

PLSQL——之SQLLDR及SPOOL操作

SQLLDR

一、作用

在 Oracle 数据库中，我们通常在**不同数据库的表间**记录进行**复制或迁移**时会用Sql Loader(sqlldr) 来导入数据，效果比起逐条 insert 来的明显。sqlldr生成的文件后缀是ctl结尾。

二、参数介绍

在命令行下执行 Oracle 的 sqlldr 命令，可以看到它的详细参数说明，要着重关注以下几个参数：

```
1  userid -- Oracle 的username/password[@servicename]
2  control -- 控制文件，可能包含表的数据
3  log -- 记录导入时的日志文件，默认为 控制文件(去除扩展名).log
4  bad -- 坏数据文件，默认为 控制文件(去除扩展名).bad
5  data -- 数据文件，一般在控制文件中指定。用参数控制文件中不指定数据文件更适于自动操作
6  errors -- 允许的误差记录数，可以用他来控制一条记录都不能错
7  rows -- 多少条记录提交一次，默认为 64
8  skip -- 跳过的行数，比如导出的数据文件前面几行是表头或其他描述
```

还有更多的 sqlldr 的参数说明请参考：[sql loader的用法](#)。

三、运用举例

1、sqlldr 的使用，有**两种使用方法**：

- (1) 只使用一个控制文件，在这个控制文件中包含数据
- (2) 使用一个**控制文件**(作为模板) 和一个**数据文件** (通常是csv或者txt结尾的文件)

一般为了利于模板和数据的分离，以及程序的不同分工会使用第二种方式，所以先来看这种用法。数据文件可以是 CSV 文件或者以其他分割符分隔的，数据文件可以用 PL/SQL Developer 或者 Toad 导出，也可以用[SQL *Plus 的 spool 格式化产出](#)，或是 [UTL FILE 包生成](#)。另外，用 Toad 还能直接生成包含数据的控制文件。

2、具体案例：

- (1) 首先，假定有这么一个表 users，并插入五条记录：
- (2) 使用一个控制文件(作为模板) 和一个数据文件
- (3) 建立数据文件，我们这里用 PL/SQL Developer 导出表 users 的记录为 users_data.csv 文件，内容如下：

```
1  1. "", "USER_ID", "USER_NAME", "LOGIN_TIMES", "LAST_LOGIN"
2  2. "1", "1", "Unmi", "3", "2009-1-5 20:34:44"
3  3. "2", "2", "", "5", "2008-10-15"
4  4. "3", "3", "隔叶黄莺", "8", "2009-1-2"
5  5. "4", "4", "Kypfos", "", ""
6  6. "5", "5", "不知秋", "1", "2008-12-23"
```

(4) 建立一个控制文件 users.ctl, 内容如下:

```
1 1. OPTIONS (skip=1,rows=128) -- sqlldr 命令显示的 选项可以写到这里边来,skip=1 用来
   跳过数据中的第一行
2 2. LOAD DATA
3 3. INFILE "users_data.csv" --指定外部数据文件, 可以写多 个 INFILE
   "another_data_file.csv" 指定多个数据文件
4 4. --这里还可以使 用 BADFILE、DISCARDFILE 来指定坏数据和丢弃数据的文件,
5 5. truncate --操作类型, 用 truncate table 来清除表中原有记录
6 6. INTO TABLE users -- 要插入记录的表
7 7. Fields terminated by "," -- 数据中每行记录用 "," 分隔
8 8. Optionally enclosed by '"' -- 数据中每个字段用 '"' 框起, 比如字段中有 "," 分隔符
   时
9 9. trailing nullcols --表的字段没有对应的值时允 许为空
10 10. (
11 11. virtual_column FILLER, --这是一个虚拟字段, 用来跳过由 PL/SQL Developer 生成的
   第一列序号
12 12. user_id number, --字段可以指定类型, 否则认为是 CHARACTER 类型, log 文件中有显
   示
13 13. user_name,
14 14. login_times,
15 15. last_login DATE "YYYY-MM-DD HH24:MI:SS" -- 指定接受日期的格式, 相当用
   to_date() 函数转换
16 16. )
```

```
1 说明: 在操作类型 truncate 位置可用以下中的一值:
2 1) insert --为缺省方式, 在数据装载开始时要求表为空
3 2) append --在表中追加新记录
4 3) replace --删除旧记录(用 delete from table 语句), 替换成新装载的记录
5 4) truncate --删除旧记录(用 truncate table 语句), 替换成新装载的记录
```

(5) 执行命令:

sqlldr [dbuser/dbpass@dbservice](#) control=users.ctl

在 dbservice 指示的数据库的表 users 中记录就和数据文件中的一样了。

执行完 sqlldr 后希望能留意一下生成的几个文件, 如 users.log 日志文件、users.bad 坏数据文件等。特别是要看看日志文件, 从中可让你更好的理解 Sql Loader, 里面有对控制文件的解析、列出每个字段的类型、加载记录的统计、出错原因等信息。

SPOOL

一、作用

SPOOL是SQLPLUS的命令, 不是SQL语法里面的东西, 用于快速导出表格

对于SPOOL数据的SQL, 最好要自己定义格式, 以方便程序直接导入,SQL语句如:

```
select
taskindex||'|'||commonindex||'|'||tasktype||'|'||to_number(to_char(sysdate,'YYYYMMDD'))
from ssrv_sendsms_task;
```

二、spool常用的设置

```

1  set colsep' ';           //域输出分隔符
2  set echo off;           //显示start启动的脚本中的每个sql命令，缺省为on
3  set feedback off;       //回显本次sql命令处理的记录条数，缺省为on
4  set heading off;        //输出域标题，缺省为on
5  set pagesize 0;         //输出每页行数，缺省为24,为了避免分页，可设定为0。
6  set termout off;        //显示脚本中的命令的执行结果，缺省为on
7  set trimout on;         //去除标准输出每行的拖尾空格，缺省为off
8  set trimspool on;       //去除重定向（spool）输出每行的拖尾空格，缺省为off

```

```

1  导出文本数据的建议格式：
2  SQL*PLUS环境设置
3
4  SET NEWPAGE NONE
5  SET HEADING OFF
6  SET SPACE 0
7  SET PAGESIZE 0
8  SET TRIMOUT ON
9  SET TRIMSPPOOL ON
10 SET LINESIZE 2500      ——LINESIZE要稍微设置大些，免得数据被截断，它应和相应的
    TRIMSPPOOL结合使用防止导出的文本有太多的尾部空格

```

注：LINESIZE要稍微设置大些，免得数据被截断，它应和相应的TRIMSPPOOL结合使用防止导出的文本有太多的尾部空格。但是如果 LINESIZE设置太大，会大大降低导出的速度，另外在WINDOWS下导出最好不要用PLSQL导出，速度比较慢，直接用COMMENT下的 SQLPLUS命令最小化窗口执行。

对于字段内包含很多回车换行符的应该给与过滤，形成比较规矩的文本文件。通常情况下，我们使用SPOOL方法，将数据库中的表导出为文本文件的时候会采用两种方法，如下述：

三、代码示范

```

1  set trimspool on
2  set linesize 120
3  set pagesize 2000
4  set newpage 1
5  set heading off
6  set term off
7
8  spool 路径+文件名
9  select col1||','||col2||','||col3||','||col4||'..' from tablename;
10 spool off

```

自测例：将ssrv_sendsms_task表中的数据导出到文本(数据库Oracle 9i 操作系统 SUSE LINUX Enterprise Server 9)

```

1  -----代码-----
2  spool_test.sh脚本如下：
3  \#!/bin/sh
4  DB_USER=zxdbm_ismp           #DB  USER
5  DB_PWD=zxin_smap            #DB  PASSWORD
6  DB_SERV=zx10_40_43_133      #DB  SERVICE NAME
7
8  send_day=`sqlplus -s $DB_USER/$DB_PWD@$DB_SERV<<! # -s 参数屏蔽打印到屏幕上的其他
    信息，只显示sql执行信息
9  set trimspool on
10 set linesize 120

```

```
11 set pagesize 2000
12 set newpage 1
13 set heading off
14 set term off
15 spool sp_test.txt
16 select
17 taskindex||' '||
18 commonindex||' '||
19 tasktype||' '||
20 to_number(to_char(sysdate,'YYYYMMDD')) from ssrv_sendsms_task;
21 spool off
22
23 -----执行结果-----
24 执行./spool_test.sh后生成sp_test.txt，内容如下：
25 83|115|1|20080307
26 85|115|11|20080307
27 86|115|10|20080307
28 84|115|2|20080307
29 6|5|14|20080307
30 7|5|12|20080307
31 9|5|15|20080307
```