

基于面向对象原型法的N层数据仓库设计

王秀娟, 曹宝香

(曲阜师范大学 计算机科学学院, 山东 日照 276826)

摘要:针对建立数据仓库时存在的系统需求不断变化、语义断层以及数据仓库改动难、代价大等问题,文中提出一种N层数据仓库体系结构模型,以及基于原型法的数据仓库建立方法。文中给出了运用面向对象原型法建立N层数据仓库的关键技术,并阐述了利用原型法进行N层数据仓库开发的优点。该设计思想在PLM产品生命周期管理系统中使用并取得了较好的决策支持效果。

关键词:N层数据仓库;原型法;数据集市;多维模型

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2009)01-0117-04

Design of N-Layer Data Warehouse Based on Object Oriented Prototyping

WANG Xiu-juan, CAO Bao-xiang

(Computer Science College, Qufu Normal University, Rizhao 276826, China)

Abstract: Due to the problems of constantly changing requirements, semantic faults and high expenses met as setting up or changing the data warehouse, puts forward an N-layer data warehouse architecture model, and one method for developing data warehouse based on the thought of object-oriented prototyping. The key technologies for establishing an N-layer data warehouse with prototyping method are given and the advantages of this method are also presented in this paper. This design has been used in PLM product life cycle management system and made a better result of decision support.

Key words: N-layer data warehouse; prototyping; data mart; multidimensional model

0 引言

数据仓库是支持决策过程的、面向主题的、集成的、随时间而变化的、持久的数据集合,它是利用大量高质量的数据源对企业提供决策支持的一种数据库技术^[1]。随着信息技术的发展,数据仓库系统正在成为企业信息资产的重要应用工具和获取信息领域投资回报的重要途径。

据 IDC 调查,世界企业 500 强中已有 85% 的企业建成或正在建设数据仓库,但是却有 70% 的数据仓库是失败的。主要表现在建立数据仓库时难以完全符合需求,修改数据仓库时难度高、代价大。

究其原因,一方面是由于构建方法的不适当。例如,传统的面向过程瀑布方法,长时间应用于传统的操

作环境,而它在以决策支持为主要目的的数据仓库环境中有很多局限性:

(1)数据仓库中的决策支持系统在实际应用中希望是动态的、不断改进的,而瀑布方法却在一开始就将系统定义为静态集合^[2]。

(2)数据仓库作为企业级的决策服务,不是包容所有的数据和所有企业用户服务的,而瀑布方法却假定用户有一致的系统要求。

(3)开发时间过长。传统的方法往往需要较长的开发周期,但数据仓库开发的核心准则是尽可能地将质量信息快速、准确地传递给管理者来支持决策。

另一方面失败的原因是,数据仓库的体系结构不科学、不易改动。已建立的数据仓库为满足用户提出的新业务需求,往往需要对数据仓库做较大的调整,这些调整可能直接影响了现有数据仓库体系结构,甚至需要对原有的体系结构全面改动重建,势必会造成巨大的人力物力的浪费。

针对以上两个方面原因,提出采用面向对象的原型法作为开发数据仓库的方法,克服建立数据仓库过

收稿日期:2008-04-21

基金项目:山东省自然科学基金资助项目(Y2003G01);山东省科技计划项目(2006GG2301001)

作者简介:王秀娟(1983-),女,山东泰安人,硕士研究生,研究方向为中间件、网络数据库、面向服务的架构;曹宝香,教授,研究方向为中间件、网络数据库、CAD、数据库和系统集成。

程中由于需求变化和语义断层带来的问题。

此外,文中又根据数据仓库的应用、构架、模型以及项目的实施方法等多个方面在国内存在与国外的差异,结合我国数据仓库项目的实际情况,对原有数据仓库体系结构进行调整和完善,提出了一种 N 层数据仓库的体系结构。

1 原型法及其特点

原型是指具备了最终系统的部分重要功能的一个早期可运行的雏形。原型法是一种能在获得用户需求后,快速实现一个系统原型,并在运行和调试的过程中,逐步调整需求,确定需求细节的方法。利用基于原型法的方法建立数据仓库具有以下两个特点:

(1)适应数据仓库决策支持系统的动态性和企业用户需求的多重性。因为,原型法是一种以模型为驱动的开发方法,它不要求应用系统的需求定义必须在系统设计之初就完全制定出来。整个系统的需求定义贯穿于整个开发过程。通过用户和开发人员的多次交流,共同评价,逐步完善和修改原型,最终演化成目标系统,因此容易满足数据仓库决策支持系统的动态性和企业用户需求的多重性。

(2)系统开发时间短、成本低,满足了管理者以快速准确的信息来支持决策的实时要求。图 1 是原型法在系统构建中应用的略图。

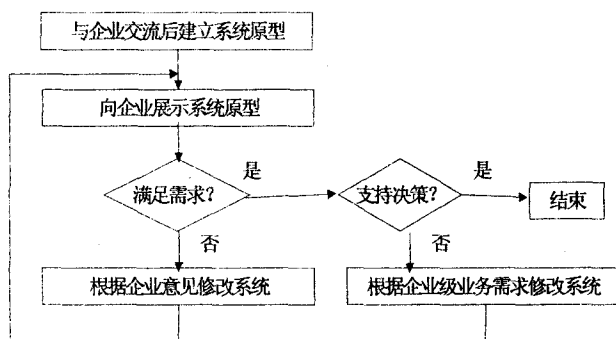


图 1 系统中应用原型法

2 N 层数据仓库体系结构

传统的数据仓库一般逻辑缺乏层次性,因此变动时难度高、代价也很大。如果在数据仓库逻辑结构中构建多个层次,将变化隔离在体系结构中各个相对独立的层次内,那么数据仓库系统构架就会变得更加稳

定并具有更高的扩展能力。这一思想主要是在数据仓库结构中建立多个独立层次,即形成了 N 层数据仓库体系,如图 2 所示。

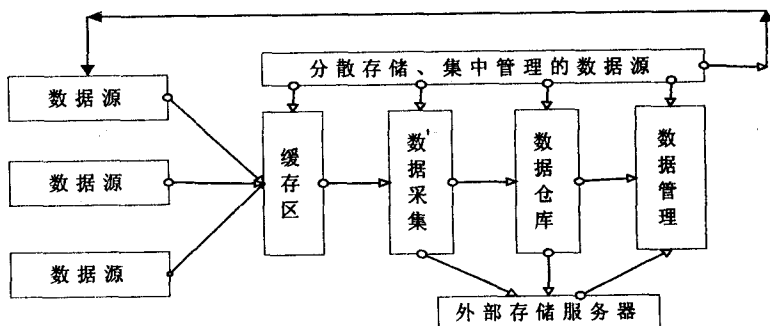


图 2 N 层数据仓库模型

如此设计可以满足数据仓库项目中手工补录的数据量大、数据更新频率快、强调数据实时性等特点。

下面以三层七表的数据仓库模型为例,介绍一下 N 层数据仓库的结构,如图 3 所示。

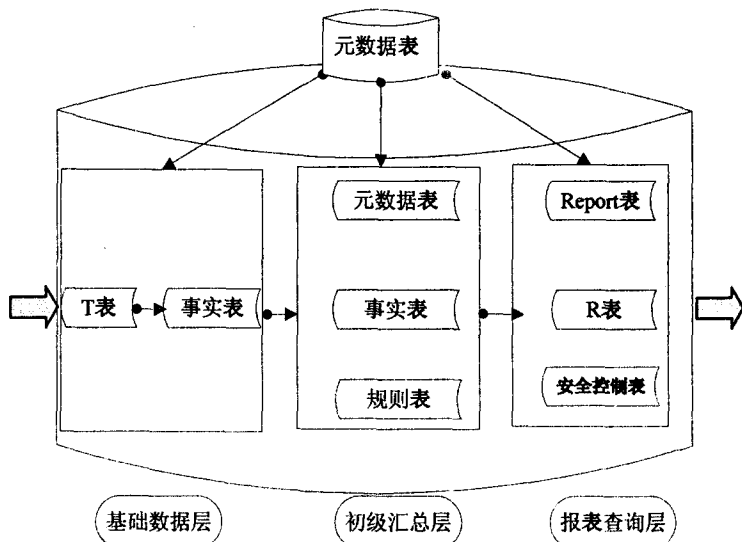


图 3 三层七表图

1)基础数据层:用来存放最低粒度数据的数据表集合,例如“交易记录”这样的基础数据。它是其他两层的数据来源,通常提到的逻辑数据模型就在该层内。

2)初级汇总层:将基础数据层中的数据按照某些业务规则进行初步汇总,以供前端查询和报表使用。

3)报表查询层:是为了应对特殊的业务需求而设计的数据表 WCUP。

4)T表:数据仓库中的临时表,存放来自缓冲区的数据,然后通过存储过程进行整合,并将结果装载到事实表中。

5)业务元数据表:记录数据仓库模型中每一个字段对应的中文业务名称。

6)安全控制表:即用户权限表,记录哪些用户可以访问数据仓库,每个用户可以查询的范围。

7)Report表:用来存放具有实际管理意义的数据报表。

8)规则表:记录某些特殊的数据运算规则,如:资产=现金+银行存款

9)事实表:是数据仓库中数量最多的表。它有两种形式:一是存在于基础数据层的事实表A,它以第三范式或星型模式的方式存储数据,逻辑数据模型的大部分数据就存储在事实表A中(后文将以实例进一步体现事实表的作用);另一种是存放于初级汇总层中的事实表B,其作用在于将基础数据层和事实表A中的数据按照某些规则进行初步的汇总后,供给前端查询和报表使用。

10)R表:有两种存在形式:一是为了便于向数据集市装载数据而建立的临时表,如为提高装载速度而建立的星型数据结构;另一种则是为了解决工具的某些技术缺陷而设计的查询加速手段。

3 面向对象原型法在N层数据仓库模型设计中的应用

原型法时间短、成本低、交互性强的特点在对项目风险的降低程度上是被广大开发者认同的。

将原型法应用于N层数据仓库模型的优点在于:

(1)N层数据仓库可以将数据的变化隔离在体系结构中各个相对独立的层次内,减弱了原型法在对系统开发过程中要依据需求不断完善原型的整体结构而造成的改动所带来的副作用。

(2)面向对象技术是一种运用类、继承、封装、聚合、消息、传递、多态性等概念来构造系统的方法^[3],可以在概念上来考虑数据仓库的结构和动态属性。而原型法恰与面向对象技术有着共同的特点,即动态定义系统的需求,易于将数据仓库的约束和用户的需求嵌入到概念模型中去。

下面举例说明运用面向对象原型法设计N层数据仓库模型的方法。

3.1 初始原型构造

在设计数据仓库时,首先要建立需求模式,需求模式是用户与系统交互活动的表达,需求模式的全体定义了系统的全部的需求^[4]。根据一系列系统需求模式,可以进行对象的定义,将对象抽象成类,对类进行细分,形成类集,画出类的树形继承图,组装成具体的

表单对象,再加上相应的实体对象及控制对象,构成系统原型。为了保证数据的集成性,在确定需求时,需求范围要适当扩大,在以后的系统分析中,可以统筹兼顾,为系统的扩展和数据的集成做好准备。在范围的选取上应先选择少数主体为主,采用星状结构构建相应的数据库,组成一个规模较小的数据仓库,这样不仅能够降低成本,也可以缩短开发时间。

3.2 构建多维数据库模型

多维模型^[5]是模型建设方法中非常适合高性能获取数据的模型,它由事实表和多维表组成。事实表记录了用来分析的数据(通过具有数字属性的度量值来描述),而维表则记录了用来分析数据的角度,每个维又可以按某种顺序划分为多个层次,每个维层次上取值的组合称作该维的维成员,如时间可以划分为年、月、日不同的层次,而“某年”,“某年某月”,“某年某月某日”都是维成员。有时同一维也可以具有多个层次结构,如时间维有两个不同的层次路径:时间一日一月一年和时间一季度。用数据立方体来描述的多维模型如图4所示。

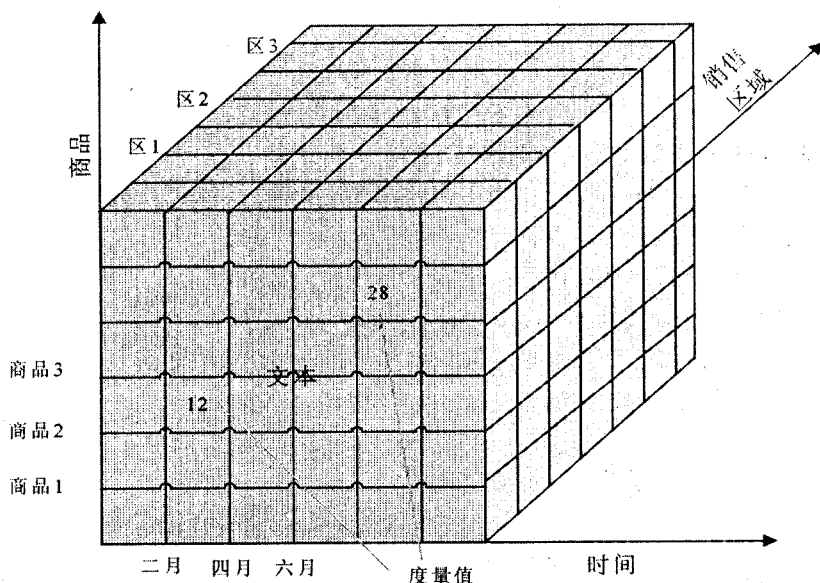


图4 多维模型

由于在维之间事实是多对多的关系,事实和每个具体的维是多对一的关系,采用面向对象技术构建多维数据模型实例如图5所示。

使用多维模型管理数据仓库的数据优点在于:第一,对业务用户而言,模型简单易用。由于事实表和维表之间的关联关系已建立,所以对于理解数据的业务意义和汇总事实表中的数据都是非常容易的。第二,数据分析非常简单。用户查询的结果会按预先设定的路径细化,而不必为细化的过程额外开发程序。第三,易于扩展。星型模型^[6]易于向维表中增加维值,而不

必打乱现有数据仓库的操作,而增加新的维表时,只需修改事实表中的相关数据。所以多维模型数据仓库的易于管理性恰好满足了原型法开发过程中需经常对系统进行改动的需求。

为这样会造成各部门之间的数据冗余,导致了数据集成困难性。为此,用原型法开发数据仓库时,为了降低项目的风险,提高数据的集成性,首先建立一个或多个数据集市,然后再集成为企业数据仓库。在集成数据集市的过程中,采用度量维度化和维度共享化^[8]的方法。

所谓度量维度化就是在同一维数据集中通过考察数据度量属性之间共有的特性,简化度量的数量,用可以描述其共性的维来替代;不同的集市之间也可以通过比较彼此之间度量属性的相似性,来构造相同的维以简化度量的数量,近而形成公共维,便于数据集市的集成。所谓维度共享就是在建立不同的数据集市时,尽量设计单位相同或满足一定转化关系的维度。如在两个不同的数据

集市中都有时间维,可以将它们的单位都设为天,或者一个是天,另一个是星期,因为天可以合成星期,故这两种时间维度就是一致的了。

度量维度化和维度共享化都使数据模型得到大大简化,将度量和维度的集中管理从数据立方体提升到了整个数据仓库的层面,从而实现更灵活、更智能的 OLAP。

3.4 采用递增集成的方法

由于面向对象原型法的设计思路是:首先根据用户需求,得到核心对象的定义和带有界面描述的需求模式的定义,并建立相交的对象和类图,得到反映系统主要功能的原型模型,然后执行原型,采用增量开发策略,每个增量单位为一个需求模式,通过与用户的交流逐步改进和形式化分析现存模型,最终完成系统的设计。如此应用于 N 层数据仓库的设计中,可以仿照这种方法先开发一个较小的主要的模型并逐步运行和调试,然后每开发一个模块便将其与前一模块集成,直到系统完成。这样做的优点在于,每个模块都经历了运行和测试,因此能及时地发现错误并修正错误。逐步集成要比将所有模块开发完后一起集成更具有鲁棒性^[9]。

(下转第 124 页)

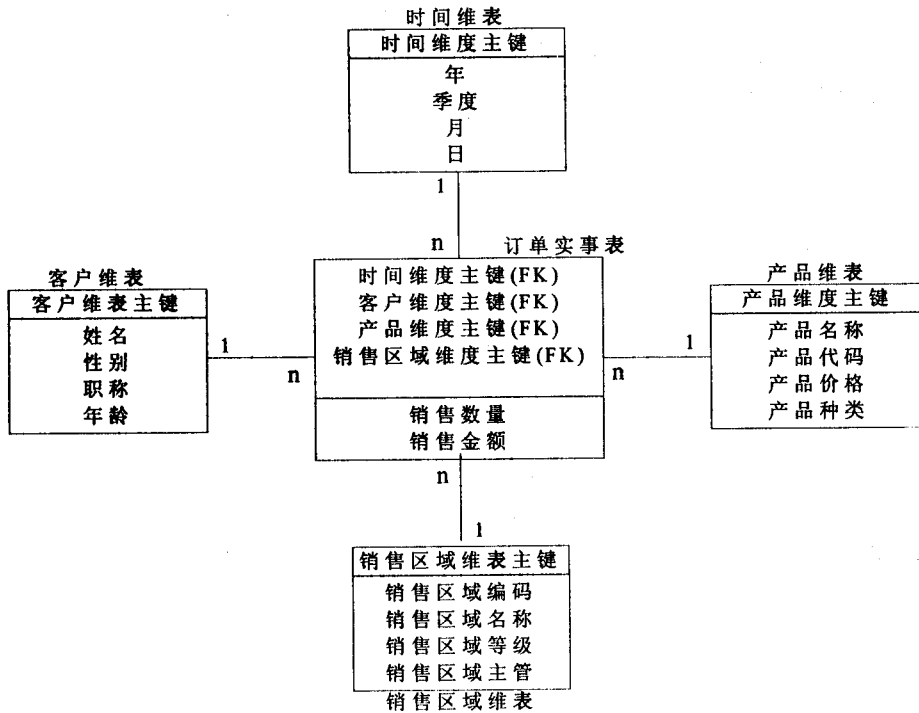


图 5 面向对象的多维模型

3.3 使用多维数据库实施的数据集市

在 N 层数据仓库体系结构下,数据集市的本质是将关系到同一类业务问题的数据在逻辑上存放在一起的^[7]。鉴于多维数据库技术具有“查询速度快,数据观察角度灵活”(如旋转 pivot,切片 slice,切块 dice,上卷 roll-up,下钻 drill-down)的特殊技术优势,文中使用多维数据库实施的数据集市如图 6 所示。

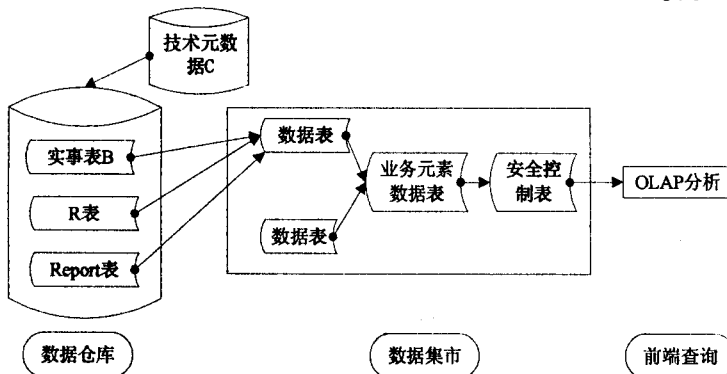


图 6 多维数据仓库中的数据集市

当用户提出的需求非常紧迫,或者缺乏支持企业范围决策支持策略系统的人员,亦或是企业的商业单元过于分散时,可以借助数据集市投资少、建设周期短、简易并且廉价的特点来部分充当数据仓库的企业职能,但是又不能用数据集市完全地替代数据仓库,因

成,每个 Agent 都有自己的职责,并与其它 Agent 通信获取信息,互相协作完成整个问题求解。与单个 Agent 相比,多 Agent 系统能够完成更为复杂更广泛的功能。大家不要产生误解,即将面向对象和面向主体对立起来,认为是两种截然不同的方法,其实它们之间的关系是包含关系,即面向主体包含面向对象,面向对象设计方法的一切优点都可以被面向主体所吸收和容纳。

从最初的 SmallTalk 到纯面向对象的 Java,面向对象的程序设计语言现在已经相当成熟;但到目前为止,还没有任何一种面向主体设计语言,就目前的技术结构而言,面向主体的设计还需要靠面向对象程序设计语言来具体实现。

(上接第 116 页)

- Oriented Programming[C]//Proceedings of NetObjectDays Conference. [s. l.]:Springer - Verlag,2002:233 - 247.
- [4] Glatard T, Montagnat J, Pennec X. Grid - enabled workflows for data intensive medical applications[C]//In: Proceedings of the 18th IEEE Symposium on Computer - Based Medical Systems. Washington, DC, USA: IEEE Computer Society, 2005: 537 - 542.
- [5] Chavez C, Lucena C. A Metamodel for Aspect - Oriented Modeling[C]//In: The First Conf. on Aspect - Oriented Software Development (AOSD 2002). The Netherlands: [s. n.], 2002.
- [6] Lions J M, Simoneau D, Pitette G, et al. Extending OpenTool/UML Using Metamodeling: An Aspect Oriented Programming Case Study[C]//In: 2nd Int'l Workshop on Aspect - Oriented Modeling with UML (AOSD). Boston: [s. n.], 2002.

(上接第 120 页)

4 结束语

基于数据仓库技术在我国市场环境、业务范围等方面与国外存在的差异,文中在对数据仓库技术深入研究和实践的基础上,提出了 N 层数据仓库的设计思想,结合面向对象原型法的开发思想,采用面向对象技术,能快速地建设 N 层数据仓库的初始模型,并与用户一起对模型评价、修正,完成目标系统的设计。

该设计思想已在 PLM 产品生命周期管理系统中使用,并取得了较好的决策支持效果。

参考文献:

- [1] Inmon W H. 数据仓库[M]. 北京:机械工业出版社,2000.
- [2] 连立贵,江平,蔡家楣.一种面向对象的螺旋式数据仓库开发方法[J]. 计算机工程,2001,27(12):52 - 53.

参考文献:

- [1] Bernhard B, Jörg P M, James O. Agent UML: a formalism for specifying multiAgent software systems [J]. International Journal of Software Engineering and Knowledge Engineering, 2001, 11(3): 207 - 230.
- [2] Deloach S A, Wood M F, Sparkman C H. MultiAgent systems engineering[J]. International Journal of Software Engineering and Knowledge Engineering, 2001, 11(3): 231 - 258.
- [3] Wooldridge M. 多 Agent 系统引论[M]. 石纯一, 张伟, 徐晋晖, 等译. 北京:电子工业出版社,2003.
- [4] 李金厚. 智能 Agent 自我的统一与分离[J]. 电子科技大学学报, 2001(1): 77 - 81.
- [5] Russell S, Norvig P. 人工智能——一种现代方法[M]. 北京:人民邮电出版社,2002.
- [7] Han Y, Kriesel G, Cremers A B. A Meta Model for Aspect [EB/OL]. 2004 - 03. In: <http://www.informatik.uni-bonn.de/gk/papers/IAI-TR-2004-3>.
- [8] Marcos D Del F, Jean B, Frédéric J, et al. AMW: A Generic Model Weaver[C]//In: IDM'05, Premières Journées sur l'Ingénierie Dirigée par les Modèles. Paris: [s. n.], 2005.
- [9] Suzuki J, Yamamoto Y. Extending UML with Aspects: Aspect Support in the Design Phase[C]//In: AOP Workshop at E-COOP'99. Portugal: [s. n.], 1999.
- [10] Katara M, Katz S. A concern architecture view for aspect-oriented softdesign[M]. [s. l.]:Springer - Verlag, 2006.
- [11] Apel S, Kastner C, Kuhlemann M, et al. Pointcut, advice, refinements, and collaborations: similarities, differences, and synergies[M]. London: Springer - Verlag, 2007.

- [3] 蔡勇, 贺广生. 面向对象的数据仓库模型设计[J]. 计算机工程与应用, 2003(6): 193 - 195.
- [4] 吴彦春, 饶文碧, 罗小琴, 等. 面向对象原型法在 MIS 开发中的应用研究[J]. 微机发展(现更名为: 计算机技术与发展), 2000, 10(4): 73 - 75.
- [5] 于宗民, 刘义宁, 祁国辉. 数据仓库项目管理实践[M]. 北京:人民邮电出版社, 2006.
- [6] 张秀国, 徐晓红. 基于 UML 的数据仓库构造方法的研究[J]. 计算机工程与设计, 2002, 23(11): 21 - 24.
- [7] 常新功, 金红, 肖玉巍. 正确使用原型法构建数据仓库[J]. 微机发展(现更名为: 计算机技术与发展), 2002, 12(4): 55 - 56.
- [8] 付燕燕, 蒋代梅, 周小兵. 支持增量式数据仓库建设的多维数据模型[J]. 北京工业大学学报, 2005, 31(4): 399 - 404.
- [9] 王珊. 数据仓库技术和联机分析处理[M]. 北京:科学出版社, 1998.