文本数据文件向远程 ORACLE 数据库加载的实现方法

张希 （东南大学自动控制系 210018）

摘要：本文以一实例论述了把一文件系统的数据文本文件自动加载到远程 ORACLE 数据库相应表中的方法和实现的具体步骤。

关键词：远程数据加载 ORACLE 数据库 文本文件

某省公路管理局信息管理网络系统是一个在网络系统，数据库系统支撑下的大型 MIS 系统。MIS 系统主要包括公路养护，文书档案管理，人事。财务管理，领导查询，规费征收等若干个分系统。其中规费征收分系统又分为三级子系统，有的子系统如，稽征所的缴费，车籍档案管理是几年前在 XENIX 下用 COBOL 语言开发的。而现在系统已开始采用 ORACLE 分布式数据库。网络系统架构全部基于 TCP/IP 协议。

本文正是在这种背景下，为保证稽征处能有效的行施职能，采用一种有效可靠的通信，加载手段，使稽征处的车籍档案库始终与稽征所的动态车籍档案保持一致（稽征所的车籍信息是经常变化的）。下面将以车籍档案中的车户档案为例，介绍实现的具体步骤。

1.将变更的车户档案记录从 COBOL 数据文件转换为定长的文本文件。同时在此文件的每条记录前加一位操作符（0－－表示此记录需要删除，1－－表示此记录是新增记录，2－－表示此记录某些项需要修改）。此文本文件即为下面需要加载的数据文件（lchehu.dat）。

2.用 FTP 将文本文件（lchehu.dat）传送至远程稽征处的数据库服务器中。具体的方法是：

·直接用 FTP 命令行来传送。

·如在 UNIX 操作系统之下，也可编写一 shell 文件，如 ccc：

```
open 202.119.13.98
oracle7
cd temp
send lchehu.dat
quit
```

（202.119.13.98 是远程服务器的地址，oracle7 是一个它的用户。temp 是此用户 缺省目录下的一个子目录。）

用 ftp＜ccc 自动发送。

3.编写一个对应文本文件格式的加载控制文件（如 lchehu.ctl 所示），然后用 ORACLE 工具 SQL＊LOAD 将传送来的文本文件（lchehu.dat）中的数据加载到 ORACLE 数据库的一临时表（lchehu-tab）中。

.lchehu.ctl 文件：

```
Load data
infile'1chehu.dat'
replace
into table 1chehu-tab(
m-id position(01:01) integer external,
chehu-id position(02:7) integer external,
chehu-name position(8:37) char,
chehu-addr plsition(38:67) char,
lian-name plsition(68:79) char,
zip-code plsition(80:85) integer external,
zu-phone position(86:93) integer external,
fu-phone position(94:97) integer external,
kanhu-addr position(98:117) char,
zhang-item position(118:130) char,
yang-id position(131) char,
jiao-id position(132) char,
suo-id position(133:135) char)
```

.临时表（lchehu-tab）的结构：

Name	Null?	Type
M-ID		NUMBER(1)
CHEHU-ID		NUMBER(6)
CHEHU-NAME		VARCHAR2(30)
CHEHU-ADDR		VARCHAR2(30)
LIAN-NAME		VARCHAR2(12)
ZIP-CODE		NUMBER(6)
ZU-PHONE		NUMBER(8)
FU-PHONE		NUMBER(4)

KANHU-ADDR	VARCHAR2(20)
ZHANG-ITEM	VARCHAR2(13)
YANG-ID	VARCHAR(1)
JIAO-ID	VARCHAR2(1)
SUO-ID	VARCHAR2(3)

.运行 SQL * LOAD:

sqlload zhang/zx3748 lchehu.ctl

(zhang/zx3748 是 ORACLE 数据库的一个用户名/口令)

4.编写一 PL/SQL 程序(lchehu1.sql),根据临时表(lchehu-tab)第一列操作符(m-id) 的不同,对稽征处车户档案(nche-tab)分别作相应的删除(0),添加(1),修改(2)。

.lch ehu2.sql 文件:

```
DECLARE
n0 lchehu-tab.m-ik%type;
n1 lchehu-tab.chehu-id%type;
n2 lchehu-tab.chehu-name%type;
n3 lchehu-tab.chehu-addr%type;
n4 lchehu-tab.lian-name%type;
n5 lchehu-tab.zip-code%type;
n6 lchehu-tab.zu-phone%type;
n7 lchehu-tab.fu-phone%type;
n8 lchehu-tab.kanhu-addr%type;
n9 lchehu-tab.zhang-item%type;
n10 lchehu-tab.yang-id%type;
n11 lchehu-tab.jiao-id%type;
n12 lchehu-tab.suo-id%type;
m9 ncehu-tab.zhang-item%type;
m10 ncehu-tab.yang-id%type;
m11 ncehu-tab.jiao-id%type;
m12 ncehu-tab.suo-ie%type;
        cursor c1 is
      select * from lchehu-tab
            where m-id=0 or m-id=2;
    cursor c2 is
      select * from lchehu-tab
          where m-id=0 or m-id=2;
begin
    open c1;
        loop
    fetch c1 into n0, n12, n1, n2, n3, n4, n5, n6, n7, n8,
n9, n10, n11;
        exit when c1%notfound;
delete from ncehu-tab
          where cehu-id=n1;
        end loop;
        close c1;
        commit;
open c2;
         loop
    fetch c2 into n0, n12, n1, n2, n3, n4, n5, n6, n7, n8,
n9, n10, n11;
        exit when c2%notfound;
  m9:=to-numbwr(n9);
  m10:=to-numbwr(n10);
  m11:=to-number(n11);
  m12:=to-number(n12);
  insert into ncehu-tab(suo-id, cehu-id, cehu-name, cehu-ad-
dr, lian-name, zip-code, zu-phone, fu-phone, kanhu-addr,
zhang-item, yang-id, jian-id)
values(m12, n1, n2, n3, n4, n5, n6, n7, n8, m9, m10, m11);
  end loop;
    close c2;
    commit;
end;
```

.ncehu-tab 的结构:

Name	Null?	Type
SUO-ID	NOT NULL	NUMBER(3)
CEHU-ID	NOT NULL	NUMBER(6)
CEHU-NAME		VARCHAR2(30)
CEHU-ADDR		VARCHAR2(30)
LIAN-NAME		VARCHAR2(12)
ZIP-CODE		NUMBER(6)
ZU-PHONE		NUMBER(8)
FU-PHONE		NUMBER(4)
KANHU-ADDR		VARCHAR2(20)
ZHANG-ID		NUMBER(13)
YANG-ID		NUMBER(1)
YANG-NAME		VARCHAR2(10)
JIAO-ID		NUMBER(1)
JIAO-NAME		VARCHAR2(10)

5.结束语

整个过程中除了程序设计的正确外,数据传输的准确无误也是极其重要的.建议采用基于 TCP/IP 的 FTP,以保证足够的可靠性。本文的方法适合任何文本数据文件向 ORACLE 数据库的加载。