# 类型和类别：数据的分类

自从20世纪80年代以来，我们见识了全球数据内容的惊人发展，出现这种情形的部分原因是因为万维网成了人们生活的一部分。我们该如何管理这些数据内容，又该如何开始弄明白这种种信息？生物学家帮忙解决这些问题。从Carl Linnaeus时期开始，一个伟大的瑞典博物学家，生物学家就一直在使用分类方法来处理庞大的数据信息，现今这种方法也称作分类法[taxonomies]。

为了将复杂的数据集合组织成各类别，人们创建了类，而这样做似乎是人类的本能使然。为组织数据集合，分类方法是提供给人们的通用词汇。例如，大多数企业为了管理其业务，需要数据分析和报告。高级管理人员需要看按照业务线划分的各类收益报告，以及技术呼叫工作也是由异常情况报告来驱动的，根据错误的类型对这些报告进行分类的。制造公司需要根据产品的类型来报告可用的库存，等等。在以上这些企业工作的人需要分析报告中的数据，总结报告信息并将其归类，或是深究一类底层数据。在企业中，类别是各部门和人员之间的通用词汇。此通用语言要求可以将各种类型的数据方案在一起并且可以划分成不同的类别。

## 这种模式的重要意义是什么？

数据模型通常捕获了各种类别[categories]、类型[types]、团体[groups]、家庭[families]或对实体进行分类的其他方式。例如，可以有从产品组[product groups]、产品类型[product types]、产品族[product families]、客户类型[customer types]、设施类型[facility types]、行业类别[industry classifications]以及很多其他的实体分类。在没有确切地理解分类建模的通用方式的情况下，通常这些分类的模型都是被独立的或实时动态地创建的。对于数据驱动型企业而言，具有一种一致健壮的方法来对数据进行分类是很重要的。

分类需要具有哪些基本信息？我们将在本章节讨论以下几种分类信息类型：

* 类别是如何关联到正在分类的数据的？例如，产品类型是如何关联到产品上的。
* 各类别之间是如何关联的，是否有不同类型的类别，一个类别能不能成为另一个类别的成员。
* 有哪些实体分类的方法？例如，一个手机模型能否同时归类到“奢侈品”和“电子设备”中。

## 本章有哪些内容？

当谈论到数据的分类时，有时我们可以交替使用术语类型[type]、类别[categorization]和分类法[taxonomy]。他们指的是同一个概念吗？为了解决这个问题。我们首先定义了什么是分类[classifications]、类型[types]、类别[categorizations]和分类法[taxonomies]，并讨论他们之间的不同之处。然后本章介绍了数据模型模式，用于满足企业对数据进行分类的需求。 这些模式介绍了不同的方法，我们就是使用这些方法来组织分类数据的。

像本书的大多数章节一样，在本章节中，模式的建模风格都是从最特定的建模风格（一级分类模式）逐渐过渡到非常灵活的建模风格（三级分类模式）。不同的概括层次可应用于不同的企业或不同的建模风格。例如，为了理解数据需求，想要做一个简单的声明的企业可以选择一级模式。若是想要更适合用于实现的灵活方案，他们可以选择二级或三级模式。

在本章节，我们使用实体产品来介绍如何使用这种模式的不同版本。当事人[PARTY]、固定资产[ASSET]、货运[SHIPMENT]、政策[POLICY]、支付[PAYMENT]、工作计划[WORK EFFORT]或其他需要分类的实体均能够以类似的方式使用这些模式。

本章节包括：

* 分类的定义
* 支持分类的各种模式
* 每种模式的关联性
* 深入了解每种模式
* 该在何时使用和何时不该使用这些模式
* 每种模式的简介和优缺点

类型[Types]、类别[categories]和分类法[taxonomies]是关联十分密切的概念，但是它们的含义稍有不同。本章所介绍的模式均支持类型[type]、类别[category]和/或分类法[taxonomy]。

类型[Types]的定义是“一些事或人都有某一特定特征或一些特征，这些特征使他们归为一组，这些事或人的定义或所指大同小异。”在我们的模式中一个类型实体以一种非常简单的方式对我们正试图分类的‘基本’实体进行了分类。例如，对订单进行分类，那么就会有个相关的订单类型，这个实体包含了实例：“销售订单[Sales Order]”和“购买订单[Purchase Order]”。可以认为两个实例就是订单的类型。

可以将类别[categories]定义为“任何一般或综合的划分”类别[categories]和类型[Types]的最大的区别是类别[categories]比类型[Types]更加全面，它包含了对实体进行分类的各种方法，也就是说，一个类别包含很多不同的“各类类型”。例如，一个人可以根据很多不同的方式进行分类（例如，根据收入水平进行分类、根据性别进行分类，根据种族进行分类等等），类别包括了以上所有不同的类型。

分类[Taxonomy]可以定义成所有元素的分类集合。分类需要满足分类学的两个方面，第一个方面是经典分类的科学严谨性，这种分类可以在自然历史中看到。“分类是指给标识物种或是指给物种命名，并使他们形成一个系统的分类。”其次，分类学已成为网站内容管理和设计的非常重要的一方面。分类法是为了支持“直观的家族群体”

分类法[Taxonomy]是指类别[categories]（和类型[types]）的分层或聚合。从本书的角度来看，通过分类[classification]我们的模式可以使用不同的方式来管理内容。例如，对“当事人”进行分类，一个分类法[taxonomy]可以包含各种当事人分类[party classification]（例如，“当事人类型[Party type]”、“产业分类[Industry classification]”、“尺码分类[Size classification]”等等）和这些类型实例之间的关系，比如说，“当事人类型”：“人员”和“组织”是如何与“尺码分类[Size classification]”实例：“大号[Large]”、“中号[Medium]”和“小号[Small]”相关联的。

类型、类别和分类也还可以有分类，这些分类在其他分类中（或与其他分类相关）。例如，对产品进行分类，产品类型可以有实例：“商品”和“服务”。那么你可以进一步将这两个实例拆分成其他分类。例如，可将“服务”拆分成“时间和材料服务”（指那些按照小时计费的服务）和“交付服务”（指那些根据产品的交付来计费的服务）。

**注：尽管我们已给出了类型[types]、类别[categories]和分类[taxonomies]的定义，但是本章节将重点主要放在一级和二级模式的“类型[types]”模型和三级模式的“类别[categories]”模型。因为我们认为“类别[categories]”更加的全面，不只是包含一个简单的“类型[types]”。当我们指分类[classification]时，这个分类包含了“类型[types]”和“类别[categories]”。由于分类[taxonomies]是由“类型[types]”和“类别[categories]”构成的，所以这些模式都支持不同复杂程度的分类[taxonomies]。**

## 一级分类模式

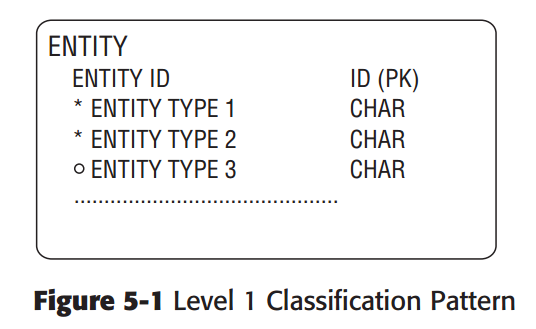
对于捕获实体的类别[categories]和类型[types]，使用属性是非常简单的方法—例如，一个属性订单类型就可以维护订单究竟是个“销售订单[sales order]”、“购买订单[purchase order]”、“工作订单[work order]”还是个“生产订单[manufacturing order]”。

### 为何我们需要这种模式？

在特殊情况下，企业通常需要非常简单容易的分类建模方法。一级分类模式为分类建模提供了这种特定的方法。所有的分类信息被当成了实体的属性。换句话说，该实体将所有的分类作为属性（例如，订单类型），这些属性的实例维护了实体的类型（例如，“销售订单[sales order]”、“购买订单[purchase order]”、“工作订单[work order]”或“生产订单[manufacturing order]”）.

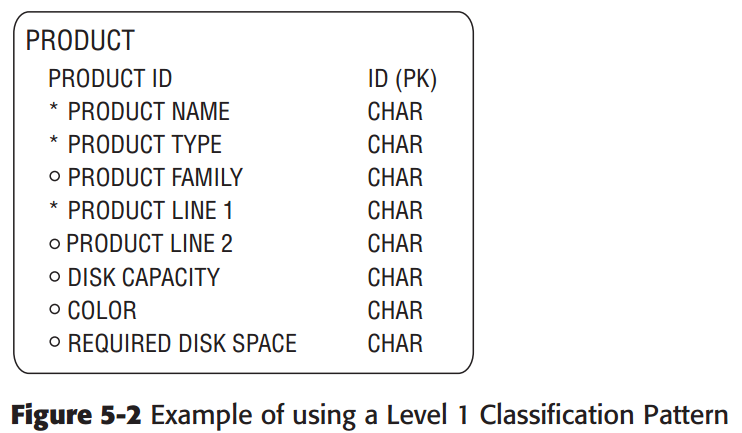
### 这种模式是如何运作的？

图5-1所示的模式为具有不同类型的实体提供了一个非常特定的模型。**实体**包含三个分类：实体类型1、实体类型2和实体类型3。此模式可包含任意数量的类型属性，出于阐释的目的，我们只是列举了其中的三个。这些分类可以或必须包含不同“类型”**实体**。例如，**实体**必须有实体类型1和实体类型2的值，但是也可以根据实体类型3进行分类。



**图5-1一级分类模式**

试想下，数据专业人士正在研究一个大型计算机软硬件零售商，欧洲电子零售商是如何对数据进行分类，并将其作为数据仓库项目的一部分。假设数据专业人士被要求提供一个范围声明，声明欧洲电子零售商现有产品的所有可能的分类。基于与管理层的会谈和以及分类模式的使用，数据专业人士制作了图5-2。



**图5-2** 一级分类模式的应用实例

图5-2展示了如何使用这种模式。欧洲电子零售商将已将产品定义为“正在、已经或将要出售给客户的硬件、软件或配件”。产品类型[Product type]、产品族[product family]、产品系列1[product line 1]和产品系列2[product line 2]包括了不同“类型”的分类，这些类型以不同的方式对产品实例进行分门别类。之所以会有产品系列1[product line 1]和产品系列2 [product line 2]属性，是为了一个产品可以有多个系列（例如，可分类为“商用[commercial use]”和“家用[home use]”）。

**注：分类的特例是指示词，例如，“男[M(ale)]”和“女[F(emale)]”，用来指示性别。或“是[Y(es)]”或“否[N(o)]”用来指示一个人是否吸烟。通过提供两个指示词可选值来维护某个实体实例的特定信息，例如，某个特定人员是男性（与女性相对）或该人员是个吸烟者（通过选择吸烟者指示词属性的值“是[Y(es)]”）。大多数分类都支持很多的可能的值，这些值可以将数据分类，当可以使用指示词将数据分类成两种特定的可能性时。然而，当使用指示词属性时，建模人员需要十分地小心，因为可能的属性值的数目可以扩展到两个以上。例如，性别指示词实例可以拓展为“男（M(ale)）”、“女（F(emale)）”和“未提供（N(ot given)）”。**

**另一种观点认为指示词在语义上与分类是不同的，因为指示词只能用于捕获实体实例的某一特定信息（或表中一行）例如，某个人员是男性（相对于女性）或该人员吸烟或不吸烟。这种观点有一定的道理，但是即使你认同这种观点，仍然使用指示词对实体进行分类，但是组织中的其他人往往不一定会认同这种观点。**

表5-1提供了产品的示例，该产品是使用了多种不同的分类方法。在表的第一行，是产品的实例“存储盘2000[Save Disk 2000]”，该产品的分类如下：产品类型[product type]（“硬件[Hardware]”）、产品族[product family]（“硬盘驱动器[Disk Drives]”）、产品系列1[product line 1]（“家用[Home Use]”）和产品系列2[product line 2]（“商用[Commercial Use]”）。正如我们先前所述，产品系列1[product line 1]和产品系列2[product line 2]是为了配合可以将一个产品分类成多个产品系列的需求；例如，“存储盘2000[Save Disk 2000]”的产品系列1[product line 1]是“家用[Home Use]”的，而其产品系列2[product line 2]则是“商用[Commercial Use]”的。在最后一行中，“标准存储卡[Standard Memory Card]”也有很多的分类：“硬件[Hardware]”（产品类型[product type]）、“计算机[Computer]”（产品族[product family]）、“家用[Home Use]”（产品系列1[product line 1]）和“商用[Commercial Use.]”（产品系列2[product line 2]）。

该**产品**还有其他三个属性。磁盘容量[Disk capacity]维护了硬盘或存储设备的可用存储空间大小，并且这个属性只适用于硬件。颜色[Color]维护的是配件的颜色，且只维护配件，比如说包装盒。必备磁盘空间[Required disk space]维护的是覆盖区（或所需空间，以MB为单位），且只有软件产品维护这一属性。

**注：也可以认为颜色[Color]、磁盘容量[Disk capacity]和必备磁盘空间[Required disk space]这些属性是将各种产品进行分类的方法。然而，本章的目的是简化示例，我们将会将讨论的重点放在产品类型[product type]、产品族[product family]和产品系列[product line]的分类上。可以将磁盘容量[Disk capacity]、颜色[Color]和必备磁盘空间[Required disk space]作为特征来建模。想要更多的了解这个概念，请参看数据模型资源手册卷1，修订版中的第3章节（Wiley,2001）。**

**表5-1** 以及分类模式范例

**（表）**

从表5-1（和图5-2）中你可以看到不是所有的类别都是强制性的。例如，“超销售软件包[Hyper Sales Software Package]”就不是根据任何产品族来分类的。数据专业人士还发现同一产品可以被分类成多个产品系列；因此，重复的属性产品系列1 和产品系列2将产品分成了两个不同的系列。这种重复的组强调了一些数据问题，这些数据是需要专业人士进一步研究的，因为一个产品可能会需要更多的产品系列。（例如，如果将一个产品分类为“家用[Home Use]”、“商用[Business Use]”和“家商两用[Home Business]”，有该怎么做呢？）

如果你对数据建模很熟悉，你会发现这种模式明显不是常规化的模式。在讨论产品系列1和产品系列2时，这种模式不符合第一范式（没有重复组），因为一个产品可以有很多的（重复的）产品系列分类。这种模式也不符合第三范式（传递依赖关系），因为每个“类型”的属性的值实际上并不依赖于产品的键，但是却依赖其自身的键。例如，第一行的“磁盘驱动器[Disk Drives]”这个值不依赖于产品的键“102”，这是因为很多产品可以跟“磁盘驱动器[Disk Drives]”一样的值。所以，根据常规的规则，应当在其自身的实体中维护“磁盘驱动器[Disk Drives]”（例如，**产品族**实体），这个实体可以具有一个值为“111”的主键和一个名称属性，例如，“磁盘驱动器[Disk Drives]”。在这个案例中，“磁盘驱动器”的值依赖于“111”或者是可以由“111”来决定。那么，如果这个值被划分到其自身的实体当中，在**产品**实体中会有一个值为“111”的外键，用来表明某事物属于“磁盘驱动器[Disk Drives]”这个分类，这样可以消除冗余信息，即在产品的若干个实例中都维护了的“磁盘驱动器[Disk Drives]”的相同数据。

因此，如果在关系型数据库中实现了这种模式，可能会产生巨大的数据冗余和数据不一致问题。例如，很多产品的实例都可能会重复值“磁盘驱动器”，且可能会导致数据不一致。当这个查询值变更成“硬盘”时，这个值的重复和冗余的实例会导致数据完整性出现问题。

或许你正在思考这样一个问题，“如果这个模型是错误的，为什么你还会将这个错误的模型作为一种可仿效的模式展现出来呢？”之所以会展示这种模式是因为为了便于与非技术受众沟通交流帮助他们验证数据需求，可以将此模式作为一种方法，以非常简单的方式标识出可能存在的数据类型。在业内很多建模人员指这种模式是个概念性的或业务的数据模型，其用途是用一种非常简单的方法向非技术受众阐释数据需求。在1975年，ASNI（美国国家标准协会）提出了以下几种分类：“概念性模式[Conceptual Schema]”、“逻辑模式[Logical Schema]”和“物理模式[Physical Schema]”以满足不同角度的数据需求，这与我们正在阐释的想法是一样的。

*因此，我们想强调下，我们几乎是一直不推荐将这这种模式（至于此方面，大多数指的是一级模式）用作任何长期或意义重大的实现的基础，因为在实现这种模式的过程中存在很大的漏洞，比如说冗余数据和不一致数据。*然而，并不是必须要向业务代表展示“常规化”的数据模型（他们并不关心这个模型是否是规范化的，他们只关心你是否能够捕获他们的需求），而是我们发现这种类型的模式可以作为阐释数据需求的有效方式，有助于同业务代表的沟通交流，以至于能够更好地理解业务代表的需求。

我们已经强调过不要将这种模式用于实现；然而，却有几个少有案例是可以的，当分类和分类的实例是完全不变的或静态的时以及当冗余数据不会产生太大的问题时，可以将这种模式用于实现。你可以说假如分类只需要一个简单属性，假如你知道这个实体永远只会有这一个分类，或是假如这个物理模型是受益于非规范化的设计，那么这种模式是个可能的实现。例如，对于人员使用性别类型属性，或许这是可能的，因为在这种情况下，其属性值永远都是“男[M(ale)]”、“女[F(emale)]”和“未提供[N(ot Given)]”。

值得一提的是很多遗留系统都是将分类作为属性（或列）来实现的。出现这种情况的一个原因是因为在最初遗留系统并不是相关联的。虽然可能这些系统已被转移到了关系型平台上，但是并没有转化遗留系统使用常规化的规则。由于在现实情形中很多系统使用这种模式实现了分类，所以我们认为理解这种模式以及其优缺点是很重要的。

### 应当在什么时候使用这种模式？

我们在以下情形中使用这种数据模型模式：

* **需要非常简单的模型来收集和验证数据需求。**这种类型的模型能够以一种非常易于理解的方式来阐释实体分类的范围。这种简单的方式更容易与商业人士达成共识。
* **作为原型的简单实现。**将分类作为属性，更加容易建立快速原型，但是要注意原型不会成为最终产品的。
* **作为一种方法，促进不同分类架构的分析。**对数据进行分类时它有助于展示业务所使用的术语。它引发了一些数据问题，例如重复产品。这种模式通常以一种非常易于理解的方式帮助启动数据建模流程。

### 这种模式的缺点是什么？

此模式的缺点如下：

* **维护冗余数据且不能将其用作意义重大或长期实现的基础。**这种模式维护了冗余数据，分类的每个值可能会被实体的不同实例不一致地重复很多次。这种模式不能作为实现的基础，除了个别情况。例如，在一个行业类型中维护“制造业”属性，该类型对当事人进行了分类，对于同一个行业类型，当事人的实例可能会具有不一致的值（例如“Manufacturing”和“Mfg”）。
* **非常的不灵活。**对于任何刚性架构，变化可能会导致基于这种模式的数据模型需要重新构建。因此如果你根据这种模式实现了数据库架构，随着时间的推移，当出现变化时，可能需要更改数据库架构，这项工作的代价是十分巨大的。一个**当事人**可能会有很多的分类的，比如说尺码[size]、行业[industry]、收入[income]、民族[race]、性别[gender]、市场领域[market segment]，并且随着时间的推移会出现更多的分类。
* **当一个实体可以多次归类到相同的“类型”中时，这个方法可能不太实用。**例如，当事人可以在同一时间归类到若干行业中。（在不同市场中运营的同一家公司可以分类为制造、医疗保险、伤残保险、电信、零售等行业。）这种模式要求有一个属性表示但是人所在的行业（行业类型1、行业类型2等等）。这种模式可能不适用于这类建模需求。比如说本章节中的示例，为支持业务的需求，需要有两个产品系列属性。
* **不维护分类的数据。**通常类别具有很多不同的属性，比如说描述、起始日期和终止日期等。为了适应这样的情况，需要将这些属性添加到实体中，即便如此，为每个类型添加这些属性也会变得十分棘手。这种模式将属性保存在整个实体中，即使有些属性是可以应用到一个特定分类当中的。
* **不维护分类的业务规则。**这种模式只能处理最基本的规则，比如说强制性分类与可选性分类。

### 简介

这一节中你可以看到一级分类模式为分类建模提供了非常特定的方法。这种模式由于其简洁性而显得十分重要。这种模式使用属性捕获实体的实例，这些实例被分类成“类型”。作为作用域声明的一部分或某一特定实体的不同分类的可视化展现，这种模式是很有用的。这种模式可以帮助你用一种非常简单的方式启动数据建模工作，展示了公司在类型分类中所使用的专业术语。它甚至可以用作原型的基础。

几乎在所有的情况下，这种模式不能用于实行数据库的基础，因为关于数据冗余和数据完整性问题，这种模式通常会存在非常严重的漏洞。此外，如果需要添加新类别或类别的属性发生变化，那么需要添加新属性，所以这种模式不是特别的灵活不足以满足将来的需求。

## 二级分类模式

当为分类数据建模时。很多企业需要更灵活和更规范化的方法。在前一种模式中，每个实体都被划分成了类型，并将这些类型当作属性来存储。在本模式中将不会将类型当作属性来存储，而是将其单独作为一个实体。在语义上，这是在前一种模式基础之上的重大飞跃。这表明分类可以独立地存在，可以在其相关实体之外单独地管理分类数据。

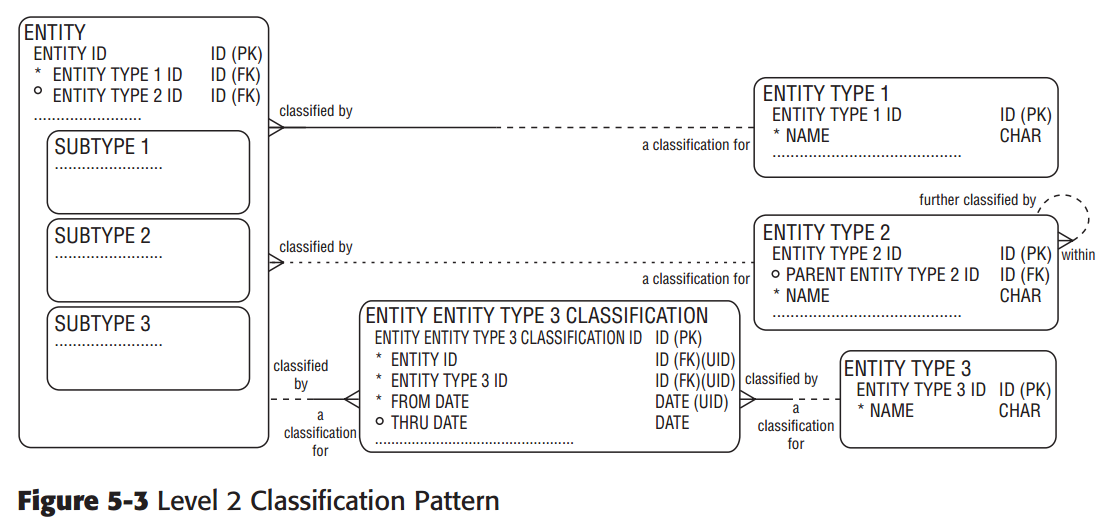
### 为何我们需要这种模式？

在这种模式中，通过将分类建模成实体，你可以创建、删除和管理这些不依赖于分类实体的“类型”。这种模式也提供了更多的灵活性，因为一个类型实体的相同实例可以对一个实体的多个实例进行分类，一个实体可以根据同一类型的多个实例来分类。例如，**当事人**[PARTY]可以根据**行业类型**[INDUSTRY TYPE]来分类，这个类型有其自身的实体来维护。因此，**当事人**[PARTY]实例可以与**行业类型**[INDUSTRY TYPE]的对个实例相关联（例如，“生产”和“销售”），同一**行业类型**[INDUSTRY TYPE]（例如，“生产”）可用于对多个**当事人**[PARTY]实例进行分类。

在前一个模式中我见到了一种非常特定的方法，将分类当作属性来建模。很多的遗留系统都是使用这种方式来实现的。二级分类模式提供了另一种方法，这种方法可以将具有缺点的一级模式架构转换成更加健壮的关系型模式，用以支持分类。一般情况下，这些模式真正的优点是，他们介绍了如何将特定模式转化为更加灵活的概括模式。

### 这种模式是如何进行工作的？

图5-3介绍的分类建模模式相比一级模式而言更加灵活，维护了较少的冗余信息。实体代表的是数据专业人士正在建模的任何数据，例如，**当事人**[PARTY,]、**产品**[PRODUCT]、**固定资产**[ASSET]、**工作计划**[WORK EFFORT]或**订单**[ORDER]。实体可以被分类成多个子类，用**子类1**[SUBTYPE 1]、**子类2**[SUBTYPE 2]和**子类3**[SUBTYPE 3]来标示。还可以使用不同的方式对实体进行分类，比如可以通过**实体类型1**[ENTITY TYPE 1]、**实体类型2**[ENTITY TYPE 2]和/或**实体类型3**[ENTITY TYPE 3]来分类，如有必要还可以添加其他的实体类型。



**图5-3** 二级分类模式

尽管使用子类和“类型”实体这两种分类方式来对一个实体进行分类，可能看起来很多余，但是通常类型建模需要这两种方式都要用到。子类可以容纳其他的额外的属性和关系，可能某一特定子类需要他们。例如，将**产品**[PRODUCT]分为子类：**货物**[GOOD]或**服务**[SERVICE]，那么**货物**[GOOD]就可能具有一些特定的属性和/或关系，比如说具有相关的**库存项**[INVENTORY ITEM(s)]，而**服务**[SERVICE]却没有这个属性。可以有一个类型实体比如说**产品类型**[PRODUCT TYPE]，该类型可以有两个值“货物”和“服务”，这个实体也可以具有特定的属性和/或关系。例如，产品类型可以与**折扣**[DISCOUNT]相关，该实体存储了指定时期内某产品类型的折扣信息。因此，“类型”实体也是需要的。

在这种模式中，你可以看出“每个**实体**[ENTITY]必须只能通过一个**实体类型1**[ENTITY TYPE 1]来进行分类”，“每个**实体**[ENTITY]可以只通过一个**实体类型2**[ENTITY TYPE 2]来进行分类”。这表明对于**实体**[ENTITY]，有些分类是强制性的，而有些却不是。例如，工作计划[WORK EFFORT]必须通过**工作计划类型**[WORK EFFORT TYPE]（比如说“项目[Project]”、“任务[Task]”或“活动[Activity]”）来分类。在同一模型中，或许“类型”实体也可以有一个可选的关系，**工作计划**[WORK EFFORT(s)]可以通过**工作计划用途类型**[WORK EFFORT TYPE]（比如说“维修[Maintenance]”、“调查研究[Research]”等。）来多工作计划进行分类。由于在**工作计划**[WORK EFFORT]的开始阶段没有给**工作计划**[WORK EFFORT]指派用途8，所以**工作计划**[WORK EFFORT]与**工作计划用途类型**[WORK EFFORT PURPOSE TYPE]之间的关系可以不是强制性的。

**实体类型3**[ENTITY TYPE 3]为**实体**[ENTITY]和“类型”实体之间的多对多关系提供了可能性。通常情况下，可能是通过同一**实体类型3**[ENTITY TYPE 3] 的多个实例对**实体**[ENTITY]实例进行分类的，但是也有可能**实体类型3**[ENTITY TYPE 3]的一个实例对多个**实体**[ENTITY]实例进行分类。例如，一个组织可以被若干个行业类型所分类，比如说一个公司可以同时归类于生产制造公司、电信公司和服务公司。在此案例中，**行业类型**[INDUSTRY TYPE]是**实体类型3**[ENTITY TYPE 3] 的一个示例，一个特定**组织**[ORGANIZATION]可以有三个**行业类型**[INDUSTRY TYPE]实例（生产制造业、电信行业和服务行业）。**行业类型**[INDUSTRY TYPE]的每个实例（比如说“生产制造业”）皆可应用于多个**组织**[ORGANIZATION]的实例。图5-3展示的是使用**实体与实体类型3分类**[ENITTY ENTITY TYPE 3 CLASSIFICATION]解决多对多关系。这种（已解决）多对多关系能够让一个**实体**[ENTITY]实例被多个**实体类型3**[ENTITY TYPE 3]的实例所分类，而且每个**实体类型3**[ENTITY TYPE 3]实例可以对多个**实体**[ENTITY]实例进行分类。

尽管这一模式只展现了三种行业类型，一种是强制性关系、一种是非强制性关系还有一种是多对多关系，但是这种模式的意图是你可以具有无数的对实体进行分类的 “类型”实体 — 有些是强制性的。一些是可选的以及多对多关系型的，具体使用哪种是需求而定。

**注：从某种角度上来看，有两种类型的分类：**

* **互斥分类— 一个产品可以是“硬件”、“软件”或“配件”，但是不可以在同一时间具有多个分类。**
* **此类型的分类可以在同一时间具有多个分类— 一个产品可以在同一时间具有多个产品系列（例如，一个产品可分为“家用”产品和“商用”产品）。**

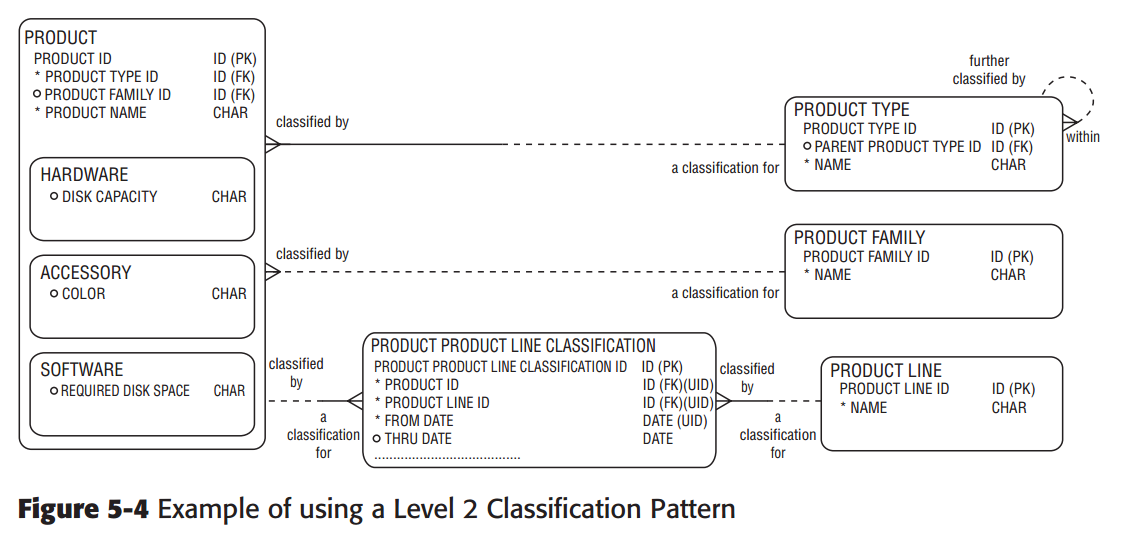
**第一种类型通常会产生一对多关系，从“类型”关联到相关的“实体”，第二种类型会牵涉到多对多关系。**

有些“类型”实体是可以递归的，换句话说，类型还可以包含类型。例如，**工作计划类型**[WORK EFFORT TYPE]实体可以具有实例“项目”和“活动”，然后“活动”实例可以进一步分解成“任务”和“工作”（或任何企业了解的分解方式）等。这种递归关系展示了“类型”实体实例是如何关联到其他“类型”实体实例。有些“类型”可能只有一组同一层次上的值，而不是随各种值进行了分级，这样他们便不需要在该“类型”实体上有一个递归关系。因此，为了表示这种关系是个可选择的选项，**实体类型2**[ENTITY TYPE 2]具有一种递归关联，可以将上一级实体类型2标识作为该实体的外键，该外键将一个实例关联到了另一个示例**实体类型2**。9

**注：在每个“类型”实体中我们只维护一个属性（例如，名称），在图5-3中，该类型的实体还可以捕获很多其他属性，比如说描述[description,]、简称[short name]、编码[code]、起始日期[from date]、终止日期[thru date]、当前称呼[current indicator]【注释1】等等。**

图5-4进一步介绍了这种模式的使用方法。我们可以继续前一节所示的案例。名称为Euro-Electronics的软硬件零售商对其数据进行了分类，将这些数据作为数据仓库项目的一部分。基于同管理人员和销售人员的会谈，以及二级分类模式的使用，数据专业人士制作了图5-4。

**（图）**



**图5-4** 二级分类模式的应用示例

数据专业人士发现产品实体具有三个主要子类，分别是：**硬件**[HARDWARE]、**配件**[ACCESSORY]和**软件**[SOFTWARE]。每个子类所维护的数据都是特定的。

数据专业人士先前发现了三种**产品**[PRODUCT(s)]分类的方法。你可以从图5-4中看到这些方法，分别使用了实体**产品类型**[PRODUCT TYPE]、**产品族**[PRODUCT FAMILY]和**产品系列**[PRODUCT LINE]。每个对产品进行分类的方法代表了一种理念，欧洲电子零售[Euro-Electronics]商使用这些理念管理、分析以及汇报其**产品**[PRODUCT(s)]。图5-4中的分类延用了图5-3所示的基本模式。因此，可以将某一特定**产品**[PRODUCT]分类为**产品类型**[PRODUCT TYPE]比如说“硬件[Hardware]”、**产品族**[PRODUCT FAMILY]比如说“磁盘驱动器[Disk drives]”或**产品系列**[PRODUCT LINE]比如说“商用[Commercial Use]”。

**产品类型**[PRODUCT TYPE(s)]是Euro-Electronics各部门最常用到的分类方法。你可以从图5-4中看出，产品[PRODUCT]有三个子类（**硬件**[HARDWARE]、**软件**[SOFTWARE]和**配件**[ACCESSORY]），Euro-Electronics正是使用这三个子类对产品信息加以分类（和管理）。我们之前就已说明之所以捕获这些子类（硬件、软件和配件）是因为他们有其自身的属性和关系。但是将“硬件”、“软件”和“配件”作为**产品类型**[PRODUCT TYPE]的实例来处理也很有意义。这种方式可以允许其他实体与**产品类型**[PRODUCT TYPE]相关联。例如，可能会有与产品类型相关并且依赖于产品类型的**价格组成规则**[PRICE COMPONENT RULE]实体或**规则**[REGULATION]（关于**产品组成规则**更多的信息请参看第8章节）。图5-5展示的是**产品类型**[PRODUCT TYPE]的其他的实例。

**（图）**

**图5-5** 产品类型

**注：此处我们使用了拇指规则：如果一个实体具有子类，且同时为其他“类型”实体或“分类”实体所分类，那么那些子类几乎总是“类型”实体或“分类”实体的实例。**

在图5-4中，**产品类型**[PRODUCT TYPE]旁边有一个递归关联。换句话说，“可以将**产品类型**[PRODUCT TYPE(s)]进一步分类成一个或多个产品类型，且每个**产品类型**[PRODUCT TYPE]只能在一个**产品类型**[PRODUCT TYPE]之内”。Euro-Electronics需要在多个层级上对产品进行分类。例如，其希望在第层级上展示的类型有：“处理器[Processors]”、“存储装置[Processors]”和“商业应用软件[Business Applications Software]”等，再是高层级上的类型，比如说，“硬件[Hardware]”、“软件[Software]”或“配件[Accessory]”。这样可以帮助高级管理人员回答如下问题：“硬件键产品去年的销售额是多少？”或“预计明年存贮装置的销售额所占硬件销售额的百分比是多少？”

在图5-5和表5-2中你可以看到**产品类型**[PRODUCT TYPE(s)]的三个层级，在这些层级中你可以将**硬件**[HARDWARE]产品分类成“处理器”或“存储设备”；将**软件**[SOFTWARE]产品分类成“商业应用软件”或“游戏软件”；或将**配建**[ACCESSORY(s)]分类成“包装箱”或“鼠标垫”。换句话说，这种模式支持将这些分类组成这些类型的层级架构或聚合架构。9

**表5-2** 类型的分级示例

**（表）**

你可能会问是否产品只能被最底层的**产品类型**分类。例如，一个**产品**不应当直接使用“硬件”来分类，而是通过“处理器”或“存储设备”来分类。答案是否定的，根据数据模型，这并不表示产品的分类必须使用最底层的**产品类型**。然而，我们认为作为通用的规则，它应当是如此。例如，如果你将一个产品“Acme视频卡”归类于“硬件” 而不是归类于“处理器”，那么你是否能够回答下面这个问题：“处理器销售额占硬件销售额的百分比是多少？”还存在以下案例：一个新产品没有合适的产品类型。例如，一个称为“Laser Pen 4000”的新产品。这个产品是个配件，但是它既不是一个包装箱或一个鼠标垫，也不适合作为这个新产品的**产品类型**实例。你是否应该通过“配件”来将该产品分类？你可以这样做，但是我们建议你创建一个新的较低层次的产品类型实例“Laser pens”，然后通过添加“Laser pens”实例的上级产品类型标识，此标识将这个产品关联到“配件”实例，从而创建了“Laser pens”到“配件”的关系。然后你可以将“Laser Pen 4000”指派为这个新**产品类型**实例。

你可能还想知道是否**产品**[PRODUCT]子类**硬件**[HARDWARE]、软件[SOFTWARE]和**配件**[ACCESSORY]应当还可以有较低层次的子类，比如说**硬件**[HARDWARE]低下可以有子类**处理器**[PROCESSOR]和**存储设备**[STORAGE DEVICE]。我们的拇指股则是指如果这个类型没有任何特定的属性和关系，那么我们将不把该类型作为子类来处理，相反我们会将其作为“类型”实体的实例。在这个示例中，**硬件**[HARDWARE]、**软件**[SOFTWARE]和**配件**[ACCESSORY]有其自己的属性和/或关系，而“处理器”、“存储设备”和其他低层次的类型却没有任何属性和关系。他们只是Euro-Electronics将其产品分类所使用的一种方法。出于这个原因，我们将不把他们作为**硬件**[HARDWARE]、**软件**[SOFTWARE]或**配件**[ACCESSORY]的子类来展现。

**产品族**[PRODUCT FAMILY]（如图5-6所示）一种不同的方法，Euro-Electronics想用该方法对产品进行分类。根据物理相似性和功能相似性，这个分类对产品加以归类。例如，产品可以被分成以下类别，比如说，“磁盘驱动器[Disk Drives]”、“运输包装箱[Carrying Cases]”、“计算机存储器[Computer Memory]”“台式机[Desktop Computers]”和“便携式笔记本[Laptop Computers]”。销售人员通过**产品族**[PRODUCT FAMILY]的分类分解产品以满足客户的不同需求。例如，销售人员可以引导一家公司查询磁盘驱动器[Disk Drives]。所以，他们可以查寻到所有被分类为**产品族**[PRODUCT FAMILY]“磁盘驱动器[Disk Drives]”的**产品**[PRODUCT(s)]，并且他们可以介绍“存储盘2000”，这是一个非常流行的产品，属于**产品族**[PRODUCT FAMILY]的分类“磁盘驱动器[Disk Drives]”。

图5-6 产品族

**产品族**[PRODUCT FAMILY]分类是可选的，正如图5-4所示，因此，它的模型可以使用一个可选的外键，产品族标识。这样做是因为有时一个**产品**[PRODUCT]没有通用的相关产品族。

根据使用产品的方法，**产品系列**[PRODUCT LINE(s)]（如图5-7）是另一种Euro-Electronics想使用的产品分类方法。例如，通过“家用”、“商用”和“政府用”，你可以看出**产品系列**[PRODUCT LINE(s)]适于利润中心是紧密相关的，在Euro-Electronics内部，利润中心通常是用于财务报表的。关于产品系列有意思的是**产品**[PROIDUCT]的同一实例可以使用很多的**产品系列**[PRODUCT LINE(s)]来对其进行分类。例如，“存储盘2000”既可以属于“家用”也可以属于“ 商用”。在图5-4中，通过使用关联实体产品与产品系列分类解决了多对多关系。“可以通过一个或多个**产品与产品系列分类**[PRODUCT PRODUCT LINE CLASSIFICATION(s)]来对**产品**[PRODUCT]进行分类，每个**产品系列**[PRODUCT LINE]可以是一个或多个**产品与产品系列分类**[PRODUCT PRODUCT LINE CLASSIFICATION(s)]的一个分类。”

表5-7 产品系列

表5-3进一步介绍了这种模式的使用方法。**产品**“存储盘2000[Save Disk 2000]”是根据三种方法来进行分类的—第一是通过**产品类型**，“存储设备[Storage Devices]”类来进行分类；然后是**产品族**[PRODUCT FAMILY]，“磁盘驱动器”；最后则是产品系类：“家用[Home Use]”和“商用[Commercial Use]”。以类似于“存储盘2000[Save Disk 2000]”的分类方法，“Scanner Disk Fob”被归类于“存储设备”和“家用”类别，只是“Scanner Disk Feb”没有根据任何的**产品族**[PRODUCT FAMILY]来进行分类。如果库存管理人员希望查看各种“硬件”产品，管理人员是可以要求将全部产品归类于“硬件”这一类别的，这以点是可以根据产品类型的递归关系推断出来的，在此递归关系中“硬件[Hardware]”是**产品类型**：“存储设备[Storage Devices]”和“处理器[Processors]”的父类别，如表5-2所示。还有另外一个示例“Carry All Case”。这个案例也是使用三种方式来进行分类的—第一种是通过**产品类型**，“包装箱[Cases]”；然后是通过**产品族**，“装运箱[Carrying Cases]”;最后是通过**产品系列**，“家用[Home Use]”和“商用[Home Business]”来进行分类的。

**（表）**

**表5-3** 二级类型模式示例

### 该在何时使用这种模式？

* 我们在以下情况下使用这种数据模型模式：  
  **当需要一种模型帮助理解数据需求时，以及将该模型作为声明范畴的一部分。我们发现这种模式很有用，有助于记录数据建模工作的范围，收集和验证数据需求，因为这种模式介绍了不同的方式，一种相对而言较为直白的方式来对实体进行分类。**它将不同的分类作为单独的实体，并展示了相关实体的关系属性。例如，在这个模式中有些分类可以是可选的，有些则可以是强制性的，有些可以是一对多的递归关系，而有些可以以多对多的方式与他们的分类实体相关联。
* **当使用规范化且很特别的方式对分类进行建模很重要的时。**跟本章节中的第一种模式不一样，这种模式不存在冗余数据（也就是说它是规范化的），当它仍然是为分类建模的特别方式（在下一个模式中，将介绍如何以更概括的方法为分类建模）。
* **需要独立地维护分类数据。企业通常将分类数据作为参考数据。**关于参考数据的管理很多企业都有流程和政策。通过将分类数据从另一个实体内部分离出来并将其作为一个独立实体，这样更易于管理。
* **当各种分类类别有其自己的属性和/或关系时。**因为这种类型的模式特地将分类的数据从数据中分类了出来，我们可以添加更多的语义到模型中。例如，**产品系列**[PRODUCT LINE(s)]可以有简称这个属性而**产品类型**[PRODUCT TYPE]且不可以有，或一个**产品**必须只能为一个**产品类型**[PRODUCT TYPE]所分类，但是却可以为一个或多个**产品与产品系列类型**[PRODUCT PRODUCT LINE CLASSIFICATION(S)]所分类（其中的每个实例都与**产品系列**[PRODUCT LINE]相关）。
* **当对不同的分类类型有良好的理解并且这些分类类型是静态的时候。**如果这些分类是非常稳定的，不需要改变的，这种模式将会是个很好的选择。
* **当不同的分类类型之间存在关系时。**例如，市场部可能会希望**产品系列**[PRODUCT LINE(S)]与**产品类型**[PRODUCT TYPE(s)]相关联。将他们作为一个独立的实体，以便可以创建**产品系列**[PRODUCT LINE(s)]与**产品类型**[PRODUCT TYPE(s)]之间的关系。

### 这种模式有哪些缺点？

此模式的弱点如下：

* **不是非常灵活。**在将来可能会添加这些新类型的分类，或该类型与实体之间的关系发生该变，而这种模式是不能适应这些新类型的分类的。例如，假如数据专业人士发掘了一种新类型，比如说产品组[PRODUCT GROUP]，那么这个专业人员将不得不创建一个新实体（具有关系和属性）以支持该分类类型。此外。如果产品到产品类型之间的关系属性变成了多对多关系（或变成可选的而不是强制性的），那么将会需要更改数据模型以适应这种改变。出于这个原因，在动态的环境中，这种模式可能不适合作为实现数据库设计的基础。
* **由于是在单独的实体中维护分类，而不是在单一的实体中维护所有分类，所以这种模式优势会限制分析能力。**这种模式没有提供在一个单一实体内将数据关联到所有分类的方法。对于回答下面这个问题比如说哪类客户对哪类产品感兴趣（包含所有的产品分类方法），将各类分类维护在同一个实体中通常是很有用的。
* **对于维护类型，没有一个标准方法。**每种类型可以具有不同的属性和关系，并可能会使用不一致的约定比如说维护每种类型的不同属性。（这也可以看作是优点！）
* **不同‘类型’实体之间的特定关系可能会导致模型更加难以管理。**如果不同类型之间存在多种关系，与使用通用的、灵活的方法来存储类型之间的关系相比，这种模式要复杂的多，比如说可以将不同的类型概括到一个单一的实体中，然后再使用递归关系，可以让所有的类型相互关联（将在下一个模式介绍这种方法）。
* **如果一个实体可以分为很多方面，然而这种模式难以管理这些分类。**有些实体（比如说当事人、产品、固定资产和工作计划）可能会有大量的分类类型。使用不同的方法来对同一实体进行分类取决于很多的方面，比如说各类数据外部数据提供者如何对其进行分类的，在不同的地理区域又是如何对其进行分类的，以及企业的不同机构又是对其是如何分类的。这种模式需要单独的实体和关系，以便使用不同的方式来对实体进行分类。

**简介**

在这个模式中，通过将分类信息从实体中分离了出来，我们强烈提出将分类数据作为一个独立的实体来维护但是还与该实体相关联。这表明一个企业可以管理分类数据、维护特定的数据以及各“类型”实体的关系，并且同一分类时日可以与数据模型中其他的实体相关联。

有些分类数据有其自身的结构。如图5-4所示的**产品类型**的递归关系。能够提供类型的聚合或分层架构通常是很重要的，因为很多企业要求数据可以基于他们的分类来“上下滚动”。你也可以看到可以或不可以有分类实体实例，以及通常有很多分类的数据实例，这些分类可以用于不同的目的。

在这一节中介绍了这种模式存在一些局限性。如果发觉新的分类，将不得不添加新的分类实体以及他们的关系。如果分类的性质发生了变化（例如，从一对多(1: M)关系变成了多对多(M: M)关系），这也需要更改数据模型。而这些更改在动态的环境中会产生负面影响。最后，尽管可以给模式添加分类类型，允许数据建模人员将不同的分类类型彼此关联，但是这种模式不能捕获不同分类类型之间的相互关系。

## 三级分类模式

很多企业需要非常灵活的方法来管理和捕获他们的分类，因为随着时间的推移实体可能会出现新的类型，随着时间的推移分类与实体之间关系本质也可能会发生变化。在前一模式中介绍了特定的分类实体。则会个模式将特定的分类实体概括进一个单一的包含各种类型的实体当中，当发生变化时，这种模式的灵活性和适应性要好得多。

### 为何我们需要这种模式？

很多企业当他们管理和查看业务的方式发生改变时，他们便添加和更改分类。所属企业利用其优势使用替代的分析方法来完成这点。做到这些需要用全新的革新方式来分析他们的数据。10例如，有些企业市场营销部为了更好地了解客户和市场，不断地提出新的方法，对产品和服务进行分类。多数企业需要多样的方法对实体进行分类。例如，通用数据模型的分配，我们正在处理的汽车分销权，有很多的方法对汽车进行分类（产地、型号、年份、引擎类型、变速器类型等等）。还有，很可能在将来需要添加新的分类。

企业可能需要简单的方法，在不必添加新实体、属性或关系的情况下，可以添加、更新和删除各种分类类型。对于大多数情况来说，当更改类型时，这种方法可以削减对现存的数据库、分析、项目和报告所造成的不良影响。三级分类模式提供了这样的灵活性。

### 这种模式是如何进行工作的？

取代为每个新“类型”提供单独的实体这种方式，三级分类模式使用一个单一的“类别”实体，该实体可以容纳任意数量的实体分类和“类别类型[category type]”，“分类类型[category type]”维护了各种随着时间推移而可能存在或出现的各种分类。例如，一个实体，比如说**产品，**可以具有很多种分类方法，或换而言之，通过不同的 “类别类型[category types]”来分类，即根据“产品类型[product type]”（“处理器[Processors]”、“商业应用软件[Business Application Software]”或“鼠标垫[Mouse Pads]”）；“产品族[product family]”（“磁盘驱动器[Disk Drives]”、“装运箱[Carrying Cases]”等）；“产品系列[product line]”（“家用[Home Use]”和“商用[Commercial Use]”等等）；或其他可能存在或可能将来会出现的分类类型来对产品实体进行分类。

这些类别也可以向上或向下滚动，进入各种较高级别的分类或较低级的分类，因此这种模式能够提供类别和类别类型的分层或聚合结构。例如，产品类别“硬件”可以进一步地分类成“存储设备”和“输入/输出设备”，产品类别“软件”则可以进一步地分类成“ 商业应用软件”和“游戏软件”。很多企业喜欢评估较高层级上的产品，然后钻研细节以支持分析的需求。同样的，类别类型也可以有分层结构（或聚合结构），例如，类别类型[category type]之“产品类型[Product type]”可以进一步分类成“产品子类别[Product subcategory]”。

在前一种模式中，特别将不同的类别作为一个独立的实体，例如如图5-3所示的**实体类型1**[ENTITY TYPE 1]、**实体类型2**[ENTITY TYPE 2]和**实体类型3**[ENTITY TYPE 3]。在图5-8中这些不同的分类类型被当作**实体类别类型**[ENTITY CATEGORY TYPE]的实例。换句话说，实体类别类型包含实例“实体类型1[ENTITY TYPE 1]”、“实体类型2[ENTITY TYPE 2]”和“实体类型3[ENTITY TYPE 3]”（以及将来可能出现的任何实体类型）或更特别的示例，“产品类型”、“产品族”和“产品系列”。那些分类类型的各个实例可以作为**实体类别**[ENTITY CATEGORY]的实例。例如，**产品类别类型**[PRODUCT CATEGORY TYPE]实例“产品类型[Product Type]”可以是产品类别实例：“硬件[Hardware]”、“软件[Software]”、“存储设备[Storage Devices]”、“游戏软件[Gaming Software]”以及图5-5所示的其他产品类型实例的一个分类类型。当新的需求出现时，这种模式可以动态地添加其他类型的分类类型和分类。例如，假如Euro-Electronics这家企业发掘了一个新的分类类别，叫做“产品细分市场[Product Market Segment]”，这个分类类别的值是“大众市场[Mass Market]”和“奢侈品市场[Luxury Market]”，这种模式会将“产品细分市场”作为一个**产品类别类型**[PRODUCT CATEGORY TYPE]，将“大众市场[Mass Market]”和“奢侈品市场[Luxury Market]”作为**产品类别**[PRODUCT CATEGORY]的实例。

**实体类别分类**[ENTITY CATEGORY CLASSIFICATION]将**实体**[ENTITY]交叉引用至**实体分类**[ENTITY CATEGORY]。**实体类别分类**[ENTITY CATEGORY CLASSIFICATION]捕获了**实体**[ENTITY]（你正试图分类的数据，比如说某一特定产品）的实例和相关的**实体类别**[ENTITY CATEGORY]（一个分类，比如说“硬件[Hardware]”）的一个实例。可以使用多种方式（例如，“存储设备[Storage Device]”、“磁盘驱动器[Disk Drive]”和“家用[Home Use]”）对实体的一个实例（例如，产品“存储盘2000[Save Disk 2000]”）进行分类，可使用**实体类别**[ENTITY CATEGORY]对**实体**[ENTITY]的多种实例（例如，类别“硬件”可以适用于多种产品）进行分类。**实体类别分类**[ENTITY CATEGORY CLASSIFICATION]解决了实体和实体类别之间的多对多关系。该模型说明了“可以使用一个或多个**实体类别分类**[ENTITY CATEGORY CLASSIFICATION(s)]对每个**实体**[ENTITY]进行分类，每个**实体类别**[ENTITY CATEGORY]可以用于定义一个或多个**实体类别分类**[ENTITY CATEGORY CLASSIFICATION(s)]”。

（图）

**图5-8** 三级分类模式

需要注意的是**实体类别**[ENTITY CATEGORY(s)]通过**实体类别类型**[ENTITY CATEGORY TYPE]来分类。这样的话，为何你需要维护分类的类型呢？换而言之，**实体类别**[ENTITY CATEGORY(s)]与**实体类别类型**[ENTITY CATEGORY TYPE(s)]之间存在哪些差异？**实体类别**[ENTITY CATEGORY(s)]是指**实体**[ENTITY]可能具有的各种分类，而**实体类别类型**[ENTITY CATEGORY TYPE(s)]是指“分类”或名称，描述此类别是哪种类型的分类。例如，**工作计划**[WORK EFFORT]（一个实体）可以分类为“项目[Project]”或“任务[Task]”，“项目[Project]”和“任务[Task]”是**工作计划类别**[WORK EFFORT CATEGORY(s)]（**实体类别**[ENTITY CATEGORY(s)]）。这些类别是在分类类型“工作计划类型[Work Effort Type]”内（对工作计划进行分类的一种方法），“工作计划类型[Work Effort Type]”是指**工作计划类别类型**[WORK EFFORT CATEGORY TYPE]（**实体类别类型**[ENTITY CATEGORY TYPE]）。还可以有其他**工作计划类别类型**[WORK EFFORT CATEGORY TYPE]实例，比如说“工作计划用途类型[Work Effort Purpose Type]”，代表了对工作计划进行分类的一种方法，通过工作计划的用途来分类，比如说这个计划是“调研[Research]”工作计划还是“生产[Manufacturing]”类工作计划。“调研[Research]”和“生产[Manufacturing]”是**工作计划类别**[WORK EFFORT CATEGORY(s)]的实例。

区分**实体类别**[ENTITY CATEGORY]和**实体类别类型**[ENTITY CATEGORY TYPE(s)]的一种有效方式是想象一份报表。**实体类别**[ENTITY CATEGORY(s)]是这份报表的列，而**实体类别类型**[ENTITY CATEGORY TYPE(s)]则是位于那些列之上的列头（在报表中也称作断点）。实体因而是直接由**实体类别**[ENTITY CATEGORY]（报表中的列）来分类的而不是为**实体类别类型**[ENTITY CATEGORY TYPE]所分类。例如，**产品类别类型**[PRODUCT CATEGORY TYPE]“产品族[Product Family]”报名中列的部分可以是产品类别“磁盘驱动器[Disk Drives]”、“装运箱[Carrying Cases]”、“计算机存储器[Computer Memory]”、“台式机[Desktop Computers]”和“手提电脑[Laptop Computers]”。如果行是指每个产品，且在报表中的值则表示单位产品的销售量，这说明对产品进行分类是使用**产品类别**[PRODUCT CATEGORY(s)]而不是**产品类别类型**[PRODUCT CATEGORY TYPE(s)]。因此，**实体类别类型**[ENTITY CATEGORY TYPE]代表分类方法，而**实体类别**[ENTITY CATEGORY]维护了可能的值，这些值直接对某物进行了分类。**产品**[PRODUCT]可以归类于**产品类别**[PRODUCT CATEGORY]“手提电脑[Laptop Computer]”，而这种分类方式是位于**产品类别类型**[PRODUCT CATEGORY TYPE]“产品族[Product Family]”之内。

另一种形象化的方法是将**工作计划**[WORK EFFORT]想象成根据“工作计划类型[Work effort type]”之“项目[Project]”来分类的“数据仓库项目#123[Data warehouse project #123]”，此**工作计划**[WORK EFFORT]还可以根据“工作计划用途类型[Work effort purpose type]”来分类。你断不能说“数据仓库项目#123[Data warehouse project #123]”归类于“工作计划用途类型[Work effort purpose type]”或“工作计划类型[Work effort type]”（这两者皆是工作计划类别类型）。但是你可以说“数据仓库项目#123[Data warehouse project #123]”是一个“项目[Project]”（**工作计划类别**[WORK EFFORT CATEGORY]），而“项目[Project]”是一个“工作计划类型”[Work effort type]（**工作计划类别类型**[WORK EFFORT CATEGORY TYPE]）。你还可以说“数据仓库项目#123[Data warehouse project #123]”是一个“发展计划[Development effort]”（**工作计划类别**[WORK EFFORT CATEGORY]），而“发展计划[Development effort]”是一个“工作计划用途类型[Work effort purpose type]”（**工作计划类别类型**[WORK EFFORT CATEGORY TYPE]）。因此，**实体类别**[ENTITY CATEGORY]与**实体**[ENTITY]相关联（通过**实体类别分类**[ENTITY CATEGORY CLASSIFICATION]），而**实体类别类型**[ENTITY CATEGORY TYPE]包含了你正在维护的分类类型。

正如前部分所示（在图5-5和表5-2中）类别类型可以有其自身的结构。**实体类别**[ENTITY CATEGORY]周围有递归关系来满足这一需求。**实体类别类型**[ENTITY CATEGORY TYPE]也可以有递归关系，用以支持**实体类别类型**[ENTITY CATEGORY TYPE(s)]的分层架构。

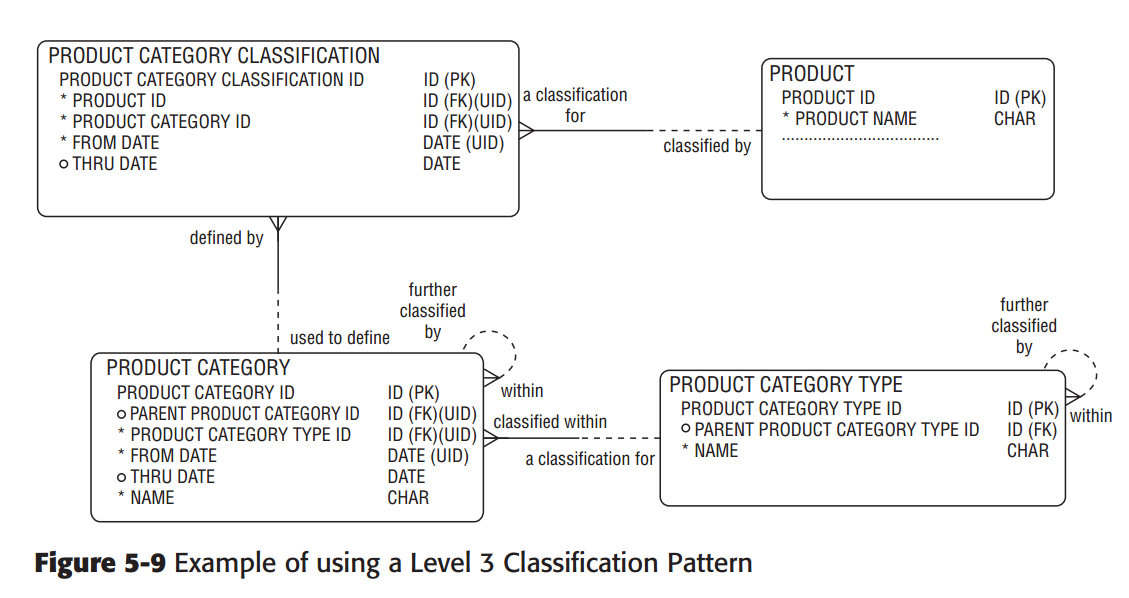
为何你需要分层的**实体类别类型**[ENTITY CATEGORY TYPE(s)]？有时展示相应的“分类”是很重要的，换句话说，就是各类别层级上的类别类型。**实体类别类型**[ENTITY CATEGORY TYPE(s)]没有层级结构，通常情况下这种说法是正确的，但是有时却并非如此。例如，你希望创建**实体类别**[ENTITY CATEGORY(s)]实例的分层架构，你想查看是否存在**实体类别类型**[ENTITY CATEGORY TYPE(s)]的分层架构，该架构形成了验证哪些实体类别[ENTITY CATEGORY(s)]可以隶属彼此的基础。换句话说，**实体类别**[ENTITY CATEGORY(s)]分层关系（或聚合关系）可能需要反映**实体类别类型**[ENTITY CATEGORY TYPE]分层关系（或聚合关系）。例如，将**产品类别类型**[PRODUCT CATEGORY TYPE]“产品类型[Product Type]”进一步分类成“产品子类别[Product Subcategory]”，可以有个业务规则：递归关联的**实体类别**[ENTITY CATEGORY]实例必须隶属于适当的**实体类别类型**[ENTITY CATEGORY TYPE]，而**实体类别类型**[ENTITY CATEGORY TYPE]的实例也可以以类似的方式分层。例如，你想检查下产品类别[PRODUCT CATEGORY]的递归关系以及实例“包装箱[Cases]”与“配件[Accessories]”之间的关联是否有效。“配件[Accessories]”可能会关联到产品类别类型[PRODUCT CATEGORY TYPE]的“产品类型[Product Type]”，而“包装箱[Cases]”则可能会关联到产品类别类型[PRODUCT CATEGORY TYPE]的“产品子类别[Product Subcategory]”。最后，产品类别类型的递归关系展示了“产品类型”的实例是实例“产品子类别”的父类别。因此，你可以得出结论：“产品类别”由类型“产品子类别”构成（例如，“包装箱”）是成立的，而产品类别具有类型“产品类型”（例如，“配件”）。这样有助于阻止向上滚动的类别，这些类别是完全不同的分类类型，例如，如果将“鼠标垫”（产品类型）隶属于“商用”（产品系列），可能会引起混淆。这也可以用于确保低层级的类别适当的滚动至上一级，且不遗漏一级。

**注：将实体类别类型[ENTITY CATEGORY TYPE]中的实例分层的另一个原因是可以创建分类，这些分类对于报告或引用是很有用的：例如，可以展示“产品类型”以及其相关的“产品子类”。**

尽管分层（或聚合）的**实体类别类型**[ENTITY CATEGORY TYPE(s)]可以作为验证相应**实体类别**[ENTITY CATEGORY(s)]的数据，但是数据并不特地执行规则“**实体类别**[ENTITY CATEGORY(s)]分层必须遵循**实体类别类型**[ENTITY CATEGORY TYPE(s)]的特定分层”。这种分层将不得不通过业务规则来获得和依靠外部代码来实现，或是通过使用本书第8章节所介绍的业务规则模式。数据模型所做的事是存储参数，这些参数展现了什么样的**实体类别类型**[ENTITY CATEGORY TYPE(s)]可以彼此隶属，以便可以指导什么样的**实体类别**[ENTITY CATEGORY(s)]可以隶属于彼此。换句话说，你还可以通过检查相关的实体类别类型[ENTITY CATEGORY TYPE(s)]是否具有隶属关系核实**实体类别**[ENTITY CATEGORY(s)]的层级结构（或聚合结构）。

**注：图5-8中的两个递归关系是一对多关系。一对多关系支持分层结构和聚合结构，在此结构中子实例只可以与一个父实例相关联。可能会需要多对多分层、聚合和对等关系，将在本章节的下个部分，增强这种模式以满足这一需求（如图5-12所示）。另一个方案是使用第4章节中已有的递归模式替代一对多关系。**

为了进一步解释这种模式，我们续用前部分所介绍的案例。Euro-Electronics的首席信息官表示公司所存在的最大的IT问题之一是市场营销部的人员不断更改方法，他们希望对产品信息进行分割和审查。例如，他们不断地提出新的分析产品信息的方案，比如说各种产品组、分组、用途、材料类型等等。他们所添加、更改或废除的新类型的类别可能需要新的实体，最后便是新增（或更改）表。这个问题已变成了他们部门捉襟见肘的编程人员的头痛事。市场销售部被迫雇用内部编程人员来执行运行他们的报告。首席信息官想要有一个灵活的架构，可以减缓编程人员的压力，同时还可以满足市场销售部的需求。基于这种需求和三级模式的使用，数据专业人士制作了图5-9.



**图5-9** 三级分类模式应用范例

图5-9捕获了与图5-4一样的数据类型，然而，却是以更加灵活的方式。在图5-4中，**产品**是如何具有三个分类类型的：**产品系列**[PRODUCT LINE]、**产品类型**[PRODUCT TYPE]和**产品族**[PRODUCT FAMILY]。这些分类类型反应了Euro-Electronics对**产品**进行分类的不同方法。例如，同一个产品可以归类于**产品类型**“存储设备[Storage Devices]”，可以归类于**产品系列**“家用[Home Use]”，以及归类于**产品族**“磁盘驱动器”[Disk Drives]。在此模式中，不同类型的分类（**产品系列**[PRODUCT LINE]、**产品族**[PRODUCT FAMILY]和**产品类型**[PRODUCT TYPE]）被当作**产品类别类型**[PRODUCT CATEGORY TYPE]的实例。

在图5-9中，各种**产品**[PRODUCT(s)]可以通过**产品类别分类**[PRODUCT CATEGORY CLASSIFICATION]直接分为多种**产品类别**[PRODUCT CATEGORY(s)]。每种**产品类别**[PRODUCT CATEGORY(s)]都代表一个分类，该分类是在特定类型分类的上下文中，例如在**产品类别类型**[PRODUCT CATEGORY TYPE]中。例如，在表5-4的前四行“存储盘2000[Save Disk 2000]”被分为四个产品类别，属于不同的类别类型，其名称分别是：“存储设备[Storage Devices]”，其**产品类别类型**[PRODUCT CATEGORY TYPE]是“产品类型[Product Type]”；“磁盘驱动器[Disk Drives]”，其**产品类别类型**[PRODUCT CATEGORY TYPE]是“产品族[Product Family]”； “家用[Home Use]”和“商用[Commercial Use]”，其产品类别类型是“产品系列[Product Line]”。在随后的四行中“Carry All Case”产品被分类成“配件[Accessory]”、“装运箱[Carrying Cases]”、“商用[Commercial Use]”和“家用[Home Use]”等产品类别，每一类别均与各自的类别类型相对应。

市场销售部希望能够很容易的更改**产品**[PRODUCT(s)]的类别。例如，市场部认识到“Carry All Case”产品的主要销售和使用对象是非商业客户，因而将其归类于**产品系列**[PRODUCT LINE]中的“家用[Home Use]”这一类别。换句话说就是自2009年2月3号开始他们不再将此产品归类于**产品系列**[PRODUCT LINE]中的“商用[Commercial Use]”这一类别。为了支持这个业务功能，**产品类别分类**[RODUCT CATEGORY CLASSIFICATION]使用起始日期和终止日期这两个属性来维护何时某一特定产品归类于某个特别**产品类别**[PRODUCT CATEGORY]的，而什么时候又不属于此**产品类别**[PRODUCT CATEGORY]的。在表5-4中，“Carry All Case”此产品在2009年2月3号之前（包含2月3号这天）一直属于“商用[Commercial Use]”这一**产品类别**[PRODUCT CATEGORY]（其**产品类别类型**[PRODUCT CATEGORY TYPE]是“产品系列[Product Line]”），然后至2009年2月4号开始，此产品被认为属于产品系列中的“家用[Home Use]”这一类别。

Euro-Electronics企业的市场销售部也希望能够灵活地将不同的**产品类别**[PRODUCT CATEGORY(s)]“隶属”于其他的**产品类别**[PRODUCT CATEGORY(s)]中。例如，表5-4所示，“存储盘2000[Save Disk 2000]”被归类于“存储设备[Storage Devices]”，而“存储设备[Storage Devices]”是在其上级**产品类别**[PRODUCT CATEGORY(s) “硬件[Hardware]”中，而“硬件[Hardware]”又被归类于**产品类别类型**[PRODUCT CATEGORY TYPE]的“产品类型[Product Type]”中。（“存储设备[Storage Devices]”是在**产品类别类型**[PRODUCT CATEGORY TYPE]中的“产品子类别[Product Subcategory]”当中）。同样地，“Carry All Case”产品被归类于“包装箱[Cases]”，而包装箱[Cases]是在其上级**产品类别**[PRODUCT CATEGORY]“配件[Accessory]”中。在业务中出现多层次类别是非常正常的情形。通常管理人员从多个的角度来分析。管理人员可能想根据需求‘纵向探索（或横向）’‘细节/总结’。在这个范例中，销售市场人员可能会问“配件的年销售额是多少？”，然后才是“包装箱”，此数据模型支持这种类型的查询方式。

**表5-4** 产品类别和类别类型

**（表）**

这种模式的主要区别在于添加新类型的类别要容易的多，只要添加其他的实例就行了而无需更改数据模型。表5-4的最后两行显示了市场销售部希望使用其他的方法对产品进行分类，例如根据一个新的分类类型“产品组[Product Group]”来分类，这种模式可以在不需要改变模型的情况下容纳此类别类型以及任何其他的新类别类型。表5-4展示了“存储盘2000[Save Disk 2000]”可以作为“便携设备[Portable Device]”归类于新的**产品组**[PRODUCT GROUP]类别中，“Carry All Case”产品可以作为“半便携设备[Semi-Portable Device]”。因此，随着时间的推移当出现其他的需求时，此数据模型（和任何基于此模型的数据库架构）支持添加新的产品分类方式。

这种模式也能够添加其他的分层或聚合的分类。例如，从图5-10和表5-5中可以看出将产品类别[PRODUCT CATEGORY(s)]汇合进损益类别中，该类别有“计算机[Computers]”和“计算机相关设备[Computer-Related Equipment]”，还有“政府企业[Government Business]”和“非政府企业[Non-Government Business]”。

**（图）**

**图5-10** 损益类别

为何Euro-Electronics想要这么分类呢? Euro-Electronics的高级管理人员想根据“计算机[Computers]”的销售收益与“计算机相关设备[Computer-Related Equipment]”的收益来获得损益信息，并且他们还想基于“政府企业[Government Business]”与“非政府企业[Non Government Business]”查询出收益。图5-10展示了如何将“产品系列[Product Lines]”和“产品族[Product Families]”中的一些类别和根据需求所创建的新类别结合起来完成这项工作。因此，只需通过添加新的类别，然后创建聚合架构，该团队可以准确的提供管理人员所需的分类，而不需要更改数据模型。表5-5展示的是如何应用图5-9中的模型捕获这些“损益报告”类别以及这些类别与其他类别之间的关系。从表中可以看出在2008年4月1号添加了如下这些新类别：“计算机相关设备[Computer-Related Equipment]”、“计算机[Computers]”、“政府企业[Government Business]”和“非政府企业[Non Government Business]”是，然后将其关联到其他现存的类别，这样是为了可以在不更改数据模型情况下获得他们所要的功能。

**表5-5** 三级分类模式范例，产品类别集合。

（表）

带符号\*的这一栏是“子类别”的**产品类别类型**[PRODUCT CATEGORY TYPE]名称，或者是说第二列的产品类别类型的名称，因为“父”类别和“子”类别可以关联到不同的**产品类别类型**[PRODUCT CATEGORY TYPE]。

这个模式可以让你添加类别，这些类别可能从不直接关联到**产品**的实例，但是可以将直接关联到**产品**的类别收集到有用的分类中，用于报告和分析。换句话说，通过创建类型为“损益报告类别[Profit and Loss Reporting Category]”的集合，企业可以根据需求提供一种便捷的方法，用于产品的一般的报告和分析。这表示Euro-Electronics可以回答以下这些问题：“计算机与计算机相关设备的收益/损失是多少？” 、“销售给政府企业与非政府企业的产品的收益/损失有多少？”**产品**除了可以根据这些类别来分类，还可以根据其他类别比如说“产品族[Product Family]”和“产品系列[Product Line]”来分类，所以企业还可以回答这样的问题，比如说“家用型的计算机的收益/损失有多少？”解决这个问题可以先统计出“计算机”的收益/损失，再通过**产品类别类型**[PRODUCT CATEGORY CLASSIFICATION]为“家用”的产品来过滤**这一结果。**

**注：虽然有些分层和聚合的类别可以很容易地添加到这种模式中，但是这种模式的一个主要局限性是每个字实例可能只能有一个上级。这个限制了分层和聚合类别的变异形，此模式可以维护这些类别。例如，Euro-Electronics想添加“产品类型[Product type]”之“处理器[Processors]”并将其作为“计算机相关设备”另一个子类别，这是不可能的，因为“处理器[Processors]”已经有了一个父实例“硬件[Hardware]”，而根据数据模型，其只能有一个父实例。在下一个模式中也就是维护隶属关系和方案的三级分类模式中我们将解决这一功能。**

需要注意表5-5，我们需十分谨慎，在同一产品类别的数据模型架构中不能将“产品系列[Product Line]”与“产品族[Product Family]”混淆。例如，我们不能将相同的产品类别中的“家庭业务[HomeBusiness]”（“产品系列”）与“手提电脑[Laptop Computers]”相混淆（“产品族”）。因为一个产品的实例可以使用“产品系列[Product Line]”和“产品族[Product Family]”这两种方法来分类。如果我们在的单一的产品类别[PRODUCT CATEGORY]中混淆了“产品系列[Product Line]”和“产品族[Product Family]”，我们可能会重复计算它们。

我们如何实现这种模式呢？在创建**产品类别**[PRODUCT CATEGORY(s)]的集合结构之前，如图5-10和表5-5所示，我们先创建一个“模版”架构，如图5-11和表5-6所示。在此架构中，你可以看出“损益报告类别”要么是“产品系列”的聚合要么是“产品族”的聚合。换句话说，Euro-Electronics是使用图5-11所示的聚合架构作为模版，用以确保图5-10和表5-5所示的损益类别的有效性。

为什么这么做呢？这种模式十分灵活，允许Euro-Electronics创建有效的**产品类别**[PRODUCT CATEGORY(s)]分层和聚合。一方面，这种模式允许自由混合类别，将其用于企业所需的各种类型的分析和报告。但是可能需要一些约束。表5-6表明了只有产品系列和产品族可以聚合在一起形成“损益报告类别”（并非“产品类型”）。因此，当企业想要添加新的“损益报告类别[Profit and Loss Reporting Category]”到图5-10时，它先要检查图5-11所示的模版，以确保添加类别层级是许可的。换句话说，当前的模版只允许将产品系列和产品族聚合在一起，创建新的损益报告类别。

**（图）**

**图5-11** 财务报表的类别类型的聚合

**表5-6** 三级分类模式范例，产品类别类型的聚合

**（表）**

### 应当在何时使用这种模式？

我们在以下情形使用这种模式：

* **当企业需要灵活的无需更改的数据模型时：**
* 需要或发掘了新类型的分类
* 分类类型与实体之间的关系性质发生改变时（比如说分类与实体的关系的技术或可选性）
* 需要删除过时的分类类型
* **当企业有很多种对实体进行分类的方法时。**一些实体（比如说当事人、产品、固定资产和工作计划）可以有很多的分类类型。一个实体可以以多种方式进行分类取决于很多的因素，比如说依据外部数据提供者来分类，依据不同的地理区域来分类以及企业的组成机构来分类。
* **需要有一个通用的模型来管理实体的各种类别和类别类型。**这种模式可以作为对实体进行分类的通用方法，企业可以为不同实体的类别建立一致的模型。此外，因为这些数据模型（和随后的数据库设计）可能非常相似，所以为类别建模有通用的规则。
* **为了提供更强大的分析功能，需要将类别结合在一起时。**有时结合类似的类别可以产生强大的分析性能。例如，将当事人的各种类别结合到**当事人类别**[PARTY CATEGORY]实体中，此外还将产品的各种类别结合到**产品类别**[PRODUCT CATEGORY]实体中，然后你可以在**当事人类别**[PARTY CATEGORY]和**产品类别**[PRODUCT CATEGORY]之间创建一个称作**市场取向**[MARKET INTEREST]的交叉引用实体，该实体维护了哪种类型的当事人对哪种类型的产品感兴趣（例如，“高收入”当事人可能对“高端、豪华的”产品感兴趣）。

### 该模式的缺点是什么？

此模式的弱点如下：

* **更难理解。**由于这种模式比二级分类模式要概括的多，所以它更加难以理解和交流。由于这种模型非常地概括，所以很难有效地使用模式本身来定义范围或用于手机和验证数据需求，比如说需要哪些类型的分类。
* **并不直接在数据模型中执行业务规则。**这种模型并未捕获特定的业务规则，比如说分类与实体之间的关系的基数或可选性。这个模型只表示相关联的实体可以被分类成多数类别，实体可以分类成不同的类别实例，每个类别和类别类型可以有任意数量的层级。这种模式为数据建模人员提供了偷懒的方法，捕获实体可能存在的所有类别，且不需要检查哪些类别是需要的以及每个类别的需求是什么。不同的类别具有不同的规则、关系和属性。即使某人使用这种模式建模，也需要分析和理解这些。
* **关于类别上卷的方式，数据模型没有维护明确的业务规则。**虽然此模型允许你存储**实体类别**[ENTITY CATEGORY]和**实体类别类型**[ENTITY CATEGORY TYPE]各层次上的数据，但是数据模型没有执行明确的规则以确保**实体类别**[ENTITY CATEGORY]的递归上卷方式与**实体类别类型**[ENTITY CATEGORY TYPE]的递归上卷方式相同。所以在产品类别类型的递归关系中，模型可能会维护“产品子类别[Product Subcategory]”隶属于“产品类型[Product Type]”。而在产品类别的递归关系中，“产品类型[Product Type]”隶属于“产品族[Product Family]”。这表明如果没有定义围绕此递归关系的业务规则和流程，那么这两个分层可能会不同步。（这个问题也可以通过第8章节所介绍的业务规则模式来解决）。
* **类别和类别类型的隶属的方式是有限的。**由于这种模式为类别与类别类型只提供了一对多关系，各种类型的分层结构和聚合结构是有限制的。还有可能会需要维护各种计划，而此模式不能满足这一需求。
* **如果只是几个‘类型’或‘类型’的数量是稳定的，则不需要这种模式。**如果只需要一种或两种实体的分类方法，或者实体分类方式的数目是非常稳定不会发生改变的，使用这种模式未免会大材小用了。
* **这种模式不能维护不同类型/类别的不同属性或特定类型的关系。**特定类型的类别可以具有该类别类型所独有的属性或关系。然而，你可以选择创建**实体类别**[ENTITY CATEGORY]和**实体类别类型**[ENTITY CATEGORY TYPE]的子类来处理这一问题。

**简介**

对于为何这种模式十分重要，有多种原因；但是，有两个是主要原因。其一，此模式提供了灵活性，可以在不更改数据模型的情况下，添加或更改新类型的类别。第二，它提供了单一的标准方法，可以以单一一致的方式处理实体的所有分类。

从表5-4中可以看出，可以将不同的**产品**[PRODUCT(s)]分为不同类型的类别。例如，可以将他们归类为产品系列：“家用[Home Use]”或“商用[Commercial Use]”，和/或将其归类于产品族；“磁盘驱动器[Disk Drives]”和/或“手提电脑[Laptop Computers]”，以及很多其他的分类类型，包括哪些随着时间推移而出现的新类型。使用这种模式，生成非常灵活的实现方案，企业可以在不改变模型的情况下添加或删除新的**实体类别**[ENTITY CATEGORY(s)]和**实体类别类型**[ENTITY CATEGORY TYPE(s)]。

由于此模式的概括特性，我们不能将它作为非技术受众声明范畴的一部分，也不能用于收集或验证数据需求。在这种模式中，将不同的类别类型当作实例，而非当作实体和属性。我们还要劝诫数据专业人员在使用这种模式时不要偷懒。虽然这种模式可以捕获所有的类别类型，但是我们不能够不分析需要什么样的类别以及这些类别的特性。因为你是以相同的方法捕获了所有的分类，你将不能够捕获某些分类类型的特定属性和关系。可以通过创建**实体类别**[ENTITY CATEGORY]和**实体类别类型**[ENTITY CATEGORY TYPE]来解决这个问题。

## 使用隶属关系和分类方案的三级分类模式

可能会有内部或外部机构指定的方案或分类。例如，金融服务数据提供者，比如说彭博资讯或路透社，提供了金融证券的分类。另一个示例是政府，政府提供了行业类型的分类，比如说标准的工业分类（SIC）模式，而不同的政府提供不同的分类方案是很常见的。

此外，分类通常是通过多种方式进行相互关联的，通常分类和分类类型具有多对多关系。9例如，可能会有个产品分类“办公产品[Office product]”，而该分类下又有子类别：“计算机零配件[Computer parts]”、“办公用品[Office supplies]”和“办公机械[Office machines]”。另一个分类“零售产品[Retail product]”也可以进一步地分类成“计算机零配件[Retail product]”和“办公用品[Office supplies]”但是没有“办公机械[Office machines]”。因此，“计算机零配件[Computer parts]”和“办公用品[Office supplies]”属于不同的上级分类，分别隶属于“办公产品[Office product]”和“零售产品”[Retail product]。分类多样隶属（或分解）方式是企业用于全面挖掘他们内部数据的路径，因而十分重要。

### 为什么我们需要这种模式？

本节的模式支持数据分类标准方案的管理。此模式支持分类方案起源处的关系，那些分类类型是该方案的成员。例如，金融机构可能会需要知道源自于某一特定的数据提供者的金融证券的分类，或源自于某一特定评级机构的某个评级分类。

这种模式还要支持创建类别之间以及类别类型之间的多对多关系这样的需求。这种模式还要能够创建“模版”分类结构，这些“模版”结构有助于确定分类数据。

### 这种模式是如何进行工作的？

在图5-12中显示“只能使用一个**实体类别类型方案**[ENTITY CATEGORY TYPE SCHEME]来组织**实体类别类型**[ENTITY CATEGORY TYPE]，而每个**实体类别类型方案**[ENTITY CATEGORY TYPE SCHEME]可以是一个或一个以上**实体类别类型**[ENTITY CATEGORY TYPE]的方案。”可以将实体类别类型方案定义为“各元素的结合……通过设计可以连接、调整和整合这些元素”11，可用于实体类别类型的分类。

**实体类别类型方案**[ENTITY CATEGORY TYPE SCHEME]捕获了方案的名称，或者换句话说，该方案就是通常所说的“标准行业分类（SIC）”方案，此方案是根据组织所在的行业对组织机构进行分类。在此模式中，可能没有明确地捕获或使用该方案架构以及该方案的所有元素。模型将某一方案与一个或多个**实体类别类型**[ENTITY CATEGORY TYPE(s)]相关联。“每个**实体类别类型方案**[ENTITY CATEGORY TYPE SCHEME]有且只能由一个数据提供者[DATA PROVIDER]（内部的或外部的）来指定。”例如，企业级的总分类账类别类型的方案可以由公司的财务会计部来提供，这是个内部的方案。再例如，外部的方案，是一个评级类别类型的结合，该方案是标准普尔公司提供给金融工具和公司的。财务会计部和标准普尔公司是**当事人**[PARTY(s)]，该当事人扮演了**数据提供者**[DATA PROVIDER]这个**当事人角色**[PARTY ROLE]。12对于企业而言，了解类别方案的来源十分重要，特别是当其企业与其他企业共享数据时。这些方案可以提供通用的分类，在该分类法中企业可以共享和比较数据。

**注：两个不同的实体类别类型[ENTITY CATEGORY TYPE(s)]可以具有相同的名称，且看起来像同一实体类别类型[ENTITY CATEGORY TYPE]，只是重复了一遍。但是也并非总是如此，假如实体类别类型[ENTITY CATEGORY TYPE]是由不同的数据提供者[DATA PROVIDER(s)]所提供的，且是位于不同的方案当中。例如，路透社和彭博资讯都可以提供固定收益证券方案，其类别的名称为“政府债券类型[Government Bond Types]”。那么他们是同一个证券类别类型[SECURITY CATEGORY TYPE]吗？一般情况而言，他们不是，即使他们的方案会具有非常相似的语义含义，即使他们指的是同一个政府债券集合。即使这些方案可以相互替换，但是他们不是完全一样的。例如，你不应当创建与两个数据提供者[彭博资讯和路透社]相关的“政府债券分类[Government Bond Classification]”方案 。**

（图）

**图5-12** 使用隶属关系和分类方案的三级分类模式

尽管图5-12中的基本结构与前一节所介绍的三级分类模式相同，但是此模式何时有所变化的，在此模式中，**实体类别隶属关系**[ENTITY CATEGORY ROLLUP]和**实体类别类型隶属关系**[ENTITY CATEGORY TYPE ROLLUP]替代了**实体类别**[ENTITY CATEGORY]和**实体类别类型**[ENTITY CATEGORY TYPE]的一对多递归关系。这些实体支持以各种方式来聚集实体类别和实体类别类型。例如， 你想将**产品类别**[PRODUCT CATEGORY]“电脑零配件[Computer parts]”分为**产品类别**[PRODUCT CATEGORY(s)]：“硬盘[Hard disks]”、“鼠标[Mouse]”和“键盘[Keyboard]”，并且对**产品类别**[PRODUCT CATEGORY]“硬件[Hardware]”也进行这样的分类。换句话说“每个**实体类别**[ENTITY CATEGORY]可以是一个或多个**实体类别隶属关系**[ENTITY CATEGORY ROLLUP(s)]的父类，也可以是一个或多个**实体类别隶属关系**[ENTITY CATEGORY ROLLUP(s)]的子类”，对于实体类别类型[ENTITY CATEGORY TYPE]和实体类别类型隶属关系[ENTITY CATEGORY TYPF ROLLUP]之间的关系也是同样如此。这种形式的递归还可以通过**实体类别隶属关系**[ENTITY CATEGORY ROLLUP]和**实体类别类型隶属关系**[ENTITY CATEGORY TYPE ROLLUP]中的起始日期和终止日期属性追踪这期间维护（或更改）这些分类的方式。

在图5-13中我们将进一步解释此模式中的概念。续用前一节所介绍的案例，Euro-Electronics企业已表示它希望有一个单一的产品参考数据的主存储库。如果那些分类类型确实是源自于某一特定当事人，企业则希望将提供分类类型的当事人作为该存储库的一部分，自然当事人也是**产品类别类型**[PRODUCT CATEGORY TYPE(s)]的起源。

数据专业人士创建了图5-13，并将其作为此计划的一部分。**产品**[PRODUCT]、**产品类别分类**[PRODUCT CATEGORY CLASSIFICATION]、**产品类别**[PRODUCT CATEGORY]和**产品类别类型**[PRODUCT CATEGORY TYPE]的功能于前一节所介绍的功能相同。为了维护指定**产品类别类型**[PRODUCT CATEGORY TYPE(s)]的当事人，数据专业人士添加了**产品类别类型方案**[PRODUCT CATEGORY TYPE SCHEME]，并将其绑定到**数据提供者**[DATA PROVIDER]这一**当事人角色**[PARTY ROLE]上。

此外，**产品类别隶属关系**[PRODUCT CATEGORY ROLLUP]和**产品类别类型**[PRODUCT CATEGORY TYPE ROLLUP]允许多对多递归关系，而**产品类别隶属关系类型**[PRODUCT CATEGORY ROLLUP TYPE]描述的是隶属关系是什么类型（例如，“销售报表隶属关系[Sales reporting rollup]”与“服务报表隶属关系[Service reporting rollup]”）。因此，**产品类别隶属关系**[PRODUCT CATEGORY ROLLUP(s)]可能会有不同的变体，并且这些变体属于不同的**产品类别隶属关系类型**[PRODUCT CATEGORY ROLLUP TYPE(s)]。同样地，**产品类别类型隶属关系**[PRODUCT CATEGORY TYPE ROLLUP(s)]显示了‘分类[labels]’隶属的方式（例如，“产品子类别[Product Subcategory]”隶属于“产品类型[Product type]”），而**产品类别类型隶属关系类型**[PRODUCT CATEGORY TYPE ROLLUP TYPE]维护的是分类类型相互隶属的方式，例如，“销售报表隶属关系[Sales reporting rollup]”和“服务报表隶属关系[Service reporting rollup]”都可以隶属于**产品类别类型隶属关系类型**[PRODUCT CATEGORY TYPE ROLLUP TYPE]“报表隶属关系[Reporting rollup]”。在此递归架构中，不同地区的企业可能也有不同的‘分类[label]’层级。

（图）

**图5-13** 使用隶属关系和分类方案的三级分类模式的应用示例

你会发现在表5-7中有**产品类别类型**[PRODUCT CATEGORY TYPE(]是源自于两个外部机构，分别源自“世界海关组织[The World Customs Organization]”和“美国管理和预算办公室[U.S. Office of Management and Budget]”。这两个机构都是**当事人**[PARTY]（未在模型图或表中显示）的实例，他们扮演了数据提供者这一角色，其当事人角色标识分别是“34”和“36”。这两个**数据提供者**[DATA PROVIDER(s)]提供了两个非常相似的方案，“协调制度（HS或HTS）[Harmonized System]”和“B计划[Schedule B]”这两种方案都被用作进出口产品的分类。在这张表中有一处很有意思，产品类别类型具有两个看似十分相似的实例，也就是，“计算机磁盘[Computer Disks]”和“磁盘、具有磁性，可用于记录的[Disks, magnetic, recorded]（声音、视频或电脑）”。“B计划”指的是为出口产品所制作的方案，“协调制度（HS或HTS）” 和其他方案均是针对进口产品的。使用不同的分类方案是经常发生的情况。方案通常通过使用不同的语义对同类型的事物以不同的方式进行分类。这是你必须要注意的地方；两种方案中的两种不同分类可能会引用同一类型的数据，但是由于他们是不同的方案，所以由不同数据提供者所提供的两种产品类别类型是不同的、独特的。13尽管从技术层面上来说，两种方案采用相同的产品类别或产品类别类型是可行的，但是我们会发现这样做会使人感到困惑，因而这样做是危险的，因为该类别或类别类型可能会有不同的含义，并且在他们中很可能包含不同的元素。

**表5-7** 三级分类方案应用示例

**（表）**

Euro-Electronics企业的数据专业人士也添加了**产品类别隶属关系**[PRODUCT CATEGORY ROLLUP]和**产品类别类型隶属关系**[PRODUCT CATEGORY TYPE ROLLUP]，分别用以捕获存在于**产品类别**[PRODUCT CATEGORY(s)]与**产品类别类型**[PRODUCT CATEGORY TYPE(s)]之间的多对多关系。例如，图5-14中的简单案例，在该案例中，“手提电脑[Laptop Computers]”既可以是“硬件[Hardware]”的子类别也可以是“办公器械[Office Machines]”的子类别。在不同的企业中这种现象是非常常见的。各种类型和类别可以是不同分类的成员。例如，在Euro-Electronics企业中，销售部认为“手提电脑[Laptop Computers]”这个类别隶属于“硬件[Hardware]”，而运维支持部则将“手提电脑[Laptop Computers]”隶属于“办公器械[Office Machines]”。不同的部门可以使用**产品类别隶属关系类型**[PRODUCT CATEGORY ROLLUP TYPE]来定义产品类别之间的隶属方式。例如，在**产品类别隶属关系类型**[PRODUCT CATEGORY ROLLUP TYPE]中，可能会有实例：“销售部门隶属关系[Sales department rollup]”和“支持部隶属关系[Support department rollup]”，用这些实例维护该类型的隶属关系。

表5-8体现的是图5-4中的含有递归关系的三级分类模式。“手提电脑[Laptop Computers]”是“硬件[Hardware]”和“办公器械[Office Machines]”的子类别。因此，每个子类别可以属于多个父类别，而每个父类别可以有很多的子类别，**产品类别**[PRODUCT CATEGORY]和**产品类别隶属关系**[PRODUCT CATEGORY ROLLUP]之间的多对多关系能够满足以上要求。你还可以从表中看到起始日期和终止日期这两个属性，这些产品类别隶属关系的有效日期是从“2009年1月1日”至今，因为他们都没有终止日期。

**（图）**

**图 5-14** 手提电脑的多重隶属关系

**表 5-8**含有隶属关系的三级分类模式的使用示例

（表）

**注：产品类别类型隶属关系[PRODUCT CATEGORY TYPE ROLLUP]的功能与产品类别隶属关系[PRODUCT CATEGORY ROLLUP]的功能相似。注意，我们使用的是源自第4章节中的二级拓展递归模式，这种模式更加灵活，以至于类别和类别类型可以以不同的方式归属彼此。对于那些热衷于更加概括的模式的人，你可以选择使用三级递归模式来替换二级拓展递归模式，使用关联替代隶属关系，这样可以使类别可以以各种可行的方式与其他类别进行关联（例如，可以使用替代方式[substitutions]、分解方式[breakdowns]、不兼容方式[incompatibilities]等等）。更多关于分层、聚合或对等关系的内容，请参阅第4章节。**

**注：此模式另一个优化处理方式是添加实体类别隶属关系规则[ENTITY CATEGORY ROLLUP RULE]和实体类别类型隶属关系规则[ENTITY CATEGORY TYPE ROLLUP RULE]，分别用以管理实体类别隶属关系[ENTITY CATEGORY ROLLUP(s)]与实体类型隶属关系[ENTITY CATEGORY TYPE ROLLUP(s)]之间的关联行为。例如，可以将“商用[Commercial Use]”这一产品族从“政府[Government]”类别中排除。更多内容请参阅第四章节中的含有规则的三级递归模式。**

### 应当在什么时候使用这种模式？

我们在以下情况使用这种数据模型：

* **当需要更加全面灵活的分类模型时。**当有多种方法对实体进行分类时，当需要一个通用的方法为类别建模时，当需要很灵活的模型时，当需要联合类别用于强有力的分析时，应当使用这种模式。以上也是使用前一种模式即三级分类模式的原因。
* **当需要捕获实体类别方案的数据提供者时，该方案指定了实体类别类型以及相互关联的方式。**当需要捕获提供类别的内部或外部当事人时，必须要使用这种模式的“方案”，换而言之，就是分类方案。
* **需要捕获分类或分类类型可能存在的多对多分层、聚合或对等关系。**这种模式允许分类具有不同的隶属方式，例如，不同的部门可以有不同的隶属结构。
* **企业需要非常灵活的模式，该模式需要经得起分类的变化。**这种模式十分的灵活，可以添加任意数量的类别和/或类别类型，类别和/或类别类型可以有任意数量的隶属层次，而且数据提供者可以提供任意数量的方案，而所有的这些操作均不需要数据模型。

### 此模式的缺点是什么？

这种模式的缺点如下：

* **这种模式具有与三级分类模式相同的缺点。**例如，不能将其用作声明范围的一部分，也不能收集和验证非技术受众的数据需求。这种模式捕获特定的类别类型或特定的类型的特定业务规则、数据或关系。它也可以让数据建模人员在没有做详细分析的情况下，捕获所有的类别。
* **这种模式非产复杂，难以理解。**这是中复杂的模式，引进了多对多关系和方案。即使对于经验丰富的专业人士而言，理解并同企业或其他IT专业人士交流这种模式也会很困难。
* **如果只需要一对多递归关系或不需要方案，那么使用这种模式未免会大材小用了。**很多时候，分类和分类类型只需要一对多关系，所以多对多关系是没有必要的。此外，可能会不需要捕获方案和相关的数据提供者。

### 简介

在此模式中，我们介绍了**实体类别类型方案**[ENTITY CATEGORY TYPE SCHEME]这一概念。在**实体类别类型方案**[ENTITY CATEGORY TYPE SCHEME]中指定了**实体类别类型**[ENTITY CATEGORY TYPE(s)]，这些类别类型是由外部或内部组织机构提供的，我们将这些组织机构称之为**数据提供者**[DATA PROVIDER]。此模式可以捕获实体类别类型的来源，以及由数据提供者规定的实体类别类型相互关联的方式。当企业使用通用的分类相互交流时，知道此分类的来源是十分重要的。这种模式也支持创建聚合或分层关系的需求，这些关系以多对多的方式存在于类别和类别类型之间。

这是种难以理解的复杂模式。数据专业人士为此模式提供详细的解释、实践范例以及实例模型图是很重要的。此外，这种模式不适合用于作为非技术受众的范围声明。在方案中需要有很多的类别和类别类型，而在方各类别中又需要灵活的分层或聚合架构。当面对这种全面的需求时，可以使用这种模式。这种模式可以用来作为非常灵活的数据库设计的基础，这种灵活的设计可以承受随时间推移而发生的多次变更。

## 模式总结

**表5-9** 是本章节中所出现的所有模式的总结

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模式 | 描述 | 何时使用 | 何时不使用 | 优点 | 缺点 |
| 一级分类模式 | 为分类建模的特定方式，一般用于开发和验证非技术受众的数据需求。所有的分类信息被当作属性包含在实体内。 | 当你想要一个简单的模型，可以用于收集信息，捕获实体分类方法，将这些分类方式作为该实体的属性。  促进对不同分类的分析，显示分类的名称，揭露分类存在的问题，和以一种直观的方式启动建模。 | 不要使用这种模式作为一个长期的数据库实现的基础，因为有严重的数据冗余和数据的完整性问题可能会出现。  当需要一种灵活的分类方法时。任何一个新的分类都需要新的属性。  但需要维护分类的过多信息时，比如说描述、简称等等。  当一个实体的分类有多重表现时（例如，一个当事人可以分类成多个行业）  当需要在数据模型中维护复杂的业务规则时。 | 是一种简单的方式，可向非技术受众展现如何对实体进行分类。 | 不是规范化的数据模型，因此所以存在数据冗余问题，而此问题又引起了数据整合问题，如果使用这种模式作为实现基础的话。  十分不灵活，因为此模式不容易适应新分类，或当实体需要更多同类型的其他类别实例时（例如，当事人可以归类于多个行业）。此种模式还重复了类别属性，而且由于随着时间的推移，可能会需要新类别的属性，所以这种模式很不稳定。  不能够维护关于分类的其他数据。  如果实体具有很多的同一类型的可行性分类（例如，产品可以被分类为多个产品系列）。  不支持复杂的业务规则。 |

（续）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模式 | 描述 | 何时使用 | 何时不使用 | 优点 | 缺点 |
| 二级分类模式 | 这是一种规范化，但很特别的方法，用来捕获、管理分类数据，将这些数据作为一个独立的实体，然后将这些实体与相关的实体相关联。将分类数据从需要分类的数据中独立了出来（但是它们是相关联的）。 | 作为理解数据需求的一种方法或是作为作用域声明的一部分，是易于理解的。  当需要维护管理每个分类类型的数据时  作为实现特定数据库设计的基础。当你 有一组静态的分类类型，且这些分类类型易于理解，将来他们的性质不会发生变化（例如，从一对多关系变成对对多关系，从可选性变成强制性），或是在未来也不需要新的分类类型。  当每个分类都具有自己独特的属性和/或关系时  当需要为具有分层或聚合架构的分类类型建立模型时 | 需要更简单的模型，将其作为与非技术受众沟通交流的一种方法时。  当需要更灵活的方法时。在这种模式中，如果需要新的分类类型或分类类型与相关实体之间的关系发生变化时（例如，基数或可选性发生变化），那么就需要更改数据模型。  当不同分类是紧密相关的时候，且重要的是以一种通用的标准的方法捕获分类类型之间关系。  当你需要一种单一标准的方法来处理类型/类别时。  当一个实体具有很多的分类方法时。 | 这种模式代表的是规范化的数据模型，所以如果使用恰当的话，是不会导致冗余数据的。  这种模式可以用作数据需求的强大的语义声明。  分类数据独立于需要分类的数据之外，但是与其是相关的。  它是强大的语义声明，因为相对而言这种模式易于理解，它可以描述关于分类实体特性的大量信息。  此模式捕了特定的业务规则比如说每个分类的关系是可选的还是强制的，是一对多关系还是多对多关系。  你可以为每个独立的分类实体捕获特有的属性/关系。 | 与一级模式相比，这种即时不是很简单，不太容易理解。  这种模式不是非常的灵活。一旦需要新的分类类型，就会需要创建新实体，一旦分类到实体的关系放生变化，就需要更改数据模型。  如果不同分类类型之间存在多种关系，那么相比使用单一标准的方法将分类类型相互关联，这种模式会变得更加复杂，更加能以管理。  它会限制分析能力，因为它没提供一种方法可以将某事物通过一个单一的实体（例如，可以解决下面这个问题：哪种类型的客户对哪种类型的产品感兴趣）关联到所有的分类。  没有维护类型的标准方法（例如，一个单一实体包含了相关实体的所有可行的类型）。 |
| 模式 | 描述 | 何时使用 | 何时不使用 | 优点 | 缺点 |
| 三级分类模式 | 是一种灵活的方法，可用于捕获管理分类数据，在此模式中，将类型的分类添加成实例是很容易的。 | 在需要经常添加、更改或更新分类类型的环境中。  作为灵活实现的基础。  当一个实体需要多种分类时。  当需要非常通用的方法为各种类别建立模型时。  当需要更强有力的分析时，在此模式中，可以将各种类型汇合进一个单一的‘类别’实体来满足这一需求。  当分类类型数据对于其分类具有分层或聚合架构时。 | 作为明确的声明范围，也可以作为收集和验证数据需求的工具。  当实体具有很少的分类方法时并且这些方法是稳定不会发生改变的。  作为捕获所有分类的一种方法。使用这模式是非常容易的并且不用去挖掘所有的分类类型。  当每个类型的分类有非常特定的属性和关系时。  当每个分类、分类类型或隶属关系具有特定的规则时，比如说某一特定分类类型的数据具有分层或聚合架构而其他的数据却没有。  当实体只有少数的分类方法时。 | 这是中非常灵活的模式，当发掘新的分类时也不必更改该模式，  此模式提供了一种标准的分类方法，可以对具有很多分类的实体进行分类。  此方法可以统一一致地管理某一实体的所有分类类型。  提供了强有力的分析性能，比如说通过交叉引用可以解决下面这个问题：哪种类型的当事人对哪种类型的产品感兴趣。  为类别和类别类型提供了灵活的分层或聚合关系。维护了类别类型相互之间的隶属关系，可以使用这些数据确认类别架构。 | 这是种概括的模式，可能会很难理解，交流起来也不容易。  不捕获分类的特定的业务规则，比如说分类到另一实体的关系的可选性和基数。  对于不同的分类，这种模式不会捕获特定的属性和关系。  只支持有限数量和类型的分层和聚合，因为类别和类别类型值存在一对多关系。而且还不维护分类方案。  不可以特别的执行业务规则：更具类别类型结构哪些类别是可以隶属的（尽管通过维护递归关系的有效类型，这种模式是有助于解决这个问题）。 |

（续）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模式 | 描述 | 何时使用 | 何时不使用 | 优点 | 缺点 |
| 维护隶属关系和方案的三级分类模式 | 这是种灵活的方案，可以用来捕获管理分类数据，此方法可以维护多对多分类隶属关系的数据以及分类方案的数据 | 与三级分类模式相同  当企业需要捕获分类数据和不同类型的分类方案的来源时  当分类和分类类型可以使用多种方式进行“隶属”时 | 与三级分类模式相同。  不需要维护方案信息时。  不需要多对多的隶属关系时 | 与三级分类模式类似。  这种分类模式很全面，可以让公司捕获数据分类的来源以及分类方案的信息。  可使用多对多对关系来容纳非常灵活的类型以及类别类型的隶属关系。  可以将多种分类作为不同的分类法的一部分。 | 类似于三级分类模式。  此模式很复杂，难以理解。  可能不需要方案和多对多递归关系，因此采用这种模式未免大材小用了。 |

**参考文献**

1 Carl Linnaeus 是家喻户晓的分类学之父. See

http: //www.ucmp.berkeley.edu/history/linnaeus.html.

2 更多关于具体和概括建模模式请参考第一章节

3 摘自 Dictionary.com at http://dictionary.reference.com/browse/type.

4 Taken from Dictionary.com at

http://dictionary.reference.com/browse/category.

5 定义源自英国伦敦的自然历史博物馆网站 http://www.nhm.ac.uk/nature-online/science-of-natural- history/taxonomy- systematics/what-is -taxonomy /what-is-taxonomy.html.

6 参看 Michael Barnwell的文章："What Is Taxonomy: Organizing Content for Better Site Performance (June 2005). Available at the Avenue A/Razorfish web site at

http: //www.avenuea-razorfish.com/articles /Taxonomylnsigh Barnwell.pdf.

7 这一特征可以称作为态射。态射描述的是域与值域之间的一种映射，非常类似于将类别映射到另一个类别。参看网址

http: //planetmath.org/?op=getobj & from=objects & id=8114.

8 关于工作计划的更多内容，请查阅数据模型资源手册卷1修订版，由L. Silverston (Wiley, 2001)撰写的，可应用于所有企业的通用数据模型库。

9 关于递归关系的更多内容请参看本书的第四章节。

10 See Competing on Analytics: The New Science of Winning by Thomas H. Davenport and Jeanne G. Harris (Harvard Business School Press, 2007).

11 摘自 Dictionary.com at http://dictionary.reference.com /browse/scheme.

12 关于上下文的角色的更多信息请参看本书的第3章节

13 我们所引用的两种方案源自http://www.census.gov/foreign-trade/.

【注释1】current indicator “当前称呼”