

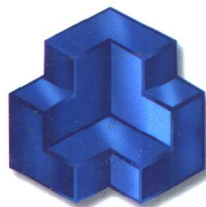
Designed for

Microsoft®
Windows NT®
Windows 98附 赠
CD-ROM

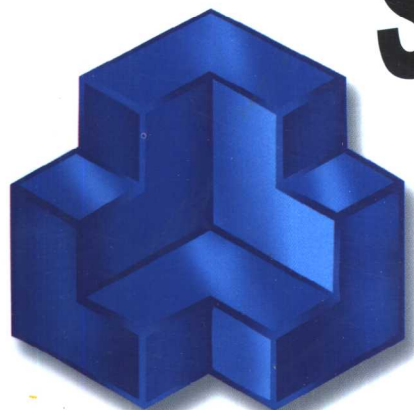
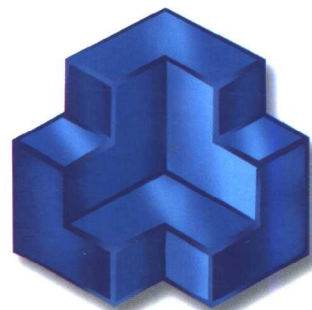
Microsoft SQL Server
2000 Analysis Services
Step By Step

微 软 公 司
核 心 技 术 书 库

(美) Reed Jacobson 著 江帆文化艺术中心 译



SQL Server 2000 Analysis Services



学习指南



机械工业出版社
China Machine Press

Microsoft Press

微软公司核心技术书库

SQL Server 2000 Analysis Services 学习指南

(美) Reed Jacobson 著

江帆文化艺术中心 译



机械工业出版社
China Machine Press

本书对Microsoft SQL Server 2000 的Analysis Services 组件进行了详细的介绍。本书使用简单而具体的示例对多维的术语及概念进行了解释,并阐述了多维分析的概念框架;描述了创建一个功能性的Analysis Services 数据库所需的工具;同时介绍了数据库的优化以及数据库访问的安全机制。

无论你是负责创建和管理Analysis Services数据库的管理员还是将电子表格转换成共享可升级表格的分析员,都将从本书中受益。

Reed Jacobson: Microsoft SQL Server 2000 Analysis Services Step By Step.

Copyright © 2001 by Microsoft Corporation.

Original English language edition copyright © 2000 by Reed Jacobson; Published by arrangement with the original publisher, Microsoft Press, a division of Microsoft Corporation, Redmond, Washington, U.S.A. All rights reserved.

本书中文简体字版由美国微软出版社授权机械工业出版社出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,侵权必究。

本书版权登记号: 图字: 01-2000-4313

图书在版编目 (CIP) 数据

SQL Server 2000 Analysis Services 学习指南 / (美) 杰克博森 (Jacobson, R.) 著; 江帆文化艺术中心译. - 北京: 机械工业出版社, 2001.7

(微软公司核心技术书库)

书名原文: Microsoft SQL Server 2000 Analysis Services Step By Step

ISBN 7-111-09006-3

I. S… II. ① 杰… ② 江… III. 关系数据库 - 数据库管理系统, SQL Server 2000 IV.TP311.138

中国版本图书馆CIP数据核字 (2001) 第038007号

机械工业出版社 (北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 杨海玲

北京昌平奔腾印刷厂印刷 · 新华书店北京发行所发行

2001年7月第1版第1次印刷

787mm × 1092mm 1/16 · 22.5印张

印数: 0 001-5 000册

定价: 49.00元 (附光盘)

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

前言

本书的读者对象

Analysis Services（分析服务）是Microsoft SQL Server 2000中的一个功能强大、但又容易被忽视的组件。使用Analysis Services可以创建快速、灵活、交互的、基于存在于大型联合数据库中的信息的报告，可以给大的、共享的数据源增加电子表格的分析功能，可以创建数字化数据的多维视图。

对于从多维数据库中检索信息的人来说，通常觉得这种交互式的操作很简单、有逻辑性而且直观。然而对于从事创建和设计多维数据库的人来说，开始常常会觉得这个概念陌生、复杂、而且有挑战性。阐述Analysis Services（或者OLAP Service，即所谓的SQL Server的先前版本）的书籍一般假定读者已经对多维的概念很熟悉了。然而对于很多人来说，阅读关于Analysis Services的书籍就好像不会游泳的人要跳入游泳池的深处一样。

这本书将给出一个关于通常的多维概念，尤其是关于SQL Server 2000的Analysis Services组件的准确且循序渐进的介绍。本书使用简单而具体的例子来解释多维的术语和概念，去掉那些容易混淆的、内在的、复杂的东西。本书还引导你掌握使用Analysis Services工具的全过程，从最简单的可能解决方案逐步过渡到处理非正常情况和优化数据库的复杂技术。

本书为以下读者对象提供有用的参考：

- 如果你是数据库的管理员，而且要负责创建和管理Analysis Services数据库，本书可为你提供所需要的基础知识。
- 如果你要决定Analysis Services在组织中的作用，本书可以为你提供所需要的观点。
- 如果你是将电子表格转换成可共享、可升级表格的分析员，本书能够为你提供所需要的技术。

本书内容的涵盖范围

本书由三部分组成。第一部分，“管理基础”，阐述多维分析的概念框架，然后描述创建一个功能化的Analysis Services数据库所需要的所有工具；第二部分，“多维表达式”，介绍给数据库增加强有力的分析能力的语言；第三部分，“高级管理”，处理在实际工作环境中遇到的问题，例如，数据库变得很大而需要优化，或者数据库的访问途径需要安全性。

Analysis Services是一个功能强大而应用广泛的产品。本书的目标是提供纯粹的介绍，而不是完整的参考。以下所列出的几个高级主题不在本书的范围中：

- Data mining（数据采集） 在大型的数据库中搜寻关系和模式的能力。这个主题范围很广，需要用一整本书的篇幅来讲述。
- Action（操作） 管理员预先分配动作的能力。例如一个用户在浏览报告时所触发的打

开用户网页的动作。动作主要是开发人员使用的工具，而本书集中在数据库管理员执行的任务上。

- **Extreme scalability (极强的扩展性)** 将一个数据库扩展到多个服务器上或者将数据从一个服务器连向另一个服务器的能力。这些是特殊的任务，将由使用Analysis Services相对较少的公司完成。

阅读本书的前提

学习本书需要安装SQL Server 2000的Analysis Services组件。第9章“处理优化”中9.4节“自动处理数据库”部分需要完全安装SQL Server 2000。对于本书中的大多数情况，只需要从SQL Server 2000标准版(Standard Edition)安装Analysis Services。只有9.3节“使用分区”需要从SQL Server 2000企业版(Enterprise Edition)安装Analysis Services。

其他系统需求如下：

- Intel (或兼容) Pentium、Pentium Pro或Pentium II 166 MHz以上的处理器。
- 64MB以上RAM，对于企业版，推荐使用128MB。
- 50MB以上的可用磁盘空间，还需要130MB供Analysis Services的典型安装需要。
- 800 × 600以上的显示器分辨率。
- 微软或兼容的鼠标，CD-ROM驱动器（或者网络环境）。
- Microsoft Internet Explorer 5.0 或以上的版本。

对于操作系统，至少要有Microsoft Windows NT 4.0工作站版本的Service Pack 5.0或以上版本，或者Microsoft Windows 2000的专业版(Professional Edition)。要实现9.3节“使用分区”的操作，需要安装SQL Server 2000的企业版，这时操作系统需要Windows NT4.0的服务器版(Server Edition)或者Windows 2000的高级服务器版(Advanced Server Edition)。

要实现第5章“Office 2000分析组件”的任务，需要Microsoft Excel 2000。对于第5章中的其他一些任务，需要Microsoft FrontPage 2000，它可以单独提供或者附带在Microsoft Office 2000 Premium Edition中。

另外，还需要安装本书随书附带光盘中的示例文件。关于如何使用随书附带光盘，请看下一部分。

如何使用随书附带光盘

光盘中的Readme.txt文件包含如何安装示例文件的指导。每一章的示例文件都是独立的，所以没有必要先完成以前的章节再来实现后面章节的任务。如果在安装时选择将示例文件不放在缺省的文件夹中，每次恢复OLAP数据库的时候，请依照以下步骤执行：

1) 选择Start→Programs→Microsoft SQL Server→Analysis Services→Analysis Manager选项来启动Analysis Manager (分析管理器)。

2) 在控制树中，展开Analysis Servers文件夹。右键单击服务器，选择Restore Database (恢复数据库)。

3) 浏览到你安装随书光盘示例文件的文件夹，选择一个章节的.cab文件，单击Open按钮。

4) 单击Restore (恢复), 然后在处理完成时关闭Restore Database Progress (恢复数据库进度) 对话框。

5) 在控制台树中, 展开已恢复的OLAP数据库文件夹和Data Sources (数据源) 文件夹。

6) 在控制台树中右键单击数据源, 单击Edit按钮, 在Connection选项卡中输入数据源的正确路径, 然后单击OK按钮。

要了解更多的使用随书光盘的信息, 请参考光盘中的Readme.txt文件。

目 录

前言

第一部分 管理基础

第1章 数据分析基础	1
1.1 理解数据仓库	1
1.1.1 数据仓库的用途	2
1.1.2 数据分析中的维	3
1.1.3 数据分析中的层次结构	6
1.2 数据仓库的结构	8
1.2.1 实际表格	9
1.2.2 维表格	9
1.2.3 可供选择的维表格的结构	11
1.3 理解Analysis Services	12
1.3.1 立方体的概念	12
1.3.2 Analysis Services的管理工具	14
1.3.3 Analysis Services用户工具	14
1.4 小结	17
第2章 分析管理器	18
2.1 预览Analysis Manager	18
2.2 准备创建立方体	24
2.3 使用Cube Wizard来设计立方体	27
2.4 处理和浏览立方体	39
2.5 小结	45
第3章 维和立方体编辑器	46
3.1 使用一个标准的星形模式维	47
3.2 使用标准的雪花形维	52
3.3 使用Time维	63
3.4 使用立方体	70
3.5 使用立方体中的量度	75
3.6 使用父子维	84
3.7 小结	91
第4章 高级维和立方体	93

4.1 创建财政立方体	93
4.2 创建跟踪互联网的立方体	99
4.3 创建预测立方体	108
4.4 创建虚拟立方体	118
4.5 小结	125
第5章 Office 2000分析组件	127
5.1 创建Excel PivotTable报表	128
5.2 创建Office PivotTable列表	145
5.3 创建本地立方体	162
5.4 小结	172

第二部分 多维表达式

第6章 MDX值	175
6.1 创建简单的MDX表达式	175
6.2 从单元格中检索值	191
6.3 计算相关基值	198
6.4 比较随时间变化的值	202
6.5 小结	208
第7章 MDX集	209
7.1 创建MDX查询前的准备	209
7.2 创建基本的MDX查询	218
7.3 使用聚合函数创建计算	224
7.4 小结	237

第三部分 高级管理

第8章 存储优化	239
8.1 设定优化存储选项	240
8.2 管理潜在聚合的存储池	250
8.3 小结	260
第9章 处理优化	261
9.1 理解OLAP处理过程	261
9.1.1 Analysis 服务器如何处理维	261

9.1.2 Analysis 服务器如何处理立方体	263	10.2 创建虚拟维	306
9.1.3 改变数据仓库中的数据	267	10.3 小结	319
9.2 管理OLAP进程	273	第11章 安全	320
9.3 使用分区	282	11.1 创建安全角色	320
9.4 自动处理数据库	291	11.2 在维中应用安全机制	334
9.5 小结	297	11.3 在立方体中应用单元格层次安全机制	345
第10章 维优化	299	11.4 小结	349
10.1 优化立方体中的维	299		

第一部分 管理基础

第1章 数据分析基础

本章目标

本章主要内容如下：

- 数据仓库的目的及结构。
- 数据库中多维性的含义。
- Analysis Services的词汇。
- Analysis Services存储信息的地方。
- Analysis Manager（分析管理器）和Analysis服务器是怎样通信的。
- 客户端的应用程序同Analysis服务器是如何通信的。

Microsoft SQL Server 2000的Analysis Services被称为多维数据库服务器。多维这个术语会引发人们联想到阿尔伯特·爱因斯坦的扭曲的时空、并行的宇宙以及使得解决积分问题变得相对简单的数学公式。而这里它的作用是支持Analysis Services，使其对数量巨大的数据的处理变得快捷、简单和容易。

在H.G.Wells的《时间机器》（The Time Machine）这个故事中，将来的城市居民的生活将会变得简单和美好，因为所有的繁重和肮脏的工作都由藏在地下的神秘的生命（它们有时也会捕食住在地面的人类）来完成。人们也许会担心与之相似，Analysis Services使得分析者处理数据更容易是因为它将复杂和令人困惑的管理任务交给了相对来说少见的数据库管理员。

实际的情况并没有H.G.Wells的故事中所说的那么戏剧性和可怕。实际上，将数据库称作多维并不十分确切。这是一个时髦的词汇，但请大家不要误会，当多维这个词应用到数据库上时，它和加速到接近光速的微粒的多维行为以及Alice探索兔子洞的多维外表毫不相干。本章将讲解数据库环境中多维性的真实含义以及Analysis Services如何简化数据分析。

1.1 理解数据仓库

假设你是一个小公司的总裁，公司需要发展壮大，但只有有限的资源来支持这种扩展。你需要做出决策，而在做出这些决策前需要特定的信息。所需要的大部分信息来自于公司以外。这就是为什么你要阅读《华尔街》杂志并且在浏览器中设置链接到www.bloomberg.com的书签的原因。而所需要的另一部分信息来自公司内部，许多信息是数字化的。数据仓库是存储和分析数字信息的工具。

Ralph Kimball也许是最著名的数据仓库理论的宣传者，他将数据仓库简单地描述为“可以

访问数据的地方”。他列举了6个数据仓库的关键要素：

- 数据仓库检索必须迅速。
- 数据仓库的数据必须有永久的一致性。
- 用户可以将数据仓库中的数据分片和分块，即抽取单独的项目（片）和跨表比较项目（块）。
- 数据仓库必须有简单易用的浏览工具。
- 数据仓库的数据必须完整和可靠。
- 高质量的仓库数据需要高质量的数据收集过程。

Analysis Services可以在公司的数据仓库战略中作为一个完整的部分，然而它不能满足数据仓库的所有需要。实际上，Analysis Services只促进了上述6个要素中的两个：Analysis Services可以使检索数据的过程变得快捷（要素1），同时也可以使分片和分块数据变得容易（要素3）。尽管像在第5章“Office 2000分析组件”中所描述的那样，可以使用Microsoft Excel 2000 PivotTable（数据透视表）报告工具作为Analysis Services数据仓库的简单的浏览器，但是Analysis Services不直接提供简单易用的浏览工具（要素4）。

Kimball描述的其他和数据仓库数据的一致性和可靠性相关的3个要素必须在对Analysis Services的使用之前得到满足。换句话说，Analysis Services假设你已经拥有了用关系数据库管理系统（如SQL Server）创建的合法的、功能性的数据仓库。

Analysis Services在已经存在的关系数据仓库中再创建一个新层。这个新层使对数据的访问变得非常快捷和灵活（Kimball的数据仓库要素中的两个）。

注意 即使Analysis Services和SQL Server 2000捆绑在一起，但它并不要求以SQL Server 2000作为关系数据库管理系统。可以将Analysis Services同Oracle、DB2、Informix或者其他的关系数据库管理工具（包括Microsoft Access）一起使用。

1.1.1 数据仓库的用途

数据仓库存储稳定的和确认的数据值。把数据仓库和事务数据库做一番比较会对你有所帮助：

- 事务数据库可以帮助人们执行活动，而数据仓库可以帮助人们做出决定。举例来说，事务数据库可以告诉旅行社某航班上哪些座位可用来帮助它做出新的机票预定；然而，数据仓库可以知道某航班的空座位的历史模式，以此来帮助航空公司的管理人员决定是否将来需要调整航班的时间表。
- 事务数据库是易变的，它的信息经常随着新的订单的加入或取消而变化。比如说新的产品产出或运到或者新的预约的建立。而数据仓库是稳定的，它的信息按标准间隔（每月、每周甚至每小时）更新。而且，在理想情况下，更新时只是加入新时间的值，而不必修改数据仓库中原来的数值。
- 事务数据库注重细节信息。旅行社在预定机票时不需要知道航班的平均空座位有多少，购买最新的Harry Potter图书的父母并不关心少儿读物生产线的存货数量。而数据仓库注重整

体信息。更改航班时间表的管理人员并不关心具体哪个座位空着，而是要知道整个的乘坐情况。这两者区别的隐含意思是数据仓库的关键字数值必须是能够总计的数字化的数值。

- 事务数据库通常向数据仓库提供需要存储的数据值。

1.1.2 数据分析中的维

在数据仓库中，用来监视业务的可总计的数值被称作量度。当寻找数字化的信息时，首要问题是想查看哪一种量度。比方说，用Shipment Units、Defects Per Hour或者Ad Campaign Responses方式查看SalesDollars。假设你在查看公司的Units Sold（单位销售）报表时，看到了113。这一个单独的数值并不能告诉你什么，你想要探索它背后更多的信息。举例来说，你的公司的销售按时间是怎么分布的。你要求每月的分析，得到以下新的报表：

January	February	March	April
14	41	33	25

公司运转了4个月，所以在报表的顶行有4个月的标签。现在不是一个值，而是有4个值，列出了每个月的原始值。新数值的数目等于月的数目。这如同计算现实世界中的线性距离一样，一条线段的长度就是它本身的长度。

假如你仍然对每月的报表不满意。对于公司销售的多种商品，每种商品在不同时间是怎么销售的呢，要求一份新的关于不同月份和不同商品的报表，如下所示：

	January	February	March	April
ColonyBlueberry Muffins			6	17
Colony Cranberry Muffins	6	16	6	8
Sphinx Bagels	8	25	21	

公司销售3种产品，所以报表的左边是3种产品的名字。每种产品的销售按月份分布，且4个月份的标签仍然排列在报表的顶部。你现在有12个数值可以参考。数值的总数量等于商品种类的数量乘以月数。这就好像计算现实世界中矩形的面积：面积等于长乘以宽。这个报表就像一个矩形。

Analysis Services中的术语OLAP

Analysis Services以前的版本称为OLAP Services。这个名字在SQL 2000中被更改了，是为了避免使用大家不熟悉的缩略词。同时也是由于Analysis Services现在包括数据采集工具，这已经超出了OLAP技术的范围。术语OLAP仍然经常出现在产品的文档中，所以需要了解它的含义。理解这个术语的历史能够帮助你理解它的含义。

在20世纪80年代，E.F.Codd提出了online transaction processing（OLAP，在线事务处理）这个术语并且定义了OLTP数据库的12条准则。他的术语和准则被广泛作为管理公司日常运作（事务）的数据库标准。在20世纪90年代，Codd又提出了online analytical processing（OLAP，在线分析处理）这个术语以及定义OLAP数据库的12条准则。这次他的准则没有得到广泛承认，但是术语OLAP被很多人用来很好的描述为推

动公司的决策制定（分析）而设计的数据库。

有些人简单地将OLAP用作数据仓库的同义词。然而，术语OLAP通常指的是使得数据仓库数据能容易访问的工具。因此，术语OLAP用在SQL Server 2000 Analysis Services中也是这个意思。

将报表比作矩形只是算术意义上的，而不是形式上的。报表可以有不同的结构，比如它可以像下表一样简单：

Colony Blueberry Muffins	January	
Colony Blueberry Muffins	February	
Colony Blueberry Muffins	March	6
Colony Blueberry Muffins	April	17
Colony Cranberry Muffins	January	6
Colony Cranberry Muffins	February	16
Colony Cranberry Muffins	March	6
Colony Cranberry Muffins	April	8
Sphinx Bagels	January	8
Sphinx Bagels	February	25
Sphinx Bagels	March	21
Sphinx Bagels	April	

无论你把数值列成上表的形式（此时数值形成一条直线）还是在网格中显示它们（此时数值形成一个矩形），如果共有4个月，每月有3个商品，那么都会有12个数值，因为产品和月份彼此独立。每种产品有它自己每月份的销售值，即使该值为零也如此。

回到矩形报表，假设你的公司在两个州销售，若想知道每种商品每月在不同州的销售情况，在表中添加另一个标签组表示州，将会得到如下所示的新表：

		January	February	March	April
WA	Colony Blueberry Muffins			3	10
	Colony Cranberry Muffins	3	16	6	
	Sphinx Bagels	4	16	6	
OR	Colony Blueberry Muffins			3	7
	Colony Cranberry Muffins	3			8
	Sphinx Bagels	4	9	15	

现在，报表有两个州标签，3个产品标签（每个产品出现两次），4个月份标签。即使有些值的栏是空白的，但它仍表示243个值。报表中数值的总数目等于州数目乘以产品数目，再乘以月份的数目。这就类似于计算现实世界中立方体的体积一样：体积等于立方体的长乘以宽再乘以高。报表并不真的看上去像一个立方体；它看上去更像个矩形。同样，你可以把它重新调整为一张表，表的开头可以和下表一样：

WA	Colony Blueberry Muffins	January	
WA	Colony Cranberry Muffins	January	3
WA	Sphinx Bagels	January	4
OR	Colony Blueberry Muffins	January	

(续)

OR	Colony Cranberry Muffins	January	3
OR	Sphinx Bagels	January	4
WA	Colony Blueberry Muffins	February	
WA	Colony Cranberry Muffins	February	16
WA	Sphinx Bagels	February	16

无论用什么方式显示报表,它都有3个独立的标签列表,报表中数值的总数目等于第一个独立的标签列表的唯一项目数(例如两个州)乘以第二个独立的标签列表的唯一项目数(3种产品),再乘以第三个独立的标签列表的唯一项目数(4个月)。由于独立的标签列表这个称谓太冗长,同时由于计算报表的数值总数的算法同计算长度、面积和体积的算法完全相同,都采用空间范围的计算方法,为了取代独立的标签列表这个名词,数据仓库的设计者借用了数学中的维(dimension)这个名词。请记住这是一个借用的术语。数据分析的维同物理的维是大不相同的。因此,你的报表有3个维——State、Product和Time。报表中的整个数值数目等于第一个维的项目数乘以第二个维的项目数,依此类推。使用维这个术语并不说明标签和数值在报表中是如何排列的,也不表明数据在数据库中是如何存储的。

每次创建新的维时,该维中的各个项目应该彼此相关。例如,它们都是产品名称,或者都是日期。从而,维中的各个项目被称为该维的成员(member)。

现在有了一个更复杂的报表。也许你既想知道产品的销售量(units),又想知道销售额(dollars),可以这样安排新表:

		January		February		March		April	
		U	\$	U	\$	U	\$	U	\$
WA	Colony Blueberry Muffins					3	7.44	10	24.80
	Colony Cranberry Muffins	3	7.95	16	42.40	6	15.90		
	Sphinx Bagels	4	7.32	16	29.28	6	10.98		
OR	Colony Blueberry Muffins					3	7.44	7	17.36
	Colony Cranberry Muffins	3	7.95					8	21.20
	Sphinx Bagels	4	7.32	9	16.47	15	27.45		

因为销售量和销售额独立于State、Product和Time维,可以认为它们组成了新的第四个维,可以称为Measure维。报表中的数值的总数仍然等于每一维的成员数量的乘积:2乘以3乘以4乘以2,等于48。但这时不存在(也不需要)模拟现实世界中的任何种类。请记住,维这个名词只是独立标签列表(independent list of labels)的简单称谓。有着4个(甚至20或者60个)独立列表同有3个独立列表的报表一样,只是使报表变大了。

在现实世界中,你要测量的对象随着它们的维的多少而发生改变。例如,一维的英寸是线性英寸;二维的英寸是平方英寸;三维的英寸是立方英寸,而立方英寸是和平方英寸或线性英寸完全不同的对象。然而在报表中,所研究的对象不随维的增加而改变,是不变的数值。四维报表中的数值同一维报表中的数值没有什么不同。在报表中,增加一个维相当于增加了一个新的、独立的细分量度的方法。

尽管给一个报表增加第四、第五维不是不可行的,但这并不表明增加一个新维的变化很小。

假设开始有一个二维的报表：30种产品和12个月份，即360个可能的数值。这时加入3种新产品将使报表的数值总数变为396，增加了10%。如果增加具有3个成员的第三维，报表的总数就变为1 080，增加了300%。考虑一个极端的情况，具有128个成员的一维报表有128个值；如果把这128个成员分为64个维数，每个维具有两个成员，报表将有18 446 774 073 709 551 616个数值，这是一个多么大的数啊！

1.1.3 数据分析中的层次结构

对于一个小的或新的机构来说，可以很容易地通过观察详细的数字来了解它的业务。举例来说，如果你有一个刚刚运作4个月、只有3种产品的公司，你可以很轻松地在最底层的细节上分析数据。心理分析指出，大多数人可以较容易地理解7个项目（或者7组项目）。分组（或者说合计）是人们处理众多项目的常用方法。一旦公司销售产品超过6个月，你会毫无疑问地开始关心每季度以及每月的销售额。与此类似，如果你的公司有超过12种产品，你也许会将这些产品分成产品系列或者产品组。但是，如何将季度或者产品系列组成合适的维呢？

通常，认为同一维中的各个成员具有“属于同一类”的属性。January和February自然可以被认为属于同一类，因而当然应处于同一维中。January和Colony Blueberry Muffins自然不属于同一类，因此就不应该处于同一维中。但是，月和季度怎样呢？它们属于同一维吗？

请记住，维实际上是报表中的一组独立标签列表。判定新成员属于新的维或属于已经存在的维时，假设新的成员作为报表的一个列的标题，已经存在的维的成员作为行的标题。如果新的成员是独立的，应该（至少有可能）使它们的每个交叉处都有数值。但是请看如下的以月份为行、以季度为列的报表：

	Qtr1	Qtr2
January	14	
February	41	
March	33	
April		25
May		29
June		39

表中一半的位置是空白的，这并不是偶然现象。January不可能出现在Qtr2中，同样，April也不可能出现在Qtr1中。这个报表看起来很愚蠢，将两个季度不放在Time维中而单独形成一个维，会使得表格的数值总数乘上二，而一半的位置保持空白。因此，表格数值数目的总数并没有发生变化。实际上，季度位于列中的报表并不能提供想要知道的每个季度的总值。

希望看到的报表更应该像如下所示的表：

	Qtr1	88
January	14	
February	41	
March	33	

Qtr2	93
April	25
May	29
June	39

月份和季度不是完全独立的成员，因此不应该处于独立的维中。季度的总量只是简单的月总量的集合，它们属于同一个维。然而，Qtr1和January还是有区别的，你可能希望January的标签比Qtr1的标签缩进更多一些。而Qtr1和Qtr2更加相似。

如果Month和Quarter这些字眼没有出现在报表中，你自然会引用January、February等作为月，也会引用Qtr1、Qtr2等作为季度的标签。月份和季度在Time这个维上，根据它们对时间的概括程度不同构成了不同层次，形成了一个层次结构。举例来说，在Time维上，January和February是月份层次上的成员，Qtr1、Qtr2是季度层次上的成员。随着时间的推移和增加更多的月份和季度，还会在这个维上增加Year层次。包含太多的成员的维往往会分化为层次结构，而定义的层次结构包含不同的层次。

当维中包含层次结构时，有时可能想看到整个层次结构，有时可能只想看到最上面的一两个层次，有时也许只想看到最底层。可以使用成员这个术语来描述维中的所有成员或维中特定层的成员。例如，Time维中的成员有年、季度和月份，而Month层次中的成员不包括任何的年和季度。细节的最底层的成员被称为leaf members（叶子成员）。维中不能没有叶子成员，但有可能只包括叶子成员，就是说它只有一个层次。例如，在Measures维中，将Units和Dollars总数求和是没有意义的。

有些层次结构（如Time）是平衡的。如果Qtr1下有月份，那么Qtr2、Qtr3和Qtr4下都有月份。在平衡的层次结构中，给层次命名是很容易的。例如，在一个典型的Time层次结构中，各个层次可以用（year）、（Quarter）和（Month）来命名，如图1-1所示。

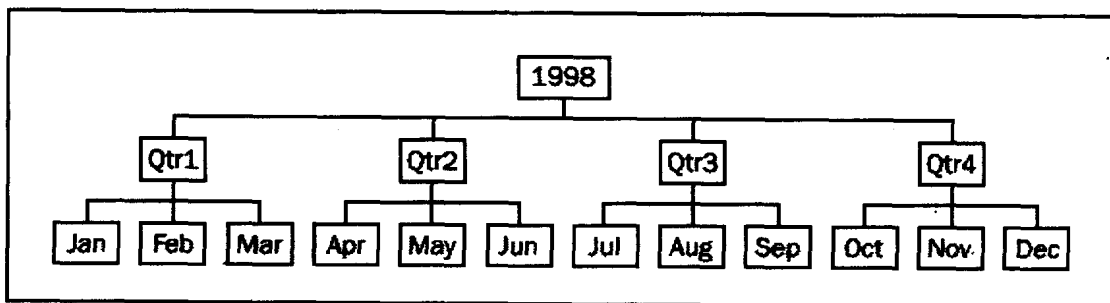


图 1-1

有些层次结构是不平衡的。一个组织的图表往往是不平衡的。举例来说，很多公司的生产部比人力资源部有着更多的人员，因此有着更多的管理的层次。在一个不平衡的层次结构中，给特定的层次命名是不方便的。但是，它们的叶子成员的下面没有子成员，如图1-2所示。

有些层次的结构介于平衡与不平衡之间。例如，在一个地理的层次结构中，可以将层次分为Country、Region和State，但对于某些State，可以跳过Region这个层次。这确实是一个平衡的

层次结构（因为可以容易地命名各个层次），但是有些成员的父亲成员缺少或者不可见。隐藏某些父辈成员的层次结构称为“ragged”（不规则的）层次结构，如图1-3所示。

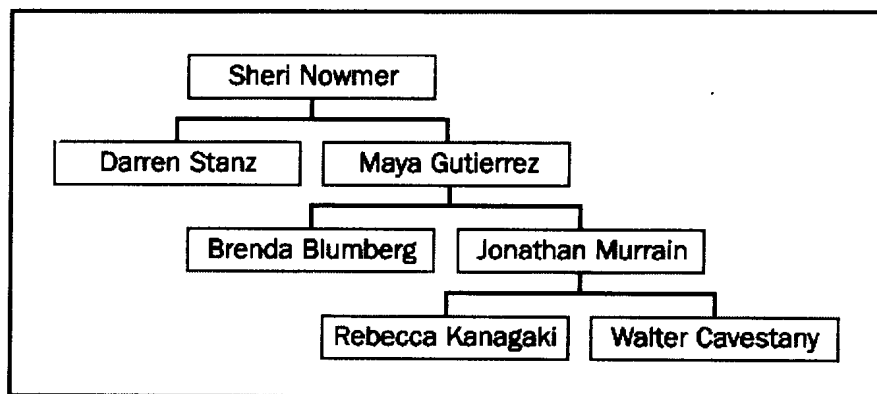


图 1-2

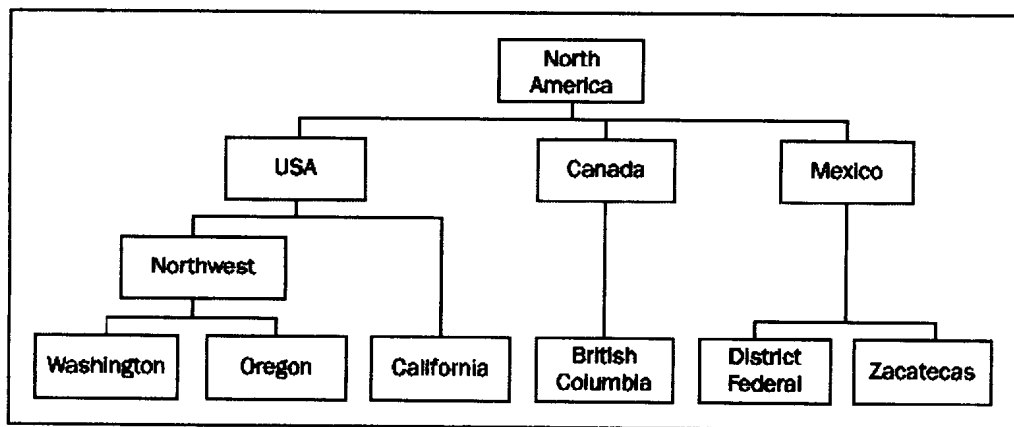


图 1-3

Analysis Services在定义平衡、或不平衡的层次结构时，不论该结构是否规则，都有着很大的灵活性。维总是有叶子成员的。层次结构定义叶子成员是怎样（和是否）被统计的。

1.2 数据仓库的结构

Analysis Services给客户应用程序在用多维创建报表时提供了便利，但是报表中显示的数据最终仍然来自于关系数据仓库。因为Analysis Services事先假定已经存在关系数据仓库，本书不涉及那些为关系数据仓库收集数据的令人头疼的问题，只是简单包含一些涉及到的示例性的关系数据仓库。无论是自己建立关系数据仓库还是请公共的数据库管理员创建数据仓库或者使用已经存在的数据仓库，在某些地方你需要给Analysis Services和关系数据仓库间建立链接。因此，要成功地使用Analysis Services，必须清楚Analysis Services对数据仓库的要求。Analysis Services要求数据仓库具有一个非常特殊的表单，即一个以实际表格为特征的表单。

1.2.1 实际表格

实际表格指的是存储量度的详细数值或实际数值的关系数据库中的表格。存储Dollars和Units随着State、Product和Month分布的实际表格有5栏，如以下的示例所示：

State	Product	Month	Units	Dollars
WA	Colony Cranberry Muffins	January	3	7.95
WA	Sphinx Bagels	January	4	7.32
OR	Colony Cranberry Muffins	January	3	7.95
OR	Sphinx Bagels	January	4	7.32
WA	Colony Cranberry Muffins	February	16	42.40

在数据库表格中，有时，一列称为一个字段，一行称为一条记录。

在上例的实际表格中，前3列（State、Product和Month）是关键字列；剩下的两列（Unit和Dollar）包含量度值。实际表格中的每一列要么是关键字，要么是量度值。

对于能够被Analysis Services使用的实际表格，必须使每个量度包含一列。销售数据库可以包含两个量度列（一个表示Dollar，另一个表示Unit）。产品存货数据库可以包含3个量度列（一个是Unit，一个是Minute，另一个是Defect）。在报表中，可以认为量度形成了独立的维。就是说，可以将数量和金额并列起来作为列的标题，也可以将它们作为行的标题。但是在实际表格中，每个量度都以独立的列的形式出现。

为了便于Analysis Services的使用，实际表格必须包含能够展现所需的量度的最底层的细节的行。换句话说，实际表格中的行都是对应每一维中叶子成员的行。Analysis Services不能使用包含组合量（比如季度和年度总量）的行的实际表格。举例来说，在包含State的维中，有3个层次结构（State、Region和Country）。只有State那一层的成员才能出现在实际表格中。Analysis Services会建立所有的总计的数值。在实际表格中，每个维中的叶子成员都表明了一个单独的行。

以上表格中的示例行解释了实际表格的概念结构。实际上，实际表格对于每一个成员总是使用整数关键字数值代替描述名来表示。因为实际表格往往包含极其多的行数（在极其大的数据仓库中，实际表格很可能会有数百万行），使用整数值可以很大程度地缩减实际表格的尺寸。关键字列可以是整数值或是日期值。实际表格的实际布局可以如以下所示：

STATE_ID	PROD_ID	Month	Sales_Units	Sales_Dollars
1	589	1/1/1998	3	7.95
1	1218	1/1/1998	4	7.32
2	589	1/1/1998	3	7.95
2	1218	1/1/1998	4	7.32
1	589	2/1/1998	16	42.40

1.2.2 维表格

实际表格只包含最底层的细节。如果有分成两类的8种产品，实际表格只能包含具体产品的