第一步：查看Oracle表空间的使用情况：

select dbf.tablespace\_name,

dbf.totalspace "总量(M)",

dbf.totalblocks as 总块数,

dfs.freespace "剩余总量(M)",

dfs.freeblocks "剩余块数",

(dfs.freespace / dbf.totalspace) \* 100 "空闲比例"

from (select t.tablespace\_name,

sum(t.bytes) / 1024 / 1024 totalspace,

sum(t.blocks) totalblocks

from dba\_data\_files t

group by t.tablespace\_name) dbf,

(select tt.tablespace\_name,

sum(tt.bytes) / 1024 / 1024 freespace,

sum(tt.blocks) freeblocks

from dba\_free\_space tt

group by tt.tablespace\_name) dfs

where trim(dbf.tablespace\_name) = trim(dfs.tablespace\_name)

第二步：找到需要扩展空间的表空间，查看其数据文件路径：

select \* from dba\_data\_files t where t.tablespace\_name = '表空间名称'

第三步：增加表空间大小：

增加表空间大小的方法有二：

1、修改数据文件的大小：

alter database datafile '全路径的数据文件名称' resize \*\*\*M

2、新增数据文件：

alter tablespace 表空间名称

add datafile '全路径的数据文件名称' size \*\*\*M

That's all .

注意：1、表空间尽量让free百分比保持在10%以上，如果低于10%就增加datafile或者resizedatafile，一般数据文件不要超过2G

2、 设置表空间数据文件自动扩展：

alter database datafile '全路径的数据文件名称' autoextend on;

第二种

ORACLE 表空间扩展方法

环境: linux系统 工具：PL/SQL DEVELOPER

第一步：查看表空间的名字及文件所在位置：

select tablespace\_name,

file\_id,

file\_name,

round(bytes / (1024 \* 1024), 0) total\_space

from sys.dba\_data\_files

order by tablespace\_name

第二步：增大所需表空间大小：

alter database datafile '表空间位置' resize 新的尺寸

例如：

alter database datafile '\oracle\oradata\anita\_2008.dbf' resize 4000m

对于oracle数据库的表空间，除了用手动增加大小外，还可以增加数据文件等方式扩展表空间大小。

方法一：增加数据文件个数

alter tablespace 表空间名称 add datafile '新的数据文件地址' size 数据文件大小

例如：

alter tablespace ESPS\_2008 add datafile '\oracle\oradata\anita\_2010.dbf' size 1000m

方法二：设置表空间自动扩展。

alter database datafile '数据文件位置' autoextend on next 自动扩展大小 maxsize 最大扩展大小

例如：

alter database datafile '\oracle\oradata\anita\_2008.dbf' autoextend on next 100m maxsize 10000m

方法三：查询表空间使用情况：

复制代码

select a.tablespace\_name,

a.bytes / 1024 / 1024 "sum MB",

(a.bytes - b.bytes) / 1024 / 1024 "used MB",

b.bytes / 1024 / 1024 "free MB",

round(((a.bytes - b.bytes) / a.bytes) \* 100, 2) "used%"

from (select tablespace\_name, sum(bytes) bytes

from dba\_data\_files

group by tablespace\_name) a,

(select tablespace\_name, sum(bytes) bytes, max(bytes) largest

from dba\_free\_space

group by tablespace\_name) b

where a.tablespace\_name = b.tablespace\_name

order by ((a.bytes - b.bytes) / a.bytes) desc;

第三种

年末了，表空间也涨到95%了，得要马上扩展表空间，增加数据文件，才能过一个安稳年。增加表空间的时候，我们先查询表空间的占用情况，使用sql如下：

select upper(f.tablespace\_name) "表空间名",

d.tot\_grootte\_mb "表空间大小(M)",

d.tot\_grootte\_mb - f.total\_bytes "已使用空间(M)",

to\_char(round((d.tot\_grootte\_mb - f.total\_bytes) / d.tot\_grootte\_mb \* 100,

2),

'990.99') "使用比",

f.total\_bytes "空闲空间(M)",

f.max\_bytes "最大块(M)"

from (select tablespace\_name,

round(sum(bytes) / (1024 \* 1024), 2) total\_bytes,

round(max(bytes) / (1024 \* 1024), 2) max\_bytes

from sys.dba\_free\_space

group by tablespace\_name) f,

(select dd.tablespace\_name,

round(sum(dd.bytes) / (1024 \* 1024), 2) tot\_grootte\_mb

from sys.dba\_data\_files dd

group by dd.tablespace\_name) d

where d.tablespace\_name = f.tablespace\_name

order by 4 desc;

查询表空间的数据文件的路径，使用sql如下：

select t.\* from sys.dba\_data\_files t where t.tablespace\_name ='PACS31'

1

我们看file\_name，就可以知道表空间的路径了。接下来，我们就可以增加数据文件了

alter tablespace PACS31 add datafile '+DATA/neupacs/pacs3104.dbf' size 10240M;

1

我们不建议使用自动扩展的，就直接加定额的大小就好了，我们使用的例子是增加10G，我们的系统又可以足够空间运行了。这个工作明年才需要做咯！！！

第四种

当前系统的数据量越来越大的，昨天还运行正常的数据库，突然无法使用了。经过定位发现是"ORA-01653 " 即表空间满了，在这里简单记录下处理办法，长期来看oracle是顶不住了，需要尽快切到hadoop的HBase里存储数据

参考如下SQL

----查询表空间使用情况---

SELECT UPPER(F.TABLESPACE\_NAME) "表空间名",

D.TOT\_GROOTTE\_MB "表空间大小(M)",

D.TOT\_GROOTTE\_MB - F.TOTAL\_BYTES "已使用空间(M)",

TO\_CHAR(ROUND((D.TOT\_GROOTTE\_MB - F.TOTAL\_BYTES) / D.TOT\_GROOTTE\_MB \* 100,2),'990.99') "使用比",

F.TOTAL\_BYTES "空闲空间(M)",

F.MAX\_BYTES "最大块(M)"

FROM (SELECT TABLESPACE\_NAME,

ROUND(SUM(BYTES) / (1024 \* 1024), 2) TOTAL\_BYTES,

ROUND(MAX(BYTES) / (1024 \* 1024), 2) MAX\_BYTES

FROM SYS.DBA\_FREE\_SPACE

GROUP BY TABLESPACE\_NAME) F,

(SELECT DD.TABLESPACE\_NAME,

ROUND(SUM(DD.BYTES) / (1024 \* 1024), 2) TOT\_GROOTTE\_MB

FROM SYS.DBA\_DATA\_FILES DD

GROUP BY DD.TABLESPACE\_NAME) D

WHERE D.TABLESPACE\_NAME = F.TABLESPACE\_NAME

ORDER BY 4 DESC;

--查看表空间是否具有自动扩展的能力

SELECT T.TABLESPACE\_NAME,D.FILE\_NAME,

D.AUTOEXTENSIBLE,D.BYTES,D.MAXBYTES,D.STATUS

FROM DBA\_TABLESPACES T,DBA\_DATA\_FILES D

WHERE T.TABLESPACE\_NAME =D.TABLESPACE\_NAME

ORDER BY TABLESPACE\_NAME,FILE\_NAME;

--修改表空间文件扩展方式:

ALTER DATABASE

DATAFILE '/u01/Oracle/oradata/orcl/ccen01.dbf' AUTOEXTEND

ON NEXT 50M MAXSIZE UNLIMITED

增加表空间大小的四种方法

Meathod1：给表空间增加数据文件

ALTER TABLESPACE app\_data ADD DATAFILE

'D:\ORACLE\PRODUCT\10.2.0\ORADATA\EDWTEST\APP03.DBF' SIZE 50M;

Meathod2：新增数据文件，并且允许数据文件自动增长

ALTER TABLESPACE app\_data ADD DATAFILE

'D:\ORACLE\PRODUCT\10.2.0\ORADATA\EDWTEST\APP04.DBF' SIZE 50M

AUTOEXTEND ON NEXT 5M MAXSIZE 100M;

Meathod3：允许已存在的数据文件自动增长

ALTER DATABASE DATAFILE 'D:\ORACLE\PRODUCT\10.2.0\ORADATA\EDWTEST\APP03.DBF'

AUTOEXTEND ON NEXT 5M MAXSIZE 100M;

Meathod4：手工改变已存在数据文件的大小

ALTER DATABASE DATAFILE 'D:\ORACLE\PRODUCT\10.2.0\ORADATA\EDWTEST\APP02.DBF'

RESIZE 100M;

第五种

Oracle表空间扩容

1 创建表空间

SQL> create tablespace my\_01 logging datafile '/oracle/app/oradata/mytablespace/my\_01.dbf' size 128M;

表空间已创建。

在表空间上创建表

SQL> create table t1 (id int) tablespace my\_01;

表已创建。

插入数据

SQL> insert into t1 values(10);

已创建 1 行。

2 表空间扩容

方法一：改变数据文件的大小

SQL> alter database datafile '/oracle/app/oradata/mytablespace/my\_01.dbf' resize 256M;

数据库已更改。

验证：

SQL> select bytes/1024/1024, tablespace\_name from dba\_data\_files where tablespace\_name='MY\_01';

BYTES/1024/1024 TABLESPACE\_NAME

--------------- ---------------

256 MY\_01

SQL> select table\_name from dba\_tables where tablespace\_name='MY\_01';

TABLE\_NAME

------------------------------

T1

SQL> select \* from t1;

ID

----------

10

方法二：添加数据文件

SQL> alter tablespace my\_01 add datafile '/oracle/app/oradata/mytablespace/my\_02.dbf' size 128M;

表空间已更改。

验证：

SQL> select sum(bytes)/1024/1024, tablespace\_name from dba\_data\_files where tablespace\_name='MY\_01' group by tablespace\_name;

SUM(BYTES)/1024/1024 TABLESPACE\_NAME

-------------------- ---------------

384 MY\_01

SQL> select table\_name from dba\_tables where tablespace\_name='MY\_01';

TABLE\_NAME

------------------------------

T1

SQL> select \* from t1;

ID

----------

10

3 查看表空间大小和使用率

select a.tablespace\_name, total, free, total-free as used, substr(free/total \* 100, 1, 5) as "FREE%", substr((total - free)/total \* 100, 1, 5) as "USED%" from

(select tablespace\_name, sum(bytes)/1024/1024 as total from dba\_data\_files group by tablespace\_name) a,

(select tablespace\_name, sum(bytes)/1024/1024 as free from dba\_free\_space group by tablespace\_name) b

where a.tablespace\_name = b.tablespace\_name

order by a.tablespace\_name;

TABLESPACE\_NAME TOTAL FREE USED FREE% USED%

--------------- ---------- ---------- ---------- ---------- ----------

MY\_01 384 381.9375 2.0625 99.46 .5371

QUR\_DT01 128 126.9375 1.0625 99.16 .8300

QUR\_DT02 128 126.9375 1.0625 99.16 .8300

QUR\_DT03 128 126.9375 1.0625 99.16 .8300

QUR\_DT04 128 126.9375 1.0625 99.16 .8300

QUR\_IDX01 128 126.9375 1.0625 99.16 .8300

QUR\_IDX02 128 126.6875 1.3125 98.97 1.025

QUR\_IDX03 128 126.9375 1.0625 99.16 .8300

QUR\_IDX04 128 126.9375 1.0625 99.16 .8300

SYSAUX 500 31.5625 468.4375 6.312 93.68

SYSTEM 680 8.625 671.375 1.268 98.73

UNDOTBS1 75 10.375 64.625 13.83 86.16

USERS 5 3.6875 1.3125 73.75 26.25

第六

打开oracle控制台，在oracle命令行中，输入命令。

一、首先，要连接Linux系统。

二、连上后，进行oracle控制台。输入命令： sqlplus / as sysdba;

三、在oracle命令行中，输入： select t1.name,t2.name from v$tablespace t1,v$datafile t2 where t1.ts# = t2.ts#;

四、这样就可以查看oracle数据库的表空间数据文件位置了。

第七

[linux下 oracle常用命令](https://www.cnblogs.com/ldybky/p/5176996.html)

打开图形化窗口：   
1）Database Configuration Assistant windows    (添加数据库实例)   
$ dbca   
2）Oracle Net Configuration Assistant windows  (配置监听)   
$ netca   
3）打开EM   
$ oemapp dbastudio                            （打开企业管理器图形界面）   
$ opemapp console                             （打开企业管理器图形界面(和上面的命令效果一样)）

常用命令：   
$ lsnrctl start｜stop｜status                  （启动｜停止｜活动状态 监听）   
$ isqlplusctl start｜stop                      （启动｜停止 isqlplus 可以在浏览器登录 5560端口）   
$ sqlplus /nolog                              （以不连接数据库的方式启动sqlplus）   
$ sqlplus system/manager @ file.sql           （执行sql脚本文件）   
$ sqlplus system/manager                      （使用system用户登录sqlplus）   
$ imp system/manager file=/tmp/expfile.dmp log=/tmp/implogfile.log ignore=y fromuser=expuser touser=impuser  （用户模式表数据导入，如果没有特别指定值，就使用默认的值）        
$ exp username/password file=/tmp/expfile.dmp log=/tmp/proV114\_exp.log                                       （用户模式表数据导出，这是最简单的导出方法）

SQL> conn / as sysdba                                  （以sysdba用户连接）   
SQL> startup   
SQL> shutdown   
SQL> shutdown immediate                                （立即关闭实例）   
SQL> desc dba\_users;                                   （查询dba\_users表结构）   
SQL> select username from dba\_users;                   （查询当前sid下的所有用户的username）   
SQL> select count(\*) from username.tablename;          （查询tablename表的行数）   
SQL> drop user username cascade;                       （删除名称为username的oracle用户）   
SQL> select distinct table\_name from user\_tab\_columns; （查看当前user模式下所有表名）

一、ORACLE的启动和关闭   
　　1、在单机环境下   
　　要想启动或关闭ORACLE系统必须首先切换到ORACLE用户，如下   
　　su - oracle

a、启动ORACLE系统   
　　oracle>svrmgrl   
　　SVRMGR>connect internal   
　　SVRMGR>startup   
　　SVRMGR>quit

a、启动ORACLE系统   
　　oracle>svrmgrl   
　　SVRMGR>connect internal   
　　SVRMGR>startup   
　　SVRMGR>quit

2、在双机环境下   
　　要想启动或关闭ORACLE系统必须首先切换到root用户，如下   
　　su － root   
　　   
　　a、启动ORACLE系统   
　　hareg －y oracle   
　　   
　　b、关闭ORACLE系统   
　　hareg －n oracle   
　　   
　　Oracle数据库有哪几种启动方式

说明：   
　　   
　　有以下几种启动方式：   
　　1、startup nomount   
　　非安装启动，这种方式启动下可执行：重建控制文件、重建数据库   
　　   
　　读取init.ora文件，启动instance，即启动SGA和后台进程，这种启动只需要init.ora文件

2、startup mount dbname   
　　安装启动，这种方式启动下可执行：   
　　数据库日志归档、   
　　数据库介质恢复、   
　　使数据文件联机或脱机，   
　　重新定位数据文件、重做日志文件。

执行“nomount”，然后打开控制文件，确认数据文件和联机日志文件的位置，   
　　但此时不对数据文件和日志文件进行校验检查。

3、startup open dbname   
　　先执行“nomount”，然后执行“mount”，再打开包括Redo log文件在内的所有数据库文件，   
　　这种方式下可访问数据库中的数据。

4、startup，等于以下三个命令   
　　startup nomount   
　　alter database mount   
　　alter database open

5、startup restrict   
　　约束方式启动   
　　这种方式能够启动数据库，但只允许具有一定特权的用户访问   
　　非特权用户访问时，会出现以下提示：   
　　ERROR：   
　　ORA-01035: ORACLE 只允许具有 RESTRICTED SESSION 权限的用户使用

6、startup force   
　　强制启动方式   
　　当不能关闭数据库时，可以用startup force来完成数据库的关闭   
　　先关闭数据库，再执行正常启动数据库命令

7、startup pfile=参数文件名   
　　带初始化参数文件的启动方式   
　　先读取参数文件，再按参数文件中的设置启动数据库   
　　例：startup pfile=E:Oracleadminoradbpfileinit.ora

8、startup EXCLUSIVE   
　　二、用户如何有效地利用数据字典   
　　 ORACLE的数据字典是数据库的重要组成部分之一，它随着数据库的产生而产生, 随着数据库的变化而变化,   
　　体现为sys用户下的一些表和视图。数据字典名称是大写的英文字符。   
　　   
　　 数据字典里存有用户信息、用户的权限信息、所有数据对象信息、表的约束条件、统计分析数据库的视图等。   
　　我们不能手工修改数据字典里的信息。   
　　   
　　 很多时候，一般的ORACLE用户不知道如何有效地利用它。   
　　   
　　 dictionary 全部数据字典表的名称和解释，它有一个同义词dict   
　　 dict\_column 全部数据字典表里字段名称和解释   
　　   
　　 如果我们想查询跟索引有关的数据字典时，可以用下面这条SQL语句:

　　 SQL>select column\_name,comments from dict\_columns where table\_name='USER\_INDEXES';   
　　   
　　 依此类推，就可以轻松知道数据字典的详细名称和解释，不用查看ORACLE的其它文档资料了。   
　　   
　　 下面按类别列出一些ORACLE用户常用数据字典的查询使用方法。

 1、用户   
　　   
　　 查看当前用户的缺省表空间   
　　 SQL>select username,default\_tablespace from user\_users;   
　　   
　　 查看当前用户的角色   
　　 SQL>select \* from user\_role\_privs;   
　　   
　　 查看当前用户的系统权限和表级权限   
　　 SQL>select \* from user\_sys\_privs;   
　　 SQL>select \* from user\_tab\_privs;

 2、表   
　　   
　　 查看用户下所有的表   
　　 SQL>select \* from user\_tables;   
　　   
　　 查看名称包含log字符的表   
　　 SQL>select object\_name,object\_id from user\_objects   
　　 where instr(object\_name,'LOG')>0;   
　　   
　　 查看某表的创建时间   
　　 SQL>select object\_name,created from user\_objects where object\_name=upper('&table\_name');   
　　   
　　 查看某表的大小   
　　 SQL>select sum(bytes)/(1024\*1024) as "size(M)" from user\_segments

       where segment\_name=upper('&table\_name');   
　　   
　　 查看放在ORACLE的内存区里的表   
　　 SQL>select table\_name,cache from user\_tables where instr(cache,'Y')>0;

3、索引   
　　   
　　 查看索引个数和类别   
　　 SQL>select index\_name,index\_type,table\_name from user\_indexes order by table\_name;   
　　   
　　 查看索引被索引的字段   
　　 SQL>select \* from user\_ind\_columns where index\_name=upper('&index\_name');   
　　   
　　 查看索引的大小   
　　 SQL>select sum(bytes)/(1024\*1024) as "size(M)" from user\_segments   
　　 where segment\_name=upper('&index\_name');

4、序列号   
　　   
　　 查看序列号，last\_number是当前值   
　　 SQL>select \* from user\_sequences;   
　　   
　　 5、视图   
　　   
　　 查看视图的名称   
　　 SQL>select view\_name from user\_views;   
　　   
　　 查看创建视图的select语句   
　　 SQL>set view\_name,text\_length from user\_views;   
　　 SQL>set long 2000; 说明：可以根据视图的text\_length值设定set long 的大小   
　　 SQL>select text from user\_views where view\_name=upper('&view\_name');

6、同义词   
　　   
　　 查看同义词的名称   
　　 SQL>select \* from user\_synonyms;   
　　   
　　 7、约束条件   
　　   
　　 查看某表的约束条件   
　　 SQL>select constraint\_name, constraint\_type,search\_condition, r\_constraint\_name   
　　 from user\_constraints where table\_name = upper('&table\_name');   
　　   
　　 SQL>select c.constraint\_name,c.constraint\_type,cc.column\_name   
　　 from user\_constraints c,user\_cons\_columns cc   
　　 where c.owner = upper('&table\_owner') and c.table\_name = upper('&table\_name')   
　　 and c.owner = cc.owner and c.constraint\_name = cc.constraint\_name   
　　 order by cc.position;

 8、存储函数和过程   
　　   
　　 查看函数和过程的状态   
　　 SQL>select object\_name,status from user\_objects where object\_type='FUNCTION';   
　　 SQL>select object\_name,status from user\_objects where object\_type='PROCEDURE';   
　　   
　　 查看函数和过程的源代码   
　　 SQL>select text from all\_source where owner=user and name=upper('&plsql\_name');

三、查看数据库的SQL   
　　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$ro