

Oracle ab V9

- pfile ändern

- create pfile='.....init.ora' from spfile;

- vi

- shutdown immediate

~~startup~~

- startup pfile='';

- create spfile from pfile='';

- connect => sqlplus / as sysdba;

22.06.09

Redo abschalten

Oracle Tuning

clz 1

clz 7

101010

STEP

FRZ

CHZ

=> RB explicit nutzen

sub transaction use Rollbacksecken [Name]

~~AD~~ (sobald commit ist RB wieder frei, nach commit wieder on)

=> RB sobald frei ist offline schalten
aber rollbacksystem ... offline (default?)

DB Link: RB in update DB nicht benutzt $\nabla \nabla$
 $\circ \circ$

Archive Redos

Statement bezogen) in Statement (select ...) no logging (schreibt keine Redos)

Objekt bezogen) Table, Tablespace, => no logging

Analyse: analyse auf große Tab nur mit (estimate)

~~g~~

Blocks: nie unter 8k limitiert 126 256

22.06.04

Oracle Tuning

dbms_stats -> V.S. analyze

analyze

```

begin;
dbms_stats gather, cascade = true;
end;

```

=>

Performance

- Tablespace auf locally managed setzen

wesentlich performanter bei updates & deletes als "Dictionary" ~~und DWA Studie~~

Index Typ =

Bitmap Index => nur sinnvoll bei J/N oder Star Schema (wenig Änderungen)

USE INDEX

↳ select /* index ("NAME") */

Tuning

Partitionierte Table

Rebuild Table inkl. Partitionierung

Speicher in pfile

shared pool size = füllungsgrad syst-tablespace

;)

dba - Studie unten used

22.06.04

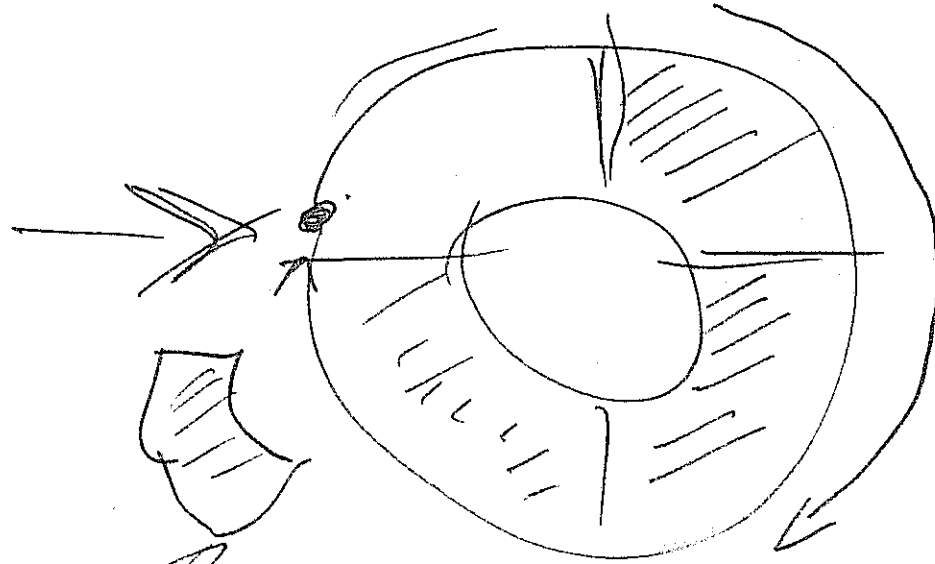
Oracle Summit

~~Tun: 7~~

Datafile im 2 GB - Bereich halbo-

oder 4 GB

RBA T_n ... T_m



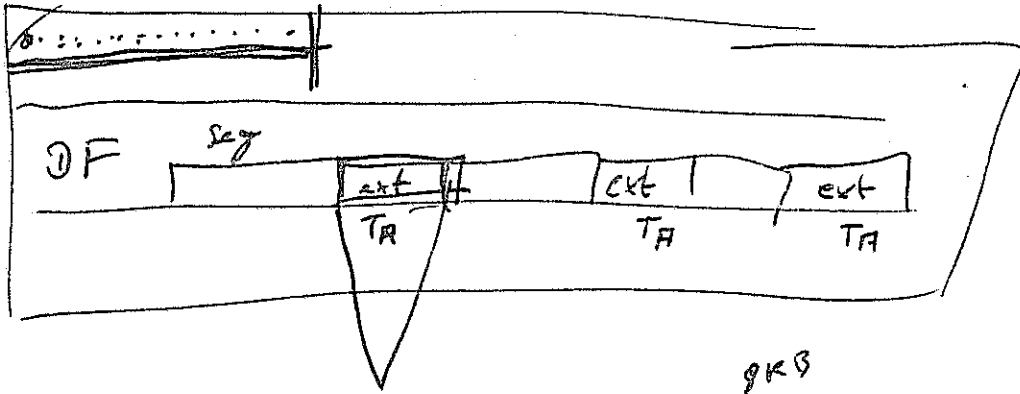
$$\frac{n}{4}$$

$$\frac{n}{8}$$

→ OLTP RB wide helix
RB-HUGE ↷ RB-DIG



TS



a) dict. ckt

b) loc. man. ckt.

n blöcke 4KB - 9KB

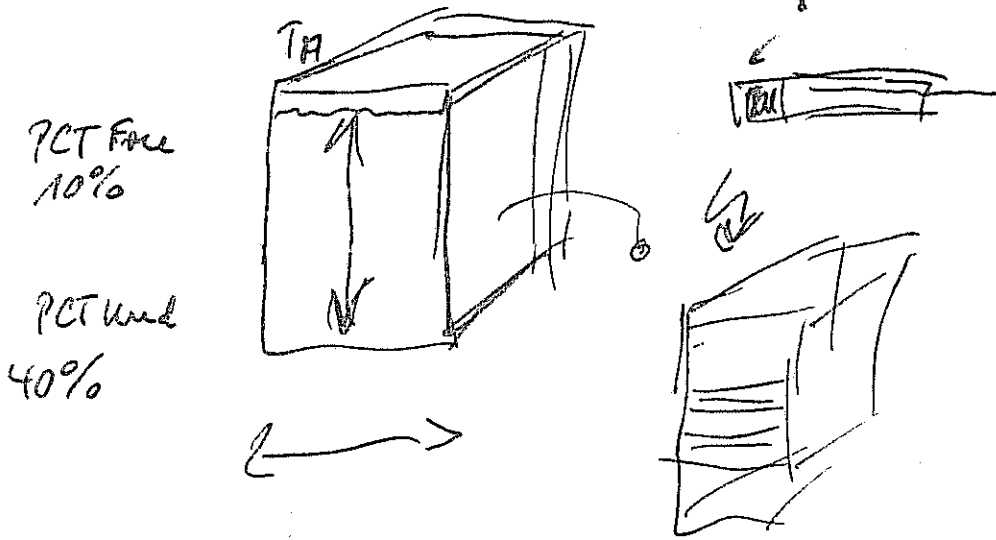


Table A 400 byte 10 5

Table X mit loc Datensatz

16 GB → df1 4GB off

→ df4 9GB

df5 - 100 MB auto ein

TOP: DFFSI / 24h Verfügbarkeit

- ~~alle~~ DB's aufw. KKBP laufen mit Archivierung
- Archivierung Pro/Contra
Anfall zeitraum, Wiederholungen
Recovery / Datenverlust
- Fehls bei Analyse-Ermittlung wg. Seiteneffekt des RB-Seg. Größe (Sup. too old)
analyze table (for all columns for all indexed)
compute vs. estimate bei großen Tabellen
- Storage: RB-Segmente Zuweisung RB
Transaktion ohne Redo Logging?
in Kombination mit archivieren führt
Wieder verwendbar / verschiebbar Daten-
Konsolidierung

Optimiere?

```
select /* index (...) */
```

→ Verwendung Bitmap Indexes

job pm Oracle - DB

stop

```
declare
v_jobnum := binary_integer;
Begin
  dbms_job.submit (v_jobnum, 'bseel-statistik.dus-statistik_falle',
  sysdate, 'sysdate + 1');

End;
```

1

```
select job, last_date from dba_jobs;
```

stop

```
execute dbms_job.remove (1)
                        ↑
                        job number
```

```
ins pfile [i_AFERET.ora]
```

```
↳ a Ende
```

```
job_queue_processes = 1
```

```
job_queue_interval = 1
```

} job - contacts Oracle
start

Aufruf Package

```
execute [Packagename].[Proceduriname]
```


~~start~~ [startup.bat]

lsnrctl start ← wird bei NT als Service dargestellt

svrmgrl < d:\oracle\... startup.sql

exit

[startup.sql]

connect internal @ dc-demo;

startup;

exit;

[shutdown.bat]

svrmgrl < d:\... shutdown.sql

lsnrctl stop

exit

[shutdown.sql]

shutdown immediate;

exit;

DR. DAHR CONSULTING GmbH

Instance = Prozesse + Arbeitsspeicher
(SGA)

↳ Redolog-Buffer
↓

Online Redolog

~~ARCH~~ ↓ ARCH

Offline Redolog

(= nur bei häufigen
Transaktion-
nicht in Datenreihen
und nicht bei Rows
bit i-part

Tabellen View

ab Vers 8.
→ auch updatable
aber nur durch
keys.

↳ constraints
↳ Trigger

Tablespace

- Temp
- system
- clustab
- ⋮
- clustab

arbeiten in Stamm
& Bewegungstabell

index
⋮
index

in einer separaten
Tablespace
damit keine
Fragmentierung

Extents = Blöcke

Storageparameter
initial
next - PCTINCREASE

PCTFREE
PCTUSED

DB

TBS

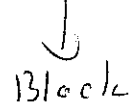
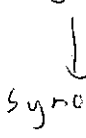
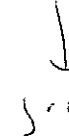
Files



Owner

Segments

Extents



grants
↳ system
↳ Objekt

synonym

Block

Rollen

sys → system

— system → Manager

— desc dba-users;

desc dba-tablespaces;

— temporary tablespace auch temporary einstellen ⇒ Performance schub

↳ alter tablespace-dba temporary;

↳ koin logs usw.

permanent;

desc dba-data-files;

① create user <username>

identified by <pw>

default tablespace <dba-name>

temporary tablespace <tem-tba>;

② grant connect, resource to <username>;

↳ Rollen

↳ system + Objektrechte

- <tab> → Objekt Rechte

→ desc dba-sys-privs; ← zeigt Rechte der Rolle an
system bzw Funktion

Eigene Rolle anlegen: - create role user-standard;

- grant create table to user-standard;

- revoke connect from user-demo; ← Rolle entfernen

DR. DAHR CONSULTING GmbH

- select * from ^{eigene} user-users ← zeigt akt. Logi-

↓

dba_ → Datenbanktabbe

↓

all_ →

- create public synonym ...

↑
gilt für alle user

⇒ Private synonyme haben Vorrang zu public synonyme

drop table ← löscht Tabelle in einer Schema

drop any table ← " " " alle "

select distinct privileges from dba_tab_privs;

⇒ Objekt privilegia

View

- create view <name> as
select spalten 1, 2, 3
from <table>
where

~~execute view~~

- desc dba-views;

show parameters

↳ zeigt alle

Einstellungen i. d. a.

Trigger

wenn Tabelle ^(drop Table) gelöscht wird ist Trigger ebenfalls
weg

⊙ wichtig: Trigger & Grant nach drop Table nachziehen

- desc dba-triggers;

```
create trigger <triggername>  
before insert on <tab.name>  
for each row  
begin  
:new.kfddname := sysdate; z.B.  
end;
```

Tablespaces

descr. dba-tablespaces;

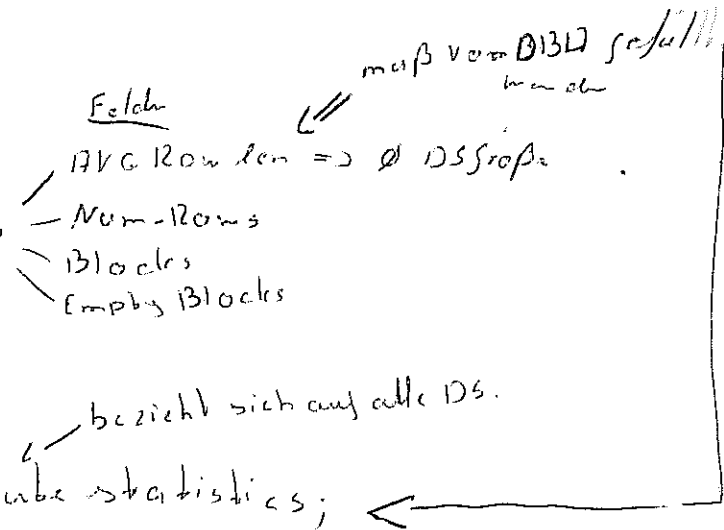
bei create table der Parameter Max extents auf 10 setzen

häufige update worktablespace

Im update dat tablespace anlegen

descr. dba-tables;

Tabellen → TBS → dba-tables
→ Extents



analyze table scott.emp compute statistics;

... estimate ...

← basiert nur auf Teilmenge DS

... delete ...

← löscht Werte wieder

Permanent => Rollbacksegmente werden off benutzt

Temporary => " " nicht "

DR. DAHR CONSULTING GmbH

disc dba_extents

```
select block_id, extent_id, blocks, segment_name from dba_extents
where tablespace_name = ' '
order by 1,2;
```

disc dba_free_space;

block_id blocks
 (zeigt an wieviel freie Blöcke man hat von Block-ID) an

```
select tablespace_name, count(*)
from dba_free_space
group by tablespace_name;
```

Anzahl der freien Extents

```
select tablespace_name, sum(bytes)
from dba_free_space
group by tablespace_name;
```

insgesamt Anzahl bytes in Tablespace

analyze table SCOTT.test compute statistics;

```
select segment_name, count(*)
from dba_extents
where owner = 'Scott'
group by segment_name
```

zwischen 30 und 100 muß Regry gefahren werden

Um changed rows auf \emptyset zurückkommen, muß --
das SQL

```
select * from table chained_rows(  
  owner_name varchar  
  \Oracle\... \adm-in\ utchaining.sql
```

dem analyze table (tablename) list chained_rows into
chained_rows

da-- create table c (copytablename) as select * from
c org.tabelle) where rowid in (select head_rowid
from chained_rows)

da-- delete from c org.tabelle) where rowid in (select
head_rowid from chained_rows);

dem insert into ~~org~~ c org tabs (select * from c temp tabs)
commit;

! temp anschließend löschen drop...

! Achtung bei Trigger & Constraints

Sanzes Schema analysieren:

```
execute dbms_utility.analyze_schema ('SCOTT', 'COMPUTE');
```

Tablespace anlegen:

! Wichtig ! Autotextent = off

```
create tablespace temp-work
```

```
datafile 'c:\... \work.dbf' size 10M;
```

oder mit
defaultstorage (initial 10k next 10k);

alter tablespace (Name)

coalesce

↑
defrag unben
Tablespaces

Kontrolle des Tablespaces

```
select initial_extens, next_extens, pct_incr from
```

```
dba_tablespaces where tablespace_name = 'Temp-work';
```

Benachbarte Blöcke zu einem Block zusammenfügen

```
=> alter tablespace temp-work coalesce;
```

Tablespace vergrößern oder verkleinern

```
alter tablespace temp-work
```

```
alter database datafile 'd:\... .dbf' resize 5M;
```

alle Files anschauen

```
dba_data_files
```

```
select file_name from
```

Rollback - Segmente

$\frac{\text{Anzahl der Transaktionen die laufen}}{4} = \text{Anzahl der Rollbacksegmente}$

desc dba_rollback_segs;

next_extent muß gleich initial_extent sein

Rollback Segment online oder offline schalten
=> alter rollback segment rb_temp online;

⚠ rb-temp schaltet man online wenn alle anderen RBS reorganisiert werden. Danach wieder offline schalten

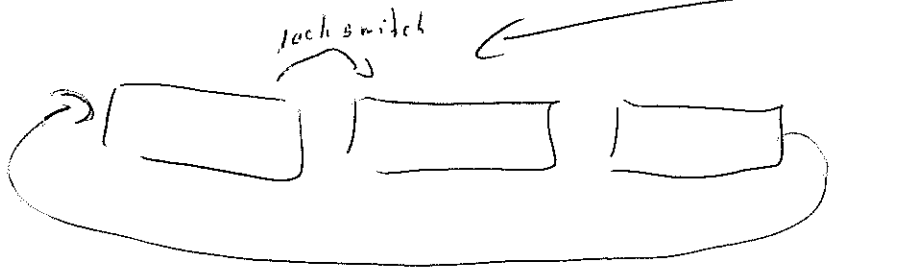
⚠ [Rollback segment reorganisieren bzw. erweitern]
1) online von Temp Segment, wenn kein existiert, dann
create rollback segment rb_temp tablespace system
storage (initial 100k next 100k)

2) create script schreiben mit

s. ⑧

- spool @r_vbs.sql
- select 'create ^{public} rollback segment '|| segment_name ||' storage
(initial size next size)'
- spool off
- alter rollback segment <segment-name> offline
- drop rollback segment <segment-name>
- creat Script aufrufen
- online schalten
- rb-temp offline schalten

online Redologs



alter system switch logfile
(um switchlog zu erzwingen)

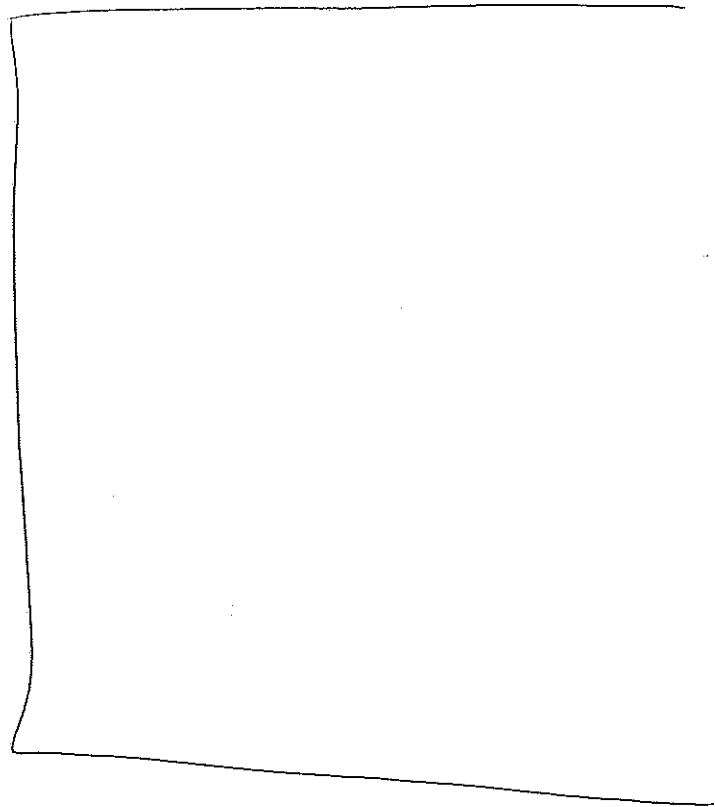
wenn lock switch erfolgt wird segment in Buffer geschrieben.

offline Redologs

online Redologs



offline Redolog



Online Redologs

desc v\$log

Online Redolog Gruppen

↳ Member

select group#, sequence#, bytes, members, archived, status
from v\$log;

mind. 2 Online RL aktiv & nicht alle löschen

member anlegen

→ alter datababase

add logfile member 'c:\temp\sa.dbf' to group 1;

drsc v\$logfiles; ⇒ zeigt alle Online Redologs pro Gruppe an

alter datababase drop logfile group 5;

Gruppe anlegen

alter datababase add logfile group 6 ('c:\...', 'c:\...')

- Trace file

↙ aus sqlplus

alter database backup controlfile to trace;

Datafile ← disc dba-data-files
umbenennen bzw. remove

∇ archive Modus muß
vorher laufen ∇

- Datafile offline schalten

wichtig

alter database datafile 'c:\...dbf' offline;

- Datafile kopieren an gewünschte Ort
(explek z.B)

(wenn bereitzustellen, database im Mount-Modus)

→ alter database rename file 'c:\...' to 'd:\...';

(alter tablespace <tblname> rename datafile 'c:\...' to 'd:\...';)

↑
(database im open Modus)

- Neues Datafile online schalten

- Altes " löschen

in Recovery-Fall

alter database recovery datafile '<name Datafile>'

Datenbank
starten & runbfahren im SVRMG12L

alter database open;
close;

startup nomount => ohne DataDictionary für create database
mount => mit " für archive einschalten
open => öffnen u. dann zu arbeiten

- archive einschalten (DB im Mount-Modus starten)

- archive log list - zeigt akt. Status an
- in Pfile archive auf true setzen
- alter database archive log;
- archive log start

aus:

→ false

alter database noarchive

archive log stop

- control file (Pfad ändern oder wenn ein defekt ist)

- shutdown
- ~~SSJ~~ in ~~Pfile~~ Pfile Pfad ändern oder hinzufügen
- controlfile kopieren
- startup

▽ Zeitstempel immer gleich von controlfile ▽
⊙

Data Recovery

- müssen
Sicherung
werden
- wissen wo Datafiles liegen (dba-dba-files)
 - ~~Controlfiles~~ Controlfiles (Pfile)
 - Online Redolog (~~dba~~ v\$logfile)
 - init.ora

Falls ein Tablespace defekt

- ~~von~~ Tablespace von Sicherung in Verzeichnis kopieren
- DIB in mount Zustand bringen
- alter databases recover automatic;

Recover mit archive-Files

- alter databases recover until [cancel] ^{'Datenzeit'} (S. 508)
- System schlägt archive-File vor
- dieses File von Sicherung zurückholen
- recover;
- ggf. alter databases recover automatic;

Import / Export

Export:

- Full export.dump
- USER
- Table

(S'4136

c:\exp cd

- fetch buffer size mind. auf 64000 setzen -> 90000 setzen
da kein Date
abgeschrieb. hoch

~~...~~ u. die akt. Stand zu exportieren (S' 4141)
die Parameter Consistent = Y setzen

Import

c:\imp usn/pm file = 'd:\...' from usn = SCOTT
to usn = demo buffer = 90000