Oracle 12c Security New Features

Johannes Ahrends

CarajanDB GmbH

DOAG
Deutsche ORACLE-Anwendergruppe e.V.

CarajanDB

- **Experten mit über 20 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, Konsolidierung, Standard Edition)
  - Replikation (Goldengate, SharePlex, Dbvisit)
  - Performance Tuning
- **Schulung und Workshops (Oracle, Toad)**

… über mich

- **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
- **Leiter der DOAG SIG Database**
- **Hobbies:**
  - Drachen steigen lassen (Kiting) draußen wie drinnen (Indoorkiting)
  - Motorradfahren (nur draußen)

Kontakt

- **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: streetkiter**
- **Facebook: johannes.ahrends**
- **Blogs:**
  - www.carajandb.com/blogs
  - www.toadworld.com

Security 12c

DOAG
Deutsche ORACLE-Anwendergruppe e.V.

Security

- **Daten Verschlüsselung, Hashing und Verstecken**
  - Oracle Data Redaction
  - SHA-2 für Passwort Verschlüsselung und dbms_crypto
- **Sonstige Security Erweiterungen**
  - Unified Auditing
  - Data Guard Rolle (SYSDG)
  - RMAN Rolle (SYSBACKUP)
  - SELECT ANY DICTIONARY mit weniger Privilegien
  - Last Login Time in USER$ gespeichert
  - Password Komplexitätsprüfung

Security

- **Änderung der Resource Rolle**
  - Kein UNLIMITED TABLESPACE mehr!!!
- **Windows**
  - Support für Oracle Home User auf Windows
  - Support für Oracle Home User für Oracle Net Services
  - Named User Support für Oracle RAC Services auf Windows

Letzter Login eines Users

- **Spalte SPARE6 in Tabelle USER$**

```
SQL> SELECT NAME, spare6 FROM sys.USER$
  2  WHERE spare6 IS NOT NULL;

NAME                 SPARE6
-------------------- -------------------
SYSTEM               06.01.2014 14:00:19
BASIS                05.12.2013 14:34:08
DEMO                 06.01.2014 14:41:14
```

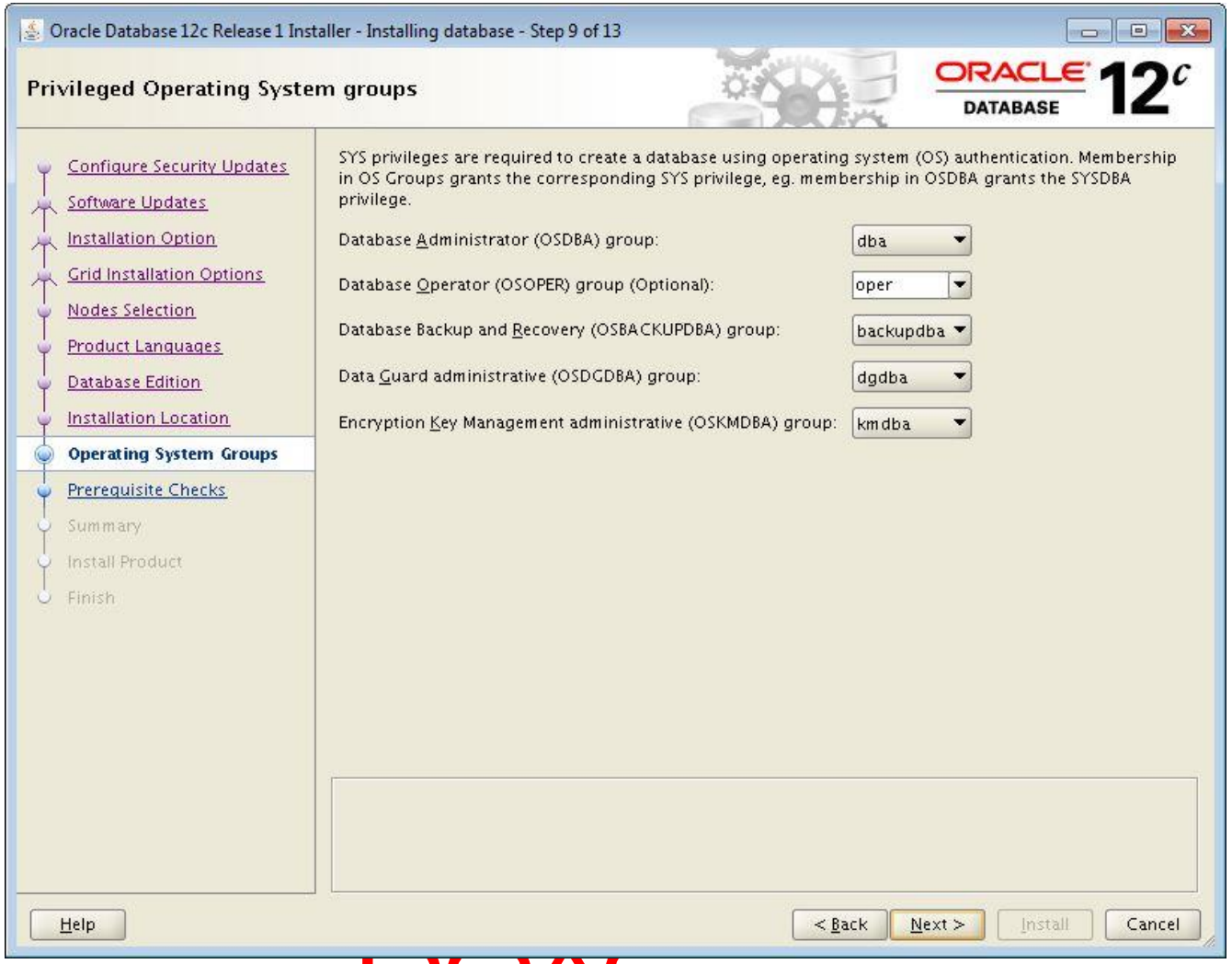
- **Anzeige bei Login mit SQL*Plus**

```
% sqlplus demo/demo

SQL*Plus: Release 12.1.0.1.0 Production on Mo Jan 6 16:21:49 2014

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Letzte erfolgreiche Anmeldezeit: Mo Jan 06 2014 15:41:14 +01:00

Verbunden mit:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options
```

Windows

- **Windows**
  - Support für Oracle Home User auf Windows
  - Support für Oracle Home User für Oracle Net Services
  - Named User Support für Oracle RAC Services auf Windows

DB Installation



Passwort Komplexitätsprüfung (1)

- **Drei Verifizierungspakete (im Script utlpwdmg.sql)**
  - `verify_function_11G` (schon seit Oracle 11g verfügbar)
  - `ora12c_verify_function`
    - mindestens 8 Zeichen
    - mindestens 1 Buchstabe und 1 Ziffer
    - Passworthistorie (mind. 3 unterschiedliche Zeichen)
    - Darf nicht enthalten: Datenbankname, Benutzername (auch revers), „oracle“, einfache Worte
    - Wird automatisch beim Default Profil gesetzt, wenn das Skript ausgeführt wird
  - `ora12c_strong_verify_function`
    - mindestens 9 Zeichen
    - mindestens 2 Großbuchstabe, 2 Kleinbuchstaben, 2 Ziffern, 2 Sonderzeichen
    - Passworthistorie (mind. 4 unterschiedliche Zeichen)

Passwort Komplexitätsprüfung (2)

- **Zwei neue Funktionen**
  - `string_distance`
    - überprüft die „Levenshtein“ Distanz zwischen zwei Zeichenketten

```
SQL> select string_distance('Test1','Zrdz2') from dual;

5
```

    - Überprüft die Komplexität eines existierenden Passworts

```
BEGIN
    IF  sys.complexity_check('MeinTest1',chars=>9,upper=>2,lower=>2,digit=>1,special=>1)
    THEN
        dbms_output.put_line( 'Erfüllt');
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        dbms_output.put_line( 'Nicht erfüllt '||SQLERRM);
END;
```

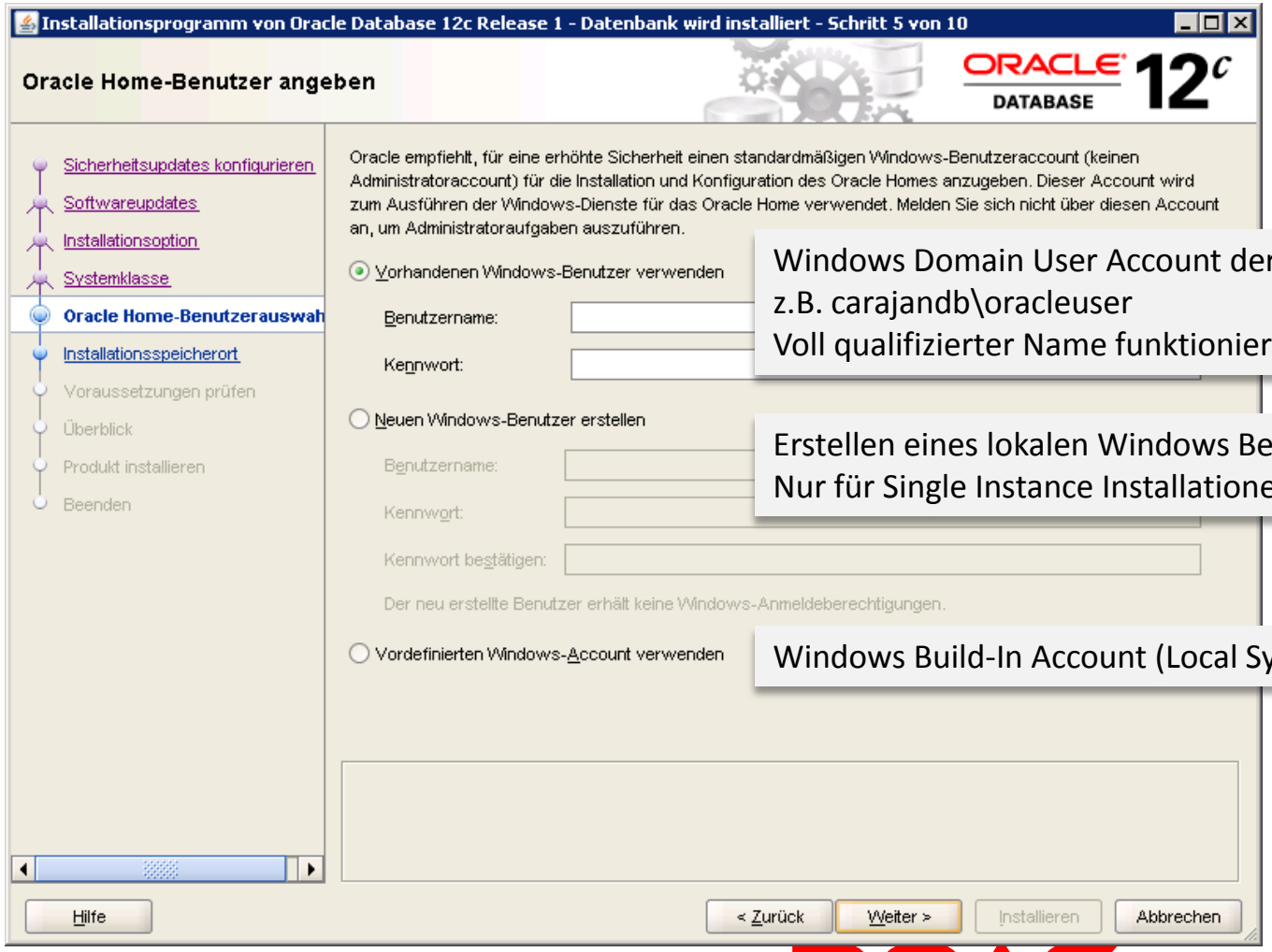
Home User Account … Was ist das?

- Der Oracle Home User ist ein sehr niedrig priorisierter Account, der KEINE Administrationsrechte hat.
- Er dient zum verwalten (starten und stoppen) von Services
- Er ist der Verwalter der Datenbank (nicht der Software!)

Wer kann Home User werden?

- **Jeder lokale oder Domainuser ohne Administrator Privilegien**
- **Ein Vordefinierter Windows Account (Build-in Account)**
  - Local System, Local Service, Network Service
  - In diesem Fall laufen alle Dienste mit lokalen Systemprivilegien (LOCAL SYSTEM) wie Oracle 11g
- **Ein MSA Account (Managed Service Account = Verwaltete Dienstkonten)**
  - Mit Windows Server 2008R2 eingeführt
  - Vergleichbar mit Lokalen Built-In Accounts (local service, local system, network service)
  - Kann nur über PowerShell erstellt und verwaltet werden (aber im AD sichtbar)
  - speziell zur Anwendungsbereitstellung auf Domain-Ebene
  - Automatische Kennwort-Verwaltung, Trennung des Dienstes von anderen Diensten auf dem Server

OHU mit Oracle Home verbinden



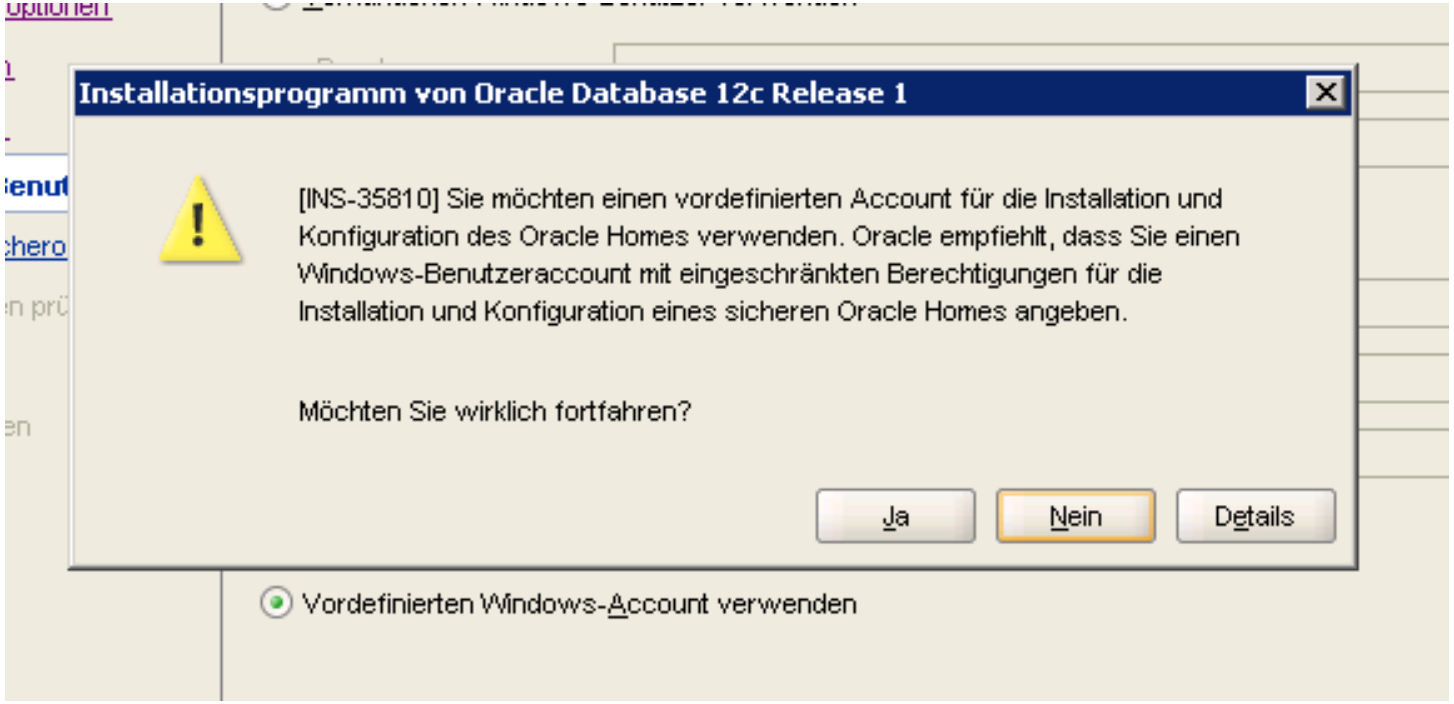
Windows Domain User Account der MSA
z.B. carajandb\oracleuser
Voll qualifizierter Name funktioniert nicht (also nicht carajandb.intra\oracleuser)

Erstellen eines lokalen Windows Benutzers (ohne Login/Anmeldeberechtigung)
Nur für Single Instance Installationen möglich

Windows Build-In Account (Local System, Local Service)

OHU mit Oracle Home verbinden

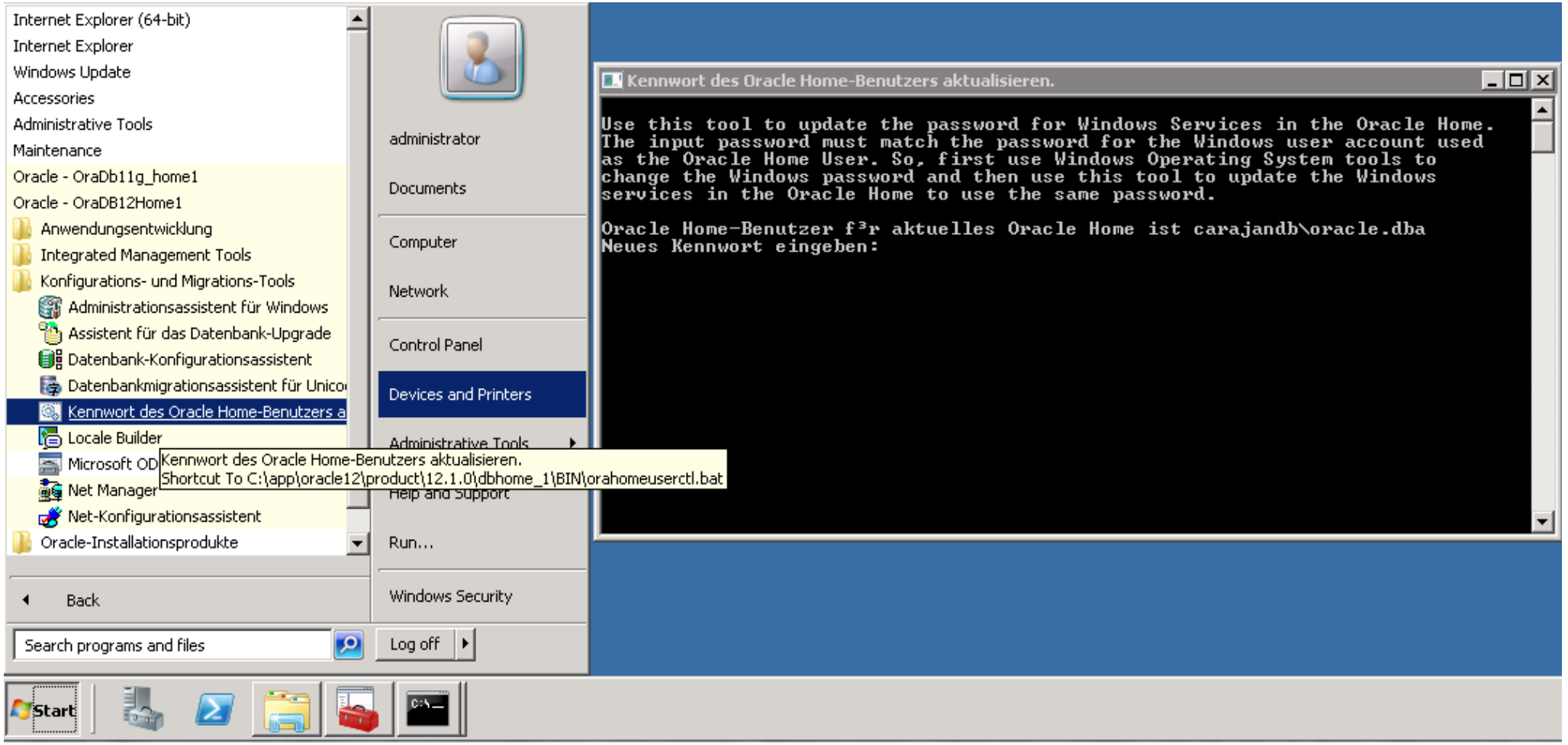
- **Warnung bei Build-In Account**



Aufgaben des OHU

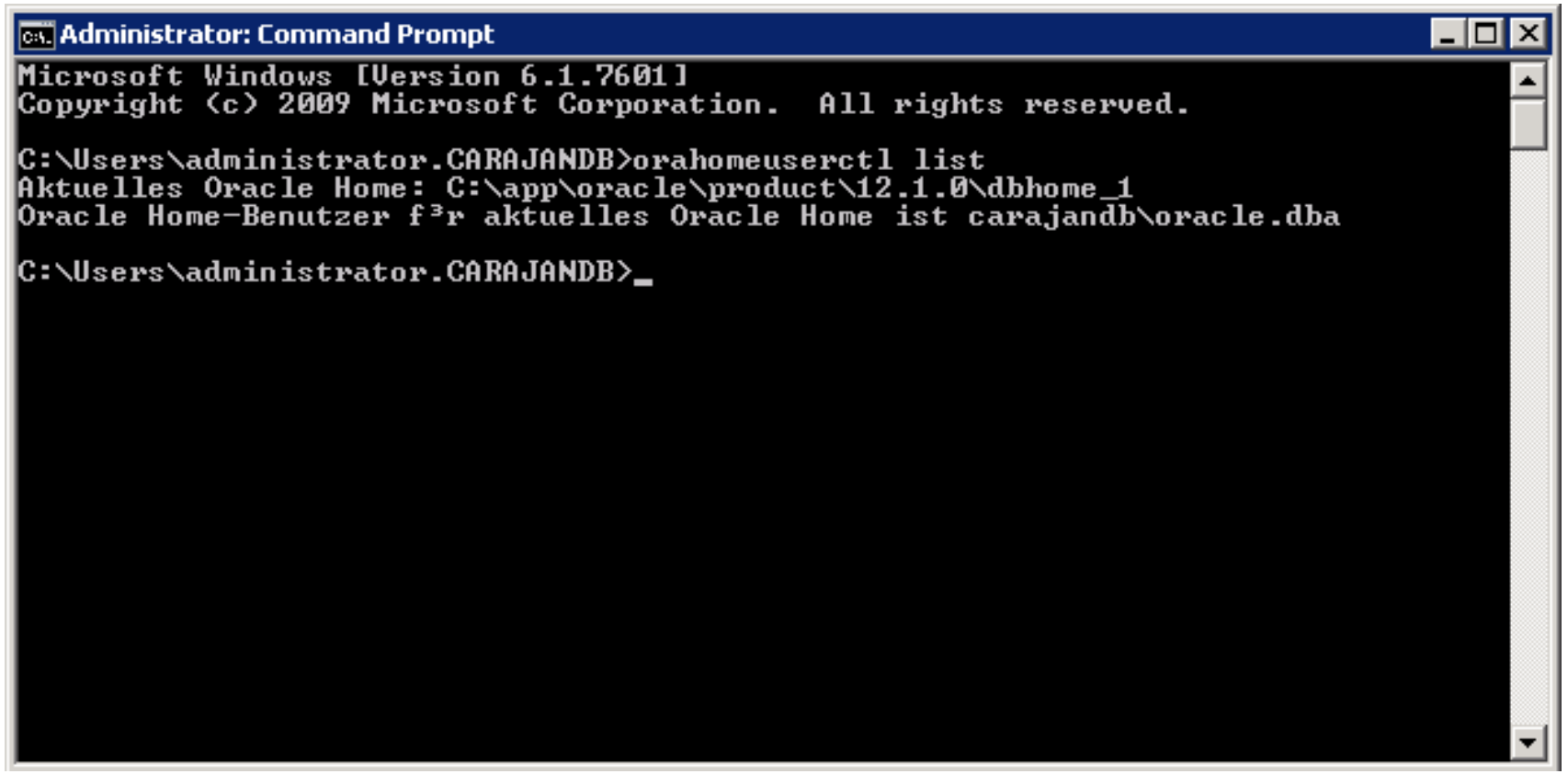
- **Der Oracle Home User hat alle Berechtigungen für das ORACLE_HOME**
  - Er ist NICHT der Besitzer der Software
  - NICHT für Administrative Aufgaben (Installation, Patch, Upgrade) geeignet
  - Er ist der Besitzer der Datenbank
- **Kann Oracle Dienste (Services) starten und stoppen**
- **Kann die Datenbank verwalten**
- **Administrator weiterhin für Oracle Home Administration**

Kennwortänderung des Home User



Oracle Home User Control Tool

- orahomeuserctl



Unified Auditing

- **Standard Auditing und Fine Grained Auditing zusammengefasst**
- **Audit records zunächst nur im Hauptspeicher**
  - Default 1MB SGA
  - Kann bei Bedarf angepasst werden
- **Audit-Tabelle AUDSYS.CLI_SWP…**
  - Partitioniert
  - Im SYSAUX Tablespace

Übergang Oracle 11g → 12c

- **11g:**
  - Standard Auditing → AUD$ Tabelle
  - Fine Grained Auditing → FGA_LOG$ Tabelle
  - SYS Auding → $ORACLE_BASE/admin/<SID>/adump
- **12c:**
  - Standard Auditing → AUD$ Tabelle
  - Fine Grained Auditing → FGA_LOG$ Tabelle
  - Unified Auditing → AUDSYS.CLI_SWP…
  - SYS Auding → $ORACLE_BASE/admin/<SID>/adump
  - PMON Auditing → $ORACLE_BASE/audit/<SID>/*.bin

Generelle Vorgehensweise (auch Oracle 11g)

- **Kein „direktes“ Verschieben von Objekten**

```
SQL> ALTER TABLE SYS.AUX MOVE TABLESPACE AUDMGM;
```

- **Statt dessen:**
  - Package dbms_audit_mgmt

```
BEGIN
   dbms_audit_mgmt.set_audit_trail_location (
      audit_trail_type => dbms_audit_mgmt.audit_trail_aud_std,
      dbms_trail_location_value => 'AUDMGM');

END;
```

Löschen von Einträgen

1. Markieren der Einträge

```
BEGIN
   dbms_audit_mgmt.set_last_archive_timestamp (
      audit_trail_type => dbms_audit_mgmt.audit_trail_unified,
      last_archive_timestamp => sysdate - 31);

END;
```

2. Löschen der Einträge

```
BEGIN
   dbms_audit_mgmt.clean_audit_trail (
      audit_trail_type => dbms_audit_mgmt.audit_trail_unified,
      use_last_arch_timestamp => TRUE);

END;
```

- FALSE löscht alle Einträge

Was wird protokolliert

- **Benutzung von Befehlen (ACTION)**
  - CREATE SESSION
  - CREATE TABLE
  - …
- **Benutzung von Privilegien (PRIVILEGE)**
  - CREATE ANY TABLE
  - ALTER SYSTEM
- **Wann?**
  - Erfolgreich (SUCCESSFUL)
  - Nicht Erfolgreich (NOT SUCCESSFUL)
- **Wer?**
  - Schema
  - Environment (USERENV)

Wie wird protokolliert?

- **CREATE POLICY**

```
CREATE AUDIT POLICY bvtest
   PRIVILEGES
      ALTER SESSION,
      ALTER SYSTEM

   ACTIONS
      LOGON,
      CREATE TABLE,
      EXECUTE ON sys.dbms_audit_mgmt

   ROLES
      DBA;
```

- **Enable Policy**

```
AUDIT POLICY bvtest
   BY <schema>
   WHENEVER NOT SUCCESSFUL;
```

Spezialfall – DML Auditing

- **DML Auditing für bestimmte User:**

```
CREATE AUDIT POLICY BV#DMLAUDIT
  ACTIONS
    INSERT, UPDATE, DELETE
    WHEN '
      SYS_CONTEXT(''USERENV'',''SESSION_USER'') = ''BVEUSGBL''
    OR
      INSTR(SYS_CONTEXT(''USERENV'',''CLIENT_INFO''),''isLocalPersUser'')=1'
  EVALUATE PER STATEMENT;
```

Unterschied ACTION - PRIVILEGE

- **ACTIONS → Ausführen des Kommandos**
- **PRIVILEGES → Nutzung des Privileges**
- **Beispiel:**
  - ACTIONS LOGON
    - Jedes Logon
  - PRIVILEGES CREATE SESSION
    - Nutzung des Privileges „CREATE SESSION“, d.h. SYS nicht!
  - ACTIONS CREATE TABLE
    - Jedes „CREATE TABLE“
  - PRIVILEGES CREATE ANY TABLE
    - Nutzung des Privileges „CREATE ANY TABLE“, d.h. Erstellen der Tabelle im eigenen Schema nicht!

Abfrage

- **UNIFIED_AUDIT_TRAIL View**

```
SELECT os_username,
       userhost,
       dbusername,
       client_program_name,
       external_userid,
       action_name,
       return_code,
       object_schema,
       object_name,
       system_privilege_used,
       unified_audit_policies,
       authentication_type
  FROM unified_audit_trail
 ORDER BY event_timestamp desc;
```

Data Redaction

- **Teil von Oracle Advanced Security**
- **Macht Daten „unsichtbar“**
  - Kein Wert bzw. „0“
  - Alternativer Wert (Random)
  - Funktionen (z.B: xxx)
- **Funktionen:**
  - Full Data Redaction → Komplette Spalte ändern
  - Partial Data Redaction → Teil der Spalte ändern

Data Redaction

- Verwaltung über DBMS_REDACT Package
- Erstellen einer Policy mit „Full Redaction“

```
BEGIN
   dbms_redact.add_policy(
        object_schema => 'demo',
        object_name   => 'kreditkarten',
        column_name   => 'sicherheitscode',
        policy_name   => 'Verstecke Kreditkarteninfos',
        expression    => 'SYS_CONTEXT(''USERENV'',''MODULE'') not like ''TOAD%''',
        function_type => dbms_redact.full);
END;
/
```

Data Redaction

- **Verwendung einer Maskierungsfunktion**

```
BEGIN
    dbms_redact.alter_policy(
        object_schema => 'demo',
        object_name   => 'kreditkarten',
        column_name   => 'kreditkartenid',
        policy_name   => 'Verstecke Kreditkarteninfos',
        function_type => dbms_redact.partial,
        expression    => 'SYS_CONTEXT(''USERENV'',''MODULE'') not like ''TOAD%''',
        function_parameters => 'VVVVFVVVVFVVVVFVVVV,VVVV-VVVV-VVVV-VVVV,X,1,12');
END;
/
```

Data Redaction

- **Nicht autorisierter Zugriff:**

```
SELECT * FROM kreditkarten
 WHERE  persid=100921

KREDITKARTENID          PERSID KARTENANBIETER GUELTIG_V GUELTIG_B SICHERHEITSCODE KREDITWUERDIG
------------------- ---------- -------------- --------- --------- --------------- -------------
XXXX-XXXX-XXXX-8471     100921              2 28-MAY-11 27-MAY-15               0             0
```

Data Redaction

- **Löschen einer Policy**

```
BEGIN
   dbms_redact.drop_policy(
        object_schema => 'DEMO',
        object_name   => 'KREDITKARTEN',
        policy_name   => 'Verstecke Kreditkarteninfos');
END;
/
```

Multitenant Database

DOAG
Deutsche ORACLE-Anwendergruppe e.V.

Common Role

- Rolle wird in der CDB definiert und ist automatisch in allen oder definierten PDBs verfügbar
- Privilegien in den PDBs können unterschiedlich sein
- Voraussetzung: Role beginnt mit “C##“
- Common Privilegien durch Angabe von „CONTAINER=ALL“

```
SQL> CREATE ROLE c##dba CONTAINER=ALL;

Role created.

SQL> GRANT CREATE SESSION TO c##dba;

Grant succeeded.
```

Common User

- **Voraussetzung wie bei der Role**

```
SQL> CREATE USER c##dbauser IDENTIFIED BY "oracle" CONTAINER=ALL;
User created.

SQL> GRANT c##dba TO c##dbauser;
Grant succeeded.

SQL> CONNECT c##dbauser/oracle
Connected.
SQL> ALTER SESSION SET CONTAINER=nachtmusik;
ERROR:
ORA-01031: insufficient privileges
```

- **Warum?**
  - Weil das Privileg „CREATE SESSION“ nur für die CDB vergeben worden ist!

Benutzer in PDB

- Schemaverwaltung in der PDB genau wie bei NON-CDB Datenbank
- Globalen Usern können in der PDB weitere Rechte gegeben werden

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL> ALTER SESSION SET CONTAINER=nachtmusik;
Session altered.
SQL> GRANT CREATE SESSION TO c##dbauser; -- User
Grant succeeded.
Oder:
SQL> grant create session to c##dba;     -- Role
Grant succeeded.
Aber nicht:
SQL> GRANT CREATE SESSION TO c##dba CONTAINER=ALL;
GRANT CREATE SESSION TO c##dba CONTAINER=ALL
*
ERROR at line 1:
ORA-65050: Common DDLs only allowed in CDB$ROOT
```

ALTER SESSION SET CONTAINER

- **Was ist mit dem Grant dieses Privilegs?**

```
SQL> CONNECT c##dbauser/oracle
SQL> ALTER SESSION SET CONTAINER=nachtmusik;
ERROR:
ORA-01031: insufficient privileges
```

- **Privileg SET CONTAINER**

```
SQL> GRANT SET CONTAINER TO c##dbauser CONTAINER=ALL;
SQL> ALTER SESSION SET CONTAINER=nachtmusik;
Session altered.
```

  - ALTER SESSION Privileg reicht nicht aus
  - Beide Container (CDB$ROOT, PDB) müssen angegeben werden, daher besser CONTAINER=ALL
  - Privileg kann an Role vergeben werden, wirkt dort aber nicht (warum?)

Objektprivilegien

- **Vergabe von Objektprivilegen an Common User ist nicht möglich**

```
SQL> GRANT SELECT ON demo.personen TO c##dbauser CONTAINER=ALL;
ERROR at line 1:
ORA-00942: table or view does not exist
```

- **Gründe:**
  - Objekt müsste in ALLEN PDBs und in der CDB existieren
  - In der CDB kann aber ein User nicht ohne Prefix „c##“ angelegt werden
  - Selbst wenn das Objekt überall existieren würden (z.B. SYNONYM) ➔ ORA-65033 (a common privilege may not be granted or revoked on a local object)

Interne Benutzer

- **SYS**
  - Ist in allen Pluggable Databases vorhanden und darf alles
- **SYSTEM**
  - Ebenfalls in allen Pluggable Databases vorhanden
    - Darf Pluggable Databases erstellen
    - Kann keine Pluggable Database öffnen oder schließen

PDB-Übergreifende Abfragen

- Seit Version 12.1.0.2
- Syntax

```
SELECT …
  FROM CONTAINERS (<tables_name>)
 WHERE CON_ID = …
   AND …
```

Voraussetzung

- **Objekt muss für Common User existieren**
  - Ev. als View (schemaname.tablename)
- **Objekt muss auch in der CDB existieren**

Beispiel 1 (1)

- **CDB:**

```
SQL> CREATE user c##demo IDENTIFIED BY demo CONTAINER=ALL;

SQL> GRANT CREATE SESSION, CREATE TABLE, CREATE VIEW, CREATE SYNONYM
     TO c##demo CONTAINER=ALL;

SQL> CREATE TABLE C##DEMO.PERSONEN(
         PERSID      NUMBER(10)                          NOT NULL,
         ANREDE      VARCHAR2(5 CHAR),
         VORNAME     VARCHAR2(50 CHAR),
         NACHNAME    VARCHAR2(50 CHAR),
         GEBURTSTAG  DATE,
         BILD        BLOB
)
LOB (BILD) STORE AS SECUREFILE (TABLESPACE  USERS)
TABLESPACE USERS
RESULT_CACHE (MODE DEFAULT);
```

Beispiel 1 (2)

- **Jede PDB:**

```
SQL> CONNECT demo/demo
SQL> GRANT SELECT ON personen TO c##demo;

SQL> CONNECT c##demo@PDB

SQL> CREATE synonym personen ON demo.personen;
```

- **Abfrage:**

```
SQL> SELECT * FROM CONTAINERS(personen);

→ ERROR at line 1:
ORA-00600: internal error code, arguments: [kkdoilsn:tabDelSeg], [], [], [],
[], [], [], [], [], [], [],
```

Wofür braucht man das?

- **Nicht für das genannte Beispiel!**
- **Auswertung / Vergleich von Performancedaten**
- **z.B. Statspack installiert in jeder PDB**
- **Neuer User c##perfstat mit Zugriff auf alle Statspack Schemata**
- **Offene Punkte:**
  - Wie kann man es vermeiden, dass die Tabellen auch in der CDB existieren?

PL/SQL Inherit Rights

- Standardmäßig Übernahme von Rechten auch über PL/SQL
- Neue Klausel „INHERIT“

```
SQL> REVOKE INHERIT PRIVILEGES on USER dbaler FROM PUBLIC;
```

Fragen?

Johannes Ahrends

www.carajandb.com

Johannes.ahrends@carajandb.com

DOAG
Deutsche ORACLE-Anwendergruppe e.V.