文章编号:1000 - 582X(2003)02 - 0060 - 03

数据仓库动态设计与维护中与/或有向图方法*

应新洋,郭　平,蒋　渝,吴元洪,林　勇
(重庆大学 计算机学院,重庆 400044)

摘　要:数据仓库(DW)可以抽象地看成基本关系上的一组具体的视图,是一个随时间不断推移变化的动态实体。当有新的查询出现时,可以将它化成具体的视图表示并加入到数据仓库中。将数据仓库表示成与/或有向图能够较好地描述数据仓库中各种查询(视图)之间的关系,这样数据仓库动态设计问题就可以模拟成状态空间搜索问题,相应的转换规则就可以定义成状态转换规则,通过状态空间搜索的转换和化简来实现数据仓库的动态设计。讨论了将视图加入到数据仓库中的动态设计方法,并通过一个实际的例子介绍了这种动态数据仓库的建立过程。

关键词:数据仓库;数据仓库动态设计;与/或有向图
中图分类号:TP311.13　　**文献标识码**:A

数据仓库(DW)的数据来自许多方面[1],包括巨大的、分散的、异构的数据库和其它信息源。一方面,数据仓库可以抽象地看成基本关系(数据源)上的一组具体的视图,另一方面,在数据仓库的应用中,如 OLAP(在线分析处理)和 DSS(决策支持系统),大量地使用了分组和集合查询,这些查询多采用专用的视图表示在数据仓库中。

随着时间的推移,数据仓库(DW)中新的查询不断出现,如何将相应的视图加入到数据仓库中,以扩展数据仓库(DW)的应用是数据仓库设计中的关键问题之一。传统的方法采用静态处理,即重新完成一次包括新老视图的重建。其缺点是:

1)确定适当的一组视图是一个很长的复杂过程;

2)在视图的确定过程中,可能有旧的视图要移去,新的视图要加入数据仓库(DW)中。

笔者提出了一种动态处理方法,即用附加的空间来存储需要加入到数据仓库(DW)中的具体的视图。这样动态处理方法的特点是:

1)新的视图存储在附加的处理空间;

2)所有的新的查询都用专用的视图表示;

3)基于具体视图上新的查询时间和空间总代价是最小的。

1　查询与视图的有向图表示[2]

数据仓库中的查询可以分为两类,一类是利用已有的视图直接进行查询,另一类是建立在视图基础上的查询。显然第 1 种形式是第 2 种形式的特例,下面只讨论第 2 种形式的查询。

查询在关系代数中是由若干基本操作构成的。考虑如下例子。

例 1　有如下基本关系,分别记为 D、E:

Department(DeptID, DeptName)和 Employee(EmpID, DeptName, Salary, DeptID),在 D、E 上的查询 $Q_1 = \sigma_{salary} > 1\,000(E \infty D)$,

$Q_2 = <DepartID, DepartName> F <AVG(Salary)> (\sigma_{salary} > 1\,000(E \infty D))$可以表示为图 1。

其中 F 为一种关系运算,○为状态节点(equivalence),也称为 OR(或)节点,表示该节点代表的查询结果可由到达该节点的任意弧提供。状态节点用相应的代数表达式表示。○为操作节点(operation),也称为 AND(与)节点,表示该节点的操作过程,操作节点用相应的代数操作表示。

查询有向图是一个直接带有根节点并表达了节点之间代数关系的非循环图。在数据仓库(DW)中视图变化的重写(rewrite)就可以用与/或有向图来表示。在

* 收稿日期:2002 - 09 - 08
基金项目:国家十五攻关资助项目(2002BA107B)
作者简介:应新洋(1977 -),男,浙江绍兴人,重庆大学硕士研究生,研究方向为数据仓库与数据挖掘,GIS。

有向图中，所有的根节点都是查询节点。

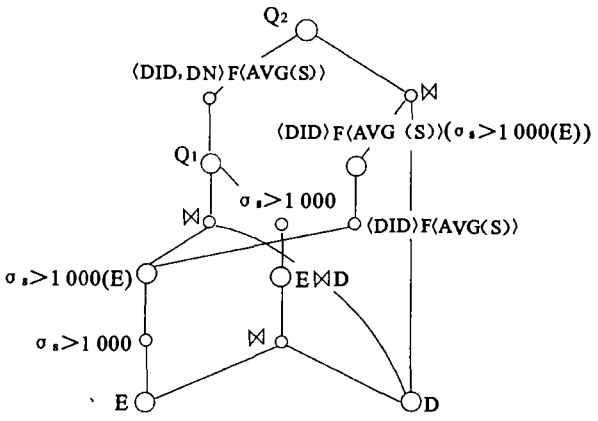

图 1　查询 Q_1，Q_2 的与或有向图表示

2　有向图的状态转换规则[3-6]

数据仓库(DW)中，查询和视图的与或(OR/AND)有向图展示了查询的多种构造过程，同时也反映了查询与基本表，中间与结果间的代数运算关系。若在包含基本表 D、E 的数据仓库中，已建立了查询 Q_1(作为视图保存在数据仓库中了)，当 Q_2 是要建立的新查询时，可以用基于 Q_1 的与/或(OR/AND)有向图(图 2)来表示 Q_2。图 1 和图 2 之间反映了数据仓库中新查询的过程。

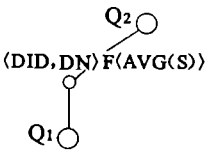

图 2　基于 Q_1 的新查询 Q_2

由此，可以得到 2 个结论：

结论 1　数据仓库可由与/或有向图表示。

结论 2　数据仓库的建立与修改可通过修改相应的与/或有向图完成。

与/或有向图表示了数据仓库的状态，有关数据仓库动态设计的问题也就变成了该状态空间的搜索与转换的问题。

为叙述方便，先给出如下概念。

定义 1　与/或有向图的查询空路径指从查询节点 Q(即状态节点)到一般节点 V(包括状态节点和操作节点)中没有其他查询节点的查询路径。

定义 2　查询 Q 在 V 上的重写指在与/或有向图中查询 Q 在已有的视图 V 基础上建立视图的整个构建过程，记为 Q^V。

一般地，可以得到以下 2 条状态转换规则。

转换规则 1　设 Q 是与/或有向图 S 中的一个查询节点，V 是 Q 的一个非下继节点，若满足条件：

1)S 中有一条从 Q 到 V 的查询空路径，或 Q、V 是同一点；

2)没有从 V 到非基本关系上的非标注的下继节点的路径。则移去所有的 V 以下的边和非标注的或非查询的节点，除非这些点和边在一条从查询节点出发的不包含 V 的路径上；移去所有的 Q 以下的边和非标注的或非查询的节点，除非这些点和边在一条从查询节点出发的不包含 Q 的路径上或在 Q 到 V 的查询空路径上。

转换规则 2　设 Q 是与/或有向图 S 中的一个查询点，V 是 Q 的一个非基本关系节点，若满足条件：

1)S 中有一条从 Q 到 V 的查询空路径；

2)有不包含 Q 到 V 的查询路径的 Q 的有向图。

则移去所有的 Q 的以下的边和非标注的或非查询的节点，除非这些点和边在一条从查询节点出发的不包含 Q 的路径上或在 Q 到 V 的查询空路径上。

下述例子说明了转换规则及其使用过程，可以用图 3 来表示例子 1(即图 1 的简化)。

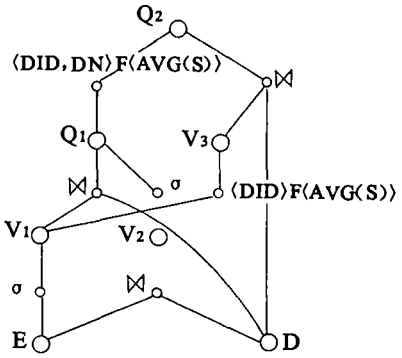

图 3　查询 Q_1，Q_2 的与或有向图表示

这里 Q_1，Q_2 为查询节点，D 为标注节点(即有初始视图 $V_0=\{D\}$)，运用转化规则可将状态图化简。

$V_1=\sigma_{salary}>1\ 000(E)$；

$Q_1=V_1\infty D$；

$Q_2=<DeptID,\ DeptN>F<AVG(Salary)>Q_1$；

Q_2 在 Q_1 上的重写为 $Q_2^{Q_1}$，相应地，Q_1 在 V_1 的重写为 $Q_1^{V_1}$，Q_2 在 V_1 的重写为 $Q_2^{V_1}$。

图 4 是查询 Q_1 在旧视图 V_1 上的重写，即 $Q_1^{V_1}$，运用转换规则 1(在图 3 的基础上)。图 5 是查询 Q_2 在旧视图 V_1 上的重写，即 $Q_2^{V_1}$，满足转换规则 1 的条件，运用规则 1(在图 3 的基础上)。

图 6 是查询 Q_2 在旧视图 Q_1 的重写，即 $Q_2^{Q_1}$，运用转换规则 2(在图 4 的基础上)，全图反映了查询 Q_1、Q_2 的整个重写过程。

图 7 是查询 Q_2 在旧视图 Q_1 上的重写，即 $Q_2^{Q_1}$，运用转换规则 1(在图 5 的基础上)。

通过上述例子的转换，设定状态图中(即数据仓库

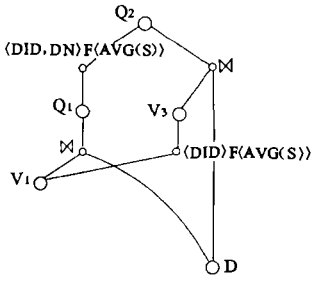

图 4 （图 3 基础上的）$Q_1^{V_1}$

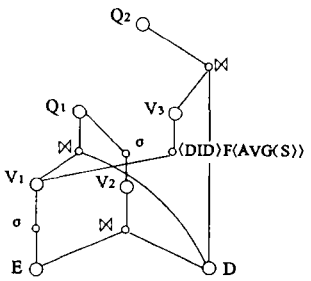

图 5 （图 3 基础上的）$Q_2^{V_1}$

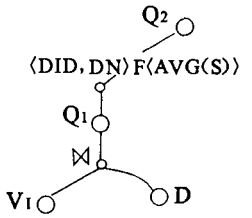

图 6 （图 4 基础上的）$Q_2^{Q_1}$

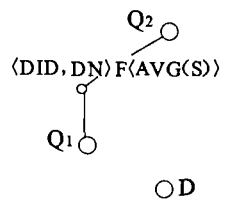

图 7 （图 5 基础上的）$Q_2^{Q_1}$

设计时）的旧视图，新的查询进入时，通过转换可得到唯一的最佳解（图 6 和图 7 的本质一样，图 6 中列出了 Q_1 查询是如何得来的过程，图 7 直接将 Q_1 作为一个旧视图，也就是已知 Q_1 是旧视图的话，查询 Q_2 的唯一表达便如图 6）。

通过以上的转换规则，可以方便地进行数据仓库设计的动态维护，并且可得出以下结论：

1）可以通过 AND/OR（与或）有向图表示多个查询的方法，将数据仓库动态设计问题模拟成状态空间搜索问题，所有的查询视图都可以在数据仓库（DW）的与或有向图中表示，相应的转换规则就可以定义成状态转换规则。

2）这种状态转换是可靠并且全面的，通过这些转换规则，可以从原始状态得到目标状态（一个最优的结果）。

3）可以设计状态转换时间和空间代价的运算规则。

4）这种动态的方法也适用于设计数据仓库（将空的视图作为旧视图）。

3 结 论

数据仓库的动态设计与维护是数据仓库应用中的基本问题。数据仓库表示成与/或有向图能够较好地描述数据仓库中各种查询（视图）之间的关系，这样数据仓库动态设计问题就可以模拟成状态空间搜索问题，相应的转换规则就可以定义成状态转换规则，通过状态空间搜索的转换和化简来实现数据仓库的动态设计。

参考文献：

[1] HAMMERGREN TOM 著. 数据仓库技术[M]. 曹增强，王备战，岳晓奎 译. 北京：中国水利水电出版社，1998.

[2] HAN JIA WEI, KAMBER MICHELINE. 数据挖掘——概念与技术（影印版）[M]. 北京：高等教育出版社. 2001.

[3] DIMITRI THEODORATOS. TIMOS SELLIS. Dynamic Data Warehouse Design [A]. Data Warehousing and Knowledge Discovery[C]. Germany: Springer-Verlag LNCS, 1999. 1 - 10.

[4] CHAUDHURI S. DAYAL U. An Overview of Data Warehousing and OLAP Technologies[J]. ACM SIGMOD Record. 1997, 26(1): 65 - 74.

[5] RAMSAK F, MARKL V, FENK R. Integrating the UB - Tree into a data - base system[A]. VLDB2000[C]. San Fransisco: Morgan Kaufmann, 2000. 263 - 272.

[6] WU MING - CHUAN. BUCHMANN ALEJANDRO P. Encoded Bitmap Indexing for Data Warehouses [A]. Proceedings of the Fourteenth International Conference on Data Engineering [C]. Orlando, Florida: IEEE Computer Society, 1998. 220 - 230.

And/or Dag Method in Dynamic Design and Maintenance of Data Warehouse

YING Xin-yang, GUO Ping, JING Yu, WU Yuan-hong, LIN Yong

(College of Computer, Chongqing University, Chongqing 400044, China)

Abstract: A data warehouse (DW) is taken as a set of materialized views defined over base relations. It is dynamic entity that evolves continually over time . As time passes, new queries can be made into new views to be added to the DW. Data Warehouse is expressed as and/or dag can describe the relations between queries in DW preferably. Thus, the design of DW can simulate the problem of state space search, relevant exchange rule may define state exchange rule. It can achieve the dynamic design of DW by exchanging and predigesting of state space search. A dynamic design way is proposed to add the new views to the DW by an example.

Key words: data warehouse; data warehouse dynamic design; and/or dag

（责任编辑 张 苹）