

基于 SQL Server 2000的企业数据仓库设计与应用

仲元昌¹ 李 刚² 王 越² 杨 武²

(重庆大学通信工程学院 重庆400044)¹ (重庆工学院计算机系 重庆400050)²

Design and Application of Enterprise Data Warehouse Based on SQL Server 2000

ZHONG Yuan-Chang¹ LI Gang² WANG Yue² YANG Wu²

(College of Communication, Chongqing University, Chongqing 400044)¹

(Department of Computer, Chongqing Institute of Technology, Chongqing 400050)²

Abstract To make up the shortage of current database in information distilling, we need to integrate mass data in data warehouse so that it can offer better reference to business manage and decision-making for enterprise. By using relational database SQL Server 2000 of Microsoft as resolution scheme of data warehouse, an enterprise data warehouse with star schema is established. Data cubes are constructed and measures are calculated by Analysis Services tools in SQL Server, and these cubes are online analyzed through OLAP, then user can easily get relationship of data or attributes which can provide reference to their business, and utilize data in existence fully and effectively.

Keywords Data warehouse, Star schema, Cube, OLAP

在激烈的市场竞争中,企业纷纷建立了自己的数据库系统,以此来收集、存储、管理业务操作数据,改善办公环境,提高操作人员的工作效率,同时,企业也希望从海量的数据中获得有价值的信息,以支持企业决策。但是,传统的数据库应用系统是面向业务的,无论是查询、统计,还是生成报表,其处理方式都只是对指定的数据进行简单的数字处理,这虽然减轻了具体操作人员的劳动强度,但不能对这些数据所包含的内在信息进行提取,对企业的中高层来说并没有相应的决策支持系统。企业需要新的技术来弥补原有数据库系统的不足,需要把已经广泛收集到的数据集成到数据仓库中,以从业务数据中提取有用的信息,帮助他们在业务管理和发展上做出及时正确的判断。

数据仓库是一个面向主题、集成的、不可更新的、时变的数据集合,它支持管理部门的决策过程^[1,2]。数据仓库系统在数据分析和决策方面为用户或“知识工人”提供服务,称为联机分析处理(On-Line Analytical Processing, OLAP)系统。而关系数据库系统的主要任务是执行联机事务处理和查询处理,称为联机事务处理(On-Line Transaction Processing, OLTP)系统^[3,4]。

由于在当前企业中微软(Microsoft,以下简称MS)的关系数据库软件SQL Server 2000(以下简称SQL Server)应用得相当广泛,并能与SQL Server数据仓库很好地衔接,企业选择SQL Server作为数据仓库,可以实现不同格式的数据共享,有利于企业内不同系统之间的整合。并且SQL Server的操作简单易行,也易于项目的迅速实施,因此本文以SQL Server作为国内某自动化设备制造企业数据仓库解决方案讨论。

1 SQL Server 数据仓库解决方案

MS SQL Server 作为一套完全的关系数据库和数据分析解决方案,具有以下特点:1)与Internet集成,提供完备的

Web功能;2)高度可扩展性和高可用性;3)企业级数据库功能;4)易于安装、部署和使用,可加速应用开发;5)作为微软管理控制台(MS Management Console, MMC)的管理单元组件,提供日程安排、存储管理、性能检测、报警和通知的核心管理服务。因此,用户可以利用SQL Server快速创建下一代的可扩展电子商务和数据解决方案。更重要的是,SQL Server中集成了OLAP功能,包含了多个可用于生成有效地支持决策支持处理需求的数据仓库的组件,提供可扩充的基于COM的OLAP接口。MS Office套件中的Access和Excel可以作为数据展现工具,另外SQL Server还支持第三方数据展现工具。

SQL Server数据仓库组件或工具主要有:1)数据仓库框架;2)数据转换服务(Data Transformation Services, DTS),提供数据输入/输出和自动调度功能,在数据传输过程中可以完成数据的验证、清洗和转换等操作,并通过集成MS Repository,共享有关的元数据;3)知识库(Repository),存储包括元数据在内的所有中间数据;4)在线分析服务(Analysis Services);5)数据透视表服务(PivotTable Services),提供客户端OLAP数据访问功能,通过这一服务,开发人员可以用VB或其他语言开发用户前端数据展现程序,并允许在本地客户机上存储数据。

2 SQL Server 数据仓库框架

SQL Server数据仓库框架是一组实现数据仓库功能的组件和应用程序接口(API),它提供一个公用界面,力求供各种生成并使用数据仓库或数据集市组件使用。SQL Server的数据仓库框架集成了数据仓库的许多组件,如:业务系统数据源、设计/开发工具、数据析取和转换工具、数据库管理系统、数据存取和分析工具、系统管理工具。它不仅是SQL Server等MS产品开发的路标,也是与其它软件集成所需的路标。

仲元昌 讲师,硕士研究生,主要从事计算机应用,通信信号处理,载波通信,医用仪器等领域的教学与研究。

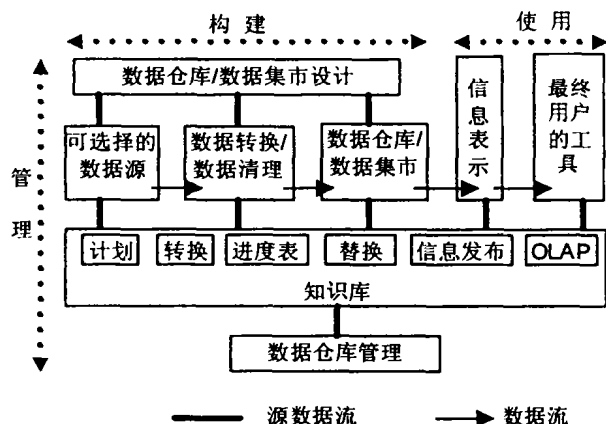


图1 SQL Server 2000数据仓库框架

SQL Server 数据仓库框架如图1所示,其目标是简化数据仓库解决方案的设计、实现和管理。该框架提供:1)与第三方供货商集成和易于由第三方供货商扩展的开放式体系结构;2)异类数据导入、导出、确认和带可选数据类型的清理服务器;3)集成的元数据用于数据仓库的设计、数据析取/转换、服务器管理和最终用户分析工具;4)用于日程安排、存储管理、性能检测、报警/时间和通知的核心管理服务。

数据仓库框架从底层向上设计,为 Microsoft 产品用户和第三方企业提供行业标准技术,很容易扩展开放式体系结构,使得组织机构能够选择同类中最好的组件并确保集成。

3 多维数据构架的设计

数据仓库设计中最重要任务就是其数据构架的设计,目前最流行的数据仓库构架是多维数据构架,主要有以下三种:1)星型架构(star schema),它由一个规模很大的无冗余数据的中心表(事实数据表, fact table)和一组规模较小的附属表(维度表, dimension table)所组成。这种架构很像星形爆发,维度表围绕事实数据表显示在射线上。该结构中,在位于架构中心的事实数据表中维护数据,其它维度数据存储在维度表中。每个维度表与事实数据表直接相关,且通过一个键列联接到事实数据表。2)雪花型架构(snowflake schema),它是星型架构的变种,其中某些维度被进一步规范化表示为主维度表和附属维度表,只将主维度表与事实数据表联接,其它维度表联接到主维度表,其构架形成类似于雪花的形状^[5]。3)星系架构(starflake schema),复杂的应用可能需要多个事实数据表共享维度表,称为星系构架^[1]。

通过分析比较发现,目前该企业的客户关系管理系统涉及的数据虽然多,但主要是以客户基本信息 Customer 表为核心,联系着客户的其他信息,各个事实表之间的关系并不复杂,因此首先排除了星系构架。其次,虽然雪花型构架采用规范化形式,可以减少冗余,但其空间节省并不是很大,而由于执行查询需要更多的连接操作,还会降低查询的性能。相比之下,采用星型构架虽然会带来一些冗余,但考虑到整个数据库并不是太大,能够提供足够的磁盘空间,因此采用星型构架。

数据仓库的建立主要是为了从中挖掘出有用的客户信息,所以在建立数据仓库的时候,只选择了一些有意义的信息,其他的信息如电话号码等则没有选取。设计好的数据仓库构架如图2所示。

在这里,我们采用对一个客户的销售信息作为事实数据表, Enterprise-potential(企业的预计销售额), Enterprise-

sold(企业实际销售额), Competitor-Potential(竞争对手预计销售额), Competitor-sold(竞争对手实际销售额)作为度量值。该模型用数据挖掘查询语言(DMQL)定义如下:

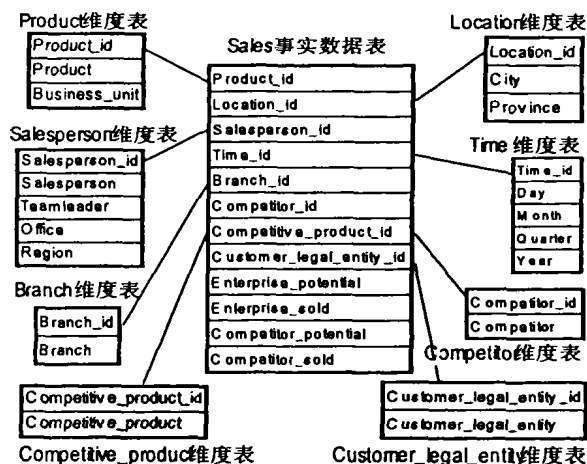


图2 销售和预测信息数据仓库的星型构架

```

Define cube Sales_star [Product, Location,
Time, Salesperson, Branch, Competitor,
Competitive-product, Customer-legal-entity];
Enterprise-potential = sum(Sales-in-Enterprise-potential),
Enterprise-sold = sum(Sales-in-Enterprise-sold),
Competitor-Potential = sum(Sales-in-Competitor-Potential),
Competitor-sold = sum(Sales-in-Competitor-sold)
Define dimension Product
as (Product-id, product, Business-unit)
Define dimension Location
as (Location-id, City, Province)
Define dimension Time
as (Time-id, Day, Month, Quarter, Year)
Define dimension Salesperson
as (Salesperson-id, Salesperson, Office, Region)
Define dimension Branch as (Branch-id, Branch)
Define dimension Competitor
as (Competitor-id, Competitor)
Define dimension Competitive-product
as (Competitive-product-id, Competitive-product)
Define dimension Customer-legal-entity
as (Customer-legal-entity-id, Customer-legal-entity)

```

4 多维数据集的构造和度量值的计算

多维数据集(或称为数据立方体, cube)是指数据的集合, 这些数据组织并汇总到一个由一组维度(Dimension)和度量值(Measure)所定义的多维结构中, 它使得管理人员(用户)可以从不同的角度(维度)、通过不同的度量值来观察分析所关心的事实数据, 逐步摆脱了对固定报表的依赖^[6]。SQL Server 的 Analysis Services 作为用于联机分析处理(OLAP)和数据挖掘的中层服务器, 其服务器可以构造用于分析的数据多维数据集, 并提供对多维数据集信息的快速客户端访问。它将数据仓库中的数据组织成包含预先计算聚合数据的多维数据集, 以便为复杂的分析查询提供快速解答, 其用户可从多维数据集和关系数据源中创建数据挖掘模型, 还可以对这两类数据应用数据挖掘模型。

度量值是在多维数据集内, 基于该多维数据集的事实数据表中的某列的一组值, 是进行聚合和分析的主要值。它通过一个聚合函数对多维数据集的每一个点求值, 根据所用的聚合函数, 度量值可分为分布的、代数的、整体的三类。通过将前述的 DMQL 语言翻译成如下 SQL 查询, 可以产生所需要的 Sales_star 多维数据集。这里, 聚合函数 sum() 用于计算星型构架中的 Enterprise-potential, Enterprise-sold, Competitor-Potential 和 Competitor-sold 的度量值。

```

Select S. Product-id, S. Location-id, S. Time-id,
      S. Salesperson-id, S. Branch-id, S. Competitor-id,
      S. Competitive-product-id,
      S. Customer-legal-entity-id
Sum(S. Enterprise-potential), Sum(S. Enterprise-sold),
Sum(S. Competitor-Potential), Sum(S. Competitor-sold)
From Sales S, P Product, L Location, T Time,
      SP Salesperson, B Branch, C Competitor,
      CP Competitive-product, CL Customer-legal-entity
Where S. Product-id = P. Product-id
and S. Location-id = L. Location-id
and S. Time-id = T. Time-id
and S. Salesperson-id = SP. Salesperson-id
and S. Branch-id = B. Branch-id
and S. Competitor-id = C. Competitor-id
and S. Competitive-product-id =
      Competitive-product-id
and S. Customer-legal-entity-id =
      CL. Customer-legal-entity-id
Group by S. Product-id, S. Location-id, S. Time-id,
      S. Salesperson-id, S. Branch-id,
      S. Competitor-id, Competitive-product-id,
      S. Customer-legal-entity-id

```

由此形成一个8维度的销售多维数据集,如图3。

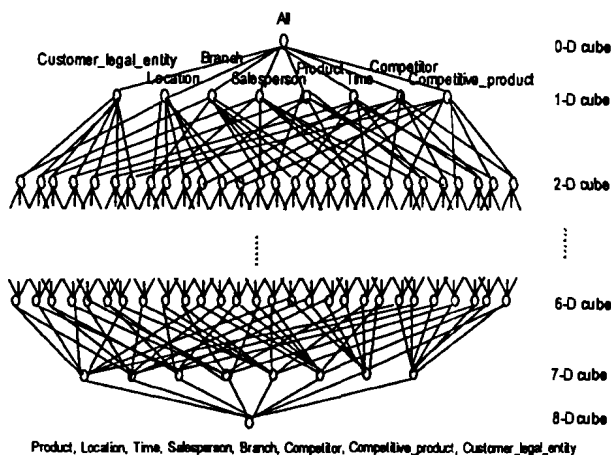


图3 8-D 销售多维数据集

以上查询创建的是 Sales_star 多维数据集的基本立方体,它包含多维数据集定义中说明的所有维度,其中每个维度的粒度在连接键层。通过改变 Group by 子句,可以产生 Sales_star 多维数据集的其他立方体。按 t. month 而不是按 s. Time-id 分组,将按月分组求和,得到度量值。去掉“Group by S. branch-id”,将按所有行业求和。去掉所有的 Group by 子句,将得到 Enterprise-potential, Enterprise-sold, Competitor-Potential, Competitor-sold 的总和。

5 多维数据集上的 OLAP 操作

在多维数据集中,可以用概念分层定义一个映射序列,将低层概念映射到更一般的高层概念,而这些映射则形成了一个层次在图3的多维数据集中,有4个维度具有概念分层,将这些维度中的属性进行全序相关,可以分别形成4个层次: Salesperson 维度: Salesperson < Teamleader < Office < Region; Product 维度: Product < Business-unit; Time 维度: Day < Month < Quarter < Year; Location 维度: City < Province。

根据概念分层,可以在多维数据集上进行 OLAP 操作,包括上卷 (roll-up)、深化 (drill-down)、切片 (slice)、切块 (dice)、数据透视 (pivot)、钻过 (drill-across)、钻取 (drill-through) 等。

对前述企业在广东和山东两省2003年的 AS、CD、ET 和 MC 四个产品事业部的预计销售值 Enterprise-potential (单位:万元)进行 OLAP 操作,可以得到相应的视图,这些操作都可以由 SQL Server Analysis Services 实现。

结论 通过建立以关系数据库 MS SQL Server 2000 为平台的企业数据仓库,并利用 SQL Server 的 OLAP 对所获得的多维数据集进行数据分析,可以更充分、更有效地利用企业现有的数据资源,方便地获得企业各业务数据或属性之间的关系,为企业决策者的业务管理和决策提供完整、及时、准确的决策参考信息。

参考文献

- 1 Songini M L. Real-time data warehousing tests IT's ability to serve businesses, users say[J]. Computerworld, 2003. 8
- 2 Whiting R. The data-warehouse advantage [J]. InformationWeek, 2003. 7
- 3 Kalnis P, Papadias D. Multi-query optimization for on-line analytical processing[J]. Information Systems, 2003. 1
- 4 Gal A, Eckstein J, Stoumbos Z. Scheduling of data transcription in periodically connected databases[J]. Stochastic Analysis & Applications, 2003. 9
- 5 Levene M, Loizou G. Why is the snowflake schema a good data warehouse design [J]. Information Systems, 2003. 5
- 6 Lechtenbörger J, Vossen G. Multidimensional normal forms for data warehouse design[J]. Information Systems, 2003. 7