CHIQL的多语句查询特征及其优化处理*

孟小峰 王 珊 林耀棠 黄锦辉 刘文瑋

(中国人民大学数据与知识工程研究所 北京 100872) (香港中文大学系统工程与工程管理学系)

摘要 CHIQL(Chinese query language)是一个正在开发中的中文数据库查询语言. 为满足各种用户的需要,CHIQL在设计上具有如下的特点:符合中国用户思维方式的较自然的中文表述方式;过程化与非过程化相结合的多语句查询. 对CHIQL而言,在实现上与传统SQL的最大区别在于它的多语句查询处理技术方面. 本文在描述了CHIQL语言特征之后,重点讨论了多语句查询独特的优化问题,提出了基于语句合并的优化方法,并定性地分析了效率改善情况.

关键词 中文查询语言(CHIQL),SQL,用户界面,数据库系统.

中图法分类号 TP311.13

信息技术的广泛普及将计算机应用带入新的时代. 在我国数据库管理系统的应用更是日益普遍. 数据库技术最初只为专业人员掌握,但现在几乎所有的人都要与数据库打交道. 因此如何提供一个好的数据库用户界面成为大家普遍关注的问题. 用户一般可分为如下3类:普通用户(casual user),对数据库甚至计算机都知知甚少,无计算机方面的专业知识;最终用户(end user),熟悉计算机上的一些工具软件,但无编程能力;数据库用户(database user),对数据库有基本的了解.

好的数据库用户界面应该不仅为数据库用户、最终用户所接收,更重要的是要为普通用户所接收. 具体来说就是指用户无需太多的学习,就可很自然地完成对数据库的查询. 在过去的时间里人们提出了许多各种各样的用户界面,但大多都需要用户具有一定的计算机或数据库方面的知识,对众多的普通用户,这些界面是不合适的.[1]

对我国用户,问题尤其严重.[2]我国数据库技术的研究和应用起步虽不算晚,但一直是拿来的比较多,而适合中国情况、自创的比较少. 就数据库管理系统本身,这样做倒也无妨. 但在用户界面方面,直接拿来显然是不合适的. 中国用户与外国用户由于语言、文化、历史背景等不同,造成在接受事物上的思维模式的不同. 首先一点,让中国用户使用英文说明一个查询显然是不合理的. 现在虽有一些中文查询界面,但是由于他们基本上是直接从英文翻译

* 本文研究得到国家自然科学基金和Chinese University of Hong Kong and the RGC 94/95 Earmarked Research Grant资助. 作者孟小峰,1964年生,博士生,讲师,主要研究领域为数据库系统,自然语言处理,并行数据库系统. 王珊,1944年生,教授,博士导师,主要研究领域为数据库系统与知识库系统,并行数据库系统. 林耀棠,1933年生,教授,博士导师,主要研究领域为数据库系统,中文检索. 黄锦辉,1960年生,博士,副研究员,主要研究领域为并行计算,数据库,中、文检索. 刘文瑋,1960年生,博士,助理教授,主要研究领域为专家系统,数据库.

本文通讯联系人:孟小峰,北京100872,中国人民大学数据与知识工程研究所

本文1996-08-06收到修改稿

过来的,因此不具有太多的意义.因为对外国用户看起来很自然的界面,不一定对中国用户是自然的.例如SQL目前已成为标准的数据库语言,几乎为所有关系数据库产品所支持.但SQL的SELECT-FROM-WHERE语言形式出自英语的表达方式,并不符合汉语的习惯.实际情况也表明中国用户常常搞不清SELECT-FROM-WHERE的关系和含义.要求中国用户去学这样的语言是很困难的.还有SQL的嵌套用法连外国用户也搞不清楚,更不要说中国用户了.

我们的目的是要开发一个中文查询语言(CHIQL),它不是SQL的翻版,但与SQL有同样的表达能力.它对中国用户来说是自然的,易于掌握的.CHIQL基于"Divide and Conquer"策略,采用了非过程化的语言结构.对复杂的查询,用非过程化语言结构表达比SQL的嵌套结构更易为普通用户接受[3],更便于理解.因此CHIQL查询具有多语句查询特征,这与SQL的单语句形式是截然不同的.受SQL的影响,人们在多语句查询处理及如何改进执行效率方面探讨的不多.本文拟就CHIQL的多语句结构,提出多语句查询的优化方法.

本文第1节给出CHIQL的语言特征,包括自然语言特性、过程化与多语句特性.第2节给出CHIQL多语句查询优化方面问题及处理方法.最后给出全文的总结.

1 CHIQL的语言特征

1.1 CHIQL的自然语言特性及简洁性

CHIQL是专为中国用户而设计的查询语言[4~6],而且要易于为普通用户所接受.毫无疑问自然语言是最适合普通用户.但是由于自然语言在语言翻译上的巨大复杂性以及在处理上的巨大时间消耗,使得完全采用自然语言形式的查询语言在事实上是行不通的.我们的方法是采用一种具有约束性的自然语言式的中文查询语言,即它在保证符合中文自然语言形式描述下,限定使用有限的构造形式,从而使CHIQL的处理变得可行而有效.

要使查询描述对中国用户变得自然,就要考虑用中文访问数据的形式,包括中文语序、用词等.比如对投影运算,我们可以有如下的描述:

在〈某表〉中,写出〈某(些)列〉,将答案送入〈某表〉中.

注意这里的中文语序.对其他关系运算可同样给出其描述.图1给出了CHIQL简单的语言定义(完整的形式化定义见文献[6]).这里采用了易为一般用户所理解的语句模板形式.每一个模板为一简单句型,用户只需填入相应信息即可,因此它易懂、易用、易记.基于此模板型语言,很适合开发出良好的GUI.

1.2 CHIQL的过程化与多语句查询特性

人们在做一件复杂事情的时候,很自然地会想到要把它先划分为若干子步骤,然后一步步去完成.这是人们做事的一个基本思维模式.由此我们想到,数据库查询语言同样要遵循这样的模式.对一个复杂的查询,比较自然的处理方法是先将它分成若干便于处理的子查询,然后将这些子查询有机地串联为最终的查询.此即为"分而治之"策略.[4]因此这必然要求整个查询以过程化的形式来描述.CHIQL正是基于此而采用过程化语言形式.一个CHIQL查询可以由一个或多个CHIQL语句组成.每个CHIQL语句是如图1所示的一个语句模板描述.因此CHIQL查询具有多语句查询的特性,它的每个子查询都是非过程化的,但整体上是过程化.

S1	简单查询:在〈某表〉中，写出〈某(些)列〉,将答案送入〈某表〉.
S2	选择性查询:在〈某表〉中,如果〈某(些)条件〉,写出〈某(些)列〉,将答案送入〈某表〉.
S3	分组查询:在〈某表〉中,按〈某(些)指定列〉分组后,写出〈某(些)列〉,将答案送入〈某表〉.
S4	选择性分组查询:在〈某表〉中,按〈某(些)指定列〉分组后,如果〈某(些)分组条件〉,写出〈某(些)列〉,将答案送入〈某表〉.
S5	复合式查询:在〈某表〉中,按〈某(些)指定列〉分组后,如果〈某(些)分组条件〉,并在其中如果〈某(些)条件〉,写出〈某(些)列〉,将答案送入〈某表〉.
S6	连接:在〈某表〉与〈某表〉中,如果〈某(些)连接条件〉,写出〈某(些)列〉,将答案送入〈某表〉.
S7	选择性连接:在〈某表〉与〈某表〉中,如果〈某(些)连接条件〉,并在其中如果〈某(些)条件〉,写出〈某(些)列〉,将答案送入〈某表〉.
S8	相交:在〈某表〉与〈某表〉中,如果某行同时出现,写出〈某(些)列〉,将答案送入〈某表〉.
S9	选择性相交:在〈某表〉与〈某表〉中,如果某行同时出现,并在其中如果〈某(些)条件〉,写出〈某些〉),将答案送入〈某表〉.
S10	合并:在〈某表〉与〈某表〉,如果某行至少出现一次,写出〈某(些)列〉,将答案送入〈某表〉.
S11	选择性合并:在〈某表〉与〈某表〉中,如果某行至少出现一次,并在其中如果〈某(些)条件〉,写出〈某些〉列〉,将答案送入〈某表〉.

图 1 CHIQL 的 11 个语句句型

过去人们有这样的观点,即"非过程化语言一定好于过程化语言". 对此 Welty 等人[3]给出强烈反驳. 非过程化语言的特点在于用户只需给出做什么,而不必关心怎么做. 但对复杂查询,由于受人们思维模式的影响,不可能是完全非过程化的,一味地非过程化只能降低语言的表达能力. 众所周知 SQL 是一个非过程化语言,但恰是这一点限制了它的表达能力. 我们曾用 Lacroix[7]的 66 个查询验证 SQL 与 CHIQL 的表达能力,结果发现 SQL 只能表达 60 个,而 CHIQL 能表达 65 个.[5]对 SQL 而言,多么复杂的查询都只能用复杂的单语句表述. 但对 66 个查询中的如下查询,用单语句是不能完成的. CHIQL 由于具有多语句特性,所以在表达能力上超过了 SQL.

例 1:列出各员工及其与本部门平均薪金的差额. 这里需先求出本部门平均薪金的值,然后与各雇员求差值. 对这样的查询,用 SQL 只能通过如下的方式来完成.

```
create table AVGSAL(deptno,
                    avgsalary )
                    as select distinct deptno,avg(salary)
                      from EMP
                      group by deptno;

  select E. eno,E. ename,E. salary - A. avgsalary
  from EMP E,AVGSAL A
  where E. deptno=A. deptno;
drop table AVGSAL;
```

由于 CHIQL 的多语句特性,它表达这样的多步查询会很自然. 具体 CHIQL 描述如下:

在雇员表中,按部门号分组后,写出部门号、平均薪金,将答案送入平均薪金表;

在雇员表与平均薪金表中,如果雇员表的部门号等于平均薪金表的部门号,写出雇员表的薪金与平均薪金表的平均薪金之差.(注:这里我们用了中文表名和列名.)

在 CHIQL 中,每个语句都可用"将答案……"子句指出查询结果的输出对象,即临时表. 当缺省此子句时,查询结果直接显示给用户. 临时表有动态和静态之分,分别对应视图和快照的概念. 每个语句可以引用其他语句的输出结果. 当语句 Q2 引用语句 Q1 的输出临时

表时(T1)时,我们称Q2依赖于Q1,简记为Q2→Q1(T1).

2 多语句查询优化

2.1 基于语句合并的优化

在查询的构造过程中,用户通常关注的是查询的语法与语义,而很少考虑查询的效率.尤其对普通用户,它更不知怎样去构造出代价低的查询.因此如何为查询选择最好的执行计划,在RDBMS中需要有专门的查询优化器来完成.传统的查询优化多是基于谓词的代价估算,去确定代价低的执行计划.因此查询的结构主要是指查询中的谓词形式,对优化器是至关重要的.

对多语句查询,当两个语句间存在依赖关系时,很多情况下可以通过构造新的谓词的方式将二者合并.合并的结果更易于优化.可合并的语句在合并后比合并前具有更好的执行计划.我们可通过如下的例子来说明这一问题.

例2:列出为工程名为Tape的工程供货的供货商的号码,所供零件的号码及数量.

用CHIQL写出的该查询为:

Q1:在工程表中,如果工程号等于Tape,写出工程号,将答案送入T1表中;

Q2:在订货表与T1表中,如果订货表的工程号等于T1表的工程号,写出供应商号,零件号及供货数量.

直接执行该CHIQL查询,将先执行Q1,再执行Q2.如果工程表及订货表在零件号上都有索引,上述查询将不能利用该存取路径.这将极大地影响其效率.

其实这一问题与用户选用什么样的语句模板有关.针对上述查询,若选择选择性连接句型S7,则可写出下面的查询:

Q3:在订货表与工程表中,如果订货表的工程号等于工程表的工程号,并在其中如果工程表的工程名等于Tape,写出供应商号、零件号及供货数量.

我们说Q3具有更好的查询结构.这里的谓词能使优化器确定更好的连接算法(如索引连接).但我们不能要求用户去做这样的选择.因为用户在构造查询时关心的只是语义和语法.查询的效率需要系统来保障.这是多语句查询存在的新的优化问题.

CHIQL是一个集过程化和非过程化于一体的多语句形式.它在局部上(语句)是非过程化的,而在整体上是过程化的.CHIQL的基础仍然是基于关系模型下的关系代数.因此不难看出,传统的SQL优化方法只适用于CHIQL的局部优化.对多语句查询来说,它不但要求每一个语句在处理上是高效的,同时更要求整个查询在处理上是高效的.因此多语句查询的优化应该比单语句优化更具挑战性.

针对多语句查询的特点,CHIQL的优化器必须在传统优化器的基础上加以扩充.据此我们提出了两级优化的方法,即将优化分为2个部分:

(1)查询级的优化,它负责在有相互依赖关系的语句间,通过分析来自数据和存取路经的语义信息,提出具有优化价值的谓词,用以合并语句的执行.如上面的例子所示.

(2)语句级优化,它是传统意义上的优化,通过谓词的代价估算,确定较优的执行计划.

查询级优化是我们讨论的重点.查询级优化的主要工作是:寻找具有依赖关系的语句,通过下面的合并算法,将一个查询作尽可能多地合并.重复此过程,直至不能合并为止.

查询级优化的中心在于语句合并,这里我们考虑 Q2→Q1(T1)语句合并的情况. 表 1 总结性地给出了两语句合并情况.

表 1　两语句(Q2→Q1(T1))的语句合并的情况

Q1 \ Q2	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
S1	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
S2	S2	S2	S5	S5	S5	S7	S7	S9	S9	S11	S11
S3	S3	S5	X	X	X	X	X	X	X	X	X
S4	S4	S5	X	X	X	X	X	X	X	X	X
S5	S5	S5	X	X	X	X	X	X	X	X	X
S6	S6	S7	X	X	X	X	X	X	X	X	X
S7	S7	S7	X	X	X	X	X	X	X	X	X
S8	S8	S9	X	X	X	X	X	X	X	X	X
S9	S9	S9	X	X	X	X	X	X	X	X	X
S10	S10	S11	X	X	X	X	X	X	X	X	X
S11	S11	S11	X	X	X	X	X	X	X	X	X

在表 1 中,左边一列是 Q1 的语句类型,上边一横行是 Q2 的语句类型. 中间是两语句合并后的语句类型. 其中 X 表示不可合并. 不难发现,表 1 具有对称性. 因此实际能合并的情况共 21 种. 表 1 的具体含义是:

(1)当 Q1 为简单查询时,Q1 可与任何句型的 Q2 合并,合并结果 CQ 的句型与 Q2 相同. 合并方法是用 Q1 中的基表替换 Q2 中的 T1.

(2)当 Q1 为选择性查询,CQ 的句型为 Q2 的同类复杂句型. 某句型的同类复杂句型是指,在 CHIQL 的 11 个句型中,我们可依据语句的相关性,将他们分为 5 个同类,即(S1,S2),(S3,S4,S5),(S6,S7),(S8,S9),(S10,S11). 其中 S2,S5,S7,S9,S10 分别是各同类 的复杂句型. 例如:

(a)Q2 为简单查询(S1),CQ 为选择性查询(S2). 合并方法为将 Q1 的〔写出……〕子句换为 Q2 的相应部分.

(b)Q2 为选择查询(S2),CQ 为选择性查询(S2). 合并方法为用 Q1 中的基表替换 Q2 中的 T1,用与合并 Q1,Q2 的〔如果……〕子句.

对其他情况将以此类推.

2.2　效率改善

通过语句合并,可以达到 2 个目的:一是减少了表的扫描;二是为语句级优化提供使用索引的机会,从而可利用更佳的扫描路径. 语句合并所带来的效率改善的具体情况举例见表 2. 由于语句合并具有对称性,对比如 S1/S2,S2/S1,我们认为是一种情况,故只考虑 S1/S2. 其他组合类同.

3　总　结

CHIQL 是一种新的数据库语言,它是专为中国数据库用户而设计. 考虑到中国用户的特点和语言习惯,CHIQL 在许多方面与 SQL 不一样. 作为过程化的语言,CHIQL 表达能力超过 SQL. 对复杂的查询,CHIQL 的表达更为自然,易懂. 在查询构造上,使用了简洁的语句模板形式.

表2 语句合并所带来的效率改善情况

合并情况	效率改善
S1/S1	减少1次表扫描
S1/S2	减少1次表扫描,若选择属性属性有索引,可改善选择效率
S1/S3	减少1次表扫描,若分组属性有索引,可改善分组效率
S1/S4,S1/S5,S2/S3,S2/S4,S2/S5	减少1次表扫描,若分组属性或选择属性有索引,可改善效率
S1/S6	减少1次表扫描,若连接属性有索引,可改善连接效率
S1/S7,S2/S6,S2/S7	减少1次表扫描,若连接属性或选择属性属性有索引,可改善效率

如何处理多语句查询是一件值得深入研究的工作.本文结合CHIQL的特点,提出了基于语句合并的两级优化策略.它对改善多语句查询的效率是极为有用的.但目前合并算法只是处理了两语句依赖的情况.从图1中不难看出,CHIQL中还存在三语句依赖情况.同时对表1中不能合并的情况如何处理都是有待进一步研究的问题.

参考文献

1 Cooper R. Introduction: the interaction between DBMS and user interface research. In: Proceedings of the First International Workshop on Interfaces to Database Systems, Glasgow: 1992.1~5.

2 Lum V Y. Advanced computerization in China by integrating cultural aspect's into technology. In: Proc. of the International workshop on Science Frontiers and Priority Setting of NSSFC. 1994.122~136.

3 Welty C, Stemple D W. Human factors comparison of a procedural and a non-procedural query language. ACM Transaction on Database Systems, 1981,86(4).

4 Lum V Y, Lee F W, Fong S K. A query interface truly for Chinese users. In: Proceedings of the forth International Conference on Database Systems for Advanced Applications, Singapore. 1995.138~148.

5 Lam C K, Lum V Y, Wong K F. On the issues of expressiveness and portability of chiql. In: Proceedings of the forth International Conference on Database Systems for Advanced Applications, Singapore: 1995.164~171.

6 Lum V Y, Wong K F, Lam G C K. Chiql—an unconventional Chinese database query language. In: Proceedings of International Conference on Computer Processing of Oriental Languages, Taejon, Korea. 1994.69~74.

7 Lacorix R, Pirotte A. Example queries in relational language. M.B.L.E., Brusseles, Technical Note 107, 1976.

8 孟小峰,王珊.嵌套查询的非嵌套化处理研究.计算机学报,1995,18(4).

THE MULTI-STATEMENT FEATURES AND OPTIMIZATION IN CHIQL

MENG Xiaofeng WANG Shan

(*Institute of Data & Knowledge Engineering Renmin University Beijing* 100872)

Lum V Y Wong K F Low B T

(*Department of Systems Engineer & Engineer Management Chinese University of Hong Kong*)

Abstract CHIQL is a new database query language to satisfy the requests of Chinese users and has the features of naturalness and simplicity, procedural and multi-statement query. As a main feature, procedurality enable users to use multi-statements to express a query with 11 templates, especially concerning complex one. In term of its implementation, multi-statement optimization is a special issue in CHIQL. This paper devises a statement merge method to improve the efficiency of CHIQL query and give some qualitative analysis about it.

Key words Chinese query language (CHIQL), SQL, user interface, database systems.

Class Number TP311.13