DOAG
Deutsche ORACLE-Anwendergruppe e.V.

CARAJAN DB

Oracle Benutzerverwaltung

Johannes Ahrends

… 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
- DOAG Themenverantwortlicher Datenbankadministration, Standard Edition
- Hobbies:
  - Drachen steigen lassen (Kiting) draußen wie drinnen (Indoorkiting)
  - Motorradfahren (nur draußen)
  - Bier brauen
  - Singen (überall)

Komponenten

- Benutzer
- Rollen
- Profile
- Auditing

Rollen

Rollen

- Beinhalten System- und Objekt-Privilegien
- Sind keinem Benutzer oder Schema zugeordnet
- Können an andere Benutzer oder Rollen vergeben werden (nicht an sich selbst)
- Können für einen Benutzer ein- oder ausgeschaltet werden

Vordefinierte Rollen

```
SQL> SELECT role FROM dba_Roles WHERE oracle_maintained='Y';

ROLE
------------------------------
CONNECT
RESOURCE
DBA
PDB_DBA
AUDIT_ADMIN
AUDIT_VIEWER
SELECT_CATALOG_ROLE
EXECUTE_CATALOG_ROLE
CAPTURE_ADMIN
EXP_FULL_DATABASE
…

89 rows selected.
```

Blog von Markus Flechtner zu ORACLE_MAINTAINED:
https://www.markusdba.de/2016/06/23/eine-hilfreiche-spalte-in-oracle-12c-oracle_maintained/

Resource Rolle

- Provides the following system privileges: CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER, CREATE TYPE.
- Be aware that RESOURCE no longer provides the UNLIMITED TABLESPACE system privilege.
- This role is provided for compatibility with previous releases of Oracle Database. You can determine the privileges encompassed by this role by querying the DBA_SYS_PRIVS data dictionary view.
- Note: Oracle recommends that you design your own roles for database security rather than relying on this role. This role may not be created automatically by future releases of Oracle Database.

Oracle Database 19c Security Guide

Vorschlag

- Eigene Rollen anlegen
- Beispiel:
  - `<FIRMENKÜRZEL>_DBA`
  - `<FIRMENKÜRZEL>_SCHEMA / <FIRMENKÜRZEL>_RESOURCE`
  - `<FIRMENKÜRZEL>_CONNECT`
- Bei Multitenant (Common Roles)
  - `C##<FIRMENKÜRZEL>_DBA`
  - `C##<FIRMENKÜRZEL>_SCHEMA / <FIRMENKÜRZEL>_RESOURCE`
  - `C##<FIRMENKÜRZEL>_CONNECT`
- Bei NON-CDB kann man keine Rollen anlegen, die mit C## beginnen

Common Role

- Rolle wird in der CDB definiert und ist automatisch in allen PDBs verfügbar
- Privilegien in den PDBs können unterschiedlich sein
- Voraussetzung: Role beginnt mit “C##“ (Serverparameter)

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

Role created.

SQL> GRANT CREATE SESSION, ALTER SESSION TO c##dba CONTAINER=ALL;

Grant succeeded.
```

Wer ist eigentlich „SYSTEM“

Wer ist „SYSTEM“?

- “When you create an Oracle database, the user SYSTEM is also automatically created and granted the DBA role”
- Ein Beispiel für einen DBA Benutzer
- … und die DBA Rolle?
- „A predefined DBA role is automatically created with every Oracle Database installation. This role contains most database system privileges. Therefore, the DBA role should be granted only to actual database administrators

Datenbankbenutzer

```
CREATE | ALTER USER <username>
  IDENTIFIED [ BY tiger |  EXTERNALLY | GLOBALLY AS '...' ]
  DEFAULT TABLESPACE user_ts
  TEMPORARY TABLESPACE temp_ts
  PROFILE profile
  DEFAULT ROLE [ role | ALL | ALL EXCEPT role | NONE ]
  PASSWORD EXPIRE
  ACCOUNT [ LOCK | UNLOCK ]
  …;

DROP USER <username> [CASCADE];
```

Spezialfall (Originalpasswort eines Benutzers wiederherstellen):

```
CREATE | ALTER USER <username>
  IDENTIFIED BY VALUES '...'
```

… IDENTIFIED BY VALUES

- Anlegen eines Benutzers, ohne das Passwort zu kennen
  - Z.B. bei Data Pump Export / Import
  - Ungültiges Passwort (`IDENTIFIED BY VALUES 'GEHEIM'`)
- Spalte `SPARE4` in der Tabelle `USER$`

```
SELECT name, spare4 FROM user$ WHERE name = 'SYSTEM'

NAME       SPARE4
---------- ----------------------------------------------------------------
SYSTEM     S:029D81DEA529106A767BDF686778897B1185B2DA5CD0D1C285D5610076DE;T:67168
           12B4964575A9F6E83F19E5A3FC7FBDB6FEBC7A0D0D574B64386359FFE6ED39C4F2E7BA
           140D07BF5B6EF1CE40593986122EE2AED5E7EA090EC60939C0A98701CF21D69DE5C3FA
           2B2E4D1DFD865B6
```

- S: Oracle 11g SHA-1 Verschlüsselung
- T: Oracle 12c SHA-512 Verschlüsselung

Vorschlag

- Nur Persönliche User als DBAs definieren
  - Mit der selbst definierten Rollen
- Technischer User für Anwendungen
  - ggf. mit Wallet Passwort
- Trennung von User und Schema
  - Dem Schema gehören die Objekte (nur DDL)
    - Schema gesperrt
  - Der User hat Zugriff auf die Objekte (nur DML)
- Common User immer mit C## Prefix

Common User

- Voraussetzung wie bei der Role

```
SQL> CREATE USER c##dbauser IDENTIFIED BY "oracle";

SQL> GRANT c##dba TO c##dbauser;

SQL> CONNECT c##dbauser/oracle

SQL> ALTER SESSION SET CONTAINER=doria;
ERROR:
ORA-01031: insufficient privileges
```

ALTER SESSION SET CONTAINER

- Privileg SET CONTAINER

```
SQL> GRANT SET CONTAINER TO c##dbauser CONTAINER=ALL;

SQL> ALTER SESSION SET CONTAINER=cello;
Session altered.
```

- ALTER SESSION Privileg reicht nicht aus
- Privileg kann nur an User vergeben werden!
- Hier muss die Option „CONTAINER=…“ angegeben werden!

Default Tablespace

- Jeder User hat
  - einen Default Tablespace
  - einen Default Temporary Tablespace
- Common User:
  - Die Tablespaces müssen in allen (!) PDBs existieren

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)

Common User 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=cello;
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
```

Default Tablespace

- Jeder User hat
  - einen Default Tablespace
  - einen Default Temporary Tablespace
- Common User:
  - Die Tablespaces müssen in allen (!) PDBs existieren

CONTAINER_DATA

- Tabellen und Views als „CONTAINER_DATA“ gekennzeichnet
- Zugriff nur mit entsprechenden Privilegien

```
SQL> SELECT view_name FROM cdb_views WHERE container_data='Y';

VIEW_NAME
------------------------------
…
DBA_PDBS
…
GV_$PDBS
…
```

- Gilt auch für Tabellen (cdb_tables) dort gibt es aber standardmäßig keine

```
SQL> SELECT table_name FROM cdb_tables WHERR container_data='YES';

no rows selected
```

Beispiel CONTAINER_DATA

```
SQL(SYS)> CREATE USER c##developer IDENTIFIED BY manager;
SQL> GRANT create Session TO c##developer;
SQL> connect c##developer/manager
SQL> SELECT * from v$pdbs;
SELECT * from v$pdbs
              *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL(SYS)> GRANT SELECT ON v_$pdbs TO c##developer;

SQL> SELECT name from v$pdbs;
no rows selected

SQL (SYS)> ALTER USER c##developer SET CONTAINER_DATA=ALL CONTAINER=CURRENT;

SQL> SELECT name from v$pdbs;

NAME
----------
PDB$SEED
CELLO
DORIA
ANDREA
```

Schema Only Account

- Seit Version 18

1. Anlegen eines „normalen“ Users

```
GRANT create session TO johannes IDENTIFIED BY manager;
```

2. Anlegen eines Schema Only Accounts ohne Passwort

```
CREATE USER besitzer NO AUTHENTICATION QUOTA UNLIMITED ON users;
GRANT create session, create table, create sequence TO besitzer;
```

3. Zugriff auf Objekte für den normalen User zulassen:

```
ALTER USER besitzer GRANT CONNECT THROUGH johannes;
```

4. Anmeldung an den User mit zugehörigem Schema

```
sqlplus johannes[besitzer]/<passwort>@<TNS-Alias>
```

Privilege Capture

- Seit 12c
- Least Privilege Analyse
- Datenbank Package mit folgenden Prozeduren:
  - `CREATE_CAPTURE` → Erstellt eine Analysepolicy
    - Unterschiedliche Granularitäten (z.B. Datenbank, Rolle, Context)
  - `ENABLE_CAPTURE` → Aktivitiert die Policy und damit die Messung
  - `DISABLE_CAPTURE` → Deaktiviert die Policy
  - `GENERATE_RESULTS` → Erstellt die Datenbank Views
    - Z.B. `dba_used_sysprivs, dba_unused_sysprivs`
  - `DROP_CAPTURE` → Löscht die Analysepolicy

Profiles

Profile

- Limitierung von Ressourcen
  - CPU
  - SESSION
  - READ‘s
  - IDLE TIME
  - CONNECT TIME
- Password Management
  - Account Locking
  - Password Aging / Expiration
  - Password History
  - Password Complexity Verification
  - Erweiterungen über PL/SQL Prozeduren

Profile

- Jedem Benutzer kann ein Profil oder das Default Profil zugeordnet werden
- Zuweisung über `CREATE/ALTER USER`
- Profil DEFAULT, für alle Benutzer ohne explizites Profile oder für nicht gesetzte Werte in anderen Profilen
- Profil DEFAULT Passwort läuft nach 180 Tagen ab
  - Vor 11g unlimited
- Wird angezogen beim Starten einer Session

Profile

- Ressourcenlimitierung

```
CREATE | ALTER ] PROFILE <profile> LIMIT
SESSIONS_PER_USER <value>
CPU_PER_SESSION <value>
CPU_PER_CALL <value>
CONNECT_TIME <value>
IDLE_TIME <value>
LOGICAL_READS_PER_SESSION <value>
LOGICAL_READS_PER_CALL <value>
COMPOSITE_LIMIT <value>
PRIVATE_SGA <value>
```

```
<value> := [ integer | UNLIMITED | DEFAULT ]
```

Profile

- Kennwort Management

```
CREATE | ALTER ] PROFILE <profile> LIMIT
FAILED_LOGIN_ATTEMPTS <value>
PASSWORD_LIFE_TIME <value>
PASSWORD_REUSE_TIME <value>
PASSWORD_REUSE_MAX <value>
PASSWORD_LOCK_TIME <value>
PASSWORD_GRACE_TIME <value>
PASSWORD_VERIFY_FUNCTION [function|NULL|DEFAULT]

<value> := [ integer | UNLIMITED | DEFAULT ]
```

Profile Resourcemanagement

- Aktivieren über Server Parameter
  `RESOURCE_LIMIT=TRUE`
  oder
  `ALTER SYSTEM SET RESOURCE_LIMIT=true;`
- Views
  `USER_RESOURCE_LIMITS`
  `USER_PASSWORD_LIMITS`
  `DBA_PROFILES`
  `RESOURCE_COST`

Default Profile

- Für alle User gesetzt, wenn kein anderes explizit angegeben wurde

> **DEFAULT**
>
> Specify `DEFAULT` if you want to omit a limit for this resource in this profile. A user assigned this profile is subject to the limit for this resource specified in the `DEFAULT` profile. The `DEFAULT` profile initially defines unlimited resources. You can change those limits with the `ALTER PROFILE` statement.
>
> Any user who is not explicitly assigned a profile is subject to the limits defined in the `DEFAULT` profile. Also, if the profile that is explicitly assigned to a user omits limits for some resources or specifies `DEFAULT` for some limits, then the user is subject to the limits on those resources defined by the `DEFAULT` profile.

- Hier ist von zwei unterschiedlichen „DEFAULTS“ die Rede!
- Keine genaue Definition in der Dokumentation auffindbar

Oracle Database Release 19 SQL Language Reference

Vorschlag

- Eigene Profiles verwenden
  - Wesentlich für die Passwortkomplexität / Alterung
- Beispiel:
  - `C##_<FIRMENKÜRZEL>_TECH_PROFILE`
  - `C##_<FIRMENKÜRZEL>_PERS_PROFILE`
  - `C##_<FIRMENKÜRZEL>_TEMP_PROFILE`
- Es gibt wenig Gründe, warum man bei der Multitenant Datenbank lokale Profiles verwenden sollte.
  - Kann allerdings bei Migrationen vorkommen

Passwort Komplexität

- Oracle Skript `$ORACLE_HOME/rdbms/admin/catpvf.sql`
- Erstellt 7 Verifizierungsfunktionen und ein zusätzliches Profile
  - `ora_complexity_check`
  - `ora_string_distance`
  - `ora12c_verify_function`
  - `verify_function_11G`
  - `verify_function`
  - `ora12c_strong_verify_function`
  - `ora12c_stig_verify_function + ora_stig_profile`
- Tipp: nicht direkt verwenden, sondern eigene auf dieser Basis erstellen

Common Profile

- Profiles werden auf CDB-Ebene gesetzt
- C## als Prefix

```
SQL> CREATE PROFILE c##techuser LIMIT
        FAILED_LOGIN_ATTEMPTS 5
        PASSWORD_LIFE_TIME 60
        PASSWORD_REUSE_TIME 60
        PASSWORD_REUSE_MAX 5
        PASSWORD_VERIFY_FUNCTION meine_verify_function
        PASSWORD_LOCK_TIME 1/24
        PASSWORD_GRACE_TIME 10
        INACTIVE_ACCOUNT_TIME 30;
```

Passwortdatei

DBA Login

- Initialisierungsparameter
  - `REMOTE_LOGIN_PASSWORDFILE= [NONE|EXCLUSIVE|SHARED]`
- Password Datei ist DB-extern
  (Variable `ORA_PWFILE`, `ORA_<SID>_PWFILE`)
- ORAPWD Utility
  - Unter Windows auch Instance Manager
- enthält Passwörter der Benutzer mit `SYSOPER`,`SYSDBA`, `SYSDG`, `SYSBACKUP`, `SYSKM`, `SYSRAC` Privileg
- wird automatisch synchronisiert

DBA Login

- `SYSOPER`: Startup, Shutdown, vollständiges Recovery
- `SYSDBA`: zusätzlich alle anderen DBA-Privilegien
- `SYSDG`: Ausschließlich die Verwendung von Data Guard Broker Befehlen (kein Login in die Datenbank möglich)
- `SYSBACKUP`: Ausschließlich die Verwendung des Oracle Recovery Managers (Kein Login in die Datenbank möglich)
- `SYSKM`: Ausschließlich die Verwaltung von Wallets (Kein Login in die Datenbank möglich)
- `SYSRAC`: Verwaltung von RAC (ab Oracle 12.2)
- `SYSASM` Verwaltung von ASM Instanzen
- Login mit

```
CONNECT SYS/<pwd>@db AS SYSDBA
DGMGRL DEMO/<pwd>@DB AS SYSDG
```

Auditing

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

Unified Auditing

- Eigenes Schema AUDSYS
- Neue Rollen AUDIT_ADMIN und AUDIT_VIEWER
- AUD$ und FGA_LOG$ zu einem Auditing zusammengefasst
- Zugriff auf OS-Audit-Informationen
- Partitionierte Tabelle (auch bei Standard Edition) `CLI_SWP...`
  - Maintenance über Drop / Create Partition
  - Standardmäßig im Tablespace SYSAUX
- Zwischenspeicherung in der SGA
  - `unified_audit_sga_queue_size = 1MB`
  - Gefahr von Verlust von Audit-Informationen
    - Optional: sofortiges Speichern in der Tabelle

Unified Auditing

- Initialisierung nicht mehr notwendig / möglich
- Auditing von Aktionen
  - Recovery Manager
  - DataPump
  - SQL*Loader

Kernel Relink

```
SELECT parameter, value from v$option
 WHERE parameter = 'Unified Auditing';

PARAMETER                      VALUE
------------------------------ ----------

Unified Auditing               FALSE
```

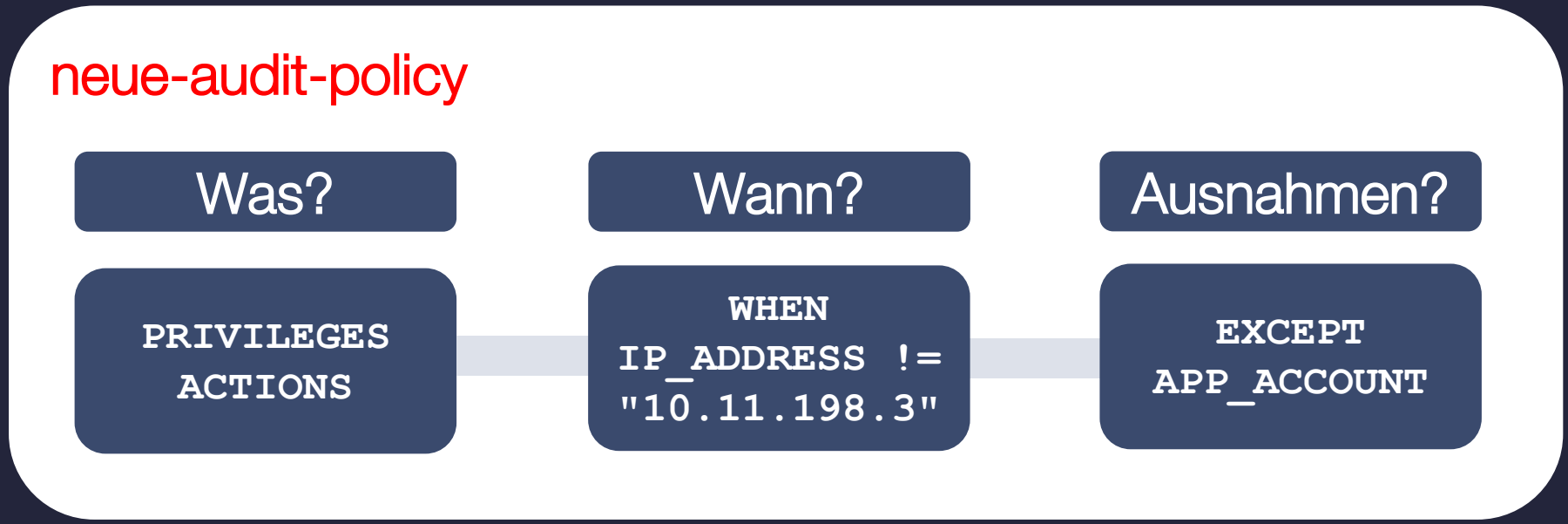
- Besagt nur, dass Unified Auditing nicht exklusiv benutzt wird, sondern das „alte“ Auditing ebenfalls genutzt werden kann (Mixed Mode)!
- Durch Kernel Relink kann das „alte“ Auditing ausgeschaltet werden

Unterschied 12.1 zu >=12.2

- Standard: Gleiche Überwachung wie in 12.1
- Neue Tabelle im Schema AUDSYS
  - AUD$UNIFIED (vorher CLI_$....)
  - Partitioniert nach Monaten
  - Mehr einzelne Felder
    - 12.1 13 Spalten mit einem großem BLOB (LOG_PIECE)
    - 12.2 101 (!) Spalten inklusive 3 CLOBs)
  - Nachteil:
    - Daten direkt lesbar
  - Vorteil:
    - Performanter

12c Auditing – Audit Policy

- Grundaufbau 12c Audit Policy



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 cartest
   PRIVILEGES
      ALTER SESSION,
      ALTER SYSTEM

   ACTIONS
      LOGON,
      CREATE TABLE,
      EXECUTE ON sys.dbms_audit_mgmt

   ROLES
      DBA;
```

Wer wird protokolliert

- Users in Bezug auf ihre Rollen

```
SQL> AUDIT POLICY apptest
     BY USER
     WITH GRANTED ROLE app_rolle;
```

Spezialfall – DML Auditing

- DML Auditing für bestimmte User:

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

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;
```

12c Auditing – Standard Policies

- Mitgelieferte Standard Policies
  - Account Management
    - Create User
    - Grant, Revoke
    - Create Role
    - Drop Role
  - Database Parameters
    - Create SPFILE
    - Alter System
    - Alter Database
  - Secure Configuration
    - Alter Any
    - Create Any
    - Alter User
    - Alter Profile

```
SQL> SELECT distinct policy_name
       FROM audit_unified_policies;

POLICY_NAME
------------------------------
AUDIT_ALL_DDL
ORA_DV_AUDPOL2
ORA_CIS_RECOMMENDATIONS
ORA_ACCOUNT_MGMT
ORA_DATABASE_PARAMETER
ORA_LOGON_FAILURES
ORA_DV_AUDPOL
ORA_SECURECONFIG
ORA_RAS_SESSION_MGMT
ORA_RAS_POLICY_MGMT
```

CIS = Center for Internet Security
RAS = Real Application Security
DV = Database Vault

Aktive Policies

- Zweo Unified Audit Policies bereits aktiv (außer in der CDB$ROOT)

```
SQL> SELECT * FROM audit_unified_enabled_policies;

POLICY_NAME          ENABLED_OPTION  ENTITY_NAME          ENTITY_TYP SUCCESS    FAILURE
-------------------- --------------- -------------------- ---------- ---------- --------
ORA_SECURECONFIG     BY USER         ALL USERS            USER       YES        YES
ORA_LOGON_FAILURES   BY USER         ALL USERS            USER       NO         YES
```

  - ORA_SECURECONFIG
    - Überwachung Audit-Konfiguration, Audit Trail und kritische Systemprivilegien
  - ORA_LOGON_FAILURES
    - Überwacht keine erfolgreichen Logins !!!
- Deaktivieren:

```
SQL> NOAUDIT ORA_SECURECONFIG;
SQL> NOAUDIT ORA_LOGON_FAILURES;
```

CarajanDB

- Experten mit über 30 Jahren Datenbank Erfahrung
- Spezialisten für
  - Datenbank Administration (Oracle und PostgreSQL)
  - Hochverfügbarkeit (RAC, Data Guard, Replication, etc.)
  - Migrationen (Unicode, PostgreSQL)
  - Performance Optimierung
  - Monitoring (OEM, Foglight, CheckMK, PEM)
- Fernwartung
- Schulung und Workshops
  - PostgreSQL
  - Oracle Multitenant
  - Toad

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: carajandb
- Facebook: johannes.ahrends
- Blogs:
  - blog.carajandb.com
  - www.toadworld.com