

实现 DBASE Ⅲ 数据库 DBF 文件向 ORACLE 数据库表的转换

二炮第四研究所 汤淑春

北京科技大学 万 静

摘要 本文详述了从 DBASE Ⅲ 的 DBF 文件到 ORACLE 数据库表的转换,它既可用于 ORACLE 数据库中数据录入,也可用于将数据从 DBASE Ⅲ 数据库 DBF 文件中移植到 ORACLE 数据库表中。

一、基本思路

先用 DBASE 的 COPY 语句把数据库 DBF 文件中的数据送入相应的文本文件中,在 ORACLE 数据库的 SQL*PLUS 语言环境下建立和原 DBF 文件相同结构的表,用 ORACLE 数据加载器 (ODL) 把文本文件中的数据加载到表中,但由于 ODL 只能加载格式 A 的文本文件而不能加载格式 B (大于 80 列的记录) 的文本文件,因此如果由 DBF 文件所产生的文本文件格式如格式 B,则不能用 ODL 直接加载。

格式 A	格式 B
记录1	记录1
记录2	..记录1的继续
记录3	记录2
记录4	..记录2的继续
.	.
.	.
.	.

解决的办法是:控制所产生的文本文件的格式使其如格式 A,分别加载到多个 ORACLE 数据表中,然后利用 SQL*PLUS 工具将数据从多个表中抽取出,合成一个字符型的表,在 SQL*PLUS 环境下,将此字符型的表转换成最终所要求的表。

二、具体实例

有二个 DBASE Ⅲ 数据文件 A. DBF, B. DBF, 其结构分别如格式 C 和格式 D。格式 C 内容如下:

Structure for database : A. dbf

Number of data records : 537

Data of last update : 01/02/90

Field	Field name	Type	width	Dec
1	IMC	Character	10	
2	FSRJ	Character	16	
3	MC	Character	20	
4	ZG	Character	4	
5	FS	Character	4	
6	IZGR	Number	5	2
7	ZGR	Character	6	
8	GB	Character	4	
9	ISL	Character	3	

** TOTAL **

72

格式 D 内容如下:

Structure for database : B. dbf

Number of data records : 640

Data of last update : 01/02/90

Field	Field name	Type	width	Dec
1	GB	Character	4	
2	MC	Character	18	
3	NF	Number	8	4
4	SF	Number	8	4
5	MF	Number	8	4
6	FF	Character	16	
7	ZF	Number	8	4
8	JFNF	Character	14	

** TOTAL **

84

对 A. DBA 来说,在 DBASE 环境下用 COPY 命令生成文本文件 A. TXT,在 SQL*PLUS 环境下建立和 A. DBA 相同结构的表,用控制文件 A. K 可

将 A. TXT 中的数据直接加载到表 A 上。

A. K 文件内容如下:

```
DEFINE RECORD ZZ AS
IMC (CHAR (10)), FSRJ (CHAR (16)),
MC (CHAR (20)), ZG (CHAR (4)),
FS (CHAR (4)), IZGR (CHAR (5)),
ZGR (CHAR (6)), GB (CHAR (4)),
ISL (CHAR (3));
DEFINE SOURCE FILE
FROM A1. TXT
LENGTH 80
CONTAINNING ZZ;
FOR EACH RECORD
INSERT INTO A (IMC, FSRJ, MC, ZG, FS, IZGR, ZGR,
GB, ISL)
VALUES (IMC, FSRJ, MC, ZG, FS, IZGR, ZGR, GB,
ISL)
NEXT RECORD
```

而对 B. DBF, 则不能直接加载, 应先用

```
COPY TO B1 FIELDS GB, MC, NF, SF SDF
COPY TO B2 FIELDS MF, FF, ZF, JFNF, SDF
```

分别生成文本文件 B1. TXT, B2. TXT, 然后在 SQL 环境下分另建立和 B. DBF、B1. TXT、B2. TXT 结构相同的表 B、B1、B2。分别用 B1. K、B2. K 控制文件将 B1. TXT、B2. TXT 中的数据加载到表 B1、B2 中。

B1. K 文件内容如下:

```
DEFINE RECORD ZZ AS
GB (CHAR (4)), MC (CHAR (18)),
NF (CHAR (8)), SF (CHAR (8));
DEFINE SOURCE FILE
FROM B1. TXT
LENGTH 80
CONTAINNING ZZ;
FOR EACH RECORD
INSERT B1 (GB, MC, NF, SF)
VALUES (GB, MC, NF, SF)
NEXT RECORD
```

B2. K 文件内容如下:

```
DEFINE RECORD ZZ AS
MF (CHAR (8)), FF (CHAR (16)),
ZF (CHAR (8)), JFNF (CHAR (14));
DEFINE SOURCE FILE
B2. TXT
LENGTH 80
```

```
CONTAINNING ZZ;
FOR EACH RECORD
INSERT B1 (GB, MC, NF, SF)
VALUES (GB, MC, NF, SF)
NEXT RECORD
```

B2. K 文件内容如下:

```
DEFINE RECORD ZZ AS
MF (CHAR (8)), FF (CHAR (16)),
ZF (CHAR (8)), JFNF (CHAR (14));
DEFINE SOURCE FILE
B2. TXT
LENGTH 80
CONTAINNING ZZ;
FOR RACH RECORD
INSERT B2 (MF, FF, ZF, JFNF)
VALUES (MF, FF, ZF, JFNF)
NEXT RECORD
```

在 SQL * CALC 下, 将数据从表 B1、B2 中抽取出来, 用 /ORACLE/TABLE 命令生成表 B3, 再回到 SQL * PLUS 状态下, 用 B4. SQL 命令将 B3 转换到所要求的表 B 上。

B4. SQL 文件内容如下:

```
INSERT INTO B (GB, MC, NF, SF, MF, FF, ZF,
JFNF)
SELECT GB, MC, NF, SF,
MF, FF, ZF, JFNF FROM B3
```

三、应 用

本文论述的方法可以用于以下二个方面:

(1) ORACLE 数据库中数据的录入。

在数据的录入过程中, 用户可根据其数据库的具体结构, 用汉化 PE II 把数据编辑成若干个文本文件。然后利用 ODL 进行加载, 这样做的优点是既使数据输入有误, 也便于检查、修改。

(2) 将数据从 DBASE III 数据库中移植到 ORACLE 数据库中。

随着数据应形技术的不断深入, 越来越多的人开始使用 ORACLE 数据库管理系统, 而数据录入的工作虽然简单, 但工作量大, 上述方法用于移植则省时省力。

参考文献:

《ORACLE 关系数据库管理系统使用说明》国家统计局计算中心。