# 使用模式

回顾这些模式之后，你很可能正打算问，“我该如何在企业中使用、应用和/或实现这些模式呢？”这个问题很难回答，因为它的答案通常都会包含“这取决于……”这样的词汇。即使开发原型系统数据模型的人和开发数据仓库的人问的是同一问题，但是他们得到的答案会是不同的。即使这两个人在开发数据模型时，都以不同的方式使用了这些模式，但是由于他们对于什么是数据模型具有不同的观点或他们所面对的受众不同！在前面的每一章节中，我们都展现了不同泛化程度的模式的优缺点，让模式的用户可以根据具体的情况决定使用哪种模式。例如，创建概念数据模型的人并且他需要将这个模型展示给业务人员看，那么他可能会选择一级模式，因为那些模式的优点适合该情形。创建数据仓库数据模型的人需要灵活的设计，他可能会应用三级模式，因为该层级模式的优点满足其特定的需求。例如，这种模式一般能够满足数据仓库源系统的各种需求【批注1】。

这章节要展示的是如何使用前面章节所介绍的模式来创建满足不同需求的数据模型想象这个流程的一个好方法是将你看到的不同层级的模式想象成可互换组件。在这一章节，我们将使用所有的组件模式并将他们联合起来创建数据模型，这些数据模型可以解决大多数都需要解决的常见的数据建模挑战。

## 本章节有哪些内容？

首先，我们介绍一个场景，这场景将会被不断的重复，并贯穿于整个章节。这个场景随着每节中情形的变更而做相应的改进。然后在本章节的余下部分，我打算处理那些我们认为可以使用这些模式创建模型的最常见的情形。这些模型包括：

* **原型数据模型和作用域数据模型【批注2】。**这些模式被用于原型，被作为作用域声明的一部份，并且这些模式若用在其他情形中，需要说明信息需求。这些模型是使用这些模式开发非常特定的数据模型的示例，一般情况下，他们被用来收集验证面向业务的受众的数据需求。
* **应用数据模型。**当需要开发应用系统时，可以使用这些模式为该应用开发一个数据模型。
* **企业数据模型。**这一节满足了下面这样的需求：使用这些模式开发反应整个企业需求的数据模型。此外，我们展现了特定的企业数据建模方案，为同类型的数据建模提供了备选数据模型结构。因此，有些数据需求的建模即使用了具体的模式也使用了泛化的模式，使用企业数据模型为各个应用提供了一个选择。
* **数据仓库数据模型。**这一节被分为两部分，介绍了如何使用这些模式开发两种不同类型的数据仓库建模方案。分别是：
* **关系型数据仓库模型。**这些模式被用来启动面向关系数据仓库数据模型的开发。这个模型可能会被用来作为数据仓库或操作型数据存储设计的基础。
* **星型模式数据仓库模型。**可以使用这些模式帮助理解数据，可使用他们揭露星型模式有待解决的复杂性。此外，还可以使用这些模式启动维度表和事实表的设计。
* **主数据管理（MDM）数据模型。**这些模型展示了在捕获带有严谨的规则（关于如何使用和引用数据的规则）的主数据和应用数据时，如何使用这些模式使用所需的灵活性。

**注：在本章节我们没有研究每种模式的细节，因为我们在其他章节中介绍了这样的内容。这样做的目的是展示我们是如何将这些模式整合在一起开发出适合某一用途的模型，并解释了为什么我们在不同的情形中选择使用某一层级的模式。**

## 场景

为了阐释我们选择使用哪个层级的模式的思维流程，我们创建了一个场景，这些场景会随着每类模型需求的变更而做相应的改进。示例展示了我们会使用哪些模型以及我们是如何将这些模式用于不同类型的工作上，比如说，用于原型、应用、企业数据模型、企业数据仓库关系型模型、数据仓库星型模式和主数据管理模型。我们会这这些小节中解释我们的方案，为什么我选择使用这种模式多于使用另外一种模式，为什么我们将不同层级的模式混合在一起。

试想下，一家坐落于酋长国—迪拜被称为金沙分销商的大型的石化行业设备经销商。在过去的几年里，随着石油收益和中东地区乃至全世界设备需求的增加，这家公司取得了令人瞩目的成长。这家公司的主要数据问题源自于它的成长。它兼并了各个国家中的小型分销公司，这些小公司都有自己的数据。该企业意识到数据是它最大的资产之一，但是它该如何充分利用这一资产呢？它希望使用一个一致的数据策略来管理集成它所有的数据，包括源自被兼并公司的各种数据格式。

首席执行官的愿景包括将这些金沙分销商发展成以顾客为中心的商业模式。要做到这一点，企业需要采取更好的方式来管理顾客数据，但是它不确定该如何达到这一目标。它不确定是否自己目前的顾客是否已经与这些新吞并的公司有业务往来。它很难发觉顾客的趋向、很难预估丢失顾客的风险甚至不知道自己又多少顾客，因为它没有集成的顾客数据。处于这些情况下，该企业雇佣了一组数据专业人士来帮助管理有价值的顾客数据。

该企业想要维护的顾客数据有：

* 顾客的名称和联系信息，比如说电话号码、邮政地址和邮件地址。

顾客的状态，例如，顾客可以处于激活状态、未激活状态、信用审核通过和调查进行中。

* 顾客的层级划分和组织的结构 — 也就是他们的哪些顾客是属于总公司的而哪些顾客是属于子公司和分支机构的。
* 一般情况下当事人所扮演的不同角色，例如，顾客、销售人员和合作伙伴。
* 顾客在重要交易的上下文中所扮演的特定角色，比如说订单的付款顾客或运达顾客【批注3】。
* 顾客的不同分类，比如说顾客类型、顾客规模、行业类型等等。

鉴于此，那就让我们从原型模型和作用域声明开始吧，将他们作为我们场景的第一部分。

## 原型模型、作用域声明

正如我们在整本书中所论述的那样，这本书中的模式具有不同层次的泛化。你可以在数据模型中混合使用这些模式来满足你具体的需求。因此，我们建议使用适合的模式，也就是说它们适合于这种工作和相关的情形。

在这一节，我们介绍了如何使用不同的模式来阐释作用域以及展示概念、业务数据需求和数据问题。然后我们将展示如何使用这些模式创建一个简单的适用于原型的数据模型。这一节所介绍的模式在捕获常见的核心术语时，还可以作为建立夸功能边界交流基线的一个起点。

## 这些模型的场景

在前一节我们向你介绍了金沙分销商。正如你所见，该企业雇佣了一组数据专业人士帮助他制定数据策略，该策略反应了它期望成为一家以顾客为中心的公司。作为此工作的一部分，数据专业人士建议为捡来的数据需求制作一份策略声明。他们为金沙经销商制作了顾客数据作用域声明 ，将该声明作为策略声明的一部分。但是，他们还想创建一个原型集，包括报告、屏幕和分析例程【批注5】，用来说明金沙经销商在将来可以位于何处。高级数据专业人士认为这些可行的原型是一种强有力的方法，可以说服高级管理层关于集成数据顾客数据的好处，还可以作为确保所需的数据都被捕获了。考虑到这一点，数据专业人士制作了表9-1，作为初始的作用域声明。

（图）

**图9-1** 在作用域声明和理解数据需求方面所使用的数据模型

## 这些模型是如何运作的？

你可以看出图9-1是非常特定的数据模型，它主要由一级模式构成。**顾客**实体展示了应用一级声明角色模式的示例，在一级声明角色模式中，你可以减实体**顾客**作为一个单独实体（独立于其他角色），该实体中可以将顾客名称作为扮演顾客这一角色的人员或组织的一个属性。可以将声明角色定义为“指派给某人或某个组织的行为活动或者是某个人或组织所需要的行为活动”。需要注意的是我们称该声明角色为顾客（CUSTOMER）而不是客户（CLIENT）或其他名称。这个很重要的一点，尽管看起不是很重要，但是这对捕获声明角色的恰当术语是非常重要的。金沙经销商吞并的一家公司可能会使用术语客户（CLIENT）或合作伙伴（PARTNER）。他们与顾客（CUSTOMER）相同吗？金沙经销商决定随任何人员或组织都使用实体名称顾客（CUSTOMER）包括“从企业购买产品、收到产品和使用产品的人或组织（此处的产品既可以是商品也可以是服务），组织或人员都可以扮演顾客这一角色”。公司总部实体是将一级声明角色模式应用于公司的另外一个示例，它扮演的是较高级别的联盟组织角色，在某种程度上来说，该组织掌控了客户的所有活动。

**注：你可以捕获其他的术语比如说客户（CLIENT），这个词或许是顾客（CUSTOMER）的同义词，他们也可能具有不同的意思，会有自己的声明角色实体。**

在图9-1中你可以看到源自第7章节的一级联系机制模式。在这个模型中，这个模式阐释了各种属性，这些属性捕获了联系顾客的不同方式。在这张图中你可以付款街道地址成分、付款城市、工作电话号码、工作邮件地址等等。除了捕获了精确的联系机制详情外，比如所电话号码，模型还捕获了联系机制类型和联系机制的目的用途，比如说付款地址成分（目的和联系机制类型）或个人电话号码（用途和联系机制类型）。第一，他们洗完捕获联系**顾客**（CUSTOMER）的所有可能的方式，比如说邮政地址、电话号码和电子邮件地址。第二，他们还需要捕获不同联系机制的类型、目的或用途。图9-1还捕获了该顾客是处于激活状态还是未激活状态。

如果顾客在过去的两年内完成了与金沙经销商的任何交易，那么可以认为这个客户处于激活状态。这个信息的捕获使用了一级状态模式，将其作为**顾客**（CUSTOMER）的属性：激活起始日期（**active from date**）、激活终止日期（**active thru date**）。一个顾客可能还有更多状态，但是数据专业人士尚未确定它们。

可以使用不同的方式对顾客（CUSTOMER）的实例进行分类。起初金沙经销商想通过顾客规模（**customer size**）、顾客类型（**customer type**）和顾客行业（**customer industry**）这些属性来捕获**顾客**。这些分类是金沙经销商报告和分析的‘基本依据’。这些报告被分解或者是向上钻取这些核心分类集。金沙经销商表明他希望捕获客户的组织结构。每个客户都是大型企业的一部分。为了支持这一需求，数据专业人士创建了总公司（PARENT COMPANY）来捕获他们客户的组织结构。总公司实体捕获了总公司的名称（**parent company name**）和规模，比如说“大型”、“中等”或“小型”（**customer size**）。金沙经销商表示它只需要按照总公司的规模对**总公司**（PARENT COMPANY）进行分类，因此**总公司**（PARENT COMPANY）只包含顾客规模属性（并且没有其他的分类属性）。

“每个**总公司**（PARENT COMPANY）必定是一个或多个**顾客**（CUSTOMER(s)）的上级机构，而每个**顾客**（CUSTOMER）只能属于一个**总公司**（PARENT COMPANY）。”这个没有处理**顾客**没有总公司的可能性，但是如果你捕获了**总公司**，那么该公司至少得包含一个**客户**。因此，在这个模型中，他们可以应用源自第4章节的一级递归模式来建立非常简单的二级分层结构。

你还可以看到当**顾客**与**订单**相关联时，模型还捕获了**顾客**的上下文角色。三个关系中的任何一个都捕获了顾客在订单上下文中所扮演的角色。从图9-1中你可以看到三个相关联的上下文角色：“每个订单的账单必须只发给一个顾客，货物也只能运送至一个顾客，并且这个订单也只能是由一个顾客来下单。”这个模型捕获了**顾客**所扮演的与**订单**相关的特定的上下文角色（付款客户、运达客户和最终用户客户）。需要注意的是数据专业人士在此处应用了二级上下文角色模式，因为他们认为二级上下文角色模式对更好的理解数据需求比一级上下文角色模式更加管用。从图9-1中，你可以看出大多数的顾客数据被用来作为实体**顾客**的属性。在一级模式的应用中，模型倾向于包含很多的属性。例如，一级联系机制模式通过属性工作国家电话代码（**work country telephone code**）、工作区域代码（**work area code**）和工作电话号码（**work telephone number**）来捕获电话号码，而在应用一级状态模式时，导致了属性激活起始日期（**active from date**）和激活终止日期（**active thru date**），这两个属性捕获了顾客实例在过去的两年内是否与金沙经销商是否有些生意往来。正如我们所说，特定的属性页可以对顾客进行分类，比如说顾客规模（**customer size**）、顾客类型（**customer type**）或顾客行业（**customer industry**）。

数据专业人士使用了图9-1所示的数据模型来验证数据需求，并将该模型作为金沙经销商主题专家的作用域声明的一部分。他们得到了以下反馈：

* 除了激活状态和未激活状态，**顾客**可能还有很多其他的状态。
* 主题专家还想要捕获**总公司**的联系信息，因为**总公司**的联系信息可能与**顾客**联系信息不同。但是，他们也表示**总公司**不可以有用于“个人”用途的联系机制，比如说个人电话号码（**personal telephone number**）。
* 金沙经销商希望捕获**总公司**和**顾客**的状态、联系机制和规模的分类。
* 金沙经销商的主题专家表明如果**顾客**有**总公司**，那么他们**顾客**的**总公司**是所有订单的付款人。换句话说，如果存在**总公司**，那么所有的账单都会发送至**总公司**而不是发给**顾客**，若是没有**总公司**，则将账单发送至**顾客**。
* 金沙经销商的主题专家还表示所有的**订单**应当交付到客户手上，绝对不能交付到**总公司**，即使是**总公司**为订单付的款。也就是说，所有的装运直接发送至**顾客**而不是**总公司**。
* 金沙经销商只能根据规模对**总公司**进行分类，而**顾客**则可以根据规模、类型和行业对其进行分类。顾客还表示他们并不知道所有的顾客类型、规模或行业，它希望在前进的过程中捕获这些。此外，公司表示同一个客户可以在同一时间从事多个行业（例如，一个可会即可以从事生产制造行业也可以从事分销行业）。

数据团队创建了一个可用的原型，该原型展示了集成顾客数据可以提供的可能性。鉴于这一点和金沙经销商主题专家的反馈，数据团队制作了图9-2，将其作为作用域声明的一部分和原型数据库设计的基础。需要注意的是在图9-2中，为了能够容纳其他的需求，使用二级状态模式和二级分类模式替换了一级状态模式和一级分类模式。此外，**总公司**如今采用了一级联系机制模式来记录与**总公司**相关的联系信息。数据模型表示了订单账单必须发送至**顾客**或者是**总公司**。其中的规则是如果存在总公司，那么订单的账单必须寄送至总公司，这个规则被记录在业务规则文档中，该文档是数据模型的附属文档。他们想到二级业务规则模式中所使用的**订单规则**实体；但是，他们认为该模式太过复杂了而不适合用于原型。

为了满足所发掘的其他需求，该团队将图9-1所示的模型转变成图9-2所示的原型模型。这是个十分简单直白的流程，有以下部分组成：

* 将分类和状态的一级模式替换成二级模式。这样满足了能够更容易地添加新分类和状态的实例这样的需求。在二级模式中，为了满足顾客可以归类于多个行业这一需求，团队还在顾客至行业类型之间使用了多对多关系。

（图）

**图9-2** 原型数据模型

* 将一级联系机制模式添加到总公司，所以包括了所需的联系机制的个别属性【批注6】。
* 将订单的上下文角色和（一个互斥弧【批注7】）添加至总公司，因为发现如果存在**总公司**的话，订单的账单都要发到**总公司**，如果没有**总公司**，那么账单将会寄给**顾客**。

注：如果原型需要更多的灵活性的话，在图9-1中的一级模式（或二级模式）也可以使用三级模式甚至是四级模式来替代。例如，如果团队认为他们需要演示如何实时添加新的分类类型的话，该团队可能会使用三级分类模式。那就是此数据模型模式的优美之处；他们本质上都提供了同样的核心功能—或多或少地提供了一定程度的灵活性。这意味着他们是可以相互替换的。随着你对本章节中模型的每个演变的了解，你可以看到关于此的详细阐释。

## 为什么我们要这么做？

这些特别的模型都是非常特定的。一级模式大多数情况下会被用来创建图9-1（除了二级上下文角色模式），一级模式和二级模式的结合是用来创建图9-2的。

那么，为什么数据团队大多数情况下选择一级模式创建图9-1？金沙经销商的数据需求是非常特定的，而且这些模型的受众是非技术人员，而他们想要看看他们的需求是如何被满足的，所以团队使用了非常具体的模式来启动分析流程，采用了非常易于理解的数据模型结构。第一个模型图的主要目的是交流和验证数据需求，用来作为作用域声明的一部分。数据团队想尽可能地让模型图更加清晰，以更加易于理解的方式展现数据需求。为了达到这一目的，他们特定为与业务相关的重要数据元素命名。在图9-1中，你可以看到主要的数据实体：**总公司**、**顾客**和**订单**。然后他们向下钻取跟多的特定业务数据，并将这些数据作为实体的属性或关系。

图9-1模型图的主要目的是让交流流程更加顺利，将模型作为帮助理解数据需求的工具。很容易向大多数受众解释它，特别是和支持材料一起使用时，比如说和用例和示例模型一起使用。

**注：在图9-1和图9-2所示的模型图中，我们省去了主键属性和外键属性。如果你认为这些属性会让你的受众感到很困惑的话，删除它们是很容易的。当我们向非技术受众展现模型图时，通常我们自己删去这些属性。**

为了更好地理解数据需求，这个模型图还可以以非常简单的方式来帮助启动数据建模流程。这一点通常会被忘记。这个模型图还可以帮助数据建模团队让授权调查区域形象化。有些高级数据建模人员可能认为这样做没有必要，想要直接跳转泛化和规范化的数据模型。但是，就我们的经验来说，我们认为数据建模人员为了在使用泛化模型之前，使用特定的数据模型有助于他们弄清楚信息需求。

这个模型很容易在短时间内结合在一起。数据团队不想投入大量的时间构建一个复杂的模型作为部分作用域声明。他们认为他们需要额昂金沙经销商客户数据问题的分析工作‘运行起来’。当将这些模型作为有效工具来使用而非投入太多时间“单独建立模型”时，这个模型有助于他们快速地发现核心术语、核心问题和核心概念。也就是说，他们使用这个方案而非高度泛化的模型，因为他们不能有效地使用高度泛化的模型来收集验证数据需求。

数据团队在捕获业务使用的正确术语方面十分谨慎（也就是**总公司**、**顾客**和**订单**）。这个问题看似无所谓，但是在数据建模工作中每个人都说同样的语言是很重要的。在与业务代表谈论数据需求时，一级模式有助于合适术语的确定。也就是说，数据团队将这个模型建成一个‘稻草人’，以便业务代表可以指出这数据团队哪里分析错了。例如，在审查图9-1所示的十分简单的数据模型后，业务代表看到他们想要总公司捕获的相关联系机制，他们指出顾客和总公司除了有“激活”状态和“未激活”状态外还可以有其他的状态等等。

**注：我们还可以使用图9-1作为原型的基础。事实上，起初使用一级模式来构建原型比较简单，因为他们是如此的特定，因而比较容易理解和用于编程。**

使用这些模式的解决方案的优点有哪些？

理解数据需求和构建原型（如图9-1和图9-2所示）时，使用特定模式（比如说一级模式和二级模式）的优点如下：

* **易于理解。**这些模型使用了一级和二级模式，模型的特定特性可以让你以十分直白简单的方式展现调查主题区域范围内所有关键实体和属性。这些模型如对于收集验证数据需求是非常有用的，除此外，还可以将他们作为作用域声明的一部分。
* **容易使用。**一级和二级模式是很容易应用的，在数据模型的开发工作中，这些模式很容易使用，而且便于在较短的时间内沟通交流。他们可以让数据团队在金沙经销商的协助下快速‘行动起来’。
* **容易实现原型。**一般来说，基于这些简单的结构，建立原型会更加容易更加迅速。
* **良好的起点。**这类模型为你启动数据开发流程提供了良好的起点。换句话说，你必须从某个地方开始，那么为什么不从最清晰简单的模型开始呢？这些模型可以快速启动确定范围流程，构建原型流程以及拓展数据模型流程使其更加稳定和灵活，比如说再某些地方添加二级模式，就像我们在此情景中所做的一样。
* **用于同业务代表的交流沟通很有效。**当我们同业务人员交谈时，我们通常将这些模型当作交流工具使用。在这个模型中展示了通用的业务术语，这样有助于业务人员参与建模练习。我们一致很乐意看到业务人员发现这些模型很容易理解，处理起来也很有意思。
* **变更成较高级别的模式也简单。**当我们发掘‘不了解’的信息时，比如说“我们知道我们捕获了订单的状态，但是我们不知道那些状态是什么”，可以非常之间的使用等同的二级模式（甚至可以使用三级模式）替换一级模式。较高级别的模式（二级、三级和四级）为位置信息提供了更多的‘存储空间’。当然，这不能成为我们逃避分析挖掘这些‘未知信息’的借口。

## 使用了这种模式的解决方案的缺陷有哪些？

将此模式应用于需求收集和原型设计有以下几点缺陷：

* **一般来说，一级模式不能为实现提供良好的基础。**但是，在数据集一般是静态的而且值一般也不会被复用的一些具体情况下，你可以将一级模式的解决方案用于实现，但是这些方案却很少被真正地被需要。此外，如果在数据库设计中实现了一级模式，他们通常会引起冗余数据问题。一级模式通常包含了很多未被泛化的属性（也就是说，他们遵循一级范式、二级范式和三级范式）。例如，如果你使用一级分类模式实现顾客规模（**customer size**）字段，该字段有值：“大型”、“中等”和“小型”，这个方案将产生很多实例，这些实例会冗余地存储值“大型”，它与使用一个指向客户规模表的外键解决方案相反，使用外键的方案是使用了二级分类模式来建模的。
* **当使用二级模式而非一级模式时，你会丢失一些特定的理解。**例如，我们在图9-2中用**状态类型**和**顾客状态**这两个实体替换了图9-1中**顾客**的两个属性：激活起始日期（**active from data**）和激活终止日期（**active thru date**）。从图9-1中可以很明显地看出“激活”状态（激活起始日期、激活终止日期）。在表9-2中这一点并未明显体现。也就是说特定的属性（激活起始日期、激活终止日期）被泛化成**状体类型**实体。
* **不能复用通用例程。**当你使用很特定的模式时，在一个模型中数据模型倾向于各不相同，因此，使用特定结构的模型不像使用一些泛化结构的模型，它不存在可以用于模式各个部分的通用例程。
* **缺乏灵活性。**尽管一级模式和二级模式更加容易理解，但是在灵活性方面还存在诸多缺陷。例如，需要将数据模型变成适合多种情形的模型，比如说当发掘新的分类类型时、同一时间需要多个顾客状态时、需要先的上下文角色类型时、要使用新的联系机制类型时或顾客组织有两个以上的层级时。

## 简介

这一节处理的是金沙经销商客户信息系统开发的开始部分。数据团队创建了一个简单的特定的数据模型，将其作为作用域声明的一部分，用来启动客户数据的讨论和数据模型的创建。该数据模型捕获了通用的术语和对金沙来说十分重要的属性。

数据建模人员在使用第一个数据模型获得主题专家的反馈后，出现了其他的需求。然后数据建模人士拓展了图9-1中的模型，并创建了可以用于初始化原型的数据模型（图9-2）。这个模型比图9-1所示的模型要更加灵活一些。

将几个以及模式替换成二级模式便可以快速创建增强版数据模型。例如，在图9-2中，在图9-1中的分类属性变成了独立的实体，多对多关系的创建让一个顾客可以有多个行业分类，其他的上下文角色的添加可以让总公司成为付款顾客，并且状态可以使用自己的实体**状态类型**来维护。这样的模型为业务人员和IT之间创建了良好的交流基线，并且一个更加灵活的初始化数据模型可以用于作为创建一些原型屏幕和阐释顾客信息需求报告的基础。

一般来说，我们认为特定的数据模型（比如说图9-1和图9-2所展示的那些数据模型）应当由其他的支持文档，比如说实例模型图和用例。尽管特定的数据模型可能对沟通很有效，但是拇指规则规定你不应当将基于一级模式的初始化模型（如图9-1所示）用于实现，因为它可能会引起很多的数据异常现象。同样，你还应当要注意，不要使用原型模型作为全面的实现，因为支持主要产品应用通常需要灵活性，而他们不具备这种灵活性。

## 应用数据模型

很多数据专业人士的‘基本’工作是为应用创建数据模型。这一节介绍了如何使用本书中的模式开发强健的、灵活的、一致的、实用的模型，这样的模型可以构成一个坚实的应用数据库设计基础。

### 此模型的场景

在想高级管理层展现过原型、再次检查项目作用域之后，数据图团队被要求制作一个成熟的数据模型，这个模型要支持一个包含以下操作的应用：顾客数据的录入、更新、访问和管理。有趣的是，金沙经销商想要建立这样的应用，但是对于可能存在的顾客状态、顾客上下文角色或声明角色还没有好的主意。这是很多应用开发项目都经常遇到的情形。

### 这些模型是如何运作的？

在查看过图9-2所介绍的模型后，数据团队决定将更加灵活的方案应用到应用数据模型中。他们首先根据图9-2确认了数据的作用域。第二，他们同主题专家（包括业务和技术）讨论开发一个成熟的产品应用所需的功能。基于这个条论，他们决定使用灵活性换取一些简洁性。为了达到这一目的，他们使用图9-3所示的二级和三级数据模型结构替换了图9-2所示的一些一级和二级数据模型模式。

主题专家统一在某一给定时刻点顾客可以有多个状态。例如，**顾客**可以在同一时刻既处于“激活状态”又处于“调查中”状态。可以从模型图的顶部看到，为了支持这样的需求，数据团队使用图9-3中所示的三级状态模式替换了图9-2所示的二级状态模式。这个新的结构支持**顾客**同时具有多个**状态类型**。

数据团队提出了另外一点是图9-2所介绍的分层结构可能不支持企业将来的需求。他们指出他们的很多顾客具有两个以上的层级结构，比如说总公司、分公司、分支机构和部门。为了支持这一需求，他们图9-3所示的二级递归模式替换了图9-2（“每个**总公司**可以是一个或多个顾客的上级机构”）所示的一级递归模式（一个简单的层级结构），图9-3展示了“一个顾客可以进一步分解成一个或多个顾客（然后分解后的顾客还可以进一步分解成一个或多个顾客，以此类推）”。

因为数据团队现在使用顾客的递归关系捕获了总公司与顾客之间的层级关系，所以他们不再需要捕获总公司实体或该实体的任何属性和关系，因为在顾客实体中捕获了它们。这不仅有助于让顾客的层级结构可以有任意数据的层级，而且还有助于整合客户和总公司的数据。

现在只用顾客实体而非使用总公司和顾客这两个实体来维护所有的分类、状态和上下文角色。这样可以让数据团队整合图9-2中的一些关系。比如说顾客规模的两个关系和指定付账角色的订单的两个关系。使用图9-3中的递归关系的最大的益处在于：如果需要变更层级（例如，变成总公司、分公司和分支机构），那么在无需改变数据模型的情况下，可以原封不动地被保留所有的分类、状态和上下文角色。

放眼未来，数据团队还需要添加支持声明角色的当事人和当事人角色概念。将通用的当事人、组织和人员的信息（比如说名称【组织名称、名字】和姓氏【人员的姓】）存在一个地方这样的方法可以让模式标准化，并且在没有冗余地捕获相同数据（当事人人所扮演的每个角色都具有的数据）的情况下可以让当事人扮演多个角色。那么。例如，扮演了顾客角色的当事人还可以扮演合作伙伴角色和/或供应商角色，而当事人数据只会被捕获一次。尽管这并不是当前应用的需求，但是团队认识到这是出于将来的考虑。例如，他们的企业可能希望特别考虑下顾客也是合作伙伴的情况。因此，这个模型包含**顾客**和**当事人角色**和**角色类型**分类的所维护的其他声明角色。这个三级声明角色让数据团队可以使用当事人角色捕获所有角色的通用和关系。

这个模型另外一个重大改动是添加了三级上下文角色模式。金沙经销商的IT主题专家同意采用灵活地方式捕获订单相关的所有当事人最终会有益于应用。他们不确定在将来会存在哪些新的订单角色，但是他们可以确定的是当新的流程、角色和规则随着时间的推移而发展时，需要加新的角色。考虑到这一点，数据团队使用三级上下文角色模式替换了图9-2（遵循二级上下文角色模式【付账角色、装运角色和订购用户角色】【批注8】）所示的三个关系，三级上下文角色是由当事人所扮演的订单角色这一关联实体构成，一个订单，也可以由订单角色类型来描述，订单角色类型是角色类型的子类。因此，当有其他的订单角色出现，比如说“顾客联络员”、“下单人员”、“取单人员”、“安装人员”、“销售人员”和“质量保证人”等等，不需要变更模型便可容纳这些角色。

原型数据模型与应用数据模型之间的另外一个变化是应用数据模型使用二级联系机制模式替换了一级联系机制模式。金沙经销商的主题专家指出同一个**顾客**实例可以与多个**电信号码**、**邮政地址**或**电子地址**（例如，电子邮件地址）相关联。此外，他们发现有些电话号码、邮件地址和电子邮件地址为多个顾客所共享，特别是那些在同一个公司的分部门工作的单个顾客。数据团队总结认为他们需要捕获每个当事人的电信号码、电子地址和邮政地址的**联系机制用途**和**联系机制用途类型**。主题专家讲述了一个销售人员的故事，该销售人员使用他的个人电话号码联系一个高级合作伙伴的客户（客户要求不使用这样的号码），然后该销售人员试图发展新的业务；这差点让金沙经销商丢失了这个账户。因此，需要为各种联系机制捕获合适的目的和用途。需要注意的除了可以使用目的（“付款”）和用途（“私人”）来对联系机制进行分类外，还可以使用**当事人电信号码**、**当事人电子地址**和**当事人邮政地址**中的免推销属性（在第7章节的结尾部分讨论过）来标识当事人的联系机制不应当用于推销目的。

### 为什么我要使用这种方法？

建立一个应用数据模型需要数据专业人士兼顾已知的和将来的需求，有些需求对企业或应用开发团队来讲不是很明显。在此应用数据模型中，数据团队打算在需要的时候综合使用特定数据模型模式和泛化数据模型模式。这一节解释了为何我们使用图9-3所示的方法。

一个成熟的顾客信息应用程序通常需要一个灵活的模型，该模型要可以容纳很多当前和将来的需求。所以，数据团队需要使用提供这种灵活性的高层级模式来替换一些低层级的模式。例如，图9-2所展示的反应总公司与顾客之间关系的分层机构就是非常特定的模型。该模型表示“每个总公司可以是一个或多个顾客的上级机构”。但是如果顾客的组织结构的层级在两个以上该如何是好呢？大多数大型的企业都是有多个层级的。为了确保这个数据模型可以处理这方面的问题，数据团队使用了二级递归模式，这个模式在创建多层级的分层结构（和聚合机构）方面提供了一些灵活性，很多的金沙经销商顾客都需要这样的分层结构。同样，图9-3所示的声明角色、上下文角色、状态和联系机制也使用了灵活的数据模型模式，这种模式为满足大量的当前的和将来的需求提供一种途径。

以一致的方式开发一个应用数据模型需要考虑的因素之一是应用数据是否能够与企业中的其他数据集成（以及外部数据）。团队可以通过以下方式来促进集成：在应用中使用一致的模式；在企业数据模型中和支持主数据的数据模型中使用同一类型的模式。例如，数据团队认为他们需要的是三级声明角色模式而不是图9-1和图9-2所示的特定的一级模式。尽管使用这种模式有很多好的评论（本章节讨论了这方面的内容，在数据模型资源手册的第一卷修订版【Wiley,2001】中详细介绍了这些信息），但是使用这个模式决定性的缘由是企业的很多其他的数据模型也使用了当事人、人员、组织和当事人角色的概念。事实上，很多第三方的工具也使用了它（例如，在写这本书时，大多数的灵性企业资源规划(ERP)包都使用了‘当事人’概念，在他们的逻辑数据模型中也有**当事人**实体）。因此，如果你使用**当事人**/**当事人**角色概念，你可以很容易地将外部数据集成到你的系统当中，各系统间的数据交换也会更加容易。

一个通常被忽略的因素是为企业各种人群构建一个知识平台。一旦数据专业人士、程序员或高级用户理解了这个模式，他们便可以将该模式应用于企业的各个地方，并且人们理解各个系统基础会更加容易。例如，如果在订单录入系统中使用了三级分类模式，而在物流报告系统和财务管理系统中也同样使用了三级分类模式，那么构建关于这些系统的工作方式的知识平台会更加容易。这些系统的功能可能是不一样的，但是系统的构件是一样的。

应用数据模型需要满足顾客信息应用系统的特定需求。业务主题专家表示他们非常确定他们只需要根据规模、类型和行业对**顾客**进行分类。他们十分确定在这个特定的应用系统中没有其他的顾客分类需求。正是这份确定性，数据团队坚持使用二级模式（该模式被用来创建初始的原型和图9-2所示的模型），决定不使用更加泛化的（和更复杂）的三级分类模式，然而，我们建议对于这类决定要谨慎考虑，因为我们已遇到过很多这样的情形，似乎所有的分类都是已知的，但是后来发现应用系统还需要其他的分类。例如，在将来可能需要其他的分类，比如说少数名族所属企业分类和顾客市场细分分类。

理想化地说，构建应用系统的模型应当考虑将来的需求，以至于它能够处理企业不断变更的需求。就我们的经验来说，与应用系统的维护相比，应用系统的开发只占了小部分成本，当需要变更数据库设计时，会产生重大的应用开发成本。例如，数据团队发现业务主题专家和IT主题专家指导三种上下文角色（运达、付款和订单的最终用户）。但是他们也确定还存在他们尚未发现的新角色。这就是为何数据团队替换了原型中的特定的二级上下文角色模式，将三级上下文角色模式用于产品应用数据模型的原因。

### 在解决方案中使用这些模式有哪些优势？

在应用数据模型中使用这些模式的优点如下：

**在模型中使用提供灵活性的泛化模式但是兼顾应用的特定需求。**应用数据模型在各种业务需求之间提供了一个很好的平衡。该模型在分类和联系机制方面使用了二级模式，因为数据团队认为这个模式可以解决这些数据区的特定需求。例如，使用二级分类模式，模型可以维护特定的需求：比如说一个顾客只能由一个顾客规模并且该顾客只能根据顾客类型进行分类。为了能灵活地适应这些数据区，在声明角色、上下文角色（订单）和状态方面使用了三级模式，因为主题专家不确定这些区域将来会有什么样的需求。通过大多数地方使用三级模式（和一组二级模式）这种方式，模型可以在不需变更模型的情况下，添加新的声明角色类型、可以为订单添加新的上下文角色、新的状态类型、新的类型、目的和联系机制用途。

**通用的集成解决方案。**这个解决方案可以作为一个基础，将数据集成到比较通用的数据结构中；例如，含有**人员**和**组织**实体的这种数据模型最有可能与企业数据模型以及其他应用模型是一致的。

一致性。该模型是以即插即用的方式使用这些模式的。使用这些模式的优点在于在需要灵活性的时候，数据团队可以将某一模式替换成

该模式更为灵活的版本，当业务需要更加特定的解决方案时，他们可以保留较为特定的模式（或者是在模型中使用较低层级的模式）。因此，模式复用同类型的结构会产生相似类型的数据模型结构，并且会模型更加标准。

### 将这些模式应用到解决方案中会存在哪些缺陷？

将这些模式应用至应用模型会存在以下几个缺陷：

**模型的有些部分不灵活而有些部分又很复杂。**使用较为特定的模式时会存在不灵活的缺点，使用较为泛化的模型时会存在复杂性的问题。数据团队必须在各种音效应用数据模型的因素之间寻找一个合适的平衡点。这一点是不容易做到的，它需要一定的经验。

**由于这些模式具有很多层级，所以缺乏通用的建模风格。**由于不同层级的模式是可以相互转换的，所以这类模式可以产生多种风格的数据模型—也就是，较为特定的二级模式和更为泛化的三级模式。有些建模人员可能对这种看起来像混合风格的模型感到不适应。数据建模人员需要说明以下观点：在模型中使用相同风格和级别的模式，在各应用中开发可复用的例程会更加容易。例如，在面向服务的架构中（SOA），可以建立通用的类似服务来调用各种数据分类，并且如果全用三级模式为它们建立模型，有助于创建一致的服务开发方式。

**使用较为泛化的模式时缺乏可理解性。**对于业务代表来说这类模式可能太过复杂了，理解这类模式的难易程度取决于他们从事数据模型工作的经验。一般来说，与泛化模型相比，特定的模型更适合用来同非技术受众沟通数据需求，因为特定的模型更加容易理解。

**使用较为泛化的模式时，会缺失一些业务规则规范。**与泛化模式相比，使用特定的模式构建的模型可以展现更多的有业务规则和规范。从这种意义上来说，特定的模式可以捕获更多的实体和关系所特有的数据需求，比如说关系的可选性，关系基数和更特定的属性和关系。例如，通过订单角色实体将三级模式用于上下文角色的这种方式不能维护下面这种情况：一个订单只能有一个特定的角色，比如说有且只能有一个“付款”顾客。一种是需要在数据模型中能够维护某些业务规则，一种是需要数据模型解决方案能够支持不断发生变化的业务需求，这两种需求需要保持平衡。

### 简介

在这一节，介绍了如何为金沙经销商创建顾客信息应用数据模型。介绍了数据团队是如何将图9-2所示的原型模型演变成成熟的应用数据模型。他们是通过使用较为灵活的模式（二级和三级模式）替换一些特定的模式（一级和二级模式）这种方式来达到这一目标的。

数据团队认为在使用灵活模型的同时要兼顾特定的需求。满足特定的需求能够让该模型成为为顾客量身定制的模型。使用灵活风格的模型，添加现的数据需求会更加容易。这意味着数据团队使用哪个层级的模式应视当前和将来的需求而定。例如，他们在三级模式中所添加的当事人和当事人角色的概念有助于将来添加新的声明角色，除此外，这还有助于一次性捕获通用的**人员**和**组织**的信息，不用管当事人可以扮演多少种角色。另一方面，数据团队维护了更多的有关顾客的特定分类，因为企业表明这些分类是顾客应用对顾客进行分类的唯一方法。

有些数据建模人员可能会反对在一个数据模型中使用不同风格和不同层级的模式，他们认为使用同类型的模式，有助于复用访问数据的通用功能并且可以使用一致的方式来管理数据。如果是这种情况的话，建模人员可以选择同一层级的模式来开发应用数据模型，从而进一步促进一致性。

## 企业数据模型

在过去的10年中，随着很多企业逐渐认识到信息是一笔有价值的资产，企业数据模型变成了更加重要的构件。一些公司使用他们的数据获得了竞争优势，比如说，网飞公司和沃尔玛公司。他们赢得优势的核心部分是具有通用的、一致的数据视图【批注9】。

企业数据模型可以帮助企业理解他们的数据视图。一个企业数据模型可以提供：

**为查看企业主要的实体、属性和关系提供了标准的方式。**例如，在一个特定的企业数据模型中，比如说在496页图9-4所示的模型中，一些核心实体是**当事人**、**当事人角色**、**当事人关系**、**联系机制**、**订单**和**产品**，该企业数据模型还展现了这些实体的一些主要属性和关系。在图9-4中还有一些其他的核心实体没有展现，这些实体也是企业数据模型的一部分，比如说**装运**、**发票**、**支付**和**工作计划**。企业数据模型可以阐释处理关系的通用方式，比如说展现**订单项**与**装运项**之间的多对多关系。尽管有些特定的应用可能会使用不同的术语来表示这些实体或者是这些应用需要不同的关系，但是企业数据模型提供了可以应用于很多应用系统的通用模型，因此，企业数据模型为企业开发一致的数据模型提供了依据。使用这些模式有助于开发一致的数据模型，因为这些模式提供了构建通用数据结构的标准方法，比如说分类、状态、声明角色、上下文角色等等。你可以使用这些模式为贵企业创建一致的企业数据模型。

**灵活的数据模型可以用于多个应用程序。**企业数据模型倾向于使用很多泛化的结构，这样这些模型可以广泛地应用于各个应用程序。这些泛化的数据模型结构通常是模式的示例。例如，特定的应用可以通过下面这种方式为顾客和顾客的上级企业构建模型：使用一个含有外键的**顾客**实体，通过外键将该实体关联至**上级顾客**实体。然而，企业模型可以展示一个更泛化的数据模型结构，通过当事人关系机构（三级递归模式）能够让扮演任何角色的当事人关联至其他的当事人（该当事人可以扮演任何其他角色）。因此，企业数据模型满足特定应用需求的同时也满足很多其他应用的需求。

提供了指导方法并确保了应用程序中的实体、属性和关系的质量，并且还展现了他们出错的地方。可以使用企业数据模型来确保应用数据模型的质量和验证应用数据模型。这样，企业数据模型可以充当一个关卡。例如，应用程序展示：在一个模型中**产品**是在**设施**中维护的。企业数据模型可以指出：根据企业的数据视图，实际上产品适于库存项相关（表示产品的一个特定实例—例如，产品XYZ，序列号是#1458），而库存项实际上是存储在设施中的条项，而产品知识一个目录条项，就产品本身而论，实际上根本没有物理存储该产品。此外，在反复核对模型时，这些模式也会有很大的帮助。这些模式提供了标准的通用事物处理方式。例如，企业数据模型可以指出管理分类或上下文角色的企业视图。

**为相同概念建模提供了更多的选择（如果企业数据模型使用了在同一模型内展现备选数据模型这样的技巧）。**有时在开发企业数据模型时我们会使用这样的一种方法：使用泛化（三级或四级模式）和特定（一级或二级模式）这两种方式来构建同一个数据模型。 例如，企业数据模型可以展现概括的方式，该方式可以支持所有**当事人角色**的分类，还可以展现特定的分类实体，比如说针对**顾客**（一个当事人角色）可以展现**顾客规模**、**行业类型**和**顾客类型**。因而你可以向不同的受众展现不同的企业数据模型视图。例如，为了确定这个模型是否足够灵活，数据架构师可能会希望查看下数据模型的泛化风格部分，而业务代表为了确保他们的需求都被满足了，可能想要查看该部分的特定视图。由于不同的应用会有不同的需求，所以这项技巧还为复用企业数据模型中的结构提供了多个备选方案。

**注：我们介绍企业数据模型方案的目的是为同类型的数据提供多种建模方式：一种是使用特定的建模方法，一种是更加泛化的建模方式。这并不表示该模式可以冗余地维护相同的数据。可以认为该企业数据模型不仅为企业提供了标准，而且还提供可备选方案。例如，当你需要特定的顾客分类方案时，你可以选择使用特定的或更加泛化的分类数据模型结构。第三种方案是使用特定的和泛化的这两种数据模型结构都使用，对于某些重要的数据使用特定模式，而其他的数据可以使用更加泛化的模型，特别是那些随着时间推移而出现的新数据类型。如果你选择了这个方案，你需要明确指出是什么样的数据以及在什么样的情形下你使用了特定的数据模型结构与泛化的数据模型结构。例如，我们可以使用一级状态模式为订单日期（非常重要的数据）建模，对于订单的其他状态则可以使用三级状态模式。另外一中说法是企业数据模型有时可能会展现特定的和泛化的数据模型结构以适应同类型的数据需求，比如说为顾客或当事人角色的分类建模。我们不认为这个方案维护了冗余数据，因为我们并没提倡使用这两种方法捕获相同的数据实例。我们认为在提供标准方面，有效的备选数据模型是非常有用的，同时在不同的情形中还可以具有多个选择。**

创建企业数据模型通常是很困难的大型任务。收集理解企业的数据架构需要组织和方法。我们经常会看到大型企业数据模型的各个部分缺乏一致性。这些模式有助于这个问题的解决因为这些模式从如何为适用于任何数据格式的通用结构建模的决定中剔除了猜测。例如，为**产品**分类建模和为**顾客**、**资产**、**工作计划**或任何其他需要多个分类类型的实体建模时，数据建模人员可以使用相同的模式。只需要变更下实体的名称，因为他们处理的是不同的数据。为**订单**状态建模可以采用与以下实体状态建模一样的模式：**装运**、**发票**、**支付**、**工作计划**和/或**产品**。这些模式允许数据建模人员专注于企业数据模型的独特方面，比如说将重点放在如何创建产品成本结构，而非如何为递归关系或上下文角色创建最好的模型。建模人员还可以决定在声明情形下使用哪个层级的模式，并且可以认识到哪些地方需要类似的模型。由于数据建模人员在为同类型的数据建模时，可以在整个企业数据模型和各个应用模型中使用一致的数据模型结构，所以这些模式提供了很大的一致性。

### 此模型的场景

金沙经销商运营委员会意识到他们有不同的需求和企业信息是一笔需要管理的重要公司资产。他们认同以下观点：在开发顾客信息应用时，还有很多其他的应用在为顾客数据建模时也有相同类型的数据需求。使用一致的数据建模结构，企业可以提高交流效益、削减维护成本、简化接口和提高数据质量。所以他们要求数据团队提供策略，要能够有效一致的管理数据。

作为企业数据战略的一部分，数据团队决定创建一个企业数据模型，展现出企业数据资产的性质。企业数据模型有以下两个目的：

* 第一，介绍整个企业当前和将来的数据需求。
* 第二，为数据建模工作提供可以复用的选择方案，以便快速启动建模工作，为集成同更多一致性，为其他的数据建模工作提供质量确保关卡。

基于业务需求和金沙运营委员会的授权，这些模式可以为数据团队的开发提供了一个粗线条的企业数据模型。图9-4展现了部分企业数据模型，该模型的重点是顾客以及与顾客相关的订单和产品。

### 这些模型是如何工作的？

在本章节的先前部分，你可以看出使用哪个层级的模式是取决于业务需求的。例如，在本章节的第一部分，创建见到的作用域声明模型时，我们使用的是一级和二级模式，因为这些模式展现了相关业务的特定属性和实体。相反，在稍后的小节中，我们需要创建成熟的应用，模型的大部分使用了二级和三级模式，因为业务要求数据产品应用要更加灵活。如今，在企业数据模型中，我们需要采用不同的方案。

在图9-4中，你可以看到一个混合型的方案，以多种方式为同类型的数据建模！那么，这是什么意思呢？在这个方案中，你可以在同一个模型中使用一级、二级和三级模式，有时甚至可以为同一结构提供两种不同的选择。例如，企业数据模型捕获了**顾客类型**、**顾客规模**和**顾客行业**，根据这些实体对**顾客**进行分类。在创建顾客信息应用程序时（如图9-3所示），你可以再次调用应用开发团队所使用的这个方案。企业说根据类型、规模和行业对顾客进行分类是很普遍的，所以这个模型展示了顾客分类建模的标准方法。你还可以从图9-4中看到**当事人角色类别分类**（以及相关的类别实体）。这是三级分类模式。这个数据模型模式提供了更泛化的分类建模方法，它包含了二级分类模式所提供的所有功能；但是，这个模式要更加全面。这个模式能够维护任何当前或将来的类别类型，不仅是可以维护顾客的分类还可以维护任何类型的当事人角色的分类。为了能够维护较高层级和较低层级的分类，这个方案还提供了维护多层级类别和类别类型的功能。

你可以看到一个类似的示例，在该示例中企业数据模型应用了两个不同层级的模式。在图9-4中，你可以看到“每个**顾客**可以进一步拆分成一个或多个**顾客**”。该模型使用二级递归模式捕获了顾客内部组织结构。你还可以看到“每个**当事人角色**可以与一个或多个**当事人关系**相互关联”。然后当事人关系关联至当事人关系类型；“上级机构与分支机构之间的关系”、“分支机构与部分之间的关系”。因此将三级递归模式应用于当事人关系可以捕获顾客的分层关系以及任何其他当事人角色之间的其他关系。企业数据模型使用这个方案可以让一个应用有两个选择：顾客的简单递归关系或非常全面的方法（在某一角色上下文内将当事人关联至在另外一个角色上下文内的其他当事人角色）。此外，值得一提的是我们不提倡冗余地捕获相同的数据。我们只是展示了同类型事物的各种可行建模方案。

你还可以从图9-4中看到混合型上下文角色模式的示例。我们使用二级上下文角色模式捕获了付款顾客。通过使用订单角色，我们还使用了三级上下文角色，这个模式可以捕获任意数目的当事人以及与当事人相关的角色。这个模式可以捕获订单销售人员、取单人员、订单录入人员、订单质量确保当事人以及任意数目的其他角色，这些角色随着时间的推移或流程的变更，模式可能会需要这些角色。

有意思的是在这个示例中其实企业数据模型展现了三个实现方案。

* 第一，开发小组只能通过外键关联至付款顾客、装运顾客和/或订单的最终用户（二级模式），如果那就是他们所关注的事项的话。（顺便说下，从订单项到顾客，还有其他的二级上下文角色模式比如说运送目标顾客展示了特定的订单项的指定运达地址）。
* 第二，如果开发组需要捕获每个可能的上下文角色，那么他们可以考虑实现三级上下文角色模式，该模式使用了**订单角色**，并关联至**当事人角色类型**和**订单**。
* 第三，特别是，如果他们需要捕获付款顾客关系（因为这个非常重要的关系）以及其他的订单角色（和关联关系），他们可以使用这两种数据模型结构，换句话说，使用混合型模式。

企业数据模型并不一定要展现每一个方案。试想下，对企业做了一些调查后，发现总的来说企业想要标准化更加泛化的三级联系机制模式、换句话说，他们在企业内达成一致意见，认为该方案就是企业查看联系数据的方式，所以企业数据模型没必要包含其他的选项。

模式将不同的建模风格和方式应用于企业数据模型。这个场景介绍了使用‘混合型’方案的方法，该方案可以维护多个层级模式，甚至是相同的数据建模结构也可以使用多个层级模式。其他的备选方案如下：

* **在整个企业数据模型中一致的使用相同层级的模式（一级、二级或三级模式）。**
* **数据模型结构都采用两种模式：特定模式和泛化模式。**你可以将这种方式看成保持企业数据模型一致性的方法；但是，对于企业已决定只使用一种方法实现某类数据模型结构的数据区域没有必要使用这个方案。在示例中，我们将联系机制作为一个已获得企业一致认可的区域。所以，在企业数据模型中创建两个可选方案可能会引起混乱，因为企业并不提倡查看这个数据类型的多个备选模式。
* **创建不同版本的企业数据模型，例如，使用二级模式的企业数据模型版本和使用三级模式的企业数据模型。**有些人会将二级企业数据模型作为原型或作为企业数据模型，专供业务代表查看。这类模型的主要目的是以一种易理解的方式阐释信息需求。那么可以有一个单独（但是同步）的三级企业数据模型，用来作为架构数据模型，这类模型是专为数据专业人士这样的受众而设计的，比如说架构师。这类模型的主要用途是充当开发稳定的可复用的数据库结构的灵活基础。

**注：基于20世纪70年代ANSI委员会关于概念性模型的探讨，有些数据专业人士还将企业数据模型看成是概念性模型的同义词。遗憾的是，概念性模型的含义在数据管理行业没有一直的定论。这也是为什么我们选择一级、二级和三级这样的新概念对这些模型进行分类的原因。在我们的脑海中，企业数据模型或者是概念性数据模型的主要的目的是以一种易于理解的方式阐释信息需求，所以一般来说我们提倡使用一级模式和二级模式来完成这个工作。但是，我们也不会忽视三级模式的使用；至于使用哪种模式要根据受众的情况来定。我们使用企业数据模型/概念性模型的准则是采用可以得到信息的模式 。‘架构型’数据模型包含了规范化的技术，是专门为技术受众而设计的，有时候被称为逻辑数据模型。但是，关于逻辑数据模型是什么或者说应当是什么，在业内没有一致定论。因此，在这本书中，我们使用层级来表示模型，我们可根据泛化的层级对模型进行分类，避免误解我们所要表达的概念型数据模型和/或逻辑数据模型的含义。**

### 为什么我要使用这个方法？

从某种程度上来说，企业数据模型必须是全体人员所能做的所有事情。

所以，包含各种数据视图是很有用的。标准化的理想解决方案是在整个企业中达成一致意见；之前的联系机制模式就是这种情况。通常不能达成一致意见有很多的原因，如果是这种情况，通常最好是为某一概念建模提供多种方法。这与为相同数据冗余地建模是不一样的，我们也不推荐为同样的数据多次建模。

这些模式提供了有效的工具，以帮助建立一致的通用结构，比如谁递归关系、角色或状态。正如图9-4所示，企业数据模型也为同类型的数据建模提供了多种建模方案，比如说使用不同层级的模式（同一模式的不同泛化层级）对数据进行分类的方法。

### 在解决方案只能够使用这些模式有哪些优点？

* **可以用于常见需求的质量数据模型。**创建一个企业数据模型是是十分艰难的，使用企业数据模型不用担忧如何创建有效的方式来捕获状态、分类、角色等等。企业数据模型提供了可复用的模式（经过实地测验的），这样数据建模人员不用将精力集中于为不常见的信息需求建模或找出有多少个相关的核心实体。
* **保持企业数据模型的一致性。**不同的数据需求使用相同类型的模式，这样企业数据模型或更加一致。例如在整个企业数据模型中始终使用相同的分类模式，该模型会提供通用的稳定的方法，随着时间的推移人们会越来越信任这样的方案。
* **企业数据建模提供了多种选择方案。**在企业数据模型中，不管你选择哪种方案，我们认为特别重要的是在此上下文中能够为同类型的数据需求提供多种建模选择方案。数据专业人士在决定使用哪种方案应用他们的应用时，只要熟悉了这些模式，就能够权衡出由企业数据模型所提供的这些选择方案的优缺点。
* **为企业数据建模工作节省了时间。**企业数据模型是一项非常耗费时间的工作，我们发现使用这些模式可以大大削减所耗的时间量。

**将这些模式用于解决方案有哪些缺点？**

企业数据模型中使用这些模式的缺点如下：

**首先，为同一类型的数据提供多个可选的模式，可能会引起混淆。**使用多个备选模式的企业数据模型，不管该模型使用了多少种备选模式，起初理解起来可能会非常困难，甚至对经验丰富的数据专业人士也有一定的难度。然而，我们发现随着时间的推移，人们大多习惯了甚至是喜欢这样的主意：能够从一组选项中挑选而不是被迫使用某一种方法为特定类型的数据建模。此外，你可以选择在企业数据模型中不混合使用备选模式。

**在同一数据模型中使用不同层级的模式会导致数据模型不一致。**因为建模人员有很多种选择，所以除非你在整个企业数据模型中使用相同层级的模式，不同数据类型的各种建模方式会引起不一致性。

### 简介

在这一节中， 我们向你介绍了如何使用不同的模式来启动企业数据建模工作。这些模式为创建通用结构提供了备选方案，比如说为角色、递归关系、状态等等提供了多个可供选择的模式。这样数据专业人士便可以将精力集中于独特的数据需求上，比如说企业的产品成本结构或**订单**与**装运**之间的关系。

企业数据模型应当展现能够支持业务需求的企业结构视图。开发企业数据模型的一种可行方案是维护不同的备选模式（用来为相同的数据需求建模）。例如，在企业数据模型中使用两种不同的方式为顾客分类建模：使用了二级分类模式和三级分类模式。这种开发企业数据模型的方案提供了选择项，可以从多个备选模型中挑选一个模式，

因而提供了一些不是太死板的标准，因为不同的应用会有不同的需求。有时应用可以选择实现企业数据模型所提供的两种风格。例如，一个应用对于重要的属性比如说订单日期可以使用一级风格的模型，而对于其他的数据状态类型可以使用三级风格的模型。这与提倡你在备选数据模型结构中冗余地维护相同的数据是不同的，而且我们完全不推荐这种做法。你还可看到在企业数据模型中，有些结构是一致的并且获得整个企业的认可，比如说联系机制。在此情形中，没有必要展现不同的备选模式。

需要适当地使用企业数据模型。将企业数据模型看作一种方法，用来展现获得认可的标准数据模型结构，但是有时它也可以展现备选方案。在这些情形中，数据建模人员可以查看企业数据数据模型，从而选择最能满足应用的方案。

## 数据仓库模型

每个企业对于自己的决策支持和报告需求都具有自己独一无二的需求。一般来说，决策支持环境为阐释趋势、描述性能提供了信息，提供了主要的业务说明以至于企业可以做出明智的决策。很多企业查看他们的数据仓库以提供这些解决方案。问题是“你如何使用这些模式帮你创建、理解以及验证决策支持环境？”

本章的这一节包括两个部分，从这两个部分你可以看到两种最常见的数据仓库建模风格：关系型建模方案和星型模式方案。第一部分的方案是以Bill Inmon所主张的方案为依据，Claudia Imhoff对该方案也做了很大的贡献。第二部分的方案是以Ralph Kimball和Margy Ross所主张的方案为依据。当然，这两种方案都有自己的优点和缺点，也都有很多的支持者和批评者。我们在构建数据仓库时，成功地使用了这两种方案，我们认为方案的使用很大程度取决于目标和情形。我们还认为这些模式对这两种方案都有帮助，所以在接下来的两个部分，将介绍如何在不同的数据仓库类型中使用这些模式。

### 这个模型的使用场景

金沙经销商依据图9-3中的应用数据模型成功实现了顾客信息的应用数据仓库。它认识到在不同的应用需要标准化和一致性，所以它创建了图9-4所示的企业数据模型。它更深入地认识到它可以将企业数据模型应用到它的下一个工作中，也就是数据仓库中。企业了解到将企业数据模型作为数据仓库的基础是业内公认的惯例。

顾客信息应用以及诸多其他的应用都需要运行报告和决策报告，随着很多不同业务领域对顾客的报告和分析要求越来越多，应用开发人员变得不堪重负。“什么样的产品最受欢迎、最受谁欢迎，在哪些地点以及在什么时候最受欢迎？”“哪类顾客会买哪种类型的产品？”

这是他们需要回答的最常见的两个问题。企业认为其决策支持需求【批注10】缺少信息技术部门的响应。高级IT和业务管理人员要求数据团队提出一个解决方案，来解决不断增长的业务决策支持需求。

数据团队决定为业务决策支持需求提供两种解决方案。一种方案是基于核心数据仓库模型的关系方法（也就是‘Inmon’方法）（尽管在这个方案中星型模式有被用于数据集市），另一个方案是基于核心数据仓库模型的星型风格的模型（‘Kimball 方法’），在数据市场中也采用这种模式。

数据仓库应用的一个主要来源是顾客信息应用数据模型（如图9-3所示）。物理数据库是基于这个数据模型创建的，在此模型中每个实体或子类可以直接转化成关系表。其他数据输入有订单录入系统，在该系统中企业维护的是顾客订单和以下实体：顾客、订单、订单项、产品和产品类别，并且他们的模型（和数据库设计）与图9-4所示的企业数据模型非常相似，具有同类型的实体、属性和关系。

除了图9-3所示的应用数据模型和订单录入系统外，这个模型的其他输入系统对于企业和IT人士来说是全新的决策支持需求，这些需求与焦点小组访谈、业务与IT人士一对一会谈所捕获的需求是一致的。数据专业人士在与业务和IT人士会谈时，发现有些需求超越了顾客信息、订单录入应用和企业数据模型的作用域。新需求如下：

* 业务希望能够根据顾客、顾客类别、顾客状态、产品、产品类别、地理范围、时间或这些因素的任意组合来报告订单的数量和质量。
* 业务希望可以能够使用不同的方式对其产品进行分类。目前，企业是根据产品线、产品类型和产品族对产品进行分类的，但是它知道在将来还会有其他的产品分类方式，比如说“产品价格区间类型”（例如，高价、中等价和低价）和“产品用途类型”（例如，重型和轻型）这样的分类。
* 其他领域的业务认为除了规模、行业和类型外，还有很多的顾客分分类方法。例如，他们想要根据市场领域（例如，石油和天然气公司、提炼公司和勘探公司）和职员数目对顾客进行分类（例如，0-1000、1001-50,000和大于50,000）。
* 他们想要根据地理范围对订单数据进行分类。例如，“我们从哪里可以获得最大的订货量，不同国家关于订单总量的税率是如何规定的？”
* IT高级管理员指明数据团队提出的解决方案要足够灵活、能够满足将来的业务需求（即使业务人员并不能确定将来或有哪些需求），而且还要十分简单，简单的解决方案能让应用团队快速构建报告和分析。

接下来的两个部分介绍了如何在两种不同的方案中使用这些模式，例如：

数据仓库数据模型—关系型方案：这个方案使用企业数据模型作为数据仓库数据模型的基础，然后将数据仓库数据模型用作实现集成数据仓库数据库设计的基础。在将数据集成到非常灵活的数据库结构中之后，然后数据被传递至各个数据集市，通常这些数据集市都是以星型模式设计为基础的。数据仓库数据模型被用来作为设计的基础，该设计可以集成不同来源的数据，而星型模式结构可以应用于大多数的报告。

数据仓库/数据市场数据模型—星型模式：有些人将这个方案成为“总线架构”方案、“星座【批注11】”方案和“一致性维度”方案。这个方案还主张使用企业范围数据仓库结构集成若干来源的数据，然后将数据传递至几个数据市场，数据市场是应用于特定应用报告需求的结构。在这个方案中，数据市场是使用以下几个维度进行分析设计的：设计事实、措施、维度、层级以及各个星型模式的组合（集成星型模式有时被称为“星座”模式）。在企业范围设计中，维度是“一致的”，很多星型模式设计都可以重复使用这些维度。例如，为了维护一致性在订单、装运、支付等事实中使用相同的顾客维度。

**注：在星型模式这部分，我们引用了多维模型的术语和概念，这些术语和概念与本书所介绍的逻辑数据模型概念是十分不同的。我们认为详细地介绍这些术语超出了本书的范围，因为我们的目的只是介绍介绍如何应用这些模式。如果你对多维建模不是很熟悉，请参看多维建模书籍，比如说Ralph和Margy Ross所著的*The Data Warehouse Toolkit*这本书介绍了以上所提到的术语和概念。**

**数据仓库数据模型—关系型方案**

在这部分，我们根据关系型数据建模方案解决了创建数据仓库模型的需求，这个方案有时与“中心辐射”架构相关（Bill Inmon方案所使用的架构）与“总线架构”（Kimball在数据仓库工具集方案中所使用的架构）相反。

之所以选择这个方案，原因之一是金沙经销商认为他的中心数据仓库设计需要使用非常灵活的模型，以至于该设计能够满足未知的需求。它认为星型模式方案涉及的是更加特定的设计，尽管它将其视作更加易于实现的方案，但是它也是不灵活的方案。

虽然这一节的方案并没有将多维方案应用于企业范围的数据仓库结构中，但是其实这个方案并不提倡将星型设计方案（也就是，多维方案）应用到数据市场的后续开发中，数据团队计划使用星型模式设计来开发数据市场，以满足各个组/或应用的特定报告需求。在下一节将会介绍一种技术，如何将这些模式应用于星型模式设计。数据团队使用关系型数据仓库方案制作了9-5所示的模型图。

注：当你使用关系型方案为数据仓库创建决策支持环境时，应当采用以下几个步骤将数据模型转化成物理数据库设计。例如，删除操作数据、添加时间因素、添加派生数据、创建关系构件、适应粒度层级、合并表、基于稳定性的分离【批注12】等等。我们不打算在此深入研究就这些步骤，因为我们在数据模型资源手册的第1卷修订版的第10章节和第11章节（Wiley,2001）已经详细介绍了这些步骤。线面这一节的目的是展现这些模式是如何形成数据模型（用于创建数据仓库的数据模型）开发的基础的。在这一部分，我们将讨论限制于如何使用这些模式没将他们作为数据模型创建工作的一部分，并不研究物理数据库设计（除了在本章节的结尾做了下简短的讨论）因为我们认为很多书都已深入介绍了物理数据库设计考虑因素，所以在此就不用重复介绍了。

此外，在物理数据库设计中数据模型还存在很多的变形，这些变形是基于容量、频率和性能方面的考虑。

### 这些模型是如何工作的？

数据团队决定创建非常灵活的数据模型因为在决策支持环境中数据需求经常会发生变化。为了复用标准的企业建模方法同时使用当前和将来的需求，数据团队决定使用图9-4所示的企业数据模型，剑气作为数据仓库的基础。为了做到这一点，数据图案随产看图9-4所示的企业数据模型所使用的模式，看看他们是如何将这些模式应用于他们的决策支持数据模型的。

在检查企业数据模型和企业所提出的决策支持需求时，他们认为数据仓库需要适应很多的当前以及将来的源系统；因此，他们非常需要灵活性能。他们还认为如果他们可以使用一致的泛化层级维护数据仓库结构，这样非常有助于复用以下例程：向数据仓库存入数据以及从数据仓库取出数据的例程。例如，当事人角色所使用的分类模式产品分类所使用的模式是一样的（也就是，三级分类模式）。当在数据仓库数据模型的各个部分一致的应用相同类型和相同层级的模式时，管理数据的例程也会非常类似，从而更加易于开发管理。

考虑到这一点，他们查看了模型的各个部分，并开始研究任何为声明性角色建立模型。他们决定使用同一类型的数据模型结构，也就是三级声明角色模式结构，如图9-4所示 。这个模式维护了单一的**当事人**，也就是**人员**或**组织**，和该当事人的若干角色，这些角色使用当事人角色来维护，比如说**顾客**。他们认为这是个理想的结构，这个结构能让他们添加满足将来需求的新角类型，并且不用管当事人扮演了多少的角色，该模式可以维护每个当事人的单一实例。他们还认为很多的内部专业人士（和一些外部专业人士）对这个模式很熟悉，所以他们可以在数据仓库工作中复用这方面的专业知识。

随后数据团队注意到企业数据模型既维护了二级递归模式也维护了三级递归模式。这两个模式分别被应用于**顾客**和**当事人角色**。二级递归模式捕获了顾客的组织结构，而三级递归模式通过当事人关系和当事人关系类型能够让任意当事人角色关联值任何其他的当事人角色。例如，总公司顾客可以关联至分公司顾客，这种关系可以通过**当事人关系类型**“总部-分部关系”和**当事人关系**来维护。三级递归模式通过各种关系能够让任何角色（比如说顾客、供应商、合作伙伴等等）关联至另外一个当事人角色。他们注意到新的客户信息应用（数据源之一），使用了二级递归模式来处理他们的顾客层级关系。尽管对于顾客信息系统的有限需求可以使用这个模式，但是他们认为这个模式对于他们的数据仓库来说不足够灵活，不能支持所有数据源的需求。例如，或许其他的应用或外部数据源捕获了其他的其他类型的顾客关系比如说哪个顾客被合并到另外一个顾客当中，哪个顾客被另外一个顾客所收购和/或哪个顾客与另外一个顾客合作。此外，他们认为他们需要能够维护数据仓库的将来需求，以维护不同角色的分层、聚合和点对点关系。例如，他们可以使用相同的结构为以下关系建模：供应商的分层关系、销售人员与顾客的关系、工作人员与顾客的关系等等。

鉴于此，数据团队使用维护当事人关系和当事人关系类型的三级递归模式。这个模式捕获了顾客所有的关联，比如说顾客组织结构。还可以灵活捕获当事人之间的其他关系，比如说，他们的内部组织与合作伙伴的组织之间的关系、顾客联系信息与他们所从事的企业之间的关系或供应商联系信息与供应商之间的关系。

数据仓库团队随后查看了企业数据模型对顾客数据进行分类的方法，因为这是个很重要的决策支持需求。他们注意到企业数据模型提供了两种顾客分类方法。第一种使用了二级分类模式，根据**顾客类型**、**顾客规模**或**行业类型**（一个顾客可以具有多个**行业类型**）进行分类。

另外一个方法是对**顾客**进行分类，也可以对其他的**当事人角色**进行分类，该分类法使用非常灵活的方式：使用了三级分类模式，将**当事人角色**关联至**当事人角色类别分类**（以及其他的分类实体）。

数据仓库团队知道他们已经要求根据很多的分类对顾客进行归类，比如说顾客类型、顾客规模、行业类型、市场细分、职员数目（有外部数据决定）、少数名族顾客状态（人力资源应用需要）、顾客估值（有另外一个应有程序决定）等等。数据团队希望在企业需要的时候可以灵活地添加**当事人角色**的分类（例如，供应商、职员、合作伙伴等当事人的分类），并且不需要添加新的分类实体。

所以，为了使用当前和将来的分类需求。为了提供灵活的数据结构（可以提供无数的顾客分类方法，可以维护分类的无数分层结构，此外还能追踪随着时间推移而发生变更的顾客分类历史记录）他们决定使用三级递归模式，将**当事人角色**关联至**当事人角色类别分类**。

需要注意的是当事人角色所使用的分了模式也适用于产品。这个模式为数据仓库提供了灵活性，可以在需求的时候创建新的产品分类。

因为很多不同类型的分类（例如，产品类型、产品系列、产品族）是在一起维护的（都在**产品类别类型**实体中），因此这种模式提供了一些非常强大的分析功能。例如，团队可以决定在稍后时刻在**产品类别**和**当事人角色类别**之间创建一个交叉引用实体来展现哪类顾客对哪些类型的产品特别感兴趣（顾客的类别可以根据他们所属的**当事人角色类别类型**“顾客类别”来区分）。

数据仓库团队还决定将其他的三级模式应用于数据仓库模型，做出这样的抉择也是出于类似的原因：提供灵活性、适应将来的需求以及维护历史记录。例如，三级上下文角色模式和三级状态模式都是非常灵活的，能够支持添加企业将来所需的新的上下文角色和状态类型，还有当状态和上下文角色在过去的某一时间发生变化时，这些模式可以追踪状态变更的历史记录。

数据仓库团队还注意到企业数据模型要适应如下业务需求可能会不太容易：报告顾客（加上其他的角色）的地点以及将订单关联至不同的地理范围（例如，国家、州、销售区域、小的行政区或区域）。企业数据模型通过**联系机制**将**当事人**和**订单**关联至地理范围，如图9-4所示。拿数据仓库来说，为了满足以上需求是否回答下面几个问题就足够了？比如说，“我们从哪些地方获得了最大的订单量，不同国家关于订单总数税额的规定如何？”回答这个问题，数据团队就要弄清楚业务代表的需求。业务代表表明对于每个订单，衙门都需要了解哪个国家（一个地理范围）与该订单的‘付款’当事人总部相对应。例如，源自XYZ公司的订单，该订单的总部位于伦敦，那么数据仓库中与该订单相关联的**国家**应该是“英国”。

为了支持这些特定的需求，数据团队有两个选择。第一，他们可以使用企业数据模型中的结构，在企业数据模型中，订单有订单通过订单角色与角色类型“付款顾客”相关联，该角色类型与一个当事人相关，该**当事人**的**联系机制**类型—“邮政地址”维护了地点类型“总部”，而该地点是与**地理范围**相关联的。他们认为这个模型的开发太过复杂了，与IT人士所提出的需求背道而驰：数据仓库的开发要十分简单，这样应用团队才能快速构建报告和分析。第二，他们在地理方位和订单之间创建了一种关系，在该关系中“订单只能源自于一个地理范围，而一个地理范围可以是一个或多个订单的地点。”他们决定使用后一种方案，如图9-5所示。**订单**和**地理范围**之间的关系可能与整个企业都相关，或者它只是数据仓库的一个特定需求。数据图团队决定将这个关系作为企业数据模型的基数来调研。值得一提的是数据仓库解决方案大部分都是以企业数据模型为基础的，但是如果数据仓库的需求也适用于企业的其他组成部分，企业数据模型也可以根据数据那些需求做相应的更新。

总之，数据团队使用了非常灵活的数据模型，试图支持将来的业务需求以及维护关系的变更历史。事实上，他们还决定在此数据仓库数据模型中使用相同层级的泛化模式，例如，统一使用三级模式。为了做到这一点他们认为需要放弃一些实现的简洁性，但是实际上，由于这些模式是非常一致的，所以存在一个很大的好处就是：可以一致地管理数据仓库中的数据和复用例程。例如，他们发现支持顾客信息的灵活分类的那部分数据模型所使用的模式与产品分类（或者是任何其他分类）所使用的模式是一样的。这意味着支持产品分类的例程和代码是可以复用的，只要稍作修改便能支持任何分类。他们还注意到在顾客应用中有些被复用的架构具有非常有用的代码和相关的专业技术。最后，基于企业数据模型开发数据仓库是众所周知的‘最佳实践’，看起来十分的自然直观。有些数据团队参与了应用和企业数据模型工作，并建立了知识和专业的技能，这些技术很容易转化到数据仓库模型中。这个模型，除了能够复用同类型的模型的代码外，还可以简化数据仓库的开发。

### 为什么我们要使用这个方法来建模？

正如我们之前所述，开发企业数据仓库有两个常见的方法和方法论。在此场景中，数据团队想要展现数据仓库数据模型的关系型方案（与使用星型模式的方案相反）。他们认为这个方案可能适合金沙经销商因为金沙经销商是个动态的环境，需要更加灵活的模型，此外，还因为在企业范围的多维模型开发中它没有使用专门的知识。

星型模式有很多的优点，当然也存在些许缺点，缺点之一便是一旦你创建了星型模式，会更加难以修改。换句话说就是，如果你想添加新的产品分类或新的上下文角色比如说产品安装人员，那么模型就需要添加新属性。添加新属性意味着重构数据仓库维度表和重要工作—变更萃取、转置和加载（ETL）例程。

在本部分介绍使用此方案的另外一个原因是该方案能让你维护非常全面的有助于分析的数据类型。例如，使用这个类型的数据模型，你可以全面维护所有的分类类型，包括源自外部源系统的分类，然后对这些分类做比较分析，例如，哪类顾客对哪些类型的产品感兴趣和/或这些顾客分别位于哪些地理范围。

数据团队想要提供关系型数据仓库的解决方案的另外一个原因是复用内部的专业技术。应用开发团队已经围绕顾客应用系统构建了代码和性能。因为数据团队重复使用相同类型的模式，在企业的其他部分也使用了这些模式，所以对他们而言开发图9-5所示的数据仓库数据模型并不是专业技术的一次飞跃。然而，在数据仓库的开发中，我们不推荐使用相同的数据结构，就如源系统的开发一样，因为为了适应数据仓库的当前和将来的需求，创建灵活的强健数据模型结构更为重要。

### 解决方案使用这些模式有哪些优势？

将这些模式应用于关系型数据仓库数据模型中会有如下几个优点：

**更加泛化的模型为决策支持数据的维护提供了非常灵活的解决方案。**例如，如果数据仓库需要添加新的产品分类方法，只需要添加一个新的产品类别类型实例和状态类型实例，不需要变更底层的数据结构。

**一致性**。在整个数据仓库数据模型中使用像层级的模式（三级），这样做可以一致地查看管理数据并且可以复用将数据移入和移出仓库的例程，这样还可以简化源系统与数据仓库之间的映射和ETL流程。因此，如果企业能够标准化地使用这些模式，将能简化系统间的交流、削减维护成本并且可以让企业利用管理数据的常见例程。然而，在数据仓库的设计中，我们要提醒各位不要为了简化ETL程序默认同类型的结构作为源系统的基础，因为创建一个坚实稳定的数据仓库数据模型和设计更加地重要。

**捕获历史记录。**数据仓库的一个关键的需求就死追踪历史记录。例如，一个顾客可以归类于“小型”企业，但是随着时间的推移他们将分类变成了“中型”规模的组织。在很多较为泛化的模式中关联实体通过起始日期和终止日期这两个属性捕获了多数的数据仓库历史记录。例如，在三级分类模式中，当事人角色类别分类实体可以维护以下两种情况：在当事人是小型企业时以及后来他们变成了中型规模的企业。其他的三级分类模式也使用了类似的方式捕获变更历史。

### 将这些模式应用于解决方案会有哪些缺陷？

将这些模式应用于关系型数据仓库数据模型中会存在以下几个缺点：

* **使用更加泛化的模式所建成的模型会更加复杂。当你使用泛化模式时，一般情况下你不应当向业务代表展示这样的数据仓库模型。**即使是经验丰富的数据专业人士，但是如果他们不理解这些模式和/或他们偏向于构建特定风格的模型，那么他们会发现基于这种模式的模型很难理解。
* **这些模式不能满足数据仓库的所有历史需求。**尽管这些模式通过关联实体中的起始日期和终止日期属性可以满足数据仓库的以下历史需求，但是还有其他方面的历史没被模式所捕获，例如，有关属性值何时发生变更的历史记录，比如说订单项数量发生变更的时间。数据仓库也需要满足这样的需求。我们不能在通用模式中解决所有适应历史的方法，因为通常他们被认为是物理数据库设计考虑因素（例如，快照表）。

### 简介

在这部分，我们介绍了如何使用这些模式创建关系型数据仓库数据模型。这个方案是由三级模式构成的，在数据仓库数据模型中一致地使用三级模式。这个方案产生了非常灵活的强健的数据模型结构，可以支持数据仓库的诸多数据需求，比如说能够满足很多源系统的需求，还能满足很多新数据需求并且不用变更数据模型。具有灵活的结构，比如说基于三级模式的那些结构，这样的灵活性能够让数据仓库经得起新数据源所产生的影响（而不需要改变底层的结构）。

数据仓库结构使用了企业数据模型中的很多数据模型结构，因此有助于企业使用更加一致的数据模型。在顾客信息应用中数据模型也存在一些类似的方面，这个方案能够让金沙经销商复用创建构件的例程和专业技术，应用系统可以使用构件为数据仓库创建ETL和项目。然而，尽管将类似的数据模型结构应用到各源系统有一些益处，但是首先要考虑的应当是开发一个坚实稳定的模型，要能满足当前以及将来的需求，该模型通常看起来与大多数源系统的数据结构是很不一样的。

数据仓库模型是十分复杂的模型，对于那些不熟悉泛化数据模型或模式的人来说，这样的模型是很难理解的。此外，基于这类数据模型的数据库，加载和/或输出报告会更加困难，不像我们将在下一节中所讨论的星型模式。但是，此方案中的数据模型用途是作为创建集中性的、灵活的数据结构的基础，该结构能够维护很多源系统的数据。将这些数据一起加入强健的、灵活的数据仓库结构中后，然后你可以根据星型模式设计将数据分配到数据市场中。

## 数据仓库/数据市场数据模型—星型模式

在此方案中，我们依据前一节的场景解决了同样的数据需求（在本章节的数据仓库小节的开端所介绍的场景）。然而，在这个场景中，数据团队使用星型模式方案开发数据仓库。

数据团队想要一个实用的简单方案，用来建模和设计数据仓库，他们知道很多企业都已成功使用星新模式、多维方案开发了企业范围的数据仓库。

他们了解到这个方案既可以用来开发企业范围数据仓库模型也可以用来开发数据市场模型。此方案大大简化了企业范围数据仓库与数据市场之间的数据同步和数据传输，因为此方案使用了同类型的模型、设计以及两者的实现。

正如我们所说，在关系型数据仓库方案中（Inmon方案）也可以将下面这个方案用于开发由企业范围数据仓库所提供的数据市场。

### 这些模型是如何运作的？

在开发数据仓库应用的模型时，数据团队要做的第一件事是分析业务需求。根据这些需求，他们意识到这个报告应用的核心方面是能够以不同的方式报告顾客订单。他们了解到将会有一个以订单项金额和数量为基础的事实表，因此，他们知道了将存在一个以订单项为基础的事实，该实体维护了订单项金额和订单项数量，将这两个属性作为度量，该事实将会有若干个关键维度，比如说顾客、产品、地理范围和时间等等。

那么为什么不直接从这里创建星型模式？企业所需的每个维度都存在很多的复杂性。例如，可能会有无数的产品类别和产品类别类型。可能会存在很多不同类型的范围。可能会存在很多不同类型的分类，而每个分类又有子分类。因此，有很多种方式可以用来设计星型模式的维度。尽管通常使用关系型模型不太提倡使用星型模式方案，但是我们认为在做任何类型的物理设计之前，至关重要的是理解数据的属性。因为星型模式是一个报告的物理设计，我强烈建议理解数据间的复杂关系和本书中的模式，这样有助于开发一个可以帮助你理解数据属性的数据模型。

因此，数据团队创建了如图9-6所示的数据模型，作为星型模式的前奏。这个模型使用了很多同类型的模式（在图9-4中企业数据模型和图9-3中顾客信息应用数据模型所展示的模式）。数据团队将这些模型和企业所提出的新需求作为星型模式设计的基础。换而言之，他们要自问下面几个问题，“开发一个坚实的数据仓库设计（使用星型模式方案），有哪些数据需求，支持数据需求需要哪些事实、度量和维度，我们可以复用哪些模式和和模型来理解数据的属性？”。

根据若干维度报告订单金额和订单数量，比如说根据顾客、产品地理范围等维度报告，团队会很容易识别出有一个一订单项实体为基础的事实表，因而他们在数据模型中添加了**订单项**（**事实**）实体，表明这个实体是候选事实表，该表中有订单项数量和订单项金额属性，这两属性可以作为候选衡量项。

因为业务需求是根据顾客、顾客类别、顾客状态、产品、产品类别、地理范围和时间来报告这个数据，数据团队识别数主要的维度，如顾客、产品、地理范围、订单和时间。将订单项关联至这些实体，他们可以推断出其他的需求；例如，如果他们知道了订单项的顾客，那么通过将顾客关联至适当的状态和/或类别，他们便可以查出顾客状态或顾客类别。

然后下一个问题是，“我们应当如何为以下这些维度数据建模：**顾客**、**产品**、**地理范围**、**订单**和**时间**？”这些实体有很多种键模型方法，并且可以将这些实体设计成满足他们需求的模型。他们可以被建立成特定模型与泛化模型，可以有若个个与订单项相关的维度，比如说每类顾客或产品类别的独特维度。为了决定如何创建最适合的维度，他们决定先使用数据模型模式为每个维度建模。他们还将决策支持需求同应用数据模型和企业数据模型中所使用的模式相比较。在他们探索决策支持应用需求时他们意识到他们需要考虑以下几个方面：

* 业务代表表示需要很多种产品分类方式，比如说根据产品类型、产品系列、产品族、产品价格区间和产品用途来分类。他们还表示在将来可能还需要新类型的产品类别。除此外业务代表还表明产品可以同时归类于若干个类别。出于以上原因，数据团队对产品维度应用了三级分类模式。
* 企业需要灵活的方式对顾客进行分类。业务代表认为除了图9-3所示的规模、行业和类型外，还有很多其他的顾客分类方式。需要注意的是在先前的小节中，企业确定顾客只能根据**顾客类型**、**顾客规模**和**顾客行业**来对顾客进行分类。这表示在你决定使用哪一层级的模式要谨慎小心。然而，这些模式允许你使用更加灵活的三级模式来替换较为特定的二级模式，正如图9-6所示，使用三级分类模式。
* 尽管起初的决策支持应用要求更具城市、州（或区域）和国家汇报订单，在将来会需要根据任何地理范围汇报，比如说省、地区、地级行政区、小行政区等等。因此，为了更好地理解地理范围维度，数据团队决定使用维护地理范围的联系机制模式。

因此，数据团队创建了图9-6。这模型的用途是作为星型模式设计的前奏，允许数据专业人士先行理解数据的属性。换而言之，不是将它作为实现模型，而是获得类似于星型模式设计的方法，在该模型中你可以分析事实和关键的维度，因此，这个模型允许你查看的主要数据需求和复杂需求，比如说复杂的分类需求。根据这个模型，然后你可以选择实现模型，该模型会包括精确的事实、度量、维度和层级。讨论完9-6所示的模型后，我们将会讨论以图9-6为基础的模型，也是真正的星型模式。

在创建初始模型时，数据团队要记住两点：

第一，星型模式的每个维度都有很多相关的复杂性，比如说分层关系、诸多的状态、各种分类等等。因此，他们需要一种查看这样复杂性的方法，便于他们在开发星型模式之前解决它。

数据团队所使用的另一个主意是应用合适的模式，可以更好地理解维度的复杂性。描述这些维度（例如，产品、顾客、地理范围）的模式要么已经为图9-4所示的企业数据模型所捕获要么为图9-3所示的应用系统所捕获，或者将他们作为未被当前数据模型所捕获的新需求。作为一个已被应用系统捕获的需求示例，图9-6所示的维度**顾客**保留了相关的**顾客状态**和**状态类型**模式，在图9-3所示的应用数据模型中可以看多这些模式。然而，你可以从图9-6中看出一个**顾客**可以具有多个**顾客状态**。

这样会出现一些有意思的问题，例如，当我们根据顾客的状态报告订单时，该订单是不是有一个相关的主要状态，如果有的话，哪个顾客状态会与订单相关联呢？还会出现下面这问题：是否我们只会关注下单时、订单关闭时的顾客状态或者是多次出现的顾客状态出现多次时。主题专家表示需要捕获整个订单生命周期内的所有顾客状态。例如，某一订单的顾客可以具有“激活”或“批准信用申请”这两个状态类型。在报告订单时这两个顾客状态是否都有效？主题专家的回答是“yes”。如果数据团队没要看出顾客、顾客状态和状态类型之间底层关系的复杂性，那么他们可能就不知道问主题专家这个问题。因此，因此，他们有可能会做一个错误的假设，比如说只使用一个顾客状态（比如说最新的状态），而实际上，订单生命周期内的所有顾客状态都是可应用的。

数据团队注意到基于图9-3所示的顾客应用程序中使用了二级分类模式捕获**顾客类型**、**顾客规模**和**顾客行业**。但是企业认为除了根据规模、行业和类型对顾客进行分类外，还有很多其他的分类方法。此外，基于图9-4的企业数据模型展现了该模型既可以使用二级分类模式也可以使用三级分类模式。因此，需求使得团队使用较为灵活的三级分类模式，将此模式应用至顾客，这样他们可以很容易支持现存的分类以及任何新分类。在图9-6的右上角可以看到该模式。然而，这个模式关于状态也会出现类似的问题：如果一个顾客在某一时刻点同时属于多个类别，那么哪个类别是与订单相关联的呢？换种说法就是，如果你问这样一个问题，“每个顾客类别的总订单量是多少？”，倘若顾客属于多个类别的话，那么报告中相加起来的总数会超过总的订单量，因为更具每个订单类别顾客订单被重复计数了。企业主体专业表示如果只想能够根据顾客类别、了解顾客可以归属于多个类别，那么采用这个模式是没有问题的。

企业想要在订单同顾客聚合顾客所订购的产品类型相关时，捕获该订单的信息。因此，产品是订单项（事实）进行分类的一个维度。然而，产品也有自己的分类。这一点可以使用三级分类模式来解释下，该模式包括**产品类别分类**、**产品类别**和**产品类别类型**。例如，在 “产品系列”、“产品族”和“产品类型”这些产品类别类型内，可以根据类别对产品进行分类。这个模型可以容纳任意数目的位于**产品类别类型**内的**产品类别**实例。企业也可以根据需要添加新的产品类别类型。为什么选择三级模式？IT高级管理层表示无论数据团队提出什么样的方案，该方案必须要十分灵活，要能够满足企业将来的需求（即使企业并不却确定将来会有什么样的需求）。这个模式还有一个问题：如果产品属于多个类别，那么哪个类别是与**订单**相关联的呢？业务代表给出了类似于状态与顾客的回答，如果所订产品属于多个类别，在报告订单项时，这些类别都是可以应用的。

在图9-6中，**订单**与**订单项**（事实）相关，而每个**订单项**与**产品**相关。 在没有理解订单必须是由订单项组成的这一底层数据模型结构时，该结构是专门针对单一产品的，设计人员可能在设计星型模式上犯错误，比如说有关**订单项**（事实）所需粒度程度的错误。例如，如果数据团队将订单金额属性放在**订单项**（事实）实体中，该属性表示一个订单的所有订单项金额的总和，它可能会导致混淆，因为只有某一个产品是与订单项相关的。

为了支持业务需求，还给图9-6添加了另外两个维度。第一，添加了地理范围，这样企业可以根据订单的装运地址对订单数据进行分类。例如，你可能会需要回答下面这样的问题，“这段时间， 运送产品最多的前三个国家是？”数据团队使用维护地理范围的联系机制模式（本书中的第七章节所示）作为这个模型的基础。这个数据模型结构允许记录不同的国际地理范围比如说**省**、**州**、**行政区**、**领域**、**国家**会任何可行的地理区域。该结构展示了不同的地理范围是如何相互关联的，以至于业务带表可以上下掘取国际地址范围，来做不同详细程度的数据分析。

数据团队添加的另外一个维度是**时间**维度。很多出版物都全面地介绍了这个维度。简而言之，**时间**维度提供了一种可以记录时间帧的方法，然后向上或向下掘取更加详细或更加广阔的视图。在此案例中，时间维度指的是订单项的日以及与每天所对应的周、月或年。这样可以报告任意数据的天、周、月或年内的订单金额和数量。在图9-6所示的数据模型中还有其他实体存在时间属性比如说顾客状态（状态起始日期、状态终止日期、起始日期和终止日期），而**顾客类别分类**、**产品类别分类**和**地理范围关联**实体都有起始日期属性和终止日期属性，这些属性维护这些关联的有效性。数据团队提供这些日期属性是为了跟企业强调对于顾客状态、顾客分类、产品分类和地理范围关联的变更，他们需要捕获决策支持日期。对于如何在构建星型模式时捕获时间，这个问题看似很繁琐，实际通常也是最颇具争议的问题。通常在“缓慢变更的维度”这一专题下解决这个问题；但是，这些模式能够帮助确定所需的是何种历史记录。

**注：我们在图9-6所示的模型中为泛化的时间维度建模，我们可以选择将该模型命名为“日期时间”。在很多的模型中，我们使用多个时间维度，每个维度都会具有不同的层级以适应不同的粒度。因此，根据情形和建模风格，可以有以下时间维度：日期时间维度、周时间维度（只有周、月和年）和/或月时间维度（只有月和年）。数据模型资源手册修订版第1卷提供了时间维度的变异形。**

在创建星型模式之前我们可以将图9-6看成是理解数据属性的一块垫脚石。很多三维建模人员略过这种工作直接根据需求创建星型模式和源系统数据库。如果你有经验（很多的三维数据建模人员都具有），你可能会十分有自信略过这种建模前奏性的工作。但是，我们发现无论经验如何，先建立数据模型都可以节省大量的时间和避免很多潜在的错误，然后使用这些模式来强调并理解潜在的复杂性，这样你可以恰当地解决这些问题。

星型模式是一个类似于星形的实现结构，该结构有一个单一的事实表（‘星’的中心）和几个维度表（星的角），维度表与相关的事实表之间是一对多关系。图9-7所示的星型模式就是物理表的样式，如果实现了该模式，这些将是存储事实、度量、维度和层级的直接输入项。星型模式解决了之前实体关系间的复杂关系，以至于程序员为了创建有用报告而写复杂的连接。它是解决复杂性的数据专业人士的工作。为了完成这个工作，数据专业人士在解决这个问题之前需要查看并理解复杂度，因此，我们发现在直接涉及星型模式之前使用这些模式查看这个复杂度是很有用的。我们参与到星型模式开发工作中，只用在做完大量测试后，团队才意识到他们对于数据关联方式存在根本性的误解。

对于创建十分简单而迅速数据分析以及报告应用，星型模式是非常常见并且很有用的技术。此外，很多的OLAP和多维分析工具都是专门为处理星型模式而设计的，这些技术要求设计人员考虑到某一维度相应的事实（中心表）、度量（事实表的度量标准）、维度（划分、切割以及拦截数据的方法）和层级（某一维度所采用的层级），比如说**产品**是在**产品类别**内，这两个实体可以表示**产品**维度的层级。

在图9-7中，你可以看到数据团队是如何将图9-6转化成星型模式结构的以及如何设计各个维度的。你应当注意到的一个关键点是由于维度是非常特定的，所以他们看起来很类似于一级或二级模式。例如，**顾客**、**顾客类别**、**产品类别**和**地理范围**维度看起来就非常类似于一级模式。顾客维度类似于一级声明角色模式，因为付款是特定的独立于其他角色之外的（也可以将这个维度看车是非常基础的特定的递归结构因为在这个维度中有不同层级的顾客组织）。**顾客类别**类似于一级分类模式因为每个类别是以在自己的层级（属性）上维护的。地理范围类似于一级联系机制模式，因为城市、州和国家分别是该维度中的不同层级（我们将术语“层级”用作通用的星型模式术语，与本书中所讨论的其他类型的层级的具有不同的含义）。顾客状态类似于二级状态模式，创建星型模式通常涉及到‘扁平’较为泛化的模式而使用更加特定的建模方式。

现在是时候仔细查看下图9-7所示的**顾客**、**顾客类别**和**顾客状态**维度。在图9-6中，有一个顾客实体，该实体展示了顾客可以进一步被分解成其他顾客，以此类推这些顾客也可以再次被分解成其他的顾客，这样维护可一个顾客组织的层级结构。这个数据模型表明顾客在他们的分层结构中有任意数据的层级。在星型模式设计中，因为一个维度只能指明层级，不能指定任何其他相关的维度，所以团队决定使用四个层级的分层结构，可以容纳总公司名称、顾客名称、顾客分支机构名称和顾客部门名称，因为团队认为这样的结构能够满足将来的需求。这里要强调的是如果在分层结构还需要其他的层级（或类型）

，将会需要更改星型模式。

在图9-6中，**顾客**是与三级分类模式相关的，还有一个**顾客类别分类**实体（以及其他相关的分类实体），该实体能为顾客分类提供灵活的方案。之所以需要这个实体是因为业务代表表示有很多不同类型的顾客类别。在图9-7中，你可以看到一个顾客类别维度，该维度先包含的层级有：顾客规模、顾客类型、顾客行业、市场细分和员工数范围，这个维度可以支持五种分类，这些分类是企业确定需要捕获的（这些属性与图9-6所示的**顾客分类**的实例相对应）。将顾客类别单独作为顾客的一个维度，可以解决下面这个问题，“如果顾客有多个类别，那么哪个类别是与订单相关联的呢？”根据业务代表的说法，这些类别全都与订单项相关联。例如，如果某一订单项的顾客同时属于若干个行业，那么每个行业都会算上该**订单项**。你可以尝试在顾客维度中记录这些**顾客**分类。然而，因为顾客属于若干个类别，所以在同一维度中维护顾客和顾客的分类会很难理解。此外，若这样做了，可能会导致错误的或误解的结果。

图9-6所示的数据模型中，将顾客状态作为三级状态模式捕获的。主题专家表示先前他们希望捕获顾客的若干状态。将顾客状态类型作为一个独立的维度，可以解决下面这个问题，“哪个顾客状态是与**订单**相关联的？”答案是订单生命周期内所有的顾客状态都与之相关。

所以，如果顾客在订单生命周期内具有“激活”和“通过信用审核”状态的话，这两个状态都会被计算。所以当你问处于“激活”状态的订单项总额，然后问处于“信用审核通过”状态的总的订单项总额时，如果某一订单项的顾客具备这两个状态，那么总额可以应用于以上的两个问题。将顾客状态类型单独作为一个维度，而不是将它作为顾客的一个层级的话，这样可以避免潜在的思维混乱：认为顾客只有一种状态。

类似的流程可以应用于**产品**维度，在该流程中，使用了**产品**维度和**产品类别**维度替换了三级分类模式，因为同一个产品分类为多个类别而且还可以使用相同的类别类型对产品进行多次分类（一个产品可以同时具有“重型”和“轻型”这两个产品用途）。因此，不是维护一个单一的**产品**维度，而是将**产品**和**产品类别**分开，作为两个不同的维度，了解产品并不意味着我们必须知道的产品类别（因为有很多类别），通过确定这一点可以避免任何混乱。**产品类别**维护了很多的分类比如说产品类型、产品系列、产品族、产品价格区间和产品用途（每个属性与图9-6所示的**产品类别**实例相对应）。当主题专家被问到，“如果**产品**具有多个类别的话，那么哪个类别是与**订单**相关联的呢？”他们的回答是所有的产品类别都可以使用，数据团队在星型模式设计中考虑到了这个需求。

**时间**维度的数据是使用图9-6中订单的订单日期推断出来的，然后将时间维度应用到星型模式设计中，并将事实表中的各个维度关联至时间维度。这样的设计提供了一种根据日期、周、月和年来报告、切分和向上钻取订单数据的方法。

地理范围维度以扁平层次展示了，图9-6中的每个子类、父类结构都使用了图9-7所示的非常特定的维度结构。例如，在此示例中，数据团队在维度中维护了城市、州和国家（以及地理范围代码），可以根据这些层级中的任何一个来报告数据。图9-6中的数据模型展示了将来需要维护国际范围，比如说需要维护领域、省、行政区等等；然而，数据团队决定简化模型，因为只要捕获地理范围维度的城市、州和国家这三个层级就可以满足当前的需求了。

订单事实包含了可应用维度的两个度量。

第一个是订单项数量，数量—例如，数量为“235”订单项是一个实例，它可以应用于特定的维度组合。这个数字是订购产品的数量，它与特定的产品、产品类别、顾客、顾客状态、顾客类别或这些维度的任意组合相关联。

其次是订单项金额，订单项的货币值：例如，“200,000”便是可以应用于特定维度的值。这个星型模式假设只用一个相关货币。如果记录了很多的货币，也可以有一个**货币类型**维度，或者使用若干的度量，一个度量代表一种货币，例如，订单项美元金额、订单项欧元金额等等。

**注：我们在此介绍的是数据仓库模型的子部分。在数据仓库中一般会有其他的星型模式，这些模式整合到星座格式中，他们中的每一个都有可能会被共享，换句话说就是“确定的”维度。例如，同一顾客维度也可以在另外一个星型模式中使用，比如说在顾客支付事实星型模式展示了数据仓库中的顾客支付历史的分析。**

### 为什么我们要使用这个方法？

类似于其他类型的数据模型或数据设计，开发星型模式设计需要数据专业人士理解数据的属性，然后平衡当前需求与将来需求，有些需求对企业或应用开发团队来说可能不是很明显。数据模型模式在以下几个方面可以促进星型模式的开发：

* 为了展示有多少通用的数据会被建模，可以在设计星型模式之前使用这些模式开发一个初始的数据模型。允许数据建模人士从各个模式当中挑选，让数据模型具有适当的灵活性，这样可以提供一些关于有效维度结构的线索。例如，如果有一个严格的分层机构，在该结构中产品是在产品线内的，而产品线是在产品族内，将他们作为层级放在同一个维度内可能会更加合适（‘levels’这个单词在此是星型模式的术语）。如果产品属于多个类别，那么在分层方式不能够灵巧地向上钻取信息，还是可能将他们放在同一个维度中；然而，使用单独分开的产品和产品类别维度可能会更加清晰，因为他们之间存在多对多关系。
* 我们能够从这些模式中得到关于如何构建维度的线索。维度通常类似于一级模式，有时候类似于二级模式（例如，当我们展现顾客状态模式的时候）。为了提供非常简单的报告数据方式，逻辑数据模型的复杂性通常会被‘铲平’。
* 金沙经销商的IT专业人士非常熟悉这些模式和关系型数据模型。将维度作为模式来解释这样的方法有助于他们捕获维度的复杂性，这些模式为他们构建多维模型提供了垫脚石。
* 这些模式能够为企业数据模型（或源系统应用数据模型）和星型模式的决策支持需求之间架一个桥梁。换句话说，在星型模式数据仓库数据模型中（图9-6）表示这些模式是为了阐释需要被理解的复杂度，然后在该星型模式的维度中解决复杂度。

### 将这些模式应用于解决方案有哪些优势？

关于决策支持企业有一些特定的需求，例如，企业认为除了规模、行业和类型外，还存在很多顾客分类方法。此外，需求表明数据团队提出的解决方案要十分灵活，要能够满足企业将来的需求（即使是企业还不能确定在将来会有哪些需求），还要十分简单，要让应用团队能够构建报告和分析，将这些模式应用于星型模式有以下几点优势：

* **越是坚实稳固的星型模式设计，错误就会越少。**在开发星型模式之前可以使用这些模式开发一个数据模型。这样做有助于你在设计星型模式时做出更加明智的抉择，从而开发出更加稳定坚固的设计。当数据没被建模或没被很好的理解时，我们发现可能会发生重大的错误，可能会导致实现星型模式的计划延迟，或者更糟糕的是人们在使用这种模式时错误地理解了数据。
* **一致的设计有助于提供开发维度的一致方法。**使用这些模式，你可以开发出更加一致的维度设计方案。例如，在一个维度内的顾客和产品分类层级可以是相似的，可以使用一级分类模式作为任何与分类相关的维度的指南。你可以使用这些模式将起初看起来复杂的三级模式转化成类似于一级或二级模式的结构。因此，可以先理解相关数据的复杂度，然后通过创建一个简单的（扁平的）维度表来解决复杂性，同时要理解这样做的潜在利弊。
* **更好地理解数据。**如果你在设计星型模式之前，没有开发一个数据模型，你可能会误解数据关联的方式，这些模式有助于更好地理解数据的特性。例如，如果你只是将产品类型和其他的产品分类作为产品为度的层级而没有理解产品数据分类的复杂性，可能会造成不可预料的、误解的星型模式。如果你使用模式创建产品类别分类数据模型结构，比如说图9-6所示的模型，你可以看到产品同时可以属于多个分类并且这些分类集使用不同的方式向上钻取产品分类信息。这些模式有助于星型模式的设计和使用。

### 将这些模式应用于解决方案的缺点？

在星型模式中使用这些模式有如下几个缺陷：

* **使用数据模型作为星型模式的基础，有时不能反应真实的数据关系。**例如，图9-6所示的数据模型展现了**订单项**与**顾客**的关系，而实际上是**订单**到**顾客**的关系。这样做是因为我们假设订单项是事实表的基础，并且我们知道订单项（事实）与顾客（“付款”当事人）之间存在关系。但是，在这个方案中，我们是在做一些假设，比如说假设事实表会是什么，所以我们不一定模拟了关系的真实特性。
* **有些人认为使用这些模式开发关系型数据模型是多此一举，星型模式开发不一定要经过这一步。**很多星型模式设计人员他们可以使用多维分析方式来理解数据特性而关系型数据模型是没必要的多余的步骤。

### 简介

在这部分，介绍了数据团队如何使用不同层级的模式创建数据模型，这些模型可以作为星型模式的基础。一个数据模型是星型模式的前奏，可以帮助你更好地理解数据需求。在使用这个方案时，你可以使用本书中的模式开发数据模型，用来展现各种需求所要求的恰当的灵活性。

在开发一个星型模式之前先使用恰当的模式为需求建模，通过这样的方式在需求和星型模式之间搭建一个桥梁。

有些经验丰富的多维数据建模人员可能会觉得在开发星型模式之前开发一个前奏型的数据模型是没有必要的。但是我们发现无论你是经验丰富的建模人员还是没什么经验的建模人员，为数据需求建立模型并理解这些需求非常有利于有效设计的开发，包括星型模式的设计。

## 主数据管理

David Friedland在著作*The Data Administrator Newsletter*中写到，“良好的主数据管理（MDM）需要保持这个数据被清理，并且需要在facet数据【批注13】的相关分类中标准化主数据模型。”主数据管理支持数据的集成，还支持整个企业的引用和主数据的分发。在这一节，我们使用模式为顾客数据创建一个标准的主数据管理数据模型。

### 此模型的场景

金沙经销商变成了中东大型企业集团恶意收购的目标。在避开恶意收购之后，CEO决定给他所有的顾客写一封信，解释下情形并感谢他们一直以来的支持。他把这封信递给了CIO，并要求CIO将这封信邮寄给每位顾客。CIO要求每个应用系统的负责人提供明确的顾客清单。结果他发现账单系统有一份清单，顾客信息应用有另外一份清单，而总分类账支付系统还用一份清单。而且，这些顾客清单有重复和不一致的地方！“什么样的公司居然不了解自己的顾客？”CEO 愤怒地咆哮着。

考虑到这样的情形，CEO召集他享誉甚高的数据团队，要求他们针对这样的顾客主数据问题给出一个解决方案。数据团队向CEO介绍了顾客数据问题的本质：

* 首先，他们告诉CEO顾客数据不是存储在一个系统当中；也就是说，他们没有OTOB（一个系统处理一件事）。
* 在不同的系统中存储顾客数据的结构是不同的，每个系统都采用了不同的语义和术语来描述各个客户信息块。例如，顾客应用系统捕获了一些联信息，而总分类账系统则捕获一些支付地址信息但没有捕获电话号码。
* 存储在每个系统中的数据是不一致的。即使是同一条数据也是如此，例如，顾客总公司的邮政地址，不同的系统通常捕获的地址也是不同的。
* 每个维护顾客数据的应用系统管理员都人为自己负责的系统应当是核心顾客数据源（也就是，记录系统）。
* 将这些模式应用到他们的数据模型中，他们就开始朝着更加一致的数据和更加一致的语义的方向前进。他们使用了这些模式来开发一个企业数据模型、顾客信息系统和数据仓库应用。但是，企业有几十个使用顾客数据的应用，并且尽管在维护更加一致的数据方面有所进步，但是在各个系统中仍然存在大量的不一致的数据和各不相同的语义。

数据团队的负责人向CEO提供了以下几种解决方案：

* **清除混乱数据。**换句话说就是，启动数据清理计划，以确保所用系统中的数据是一致的。每个应用组需要为清理系统数据提供一些资源，为了配合这些工作，企业需要开始实施一个数据清理流程。为了确保数据是清晰同步的，系统检查需要遵循这个流程。这些措施将会耗费时间和资源。此外，然后这项工作会被拓展成一个数据质量项目，这样不仅能清理数据，而且制定了一项可持续的项目，包括开发测量流程和不断地提高数据质量。
* **创建一个全新的主数据管理（MDM）数据库，该数据库集成标准化了所有顾客信息的主数据。**主数据管理系统将配合所有的系统，表转化所有的系统，并从主数据源中迁移出顾客数据，以达到维持顾客一致性的目的。作为此工作的一部分，企业认为各种情形下的数据 应当是顾客数据的主要来源（记录系统）。这项工作需要投入新的技术和重要的资源，包括时间和金钱。
* **开发数据管理项目。**这个项目将建立主管部门、规则、政策、决定权、流程、标准、职责和数据管理工作，所有的这些都有助于更加有效地管理数据资产。
* **完成以上所述工作。**清理数据（以及开发数据质量项目）、创建一个全新的主数据管理数据库和制定一个数据管理项目。从短期来说数据清理是有帮助的，有助于在主数据管理应用启动同步集成各系统数据之前设置数据。每个现存的系统都将会确定主数据源存在地，规则将会为恰当的匹配所确定，为恰当地匹配、巩固和同步数据，将会确定一些规则，然后他们会开发（和/或）购买一项管理主数据（比如说顾客数据）的技术。企业也可以建立一个数据管理项目来支持主数据管理工作和数据的集成以及一般管理。这个方案在时间、经理和成本上的投资最大。

**注：当然，为了解决这个问题，金沙经销商还评估了很多其他的解决方案，比如说成熟的数据质量项目、实现一个元数据管理项目、重建控制数据的业务流程以及诸多其他解决方案。这个实例的目的是在初步评估之后被数据组发现的。**

数据组为每个备选方案制作了一份草拟的项目计划，初步规划了每个项目的时间和金钱。根据数据组的分析，CEO和CIO，作为运营委员的成员，决定采用最后一种方案也是最昂贵的方案。

**注：这个解决方案最重要的原理是金沙经销商意识到数据是一笔资产，所有的资产都需要加以管理。他们认为这个解决方案是管理这笔数据资产最好的方法。它还人道MDM的关键驱动因素是服从问题、兼并与收购、服务性架构、更好的销售以及金沙顾客所需的服务。最后，它决定想要成为一个善于分析的竞争者，要做到这点，它认识到当需要保证数据质量时，MDM为维护有质量的数据提供了非常坚实的基础。**

数据团队将精力集中于MDM解决方案第一次迭代的主要问题上。主要问题之一是如何为那些参与业务的顾客和其他当事人提供完整的视图。他们还意识到还用很多复杂的包装解决方案，可以用与身份认证管理和匹配、业务规则、决定性的系统记录和主数据管理的其他方面。他们知道从头开发实现这个MDM解决方案是一项非常困难的工作，所以他们决定使用数据模型作为同潜在的MDM解决方案提供商沟通数据需求的一种方式。团队想要从数据模型的小处着手大处着眼，所以他们起初将焦点放在顾客主数据的非常小的子集上。这包括捕获顾客子集的语义、控制层级的描述以及金沙经销商管理顾客数据分类所需的灵活性。他们决定首先将精力集中于以下几个方面的数据：

顾客的状态数据

顾客与其他人之间的关系，比如说与总公司/分公司之间的关系。

顾客的分类

顾客的各种联系方式，如何关联这些联系方式，特别是如何关联世界各地的地址。

与各科相关的业务规则，例如，规定决定哪个系统是记录系统的因素或规定产生顾客评估结果的因素（也就是，评估一个顾客对金沙经销商而言的价值有多大）。

主数据管理一个重要的方面是身份认证管理，有些议论担心是肺炎应当将这个方面作为数据建模工作的一部分。身份认证管理专注于查看相同数据的多个来源并决定从哪里引用相同的数据实例。例如，如果数据库中存在一条记录，比如说“Steve Jones”的家位于“科罗拉多丹佛的101号大街”，当另外一个数据库中存在另外一条记录，这是指同一实例吗？比如说“Steve Jones”的家位于“科罗拉多丹佛123号大街”。或者以上所说的情况指的是两个人人，一个人叫“Steve Jones”，而另外一个也叫“Steve Jones”，并且他们居住在同一条大街？身份认证管理其实是有一些策略和方法的，但是数据团队决定此时不在数据模型中解决这方面问题，因为这个问题需要交叉引用源系统以及指定记录系统（哪个系统将处理基本数据记录），他们认为可以在元数据模型中捕获这样的信息。没有将这个问题作为数据模型的一部分来处理的另外一个原因是：有很多要考虑的设计要素，而这些要素影响这个交叉引用的处理方式。（例如，主数据管理方案通常有广泛的处理身份认证管理的方案）。

### 这个模型是如何运作的？

顾客主数据解决方案需要集成同步以下的数据类型：

* 顾客的各种状态、这些顾客状态的分类以及管理描述这些分类组的结构和规则。例如，“激活”、“未激活”、“待定”、“信用审核通过”和“调查中”这些都是可能的状态，而“激活状态”可以被进一步分解成其他的状态比如说“高度活跃”和“低度活跃”【批注14】。
* 顾客与各当事人之间的关系，包括顾客组织机构各个成分之间（顾客的内部结构）所存在的关系、与顾客企业相关的联系方式、销售人员与顾客联络员之间的关系以及顾客与其他当事人之间的关系。
* 各种顾客分类方式，比如说根据行业、规模、职员数量和/或现存应用或将来应用所需的任何其他分类，包括外部当事人所提供的数据。此外，由于这类数据是采用不同系统中的各种各样的格式来维护的，所以为了同步集成数据，需要能够将这些不同的数据格式映射到通用格式。
* 一个顾客可以具有不同的联系机制，比如说电话号码、手机号码、电子邮件、邮政地址和/或任何其他类型的联系信息。由于金沙经销商的各科遍布世界各地，所以地址信息必须适应国际需求，这些需求反应了世界各地不同的地址结构。
* 可以应用于主数据管理的各种规则。这些规则包括如何让不同的源系统记录保持一致和可以应用于顾客或其他当事人角色的规则。

所以，为了支持顾客主数据管理的这些需求，我们将介绍数据团队是如金额用用更加泛化的模式来为以上数据类型建模的，例如，他们如何为以下数据类型建模：

* 顾客状态
* 顾客关系
* 顾客分类
* 顾客联系信息（联系机制）
* 顾客业务规则（特别是那些决定记录系统和顾客评估的规则）

为了满足先前所提的主数据管理需求，数据团队开发了图9-8所示的模型。在图中，你可以看到应用了三级状态模式，这个模式可以让**当事人角色**关联到任意数目的**状态类型**，任何状态类型都可以应用至多个当事人角色。例如，顾客主数据涉及到的诸多顾客状态类型，比如说“激活”、“信用审核通过”、“待定”和“调查中”。因此，顾客可以同时具有任意数目的状态，或当事人随着时间的推移可以具有很多的状态类型。这个数据模型结构也适用于其他类型的角色，比如说**供应商**、**合作伙伴**和**工作人员**，因为他们每个人都会涉及到很多的状态类型，状态类型对于主数据管理的其他领域也是非常重要的。

（图）

**图9-8** 主数据管理数据模型

图9-8应用了状态类别模式，该模式可以对状态进行灵活的归类，以便相同的**状态类型**（比如说“激活”和“未激活”）可以被分类成多个**状态类型类别**。例如，当顾客在购买产品时，可以有个**状态类型类别**是“顾客购买状态”，该类别可能是很多**状态类型**（比如说“激活”、“信用审核通过”和“信用审核未通过”）的一个分类。不同的状态类型集合可以用于捕获不同的信用状态。例如，可以有一个**状态类型类别**是“顾客信用审核状态”，该类别可以是**状态类型**“信用审核未通过”、“信用审核待定”或“信用审核通过”的一个分类。需要注意的是相同的状态类型“信用审核通过”和“信用审核未通过”可以根据多个**状态类型类别**进行分类，这个模式通过关联实体**状态类型类别分类**满足了这一需求。此外，在**状态类型类别类型**中可以维护更加泛化的类别类型（也就是，更高层级的类别类型）。例如，刚刚提到的两个类别，“顾客信用审核状态”和“顾客购买状态”，可以在“顾客状态”这一**状态类型类别类型**中被分类。再例如，状态类型类别类型可以是“交易状态”，该**状态类型类别类型**可以包含以下几个**状态类型类别**：“订单状态”、“装运状态”或“发票状态”。该模型还使用了维护多种隶属关系和规则的状态类型模式，该模式维护了状态彼此关联的各种方式和这些状态相互关联的规则。状态可以通过多种方式彼此关联。例如，在层级结构中，一种状态可以是另外一种状态的进一步分解，一种状态可能与另外一种状态时相互排斥的或者是一种状态可以替代另外一种状态。例如，有可能状态层次结构中的某一状态层级结构的最高层级被“列入黑名单”。然后在黑名单状态中你可以有一个状态“因未支付账单被注销而列入黑名单”和另外一个状态“因道德问题列入黑名单”；因此，状态类型关联和相关的实体使用递归模式，这种模式允许你以不同的方式将状态相互关联起来。

团队还认为：关于状态关联的规则是很重要的。例如，可能有一种状态类型需要出现在另外一种状态之前。所以，数据团队决定使用维护多重隶属关系和规则的状态类型模式（该模式是以维护规则的三级递归模式为基础的）。这种模式允许数据团队为不同的状态类型创建不同的分层关系、聚合关系或对等关系。此外，通过使用状态类型关联和状态类型关联规则，他们可以维护不同**状态类型**实例之间的行为规则。因此，在特别的实体**状态类型类别**中维护了两个状态不能同时存在的规则。例如，在实体**状态类型类别**中有一个关于“顾客购买状态”的规则规定顾客不可以在同一时间既处于“激活”状态，又处于“待定”状态。再比如说某一种状态可能暗含了另一种状态。例如，在“装运流程状态”的上下文中顾客处于“等待装运”的状态，那么可以认为该顾客在该上下文中处于“激活”状态。这个解决方案提供了灵活性，并对所有状态类型的引用数据的管理加以控制。

顾客主数据最重要的一方面是顾客的各种关系。这些关系中可以包括顾客与金沙经销商之间的关系，例如，“谁是这位顾客的销售人员？”好可能包含该顾客内部组织的关系，比如说“这位顾客的总公司是？”你或许需要其他的关系有：顾客联络员与顾客组织之间的关系以及销售人员和顾客联络员之间的关系（这个关系不同于销售人员与顾客之间的关系）。这些关系是通过使用维护规则的三级递归模式来捕获的，在模型中使用实体模型**当事人关系**、**当事人关系类型**和**当事人关系规则**来维护。不同的关系可以有不同的类型，在当事人关系类型中可以维护这些关系类型，比如说“顾客联络员与顾客组织之间的关系”、“销售人员与顾客联络员之间的关系”、“上级机构与子机构之间的关系”。不同类型的关系之间可能还存在规则，比如说某些关系是不被允许的活暗含的。例如，某一顾客可能会要求某位特定的销售人员不能使用他们的账户（排斥）。这个规则是**当事人关系规则**的一个示例。

**注：该模式被用来描述当事人关系、当事人关系类型、当事人关系规则、状态类型关联、状态类型关联类型和状态类型关联规则，事实上，将同一模式应用于数据模型中的不同的数据集合，这个模式也就是维护规则的三级递归模式。**

一个企业是如何对其顾客进行分类的是非常重要的 引用数据。大多数报告和决策支持应用使用这些分类，以全面挖掘顾客的各种信息。在业务声明周期的不同阶段，可能会使用很多不同的方式对顾客加以分类，同一企业的各个机构成分可能会根据他们的上下文对顾客做出不一样的分类。例如，本章节的前一部分所介绍的顾客应用数据库，特别是根据**顾客规模**、**顾客类型**和**顾客行业**（如图9-3）对顾客加以分类的部分。顾客主数据应用必须十分灵活以能够处理企业内（外）不同视角下的各种分类。

从图9-8的右边可以看出数据团队决定使用第五章节中的维护规则和模式的三级分类模式来支持顾客主数据的需求。例如，市场不停通过分类模式来细分顾客，在该部门顾客可细分为“核心顾客”、“主要顾客”和“战略性合作伙伴”。“核心顾客”可以定义为某个顾客的损益将会对金沙经销商产生全局性的影响。“主要顾客”指的是其损益会对企业产生局部影响的顾客。“战略性合作伙伴”对金沙经销商而言是最重要的顾客。这些顾客是金沙经销商在任何情况下都不能都丢失的顾客。因此，“核心顾客”、“主要顾客”和“战略性合作伙伴”是当事人角色类别，这些类别属于“顾客细分类型”**当事人角色类别类型**。市场部门还有其他的当事人角色类别，比如说属于“顾客估价”这一当事人角色类别类型的“高价值顾客”和“中等价值顾客”。这两个当事人角色类别类型被认为是“市场顾客分类模式”（**当事人角色类别类型模式**）的一部分。该模式是由“市场部门”所提供（**数据提供者**）。

当然，这个示例只介绍了企业的一部分。为了满足需求，销售部门、会计部门和IT部门也都有自己的方式对引用数据进行分类和再分类。MDM解决方案还需要捕获企业各领域的结构和语义，灵活的**当事人角色**数据模型结构可以很好地适应将来的这些需求。此外，外部**数据提供者**可以提供给金沙经销商**当事人角色类别类型**的分类模式。例如，OPEC对所有的石油**顾客**使用一个分类模式，或Dunn和Bradstreet会基于你想要捕获的行业类型而使用他们自己的外部分类模式。大多数主数据管理解决方案都需要这些分类具有若干不同的分层关系和/或聚合关系。因此，当事人角色类别和当事人角色类别类型有多个相关的隶属关系（分别是**当事人角色类别隶属关系**和**当事人角色类别类型**隶属关系）。这些实体支持存在于不同**当事人角色类别**和/或**当事人角色类别类型**之间的任意类型的分层、聚合或对等关联。当事人角色类别“战略性顾客”可能与较低层级的分类相关联比如说与“全球性战略顾客”和“国家性战略顾客”相关。这些分类是通过**当事人角色类别隶属关系**实例关联起来的。该实例术语当事人角色类别列示关系类型“市场报告分类”。可以在**当事人角色类别类型**“顾客细分类型”中对**当事人类别分类**的“核心顾客”、“主要顾客”、“战略性顾客”加以分类。这些不同类型的分类也可能有一个分层结构。例如，“顾客细分类型”和“顾客行业类型”在**当事人角色类别类型**“顾客分类”中。顾客主数据的核心需求是捕获顾客的各种联系方式、**联系机制**的不同的类型、目的、用途以及其他的联系机制分类。

在图9-8中，为了在处理顾客联系信息是取得很多的灵活性，数据团队使用了三级联系机制模式和包含灵活地址成分的联系机制模式。联系信息是主数据管理非常重要的一方面，因为它有助于身份认证管理；因而，通常需要很多的灵活性。第一，使用这个三级模式，你可以捕获**当事人**的任何**电子地址**、**电信号码**或**邮政地址**。人员和组织通过很多不同的方式取得联系。这些人员或组织可以是顾客、供应商、合作伙伴或竞争对手。他们可以同时扮演四个这样的角色（可能会更多）。但是不管他们扮演的角色是什么，他们的联系信息是一样的。例如“K Morris Corporation”既是金沙的顾客也是金沙的供应商。“K Morris Corporation”的总部是位于沙特阿拉伯王国的利雅得。无论它扮演的是供应商角色还是顾客角色，这个地址都是一样的。因此，图9-8所示的这种灵活的数据模型结构可以适应此场景以及很多其他的诸多场合。

**联系机制类别分类**、**联系机制类别**和**联系机制类别类型**提供了维护类型、用途、优先级、地点、**联系机制**或**当事人联系机制**的任何其他类型的分类。这个模型展示了可以通过一个或多个**当事人联系机制**来联系每个**当事人**，每个当事人联系机制都可能与不同的联系机制类别相关联（通过**联系机制类别分类**关联实体）。要么是**联系机制**关联至联系**机制类别分类**，要么是**当事人联系机制**关联至**联系机制类别分类**，因而任何一个都可以根据无数的目的、用途、优先级或其他分类来进行分类。例如，顾客联络员可能有一个联系机制（**当事人联系机制**），该联系机制的联系机制类别有“邮政地址”、“付款”、“首要”、“公用”。这些分类都分别与**联系机制类别类型**中的“联系机制类型”、“目的类型”、“优先级别”、和“用途类型”相对应。市场引用可能只想联系信息包含一个目的类型“销售”，而不想使用任何只用于“个人”用途的**联系机制**。

世界各地不同的邮政地址结构可能会引起大多数企业的问题。例如，在莫斯科的地址结构与曼谷的地址结构是不同的。不同国家的不同邮政组织数对于他们的司法关系区的划分结构也稍有不同。为了处理这方面问题，数据团队使用第七章节中的维护灵活地址成分的联系机制模式，这种模式在维护涵盖很多行政辖区的国际地址时可以同更大的灵活性。试想下董事会希望给世界各地的股东发送年度报告。到那时不能在信封上写出正确的地址结构。这种模式能够以非常灵活的方式帮助处理国际地址。

主数据管理非常重要的一方面是维护业务规则的能力。因此，数据团队决定使用第8章节中的三级业务规则模式，这种模式为维护业务规则提供了很大的灵活性。主数据管理一条常见的业务规则是如何得知某一源系统的顾客（（例如，居住在纽约州、纽约市100号大街的Kathy Moris））与另外一个源系统的顾客（例如，居住在纽约州、纽约市100号大街的Kathy Moris）是同一个当事人。例如，有一个匹配规则（业务规则的一个子类）实例，该规则用来维护顾客具有完全相同的名称的情况（一个泛化因素在**规则因素类型**中维护）或者是使用一个共用的同义词集（另外一个泛化因素）来匹配名称，并且这些联系机制是一样的（另外一个泛化因素），那么就可以认为该顾客是同一个当事人（在**规则结果类型**中维护结果）。

在主数据管理模型黄总还有其他规则可以应用于特定的角色类型，比如说顾客、供应商、当事人、工作人员（保留雇员和合约商）等等。例如，这个结构可以维护业务规则，比如说每个顾客每年都必须要有信用等级再评估，或者欧洲的每个顾客必须要有一个有效的欧盟税务标识。这些可以是**顾客规则**（**顾客规则**的一个子类）的示例。关于顾客数据的另外一个重要的规则集合是什么样的数据可以作为主数据，这样的主数据可以同哪些人分享。例如，由于特定的规则，有些顾客主数据不能为企业各组成部分所共享。

**注：正如我们早前在本章节所提醒的那样，数据源的捕获和映射不在这个初始的数据模型范围之内，尽管我们认识到这个很重要。虽然我们可以使用三级业务规则模式捕获这个需求的方方面面，但是我们可以为这个需求开发一个特定的数据模型，在主要主键和源系统主键之间创建一个交叉引用实体。对于这个需求有很多不同的方案，通常认为它是物理数据库设计的依据。数据团队感兴趣的是捕获主数据管理需求和确保MDM方案足够灵活以能够满足企业的需求。换句话说就是，图9-8不能直接解决数据‘来源和标签’的建模问题【批注15】，需要返回到原来的原系统，交叉应用源系统的数据，在这一节中应当认为这些数据是系统的记录。**

**为什么我们要使用这种方式来完成这个模型？**

当数据建模团队为金沙经销商的顾客主数据建模时，他们需要考虑一下几个因素。

第一，他们需要确保他们有一个足够灵活的解决方案，该方案要能够捕获所有当前和将来的数据，这些数据构成了顾客主数据，还要能够使用源自不同类型的原型系统的格式。他们通过使用非常灵活的数据模型模式（大多数情况下是使用三级模式）达到了这一要求。例如，你可以从图9-8中看到三级声明角色模式、三级状态类型模式（针对**当事人角色**）、三级联系机制模式、三级分类模式（针对**当事人角色**）、维护规则的三级递归模式（针对**当事人角色**）和三级业务规则模式（针对**角色类型**）。这些模式通过添加其他的模式提供了更多的灵活性，比如说状态类别模式，维护多重隶属关系和规则的状态类型模式和维护灵活地址成分的联系机制模式。

第二，为了提高数据质量，顾客主数据需要捕获各种数据集成规则。例如，顾客状态数据的模型使用了维护多重隶属关系和规则的状态类型模式和强健的状态数据集，在该模型中，可以将数据集成规则用于确保应用了合适的状态。在这个模型中，每个**状态类型关联**都会被**状态类型关联规则**所约束，比如在**状态类型类别**“顾客购买状态”的上下文中，状态“激活”和“待定”是不应当同时存在的。维护灵活的关系和规则的另外一个方式是使用维护规则的三级递归模式为当事人关系结构建模（分层、聚合和对等关联）。这样可以发现当事人与当事人之间的任何关联方式，比如说在社会网络分析中使用这种模式。它允许他们维护关系的制约，比如说某个当事人不应当与某一特定的顾客相关联。最后，三级业务规则模式的应用允许数据模型维护诸多其他的业务规则。

对于企业主数据管理解决方案需要什么，图9-8可以用来作为业务需求声明的一部分。例如，分类数据模型结构表明使用灵活方式将顾客数据分类和再分类成不同的类别和类别类型是很重要的。此外，这个模型展示了这些类别和类别类型都可以具有多个层级和分层（或聚合），这些层级可能是某个特定模式的一部分和/或源自于某个提供者。此外，金沙经销商可以直接决定使用这个模式（在MDM预包装解决方案内使用或者用于顾客MDM的实现），将其作为基础，用来实现主数据管理数据库解决方案。

### 将这些模式用于解决方案有哪些优点？

将这些模式用于主数据管理数据模型有以下几个优点：

* **这些模式可以适应很多不同类型的源系统数据结构。**在这个示例中介绍的泛化模式可以适应各种源系统数据库结构。例如，源系统数据结构可以通过很多不同的方式来维护联系信息，比如说使用一些特定的字段来维护电话号码，可以使用一张电话号码表来维护电话号码，还可以使用一张联系信息表等等。在主数据管理数据模型中，将非常泛化的模式应用于联系机制，你可以容纳源自各个源系统的各种结构。可以认为这个模型是“最小公分母”模型，该模型几乎解决了所有的源系统数据的需求。
* **使用更加泛化的模式，为使用当前和将来的需求提供了更大的灵活性。**例如，如果需要为顾客主数据捕获新的分类类型、角色、状态或关系的话，这些都可以被容纳，并且可以保留底层的模型，不必做任何变更。
* **各种规则模式提供了动态维护很多规则业务规则的方法，该方法有助于控制数据的集成。**例如，对于相似数据应当以何种方式出现，所捕获的业务规则维护了其标准或因素（这就是**匹配规则**）。这个数据模型结构还可以维护很多其他的规则，比如说有关顾客或其他当事人角色的需求的规则。并且这个模型维护支配状态分类或关系的规则。这个模式有助于数据质量的提高和数据的管理。
* **考虑到使用更加一致的管理主数据方案，在整个模型中，可以使用相同层级的模式。**一个模型可以重复应用同类型的模式，相同数据模型的分类、递归关系和规则可以应用于很多领域。例如，数据团队认为他们可以一致的将三级分类模式用于当事人角色（包括顾客）、联系机制和业务规则的分类。
* **标准化的模式可能会有助于削减成本。**如果这些模式在整个企业中变得标准化，那么该模型有助于开发共同的代码、惯例和服务，因此，可以减少系统开发和维护的时间和成本。数据团队可以在应用系统中、企业数据模型中和数据仓库中使用同类型的模式。

### 将这些模式应用于解决方案会有哪些缺点？

将这些模式应用于主数据管理数据模型存在以下几点缺陷：

* **当使用更加泛化的模式时，模型会变得更加难以理解。**如果数据专业人士对模式不是很熟悉，那么起初阅读这种数据模型会很困难，它的复杂程度甚至使经验丰富的数据专业人士看起来都像个外行或未经训练的新手。数据团队制作这个模型的目的是描述MDM解决方案购买方的数据需求。在此模型中泛化的模式可能不是描述金沙经销商需求的最有效方式。此外，数据团队将必须‘走向’潜在的供应商，通过模型来描述他们需要实现的需求。源自模型的这些需求并不是不言自明的，除非供应商对模式非常的熟悉。
* **使用较为泛化的模式的话，将各种源系统映射到这个模型可能会更加复杂。**使用这些泛化的模式来映射、集成和同步很多源自不同系统的数据的挑战性可能会更大，因为他们更具挑战性，所以企业对于这类模式的在其他应用中的使用制定了标准，这样的话，实际上映射起来会更加容易。
* **如果使用更加泛化的模式的话，在数据模型的关系和属性方面，不能体现很多特定的规则。**泛化数据模型提供了灵活性，但是有时牺牲了具体的需求。使用泛化模式创建这些模型，你可能不能捕获特定的属性和/或关系。例如，**产品族**分类（二级分类模式）可以有一些特定的属性或关系，这些属性和关系只能应用于**产品族**。通过使用三级分类模式，你可能不可以捕获这些特定属性或关系。或者是有一个规则规定顾客只能归类于顾客估摸。使用泛化模式的话，你不能直接在数据模型中实施这条规则，因为你可以让顾客归类于任何规模。

### 简介

主数据管理解决方案需要能够灵活地管理源自多处的数据，为了提高数据质量还需要为主数据创建强健的规则。

主数据管理数据模型主要使用三级模式来灵活地维护顾客（以及其他相关的当事人）数据、当事人分类、联系机制信息、当事人之间的关系和当事人状态。

主数据管理数据模型也包含了维护**业务规则**的方法，业务规则以多种途径帮助管理主数据。比如说谁可以查看某些类型的主数据，当事人是如何匹配的以及顾客和其他当事人角色的规则。使用维护规则的三级递归模式来维护**状态类型**和**当事人关系**的其他规则。因此，模式的灵活性也受管理关系（可能存在于引用数据中）的规则约束。

这是个实用了很多灵活数据模式的复杂的数据模型。应为这些模型使用了很多概括的概念，所以有一些受众会有一段时间内很难理解这类模型。一些数据专业人士可能会认为在主数据管理数据模型中的那些建模规则是没有必要的。尽管在有些企业中情况确实如此，但是其他的企业（规则驱动型企业）可能需要使用这些功能强大的结构来帮助确认他们的主数据的集成，同时这种结构也为主数据管理提供了非常灵活的基础，可以支持来自各源系统的各种各样的需求。

## 关于模式应用的其他想法

在这一节，我们通过讨论一些常见的物理数据库设计概念和一些使用这些模式的其他方式来解决一些问题，这些问题可能还会影响你对模式的选择。

### 物理数据库设计

尽管在本章节的有些地方使用了一些物理数据库概念，比如说在星型模式这部分，但是本章节的大多数地方都没有将这些模型转化成物理数据库设计。将这些模型转化成物理数据库设计需要考虑到很多因素，比如说：

数据的容量：各个实体有多少个实例？

数据访问频率：期待哪种创建、读取、更新和删除活动？

物理数据库设计标准：类似于数据建模标准，物理数据库设计标准从属于不同的风格，比如说不同的命名约束，例如，有些物理设计标准限制了表或列的名称长度，这通常都基于所使用的数据库平台。

子类的实现：子类可以按照以下四种基本方式转化成表：

1. 父类和子类变成两张表：一张表是‘数据’表，一张表是‘类型’表。但是，在我们的大多数模式中，已经存在‘类型’实体，这也意味着子类将会继承超类（或‘数据’表）的属性和关系，这样就只需创建一个表。
2. 将父类变成一张表，每个子类作为一张表。
3. 仅将子类变成表，超类的属性向下传递给每个子类。
4. 混合或结合使用这些方式，例如，为超类和某个子类创建一个组合表，然后为其他的子类都创建一个表。

我们不打算深入讨论这些或其他的物理数据库设计问题，因为这些内容超出了本书的预定范围；数据库设计的原则不依赖于通用模式的用途，无论是否使用这些模式，数据库设计原则都是可应用的。此外，已有大量的书籍涉及这个主题。

### 模式的其他应用

我们已经介绍了很多通用的应用，在这些应用中都使用了这些模式。然而，这些模式在很多其他的应用中也有用。例如，可以使用这些模式开发传递数据的标准方法和/或创建共同的或‘通用的’XML模式。这些模式还有助于建立关于数据管理方案的公用数据语义。在数据管理方案中，数据团队可以使用一级模式为业务代表开发公用的数据语义集，还可以使用三级和四级模式为技术代表开发另外一个数据语义集。无论是否需要理解数据是否需要为数据建模，建模人士都可以使用这些模式来快速启动模型、比较方案、应用一致的结构以及确保模型的质量。

### 使用这些泛化模式的其他注意事项

你可以从本章节和本书中看出泛化模式在灵活性、一致性、可复用性以及功能效率方面的优点。在本书中我及介绍了具体的模式也介绍了泛化的模式。当在泛化模式与具体模式之间做出选择时，我们在整本书中都表明了模式的优点和缺点。关于泛化模式的使用，我们认为有些因素还是值得一提的，比如说：

* 泛化结构将变化流程从经典的‘数据架构师—DBA—开发人员—测试人员’流程迁移至数据变化流程，通常对于数据变化，组织没有规范的流程，就如他们的应用变化没有规范流程一样。这个问题可以通过组织致力于使用灵活的数据建模结构来解决。事实上，我们认为在授权用户控制数据方面，使用泛化模式是很重要的一部。就用户的需求来说，经典的数据架构—DBA—开发人员—测试人员流程通常太过繁琐缓慢。泛化模式促使思考模式的转移【注释16】，从这种典型的开发流程转移至数据变化流程。
* 在使用泛化的数据模式时，数据管理工作是很重要的，因为新实例会对模型产生很大的影响。例如，新实例可能会导致新角色、分类、状态、联系机制类型或业务规则的产生。数据管理工作失职指派负责数据的当事人，在数据模型中泛化的模型需要管理更多的实例。在数据管理工作中使用泛化模式的优点在于他们通常可以在不变更模型的情况下添加或变更数据类型，因而能够有助于数据管理。如果将数据管理纳入计划和预算，如果企业投资数据管理工作，那么它将会认识到更多的灵活数据结构的益处。

当业务规则发生变化时，你就会认识到使用更加泛化的方案的主要优点，特别是项目团队不能满足将来需求的时候。【批注17】这种脱节可能会导致挫败感和‘超安全标准设计【批注18】’的感觉。通过指导和交流，你可以解决这种脱节，并且可以向企业展示泛化数据结构的优点。

* 有时DBA或开发人员可能会提出这样的问题，泛化数据模型的性能是否与具体数据模型的性能一样。就我们的经验来说，影响性能好坏的主要原因除了泛化级别外还有其他的因素。

## 模式应用总结

**表9-1** 包含了相关模式应用的总结。

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | 介绍 | 优点 | 缺点 |
| 声明作用域和理解数据需求的数据模型/原型数据模型 | 这是一个使用了一级和二级模式的特定数据模型。在这里有两个目的。第一，创建非常易于理解的数据模型，该模型可以用来收集验证数据需求。第二，创建一个模型，根据这个模型容易快速建立原型，达到阐释需求的目的，这样便于解决变化。 | 易于理解；  在数据模型开发工作中易于使用；  易于原型的实现；开发数据模型的良好开端；  同业务带表做有效沟通；  稍加变动便可以变成较高层级的模式。 | 一般情况下，使用一级模式不能为实现提供良好的基础。  不能复用共同的惯例。  缺乏灵活性。 |
| 应用数据模型 | 满足一个成熟的产品应用需求的数据模型。这是个需要满足当前和将来应用的数据模型。一般而言，当静态的数据需求非常明确的时候，使用二级模式，部分应用需要更大的灵活性时，可以使用三级模式。 | 特定模式与泛化模式的平衡；  满足应用的特定需求；  通用的集成解决方案；  通过复用同类型的数据模型结构，这些模式提供了一致性。 | 有些部分不灵活，有些部分很复杂；  多重风格的模式，因而，形成多重风格的模型  在使用更泛化的模式，缺乏理解性；  使用更泛化的模式时，缺少了一些业务规则的说明。 |

（续）

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | 介绍 | 优点 | 缺点 |
| 企业数据模型 | 在这一节中，我们介绍了开发企业数据模型的方案，这个方案通过使用不同层级的模式，以多种方式为同类型的数据建模。因此，这些模式可以提供标准的数据模型结构，但是有时还可以有备选方案，因为不同的应用有时具有的不同的需求。 | 为常见的需求提供了质量数据模型；  这些模式保证了真个企业数据模型的一致性；  为企业数据模型提供了备选方案；  为企业数据模型建模工作节省了时间。 | 对于同类型的数据具有多个备选模型，起初会引些混乱。  在同一数据模型中使用不同层级的模式会导致不一致性。 |
| 关系型数据仓库数据模型 | 对于企业数据仓库模型的创建来说，这个模型是个非常灵活的方案，企业数据仓库模型被认为是更加贴近于关系型数据模型。企业数据仓库设计时以关系型数据模型为基础的，通常都是基于企业数据模型。有时这个模型与应用“中心辐射”架构（在企业信息工厂中使用的架构）的理念相关，该架构与“总线架构”相反（在Kimball的“数据仓库工具集”方法中使用了这个架构）。 | 三级模式为维护支持数据的决策提供了灵活的解决方案。  一致性—在整个模型中都使用了同一层级的模式。  捕获历史记录—三级模式可以追踪随着时间的推移关系发生了什么样的变化。 | 使用泛化模式的结果是更加复杂的模型。  这些模式不能容纳数据仓库所需的所有历史记录。 |

（续）

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | 介绍 | 优点 | 缺点 |
| 星型模式—基于数据仓库的数据模型/星型模式设计 | 这是一个将数据模型作为星型模式设计前导的方案，在此数据模型中使用这些模式，你可以更好地理解各维度本身固有的复杂性。因此，因而，这个方案可以让你看到星型模式设计所需要解决的复杂性。 | 是几乎没有错误的更加坚实稳定的星型模式设计。  一致的设计为开发维度提供了更加一致的方法。  使用通用的模式开发数据模型，可以让人们更好地理解数据。 | 该数据模型是用来作为星型模式的基础，有是不能展示真实的数据关系。  有些人认为使用这些模式开发关系型数据模型，对星型模式开发来说是没有必要的多余步骤。 |
| 主数据管理 | 在这一节，我们创建了主数据管理数据模型，该模型可以灵活地处理多个主数据源（和目的地），还可以维护主数据的有关规则的参数。主数据主要是通过的三级模式来维护的，主要是维护顾客（以及其他当事人）、联系信息、当事人之间的关系、当事人的分类和当事人的状态。 | 可以容纳多种源系统格式。  更加泛化的模式的使用提供了更大的灵活性，可以适应当前和将来的需求。  各种规则模式提供了动态维护多种业务规则的方法，这些业务规则有助于控制数据的集成。  在模型中使用相同层级的模式，这样能够使用更加一致的方法来维护主数据。  如果一致的使用这些模式并在整个企业中标准化这些模式，那么通过提供公用的服务和惯例，使得模型标准化，并可能减少成本。 | 当使用较繁华的模式时，模型会很难理解。  当使用较泛化的模式时，在数据模型的关系和属性方面不能体现很多特定的规则。 |

**参考文献**

1 参看Thomas H. Davenport 和 Jeanne G. Harris所著的*Competing on Analytics: The New Science of Winning*（哈佛商学院出版社，2007）。

2 更多关于混合型上下文角色模式的信息请参考本书的第三章节。

3 更多关于如何社会化模式以及如何让这些模式获得认可的信息请参看本书的第10章节。

4 请参看William H.Inmon所著的*Building the Data Warehouse*的第四版(Wiley, 2005)。

5参看 W. H. Inmon、Claudia Imhoff和 Ryan Sousa所著的*The Corporate Information Factory，*第二版 (Wiley, 2000)。

6 参看Ralph Kimball 和 Margy Ross所著的The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling，第二版(Wiley, 2002)。

7 源自David Friedland的*The Data Administrator Newsletter："Considerations for Managing Risk in a Post-SOX Environment"*，2007年4月1日出版。访问地址：http:/ /www.tdan.com/view-articles/4937.

8 在CIO杂志中，Tony Fisher在*"Demystifying Master Data Management"*一文中所使用的术语(CXO Media Inc., 2008)。

9 参看L. Silverston所著的数据模型资源手册修订版卷1的第2章节：企业通用数据模型(Wiley, 2001)。

10 关于物理数据库设计考虑因素一本很有价值的书是Sam S. Lightstone、Toby J. Teorey和 Tom Nadeau所著的第四版*physical database design considerations is Physical Database Design: The Database Professional's Guide to Exploiting Indexes, Views, Storage, and More*(Morgan Kaufmann, 2007)。

11 关于泛化的其他考虑因素的一些材料由行业思想领袖Karen Lopez提供。

【批注1】原文：that type of pattern can generally accommodate the wide variety of data requirements of the source systems feeding the data warehouse.译文：这种模式一般能够满足数据仓库源系统的各种需求

这里的原系统（the source systems）与数据仓库之间（the data warehouse）

是什么样的关系，feed这个词的一般意思是“供给”、“靠……为生”。

【批注2】 Scoping data model ?Scope :“作用域”、“范围”

【批注3】 Bill to customer收到发票或账单的顾客。 Ship to customer:是指收取商品的顾客。

【批注4】原文：The patterns described in this section can also be used as a starting point to establish a baseline for communication across functional boundaries as they capture common core terminology.

这一节所介绍的模式在捕获常见的核心术语时，还可以作为建立夸功能边界交流基线的一个起点。

【批注5】原文：a prototype set of reports, screens, and analytic routines

报告、屏幕和分析例程

【批注6】individual attributes暂译“个别属性”

【批注7】 exclusive arc暂译“互斥弧”

【批注8】 ordered for use by 暂译“订购用户角色”

【批注9】view of their data landscape 暂译“数据视图”

【批注10】Decision support needs 暂译“决策支持需求”

【批注11】Constellation 暂译“星座”，指的是一种模式。

【批注12】separating based on stability“基于稳定性的分离 ”？

【批注13】 facet data “facet 数据”？

【批注14】 "High level of activity" 、 "Low level of activity."“高度活跃”和“低度活跃” 指的是状态。

【批注15】'sourcing and tagging' 来源和标签？

【批注 16】paradigm shift “思考模式的转移”

【批注 17】typically project teams are not compensated for meeting future needs？特别是在项目组不能满足将来需求的时候？

【批注18】 overdesign “超安全标准设计？”