

# Delphi 中数据过滤方法比较

## The Comparisons of the Methods of Data filtering with Delphi

周华伟 (通信指挥学院 武汉 430010)

董 强 (解放军理工大学指挥自动化学院)

熊焕宇 (通信指挥学院 武汉 430010)

**摘要:**文章介绍了两种数据过滤方法在 Delphi 开发环境中的应用,详细分析了这两种数据过滤方法的性能,并指出了实际开发中需要注意解决的问题。

**关键词:**Delphi 数据库 数据过滤

### 1 引言

Delphi 集成开发环境具有强大得数据库处理能力,可以访问多种数据库系通,包括本地数据库系统和远程数据库系统。在实际应用中,数据库表中往往存放着大量的数据记录,而应用程序中可能只需对其中的一部分进行操作,为了缩减操作的数据,提高系统整体性能,对数据库表中记录的过滤就显得非常必要,特别是对于网络环境下的数据库操作,过滤技术将会大幅度减少数据传递量,显著改善应用程序的性能。

### 2 数据过滤方法

Delphi 中可以通过多种方法来实现对数据得过滤,如使用 SQL 语句过滤、设置 Filter 属性过滤、给字段设置限制值、在 OnFilterRecord 事件中过滤及通过其他一些函数或过程。在本文中,我们只分析两种比较常用的方法。

#### 2.1 使用 SQL 语句实现数据过滤

使用 SQL 语句实现数据的过滤是普遍采用的方法,它是通过设置数据访问组件的 SQL 属性来实现的。如设置 Query 组件的 SQL 属性可采用 Query1. SQL. Add ('SELECT \* FROM demo')。此方法可以实现数据记录的复杂筛选,特别是当记录数量很大或条件很复杂时,应尽量使用 SQL 语言来筛选。因为在 SQL 语句提交给数据库系统时,数据库系统都会通过一系列的优化查询方法来提高查询效率。但是通过 SQL 语句实现的过滤是一次性的,当 SQL 语句返回查询结果集之后就失

去了对结果集的范围限制。如果数据库中的某些记录发生改变,有些数据访问组件不能察觉,必须再次访问数据库才能实现刷新记录,过滤掉不符合条件的记录。

#### 2.2 设置 Filter 属性实现数据过滤

在通过 SQL 语句返回数据集后,还可以再次设置 Filter 属性来实现对数据集的二次过滤。方法是先设置 Filtered 属性为 False,再设置 Filter 属性为过滤条件字符串,最后将 Filtered 属性值设置为 True,代码如下:

```
adoquery1.Filtered := False;  
adoquery1.Filter := Edit2.Text ;//设置过滤条件  
adoquery1.Filtered := True;  
adoquery1.First;
```

Filter 属性设置也支持通配符,例如:[ degree ] = E \*。与 Filter 属性设置有关的还有 FilterOption 属性,它是一个集合属性,有两个成员:foCaseInsensitive(区分大小写),foNoPartialCompare(\* 不作为通配符)。

### 3 模型建立

某单位下属多个仓库,这些仓库中存着放一批设备(每个仓库可以存放全部或部分设备)。该单位需要按设备类型统计所属仓库设备的变化情况(分调入、调出两种情况)。根据需求,我们建立了三张数据库表,分别是设备信息表(SheBeiInfo)、仓库信息表(CangKuInfo)、设备变化情况表(SheBeiBHQK),如表 1 所示。其中设备信息表中的设备编码采用的是层次编码,如 XX 表示电台,XXYY 表示无线电台等,以便于统

计汇总。

表 1 数据库表结构

表名	字段中文名	字段名	字段类型	说明
SheBeiInfo (设备信息表)	设备编码	SBBM	数字	主键
	设备名称	SBMC	字符	
	参考单价	CKDJ	数字	
CangKulInfo (仓库信息表)	仓库编码	CKBM	数字	主键
	仓库名称	CKMC	字符	
	仓库驻地	CKZD	字符	
SheBeiBHQK (设备变化情况表)	仓库编码	CKBM	数字	外键
	设备编码	SBBM	数字	外键
	时间	SJ	日期	
	变化数量	SL	数字	
	变化类别	BHLB	数字	1: 调出 2: 调入

如果该单位需要对每类设备变化情况按目录树的方式进行统计汇总,技术上可以采取两种方法:

一是统计每类设备变化情况时均通过 SQL 语句访问一次数据库,返回需要的数据,记作方法 1。例如我们需要统计仓库 ck 中设备编码为 sb 的设备在时间段 time1 与 time2 之间调出的数量情况,其 SQL 语句设置为 'Select Sum (sl) as sl, bhlb From SheBeiBHQK a, SheBeiInfo b, CangKulInfo c where a. SBBM = b. SBBM and a. CKBM = c. CKBM and TO\_CHAR (a. SJ, ' + ' + YYYY - MM - DD' + ' between ' + QuotedStr (time1) + ' and ' + QuotedStr (time2) + ' and a. ckbm = ' + ck + ' and a. sbbm like ' + QuotedStr (sb + '%' ) + ' and bhlb = 1'。

二是通过 SQL 语句一次把所需数据从数据库读入内存,然后再按设备类型对返回的数据集进行过滤统计,记作方法 2。我们还是统计仓库 ck 中设备编码为 sb 的设备在时间段 time1 与 time2 之间调出的数量情况,第一步先把仓库 ck 中的所有设备在时间段 time1 与 time2 之间调出的数量和设备编码读入道内存中,其 SQL 语句设置为 'Select Sum (sl) as sl, a. sbbm as sbbm From SheBeiBHQK a, SheBeiInfo b, CangKulInfo c where a. SBBM = b. SBBM and a. CKBM = c. CKBM and TO\_CHAR (a. SJ, ' + ' + YYYY - MM - DD' + ' between ' + QuotedStr (time1) + ' and ' + QuotedStr

(time2) + ' and a. ckbm = ' + ck + ' and bhlb = 1 group by a. sbbm'。第二步是通过设置 filter 属性过滤 SQL 语句返回的数据集,把每条记录的 'sl' 相加得到各设备类型的调出数量,代码如下:

```
sl := 0;
adoquery1. Filtered := False;
adoquery1. Filter := 'SBBM like ' + QuotedStr (sb +
%');
adoquery1. Filtered := True;
adoquery1. First;
while not adoquery1. eof do
begin
sl := sl + adoquery1. FieldByName ('sl'). asinteger;
adoquery1. Next;
end;
```

两种方法之间的区别就是方法 1 要不断的读取数据库,需在连接数据库上话费很多时间。而方法 2 虽然只读取一次数据库,但是要对数据集过滤统计,当数据集记录量较大时统计将会非常缓慢。为了缩短系统的响应时间,在系统开发时应把两种方法结合起来,尽量结合两者之间的优点,避免各自的不足。为了提高 filter 的过滤性能,可以通过设置 SQL 语句的属性把返回数据集的记录量控制在一个较佳的性能点,以充分发挥 filter 过滤性能。但是如果设置的条件过多,就退化成了第一种情况,将会在读取上花费太多的时间。

## 4 性能分析

为了便于分析,分别对 SQL 访问网络数据库的时间开销和 Filter 实现过滤主的时间开销作了测试。通过 SQL 访问网络数据库的时间开销主要是 CPU 开销和通信开销,因此测试的数据主要包括决定返回数据量的记录数和字段数,结果如表 2 所示。通过 Filter 实现过滤主要与数据集的记录数的多少有关,因此测试的是一定数据集规模下实现过滤的时间。构建测试环境的服务器采用 Oracle 数据库系统,设备信息表有 1000 条记录、仓库信息表有 100 条记录、设备变化情况表有 100000 条记录,结果如表 3 所示。

假如该单位仓库中共有 1000 种物质,如果采用方法 1 统计,所用时间开销为  $1000 * 0.34 = 340$  秒 (1000

表示共需统计 1000 次, 0.34 为 SQL 返回一条记录一个字段的的时间开销)。如果采用方法 2, 只需读取一次数据库, 然后对数据集过滤 1000 次。假设 SQL 语句返回的数据集有一万条记录, 则完成统计所需的时间为读取数据库的时间加上 1000 次数据过滤的时间, 即  $1.2 + 1000 * 0.01 = 11.2$  秒。表 4 是具体比较结果。

表 2 SQL 时间开销

返回记录数 \ 字段	1	2	3
1	0.34 ~	0.34 ~	0.34 ~
100	0.35 ~	0.35 ~	0.35 ~
500	0.37 ~	0.37 ~	0.37 ~
1000	0.38 ~	0.38 ~	0.38 ~
5000	0.54 ~	0.55 ~	0.55 ~
1 万	1.1 ~	1.2 ~	1.5 ~
5 万	4.9 ~	5.8 ~	6.1 ~
10 万	9.5 ~	11.4 ~	12.4 ~

表 3 Filter 过滤时间开销

返回记录数	时间开销
100	0.00013 ~
500	0.0004 ~
1000	0.0007 ~
5000	0.003 ~
1 万	0.01 ~
10 万	0.06 ~
50 万	0.21 ~
100 万	0.62 ~

从上表中我们可以看出在 SQL 语句返回数据集记录小于 1 万时, 方法 2 显著优于方法 1。当数据集记录大于 1 万条时, 方法 2 的时间开销明显增大。如果要使时间开销在一个能容忍的范围内, 就需要采取措施将返回的记录量控制在一个恰当的范围, 具体做法就是设置限制条件, 将可以用一条 SQL 语句完成的任务分解成多条 SQL 语句执行, 把需要的数据读入内存, 而后分别对其过滤统计。

如上例中需要统计仓库的调入、调出两种情况, 变化类别 (BHLB) 的限制条件可以直接在 SQL 语句中设

置, 也可以在 Filter 过滤条件中设置。如果‘变化类别’

表 4 时间开销对比

返回记录数 \ 方法	方法 1	方法 2
1	340 ~	—
100	340 ~	0.48 ~
500	340 ~	0.77 ~
1000	340 ~	1.08 ~
1 万	340 ~	11.2 ~
10 万	340 ~	71.4 ~
50 万	340 ~	约 210 ~
100 万	340 ~	约 620 ~

在 Filter 过滤条件中设置, 上例方法 2 的 SQL 语句就变为 'Select Sum (sl) as sl, a.sbbm as sbbm From SheBeiBHQK a, SheBeiInfo b, CangKuInfo c where a.SBBM = b.SBBM and a.CKBM = c.CKBM and TO\_CHAR (a.SJ, '+-+YYYY-MM-DD'+) between '+ QuotedStr (time1) + 'and '+ QuotedStr (time2) + 'and a.ckbm = '+ ck + 'group by a.sbbm', filter 语句变为 bhlb = 1 and SBBM like '+ QuotedStr (sb + %')。如果‘变化类别’加入到 SQL 语句中, 那么查询调入情况需执行一条 SQL 语句, 如上例方法 2 所示。查询调出情况需执行另一条 SQL 语句, 就是把上例方法 2 中的变化类别设置成为调出 (bhlb = 2)。这样每条 SQL 语句返回的数据集的记录数就基本上减少了一半, Filter 的过滤性能将会有显著改善。

### 5 结束语

本文主要分析了 Delphi 中数据过滤的方法, 并给出了特定数据规模下的性能测试结果。在实际运用中, 由于事先不知道数据库的数据规模, 所采用的方法可能随着数据规模的增长而严重影响系统性能。这就需要开发人员对未来的数据规模有一个估算, 使所采用的数据过滤方法能在不同数据规模中取得平衡。

### 参考文献

1 李俊生, 数据过滤技术在 Delphi 中的应用 [J], 计算机应用, 2001, 21(7): 91-92.