

一种异构数据库数据的直接转换技术

梁允荣 杨 茜 (北京理工大学 100081)

摘要:本文阐述了如何利用 PowerBuilder 与数据库系统连接和可视化编程的强大功能,实现了异种数据库的直接转换方法。该方法为用户提供了可视化集成界面,完成了分布在网络不同结点上的目前常用几种数据库(ORACLE、SYBASE、INFORMIX、MS SQL Server、SQL anywhere 和 FoxPro)的两两互相转换。

关键词:异构数据库 数据转换 可视化

一、前言

当前,在计算机应用领域中,数据库的应用非常广泛,而且应用领域还在不断地扩展,由于计算机发展的历史和网络飞速发展的现状,分布式数据的管理和访问就成为数据库技术必须解决的问题。由于一个事物所涉及的数据可能分布在多个结点上,就要求数据库系统具备一个优化的分布策略。由此引发出安全问题、并发控制问题、数据分布透明性问题、分布式查询、分布式更新以及异种数据库数据转换和共享等等问题。

数据库异构问题概括起来包括三个部分的异构:

1. 硬件的异构,指不同的计算机系统,甚至不同的网络系统的互连;
2. 操作系统的异构;
3. 数据库管理系统(DBMS)的异构。

为解决异种数据库的共享,必须先解决异种数据库的转换问题,在这种情况下,一些著名的数据库公司纷纷扩充产品的性能,以期支持异种数据库之间的数据转换,然而它们支持的只是单向的由外向内的转换,没有能够解决连网环境下分布数据库的转换问题,这在技术上有一定复杂性,而且目前还很难用一个通用的 DBMS 来解决这样的问题。因此研究异种数据库在集成环境下直接进行转换的方法势在必行。

二、异构型数据库数据转换设计思想

我们在前一个版本实现数据转换方法时,由于缺少工具软件的支持,是使用中间数据库方法,即在实现两个具体数据库之间的数据转换时,要依据关系定义,字段定义,从源数据库中读出数据通过中间数据库灌入到目的数据库中如图 1 所示。

这种方法在实现过程中难度大,但所需转换模块最

少,可扩展性强,缺点是转换精度受影响,转换时间长。

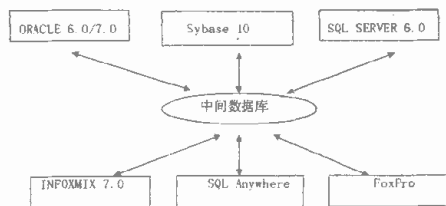


图 1 利用中间数据库的转换图

为了克服上述缺点,在新的版本中,借助了 Power Builder 与数据库连接的强大功能,通过设置源数据库与目的数据库两个不同的传送变量,同时连接两个数据库,实现异种数据库之间的直接转换。这样由于转换模块之间结构相同,只要把数据类型的匹配关系选择好,就可以完成转换。这种方法在现有的应用数量下扩展起来也比较容易,其转换速度和精度却大大提高了。如图 2 所示。

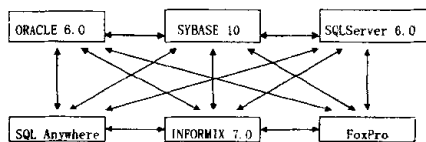


图 2 直接转换的转换图

三、异种数据库之间数据的直接转换

1. 不同数据库系统之间的数据类型分析

本系统所进行数据转换的不同 DBMS 系统有 ORACLE(Unix)、SYBASE(Windows NT)、INFORMIX(Win-

windows NT、MS SQL Server (windows NT) \ SQL AnyWhere 和 FoxPro2.5 的特点和数据类型。为实现它们两两之间的数据转换,必须逐一分析它们自身的数据类型特征,然后将它们之间相同或相近数据类型予以一一对应,为此必须对上述几种数据库系统的数据类型特点做分析比较。在每个 DBMS 中都定义了自己系统的数据类型,但不论数据类型在各个系统中如何变化,其功能都满足用户的数据处理的基本要求,如数值型:包括整型、实型、浮点型、双精度型等;字符型:包括定长、变长等;日期型:包括年月日和小时分秒等;长字符型:包括文本类型;还有钱币型。随着数据库系统的不断发展和版本的不断升级数据类型的种类也增多了,如超文本型和二进制型处理多媒体和大文本的数据类型也出现了。这些带有共性的东西,给系统间的数据转换带来了可能和方便,其自身定义与扩充互相之间的区别,也给系统间的数据转换带来了许多困难,在转换中重要的事情是找出其间的对应关系。下面列出各种数据库系统与其他数据库类型的对应关系表。

(1)SQL Any Where:见表1。

表1 SQL Any Where 与其他数据库数据类型的对应关系

源数据库数据类型	目的数据库数据类型				
	ORACLE	SYBASE	SQL Server	INFORMIX	FoxPro
SQLAnyWhere	ORACLE	SYBASE	SQL Server	INFORMIX	FoxPro
char (n)	CHAR (n)	char (n)	char (n)	archar (n)	C (n)
varchar (n)	CHAR (n)	varchar (n)	varchar (n)	varchar (n)	C (n)
numeric (n, m)	NUMBER	numeric	numeric	decimal	N (n, m)
integer	NUMBER	int	int	integer	N (10, 0)
smallint	NUMBER	smallint	smallint	smallint	N (5, 0)
double	NUMBER	double precision	double precision	float	N (15, 8)
binary	LONG RAW	binary	binary	byte	M
long binary	LONG RAW	image	image	byte	M
date	DATE	datetime	datetime	date	D
time	DATE	datetime	datetime	time	D
timestamp	DATE	timestamp	timestamp	datetime	D
float	NUMBER	float	float	real	N (15, 8)
longvarchar	TEXT	text	text	text	M
tinyint	NUMBER	tinyint	tinyint	smallint	N (3, 0)

- (2)SYBASE:见表2。
- (3)ORACLE:见表3。
- (4)MS SQL Server。
- (5)Informix:见表4。
- (6)FoxPro:见表5。

表2 SYBASE 与其他数据库数据类型的对应关系

源数据库数据类型	目的数据库数据类型				
	ORACLE	SQLAny Where	SQLServer	INFORMIX	FoxPro
SYBASE	ORACLE	SQLAny Where	SQLServer	INFORMIX	FoxPro
char (n)	CHAR (n)	char (n)	char (n)	char (n)	C (n)
varchar (n)	CHAR (n)	varchar (n)	varchar (n)	varchar (n)	C (n)
nchar (n)	CHAR (n)	char (n)	char (n)	char (n)	C (n)
nvarchar (n)	CHAR (n)	varchar (n)	varchar (n)	varchar (n)	C (n)
numeric (n, m)	NUMBER (n, m)	numeric (n, m)	numeric (n, m)	decimal (n, m)	N (n, m)
int	NUMBER	integer	int	integer	N (10, 0)
smallint	NUMBER	smallint	smallint	smallint	N (5, 0)
tinyint	NUMBER	tinyint	tinyint	tinyint	N (3, 0)
double	NUMBER	double	float	float	N (15, 5)
precision (15, 5)	NUMBER (15, 5)	double	float	float	N (15, 5)
money	NUMBER (19, 4)	numeric (19, 4)	money	decimal (19, 4)	N (15, 4)
small money	NUMBER (10, 4)	numeric (10, 4)	small money	decimal (10, 4)	N (10, 4)
text	LONG	long varchar	text	text	M
binary	RAW	binary	binary	byte	M
long binary	LONGRAW	long binary	long binary	byte	M
varbinary	RAW	binary	varbinary	byte	M
image	LONG RAW	long binary	image	byte	M
datetime	DATE	datetime	datetime	datetime	D
small datetime	DATE	datetime		time	D
timestamp	DATE	timestamp		datetime	D
float	NUMBER	float		real	N (15, 5)
real	NUMBER	float	real	float	M
identity	NUMBER	integer	identity	integer	N (10, 0)

表3 ORACLE 与其他数据库数据类型的对应关系

源数据库数据类型	目的数据库数据类型					
	ORACLE	SQLAnyWhere	SYBASE	SQL Server	INFORMIX	FoxPro
ORACLE	ORACLE	SQLAnyWhere	SYBASE	SQL Server	INFORMIX	FoxPro
CHAR (n)	CHAR (n)	char (n)	char (n)	char	char (n)	C (n)
NUMBER (n, m)	numeric (n, m)	numeric (n, m)	numeric (n, m)	numeric (n, m)	decimal (n, m)	N (n, m)
DATE	datetime	numeric		datetime	D	
LONG	longvarchar	text	text	text	M	

2. 异构数据库数据直接转换过程

为了实现数据转换,我们采用 PowerBuilder 与数据库连接极其可视化编程的强大功能,实现了两两数据库的转换。其转换过程如下:

- (1)在可视化界面下接收源数据库和目的数据库名和源数据库表和目的数据库表名。
- (2)设置两个不同的传送变量,同时建立网络不同结点上的两个异构数据库连接。
- (3)从网络结点上的源数据库的系统目录表中得到源表的数据结构信息。
- (4)查询源数据库与目的数据库数据类型的对应关系,生成目标表的结构信息。

(5)调用 create 语句,在目标数据库结点上建立目标表结构。

(6)再通过在源数据库中设置光标,逐行读取源表记录,调用 insert 语句,写入目的数据库的表中,直到读取最后一个记录。

(7)在界面下可显示源数据库与目的数据库的表结构和表内容,提供检查手段。

(8)可连续重复执行(1)-(7)的步骤。

表 4 Informix 与其它数据库数据类型的对应关系

源数据库数据类型	目的数据库数据类型				
	ORACLE	SYBASE	SQL Server	SQL Anywhere	FoxPro
char(n)	CHAR(n)	char(n)	char(n)	char(n)	C(n)
varchar(n)	CHAR(n)	varchar(n)	varchar(n)	varchar(n)	C(n)
N(n,m)	NUMBER	decimal(n,m)	decimal(n,m)	numeric(n,m)	
integer	NUMBER	int	int	integer	N(10,0)
smallint	NUMBER	smallint	smallint	smallint	N(5,0)
real	NUMBER	real	real	float	N(15,5)
float	NUMBER	double precision	double precision	double	N(15,5)
text	LONG	text	text	longvarchar	M
byte	LONGRAW	binary	binary	binary	M
date	DATE	datetime	datetime	date	D
time	DATE	datetime	datetime	time	D
datetime	DATE	datetime	datetime	datetime	D
money	NUMBER	decimal(16,4)	decimal(16,4)	numeric(16,4)	N(16,4)
serial	NUMBER	int	int	integer	N(3,0)

表 5 FoxPro 与其他数据库数据类型的对应关系

源数据库数据类型	目的数据库数据类型				
	ORACLE	SYBASE	SQL Server	INFORMIX	SQLAnywhere
FoxPro					
C(n)	CHAR(n)	char(n)	char(n)	char(n)	char(n)
NUMBER(n,m)	NUMBER(n,m)	numeric(n,m)	numeric(n,m)	decimal(n,m)	numeric(n,m)
D	DATE	datetime	datetime	datetime	date
L	CHAR(1)	char(1)	char(1)	char(1)	char(1)
M	NUMBER	text	text	text	longvarchar

3. 在实现数据库直接转换的几个关键技术

(1)用户动态的选择源、目的数据库保证任意两两数据库间转换,这种连接方式是通过配置文件 database.ini 来实现的。

(2)用户可以动态的选择数据库中的表,为了得到表结构信息,使用了 Data Windows 按该表名对应的几个系统目录表进行查询,并使用 Data Store 接受查询结果,在实施中的关键在于不同的 DBMS 中系统目录表不同,有时为了得到一个源表的信息,可能要检索几个系统目录

表才能得到。

如:在 Sybase 检索列的信息语句为:

```
SELECT syscolumns. name , systypes. Name ,
syscolumns. Length , syscolumns. scale , syscolumns. Prec
, syscolumns. colid
FROM syscolumns , sysobjects , systypes
WHERE ( syscolumns. id = sysobjects .id ) and (
syscolumns. usertype = systypes. usertype ) and ( ( sysob-
jects. name = ;tab-name ) )
```

(3)被转换的所有数据类型中,日期、时间类型在不同的数据库系统中差别比较大,所以转换起来比较困难。

用 PowerBuilder 的函数 GetDynamicDate()、GetDynamicDatetime()、GetDynamicTime()可以读出不同源数据库的值,再向目的数据库插入值时,就需要使用不同目的数据库的时间转换函数将该数据转换为数据库可以接受的形式。

ORACLE 所用的函数为 TO-DATE(要转换的时间串,所需的格式如:”DD-MON-YY HH:MI A.M.”)。

INFORMIX 的函数是:DATE or DATETIME(要转换的时间串)所需的格式。如 YEAR TO SECOND。

SQL Any Where 中用的函数为:DATE, DATE-TIME, TIME。

Sybase 和 SQL Server 的函数是:DATETIME。

(4)数据灌入的方法因数据库的特点不同而不同。

数据灌入的方法有两种:一种是用嵌入 SQL 语句 INSERT INTO 将记录一条一条加入目的表中;另一种是利用 PB5 中的 DATASTORE 实现数据从一个表到另一个表的数据传递。这两种方法的主要差别是:

前者是一条记录一条记录的插入,效率比较低,但插入记录时对表和记录没有任何要求;后者是将要传递的记录都放在源表的缓冲区中,同时开辟目的表的缓冲区,然后将数据从源缓冲区拷贝到目的缓冲区中,这样做效率高(尤其对于大数据量效果非常明显),但要求目的表必须有关键字,这样才能将拷贝到目的缓冲区的数据用 UPDATE 来修改目的表。而有些数据库,如 ORACLE 并未要求表有关键字,甚至它允许有相同的记录存在。

参考文献

- [1] 俞盘祥. ORACLE 数据库系统基础. 北京,清华大学出版社,1995年11月
- [2] “异构数据库的转换和联合使用”课题组. 异构数据库的转换和联合使用技术总结. 北京,北京理工大学计算机中心,1995年7月

(来稿时间:1998年5月)