数据仓库系统中逻辑建模的方法研究①

Research on the Method of Building Logic Model in Data Warehouse

谷 岩　郭 庆　(广州大学信息学院 510091)

摘要:数据仓库是按照主题来建模的,由于数据仓库系统的原始需求不明确,无法确定系统的功能,因此不能采用功能驱动的开发方法进行数据仓库系统的开发,而只能采用数据驱动的方式。数据仓库的逻辑建模是系统开发过程中最重要的一步,是系统实现的基础和成败的关键。本文给出了数据仓库逻辑建模的一种解决方案。以此为基础,结合已有的业务系统,可以逐步构建出完整的、健壮的数据仓库系统。

关键词:数据仓库 主题域 概念模型 逻辑模型

1 引言

面向 OLTP 的业务处理系统是按照应用需求建立它的模型,而且系统开发过程的每个阶段都有明确的业务处理功能,因此系统的开发是基于功能驱动的。而数据仓库是面向主题的、集成的、不可更新的、随时间而变化的,它是按照主题来建模的,由于数据仓库系统的原始需求不明确,无法确定系统的功能,因此不能再采用功能驱动的开发方法进行数据仓库系统的开发。数据仓库系统的开发是在现有数据库系统的基础上进行的,它着眼于有效地抽取、综合、集成和挖掘已有数据库的资源,服务于企业决策者的需要,因此数据仓库的设计是数据驱动的,需要有一套特殊的设计方法进行逻辑建模。逻辑建模过程包括:

(1) 概念模型的设计

(2) 逻辑模型的设计

2 数据仓库概念模型的设计

2.1 概念模型的定义

在数据仓库开发之前可以通过系统的需求分析,了解系统的大致数据需求,界定系统的边界,确定系统的工程范围。基本方法是对原有业务系统的各数据库子系统及其相互之间的联系进行分析,提炼出系统有用的数据。同时要明确所需要构建的内容,确定所选择的主题域。由于无法确定用户明确而又详细的需求,因此可以尽可能地了解用户的一些基本需求方向和数据需求:用户需要做哪些决策? 用户感兴趣的是什么问题? 解决这些问题需要什么样的信息? 例如对一个从事商品销售的商业企业来说,必须通过建立数据仓库来准确掌握商品的销售情况和库存情况,辅助企业进行营销策略制定的决策,因此通过分析可以确定系统的边界,即定义系统的概念模型,该商业企业数据仓库的概念模型见表 1。

表 1 商业企业数据仓库的概念模型定义

目的	辅助企业进行营销策略制定的决策
用户的决策分析	1. 客户的购买趋势
	2. 商品供应市场的变化趋势
	3. 供应商和客户的信用等级
	4. 商品的销售变化趋势
数据的需求分析	1. 商品销售量
	2. 商品的销售利润率
	3. 商品采购量
	4. 商品库存量
	5. 各部门的销售业绩
	6. 一般客户和重点客户情况
	7. 供应商情况

2.2 概念模型的分析

概念模型的分析实际上就是进行系统的需求分析,常见的分析方法包括用户驱动分析方法和数据驱动分析方法。用户驱动分析方法主要从用户需要的角

① 基金项目:广东省科技攻关项目(2004B10101044)、 广州市科技攻关项目(2004Z3 - D0391)

度进行分析,生成的数据仓库系统虽然能使用户满意,但容易产生与源数据库的不一致,因此较适合数据集市的建立;而数据驱动分析方法主要从现有数据库出发,实现起来比较容易,同时较适合全局范围内数据仓库的建立。

概念模型的分析主要进行数据仓库范围内的主要对象的分析,以及系统的主要主题域及主要主题域之间联系的确定。概念模型一般用 E－R 图来表示,图中反映了系统由哪些对象组成以及各对象之间存在着怎样的联系。

根据表 1 所反映的用户决策分析的需求,可以确定系统存在着四个基本主题:销售主题、商品主题、客户主题和供应商主题。而这四个主题之间存在相互关系,其中供应商主题和商品主题之间存在着供货关系,商品主题和销售主题之间存在着供应关系,客户主题和销售主题之间存在着购买关系,由这四个主题及四个主题之间的关系形成了数据仓库的概念模型。

2.3 概念模型的设计

上节所给出的概念模型仅适合于传统数据库的设计,它不能完全用于数据仓库的设计,因此需要在信息包图等设计工具的基础上,完成星形模型或第三范式模型的设计。

(1) 信息包图。为了对数据进行完整、规范的分析,可以采用信息包图来描述用户的信息需求状况。信息包图将完成下列工作:

① 定义系统中的主题范围;

② 跟踪系统活动完成和运行的关键指标;

③ 决定数据如何传递给数据仓库的用户;

④ 建立数据层次;

⑤ 估计数据仓库大小;

⑥ 确定数据仓库中数据更新的频率。系统的信息包图见表 2。

表 2　系统的信息包图

维度　粒度							
所有时间周期	所有地区	所有商品	所有销售部门	所有客户	所有供应商	回收款	利润率
年	区域	商品种类	部门	客户	供应商	回收款	利润率
季	省	商品小类	组				
月	市	商品	个人				
周	县						
日							
指标/实际情况 销售预测、销售预算、实际销售、预测偏差、预算偏差							

(2) 星型模型的设计。星型模型是一种多维的数据关系,它由一个事实表和一组维表组成。每个维表都有一个维作为主键,所有这些维组合成事实表的主键。事实表的非主属性称为事实,它们一般都是数值或其它可进行计算的数据,而维一般情况下都是文字、时间等类型的数据。

从系统的信息包图可以确定上述系统的主题是商品销售的趋势分析,而指标实体是销售趋势,因此可以用星形模型表示维数实体和指标实体之间的关系,指标实体应位于星形模型的中心。从系统的信息包图还可以发现,用户在销售趋势分析中所需要的信息包括有销售时间、销售地区、销售部门、客户、供应商、回收款、利润率,这些信息构成了星型模型的维实体。因此,以销售趋势为主题的星形模型见图 1。

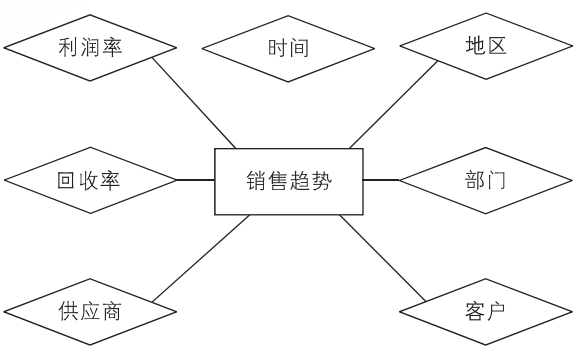

图 1　以销售趋势为主题的星形模型

指标实体与现实世界的事务或事件有关,与每个相关维度的一个点对应。指标实体有 5 个特性:

① 提供定量数据、商务指标数据或实际数据;

② 包括多个访问指标数据的路径、维数或指针;

③ 包括相关的标准数据;

④ 构成每个维数内最低一级的类别和一个信息包图中的指标;

⑤ 能够扩展成很大的表格。

指标实体是分析活动的核心。围绕指标实体的是维度实体,数据仓库的用户用维度实体在指标实体中组织和过滤数据。维度实体有 6 个特性:

· 表示维度体系;

· 访问和过滤指标实体;

· 包含一个完整的维度体系的编码、关键字以及它们的相关表示;

· 映射到信息包图的列;

· 实现时能得到较小的表;

· 提供访问数据仓库的网关。

采用星型模型可以提高系统的查询效率,因为在模型设计时对各个维作了大量的预处理。例如按照维可以进行预先的统计、分类、排序等。因此对于基于星型模型的数据仓库,生成报表的速度会很快。但由于存在大量的预处理,其建模过程相对来说就会慢。同时当业务问题发生变化,原来的维不能满足要求时,就需要增加新的维,而由于事实表的主键由所有维表的主键组成,因此这种维的变动将是非常复杂、耗时。星型模型的另一个缺点是数据的冗余量很大。因此我们不难看出星型模型比较适合于预先定义好的问题,如用户需要产生大量的查询报表,而不适合于动态查询多、系统可扩展能力要求高或者数据量很大的用户需求。因此,星型模型可以应用在一些要求大量报表的部门数据集市中。

(3) 第三范式的建模方法。在数据仓库的模型设计中也可采用第三范式的数据模型,它可以实现数据访问的灵活性和高效的数据存储。采用第三范式的数据模型有非常严格的数学定义,它要求数据仓库中的基本数据必须进行第三范式的规范化处理。范式是数据库逻辑模型设计的基本理论,一个关系模型可以从第一范式到第五范式进行无损分解,这个过程称为规范化(Normalize)。一个符合第三范式的关系必须满足以下三个条件:

① 所有非主属性必须完全依赖整个主键,而非主键的一部分;

② 所有的主要属性都完全依赖于不属于它们的键;

③ 没有属性完全依赖于任一非主属性集。

因此,在将数据模型从非规范到第三范式的转换过程中,需要采取以下三个步骤:

· 第一步:消除所有的重复元组,实现第一范式;

· 第二步:将实体的所有非主属性依赖于所有的主键列;

· 第三步:将所有非主键列直接依赖于主键列。

(4) 数据仓库的反规范处理。采用第三范式对数据模型进行规范化处理后,发现这些规范化处理在数据仓库的实现过程中存在许多问题。由于数据库引擎的限制,如果完全按照第三范式创建表,在进行常规的数据操作时会严重影响系统的查询效率。例如当系统的数据量很小时,如果只有几个 GB,那么进行多表连接之类复杂查询的响应时间是可以忍受的,但如果数据量扩展到几百 GB 甚至到 TB,一个表中的记录往往有几百万、几千万甚至更多,这时进行多表连接这样的复杂查询,响应时间是不可忍受的,这时就有必要把几个表合并,以尽量减少表的连接操作。因此在实际应用过程中不得不对逻辑模型进行不规范处理(De – Normalize),以提高系统的响应速度,这当然是以增加系统的复杂度、维护工作量、磁盘使用比率(指原始数据与磁盘大小的比率)并降低系统执行动态查询能力为代价的。下面将列出一些常见的反规范处理方法:

① 多表连接:在设计模型时对表进行合并,即所谓的预连接(Pre – Join)。当数据规模小时,也可以采用星型模型,这样能提高系统速度,但增加了数据冗余量。

② 表的累计:在模型中增加有关小计数据(Summarized Data)的项。当然这也增加了数据冗余,而且如果某项问题不在预建的累计项内,需临时调整。

③ 数据排序:对数据事先排序。但随着数据仓库系统的运行,不断有新的数据加入,数据库管理员的工作将大大增加。大量的时间将用于对系统的整理,系统的可用性随之降低。

④ 大表扫描:通过使用大量的索引,可以避免对大量数据进行扫描。但这也将增加系统的复杂程度,降低系统进行动态查询的能力。

当然,不规范处理的程度取决于数据库引擎的并行处理能力。我们在选择数据库引擎时,除了参考一些相关的基准测试结果外,最好是能根据自己的实际情况设计测试方案,从几个数据库系统中选择最适合自己企业决策要求的一种。

由于中央数据仓库的数据模型反映了整个企业的业务运行规律,在中央数据仓库进行反规范处理容易影响整个系统,不利于系统今后的扩展,而且反规范处理产生的数据冗余将使整个系统的数据量迅速增加,从而增加 DBA 的工作量和系统投资。因此,当系统性能下降而需要进行反规范处理时,比较好的办法是选择问题较集中的部门数据集市实施这种措施。这样既能有效地改善系统性能,又不至于影响整个系统,在一些成功的大型企业级数据仓库案例中,基本上都是采用这种方法。

实际应用中的数据仓库将面临两种用户负载:一种是重复性的查询问题,另一种是交互性的查询问题。用户的动态查询具有明显的交互性特征,因为这种查询是在一个问题答案的基础上进行进一步的探索,这种交互过程就是数据挖掘(Data Mining)过程。部门数据集市主要面对第一种负载,适合用星型模型进行设计,因为数据集市主要面临数据量不大、报表较固定的查询问题;而对于中央数据仓库,考虑到系统的可扩展能力、投资成本和易于管理等多种因素,最好采用第三范式的建模方法。

3　数据仓库的逻辑模型的设计

数据仓库的逻辑模型设计是系统实现的基础和成败的关键。数据仓库的逻辑模型设计主要包括主题的确定、分析内容的细化以及粒度的设计。

(1) 主题的确定。主题是一个逻辑概念,它应该能够完整、统一地描述出分析对象所涉及的各项数据以及相互联系。划分主题的根据主要来源于两方面:对原有固定报表的分析和对业务人员的访谈。原有固定报表能较好地反映出以往工作对数据分析的需求,而且其数据含义和格式相对成熟、稳定,在模型设计中需要大量借鉴。但仅仅满足于替代目前的手工报表还远远不应是构建数据仓库系统的目标,还应该通过业务访谈,进一步挖掘出日常工作中潜在的更广、更深的分析需求。只有这样,才能真正了解构建数据仓库模型所需的主题划分。

在上述的概念模型中已经确定了多个基本的主题域,如销售主题、商品主题、客户主题和供应商主题。在逻辑模型的设计阶段,就要通过对多个基本主题域的分析,按轻重缓急确定首先需要建立的主题域。由于销售主题是商业企业的最基本的核心业务,其它主题域都是紧紧围绕该主题域而展开的,通过销售主题域的建立,可以全面了解企业的经营状况,并为制定长远的营销策略起到辅助决策的作用,因此可以将销售主题域确定为首先要实现的主题。确定了主题域后,还需要定义能充分反映主题域的属性组,这些属性表示了每个主题的维度。系统中各主题域的详细描述见表 3。

表 3　主题域的详细描述

主题域名	主 键	属 性
销售	销售单号	销售单信息:销售单号、销售地址 销售信息:客户号、商品号、销售数量、销售单价、销售时间
商品	商品号	商品信息:商品号、商品名、规格 商品采购信息:商品号、供应商号、供应单价、供应数量 商品库存信息:商品号、库房号、库存量、日期
客户	客户号	客户信息:客户号、客户名、客户类型、住址、电话、帐号
供应商	供应商号	供应商信息:供应商号、供应商名、供应商类型、电话、帐号

(2) 分析内容的细化。主题的划分实际上是与分析内容的范围直接相关的,一旦主题划分清楚了,下一步就是细化分析的具体内容以及根据分析内容的性质确定它在数据仓库中的位置。通常,维元素对应的是分析角度,而度量对应的是分析所关心的具体指标。一个指标究竟是维元素、度量还是维属性,取决于其具体的业务需求,但从实际操作中可以总结出这样的概念性经验:作为维元素或维属性的通常是离散型的数据,只允许有限的取值;作为度量的是连续型数据,取值无限。在细化分析内容的过程中,务必解决指标的歧义问题。在不同报表中以及在业务访谈中同一名称的指标,是否是在同样条件限定下,通过同样方法提取或计算得到的,它们之间的相互关系是什么,这些问题都必须从熟悉业务的分析人员那里得到准确、清晰的答案,否则将会影响到模型设计、数据提取、数据展现等多个方面。

(3) 粒度的设计。粒度是指数据仓库中数据单元的详细程度和级别。数据越详细,粒度就越小,级别也

就越低;数据综合度越高,粒度就越大,级别也就越高。数据仓库模型中所存储的数据的粒度,将对系统产生多方面影响。事实表中以维度的各种层次作为最细粒度,将决定存储的数据能否满足信息分析的功能需求,而粒度的层次划分以及聚合表中粒度的选择,将直接影响查询的响应时间。为了既能提高访问和分析的效率,又能提供非常详细有力的数据分析能力,可以对数据进行多重粒度划分,即把数据按照综合程度高低划分成当前细节级、轻度综合级、高度综合级和早期细节级。

在数据仓库的逻辑模型中,高粒度的数据是在低粒度数据的基础上按一定的统计模型分析后得出的,这些数据有两种逻辑表示方法:第一种方法是在基础数据表上建立高复杂度的逻辑视图;第二种方法是把分析结果存入新数据表中。第一种方法能够保证数据的一致性,且可以及时地反映底层数据的变化,但缺点是响应速度慢,同时低粒度的底层数据在一定年限后要进入早期细节级,一般情况下要转移到备份表中,那么这些数据上通过视图表征的逻辑数据就会丢失。而第二种方法是把分析后的结果数据存入新表,虽然这些数据是蕴含在底层数据中,但这些数据是通过一定的分析计算得到的,因此具有一定的独立性,并且也不完全依赖于底层数据,也不违反关系数据库的一致性要求。将这些综合数据保存起来后,可以大大提高系统的查询效率,同时也解决了粒度的相关性问题。

4 结束语

数据仓库建设是一个系统工程,是一个不断建立、发展、完善的过程,只有稳固的数据仓库基础设施才能支撑灵活多样的数据仓库应用,数据仓库的逻辑建模又是系统开发过程中的最重要的一步,是系统实现的基础和成败的关键。本文给出了企业的数据仓库逻辑建模的一种解决方案。数据仓库系统将以此为基础,对整个系统的建设提出一个全面、清晰的远景规划及技术实施蓝图,并结合已有的业务系统,逐步构建出完整的、健壮的数据仓库系统。

参考文献

1 陈京民等,数据仓库与数据挖掘技术[M],北京电子工业出版社,2002。

2 王 珊等,数据仓库技术与联机分析处理[M],北京科学出版社,1998。

3 Jonathan G Geiger. (2000). The Data Warehouse Model [EB/OL], September 20, www. datawarehouse. com.

4 Chuck Ballard etc. (1998). Data Modeling Techniques for Data Warehouse[M], IBM.