

MS SQL SERVER

数据库触发器技术的实际应用

湖南大学电气与信息工程学院 魏育成 王耀南 资刚

SQL SERVER 数据库是当今生产信息管理系统开发中有代表性的大型网络数据库之一。本文从一个开发实例出发, 讨论了其触发器技术的基本概念和具体应用, 展现了触发器在维护数据完整性和有效性方面的强大功能。



概 述

Microsoft 公司提供的 MS SQL SERVER 是当今流行的网络数据库之一, 它性能可靠, 数据管理功能强, 与 NT 集成更是如虎添翼。在诸多 SQL SERVER 提供的数据处理技术中, 触发器 (TRIGGER) 技术以其出色的灵活性和可靠性成为数据库工程师的有力工具之一。

触发器是一种特殊类型的存储过程, 它能自动执行而成为一个 SQL 修改语句的一部分。一个触发器基于一个表来创建并和一个或多个数据修改操作 (INSERT、UPDATE、DELETE) 相关联, 当出现一次这样的操作时, 触发器会被自动激活, 进行相关设定的数据处理。在触发器中可以使用 SQL 语句, 还可以使用流控制语句, 因此可以在触发器中进行判断和分析, 根据不同的条件采取不同的行动。

由于触发器具有以上特点, 在实际开发中, 常常得到以下两个方面的应用: 其一, 维护数据的完整性。在一些

相互关联的数据表中, 一张表经常要参照另一张表的记录内容, 而且单个数据表的不同字段之间也会相互依赖, 即一个字段是另一个字段的衍生。象这种被成为级联式的操作就可以使用触发器来完成。比如, 自动备份某生产设备的检修记录; 还有, 当设备的累计运行时间和设计寿命改变时, 自动计算累计运行百分比等。其二, 保证数据的有效性和安全性。在对数据表进行相关操作时, 难免会产生非法数据 (比如生产设备的累计运行时间超过其设计寿命), 出现这种情况, 触发器就会进行判断, 自动拒绝此类操作。

笔者近期完成了一个大型矿山 CIMS 系统的开发工作, 本系统采用 C/S 结构模式, 服务端数据管理采用 MS SQL SERVER 7.0, 客户端开发工具采用中文版 VB 6.0, 两者之间利用 ODBC 连接。在实际开发过程中, 曾利用触发器技术成功解决了一些生产数据的管理问题, 下面就以一个较典型的“生产设备管理系统”为例, 简单介绍一

下触发器在维护数据完整性和有效性方面的应用。

概 念

在我们讨论触发器技术的应用之前,有必要向大家介绍几个相关的基本概念。

(1)触发器的执行过程。在一张数据表上最多只能创建三个触发器(UPDATE TRIGGER、DELETE TRIGGER、INSERT TRIGGER),其中以UPDATE触发器的操作过程最为复杂,如果用户发出更新一条记录的操作,则SQL SERVER执行以下过程:

① 在事务日志中记录一个begin trans标志并开始一个事务(BEGIN TRANSACTION)。

② 在事务日志中写入此数据表中受到影响的行的旧记录内容,并从数据表中删除旧记录。

③ 在事务日志中写入此数据表中受到影响的行的新记录内容(更新后的内容),并往数据表里插入修改过的新记录。

④ 执行UPDATE触发器。

⑤ 在事务日志中记录一个commit trans标志提交一个事务(COMMIT TRANSACTION),将所有的修改写入磁盘。

从上面的操作可知,UPDATE操作在事务处理上实际是先一个DELETE操作而后执行一个INSERT操作。而数据库管理系统(DBMS)将整个操作过程视为一个事务整体,这样,我们就可以在执行UPDATE触发器时进行检测,如果发现不合理的操作,就可以回滚事务(ROLLBACK TRANSACTION),从而不提交修改并提醒用户注意。

(2)两个特殊的表——INSERTED表和DELETED表

触发器工作引入两个特殊的表——inserted表和deleted表,这两张表实际上是事务日志的视图(VIEW),所以它与创建了触发器的表(以下简称“触发器表”)具有相同的结构。

针对insert、update、delete三种操作,inserted表和deleted表的内容如表1:

参照我们前面介绍的update触发器的执行过程,可以知道,分别在第②步和第③步,触发器表中受到影响的旧行和新行已被拷贝到事务日志之中,这也就提供了inserted表和deleted表的记录来源。

(3)特殊的语法功能——update()函数

使用一个带update()函数的if语句,可以使我们

语句	Inserted表的记录	Deleted表的记录
insert	触发器表中添加的行	空
update	触发器表中修改的行(新行)	触发器表中修改前的行(旧行)
delete	空	触发器表中删除的行

表1 Inserted表、deleted表在不同操作下的不同内容

测试数据表中某一特定列是否被修改,从而进行相应的操作。此函数只能在insert或update触发器中使用,其基本语法如下:

```
if update(column_name)
```

.....

(4)向VB客户端返回一个消息——raiserror语句

正如我们前面所介绍的,如果update操作有误,我们可以在update触发器中回卷整个事务,而不提交修改。一般为了使用户知道操作错误,我们常在rollback操作之后利用raiserror语句向客户端发送一个错误提示,VB接到这个消息之后,就能中断程序正常执行,掉入错误陷阱中执行相应出错处理,这就使得在客户端应用程序中编写一般错误处理程序变得容易了。Raiserror语句的语法结构如下:

```
RAISERROR (msg_id , severity, state)
```

其中:msg-id为用户在系统表sysmessages中自定义的错误号;

severity为用户定义的错误安全级别;

state为表示出错状态的一个整数。

在使用raiserror语句之前,需先用系统存储过程sp-addmessages向SQL SEVER库系统表中申请一个错误。sp-addmessages的语法结构如下:

```
sp-addmessages msg-id, severity,text of message
```

如:sp-addmessages 50009,16,“该设备超时运行,需要检修!”

```
raiserror (50009,16,10)
```

注:申请完自定义错误之后,我们可以在SQL SERVER 7.0提供的Enterprise Manager中Server菜单的Message选项中查询到此错误。

实 现

有了以上几个重要概念,就不难理解下面这个实际例子。

假设SQL SERVER库中存在两张数据表——设备

表 (Device Table) 和设备检修记录表 (Old Device Table), 分别存放工厂的所有设备记录和设备的历史检修记录。表间关系如图 1 所示:

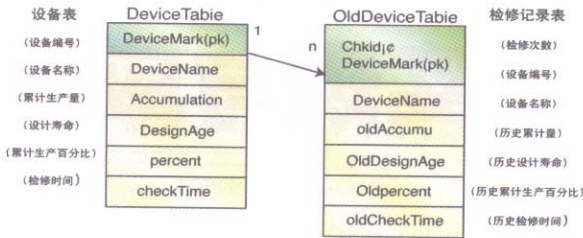


图 1 设备表和和历史检修记录表的关联关系

说明:

DeviceTable表以DeviceMark为主关键字 (Primary Key), OldDeviceTable表以 CheckId 和 DeviceMark 为双主关键字 (Double Keys)。两者之间通过DeviceMark形成一对多的PK/FK引用关系, 这样既可以使每张表中每行记录各不相同, 又能保证 OldDeviceTable 中的记录均来自 DeviceTable。

DeviceTable的Accumulation字段由底层工控机定时采集每天每台设备的生产量按行 (一个设备记录为一行) 进行累加、刷新。上层管理工作站对DeviceTable表的所有字段均有修改权。

用户的功能要求:

(1) 无论是每天底层工控刷新 Accumulation 字段, 还是上层管理工作站修改 Accumulation 或 DesignAge 字段, 设备累计运行百分比 (percent=Accumulation/Design Age) 均能自动计算出来。

(2) 当 percent 超过 100% 时, 向客户端返回一个消息, 提醒用户注意, 进行设备检修, 更换等。

(3) 当对设备进行完检修和更换之后, 由操作员将 CheckTime 字段更新, 此时, 要自动将上次对此设备进行检修的记录 (即更新前的旧记录) 写入 OldDeviceTable 表中, 并统计历史检修次数 (CheckId), 供以后查询参考。

分析以上三点要求, 可以知道这实际上是一个如何维护表 DeviceTable 和 OldDeviceTable 之间的数据完整性和有效性的问题, 我们很自然地想到要在 DeviceTable 表上创建一个触发器来实现这些功能。

在编程之前, 有两点值得注意:

① 由于触发器是行触发的, 所以只针对修改的行进行相应的操作 (即需参照 inserted 表和 deleted 表)。

② 触发器不能被自身的更新操作调用, 所以在 UPDATE 触发器中可以大胆使用 update 语句。

下面, 我们在 DeviceTable 上建立一个 UPDATE 触发器 (updateDeviceTable), 引入触发器之后的表间关系如图 2 所示:

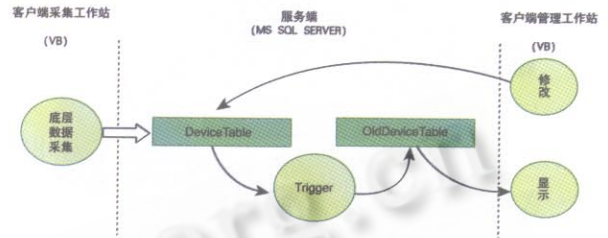


图 2 在 DeviceTable 上创建 update 触发器之后的表间关系

触发器的内容如下:

```
create trigger updateDeviceTable on DeviceTable for
update as
```

```
/*判断设备累计运行时间是否大于其设计寿命,如果是,
则回卷事务, 申请一个错误*/
```

```
if (select accumulation from inserted where
DeviceMark=inserted.DeviceMark)>
```

```
(select destime from inserted where DeviceMark
=inserted. DeviceMark)
```

```
begin
```

```
rollback transaction
```

```
raiserror (50009,16,10)
```

```
end
```

```
else
```

```
/* 计算设备的累计运行百分比*/
```

```
update DeviceTable
```

```
set percent= Accumulation/ DesignAge from
inserted where DeviceMark=inserted.DeviceMark
```

```
/* 判断该设备是否进行了检修, 如果是, 则自动计算
历史检修次数, 并备份检修记录*/
```

```
if update(CheckTime)
```

```
begin
```

```
declare @id int
```

```
if (select count(*) from OldDeviceTable,inserted
where
```

```
OldDeviceTable.DeviceMark=inserted.DeviceMark)=0
```

```
select @id=1
```

```
else
```

```
select @id=max(OldDeviceTable.CheckId)+1 from
OldDeviceTable, inserted
where OldDeviceTable. DeviceMark=inserted.
DeviceMark insert OldDeviceTabl
(CheckId,DeviceMark,DeviceName, OldAccumu,
OldDesign Age,OldPercent,OldCheckTime)
Select@id,DeviceMark,DeviceName,Accumulation,Desig-
nAge,percent,CheckTime From deleted
end
```

建立好以上触发器，在客户端利用 VB 编程实现用户需求就相对容易多了。我们可以利用 DBGrid 对 SQL 库进行 ODBC 绑定，对 DeviceTable 和 OldDeviceTable 的记录内容进行操作。而且，我们可以针对超时运行的设备给出错误提示，提醒用户注意，具体做法是在 VB 客户端程序中设置错误陷阱，程序如下：

```
.....
On Error GoTo errcontrol
.....
```

```
Exit Sub
errcontrol:
MsgBox “操作错误” & CStr(Err.Number) &
“--” & Err.Description
End Sub
```

结 论

从以上这个应用实例，我们可以体会到 SQL SERVER 数据库的触发器技术在数据管理方面的强大功能。在实际网络系统的开发工作中，由于触发器的引入，使开发者的生产率大大提高，如果能再进一步将触发器技术与数据库视图技术、存储过程技术结合起来灵活使用，往往会取得更加令人满意的效果。■

参考文献

- (1) D. solomon, R. Rankins 等著. Microsoft SQL SERVER 7.0 开发指南. 清华大学出版社. 1999 年 4 月
- (2) 闪四清主编. SQL SERVER 7.0 实用教程. 人民邮电出版社. 1999 年 6 月