

DB Verwalten

&

Admin Book

Uwe Schimanski

Seab@er Software

Goch, den 18. Juli 2019

Copyright © 2018 Uwe Schimanski

PUBLISHED BY PUBLISHER

SEABAER-AG.DE

Licensed under the Creative Commons Attribution-NonCommercial 3.0 Unported License (the “License”). You may not use this file except in compliance with the License. You may obtain a copy of the License at <http://creativecommons.org/licenses/by-nc/3.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

First printing, January 2019

Inhaltsverzeichnis

Vorwort	v
1 Verwaltung	1
1.1 SQLPLUS	1
1.1.1 Konfiguration	1
1.1.2 SqlPlus Einstellungen sichern	3
1.2 Connect	4
1.3 Datenbank starten manuell	5
1.4 Datenbank von Hand herunterfahren	6
1.5 Automatisches Shutdown und Startup	6
1.6 Instanz	8
1.6.1 Überprüfen	8
1.6.2 Löschen einer Instanz (Windows)	8
1.7 SGA & PGA	9
1.7.1 SGA abfragen	9
1.7.2 PGS abfragen	10
1.7.3 PGA Optimale Größe ermitteln	10
1.7.4 SGA / PGA anpassen	10
1.7.5 Dynamic Resize Views	11
1.8 Archivelog Modus	11
1.8.1 Modus Prüfen	11
1.8.2 Archivelog Einschalten	11
1.8.3 Log Switch	12
1.8.4 Archivelog ausschalten	12
1.8.5 Archivelog anzeigen	12
1.8.6 Archivelog Format	12
1.9 Flashback	13
1.9.1 Flashback einschalten	13
1.9.2 Flashback Status abfragen	13
1.9.3 Flashback Nummer abfragen	13
1.9.4 Flashback Size	13
1.9.5 Restore Point abfragen	14
1.9.6 Flashback ausschalten	14
1.9.7 Delete Old Flashback Files	14
1.10 Control Files	15
1.10.1 Verwalten der Control Files	15
1.10.2 Sicherung der Control Files	15
1.10.3 Zusätzliche Control-Dateien	16
1.10.4 Vorhalte Zeit	16
1.10.5 Restore	16
1.11 SPFile wieder herstellen	17
1.12 Redo Logs	18
1.12.1 Informationen abfragen	18

1.12.2	Anlegen weiterer Redo Log Dateien	19
1.12.3	Löschen einer Redo Log Gruppe / Datei	19
1.12.4	Umbenennen oder Pfad ändern	20
1.13	Tablespaces	21
1.13.1	Neuanlegen von Tablespaces	21
1.13.2	Erweitern von Tablespaces	21
1.13.3	Verändern der Default Storage Parameter	21
1.13.4	Lage bzw. Umbenennen	22
1.13.5	Löschen von Tablespace	22
1.13.6	Tablespace offline/online setzen	22
1.13.7	Datafile offline/online setzen	22
1.13.8	Temp Tablespace verkleinern / vergrößern	23
1.13.9	Datafile löschen	24
1.13.10	Autoextend Informationen	24
1.13.11	Tablespace umwandeln	24
1.13.12	Überwachen von Tablespaces	25
1.14	Rollback Segmente	27
1.14.1	Anlegen und Verwalten	27
1.14.2	Löschen	27
1.14.3	Überwachen	28
1.15	Deadlocks	29
1.15.1	Anzeigen und löschen	29
1.16	Extents	31
1.16.1	Freie Extents zusammen fassen	31
1.16.2	Blöcke / Extents reorganisieren	31
1.17	Zeitzone	33
1.18	Netzwerk	34
1.18.1	Tnsnames.ora & Sqlnet.ora	34
1.18.2	Listener Trace	34
1.18.3	Sqlnet Trace	35
1.19	Oracle User	36
1.19.1	Liste aller verbundenen User	36
1.19.2	Verbundene User löschen	36
1.19.3	Listet alle Oracle User aus	36
1.19.4	Angemeldeten User anzeigen	36
1.19.5	Liste aller User Tables	36
1.19.6	Anzeige der SYS Berechtigungen	37
1.20	DB Informationen	38
1.20.1	Bootstrap-Segment	38
1.20.2	Anzeige aller aktuellen Werte	38
1.20.3	Hidden Parameter	38
1.20.4	Einstellungen der Datenbank	39
1.20.5	Datenbank Service Name	39
1.20.6	NLS Parameter	39
1.21	Oracle Audit	40
1.21.1	Audit Status abfragen	40
1.21.2	Audit einschalten	40
1.21.3	Audit Actions abfragen	41
1.21.4	Audit Regel erstellen	41
1.21.5	Audit Regel löschen	45
1.22	Datenbank Check	46
1.22.1	Datenbank Überprüfen	46
1.22.2	Datenbank reparieren	47

1.23	Corrupte Blöcke	48
1.23.1	Anzeigen	48
1.23.2	Repair Tabellen erstellen	48
1.23.3	Tabellen Corruption finden	49
1.23.4	Corruption Blocks beheben	50
1.23.5	Index Einträge zu Corrupt Data Blocks finden	50
1.23.6	Überspringen der Corrupten Blöcke	51
1.23.7	Object Typen	51
1.23.8	Raparatur von Hand	51
1.24	Jobs und Task	53
1.24.1	Jobs auflisten	53
1.24.2	Job erstellen	54
1.24.3	Jobs starten	55
1.24.4	Job modify	55
1.24.5	Jobs disable	55
1.24.6	Jobs enable	56
1.24.7	Job Zeiten ändern	56
1.24.8	Jobs löschen	56
1.24.9	Tasks anzeigen	56
1.25	Spfile oder Pfile	58
1.26	DB Tuning	58
1.26.1	SQL optimieren	58
1.26.2	Parallele Operationen ausführen	59
1.26.3	Buffer Cache messen	59
1.26.4	Weitere Tuning Parameter	59
1.27	Oracle 32/64 Bit	59
1.28	Status DML	60
1.29	Uptime DB	61
1.30	Oracle Verbindungs Informationen	62
1.31	Oracle Version	63
1.32	Patch	64
1.32.1	Status	64
1.32.2	Check Conflict	64
1.32.3	Installation	64
1.32.4	Error Code 73	65
1.32.5	Error Code 74	65
1.32.6	Error Code 104	65
1.33	Database Optionen enable / disable	66
1.34	Feature Report	66
1.35	Ausführungsplan	66
1.36	Papierkorb	67
1.36.1	Anzeigen	67
1.36.2	Leeren	67
1.37	Character Set	67
1.37.1	Ändern	67
1.38	Prozesse	68
1.38.1	Anzeigen	68
1.39	DBA errors	68
1.39.1	Anzeigen	68
1.40	Database verschieben	68
1.41	Open Cursors	69
1.42	External Jobs	70
1.43	Statspack	70

1.43.1	Installation	70
1.43.2	Snapshot erstellen	71
1.43.3	Snapshot Job	71
1.43.4	Report erstellen	72
1.43.5	Snapshot löschen	72
1.43.6	Snapshot anzeigen	73
1.43.7	Statspack Analyse	73
1.43.8	Statsack löschen	73
1.44	AWR	73
1.44.1	Diagnostic / Tuning Pack deaktivieren	74
1.45	Oracle Trace	74
1.45.1	Methode 1	74
1.45.2	Methode 2	75
1.45.3	Auswerten	75
1.46	SQL Statements anzeigen	75
1.47	Logging / Nologging	77
1.48	HugePages	77
1.49	ASH	79
1.50	Schema Größe	79
1.51	Listener	80
1.51.1	Logging Disable	80
1.51.2	TNS_ADMIN	80
1.51.3	IPC Protokoll	80
1.52	Invalid Objects	80
1.52.1	Anzeigen	80
1.53	DB Console	81
1.53.1	Enterprise Agent	81
1.54	Change Schema	81
1.55	Alert Log	81
1.56	Resource Limits	82
1.57	Dispatcher	82
2	Errors & Error Code	85
2.1	ORA-00020	85
2.2	ORA-00392	85
2.3	ORA-609	85
2.4	ORA-01110 / ORA-01113	86
2.5	ORA-01187	86
2.6	ORA-01536	86
2.7	ORA-01555	86
2.8	ORA-01591	87
2.9	ORA-01624	87
2.10	ORA-02030	87
2.11	ORA-03135	88
2.12	ORA-04061 / ORA-04068	88
2.13	ORA-12514	88
2.14	ORA-12518	88
2.15	ORA-12577	89
2.16	ORA-12637	89
2.17	ORA-12638	89
2.18	ORA-39127	89
2.19	TNS-00505	89
2.20	Library Cache	90

Vorwort

Beruflich beschäftige ich mich mit Oracle seit 1999. In dieser Zeit habe ich die Oracle Kurse Oracle Admin I und II, den Oracle Dataguard und Oracle Rman belegt.

Alle Informationen, die ich zusammentragen konnte in der Zeit, sind in diesem Buch eingeflossen. Ebenso sind meine Erfahrungen mit Oracle auch in diesem Buch enthalten.

Hauptsächlich habe ich Oracle auf Linux Servern für unseren Kunden betreut. Die Informationen können auch für Oracle auf Windows übernommen werden.

Diese Dokumentation wurde für die Oracle Datenbanken 10G R2 und 11G R1 geschrieben und auch getestet. Alles sollte auch unter Oracle 12c und auch 18c funktionieren.

Meinen Dank gilt vor allen meiner Familie, die mich bei dem erstellen des Buches immer unterstützt haben.

Bei Fragen und Anregungen bin ich unter der folgenden Mail Adresse zu erreichen:



Kapitel 1

Verwaltung

1.1 SQLPLUS

1.1.1 Konfiguration

Wenn eine SQLPLUS Session gestartet wird, so wird die Datei `glogin.sql` ausgeführt, die sich im Verzeichnis `$ORACLE_HOME/sqlplus/admin` befindet. Gibt es in diesem Verzeichnis auch noch eine `login.sql`, so wird diese auch geladen.

```
uws@tux>cat glogin.sql
prompt
prompt Loading glogin.sql ...
prompt
--
--  Laden von SQL Scripten
--
@/daten/oracle/sqlplus/config/MeinScript.sql
--
--  User settings
--
alter session set nls_date_format='DD.MM.YYYY HH24:MI:SS';

set serveroutput on
set trimspool on
set linesize 200
set pagesize 80
set SQLPLROMPT "_User'@'_CONNECT_IDENTIFIER>"

define _EDITOR=vi

col osuser for a15 heading 'Username'
col username for a15 heading 'Oracle Name'
col client_info for a20
col machine for a25 heading 'Computer Name'
col member for a35
col program for a20 heading 'Programm'
--
--  for Show Parameter
--
col name_col_plus_show_param for a36 heading 'Name'
col value_col_plus_show_param for a30 heading 'Value'
--
--  for Show SGA command
--
col name_col_plus_show_sga for a24
col units_col_plus_show_sga for a15
```

Listing 1.1: Beispiel `glogin.sql`

Folgende Variablen gibt es für den Sqlprompt.

Variable	Beschreibung
_CONNECT_IDENTIFIER	Oracle SID
_DATE	Datum
_O_RELEASE	Release Nummer der Datenbank
_PRIVILEGE	Privilegien
_SQLPLUS_RELEASE	Release Nummer von Sqlplus
_USER	Oracle User

Tabelle 1.1: Sqlprompt Variablen

Beispiel für die Anzeige der Versionsnummer.

```
define _CURRVER=NN
col ver noprint new_value _CURRVER
set termout off
SELECT
  substr(version, 1, instr(version, '.', 1, 2) -1) AS ver
FROM
  product_component_version
WHERE
  product like 'Oracle Database%';
set termout on
set SQLPROMPT "<'<'_CURRVER'>:'_USER'@'_CONNECT_IDENTIFIER">"
```

Listing 1.2: Beispiel glogin.sql

Anstelle der substr Anweisung kann auch nur die Abfrage `version as ver` stehen. Alle set Optionen mit den Einstellungen kann man sich mit `show all` anzeigen lassen.

```
sql>show all

appinfo is OFF and set to "SQL*Plus"
arraysize 15
autocommit OFF.
.
```

Listing 1.3: Shwo all

Den Hostnamen in dem Sqlprompt kann man folgendermaßen realisieren.

```
column host new_value host
set termout off
SELECT host_name host FROM v$instance;
set termout on
set SQLPROMPT "&host' : '_USER'@'_CONNECT_IDENTIFIER">"
```

Listing 1.4: Hostname

Informationen über die aktive Sqlplus Session wird im nächsten Beispiel gezeigt.

```
set feedback off
set heading off

SELECT '***** Current connection details *****' FROM dual
UNION ALL
SELECT 'Starting: ' || to_char(sysdate, 'DD.MM.YYYY HH24:MI:SS') FROM dual
UNION ALL
SELECT 'Language: ' || SYS_CONTEXT('USERENV', 'LANGUAGE') FROM dual
UNION ALL
SELECT 'SID      : ' || SYS_CONTEXT('USERENV', 'SID') FROM dual
UNION ALL
```

```

SELECT 'DB User : '||SYS_CONTEXT('USERENV', 'SESSION_USER') FROM dual
UNION ALL
SELECT 'Instance: '||SYS_CONTEXT('USERENV', 'INSTANCE_NAME') FROM dual
UNION ALL
SELECT 'Server : '||SYS_CONTEXT('USERENV', 'SERVER_HOST') FROM dual
UNION ALL
SELECT 'Database: '||SYS_CONTEXT('USERENV', 'DB_UNIQUE_NAME') FROM dual
UNION ALL
SELECT '***** Current connection details *****' FROM dual;

set feedback on
set heading on

uws@tux>sqlplus /nolog
SQL*Plus: Release 11.2.0.2.0 Production on Di Okt 23 11:27:10 2018

Copyright (c) 1982, 2014, Oracle. All rights reserved.

SP2-0640: Not connected
@>connect hr/hr
***** Current connection details *****
Starting: 23.10.2018 11:27:33
Language: GERMAN_GERMANY.AL32UTF8
SID      : 49
DB User  : HR
Instance: xe
Server   : tux
Database: XE
***** Current connection details *****
HR@XE>

```

Listing 1.5: Informationen

1.1.2 SqlPlus Einstellungen sichern

Hat man in einer Sqlplus Session die Konfiguration abgeändert und möchte diese Einstellungen in einer Datei abspeichern, so geschieht das mit dem Befehl `store`. Diese Datei kann dann jederzeit geladen werden und die SqlPlus Einstellungen werden wieder auf dem stand zurückgesetzt.

```

sql>STORE set /home/uws/sql/sqlsession.sql replace

sql>@/home/uws/sql/sqlsession

```

Listing 1.6: Sqlplus Einstellungen speichern

1.2 Connect

Das Anmelden an der Datenbank kann mit dem Konsolenbefehl `sqlplus` erfolgen.

```
oracle@woby1002>sqlplus / as sysdba

SQL*Plus: Release 11.2.0.2.0 Production on Do Sep 13 13:02:41 2018

Copyright (c) 1982, 2010, Oracle. All rights reserved.

Verbunden mit:
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
SYS@cad10>
```

Listing 1.7: Connect to DB

Unter Windows muss vorher die Variable `ORACLE_SID` gesetzt werden.

```
C:\>set ORACLE_SID=cad10
C:\>sqlplus / as sysdba
```

Listing 1.8: Windows connect

Von einem Client aus muss die `ORACLE SID` mit angegeben werden.

```
oracle@woby1003>sqlplus sys@cad10 as sysdba
```

Listing 1.9: Client Connection

Das verbinden mittels Easy Connect erfolgt mit der Syntax: `sqlplus <name>/<passwd>@<host>:<port>/<oracle_si`

```
oracle@woby1002>sqlplus sys/oracle@woby1000:1521/pds
```

Listing 1.10: Easy Connect

Wurde ein Domain User in Oracle angelegt, erfolgt die Anmeldung folgendermaßen.

```
C:\>sqlplus /@cad10
```

Listing 1.11: Domain User connect

Sollte die Anmeldung mit dem Domain User mit der Fehlermeldung `ORA-01017: invalid username/password; logon denied` abgelehnt werden, so ist der Parameter `os_authent_prefix` zu überprüfen. In diesem Parameter darf sich kein Wert befinden.

```
sql>show parameter os_authent_prefix

NAME                                TYPE        VALUE
-----                                -
os_authent_prefix                    string      OPS$

sql>alter system set os_authent_prefix = '' scope=spfile;
sql>startup force
```

Listing 1.12: Check Parameter

1.3 Datenbank starten manuell

```
oracle@woby1002>sqlplus /nolog
@>connect sys/<passwd> as sysdba
SYS@cad10>startup;
```

Listing 1.13: Start DB

Wird nur startup eingegeben, so wird die Instance gestartet, die mit ORACLE_SID gesetzt worden ist.

Syntax von Startup:

startup [optionen[optionen...]] Ist die datenbank mit einem startup Befehl in den Zustand nomount

<i>Option</i>	<i>Beschreibung</i>
restrict	Datenbank nur für Benutzer mit Restrict Session Priveleg zugänglich
force	Shutdown und Neustart der Instance. Der Shutdown erfolgt als abort
pfile=filespec	Startet die Instance mit der angegebenen Init.ora Datei
mout[dbname]	Mounted die DB
open[dbname]	Öffnet die DB
recover[dbname]	Führt zunächst ein vollständiges Recovery durch
nomount	Nicht Mounten
quit	Unterdrückt die Ausgabe der System Global Area Informationen der Instanz
upgrade	Startet die Datenbank in Open upgrade Modus und setzt System Initialisierungs Parameter, die für Update Scripts gebraucht werden. Diese Option wird gebraucht, wenn die Datenbank das erste mal gestartet wird unter einer neuenVersion
downgrade	Startet die Datenbank in Open downgrade Modus und setzt System Initialisierungs Parameter, die für Update Scripts gebraucht werden

Tabelle 1.2: Optionen Startup

oder mount gebracht worden, kann im Anschluss der Zustand nur mit dem alter database Befehl geändert werden.

```
sys@cad10>startup mount;
sys@cad10>alter database <dbname> mount;
sys@cad10>alter database <dbname> open;
```

Listing 1.14: Alter Database

1.4 Datenbank von Hand herunterfahren

```
oracle@woby1002>sqlplus /nolog
@>connect sys/<passwd> as sysdba
SYS@cad10>shutdown immediate;
```

Listing 1.15: Stop DB

<i>Befehl</i>	<i>Beschreibung</i>
shutdown [normal]	Wartet auf alle angemeldeten User, Verhindert neue 'Connects', Close und Disconnect der Datenbank, Instance Shutdown, Konsistenter Zustand der Datenbank
shutdown immediate	Aktuelle Befehle werden abgearbeitet, weitere Befehle nicht mehr möglich, Pmon beendet alle Sessions und führt ein Rollback aller offenen Transaktion durch Close und Dismount der Datenbank, Instance Shutdown, Konsistenter Zustand der Datenbank
shutdown abort	Absolutes Shutdown ("Nothalt"), kein Rollback, Inkonsistenter Zustand der Datenbank
shutdown transactional	Der Abschluß von Transactionen wird zugelassen

Tabelle 1.3: Optionen Shutdown

	<i>Abort</i>	<i>Immediate</i>	<i>Transactional</i>	<i>Normal</i>
Lässt neue Anmeldungen zu	Nein	Nein	Nein	Nein
Wartet, bis aktuelle Sessions beendet sind	Nein	Nein	Nein	Ja
Wartet, bis aktuelle Transaktionen beendet sind	Nein	Nein	Ja	Ja
Erzwingt Checkpoint und schließt Dateien	Nein	Ja	Ja	ja

1.5 Automatisches Shutdown und Startup

Wurde das Betriebssystem (Oracle Linux, SLES) mit dem Package Oracle installiert, so geschieht das Starten und Anhalten der Datenbank automatisch. Für das Starten / Herunterfahren ist das Script oracle im Verzeichnis /etc/init.d zuständig. Anschließend wird die Datei /etc/profile.d/oracle.sh, bei einer C-Shell die Datei oracle.csh, ausgewertet. In dieser Datei sollte das ORACLE_HOME Verzeichnis definiert sein. Ist ORACLE_HOME nicht richtig definiert, so wird die Datei /etc/oratab ausgelesen.

Wurde das Betriebssystem ohne das Package Oracle installiert, so muss man sich selbst um ein Start / Stop Script kümmern. Im Verzeichnis /etc/init.d wird die Datei dbora erstellt mit nachstehendem Inhalt.

```
oracle@woby1002>cat /etc/init.d/dbora
#!/bin/env bash
#
### BEGIN INIT INFO
# PROVIDES: oracle
# Required-Start: $network $syslog $remote_fs raw
# Should-Start: ocfs2 sshd
# Required-Stop:
# Default-Start: 3 5
# Default-Stop: 0 1 2 6
# Description: Start / Stop the Oracle Database
### END INIT INFO

ORA_HOME=/u01/app/oracle/product/11.1.0/db_1
ORA_OWNER=oracle
LOG=/var/log/oracle.log
export ORA_HOME ORA_OWNER
```

```

if [ ! -f $ORA_HOME/bin/dbstart ]
then
echo "Oracle startup: cannot start."
exit
fi
case "$1" in
'start')
echo "" >> ${LOG}
echo "#####" >> ${LOG}
date +" %T %a %d.%m.%Y : Start the Oracle Database" >> ${LOG}
echo "#####" >> ${LOG}
echo "" >> ${LOG}
su - $ORA_OWNER -c "${ORA_HOME}/bin/dbstart ${ORA_HOME}" >> ${LOG}
su - $ORA_OWNER -c "${ORA_HOME}/bin/emctl start dbconsole" >> ${LOG}
su - $ORA_OWNER -c "${ORA_HOME}/bin/lsnrctl start" >> ${LOG}
touch /var/lock/subsys/dbora
echo "" >> ${LOG}
echo "#####" >> ${LOG}
date +" %T %a %d.%m.%Y : Finished." >> ${LOG}
echo "#####" >> ${LOG}
echo "" >> ${LOG}
;;
'stop')
echo "" >> ${LOG}
echo "#####" >> ${LOG}
date +" %T %a %d.%m.%Y : Stop the Oracle Database" >> ${LOG}
echo "#####" >> ${LOG}
echo "" >> ${LOG}
su - $ORA_OWNER -c "${ORA_HOME}/bin/lsnrctl stop" >> ${LOG}
su - $ORA_OWNER -c "${ORA_HOME}/bin/emctl stop dbconsole" >> ${LOG}
su - $ORA_OWNER -c "${ORA_HOME}/bin/dbshut ${ORA_HOME}" >> ${LOG}
rm -f /var/lock/subsys/dbora
echo "" >> ${LOG}
echo "#####" >> ${LOG}
date +" %T %a %d.%m.%Y : Finished." >> ${LOG}
echo "#####" >> ${LOG}
echo "" >> ${LOG}
;;
esac

```

Listing 1.16: dbora

Mit dem Befehl `chmod 775 /etc/init.d/dbora` werden die Rechte für diese Datei gesetzt. Das Verlinken für die einzelnen Runlevels erfolgt mit dem Befehl `insserv dbora`. In der Datei `/etc/oratab` muss folgender Eintrag vorhanden sein.

```

oracle@woby1002>cat /etc/oratab
<ORACLE_SID>:/u01/app/oracle/product/11.1.0/db_1:Y

```

Listing 1.17: oratab

Wurde der Server mit dem Package Oracle installiert, so muss man in der Datei `/etc/sysconfig/oracle` die folgenden Variablen anpassen.

```

START_ORACLE_DB="YES"
START_ORACLE_DB_LISTENER="YES"

```

Die `oratab` Datei muss wie oben beschrieben aussehen.

1.6 Instanz

1.6.1 Überprüfen

```
sql>SELECT instance_name, host_name, status FROM v$instance;
```

INSTANCE_NAME	HOST_NAME	STATUS
cad01	woby1002	open

Listing 1.18: Check Instanz

1.6.2 Löschen einer Instanz (Windows)

```
C:\>oradim -delete -sid myoracle
```

Listing 1.19: Delete Instanz

1.7 SGA & PGA

1.7.1 SGA abfragen

Die Werte der SGA kann man mit dem nachfolgenden Befehlen abgefragt werden. Mit dem View v\$sgastat werden die Statistiken der SGA angezeigt.

```
sql>SHOW PARAMETER sga_
```

NAME	TYPE	VALUE
sga_max_size	big integer	612M
sga_target	big integer	608M

```
sql>SHOW sga
```

Total System Global Area	640294912 bytes
Fixed Size	1301784 bytes
Variable Size	339739368 bytes
Database Buffers	293601280 bytes
Redo Buffers	5652480 bytes

```
sql>SELECT * FROM v$sga
```

Name	Value
Fixed Size	1301784
Variable Size	339739368
Database Buffers	293601280
Redo Buffers	5652480

```
sql>SELECT * FROM v$sgainfo;
```

NAME	BYTES	RES
Fixed SGA Size	1301784	No
Redo Buffers	5652480	No
Buffer Cache Size	293601280	Yes
Shared Pool Size	314572800	Yes
Large Pool Size	8388608	Yes
Java Pool Size	12582912	Yes
Stream Pool Size	4194304	Yes
Shared IO Pool Size	0	Yes
Granule Size	4194304	No
Maximum SGA Size	640294912	No
Startup overhead in Shared Pool	46127344	No
Free SGA Memory Available	0	

```
sql>SELECT * from v$sgastat ORDER BY pool;
```

POOL	NAME	Bytes
shared pool	Event statistics ptr area	680
streams Pool	free memory	4194304
	Log_buffer	5652480
	fixed area	1301784
	buffer cache	293601280

765 Zeilen ausgewählt

Listing 1.20: SGA Info

1.7.2 PGS abfragen

Die PGA werte werden mit dem nachfolgenden Befehl angezeigt.

```
sql>SHOW PARAMETER pga
NAME                                TYPE                                VALUE
-----                                -
pga_aggregate_target                big integer                        202M
```

Listing 1.21: PGA Info

1.7.3 PGA Optimale Größe ermitteln

Um die optimale Größe der PGA festlegen zu können, kann man mit der Hilfe des Views v\$pga_target_advice sich die Informationen anzeigen lassen.

```
sql>SELECT round(pga_target_for_estimate/1024/1024) target_mb ,
2>estd_pga_cache_hit_percentage cache_hit_perc ,
3>estd_overalloc_count
4>FROM v$pga_target_device;

TARGET_MB  CACHE_HIT_PERC  EST_OVERALLOC_COUNT
-----  -
          25             90                9
          51             90                9
         101             90                9
         152             90                9
         202             100               9
         242             100               9
         283             100               9
         323             100               9
```

Listing 1.22: PGA optimale Größe

Der optimale Wert in der obigen Auflistung ist 202MB. Eine Erhöhung auf 242MB würde keine Steigerung bringen. Eine Verringerung auf 152MB würde ein Verlust des Cache_Hit_Perc auf 90 zu folge haben.

1.7.4 SGA / PGA anpassen

Die Werte für die SGA_TARGET können nicht größer sein, als der Wert der in der SGA_MAX_SIZE definiert worden ist. Die SGA_TARGET kann zur Laufzeit vergrößert werden. Soll der Wert größer als die SGA_MAX_SIZE sein, so muss erst dieser Wert verändert werden. Hierzu ist ein Neustart der Datenbank nötig, da es sich hierbei um einen Initialisierungs Parameter handelt.

```
sql>ALTER SYSTEM SET sga_max_size = 800m comment='Affect after restart'
      scope=spfile;

sql>STARTUP force;

sql>ALTER SYSTEM SET sga_target = 700m;
```

Listing 1.23: SGA setzten

Der Wert für PGA_AGGREGATE_TARGET kann zur Laufzeit der Datenbank abgeändert werden.

```
sql>ALTER SYSTEM SET pga_aggregate_target = 210m;
```

Listing 1.24: PGA setzten

Hinweis:

F2ur 32bit Betriebssysteme kann und darf die sga_max_size nicht größer als

1.7.5 Dynamic Resize Views

In der nachfolgenden Tabelle sind die Views aufgelistet, die Informationen über die Dynamic SGA resize operations enthalten.

<i>View Name</i>	<i>Beschreibung</i>
v\$sga_current_resize_ops	Informationen über die SGA resize operations, die zur Zeit laufen. Als operations kann ein grow oder ein shrink ausgeführt werden
v\$_sga_resize_ops	Informationen über die letzten SGA resize operations
v\$sga_dynamic_components	Informationen über die dynamischen Components in der SGA. Hier sind alle Informationen seit dem starten enthalten.
v\$sga_dynamic_free_memory	Informationen über die Summe der SGA memory available für zukünftige dynamische SGA resize operations
v\$memory_dynamic_components	Wie v\$sga_dynamic_components, zusätzlich mit SGA und PGA Informationen

Tabelle 1.4: Dynamic Resize Views

1.8 Archivelog Modus

1.8.1 Modus Prüfen

Den Archivelog Modus kann man sich mit einer Abfrage anzeigen lassen.

```
sql>SELECT name, log_mode FROM v$database;
```

```
NAME      LOG_MODE
-----  -
CAD10     ARCHIVELOG
```

Listing 1.25: Archivelog Modus prüfen

1.8.2 Archivelog Einschalten

Der Archive Modus kann nur eingeschaltet werden, wenn die Datenbank sich im mount Status befindet. Wenn man den Archive Modus eingeschaltet hat, so sollte man regelmäßig ein Backup durchführen, da sonst die Festplatte schnell voll läuft. Es ist daher ratsam, ein Full Backup der DB durhzuführen. Informationen über ein Backup gibt es in der Dokumentation, Backup & Restore.

```
sql>SHUTDOWN immediate;

sql>STARTUP mount;

sql>ALTER DATABASE archivelog;

sql>ALTER DATABASE open;

sql>archive log list;
Datenbank-Log_Modus           Archive-Modus
Automatische Archivierung     Aktiviert
Archivierungsziel             USE_DB_RECOVERY_FILE_DEST
Aelteste Online-Log-Sequenz   3487
Naechste zu archivierende Log-Sequence 3489
Aktuelle Log-Sequenz          3489
```

Listing 1.26: Archivelog einschalten

1.8.3 Log Switch

Einen Log Switch kann man mit `switch logfile` durchführen.

```
sql>ALTER SYSTEM switch logfile;
```

Listing 1.27: Log Switch

1.8.4 Archivelog ausschalten

Den Archivelog Modus kann man nur im mount Status der Datenbank abschalten.

```
sql>SHUTDOWN immediate;
sql>STARTUP mount;
sql>ALTER DATABASE noarchivelog;
sql>ALTER DATABASE open;
```

Listing 1.28: Archivelog ausschalten

1.8.5 Archivelog anzeigen

Informationen über die Archivelogs kann man sich mit denfolgenden Abfragen sich anzeigen lassen.

```
sql>SELECT recid, stamp, thread#, sequence#, applied, archived,
2>FROM v$archived_log order by sequence#;
```

RECID	STAMP	THREAD#	SEQUENCE#	S	APP	ARC	S
254129	805444382	1	127096	A	YES	YES	A
254130	805444383	1	127097	A	YES	YES	A

Listing 1.29: PGA setzten

1.8.6 Archivelog Format

Mit dem Initialisierungs Parameter `LOG_ARCHIVE_FORMAT` wird der Name des Archivelogs festgelegt. In Oracle 12c müssen die Variablen `%s`, `%r` und `%t` genommen werden.

Variable	Beschreibung
<code>%s</code>	Log Sequence Nummer
<code>%S</code>	Log Sequence Nummer, mit Nullen ausgefüllt
<code>%t</code>	Thread Nummer
<code>%T</code>	Thread Nummer, mit Nullen ausgefüllt
<code>%a</code>	Activation ID
<code>%d</code>	Database ID
<code>%r</code>	Resetlog ID

Tabelle 1.5: Formatierung

1.9 Flashback

1.9.1 Flashback einschalten

Für ein Flashback Database muss der Flashback Modus eingeschaltet werden. Dieses kann man nur durchführen, wenn sich die Datenbank im Archive Modus befindet. Ab der Version 11GR2 braucht man die Datenbank nicht mehr herunterfahren, um den Modus anzuschalten. Es reicht hierfür das `alter database flashback on`. Schaltet man den Flashback Modus an, so wird ausreichend Platz in der Flash Recovery Area benötigt.

```
sql>SHUTDOWN immediate;

sql>STARTUP mount

sql>ALTER DATABASE flashback on;

sql>ALTER DATABASE open;
```

Listing 1.30: Flashback einschalten

1.9.2 Flashback Status abfragen

```
sql>SELECT flashback_on FROM v$database;

FLASHBACK_ON
-----
YES
```

Listing 1.31: Flashback Status

1.9.3 Flashback Nummer abfragen

```
sql>SELECT oldest_flashback_scn, oldest_flashback_time FROM
v$flashback_database_log;

OLDEST_FLASHBACK_SCN OLDEST_FLASH
-----
351592 11-OCT-11

sql>SELECT * FROM v$flashback_database_stat;

BEGIN_TIME END_TIME FLASHBACK_DATA DB_DATA REDO_DATA ESTIMATED
-----
11-OCT-11 11-OCT-11 491520 507984 751392 0

sql>SELECT current_scn FROM v$database;

CURRENT_SCN
-----
352705
```

Listing 1.32: Flashback Nummer

1.9.4 Flashback Size

```
sql>SELECT estimated_flashback_size/1024/1024 "Size in MB"
2>FROM v$flashback_database_log;

Size in MB
-----
7077,7256
```

Listing 1.33: Flashback Size

1.9.5 Restore Point abfragen

```
sql>SELECT scn, time, restore_point_time
2>FROM v$restore_point;
```

Listing 1.34: Restore Point

1.9.6 Flashback ausschalten

Um den Flashback Modus wieder abzuschalten, muss die Datenbank heruntergefahren werden. Anschließend wird die Datenbank im mount Status geöffnet. Ab der Version 11G kann man den Modus ohne die Datenbank herunterfahren zu müssen mit `alter database flashback off` ändern.

```
sql>shutdown immediate

sql>startup mount

sql>ALTER database flashback off;

sql>ALTER database open;
```

Listing 1.35: Flashback ausschalten

1.9.7 Delete Old Flashback Files

Normalerweise werden die Flashback Dateien automatisch gelöscht, spätestens wenn die Flash Recovery Area an die 100% Belegung geht. Möchte man aber per Hand die alten Flashback Dateien löschen, so ist der Parameter `db_recovery_file_dest_size` auf die Größe zu setzen, die die Summe der Archivelogs und Backup Sets Größe ist.

Ist der Wert von `db_recovery_file_dest_size` auf 100G gesetzt und die Archivelogs und Backup Sets belegen zusammen 20%, so kann als neuer Wert 25G genommen werden. Die alten Flashback Dateien werden dann automatisch gelöscht. Danach kann der alte Wert wieder eingestellt werden.

```
sql>SELECT file_type, number_of_files
2>FROM v$flash_recovery_area_usage
3>WHERE file_type in ('ARCHIVED LOG', 'BACKUP PIECE', 'FLASHBACK LOG');
```

FILE_TYPE	NUMBER_OF_FILES
ARCHIVED LOG	17
BACKUP PIECE	27
FLASHBACK LOG	526

```
sql>ALTER system SET db_recovery_file_dest_size = 25G;

sql>SELECT file_type, number_of_files
2>FROM v$flash_recovery_area_usage
3>WHERE file_type in ('ARCHIVED LOG', 'BACKUP PIECE', 'FLASHBACK LOG');
```

FILE_TYPE	NUMBER_OF_FILES
ARCHIVED LOG	17
BACKUP PIECE	27
FLASHBACK LOG	248

Listing 1.36: Flashback Dateien löschen

1.10 Control Files

1.10.1 Verwalten der Control Files

Die Anzahl und wo sich die Control Files sich befinden kann man mit verschiedenen Abfragen sich anzeigen lassen.

```

sql>col name for a50
sql>SELECT * from v$controlfile;

STATUS NAME                                IS_ BLOCK_SIZE FILE_SIZE_BLK
-----
      /u02/oracle/db/cad01/control.dbf NO          16384          594

sql>SHOW parameter control

NAME                                TYPE                VALUE
-----
control_file_record_keep_time      integer             7
control_files                       string              /u02/oracle/db/cad01/
control_management_pack_access     string              NONE

sql>SELECT name, value FROM v$parameter WHERE name = 'control_files';

NAME                                VALUE
-----
control_files /u02/oracle/db/cad01/control.dbf

```

Listing 1.37: Control Files anzeigen

1.10.2 Sicherung der Control Files

Sicherung der Control Dateien als Sql-Script. Ohne die Angabe von as wird die Datei im udump Verzeichnis abgelegt.

```

sql>ALTER database BACKUP controlfile to trace as
2>' /u01/app/oracle/admin/cad01/backup/crctl.sql' reuse;

```

Listing 1.38: Backup als Sql-Script

Eine Binär Kopie wird mit dem folgenden Statement erstellt.

```

sql>ALTER database BACKUP controlfile as
2>' /u01/app/oracle/admin/cad01/backup/control.bkp';

```

Listing 1.39: Backup als Binär Datei

1.10.3 Zusätzliche Control-Dateien

Das einrichten zusätzlicher Control-Dateien wird mit den nachfolgenden Schritten gemacht.

- In einer SQLplus Session folgenden Beehl absetzen:

```
sql>ALTER system SET control_files =
2>' /u02/oracle/db/cad01/control.dbf ',
3>' /u02/oracle/db/cad01/control1.dbf ' scope=spfile;
```

Listing 1.40: Control File

- Shutdown der Instance
- Mit dem Explorer eine vorhandene Control-Datei kopieren.
- Starten der Instance

Tip:

Bevor man Änderungen an der Datenbank vornimmt, ist es zu Empfehlen, die Controlfiles mit einem Backup zu sichern.

1.10.4 Vorhalte Zeit

Der Wert gibt die Tage an, wie lange Daten im Controlfile vorgehalten werden.

```
sql>ALTER system SET control_file_record_keep_time = 30;
```

Listing 1.41: Vorhalte Zeit

1.10.5 Restore

Mit einem Recovery Catalog kann man die Control Files einfacher zur $\frac{1}{4}$ ckspielen. Alle Dateien werden in dem Platz hergestellt, die im Parameter control_files stehen. Wie man die Control Files ohne Recovery Catalog wieder herstellt, wird im Buch Backup & Restore beschrieben.

```
sql>STARTUP nomount

sql>exit

uws@tux>rman target / catalog catdb/catdb@cat01
rman>restore controlfile;
```

Listing 1.42: Restore

1.11 SPFile wieder herstellen

Hat man eine Änderung an der Datenbank vorgenommen und sie startet nicht mehr, so kann man einen PFile erstellen und dann daraus wieder einen SPFile. Vorher wird der PFile bearbeitet. Den SPFile kann man nicht mit einem Text Editor bearbeiten, da dieser File in einem Binär Format vorliegt. Folgendes Szenario beschreibt den Fehler und deren Behebung.

```

sql>ALTER system SET log_archive_dest = '/u03/oracle/archive' scope = spfile
;

System altered.

sql>shutdown immediate
Database closed.
Database dismounted.
Oracle Instance shutdown.

sql>startup mount
ORA-16032: parameter LOR_ARCHIVE_DEST destination string cannot be
  tztranslated

sql>CREATE pfile='/u01/app/oracle/product/12.0.1/db_1/dbs/pfilecad01.ora'
2>FROM spfile;
File created

sql>host
uws@tux>vi /u01/app/oracle/product/12.0.1/db_1/dbs/pfilecad01.ora

Die Zeile mit LOG\verb|_|ARCHIVE\verb|_|DEST entfernen.

uws@tux>exit

sql>CREATE spfile='/u01/app/oracle/product/12.0.1/db_1/dbs/spfilecad01.ora'
2>FROM pfile='/u01/app/oracle/product/12.0.1/db_1/dbs/pfilecad01.ora';
File created

sql>startup open
Oracle instance startet.
Total System Global Area   318046208 bytes
Fixed Size                   1299652 bytes
Variable Size               184552252 bytes
Database Buffers            125829120 bytes
Redo Buffers                 6365184 bytes
Datenbank mounted
Datenbank open

```

Listing 1.43: Beispiel

Wenn die Datenbank nochläuft, so kann man sich einen pfile / spfile aus dem Speicher erstellen.

```

sql>CREATE spfile FROM memory;

sql>CREATE pfile='/u01/app/oracle/product/12.0.1/db_1/dbs/pfilecad01.ora'
2>FROM memory

```

Listing 1.44: Memory

1.12 Redo Logs

1.12.1 Informationen abfragen

Die Größe der Standby Logfiles können mit dem View `v$standby_log` abgefragt werden.

```
sql>SELECT group#, status, sequence#, bytes/1024/1024 "MB"
2>FROM v$standby_log order by group#;
```

GROUP#	STATUS	SEQUENCE#	MB
1	ACTIVE	10655	50
2	CURRENT	10656	50
3	INACTIVE	10654	50

Listing 1.45: Standby Logfiles

Auch die Größe der Redo Logs kann mit dem View `v$log` abgefragt werden.

```
sql>SELECT group#, status, sequence#, bytes/1024/1024 "MB"
2>FROM v$log order by group#;
```

GROUP#	STATUS	SEQUENCE#	MB
1	ACTIVE	1541	50
2	CURRENT	1542	50
3	ACTIVE	1540	50

Listing 1.46: Redo Log

Eine History für die Logfile Switches kann man mit dem View `v$loghist` abfragen.

```
sql>SELECT * from v$loghist
```

Listing 1.47: Log History

Eine Auflistung über die vorhandenen Redo Logs stehen in dem View `v$logfile`.

```
sql>SELECT * FROM v$logfile ORDER BY group#;
```

GROUP#	STATUS	TYPE	MEMBER	IS_
1	Online		/u01/oradata/cad01/redo01.log	No
1	Online		/u02/oradata/cad01/redo01a.log	No
2	Online		/u02/oradata/cad01/redo02.log	No
2	Online		/u02/oradata/cad01/redo02a.log	No

```
sql>col "Archive Dest" for a50
```

```
sql>SELECT dest_id "ID",
2>status "Status",
3>destination "Archive Dest",
4>error "Error"
5>FROM v$archive_dest
6>WHERE status = 'VALID';
```

ID	STATUS	ARCHIVE DEST	ERROR
1	VAILD	USE_DB_RECOVERY_FILE_DEST	
2	VALID	dg_cad10	
32	VALID	USE_DB_RECOVERY_FILE_DEST	

Listing 1.48: Redo Log Files

Folgende Views gibt es für die Redo Logs.

View Name
v\$archived_log
v\$archive_dest
v\$archived_processes
v\$backup_redolog
v\$log
v\$log_history
v\$logfile

Tabelle 1.6: Formatierung

1.12.2 Anlegen weiterer Redo Log Dateien

Der neue Dateiname wird im Control File eingetragen, daher sollte der Name mit absoluten Pfad angegeben werden, um ein problemloses Auffinden zu garantieren.

Gibt es für die Datenbank eine Standby Datenbank, so müssen auf beiden Seiten die Redo Logs angelegt werden. Vor dem erstellen / löschen sollte der Parameter `standby_file_management` auf `manual` gestellt werden.

```

sql>ALTER system SET standby_file_management = manual;
sql>ALTER database ADD standby logfile
  2>' /u02/oracle/oradata/cad01/stby05.log' size 100M;
sql>ALTER system SET standby_file_management = auto;

sql>-- Anlegen eines neuen Redo Log File
sql>ALTER database ADD logfile
  2>' /u02/oracle/oradata/cad01/redo01.log' size 50M;

sql>-- Anlegen einer Log File Gruppe
sql>ALTER database ADD logfile
  2>('/u02/oracle/oradata/cad01/redo02.log'
  3>' /u03/oracle/oradata/cad01/redo02a.log') size 50M;

sql>-- An einer vorhandenen Redo Log Gruppe ein weiteres Log File anhaengen
sql>ALTER database ADD logfile member
  2>' /u02/oracle/oradata/cad01/redo01b.log' TO group 1;

sql>-- An einer vorhandenen Standby Redo Log Gruppe ein weiteres Log File
  anhaengen
sql>ALTER database ADD standby logfile member
  2>' /u02/oracle/oradata/cad01/stby04.log' TO group 1;

sql>-- Eine Standby Redo Log Datei anlegen
sql>ALTER database ADD standby logfile group 4
  2>' /u02/oracle/oradata/cad01/stby06.log' size 100M;

```

Listing 1.49: Redo Log anlegen

Nach dem Anlegen / Spiegeln der Log Dateien sollte der nachfolgende Befehl mehrfach ausgeführt werden.

```

sql>ALTER system SWITCH logfile;

```

Listing 1.50: Log Switch

1.12.3 Löschen einer Redo Log Gruppe / Datei

Drop logfile entwertet nur die entsprechenden Einträge in der Control datei, löscht aber nicht die Datei. Eine Gruppe kann nur gelöscht werden, wenn sie nicht aktiv ist. Eventuell muss ein Log Switch gemacht werden.

```
sql>ALTER database DROP logfile group 3;

sql>ALTER database DROP logfile member
2>' /u02/oracle/oradata/cad01/redo01a.log';
```

Listing 1.51: Redo Log löschen

1.12.4 Umbenennen oder Pfad ändern

```
sql>SHUTDOWN immediate;
sql>host
uws@tux>mv /u02/oracle/oradata/cad01/redo01a.log /u03/oracle/oradata/cad01/
redo01a.log
uws@tux>exit
sql>startup mount
sql>ALTER database RENAME file '/u02/oracle/oradata/cad01/redo01a.log'
2>TO '/u03/oracle/oradata/cad01/redo01a.log';
sql>commit;
sql>ALTER database Open;
```

Listing 1.52: Pfad ändern

1.13 Tablespaces

1.13.1 Neuanlegen von Tablespaces

Der `create tablespace` Befehl muss bei geöffneter Datenbank abgesetzt werden. Der angelegte Tablespace ist, falls nicht `Offline` angegeben wurde `Online`. Die Datei muss nicht vorhanden sein, sie wird von Oracle neu angelegt.

```
sql>CREATE tablespace ts1 DATAFILE '/u02/oracle/oradata/cad01/ts1.ora'  
2>SIZE 100m AUTOEXTEND on NEXT 50m MAXSIZE 300m;
```

Listing 1.53: Tablespace anlegen

Die maximale Größe einer Tablespace Datei ist 32GB. Möchte man größere Dateien erstellen, so muss man hierzu die Option `bigfile` angeben. Die Angabe der Größe kann in KB(K), MB(M), GB(G) oder TB(T) erfolgen.

```
sql>CREATE BIGFILE tablespace ts2 DATAFILE '/u02/oracle/oradata/cad01/ts2.  
ora'  
2>SIZE 50G AUTOEXTEND on NEXT 1G MAXSIZE 100G;
```

Listing 1.54: Tablespace als Bigfile anlegen

Einen komprimierten Tablespace kann man mit der Option `default compress` anlegen.

```
sql>CREATE tablespace ts3 DATAFILE '/u02/oracle/oradata/cad01/ts3.ora'  
2>SIZE 100m DEFAULT compress;
```

Listing 1.55: Tablespace compress

1.13.2 Erweitern von Tablespaces

Es gibt drei Möglichkeiten um einen Tablespace zu erweitern.

1. Der Tablespace wird durch zusätzliche Dateien erweitert.

```
sql>ALTER tablespace ts2 ADD DATAFILE  
2>' /u02/oracle/oradata/cad01/ts4.ora' size 100M;
```

Listing 1.56: Zusätzliche Datei

2. Das Datenfile des Tablespaces wird erweitert.

```
sql>ALTER database DATAFILE  
2>' /u02/oracle/oradata/cad01/ts1.ora' resize 500M;
```

Listing 1.57: Datenfile erweitern

3. Das Datenfile des Tablespaces wird automatisch um eine vorgegebene Größe bis zu einer maximalen Größe erweitert.

```
sql>ALTER database DATAFILE '/u02/oracle/oradata/cad01/ts3.ora'  
2>AUTOEXTEND on NEXT 50M MAXSIZE 500M;
```

Listing 1.58: Automatisch vergrößern

1.13.3 Verändern der Default Storage Parameter

Die Default Storage Parameter eines Tablespaces lassen sich mit dem SQL-Befehl `alter tablespace` abändern.

```
sql>ALTER tablespace ts1 DEFAULT STORAGE (maxextents 150);
```

Listing 1.59: Default Storage Parameter

1.13.4 Lage bzw. Umbenennen

Es gibt zwei Möglichkeiten für die Änderungen der Lage bzw. den Dateien einen neuen Namen zu vergeben. 1. Die erste Möglichkeit erfordert, dass vor dem Umbenennen der Datei der Tablespace oder das Datafile offline genommen werden muss.

```
sql>ALTER tablespace ts2 offline;

sql>ALTER tablespace ts2 RENAME datafile
2>' /u02/oracle/oradata/cad01/ts2.ora' TO
3>' /u03/oracle/oradata/cad01/ts2.ora';

sql>ALTER tablespace ts2 online;
```

Listing 1.60: Tablespace offline

2. Die Datenbank muss sich im Mount Status befinden.

```
sql>shutdown immediate;
sql>startup mount;

sql>ALTER tablespace ts2 RENAME datafile
2>' /u02/oracle/oradata/cad01/ts2.ora' TO
3>' /u02/oracle/oradata/cad01/ts3.ora';
sql>commit;
sql>ALTER database open;
```

Listing 1.61: Database Mount Status

Rename ändert nur die Einträge in der Control Datei. Das bedeutet, es muss nach dem Offline / Mount Status die Zielfeile mit dem gleichen Namens und Formats kopiert oder umbenannt werden. Erst danach kann der Tablespace online und die Datenbank geöffnet werden.

1.13.5 Löschen von Tablespace

```
sql>DROP tablespace ts2 [including contents];
```

Listing 1.62: Tablespace löschen

Enthält der Tablespace Datenbankobjekte, muss die Including Contents Option verwendet werden, ansonsten wird der Befehl abgewiesen. Der Drop Tablespace Befehl löscht nicht die Daten Dateien auf OS-Ebene. Es empfiehlt sich, den Tablespace vor dem Drop Befehl offline zu setzen, da der Befehl, solange Benutzer auf dem Tablespace aktiv sind, nicht durchgeführt wird.

Einzelne Daten Dateien eines Tablespaces können nicht gelöscht werden, sondern immer nur ein kompletter Tablespace. Der Drop Befehl wird auch abgewiesen, falls Constraints für Tabellen definiert sind, die Spalten von Tabellen des zu löschenden Tablespaces referenzieren.

1.13.6 Tablespace offline/online setzen

Das Kommando kann nur im geöffneten Zustand der Datenbank abgesetzt werden.

```
sql>ALTER tablespace ts2 offline;

sql>ALTER tablespace ts2 online;
```

Listing 1.63: Tablespace offline/online

1.13.7 Datafile offline/online setzen

Anstelle der Angabe des Datafiles, kann man auch die File_id des Datafiles nehmen.

```
sql>ALTER database datafile
2>' /u02/oracle/oradata/cad01/temp01.dbf' offline;
```

```
sql>ALTER database datafile 15 online;
```

Listing 1.64: Datafile offline/online

1.13.8 Temp Tablespace verkleinern / vergrößern

```
sql>SELECT tablespace_name, file_name FROM dba_temp_files;
```

```
TABLESPACE_NAME FILE_NAME
-----
TEMP            /u02/oracle/oradata/cad01/temp01.dbf
```

```
sql>ALTER database tempfile
2>' /u02/oracle/oradata/cad01/temp01.dbf' RESIZE 20G;
```

Listing 1.65: Resize

Mit diesem Befehl wird der Temp File auf 20GB Größe gesetzt. Es kann auch anstatt der Angabe in Gigabyte die Größe in Megabyte angegeben werden, hierbei wird anstelle des G das M genommen.

Gibt es bei diesem Befehl den Error Code ORA-03297: file contains used data beyond request RESIZE value, so gibt es noch andere Methoden. Für eine Oracle 9i oder kleiner gibt es die folgende Methode.

```
sql>CREATE temporary TABLESPACE temp2 TEMPFILE
2>' /u02/oracle/oradata/cad01/temp02.dbf' SIZE 100M REUSE AUTOEXTEND on
3>NEXT 10M MAXSIZE unlimited EXTEND MANAGEMENT local UNIFORM size 1M;
```

```
sql>ALTER database DEFAULT temporary tablespace temp2;
```

```
sql>DROP tablespace temp INCLUDING contents and datafiles;
```

```
sql>CREATE temporary TABLESPACE temp TEMPFILE
2>' /u02/oracle/oradata/cad01/temp01.dbf' SIZE 500M REUSE AUTOEXTEND on
3>NEXT 100M MAXSIZE unlimited EXTEND MANAGEMENT local UNIFORM size 1M;
```

```
sql>ALTER database DEFAULT temporary tablespace temp;
```

```
sql>DROP tablespace temp2 INCLUDING contents and datafiles;
```

Listing 1.66: Oracle 9i

Für Oracle 9i oder Größer gibt es noch eine weitere Methode, den Tablespace zu verkleinern. Hierbei dürfen aber keine Oracle User angemeldet sein.

```
sql>SELECT tablespace_name, file_name, bytes FROM dba_temp_files;
```

```
TABLESPACE_NAME FILE_NAME BYTES
-----
TEMP            /u02/oracle/oradata/cad01/temp01.dbf 12800000
```

```
sql>ALTER database tempfile '/u02/oracle/oradata/cad01/temp01.dbf'
2>DROP including datafiles;
```

```
sql>ALTER tablespace temp ADD tempfile
2>' /u02/oracle/oradata/cad01/temp01.dbf' SIZE 512m
3>AUTOEXTEND on NEXT 250m MAXSIZE unlimited;
```

Listing 1.67: Oracle >9i

1.13.9 Datafile löschen

Vor dem löschen des Datafiles sollte man das Datafile nicht offline setzen, sonst gibt es bei dem löschen folgende Fehlermeldung.

```

sql>ALTER database datafile '/u02/oracle/oradata/cad01/ts3.ora' offline;

sql>ALTER tablespace users DROP datafile
2>' /u02/oracle/oradata/cad01/ts3.ora';

ORA-03264: Offline-Datendatei von local verwaltetem Tablespace kann nicht
gel"oscht werden.

```

Listing 1.68: Datafile offline

Die Datendatei muss mit dem Befehl `recover datafile` wieder hergestellt werden und dann wieder online genommen zu werden. Nun kann man die Datendatei löschen.

```

sql>RECOVER datafile '/u02/oracle/oradata/cad01/ts3.ora';
Media Recovery abgeschlossen.

sql>ALTER database datafile '/u02/oracle/oradata/cad01/ts3.ora' online;

sql>ALTER tablespace users DROP datafile
2>' /u02/oracle/oradata/cad01/ts3.ora';

```

Listing 1.69: Datafile online

1.13.10 Autoextend Informationen

In der Spalte `increment_by` steht der Wert nicht in Bytes oder in MB, sondern dieser Wert wird mit der Formel `increment_by = (autoextend on size) * 1024 / 8` eingetragen. Möchte man wieder den Wert in MB ausgeben, so muss das mit der Formel `increment_by * 8 / 1024` geschehen.

```

sql>COL tablespace_name FOR a10 HEADING 'Tablespace'
sql>COL file_name FOR a50
sql>COL "File Size" FOR a10
sql>COL "max. Size" FOR a10
sql>COL "erweitern um" FOR a13
sql>SET lin 200
sql>SELECT tablespace_name, file_name,
2>ROUND(bytes/1024/1024) || ' MB' "File Size",
3>ROUND(maxbytes/1024/1024) || ' MB' "max. Size",
4>autoextensible,
5>ROUND(increment_by*8/1024) || ' MB' "erweitern um"
6>FROM dba_data_files ORDER BY tablespace;

```

Listing 1.70: Abfrage Autoextend

1.13.11 Tablespace umwandeln

Man kann aus einem normalen Tablespace einen Temporären Tablespace machen und natürlich aus einem Temporären Tablespace einen normalen Tablespace.

```

sql>ALTER database my_ty temporary;

sql>ALTER database my_temp_ts permanent;

```

Listing 1.71: Tablespace umwandeln

1.13.12 Überwachen von Tablespaces

Folgende Views gibt es für die Anzeige von Informationen über die Tablespaces. In der nächsten Abfrage werden die Tablespaces angezeigt, die vorhanden / verfügbar sind.

View Name	Beschreibung
user_tablespace	Anzeige aller Benutzer Tablespaces
dba_tablespaces	Anzeige aller Tablespaces in der Datenbank
dba_datafiles	Anzeige der Tablespace Datendateien
v\$datafile	Anzeige der Tablespace Datendateien
dba_temp_files	Anzeige der Temporären Tablespaces
dba_free_spaces	Anzeige von Informationen über den freien Platz im Tablespace
filext\$	Anzeige Autoextend ON
dba_segments	Anzeige der Objekte in dem Tablespace

Tabelle 1.7: Tablespace Views

```
sql>SELECT tablespace_name, status, logging, forec_logging, bigfile
2>FROM dba_tablespaces;
```

```
TABLESPACE_NAME STATUS LOGGING FOR BIG
-----
SYSTEM          ONLINE LOGGING NO NO
SYSAUX          ONLINE LOGGING NO NO
UNDOTBS1        ONLINE LOGGING NO NO
TEMP            ONLINE NOLOGGING NO NO
USERS           ONLINE LOGGING NO NO
```

Listing 1.72: Tablespace auflisten

Nun folgt eine Auflistung, ob Tablespaces komprimiert sind.

```
sql>SELECT tablespace_name, def_tab_compression
2>FROM dba_tablespaces;
```

```
TABLESPACE_NAME DEF_TAB
-----
SYSTEM          DISABLED
SYSAUX          DISABLED
```

Listing 1.73: Tablespace komprimiert

Zuordnung Tablespace Namen, Datendateien, Größe und Status.

```
sql>COL file_name FOR a60
sql>COL tablespace_name FOR a10 HEADING 'Tablespace'
sql>SELECT tablespace_name, file_name,
2>ROUND(bytes/1024/1024) "Size in MB", status
3>FROM dba_data_files;
```

```
Tablespace FILE_NAME Size in MB STATUS
-----
TS1 /u02/oracle/oradata/cad01/ts1.ora 100 AVAILABLE
TS2 /u02/oracle/oradata/cad01/ts2.ora 50 AVAILABLE
.
.
```

Listing 1.74: Tablespace Info

Namen der Daten Dateien (in der Control Datei), wichtig für Rename.

```
sql>COL name FOR a60
sql>SELECT * from v$dbfile;
```

```
FILE# NAME
-----
1 /u02/oracle/oradata/cad01/system01.ora
2 /usr/oracle/oradata/cad01/undotbs01.ora
```

·
·

Listing 1.75: Datendatei Namen

Anzeige des freien Speicherplatzes in den Tablespaces.

```
sql>SELECT * FROM dba_free_spaces WHERE tablespace_name = 'USERS';
```

TABLESPACE_NAME	FILE_ID	BLOCK_ID	BYTES	BLOCKS	RELATIVE_FNO
USERS	4	328	102170624	12472	4

Listing 1.76: Tablespace freier Speicher

Informationen über den Temp. Tablespace.

```
sql>SELECT tablespace_name, file_name, round(bytes/1024/1024) "MB"
2>FROM dba_temp_files;
```

TABLESPACE_NAME	FILE_NAME	MB
TEMP	/u02/oracle/oradata/cad01/temp01.ora	1000

Listing 1.77: Temp. Tablespace

Abfrage, wann die Tablespaces erstellt worden sind.

```
sql>SELECT a.tablespace_name, a.file_name, b.creation_time
2>FROM dba_data_files a, v$datafile b
3>WHERE a.file_id=b.file#
4>ORDER BY tablespace_name;
```

Listing 1.78: Erstell Datum

1.14 Rollback Segmente

1.14.1 Anlegen und Verwalten

Seit Oracle 9i sind keine Rollback Segmente mehr nötig. Die Verwaltung erfolgt automatisch. Die hier beschriebene Vorhergehensweise gilt nur bis zu der Oracle Version 8.

Das Anlegen von neuen Rollback Segmenten wird mit dem nachfolgenden Befehl gemacht.

```
sql>CREATE rollback segment rb21 TABLESPACE rbs STORAGE
2>(initial 500k next 500k minextents 5 maxextents 100
3>pctincrease 0 optimal);
```

Listing 1.79: Rollback Segmente anlegen

Parameterwahl:

Parameter	Beschreibung
initial=next pctincrease 0 (default)	Extents sollten die gleiche Größe haben
minextents	ca. 5
maxextents	groß wählen ca. 100
extentgröße	ca 500k - 1m
optimal	Es wird versucht das Rollback Segment in der angegebenen Größe zu halten.

Tabelle 1.8: Parameter Create Rollback Segmente

Nach erfolgreichem create ist das Rollback Segment nicht automatisch aktiviert. Soll das Rollback Segment aktiviert werden, ist folgender Vorgang zu machen.

- Shutdown der Instance
- Eintragen unter ROLLBACK_SEGMENTS in der init.ora Datei.
- Startup der Instance

Alternativ kann das Rollback Segment auch mit einem Befehl online genommen werden.

```
sql>ALTER rollback segment rb21 online;
```

Listing 1.80: Rollback Segment online

1.14.2 Löschen

Folgende Vorhergehensweise ist nötig, damit man Rollback Segmente löschen kann.

- Shutdown der Instance
- Aus ROLLBACK_SEGMENTS in der init.ora Datei austragen.
- Startup der Instance
- sql>DROP rollback segment rb21;

Alternativ dazu kann man das Segment auch in den Modus offline nehmen und anschließend das Rollback Segment löschen.

```
sql>ALTER rollback segment rb21 offline;
sql>DROP rollback segment rb21;
```

Listing 1.81: Löschen

1.14.3 Überwachen

Mit der nächsten Abfrage kann man sich die Rollback Segmente anzeigen lassen.

```
sql>SELECT segment_name , status FROM dba_rollback_segs ;
```

SEGMENT_NAME	STATUS
SYSTEM	ONLINE
__SYSMU037_131194839\$	OFFLINE

Listing 1.82: Rollback Segmente anzeigen

1.15 Deadlocks

1.15.1 Anzeigen und löschen

Für die Anzeige der blockierenden Sessions gibt es folgende Tabellen / Views.

- dba_kgllock
- dba_locks
- dba_lock
- dba_lock_internal
- dba_dml_locks
- dba_ddl_locks
- dba_waiters
- dba_blockers
- v\$session
- v\$locked_object

Mit den nachfolgenden Befehlen kann man sich in SQL die blockierenden Sessions anzeigen lassen, um sie anschließend zu löschen.

```
sql>SELECT sid, serial#, username FROM v$session
2>WHERE sid in (SELECT blocking_session from v$session);

SID SERIAL# USERNAME
-----
144 8982      HR

sql>ALTER system KILL session '144,8982' immediate;
```

Listing 1.83: List Deadlocks

Anstelle die Session zu killen, kann man auch die Session disconnecten.

```
sql>ALTER system DISCONNECT session '<sid>,<serial#>' post_transaction;

sql>ALTER system DISCONNECT session '<sid>,<serial#>' immediate;
```

Listing 1.84: Disconnect

Eine zweite Methode, sich einen Lock anzeigen zu lassen, kann man mit den Views v\$locked_object und v\$session durchführen.

```
sql>SELECT session_id, oracle_username, locked_mode FROM v$locked_object;

SESSION_ID ORACLE_USERNAME LOCKED_MODE
-----
          317 scott                3
          317 scott                2

sql>SELECT serial#, sid, username from v$session WHERE sid in
2>(SELECT session_id from v$locked_object);

SERIAL#    SID ORACLE_NAME
-----
        4305    317 scott
```

Listing 1.85: List Deadlocks

Mit diesen Informationen kann nun der Process gelöscht werden.

```
sql>ALTER system KILL session '317,4305' immediate;
```

Listing 1.86: Kill Process

Anzeige, wer hier wehn blockiert.

```
sql>SELECT * FROM dba_waiters;
```

Listing 1.87: Blocking Session

Wie lange wartet schon die Session.

```
sql>SELECT session_id, last_convert sekunden, last_convert/60 Minuten  
2>FROM dba_locks WHERE session_id in  
3>(SELECT blocking_session from v$session);
```

```
sql>SELECT * FROM v$locked_object;
```

```
sql>SELECT serial#, sid, username FROM v$session WHERE sid in  
2>(SELECT session_id from v$locked_object);
```

Listing 1.88: Dauer Blocking Session

Welche SQL Anweisung wird ausgeführt. Siehe auch Status DML.

```
sql>SELECT sql_text FROM v$sql WHERE sql_id in  
2>(SELECT sql_id FROM v$session WHERE sid in  
3>(SELECT session_id FROM v$locked_object));
```

```
sql>SELECT sql_text, executions FROM v$sqlarea WHERE  
2>UPPER(sql_text) LIKE 'DELETE%' OR  
3>UPPER(sql_text) LIKE 'UPDATE%';
```

Listing 1.89: SQL Anweisungen

1.16 Extents

1.16.1 Freie Extents zusammen fassen

Hat man einen Tablespace mit `resize` vergrößert, so kann man den neuen freien Extent den vorhandenen zuschlagen. Das ganze funktioniert auch, wenn man aus dem Tablespace Tabellen gelöscht hat. Gibt es in dem Tablespace z.B. ein zweites Datafile, so kann man die freien Extents des zweiten Files nicht dem ersten File zuordnen.

```
sql>ALTER tablespace <TablespaceName> COALESCE;
```

Listing 1.90: Freie Extents

Beispiel:

Der Tablespace ist 30 Blocks groß und in einem Tablespace sind 4 Tabellen mit je 5 Blöcken erstellt worden. Es bleiben also noch 10 Blöcke übrig. Nun wird die letzte Tabelle gelöscht und es gibt nun zwei Extents (5 und 10 Blöcke). Möchte man nun eine neue Tabelle mit 15 Blöcken erstellen, so gibt Oracle eine Fehlermeldung aus, dass kein Platz vorhanden sei. Mit dem oberen Befehl wird nun der Extent zusammengefasst und die Tabelle kann nun erstellt werden.

Wird nach dem Erstellen der 4 Tabellen der Tablespace um 5 Blöcke vergrößert, so gibt es auch hier zwei Extents (10 und 5 Blöcke). Auch hier würde die Erstellung der Tabelle fehlschlagen.

1.16.2 Blöcke / Extents reorganisieren

Sind in der Datenbank Tabellen mit vielen Datensätzen und werden einige dieser Datensätze gelöscht, so wird der physikalische Platz dafür nicht freigegeben. Nicht verändert wird die so genannte High Water Mark. Ab der Version 10G gibt es nun die Option `shrink space`.

Die Option `shrink space` kann auf Tabellen, Indizes, Index, Partition, Subpartition, LOB Segmente (ab 10.2) und Materialized Views angewendet werden.

Damit man diese Option auch ausführen kann, muss der Tablespace mit der Option `Segment Management Auto` angelegt worden sein.

```
sql>SELECT tablespace_name, segment_space_management FROM dba_tablespace;
```

TABLESPACE_NAME	SEGMENT
SYSTEM	MANUAL
SYSAUX	AUTO
UNDOTBS1	MANUAL
TEMP	MANUAL
USERS	AUTO
MYDATA	AUTO

Listing 1.91: Show Segment Management

Informationen über die Blöcke und Extents kann man sich mit den nachfolgenden SQL Statements anzeigen lassen.

```
sql>SELECT blocks, extents FROM user_segments
2>WHERE segment_name = '<TableName>';
```

```
sql>ANALYZE table <TableName> COMPUTE statistics;
```

Table analyzed

```
sql>SELECT empty_blocks, avg_space FROM user_tables
2>WHERE table_name = '<TableName>';
```

```
sql>SELECT owner, segment_name, segment_type -- extents in Datafile
2>FROM dba_extents a, dba_data_files b
3>WHERE a.file_id = b.file_id
4>AND b.file_name like '%cae03.dbf';
```

```
OWNER   SEGMENT_NAME SEGMENT_TYPE
-----
CAE_01  P2STREET_IMP TABLE
CAE_01  P2STREET_IMP TABLE
.
```

Listing 1.92: Info Blöcke / Extents

Für das `shrink space` muss als erstes das `row movement` für die Tabelle eingeschaltet werden.

```
sql>ALTER table <TableName> ENABLE row movement;
```

Listing 1.93: Row Movement

Nun können die Blöcke / Extents reorganisiert werden. Zusätzlich zu der `shrink space` Option kann man noch `compact` oder `cascade` angeben. Mit der Option `compact` wird die High Water Mark nicht neu geschrieben und bei `cascade` wohl.

```
sql>ALTER table <TableName> SHRINK space [compact] [cascade];

sql>ALTER index <IndexName> SHRINK space;

sql>ALTER table <TableName> MODIFY partition <PartitionName>
2>SHRINK space cascade;
```

Listing 1.94: Shrink Space

1.17 Zeitzone

Die eingestellte Zeitzone für die Datenbank oder auch Session, kann man mit dem nachfolgenden Befehl abfragen.

```
sql>SELECT dbtimezone from dual;

DBTIME
-----
+02:00

sql>SELECT sessiontimezone from dual;

SESSIONTIMEZONE
-----
+01:00
```

Listing 1.95: Azeige Zeitzone

Auch welche Zeitzone es in Oracle gibt, kann man abfragen.

```
sql>SELECT * from v$timezone_names;
TZNAME                                TZABBREV
-----
America/Rainy_River                  CST
America/Rainy_River                  CDT
America/Rainy_River                  CWT
America/Rainy_River                  CPT
America/Rankin_Inlet                  zzz
America/Rankin_Inlet                  CST
America/Rankin_Inlet                  CDDT
```

Listing 1.96: Liste Zeitzenen

Die zeitzone kann man auch nachträglich für die Datenbank oder für die Session ändern. Sind schon Tabellen erstellt worden, wo der Datentype für die Spalte Timestamp with local time zone ist, so kann die Zeitzone nicht mehr verändert werden. Mit CET wird die Standard Zeit und mit CEST die Sommerzeit angegeben.

Wird die Zeitzone mit alter database geändert, so greift die Änderung erst nach einem Neustart der Datenbank.

```
sql>ALTER database SET time_zone = '+05:00';

sql>ALTER database SET time_zone = 'Europe/Berlin';

sql>ALTER session SET time_zone = '+02:00';
```

Listing 1.97: Zeitzone ändern

Welcher Timezone Definition File für die Datenbank zuständig ist, kann mit der folgenden Abfrage angezeigt werden.

```
sql>SELECT * from v$timezone_file;

FILENAME      VERSION
-----
Timez1rg.dat      4
```

Listing 1.98: Timezone Definition File

1.18 Netzwerk

1.18.1 Tnsnames.ora & Sqlnet.ora

Anstatt einer lokalen `tnsnames.ora`, kann man auch eine im Netzwerk auswerten. Hierzu ist in der lokalen Datei der Eintrag `ifile=p:\oracle\network\tnsnames.ora` zu setzen. Die Angabe von UNC Pfaden wird auch unterstützt, es müssen aber anstelle von `\\` die `//` genommen werden. In der `tnsnames.ora` können auch mehrere `ifile` Angaben stehen.

```
ifile=//<ServerName>\Network\Oracle\admin\tnsnames_1.ora
ifile=p:\apps\oracle\admin\tnsnames_2.ora
```

Listing 1.99: Tnsnames.ora

In der `sqlnet.ora` befindet sich eine Variable, die die Suchreihenfolge festlegt, welche Datei, Pfad oder Service genommen werden soll, wenn nach einer Oracle SID gesucht wird.

```
NAMES.DIRECTORY_PATH=(TNSNAMES , ONAMES , HOSTNAME)
```

Listing 1.100: Sqlnet.ora

Auf den Clients kann man in der `tnsnames.ora` anstelle des Service Name die SID angeben. Dieses hat den Vorteil, das bei einem Shutdown von dem Client die DB trotzdem erreichbar ist. Bei einem Service Name wird versucht, die DB auszuwerten und diese ist nicht online.

```
uws@tux>grep -i "sid" tnsnames.ora
(SID = cad01)
```

Listing 1.101: SID

1.18.2 Listener Trace

Um das Trace für den Listener einzuschalten, können in der Datei `listener.ora` folgende Parameter eingetragen werden.

Parameter	Wert
TRACE_LEVEL_LISTENER	[OFF USER ADMIN]
TRACE_FILE_LISTENER	Dateiname (listener)
TRACE_DIRECTORY_LISTENER	\$ORACLE_BASE/log/listener
LOG_FILE_LISTENER	Dateiname (listener)
LOG_DIRECTORY_LISTENER	\$ORACLE_BASE/log/listener

Tabelle 1.9: Trace Parameter Listener

Die Dateien bekommen automatisch eine Extension angehängt. Bei dem Trace-File ein `.trc` und bei dem LOG-File ein `.log`. Das Verzeichnis muss vorhanden sein. Nach dem Eintragen der Parameter in der `listener.ora` muss der Listener neu gestartet werden.

```
uws@tux>tail -n 8 listener.ora

# TNS Listener Trace Parameter

TRACE_LEVEL_LISTENER = USER
TRACE_FILE_LISTENER = listener
TRACE_DIRECTORY_LISTENER = $ORACLE_BASE/log/listener
LOG_FILE_LISTENER = listener
LOG_DIRECTORY_LISTENER = $ORACLE_BASE/log/listener

uws@tux>lsnrctl [start|stop|reload]
```

Listing 1.102: Beispiel Trace

1.18.3 Sqlnet Trace

In der Datei sqlnet.ora können für das Trace folgende Parameter eingetragen werden.

Parameter	Wert
TRACE_LEVEL_CLIENT	[OFF USER ADMIN]
TRACE_FILE_CLIENT	Dateiname (client)
TRACE_DIRECTORY_CLIENT	\$ORACLE_BASE/log/sqlnet
TRACE_UNIQUE_CLIENT	[ON OFF]
LOG_FILE_CLIENT	Dateiname (client)
LOG_DIRECTORY_CLIENT	\$ORACLE_BASE/log/sqlnet
TRACE_LEVEL_SERVER	[OFF USER ADMIN]
TRACE_FILE_SERVER	Dateiname (unixsrv_6789.trc)
TRACE_DIRECTORY_SERVER	\$ORACLE_BASE/log/sqlnet
LOG_FILE_SERVER	Dateiname (unixsrv.log)
LOG_DIRECTORY_SERVER	\$ORACLE_BASE/log/sqlnet

Tabelle 1.10: Trace Parameter Listener

Die Extensions sind nicht zwingend anzugeben. Sind sie nicht angegeben, so werden sie automatisch angehängen (siehe Listener Trace). Auch hier muss der Listener neu gestartet werden.

```
uws@tux>tail -n 14 sqlnet.ora

# SQLNET Trace Parameter

TRACE_LEVEL_CLIENT = USER
TRACE_FILE_CLIENT = listener
TRACE_DIRECTORY_CLIENT = $ORACLE_BASE/log/sqlnet
TRACE_UNIQUE_CLIENT = ON
LOG_FILE_CLIENT = client
LOG_DIRECTORY_CLIENT = $ORACLE_BASE/log/sqlnet
TRACE_LEVEL_SERVER = USER
TRACE_FILE_SERVER = unixsrv_5678.trc
TRACE_DIRECTORY_SERVER = $ORACLE_BASE/log/sqlnet
LOG_FILE_SERVER = unixsrv.log
LOG_DIRECTORY_SERVER = $ORACLE_BASE/log/sqlnet

uws@tux>lsnrctl [start|stop|reload]
```

Listing 1.103: Beispiel Trace

Die Auswertung der Log-Datei wird mit `trcasst -odt -e0 -s <FileName>` vorgenommen.

1.19 Oracle User

1.19.1 Liste aller verbundenen User

Alle angemeldeten User an einer Oracle Instance kann mit dem nachfolgenden Statement angezeigt werden.

```
sql>SELECT sid, serial#, username, status, osuser, machine, program
2>FROM v$session ORDER BY sid;
```

Listing 1.104: Liste Connect User

1.19.2 Verbundene User löschen

Mit dem nachfolgenden Befehl kann man einen verbundenen Oracle User von der Instanz löschen.

```
sql>SELECT 'ALTER system KILL session ''' || sid || ',' || serial# || ''';'
2>FROM v$session WHERE username = 'UserName>';
```

Listing 1.105: Verbundenen User löschen

1.19.3 Listet alle Oracle User aus

Alle User, die in Oracle angelegt worden sind, stehen in den Tabellen / Views dba_users und all_users drin.

```
sql>SELECT username FROM dba_users ORDER BY username;
```

```
sql>SELECT * from all_users;
```

```
USERNAME USER_ID CREATED
-----
uws          60 17.01.12
sys           0 20.10.11
```

Listing 1.106: Liste Oracle User

1.19.4 Angemeldeten User anzeigen

Den Namen des angemeldeten User kann man mit den folgenden Abfragen sich anzeigen lassen.

```
sql>SHOW user
USER is "uws"
```

```
sql>SELECT sys_context('userenv','OS_USER') AS user FROM dual;
```

```
USER
-----
UWS
```

Listing 1.107: Angemeldeten User anzeigen

1.19.5 Liste aller User Tables

```
sql>SELECT * FROM user_tables;
```

```
sql>SELECT owner, table_name, tablespace_name FROM all_tables;
```

```
OWNER TABLE_NAME TABLESPACE_NAME
-----
uws    t_city          contact
uws    t_company       contact
```

Listing 1.108: User Tables

1.19.6 Anzeige der SYS Berechtigungen

```
sql>SELECT * FROM v$pwfile_users;
```

USERNAME	SYSDB	SYSOP	SYSAS
SYS	TRUE	TRUE	FALSE

Listing 1.109: SYS Berechtigungen

1.20 DB Informationen

1.20.1 Bootstrap-Segment

Die im Data Dictionary enthaltenen Informationen über das Bootstrap-Segment Abfragen.

```
sql>SELECT * FROM dba_segments WHERE segment_type = 'CACHE';
```

Listing 1.110: Bootstrap-Segment

1.20.2 Anzeige aller aktuellen Werte

```
sql>SHOW parameters;
```

Listing 1.111: Anzeige aller Werte

1.20.3 Hidden Parameter

In Oracle gibt es versteckte Parameter. Diese Parameter fangen mit einem Underscore an und können nicht mit dem show parameter Befehl angezeigt werden. Die versteckten Parameter werden in den beiden Tabellen / Views x\$ksppi und x\$ksppsv verwaltet.

```
sql>CONNECT sys/xxx as sysdba

sql>COL ksppdesc FOR a100
sql>COL ksppinm for a50
sql>SELECT ksppinm, ksppdesc FROM x$ksppi WHERE substr(ksppinm,1,1) = '_';

sql>COL ksppstvl FOR a30
sql>SELECT ksppinm, ksppdesc
 2>FROM x$ksppi a, x$ksppsv b
 3>WHERE a.indx = b.indx and substr(ksppinm,1,1) = '_';

sql>SET linesize 200
sql>COL Param FOR a45 wrap head 'Underscore Parameter'
sql>COL Descr FOR a100 wrap head 'Description'
sql>COL SessionVal FOR a15 head 'Value|Session'
sql>COL InstanceVal FOR a15 head 'Value|Instnc'
sql>SELECT
 2>a.ksppinm Param,
 3>b.ksppstvl SessionVal,
 4>c.ksppstvl InstanceVal,
 5>a.ksppdesc Desc
 6>FROM
 7>x$ksppi a,
 8>x$ksppcv b,
 9>x$psppsv c
10>WHERE
11>a.indx = b.indx AND
12>a.indx = c.indx AND
13>a.ksppinm LIKE '/_%' escape '/'
14>ORDER BY 1;
```

Listing 1.112: Hidden Parameter

Wurde ein Versteckter Parameter geändert, so wird er dann in der Tabelle / View v\$parameter eingetragen.

```
sql>SELECT * FROM v$parameter WHERE substr(name,0,1) = '_';
```

Listing 1.113: Hidden Parameter Changed

1.20.4 Einstellungen der Datenbank

```
sql>SELECT comp_name, version, status, startup
2>FROM dba_registry ORDER BY comp_name;
```

COMP_NAME	VERSION	STATUS	STARTUP
JSERVER JAVA Virtual Machine	11.2.0.1.0	VALID	
OLAP Analytic Workspace	11.2.0.1.0	VAILD	
OLAP Catalog	11.2.0.1.0	VALID	
OWB	11.2.0.1.0	VALID	
Oracle Application Express	3.2.1.00.10	VALID	
.			
.			

Listing 1.114: Einstellungen DB

1.20.5 Datenbank Service Name

```
sql>SHOW parameter service_name
```

NAME	TYPE	VALUE
service_names	string	cad01

```
sql>SHOW parameter db_name
```

NAME	TYPE	VALUE
db_name	string	cad01

```
sql>SHOW parameter db_domain
```

NAME	TYPE	VALUE
db_domain	string	cad01

Listing 1.115: Datenbank Service Name

1.20.6 NLS Parameter

```
sql>SELECT * FROM nls_database_parameters ORDER BY parameter;
```

PARAMETER	VALUE
NLS_CALENDAR	GREGORIAN
NLS_CHARACTERSET	AL32UTF8
NLS_COMP	BINARY
NLS_CURRENCY	\$
NLS_DATE_FORMAT	DD-MON-RR
NLS_DATE_LANGUAGE	AMERICAN
NLS_DUAL_CURRENCY	\$
NLS_ISO_CURRENCY	AMERICA
NLS_LANGUAGE	AMERICAN
NLS_LENGTH_SEMANTICS	BYTE
NLS_NCHAR_CHARACTERSET	AL16UTF16
NLS_NCHAR_CONV_EXCP	FALSE
NLS_NUMERIC_CHARACTERS	.,
NLS_RDBMS_VERSION	11.2.0.2.0
NLS_SORT	BINARY
NLS_TERRITORY	AMERICA
NLS_TIME_FORMAT	HH.MI.SSXF AM

```
NLS_TIMESTAMP_FORMAT      DD-MON-RR HH.MI.SSXFF AM
NLS_TIMESTAMP_TZ_FORMAT   DD-MON-RR HH.MI.SSXFF AM TZR
NLS_TIME_TZ_FORMAT        HH.MI.SSXFF AM TZR
```

Listing 1.116: NLS Parameter

Parameter	Wert	Beschreibung
NLS_CHARACTERSET	AL32UTF16	Used to store data in DB
NLS_NCHAR_CHARACTERSET	AL16UTF16	Used to store data in columns as nchar, nvarchar2 or nvarchar2

Tabelle 1.11: Characterset

1.21 Oracle Audit

1.21.1 Audit Status abfragen

Wenn das Auditing für die Datenbank eingeschaltet ist, so steht als Wert db bei dem Parameter `audit_trail`. Mit den folgenden Befehlen kann man diese Informationen abfragen.

```
sql>SHOW parameter audit

NAME                                TYPE        VALUE
-----
audit_file_dest                      string      /u01/app/oracle/admin/cad01/adump
audit_sys_operations                 Boolean     FALSE
audit_trail                          string      NONE

sql>COL name for a30
sql>COL value for a50
sql>SELECT name, value FROM v$parameter
 2>WHERE name like 'audit';

NAME                                VALUE
-----
audit_sys_operations                 FALSE
audit_file_dest                      /u01/app/oracle/admin/cad01/adump
audit_trail                          NONE
```

Listing 1.117: Audit Status

Der Parameter `audit_file_dest` wird benötigt, wenn man die Option `os`, `xml` oder `xml,extended` verwendet. Hier wird der Pfad angegeben, wo die Files abgelegt werden sollen.

Wir der Parameter von `audit_sys_operations` auf `TRUE` gestellt, so werden alle User connections mit `sysdba` oder `sysoper` Privilegien protokolliert. Diese Informationen werden in dem Pfad abgelegt, der in dem Parameter `audit_file_dest` eingetragen worden ist.

1.21.2 Audit einschalten

Ist kein Auditing eingeschaltet, so kann das mit dem Befehl `alter system` und anschließenden Neustart der Datenbank das Auditing eingeschaltet werden.

```
sql>ALTER system SET audit_train = '<option>' scope = spfile;
```

Listing 1.118: Audit einschalten

Für das einschalten des Auditings gibt es folgende Optionen:

Option	Beschreibung
NONE oder FALSE	Auditing abgeschaltet
DB oder TRUE	Auditing ist eingeschaltet und die Werte werden in die Datenbank geschrieben
DB,EXTENDED (<10gR1)	Wie DB, aber SQL_BIND und SQL_TEXT werden auch gefüllt
DB_EXTENDED (=>10gR1)	
XML	Speichert die Informationen in einem XML File
XML,EXTENDED (=>10gR2)	Wie XML, aber SQL_BIND und SQL_TEXT werden gefüllt
OS	Alle Informationen werden direkt in das Operating System Auditing geschrieben.

Tabelle 1.12: Characterset

1.21.3 Audit Actions abfragen

Wurde oder ist das Auditing eingeschaltet, so kann man nachschauen, ob schon Aktivitäten protokolliert werden soll.

```

sql>SELECT * FROM dba_stmt_audit_opts UNION
2>SELECT * FROM dba_priv_audit_opts;
USER_NAME PROXY_NAME AUDIT_OPTION SUCCESS FAILURE
-----
ALTER ANY PROCEDURE BY ACCESS BY ACCESS
ALTER ANY TABLE BY ACCESS BY ACCESS
ALTER DATABASE BY ACCESS BY ACCESS
ALTER PROFILE BY ACCESS BY ACCESS
ALTER SYSTEM BY ACCESS BY ACCESS
ALTER USER BY ACCESS BY ACCESS
AUDIT SYSTEM BY ACCESS BY ACCESS
CREATE ANY JOB BY ACCESS BY ACCESS
CREATE ANY LIBRARY BY ACCESS BY ACCESS
CREATE ANY PROCEDURE BY ACCESS BY ACCESS
CREATE ANY TABLE BY ACCESS BY ACCESS
CREATE EXTERNAL JOB BY ACCESS BY ACCESS
CREATE PUBLIC DATABASE LINK BY ACCESS BY ACCESS
CREATE SESSION BY ACCESS BY ACCESS
CREATE USER BY ACCESS BY ACCESS
DATABASE LINK BY ACCESS BY ACCESS
DROP ANY PROCEDURE BY ACCESS BY ACCESS
DROP ANY TABLE BY ACCESS BY ACCESS
DROP PROFILE BY ACCESS BY ACCESS
DROP USER BY ACCESS BY ACCESS
EXEMPT ACCESS POLICY BY ACCESS BY ACCESS
GRANT ANY OBJECT PRIVILEGE BY ACCESS BY ACCESS
GRANT ANY PRIVILEGE BY ACCESS BY ACCESS
GRANT ANY ROLE BY ACCESS BY ACCESS
PROFILE BY ACCESS BY ACCESS
PUBLIC SYNONYM BY ACCESS BY ACCESS
ROLE BY ACCESS BY ACCESS
SYSTEM AUDIT BY ACCESS BY ACCESS
SYSTEM GRANT BY ACCESS BY ACCESS

```

Listing 1.119: Audit Actions

1.21.4 Audit Regel erstellen

Syntax der Audit Regel Erstellung.

audit {statement_option|privilege_option} [by {session|access}] [whenever {successfull|unsuccessfull}]
 Nachfolgend sind einige Beispiele aufgelistet, die das Auditing einschalten.

```

sql>AUDIT role;

sql>AUDIT role WHENEVER successfull;

sql>AUDIT role WHENEVER not successfull;

sql>AUDIT select table, update table;

sql>AUDIT delete any table;

sql>AUDIT create any directory;

sql>AUDIT directory;

sql>AUDIT read on directory bfile_dir;

sql>AUDIT select on hr.employees;

sql>AUDIT select on hr.employees WHENEVER successfull;

sql>AUDIT select on ht.employees WHENEVER not successfull;

sql>AUDIT insert, update on oe.customers;

sql>AUDIT all on hr.employees_seq;

sql>AUDIT alter, grant, insert, update, delete on default;

```

Listing 1.120: Audit Examples

SQL Befehle für Auditing.

SQL Befehl kurz	SQL Befehl und Überwachungs Tätigkeiten
Alter System	alter system
Cluster	create cluster alter cluster drop cluster truncate cluster
Context	create context drop context
Database Link	create database link drop database link
Dimension	create dimension alter dimension drop dimension
Directory	create directory grant privilege on directory drop directory rvoke privilege on directory
Execute Procedure	call
Index	create index alter index analyze index drop index
Materialized View	create materialized view alter materialized view drop materialized view
Not Exist	Alle SQL Statements, wo als Rückgabewert Object does not exist

continued on next page.

continued from previous page.

SQL Befehl kurz	SQL Befehl und Überwachungs Tätigkeiten
Outline	create outline alter outline drop outline
Procedure	create function create library create package create package body create procedure drop function drop library drop package drop procedure grant privilege on procedure, function, package revoke privilege on procedure, function, package
Profile	create profile alter profile drop profile
Public Database Link	create public database link drop public database link
Public Synonym	create public Synonym drop public synonym
Role	create role alter role drop role set role
Rollback Segment	create rollback segment alter rollback segment drop rollback segment
Sequence	any statement containing sequence.currval or sequence.nextval create sequence alter sequence drop sequence grant privilege on sequence revoke privilege on sequence
Sessions	Logons
Synonym	create synonym drop synonym
System Audit	audit sql_statements noaudit sql_statements
System Grant	grant system_privileges_and_roles revoke system_privileges_and_roles
Table	create table alter table comment on table table, view, materialized view comment on column table.column, view.column, materialized.column delete from table, view insert into table, view drop table lock table table, view grant privilege on table, view, materialized view

continued on next page.

continued from previous page.

SQL Befehl kurz	SQL Befehl und Überwachungs Tätigkeiten
	revoke privilege on table, view, materialized view
	select from table, view, materialized view
	truncate table
Tablespace	update table, view
	create tablespace
	alter tablespace
	drop tablespace
Trigger	create trigger
	alter trigger, mit enable und disable clauses
	drop trigger
	alter table, mit enable all triggers clause und disable all triggers clause
Type	create type
	create type body
	alter type
	drop type
	drop type body
	grant privilege on type
	revoke privilege on type
User	create user
	alter user
	drop user
View	create view
	drop view

Tabelle 1.13: Audit SQL Befehle

Schema Objecte für Auditing.

SQL Befehl kurz	SQL Befehl und Überwachungs Tätigkeiten
Table	alter
	audit
	comment
	delete
	flashback (nur flashback queries)
	grant
	index
	insert
	lock
	rename
	select
	update
View	audit
	comment
	delete
	flashback (nur flashback queries)
	grant
	insert
	lock
	rename
	select
	update

continued on next page.

continued from previous page.

SQL Befehl kurz	SQL Befehl und Überwachungs Tätigkeiten
Sequence	alter audit grant select
Procedure, Function, Package	audit execute grant
Materialized View	alter audit comment delete index insert lock select update
Mining Model	audit comment grant rename select
Directory	audit grant read
Library	execute grant
Object Type	alter audit grant
Comtext	audit grant

Tabelle 1.14: Audit SQL Objecte

1.21.5 Audit Regel löschen

Die Syntax um eine Audit Regel zu löschen, ist die gleiche wie eine Regel zu erstellen. Das Löschen einer Regel wird mit dem Befehl `noaudit` durchgeführt. Nachfolgend sind einige Beispiele aufgeführt.

```
sql>NOAUDIT select table by hr;
sql>NOAUDIT delete any table;
sql>NOAUDIT select on hr.employees;
sql>NOAUDIT select on hr.employees WHENEVER successful;
```

Listing 1.121: Audir Regel löschen

1.22 Datenbank Check

In der Oracle Version 10G gab es im Rman zwei Validate Befehle. Bei backup validate wurde die Datenbank auf logische und physische Korruption hin überprüft. Als zweiten Befehl gab es validate backupset, der die Backupsets der Sicherung überprüft. In der Version 11G wurde der Validate Befehlsatz nochmal erweitert. Nun können einzelne Datenbankdateien und auch Blöcke untersucht werden.

Um eine Datenbank überprüfen zu können, muss sich die Datenbank im Archive Modus befinden.

1.22.1 Datenbank Überprüfen

Um eine Datenbank zu überprüfen, muss man sich mit rman an die Datenbank anmelden. Mit list failure kann man sich den Fehler anzeigen lassen und mit der Angabe der Fehlernummer und der Option detail die exakte Beschreibung des Fehlers. Mit advise failure kann man sich Lösungsvorschläge anzeigen lassen.

```
uws@tux>rman target /
rman>list failure;

using target database control file instead of recovery catalog
List of Database Failures
=====

Failure ID Priority Status Time Detected Summary
-----
180          HIGH    OPEN   22-APR-10   One or more non-system datafiles
are missing

rman>list failure 180 detail;
```

Listing 1.122: List Failure

Eine Überprüfung der Datenbank wird mit dem Befehl validate database eingeleitet.

```
uws@tux>rman target /
rman>validate database;

Starting validate at 22-APR-10
Using target database control file instead of recovery catalog
.
.
```

Listing 1.123: Validate Database

Mit den nachfolgenden Beispielen kann man das Überprüfen der Datenbank feiner vornehmen. In der Oracle Version 10G gab es nur den backup validate Befehl. Sollen auch die Archive Logs überprüft werden, so gibt man die Option archivelog all mit an.

```
rman>validate datafile 1;

rman>validate datafile 1 block 815,977 datafile 3 block 210 to 305;

rman>validate datafile '/u02/oracle/oradata/cad01/users01.dbf';

rman>validate tablespace users;

rman>backup validate check logical database;

rman>backup validate check logical database archivelog all;
```

Listing 1.124: Validate Beispiele

1.22.2 Datenbank reparieren

Nachdem man festgestellt hat, dass die Datenbank defekt ist, kann man sich mit dem Befehl `advice failure` Vorschläge anzeigen lassen, wie man die Fehler beheben kann. Hierbei gibt es eines der Optionen `all` oder `critical` an.

```
uws@tux>rman target /
rman>advice failure [all] [critical];
```

Listing 1.125: Advice failure

Mit `repair failure preview` kann man sich eine Vorschau anzeigen lassen, wie das System diesen Fehler beheben möchte. Ohne die Option `preview` wird die Datenbank dann repariert.

```
uws@tux>rman target /
rman>repair failure preview;

Strategy: The repair includes complete media recovery with no data loss
Repair Script: /home/oracle/cad01/reco_468293748.hm

Contents of repair script:
# restore and recover datafile
sql 'alter database datafile 4 offline';
restore datafile 4;
recover datafile 4;
sql 'alter database datafile 4 online;

rman>repair failure [noprompt];

Strategy: The repair includes complete media recovery with no data loss
Repair Script: /home/oracle/cad01/reco_468293748.hm

Contents of repair script:
# restore and recover datafile
sql 'alter database datafile 4 offline';
restore datafile 4;
recover datafile 4;
sql 'alter database datafile 4 online;
Do you really want to execute the above repair (enter YES or NO)? YES

Executing repair script

sql statement: alter database datafile 4 offline

Starting restore at 22-Apr-10
Using channel ORA_DISK1
.
.
.
sql statement: alter database datafile 4 online;
repair failure complete
```

Listing 1.126: Repair failure

1.23 Corrupte Blöcke

1.23.1 Anzeigen

Gibt es in der Datenbank Corrupte Blöcke, so werden sie in dem View v\$database_block_corruption angezeigt.

```
sql>SELECT * FROM v$database_block_corruption;

FILE# BLOCK# BLOCKS CORRUPTION_CHANGE# CORRUPTION
-----
1 5867 9 0 corrupt
```

Listing 1.127: List Corrupt Blocks

Die Zuordnung der Block Nr. zu dem Segment Name und Segment Type kann man sich mit der folgenden Abfrage anzeigen lassen.

```
sql>COL owner FOR a10
sql>COL segment_name FOR a20
sql>COL segment_type for a15
sql>SELECT owner, segment_name, segment_type FROM dba_extents
2>WHERE file_id = <nr> and <block_nr>
3>BETWEEN block_id and block_id + blocks -1;

OWNER          SEGMENT_NAME          SEGMENT_TYPE
-----
HR              SAL                    Table
```

Listing 1.128: List Segment Info

Eine Zuordnung von einer Tabelle zu einem LOG-Segment kann man sich folgendermaßen anzeigen lassen.

```
sql>SELECT owner, table_name, column_name from dba_lobs
2>WHERE segment_name = 'SYS<identifier>$$';

OWNER TABLE_NAME COLUMN_NAME
-----
XDB XDB$RESOURCE "XMLDATA"."XMLLOB"
```

Listing 1.129: List LOB-Segment

1.23.2 Repair Tabellen erstellen

Um die Datenbank händisch zu reparieren, gibt es das DBMS_REPAIR Package. Folgende Funktionen gibt es in dem Package.

<i>Funktion</i>	<i>Beschreibung</i>
admin_tables	Benötigt Admin Funktion (create, rop, purge) für das reparieren
check_object	Aufspüren und Anzeigen der defekten Blöcke in Tables / Views
dump_orphan_keys	Anzeigen der Index Einträge für den dazugehörenden corrupten Block
fix_corrupt_block	Die defekten Blöcke werden markiert
rebuild_freelist	Neuaufbau der Free List für die Objecte
segment_fix_Status	Liefert die Fähigkeit den corrupten Block zu reparieren, wenn der segment space management auf AUTO steht
skip_corrupt_block	Überspringen von corrupten Blöcken bei einem check object. Wird die Option nicht gesetzt, gibt es eine ORA-1578 Fehlermeldung

Tabelle 1.15: Funktionen DBMS_REPAIR

Als erstes werden zwei Tabellen erstellt. Die Tabellen Namen müssen groß geschrieben werden, sonst gibt es eine Fehlermeldung und sie werden nicht erstellt.

```

sql>BEGIN
 2>dbms_repair.admin_tables
 3>(
 4> table_name => 'REPAIR_TABLE',
 5> table_type => dbms_repair.repair_table,
 6> action      => dbms_repair.create_action,
 7> tablespace => 'USERS'
 8>);
 9>END;
10>/

sql>BEGIN
 2>dbms_repair.admin_tables
 3>(
 4> table_name => 'ORPHAN_KEY_TABLE',
 5> table_type => dbms_repair.orphan_table,
 6> action      => dbms_repair.create_action,
 7> tablespace => 'USERS'
 8>);
 9>END;
10>/

```

Listing 1.130: Create Admin Tables

1.23.3 Tabellen Corruption finden

Corrupte Blöcke kann man nachfolgend suchen und anschließend kann man sich den gefundenen Block anzeigen lassen.

```

sql>SET serveroutput on
sql>DECLARE num_corrupt int;
 2>BEGIN
 3>num_corrupt := 0;
 4>dbms_repair.check_object
 5>(
 6> schema_name => 'HR',
 7> object_name => 'SAL',
 8> repair_table_name => 'REPAIR_TABLE',
 9> corrupt_count => num_corrupt);
10>dbms_output.put_line('Number Corrupt: ' || to_char(num_corrupt));
11>END;
12>/

sql>Number Corrupt: 1

sql>SELECT object_name, block_id, corrupt_type, marked_corrupt,
 2>corrupt_description, repair_description FROM repair_table;

```

Listing 1.131: Table Corruption

Folgende Optionen gibt es für die Funktion `check_object`.

<i>Option</i>	<i>In/Out</i>	<i>Standard Wert</i>
schema_name	In	
object_name	In	
partition_name	In	
object_type	In	table_object
repair_table_name	In	repair_table
flags	In	
block_start	In	
block_end	In	
corrupt_count	Out	

Tabelle 1.16: Optionen check_object

1.23.4 Corruption Blocks beheben

Nachdem mit check_object corrupte Blöcke gefunden worden sind, kann man mit den Informationen aus der Repair Tabelle die Blöcke reparieren.

```

sql>SET serveroutput on;
sql>DECLARE num_fix int;
2>BEGIN
3>num_fix := 0;
4>dbms_repair.fix_corrupt_blocks
5>(
6> schema_name => 'HR',
7> object_name => 'SAL',
8> object_type => dbms_repair.table_object,
9> repair_table_name => 'REPAIR_TABLE',
10> fix_count => num_fix
11>);
12>dbms_output.put_line('Number Fix: ' || to_char(num_fix));
13>END;
14/

sql>Number Fix: 1

sql>SELECT object_name, block_id, marked_corrupt FROM repair_tale;

OBJECT_NAME  BLOCK_ID  MARKED_CORRUPT
-----
SAL          5187     TRUE

```

Listing 1.132: Corruption Block repair

Bei der Ausgabe von TRUE ist der Block als Corrupt markiert.

1.23.5 Index Einträge zu Corrupt Data Blocks finden

Mit der Option dump_orphan_keys kann man die Index Einträge zu den Corrupt Data Blocks finden. Corrupte Index Blöcke können mit einem rebuild repariert werden.

```

sql>SET serveroutput on;
sql>DECLARE num_orphans int;
2>BEGIN
3>num_orphans := 0;
4>dbms_repair.dump_orphan_keys
5>(
6> schema_name => 'HR',
7> object_name => 'SAL_S',
8> object_type => dbms_repair.index_object,

```

```

9> repair_table_name => 'REPAIR_TABLE',
10> orphan_table_name => 'ORPHAN_KEY_TABLE',
11> key_count => num_orphans
12>);
13>dbms_output.put_line('Orphan Key count: ' || to_char(num_orphans));
14>END;
15/

sql>Orphan Key count: 5

```

Listing 1.133: Find Index

1.23.6 Überspringen der Corrupten Blöcke

Mit der Option `skip_corrupt_blocks` werden die Blöcke markiert, die bei einem Index und Tabellen Scan übersprungen werden sollen.

```

sql>BEGIN
2>dbms_repair.skip_corrupt_blocks
3>(
4> schema_name => 'HR',
5> object_name => 'SAL',
6> object_type => dbms_repair.table_object,
7> flags=> dbms_repair.skip_flag
8>);
9>END;
10/

sql>SELECT owner, table_name, skip_corrupt FROM dba_tables
2>WHERE owner = 'HR';

```

Listing 1.134: Skip Corrupt Blocks

1.23.7 Object Typen

Mit den folgenden Abfragen kann man sich die Object Typen anzeigen lassen.

```

sql>SELECT object_type, object_name FROM user_objects
2>ORDER BY object_type, object_name;

sql>SELECT object_type, object_name FROM dba_users
2>WHERE username = 'SYS' ORDER BY object_type, object_name;

```

Listing 1.135: Object Typen

1.23.8 Reparatur von Hand

Als erstes wird die Tabelle mit dem `move` Befehl verschoben.

```
sql>ALTER table <TableName> MOVE;
```

Listing 1.136: Move Table

Anschließend schauen wir mal nach, ob auch Indexe auf der Tabelle vorhanden sind, die dann auch verschoben werden.

```

sql>SELECT 'alter index ' ||owner|| '.' ||index_name|| ' rebuild onlie;'
2>FROM dba_indexes WHERE table_name = '<TableName>';

sql>ALTER index <IndexName> MOVE;

```

Listing 1.137: Move Index

Zu guter letzt wird die Datenbank Datei auf Fehler überprüft.

```
uws@tux>dbv file=/u01/oracle/oradata/cad01/system01.dbf blocksize=8192
```

Listing 1.138: Check Datei

1.24 Jobs und Task

1.24.1 Jobs auflisten

Ausser den Scheduler Jobs, gibt es noch die Views `dba_jobs`, `all_jobs`, `user_jobs` und `dba_jobs_running`.

```
sql>SELECT job, next_date, next_sec, failures, broken FROM dba_jobs;

JOB NEXT_DATE NEXT_SEC FAILURES B
-----
27 01-JUN-10 16:00:00          0 N
```

Listing 1.139: List Jobs

Eine Auflistung, welche Jobs gerade ausgeführt werden.

```
sql>SELECT sid, r.job, log_user, r.this_date, r.this_sec
2>FROM dba_jobs_running r, dba_jobs j WHERE r.job = j.job;

SID JOB LOG_USER THIS_DATE THIS_SEC
-----
12 27 SYS          01-JUN-10 16:01:00

sql>SELECT owner, program_name, enabled FROM dba_scheduler_programs;

OWNER PROGRAM_NAME          ENABLED
-----
SYS    JDM_XFORM_SEQ_PROGRAM TRUE
SYS    JDM_PROFILE PROGRAM    TRUE
```

Listing 1.140: Running Jobs

Mit der nächsten Abfrage kann man sich das Script für den Job anzeigen lassen.

```
sql>SELECT owner, job_name, job_action FROM dba_scheduler_jobs;

OWNER JOB_NAME JOB_ACTION
-----
SYS    DEFRAG      begin
        execute immediate 'alter table
        "<ts_name>". "<TableName>" enable row movement; '
        execute immediate 'alter table
        "<ts_name>". "<TableName>" shrink space; '
        end;
```

Listing 1.141: Job Script

Diese Abfrage listet alle Scheduled Jobs auf.

```
sql>SELECT
2> owner,
3> job_name,
4> to_char(start_date, 'DD.MM.YYYY HH24:MI:SS'),
5> to_char(next_run_date, 'DD.MM.YYYY HH24:MI:SS'),
6> state,
7> enabled
8> FROM
9> dba_scheduler_jobs
10> WHERE
11> job_name = 'COUNT_USER_PROJECT';
```

Listing 1.142: Liste Scheduled Jobs

Anzeige der eigenen Jobs.

```

sql>COL interval FOR a30
sql>SELECT job, next_date, next_sec, interval FROM user_jobs;

JOB NEXT_DATE          NEXT_SC INTERVAL
-----
1 18.03.2013 14:00:00 14:00:00 trunc(sysdate)+30/1440

```

Listing 1.143: Eigene Jobs

Detail Anzeige eines Jobs.

```

sql>COL log_date FORMAT a35
sql>COL job_name FORMAT a30
sql>COL additional_info FORMAT a40
sql>COL status FORMAT a10

sql>SELECT
  2> log_date,
  3> owner,
  4> job_name,
  5> status,
  6> error#,
  7> additional_info
  8>FROM
  9> dba_scheduler_job_run_details
 10>WHERE
 11> job_name = 'My_JOB';

```

Listing 1.144: Detail Job

1.24.2 Job erstellen

Mit DBMS_SCHEDULER.CREATE_JOB wird ein neuer Job erstellt.

```

prompt Create Cron Job
BEGIN
  DBMS_SCHEDULER.CREATE_JOB (
    job_name      => 'count_user_project',
    job_type      => 'STORED_PROCEDURE',
    job_action    => 'PROJUSER.COUNT_PROJECT_USER',
    number_of_arguments => 0,
    start_date    => '07.09.2018 11:40:00 Europe/Berlin', /* NULL */
    repeat_interval => 'FREQ=MINUTELY;INTERVAL=5', /* every 5 minutes */
    end_date      => NULL,
    auto_drop     => FALSE,
    enabled       => TRUE,
    comments      => 'Count User for each project'
  );
END;
/

```

Listing 1.145: Job erstellen

Nachfolgend einige Beispiele für das repeat_interval.

<i>repeat_interval</i>	<i>Beschreibung</i>
freq=secondly	Run every second
freq=minutely; interval=5	Run every 5 minutes
freq=hourly	Run every hour
freq=daily; byhour=3	Run at 3 am every day
freq=daily; byhour=8,20	run at 8 am and 8 pm every day
freq=daily; interval=10	Run every 10 days
freq=weekly; byday=fri	Run every friday
freq=weekly; byday=tue; byhour=6; byminute=23	Run every Tuesday at 06:23
freq=monthly; bymonthday=1	Run on the first day of every month
freq=monthly; bymonthday=-1	Run on the last day of every month
freq=yearly; bymonth=sep; bymonthday=20	Run yearly on September 20th

Tabelle 1.17: Beispiele repeat_interval

1.24.3 Jobs starten

Jobs werden mit dem Befehl `dbms_job.run` gestartet.

```
sql>BEGIN
2>dbms_job.run(27);
3>END;
4>/
```

Listing 1.146: Job starten

Scheduler Jobs werden folgendermaßen gestartet.

```
sql>BEGIN
2> DBMS_SCHEDULER.RUN_JOB (
3>  JOB_NAME => 'J_SET_EXPORT_DATE_NAVIS',
4>  USE_CURRENT_SESSION => FALSE);
5>END;
6>/
```

Listing 1.147: Scheduler Job starten

1.24.4 Job modify

Einen Scheduler Job kann man folgendermaßen modifizieren.

```
sql>BEGIN
2> DBMS_SCHEDULER.SET_ATTRIBUTE (
3>  NAME          => 'MY_JOB',
4>  ATTRIBUTE     => 'start_date',
5>  VALUE         => '27.09.2018 10:00:00 Europe/Berlin');
6>END;
7>/
```

Listing 1.148: Scheduler Job modify

1.24.5 Jobs disable

Mit `dbms_scheduler.disable` kann man Jobs disablen. Die Angabe bei Name ist der Job Name und mit einem Komma getrennt, kann man auch mehrere Jobs disablen. Man kann nur seine eigenen Jobs disablen, nicht die von anderen DB Usern. Möchte man mehrere Jobs gleichzeitig disablen, so werden die Jobs mit einem Komma getrennt angegeben ('job1','job2').

```
sql>BEGIN
2> DBMS_SCHEDULER.DISABLE('My_Job'); -- Optionale Angabe name => 'My_Job'
```

```
3>END;
4>/
```

Listing 1.149: Job disablen

1.24.6 Jobs enable

Einen deaktivierten Job kann man auch wieder aktivieren. Auch hier werden mehrere Jobs mit einem Komma getrennt angegeben.

```
sql>BEGIN
2> DBMS_SCHEDULER.ENABLE('MY_JOB')
3>END;
4>/
```

Listing 1.150: Job enablen

1.24.7 Job Zeiten ändern

Jobs werden mit dem Befehl `dbms_job.change` geändert. Die Syntax hierzu ist (`job, what,next_date, interval`).

```
sql>EXEC dbms_job.change (1, NULL, trunc(sysdate)+15/24, 'trunc(sysdate+1)
+30/1440');

sql>EXEC dbms_job.interval (1, 'sysdate+(1/48)');
```

Listing 1.151: Job Zeiten ändern

<i>Beispiel</i>	<i>Beschreibung</i>
sysdate+1	Tägliche Ausführung
sysdate+7	Einmal pro Woche
sysdate+1/24	Jede Stunde
sysdate+1/48	Alle 30 Minuten
sysdate+10/1440	Alle 10 Minuten
sysdate+1/96	Alle 5 Minuten
sysdate+30/86400	Alle 30 Sekunden

Tabelle 1.18: Beispiele Zeiten

1.24.8 Jobs löschen

Erstellte Jobs werden wieder mit `dbms_jobs.remove` entfernt. Die Angabe nach dem Befehl ist die Job Nummer. Anstelle der Job Nummer kann auch der Job Name angegeben werden.

```
sql>EXEC dbms_job.remove(1);

PL/SQL procedure successfully completed
```

Listing 1.152: Job löschen

1.24.9 Tasks anzeigen

Für das Anzeigen der Tasks gibt es verschiedene Views, die in der unteren Tabelle aufgelistet sind.

```
sql>SELECT client_name, job_name, job_status, job_start_time
2>FROM dba_autotask_job_history
3>ORDER BY job_start_time;
```

Listing 1.153: List Task

<i>View Name</i>	<i>Beschreibung</i>
dba_autotask_client_job	Informationen über laufende Jobs
dba_autotask_client	Statistic Data über Maintenance Tasks
dba_autotask_job_history	Historie über gelaufene Tasks
dba_autotask_window_clients	List the Windows that belong to MAINTENANCE_WINDOW_GROUP
dba_autotask_client_history	Provides per window history of job execution counts for each automated maintenance task.

Tabelle 1.19: Task Views

1.25 Spfile oder Pfile

Mit dem nachfolgenden Befehl kann man abfragen, ob die Datenbank mit dem Spfile oder mit dem Pfile gestartet worden ist. Gibt es keine Einträge in der Spalte Value, so wird der Pfile verwendet.

Mit der zweiten Abfrage wird direkt ausgegeben, mit welchem File die Datenbank gestartet wurde.

```
sql>COL value FOR a50
sql>SELECT name, value FROM v$spparameter ORDER BY name;

sql>SELECT decode(value, NULL, 'PFILE', 'SPFILE') "Start File Type"
2>FROM v$parameter WHERE name = 'spfile';

Start
-----
SPFILE
```

Listing 1.154: Abfrage Spfile

Oracle sucht das Startparameterfile in der folgenden Reihenfolge.

Im Init\${ORACLE_SID}.ora kann ein Eintrag vorhanden sein, wo sich gegebenenfalls der Spfile befindet.

Parameterfile	Linux	Windows
spfile\${ORACLE_SID}.ora	\${ORACLE_HOME}/dbs	%ORACLE_HOME%\database
spfile.ora	\${ORACLE_HOME}/dbs	%ORACLE_HOME%\database
init\${ORACLE_SID}.ora	\${ORACLE_HOME}/dbs	%ORACLE_HOME%\database

Tabelle 1.20: Reihenfolge Parameterfile

```
--- schnipp ---
spfile='d:\Orahome\dba\spfile.ora'
--- schnapp ---
```

```
sql>SELECT isspecified, count(*)
2>FROM v$spparameter
3>GROUP BY isspecified;

ISSPEC COUNT(*)
-----
TRUE      40
FALSE     308
```

Listing 1.155: Abfrage Spfile

Wird ein Spfile verwendet, so bekommt man mit der Abfrage als Ausgabe einen Wert für TRUE. Bei einem Pfile fehlt TRUE, es gibt nur einen Wert für FALSE.

1.26 DB Tuning

1.26.1 SQL optimieren

```
sql>ALTER system SET cursor_sharing = 'FORCE' scope=both;
```

Listing 1.156: SQL optimieren

Als Wert können FORCE oder EXACT angegeben werden. Der Wert FORCE nimmt dann wenn es geht immer das gleiche SQL Statement und ersetzt Werte durch Variablen.

Hierzu gibt es die Views v\$sql, v\$sqlarea, v\$sqltext, v\$sqltext_with_newlines und v\$sql_plan.

1.26.2 Parallele Operationen ausführen

Anzahl der gleichzeitigen Operationen pro CPU. Es sollte hierzu der Parameter `parallel_automatic_tuning` auf `TRUE` gestellt werden. Nach der Änderung ist ein Neustart der Datenbank notwendig.

```
sql>ALTER system SET parallel_automatic_tuning = TRUE scopr=spfile;
sql>ALTER system SET parallel_threads_per_cpu = 4 scope=both;
```

Listing 1.157: Parallele Operationen

1.26.3 Buffer Cache messen

Der ermittelte Wert sollte über 90% liegen.

```
sql>SELECT 1-(a.value / (b.value + c.value)) "HIT Ration"
2>FROM v$sysstat a, v$sysstat b, v$sysstat c
3>WHERE a.name = 'physical reads'
4>AND b.name = 'db block gets'
5>AND c.name = 'consistent gets';
```

```
HIT Ratio
-----
,999432705
```

Listing 1.158: Buffer Cache

1.26.4 Weitere Tuning Parameter

Die nachfolgenden Parameter sollten auf jedenfall gesetzt werden. Als erstes wird im Dispatcher der Oracle Aurora Server definiert.

```
sql>SHOW parameter dispatcher

NAME                TYPE        VALUE
-----
dispatchers         string      (PROTOCOL=TCP) (SERVICE=CAD01XDB)

sql>ALTER system SET dispatchers = '(PROTOCOL=TCP) (SERVICE=CAD01XDB)
2>(PRE=oracle.aurora.server.SGIopServer)' scope=both;
```

Listing 1.159: Dispatcher

Ein versteckter / undokumentierter Parameter wird gesetzt und ein normaler Parameter.

```
sql>ALTER system SET optimizer_secure_view_merging = FALSE scope=both;
sql>ALTER system set "_b_tree_bitmap_plans" = FALSE scope=both;
```

Listing 1.160: Parameter setzen

1.27 Oracle 32/64 Bit

Um herauszufinden, ob Oracle in der 32 Bit oder 64 Bit Version installiert worden ist, kann man folgende select Abfrage absetzen.

```
sql>SELECT distinct address from v$sql where rownum < 2;

ADDRESS
-----
ODFOBEB8
```

Listing 1.161: 32 / 64 Bit

Ist die Ausgabe des select Statement 8 stellig, so handelt es sich um die 32 Bit Oracle Version. Bei einer 16 stelligen Ausgabe ist die 64 Bit Version installiert.

1.28 Status DML

```
sql>COL pid FORMAT a10
sql>COL username FORMAT a10
sql>COL program FORMAT a45

sql>SELECT s.inst_id, s.sid, s.serial#, p.spid, s.username, s.programm
2>FROM gv$session s
3>JOIN gv$session p ON p.addr = s.paddr and p.inst = s.inst_id
4>WHERE s.type != 'BACKGROUND';
```

Listing 1.162: Status DM

1.29 Uptime DB

Möchte man wissen, wann die Datenbank das letzte Mal neu gestartet wurde, so kann man das mit der nachfolgenden Abfrage sich anzeigen lassen.

```
sql>SELECT to_char(startup_time, 'DD-MON-YYYY HH24:MI:SS') "DB Startup Time"  
2>FROM v$instance;
```

```
DB Startup Time
```

```
-----  
19-Sep-2011 13:04:34
```

Listing 1.163: Uptime DB

1.30 Oracle Verbindungs Informationen

Mit der Funktion `sys_context` kann man seine aktuellen Verbindungsinformationen abfragen.

```
sql>SELECT sys_context('userenv','OS_USER') "OS User" FROM dual;

OS User
-----
Oracle
```

Listing 1.164: Verbindungs Informationen

Die nachfolgende Tabelle listet alle Parameter auf, die man mit dem Namespace `userenv` abfragen kann. Bis auf dem Parameter `GLOBAL_UID`, sind alle Parameter auch in 10g abzurufen. Die Tabelle wurde von <http://www.techonthenet.com> übernommen und teilweise ins deutsche übersetzt.

Parameter	Beschreibung
ACTION	Returns the position in the Module
AUDITED_CURSORID	Returns the Cursor ID of the SQL that trigger the audit
AUTHENTICATED_IDENTITY	Returns the identity used in authentication
AUTHENTICATION_DATA	Authentication Data
AUTHENTICATION_METHOD	Returns the method of authentication
BG_JOB_ID	Ist die Session ein Hintergrundprozess, so ist die Rückgabe Wert die Job ID, andernfalls wird eine Null zurückgegeben
CLIENT_IDENTIFIER	Returns the Client identifier
CLIENT_INFO	User Session Informationen
CURRENT_BIND	Bind Variables for fine-grained auditing
CURRENT_SCHEMA	Anzeige des default schemas
CURRENT_SCHEMAID	Anzeige der default schema ID
CURRENT_SQL	SQL Anzeige, das den audit event trigger
CURRENT_SQL_LENGTH	Returns the length of the current SQL statment thar triggered the audit event
DB_DOMAIN	Anzeige der Domain von der DB
DB_Namespace	Anzeige des Datenbank Namen
DB_UNIQUE_NAME	Anzeige des Datenbank Unique Namen
ENTRYID	Available auditing entry identifier
ENTERPRISE_IDENTITY	Returns the user's enterprise-wide identity
FG_JOB_ID	Ist die Session ein Foreground Prozess, so ist der Rückgabe Wert die Job ID, andernfalls wird eine NULL zurückgegeben
GLOBAL_CONTEXT_MEMORY	The number used in the SGA by the globally accessed context
GLOBAL_ID	The global user ID from Oracle internet Directory for enterprise security logins. Returns NULL for all other logins
HOST	Name der verbundenen Host Maschine
IDENTICATION_TYPE	Anzeige wie das User Schema erstellt worden ist
INSTANCE	Identifizier Nummer der aktuellen Instance
INSTANCE_NAME	Name der aktuellen Instance
IP_ADDRESS	IP-Adresse der verbundenen Host Maschine
ISDBA	Rückgabe Wert ist TRUE, wenn der Oracle User DBA Rechte hat
LANG	Die abgekürzte ISO Bezeichnung der Sprache
LANGUAGE	Anzeige der Sprache, Land und Character Set der Session
MOGULE	Anzeige des Programm Namens von DBMS_APPLICATION_INFO Package
NETWORK_PROTOCOL	Anzeige des Netzwerkprotokolls
NLS_CALENDAR	Kalendereinstellungen der Session
NLS_CURRENCY	Anzeige der Währung der Session
NLS_DATE_FORMAT	Datumsformat der Session

continued on next page.

continued from previous page.

Parameter	Beschreibung
NLS_DATE_LANGUAGE	Datumssprache der Session
NLS_SORT	Binary oder sprachliche Sortierung
NLS_TERRITORY	Anzeige der Ländereinstellung
OS_USER	Anzeige des OS Benutzers
POLICY_INVOKER	The invoker of row-level security polica functions
PROXY_ENTERPRISE_IDENTITY	The Oracle Internet Directory DN when the proxy user is an enterprise user
PROXY_GLOBAL_UID	The global user ID from Oracle Internet Directory for enterprise user security proxy users. Returns NULL for all other proxy users
PROXY_USER	The name of the user who opend the current session on behalf of SESSION_USER
PROXY_USERID	The identifier of the user who opend the current session on behalf of SESSION_USER
SERVER_HOST	Anzeige des Server Host Namens
SERVICE_NAME	Anzeige des Service Namens
SESSION_USER	Anzeige des angemeldeten Datenbank Users
SESSION_USERID	Anzeige der ID des angemeldeten Datenbank Users
SESSIONID	Anzeige der auditing Session ID
SID	Anzeige der Session ID
STATEMENTID	Anzeige der auditing Statement ID
TERMINAL	Anzeige des OS identifier von der aktuellen Session

Tabelle 1.21: Liste userenv

1.31 Oracle Version

Möchte man herausfinden, welche Oracle Version (Enterprise oder Standard) installiert worden ist, so kann man die folgende SQL Abfrage absetzen.

```
sql>SELECT * from v$version;
```

```
BANNER
```

```
-----
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
PL/SQL Release 11.1.0.6.0 - Production
CORE 11.1.0.6.0 Production
TNS for Linux: Version 11.1.0.6.0 - Production
NLSRTL Version 11.1.0.6.0 - Production
```

Listing 1.165: Oracle Version

In der Datei context.xml, die sich in dem Verzeichnis

\$ORACLE_HOME/inventory/components21/oracle.server/<version> befindet, nach dem Parameter s_serverInstallType suchen. Dort gibt es einen VAL Wert SE für Std. Edition oder EE für die Enterprise Edition.

1.32 Patch

1.32.1 Status

Den aktuellen Patch Status der Datenbank kann man auf der Kommandozeile mit dem nachfolgenden Befehl abfragen. Eine detaillierte Asugabe erhält man, wenn man noch die Option `-detail` mit angibt.

```
uws@tux>$ORACLE_HOME/OPatch/opatch lsinventory [-detail]

Invoking OPatch 11.1.0.6.0
[
Oracle Interim Patch-Installationsprogramm Version 11.1.0.6.0
Copyright (C) 2007, Oracle Corporation. All Rights reserved. Alle Rechte
  vorbehalten
]
Oracle Home: /u01/app/oracle/product/11.1.0/db_1
Zentrales Bestandsverzeichnis: /u01/app/oraInventory
Von: /etc/orainst.loc
OPatch-Version: 11.1.0.6.0
OUI-Version: 11.1.0.6.0
Speicherort der Log-Datei: $ORACLE_HOME/cfgtools/opatch/opatch2011-12-06_10
  -43-12AM.Log
.
.

sql>COL action_time FORMAT a40
sql>COL comments FORMAT a50
sql>SELECT action_time, comments, bundle_series from registry$history;
```

ACTION_TIME	COMMENTS	BUNDLE_SERIES
22.07.13 10:59:36,011161	PSU 11.2.0.3.7	PSU
14.06.14 16:52:48,635689	PSU 11.2.0.3.7	PSU

Listing 1.166: Patch Status

Die Opatch Version kann man mit `opatch version` abfragen.

```
uws@tux>$ORACLE_HOME/OPatch/opatch version
Invoking OPatch 11.2.0.1.5
OPatch Version: 11.2.0.1.5

OPatch succeeded
```

Listing 1.167: OPatch Version

1.32.2 Check Conflict

Die einzuspielende Patche können im Vorfeld überprüft werden, ob die Installation funktioniert oder fehlschlägt.

```
uws@tux>$ORACLE_HOME/OPatch\opatch prereq CheckConflictAgainstOHWithDetail -
  phBaseDir ./1334361
```

Listing 1.168: Patch check

Kommt es hierbei zu einer Fehlermeldung `OPatch failed with error code 73` und in der Log-Datei im Verzeichnis `$ORACLE_HOME/cfgtoollogs/opatch` gibt es den Eintrag `OUI-067073 invalid patch location`, so ist anstelle der Option `-phBaseDir` nur `-ph` zu nehmen.

1.32.3 Installation

Die Patche werden mit `opatch apply` installiert.


```
uws@tux>cd <PatchPatch>
uws@tux>$ORACLE_HOME/OPatch/opatch apply
```

Listing 1.169: Patch Installation

Zum Abschluss müssen noch die Datenbanken auf dem Patch Stand gebracht werden.

```
uws@tux>sqlplus / as sysdba
sql>@?/Bundel/Patch22/catcpu.sql

sql>SELECT comments FROM registry@history;

COMMNTS
-----
Patchset 11.2.0.2
Patch 22
```

Listing 1.170: Database Patch

1.32.4 Error Code 73

Wird der Patch mit der Fehlermeldung `OPatch failed with error code 73` beendet und die Fehlermeldung lautet: `Following executables are active:`

`/u01/app/oracle/product/11.2.0/dbs_1/dbs/libclntsh.so.11.1`, muss man die Library entsperren.

```
uws@tux>fuser -f /u01/app/oracle/product/11.2.0/dbs_1/dbs/libclntsh.so.11.1
```

Listing 1.171: Library entsperren

1.32.5 Error Code 74

Wird der Patch mit der Fehlermeldung `OPatch failed with error code 74` beendet, so gibt es hierfür folgenden Workaround:

- Kopieren der Datei `fuser` von `/bin` nach `/sbin` als root
- `Opatch apply` als Oracle User ausführen

Wird auf einer Windows Maschine folgende Meldung angezeigt:

`Following Files are active: C:\Oracle\product\11.2\db_1_bin\oci.dll`

so muss das Oracle Verzeichnis umbenannt werden und der Server rebooted. Nach dem Neustart ist das Verzeichnis wieder auf den alten Namen abzuändern. Danach kann der Patch applied werden.

1.32.6 Error Code 104

Wird der Patch mit der Fehlermeldung `OPatch failed with error code 104` beendet, so übergibt man dem `Opatch` Aufruf die Option `invPtrLoc` mit.

```
uws@tux>opatch apply -invPtrLoc /u01/app/oraInventory/oraInst.loc
```

Listing 1.172: Option angeben

1.33 Database Optionen enable / disable

In der Oracle Version 11G R2 und neuer gibt es ein neues Tool, um Database Options zu aktivieren oder deaktivieren. Das Tool `chopt` befindet sich im Verzeichnis `$HOME_HOME/bin` und mit diesem Tool kann man folgende Optionen aktivieren / deaktivieren. Als erstes muss die Datenbank heruntergefahren werden,

Option	Beschreibung
dm	Oracle Data Mining RDBMS Files
dv	Oracle Database Vault
lbac	Oracle Label Security
olap	Oracle OLAP
partitioning	Oracle Partitioning
rat	Oracle Real Application Testing
ode_net_2	Oracle Database Extension for .NET 2.0

Tabelle 1.22: Database Optionen

danach kann man mit dem Tool die einzelnen Optionen aktivieren / deaktivieren.

```
sql>shutdown immediate;
uws@tux>chopt disable dm
uws@tux>chop enable partitioning
```

Listing 1.173: Options disable / enable

1.34 Feature Report

Eine Auflistung, welche Features in Oracle benutzt werden, kann man mit dem Package `dbms_feature_usage_report` erhalten. Die Ausgabe kann als HTML-Seite oder als Text Datei erfolgen. Für die Ausgabe als HTML-Seite wird `display_html` und für die Text Datei `display_text` genommen.

```
sql>SPPOOL /tmp/usage_report.html
sql>SELECT output FROM table(dbms_feature_usage_report.display_html);
sql>SPPOOL off
```

Listing 1.174: Feature Report HTML

1.35 Ausführungsplan

Mit der Option `autotrace on` wird nach jedem SQL Statement der Ausführungsplan angezeigt. Hierbei muss die Tabelle `explain_plan` oder `plan_table` vorhanden sein und der Oracle User muss die Rolle `plustrace` haben. Die Rolle kann man mit `plustrace.sql` aus dem `rdbm` Verzeichnis erstellen, falls sie nicht vorhanden ist.

```
sql>SET autotrace on
```

Listing 1.175: autotrace on

Um nur den Ausführungsplan anzuzeigen, gibt man `trace explain` an.

```
sql>SET autotrace trace explain
```

Listing 1.176: Nur Ausführungsplan anzeigen

Ausgabe der Statistic.

```
sql>SET autotrace trace statistics
```

Listing 1.177: Statistic Ausgabe

Die Antwortzeiten schaltet man hiermit ein.

```
sql>SET timing on
```

Listing 1.178: Antwortzeiten einschalten

Mit dem Statement `explain plan for` kann man direct den Ausführungsplan speichern und ihn sich dann anzeigen lassen.

```
sql>EXPLAIN PLAN FOR select sysdate from dual;
```

```
EXPLAIN PLAN ausgefuehrt.
```

```
sql>SELECT * from table(dbms_xplan.display);
```

```
PLAN_TABLE_OUTPUT
```

```
-----
```

```
Plan hash value: 1388734953
```

```
-----
```

ID	Operation	Name	Rows	Cost (%CPU)	Time
0	SELECT STATEMENT		1	2 (0)	00:00:01
1	FAST DUAL		1	2 (0)	00:00:01

```
-----
```

Listing 1.179: Ausführungsplan speichern

1.36 Papierkorb

1.36.1 Anzeigen

In Oracle gibt es zwei Tabellen / Views, um sich den Inhalt des Papierkorbes anzeigen zu lassen. Einer davon ist der globale Papierkorb und der andere ist für den User papierkorb zuständig.

```
sql>SELECT * FROM recyclebin;
```

```
sql>SELECT * FROM user_recyclebin;
```

Listing 1.180: Anzeige Papierkorb

1.36.2 Leeren

Den Inhalt eines Papierkorbes kann man mit dem Befehl `purge` löschen.

```
sql>PURGE recyclebin;
```

```
sql>PURGE table <name>;
```

Listing 1.181: Papierkorb leeren

1.37 Character Set

1.37.1 Ändern

Vor dem ändern des Character Sets, muss ein Backup der Datenbank gemacht werden, da mit dem `alter database` statement kein `rollback` gemacht werden kann.

```

sqlshutdown immediate;
sql>startup mount;
sql>ALTER system enable restricted Session;
sql>ALTER system SET job_queue_processes = 0;
sql>ALTER system SET aq_tm_processes = 0;
sql>ALTER database open;
sql>ALTER database character set AL32UTF8;
sql>ALTER database character set internal_use AL32UTF8;
sql>shutdown immediate;
sql>startup;

```

Listing 1.182: Character Set ändern

Kommt die Fehlermeldung ORA-12712 bei `internal_use`, so muss der neue Zeichensatz eine Obermenge des alten Zeichensatzes sein.

1.38 Prozesse

1.38.1 Anzeigen

```

sql>SELECT pid, spid, username, program FROM v$aq_tm_processes
2>WHERE background = 1;

```

PID	SPID	USERNAME	PROGRAM
2	5547	oracle	oracle@tux (PMON)
3	5552	oracle	oracle@tux (VKTM)
4	5559	oracle	oracle@tux (DIAG)

Listing 1.183: Prozesse anzeigen

1.39 DBA errors

1.39.1 Anzeigen

```

sql>COL text FORMAT a60
sql>SELECT owner, text FROM dba_errors;

```

OWNER	TEXT
sys	PL/SQL: SQL Statement ignored
sys	PL/SQL: ORA-01775: Looping chain of synonyms

Listing 1.184: DBA errors anzeigen

1.40 Database verschieben

Soll eine Datenbank in ein anderes Verzeichnis verschoben werden, so ist folgender Ablauf nötig. Als erstes erstellen wir einen Pfile und fahren dann die Datenbank runter.

```

sql>CREATE pfile FROM spfile;
sql>shutdown immediate;

```

Listing 1.185: Pfile erstellen

Nun ändern wir die Pfade für die Control Dateien im Pfile und verschieben alle Datenbank Dateien auf ihre neue Position. Anschließend starten wir die Datenbank mit dem Pfile.

```
sql>startup mount pfile='/u01/app/oracle/product/11.2.0/db_1/dbs/initcad01.ora';
```

Listing 1.186: Database start

Alle Pfade der Datenbank Dateien werden nun abgeändert.

```
sql>COL member FORMAT a60
sql>SELECT group#, member FROM v$logfile;

GROUP# member
-----
1 /u01/oradata/cad01/redo01.log
2 /u01/oradata/cad01/redo02.log
3 /u01/oradata/cad01/redo04.log

sql>ALTER database RENAME file '/u01/oradata/cad01/redo01.log' TO
2>'/u02/oradata/cad01/redo01.log';
sql>ALTER database RENAME file '/u01/oradata/cad01/redo02.log' TO
2>'/u02/oradata/cad01/redo02.log';
sql>ALTER database RENAME file '/u01/oradata/cad01/redo03.log' TO
2>'/u02/oradata/cad01/redo03.log';

sql>COL name FORMAT a60
sql>SELECT * from v$dbfile;

FILE NAME
-----
1 /u01/oradata/cad01/system01.dbf
2 /u01/oradata/cad01/undotbs01.dbf
3 /u01/oradata/cad01/sysaux01.dbf
.

sql>ALTER database RENAME file '/u01/oradata/cad01/system01.dbf' TO
2>'/u02/oradata/cad01/system01.dbf';
sql>ALTER database RENAME file '/u01/oradata/cad01/undotbs01.dbf' TO
2>'/u02/oradata/cad01/undotbs01.dbf';
sql>ALTER database RENAME file '/u01/oradata/cad01/sysaux01.dbf' TO
2>'/u02/oradata/cad01/sysaux01.dbf';
.
.

sql>ALTER database RENAME file '/u01/oradata/cad01/temp01.dbf' TO
2>'/u02/oradata/cad01/temp01.dbf';

sql>ALTER database OPEN;
sql>CREATE spfile FROM pfile;
```

Listing 1.187: Pfade ändern

1.41 Open Cursors

Informationen über Open Cursors erhält man mit den folgenden Abfragen.

```
sql>SELECT sid, serial#, username, program
2>FROM v$session
3>WHERE status = 'ACTIVE';

SID SERIAL# USERNAME PROGRAM
-----
517 4211 oracle@tux (w000)
520 6209 SCOTT SQL Developer
```

```

.
.

sql>SELECT * FROM v$open_cursor WHERE sid=520;

sql>-- Abfrage ueber Total Curors Open by session
sql>SELECT a.value, s.username, s.sid, s.serial#
2>FROM v$sesstat a, v$statname b, v$session s
3>WHERE a.statistic# = b.statistic#
4>AND s.sid = a.sid
5>AND b.name = 'opened cursors current';

VALUE USERNAME SID SERIAL#
-----
5 SCOTT 328 526
12 SYS 2 250

```

Listing 1.188: Open Cursor

1.42 External Jobs

Werden mittels Scheduler External Jobs erstellt, die OS Befehle ausführen müssen, so ist in der Datei \$ORACLE_HOME/rdbms/admin/externaljob.ora der User und die User Gruppe zu setzen.

```

uws@tux>cat externaljob.ora

run_user = oracle
run_group = oinstall

```

Listing 1.189: Beispiel Externaljob.ora

1.43 Statspack

Für das Statspack wird bei der Installation der User perfstat angelegt. Dieser User wird dann der default tablespace und der temporary tablespace zugewiesen. Oracle empfiehlt zwar den tablespace sysaux für den default zu benutzen, aber besser ist es, einen neuen Tablespace dafür anzulegen. Nach dem löschen des Statspacks kann der Tablespace ebenso gelöscht werden.

1.43.1 Installation

Als erstes legen wir einen neuen Tablespace an.

```

sql>CREATE tablespace statspack DATAFILE
2>' /u02/app/oracle/oradata/cad01/statspack01.dbf '
3>SIZE 100m AUTOEXTEND on NEXT 50m MAXSIZE 2000m
4>EXTENT MANAGEMENT local
5>SEGMENT SPACE MANAGEMENT auto;

Tablespace created

```

Listing 1.190: Tablespace erstellen

Nun wird mit einem SQL-Script das Statspack installiert. Es wird auch der User perfstat damit erstellt. Das SQL-Script ruft folgende andere Scripte auf:

- SPCUSR.SQL => Erstellt den User perfstat
- SPCTAB.sql => Erstellt die Tabellen
- SPCPKG:SQL => Erstellt Packages

```

sql>@?/rdbms/admin/spcreate.sql
Enter Value for perfstat_password:

Enter Value for default_tablespace:

Enter Value for temporary_tablespace:

```

Listing 1.191: Statspack installieren

1.43.2 Snapshot erstellen

Ein Snapshot wird Standardmäßig mit dem Level 5 erstellt. Für den Snapshot Level sind folgende Werte zulässig: 0,5,6,7 & 10. Die Snapshot Level bauen auf den anderen Snapshot Level auf. Einen Statspack

Level	Beschreibung
Level 0	Snapshot von general statistic, rollback segment, row cache, SGA, system events, background events, session events, system statistics, wait statistics und latch informationen
Level 5	Zusätzlich werden Informationen zu high resource usage SQL statements gesammelt
Level 6	SQL Plan und SQL Plan usage information für high resource usage SQL statements
Level 7	Segment Level Statistiic, including logical and physical read, row lock, itl und buffer busy waits
Level 10	Child Latch statistics

Tabelle 1.23: Statspack Level

Snapshot wird nun erstellt.

```

sql>connect perfstat/<Passwd>

sql>execute statspack.snap

sql>-- Snapshot mit Level 7
sql>execute statspack.snap (i_snap_level => 7);

```

Listing 1.192: Statspack Snapshot erstellen

Den Wert dauerhaft speichern wird mit `i_modify_parameter` gemacht und die Werte können mit dem View `stats$statspack_parameter` abgefragt werden.

```

sql>execute statspack.snap (i_snap_level => 7, i_modify_parameter => 'true')
;

sql>SELECT * FROM stats$statspack_parameter;

```

Listing 1.193: Statspack Parameter speichern

1.43.3 Snapshot Job

Mit dem Script `spauto.sql` im Verzeichnis `$ORACLE_HOME/rdbms/admin` wird ein Datenbank Job eingerichtet, der zu jeder vollen Stunde mit dem Snapshot Level 5 läuft. Damit es zu keiner Überschneidung mit AWR kommt, ist es besser den Job eine halbe Stunde später auszuführen.

```

sql>connect perfstat/<Passwd>

sql>@?/rdbms/admin/spauto.sql

sql>-- Job einmal pro Stunde ab 10:30 uhr ausfuehren
sql>exec dbms_job.change (1, NULL, to_date('18.03.2013 10:30:00', 'DD.MM.
      YYYY hh24:mi:ss'), 'sysdate+(1/48)');

sql>-- Aendern des Intervalls

```

```

sql>exec dbms_job.change (1, NULL, NULL, 'sysdate+(1/48)')

sql>-- Anzeige des Jobs
sql>SELECT job, next_date, next_sec, interval FROM user_jobs;

JOB NEXT_DATE          NEXT_SEC INTERVAL
-----
1 18.03.2013 10:30:00 10:30:00 sysdate+(1/48)

```

Listing 1.194: Statspack Job

1.43.4 Report erstellen

Nach einiger Zeit kann dann ein Report erstellt werden. Nach dem Aufrufen des Scripts spreport.sql werden alle Snapshots aufgelistet. Nach der Eingabe der Snap-ID wird der Report erstellt. Der Standardmäßige Dateiname lautet: sp_<start_snap_id>_<end_snap_id>.lst.

```

sql>connect perfstat/<Passwd>

sql>@?/rdbms/admin/spreport.sql

```

Listing 1.195: Report erstellen

Im Silent Modus gibt man vorher die definierten Variablen an.

```

sql>connect perfstat/<Passwd>
sql>DEFINE begin_snap=1
sql>DEFINE end_snap=5
sql>DEFINE report_name=snap1_5
sql>@?/rdbms/admin/spreport.sql

```

Listing 1.196: Report Silent

1.43.5 Snapshot löschen

Um alte Snapshots zu löschen, gibt es das Script sppurge.sql. Damit werden alle Snapshots gelöscht.

```

sql>connect perfstat/<Passwd>

sql>@?/rdbms/admin/sppurge.sql

```

Listing 1.197: Snapshot löschen alle

Einzelne Snapshots kann man folgendermaßen löschen.

```

sql>DELETE FROM stats$snapshot where snap_id < 1000;

sql>DELETE FROM stats$sql WHERE snap_time < sysdate-180;

```

Listing 1.198: Snapshot löschen Query

Einen Lösch Job kann man mit statspack.purge einrichten.

```

sql>EXEC statspack.purge(<days>);

```

Listing 1.199: Lösch Job

1.43.6 Snapshot anzeigen

Erstellte Snapshots werden in der Tabelle stats\$snapshot verwaltet.

```

sql>connect perfstat/<Passwd>

sql>SELECT snap_id, snap_time, snapshot_exec_time FROM stats$snapshot;

```


SNAP_ID	SNAP_TIME	SNAPSHOT_EXEC_TIME
1	18.03.2013 15:30:00	129.7
2	18.03.2018 16:30:00	127,4

Listing 1.200: Snapshot anzeigen

1.43.7 Statspack Analyse

Im Internet gibt es die Web Seite burleson-dba.com/SP, in der man den Report in einem Feld einfügt und dann die Auswertung des Reports erhält.

Ebenso gibt es von Oracle das Programm Trace File Analyzer (TFA). Das Programm gibt es für Linux und Windows.

Auch mit dem Programm Orasrp kann man Trace Files auswerten.

Syntax: `orasrp --google-charts tracefile.trc output.html`

1.43.8 Statsack löschen

Nach dem löschen des Statspacks kann auch der Tablespace gelöscht werden. Das SQL-Script `spdrop.sql` ruft folgende andere Scripts auf:

- SPDTAB.SQL => Drop Tables, sysnonyms, Packages
- SPDUSR.SQL => Drop User perfstat

```
sql>connect perfstat/<Passwd>
sql>@?/rdbms/admin/spdrop.sql
sql>connect sys/<Passwd> as sysdba
sql>drop tablespace statspack cascade;
```

Listing 1.201: Statspack löschen

1.44 AWR

Das Automatic Workload Repository (AWR) gibt es seit der Version 10g und sammelt auch Performance Informationen über die Datenbank. Der Unterschied zu dem Statspack ist, das die AWR Lizenz kostenpflichtig ist. AWR ist in allen Oracle Datenbanken installiert und sammelt zu jeder vollen Stunde Informationen über die Datenbank. Laut Oracle begeht man auch einen Lizenzbruch, wenn man die Tabellen `DBA_HIST_*` in eigene SQL Abfragen benutzt.

Folgende Tablllen dürfen laut Oracle trotzdem benutzt werden.

- DBA_HIST_SNAPSHOT
- DBA_HIST_DATABASE_INSTANCE
- DBA_HIST_SNAP_ERROR
- DBA_HIST_SEG_STAT
- DBA_HIST_SEG_STAT_OBJ
- DBA_HIST_UNDOSTAT

Mit der nachfolgenden Abfrage kann man sich anzeigen lassen, ob AWR schon mal benutzt worden ist und ob es aktiv ist.

```

sql>COL name FORMAT a40
sql>COL detected FORMAT 9999
sql>COL samples FORMAT 9999
sql>COL used FORMAT a5
sql>COL interval for 9999999
sql>SET linesize 200
sql>SET pagesize 80
sql>SELECT
  2> name,
  3> detected_usages detected,
  4> total_samples samples,
  5> currently_used used,
  6> to_char(last_sample_date, 'DD.MM.YYYY HH24:MI') Last_Sample,
  7> sample_interval interval,
  8>FROM dba_feature_usage_statistics
  9>WHERE name = 'Automatic Workload Repository'
10>OR name like 'SQL%';

```

Name	DETE	SAMPLES	USED	LAST_SAMPLE
INTERVAL				

Automatic Workload Repository 604800	2	239	FALSE	16.12.2017 00:02
SQL Monitoring and Tuning pages				
SQL Tuning Advisor				
SQL Worload Manager				
.				
.				

Listing 1.202: AWR aktiv

1.44.1 Diagnostic / Tuning Pack deaktivieren

Um das Diagnostic / Tuning Pack zu deaktivieren, gibt es den Oracle Parameter `control_management_pack_access`. Für diesen Parameter gibt es drei Werte: `DIAGNOSTIC+TUNING`, `DIAGNOSTIC` und `NONE`.

```
sql>ALTER system SET control_management_pack_access = NONE scope=both;
```

Listing 1.203: Pack deaktivieren

1.45 Oracle Trace

Mit dem Event 10046 kann man sich selbst einen Trace File erstellen. Dieser Event erfasst alle Anwendungs-SQL.

1.45.1 Methode 1

Der Parameter `sql_trace` wird auf `true` gestellt und der gesamte SQL-Code wird erfasst. Die erstellte Trace Datei kann sehr groß werden. Um dieses zu begrenzen, gibt es den Parameter `max_dump_file_size`.

```

sql>ALTER session SET max_dump_file_size=2048; -- Angabe in bytes
sql>ALTER session SET sql_trace=true;

sql>ALTER session set sql_trace=false; -- Trace ausschalten

```

Listing 1.204: Methode 1

1.45.2 Methode 2

```
sql>ALTER session SET timed_statistics=treu;
sql>ALTER session SET max_dump_file_size=unlimited;
sql>ALTER session SET trace_file_identifier='uws'; -- char(60)
sql>ALTER session SET events '10046 trace name context forever, level 12';
sql>SELECT * from dba_data_files;
sql>ALTER session SET events '10046 trace name context off';
```

Listing 1.205: Methode 2

Die erstellte Trace Datei wird im User Dump Bereich abgelegt.

```
sql>SHOW parameter user_dump_dest

NAME                TYPE        VALUE
-----
user_dump_dest      string      /u01/app/oracle/diag/rdbms/cad01/cad01/trace
```

Listing 1.206: User Dump Bereich

Für das Event 10046 gibt es 5 verschiedene Levels.

Level	Beschreibung
0	Kein Trace
1	Parse-, Ausführungs- und Fetch-Statistiken
4	Level 1 und SQL Bindevariablen-Inhalten
8	Level 1 und Oracle Wait-Ereignis Informationen
12	Level 1, 4 und 8 zusammen

Tabelle 1.24: Trace Levels

1.45.3 Auswerten

Die Auswertung des erstellten Trace-Files wird mit dem Programm tkprof bemacht.

```
uws@tux>tkprof <TraceFile> <OutputFile> print=10 sort=fchela
```

Listing 1.207: Auswerten

1.46 SQL Statements anzeigen

Um abgesetzte SQL-Statements sich anzeigen zu lassen, gibt es verschiedene Views. Die wichtigsten sind: v\$sql, v\$sqlarea, v\$sqltext. Es gibt noch mehrere Views, in denen Informationen über die abgesetzten SQL-Statements enthalten sind.

```
uws@tux>cat ShowSqlStatements.sql
SELECT
  module ,
  sql_text ,
  username ,
  disk_reads_per_exec ,
  buff_gets ,
  disk_reads ,
  parse_calls ,
  sorts ,
  executions ,
  rows_processed ,
  hit_ratio ,
  first_load_time ,
```

```

sharable_mem ,
persistent_mem ,
runtime_mem ,
cpu_time ,
elapsed_time ,
address ,
hash_value
FROM
(SELECT
  module ,
  sqltext ,
  u.username ,
  round((s.disk_reads/decode(s.executions,0,1 s.executions)),2)
    disk_reads_per_exec ,
  s.disk_reads ,
  s.buffer_gets ,
  s.parse_calls ,
  s.sorts ,
  s.executions ,
  s.rows_processed ,
  100 - round(100*s.disk_reads/greatest(s.buffer_gets,1),2) hit_ratio ,
  s.first_load_time ,
  sharable_mem ,
  persistent_mem ,
  runtime_mem ,
  cpu_time ,
  elapsed_time ,
  address ,
  hash_value
FROM
  sys.v_sql s ,
  sys.all_users u
WHERE
  s.parsing_user_id=u.user_id
  and upper(u.username) not in ('SYS','SYSTEM')
ORDER BY
  4 desc
)
WHERE
rownum <= 40;

```

Listing 1.208: Show SQL-Statement

Die Werte für address und hash_value stammen aus der obigen Abfrage.

```

uws@tux>cat ShowSqlText.sql
SELECT
*
FROM
  sys.v_$sqltext
WHERE
  address = '000007FFB28D3FE8'
AND
  hash_value = '1545474800'
ORDER BY
  address ,
  hash_value ,
  command_type ,
  piece;

```

Listing 1.209: Sql Text

1.47 Logging / Nologging

Ein Logging / Nologging kann für die ganze DB, Tablespaces, Tables und Indexe gesetzt werden. Standardmäßig ist das Logging eingeschaltet.

```
sql>ALTER table <TableName> nologging;

sql> ALTER tablespace <TablespaceName> nologging;

sql>ALTER database nologging;

sql>ALTER table <TableName> [force] logging;
```

Listing 1.210: Beispiele Logging / Nologging

Für welche Tabellen ein Logging eingeschaltet ist, kann man mit der nachfolgenden Abfrage sich anzeigen lassen.

```
sql>SELECT owner, table_name, tablespace_name, logging FROM dba_tables;

sql>SELECT table_name, logging FROM user_tables WHERE logging = 'NO';

TABLE_NAME LOG
-----
CITY        No
CoOUNTRY   No
```

Listing 1.211: Show Table logging

1.48 HugePages

Unter Linux gibt es schon seit längerer Zeit die HugePages. Für die Speicherverwaltung des Shared Memory werden die sog. Page Tables verwendet. Diese werden unter Linux für jeden Prozess kopiert. Da der Server mit Speicherseiten von 4kB arbeitet, kommt man bei 8 Byte pro Speicherseite für die Page Tabele, bei einer SGA von 30GB und von etwa 200 Prozesse, auf:

$$30 \text{ GB} / 4 \text{ kB} * 8 \text{ Byte/Prozess} * 200 \text{ Prozesse} = 12 \text{ GB}$$

Die HugeTables können nicht mit AMM (Automatic Memory Management) benutzt werden. Es sollte also nicht der Parameter MEMORY_TARGET und MEMORY_MAX_TARGET gesetzt sein. Besser die Parameter SGA_TARGET und SGA_MAX_TARGET.

Die HugePages Einstellungen kann man folgendermaßen ermitteln.

```
uws@tux>grep -i "Huge" /proc/meminfo

HugePagesTotal:    0
HugePages_Free:    0
HugePages_Rsvd:    0
HugePagesSize:    2048 kB
```

Listing 1.212: HugePages

Man kann die Größe der HugePages auch mit einer Faustformel ermitteln. Hierbei wird die Größe der SGA in MB durch die Hugepagesize geteilt und als Reserve werden noch 10 hinzugefügt. Bei einer SGA Größe von 30 GB ergibt sich folgende HugePages:

$$30720 \text{ MB} / 2 \text{ MB} + 10 = 15365$$

Von dem Oracle Support (ID 401749) gibt es ein Script, das die Größe der HugePage bei einer laufenden DB ermittelt. Das nachfolgende Script ist ein älteres Script. Auf meiner Download Seite kann man das neuere Script herunterladen.

```
uws@tux>cat HugePages.sh
#!/bin/env bash
#
```

```
# Setting check for HugePages
# The Script is from:
# docs.oracle.com/cd/E37670_01/E37355/html/ol_config_hugepages.html
#
KERN='uname -r | awk -F. '{printf("%d.%d\n", $1, $2);}'
#
# Find HugePage size
#
HPG_SZ='grep HugePageSize /proc/meminfo | awk {'print $2'}'
NUM_PG=1
for SEG_BYTES in `ipcs -m | awk {'print $5'} | grep "[0-9][0-9]*" `
do
  MIN_PG='echo "$SEG_BYTES/($HPG_SZ*1024)" | bc -q'
  if [ $MIN_PG -gt 0 ]; then
    NUM_PG='echo "$NUM_PG+$MIN_PG+1" | bc -q'
  fi
done
#
# Finish and output the results
#
case $KERN in
  '2.4') HUGETLB_POOL='echo "$NUM_PG+$HPG_SZ/1024" | bc -q';
        printf "\nRecommend setting: vm.hugetlb_pool = $HUGETLB_POOL\n";;
  '2.6') printf "\nRecommend setting: vm.nr_hugepages = $NUM_PG\n";;
  *)    printf "\nUnrecognized kernel version &KERN. Exiting.\n";;
esac

uws@tux>./HugaPages.sh

Recommended setting: vm.nr_hugepages = 3848
```

Listing 1.213: HugaPages.sh

Die HugePages werden dann aktiviert mit:

```
uws@tux>echo 3848 >/proc/sys/vm/nr_hugepages
```

Listing 1.214: HugaPages aktivieren

Damit diese Einstellung auch einen Reboot überlebt, so trägt man in der /etc/sysctl.conf folgende Zeile ein.

```
uws@tux>grep huge /etc/sysctl.conf

vm.nr_hugepages = 3848
```

Listing 1.215: sysctl.conf

Zum Abschluß der Konfiguration muss der Wert noch in der Datei /etc/security/limits.conf gesetzt werden. Hierzu wird der Wert der HugePages mal den Wert der HugePageSize genommen. In diesem Beispiel also $3848 \cdot 2048 = 7880704$.

```
uws@tux>grep -i "memlock" /etc/security/limits.conf

oracle soft memlock 7880704
oracle hard memlock 7880704
```

Listing 1.216: limits.conf

Nach einem Neustart der Datenbank, sthet dann in der alert<sid>.log folgendes.

```
***** Large Pages Information *****
Per process system memlock(soft) limit = 7.5GB

Total Shared Global Region in Large Pages = 7.5GB (100%)
```

```

Large Pages used by this instance: 3848 (7.5GB)
Large Pages unused system wide = 2 (4096 KB)
Large Pages configured system wide = 3848 (7.5GB)
Large Page size = 2048 KB
*****

```

Listing 1.217: Auszug alert<sid>.log

Für die Huge Pages gibt es den Initialisierungs Parameter `use_large_pages` mit den Optionen `TRUE`, `FALSE`, `AUTO` oder `ONLY`.

`TRUE` = Nutzt die SGA, wenn keine Huge Pages mehr vorhanden sind.

`FALSE` = Keine Huge Pages verwenden

`ONLY` = Instanz startet nicht, wenn keine Huge Pages aktiviert werden können => ORA-27102

`AUTO` = Ab 11.2.0.3 wird die Verwendung automatisch erzwungen

Links für weitere Informationen:

https://www.pipperr.de/dokuwiki/doku.php?id=linux_huge_pages_linux_oracle

<https://oraculix.wordpress.com/2015/05/21/linux-huge-pages-schnellstart/>

1.49 ASH

Steht in der `alert<oracle_sid>.log` die Meldung: Active Session History (ASH) performed an emergency flush. This may mean that ASH is undersized. Die Total Size wird in Bytes angegeben.

```

sql>SELECT total_size, awr_flush_emergency_count FROM v$ash_info;

TOTAL_SIZE  AWR_FLUSH_EMERGENCY_COUNT
-----
33554432    30

sql>ALTER system SET "_ash_size" = 50331648;

```

Listing 1.218: ASH

Einen Report kann man mit `@?/rdbms/admin/ashrpt.sql` erstellen.

1.50 Schema Größe

Die Größe der einzelnen Schemas kann man mit den folgenden Abfragen sich anzeigen lassen.

```

sql>connect scott/tiger@example
sql>SELECT sum(bytes/1024/1024) "MB" FROM user_segments;

MB
---
6.435

sql>COL sizet HEADING 'Total|Schema|Size MB'
sql>COL owner HEADING 'Schema|Name'
sql>SELECT owner, sum(bytes/1024/1024) sizet FROM dba_segments
2>GROUP BY owner
3>ORDER BY sized desc;

          Total
Schema  Schema
Name    Size MB
-----
Scott   6.435

```

Listing 1.219: Show Schema Size

1.51 Listener

1.51.1 Logging Disable

Bei vielen Verbindungen kann die Log-Datei sehr groß werden. Das setzen des current Listeners muss nur dann erfolgen, wenn es mehrere Listener auf dem Server gibt, sonst kann er entfallen.

```
uws@tux>lsnrctl
lsnrctl>set current listener <ListenerName>
lsnrctl>set log_status off
lsnrctl>exit
```

Listing 1.220: Logging Disable

1.51.2 TNS_ADMIN

Mit der Variable TNS_ADMIN kann man den Pfad angeben, wo die listener.ora sich befindet. Unter Windows muss in der Registry unter HKLM\Software\Oracle_<OracleHome> ein neuer String mit dem Namen TNS_ADMIN erstellt werden. Dieser String bekommt den Wert <%ORACLE_HOME%>\network\admin zugewiesen.

Bild einfügen!!!

1.51.3 IPC Protokoll

In der listener.ora kann der IPC Eintrag gelöscht werden. Diese Verbindungsart wird nicht mehr benötigt. Anschließend muss der Listener neu gestartet werden.

```
uws@tux>grep -i "IPC" listener.ora
#(ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
```

Listing 1.221: IPC Protokoll

1.52 Invalid Objects

1.52.1 Anzeigen

Alle Objekte werden in der Datenbank in der Tabelle dba_registry verwaltet.

```
sql>COL comp_id FORMAT a10
sql>COL comp_name FORMAT a30
sql>COL version FORMAT a10
sql>COL status FORMAT a10

sql>SELECT comp_id, comp_name, version, status
2>FROM dba_registry
3>WHERE status = 'INVALID';
```

COMP_ID	COMP_NAME	VERSION	STATUS
ORDIM	Oracle Multimedia	11.2.0.3.0	INVALID

Listing 1.222: Invalid Objects

1.53 DB Console

Gibt es mehrere Instanzen auf einer Maschine, so muss man vorher die Variable ORACLE_UNQNAME setzen. Der Wert für diese Variable ist der Wert der ORACLE_SID. Eventuell muss auch die Variable ORACLE_HOSTNAME gesetzt werden.


```
uws@tux>export ORACLE_UNQNAME=cad01
uws@tux>export ORACLE_HOSTNAME=tux.home.net
uws@tux>emctl status dbconsole
uws@tux>emctl start dbconsole
uws@tux>emctl stop dbconsole
```

Listing 1.223: DB Console

1.53.1 Enterprise Agent

Den Enterprise Agenten kann man folgendermaßen stoppen und starten.

```
uws@tux>cd <AgentPath>\bin
uws@tux>export ORACLE_UNQNAME=cad01
uws@tux>export ORACLE_HOSTNAME=tux.home.net
uws@tux>emctl status agent
uws@tux>emctl start agent
uws@tux>emctl stop agent
```

Listing 1.224: Enterprise Agent

1.54 Change Schema

Den Zugriff auf ein anderes Schema während einer Sqlplus Session zu wechseln, kann man folgendermaßen machen.

```
sql>ALTER session SET current_schema = <SchemaName>;
sql>ALTER session SET current_schema = scott;
Session wurde geaendert.
```

Listing 1.225: Change Schema

1.55 Alert Log

In der Datenbank gibt es verschiedene Views / Tabellen, indem die Alerts abgelegt werden.

- dba_alert_history
- dba_outstanding_alerts
- dbms_alert_info
- x\$dbgalertext
- v\$diag_alert_ext

```
sql>COL host_id FORMAT a10
sql>SELECT instance_name, host_id, time_suggested, reason, suggested_action
2>FROM dba_alert_history;

INSTANCE_NAME HOST_ID TIME_SUGGESTED REASON
SUGGESTED_ACTION
-----
-----
cad10 tux 08.03.2015 12:03:33,41700 +02:00 Instanz cad10.cad10
auf Knoten TUX zum Zeitpunkt ... Keine Aktion erforderlich

sql>COL component_id for a10
```

```

sql>COL filename for a80
sql>SELECT distinct component_id, filename from v$diag_alert_ext;

COMPONENT_  FILENAME
-----
clients     C:\ORACLE\diag\clients\user_adm-joba\host_32610>alert\log.xml
rdbms       C:\ORACLE\diag\rdbms\cad10\cad10>alert\log.xml

sql>SELECT originating_timestamp, message_text
2>FROM v$diag_alert_ext
3>WHERE component_id like '%rdbms%'
4>ORDER BY originating_timestamp;

```

Listing 1.226: Alert Query

Anstelle der where Bedingung kann auch die Bedingung where trim(component_id)='rdbms' genommen werden.

1.56 Resource Limits

Die Resource Limits kann man sich mit der folgenden Abfrage anzeigen lassen.

```

sql>SELECT resource_name ,
2>current_utilization ,
3>max_utilization ,
4>limit_value
5>FROM v$resource_limit;

RESOURCE_NAME  CURRENT_UTILIZATION  MAX_UTILIZATION  LIMIT_VALUE
-----
processes      45                   226              500
sessions       53                   298              800
enqueueer_locks 43                   426             10960
.
.

```

Listing 1.227: Resource Limits

1.57 Dispatcher

In den nachfolgenden Views kann man sich Informationen über den Dispatcher anzeigen lassen.

- v\$queue
- v\$dispatcher
- v\$dispatcher_rate

Der Wert des Dispatchers kann zur Laufzeit geändert werden.

```

sql>SHOW parameter dispatcher

NAME          TYPE          VALUE
-----
Dispatchers  string       (PROTOCOL=TCP)(SERVICE=cad10XDB)

sql>ALTER system SET dispatchers = '(PROTOCOL=TCP) (SERVICE=cad01XDB)
2>(PRE=oracle.aurora.server.SGIopServer)' scope=both;

```

Listing 1.228: Dispatcher

Kapitel 2

Errors & Error Code

2.1 ORA-00020

Alls fehlermeldung erscheint Maximum number of processes (%s) exceeded. Ein Anmelden an die Datenbank funktioniert nun nicht mehr, da die maximale Anzahl an Prozesse erreicht ist. Ab der Oracle Version 10.1 gibt es den Parameter `-prelim`, mit dem man sich an die Datenbank Anmelden kann, um ein Shutdown durchzuführen.

```
uws@tux>sqlplus -prelim / as sysdba
```

Listing 2.1: ORA-00020

2.2 ORA-00392

ORA-00392: Log 3 of thread is being cleared, operation not allowed.

Lösung:

```
sql>ALTER database clear unarchived logfile group 3;
```

```
sql>ALTER datbase open resetlogs;
```

Listing 2.2: ORA-00392

2.3 ORA-609

Der Wert des Parameters `INBOUND_CONNECT_TIMEOUT` sollte erhöht werden. Standardmäßig sind 60 Sekunden vorgesehen. Dieser Parameter wird in der `listener.ora` und der `sqlnet.ora` eingetragen. Anschließend ist der Listener neu zu starten.

- Listener.ora: `INBOUND_CONNECT_TIMEOUT_<ListenerName> = 120`
- sqlnet.ora: `SQLNET.INBOUNT_CONNECT_TIMEOUT = 180`

```
uws@tux>lsnrctl stop
```

```
uws@tux>lsnrctl start
```

Listing 2.3: Listener restart

2.4 ORA-01110 / ORA-01113

Kommt es nach einem Online setzen eines Tablespaces / Datafiles zu einem ORA-01113: For file '12' is Media recovery needed, so führt man ein Recover des DB Files durch

```
sql>recover datafile '/u02/app/oracle/oradata/cad10/uws.dbf';
Media recovery success
```

Listing 2.4: Recover DB File

2.5 ORA-01187

In der Log-Datei von einem Rman Job gibt es folgende Fehlermeldung:

ORA-01187: cannot read from file 201 because is failed verification test.

```
sql>SELECT file_name FROM dba_temp_files;
select file_name from dba_temp_files
      *
ERROR at line 1:
ORA-01187: cannot read from file 201 because is failed verification tests
ORA-01110: data file 201: '/u02/app/oracle/oradata/cat10/temp01.dbf'
```

Listing 2.5: ORA-01187

Lösung: Einen neuen Temporären Tablespace anlegen und den alten löschen.

```
sql>CREATE temporary tablespace temp2 tempfile
 2>' /u02/app/oracle/oradata/cad10/temp02.dbf' SIZE 100m reuse
 3>AUTOEXTEND on NEXT 10m MAXSIZE 32000m;

sql>ALTER database default temporary tablespace temp2;

sql>DROP tablespace temp including contents and datafiles;
```

Listing 2.6: Temp. Tablespace

2.6 ORA-01536

Kommt es bei dem anlegen einer Tabelle oder View zu der Fehlermeldung ORA-01536: Space Quota exceeded tor tablespace, so muss eine Berechtigung gesetzt werden.

```
sql>GRANT unlimited tablespace TO <User>;
```

Listing 2.7: ORA-01536

2.7 ORA-01555

Die Fehlermeldung ORA-01555; Snapshot to old kommt, wenn man Daten ändern (insert / update) und in einer anderen Session eine Abfrage auf die Tabelle macht. Der Parameter UNDO_MANAGEMENT sollte auf AUTO stehen und dann kann man den Parameter UNDO_RETENTION erhöhen.

```
sql>ALTER system SET undo_retention = 3600 scope=both;
```

Listing 2.8: ORA-01555

2.8 ORA-01591

Lösung:

```
sql>SELECT local_tran_id, global_tran_id, state, fail_time, force_time
2>retry_time, os_user, os_terminal, host, db_user
3>FROM dba_2pc_pending
4>WHERE state = 'prepared';

sql>commit force '<local_tran_id>';
```

Listing 2.9: ORA-01591

2.9 ORA-01624

Bei dem Versuch eine Redo-Log datei zu löschen, die aber noch nicht archiviert worden ist, kommt es zur folgenden Fehlermeldung.

```
sql>ALTER database DROP logfile group 2;
alter database drop logfile group 2
*
*
ERROR at line 1:
ORA-01624: Log 2 needed for crash recovery of instance cad01 (thread 1)
ORA-00312: Online-Log 2, Thread 1: '/u02/app/oracle/oradata/cad01/redo01.log
'
ORA-00312: Online-Log 2, Thread 1: '/u02/app/oracle/oradata/cad01/redo01a.
log'
\end\{lstlisting}
Damit man die Redo-Log Gruppe 1 "oschen kann, muss ein checkpoint manuell
durchgef"uhrt werden.
\listBash
\begin\{lstlisting}[captionpos=b, caption=Checkpoint]
sql>ALTER system CHECKPOINT;

System altered
```

Listing 2.10: ORA-01624

2.10 ORA-02030

```
sql>GRANT select on v$transaction to <UserName>;
ORA-02030: can only select from fixed tables/views

sql>select object_name, object_type FROM dba_objects
2>WHERE object_name = 'V$TRANSACTION';

OBJECT_NAME      OBJECT_TYPE
-----
V$TRANSACTION    SYNONYM

sql>SELECT owner, synonym_name, table_owner, table_name FROM dba_synonyms
2>WHERE synonym_name = 'V$TRANSACTION';

OWNER      SYNONYM_NAME      TABLE_OWNER  TABLE_NAME
-----
PUBLIC     v$TRANSACTION     SYS           V_$TRANSACTION

sql>GRANT select ON v$TRANSACTION to <UserName>;
```

Listing 2.11: ORA-02030

2.11 ORA-03135

Der Wert des Parameters `INBOUND_CONNECT_TIMEOUT` sollte auf 0 gesetzt werden. Anschließend ist der Listener neu zu starten.

- `listener.ora: INBOUND_CONNECT_TIMEOUT_<ListenerName> = 0`
- `sqlnet.ora: SQLNET.INBOUND_CONNECT_TIMEOUT = 0`

```
uws@tux>lsnrctl stop
uws@tux>lsnrctl start;
```

Listing 2.12: ORA-03135

2.12 ORA-04061 / ORA-04068

```
ORA-04068: existing state of Packages of packages has been discarded
ORA-06061: exlstng state of package body "<Name>" has been invalidated
```

Listing 2.13: ORA-04061

Action:
Try again

2.13 ORA-12514

Ein `tnsping` auf eine Instanz / Service funktioniert, aber eine `Sqlplus` Verbindung gibt die Fehlermeldung `ORA-12514 TNS: could not resolve the connect identifier specified` aus. Hierbei muss der Service Name der Datenbank überprüft werden. Dieser Wert muss dann in der `tnsnames.ora` eingetragen werden.

```
sql>SHOW parameter service

NAME                TYPE        VALUE
-----
Service_name        string      cad10.tux.seabaer-ag.de

uws@tux>grep cad10 tnsnames.ora

(SERVICE_NAME = cad10.tux.seabaer-ag.de)
```

Listing 2.14: Check Service Name

2.14 ORA-12518

Ein `tnsping` auf eine Instanz / Service funktioniert, aber eine `Sqlplus` Verbindung gibt die Fehlermeldung `ORA-12518` aus, so muss der Datenbank Dienst für die Instanz noch erstellt werden. Hierbei ist darauf zu achten, dass die Password Datei mit dem richtigen Namen angelegt worden ist.

```
C:\Oracle>orapwd file=pwd<Oracle_SID>.ora password=oracle
C:\Oracle>oradim -new -sid <Oracle_SID> -initpwd -spfile
```

Listing 2.15: Create Service

2.15 ORA-12577

Auf dem Server wurde Oracle und auch der Oracle Client installiert und bei der Anmeldung an der Datenbank kommt die Fehlermeldung ORA-12577: TNS: Protocol Adapter not loaded, so muss in der Path Variable als erstes der Pfad zu Oracle stehen, danach erst der Pfad zu dem Oracle Client.

2.16 ORA-12637

Die Anmeldung an einer Instanz schlägt mit der Fehlermeldung fehl:

ORA-12637: Packet received failed

In der `sqlnet.ora` den Wert von `SQLNET.AUTHENTICATION_SERVIE` auf `NONE` stellen.

2.17 ORA-12638

Eine lokale Anmeldung mit `Sqlplus` an die Datenbank schlägt fehl und es kommt die Fehlermeldung:

ORA-12638: Credentials retrieval failed

In der `sqlnet.ora` den Wert von `SQLNET.AUTHENTICATION_SERVICE` von `(NTS)` auf `(NONE)` abzuändern.

2.18 ORA-39127

Nach dem löschen der OLAP Option bekommt man bei einem Datapump Export die folgende Fehlermeldung.

```
ORA-39127: unexpected error from call to export_string :=SYS.DBMS_CUBE_EXP .
  schema_info_exp('SYSTEM',0,1,'11.01.00.00.00',newblock)
ORA-06550: line 1, column 13:
PLS-00201: identifier,SYS.DBMS_CUBE_EXP' must be declared
ORA-06550: line 1, column 7:
PL/SQL: Statement ignored
ORA-06512: at "SYS.DBMS_METADATA", line 5929
```

Listing 2.16: ORA-39127

Das Package `DBMS_CUBE_EXP` muss aus dem Dictionary gelöscht werden.

```
sql>CREATE table sys.exppkgact$_backup as select * FROM sys.exppkgact$;

sql>DELETE FROM sys.exppkgact$ WHERE
  2>package = 'DBMS_CUBE_EXP' and schema = 'SYS';

sql>commit;
```

Listing 2.17: Delete Package

2.19 TNS-00505

Diese Meldung ist keine Fehlermeldung, sondern eine Informations Meldung. Die einzige Möglichkeit besteht, alle TNS Meldungen auszuschalten mit dem Parameter `DIAG_ADR_ENABLED`.

`listener.ora`: `DIAG_ADR_ENABLED_<ListenerName> = OFF`

`sqlnet.ora`: `DIAG_ADR_ENABLED = OFF`

```
uws@tux>lsnrctl stop
uws@tux>lsnrctl start
```

Listing 2.18: Listener restart

2.20 Library Cache

Im Alert.log steht folgende Meldung:

```
Memory Notification: Library Cache Object loaded into SGA
Heap Size 2129K exceeds notification threshold (2048K)
Details in trace file /u01/.../.../cad01_ora_4711.trc
```

Listing 2.19: Alert.log Meldung

Diese Meldung besagt, dass der Wert den eingestellten Wert des versteckten Parameter `_kg1_large_heap_warning_threshold` übersteigt. Mit dem nachfolgenden Select Statement kann man sich den Wert anzeigen lassen:

```
sql>SELECT * FROM (
2>SELECT a.ksppinm as parameter,
3>a.ksppdesc AS description,
4>b.ksppstvl AS session_value,
5>c.ksppstvl AS instance_value
6>FROM x$ksppi a,
7>x$ksppsv b,
8>x$ksppsv c
9>WHERE a.indx = b.indx
10>a.indx = c.indx
11>AND a.ksppinm LIKE '/_%' ESCAPE '/'
12>ORDER BY a.ksppinm)
13>WHERE parameter in ('_kg1_large_heap_warning_threshold');
```

Listing 2.20: Select Statement

Dieser Wert kann mit einem `alter system` Befehl erhöht werden. Anschließend muss die Datenbank neu gestartet werden.

```
sql>ALTER system SET '_kg1_large_heap_warning_threshold' = 244140625 scope=
spfile;
```

Listing 2.21: Increase Value