

创建表空间及用户（类似mysql的database）

```
sqlplus /nolog
conn / as sysdba
CREATE TABLESPACE T_ZS_SUDY DATAFILE 'T_ZS_SUDY.dbf' SIZE 5M AUTOEXTEND ON NEXT 5M MAXSIZE UNLIMITED;
-- CREATE TABLESPACE sudy_imp DATAFILE '/opt/dbdata/local/sudy_imp.dbf' SIZE 2048M AUTOEXTEND ON NEXT 100M EXT MANAGEMENT
LOCAL SEGMENT SPACE MANAGEMENT AUTO;
create user ZS_SUDY identified by sudy#12344 default tablespace T_ZS_SUDY;
grant connect ,resource ,dba to ZS_SUDY;
```

账号密码 ZS\_SUDY sudy#12344

```
create user sudy_imp identified by sudy_imp default tablespace SUDY_IMP_TS ;
grant connect ,resource ,dba to sudy_imp;
```

删除用户 - 查看用户

```
select saddr,sid,serial#,paddr,username,status from v$session where username is not null;
select saddr,sid,serial#,paddr,username,status from v$session where USERNAME='SUDY_IMP';
SQL> SELECT SID,SERIAL# FROM V$SESSION WHERE USERNAME='SUDY_IMP';
```

SID	SERIAL#
33	177

```
SQL> ALTER SYSTEM KILL SESSION '33,177';
```

```
SQL> drop user sudy_imp cascade;
```

查看表空间

```
select tablespace_name from user_tablespaces;
```

删除表空间

```
drop tablespace test_temp including CONTENTS and datafiles;
```

```
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;
```

查看表空间下用户

```
select distinct s.owner from dba_segments s where s.tablespace_name ='SUDY_IMP_TS';
```

1、创建表空间

```
CREATE TABLESPACE T_ZS_JSJM DATAFILE 'T_ZS_JSJM.dbf' SIZE 5M AUTOEXTEND ON NEXT 5M MAXSIZE UNLIMITED;
```

```
create tablespace T_ZS_JSM logging datafile 'F:\oracleData\eosjq.dbf' size 64m autoextend on next 64m maxsize 2048m
extent management local;
```

--表空间名称

```
CREATE TABLESPACE T_ZS_SUDY
```

--数据文件存放位置

```
DATAFILE ' /opt/oracle/app/oracle/oradata/yqcs/T_alter table dbf'
```

--表空间默认1G,并自动增加

```
SIZE 1024M AUTOEXTEND ON
```

--每次增加100M

```
NEXT 100M
```

--本地管理 本地管理表空间是自己管理分配，而不是象字典管理表空间需要系统来管理空间分配，本地表空间是通过在表空间的每个数据文件中维持一个位图来跟踪在此文件中块的剩余空间及使用情况。并及时做更新。

```
EXTENT MANAGEMENT LOCAL
```

--段空间自动管理

```
SEGMENT SPACE MANAGEMENT AUTO
```

## 2、创建用户

```
CREATE USER JSJMZS IDENTIFIED BY jsjmzs#2017 DEFAULT TABLESPACE USERS;
```

--在T\_ZS\_ZSCS表空间下创建ZS\_SUDY用户，并设置密码为sudy#12344(用户密码现场自行设置)

```
create user ZS_SUDY identified by sudy#12344 default tablespace T_ZS_SUDY ;
```

## 3、成功创建用户后，进行授权。

--授权ZS\_SUDY用户登录权限

```
grant connect ,resource ,dba to ZS_SUDY;
```

说明：数据库用户创建完毕，既可以用改用户去登录，然后导入数据库脚本。

DBA： 拥有全部特权，是系统最高权限，只有DBA才可以创建数据库结构。

RESOURCE：拥有Resource权限的用户只可以创建实体，不可以创建数据库结构。

对于普通用户：授予connect, resource权限。

对于DBA管理用户：授予connect, resource, dba权限。

例：

```
GRANT ALL PRIVILEGES TO JSJMZS;
```

```
GRANT CONNECT,RESOURCE,DBA TO JSJMZS;
```

```
GRANT CONNECT TO jsjm_zs; //可以登录Oracle，不可以创建实体，不可以创建数据库结构。
```

```
GRANT CREATE SESSION TO jsjm_zs; //创建会话的权限
```

## 4、删除表空间

```
drop tablespace test_temp including CONTENTS and datafiles;
```

## 5、修改用户密码

```
alter user test identified by new_password;
```

## 6、删除用户

```
drop user usr_zs cascade; --//执行该语句请小心，会级联删除该用户下所有对象。
```

## 7、删除tablespace

```
DROP TABLESPACE tablespace_name INCLUDING CONTENTS AND DATAFILES;
```

## 9、修改表空间名称

```
alter tablespace T_ZS_D rename to JQZS;
```

## 10、查看oracle的sid

```
select INSTANCE_NAME from v$instance;
```

11、查询数据库数据存储路径

```
select file_name From Db_Data_Files;
select name from v$controlfile;
```

12、修改字段名称

```
/*修改原字段名name为name_tmp*/
alter table tb rename column name to name_tmp;

/*增加一个和原字段名同名的字段name*/
alter table tb add name varchar2(40);

/*将原字段name_tmp数据更新到增加的字段name*/
update tb set name=trim(name_tmp);

/*更新完，删除原字段name_tmp*/
alter table tb drop column name_tmp;
```

13、查询哪些对象被锁: select object\_name,machine,s.sid,s.serial# from v\$locked\_object l,dba\_objects o ,v\$session s where l.object\_id=o.object\_id and l.session\_id=s.sid;

14、杀死一个进程: (其中160, 48分别是上面查询出的sid,serial#)

```
alter system kill session '160,48';
```

15、替换字符串

```
select t.*, t.rowid from t_zs_examinee t where t.recruittermid=64;
update t_zs_examinee t set t.zkzh=replace(t.zkzh,'1156180','1156250') where t.recruittermid=64;
```

16、查看oracle版本

方法一: v\$version

```
SQL> select * from v$version;
```

BANNER

```
-----
Oracle Database 11g Enterprise Edition Release 11.1.0.7.0 - 64bit Production
PL/SQL Release 11.1.0.7.0 - Production
CORE      11.1.0.7.0      Production
TNS for Linux: Version 11.1.0.7.0 - Production
NLSRTL Version 11.1.0.7.0 - Production
```

方法二: product\_component\_version

```
SQL> select * from product_component_version;
```

PRODUCT	VERSION	STATUS
NLSRTL	11.1.0.7.0	Production
Oracle Database 11g Enterprise Edition	11.1.0.7.0	64bit Production
PL/SQL	11.1.0.7.0	Production
TNS for Linux:	11.1.0.7.0	Production

17、查看具体的被锁时间: select username,lock\_date from dba\_users where username='zs\_zscs';

解锁: alter user zs\_zscs account unlock;

## 18、linux服务器连接oracle的方法

```
$ su - oracle
$ sqlplus /nolog
sql> conn / as sysdba
sql> startup (一般不需要加参数, 只要设置好环境变量)
sql> quit (退出sql模式)
```

```
$ lsnrctl start (启动监听器) 关闭oracle
$ lsnrctl stop (关闭监听器, 在这之前, 应该先关闭应用程序)
```

```
$ sqlplus /nolog
sql>shutdown 其参数 : shutdown有四个参数, 四个参数的含义如下:
```

normal 需要在所有连接用户断开后才执行关闭数据库任务, 所以有的时候看起来好象命令没有运行一样! 在执行这个命令后不允许新的连接

immediate 在用户执行完正在执行的语句后就断开用户连接, 并不允许新用户连接。

transactional 在拥护执行完当前事物后断开连接, 并不允许新的用户连接数据库。

abort 执行强行断开连接并直接关闭数据库, 不做任何等待, 。

前三种方式不回丢失用户数据。第四种在不的己的情况下, 不建议采用!

经常遇到的问题:

- 1) 权限问题, 解决方法, 切换到oracle用户;
- 2) 没有关闭监听器, 解决方法: 关闭监听器
- 3) 有oracle实例没有关闭, 解决办法: 关闭oracle实例
- 4) 环境变量设置不全, 解决办法: 修改环境变量

## 19、修改数据库操作如下

```
su - oracle
sqlplus /nolog
connect /as sysdba
shutdown immediate
sql> conn usr_zs/swzs1204 用户名/密码
sql> select sequence_name from user_tables; //查询当前用户的所有表
SQL> alter table t_swrs_examinee add wyTotalScores VARCHAR2(200);
```

## 20、用户被锁住了解决办法:

第一步: 清除session

```
SELECT s.sid, s.serial# FROM v$locked_object lo, dba_objects ao, v$session s WHERE ao.object_id = lo.object_id AND
lo.session_id = s.sid;
ps -ef|grep $ORACLE_SID|grep -v ora_|grep LOCAL=NO|awk '{print $2}'|xargs kill
```

第二步:

```
sqlplus /nolog
sql>connect /as sysdba
sql>shutdown immediate
```

第三步:

查看监听状态: lsnrctl -----> status -----> exit

停止监听: lsnrctl stop

启动oracle

第一步: 启动监听 > lsnrctl start

第二步: sqlplus /nolog

第三步: connect /as sysdba

第四步: startup

21、查询表空间: SELECT \* FROM v\$tablespace;

22、11g oracle导出表时会默认不导出数据为空 (oracle11g 导出表报EXP-00011: table不存在)

1) Oracle11g默认对空表不分配segment, 故使用exp导出Oracle11g数据库时, 空表不会导出。

2) 设置deferred\_segment\_creation 参数为FALSE后, 无论是空表还是非空表, 都分配segment。

在sqlplus中, 执行如下命令:

SQL>alter system set deferred\_segment\_creation=false; 查看:

SQL>show parameter deferred\_segment\_creation;

该值设置后只对后面新增的表产生作用, 对之前建立的空表不起作用。

解决方案: 1. 用以下这句查找空表并生成执行命令: select 'alter table '||table\_name||' allocate extent;' from user\_tables where num\_rows=0;

2. 将查询结果的内容导出, 执行导出的语句。如: alter table T\_AUTHUSERMEMEBER allocate extent;

3. 然后再用exp的方式去导出数据库, 就可以完整的导出包括空表的数据库。

23、EXP/IMP 命令参数

一、EXP:

注: 导出之前先进入oracle用户模式下, su - oracle

1、完全:

EXP SYSTEM/MANAGER BUFFER=64000 FILE=C:\FULL.DMP FULL=Y

如果要执行完全导出, 必须具有特殊的权限

2、用户模式:

EXP SONIC/SONIC BUFFER=64000 FILE=C:\SONIC.DMP OWNER=SONIC

EXP userName/password@databaseName BUFFER=64000 FILE=C:\SONIC.DMP OWNER=webplus;

例子:

exp usr\_zs/swzs1204 BUFFER=64000 FILE=/home/oracle/swzs\_1209.dmp OWNER=usr\_zs;

EXP jsjms/jsjms BUFFER=64000 FILE=/opt/jsjms\_180227.dmp OWNER=jsjms;

exp jsjms/jsjms@ORCL file=/opt/jsjms\_180227.dmp full=y

connect /as sysdba

871 connect /as user\_zs

872 sqlplus /nolog

873 su -oracle

874 su - oracle

875 su - oracle

876 exp jsjms/jsjms BUFFER=64000 file=/opt/yfli/zs0721.dmp owner=jsjms

877 exp jsjms/jsjms@sudytech file= /opt/yfli/zs0721.dmp owner=jsjms;

878 exp jsjms/jsjms@sudytech file= /opt/yfli/zs0721.dmp owner=jsjms FULL=Y

```

879 exp jsjms/jsjms@sudytech file=/opt/zs0721.dmp owner=jsjms FULL=Y
880 EXP jsjms/jsjms@sudytech file=/opt/zs0721.dmp owner=jsjms FULL=Y
881 history
882 create directory abc as /opt/leeyefang
883 su - oracle
884 expdp jsjms/jsjms@sudytech schemas=scott dumpfile=zs0721.dmp directory=abc;
885 su -oracle

```

sql: create directory abc as '/opt/leeyefang'; 注：在对应的目录下要创建文件夹XXX

c:/> expdp jsjms/jsjms@ORCL directory=abc dumpfile=hz.dmp logfile=hz.log version=10.2.0.3.0

这样用户SONIC的所有对象被输出到文件中。

### 3、表模式：

```
EXP SONIC/SONIC BUFFER=64000 FILE=C:\SONIC.DMP OWNER=SONIC TABLES=(SONIC)
```

这样用户SONIC的表SONIC就被导出

例子：

```
exp jsjms/jsjms file=/opt/yfli/zs0721.dmp tables=table_name like 'T_ZS%'; 导出部分表数据
```

exp参数：

关键字 说明（默认）

---

USERID 用户名/口令  
 FULL 导出整个文件 (N)  
 BUFFER 数据缓冲区的大小  
 OWNER 所有者用户名列表  
 FILE 输出文件 (EXPDAT.DMP)  
 TABLES 表名列表  
 COMPRESS 导入一个范围 (Y)  
 RECORDLENGTH IO 记录的长度  
 GRANTS 导出权限 (Y)  
 INCTYPE 增量导出类型  
 INDEXES 导出索引 (Y)  
 RECORD 跟踪增量导出 (Y)  
 ROWS 导出数据行 (Y)  
 PARFILE 参数文件名  
 CONSTRAINTS 导出限制 (Y)  
 CONSISTENT 交叉表一致性  
 LOG 屏幕输出的日志文件  
 STATISTICS 分析对象 (ESTIMATE)  
 DIRECT 直接路径 (N)  
 TRIGGERS 导出触发器 (Y)  
 FEEDBACK 显示每 x 行 (0) 的进度  
 FILESIZE 各转储文件的最大尺寸  
 QUERY 选定导出表子集的子句

### 24、IMP：具有三种模式（完全、用户、表）

### 1、完全：

```
IMP SYSTEM/MANAGER BUFFER=64000 FILE=C:\FULL.DMP FULL=Y
IMP cxlu/cxlu BUFFER=64000 FILE=D:\150731.DMP FULL=Y
```

### 2、用户模式：

```
IMP SONIC/SONIC BUFFER=64000 FILE=C:\SONIC.DMP FROMUSER=SONIC TOUSER=SONIC
```

这样用户SONIC的所有对象被导入到文件中。必须指定FROMUSER、TOUSER参数，这样才能导入数据。

```
IMP ids/12344@orcl BUFFER=64000 FILE=E:\电信\linux1225\ids.dmp FROMUSER=ids TOUSER=ids
imp eos_nd/eos_nd BUFFER=64000 FILE=D:\nd20170322.DMP FROMUSER=eos_nd TOUSER=eod_nd
imp jsjmzs/jsjmzs BUFFER=64000 FILE=/jsjm/CorePlus_tempuser_exp.dmp FULL=y
```

```
IMP jsjmzs/jsjmzs@sudytech BUFFER=64000 FILE=/jsjm/CorePlus_tempuser_exp.dmp FROMUSER=jsjmzs TOUSER=jsjmzs
```

### 3、表模式：

```
IMP SONIC/SONIC BUFFER=64000 FILE=C:\SONIC.DMP OWNER=SONIC TABLES=(SONIC)
```

这样用户SONIC的表SONIC就被导入。

### imp参数：

关键字	说明（默认）
-----	--------

USERID	用户名/口令
FULL	导入整个文件 (N)
BUFFER	数据缓冲区大小
FROMUSER	所有人用户名列表
FILE	输入文件 (EXPDAT.DMP)
TOUSER	用户名列表
SHOW	只列出文件内容 (N)
TABLES	表名列表
IGNORE	忽略创建错误 (N)
RECORDLENGTH	IO 记录的长度
GRANTS	导入权限 (Y)
INCTYPE	增量导入类型
INDEXES	导入索引 (Y)
COMMIT	提交数组插入 (N)
ROWS	导入数据行 (Y)
PARFILE	参数文件名
LOG	屏幕输出的日志文件
CONSTRAINTS	导入限制 (Y)
DESTROY	覆盖表空间数据文件 (N)
INDEXFILE	将表/索引信息写入指定的文件
SKIP_UNUSABLE_INDEXES	跳过不可用索引的维护 (N)
ANALYZE	执行转储文件中的 ANALYZE 语句 (Y)
FEEDBACK	显示每 x 行 (0) 的进度
TOLD_NOVALIDATE	跳过指定类型 id 的校验
FILESIZE	各转储文件的最大尺寸
RECALCULATE_STATISTICS	重新计算统计值 (N)

### 25、删除归档日志

步骤一：现在来清理一下archivelog归档日志，生产环境建议备份

查询日志目录位置

```
show parameter recover;
```

删除归档日志, USERDB是数据库实例名

```
cd /u01/app/oracle/flash_recovery_area/USERDB/archivelog
```

使用root帐户删除该目录下的文件或者备份其它地方

步骤二: 使用rman 操作, 内容太多, 只做部分截图

```
[oracle@userbeta archivelog]$ rman
RMAN> connect target /
crosscheck archivelog all;
delete expired archivelog all;
//此步会有提示, 输入 YES 回车
host; //退出rman
```

步骤三: 确认是否操作成功

```
# sqlplus /nolog
SQL> connect /as sysdba
SQL> select * from V$FLASH_RECOVERY_AREA_USAGE;
```

1. 进入rman

```
[oracle@dthxdb2 ~]$ /u01/app/oracle/product/10.2.0/db/bin/rman
```

2. connect target /

```
RMAN> connect target /
```

3. crosscheck archivelog all;

crosscheck archivelog all: 验证的是DB的归档日志即log\_archive\_dest参数指定位置的文件, 当手工删除了归档日志以后, Rman备份会检测到日志缺失, 从而无法进一步继续执行Rman备份, 所以此时需要手工执行crosscheck过程, 查看所有的归档日志文件是否都是正常的然后再来执行Rman备份。

4. delete expired archivelog all;

这时候我们再去OEM中就看不到这些日志文件了, 如果你的从来没有做过这个动作的话, 我们可以比较这个动作前的controlfile和动作后的controlfile的文件大小