CHAPTER 3

数据仓库系统的设计与开发　第3章

通过前两章的介绍，我们对数据仓库的概念、体系结构、存储结构和ETL过程等内容都有了一定的了解，那么如何建立一个数据仓库系统？在实际工作中数据仓库系统是如何设计和开发出来的？特别是在数据仓库创建以后，又怎样在此基础上建立多维数据模型？这些问题实际上属于数据仓库的实践范畴，与具体的应用系统环境特别是所使用的数据库环境、开发工具甚至开发人员都有密切的关系。本章以微软公司的SQL Server 2005为应用开发环境，通过实例介绍数据仓库的设计与开发过程。

3.1　数据仓库系统的设计与开发概述

3.1.1　建立数据仓库系统的步骤

数据仓库系统的建立在一定程度上说，是一个复杂甚至漫长的过程，因为数据仓库系统的开发涉及源数据系统、数据仓库对应的数据库系统及数据分析与报表工具等诸多应用问题。因此，数据仓库系统的创建不是一蹴而就的，将数据从原有的操作型业务环境移植到数据仓库环境本身就是一项复杂而艰巨的工作。一般来说，一个数据仓库系统的建立需要经过如下步骤。

（1）收集和分析业务需求。用户需求往往不确定，在数据仓库环境中，决策支持分析人员往往是企业或事业组织的中上层管理人员，他们对决策分析的需求不能预先作出规范说明。他们对开发人员说："让我看看能得到什么，然后我才能告诉你我真正需要什么"。因此，数据仓库应该在海量的数据中为用户提供有用、及时、全面的信息，以帮助用户作出正确的决策。

（2）建立数据模型和数据仓库的物理设计。通过设计数据仓库的概念模型、逻辑模型和物理模型，可以得到企业或事业数据的完整而清晰的描述信

息。数据仓库的数据模型通常是面向主题建立的,同时又为多个面向应用的数据源的集成提供统一的标准。数据仓库的核心内容包括组织的各个主题域、主题域之间的联系、描述主题的码和属性组等。

(3) 定义数据源。也叫做定义记录系统(system of records),往往会形成一个操作型数据存储区,数据仓库中的数据来源于多个已有的操作型业务系统。一方面,各个系统的数据都是面向应用的,不能完整地描述企业中的主题域;另一方面,多个数据源的数据之间存在着许多不一致,如命名、结构和单位不一致等,甚至数据的内容也可能不一致。所以必须在已有系统中定义记录系统。记录系统是一个内容正确并在多个数据源间起决定作用的操作型数据源。它的特点是:数据最完整、最准确、最及时,结构最适合于数据仓库,并且与外部数据源最为接近。

(4) 选择数据仓库技术和平台。技术和平台选型对建设数据仓库来说非常重要,而且一旦选定,在数据仓库系统实施完成后将很难改变,平台及技术的切换成本非常高,所以选型一定要充分重视和高度谨慎。

(5) 从操作型数据库中抽取、清洗及转换数据到数据仓库。本部分内容参见第 2 章。

(6) 选择访问和报表工具,选择数据库连接软件,选择数据分析和数据展示软件。根据用户的具体情况及其分析需求和数据量大小等因素选择。

(7) 更新数据仓库。确定数据仓库的更新策略,开发或配置数据仓库更新子系统,实现数据仓库数据的自动更新。

3.1.2 数据仓库系统的生命周期

数据仓库系统的开发与设计是一个动态的反馈和循环过程。一方面,数据仓库的数据内容、结构、粒度、分割以及其他物理设计根据用户所返回的信息需要不断地调整和完善,以提高系统的效率和性能。另一方面,通过不断地理解需求,使得最终用户能作出更准确、更有用的决策分析。

一个数据仓库系统包括两个主要部分:一是数据库,用于存储数据仓库的数据;二是数据分析应用系统,用于对数据仓库数据库中的数据进行分析。因此,数据仓库系统的设计也包括数据仓库数据库设计和数据仓库应用设计两个方面。事实上,系统的设计开发是基于数据仓库的规划、需求分析及数据模型建立等前期工作的,数据仓库系统在经过分析与设计两个重要阶段后,就会进入数据仓库系统的实施阶段,实施完成后便转入系统维护阶段。在系统的使用和维护过程中,用户会提出新的需求,同时也会有新技术出现,因此数据仓库系统在用户使用评价和新需求确认的基础上,进入新一轮的分析、设计开发、实施与维护的循环。这个开发与使用过程是一个不断循环、完善和提高的过程。在一般情况下,一个数据仓库系统不可能在一个循环过程中完成,而是经过多次循环开发,每次循环都会为系统增加新的功能,使数据仓库的应用得到新的提高。图 3.1 示意了这个循环的开发过程,这个过程也叫数据仓库系统的生命周期。

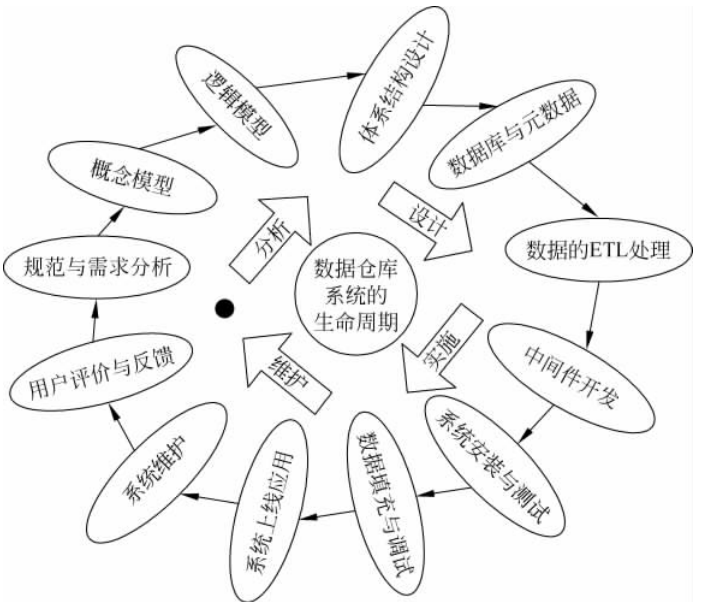

图 3.1　数据仓库系统的生命周期

3.1.3　建立数据仓库系统的思维模式

1. 自顶向下

自顶向下(top-down)模式首先把 OLTP 数据通过 ETL 汇集到数据仓库中,然后再把数据通过复制的方式推进各个数据集市中,其优点如下。

(1) 数据来源固定,可以确保数据的完整性。

(2) 数据格式与单位一致,可以确保跨越不同数据集市进行分析的正确性。

(3) 数据集市可以保证有共享的字段。因为都是从数据仓库中分离出来的。

2. 自底向上

自底向上(bottom-up)模式首先将 OLTP 数据通过 ETL 汇集到数据集市中,然后通过复制的方式提升到数据仓库中,其优点如下。

(1) 构建数据集市的工作相对简单,易成功。

(2) 这种模式可实现快速数据传送。

3.1.4　数据仓库数据库的设计步骤

数据仓库数据库的设计如图 3.2 所示,主要工作包括收集、分析和确认业务分析需求,分析和理解主题和元数据、事实及其量度、粒度和维度的选择与设计,数据仓库的物理存储方式的设计等。

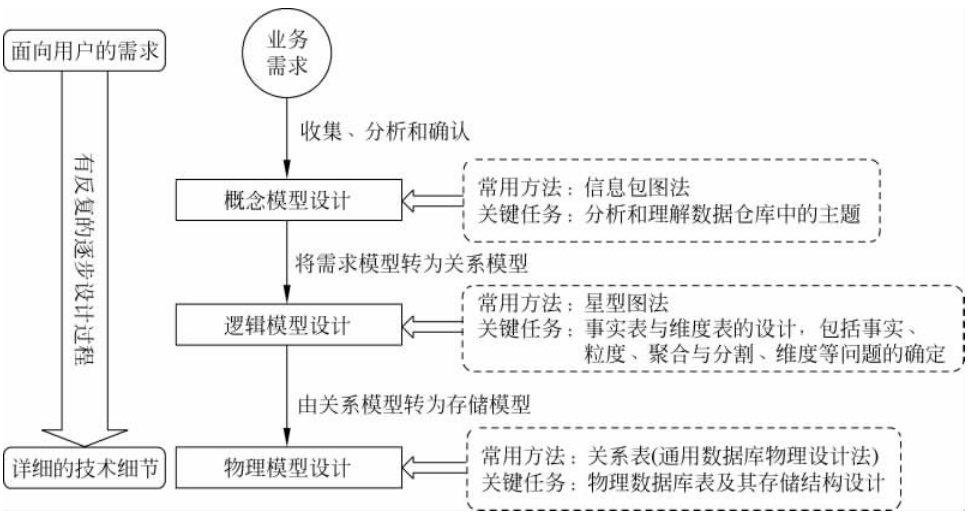

图 3.2　数据仓库数据库设计示意图

3.2　基于 SQL Server 2005 的数据仓库数据库设计

SQL Server 2005 集成了三个服务来实现数据仓库系统的开发：SQL Server 2005 Analysis Services、SQL Server 2005 Integration Services 和 SQL Server 2005 Reporting Services，同时还提供了一个数据仓库与商业智能应用系统的开发环境——SQL Server Business Intelligence Development Studio。它们的关系如图 3.3 所示。

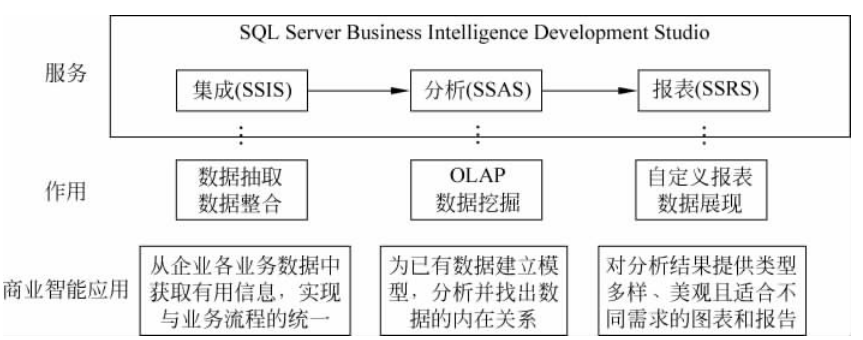

图 3.3　MS SQL Server 2005 的数据仓库架构

(1) SSAS(分析服务)提供了所有业务数据的统一整合视图，可以作为传统报表、在线分析处理、关键性能指示器(Key Performance Indicators，KPI)记分卡和数据挖掘的基础。同时提供了一个元数据模型以满足不同需求。其中的所有多维数据集和维度定义都可从统一空间模型(UDM)中查阅。UDM 是一个中心元数据库，其中定义了业务实体、业务逻辑、计算和度量，可作为所有报表、电子表格、OLAP 浏览器、KPI 和分析应用程序的源来使用。在 SQL Server 2005 中，关系数据库和多维数据库之间的界限变得模糊。可以将数据存储在关系数据库或多维数据库中，还可以使用主动缓存功能，充分利用两种数据库各自的优点。

(2) SSIS(集成服务)具有出色的 ETL 和整合能力,提供了构建企业级 ETL 应用程序所需的功能和性能,使得组织机构能更加容易地管理来自于不同的关系型和非关系型数据源的数据。SSIS 是可编程的、可嵌入的和可扩展的,这些特性使其成为理想的 ETL 平台。

(3) SSRS(报表服务)是一个基于服务器的完整报告平台,可创建、管理和交付传统报告和交互式报告。它包括创建、分发和管理报告所需的一切工具和信息。同时,其标准模块化设计和应用程序编程接口(API)使软件开发人员、数据提供商和企业能够集成原有系统或第三方应用程序中的报表功能。

在 SQL Server 2005 中可以安装示例数据库 Adventure Works DW(数据仓库示例数据库),其主要数据都源于另一示例数据库 Adventure Works(OLTP 示例数据库)。

本节以 SQL Server 2005 作为数据仓库环境来讲解数据仓库数据库的设计过程。示例数据是 SQL Server 2005 自带的示例数据库 Adventure Works 和 Adventure Works DW。

3.2.1　分析组织的业务状况及数据源结构

下面以 SQL Server 2005 示例数据库 Adventure Works DW 中所描述的 Adventure Works Cycles 公司的用户需求为例,介绍系统需求收集与分析过程。有关详细的业务信息参见 SQL Server 2005 有关数据仓库示例的帮助文档。

1. 公司概况

Adventure Works Cycles 是一家虚构的大型跨国制造公司,经营自行车及其相关配套产品,主要生产金属复合材料的自行车,产品远销北美、欧洲和亚洲市场。Adventure Works Cycles 公司总部设在华盛顿州的伯瑟尔市,雇用了 500 名工人。此外,在 Adventure Works Cycles 市场中还活跃着一些地区销售团队。

2000 年,Adventure Works Cycles 购买了位于墨西哥的小型生产厂 Importadores Neptuno。Importadores Neptuno 为 Adventure Works Cycles 产品系列生产多种关键子组件,这些子组件将被运送到伯瑟尔市进行最后的产品装配。2001 年,Importadores Neptuno 转型成为专注于旅游登山车系列产品的制造商和销售商。

实现一个成功的会计年度之后,Adventure Works Cycles 现在希望通过以下方法扩大市场份额:专注于向高端客户提供产品、通过外部网站扩展其产品的销售渠道、通过降低生产成本来削减其销售成本。

2. 原材料采购、生产和销售等环节的业务流程介绍

(1) 原材料采购与仓储业务流程。

在公司内部由采购部负责原材料采购,采购部门下设一个经理和多个采购员。一种原材料有多个供应商,一个供应商可以提供多种原材料。原材料和供应商之间是多对多的关系。每个采购员负责多种原材料的采购,一种原材料只能由一个采购员来采购。采购员和商品之间是一对多的关系。采购员只需了解原材料和供应商的联系,而采购部门经理需要管理员工,并且还需要了解原材料的库存情况,以确定需要采购的商品并将任务分配给每个采购人员。

公司为了防止产品过分依赖于原材料价格,还需要对原材料进行批量存储,因此设立仓

库管理部门,专门负责原材料的存储管理,仓库管理部门管理多个仓库,下设一个经理和多个仓库管理员,每个仓库中拥有多个仓库管理员,每个管理员只能在一个仓库中进行工作。仓库管理员需要知道他所管理的仓库中存储的原材料的种类、数量、存储的时间、原材料的保值期及原材料进入仓库和离开仓库的时间等信息。一种原材料可以保存在多个仓库中,一个仓库可以保存多种原材料。仓库管理部门经理不但需要处理仓库管理员需要的数据,而且需要知道仓库管理员的基本信息(例如仓库管理员的家庭住址和电话等)。

(2) 产品销售业务流程。

Adventure Works Cycles 的自行车及其相关产品远销北美、欧洲和亚洲市场。公司有网络销售和批发商销售两种销售渠道,因此,客户也分为两类,一类是从在线商店购买产品的消费者,通常是个人;一类是商店,即从 Adventure Works Cycles 销售代表处购买产品后进行转售的零售店或批发店。

对于销售部门,销售员关心的是商品的信息,即每种商品的价格、质量、颜色和规格等,以便向顾客推销相关的产品。因此,销售员最需要的数据就是商品的相关信息。销售部门经理一方面需要了解商品的销售情况,以便在某种商品缺货的时候通知仓储部门运送商品;另一方面,销售部经理还需要了解每个销售员的工作业绩,对每个销售员进行考核。因此,销售部门经理需要了解商品、顾客和部门员工的情况。

3. 对数据源结构的分析与理解

从上面的分析可以看出,业务数据确实是多维的。不同部门对数据的需求不同,同一部门人员对数据需求也存在差异。如果考虑数据需求的层次问题,管理人员和不同的业务人员对数据要求的程度也各不相同。管理人员可能需要综合度较高或较为概括的数据,而业务人员需要细节数据。

实际上,对业务的理解是所有信息系统建设过程所需要的,只不过在设计数据仓库时需要从业务蕴涵的数据视角来理解业务。数据仓库是以历史数据为基础的,这一步本质上是理解这些历史数据的来源。

通常,操作型业务数据是数据仓库数据库的来源和基础,只有对它的内容足够了解和理解,才能很好地设计数据仓库和对数据进行 ETL 处理。

首先是要了解数据源(操作型业务数据)的结构,例如,Adventure Works 示例数据库把 Adventure Works Cycles 公司的业务数据分成 5 大部分,分别是表示人力资源的 Human Resources、表示人员信息如客户或供应商联系人等的 Person、表示产品信息的 Production、表示采购信息的 Purchasing 和表示销售信息的 Sales。

其次是要明确数据的内容,数据内容包括某个业务领域的数据表结构及其主外键关系,还包括各个数据表的具体字段构成情况。Adventure Works 示例数据库的业务数据内容简述如下。

(1) 个人客户相关数据。个人客户是公司客户类型的一大类,即从 Adventure Works Cycles 在线商店购买产品的消费者。若 Sales. Customer 表的 CustomerType 列值为 I,则表示客户类型为个人,若为 S 则为批发商。与个人客户相关的表有 5 个,分别是:Person. Contact 表示客户的联系方式;Sales. Customer 表示客户的类型;Sales. Individual 表示个人客户的具体信息,其中 Demographics 列还以 XML 格式对个人客户的收入、爱好和车辆数目等进

行了统计；而对于客户的订单信息则放在 Sales. Sales Order Header 和 Sales. Sales Order Detail 两个表中。

(2) 产品相关数据。Adventure Works Cycles 公司提供 4 类产品，包括公司自己生产的自行车、自行车零部件(替换件，如车轮、踏板或刹车等部件)，还有从供应商处购买来转售给客户的自行车装饰和自行车附件等。和产品相关的表比较多，结构也较为复杂，产品相关的数据内容如表 3.1 所示。

表 3.1　产品(production)相关的表及其数据内容

数　据　表	数据表内容解释
Bill of Materials	制造自行车和自行车子部件的所有零部件列表(BOM 结构)，Product Assembly ID 列表示父级产品(即主产品)，ComponentID 表示组装用的零件
Culture	列出使用了哪些语言来本地化产品说明
Location	列出产品和零件的库存位置
Product	由公司销售或用来制造自行车和自行车组件的各种产品信息
Product Category	产品分类，例如自行车或附件
Product Cost History	列出不同时间点的产品成本
Product Description	列出各种语言的详细产品说明
Product Inventory	按地点统计的产品库存量
Product List Price History	列出不同时间点的产品价格
Product Model	与产品关联的产品型号
Product Model Product Description Culture	给出产品型号、产品说明及其本地化后的语言之间的交叉引用
Product Photo	列出所售产品的图像
Product Review	给出客户对产品的评价
Product Subcategory	产品类别的子类别

(3) 原材料采购相关数据。采购部门购买自行车生产中需使用的原材料和零件，同时也购买一些产品直接转售，例如自行车装饰件和自行车附件，像水瓶和打气筒等。和原材料采购相关的表的数据内容如表 3.2 所示。

表 3.2　原材料采购(purchasing)相关的表及其数据内容

数　据　表	数据表内容
Person. Address	客户的通信地址信息
Person. Contact	供应商雇员的姓名，与 Vendor Contact 表相关联，将联系人映射到供应商。XML 数据类型的列 Additional ContactInfo 包含了联系人的其他联系方式(手机和传真等)
Product Vendor	将供应商与其提供的产品建立对应关系
Purchase Order Detail	采购订单的明细信息，包括订购的产品、数量和单价等
Purchase Order Header	采购订单的头信息，包括应付款总计、订购日期和订单状态等。Purchase Order Header 表与 Purchase Order Detail 表构成主—从关系
Ship Method	用于维护产品标准发货方法的查找表。Ship Method ID 列包含在 Purchase Order Header 表中

续表

数　据　表	数据表内容
Vendor	供应商的详细信息,例如供应商的名称和账号
Vendor Address	将客户链接到 Address 表中的地址信息。按类型对地址进行分类,例如开票地址、家庭住址和发货地址等。Address Type ID 列映射到 Address Type 表中
Vendor Contact	这是一个关联表。连接 Contact 和 Vendor 两个表

以上对比较重要的业务数据表进行了解释,其目的是提供一个可供项目参考的示例,若要完全理解业务数据,还要对操作型业务数据库中的表结构及表间关系进行深入的理解。这里以原材料采购数据中的表 Purchasing. Purchase Order Header 为例子分析此表的具体结构,如表 3.3 所示。从后续的 ETL 处理及业务分析和业务规则挖掘等操作中会发现,对这些业务数据表的理解和分析是相当重要的。因此,在实际项目实施中,应该对重要的业务数据表进行类似的分析。

表 3.3　Purchasing. Purchase Order Header 的表结构

列	数据类型	说　　明
Purchase Order ID	int	主键
Revision Number	tinyint	用于跟踪一段时间内采购订单变化的递增编号
Status	tinyint	订单状态:1=等待批准,2=已批准,3=已拒绝,4=完成
Employee ID	int	创建采购订单的雇员。指向 Employee. Employee ID 的外键
Vendor ID	int	采购订单所采购的产品的供应商。Vendor. Vendor ID 的外键
Ship Method ID	int	发货方法,Ship Method. Ship Method ID 的外键
Order Date	datetime	采购订单的创建日期
Ship Date	datetime	预计供应商的发货日期
Sub Total	money	采购订单小计
Tax Amt	money	税额
Freight	money	运费
Total Due	money	应付款(SubTotal+TaxAmt+Freight)
Modified Date	datetime	上次更新日期和时间

3.2.2　组织需求调研,收集分析需求

数据仓库应用系统不同于事务处理业务系统,其数据分析需求刚开始时并不十分明确,而数据仓库的数据来源往往来自各操作型业务数据库的历史数据和当期数据,因此,项目需求的收集与分析需要从历史数据与用户需求两个方面同时着手,采用"数据驱动+用户驱动"的设计理念。

数据驱动是根据当前业务数据的基础和质量情况,以数据源的分析为出发点构建数据仓库。另一方面,用户驱动则是根据用户业务的方向性需求,从业务需要解决的具体问题出发,确定系统范围和需求框架,也叫需求驱动。

数据仓库的用户一般是企业或事业单位的管理者,在设计数据仓库系统时充分考虑用户的分析需求是十分必要的。同时,由于数据仓库的构建必须基于业务数据库等数据源,数

据源的结构也是不得不考虑的问题。如图 3.4 所示，常常采用“两头挤法”找出数据仓库系统的真正需求。

图 3.4　用户驱动与数据驱动相结合示意图

在 3.2.1 节的分析和理解业务数据库(Adventure Works)的过程中明确了企业的具体业务映射到数据库系统的细节，对从现有数据源中如何获取企业数据仓库需求对应的数据已经心中有数。现在我们从企业的各个视角对此企业数据仓库的分析需求做进一步的分析，发现企业需要且可以构造的主题。

实际上，企业每个部门都有观察企业业务的不同视角，这是需求多样性的一个方面。例如，对于 Adventure Works Cycles 公司来说，销售部门、采购部门和仓库管理部门等都有相应的视角，尽管这些业务是相关的，但是对数据的需求，特别是对分析数据的需求必然有所不同。

创建企业级数据仓库是一个面向企业各部门特别是管理部门的工程，但它的需求在前期可能是相当模糊的，因此，我们应该高度重视需求的调研与分析，尽量多地挖掘出用户的当前需求和潜在需求。

1. 关于用户需求的调研

用户方面，从组织机构的上层开始交流是非常有益的。上层行政官员可以提供许多令人惊奇的有关业务操作及其希望从该组织得到的内容。除此之外，还应该包括负责数据仓库项目或有关业务领域的行政职员，以及来自相关业务领域的负责向高级行政官员汇报的主管经理和为高级行政人员和主管经理准备报告的业务分析员。

根据不同的交谈对象，所提问题也应有所不同。很明显，针对高级行政人员和基层职员应该有不同的问题。通常也有些共性的问题域，例如：

(1) 他们的工作成绩是怎样得来的？即什么因素决定工作的成功与失败？

(2) 工作所需信息的分析过程需耗费多长时间？从这些分析中可作出什么决策？

(3) 信息分发的方式是什么？是报告、论文还是电子邮件？

(4) 怎样弥补信息的空缺？

(5) 分析这些数据需要哪一级的详细程度？

(6) 业务报表的来源是什么？谁对报告的制定、维护和分发负责？

对需求按照业务视角进行分类以后，可以进一步细化需求的提问，一般从业务目标、当前信息源、涉及的主题范围、关键性能指标和信息频率等方面分别提问。

(1) 业务目标：部门职责和目标是什么？怎样将这些目标融进企业的目标之中？要达到这些目标有哪些需要？成功的关键因素是什么？集团公司实现这些目标的障碍有哪些？需要购买外部数据吗？从哪里购买？

(2) 当前信息源和日常报表需求：在现有的日常报表过程中，当前传递了哪些信息？从何处获取这些分析数据？现在是如何加工处理的？这些信息的详细程度怎样？是太详细了，还是太粗略了？哪些操作会产生关于重要主题领域的数据和信息？

(3) 主题领域：有关这方面的问题可以帮助确定对业务活动来说什么主题是很重要的。随着用户地位的不同，在他们的数据领域中，各种不同的领域或维度被明确地提出，这就使得对信息的整合变得容易了。这些问题集中在数据仓库中的数据应怎样被检索及用户怎样分析和筛选这些数据等。关于主题领域的问题如：

① 哪些维度或领域对数据的分析是有价值的？这些维度有固有的层次结构吗？

② 作出业务决策仅仅需要当地的有关信息吗？

③ 某些产品是否仅仅在某一地区销售？

(4) 关键性能指标：不同的用户会有不同的看法。例如，部门的绩效是怎样监测的？部门内部提供哪些关键的指标？

(5) 信息频率：可以从用户处理信息的时间灵敏度获得信息频率。如用户需要多长时间对数据更新一次？适当的时间结构是什么？在数据仓库中，对信息有实时性需求吗？

2. 对用户需求调研结果的分析

在与用户交流阶段，应该确定数据仓库需要访问的有关信息。用户应该清晰地确定所需的信息。例如，数据仓库的用户需要得到有关产品收入的详细统计信息，包括过去 5 年中的年龄、组别、性别、位置和经济状况等信息。

然后根据用户的信息需求，抽取出信息的度量值和维度信息，例如对于需要观察的产品收入，可以确定其度量指标和维度如下。

(1) 度量指标：包括产品销售的实际收入、产品销售的预算收入及产品销售的估计收入。

(2) 维度：包括已经销售的产品信息、销售地点(位置信息)和顾客信息(如年龄组别、性别、位置和经济状况)等。

假定 Adventure Works 的销售和营销团队以及高级管理人员对数据分析有如下需求。

(1) 目前的报表是静态的。用户无法通过交互方式探测报表中的数据以获取更详细的信息，例如，他们可以处理 Microsoft Office Excel 透视表。虽然现有的一组预定义报表足以供许多用户使用，但更高级的用户却需要对数据库进行直接查询访问，以进行交互式查询和访问专用报表。

(2) 查询性能差异很大。例如，有些查询只需几秒钟便可返回结果，而另一些查询需要几分钟才能返回结果。

(3) 用户所在的业务部门不同，其感兴趣的数据视图也不同。每个组都很难理解与其不相关的数据元素。

(4) 业务用户很难构造一些专用查询，以组合两个相关的信息集(如销售额和销售配额)。此类查询会占用大量的数据库空间，因此，公司要求用户向数据仓库团队请求跨主题区域的数据集。

(5) 希望通过一个通用的元数据层提供统一的数据访问以进行分析和报告。

(6) 简化用户的数据视图，从而加速交互式查询、预定义查询以及预定义报表的开发。

总之，通过对历史数据和需求的分析，可以明确用户正在使用的数据现状、他们如何使用这些数据及他们将利用数据仓库干什么。充分的交流为数据仓库的总体设计打下基础。

3.2.3　采用信息包图法设计数据仓库的概念模型

在收集分析需求并做了详细的需求调研之后，我们对企业需求有了一个比较清晰的了解，这时可以对数据仓库的概念模型作设计，通常采用面向主题的自顶而下的设计方法，数据仓库的概念模型将面向主题，也就是面向对象，示例数据库中的对象如客户、产品和供应商等多维信息。终端用户通过各种维度来获取业务数据，其中时间是最基本、最关键的维度。对于面向主题的数据仓库同传统的数据库设计一样需要经历概念模型设计、逻辑模型设计和物理模型设计三个阶段(如表 3.4 所示)。与之相对应，数据仓库的设计方法分别是针对数据仓库的信息包图设计、星型图模型设计和物理数据模型设计，要求如下。

表 3.4　数据仓库与 OLTP 数据库的设计方法比较

设 计 阶 段	数 据 仓 库	OLTP 数据库
概念模型	信息包图	数据流程图
逻辑模型	星型图模型	实体关系图
物理模型	物理数据模型	物理数据模型

(1) 数据仓库的概念模型通常采用信息包图法来进行设计，要求将信息包图的 5 个组成部分(名称、维度、类别、层次和度量)全面地描述出来。

(2) 数据仓库的逻辑模型通常采用星型图法来进行设计，要求将星型图的 5 类逻辑实体(度量逻辑实体、维度逻辑实体、层次逻辑实体、详细信息逻辑实体和类别逻辑实体)完整地描述出来。

(3) 数据仓库的物理模型通常采用物理数据模型法来进行设计，要求将物理数据模型的 5 类表(事实表、维表、层次表、详细信息表和类别表)详细地描述出来。

在与用户交流的过程中，前两节确定了数据仓库所需要访问的信息，这些信息包括当前的、将来的以及与历史相关的数据。本节将确认操作数据、数据源以及一些附加数据需求，建立信息包图，进而确定数据仓库中的主题和元数据，有效地完成查询和数据之间的映射，完成概念数据模型的设计。

1. 信息包图法简介

由于数据仓库的多维性，利用传统的数据流图进行需求分析已不能满足需要。因此，数据仓库的建模包括超立方体(hypercube)法及信息包图法。

超立方体法也是采用自上而下的方法设计，其步骤如下。

(1) 确定模型中需要抓住的业务过程，例如销售活动或销售过程。

(2) 确定需要捕获的度量值，例如销售数量或成本。

(3) 确定数据的粒度，即需要捕获的最低一级的详细信息。

由于超立方体法在表现上缺乏直观性，尤其是当维度超出三维后，数据的采集和表示都比较困难，这时可以采用信息包图法在平面上展开超立方体，即用二维表格反映多维特征。信息包图提供了一个多维空间来建立信息模型，并且提供了超立方体的可视化表示。

信息包图定义主题内容和主要性能指标之间的关系，其目标就是在概念层满足用户需求。信息包图拥有三个重要对象：(度量)指标、维度和类别。利用信息包图设计概念模型

就是要确定这三个方面的内容。

(1) 确定指标。(度量)指标表明在维度空间衡量业务信息的一种方法,是访问数据仓库的关键所在,是用户最关心的信息。成功的信息包可以保证用户从信息包中获取需要的各个性能指标参数。

(2) 确定维度。维度提供了用户访问数据仓库信息的途径,对应超立方体的每一面,位于信息包图第一行的每一个栏目中。

(3) 确定类别。类别是在一个维度内为了提供详细分类而定义的,其成员是为了辨别和区分特定数据而设,它说明一个维度包含的详细信息,一个维度内最底层的可用分类又称为详细类别。

信息包图法也叫用户信息需求表,就是在一张平面表格上描述元素的多维性,其中的每一个维度用平面表格的一列表示,例如时间、地点、产品和顾客等。而细化本列的对象就是类别,例如时间维度的类别可以细化到年、月、日,甚至小时。平面表格的最后一行(代表超立方体中的单元格)即为指标度量值,例如,某年在某销售点的某类产品的实际销售额。创建信息包图时需要确定最高层和最低层的信息需求,以便最终设计出包含各个层次需要的数据仓库。

对较复杂的业务进行需求分析时,有时一张信息包图不能反映所有情况,需要设计多张不同的信息包图来满足全部需求,此时应保证多个信息包图中出现的维度信息和类别信息完全一致。

总之,信息包图法是一种自上而下的数据建模方法,即从用户的观点开始设计(用户的观点是通过与用户交流得到的),站在管理者的角度把焦点集中在企业的一个或几个主题上,着重分析主题所涉及数据的多维特性,这种自上而下的方法几乎考虑了所有的信息源,以及这些信息源影响业务活动的方式。整个数据仓库数据库的设计过程分为如下 4 个阶段。

(1) 采用自顶向下的方法对业务数据的多维特性进行分析,用信息包图表示维度和类别之间的传递和映射关系,建立概念模型。其中,类别是按一定的标准对一个维度的分类划分,如产品可按颜色、质地、产地和销地等不同标准分类。

(2) 对企业的大量数值指标实体数据进行筛选,提取出可利用的中心度量指标(也称为关键性能指标和关键业务度量值),例如产品收入、产品成本或设备运行时间等。

(3) 在信息包图的基础上构造星型图,对其中的详细类别实体进行分析,进一步扩展为雪花图(可选),建立逻辑模型。

(4) 在星型图或雪花图的基础上,根据所定义的数据标准进一步对实体、键属性、非键属性、数据容量和更新频率等进行定义,完成物理数据模型的设计。

2. 信息包图的建立

利用信息包图可以完成以下工作。

(1) 定义业务中涉及的共同主题范围,例如时间、区域、产品和客户等。

(2) 设计可以跟踪的、确定一个业务事件怎样被运行和完成的关键业务指标。

(3) 决定数据怎样被传递给数据仓库的用户。

(4) 确定用户怎样按层次聚合和移动数据。

(5) 确定在给定的用户分析或查询中实际包含了多少数据。

(6) 定义怎样访问数据、估计数据仓库大小、确定数据仓库里数据的更新频率。

下面以 Adventure Works DW 示例数据仓库中的 Adventure Works Cycles 公司的销售情况为例说明信息包图的制作。通过对 Adventure Works Cycles 公司近年来销售情况的进一步了解和分析,可以得到如下结论。

(1) 获取各个业务部门对业务数据的多维特性分析结果,确定影响销售额的维度,包括时间、区域、产品和客户等维度。

(2) 对每个维度进行分析,确定维度与类别之间的传递和映射关系,如在 Adventure Works 业务数据库中,时间维有年度、季度、月和日等级别,而区域分为国家、省州、城市和具体的销售点。

(3) 确定用户需要的度量指标体系,这里以销售情况作为事实依据确定的销售相关指标包括实际销售额、计划销售额和计划完成率等。

有了以上的分析,就可以画出销售分析的信息包图,如图 3.5 所示。此信息包图以销售分析为主题,归纳事实和指标,归纳维度和层次,确定数据的粒度和类别。

维度→		信息包图：销售分析			
类别↓	时间维	区域维	产品维	客户维	广告维(待用)
	年度(5)	国家(10)	产品类别(500)	年龄分组(7)	广告费分组(5)
	季度(20)	省州(100)	产品名称(9000)	收入分组(8)	
	月(60)	城市(500)		信用组(2)	
	日(1800)	销售点(8000)			
度量指标：实际销售额、计划销售额、计划完成率					

图 3.5　销售分析的信息包示意图

图 3.5 仅为示意图,图中的第一行表示各个维度,每一列表示不同维度的类别,其中括号内的数字表示各类别的数目(这里的数字仅为示意)。通常一个维度的类别种类不能太多,建议一个维度的类别数不超过 7,这将有助于用户检索和理解数据,提高数据的可利用性。在指标栏里,定义了三种度量指标,即实际销售额、计划销售额和计划完成率,并且以这三个指标为中心展开分析。其他业务分析需求的信息包图也可采用类似的方法。

3. 设计基于主题域的概念模型

通过信息包图实际上确定了数据仓库的主题和大部分元数据。

所谓主题(subject),是指在较高层次上将业务数据进行综合、归类和分析利用的一个抽象概念,每一个主题基本对应业务的一个分析领域。如在前面信息包图示例中,“销售分析”就是一个分析领域,也称为一个应用主题。

面向主题的数据组织方式,就是在较高层次上对分析对象数据的一个完整并且一致的描述,能刻画分析对象所涉及的各项业务数据,以及数据之间的联系。与 OLTP 数据库面向事务处理应用进行数据组织的特点不同,数据仓库中的数据是面向主题组织的。如一个数据仓库系统涉及的主题可能是产品销售分析、货物发运分析等。

主题是根据分析需求确定的。如在生产企业中,对于材料供应,在 OLTP 系统中,我们往往关心的是怎样更方便和更快捷地进行材料供应的业务处理;而在做分析处理时,我们

就更关心材料的不同采购渠道和材料供应是否及时、材料质量状况等。典型的主题领域包括顾客、产品、订单和财务或是其他某项活动。

主题域是对某个主题进行分析后确定的主题边界。在大多数数据仓库的设计过程中,都有一个主题域的选择过程。主题域的确定通常由最终用户和数据仓库的设计人员共同完成。例如,对于 Adventure Works Cycle 公司的管理层可能需要分析的主题包括供应商、商品、客户和仓库等。其中商品主题的内容包括记录超市商品的采购情况、商品的销售情况和商品的库存情况等;客户主题包括的内容可能有客户购买商品的情况;仓库主题包括仓库中商品的存储情况和仓库的管理情况等。确定主题边界(主题域)实际上是进一步理解业务关系,因此在确定分析主题后,还需要对这些主题进行细化以获取每一个主题应有的边界。例如,根据分析需求所确定的 Adventure Works Cycle 公司的分析主题及主题域结构如图 3.6 所示。

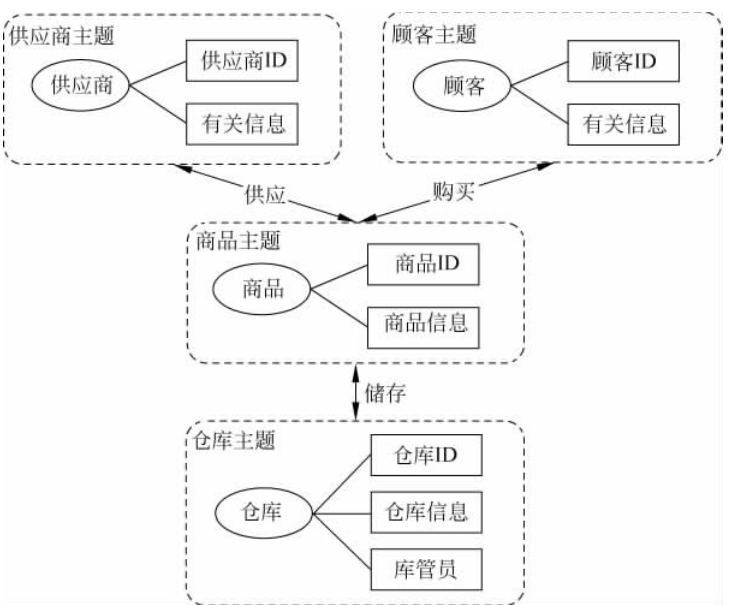

图 3.6　主题及主题域的划分

由于数据仓库的设计是一个不断改进和完善的螺旋式发展过程,在刚开始时选择部分比较重要的主题作为数据仓库设计的起点是很有必要的。例如,在 Adventure Works DW 数据仓库的概念模型设计中,在对需求进行分析后,认识到"商品"主题既是一个销售型企业最基本的业务对象,又是进行决策分析的主要领域,通过"商品"主题的建立,经营者就可以对整个企业的经营状况有较全面的了解。先实施"商品"主题可以尽快地满足企业管理人员建立数据仓库的最初要求。通过将主题边界的划分应用到已经得到的关系模型上,即将主题域的划分和事务处理数据库中的表结合起来,便能形成原始的概念模型,例如在上述主题示例中,商品主题可能涵盖的关系表有商品表、供应关系表、购买关系表和仓储关系表;仓库主题可能涵盖的关系表有仓库关系表、仓库表、仓库管理关系表和管理员表。把这些表的键和字段联系起来,就可以形成图 3.7 所示的原始概念模型(实体关系图)。

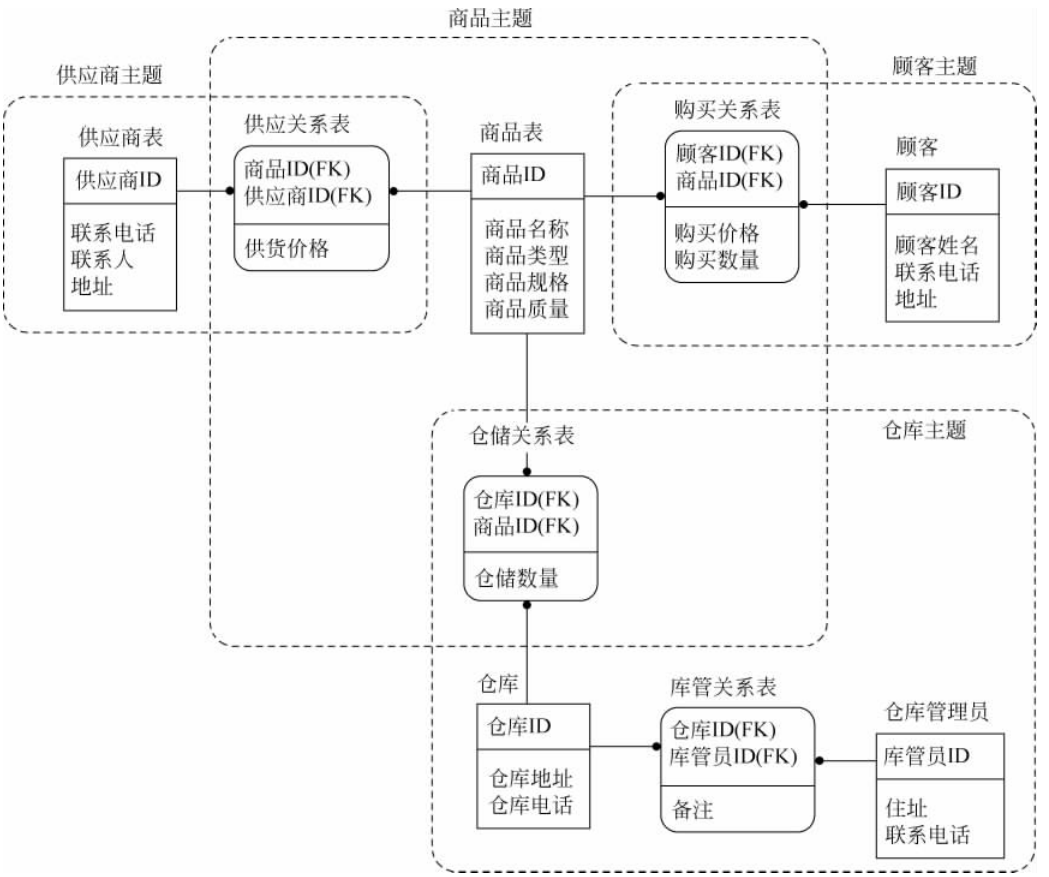

图 3.7　划分了主题域的原始概念模型(ER 图)

3.2.4　利用星型图设计数据仓库的逻辑模型

1. 根据分析需求与信息包图制作星型图或雪花图

在传统的数据库逻辑模型设计中,根据需求分析阶段获得的数据流图,利用实体关系图(ER 模型)将概念模型转换为逻辑模型。数据仓库系统通常是在信息包图的基础上构建星型图,进一步完成逻辑模型设计。

星型图因其外观似五角星而得名,它支持从业务决策者的角度定义数据实体,满足面向主题数据仓库设计的需要,而信息包图又为星型图的设计提供了完备的概念基础。同信息包图中的三个对象相对应,星型图拥有三个逻辑实体,即维度、指标和类别。

位于星型图中心的实体是(度量)指标实体,对应信息包图中的指标对象,是用户最关心的基本实体和查询活动的中心,为用户的业务活动提供定量数据。每个指标实体代表一系列相关事实,完成一项指定的功能,在一般情况下代表一个现实事务的综合水平,仅仅与每个相关维度的一个点对应。位于星型图星角上的实体是维度实体,对应信息包图中的维度对象,其作用是限制用户的查询结果,将数据过滤使得从指标实体查询返回较少的行,从而缩小访问范围。另一个实体是详细类别实体,它对应信息包图中的类别对象。一个维度内

的每个单元就是一个类别,代表该维度内的一个独立层次,它要求更加详细的信息才能满足用户的需要,与相应的事务处理业务数据库结构产生映射。

因此,从信息包图转换成星型图,需要定义如下三个实体。

(1)(度量)指标实体。使用每一个指标,同时确定是否存储经过计算的指标。

(2)维度实体。一个维度实体对应指标实体中的多个指标。用户利用维度实体来访问指标实体,一个维度实体对应信息包图中的一个列。

(3)详细类别实体。对应现实世界的某一实体。

在星型图中,用户通过维度实体获得指标实体数据,其中指标实体与维度实体间的联系通过每个维度中的最低一层的详细类别实体连接。

当多个信息包图转换成星型图时,可能出现维度实体的交叉重叠,为了保证实体的一致性需要进行统一处理,确定它们是同一实体在不同层次上的数据反映,还是两个不同的实体。当多个维度实体相关并且存在共性时,可能需要进行合并处理。

在 3.2.3 节的示例信息包图(图 3.5)基础上构造一个星型图,如图 3.8 所示。

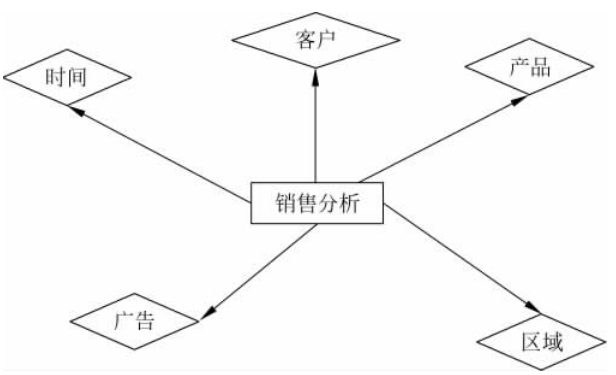

图 3.8 Adventure Works Cycles 公司销售分析星型图

根据实际业务需求,可以把产品类别实体连接到星型图中,就可以进一步得到企业数据仓库的雪花模型。如 Adventure Works 业务示例数据库中,通过表 Product Category、Product Subcategory 和 Product 对产品进行了层次分类,把这三个表对应的实体挂到图 3.8 的星型图中,便形成了如图 3.9 所示的雪花图。

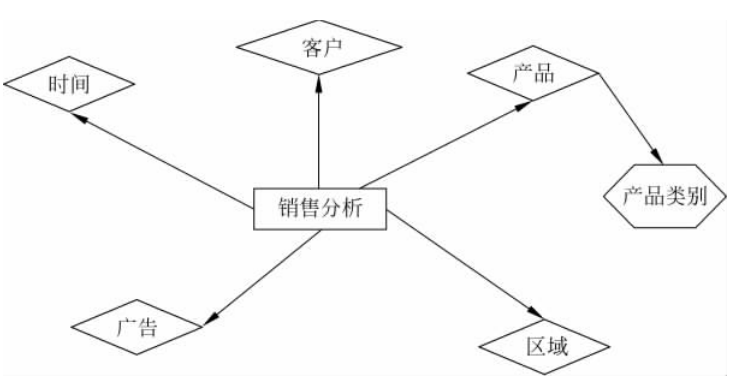

图 3.9 在星型图基础上构建的雪花型模式图

2. 确定主题的属性组

根据概念模型中定义的主题及主题域，可以进一步确定在主题的逻辑关系模式中包含所有的属性及与系统相关的行为。数据仓库中数据存储的逻辑结构也需要在逻辑模型的设计阶段完成定义，需要给主题增加属性组和其他所需信息。以 Adventure Works Cycle 公司数据仓库为例，在“商品”“销售”和“客户”主题上增加能进一步说明主题的属性组（如表 3.5 所示）。

表 3.5　主题的详细描述

主　题　名	公　共　键	属　性　组
商品	商品号	基本信息：商品号、商品名、类型和颜色等 采购信息：商品号、应商号、供应价、供应日期和供应量等 库存信息：商品号、库房号、库存量和日期等
销售	销售单号	基本信息：销售单号、销售地址等 销售信息：客户号、商品号、销售价、销售量和销售时间等
客户	客户号	基本信息：客户号、客户名、性别、年龄、文化程度、住址和电话等 经济信息：客户号、年收入和家庭总收入等

3. 事实表及其特征

度量是客户发生事件或动作的事实记录，例如客户打电话，可能选择的度量有通话时长、通话次数和通话费用等；客户购买商品，可能选择的度量有购买的次数、购买商品的金额和购买商品的数量等。度量变量的取值可以是离散的数值，也可以是连续的数值。例如，客户通话次数是离散的数值，而客户购买商品的金额是连续的数值。度量变量也可以在某个元素集合内取值。例如，客户对公司服务质量的评定可以是“优”“良”“中”和“差”中的一个。业务事实是对某个特定事件的度量，是各个维度的交点。

事实表则是在星型模式或雪花型模式中用来记录业务事实并作相应指标统计的表，同维表相比，事实表具有如下特征。

（1）记录数量很多，因此事实表应当尽量减小一条记录的长度，避免事实表过大而难以管理。

（2）事实表中除了度量变量外，其他字段都是维表或者中间表（对于雪花型模式）的关键字（外键）。

（3）如果事实相关的维度很多，则事实表的字段数也会比较多。

4. 事实表的类型与设计

事实表是星型模式或类似结构的核心，包含了基本业务事务的详细信息。事实表一般包含两个部分：一部分是由主键和外键所组成的键部分；另一部分是用户希望在数据仓库中了解的数值指标，这些指标是为每个派生出来的键而定义和计算的，称为事实或度量指标。由于事实是一种度量，所以事实表中的这种指标往往需要具有数值化和可加性（或可平均等）的特征。但是在事实表中，只有那些具有完全可加性的事实才能根据所有的维度进行

累加而具有意义。而事实表中有一些事实表示的是某种强度,这类事实就不具有完全加法性,而是一种半加法性。例如,账目余款反映的是某个时间点的数据,它可以按照地点和商品等大多数维度进行累加,但是对于时间维度则例外,将一年中每个月的账目余款进行累加是毫无意义的,而决策者则可能需要了解所有地区和所有商品账目余款的累加值。在事实表中还有一些事实是非加法性的,即这些事实具有对事实的描述特性,在这种情况下一般要将这些非加法性事实转移到维度表中。

按照事实表中度量的可加性情况,可以把事实表及其包含的事实分为如下 4 种类型。

(1) 事务事实。以组织事件的单一事务为基础,通常只包含事实的次数。例如银行的 ATM 提款机的提款次数,使用某种服务的次数等。

(2) 快照事实。以组织在某一特定时间的特殊状态为基础。也就是只有在某一段时间内才出现的结果。

(3) 线性项目事实。这类事实通常用来储存关于企业组织经营项目的详细信息。包括表现与企业相关的个别线性项目的所有度量条件,如销售数量、销售金额、成本和运费等数值数据,也就是关键性能指标。此类事实运用范围广,如采购、销售和库存等。

(4) 事件事实。通常表示事件发生与否及一些非事实本身具备的细节。它所表现的是一个事件发生后的状态变化。如哪些产品在促销期间内没有卖出(有还是没有)。

在事实表模型的设计中还需要注意到派生事实。派生事实主要有两种,一种是可以用同一事实表中的其他事实计算得到,例如销售行为中的商品销售均价可以用商品的销售总金额和销售数量计算得到,对于这些派生事实一般不保留在事实表中;另一种是非加法性事实,例如各种商品的利润率等各种比率。

在事实表模型的设计中必须要分析和确定事实表中的这些事实特性,可能需要经过多次反复来确定。首先,通过调查确定所有可能的基本事实和派生事实;然后,对所有的事实按照功能或某种方式进行排序,以删除重复的事实;接着,确认那些基于不同准则但是有相同性质的派生事实,例如公司门市销售总额与地区销售总额虽由于维度的不同而被定义为不同的事实,但实际计算方法是一样的;最后,再一次确定事实表模型,在确认中要检查所有的计算派生事实的基本事实是否已经包含在模型中,并且与用户取得一致。

在设计事实表时,一定要注意使事实表尽可能地小,因为过于庞大的事实表在表的处理、备份和恢复及用户的查询等方面需要较长的时间。在实际设计时,可以利用减少列的数量、降低每一列的大小和把历史数据归档到单独的历史事实表中等方法来降低事实表的大小。另外,在事实表中还要解决好数据的精度和粒度的问题,下面将阐释粒度的设计方法。

5. 粒度的选择与设计步骤

第 2 章指出数据仓库中的数据可分为 4 个级别,即早期细节级、当前细节级、轻度综合级和高度综合级。源数据经过 ETL 处理后,首先进入当前细节级,并根据具体需要进一步综合,从而进入轻度综合级甚至高度综合级,老化的数据将进入早期细节级。数据仓库中存在的这种不同综合级别,就是“粒度”的直观表现。

粒度模型也是数据仓库设计中十分重要的问题之一。所谓粒度,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。

(1) 粒度的不同选择会导致逻辑模型的差异

先看一个粒度设计的例子，如果 Adventure Works Cycles 公司的管理者想按照国家(country)、区域(region)、子区域(subregion)和子区内的销售员这样的层次关系来查看公司的销售情况，其雪花型模式的 ER 图如图 3.10 所示，它是通过将地理概念层次的国家、区域和子区域嵌入到销售员维度得到的。

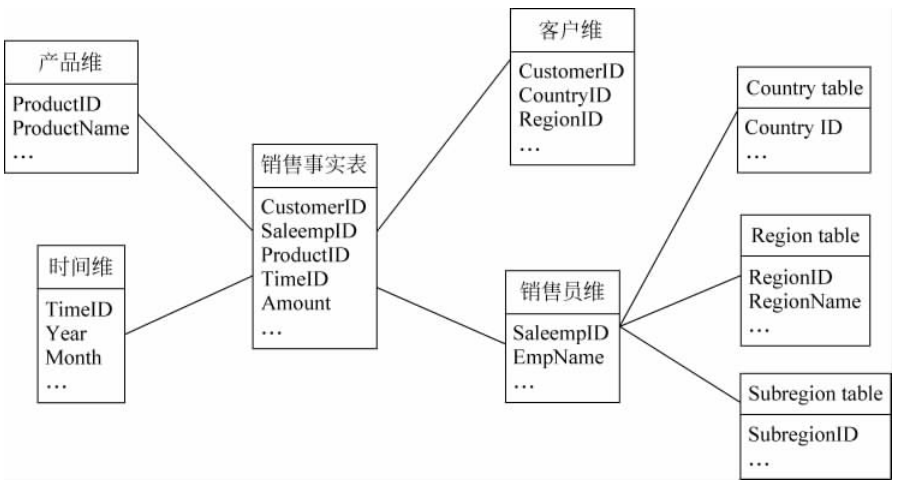

图 3.10　细化到销售员层次的逻辑模型

如果公司的决策者认为不需要了解具体到某个销售人员的情况，而只需要了解各个地理区域的销售情况，则没有必要把销售员维作为一个维度，把地域相关的表综合成为地理维度就可以了，设计结构如图 3.11 所示。

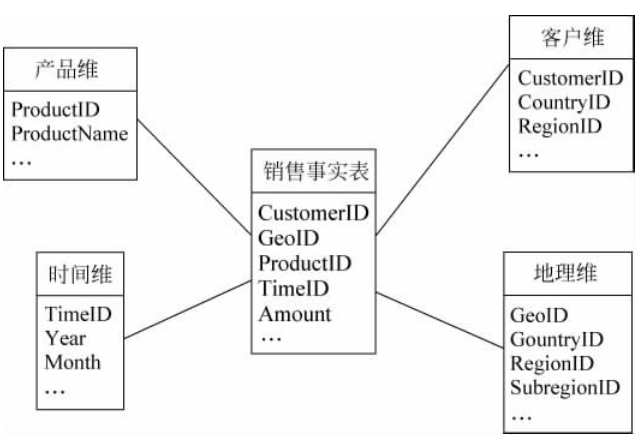

图 3.11　细化到子区域层次的逻辑模型

由以上实例可知，对事实粒度需求的不同，会直接导致数据仓库逻辑设计的差异。

(2) 粒度的不同选择会导致数据存储容量的差异

粒度对数据仓库最直接的影响就是存储容量。例如，按月统计的客户购买数据和按次记录客户购买数据(即记录每笔销售)，两者的数据量相差极大，假定每个字段为 8 字节，每

个客户一天有 5 次消费,则 1 个客户 1 个月的消费细节数据的数据量为 8×6×30×5=7200 字节,而 1 个客户 1 个月的消费汇总数据的数据量为 8×4=32 字节,如图 3.12 所示。

图 3.12　不同粒度的事实表示例

(3) 粒度的设计步骤

由以上的分析可知,数据仓库分析功能和存储空间是一对矛盾。如果粒度设计得很小,则事实表将不得不记录所有的细节,储存数据所需要的空间将会急剧膨胀;若设计的粒度很粗,决策者则不能观察细节数据。粒度设计主要完成以下两个步骤。

① 粗略估算数据量,确定合适的粒度级的起点。即粗略估算数据仓库中将来的数据行数和所需的数据存储空间,例如,预估一年及 5 年内表中的最少行数和最多行数,并对每张表确定键码的长度和原始表中每条数据是否存在键码。

② 确定粒度的级别。在数据仓库中确定粒度的级别时,需要考虑这样一些因素:分析需求类型、数据最低粒度和存储数据量。

分析需求类型直接影响到数据仓库的粒度划分,将粒度的层次定义得越高,就越不能在该数据仓库中进行更细致的分析。例如,将粒度的层次定义为月份时,就不可能利用数据仓库进行按日汇总的信息分析。因此,数据粒度的划分策略一定要保证数据的粒度确实能够满足用户的决策分析需要,这是数据粒度划分策略中最重要的一个准则。

数据仓库通常在同一模式中使用多重粒度,这是以数据仓库中所需的最低粒度级别为基础设置的。例如,当前(当年)数据的数据粒度和历史数据的数据粒度可以不同,形成双重粒度,即用低粒度数据保存近期的财务数据和汇总数据,对时间较远的财务数据只保留粒度较大的汇总数据。这样既可以对财务近况进行细节分析,又可以利用汇总数据对财务作趋势分析。

定义数据仓库粒度的另外一个要素是数据仓库可以使用多种存储介质的空间量。如果存储资源有一定的限制,就只能采用较高粒度的数据粒度划分策略。这种粒度划分策略必须依据用户对数据需求的了解和信息占用数据仓库空间的大小来确定。随着存储硬件的加速发展,这个因素的影响将越来越不重要。

(4) 粒度设计示例

下面以 Adventure Works Cycles 公司的生产部门数据仓库设计为例,如图 3.13 所示。这里采用多重粒度设计。左边是操作型业务数据,记录完成若干给定部件的生产线运转情况,每一天都会积累许多记录,是生产业务的详细数据,最近 30 天的业务详细信息都存储在 OLTP 环境中。

图 3.13 的右边是轻度汇总级的数据,轻度汇总级包括两个表,一个汇总某一部件在 3 个月中的生产情况,另一个汇总部件的组装情况,汇总周期为 1 年。生产档案表则包括每个

生产活动的详细记录。

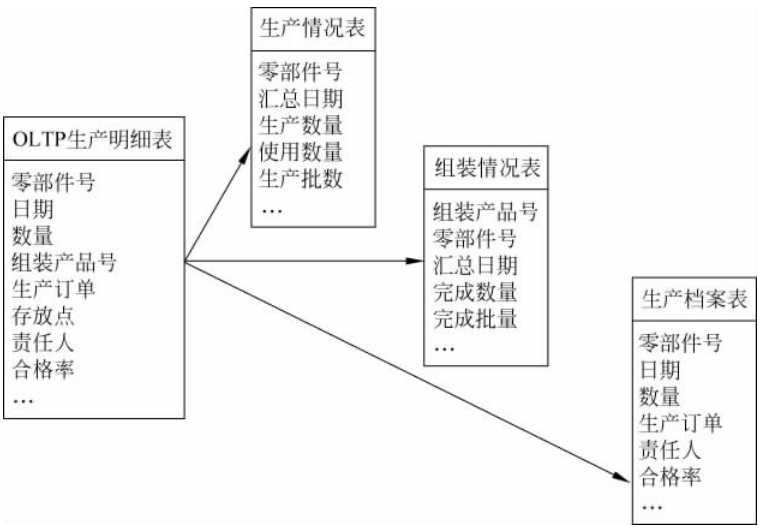

图 3.13　Adventure Works Cycles 公司的生产业务的多重粒度设计示例

6. 关于数据仓库的聚合模型

在事实表中存放的度量变量，根据其实际意义可分成可加性度量变量和非可加性度量变量。可加性度量变量是指将变量相加后得到的结果仍然具有实际意义，可以把此结果计算后放在事实表中，以便在以后的查询中直接使用，这个相加的结果就是聚合。例如每个月的销售金额，通过将 3 个月的销售金额相加，就可以得到 1 个季度的销售金额；通过将 12 个月的销售金额相加，可以得到全年的销售总金额。

确定了数据仓库的粒度模型以后，为提高数据仓库的使用性能，还需要根据用户需求设计聚合。数据仓库中各种各样的聚合数据主要是为了使用户获得更好的查询性能，因此聚合模型的好坏将在很大程度上影响到数据仓库的最终使用性能。

在数据仓库的聚合设计中，应该对每个维进行审查，以确定哪些属性经常用于分组，这些属性的组合有多少。例如，假定某一主题有 4 个维度，每个维度有 3 个可以作为聚合的属性，那么最多可以创建 256 个不同的聚合。在实际工作中没有必要创建这么多聚合，我们要根据用户的分析需求来创建用户经常使用的聚合。

数据仓库的聚合模型设计与数据仓库的粒度模型紧密相关，如果数据仓库的粒度模型只考虑了细节数据，那么就可能需要多设计一些聚合；如果粒度模型为多层数据结构，则在聚合模型设计中可以少考虑一些聚合。

7. 关于数据的分割处理

数据分割是把数据分散到各自的物理单元中去，使它们能独立地处理。分割是数据仓库中继粒度问题之后的又一个主要的设计问题。

为什么分割如此重要呢？因为小的物理单元能为操作者和设计者在管理数据时提供更大的灵活性。数据仓库的本质之一就是灵活地访问数据，因此对所有级别为“当前细节”的

数据仓库数据都可考虑使用分割技术。分割的原理如图 3.14 所示,由于全部销售记录过于庞大,可以按照不同的年度把它分割为若干个物理单元。

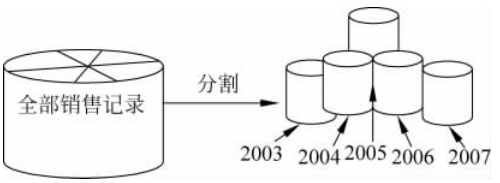

图 3.14　数据分割处理

在项目实施时,根据事实表的特点和用户的查询需求,可以选用时间、业务类型、区域和下属组织等多种数据分割类型。

8. 星型图中的维度表简介

在图 3.10 所示的星型图中,处于星型结构中央的事实表不仅包含度量指标,而且也包含维度表的外键和事实表的主键,而维度表除了代表维的主键之外,还有其他属性字段,例如客户维度表包含一个主键 Customer Key 和有关客户信息的其他字段。维度由主键和维属性构成,维属性即是维度表里的列。维元素定义维度表中的层次关系,属性则以用户熟悉的术语描述维元素。

在设计过程中,来自数据源的数值数据字段到底是一个度量事实还是一个维度的属性?一般情况下,如果数值数据字段的度量经常改变,那么它就是事实;如果它是离散值性质的描述属性,且几乎保持为常数,那么它就是维属性。

(1) 维度表应有的数据特征

① 维度通常使用解析过的时间、名字或地址元素,这样可以使分析查询更灵活。例如时间可分为年、季、月、周和日等;个人名字可以分为姓氏和称谓等;地址则可以用地理区域来区分,如国家、省、市和县等。

② 维度表通常不使用业务数据库的键值作为主键,而是对每个维度表另外增加一个额外的键值字段作为主键来识别维表实体,最常使用的字段类型是 Identity 类型。在维表中新设定的键也叫代理键。

③ 维度表应该包含随时间变化的数据记录字段,当数据集市或数据仓库的数据随时间变化而有额外增加或改变时,维表的数据行应有标识此变化的字段。

(2) 维度表中维度的分类

维度的类型包括结构维、信息维、分区维、分类维、退化维和一致维等多种类型。

① 结构维。结构维表示在层次结构组成中的信息度量。如年、月和日可以组成一个结构维。又如将销售量用作一个度量,与这个度量相关的维度表是包含产品属性的产品信息表。产品信息表的维包括产品名(product_name)、品牌(product_brand)、类别(product_category)和产品家族/系列(product_family)等。这些维度也组成一个结构维,在这个示例中,如果再增加一个时间信息表,由年、月和日组成的时间信息对象建立一个时间结构维。通过这两个结构维,可以用 OLAP 模型确定某类特殊产品在某一特定时期的销售总量。

② 信息维。信息维是由计算字段建立的。假如用户想通过销售利润了解所有产品的

销售总额，进而分析是否可以通过增加销售来获得更多的利润。因为销售量大，可能因薄利多销，导致利润并不高；反之，若某产品利润率高，可能销量小却利润高。因此，可以就利润建立一个信息维（包括单品利润、总利润等属性），对销售总量建立一个度量，进而分析利润与销量的关系。

③ 分区维。分区维以同一结构生成两个或多个维时，这些维结构相同，只是数值不同。例如，对于时间维，每一年都有相同的季度、相同的月和相同的天（除了闰年以外）。假定把度量事实表分割为 2007 年的数据和 2008 年的数据，那么在 OLAP 分析中将频繁使用时间分区维来分割数据仓库中的数据。其中一个时间维是针对 2007 年的数据，而另一个时间维针对 2008 年的数据。

④ 分类维。分类维是通过对一个维的属性值分组而创建的。例如，客户表中有家庭收入属性，如果希望查看客户根据收入的购物方式，就可以生成一个含有家庭收入的分类维。

⑤ 退化维。当维表中的主键在事实表中没有与外键关联时，这样的维称为退化维。退化维与事实表并无关系，但有时在查询限制条件（如订单号码、出货单编号等）中需要用到退化维。以销售分析为例，通常是把出货日期作为事实时间，而把订单日期或需求日期等作为查询条件，这里，订单日期或需求日期就是退化维。

⑥ 一致维。当有多个数据集市要合并成一个企业级数据仓库时，可以使用一致维来集成数据集市，以便确保数据仓库可以使用每个数据集市的事实。

（3）维度表中维度的层次与级别

“维”一般包含着层次关系，如在时间维度上，按照“年-季-月”形成了一个层次，其中“年”“季”和“月”成为这个层次的三个级别。同样，在建立产品维度时，可以将“大类-子类-单品”划为一个层次，其中包含“大类”“子类”和“单品”三个级别，维度的层次在数据仓库中通常采用合并维分层结构和雪花分层结构两种实现方式。

① 合并维分层结构。合并维分层结构是将不同分层结构的信息对象完全合并到同一个维中。如产品维表可能就包含产品总类、产品类别、产品详细类别及产品名称等，合并维分层结构是星型模式的标准实现方法。其优点是查询简单，由于所有的分层结构都合并在同一维表中，因此不需要额外的表连接；其缺点是通常不符合第三范式，存在数据重复，需要较多的硬盘存储空间。

② 雪花分层结构。所有类别用规范化的独立表来存储数据。例如，将产品详细类别、产品类别及产品总类这三个分层结构分别独立成一个表，再用主键与外键来维持表间联系。雪花分层结构实际上是将星型模式进行规范化。其优点是因做过规范化，所以没有冗余数据，可能会节省硬盘空间；其缺点是查询需要作表连接，较麻烦。

9. 关于缓慢变化维的处理

维度可以根据其变化快慢分为无变化维度、缓慢变化维度和剧烈变化维度三类。例如，员工或客户的身份证号、姓名和性别等信息数据基本属于不变维范畴；政治面貌和婚姻状态属于缓慢变化维范畴；而工作经历、工作单位和培训经历等在某种程度上属于急剧变化维度。通常情况下，把其中不常变动的部分单独抽出来作为一个维表，按照缓慢变化维方式进行处理。对于维度的缓慢变化，可以根据不同的情况采取不同的方法来处理。

(1) 历史数据需要修改的情况。这种情况主要是发生在业务数据库中的数据出现错误,在分析过程中需要修改。处理办法是用直接覆盖法,即使用 UPDATE 语句来修改维度表中的数据。

(2) 新增数据维度成员改变了属性的情况。若某维度成员新加入了一列,该列在历史数据中无值,而在当前数据和将来数据中有值,且可以查询。解决方法是使用存储过程或程序生成新的维度属性,在后续的数据中可基于新属性进行查询。

(3) 历史数据保留,新增数据也要保留的情况。在这种需求下的解决方法是创建额外字段来记录这些数据之间的关系,例如在该维度打上时间戳,即将历史数据生效的时间段作为它的一个属性,在与原始匹配生成事实表时将按照时间段进行关联。这种方法的最大优点在于数据更改时,不需要创建额外的数据行,也不需要改变维表中的键值结构,因此可以在现有的数据行中查看所有历史记录。而最大的缺点是由时间点来判断更新的数据查询性能会降低,如果数据经常变化,则此方法并不适合。

10. 常用维度的设计模式

在数据仓库的逻辑模型设计中,有一些维度是经常使用的,它们的设计也形成了一定的设计模式和原则。

(1) 时间维度

时间维度是最常见的维度,数据仓库存储的是系统的历史数据,业务分析最基本的维度就是时间维度。时间维度通常包含年、季、月、星期和日 5 个层次,实际应用可能还会在月和星期之间增加旬层次,对日可能还会进一步分类,如节假日和工作日,以及周末和非周末。进行这些分类的目的是为了满足业务分析的需求。因为很多业务在周末节假日与正常工作日会有明显不同,分析业务在这个维度属性的变化会很有意义。

另一类型常见的时间维度是按照财年定义的时间维度,这在财务分析方面是必须使用的。

(2) 地理维度

地理维度如国家、区域和子区域等。地理维度的展示可与地理信息系统结合起来,使得最终用户能够得到更加直观的分析结果。

(3) 机构维度

机构维度是指实施项目的组织单位的内部组织机构的层次属性,机构维度有利于对企业各个部门或者各个分公司之间进行对比分析。

尽管一些企业的组织架构(子公司)是按照省市区域组织的,但机构维度同地理维度在本质上是不同的,地理维描述的是地理信息,而机构维描述的是企业的组织架构。

(4) 客户维度

企业总是要服务客户的,因此客户维度通常是必不可少的。分析客户背景信息对客户消费行为的影响,通过客户背景信息对客户群体进行合理分类都是企业市场策略分析的重要方面。常用的客户背景信息包括客户年龄、性别、婚姻状况、爱好和教育程度等。

3.2.5 数据仓库的物理模型设计

本节将从逻辑模型设计转向物理模型设计,数据仓库的物理模型设计基本遵循传统的数据库设计方法。事实上,数据仓库的物理模型就是数据仓库逻辑模型在物理系统中的实

现模式。其中包括了逻辑模型中各种实体表的具体化，例如表的数据结构类型、索引策略、数据存放位置和数据存储分配等。星型图中的指标实体和详细类别实体通常转变为具体的物理数据库表，而维度实体则可能作为查询参考、过滤和聚合数据使用，不一定直接转变为物理数据库表。

1. 物理模型设计的主要工作

在进行物理模型的设计时，需要考虑的因素有 I/O 存取时间、空间利用率及维护成本等。在物理模型设计阶段，需要完成以下工作。

(1) 定义数据标准，规范化数据仓库中的数据。

(2) 选择数据库架构(关系数据库的星型模式、多维数据库的 Cube)及其具体的数据库管理系统软件和版本等。

(3) 根据具体使用的数据库管理系统，将实体和实体特征物理化，具体包括如下内容。

① 字段设计。如字段数据类型选择、数据完整性(字段的物理结构)控制等。

② 物理记录设计(物理存取块与物理记录的处理)。主要解决存储空间的有效利用问题。

③ 反向规范化。根据需要，用来提高数据的查询性能。

④ 分区。

(4) 数据容量和使用频率分析，以定义规模，确定数据容量、响应时间要求和更新频率等。确定外部存储设备等物理环境。

(5) 物理文件的设计。指针、文件组织和簇文件。

(6) 索引的使用与选择。

(7) RAID。

2. 物理存储结构设计的原则

在物理设计时，常常要将数据按其重要性、使用频率及响应时间要求进行分类，将不同类型的数据分别存储在不同的存储设备中。重要性高、经常存取并对响应时间要求高的数据存放在高速存储设备上；存取频率低或对存取响应时间要求低的数据则可以存放在低速存储设备上。另外，在设计时还要考虑数据在特定存储介质上的布局。存储结构设计原则包括如下几点。

(1) 不要把经常需要连接的几张表放在同一存储设备上，这样可以利用存储设备的并行操作功能加快数据查询的速度。

(2) 建议把整个组织共享的细节数据放在一个集中式服务器(或集群)上，以提高这些共享数据的访问性能。

(3) 建议将数据库表和索引分放在不同物理存储设备上，一般可以将索引存放在高速存储设备上，而将表存放在一般存储设备上，以加快数据的查询速度。

(4) 建议在系统中使用廉价冗余磁盘阵列(Redundant Array of Inexpensive Disk，RAID)。

3. 数据仓库索引设计的特殊性

数据仓库的数据量通常较大，且数据一般很少更新，所以可以通过设计和优化索引结构来提高数据存取性能。数据仓库中的表通常要比联机事务处理系统中的表建立更多的索

引,表中应用的最大索引数应与表的数据量规模成正比,设计人员甚至可以考虑对部分数据表建立专用索引和复杂索引,以获取较高的存取性能。数据仓库是个只读的环境,建立索引可以取得灵活性,对性能极为有利。但是,表有很多索引,那么数据加载时间就会延长,因此数据仓库索引的建立也需要综合考虑。在建立索引时,可以按照索引使用的频率由高到低逐步添加,直到某一索引加入后,使数据加载或重组表的时间过长时,就结束索引的添加。

最初,一般都是按主键和外键建立索引。如果表数据量过大,则可能需要另外增加索引。如果一个表中所有用到的列都在索引文件中,就不必访问事实表,只要访问索引就可以达到访问数据的目的,以此来减少 I/O 操作。如果表太大,并且经常要对它进行长时间的扫描,那么还可以考虑添加一张概要表以减少数据的扫描任务。

4. 存储优化与存储策略

确定数据的存储结构和表的索引结构后,需要进一步确定数据的存储位置和存储策略,以提高系统的 I/O 效率。下面介绍几种常见的存储优化方法。

(1) 表的归并与簇文件(clustering files)。当多个表的记录分散存放在几个物理块中时,这些表的存取和连接操作的代价会很大。这时可以将需要同时访问的表在物理上顺序存放,或者直接通过公共键将相互关联的记录放在一起。即簇文件设计模式,这种设计模式通常在访问序列经常出现或者表之间具有很强的访问相关性时有较好的性能效果,对于很少出现的访问序列和没有强相关性的表,使用表的归并时没有效果。

(2) 反向规范化,引入冗余。一些表的某些属性可能在对多个其他表的查询时经常用到,则可考虑将这些属性复制到多个主题相关表中,可以减少查询时连接表的个数。另外,根据需要可以存储导出数据属性,即在原始数据的基础上进行总结或计算,生成导出数据(derived data)并作为冗余列存储在表中,以便在应用中直接使用这些导出数据,免去计算或汇总过程。

(3) 表的物理分割(分区)。每个主题中的各个属性或记录的存取频率是不同的。将一张表按各属性或记录的存取频率分成多张表,将具有相似访问频率或访问相关性强的数据组织在一起。

以上完成了数据仓库从概念模型到物理模型的整个设计过程,但事实上,数据仓库系统的设计过程不是这样完全分离的三部曲,而是一个动态、循环和反馈的过程,数据仓库的设计过程实际上是一个螺旋式动态变化的过程。其中,数据仓库的数据内容、结构、粒度、分割与维度等的设计需要在不断与用户沟通、听取用户反馈信息的基础上不断地调整与完善,只有不断启发与理解用户的分析需求,才能挖掘出用户的潜在真实需求,向用户提供更准确和更有用的决策信息。

在完成数据模型的设计之后,就可以在具体的数据库管理系统(如 SQL Server 2005)中根据设计的数据模型建立数据仓库数据库,其表结构及表间关系参见 SQL Server 自带的示例数据库(Adventure Works DW)。

3.3 使用 SQL Server 2005 建立多维数据模型

在完成数据模型设计并据此建立数据仓库数据库之后,就可以导出数据,建立多维数据模型(也叫 Cube),形成针对某主题的数据集市,进行 OLAP 分析了。这一节将通过示例演

示建立多维数据模型，一步一步地展示数据的导出过程。

3.3.1　SQL Server 2005 示例数据仓库环境的配置与使用

1. 本节的示例操作需要在 SQL Server 2005 数据库环境中安装下列组件、示例和工具

(1) Microsoft SQL Server 2005 Database Engine。

(2) Microsoft SQL Server 2005 Analysis Services。注意，只有标准版、企业版和开发版有此功能，工作组版及 Express 免费版没有此功能。

(3) Business Intelligence Development Studio。

(4) Adventure Works DW 示例数据库。

(5) Analysis Services 教程示例项目。

2. 权限方面

必须是 Analysis Services 计算机上本地管理员组(Administrators)的成员或 Analysis Services 实例中的服务器角色的成员。如果需要使用示例数据库，必须拥有对 SQL Server 2005 Adventure Works DW 数据库的读取权限。

3. 如何安装示例数据库

如果在初始安装 SQL Server 2005 过程中没有安装示例数据库或示例，可以通过如下步骤安装示例数据库。

(1) 安装并附加示例数据库。

从“添加或删除程序”中选择 Microsoft SQL Server 2005，然后单击“更改”按钮。按照 Microsoft SQL Server 维护向导中的步骤操作。

从“选择组件”中选择“工作站组件”，然后单击“下一步”按钮。

从“更改或删除实例”中单击“更改已安装的组件”。

在“功能选择”对话框中，展开“联机丛书和示例”结点。

选择“示例”，展开“数据库”结点，然后选择要安装的示例数据库。单击“下一步”按钮(如图 3.15 所示)。

若要安装并附加示例数据库，则从“安装示例数据库”中选择“附加示例数据库”，然后单击“下一步”按钮。数据库文件创建并存储在文件夹 n:\Program Files\Microsoft SQL Server\MSSQL.n\MSSQL\Data 中。数据库已附加并可以使用。

若要安装示例数据库文件但不附加，则在“安装示例数据库”中选择“安装示例数据库”，然后单击“下一步”按钮。Adventure Works 数据库文件创建在文件夹 n:\Program Files\Microsoft SQL Server\90\Tools\Samples\Adventure Works OLTP 中。Adventure Works DW 文件创建在 n:\Program Files\Microsoft SQL Server\90\Tools \Samples\Adventure Works Data Warehouse 文件夹中。必须先附加数据库，然后才能使用它。有关详细信息，请参阅分离和附加数据库。

(2) 安装示例程序源代码。

在示例数据库安装完成后，再执行以下方法之一来安装示例程序代码。

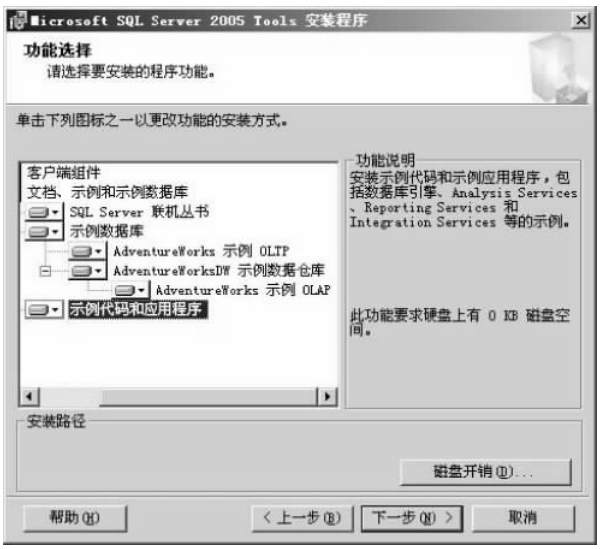

图 3.15　示例数据库安装界面

① 选择“开始”→“所有程序”→Microsoft SQL Server 2005 命令，单击“文档和教程”，然后单击“示例”，再单击“Microsoft SQL Server 2005 示例”。

② 使用 Windows 资源管理器，定位到 n:\Program Files\Microsoft SQL Server\90\Tools\Samples\，然后双击 SqlServerSamples.msi 启动安装程序(如图 3.16 所示)。

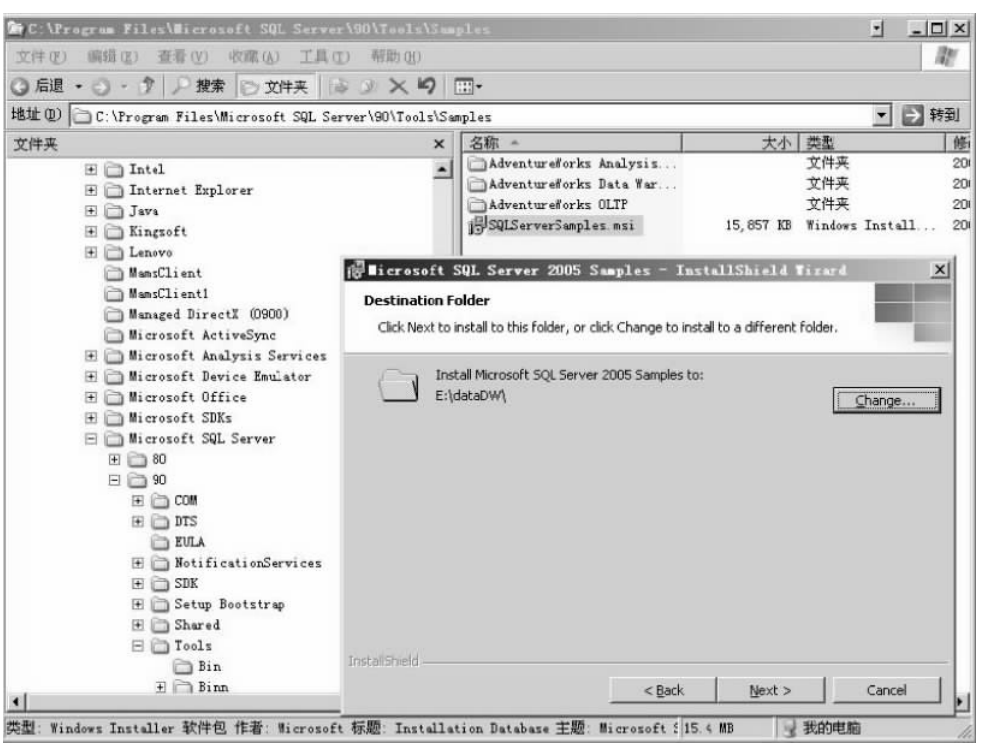

图 3.16　示例程序代码安装

(3) 部署 Adventure Works DW Analysis Services(已经建好的一个示例分析项目),测试数据仓库环境的配置是否正确。

首先确保已经安装了 Adventure Works 和 Adventure Works DW 示例数据库;确保已经安装了 Analysis Services。

在 SQL Server Business Intelligence Development Studio 工具栏中选择“文件”→“打开”命令,然后单击“项目/解决方案”。浏览到 n:\Program Files\Microsoft SQL Server\90\Tools\Samples\Adventure Works Analysis Services Project,选择文件 Adventure Works. sln,然后单击“打开”按钮。

在解决方案资源管理器中右击 Adventure Works DW,在弹出的快捷菜单中选择“部署”命令。部署成功即说明配置正确。

4. 利用示例数据仓库(Adventure Works DW)环境及帮助系统学习

选择“开始”→“所有程序”→ Microsoft SQL Server 2005 命令,单击“文档和教程”项,然后单击“教程”按钮,再单击“SQL Server 教程”项。

进入帮助系统后,单击左边目录树结点“SQL Server 2005 教程”→“SQL Server 2005 Analysis Services 教程”,下面共有 10 课,前三课只是定义一个多维数据集,可以根据自己的学习要求选择是否练习。各课学习内容如下。

第 1 课:在 Analysis Services 项目中定义数据源视图。使用 BI Development Studio 在 Analysis Services 项目中定义一个数据源视图。

第 2 课:定义和部署多维数据集。使用多维数据集向导定义一个多维数据集及其维度,然后将该多维数据集部署到 Analysis Services 的本地实例中。

第 3 课:修改度量值、属性和层次结构。改进多维数据集的用户友好特性,并逐渐增加对相关更改的部署,根据需要处理多维数据集及其维度。

第 4 课:定义高级属性和维度属性。使用组合键来定义引用维度关系以及为属性成员排序,并定义自定义错误处理。

第 5 课:定义维度和度量值组之间的关系。为退化维度定义一个事实关系,并定义一个多对多关系。

第 6 课:定义计算。定义计算成员、命名集和脚本。

第 7 课:定义关键性能指标。

第 8 课:定义操作。

第 9 课:定义透视和翻译。定义多维数据集的视图以及元数据的翻译。

第 10 课:定义管理角色。定义管理角色和用户角色。

3.3.2　基于 SQL Server 2005 示例数据库的多维数据模型

本节将针对 Adventure Works Cycle 公司的销售分析需求,从 Adventure Works DW 示例数据库中导出数据,建立并部署“销售分析”多维数据集,进而从多角度对 Adventure Works Cycle 公司的销售状况作分析研究。

1. 创建一个新的数据仓库分析项目

打开 Visual Studio 2005 新建项目,选择 Analysis Services 项目,并将项目名称更改为“销售分析示例”(如图 3.17 所示)。

图 3.17　新建分析项目

具体操作路径:选择“开始”→“所有程序”→Microsoft SQL Server 2005 命令,再单击 SQL Server Business Intelligence Development Studio,将打开 Microsoft Visual Studio 2005 开发环境。

2. 定义数据源

在“数据源”文件夹上右击,在弹出的快捷菜单中选择“新建数据源”命令(如图 3.18 所示)。

图 3.18　选择“新建数据源”命令

启动新建数据源向导，单击图 3.19 中的“新建”按钮。

图 3.19　选择如何定义连接

出现“连接管理器”对话框（如图 3.20 所示），在“提供程序”下拉列表框中确保已选中“本机 OLE DB\ Microsoft OLE DB Provider for SQL Server”选项。选择数据源账号为“使用服务账户”并命名数据源为“销售分析数据源”。

图 3.20　选择要连接的数据库

3. 定义数据源视图

选择“数据源视图”文件夹,新建一个数据源视图(如图 3.21 所示)。数据源选择上一步新建的“销售分析数据源”。在“可用对象”列表框中,选择下列表(同时按住 Ctrl 键可选择多个表)。

图 3.21 定义数据源视图

(1) DimCustomer(客户维表)。

(2) DimGeography(地理维表)。

(3) DimProduct(产品维表)。

(4) DimTime(时间维表)。

(5) FactInternetSales(网上销售事实表)。

在定义好的数据源视图中可以修改表的默认名称(右击要改名的表,从弹出的快捷菜单中选择“属性”命令),让视图更易理解,如图 3.22 所示。

4. 定义多维数据集

右击“多维数据集”,从弹出的快捷菜单中选择“新建多维数据集”命令;已选中“使用数据源生成多维数据集”选项和“自动生成”选项;在“时间维度表”下拉列表中选择“时间”别名(如图 3.23 所示)。

下一步设置时间维,将时间属性名称映射到已指定为“时间”维度的维度表中的相应列,如图 3.24 所示。

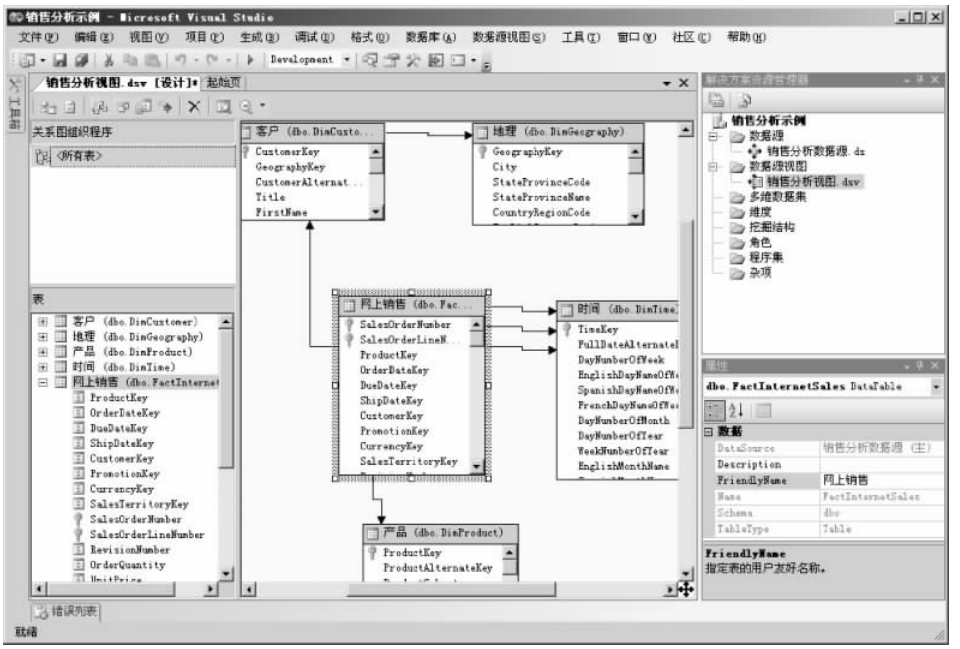

图 3.22　数据源视图(包括星型图)

图 3.23　标识事实数据表和维度表界面

图 3.24　选择时间段

选择事实表的度量值(去掉不是度量值的列),可以对度量值重新命名,如图 3.25 所示。

图 3.25　选择度量值

设置和校验维度的属性及层次结构,在“查看新建维度”页上,通过展开树控件显示该向导检测到的三个维度的层次结构和属性,查看其中每个维度的维度层次结构(可根据需要去掉部分维度属性),如图 3.26 所示。

图 3.26　设置和校验维度属性

在“完成向导”页上，将此多维数据集的名称更改为“销售分析多维数据集”，单击“完成”按钮，便完成了多维数据集的定义，此时仍可以对维度或度量等名称做更改，以便最终用户理解与使用。图 3.27 显示了多维数据集中的维度表和事实数据表(事实数据表是黄色的，维度表是蓝色的)。

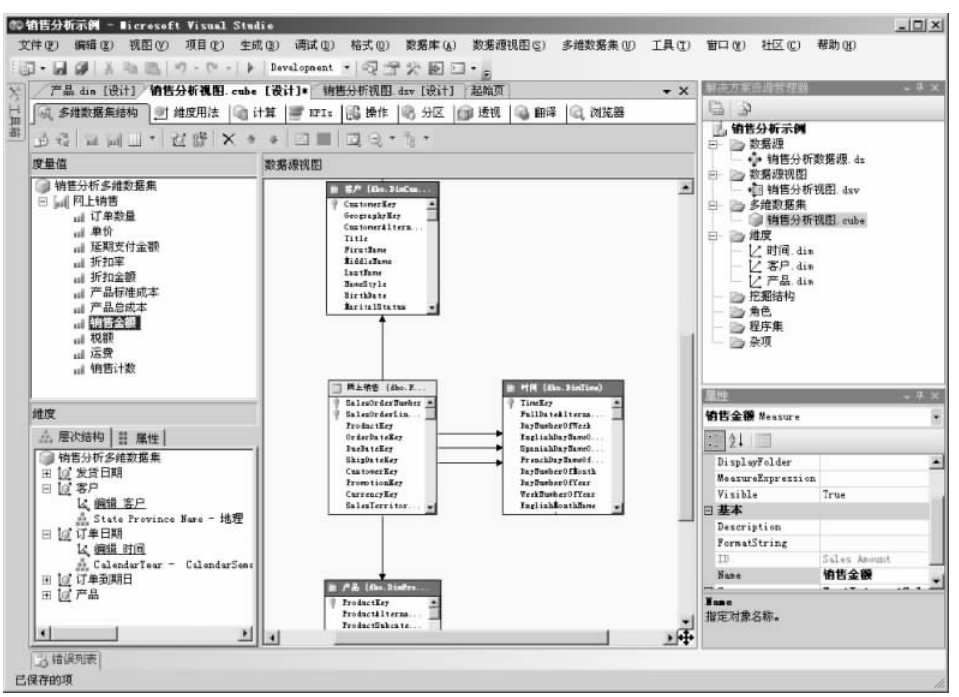

图 3.27　多维数据集的结构

在维度设计器的“维度结构”选项卡上,可以添加、删除和编辑层次结构、级别和属性,如图 3.28 所示。

图 3.28 订单日期时间维度的编辑

5. 部署“销售分析示例”项目

若要查看刚才建立的销售分析多维数据集中的数据,必须将其所在的项目部署到分析服务(Analysis Services)的指定实例,然后可以处理多维数据集及其维度。

(1) 部署配置。

在解决方案资源管理器中,右击根结点“销售分析示例”项目,从弹出的快捷菜单中选择“属性”命令。在弹出的对话框中更改“数据库”对应值为 Analysis Services,如图 3.29 所示。

图 3.29 项目部署的配置属性

（2）部署项目。

在解决方案资源管理器中，右击“销售分析示例”项目，从弹出的快捷菜单中选择“部署”命令，或者在菜单栏上选择“生成”菜单，单击“部署 销售分析示例”。

若服务器没有安装 Analysis Services 或没启动数据库服务器，将报错“无法建立连接”，进而部署失败。

查看“输出”窗口和“部署进度 - 销售分析示例”窗口的内容，验证是否已生成、部署完成多维数据集，没有出现错误，且在右下角显示“部署成功完成”即表示部署成功，如图 3.30 所示。

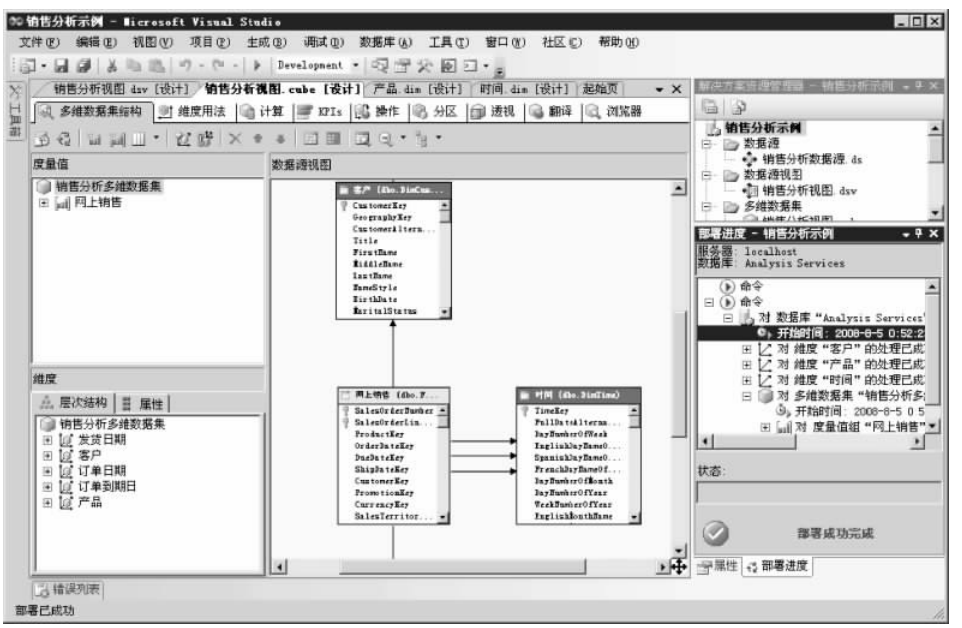

图 3.30　项目部署信息

6. 浏览已部署的多维数据集

部署完成后，就可以浏览多维数据集中的实际数据了。浏览“销售分析示例”多维数据集及其每个维度，以确定为了改进此多维数据集的功能而需要执行的更改。

在解决方案中单击“客户”维度，然后选择“浏览器”选项卡。

在这里，可以从各个角度（如国家、省/州、姓氏甚至电话号码等）浏览客户结构，现在有关“客户”级别的信息只显示客户的电子邮件地址，而不显示客户的姓名，需要通过后面的更改显示客户姓名。按省/州分类浏览客户如图 3.31 所示。

单击在解决方案的“多维数据集”目录下的子项“销售分析图.cube”，切换到“浏览器”选项卡，内容区分为三个窗口：左边窗口显示事实表和维度表的元数据信息，右上窗口为维度筛选器，右下窗口为报表数据显示窗口，如图 3.32 所示。

浏览多维数据集的操作方法：从元数据窗口拖动有关内容到右边数据显示区或筛选器中即可形成一个初步的报表，显然还很粗糙，特别是显示格式等有待在后续的操作中改进。

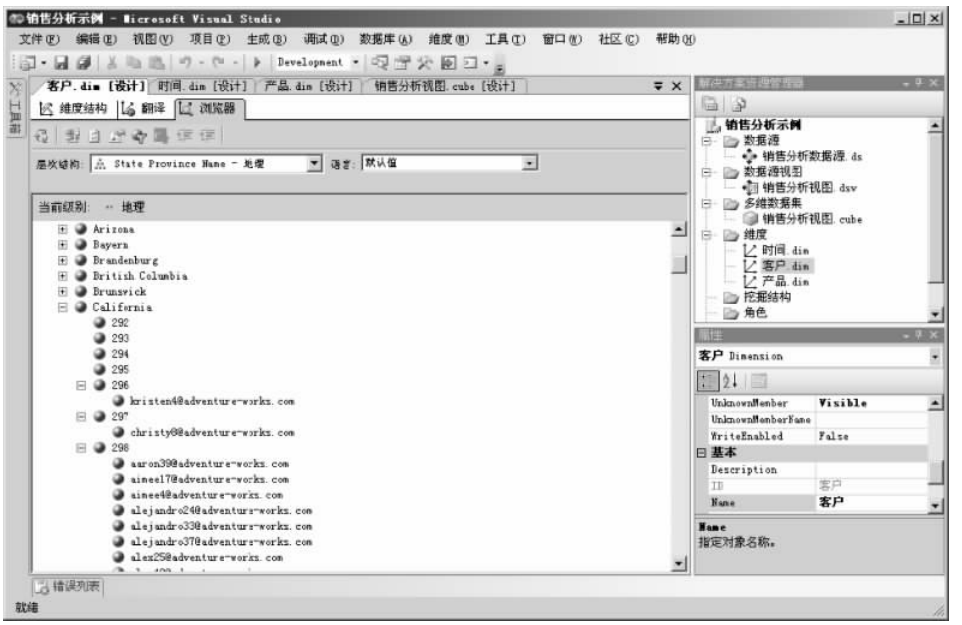

图 3.31　按省/州分类浏览客户

图 3.32　浏览多维数据集界面

图 3.32 所示的展示数据的操作步骤如下。

(1) 将事实表中的“销售额”度量值拖到数据显示区的“将合计或详细信息字段拖至此处”区域。

(2) 将客户维度表的“英语国家/地区区域名”属性层次结构拖到数据显示区的“将行字段拖至此处”区域。

(3) 将产品维度表的“产品系列(Product Line)”拖到数据显示区的“将列字段拖至此处”区域；或者右击“产品系列”,从弹出的快捷菜单中选择“添加到列区域”命令。

(4) 将“订单日期”维度的“季度”拖到数据显示区的“将筛选器字段拖至此处”区域,并单击“季度”下拉框,不选第 4 季度(即报表只需要 1～3 季度的数据)。

(5) 右击“订单日期”维度的“年度”属性层次结构中的 2002 成员，然后单击“添加到子多维数据集区域”。再单击“筛选表达式”下单元格的下拉框，复选 2003 和 2004 年度，即将选择 2002、2003 和 2004 这三年的数据作报表。

7. 提高多维数据集的可用性和易用性

(1) 修改度量值的有关属性。例如，将多维数据集中的货币和百分比度量值指定格式设置属性。在“多维数据集结构”选项卡中修改度量值属性后，需要在“浏览器”选项卡中单击“重新连接”，才能看到更新后的格式，如图 3.33 所示。

图 3.33　修改度量值的显示格式为 999999.00

(2) 修改维度的层次结构和有关属性。例如，修改“客户”维度的层次结构、删除没有用的属性等。切换到“客户”维度的维度设计器，并选择“维度结构”选项卡，删除不使用的属性，将剩下的属性改为汉字名，以便于最终用户的理解与使用，如图 3.34 所示。

图 3.34　修改客户维度的属性(改名或删除等操作)

(3) 增加维度属性并对维度属性分组。例如,在“客户”维度的“维度结构”选项卡上,将 EmailAddress 列从“数据源视图”窗格的 Customer 表拖动到“特性”窗格中(添加电子邮件属性);通过按住 Ctrl 键选择电子邮件和电话两个属性,然后将其 AttributeHierarchyDisplayFolder 设置为“联系人”(实现维度属性分组),如图 3.35 所示。

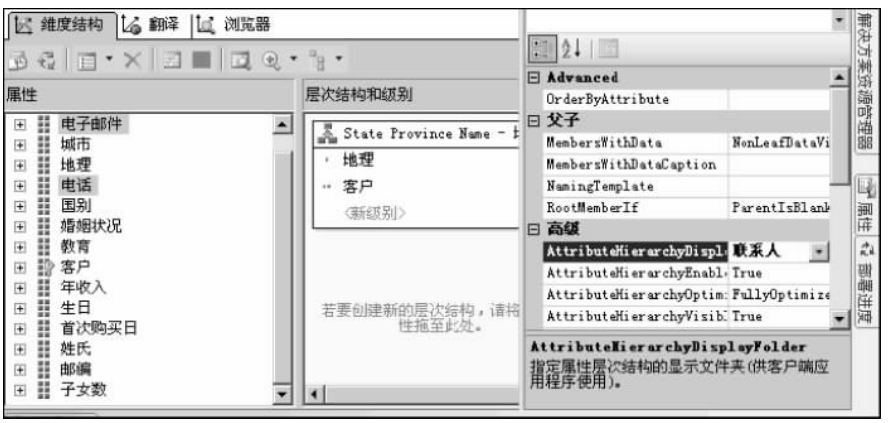

图 3.35　给客户维度增加电子邮件属性并分组

(4) 将命名计算列作为维度的成员名称。编辑数据源视图的客户表,右击 Customer,从弹出的快捷菜单中选择“新建命名计算”命令,增加计算列“全名”,如图 3.36 所示。并将“客户”级别的每个成员名称(客户主键)改为客户的“全名”(原来默认为客户的电子邮件地址),如图 3.37 所示。

图 3.36　增加计算列“全名”

(5) 重新部署并查看更改。在 BI Development Studio 的“生成”菜单上,单击“部署 销售分析示例”。部署完成后,再打开客户维度的浏览器,可以发现显示的客户标识内容已不再是电子邮件地址,而是客户的全名(First Name+Middle Name+Last Name),如图 3.38 所示。

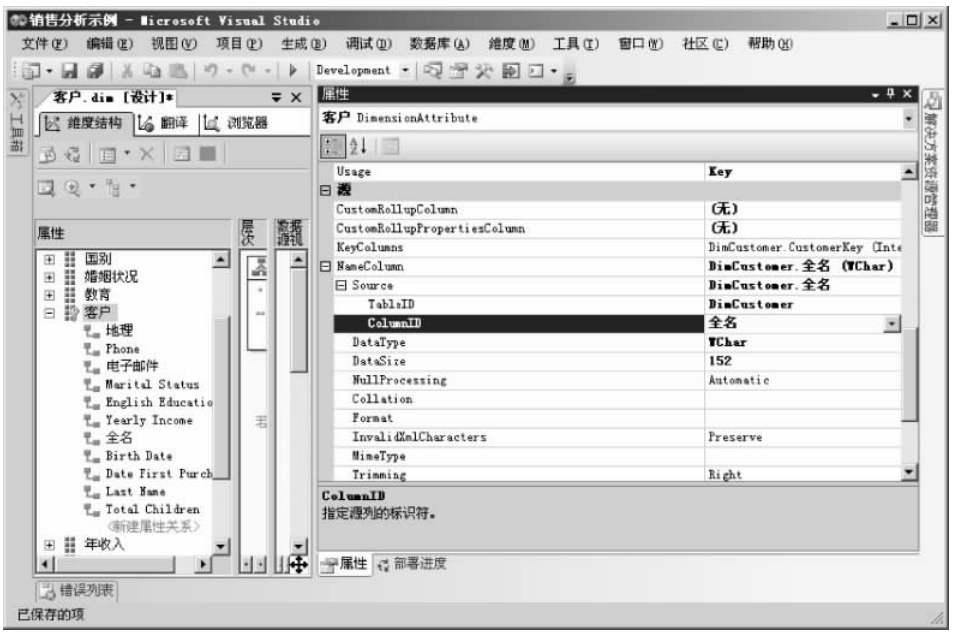

图 3.37　更改客户主键的 Column ID 为“全名”列

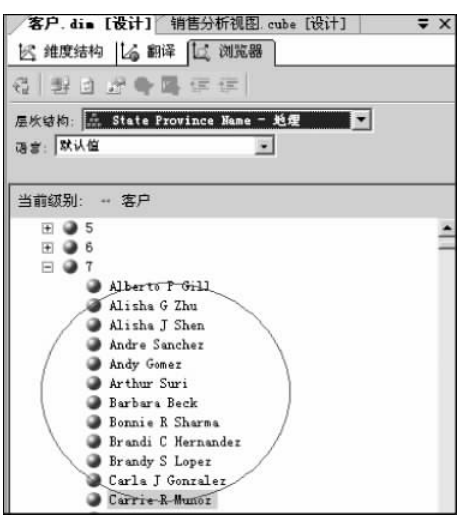

图 3.38　客户的标识内容(Column ID)已由电子邮件改为全名

(6) 灵活快速地导出各类统计报表。在多维数据集建立完成后,就可以快速灵活地导出针对该主题的各类统计报表,通常只需将左边多维数据集窗口中的度量值或维度属性根据需要拖曳到右边的数据显示区或筛选器中,即可设置好度量指标、筛选条件和分组条件,产生满足各种需要的报表。

如图 3.39 所示,将订单数量作为度量指标,将月份作为行字段分组条件,将国别作为列字段分组条件,将家庭子女数作为筛选字段,并在筛选器中添加一个“客户”维度的筛选条件,即要求客户年收入大于等于 10 万元。设置好这些内容后,便可立即得到一个统计报表,

即按国别、月份分组统计年收入不低于 10 万元的独生子女家庭的订单数量的分布情况表。

图 3.39　按国别、月份分组统计年收入不低于 10 万元的独生子女家庭的订单数分布情况

3.4　小结

本章主要介绍了数据仓库的设计与开发过程。

建立一个数据仓库系统通常需要经历收集与分析业务需求、建立数据仓库的概念和逻辑模型、对数据仓库作物理设计、定义数据源、选择数据仓库技术与平台、数据的 ETL 处理、选择数据分析与数据展示软件、数据仓库的更新设计等步骤。

数据仓库应用系统的开发包括两个主要部分,一是数据仓库数据库的开发与设计,用于存储数据仓库的数据;二是数据分析应用系统的开发。可使用信息包图法、运用信息包图法进行概念模型设计;利用星型图进行数据仓库的逻辑模型设计。

最后介绍了使用 SQL Server 2005 建立数据仓库首先要配置环境,然后建立多维数据模型。

3.5　习题

1. SQL Server SSAS 提供了所有业务数据的统一整合视图,可以作为传统报表、_____、关键性能指示器记分卡和数据挖掘的基础。

2. 数据仓库的概念模型通常采用_____来进行设计,要求将其 5 个组成部分(包括名称、_____、_____、层次和_____)全面地描述出来。

3. 数据仓库的_____通常采用星型图法来进行设计,要求将星型图的各类逻辑实体

完整地描述出来。

4. 按照事实表中度量的可加性情况，可以把事实表对应的事实分为 4 种类型：_____、_____、_____和事件事实。

5. 确定了数据仓库的粒度模型以后，为提高数据仓库的使用性能，还需要根据用户需求设计_____。

6. 在项目实施时，根据事实表的特点和用户的查询需求，可以选用_____、业务类型、_____和下属组织等多种数据分割类型。

7. 当维表中的主键在事实表中没有与外键关联时，这样的维称为_____。它与事实表并无关系，但有时在查询限制条件(如订单号码、出货单编号等)中需要用到。

8. 维度可以根据其变化快慢分为_____维度、_____维度和_____维度三类。

9. 数据仓库的数据量通常较大，且数据一般很少更新，可以通过设计和优化_____结构来提高数据存取性能。

10. 数据仓库数据库常见的存储优化方法包括表的归并与簇文件、_____、表的物理分割(分区)。

11. 什么是信息包图法？它为什么适用于数据仓库的概念模型的设计？

12. 简述数据仓库系统设计过程。

13. 一个数据仓库系统的建立通常需要经过哪些步骤？

14. 运行 SQL Server 的 Adventure Works DW 示例数据库，建立多维数据模型练习。