

基于 SQL Server 2005 查询计划的错误及排除^①

An Error from Querying Plan of SQL Server 2005 and Solutions

蒋冠雄 (绍兴文理学院 计算机系 浙江绍兴 312000)

摘 要 : 本文给出了 SQL Server 2005 查询中发生的一种错误现象 , 并深入分析了错误发生的原因 , 确定了 SQL Server 内部查询计划的执行过程具有潜在问题 , 最后针对该错误的形成给出了三种解决方案。

关键词 : SQL Server 2005 查询错误 查询计划 SQL

1 现象

SQL Server 2005 是微软最新的数据库平台 , 无论从可用性、功能性和规模性的支持上都有了很大的提高 , 许多新的项目中 , SQL Server 2005 已经成为了首选的数据库平台。在最近一个采用 SQL Server 2005 的项目中 , 系统正常运行了三个月 , 原来可以正确执行的一个操作突然出错了。为了说明错误现象 , 现将涉及的数据库表格简化如下 , 表之间的关系则如图 1 所示。

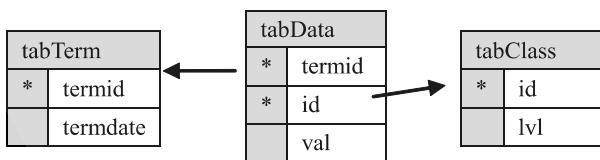


图 1 数据库关系图

类别表 tabClass (类别标识 id int IDENTITY(1, 1) , 类别等级 lvl int)

期数表 tabTerm (期号 termid int , 期日期 termdate datetime)

数据表 tabData (期号 termid int , 类别标识 id int , 采集数据 val float)

在其上 , 定义视图 :

```
CREATE VIEW tabDataView AS
```

```
SELECT tabClass. lvl , tabData. id , tabData. termid ,
tabData. val , tabTerm. termdate FROM tabClass INNER
JOIN tabData ON tabClass. id = tabData. id INNER JOIN
```

```
tabTerm ON tabData. termid = tabTerm. termid
```

引起错误的 SQL 语句为 :

```
SELECT 1/val FROM tabDataView WHERE lvl = 3 AND
termid = 60
```

错误消息是“消息 8134 , 级别 16 , 状态 1 , 第 1 行。遇到以零作除数错误。”但可以确定条件范围内的数据中并没有为 0 的数据。

2 分析

2.1 条件范围内并没有能够引起除零错误的记录

直接对条件范围内的记录添加 val = 0 的条件进行查找 , 结果记录集为空。进一步尝试定位引起除零错误的记录 , 考虑通过添加限定条件来缩小引起错误的记录范围 :

```
SELECT 1/val FROM tabDataView
```

```
WHERE lvl = 3 AND termid = 60 AND id > = 200。
```

在测试中 , 当 id > = 200 时错误仍然存在 , 当 id > = 400 时 , 错误消失。有理由认为引起错误的记录 id 在 200 和 400 之间 , 但执行

```
SELECT 1/val FROM tabDataView
```

```
WHERE lvl = 3 AND termid = 60 AND id > = 200
AND id < = 400
```

却没有引起任何错误。这就证明了条件范围内并没有能够引起除零错误的记录。

2.2 除零错误由条件范围外的记录引起

首先 tabData 表中的确存在 val = 0 的记录。因为

^① 绍兴文理学院校级科研项目(07LG007)

对于 $l < 3$ 的所有记录, val 没有意义, 所其值皆为默认值 0。将这些记录的 val 设置成 1, 错误不再出现。这就证明了的确是由条件范围外的记录引起了该错误。那么, 条件范围外的记录怎么会引起错误呢?

2.3 错误的原因和查询计划有关

通过 SQL Server 2005 提供的执行计划显示功能, 可以得到相应的执行计划。显示执行计划可以采用图形方式(在查询分析器中)、文本方式和 XML 方式。其中文本方式相关的命令为:

```
SET SHOWPLAN_TEXT ON -- 显示执行计划
```

```
Go
```

```
SELECT 1/val FROM tabDataView WHERE l = 3 AND termid = 60
```

得到的结果整理、简化并添加标记后如下所示:

```
l - -[ 6 ]Nested Loops( Inner Join )
  l - -[ 5 ]Index Seek( tabTerm , termid = 60 )
    l - -[ 4 ]Nested Loops( Inner Join , tabData. id )
      l - -[ 3 ]Compute Scalar( 1/tabData. val )
        l | - - [ 2 ]Index Seek( tabData , tabData. termid = 60 )
          l - -[ 1 ]Index Seek( tabClass , tabClass. l = 3 AND tabClass. id = tabData. id )
```

该查询分 6 步完成, 其中第 [2] 步为用 tabData 的主索引选出所有 termid = 60 的记录, 然后执行第 [3] 步, 也就是对第 [2] 步选出的记录进行标量计算 $1/val$ 。显然, 第 [2] 步选出的记录包括了 $l < 3$ 的记录, 也就是包含了 $val = 0$ 的记录, 所以引起了除零错误。

3 解决方法

3.1 根据业务特征, 利用触发器解决

因为 $l < 3$ 的记录的 val 字段并没有意义, 所以其取值可以任意, 不妨设置为 1。如果不修改系统代码, 可以通过设置默认值或触发器实现。考虑到默认值会影响到 $l = 3$ 的记录, 所以推荐使用触发器, 例如:

```
CREATE TRIGGER flxti ON tabData AFTER INSERT , UPDATE AS BEGIN SET NOCOUNT ON ; UPDATE tabData SET val = 1
```

```
FROM inserted A , tabClass B
```

```
WHERE tabData. id = A. id AND A. id = B. id And B.
```

```
l < 3
```

```
END
```

3.2 使用视图提示, 强制连接顺序

如果业务逻辑不允许随意设置 val 的值, 那么就可以考虑使用强制查询计划, 将上述的执行计划中的标量运算 $1/val$ 延迟到 $l = 3$ 的记录被选出之后, 使用视图提示的强制连接顺序的语句如下所示:

```
SELECT 1/val FROM tabDataView
```

```
WHERE l = 3 AND termid = 60
```

```
OPTION ( FORCE ORDER )
```

执行正常, 分析执行计划可以看到, 该语句首先完成所有的连接运算, 最后才进行 $1/val$ 的标量运算, 因此避免了原来的错误。

3.3 通过使用计划强制来指定查询计划

SQL Server 2005 引入了 USE PLAN 查询提示来强制查询优化器为查询使用指定的查询计划。当查询的计划不理想时, 可以使用 USE PLAN 查询提示。该方法需要用到 XML 格式的查询计划, 表示方式比较复杂, 而且受到较多因素的限制和影响, 所以本文不做进一步的展开。

4 小结

文中分析了引起错误的原因并给出了三个有效的解决方案, 但受条件和水平的限制, 最终并没能完全分析出错误的根本原因。

在项目中, 该错误是在第 56 期的数据开始出现的, 原来的数据应该有同样的问题, 但却一直没有表现出来。如果把数据表格复制出来, 也不一定能重现该错误。这可能和统计信息对查询计划的影响有关, 但尚未能证实这一点。

(由于错误的重现无法通过自行创建数据库来实现, 因此需要的读者可以和本文作者联系以获得存在该错误的数据库文件。)

参考文献

- 1 微软公司. 数据库程序设计 SQL Server 2000 数据库程序设计. 北京: 高等教育出版社, 2004.