

# MS SQL Server 2000 中流水号问题的解决方案

## A Solution for Problems in Serial Numbers Generation in MS SQL Server 2000

狄颖跃

DI Ying-yue

(广西柳工机械股份有限公司信息技术部, 广西柳州 545007)  
(Guangxi Liugong Machinery Co., Liuzhou, Guangxi, 545007, China)

**摘要:**以 MS SQL Server 2000 为例分析几种常见流水号生成方法及其不足, 针对这些不足给出 Int、Char 数据类型流水号的解决方案。这些方案具有并发性好, 不会引起阻塞, 计算效率高等特点。

**关键词:**流水号 MS SQL Server 事务 IDENTITY uniqueidentifier

中图分类号: TP131.138SQ 文献标识码: A 文章编号: 1002-7378(2006)04-0272-03

**Abstract:** The generation ways of serial numbers in MS SQL Server 2000 are discussed. A solution for the data type serial numbers of Int, Char are presented in order to solve the shortcomings. The solution reveals well concurrent, high efficiency and no jams.

**Key words:** serial numbers, MS SQL Server, transaction, IDENTITY, uniqueidentifier

现在的信息系统大都离不开数据库的支持, 而所有的数据库开发几乎都会遇到流水号的生成问题。看似简单的一个问题往往关系到一个系统设计的合理性, 甚至决定一个系统设计的成败。本文以 MS SQL Server 2000 为例深入浅出的讨论几种常见流水号生成方式, 并给出最终解决方案。

### 1 常见流水号生成方法及其不足

目前常见的流水号的数据类型主要有两种, 一种是整型(int), 另一种是字符型 Char(Length)<sup>[1]</sup>。生成流水号的方法很多, 常见的有使用 IDENTITY 属性、使用辅助表、使用 Max 函数生成流水号, 以及专门针对 Char 数据类型的生成流水号方法。

#### 1.1 使用 IDENTITY 属性生成流水号

这是 MS SQL Server 2000 数据库本身具有的一种功能。数据库表如下:

```
TradeList(T_ID int identity(1,1), TLTime  
datetime, ……)
```

这种方法的好处是简单易用, identity 列的值由

系统自动维护, 每插入一笔记录系统都会递增的生成一个唯一整型流水号, 不需人为干涉。这种方法的另一个好处是并发性较高, 可以满足有大量数据并发的情况。但是要想取得刚刚插入的流水号, 很多人都会使用 @@IDENTITY 全局变量, 例如:

```
Insert Into TradeList ……
```

```
Select @ID = @@IDENTITY
```

以上代码只能在没有并发情况发生的系统中使用, 如果在多用户特别是大量并发 Insert 的情况下, 无法保证取得的值就是刚刚插入的值, 因为可能在你取值后其他并发事务中的 Insert 语句刚好插入一条记录而改写了 @@IDENTITY 全局变量。

#### 1.2 使用辅助表生成流水号

辅助表结构: Max\_ID(ID int ……)

辅助表中存储当前最新的流水号, 获取流水号的代码如下:

```
Declare @MaxID int
```

```
Set @MaxID = 0
```

```
Begin transaction
```

```
Select @MaxID = ID + 1 from Max_ID
```

```
Update Max_ID set ID = @MaxID
```

```
Commit Transaction
```

这个方法取得的流水号和更新辅助表必须在同

一个 Transaction 中完成,只要程序统一规定通过上述代码获取流水号,就可以保证取得顺序递增和唯一的流水号。这种方法适合于少量用户和并发并不频繁的情况,如果每秒有上千次甚至更多调用时就会出现堵塞。所有的并发进程都会在此等候取得最新的流水号,进而成为整个程序的瓶颈。

### 1.3 使用 Max 函数生成流水号

```
代码如下:
declare @MaxID int
Begin Transcation
    Select @MaxID = Max (T _ ID)    from
TradeList
    Insert into TradeList Values(@MaxID.....)
.....
```

这种方法简单而易用,但是由于 Max 函数的局限性非常容易引发堵塞,尤其在表 TradeList 数据量很大的情况下会引起严重堵塞,并发性非常差。所以这种方法适合单用户、并发量不大并且数据量不是很大的情况。

### 1.4 针对 Char 数据类型的流水号生成方法

一般 Char 类型的流水号都有特殊用途,在 Int 型流水号无法满足需求的情况才使用。例如 0606270001 这个流水号就有特殊意义,它代表 2006 年 6 月 27 号第一笔交易记录。这种类型流水号无法用 Int 型表示,所以一般都会在程序里专门写个函数处理这类情况。比如 00001 这类流水号的处理方式为:

先使用 Max 函数取得最大流水号,然后转换成数字后加 1,再用 string(字符总长度-转换后的数字位长度,"0")函数得到最终字符串。这种方法的缺点正如前面所述容易造成程序堵塞,所以程序的并发性不是很好。

## 2 流水号问题的解决方案

针对上述流水号生成方法的不足,根据两种流水号数据类型的不同分别给出两种解决方案。

### 2.1 Int 数据类型流水号

这种数据类型流水号最简单的方法是采用 IDENTITY 属性生成,其代码如下。

```
表结构:TradeList(T _ ID int identity(1,1),
TLTime datetime,.....)
Declare @MaxID int
Set @MaxID=0
Begin transaction
    Insert into TradeList (TLTime,.....) Values
```

```
(getdate(),.....)
```

```
    Select @MaxID=SCOPE _ IDENTITY()
Commit Transaction
Select @MaxID
```

这种方法和前面介绍的方法唯一区别是采用了 SCOPE \_ IDENTITY() 函数。SCOPE \_ IDENTITY() 和 @@IDENTITY 都是通过读取当前 session 中最后的 IDENTITY 值来得到最后插入的流水号,与 @@IDENTITY 不同的是 SCOPE \_ IDENTITY() 函数的作用范围仅仅在当前代码有效,相当于局部变量,而 @@IDENTITY 则相当于全局变量,为了证明这点可以为表 TradeList 建立一个触发器,并且触发器里面要有 Insert 语句,这时你就会发现 @@IDENTITY 变量会被触发器改变而 SCOPE \_ IDENTITY() 则不会。

在此方法基础上增加 GUID(全局唯一标识符)就可以得到另外一种方法,数据类型 uniqueidentifier 就是全局唯一标识符,这种类型数据是由计算机经过特殊运算产生,在全世界范围内几乎不会有重复值出现。方法的表结构为: TradeList(T \_ ID int identity(1,1), Global \_ ID uniqueidentifier.....), 语句如下:

```
Declare @ MyglobalID uniqueidentifier, MyT
 _ ID int
Set MyglobalID=-1
Select @ MyglobalID = NEWID() /*
NEWID 用于创建 uniqueidentifier 类型的唯一值
*/
Insert Into Tradelist Calues (@MyglobalID
.....)/* 使用全局 GUID 获取刚刚插入的 T _ ID
值 */
Select @ MyT _ ID = T _ ID from TradeList
where Global _ ID=@MyglobalID
Select @ MyT _ ID
```

这种方案的好处显而易见,甚至不用 Begin transaction 等语句就能解决并发问题,而且并发性很好,不会引起阻塞。缺点是需要一个 uniqueidentifier 类型字段的开销,不过比起优点这点开销还是值得的。本例也可使用辅助表的方法实现,此处不另外说明。

### 2.2 Char 数据类型流水号

这种类型的一个比较好的方法是采用辅助表,其流水号形呈多样化,例如:0606270001(年 2 位,月 2 位,日 2 位,四位当天产生的记录)。

辅助表结构如下:Assistant\_MaxID (MaxID

```
char(10),Global_ID uniqueidentifier……)
Declare @MyID char(10),@DateID char(6),
@IndexID int
Begin transaction
/* 获取当前最大流水号 */
Select @DateID = left(MaxID,6),@
IndexID=right(MaxID,4) from Assistant_Table
/* 判断流水号是否为当天的流水号 */
IF (@DateID = convert(char(6),getdate(),
12))
Set @IndexID =@IndexID+1 /* 是当天
流水号,则把当天记录加 1 */
Else
BEGIN /* 不是则日期部分修改为当
前日期,当天记录部分为 1 */
Set @DateID= convert(char(6),getdate(),
12)
Set @IndexID =1
End
/* 获取最终流水号 */
Set @MaxID=@DateID + REPLICATE
('0', 4 - Len (@IndexID)) + CAST (@IndexID as
```

char)

```
Update Assistant_MaxID set MaxID =
@MaxID /* 更新辅助表记录 */
Commit Transaction
```

这种类型流水号一般具有特殊意义,计算起来相对复杂一些。根据流水号定义结构的不同计算方法也相应变动,但整个计算及对辅助表操作过程必须在 Transaction 中来防止其他用户读取到相同数据。这种结构虽然也有效率问题但要比使用 Max 函数直接操作原始表进行计算效率高。

### 3 结束语

以上是作者在多年程序开发当中总结的一些经验。流水号的使用非常广泛,其中 Char 类型的流水号比较复杂。作者建议如无特殊需求一般使用 Int 型流水号,这种类型的流水号计算简单且并发性好,能够满足绝大多数程序需要。

参考文献:

- [1] 徐晨,李文炬. SQL Server 200 培训教程[M]. 北京:清华大学出版社,2002:157-182.

(责任编辑:凌汉恩 邓大玉)

## 广西科学院获 6 项 2006 年度广西科技进步奖

2006 年度广西科技进步奖 9 月份评出结果,广西科学院共有 6 个项目获奖,其中获一等奖 1 项,获二等奖 2 项,获三等奖 3 项。

广西科学院黄日波院长作为第一完成人的“高密度培养高活力  $\alpha$ -乙酰乳酸脱羧酶”项目,获广西科技进步一等奖;广西科学院应用物理研究所于清武所长主持的“中重型汽车无石棉制动器衬片研究”项目,获广西科技进步二等奖;广西计算中心海蓝数据公司卿旭副总经理主持的“大规模图文资料数字化系统”项目,获广西科技进步二等奖;广西植物研究所李锋所长作为第二完成人的“罗汉果规范化种植研究”项目、广西海洋研究所童万平所长主持的“方斑东风螺的人工育苗技术研究”、广西植物研究所研究员李光照主持的“中国蜘蛛抱蛋属植物系统进化研究”项目,均获广西科技进步三等奖。

(罗海鹏)