ATENÇÃO: ORDEM DE IMPRESSAO
IMPRIMIR pg 2
IMPRIMIR pg 3 NO VERSO DA pg 2
IMPRIMIR pg 4
IMPRIMIR pg 5 NO VERSO DA pg 4

windows services
C:\> net stop oracleservice*SID* to stop windows oracle service
C:\> net start oracleservice*SID* to start windows oracle service

SQL Performance Tuning

hints
sintax: /*+ *hint* */
example: /*+ ORDERED FIRST_ROWS */

Hints for Optimization Aproaches and Goals
- ALL_ROWS
- FIRST_ROWS
- RULE
- CHOOSE

Hints for Access Methods
- FULL(*alias*)
- INDEX(*alias*,*index_name*)
- NO_INDEX(*alias*)

Hints for Join Order
- ORDERED — specify the tables order to access
- LEADING — specify the first main table to access

Hints for Join Operation
- USE_NL(*alias*)
- USE_HASH(*alias*)
- USE_MERGE(*alias*)

Other Hints
- NO_EXPAND — used to not expand OR predicates
- PUSH_SUBQ — force the subqueries to run first
- CURSOR_SHARING_EXACT
- CACHE
- NOCACHE

optimizer modes
alter session set optimizer_mode=rule;
alter session set optimizer_mode=choose;
alter session set optimizer_mode=all_rows;
alter session set optimizer_mode=first_rows;

explain plans
set autot on — shows plan and statistics using SQL*PLUS
set autot on exp — shows only the plan
set autot on stat — shows only the statistics
set autot trace — do not show the query's result-set

explain plan for *select … from …*
select operation, options, object_name, id, parent_id order by id;

trace in session
alter session set sql_trace = true;
alter session set sql_trace = false;
dbms_system.set_sql_trace_in_session *SID*;

tkprof
cd $ORACLE_BASE/admin/INST_NAME/udump/
tkprof sid_ora_*nnnn*.trc plan.txt explain = user/passw@host

consult views

index_stats	v$sqlarea
v$db_cache_advice	v$sql
v$sgastat	v$sql_plan
v$sesstat	
v$session_wait	
v$system_event	
v$sysstat	
v$waitstat	

ORACLE

Oracle9*i* Database

Simple and Quick Reference Guide

SQL and PL/SQL Programming
SQL*Plus Reference
Database Administration
SQL Performance Tuning

written by
Roverli P. Ziwich
Oracle Certified Professional
roverli@roverli.net
www.roverli.net

Revision 1.0 (Jun-2005)

SQL and PL/SQL Programming

types	char		
	varchar2	varchar2(100)	
	number	number(10)	number(16,4)
	date	timestamp	
	clob	blob	

STATEMENT SINTAX

if

```
if  (… and …) or (… and …) then
    commands
elsif (… or …) then
    commands
else
    commands
end if;
```

while

```
while (… and …) or (… and …) loop
    commands
end loop;
```

for

```
for variable in 1..999 loop
    commands
end loop;
```

cursor

```
for alias in (select field from table) loop
    v_variable := alias.field;
    commands
end loop;
```

OBJECT SINTAX

drop procedure *PROC_NAME*;
create or replace procedure *PROC_NAME*

```
(   … ,
    p_param   IN OUT table.field%TYPE := def_value
)
declare
    v_variable  type1 := default_value;
    err_num     number;
    err_msg     varchar2(100);
begin
    v_variable := 10;
    if condition then
        return;
    end if;
    …commands…
    raise_application_error( -20101, 'erro' );
    rollback;
exception
    when others then begin
        err_num := SQLCODE;
        err_msg := SUBSTR(SQLERRM, 1, 100);
    end;
end;
```

drop function *FUNC_NAME*;
create or replace function *FUNC_NAME*

```
(   … ,
    p_param   IN OUT table.field%TYPE := def_value
)
return type
is
    v_var   type := default_value;
begin
    …commands…
    return v_var;
end;
```


drop index *index_name*;
analize index *index_name* validate structure;

consult views

dba_tablespaces	dba_tables
dba_data_files	dba_tab_columns
dba_segments	dba_constraints
dba_extents	dba_cons_columns
dba_free_space	
dba_objects	v$tablespace
dba_indexes	v$datafile
dba_ind_columns	v$object_usabe (to index usage)

DATABASE MANAGEMENT

show parameter *name*;
shutdown [normal | transactional | immediate | abort];
startup [nomount | mount];
alter database open;

redo log / archive log / undo segments
archive log list;
alter system switch logfile;
alter system checkpoint;
alter system archive log [start / stop];
alter database [no]archivelog;

alter database add logfile group *n* ('/u02/…/log3a.rdo') size 5 M;
alter database drop logfile group *n*;
alter database add logfile member ('/u02/…/log3a.rdo') to group 1, …;
alter database drop logfile member '/u02/…/log3a.rdo';
alter database rename file '/u02/…/log3a.rdo' to '/u03/…/log3a.rdo';
alter database clear logfile group *n*;
alter database clear unarchived logfile group *n*;

parameter BACKGROUND_DUMP_DEST
parameter UNDO_SUPPRESS_ERRORS
alter system set UNDO_TABLESPACE = *undo_tbs_name*;
alter system set UNDO_RETENTION = *n_seconds*;

sga
show sga;

roles
parameter MAX_ENABLED_ROLES

consult views

dict / dictionary	v$fixed_table
database_properties	v$parameter
dba_temp_files	v$spparameter
	v$controlfile
v$database	v$controlfile_record_section
v$instance	v$session
v$sga	v$version
v$log	v$thread
v$logfile	v$undostat
v$loghist	v$rollname
v$archive	v$rollstat
v$archived_log	v$transaction

NETWORK

tnsping

listner
start / stop
status
service
status *listner_name*
show all
show trc_file

OBJECTS

tablespaces and datafiles

```
create tablaspace tbs_name datafile '/u01/…/data1.dbf size 5 M
        minimum extent 5 M
        extent management local uniform size 256 K;
        extent management dictionary default storage
                (initial 1 M next 1 M pctincrease 0);

alter tablespace add datafile '/u02/…/data02.dbf' size 40M;
alter tablespace
        [offline normal / offline immediate / online]
        [egin backup]
        [read only / read write];
alter tablespace tbs_name rename datafile '/u02/…/data01.dbf'
        to '/u03/…/data01.dbf';

alter database datafile '/u02/…/data01.dbf'
        [resize 10 MB]
        [autoextend on]
        [next 2M maxsize 200M];
alter database rename file '/u02/…/data01.dbf' to '/u03/…/data01.dbf';

drop tablespace tbs_name [including contents and datafiles];

create undo tablespace undo_tbs_name
        datafile '/u01/…/undo01.dbf' size 50 M;
create temporary tablespace temp_tbs_name
        tempfile '/u01/…/temp01.dbf' size 50 M;

alter database default temporary tablespace temp_tbs_name;

dbms_space_admin.tablespace_migrate_to_local('SYSTEM');
```

tables

```
drop table tab_name [cascade constraints];
truncate table tbs_name [cascade constraints];
create [global temporary] table tab_name
        ( filed1   type1 [NULL] default value, … )
        tablespace tbs_name
        [on commit preserve rows];

alter table tab_name
        rename to newtab_name;
        move tablespace newtbs_name;
        rename column col_name to newcol_name;
        add col_name type [NULL];
        modify column col_name newtype [NULL];
        drop column col_name;

alter tale tab_name drop primary key
alter tale tab_name add (constraint constr_name
        primary key (columns, …) )

alter tale tab_name drop constraint constr_name
alter tale tab_name add (constraint constr_name
        foreign key (columns, …) )
                references fk_tab_name (columns, …)
                [on delete cascade] );
        unique (columns, …);
        check (column in ('S', 'N') );
```

indexes

```
create [bitmap] index index_name
        on tab_name (columns, …)
        [storage (initial 200K next 200K maxextents 50)]
        [tablespace tbs_name];
alter index index_name
        coalesce;
        rebuild [online];
        [no]monitoring usage;
```



```
drop trigger TRIG_NAME;
create or replace trigger TRIG_NAME
    after insert or update or delete on table_name
    for each row
    declare
        v_var1      type1 := def_value;
        v_var2      type2 := default_value;
    begin
        if inserting then
            v_var1 := :new.table_field;
        elsif updating then
            v_var1 := :new.table_field;
            v_var2 := :old.table_field;
        elsif deleting then
            v_va2 := :old.table_field;
        end if;
    end;

drop view VIEW_NAME;
create or replace [force] view VIEW_NAME as subquery [with read only];

drop table TABLE_NAME;
create [global temporary] table TABLE_NAME
        ( filed1   type1 [NULL] default value, … )
        tablespace tbs_name
        [on commit preserve rows];

drop synonym SYNON_NAME;
create [public] synonym SYNON_NAME for schema.object_name;
```

OPERATORS

arithmetic operators
+ – * /

logical operators
not and or

comparison operators
< > <> != = <= >= [not] in any some all
[not] between x and y [not] exists [not] like is [not] null

set operators
UNION UNION ALL INTERSECT MINUS

string operators
|| concatenation

FUNCTIONS

numeric functions

ABS(n)	absolute value of n
CEIL(n)	smallest integer greater than or equal to n
FLOOR(n)	largest integer equal to or less than n
LOG(n,m)	logarithm, base m, of n
MOD(m,n)	remainder of m divided by n, returns m if n is 0
POWER(m,n)	m raised to the nth power
ROUND(n[,m])	n rounded to m places right of the decimal point
SQRT(n)	Returns square root of n
TRUNC(n[,m])	n truncated to m decimal places
TO_CHAR(d[,fmt])	converts n to VARCHAR2, using format fmt

string functions

ASCII(s)	decimal representation of the first byte of s
CHR(n)	character having the binary equivalent to n
CONCAT(s1,s2)	s1 concatenated with s2
INITCAP(s)	s with the first letter of each word in uppercase
INSTR (s1,s2[,n[,m]])	searches s1 beginning with n character for the mth occurrence of s2 and returns the position
LENGTH(s)	length of s in characters
LOWER(s)	s with all letters lowercase
LPAD(s1,n[,s2])	s1 left-padded to length n with s2
RPAD(s1,n[,s2])	s1 rigth-padded to length n with s2
LTRIM(char1,n)	removes characters from the left of char
RTRIM(s)	s with all the rightmost characters in set removed
REPLACE(s,search_str[,replace_str])	s with search_str's replaced with replac_str

SOUNDEX(s) character string that represents the phonetic of s
SUBSTR(s,m[,n]) a portion of char, beginning at m, n characters long
UPPER(s) s with all letters uppercase

date functions
ADD_MONTHS(d,n) date d plus n months
LAST_DAY(d) last day of the month that contains d
MONTHS_BETWEEN(d1, d2)
number of months between d1 and d2
ROUND(d[,fmt]) d rounded to the unit specified by fmt
SYSDATE current date and time
SYSTIMESTAMP current date and time in timestamp type
TRUNC(d,[fmt]) d with date truncated to the unit in fmt
TO_CHAR(d[,fmt]) converts d to VARCHAR2 using format fmt
TO_DATE(s[,fmt]) converts s to DATE using format fmt
TO_NUMBER(char[,fmt]) converts s to a NUMBER using format fmt

group functions
DISTINCT COUNT MAX MIN SUM AVG ALL

lob functions
DBMS_LOB.GETLENGTH(*table.lob_field*)

other functions
NVL(e1,e2) If *e1* is null, returns *e2*; if *e1* is not null, returns *e1*
USERENV('option')
option: 'TERMINAL' machine name
'SESSIONID' session id
'INSTANCE' instance name
SYS_CONTEXT('USERENV', 'IP_ADDRESS')
IP address of the connected machine

variables
UID an integer that uniquely identifies the current user
USER the current Oracle user
SQL%ROWCOUNT number of rows affected by a DML command

FORMATS

date/time formats
date type dd/mm/yyyy hh24:mi:ss
d day of weak (1-7)
ddd day of year (1-366)
SSSSS seconds past midnight (0-86399)
w week of month (1-5)
ww week of year (1-53)
ff*n* fraction of seconds (*n* digits)
timestamp type dd/mm/yyyy hh24:mi:ss:ff6

alter session set NLS_DATE_FORMAT = 'dd/mm/yyyy hh24:mi';
to_char(*value*, '99.999,99', 'NLS_NUMERIC_CHARACTERS='.,'')

OBJECT COMMANDS

object commands
disable alter trigger *trig_name* disable;
enable alter trigger *trig_name* enable;
compile alter procedure *proc_name* compile;
alter function *func_name* compile;
alter trigger *trig_name* compile;
alter view *view_name* compile;

dbms commands
dbms_output.put_line('message');
prints a message inside a PL/SQL block

SQL*Plus Reference

set options
set lines 999
set pages 999
set long 15728640
set heading off
set feedback off
set escape '\'
set verify off
set timing on
set serveroutput on

variable options
&var_name
&&var_name
undef var_name

commands
ed edit buffer
/ execute a command

Database Administration

USER

user
create user *user_name*
identified by *passw*
[default tablespace *table_space_name*]
[temporary tablespace *temp_tbspace_name*]
[quota unlimited on *tbspace_name*];

alter user *user_name*
[identified by *passw*]
[default tablespace *table_space_name*]
[temporary tablespace *temp_tbspace_name*]
[quota unlimited on *tbspace_name*];
alter user *user_name* default role all except *role_name*;

grant connect, resource, dba to *user_name*;

consult views
dba_users	all_users

GRANTS AND ROLES

grants
grant role grant *role_name* to *user_name*;
grant objects grant select,update on *table_name* to *user_name*;
grant insert,delete on *view_name* to *role_name*;
grant all on *table_name* to *user_name*;
grant execute on *proc_name* to *user_name*;
grant execute on *func_name* to *user_name*;

roles
create role create role *role_name*;
drop role drop role *role_name*;

revoke
revoke role revoke *role_name* from *user_name*;
revoke object revoke *privilege* on *object_name* from *user_name*;

consult views
dba_role_privs	dba_tab_privs