

SQLPLUS 使用技巧

```
select last_name,salary,commission_pct
from employees
where rownum<=40;
```

```
set pagesize 100
set feedback off
set timing on
```

技巧 1

Sql*plus 中使用绑定变量:

```
conn scott/tiger;
variable x number;
exec :x := 7788;
SELECT empno,ename from scott.emp where empno=:x;
```

技巧 2

save,get,edit 的使用

(1)使用 save 命令将当前 sql 缓冲区的内容保存到 sql 脚本中。当执行该命令时，默认选项为 create，即建立新文件。

```
SQL> save c:\a.sql create
已创建 file c:\a.sql
```

- 当执行命令之后，就会建立新脚本文件 a.sql，并将 sql 缓冲区内容存放到该文件中
- 如果 sql 已经存在，使用 replace 选项可以替代已存在的 sql 脚本
- 如果要给已存在的 sql 脚本追加内容，可以使用 append 选项。

(2)get

该命令与 save 命令作用恰好相反，用于将 sql 脚本中的所有内容装载到 sql 缓冲区中。

```
SQL> get c:\a.sql
1* select ename from emp where deptno=20
```

(3)ed[it]

- 该命令不仅可用于编辑 sql 缓冲区内容，也可以用于编辑 sql 脚本文件。
- 当运行该命令时，会启动默认的系统编辑器来编辑 sql 脚本。运行方法为：
sql>edit c:/a.sql

技巧 3

&、&&的使用及区别

- 我们经常会 sql*plus 中定义变量，并且在运行 sql 脚本时可以为这些变量动态输入数据。

- &标号所定义的替代变量只在当前语句中起作用;而&&标号所定义的变量会在当前 sql*plus 环境中一直生效。

```
conn hr/hr;
```

```
select employee_id from employees where employee_id=&id1
```

输入 192

再次执行

```
select employee_id from employees where employee_id=&id1
```

需要再次输入

```
select employee_id from employees where employee_id=&&id1
```

输入 192

再次执行

```
select employee_id from employees where employee_id=&&id1
```

需要再次输入

技巧 4

spool 使用的技巧

批量数据文件 offline 脚本自动生成。

实际技巧:

```
startup mount
```

```
alter database rename file 'C:\ORACLE\PRODUCT\10.2.0\ORADATA\SAMPLE\test01.DBF'
```

to

```
'c:\oradata\test01.dbf';
```

```
alter database rename file 'C:\ORACLE\PRODUCT\10.2.0\ORADATA\SAMPLE\test02.DBF'
```

to

```
'c:\oradata\test02.dbf';
```

```
-----  
#!/bin/ksh
```

```
ORACLE_SID=ora92
```

```
export $ORACLE_SID
```

```
echo "ORACLE_SID=ora92" >> /tmp/datafile_offline.sql
```

```
echo "\nexport $ORACLE_SID" >> /tmp/datafile_offline.sql
```

```
echo "\nsqlplus -S '/' as sysdba" << EOF" >> /tmp/datafile_offline.sql
```

```
LOGFILE=/tmp/datafile_offline.sql
```

```
sqlplus -S '/' as sysdba" << EOF
```

```
set head off
```

```
set pagesize 5000
```

```
spool $LOGFILE append
```

```
select 'alter database datafile '||file_id||' offline;'
```

```
from dba_data_files
```

```
where tablespace_name='TEST1';
```

```
spool off
```

```
exit
```

```
EOF
```

```
echo "\nexit" >> /tmp/datafile_offline.sql
```

```
echo "\nEOF" >> /tmp/datafile_offline.sql
```

SET TERM 是否在屏幕上显示输出的内容，主要用与 SPOOL 结合使用。

```
SQL> SET TERM[OUT] {ON|OFF}
```

在用 spool 命令将一个大表中的内容输出到一个文件中时，将内容输出在屏幕上会耗费大量的时间，设置 set term off 后，则输出的内容只会保存在输出文件中，不会显示在屏幕上，极大的提高了 spool 的速度。

将 SPOOL 输出中每行后面多余的空格去掉

```
SQL> SET TRIMS[OUT] {ON|OFF}
```

动态生成 spool 命令所需的文件名。

在我们上面的例子中，spool 命令所需要的文件名都是固定的。有时我们需要每天 spool 一次，并且每次 spool 的文件名都不相同，如文件名包含当天的日期，该如何实现呢？

```
column dat1 new_value filename;  
select to_char(sysdate,'yyyymmddhh24mi') dat1 from dual;  
spool c:\&&filename..txt  
select * from employees;  
spool off;
```

技巧 5

如何指定缺省的编辑脚本的目录：

- 在 sql*plus 中，可以用 save 命令，将上一条执行的 sql 语句保存到一个文件中，但是如何设置该文件的缺省目录呢？
- 通过 SQL> set editfile c:\temp\file.sql 命令，可以设置其缺省目录为 c:\tmpe，缺省文件名为 file.sql。

技巧 6

如何忽略空行

```
set sqlblanklines on
```

技巧 7

如何能够让 sqlplus 具备分屏显示功能

```
set pause on
```

这时候 pagesize 值决定了一屏显示多少条记录

sho all 的使用