



1 引言

LOTUS NOTES 是群件的典型产品,

除了具有办公自动化系统所需要的集成通信、协同操作和协同运行的基本功能外,还有大型文档型数据库(也即其中数据的存储是以文档的形式而不是N维数据列表)作为后台支持,为信息管理模式提供了一个出色的平台,为企业的办公室自动化OA提供了一个很好的计算机环境,尤其是那些需要对大量非关系化数据进行处理、有大量协同操作和通信业务的企业,例如企业中的审批文件流程、流程控制、安全性要求较高的文档处理,NOTES无疑是个很好的选择。

随着计算机的普及,办公自动化和无纸办公已经日益提到各企业的议事日程上来,而普通的信息管理系统大多以大型关系型数据库为后台支持,如ORACLE、SYBASE、SQL SERVER等等,这对于数据的计算和统计要求较高的企业是非常实用的,但对于那些日常办公的流程和通信要求较高,而且非关系化的数据较多的企业则有诸多局限性。从这一方面来说,NOTES就很好地解决了这一问题。NOTES的数据库成为文档型数据库,对于非关系型数据的处理非常方便,存取也很方便直观。从另一方面来说,由于企业对数据处理的要求是多方面的,那么就有可能存在同时处理大量关系型数据和非关系型数据,因此出现了关系型数据库与文档型数据库并存的局面,其中还有可能一部分数据需要交互,以此来完成整个工作流的处理过程,例如在企业 and 政府部门中,除了完成日常的工作处理流程,还有大量的待处理工作数据。那么,在NOTES数据库与大型关系型数据库之间建立联系,以便于数据交互就变得非常重要。本文将结合实例对于从关系数据库到文档数据库以及与之相反的过程这两方面来探讨数据交换的流程。



2 数据交互的过程

在使用 LOTUS NOTES 与关系型数据

LOTUS NOTES 与关系型数据库的数据交换



郝春艳 (武汉大学信息管理学院 430072)

摘要: LOTUS NOTES 是办公自动化的软件设计的常用工具,本文对关于企业中大型关系型数据库和LOTUS NOTES的文档型数据库的数据交互进行的细致的探讨,并结合一定的案例给予了解决方案。

库进行数据交互之前,首先有一些准备工作。由于 LOTUS NOTES 使用的是文档数据库,数据保存的形式都是文档型的,这种方式和关系数据库有很大的区别,因为关系数据库都是以有条理的二维数据表的形式存储数据,所以两者之间的数据转换还是不太方便的,但是借助于 LOTUS NOTES 提供的扩展类库,也就是 LOTUS NOTES 为了方便与其他程序的数据接口而设计函数库,再利用SQL语句就可以比较顺利进行数据转化,但在网络上相对来说速度是比较慢的。首先,要定义 ODBC 数据源,如果是 SQL SERVER 7.0 的数据源,则必须在 ODBC 的系统数据源中定义,然后在 NOTES 的应用程序中引入扩展类,在 Global 对象中放入 Uselsx "lsxodbc" 语句,其作用就是使用系统定义的外部类应用程序,通过加载 ODBC LSX 使 ODBC 类在编程环境中有效,最后使用 ODBCConnection 的成员函数 Connectto 对此数据源进行连接,通过它进行数据的提取和保存,更新NOTES数据库和关系型数据库的数据。下面我们就用一定实例来分别说明如何将文档数据库的数据转换成关系数据库的数据,如何将关系数据库的数据转换成文档。



3 从关系数据库到NOTES文档数据库的数据交换

用NOTES扫描关系型数据库的数据,生成NOTES的待处理文档,利用关系数据库的二维数据,生成在工作流中待处理文档格式。这种情况通常是在企业或政府部门中,对日常处理数据

时,由于有一些数据需要审查,所以必须将关系数据库的数据返回处理的工作流中。这样的任务是频繁而必要的,在本程序中可以直接扫描数据库生成要处理的文档,其中studyform时声称此文档的表单名称,而form是文档的缺省条目,表示显示此文档时采用什么样的缺省表单。生成空白文档后,给文档的条目赋值,并利用SAVE函数,保存文档。在本程序中引用了条目的概念,也就是显示该文档的表单原先所定义的阈,存储成文档后即称为条目,但是文档中的条目是可以动态生成的,不受已定义的表单的约束,除非你在表单中定义了相应的阈,否则所生成的条目在浏览时是不显示出来的,但可以当作文档的一个属性在程序中引用。具体程序如下:

```
// 首先声明局部变量
// 先定义关于连接数据库信息的变量
DIM session As NEW NOTESSESSION
DIM db AS NOTESDATABASE
DIM con AS NEW ODBCConnection
DIM qry AS NEW ODBCQUERY
DIM result AS NEW ODBCRESULTSET
DIM doc AS NOTESDOCUMENT
// 建立与数据源 study 的连接 (study 是事先
在 ODBC 数据源中定义的)
con.CONNECTTO ("study")
SET qry.CONNECTION=con
SET result.QUERY=query
// 设定数据库与文档的当前值
SET db=session.CURRENTDATABASE
// 设定对数据库进行操作的 SQL 语句
```

```

qry.SQL=" SELECT code FROM studys"
//执行SQL语句
result.EXECUTE
//建立新文档,并用数据库中的值初始化新文档中的条目值,保存文档
IF con.CONNECT THEN
CALL doc=db.COMPOSEDOCUMENT
doc.FORM=" studform"
//为新文档的交互界面定义已经存在的表格形式
doc.code=result.getvalue("code")
//为code赋值
doc.SAVE(TRUE,TRUE)
END IF
//操作完之后关闭并断开与数据源的连接
result.CLOSE(DB-CLOSE)
con.DISCONNECT

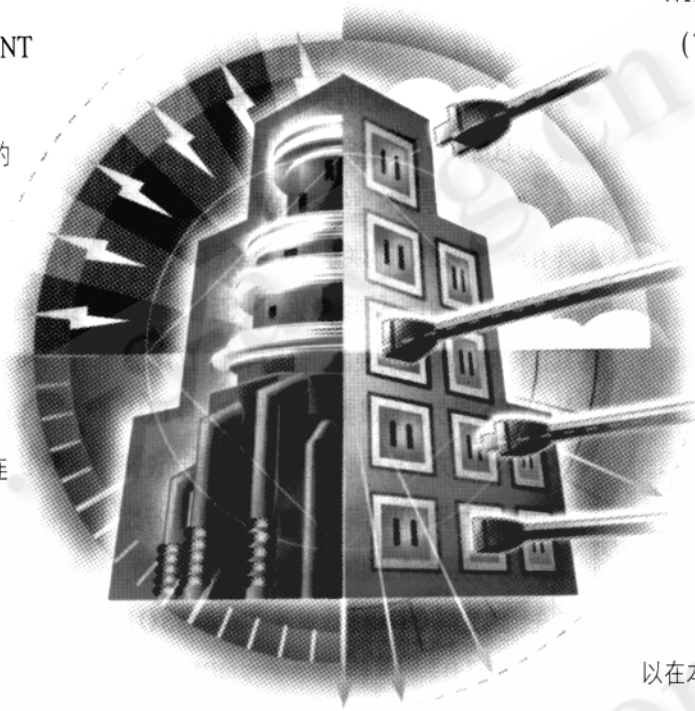
```



4 从NOTES文档数据库到关系型数据库的数据交换

用NOTES更新关系型数据库的数据,就是将文档数据转换成关系型数据,此类的操作包括添加一个新的记录Addrow,产出一个记录Deleterow,更新一个记录Updaterow等。这样的转换相对于相反的方向的转换来说是比较繁琐的,尤其是对于SQL语句的书写容易出错。在本程序中,假如code,name为表studys中的两个字段,同时也是uidoc文档中的两个阈值,本程序是将表中的两个字段进行更新,在嵌入的SQL语句中可以引用任何合法的SQL语句,但是需要注意的是在执行WHERE条件时,不能直接进行字符串的引用,而是需要使用SETPARAMETER函数,或者用字符串连接的方式进行,这其中尤其要注意的是WHERE条件中的单引号,要当作字符串来处理。在本程序中还要注意的SetValue函数,当关系数据库的

数据被成功修改以后,此函数将返回TRUE,否则返回FALSE。在编程环境中写下如下语句:
//首先声明局部变量
DIM session As NEW NOTESSESSION
DIM db AS NOTESDATABASE



```

DIM
uidoc AS NOTESUIDOCUMENT
DIM work AS NEW NOTESUIWORKSPACE
DIM con AS NEW ODBCCONNECTION
DIM qry AS NEW ODBCQUERY
DIM result AS NEW ODBCRESULTSET
//建立与数据源study的连接,数据源必须是ODBC系统数据源中事先设定好的)
con.CONNECTTO("study")
SET qry.CONNECTION=con
SET result.QUERY=query
//设定数据库与文档的当前值

```

```

SET db=session.CURRENTDATABASE
SET uidoc=work.CURRENTDOCUMENT
//设定对数据库进行操作的SQL语句
qry.SQL=" UPDATE studys SET code=" +
uidoc.fieldgettext("code")+"" +
WHERE name="" +uidoc.fieldgettext
("name")+""
//执行SQL语句
result.EXECUTE
IF con.CONNECT THEN
CALL result.setvalue("code",
uidoc.fieldgettext("code"))
END IF
//操作完之后关闭并断开与数据源的连接
result.CLOSE(DB-CLOSE)
con.DISCONNECT

```

当然除了这些操作之外,NOTES还可以在本系统环境中对关系型数据库进行更多的操作,例如增加表、删除表等。只需将相应的位置用相应的SQL语句替换即可,可以用CREATE TABLE来创建表,用INSERT、DELETE、UPDATE来更改表的内容。



5 结束语

本文从两个方面介绍了在文档数据库与关系数据库之间的数据交换,但是笔者认为从应用的角度来说,除非必要的数据库交换,否则我们并不赞同利用NOTES对关系型数据库进行大量的数据处理,因为这并不是NOTES的优势所在。 ■

LOTUS NOTES 与关系型数据库的数据交换



郝春艳 (武汉大学信息管理学院 430072)

参考文献

1 罗琳、陈怡之、陈斌、张宏林等编著,《Notes编程疑难详解》,人民邮电出版社。