第31卷 第16期 计 算 机 工 程 2005年8月
Vol.31 № 16 Computer Engineering August 2005
·博士论文· 文章编号：1000—3428(2005)16—04—03 文献标识码：A 中图分类号：TP311.13

多数据库系统中的全局查询转换方法研究

李瑞轩[1]，霍晓丽[1]，文珠穆[1]，卢正鼎[1]，李 兵[2]

(1. 华中科技大学计算机科学与技术学院，武汉 430074；2. 湖北大学数学与计算机科学学院，武汉 430062)

摘 要：使用查询树作为查询转换的内部表示，通过使用多数据库规范树对全局查询树进行规范化处理，并给出了相应的转换规则。最后给出了多数据库查询转换算法，并对查询转换的等价性进行了分析。

关键词：多数据库系统；查询转换；查询树；多数据库规范树

Research on Global Query Transformation in Multidatabase Systems

LI Ruixuan[1], HUO Xiaoli[1], WEN Zhumu[1], LU Zhengding[1], LI Bing[2]

(1. College of Computer Science & Technology, Huazhong University of Science & Technology, Wuhan 430074;
2. College of Mathematics and Computer Science, Hubei University, Wuhan 430062)

【Abstract】 This paper uses query tree as the internal representation for query transformation, employs multidatabase normal tree (MNT) to standardize the global query tree and gives the transformation rules during the normalization. Finally, it presents the algorithms for multidatabase query transformation and discusses the equivalence of the transformation.

【Key words】 Multidatabase systems (MDBS); Query transformation; Query tree; Multidatabase normal tree

查询处理是多数据库系统需要解决的关键问题之一。多数据库查询处理除了要解决传统数据库查询处理需要解决的问题，还需解决多数据库系统中的特殊问题[1]。

多数据库系统的目标是给用户提供一个统一的访问界面，用户只需要使用多数据库查询语言提交对多数据库系统的查询，就可以获得他所想要的数据，而不必关心各个局部数据库系统的差异[2]。因此，对于用户提交的全局查询，要求多数据库查询处理能自动地将全局查询转换成与局部数据库对应的局部查询，生成查询执行计划，交付给有关的局部数据库去执行，通过合并各个局部数据库返回的结果，以组成最终的全局查询结果[3]。可见，查询转换是多数据库系统查询处理中的关键技术之一。

1 查询转换的内部表示

多数据库用户使用全局查询语言来表达全局查询，但要得到查询结果，必须对数据库中的对象进行具体操作。这里使用对象结构化查询语言MOSQL作为多数据库的全局查询语言，使用对象关系代数作为查询的内部表示方法。MOSQL的语法结构与文献[4]给出的查询语言类似，它基本取自于SQL，只不过SQL主要以集合为基础，MOSQL则提供了更为丰富的数据集，如表、包等。

为了将查询表达式转换为对类的操作序列，需要有一种查询的内部表示，即查询表达式的内部结构，称之为查询树。查询树由全局用户发出的查询经过查询优化器的分析程序进行词法和语法分析后得到。

定义1 查询树是一棵树T=(V，E)，其中：

(1) V是节点集，每个非叶节点是类操作符，叶节点是类名，即查询涉及的类。

(2) E是边集，两节点有边(V_1，V_2)，当且仅当V_2是V_1的操作分量。

例1 现有查询Q_1：“查询在北京地区所属部门工作的职员的工号”，使用MOSQL语言表达的查询Q_1如下：

```
SELECT emp_id FROM employee e, department d
WHERE e.dno = d.dno AND d.address = 'Beijing';
```

其相应的代数表达式为

$$E_1=\Pi_{emp\_id}\sigma_{address='Beijing'}(employee\ dno\overset{\infty}{=}dno\ department)$$

其相应的查询树如图1所示。

更一般地，对于查询表达式$E=\Pi_A(C_1\cup(C_2\infty C_3))\cap\sigma_Q(C_1\infty\Pi_B(C_2))$，其查询树如图2所示。

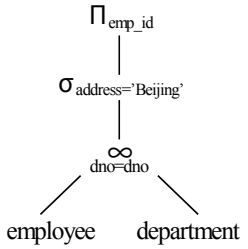


图1 E_1的查询树

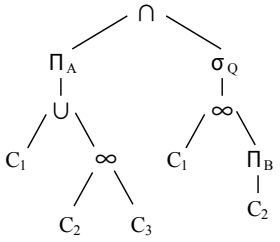


图2 E的查询树

2 查询转换规则

多数据库中的查询表示借鉴了传统数据库的表示，但是在查询转换的处理方法上却有所不同。传统的转换要将查询表达式变成析取条件范式，而由于多数据库系统的特殊性，

基金项目：国家自然科学基金资助项目（60403027）；国家“十五”科技攻关计划基金资助项目（2002BA103A04）；湖北省教育厅科学研究计划基金资助项目（2003A011）

作者简介：李瑞轩(1974—)，男，博士、讲师，主研方向为分布式异构系统集成、分布式系统安全；霍晓丽，副教授、博士生；文珠穆，博士生；卢正鼎，教授、博导；李 兵，博士、副教授

定稿日期：2004-06-21 **E-mail**：rxli@public.wh.hb.cn

需要将查询表达式转换成合取条件范式。下面给出相关定义和规则。

定义 2 形如 P ∞ Q 或 P *=* Q 的表达式，称为简单条件式，简记为(P, Q)。

定义 3 形如 P_1 and P_2 and…and P_n 的表达式，称为合取条件式。其中 n≥1，P_1，P_2，…，P_n 都是简单条件式。

定义 4 形如 P_1 or P_2 or…or P_n 的表达式，称为析取条件式。其中 n≥1，P_1，P_2，…，P_n 都是简单条件式。

定义 5 一个由合取条件式的析取组成的表达式，称为析取条件范式。该表达式具有形式 A_1 or A_2 or…or A_n，(n≥1)，其中 A_1，A_2，…，A_n 都是合取条件式。

定义 6 一个由析取条件式的合取组成的表达式，称为合取条件范式。该表达式具有形式 A_1 and A_2 and…and A_n，(n≥1)，其中 A_1，A_2，…，A_n 都是析取条件式或者简单条件式。

在将全局查询转换为局部查询的过程中，必须制定局部查询结果合并计划，即对查询后处理进行定义。查询后处理可以用表达式来表示，在将查询后处理转换成表达式时应遵循以下规则：

规则 1 如果全局子查询 P 和 Q 的执行结果之间存在连接或外连接运算，则其查询后处理表示成简单条件式 P ∞ Q 或 P *=* Q。

规则 2 如果查询后处理中两个运算得到的中间结果 P1 与 P2 之间存在逻辑与的关系，则其查询后处理表示成表达式 (P1 and P2)。

规则 3 如果查询后处理中两个运算得到的中间结果 P1 与 P2 之间存在逻辑或的关系，则其查询后处理表示成表达式 (P1 or P2)。

规则 4 重复运用规则 1、规则 2 和规则 3，任意一个查询后处理都可以表示成多个简单条件式的合取或析取形式。

由于多数据库系统的分布性和局部数据库的自治性，在向局部数据库下发查询条件时，加强查询条件下发会造成数据的丢失，减弱查询条件下发会造成数据多取，但多取的数据可以在查询后处理中删除。

在析取条件范式 A_1 or A_2 or…or A_n 中，它的每个子树都是由合取条件式组成的，在划分条件的时候，如果去掉条件 A_i or A_j or…or A_k 那么就相当于加强了查询条件，显然会少取一些数据，所以，在多数据库系统中采用析取条件范式是不可行的。

在合取条件范式 A_1 and A_2 and…and A_n 中，它的每个子树都是由析取条件式组成的，在划分条件的时候，如果去掉条件 A_i and A_j and…and A_k，那么就相当于减弱了查询条件，显然会多取一些数据，但是多取的数据可以在查询后处理中删除，所以，在多数据库系统中采用合取条件范式。

定理 1 分配律等值式

P and (Q or R) ⇔ (P and Q) or (P and R) ，其中 P、Q 和 R 是任一条件式。

证明：若 t 为条件式 S 的任一结果，则设定将其关系表示成 t∈S。

设 t∈(P and (Q or R))，则 t∈P 且 t∈(Q or R)，即有 t∈P 且 t∈Q 或 t∈P 且 t∈R，即 t∈(P and Q)或 t∈(P and R) ，所以 t∈((P and Q) or (P and R))，即 P and (Q or R) ⇒ (P and Q) or (P and R)。

又设 t∈((P and Q) or (P and R))，则 t∈(P and Q)或 t∈(P and R)，即有 t∈P 且 t∈Q 或 t∈P 且 t∈R，即 t∈P 且 t∈(Q or R)，所以 t∈(P and(Q or R))，即(P and Q) or (P and R) ⇒ P and (Q or R)。

综上所述，P and (Q or R) ⇔ (P and Q) or (P and R)成立。

规则 5 运用定理 1，多个简单条件式的合取或析取形式都可转换成合取条件范式的形式。

3 多数据库规范树

在多数据库系统内部，查询表达式实际上表示成查询树结构，在查询转换过程中需要对查询树按多数据库查询处理的要求进行规范化，下面给出相关定义。

定义 7 条件树是一棵完全二元树 CT=(V，E)，其中：

(1) V 是节点集，每个分枝结点是合取 and 或析取 or 连接符，树叶结点是简单条件式；

(2) E 是边集，两结点有边{V_1，V_2}，当且仅当 V_2 是 V_1 的连接式。

定义 8 表示析取条件范式 A_1 or A_2 or…or A_n 的条件树称为规范树 (NT)。其中，分枝结点是合取 and 或析取 or 连接符，叶结点是简单条件式。

定义 9 规范树 NT 中表示合取条件式 A_i(1≤i≤n)的子树称为合取条件子树。合取条件子树的分枝结点都是合取 and 连接符。

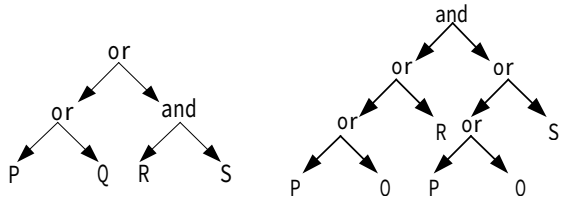
定义 10 多数据库规范树 (MNT) 是表示合取条件范式 A_1 and A_2 and…and A_n 的条件树。其中，分枝结点是合取 and 或析取 or 连接符，叶结点是简单条件式。

定义 11 多数据库规范树 MNT 中表示析取条件式 A_i(1≤i≤n)的子树称为析取条件子树。析取条件子树的分枝结点都是析取 or 连接符。

在多数据库查询树的处理过程中，可使用下述规则构建多数据库规范树。

规则 6 (and 连接符上移规则) and 的左、右子树分别与以其兄弟为根的子树用 or 连接符连接，构成的两棵新树作为 and 的新左、右子树。

全局查询后处理操作包含了全局查询信息和相应全局模式集成信息中涉及的所有场地间运算，可以表示成一个由 AND、OR 和简单条件式构成的条件树，如图 3(a)中示例。按照规则 5，任何查询后处理表达式可以转变成等价的合取条件范式形式，相应地，反复运用规则 6 可将条件树转换成等价的多数据库规范树，如图 3(b)中示例。



(a) (P or Q) or (R and S) (b) (P or Q or R) and (P or Q or S)

图 3 多数据库规范树示例

4 全局查询转换与等价性分析

4.1 求解全局查询的基本数据集合查询表达

为了便于查询转换算法的设计与实现，下面引入基本数据集合的概念。

定义 12 给定查询语句的基本数据集合能用关系代数表达，且能产生该查询语句最终结果的最小关系。

这里，“最小”是指去掉任何一行或一列都不足以得到最

终结果。每个基本数据集合总能由一个 SELECT 查询语句表达（因为能用关系代数表达的总能用 SELECT 语句表达），这个查询语句就称为原查询语句的基本数据集合查询表达。显然，基本数据集合的查询表达和原查询语句在语义上是等价的。

在查询转换的过程中，首先要去掉 SELECT 语句中那些不能由关系代数表达的特征，得到与之对应的基本数据集合的查询表达。在 MOSQL 的 SELECT 语句中，不能由基本关系代数运算表达的特征有：列别名、表达式、聚集函数、分组、排序。算法 1 给出了去除这些特征得到基本数据集合查询表达的算法。

算法 1 去除全局查询语句中的非关系代数特征

输入：全局查询语句

输出：去除非关系代数特征的查询表达

算法：

（1）去掉 select_list 中的聚集函数，如果聚集函数参数中的列名不在当前的 select_list 中，则将其添加到 select_list 中；

（2）去掉出现在 select_list 中的表达式，如果表达式中的列名不在当前的 select_list 中，则将其添加到 select_list 中；

（3）把 select_list 中没有别名的列表达式加上别名；

（4）将出现在 where_list 中，却不在 select_list 中的列加到 select_list 中；

（5）对 group_list，having_list，order_list 重复（4）中对 where_list 的操作，并去掉相应的 GROUP BY、HAVING、ORDER BY 子句。

4.2 将查询语句转换为用 UNION 连接的多个合取查询

合取查询是 WHERE 子句中的条件只包含 AND 和 NOT 两种连接符，而且每个 NOT 作用域内部都没有 AND 的查询。当某个查询不是合取查询时，需要将它转换为多个合取查询来处理。根据德·摩根定理，每个布尔表达式都可以规范为合取式的析取形式，即具有 A_1 OR⋯OR A_n，其中 A_i（$1\le i\le n$）是一个合取式，则任何查询的基本数据集合查询表达都可表示为：

SELECT select_list FROM table_list WHERE A_1 OR A_2 OR A_n (1)

可以将其转换为下列语句：

SELECT select_list FROM table_list WHERE A_1

UNION SELECT select_list FROM table_list WHERE A_2

…

UNION SELECT select_list FROM table_list WHERE A_n (2)

可以证明语句(2)与语句(1)等价。因此，下面主要对合取查询进行处理。

4.3 查询转换及其等价性分析

对于一条全局查询语句 GQ：SELECT select_list FROM from_list WHERE where_list，其中，假设已经经过预处理使得所有在 where_list 中出现的列都出现在 select_list 中。现将 from_list 分为 r 组 from_list_i（i = 0,⋯, r），相应地，把属于 from_list_i 的列分到 select_list_i 组，把只与 select_list_i 中的列相关的 where_list 中的条件都分在 where_list_i 中，将 where_list 中不属于任何 where_list_i 中的条件分到 where_list_0 中，其中，where_list_i（i = 0,⋯, r）共同组成 where_list 的合取范式。注意，如果 GQ 的查询条件不能完全由合取范式构成，则可以将其转换为用 UNION 连接的多个合取查询，然后再分别处理。忽略属性在元组中出现的先后次序，则原 SELECT 语句可改写为：

SELECT select_list_1, select_list_2,⋯, select_list_r

FROM from_list_1, from_list_2,⋯, from_list_r

WHERE where_list_1 and where_list_2 and⋯where_list_r and where_list_0

其中，select_list_i 是全局表 from_list_i 的选择项表达式，where_list_i 是要下发的条件（where_list_i 可能为空），经转换后得到 r+1 条全局子查询语句：

GSQ_1：SELECT select_list_1 FROM from_list_1 WHERE where_list_1;

……

GSQ_i：SELECT select_list_i FROM from_list_i WHERE where_list_i;

……

GSQ_r：SELECT select_list_r FROM from_list_r WHERE where_list_r;

GSQ_{r+1}：SELECT select_list FROM T_1, T_2,⋯, T_r WHERE where_list_0;

其中，GSQ_1, GSQ_2,⋯, GSQ_r 还要根据各自涉及到的局部数据源再行分解，如 GSQ_i 涉及 m 个局部数据源 LDS_1,⋯, LDS_m，则需要将 GSQ_i 分解为针对这 m 个数据源的局部查询 LQ_{i1},⋯,LQ_{im}，再下发给局部数据源去执行；然后对这 m 个局部查询的结果进行合并，得到全局子查询 GSQ_i 的查询结果集 T_i。GSQ_{r+1} 由多数据库全局查询处理器执行，用于对 GSQ_1,⋯, GSQ_r 的查询结果集进行处理，主要是处理不能分派到 GSQ_1,⋯,GSQ_r 的查询条件。很显然，GSQ_{r+1} 最后的执行结果与原全局查询 GQ 所要求的结果是相同的。

5 结论

查询转换是多数据库查询处理中的重要技术，如何有效地将全局查询转换为能够在局部数据库上局部查询，且保证最终的查询结果符合用户最初的查询请求，是查询转换的关键所在。本文通过使用多数据库规范树对全局查询树进行规范化处理，给出了相应的转换规则，给出了多数据库查询转换算法，并对查询转换的等价性进行了分析。本文所提出的查询转换方法已在我们自行研制的多数据库原型系统 Panorama 中进行了实现，实验证明它能够保证查询的正确处理[5]。在未来的工作中，我们将针对多数据库系统的查询优化进行重点研究。

参考文献

1 Chen A, Koh J, Kuo T, et al. Schema Integration and Query Processing for Multiple Object Databases. Integrated Computer-aided Engineering, 1995, 2(1): 21-34

2 Elmagarmid A, Rusinkiewicz M, Sheth A. Management of Heterogeneous and Autonomous Database Systems. San Francisco: Morgan Kufmann Publishers, 1999

3 Roantree M, Murphy J, Hasselbring W. The OASIS Multidatabase Prototype. ACM SIGMOD Record, 1999, 28(1): 97-103

4 Litwin W. O*SQL: A Language for Object Oriented Multidatabase Interoperability. In Proceedings of the IFIP WG2.6 Database Semantics Conference on Interoperable Database Systems, Lorne, Victoria, Australia, 1992: 119-137

5 卢正鼎，李 兵，肖卫军等．基于 CORBA/XML 的多数据库系统研究与实现．计算机研究与发展，2002, 39(4): 443-449