一种实用有效的循环查询处理策略*

王意洁 胡守仁

(国防科技大学计算机系 湖南长沙 410073)

摘 要 通过分析研究面向对象数据库及其循环查询的特点，本文提出了一种实用有效的循环查询处理策略，并对其进行了正确性证明和性能分析。

关键词 面向对象数据库，循环查询，循环成分。

分类号 TP392

A Practical Efficient Cyclic Query Processing Strategy

Wang Yijie Hu Shouren

(Department of Comoputer Science, NUDT, Changsha, Hunan, 410073)

Abstract In this paper, according to the features of object-oriented database and cyclic query, a practical efficient cyclic query processing strategy is proposed, its correctness is proved, and its performance is analysed.

Keywords object-oriented database, cyclic query, cyclic component.

目前，人们对面向对象数据库 (OODB) 中的循环查询的研究重点仍停留在[1],[2]：(1) 如何定义循环查询；(2) 如何用查询语言精确地描述循环查询。这是由于 OODB的复杂性和灵活性，以及 OODB缺乏规范的形式化基础和完善的统一标准引起的，这必然导致人们对循环查询的处理方法的研究不够深入。在深入研究 OODB及其循环查询的本质特点的基础上，我们提出了一种具有一定普遍性和实用性的循环查询处理策略。

1 面向对象数据库及其循环查询

1.1 面向对象数据库

一个面向对象数据库可以视为一个按类划分并且通过各种关系相互关联的对象的集合。在内涵级和外延级两个层次上都可以用图来表示面向对象数据库。在内涵级，数据库可以定义为一个由相互关联的对象类构成的集合，采用模式图 (SchemaGraph) 来表示。图 1是一个机动车数据库的模式图。方形结点代表实体类 (Entity Class)，圆形结点代表域类 (Domain Class)。实体类中的对象代表应用领域中感兴趣的各种实体。每一个实体对象都有系统赋予的唯一的对象标识 (Oid)。域类中的对象是指为定义其它实体对象所需的具体值 (如：整数 8)。类与类之间的关系由模式图中的边表示。类的属性的定义域既可以是域类也可以是实体类，ODMG-93标准将实体类属性定义为关系 (relationship)，而且只支持二元关系 (包括一对一，一对多和多对多三种)，不支持多元关系。同时，关系是双向性的。如果类 A 的属性 A_attr的定义域是类 B，那么类 B中必有属性 B_attr且其定义域为类 A，属性 A_attr和 B_attr是相互对应的。例如，类 Employee的属性 oldschool和类 University的属性 graduate；如果类 A 的属性 A_attr的定义域仍是类 A，那么类 A 中一定有属性 A_attr′，且它的定义域也是类 A，属性 A_attr和 A_attr′ 是相互对应的，例如，类 Employee的属性 subordinate和属性 superior。在外延级，数据库可以视为由属于不同类的相互关联的对象实例构成的网络，采用对象图 (ObjectGraph) 来表示。

* 本文得到九五国防预研项目的资助
王意洁，女，1971年生，博士生

1.2 查询处理

通过分析 OQL查询，可以将其转化为一个查询图。查询图是模式图的进化子图，也就是说，查询图是由模式图的子图依据类与类之间的继承关系进化而成。查询图可以是线型结构、树型结构或网络结构。在查询图中，每个结点表示一个类，结点之间的边上可能标有两种符号——相关符号“+”

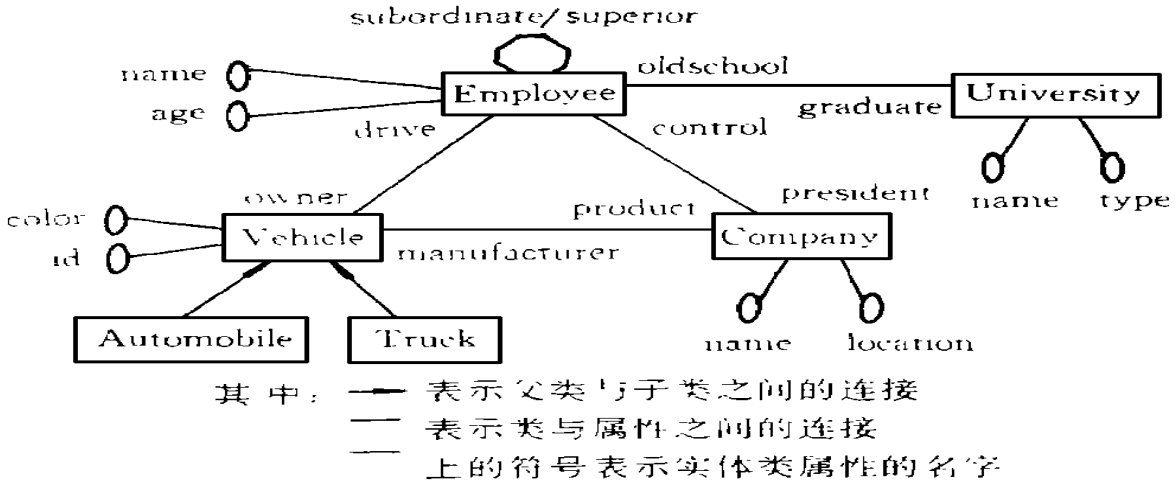

图 1 机动车数据库模式图

```
Query1:
select x.id
from x in Antomobile
where x.color= "blue"
    and x.manufacturer.location= "Detroit"
    and x.manufacturer.president.age< 50
```

图 2 OQL查询

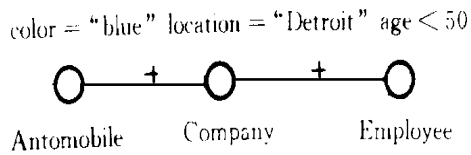

图 3 查询 1的查询图

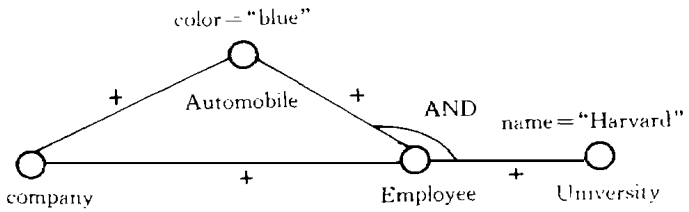

图 4 查询 2的查询图

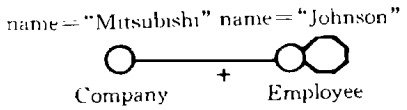

图 5 查询 3的查询图

和不相关符号“-”。如果某边标有“+”，那么查询图表示的查询对通过该边相连的两个类中相互关联的对象实例感兴趣；如果某边标有“-”，那么查询图表示的查询对通过该边相连的两个类中不相互关联的对象实例感兴趣。查询图中包含的分支有两种——“与”分支和“或”分支。“与”分支是指位于分支顶点的类中的对象实例只有与分支上的各类均满足相应边上的符号规定的关联关系，才是查询感兴趣的对象实例。“或”分支是指位于分支顶点的类中的对象实例只要与分支上的各类之一满足相应边上的符号规定的关联关系就是查询感兴趣的对象实例。数据库中被某查询涉及到的部分可以用查询对象图表示，它是对象图的进化子图，也就是说，查询对象图是由对象图的子图依据类与之间的继承关系进化而成。例如，查询 1：“找出所有位于底特律且董事长年龄小于 50岁的公司所制造的蓝色小汽车的车号。”(如图 2所示)。

经过分析，可以将查询 1转化为查询图 (如图 3)。查询处理的过程是首先找出符合全部查询要求

的对象 (即“合格”的对象), 然后对这些对象进行属性提取或其它操作以满足用户的需求。我们把这些“合格”的对象以及它们之间的关系构成的集合称为查询的前期结果。查询的前期结果可以用前期结果图来表示, 它是对象图的进化子图, 是查询对象图的子图。通过前面的分析可知, 我们采用无向连通图对查询进行表示。由于 OODB的复杂性和灵活性, 两个不同的类之间可能存在多个类与属性之间的连接, 那么一个查询中就有可能涉及两个类之间的多个类与属性之间的连接, 而且在查询图中以平行边的形式存在 (即查询图为多重图)。这是一种极其特殊、极其复杂的查询。目前, 我们对查询的研究重点是查询图中只含自圈不含平行边。

1.3 循环查询

所谓循环查询, 是指其查询图中存在环路 (cycle) 的查询[1], [2]。根据前面的论述, 从图论的角度出发[5], [6], 我们研究的查询具有以下特点: 它们的查询图是只含自圈不含平行边的无向连通图。在这个前提下, 查询图中的环路是指查询图中的自圈和顶点个数大于、等于 3的双连通成分。[5], [6] 例如, 查询 2 “找出所有符合下列条件的蓝色小汽车的车号: (1) 由制造它们的公司董事长自己驾驶; (2) 制造它们的公司董事长毕业于哈佛大学”, 在它的查询图 (如图 4) 中, Automobile-Company-Employee-Automobile是一个顶点个数等于 3的双连通成分 (即环路), 因此查询 2是一个循环查询。查询 3: “找出 Mitsubishi公司的所有叫 Johnson的下属职员”, 在它的查询图 (如图 5) 中, Employee形成一个自圈 (即环路), 因此查询 3也是一个循环查询。

2 循环查询的处理策略

目前, 在 OODB的研究领域中, 人们对循环查询的定义和表示问题已经研究得比较成熟, 但关于循环查询处理策略的研究工作才刚刚开始。文献 [1]探讨了循环查询的处理策略, 但是它的研究对象是一类特殊的循环查询, 即满足下述条件的循环查询: (1) 其查询图由一个顶点构成且形成自圈, 或者其查询图是顶点个数大于等于 3且每个顶点的度均为 2的双连通图; (2) 只有一个目标类查询的结果只返回某一个类的有关信息)。从某种意义上说, 上述查询只是循环查询的一种特例, 所以文献 [1] 中提出的循环查询处理方法缺乏普通性和实用性。

概括地说, 研究循环查询的处理策略可以有两种途径: (1) 将循环查询转化为非循环查询 (acyclic query, 又称树型查询), 然后利用非循环查询的有关算法进行查询处理; (2) 直接研究循环查询的处理算法。在 OODB的研究领域中, 人们已经对非循环查询的处理进行了深入的研究, 研究成果包括串行查询处理[7], [8]和并行查询处理[3], [10]两方面。但是, 人们对 OODB中的循环查询还缺乏深刻的认识, 已有的研究工作[1], [2]也不够深入。鉴于此, 我们认为目前选择第一条途径进行循环查询的研究更为合适。

我们依据“分而治之”的原则对循环查询进行处理, 具体的处理策略是: (1) 识别查询中的循环成分; (2) 分别处理各循环成分, 使循环查询转化为非循环查询; (3) 利用非循环查询的有关算法进行查询处理。

2.1 识别循环成分

识别查询中的循环成分, 也就是识别查询图中的环路。具体地说, 我们主要研究以下两种环路: (1) 由一个顶点构成的自圈; (2) 顶点个数大于等于 3且任何顶点均不构成自圈的双连通成分。同一查询图中的任何两个环路之间不存在相同的顶点。

根据查询图中的各种环路的具体特点, 充分利用图论的有关算法, 我们提出了循环成分识别算法, 具体描述如下:

设查询图的顶点集合为 V, 边集合为 E。

STEP-1 依次判断 V 中的顶点是否构成自圈, 这个过程可在 O (‖V‖) 时间内完成, 找出的所有自圈构成集合 S_Set。

STEP-2 利用图论的经典算法——无向连通图的双连通成分识别算法 ([6]) 在 O (‖E‖) 的时间内找出查询图中的所有顶点个数大于等于 3的双连通成分, 它们构成的集合记为 BC_Set。

2.2 处理循环成分

对各循环成分分别进行处理是“分而治之”原则的具体体现。在查询对象图中，与查询图中的某个环路相匹配的对象和对象之间的关联关系也构成一个环路，我们称其为对象环路。对循环成分进行处理的目的就是要找出查询对象图中的所有对象环路。

(1) 自圈

对于查询图中的自圈，我们可以采用文献[1]中提出的处理方法：正向遍历(forward traversal)与嵌套循环检索(nested-loop retrieval)相结合的方法。这种方法可以在多项式时间开销内得到处理结果。

(2) 双连通成分

对于查询图中的双连通成分，文献[1]中也提出了两种处理方法：正向遍历与嵌套循环检索相结合的方法和反向遍历(reverse traversal)与嵌套循环检索相结合的方法。但是，这两种处理方法的适用范围仅限于满足下列条件的双连通成分(设双连通成分的顶点集合为 V_c)：(1)双连通成分中的每个顶点均与 V_c 中的其它两个顶点相连；(2)只有一个顶点的度大于2，或者每个顶点的度均为2且只有一个目标类。

a 双连通成分处理算法

通过对查询图中的各种双连通成分进行分析研究，并结合有关的图论知识，我们提出了具有一定普遍性的双连通成分处理算法，具体描述如下：

设双连通成分的顶点数为 n，边数为 e。

STEP-1 利用图论中求无向连通图生成树的算法[9]在 $O(n+e)$ 的时间内找出双连通成分的以某顶点 V_i ($i=1, \cdots, n$) 为根的生成树。

STEP-2 在双连通成分对应的查询对象图中，根类的每个对象向通过生成树能到达的所有其它类的对象传播自己的循环标记。这个过程可以在 $O(\sum_{i=1}^{n} Num_i)$ 时间内完成，其中 Num_i 表示顶点 V_i ($i=1, \cdots, n$) 所代表的类中的对象数目。最终，每个对象都拥有一个循环标记集合(设对象 O_j 的循环标记集合为 $TSet\_O_j$)。

STEP-3 对双连通成分对应的查询对象图中的所有对象和对象之间的关联关系进行“合格”标记。

STEP-4
```
FOR i = 1 TO n Do /* 依次对每个类进行处理 */
    BEGIN
      FOR j = 1 TO Num_i DO /* 依次对顶点 V_i 所代表的类中的每个对象进行处理 */
      BEGIN
        对顶点 V_i 所代表的类中的“合格”对象 O_j 进行以下处理;
        (1) 对每一个与 O_j “合格”关联的其它类中的“合格”对象 O_k 进行如下判断;
            IF TSet_O_j ∩ TSet_O_k ≠ ∅
            THEN 保留 O_j 与 O_k 之间的关联关系的“合格”标记;
            ELSE 取消 O_j 与 O_k 之间的关联关系的“合格”标记;
        (2) IF O_j 与每个相邻类中的至少一个“合格”对象之间存在“合格”关联关系
            THEN 保留 O_j 的“合格”标记;
            ELSE 取消 O_j 的“合格”标记，并取消与 O_j 有关的所有关联关系的“合格”标记;
      END
    END
```

这个处理过程可以在 $O(\sum_{i=1}^{n} Num_i)$ 时间内完成。

STEP-5 IF STEP-4中取消了某个对象或某个关联关系的“合格”标记

THEN GOTO STEP-4

ELSE BEGIN

(设集合 OC_Set用于存放对象环路)

(1) OC_Set=∅;

(2) 对查询对象图中的“合格”对象和“合格”关联关系进行如下处理;

设双连通成分中的每个类所含“合格”对象的数目为 Q_Num$_i$ (i= 1, …, n).

◦ 从每个类 C$_i$ (i= 1, …, n) 中取一个“合格”对象 O$_i$ (i= 1, …, n), 这 n个“合格”对象构成了一个 n元组, 总共可以形成$\prod_{i=1}^{n} Q_Num_i$不个同的 n元组;

◦ 对每一个 n元组进行如下判断:

IF其中的 n个“合格”对象及它们之间“合格”关联关系与查询图中的环路完全匹配

THEN n个“合格”对象及它们之间的“合格”关联关系构成一个对象环路 (O$_1$, O$_2$, …, O$_n$), OC_set= OC_set+ {(O$_1$, O$_2$, …, O$_n$)};

END /* 这个过程可以在 $O(\prod_{i=1}^{n} Q\_Num_i)$ 的时间内完成 */

b. 正确性证明

定理 1 双连通成分处理算法能找出查询对象图中所有的对象环路.

证明: 设算法得出的对象环路集合为 OC_Set, 查询对象图中所有的对象环路构成集合 AOC_Set.

首先, 证明 OC_Set⊆AOC_Set.

∀OC∈ OC_Set, OC必满足 STEP-5中的条件, 即: OC中的对象和对象之间的关联关系与查询图中的环路完全匹配, 所以 OC是一个对象环路, 即 OC∈ AOC_Set, 因此 OC_Set⊆AOC_Set.

然后, 证明 AOC_Set⊆OC_Set.

∀OC∈ AOC_Set, OC中包含一个根类对象 ro. 设它的循环标记为 t_ro. 由于 OC中的其它对象都与 ro直接或间接关联, 根据算法 STEP-2, ro将 t_ro沿生成树传播给 OC中的其它对象. 算法 STEP-3对 OC中的对象和对象之间的关联关系进行“合格”标记. 算法 STEP-4依次对 OC中的对象及其关联关系进行检测. 设第一个被检测的对象为 o1, 由于 OC中所有对象均具有循环标记 t_ro, 所以 o1的循环标记集合与 OC中的和 o1相关联的任何对象的循环标记集合的交集均不为空, 因此保留 o1与 OC中其它对象之间的关联关系的“合格”标记. 设 o1的相邻类数目为 A_Num, 根据对象环路的性质, OC中必有 A_Num 个对象与 o1相关联, 而且这 A_Num 个对象分别属于 o1的 A_Num 个相邻类, 它们之间的关联关系具有“合格”标记 (根据前面的证明), 所以 o1与每个相邻类中至少一个“合格”对象之间存在“合格”关联关系, 因此保留 o1的“合格”标记. 至此, 对 o1的检测工作结束, OC中任何对象和任何关联关系的“合格”标记都没有受影响. 依此类推, 对 OC中的任何其它对象进行检测也都不会影响 OC中任何对象和任何关联关系的“合格”标记. 进一步说, 无论 STEP-4执行多少次, OC中任何对象和任何关联关系的“合格”标记都不会受影响. 算法 STEP-5对 OC中的“合格”对象和“合格”关联关系进行处理, 显然, OC与查询图中的环路完全匹配, 所以应该将 OC加入 OC_Set, 即: OC∈ OC_Set. 因此, AOC_Set⊆OC_Set.

故 OC_Set= AOC_Set, 即双连通成分处理算法能找出查询对象图中所有的对象环路.

2.3 循环查询的处理

我们将循环查询中的循环成分视为一个特殊的类, 该特殊类的属性由循环成分中各类的属性共同

组成，该特殊类的对象即为与循环成分相匹配的对象环路，利用 2.2节中的处理算法可以得到特殊类的对象，这样，循环查询就转化为非循环查询。然后，利用非循环查询的有关算法（包括优化算法和执行算法）对转化后的查询进行处理以得到最终的查询结果。

2.4 小结

与现有的循环查询处理方法[1],[2]相比，我们提出的循环查询处理策略具有以下特点：（1）它适用于多种循环查询，而且对查询的目标类数目没有限制，具有一定的普遍性；（2）它能在多项式时间内得到查询结果，具有一定的实用性。另外，可以通过优先选择操作的执行，即：首先执行选择操作，然后进行循环查询的转化，最后利用非循环查询的有关算法进行查询处理，进一步提高循环查询处理策略的执行效率。

3 小结

循环查询是数据库领域中的一个重要研究课题，它的研究途径有两种：（1）将循环查询转化为非循环查询，然后利用非循环查询的有关算法进行查询处理；（2）直接研究循环查询的处理算法。根据OODB及其循环查询的研究现状，我们采用第一种途径对 OODB中的循环查询进行了较为充分的研究。针对 OODB中循环查询的具体特点，提出了一种实用有效的循环查询处理策略，它适用于多种循环查询，对查询的目标类数目没有限制，并能在多项式时间内得到查询结果。

参考文献

1 Kim, Kyung-Chang, W. Kim and A. Dale. Cyclic Query Processing in Object-Oriented Databases. Proc of 5th Intl. Conf. on Data Engineening, 1989. 564~571

2 何炎祥，郑振楣，石树刚. 面向对象数据库. 武汉大学出版社，1995

3 王意洁，胡守仁. 面向对象数据库中的并行查询执行，计算机学报，1997，20（Suppl.）：110~115

4 Rick Cattell, et al. The Object Database Standard: ODMG-93. Morgan Kanfmann, 1994

5 肖位枢. 图论及其算法. 北京 航空工业出版社，1993

6 曹新谱. 算法设计与分析，长沙湖南科学技术出版社，1984

7 O. Deux et al. The Story of O2. IEEE Trans know ledge Data Eng., 1990, 2（1）：91~108

8 J. Orenstein, et al. Query Processing in the ObjectStore Database System. in Proc. of the ACM SIGMOD Conf on Management of Data, 1992

9 楼世博，金晓龙，李鸿祥. 图论及其应用. 北京：人民邮电出版社，1982

10 Russell F. Haddleton. An Implementation of a Parallel Object-Oriented Database Systen. Technical Report CS-95-49, Computer Science Department, University of Virginia, December 20, 1995