

面向关系模式——一种全新的数据库框架^{*}

Relation Oriented Model——A New Data Model Frame

吴怀谷 周明天

(电子科技大学计算机学院 成都610054)

Abstract In order to organize and transform data relations dynamically according to the daily requirements, a new data model, Relation Oriented model (RO model) is proposed. RO model is described at the physical, data, relational and external schema. In physical schema, a middleware mechanism is used to realize the distributed processing. In data schema, data group (DG) is defined as the independent atomic unit of data organizing. In relational schema, the data relations can be organized and transformed dynamically, even on runtime environment. In external schema, RO model supports the workflow. The building of RO model will reduce the overhead in repeating developments of database applications to a large extent.

Keywords Relation Oriented model (RO model), Physical schema, Data schema, Relation schema, External schema, Data group (DG), Dynamic data organizing, RODBMS

1. 引言

从 E. F. Codd 提出关系数据库模式以来,数据库中的数据关系尽管可以通过一些关系运算进行一定程度的变化,数据关系的组织方式在数据库定义时就已经相对固定了,这种方式在 RDBMS 中体现为表和属性,而在 OODBMS 中则体现为对象和属性(Properties)。这种关系和数据的绑定极大地限制了数据组织的灵活性,并增加了数据的冗余。现实中的数据处理要求更高级的统计分析、决策支持、知识发现以及对工作流的支持。而与之相适应的高级数据库工具,如数据仓库、数据挖掘和基于工作流的 WFMS 等,都是收集传统 DBMS 上的数据并按某种预定模式进行独立的处理。它们有共同的突出缺陷:关系不能动态转换,不能反映及时的变化,元数据不能被操纵,更不能对元数据库做出反馈。现有的这些工具只是对传统 DBMS 的一种妥协方案。只有将高级关系处理功能与底层数据库结合为一个整体,才能实现能真正动态组织数据的 DBMS。本文中动态包括以下两层含义:可按实际需要变化,可在运行时变化。

数据存储的焦点或许是数据本身,但数据操作的核心应该是关系,因为数据只代表历史,只有关系才能指导将来。一种新的数据库框架——面向关系模式 (Relation Oriented Model) 正是在这种背景下产生的。

2. 面向关系模式的目标

面向关系模式 (RO 模式) 的主要目标是动态反映

实际工作中的复杂数据组织关系。在 RO 模式中,数据库应用编程的核心将是数据之间的关系。RO 模式应解决的主要问题是:

- 具备良好的伸缩性,可简化为传统数据库,也可实现具备高级分析功能的复杂数据库。
- 为适应实际工作的灵活需要,RODBMS 应具有数据关系的灵活定制功能。
- 应该具备良好的动态关系重组和转换功能。能动态修改数据库结构。
- 应该提供对高级事务模型 (ATM) 和工作流的支持。

3. RO 模式的体系结构

RO 模式重点在于动态数据组织,它也是解决上述问题的核心。要实现动态数据组织,首先必须改变传统的绑定方式,把数据和关系分离。RO 模式的体系结构按如下四层体系描述:物理层、数据层、关系层和外部接口层,如图1。

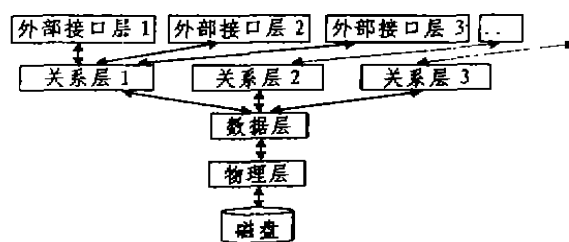


图1 关系模式体系结构图

^{*} 本文的工作得到电子科学研究院基金的支持。

3.1 物理层

物理层详细说明数据库在物理存储设备上存储的细节,对数据库设计人员和编程人员是透明的,由 DBMS 直接管理。RO 模式在数据层和物理层之间嵌入了一种类似 CORBA 的机制。如图2所示,该中间件完成数据层的请求在分布式环境中的分发。一个 RO 模式的 DBMS 只需要在主机上保存一个数据层的数据组 DG(参见“数据层”)的映射,具体的数据可以分布存储在远程存储设备上。

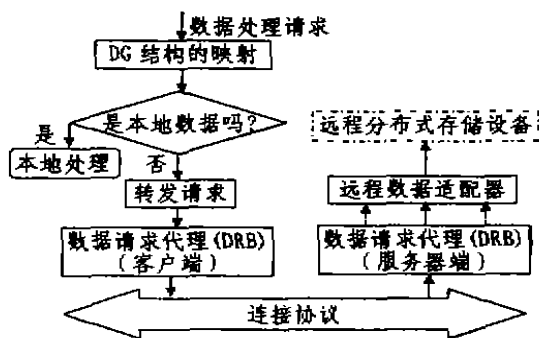


图2 分布环境下的数据请求流程图

3.2 数据层

数据和关系的分离不能简单地分离数据的存储和操作。否则,将极大地增加数据库设计人员的负担。因此,数据需要一种基本的组织方式,数据层就是用来描述基本数据组织的。数据组织的基本单位叫做数据组(DG),DG 由具有共同性质的一组数据组成,如人的姓名、身高或照片,类似于传统数据库中表的属性。为了处理上的一致,DG 的数据类型可以用数据组定义语言(DGDL)定义为对象。每个 DG 上可以定义独立的方法,本文称为纵向方法。所有 DG 对象从基本 DG 对象 DGObject 继承,DGObject 上定义了单个 DG 上的标准纵向方法,主要包括:

- 单 DG 的基本数据操作方法,主要包括输入(input)、刷新(update)、删除(delete)、前移(previous)、后移(next)、到首数据项(first)和到尾数据项(end)等。
- 单 DG 上基本的关系操作,如选择(select)、分组(group)和排序(sort)等。
- 单 DG 上通用的算术运算,如求和(sum)、计数(count)、平均值(average)和标准差等。

用户可用 DGDL 自定义 DG 上的特殊处理方法。使用 DGDL,DG 可按如下方式创建:

```

Create DG DGname extends DGObject
Property {Item 1; Item 2; ...; Item n;}
Method {User Extended Methods; ...}
  
```

系统赋予每个 DG 一个唯一的 DGID,使 DG 可以被 DBMS 标识。独立的 DG 在 RODBMS 中组织成各

种关系主要表现为不同 DG 的数据项之间的连接。因此每个 DG 的每个数据项必须包含对其它 DG 的数据项的地址引用(AR)。DG 的组成方式如图3所示。构成 DG 数据项的基本准则是:

- 所有的数据都必须分组存储在 DG 中。
- 一个 DG 中的每个数据项都是不同的。
- DG 中的数据项至少在基本的数据输入操作和最常用的普通查询中应该是原子数据。
- 为了简化 DG 的处理,DG 对象的属性的类型不能是结构类型,而必须是基本类型。
- DG 提供对 NULL 的支持。

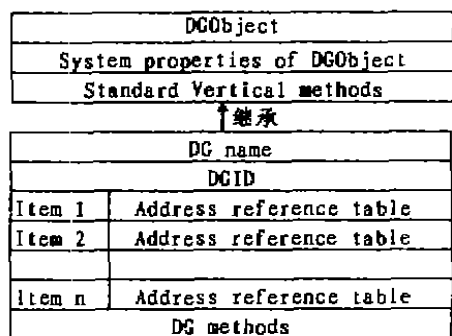


图3 数据组结构图

按照上述准则,DG 的域完整性由其所有属性的值域决定,表示如下:

$$D = \{t_{p1} \times d_{p1}, t_{p2} \times d_{p2}, \dots, t_{pn} \times d_{pn}\}$$

其中,D 表示 DG 的值域, t_{pi} 表示第 DG 的第 i 个属性 P_i 的类型, d_{pi} 表示 P_i 的值域。DG 的值完整性由准则2和3来保证。

DG 无冗余地独立存储在二级存储设备(the secondary storage)上。任何 DG 的增加和删除都不会影响系统其它 DG 的数据。DG 是分离数据和关系的核心。

以销售数据库为例。销售记录包括商品、时间、数量、价格、分店位置等信息。这些销售记录可按如图4所示组织成 DG。

3.3 关系层

关系层描述 DG 上的所有关系并完成对数据的操作,使数据库的应用编程全部面向关系,是 OR 模式的核心。这些关系定义中通用的关系处理可以由系统提供,而更多高级的关系则将由数据库设计者用一种面向对象且兼容 SQL 的关系定义语言(RDL)定制。同时,还可提供由 DG 组成兼容传统数据模式的标准连接模板,包括:关系连接,网状连接,层次连接。用户自定义关系实现了数据组织的多样性,同时也损失了数据的参照完整性,增加了数据约束方面的工作负担。在

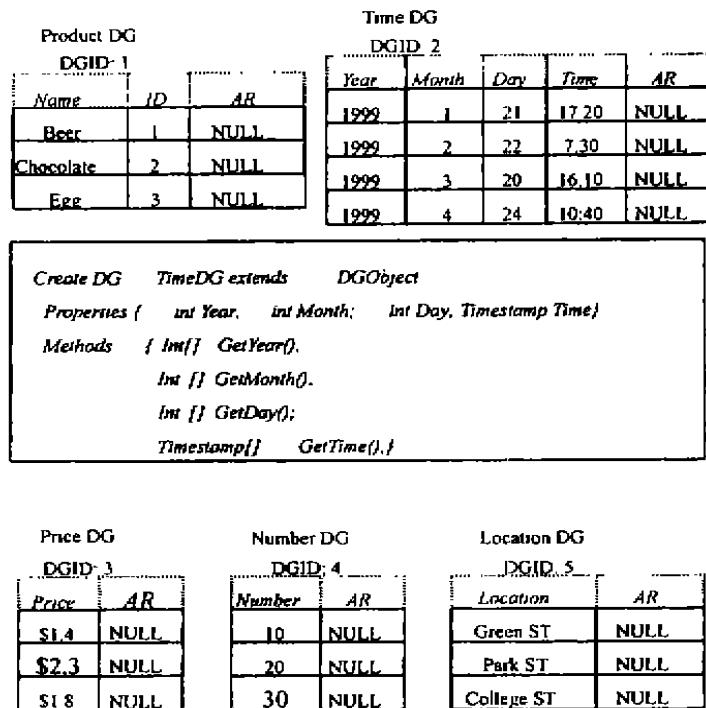


图4 销售数据库 DG 组成图

RO 模式运用于具体的行业时,还应该建立基于行业的标准约束方法。

对数据库的操作是对由按一定关系组织的数据的操作,因此,对数据库的操作主要表现为对关系的操作。RO 模式中的关系表示为 DG 的连接和对连接后的 DG 的操作。可以将一个事务看作一个对象,该事务所涉及的 DG 是对象的属性,事务所包括的 DG 的连接以及在此连接上的操作表示为对象的各种方法,为了保证事务的实体完整性,事务对象必须包含一个 DG 间的连接方法(DGConnect)。一种特殊的情况是,一个事务对象可以只包括一个 DG。此时,DGConnect 为空操作,用 RDL,可按如下方式表示事务对象的创建:

```

Create TransactionObject TransactionName
Properties {Property DG 1;
  
```

```

Property DG 2;
.....
Property DG n;}
Methods {void DGconnect();
User Extended Methods.....}
  
```

在 RO 模式中,这些建立在 DG 连接基础上的方法称为横向方法(Horizontal Method)。所有 OLAP、OLTP 等高级统计关系和复杂的操作,如数据清洗、提取、数据校验和多维数据查询等都将通过定义横向方法来实现。我们可以用一种 DG 关联组织方法输入数据,而用另一种 DG 关联组织方法查询数据,甚至,我们可以在运行时重组 DG 的组织方式。在横向方法中用于在分析后对元数据和关系进行反馈的方法称为反馈方法。

在上面的销售处理中,输入关系可以是一个简单的一维关系,如图5。这种关系就象 POS 上的流水账记录:商品、数量、价格、销售时间和分店。



图5 销售数据库一维输入关系图

当我们需要进行数据分析时,关系可以动态转化为三维关系,并同时取消无用信息的连接(本例中无用

信息为价格)而加快分析的速度。三维查询关系见图6。查询结果表示了月份和分店位置对商品销售数量的影

响。

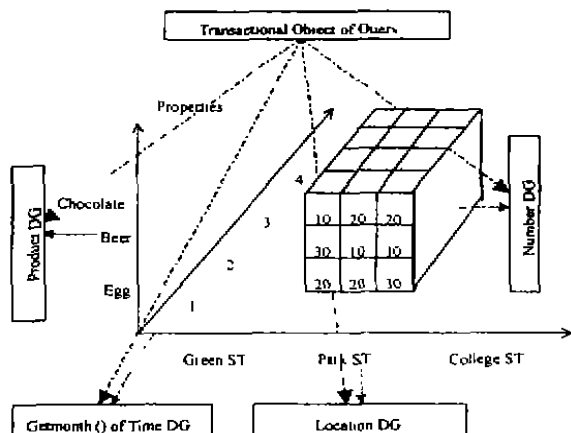


图6 销售数据库三维查询关系图

3.4 外部接口层

外部接口层在 RO 模式中有两层作用：

- 外部接口层允许数据访问在关系层上按用户的

需要被进一步定制。该层的关系定义通常是关系层指定关系的子集，是一种临时的关系，类似于 RDBMS 中的视图。

·外部接口层更重要的作用是实现 RO 模式对工作流的支持。工作流是一组相关的事务组成的工作流程。这组事务是为完成一个特定的任务而按某种合作的关系组成的一个整体活动(Activity)。使用特定的事务定义语言(TDL)将活动作为对象，则可在外部接口层将关系层定义的事务对象作为活动的属性，事务合作的关系定义为活动的方法。

```
Create ActivityObject ActivityName
Properties!Property transaction 1;
Property transaction 2;
...
Property transaction n;}
Methods{void transaction();
User Extended Methods...}
```

通过这种方法，RO 模式可以在数据库中对高级事务模型和工作流的支持。近几年，ATM 在数据库界进展缓慢的原因是传统数据库的死板不能适应 ATM 的灵活变化。现在 RO 模式有望在这一领域取得进展。例如如图7中所示的根据销售下定单的活动。

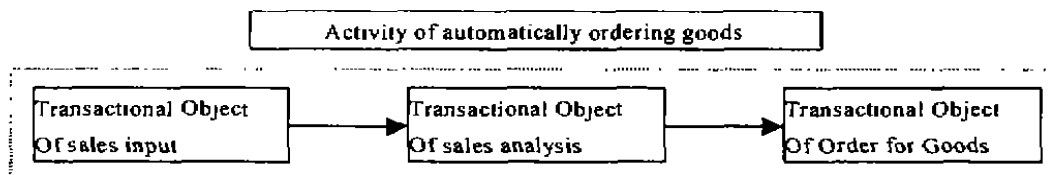


图7 下定单活动组织图

4. 比较 RO 模式与其它数据模式

4.1 RO 模式与关系模式

关系模式对数据的组织和约束进行了严格的控

制，随着数据库应用的普及，现实中许多复杂的数据关系的处理对这种绑定的模式提出了极大的挑战。相比之下，RO 模式更灵活，更适应复杂数据处理。两者的主要差别主要体现在以下几方面：

	关系模式	RO 模式
数据处理能力	以关系代数作为处理数据关系的主要手段。具有严格的数据约束和良好的完整性。缺乏复杂数据处理功能，尤其是统计分析能力。	由用定义处理数据的方法，可以进行复杂数据处理。没有数据的参照完整性，增加了数据不一致的可能性，需要特定的行业标准。
动态数据组织	主要通过 select 重组数据关系，操作方便，但是不能改变数据组织的角度。简单地说，按一维方式组织的数据难以转化为多维数据组织。	DG 的动态组织可以保证数据组织方式的动态建立、重定义和转化。不过这种复杂的转化存在运行效率的问题，需要妥善的转化算法。
数据冗余	用表组织数据。由于很多属性，尤其是关键属性可能属于多个表，因而不可避免地有数据冗余。	用独立存储的 DG 组织数据，可以做到零冗余，适用于存储多媒体数据，增加了地址引用的连接冗余。
个性化应用	使用关系数据库的应用必须遵从由数据库提供的标准，从而限制用户自身特色的个性化应用。这正是数据库应用不容易普及的关键原因。	每个 DG 的变化都不会影响无关 DG 的数据，使 RO 模式上的应用能够方便地建立个性化的处理。大大减少了重复开发应用的代价。

4.2 RO 模式与面向对象数据库

RO 模式使用了面向对象的概念，但是它和面向

对象数据库在两个方面有很大的差别：

	面向对象数据库	RO 模式
使用对象概念的范围	对象主要用来组织数据	具有层次化的对象模型。数据、关系和事务的组织均是面向对象。
数据库设计和处理的重心	重心数据存储本身。面向对象数据库是对象的一种持久化。	重心是 DG 间关系的组织和转化,关系是 RODBMS 的核心。

4.3 RO 模式与数据仓库

的复杂数据分析能力,其主要区别如下:

RO 模式支持 OLAP 和 OLTP,实现了数据仓库

	数据仓库	RO 模式
灵活的数据组织	数据仓库往往按固定的算法确定要提取的数据和结构。缺乏灵活性。适合算法固定的工程数据处理。	RO 模式可以按多种算法动态重组数据关系。方法改变灵活。尤其适用于数据挖掘中的试探性算法。
元数据的来源	元数据来自基本的传统数据库,数据仓库缺乏对元数据的控制能力。	元数据来自 RODBMS 的 DG,关系组织可以自由控制元数据。
分析的反馈能力	数据处理的结果并不能直接反馈到元数据库。	处理结果可以通过反馈方法直接对元数据进行反馈。

结束语 RO 模式最大的特点在于它独特的动态数据组织能力。RODBMS 具有出色的可伸缩性,既可组织成传统数据库,又可扩展为适应实际工作的复杂数据库,甚至可作为可动态操作的数据仓库。RO 模式未来的研究重点在以下三个方面:①RO 模式是一个复杂而灵活的数据模式,效率是它最主要的问题;②RDL 的实现。怎样实现动态关系的建立和转化?③DGD L 的实现,DGD L 具体怎样组织数据?

参 考 文 献

- 1 Ramakrishna R, Gehrke J Database Management System (Second Edition). 清华大学出版社,2000
- 2 王能斌.数据库系统原理.机械工业出版社,2000
- 3 Ambler S W. Mapping Objects To Relational Databases SIGS Books/Cambridge University 发行,1998,版本号:1999年2月26日。Available at: <http://www.AmbySoft.com/mappingObjects.pdf>

(上接第82页)

本数据的最小 I-图,从实验结果可以看出,该算法是有效的,由于 Markov 网是一个无向图,不能表示属性之间的因果关系,如何由发现的 Markov 网得到等价的 Bayesian 网是我们下一步研究的课题。

参 考 文 献

- 1 Pearl J Probabilistic reasoning in intelligent systems. Morgan Kaufmann,1988
- 2 Xiang Y, Wong S K M. Learning conditional independence relations from a probabilistic model. [Technical report]. University of Regina,1994

- 3 Suzuki J. Learning bayesian belief network based on the MDL principle. An efficient algorithm using the branch and bound technique. In: Proc. of the intl conf on machine learning, Bari, Italy, 1996
- 4 Herckerman D, Geiger D, Chickering D M Learning bayesian networks: the combination of knowledge and statistical data: [Technical Report MSR-TR-94-09]. Microsoft Research, 1994 To appear, Machine Learning Journal
- 5 Cheng J, Bell D A, Liu W. An algorithm for bayesian belief network construction from data. In: Proc. of sixth AI & STAT'97 (pp. 83-90). Ft. Lauderdale, Florida, 1997