关系数据库中“DIVISION”查询机理的研究与应用

Research and Application of “DIVISION” Query in Relational Database

顾　平　（广西大学计算机与电子信息学院　530005）

摘要：介绍了关系数据库中“DIVISION”的查询机理，给出了等价的数理逻辑表达式，得出了实用的 SQL 查询方法，最后解决了一个实际的关系数据库查询问题。

关键词：关系数据库　关系代数　数理逻辑　SQL 语言

1　概述

在关系数据库中，要实现对某些复杂问题的查询，比较有效的方法是根据关系代数理论，针对实际查询问题，先写出该查询的关系代数式，然后再将该关系代数式转换成计算机能够处理的 SQL 语言，以实现在具体数据库管理系统（DBMS）上的数据查询。

2　关系数据库中的关系代数

关系数据库产品之所以占有绝对的市场份额，它的最大特点是有以关系代数为基础的坚实的理论支撑及以关系代数为基础的查询优化。在关系数据库的查询中，用到了关系代数的八种基本操作：INTERSECTION（交∩）、UNION（并 ∪）、DIFFERENCE（差 －）、PRODUCT（笛卡尔积 X）、SELECT（选择 σ）、PROJECTION（投影 Π）、JOIN（连接）和“DIVISION”（除 ÷）。用这八种基本的关系操作可以完成一般的查询要求[1]。其中，交（∩）、并（∪）、差（－）、笛卡尔积（X）、选择（σ）、投影（Π）和 JOIN 关系代数运算能较容易地转换成 SQL 语句，只有“DIVISION”（除 ÷）运算，转换成 SQL 语句比较困难。因为 ANSI SQL－92 和 ANSI SQL－99 中均没有提供专门描述“DIVISION”（÷）运算的谓词，所以，必须用其它的 SQL 谓词来模拟“DIVISION”（÷）运算。

3　“DIVISION”的查询机理及实现

我们以一个实际的批发销售系统为例，剖析“DIVISION”的查询机理以及“DIVISION”操作在该批发销售系统中的实现。

在批发销售系统数据库中有四个关系（表）：

客户 CUSTOMERS（客户号 CNO，客户名 CNAME，电话 PHONE…）

表 1　客户登记表

CNO	CNAME	PHONE	…
c001	王平	8258695	…
c002	张开宏	3786956	…
…	…	…	…

代理商 AGENTS（代理商号 ANO，代理商名 ANAME，城市 CITY，佣金 COMMISSION）

表 2　代理商登记表

ANO	ANAME	CITY	COMMISSION
a01	陈志刚	上海	0.05
a02	寇武	广州	0.06
A03	王哲	广州	0.05
…	…	…	…

商品 PRODUCTS（商品号 PNO，商品名 PNAME，城市 CITY，数量 QUANTITY，单价 PRICE）

表 3　商品登记表

PNO	PNAME	CITY	QUANTITY	PRICE
p01	衣柜 1	香港	15	3500
p02	衣柜 2	广州	22	2800
p03	书柜 1	香港	38	2500
…	…	…	…	…

订单 ORDERS(订单号 ONO,时间 DATE,客户号 CNO,代理商号 ANO,商品号 PNO,数量 QTY,金额 YUAN)

表 4　订货表

ONO	DATE	CNO	PNO	ANO	QTY	YUAN
1001	1-5	c001	p03	a02	5	12500
1002	1-7	c001	p01	a02	3	10500
1009	3-9	c002	p03	a01	6	15000
…	…	…	…	…	…	…

在这个批发销售系统中,若要查询这样的客户:他们订了广州代理商(所有在广州的代理商)的货。根据查询要求,我们写出 SQL 查询语句:

SELECT CNAME FROM CUSTOMERS WHERE CNO IN (SELECT CNO FROM ORDERS WHERE ANO = ALL(SELECT ANO FROM AGENTS WHERE CITY = '广州'))

在对系统进行调试的过程中,我们发现这个查询语句在应用的时候有局限性,该语句只是在最内层子查询:SELECT ANO FROM AGENTS WHERE CITY = '广州'的值为单值时,最终的查询结果才正确。当最内层子查询结果为多值时,结果就不正确了。这种查询结果的随机性说明查询方法不正确。因此,我们必须寻找一种完全可靠的查询方法来实现这类查询。在系统设计的过程中我们感到,对于诸如此类复杂的查询,最好先写出查询的关系代数式,然后再将其转换为 SQL 语句。这个查询的关系代数式主要为一"DIVISION"(?)运算:

$$\Pi_{CNAME}((\Pi_{CNO,ANO}(ORDERS) \div \Pi_{ANO}(\sigma_{CITY='广州'}(AGENTS))) JOIN\ CUSTOMERS)$$

由于 ANSI 的 SQL-92 标准及 SQL-99 标准均没有提供专门描述"DIVISION"(÷)运算的谓词,因此,我们借助于数理逻辑,运用一个等价的数理逻辑表达式:

$$\forall_z(\exists_y p(z,y)) \longleftrightarrow \neg\exists_z(\neg\exists_y p(z,y))$$

将带有全称量词$\forall$的逻辑,转换为等价的带有存在量词$\exists$的逻辑,以此来模拟"DIVISION"(÷)运算。

在这个等价的数理逻辑中,z 表示代理商表中那些在广州的代理商,y 表示订单表中的一行,p(z,y)表示订单表中客户通过 z 订货。

对于查询问题:找出通过在广州的全部代理商订货的客户。我们可以根据以上等价数理逻辑表达式右边的式子,先设一个反例:在广州的代理商没有为客户提供货物。然后,再使这个反例不成立。即为我们要查找的问题:找出通过在广州的全部代理商订货的客户。

用 SQL 查询语句描述该查询。首先,用 SQL 中表/示不存在($\neg\exists$)谓词——NOT EXISTS 来描述这个反例,设为 condition 1:

A. CITY = '广州' AND NOT EXISTS(SELECT * FROM ORDERS O WHERE O. CNO = C. CNO AND O. ANO = A. ANO)

这个反例表示:客户 C. CNO 没有通过在广州的代理商 A. ANO 订货。要使这个反例不成立,则有 condition 2:

NOT EXISTS(SELECT * FROM AGENTS A WHERE condition 1)

condition 2 表示:客户 C. CNO 通过了所有在广州的代理商 A. ANO 订货。所以,要查询通过在广州的全部代理商订货的客户,SQL 查询语句为:

SELECT C. CNAME FROM CUSTOMERS C WHERE condition 2;

将 condition 1,condition 2 代入此查询语句,则有:

SELECT C. CNAME FROM CUSTOMERS C WHERE NOT EXISTS(SELECT * FROM AGENTS A WHERE A. CITY = '广州' AND NOT EXISTS(SELECT * FROM ORDERS O WHERE O. CNO = C. CNO AND O. ANO = A. ANO));

这就是我们要找的"通过在广州的全部代理商订货的客户"的 SQL 语句。

为了验证这个从理论上推导出来的查询语句的正确性,我们在实际应用系统中设置了多种可能的情况,采用多种测试参数,对该句进行了测试,其中一种就是针对前面那个有局限性的查询方法,将数据库中 AGENTS 表中广州代理商这个参数设置为单值和多值两种情况。经过反复测试,所采用的全部参数均得出了正确结果。

(下转第 57 页)

(上接第 62 页)

4 结束语

在许多数据库应用系统的开发中,常常会遇到关系代数"DIVISION"运算所表述的查询问题。实现这种查询最有效的方法就是用一个嵌套了两个带有 NOT EXISTS 谓词子查询的 SQL 语句。

参考文献

1 《Fudamentals of Database System》,Ramez Elmasri、Shamkant Navathe, Boston: Addison Wesley,2004.

2 《SQL 宝典》,Alex 著,陈冰等译,电子工业出版社,2003。

3 《Expressive power of SQL》,Leonid Libkin, Theoretical Computer Science 2003, 296:379 –404.

4 《SQL Server 2000 数据库编程 》,张长富,北京希望电子出版社,2001。

5 《Database Principles, Programming, and performance》, Patrick O'Neil、Elizabeth O'Neil, Morgan Kaufmann Publishers,2000.

6 《Aggregate Operators in Constraint Query Language》, Michael Benedikt、Leonid Libkin, Journal of Computer and System Sciences, 2002, 64: 628 – 654.