

基于 ORACLE7 的 SQLLOAD 使用技巧

田振光 王雪莹 (冶金部信息研究中心 100711)

摘要:本文概括介绍 ORACLE7 环境下利用解释执行程序 AWK、DOS 中的文件拼接以及数据库 Triggers, 实现 SQLLOAD 的参数传递及数据复盖。

关键词: 参数传递 数据复盖 AWK 数据库 Triggers SQLLOAD

一、引言

笔者在参加研制冶金经济信息系统的过程中, 曾用到 ORACLE7 环境下的 SQLLOAD, 用它将数据文件批量装入数据库中。经反复研究使用发现其具有一定局限性, 比如某数据文件中不含日期列, 而希望利用 SQLLOAD 装库时将日期参数传递过去, 从而使装入数据库中的数据具有日期列; 向数据库中装入数据时, 希望达到有条件数据复盖。这两项功能直接使用 SQLLOAD 都无法实现, 为解决这些问题, 我们利用了 UNIX 中的解释执行程序 AWK、DOS 中的文件拼接及数据库 Triggers, 结合 SQLLOAD 实现参数传递及数据复盖。

二、处理流程

在利用 AWK、文件拼接以及数据库 Triggers 的过程中, 要遵循一定的结构顺序, 其处理流程如图 1 所示。

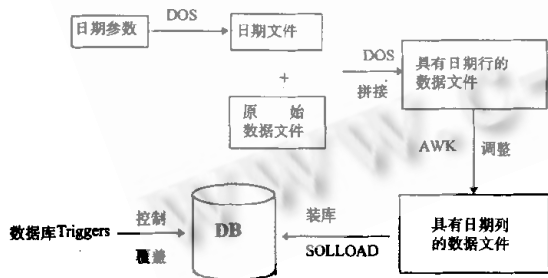


图 1 处理流程

首先利用 DOS 环境将日期参数存放于一个文件中, 称之为日期文件; 将日期文件与原始数据文件顺序拼接, 生成具有日期行的数据文件; 利用 AWK 对具有日期

行的数据文件进行结构调整, 生成具有日期列的数据文件; 利用 SQLLOAD 将文件装入数据库中, 装库过程由数据库 Triggers 控制数据复盖。

三、实现技术

在实现参数传递及数据复盖的过程中, 有几项技术很关键, 主要包括文件拼接、文件结构的转换与调整、数据复盖的控制及数据的顺序控制等。

1. 文件拼接

文件拼接是实现参数传递的基础, 它包括三个主要内容, 即以 DOS 中的批处理文件为参数入口, 将日期参数(如年、月、日)传入批处理文件中; 使用 echo 命令生成一个日期文件; 使用 copy 命令将日期文件和原始数据文件顺序拼接, 生成具有日期行的数据文件, 并将其存储于服务器上的指定位置。

关于文件拼接的具体用法见程序 1。

2. 文件结构的转换与调整

经过拼接后的数据文件, 虽含有日期参数, 但它是独立记录——行的形式出现的, SQLLOAD 不能直接使用这种日期值, 而必须将行形式日期转换成列形式日期, 使数据文件中每一行的固定位置都具有日期值, 这种方式可供 SQLLOAD 使用。实践证明, 实现这种文件结构转换的简捷方法就是利用 UNIX 中的解释执行程序 AWK。AWK 可以对某一数据文件按一定要求进行处理, 生成一个新的文件。利用 AWK 实现文件结构的转换方法简单、速度快。其具体用法见程序 2 和程序 3。

程序 2 中的 NR 为 AWK 中的关键字, 表示记录号, NR = 1 表示输入文件为第一条记录, NR > 1 表示输入文件为第 2 条记录以后(包括第 2 条记录)的所有记录。\$ 1 表示输入文件的第一个字段, \$ 0 表示输入文件的每

条记录(所有字段)。

3. 数据复盖的控制

在实际装库过程中,常常希望能够有条件的复盖数据库中的原来数据,而 SQLLOAD 只提供了无条件复盖数据的功能。为解决这个问题,我们利用数据库 Triggers 实现有条件的数据复盖。其具体用法见程序 4。

程序 4 的功能是完成表 gtrb-dat 日期相同的数据的复盖。这里的 LNO 是表中的列名,表示记录的顺序, rq 也是表中的列名,表示日期。这是一个行 Triggers, new.lno = 1 表示被装入数据为第一行时,执行 Trigger 体的内容。rq = :new.rq 表示表中的日期值与数据文件的日期值相等。

4. 数据的顺序控制

ORACLE 数据库中的数据存储是无序的,人们通常希望从数据库中提取出的数据是有序的,而且能保持数据文件的顺序。可以利用 SQLLOAD 控制文件中的选项 sequence 将装库的顺序号写入表中的指定列,从而使数据库中的数据顺序化。其具体用法见程序 5。

程序1: rb.bat

```
echo off
cls
if %0%1%2%3==%0%1%2 goto err1
if not exist c:\data\data%3.bas goto err2
if not exist c:\data\gt%3.smb goto err3
if not exist c:\data\ks%1%3%2.dp1 goto err4
if not exist c:\data\ks%3.smb goto err5
if not exist c:\data\jy%3.dat goto err6
choice DATE: %1-%2-%3
if errorlevel 2 goto err7
echo %1%2%3>F:\load\rb\rq.txt
copy c:\data\data%3.bas F:\load\rb\gtrb_d.dat
copy F:\load\rb\rq.txt+c:\data\gt%3.smb F:\load\rb\gtrb_s.dat
copy F:\load\rb\rq.txt+c:\data\ks%1%3%2.dp1 F:\load\rb\ksrb_d.dat
copy F:\load\rb\rq.txt+c:\data\ks%3.smb F:\load\rb\ksrb_s.dat
copy F:\load\rb\rq.txt+c:\data\jy%3.dat F:\load\rb\jyrb_d.dat
echo *****
echo * data of %1-%2-%3 *
echo *****
goto end
:err1
echo ERROR: RB yy mm dd
goto end
:err2
echo ERROR: file not Found c:\data\data%3.bas
goto end
:err3
echo ERROR: file not Found c:\data\gt%3.smb
goto end
:err4
echo ERROR: file not Found c:\data\ks%1%3%2.dp1
goto end
:err5
echo ERROR: file not Found c:\data\ks%3.smb
goto end
:err6
echo ERROR: file not Found c:\data\jy%3.dat
goto end
:err7
echo ERROR: date error
goto end
:end
```

程序2: rb_awk

```
BEGIN {
  FS=" "
  OFS=" "
}
NR==1 {rq=$1 }
NR>1 {print rq,$0}
END {}
```

程序3: rb

```
#####
# /usr/users/user17/ #
# programname: rb #
# program-run: .rb #
# Tian zhenguang #
# 1996-12-06 #
#####
cd /usr/data/load/rb
cp gtrb_d.dat gtrb_d.txt
awk -f rb_awk gtrb_s.dat>gtrb_s.txt
awk -f rb_awk ksrb_d.dat>ksrb_d.txt
awk -f rb_awk ksrb_s.dat>ksrb_s.txt
awk -f rb_awk jyrb_d.dat>jyrb_d.txt
sqlldr rb/rb gtrb_dat data="gtrb_d.txt" silent=header
sqlldr rb/rb gtrb_smb data="gtrb_s.txt" silent=header
sqlldr rb/rb ksrb_dat data="ksrb_d.txt" silent=header
sqlldr rb/rb ksrb_smb data="ksrb_s.txt" silent=header
sqlldr rb/rb jyrb_dat data="jyrb_d.txt" silent=header
cd
#end of rb
```

程序4: gtrb_i.sql

```
create or replace trigger gtrb_i
before insert on gtrb_dat
for each row
when (new.lno=1)
begin
  delete gtrb_dat where rq=:new.rq;
end;
```

程序5: gtrb_dat.ctl

```
load data
append into table gtrb_dat when (20)!='
trailing nullcols
(rq position(1:06) char,
dm position(8:11) char,
gtrb01 position(12:20) decimal external,
gtrb02 position(21:29) decimal external,
gtrb03 position(30:38) decimal external,
gtrb04 position(39:47) decimal external,
gtrb05 position(48:56) decimal external,
gtrb06 position(57:65) decimal external,
gtrb07 position(66:74) decimal external,
gtrb08 position(75:83) decimal external,
gtrb09 position(84:92) decimal external,
gtrb10 position(93:101) decimal external,
gtrb11 position(102:110) decimal external,
gtrb12 position(111:119) decimal external,
gtrb13 position(120:128) decimal external,
lno sequence(1,1))
```

四、结束语

ORACLE7 是目前比较流行的数据库管理系统,基于它的 SQLLOAD 虽具有一定的局限性,但它批量处理数据的强大功能仍不失为一个好的工具软件。本文所介绍的利用 AWK、文件拼接以及数据库 Triggers 来实现参数传递及数据复盖,目的在于更好地使用 SQLLOAD,扬其长,避其短,以求灵活运用之效果。

本文提到的参数是指日期参数,读者在运用过程中不必受此限制,可用同样的方法推广到其他参数。

(来稿时间:1997年3月)