pdbs
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	PDB3	READ WRITE	NO

# Wechsel des Containers

## ◆ Bedeutet:

- ▶ **Aktuelles Schema in Session ist das vom üblichen Benutzer im Container**
- ▶ **Login Trigger werden nicht gezündet**
  - **Workaround: AFTER DDL Trigger mit ALTER SESSION SET CONTAINER schreiben**
- ▶ **Package Statuswerte werden nicht über die Containergrenze hinweg zur Verfügung gestellt**
- ▶ **Transaktionen können nicht über mehrere Container gehandelt werden. Wenn eine Transaktion beginnt und der Container gewechselt wird, kann dort kein DML, Commit oder Rollback Anweisung durchgeführt werden, solange nicht in den Ursprungscontainer gewechselt wurde.**

# Benutzer anlegen in CDB

- ◆ Benutzer, die in einer CDB angelegt werden, müssen mit dem Prefix **c##** oder **C##** beginnen.
- ◆ Dadurch erhofft sich Oracle weniger Kollisionen mit bestehenden Benutzern in einer PDB 😊
- ◆ Oracle selbst verwendet einen undokumentierten Parameter um Benutzer (z.B. SYSTEM) für alle PDB's verfügbar zu machen:
  - ▶ `ALTER SESSION SET "_oracle_script"=true;`
  - ▶ `ALTER SESSION SET "_common_user_prefix"='';`
  - ▶ `CREATE USER marco IDENTIFIED BY marco;`Benutzer wurde erstellt.

# Benutzer erstellen

## ◆ Lokal in einer PDB:

```
SQL> create user scott identified by tiger container=all;
create user scott identified by tiger container=all
*
```

FEHLER in Zeile 1:  
ORA-65096: Ung<sup>3</sup>ltiger allgemeiner Benutzer- oder Rollename

## ◆ ALTER SESSION SET CONTAINER=PDB1;

```
SQL> create user scott identified by tiger container=current;
Benutzer wurde erstellt.
```

## ◆ In Root DB:

```
SQL> create user c##scott identified by tiger container=all;
create user c##scott identified by tiger container=all
*
```

FEHLER in Zeile 1:  
ORA-65050: Allgemeine DDLs nur in CDB\$ROOT zulässig

## ◆ ALTER SESSION SET CONTAINER=CDB\$ROOT;

```
SQL> create user c##scott identified by tiger container=all;
Benutzer wurde erstellt.
```

```
SQL> create user c##scott identified by tiger container=current;
create user c##scott identified by tiger container=current
*
```

FEHLER in Zeile 1:  
ORA-65094: Ung<sup>3</sup>ltiger lokaler Benutzer- oder Rollename

# Anmelden am Container

## ◆ Prüfen Sie im Listener, welche Container es gibt:

```
Services ■bersicht...  
Dienst "o12c" hat 1 Instanzen.  
  Instanz "o12c", Status READY, hat 1 Handler f³r diesen Dienst...  
Dienst "o12cXDB" hat 1 Instanzen.  
  Instanz "o12c", Status READY, hat 1 Handler f³r diesen Dienst...  
Dienst "pdb1" hat 1 Instanzen.
```

## ◆ Bei zwei Containern **pdb1** und **pdb2**:

- ▶ SQL> connect scott/TIGER@172.30.30.1/**pdb1**
- ▶ SQL> connect scott/TIGER@172.30.30.1/**pdb2**

## ◆ Root DB (**o12c**):

- ▶ SQL> connect scott/TIGER@172.30.30.1/**o12c**

# Automatische Anmeldung am Container

- ◆ Wenn Benutzer nur lokal in einem Container angelegt wurden:
- ◆ TNSNAMES.ORA `DEMO =`

```
(DESCRIPTION =  
  (ADDRESS_LIST =  
    (ADDRESS = (PROTOCOL = TCP) (HOST = 172.15.15.1) (PORT = 1521))  
  )  
  (CONNECT_DATA =  
    (SERVICE_NAME = demo)  
  )  
)
```
- ◆ Parameter local setzen:
- ◆ Win:
  - ▶ `set LOCAL=demo`
- ◆ Unix:
  - ▶ `export TWO_TASK=demo`
- ◆ Anmeldung:
  - ▶ `sqlplus system/sys`
  - ▶ => Container `demo`

# Database Links

## ◆ Sie können mittels Database Links auf die verschiedenen Container zugreifen:

- ▶ `CREATE DATABASE LINK pdb1  
CONNECT TO scott IDENTIFIED BY TIGER  
USING '172.30.30.30/pdb1';`
- ▶ `CREATE DATABASE LINK pdb2  
CONNECT TO scott IDENTIFIED BY TIGER  
USING '172.30.30.30/pdb2';`
- ▶ `SELECT * from emp@pdb1;`
- ▶ `SELECT * from emp@pdb2;`



# Rechte und Rollen

Recht/Rolle	Beschreibung
CDB_DBA (Rolle) SET CONTAINER (Recht)	Administration des CDB
PDB_DBA (Rolle) CONNECT (Rolle)	Administration der Container
CLONE PLUGGABLE DATABASE	Kloning eines Containers
CREATE PLUGGABLE DATABASE	Neuen Container erstellen

**Hinweis: nach dem Anlegen eines neuen Containers hat der Admin Benutzer die folgenden Rechte und Rollen:**

```
SQL> select * from session_privs;  
  
PRIVILEGE  
-----  
SET CONTAINER  
CREATE PLUGGABLE DATABASE  
CREATE SESSION
```

```
SQL> select * from session_roles;  
  
ROLE  
-----  
PDB_DBA
```

# 15 Jahre MuniQSoft GmbH

## ◆ Tätigkeitsbereiche:

- ▶ Oracle Support  
Hotline: Mo-Fr 8.00 – 18.00 Uhr  
Erweiterung um Rufbereitschaft auch am Wochenende möglich
- ▶ Oracle IT-Consulting & Services, Schwerpunkt auch RAC
- ▶ Oracle Schulungen (SQL, PL/SQL, DBA, APEX, B&R, MySQL...51  
verschiedene Schulungen, gerne auch Inhouse )  
**Seit Sommer 2013: Pimp up my Standard Edition**
- ▶ Software-Lösungen
- ▶ Oracle Lizenzen

MuniQSoft GmbH  
**Schulungszentrum**  
Grünwalder Weg 13a  
82008 Unterhaching  
Tel.: 089 / 679090 40



MuniQSoft GmbH  
**IT-Consulting, Support,  
& Software-Lösungen**  
Witneystr. 1  
82008 Unterhaching  
Tel.: 089 / 6228 6789 0