

基于数据库查询过程优化设计

王爱军*

(第三军医大学成都军医学院 成都 610083)

【摘要】在大型关系数据库管理与开发中,优化设计极大地提高数据库的性能。通过对一大型数据库查询语句执行过程的讨论,提出了对同一表格进行多个选择运算的优化、多个表格的选择运算的优化、多个表格的联结兼选择运算的优化、应用窗口优化和索引优化设计方法,经实际应用验证,其优化过程简便、有效,可达到减少计算量和内存需求,提高了查询响应的速度,极大地改善数据库性能,具有较强的实用性。

关键词 关系数据库;查询优化;窗口优化;索引优化

中图分类号 TP311.132.3 文献标识码 A

Introducing an Optimized Design in Database Query Process

Wang Aijun

(Chengdu School of Third Military Medical University Chengdu 610083)

Abstract In the management and development of large-scaled database, optimized design can improve its functions greatly. According to the discussion on performing query language of a large-scaled database, the authors propose an optimized design—optimizing multi-choice operation on the same form, choice operation on multi-forms, connection & choice operation on multi-forms, the using window and index. In practical application, the process of optimization, which is simple and effective, can decrease the amount of counting and internal storage, can speed up query execution and performance and has a stronger usage.

Key words relational database; query optimization; window optimization; index optimization

数据库设计的重要标准是需具有快速的查询响应速度,特别对客户/服务器形式的数据库开发,查询速度直接影响数据库的推广与应用。为了达到这一目的,除在数据库的物理层设计、关系的规范化等方面进行改进外,一个简便、直接、有效的方法就是对多种形式的SQL查询语句进行优化设计^[1-3],以减少计算量和内存需求,通过对SQL语法结构及运算次序的调整来提高查询的响应速度。本文分析了SQL查询语句对查询响应速度的影响,讨论了联结、选择、逻辑等运算,通过对SQL语句执行过程的分析,给出了其优化形式,这种优化方法不仅可用于关系型数据库的开发、应用与管理,也可用于Internrt数据库访问技术^[4]。

1 SQL查询语句的优化过程

在数据库的管理与应用中,关系是结构的基本单位,SQL语言支持关系运算,不同于其他数据库的标准语言。在许多流行的产品中,SQL语言可以执行关系的投影、选择、联结等运算,通过其语法和语句来实现。由于对同一查询语句有多种不同的语法表达形式,本文则对各种形式进行分析。

1.1 对同一表格进行多个选择运算

关系在数据库中的具体表现即为表格,选择运算是数据库中的常用操作之一,用户通过运算,可以从数据库中得到有选择性的数据。假定在一个有关电话用户的数据库管理系统中,有表格Customer(Name,

2002年12月20日收稿

*男 41岁 讲师 在职硕士生 主要从事计算机在新应用领域中的基础研究

Tel, City, Dues, DueDate, Pay, PayDate), 括号内的各项为表格的属性, 下面的一个SQL查询语句返回所有Chengdu地区的用户中, 如2002年8月10日交费的用户姓名、电话号码、应交款和所交款为:

```
SELECT Name, Tel, Dues, Pay
FROM Customer
WHERE City = 'Chengdu' AND PayDate= '10-AUG-2002'
```

若表格Customer中有用户 10^7 个, 其中Chengdu地区有用户 10^6 个, 上例中交费用户20 000个, 其中Chengdu地区为2 000个。则执行完这一查询语句后, 系统返回一个包含2 000行的数据表格, 在查询语句中有两个选择条件: City = 'Chengdu', PayDate= '10-AUG-2002', 在执行完第一个选择条件后, 得到一个 10^6 行的临时表格, 第二个选择条件是对这个表格再进行选择运算, 从而得到所需结果。如果将选择条件的次序对换, 即将WHERE后的语句改写为WHERE PayDate= '10-AUG-2002' AND City = 'Chengdu', 则先得到20 000行的临时表格, 再得到同样的结果, 可见, 选择条件的次序极大地影响查询语句的计算量, 前者临时表格的行数是后者的50倍, 所以要提高查询响应速度, 可将较严格的选择条件写在前面, 较弱的条件放在后面, 可在执行过程中迅速将不满足条件的行删除。

1.2 多个表格的选择运算

对多个表格进行选择运算时, 表格的次序也影响查询响应速度。如上例中还有一个与Customer类似的表格Customer1, 交费的用户为3 000个, 则SQL语句返回包含所有23 000个交费的用户有关数据如下:

```
SELECT Name, Tel, Dues, Pay
FROM Customer, Customer1
WHERE PayDate= '10-AUG-2002'
```

在具体执行过程中, 先对表格Customer进行选择运算, 得到20 000行的临时表格, Customer1进行选择运算, 将得到的3 000行插入到临时表格中, 需做3 000次插入运算。若将FROM语句后面的表格次序对换, 即为“FROM Customer1, Customer”, 则得到一个3 000行的临时表格后, 需做20 000次的插入运算, 其计算量远大于第一种情况, 故可将返回较多行的表格排在前面, 较少行的表格排在后面, 以便减少插入运算。

1.3 多个表格的联结兼选择运算

联结运算是数据库查询中常用运算之一, 由于联结运算会产生一个很大的临时表格, 特别是数个表格的联结运算, 不优化的SQL查询语句会产生巨大的计算工作量, 且增加对内存的需求, 严重时可使软件在执行中出现异常。在上例中设该数据库管理系统中还有一个表格Address(Tel, Street, ZipCode), 包含了有关用户的地址信息, 若想得到这一天交费的用户地址, 下面两个SQL语句都可以单独完成操作:

```
1) SELECT *
   FROM Customer, Address
   WHERE Customer.Tel = Address.Tel
   AND Customer.PayDate= '10-AUG-2002';
2) SELECT *
   FROM Address
   WHERE Tel =
   ( SELECT Tel
     FROM Customer
     WHERE PayDate= '10-AUG-2002'
   );
```

两个查询语句返回同样的数据结果, 但查询响应的的时间不同, 在1)中先进行两个表格的联结运算, 得到一个 10^7 行的临时表格, 再进行选择运算, 得到所需的结果。而2)中先对表格Customer进行选择运算, 立刻删除了大量多余数据, 得到一个仅有20 000行的临时表格, 再对这个表格与Address进行联结运算而得到最终结果。可见, 1)中的计算量要比2)大得多, 这是因为在1)中联结运算的两个表格都是 10^7 行, 选择运算也是在巨大的表格上进行, 而2)中的联结运算仅对一个20 000行的表格进行。故只要对SQL查询语句作适

当的优化设计, 就可以大大提高数据库的性能。

1.4 窗口优化

数据库管理系统中贮存的数据表格一般都是整个数据的总汇, 表现形式较为单一, 远不能满足不同用户需要有不同的数据表现形式的要求, 用户要想得到所需的数据, 必须对数据库中的总表格进行运算, 则会产生一些问题, 如总表格的规模较大, 用户需要的数据往往仅是其中的一小部分, 而每次运算都耗时。另外, 总数据表格中含有一些保密数据等, 解决的方法是对每个用户设置访问权限, 但在这个层次上的设置会显得复杂而又低效, SQL中的窗口功能提供了解决这一问题的途径。利用窗口能提供不同的数据表格形式, 通过对原始数据的加工处理, 形成新的向某用户开放的窗口, 用户只要对数据进行检索查询, 因而计算量小, 查询响应速度快。同时由于用户只能进入窗口所提供的数据, 无法进入总的原始数据, 数据库管理员可以根据用户的类型提供不同的数据表格, 而保留保密部分, 从而较容易地达到设置权限的目的。如电信公司的维修部门只对用户的地址感兴趣, 不需了解交费的情况, 数据库管理员可以为其产生一个包含用户地址的窗口, 维修部门只对窗口操作, 无需进入巨大的总数据库, 查询响应的速度就会得到相应地提高, SQL为维修部门产生一个名为Maint-Dept窗口为:

```
CREATE VIEW Maint-Dept AS
SELECT Name, Tel, Street, City, ZipCode
FROM Customer, Address
WHERE Customer.Tel = Address.Tel
```

1.5 索引优化

定义数据库的索引可以加快数据库访问的速度。索引的本质是一系列指针, 指向经过索引的表格, 一旦对某一属性定义了索引, 当对该属性进行检索时, DBMS不会像往常一样在表格里逐个检查该属性字段, 而是直接检索该属性的索引, 并获得一个指向所需记录的指针, 最终获得该记录。索引加快了SELECT的速度, 但放慢了INSERT和UPDATE的速度, 对一些实际情况要权衡建立索引的必要, 但执行SELECT的频率一般远大于执行INSERT和UPDATE的频率, 所以索引仍然是数据库设计中常用的手段之一。

除上述可优化过程外, 还有其他一些可优化手段, 如SQL的选择条件包含了两个属性的比较:

```
SELECT *
FROM Customer
WHERE Dues > Pay
```

其中, 大于不等式的右边含有属性Pay, 其逻辑运算计算量大, 要避免使用, 尽量与常数比较。另外逻辑运算中的!=也要避免使用。

2 结束语

在数据库的管理和开发过程中, 优化设计可以提高数据库的性能, 特别对大型数据库, 优化过程更为重要, 不仅可以提高查询响应速度, 还可以减少对内存的需求。通过对SQL语句的过程分析和实例验证, 本文提出的优化过程简单、方便、有效, 从SQL的语法入手, 通过改变SQL语句的被执行次序, 达到减少计算量、提高查询响应速度的目的, 其优化过程可以应用于数据库的具体管理与开发中, 也可用于Internet数据库访问技术中。

参 考 文 献

- [1] Dan S, Tenney R L. Relational database systems[M]. Academic Press, 1995
- [2] 吴炜煜. 工程数据管理系统[M]. 北京: 清华大学出版社, 1996
- [3] 陈 红, 金云程. 关系数据库中并发事务的优化方法[J]. 计算机工程与应用, 1997, 34(9): 56-58
- [4] 吴 跃, 余 水, 傅 彦, 等. Internet数据库访问技术[J]. 电子科技大学学报, 2001, 30(1): 58-61

编 辑 徐培红