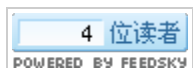


[空间](#)[博客](#)[好友](#)[相册](#)[留言](#)

用户操作

[\[留言\]](#) [\[发消息\]](#) [\[加为好友\]](#)

订阅我的博客



ilovemilk的公告

文章分类

-  [Delphi编程](#)
-  [JAVA技术](#)
-  [Unix/Linux/Bsd](#)
-  [Web开发](#)
-  [病毒讨论](#)
-  [感言](#)
-  [技术讨论](#)
-  [明日之星Python](#)
-  [软件开发](#)
-  [数据库技术](#)
-  [系统架构](#)

oracle dba常用SQL [收藏](#)

常用SQL查询:

1、查看表空间的名称及大小

```
select t.tablespace_name, round(sum(bytes/(1024*1024)),0) ts_size
from dba_tablespaces t, dba_data_files d
where t.tablespace_name = d.tablespace_name
group by t.tablespace_name;
```

2、查看表空间物理文件的名称及大小

```
select tablespace_name, file_id, file_name,
round(bytes/(1024*1024),0) total_space
from dba_data_files
order by tablespace_name;
```

3、查看回滚段名称及大小

```
select segment_name, tablespace_name, r.status,
(initial_extent/1024) InitialExtent,(next_extent/1024) NextExtent,
max_extents, v.curext CurExtent
From dba_rollback_segs r, v$rollstat v
Where r.segment_id = v.usn(+)
order by segment_name;
```

4、查看控制文件

```
select name from v$controlfile;
```

存档

2009年12月(10)
2009年11月(35)
2009年10月(50)
2009年09月(8)
2009年06月(1)
2009年05月(1)
2009年04月(2)
2009年03月(2)
2009年02月(1)
2009年01月(1)
2008年12月(4)
2008年11月(3)
2008年10月(2)
2008年09月(6)
2008年07月(6)
2008年05月(6)
2008年04月(3)
2008年02月(1)
2008年01月(1)
2007年11月(4)
2007年08月(4)
2007年05月(1)
2007年04月(2)
2007年03月(2)

5、查看日志文件

```
select member from v$logfile;
```

6、查看表空间的使用情况

```
select sum(bytes)/(1024*1024) as free_space,tablespace_name  
from dba_free_space  
group by tablespace_name;
```

```
SELECT A.TABLESPACE_NAME,A.BYTES TOTAL,B.BYTES USED, C.BYTES FREE,  
(B.BYTES*100)/A.BYTES "% USED",(C.BYTES*100)/A.BYTES "% FREE"  
FROM SYS.SM$TS_AVAIL A,SYS.SM$TS_USED B,SYS.SM$TS_FREE C  
WHERE A.TABLESPACE_NAME=B.TABLESPACE_NAME AND A.TABLESPACE_NAME=C.TABLESPACE_N  
AME;
```

7、查看数据库库对象

```
select owner, object_type, status, count(*) count# from all_objects group by owner, object_type,  
status;
```

8、查看数据库的版本

```
Select version FROM Product_component_version  
Where SUBSTR(PRODUCT,1,6)='Oracle';
```

9、查看数据库的创建日期和归档方式

```
Select Created, Log_Mode, Log_Mode From V$Database;
```

10、捕捉运行很久的SQL

```
column username format a12  
column opname format a16  
column progress format a8
```

```

select username,sid,opname,
       round(sofar*100 / totalwork,0) || '%' as progress,
       time_remaining,sql_text
from v$session_longops , v$sql
where time_remaining <> 0
and sql_address = address
and sql_hash_value = hash_value
/

```

11. 查看数据表的参数信息

```

SELECT  partition_name, high_value, high_value_length, tablespace_name,
        pct_free, pct_used, ini_trans, max_trans, initial_extent,
        next_extent, min_extent, max_extent, pct_increase, FREELISTS,
        freelist_groups, LOGGING, BUFFER_POOL, num_rows, blocks,
        empty_blocks, avg_space, chain_cnt, avg_row_len, sample_size,
        last_analyzed
FROM dba_tab_partitions
--WHERE table_name = :tname AND table_owner = :towner
ORDER BY partition_position

```

12. 查看还没提交的事务

```

select * from v$locked_object;
select * from v$transaction;

```

13. 查找object为哪些进程所用

```

select
p.spid,
s.sid,
s.serial# serial_num,
s.username user_name,
a.type object_type,
s.osuser os_user_name,
a.owner,
a.object object_name,
decode(sign(48 - command),
1,
to_char(command), 'Action Code #' || to_char(command) ) action,

```

```

p.program oracle_process,
s.terminal terminal,
s.program program,
s.status session_status
from v$session s, v$access a, v$process p
where s.paddr = p.addr and
      s.type = 'USER' and
      a.sid = s.sid and
      a.object='SUBSCRIBER_ATTR'
order by s.username, s.osuser

```

14. 回滚段查看

```

select rownum, sys.dba_rollback_segs.segment_name Name, v$rollstat.extents
Extents, v$rollstat.rssize Size_in_Bytes, v$rollstat.xacts XActs,
v$rollstat.gets Gets, v$rollstat.waits Waits, v$rollstat.writes Writes,
sys.dba_rollback_segs.status status from v$rollstat, sys.dba_rollback_segs,
v$rollname where v$rollname.name(+) = sys.dba_rollback_segs.segment_name and
v$rollstat.usn (+) = v$rollname.usn order by rownum

```

15. 耗资源的进程 (top session)

```

select s.schemaname schema_name, decode(sign(48 - command), 1,
to_char(command), 'Action Code #' || to_char(command) ) action, status
session_status, s.osuser os_user_name, s.sid, p.spid, s.serial# serial_num,
nvl(s.username, '[Oracle process]') user_name, s.terminal terminal,
s.program program, st.value criteria_value from v$sesstat st, v$session s, v$process p
where st.sid = s.sid and st.statistic# = to_number('38') and ('ALL' = 'ALL'
or s.status = 'ALL') and p.addr = s.paddr order by st.value desc, p.spid asc, s.username asc, s.os
user asc

```

16. 查看锁 (lock) 情况

```

select /*+ RULE */ ls.osuser os_user_name, ls.username user_name,
decode(ls.type, 'RW', 'Row wait enqueue lock', 'TM', 'DML enqueue lock', 'TX',
'Transaction enqueue lock', 'UL', 'User supplied lock') lock_type,
o.object_name object, decode(ls.lmode, 1, null, 2, 'Row Share', 3,
'Row Exclusive', 4, 'Share', 5, 'Share Row Exclusive', 6, 'Exclusive', null)
lock_mode, o.owner, ls.sid, ls.serial# serial_num, ls.id1, ls.id2
from sys.dba_objects o, ( select s.osuser, s.username, l.type,

```

```
l.lmode, s.sid, s.serial#, l.id1, l.id2 from v$session s,
v$lock l where s.sid = l.sid ) ls where o.object_id = ls.id1 and o.owner
<> 'SYS' order by o.owner, o.object_name
```

17. 查看等待 (wait) 情况

```
SELECT v$waitstat.class, v$waitstat.count count, SUM(v$sysstat.value) sum_value
FROM v$waitstat, v$sysstat WHERE v$sysstat.name IN ('db block gets',
'consistent gets') group by v$waitstat.class, v$waitstat.count
```

18. 查看sga情况

```
SELECT NAME, BYTES FROM SYS.V_$SGASTAT ORDER BY NAME ASC
```

19. 查看cached object

```
SELECT owner,          name,          db_link,          namespace,
       type,          sharable_mem,          loads,          executions,
       locks,          pins,          kept          FROM v$db_object_cache
```

20. 查看V\$SQLAREA

```
SELECT SQL_TEXT, SHARABLE_MEM, PERSISTENT_MEM, RUNTIME_MEM, SORTS,
VERSION_COUNT, LOADED_VERSIONS, OPEN_VERSIONS, USERS_OPENING, EXECUTIONS,
USERS_EXECUTING, LOADS, FIRST_LOAD_TIME, INVALIDATIONS, PARSE_CALLS, DISK_READS,
BUFFER_GETS, ROWS_PROCESSED FROM V$SQLAREA
```

21. 查看object分类数量

```
select decode (o.type#,1,'INDEX' , 2,'TABLE' , 3 , 'CLUSTER' , 4, 'VIEW' , 5 ,
'SYNONYM' , 6 , 'SEQUENCE' , 'OTHER' ) object_type , count(*) quantity from
sys.obj$ o where o.type# > 1 group by decode (o.type#,1,'INDEX' , 2,'TABLE' , 3
, 'CLUSTER' , 4, 'VIEW' , 5 , 'SYNONYM' , 6 , 'SEQUENCE' , 'OTHER' ) union select
'COLUMN' , count(*) from sys.col$ union select 'DB LINK' , count(*) from
```

22. 按用户查看object种类

```
select u.name schema, sum(decode(o.type#, 1, 1, NULL)) indexes,
sum(decode(o.type#, 2, 1, NULL)) tables, sum(decode(o.type#, 3, 1, NULL))
clusters, sum(decode(o.type#, 4, 1, NULL)) views, sum(decode(o.type#, 5, 1,
NULL)) synonyms, sum(decode(o.type#, 6, 1, NULL)) sequences,
sum(decode(o.type#, 1, NULL, 2, NULL, 3, NULL, 4, NULL, 5, NULL, 6, NULL, 1))
others from sys.obj$ o, sys.user$ u where o.type# >= 1 and u.user# =
```

```
o.owner# and u.name <> 'PUBLIC' group by u.name order by  
sys.link$ union select 'CONSTRAINT' , count(*) from sys.con$
```

23。有关connection的相关信息

1) 查看有哪些用户连接

```
select s.osuser os_user_name, decode(sign(48 - command), 1, to_char(command),  
'Action Code #' || to_char(command) ) action, p.program oracle_process,  
status session_status, s.terminal terminal, s.program program,  
s.username user_name, s.fixed_table_sequence activity_meter, '' query,  
0 memory, 0 max_memory, 0 cpu_usage, s.sid, s.serial# serial_num  
from v$session s, v$process p where s.paddr=p.addr and s.type = 'USER'  
order by s.username, s.osuser
```

2) 根据v.sid查看对应连接的资源占用等情况

```
select n.name,  
v.value,  
n.class,  
n.statistic#  
from v$statname n,  
v$sesstat v  
where v.sid = 71 and  
v.statistic# = n.statistic#  
order by n.class, n.statistic#
```

3) 根据sid查看对应连接正在运行的sql

```
select /*+ PUSH_SUBQ */  
command_type,  
sql_text,  
sharable_mem,  
persistent_mem,  
runtime_mem,  
sorts,  
version_count,  
loaded_versions,  
open_versions,  
users_opening,  
executions,  
users_executing,  
loads,
```

```

first_load_time,
invalidations,
parse_calls,
disk_reads,
buffer_gets,
rows_processed,
sysdate start_time,
sysdate finish_time,
'>' || address sql_address,
'N' status
from v$sqlarea
where address = (select sql_address from v$session where sid = 71)

```

24. 查询表空间使用情况

```

select a.tablespace_name,
       round(a.abytes / 1024 / 1024) totalmb,
       round((a.abytes - f.fbytes) / 1024 / 1024) usedmb,
       round(f.fbytes / 1024 / 1024) freemb,
       (1 - round(f.fbytes / a.abytes, 2)) * 100 "%USED",
       round(f.fbytes / a.abytes, 2) * 100 "%FREE"
  from (select tablespace_name, sum(bytes) abytes from dba_data_files group by tablespace_name) a,
       (select tablespace_name, sum(bytes) fbytes from dba_free_space group by tablespace_name) f
 where a.tablespace_name = f.tablespace_name

```

也可以这样（这个查询较快）：

```

select a.tablespace_name,
       round(a.abytes / 1024 / 1024) totalmb,
       round((a.abytes - f.fbytes) / 1024 / 1024) usedmb,
       round(f.fbytes / 1024 / 1024) freemb,
       (1 - round(f.fbytes / a.abytes, 2)) * 100 "%USED",
       round(f.fbytes / a.abytes, 2) * 100 "%FREE"
  from (select tablespace_name, sum(bytes) abytes from dba_data_files group by tablespace_name) a,
       (select tablespace_name, sum(bytes) fbytes from dba_free_space group by tablespace_name) f

```

```
where a.tablespace_name = f.tablespace_name
```

25. 查询表空间的碎片程度

```
select tablespace_name,count(tablespace_name) from dba_free_space group by tablespace_name  
having count(tablespace_name)>10;
```

```
alter tablespace name coalesce;  
alter table name deallocate unused;
```

```
create or replace view ts_blocks_v as  
select tablespace_name,block_id,bytes,blocks,'free space' segment_name from dba_free_space  
union all  
select tablespace_name,block_id,bytes,blocks,segment_name from dba_extents;
```

```
select * from ts_blocks_v;
```

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

26. 查询有哪些数据库实例在运行

```
select inst_name from v$active_instances;
```

27. 查询表占用空间大小

```
select owner,segment_name,bytes/1024/1024 kb from dba_extents where owner='SCOTT' and seg  
ment_name='EMP';
```

```
=====  
##### 创建数据库----look $ORACLE_HOME/rdbms/admin/buildall.sql #####
```

```
create database db01  
maxlogfiles 10  
maxdatafiles 1024  
maxinstances 2  
logfile  
GROUP 1 ('/u01/oradata/db01/log_01_db01.rdo') SIZE 15M,  
GROUP 2 ('/u01/oradata/db01/log_02_db01.rdo') SIZE 15M,
```



```
GROUP 3 ('/u01/oradata/db01/log_03_db01.rdo') SIZE 15M,  
datafile 'u01/oradata/db01/system_01_db01.dbf') SIZE 100M,  
undo tablespace UNDO  
datafile 'u01/oradata/db01/undo_01_db01.dbf' SIZE 40M  
default temporary tablespace TEMP  
tempfile 'u01/oradata/db01/temp_01_db01.dbf' SIZE 20M  
extent management local uniform size 128k  
character set AL32UTE8  
national character set AL16UTF16  
set time_zone='America/New_York';
```

```
##### 数据字典 #####
```

```
set wrap off
```

```
select * from v$dba_users;
```

```
grant select on table_name to user/rule;
```

```
select * from user_tables;
```

```
select * from all_tables;
```

```
select * from dba_tables;
```

```
revoke dba from user_name;
```

```
shutdown immediate
```

```
startup nomount
```

```
select * from v$instance;
```

```
select * from v$sga;
```

```
select * from v$tablespace;
```

```
alter session set nls_language=american;
```

```
alter database mount;
```

```
select * from v$database;
```

```
alter database open;

desc dictionary

select * from dict;

desc v$fixed_table;

select * from v$fixed_table;

set oracle_sid=foxconn

select * from dba_objects;

set serveroutput on

execute dbms_output.put_line('sfasd');

##### 控制文件 #####

select * from v$database;

select * from v$tablespace;

select * from v$logfile;

select * from v$log;

select * from v$backup;

/* 备份用户表空间 */

alter tablespace users begin backup;

select * from v$archived_log;

select * from v$controlfile;

alter system set control_files='$ORACLE_HOME/oradata/u01/ctrl01.ctl',

'$ORACLE_HOME/oradata/u01/ctrl02.ctl' scope=spfile;

cp $ORACLE_HOME/oradata/u01/ctrl01.ctl $ORACLE_HOME/oradata/u01/ctrl02.ctl

startup pfile='../initSID.ora'
```

```
select * from v$parameter where name like 'control%' ;

show parameter control;

select * from v$controlfile_record_section;

select * from v$tempfile;

/* 备份控制文件 */

alter database backup controlfile to '../filepath/control.bak';

/* 备份控制文件，并将二进制控制文件变为了 asc 的文本文件 */

alter database backup controlfile to trace;

##### redo log #####

archive log list;

alter system archive log start; -- 启动自动存档

alter system switch logfile; -- 强行进行一次日志 switch

alter system checkpoint; -- 强制进行一次 checkpoint

alter tablespace users begin backup;

alter tablespace offline;

/*checkpoint 同步频率参数 FAST_START_MTTR_TARGET, 同步频率越高，系统恢复所需时间越短 */

show parameter fast;

show parameter log_checkpoint;

/* 加入一个日志组 */

alter database add logfile group 3 ('/$ORACLE_HOME/oracle/ora_log_file6.rdo' size 10M);

/* 加入日志组的一个成员 */

alter database add logfile member '$ORACLE_HOME/oracle/ora_log_file6.rdo' to group 3;
```

/* 删除日志组：当前日志组不能删；活动的日志组不能删；非归档的日志组不能删 */

```
alter database drop logfile group 3;
```

/* 删除日志组中的某个成员，但每个组的最后一个成员不能被删除 */

```
alter databse drop logfile member '$ORACLE_HOME/oracle/ora_log_file6.rdo';
```

/* 清除在线日志 */

```
alter database clear logfile '$ORACLE_HOME/oracle/ora_log_file6.rdo';
```

```
alter database clear logfile group 3;
```

/* 清除非归档日志 */

```
alter database clear unarchived logfile group 3;
```

/* 重命名日志文件 */

```
alter database rename file '$ORACLE_HOME/oracle/ora_log_file6.rdo' to '$ORACLE_HOME/oracle/ora_log_file6a.rdo';
```

```
show parameter db_create;
```

```
alter system set db_create_online_log_dest_1='path_name';
```

```
select * from v$log;
```

```
select * from v$logfile;
```

/* 数据库归档模式到非归档模式的互换，要启动到 mount 状态下才能改变 ;startup mount; 然后再打开数据库 .*/

```
alter database noarchivelog/archivelog;
```

```
achive log start; --- 启动自动归档
```

```
alter system archive all; -- 手工归档所有日志文件
```

```
select * from v$archived_log;
```

```
show parameter log_archive;
```

分析日志文件 logmnr

1) 在 init.ora 中 set utl_file_dir 参数

2) 重新启动 oracle

3) create 目录文件

```
desc dbms_logmnr_d;
```

```
dbms_logmnr_d.build;
```

4) 加入日志文件 add/remove log file

```
dbms_logmnr.add_logfile
```

```
dbms_logmnr.removefile
```

5) start logmnr

```
dbms_logmnr.start_logmnr
```

6) 分析出来的内容查询 v\$logmnr_content --sqlredo/sqlundo

实践:

```
desc dbms_logmnr_d;
```

```
/* 对数据表做一些操作，为恢复操作做准备 */
```

```
update 表 set qty=10 where stor_id=6380;
```

```
delete 表 where stor_id=7066;
```

```
/******
```

utl_file_dir 的路径

```
execute dbms_logmnr_d.build('foxdict.ora','$ORACLE_HOME/oracle/admin/fox/cdump');
```

```
execute dbms_logmnr.add_logfile('$ORACLE_HOME/oracle/ora_log_file6.log',dbms_logmnr.newfile);
```

```
execute dbms_logmnr.start_logmnr(dictfilename=>'$ORACLE_HOME/oracle/admin/fox/cdump/foxdict.ora');
```

```
##### tablespace #####
```

```
select * from v$tablespace;
```

```
select * from v$datafile;
```

```
/* 表空间和数据文件的对应关系 */
```

```
select t1.name,t2.name from v$tablespace t1,v$datafile t2 where t1.ts#=t2.ts#;
```

```
alter tablespace users add datafile 'path' size 10M;
```

```
select * from dba_rollback_segs;
```

```
/* 限制用户在某表空间的使用限额 */
```

```
alter user user_name quota 10m on tablespace_name;
```

```
create tablespace xxx [datafile 'path_name/datafile_name'] [size xxx] [extent management local/dictionary] [default storage(xxx)];
```

```
exmple: create tablespace userdata datafile '$ORACLE_HOME/oradata/userdata01.dbf' size 100M AUTOEXTEND ON NEXT 5M MAXSIZE 200M;
```

```
create tablespace userdata datafile '$ORACLE_HOME/oradata/userdata01.dbf' size 100M extent management dictionary default storage(initial 100k next 100k pctincrease 10) offline;
```

```
/*9i 以后， oracle 建议使用 local 管理，而不使用 dictionary 管理，因为 local 采用 bitmap 管理表空间，不会产生系统表空间的自愿争用；*/
```

```
create tablespace userdata datafile '$ORACLE_HOME/oradata/userdata01.dbf' size 100M extent management local uniform size 1m;
```

```
create tablespace userdata datafile '$ORACLE_HOME/oradata/userdata01.dbf' size 100M extent management local autoallocate;
```

```
/* 在创建表空间时，设置表空间内的段空间管理模式，这里用的是自动管理 */
```

```
create tablespace userdata datafile '$ORACLE_HOME/oradata/userdata01.dbf' size 100M extent management local uniform size 1m segment space management auto;
```

```
alter tablespace userdata mininum extent 10;
```

```
alter tablespace userdata default storage(initial 1m next 1m pctincrease 20);

/*undo tablespace( 不能被用在字典管理模下 ) */

create undo tablespace undo1 datafile '$ORACLE_HOME/oradata/undo101.dbf' size 40M extent man
agement local;

show parameter undo;

/*temporary tablespace*/

create temporary tablespace userdata tempfile '$ORACLE_HOME/oradata/undo101.dbf' size 10m ex
tent management local;

/* 设置数据库缺省的临时表空间 */

alter database default temporary tablespace tablespace_name;

/* 系统 / 临时 / 在线的 undo 表空间不能被 offline*/

alter tablespace tablespace_name offline/online;

alter tablespace tablespace_name read only;

/* 重命名用户表空间 */

alter tablespace tablespace_name rename datafile '$ORACLE_HOME/oradata/undo101.dbf' to '$OR
ACLE_HOME/oradata/undo102.dbf';

/* 重命名系统表空间，但在重命名前必须将数据库 shutdown, 并重启到 mount 状态 */

alter database rename file '$ORACLE_HOME/oradata/system01.dbf' to '$ORACLE_HOME/oradata/sy
stem02.dbf';

drop tablespace userdata including contents and datafiles; ---drop tablespce

/*resize tablespace,autoextend datafile space*/

alter database datafile '$ORACLE_HOME/oradata/undo102.dbf' autoextend on next 10m maxsize 5
00M;

/*resize datafile*/

alter database datafile '$ORACLE_HOME/oradata/undo102.dbf' resize 50m;
```

```
/* 给表空间扩展空间 */

alter tablespace userdata add datafile '$ORACLE_HOME/oradata/undo102.dbf' size 10m;

/* 将表空间设置成 OMF 状态 */

alter system set db_create_file_dest='$ORACLE_HOME/oradata';

create tablespace userdata;---use OMF status to create tablespace;

drop tablespace userdata;---user OMF status to drop tablespace;

select * from dba_tablespace/v$tablespace/dba_data_files;

/* 将表的某分区移动到另一个表空间 */

alter table table_name move partition partition_name tablespace tablespace_name;

##### ORACLE storage structure and relationships #####

/* 手工分配表空间段的分区 (extend) 大小 */

alter table kong.test12 allocate extent(size 1m datafile '$ORACLE_HOME/oradata/undo102.dbf');

alter table kong.test12 deallocate unused; --- 释放表中没有用到的分区

show parameter db;

alter system set db_8k_cache_size=10m; --- 配置 8k 块的内存空间块参数

select * from dba_extents/dba_segments/data_tablespace;

select * from dba_free_space/dba_data_file/data_tablespace;

/* 数据对象所占用的字节数 */

select sum(bytes) from dba_extents where onwer='kong' and segment_name = 'table_name';

##### UNDO Data #####

show parameter undo;

alter tablespace users offline normal;
```



```

alter tablespace users offline immediate;

recover datafile '$ORACLE_HOME/oradata/undo102.dbf';

alter tablespace users online ;

select * from dba_rollback_segs;

alter system set undo_tablespace=undotbs1;

/* 忽略回滚段的错误提示 */

alter system set undo_suppress_errors=true;

/* 在自动管理模式下，不会真正建立 rbs1；在手工管理模式则可以建立，且是私有回滚段 */

create rollback segment rbs1 tablespace undotbs;

desc dbms_flashback;

/* 在提交了修改的数据后，9i 提供了旧数据的回闪操作，将修改前的数据只读给用户看，但这部分数据不会又恢复
在表中，而是旧数据的一个映射 */

execute dbms_flashback.enable_at_time('26-JAN- 04:12:17:00 pm ');

execute dbms_flashback.disable;

/* 回滚段的统计信息 */

select end_time,begin_time,undoblks from v$undostat;

/*undo 表空间的大小计算公式：UndoSpace=[ UR * (UPS * DBS)] + (DBS * 24)

UR :UNDO_RETENTION 保留的时间（秒）

UPS：每秒的回滚数据块

DBS: 系统 EXTENT 和 FILE SIZE( 也就是 db_block_size)*/

select * from dba_rollback_segs/v$rollname/v$rollstat/v$undostat/v$session/v$transaction;

show parameter transactions;

show parameter rollback;

```

/* 在手工管理模式下，建立公共的回滚段 */

```
create public rollback segment prbs1 tablespace undotbs;
```

```
alter rollback segment rbs1 online; ---- 在手工管理模式
```

/* 在手工管理模式中 ,initSID.ora 中指定 undo_management=manual 、 rollback_segment=('rbs1','rbs2',...) 、

transactions=100 、 transactions_per_rollback_segment=10

然后 shutdown immediate ,startup pfile=...\???.ora */

Managing Tables

/*char type maxlen=2000;varchar2 type maxlen=4000 bytes

rowid 是 18 位的 64 进制字符串 (10 个 bytes 80 bits)

rowid 组成：object#(对象号)--32bits,6 位

rfile#(相对文件号)--10bits,3 位

block#(块号)--22bits,6 位

row#(行号)--16bits,3 位

64 进制：A-Z,a-z,0-9,/,+ 共 64 个符号

dbms_rowid 包中的函数可以提供对 rowid 的解释 */

```
select rowid,dbms_rowid.rowid_block_number(rowid),dbms_rowid.rowid_row_number(rowid) from  
table_name;
```

```
create table test2
```

```
(
```

```
id int,
```

```
lname varchar2(20) not null,
```

```
fname varchar2(20) constraint ck_1 check(fname like 'k%'),
```

```
empdate date default sysdate)

) tablespace tablespace_name;


create global temporary table test2 on commit delete/preserve rows as select * from kong.authors
;

create table user.table(...) tablespace tablespace_name storage(...) pctfree10 pctused 40;

alter table user.tablename pctfree 20 pctused 50 storage(...);---changing table storage

/* 手工分配分区，分配的数据文件必须是表所在表空间内的数据文件 */

alter table user.table_name allocate extent(size 500k datafile '...');

/* 释放表中没有用到的空间 */

alter table table_name deallocate unused;

alter table table_name deallocate unused keep 8k;

/* 将非分区表的表空间搬到新的表空间，在移动表空间后，原表中的索引对象将会不可用，必须重建 */

alter table user.table_name move tablespace new_tablespace_name;

create index index_name on user.table_name(column_name) tablespace users;

alter index index_name rebuild;

drop table table_name [CASCADE CONSTRAINTS];

alter table user.table_name drop column col_name [CASCADE CONSTRAINTS CHECKPOINT 1000];-
--drop column

/* 给表中不用的列做标记 */

alter table user.table_name set unused column comments CASCADE CONSTRAINTS;

/*drop 表中不用的做了标记列 */

alter table user.table_name drop unused columns checkpoint 1000;
```

/* 当在 drop col 是出现异常，使用 CONTINUE ，防止重删前面的 column*/

ALTER TABLE USER.TABLE_NAME DROP COLUMNS CONTINUE CHECKPOINT 1000;

select * from dba_tables/dba_objects;

managing indexes

/*create index*/

example:

/* 创建一般索引 */

create index index_name on table_name(column_name) tablespace tablespace_name;

/* 创建位图索引 */

create bitmap index index_name on table_name(column_name1,column_name2) tablespace tablespace_name;

/* 索引中不能用 pctused*/

create [bitmap] index index_name on table_name(column_name) tablespace tablespace_name pctfree 20 storage(initial 100k next 100k) ;

/* 大数据量的索引最好不要做日志 */

create [bitmap] index index_name table_name(column_name1,column_name2) tablespace_name pctfree 20 storage(initial 100k next 100k) nologging;

/* 创建反转索引 */

create index index_name on table_name(column_name) reverse;

/* 创建函数索引 */

create index index_name on table_name(function_name(column_name)) tablespace tablespace_name;

/* 建表时创建约束条件 */

create table user.table_name(column_name number(7) constraint constraint_name primary key deferrable using index storage(initial 100k next 100k) tablespace tablespace_name,column_name2 va

```
rchar2(25) constraint constraint_name not null,column_name3 number(7)) tablespace tablespace_
name;

/* 给创建 bitmap index 分配的内存空间参数，以加速建索引 */

show parameter create_bit;

/* 改变索引的存储参数 */

alter index index_name pctfree 30 storage(initial 200k next 200k);

/* 给索引手工分配一个分区 */

alter index index_name allocate extent (size 200k datafile '$ORACLE/oradata/..');

/* 释放索引中没用的空间 */

alter index index_name deallocate unused;

/* 索引重建 */

alter index index_name rebuild tablespace tablespace_name;

/* 普通索引和反转索引的互换 */

alter index index_name rebuild tablespace tablespace_name reverse;

/* 重建索引时，不锁表 */

alter index index_name rebuild online;

/* 给索引整理碎片 */

alter index index_name COALESCE;

/* 分析索引，事实上是更新统计的过程 */

analyze index index_name validate structure;

desc index_state;

drop index index_name;

alter index index_name monitoring usage;----- 监视索引是否被用到
```

```
alter index index_name nomonitoring usage; ---- 取消监视
```

```
/* 有关索引信息的视图 */
```

```
select * from dba_indexes/dba_ind_columns/dbs_ind_expressions/v$object_usage;
```

```
##### 数据完整性的管理 (Maintaining data integrity) #####
```

```
alter table table_name drop constraint constraint_name; ----drop 约束
```

```
alter table table_name add constraint constraint_name primary key(column_name1,column_name2  
); ----- 创建主键
```

```
alter table table_name add constraint constraint_name unique(column_name1,column_name2); ---  
创建唯一约束
```

```
/* 创建外键约束 */
```

```
alter table table_name add constraint constraint_name foreign key(column_name1) references tabl  
e_name(column_name1);
```

```
/* 不校验老数据，只约束新的数据 [enable/disable : 约束 / 不约束新数据 ;novalidate/validate: 不对 / 对老  
数据进行验证 ]*/
```

```
alter table table_name add constraint constraint_name check(column_name like 'B%') enable/disab  
le novalidate/validate;
```

```
/* 修改约束条件，延时验证， commit 时验证 */
```

```
alter table table_name modify constraint constraint_name initially deferred;
```

```
/* 修改约束条件，立即验证 */
```

```
alter table table_name modify constraint constraint_name initially immediate;
```

```
alter session set constraints=deferred/immediate;
```

```
/*drop 一个有外键的主键表，带 cascade constraints 参数级联删除 */
```

```
drop table table_name cascade constraints;
```

```
/* 当 truncate 外键表时，先将外键设为无效，再 truncate;*/
```

```

truncate table table_name;

/* 设约束条件无效 */

alter table table_name disable constraint constraint_name;

alter table table_name enable novalidate constraint constraint_name;

/* 将无效约束的数据行放入 exception 的表中，此表记录了违反数据约束的行的行号；在此之前，要先建 exceptions 表 */

alter table table_name add constraint constraint_name check(column_name >15) enable validate exceptions into exceptions;

/* 运行创建 exceptions 表的脚本 */

start $ORACLE_HOME/rdbms/admin/utlexcpt.sql;

/* 获取约束条件信息的表或视图 */

select * from user_constraints/dba_constraints/dba_cons_columns;

##### managing password security and resources #####
####

alter user user_name account unlock/open;---- 锁定 / 打开用户 ;

alter user user_name password expire;--- 设定口令到期

/* 建立口令配置文件 ,failed_login_attempts 口令输多少次后锁， password_lock_times 指多少天后口令被自动解锁 */

create profile profile_name limit failed_login_attempts 3 password_lock_times 1/1440;

/* 创建口令配置文件 */

create profile profile_name limit failed_login_attempts 3 password_lock_time unlimited password_life_time 30 password_reuse_time 30 password_verify_function verify_function password_grace_time 5;

/* 建立资源配置文件 */

create profile prfile_name limit session_per_user 2 cpu_per_session 10000 idle_time 60 connect_ti

```

me 480;

alter user user_name profile profile_name;

/* 设置口令解锁时间 */

alter profile profile_name limit password_lock_time 1/24;

/*password_life_time 指口令文件多少时间到期, password_grace_time 指在第一次成功登录后到口令到期有多少天时间可改变口令 */

alter profile profile_name limit password_lift_time 2 password_grace_time 3;

/*password_reuse_time 指口令在多少天内可被重用 ,password_reuse_max 口令可被重用的最大次数 */

alter profile profile_name limit password_reuse_time 10[password_reuse_max 3];

alter user user_name identified by input_password; ----- 修改用户口令

drop profile profile_name;

/* 建立了 profile 后, 且指定给某个用户, 则必须用 CASCADE 才能删除 */

drop profile profile_name CASCADE;

alter system set resource_limit=true; --- 启用自愿限制, 缺省是 false

/* 配置资源参数 */

alter profile profile_name limit cpu_per_session 10000 connect_time 60 idle_time 5;

/* 资源参数 (session 级)

cpu_per_session 每个 session 占用 cpu 的时间 单位 1/100 秒

sessions_per_user 允许每个用户的并行 session 数

connect_time 允许连接的时间 单位分钟

idle_time 连接被空闲多少时间后, 被自动断开 单位分钟

logical_reads_per_session 读块数

private_sga 用户能够在 SGA 中使用的私有的空间数 单位 bytes

(call 级)

cpu_per_call 每次 (1/100 秒) 调用 cpu 的时间

logical_reads_per_call 每次调用能够读的块数

*/

```
alter profile profile_name limit cpu_per_call 1000 logical_reads_per_call 10;
```

```
desc dbms_resouce_manager; --- 资源管理器包
```

```
/* 获取资源信息的表或视图 */
```

```
select * from dba_users/dba_profiles;
```

```
##### Managing users #####
```

```
show parameter os;
```

```
create user testuser1 identified by kxf_001;
```

```
grant connect,createtable to testuser1;
```

```
alter user testuser1 quota 10m on tablespace_name;
```

```
/* 创建用户 */
```

```
create user user_name identified by password default tablespace tablespace_name temporary tablespace tablespace_name quota 15m on tablespace_name password expire;
```

```
/* 数据库级设定缺省临时表空间 */
```

```
alter database default temporary tablespace tablespace_name;
```

```
/* 制定数据库级的缺省表空间 */
```

```
alter database default tablespace tablespace_name;
```

```
/* 创建 os 级审核的用户，需知道 os_authent_prefix，表示 oracle 和 os 口令对应的前缀，'OPS$' 为此参数的值，此值可以任意设置 */
```

```
create user user_name identified by externally default OPS$tablespace_name tablespace_name temporary tablespace tablespace_name quota 15m on tablespace_name password expire;
```

```

/* 修改用户使用表空间的限额，回滚表空间和临时表空间不允许授予限额 */
alter user user_name quota 5m on tablespace_name;

/* 删除用户或删除级联用户（用户对象下有对象的要使用 CASCADE，将其下一些对象一起删除）*/
drop user user_name [CASCADE];

/* 每个用户在哪些表空间下有什么限额 */

desc dba_ts_quotas;select * from dba_ts_quotas where username='...';

/* 改变用户的缺省表空间 */

alter user user_name default tablespace tablespace_name;

##### Managing Privileges #####

grant create table,create session to user_name;

grant create any table to user_name; revoke create any table from user_name;

/* 授予权限语法 ,public 标识所有用户 ,with admin option 允许能将权限授予第三者的权限 */
grant system_privs,[.....] to [user/role/public],[...] [with admin option];

select * from v$pwfile_users;

/* 当 O7_dictionary_accessibility 参数为 True 时，标识 select any table 时，包括系统表也能 select，否则，不包含系统表；缺省为 false*/

show parameter O7;

/* 由于 O7_dictionary_accessibility 为静态参数，不能动态改变，故加 scope=spfile,下次启动时才生效 */
alter system set O7_dictionary_accessibility=true scope=spfile;

/* 授予对象中的某些字段的权限，如 select 某表中的某些字段的权限 */

grant [object_privs(columnn,...)],[...] on object_name to user/role/public,... with grant option;

/*oracle 不允许授予 select 某列的权限，但可以授 insert ,update 某列的权限 */

grant insert(column_name1,column_name2,...) on table_name to user_name with grant option;

```

```

select * from dba_sys_privs/session_privs/dba_tab_privs/user_tab_privs/dba_col_privs/user_col_privs;

/*db/os/none 审计被记录在 数据库 / 操作系统 / 不审计 缺省是 none*/

show parameter audit_trail;

/* 启动对表的 select 动作 */

audit select on user.table_name by session;

/*by session 在每个 session 中发出 command 只记录一次， by access 则每个 command 都记录 */

audit [create table][select/update/insert on object by session/access][whenever successful/not successful];

desc dbms_fga; --- 进一步设计，则可使用 dbms_fgs 包

/* 取消审计 */

noaudit select on user.table_name;

/* 查被审计信息 */

select * from all_def_audit_opts/dba_stmt_audit_opts/dba_priv_audit_opts/dba_obj_audit_opts;

/* 获取审计记录 */

select * from dba_audit_trail/dba_audit_exists/dba_audit_object/dba_audit_session/dba_audit_statement;

##### Managing Role #####

create role role_name; grant select on table_name to role_name; grant role_name to user_name;
set role role_name;

create role role_name;

create role role_name identified by password;

create role role_name identified externally;

set role role_name ; ---- 激活 role

```

```
set role role_name identified by password;

alter role role_name not identified;

alter role role_name identified by password;

alter role role_name identified externally;

grant priv_name to role_name [WITH ADMIN OPTION];

grant update(column_name1,col_name2,...) on table_name to role_name;

grant role_name1 to role_name2;

/* 建立 default role, 用户登录时, 缺省激活 default role*/

alter user user_name default role role_name1,role_name2,...;

alter user user_name default role all;

alter user user_name default role all except role_name1,...;

alter user user_name default role none;

set role role1 [identified by password],role2,...;

set role all;

set role except role1,role2,...;

set role none;

revoke role_name from user_name;

revoke role_name from public;

drop role role_name;

select * from dba_roles/dba_role_privs/role_role_privs/dba_sys_privs/role_sys_privs/role_tab_privs/session_roles;

##### Basic SQL SELECT #####

select col_name as col_alias from table_name ;
```

```
select col_name from table_name where col1 like '_o%'; ---- '_' 匹配单个字符
```

```
/* 使用字符函数 ( 右边截取, 字段中包含某个字符, 左边填充某字符到固定位数, 右边填充某字符到固定位数 ) *  
/
```

```
select substr(col1,-3,5),instr(col2,'g'),LPAD(col3,10,'$'),RPAD(col4,10,'%') from table_name;
```

```
/* 使用数字函数 ( 往右 / 左几位四舍五入, 取整, 取余 ) */
```

```
select round(col1,-2),trunc(col2),mod(col3) from table_name ;
```

```
/* 使用日期函数 ( 计算两个日期间相差几个星期, 两个日期间相隔几个月, 在某个月份上加几个月, 某个日期的  
下一个日期,
```

```
某日期所在月的最后的日期, 对某个日期的月分四舍五入, 对某个日期的月份进行取整 ) */
```

```
select (sysdate-col1)/7 week,months_between(sysdate,col1),add_months(col1,2),next_day(sysdate  
, 'FRIDAY'),last_day(sysdate),
```

```
round(sysdate,'MONTH'),trunc(sysdate,'MONTH') from table_name;
```

```
/* 使用 NULL 函数 ( 当 expr1 为空取 expr2/ 当 expr1 不为空取 expr3/ 当 expr1=expr2 返回  
空 ) */
```

```
select nvl(expr1,expr2),nvl2(expr1,expr2,expr3),nullif(expr1,expr2) from table_name;
```

```
select column1,column2,column3, case column2 when '50' then column2*1.1
```

```
when '30' then column2*2.1
```

```
when '10' then column3/20
```

```
else column3
```

```
end as ttt
```

```
from table_name ; ----- 使用 case 函数
```

```
select table1.col1,table2.col2 from table1
```

```
[CROSS JOIN table2] | ----- 笛卡儿连接
```

```
[NATURAL JOIN table2] | ----- 用两个表中的同名列连接
```

[JOIN table2 USING (column_name)] | ----- 用两个表中的同名列中的某一列或几列连接

[JOIN table2

ON (table1.col1=table2.col2)] |

[LEFT|RIGHT|FULL OUTER JOIN table2 ----- 相当于 (+)=,=(+) 连接，全外连接

ON (table1.col1=table2.col2)]; -----SQL 1999 中的 JOIN 语法；

example:

```
select col1,col2 from table1 t1
```

```
join table2 t2
```

```
on t1.col1=t2.col2 and t1.col3=t2.col1
```

```
join table3 t3
```

```
on t2.col1=t3.col3;
```

```
select * from table_name where col1 < any (select col2 from table_name2 where continue group by col3);
```

```
select * from table_name where col1 < all (select col2 from table_name2 where continue group by col3);
```

```
insert into (select col1,col2,col3 form table_name where col1> 50 with check option) values (value 1,value2,value3);
```

```
MERGE INTO table_name table1
```

```
USING table_name2 table2
```

```
ON (table1.col1=table2.col2)
```

```
WHEN MATCHED THEN
```

```
UPDATE SET
```

```
table1.col1=table2.col2,
```

```
table1.col2=table2.col3,
```

...

WHEN NOT MATCHED THEN

INSERT VALUES(table2.col1,table2.col2,table2.col3,...); ----- 合并语句

CREATE/ALTER TABLE

alter table table_name drop column column_name ; ---drop column

alter table table_name set unused (col1,col2,...); ---- 设置列无效，这个比较快。

alter table table_name drop unused columns; --- 删除被设为无效的列

rename table_name1 to table_name2; --- 重命名表

comment on table table_name is 'comment message'; ---- 给表放入注释信息

create table table_name

(col1 int not null,col2 varchar2(20),col3 varchar2(20),

constraint uk_test2_1 unique(col2,col3)); ----- 定义表中的约束条件

alter table table_name add constraint pk_test2 primary key(col1,col2,...); ---- 创建主键

/* 建立外键 */

create table table_name (rid int,name varchar2(20),constraint fk_test3 foreign key(rid) references other_table_name(id));

alter table table_name add constraint ck_test3 check(name like 'K%');

alter table table_name drop constraint constraint_name;

alter table table_name drop primary key cascade; ---- 级联删除主键

alter table table_name disable/enable constraint constraint_name; ---- 使约束暂时无效

/* 删除列，并级联删除此列下的约束条件 */

alter table table_name drop column column_name cascade constraint;

select * from user_constraints/user_cons_columns; --- 约束条件相关视图

Create Views

CREATE [OR REPLACE] [FORCE|NOFORCE] VIEW view_name [(alias[,alias]...)]

AS subquery

[WITH CHECK OPTION [CONSTRAINT constraint_name]]

[WITH READ ONLY [CONSTRAINT constraint_name]]; ----- 创建视图的语法

example: Create or replace view testview as select col1,col2,col3 from table_name; ----- 创建视图

/* 使用别名 */

Create or replace view testview as select col1,sum(col2) col2_alias from table_name;

/* 创建复杂视图 */

Create view view_name (alias1,alias2,alias3,alias4) as select d.col1,min(e.col1),max(e.col1),avg(e.col1) from table_name1 e,table_name2 d where e.col2=d.col2 group by d.col1;

/* 当用 update 修改数据时，必须满足视图的 col1>10 的条件，不满足则不能被改变 */

Create or replace view view_name as select * from table_name where col1>10 with check option;

/* 改变视图的值 . 对于简单视图可以用 update 语法修改表数据，但复杂视图则不一定能改。如使用了函数，group by ,distinct 等的列 */

update view_name set col1=value1;

/*TOP-N 分析 */

select [column_list],rownum from (select [column_list] from table_name order by Top-N_column)
where rownum<=N;

/* 找出某列三条最大值的记录 */

example: select rownum as rank ,col1 ,col2 from (select col1 ,col2 from table_name order by col2 desc) where rownum<=3;

Other database Object

CREATE SEQUENCE sequence_name [INCREMENT BY n]

[START WITH n]

[{MAXVALUE n | NOMAXVALUE}]

[{MINVALUE n | NOMINVALUE}]

[{CYCEL | NOCYCLE}]

[{CACHE n | NOCACHE}]; ----- 创建 SEQUENCE

example:

CREATE SEQUENCE sequence_name INCREMENT BY 10

START WITH 120

MAXVALUE 9999

NOCACHE

NOCYCLE;

select * from user_sequences ; --- 当前用户下记录 sequence 的视图

select sequence_name.nextval,sequence_name.currval from dual; -----sequence 的引用

alter sequence sequence_name INCREMENT BY 20

MAXVALUE 999999

NOCACHE

NOCYCLE; ----- 修改 sequence, 不能改变起始序号

drop sequence sequence_name; ---- 删除 sequence

CREATE [PUBLIC] SYNONYM synonym_name FOR object; ----- 创建同义词

DROP [PUBLIC] SYNONYM synonym_name; ---- 删除同义词

CREATE PUBLIC DATABASE LINK link_name USING OBJECT; ---- 创建 DBLINK

select * from object_name@link_name; ---- 访问远程数据库中的对象

/*union 操作，它将两个集合的交集部分压缩，并对数据排序 */

```
select col1,col2,col3 from table1_name union select col1,col2,col3 from table2_name;
```

/*union all 操作，两个集合的交集部分不压缩，且不对数据排序 */

```
select col1,col2,col3 from table1_name union all select col1,col2,col3 from table2_name;
```

/*intersect 操作，求两个集合的交集，它将对重复数据进行压缩，且排序 */

```
select col1,col2,col3 from table1_name intersect select col1,col2,col3 from table2_name;
```

/*minus 操作，集合减，它将压缩两个集合减后的重复记录，且对数据排序 */

```
select col1,col2,col3 from table1_name minus select col1,col2,col3 from table2_name;
```

/*EXTRACT 抽取时间函数．此例是抽取当前日期中的年 */

```
select EXTRACT(YEAR FROM SYSDATE) from dual;
```

/*EXTRACT 抽取时间函数．此例是抽取当前日期中的月 */

```
select EXTRACT(MONTH FROM SYSDATE) from dual;
```

增强的 group by 子句 #####
###

```
select [column,] group_function(column)...
```

```
from table
```

```
[WHERE condition]
```

```
[GROUP BY [ROLLUP] group_by_expression]
```

```
[HAVING having_expression];
```

```
[ORDER BY column]; -----ROLLUP 操作字，对 group by 子句的各字段从右到左进行再聚合
```

example:

/* 其结果看起来象对 col1 做小计 */

```
select col1,col2,sum(col3) from table group by rollup(col1,col2);
```

/* 复合 rollup 表达式 */

```
select col1,col2,sum(col3) from table group by rollup((col1,col2));
```

```
select [column,] group_function(column)...
```

```
from table
```

```
[WHERE condition]
```

```
[GROUP BY [CUBE] group_by_expression]
```

```
[HAVING having_expression];
```

[ORDER BY column]; -----CUBE 操作字，除完成 ROLLUP 的功能外，再对 ROLLUP 后的结果集从右到左再聚合

example:

/* 其结果看起来象对 col1 做小计后，再对 col2 做小计，最后算总计 */

```
select col1,col2,sum(col3) from table group by cube(col1,col2);
```

/* 复合 rollup 表达式 */

```
select col1,col2,sum(col3) from table group by cube((col1,col2));
```

/* 混合 rollup,cube 表达式 */

```
select col1,col2,col3,sum(col4) from table group by col1,rollup(col2),cube(col3);
```

/*GROUPING(expr) 函数，查看 select 语句中以何字段聚合，其取值为 0 或 1*/

```
select [column,] group_function(column)...,GROUPING(expr)
```

```
from table
```

```
[WHERE condition]
```

```
[GROUP BY [ROLLUP] group_by_expression]
```

```
[HAVING having_expression];
```

```
[ORDER BY column];
```

example:

```
select col1,col2,sum(col3),grouping(col1),grouping(col2) from table group by cube(col1,col2);
```

/*grouping sets 操作，对 group by 结果集先对 col1 求和，再对 col2 求和，最后将其结果集并在一起 */

```
select col1,col2,sum(col3) from table group by grouping sets((col1),(col2));
```

发表于 @ 2008年07月17日 14:39:00 | [评论\(0\)](#) | [举报](#) | [收藏](#)

[旧一篇: 在FreeBSD7.0上构建基于postfix的邮件系统\(第一部分\)](#) | [新一篇: oracle分析函数](#)

发表评论 “评论王”评选开始啦，评论越多好礼越多！

表情:



评论内容:

用户名: huapuyu3

发表评论