一种扩充语义的实视图重写查询技术

荀亚玲,张继福,刘爱琴
XUN Ya-ling,ZHANG Ji-fu,LIU Ai-qin

太原科技大学 计算机学院,太原 030024
School of Computer Science and Technology,Taiyuan University of Science and Technology,Taiyuan 030024,China
E-mail:xunyl55@126.com

XUN Ya-ling,ZHANG Ji-fu,LIU Ai-qin.Rewriting query technology using materialized view based on extended semantics.Computer Engineering and Applications,2008,44(12):157-160.

Abstract: Aggregation queries with group-by is now a fundamental issue of data warehouse in ROLAP.Materialized views can significantly improve the performance of query.The paper makes an extension to general rewriting query using materialized views considering the hierarchical relation of attributes of dimension tables and discusses the restrict condition of rewriting query using single materialized view,then presents a method of rewriting query.In the final a local optimal search algorithm is described to find an approximately optimal rewriting query when we consider multi-view to rewrite query.The experimental result shows the algorithm proposed is more efficient.

Key words: ROLAP;materialized view;rewriting query;query optimization

摘 要: 分组聚集查询已成为数据仓库领域研究的核心问题之一,实视图是提高分组聚集查询性能的有效手段。利用维属性间的层次关系,对一般意义上的实视图重写查询进行了扩展,讨论了单一视图重写查询的限制条件,并给出重写方法,在此基础上,提出了一种利用多个实视图重写查询的优化选择算法,并通过实验表明,该算法进一步提高了分组聚集查询效率。

关键词: ROLAP;实视图;重写查询;查询优化
文章编号: 1002-8331(2008)12-0157-04 **文献标识码:** A **中图分类号:** TP311

1 前言

实视图是提高查询效率的有效手段,特别是对于数据仓库而言,由于数据量庞大,对效率查询的要求高。在响应查询时,如果能直接利用实视图,就可以避免相应的重新计算,提高查询处理的性能,因而在数据仓库中,实视图是必不可少的。目前,国内外已有不少相关方面的研究,包括利用实视图优化查询[1-4]、实视图的选择与分割[6]、实视图更新与维护[7,8]等。

在利用实视图优化查询方面,为了减少可能响应查询的实视图的搜索空间,Goldstein 在文献[1]中,提出了过滤树的概念,但其构造条件较为宽松, 搜索得到的候选视图集合仍然较大,且在每次启动时都需要重新构造过滤树,这明显增加了系统的启动时间,基于此,文献[3]提出了层次索引和视图合并两种方法,使得搜索得到的候选视图集合更小。有了能够响应查询的实视图,就要利用这些实视图替代原始查询,文献[3]只考虑单视图替代问题,它考虑将原始查询进行分解,然后利用实视图替代查询的不同部分,从而完成整个查询替代过程,而文献[4]则提出基于关系代数树的多视图替代方法,但其却未考虑聚集操作。最后一个需要解决的问题是:给定一个查询和一组实化视图, 用较少的时间尽可能找出一个执行代价最小的查询重写,文献[9]给出一个线性代价模型,在这个模型中,回答查询的代价等于出现在查询中的视图的行数,文献[3]中,提出一种找出较优重写查询的启发式算法,该算法的时间复杂度是多项式级的,显著改善了查询的性能。但这些概念和方法都未考虑维层次结构的特点。

在提高 ROLAP 性能方面,还有采用新的索引技术、改进和 ROLAP 密切相关的连接和聚集操作算法等,且近几年,在对这些技术的研究中, 均深入考虑了维属性间层次关系的语义特性,而在利用实视图重写查询的相关研究中,对这一语义特性的考虑却很少。因此,本文在充分考虑维属性间层次关系语义特性的基础上,对一般意义上的实视图重写查询进行了扩充。

另外,对于任何一个给定查询,在大多数情况下,只利用一个实视图是无法完全计算的,因此,很多研究[3,5]都考虑将给定查询,在一定条件下进行分解,然后利用不同的实视图计算这些被分解的子查询,从而完成对整个查询的计算。本文,基于此思想,在考虑维属性间层次关系这一语义特性的基础上,首先给出了查询分解的条件,然后,在此基础上讨论了单一视图重写查询的限制条件并给出重写方法,最后提出了一种利用多个实视图重写的查询优化选择算法。

2 基本概念及定义

针对实视图和查询,本文仅考虑在单语句查询上,且为包语义的聚集查询。其基本形式如下:

基金项目: 山西省自然科学基金(the Natural Science Foundation of Shanxi Province of China under Grant No.2006011041)。
作者简介: 荀亚玲,女,硕士研究生,研究方向:数据仓库与数据挖掘;张继福,男,博士,教授,研究方向:数据挖掘与人工智能。
收稿日期: 2007-08-09 **修回日期:** 2007-11-28

SELECT S,$AggS$ FROM $Tset$ /* S 是选择非聚集属性,$AggS$ 是聚集属性,$Tset$ 为查询表集 */

WHERE Jc /* J_c 为连接条件 */

GROUP BY GA_h,GA_f /* GA_h 为各个维的层次分组属性集,GA_f 为各个维的非层次分组属性集 */

HAVING HP /* HP 为分组和聚集条件表达式 */

对具有层次语义的维表 D_i,其维属性间的层次关系表示为 $L_1^i \rightarrow L_2^i \rightarrow \cdots \rightarrow L_n^i$,直接用 L_j^i 来表示相应的属性,维表的主关键字为 ID,可视为 L_0^i。

当一个查询无法只利用一个实视图完全计算时,就需要考虑将该查询进行分解,因此,同样需要利用维属性间的层次关系,将有关查询分解[3,5]的概念进行扩展。

定义 1 查询可分。给定一个查询 Q,若 Q 满足以下条件,那么 Q 可分解为查询 Q_1 与查询 Q_2:

(1)查询 Q 的表集可分解为 T_1 和 T_2,表集 T_1 和 T_2 满足 $Tset(Q)=T_1 \cup T_2$,$T_1 \cap T_2=\varnothing$,Q_1 与 Q_2 是 Q 关于 T_1 和 T_2 分解出来的子查询;

(2)$S(Q_1) \supseteq \{a | a \in attr(T_1) \cap \{S_{T_1}(Q) \cup GA_h(Q) \cup GA_f(Q) \cup Con_{T_1}(Q_2)\}\}$,其中 $Con_{T_1}(Q_2)$ 表示 Q_2 中与查询 Q_1 作连接的属性,$attr(T)$ 表示表 T 中的所有属性集合;

(3)$GA_f(Q_1) \supseteq GA_f(Q)$,$GA_h(Q_1) \supseteq GA_h(Q)$ 且 $GA_h(Q)=GA_h(Q')$,$GA_h(Q)=GA_h(Q')$;

(4)$Agg(Q_1) \supseteq \{Agg(Q) | Agg(Q) \in attr(T_1)\}$;

(5)$Jc(Q)$ 可分解为 $Jc_{T_1}(Q_1)$ 和 $Jc_{T_2}(Q_2)$。

其中:Q' 是 Q_1 和 Q_2 再次作连接和聚集得到的查询。

为了在查询之间进行比较提供基本语义,下面给出问题的几个定义。

定义 2 查询展开。查询 Q 在视图 V 上的展开,是通过将 Q 中的视图及相关属性用相应的基表及基表属性替代得到的,表示为 Q^{ext}。

定义 3 等价重写。给定一个查询 Q 和一组视图 V,一个查询 P 若只使用了 V 中的视图,并且 P^{ext} 与 Q 等价,则 P 是查询 Q 使用 V 的等价重写。

有时候,尽管有些视图包含了不属于查询的表,但它们仍有可能用于回答查询,这样的表称之为多余表。下面给出多余表的定义。

定义 4 多余表集。若视图表集 $Tset$ 满足条件:$Tset(V)=T \cup Tm$,$T \cap Tm=\varnothing$。且 T 和 Tm 满足以下条件:$S(Q) \subseteq attr(T) \cap attr(Tm)$,$GA_h(Q) \subseteq attr(T) \cap attr(Tm)$,$GA_f(Q) \subseteq attr(T) \cap attr(Tm)$,$AggS(Q) \subseteq attr(T)$,且 T 中每一元组只和 Tm 中的一个元组作连接,则称 Tm 为 V 相对于 Q 的多余表集。

定理 1 有一视图 V 和一查询 Q,如果 M 为视图 V 相对于查询 Q 的多余表集,即 $Tset(V)=R \cup M$(其中,M 为多余表集),那么,多余表集不会影响用于改写查询的实视图结果。

证明 首先证明关系 R 与多余表集 M 之间的连接不影响关系 R 的结果行。

由多余表集的定义可知:T 中每一元组只和 Tm 中的一个元组作连接,因此可以将多余表 M_i 视为以 M_i 中的列对关系 R_i 进行扩展,即关系 R_i 和 M_i 作连接不会使结果行发生变化。这样就保证了查询所需的那些元组不会遭到破坏。

再者,多余表集 M 定义中的前 4 个条件说明,M 在查询 Q 是投影无用的、分组无用的和聚集无用的,因此在实视图中,可以删除多余表集的投影、分组和聚集属性。

由此可见,多余表集不会影响用于改写查询的实视图结果。(证毕)

3 利用实视图重写查询的充分条件

实视图在什么情形下可用,是考察视图重写查询中必须要面对的问题。当给定查询满足定义 1 时,被分解为若干个可由单一视图计算的子查询,因此,这里只考虑利用单一视图来重写查询。

一般的,当一个实视图满足以下条件时,可以认为该视图对查询可用:(1) 视图必须应用等价或逻辑上更弱的谓词;(2)视图在聚集属性上的聚集程度要更粗略一些;(3)查询输出所需要的属性、进一步聚集所需要的分组属性以及需要和其它视图做连接的连接属性都应在视图的输出属性中。总之,实视图中应该包含足够的信息用来回答查询。因此参照文献[1,10],在考虑维属性间层次关系这一语义特性的基础上,给出单视图重写查询的充分条件如下:

(1)查询表集 $Tset$ 应满足:$Tset(Q) \subseteq Tset(V)$ 且 $[Tset(V)-Tm] \subseteq Tset(Q)$,其中 Tm 为 V 相对于 Q 的多余表集;

(2)对于层次分组属性集 GA_h 应满足:设 $GA_h(Q)$ 中的分组属性记为 L_j^i,则应存在 $L_k^i \in GA_h(V)$,其中 $k \leqslant j$;若 $k<j$,则还应有 $GA_h(Q) \subseteq S(V)$;

(3)对于非层次分组属性集 GA_f 应满足:$GA_f(Q) \subseteq GA_f(V)$,若 $GA_f(Q) \subset GA_f(V)$ 成立,则还应有 $GA_f(Q) \subseteq S(V)$;

(4)非聚集输出属性集 S 应满足:

①若 $S(Q)$ 中的非聚集输出属性为层次属性 L_h^i,且条件(2)中 $k<j$ 成立,则应有 $L_s^i \in S(V)$,其中 $s \leqslant h$;

②若 $S(Q)$ 中的非聚集输出属性为层次属性 L_h^i,且条件(2)中 $k=j$ 成立,则应有 $S(Q) \subseteq S(V)$;

③若 $S(Q)$ 中的非聚集输出属性为非层次属性,则应有 $S(Q) \subseteq S(V)$;

(5)聚集输出属性 $AggS$ 集应满足:$AggS(Q) \subseteq AggS(V)$;

(6)连接表达式 Jc 和 Having 条件中的 HP 应满足:存在 Jc' 和 HP' 满足:

①$Jc(Q)=Jc(V) \& Jc'$;$HP(Q)=HP(V) \& HP'$;

②Jc' 和 HP' 中的属性存在于 $S(V)$ 和 $Agg(V)$ 中。

在检测连接条件给视图的 $Jc(V)$ 作补偿谓词的时候,同样需要综合利用等价类匹配测试、列范围匹配测试和其他谓词匹配测试[1]。

4 重写查询方法

当实视图 V 与查询 Q 满足上述单视图重写的查询条件时,参照文献[3,10],给出以下实视图重写查询方法。

步骤 1 用实视图 V 替换查询 Q 中 $Tset$;

步骤 2 对于满足条件(6)的 Jc' 和 HP',分别去替换 Q 中的 Jc 和 HP;

步骤 3 对于 $AggS(Q)$ 和 HP 中的聚集属性 $Agg(A)$,有以下几种情况(在此,以 N 表示某个聚集函数的别名):

(1)Agg 是 Min/Max,则以 Min(N)/Max(N)替换 Q 中的 Min(A)/Max(A);

(2)若 Agg 是 Sum,则以 $Sum(N)$ 替换 Q 中的 $Sum(A)$;

(3)若 Agg 是 $Count$,则以 $Sum(N)$替换 Q 中的 $Count(A)$;

因为 $AVG(A)$能转化成 $Sum(A)/Count(*)$,所以在此不考虑该函数;

步骤 4　对于层次分组属性集 $GA_h(Q)$中的任一属性 A,若 $A \in GA_h(V)$,则在 $GA_h(Q)$中去除属性 A;

步骤 5　对于非层次分组属性集 $GA_f(Q)$中的任一属性 A,若 $A \in GA_f(V)$,则在 $GA_f(Q)$中去除属性 A。

利用以上方法重写查询,得到等价查询 Q'。

5　优化选择算法

优化选择算法要解决的问题是:给定一个查询和一组实视图,要用较少的时间尽可能找出一个执行代价最小的重写查询。

利用实视图重写查询的效益可以用线性代价模型[9]来衡量,在这个模型中,回答查询的代价,等于出现在查询中的视图的行数,该模型没有考虑作连接的代价,而本文所考虑的实视图重写查询,一个查询能利用多个视图来重写,从而需要考虑作连接的代价。

寻找最优查询重写的一个直接方法是采用穷尽搜索算法,它基于视图集合穷举生成所有可能的查询重写,然后从中找出代价最省的查询。由于穷尽搜索算法开销很大,不适合大量复杂的查询,于是本文提出一个局部最优算法。由于表与表之间的连接代价与它们的广义笛卡尔积的大小直接相关,可以考虑以笛卡尔积为代价的代价模型。

局部最优算法 Locoptimal 算法基本思想:

(1)当给定查询 Q 被分解为可以用单视图替代的子查询后,在这些子查询执行查询重写算法时,无需考虑视图之间作连接的代价,因此,考虑利用线性代价模型[9]来衡量利用实视图重写查询的效益。

(2)当给定查询 Q 被分解出来的子查询均完成查询重写后,查询 Q 结果的获取,则需要考虑这些被重写查询后的子查询作连接的代价,此时,考虑以这些子查询的笛卡尔积为代价模型。

局部最优算法 Locoptimal 算法描述如下:

输入:查询 Q 和一组实视图$\{V\}$。

输出:最优查询重写 Q'。

1　$\{V\}=sort(\{V\})$,$open=decompose(Q)$(即 $open=\{Q_1,Q_2,\cdots,Q_n\}$,其中 Q_i 为 Q 分解出来的子查询)。

2　while($open \neq \emptyset$)

①　$open=open-Q_i$。

②　for each V in $\{V\}$\{

If Q_i and V satisfy the sufficient conditions of rewriting query using materialized view

Then $Q_i'=rewrite(Q_i,V)$\}

3　$Q'=join(Q_1',Q_2',\cdots,Q_n')$

4　return Q'

如果用 m 表示$\{V\}$中实视图的个数,t 表示查询 Q 涉及的关系数,n 表示查询 Q 被分解出来的子查询的个数。那么,该算法开销可分为以下几部分来计算:

(1)对分解出来的子查询执行查询重写算法(利用线性代价模型):

①算法实现时,首先将所有的实视图$\{V\}$按 V 的大小(V 的记录数)从小到大进行排序,这部分的开销为:

获取每个实视图的大小:

$$Cost1=\sum_{i=1}^{m} Num\_V_i \quad (Num\_V_i \text{ 表示视图 } V_i \text{ 的记录数})$$

对实视图进行排序:

$$Cost2=\sum_{i=1}^{m-1}(m-i)=\frac{m(m-1)}{2}\approx\frac{m^2}{2} \quad (\text{该算法采用冒泡排序算法})$$

②对分解出来的 n 个子查询,顺序扫描排序后的实视图集合$\{V\}$,判断是否该实视图能否用来重写查询,这部分的开销为:

$$Cost3=m*n$$

(2)当给定查询 Q 被分解出来的子查询均完成查询重写后,将改写后的子查询 $Q_1',Q_2',\cdots,Q_n'$进行连接得到最后的查询结果(利用子查询的笛卡尔积作为代价模型),这部分的开销为:

$$Cost4=\prod_{i=1}^{n} Num\_Q_i' \quad (Num\_Q_i' \text{表示查询 } Q_i' \text{包含的记录数})$$

所以,算法的总开销为以上四部分开销的总和,其时间复杂度为多项式级的。而穷尽搜索算法的时间复杂度为 $O(mt2^t)$,其时间复杂度为查询涉及关系数的指数。相比之下,局部最优算法 Locoptimal 算法更具有优势。

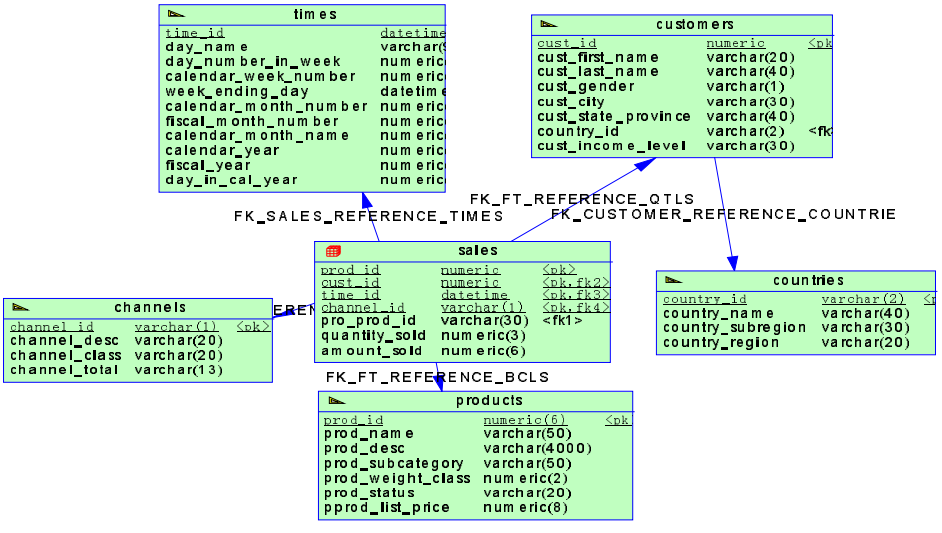

图 1　实验中数据仓库的表结构

6 实验分析

在 PentiumⅥ-3.0G CPU,512 M 内存,Windows XP 操作系统,DBMS 为 ORACLE9i,用 VB 6.0 实现了 Locoptimal 算法,实验中用到的数据是标准数据库,共有 5 个维表和 1 个事实表,其结构如图 1 所示(其中,事实表的记录为 1016271)。

在实验中,建立了 23 个实视图,实视图是从图 1 中的维表、事实表建立的,每个实视图包含了 1 至 5 个关系,分组属性从实视图涉及关系的属性中,随机选取 15%到 30%,并随机生成相应的查询测试集,分组属性的选取比例为 5%到 20%,使实视图被选中的概率达到 60%左右。

在图 2 中,比较了五种关系下局部最优算法 Locoptimal、穷尽搜索算法 Exh 找出的最优查询重写和原始查询的时间效率。可以看出,视图替代查询中的关系数越多,性能提高得越明显。这是因为实视图已经对原始数据进行了投影、连接、分组等预处理,避免了查询时相应的重新计算。所以利用实视图响应查询会更高效一些。而且,Locoptimal 算法是在对实视图排序的基础上,顺序查找,第一个满足条件的视图将会被利用,从而避免了穷尽搜索算法的完全扫描操作,因此,Loc 算法更优,且实视图数量越大,Loc 算法比 Exh 算法的优势会更明显。

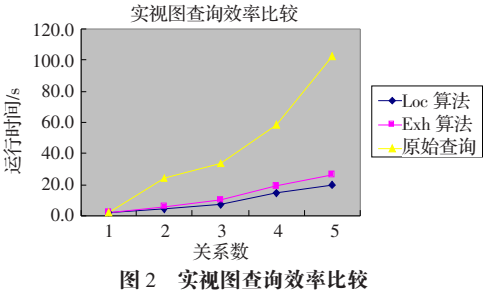

图 2 实视图查询效率比较

7 总结

利用实视图,虽然增加了数据仓库中数据的冗余度,也相应增加了维护实视图的系统开销,但利用实视图优化查询仍不失为一种有效的查询优化方法。本文利用维属性间的层次关系,对一般意义上的实视图重写查询进行了扩展,讨论了单一视图重写查询的限制条件,并给出重写方法,在此基础上,提出了一种利用多个实视图重写查询的优化选择算法。

参考文献:

[1] Goldstein J,Larson P.Optimizing queries using materialized views:a practical,scalable solution[C]//Proc of SIGMOD,2001:331-342.

[2] Afrati F,Chirkov R.Selecting and using views to compute aggregate queries[C]//Proceedings of the Tenth International Conference on Database Theory(ICDT-2005),Edinburgh,Scotland,January 2005:117-129.

[3] 陈长清,程恳.一种利用实化视图快速响应查询的技术[J].计算机工程与科学,2005,27(6):57-61.

[4] 周丽娟,柳池,刘大昕.在数据仓库中使用实视图优化查询[J].计算机工程与应用,2004,40(16):181-183.

[5] Chaudhuri S,Krishnamurthy S,Potamianos S,et al.Optimizing queries with materialized views[C]//Philip S Y,Arbee L P C.Procesdings of the 11th International Conference on Data Engineering,Taipei,1995.Los Alamitos:IEEE Computer Society,1995:190-200.

[6] Valluri S R,Vadapalli S,Karlapalem K.View relevance driven materialized view selection in data warehousing environment[C]//The 13th Australasian Database Conference ADC2002,Meboume,Australia,Conferences in Research and Practice in Information Technology,2002.

[7] Lee A,Nica A.The EVE approach:view synchronization in dynamic distributed environments.IEEE TKDE,2002.

[8] Chen S,Chen J,Zhang X.Detection and correction of conflicting source updates for view maintenance[C]//Proceedings of IEEE ICDE,2004.

[9] Harinarayan V,Rajaraman A,Ullman D J.Implementing data cubes efficiently[C]//Inderpal S M.Proceeding of ACM SIGMOD International Conference on Management of Data,Montreal,1996:205-216.

[10] Srivastava D,Dar S,Jagadish H V,et al.Answering queries with aggregation using views[C]//Proceedings of the 22nd VLDB Conference India,1996.

(上接 146 页)

表 1 各算法的分类正确率及时间

数据集		传统 SVM	LI 等人 SVM	本文 SVM
Iris	r/%	93.33	95.37	97.34
	t/s	1.47	1.32	1.18
Glass	r/%	73.35	74.26	78.23
	t/s	3.2	2.8	2.2
Vowel	r/%	56.69	76.17	86.45
	t/s	6.3	5.7	5.2
Mushroom	r/%	52.63	71.26	77.78
	t/s	327	281	242

的基于样本点估计的支持向量机,从理论上做了相应的推导,并在实验的基础上,得出新的支持向量机比以往传统的支持向量机在分类的精度,以及在数据样本的训练时间上均有所改善,即新的支持向量机有更好的性能。

参考文献:

[1] Burges C J C.A tutorial on support vector machines for pattern recognition[J].Data Mining and Knowledge Discovery,1998,2(2):121-167.

[2] Lin Chih-Jen.A formal analysis of stopping criteria of decomposition methods for support vector machines[J].IEEE Transactions on Neural Networks,2002,13(5):1045-1052.

[3] Keerthi S S,Shevade S K,Bhattacharyya C.A fast iterative nearest point algorithm for support vector machine classifier design[J].IEEE Transactions on Neural Networks,2000,11(1):124-136.

[4] Kumar R,Jayaraman V K,Kulkarni B D.An SVM classifier incorporating simultaneous noise reduction and feature selection:illustrative case examples[J].Pattern Recognition,2005,38(1):41-49.

[5] Lee KiYoung,Kim Dae-Won,Lee K H.Possibilistic support vector machines[J].Pattern Recognition,2005,38(8):1325-1327.