

新编高等院校信息管理与信息系统专业核心教材

数据仓库 与数据挖掘技术

Data Warehouse and Data Mining Techniques

陈京民 等编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

www.phei.com.cn

Data Warehouse and Data Mining Techniques

数据仓库与数据挖掘技术

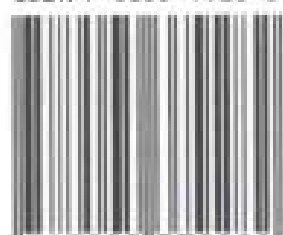
内容简介

本书介绍了数据仓库与数据挖掘技术的基本概念、基本原理、开发方法、开发工具、应用领域与管理方法等内容。全书共分为13章,包括数据仓库原理、数据仓库体系结构、数据仓库的开发工具、数据仓库规划分析方法、数据仓库开发实施方法、数据仓库应用管理方法、联机分析、数据挖掘基本原理、数据挖掘应用工具等内容。每章后都附有一定数量的习题,以帮助读者对全书的理解。

本书既可以作为高等院校信息管理与信息系统专业、通信专业、自动控制专业以及相关信息专业本科生的教材,也可作为管理类有关专业研究生的教材,同时还可作为企业、事业等单位从事信息管理与数据仓库开发应用工作人员的参考书。



ISBN 7-5053-7928-3



9 787505 379282 >



责任编辑:刘宪兰

特约编辑:明足群

封面设计:张 昱

本书贴有激光防伪标志,凡没有防伪标志者,属盗版图书。

ISBN 7-5053-7928-3/TP·4611 定价:33.00 元

新编高等院校信息管理与信息系统专业核心教材

数据仓库与数据挖掘技术

Data Warehouse and Data Mining Techniques

陈京民 等编著



A0975778

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

数据仓库与数据挖掘技术/陈京民等编著. —北京:电子工业出版社,2002.8
新编高等院校信息管理与信息系统专业核心教材
ISBN 7-5053-7928-3

I. 数… II. 陈… III. ①数据库系统—高等学校—教材②数据采集—高等学校—教材 IV. TP311.13
②TP274

中国版本图书馆 CIP 数据核字(2002)第 060968 号

责任编辑:刘宪兰 特约编辑:明足群

印 刷:北京牛山世兴印刷厂

出版发行:电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×980 1/16 印张:26.5 字数:551 千字

版 次:2002 年 8 月第 1 版 2002 年 8 月第 1 次印刷

印 数:5 000 册 定价:33.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010)68279077

总序

Z O N G X U



20 世纪 70 年代,当强大的信息化巨潮还蕴藏在大洋深处,我们的陆地只有一阵微风吹来之时,有识之士们就开始推动信息化专业人才的培养计划,为迎接即将到来的信息化巨潮扩军备战。他们一方面推动着信息技术的普及;一方面根据不同领域的需求,从不同的角度创办了不同类型的信息化专业,这就是管理信息系统专业、经济信息管理专业、科技信息管理专业、医学信息管理专业、林业信息管理专业、农业信息管理专业……实际上,这些专业培养目标可以概括为:为各行业、各部门培养以 CIO 为目标的信息化专门人才。从这一点上看,这些专业的课程设置应当具有相当大的共同性。1996 年,出于多种考虑,教育部将这些专业合并为一个——信息管理与信息系统专业。

以 CIO 为目标的信息化专门人才是一类管理人才。但是他们所管理的主要对象是信息。这样的知识需求,将信息管理与信息系统专业定位于管理学科,与信息学、经济学、法学等学科交叉。这样的学科特点,给课程建设和教材建设带来不少困难。近 30 年来,尽管我们与许多的同行已经进行了不懈的努力,把信息管理与信息系统专业的课程建设和教材建设向前推进了一大步,但是仍然不尽人意,许多课程和教材还没有体现信息管理专业的特色和需要。在多次有关的研讨会上,大家一致呼吁编写一套真正体现信息管理与信息系统专业特色的教材。

新编和出版一套专业教材是要冒风险的。而编写和出版一套以瞬息万变的信息和信息技术为管理对象的专业教材就要冒更大的风险。国内信息业界著名的出版商——电子工业出版社,以超人的胆略愿意同我们一道承担这一风险,组织编写出版一套新的信息管理与信息系统专业核心教材。这套教材冠以“新编”二字,是试图在其体系上能比已有教材更体现信息专业的特色,同时在内容上要能反映最新信息技术的进步以及最新信息管理思想和方法。

目前,国内开设信息管理与信息系统专业的高等院校已经超过 200 所。这样一个数字一方面表明信息化已经深入人心,信息化队伍的规模正在急速扩大,信息化队伍的素质正在不断提高;另一方面,也给我们添加了巨大的压力,

使我们深感责任重大。好在国内本领域的三位知名学者——黄梯云、陈禹、马费成以及其他一批有名专家和后起之秀愿意与我们共担风险，鼓舞了我们挑起这副重担的勇气。同时，我们也把这套教材的不断精化寄希望于广大的同仁，愿我们把这套教材越改越好，永改永新。

新编高等院校信息管理与信息系统
专业核心教材编委会
2002 年 5 月

前言

Q I A N Y A N

信息技术的迅速发展已将我们从简单的批处理、联机事务处理的信息处理时代，带入了联机分析处理、数据仓库和数据挖掘的信息分析时代。数据仓库在短短的几年内已经从一种单纯的理论研究发展成信息管理与信息系统开发领域中一种实用性极强的技术。这一发展具有其内在的动力和外在的推动。由于大多数企业在早期的信息化进程中构建了比较完善的信息处理系统——联机事务处理系统，这些系统为企业的业务快速、准确处理提供基本条件，同时为企业积累大量的、有价值的业务信息。这些处理只能支持企业的日常业务工作，而对企业的经营管理决策却无法提供支持。许多企业的经营管理人员在日趋严重的市场竞争压力下，开始着手建立自己的数据存储——数据集市，用于经营管理决策，以应对日益严酷的市场竞争压力。这些因素最终促进了数据仓库技术的发展。

数据仓库的建立不仅需要各种建设工具，而且还需相应的数据支持。数据仓库的建设必须基于比较完善的信息化构架，只有在一定的信息化基础上，才能进行数据仓库的建设。数据仓库的建设还是企业经营管理决策与信息化结合的过程，只有依照企业的管理决策实际情况，才能建设一个支持企业管理决策的数据仓库。数据仓库的建设还是各种先进的信息处理技术与企业管理决策结合的过程，只有将 OLAP 技术、数据挖掘技术与数据仓库中的庞大数据相结合，与企业先进的管理决策方法相结合，才能使数据仓库在企业的经营管理决策中发挥巨大的作用。数据仓库建设的成功不仅取决于技术人员对数据仓库开发方法与开发工具的熟练应用，更重要取决于数据仓库能否得到熟练应用。可以毫不夸张地说，数据仓库的成功关键在于用户的应用情况，而不是数据仓库开发技术的熟练应用。因此，本书在集中介绍数据仓库的开发模型与开发方法后，还用相当的篇幅介绍数据仓库的管理与应用。

本书还介绍了大量的数据仓库应用情况与应用案例，使读者了解如何利用数据仓库来降低企业的运营成本、建立更好的客户关系管理、提高产品的质量等。本书还介绍数据仓库开发应用的生命周期，数据仓库的整个开发过程从规划分析到设计实施、终结于应用管理，使读者了解数据仓库开发应用的完整周期，以及如何处理在不同阶段中所遇到的问题。为使读者通过实际的数据仓库开发利用，加深对数据仓库与数据挖掘的了解，本书还介绍一些数据仓库与数据挖掘的工具，使读者通过这些工具的实际应用，进一步加强数据仓库与数据挖掘技术的了解。

全书共分 13 章。第 1 章主要介绍数据仓库的产生背景、发展、用户类型、总体结构与使用技术；第 2 章介绍数据仓库的开发工具——Oracle 9i 在数据仓库开发中的应用；第 3 章介绍另一种数据仓库开发工具——SQL Server 2000 在数据仓库开发应用中的作用；第 4 章介绍数据仓库应用的前台开发工具——Delphi 6.0 在数据仓库开发中的应用；第 5 章从理论上介绍数据仓库的开发模型——概念模型、逻辑模型、物理模型、元数据模型和数据粒度模型；第 6 章叙述数据仓库开发应用的完整周期，涉及到数据仓库开发规划、需求分析、设计、实施、使用及支持等；第 7 章阐述数据仓库的实际开发过程与运行过程的技术支持；第 8 章介绍联机分析技术（OLAP）的基本概念、结构、实施以及 OLAP 工具评价标准；第 9 章阐述数据挖掘技术的发展过程，数据挖掘工具及其应用；第 10 章详细介绍统计分析类数据挖掘技术、工具、应用及其应用中的问题；第 11 章进一步介绍知识类数据挖掘技术与工具；第 12 章介绍文本挖掘、Web 挖掘等其他类型的数据挖掘工具与应用；第 13 章说明数据仓库的实际应用，应用中的问题和数据仓库开发应用中的管理问题。

本书第 1, 2, 3, 4, 8 章由陈京民编写；第 5, 13 章由杜冬军编写；第 6, 7 章由杜冬军、陈京民编写；第 9 章由俞强编写；第 10, 11 章由俞强、陈京民编写；第 12 章由朱惠云编写。全书最后由陈京民修改、统稿。

在本书的编著过程中，得到张基温教授的大力支持与帮助，且对本书提出了非常宝贵的评审与修改意见，在此表示衷心的感谢。

感谢刘宪兰在本书出版工作中所给予的大力支持与帮助。

由于编者水平有限，书中难免存在不当之处，而数据仓库与数据挖掘又正处于日新月异的发展过程之中，恳切希望各位读者批评指正。

电子邮件地址：cjm20020101@sina.com.cn

作 者

2002 年 5 月

目 录

第 1 章 数据仓库导论	(1)
1.1 数据仓库的发展及展望	(2)
1.1.1 从传统数据库到数据仓库	(2)
1.1.2 数据仓库的定义与基本特性	(5)
1.1.3 数据仓库的几个重要概念	(8)
1.1.4 数据仓库的未来发展	(10)
1.2 数据仓库的应用	(12)
1.2.1 数据仓库的两类用户——信息的使用者与知识的挖掘者	(12)
1.2.2 信息使用者的数据仓库应用	(13)
1.2.3 知识挖掘者的数据仓库应用	(13)
1.3 数据仓库总体结构	(15)
1.3.1 数据仓库的总体参考框架	(16)
1.3.2 数据仓库基本功能层	(16)
1.3.3 数据仓库的管理层	(24)
1.3.4 数据仓库的元数据管理层	(25)
1.3.5 数据仓库的环境支持层	(26)
1.4 数据仓库技术	(28)
本章小结	(31)
习题	(31)
第 2 章 Oracle 的数据仓库设计与使用	(33)
2.1 Oracle 数据仓库开发工具简介	(34)
2.1.1 Oracle 数据仓库的技术基础工具	(34)
2.1.2 Oracle 数据仓库的分析应用工具	(35)
2.1.3 Oracle 数据仓库创建工具	(36)
2.1.4 Oracle 数据仓库维护工具	(36)
2.2 Oracle 数据仓库创建	(36)
2.2.1 Oracle 数据仓库的创建	(36)
2.2.2 Oracle 数据仓库表空间的创建	(43)

2.2.3	Oracle 数据仓库表的创建	(48)
2.3	Oracle 数据仓库的维与立方创建	(56)
2.3.1	Oracle 数据仓库的维创建	(56)
2.3.2	Oracle 数据仓库的立方创建	(62)
2.4	Oracle 数据仓库的应用工具简介	(67)
2.4.1	Oracle 数据仓库的 OLAP 应用	(67)
2.4.2	Oracle 数据仓库的数据挖掘应用	(68)
	本章小结	(70)
	习题	(71)
第 3 章	SQL Server 的数据仓库设计与使用	(73)
3.1	SQL Server 数据仓库开发工具及应用	(74)
3.2	SQL Server 的数据仓库创建	(76)
3.2.1	创建数据库	(76)
3.2.2	创建表	(77)
3.3	SQL Server 中的数据仓库访问与操纵	(79)
3.3.1	Analysis Manager 数据库的创建与数据源的确定	(79)
3.3.2	用 Analysis Services 创建维	(82)
3.3.3	用 Analysis Services 创建多维数据集	(86)
3.3.4	用查询分析器 (Transact-SQL) 访问数据仓库	(92)
3.3.5	用 Microsoft English Query 操纵数据仓库	(92)
3.4	SQL Server 中的数据提取与加载	(95)
3.4.1	SQL Server 的数据复制工具与应用	(95)
3.4.2	DTS 的数据导出工具 (DTS Export Wizard)	(98)
3.4.3	DTS 的数据导入工具 (DTS Import Wizard)	(101)
3.4.4	DTS 的数据转换	(103)
3.5	SQL Server 中的数据挖掘工具与应用	(104)
3.5.1	SQL Server 中的数据挖掘工具	(104)
3.5.2	决策类数据挖掘工具的应用	(105)
3.5.3	聚类分析的数据挖掘工具应用	(110)
	本章小结	(113)
	习题	(114)
第 4 章	Delphi 中的数据仓库设计与使用	(115)
4.1	Delphi 简介	(116)

4.1.1	Delphi 的开发集成环境组成	(116)
4.1.2	Delphi 的菜单栏与应用	(117)
4.1.3	Delphi 的工具栏与应用	(117)
4.1.4	Delphi 的组件板与应用	(117)
4.1.5	Delphi 的对象检查器与应用	(118)
4.1.6	Delphi 的窗体与应用	(118)
4.1.7	Delphi 的代码编辑器与应用	(118)
4.1.8	Delphi 应用程序的设计过程	(119)
4.2	Delphi 中的数据库组件	(121)
4.3	DecisionQuery 组件	(122)
4.3.1	DecisionQuery 组件的主要属性	(122)
4.3.2	DecisionQuery 组件的主要方法	(124)
4.3.3	DecisionQuery 组件的主要事件	(124)
4.3.4	利用 DecisionQuery 组件选择需要分析的数据维	(125)
4.3.5	利用 DecisionQuery 组件选择数据的分析公式	(126)
4.4	DecisionCube 与 DecisionSource 组件	(126)
4.4.1	DecisionCube 组件的主要属性	(127)
4.4.2	DecisionCube 组件的主要方法	(129)
4.4.3	DecisionCube 组件的主要事件	(129)
4.4.4	DecisionSource 组件的主要属性	(130)
4.4.5	DecisionSource 组件的主要方法	(131)
4.4.6	DecisionSource 组件的主要事件	(131)
4.5	DecisionPivot 组件、DecisionGrid 组件与 DecisionGraph 组件	(132)
4.5.1	DecisionPivot 组件的主要属性	(132)
4.5.2	DecisionPivot 组件的主要方法	(132)
4.5.3	DecisionPivot 组件的主要事件	(133)
4.5.4	DecisionGrid 组件的主要属性	(133)
4.5.5	DecisionGrid 组件的主要方法	(134)
4.5.6	DecisionGrid 组件的主要事件	(134)
4.5.7	DecisionGraph 组件的主要属性	(134)
4.5.8	DecisionGraph 组件的主要方法	(140)
4.5.9	DecisionGraph 组件的主要事件	(141)
	本章小结	(141)

习题	(141)
案例 4.1	(141)
第 5 章 数据仓库开发模型	(145)
5.1 数据仓库的各种数据模型	(146)
5.2 数据仓库概念模型	(147)
5.2.1 概念数据模型	(147)
5.2.2 规范的数据模型	(150)
5.2.3 星型模型	(152)
5.2.4 雪花模型	(153)
5.3 中间层逻辑模型	(154)
5.4 物理数据模型	(155)
5.4.1 事实表模型设计	(156)
5.4.2 维模型设计	(157)
5.4.3 数据仓库物理数据模型的性能问题	(157)
5.5 元数据模型	(159)
5.5.1 元数据的类型与组成	(159)
5.5.2 元数据在数据仓库中的作用	(161)
5.5.3 元数据的收集	(164)
5.5.4 元数据的存储、管理与维护	(166)
5.5.5 元数据的用户与使用方法	(168)
5.5.6 元数据管理模型	(170)
5.6 数据仓库的粒度模型	(170)
5.6.1 数据粒度的划分	(171)
5.6.2 确定粒度的级别	(172)
本章小结	(173)
习题	(173)
第 6 章 数据仓库开发应用的阶段	(175)
6.1 数据仓库的生命周期	(176)
6.1.1 数据仓库的阶段性的	(176)
6.1.2 数据仓库的螺旋式开发方法	(177)
6.1.3 数据仓库的开发特点	(178)
6.2 数据仓库的规划	(179)
6.2.1 选择数据仓库实现策略	(180)

6.2.2	确定数据仓库的开发目标和实现范围	(181)
6.2.3	数据仓库的结构	(182)
6.2.4	数据仓库使用方案和项目规划预算	(184)
6.3	数据仓库的需求定义	(185)
6.3.1	定义业主的需求	(185)
6.3.2	定义设计者的需求	(185)
6.3.3	开发者的需求定义	(186)
6.3.4	最终用户的需求定义	(188)
6.3.5	数据仓库的数据模型设计	(189)
6.4	数据仓库的设计和 implement 阶段	(189)
6.4.1	数据仓库的数据源确定以及与业务处理系统接口的设计	(190)
6.4.2	数据仓库的体系结构与数据库设计	(190)
6.4.3	数据仓库的中间件设计	(192)
6.4.4	数据仓库的数据抽取	(193)
6.4.5	数据仓库的数据加载	(194)
6.4.6	数据仓库数据的复制与发行	(194)
6.4.7	数据仓库的测试	(195)
6.5	数据仓库的使用、支持和增强阶段	(196)
6.5.1	数据仓库的用户培训及支持	(196)
6.5.2	数据仓库的使用方式	(197)
6.5.3	数据仓库使用中的数据刷新	(198)
6.5.4	数据仓库的增强	(199)
	本章小结	(201)
	习题	(201)
第 7 章	数据仓库的开发过程	(203)
7.1	数据仓库的概念模型设计	(204)
7.1.1	概念模型的需求调查	(204)
7.1.2	概念模型的定义	(205)
7.1.3	概念模型的分析	(209)
7.1.4	概念模型的设计	(210)
7.1.5	概念模型文档与评审	(212)
7.2	数据仓库的逻辑模型设计	(213)
7.2.1	分析主题域	(213)

7.2.2	粒度层次的划分	(214)
7.2.3	确定数据分割策略	(215)
7.2.4	关系模型定义	(216)
7.2.5	数据仓库的实体定义	(216)
7.2.6	数据仓库的数据抽取模型	(217)
7.2.7	逻辑模型的评审	(219)
7.3	数据仓库物理模型的设计	(220)
7.3.1	数据仓库设计的规范	(220)
7.3.2	确定数据结构类型	(220)
7.3.3	确定索引策略	(221)
7.3.4	确定数据存放位置	(222)
7.3.5	确定存储分配	(223)
7.3.6	数据仓库物理模型的评审	(224)
7.4	数据仓库的运行技术管理	(226)
7.4.1	数据加载的一些问题	(226)
7.4.2	故障恢复管理	(227)
7.4.3	访问控制与安全管理	(227)
7.4.4	数据增长的管理	(228)
	本章小结	(230)
	习题	(230)
第8章	OLAP 技术	(231)
8.1	OLAP 技术基本概念	(232)
8.1.1	OLAP 的发展	(232)
8.1.2	OLAP 的特性	(233)
8.2	OLAP 与多维分析	(233)
8.2.1	几个基本概念	(233)
8.2.2	多维分析	(236)
8.2.3	维的层次关系	(237)
8.2.4	维的类关系	(238)
8.2.5	OLAP 与数据仓库关系	(239)
8.3	OLAP 的实施	(240)
8.4	基于多维的 OLAP	(241)
8.4.1	多维数据库	(241)

8.4.2	多维数据库的数据存储	(242)
8.4.3	多维数据库与数据仓库	(243)
8.5	关系 OLAP	(244)
8.5.1	ROLAP 的三个规则	(245)
8.5.2	ROLAP 的多维表示方法	(245)
8.6	OLAP 的选择与评价标准	(248)
8.6.1	MOLAP 与 ROLAP 的比较	(248)
8.6.2	OLAP 的衡量标准	(250)
8.6.3	OLAP 服务器和工具的评价标准	(253)
	本章小结	(255)
	习题	(255)
第 9 章	数据挖掘技术导论	(257)
9.1	数据挖掘概述	(258)
9.1.1	数据挖掘的发展	(258)
9.1.2	数据挖掘的定义	(259)
9.1.3	数据挖掘与数据仓库关系	(261)
9.2	数据挖掘技术与数据挖掘工具	(262)
9.2.1	常用数据挖掘技术	(262)
9.2.2	常用数据挖掘工具	(265)
9.2.3	数据挖掘工具的评价标准	(266)
9.2.4	常用数据挖掘工具的选择	(268)
9.3	数据挖掘技术的应用过程	(269)
9.3.1	数据挖掘过程	(269)
9.3.2	数据挖掘的用户	(274)
9.4	数据挖掘的应用范围	(274)
9.4.1	客户的细分应用	(276)
9.4.2	客户盈利能力分析	(277)
9.4.3	客户的获取与保持分析	(279)
9.4.4	市场营销中的应用	(280)
9.4.5	数据挖掘的其他应用	(281)
	本章小结	(282)
	习题	(283)
第 10 章	统计类数据挖掘技术	(285)

10.1	统计分析类数据挖掘技术	(286)
10.1.1	统计与统计类数据挖掘技术	(286)
10.1.2	数据的聚集与度量技术	(287)
10.1.3	柱状图数据挖掘技术	(287)
10.1.4	线性回归数据挖掘技术	(288)
10.1.5	非线性回归数据挖掘技术	(290)
10.1.6	聚类数据挖掘技术	(292)
10.1.7	最近邻数据挖掘技术	(295)
10.2	统计分析类工具	(297)
10.2.1	统计类数据挖掘工具与商业分析员	(297)
10.2.2	统计类数据挖掘工具的功能	(298)
10.2.3	统计类数据挖掘工具——SPSS	(299)
10.3	统计分析类工具的用途	(302)
10.3.1	趋势分析	(302)
10.3.2	时序分析	(303)
10.3.3	周期分析	(304)
10.4	统计分析类工具应用中的问题	(305)
10.4.1	统计类数据挖掘的预处理问题	(305)
10.4.2	统计分析遵循的基本原则	(307)
10.4.3	统计分析的步骤	(308)
10.4.4	统计类数据挖掘的性能问题	(309)
	本章小结	(310)
	习题	(310)
第 11 章	知识类数据挖掘技术	(313)
11.1	知识发现系统的一般结构	(314)
11.1.1	知识发现的定义	(314)
11.1.2	知识发现系统的结构	(315)
11.2	知识发现技术	(317)
11.2.1	规则型知识挖掘技术	(317)
11.2.2	神经网络型知识挖掘技术	(319)
11.2.3	遗传算法型知识挖掘技术	(321)
11.2.4	粗糙集型知识挖掘技术	(324)
11.3	知识发现技术的运用	(325)

11.3.1	关联规则的应用	(325)
11.3.2	神经网络的应用	(327)
11.3.3	遗传算法的应用	(328)
11.3.4	粗糙集的应用	(330)
11.4	知识发现工具的应用	(332)
11.4.1	知识发现工具的系统结构	(332)
11.4.2	知识发现工具运用中的问题	(334)
11.4.3	知识发现的价值	(336)
11.4.4	知识类数据挖掘工具简介	(337)
	本章小结	(338)
	习题	(339)
第 12 章	其他数据挖掘技术和工具	(341)
12.1	文本挖掘技术	(342)
12.1.1	信息检索系统	(342)
12.1.2	文本分析和语义网络	(344)
12.1.3	文本挖掘	(345)
12.2	Web 挖掘技术	(348)
12.2.1	Web 的特点	(348)
12.2.2	Web 内容挖掘	(349)
12.2.3	Web 结构挖掘	(350)
12.2.4	Web 使用记录的挖掘	(352)
12.2.5	Web 数据挖掘的应用	(353)
12.3	分类分析技术	(354)
12.4	可视化数据挖掘技术	(359)
12.4.1	数据可视化技术	(359)
12.4.2	可视化数据挖掘技术	(360)
12.5	地理信息系统与空间数据挖掘	(363)
12.5.1	地理信息系统	(363)
12.5.2	空间数据挖掘	(365)
12.6	分布式数据挖掘	(366)
12.6.1	概述	(366)
12.6.2	适合水平式数据划分的分布式挖掘方法	(367)
12.6.3	适合垂直式数据划分的分布式挖掘方法	(368)

本章小结	(369)
习题	(370)
第 13 章 数据仓库的应用与管理	(371)
13.1 数据仓库在信息管理中的实际应用	(372)
13.1.1 分层决策体系	(372)
13.1.2 数据抽样分析	(375)
13.1.3 发挥历史数据的经济效益	(376)
13.1.4 回扣分析	(377)
13.1.5 客户关系管理	(378)
13.2 数据仓库应用与数据挖掘中的法律问题	(379)
13.2.1 数据的隐私权问题	(380)
13.2.2 数据隐私权的处理	(380)
13.3 数据仓库开发与应用的成本/效益分析	(383)
13.3.1 数据仓库投资回报的定量分析	(383)
13.3.2 数据仓库投资回报的定性分析	(384)
13.4 数据仓库的开发与运行管理	(385)
13.4.1 数据仓库开发与应用的组织结构	(386)
13.4.2 数据仓库的项目开发管理	(388)
13.4.3 数据仓库应用的阶段性	(393)
13.4.4 数据仓库的运行管理	(396)
13.4.5 数据仓库的评价	(398)
本章小结	(400)
习题	(400)
参考文献	(403)



第 1 章

数据仓库导论

引 言

信息技术的不断推广应用,将企业带入了一个信息爆炸的时代。每日、每时、每刻都有如潮水般的信息出现在管理者的面前,等待管理者去处理、去使用。这些管理信息的处理类型主要有事务型处理和信息型处理两大类。事务型处理,也就是通常所说的业务操作处理。这种操作处理主要是对管理信息进行日常的操作,对信息进行查询和修改,目的是满足组织特定的日常管理需要。在这类处理中,管理者关心的是信息能否得到快速的处理,信息的安全性能否得到保证,信息的完整性是否会遭到破坏。信息型处理则是指对信息做进一步的分析,为管理人员的决策提供支持。例如,为决策支持系统、经理信息系统、战略信息系统等提供信息分析的支持。这种类型的信息处理在现代企业中的应用越来越广泛,越来越引起管理人员的重视。管理信息的信息型处理,必须访问大量的历史数据,才能完成。而不像事务型处理那样,只对当前的信息感兴趣。因此,在信息型处理中,产生了与操作型处理所采用的传统数据库有很大差异的数据环境要求。

通过本章学习,可以了解:

- ◆数据仓库的发展过程
- ◆两种不同类型的用户对数据仓库的应用
- ◆数据仓库的总体结构框架
- ◆数据仓库的功能结构
- ◆数据仓库的环境支持结构

1.1 数据仓库的发展及展望

传统数据库在联机事务处理(OLTP)中获得了较大的成功,但是对管理人员的决策分析要求却无法实现。因为,管理人员希望对组织中的大量数据进行分析,了解组织业务的发展趋势,而传统数据库中只保留当前的管理信息,缺乏决策分析所需要的大量历史信息。为满足管理人员的决策分析需要,在数据库基础上产生了能够满足决策分析所需要的数据环境——数据仓库(DW, Data Warehouse)。

1.1.1 从传统数据库到数据仓库

如何有效地管理企业在经营过程中所产生或收集的大量数据与信息,一直是信息管理人员所面临的一个重要问题。20世纪70年代所出现的关系数据库在收集、存储、处理数据中发挥了重要的作用。随着市场竞争的加剧,信息系统的用户已经不满足于仅用计算机去处理日复一日的事务数据,而是需要信息——能够支持决策的信息去帮助管理决策。这就需要一种能够将日常业务处理中所收集到的各种数据转变为具有商业价值信息的技术,而传统数据库系统已经无法承担这一责任。

传统数据库对日常事务处理十分理想,但是要基于事务处理的数据库帮助决策分析,就产生了很大的困难。其原因主要是传统数据库的处理方式和决策分析中的数据需求不相称,导致传统数据库无法支持决策分析活动。这些不相称性主要体现在决策处理中的系统响应问题,决策数据需求的问题和决策数据操作的问题。

1. 决策处理的系统响应问题

在传统的业务处理系统中,用户对系统和数据库的要求是数据存取频率要高,操作时间要快。由于用户对数据操作时间的短暂,使系统能够在多用户的情况下,也可保持较高的系统响应时间。

在决策分析处理中,用户对系统和数据的要求则发生了很大的变化。在决策分析中,有的决策问题请求,可能导致系统长达数小时的运行,有的决策分析问题的解决则需要遍历数据库中大部分数据。这就必定消耗大量的系统资源,而这些是事务联机处理系统所无法承担的。

2. 决策数据需求的问题

在进行决策分析时,需要有全面、正确的集成数据,这些集成数据不仅包含企业内部各部门的有关数据,而且还要包含企业外部的、甚至竞争对手的相关数据。但是在传

统数据库中,只存储了本部门的事务处理数据,而没有与决策问题有关的集成数据,更没有企业外部数据。如果将数据的集成问题交给决策分析程序解决,将大大增加决策分析系统的负担,使原先执行时间冗长的系统运行时间进一步延长,用户将更加难以接受。而且每次用户进行一次决策分析,都需要进行一次数据的集成,将极大地降低系统运行效率。如果数据库能够完成数据的集成,就可以大大提高系统的运行效率。

在决策数据的集成中需要解决因事务处理的分散,而造成的数据凌乱问题。企业的数据凌乱原因多种多样,有的是历史原因造成的,例如,在企业兼并活动完成后,被兼并企业的信息系统与原企业系统的不兼容。有的是系统开发的短视所造成的,例如,在系统开发中,由于资金的缺乏,只考虑了一些关键系统的开发,而对其他系统未予开发,使决策数据无法集成。面对这些凌乱的数据,还可能在决策分析应用中发生数据的不一致性。例如,同一实体的属性在不同的应用系统中,可能具有不同的数据类型、不同的字段名称。例如,客户的性别在销售系统中可能用逻辑值“M”和“F”表示,在财务系统中可能用数字“0”和“1”表示。或者同名的字段在不同的应用中有不同的含义,表示了不同实体的不同属性。例如,名称为“GH”的字段名称在人事管理系统中表示为职工的“工号”含义,但是在销售管理系统中表示为“购货号”。要解决这些问题,必须在决策分析之前对这些数据进行转换。

在决策分析中,系统常常需要从数据库中抽取数据、查找有用的数据,然后将这些数据置于其他文件或数据库中,供用户使用。这些被抽取出来的数据,有可能被其他用户再次抽取。由于这种不加限制的数据连续抽取,使企业的数据空间构成了一个错综复杂的数据“蜘蛛网”,即形成了自然演化体系结构。在这个数据“蜘蛛”网中,有可能两个节点上的数据来自同一个原始数据库。但是由于数据抽取的时间基准、抽取方法、抽取级别等方面的差异,有可能使这两个节点的数据不一致。这样,面对同一问题的决策分析,由于数据的出发基准不同,而导致截然相反的答案。也就是说,由于决策分析过程中所形成的自然演化体系,造成了数据可信度的降低,必然导致数据转化为信息的不可行与不可信,使企业无法将大量宝贵的信息资源转化为企业的核心竞争力。

数据的集成还涉及外部数据与非结构化数据的应用问题。决策分析中经常用到系统外数据,例如,行业的统计报告,咨询公司的市场调查分析数据。这些数据必须经过格式、类型的转换,才能被决策系统应用。许多系统在对数据进行一次集成以后,就与原来的数据源断绝了联系。这样在决策分析中,所分析的数据可能是几个月前甚至是一年以前的,其结果必然导致决策的失误。因此在决策分析系统中要求数据能够进行定期的、及时的更新,数据的更新期可能是一天,也可能是一周。而传统数据库系统缺乏数据动态更新的能力。

为完成事务处理的需要,传统数据库中的数据一般只保留当前的数据。但是对于决策分析而言,历史上的、长期的数据却具有更重要的意义。利用历史数据可对未来的发

展进行正确的预测，而传统的数据库却无法长期保留大量的历史数据。

在决策分析过程中，决策人员往往需要的并不是非常详细的数据，而是一些经过汇总、概括的数据。但在传统数据库中为支持日常的业务处理需要，只保留一些非常详细的数据，这对决策分析十分不利。

3. 决策数据操作的问题

在对数据的操作方式上，业务处理系统远远不能满足决策人员的需要。业务处理系统的结构基本上是一种典型的结构体系，操作人员只能使用系统所提供的有限参数进行数据操作，用户对数据的访问受到很大的限制。而决策分析人员对数据的操作则希望以专业用户的身份，而不是参数用户的身份进行。他们往往希望用各种工具对数据进行多种形式的操作，希望将数据操作的结果以商业智能的方式表达出来。传统的业务处理系统只能以标准的报表方式为用户提供信息，使用户很难理解信息的内涵，无法正确地用于管理决策。

由于系统响应问题、决策数据需求问题和决策数据操作问题的存在，导致企业无法使用现有的业务处理去满足决策分析的需要。因此，决策分析需要一个能够不受传统事务处理的约束，高效率处理决策分析数据的支持环境，数据仓库正可满足这一要求的数据存储和数据组织技术。

4. 数据仓库与传统数据库的比较

数据仓库虽然是从数据库发展而来的，但是两者在许多方面都存在着相当大的差异（见表 1-1 的数据仓库与数据库对比表）。从数据存储内容看，数据库只存放当前值，而数据仓库则存放历史值；数据库中数据的目标是面向业务操作人员的，为业务处理人员提供信息处理的支持，而数据仓库则是面向中高层管理人员的，为其提供决策支持。数据库内数据是动态变化的，只要有业务发生，数据就会被更新，而数据仓库则是静态的历史数据，只能定期添加、刷新。数据库中的数据结构比较复杂，有各种结构以适合业务处理系统的需要，

表 1-1 数据仓库与数据库对比表

	数据库	数据仓库
数据内容	当前值	历史的、存档的、归纳的、计算的数据
数据目标	面向业务操作程序，重复处理	面向主题域，分析应用
数据特性	动态变化，按字段更新	静态，不能直接更新，只能定时添加、刷新
数据结构	高度结构化、复杂，适合操作计算	简单，适合分析
使用频率	高	中到低
数据访问量	每个事务只访问少量记录	有的事务可能需要访问大量记录
对响应时间的要求	以秒为单位计算	以秒、分钟、甚至小时为计算单位

而数据仓库中数据的结构则较为简单。数据库中数据的访问频率高,但是访问数据的量少,而数据仓库的访问频率低但是访问数据量要远高于数据库的访问量。数据库在访问数据时要求响应速度很快,其响应时间一般要求在数秒以内,而数据仓库的响应时间则可长达数小时。

1.1.2 数据仓库的定义与基本特性

在数据仓库的发展过程中,许多人对此做出了贡献。其中,Devlin 和 Murphy 在 1988 年发表了一篇关于数据仓库论述的最早文章。而 William H.Inmon 在 1993 年所写的论著《Building the Data Warehouse》则首先系统性地阐述了关于数据仓库的思想、理论,为数据仓库的发展奠定了历史基石。为此,W. H. Inmon 被尊为数据仓库之父。在《Building the Data Warehouse》中,他将数据仓库定义为“一个面向主题的、集成的随时间变化的非易失性数据的集合,用于支持管理层的决策过程”。关于数据仓库的定义还有:“数据仓库是一种体系结构,一种独立存在的不影响其他已经运行的业务系统的语义一致的数据仓储,可以满足不同的数据存取、文档报告的需要”。数据仓库“是一个不断发展的过程,将多个异质的原始数据融合在一起,用于支持结构化的在线查询、分析报告和决策支持”。

从 W.H.Inmon 关于数据仓库的定义中,可以发现数据仓库具有这样一些重要的特性:面向主题性、数据集成性、数据的时变性、数据的非易失性、数据的集合性和支持决策作用。

1. 面向主题性

面向主题性表示数据仓库中数据组织的基本原则,数据仓库中的所有数据都是围绕着某一主题组织、展开的。由于数据仓库的用户大多是企业的管理决策者,这些人所面对的往往是一些比较抽象的、层次较高的管理分析对象。例如,企业中的客户、产品、供应商等都可作为主题看待。从信息管理的角度看,主题就是在一个较高的管理层次上对信息系统中的数据按照某一具体的管理对象进行综合、归类所形成的分析对象。从数据组织的角度看,主题就是一些数据集合,这些数据集合对分析对象进行了比较完整的、一致的数据描述,这种描述不仅涉及数据自身,还涉及数据之间的联系。

数据仓库的创建、使用都是围绕着主题实现的。因此,必须了解如何按照决策分析来抽取主题,所抽取出的主题应该包含哪些数据内容,这些数据内容应该如何组织。在进行主题抽取时,必须按照决策分析对象进行。例如,在企业销售管理中的管理人员所关心的是本企业哪些产品销售量大、利润高?哪些客户采购的产品数量多?竞争对手的哪些产品对本企业产品构成威胁?根据这些管理决策的分析对象,就可以抽取“产品”、“客户”等主题。

确定主题以后,需要确定主题应该包含的数据。此时,应该注意不能将围绕主题的数据与业务处理系统中的数据相混淆。例如“产品”主题在销售业务处理系统中已有数据存在,但是这些数据未必都能用于数据仓库。因为在业务处理系统中,数据组织的目的在于如何更加有效地处理产品的销售业务。因此,可能采用“产品订单”、“产品订购细则”、“产品库存”、“客户”等数据库来描述产品的销售活动。但是在对产品销售所进行的决策分析中,分析哪些客户订购产品量大时,只有客户才是所需要分析的对象。而“产品订单”、“产品订购细则”、“产品库存”等数据只是业务处理系统中的业务操作数据。但是,仅仅使用业务处理系统中的“客户”数据,又不能完成对“客户”的分析,因为还需要了解客户的产品采购量、最后一次采购时间、购买竞争对手的产品等数据,这就需要围绕“客户”这一主题重新进行数据的组织。在围绕“客户”主题进行数据组织时,不适合决策分析要求的数据可能需要抛弃。例如“产品库存”就不需在数据仓库中出现。有的则要将关于某一主题的、散落在其他业务处理系统中的信息组织进来。例如,客户的“信用”信息存在于财务处理系统中,在进行客户的决策分析时,需要了解这一信息,就要将其组织进来。而有的信息则可能存在于企业的外部系统中,在决策分析中需要使用,也要将其组织到所分析的主题中。例如,客户购买竞争对手产品的信息是从企业的销售代理商或市场调查公司那里所获取的,不是企业内部的数据,但是也需要组织到“客户”主题中。

在主题的数据组织中应该注意,不同的主题之间可能出现相互重叠的信息。例如,“客户”主题与“产品”主题在产品购买信息方面有相互重叠的信息。这种重叠信息往往来源于两个主题之间的联系,例如,“客户”主题与“产品”主题在产品购买信息方面的相互重叠,是源于与客户和产品都有关的销售业务处理系统。这种主题间重叠是逻辑上的重叠,而不是同一数据内容的物理存储重复。

主题在数据仓库中可用多维数据库方式进行存储。如果主题的存储量大,用多维数据库存储时,处理效率将降低。为提高处理效率,可以采用关系数据库方式进行存储。应该注意主题只是逻辑上的一个概念,一个主题在数据仓库中存储时可能需要几个表来实现。此时,这些表之间的相互联系需要通过表的主键来实现,这些主键就构成了主题的公共主键。实际存储的主题数据是需要经过综合处理的,而不再是业务处理系统中的详细数据。由于数据仓库的数据存储容量巨大,应该根据用户对主题中不同表的关心程度分别存储在不同的存储设备上。一般将年代久远的、细节性的、查询概率低的数据存储在磁带等慢速存储设备上,将近期的、综合的、查询概率高的数据置于磁盘等高速存储设备上。

在主题的划分中,必须保证每个主题的独立性。也就是说,每一个主题要具有独立的内涵,明确的界线。在划分主题时,需要保证对主题进行分析时所需要的数据都可以在此主题内找到。如果对主题进行分析时,涉及主题外的其他数据,就要考虑将这些数

据组织到主题中,以保证主题的完备性。

由于主题是在较高层次上的数据抽象,这就使面向主题的数据组织可以独立于数据的处理逻辑,可以很方便地在这种数据环境上进行管理决策的分析处理。

2. 数据集成性

数据仓库的集成性是指根据决策分析的要求,将分散于各处的源数据进行抽取、筛选、清理、综合等集成工作,使数据仓库中的数据具有集成性。

数据仓库所需要的数据不像业务处理系统那样直接从业务发生地获取,而是从与业务处理发生直接联系的业务处理系统那里获取。这里所指的业务处理系统应该包含这样一些系统:传统的以客户机/服务器为基本框架的在线事务处理系统(OLTP)、从早期事务处理系统发展起来的企业业务流程重组(BPR)以及基于因特网的电子商务(EC)。这些业务处理系统中的数据往往与业务处理联系在一起,只为业务的日常处理服务,而不是为决策分析服务。这样,数据仓库在从业务处理系统那里获取数据时,并不能将原数据库中的数据直接加载到数据仓库中,而是需要进行一系列的数据预处理。即数据的抽取筛选、清理和综合等集成工作。也就是说,首先要从源数据库中挑选出数据仓库所需要的数据。然后,将这些来自不同数据库中的数据按照某一标准进行统一,即将数据源中数据的单位、字长与内容统一起来,将源数据中字段的同名异义、异名同义现象消除掉,这些工作通称为数据的清理。在将源数据加载进数据仓库后,即源数据装入数据仓库后,还需要将数据仓库中的数据进行某种程度的综合,即根据决策分析的需要对这些数据进行概括和聚集处理。

3. 数据的时变性

数据仓库的时变性,就是数据应该随着时间的推移而发生变化。尽管数据仓库中的数据并不像业务数据库那样直接反映业务处理的目前状况,但是数据也不能长期不变。如果依据10年前的数据进行决策分析,那决策所带来的后果将是十分可怕的。因此,数据仓库必须能够不断地捕捉业务系统中的变化数据,将那些变化的数据追加到数据仓库中去,也就是在数据仓库中不断地生成业务数据库的快照,以满足决策分析的需要。数据快照生成的间隔,有的是每天一次,有的是每周一次,可以根据快照生成速度的快慢和决策分析的需要而定。快照的生成时间一般选择在业务系统处理较空闲的夜间或假日进行。这些快照是业务处理系统的某一时间的瞬态图,而这些瞬态图则构成数据仓库中数据的不同画面,这些画面的连续播放可以产生数据仓库的连续动态变化图,这对高层决策者的决策分析十分有益。

数据仓库数据的变化,不仅反映在数据的追加方面,而且还反映在数据的删除上。尽管数据仓库中的数据可以长期保留,不像业务系统中的数据那样只保留数月。但是在

数据仓库中的数据存储期限还是有限的，一般保留 5~10 年，在超过限期以后，也需要删除。

数据仓库中数据的变化性还表现在概括数据的变化上。数据仓库中的概括数据是与时间有关的，概括数据需要按照时间进行综合，按照时间进行抽取。因此，在数据仓库中的概括数据必须随着时间的变化而重新进行概括处理。为满足数据仓库中数据的时变性需要所进行的操作，一般称为数据刷新。

4. 数据的非易失性

数据仓库的数据非易失性是指数据仓库中的数据不经常进行更新处理，因为数据仓库中数据大多表示过去某一时刻的数据，主要用于查询。不像业务系统中的数据库那样，需要经常进行修改、添加，除非数据仓库中的数据是错误的。数据仓库的操作除了进行查询以外，还可定期进行数据的加载，即追加数据源中新发生的数据。数据在追加以后，一般不再修改，因此在数据仓库中可以通过使用索引、预先计算等方式提高数据仓库的查询效率。数据的非易失性，可以支持不同的用户在不同的时间查询相同的问题时，获得相同的结果。消除了以往决策分析过程中面对同一问题，而结论不同的尴尬。

5. 数据的集合性

数据仓库的集合性意味着数据仓库必须以某种数据集合的形式存储起来。目前数据仓库所采用的数据集合方式主要是以多维数据库方式进行存储的多维模式，以关系数据库方式进行存储的关系模式或以两者相结合的方式存储的混合模式。

6. 支持决策作用

数据仓库组织的根本目的在于对决策的支持。不同层次的管理层均可利用数据仓库进行决策分析，提高自己工作的管理决策质量和效果。因此，在数据仓库的实际应用中，其用户有高层的企业决策者、中层的管理者和基层的业务处理者。

企业管理者已经不满足于由信息管理部门所提供的静态报告。由于信息管理部门中的 IT 人员缺乏业务决策者所特有的、敏锐的商业洞察力和业务知识，无法为管理决策者提供有利于管理决策的信息；而管理决策者可从貌似平淡的数据中敏锐地发现众多的商机。因此，单纯地依靠信息管理部门提供信息已经不能满足管理决策的需要。数据仓库为决策者对数据的自我分析提供了便利，提供了辅助决策分析的有力工具。

1.1.3 数据仓库的几个重要概念

在数据仓库中关于维、立方体、超立方体、聚集的概念十分重要，只有理解了这些

概念，才能了解数据仓库。

1. 维

数据仓库是用于决策支持的。管理人员在进行决策分析时，经常需要选择一个对决策活动有重要影响的因素去进行决策分析。因此，用户在使用数据仓库时，也要有一个出发点。管理人员可从客户的角度、产品的角度，或者从供应商、地点、渠道、事件发生的时间等角度，讨论决策问题。用户的这些决策分析角度或决策分析出发点构成了数据仓库中的维，数据仓库中的数据就按照这些维来组织，维也就成了数据仓库中识别数据的索引。数据仓库中的维还可以作为数据仓库操作过程的路径，这些路径通常位于维的不同层次结构中。例如，客户可以按照地理位置进行分组：街道、县、市、省。这样就可以按照街道、县、市、省的先后次序进行数据的“上卷”和“下钻”。这里的所谓数据“上卷”是指用户在数据仓库的应用中，从较低层次的数据开始逐步地将数据按照层次进行概括处理。而“下钻”则是指从数据仓库中的高层数据开始逐步向低层数据探索，了解概括性数据的具体细节。

数据仓库中的维，一般具有层次性。水平层次由维度层次中具有相同级别的字段值构成，例如图 1.1 维度层次关系中的东部、中部和西部层次。垂直层次则由维度层次结构中具有不同级别的字段值构成。例如，图 1.1 中的东部、上海层次。

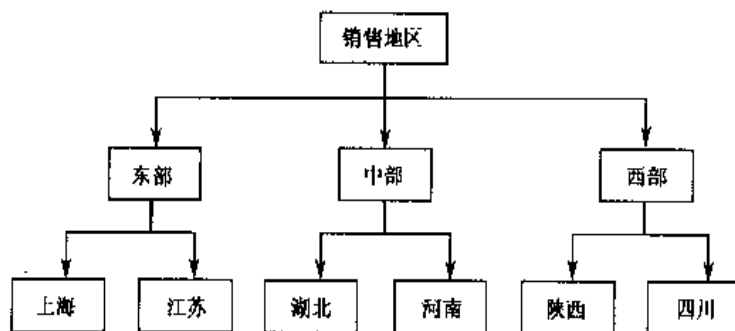


图 1.1 维度层次关系

在数据仓库的设计中要根据用户需求调查所获取的维，构成数据仓库的模型。这些模型主要有星型模型和雪花模型等。

2. 数据立方体

当用户观察某一事务的角度不同时，围绕该事务会产生多个观察角度，也就是说产生了多维。数据仓库中的多种维交点，就是数据仓库用户所需要观察的事务。例如在图 1.2 数据立方体中由客户、产品、时间三个维所构成的立方体表示哪些客户、在什么时间购买了哪些产品。三个维的交点就是所购买的产品数量或价格等事务，也就是立方体的顶点。数据仓库的立方体实际上是一个包含用户需要观察数据的集合体，它提供企业所感

兴趣的商业事务。在这里最重要的是购买的产品与价格等信息, 这些信息构成了立方体的粒度, 即维交叉时所导致的细节等级, 如果某个客户购买了某种产品, 结果就是一个基本粒度或原子事务。立方体作为基本事务的聚合, 是一种适合通过 SQL 或其他接口进行查询的完整的数据结构。一般而言, 立方体可以转换成星型模型, 而星型模型也可以转换成立方体。在数据仓库的实际应用中, 高层的数据聚合存储采用立方体处理, 效率较高。而以细节为基础, 维变化的上卷聚合采用星型模型处理效率更高并且更灵活。

数据仓库中的立方体在有的资料中也称为多维数据集, 两者的含义是一致的。如果数据立方体的维度超过 3 个, 就称其为超立方体, 也称这种超立方体为超维数据集。

3. 聚集

聚集或聚合是指收集了基本事务数据的结构。在一个立方体中包括很多层次, 这些层次可以向用户提供某一层次的概括数据。因为管理者在进行决策分析的过程中, 并不是要观察每一个详细的数据, 而是根据自己的管理范围进行总体情况的了解。例如, 地区销售经理想了解本地区的销售总量、未来的销售趋势、客户的类型, 那就需要按照本地区的城市、街道、产品种类和客户类型进行概括, 也就是进行聚集。通过聚集, 形成基于维的有决策分析意义的一些数据交集。

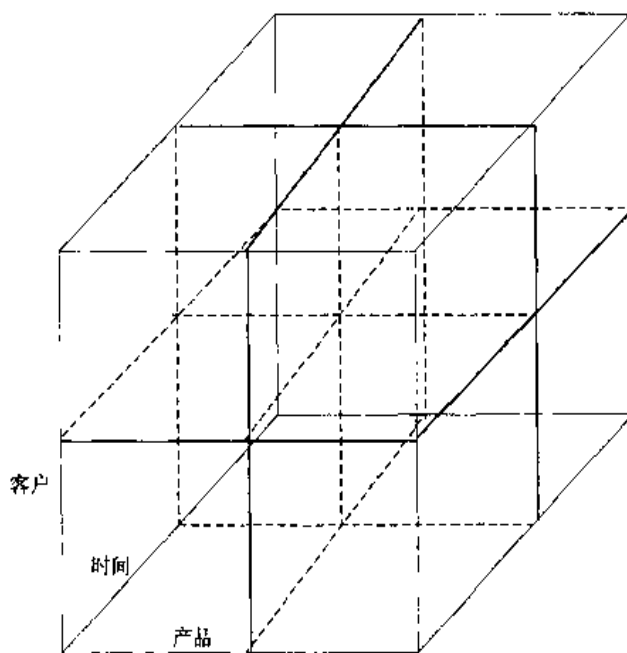


图 1.2 数据立方体

1.1.4 数据仓库的未来发展

随着数据仓库应用的扩展, 对数据仓库提出越来越多的要求。其中, 主要有基于关

系对象数据库的数据仓库、网络的影响、操作型数据仓库要求、Web 中的代理技术等。

1. 基于关系对象数据库的数据仓库

关系对象数据库的出现使数据仓库设计人员有能力将对象引入数据仓库环境中。关系对象数据库的出现，也将使数据仓库的平台性能得到很大的改善。数据仓库开发人员能够很容易地将多媒体数据、复杂的数据类型和其他各种类型的数据引入数据仓库。这就可使数据仓库满足用户的更多需求。对象技术引入数据仓库以后，一方面产生对数据仓库的更多数据、更多用户和更多可扩展性要求，同时对象技术也可用来提高数据仓库的性能以缓解扩展性要求的压力。

因为在对象技术引入数据仓库以后，用户可以定义适合某种数据类型的最佳操作。例如，基于关系型的数据仓库在处理“时间序列”数据时效率十分低下。因为在数据仓库中对基于时间序列数据的操作，常会要求查看这些数据如何按照时间序列变化。而这些数据在实际存储中并不按照时间的标识存储在一起，这些在时间上连续的数据可能被分散存储在大量的、不连续的磁盘中。在对这些数据的操作中，数据查询效率将十分低下。在关系对象数据库中则可以将这些具有时间序列的数据作为一个对象看待，而且在存储这些数据时将它们作为一个整体存储在磁盘上，这将大大地减少读取数据的时间。

关系对象数据库作为数据仓库平台，不仅为复杂数据类提供可扩展功能，而且还为数据仓库平台提供对数据处理的功能扩展方面。当用户的需求增长时，用户可用客户端的特定功能，扩展数据仓库平台的性能。数据仓库平台的可扩展性功能对数据仓库的应用是十分关键的，因为数据仓库的应用存在不断扩展的趋势。具有可扩展性的关系对象数据库可以满足用户在这些方面的要求。

2. 网络的影响

未来的数据仓库将越来越依赖于网络进行数据的传输和数据的使用请求处理。用户可以借助内部网络或外部网络使用数据仓库，这就需要数据仓库具有网络使用方面的能力。网络的使用能力不仅涉及企业内部的局域网，而且更多地涉及因特网。这就要求 Web 网关不仅能够将来自 Web 服务器的超文本语言（HTML 或 XML）格式转换成特定数据引擎的 API，而且能够将数据引擎中的答案转换为（HTML 或 XML）格式，实现数据源的抽取、转换和装载，在不同软件工具间进行元数据和内容的交换，且为数据仓库集成数据。

3. 操作型数据仓库

“操作型数据仓库”能以一种可以接受的标准对数据仓库进行操作。这些标准包括可预测性、可利用性和可访问性。在操作型数据仓库中有时还要对数据仓库进行修改。在日前的数据仓库中，对数据的更新是通过加载程序，将每个数据更新周期中所发生变

化的数据整批添加到数据仓库中去。在实际操作中常会由于某种现实的变化,需要对数据仓库中的记录进行少量的修改;随着外部数据源的增加,管理决策分析对及时数据需求的紧迫性,对数据的修改要求越来越强烈。但是,这种要求至少在目前无法完成。

4. Web 应用中的代理技术

数据仓库的 Web 应用主要是指用户利用合作伙伴的数据仓库或对本组织的 Intranet 系统的多维数据集进行决策分析活动。Web 的数据仓库访问,意味着可为企业带来大量的用户,而且需要为 Web 数据仓库提供更多的数据,尤其是图像数据。

因此 Web 数据仓库的实现,必须要求基于 HTTP 或 HTTPS 的数据源能够得到支持。从数据仓库的 Web 应用看,实际上是一种巨大的分布式计算环境应用。在大量的分布式计算中,单纯的依赖目前所采用的数据发布和订阅模型中的预先建立的协调性是无法实现的。这需要一种系统的代理来完成,即依靠软件代理系统实现。这种代理可以通过网络的下载程序和客户的特性以及合作处理,完成数据的推式定制工作,使数据仓库的 Web 应用与管理更加便利。

1.2 数据仓库的应用

1.2.1 数据仓库的两类用户——信息的使用者与知识的挖掘者

从数据仓库的最终用户来看,可以分成信息的使用者和知识的挖掘者两大类型。

信息的使用者是以一种可以预测的、重复的方式来使用数据仓库。信息使用者在使用数据仓库之前知道他们要了解什么,常常是每天都对数据仓库进行有规则的数据访问,在访问过程中往往只访问很少的一部分数据,而且对数据的访问常常能够获得结果。信息使用者通常要观察一些概括性数据或聚集数据,对一些元数据或详细数据很少用到。从信息使用者的工作性质看,他们往往是一些业务员性质的用户,使用一些预先定义好的查询,在概括性数据上进行运作,执行一些简单的处理。因此,适合他们的数据存储模式是星型结构。

知识的挖掘者对数据仓库的使用是不规则的,有时很长时间不使用数据仓库,有时却连续地长期使用。在使用数据仓库中,常需要对数据仓库中的海量数据进行挖掘。挖掘的目标可能是:在企业所面对的客户群中哪些客户是使企业盈利的客户?这些盈利客户应该具有哪些特征?这些盈利客户在采购过程中常常采购哪些产品?所采购的这些产品相互间具有什么关系?知识挖掘者在进行知识的挖掘过程中,常常一无所获;但是一次偶然的得手,会使数据仓库的巨大投资得到丰厚的回报。知识挖掘者往往是一些专业用户,他们负责管理报告的筹建与分析,在数据仓库的使用中,很少进行预先定义的查

询,而是提交一些复杂的、动态的查询,要求数据仓库进行一些复杂的数据处理。

1.2.2 信息使用者的数据仓库应用

数据仓库的信息使用者常常是在战术管理层上利用数据仓库监控企业战略实施的效果。即通过对企业经营状况的关键指标的监控,判断某一经营战略是否有效,且将具体的评价效果反馈给知识的挖掘者。知识挖掘者再利用数据仓库探索为什么会产生这样的效果,从数据仓库中去挖掘产生这些现象的内在知识。

信息使用者在利用数据仓库进行企业运行状态监控时,所涉及的监控指标可能有:资金的流动速度、客户的生存期价值、客户的满意度、未兑现票据量、未兑现票据比率、客户的欺诈趋势以及客户平均采购量等。

信息使用者在使用数据仓库时,往往只涉及数据仓库的运作领域,在提交查询要求后,常常希望在很短的时间内得到响应。这些用户所提交的查询大多是一些预先定义好的查询,为完成这些预先定义好的查询,数据仓库的设计者必须在数据仓库的设计过程中知道这些预定义查询的需求、预定义查询的数据结构和数据内容。信息使用者在使用数据仓库时常常使用一些概括性的数据。概括性的数据在数据仓库设计时就要根据用户的需求假设进行筹建,要确定概括哪些数据?如何概括?何时概括?这些事先的假设能否确定,对信息使用者使用数据仓库十分重要。如果概括数据能够满足用户的分析标准,可以节省用户决策分析中的大量时间和精力;如果不能满足用户的分析需要,则概括数据就毫无价值。信息使用者在数据仓库的使用中大多使用数据集市(data mart),支持自己的分析应用。

1.2.3 知识挖掘者的数据仓库应用

数据仓库的知识挖掘者对数据仓库的应用主要涉及这样两个方面:一是对从不知晓的企业运营的内在知识进行挖掘,希望挖掘出隐含在企业数据内部的一些商业知识、商业模式,为制定企业的发展战略、培养企业的核心竞争力提供帮助;另一个是针对企业过去的成功或失败,探索成功与失败的原因,使企业继续保持成功或免蹈覆辙。

知识挖掘者使用数据仓库一般主要有概括分析、数据抽取、建模分析和分类处理 4 个过程。

1. 概括分析

知识挖掘者在使用数据仓库时,首先要对数据仓库中的数据外部特征进行分析,确保数据的完整性和准确性,评价是否有充分的样本数据可以进行数据抽取、建模与分类

处理。知识挖掘者所进行的概况分析内容可能有：常来采购的客户性别比例多大？共有多少客户？经常进行采购的客户数量比例情况如何？客户的平均采购标准是多少？有多少客户超过平均采购标准？有多少客户低于平均采购标准？通过这些类似的概况分析可使知识挖掘者了解到数据仓库中所包含数据的概况，才能对数据仓库进行更深一步的知识挖掘。

2. 数据抽取

知识挖掘者在了解数据仓库中的数据概况后，可以根据知识挖掘的需要对数据仓库中的数据进行抽取。数据抽取工作是根据知识挖掘的需要和概况分析的结果，将需要进行分析的数据从数据仓库中抽取出来。按照数据分析的目的，对这些数据进行组织，然后将组织好的数据送入数据集市或知识挖掘库中。

3. 建模分析

知识挖掘中的建模分析是知识挖掘者使用数据仓库的核心工作，是开发一种用于描述客户、产品或销售商模型的过程。在完成建模分析以后，就可以利用所建的模型对数据仓库中的实体与模型的关联程度进行分类分析。

例如，在企业中常常需要通过建模，分析哪些是可能拖延支付货款的客户。为完成这一知识挖掘，首先需要利用统计学和行为科学来确认经常拖延支付货款的客户特征。此时，知识挖掘者可以使用统计类的数据挖掘工具，开发这个模型。然后，就模型的分析结果对数据仓库中的所有客户数据进行分类，从中找出那些可能拖延支付货款的客户。最后，将这些客户名单作为重点关注的客户目标，通知有关的销售系统和管理人员。

在数据仓库的知识挖掘应用中，经常采用的模型有客户分类、后继产品购买分析、信用分析、欺诈检测、客户生存期价值分析和客户推销响应分析等。

4. 分类处理

分类处理是知识挖掘者在知识挖掘过程中的最后一项活动。通过建模分析，知识挖掘者从所建模型中分析出所需要的知识，就可以根据所挖掘出来的知识对数据仓库中的所有数据进行分类。分类的目的是针对不同的事务采取正确的对策，这也是知识挖掘者挖掘知识的最终目的。

知识挖掘者是一些承担管理任务的管理者或提出管理建议的商业分析员，他们是一些专业用户，经常会向系统提出一些复杂的、动态的查询。他们在进行数据仓库应用时，对系统的响应时间要求不高，因为他们主要的工作是进行一些长期决策的制定。在时间上的要求并不是很苛刻，但是他们常常会对数据仓库中的数据进行寻根问底的查询，可能用上数小时将数据仓库来个“底朝天”，因此知识挖掘者对数据仓库的应用常常导致数

据仓库性能的下降。为保证其他数据仓库使用者的应用效率，常为知识挖掘者构建一个类似数据集市的数据集合——知识挖掘库，为知识挖掘者进行特定的知识挖掘操作提供便利。

一般情况下，知识挖掘者除拥有专门领域知识外，还拥有较深厚的计算机知识背景，具有处理详细、复杂数据的能力。并且了解数据仓库的结构和数据仓库的使用工具，能够从管理人员所提供的需求草图中创建分析，对数据仓库进行挖掘。

1.3 数据仓库总体结构

数据仓库是近年 IT 技术和信息管理迅速发展的结果。如果从数据仓库的概念结构看，应该包含数据源、数据准备区、数据仓库数据库、数据集市/知识挖掘库以及各种管理工具和应用工具（见图 1.3 的数据仓库的概念结构）。数据仓库在创建以后，首先要从数据源中抽取所需要的数据到数据准备区，在数据准备区中经过数据的净化处理，再加载到数据仓库数据库中，最后根据用户的需求将数据发布到数据集市/知识挖掘库中。当用户使用数据仓库时，可以通过 OLAP 等数据仓库应用工具向数据集市/知识挖掘库或数据仓库进行决策查询分析或知识挖掘。数据仓库的创建、应用可以利用各种数据仓库管理工具辅助完成。

为了更清楚地了解数据仓库的功能和数据仓库的组织，可用 Zachman 信息框架将整个数据仓库体系结构分解成许多更容易理解的框架结构。这些框架结构仅仅是个帮助理解数据仓库的参考框架。在数据仓库的实际创建中，应该根据所选用的数据仓库创建工具来具体确定数据仓库的结构框架，因此这里所介绍的结构框架在实际应用中可能有所变化。

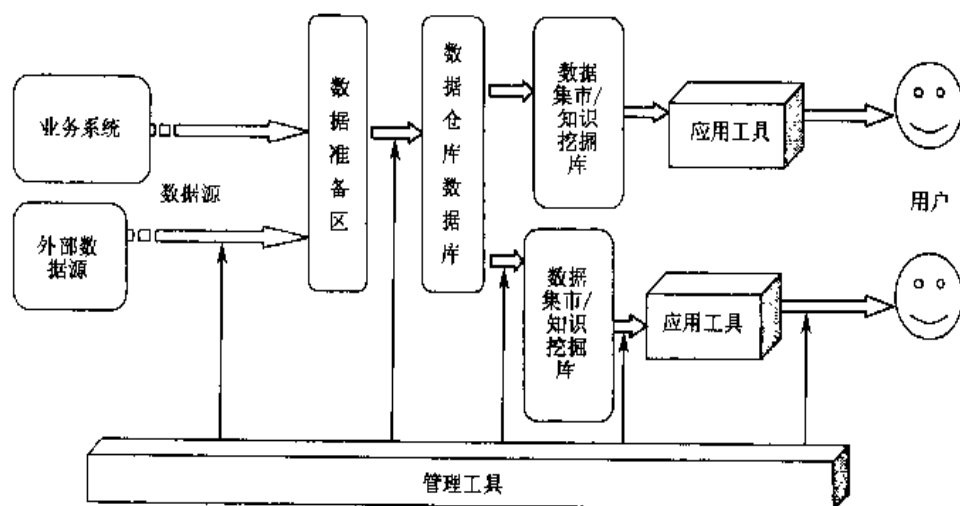


图 1.3 数据仓库的概念结构

1.3.1 数据仓库的总体参考框架

为实现数据仓库的功能，数据仓库的总体层次结构应该由数据仓库基本功能层、数据仓库管理层和数据仓库环境支持层（见图 1.4 的数据仓库总体框架结构）组成。

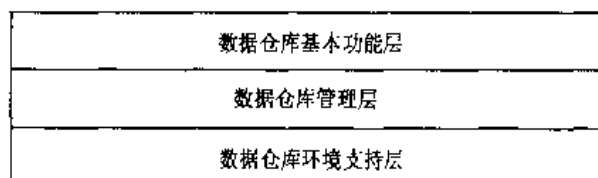


图 1.4 数据仓库总体框架结构

数据仓库的基本功能层应该包含从数据源抽取数据，对所抽取的数据进行筛选、清理，将清理后的数据加载到数据仓库中，根据用户的需求设立数据集市，完成数据仓库的复杂查询、决策分析和知识的挖掘等功能。

数据仓库的管理层包含数据管理与元数据管理两部分。数据管理与元数据管理主要负责对数据仓库中的数据抽取、清理、加载、更新与刷新等操作进行管理。只有使这些操作正常完成，才能源源不断地为数据仓库提供新的数据源，才能使数据仓库的使用者正确地利用数据仓库进行决策分析和知识挖掘。

数据仓库环境支持层主要包含数据传输和数据仓库基础两大部分。这两大部分对于数据仓库的创建和使用来说是必不可少的，没有这两个数据仓库的支持环境，数据仓库的创建与使用是无法实现的。

这里所列出的数据仓库总体结构框架，并不是每个层次和功能结构块都需要在数据仓库创建中生成。其中的数据源功能块与数据传输、数据仓库基础结构基本上可以采用组织中原有的信息系统，或在原系统的基础上略作修改就可满足需要。数据仓库的创建主要完成数据仓库结构、数据集市/知识挖掘库结构和存取与使用功能块，以及数据管理和元数据管理的设计与实现。

1.3.2 数据仓库基本功能层

数据仓库的基本功能部分包含数据源、数据准备区、数据仓库结构、数据集市或知识挖掘库，以及存取与使用功能部分（见图 1.5）。

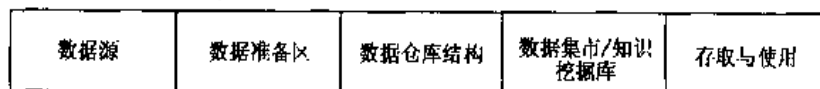


图 1.5 数据仓库功能结构

1. 数据源

数据仓库的数据源是指存储在数据仓库中的数据来源。从数据仓库在使用过程中所涉及的数据来源看,数据源结构应该包含业务数据、历史数据、办公数据、Web 数据、外部数据以及数据源元数据(见图 1.6)。

业务数据	历史数据	办公数据	Web 数据	外部数据	数据源元数据
------	------	------	--------	------	--------

图 1.6 数据源结构

(1) 业务数据

业务数据是指那些从组织目前正在运作的业务处理系统那里收集到的,并且保存在业务处理系统的数据存储中的数据。业务处理系统的数据存储可能是由关系型数据库、非关系型数据库或文件系统所构成的。对业务数据,必须分析哪些数据可以加载到数据仓库的事实表和维表中。若因各种原因,在数据仓库生成以后,才能决定某个业务数据需要加入数据仓库,那该业务数据就是一个新的数据源。需要先对数据仓库的原始数据模型进行维度分析,再根据现有数据模型定义新的事实表或扩充原有的事实表,且为源数据定义新的维表。

(2) 历史数据

历史数据是指组织在长期的信息处理过程中所积累下来的数据,这些数据一般从业务处理系统那里进行脱机处理,以磁带或其他脱机存储设施保存,对业务系统的当前运行不起作用。但是这些历史数据对于数据仓库的用户却具有重要的使用价值,尤其是知识挖掘用户在进行知识挖掘时,需要大量的历史数据。这些数据需要根据数据仓库模型和用户的决策分析需求,确定是否要加载进数据仓库。

(3) 办公数据

办公数据主要是指组织内部的办公系统数据,这些数据在表现形式上有电子数据和非电子数据两种。以电子数据方式保存的数据,主要指以电子表格、数据库或文字处理文档等形式保存的数据。非电子数据主要是指那些文字描述的文件、通知、会议纪要等公文。从数据的结构形式看,办公数据有的是以二维表格形式表示的结构化数据,有的是以文字文档处理文件表示的非结构化数据。因此,办公数据源的数据结构是十分复杂的,这给数据仓库的数据抽取和加载增加了很大的难度。有时甚至需要人工处理以后,才能加载到数据仓库中。办公数据在数据仓库中常常用于支持对部门的决策分析。

对于办公数据中的非电子数据的抽取和加载首先要用扫描仪将书面文档转变为电子图像,然后利用可视文字识别软件(OCR)将图像文件转换为文本文件,最后还要创建能够描述和组织文档内部信息的元数据。经过这些处理以后,非电子数据才能加载进数

据仓库。

(4) Web 数据

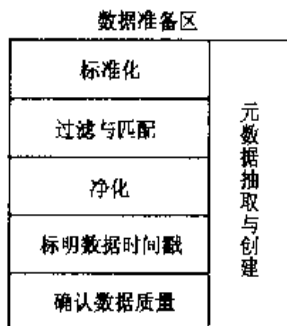
Web 数据是企业通过因特网所获取的数据。这些数据可以通过企业的电子商务系统获取,也可通过网络调查获取。Web 数据大多是 HTML 格式,需要将其转换成数据仓库的统一格式才能加载进数据仓库。

(5) 外部数据

外部数据指那些不为企业所操作、拥有或控制的数据。这些数据有的是电子形式的,例如证券市场的证券数据,或市场咨询部门的研究报告;有的数据是非电子形式例如报刊、政府公告等。这些数据源的使用难度与处理方式与办公数据源相同。

(6) 数据源元数据

数据源数据属于元数据管理层管理范围,在数据仓库中的所有数据都要通过元数据管理层来进行管理和控制。源数据的元数据描述关于源数据的一些说明,包含源数据的来源,源数据的名称、源数据的定义、源数据的创建时间等对源数据进行管理所需要的信息。源数据的来源说明了源数据是从哪个系统、哪个历史数据、哪个办公数据、哪个 Web 页上、哪个外部系统抽取来的。源数据的名称,说明源数据现在和过去的名称。源数据的定义,说明源数据在数据仓库中的作用、用途及数据类型、长度等说明。数据的创建变化时间,需要说明源数据在数据源的创建时间和在数据仓库中的创建时间以及源数据的变化。这些信息主要用于对源数据的管理。



2. 数据准备区功能结构

由于数据仓库的数据来源十分复杂,这些数据在进入数据仓库之前常常需要在数据准备区内进行筛选、清理等标准化处理。因此,数据准备区的功能结构部分由数据标准化处理、数据的过滤与匹配、数据的净化处理、标明数据的时间戳、确认数据质量与元数据抽取和创建等操作组成(参见图 1.7)。

图 1.7 数据准备区功能结构图

(1) 数据的标准化处理

在数据准备区的标准化处理主要是将同名不同内容的、同内容不同名的、同名同内容但不同结构的数据进行标准化处理。例如,在不同的数据源中关于销售地点“北京市”,有的系统用了“北京”,有的用了“北京市”,有的甚至用了“京”等值,但是它们的实际含义都是一致的,为此需要对这些值进行统一处理,以便在数据仓库的使用中不至于产生混乱。

(2) 数据的过滤与匹配

数据的过滤与匹配主要是对进入数据仓库的数据按照用户的需要进行筛选,将用户不需要的数据从数据源中剔除,而留下的数据要能够与数据仓库用户的需求相匹配。

(3) 数据的净化处理

数据的净化处理，主要是对准备加载到数据仓库中的数据进行正确性判断，将那些数据内容错误、格式错误或类型错误的数据进行修正、净化处理。例如，数据仓库中的客户邮政编码是字符类型，但在有的数据源中却以数字类型表示。此时，就需要将其转换为字符类型。

(4) 标明数据的时间戳

由于在数据仓库中要经常进行数据的概括，以分析事务的发展趋势。数据的概括与发展趋势的分析，都需要指明数据的时间属性，因为数据的概括往往是基于时间进行的，而趋势的分析也是以时间为基轴描绘的。因此，在将数据加载到数据仓库之前必须完成数据的时间戳设置，使数据具有时间属性。

(5) 确认数据质量

数据仓库中数据质量的高低是数据仓库能否成功的关键因素之一。例如，在对客户进行邮寄广告促销，由于客户名称的错误，可能惹怒客户，导致客户转向其他供应商，造成客户的流失。有的却会因客户地址的错误，而造成邮寄广告的费用。如此众多的信息应用失败，都是由于数据质量的低劣所造成的。

尽管在数据的标准化处理、数据的过滤与匹配和数据的净化处理过程中，已经对数据源进行了数据质量的提高操作，但是在将数据加载进入数据仓库之前，还需要用各种方法来确认数据的质量。最好的方法是在数据源完成数据质量的确认处理。但是，由于各种各样的原因往往很难在数据源完成数据质量的确认，尤其是那些大量的外部数据源。因此，需要在数据准备区通过手工的方式或软件自动检测的方式，完成数据质量的确认。

(6) 元数据抽取与创建

在数据的求精过程中，还需要从数据源中确定这些源数据的元数据内容，完成元数据的名称与定义及其有关描述，为今后管理数据仓库提供基础。

3. 数据仓库功能结构

数据仓库的功能结构部分由数据重整，数据仓库创建以及元数据管理部分组成（见图1.8）。

数据重整	数据仓库创建	元数据管理
集成与分解	建模	元数据浏览与导航
概括与聚集	概括	
预算与推导	聚集	元数据创建
翻译与格式化	调整与确认	
转换与映像	建立结构化查询	创建词汇表

图 1.8 数据仓库功能结构

(1) 数据重整

为使数据仓库能够更好地为用户服务所进行的一系列预操作,称为数据重整。其中包含数据的集成与分解,数据的概括与聚集,数据的预算与推导,数据的翻译与格式化,数据的转换与映像和元数据创建。

● 数据的集成与分解

对来自不同系统中的数据进行集成,以创建新的数据。有时按照数据处理的需要,在将数据存储到数据仓库过程中,可能要将一个表中的数据分解成数据仓库中的两个或多个数据块。例如,数据在存储到数据仓库时,可能要按照日期、地理位置、生产日期等维度进行分解;有的可能要将不同数据源中的数据,按照数据仓库用户的使用要求集成在一起。

● 数据的概括与聚集

数据仓库在存储数据时,经常按照数据的时间顺序、业务范围、发生地域等进行分割存储,以便于用户的分析和提高数据仓库的使用效率。但是,在实际操作中又经常需要对数据进行概括与聚集处理。数据的概括与聚集处理,就是根据某一属性将数据进行汇总。例如,客户每天的采购就是特定客户在特定一天内的所有采购总和。每周的采购就是客户在某一周内的所有采购总和。数据的概括处理就要依据用户使用数据仓库的需要,预先进行这些数据的汇总与叠加操作,为用户使用数据仓库提供便利。

● 数据的预算与推导

为了提高数据仓库的信息使用者的数据仓库使用效率,在数据仓库中需要事先对信息使用者的常规操作进行预先设置,即无需用户干预就可实现数据的预算与推导。这些预算与推导的结果都是事先进行的计算,并且作为数据仓库的字段存储在数据仓库中。作为数据预算和推导的算法,也应作为数据仓库的元数据进行存储和管理。这样在加载新的数据后,就可以按照这些预算与推导的算法重新进行预算和推导。

● 数据的翻译与格式化

对来自不同数据源的数据进行翻译和格式化,便于今后的统一处理。例如,对客户性别表示,有的系统用“M”表示男,用“F”表示女。而在其他的系统中则用“0”表示男,用“1”表示女。这些数据汇集到数据仓库后,必须按照统一的格式进行翻译,便于系统的数据操作。

● 数据的转换与映像

对存储好的数据进行转移或再映像到数据源中,有利于对新生成或发生变化的数据进行持续更新。

因为数据仓库的数据源中的数据结构基本上是标准关系模式,而数据仓库则大多采用星型模型或雪花模型。这两者的差异必须依靠数据的转换与映像来消除,就是将这两者不同的数据模式以某种方式连接起来,将数据源数据转化为适合数据仓库事实表的行

的过程。这一过程的自动实施,必须依赖于不同模式之间的映像关系。

● 元数据创建

在数据重整过程中,需要从集成数据、概括数据和衍生数据中捕获元数据。确定数据的粒度和分割程度,数据的翻译和转移规则,捕获映射规则及数据源和数据仓库之间的映射关系,这些都是元数据创建的内容。

(2) 数据仓库创建

作为数据仓库的核心功能,数据仓库创建应该完成数据仓库的建模、数据的概括、数据的聚集、数据的调整与确认、建立结构化查询和创建词汇表。

● 数据仓库的建模

从已经创建的数据模型中导出数据仓库的数据模型(星型模型或雪花模型),如果没有数据模型,就要构造新的数据模型。在数据仓库模型的设计过程中,需要完成数据的分割、主题域和粒度的确认,实际数据库的设计模型和数据仓库的物理数据库模式的定型等工作。

● 数据的概括

根据用户的需要对数据进行概括,此处需要从初步的概括数据中创建用户所需的高度概括数据。概括程度与聚集程度有关,例如,每周的概括程度要低于每季度的概况程度。

● 数据的聚集

从拥有大批量数据的数据仓库中进行查询分析,是一个非常费时的操作。例如,在一个有1千个产品和10万个客户的数据仓库中,为执行一个概括性查询,就要涉及1亿个记录,需要较长的时间才能完成。这对经常查询的信息使用者而言是无法接受的。因此,在数据仓库中,常要根据一些典型的查询需求,对数据仓库中数据进行聚集处理。例如,可以对产品的地区分布、品牌的分布进行事先聚集,才能使用户在数据仓库的使用中每次都感受到信息使用时间的一致性。

● 数据的调整与确认

在数据完成概括聚集以后,需要对概括与聚集后的数据进行确认,如果数据概括、聚集的效果不好,还需要进行一些调整,以保证数据仓库的使用效果。

● 建立结构化查询

为了提高一些结构化查询,可以预定义这些查询,且将这些结构化查询作为元数据存储在元数据库中。当用户进行数据仓库的实际查询应用时,只要从元数据库中取回,可以大大提高数据仓库的运行效率。

● 创建词汇表

在创建数据仓库的过程中,需要根据所捕获的元数据建立元数据的词汇表。在词汇表中一般需要包含元数据的名称、别名、简述、创建时间、上次更新时间、关键字、数

据来源、转移/转换信息、概括或推导算法等内容。

由于元数据可能是一个实体集、一个实体、一个属性，因此元数据词汇表的具体格式需要依据元数据类型而定。

4. 数据集市/知识挖掘库结构

数据集市/知识挖掘库的功能结构与数据仓库的功能结构极为相似（见图 1.9），只是数据集市的设立目的在于为某个部门、或某个领域的用户提供服务，而数据仓库的目的则在于为全体用户提供服务。因此，可以将数据集市/知识挖掘库看成数据仓库的一个逻辑上或物理上的子集，数据集市/知识挖掘库中也包含用户所需要查询的详细数据和概括性数据。从数据集市/知识挖掘库所包含的主题与历史数据量看，都比数据仓库少。

求精与重整	数据集市/知识挖掘库创建	元数据管理
过滤与匹配	建立模型	元数据游览与导航
集成与分割	概括	
概括与聚集	聚集	元数据的抽取与创建
预测与推导	调整与确认	
标明时间时间维的数据源	建立结构化查询	创建词汇表

图 1.9 数据集市功能结构

数据集市/知识挖掘库的求精与重整的工作包含将从数据仓库中抽取的数据按照用户的需求进行过滤与匹配操作，将数据仓库集成到以主题域为标志的数据集市/知识挖掘库中，根据数据集市/知识挖掘库用户的要求创建新的概括和聚集，将经过初步概括的数据求精为高度概括的数据，对所有具有时间戳和来源戳的数据预测和导出新的推导数据。数据集市/知识挖掘库在实际应用中是否需要创建，应该根据实际情况考虑。在有的数据仓库应用中，只有数据集市，而没有数据仓库，数据集市/知识挖掘库直接从各种数据源采集数据。此时，数据集市/知识挖掘库的求精与重整功能如同数据仓库一样繁重。有的数据仓库已经完成所有数据集市/知识挖掘库所需要的数据求精与重整工作，可以直接从数据仓库向数据集市传送最终用户所需数据，那么数据集市/知识挖掘库的数据求精与重整过程也可省略。

数据集市/知识挖掘库的创建功能与数据仓库创建功能类似，两者的差异只在数据仓库的创建功能是为满足所有客户的需求，而数据集市/知识挖掘库的创建功能只是为了满足某个部门，某些用户的特定需求而已。

5. 数据仓库的数据存取与使用结构

数据仓库的数据存取与使用结构主要实现数据仓库的最终功能，为数据仓库的最终

用户提供进行决策分析和挖掘知识的功能。为达到这一目的,可将数据仓库的数据存取与使用结构分成数据仓库存取与检索部分,以及数据仓库分析与报告部分两大块(见图1.10)。

数据仓库存取与检索	数据仓库分析与报告	元数据管理
数据仓库直接存取	报表工具	元数据管理与报表
数据集市存取	分析工具	
数据集市重整	分析建模工具	元数据抽取与创建
转换为多维结构	数据挖掘工具	
创建局部存储	新产品应用程序	

图 1.10 数据仓库数据存取与使用结构

数据仓库存取与检索部分为用户提供访问数据仓库或数据集市的功能,利用这些功能可将用户检索的数据转换成多维数据并且存入多维数据库。可将数据仓库或数据集市中的数据“卸载”下来,成为局部存储数据,便于用户进行局部分析、数据挖掘、翻译转换等处理。这就需要解决如何从预定义的查询到即席的查询、迭代的查询、细剖查询的实现。

为了用户使用方便,在这里还提供管理与使用数据仓库元数据功能。这些功能可以帮助用户了解数据仓库或数据集市的名称、描述说明、数值、价值来源以及版本等内容,了解数据的名称、数值等内容和数据从抽取到存入数据仓库或数据集市的转移过程,了解数据的定位和数据的可靠性,以及如何存取和使用数据。利用这些功能可以帮助用户掌握数据的正确内容、信息的粒度、信息的概括程度、原始数据的来源和日期,并且可以按照其上下文来查看数据,将数据转化为信息。还可以验证数据源的质量,在数据抽取和存储过程中用于判断数据的可靠性和质量。

数据仓库分析与报告为最终用户使用数据仓库提供一组工具,可使用户依靠数据仓库或数据集市进行决策分析或知识挖掘。这些工具包括报表处理工具,分析与决策支持工具,业务建模与分析处理工具,数据挖掘工具等。

具体地说,这些工具有地理信息系统(GIS)、数据采集工具、联机分析处理(OLAP)、可视化工具、经理信息系统(EIS)、统计工具、因特网浏览器、元数据浏览器、第四代语言、图形用户界面建立程序、电子表格、报表生成器和数据访问工具。

地理信息系统可以利用数据仓库中的数据图示化地表达数据关系。例如,通过GIS了解生活在某一特定销售点范围之内的客户数量,或在两个销售点之间的平均到达时间。利用GIS还可确定对公司产品感兴趣的潜在客户居住区域,帮助企业确定新的销售点位

置。

数据采集工具和统计工具就是数据仓库中的数据挖掘工具。利用这些工具，可用图形或表格方式显示隐藏在大量数据背后的商业规律。例如，哪些客户可能会在信用上发生问题，哪些客户可能会对企业的促销手段作出积极的反应。

联机分析处理（OLAP）和经理信息系统（EIS）能够以便利的手段让客户完成复杂的数据查询，且能以形象化的图形、图像和表格的方式给出决策分析的结果。

因特网浏览器主要为用户的 Web 数据仓库使用提供便利。

电子表格作为办公处理软件，许多企业已经拥有，无须另行购买安装；而且许多用户能够熟练地操作，具有一定的电子表格分析数据经验。因此，电子表格也可以作为数据仓库的分析工具加以应用。但是，将数据仓库中的数据转入电子表格还需费一番周折。

可视化工具、元数据浏览器、第四代语言、图形用户界面（GUI）建立程序、报表生成器、数据访问工具等都可作为数据仓库的访问分析工具使用，只是在实际应用中各有千秋。例如，OLAP 可以提供极强的数据查询，但是报表的生成能力就不如报表生成器；第四代语言与 GUI 建立工具可以提供有限制的查询界面，且能指导用户完成查询。这对数据仓库的安全使用与指导新用户使用数据仓库十分有利，但不利于有经验的用户对数据仓库的知识挖掘。此外，根据需要也可以用第四代语言建立一个 OLAP 工具。

1.3.3 数据仓库的管理层

数据仓库的运行除依靠上面所介绍的数据仓库基本功能以外，还需要有对这些基本功能进行管理与支持的结构框架，这样数据仓库才能正常运行与使用。数据仓库管理层由数据仓库的数据管理和数据仓库的元数据管理组成，数据仓库的环境支持层由数据仓库数据传输层和数据仓库基础层组成。

数据仓库的数据管理层（见图 1.11）包含数据抽取、新数据需求与查询管理，数据加载、存储、刷新和更新系统，安全性与用户授权管理系统以及数据归档、恢复及净化系统部分。

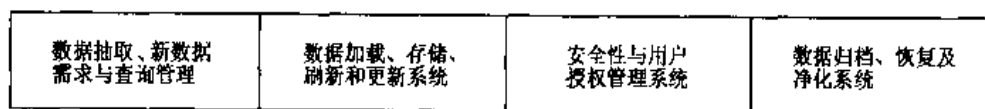


图 1.11 数据仓库的数据管理层

数据管理层中的数据抽取、新数据需求与查询管理主要负责完成从数据源中抽取数据的管理；客户在数据仓库应用中出现对新数据的要求时，从新的数据源或当前数据源中按照用户需求追踪和充实新数据；以及对数据查询中的并行处理工作的管理。

数据仓库中的数据加载、存储、刷新和更新系统则负责对从数据源中所抽取的数据在完成筛选、净化处理以后,将这些数据加载、存储到数据仓库中;捕获数据源中的数据变化,用最新数据充实数据仓库;根据用户的需求和数据仓库管理的要求对数据仓库进行更新等工作。

安全性与用户授权管理系统主要负责数据仓库的安全管理工作,禁止用户对数据仓库进行某些非法操作;根据用户的管理权限和工作需要给予用户对数据仓库的不同操作权限。

数据仓库的数据归档、恢复及净化系统主要负责定期对数据仓库中的数据进行归档、备份,以便在数据仓库遭到破坏时可以恢复,而净化系统则负责对从数据源所抽取的数据进行数据的筛选、数据标准的统一、数据内容的统一等各种求精与重整净化工作的管理。

1.3.4 数据仓库的元数据管理层

数据仓库的有效性完全建立在数据的定义(元数据)之上。元数据已经渗透到数据仓库的各种活动中,数据源的性质由所获取数据的定义来刻画,增加时间戳就需要有与元数据相关的时间信息,元数据还要为数据仓库的数据操作提供索引。

数据仓库的元数据管理层(见图1.12)负责管理数据仓库所使用的元数据,其中包括数据仓库、数据集市和词汇表管理,元数据抽取、创建、存储和更新管理,预定义的查询和报表以及索引管理,刷新与复制管理,登录、归档、恢复与净化管理。

数据仓库、数据集市 和词汇表管理	元数据抽取、创建、 存储和更新管理	预定义的查询、 报表和索引管理	刷新与复制管理	登录、归档、恢复 与净化管理
---------------------	----------------------	--------------------	---------	-------------------

图 1.12 数据仓库的元数据管理层

1. 数据仓库、数据集市和元数据词汇表管理

元数据管理层利用元数据词汇表,管理数据仓库和数据集市中逻辑数据模型和物理数据模型,以及与技术 and 业务相关的数据说明。

2. 元数据抽取、创建、存储和更新管理

元数据在数据仓库对数据源进行数据抽取、清理、加载等操作过程中,需要对所涉及的元数据进行抽取、创建、存储和更新处理。即从数据源中将关于这些数据的说明抽取出来,如果在元数据库中这些元数据不存在,就要筹建且存储在元数据库中;如果这些元数据已经存在于元数据库中,则要根据最新情况进行更新。

3. 预定义的查询、报表和索引管理

在元数据管理中还需要对设计人员为数据仓库用户预定义的查询和报表进行管理，将预定义的查询和报表处理方式甚至处理结果置于元数据库中；当用户需要进行相同的预定义查询和报表时，就可以提供相应的结果。预定义的查询和报表处理方式也需要存储在元数据中。元数据管理层还需要实现大型数据仓库的多级索引、数据压缩、复合键和数据版本等方面的管理。

4. 刷新、复制、恢复、登录、归档与净化管理

当数据仓库所连接的数据源发生变化时，数据仓库的内容也要定期刷新。这些刷新工作需要依靠元数据库中所包含的有关说明。为保证数据仓库的安全，需要经常定期进行复制。在数据仓库遇到破坏后，可从备份中将数据仓库恢复。数据仓库的备份与恢复工作也有赖于元数据的帮助。用户在使用数据仓库时需要进行身份的验证，对用户的登录管理也离不开元数据的支持。源数据在加载进数据仓库之前必须进行净化处理，而净化处理的规则亦需要元数据说明。

1.3.5 数据仓库的环境支持层

1. 数据仓库的数据传输层

数据仓库中不同结构之间的数据传输，需由数据仓库的传输层完成。数据传输层包含数据传输和传送网络、客户/服务器代理和中间件、复制系统，以及数据传输的安全保障系统（见图 1.13）。

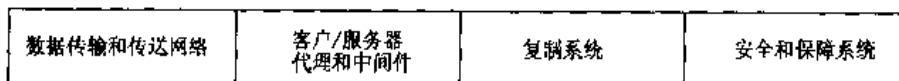


图 1.13 数据仓库的数据传输层

（1）数据传输层的组成

在数据传输层中的数据传输和传送网络包含如下一些系统。

- 网络协议。例如，TCP/IP，SNA/APPN，IPX。
- 网络管理框架。例如，HP 公司的 Open View，IBM 公司的 Net View，SunSoft 公司 SunNet Manager 等。
- 网络操作系统。Windows NT，Windows 2000，Linux。理想的操作系统性能应该具有广泛性，这样才能支持适合典型数据仓库环境中的各种应用程序。对于数据仓库，64 位操作系统是比较合适的。因为 64 位操作系统一般可以提供寻址更多文件、更大型文件、

更大的进程数据空间、更多的共享库段和更多的随机访问内存的能力，能够使数据仓库获得更高的性能并且减少内存和外存的交换。

从数据仓库的角度看，网络操作系统的性能应该支持内核线程、高达 4TB 的内存、最大为 1TB 的特大型文件系统、应用程序所用页面大小可变以及并行处理。且有日志文件系统、内存分页管理功能、动态加载核心模块功能，可为数据仓库提供良好的可恢复性。操作系统应该支持开放系统标准，支持系统的互操作，这才能使数据仓库在多操作系统环境中运行。

- 网络类型。数据仓库中的网络问题在于带宽，在数据仓库的网络配置中可将用户和系统数据分隔到不同的网络中，以增加系统的整体带宽；系统数据流量可以通过 100Base-TX 以太网、FDDI，ATM，千兆位或 HIPPI 接口；用户数据流量则放在 10/100Base-TX 以太局域网。

(2) 客户/服务器与中间件

在客户/服务器代理和中间件部分包含如下一些系统。

- 数据库网关。数据库网关便于将数据仓库联结到其他软件产品上。例如，Information Buidler 公司的 EDA/SQL，Sybase 公司的 MDI，IBM 公司的 DRDA/DDCS。

- 数据仓库的中间件。数据仓库的中间件一般用于补充数据仓库中其他组件功能的不足。例如，Pine Cone System 公司开发的 Usage Tracker 软件可以用于监视数据库与查询管理程序之间的 TCP/IP 包，可以提供关于数据仓库用户、被访问数据库以及访问时间等信息。利用这些信息可对数据仓库的结构进行调整，提高数据仓库的性能。目前许多数据库管理系统开始将各种中间件的功能添加到数据库管理系统中，因此，在选择中间件之前需要了解中间件的功能是否已经在数据库管理系统中存在。

- 对象请求代理。IBM SOM，HP 公司的 ORB Plus，DEC 公司的 Object Broker。

- 传输层的数据仓库数据发布和复制系统，主要用于将数据源中的源数据库数据拷贝到数据仓库的目标数据库上，或将数据仓库中的源数据库数据拷贝到数据集市的目标数据库上。源数据库和目标数据库可以在同一台机器上，也可以不在同一台机器上。数据的复制可以根据指定的时间进行数据发送，还可以在数据发送过程中对发送数据进行修改，然后发送到目标数据库上。

(3) 数据复制系统

在传输层的复制系统中有如下一些系统。

- 发布与复制系统。IBM 的关系型数据传播器 DpropR 和非关系型的 DpropNR，Sybase 的 Replication Server，Oracle 的 Symmetric Replication。

- 数据库网关内定义的复制工具。Information Builder 公司的 EDA/SQL。

- 数据仓库产品。Prism Solution 公司的 Warehouse 和 Change Manager。

2. 数据仓库的基础层

在数据仓库的基础层中包含系统管理、工作流程管理、存储系统和处理系统等部分(见图 1.14)。

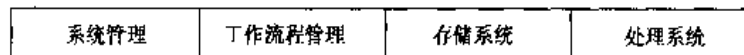


图 1.14 数据仓库的基础层

数据仓库的系统管理部分为数据仓库的设计者和最终用户提供执行、管理、终止工具和应用程序等功能。

工作流程管理部分主要支持处理集成和管理,以协调各种工具、应用程序和操作有条不紊地进行,正确完成对数据仓库和数据集市的抽取、刷新、复制、更新、聚集、概括以及其他维护任务和系统管理任务。利用工作流程的管理实现对数据仓库和数据集市的自动维护与刷新,且可提供预定义的报表和查询结果,以提高系统的设计者和最终用户的工作效率。

基础结构层中的存储系统为数据源、数据仓库、数据集市中的数据库目录提供数据库和文件管理器,为数据仓库的存取与使用提供多维的和本地的存储。

基础结构层中的处理系统实际上是数据仓库核心的基本操作环境,即数据源、数据仓库、数据集市、数据仓库存取与使用、中间件的操作环境。

对于数据仓库的基础结构层还需要考虑配置管理程序,存储管理程序,安全性管理程序,软件分布管理程序,特许证管理程序,性能监控程序和容量分析程序等。

1.4 数据仓库技术

尽管在许多情况下,数据仓库的创建与使用技术并不比数据库创建使用的技术复杂,但是数据仓库的创建与使用技术也有许多特定要求,只有了解这些技术及其特点,才能更好地创建与使用数据仓库。目前用于数据仓库的主要技术包含数据管理技术,数据存储技术和数据仓库接口技术。

1. 数据管理技术

在数据管理技术中包含大批量数据管理技术,数据仓库索引与数据监视技术,元数据管理技术,数据压缩技术和复合键码技术。

(1) 大批量数据管理

在数据仓库的所有技术中最重要的是管理大批量数据技术。如果不能管理大批量数据,那么数据仓库的创建与使用是不可能的。管理大批量数据包括管理大批量数据能力

和管理好大批量数据的能力,即管理大批量数据技术要求管理能力的满足和管理的高效率两方面要求。一般数据仓库对大批量数据的管理可以通过对文件的寻址、索引,数据的外延,有效的溢出管理等技术来实现。

(2) 数据仓库的高效索引与数据监视技术

数据仓库要对大批量数据进行各种不可预知的查询或访问,须用双重粒度级和数据分割等技术实现对数据仓库的灵活访问。这些技术必须依靠二级索引、稀疏索引、动态索引和临时索引等索引技术的支持。尽管有了这些索引技术,还要利用各种高效的访问索引技术才能使这些索引发挥作用。例如,在数据仓库中可以采用位映像、多级索引、索引项的压缩、创建选择索引范围或将索引全部装入内存等索引使用技术。这样在高效的索引技术支持下,才可实现数据仓库的有效应用。

为了保证数据仓库的正确运行,还需要对数据仓库中的数据进行有效的监视。这一技术在数据仓库应用技术中是不可缺少的。

(3) 元数据管理技术

数据仓库的元数据管理技术比在数据库管理中更为重要,因为,数据仓库的开发是在一种启发式的、反复的开发周期上进行的。数据仓库的用户只有通过对元数据的正确、实时访问,才能更有效地使用数据仓库。如果缺乏元数据管理技术支持,很难想像数据仓库的用户如何去使用数据仓库。

(4) 数据压缩技术

数据仓库的大批量数据存储,在许多情况下还需要压缩技术的支持。由于数据仓库中的数据很少发生变化,因此压缩技术比较适合数据仓库。尽管数据压缩技术需要消耗CPU资源,但是可以减少I/O操作,这就可使数据仓库的效率得到极大的提高。

(5) 复合键码技术

由于数据仓库中的数据具有随时间等维度变化的特征,因此,数据仓库的复合键码管理技术是必不可少的。只有依赖复合键码管理技术的支持,才能确定某个客户在哪天采购了哪些商品。

2. 数据存储技术

数据的存储技术包含多介质存储设备的管理技术,数据存储的控制技术,数据的并行存储与管理技术,可变长技术和锁切换技术。

(1) 多介质存储设备的管理技术

在数据仓库中管理大批量数据时,为了达到效率和费用的平衡,通常要求数据仓库能够实现多存储介质设备的管理,即能够对磁盘、磁带等各种介质的存储设备实现有效的管理。这一技术对海量数据仓库的有效管理是必需的。

(2) 数据存储的控制

数据仓库的高效运行技术之一在于数据仓库设计者可以在物理块或物理页上对数据存储进行有效的控制。数据仓库设计者可以利用这一技术对数据存储的物理地址进行调整,使其更加适合数据仓库的使用要求。

(3) 数据的并行存储与管理

考虑数据仓库的多用户环境,数据仓库应该具备数据的并行存储与管理技术。这样才能使数据仓库的性能得到极大的提高。

(4) 可变长技术

在数据仓库中如果经常发生变长数据的变动,将严重影响数据仓库的系统性能。因此,在数据仓库中需要有效管理变长数据的技术。

(5) 锁切换技术

由于数据仓库中的数据很少发生变化,因此,长期在加锁管理下进行数据仓库的操作将降低数据仓库的效率。为使数据仓库高效率地运行,要为数据仓库提供锁切换管理技术,使用户可在需要的时候进行加锁操作或开锁操作。

3. 数据仓库接口技术

数据仓库的接口技术包含多技术接口技术,语言接口技术和数据的高效率加载技术。

(1) 多技术的接口

对于数据仓库的创建和运行来说,能够使用各种不同的技术获取或传送数据是很重要的。因为面对多种多样的数据源,数据仓库不可能只用一种技术来完成数据的抽取与传送。这就需要数据仓库具有支持各种技术的接口,如果这种接口技术能够在批处理方式下运行,将极大地提高数据仓库的运行效率。

(2) 语言的接口

数据仓库的实际应用必须依赖某种语言来完成,这种语言接口能够一次访问一组数据或者一条记录,并且支持一个或多个索引,能够使用 SQL 语言,能够插入、删除或更新数据。即数据仓库的语言接口必须健壮,能够容易地进入数据仓库的接口并且访问数据。

(3) 数据的高效率加载

在数据仓库的实际应用中,经常需要从数据源加载数据。如果数据的加载是低效率的,那么数据仓库的使用性能不可能提高。不能想像数据仓库的加载需要 24 小时才能完成,同时用户要求每天加载一次。那数据仓库每天只能用于进行数据的加载,而根本无法提供决策分析了。



本章小结

数据仓库是近年来在信息管理领域得到迅速发展的一种面向主题的、集成的、随时间变化的、非易失性数据的集合，其目的在于支持管理层的决策。数据仓库已经成为有竞争力企业的基础。对数据仓库的使用主要涉及信息的使用者和知识的挖掘者两大类用户，这两类用户对数据仓库的使用与要求各有差异。数据仓库通常主要包含数据仓库数据库、数据集市/知识挖掘库、数据源、数据准备区，以及各种管理工具和服务工具。数据仓库创建以后，先要从数据源中抽取所需的数据到数据准备区，在数据准备区中经过数据的净化处理，再加载到数据仓库数据库中，最后根据用户的需求将数据发布到数据集中。当用户使用数据仓库时，可以通过 OLAP、数据挖掘等数据仓库应用工具，向数据集市/知识挖掘库或数据仓库进行决策查询分析或知识挖掘。在数据仓库的创建和应用中，可以利用各种数据仓库管理工具辅助完成。

本章还对数据仓库的总体结构框架和层次环境进行了介绍，其目的在于使读者对数据仓库有个整体的了解。在数据仓库的实际创建与应用中，可以根据实际情况进行增删。因为数据仓库是一个系统的体系结构，而不是软件产品或应用程序。



习题

- 1-1 比较数据仓库与数据库的相同点与不同点。
- 1-2 为什么不能依靠传统的业务处理系统进行决策分析？
- 1-3 在将数据源中的数据加载到数据仓库之前需要完成哪些工作？为什么要进行这些工作？
- 1-4 数据仓库的信息使用者与知识挖掘者对数据仓库的应用有何不同？

1-5 假设有人要求你创建一个数据仓库，主要是分析关于客户的人口统计（收入、家庭人口、家庭位置和爱好等）。数据仓库的目的在于将特定的产品推销给合适的潜在客户群。这个数据仓库应该从哪些地方获取数据源？数据仓库的体系结构应该包含哪些部分？

DB2提供约束作为使用数据库系统来实施这些规则的一种方式。如果想通过使用数据库系统来实施这些商业规则，那就不必在应用程序中编写代码来实施这些规则。然而，如果某个商业规则只适用于某一个应用程序，则应该在这个应用程序中编写此规则，而不应使用全局数据库约束。

DB2提供下列类型的约束：

- NOT NULL约束
- UNIQUE约束。
- PRIMARY KEY约束。
- FOREIGN KEY约束。
- CHECK约束。

可使用SQL语句CREATE TABLE和ALTER TABLE来定义约束。

3.2.2 用户定义类型和大对象

数据库中的每个数据元素都存储在表的列中，并且每列都定义有一个数据类型。数据类型对可放入列中的值的类型进行限制，并限制对这些值可执行的操作。例如，一个整型列只能包含一个固定范围内的数。DB2包括一组具有定义的特性和行为的内部数据类型：字符串、数字、日期时间值、大对象、空值、图形字符串、二进制字符串和数据链路。

但是，有时内部数据类型可能无法满足应用程序的需要。为此，DB2提供了用户定义类型(UDT)，使得能够定义应用程序需要的不同数据类型。

UDT基于内部数据类型。定义UDT时，也应定义对该UDT有效的操作。例如，可以定义基于DECIMAL数据类型的MONEY数据类型。然而，对于MONEY数据类型，可能只允许加法和减法运算，而不允许乘法和除法运算。

大对象(LOB)允许您存储并处理数据库中大而复杂的数据对象：如音频、视频、图像和大文档。

UDT和LOB相结合会使您的工作能力大大提高。使您不再限于使用由DB2提供的内部数据类型来建立商业数据模型和捕捉该数据的语义，可以使用UDT为高级应用程序定义大而复杂的数据结构。

除了扩充内部数据类型之外，UDT还有以下几个好处：

1) 支持应用程序中面向对象的程序设计。可以将类似对象分组为相关的数据类型。这些类型具有名称、内部表示和特定的行为。通过使用UDT，可以让DB2知道新类型的名称以及该类型在内部如何表示。LOB是新类型可能的内部表示法之一，并且最适合表示大而复杂的数据结构。

2) 通过强类型转换和封装保证数据完整性。强类型转换保证：只有定义在特殊类型上的函数和运算才可应用于该类型。封装确保UDT的行为受可应用于这些行为的函数和运算符的限制。在DB2中，可以以用户定义函数(UDF)的形式提供UDT的行为；为满足广泛的用户需求，可编写UDF。



第 2 章

Oracle 的数据仓库 设计与使用

引 言

Oracle 作为一种大型关系数据库在联机事务处理中得到广泛的应用。随着数据仓库应用范围的扩大, Oracle 公司在 20 世纪的 90 年代开始提供数据仓库产品。在 2001 年, Oracle 公司推出了 Oracle 9i。Oracle 9i 由数据库、应用服务器、开发工具包组成, 数据仓库的创建和管理功能是其中的重要组成。随着 Oracle 9i 应用的逐步扩大, 许多大型企业在数据仓库的开发中选用了 Oracle 9i。

通过本章学习, 可以了解或掌握:

- ◆ Oracle 数据仓库的开发工具种类
- ◆ Oracle 数据仓库数据库创建工具使用
- ◆ Oracle 数据仓库数据库的表空间创建工具使用
- ◆ Oracle 数据仓库数据库的表创建工具使用
- ◆ Oracle 数据仓库数据库的维创建工具使用
- ◆ Oracle 数据仓库数据库的立方创建工具使用
- ◆ Oracle 数据仓库数据库的数据挖掘工具种类

2.1 Oracle 数据仓库开发工具简介

Oracle 9i 数据库内置高级 OLAP、数据挖掘和数据仓库功能，使用户在建立商业智能应用时无需再像过去那样，先从数据仓库中采集数据，然后才能在专门的分析服务器中进行处理；能够以更简单的技术、更少的投资，实现准确、及时的智能化信息管理。

Oracle 的数据仓库产品可划分为：技术基础工具、分析应用工具、数据仓库创建工具和数据仓库维护工具 4 类。

2.1.1 Oracle 数据仓库的技术基础工具

Oracle 数据仓库的工具可以实施数据仓库和/或数据集市解决方案、简化管理、提供组织范围的数据访问和使用，且在前台实现信息的智能处理。Oracle 9i 技术基础类产品可以分为以下 4 种。

1. Oracle Warehouse Builder

Oracle Warehouse Builder 为企业数据仓库解决方案的设计、实施和管理，提供一个完善的、集成的框架。

2. Oracle 9i 数据库

Oracle 9i 数据库用于数据仓库数据库的创建与使用，提供较好的数据存储性能，能够较好地完成数据仓库的创建工作。

3. Oracle 数据集市套件

Oracle 数据集市套件在一个软件包中提供构建数据集市所需的一些软件，例如数据集市设计工具、从运行系统中提取数据的图形工具、用于最终用户的查询和分析工具——Discoverer、基于 WWW 的服务器的可与企业内部网连接，对数据集市进行访问的工具。

4. Common Warehouse Metadata (CWM)

CWM 主要用于构建、维护、管理和使用数据仓库，包括技术和商业元数据、对数据进行管理和分析的工具，以及元数据仓库之间仓库元数据的交换。

2.1.2 Oracle 数据仓库的分析应用工具

Oracle 数据仓库开发工具中还包括可以满足企业需求的分析应用工具，主要可以满足面向高层、面向底层、面向 Oracle 应用客户，以及用于平衡高层和低层的分析应用。

1. 面向高层的分析工具

面向高层发展的分析应用工具有 Oracle Front Office 和 Oracle Sales Analyzer。

Oracle Front Office 工具提供若干种应用，主要用于管理客户关系的全面产品，其范围覆盖从市场营销到销售与服务。

Oracle Sales Analyzer 工具主要用于分析各种销售和营销数据。数据的来源可以包括内部订单输入和发运系统以及由第三方提供的数据。Oracle Front Office 与 Oracle Sales Analyzer 相结合后，可以提供有关销售情况的完整情况，从销售效果到销售环境，以至定义新的产品和市场类别。Oracle 的高层应用工具采用图形用户接口，更易于用户的使用，并且能够支持移动操作。

2. 面向底层的分析应用

面向底层的分析应用主要包括 Analyzer Activa, Oracle Financial Analyzer 和 Financial Analyzer 工具。

Analyzer Activa 工具是个覆盖全面、基于活动的成本计算和管理软件包，具有实现动态成本计算与管理的能力，并且能够与客户机/服务器技术和早期系统集成。Activa 还允许用户从更细微的角度——客户、产品和分销渠道来衡量收益率。

Oracle Financial Analyzer 工具包含财务分析、规划、预算和报告功能，能够满足用户的底层需求。

Financial Analyzer 工具则可通过直接链接源系统（例如账务系统），自动创建 OLAP 系统，以确保数据仓库应用中的数据一致性。

3. 用于平衡高层和底层发展的分析应用

用于平衡高层和底层发展的分析应用工具是 Balanced Scorecard。该工具为四个主要应用领域进行信息分析提供框架，这四个领域分别为财务、客户、内部业务和学习/发展。通过这些领域观察企业，管理人员就可确定哪项工作是公司战略获得成功的关键。

4. 面向 Oracle 应用客户的分析应用工具

面向 Oracle 应用客户的分析应用工具是 Oracle 商业信息系统（OBIS）。该系统提供

一种性能框架，能使用户设定希望跟踪的主要性能指标（KIP），并且围绕这些性能指标定义误差级别。OBIS 主要由事实管理、目标管理和异常管理组成。

2.1.3 Oracle 数据仓库创建工具

Oracle 的数据仓库创建工具用于数据仓库创建、表空间创建和表的创建。创建数据仓库的工具主要有 Oracle 数据库构造助手（Oracle Database Configuration Assistant），而创建表空间和事实表与维表则可使用 Oracle 企业管理器（Oracle Enterprise Manager）。

2.1.4 Oracle 数据仓库维护工具

Oracle 数据仓库的维护工具主要是指对数据仓库进行数据装载、清理等操作的工具。例如，在企业管理器中运行导入、导出和装载操作，完成数据仓库的数据装载；有在 NT 环境下的 Oracle 数据集市工具集，将数据移入数据仓库的基于引擎的工具和代码生成工具，Oracle 的透明网关技术。

2.2 Oracle 数据仓库创建

数据仓库的创建基础是构建数据仓库的事实表与维表。在 Oracle 中创建数据仓库，通常经历创建数据仓库、创建数据库表空间和创建数据库表这几个阶段。

2.2.1 Oracle 数据仓库的创建

1. Oracle 数据库构造助手启动

Oracle 数据仓库的创建，可以采用 Oracle 数据库构造助手（Oracle Database Configuration Assistant）进行。

Oracle Database Configuration Assistant 的启动顺序为“开始”→“程序”→“Oracle-OraHome90”→“Configuration and Migration Tools”→“Oracle Database Configuration Assistant”（参见图 2.1）。进入 Oracle Database Configuration Assistant 的欢迎使用对话框。在欢迎使用对话框中单击“下一步”按钮，进入 Oracle Database Configuration Assistant 的操作选择对话框（参见图 2.2）。

2. 数据仓库创建

在 Oracle Database Configuration Assistant 的操作选择对话框中可以选择“创建数据库”选项，以创建新的数据库或数据库模板。选择“管理模板”选项，则可使用 3 种方

法创建模板：从现有模板创建、从现有数据库(仅限结构)创建、从现有数据库(结构及数据)创建。这里选择了“创建数据库”选项，然后单击“下一步”按钮，进入数据库模板选择对话框（参见图 2.3）。

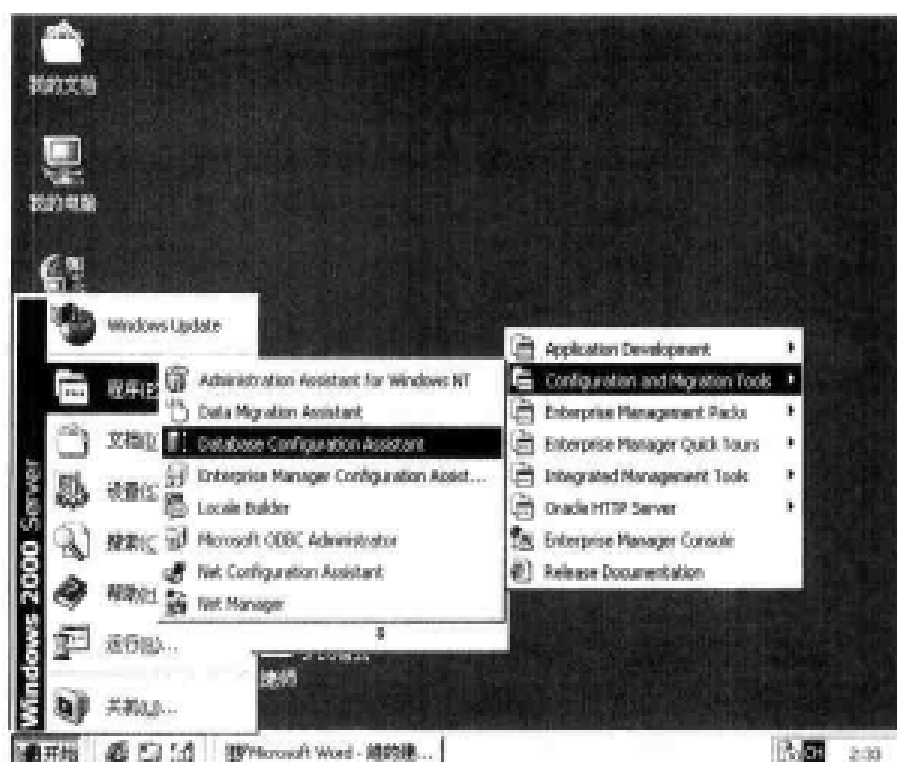


图 2.1 “Oracle Database Configuration Assistant” 的启动



图 2.2 Oracle Database Configuration Assistant 操作选择对话框

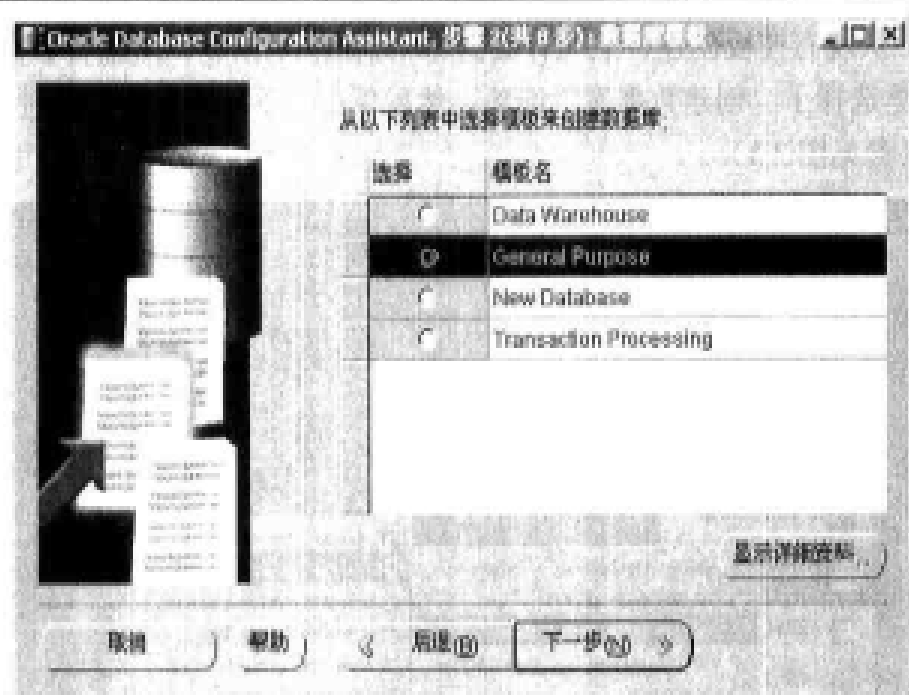


图 2.3 数据库模板选择对话框

在数据库模板选择对话框中根据需要选择“Data Warehouse”，“General Purpose”，“New Database”和“Transaction Processing”4个不同的单选项，分别创建数据仓库、一般目的数据库、新数据库和联机事务处理数据库。根据选择结果，Oracle 可以提供适合不同环境的数据库类型。这里选择了“Data Warehouse”选项创建数据仓库。单击“下一步”按钮，进入“数据库标识”对话框（参见图 2.4）。

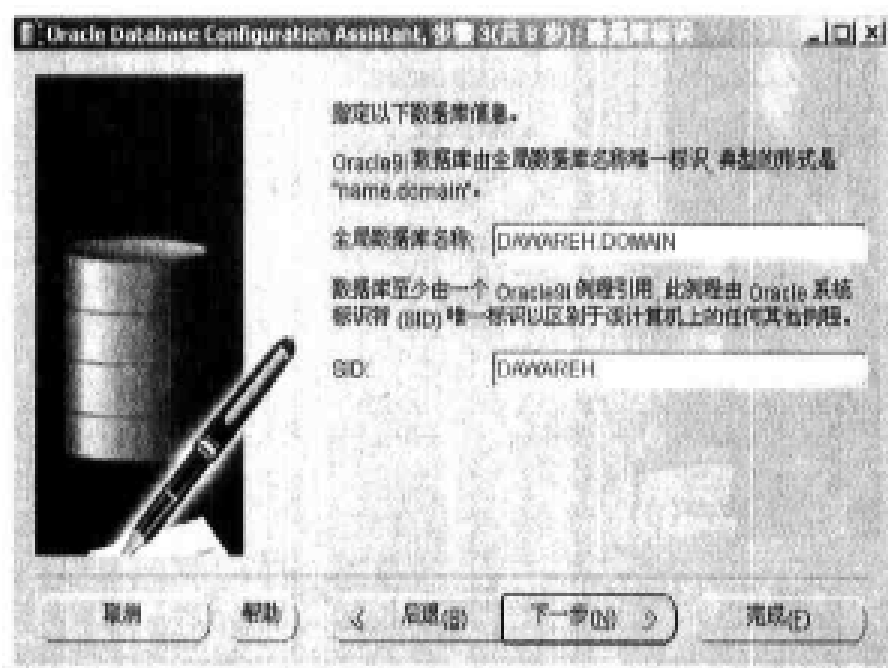


图 2.4 “数据库标识”对话框

在“数据库标识”对话框中, 用户需要确定全局数据库名和 SID。全局数据库名是将数据库与任何其他数据库惟一标识出来的数据库全称。对于任何数据库, 至少有一个引用数据库的例程。每个数据库例程对应一个 SID 和一系列数据库文件。在创建 SID 时, 会同时创建数据库例程及其数据库文件(初始化参数文件、控制文件、重做日志文件和数据文件等)。确定数据库标识后, 单击“下一步”按钮, 进入“数据库连接选项”对话框(见图 2.5)。

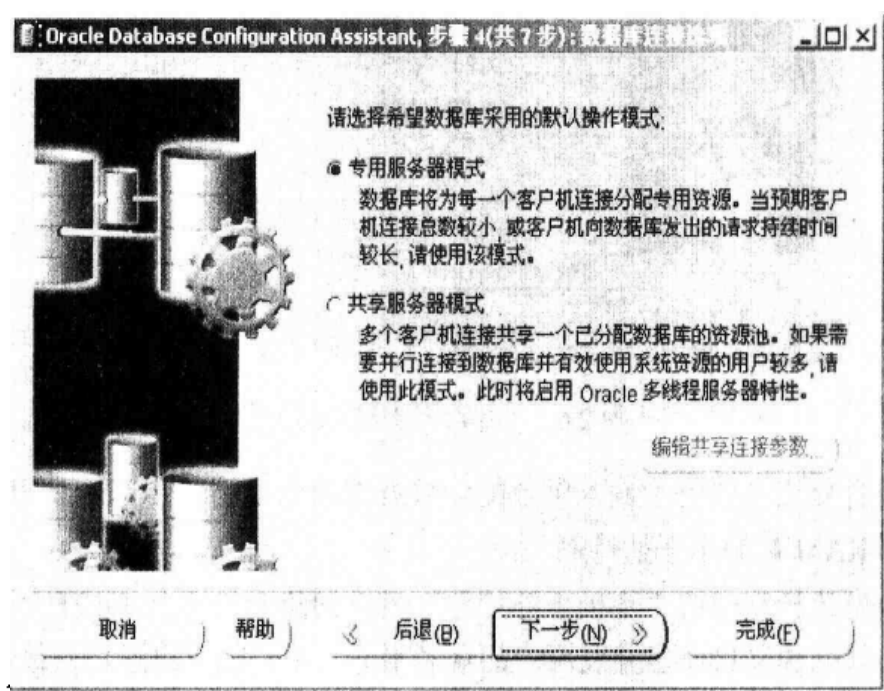


图 2.5 “数据库连接选项选择”对话框

在“数据库连接选项”对话框中, 可以选择“专用服务器模式”与“共享服务器模式”选项。如果在“数据仓库”环境中使用数据库, 或只有少数客户机连接数据库, 且客户机对数据库发出持久的、长时间运行的请求, 就要选择“专用服务器模式”。在联机事务处理(OLTP)环境中, 或大量用户需要连接到数据库, 并且需要有效地使用可用系统资源且内存有限制时, 则选择“共享服务器模式”。这里选择了“专用服务器模式”, 然后单击“下一步”按钮, 进入“初始化参数”对话框(见图 2.6)。

3. 数据仓库数据库初始参数设置

在“初始化参数”对话框中有“内存”、“归档”、“数据库大小”和“文件位置”4 个标签页需要设置。

在内存标签页上可以选择“典型”和“自定义”选项。典型选项对于大多数环境和不熟悉高级数据库创建过程的用户来说较为合适。确定“典型”选项后, 再确定最大并发连接用户数, 在这里输入任意给定时间可能同时连接到数据库的大概用户数。用于 Oracle

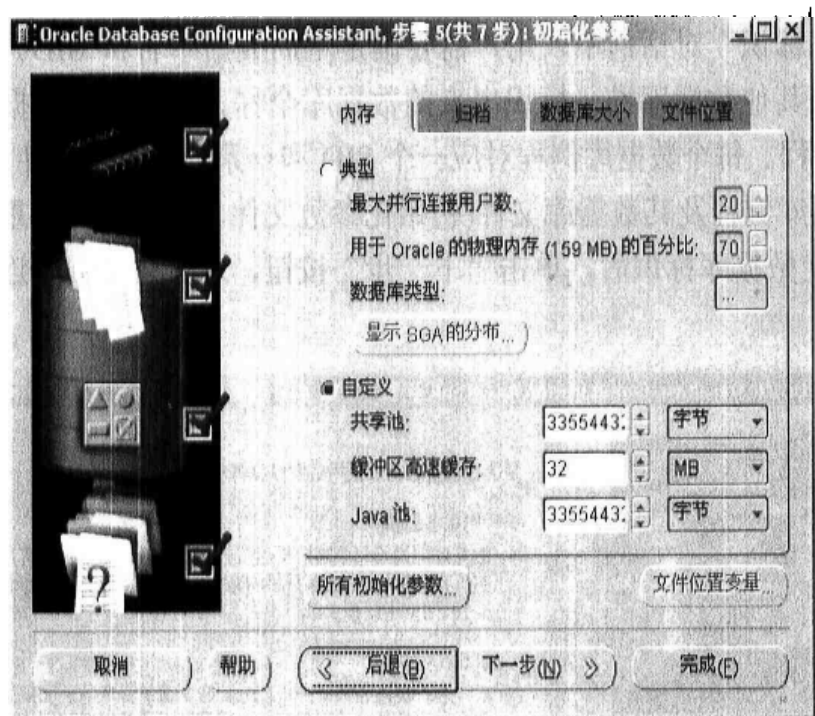


图 2.6 “初始化参数”对话框

的物理内存的百分比主要用于输入可分配给数据库的全部物理 RAM 的百分比，计算机上现有的全部 RAM 数显示在括号中。

数据库类型选择项中的“联机事务处理”可以创建一种适合 OLTP 环境的数据库环境，即每天必须处理来自许多并发用户的众多事务。用户能够快速访问最新数据。因此，数据库的性能取决于数据吞吐量(速度)和数据可用性。而“数据仓库”选择项，则可创建一种适合数据仓库环境的数据库环境，即可以处理各种各样的查询(通常是只读查询，如 SELECT 语句)，包括从几条记录的简单读取到从许多不同的表中查询数千条记录的大量复杂查询。因此，数据库性能取决于响应时间。而“多用途”选项，则可以创建适用于混合工作负荷环境(OLTP 和数据仓库)的数据库。

“自定义”选项可用自定义方式创建数据库。此时，就要确定共享池大小，以确定输入共享的 SQL 和 PL/SQL 语句的区域大小(字节)，较大的值可以提供较好的处理性能。高速缓存中的每个缓冲区就是一个 Oracle 数据块的大小(由初始化参数 DB_BLOCK_SIZE 指定)，高速缓存中的每个数据库缓冲区都可保留一个从数据文件读取的单一数据块。Java 池内存用于 JVM 中所有特定会话的 Java 代码和数据的服务器内存。

在“归档”页中的“日志模式”选项中，可将数据库置为归档日志模式，且使填满的重做日志文件在再次使用之前归档。这样，可使数据库从例程和磁盘故障中完全恢复。如果选定此项，则必须启用“自动归档”并且提供下列所述域的信息。自动归档例程可以配置为具有附加的后台进程，即归档程序(ARC0)，它在重做日志文件组变为非活动状态后自动将其归档。日志归档文件名格式输入归档日志文件的格式或接受默认输入项。

归档日志目标输入框,输入存放脱机重做日志文件的目录,或者采用默认输入项。

在“数据库大小”页中需要确定“块大小”或采用默认值,一般 Oracle 9i 数据库块的大小以字节为单位。如果数据库用于决策支持系统(DSS),就要使用较大的块(8K);而 OLTP 则可以使用较小的块(4K)。“排序区域大小”则选择排序区域大小的默认值。“数据库字符集”是在计算机屏幕上显示字符时所使用的编码方案,有数据库字符集和国家字符集两种可用的字符集。

“文件位置”页用于确定初始化参数文件的路径,或者直接采用默认路径。此文件包含配置参数值,控制数据库例程的内存分配和进程设置,每次启动数据库例程时都要读取此文件。文件的默认位置是 Oracle_Home/dbs/spfile.ora。

在确定这些标签页后,单击“下一步”,进入“数据库存储”对话框(见图 2.7)。

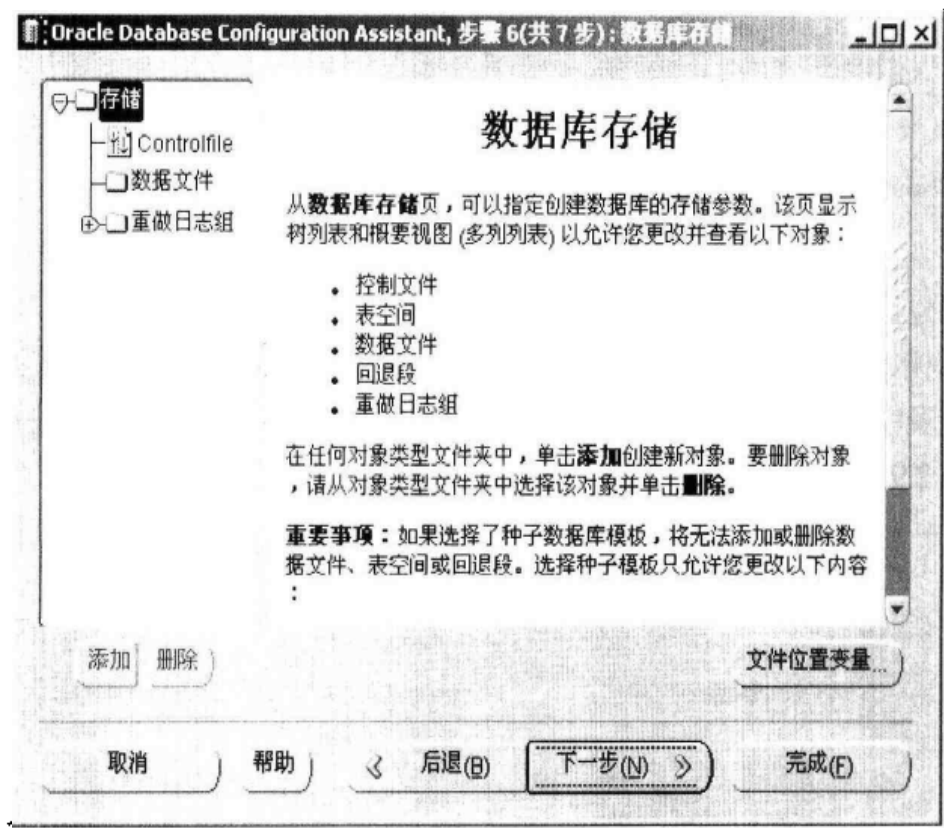


图 2.7 “数据库存储”对话框

4. 数据库的存储参数与创建选择设置

在“数据库存储”对话框中,可以指定数据库的存储参数。该页显示树列表和概要视图(多列列表),允许用户更改并且查看控制文件、表空间、数据文件、回退段、重做日志组和注释等对象。如果正在使用“自动撤消管理”,则不必配置回退段。从任何对象类型文件夹中,单击添加,创建新对象。若要删除某对象,可从对象类型文件夹中选择该对象并且单击删除。选择该类型模板允许更改以下内容:数据库名称、数据文件的目标

位置、控制文件或日志组、INIT.ora。确定这些选项后，单击“下一步”按钮，进入“创建选项”对话框（见图 2.8）。

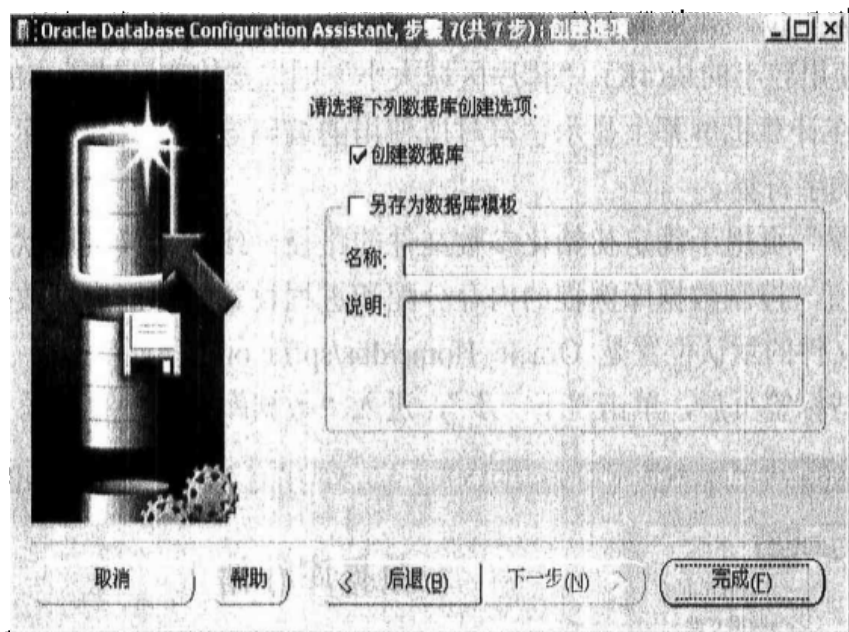


图 2.8 “创建选项”对话框

在“创建选项”对话框中选择“创建数据库”，可以顺利创建数据库。如果选择“另存为数据库模板”选项，则将数据库创建参数另存为模板，模板会自动添加到可用数据库模板的列表中。这里选择了“创建数据库”，然后单击“完成”按钮，进入“概要”对话框（见图 2.9）。

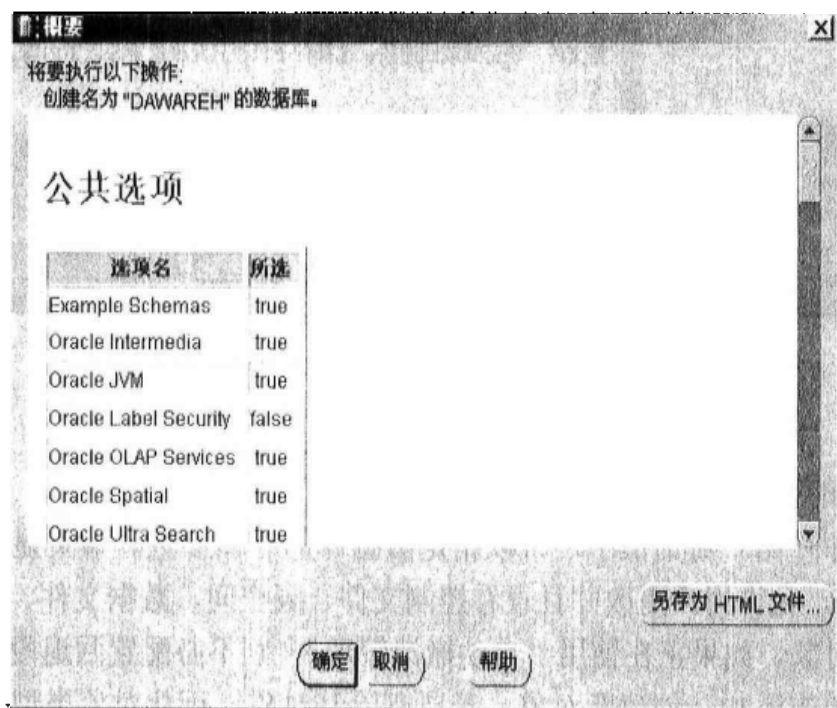


图 2.9 “概要”对话框

在“概要”对话框中可以浏览前面所做的数据仓库创建设置。如果对这些设置认同,就可以单击“确定”按钮,返回“创建选项”对话框后,单击“完成”按钮,进入数据库创建过程。数据库创建成功后,可以在 Oracle Enterprise Manager 的窗口中查看所创建成功的数据库。

2.2.2 Oracle 数据仓库表空间的创建

1. Oracle 企业管理器

在完成数据仓库的创建以后,还需要对数据仓库设置表空间,便于更好地存储、管理数据表。数据仓库的表空间的创建,可以使用 Oracle 的企业管理器 (Oracle Enterprise Manager) 设置。

Oracle 企业管理器是一个管理框架,它由如下 3 个层次组成。Console 及其集成工具为管理员提供管理整个 Oracle 环境的图形界面。Management Server 和数据库资料档案库为处理系统管理任务提供可伸缩的中间层。Intelligent Agents 安装在每个网络节点上,用来监视节点提供的服务,执行来自 Management Server 的任务。当需要管理数据库,而不需要监视事件或调度作业时,还可以用独立模式启动 Console。

Console 是用于所有 Oracle 企业管理器操作的主界面。它提供菜单、工具栏、联机帮助和导航器,以访问 Management Server 服务、Oracle 工具以及其他集成功能。导航器是 Console 的主要导航组件,允许通过主体/详细资料视图对所有受管目标和相关功能进行快捷访问。在导航器中选择对象,即可在 Console 的右侧显示区窗格中显示该对象的相关信息或合适的用户界面功能。Console 屏幕的格式和所显示的应用程序取决于所购买的产品和用户首选项。

Oracle 企业管理器的启动,可以通过“开始”→“程序”→“Oracle-OraHome90”→“Oracle Enterprise Manager”(参见图 2.10)系列命令进入企业管理器登录信息对话框。

在企业管理器的登录信息对话框中主要完成“独立启动”或登录到 Oracle 管理服务器 (Oracle Management Server) 的选择。独立启动时,将被直接连接到数据库。允许一个人使用一个或多个应用程序。如果登录到 Oracle 管理服务器,则可以完成事件、作业、共享、封锁、组、寻呼、电子邮件、调度、通知、历史记录收集,以及在 Web 浏览器中运行应用程序的其他功能。

登录 Oracle 管理服务器,先要保证计算机已经启动 Oracle 管理服务器,再在 Oracle 管理服务器登录对话框中输入身份证明。启动 Oracle 管理服务器,可以按照以下命令执行:“开始”→“设置”→“控制面板”→“管理工具”→“服务”,在服务窗口中右键单击“Oracle OraHame90Management Server”选择弹出式菜单中的“启动”命令,或选择“操作”菜单中的“启动”命令,进入“Oracle Management Server”(见图 2.11)。

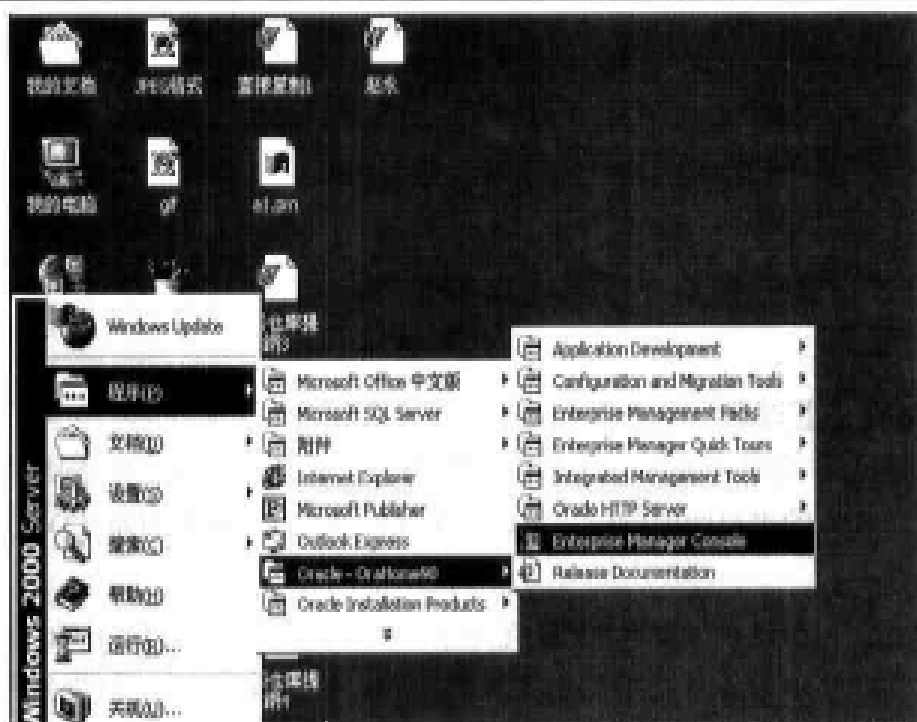


图 2.10 “Oracle Enterprise Manager” 的启动



图 2.11 “Oracle Management Server” 的启动

如果在资料档案库创建过程中选择不存储资料档案库身份证明,则在启动 Management Server 时,必须在“控制面板”的“启动参数”字段中输入用于创建数据库用户的该资料档案库的用户名以及用户口令。“启动参数”字段在服务列表下。该登录对话框中的各项内容说明如下。

管理员是 Oracle 管理服务器管理员的名称。此时是以 Management Server 维护网络资源的管理员身份登录,而不是以一个数据库用户的身份登录到某个数据库。Enterprise 的

默认管理员是 sysman，默认口令是 oem_temp。Management Server 键入或选择正在运行 Oracle Management Server 的节点名。

2. 表空间的创建

由于数据仓库庞大的信息量，因此需要对数据在 Oracle 中的物理存放位置进行设置。在数据仓库中需要系统支持的表空间和应用表空间。

系统支持的表空间在安装 Oracle 9i 时就完成了设置。如果在安装 Oracle 中接受了 Oracle 所提供的初始数据库，将产生由 SYSTEM, ROLLBACK, TEMPORARY, TOOLS, USERS 等系统所支持的表空间。其中 ROLLBACK 需要 500MB, TEMPORARY 需要 2GB。

应用表空间则是用于存放数据仓库数据的表空间。在建立应用表空间之前要先建立数据库，然后估计存放在该数据库中的数据和索引所需的空间。如果一时难以断定表空间的大小，可以根据数据仓库数据库的数据源初步估算数据仓库中数据的行数。

在 Oracle 中创建数据仓库时，一般将数据仓库的数据与索引分别存放在不同的表空间中，而维所在的表空间需要设置为“共享”属性，因为这些维一般要被多个表或不同的用户所共享。

表空间的创建可以在 Oracle Enterprise Manger Console 中进行，在 Oracle Enterprise Manger Console 中用鼠标连续展开“网络”、“数据库”、“OEMREP2”（已经创建的数据仓库名）（参见图 2.12），此时出现“数据库连接信息”对话框（见图 2.13）。在用户名和口令输入框中分别输入用户名和口令后，单击“连接身份”下拉框，选择其中的 SYSOPER 或 SYSDBA 身份。单击“确定”按钮将把 Oracle Enterprise Manger Console 中所选中的数据库展开，然后单击“对象”菜单，从下拉菜单中选择“创建”菜单项，弹出“对象创建列表”对话框（见图 2.14）。

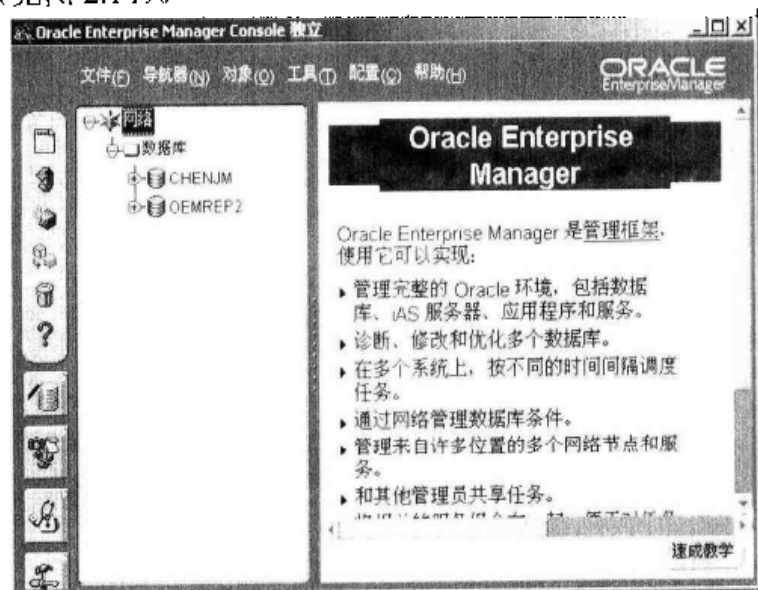


图 2.12 在 Oracle Enterprise Manger Console 中创建表空间



图 2.13 “数据库连接信息”对话框

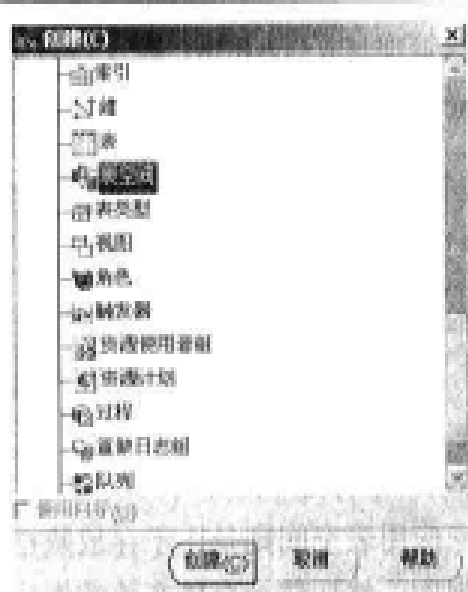


图 2.14 “对象创建列表”对话框

在“对象创建列表”对话框中选择“表空间”选项，单击“创建”按钮。调出“创建表空间”对话框（见图 2.15）。对话框中包含“一般信息”标签页和“存储”标签页。



图 2.15 “创建表空间”对话框

在“一般信息”页中有用于定义表空间的名称。“名称”用于输入要创建的表空间名

称。在对话框中的文件多栏列表中输入“文件名”、“文件目录”、“大小”和“已使用”等情况。且可以用 MB 或 KB 指定属于表空间的新数据文件大小。本地管理的临时表空间是“临时文件”，而不是“数据文件”。编辑(铅笔图标)显示“编辑数据文件”属性工作表，可为“数据文件”列表中所选的数据文件编辑文件说明。单击表中的垃圾桶图标，可以移去“数据文件”列表中所选中的数据文件。移去的文件只能是尚未提交给表空间的新增数据文件。

如果状态为“脱机”，可以选择单选按钮，将状态更改为“联机”。如果状态为“联机”，选择单选按钮，出现一个弹出式菜单，上有“正常脱机”、“临时脱机”、“立即脱机”或“脱机恢复”选项。在创建模式中“联机”为默认值。如果状态为“联机”，还可使用“只读”选项。如果表空间处于只读状态，则可启用“可写模式”菜单选项。选择“可写模式”菜单选项时，表空间成为可读写表空间，状态变为“联机”。如果数据库中不存在打开的事务处理，或表空间中存在活动的回退段，则取消“只读”选项。

类型选择项中的“永久”项可以指定表空间用于存放永久性数据库对象。临时选项则指定表空间仅用于存放临时对象(排序段)。任何永久性对象都不能驻留于临时表空间中。如果选择“临时”作为该表空间的类型，则选择作为默认临时表空间复选框即被选中。如果应用程序连接到 Oracle，则可以在“存储”页上选择“本地管理”(由表空间进行的区管理)和“在字典中管理”(由数据字典进行的区管理)。管理方法确定后不能变更。

“本地管理”选项为各区的表空间在每个数据文件中保留一个位图，用来跟踪记录该数据文件中块的空闲状态或使用状态。本地管理的表空间对区进行本地管理，可以避免递归的空间管理操作。在字典管理的表空间中，区空间的占用或释放将导致回退段或数据字典表中空间的占用或释放操作，将会发生递归的空间管理操作。对区进行本地管理，能够自动跟踪记录临近空闲空间的情况，避免进行空闲区的合并操作。本地管理的区的大小可由系统自动确定。本地管理功能包括“自动分配”选项，该选项选中后，区大小由系统自动确定。由于 Oracle 可以确定各区的最佳大小，所以区大小是可变的。选中“统一”选项，则可以指定区大小，也可使用默认值(1 MB)。

用 KB 或 MB 为单位可以分配指定对象的区“大小”。启用事件记录的“是”选项将在启用事件记录时创建重做日志。该操作所用时间比不启用事件记录所用时间长，但在遇到意外失败的情况下可以恢复更新。“否”选项确定不启用事件记录，操作时间较短。但在遇到意外失败的情况下将无法恢复更新。“块大小”选项的选择取决于 init.ora 文件中所定义的内容。

“在字典中管理”选项主要用于低于 8i 版本的使用，这里就不介绍了。在确定以后，可以单击“显示 SQL”按钮将前面的创建表空间设置以 SQL 语句的形式表现出来，这对学习用 SQL 语句创建表空间是有益的。在完成所需的设置后，单击“创建”按钮，系统开始创建表空间；创建成功后，出现表空间创建成功提示框(见图 2.16)。表空间创建成

功后，可以依次展开 Oracle Enterprise Manger Console 中的“数据库”、“存储”和“表空间”，看见所创建的表空间。



图 2.16 表空间创建成功提示框

2.2.3 Oracle 数据仓库表的创建

1. 数据库表的创建

在完成表空间的创建后，就可进行表的创建。表创建向导的启动以与表空间创建向导方式相同，在图 2.14 的“创建”对话框中选择“表”，单击“创建”按钮，就可进入表创建向导的“简介”对话框（见图 2.17）。

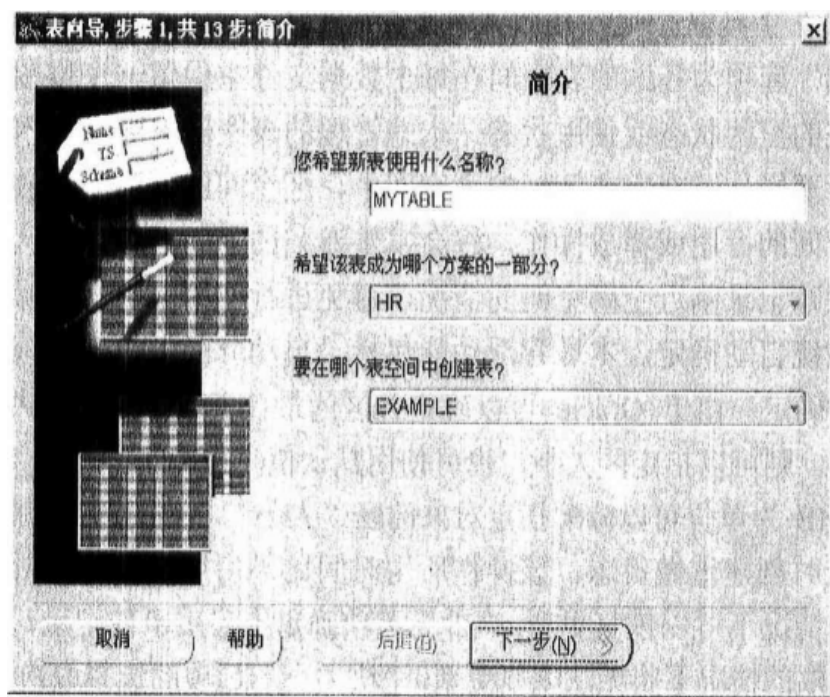


图 2.17 表创建向导“简介”对话框

在“简介”对话框中确定新表的表名、方案和表空间。表名可以是任意一个有效的 Oracle 标识符。表的方案类型可以从“希望成为哪个方案的一部分？”下拉列表中选择。表的所在表空间，可以从“要在哪个表空间中创建表？”下拉列表中选择。这里的表空

间是位于已选方案中的所有表空间。选择确定后，单击“下一步”按钮，进入“列定义”对话框（见图 2.18）。

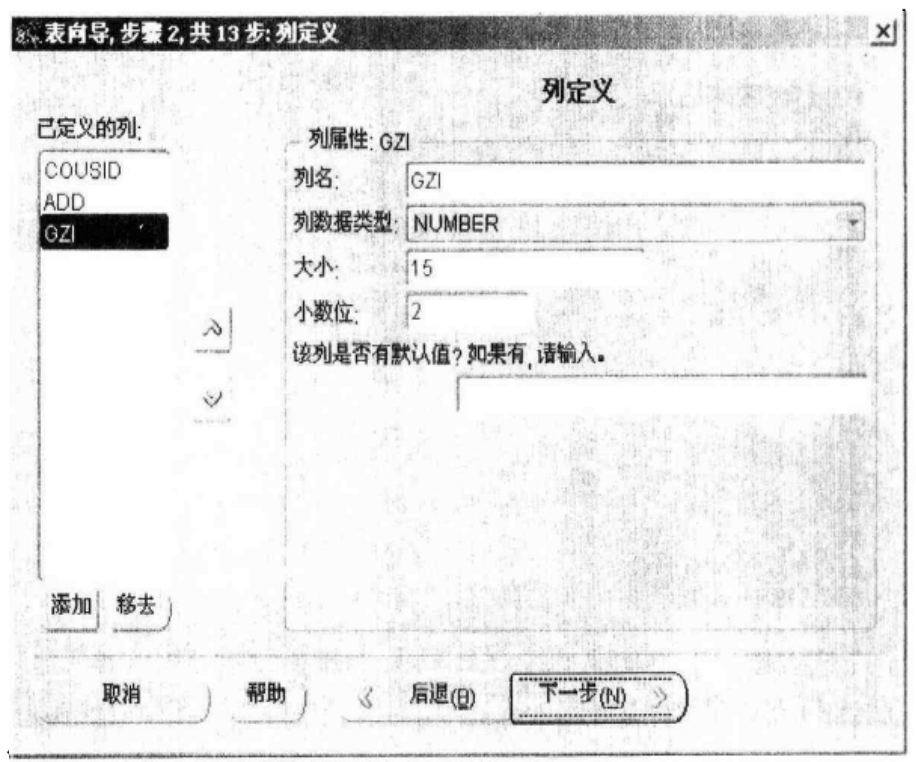


图 2.18 “列定义”对话框

在“列定义”对话框中的“列名”输入框中输入表的列名。从“列数据类型”下拉列表中为该列选择数据类型。在“大小”输入框中输入该列的字节数。“小数位数”用于指定小数点右边数字的位数。在“该列是否有默认值？如果有，请输入”输入框可以输入一个表达式，作为该列的默认值。对 `VARCHAR2` 或 `CHAR` 数据类型的默认值必须用单引号将该值括起来。在确定一个列定义后，可以单击“已定义的列”浏览框下的“添加”按钮再定义下一个列，“移去”按钮则可以将所选中的列从表中移去。“上箭头”和“下箭头”可对已定义的列进行位置的重新调整。完成表中的所有列的定义后，单击“下一步”按钮，进入“主关键字定义”对话框（见图 2.19）。

2. 数据库表列属性设置

在“主关键字定义”对话框中，可以定义表中各列的主关键字约束条件。若在“是否要为该表创建主关键字？”选项选择了“不，不创建主关键字”将不定义该表的主关键字。如果选择“是，创建主关键字”，则将创建主关键字，此时，前面在“列定义”对话框上定义的所有列都将出现在多栏列表中，这些列均被选为主关键字。单击各列的“次序”字段，即可显示它们在主关键字中的位置。如果单击“次序”列条目将某列名拖出列的序列，该列将不作为关键字处理。数据仓库的事实表中关键字的确定是必须的。

确定表的主关键字后,单击“下一步”按钮,进入“空约束条件和惟一性约束条件”对话框(见图 2.20)。

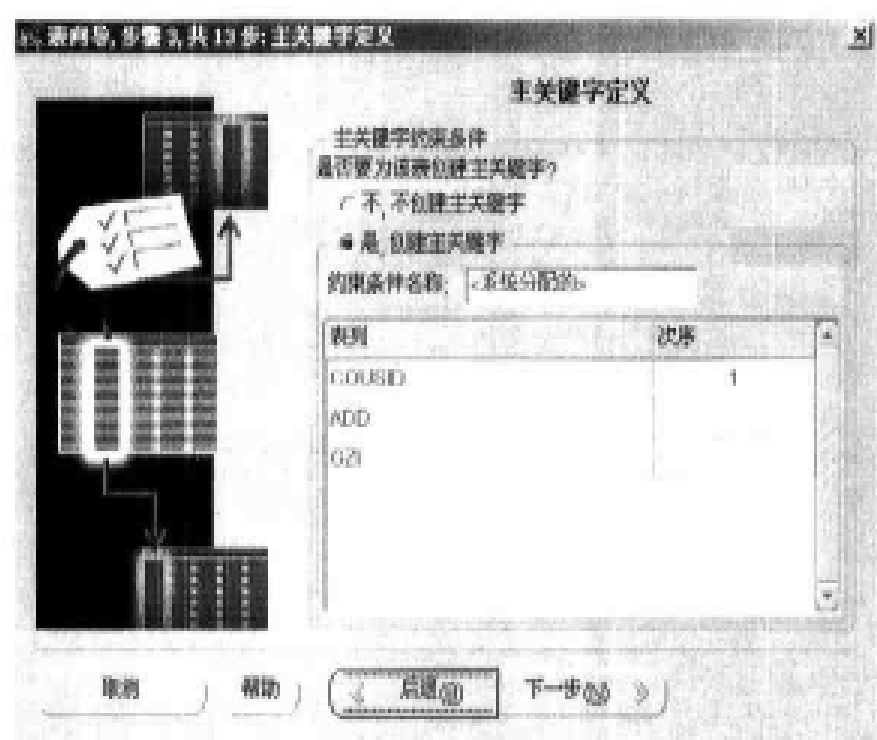


图 2.19 “主关键字定义”对话框

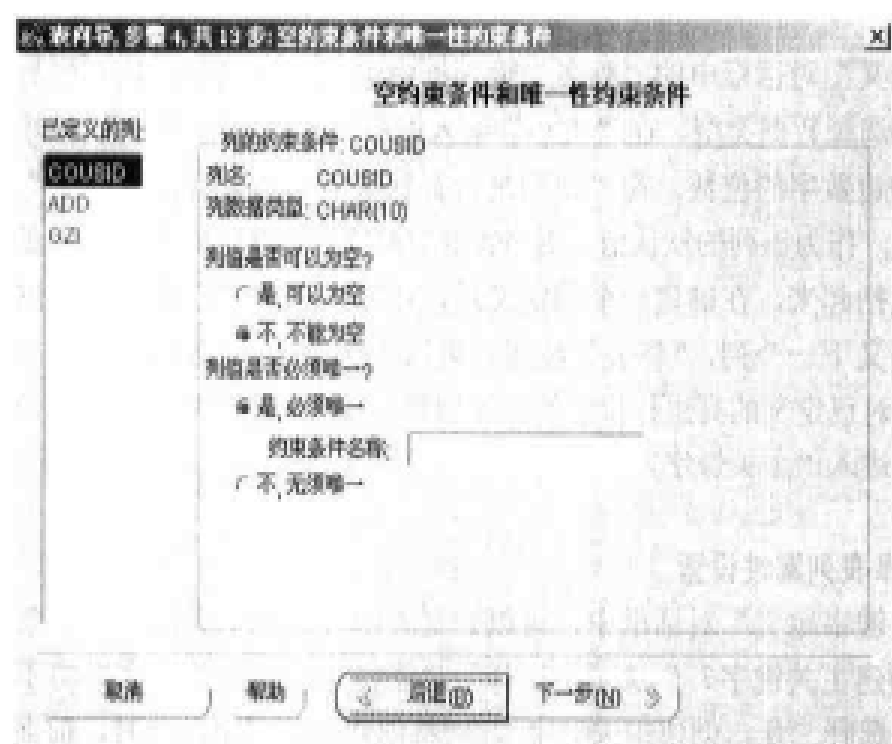


图 2.20 “空约束条件和惟一性约束条件”对话框

在“空约束条件和惟一性约束条件”对话框中,可以定义表中各列的空约束条件和

惟一性约束条件。单击已定义列列表中的一列，可以查看或修改其空约束条件和惟一性约束条件。“列值是否可以为空？”选项用于确定选中列的值是否可以为空。“列值是否必须惟一？”选项用于确定列值是否惟一的，一般在该列为主关键字时，选择“是”，否则可以选择“否”。在确定空约束条件和惟一性约束条件后，单击“下一步”按钮，进入“外约束条件”对话框（见图 2.21）。

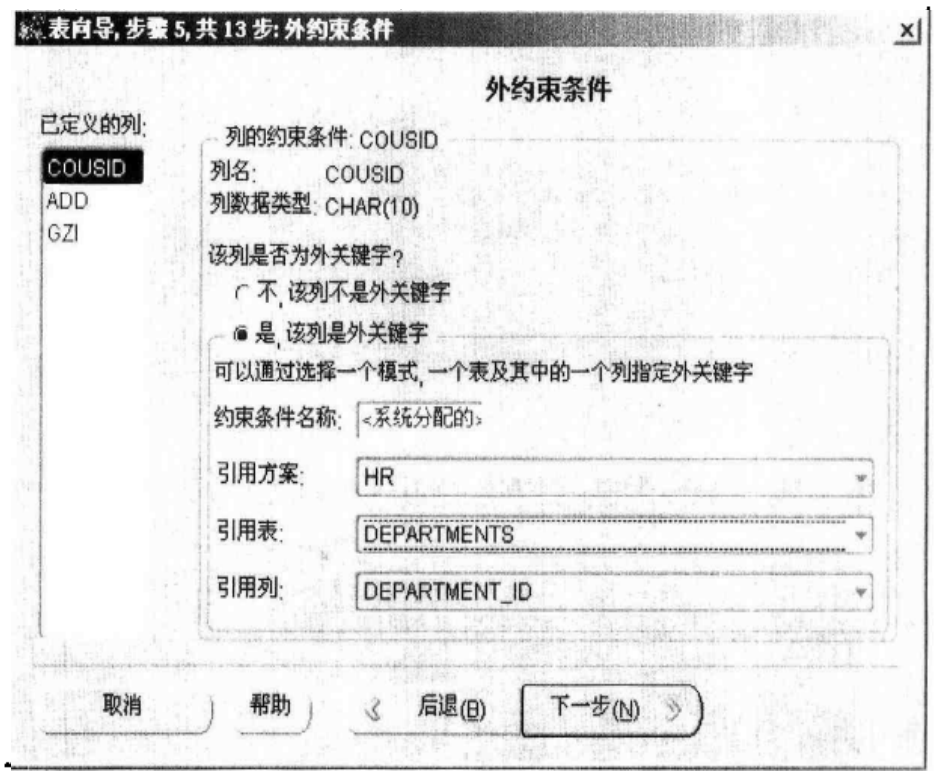


图 2.21 “外约束条件”对话框

在“外约束条件”对话框中，可以定义外关键字约束条件。每列的列名称和数据类型均已显示出来。如果选择“该列是否为外关键字？”选项的“是，该列是外关键字”，还需要从“引用方案”的下拉列表中选择当前数据库中的可用方案。确定引用方案后，还要依次从“引用表”下拉列表和“引用列”下拉列表中分别选择引用方案中的可用表和引用表中的可用列。数据仓库中的事实表与维表的连接就需要依靠“外约束条件”的设置。在“外约束条件”设置完成后，单击“下一步”按钮，进入“检查约束条件”对话框（见图 2.22）。

在“检查约束条件”对话框中可以指定表中列的检查条件。如果该列有检查条件，则选择“该列是否具有检查条件？”的“是，该列具有检查条件”。选中后，在“该列的检查条件是什么？”文本框中输入检查条件，本对话框可以用于数据仓库的数据加载过程中的数据清理操作。确定“检查约束条件”后，单击“下一步”按钮，进入“存储信息”对话框（见图 2.23）。

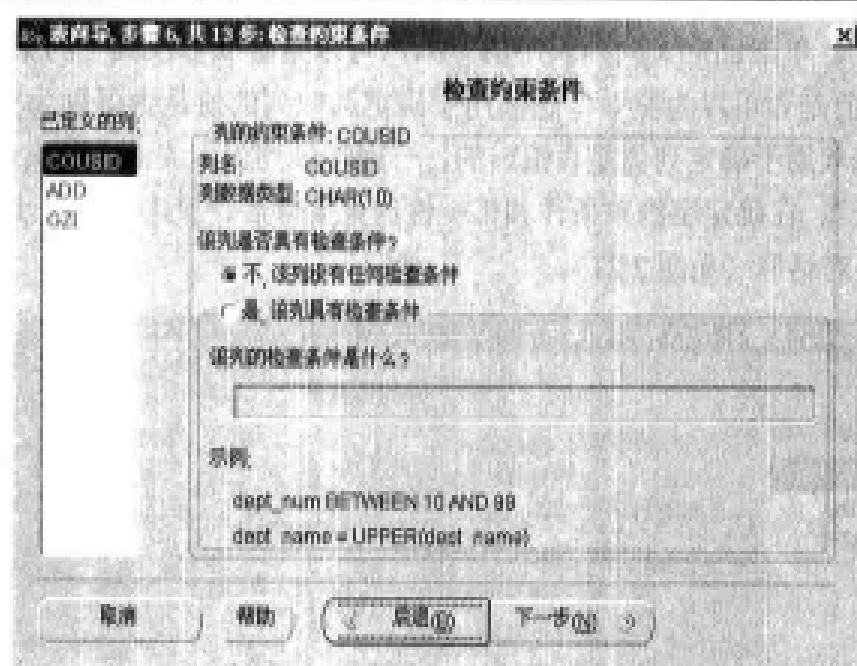


图 2.22 “检查约束条件”对话框



图 2.23 “存储信息”对话框

3. 数据库表存储属性设置

在“存储信息”对话框中，可以确定“初始行数”、“增长速率”、“更新操作”和“插入操作”等选项。“初始行数”需要根据数据仓库初始加载的数据量来确定，“增长速率”需要根据数据源的数据增长速率确定，“更新操作”可以选择“低或无”，而“插入操作”则要选择“高”选项。在完成存储信息设置后，单击“下一步”按钮，进入“分区选项”对话框（见图 2.24）。

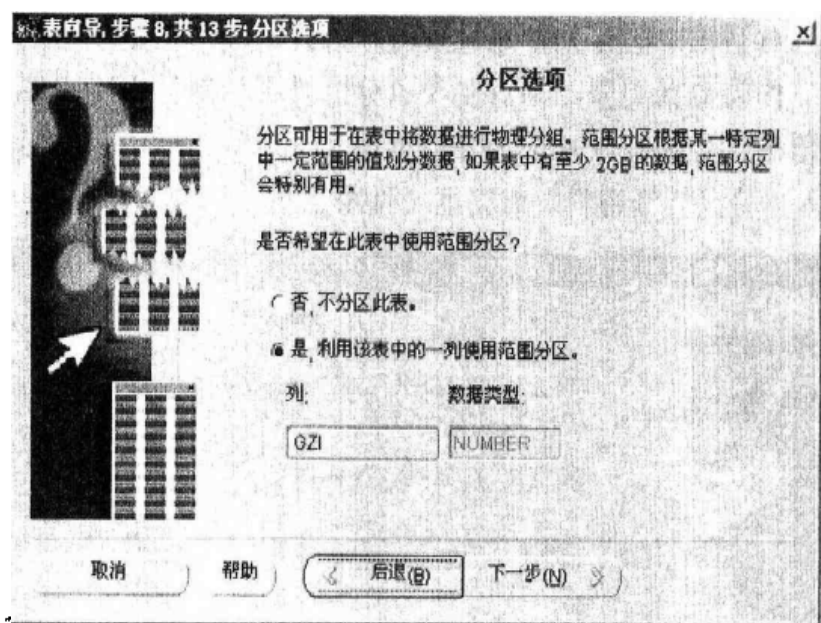


图 2.24 “分区选项”对话框

在“分区选项”对话框中可将非常大的表和索引分成较小的、更便于管理的分区, 从而解决支持巨型表和索引的问题。完成分区定义后, SQL 语句就可以访问各分区, 而不是全部表和索引。这样可以分析各个数据分区, 单独备份和恢复每个分区, 可以减少数据损坏造成的影响; 且将分区映射到磁盘驱动器, 实现 I/O 负载平衡。在选择了“是, 利用该表中的一列使用范围分区”后, 就要在表中选择一列作为分区根据。分区后的表中不能包含任何数据类型为 LONG 或 LONG RAW 的列。分区一般在表较大时采用, 比较理想。数据仓库中, 一般对事实表进行分区设置。在设置完成后, 单击“下一步”按钮, 进入“分区数据”对话框 (见图 2.25)。

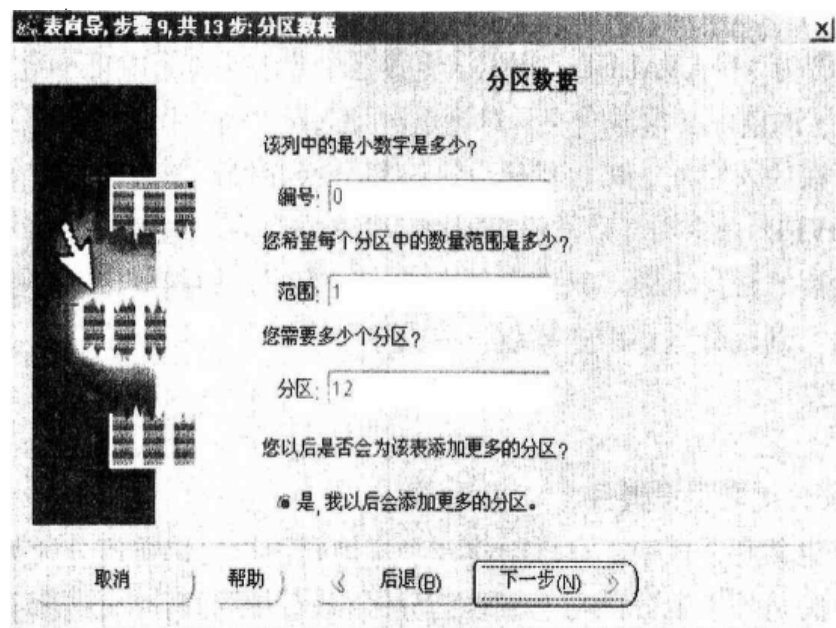


图 2.25 “分区数据”对话框

“分区数据”对话框中的有关参数设置与“分区选项”对话框中的数据类型有关。如果选择“数字”作为数据类型,则需要在“分区数据”对话框中设置列的最小数字、每个分区的数量范围、分区数以及是否今后再增加分区等。完成这些设置后,单击“下一步”按钮,进入“分区详细资料”对话框(见图 2.26)。

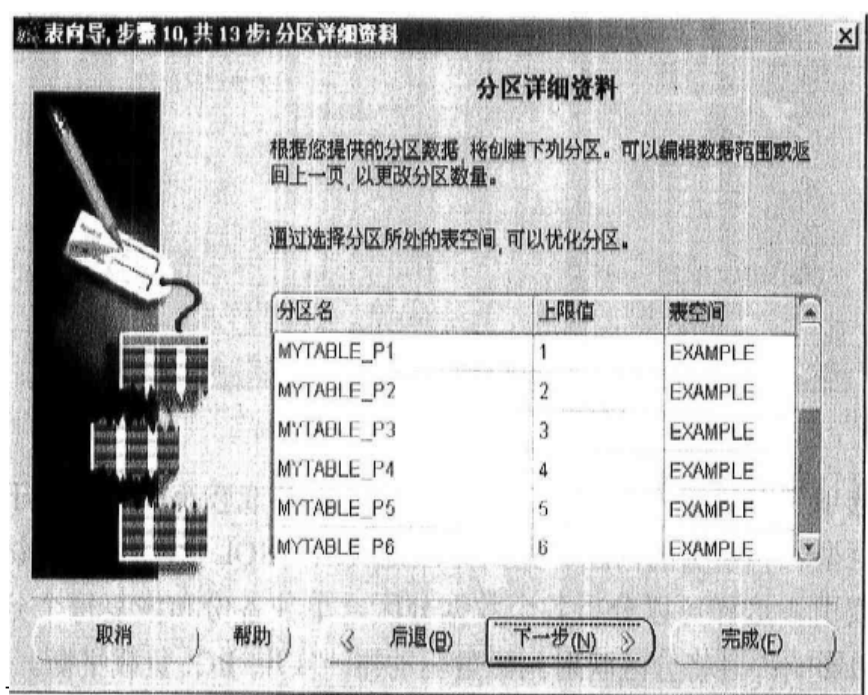


图 2.26 “分区详细资料”对话框

在“分区详细资料”对话框中,将根据分区数据创建分区。可以修改有关分区数据或返回上一对话框更改分区数量。分区名列出属于同一父级表的所有分区,这里各表分区名必须是唯一的。上限值为此分区允许的最大值,可以更改任何分区的上限值,但最后一个分区的值若为 MAXVALUE,则无法更改这个分区的值。如果不希望 MAXVALUE 作为最后一个分区的值,必须返回前一对话框,并选择“是,我以后会添加更多的分区”。表空间为分区所在的表空间,可以优化分区。将分区存储在单独的表空间的优点为:缩短因执行 RECOVER 命令而导致的停机时间。因为分区后,恢复单位变小,从而减少恢复脱机表空间所需的磁盘资源。只有存储在恢复表空间中的分区才会脱机,从而减少不可用的数据量。确认分区详细资料后,单击“下一步”按钮,进入“分区索引定义”对话框(见图 2.27)。

4. 数据库表分区索引设置

在分区索引定义对话框中,可以创建一个本地索引以更快访问分区数据。Oracle 创建本地索引的目的是使本地分区与基础表的分区相同。因为每个分区都有一个索引,所以索引是分区的“本地”索引。在前缀索引中,分区关键字的顺序与索引关键字是相关

的。例如, Column 1 是分区关键字, 索引关键字的顺序为 Column 1、Column 2、Column 3, 则索引带前缀。如果 Column 1 是分区关键字, 索引关键字的次序为 Column 2、Column 1、Column 3, 则索引不带前缀。表列中的列为本表所有列。排序中输入索引的次序。如果选择了创建主关键字, 而在“主关键字定义”页中为表列指定的次序与在此对话框中指定的次序不同, 则除了本地索引外, 还会创建一个全局索引。全局索引可以包含多个分区的值, 即索引的值可以跨越多个表。完成分区索引设置后, 单击“下一步”按钮, 进入“分区索引详细资料”对话框(见图 2.28)。

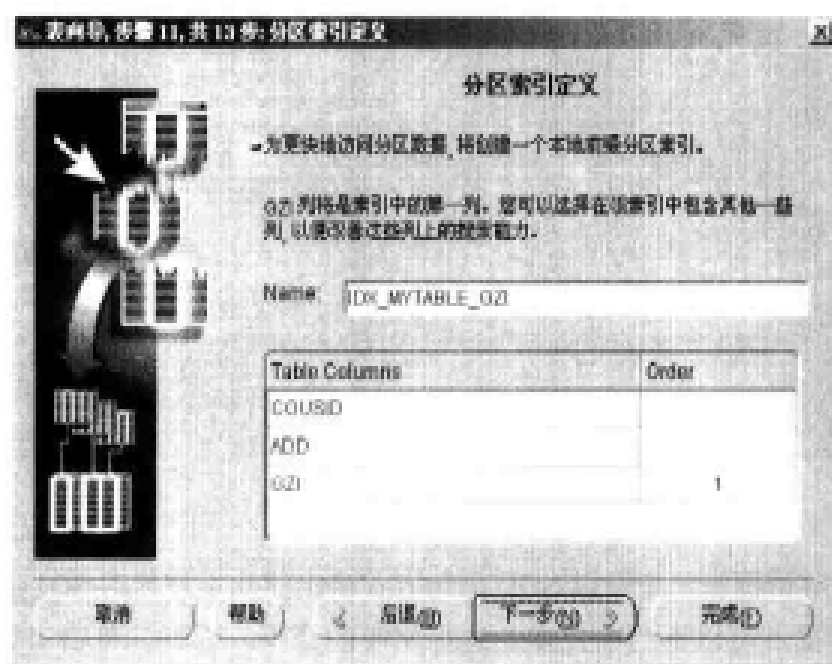


图 2.27 “分区索引定义”对话框



图 2.28 “分区索引详细资料”对话框

在“分区索引详细资料”对话框中可以命名索引分区，且将这些索引分区放在一个单独的表空间中。确认分区索引详细资料后，单击“下一步”按钮，进入“概要”对话框（见图 2.29）。

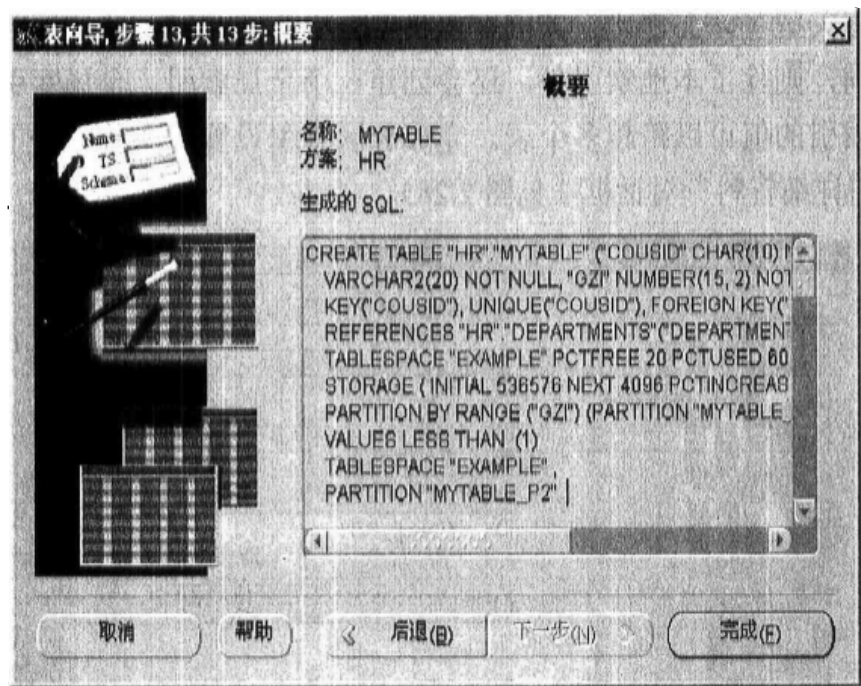


图 2.29 “概要”对话框

在“概要”对话框中，主要以 SQL 语句方式给出前几步设置的 SQL 语句表示。若要更改表和索引定义，可以返回前几步，进行相应的修改。否则，单击“完成”按钮，完成表的创建。

2.3 Oracle 数据仓库的维与立方创建

2.3.1 Oracle 数据仓库的维创建

在 Oracle Enterprise Manager Console 中依次展开“数据库”、实际的数据库名、“OLAP”，用右键单击“维”，选择弹出菜单中的“使用向导创建”菜单项（参见图 2.30），如果对创维过程熟悉后，可以直接选择“创建”菜单项。在弹出的创建维多项标签页中完成维创建的设置。这里选择“使用向导创建”菜单项后，进入“创建维向导欢迎框”。单击其中的“下一步”按钮，进入“选择维类型”对话框（见图 2.31）。

在“选择维类型”对话框中，选择“创建维对象”还是“创建时间维对象”。时间维中的各级一般用于指定时段，如年和月，时间维主要用于 OLAP 计算中按时段对数据进行分类。时间维包括两个必需的属性：End-Date(结束日期)和 Time-Span(时间长度)，必须

为所有维级指定这两个属性。在创建时间维前必须已经存在维表，有用来指定所有的时间单位。选择“创建维对象”后，单击“下一步”按钮，进入“名称和方案”对话框（见图 2.32）。

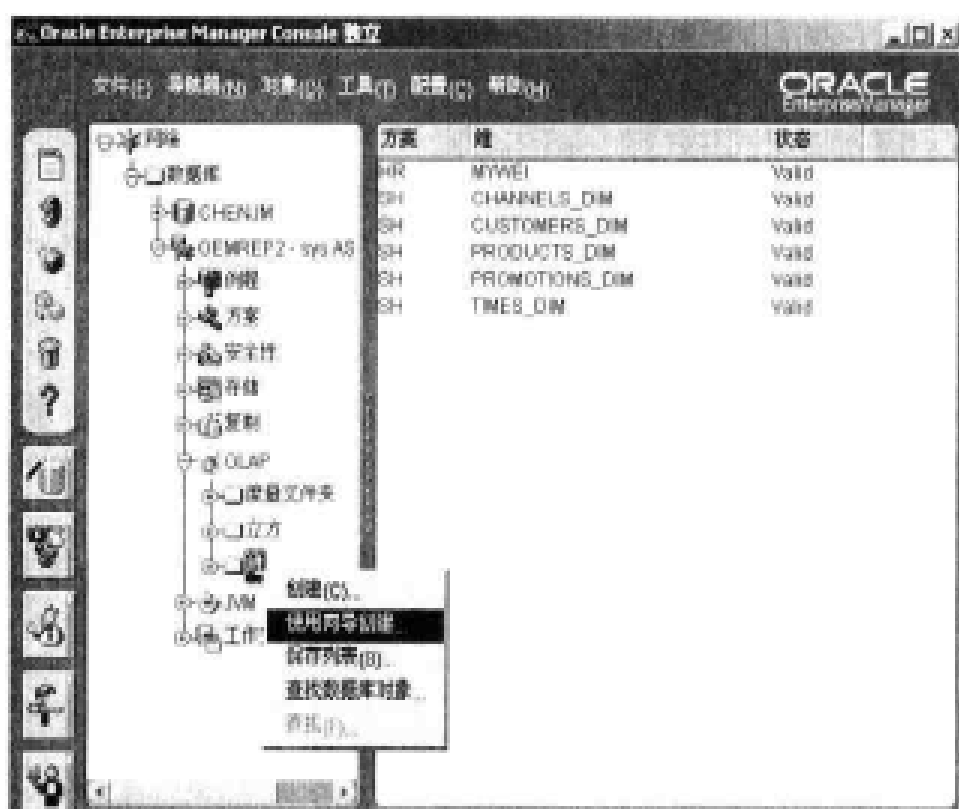


图 2.30 使用向导创建

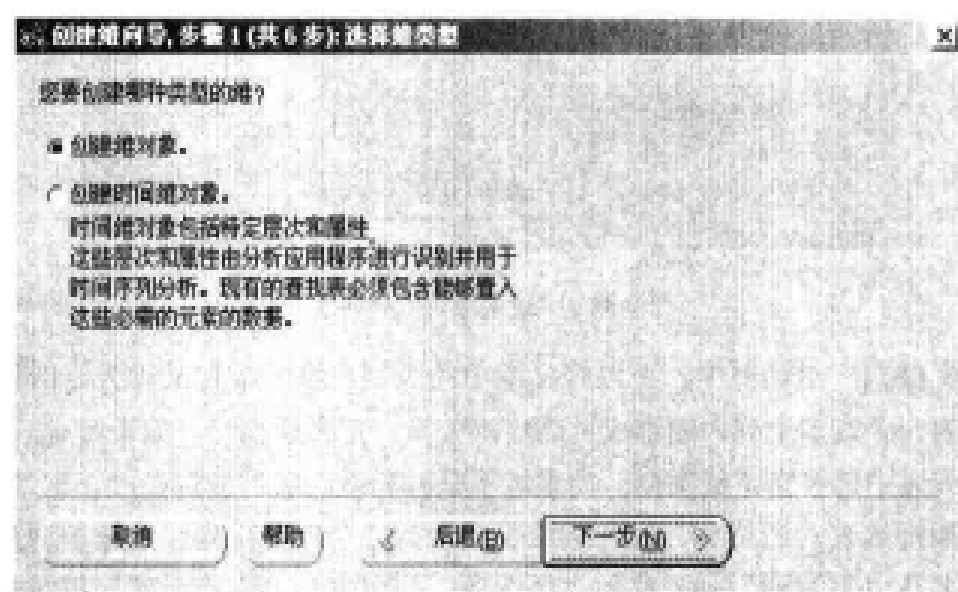


图 2.31 “选择维类型”对话框

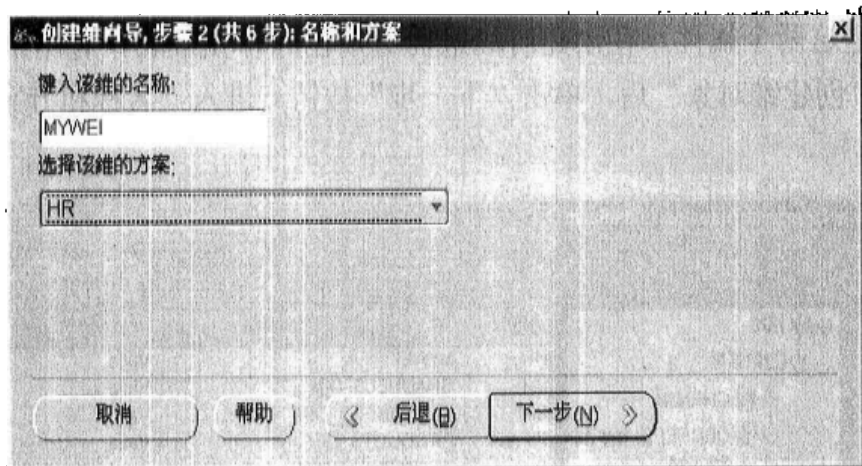


图 2.32 “名称和方案”对话框

在“名称和方案”对话框中输入维的名称和确定该维所选择的方案，维名称直接输入维名称输入框，而维的方案则从下拉列表中选择。确定名称和方案后，单击“下一步”按钮后，进入“定义级别”对话框（见图 2.33）。



图 2.33 “定义级别”对话框

在“定义级别”对话框中，可以定义维中的级别。维的级别又称为维的层次，标识存储在维表的一个或多个列内的维成员的层次关系。定义级别时，如果维是一个时间维，可以从下拉列表中选择指定级类型。如果维不是时间维，级类型将是“正常”。在“级别属性”框中指定维表的名称和方案。从维表中选择一个或多个列作为级的源列。使用箭头按钮将列名从“可用列”列表移到“已选列”列表中。单击“新建”按钮且在“级别属性”框的名称中输入级的名称。

在数据仓库的“星形”模型中，维表未规范化，该维的各级分别对应于单个表内的

各列。在“雪花”模型中，维表是规范化的，该维的各级分别对应于不同表中的各列。完成维的级别定义后，单击“下一步”按钮，进入“定义属性”对话框（见图 2.34）。



图 2.34 “定义属性”对话框

在“定义属性”对话框中，(启用 OLAP)“维”属性工作表的“属性”部分为维指定级属性集合。级属性代表给定级别上维成员的有关信息。例如，在产品维中，可能存在与每个产品和每个产品类别相关联的商标名。该属性集合可能被称作“商标”，可以包含两个级属性：一个级属性将列与产品 ID 级相关联，一个级属性将列与该产品类别级相关联。只要每个级属性在一个单独的属性集中定义，一个级别可多个级属性。

每个属性是一组级属性对(一对或多对)。单击“新建”按钮可以创建新的属性。单击“删除”按钮则删除属性。

属性类型 Long_Description 和 Short_Description 的属性被 OLAP 客户机应用程序用于数据显示区的列、行和页标题。如果要为维用于 OLAP 元数据，要为所有级别指定详细说明和简要说明。如果没有这些属性的列，或者不将该维用于 OLAP 目的，则可以创建不具有这些属性的维。在将该维用于 OLAP 客户机应用程序以前，必须确保维表中存在必要的列，且创建类型为详细说明和简要说明的属性。End_Date 和 Time_Span 只针对时间维，OLAP 客户机应用程序需要 End_Date 和 Time_Span 类型的属性进行涉及时间的计算。End_Date 指定时段中的最后一天。Time_Span 指定时段中包含时段的跨度。如果拥有的列未包含所有级别的 End_Date 和 Time_Span 信息，就无法将该维创建为时间维。

“属性”框中为已选的属性名称和属性类型。“可用级别”框中为维表可用级别。“选定的级别”为从“可用级别”列表框中选择的级别。“属性源列”列表框将指示存储该级属性信息的位置。

从“可用级别”框选择一个或多个级别，然后使用箭头按钮将其移到“选定的级别”框。对于详细说明和简要说明（以及时间维的 `End_Date` 和 `Time_Span`），必须选择所有可用的级别。对于每个已选级，单击对应的“源列”框，将显示列的下拉列表，为级属性选择源列。在属性设置成功以后，单击“下一步”按钮，进入“定义层次”对话框（见图 2.35）。

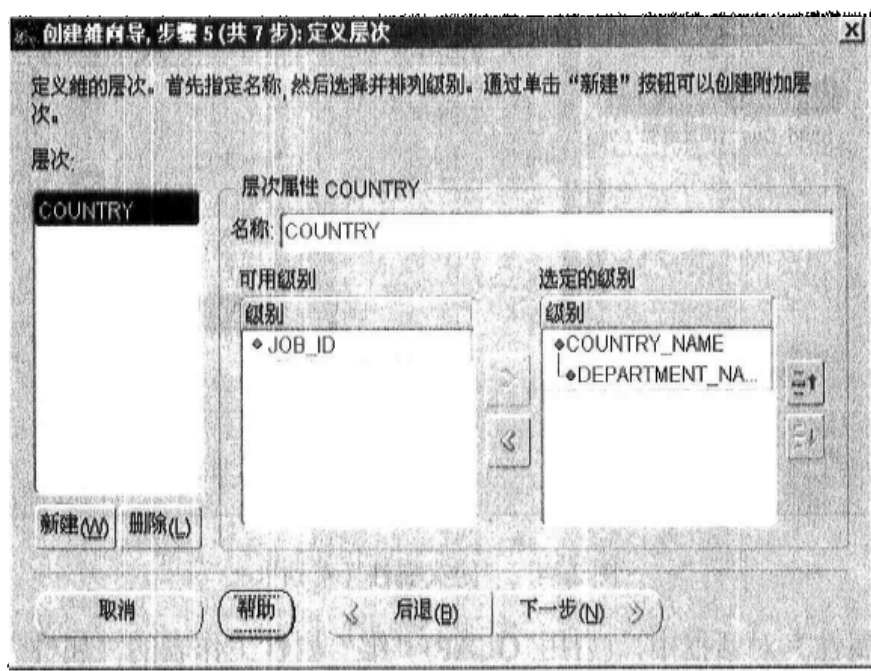


图 2.35 “定义层次”对话框

“定义层次”对话框用于定义维的层次。层次用于建立两个级之间的父子关系。维可以具有一个层次或多个层次，也可能不具有层次。如果维具有多个层次，则相同级可在“定义层次”对话框中，单击“新建”且在“属性”框中键入层次的名称进行层次的定义。层次名将在“层次”框中显示。如果创建时间维，在默认情况下，“日历”和“财政”层次将显示在“层次”框中。可以将级的有序集合映射到这些层次，也可以删除它们，以创建自己需要的层次。使用“属性”框按层次关系对各组级进行排序。可以用箭头按钮将级从“可用级别”列表移到“选定的级别”列表中。在“选定的级别”列表中，层次的级按降序排列。可用“选定的级别”框右边的箭头按钮在层次内重新确定级的位置。确定层次以后，单击“下一步”按钮，进入“指定连接”对话框（见图 2.36）。

在“指定连接”对话框中，可以指定某层次中两级之间的外关键字关系。只有当维的层次级来自于独立维表（如“雪花”模型）时，才显示“指定连接”对话框，指定层次的连接关键字，可以在“层次”框中单击层次名称。该层次中要求连接的父子级别将在“级别对”框中显示。选择级别对时，父级的源列或列集合将在“指定连接”中的“父列”下显示。然后，单击对应的“子列”框，从下拉列表中选择包含父列值的关键字列。完成“连接指定”后单击“下一步”按钮，进入“OLAP”对话框（见图 2.37）。

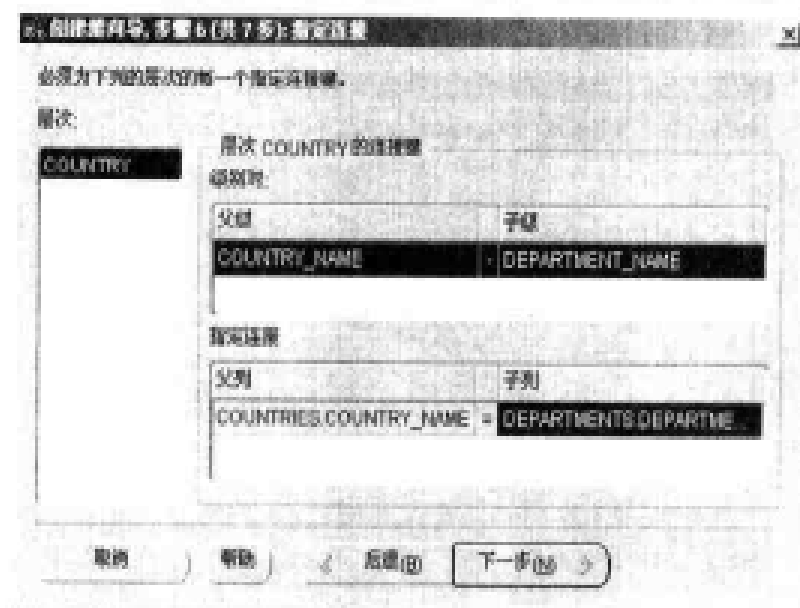


图 2.36 “指定连接”对话框



图 2.37 “OLAP”对话框

在“OLAP”对话框中可以指定与维相关联的 OLAP 专用信息。如果维不用于 OLAP 目的，OLAP 选项可以忽略。只单击“下一步”便可进入“概要”对话框。OLAP 客户机应用程序使用 OLAP 选项处理查询和标记显示的查询结果。如果没有为 OLAP 选项提供值，将提供默认值。OLAP 选项指定维的复数名称以及每个维组件的显示名称。OLAP 客户机应用程序使用此信息标记数据的显示。OLAP 选项还包括默认的显示层次，当 OLAP 客户机应用程序按层次累积数据时将使用该显示层次。在确定了 OLAP 的显示信息后，单击“下一步”按钮，进入“概要”对话框（见图 2.38）。

在“概要”对话框中，左边以图视方式表达了所创建的维，右边文本框以 SQL 语句表示所创建的维。如果对所创建的维不满意可以后退，进行修改。对所创建的维满意后，

单击“完成”按钮，进行维的创建。维创建成功后，将显示维创建成功提示框。



图 2.38 “概要”对话框

2.3.2 Oracle 数据仓库的立方创建

立方可以表示存储在数据仓库事实表中的多维数据。立方的创建需要在 Oracle Enterprise Manager Console 中依次展开“网络”、“数据库”、某一数据库名、“OLAP”，右键单击“立方”，从弹出菜单中选择“创建”或“使用向导创建”菜单项进行立方体的创建（参见图 2.39）。这里选择“使用向导创建”菜单项进行立方的创建，先进入“创建立方向导”的欢迎对话框。在欢迎对话框中单击“下一步”按钮，进入“提供一般信息”对话框（见图 2.40）。

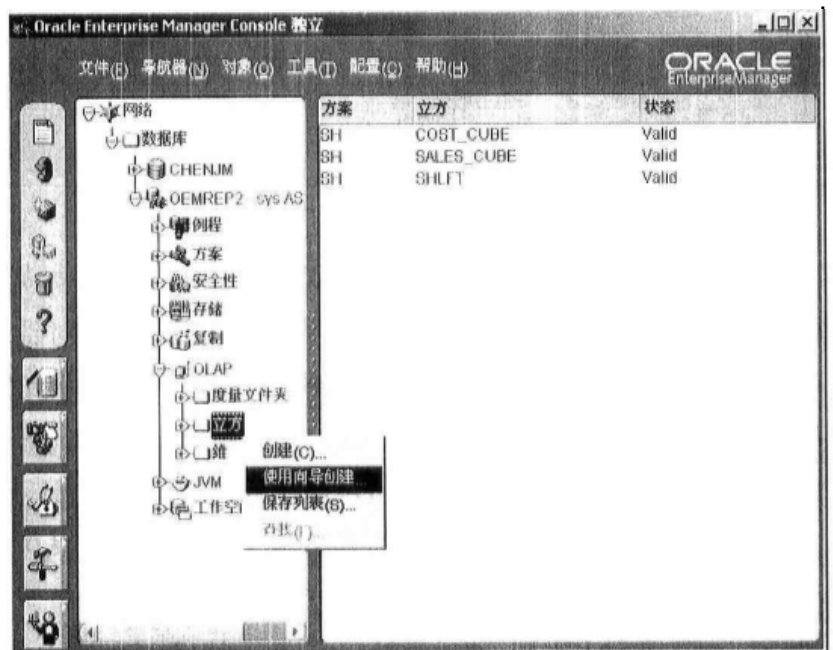


图 2.39 使用向导创建立方体

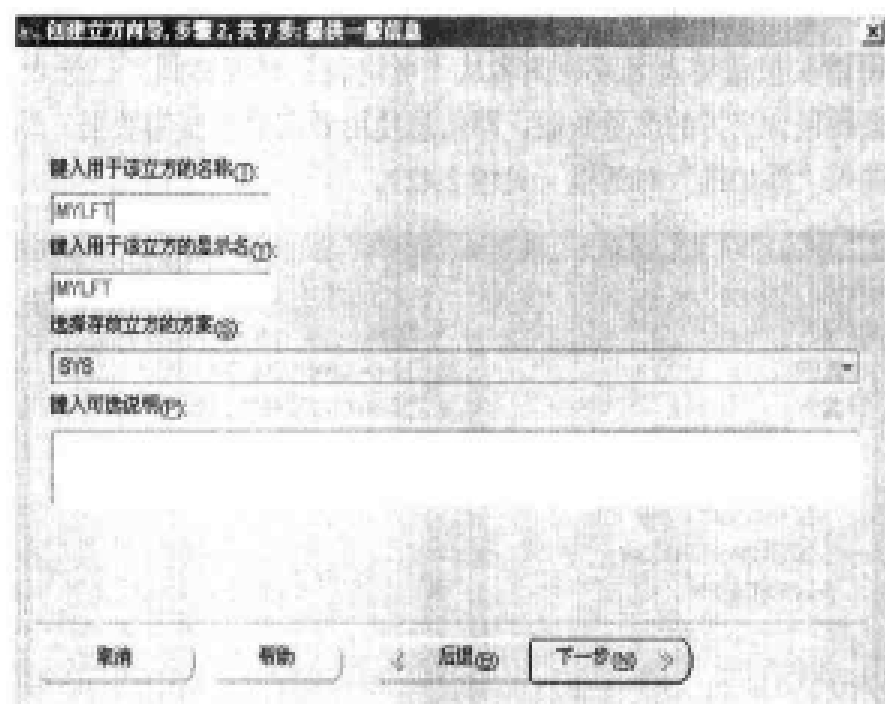


图 2.40 “提供一般信息”对话框

在“提供一般信息”对话框中,需要在“键入用于该立方的名称”输入框中输入所创建的立方名称,且从下拉列表中选择创建立方的方案。在“键入用于该立方的显示名”中输入提供给 OLAP 管理使用的显示名称,在“键入可选说明”框中输入给 OLAP 管理使用的说明。然后,单击“下一步”按钮,进入“选择事实表”对话框(见图 2.41)。

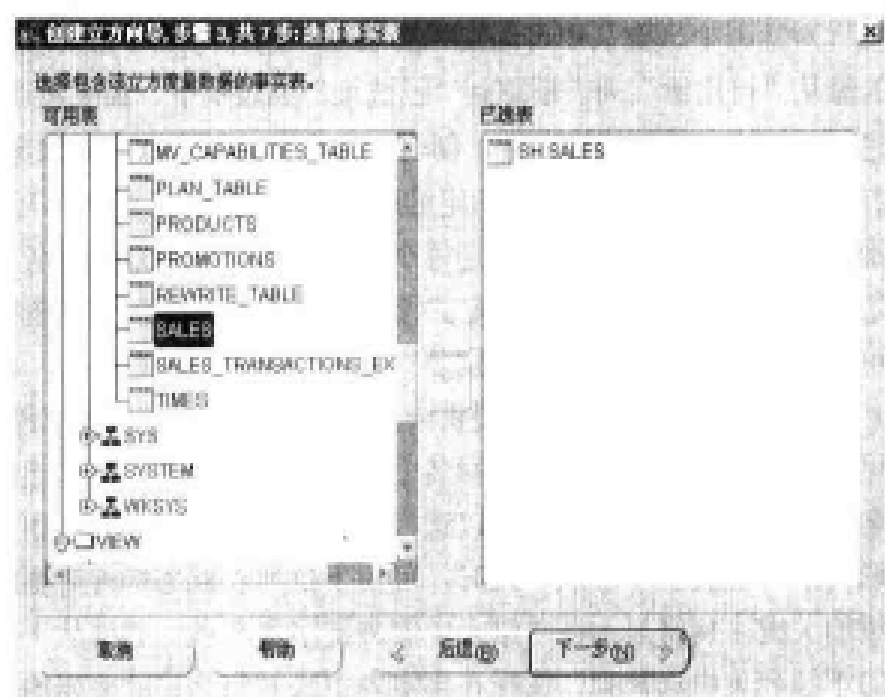


图 2.41 “选择事实表”对话框

在“选择事实表”对话框中，要在“可用表”列表框中选择一个事实表或视图作为立方的基础。用箭头按钮将表名或视图名从“可用表”列表移到“已选表”窗口中。事实表或视图中必须包含立方的度量数据，即信息使用者需要观察的数据。然后，单击“下一步”按钮，进入“添加维”对话框（见图 2.42）。

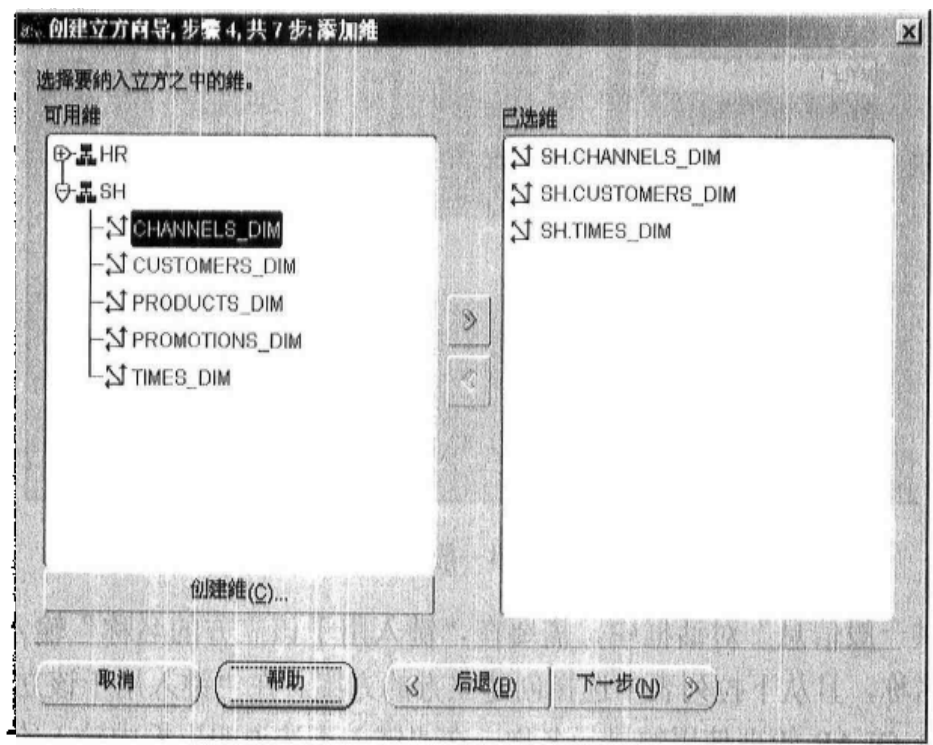


图 2.42 “添加维”对话框

在“添加维”对话框中，要从“可用维”列表框中选择与立方相关联的维，且用箭头按钮将选中的维从“可用维”列表框移到“已选维”列表框中。如果没有所需要的维，可以单击“创建维”按钮启动创建维向导，创建一个新维。每个立方必须至少有一个维。选中维后，单击“下一步”按钮，进入“指定维属性”对话框（见图 2.43）。

在“指定维属性”对话框中，需要指定每个与立方相关联的维的别名、默认层次和事实表与维表之间的连接属性。上一步指定的维将在“维（别名）”列表框中列出。选中这些维后，就可以为其指定属性。在“维别名”框中，输入维的逻辑名，这样就可以在相同的实际维上，创建不同的逻辑维。不提供维别名时，就使用维对象的名称。

在“计算层次”框中，指定在维或维别名上聚集数据的层次。在“连接级”框中，指定维或维别名连接到事实表所在级。可以选择维的叶级作为连接级，如果有多个层次与维对象相关联时维可能有多个叶级。此时，可用下拉列表框选择所需要的叶级。指定连接级后，该级的关键字列将显示在“维表中的关键字列”下。通过从“事实表中的外关键字列”下的下拉列表中选择列，选择在事实表中映射到此关键字列的外关键字列。使维表的关键字与事实表中的外关键字列建立某种约束。维属性指定后，单击“下一步”

按钮, 进入“指定度量”对话框 (见图 2.44)。

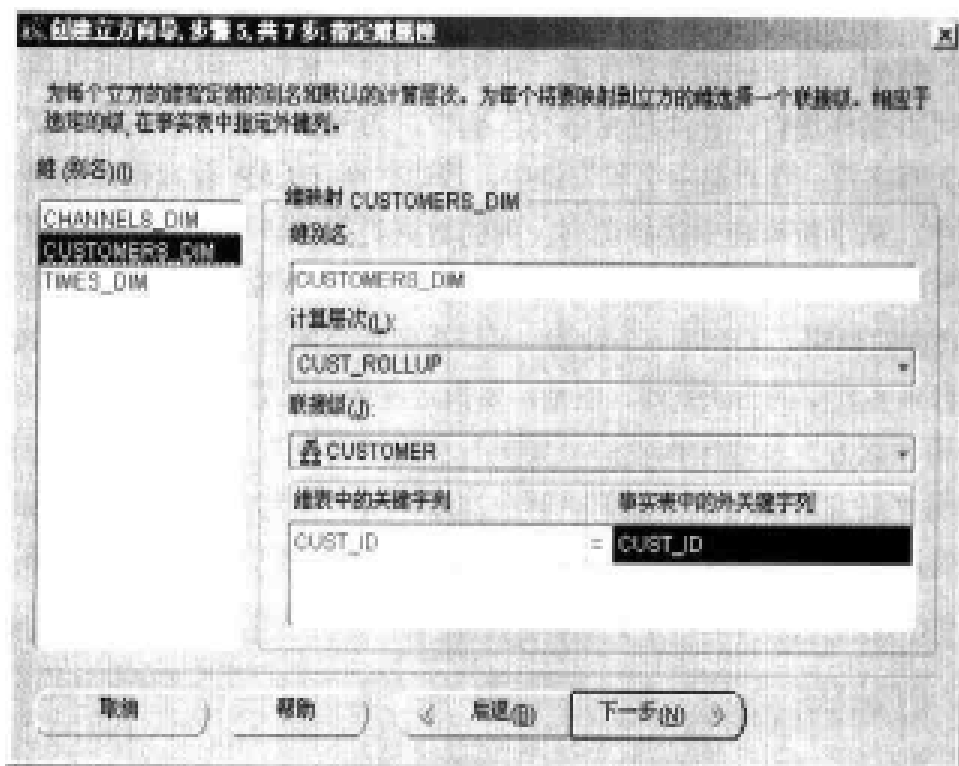


图 2.43 “指定维属性”对话框



图 2.44 “指定度量”对话框

在“指定度量”对话框中可以指定立方包含的度量，一般每个立方必须有一个度量。默认情况下，事实表中数据类型为 NUMBER 的所有列均作为度量在“度量”框中列出。单击“新建”按钮，可以创建新的度量。单击“删除”按钮，可以删除度量。度量的属性指定操作可以这样进行：在“度量名称”框中，为度量提供一个名称，默认情况下度量名称使用大写字母；在“显示名称”框中，提供一个 OLAP 管理使用的显示名称；在“源列”框中，从下拉列表中选择源列；列的数据类型将显示在“数据类型”框中，不能选择数据类型是 BLOB, CLOB, NCLOB, RAW 或 LONG RAW 的列；在“说明”框中，输入对立方的说明。基于同一个源列，可以指定多个度量，例如 Sales 和 Percent_Sales 便可同时来自事实表中的 Sales 列。但是，要指定度量的聚集方法，必须在立方向导成功完成后，使用“立方”属性工作表的“聚集”页或“度量”属性工作表。在创建立方之后，可以使用“立方”属性工作表的“解决次序”页，指定对度量运行查询时执行计算的次序。在确定度量以后，单击“下一步”按钮，进入“概要”对话框（见图 2.45）。

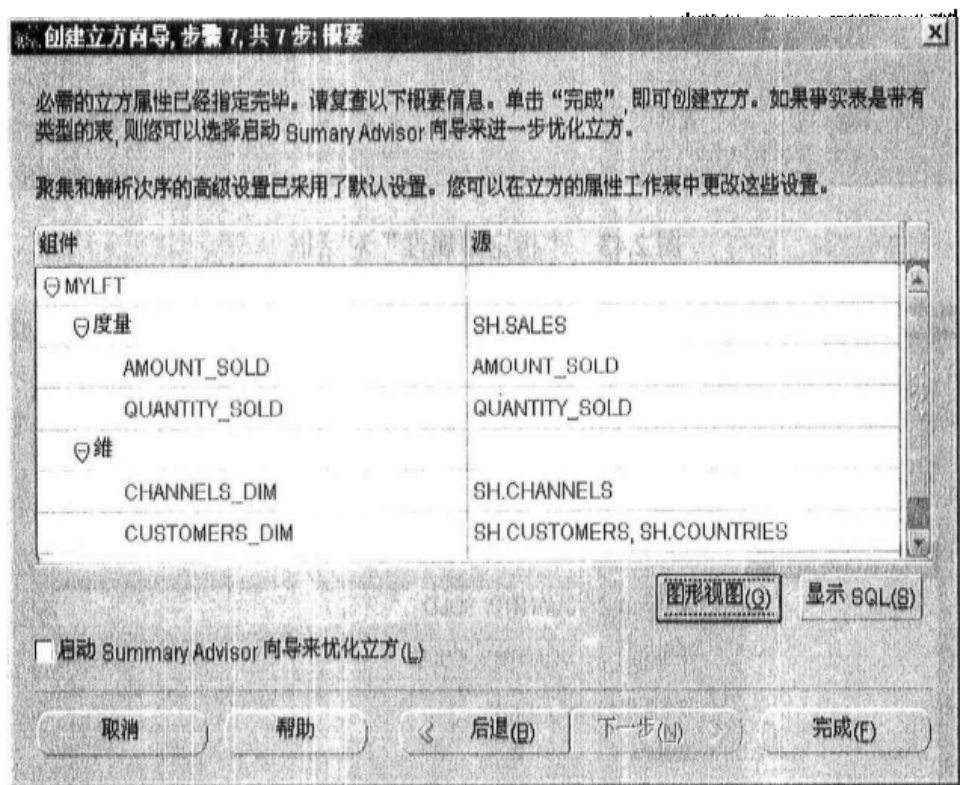


图 2.45 “概要”对话框

在“概要”对话框中可以查看有关立方的概要信息。其中立方的度量和维在“组件”框中列出，每个度量的源列和每个维的源维表在“源”框中列出。单击“图形视图”按钮，可用图形显示方式查看立方拓扑结构（见图 2.46）。单击“显示 SQL”，则可查看为创建立方体所生成的 SQL 语句（见图 2.47）。如果希望生成实体化视图以优化方式对立方运行查询，可以选择“启动 Summary Advisor 向导”选择框，使用 Summary Advisor 向导优化立方。单击“完成”按钮，即可创建该立方。创建立方之后，还可使用“立方”

属性页指定其度量的聚集方法，并且解决其维的次序以及使用 Cube Viewer 查看与立方相关联的数据。

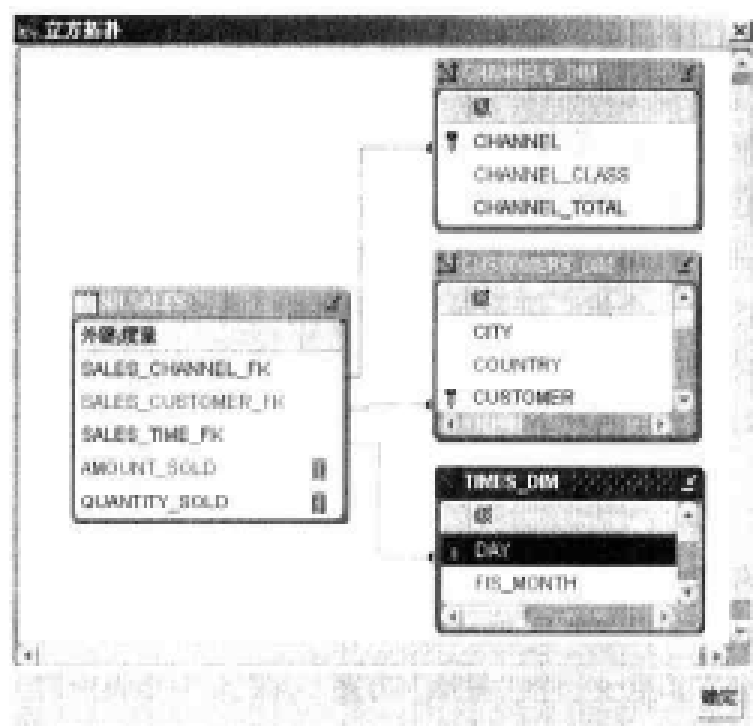


图 2.46 立方拓扑结构

```

declare CHANNELS_DIM number;
CUSTOMERS_DIM number;
TIMES_DIM number;
tmp number;
begin
CWM_OLAP_CUBE.Create_Cube(SYS, MYLFT, MYLFT, 1);
CHANNELS_DIM = CWM_OLAP_CUBE.Add_Dimension(SYS, MYLFT, 'CH');
CWM_OLAP_CUBE.Set_Default_Calc_Hierarchy(SYS, MYLFT, 'CH');
CWM_OLAP_CUBE.Map_Cube(SYS, MYLFT, 'SALES', 'SALES');
CUSTOMERS_DIM = CWM_OLAP_CUBE.Add_Dimension(SYS, MYLFT, 'CU');
CWM_OLAP_CUBE.Set_Default_Calc_Hierarchy(SYS, MYLFT, 'CU');
CWM_OLAP_CUBE.Map_Cube(SYS, MYLFT, 'SALES', 'SALES');
TIMES_DIM = CWM_OLAP_CUBE.Add_Dimension(SYS, MYLFT, 'T');
CWM_OLAP_CUBE.Set_Default_Calc_Hierarchy(SYS, MYLFT, 'T');

```

图 2.47 为创建立方体生成的 SQL 语句

2.4 Oracle 数据仓库的应用工具简介

2.4.1 Oracle 数据仓库的 OLAP 应用

1. Oracle 的 OLAP 应用

Oracle 9i 通过 OLAP 服务提供集成的数据仓库支持，以及存储在数据库内部的 OLAP

元数据。若要管理 OLAP 服务，需要右键单击 OLAP 文件夹，启动 Oracle OLAP Instance Manager。

如果需要管理 OLAP 服务，右键单击 OLAP 文件夹，然后启动 Oracle OLAP Services Instance Manager。使用该工具可以启动、停止和暂停 OLAP 服务，创建新的服务和移去现有的服务，更改配置参数，监视客户机会话和查看系统日志。

OLAP 元数据包括维、立方和度量文件夹 3 个组成部分。维对象将数据仓库维表组织成级、属性和层次。可用向导或者属性工作表定义“维”。立方是多维数据的逻辑表示，而且必须至少包含一个维。可用向导或者属性工作表来定义“立方”。度量文件夹将度量组织成可管理的组。可用属性工作表定义“度量文件夹”，并且置入从立方派生的度量。为获得最佳性能，可用 Summary Advisor 向导建议并且创建“立方”对象的实体化视图。

2. Summary Advisor 的使用

Summary Advisor 前提条件是 Summary Advisor 向导运行于“星形”方案/量纲模型上。假定存在数据仓库和至少一个“星形”方案，即在一组维表中的一个事实表。此外，必须满足以下前提条件才能运行 Summary Advisor 向导：对事实表中的每个维关键字，外关键字约束条件必须指定哪个维关键字引用维表中的主关键字列；必须为维表定义维对象；Summary Advisor 向导使用包含在维对象中的层次元数据，确定应该创建哪些概要。要求结构统计信息的有：要分析的所有事实表，与这些事实表链接的所有维表以及在这些事实表上所建的任何现有实体化视图；表级和列级别的统计信息。对于 RECOMMEND_MV_W(工作量)，必须存在工作量统计信息。这些统计信息由 Oracle Trace 收集。如果没有找到外关键字约束条件、维对象或结构统计信息，Summary Advisor 向导将不提出建议案。

2.4.2 Oracle 数据仓库的数据挖掘应用

Oracle 9i 的数据挖掘工作主要由 Oracle 9i Data Mining 完成。Oracle 9i Data Mining 是 Oracle 9i 数据库企业版的一个附加选择，它为数据库分类、预测及关联提供了数据挖掘功能，在实现中需要通过基于 Java 的 API（应用编程界面）实现挖掘模型的建立和使用。

Oracle 9i Data Mining 将数据挖掘功能内嵌入 Oracle 9i 数据库，使数据准备、模型建立和数据挖掘及模型评分（data mining & scoring）活动都保留在数据库内进行。这样就不必将海量数据卸载到外部专用分析数据库，进行数据挖掘和模型评分分析。

由于 Oracle 9i 的可伸缩性使 Oracle 9i Data Mining 能够分析大量的数据并且侦察到其中的微妙模式和关系。可以使企业建立由数据挖掘结果所驱动的商业应用，为企业的商

务智能应用提供非常理想的基础架构。由于数据挖掘是与数据库直接联系在一起进行的,这就使数据挖掘的结果更加切实可行。数据挖掘的结果,可用 Oracle Discover 表示给用户。

1. 利用 Oracle 9i Data Mining 增强企业对客户的预测与观察能力

Oracle 9i Data Mining 使企业能够系统地提取、集成其所经营范围的商业信息,开发人员能够利用基于 Java 的 API 来增加数据挖掘的观察能力和预测能力,增强数据挖掘在客户关系管理(CRM)、企业资源计划(ERP)等方面的应用。例如,利用 Oracle 9i Data Mining 可使企业预测客户的流失,识别可能流失的客户,发现那些对报价最可能做出响应的、可能成为盈利的客户。

利用 Oracle 9i Data Mining 对数据进行挖掘并且建立预测模型以后,Oracle 9i Data Mining 就可利用这些模型对其他数据进行评分和预测。例如,对某个客户的历史数据以及某次响应按照这种方法进行评分,即可以对客户的喜好进行评价,并且制定具有个性化的交叉销售方案。

2. 利用 Oracle 9i Data Mining 进行预测和分类

Oracle 9i Data Mining 的预测方法主要是 Naive Bayes 数据挖掘算法,利用该算法可以进行预测和分类。通过查找数据中所存在的模式——客户的过去状况作为客户未来的预测算子,对客户未来行为进行预测。典型的预测应用可以估计某个结果的可能性,例如,“0, 1”或“是,否”或“A, B, C, D”等方式进行。

预测模型的结果可以组合使用,从而提供更有价值的商务信息。例如,利用 Oracle 9i Data Mining 分别建立客户合作期间价值的模型(LTV)和客户流失模型(CRO),将这两个预测模型的预测结果的概率值相乘($P(LTV) \times P(CRO)$),就可获得关于如何编制营销预算的有价值的参考。

3. 利用 Oracle 9i Data Mining 进行关联处理

在 Oracle 9i Data Mining 中还提供关联规则(Association Rules)数据挖掘算法,利用这个算法可以挖掘隐藏在数据里的“相关联”事件或同时发生的事件。这些相关联的事件处理在信息管理中随处可见。例如,在商业企业中经营管理人员需要了解某个客户在什么情况下,最可能购买什么产品?在生产制造企业中,产品质量管理人员需要了解哪些生产设备与产品的质量存在关联?在医疗机构中,医务人员需要了解某种疾病的疗效与病人的状况、药物的属性存在哪些相关的因素?

用户使用 Oracle 9i Data Mining 的目的在于了解客户的进一步需求,使企业尽快地为客户需求提前做好准备,或使自己的工作取得更好的效果。

4. 基于 Java API 的 Oracle 9i Data Mining

数据仓库的应用开发人员需要通过基于 Java 的 API 才能访问 Oracle 9i Data Mining 的功能。在对 Oracle 9i Data Mining 的访问过程中需要涉及数据准备、模型建立、模型评分操作，这些操作可以通过 Oracle Discover 进行。

Oracle Discover 的性能具有：检索部分结构；使用数组接口，获取多行数据；选择自动或者人工执行查询；Oracle Discover 的管理；快速大量加载数据库表和视图；访问 Designer/2000 中的设计信息；使用本地数据安全措施，控制对数据对象的访问；为逻辑分组信息确定商业区域；建立复杂的文件夹，用于建立数据视图，以及对最终用户隐藏复杂性；建立和维护自动关联条件和深入关系；维护缺省文件夹名、项名称、头标志和缺省格式；为各项自动建立值列表；建立和自动维护汇总表；使所有用户都能访问 Workbook。

Oracle Discover 的界面特点有：用 C++ 开发，为 Windows 95 和 Windows NT 而设计，为随意查询、报告生成、图表生成、深入分析和 Web 公布提供简单界面；通过 Wizard 界面建立随意查询，管理最终用户层；具有联机帮助、提示卡和交互式功能；且有针对零售/饭店、电信、医疗保健/政府、银行业、金融等行业的教程。

Oracle Discover 还具有强大的查询功能，可以使用图形查询生成器；自动识别相关数据；使用逻辑操作符对条件进行组合；自动进行数据分组；对相关数据自动进行关联（相等、不等、自关联和外部关联）；建立已计算项（用户定义的表达式）；支持各种统计计算，例如，平均、最小、最大、总和、计数、标准偏差、方差和百分比等；在查询执行之前，可以预测查询所需时间；自动汇总重定向，并且具有 Web 汇总功能。



本章小结

Oracle 数据仓库开发应用工具主要有技术基础工具、分析应用工具、数据仓库创建工具和数据库维护工具四大类。Oracle 数据仓库的创建主要是构建数据仓库数据库、数据库表空间和数据库表。在创建数据仓库数据库时可以利用 Oracle 数据库构造助手进行，在构建过程中要在数据库的模板中选择“Data Warehouse”选项，并且确定数据库的标识、初始参数和存储参数。

在完成数据仓库数据库的创建后，需要创建表空间，然后在表空间中创建表。表空间的创建可以在 Oracle 的企业管理器支持下进行。Oracle 的表空间分系统表空间和应用表空间两部分。前者已经在 Oracle 安装之时设置完毕，用户只需要进行应用表空间的设置。在表空间的设置中需要确定表空间的一般信息和存储空间的设置。

完成表空间设置后,就可以在表空间中创建表。表的创建也可以在 Oracle 的企业管理器中实现。在创建表的过程中,需要确定表的名称、方案、表空间以及列的属性、表的存储属性、表的分区索引设置。

完成 Oracle 的表创建后,就可以在企业管理器中为表创建维与立方。维的创建需要确定维的类型、维的名称与方案、维的级别、维的层次、维的层次间连接关系以及与维相连接的 OLAP 信息。在立方体的创建中,需要确定维的一般信息、事实表、维以及维的属性和立方中的度量值等。

在完成 oracle 的数据仓库数据库、表空间、表、维、立方的创建后,Oracle 的数据仓库创建已经基本结束,可以对数据仓库进行有关的操作了。



习题

- 2-1 Oracle 中的数据仓库技术包含哪几种?
- 2-2 Oracle 的数据仓库数据库如何创建?
- 2-3 为什么要创建表空间,在 Oracle 中怎样筹建表空间?
- 2-4 数据仓库的维在数据仓库中可以发挥什么作用?如何利用 Oracle 工具创建维?
- 2-5 数据仓库的立方在数据仓库中可以发挥什么作用?如何利用 Oracle 工具创建立方?



第 3 章

SQL Server 的数据仓库设计与使用

引 言

Microsoft SQL Server 2000 (以下简称 SQL Server) 是微软公司近年推出的一个中型的高性能关系型数据库管理系统, 其中包含大量的数据仓库创建、操作、数据清理、数据加载、数据使用和数据挖掘工具。由于 SQL Server 与 Windows 的捆绑特性以及其高性能, 许多用户在数据仓库的开发中, 都用 SQL Server 作为数据仓库的开发工具, 并且取得良好的效果。

通过本章学习, 可以掌握:

- ◆ SQL Server 数据仓库创建工具的使用
- ◆ SQL Server 数据仓库数据加载工具的使用
- ◆ SQL Server 数据仓库数据操作工具的使用
- ◆ SQL Server 数据仓库的数据挖掘工具使用
- ◆ SQL Server 数据仓库开发工具的数据仓库实际开发

3.1 SQL Server 数据仓库开发工具及应用

SQL Server 中提供的数据库设计、建立、数据加载、数据使用和数据挖掘的工具可以实现对数据库进行创建、操作、管理与应用的支持（见表 3-1 的 SQL 的数据仓库开发工具）。

表 3-1 SQL 的数据仓库开发工具

数据仓库工具名称	在数据仓库中的作用
关系型数据库	数据仓库的创建和维护
数据转换工具	数据仓库的数据加载
数据复制工具	分布式数据仓库的数据发布、加载
OLE DB	应用系统与数据源的接口
Analysis Services	数据挖掘与分析
English Query	数据仓库的语言查询
Meta Data Services	数据仓库的元数据浏览
PivotTable	客户端多维数据的定制与操作

当数据仓库完成物理模型设计以后，就需要进行数据仓库的物理创建。此时，需要完成这样一些工作：创建数据准备区、创建数据仓库、从业务系统提取数据、清理和转换数据、将数据加载进数据仓库、将数据发布到数据集市。在创建了数据仓库后，用 SQL 查询、OLAP 应用、数据挖掘、Web 访问等工具对数据仓库进行操作和访问。

1. 创建数据准备区

为能顺利地对准备装入数据仓库的数据进行析取、清理和转换，在数据仓库中需要创建一些单独的数据库或在数据仓库数据库中创建一些项目作为数据准备区。在数据准备区中可以从数据源中析取数据，将数据转换为数据仓库常用格式，检查其一致性和引用完整性，并且准备装入数据仓库数据库。数据准备区的创建可以采用 SQL Server 中的数据库创建与表创建工具实现。

2. 创建数据仓库

数据仓库的框架通常由事实表和一些维表组成，且在所有表中的主要字段上建立索引。这可以用 SQL Server 中的数据库创建工具与表创建工具实现。

3. 从业务系统提取数据

数据仓库的数据必须从包含数据源的业务处理系统中获取，在获取过程中需要对数据源数据进行抽取。SQL Server 中的数据抽取工具主要有 Transact-SQL、分布式查询、DTS、

命令行应用程序、bcp 实用工具、从文本文件加载的 BULK Insert 语句和 ActiveX 脚本。

4. 清理和转换数据

数据从业务系统抽取来后，还需要对数据进行数据一致性的协调、格式化处理等清理工作，且对数据进行必要的转换。SQL Server 提供 Transact-SQL 查询、DTS 包、命令行应用程序、ActiveX 脚本等工具以完成这些工作。

5. 将数据加载进数据仓库

数据完成清理和转换后，可以加载进数据仓库。为此，SQL Server 提供 Transact-SQL、DTS 和 bcp 实用工具。

6. 将数据发布到数据集市

数据仓库的用户在使用数据仓库时，所面对的是不同的数据集市。SQL Server 提供数据复制技术来完成数据集市的初始装载，并且提供各种数据加载工具实现对数据集市的的数据加载。

7. SQL 查询

SQL 查询的实现往往需要数据库专家来实现，经常与定期执行的预定义报表一起使用。SQL Server 提供 Transact-SQL 来实现 SQL 查询。

8. OLAP 应用

OLAP 提供对数据仓库数据进行快速访问的技术，SQL Server 提供 Analysis Services 创建和管理 OLAP 应用。

9. 数据挖掘

数据挖掘技术是数据仓库的一个重要应用，SQL Server 在 Analysis Services 中提供了数据挖掘技术的创建和管理功能。

10. Web 访问

数据仓库的 Web 应用越来越得到用户的欢迎，SQL Server 所提供的 Analysis Services、English Query 与因特网信息服务（IIS）一起，可用多种方法在 Web 上对数据仓库进行查询与更新。

11. 更新数据仓库数据

数据仓库在实际应用中还要定期进行数据更新和维护，这些工作可以依靠 SQL Server

的 Transact-SQL, DTS 和 bcp 实用工具完成。

3.2 SQL Server 的数据仓库创建

数据仓库的物理创建是依据数据仓库设计阶段所确定的数据仓库物理模型构造数据库, 在 MS SQL Server 中只能以服务器成员的角色或获得授权, 才能创建数据库。

3.2.1 创建数据库

在 SQL Server 中创建数据库的方法有多种, 例如企业管理器、建库向导、建库语句等。这里只介绍用企业管理器创建数据库的方法, 其他方法可以参考有关资料。

用 SQL Server 企业管理器控制台创建数据库时, 首先要在 Windows 窗口中执行“程序”→“Microsoft SQL Server”→“企业管理器”菜单项命令, 启动企业管理器控制台。依次单击“Microsoft SQL Server”→“SQL Server 组”等左边的“+”号, 直至出现以文件夹图标表示的“数据库”节点(参见图 3.1), 然后用图 3.1 中的方法建立数据库。此时, 将出现一个“数据库属性”设置对话框(见图 3.2)。该对话框中有常规和数据文件、事务日志 3 个标签页。

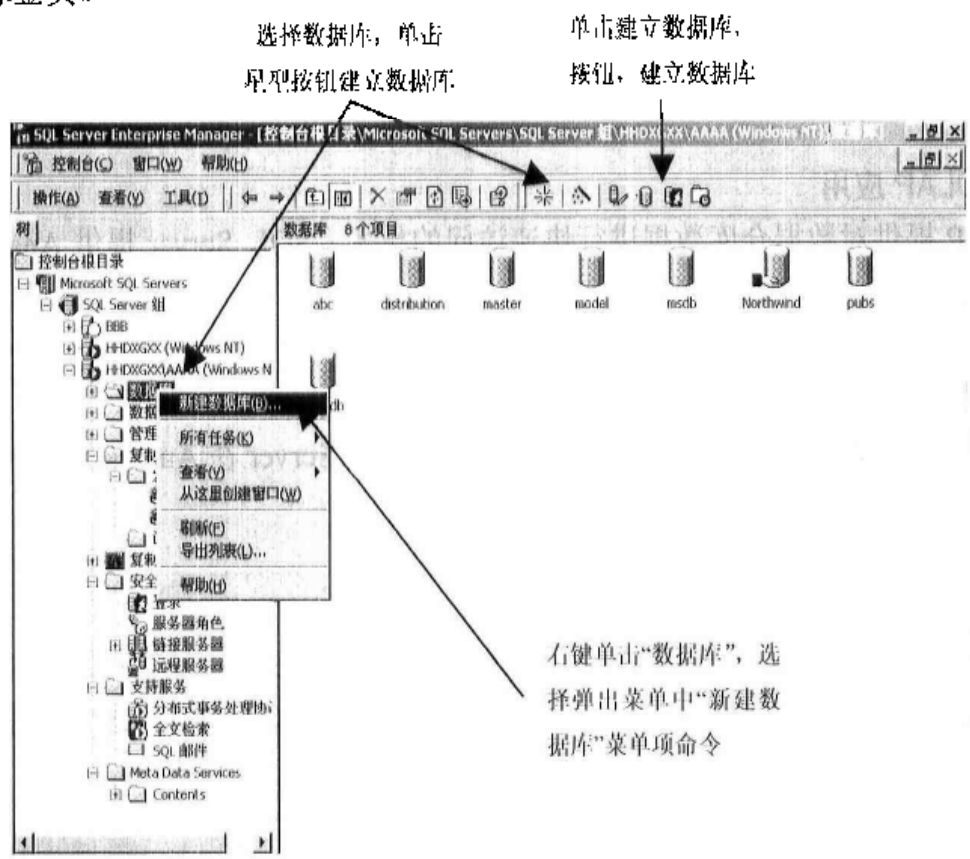


图 3.1 用 Enterprise Manager 建立数据库

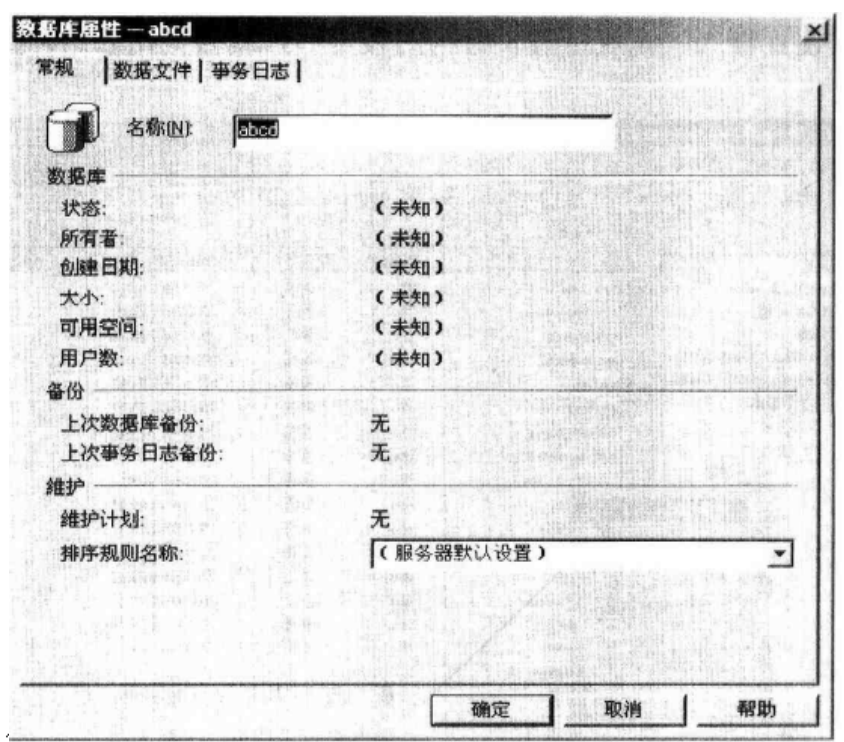


图 3.2 “数据库属性”设置对话框

在常规标签页中的名称文本框中输入数据库名称。

在数据文件标签页的列表框中的位置列中输入文件所存放处，必要时可以单击其左边的省略号按钮，进行存放地点选择，并且设置文件初始大小、文件组名称等属性。在该页下部的文件属性部分有选择文件自动增长选项。如果选择自动增长，还需确定文件增长方式：按字节还是按百分比增长。且对最大文件大小进行选择：文件增长不受限制或设置文件增长最大值。

在事务日志标签页中，所具有的事务日志文件设置信息与数据文件标签页中的设置一样，只是设置对象是数据库的日志文件。

在完成这些属性设置后，单击“确定”按钮，就可成功创建数据库。

3.2.2 创建表

在完成数据库的建立后，还要按照在数据仓库设计中所确定的数据模型确定数据仓库的基本结构——事实表或维表。

用 SQL Server Enterprise Manager 创建表时，首先要在 SQL Server 中打开准备创建表的服务器，打开“数据库”节点，选择需要在其中创建表的数据库且将其图标展开，右键单击“表”节点，弹出快捷菜单，选择“新建表”快捷命令（见图 3.3）。调出表结构输入对话框（见图 3.4）。输入表的列（字段）名、数据类型、长度、值是否允许空、列（字段）的描述、默认值、精度和小数位数等。这些值均根据数据仓库设计中的事实表

模型和维表模型设置输入。完成这些表结构的设置后,就可关闭表结构输入窗口,结束表的创建。

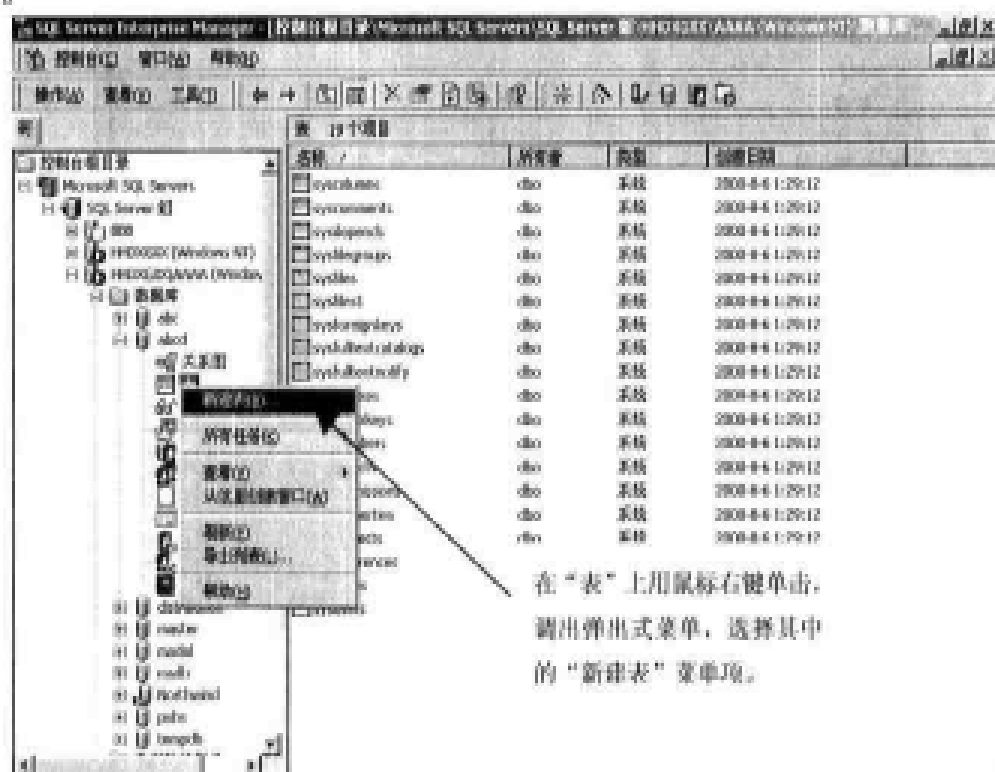


图 3.3 “新建表”快捷命令

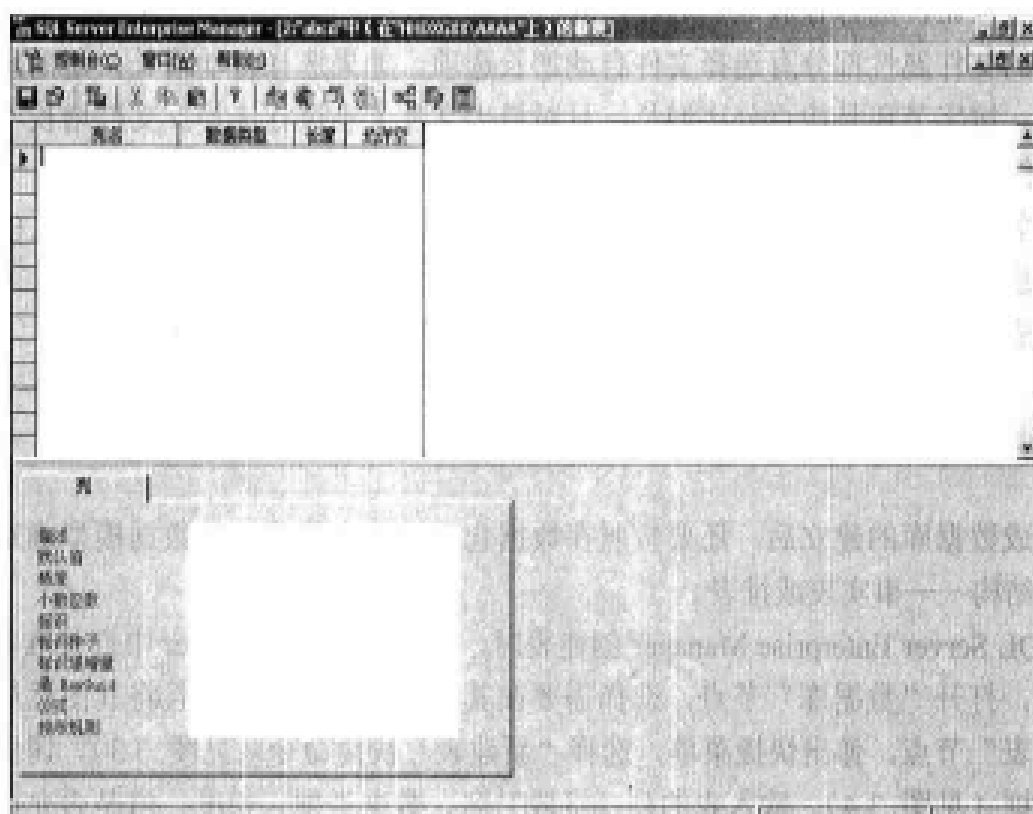


图 3.4 表结构输入对话框

3.3 SQL Server 中的数据仓库访问与操纵

在数据仓库的访问中除了访问事实表以外，用户还要经常访问多维数据集。因此，多维数据集的建立是数据仓库创建中的必不可少部分。Analysis Services 系统是一个管理多维数据集的有力工具，可以创建用于数据仓库数据访问、分析多维数据集和知识发现的数据挖掘模型。在使用 Analysis Manager 以前，须从 SQL Server 的安装光盘上将其安装到机器上。

3.3.1 Analysis Manager 数据库的创建与数据源的确定

在用 Analysis Services 创建数据多维数据集并且确定数据源后，用户就可以有效地访问数据仓库中的数据。

1. Analysis Manager 数据库的创建

在 Analysis Manager 中所创建的数据库是一个虚拟的用于存放 OLAP 服务结构（多维数据集、维度等）的对象。创建数据库时，在 Analysis Manager 控制台的树形结构中右键单击服务器，从弹出的菜单项中选择“新建数据库”（参见图 3.5），调出“数据库”对话框。在“数据库名称”文本框中输入数据库的名称，例如，Foodmart 2000。在“描述”文本框中输入数据库的描述说明。最后，单击“确定”按钮，完成数据库的创建。在这个数据库中包含数据源、多维数据集、共享维度、挖掘模型和数据库角色 5 个对象。

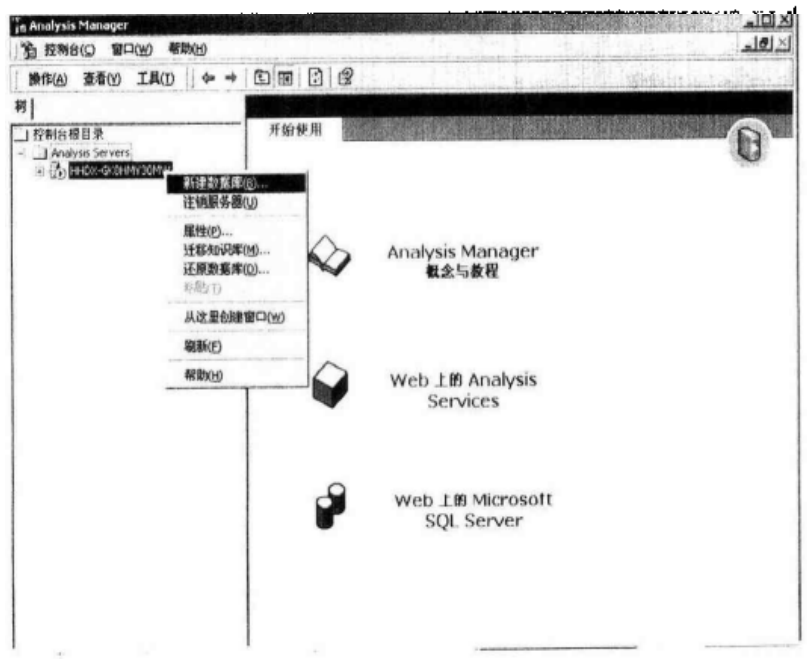


图 3.5 在 Analysis Manager 控制台上创建数据库

2. Analysis Manager 中 ODBC 数据源的确定

在完成 Analysis Manager 数据库的创建后,不必为其存入数据,但是需要为其指定数据源。Analysis Manager 提供一个 OLE DB Provider 数据源清单,用于数据源的指定。其中应用较广泛的数据源有 Microsoft OLE DB Provider for ODBC Drivers 和 Microsoft OLE DB Provider for SQL Server。

指定数据源时,首先在 Analysis Manager 控制台的树形结构中选中需要指定数据源的数据库节点,例如 FoodMart 2000。右键单击“数据源”,弹出快捷菜单,选择其中的“新数据源”菜单项(参见图 3.6)。进入“数据链接属性”对话框(见图 3.7)。窗口中有“提供者”、“连接”、“高级”和“所有”4 个标签页。

首先在“提供者”标签页上的“选择您希望连接的数据”列表框中选择 Microsoft OLE DB Provider for ODBC Drivers 选项,然后单击“下一步”按钮,进入“连接”标签页。

“连接”标签页的内容与数据源类型的选择有关。当选择 Microsoft OLE DB Provider for ODBC Drivers 类型数据源后,“连接”标签页上有 3 个部分:指定数据源部分、输入登录服务器的信息部分和输入要使用的初始目录部分。

如果在指定数据源部分选择“使用数据源名称”选项,可以在下拉列表框中选择系统现有的 ODBC 数据源;如果选择“使用连接字符串”选项,就可以单击“编译”按钮重新创建一个 ODBC 数据源。

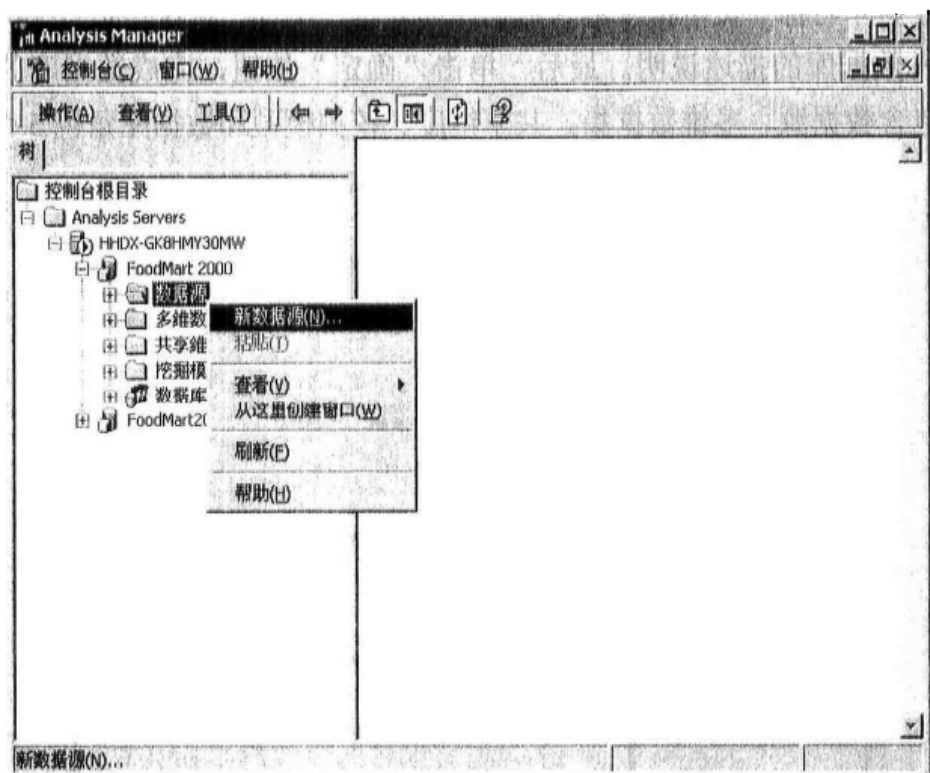


图 3.6 指定数据源

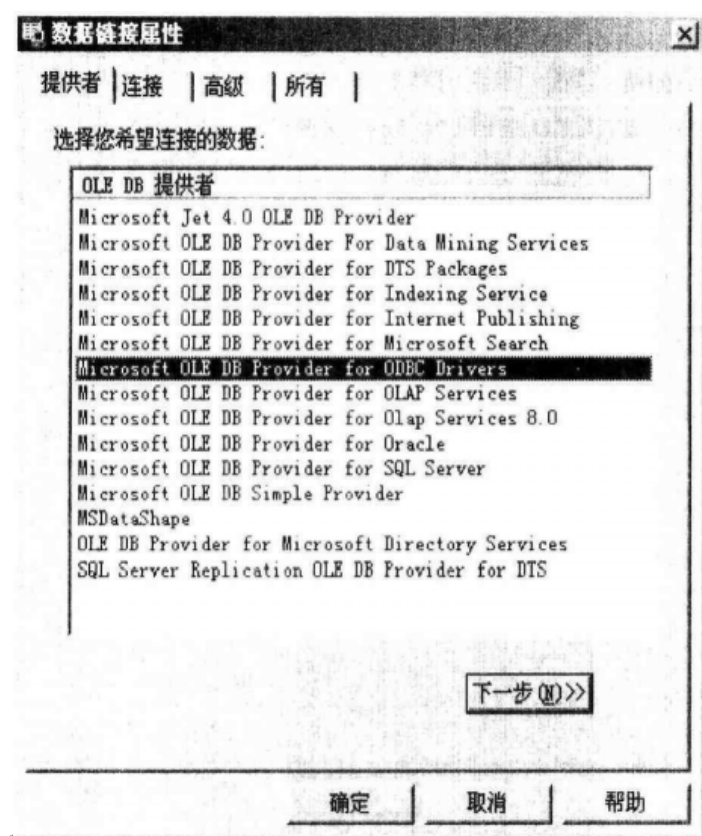


图 3.7 “数据链接属性”对话框

在数据源要求用户在连接时给出用户名和口令时，就需要在输入登录服务器的信息部分的“用户名称”文本框中输入用户名，“口令”文本框中输入口令。复选框“空白密码”用于禁止输入口令，“允许保存密码”用于使机器保存用户口令。

在“输入要使用的初始目录”下拉列表框中选择相应的位置。完成这些设置后，可以单击“测试连接”按钮，测试连接是否成功，测试结果将以对话框方式告知。

高级标签页用于设置一些连接数据源的高级选项。其中的网络设置部分，用于设置用户连接网络的网络安全等级。其他部分的连接超时设定文本框设置连接超时的时间（秒），访问权限列表框设置连接数据源的只读（Read）、读写（ReadWrite）、可共享（ShareDenyNone）、除读以外的共享（ShareDenyRead）、除写以外的共享（ShareDenyWrite）、其他共享（ShareExclusive）和可写（Write）7种权限。

在所有标签页中，用一个列表框显示前面所进行的设置内容，如果对设置不满意，可以单击“编辑值”按钮，编辑这里的设置。完成设置后，可以单击“确定”按钮，完成数据源指定操作。

3. Analysis Manager 中 SQL Server 数据源的确定

如果在数据链接属性对话框的提供者标签页中选择 Microsoft OLE DB Provider for SQL Server，单击“下一步”按钮后，进入“数据链接”标签页（见图 3.8）。

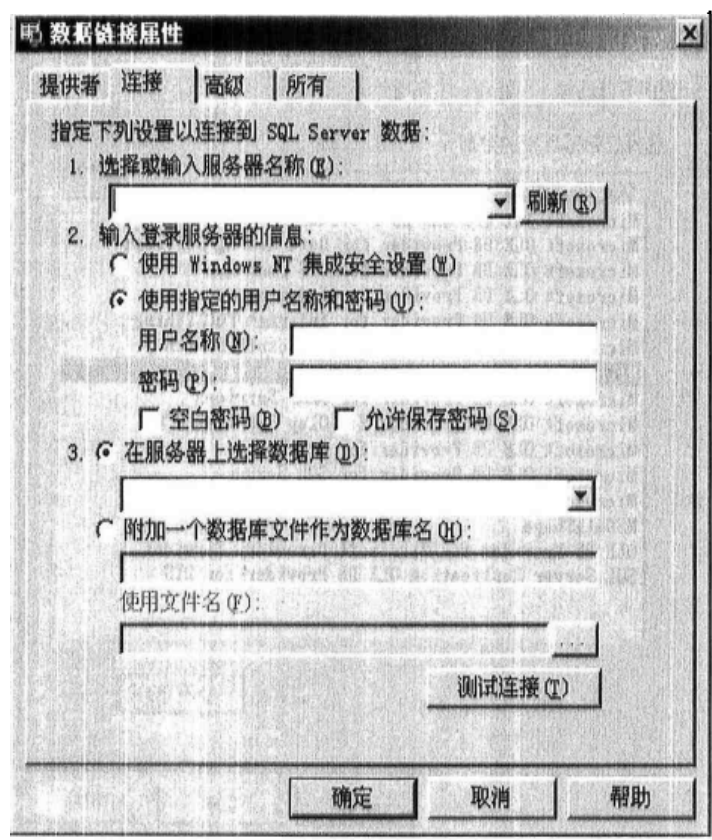


图 3.8 “数据链接”标签页

在“数据链接”标签页中的“选择或输入服务器名称”下拉列表框中指定将要连接的服务器名称。

确定“输入登录服务器的信息”选项。如果选择“使用 Windows NT 集成安全设置”选项，则意味着所有 Windows NT 用户都可直接登录到 SQL Server 系统中，不需要再次认证。如果选择 SQL Server 认证模式，则要选择“使用指定的用户名称和密码”选项。然后在“用户名称”文本框中输入 SQL Server 的登录名称，在“密码”文本框中输入用户的口令。选择“空白密码”复选框，将不需要输入口令。选择“允许保存密码”复选框，将允许系统保存口令，下次用户登录就不必重新输入口令。

选择将要使用的数据库。选中“在服务器上选择数据库”选项，就可以从下拉列表框中选择将要连接的数据库。如果选择“附加一个数据库文件作为数据库名”选项，就可以在文本框中输入需要连接的数据库名称，且可以在“使用文件名”文本框中输入将要连接的表名称。

其他标签页上的选项设定同 ODBC 数据源指定过程基本一致，这里不再复述。

3.3.2 用 Analysis Services 创建维

在确定数据源以后，就可以创建数据维了。用 Analysis Services 创建维的过程如下。

1. 调出维度向导欢迎对话框

在 Analysis Manager 控制台左边的树形结构中, 依次单击服务器节点、数据库 FoodMart2000 节点。右键单击“共享维度”节点, 调出包含“新建维度”的弹出式菜单, 可以从其下级菜单项中选择“向导”和“编辑器”菜单项(参见图 3.9)。这里选择“向导”菜单项, 调出维度向导欢迎对话框。

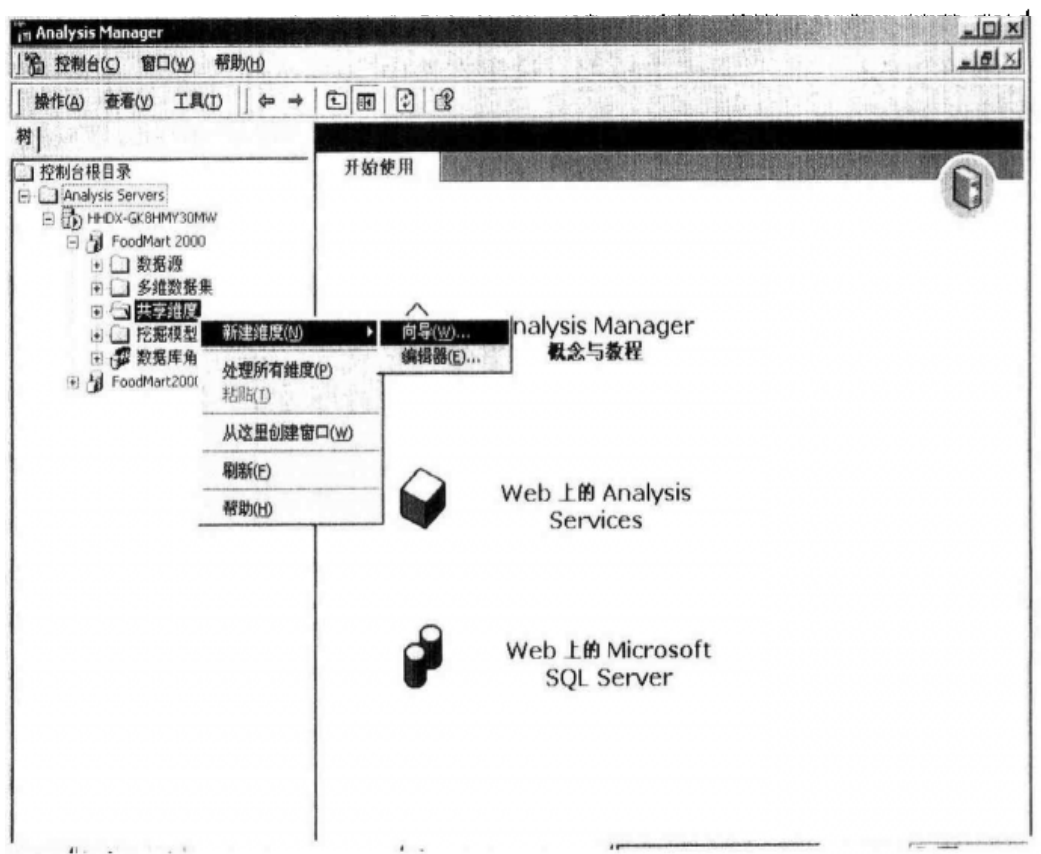


图 3.9 Analysis Services 创建维示意图

2. 调出“选择维度的创建方式”对话框

在维度向导欢迎对话框中单击“下一步”, 调出“选择维度的创建方式”对话框。该对话框中包含星型架构、雪花架构、父子维度、虚拟维度、挖掘模型 5 种维结构选项。

星型架构可从一个表中选择一列或几列, 每列都可作为维的一个层次。如果选择多个列, 列就要有一种逐渐变化的信息。例如, 可以选择 Country, Province, City, Store 作为 Store 维的层次; 雪花架构可以创建多个相关的维表, 从多个维表中可以选择一个或多个列, 每列都可作为维的一个层次; 父子维度要从表中选两列, 一列为维成员, 另一列为父成员。例如, 用 Employee 和 Manager 可以创建一个 Organization 维; 虚拟维度中的维成员来自另一个维中的成员, 数据在运行时才计算, 不占用磁盘空间; 挖掘模型要

用一个列和一种数据挖掘工具来构造。这里选择星型架构后，单击“下一步”按钮，进入“选择维度表”对话框。

3. “选择维度表”对话框

在“选择维度表”对话框中有一个“可用的表”列表框和“详细信息”列表框，前者列出当前数据库中的所有数据源，可以从中指定数据源和其中的表作为维表。如果没有数据源或数据源不满足需要，可以单击“新数据源”按钮新建一个数据源。选定表后，可以单击“浏览数据”按钮显示表中数据。这里在选中 `customer` 表后，单击“下一步”按钮，进入“选择维度类型”对话框。

4. “选择维度类型”对话框

在“选择维度类型”对话框中有“标准维度”和“时间维度”两个选项。选择“标准维度”，就在选择的维表中指定一列或若干列为维的成员。选择“时间维度”，就可以从下拉列表选择一个日期列定义维表。这里选择了“标准维度”后，单击“下一步”，进入“选择维度的级别”对话框。

5. “选择维度的级别”对话框

在“选择维度的级别”对话框中，有“可用的列”和“维度级别”两个列表框。用向右箭头可将“可用的列”中选定的列移到“维度级别”中，并且可用上下箭头将所选中的列排定正确的层次——从父层到子层。“自动对级别成员计数”选择项可对所选级别的成员进行计数，进行多维数据集处理时需要此计数。处理拥有大量成员的级别可能需要花些时间。若要加快维度的创建过程，可以取消选择“自动对级别成员计数”。如果关闭了级别成员的自动计数，就必须使用维度编辑器的属性窗格手工输入一个估计的级别数，或者在多维数据集的处理过程中对个别提示进行相应操作。在选择移动 `country`, `state_province`, `city`, `lname` 列后，单击“下一步”按钮，进入“指定成员键列”对话框。

6. “指定成员键列”对话框

在“指定成员键列”对话框中，提供包含默认关键字列的“级别”列表框。若要修改某列关键字，先在“名称”列中指定列，然后在“成员键列”的下拉列表框中选一个新列为关键字列。最后，单击“下一步”按钮，进入“选择高级选项”对话框。

7. “选择高级选项”对话框

在“选择高级选项”对话框中可以进行关于维的 3 个选择：确定“成员的排序依据和惟一性”的选项；能够指定维表成员的存储位置和分组形式的“存储模式和成员分组”

选项；以后添加或删除维成员时，不必重新处理多维数据集的“可更改维度”选项。这些选项可以同时选中，在选择所有的选择项后，单击“下一步”按钮，进入“指定排序依据和惟一性”对话框。

8. “指定排序依据和惟一性”对话框

在“指定排序依据和惟一性”对话框中，可在对话框的“级别”列表框中的“排序依据”列中，指定维的排序依据，是按名称、键、列中的哪一个排序；在“键惟一”列中，可以指定关键字的惟一性范围，有“在维度中”、“在级别成员中”、“在兄弟表中”三种选择；在“名称惟一”列中，可以指定级别名称的惟一性范围，有“在维度中”、“在级别成员中”、“在兄弟中”和“不惟一”四种选择。在采用默认属性后，单击“下一步”按钮，进入“指定存储模式和成员分组”对话框。

9. “指定存储模式和成员分组”对话框

在“指定存储模式和成员分组”对话框中，可以确定维表的存储模式和成员组。选择“存储为多维 OLAP(MOLAP)”单选框，将按照以优化结构、提高查询能力的 MOLAP 模式存储。还可以选择“为最低级别创建成员组”复选框，创建维成员组。如果选择“存储为关系 OLAP(ROLAP)”单选框，将按照关系型数据库模式进行存储。这里选择 MOLAP 模式存储后，单击“下一步”按钮，进入“设置可更改属性”对话框。

10. “设置可更改属性”对话框

在“设置可更改属性”对话框中有两个选择项。如果确定新维没有更改属性，就选择“否，新建的维度不可更改”项。否则选择“是，新建的维度可更改”项。在确定“否，新建的维度不可更改”选项后，单击“下一步”按钮，进入“维度向导的结束”对话框。

11. 向导结束对话框

在向导结束对话框中，可以确定新建维的名称、维的层次结构名称和预览新建立维的层次结构。在“维度名称”下拉列表框中确定新建维的名称。选择复选框“创建维度的层次结构”后，就可以在“层次结构名称”文本框中输入新建立维的层次结构名称。然后，可以在“预览”列表框中预览维的层次结构。最后，单击“完成”按钮，完成维的创建。

系统将调出“维度编辑器”对话框（见图 3.10），在此窗口中可对刚创建的维度进行编辑，或新建其他维。如果退出此窗口，将返回 Analysis Manager 控制台。

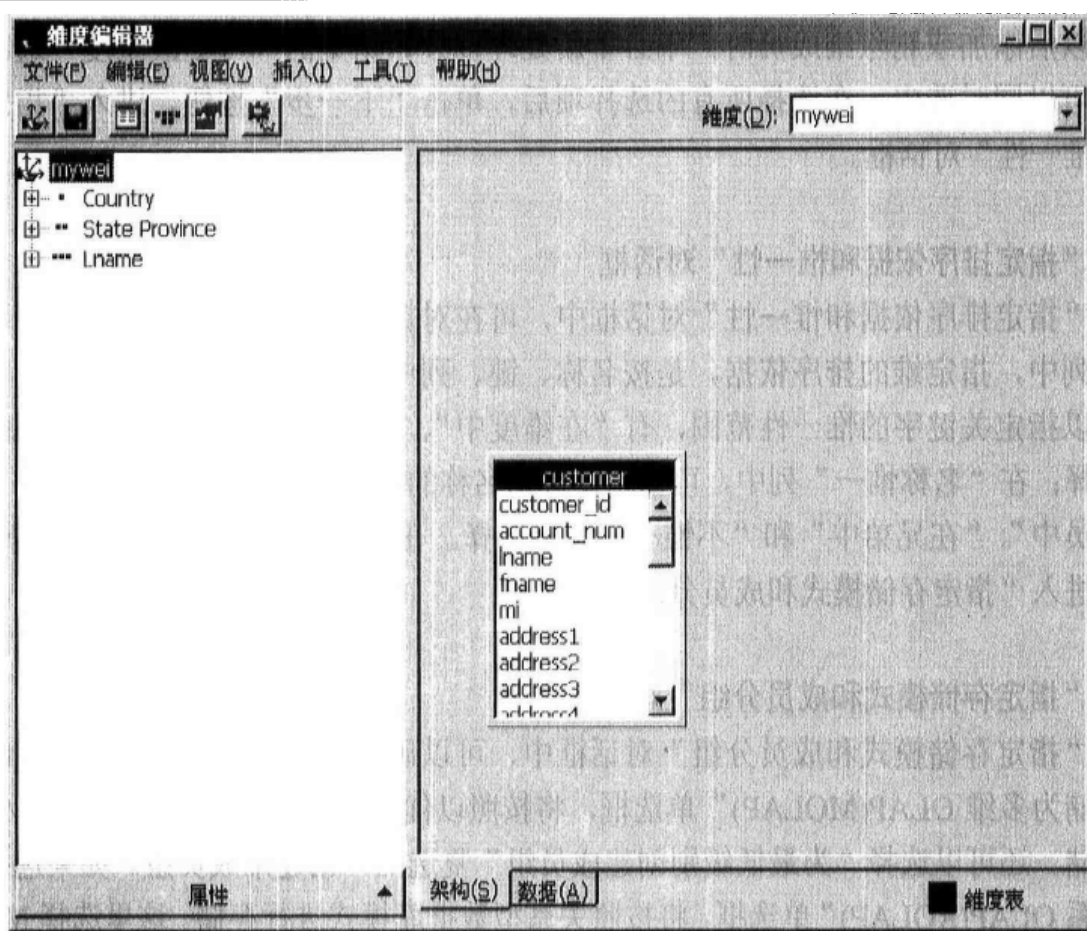


图 3.10 “维度编辑器”对话框

3.3.3 用 Analysis Services 创建多维数据集

在创建维以后，就可进行多维数据集的创建。使用 Analysis Services 中的“多维数据集”向导可以创建正常多维数据集、虚拟多维数据集和链接多维数据集 3 种类型多维数据集。正常多维数据集由事实表和维表组成，构成数据仓库的星型模型。虚拟多维数据集以一种类似视图查看基表的方式查看其他的多维数据集，是一种联结多个多维数据集的虚拟多维数据集。链接多维数据集是基于一个已有的多维数据集而建立的多维数据集。多维数据集的创建也与维创建一样，可用向导或编辑器完成（参见图 3.11）。

1. 进入多维数据集创建

在 Analysis Manager 控制台左边的树状结构中，依次单击服务器节点、数据库节点。右键单击“多维数据集”节点，调出包含“新建多维数据集”、“新建虚拟数据集”和“新建链接数据集”3 个菜单项的弹出式菜单。这里选择“新建多维数据集”→“向导”后，调出“多维数据集向导”欢迎对话框。在欢迎窗口中单击“下一步”按钮，进入“从数据源中选择事实数据表”对话框。

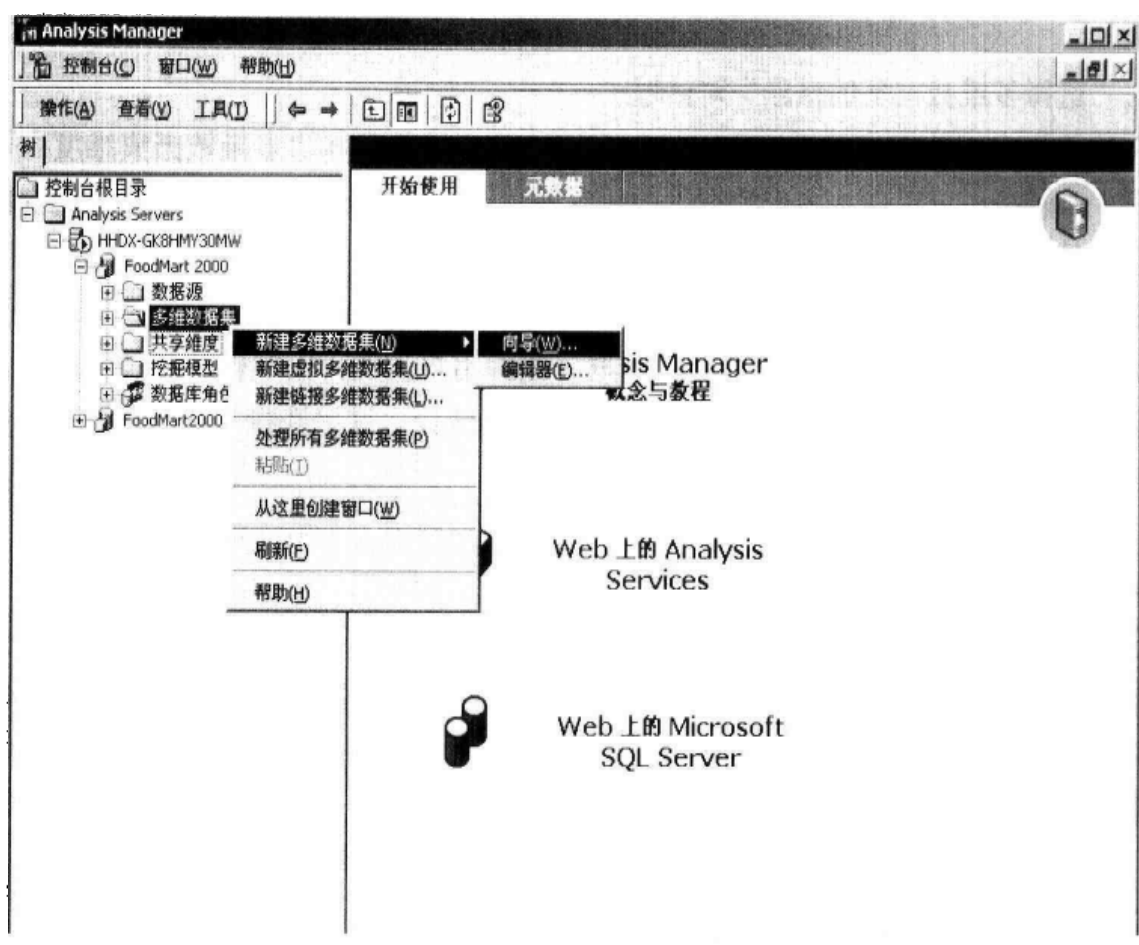


图 3.11 用 Analysis Services 创建多维数据集示意图

2. “从数据源选择事实数据表”对话框

在“从数据源中选择事实数据表”对话框中有“数据源和表”及“详细信息”列表框。一旦在“数据源和表”中选中某个表，该表的列信息将出现在“详细信息”中。若在“数据源和表”列表框中无法找到所需要的表，可以单击“新数据源……”按钮，创建一个新的数据源。单击“浏览数据”按钮，可以查看当前选中表的信息。这里确定事实表 `sales_fact_1998` 后，单击“下一步”按钮，进入“选择用于定义度量列的数字列”对话框。

3. “选择用于定义度量列的数字列”对话框

在“选择用于定义度量列的数字列”对话框中，有“事实数据表数字列”和“多维数据集度量值”两个列表框。前者包含当前事实表中数字类型列，单击“>”或“>>”按钮可以将选中的列或所有列移入“多维数据集度量值”列表框。这里选择并且移动 `store_sales`，`store_cost`，`unit_sales` 3 个度量列后，单击“下一步”按钮，进入“选择多维数据集的维度”对话框。

4. “选择多维数据集的维度”对话框

在“选择多维数据集的维度”对话框中有“共享维度”和“多维数据集维度”两个列表框。前者给出可以作为多维数据集的所有共享维，用“>”或“>>”按钮将选中的列或所有列移入“多维数据集维度”列表框。如果尚未创建维，可以单击“新建维度”按钮，创建一个新维。这里选择 customers, store, time 3 个共享维后，单击“下一步”按钮，出现一个事实数据表行数计算的提示框。提醒用户，统计需要耗费较长的时间。单击“是”按钮，将进行统计；单击“否”，则不进行统计。这里选择“是”按钮，计算完成后，进入“多维数据集向导完成”对话框。

5. “多维数据集向导完成”对话框

在“多维数据集向导完成”对话框中（见图 3.12），可在多维数据集名称的文本框中输入多维数据集名称“销售数据 1998”，在多维数据集结构列表框中观察多维数据集的树形结构。单击“浏览”按钮观看多维数据集的数据。最后单击“完成”按钮，完成多维数据集的创建。

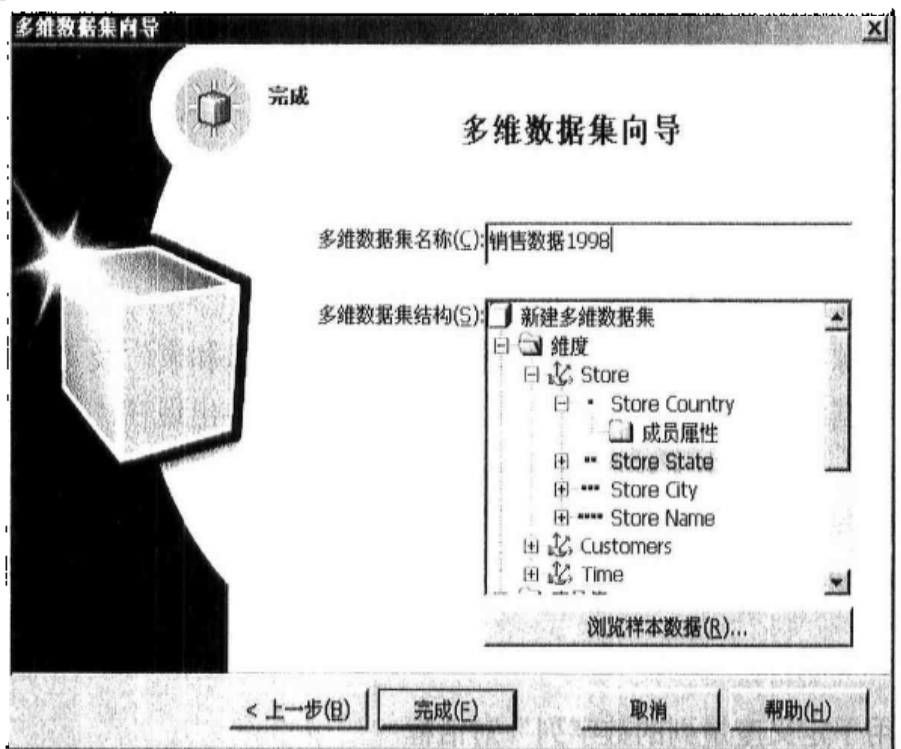


图 3.12 “多维数据集向导完成”对话框

系统接着自动显示“多维数据集编辑”对话框（见图 3.13），在该窗口中图示一个由 Sales_fact_1998 事实表和三个名为 customers, store, time by day 的维表构成的星型架构模型。用户可在该窗口中编辑多维数据集。在关闭“多维数据集编辑”对话框后，系统将显示一个“存储设计”提示对话框，提醒用户对多维数据集设置存储方式且对多维数

据集进行处理，只有经过处理的多维数据集才能在以后使用。在单击“是”按钮后，系统将启动“存储设计向导”欢迎对话框（见图 3.14）。

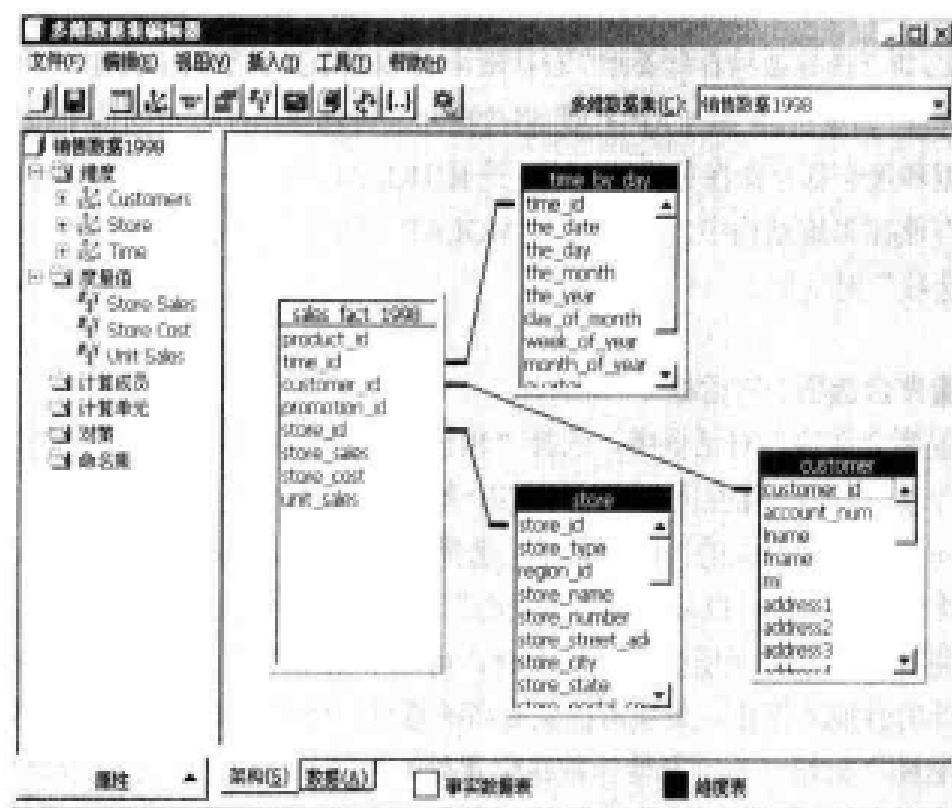


图 3.13 “多维数据集编辑”对话框

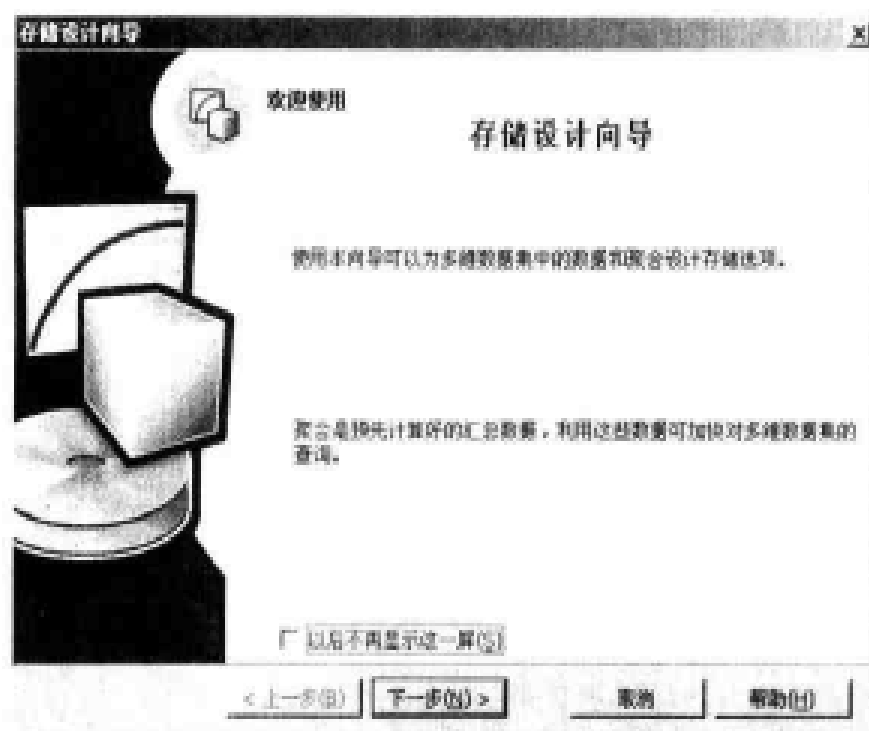


图 3.14 “存储设计向导”欢迎对话框

6. “存储设计向导”欢迎对话框

在“存储设计向导”欢迎对话框中,单击“下一步”按钮,将调出“选择数据存储类型”对话框。在“选择数据存储类型”对话框中有 MOLAP 类型、ROLAP 类型和 HOLAP 类型 3 个单选项。如果选择 MOLAP 类型,将数据和聚合都存储在多维结构中;选择 ROLAP 类型,将数据和聚合都存储在关系结构中;选择 HOLAP 类型,将数据存储 in 关系数据库中,而聚合存储在多维结构中。这里选择 MOLAP 类型后,单击“下一步”按钮,进入“设置聚合选项”对话框。

7. “设置聚合选项”对话框

在“设置聚合选项”对话框中,选择“预计占用的存储空间达到”选项,可以输入用来存储聚合表的硬盘存储空间(MB 或 GB 单位);选择“性能提升达到”选项,可以指定查询语句达到指定性能的性能百分比;选择“直到单击‘停止’”选项,可用手工来控制查询的优化程度,还可以从“性能与大小”框图中观看性能所达到的程度。单击“开始”按钮,将开始设计基于所选方式的聚合,可以观察到“性能与大小”框图的变化,一直达到设计的性能才停止。如果对性能提高的进度、所达到的性能和空间数据不满意,可以单击“继续”按钮(原“开始”按钮位置处)继续设计。若要重新设计聚合选项,可以单击“重置”按钮。当对设计满意时,单击“下一步”按钮,进入“存储向导完成”对话框。

8. “存储向导完成”对话框

在“存储向导完成”对话框中,如果选择“立即处理”选项,并且单击“完成”按钮,系统将对多维数据集的聚合立即进行处理。如果选择“保存,但现在不处理”选项,系统将只保存聚合的设计而进行多维数据集的处理。这里选择“立即处理”选项,并且单击“完成”按钮,系统完成聚合处理后,进入出系统的“处理”对话框。如果处理成功,应在“处理”对话框中显示“已成功完成处理”信息。此时可以单击“关闭”按钮,完成多维数据集的全部设计工作。

完成多维数据集的设计后,可在 Analysis Manager 控制台的左边树形结构中一直展开到“多维数据集”节点,再选中“多维数据集”节点中的“销售数据 1998”后,单击右边多维数据集显示框中的“数据”超链接按钮(参见图 3.15),就可以在右边多维数据集描述框中查到该多维数据集的数据(参见图 3.16)。

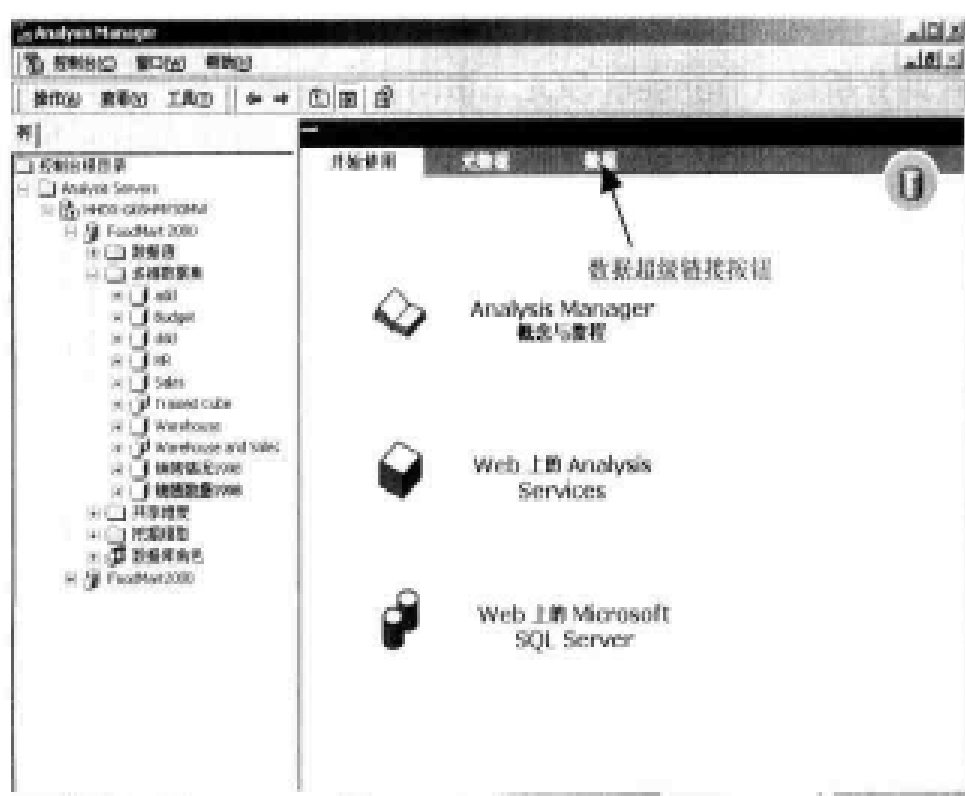


图 3.15 Analysis Manager 控制台的多维数据集

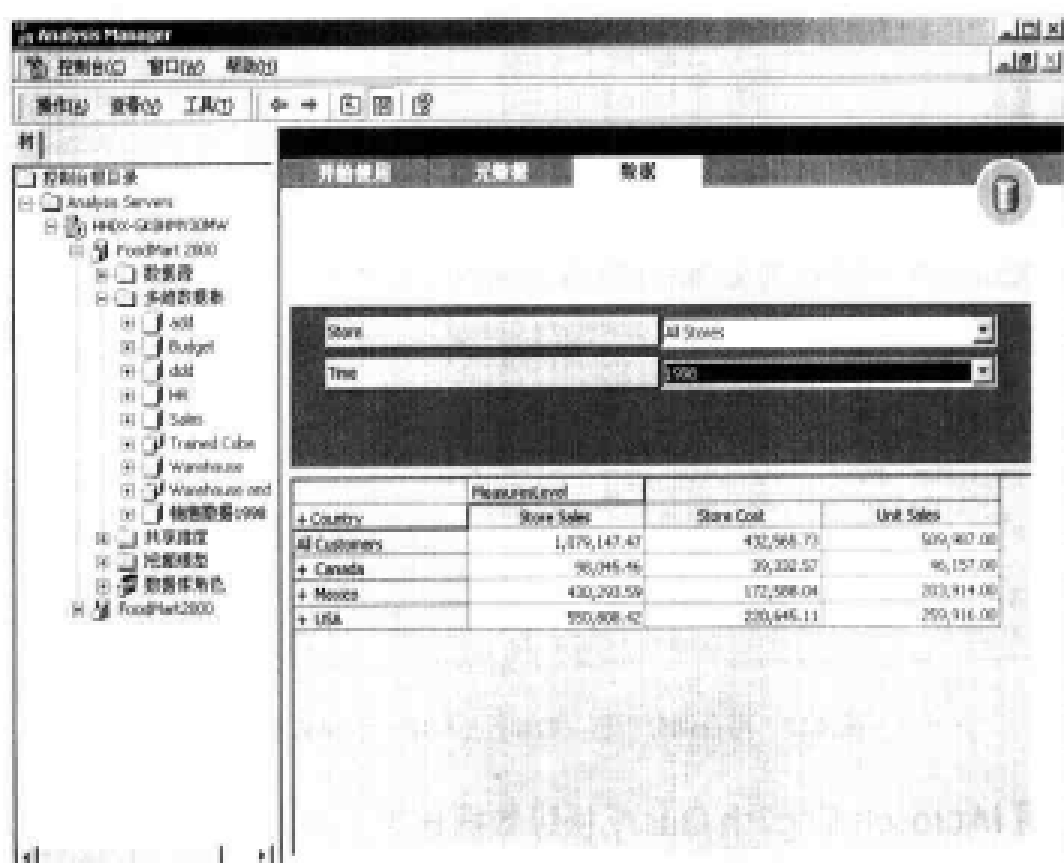


图 3.16 Analysis Manager 控制台的多维数据集浏览图

3.3.4 用查询分析器 (Transact-SQL) 访问数据仓库

Transact-SQL 语言是微软公司在 SQL Server 中的 SQL-3 的实现。这是一种交互式查询语言, 且还增加了变量、运算符、函数、流程控制等语言功能, 使其功能更加强大, 用于对关系数据库的定义、操纵、控制等操作。数据仓库的各种数据均来自各种数据库系统, 使数据仓库与数据库密不可分。因此, 可以使用 Transact-SQL 对数据仓库进行各种操作。

查询分析器是执行和分析 Transact-SQL 语句的良好工具, 用“开始”→“程序”→Microsoft SQL Server→“查询分析器”菜单项启动“查询分析器”。在“查询分析器”的 Query 语句输入框中, 输入 Select 语句, 单击工具栏上绿色的三角按钮, 完成所输入的 Select 语句在数据仓库中对事实表和维度表查询。例如, 用 Select * from orders where OrderID>11000 and EmployeeID=2 语句可以实现对事实表 orders 中订单号大于 10500, 同时雇员编号为 2 的销售情况的查询 (参见图 3.17)。

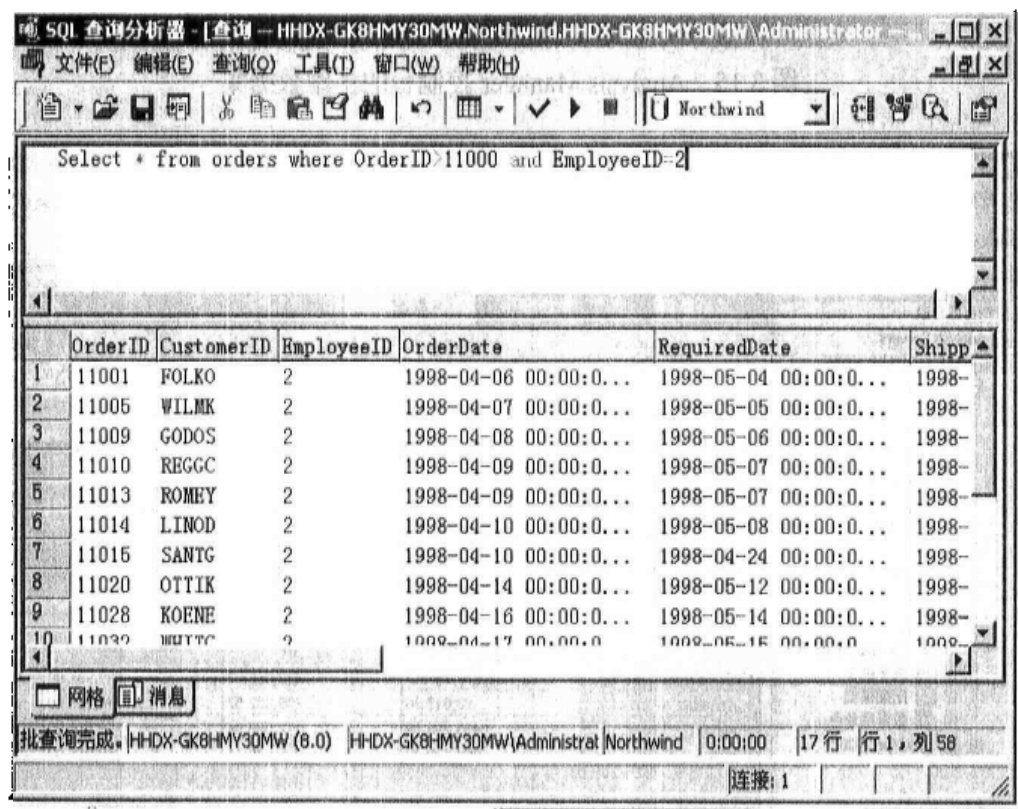


图 3.17 用查询分析器 (Transact-SQL) 访问数据仓库

3.3.5 用 Microsoft English Query 操纵数据仓库

Microsoft English Query 作为与 Microsoft SQL Server 2000 集成在一起的工具, 可用

自然语言——英语对数据仓库进行操作。用 Microsoft English Query 操纵数据仓库多维数据集数据过程见如下 5 点。

1. Microsoft English Query 的启动

按照“程序”→“Microsoft SQL Server”→“English Query”→“Microsoft English Query”的顺序启动 Microsoft English Query 服务，选择 File/New Project 菜单，弹出 New Project 对话框。从 New 标签页的左边列表框选择创建对象的类型为 English Query Projects，从右边列表框选择 OLAP Project Wizard 工具（参见图 3.18），在 Name 文本框中输入将要创建的工程名，然后在 Location 文本框中指定该工程所在的位置。最后单击“打开”按钮，进入“Select an Analysis Server”对话框。

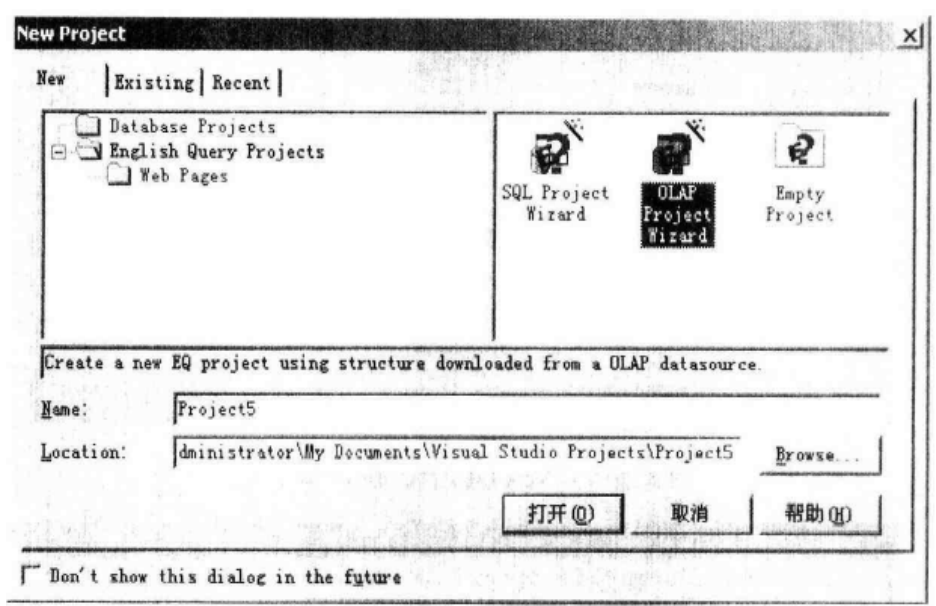


图 3.18 Microsoft English Query 查询

2. Select an Analysis Server 分析服务器的进入

出现“Select an Analysis Server”对话框后（见图 3.19），在 Analysis 文本框中输入分析服务器的名称，Database 下拉列表框中选择需要查询的数据库名。单击“OK”按钮，进入“New OLAP Cubes”对话框。

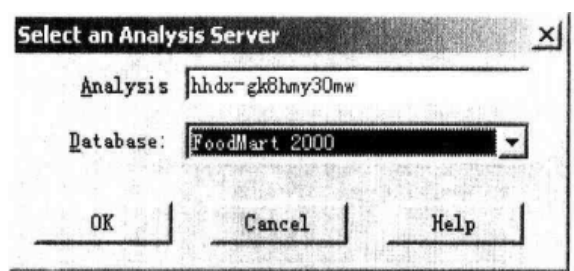


图 3.19 “Select an Analysis Server”对话框

3. 多维数据集的选择

出现“New OLAP Cubes”对话框后（见图 3.20），从 Available 列表框中选择将要使用的多维数据集，用向右箭头或双击选中的多维数据集，将其移入到 Selected 列表框中。最后，单击“OK”按钮，进入“Project Wizard”对话框（见图 3.21）。

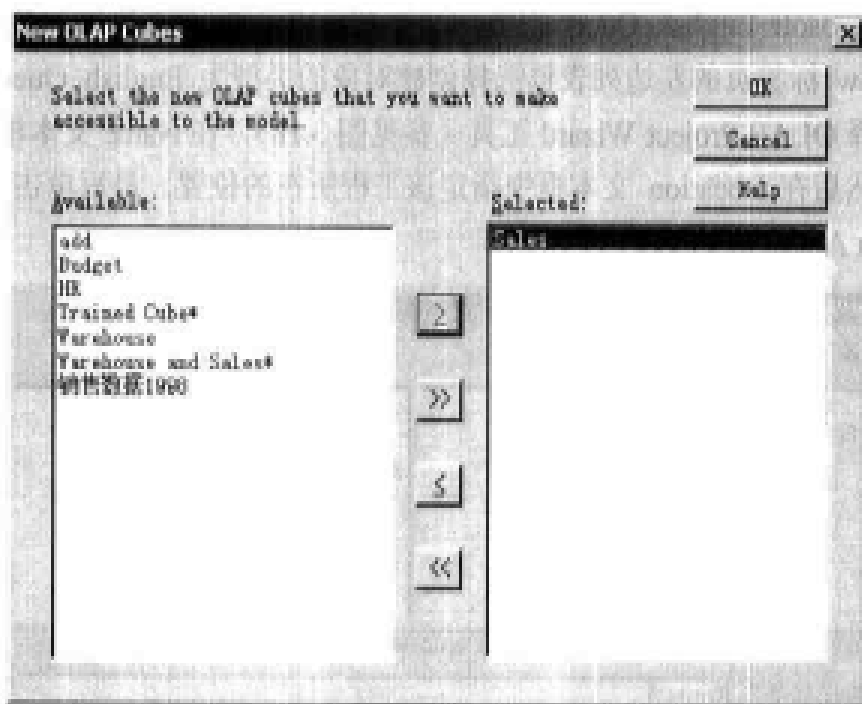


图 3.20 “New OLAP Cubes”对话框

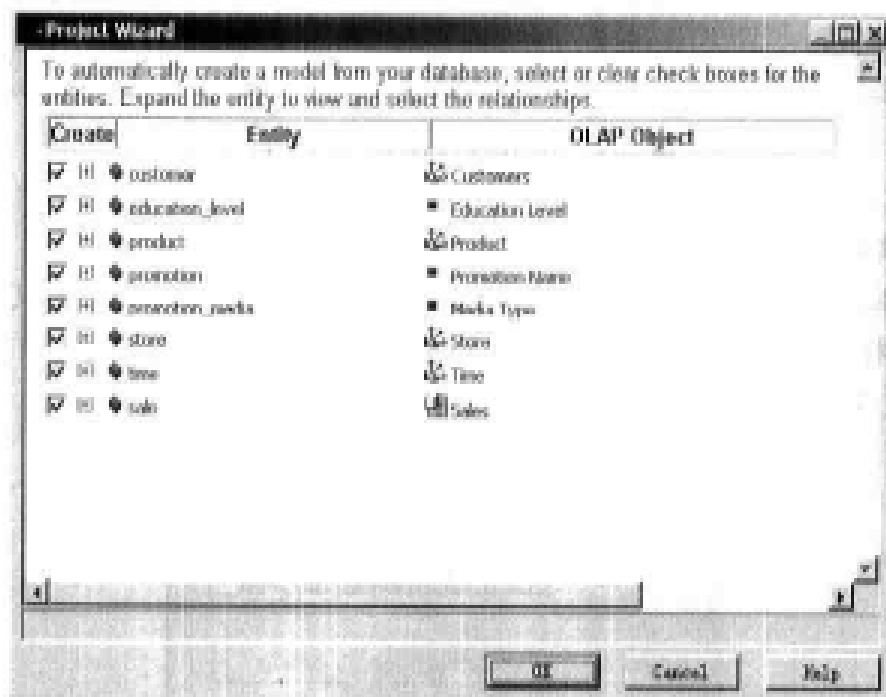


图 3.21 “Project Wizard”对话框

4. 英语查询模型的创建

出现“Project Wizard”对话框后，在对话框中的左边列出了所创建模型的实体（Entity），右边列出了与实体相对应的 OLAP 对象（OLAP Object）。如果希望改变这种对应关系，就需要单击实体左边的加号（+）。确定后，单击“OK”按钮，完成英语查询模型的建立。

5. Query 语句的执行结果

出现英语查询模型的对话框后，在该对话框中选择 DebugStart 菜单项。在出现“Model Test”对话框后，在对话框的 Query 下拉列表框中输入需要使用的语句，单击对话框工具栏上的绿色三角形按钮（执行按钮），就可得到 Query 语句的执行结果（见图 3.22）。

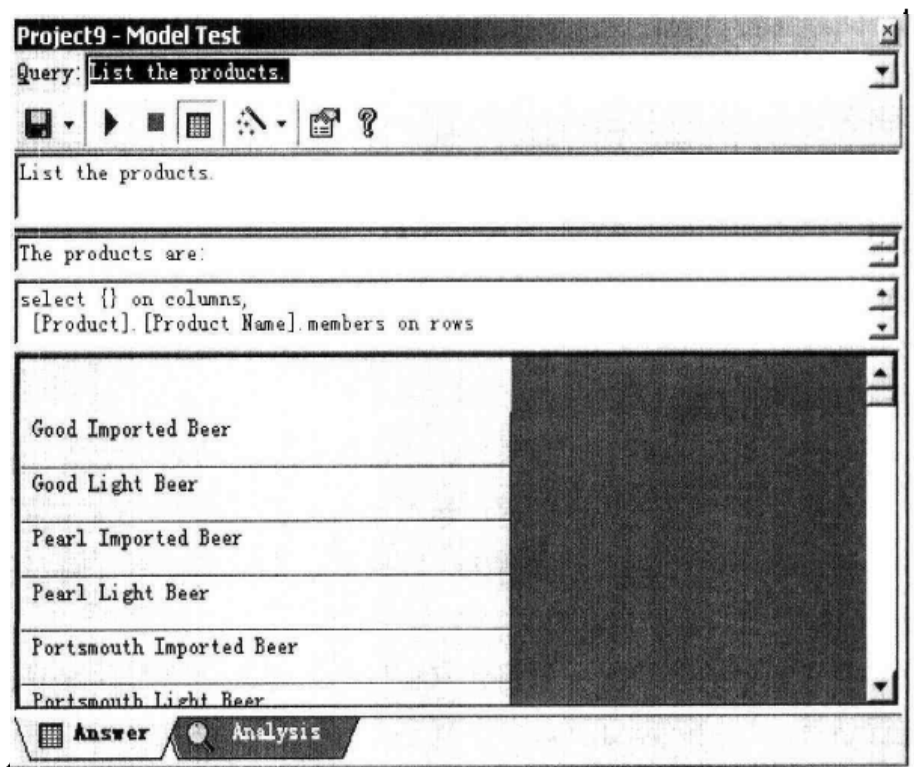


图 3.22 Query 语句执行结果图

3.4 SQL Server 中的数据提取与加载

SQL Server 中的数据析取和加载工具很多，这里只介绍其中的数据复制技术和数据转换服务（DTS）。

3.4.1 SQL Server 的数据复制工具与应用

利用 SQL Server 的复制向导，可以简化复制的配置和执行。在 SQL Server 企业管理

器中可以启动复制向导，在“工具”菜单上指向“复制”子菜单，然后单击适当的向导。

SQL Server 中的数据复制是指将一个系统中的数据通过网络分布到地理位置不同的其他系统中。在数据仓库中，经常利用复制技术完成数据仓库框架构成以后的数据加载。

在 SQL Server 中可以利用图形化工具创建复制。首先选择准备复制的数据库服务器，然后在 SQL Server 的企业管理器中的“工具”菜单中打开向导菜单项，调出“选择向导”对话框，选择其中的“复制”节点，可以找到 5 个有关复制的向导工具：创建发布向导、创建强制新订阅向导、创建请求订阅向导、禁用发布或分布向导、配置发布和分布向导。

1. 创建发布向导

利用发布向导完成这样一些操作：选择发布数据库；使用发布模板；选择发布类型；选择可更新的订阅或可传送的订阅（快照复制或事务复制可以使用的选项）；指定订阅服务器类型；指定发布的数据和数据库对象项目；选择发布名称和描述；自定义发布属性，包括筛选列、筛选行、启用动态筛选器、验证订阅信息、优化同步、允许匿名订阅以及设置快照代理调度，以完成数据发布的创建。数据发布的开始需要在数据发布服务器上打开 SQL Server 企业管理器，展开一个服务器组，展开复制文件夹，右击发布文件夹，然后单击“新建发布”命令，按照向导提示完成数据的分布创建。

2. 创建强制新订阅向导

利用强制订阅可以简化和集中订阅管理，不必对每个订阅服务器进行管理。当同步强制订阅时，分发代理程序或合并代理程序运行于分发服务器。强制订阅在发布服务器上创建，复制代理程序不经订阅服务器请求，就将更新数据传播给订阅服务器。数据更改也可按照调度强制发布给订阅服务器。

在强制订阅中，集中的分发服务器将建立调度，按照此调度与远程的、偶尔连接的订阅服务器进行连接。使用强制订阅，分发代理程序（用于快照发布和事务发布）或合并代理程序（用于合并发布）可以运行于分发服务器。如果需要从分发服务器卸载代理程序处理，则可以在订阅服务器上运行代理程序。

建立订阅时要考虑的因素是需要订阅的类型（强制、请求或匿名）以及运行复制代理程序的位置。如果用户是订阅服务器上的 sysadmin 或 db_owner 角色的成员，可以建立强制订阅。对于建立强制订阅的 db_owner 角色的成员，订阅服务器必须由 sysadmin 角色的一位成员注册。

为了创建订阅，发布服务器上必须有发布，订阅服务器上也必须有订阅数据库。可以在创建订阅之前创建订阅数据库，或在创建强制订阅向导中指定新的订阅数据库。可为任何在发布服务器和分发服务器的属性中启用的订阅服务器，创建强制订阅。

强制订阅和请求订阅称为署名订阅，因为有关订阅和订户的信息存储在发布服务器

上, 有关订阅服务器的性能信息存储在分发服务器上。这与匿名订阅 (一种请求订阅类型) 不同, 匿名订阅保存少量或不保存有关订阅和订阅服务器的信息。

从企业管理器中创建强制订阅的步骤是: 在发布管理器控制台中打开 SQL Server 企业管理器, 依次展开服务器组、复制文件夹、分布内容文件夹, 右击想要订阅的发布, 然后单击“强制新订阅”命令 (参见图 3.23), 进入“欢迎使用强制订阅向导”对话框。依照向导提示完成数据强制订阅的创建。

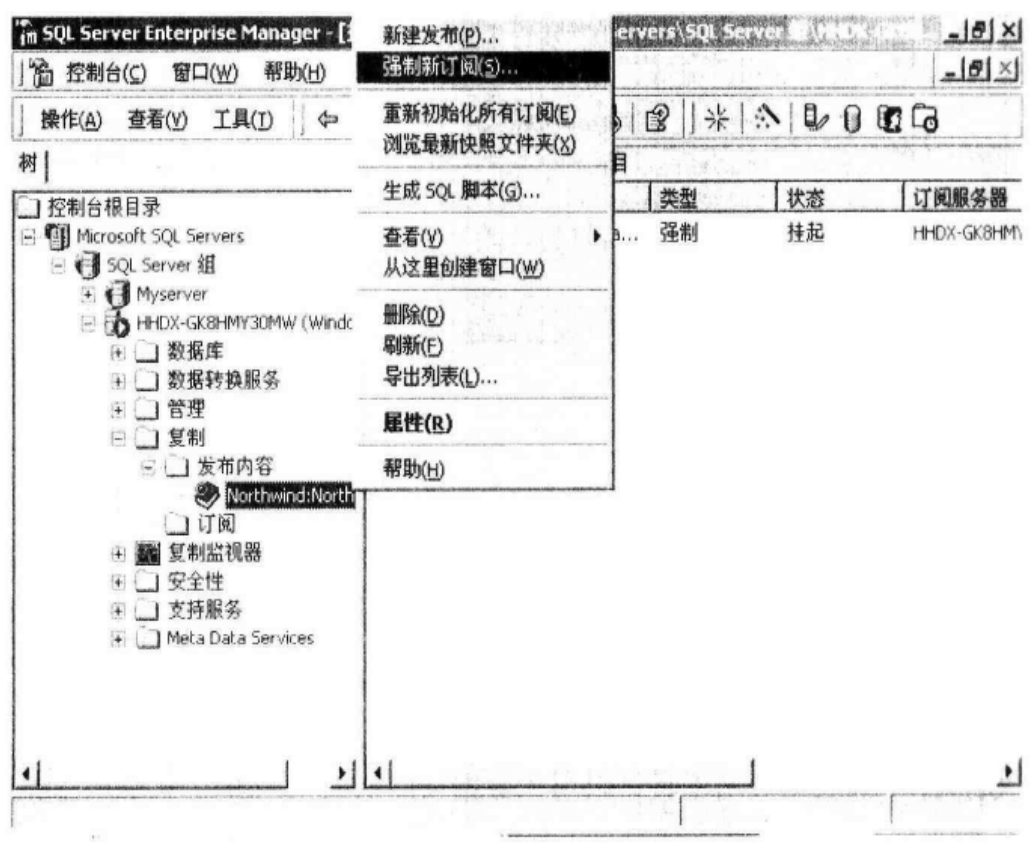


图 3.23 创建强制新订阅

3. 创建请求订阅向导

从企业管理器中创建强制订阅的步骤是: 在发布管理器控制台中打开 SQL Server 企业管理器, 依次展开服务器组、复制文件夹, 右击订阅文件夹 (参见图 3.24)。进入“创建请求订阅向导”欢迎对话框。利用该向导可以完成选择数据发布 (选择数据源)、选择在某个发布上创建订阅的数据库、设置初始化调度和同步调度等工作。

4. 禁用发布或分布向导

在 SQL Server 的企业管理器中的“工具”菜单中打开向导菜单项, 调出“选择向导”对话框, 选择其中的“复制”节点, 选择“禁用发布或分布向导”菜单项, 进入“欢迎使用禁用发布或分布向导”对话框。利用该向导可以完成“除去所选服务器上的所有发

布”或“除去对应已除去发布的所有订阅”，这些设置不会影响该服务器从其他发布服务器接收到的订阅。

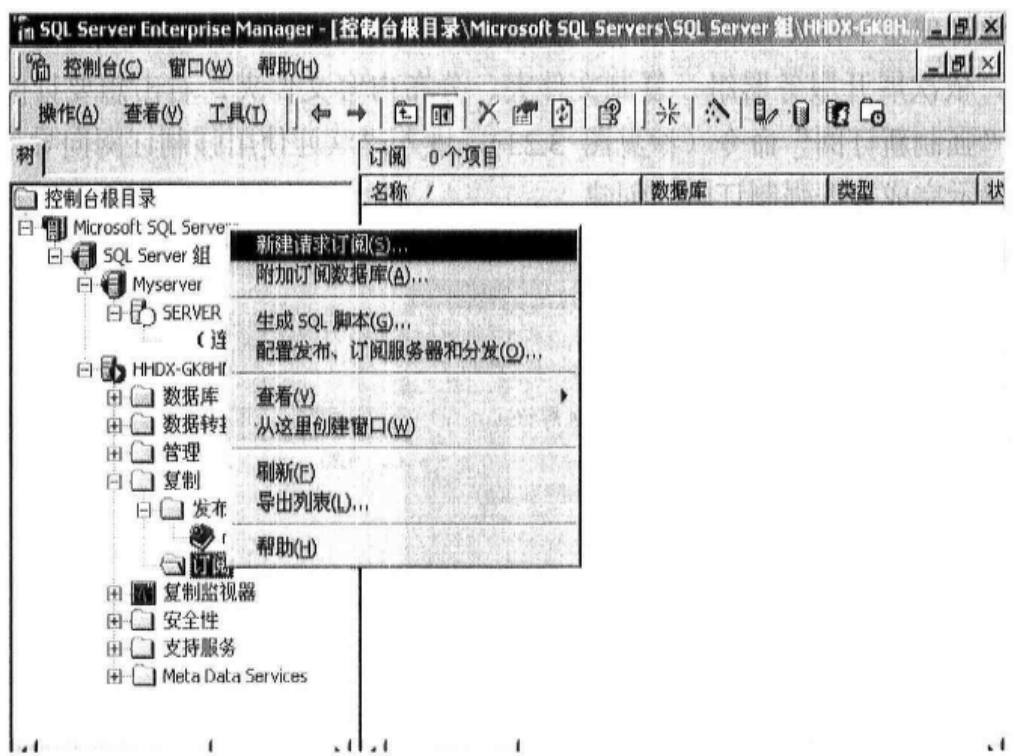


图 3.24 创建请求订阅

3.4.2 DTS 的数据导出工具 (DTS Export Wizard)

SQL Server 中的数据导出工具可以用于数据仓库的数据加载。在 SQL Server 的企业管理器中，依次选择菜单项“工具”→“数据转换服务”→“导出数据”（参见图 3.25）。进入“数据转换服务导入/导出向导”欢迎对话框，单击“下一步”按钮，进入“选择数据源”对话框。

1. “选择数据源”对话框

在“选择数据源”对话框中，首先从“数据源”下拉列表框选择数据源类型；随着数据源的变化，对话框中的其他属性设置将会改变。这里选择“用于 SQL Server 的 Microsoft OLE DB 提供程序”数据源。然后从“服务器”下拉列表框中选择数据源所在服务器名称；如果选择使用 Windows 身份验证选项，将确定为 NT 认证；选择使用 SQL Server 身份验证选项，将确定为 SQL Server 认证方式；如果选择 SQL Server 认证方式，还需要在用户名文本框中输入 SQL Server 的登录账户名，在密码文本框中输入账户的口令；从数据库下拉列表框中选择该数据源所在的数据库；“刷新”按钮用于刷新窗口内容，恢复系统的默认值。选定以上这些选项后，单击“下一步”，进入“选择目的地”对话框。

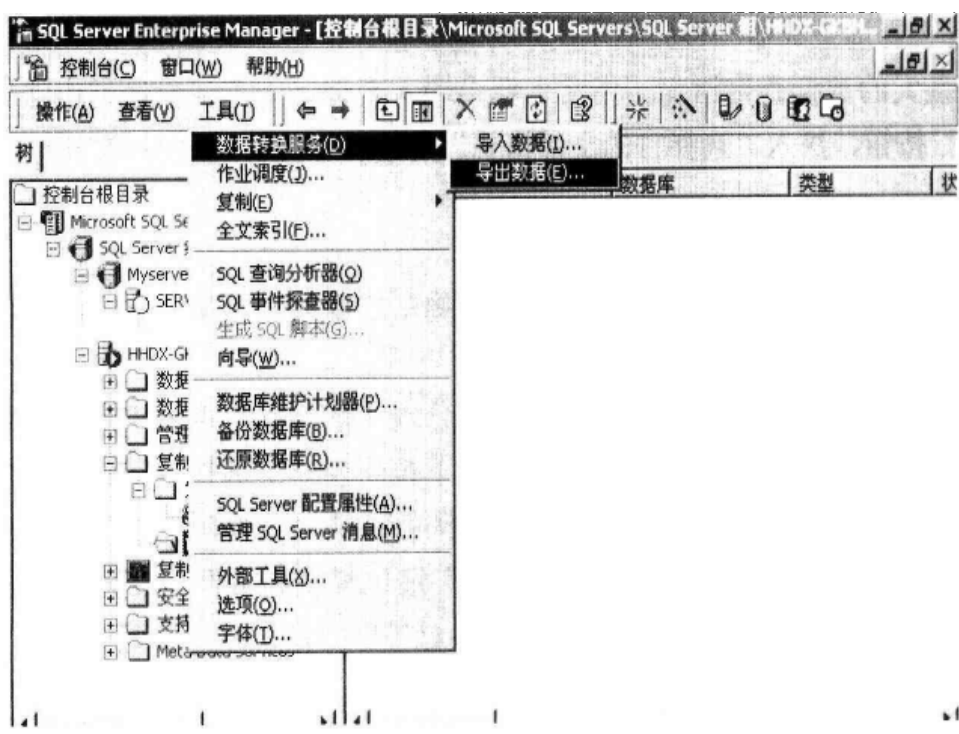


图 3.25 DTS 的数据导出工具选择

2. “选择目的地”对话框

在“选择目的地”对话框中，有一个“目的”数据源选择下拉列表框，从中可以选择导出数据目标源类型。同样，数据目标源类型的不同也将导致其他属性设置的不同，这里选择“文本文件”数据类型。接着，可在“文件名”文本框中输入文本文件的完整名称，或者单击省略号按钮，浏览选择文本文件。最后，单击“下一步”按钮，进入“指定表复制或查询”对话框。

3. “指定表复制或查询”对话框

在“指定表复制或查询”对话框中，有 3 个单项选择按钮：选择“从源数据库复制表和视图”选项，可从源数据库中拷贝若干个表的内容到文本文件（选定的导出数据目标源文件）中；选择“用一条查询指定要传输的数据”选项，可用一个查询语句来传输所选择的数据。选择“在 SQL Server 数据库之间复制对象和数据”选项，可在源数据源与目标数据源都是 SQL Server 时，用于两者之间的对象与数据的传递。这里选择“用一条查询指定要传输的数据”选项后，单击“下一步”，进入“键入 SQL 语句”对话框。

4. “键入 SQL 语句”对话框

在“键入 SQL 语句”对话框中，有一个“查询语句”文本框，用于查询语句的输入；“浏览”按钮用于选择包含 SQL 语句的脚本文件；“分析”按钮用于检查查询语句的语

法是否正确；“查询生成器”按钮用于图示化建立查询语句。单击“查询生成器”按钮可以生成功能强大的数据查询语句。当文本框中输入查询语句，并且经过语法检查后，单击“下一步”按钮，进入“选择目的文件格式”对话框。

5. “选择目的文件格式”对话框

在“选择目的文件格式”对话框中，首先确定生成的文本文件的字段之间采用分隔符隔离还是采用固定字段长度隔离。

选择“带分隔符，各列之间可用任何字符分隔”选项后，再从“文件类型”下拉列表框中选择文本文件的类型；从“行分隔符”下拉列表框中选择数据行之间的分隔符；从“列分隔符”下拉列表框中选择列之间的分隔符；从“文本限定符”下拉列表框中选择文本内容之间的分隔符；“第一行含有列名”选项将文本文件中的第一行作为列名处理，而不是数据。“转换”按钮可以根据需要，对导出的数据进行必要的转换。选项确定后，单击“下一步”按钮，进入 DTS “保存、调度和复制包”对话框。

6. “保存、调度和复制包”对话框

在 DTS “保存、调度和复制包”对话框中，如果选择“立即执行”选项，将立即进行数据的复制，该选项可以用于数据仓库创建后的首次数据加载；选择“调度 DTS 包以便以后执行”选项，将调度 DTS 包在以后指定时间执行，该选项可以用于数据仓库的数据定期加载；选择“保存”选项，将把本次导出的数据作为 DTS 包保存。

如果选择“保存”选项，还要确定包的类型。选择 SQL Server 选项，包的类型是 SQL Server 中的表；选择 SQL Server Meta Data Services 选项，包的类型是数据仓库类型；选择“结构化存储文件”选项，包的类型是结构化的类型；选择 Visual Basic 文件，包的类型是 Visual Basic 文件类型。确定这些选择后，单击“下一步”按钮，进入“保存 DTS 包”对话框。

7. “保存 DTS 包”对话框

在“保存 DTS 包”对话框中，可以确定 DTS 包的保存信息：在“名称”文本框中输入 DTS 包的名称；“描述”文本框中输入 DTS 包的说明；“所有者密码”文本框中输入 DTS 包的所有者口令；“用户密码”文本框中输入 DTS 包使用者的口令；且在两个选择按钮中确定 DTS 包的认证方式。完成这些设置后，单击“下一步”按钮，进入“正在完成 DTS 导入/导出向导”对话框。如果发现设置有误，可以单击“上一步”按钮，对前面设置返回修改。如果对前面设置确认正确，单击“完成”按钮，完成本次数据导出操作。在“完成”按钮确定后，系统根据设置立即完成数据的导出复制功能。可从系统提供的“正在执行包”对话框中查看执行结果。单击“正在执行包”对话框中的“完成”按钮，将返

回企业管理器控制台。

3.4.3 DTS 的数据导入工具 (DTS Import Wizard)

SQL Server 中的数据导入工具可以用于从业务系统中将数据析取到数据准备区，然后加载到数据仓库。为了使用 DTS 的数据导入工具，可在 SQL Server 的企业管理器中，依次选择菜单项“工具”→“数据转换服务”→“导入数据”（参见图 3.25）。确定后，进入“数据转换服务导入/导出向导”欢迎对话框，前两步操作与数据的导出相似，这里不再重复叙述。

只是在数据导入过程中，如果进入“指定表复制和查询”对话框时（见图 3.26），将遇到 3 个单项选择按钮：“从源数据库复制表和视图”选项、“用一条查询指定要传输的数据”和“在 SQL Server 数据库之间复制对象和数据”选项。其中的“用一条查询指定要传输的数据”选项可以利用 SQL 的 Select 语句的强大查询功能对数据准备区中的数据按照行、列筛选和清理，完成数据区的清理工作。下面详细介绍该功能的使用方法。在选择“用一条查询指定要传输的数据”选项后，单击“下一步”，进入“键入 SQL 语句”对话框。

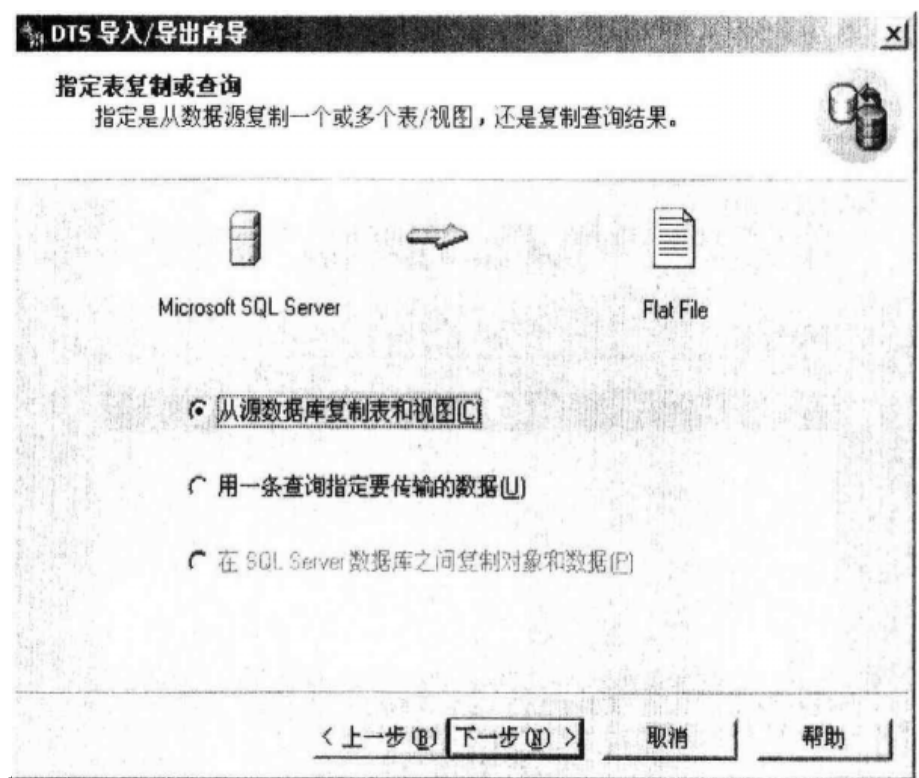


图 3.26 “指定表复制或查询”对话框

在“键入 SQL 语句”对话框中，有个用于输入查询语句的文本框，如果用户对 SQL 查询语句熟悉，可以直接在此框内输入 Select 语句；如果对 Select 语句并不熟悉，则可

单击“查询生成器”，进入“选择列”对话框。

在“选择列”对话框中有两个列表框，其中一个为“源表”，另一个为“选中的列”。前者列出所选源数据库中的所有表，单击表树形结构，可以将表内的列名全部展开；双击所选中的列名，可将选中列移入“选中的列”列表框。这样，就可完成数据的按列筛选。单击“下一步”按钮，进入“指定列顺序”对话框。

在“指定列顺序”对话框中有两个列表框，一个为“选中的列”，另一个为“排序”。前者列出了前一对话框中所选定的所有列，如果要对某些列进行排序，就可以将这些列移入“排序”对话框。在“排序”对话框中可以用“上移”和“下移”按钮对其进行排序。单击“下一步”按钮，进入“指定查询条件”对话框。

在“指定查询条件”对话框中，可以完成数据源的按行选择抽取。如果选择“全部行”选项，可以单击“下一步”按钮，进入下一对话框。如果选“只有满足条件的行”选项，还需要在“列”、“运算符”和“值/列”的下拉列表中通过对“列”、“运算符”和“值/列”的选择来构造查询条件。如果单击“值/列”的下拉列表，则可选择某个列名。单击“值/列”的省略号则是选择所选列的值。在“指定查询条件”对话框中所指定的查询条件中的子查询条件，最多只能由三个子查询条件组成。单击“下一步”按钮，返回“键入 SQL 语句”对话框。在此框中单击“分析”按钮，系统给出 SQL 语句的分析结果；如果显示“SQL 语句有效”，可以单击“下一步”按钮，进入“选择源表和视图”对话框（见图 3.27）。

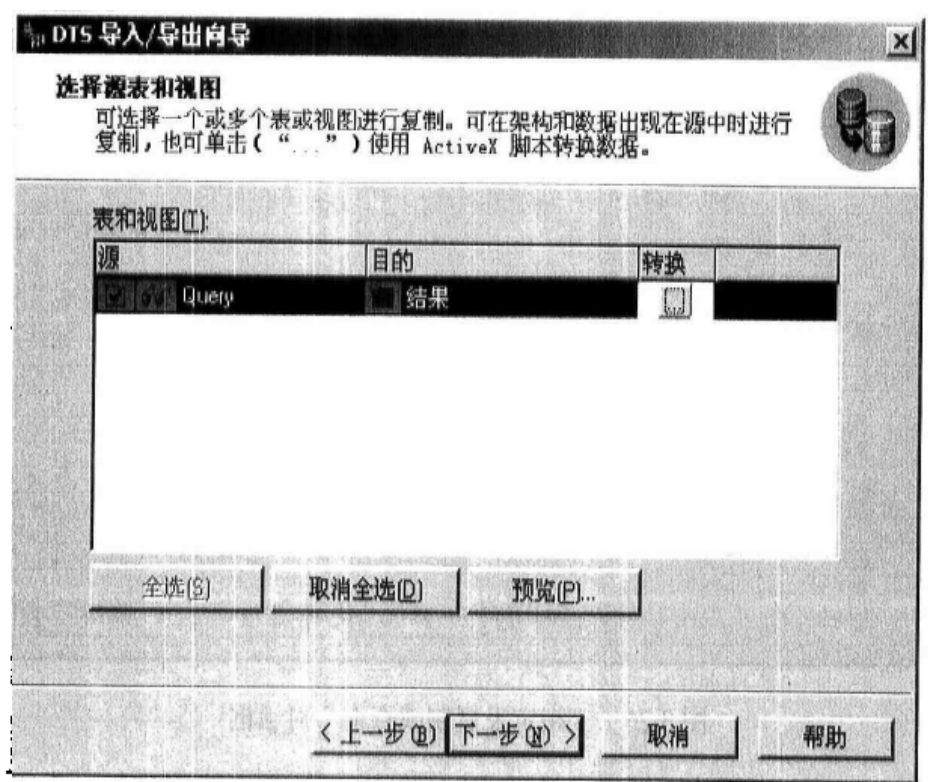


图 3.27 “选择源表和视图”对话框

在“选择源表和视图”对话框中，可以选择一个或多个表或视图进行复制。可在架构和数据出现在源中时进行复制，也可以单击省略号按钮，使用 ActiveX 脚本转换数据。当确定后，可以单击“预览”按钮，观看复制的结果。如果单击省略号按钮，可以进入“列映射和转换”对话框。该对话框的作用将在“DTS 的数据转换”小节中讨论。设置完成后，在“选择源表和视图”对话框中单击“下一步”按钮，进入“保存、调度和复制包”对话框。该对话框的作用如同数据的导出向导的第 6 步，其操作过程一样，不再重复叙述。在“保存、调度和复制包”对话框中单击“下一步”按钮，进入数据导入向导的完成对话框。在其对话框中单击“完成按钮”，系统将根据向导中的设置执行 DTS 包，并且显示执行正确信息。确定完成后，返回企业管理器的控制台，可以单击数据导入目的数据库中的表，找到已经导入的数据。

3.4.4 DTS 的数据转换

利用 DTS 进行数据导入 / 导出过程中，还可以对数据进行各种转换。转换过程在进入“指定表复制或查询”对话框后按照以下选择进行。

1. “指定表复制或查询”对话框

在数据导入 / 导出的“指定表复制或查询”对话框中，如果选择“从源数据库复制表和视图”选项后，单击“下一步”按钮，可以进入“选择目的文件格式”对话框。如果选择“用一条查询指定要传输的数据”选项后，单击“下一步”，进入“键入 SQL 语句”对话框。在“键入 SQL 语句”对话框中利用“查询生成器”，也可以按向导进入“选择目的文件格式”对话框（见图 3.28）。

2. “选择目的文件格式”对话框

在“选择目的文件格式”对话框中单击“转换”按钮，可以进入“列映射和转换”对话框。

3. “列映射和转换”对话框

在“列映射和转换”对话框中，有列映射和转换两个标签页。在列映射标签页中，可对需要导入的数据进行简单的映射转换，将关系表中指定列的名称、数据类型、长度、非空特性等映射成另外的类型。若要进行复杂的数据转换，就要选择转换标签页，在该页中有两个选项：一个是将源表中的列简单地拷贝到目标列中的“直接将源列复制到目的列”选项，另一个是可以使用脚本执行数据转换的“在将信息复制到目的时对其进行转换”选项。如果选中后一个选项，还需要从语言下拉列表框中选择 JScript Language 或

VB Script Language 脚本语言。转换数据的脚本程序可以事先用其他文本编辑器编辑好，存于某个目录中，用“浏览”按钮搜索。最后，单击“确定”按钮，返回“选择目的文件格式”对话框，完成数据的转换。

数据的转换处理也可以在“选择源表和视图”对话框中进行，单击列表框的“转换”列的省略号按钮，也可进入“列映射和转换”对话框，进行数据转换的设置。

数据转换的设置和数据仓库中经常用于实现从数据源所抽取的数据按照数据仓库设计要求，进行数据的清理和转换工作。

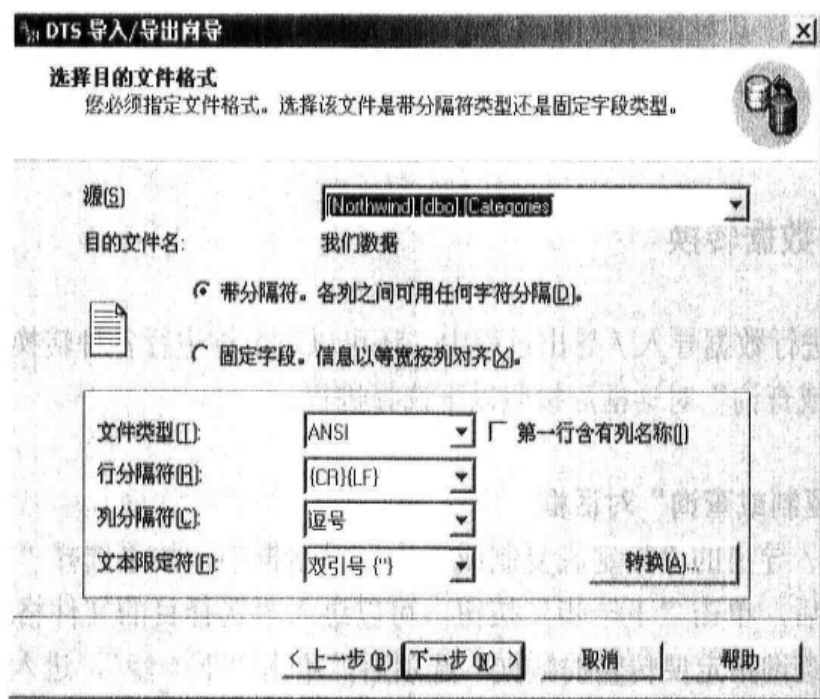


图 3.28 “选择目的文件格式”对话框

3.5 SQL Server 中的数据挖掘工具与应用

在 SQL Server 中已将数据的分析和预测集成到 Analysis Services 中，且在决策支持对象 (DSO) 和 PivotTable 中也扩展了对数据挖掘技术的支持，使用户能够很方便地将数据挖掘工具用于分析 OLAP 多维数据集的数据。

3.5.1 SQL Server 中的数据挖掘工具

由于 Analysis Services 可对关系数据库和多维数据源中的数据进行挖掘，因此任何利用 OLE DB 可以访问的关系数据源数据以及通过 Analysis Services 创建的多维数据集的数据都可以训练挖掘模型。而且 SQL Server 系统的可扩展性使第三方工具能够与 SQL Server 的数据挖掘工具组装使用，增加了系统的性能与灵活性。

同时, Analysis Services 自身新增加了许多帮助设计、创建、训练和浏览数据挖掘模型的向导和工具, 可以基于 OLAP 数据挖掘模型创建维度和虚拟多维数据集, 使 Analysis Services 的数据挖掘功能更为强大。

在 MS SQL Server 2000 中, 数据挖掘工具已经和 PivotTable 集成在一起, 使 PivotTable 将数据挖掘模型作为多维数据集处理。因此, 终端用户能够创建客户端的数据挖掘模型, 或者从服务器的多维数据集中查看其他数据挖掘模型的信息。

数据挖掘模型决定如何对大量历史数据进行分析。数据挖掘工具则提供用于处理数据挖掘的分类、分割、关联和分析数据所需要的决策制定能力, 提供有关分析对象的预测、变化的信息, 并且允许用户决定采用自动模式进行数据挖掘或采用交互方式完成数据的挖掘。

在 Analysis Services 中所提供的数据挖掘模型主要是 Microsoft 决策树模型和 Microsoft 数据聚集模型两种。

3.5.2 决策类数据挖掘工具的应用

依次打开 SQL Server 的 Analysis Manager 控制台、指定的服务器节点、FoodMart 2000 数据库节点, 右键单击“挖掘模型”节点(参见图 3.29), 在弹出式菜单中选择“新建挖掘模型”, 可以建立新的数据挖掘模型。选择“处理所有模型”, 可对已有的数据挖掘模型进行数据聚合处理。

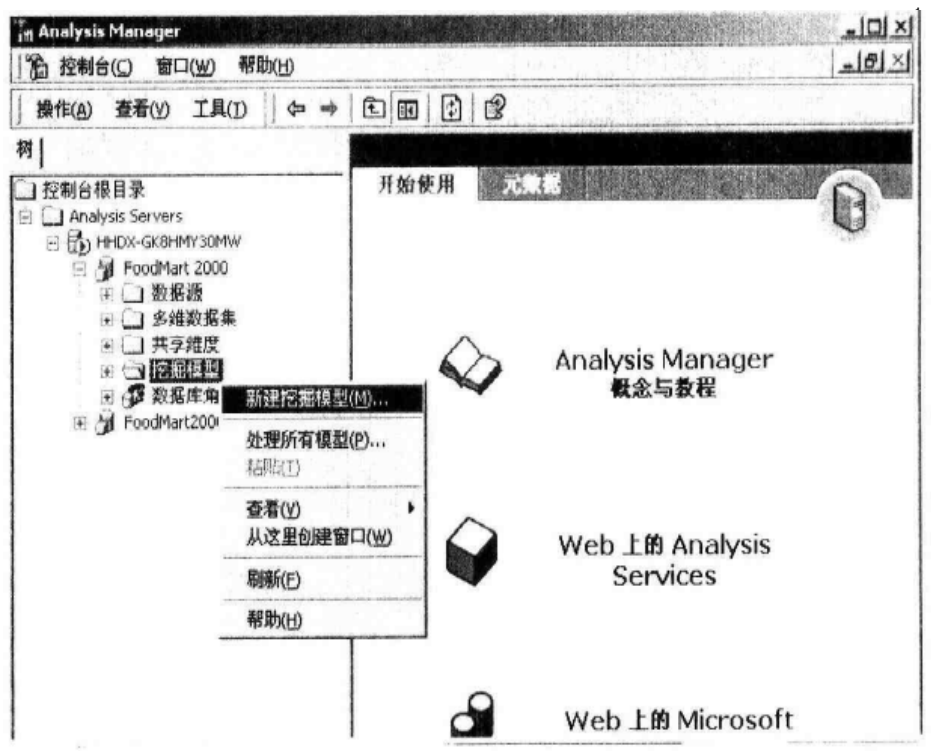


图 3.29 “挖掘模型”工具的进入

1. 挖掘数据源选择

如果选择了“新建挖掘模型”菜单项，则将出现“挖掘模型向导”欢迎对话框。单击“下一步”按钮后，将出现“选择源类型”对话框。可以选择“关系数据”选项或“OLAP数据”选项。选择前者将建立一个基于关系型数据源的挖掘模型，用于查询 Analysis Services 所支持的任意关系数据源中的数据；选择后者，将建立一个基于多维数据结构的挖掘模型，用于查询多维数据集数据；如果选择了预测的实体，还可以创建用于浏览的维度和虚拟多维数据集。

2. 挖掘事例表选择

在选择“关系数据”后，单击“下一步”按钮，进入“选择事例表”对话框（见图 3.30）。框中有两个选项：一个是“单个表包含数据”选项，另一个是“多个表包含数据”选项。这里选择“单个表包含数据”，然后从“可用的表”列表框中选择一个表，被选中表的列内容就出现在“详细信息”列表框中。若要重新创建一个数据源，可以单击“新数据源”按钮，创建一个新的作为数据挖掘模型基础的数据源。单击“浏览数据”按钮，可以浏览选定表中的数据。

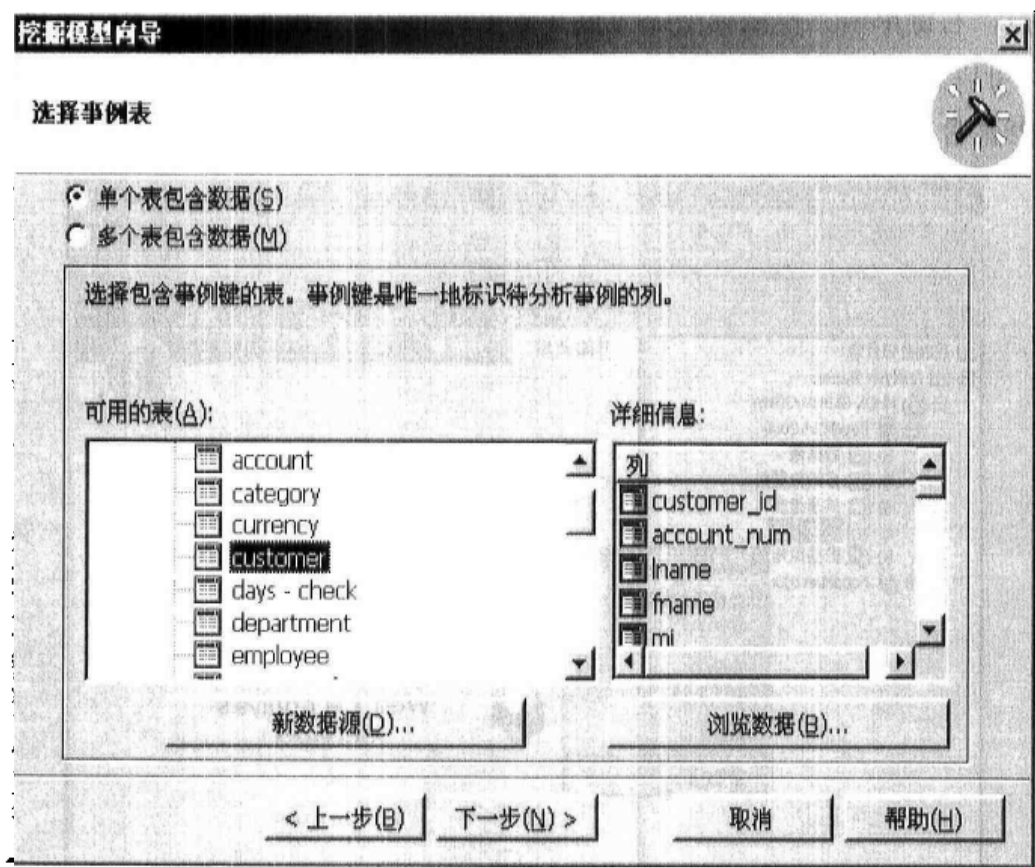


图 3.30 “选择事例表”对话框

3. 挖掘技术选择

在“选择事例表”对话框中，单击“下一步”按钮后，将出现“选择数据挖掘技术”对话框（见图 3.31）。SQL Server 提供“Microsoft 聚集”和“Microsoft 决策树”两种数据挖掘技术。在选定“Microsoft 决策树”算法以后，单击“下一步”按钮，进入“选择键例”对话框。

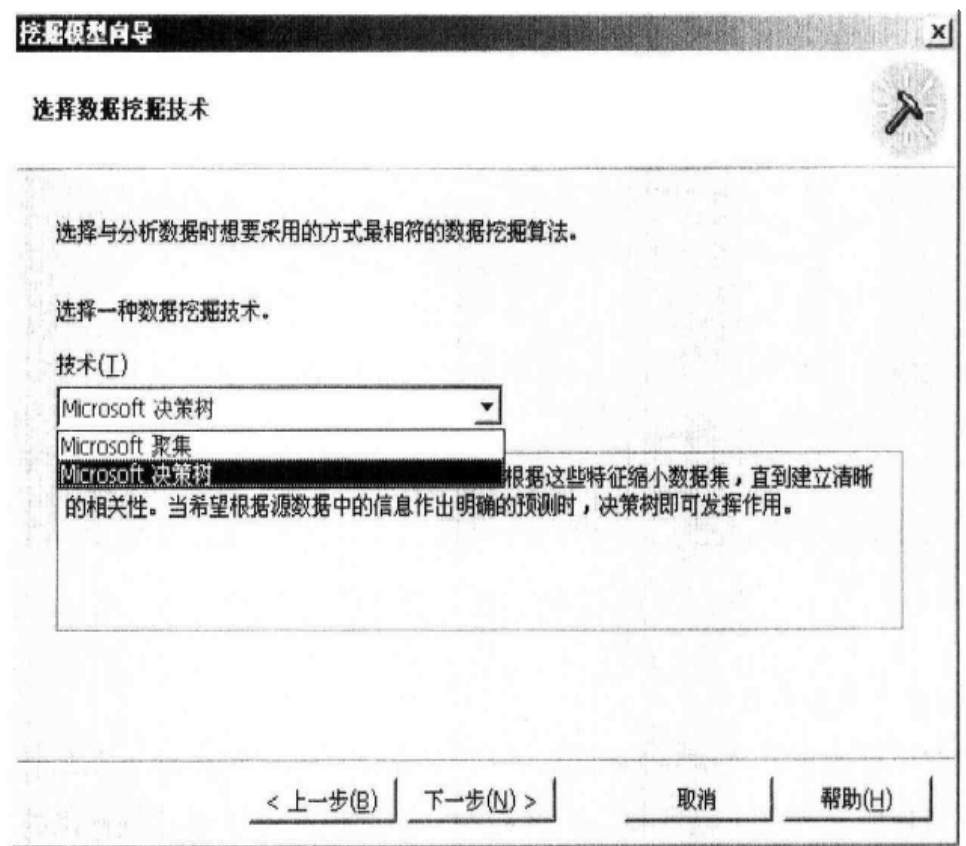


图 3.31 “选择数据挖掘技术”对话框

在“选择键例”对话框中提供“事例键列”下拉列表框，从中选择一个可以标识表元组惟一的列。这里选择 `customer_id` 列，然后单击“下一步”按钮，进入“选择输入列和可预测列”对话框。

4. 挖掘参数选择

在“选择输入列和可预测列”对话框（见图 3.32）中，需要为挖掘模型确定对哪些列数据进行分析，所分析的对象或分析后输出哪些结果列。窗口提供“可用的列”、“可预测列”和“输入列”3 个列表框。在“可用的列”列表框中，可以选择作为分析结果输出的列，然后单击向右箭头，将其移入“可预测列”列表框。在“可用的列”列表框中选择作为分析输入数据的列，然后单击向右箭头，将其移入“输入列”列表框。这里选择了 `birthdate`, `marital_status`, `yearly_income` 和 `gender` 数据列。然后单击“下一步”按钮，

进入“挖掘模型向导”完成对话框。

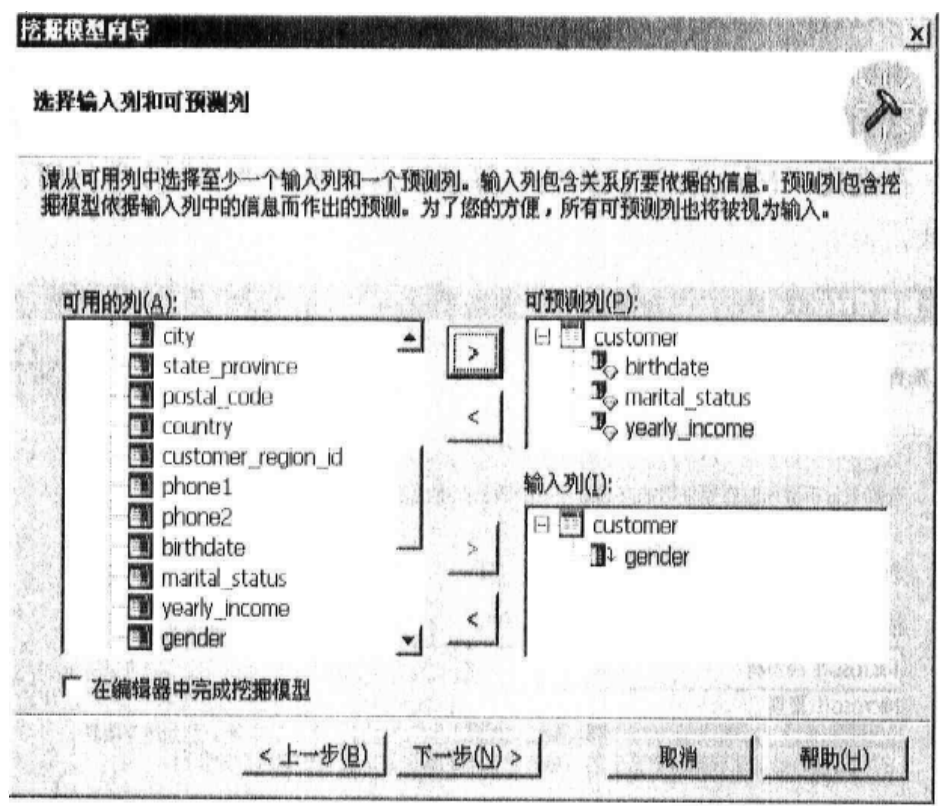


图 3.32 “选择输入列和可预测列”对话框

5. 挖掘模型保存选择

在“挖掘模型向导”完成对话框中，需要确定模型的名称和是否立刻对模型进行处理。在模型名称的文本框中输入模型的名称后，如果现在就需要对数据挖掘模型进行处理，就选择“保存并开始处理”选项，处理以后将生成模型且用数据训练该模型。否则选择“保存，但现在不处理”选项。最后，单击“完成”按钮，完成数据的决策树模型创建。

在数据挖掘模型创建和模型处理完成以后，将出现“处理”对话框（见图 3.33）。在此对话框中显示对数据挖掘模型的详细处理过程信息，以便了解系统是如何处理模型数据的。在“处理”对话框中出现“已成功完成处理”提示后，可以单击“关闭”按钮。系统将调出“关系挖掘模型编辑器”对话框。关闭该对话框后，返回 Analysis Manager 控制台，可以从挖掘模型节点中看到刚刚建立的挖掘模型。

6. 挖掘结果浏览

当数据挖掘模型生成以后，可以通过模型浏览器进行观察。例如，在 Analysis Manager 控制台中，依次打开指定的服务器节点、FoodMart 2000 数据库节点、挖掘模型节点。右键单击 Customer Pattern Discovery 模型，从弹出的菜单中选择“浏览”选项（参见图 3.34），

调出“数据挖掘模型浏览器”对话框（见图 3.35），可从窗口所显示的模型发现隐藏在众多数据后面的一些营销规律、客户规律等对用户具有重要价值的信息。

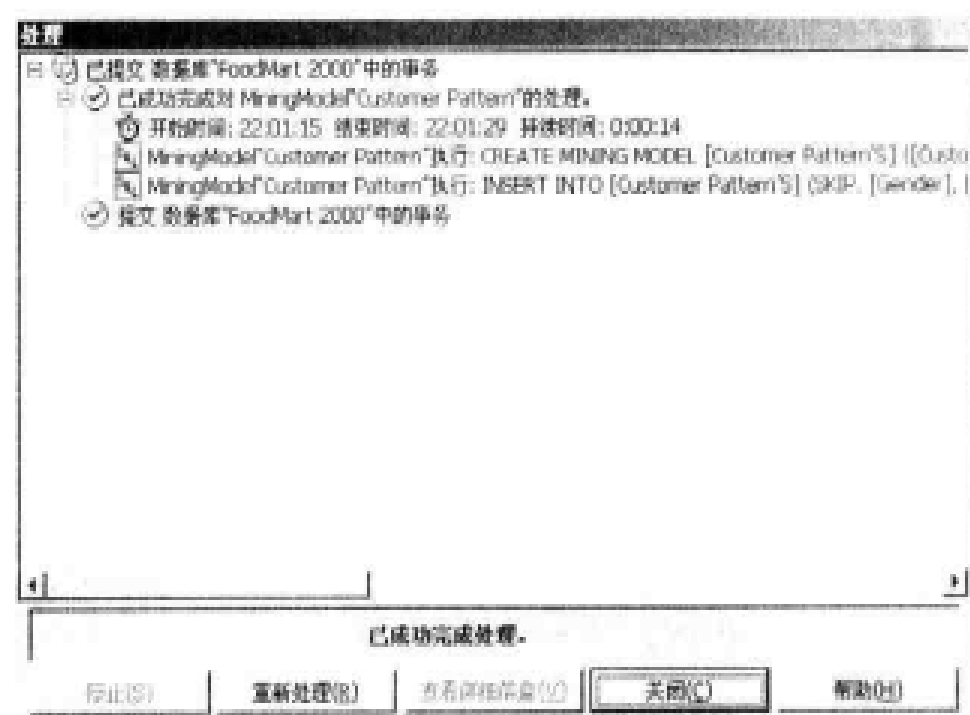


图 3.33 “处理”对话框



图 3.34 数据挖掘模型浏览

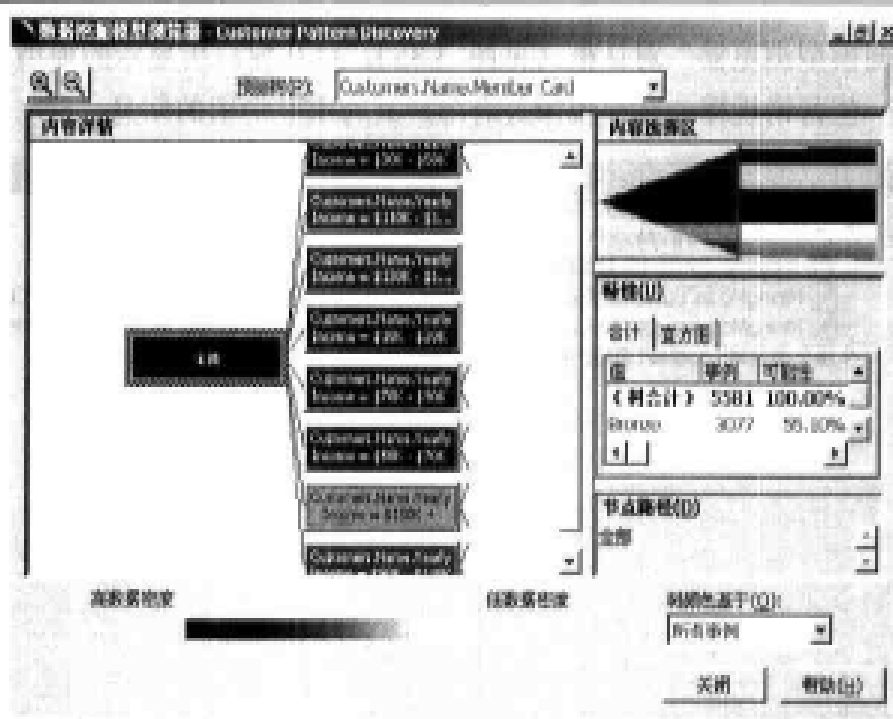


图 3.35 “数据挖掘模型浏览器”对话框

3.5.3 聚类分析的数据挖掘工具应用

如果用户在“选择源类型”对话框（见图 3.36）中选择“OLAP 数据”选择项，就可以对多维数据集数据进行挖掘。单击“下一步”按钮，进入“选择源多维数据集”对话框（见图 3.37）。

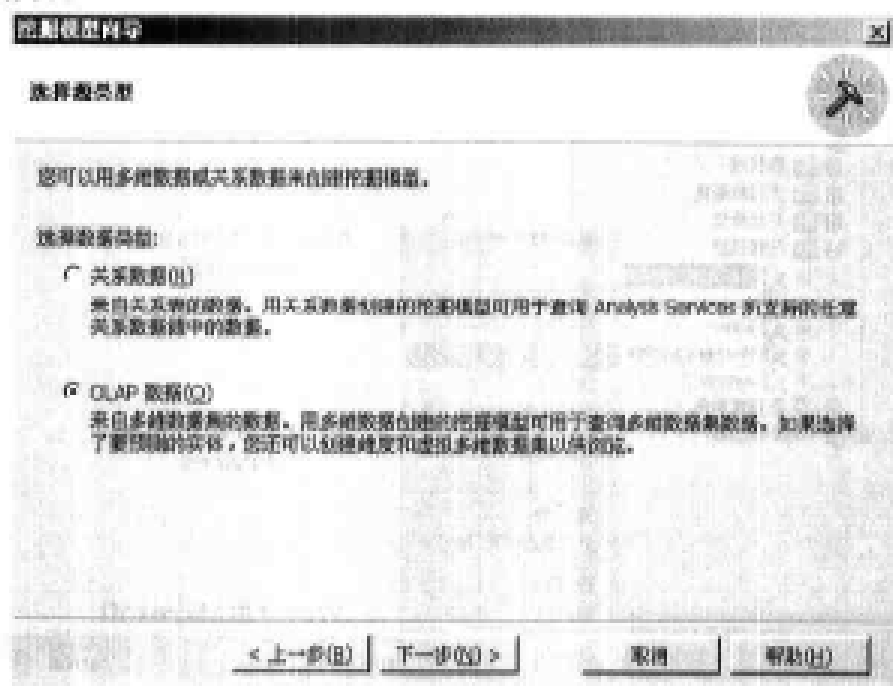


图 3.36 “选择源类型”对话框

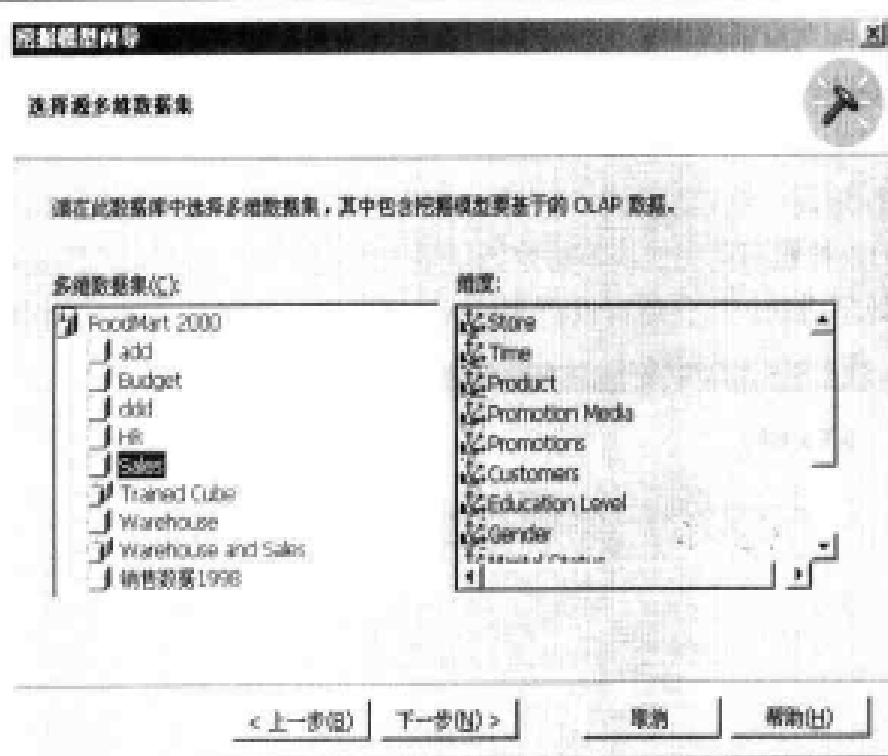


图 3.37 “选择源多维数据集”对话框

在“选择源多维数据集”对话框中，选择多维数据集和对应的维度后，就可以单击“下一步”按钮，进入“选择数据挖掘技术”对话框（见图 3.31）。在“选择数据挖掘技术”对话框中选择“Microsoft 聚集”挖掘算法，按“下一步”按钮后，进入“选择事例”对话框（见图 3.38）。

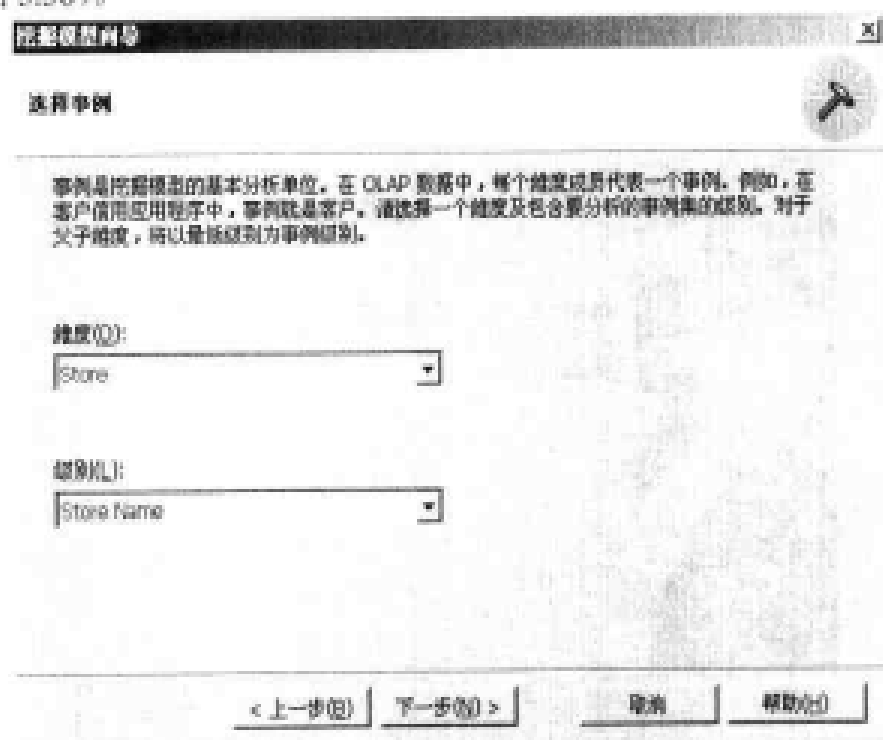


图 3.38 “选择事例”对话框

在“选择事例”对话框中要为所挖掘的多维数据集确定“维度”和“级别”。然后单击“下一步”按钮后,进入“选择培训数据”对话框(见图 3.39)。在“选择培训数据”对话框中除在“选择事例”对话框中所选择的维度和级别以外,至少还要选择一个项目,对挖掘模型进行培训。选定后,单击“下一步”按钮,进入“挖掘模型向导完成”对话框(见图 3.40)。按照前面完成决策树挖掘模型的操作步骤,结束聚集挖掘模型的操作。当模型保存、处理成功以后,就可以在 Analysis Manager 控制台中用“数据挖掘模型浏览

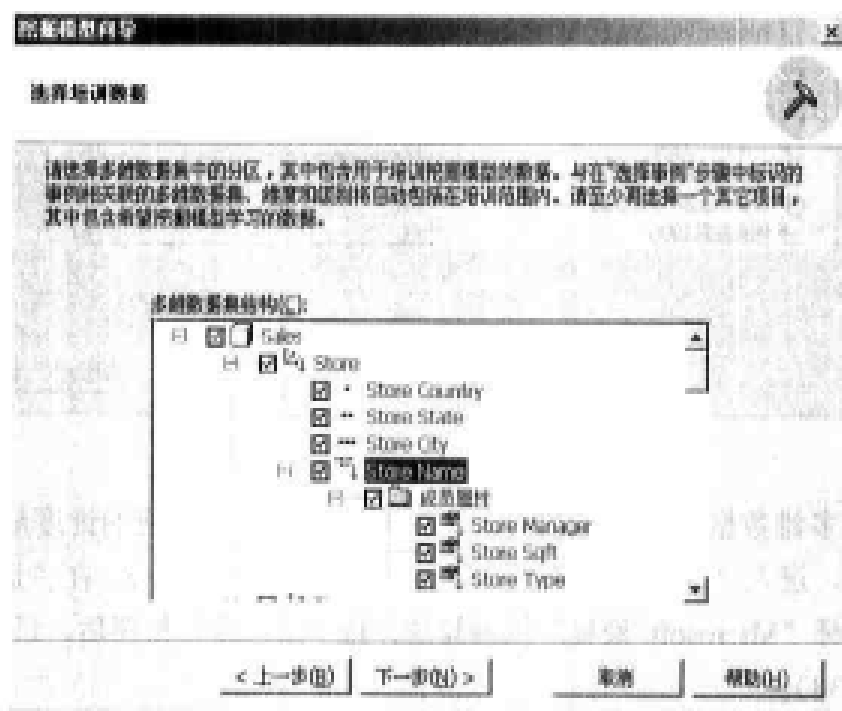


图 3.39 “选择培训数据”对话框

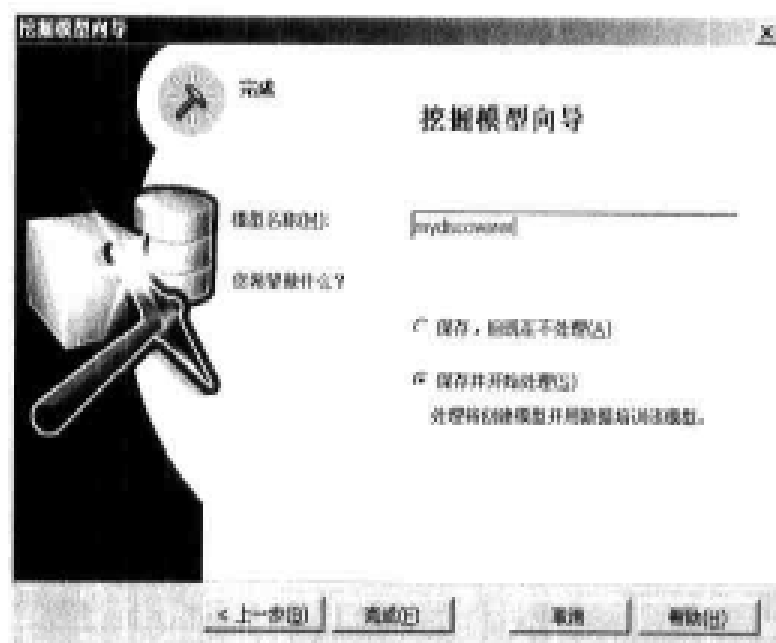


图 3.40 “挖掘模型向导完成”对话框

器”对模型进行观察（参见图 3.41），其浏览过程同决策类数据挖掘模型的浏览步骤。

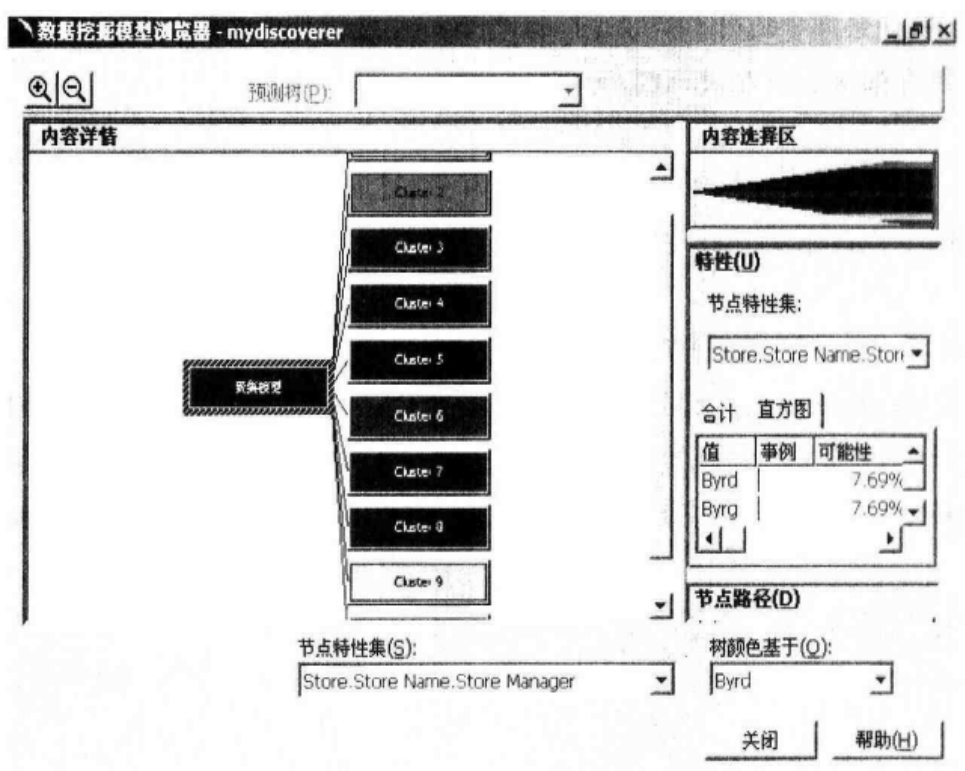


图 3.41 聚集数据挖掘模型浏览



本章小结

Microsoft SQL Server 2000 所提供的数据仓库创建、管理与使用的工具，覆盖了数据仓库应用的全过程。本章主要介绍了创建数据仓库数据库的工具，创建事实表和维表的工具，对数据仓库进行数据加载和转换的工具，对数据仓库中数据进行操纵的工具，对数据仓库中数据进行挖掘的工具。利用这些工具可以完成对数据仓库的初步创建与基本应用。

SQL Server 的数据仓库是利用数据库创建工具完成的。在创建数据仓库数据库的过程中需要设置数据库的常规、数据文件和事务日志标签页上的有关参数。数据仓库中的事实表和维表的创建需要利用 SQL 的表创建工具来完成。SQL 的维创建可以使用维度向导来完成，在设置维度过程中需要确定维度表、维度类型、维度级别、成员键列等维度的属性。在完成维度的设置后，需要创建多维数据集，即通常所说的立方体。在多维数据集的创建过程中需要确定事实表，事实表中的度量列、维度等属性。

在 SQL 的数据仓库创建过程中还需要利用复制工具和 DTS 数据导入与导出工具完成数据的加载和发布操作。这些工具不仅可以完成数据的加载和发布，且还可以完成数据

的筛选和转换。

在 SQL 数据仓库中还可利用 Analysis Services 进行数据挖掘的操作。SQL Server 的数据挖掘主要有聚集算法和决策树两种算法。



习题

- 3-1 利用 SQL Server 的数据仓库创建工具完成数据仓库的创建。
- 3-2 利用 SQL Server 的数据复制工具在两个数据服务器之间实现数据的发布和订阅功能。
- 3-3 如何对业务系统中的数据进行清理，筛选数据的条件是销售地域、销售产品的限制。
- 3-4 数据仓库在从业务系统中析取数据时，需要对数据进行转换。如何利用 SQL Server 进行数据的转换？
- 3-5 需要对数据仓库进行数据挖掘，如何利用 SQL Server 工具实现？

第 4 章

Delphi 中的 数据仓库设计与使用

引 言

【学习目标】

数据仓库在创建成功以后，用户可以直接登录数据仓库进行操作。但是已经习惯联机事务处理(OLTP)的大多数用户还缺乏数据仓库的使用经验，他们往往希望能以一种类似 OLTP 的方式使用数据仓库，即用 OLTP 对数据仓库进行操作。但要缺乏计算机应用能力的用户熟练地使用 Oracle 或 SQL Server 等数据仓库工具所提供的 OLAP 功能，也具有一定的难度。因此在数据仓库的应用中，尤其是数据仓库应用的早期，为用户提供一种定制的数据仓库应用方式，是解决这一矛盾的关键。用户通过定制的数据仓库应用工具，可以逐步地熟悉数据仓库的应用，逐步地增加数据仓库的应用能力。很显然，这种定制的应用方式应该和用户已经熟悉的业务处理方式相一致。也就是说，应该利用开发传统业务处理系统的方式开发这种数据仓库的定制应用模式。

通过本章学习，可以掌握：

- ◆ Delphi 6.0 的功能与程序设计环境
- ◆ Delphi 6.0 的程序设计方法
- ◆ Delphi 6.0 的数据仓库组件
- ◆ Delphi 6.0 数据仓库组件的应用
- ◆ Delphi 6.0 数据仓库的数据挖掘组件的应用

4.1 Delphi 简介

Delphi 发展到现在已经出现第六代产品——Delphi 6.0, 使 Delphi 的性能得到很大的提高。其中, 用于数据仓库设计的组件对于数据仓库应用系统的设计十分方便。从数据仓库的应用角度看, 主要由 OLAP 和数据挖掘组成。Delphi 中的数据仓库组件可以很方便地完成数据仓库定制应用的设计和数据挖掘的应用。

Delphi 作为 Borland 公司用 Object Pascal (对象 Pascal) 语言和汇编语言编写成的一个应用程序开发工具, 由于其易用性高, 受到许多系统开发人员的喜爱, 被大量地用于信息系统的开发。因为 Delphi 是基于 Pascal 发展的, 因此, 在使用 Delphi 前最好熟悉一下对象 Pascal 的语法和使用方法。这里只简单介绍 Delphi 的集成开发环境和利用 Delphi 进行应用程序开发的方法, 便于大家应用 Delphi 中的数据仓库组件进行数据仓库应用系统的开发。

4.1.1 Delphi 的开发集成环境组成

如果已经安装 Delphi 6.0, 可以通过 Windows 的“开始”→“程序”→“Borland Delphi6.0”→“Delphi6.0”菜单命令启动 Delphi 的集成开发环境 (IDE) (参见图 4.1)。

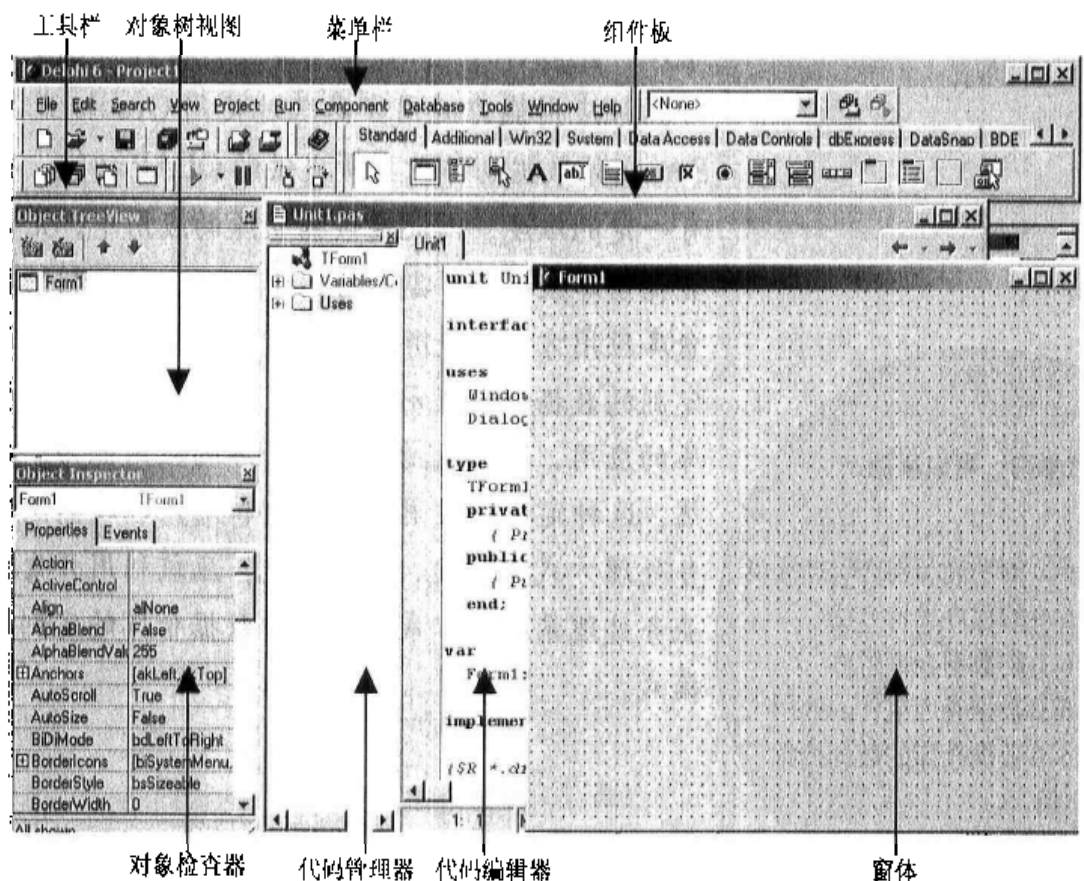


图 4.1 Delphi 集成开发环境的组成

从图 4.1 中可以发现, Delphi 的集成开发环境由菜单栏、工具栏、对象树视图、组件板、对象检查器、代码管理器、代码编辑器和窗体组成。

4.1.2 Delphi 的菜单栏与应用

Delphi 的菜单栏提供 Delphi 的所有人机界面, 设计人员通过它可以对 Delphi 进行方便的操作。在菜单栏中包含的菜单项有 File(文件), Edit (编辑), Search (查询), View (浏览), Project (项目), Run (运行), Component (组件), Database (数据库), Tools (工具) 和 Help (帮助)。

File (文件) 菜单项用于对文件的各种操作。

Edit (编辑) 菜单项用于在设计过程中对应用程序文本或组件的编辑操作。

Search (查询) 菜单项用于在代码编辑器中查找、替换特定的文本。

View (浏览) 菜单项用于激活或关闭 Delphi 集成开发环境中各种辅助工具及打开集成调试窗口。

Project (项目) 菜单项用于编译、创建和管理应用程序的工程项目。

Run (运行) 菜单项用于调试运行当前应用程序的工程项目。

Component (组件) 菜单项用于执行与组件相关的操作。

Database (数据库) 菜单项用于调用支持数据库应用系统设计的辅助工具。

Tools (工具) 菜单项用于调用 Delphi 中的各种辅助工具。

Help (帮助) 菜单项用于提供关于 Delphi 应用的各种帮助信息。

由于 Delphi 开发工具的复杂性, 不可能在这里用一章的篇幅中介绍清楚。读者可以使用随机帮助系统, 了解与熟悉 Delphi 的各种功能。

4.1.3 Delphi 的工具栏与应用

Delphi 的工具栏提供一些与常用菜单对应的快速按钮, 用于执行所对应的菜单命令。如果对工具条上按钮不熟悉, 可将鼠标停留在按钮上片刻, 就会出现带有按钮名称的提示框。

4.1.4 Delphi 的组件板与应用

用 Delphi 进行系统开发时, 大部分界面与功能都是利用组件板上的组件实现的。Delphi 6.0 的组件板提供 VCL 和 CLX 两大类型组件。VCL 用于建立在 Windows 操作系统上运行的应用程序, 而 CLX 则用于建立可以同时 Windows 和 Linux 操作系统上运行的应用程序。

4.1.5 Delphi 的对象检查器与应用

Delphi 的对象检查器 (Object Inspector) 用于设置或监视窗体中组件的属性和事件。可用 F11 键或用 “View” → “Object Inspector” 菜单打开。对象检查器显示的是在窗体、数据模块或框架中所选定的组件, 否则为当前的窗体 (参见图 4.2)。

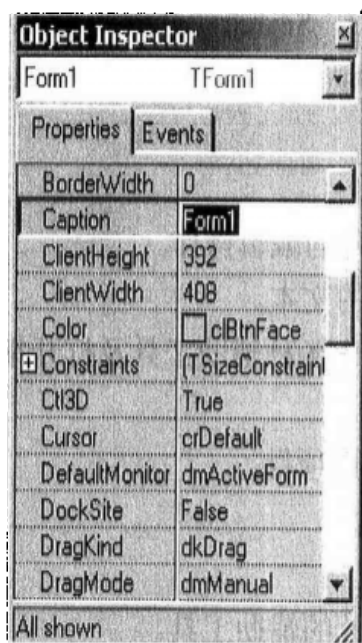


图 4.2 对象检查器

对象检查器由对象选择器、属性标签页和事件标签页组成。

对象选择器: 位于对象检查器的顶部, 可从其下拉框中选择已经设置的组件。

属性标签页 (Properties): 用于在程序设计阶段设置组件的属性。

事件标签页 (Events): 事件是指组件对某个消息的反应, 例如在按钮上单击鼠标左键, 就会触发该按钮的 OnClick 事件。组件所执行的动作都要通过某个事件来实现, 每个事件的代码相当于一个子程序。为某个组件的某种事件的动作编制程序时, 先要选定该组件, 然后在事件标签页中选择对应的事件, 双击该事件右边的空白框, 可以打开对应的代码编辑器。此时, 光标处于编制程序的正确位置, 设计人员可以开始编写该事件的处理程序。

4.1.6 Delphi 的窗体与应用

窗体是 Windows 应用程序中的可视化组件和非可视化组件的容器。在创建一个 Windows 应用程序时, Delphi 自动建立一个窗体, 在设计时可对窗体的大小和位置进行调整。将鼠标置于窗体的上下左右边界线处, 光标成为双向箭头时, 就可改变窗体的高度或宽度。用鼠标左键选中窗体标题栏, 可在屏幕上将窗体拖拉到合适的位置。

窗体上的组件位置及尺寸可以调整。选中窗体上的组件, 可以拖动组件在窗体上移动。将鼠标置于所选中组件的边框黑方块上, 在光标变成双向箭头时, 可以改变组件的大小。

4.1.7 Delphi 的代码编辑器与应用

Delphi 的代码编辑器是一个功能强大的, 用于对组件事件进行编码的工具。在代码

编辑器中输入一个组件的名称和一个圆点后，就会在代码编辑器中出现包含该组件所有属性和方法的提示列表框，选中列表框中某个属性和方法后，代码编辑器就会自动将其插入代码行中（参见图 4.3）。

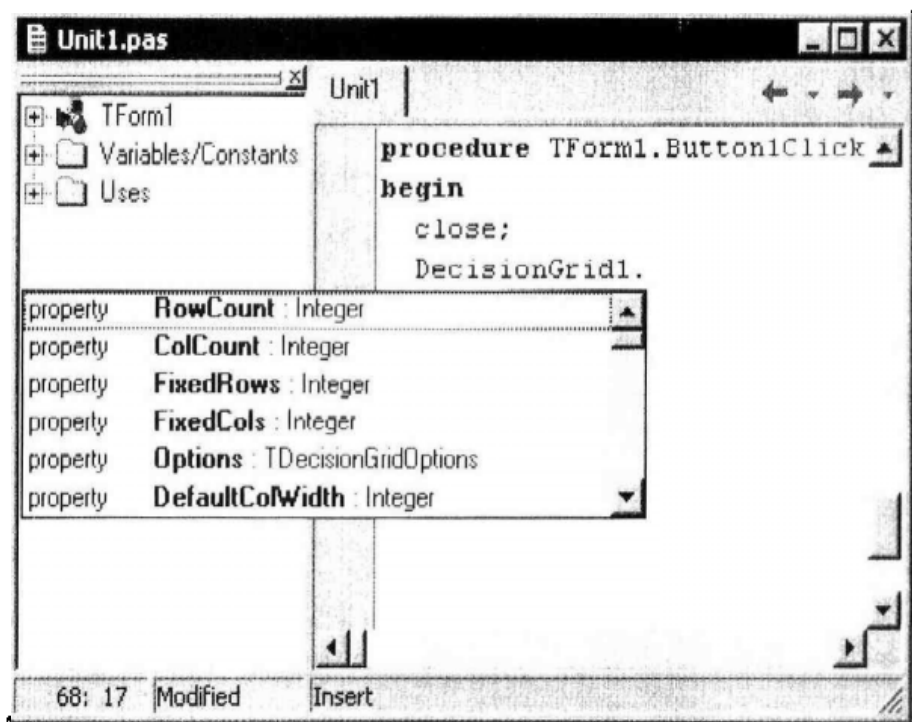


图 4.3 代码编辑器

在代码行中输入方法（函数或过程）的名称和左括号后，就会出现参数提示框，提示需要输入的参数类型。

4.1.8 Delphi 应用程序的设计过程

在应用 Delphi 设计应用程序时，可以按照建立工程项目文件、添加组件、确定组件属性、为组件编写代码、保存文件和运行程序等步骤进行。

1. 建立工程项目文件

Delphi 是利用工程项目（Project）来管理组织应用系统的开发。当一个应用工程项目建立且编译以后，就形成了与工程项目同名的后缀为 .exe 的 Windows 应用程序。

在一个 Windows 应用程序中至少包含一个窗体(Form)和与窗体对应的程序单元(Unit)，以及一些与窗体无关的单元。

创建 Windows 应用程序的方法有如下 3 种。

(1) 在 Windows 环境下进入 Delphi 的 Delphi 集成开发环境时，缺省创建了一个工程项目 Project1.dpr，一个窗体 Form1.dfm，一个单元文件 Unit1.pas。

(2) 用 Delphi 的“File”→“New Application”菜单项命令, 创建 Windows 应用程序。

(3) 用 Delphi 的“File”→“New...”→“Appliction”菜单项命令, 创建 Windows 应用程序。

2. 添加组件

Windows 应用程序的设计, 一般要在 Windows 窗体上用组件完成用户界面的设计。而 Windows 应用程序的实现是基于组件构造的。在窗体上添加组件的方法有:

用鼠标在组件板上单击需要的组件, 然后在窗体的合适位置上单击一次, 就可以将组件添加到窗体上;

用鼠标双击组件板上所需要的组件, 组件将添加到窗体的中心位置。

当将需要的组件添加到窗体上后, 可用鼠标将组件分别移到合适的位置上。

3. 确定组件属性

组件在添加到窗体上后, 其属性都采用缺省值。这些属性可在运行时候设置, 也可以在设计时候设置, 或者保持缺省值。如果在设计时候设置, 先要选中设置属性值的组件, 然后在对象检查器中的属性页(Properties)上对设置的属性值进行修改。有的组件属性值只能选择 Delphi 所给定的值, 例如 true 和 false。有的则需要单击省略号按钮进入属性设置框, 进行选择。有的属性值可在属性输入框中自行输入。由于 Delphi 利用组件的 Name 属性来惟一标识组件, 因此, 同一窗体上的组件 Name 属性不能相同。

4. 为组件编写代码

在窗体上添加组件以后, 只是完成了 Windows 应用程序的界面设计。应用程序的功能必须在为组件编写代码以后, 才能实现。为窗体或组件编写代码的过程如下。

(1) 选择代码编写对象

单击需要编写代码的窗体或组件, 其属性与事件将在对象检查器中显示。

(2) 选择事件页

单击对象检查器的 Events 标签页, 空白的预置事件右栏表示该对象尚未添加事件的处理过程。

(3) 选择事件

选中 Events 标签页上需要进行处理的事件, 双击其右边空白栏; Delphi 自动在单元文件中添加该事件的声明, 且将光标置于代码编辑器中该事件处理过程设计处——在 begin 和 end 代码之间。

(4) 编写代码

按照事件处理的需要编写处理语句。

5. 保存文件

使用 Delphi 的 File|Save All 菜单命令，将工程文件与单元文件保存在合适的目录下。

6. 编译、运行程序

使用 Delphi 的“Project”→“Compile”菜单命令、“Project”→“Build”菜单命令或直接运行程序，都可调用 Delphi 的编译器。编译完成后，生成后缀为.exe 的可执行文件。在生成后缀为.exe 的可执行文件后，可以脱离 Delphi 的集成开发环境，直接在 Windows 环境中运行该应用程序。如果要在 Delphi 的集成开发环境中运行程序，可以选用 Delphi 的 Run|Run 菜单命令或按 F9 键。

可以使用本章末的引导案例，依据以上所介绍的 Windows 应用程序设计过程，设计数据仓库应用系统。

4.2 Delphi 中的数据仓库组件

Delphi 中的数据仓库组件主要由组件板上的决策立方体（Decision Cube）组件组中的决策立方体（DecisionCube）组件、决策查询（DecisionQuery）组件、决策源（DecisionSource）组件、决策中枢（DecisionPivot）组件、决策栅格（DecisionGrid）组件和决策图表（DecisionGraph）组件组成（参见图 4.4）。

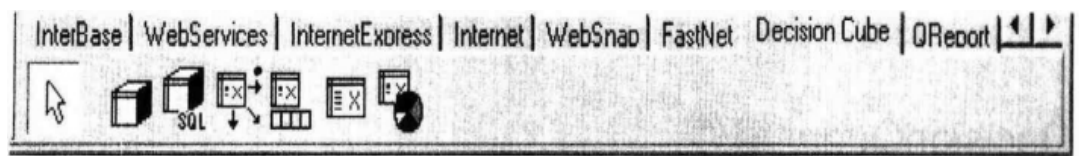


图 4.4 Decision Cube 组件组

Decision Cube 组件组可与数据库基本表进行连接，反映数据的变化情况，对数据表进行数据统计、分析和图形显示。Decision Cube 组件组的应用十分灵活，而且功能十分强大，是一个良好的数据仓库应用系统开发工具。利用这些工具所开发出来的应用系统，才能使数据仓库真正用于对管理决策的支持。而且可将数据仓库应用系统与业务信息系统的开发应用整合在一起，使数据仓库的应用与其他业务系统构成一个整体，不必另行设计和开发系统。

用 Delphi 进行数据仓库应用系统的开发，需要依靠 Decision Cube 组件组中的各种组件的相互配合（参见图 4.5）。其中的 DecisionQuery 组件处于与物理数据库进行交往的底层。DecisionCube 组件主要对 DecisionQuery 组件从物理数据库中所取得的数据进行分析，并且将其转变为一个多维表的结构，然后通过 DecisionSource 组件提交给 DecisionGrid 等

组件显示出来。DecisionSource 组件在数据仓库的应用中起到了一个桥梁作用，将 DecisionCube 组件处理后的数据提交给 DecisionPivot、DecisionGrid 和 DecisionGraph 组件。DecisionPivot 组件主要用于对数据仓库中的数据操作进行导航，它提供一些简单明了的按钮，便于用户对数据进行操作。DecisionGrid 组件主要用于对数据的分析结果进行显示，还可改变数据显示区的颜色，以及数据的行、列排放方式，即实现多维数据集的旋转分析。DecisionGraph 组件用来将所分析的数据以可视化的方式进行显示，有利于用户对数据进行直观的分析。

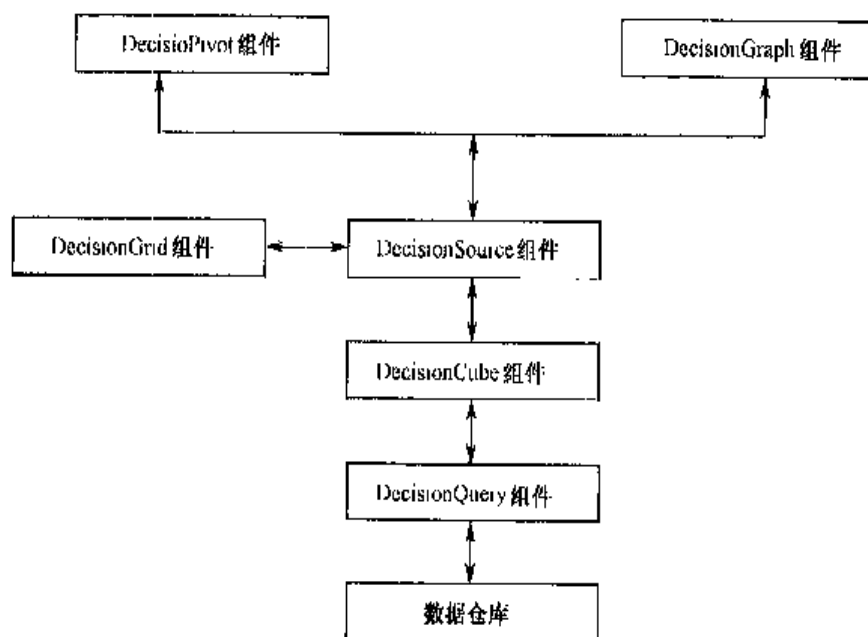


图 4.5 Delphi 数据仓库各种组件关系图

4.3 DecisionQuery 组件

DecisionQuery 组件专门用于为 Decision Cube 组件板上的组件提供一个数据集，并且通过在 DecisionQuery 组件的 SQL 属性中设置 SQL 语句，以进行数据的多维查询。在设置 SQL 语句时，必须要有 SUM 等计算字段；在查询两个以上字段时，还需要用 Group By 进行分组统计。DecisionQuery 组件在从数据集中获取数据后，就将数据传递给 DecisionCube 组件。

DecisionQuery 组件继承于 Tquery，因此 Tquery 中的一些属性和方法都可以使用。

4.3.1 DecisionQuery 组件的主要属性

1. Active 属性

Active 属性用于激活数据集。只有在 Active 为 true 时，数据集中的数据才能被传递

到 Decision Cube 组件组的其他组件中进行分析 and 处理。

2. AutoRefresh 属性

AutoRefresh 属性用于确定当数据库中数据变化时，是否自动更新表格中显示的数据。

3. CachedUpdates 属性

CachedUpdates 属性为 True 时，激活数据集的缓冲区更新。

4. DatabaseName 属性

DatabaseName 用于指定所用数据库的名字或别名。

5. DataSource 属性

DataSource 属性用于指定不是当前数据集使用的另外一个数据源，当前字段可以从中提取数据。

6. Filter 属性

Filter 用于设置筛选条件，也可以在 SQL 属性中用 Where 代替。

7. Filtered 属性

Filtered 属性与 Filter 连用，为 True 时，可以用 Filter 属性设置筛选条件。

8. ParamCheck 属性

如果在运行中改变 SQL 属性时，ParamCheck 属性可以决定是否应该重新产生一个查询的参数列表。

9. Params 属性

Params 用于设置在 SQL 属性中所用的参数。

10. RequestLive 属性

当 RequestLive 属性值为 True 时，才能修改数据集数据，否则只能阅读数据。

11. SQL 属性

SQL 属性用于存放 Select 语句，以建立数据集。

12. UpdateObject 属性

在 CachedUpdates 为 True 时, 可以指定一个 TUpdateSQL 更新一个只读的结果集。

4.3.2 DecisionQuery 组件的主要方法

1. ExecSQL 方法

ExecSQL 方法用于执行 INSERT, UPDATE, DELETE 和 CREATE TABLE 等 SQL 语句。

2. Free 方法

Free 方法用于撤消一个对象, 并且将其所占用资源释放出来。

3. GetDetailLinkFields 方法

GetDetailLinkFields 方法用于在列表中显示主从表的关联字段。

4. Open 方法

Open 方法用于执行 SELECT 语句, 其他 SQL 语句必须用 ExecSQL 语句执行。

5. ParamByName 方法

ParamByName 方法用于为 SQL 语句设置参数值。

6. Prepare 方法

Prepare 方法用于在 ExecSQL 执行前使用, 用于通知 BDE 和远程数据库服务器为当前查询分配资源, 并且进行优化处理。

7. UnPrepare 方法

UnPrepare 方法用于释放在 Prepare 方法为当前查询所分配的资源。

4.3.3 DecisionQuery 组件的主要事件

DecisionQuery 组件的主要事件分如下三大类。

1. After 类

After 类事件在某些情况出现后或调用某个方法后, 会被触发。

2. Before 类

Before 类事件在某些情况出现前或调用某个方法之前，会被触发。

3. On 类

On 类事件在某些情况出现时或调用某个方法时，会被触发。

这些事件的选用需要根据对象的事件发生的时间顺序来确定。例如某些操作需要在 Open 事件发生后执行，就需要在 AfterOpen 事件中编写这些操作的程序；如果需要在 Open 事件发生前执行，就需要在 BeforeOpen 事件中编写这些操作的程序。

4.3.4 利用 DecisionQuery 组件选择需要分析的数据维

先将 DecisionQuery 组件放置在窗体上，然后鼠标右键单击 DecisionQuery 组件，调出快捷菜单，选择其中的 DecisionQuery Editor... 菜单项，打开“DecisionQuery Editor”对话框（见图 4.6），选择其中的 Dimensions/Summaries 页。

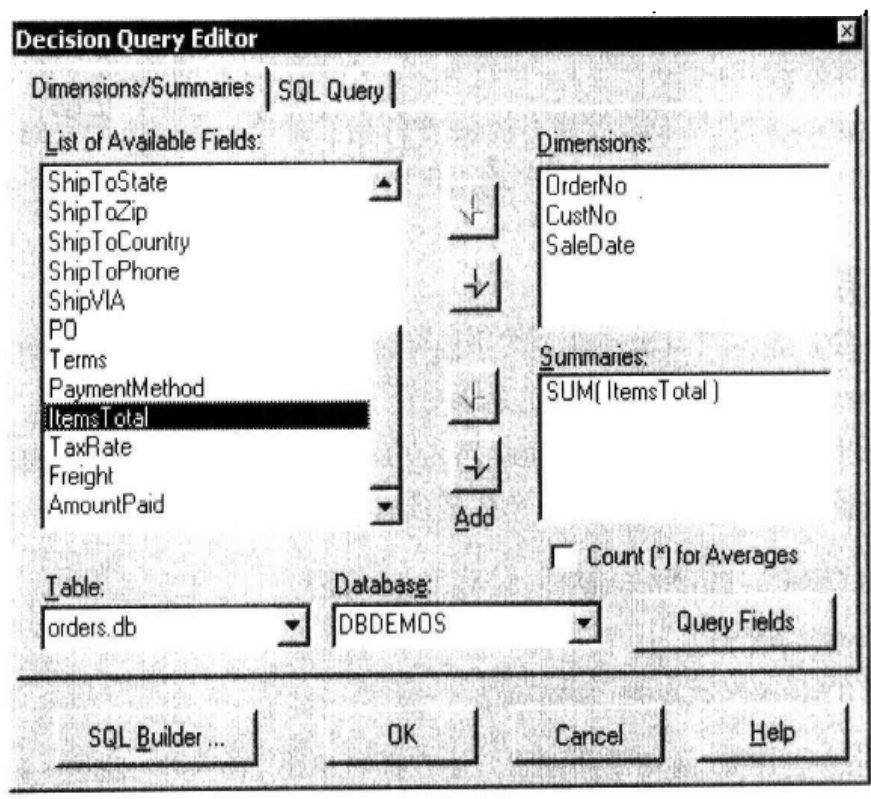


图 4.6 “Decision Query Editor”对话框

1. 选择 Database 的数据库别名

选择 Database 中的数据库别名，这里选了 Delphi 自带数据库 DBDEMOS。

2. 选择数据库表

在 Table 中选择数据库中的数据表, 这里选择了 Orders.db 数据表。

在 List of Available Fields 列表框中选择需要分析的数据维, 即准备进行数据的多维分析。所选择的维的个数必须小于等于在 Decision Cube 组件中的 MaxDimension 属性中所设置的值, 否则系统会出现错误提示。这里选择了 OrderNo, CustNo 和 SaleDate 三个维进行分析。应注意, 只有移入到 Dimensions 列表框中的字段才作为维处理, 因此这里所移动的字段常常是日期、地区等字段。

4.3.5 利用 DecisionQuery 组件选择数据的分析公式

1. 选择分析字段

在 DecisionQuery Editor 编辑窗口的 List of Available Fields 列表框中选择所要计算的字段, 也就是用户希望观察的事实表中的数据。这里选择 Amt_Paid 字段, 将其移入 Summaries 列表框中。

2. 选择计算公式

单击 Summaries 列表框左边的向右箭头, 在弹出的菜单中选择所需要的计算公式 sum, average, count, 在 Summaries 列表框中将出现所设置的公式, 这里选择了 sum。

3. 设置所有需要计算的字段

重复 1.~2.步, 直到所有需要计算的字段全部设置完毕, 在 Summaries 列表框下面的 Count (*)for Averages 复选框, 用于决定是否对所有字段值计算平均值。

完成以上操作后, 可以选择 Decision Query Editor 编辑窗口的 SQL Query 页, 其中的 Query Text 列表框将出现对应以上操作所生成的 SQL 语句:

```
SELECT OrderNo, CustNo, SaleDate, SUM( ItemsTotal )  
FROM "orders.db"  
GROUP BY OrderNo, CustNo, SaleDate
```

必要时可对这些 SQL 语句重新编辑。用户还可单击 Decision Query Editor 编辑窗口的“SQL Builder”按钮, 进入 SQL Builder 操作窗口, 进行 SQL 的可视化生成。

4.4 DecisionCube 与 DecisionSource 组件

DecisionCube 组件用于完成数据仓库的立方体创建, 完成维、观察数据和数据集内

字段的映射关系，即用多维结构对数据集进行数据组织、联系和分析。

4.4.1 DecisionCube 组件的主要属性

1. DataSet 属性

DataSet 属性用于设置需要分析的数据集。该数据集提供数据表中所选择的维数和汇总数据，这里选择 TdecisionQuery 组件的 Name 值 DecisionQuery1。

2. DimensionMap 属性

DimensionMap 属性用于通过 DimensionMap 属性值，访问分析数据表中的数据维和统计维。数据维可以是希望分析的产品、地域、顾客和时间等，统计维一般是产品的销售数量、金额等。单击 DimensionMap 属性右边的省略号可以调出 DecisionCube 组件的属性编辑对话框（见图 4.7）（也可以用鼠标右键单击 DecisionCube 组件，选择弹出式菜单中的 DecisionCube Editor 菜单项调出编辑窗口），对要分析的数据字段和统计字段进行编辑。

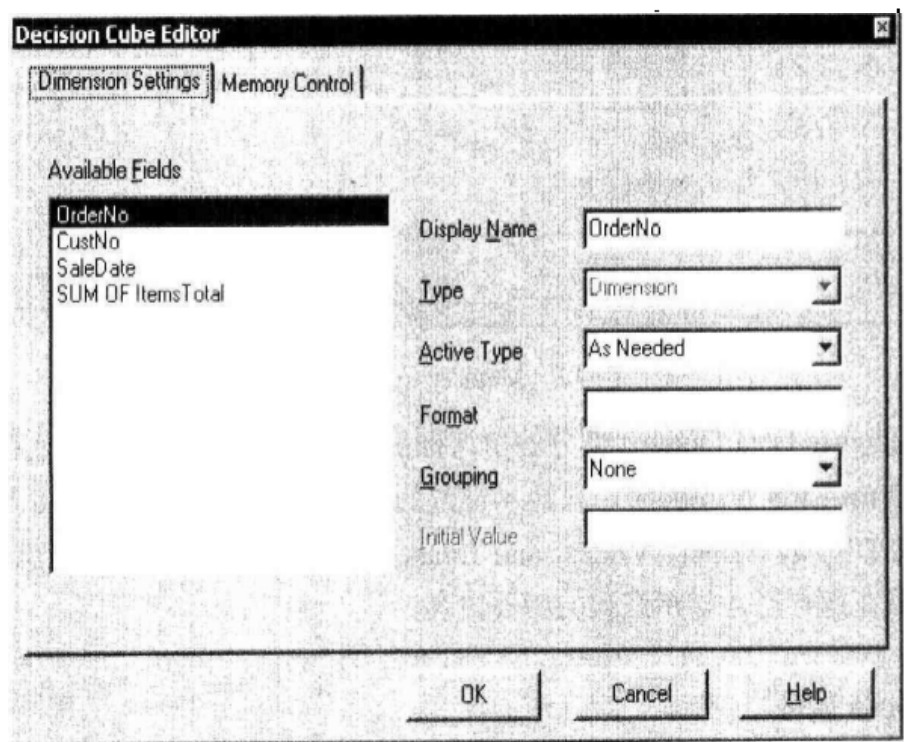


图 4.7 DecisionCube 组件属性编辑对话框

在 DecisionCube 组件属性编辑窗口中的 Dimension Settings 页，用于设置对应数据集中所用字段的一些属性。标签页左边是当前所选定的 4 个字段，右边是在左边列表框中所选定的字段属性。字段属性包含字段在人机界面上所显示的名字 Display Name；维数类型 Type（主要有 Dimension, Sum, Average, Count 等值）；是否激活所设置的类型 Active

Type (主要有 Active, As Needed, Inactive 等值), 使用的格式 Format, 分组方式 Gruoping (有 None, Year, Quatrter, Month, Single Value 等值) 以及所设置的初值 Initial Value (与 Gruoping 所选择值有关的初始值)。

DecisionCube 组件属性编辑对话框中的 Memory Control 页(见图 4.8)分 Cube Maximums 和 Designer Data Options 两部分。Cube Maximums 用于设置 DecisionCube 可以使用的最大内存, 在 Maximum 行中设置统计信息的最大维数 Dimensions、汇总字段 Summaries 和单元格 Cells 的数量。如果 Cells 值为 0, 则 Cells 的数量由维数和汇总数决定。

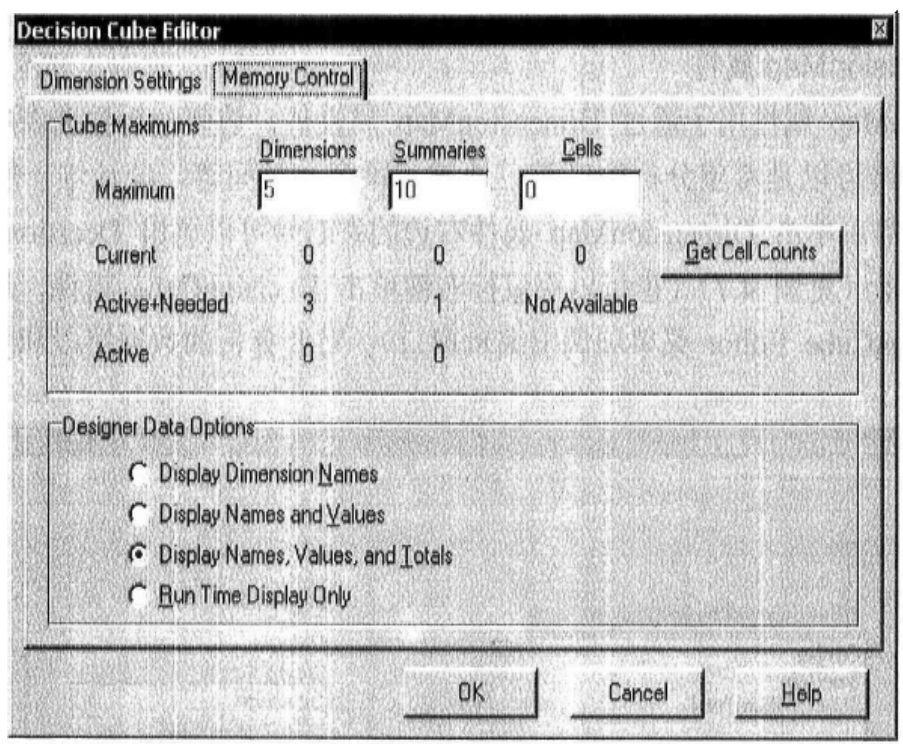


图 4.8 Memory Control 页

在选中 Designer Data Options 部分的单选框后, 可以确定设计时所使用的数据选项。选中 Display Dimension Names 时, 只显示维数的名字; Display Names and Values 显示维数的名字和值; Display Names, Values, and Totals 显示维数的名字、值和汇总数; Run Time Display Only 只在运行时才显示维数的相关内容。

3. MaxCells 属性

MaxCells 属性用于设置 Tdecision Cube 组件显示最大单元格数, 作用同图 4.8 中的 Cells 编辑框。

4. MaxDimensions 属性

MaxDimensions 属性用于设置 Tdecision Cube 组件显示最大维数, 作用同图 4.8 中的 Dimensions 编辑框。

5. MaxSummaries 属性

MaxSummaries 属性用于设置 Tdecision Cube 组件显示最大汇总字段数，作用同图 4.8 中的 Summaries 编辑框。

6. ShowProgressDialog 属性

当 ShowProgressDialog 属性设置为 true 时，在激活数据集时将显示一个进度条，指示计算进度。

4.4.2 DecisionCube 组件的主要方法

1. GetDetailSQL 方法

GetDetailSQL 方法返回一个能够产生 DecisionCube 组件数据集的 SQL 说明，该 SQL 语句可以描述和查询 DecisionCube 组件数据集中的记录。

2. GetDimensionName 方法

GetDimensionName 方法用于获取维数的名字。

3. GetMemoryUsage 方法

GetMemoryUsage 方法用于获取 DecisionCube 组件已经使用的内存数量。

4. GetSQL 方法

GetSQL 方法可以返回一个 SQL 语句，用于查看 DecisionCube 组件使用的部分数据。

5. GetSummaryName 方法

GetSummaryName 方法用于返回汇总字段的名称。

6. ShowCubeDialog 方法

ShowCubeDialog 方法用于在系统运行时显示 DecisionCube 组件的属性编辑窗口，用户可在系统运行时重新设置使用的维数和汇总字段。

4.4.3 DecisionCube 组件的主要事件

1. AfterClose 事件

在表格中数据显示关闭后，触发 AfterClose 事件。

2. AfterOpen 事件

在表格中数据被激活后，触发 AfterOpen 事件。

3. BeforeClose 事件

在表格中数据显示之前，触发 BeforeClose 事件。

4. BeforeOpen 事件

在表格中数据被激活之前，触发 BeforeOpen 事件。

5. OnLowCapacity 事件

当表格中数据使用的内存超过 Capacity 属性所设置的内存大小时，触发 OnLowCapacity 事件。

6. OnRefresh 事件

当表格的维数变化时，触发 OnRefresh 事件。

4.4.4 DecisionSource 组件的主要属性

DecisionSource 组件起到将 DecisionCube 组件与 DecisionGrid, DecisionGraph 与 DecisionPivot 组件连接的作用。DecisionSource 组件的主要属性有如下 5 个。

1. ControlType 属性

ControlType 属性用于设置如何根据 DecisionPivot 组件提供的消息，决定行或列的打开方式。其值为 xtCheck 时可以改变打开的维数，为 xtRadio 时单行或单列的维数总是 1，为 xtRadioEx 时总是打开一维的行和列。

2. DecisionCube 属性

DecisionCube 属性用于指定连接决策数据集的决策立方 DecisionCube 组件。

3. SparseCols 属性

SparseCols 属性用于确定是否显示行所对应的没有汇总数据的列。

4. SparseRows 属性

SparseRows 属性用于确定是否显示列所对应的没有汇总数据的行。

5. CurrentSum 属性

CurrentSum 属性用于指示当前汇总的索引号。

4.4.5 DecisionSource 组件的主要方法

1. CloseDimIndexRight 方法

CloseDimIndexRight 方法可以关闭一个维的显示。

2. DrillDimIndex 方法

DrillDimIndex 方法可以将所有值汇总到一起。

3. GetDataAsString 方法

GetDataAsString 方法可将一个单元格中数据以字符串格式表示。

4. GetDataAsVariant 方法

GetDataAsVariant 方法用于获取单元格中数据值。

4.4.6 DecisionSource 组件的主要事件

1. OnAfterPivot 事件

在决策中枢改变后，触发 OnAfterPivot 事件。

2. OnBeforePivot 事件

在决策中枢改变前，触发 OnBeforePivot 事件。

3. OnLayoutChange 事件

当决策中枢的改变被激活时，触发 OnLayoutChange 事件。

4. OnNewDimensions 事件

当 DecisionCube 组件所提供的数据改变时，触发 OnNewDimensions 事件。

5. OnStateChange 事件

当 DecisionCube 组件属性改变时，触发 OnStateChange 事件。

6. OnSummaryChange 事件

当 CurrentSum 属性值改变时，触发 OnSummaryChange 事件。

4.5 DecisionPivot 组件、DecisionGrid 组件与 DecisionGraph 组件

利用前面所介绍的决策查询 (DecisionQuery) 组件可以创建所要进行决策分析的数据集，然后利用决策立方体 (DecisionCube) 组件建立映射关系，且用决策源 (DecisionSource) 组件连接到决策数据集后，可以利用决策中枢 (DecisionPivot) 组件、决策栅格 (DecisionGrid) 组件和决策图表 (DecisionGraph) 进行决策的分析，即对数据集中的数据进行决策分析。

决策中枢 DecisionPivot 组件用于对表格中的数据进行导航，可对数据维进行展开或合并。DecisionGrid 组件以多维表格方式显示数据集中所分析的数据，可以改变维数，并且可对数据显示区域的颜色及行、列的排列方式进行调整。DecisionGraph 组件以图形方式显示数据集中所分析的数据，图形能够根据统计数据的变化而改变。

4.5.1 DecisionPivot 组件的主要属性

1. DecisionSource 属性

DecisionSource 属性用于指定一个 DecisionSource 组件为所操作的数据源。

2. GroupLayout 属性

GroupLayout 属性用于指定 DecisionPivot 组件的按钮排列方式：值为 xtHorizontal 时，水平排列；为 xtLeftTop 时，从左上角开始排列；为 xtVertical 时，垂直排列。

3. Visible 属性

Visible 属性用于确定 DecisionPivot 组件在运行中是否可见。

4.5.2 DecisionPivot 组件的主要方法

1. Hide 方法

Hide 方法可以隐藏 DecisionPivot 组件。

2. Show 方法

Show 方法可以显示 DecisionPivot 组件。

4.5.3 DecisionPivot 组件的主要事件

1. OnEnter 事件

当 DecisionPivot 组件被激活时，触发 OnEnter 事件。

2. OnExit 事件

当 DecisionPivot 组件从激活状态进入非激活状态时，触发 OnExit 事件。

3. OnClick 事件

当 DecisionPivot 组件被单击时，触发 OnClick 事件。

4. OnDblClick 事件

当 DecisionPivot 组件被双击时，触发 OnDblClick 事件。

5. OnDragDrop 事件

当拖曳 DecisionPivot 组件并释放时，触发 OnDragDrop 事件。

6. OnDragOver 事件

当在 DecisionPivot 组件上拖曳鼠标时，触发 OnDragOver 事件。

7. OnEndDrag 事件

当在 DecisionPivot 组件上停止拖曳鼠标时，触发 OnEndDrag 事件。

8. OnResize 事件

当改变 DecisionPivot 组件大小时，触发 OnResize 事件。

9. OnStartDrag 事件

当在 DecisionPivot 组件上开始拖曳鼠标时，触发 OnStartDrag 事件。

4.5.4 DecisionGrid 组件的主要属性

1. CaptionColor 属性

CaptionColor 属性用于确定维数名所在单元格的背景颜色。

2. DataColor 属性

DataColor 属性用于确定汇总数据所在单元格的背景颜色。

3. DataSumColor 属性

DataSumColor 属性用于确定维数汇总和数所在单元格的背景颜色。

4. DecisionSource 属性

DecisionSource 属性用于确定 Decision Grid 组件所对应的 DecisionSource 数据源。

4.5.5 DecisionGrid 组件的主要方法

1. CellDrawState 方法

CellDrawState 方法可以从表格的数据单元格中获取数据。

2. CellRect 方法

CellRect 方法可以获取选定单元格的屏幕坐标。

3. CellValueArray 方法

CellValueArray 方法可以返回所有包含汇总数据的单元格所对应维数字段的值。

4.5.6 DecisionGrid 组件的主要事件

DecisionGrid 组件的事件均以 On 开头，这些事件只能在出现某种情况或调用某种方法时触发。从 DecisionGrid 组件所对应对象观察器中的事件页 (Events) 中的事件名称可以发现，这些事件主要由鼠标、键盘在 DecisionGrid 组件上的单击、双击和拖曳为主。

4.5.7 DecisionGraph 组件的主要属性

1. AxisVisible 属性

AxisVisible 属性用于确定是否显示 Decision Graph 组件的坐标轴。

2. BackColor 属性

BackColor 属性用于确定 DecisionGraph 组件的背景的颜色。

3. BackImage 属性

BackImage 属性用于确定 DecisionGraph 组件的图像。选中该属性框后，单击右边的省略号框，调出图像编辑框 (Picture Editor) (见图 4.9)，利用其中的“Load...”钮选择合适的图像。

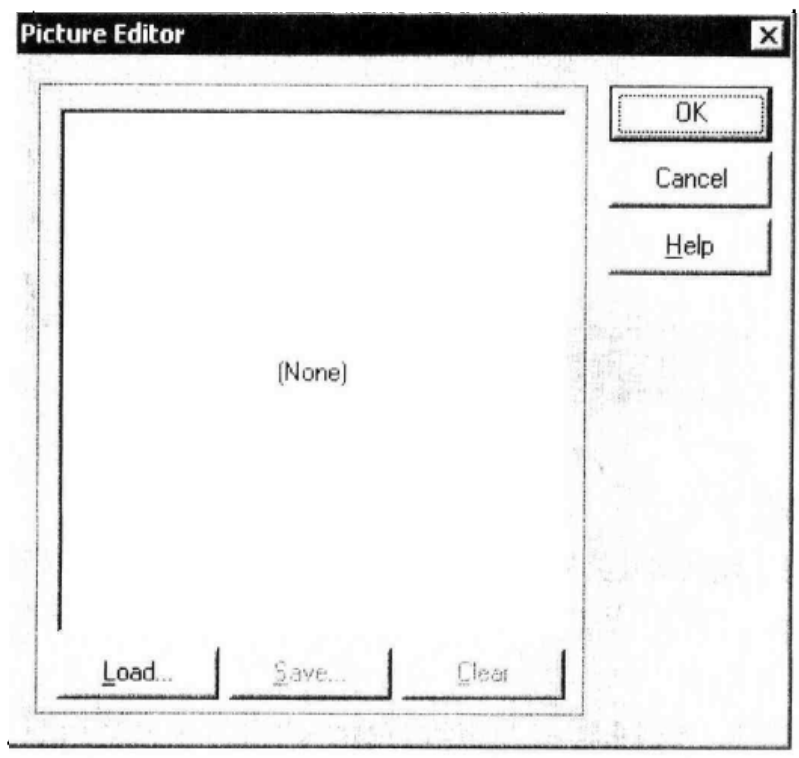


图 4.9 图像编辑框

4. DataSource 属性

确定 Decision Graph 组件所对应的 DataSource 数据源。

5. Fram 属性

确定 Decision Graph 组件框架属性。选中该属性框后，单击右边的省略号框，调出框架编辑框 (Border Color Editor)，可用 Visible 决定是否显示框架、Width 决定框架的宽度、Style 决定框架的风格、Color 决定框架的颜色。

6. Chart 的属性

在 DecisionGraph 组件中还有许多关于 Chart 的属性，可用鼠标右键单击 Decision Graph 组件，从弹出菜单中选择 Editor Chart... 菜单项进行设置。在调出的 Chart 属性编辑对话框 (图 4.10) 中有 Series, General, Axis, Titles, Legend, Panel, Paging, Walls 和 3D 共 9 个标签页，分别对表格的图形显示方式进行各种设置。

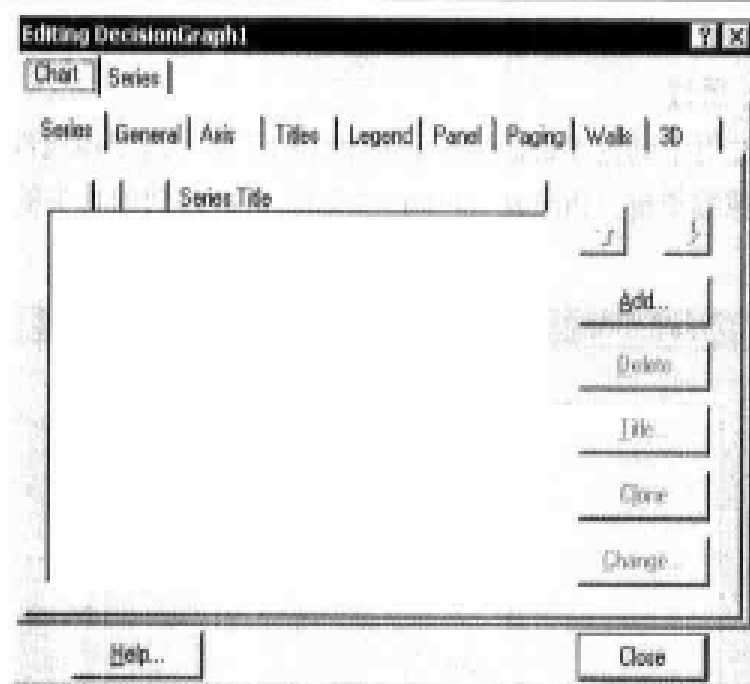


图 4.10 Chart 属性编辑对话框

(1) Series 标签

Series 标签用于设置图表类型，先在 Series 的图形列表框中选择要设置的字段，然后，单击右边的 Change...按钮，从打开的图表类型选择对话框中确定所需图表类型。

(2) General 标签页

在 General 标签页(见图 4.11)上可用 Print Preview 按钮预览图表的打印效果，用 Export 按钮将图表输出到文件或剪贴板上，用 Margins 设置图表上、下、左、右边界的距离，用 Zoom 放大显示图表，用 Allow Scroll 决定是否设置水平与垂直滚动条。

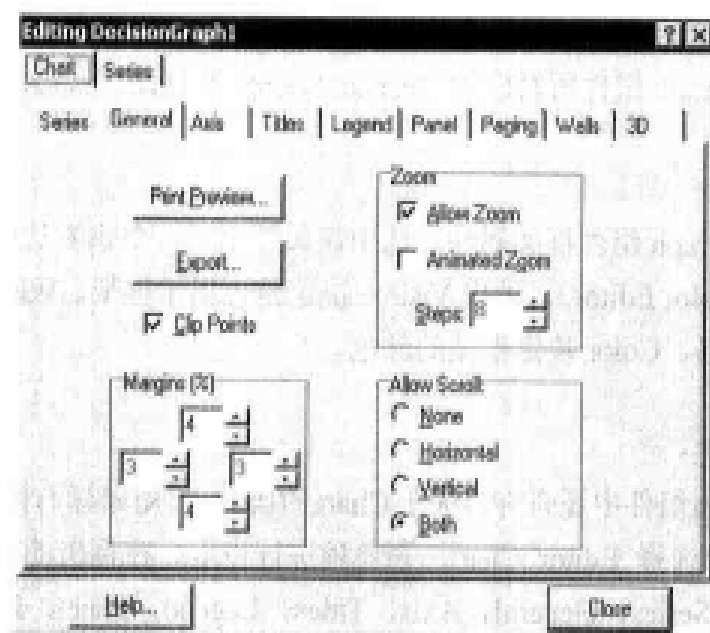


图 4.11 General 标签页

(3) Axis 标签页

Axis 标签页分左右两部分(见图 4.12), 左边的 Axis 部分 5 个单选按钮 (Left, Right, Top, Bottom, Depth) 分别确定当前所设置属性的坐标轴, 右边的 6 个标签页 (Scales, Title, Labels, Ticks, Minor, Position) 设置当前坐标轴的属性。Axis 标签左上的 Show Axis 选项确定是否显示坐标轴, 左下的 Visible 选项确定当前所选坐标轴是否可见。

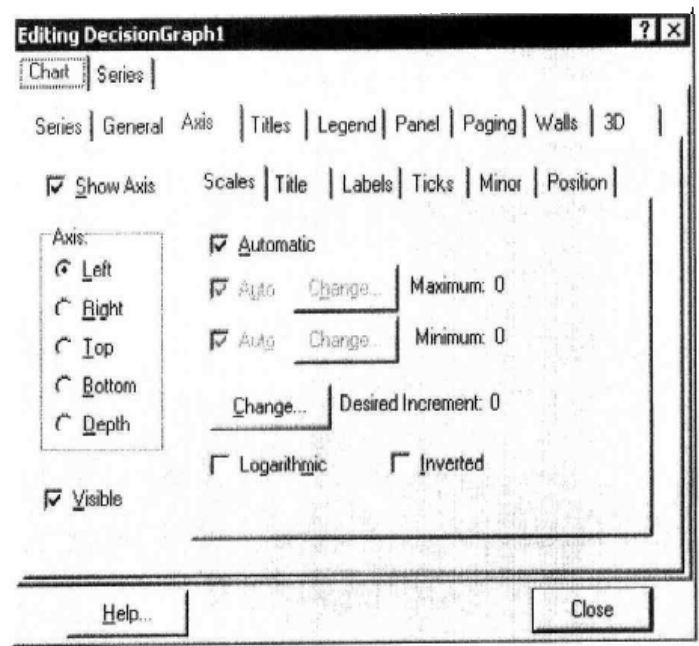


图 4.12 Axis 标签页

(4) Titles 标签页

Titles 标签页(见图 4.13)用于设置图表标题的属性, 图表标题是否可见 (Visible), 标题位置 (Alignment), 背景颜色 (Back Color) 和标题字体 (Font) 等。

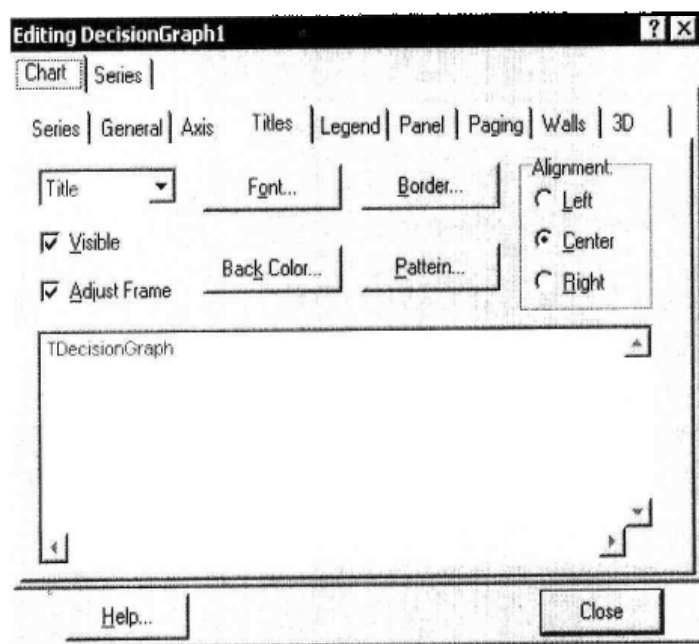


图 4.13 Titles 标签页

(5) Legend 标签页

Legend 标签页 (见图 4.14) 用于设置图表的属性, 例如图表的类型 (Legend Style)、图表中文本内容的风格 (Text Style)、使用的框架 (Frame) 和字体 (Font) 等。

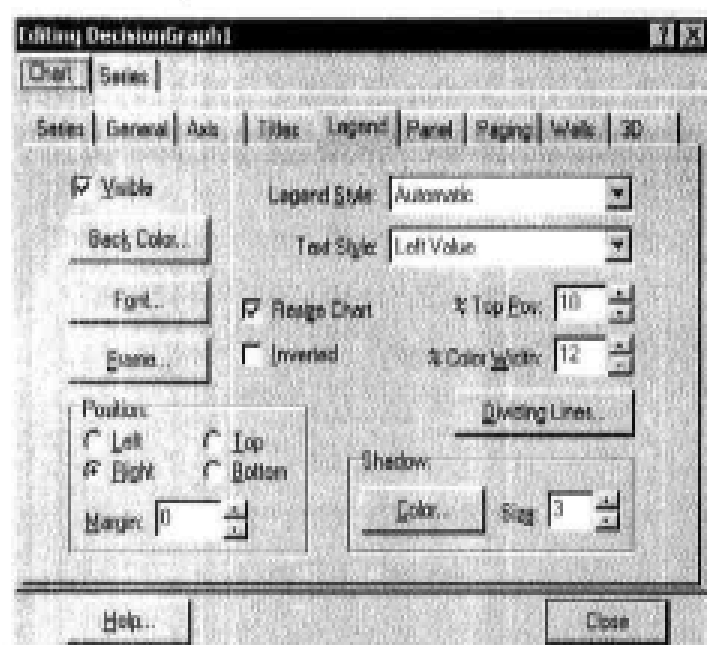


图 4.14 Legend 标签页

(6) Panel 标签页

Panel 标签页 (见图 4.15) 用于设置图表所在面板的属性, 例如面板的凹凸 (Bevel Inner 和 Bevel Outer)、面板的颜色 (Panel Color)、面板的背景图像 (Back Image) 等。

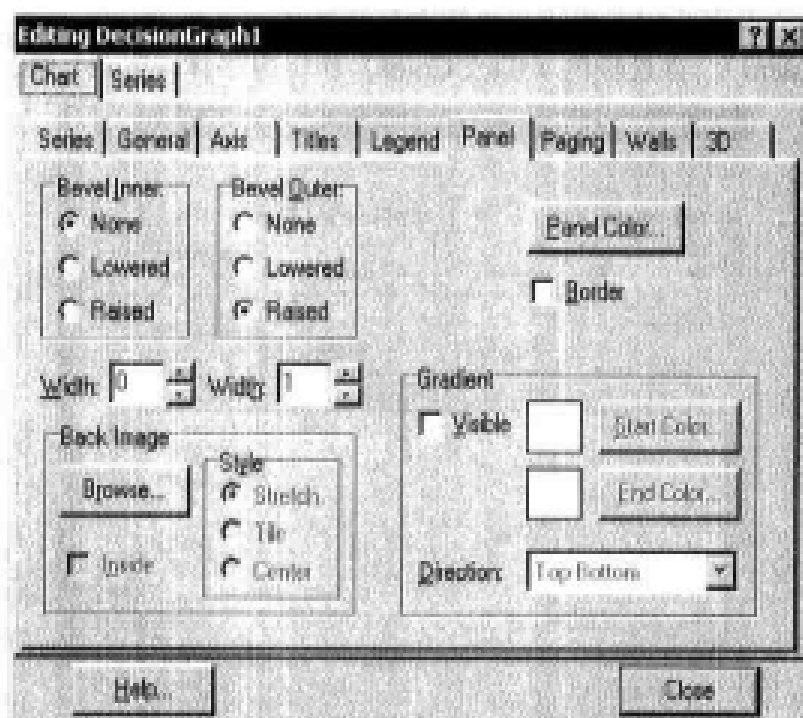


图 4.15 Panel 标签页

(7) Paging 标签页

Paging 标签页（见图 4.16）用于设置图表所在页面的属性，可以用 Points per Page 选项确定图表的显示范围。

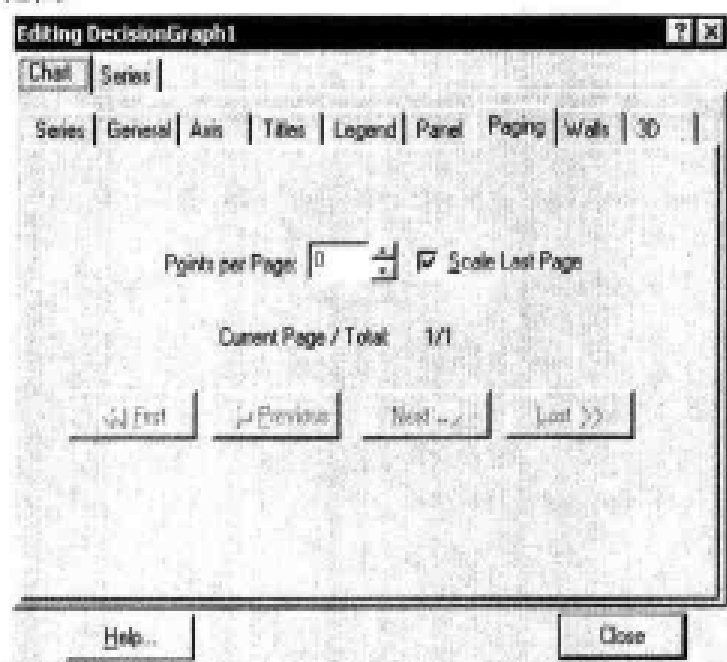


图 4.16 Paging 标签页

(8) Walls 标签页

Walls 标签页（见图 4.17）通过 Left Wall, Bottom Wall 和 Back Wall 三个标签页设置左边、下边和后面的阴影部分的属性，主要是背景颜色（Background）、边界类型（Border）和样式（Pattern）等。

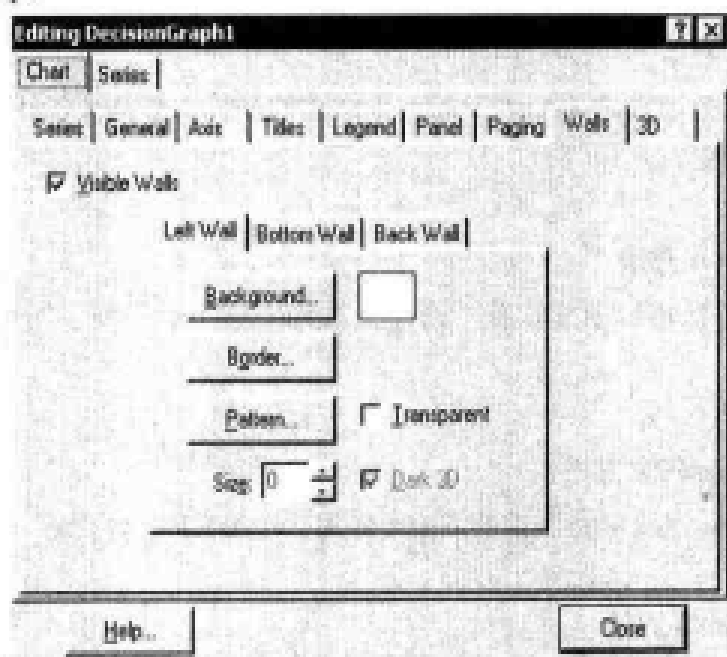


图 4.17 Walls 标签页

(9) 3D 设置

3D 标签页（见图 4.18）设置图表的 3 维显示方式，主要有图表是否以立体方式显示（3Dimensions）、图表立体程度（3D%）、图表的旋转（Rotation）、图表的仰角（Elevation）、图表的水平垂直偏移量（Horiz Offset 和 Vert Offset）等。

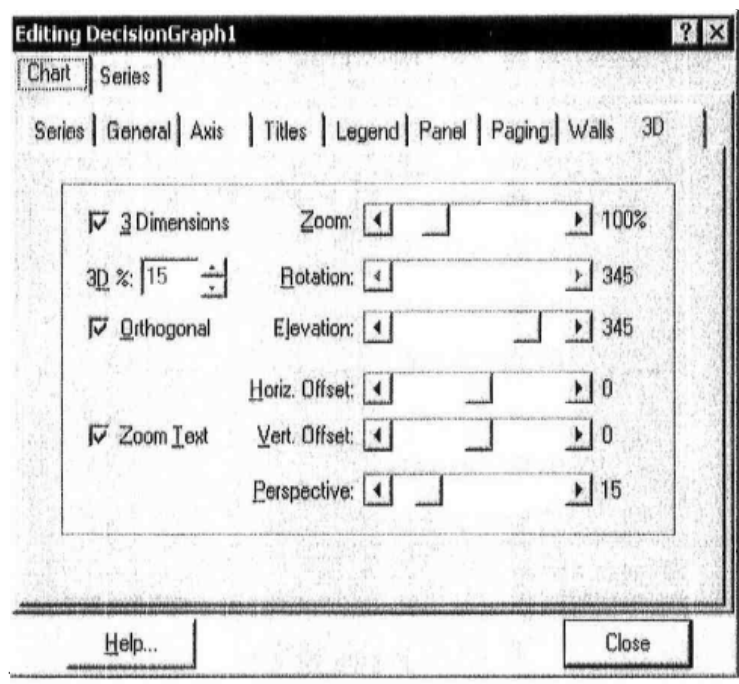


图 4.18 3D 标签页

4.5.8 DecisionGraph 组件的主要方法

1. GetASeries 方法

GetASeries 方法可以返回 Chart 中第一个活动的序列。如果 Chart 没有序列或没有活动的序列，则返回 NIL。

2. GetAxisSeries 方法

GetAxisSeries 方法可以返回依赖于特定坐标轴的第一个序列，如果坐标轴没有序列，则返回 NIL。

3. SaveToBitmapFile 方法

SaveToBitmapFile 方法可将当前的 Chart 作为一个 BMP 文件保存。

4. SaveToMetafile 方法

SaveToMetafile 方法可将当前的 Chart 作为一个 WHF 文件保存。

4.5.9 DecisionGraph 组件的主要事件

DecisionGraph 组件的主要事件均以 On 开头，这些事件在某种状况出现或调用某种方法时触发这些事件。



本章小结

作为信息系统常用的开发工具 Delphi 提供了以 DecisionCube 组件组为主的数据仓库开发工具。由于越来越多的信息系统开始采用 Delphi 进行开发，因此，利用 DecisionCube 组件组开发出的数据仓库应用系统与业务系统具有十分良好的同一平台作业能力。本章主要介绍了用 DecisionCube 组件组中的 DecisionQuery 组件、DecisionCube 组件与 DecisionSource 组件对数据集市（数据商场）的访问与操纵，用 DecisionPivot 组件、DecisionGrid 组件和 DecisionGraph 组件对数据仓库中的数据进行统计采掘和图形采掘，且在案例中介绍了一个数据仓库应用系统的开发。至于数据仓库一般可以采用第3章和第2章中所介绍的 MS SQL 2000 与 Oracle 9i 进行设计。在 Delphi 中可以利用数据库引擎管理器（BDE）、ODBC 管理器和数据库转换器（Data Pump）等工具对 SQL、Oracle 等数据仓库中的数据进行提取与加载。读者可以通过 Delphi 6.0 的有关资料详细了解这些功能。



习题

- 4-1 请说明在 Delphi 中是如何构造数据集市的。
- 4-2 请说明在 Delphi 中是如何对数据集市进行统计类数据采掘的。
- 4-3 请说明在 Delphi 中是如何对数据集市进行图形数据采掘的。
- 4-4 利用 Delphi 开发工具开发一个数据仓库应用系统。
- 4-5 请思考一下，如果在数据仓库应用系统中需要从不同的基表中构造数据集市，应该如何实现？

案例 4.1

数据仓库应用系统设计

利用 Delphi 的 DecisionCube 组件组进行数据仓库应用系统的设计步骤如下。

1. 使用 Delphi 的菜单 “File” → “New” → “Application” 建立一个新工程项目文件。
2. 打开 DecisionCube 组件板, 将组件板上 6 个组件全部添加到窗体 Form1 上, 组件的名字 Name 保持默认值: DecisionQuery1, DecisionCube1, DecisionSource1, DecisionPivot1, DecisionGrid1, DecisionGraph1, DecisionQuery1, DecisionCube1 和 DecisionSource1 组件是系统运行时看不见的非可视组件, 可以任意摆放。DecisionPivot1 的 Align 属性选择 alBottom 值, DecisionGrid1 和 DecisionGraph1 的 Align 属性分别选择 alLeft 和 alRight。将 DecisionPivot1 放置于窗体底部, DecisionGrid1 和 DecisionGraph1 分别置于窗体的左右两边。将 Standard 组件板上的 Button 组件添加到 DecisionGrid1 的左上角。
3. 设置组件的属性值。
首先确定数据集和数据分析维数。将 DecisionQuery1 的 DataName 值设为 DBDEMOS, SQL 值设为 Select EventNo, CustNo, NumTickets, Sum(Amt_Paid) from reservant group by EventNo, CustNo, NumTickets, Active 值设为 True。
然后确定数据采掘工具的数据来源。将 DecisionPivot1、DecisionGrid1 和 DecisionGraph1 的 DecisionSource 属性值都设为 DecisionSource1。Button 的 Caption 属性值改为 “退出”。
4. 用鼠标右键单击 Decision Graph 组件, 从弹出菜单中选择 “Editor Chart...” 菜单项进行 Decision Graph 组件的图形属性设置 (参见图 4.19)。

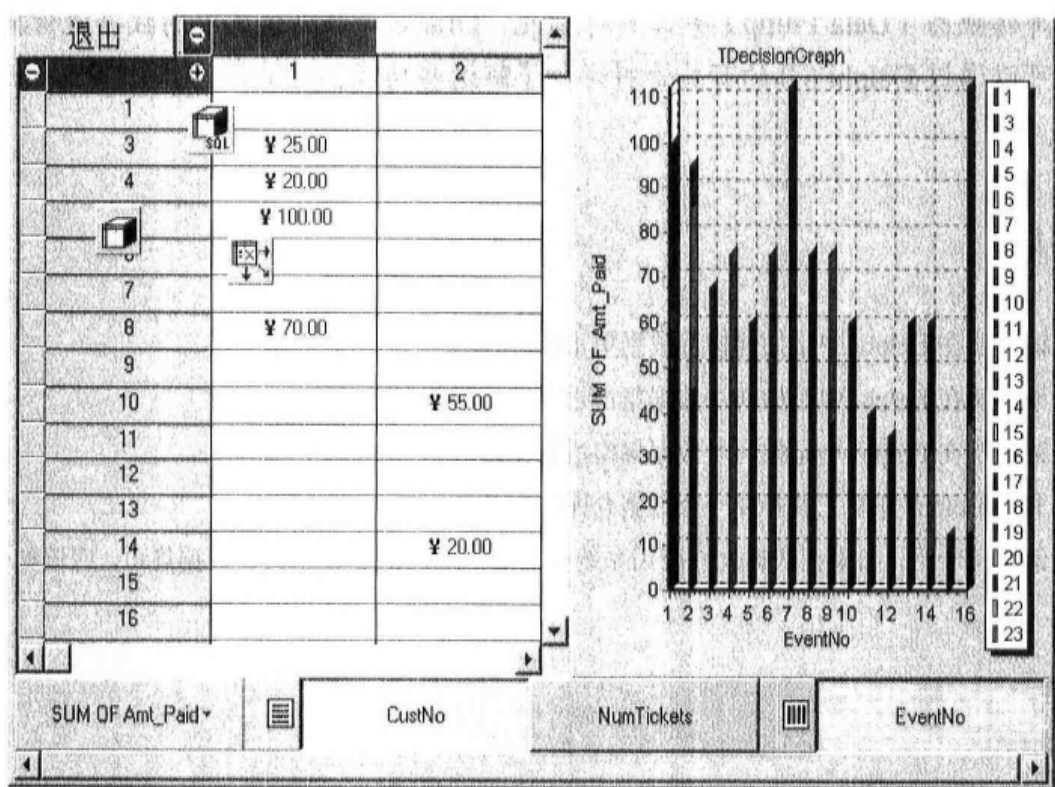


图 4.19 数据仓库应用系统设计界面

5. 为 Button1 的 OnClick 事件编写简单处理程序: Close。
6. 选择 File 菜单中的 Save all 菜单项, 保存数据仓库应用系统。保存时可以分别为工程项目文件 (Project) 和单元文件 (Unit) 重新命名。
7. 从 Delphi 的 Run 主菜单中选择 “Run...” 菜单项或单击 F9 键, 运行应用系统。
8. 在运行中单击 DecisionPivot1 的按钮观察窗体上表格数据和图形的变化。

原书缺页



第 5 章

数据仓库开发模型

引 言

设计一个能够真正支持用户进行决策分析的数据仓库，并非是一件轻而易举的事情。这需要经历一个从现实环境到抽象模型，从抽象模型到具体实现的过程。要完成这一过程，必须依靠各种不同的数据模型。在从现实到抽象的过程中需要依靠概念模型的支持，将现实的决策分析环境抽象成一个概念数据模型。然后，将此概念模型逻辑化。最后，还要将逻辑模型向数据仓库物理模型转化；一旦完成数据仓库的物理模型，就可以说数据仓库的具体实现有了可靠的设计方案。

通过本章学习，可以掌握：

- ◆数据仓库的数据模型
- ◆数据仓库的概念模型
- ◆数据仓库的星型和雪花模型
- ◆数据仓库的中间层逻辑数据模型
- ◆数据仓库的物理模型
- ◆元数据在数据仓库中的作用与管理
- ◆数据仓库的粒度模型

5.1 数据仓库的各种数据模型

在创建数据仓库之时，需要使用各种数据模型对数据仓库进行描述。数据仓库的开发人员依据这些数据模型，才能开发一个满足用户需求的数据仓库。数据仓库的各种数据模型在数据仓库的开发中作用十分明显，主要体现在模型中只含有与设计有关的属性。这样就排除了无关的信息，突出与任务相关的重要信息。使开发人员能够将注意力集中在数据仓库开发的主要部分。模型有更好的适应性，更易于修改。当用户的需求改变时，仅对模型作出相应的变化就能反映这个改变。

数据模型就是对现实世界进行抽象的工具。在信息管理中需要将现实世界的事物及其有关特征转换为信息世界的信息，才能对信息进行处理与管理，这就需要依靠数据模型作为这种转换的桥梁。这种转换经历了从现实到概念模型，从概念模型到逻辑模型，从逻辑模型到物理模型的转换（参见图 5.1）。在数据仓库的开发中同样也要经历概念模型、逻辑模型与物理模型的三级模型开发。图 5.1 为现实与不同模型的变化联系，显示了业务处理系统开发中的数据模型变化关系，这种关系在数据仓库的开发过程中也是同样存在的，只是在具体应用中稍有变化。

现实世界是存在于现实之中的各种客观事物，它反映客观事物及其相互之间的关系。

概念世界是现实情况在人们头脑中的反映，人们需要利用一种模式将现实世界在自己的头脑中表达出来。

逻辑世界是人们为将存在于自己头脑中的概念模型转换到计算机中的实际物理存储过程中的一个计算机逻辑表示模式。通过这个模式，人们可以很容易地将概念模型转换成计算机世界的物理模型。

计算机世界是指现实世界中的事物在计算机系统内的实际存储模式，只有依靠这个物理存储模式，人们才能实现利用计算机对现实世界的信息管理。

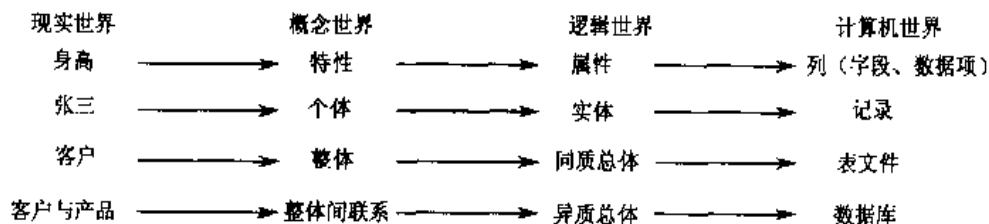


图 5.1 现实与不同模型的变化联系

作为数据仓库设计的模型，除了描绘概念世界的概念模型、描述逻辑世界的逻辑模型和描述计算机世界的物理模型以外，还有元数据模型和数据粒度模型（参见图 5.2）。

数据仓库的设计也就是在概念模型、逻辑模型和物理模型的依次转换过程中实现的。

作为数据仓库的灵魂——元数据模型则自始至终伴随着数据仓库的开发、成长与使用。元数据模型的构建、实施与使用不可能脱离数据仓库的概念模型、逻辑模型与物理模型的设计与实施。数据粒度模型也在数据仓库的创建中发挥着指导者的作用，指导着数据仓库的具体实现。

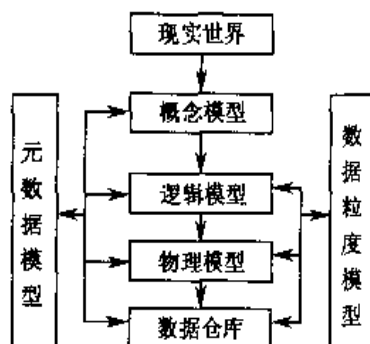


图 5.2 数据模型关系图

5.2 数据仓库概念模型

数据仓库概念模型的设计需要给出一个数据仓库的粗略蓝本，以此为工具来确认数据仓库的设计者是否已经正确地了解数据仓库最终用户的信息需求。在概念模型的设计中，必须将注意力集中在对商务的理解上，保证数据仓库的所有业务处理都被归纳进概念模型。

5.2.1 概念数据模型

在构建数据仓库的概念模型时，可以采用在业务数据处理系统中经常应用的企业数据模型——ER 图（ERD）。这是一种描述组织业务概况的蓝图，包括整个组织系统中各个部门的业务处理及其业务处理数据（参见图 5.3）。蓝图的设计中涉及各个部门所需要的元数据，并且提供本部门所拥有系统的元数据。数据仓库的设计者通过这个蓝图可以了解哪些部门需要哪些共同的数据，而这些数据在目前的各个部门的系统中存在哪些差异，这些差异主要是指数据的定义、属性、业务处理规则等方面的不同点。作为企业的数据模型，不仅能够作为数据仓库设计的蓝图，而且还应作为业务操作数据库的设计依据。只有这样，才能使基于数据模型设计的数据仓库与业务数据处理系统能够得到更好的协调。数据仓库与操作型数据库一样，也存在高层模型（ERD，实体关系层）、中层模型（DIS，逻辑层）和低层模型（物理层）3 个层次数据模型。

尽管在数据仓库的设计过程中，可以采用为业务数据处理系统设计所采用的数据模型作为设计框架，但是在实际设计中用于数据仓库设计的数据模型与业务数据处理系统

的三级数据模型仍有一定的差距。

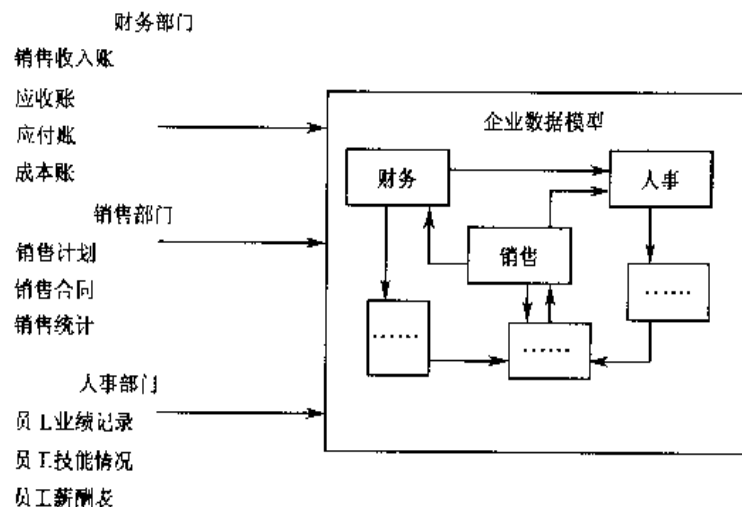


图 5.3 企业数据模型

1. 数据类型的差距

在数据仓库的数据模型中不包含操作型的数据，只包含用户感兴趣的分析数据、描述数据和细节数据，例如商品的销售数量、企业的利润等是常见的分析数据；销售时间、销售地点是用户感兴趣的描述数据；所销售产品的详情、购买商品的客户详情，则是用户感兴趣的细节数据。

2. 数据的历史变迁性

数据仓库的数据模型扩充了码结构，增加了时间属性作为码的一部分。在数据仓库的数据模型中需要反映组织的历史变迁、业务的发展，这就需要用时间属性来描绘这些数据，而这是在业务数据处理系统中不存在的。业务数据处理系统只包含当前数据。

3. 数据的概括性

数据仓库的数据模型中增加了一些衍生数据，专门用于分析的数据仓库数据需要有一些概括性的数据，这些数据在业务处理系统的数据模型中是不需要存在的。

传统的企业数据模型设计，主要采用实体关系图（ERD）。实体关系图用实体以及实体间的关系来描述。这种描述方式在传统业务处理的数据系统设计中得到很好的应用。如果将实体关系图直接用于为数据仓库开发服务的数据模型设计中，就略显不足，因为传统的实体关系图无法表述数据仓库中所需要的分析数据、描述数据和细节数据的关系，无法反映时间属性的存在与作用，更无法表现数据的导出关系。为解决这些问题，可将传统的数据模型构造工具 ERD 稍作修改，将原 ERD 中的实体分成指标实体（事实实体）、

维度实体和详细类别实体（引用实体），这样构造的数据模型才能反映出数据仓库所特有的数据模型特征，而不是与传统数据模型完全雷同的数据模型。利用分类实体所构成的数据模型，可以很直观地观察、理解在数据仓库中的实体以及这些实体之间的关系。

数据模型中的指标实体用矩形表示（见图 5.4（a）），它往往处于数据模型的中心，是数据仓库活动的中心。指标实体往往最后形成数据仓库中的物理实体——事实表，但是在高层模型中是现实世界中的业务处理或某一事件（例如，销售、服务等）的逻辑表示。高层模型中的指标实体体现了在现实世界中的事务处理值，这些值只与每个相关维度的一个点相对应。这些值是从操作型业务系统中所获取的数据，反映用户的真实商业活动状况，是管理人员衡量业务活动好、坏和业务处理困难程度的基础。由于指标实体的数据需要根据现实所发生的状况进行追加，因此，指标实体的数据量将随着时间的推移而日益膨胀，对指标实体数据的管理是数据仓库管理的重点。

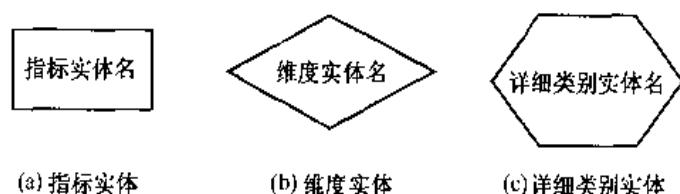


图 5.4 数据模型的实体图形符号

作为高层数据模型中的指标实体应该具有这样一些特性：可为用户提供定量的数据、商务数据或实际数据的基本分析点，是用户进行分析活动的中心和灵魂；包含多种访问指标数据的路径、维度或指标；包含相关的标准数据；构成每个维度中最低一级的类别和一个信息组中的指标；能够扩大成为很大的表格，容纳今后日益增加的数据。

数据模型中的维度实体用菱形表示（见图 5.4（b）），维度实体在数据仓库中主要用于对实体指标的过滤和重新组织提供指导。可将用户对指标实体的查询结果按照维度指标进行筛选，只允许与维度指标相关的数据返回用户。另外，维度实体为数据仓库的整体构建发挥了重要的作用，使不同的指标实体之间建立联系，使指标实体与详细类别实体之间建立联系。这样，就可以使用户对数据仓库进行轻松的访问与浏览。作为维度实体应该具有这样一些特性：可以形成一个维度体系，具备访问和过滤指标实体的能力；提供相关的非标准实体，包括一个完整的维度体系编码、关键词以及相关的表示；可以映射到用户所需要信息的列，在物理数据仓库中是较小的表，可对前台用户的应用程序进行数据填充、指引用户的数据仓库查询分析。

数据模型中的详细类别实体用六角形表示（见图 5.4（c））。详细类别实体在数据仓库中也用物理数据库表示。详细类别实体通常与现实世界中的某个实体相对应，可能是一个客户、一个产品或一个销售点。这些实体以更详细的数据向用户提供决策分析支持，使用户在决策过程中获得有力的帮助。详细类别实体具有终止操作的作用，用户常常通

过维度实体得到指标实体数据，而在操作到详细类别实体时则停止操作。详细类别实体应该具有这样一些特性：包含参考数据和有助于完成指标数据智能的支持信息，提供更定性的数据，与事务结构有映射关系，包含标准的数据结构，数据量比指标实体少，但比维度实体多，数据可能是数值型的、定性的或说明性的。

5.2.2 规范的数据模型

用于业务数据处理系统的数据库设计目标与数据仓库的设计目标有明显差异。传统的数据库设计是基于某个范式的，具有规范化的特点，系统所需要的是快速响应和高效的数据存储。数据仓库为了高效地检索数据信息，通常是不规范化的。通过对数据仓库中包含的结构进行非规范化，可以提高信息的检索性能和可利用性。

数据的规范化是将数据结构分解成最小组成部分的过程。规范化主要强调实现存储的灵活性和高效性，可使规范化的结构占用最小的存储空间，增强数据库的存储效率。从关系数据库开始，这种数据驱动系统使用的是数据的改变，而不是通过数据结构和程序的改变来增强的，人们一直将数据的规范化作为构建数据库系统的惟一目标。数据仓库设计的最终目标乃是实现对大量数据的快速访问。虽然，数据仓库也要努力实现灵活性和高效性，但是在数据仓库设计中通常为提高快速访问效率而牺牲灵活性和存储的高效性。数据仓库的特点是结构越简单，性能就越好。表 5-1 给出了数据仓库的数据和普通数据库系统数据之间的区别。

表 5-1 数据仓库的数据与普通数据库系统数据之间的区别

长期的框架	短期的框架
静态	快速变化
数据通常是汇总的	记录级的访问
特殊查询访问	标准查询访问
定期更新	实时更新
数据驱动	事件驱动

当为一个联机操作系统创建关系数据库时，为实现数据访问的灵活性和高效的数据存储，创建一个第三范式的数据模型。

1. 第一范式

数据模式规范的第一范式是，取消数据模式中的重复元组所得到的数据模式。第一范式具有以下的特点：

- 所有的属性都是原子化的；

- 它们不可能有相同的一组值；
- 它们不可能有任何的嵌套关系。

2. 第二范式

数据模式的第二范式是在第一范式的基础上，消除非关键列对关键列的部分依赖关系所得到的数据模式。第二范式需要保证所有非主键列完全依赖于主键列

3. 第三范式

通过分解消除第二范式中的传递依赖（对非主键列的依附），就得到数据模式的第三范式。第三范式具有以下特征：

- 所有的非主要属性都完全依赖于每一个键；
- 所有的主要属性都完全依赖于不属于它们的键；
- 没有属性完全依赖于任一非主属性集。

因此，在将数据模式从非规范到第三范式的转换过程中，需要采取以下3个步骤。

- 第一步：消除所有的重复元组，实现第一范式。
- 第二步：将实体的所有非主属性依赖于所有的主键列。
- 第三步：将所有非主键列直接依赖于主键列。

4. 数据仓库的反规范化处理

在数据仓库中对数据模型进行规范化处理后，发现这些规范化处理在数据仓库的实际应用中并不理想。因为经过规范化处理后的数据模型形成了一系列的小表，每个表的数据量较小。为完成对这些小表的处理需要应用程序对这些表进行动态的互联操作，这就需要在不同表中进行 I/O 操作。对于较少的、小容量表 I/O 操作也许不会产生较大的影响，但是对于数据量十分庞大的数据仓库，这种多表的连接操作在时间上是很难被用户接受的。提高 I/O 操作的最好方法就是使这些小表合并在一起，即进行数据的反规范化处理。

在数据仓库的应用中有一些基本数据，如果按照规范化处理原则应该存放在基本表中，而其他各种变动性数据则存放在各自的变动表中。这样对各种变动表的查询操作都要涉及基本表和变动表，也就是说至少要涉及两个以上的表操作。如果将基本表的数据作为冗余数据插入各种变动表中，在对数据仓库的操作中就可以减少表的连接操作。也就是说，利用数据模型的反规范化处理可以提高数据仓库的运行效率。由此可见，在数据仓库的模型构建中，为了提高数据仓库的运行效率，有时需要采用反规范化处理。

5.2.3 星型模型

ER 数据模型作为一种数据仓库的设计基础,在实际应用中存在很多缺点。如图 5.5 所示的简单 ER 数据模型中有 5 个相互关联的简单实体。从数据模型的设计角度来看,数据仓库设计所有实体之间的关系是对等的,仅仅从数据模型的角度来设计数据仓库会产生一种“平等”效应。实际上,由于种种原因,数据仓库的实体绝不会是相互对等的。一些实体,要求有它们自己的特别处理,根据在数据仓库中建立实体时将载入数据实体的数据量,考虑数据仓库中数据的一种结构设计。在实际工作中,代表供应商、客户、产品、发货的实体数据量只是一些说明订单的实体,而订单实体则是管理者所关心的分析对象。这样,在数据仓库的应用中将有大量的数据载入订单实体表,而其他实体表中的数据载入量则相对较少。因此,需要一种不同的数据模型设计处理方式,用来管理数据仓库中载入某个实体的大量数据的设计结构,这就是“星型模型”。

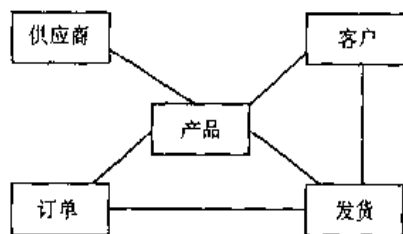


图 5.5 一个简单的 ER 数据模型

星型模型是最常用的数据仓库设计结构的实现模式。它使数据仓库形成一个集成系统,为最终用户提供报表服务,为用户提供分析服务对象。数据仓库的拓扑结构是多变的,从各种数据源(普通数据库系统和文件)中提取数据,加载到数据仓库中。这个数据仓库就可用于填充各种面向过程的数据集市。这些数据集市往往组成星型模式拓扑结构,以达到较高的检索效果。星型模式通过使用一个包含主题的事实表和多个包含事实的非正规化描述的维度表,支持各种决策查询。星型模型可以采用关系型数据库结构,模型的核心是事实表,围绕事实表的是维度表。通过事实表将各种不同的维度表连接起来,各个维度表都连接到中央事实表。维度表中的对象通过事实表与另一维度表中的对象相关。通过事实表将多个维度表进行关联,就能建立各个维度表间的对象之间的联系。每个维度表通过一个主键与事实表进行连接(参见图 5.6)。

事实表主要包含描述特定商业事件的数据。一般情况下,事实表中的数据不允许修改,新的数据只是简单地增加进事实表中。维度表主要包含存储在事实表中数据的特征数据。每个维度表利用维度关键字通过事实表中的外键约束于事实表中的某一行,实现与事实表的相关联。这种结构使得用户很容易地从分析维度表中的数据开始,获得维度

关键字，以便链接到中心的事实表中进行查询，这就可以减少在事实表中扫描的数据数量，而提高查询性能。

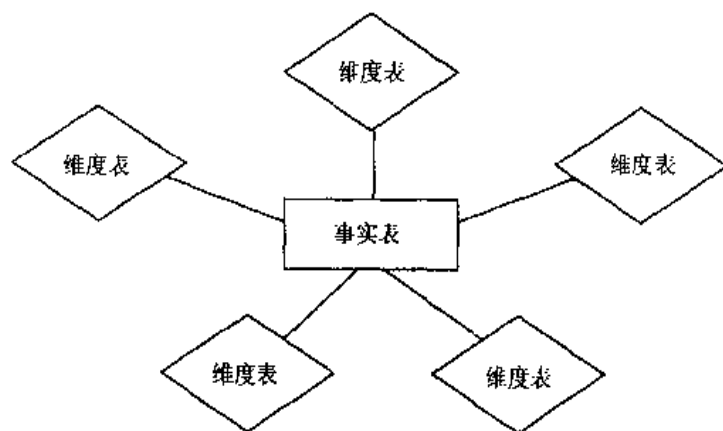


图 5.6 星型模型的结构示意图

5.2.4 雪花模型

雪花模型是对星型模型的扩展，每个维度都可向外连接到多个详细类别表。在这种模式中（参见图 5.7）。维度表除了具有星型模型中的维度表功能外，还连接上对事实表进行详细描述的详细类别表。详细类别表通过对事实表在有关维上的详细描述，达到了缩小事实表、提高查询效率的目的。

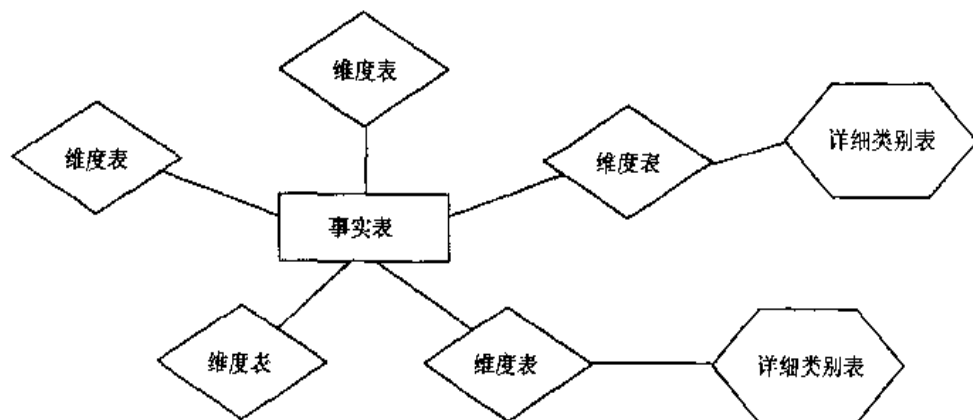


图 5.7 雪花模型结构示意图

雪花模型对星型模型的维度表进一步标准化，对星型模型中的维度表进行了规范化处理。雪花模型的维度表数据中存储了正规化的数据，这种结构通过把多个较小的标准化表（而不是星型模型中的大的非标准化表）联合起来改善查询性能。由于采取了标准化及维的较低的粒度，雪花模型提高了数据仓库应用的灵活性。

5.3 中间层逻辑模型

中间层数据模型亦可称为逻辑模型，它是对高层概念模型的细分，在高层模型中所标识的每个主题域或指标实体都需要与一个逻辑模型相对应。高层概念模型与中层逻辑模型的对应关系如图 5.8 所示。

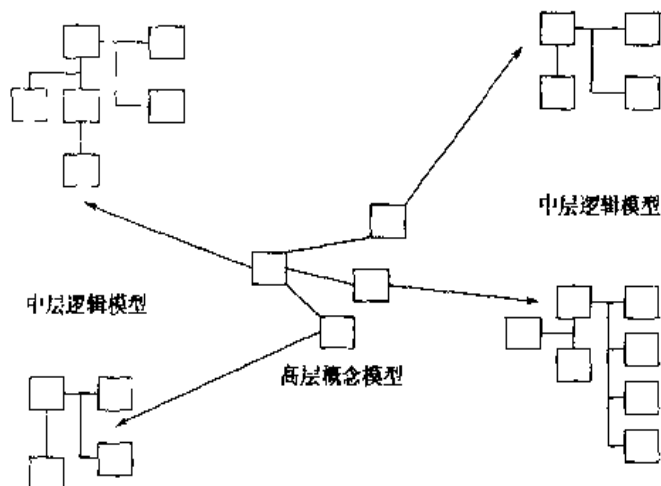


图 5.8 高层概念模型与中层逻辑模型对应关系

图 5.8 中高层概念模型中有四个实体或主题域，每个主题域都扩展成中层逻辑模型。在逻辑模型中有 4 个基本结构：基本数据组、二级数据组、连接数据组和类型数据组（参见图 5.9）。

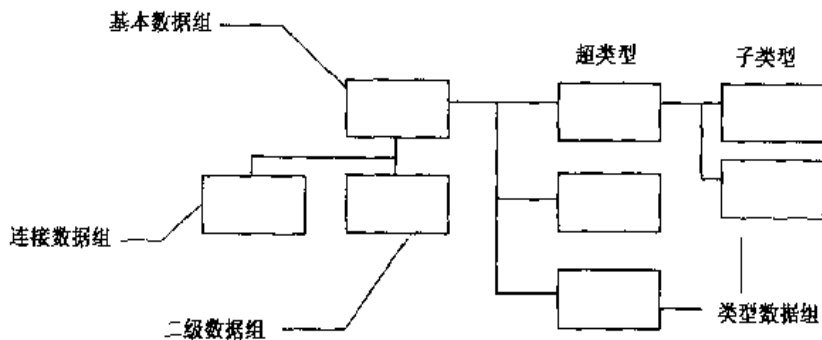


图 5.9 逻辑模型的基本结构

基本数据组中存在着惟一的主要主题域，它有在每个主要主题域只出现一次的属性。与所有的数据组一样，基本数据组包含属性和键码。

二级数据组有对每个主要主题域可以存在多次的属性。从初始数据组有一链接指向二次数据分组。有多少个可以出现多次的不同数据组，就含有多少个二级数据组。

连接数据组用于本组主要主题域与其他主要主题域之间的联系，体现了高层概念模

型中实体间的关系。它将数据从一个实体与另一个实体联系起来。一个概念层确定的关系导致了逻辑层的确认。一般情况下,连接数据组往往是一个主题的公共码主键。从而建立了两个主题域间的相互联系。

类型数据组指数据的类型。数据的“类型”由指向右边的不同数据组组成。主要有左边的超类型数据组和右边的子类型数据组。

除连接数据组以外的三种数据组划分标准,基于这些数据不同的稳定性。基本数据组的稳定性要大于二级数据组,而二级数组的稳定性又大于类型数据组。例如,在图 5.10 的金融企业客户主题逻辑模型中,其客户姓名、性别和开户时间等有关客户固定描述信息的数据项内容是基本不变的,所以它们可列入基本数据组。而客户的住址、文化程度、电话等项虽然也基本稳定,但存在改变的可能性,因而可以列入二级数据组;而客户的贷款情况、存款担保和信用卡消费记录等则是变动频繁的数据项,所以列入类型数据组。

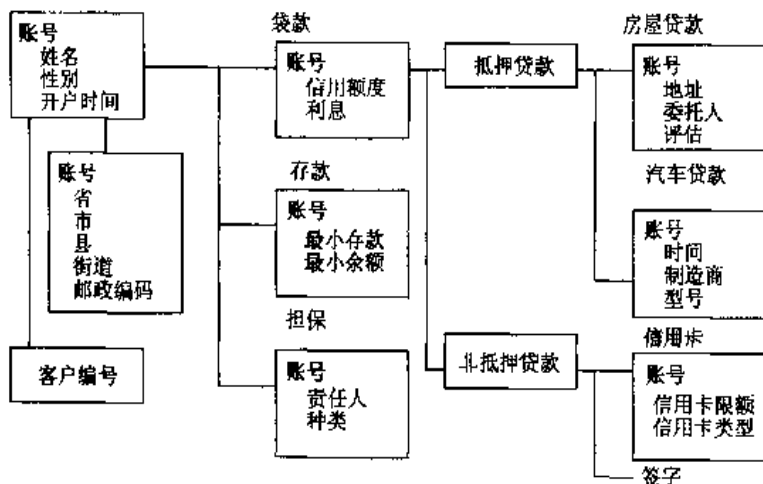


图 5.10 金融企业客户主题逻辑模型

通过中层逻辑模型的设计,可向用户提供一个比概念模型更详细的设计结果,使用户了解到数据仓库能够给他们提供一什么信息。逻辑模型也就成为数据仓库开发者与使用者相互之间进行数据仓库开发的交流与讨论的工具。在逻辑模型中已经具有各种数据的一些属性,使数据仓库的设计向数据仓库物理模型更加迈进了一步。

在中层逻辑模型设计中,数据仓库开发者关心的是数据仓库的结构和完整性,需要保证数据仓库的所有数据元素包含在数据模型中。在设计中对这些数据元素来自何处,如何获取不感兴趣,只关心这些数据元素是否满足用户的信息需求。

5.4 物理数据模型

物理数据模型是依据中间层的逻辑数据模型创建的。它通过确定模型的键码属性和

模型的物理特性, 扩展中间层数据模型而建立的。此时, 物理数据模型就由一系列表所构成, 其中最主要的是事实表模型和维表模型。在物理模型的设计中还要根据物理模型的性能, 对有关的表模型进行调整, 并且确定有关的索引设置。

5.4.1 事实表模型设计

物理模型中的事实表来源于逻辑模型, 例如, 根据图 5.10 的金融企业客户主题逻辑模型可以设计出以下事实表模型。

1. 客户事实表

客户基本情况表 (账号 Integer9, 姓名 Character12, 出生地 Character20, 开户时间 Date……)

客户变动情况表 (账号 Integer9, 省 Character20, 市 Character20, 县 Character20, 街道 Character20, 邮政编码 Character6……)

2. 客户贷款事实表

客户房屋贷款情况表 (账号 Integer9, 地址 Character50, 委托人 Character12, 评估 Memo……)

客户汽车贷款情况表 (账号 Integer9, 时间 Date, 制造商 Character40, 型号 Integer10, 颜色 Character8……)

3. 客户存款事实表

客户存款表 1 (账号 Integer9, 时间 Date, 最小存款数 Number7.2, 最小余额 Number7.2……)

客户存款表 2 (账号 Integer9, 时间 Date, 最小存款数 Number7.2, 最小余额 Number7.2……)

4. 客户担保事实表

客户担保表 1 (账号 Integer9, 时间 Date, 责任人 Character12, 种类 Character2, 担保金额 Number10.2……)

事实表是星型模型结构的核心。事实表中一般包含两部分, 一个是由主键和外键所组成的键部分, 另一个是用户希望在数据仓库中所了解的数值指标。这些指标是为每个派生出来的键而定义和计算的, 称为事实或指标。由于事实是一种度量, 所以事实表中的这种指标往往需具有数值化和可加性的特征。

事实表是数据仓库中的最大表，因为它包含大量的基本业务详细信息。在设计事实表时，一定要注意使事实表尽可能地小，因为过于庞大的事实表在表的处理、备份和恢复、用户的查询等方面需要较长的时间。在实际设计时，可以利用减少列的数量，降低每列的大小，把历史数据归档到单独的事实表中等多种方法降低事实表的大小。另外，在事实表中还要解决好数据的精度和粒度的问题。

5.4.2 维模型设计

维度表模型也需要根据逻辑模型设计，在设计过程中考虑维度表模型是用户分析数据的窗口。维度表应该含有商业项目的文字描述，维度的设计提供了维度属性的定义，这些属性是丰富的。一个对象的维度包含大量的属性。这些属性应具有这样一些特征：可用文字描述、离散值、有规定的限制、在分析过程中可以提供行标题。

设计维度表的主要目的是把参考事实表的数据放置在一个单独的表中。最常用的维度表数据应该直接参考事实表，而不是通过其他维度表间接参考事实表。这种方法可以最小化表的连接数量，提高系统的性能。例如根据图 5.10 的逻辑模型，可以设计出金融企业客户主题维度表模型。

客户主题维度表模型

时间维度表（年 Date，月 Date，[日 Date]）。

地点维度表（省 Character20，市 Character20，县 Character20，街道 Character20）。

贷款维度表（抵押贷款 Character20，非抵押贷款 Character20）。

在观察维度表中的维度对象时，其属性可以看做描述该项目的各种信息。例如，对于产品信息的维度，可以说它们被描述为具有特定的颜色。这个属性应当能用文字表示，比如“红色”。每个具有不同颜色的产品都应在该维度中有自己的记录，使之离散分布开来。而且维属性可以用做分析的标题，用于客户端用户进行选择查询的参考，例如按照产品的颜色维查询，或按照产品的类型维查询等。维属性在分析数据仓库中的数据时非常有用。从某种角度讲，维属性就是用户获取数据的窗口。

5.4.3 数据仓库物理数据模型的性能问题

数据仓库物理数据模型设计的一个主要问题，是确定和提高数据仓库系统的性能。在数据仓库的设计中，确定操作性能的第一步意味着决定数据的粒度和分割。由于数据粒度的划分问题较复杂，将在 5.6 节中详细介绍，数据分割则在 7.2 节中另行介绍。因为数据仓库的数据量很大，分析处理时涉及的数据范围较广，涉及大规模数据查询，这就要求提高数据仓库系统的 I/O。因此，物理数据模型设计的另一个主要内容是物理的 I/O 设

计问题。即如何更快地将数据从存储器调入计算机,或者将数据从计算机送到存储器。

数据在计算机和存储器之间的调入是按块进行的。因为在计算机中存储器和计算机间的传输速度比计算机运算速度慢很多。计算机内部的运算速度以毫微秒为计算级,而数据的传输速度是以毫秒为计算级。因此,物理的 I/O 是一个影响数据仓库性能的主要因素。数据仓库设计者的工作是要设计好数据的物理组织,以便在最短的时间内返回最多的数据记录。

在进行数据仓库的物理数据模型设计时,考虑数据仓库数据量大但是操作单一的特点,可以采取以下一些技术提高数据仓库性能技术。

1. 合并表

在数据仓库中,存在着一些例行的分析处理,它们要求的查询也是例行的,具有一定的固定性。某个例行的查询涉及固定几个表的数据项,需要首先对几个表进行连接操作。如果这几个表的记录分散存放在几个物理块中时,多个表的存取和连接操作的代价会很大。为了节省 I/O 开销,可以把这些表的记录混合存放在一起,就可减少表连接操作的代价。

2. 建立数据序列

在数据仓库环境中,经常按照某个固定的顺序访问并且处理一组数据记录,但这些数据记录最初可能分布在不同的物理块中。可将数据严格地按照处理顺序存放到一个或几个连续的物理块中,形成所谓的数据序列,就可以在同一次调页查询中处理更多的记录,将物理 I/O 降到最低。

3. 引入冗余

由于数据分析的处理数据范围是广泛的,通常涉及不同表的多个属性,一些表的某些属性可能在许多地方都要用到。如果这些属性的值不是经常更新的话,那么将这些属性复制到多个主题中,从而减少处理时被存取表的个数。这种方法就是为提高数据仓库的处理性能而采取的数据模型非规范化处理。

这种引入冗余的方法与合并表方法是不同的。合并表是将两个或多个相关表的相关记录物理上存放在一起,但逻辑上仍是两个或多个表,即没有改变各表的关系模式;而且合并表只是对表记录的存取策略的改进,并没有冗余的数据。引入冗余的方法则是对表的关系模式的改变。引入冗余后,需要维护数据各个拷贝间的一致性,在这些数据上的修改操作将变得极为复杂。因而在操作型数据库中,引入冗余的方法并不可取:一是它破坏了关系模式的规范化;二是因为操作型数据环境中的数据是联机更新的,引入冗余势必增加修改操作的代价。在数据仓库中数据是稳定的,适当引入冗余也就成了提高

系统性能的一种有效方法。

4. 表的物理分割

表的物理分割主要依据是数据的存取频率和数据的稳定性。每个主题中的各个属性的存取频率是不同的。可将一张表按各属性被存取的频率分成两个或多个表，将具有相似访问频率的数据组织在一起，将使每次数据的访问更有效。表的物理分割还可依照表中的属性稳定性程度不同来划分。对同一表的属性进行稳定性分析，将更新频繁的属性划为一个表，其他属性则划分为一个表。这种划分可使数据仓库的数据加载，集中在数据更新频繁的表上，使数据的加载与概况处理的效率更高。

5. 生成导出数据

如果事先在原始数据的基础上进行总结或计算，生成导出数据，就可以在应用中直接使用这些导出数据，既减少了 I/O 的次数，又免去了计算或汇总的步骤。它的另一个好处是在更高级别上建立公用数据源，避免不同用户重复计算可能产生的偏差。

6. 建立广义索引

数据仓库的数据量巨大，所以要依靠各种各样的索引技术来提高涉及大数据量的查询的速度。在从操作型数据环境抽取数据且向数据仓库中装载的同时，就可以根据用户的需要建立许多“广义索引”。每次向数据仓库装载时，就重新生成这些“广义索引”的内容。这样就不需要为了建立“广义索引”而重新去扫描仓库。对于一些经常性的查询，建立这种“广义索引”来代替对关系表的查询要方便得多。

在完成数据仓库的物理数据模型以后，就可根据物理数据模型的特性，选择某个数据仓库创建工具，完成数据仓库的物理实现。

5.5 元数据模型

数据仓库中的元数据是关于数据的数据，其基本含义和作用已经在第 1 章中有所描述。鉴于元数据在数据仓库中的重要性，无论怎样强调都不为过。正是有了元数据，才使得数据仓库的最终用户可以随心所欲地使用数据仓库，对数据仓库进行各种模式的探讨。因此，这里就元数据模型的设计、实施、应用及管理做进一步介绍。

5.5.1 元数据的类型与组成

元数据作为数据的数据，可对数据仓库中的各种数据进行详细的描述与说明；说明

每个数据的上下文关系,使每个数据具有符合现实的真实含义,使最终用户可以了解这些数据之间的关系。

根据元数据在数据仓库中所承担的任务,可将元数据分成静态元数据和动态元数据两大类(参见表 5-2)。静态元数据主要与数据结构有关,其中包括名称、描述、格式、数据类型、关系、域和业务规则等类。动态元数据主要与数据的状态与使用方法有关,其中包括数据质量、统计信息、状态和处理等类。

静态元数据中的名称用于为系统提供识别、区分数据的符号,例如, Customer_ID, Employee_ID, Customer_Name 等。

表 5-2 元数据分类

元 数 据										
静态元数据							动态元数据			
名称类	描述类	格式类	数据类型类	关系类	域类	业务规则类	数据质量类	统计信息类	状态类	处理类

元数据的描述主要对数据仓库中的各种数据元素进行说明,例如“销售金额”数据的描述是:向客户所销售产品的总金额。

元数据的格式用于提供数据仓库中数据的表达规则,例如“销售总量”数据的格式采用整数部分从小数点开始向左三位一组,每组用逗号分开,小数部分保留两位。

元数据中的数据类型用于说明数据仓库中的数据所持有的类型,这些类型可能有图像、布尔、整数、实数等。例如,“销售金额”的数据类型为实数类型。

元数据的关系用于说明数据仓库中各种数据对象之间的关系。例如,“客户”数据与“商品”数据之间存在购买关系。

元数据的域用于说明数据仓库中数据的有效值范围,例如,“Customer_ID”的域为以字母 A 或 B 开头后接七位 0 至 9 的字符。

元数据的业务规则用于说明数据仓库中数据在业务处理中所要遵守的规则,例如,“Customer_ID”表示客户的编号,开头字母为 A 是集体客户,开头字母为 B 是个人客户。

元数据的数据质量用于描述数据仓库中数据的精确度、完整性、一致性和有效性。例如,在数据仓库中的数据提取日志记录了数据抽取过程中的数据抽取过程,可以作为验证集成在数据仓库中的数据质量。

元数据的统计信息统计数据访问的用户、访问用户、访问时间与访问次数,这些统计信息对于数据仓库性能的提高具有较高的参考价值。

元数据状态用于跟踪数据仓库的运行状况,例如数据最近一次的备份时间,备份所需要的时间,出现的错误等状况。这些系统运行中的状况有助于数据仓库管理人员对数据仓库性能的了解。

元数据的处理描述数据仓库系统的使用方法和管理的特性,例如数据的使用方法、

概况数据的概况公式等。

5.5.2 元数据在数据仓库中的作用

从元数据的类型与作用来看,元数据实际上是要解决何人在何时、何地为了什么原因、怎样使用数据仓库的问题。

DSS 数据分析员为能有效地使用数据仓库环境,往往需要元数据的帮助。尤其是在 DSS 数据分析员进行信息分析处理时,他们首先需要去查看元数据。元数据还涉及数据从操作型环境到数据仓库环境中的映射。当数据从操作型环境传入数据仓库环境时,数据需要经历一系列重大的转变,包含数据的转化、过滤、汇总、结构改变等过程。数据仓库的元数据能够及时跟踪这些转变。当 DSS 数据分析员需要将数据从数据仓库环境追溯到操作型环境时,就要利用元数据来追踪这种转变。另外,由于数据仓库中的数据会存在很长一段时间,其间数据仓库往往可能改变数据的结构。随着时间的流逝来跟踪数据结构的变化,是元数据另一个常见的使用功能。

元数据描述数据的结构、内容、码、索引等项内容。在传统的数据库中,元数据是对数据库中各个对象的描述,数据库中的数据字典就是一种元数据。在关系数据库中,这种描述就是对数据库、表、列、观点和其他对象的定义。但在数据仓库中,元数据定义数据仓库中的许多对象——表、列、查询、商业规则或数据仓库内部的数据转移。元数据是数据仓库的重要构件,是数据仓库的指示图(roadmap),指出数据仓库中的各种信息的位置和含义。理解元数据对于了解数据仓库各构件的正确功能是非常重要的。数据抽取程序必须了解数据源的元数据和目标数据仓库的元数据。用户为能正确有效地检索数据,也须了解数据仓库的元数据。因此,设计一个描述能力强,内容完善的元数据,对数据仓库进行有效的开发、管理具有决定性的重要意义。

1. 数据仓库元数据的重要性

(1) 为数据仓库服务与 DSS 分析员及高层决策人员服务提供便利

元数据为 DSS 分析员及高层决策人员提供他们使用数据进行分析的基础。例如,数据仓库元数据的广义索引中存有每次数据装载时产生的有关决策的数据,在做决策时,可以先去查找这部分数据,再决定是否进行进一步的搜索。

(2) 解决操作型环境 and 数据仓库的复杂关系

操作型环境 and 数据仓库之间有着复杂的、多方面的区别。因此,从操作型环境到数据仓库的转换也是复杂的、多方面的。元数据应包含对这种转换的描述。元数据要将这种转换清晰地表示出来,把从哪些数据源用怎样的转换逻辑转换成数据仓库中的哪些日的数据等内容描述出来。这样,当从数据仓库向数据库回溯时,便能根据数据变换的历

史,找到原始依据。数据仓库的元数据还要将这种转换管理起来,既保证这种转换是正确的、适当的或合理的,又要使其是可变的、灵活的。事实上,因为用户的需求是不确定的,只有保证元数据的灵活性、可变性,才能真正保证其合理性和正确性。

(3) 数据仓库中数据的管理

除了描述和管理从数据库到数据仓库的转换外,数据仓库当然还要管理好数据仓库中的数据。一方面,数据仓库中的数据量很大,对数据所进行的一些处理,例如划分不同的粒度层次、进行分割策略的选择、建立各种各样的索引等等,都需要在元数据中进行描述和管理;另一方面,数据仓库包含着较长时期内的数据,不同时期不同需求的数据从“形式”到“内容”都可能不同。此外,决策需求的不断变化和增加,需要不断地完善主题或增加主题,也就要不断地修改元数据或增加新的元数据内容。

2. 元数据在数据仓库开发期间的使用

数据仓库的开发过程是一个构造工程的过程,它必须提供清晰的文档。该过程产生的元数据主要用于数据仓库的应用管理目的,例如必须描述数据仓库目录表的每个运作的模式,还须捕获用于数据的转化、净化、转移、概括和聚集的商业规则与处理规则。

数据仓库的元数据设计需要改变传统数据库设计的观念,数据仓库重点是向分析者提供大量的数据关系,而这些关系一般含有大量的冗余。由于数据仓库并不经常更新数据,所以在数据冗余方面的开销远不如在存储方面的开销。如果把相关的部分存储为表,一般可以节省处理的开销。分析者的观点既适用于减少数据冗余的情况,也适用于其他理想的情况。

系统分析者的另一个手段是突出操作系统的当前元数据。大多数业务处理系统的应用程序只操作数据库的当前结构和数据,旧的数据和数据库结构一起归档。在数据仓库内部,运作数据库版本的结构是非常重要的,必须用元数据来抽取历史数据。数据仓库元数据的版本是一个重要的因素。

元数据在数据仓库设计中的另一个重要作用是在抽取、求精和重构工程过程中,时刻保持从资源到数据仓库之间的映射关系。这些关系可用于以下3个目的。

(1) 确认数据质量

映射关系包含有关数据在存储到数据仓库之前所经历的各种变化的信息。如果希望在对数据仓库进行分析和解释的基础上所制定的决策是准确的,那么这种“审查足迹”的方法是非常重要的。

(2) 同步化和刷新

随着新的业务处理信息的产生以及对仓库的更新,新的数据必须经过转换,使其等价于以前加载的仓库数据。因此,在元数据库中保持映射关系和转换算法对于重复刷新数据的转换过程是必不可少的。

(3) 映射

映射在反映最终用户所关心的商业规则和数据之间建立一种关系。如果没有那种映射关系,用户所得到的只是无益于决策的孤立数据块。

3. 元数据在数据源抽取中的作用

数据源块的元数据用于数据库的定义,以及向数据仓库及其定义提供从办公系统和外部来源中抽取的数据条目。元数据对多个来源的数据集成发挥着关键作用。

(1) 资源领域的确定

原始数据包含在各种技术中,从基于文件的系统到关系数据库系统。信息的域名一般是隐含的,也是需要了解的。利用元数据可以确定将数据源的哪些资源域加载到数据仓库中去,而传统的数据字典则是无法做到的。

(2) 跟踪历史数据结构变化的过程

数据仓库的数据存储需要一致的数据结构,但是元数据中有各种各样的数据结构,要将这些原始格式转化为数据仓库目标格式。但是,数据源所在的应用系统可能发生数据库结构的变动、合并或重组。具体的变化可能包括属性长度的变化、数据类型的变化、编码方案的变化和关键字段的变化等。元数据需要跟踪这些变化,才能将各种结构的数据源正确转换到数据仓库中。

(3) 属性到属性的映射

多个资源中的相似字段必须映射到一起,以便能把这些字段中的数据加载到数据仓库内的同一目标字段中,元数据的属性信息就需指出哪些字段可以映射到一起。

(4) 属性转换

一般情况下,从多个数据字段新形成的信息字段存在不同的格式(长度和数据类型),必须用元数据来指出每个新字段的格式。通过将数据仓库中通用的新字段格式与目标字段格式进行对比,可以定义转换过程。这些转换过程将数据修改为兼容的格式,以便加载到数据仓库中。典型的转换有截取、补充和取舍。

4. 元数据在数据求精与重构工程上的作用

数据求精与重构工程负责净化资源中的数据、增加资源戳和时间戳、将数据转换为符合数据仓库的数据格式、预算概括和衍生数据的值。

(1) 集成与分割

分割过程将单一的数据块分成数据仓库中的两个或多个数据块。当业务系统将数据存储到一张表里,而按性能需要最好将该表分离成数据仓库中独立的表时,就要进行分割操作。一般要求数据对象所含的概念是独立的。要用新数据的元数据为目标数据仓库推导出两个或多个元数据块。因此需要制定一个方案,将新数据定位到多个目标中。分

割的另一个原因是很容易将一组数据分成许多各自独立的部门, 可以进行不同形式的分析。例如, 数据分割可以按日期、商业生产线、地理位置或者部门单位等方式进行。此时, 元数据需要指出与给定数据有关的属性。

(2) 概括与聚集

概括最简单的形式是形成累积的结构, 通过把各种属性累加到一起就可做到这一点。例如, 客户每天的订货就是特定客户在特定的一天中所有订货的总和。通过查找有关特定客户在特定的一天中的所有订货情况就可获得所需的信息, 然后将总数累计到一个新的字段中。因此, 概括过程就是向需要容纳概括总数的数据中增加新的数值。概括过程需要指明的问题有: 将哪些字段增加到总数中, 如何形成总数, 总数应存储到什么地方等。为了能够重复进行概括过程, 以上问题必须作为元数据存储起来, 今后的数据概括就可以在元数据指导下进行。

(3) 预算与推导

预算与推导是应用于数据仓库的、无需用户干预或要求的计算结果。这些结果经过计算、存储, 可以用做数据仓库中的数据字段。预算与推导创建附加的数据字段, 用于从已有数据仓库字段预算和推导新字段的算法也须作为数据仓库内部的元数据进行存储和管理。

(4) 转换与再映像

向数据仓库提供数据的业务数据源通常组织成标准化的或基本标准化的关系表, 而分析所需的模式一般是星型模式, 其中事实表连接到许多维表中。转换与再映像就是把数据源信息转化为适合于数据仓库事实表的行的过程, 再映像过程包括将许多表组成事实表的行。在元数据中需要保存这些转换与再映像的方案。

5.5.3 元数据的收集

元数据几乎遍布在数据仓库中的任何一个地方和数据仓库的环境中。例如, 在收集业务数据的业务处理系统中有元数据, 存储业务数据的数据库有元数据, 抽取数据源的中间件有元数据, 数据仓库数据库有元数据, 数据仓库设计系统有元数据, 数据仓库管理系统有元数据, 数据仓库的用户工具也有元数据。

面对如此众多的元数据来源, 在元数据的收集过程中应该尽量采用自动收集方式进行。否则, 单纯依靠人工进行元数据的采集, 可能遗漏一些重要的元数据, 或采集一些错误的元数据。这将对数据仓库的开发与应用产生严重的后果。

元数据的收集一般不会给开发人员带来额外的工作量, 相反将有益于数据仓库的开发。例如, 对于那些描述现有业务处理系统中数据库结构的元数据, 数据仓库开发人员在分析数据仓库数据源时, 已经了解了这些数据库的结构, 他们所要做的只是将其

存入元数据库中而已

1. 数据源的元数据

元数据的主要来源之一也就是数据仓库数据的来源地, 包含业务处理系统的数据库、可以获得的外部数据和手工处理的数据。例如, 由销售部门应用的销售业务处理系统中的商品销售报表、从市场调查咨询公司购买的市场调查数据、在销售员记事簿上的潜在客户联系数据。对于存储在系统中数据的物理结构是一种比较容易收集的元数据, 这些数据的物理结构、含义以及类型可以编制成文档, 在可能的情况下, 尽可能使用扫描程序对这些数据的物理结构进行扫描分析。如果数据源有库结构表, 那元数据的收集工作就更简单了。如果无法进行自动扫描处理获取元数据, 就只能采用手工方式进行处理, 好在使用手工获取元数据的数据量一般都比较小, 容易分析编写元数据文档。

2. 数据模型的元数据

数据仓库中元数据的第二个重要来源是有关数据模型的信息, 从数据模型中可以了解关于组织业务的实体、关系和规则。随着数据仓库的设计开发与应用, 数据模型也在不断变化, 这就需要对初始的数据模型进行修改, 使其正确反映数据模型与数据仓库之间的关系。这样, 才能了解一些特定的实体是如何在数据仓库中实施的。因此, 在设计数据模型以后, 必须将其存入元数据库中。在收集企业数据模型和元数据以后, 必须要使两者之间一一对应起来, 为未来的数据仓库变动影响分析与用户使用数据仓库时的分析奠定必要的基础。在实现这种一一对应的指定联系后, 还需要将元数据定义、业务规则、有效值和使用指南都从企业数据模型中移入元数据库。这些元数据有利于用户对数据仓库的访问, 且能够对所获取的信息作出合理的解释。

在从数据模型中收集元数据时, 应该尽可能使用数据模型设计的 CASE 工具来实施。对于比较重要的数据模型与元数据的对应关系的确认最好采用手工方式完成, 以保证两者之间的联系百分之百正确。

3. 数据源与数据仓库映射的元数据

数据源与数据仓库之间的映射关系, 是十分重要的, 它决定数据仓库的数据在从数据源中抽取、转换、加载到数据仓库过程中发生了哪些变化。将数据源加载到数据仓库之时的操作如果是数据仓库开发人员手工完成的, 就必须利用电子表格或数据库方式将这些映射关系进行明确的定义, 然后合并到元数据库中。如果数据源到数据仓库的数据抽取、转换、加载是由专门的数据仓库开发工具完成的, 也需要将这种映射关系并入元数据库中。且要提供访问这些映射规则的方式与工具。

4. 数据仓库应用的元数据

收集用户使用数据仓库的元数据是数据仓库元数据模型构造中最后的、也是最困难的、最重要的内容。如果通过元数据能够了解哪些用户在使用数据仓库中的使用频率,那就能够使数据仓库管理者为那些高频率的使用对象建立相应的数据集市或增加概括数据,或将那些很少有人使用的概括、聚集数据释放,收回这些数据所占据的磁盘空间。

收集这些元数据,必须依靠某种系统监控工具截取并且解释每个查询,然后将数据传送到元数据库中进行分析跟踪。如此还要能够确认新查询操作,将新查询操作及其用于解决决策问题的描述编入查询操作目录。这样可为数据仓库的所有用户提供一种数据仓库应用的蓝本,使一些不熟悉数据仓库应用的用户,可以通过对数据仓库查询操作目录的阅读,了解其他用户在解决决策问题时,是如何使用数据仓库的,对自己的数据仓库应用产生某种启迪。

数据仓库的应用元数据收集往往依赖手工,尤其是对新查询的应用描述,必须在数据仓库管理人员进行多次确认以后,才能编写进元数据库。数据仓库应用的元数据收集虽然花费精力较大,但是收益更大。

5.5.4 元数据的存储、管理与维护

1. 元数据的存储

数据仓库开发阶段产生的元数据要能够得到有效的应用,必须进行适当的组织和存储。元数据组织与存储的方法一般有以下两种。

(1) 使用商业或数据仓库信息目录

信息目录可存储和管理元数据,用于数据仓库应用程序。数据仓库的所有内部程序都可访问该目录,如抽取程序、求精与重构工程程序和转换程序等。最终用户还可用该目录进行元数据的浏览、导航、数据抽取和查询。

(2) 使用元数据库/数据字典

元数据库或数据字典是一种一般意义上的分类方法,通常用于存储、分类和管理元数据。元数据库可用一种“信息模型”的分类方法进行管理,“信息模型”中含有各种类型的元数据及其相互关系。元数据库是一种非常灵活的、一般意义上的元数据管理方法,而不仅仅简单地管理数据仓库元数据。例如,数据源的定义也可以在元数据库或数据字典内部进行持久的管理。

2. 元数据的管理

实现对元数据的有效管理,一般需要如下4个管理功能。

(1) 将元数据组织为易于理解的分类方案

将元数据组织为易于理解的分类方案主要依靠元数据库或数据字典的信息。这种分类方案允许元数据管理人员定义分类。元数据库或数据字典的信息模型还应具有可扩充性——能够随着技术的发展,定义元数据的新类,且将元数据与旧数据联系在一起。

(2) 效果分析和查找能力的有效范围

效果分析和查找能力既能检索元数据的信息,也能弄清元数据之间的关系。

(3) 将设计和开发元数据与运作元数据分隔成各自独立的功能

这些功能一般用于分隔逻辑分析模型和物理数据库模型。该功能通常称为软件开发生存期分割。

(4) 反映修改历史的元数据版本信息

版本信息能够反映元数据版本变化日期以及修改操作人。版本信息和日期对于数据仓库是非常重要的,因为数据仓库的历史数据必须有明确的元数据描述,并且指出其组织方式和生成时间。

3. 元数据的维护

在元数据存储进系统后,就需要对元数据经常进行维护,才能保证元数据的可用性。元数据的维护方式取决于元数据产生之时的收集方式,变化频率以及元数据量。

反映数据源和数据仓库结构的物理元数据维护可以采用自动维护方式。如果有库结构表,那自动维护方式就更容易实现;即使是手工维护,也是容易实现的。因为库结构的变动需要通过多次分析、讨论,才能决定,因此,就库结构变化的元数据收集工作量是很小的。由于库结构变化情况较少,只需要对物理数据结构的元数据收集变化部分即可。

至于业务规则和数据模型的元数据维护,则需要依靠手工完成。即使 CASE 工具可反映模型的变化,一般也采用手工方式维护。因为模型的变化往往需要经过多次讨论和评估,而且模型元数据量一般较少,可以采用全部刷新方式进行维护,不必费时寻找变化的模型。

数据源与数据仓库的映射维护则可自动进行,因为这种映射工作是由数据仓库工具完成的,而这些工具与元数据库之间可以建立接口。如果两者之间不能建立接口,那就需要进行手工维护。

对数据仓库使用元数据的维护则需要定期进行追加,而不是进行刷新。在用户产生一个新的查询使用时,需要生成或修改对查询的描述。这些元数据的维护必须手工方式进行,还要定期对使用元数据进行维护评审,找出那些没有应用描述的新的数据仓库查询操作。在维护过程中一定要注意与用户的配合,以获取对查询操作的最正确描述。

5.5.5 元数据的用户与使用方法

数据仓库的成功很大程度上取决于元数据的恰当应用，这就需要将元数据以合适的方式提供给元数据用户。元数据的用户主要有数据仓库开发人员、数据仓库维护人员和数据仓库用户三大类。

1. 元数据的数据仓库开发用户

数据仓库开发人员使用的元数据主要包括数据源的物理结构、企业数据模型和数据仓库数据模型。在数据仓库开发工作中需要对数据源元数据进行分析，根据分析结果在数据源和数据仓库之间建立映射。首先通过查询名称中包含业务术语的各种数据，利用这些数据元去识别数据仓库的数据源；在确认候选数据源后，利用企业数据模型的元数据去确定是否需要将其映射到数据仓库中。若有需要，可以通过对数据源的物理结构与数据仓库的物理模型进行对比，生成从数据源到数据仓库的映射，当然，这种映射也是一种元数据。数据仓库开发人员所关心的是在数据仓库的开发中是否采用了准确的、完整的元数据。在对元数据访问过程中，往往希望对元数据库进行直接访问。

2. 元数据的数据仓库维护用户

在数据仓库开发好后，数据仓库维护人员要对数据仓库进行维护，元数据在数据仓库的维护工作中可以发挥重要的作用。

维护人员用元数据能够了解数据源的变化，数据仓库的变化对数据仓库的性能，应用等方面的影响。即由于业务处理的需要对某个业务数据源进行调整，或由于数据仓库中某个表中的某些列在追加数据时会发生变化，而表的另一些列在数据追加时却很少追加新的数据。为节省磁盘存储空间，希望将表分开存储。此时，数据仓库维护人员需要利用元数据研究这些变化所产生的影响：对数据抽取、清理、加载等程序所带来的影响；数据仓库中哪些表的结构需要改变；有哪些数据集市与概况数据的结构会发生变化；已经保存起来的用户查询语句是否受到影响，是否要修改；对那些受到影响的用户要否重新提供培训；已经变化的数据可能影响哪些决策问题的分析。这些变化分析，如果没有元数据的支持是难以完成的。

数据仓库维护人员还可利用元数据保持数据仓库的完整性和正确性。例如，用户在数据仓库的应用过程中可能发现，利用数据仓库数据与利用业务处理数据所得到的决策结论有时是自相矛盾的。此时，数据仓库维护人员必须利用元数据就此数据在业务处理系统中的原状、在从业务处理系统中向数据仓库转移的过程中所发生的变化做出合理的解释，以确立用户对数据仓库的使用信心。

数据仓库维护人员对元数据的使用涉及所有的元数据，并且要求能够直接对元数据进行访问。

3. 元数据的数据仓库用户

元数据对于数据仓库用户的重要性更加突出。如果没有元数据，没有合适的、容易访问的元数据，用户不可能使用数据仓库，即使用了数据仓库也不可能对应用结果做出正确的解释。数据仓库用户对元数据的访问范围远小于元数据的数据仓库开发用户和维护用户，但是对元数据访问的要求却要高于其他类型用户。

数据仓库用户在使用元数据时，主要希望通过元数据了解数据仓库中有什么数据，这些数据是从什么地方来的。具体地说，他们希望了解的是按照某个主题查看数据仓库内容，且希望对所看到的数据就其完整性、业务含义、有效值范围和使用规则进行说明。例如，用户可能希望从销售订单处理系统、销售合同完成系统、销售账务处理系统中了解每月的商品销售情况。

数据仓库用户使用元数据的第二个主要方面，是希望利用已经存在的查询信息。例如，用户选定某个主题后，有关该主题的表和对这些表可以进行的查询以及对这些查询的描述都能列示出来，用户可以选择可用的查询或对某个查询稍加修改以后，就能用于数据仓库的查询操作，以减少用户的查询编程工作。

数据仓库用户在使用元数据时，应该能够以一种易于理解与访问的方式进行。为此，可以为数据仓库的用户提供一本完整的元数据使用手册，以方便用户的使用。

4. 元数据的使用方法

目前元数据的使用方法主要有这样几种：元数据与分析数据同时各自显示，元数据作为分析数据帮助，元数据的直接查询，元数据与分析数据的联动。

元数据与分析数据同时各自显示是指在一台计算机上分别用两种工具显示元数据和分析数据。这样，用户可以通过在一个工具中浏览元数据，在另一个工具中编写查询分析数据的程序，或利用元数据帮助理解查询工具中所显示的分析数据。

用户在元数据作为分析数据帮助这种元数据使用方式下，可以利用系统的帮理解所查询的分析数据。系统要将元数据从元数据库中填入系统的帮助文本，用户如果需要查看最新的元数据，需要对帮助系统进行刷新。

元数据的直接查询工具可以直接地、动态地访问元数据，能为用户提供最新的帮助系统。

元数据与分析数据实现互动以后，用户在元数据浏览器中浏览元数据时，可以将所选定的表或查询自动地调入查询工具。反之，用户在查询工具中进行查询分析时，可在元数据工具中查看相应的元数据解释。这是一种比较好的元数据使用方法。

5.5.6 元数据管理模型

由于元数据在数据仓库的开发中具有极其重要的作用,越来越多的数据仓库研究人员与数据仓库开发商开始重视元数据的管理,纷纷提出各种元数据管理模型。其中,Zachman 在数据仓库的结构分析中提出了元模型。Ralph Kimball 则提出将数据仓库与计算机的总线结构相比较,利用一种总线结构,将其他需要元数据或产生元数据的设施都连接在这一总线上,这样就可以实现数据的内部移动。Peteer Keen 提出的元模型是一种立方体结构,模型需要业务技术平台有接触、范围、响应为代表的三个立方体结构。

在数据仓库开发商中,微软公司提出开放信息模型(OIM, Open Information Model)。在微软公司所发布的数据仓库中,OIM 成为存储和管理对象定义模型的现行标准。从 1998 年 12 月开始,微软公司开始将元模型转移到元数据联盟(META Data Coalition)上,且将 CA/Platinum 作为数据仓库从 NT 环境到 UNIX 环境的接口。与此同时,Oracle 与 IBM 公司开始在 OLAP 委员会的元数据 API(MDAPI 2.0)基础上寻找元数据模型的解决方案。

在讨论元数据模型中,必须提到元数据交换规则(MDIS)。这是一个包括微软在内的有上百个成员的元数据联盟所提出的元数据交换规则,规则涉及数据库、文件、关系、用户自定义、专用元数据等不同对象类型。它提供了一个公用的 API 以支持用批处理方式加载元数据,最新的规则标准可以从 www.mdiss.com 中找到。

5.6 数据仓库的粒度模型

数据仓库开发者在数据仓库开发过程中,还要解决的主要问题之一是构造数据仓库的粒度模型。所谓粒度是指数据仓库中数据单元的详细程度和级别。数据越详细,粒度就越小,级别也就越低;数据综合度越高,粒度就越大,级别也就越高(参见表 5-3)。

表 5-3 粒度和数据细节之间的关系

数据详细程度	数据综合程度	数据粒度
低(如事务)	低	非常高
高(如汇总)	高	中等到低

粒度可定义成数据仓库中数据细节的最高层次,如事务层次。这种数据层次是高度细节化的,能使用户按所需的任何层次进行汇总。在传统的业务处理环境中,对数据的处理和操作都是在详细数据级别上的,即最低级的粒度。在数据仓库环境中用户的目的在于分析处理。根据粒度的划分标准可将数据划分为详细数据、轻度总结、高度总结三级或更多级粒度。不同粒度级别的数据用于不同类型的分析处理。粒度的具体划分将直

接影响数据仓库中的数据量以及查询质量。在数据仓库开始分析时,需要确定合理的数据粒度,建立合适的数据粒度模型,指导数据仓库设计和其他问题的解决。如果数据粒度定义不当,将会影响数据仓库的使用效果,使数据仓库达不到设计数据仓库的目的。

5.6.1 数据粒度的划分

适当划分粒度的第一步,是估算数据仓库中将来使用的数据行数 and 所需的直接存取存储设备数(DASD)。

划分数据粒度,先要估算数据仓库中需要建立的表数目,估算每个表的大致行数,通常需要估计行数的上、下限。由于数据仓库的数据存取是通过存取索引来实现的,而索引是对应表中的行来组织的,即在某个索引中每行总有一个索引项。索引的大小只与表的总行数有关,而不与表数据量有关。所以,粒度的划分是由总的行数而不是总的数据量来决定的。

在估算数据仓库所需要的存储空间时,可对每个表估算其一年所需要的存储空间,然后估算其最长的保留年数所需要的存储空间,假设每个表要在数据仓库中保留五年。那么,需要计算出每个表在数据仓库中保留五年所需要的存储空间总和,就是数据仓库所需要的全部存储空间。

每个表的存储空间,应该是每一个表的数据存储空间和索引存储空间之和。精确计算表的每年实际存储空间往往是很难的,只能给出表的最大估算空间和最小估算空间。为此需要估算每个表每年需要最多的行数和最少的行数,然后,估算出每行占用空间的最大字节数和最小字节数。至于每个表的索引存储空间,则只要估算出键码的占用字节数与索引的行数,便可计算出来。这样,每个表每年的存储空间就可以用表的存储空间与相应的索引空间之和表示。

在计算出数据仓库所需要占用的存储空间以后,需要根据所需要的存储空间大小确定是否划分粒度?如果需要划分,又应该怎样划分?

一般情况下,如果表的数据行数在第一年就达到了100 000行左右,数据仓库只有单一的粒度(即只有细节数据)就不太合适了,应该考虑粒度的划分,可以增加一个综合级别。如果数据行超过了1 000 000行,那么就要考虑采用多重数据粒度。数据行数在五年来如果预计达到1 000 000行,那么也就不能仅有细节级的数据,必须考虑选择粒度的划分。数据仓库表中数据的总行数和相应的数据粒度划分方法可以参考表5-4。应该注意,一年期数据量与数据粒度划分策略与五年期的划分策略是有差别的。一般情况下,一年期数据量所对应的粒度划分策略要比五年期的严格。这是基于计算机硬件性能价格比的快速提高与数据处理软件功能的日趋强大和用户水平逐步提高的现状,所做的一种乐观估计。

表 5-4 数据仓库的存储空间与数据粒度划分策略对照表

一年数据		五年数据	
数据量(行数)	粒度划分策略	数据量(行数)	粒度划分策略
10 000 000	双重粒度并且仔细设计	20 000 000	双重粒度并且仔细设计
1 000 000	双重粒度	10 000 000	双重粒度
100 000	仔细设计	1 000 000	仔细设计
10 000	不考虑	100 000	不考虑

5.6.2 确定粒度的级别

在数据仓库中确定粒度时, 需要考虑这样一些因素: 要接受的分析类型、可接受的数据最低粒度、能够存储的数据量。

计划在数据仓库中进行的分析类型将直接影响数据仓库的粒度划分。将粒度的层次定义越高, 就越不能在该仓库中进行更细致的操作。例如, 将粒度的层次定义为月份时, 就不可能利用数据仓库进行按日汇总的信息分析。

数据仓库通常在同一模式中使用多重粒度。数据仓库中, 可以有今年创建的数据粒度和以前创建的数据粒度。这是以数据仓库中所需的最低粒度级别为基础设置的。例如, 可用低粒度数据保存近期的财务数据和汇总数据, 对时间较远的财务数据只保留粒度较大的汇总数据。这样既可以对财务近况进行细节分析, 又可以利用汇总数据对财务趋势进行分析, 这里的数据粒度划分策略就需要采用双重数据粒度。

定义数据仓库粒度的另外一个要素, 是数据仓库可以使用多种存储介质的空间量。如果存储资源有一定的限制, 就只能采用较高粒度的数据粒度划分策略。这种粒度划分策略必须依据用户对数据需求的了解和信息占用数据仓库空间大小来确定。

选择一个合适的粒度是数据仓库设计过程中所要解决的一个复杂的决定, 因为粒度的确定实质上是业务决策分析、硬件、软件和数据仓库使用方法的一个折衷。在确定数据仓库粒度时, 可以采用多种方法达到既能满足用户决策分析的需要, 又能减少数据仓库的数据量。如果主题分析的时间范围较小, 可以保持最小的数据粒度, 但是只保持较少时间的细节数据。例如, 在分析销售趋势主题中, 分析人员只利用回溯一年的数据进行比较。那保存销售主题的数据只需要 15 个月的数据就足够解决问题了, 不必保存大量的、时间过长的数据。

还有一种可以大幅降低数据仓库容量的方法, 就是只采用概括数据。这样处理后, 确实可以降低数据仓库的容量, 但是有可能达不到用户管理决策分析中对数据粒度的要求。因此, 数据粒度划分策略一定要保证数据的粒度确实能够满足用户的决策分析需要, 这是数据粒度划分策略中最重要的一个准则。



本章小结

数据仓库的开发框架必须依靠数据模型指导。数据仓库的概念模型是描述数据仓库的基本蓝图，数据仓库的逻辑模型是对概念模型的细化，数据仓库的物理模型是数据仓库实施的施工图。

数据仓库数据模型的规范并不是必需的，在某些情况下需要进行反规范化处理，以提高数据仓库的运行效率。星型模型与雪花模型是适合数据仓库设计的数据模型。

数据仓库物理模型包含事实表模型、维模型、元数据模型以及粒度模型。事实表模型是数据仓库存放查询数据的场所，需要采用表的合并、数据的序列设计、引入冗余、对表进行物理分割、生成导出数据以及建立广义索引等技术提高其操作效率。

元数据作为数据仓库中数据的数据，实质上是数据仓库的灵魂，它在数据仓库的开发、应用与维护中发挥着重要的路标作用。因此，需要根据元数据在不同阶段、不同用户、不同的使用方式采用合适的管理方法。

数据粒度的确定最终将影响数据仓库的使用效果，对数据粒度的确定需要根据数据仓库的使用目标、数据仓库的存储空间和数据仓库的使用效率综合确定。



习题

5-1 在一般的信息管理中采用哪些概念模型来描述信息处理的对象，这些概念数据模型是否适合数据仓库的开发环境？

5-2 航空公司希望分析在其服务旅客中的常客旅行趋势，可为公司正确定位航空市场中的常客市场，并且希望跟踪不同航线上旅客的季节变化情况和增长；跟踪在不同航班上所消费的食品和饮料情况，帮助航空公司安排不同航线上的航班和食品供应。现在所面对的任务是为其设计一个数据仓库的概念模型、逻辑模型和物理数据模型。

5-3 为了建立 5-2 题中的数据仓库，需要哪些元数据？这些元数据在不同的阶段应该发挥什么作用？

5-4 5-2 题中航空公司希望将旅客数据至少保持三年，公司每天有 100 条航线，共 300 架次飞行，每架次的旅客平均为 100 人。每架次的食品种类有 50 种，前后共采购过 1000 种。食品受到季节影响较大，每年的食品价格呈现一种周期性变化。食品的详细数据只需一年就可以。请为航空数据仓库设计一个合适的数据粒度模型。

第 6 章

数据仓库开发 应用的阶段

引 言

数据仓库作为决策支持系统 (DSS) 的基础, 具有面向主题的、集成的、不可更新的、随时间不断变化的特性。这些特点说明了数据仓库从数据组织到数据处理, 都与原来的数据库有很大的区别, 这也就需要在数据仓库系统设计时寻求一个适合于数据仓库设计的方法。在一般系统开发中首先需要确定系统的功能, 这些系统的功能一般是通过对用户的需求分析得到的。从数据仓库的应用角度来看, DSS 分析员一般是企业的中上层管理人员, 他们对决策分析的需求不能预先做出规范的说明, 只能给设计人员一个抽象的 (模糊的) 描述。这就要求设计人员在与用户不断的交流中, 将系统需求逐步明确与完善。因此, 数据仓库的开发过程实际上是一个用户和设计人员对其不断了解、熟悉和完善的过程。

通过本章学习, 可以了解:

- ◆数据仓库生命周期中不同阶段的任务
- ◆数据仓库规划内容
- ◆数据仓库的需求定义
- ◆数据仓库的设计实施任务
- ◆数据仓库的使用支持与增强工作

6.1 数据仓库的生命周期

按照生命周期法可将数据仓库开发的全部过程分成：数据仓库规划分析阶段、数据仓库设计实施阶段以及数据仓库的应用等三个阶段。

这三个阶段不是简单的循环往复，而是不断完善、提高的过程。因为，在一般情况下数据仓库系统都不可能在一个循环过程中完成，而是经过多次循环开发，每次循环都会给系统增加新的功能。这种循环的工作永远不会终结，数据仓库系统也就一直处于一个不断完善、不断提高的循环往复过程中。

6.1.1 数据仓库的阶段性的

数据仓库的成功开发应用，需要逐步完善、逐步成长，形成一个不同阶段的开发应用过程。根据诺兰（Nolan）的“阶段理论”，可将数据仓库的开发应用过程划分为创始阶段、成长阶段、控制阶段和成熟阶段。

1. 创始阶段

在数据仓库成长或决策支持的创始阶段，为了迎合一种明确的商业需求，倾向于建立一个数据仓库来提供报表和查询。此时，系统的重点聚焦在单个问题或单个业务单位或部门的单个主题上。这些数据库通常是从企业重要的交易业务文档和数据库中复制或摘录出来而形成的。在这个阶段，使用数据库的主要目的是提供更有效的管理报表，数据仓库的真正潜力是难以预料的。因此，相应的解决方案往往是由对数据仓库有需求的独立部门提出的。这种建造早期的数据仓库的方法在短期内有效，且对以后数据仓库的建造具有指导作用。但是，这些数据仓库只能称之为数据集市，这种方法会限制企业各个部门分享信息。

2. 成长阶段

在随后的情况下，为开展新的商业活动或产生新的管理报表，由此建立的数据仓库通常是前两个或前三个决策支持或商业活动管理的解决方案。它们通常在范围和数据方面受到限制，需要企业内部得到支持，获得再投资或扩大数据量。在这个阶段，更多的数据仓库为更多的应用而建立。但是，由于决策人员仅对业务自身领域内的内容感兴趣，互不相同而分散的解决方案要求数据库内容的多个拷贝，会引起数据的冗余现象发生。随着多个部门分散的数据仓库的建立引起了大量的问题：单一的业务聚集在单一的主题上，没有数据和使用的集成；对冗余数据的管理和存储、数据间的转换没有统一的

标准,存在着多种转换、汇总方法;虽然可以快速提交报表但是可以访问的细节内容太少;数据集市仅在每个团体或业务单位使用;客户数据没有被充分利用。

为了解决出现的这些问题,管理者必须把注意力放在重点业务需求上,需要对共享信息环境的重要意义和有用性具备洞察力,这种环境提供且将实现“惟一的真实版本”。另外,必须获取和利用来自所有部门的信息与客户有密切联系的应用。

3. 控制阶段

在这个阶段要用控制和整合的方法去将应用系统整合,把聚焦点正确地转移到“集中化方法”上。在这一阶段,需要完全改变聚焦点以达到实现成功的数据仓库,以求在企业级的真正数据仓库中,为企业决策分析提供强有力的支持。此时,数据仓库的实现具有一些成熟的特征:惟一的真实版本(关于每个历史主题),公司的存储和交易的历史,数据和表格的整合主题领域,详尽的历史数据,多样化复杂查询,随时小结,在线或实时分析,支持基础设施的整合信息结构,以及集中的知识管理。从而将多个数据仓库结合起来,形成一个决策支持环境。

4. 成熟阶段

随着组织在使用决策支持和数据仓库过程中不断的改进。在将核心信息和客户的交易历史或业务的其他主要领域进行合并和集中后,信息转化为决策知识的比率将大幅度提高,数据仓库逐渐成熟。成熟阶段的数据仓库,可以利用集中和分布的结构将遍及系统内部和外部的信息资源进行整合,实现处理流程的相互连结和决策信息的相互交换,以及适时分配合理的资源。成熟的数据仓库具有以下特征:企业聚焦于集成的信息,大量的来源和不断发展的主题领域,有多种用途的单一业务模型,数据的快速采集与加入,广泛的交易采集和使用,以客户为中心,惟一的真实版本,广泛的访问和管理安全,跨部门的应用,从属的数据集市或从属的数据仓库以及使用数据仓库支持决策。

6.1.2 数据仓库的螺旋式开发方法

数据仓库的开发与应用的阶段是对数据仓库开发应用的生命周期描述。按照生命周期法可将数据仓库开发应用的全过程分成:数据仓库规划分析、数据仓库的设计实施和数据仓库的使用维护三个阶段(参见图 6.1)。完成这三个阶段任务后并不意味着数据仓库开发的终止,而是数据仓库开发向更高阶段发展的一个转变。一方面通过这三个阶段的数据仓库开发,积累了数据仓库的开发应用经验,可以转向其他主题的数据仓库开发应用。另一方面通过对原数据仓库的开发应用经验积累,可对原数据仓库提出改进的建议,使原数据仓库通过改进得到提高。这就是所谓的螺旋式周期性开发方法。这种开发

方法目前在数据仓库的开发应用中占据主导地位。

数据仓库规划分析阶段的工作内容主要包括：调查、分析数据仓库环境，完成数据仓库的开发规划，确定数据仓库开发需求；建立包括实体关系图、星型模型、雪花模型、元数据模型以及数据源分析的主题区数据模型，并且根据主题区数据模型开发数据仓库逻辑模型。

数据仓库设计实施阶段的工作内容主要包括：根据数据仓库的逻辑模型设计数据仓库体系结构；设计数据仓库与物理数据库；用物理数据库元数据填充面向最终用户的元数据库；为数据仓库中每个目标字段确认它在业务系统或外部数据源中的数据来源；开发或购买用于抽取、清洁、变换和合并数据等中间件的程序；将数据从现有系统中传送到仓库中，填充数据仓库且测试。

数据仓库的使用维护阶段的工作内容主要包括：数据仓库的投入使用，且在使用中改进、维护数据仓库；对数据仓库进行评价，为下一个循环开发提供依据。

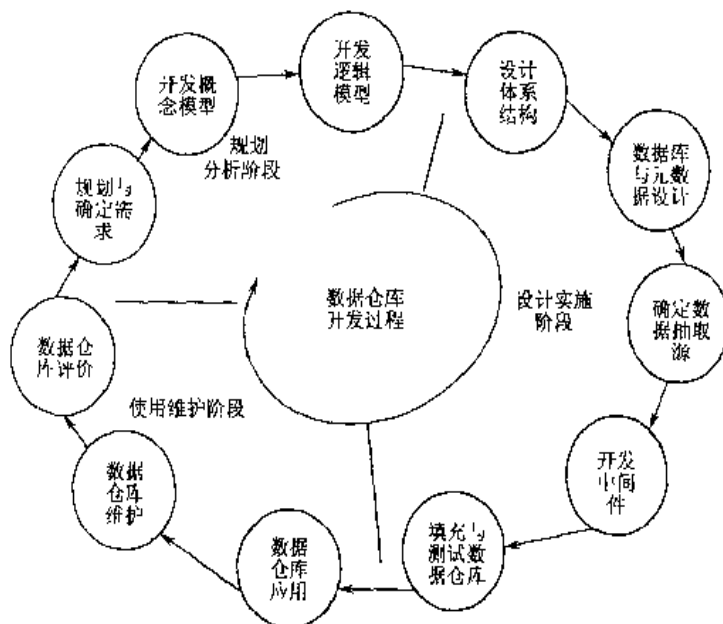


图 6.1 数据仓库的生命周期开发应用全过程

6.1.3 数据仓库的开发特点

数据仓库的使用就是在数据仓库中建立 DSS 应用。这与业务处理系统的应用环境有本质的区别，这也导致数据仓库开发与传统的系统开发在开发出发点、需求确定、开发过程中都有相当大的不同。

1. 数据仓库开发是从数据出发的

创建数据仓库是在原有的数据库系统中的数据基础上进行的，即从存在于操作型环

境中的数据出发,进行数据的创建工作。我们将这种从已有数据出发的数据仓库设计方法称为“数据驱动”的设计方法。“数据驱动”的设计方法就是利用以前所取得的工作成果进行系统建设,就要首先识别当前系统与以前系统的“共同性”,即在进行数据仓库设计前,需要知道原有的数据库系统中已有什么,它们对当前系统设计有什么影响。尽可能利用已有的数据、代码等,而不是全部从头开始做。这是“数据驱动”设计方法的出发点,也是其目的所在。“数据驱动”的设计方法不再是面向应用的,从应用需求出发。数据仓库的设计是从已有的数据库系统出发,按照分析领域对数据及数据之间的联系重新考察,组织数据仓库的主题。“数据驱动”设计方法的中心是利用数据模型有效地识别原有的数据库中的数据和数据仓库中主题的数据“共同性”。

2. 数据仓库使用的需求不能在开发初期明确

面向应用的数据库系统设计往往有一组较确定的应用需求,这是数据库系统设计和开发的出发点和基础。而在数据仓库环境中,并不存在操作型环境中的固定的且较确切的物流、数据处理流和信息流,数据的分析处理的需求更加灵活,更没有固定的模式,甚至可以说用户自己也对所要进行的分析处理不能事先确定。因而在数据仓库的开发初期不能明确了解数据仓库用户使用的需求。

3. 数据仓库的开发是一个不断循环的过程,是启发式的开发

数据仓库的系统开发是一个动态反馈和循环的过程。一方面数据仓库的数据内容、结构、粒度、分割以及其他物理设计应该根据用户所返回的信息不断地调整和完善,以提高系统的效率和性能。另一方面,通过不断地理解用户的分析需求,不断地调整和完善,以求向用户提供更准确、更有效的决策信息。

6.2 数据仓库的规划

数据仓库的开发应用规划是开发数据仓库的首要任务。只有制定了正确的数据仓库规划,才能使组织主要力量有序地实现数据仓库的开发应用。在数据仓库规划中一般需要经历这样几个步骤:选择实现策略,确定数据仓库的开发目标和实现范围,选择数据仓库体系结构,建立商业和项目规划预算。

当数据仓库规划完成后,需要编制数据仓库规划说明书,说明数据仓库与企业战略的关系,以及与企业急需处理的、范围相对有限的开发机会,重点支持的职能部门和今后数据仓库开发工作的建议,实际使用方案和开发预算,作为数据仓库实际开发的依据。

6.2.1 选择数据仓库实现策略

数据仓库的开发策略主要有自顶向下、自底向上和这两种策略的联合使用。自顶向下策略在实际应用中比较困难,因为数据仓库的功能是一种决策支持功能。这种功能在企业战略的应用范围中常常是很难确定的,因为数据仓库的应用机会往往超出企业当前的实际业务范围。而且在开发前就确定目标,会在实现预定的目标后就不再追求新的应用,使数据仓库丧失更有战略意义的应用。由于该策略在开发前就可以给出数据仓库的实现范围,能够清楚地向决策者和企业描述系统的收益情况和实现目标,因此是一种有效的数据仓库开发策略。该方法使用时需要开发人员具有丰富的自顶向下开发系统的经验,企业决策层和管理人员完全知道数据仓库的预定目标并且了解数据仓库能够在哪些决策中发挥作用。

自底向上策略一般从某个数据仓库的原型开始,选择一些特定的为企业管理人员所熟知的管理问题作为数据仓库开发的对象,在此基础上进行数据仓库的开发。因此,该策略常常用于一个数据集市、一个经理系统或一个部门的数据仓库开发。该策略的优点在于企业能以较小的投入,获得较高的数据仓库应用效益。在开发过程中,人员投入较少,也容易获得成效。当然,如果某个项目的开发失败可能造成企业整个数据仓库系统开发的推迟。该策略一般用于企业希望对数据仓库的技术进行评价,以确定该技术的应用方式、地点和时间,或希望了解实现和运行数据仓库所需要的各种费用,或在数据仓库的应用目标并不是很明确时,数据仓库对决策过程影响不是很明确时使用。

在自顶向下的开发策略中可以采用结构化或面向对象方法,按照数据仓库的规划、需求确定、系统分析、系统设计、系统集成、系统测试和系统试运行的阶段完成数据仓库的开发。而在自下而上的开发中,则可以采用螺旋式的原型开发方法,使用户可以根据新的需求对试运行的系统进行修改。螺旋式的原型开发方法要求在较短的时间内快速生成可以不断增加功能的数据仓库。螺旋式的原型开发方法适用于这样一些场合:企业的市场动向和需求无法预测,市场的时机是实现产品的重要组成部分,不断地改进对于企业的市场调节是必须的;持久的竞争优势来自连续不断的改进,系统的改进是基于用户在使用中的不断发现。

自顶向下和自底向上策略的联合使用具有两种策略的优点,既能快速地完成数据仓库的开发与应用,还可建立具有长远价值的数据仓库方案。但在实际使用中难以操作,通常需要能够建立、应用和维护企业模型、数据模型和技术结构的、具有丰富经验的开发人员,能够熟练地从具体(如业务系统中的元数据)转移到抽象(只基于业务性质而不是基于实现系统技术的逻辑模型);企业需要拥有由最终用户和信息系统人员组成的有经验的开发小组,能够清楚地指出数据仓库在企业战略决策支持中的应用。

6.2.2 确定数据仓库的开发目标和实现范围

为确定数据仓库的开发目标和实现范围,首先需要对企业管理者等数据仓库用户解释数据仓库在企业管理中的应用和发展趋势,说明企业组织和使用数据来支持跨功能系统的重要性,对企业经营战略的支持,以确定开发目标。在该阶段确认与使用数据仓库有关的业务要求,这些要求应该只支持最主要的业务职能部门。将使用精力集中在收益明显的业务上,使数据仓库的应用立即产生效果,不应该消耗太多的精力在各个业务上同时铺开数据仓库的应用。在确定开发目标和范围以后,应该编制需求文档,作为今后开发数据仓库的依据。

数据仓库开发的首要目标是确定所需要信息的范围,确定为用户提供决策帮助时,在主题和指标领域需要哪些数据源。这就需要定义:用户需要什么数据?面向主题的数据仓库需要什么样的支持数据?为成功地向用户提交数据,开发人员需要哪些商业知识?哪些背景信息?例如,当前系统或外界数据提供者能否提供这些数据?更新和维护这些数据需要哪些数据源?这就需要定义整体需求,以文件形式整理现存的记录系统和系统环境,对使用数据仓库中数据的候选应用系统进行标识、排序,构造一个传递模型,确定尺度、事实及时间标记算法,以便从系统中抽取信息且将它们放入数据仓库。通过信息范围确定可为开发人员提供一个良好的分析平台,和用户一起分析哪些信息是数据仓库需要的,进行商业活动需要什么数据。开发人员可以和用户进一步定义需要,例如数据分级层次、聚合的层次、加载频率以及需要保持的时间表等。

数据仓库开发的另一个重要目标是确定利用哪些方法和工具访问和导航数据?虽然用户都需要存取并且检索数据仓库的内容,但是所存取的粒度有所不同,有的可能是详细的记录,有的可能是比较概况的记录或十分概况的记录。用户要求的数据概括程度不同,将导致数据仓库的聚集和概括工具的需求不同。数据仓库还要具有一定功能来访问和检索图表、预定义的报表、多维数据、概况性数据和详细数据。用户从数据仓库中获取信息,应该有电子表格、统计分析器和支持多维分析的分析处理器等工具的支持,以解释和分析数据仓库中的内容,产生并且验证不同的市场假设、建议和决策方案。为将决策建议和各种决策方案向用户清楚地表达出来,需要利用报表、图表和图像等强有力的信息表达工具。

数据仓库开发的其他目标,是确定数据仓库内部数据的规模。在数据仓库中不仅包含当前数据,而且包含多年的历史数据。数据的概括程度决定了这些数据压缩和概括的最大限度。如果要让数据仓库提供对历史记录进行决策查询的功能,就必须支持对大量数据的管理。数据的规模不仅直接影响决策查询的时间,而且还将直接影响企业决策的质量。

在数据仓库的开发目标中还有：根据用户对数据仓库的基本需求，确定数据仓库中数据的含义；确定数据仓库内容的质量，以确定使用、分析和建议的可信级别；哪种类型的数据仓库可以满足最终用户的需要，这些数据仓库应该具有怎样的功能；需要哪些元数据，如何使用数据源中的数据等。

数据仓库的开发目标多种多样、十分复杂，需要开发人员和用户在开发与使用的过程中不断交互完善。因此，在规划中需要确定数据仓库的开发范围。使开发人员能够根据需求和目标的重要性逐步进行，并且在开发中吸取经验教训，为数据仓库在企业中的全部实现提供技术准备。因此在为数据仓库确定总体开发方向和目标以后，就必须确定一个有限的能够很快体现数据仓库效益的使用范围。在考虑数据仓库的应用范围时，主要从使用部门的数量和类型、数据源的数量、企业模型的子集、预算分配以及开发项目所需要的时间等角度分析。在分析这些因素时可从用户的角度和技术的角度两方面进行。

从用户的角度应该分析哪些部门最先使用数据仓库？是哪些人员为了什么目的使用数据仓库？以及数据仓库首先要满足哪些决策查询？因为这些决策查询往往确定了对来自数据源数据的聚集、概括、集成和重构等技术要求，同时决策查询的范围还确定了关于数据维数、报表的种类，这些因素都将确定数据仓库定义时所需要的数量关系。查询的格式越具体，越容易提供数据仓库的维数、聚集和概括的规划说明。

从技术的角度分析，应该确定数据仓库元数据库的规模，数据仓库的元数据库是存储数据仓库中数据定义的模型。数据定义存储在仓库管理器的目录中，可以作为所有查询和报表工具构造和查询仓库的依据。元数据库的规模直接表示数据仓库中必须管理的数据规模。通过对元数据库规模的确定，实际上就确定了数据仓库所需要管理的数据数量。

6.2.3 数据仓库的结构

数据仓库结构可以进行灵活的选择，可将组织所使用的各种平台进行恰当的分割，把数据源、数据仓库和最终用户使用的工作站分割开来进行恰当设计。

1. 数据仓库的应用结构

(1) 基于业务处理系统的数据仓库

在这种结构中，将运作的数据用于无须修改数据的只读应用程序中。具有这种结构的数据仓库元数据库是一种虚库，它指向业务数据库的元数据，而不是数据仓库自身的元数据。在数据仓库元数据库的直接指导下，对仓库的查询就是简单地从业务数据库中抽取数据。

(2) 单纯数据仓库

利用在数据仓库中的数据源净化、集成、概括和集成等操作,将数据源从业务处理系统中传输进集中的数据仓库,各部门的数据仓库应用只在数据仓库中进行。这种结构经常发生在多部门、少用户使用数据仓库的情况下。这里的集中仅是逻辑上的,物理上可能是分散的。

(3) 单纯数据集市

数据集市是指只在部门中使用的数据仓库,因为企业中的每个职能部门都有自己的特殊需要,而统一的数据仓库可能不满足这些部门的特殊要求。这种体系结构经常发生在个别部门对数据仓库的应用感兴趣,而组织中其他部门却对数据仓库的应用十分冷漠之时,由热心的部门单独开发时所采用。

(4) 数据仓库和数据集市

企业各部门拥有满足自己特殊需要的数据集市,其数据从企业数据仓库中获取,而数据仓库则从企业各种数据源中收集和分配。这种体系结构是一种较为完善的数据仓库体系结构,往往发生在组织整体对数据仓库的应用感兴趣之时所采用的体系结构。

2. 数据仓库的技术平台结构

(1) 单层结构

单层结构主要是指在数据源和数据仓库之间共享平台,或者让数据源、数据仓库、数据集市与最终用户工作站使用同一个平台。共享一个平台可以降低数据抽取和数据转换的复杂性,但是共享平台在应用中可能遇到性能和管理方面的问题。这种体系结构一般在数据仓库规模较小,而组织的业务系统平台具有较大潜力之时所采用。

(2) 客户/服务器两层结构

一层为客户机,一层为服务器。最终用户访问工具在客户层上运行,而数据源、数据仓库和数据集市位于服务器上。该技术结构一般用于普通规模的数据仓库。

(3) 三层客户/服务器

基于工作站的客户层、基于服务器的中间层和基于主机的第三层。主机(宿主)层负责管理数据源和可选的源数据转换;服务器运行数据仓库和数据集市软件,并且存储仓库的数据;客户工作站运行查询和报表运用程序,且还可以存储从数据集市或数据仓库卸载的局部数据。在数据仓库稍具规模,两层数据仓库结构已经不能满足客户的需求,要将数据仓库的数据存储管理、数据仓库的应用处理和客户端应用分开之时,可以采用这种体系结构。

(4) 多层式结构

这是在三层客户/服务器上发展起来的数据仓库结构。在该结构中从最内数据层到最外层的客户层依次是,单独的数据仓库存储层,对数据仓库和数据集市进行管理的数据

仓库服务层,进行数据仓库查询处理的查询服务层,完成数据仓库应用处理的应用服务层和面向最终用户的客户层。体系层次可能多达五层,这种体系结构一般用于超规模数据仓库系统。

6.2.4 数据仓库使用方案和项目规划预算

数据仓库的实际使用方案与开发预算,是数据仓库规划中最后需要确定的问题。因为数据仓库主要用于对企业管理人员的决策支持,确保其实用性是十分重要的,因此需要让最终用户参与数据仓库的功能设计。这种参与是通过用户的实际使用方案进行的,使用方案是一个非常重要的需求原型。实际使用方案必须有助于阐明最终用户对数据仓库的要求,这些要求有的只使用适当的数据源就可以得到基本满足,而有的则需要来自企业外部的数据源,这就需要通过使用方案将这些不同的要求联系起来。

实际使用方案还可以将最终用户的决策支持要求与数据仓库的技术要求联系起来。因为当用户确定最终要求后,就为数据仓库的开发提供了一个有效范围。其次,可以确定数据仓库元数据库,为元数据库的范围确定一个界限。还可确定所需要的历史信息数量,当根据特定的用户进行数据仓库的规划时,就可确定数据的抽取、净化、集成、转换、概括和聚集等操作的复杂程度。并且可以确定最终用户所关心的维度(时间、方位、商业单位和生产企业),因为维度与所需要的概括操作有明显的关系,必须选择对最终用户有实际意义的维度,例如“月”、“季”、“年”等。最后,还可确定数据集市/数据仓库的结构需要,使设计人员确定采用单纯数据仓库结构,还是单纯数据集市结构或两者相结合的结构。

在实际使用开发方案确定后,还需要对开发方案的预算进行估计,确定项目的投资数额。投资方案的确定可以依据以往的软件开发成本,但是这种预算的评估比较粗糙。另一种方法是参照结构进行成本评估。也就是说,将数据仓库实际使用方案所确定的构件进行分解,根据各个构件的成本进行预算估算。数据仓库的构件包含在数据源、数据仓库、数据集市、最终用户存取、数据管理、元数据管理、传输基础等部分中,这些构件有的在企业原有信息系统中已经具备,有的可以选择商品化构件,有的则需要自我开发。根据这些构件的不同来源,可以确定比较准确的预算。

在完成数据仓库规划后,就需要编制数据仓库开发说明书,说明系统与企业战略目标的关系,以及系统与企业急需处理的、范围相对有限的开发机会,所设想的业务机会的说明以及任务概况说明、重点支持的职能部门和今后工作的建议。数据仓库项目应由明确的业务价值计划开始,在计划中需要阐明期望取得的有形和无形利益。有形利益包括允许顾客直接访问数据仓库而降低了顾客服务成本等。无形利益包含利用数据仓库使决策完成得更快更好等利益。业务价值计划最好由目标业务主管来完成,因为数据仓库

是用户驱动的，应该让用户积极参与数据仓库的建设。在规划书中要确定数据仓库开发目标的实现范围、体系结构和使用方案及开发预算。

6.3 数据仓库的需求定义

数据仓库的需求定义是在数据仓库规划以后，为数据仓库的分析设计所做的准备工作。在对数据仓库的需求定义中需要对数据仓库应当具有的功能进行精确的说明，它需要业主、用户团体和开发、设计人员在整个开发过程中密切合作。需求定义除要确定数据仓库必须的特征和功能外，还需要明确指出使用数据仓库的操作环境。在创建数据仓库的过程中对于需求的理解是必不可少的，需要对来自多个领域的需求进行详细分析，这些不同领域的数据仓库需求往往是不一样的。数据仓库的需求分析根据不同领域的需求分析可以划分为来自业主的需求、来自设计的需求、来自开发者的需求和来自最终用户的需求等几个方面。

6.3.1 定义业主的需求

数据仓库的业主（投资者）关心的是创建数据仓库的目标，建立数据仓库给组织战略带来的影响，创建数据仓库所需的投资费用的多少以及所具有的应用前景。业主把数据仓库看做在整个商业活动中的一个 DSS（决策支持系统）构件。在 DSS 中，业主利用数据仓库的分析和合理的论据做出商业决策。

业主常常参与数据仓库概念模型的认可和评审开发，批准开发方案，为系统以后的不断升级和连续投资进行决策。在数据仓库开发的初始阶段，需要为投资者提供一种模型或方法进行数据仓库规划，包括预测投资费用、预测收益以及风险管理规划等。

规划的具体内容是对这样几个方面进行估计：数据仓库产品和服务所提供的范围、所需的人员和技术、对已有的商业环境和人员的影响、对投资的影响。

费用和收益分析主要是通过估算数据仓库方案的最初投资以及日后的维护和升级费用等数据仓库投资费用，并且估算将数据仓库信息发送给用户所获得的盈利情况，然后，据此计算初步的投资回报率（ROI, Return On Investment）。

风险管理可以帮助业主（投资者）选择实现途径，并且权衡各种实现方案的利弊。

6.3.2 定义设计者的需求

数据仓库设计者不但要收集业主对数据仓库的商业需求，还要收集实现这些商业需求所需的技术要求。设计者介于投资者和实现者之间，前者从较高程度的概括上考虑数

据仓库的商业需要,后者关心的是数据仓库详细规格说明。从商业的最终目的来看,设计者见到的是数据仓库中作为信息的商业实体和处理过程,以及能够满足最终用户需求的各种组织信息的方法。从技术角度来看,设计者所要解决的是数据仓库同已有业务处理系统以及应用程序之间的相互影响,使其保持一致性。

设计者需要根据数据仓库方案参照框架结构,选择数据仓库的实现方法。设计者还要设计一种结构框架,展示数据仓库应用程序的主要部分以及单独部分是如何有机地结合在一起的。用于结构设计的一种最有效的方法,是企业结构规划(EAP)法。在这种方法中,设计者可以实现以下3种结构。

1. 数据结构

数据结构描述的是数据项及其相互关系。数据特征化是一个最基本的活动,因为如果没有定义必须创建或修改的数据,就无法开发应用程序,数据结构一般用实体-关系模型、星型模型或雪花模型表示。

2. 应用程序结构

一个系统就是多个应用程序的联合,同时,应用程序决定了系统的功能。数据结构定义完成之后才能定义应用程序结构。应用程序结构是对应用程序功能和接口的分类,它通过一个“CRUD”阵列与数据结构相对应来确定。每个应用程序都是用它所创建(C)、读取(R)、更新(U)或删除(D)的一个或多个数据项来相互联结的。“CRUD”阵列方法的具体应用将在第7章详细介绍。

3. 技术结构

技术结构是对所有技术成分的描述,它通过将一个系统分解为各个技术模块(如服务器计算机、用户工作站、图形用户界面、关系数据库系统、数据字典和元数据库等)而建立的。

按照第1章所介绍的数据仓库参照结构进行分类,就可以开发以上的各种结构。通过找出参照结构的各个模块的各种数据和元数据要素及其相关关系,开发数据结构。通过参照结构各个模块的所有应用程序及其特征和功能关系,开发应用程序结构。通过参照结构的块和层的各种选择,开发技术结构。

6.3.3 开发者的需求定义

开发者负责数据仓库各个构件的物理安装和集成。开发者需要把设计者开发的数据结构、应用程序结构和技术结构进一步分解为特定的应用程序、接口、计算机、数据库

和用户界面。从系统开发者的角度来看,数据仓库必须清晰地从提供给最终用户的数据形式和格式中分离出数据源(可得到的数据)。

开发者的需求是在决策基础上对设计者的需求所做的提炼,它考虑了平台选择以及在所选平台上对数据结构和应用程序结构的分割。开发者的需求还与技术结果的详细描述和规格说明有关,如程序设计语言、关系数据库系统访问和通信协议。开发者还要确定技术需求、使用需求、数据仓库产品就绪需求和开发与试用人员及其技术需求。

1. 技术需求

需要确定数据源中的数据和元数据、数据和元数据抽取、数据存储、数据分配和管理、网络和通信、在线事务处理程序和操作环境、工作流程管理与标准。

确定数据仓库和数据集市中的数据求精和重构工程、网络和通信、数据仓库和数据集市处理程序和操作环境、数据仓库和数据集市数据和元数据存储、元数据分类、工作流程管理与标准。

确定最终用户存取工具块中的存取及检索中间件、局部存储、多维数据环境、元数据查询与报表工具、标准、商业用户工具(如分析与报表、商业建模、数据挖掘、数据冲浪、在线分析处理)等。

2. 试用需求

主要用于衡量数据仓库是否能够随时方便地提供存取和分配信息的能力,即衡量为最终用户工具和指定的数据仓库应用程序提供一系列存取方法以及提供一条连接到局域网(LAN)的工作站连接路径。因此,在使用需求中需要考虑存取方法、发送方法、访问工具、连接需要和客户平台需求等。

3. 数据仓库产品就绪需求

数据仓库能否成功还取决于是否是一个关键任务系统,是否能够经常用于对企业运作和战略的决策支持。因此需要在数据仓库产品就绪需求中考虑管理的鲁棒性和可用性;维护信息的一致性、准确性、可靠性和准确性;用于更新和维护数据仓库元数据库及元数据的策略和过程的定义;提供存取控制,确保传输机制、数据库、计算机和通信机制准备就绪,随时可用,为用户提供技术支持和桌面帮助能力;提供安全存取的策略和过程等。

4. 开发与使用人员及其技术需求

数据仓库的开发需求主要考虑在开发过程中所需要的不同技术。在不同的开发过程中需要不同的技术,可以在分析数据仓库结构层次和块结构的基础上将技术进行分类。数据仓库的开发涉及大量的工作人员,对这些人员的作用可以针对仓库的结构将工作人

员的作用表示到对应的层次、块状结构中。

6.3.4 最终用户的需求定义

数据仓库相对最终用户而言是一个黑箱,他们只能通过查询和报表工具以及数据仓库内部信息的某种映射关系来访问数据仓库内部数据。最终用户的需求主要体现在对工作流程的分析、决策的查询需求、报表需求、操作需求和数据需求等方面。最终用户从数据仓库中检索到所需要的数据后,可以指定数据分析的类型。这些数据的分析操作主要是对数据项进行揭示更多细节的分片和细剖,寻找企业隐含行为的数据挖掘,对数据进行卸载或局部修改、建立商业模型。在对数据进行分析时可从二维或多维的、电子表格的、关系的、报表的、图表的和运营样本的数据等方面进行分析。

通过数据仓库的辅助决策功能,可以使用数据仓库的功能将现有的工作流程改进,形成一个改进的工作流程,将数据仓库的功能合并到改进后的工作流程中。不同用户对查询所提出的要求有所不同,分别来自销售部门、市场部门、生产部门所关心的问题 and 数据是不同的,因而他们对于查询的数据需求也就不一样。每个最终用户对报表的要求不同,每个部门的报表需求的范例格式也就不同,单一的报表工具很难满足数据仓库的所有报表要求。总之,数据仓库可由各种不同的用户和工具进行访问,每个工具都以某种报表为目标进行优化,可以进行报表的编写、批处理文件的创建、联机分析处理、数据采集等等。因此,在对最终用户需求分析时,需要用户提供各类查询要求所用的报表样例,最终用户还可指定数据查询的要求。从数据仓库中检索出数据后,最终用户还可进行数据分析的类型指定,包括对数据进行操作的类型和数据输出的类型的指定。通过分析那些从数据仓库中最终输出的报表样例,可以确定要将哪些数据存储在数据仓库中。为数据仓库中存储的数据建立模型,是实现数据仓库的开始。数据模型是面向主题建立的,同时可为多个面向应用的数据源的集成提供统一标准。数据仓库的数据模型一般包括企业的各个主题域,各个主题域之间的关系,描述主题的键码和属性等。

在进行数据仓库的最终用户需求分析时,还需要对数据仓库的主题域、信息的粒度、数据仓库的维度进行分析。数据仓库的主题域是数据仓库设计必须考虑的问题。主题域选择得好,可使数据仓库最大限度地发挥作用。例如在市场销售中,销售部门较关心的主题有市场研究、竞争分析、购买者特点分析、替代产品的竞争压力、销售预算决策、促销决策、销售渠道决策和市场趋势预测等。通过对这些主题的讨论,可使销售部门在市场竞争中处于主动地位。

数据仓库的粒度是指信息的详细程度。粒度与数据仓库的聚集和概括操作有直接关系,粒度越低,信息的数量越多。通常业务数据的粒度都非常低,而决策人员并不需要这些浩如烟海的详细信息,而是粒度较高的概括性信息,这样才能保证决策的实用性。

这就需要数据仓库能够对业务数据进行概括和聚集。通常粒度越高,转换和概括业务数据所需要的处理次数就越多,高粒度的数据需要的存储空间较小,用户越能够快速、方便地进行查询。

数据仓库的维度是指在数据仓库中利用多维分类机制,组织业务数据和历史数据,提供决策人员进行决策分析。数据仓库需要对数据源中的业务数据标明时间标记,为数据建立时间参数。数据仓库可将同一时间范围内产生的数据聚集起来,作为对某个决策查询的响应。在决策查询中经常使用的一些维有时间、客户群、产品族、地理分布、单位结构、单位特征和产业特征等。

因为数据仓库的建立必须从企业的业务过程和相应的信息主题分析入手,只有依赖于业务过程和相应的信息才能获取有关的企业数据模型背景。这就需要调查人员充分了解业务过程、以及对信息的要求和所做出的决策。在调查刚开始时不是问用户需要“什么数据”,而是要问用户所需要的数据是用来支持什么样的决策/分析标准。这样才能够使用户更容易地表达对数据仓库的希望和需求。

6.3.5 数据仓库的数据模型设计

在进行数据仓库的用户需求调查以后,就可确定数据仓库的开发主题区。通常数据仓库的主题区要覆盖企业的某个完整业务对象,例如销售信息、财务信息或顾客信息等。在数据仓库的数据模型开发中,必须从企业的业务过程和相应的信息主题分析入手,只有依赖于对业务过程和对信息分析,才能获取有关模型建立的背景资料。因此需要充分理解业务过程,业务过程对信息的要求,以及在管理决策分析中根据信息进行的决策。在建模开始时不要急于考虑用户需要“什么数据”,而要分析用户所需要的数据是用来支持什么样的决策以及相应的信息分析标准和度量尺度。在建模时还需要注意确定元数据的值,为最终用户建立元数据库做准备。

数据仓库的数据模型应该尽可能完备,同时要有实现部分模型的决策,并在数据仓库模型中体现出来。一个强大而持久的数据仓库模型,不仅要能提供数据仓库开发的依据,还要为在管理决策支持活动中使用数据提供业务过程方面的背景知识。数据结构是总体结构中的主要组成部分,它应该给出数据仓库的实施规划,能够在实际中支持某种动作的数据仓库,并且支持“WHAT IF”分析和决策支持的多维“数据立方体”,为数据与由一个数据仓库支持的核心业务过程之间建立映射关系提供基础。

6.4 数据仓库的设计和实施阶段

数据仓库的设计和实施是从数据仓库的数据模型开始的。在数据仓库的设计和实

中需要建立数据仓库与业务处理系统的接口,完成数据仓库的体系结构设计,实现数据仓库物理仓库与元数据库,确定数据源及其源数据的抽取准则、完成数据仓库的中间件设计。

6.4.1 数据仓库的数据源确定以及与业务处理系统接口的设计

1. 数据仓库的数据源确定

要为数据仓库从数据源中抽取为管理决策分析所使用的的数据源,首先要对所抽取的数据源进行正确的定义,数据源的定义要确定数据仓库主题所需各数据源的详细情况,包括数据源所在计算机平台、拥有者、数据结构、使用该数据源的处理过程、数据仓库更新计划等。这个定义涉及数据仓库中每个目标列,以及每个目标列在业务系统或外部数据源中的数据来源。

为了保证数据的更新需要,还需要为数据仓库中的每个目标列确认它在业务系统或外部数据源中的数据来源规则,以便利用数据获取中间件,从源系统中获取数据,并且加载到数据仓库中。数据获取中间件是在仓库开发者所制订的数据来源规则基础上开发的,或根据这些规则设置有关的数据抽取参数。规则还需要规定在把数据加载到数据仓库数据库之前对它所做的清洁和增强操作。

在数据源的定义中还要确定数据源抽取原则,这样才能确定从哪些数据源中抽取所需数据,数据如何转换,装载到主题的哪个数据表中。

2. 数据仓库与业务处理系统的接口设计

在确定了数据仓库的数据源以后,就需要开始考虑数据仓库与作为数据源的业务处理系统的接口设计。

由于在业务处理系统环境中,各个应用系统都有自己独立的、特殊的需求,在各自的过程中没有考虑到以后与其他系统的集成问题。在其基础上建立的数据仓库,需要完成与业务处理系统接口的设计。所设计的接口应该具有这样一些功能:从面向应用和操作环境生成完整的数据;数据基于时间的转换;数据的聚集;对现有数据系统的有效扫描,以便今后数据仓库的数据追加。数据追加的方法主要有:对操作型数据打上时间戳、使用系统日志或审计日志、修改程序代码、使用前映像或后映像文件。

6.4.2 数据仓库的体系结构与数据库设计

在数据仓库逻辑模型设计基础上可以进行数据仓库的体系结构设计。此时,需要确定数据仓库的三个层次——信息获取层、信息存储层和信息传递层的结构。在完成数据

仓库的体系结构和数据库设计后,可以提交数据仓库定义、中间件的设计要求、数据仓库测试数据和用户验收与质量保证计划,以及数据模型与物理数据库的映射文档,逻辑设计与物理设计之间的映射文档。文档确认了数据模型中的每个实体与在数据仓库中相应表格名称之间完整的映射关系。

1. 数据仓库的体系结构设计

数据仓库的体系结构框架是影响数据仓库性能的关键因素之一,数据仓库的体系结构框架决定了数据加载、访问和传递的方式。在确定数据仓库结构时需要考虑最终用户和数据使用部门的数目、数据的多样性和数量、更新周期,以及存储访问的难度。在数据仓库体系结构中应该设计三个独立的数据层次:信息获取层、信息存储层和信息传递层。

信息获取层负责数据的收集、提纯、净化和聚合,以及从组织外部数据源和组织的业务处理系统中获取数据。这些数据应该是准确的,并且要被用于各个部门进行决策支持,因此需要有通用的含义。在分析馈送系统时,某些应用程序将被指定为记录系统。这意味着,大多数可靠的应用程序将被用做特定数据加载的数据源。例如,当加载顾客指示数据时,要确保其来源是顾客信息文件,如果这样的文件不止一个,就应该确保差别只在于更新周期的不同,而不是与它有关的实际数据值存在差异。

信息存储层提供包含时点信息的单一逻辑信息,这种数据通常以最分散的方式存放。需要尽可能使物理设计符合数据模型,这对最终产生满足各种设计要求的灵活性是十分重要的。

信息存储层是一个保存数据的区域,这些信息是在信息传递层次中可以得到的信息,因为数据仓库必须存储来自许多不同地方的业务数据。对于支持集成传递要求所必需的性能水平,单一的设计会产生消极影响。因此数据仓库的一个重要特征就是灵活性,在体系结构中需要利用信息传递层来实现灵活性。

信息传递层是数据仓库结构中支持一套共用的表示工具和分析工具的组成部分,它通过可以在职能工作站上进行报表生成和查询提供数据需求。这是最终用户与数据仓库交流的层次,是数据仓库与用户接触的首要地点。其中,数据集市作为数据仓库的对外联系场所,是传递信息的载体,但是只存放了某些业务职能所需要的数据,它具有业务用户要求的特定格式和粒度。例如,“客户收入”的数据字段在信息存储层通常是按月保存的,“财务管理”的职能部门要求对迄今为止的情况加以汇总。这一数据是从所有存在数据仓库中的每月数据值中衍生出来的,可以在信息传递层上得到。为了做到这一点,数据仓库必须存储必要的衍生规则或概括规则。

为了对不同的数据进行操作,必须要有相应的技术体系结构,数据仓库的技术体系结构通常包含设计模块、数据获取模块、数据管理员模块、管理模块、信息目录模块、

数据访问模块、中间件模块和数据传递模块等。在这些技术体系的支持下，才能使数据仓库正常运转。

2. 数据仓库数据库设计

数据仓库的数据库主要包含存储用户进行决策分析的数据库和描绘数据的元数据库。

存储用户分析数据的数据库可以采用关系型数据库、多维数据库和对象数据库实现。在目前的数据库技术条件下，数据仓库开发人员往往采用关系数据库实现数据仓库。在数据量较少的情况下，可以采用多维数据库实现数据仓库的数据存储。

元数据库是数据仓库的灵魂。没有元数据库，用户就无法对数据仓库数据进行良好的定义、组织、管理。数据管理是企业信息价值链的基础。元数据库被用于支持数据仓库和传统应用程序开发项目而组织和使用元数据的过程，元数据库是了解企业背景和核心业务过程中信息内容的一把钥匙，是数据后勤系统的核心，组织和使用元数据是企业数据仓库战略的关键性成功因素。元数据库为过程之间、工具之间和数据库之间的管理连接奠定了基础，是数据仓库的基础。元数据库对于重要业务过程的自动化和信息化具有基础性的作用。元数据库还是存放各种有关模型的地方。就数据和过程（元数据）的技术定义而言，它同样也可以将模型转换成应用程序设计规格的数据集市。无论应用程序是面向对象技术启动的，还是用高级决策支持系统的强大报表生成和查询工具启动的，元数据库都将成为企业数据资源的协调点。因为数据仓库用于组织企业数据，并使其能够为管理者所用，而元数据库则用于存储和管理一个时常变化的企业模型。

6.4.3 数据仓库的中间件设计

数据仓库的中间件能将数据仓库的各个组成部分，以人们不易察觉的方式无缝地整合在一起。数据仓库中的中间件主要包括进行数据抽取、转换、复制的拷贝中间件，用于数据库访问的网关中间件，对数据仓库进行监控的中间件。

拷贝中间件主要有如下 4 种。

1. 代码发生器

代码发生器产生专用的数据获取程序，在数据结构定义和由数据获取模块规定的清洁规则和增强定义的基础上，生成专用的 3GL/4GL 获取程序。

2. 数据复制工具

数据复制工具能够捕捉一个系统中源数据库的变化，并且可将这些变化加到数据准备区的源数据库副本上。

3. 数据泵

数据泵通常在既非源数据库系统也非目标数据库系统的服务器上运行，在由用户确定的间隙将数据吸入泵服务器中，然后将获取的数据加入数据仓库。

4. 广义数据获取工具和设备

广义数据获取工具和设备主要用于从源系统中将数据拷贝到目标系统上。

拷贝中间件还应该能够进行数据清洁工作，可对记录或字段重组、去除业务数据、供给已丢失的字段值和检查数据的完整性和一致性，包含对字段值的解码和转换，增加数据的时间戳（如果源数据中没有这一属性）以反映数据的当前值，数据的概括或者衍生值的计算。许多数据清洁中间件可以用于对数据进行清洁或增强，有的中间件注重数据的结构变化，另一些中间件则主要用于数据内容的清洁。

拷贝中间件还应该能对准备加入数据仓库的源数据按照数据仓库的结构进行变换和合并，以适应数据仓库结构的要求。

这些拷贝中间件可从商品化的中间件中选择，或由数据仓库开发人员自行开发。

用于数据库访问的网关中间件，主要用于解决数据仓库与数据源和客户端之间的网络协议不同所造成的数据传输困难的问题。

对数据仓库进行监控的中间件主要用于对数据仓库的应用选择适当的资源，例如，可以根据系统的负载选择恰当的计算机完成数据库事务；如果有的计算机不可用，可以自动选择可用的计算机进行事务处理；可以根据用户对数据仓库的使用频率以及数据量来调整数据仓库。

网关中间件和监控中间件的实现很少自行设计开发，一般采用商品化的中间件。只是在选择过程中一定要注意构成数据仓库的数据库是否已经具有这些功能，如果数据库中已有这些功能，就不必再购买其他商品化中间件。

6.4.4 数据仓库的数据抽取

已经建立的物理数据仓库仅提供一个供用户访问的数据存储结构，其中并没有任何数据资源。为了能在数据仓库中使用数据资源，必须将数据资源从外部抽取到数据仓库中。

数据仓库的数据抽取是数据仓库成功的关键。“垃圾进，垃圾出”的原则说明了数据抽取的重要性。在操作数据上执行的数据抽取，应该依据元数据中定义的标准数据格式处理数据。例如，一个传统的业务处理系统可用 20 个字符的字段存放姓名信息，然而在数据仓库设计中的域标准中用 30 个字符进行存储。在抽取过程中，应该在将数据传递到

数据仓库系统之前，从元数据存储中读到这种域定义，将数据转换或修补以适合新标准。

数据的抽取处理实际上被个别情况所驱动。任何数据抽取的设计基础由实际应用、数据容量和数据的易变性所决定。因此，在数据模型中必须清晰地定义数据仓库的数据需求、数据量、数据的易变性等细节问题。

6.4.5 数据仓库的数据加载

在数据被抽取后，可把数据加载到数据仓库中。在数据加载之前，首先需要对准备加载的数据进行清理，即对数据按照标准进行格式化处理，这些清理工作可在一个专门的数据清理区或数据准备区内进行。数据的清理工作必须严格依据元数据的定义进行，一旦数据清理结束，可将经过净化和转换的数据加载到合适的数据仓库事实表中。在数据加载后，还要更新元数据仓库中的元数据，以反映刚完成的数据加载活动，并且对受影响的概括数据重新概括处理。数据的加载活动应该使用标准方法和公用工具，例如在关系 DBMS 中可以使用 SQL，或专门用于管理数据仓库的 DBMS 加载工具。这样可在提供加载数据仓库的最有效方式的同时，最小化定制开发工具的需要。否则，需要根据数据抽取和转换过程的需要，自行设计一些定制加载过程。

6.4.6 数据仓库数据的复制与发行

在分布式数据环境中，开始越来越多地采用复制技术。复制就是在分布式的站点上创建和维护与原始数据有关的拷贝过程。数据复制不仅是把数据从一个站点拷贝到另一个站点，它还包括用简单的接口解决一些复杂的问题。一个完整的复制结构应能完成以下的复杂任务：

- 不会受到系统失败等问题的影响，保证提供可靠的数据复制；
- 只传送符合数据完整性规则的一致数据；
- 可以优化传送过程，减少在捕获或修改数据和复制品作为结果传送之间的等待时间。

复制技术具有一种持久的能力，它可以将一个数据库中的数据拷贝到与其连接的机器上的另一个数据库中；可从一个系统向另一个系统不断发送数据，同时在适当的控制下进行修改。复制服务器一般是运行数据库的另一个机器，可以管理数据和目标数据之间的复制过程。复制服务器负责修改源数据，将所有修改后的复制信息都发送到目标服务器。利用复制技术可以很方便地保持数据仓库数据与数据源的同步，以及数据集市与数据仓库的同步。同一个复制程序可将所复制的数据分配给多个目标，可以用做多个数据集市复制数据的分配通道。复制可以方便地建立“备用的”数据库系统。所谓“备用

的”数据库系统，是指与源系统等价的系统。当源系统发生故障时，应用程序可以切换到目标系统，因为它含有与源系统完全相同的数据。

在数据仓库中，数据的发行能够保证数据仓库的各个子系统从中心数据仓库中得到所需要的一切数据。数据仓库的子系统从抽取、净化和加载过程中获得正确的源数据。为了支持这种数据的发送，要求在整个系统中开发一套标准工具。为了完成数据的发行，为了使它对所有的用户有用，就要保证从数据仓库到数据用户、标准报表、数据分析系统或部门数据中心的数据移动工作的协调进行。技术和数据结构应保证数据发行系统完成以下的功能：

- 保证数据发行以适时和有效的方式进行；
- 保证只发送被排序的数据；
- 建立正确的和所需要的服务水平标准。

6.4.7 数据仓库的测试

在完成数据仓库的实施阶段中，需要对数据仓库进行各种测试。测试工作主要包括单元测试和系统集成测试。

1. 单元测试

当数据仓库的每个单独组件完成后，就需要对它们进行单元测试，单元测试的目的是寻找存于单个程序、存储过程和其他位于一些独立环境中的模块的错误。在测试过程中不仅要求单元能对各种正常情况进行正确处理，而且也要求对各种错误情况具有防御能力，不至于由于某个用户的误操作导致系统的崩溃。

2. 系统集成测试

在完成数据仓库单元测试以后，需要进行数据仓库的集成测试，测试目的是验证每个单元与数据仓库系统和子系统之间的接口完好，能够正常传递数据，执行系统的整体功能。系统的集成测试需要对数据仓库的所有组件进行大量的功能测试和回归测试。在测试过程中必须对所有的测试方案和测试结果进行详细记录，以便对测试中所发现的错误进行再现测试。在测试之前必须依据数据仓库的所有组件功能、数据仓库应用方法和数据仓库开发计划，制定详细的测试计划。

在完成数据仓库的系统集成测试以后，就可以进行数据仓库数据的首次加载。完成数据仓库的首次加载后还需要从数据质量、可用性和性能等方面，测试用户对数据仓库的满意程度。

在数据仓库交付用户使用之前，需要对数据仓库进行交付测试。通过交付测试，了

解数据仓库在哪些方面真正地满足用户的需要。在交付测试中需要用户真实地使用数据仓库，观察用户在使用数据仓库过程中与数据仓库的交互方式，用户如何使用数据仓库完成决策分析工作？在进行管理决策分析时，数据仓库是否能够满足用户的需要？数据仓库需要进行哪些变动才能更好地为用户服务？通过交付测试不仅要使用户接受数据仓库，更重要的是能够清楚地定义所交付的数据仓库在实际应用中还存在哪些缺点？并且收集数据仓库在可用性方面、改进内容、数据准确性以及数据仓库需要补充的内容等方面的用户意见。这些信息将为数据仓库开发者在今后的数据仓库维护或数据仓库新主题开发中提供帮助。

6.5 数据仓库的使用、支持和增强阶段

当数据仓库部署完成并通过实况测试后，就可提交用户试用，且在用户的使用过程中对数据仓库的质量、可用性和性能等方面进行评估。此时，实际上已经进入用户对元数据库和实际数据仓库雏形的试用、探索期，以逐渐理解它们和管理决策分析中可以用到的数据，使数据仓库在使用中不断发展。如果数据仓库建设不完善，在必要时，需要延长数据仓库的试用期，或者返回数据仓库的规划分析、设计实施阶段，通过对数据仓库的再开发来提高用户的满意度。

在数据仓库投入使用之前，必须制定好数据库的备份和恢复措施，以及数据仓库用户的培训计划，并且完成数据仓库运行手册的编制工作。还需制定一个详细的使用计划，以确保数据仓库的使用成功。

6.5.1 数据仓库的用户培训及支持

用户在使用数据仓库以前必须经过必要的培训，因为用户已经习惯了通过业务系统、纸质报表、外部信息摘录等方式来收集决策中所需要的各种信息。数据仓库将是他们第一次面对在一个系统中就可以收集到所有决策信息的系统，在数据仓库使用中必然遇到各种各样的问题。因此对用户的培训与使用的支持是必需的。

1. 用户的培训

在培训中不仅要向用户解释清楚数据仓库的作用与原理，更重要的是用各种例子向他们说明如何使用数据仓库。

在培训过程中需要向用户介绍：数据仓库的基本概念，数据仓库的基本模型，数据仓库中的数据来源，用户如何访问数据仓库，数据仓库中的预定义报表是怎样向用户提供数据仓库中的信息，数据仓库中所有数据分析与挖掘工具的类型，如何利用标准报表

和应用系统帮助用户获得数据仓库信息，用户如何使用数据仓库中的各种工具。

在培训工作中，还需要了解用户的计算机使用水平，有的用户可能对日常的计算机使用方法都不很清楚。此时，仅仅培训他们使用数据仓库的方法是不够的，还需要培训他们的计算机使用技能，然后，才培训他们使用数据仓库的方法。

在培训工作之前必须选择一个比较好的数据仓库应用案例，作为培训教材。通过使用案例向用户演示如何利用数据仓库去解决实际工作中的问题，使用户了解通过数据仓库的应用，使其管理决策工作如何得到改善、决策质量如何得到提高、决策结果对他们的工作具有多么重要的帮助。

2. 对数据仓库用户的支持

对数据仓库用户的支持首先在于对数据仓库应用成功案例的推广。对数据仓库成功应用项目的推广目的在于，让更多的用户熟悉数据仓库的应用。向尚未应用过数据仓库的用户及时推广介绍其他用户的成功应用经验，使新用户能够从成功的案例中获取数据仓库的应用灵感。数据仓库开发管理人员必须牢记数据仓库能否成功应用的关键在于用户的持续应用，因为数据仓库只是一个为用户提供了具有附加开发价值的动态系统，其真正使用价值必须通过用户持续不断的开发应用才能体现出来。

在数据仓库的应用过程中，数据仓库管理人员必须给予用户积极支持，这些支持在不同应用阶段表现形式不一样。在数据仓库应用的初始阶段，用户可能对系统登录一类的简单操作也需要支持。但是随着时间的推移，用户对数据仓库应用越来越熟练以后，需要支持的问题也越来越复杂。用户们可能会对系统的处理结果提出各种各样的异议。此时，所需要的支持可能不是数据仓库技术人员可以解决的，需要技术人员、商业分析人员与用户一起进行讨论，才能给予圆满的解答。也只有此时，才表明数据仓库具有真正的使用价值。

数据仓库管理组在对用户的支持过程中还需要经常观察、了解用户所进行的各种操作，对于一些经常进行的、复杂的操作，要设法建立存储过程，以简化用户的操作，并且经常向用户提供数据仓库中的数据与元数据的信息，及时将发生变化的数据与元数据通知用户。将支持队伍的联系方式与电话号码置于系统的帮助系统中，是一种深受用户欢迎的方法。

6.5.2 数据仓库的使用方式

数据仓库的使用是由数据仓库应用程序及工具来执行的，统称为决策支持工具。用这些工具可以检索、操作和分析数据，进行辅助决策工作。这些工具可以按两种方式使用：验证和发现。在验证模式中，用户做出商业问题的假设后可以通过存取数据仓库中

的数据来验证此假设。用于实现验证模式的工具包括查询工具、报表系统、多维分析工具。在发现模式中，工具试图发现数据中类似于购买方式或不同购买项目间联系的特征，用户预先不知道发现的方式和关系，数据挖掘工具就是发现模式的应用。验证和发现模式的使用通常分为三种方法——信息处理、分析处理和数据挖掘。

1. 信息处理

信息处理支持决策支持的验证模式，包含数据分析和基本的统计分析、查询和服务等技术。存取和处理的数据可以是历史的，也可以是近期的，且按程度进行概括。其结果以报表和图表的形式给出。

2. 分析处理

分析处理也支持决策支持的验证模式，其目的是要按用户的要求提供数据。用于分析处理的数据不论概括形式还是细节形式都是历史数据。

3. 数据挖掘

数据挖掘支持决策支持的发现模式。数据挖掘工具浏览细节性的事务数据，以便发掘隐藏的模式和关系。数据挖掘工具主要用于了解顾客的行为、商业的隐含模式，其结果通常出现在冗长的报告中。

6.5.3 数据仓库使用中的数据刷新

随着对数据仓库使用次数的增多，人们对数据仓库的要求越来越高。当最初的用户积极使用数据仓库后，总会提出更多的需求——从已有资源中得到更多的信息，获取更多的尚未开发的数据源以及外部资源(如文档、电子表格)和行业数据源。

1. 从已有数据资源中获取更多数据

从已有数据资源中增加更多的数据主要包括以下过程：扩充数据仓库的数据模型；将新数据添加到数据仓库中的事实表和维度表中；将数据求精和重构工程活动扩展到新数据中。如果新数据完全由已有的数据仓库元数据定义，那么只使用数据求精与重构过程活动即可，因为已有的事实表可以容纳这些数据。

2. 从单位内部获取新的数据源

增加新的数据源，包括扩充数据仓库的数据模型，是一项更为复杂的任务。必须对原始数据模型进行维度分析，还要定义新的事实表及对已有事实表的扩充，为数据定义

新的维表。通过分析还应确定新数据与数据仓库中已有数据的重叠部分，再经过求精与重构过程将新数据添加到数据仓库中。

3. 获取新的或更多的行业数据源

一般行业数据源是指可以购买的商业信息资源，可以通过客户调查公司提供的数据库了解未来客户的情况、潜在客户的习惯、购买模式和收入情况等。行业数据源为市场活动提供大量必需的环境信息，还可提供有关日常活动、竞争对手、产品调查、客户调查等诸如此类的信息。

合并行业数据源对于本地运作的应用程序来说是相当困难的，这是因为行业资源的数据格式是固定的，可能与数据仓库的格式是不相容的。前面讨论的数据抽取技术非常适合于数据库的驻留数据。行业资源一般提供结构固定的大型文件，在进行抽取和加载数据仓库之前，这些文件必须先放入本地数据库中。

6.5.4 数据仓库的增强

由于数据仓库是根据不同的主题数据库设计的，当数据仓库投入使用后，一方面需要对已经完成的主题数据库根据用户的使用情况加以改进，以保证用户的持续满意度。此时，在数据仓库的使用和管理应用中注意管理与数据仓库有关的元数据，即监视馈送系统带来的数据变化。由于对数据仓库加载的数据来自业务系统中的数据，因此会不断地发生变化。数据仓库管理小组需要监视这些变化，在需要的时候将这些变化集成到数据仓库中。另一方面还需要对未完成的主题数据库进行设计，以完善企业的整体数据仓库。此时，需要重新回到数据仓库开发的第一阶段对新的主题数据库进行设计，进入数据仓库开发的下个螺旋循环阶段。

数据仓库中除了管理当前运作的数据库外，还管理着历史数据。因此，数据仓库在投入使用后，就会带来数据仓库中需要存储数据的不断增长。在数据仓库的不断使用过程中，用户会对数据仓库的使用提出更高的要求。数据仓库的开发工作是逐步完善的过程，数据仓库的开发者还要在数据仓库投入使用后，在用户使用的基础上对数据仓库特征和功能进行快速的评估，不断地理解、考虑用户的新需求，完善系统。随着用户对数据仓库的理解、使用不断加深，需要在以下领域中对数据仓库进行增强、扩充。

1. 数据仓库的局限性

由于在数据仓库的开发中不可能将元数据库建设得十分完善，使具有局限性的元数据库无法系统地、全面地满足商务查询要求。这种局限可能是由于缺少某些数据的概括或聚集所产生的。因此在数据仓库的应用中需要根据用户的实际使用情况，逐步完善数

据仓库元数据库。

2. 缺乏外部数据源

某些商务查询需要与环境因素有关的、来自行业数据源的信息。而这些数据资源不是在数据仓库开发初期认识不足,就是在开发初期一时难以获取。通过用户对数据仓库的应用,可以明了数据仓库究竟需要哪些外部数据资源,这些外部资源可从哪里获取。数据仓库开发人员可以据此增加新的外部数据资源。

3. 数据仓库数据加载性能不能满足要求

在数据仓库的应用中,用户可能发现数据仓库核心构件的性能不满足数据加载的要求,最终用户访问工具从数据仓库加载数据,需要花费很长时间。只有通过用户对数据仓库的具体应用,数据仓库开发者才能真正地认识到哪些数据仓库构件是改善数据仓库使用性能的关键,才能够为数据仓库性能的改善提供必要依据。

4. 数据仓库应用范围的扩大

只有随着数据仓库的实际应用的开展,才能了解究竟有哪些部门希望设置自己的数据集市。数据仓库的开发者才有可能对数据仓库元数据模型的使用范围进行必要的、正确的扩大,使数据仓库的应用范围得到逐步扩大。

5. 数据仓库整体性能的调整

为了增强数据仓库的性能,需要对数据仓库进行调整。开发人员将通过对用户在数据仓库使用中所反映情况的详细分析,找出影响数据仓库性能的来源。这些影响数据仓库性能的问题是来自网络、应用程序、用户、数据库结构中的哪些部分,这样才能为提高数据仓库的性能进行正确的调整。

6. 数据仓库重新规划

随着数据仓库用户的增加或用户对数据仓库应用主题域的扩展,需要对数据仓库重新设计开发。为此,首先要对数据仓库重新规划,规划内容包括对数据仓库的容量,数据仓库的应用范围、数据仓库的用户、数据仓库的数据来源、数据仓库的远程应用和操作等。这些规划的内容与依据必须根据用户在数据仓库的实际应用情况而定。



本章小结

数据仓库的开发应用是一个基于不断循环、逐步增长的生命周期模式，可以将其分成数据仓库的规划分析、设计实施和应用三个阶段，这就是数据仓库的螺旋式的开发。

在数据仓库的规划中，需要确定数据仓库的实现策略、数据仓库的开发目标、数据仓库的体系结构。在数据仓库的需求定义中需要分别对数据仓库的业主、设计者、开发者和最终用户的需求进行定义，才能做好数据仓库的开发。

在数据仓库的设计过程中要完成数据仓库的数据源以及与业务处理系统的接口设计，数据仓库的体系结构与数据仓库数据库的设计，数据仓库的中间件设计，数据仓库的数据抽取设计，数据仓库的数据加载设计，数据仓库的数据复制与发行设计以及数据的测试等设计工作。

数据仓库在设计完成并且加载数据以后，就进入数据仓库的应用阶段。数据仓库的应用并不意味着数据仓库开发的结束，而是数据仓库开发中对数据仓库重新认识的开始。也就是所谓的数据仓库的使用、支持与增强阶段。在这一阶段中所从事工作的重要性并不亚于前面两个阶段。只有在这一阶段取得数据仓库应用的成功经验教训，才能使数据仓库的开发得到螺旋式提升。



习题

- 6-1 为什么说数据仓库的开发是一个不断循环，逐步提升的开发过程？
- 6-2 数据仓库的生命周期应该包含哪几个阶段？需要完成哪些工作？
- 6-3 在数据仓库的需求分析中需要对哪些人员进行需求调查，应该调查哪些内容？
- 6-4 数据仓库的设计包含哪些内容？
- 6-5 怎样通过数据仓库的应用来增强数据仓库的功能与作用？

原书缺页



第 7 章

数据仓库的开发过程

引 言

作为面向决策支持、数据分析的数据仓库生命周期,从规划开始到使用、增强结束,涉及多种任务,开发设计尤其繁重。数据仓库的实质性开发工作如同传统的数据库开发一样,也要经历数据仓库需求分析、数据仓库概念模型设计、数据仓库逻辑模型设计、数据仓库物理模型设计4个阶段。数据仓库的开发与传统数据库开发不同,有其自身的特点。数据库在需求分析阶段就开始了解数据库所具有的功能,而数据仓库则在需求分析阶段不可能了解到数据仓库应该具备什么功能。而且数据仓库的庞大数据存储量远远地超过了数据库的数据容量,对其进行的数据操作要求与数据库也有非常大的不同。因此,数据仓库的开发过程表现出与传统数据库开发不同的特性。

通过本章学习,可以了解:

- ◆数据仓库的概念模型的具体设计方法
- ◆依据概念模型设计数据仓库逻辑模型的方法
- ◆依据逻辑模型完成数据仓库物理模型设计方法
- ◆创建数据仓库以后的各种数据仓库运行管理

7.1 数据仓库的概念模型设计

概念模型是联系主观与客观的桥梁，它是一个为一定的目标设计系统、收集信息而服务的概念性工具。具体到计算机系统设计中，概念模型是客观世界到计算机世界的一个中间层次。人们首先将现实世界抽象为信息世界，然后将信息世界转化为计算机世界。概念模型就是信息世界中的一种架构。因此，概念模型的设计要求创建一种基于对象，代表实际业务的模型。由于概念模型是面向现实的，所以在认识和设计系统时，概念模型易于修改而且适应性很强。概念模型的设计可以分为用户的需求调查、模型定义、模型分析和模型设计几个阶段。

7.1.1 概念模型的需求调查

当用户需要开发一个数据仓库时，往往提出一个数据仓库开发的任务书。在任务书中对组织的背景和组织所在行业的发展进行必要的论述，说明组织目前所要完成的业务功能以及业务范围，并就行业的发展现状，提出组织的战略发展目标。然后，就实现这一发展战略需要数据仓库在决策方面提供哪些支持。例如，某超市连锁企业的数据仓库任务书的内容有：

数据仓库用于支持对存在激烈市场竞争的零售行业分析，数据仓库能向管理部门提供关于客户、客户购买行为，以及国内外零售行业的市场信息。

为完成这一数据仓库的开发任务，数据仓库开发者首先要向有关人员和部门进行调查，描绘关于这一数据仓库以及数据仓库所在环境的完整画面。调查的范围需要从组织中负责数据仓库开发的项目负责人开始，而后扩展到知识用户、信息用户和信息管理人员。通过调查不仅需要描绘数据仓库的整体概括，还要了解数据仓库生存的环境是否得到组织的有力支持；了解以高层管理人员为主的知识用户，以业务管理人员为主的信息用户和信息管理人员是否对数据仓库的开发持认同态度？是否人力支持数据仓库的开发？在调查中还需要进一步了解，哪些主题是数据仓库开发项目中投资收益（ROI）最高的项目。只有选择了一个 ROI 高的数据仓库开发项目，使数据仓库的应用取得实际成效，才能使数据仓库获得广大管理人员的认同。

在数据仓库开发需求调查中一定要注意：不应向被调查人员询问数据仓库应该具有什么功能，而是从管理决策工作中关于数据的需求问题、用户基本情况、用户使用信息的情况、对数据仓库的看法和评价等角度进行调查。

为了成功构建数据仓库，先要明确用户的信息需求，设计者和用户之间需要进行大量的交流与沟通的桥梁。这正是项目负责人所起的作用。一方面，项目负责人代表着用

户向设计者提出对系统的希望和要求,且以这种方式保证数据仓库始终沿着项目所要求的正确方向发展。另一方面,项目负责人又作为设计者的代言人向用户通报数据仓库的进展。项目负责人这样做的前提是他对于数据仓库有充分的了解并且愿意协助设计者成功地完成数据仓库的构建,这些还有助于设计者制定数据仓库用来辅助决策。通过项目负责人,可在设计者和用户之间方便地进行双向交流。为此,项目负责人在数据仓库开发的用户需求认识阶段中的主要任务是协同设计者进行系统定义,界定系统的边界。

在对数据仓库开发项目负责人调查过程中需要了解管理人员在信息需求方面的内容有:哪些事务或业务与任务说明书中的业务需要相关,与这些事务或业务有关的数据保存在哪些系统里?管理人员在进行决策分析时,一般需要多长时间的数据,1年的,2年的,还是5年的?现在组织中使用的业务处理系统是否能够提供这些决策分析数据。与用户有关的调查则需要了解:用户是哪些人,他们应该怎样与数据仓库发生关系?这些用户是否拥有自己的计算机系统,在这些系统中配置了哪些信息处理系统,这些系统的环境怎样?用户在工作中是否使用了数据分析工具,他们在分析工作中经常做哪些方面的分析,是市场的,还是金融的?用户在使用分析报告时喜欢静态文本方式的,还是动态在线方式的?在对数据仓库的评价方面,则要求项目负责人就数据仓库的应用成功因素发表意见,并且提出一些希望数据仓库解决管理中的哪些主要问题?

在对知识用户与信息用户的调查中需要了解关于信息的来源:用户在组织中承担什么工作,在工作中所需要的信息,信息中是否有战略信息,这些信息的来源在哪里?这些信息采用哪些处理方式,在所在的部门中使用哪些信息系统,这些系统提供哪些分析信息,以及提供信息的方式。还要了解关于用户的一些基本情况:用户在使用什么样的计算机系统,系统中有哪些程序,对这些程序应用的熟练程度等;还要了解用户对数据仓库的认识:是否了解数据仓库,数据仓库在管理中能够达到什么目的,用什么标准来衡量是否达到了管理目的,在工作中还需要哪些目前不能获得的信息。在调查知识用户与信息用户对数据仓库的评价中所涉及到的调查内容,与向项目负责人调查一样。

对组织中的信息管理人员调查,则集中在关于组织所使用的系统环境:组织中是否有DSS的应用,这些DSS的用户是谁?用户在日常工作中需要系统提供什么支持,系统可以提供哪些主题数据?系统的文档,尤其是数据字典的文档是否齐全。系统的基本概况怎样,谁负责这些系统。对数据仓库开发的看法:数据仓库构建后,应该达到什么目的,它与目前的系统应该如何相处,在数据仓库的开发中最难解决的问题是什么?对数据仓库的评价调查内容与向项目负责人调查内容一样。

7.1.2 概念模型的定义

完成概念模型的需求调查以后,可以开始进行概念模型的定义。在概念模型的定义

过程中需要确定系统的范围以及所涉及的对象。模型的设计先要明确所需要构建的内容,设计模型的起点是所选择的主题域。数据仓库是面向决策进行分析的数据库,无法在数据仓库设计时就确定用户明确而详细的需求,只有一些基本的需求方向、基本的数据需求摆在设计者的面前:要做的决策有哪些?决策者感兴趣的是什么问题?解决这些问题需要什么样的信息?

作为传统的业务处理系统开发,在其开发分析中需要明确业务处理的具体功能,即系统的开发是基于功能驱动的。数据仓库的开发则是基于数据驱动的,数据仓库开发人员在数据仓库形成与应用之前是不可能了解数据仓库的功能的。因此,无法采用功能驱动开发方法进行数据仓库的开发。但是,数据仓库开发人员在数据仓库开发之前可以通过数据仓库的需求分析,了解数据仓库用户的大致数据需求,即在决策过程中需要什么信息。这样,就可以界定一个数据仓库的大致系统边界,集中精力进行主要部分的开发。因而,界定边界的工作也可看做是数据仓库系统设计的需求分析,因为它将决策者的数据分析的需求用系统边界的定义形式反映出来。

例如,以某个超市的数据仓库设计为例。日趋激烈的市场竞争要求超市经营者更加准确地了解超市经营状况,跟踪市场趋势,更加合理地制定商品的采购与销售策略。由于超市业务处理的需要,已经建立一些分散的数据库,分别处理各自的业务。如在人事、采购、库存、销售等几个部门分别存储着人事、采购、库存、销售的数据库,但是各个部门的数据都是按自己部门的业务处理需要加以组织的。这样的组织使得数据各自为政、缺乏全局性,管理层想要在这些数据库的基础上得到一些全局报表、进行一些分析工作是比较困难的。因此,超市的高层领导决定要在原有的数据库系统基础上建立一个数据仓库。为实现该数据仓库概念模型的定义,首先需要分析用户的决策需求;其次,分析为实现这些决策分析,数据仓库应该提供哪些信息?

1. 数据仓库用户的决策分析

从决定数据仓库的开发初衷来说,超市的管理者最迫切的需求是能更准确地把握超市商品的销售情况和库存情况。

为制定一个较长时期的营销策略,超市经营者目前所要进行的分析有:客户的购买趋势、商品供应市场的变化趋势,供应商和客户的信用等级等情况。

2. 支持决策的数据需求分析

管理决策者完成以上的决策分析,需要商品销售量、商品采购量、商品库存量、客户情况和供应商情况这样一些数据。

3. 数据需求分析工具

为了对数据进行完整的、规范的分析，可以采用用户信息需求表来描述用户的信息需求状况（见表 7-1）。在用户信息需求表中列出概念模型定义中所确定的数据仓库用户决策分析问题以及所需要的信息。在列出所有需要信息的同时，还要明确这些信息的详略程度。例如，对客户购买商品趋势分析时，可能需要根据客户购买商品时所在的国家、省、市、街道、商店进行分析。此时，应将这些不同层次的信息按照层次的高低依次填写在用户信息需求表中，并且在所需要的信息名称后标明这些信息可能分成多少个组别，才能满足决策分析的需要。利用这张表可以为客户购买商品趋势分析的主题确定不同的维：日期、地点、商品等，且可进一步确定维的层次，例如，日期维的层次有：年、季、月。接着，还可进一步确定不同层次中的类，例如，在商品维中的“商品种类”层就有 7 个类。

表 7-1 用户信息需求表

需求信息类	日期	地点	商品	年龄组	经济状况	信用
需求信息 1 层	年 (4)	国家 (15)	商品种类 (7)	年龄组 (8)	经济类 (10)	信用 (10)
需求信息 2 层	季 (16)	省 (60)	商品小类 (40)
需求信息 3 层	月 (48)	市 (200)	商品 (220)			
需求信息 4 层	街道 (2100)			
需求信息 5 层		商店 (20000)				
.....					

信息需求单位:

信息需求采集人:

信息需求表填写时间:

4. CRUD 矩阵

概念模型的定义，不仅需要构建一个 ERD 模型，还要了解 ERD 模型中每一个实体的诞生与消亡事件。因为只有在实体诞生以后，数据仓库才能从数据源中获取关于这一实体的数据。当这个实体消亡后，还需要将该实体的消亡状况在数据仓库的元数据中记录下来。为了提高系统的处理效率，在业务处理系统中常将一些历史数据删除，但是在数据仓库中这些历史数据却需要保留下来。

例如，在销售业务处理系统中，某个客户第一次购买产品，系统会将一些相关信息记录在案。但是某个已经记录在案的客户，如果两年中没有订购产品，就要在业务系统中将其置于停顿状态；如果某个客户三年没有订购产品，就要将其从业务系统中删除。而在数据仓库中，该客户的信息必须长期保留，因为管理人员可能需要了解五年中的客户信息，数据仓库就需要提供销售情况的五年快照。这些快照的信息包含客户的第一次

订购时间、最后一次订购时间，目前的状况。为获取这些信息，在数据仓库的高层模型中就需要使用 CRUD 工具反映实体的生成、引用、更新和删除情况。利用 CRUD 矩阵还可以使数据源与数据仓库的联系得到确认。在 CRUD 矩阵使用中只描述那些重要的数据实体事件，对并不重要的实体可不考虑。

在实体的 CRUD 事件中，最重要的是 CD 事件，因为 CD 事件提供了数据仓库的数据源的数据质量和数据完整信息。同时，CD 事件对数据仓库的时间标识机制会产生较大的影响；而 U 事件对数据仓库的维护具有重要的作用，只有了解了数据源的更新情况，才能确定数据仓库中数据的刷新处理。在使用 CRUD 矩阵进行概念模型设计时，可以了解到数千种潜在的数据仓库应用关系，这些应用将产生大量的实体与功能关系 CRUD 矩阵（见表 7-2）。因此，在应用 CRUD 矩阵时，一定要和用户以及业务系统的使用人员保持紧密的联系，对实体与功能关系的 CRUD 矩阵进行仔细的分析，寻找对数据仓库真正有用的数据源。

表 7-2 实体与功能关系 CRUD 矩阵

功 能	用 户	订 单	产 品	销售线索	销 售 额
订单输入	CRUD	CRUD	R	RU	RU
订单处理		CRUD		CRUD	
产品管理	R	R	RU		R
预算系统	R	R	R	RU	R
财务计算	RU	R	RU	R	R
制造控制	R	RU	CRUD		R
后勤	R	RU	R		RU
生产控制		RU			

C: Create 产生、R: Read 引用、U: Update 更新、D: Delete 删除。

5. 企业业务处理系统数据存储表

数据仓库分析人员在数据仓库的概念模型定义中还要了解组织现行业务处理系统的数据存储方式，从中找到数据仓库的数据映射源的物理状况，这对数据仓库的创建与刷新是十分重要的。因此，需用数据存储模式表（参见表 7-3）将所有的数据源存储模式列出。数据存储表的第一列给出组织现有的各种业务处理系统，其他列为这些业务处理系统中数据的存储模式（打勾者所对应的数据模式）。根据这张表，数据仓库设计人员还需要对每个数据源进行分析：这些数据源存储模式的管理者是否能为数据仓库的建设提供某种程度的支持？客户/服务器之间的联结通过哪种通信协议给予支持？数据源的存储模式使用哪些数据操作语言？在了解这些情况以后，数据仓库设计人员可将数据仓库与特定的业务处理系统中的数据源成功地连接在一起。

在了解组织现有数据源的存储模式时，还要了解现有业务处理系统的数据库是如何管理的，便于为数据仓库与数据源的连接寻找合适的中间件。

表 7-3 现行业务处理系统的数据存储模式表

	Oracle	Sysbase	SQL Server	VFP	其他存储模式
订单输入	√			√	
订单处理	√			√	
产品管理		√			
预算系统					√ (Excel)
财务计算			√		
制造控制			√		
后勤				√	
生产控制			√		
外部数据源					
销售代理商				√	
市场调查公司			√		

7.1.3 概念模型的分析

完成概念模型的定义后，还要进一步考察模型中的用户要求和系统环境。分析数据仓库范围内的主要对象，确定系统的主要主题域以及主要主题域之间的联系。分析阶段将详细检查定义阶段所提出的要求，并且研究任何有可能提供解决方法的环境。数据仓库的设计者通过对用户的访问，得到用户对数据仓库结构以及数据仓库存在环境的要求。将分析结果转变成概念模型，提交给被访问者进行确认，以保证设计者对当前环境的正确性理解。概念模型是进一步分析的基础，它为设计者提供系统的使用情况。

概念模型是设计者与用户交流的工具，它必须简捷明了，使技术人员和非技术人员都能轻松了解模型中所包含的信息。模型中要确定系统所包含的主题域，并且要对每个主题域及其关系要有较明确的描述。

概念模型的设计不是建立一个业务用户及其行为的详细说明，而是交流对业务过程的认识。随着数据仓库开发的深入，该模型将被进一步提炼成详细的逻辑模型和物理模型。

概念模型一般用 E-R 图的形式表示，图中各个对象（实体）间存在着相互的联系。在 E-R 图中，用长方体表示实体，对应于数据仓库中主题，在框内写上主题的名字。椭圆表示主题的属性，并用无向边把主题和属性连接起来。用菱形表示主题之间的联系，菱形框内写上联系的名字。用无向边把菱形分别与有关的主题连接，在无向边旁标上联系的类型。若主题之间的联系也具有属性，则把属性和菱形也用无向边连接上。

仍以超市数据仓库为例。在界定系统的边界后，就需要确定其业务过程中涉及的主要主题域。

根据以上对原有分散的数据库系统的分析，考虑到超市经营者的决策分析要求，在上一步系统边界划分的基础上，这里首先确定超市数据仓库的三个基本主题：销售主题、商品主题和客户主题（参见图 7.1）。

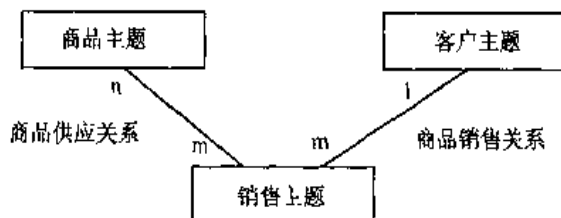


图 7.1 超市数据仓库概念模型

主题之间的联系有：各种商品通过销售与客户发生关系，一种商品可以发生多次销售，每次销售可以包含多种商品。这样，在商品主题与销售主题之间就存在多对多的关系。客户主题可能与销售主题发生这样一些关系：有的客户可能会发生多次销售活动，每个销售活动只针对某个特定的客户。这样在客户与销售主题之间就存在一对多的关系。而客户主题与商品主题间并没有直接的关系，它们之间的联系是经过销售主题产生间接联系。这样三个主题的概念模型就可以用 ERD 表示（参见图 7.2）。

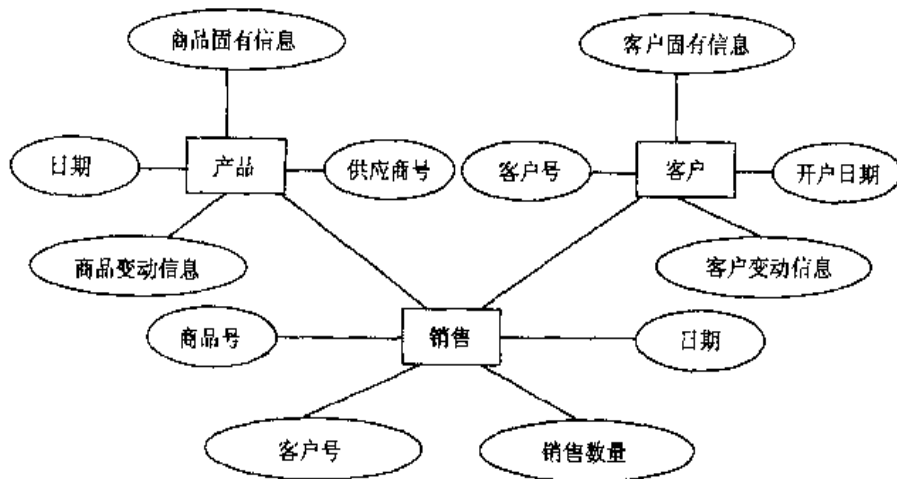


图 7.2 商品、销售和客户的概念模型

7.1.4 概念模型的设计

概念模型的设计是整个概念模型开发过程中的第三阶段。设计阶段依据概念模型分析以及分析过程中所收集的任何数据，完成星型模型和雪花模型的设计。如果仅依赖

ERD, 那只能对商品、销售、客户主题设计成如图 7.2 的概念模型。这种概念模型适合于传统数据库的设计, 但不适合数据仓库的设计。

1. 星型模型的设计

在数据仓库的概念模型设计中, 常常使用星型模型和雪花模型。为设计星型模型, 需要确定概念模型中的指标实体和维度实体。在表 7-1 的用户信息需求表中, 可以确定该用户的主题是商品销售的趋势分析。因此用户的指标实体是销售趋势, 该指标实体应位于星型模型的中心。此外, 从表 7-1 中还可以发现, 用户对销售趋势分析中所需要的信息有销售日期、销售地点、销售商品、客户的年龄、客户的经济状况和客户的信用状况, 这些信息也就构成了星型模型的维实体。因此, 最终可以获得销售主题的星型模型 (见图 7.3)。

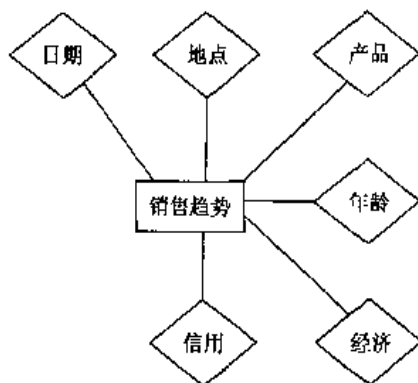


图 7.3 销售主题的星型模型

2. 雪花模型的设计

当构成了星型模型以后, 如果用户希望对相关的维度进行深入的分析, 了解销售趋势所产生的更深入原因。这就需要对星型模型进行修改, 使其更深入地反映销售趋势变化的原因。为此, 就需要设计一个雪花模型。在星型模型的维度实体增加需要进行深入分析的详细类别实体: 商品细节实体和客户细节实体, 产生销售主题的雪花模型 (见图 7.4)。

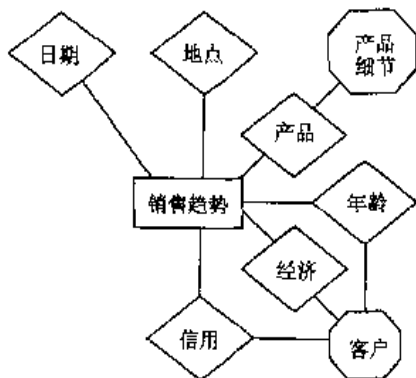


图 7.4 销售主题雪花模型

7.1.5 概念模型文档与评审

完成概念模型设计以后，必须编制数据仓库开发的概念模型文档，且对概念模型进行评价。

1. 概念模型设计文档

概念模型的设计文档主要包含数据仓库开发需求分析报告、概念模型分析报告、概念模型和概念模型的评审报告。

数据仓库开发需求分析报告是概念模型设计的依据，如果需求分析报告得不到用户的认同，概念模型的分析设计就很难令人信服。数据仓库需求分析报告应该以用户的决策问题、决策分析中所需要的信息为主线展开。概念模型分析报告和概念模型是概念模型设计的主要成果。概念模型的评审报告则是对概念模型评审的内容与结论。

2. 概念模型的评审

在数据仓库开发过程中需要经常进行阶段性成果的评审，这种评审对于数据仓库的正确开发是极其重要的。通过经常性的评审，可将数据仓库开发时期的错误及时加以纠正，避免这些错误在完成数据仓库以后才被发现，造成巨大的损失。

在概念模型评审中需要确定概念模型是否完整、准确地描述了用户的决策分析环境。通过概念模型的评审，使得数据仓库开发人员可以找到一个比较理想的数据仓库解决方案，并且能够进一步获得用户的积极支持。

在评审过程中需要由项目负责人就评审的目标向评审人员介绍，确定评审的议程，由专门人员记录评审过程。在评审过程中，还需要对评审问题进行引导，使评审工作能够按照预定方向发展。

如果概念模型评审是数据仓库开发项目确定以来首次进行的评审工作，还需要就数据仓库开发工作中的“用户参与”这一关键问题进行确认。主要确认用户是否已经和项目开发成员之间建立了稳定的联系？用户怎样参与到数据仓库的不同开发阶段中？用户是否乐于接受数据仓库？怎样对用户进行技术支持和培训？开发人员怎样了解用户不断变化的决策支持的信息需求？在数据仓库的开发过程中建立稳固的开发人员与用户的关系是极其重要的。这种关系是否得到建立必须在概念模型评审中得到确认。

3. 概念模型的评审人员

在概念模型评审中需要数据仓库项目负责人，数据仓库分析人员，数据仓库设计人员和数据仓库用户参加。参加评审人员应该控制在 10 人以下，不宜过多。如果参加人员

过多,有的评审人员就不愿意将自己的真实意见表达清楚。在参加评审人员中,用户尤其是主题用户的参加是十分重要的。只有让用户清楚地全面地表达自己对概念模型的看法后,才能使数据仓库的开发模型得到更好的改进。

4. 概念模型的评审内容

在进行概念模型的评审之前,概念模型设计人员必须准备好数据仓库开发任务书、用户决策分析信息需求调查表、数据仓库主题说明书、E-R图、星型模型和雪花模型等概念模型设计成果。

在对概念模型评审之时,要将注意力集中在数据仓库开发任务书是否真实地反映用户开发数据仓库的主要目的,用户决策分析信息需求调查表是否准确全面地描述了用户的决策分析的信息需求,数据仓库主题是否能够全面地包含用户的决策信息,ER图、星型模型和雪花模型是否真实地反映用户决策分析的环境。

7.2 数据仓库的逻辑模型设计

尽管应用星型模型和雪花模型可在概念模型设计中建立数据仓库的概念模型,但是无法直接依靠概念模型实现数据仓库的物理模型,还要依靠逻辑模型作为概念模型到物理模型转换的桥梁。数据仓库的逻辑模型应该与数据仓库物理实现时所使用的数据库有关。由于目前数据仓库一般都建立在关系数据库基础上,因此,数据仓库设计过程中所采用的逻辑模型主要是关系模型。利用关系模型不仅可以创建星型模型与雪花模型中指标实体的关系模式,而且还可创建星型模型与雪花模型中的维度实体和详细类别实体的关系模式。

在进行数据仓库的逻辑模型设计时,一般需要完成分析主题域,确定装载到数据仓库的主题,确定粒度层次划分,确定数据分割策略,关系模式的定义和记录系统定义,确定数据抽取模型等。逻辑模型的最终设计成果应该包含每个主题逻辑定义,且将相关内容记录在数据仓库的元数据中,其中包括粒度划分、数据分割策略、表划分和数据来源等。

7.2.1 分析主题域

在概念模型设计中,已经确定了几个基本的主题域。但是,数据仓库的设计方法是一个循环的过程,在进行数据仓库的设计时一般是一次先建立一个主题或几个主题。所以,在建立数据仓库多个基本主题域时,要对在概念模型设计阶段中确定的多个基本主题域进行分析,从中选择首先需要建立的主题域。

在关于超市的数据仓库的概念模型设计中，首先确定了它的三个基本主题域：“商品”主题域、“销售”主题域和“客户”主题域。进行分析后，可以认为“销售”主题既是一个超市的最基本的业务对象，因为商品的销售等是超市的基本业务，又是进行决策分析的最主要领域，因而把“销售”主题域定义为需要首先建立的主题。通过“销售”主题的建立，超市经营者可对整个超市的经营状况有较全面的了解，尽快地满足超市经营者建立数据仓库的最初要求。

当数据仓库中的主题定义好之后，逻辑模型也就基本构成了。此时，需要在主题的逻辑关系模式中包含所有的属性以及与系统相关的行为。数据仓库中的数据存储结构也需要在逻辑模型的设计阶段完成定义，向里面增加所需要的信息、增加充分代表主题的属性组。以超市数据仓库模型为例，对“商品”、“销售”和“客户”的主题分别增加能够进一步说明主题的属性组，即主题的详细描述（见表 7-4）。

表 7-4 主题的详细描述

主 题 域	公共属性	属性组
商品	商品号	商品固有信息：商品号、商品名、类型、颜色等 商品采购信息：商品号、供应商号、供应价、供应日期、供应量等 商品库存信息：商品号、库房号、库存量、日期等
销售	销售单号	销售单固有信息：销售单号、销售地址等 销售信息：客户号、商品号、销售价、销售量、销售时间等
客户	客户号	客户固有信息：客户号、客户名、性别、年龄、文化程度、住址、电话等 客户经济信息：客户号、月收入、家庭总收入等

7.2.2 粒度层次的划分

在数据仓库的逻辑设计中还要解决的一个重要问题是决定数据仓库粒度的层次划分，粒度层次的划分适当与否直接影响到数据仓库中要存储的数据量和查询方法。具体确定数据仓库的粒度划分的方法，在第 5.6 节中已经具体介绍。通过粒度的划分就决定了在数据仓库中采取的是单一粒度还是多重粒度，以及粒度划分的层次。

例如在超市经营上千种商品，商品的来源也有很多，每日的商品销售数据更多。在超市的业务处理环境中每时都在生成新的数据，进入超市数据仓库“销售”主题的数据量会有很多，因而就需要采取多重粒度进行数据的划分，并且需要充分考虑“销售”主题中各项内容的特点以及对数据分析的要求，细致地进行粒度划分形式的选择，合理地确定粒度划分层次。如：考虑到商品销售记录的数据量很大，且对商品销售的分析主要是进行销售统计以及销售趋势进行分析的特点，因此，商品销售数据的综合层次要丰富

一些,如每种商品的周统计销售数据、月统计销售数据以及季统计销售数据,每小类商品的周统计销售数据、月统计销售数据以及季统计销售数据等。考虑到库存数据不能累加的特点,可以采取样本数据的粒度形式。

7.2.3 确定数据分割策略

数据的分割是指把逻辑上整体的数据分割成较小的、可以独立管理的物理单元进行存储的方法。使用数据分割能够便于数据的重构、重组和恢复,以提高创建索引和顺序扫描的效率。使用数据分割同时也可有效地支持数据概括。这种数据的分割策略定义,必须在逻辑模型的设计过程中完成,这样才能为数据仓库的物理实施提供设计依据。

在上述超市数据仓库建设的例子中,可以采用的分割形式是按时间对数据进行分割,即将在同一时间内的数据组织在一起。如由于超市的管理者经常关心的是商品在某个季节的销售情况,从而将超市的销售数据按季节进行分割,可以大大减少数据检索的范围,从而达到减少物理 I/O 次数,提高系统性能的目的。按照时间进行数据分割,还可以用时点采样形式进行,例如可以按照商品的库存信息进行数据分割。

在超市数据仓库中按时间进行数据分割是可行的。一是超市的数据仓库在获取数据时是按时间顺序进行的,同一时间的数据可以连续获得,因而按时间进行数据分割是简单可行的。二是超市的数据仓库中数据的综合常常在时间维上进行,如要获得某商品某季节的销售总量等等,按时间进行数据分割便于进行统计。还可以按业务类型、地理分布等对数据进行分割。在许多情况下,数据分割采用的标准不是单一的,往往是多个标准的组合。例如,按照季节和业务类型进行数据分割,将同一时间和同一业务的数据合并在一起存储。

在设计数据仓库的数据分割时,最主要的是选择适当的分割标准。选择适当的数据分割标准一般需要考虑以下 3 个方面的因素。

1. 数据量

数据量的大小是决定是否进行数据分割和如何分割的主要因素。如果数据量较小,可以不进行数据分割,或只用单一标准将数据分割。如果数据量较大,就要考虑采用多重标准的组合来较细致地分割数据。

2. 数据分析处理的对象

数据分割是跟数据处理的对象紧密联系的,不同主题内数据分割的标准不同。如“商品”主题内对于数据的分割更多地采用商品大类、商品小类和时间标准,因为人们经常对商品进行分类分析或时间分析。而在“供应商”主题内数据分割的标准则更多地用地

理位置即供应商的地址和时间进行分割。

3. 粒度分割的策略

进行数据分割设计时,更重要的是将数据分割标准与粒度层次的划分策略同一起来。例如,“商品”主题关于商品销售数据的粒度按时间和商品类别来划分,那么,就应该对每一粒度层次上的数据都按时间和商品类别的组合标准进行分割,便于对分割后的数据在时间和类别方面,综合为更高层次粒度的数据。因此,在进行数据仓库的设计时需要把数据分割和粒度划分结合起来考虑。

7.2.4 关系模型定义

不管数据仓库的概念模型是 ER 模型、星型模型还是雪花模型,其最后的物理实现必然是以各种表来完成的。这些表有的是由指标实体转换而成,有的是由维实体而来,有的是从详细类别实体所来。

指标实体在转换成事实表时,往往会形成多个事实表。例如在图 5.10 的金融企业客户主题逻辑模型中,可以包含客户基本情况表、客户变动情况表、贷款情况表、存款情况表、房屋抵押贷款表、汽车抵押贷款表等。这些表之间需要依靠主题间的公共码键——账号联系在一起,形成一个完整的主题域:公共码键:账号,其他的事实表模型定义可见 5.4.1 节的事实表设计。

在构造数据仓库的逻辑模型时,还需要创建有关的维表和详细类别表。事实表必须依靠外键与维表建立联系。

7.2.5 数据仓库的实体定义

在设计逻辑模型时,必须对逻辑模型中的每个实体进行具体的定义。在定义之前必须明确实体究竟是另一个实体的部分还是具有独立性的实体。

在逻辑数据模型中不仅要确定实体、实体之间的关系和实体所具有的列,还要进一步确定实体列中的主键列,实体之间关系的外部键列,实体物理存储的一些特性。

要从实体的众多事实数据中识别用户所需要的数据就需要在不同的列中选择某个可以惟一识别数据表行的列作为主键列,主键列通常由一个列或多个列组成,要求主键列必须识别实体的一个实例。例如, CustomerNumber 可以作为客户实体的主键列,以识别每个客户实例。

在实体的列中除了主键列的确定外,还要确定一些候选键列,例如,客户实体常用 CustomerNumber 作为主键列,但是用户在数据仓库的应用中可能对客户的名称要比客户

的编号更熟悉,因此可以选择 CustomerName 作为客户实体的候选键列,这就为用户的访问提供了便利。

为在数据仓库的物理模型中表示实体之间的联系,必须确定实体的外部键列,实体的外部键列是值存在于某个实体中的某一列或某一组列,它们的值在其他实体中作为主键处理。例如,在订单细节实体(Order_Detail)中客户编号列(Customer_Number)用于描述签订订单的客户,而 Customer_Number 则是客户实体中的主键,此时 Customer_Number 就成为订单细节实体的外部键列,用此键列可以将其与客户实体关联起来。

在实体中对列还需要确定是否可以为空值。一般情况下,作为主键或候选键的列都不能为空值。因为,空值列是不能被系统识别的。

在完成所有实体的物理分析以后,需要列出每个实体所有列的具体特征,即订单细节(Order_Detail)实体特性(见表 7-5)。

表 7-5 订单细节(Order_Detail)实体特性表

列 名	列的键属性	值 范 围	完整性约束	类型与大小
Customer_Number	主键列、外部键列	来自客户实体的合法客户键列	没有客户键列,数据就不存在	Char(10)
Order_Number	主键列、外部键列	来自订单实体的合法键列	没有订单键列,数据就不存在	Char(10)
Product_Number	主键列、外部键列	来自商品实体的合法键列	没有商品键列,数据就不存在	Char(10)
Product_Price		正的金数额		Money(float)
.....

在物理数据模型的设计中还要确定实体的容量与实体数据的更新频率(见表 7-6),作为物理数据库容量需求与数据加载的依据。

表 7-6 实体容量与实体数据的更新频率表

实 体	容 量	更新频率
Customer	中等容量,有 100 个重点客户,2 000 个跟踪客户	每月对客户情况进行一次分析,更新频率也为每月一次
Product	小容量,500 种商品	人约有 500 种商品,商品的更新是每月一次,数据更新也照此
Order_Detail	大容量,其上限是 354 000 000 000,考虑到各种客户类型与各种商品的组合情况,一般很少达到	数据每月汇总一次,但是业务处理系统的数据每日需要更新一次,因此更新频率为每日一次
.....

7.2.6 数据仓库的数据抽取模型

数据仓库的数据抽取模型由数据抽取处理过程、数据源表、数据源抽取过滤条件与

连接表、数据抽取过程的排序与聚集表、数据抽取的目标列与源列对应关系表等组成。

数据仓库的抽取处理是传统的数据处理过程，其输入是数据仓库数据源的各种业务操作处理系统的数据库，输出部分是数据仓库。熟悉数据流程图的数据仓库开发人员，能够很容易地给出数据仓库的数据抽取的数据流程图（见图 7.5）。

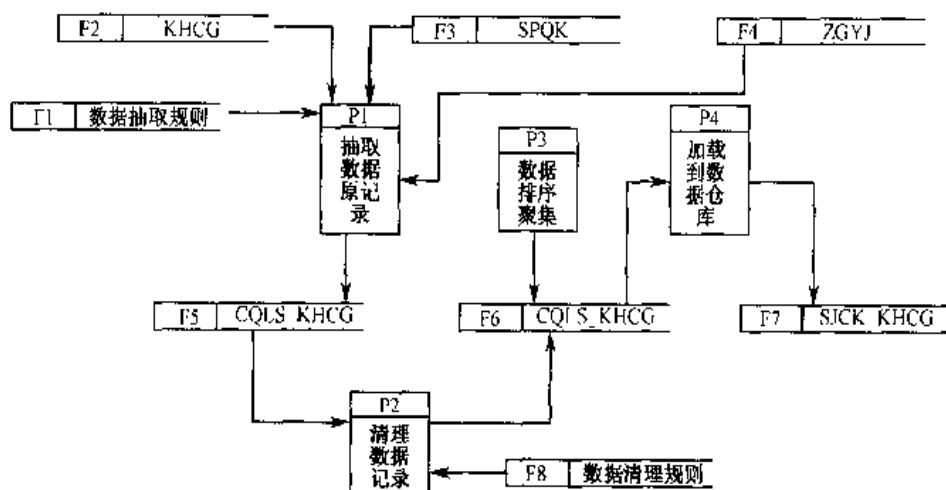


图 7.5 数据仓库数据抽取流程图

为实现数据仓库的正确数据抽取还要利用数据抽取规则确定从哪些数据源中抽取哪些数据，这些数据基于什么样的数据平台，即数据源抽取对象（见表 7-7）。只有分析清楚这些数据源平台以后，才能确定采用适当的数据抽取方式。

表 7-7 数据源抽取对象表

数据源平台	数据源名称	抽取对象	备注
Windows/SQL Server	XSSJ	KHCG	客户采购商品表
Windows/SQL Server	XSSJ	SPQK	商品情况表
Windows/Access	RSGL	ZGYJ	职工业绩表
.....

在数据的抽取分析中还需要分析所抽取的数据应该满足哪些条件，这些条件可能是一些复合条件，而且可能来自不同的表。因此，要在数据源抽取规则表中给出需要根据某种过滤条件与连接条件进行数据抽取的表及其列名和具体的过滤与连接条件（见表 7-8）。

表 7-8 数据源抽取规则表

抽取表的列名	过滤条件	过滤值	连接条件	备注
KHCG.CGSL	<	50000	AND	采购商品数量小于 50000
KHCG.CGSL	>	500	AND	采购商品数量大于 500
SPQK.SPID	≠	'AB'	OR	商品前两位非'AB'
.....

将数据从数据源抽取到数据准备区后,还需要对所抽取的数据进行各种清理工作,这些数据的清理内容必须在逻辑模型设计过程中确定下来。数据的清理内容可能包含数据类型的转换,例如将整型数据转变为实数类型,或将数据的日历格式进行统一,或将数据值中按照数据粒度模型进行汇总、聚集处理,见表7-10中的转换公式。

在数据完成清理工作后,准备加载到数据仓库之前还需要对数据进行排序、分组,以使加快数据的加载工作。因此,在数据仓库的逻辑模型设计过程中需要列出数据排序或分组的标准,即数据抽取过程的排序与聚集情况表(见表7-9)。

表 7-9 数据抽取过程的排序与聚集情况表

表列名	排 序	聚 集	备 注
CQLS_KHCG.CGSL	降序	是	按照采购数量从大到小排序、按照日期进行分组
.....

当完成数据的排序与分组以后,就可以将数据从数据准备区加载到数据仓库中,要完成数据的加载,必须确认哪些数据源加载到哪些目标数据列上(参见表7-10)。

表 7-10 数据抽取的目标列与源列对应关系表

目标列	源列	转换公式	备 注
SJCK_KHCG.KHZY	KHCG.KHZY	直接转换	客户职业
SJCK_KHCG.CGRQ	KHCG.CGRQ	将月年日的日期格式转换成年月日格式	客户采购日期
.....

7.2.7 逻辑模型的评审

在完成逻辑模型的设计以后,应该将逻辑模型设计方案整理成文档,并且组织有关人员逻辑模型进行评审。

逻辑模型的文档内容应该包含主题域分析报告,数据粒度划分模型,数据分割策略,指标实体、维实体与详细类别实体的关系模式和数据抽取模型。

对逻辑模型的评审主要集中在主题域是否可以正确地反映用户的决策分析需求。从用户对概括数据使用的要求,评审数据粒度的划分和数据分割策略是否可以满足用户决策分析的需要。评审从指标实体、维实体和详细类别实体转换而来的各种关系模式是否满足关系第三范式要求,为提高数据仓库的运行效率是否需要对这些关系模式进行反规范化处理。数据的抽取模型是否正确地建立了数据源与数据仓库的对应关系,数据的约束条件和业务规则是否在这些模型中得到了正确的反映。

7.3 数据仓库物理模型的设计

数据仓库的物理模型就是逻辑模型在数据仓库中的实现模式。其中包括逻辑模型中各种实体表的具体化,例如表的数据结构类型、索引策略、数据存放位置以及数据存储分配等等。在进行物理模型设计实现时,所考虑的因素有 I/O 存取时间、空间利用率和维护的代价。

为了确定数据仓库的物理模型,设计人员必须做这样几个方面的工作:先要全面了解所选用的数据库管理系统,特别是存储结构和存取方法;其次,了解数据环境、数据的使用频率、使用方式、数据规模以及响应时间要求等,这些都是对时间和空间效率进行平衡和优化的重要依据;最后,还要了解外部存储设备的特征。只有这样才能在数据的存储需求与外部存储设备条件中获得平衡。

7.3.1 数据仓库设计的规范

由于在数据仓库中包含多种表、列与域等,为保证数据仓库的设计、实施和管理保持稳定,不产生混乱,需要对物理数据模型中的实体、表、列等进行规范化处理。使整个数据仓库的物理数据模型能够保持一致。数据仓库的规范化内容主要有完整清晰的数据定义,合适的数据格式等。

完整清晰的数据定义能使数据仓库开发人员和用户很清晰地了解所定义的数据,在尽可能的情况下采用完整的定义方式,或者使用一些人所众知的缩写方式。例如,客户编号可以使用 CustomerNumber 或 CutNo。对于数据定义的格式必须大小一致,为提高数据定义的可读性,可以采用大小写混合方式。例如, CustomerNumber 可能比 CUSTOMERNUMBER 更容易阅读理解。在使用比较长的字符描述数据定义时,可以采用适当的下划线或连字符来提高数据定义的可读性,例如, CUSTOMER_NUMBER 要比 CUSTOMERNUMBER 的可读性更高。

从数据仓库的开发标准看,实际上不仅包含数据的定义标准,还应该为数据仓库中的每个组件或部件都确定相应的设计标准,这样才能减少数据仓库开发与使用中的混乱。

7.3.2 确定数据结构类型

在数据仓库的结构中,可能包含这样一些数据类型的任意组合:细节数据、概括数据、外部数据、多维数据、数据子集、专门数据缓存、复制数据和存档数据。数据仓库设计人员必须确定符合设计目标的数据结构类型(见图 7.6)。

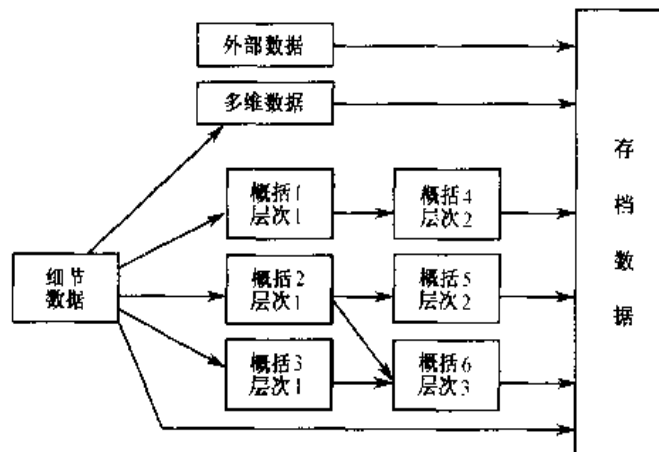


图 7.6 数据仓库的数据结构类型

数据仓库的基础是由最基本的细节数据组成的。细节数据是数据子集和数据概括的基本来源，这些子集和概括都是从细节数据中衍生出来的。数据仓库中的细节数据与操作环境中的细节数据的差异在于：数据仓库中的细节数据只包括决策支持所必需的数据，且应包含时间属性。虽然数据仓库的基础是规范化的数据模型，但在数据仓库中为了提高数据仓库的运行效率，需要进行数据的非正规化处理。在完全规范化的环境中，数据实现的名称和数据格式太多，给数据处理带来过多的表格以及过多的表联结。在数据仓库中进行数据非规范化处理的优点有：

- 能够减少对表联结的需求，提高数据仓库性能；
- 能够减少编写专门决策支持应用程序的必要性，因为运用一些专门的查询工具，可以更容易访问数据；
- 可让用户以直观的易于理解工具查看数据，例如通过电子报表。

从理论上讲，可对细节数据进行任何查询和报表生成程序。由于这对硬件和软件的要求太高，连接大量的报表也更为复杂，所以需要数据仓库中存储的数据进行非规范化处理，按业务处理和查询的要求添加衍生数据和概括数据，以取得较高的性能。例如，将“最后订货日期”和“最后发货日期”等这样的字段也加入数据仓库，可以提高查询效率。所以在数据仓库中，最低级的细节数据应是较为具体的概括数据，细节数据往往只存放一段时间，而概括数据的存储时间较长。这些数据结构类型要在元数据中加以说明。

7.3.3 确定索引策略

数据仓库的数据量很大，要对数据的存取路径进行仔细的设计和选择。由于数据仓库的数据一般很少更新，因而可以设计索引结构来提高数据存取效率。在数据仓库中，

设计人员可以考虑对各个数据存储建立专用的、复杂的索引,以获取较高的存取效率。虽然建立专用的、复杂的索引需要付出一定的代价,但建立后就不需要过多的维护。

数据仓库中的表通常比 OLTP 环境中的表建有更多的索引。表中应用的最大索引数应与表格的规模成正比。数据仓库是个只读的环境,建立索引可以取得灵活性,对提高性能极为有利。但表若有很多索引,那么数据加载时间就会很长。因此,索引的建立需要进行综合的考虑。在建立索引时,可以按照索引使用的频率,由高到低逐步添加;直到某个索引加入后,使数据加载或重组表的时间过长时,就结束索引的添加。

最初,一般都按主关键字和大多数外部关键字建立索引,通常不要添加很多的其他索引。对表建立大量的索引后,在对表具体使用过程中进行分析时,可能需要许多索引,但维护时间也随之增加。如果从主关键字和外部关键字着手建立索引,并且按照需要添加其他索引,就会避免建立大量的索引带来的后果。如果表格过大,而且需要另外增加索引,那么可将表进行分割处理。如果一个表中所有用到的列都在索引文件中,就不必访问事实表,只要访问索引就可达到访问数据的目的,以此减少 I/O 操作。如果表太大,并且经常需要对它进行长时间的扫描,就要考虑添加一张概括表以减少数据的扫描任务。

7.3.4 确定数据存放位置

同一个主题的数据并不要求存放在相同的介质上。在物理设计时,常常要按数据的重要性、使用频率以及对响应时间的要求进行分类,且将不同类型的数据分别存储在不同的存储设备中。重要性高、经常存取、对响应时间要求高的数据存放在高速存储设备上。存取频率低或对存取响应时间要求低的数据则可以放在低速存储设备上。另外,在设计时还要考虑数据在特定存储介质上的布局。在设计数据的布局时要注意遵循以下 5 个原则。

- 不要把经常需要连接的几张表放在同一设备上。
- 把要进行公共连接的表放在同一服务器上。
- 如果几台服务器之间的连接会造成严重的网络业务量的问题,则要考虑服务器复制表格。
- 考虑把整个企业共享的细节数据放在主机或其他集中式服务器上。
- 别把表格和它们的索引放在同一设备上。

在对服务器进行处理时往往要进行大量的等待磁盘数据的工作,则还可以在系统中使用 RAID (Redundant Array of Inexpensive Disk, 廉价冗余磁盘阵列)。这种阵列可以支持数据仓库系统所要进行的大量并行工作,还可以从任何一个磁盘故障中恢复过来。同时,服务器仍保持联机进行对用户透明的处理。数据在分段过程中被分成几部分写进多个磁盘中。当某一个磁盘出现故障时,数据可以通过检查余下的数据得以重建。这种阵

列可以用到以下的4种技术。

- 磁盘镜像：这一技术使用附在同一控制器上的两个驱动器。
- 磁盘复制：其中每个驱动器都有自己的控制器。
- 奇偶性校验：在数据中额外加入一位校验位，以保证该信息的正确传递。
- 磁盘分段：通过某种算法按扇区或字节将数据分布在多个磁盘上。

RAID 技术具有容错能力，能够满足对存储能力、性能和可靠性不断提高的要求。其实现原理是将数据写入多张磁盘中，如果一张磁盘发生故障，就可以从其他存放冗余数据的磁盘上访问数据。RAID 阵列分为如下6个级别实现。

● RAID0：在这一级别上，数据记录在多组驱动器的扇区上交错地分布着，没有奇偶校验，这称为分段，不提供任何冗余。

● RAID1：称为镜像。在这一级别上，数据被冗余地写在成对的驱动器上，可以独立地从每个驱动器上读取数据。这种方法的缺点是，因为它包含数据的完整拷贝，所以要求双倍磁盘容量。RAID1 由于需要双份的磁盘、控制器和电源等设备，其价格较高，目前除用于数据库日志，操作系统软件和数据库引擎的存放以外，已经较少使用。

● RAID2：数据记录在成组的驱动器上位交错，有些驱动器上存储有纠错代码。但是和现在的磁盘驱动器不兼容，目前很少有人使用。

● RAID3：数据记录在成组驱动器上位交错，但只有一个驱动器存有奇偶校验信息。这种方法适用于传送大量信息并且要求很大带宽的应用程序，价格要比 RAID1 便宜，因此有不少人采用。

● RAID4：这一方法需要一个专有的奇偶校验驱动器，数据记录扇区交错地存放在成组的驱动器上。和 RAID3 相比较，所存储的信息不采用字节/比特为单位，而是以块为单位，因此 I/O 传输速率要高于 RAID3。

● RAID5：如果采用这一水平的技术，则数据记录在成组的驱动器上扇区交错地存放着，但所有驱动器都有奇偶校验信息。RAID5 的读写操作只访问所需要的驱动器就可以，而不必像 RAID3 和 RAID4 那样访问一个集合中的所有驱动器，因此 I/O 速度更快。

7.3.5 确定存储分配

在数据仓库的物理模型设计中，需要确定不同数据的存储分配。数据可以集中在一台服务器上，也可以按工作小组部门、主题区或应用程序分散在多个服务器上。按照部门或工作小组进行数据分区时，各个部门数据的数据结构是针对每个部门具体的用户群而定的。虽然存在对部门数据的合理需求，但同时各个部门之间的数据是相互比较、联系的，仍需要将各个部门的数据集市看做整个公司数据仓库体系结构的一个部门进行综合设计。

例如,每个部门或主题区都有一个适当的服务器,要么特定的应用程序也有特别的服务器。在大多数情况下,如果所要访问的数据都是同一部门的,就按部门将数据分散在多台服务器上,以提供最优的性能。在少数情况下,才有必要扫描全公司的数据,才有必要将多个服务器上的数据连接起来。数据可以横向分区,把部门数据安置在部门服务器上,再利用关键字将多个部门级的服务器连接起来,就可以在多个部门服务器上进行公司全局数据查询。

在多台服务器上也可以进行纵向数据分区。利用纵向数据分区,从一张规范表中取出一部分列作为一张独立的表进行存储,对数据仓库的性能也是有利的。当某个单一实体中的数据被两个不同的工作小组使用时,可以考虑利用这些方法。例如,一个部门只对工资数据操作,另一部门只对基本人员的信息操作,可将实体分为两张具有相同关键字的物理表分别存储在不同的服务器上进行操作。为了支持需要两个表全面数据的用户,就将这两张表连接起来。

由用户控制分区,在性能的提高方面是有效的。如果分区后能够只对表的某一分区运行大部分查询,那么就将表设计得小一些,大表的扫描也可以减少一些。在分区的基础上重新组织数据也更加方便。

在为多个地区服务的数据仓库中,数据可以按照地区进行分区。按地理区域进行数据的分区是常见的,尤其是在销售表中。

7.3.6 数据仓库物理模型的评审

作为数据仓库的物理模型评审主要涉及所有的数据定义语言,DBMS 的安装参数,联机过程或批过程的描述,已知的预期数据使用情况,数据量和事务量,预计的数据增长速度以及物理设计文档。物理设计评审的目标是要确定物理模型在满足数据仓库使用的灵活性、性能、数据完整性,系统可用性,数据的当前性和用户的满意度等方面的结果。

具体的评审项目有表空间、分区、表格、数据压缩、控制表和引用表、索引、数据量、数据分布、线路通信量、数据仓库的更新、概况数据、预期变动和数据的文档化。

- 表空间:在物理模型评审时需要大于 1GB 的未分区表空间在加载数据和重组方面的时间进行评价,一个表空间中有多个表时,在表进行连接过程中是否会出现竞争情况。

- 分区:数据仓库一共选择了多少个分区,为什么要选择这些分区,在分区时是否考虑了并行处理,所有的分区是否一样要求。

- 数据表:如果有非规范化的数据表,需要说明可以改进的性能和所付出的代价;数据类型和数据长度是否能够支持未来的用户需求;各列是否允许零值,在这些列中是

否必须将零或空格与数据的缺少分开：是否用同样的数据类型和数据长度规定相应的数据元（主键列/外键列相关的列），如果没有规定相同类型和长度，在连接这些表格时是否会发生问题。

- 数据压缩：数据压缩后能节省多少磁盘空间，读取/更新数据的频率怎样，需要消耗多少 CPU 时间。

- 控制表和引用表：是否会出现瓶颈；除关键字外还有哪些数据？是否规定了引用完整性，对恢复过程有哪些影响，是否制定了恢复过程的规划？

- 索引：是否对所有关键字和大部分外键进行了索引，如果没有，为什么不进行？是否用保证惟一性所必需的最少列规定了主关键字？常用的访问路径是否可由某个索引支持，是否已经了解对每种索引的要求？

- 数据量：预期的数据量有多少？是否会有冗长的加载、转储或重组过程出现？向大表中添加索引是否困难，如果困难，是否可以采用分区或增加概况数据的方法卸载某些数据？是否对比较繁重的特别处理工作转移到资源竞争较少的平台上做了准备。

- 数据分布：细节数据和概括数据分布在哪些平台上，数据将怎样分布（横向分区还是纵向分区，共享提取还是个人提取，或时间戳记提取，复制），是否考虑了连接表对各独立服务器的影响，数据复制是采用同步复制还是异步复制？

- 线路通信量：整个系统产生多少线路通信量，主要是局部地点还是远程地点：远程可重用性是否重要，查询的结果案集有多大，远程传输是否有问题，远程用户的优先权有哪些？

- 数据仓库的更新：怎样对数据仓库进行更新（提取、传播、检查审计）。如果采用数据的传播和复制，那么将实时进行还是延迟进行，这将对源数据产生什么影响？就全部的数据更新量而言，是否存在潜在的瓶颈？谁负责更新过程？怎样处理清洁过程和集成过程？

- 概况数据：如何产生/维护概括表，如何凭借每次加载的数据从头重建表格，如何用新数据修改概况表？如何保证概括数据的完整性？每张表代表的是何时点，这个信息记录在哪里？如果将不正确的数据传送给了概况表，应该怎样纠正？是否会在高峰期间更新概括表。

- 预期变动：为满足用户需求，重组织数据的可能性有多大？数据库设计的灵活性有多大？数据的体系结构是否具有可伸缩性？如果向系统添加大量的用户，是否可将它轻易地转移到一个较大的平台上去？操作系统、最终用户工具和 DBMS 是否要在较大的硬件平台上运作？

- 数据的文档化：所有的用户是否都能得到数据定义和最终用户目录；是否能够得到关于数据所有权、数据来源和数据交换的信息；谁来维护数据的定义并且保证各个平台的一致性，将已经存档的数据编成文档。

如果对以上的评审项目内容都有一个满意的回答,那数据仓库的物理数据库将基本上能够应对数据仓库用户的应用。

7.4 数据仓库的运行技术管理

数据仓库在创建后,通过测试就可以进入使用阶段,在使用阶段中需要不断加强对数据仓库的运行技术管理。这些技术管理工作涉及这样几个方面:数据的加载、复制,数据源的同步化,故障恢复,访问控制与安全性和数据增长管理等。

7.4.1 数据加载的一些问题

1. 数据准备区

由于数据仓库的数据抽取、清理、加载需要较长的工作时间,因此常常设置一个数据准备区的临时数据库,专门用于数据抽取、清理和加载的操作。在数据准备区里可以设置数据抽取、清理和加载的重新启动机制。在数据的抽取清理、加载的过程中,常常由于系统的原因或其他一些不可预计的因素导致这些活动的失败。如果失败以后重新开始,将浪费系统的大量资源。为此,可以设置数据抽取、清理和加载的监控机制,对这些活动进行动态监控。一旦失败,就可以从失败处重新启动,而不必从头开始。例如,系统的数据抽取、清理和加载需要 8 个步骤才能完成,当系统完成了其中的 6 个步骤,进入第 7 个步骤后失败,系统重新启动以后,可在第 7 步重新开始。为完成这一机制,需要能够将数据的抽取、清理和加载活动明确地分成若干步骤,且在进入某个步骤之时,保留当前的状况。因此数据准备区实际上也是一个不小的数据库。

2. 数据加载方式的选择

数据加载的方式一般考虑批处理。因为数据的加载活动涉及的系统资源较多,需要数据源和数据仓库的处理期、内存和外部存储设备。大多数数据源作为业务处理系统,在白天要为用户提供实时服务,因此数据仓库的数据加载往往选择在节假日或夜间进行。这就需要数据加载处理与其他的批处理系统协调好。

3. 大批量数据加载的处理

有的数据源禁止单纯的大容量数据加载,这就需要采用一些特殊的技术处理大量数据的加载。在大量数据加载过程中往往还涉及系统资源的使用限制问题,在大量数据复制过程中往往需要数据源和数据仓库的处理器、网络与内存各方面的支持,而这些宝贵资源在实际应用中往往会遇到很大的限制。

大量数据加载往往导致数据的刷新。对数据仓库的刷新，实际上是不容许的，因为数据的刷新将导致数据仓库中历史数据的丢失。因此，大量数据的加载与刷新活动只能在数据仓库刚建立好后的第一次数据加载的活动中进行，以后的数据加载往往需要采用增量数据加载方法。若要进行增量数据加载，首先要在数据准备区的加载处理中完成一些必要的准备工作。例如，在进行数据加载的 Transact-SQL 中设置一个影子关键列表，其中包含所要抽取数据的关键列与业务处理系统中对应变化信息的记录。可以记录哪些数据发生了变化？如何改变的？这样在数据的抽取过程中可以依据影子列表中的变化情况，采用选择性数据抽取，只抽取那些发生变化的数据，实现数据的增量抽取。

大批量数据加载可以采用数据复制技术实现，数据的复制技术可以保证数据加载过程中的完整性约束，不会受到系统失败等不测因素的影响，并且可以对数据的传送进行优化处理。

7.4.2 故障恢复管理

数据仓库一旦开始运行，来自管理方面和用户方面不断进行存取的压力也会增加。如果没有及时制定故障恢复规划，对数据仓库使用的影响是非常大的。对故障恢复的模拟应该作为开发和运行活动的一部分。仓库一旦建成并开始运行，就很难再关闭或停止服务器的运行，重新安装数据库管理系统。

- 在故障恢复管理中可以采用这样一些步骤：
- 停止包括操作系统(OS)在内的服务器；
- 重新安装和重新配置操作系统；
- 重新标定驱动器；
- 重新安装和重新配置关系数据库系统、监控程序和中间件；
- 对数据重新加载和重新索引。

7.4.3 访问控制与安全管理

安全性以及对数据仓库内部数据访问的管理，同样是非常重要的。随着概括度的增加，数据的价值也不断增加。可以帮助企业进行决策的概括信息，对于市场竞争是很有价值的。控制对数据仓库的访问是一个重要的问题，同时任务也是复杂的，主要由以下 3 种因素造成。

1. 数据仓库应用的公开性与安全的矛盾

数据仓库的建立主要用于公开收集企业的数据。将这些数据用于决策支持，也可帮

助分析者和操作人员改善操作,获取企业战略上的和持久的竞争优势。但是数据仓库的安全性控制则要求限制数据运行的公开程度。这就形成了一对鲜明的矛盾。

2. 用户的不同访问要求

在数据仓库的操作中,用户按照不同的概括度访问数据仓库内的数据。某个用户可从高度概括的数据入手,不断“细剖”详细的数据;而其他用户则可以在另一概括度上进行操作。这样,在安全控制上很难管理每个用户对数据表的访问。

3. 知识发现过程对安全的影响

大多数用户通过“知识发现过程”使用数据仓库。由于用户需要进行深入的探索,安全控制就与这一过程发生矛盾。因为数据仓库不管理业务处理系统中的操作数据,所以安全性的隐患并不是造成数据的破坏,而是泄露企业的秘密和策略。消除这种隐患的方法是在“需要了解”的基础上进行访问。安全性必须对细剖能力进行限制,且对特定的概括数据表和运作的详细内容提供访问控制,并且还要限制对数据源的使用,如创建临时表和即席查询等。

有些隐患的危害较大,一些不怀好意的用户可能会使大量的资源处于停顿状态,从而使数据仓库无法使用。管理无法控制的查询、创建临时表、将资源范围用于用户侧面描述,都能指出这些隐患。因此,访问控制的设计和安全性规划是运行过程中的核心任务。根据客户/服务器应用程序的性质,用单一的控制方法管理安全性是比较困难的。用户标识和口令对于工作站、网络存取、远程登录服务器、远程登录一个或多个数据库来说经常是不相同的。当一个用户离开时,净化程序必须消除对多个系统的访问控制。规划应用程序可以管理对多个访问控制的净化和消除,也有助于管理数据仓库内部的安全性。

7.4.4 数据增长的管理

数据仓库中数据存储量已经从 GB 级发展到 TB 级,甚至 PB 级。因此,数据仓库中存储的数据量远远大于运作数据库的数据量,这是因为数据仓库除了管理当前运作的数据外,还管理历史信息。数据仓库中的数据增长往往会影响到数据仓库的使用效率。

因此需要利用一些通用的商业和管理实践,控制和管理数据量的增加。

1. 概括技术

大量使用概括技术可以明显地减少数据量。当用户把非常详细的信息转化为高度概括的信息时,可以从多方面减少所需的存储量。但是,为了提供细剖详细数据的能力,

必须把数据存储起来，并且在访问时与概括的程度无关。

2. 对细剖数据的控制

控制细剖的程度可以大大减少数据量。尽管直接的反应是“我需要所有的数据”，但最终用户一般可以用比实际需要更少的详细数据来管理他们的任务，还应提供对细剖数据的访问路径，以便满足对低粒度数据的偶然需要。

3. 历史数据的限制

限制必须存储到数据仓库中的历史数据的长度。这些年来商业特征发生了巨大变化，在一定的时间内可以是周期性的或者重复性的。把存储的历史信息限制到上一个商业周期可能比分析具有边缘值的数据更有价值。

4. 数据使用范围的限制

利用能够改变收集数据环境的商业事件知识限制必须管理的数据范围。例如，当两个公司合并时，它们各自的历史数据的价值可以是不同的。

5. 睡眠数据的移出

虽然在数据仓库的应用中应该根据历史情况删除不再使用的详细数据，但是用户往往还会过高估计历史数据保存的年限，有的数据实际上只需要 3 年的数据就可以，但是用户在设计过程中要求保存 5 年的历史数据。有的用户在数据仓库设计中还可能提出实际上对决策分析没有什么价值的详细数据，使这些无用的数据在数据仓库中大量地积存下来。而且在数据仓库设计中为满足所有用户的可能需要存放了大量的详细数据和概括数据，而不少数据在数据仓库中长期无人使用。这些都造成数据仓库中的大量睡眠数据。随着睡眠数据的增加，使用于查询处理的实际可用数据百分比在不断降低；最后导致数据仓库数据的使用效率急剧下降，使用户对日益延长的数据仓库使用时间表示不满，并且造成数据仓库投资的浪费。

解决这个问题的一种办法就是找出并移出查询时很少用到的数据。将这些很少使用的数据移出数据仓库，或减少存储量，可以提高查询处理的效率；或采用邻线存储系统的二级存储模式。邻线存储系统就是一种处于在线和离线之间的存储系统，这种系统虽然不是在线联机状态，但是可以为用户提供一个合理的访问时间。由于它的价格比在线式的存储系统要低廉，因此适合睡眠数据的存储。



本章小结

在数据仓库的开发过程中，首先需要了解用户的信息需求及来源、用户工作中与决策有关的内容以及用户对数据仓库的希望。在概念模型建设过程中还可利用用户信息需求表，列出用户决策信息的维与层次状况，且要利用 CRUD 矩阵来寻找数据仓库中的数据源。

在数据仓库的逻辑模型设计过程中不仅需要分析主题域，还要进行数据粒度层次的划分，并且确定数据的分割策略以及数据仓库的数据抽取模型。

在数据仓库的物理模型设计过程中，需要完成数据仓库的设计规范、数据仓库实体的定义、并且确定数据仓库的索引、数据的具体存放位置和存储分配。

在数据仓库的概念模型设计、逻辑模型设计和物理模型设计完成后，需要分别进行概念模型的评审、逻辑模型的评审和物理模型的评审，以保证下一阶段开发工作的顺利进行。

数据仓库实施以后，还要加强对数据加载、故障恢复、安全控制、数据增长的管理工作，这样才能保证数据仓库的正常运行。



习题

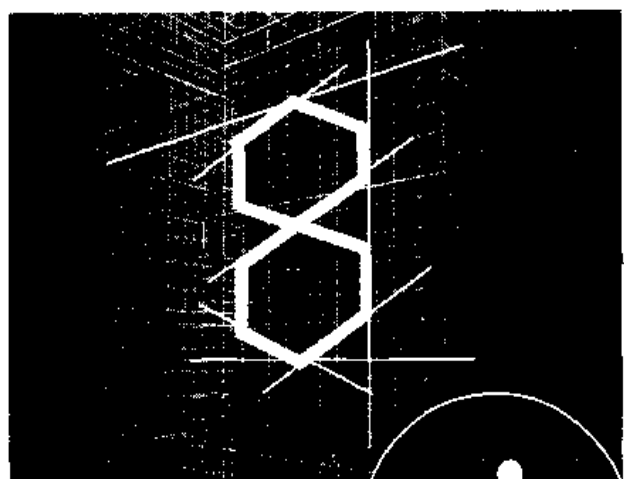
7-1 请依照表 7-1 的购买商品趋势分析表，设计一个数据仓库的星型模型，并且给出维表的层次结构。

7-2 请为航空公司的航班分析数据仓库确定其主题的详细描述。

7-3 为航空公司的航班分析数据仓库设计一种数据抽取和转换方案，并且提出选择此方案的理由。

7-4 请分析不同的 RAIDi 的工作原理和在数据仓库中的应用场合。是否有需要对这些 RAIDi 保护的磁盘进行备份。

7-5 在数据仓库的开发过程中需要对哪些模型进行评审，不同模型的评审内容有哪些？



第 8 章

OLAP 技术

引 言

数据仓库创建以后，企业的经理、主管和商业分析员等知识工人就开始使用各种方法对数据仓库进行操作。其中应用较多的是一些复杂的数据查询，这些查询应该是多角度的、多视图模式的、下钻上卷的、可旋转的。完成这些查询仅拥有大量数据的数据仓库是不够的，知识工人必须依靠一种工具、一种接口，使他们能够轻松自如地在数据仓库的数据海洋中畅游。

通过本章学习，可以了解：

- ◆ OLAP 的基本概念
- ◆ OLAP 的基本结构
- ◆ OLAP 的应用
- ◆ OLAP 的多维分析
- ◆ OLAP 的多维模型与关系模型

8.1 OLAP 技术基本概念

数据仓库是一种管理决策分析的基础。若要有效地利用数据仓库的信息资源,须有强大的工具对数据仓库中的信息进行分析决策。在线分析处理或联机分析处理(OLAP, On-Line Analytical Processing)就是一个得到广泛应用的数据仓库使用技术。

OLAP 专门用于支持复杂的决策分析,支持信息管理和业务管理人员决策活动的一种决策分析工具。它可以根据分析人员的要求,迅速、灵活地对大量数据进行复杂的查询处理,并且以直观的、容易理解的形式将查询结果提供给各种决策人员,使他们迅速、准确地掌握企业的运营情况,了解市场的需求。

OLAP 技术主要有两个特点:一是在线性(On-Line),表现为对用户请求的快速响应和交互式操作,它的实现是由客户机/服务器体系结构完成的;二是多维分析(Multi-Analysis),这也是 OLAP 技术的核心所在。

8.1.1 OLAP 的发展

在 20 世纪的 60 年代末期, E.F.Codd 提出关系数据模型以后,促进了关系数据库与联机事务处理的发展。随着关系数据库的大规模应用,管理人员对数据库中的数据查询要求越来越复杂,查询中所涉及的数据不是一张关系表中的一两条记录,而是涉及多个关系中的成千上万条记录,数据量从早期的兆字节(MB)、千兆字节(GB)发展到兆兆字节(TB)、千兆兆字节(PB),而且在查询中还需要对各种数据进行综合分析处理。为了满足这些要求,许多软件开发商就开发了各种关系型数据库的前端产品。利用专门的数据综合引擎和直观的数据访问界面,以统一复杂查询中各种混乱的应用逻辑,使系统在很短的时间内响应用户的复杂查询。E.F.Codd 在 1993 年将这类技术称为 OLAP。Codd 认为联机事务处理(OLTP)已不能满足终端用户对数据库查询分析的需要,SQL 对大数据库的简单查询也不能满足用户分析的需求。用户的决策分析需要对关系数据库进行大量计算才能得到结果,而简单查询的结果并不能满足决策者提出的需求。因此 Codd 提出了多维数据库和多维分析的概念,即 OLAP。这一类技术也就与 OLTP 有了完全的分。

OLAP 主要针对特定问题的联机数据查询和分析。在查询分析中,系统首先要对原始数据按照用户的观点进行转换处理,使这些数据真正反映用户眼中问题某个真实方面(“维”);然后以各种可能的方式对这些数据进行快速、稳定、一致和交互式的存取,并且允许用户对这些数据按照需要进行深入的观察。

8.1.2 OLAP 的特性

根据 OLAP 产品的实际应用情况和用户对 OLAP 产品的需求,人们提出了对 OLAP 更简单明确的定义,即共享多维信息的快速分析。因此,OLAP 应该具有以下几个方面的特性。

1. 快速性

用户对 OLAP 的快速反应能力有很高的要求。要求系统能在 5 秒钟内对用户的多数分析要求做出反应。如果终端用户在 30 秒内没有得到系统响应就会变得不耐烦,因而可能失去分析主线索,影响分析质量。对于大量的数据分析要达到这个速度并不容易,因此就更需要一些技术上的支持,如专门的数据存储格式、大量的事先运算、特别的硬件设计等。

2. 可分析性

OLAP 系统应能处理与应用有关的任何逻辑分析和统计分析。尽管系统可以事先编程,但并不意味着系统定义了所有的应用。在应用 OLAP 的过程中,用户无需编程就可以定义新的专门计算,将其作为分析的一部分,且以用户所希望的方式给出报告。用户可在 OLAP 平台上进行数据分析,也可连接到其他外部分析工具上,如时间序列分析工具、成本分配工具、意外报警、数据挖掘等。

3. 多维性

多维性是 OLAP 的关键属性。系统能够提供对数据分析的多维视图和分析,包括对层次维和多重层次维的支持。事实上,多维分析是分析企业数据最有效的方法,是 OLAP 的灵魂。

4. 信息性

不论数据量有多大,也不管数据存储在何处,OLAP 系统应能及时获得信息,并且管理大容量信息。这里有许多因素需要考虑,如数据的可复制性、可利用的磁盘空间、OLAP 产品的性能以及与数据仓库的结合度等。

8.2 OLAP 与多维分析

8.2.1 几个基本概念

在 OLAP 中有维、维的层次、维成员、多维数据集、数据单元、多维数据集的度量

值等基本概念，其中维、维的层次概念已在第1章中介绍，这里不再重复。

1. 维成员

维成员是维的一个取值，如果维已经分成了若干个维，那维成员就是不同维层次取值的组合。例如，某公司的销售数据在省、市、县地理维的三个层次，那“江苏省扬州市邗江县”就构成了地理维的一个维成员。维成员并不一定要在维的每一个层次上都取值。例如，“江苏省扬州市”、“扬州市邗江县”、“江苏省”都是地理位置维的维成员。维成员的值并不是人们在数据仓库中所关心的对象，人们常常是用这些维成员去描述他所关心的其他对象——主题在维中的位置。例如，企业的销售管理人员只对销售数据感兴趣，但是在观察销售数据时，却需要以地理位置维、时间维或产品维的维成员值去描述销售数据。例如，对一个销售数据而言，维成员“江苏省扬州市”表示该销售数据是“江苏省扬州市”的销售数据，“江苏省扬州市”是该销售数据在地理位置维上的位置描述。

2. 多维数据集

多维数据集是决策支持的支柱，也是 OLAP 的核心，有时也称为立方体或超立方。OLAP 展现在用户面前的是一幅幅多维视图。多维数据集可以用一个多维数组来表示，例如，经典的时间、地理位置和产品的多维数据集可以表示为：（时间，地理位置，产品，销售数据）。可以看出，在多维数据集中可用（维 1，维 2，……，维 n ，观察变量）的方式进行表达。对于三维数据集可用图 8.1 的可视化方式表达得更清楚，但是在多维结构中并不要观察维度结构，而是观察由维结构所描述的观察变量。也就是说，要在这个三维结构上再添加销售数据，这就得到了一个由三维所对应的销售数据。实际上也就是一个四维结构，当然这种四维结构就很难用可视化方式表达清楚。我们可以用一个四维表的方式来显示那些超三维的多维数据集。例如，由时间、地理位置、产品和促销方式所构成的四维数据集，就可以用表 8-1 的方式来表达。这种超三维的数据集表示方式在许多数据仓库工具中都被采用。

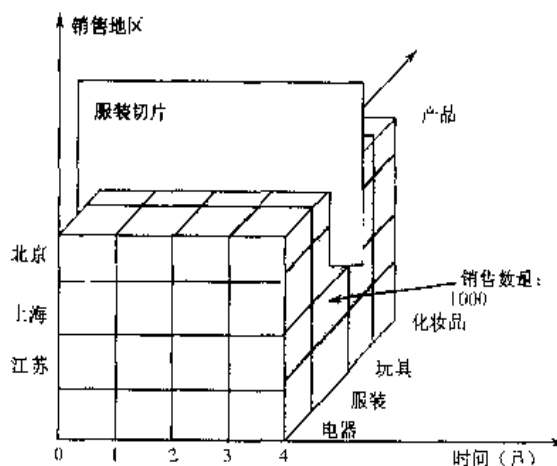


图 8.1 以时间、销售地区、产品三个维度所构成的多维数据集

表 8-1 三维以上的多维数据集

时间 ID					
2002-01-31					
2002-02-28					
2002-03-31					
地理位置 ID					
320112					
320218					
320232					
产品 ID					
A11					
A12					
B11					
B12					
促销方式 ID					
ABC					
BAC					
CAB					
时间 ID	地理位置 ID	产品 ID	促销方式 ID	销售数据	其他数据
2002-01-31	320112	A11	ABC	6484
2002-01-31	320218	A12	BAC	5739
2002-01-31	320232	B11	CAB	5733
2002-01-31	320112	B12	ABC	7945
2002-01-31	320218	A11	BAC	7545
2002-02-28	320232	A12	CAB	7846
2002-02-28	320112	B11	ABC	1237
2002-02-28	320218	B12	BAC	7878
2002-02-28	320232	A11	CAB	8364
2002-03-31	320112	A12	ABC	5488
2002-03-31	320218	B11	BAC	3778
2002-03-31	320232	B12	CAB	7893
2002-03-31	320112	A11	ABC	8884
2002-03-31	320218	B12	BAC	7892

3. 数据单元

多维数据集的取值为数据单元。当在多维数据集中的每个维都选中一个维成员以后，这些维成员的组合就惟一确定了观察变量的值。数据单元也就可以表示为：（维 1 维成员，维 2 维成员，维 3 维成员，维 4 维成员，观察变量值）。例如，在图 8.1 中的时间、销售

地区、产品维度上分别选取了“上海”、“2002 年 4 月”和“服装”，则可以惟一确定观察变量的值（10 000），这样该数据单元应该为（上海，2002 年 4 月，服装，10 000）。

4. 多维数据集的度量值

在多维数据集中有一组度量值，这些值是基于多维数据集中事实表的一列或多列，这些值应该是数字。度量值是多维数据集的核心值，是最终用户在数据仓库应用中需要查看的数据。这些数据一般是销售量、成本和费用等。

8.2.2 多维分析

OLAP 的多维分析是指对多维数据集中的数据用切片、切块、旋转等方式分析数据，使用户从多个角度、多个侧面去观察数据仓库中的数据。这样才能深入地了解数据仓库中数据所蕴涵在后面的信息，才能使用户深入地挖掘隐藏在数据背后的商业模式。

1. 多维的切片

在多维分析过程中，如果对多维数据集的某个维选定一维成员，这种选择操作，就可以称为切片（slice）。也就是说如果有（维 1，维 2，……，维 i ，……，维 n ，观察变量）多维数据集，对维 i 选定了某个维成员，那（维 1，维 2，……，维 i 成员，……，维 n ，观察变量）就是多维数据集（维 1，维 2，……，维 i ，……，维 n ，观察变量）在维 i 上的一个切片。这种切片的数量完全取决于维 i 上的维成员个数，如果维数越多，可以做的切片越多。

很显然，这个切片，不一定是我们想像中的一个二维的“平面”切片。切片的维数取决于原来多维数据集的维数。只有在多维数据集是三维的情况下，才能获得一个二维“平面”切片。例如，对图 8.1 中的多维数据集，如果选定了产品维上的一个维成员“服装”，那就可以得到一个关于服装的、在不同地区、不同时间的二维“平面”切片；从这个切片上，可以更深入地观察“服装”在不同地区、不同时间中的销售情况。

在切片的概念中，有两个重要的概念必须掌握：一是多维数据集的切片数量多少是由所选定的那个维的维成员数量的多寡所决定的，另一个是进行切片操作的目的是使人们能够更好地了解多维数据集，通过切片的操作可以降低多维数据集的维度，使人们能将注意力集中在较少的维度上进行观察。

2. 多维的切块

与切片类似，如果在一个多维数据集上对两个及其以上的维选定维成员的操作可以称为切块。即在（维 1，维 2，……，维 i ，……，维 k ，……，维 n ，观察变量）多维数

据集上, 对维 i , …… , 维 k , 选定了维成员, 那 (维 1, 维 2, …… , 维 i 成员, …… , 维 k 成员, …… , 维 n , 观察变量) 就是多维数据集 (维 1, 维 2, …… , 维 i , …… , 维 k , …… , 维 n , 观察变量) 在维 i , …… , 维 k 上的一个切块。很显然, 在 $i=k$ 时, 切块操作就退化成切片操作。

实际上切块操作也可以看成进行多次切片操作以后, 将每次切片操作所得到的切片重叠在一起而形成的。例如, 通过对图 8.1 中的多维数据集集中的产品维先后按照“服装”和“玩具”进行两次切片操作, 所获得的两个切片可以组成一个在产品维上选取 (服装, 玩具) 的维切块。

3. 旋转

在对数据仓库中的多维数据集进行显示操作过程中, 用户常常希望能将多维数据集改变其显示的维方向, 也就是说进行多维数据集的旋转 (rotate) 操作。这种旋转操作可将多维数据集中的不同维进行交换显示, 使用户更加直观地观察数据集中不同维之间的关系。

4. 其他 OLAP 操作

在 OLAP 的分析中, 还有“钻过” (drill_across) 和“钻透” (drill_through)。前者指对多个事实表进行查询, 后者指对立方体操作时, 利用数据库关系, 钻透立方体的底层, 进入后端的关系表。

OLAP 的其他操作还有统计表中最高值和最低值的项数, 计算平均值、增长率、利润、投资汇报率等统计计算。

8.2.3 维的层次关系

在 OLAP 应用中, 经常涉及对维的层次关系分析。维的层次关系, 可用一个层次图来表示。例如, 在销售地区维上就包含了全国与各省两个层次的简单层次关系 (参见图 8.2), 也可以包含全国、各省、市、县较复杂的层次关系 (参见图 8.3)。

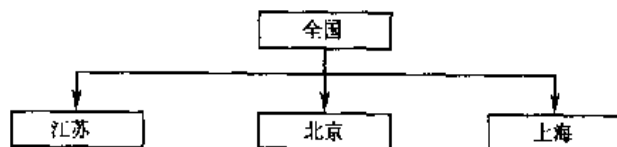


图 8.2 销售地区维的简单层次关系图

这种不同层次关系的出现完全取决于用户的分析应用需要以及对数据组织的详略要求。

有关维的层次信息需要存储在元数据中。这样，OLAP 在进行查询时，可以通过元数据的信息区分不同的维层次，完成用户的查询需求。在维的层次描述中，如果维的层次越高，所对应的综合层次也越高，粒度也就越粗。如果维的层次越多，粒度的层次也越丰富。

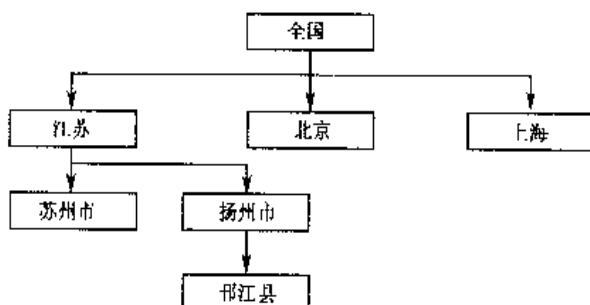


图 8.3 销售地区维的较复杂层次关系图

8.2.4 维的类关系

在 OLAP 的应用中，常常涉及对维成员的分类与归纳，即在查询中根据用户关于类别的要求对所有维成员进行分类，在分类的基础上归纳出类的共同特征或区别于其他类的特征。

应该注意维的层次与类是两种完全不同的概念，两者不仅含义不同，而且操作方式也不同。从其含义看，维层次表达的是维成员的不同综合层次，而类则表达具有某种同一特征的维成员子集。类的划分，只能依据同一层次的维成员集合来划分。例如，在产品的层次中，可以将“西服”、“羽绒服”、“皮衣”划分为服装类。而不能将不具有相同特征的维成员划分在同一类中，也不能将不是同一层次的维成员划分在同一类中。

在 OLAP 的应用中，有的需要按照维的层次关系进行分析，有的需要按照维成员的种类进行分析。这两种分析的操作是不同的。维层次的分析，主要由从高层维到低层维的“钻取”分析和由低层维到高层维的“汇总”分析。“钻取”分析是一个从综合性数据开始逐层向下寻找细节数据的过程，这种分析经常应用。人们在先看到高层的综合数据后，往往会问“为什么会这样？”，那就需要向细节寻找原因，就要用“钻取”操作逐层向下寻找答案。而维成员的分类归纳分析在用户中也是普遍采用的，在市场营销活动中，管理人员常常需要对某一方面具有共同特征的客户采用相同的推销手段，此时，就需要对数据仓库中的数据进行分类分析。

在 OLAP 的实际应用中，常在维的层次关系上进行分析的同时，又在维成员的类关系上进行分析。此时，所涉及到的多维数据集就是一个复杂的数据关系，如图 8.4 所示的维的层次与类组合图。

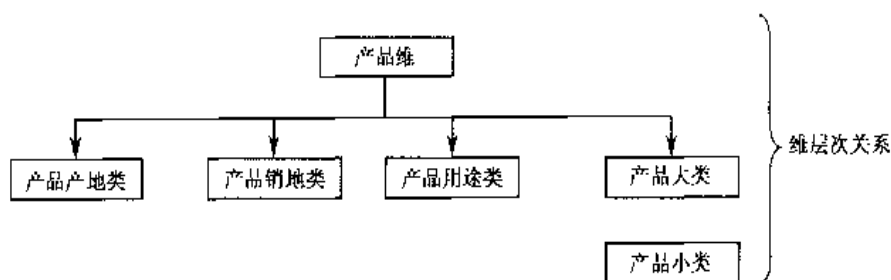


图 8.4 维的层次与类组合图

8.2.5 OLAP 与数据仓库关系

在数据仓库中，OLAP 和数据仓库是密不可分的，但是两者具有不同的概念。数据仓库是一个包含企业历史数据的大规模数据库，这些历史数据主要用于对企业的经营决策提供分析和支持。数据仓库中的数据是不能用于联机事务处理系统（OLTP）的，而 OLAP 技术则利用数据仓库中的数据进行联机分析，将复杂的分析查询结果快速地返回用户。OLAP 利用多维数据集和数据聚集技术对数据仓库中的数据进行组织和汇总，用联机分析和可视化工具对这些数据迅速进行评价。从图 8.5 中可以发现 OLAP 用多维结构表示数据仓库中的数据，创建组织和汇总数据的立方体，这样才能有效地提高用户复杂查询的要求。因此数据仓库的结构将直接影响立方体的设计和构造，也就影响 OLAP 的工作效率。从 OLAP 使用的效率角度考虑，在设计数据仓库时应该考虑这样一些因素：

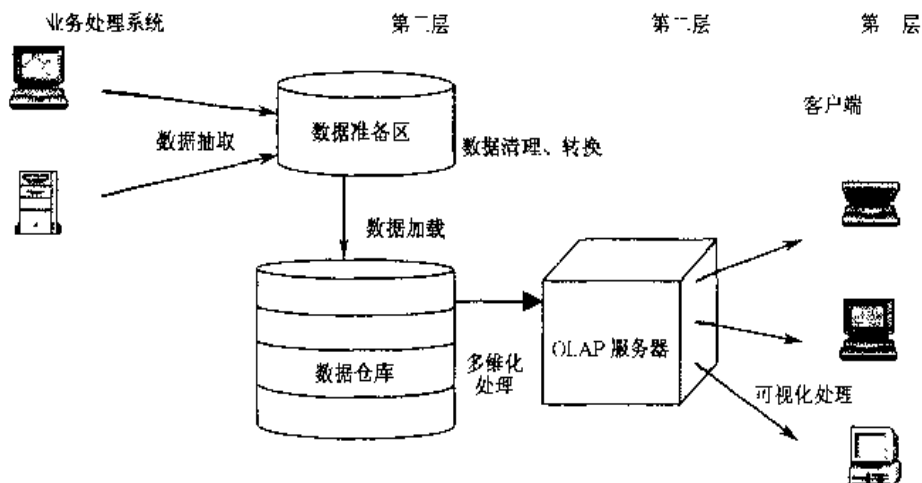


图 8.5 数据仓库与 OLAP 关系图

- 尽可能使用星型架构，如果采用雪花结构，就要最小化事实表底层维度表以后的维度表数量；
- 为用户设计包含事实表的维度表，这些维度表应该包含有意义的、用户希望了解

的信息:

- 维度表的设计应该符合通常意义上的范式约束, 维度表中不要出现无关的数据;
- 事实表中不要包含汇总数据, 事实表中所包含的用户需要访问的数据应该具有必需的粒度, 这些数据应该是同一层次的数据;
- 对事实表和维度表中的关键字必须创建索引, 同一种数据尽可能使用一个事实表;
- 保证数据的参考完整性, 使事实表中的所有数据都出现在所有的维度表中, 避免事实表中的某些数据行在立方体进行聚集运算时没有参加进来。

8.3 OLAP 的实施

OLAP 是介于客户与数据仓库之间的数据分析处理系统, 它需要对来自数据仓库的数据进行多维处理和分析, 因此在系统的构造中常常采用如图 8.5 所示的三层客户/服务器结构。

这种三层客户/服务器的结构通常将数据仓库、OLAP 服务器与客户端严格区分。系统的主要处理, 例如数据存取、后台数据处理、报表的预处理都由 OLAP 服务器上的应用完成, 而不是由客户端完成的。若要实现这个系统, 必须解决如何组织 OLAP 所用的数据, 即 OLAP 服务器如何设计, 如何从数据仓库或数据集市获取数据。还要解决如何与客户端的客户需求进行沟通, 即如何根据客户的需要对多维数据集进行分析, 且将分析的结果以可视化的方式传递给客户端。

OLAP 的处理基础是数据仓库, 在数据仓库中包含大量的业务处理系统的操作细节数据和其他的综合数据。OLAP 的作用主要进行管理分析与决策。在管理分析与决策中, 人们所关心的大多是综合性数据, 需要从综合性的、总的范围来观察数据。因此, 在 OLAP 设计过程中先要解决怎样组织数据仓库中的综合数据, 以满足客户端用户的多维数据分析的需要。

目前, 应用比较广泛的一些 OLAP 系统几乎都采用了三层结构。但是系统在具体实现时, 如果将多维数据存储于客户端和/或 OLAP 服务器, 就可能产生“胖”客户端系统。这种系统由于客户在进行在线分析处理时, 需要将数据加载到客户端, 容易产生网络“瓶颈”。因此, 在客户端较多的情况下就需要采用“瘦”客户端结构来实现 OLAP 系统。

“瘦”客户端系统中的多维数据集不存储在客户端, 而是存储在 OLAP 服务器中。这样在网络中所需要传输的只是经过分析处理以后的结果, 而不是多维数据集。在多维数据集存储量很大、用户较多的情况下还可以在数据仓库与 OLAP 服务器之间再增加一个服务器与存储设备, 专门用于多维数据集的存储与处理。

OLAP 系统在进行数据组织处理时, 可以采用建立专门的多维数据库系统, 或者利

用现在应用比较普遍的关系数据库技术来模拟多维数据集，这样在 OLAP 的实现中，就产生基于多维数据库的 OLAP 系统与基于关系数据库的 OLAP 系统。

8.4 基于多维的 OLAP

OLAP 系统在具体实现时，需要解决采用多维数据库系统还是采用关系数据库系统存储数据的问题。如果采用多维数据库(MDDB-Multi Dimensional DataBase)存储、显示数据，那么这种 OLAP 系统就是基于多维的 OLAP，即 MOLAP。

8.4.1 多维数据库

多维数据库可在 OLAP 系统中直观地表达现实世界中的多对多关系。例如，要在系统中存放两个产品（电器、服装）与不同地区（江苏、上海、北京）的销售情况，用关系数据库来存储这些数据（见表 8-2）和用多维数据库来存储这些数据（见表 8-3）所得到的结果是不同的。由于关系数据库采用关系表来表达某种产品在某一地区的销售情况，而多维数据库则采用二维表格的方式表达这些数据的关系。这就使二维表格比关系表达式所表达的关系更加清晰明了，而且所消耗的存储容量更少。

表 8-2 关系数据库存储数据的方式

产品名称	销售地区	销售数量
电器	江苏	940
电器	上海	450
电器	北京	340
服装	江苏	830
服装	上海	350
服装	北京	270

表 8-3 多维数据库存储数据的方式

	江 苏	上 海	北 京
电器	940	450	340
服装	830	350	270

在关系数据库中对这些数据进行单项查询时，比较容易处理。例如，查询上海地区所销售的电器数量只需要进行一个简单的检索就可以了。如果查询电器的销售总量，那就比较麻烦了，需要对关系数据库的所有记录进行查询，且对销售数量进行汇总，此时系统的效率必然降低。而多维数据库则只需要对库按行或按列进行统计就可，其性能远远优于关系数据库。

在 OLAP 中, 为给用户提供一个一致的系统查询响应时间, 常将查询经常用到的综合数据预先统计汇总, 存储在数据库中, 以加快查询的响应时间。为了达到这个目的, 在关系数据库中需要增加一行汇总数据 (参见表 8-4)。由于关系数据库将需要进行汇总的数据均在事先完成了汇总工作, 在进行查询时就不必再进行求和汇总, 只要从表中读取单个记录, 就可完成求和查询。这样的数据处理显然可以获取快速的响应时间。但是在数据仓库中, 如果历史数据庞大, 这种事先的求和汇总也需要较长的计算时间。更加糟糕的是, 在产品列和销售地区列中出现的“汇总”数据项完全破坏了列的定义。用户在查询过程中必须了解这种例外情况的出现。

表 8-4 具有汇总数据项的关系数据库

产品种类	销售地区	销售数量
电器	江苏	940
电器	上海	450
电器	北京	340
电器	汇总	1730
服装	江苏	830
服装	上海	350
服装	北京	270
服装	汇总	1450
汇总	江苏	1770
汇总	上海	800
汇总	北京	610
汇总	汇总	3180

多维数据库 MDDB 在 OLAP 系统中的优势, 表现在查询速度快和结构清晰明了。在 MDDB 中, 数据可以按照行或列进行累加。在 MDDB 中没有重复出现的信息, 因此, 其统计速度远快于关系型数据库。如果将汇总等数据也存储在数据库中, 只要在原数据库中增加一行、一列就可以 (参见表 8-5), 实现较为简单。

表 8-5 具有汇总值的多维数据库

	江苏	上海	北京	汇总
电器	940	450	340	1730
服装	830	350	270	1450
汇总	1770	800	610	3180

8.4.2 多维数据库的数据存储

在多维数据库中二维数据容易理解, 但是当维数扩展到三维或更高的维度时, 多维

数据库 MDDB 就成了一种“超立方”体的结构,对其理解就产生了困难。但是,在 MDDB 中,其数据的存储是由许多类似于数组的对象来完成的。在这些对象中包含经过高度压缩的索引和指针,利用这些索引和指针将许多存储数据的单元块联结在一起。每个单元块都按照多维数组的方式存储,相互之间通过直接偏移计算进行存取。在索引中只用一个较小的数来标识单元块,因此多维数据库的索引比较小,只占用数据空间的一小部分,可以全部存放在内存中。但是在多维的实际分析中,可能需要将任一维与其他维进行组合,因此需要“旋转”数据立方体已经切片的视图,即用多维方式显示数据。

在 MDDB 中,并非维之间的任何组合都会产生实际的值。在实际组合中往往由于各种原因导致某些组合没有具体的值,或值是空的或者为零。例如,在表 8-5 中,如果该公司在北京地区没有进行电器的销售活动,那在电器行和北京列所交界的单元格的值就是 0,而不是 340。这就产生了多维数据库的稀疏矩阵问题,稀疏矩阵使数据库中产生大量的无数据空间,导致存储空间的浪费。为此,多维数据库常要采用压缩技术来解决空间的浪费问题。

8.4.3 多维数据库与数据仓库

多维数据库为终端用户提供一种可对数据进行灵活访问的信息结构,利用多维数据库可以对数据进行切片、切块,动态地观察汇总数据与细节数据的关系。数据仓库中的细节数据则为多维数据库提供非常健全和便捷的数据源。由于 OLAP 的应用,需要多维数据库定期刷新,数据需要定期地从数据仓库中导入多维数据库中。由于业务处理系统中的数据在导入数据仓库时就被集成了,多维数据库不必再从业务处理系统中抽取与集成数据。基于多维 OLAP 的用户如果对细节数据的分析感兴趣,还可通过数据仓库所保留的细节数据进行分析。

在实际应用中,数据仓库与多维数据库是有差别的。从所存储的数据量看,数据仓库存储了大量的数据,而多维数据库只存储某些类型用户所需要的集成数据,在数据量上要远低于数据仓库;数据仓库只允许少量的分析人员进行少量的灵活访问,而多维数据库允许众多的用户进行大量的非预知的数据访问和分析;从数据存储的时间范围看,数据仓库所存储的数据可能长达 5~10 年,而多维数据库中数据则只保存大约 1 年左右的时间。

多维数据库实际上是与 OLAP 的应用共存的,两者构成基于多维的 OLAP。但是 OLAP 仅是一种技术,而数据仓库是一个体系结构的基础,两者是一种互补和共生的关系。在 OLAP 的实际应用中,有不少人希望直接从业务处理系统中抽取数据,也就是在图 8.5 中将其中的第三层去掉,将第二层的 OLAP 服务器直接与业务处理系统联结。这样,在 OLAP 的应用中,系统设计就十分简便,对于设计人员而言十分诱人。因为这种体系的设计直

截了当，容易实现。但是这种 OLAP 的体系结构在实际应用中有一些很严重的问题。

1. 增加数据抽取部分的工作量

在 OLAP 的应用中，对数据所进行的抽取工作量是很大的：由于不同部门的业务差别，每个部门都要开发一套适合本部门多维数据库的数据抽取、清理和转换程序。这就势必造成这些数据抽取程序的大量重复，使开发量增加许多。而用数据仓库结构则只需要一套数据抽取、清理和转换程序就可以。

2. 缺乏统一的数据源和结论

如果多维数据库从传统业务处理系统中抽取数据时，没有一个统一的数据集成环境，每个部门的多维数据库按照本部门的数据集成方法进行数据抽取，其后果是不能形成一个统一的集成的数据源，而这在数据仓库中是很容易做到的。所带来的后果是在企业的重大问题决策时，由于数据源的不统一，各部门依据本部门多维数据库所进行的 OLAP 处理结果将有很大的差异，使决策问题很难有一个正确的判断。

3. 加大系统的维护工作量

OLAP 系统开发成功以后，需要经常对其进行维护，才能保持 OLAP 的生存周期。业务处理系统的任一变化，必然影响 OLAP 中的数据抽取程序部分。这种改变将涉及所有部门的 OLAP 系统。在数据仓库中只需要进行很少的变动，就可应付业务处理系统的变化所带来的麻烦，大大减少了 OLAP 在实际应用中的工作量。

4. 缺乏对元数据的有效管理

在基于数据仓库的 OLAP 体系中，可对元数据进行有效的管理。在直接从业务处理系统中抽取数据的 OLAP 系统中，由于将数据导入多维数据库的复杂性，使元数据的管理和控制遭到破坏。

5. 加大 OLAP 系统的开发投入

由于在各个部门的 OLAP 应用中都需要对业务处理系统进行数据吸取，导致大量的、重复的数据传输。在数据仓库构架的 OLAP 系统中，这种数据的传输工作只需定期地进行一次就可以了，大大降低了对硬件系统的投入要求。

8.5 关系 OLAP

如果在 OLAP 的实现中采用关系型数据库 (RDBMS)，这种 OLAP 就是基于关系的 OLAP，即 ROLAP。

8.5.1 ROLAP 的三个规则

对于 ROLAP 来说, 应该遵照这样三个规则: 支持 OLAP 原则, 数据存储在某一个关系型数据库中, 支持某种形式的聚集导航。

1. 支持 OLAP 原则

ROLAP 尽管是将数据存储的关系数据库中的, 但是它的基本功能依然是在线分析处理。因此, 应该和任何 OLAP 一样支持数据的多维特性, 能够对数据进行切片、切块、旋转, 并且进行可视化显示。

2. 数据存储在某一个关系型数据库中

ROLAP 的数据自然应该存放在关系型数据库中。问题是这些存储在关系数据库中的数据如果在存储到数据库之前, 为适应 OLAP 的应用需要, 进行了某种方式的处理。那么其他程序在应用这个关系数据库中的数据时, 应该有某种方法对处理的数据进行反处理, 以保证其他程序的应用。

3. 支持某种形式的聚集导航

聚集导航是一种为用户的不同查询, 选择一个最小可用表的软件。因为在数据仓库中除了基本事实表以外, 还包含一些概要表。创建这些概要表的目的是为用户的查询提供便捷的方式。假设在某个数据仓库中包含大约 8 亿条的销售信息, 这些信息涉及客户、产品、产品的销售时间 (具体的年、月、日), 但是用户在对这些信息查询时, 只需要关于客户和产品的销售年份数据。为满足用户的这种需要, 可在数据仓库中构建一个概要表。概要表所包含的信息可能只涉及 200 万条记录, 显然这样处理, 大大地提高了数据仓库的工作效率。

虽然在概要表的支持下, 提高了数据仓库的效率。但是, 用户又怎么了解到有这种概要表的存在? 又怎么知道在哪些情况下, 应该使用这种概要表? 在哪些情况下不需要使用这些概要表? 聚集导航器就可以为用户的不同查询提供寻找最小可用概要表。当然, 聚集导航器对概要表的了解要么是由数据仓库管理员通知每个概要表的大小, 要么是由聚集导航器利用自身所具备的例程定期检查每个表的大小。

8.5.2 ROLAP 的多维表示方法

1. 星型模式在关系数据库中的表示

多维数据库在关系数据库中表示时, 需要分成两大类型: 一类是用于存储事实度量

值与各个维主码的事实表；另一类是维表，在维表中至少保存描述该维的层次关系、成员的类别等元数据。在最简单的情况下，只用一行列出维表的所有合法值。利用维表可在事实中衍生出维的列

事实表通过每个维的主码值与维表联系在一起，构成如图 8.6 所示的星型模式的关系数据库。在图中心的销售情况表是一个事实表，在表中存储了产品 ID、销售商 ID、地址 ID 和时间 ID 四个维表的主码。通过这四个主码将四个维表与事实表联系在一起，构成星型模式。也就是说，应用二维关系表实现了多维数据模式。构成这种星型模式后，就可以在关系数据库中模拟 OLAP 中的多维查询，并且通过维表的主码，对事实表和维表进行连接操作。在一次查询操作中，可以获得查询对象的事实值以及对数据的多维描述（对应维上的维成员），并且在这种 ROLAP 模式中，用户和分析人员可以应用存储在维表中的用户习惯描述（元数据）来说明一个查询需求，而这种需求可被 ROLAP 依靠维表转换成维的代码或值，完成用户的最终需求请求。

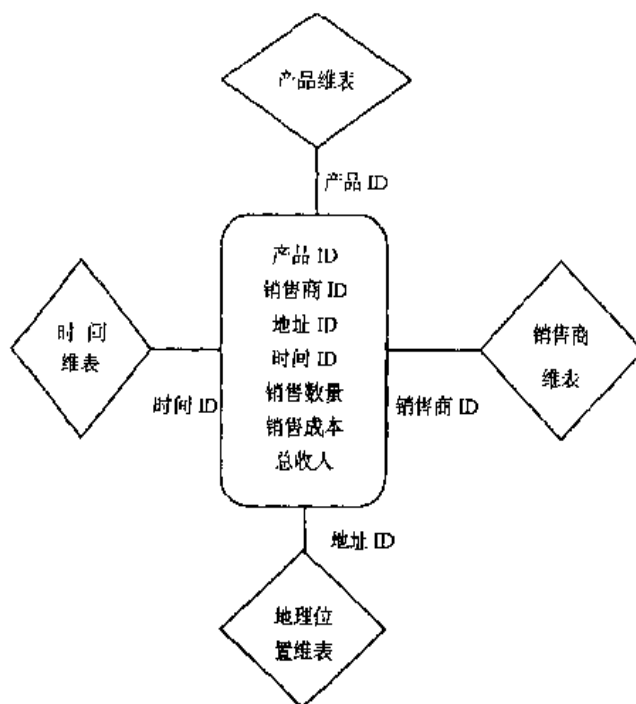


图 8.6 星型模式的关系数据库表示

2. 雪花模式在关系数据库中的表示

由于在管理决策中，管理人员了解所决策对象时，常要多角度、多层次地观察决策对象，也就是说在 OLAP 中需要多层次观察维度。由于现实中用户在使用 OLAP 时，常常提出包含维层次和维类别的复杂分析要求。对这种复杂的维关系，如果仅用一个维表

来描述，必然产生大量的冗余数据。为了解决由于数据的冗余而造成的存储空间浪费，可用多张维表来描述复杂的维关系。例如，在产品维上划分产地类、销地类、用途类、产品类别等若干类，这样在星型模式的角上就出现了分支，也就是说星型模式变成了“雪花”模式。在图 8.7 中所示的雪花模式的关系数据库就是在图 8.6 的星型模式的产品维度上演变而来的。

在 ROLAP 的实现中，常常需要根据维表的复杂程度选用合适的模式。对一些维层次复杂、成员类型多的就可以采用多张表描述，而对一些简单的维可用一张表来描述。由于 ROLAP 中的事实表和维表都要使用二维关系表存放，在多维数据集的构造中，必须通过维表和事实表的联结来实现。如果在 ROLAP 中每个维都需要通过一次联结操作，这就给 ROLAP 带来了一个严重的性能问题。特别是在维数增加、事实表加入时，ROLAP 的处理时间将使用户无法容忍。因此在 ROLAP 产品中，常常采用各种索引技术来提高系统的性能。

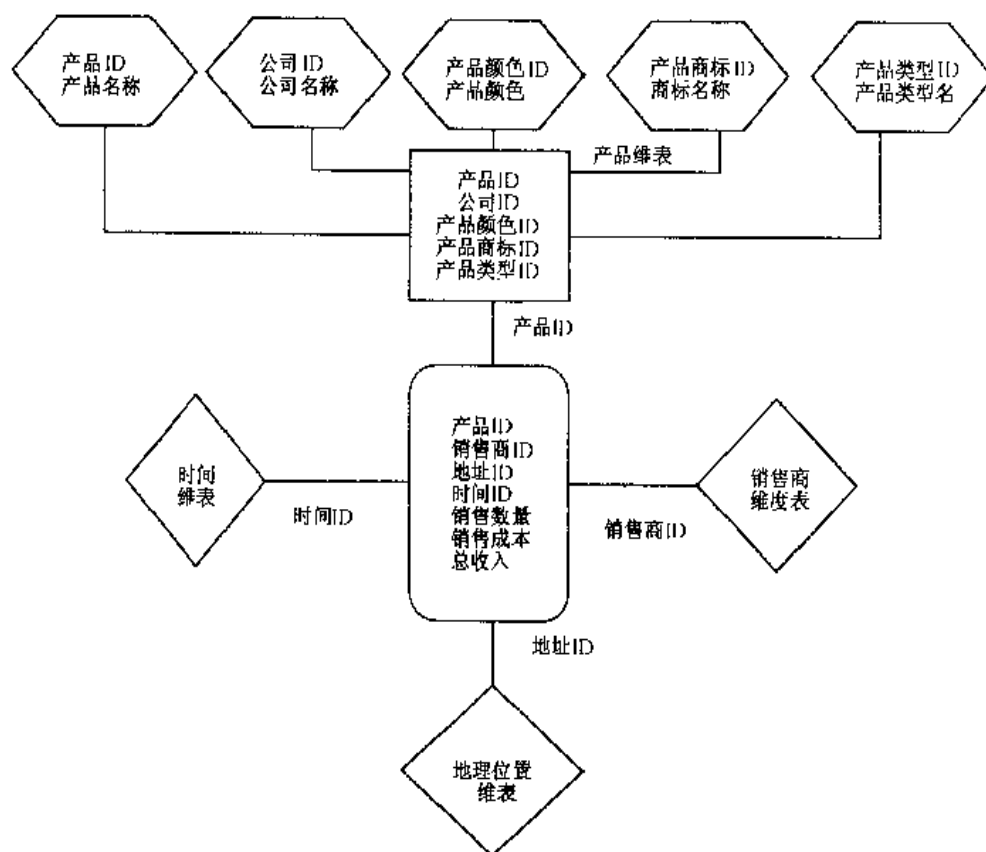


图 8.7 雪花模式的关系数据库表示

8.6 OLAP 的选择与评价标准

8.6.1 MOLAP 与 ROLAP 的比较

在应用 OLAP 时,人们遇到了究竟采用 MOLAP 还是采用 ROLAP 的问题。为了解决这一问题,首先要对这两种模式进行对比分析。以便在 OLAP 设计中,根据这两种模式的特点进行选择。为了衡量这两种模式的特点,通常需要对 MOLAP 与 ROLAP 的查询性能、数据加载性能、分析能力、数据集市的大小、维的管理和维护能力进行比较。

1. 查询性能

MOLAP 的查询能力一般较好,因为在多维数据库中常常根据用户的需求,事先做好许多计算。这些计算有的是立方体或超立方体中的所有值,有的是其中的一部分。由于计算的预先性,自然使 MOLAP 的查询能力可以预测,并且较为理想。而在 ROLAP 中进行查询分析,其结果往往很难预计。有的时候很快,有的时候则要很长时间才能获取答案。虽然在 ROLAP 的实际应用中,可以通过构造索引表和概要表来加快查询速度,但是对一些特殊的查询响应,ROLAP 的速度远不如 MOLAP。

2. 数据加载性能

在数据加载的操作中,MOLAP 除要完成数据的装载外,还需要对所有立方体中的所有值进行计算。这样,MOLAP 所需要的数据加载时间就比较长,因此在许多 MOLAP 中多维数据库的加载往往一个月才进行一次。而对于 ROLAP 来说,在数据加载过程中所要完成的操作是数据装载、索引和概要表的创建。由于在 ROLAP 中所进行的概要表创建量一般较少,因此 ROLAP 的加载时间要比 MOLAP 的短。有的 ROLAP 在实际应用中,甚至每天对关系型数据仓库和数据集市刷新一次。

3. 分析能力

由于 OLAP 的本质就是对数据库中的数据进行分析,因此,OLAP 的分析能力应该是衡量两者优劣的一个重要指标。MOLAP 在分析过程中的精度较高,具有分析的优势;而 ROLAP 的分析结果往往由于 SQL 语言的约束,使 ROLAP 的分析效果往往不如 MOLAP。因为对于现实中的许多问题的解决是很难用一条 SQL 语句来实现的,因此,许多 ROLAP 的供应商采用多种方法来解决这一问题,例如,用多个 SQL 语句来完成用户的查询分析请求,且将中间处理结果存放在一些临时表中。或在客户机与服务器之间再增加一台机器进行中间结果的处理。

4. 数据集市的大小

MOLAP 在实际应用中的数据存储量往往增长较快, 尤其所创建的多维模式中拥有多个维时。表 8-3 中的数据初看似乎要比表 8-2 中的数据量少, 如果再增加一个颜色维, 颜色维成员只有“红”“蓝”两种。要完成 MOLAP 的构建, 需要再增加一倍的存储空间。在所增加的空间中有的可能没有实际值出现, 会使多维表形成一个稀疏矩阵, 在稀疏矩阵中浪费了大量的存储空间。尽管可以采用各种方法来压缩稀疏矩阵, 但是没有一个能够解决各种情况下的稀疏矩阵方法。这种稀疏矩阵的出现, 必然随着维数的增加, 而呈现爆炸性的增长趋势。

作为 ROLAP 中使用的关系数据库, 一般不会出现稀疏矩阵的情况。而且在实际应用中, ROLAP 数据库可以支持无限增长的数据存储要求, 只要磁盘空间足够大。但是大多数的多维数据库的容量不能无限增长。

5. 维的管理

ROLAP 数据库由于采用星型模式构建, 星型模式的维表可能很宽, 可以包含很多列。例如, 客户维表可能包含这样一些列: 客户名称、家庭地址、家庭所在地邮编、办公地址、办公所在地邮编、客户类型、第一次采购日期、最后一次采购日期、采购数量等。作为管理人员可对这样一个表中的所有列进行查询、汇总、钻取等操作, 例如, 要求 ROLAP 依照国家的顺序列出销售总量, 然后钻取到省、市、县、直至所在地邮编。接着, 还可按照客户的最后一次采购日期进行分解。

这样比较复杂的操作在 MOLAP 中就比较难以完成, 因为在多维数据库中的操作是受到多维表中所包含的不同维的“层次”设置制约的。因为过多的维层次的设置将使维表需要的存储量成爆炸性的增长, 这是系统无法承受的。

6. 维护能力

OLAP 在构建成功以后, 需要不断地对其维护。MOLAP 能够较好地进行自我维护。在数据更新加载时, 只需要用 SQL 语句对其输入数据就可以了。而 ROLAP 在维护与聚集时却比较困难, 因为在数据加载和聚集时要填充多个结构, 需要打开或关闭索引。加载完成后, 还要考察其性能是否下降。如果性能下降, 需要增加索引或概要表。

从上面的 MOLAP 与 ROLAP 对比分析中, 可以说很难确定两者孰优孰劣。只能根据具体情况而定, 决定因素很多, 应用规模是一个主要因素。如果需要建立一个大型的、功能复杂的企业级数据仓库, 那就可能需要选择 ROLAP。例如, SQL Server 2000 中建立的维度表超过 1 千万个以上维成员时, 就不能采用 MOLAP 模式。如果希望建立一个目标单一、维数不是很多的分析型数据集市, 那么 MOLAP 可能是一个较佳的选择。

由于 MOLAP 与 ROLAP 在实际应用中各有千秋,人们自然希望有一个综合两者长处的 OLAP,这就是混合 OLAP (HOLAP)。HOLAP 将多维数据集市的数据按照多维结构存储在分析服务器上,但是不保存源数据。因此,HOLAP 的多维数据查询分析性能同 MOLAP 一样优越,若查询源数据则就不如将源数据存储在 MOLAP 中那样快速。但是对源数据的各种钻取操作,可像 ROLAP 一样灵活。

8.6.2 OLAP 的衡量标准

1993 年 E.F.Codd 提出了关于 OLAP 的 12 条标准,其目的是加深对 OLAP 的理解。事实上,这些标准已经成为 OLAP 工具所应该具有的关键特性的最小描述 (EMC)。其后这 12 条标准扩充到 18 条。因此对于在数据仓库中设计、使用 OLAP 的用户来说,了解这 18 条标准是必不可少的。

1. 多维性

OLAP 的多维性是必须具备的,作为 OLAP 的用户——管理人员所面对的企业和他所管理的对象是多维的、多角度的。这就决定了 OLAP 要能够为管理人员提供多维视图,来考察企业、考察管理对象。为满足多维性,关于企业的数据空间也应该是多维的。多维性能够使用户对多维数据进行切片、切块和旋转,轻松地完成传统方法需要很长时间才能实现的分析。

2. 直观性

直观性就是要求 OLAP 能够为用户提供直观、易做的数据操作,即只要轻松地利用鼠标的单击、双击和拖放,键盘的键击,GUI 的引导使用户轻而易举地完成数据的定位、向上的汇总、向下的钻取等复杂的数据分析操作。

3. 可访问性

可访问性是指存储在 OLAP 中的数据能够以合适的方式存储,便于用户的访问和查询。物理数据可以来源于任何系统类型,但是对用户是透明的,只有工具才需要了解这些数据源。OLAP 工具应该将自己的逻辑模式映射到物理数据存储,并可访问数据,还能进行所需要的转换,以给出单一的、连续一致的用户视图。

4. 解释性批处理提取

解释性批处理提取在 OLAP 中常常采用在 OLAP 引擎或服务器上存储立方体使用的混合多种工具来实现。

5. OLAP 分析模型

OLAP 分析模型是指在高层获取 OLAP 所支持的分析数据, 包括静态描述性报告、解释性分析、假设性分析和预测性分析等。

6. 客户机/服务器结构性

OLAP 应该建立在客户机/服务器的体系结构上, 使用户通过客户机与服务器的松弛耦合实现 OLAP。这种松弛的耦合可使不同的客户机连接到不同的服务器上, 而且能使多维数据库服务器被不同的应用系统和工具访问。服务器能够实现企业数据库的逻辑模型与物理模型之间的映射和一致性, 保证统一的公共概念模型、逻辑模型和物理模型的建立。客户端则实现应用逻辑和用户界面的操作, 使各种客户只需用最少的工作、使用最少的程序, 就能进行各种数据决策分析。

7. 透明性或开放性

透明性和开放性主要指 OLAP 对用户的透明和开放, 其次是指 OLAP 的数据源对用户的透明和开放。

OLAP 对用户的透明和开放, 要求 OLAP 应该处于一个真正的开放系统中, 分析工具可以嵌入分析人员所指定的任何位置, 既不影响工具的效能, 也不增加系统的复杂性。数据源的透明和开放, 可使用户只关心他要查询分析的问题, 而不必了解这些分析的数据是来自哪里。

8. 多用户性

OLAP 的多用户性可为多个用户同时在一个分析模型上进行操作, 或在同一个数据模型上建立不同的模型。多用户的要求必然要求 OLAP 在实际应用中, 必须保证数据的完整性和安全性, 并且能够进行数据的并发处理。

9. 处理非正规数据性

OLAP 的处理非正规数据性要求系统实现对从老的业务处理系统中获取的 OLAP 数据进行解耦, 而没有返回数据源的传播和计算。实际上这一标准, 要求系统能够达到“强聚合, 弱耦合”的系统一般设计标准。

10. 存储 OLAP 结果

OLAP 结果标准要求能将决策分析和数据源分开, OLAP 用户不能在公共概念视图的基础上对企业数据进行分析。在进行实际分析过程中要将分析的中间结果和最终结果,

另外使用一个存储区进行存放

11. 提取丢失值

OLAP 中的丢失值并不是零值，OLAP 中的空值可能是未知值，也可能是丢失值。提取丢失值是系统处理空值的一种方式。

12. 处理丢失值

OLAP 在处理丢失值时，OLAP 引擎应该忽略这些丢失值。

13. 弹性报告

OLAP 的弹性报告要求能从各方面提供从数据模型中分析出来的数据和信息，用户可按任何想要的方式来操作、分析、综合和查看数据，充分反映数据的多维特征，具有较高的灵活性。而且可由分析人员根据需要对各维的报告进行旋转、汇总及合并操作，以用户所需要的任何方式显示，并且报告的输出不应随着维数的增加而被削弱。

14. 一致性能报告

该标准要求 OLAP 能为用户提供时间可预计的报告。用户在其操作过程中系统的响应时间是一致的，这就需要对立方体进行预定义和计算。即使在数据的维数与综合层次增加时，提供给用户的报告能力和响应时间不应该有明显的下降。

15. 对物理层的自动调整

在 OLAP 环境中的关系模型中独立的物理数据，需要对物理层自动进行调整，关系模型需要具备在数据未卸载、删除或重定义结构时，能够对底层物理结构进行改变的能力。

16. 通用维

在 OLAP 中需要具备通用维，即在结构和操作能力方面完全一致的维。具有适用所有维的逻辑结构，提供给某一维的任何功能都能提供给其他维。这点目前还有许多争议，因为除了时间维，多数维都有自己的特性，相互之间总是存在差异。

17. 无限维与聚合层

OLAP 的维数不应该小于 15 个，而且用户可在任意给定的路径上建立任意多个聚集层次，给定联合路径的概括级别数据也是无限的。但是，实现这一标准很难，因为无限维与聚合层会导致数据在有限空间内的膨胀，将会用完整个存储空间。

18. 无限制跨维操作

在多维的数据分析中,所有维的生成和处理是平等的,OLAP 工具应该能够处理维间的相关计算,而不需要用户定义计算。若在计算时需要按照语言定义规则,则此种语言应该允许计算和数据操作跨越任何数目的数据维,而不必限制数据单元间的任何关系,也无需考虑每个单元包含的通用数据属性数目。也就是说,OLAP 的无限制跨维操作要能够在维之间进行符号操作,而不是仅仅对可测量数据操作。

尽管 Cood 在 1993 年就提出了 OLAP 的 12 条评价标准,以后又增加为 18 条,但是这些标准是基于对客户的研究提出的,到目前为止,还有较大的争议。随着对 OLAP 技术研究和对 OLAP 理解的深入,有人提出 OLAP 的更为简洁的定义,以加深对 OLAP 的理解。如 Nigel Pendse 提出的 FASMI (Fast Analysis Of Shared Multidimensional Information)。他将 OLAP 所满足的特点用五个词来描述:“Fast”指对用户请求的快速响应;“Analysis”指可以应用多种统计分析工具、算法对数据进行分析;“Shared”指多个用户在同时存取数据时,应该保证系统的安全性;“Multidimensional”则体现了 OLAP 应用中的多维实质;“Information”指应用所需的数据及其导出信息。若要实现 FASMI,可以采用的技术包括客户/服务器结构、时间序列分析模型、并行处理技术、面向对象技术、数据存储优化和多线索技术。

8.6.3 OLAP 服务器和工具的评价标准

目前,市场上所提供的 OLAP 工具很多,为了能在 OLAP 的设计应用中,选择适当的产品。必须从 OLAP 所具有的功能、访问性能和引擎功能和管理能力等方面对 OLAP 工具进行评价。

1. OLAP 功能

OLAP 作为一种数据分析技术,主要通过对现有的数据进行计算、转换产生新的信息,并且显示给用户。这就要求 OLAP 能够完成这样一些功能:

支持多维数据集中的维与层次,能沿某个维或一组维进行数据的聚集、汇总、预计计算和派生;能对某个维或一组维提供计算逻辑、公式和分析例程进行某种形式的操作;能够实现从一个维到另一个维的转换;能进行交叉维的计算,如在不同维之间进行成本分配,或在电子表格中按照不同维进行损益表的计算;能提供强大的分析模型,包括对选中维及维的元素的逻辑、公式、分析例程、聚集数据汇总数据和派生数据等,如在给定财务数据上计算内部回报率的财务模型;能够提供大量的函数,如财务、统计、代数、市场等各种函数;能够提供强大的计算和逻辑比较能力,如对数据的分级、比较、归类、

百分比、极值、均值等；具有智能化的与时间相关的处理，如按照给定时间段的日历安排；能够提供强大的导航分析，可以沿单个或多个维的轴、交叉表进行浏览或钻取。

2. 访问性能

作为由广大管理人员组成的 OLAP 用户，在使用 OLAP 时希望得到多种访问数据工具的选择，能够将广大用户所熟悉的访问工具融合进 OLAP。这些选择可能包含：

电子表格，作为常用的电子表格 EXCEL 已经被相当多的用户所认可，因此，在 OLAP 中至少应该提供将数据加载进电子表格的功能，以满足用户将从 OLAP 所获取的数据移作他用；在 OLAP 中有一些经常性用户，他们往往需要进行一些特定的应用，如果能够向这些用户提供功能丰富的、满足他们特定要求的、私有客户工具，无疑将增强 OLAP 的功能；能否与第三方工具结合，主要是指能否通过 API 将用户比较熟悉或功能更加强大的第三方工具加入 OLAP，以完成用户的需求；能否提供一些“非事实标准”接口，例如，VB, Pb, VC 等应用环境，或 OLE, DDE, CORBA 等接口，也是衡量 OLAP 工具访问性能的一个评价标准。

3. 引擎功能

OLAP 的服务引擎都应满足分析模型及应用在功能、规模和技术特征上的要求。这些要求主要集中在能否满足进行交互式预测和预算的应用程序的读写功能；能否满足在工作组情况下所进行的多用户读写操作，尤其是写操作往往导致重新计算派生的和经过计算所得到的信息，这些信息可能影响多个维及维的层次，使写锁的作用范围远超过初次的写范围，导致系统性能的下降；能否满足多数据库间的交互机制，因为在一个 OLAP 应用程序中虽然有一个数据库，往往出现多数据库之间的交互机制，因为在一个数据库中所产生的数据可能进入其他数据库；能否满足 OLAP 应用程序对数据范围的要求，在 OLAP 的用户界面中，可能需要数字、时间、日历、描述、BLOB 等，这样才能显示更多的图像类型，增加动态显示和执行报表的功能，有利于复杂分析的表达。

4. 管理能力

OLAP 并不像一般的业务操作系统，用户对其提出了强大的处理功能与便捷的使用要求，这必然要求 OLAP 能够提供有力的管理工具。这些管理工具应该具有这样一些功能：可以定义维的分析模型；能够生成并维护元数据存储；具有访问和使用控制的权限，主要解决如何控制用户对模型和数据的访问问题；从数据仓库或数据集市加载分析模型的管理问题；协调用户对多维数据的访问级别，保证用户进行不受其他用户干扰的分析；能为增强数据库的性能，或者为修改维模型，或者为修改数据而重新组织数据库；可将数据传送给客户，以便进一步分析或做本地分析。



本章小结

OLAP 作为数据仓库的一种应用工具,是数据仓库应用中必不可少的。OLAP 的操作必须依赖存在于数据仓库与 OLAP 之间的数据集市。由于 OLAP 需要对数据集市进行多维度的观察分析,因此数据集市可以采用立方体、超立方体的多维结构。

根据多维结构存储模式的不同,可将 OLAP 分成基于多维数据库的 MOLAP 和基于关系数据库的 ROLAP 两大类型。这两大类 OLAP 在实际应用中各有千秋,需要根据 OLAP 应用中的实际需要进行选择。在 MOLAP 和 ROLAP 两大类 OLAP 的基础上,有的 OLAP 生产公司推出了混合 OLAP——HOLAP 产品,希望同时兼有 MOLAP 和 ROLAP 两种类型 OLAP 的优点。

Codd 在 1993 年提出关于 OLAP 的 12 条标准以后,又将其扩充到 18 条。希望利用这 18 条作为衡量 OLAP 质量的标准,但目前对这 18 条中的有关标准争议较大。因此,在选择 OLAP 工具时,可从 OLAP 的功能、访问数据的性能、引擎的功能和管理的能力等方面考虑。



习题

- 8-1 什么是 OLAP? OLAP 是一种技术还是一种数据库?
- 8-2 OLAP 的系统结构有哪儿种? 不同的结构在进行在线分析时各有什么特点?
- 8-3 MOLAP 和 ROLAP 在 OLAP 的数据存储中各有什么特点? 在什么情况下,选择 MOLAP? 在什么情况下,选择 ROLAP?
- 8-4 OLAP 中的数据切片是如何实现的?
- 8-5 OLAP 中的钻取操作可以用来为哪些决策提供帮助?
- 8-6 请用一种 OLAP 工具完成对航空公司数据仓库的数据进行多维分析:总航班数与利润数,飞行时间、旅客数量与消耗食品,旅客服务成本与旅客的机票收入。



第 9 章

数据挖掘技术导论

引 言

信息在组织发展中的重要作用越来越得到人们的认同,许多组织都开发了各种信息收集处理系统。这些系统不仅给组织带来信息处理的便利,也给组织带来珍贵的财富——大量宝贵的数据。这些数据背后隐藏着极为重要的商业知识,但是这些商业知识是隐含的、事先未知的、具有潜在有用的价值。问题是如何才能发掘这些知识,传统的信息处理工具已经不能应付这一要求,人们需要一种方法,自动地分析数据、自动地发现和描述数据中隐含的商业发展趋势、自动地标记数据、对数据进行更高层次的分析,以更好地利用这些数据。

通过本章学习,可以了解:

- ◆数据挖掘的发展
- ◆数据挖掘的基本原理
- ◆数据挖掘的分类
- ◆数据挖掘的应用范围
- ◆数据挖掘的应用过程

9.1 数据挖掘概述

1989年8月,在第11届国际人工智能联合会议的专题研讨会上,首次提出基于数据库的知识发现(KDD, Knowledge Discovery in Database)技术。该技术涉及机器学习、模式识别、统计学、智能数据库、知识获取、专家系统、数据可视化和高性能计算等领域,技术难度较大,一时难以应付信息爆炸的实际需要。到了1995年,在美国计算机年会(ACM)上,提出了数据挖掘(DM, Data Mining)的概念,即通过从数据库中抽取隐含的、未知的、具有潜在使用价值信息的过程。由于数据挖掘是KDD过程中最为关键的步骤,在实际应用中对数据挖掘和KDD这两个术语的应用往往不加区别。

9.1.1 数据挖掘的发展

在促进数据挖掘诞生、发展、应用的众多原因中主要有4种:超大规模数据库的出现、先进的计算机技术、经营管理的实际需要和对这些数据的精深计算能力。

1. 超大规模数据库的出现

大规模数据库,尤其是数据仓库的出现,促使数据挖掘得到迅速发展与应用。依靠计算机自动收集的各种业务处理数据,使许多大规模数据库或数据仓库拥有大量的业务处理数据、市场变化数据,使数据挖掘技术有了赖以生存的基础。如果没有这些大规模数据库,很难想象数据挖掘技术对什么进行挖掘。

2. 先进的计算机技术

计算机技术在过去的短短几十年内得到了快速的发展,尤其是近几年的网络技术和并行处理体系的发展,使人们拥有计算能力更强、运算速度更快的计算机体系结构。以前需要大量时间、大量人力的工作,现在只要很少的时间和人力就可以解决了。使大量的管理人员得以将自己的精力从繁重的日常信息处理工作中解脱出来,有时间、有能力对激增的数据进行高层次的分析,从中寻找那些对企业战略发展具有重要意义的商业规律和市场趋势。这些先进的计算机技术水平成为促进数据挖掘技术发展的第二个重要因素。

3. 经营管理的需要

进入21世纪以后,全球经济一体化的进程日益加快,企业所面临的市场竞争压力日趋严重,企业经营者希望能够从企业积累的大量历史数据中找到应对日趋严重的竞

争压力良方,希望能够从这些数据中找到经营管理中问题的根本原因。例如,经营管理者希望了解企业的某些产品为什么销售业绩良好,是产品自身的原因还是销售的原因?如果是销售的原因,这些产品的销售人员在销售中采用了什么销售方式?出于这些原因的考虑,使企业经营管理人员,特别是决策人员,希望能用某种工具从这些数据中去找原因。能够快速地从大量数据中挖掘出对经营管理有用的信息,以应对瞬息万变的市场压力。

4. 对数据挖掘的精深计算能力

大规模数据的挖掘需要复杂的、精深的计算能力,这些精深的计算能力主要基于统计学、集合论、信息论、认识论和人工智能等各种学科理论。这些数据挖掘方法和技术形成了许多具有特点的应用领域。也正是在这些精深计算能力,成为促进数据挖掘诞生和发展的中坚力量。

因此可以说,数据挖掘是信息技术发展到一定阶段的必然产物,是拥有大规模数据库、高效的计算能力、经营管理的压力和有效的计算方法后的产物,是从存放在数据库、数据仓库或其他信息库大量数据中挖掘有用知识的一个过程。

9.1.2 数据挖掘的定义

数据挖掘的定义现在很多,在不同的教科书上有不同的定义。表达方式虽然不同,但本质都是一样的。这里主要从技术角度和商业角度给出数据挖掘的定义。

1. 数据挖掘的技术定义

从技术角度看,数据挖掘是从大量的、不完全的、有噪声的、模糊的、随机的实际数据中,提取隐含在其中的、人们不知道的、但又是潜在有用的信息和知识的过程。

什么是知识?从广义上理解,数据、信息也是知识的表现方式,但是人们更将概念、规则、模式、规律和约束等看做知识。这里所说的知识都是相对的,是有特定前提和约束条件的,在特定领域中具有实际应用价值。同时还要能够易于被用户理解,最好能用自然语言表达所发现的结果。

人们将数据看做形成知识的源泉,好像从含金的大量矿石中淘金一样。原始数据可以是结构化的,如关系数据库中的数据;也可以是半结构化的,如文本、图形和图像数据;甚至是分布在网络上的异构数据。发现知识的方法可以是数学的,也可以是非数学的;可以是演绎的,也可以是归纳的。发现的知识可以用于信息管理,查询优化,决策支持和过程控制等。因此,数据挖掘是一门交叉学科,它把人们对数据的应用从低层次的简单查询,提升到从数据库中挖掘知识,提供决策支持。在这种需求推动下,汇集了

不同领域的研究者,尤其是数据库技术、人工智能技术、数理统计、可视化技术、并行计算等方面的学者和工程技术人员,投身到数据挖掘这一新兴的研究领域,形成新的技术研究和开发热点。

2. 数据挖掘的商业定义

从商业应用角度看,数据挖掘是一种崭新的商业信息处理技术。其主要特点是对商业数据库中的大量业务数据进行抽取、转化、分析和模式化处理,从中提取辅助商业决策的关键知识,即从一个数据库中自动发现相关商业模式。实际上多年前,统计学家就开始手工挖掘数据库,从数据库中寻找符合统计学规律的有意义的模式。这也是统计学类型的数据挖掘技术,是目前数据挖掘技术中最为成熟的重要原因之一。

数据挖掘是利用统计学和机器学习的技术,探求那些符合市场、客户行为的模式。目前,数据挖掘已经可使挖掘技术自动化,将数据挖掘与商业数据仓库相结合,以适当的形式将挖掘结果展示给企业经营管理人员。对于数据挖掘的应用不仅依靠良好的算法建立模型,而且更重要的是要解决如何将数据挖掘技术集成到当今复杂的信息技术应用环境中。其次,还要有数据挖掘分析人员参与,因为数据挖掘技术不具备人所特有的经验和直觉,不能区分哪些挖掘出的模式在现实中是有意义的,哪些是无意义的。因此,数据挖掘分析人员的参与是必不可少的。

简而言之,数据挖掘是一类深层次的数据分析。数据分析本身已经有很多年的历史,只不过以往数据挖掘收集和分析的目的在于科学研究,而且限于当时计算能力的限制,对大数据量进行分析的复杂数据分析方法无法得到实际的应用。现在,由于业务处理自动化系统的实现,在商业领域中生成了大量的业务数据。这些数据并不是为了分析的目的而收集的,而是由于业务处理操作而获取、积累的。面对这些数据,所有企业都面临一个共同的问题:企业所积累的数据量越来越大,但其中能被企业直接利用的真正有价值的信息却很少。因此从大量的数据中经过深层次分析,获得有利于商业运作、提高商业竞争力的信息,就像从矿石中发掘金子一样困难。

数据挖掘可以描述成:按企业既定业务目标,对大量的企业数据进行探索和分析,揭示隐藏的、未知的或验证已知的商业规律,且进一步将其模式化的数据处理方法。它最吸引人的地方就是能够建立预测型而不是回顾型的模型。将数据挖掘工具与传统的数据分析工具进行比较(见表 9-1),可以发现传统数据分析工具的分析重点在于向管理人员提供过去已经发生什么,描述过去的事实。例如,上个月的销售成本是多少。而挖掘工具则在于预测未来的情况,解释过去所发生事实的原因。例如,下个月的市场需求情况怎样,或者某些客户为什么会转向竞争对手。分析的目的也不同,前者是为了从过去的事实中列出管理人员感兴趣的事实,例如,哪些是公司最大的客户。而后者则是要找出未来可能成为公司最大的客户。从两者进行分析时所需要的数据量看,也有明显的差

异,前者需要的数据量并不很大,而后者则需要海量数据才能运行。两者的启动方式也有较大的差别,前者主要依靠各种人员启动,后者则依靠数据本身和系统来启动。当然,前者的技术已经相当成熟,而后者除统计分析工具外,其他的工具则处于发展阶段中。

表 9-1 数据挖掘工具与传统数据分析工具的比较

	传统数据分析工具 (DSS/OLSS)	数据挖掘工具
工具特点	回顾型的、验证型的	预测型的、发现型的
分析重点	已经发生了什么	预测未来的情况、解释发生的原因
分析目的	从最近的销售文件中列出最大客户	锁定未来的可能客户,以减少未来的销售成本
数据集大小	数据维、维中属性数、维中数据均是少量的	数据维、维中属性数、维中数据均是庞大的
启动方式	企业管理人员、系统分析员、管理顾问启动与控制	数据与系统启动,少量的人员指导
技术状况	成熟	统计分析工具已经成熟,其他工具正在发展中

这两种数据分析工具的差别,根源在于探索数据关系时所采用的方法不同。传统的数据分析工具是对过去情况的验证,而数据挖掘技术则是基于发现型的、预测型的,运用模式匹配等各种算法对数据之间的关系进行挖掘。

9.1.3 数据挖掘与数据仓库关系

根据数据挖掘的定义可以看出,数据挖掘包含一系列旨在从数据库中发现有用而未发现的模式的技术,如果将其与数据仓库紧密联系在一起,将获取意外的成功。传统的观点认为数据挖掘技术扎根于计算科学和数学,不需要也不得益于数据仓库。这种观点并不正确,成功的数据挖掘的关键之一就是访问正确、完整和集成的数据,才能进行深层次的分析,寻求有益的信息。而这些正是数据仓库所能够提供的,数据仓库不仅是集成数据的一种方式,而且数据仓库的联机分析功能——OLAP 还为数据挖掘提供了一个极佳的操作平台。如果数据仓库与数据挖掘能够实现有效的联结,将给数据挖掘带来各种便利和功能。

首先,由于大多数数据挖掘工具要在集成的、一致的、经过清理的数据上进行挖掘。这就需要在数据挖掘中有一个费用昂贵的数据清理、数据变换和数据集成过程,作为数据挖掘的预处理。而已经完成数据清理、数据变换和数据集成的数据仓库,完全能为数据挖掘提供它所需要的挖掘数据。使数据挖掘免除了数据准备的繁杂过程。

其次,在数据仓库的构造过程中已经围绕数据仓库组建了包括数据存取、数据集成、数据合并、异种数据库的转换、ODBC/OLE DB 的连接、Web 访问和服务工具以及报表与 OLAP 分析工具等全面的数据处理和数据分析基础设施。在数据挖掘过程中所需要的数据处理与分析工具完全可在数据仓库的数据处理与数据分析工具中找到,根本没有必要为数据挖掘重新设置同样的基础设施。

此外,在数据挖掘过程中,常常需要进行探测式的数据分析,穿越各种数据库,选择相关数据,对各种数据选择不同的粒度,以不同的形式提供知识或结果。而数据仓库中的 OLAP 完全可为数据挖掘提供有关的数据操作支持,例如,对数据立方体或数据挖掘中间结果进行数据的下钻、上卷、旋转、过滤、切块或切片,且以 OLAP 的可视化功能为数据挖掘过程或挖掘结果提供良好的操作平台,这些都将极大地增强数据挖掘的功能和灵活性。

最后,在数据挖掘过程中,如果将数据挖掘与数据仓库进行有效的联结,将增加数据挖掘的联机挖掘功能。用户在数据挖掘的过程中,可以利用数据仓库的 OLAP 与各种数据挖掘工具的联结,使用户可以为数据挖掘选择合适的数据挖掘工具,能够在数据挖掘过程中灵活地组织挖掘工具以增强数据挖掘能力,同时还为用户灵活地改变数据挖掘的模式与任务提供便利。

9.2 数据挖掘技术与数据挖掘工具

数据挖掘技术其实是信息技术逐渐演化的结果,是人们长期对数据库技术进行研究和开发的结果。起初,各种商业数据仅仅存储在计算机的数据库中,然后发展到对数据库中的商业数据进行查询和访问,进而发展到对数据库的即时遍历。数据挖掘是革命性的变革,使数据库技术的应用进入一个更高级的阶段,它不仅能对过去的数据进行查询和遍历,并且能够找出数据之间的潜在关系,从而加大信息应用的深度。随着海量数据搜集、强大的多处理器计算机和数据挖掘算法这三种基础技术的发展成熟,数据挖掘技术在商业应用中开始得到广泛的重视。

9.2.1 常用数据挖掘技术

常用的数据挖掘技术可以分成统计分析类、知识发现类和其他类型的数据挖掘技术三大类。

1. 统计分析类

统计分析(或称数据分析)技术中使用的数据挖掘模型有线性分析和非线性分析、回归分析、逻辑回归分析、单变量分析、多变量分析、时间序列分析、最近邻算法和聚类分析等技术。利用这些技术可以检查那些异常形式的数据,然后,利用各种统计模型和数学模型解释这些数据,解释隐藏在这些数据背后的市场规律和商业机会。例如,可以使用统计分析工具寻求最佳商业机会,增加市场份额和利润;利用全面质量管理程序,提高产品或服务的质量,使客户更加满意;通过对流水线产品制造的调整或企业业务过

程的重整, 增加利润。在所有的数据挖掘技术中, 统计型数据挖掘工具是数据挖掘技术中最成熟的一种, 已经在数据挖掘中得到广泛的应用。

2. 知识发现类

知识发现类数据挖掘技术是与统计类数据挖掘技术完全不同的一种挖掘技术。它可以从数据仓库的大量数据中筛选信息, 寻找市场可能出现的运营模式, 发掘人们所不知道的事实。

知识发现类数据挖掘技术包含人工神经网络、决策树、遗传算法、粗糙集、规则发现和关联顺序等。

人工神经网络是模拟人脑神经元结构, 以 MP 模型和 Hebb 学习规则为基础, 建立三大类多种神经网络模型。前馈式网络以感知知识、反向传播模型、函数性网络为代表, 可用于预测和模式识别等方面; 反馈式网络以 Hopfield 的离散模型和连续模型为代表, 分别用于联想记忆和优化计算; 自组织网络以 ART 模型、Koholon 模型为代表, 用于聚类。神经网络的知识体现在网络连接的权值上, 是一个分布式矩阵结构; 神经网络的学习体现在神经网络权值的逐步计算上 (包括反复迭代或累加计算)。

决策树是一个类似于流程图的树结构, 其中每个内部节点表示在某个属性上的测试, 每个分枝代表一个测试输出, 而每个树叶节点代表类或类分布。由于每个决策或事件 (即自然状态) 都可能引出两个或多个事件, 导致不同的结果, 把这种决策分支画成图形很像一棵树的枝干, 故称决策树。树的最顶层节点是根节点, 内部节点用矩形表示, 而树叶节点用椭圆表示。

遗传算法是近几年发展起来的一种崭新的全局优化算法, 借用了生物遗传学的观点, 通过自然选择、遗传、变异等作用机制, 实现各个个体的适应性的提高; 解决问题时, 要对待解决问题的模型结构和参数进行编码, 一般用字符串来表示, 这个过程就将问题符号化、离散化了。遗传算法由三个基本过程组成: 繁殖 (选择) 是从一个旧种群 (父代) 选出生命力强的个体, 产生新种群 (后代) 的过程; 交叉 (重组) 选择两个不同个体 (染色体) 的部分 (基因) 进行交换, 形成新个体的过程; 变异 (突变) 对某些个体的某些基因进行变异的过程。

标准遗传算法是不收敛于全局最优解的, 而当保留当前所得的最优值时就是收敛于全局最优解的。这种收敛性只是指计算时间趋向无穷时的可以以概率 1 达到全局最优解。

粗糙集 (RS) 能够在缺少关于数据先验知识的情况下, 只以考察数据的分类能力为基础, 解决模糊或不确定数据的分析和处理问题。粗糙集用于从数据库中发现分类规则的基本思想是将数据库中的属性分为条件属性和结论属性, 对数据库中的元组根据各个属性不同的属性值分成相应的子集, 然后对条件属性划分的子集与结论属性划分的子集之间上下近似关系生成判定规则。所有相似对象的集合称为初等集合, 形成知识的基本

成分。任何初等集合的并集称为精确集，否则，一个集合就是粗糙的（不精确的）。每个粗糙集都具有边界元素，也就是那些既不能确定为集合元素，也不能确定为集合补集元素的元素。而精确集是完全没有边界元素的。

关联规则是数据挖掘的一种主要形式，是与大多数人想象的数据挖掘过程最为相似的一种数据挖掘形式，即在大型数据库中“淘金”——人们感兴趣的规则。在关联规则系统中，规则是“如果怎么样、怎么样、怎么样，那么就怎么样”的简单形式表示的。根据规则中所处理的值类型，关联规则可以分成布尔关联规则和量化关联规则两种。根据关联规则集涉及不同的抽象层次，关联规则可分成多层关联规则和单层关联规则。关联规则的评价标准可用正确率、覆盖率和兴趣度来衡量。

3. 其他数据挖掘技术

其他数据挖掘技术中包含文本数据挖掘、Web 数据挖掘、分类系统、可视化系统、空间数据挖掘和分布式数据挖掘等。

文本数据挖掘和 Web 数据挖掘是近几年新发展起来的崭新数据挖掘技术。前者主要是为了满足对非结构化信息的挖掘的需要，后者则是针对日益发展的因特网技术所带来的大批量网络信息的挖掘。

分类系统应该说也是一种知识发现技术，但是它的实现可以采用各种知识发现类技术的支持，而且在数据挖掘中具有特殊重要的作用。本书将在第 12 章中单独介绍。

可视化系统则是为使数据挖掘能以图形或图像的方式在屏幕上显示出来，且能交互处理。这样，可以很清楚地发现隐含的和有用的知识。可视化技术可分为两类：表示空间数据场的体可视化技术和表示非空间数据的信息可视化。可视化数据挖掘可以分为数据可视化、数据挖掘结果可视化、数据挖掘过程可视化和交互式数据可视化挖掘。

空间数据挖掘则是基于地理信息系统的数据挖掘技术。地理信息系统（GIS）的应用领域现已扩展到航天、电信、电力、交通运输、商业、市政基础设施管理、公共卫生及安全、油气等其他矿产资源的勘测等诸多领域。在这些领域中的数据挖掘技术可用于地图、预处理后的遥感数据、医学图像数据和 VLSI 芯片设计空间数据库中非显式的知识、空间关系和其他有意义的模式的提取。空间数据挖掘方法目前主要有空间数据分类、空间数据关联分析和空间趋势分析等。

分布式数据挖掘是基于分布式数据库的，利用分布式算法从分布式数据库中挖掘知识的技术。分布式数据挖掘技术主要用于对水平方式分布或垂直方式分布的数据库系统中数据的挖掘。水平分布式数据挖掘算法只需要首先完成各个站点的局部数据分析，构建局部数据模型；最后，组合不同数据站点上的局部数据模型，获得全局数据模型即可。垂直式分布的数据库系统，则需要采用汇集型数据挖掘方法来实现。分布式数据挖掘将更加有利于对分布式数据库数据资源的利用。

9.2.2 常用数据挖掘工具

由于数据挖掘工具在企业经营管理、政府行政管理决策支持以及科学研究等领域获得了广泛的应用,许多软件开发商或研究机构纷纷推出各式数据挖掘商品化工具。这些工具可以按照使用方式、所采用的挖掘技术和应用范围进行分类。

1. 按使用方式分类的数据挖掘工具

数据挖掘工具按照使用方式可以分成决策方案生成工具,商业分析工具和研究分析工具三大类。

决策方案生成工具往往是针对某个特定行业或特定问题而开发的一类数据挖掘工具,例如,金融行业的欺诈检查工具,零售行业的客户流失分析工具等。

商业分析工具有两种类型。一种是为用户提供一个黑箱,用户只需要将需要分析的对象和相关的一些环境因素提供给工具,数据挖掘工具将自动给出数据挖掘的结果,其内部的一些复杂模型并不向用户展示。这种类型的数据挖掘工具适合管理人员使用。另一种数据挖掘工具则向用户展示数据挖掘模型,用户可以根据需要去选择数据挖掘模型或对数据挖掘模型进行适当的控制。例如,这类工具可以将决策树展示给用户,用户可对决策树进行切片处理,这一类工具主要为企业管理顾问或商业分析人员服务。

研究分析工具为用户提供更大的数据挖掘应用的自由空间,其用户主要是数据挖掘研究人员或商业分析人员。这些工具包含一些数据挖掘研究领域的最新研究成果,例如文本挖掘、Web挖掘或图形以及可视化工具等。

2. 按数据挖掘技术分类的数据挖掘工具

按照数据挖掘的技术可以分成基于神经网络的工具,基于规则和决策树的工具,基于模糊逻辑的工具和综合性数据挖掘工具等。

基于神经网络的工具由于有非线性数据的快速建模能力,在实际应用中越来越流行。开发过程基本上是首先进行数据聚类,然后分类计算权值。神经网络很适合非线性数据和含噪声数据,所以在市场数据库的分析和建模方面应用广泛。

基于规则和决策树的工具采用规则发现或决策树分类技术,发现数据模式和规则,其核心是某种归纳算法。这类工具通常是对数据库的数据进行开发,生产规则和决策树,然后对新数据进行分析和预测。这类工具的主要优点是,规则和决策树都是可读的。

基于模糊逻辑工具的数据挖掘方法是应用模糊逻辑进行数据查询和排序等。该工具使用模糊概念和“最近”搜索技术的数据查询工具,让用户指定目标,然后对数据库进行搜索,找出接近目标的所有记录,且对结果进行评估。

综合性工具采用多种数据挖掘方法,这类工具一般规模较大,适合对大型数据库的数据挖掘。综合性数据挖掘工具的数据挖掘能力很强,但价格昂贵,并且用户需要花很长时间进行学习,才能掌握这类工具的应用。

3. 按应用范围分类的数据挖掘工具

按照数据挖掘的应用范围可将挖掘工具分成专用型数据挖掘工具和通用型数据挖掘工具两类。

(1) 专用型的数据挖掘工具

专用型数据挖掘工具主要用于某个特定领域。例如,美国加州理工学院与日本的 Kayyad 设计的 SKICAT,能够对大规模的空间数据进行分析,识别遥远空间的星体。芬兰赫尔辛基大学所研制的 TASA,能够采用特殊算法处理网络通信中的数据对网络通信故障发出警报。由于专用型的数据挖掘工具针对性较强,采用一些特殊的算法对特定的数据集进行处理,数据挖掘的效率较高,挖掘出的知识可靠性也高,但是应用范围受到限制。

(2) 通用型的数据挖掘工具

通用型数据挖掘工具一般不考虑所挖掘对象的实际含义,只提供各种通用挖掘算法,允许用户自定义数据源进行多模式挖掘。由于这种类型挖掘算法的通用性,在数据的挖掘过程中很难进行算法的优化。因此,数据挖掘效果往往不能使所有用户都满意。

通用型数据挖掘工具有 IBM 公司的 IM 智能挖掘器。这是一套包括 Explorer, Diamond 和 Quest 在内的软件产品,可以用来提供高端数据挖掘解决方案。其中的 Explorer 是一种聚类的神经网络工具, Diamond 是一种可视化数据挖掘软件产品,而 Quest 则提供关联规则、分类规则、序列模式与相似序列等模式。

SPSS 公司的统计软件包 SPSS 在统计领域处于领先的地位,其中的线性回归分析结果和类似的数据挖掘工具对数据挖掘的结果是一致的,而这些挖掘工具采用的是传统统计方法。

Red Brick 系统公司的 Red Brick 数据挖掘工具为第一个将数据挖掘解决方案与数据库集成在一起的数据挖掘选项。与数据库的联结,减少了传统数据挖掘中需要的大量数据准备时间,并且提供扩展的 SQL 语言;用户可以使用 SQL 语言建立、存取和访问数据仓库中的模型。

9.2.3 数据挖掘工具的评价标准

随着数据挖掘技术日益发展的同时,出现了许多数据挖掘工具。如何选择满足需要的数据挖掘工具,成了数据挖掘应用中首要解决的问题。在选择数据挖掘工具时,一般

可以参照以下评价标准。

1. 模式种类的数量

数据挖掘工具能够提供的模式越多，它的知识发现能力越强；多种类型模式的结合应用，有助于降低问题的复杂性。例如，可以先用聚类将数据集分组，再在各数据组上挖掘预测性模式，要比单纯在整个数据集上进行数据挖掘更加有效。

2. 解决复杂问题的能力

由于挖掘数据量一般都比较大会比较大，因此，算法的时空复杂性成为许多挖掘工具实际应用中的重要限制因素。如果算法的复杂性随着数据量的增大、模式精细度的提高、准确度要求的增加而呈现指数增长，就将严重限制数据挖掘工具的应用。

为了了解数据挖掘工具解决复杂问题的能力大小，可从挖掘工具的模式应用、数据选择和转换能力、可视化程度、扩展性等方面考察。

多种类别模式的结合使用往往有助于发现有用的商业模式，降低问题的复杂性。特别是与分类有关的模式，可用不同的算法来实现，以适应不同的需求环境。数据挖掘工具如果能够多种途径产生同种模式，可以提高其解决复杂问题的能力。

数据选择和转换能力对挖掘工具解决复杂问题能力的影响也是相当大的。因为知识模式通常被大量的数据项所隐藏，这些数据有的是冗余的，有的是完全无关的。这些数据项的存在会影响有价值模式发现的能力。数据挖掘工具的一个很重要功能，就是能够减低数据的复杂性，提供选择正确数据项和转化数据值的能力，这些能力都将增加数据挖掘工具解决复杂问题的能力。

可视化工具不仅为用户提供了直观、简洁的数据挖掘方法，方便了用户使用数据挖掘工具；更重要的是可视化工具有助于用户对重要数据的定位，对模式质量的评价，从而降低解决复杂问题时建模的难度。

数据挖掘工具的扩展性也是提高挖掘工具解决复杂问题能力的一个重要因素。数据挖掘工具的扩展性可以提高处理大量数据的效率。这就要在选择数据挖掘工具时了解挖掘工具能否充分利用硬件资源？是否支持并行计算？当处理器的数量增加，计算规模是否相应增长？是否支持数据并行存储？为单处理器的计算机编写的数据挖掘算法不会在并行计算机上自动以更快的速度运行。为了更好地发挥并行计算机的优点，需要有支持并行计算的算法。

3. 操作性能

操作性能的好坏是一个影响挖掘工具性能的重要因素。图形界面友好的工具可以方便用户，引导用户执行任务，为用户节省数据挖掘时间。具有嵌入技术（API）的挖掘工

具能使数据挖掘工具的性能得到提高,应用程序能够嵌入挖掘工具,缩短开发时间。如果数据挖掘工具能够允许用户通过 GUI、程序设计语言或 SQL 语言将模式运用到已经存在或新增加的数据上,或将模式导出到程序或数据库中,将极大地提高挖掘工具的易操作性。

4. 数据获取能力

数据挖掘工具的使用基础是数据库或数据仓库。因此,一个优秀的数据挖掘工具可以使用 SQL 语句直接从数据库或数据仓库中读取数据,这样可以简化数据准备工作,并且可以充分利用数据库的优点。没有一种工具可以支持所有类型的数据库或数据仓库,但应该能够通过通用接口连接大多数流行的数据库或数据仓库,这将提高数据挖掘工具的使用范围。

5. 挖掘结果的输出

数据挖掘工具不仅能够将挖掘结果以多种方式输出,而且要求输出的结果便于用户的理解与应用。传统的查询工具、可视化工具可以帮助用户理解数据挖掘结果。因此,数据挖掘工具能否提供与传统工具集成的简易途径和接口,是衡量数据挖掘工具好坏的标准。如果这些挖掘结果的输出能够以图形、报告、逻辑公式等可视化方式输出,或以先验知识方式输出,为今后的数据挖掘提供准备,都能提高数据挖掘工具的性能。

6. 噪声数据的处理及挖掘工具的鲁棒性

在许多情况下,数据源都包含噪声,数据挖掘工具应该能对携带噪声的数据进行挖掘,或对带噪声数据适当处理后也能进行正常的数据挖掘。噪声数据的处理从另一个角度说明挖掘工具需要具有一定的鲁棒性。从数据挖掘工具的目标看,希望对未知的对象做出正确的判断,但要求挖掘工具能对所有的对象做出这种预测是不可能的。不过,数据挖掘工具至少要有一定的数据误差处理能力,能够应对非法输入、内存空间不足等异常情况。

9.2.4 常用数据挖掘工具的选择

由于数据挖掘工具种类繁多,用户在选择挖掘工具时,要从工具的实用性和技术性方面进行考察。

数据挖掘工具的实用性是指用户在选择数据挖掘工具时,确定是否有专用挖掘工具。如果有专用挖掘工具并且能够胜任用户的数据挖掘应用,应该首先考虑采用专用挖掘工具。如果采用通用数据挖掘工具,就要考察这些工具是否能够提供有用的知识,所提供

的知识能否阻止不希望发生的事情发生, 能否促进希望发生的事情发生, 能否减低获取这些新知识的成本或缩短知识发现的时间。

从技术性方面考察数据挖掘工具时, 需要根据数据挖掘工具评价标准, 选择那些技术性能指标良好的数据挖掘工具。

9.3 数据挖掘技术的应用过程

启动一个数据挖掘项目很容易, 但是完成它却很难。如果不能获得一个商业模式, 数据挖掘项目将被迫停止。容易启动的理由是数据挖掘项目往往将注意力集中在一些对于企业来说很重要, 同时也很难解决的问题上, 例如提高客户利润是任何企业的目标, 因为大多数数据挖掘项目关心的是如何提高客户的利润率, 而且它们也很有可能提高它。但是, 数据挖掘技术本身的复杂性, 再加上需要消耗大量的人力和财力, 因此, 数据挖掘项目的成功需要花费相当的心血, 依照规范的过程进行操作。

9.3.1 数据挖掘过程

数据挖掘过程一般需要经历确定挖掘对象、准备数据、建立模型、数据挖掘、结果分析与知识应用这样几个阶段 (见图 9.1), 这些阶段在具体实施中可能需要重复多次。为完成这些阶段的任务, 需要不同专业人员参与其中, 这些专业人员主要是业务分析人员、数据分析人员和数据管理人员。

1. 确定挖掘对象

定义清晰的挖掘对象, 认清数据挖掘的目标是数据挖掘的第一步。数据挖掘的最后结果往往是不可预测的, 但要探索的问题应是有预见性的、有目标的。为了数据挖掘而挖掘数据带有盲目性, 往往是不会成功的。在定义挖掘对象时, 需要确定这样一些问题: 从何处入手? 需要挖掘什么数据? 要用多少数据? 数据挖掘要进行到什么程度? 虽然在数据挖掘中, 常常事先不能确定最后挖掘的结果到底是什么? 有的挖掘技术是不需要因变量的无教师挖掘或聚类分析, 但是在为这些挖掘工具准备数据过程中就已经表明了挖掘者的意向。例如, 选择的数据是描述使用信用卡客户的实际支付情况, 那么数据挖掘者的挖掘工作就可能是围绕着获取信用卡使用者的实际支付情况展开的。

在数据挖掘的第一步中, 有时还要用户提供一些先验知识, 例如概念树等。这些先验知识可能是用户的业务领域知识或是以前数据挖掘所获得的初步成果。这就意味着数据挖掘是一个过程, 在挖掘过程中可能提出新的问题, 可能尝试用其他方法来检验数据, 在数据的子集上进行同样的研究。

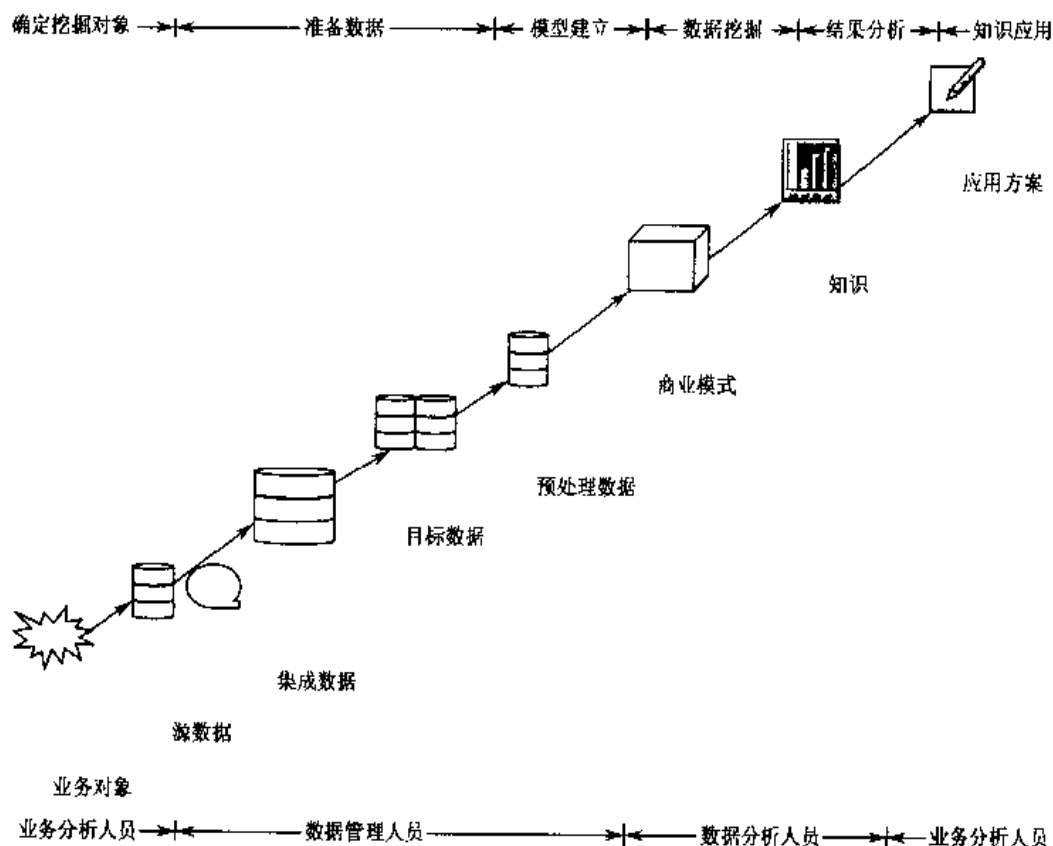


图 9.1 数据挖掘过程

有时业务对象是一些已经理解的数据，但是在一些情况下还需要对这些数据进行挖掘。此时的数据挖掘目标可能不是在寻找新的有价值的信息，而是通过数据挖掘验证假设的正确性，或是通过同样方式的数据挖掘查看模式是否发生了变化。如果在经常性的同样数据挖掘中有一次挖掘没有出现与以前同样的结果，这就意味着模式已经发生了变化，需要进行更深入的挖掘。例如，在将数据挖掘应用于客户关系管理中时，就需要对客户关系管理的商业主题进行仔细的定义。每个 CRM 应用都有一个或多个商业目标，要为每个目标建立恰当的模型。例如，“提高客户对企业促销的响应率”或“提高每个客户响应的价值”这两个目标所需要的模型是不同的，并且在定义问题的同时，也生成了评价 CRM 应用结果的标准和方法，即确定了数据挖掘结果的评价指标。

2. 准备数据

(1) 数据的选择

在确定数据挖掘的业务对象后，需要搜索所有与业务对象有关的内部和外部数据，从中选出适合于数据挖掘应用的数据。如果数据挖掘是基于数据仓库的，那么数据的选择将比较简单，因为数据仓库已经为数据挖掘者准备好可用于数据挖掘的基本数据。否则，就要从各种数据源去选择用于数据挖掘的数据。这就意味着需要集成和合并数据到

单一的数据挖掘库中,并且协调来自多个数据源的数据在数值上的差异。对这些数据值差异的协调是解决数据挖掘质量的关键。多个数据源中出现的差异主要在数据定义和使用的方法上。有些数据值的矛盾很容易发现,如同样的客户有几个不同的(不同的系统正在使用)的地址。也有一些非常难以捉摸,如同一个客户有不同的名字,最糟的是有不同的客户关键字。在数据准备阶段,这些问题必须解决好。

在进行数据选择时,根据数据挖掘的需要,分析清楚哪些数据是数据挖掘中比较重要的数据源。例如,在数据挖掘中希望描述对某个产品广告促销敏感客户与不敏感客户的特征,现在已经有客户对某种产品促销广告反应、客户的购买、客户的家庭、销售员、产品以及时间信息,共有6个维。显然在这6个信息中关于客户的3个维与产品维是必须的,而销售员维信息除非想了解销售员与较多(或较少)的敏感客户的接触情况,否则与目前所要研究的问题没有明显的联系;而时间维则在确定季节与产品广告的推出关系研究中是有作用的。

在确定了数据维后,还需要确定维中所包含的数据域描述能力的情况。例如,客户的家庭和时间维中的描述能力较差,需要增加家庭中的婚姻状况、性别、收入水平、家庭类型等数据元素;而时间维则希望包含日期、时间等。

对数据的选择,必须在建立数据挖掘模型之前完成。例如,数据挖掘在应用于客户关系管理之前,必须理解所使用的数据,需要通过收集各种数据描述(如平均值、标准差等统计量)和注意数据分布来开始了解数据。有时还需要为多元数据建立交叉表(枢轴表),以进一步理解数据。

(2) 数据的预处理

在选择数据后,还需要对数据进行预处理,对数据进行清洗,解决数据中的缺值、冗余、数据值的不一致、数据定义的不一致、过时的数据等问题。这些数据都是一些脏数据。在脏数据基础上不可能建立一个良好的挖掘模型。在数据预处理中,有时还需要对数据进行分组,以提高数据挖掘的效率、降低模型的复杂程度。

在数据预处理完成后,有时还要建立一个数据挖掘库,因为操作性数据库和共有数据仓库中所提供的数据格式并不满足数据挖掘的需要,而且数据挖掘的进行还可能影响到其他系统的应用。数据挖掘过程中的数据选择与预处理是组成数据准备的核心。在这些步骤中所花费的时间或精力要比其他步骤的总和还多。在数据准备和模型建立过程中可能反复多次,因为在建立模型过程中可能又会发现新的问题,对这些新问题的解决又需要修改数据。一般而言,在数据准备阶段中可能花费全部数据挖掘过程50%~90%的时间和精力。

3. 挖掘模型的构建

将数据转化成 一个分析模型,这个分析模型是针对挖掘算法建立的。建立一个真正

适合挖掘算法的分析模型，是数据挖掘成功的关键。

模型的建立必须从数据的分析开始，首先为模型选择变量。理想的情况是将拥有的全部变量加入数据挖掘工具中，找到那些最好的预示值，但在实际操作中，这是非常棘手的，其中一个原因是建立模型的时间随着变量的增加而延长；另一个原因就是盲目性，许多无关紧要的变量将被加入，却很少、甚至不能提高预测能力。

接着，从原始数据中构建新的预示值。例如，使用债务-收入比预测信用风险能够比单独使用债务和收入产生更准确的结果，并且更容易理解。

下一步，就需要从数据中选取一个子集或样本来建立模型。使用所有的数据会花费太长的数据挖掘时间或者需要性能更高的计算机。对大多数数据挖掘问题来讲，使用经过随机挑选的数据子集不会引起信息不足。建立模型的两种选择是要么使用所有数据建立少数几个模型，或者建立多个以数据样本为基础的模型。后者常能建立更准确的、性能更好的模型。

最后，需要转换变量，使之和选定用来建立模型的算法一致。

模型的建立与研究业务有关，如果研究目标是了解客户在接收促销广告后的 0~2 周、2~3 周、3~6 周及 6 周以上不同时间段内的反应情况，从而帮助促销人员缩短客户反应的时间，了解各种可能妨碍客户反应的因素。显然，这里所建模型的目标就是要反映客户在 0~2 周、2~3 周、3~6 周及 6 周以上时间内对广告反应的各种相关因素。模型建立后，需要从模型的准确性、可理解性和性能方面进行考察。

在模型的建立过程中还应该注意到模型的建立是一个迭代循环的过程，需要研究各种可供选择的模型，从中找出一个最能解决目前商业问题的模型。在寻找模型的过程中，所获悉的知识或许要求重新修改正在使用的数据甚至修改问题的定义。

如果数据挖掘的应用项目基于一种有监督（教师）学习的方式中，就可利用有监督学习的特点，在建立模型时由教师提供一定的帮助，例如预先定义类别，提供属于这些类别的正例和反例。具体地说，在建立模型之前就有来自以前所建立模式的历史数据，这些数据与现在使用的数据非常相似。或者要将数据分为两组，第一组用来训练或评估模型，接着使用第二组数据来测试模型。当训练和测试周期完成之后，模型也就建立起来了。

（1）模型的准确性

在模型建立以后，要对模型进行评价，评价模型的指标主要是模型的准确性和模型的可理解性。就模型的准确性而言，数据挖掘不能完全替代统计分析，数据挖掘模型的准确性一般需要通过时间来检查。统计分析往往会遗漏被数据挖掘所发现的重要关联关系，利用数据挖掘工具可以发现为什么有的客户需要 6 周以上时间才能对广告作出反应的原因。这种原因在一个给定出错范围内正确性的百分比，则往往需要统计分析来完成。

（2）模型的可理解性

模型的可理解性往往需要从多方面进行考察。首先要使数据挖掘人员了解不同的输入对结果产生什么影响。从单纯的数据挖掘模型来说,神经网络模型的可理解性一般较差,而决策树的可理解性最强。其次,模型应该能使数据挖掘人员了解预测为什么会成功或为什么会失败,如果模型能够提供最终数据挖掘结果分析报告,那将有助于对模型的理解。接着,模型应该能对复杂数据集产生预测结果,在这一方面,决策树的能力较差。最后,模型还应能够对模型所产生的结果进行检测,即模型应该提供将预测数据与已知结果进行比较的功能。

(3) 模型的性能

模型的性能主要由模型的构造速度和从模型中获取预测结果的速度来确定。一般情况下,神经网络的模型构造速度较慢。

4. 数据挖掘

对所得到的经过转化的数据进行挖掘,除了完善与选择合适的算法需要人工干预外,数据挖掘工作都由挖掘工具自动完成。

5. 结果分析

当数据挖掘出现结果后,要对挖掘结果进行解释并且评估。具体的解释与评估方法一般根据数据挖掘操作结果所制定的决策成败来定,但是管理决策分析人员在使用数据挖掘结果之前,又希望能够对所挖掘的结果进行评估,以保证数据挖掘结果在实际应用中的成功率。因此,在对挖掘结果进行评价时,可以考虑这样几个方面的问题:首先,用建立模型相同的数据集在模型上进行操作所获结果要优于用不同的数据集在模型上的操作结果;其次,模型的某些结果可能会比其他预测结果更加准确;最后,由于模型是以样本数据为基础建立的,因此实际结果往往要比建模时的结果差。而且应该注意到可视化技术是一种良好的结果分析工具,在许多情况下,利用可视化技术可将数据挖掘结果表现得更清楚,更有利于对数据挖掘结果的分析。

6. 知识的应用

数据挖掘的结果经过业务决策人员的认可,才能实际利用。要将通过数据挖掘得出的预测模式和各个领域的专家知识结合在一起,构成一个可供不同类型的人使用的应用程序。也只有通过对挖掘知识的应用,才能对数据挖掘的成果做出正确的评价。但是在应用数据挖掘成果时,决策人员所关心的是数据挖掘最终结果与用其他候选结果在实际应用中的差距。如果结果是根据某种类型的得分或权值计算的,那就可以按照获选边际率 $= (\text{最终结果得分} - \text{候选结果得分}) / \text{最终预测结果得分} \times 100\%$ 的公式进行决断。一般情况下,获选边际率的值越高,则预测结果为真的可能性越大。因此,在实际决策应用中,

通常只选择那些获选边际率超过一定百分比的数据行进行预测使用。

为将数据挖掘结果能在实际中得到应用，需要将分析所得到的知识集成到业务信息系统的组织机构中去，使这些知识在实际的管理决策分析中得到应用。

9.3.2 数据挖掘的用户

如果从数据挖掘的过程看，不同的数据挖掘过程需要不同专长的人员，大体有业务分析人员、数据分析人员和数据管理人员。

业务分析人员或称为企业管理顾问。要求这些人员精通业务，能够解释业务对象，并且能够根据具体业务对象要求确定用于数据定义和挖掘算法。

数据分析人员——要求这些人员精通数据挖掘分析技术，且对统计学能够较熟练的掌握，有能力把业务需求转化为数据挖掘的各步操作，并且能为每步操作选择合适的技术。

数据管理人员——这些人员需要精通数据管理技术，能从数据库或数据仓库中收集数据挖掘所需要的数据。

由此可见，数据挖掘项目的实施是一个不同类型专家合作的过程。这一过程需要反复进行，才能不断地趋近事物的本质，不断优化问题的解决方案。

从数据挖掘的过程看，数据挖掘的用户有业务分析员、数据分析员和数据管理员。但是实质上，这些人员通过对数据挖掘的应用，在帮助企业的高层管理人员进行管理决策，也就是说数据挖掘的真正受益者是那些企业的高层管理人员，只有他们才要了解那些能够带来竞争成功的原因。这些原因的了解，需要通过对企业以往的经营状况分析才能获取。通过这些分析，可以获取商业知识，用于企业经营，对企业的产品、市场、价格与服务进行调整，增加企业经营战略成功的机会。

目前，企业在其日常管理中越来越依赖于信息的处理与分析结果。管理人员使用信息的处理结果（查询或报表等处理）是很简单的。但是使用的效果依赖于企业的商业分析员或管理顾问，商业分析员或管理顾问针对管理人员所需要解决的问题，对特定的数据进行访问处理，用数据去验证问题的结论，这种信息应用模式是分析员驱动的模式。在数据仓库应用中的联机分析处理（OLAP）中，管理人员可以利用多维数据库或数据立方体来了解客户的行为，或生成市场分析报告。这种信息的分析应用，需要在企业管理顾问的支持下才能实现。这种信息应用也可称为广义的数据挖掘应用。由数据驱动的数据挖掘也需要用户为数据挖掘指定具体的挖掘范围。

9.4 数据挖掘的应用范围

可以这么说，数据挖掘的应用是极其广泛的，只要有数据的地方，基本上都有数据

挖掘的用武之地。针对特定领域的应用,人们开发许多专用的数据挖掘工具,包括生物医学、DNA 分析、金融、零售业和电信业等。这些数据挖掘将数据分析技术与特定领域知识结合在一起,提供满足特定任务的数据挖掘解决方案。在过去的 10 年内,人们开发了许多数据挖掘系统和产品。选择一个满足自己需要的数据挖掘产品,需要的是多角度考察各种数据挖掘系统的特征。其中包括数据类型、系统问题、数据源、数据挖掘的功能和方法,数据挖掘系统与数据库或数据仓库的耦合性,可伸缩性,可视化工具和图形用户界面。

数据挖掘的应用面是非常广泛的,它的应用不仅限于某个行业。每个行业都可利用数据挖掘技术,挖掘可能隐藏在数据中的知识。一般而言,大部分组织都可利用数据挖掘完成以下的任务:发现知识、数据可视化和纠正数据。

发现知识——知识发现的目的,是从公司数据库中存储的大量数据中找出隐含的关系,模式或相关性。

数据可视化——数据分析人员应该对要处理的存储在企业数据库中的大量信息有所了解,在分析之前,必须先使大量的数据“人性化”,以一种良好的方法来显示数据。

纠正数据——在合并大规模数据库时,许多企业发现数据是不完整的而且通常包含矛盾和错误的信息。数据挖掘技术能够以最可靠的方式来识别和纠正这些问题。

商业数据库正在以一个空前的速度增长,商业领域出现了大量的业务数据。分析这些数据也不再是单纯为了研究的需要,更主要的是为企业经营决策提供真正有价值的信息,从而获得竞争优势。所以,企业必须利用数据挖掘技术从大量的数据中进行深层的分析,获得有利于商业运作、提高竞争力的信息,为企业的经营决策提供最大限度的支持。

通过表 9-2 数据挖掘演变的比较,可以看到数据挖掘逐渐演变的过程。从商业数据到商业信息的进化过程中,每一步的前进都是建立在上一步基础之上的。并且可以看出,第四步的前进是革命性的。因为从用户的角度来看,这一阶段的数据库技术已经可以快速回答经营管理上的很多问题。

表 9-2 数据挖掘演变表

数据挖掘阶段	技术/方法	主要厂商/产品	提供的信息类型
数据搜集 (20 世纪 60 年代)	计算机、磁带、磁盘	IBM, CDC	提供历史性的、静态的数据信息
数据访问 (20 世纪 80 年代)	关系数据库 (RDBMS), 结构化查询语言 (SQL), ODBC	Oracle, Sybase Informix, IBM, Microsoft	在记录级提供历史性的、动态的数据信息
数据仓库、决策支持 (20 世纪 90 年代)	联机分析处理 (OLAP)、多维数据库、数据仓库	Pilot, Cornshare Arbor, Cognos, Microstrategy	在各种层次上提供回溯的、动态的数据信息
数据挖掘 (正在流行)	高级算法、多处理器计算机、海量数据库	Pilot, Lockheed IBM, CCI 其他初创公司	提供预测性的信息

数据挖掘在商业决策支持中表现出极其广泛的应用前景，可以帮助组织在商业决策活动中做出合理的决策。

9.4.1 客户的细分应用

1. 背景

客户的细分是将一个大的消费群体划分成一个个细分群体的过程。同属一个细分群体的消费者彼此相似，而隶属不同的细分群体的消费者是彼此不相同的。

细分可让经营管理者在比较高的层次上查看整个数据库的数据，也可以使经营管理者以不同的方法处理不同细分的群体客户。

2. 客户细分的不同用途

(1) 营销策略的制定

在市场营销活动中，企业不可能将有限的产品推广资源撒向所有的客户。企业必须了解各种客户群在市场活动中的变化，知道不同客户群对产品价格敏感性，知道产品价格的变动将引起客户群怎样变化。这对企业的营销策略制定是十分重要的，企业可以据此制定正确的营销策略，以获得更大的盈利。

(2) 企业客户群的分析

利用人口统计学细分，了解企业的客户住在哪里、有多大的购买能力、受过多少教育等等。至少知道他们住在哪里，可以帮助管理者在众多媒体中选择接触到他们的方式。

(3) 客户行为的分析

通过客户心理学的细分，可以知道客户在想什么和为什么这样想，了解客户的心理和行为构成后，能给企业关于客户行为将发生怎样变化的重要提示。这将有助于企业对流失客户的防范，对盈利客户的挖掘，对新客户的开发。

(4) 竞争态势分析

市场竞争中的决定力量是客户群的大小。企业可以利用客户群的细分，了解企业各种产品可能所拥有的客户数与竞争对手产品所拥有的客户数，以此判断企业在市场竞争中与竞争对手相比较，究竟在哪些产品中占据优势，在哪些产品中处于劣势。

3. 客户细分的方法

在开始为公司建立和配置一个新的客户细分蓝图前，先要决定这个细分蓝图究竟用于战略计划和沟通，还是只为某个目的或市场推广活动的。细分的目的对细分技术没有什么影响，但对蓝图的配置有很大差别。

进行客户细分的步骤主要有有关数据的收集，设计细分方案，验证细分方案。

客户细分中常见的问题有系统是否受现有客户数据影响？或者它只是依靠自己，未做任何修改，以更好地反映客户固有的差别？

系统是否能够全面反映企业战略目标和计划？如果细分过于复杂，没有人能够理解它、描述它或无法和别人谈论它，显然它对于战略目标不能发挥作用。

细分方案能否运用到企业现有数据上？细分方案的企业数据有意义吗？如果企业的客户群都是 12 岁以下的儿童，那么使用对全国人口作为分类细分蓝图是不可能发挥作用的。

反之，细分方案有没有可用的数据，或者只是基于很小的样本库？细分蓝图是如何建立起来的？它是否只建立在和几百人交谈的结果上，然后映射到一个大的数据库中。此时，虽然蓝图创建的成本低廉，但是它可能带来实际应用的局限性和可变性。

当数据量增大时，选择相关的人口调查属性的筛选条件是比较困难的。当客户数量不断增长和每位客户的细分因素增多时，得出这样的行为模式的复杂程度也同样增大。随着近几年客户数据库规模膨胀，用手工对潜在客户群进行市场细分几乎是不可能的。

数据挖掘技术可以帮助企业完成对潜在客户的筛选工作。通过由数据挖掘技术得出的潜在客户名单和一些客户感兴趣的优惠措施系统结合起来，是市场营销决策的需要。

9.4.2 客户盈利能力分析

1. 背景

客户盈利能力分析是数据挖掘的基础。数据挖掘技术通过帮助理解和提高客户盈利能力，使企业在市场竞争中获取优势。如果没有评价客户盈利能力的尺度，就无法使客户盈利能力达到最大化。通常在建立数据挖掘应用系统的时候，这条规则很有可能得不到满足，因为在启动这些数据挖掘的项目和应用时并没有完全理解整个系统的目标是什么。若要提高投资回报率，总要保持和客户之间的关系才能获取更大的利润。这是数据挖掘系统或任何决策支持系统的第一步，也是最重要的一步，同时也是很难的一步。

2. 为什么要计算客户盈利能力

在现实的商业竞争中，如果不知道客户的价值，就很难判断什么样的市场策略是最佳的。企业如果不知道能从客户那里获取多少利润，就有可能正在浪费投资；如果不知道什么样的客户是有价值的，也不知道能够从正在给你的客户提供服务中从竞争对手那里争夺过来多少客户，也不知道每位客户的盈利能力，那么市场营销策略就会非常盲目。客户群中，每个客户的盈利能力并不都是一样的，存在着很大的区别，如同客户对市场营销手段的反应有很大差别一样。各种市场广告和促销活动的花费可能基本上差不多，可是对不同的客户会产生积极或消极的影响。一般情况下，在客户身上的花费越多，他

们保持更高的忠诚度和购买更多产品的可能性就越大。

3. 如何进行数据挖掘, 使得客户盈利能力达到最大化

数据挖掘技术可以用来预测在不同的市场活动情况下客户盈利能力的变化。一个企业惟一需要做的一件事情就是基于市场营销策略, 预测盈利能力。首先, 企业需要设定一些优化的目标。设定优化目标的意思就是企业必须确定一种计算客户盈利能力的方法。这可以是一种简单的计算公式, 如从每位客户身上获得的收入减去提供产品、服务、市场活动、促销活动的成本, 再减去通常由客户所负担的那些固定费用。也许是一个更复杂的公式。然后从客户的交易记录中发现一些行为模式, 且用这些行为模式预测客户盈利能力的高低。做到这一点必须有两个要素: 一个是记录潜在客户的行为特征和发展成为客户行为特征的历史数据, 另一个则是计量客户盈利的标准。如果没有过去的历史记录, 数据挖掘系统就没有用来进行训练和分析的目标。如果没有计量盈利的标准, 数据挖掘系统也就没有优化计算的目标。

4. 注意事项

在进行客户盈利能力分析的时候, 应该注意以下几个问题。

(1) 忠诚度在客户盈利能力上的作用。有效的客户关系管理带来的最大收益就是改进客户忠诚度。如今, 提供有竞争力的产品是取得商战胜利的必要条件, 而获得一个忠诚的客户基础, 企业所需要做的远不只是以客户满意的方式提供有竞争力的产品。忠诚度意味着客户不断地回来找你, 购买你的产品或服务, 即便你没有最好的产品、最低的价格或最快速的交付手段。良好的关系建立在一段时间内的同客户发生的所有交互行为之上, 它带来客户价值和明显的公司收益。客户获得的全部价值不仅包括他们获得的产品或服务, 也包括获得该产品/服务的方式。那些能将两方面都做到最好的公司常常是其领域的佼佼者, 他们获得更多的市场份额和利润。

(2) 在谈论客户价值时, 一般指客户的盈利能力。当然, 也计算客户带来的收入。但是讨论客户收入的问题时, 它并不能指出哪些客户是真正重要的。一个客户可能带来许多营业收入, 但是他需要许多细致的服务, 即需要很高的花费。这样, 有时值得去做, 有时就不值得去做。收入很高, 但是盈利可能很低, 甚至是负的。有时候, 从发展的总体策略角度来看, 投资发展这类客户是值得的。在较短的时间内, 可为争取市场份额和巩固市场地位而放弃盈利, 但是长期的把钱花在这类用户上也许并不明智。即使企业在市场中处于主导地位, 市场的竞争性仍然很难将这些无利可图的客户转变成有利可图的客户。因此, 一开始就将注意力集中在有利可图的客户上是比较明智的。

(3) 数据挖掘技术一般在工商业中得到广泛应用, 它可以从充足的数据中挖掘出准确和有用的信息, 帮助解决一些问题。这些问题常与大量的客户有关。从而数据挖掘技

术在工商业中的应用也就和大量的客户数据相关。所以，用客户盈利能力指标可以很好地评估数据挖掘系统发挥的价值。

9.4.3 客户的获取与保持分析

对大多数行业来说，企业的增长需要不断地获得新的客户，保持原有的客户。新的客户包括以前没有听说过该企业产品的人、以前不需要该产品的人和竞争对手的客户。利用数据挖掘能够辨别潜在客户群，从竞争对手那里夺取客户并且保持自己的客户不被竞争对手夺取。

1. 获取客户的方法

在大多数商业领域里，业务发展的主要指标里都包括新客户的获取能力。新客户包括发现那些对你的产品不了解的客户，他们可能是你的产品的潜在消费者，也可能是以前接受竞争对手服务的客户。其中有些客户可能以前是你的客户，可能掌握了较多有关他的信息，这存在着有利的一面，当然也存在着不利的一面，他们可能是因为服务太差而转向其他商家的。在各种情况下，数据挖掘技术都可帮助我们对潜在的客户进行细分，并且提高市场推广活动的反馈率，提高客户的响应率。

一些传统的获得客户的方法，比如开展一次大规模的联合活动（杂志广告、户外广告牌），或者根据所了解的目标客户群情况进行直销活动（电话推销、邮寄广告等），都是企业目前采取较多的获取客户的方法。在这样的市场策略下，利用数据挖掘技术获得新客户的方法和大规模销售的策略相似。企业只要选出一些感兴趣的人口调查属性，这和大众市场广告时用的一些属性相同，然后寻找符合这些特征的客户名单即可。这些客户的名单可从有关的客户信息库中获取。

2. 数据挖掘在获取客户中的作用

在发展新客户的过程中，应用数据挖掘技术的目的是建立一个预测分析模型。但是，企业对于那些还不是他的客户的人的了解程度肯定远没有对现有的客户了解程度高，关键在于寻找那些已知信息和想要得到的行为模型之间的关系。首先，企业必须获得一些潜在客户的名单，在潜在客户名单中列出可能对你的产品和服务感兴趣的消费者信息（姓名、性别、年龄、职业、消费习惯和广告反应等）。接下来，企业要做的就是通过一些小规模的实验活动，收集分析用的数据。当有了实验活动中取得的反馈数据后，企业就可以对客户的反应模式进行实际分析。在这个阶段中，挑选一些需要预测的而且对企业感兴趣的行为模式，并且决定在什么样的粒度上进行分析。一旦原始数据准备好以后，就可进行数据挖掘的工作。数据挖掘软件将依据所选择的反应模式的类型预测一些指标变

量,通过这些指标变量,可以找出那些对企业所提供的服务感兴趣的客户。

3. 利用数据挖掘保持客户

随着行业竞争的越来越激烈和获得一个新客户的开支越来越大,保持原来的客户工作就越来越有价值。

在实际工作中利用数据挖掘工具为已经流失的客户建模,识别导致他们转移的模式。然后就用这些模式找出当前客户中相似的背叛者,以便企业针对客户的需求,采取相应措施防止这些客户的背叛。客户一般情况下分为三类:第一类是无价值或低价值的客户;第二类是不会轻易走掉的有价值的客户;第三类是不断地寻找更优惠的价格和更好服务的有价值客户。传统的市场活动是针对前两类客户的,而实际上特别需要用市场手段来维护的客户是第三类客户,这样做会降低企业运营成本。

在获取客户的分析决策中,可用关联分析来获取新客户,在保持客户的决策分析中,可以采用序列统计挖掘,了解客户的消费或忠诚度的变化,以此来保持客户。

9.4.4 市场营销中的应用

1. 商品促销

当今的市场竞争不论在哪个行业中都是非常激烈的,市场促销活动此起彼伏。这些促销活动能否成功,在很大程度上取决于正确的信息。这些信息包括客户保持购买趋势分析和商品销售趋势预测等。只有充分了解客户,才能正确定位,提高客户响应率,降低促销活动的成本。

企业利用数据挖掘技术可以通过从销售记录中挖掘关联信息,可以了解某些商品具有关联销售的可能性,就可以向已经购买相关商品的客户推销关联商品,这将获得极高的成功率。

企业还可以对各种促销活动前后商品销售情况进行多维分析,了解商品在促销活动中由于哪些商品的降价而导致商品销售的联动上升。这种分析的目的在于使企业不仅可以利用商品的促销活动增加商品的销售数量,更加重要的是使企业在商品促销的活动中保持企业的盈利水平,甚至使盈利水平得到提高。

2. 一对一营销

一对一营销不只是每逢客户生日或纪念日时给他寄一张贺卡。在科技发展的今天,每个人都可拥有一些自己独特的商品或服务,比如按照自己的尺寸做一套很合身的衣服。当然实际市场的营销不只是裁剪一套合适的衣服,在现实存在许多一对一销售的成功案例,航空公司、旅店超市等都可以进行客户的一对一销售。但是,这些一对一销售如

果没有数据挖掘技术的支持是很难做好的。企业可以利用数据挖掘中的分类与聚类技术把大量的客户分成不同的类,每个类里的客户拥有相似的属性。这样,企业完全可给每个不同类客户提供完全不同的服务,提高客户的满意度。细致而切实可行的客户分类对企业的一对一营销经营战略具有重要的作用。

3. 交叉销售

交叉销售是指企业向原有客户销售新的产品或服务的过程。交叉销售是建立在双赢的基础之上的,客户因得到更多更好符合其需求的服务而获益,企业也因销售增长而获益。在企业所掌握的客户信息中,尤其是以前购买行为的信息中,可能正包含着这个客户决定下一次购买行为的关键因素。数据挖掘可以帮助企业寻找影响客户购买行为的因素。

例如,那些购买了婴儿纸尿裤的客户会对其他婴儿用品很感兴趣;而有的交叉销售则是向客户提供与他们已经购买的产品或服务相关的新产品或新服务。又如,电信公司向已经使用标准长途电话服务的客户推销优质长途电话服务。

在对交叉营销进行分析时,具体的数据挖掘过程有建模、预测模型和对预测结果进行评分。

建模过程是用数据挖掘的一些算法对数据进行分析,产生一些数学模型。这些模型可以用来对客户将来的行为进行预测分析。在交叉营销分析中,对每一种情况都需要建立一个模型。而且建模阶段又可分为几个子过程,也就是说,对每种交叉营销进行分析的过程都是独立的。在分析时,各种情况所针对的客户可能会有重复,但实际上建模时都被独立处理。在用预测模型对数据进行分析以后,需要对预测结果进行评分。评分的主要目的是决定向客户提供哪一种交叉营销服务最合适。通过使用一些筛选条件对所有的客户进行一遍挑选、评分,从而决定服务的方式。

9.4.5 数据挖掘的其他应用

1. 数据挖掘与供应链管理

利用数据挖掘还可实现对供应链的优质管理。供应链管理的核心,是在生产商、供应商、分销商、零售商和最终客户之间,通过实现供应链环节中各企业的信息沟通、数据互换和协同工作,改造和整合企业的内部和外部业务流程,实现整体上更为高效的生产、分销、销售和服务活动,通过缩短交货周期、降低周转库存、缩小客户响应时间,增加企业的盈利能力。利用数据挖掘技术可以有效地描绘供应链的变化情况,了解供应链中哪些环节可能发生问题,及时对不协调环节进行处理,保持供应链的畅通。

2. 产品质量的保证

产品质量的保证主要在于产品的设计阶段，只有利用良好的设计才能保证产品质量。遗憾的是，在产品的设计过程中设计人员需要面对的是成千上万的质量数据。他们需要明确在这些因素中哪些是影响产品质量的关键因素，这些因素应该怎样控制。利用数据挖掘技术可以建立产品质量的控制模型，在产品的设计因素与产品质量之间可以用聚类、分类、关联规则、回归、相关分析、序列分析等数据挖掘模型，找出影响产品质量的主要因素以及控制条件。使产品质量得到有效的控制。

3. 风险评估和诈骗检查

风险评估与诈骗检查几乎在每个行业中都会遇到，利用数据挖掘中的神经网络分析模型可以探索具有诈骗倾向的客户，这就有可能使企业对这些客户加强监控，防止诈骗的发生。在数据挖掘中的孤立点分析，也可识别那些具有诈骗倾向的客户。

4. 犯罪活动的侦察

数据挖掘不仅可在商业活动中得到应用，而且还可以在犯罪活动的侦察中发挥重要作用。例如，利用聚类分析工具可将有关的案例进行分组，从中找到破案的线索。利用孤立点分析工具，可以探索异常资金的转移活动后面可能有洗黑钱活动的存在。利用关联分析工具可以识别不同人与相应活动的联系，识别某人存在犯罪活动的可能性。这些数据挖掘工具都有利于犯罪活动调查人员聚焦可疑线索，加快破案进度。



本章小结

数据挖掘技术作为基于机器学习、模式识别、统计学等领域而发展起来的从数据中获取知识的技术越来越得到人们的青睐。

数据挖掘技术可以分成比较成熟的统计类型挖掘技术、快速发展的知识挖掘技术和正处于萌芽状态的其他数据挖掘技术。这些技术由于在实际应用中显示其强大的信息处理和获取能力，正在各个部门得到广泛的应用。其中有客户的分类、客户盈利能力分析、客户开发与保持、市场营销等。

数据挖掘技术的应用先要选择良好的数据挖掘工具，在选择数据挖掘工具时需要考虑数据挖掘工具所能提供数据挖掘模式的数量、解决复杂问题的能力、操作能力、获取数据的能力、挖掘结果的提供方式和挖掘工具的鲁棒性等特性。

数据挖掘技术的引用过程一般要经历确定挖掘对象、数据准备、建立模型、挖掘结

果分析与应用几个阶段。在挖掘过程中需要业务分析人员、数据分析人员和数据管理人员的相互配合。



习题

- 9-1 从数据挖掘与数据库、统计学、机器学习的关系，讨论什么是数据挖掘
- 9-2 给出一个数据挖掘在商务处理中的应用是十分重要的，这种数据挖掘是什么类型的？具有什么功能？能否利用通常的数据查询技术或统计分析来替代？
- 9-3 在数据挖掘过程中需要涉及哪些过程？
- 9-4 在数据挖掘过程中最重要的步骤是哪些？为什么？
- 9-5 在现实中有哪些人需要使用数据挖掘技术来帮助他的工作？给出未在本章提及的几个例子。

第10章

统计类数据挖掘技术

引言

统计技术为人们所知已有 100 多年以上的时间，但是统计技术在数据挖掘中的应用却是近几年内所发生的事情。实际上，早在数据挖掘技术正式出现之前，就有大量的统计学家开始将统计技术用于对数据库中大量的信息进行处理，开始承担数据挖掘的工作。目前所用的一些经典数据挖掘技术，例如，CART 和 CHAID 都来自统计技术。此外，在数据挖掘中的概率、独立性、偶然性和过适应性等基本概念都来源于统计技术。在数据挖掘中许多实用的挖掘工具都是基于统计技术构造的。统计技术作为一种成熟的数据分析技术，在许多数据挖掘工具中得到广泛的应用。

通过本章学习，可以了解：

- ◆ 统计技术与其在数据挖掘中的应用
- ◆ 普通的统计类数据挖掘技术
- ◆ 回归类数据挖掘技术
- ◆ 聚类数据挖掘技术
- ◆ 最近邻数据挖掘技术
- ◆ 统计类数据挖掘技术在时序数据和序列数据中的应用
- ◆ 统计类数据挖掘工具
- ◆ 统计类数据挖掘工具的应用

10.1 统计分析类数据挖掘技术

10.1.1 统计与统计类数据挖掘技术

1. 统计

统计是数据搜集和描述数学的一个分支。在统计中总要涉及数据，并且常有足够多的数据使得普通人无法明了所有的数据。对于一般人而言，处理数以万亿比特计的数据，且要清楚数据的意义和从数据中归纳出模式，其难度是可想而知的。因此，必须借助于数学模型为手段，对这些数据进行归纳、推断和预测，寻找数据间的模式。所谓数学模型，就是根据社会现象的内在、外在因素变量及其相互关系，进行抽象和假设，构造一个或一组反映数量关系的数学方程式。利用数学模型，揭示事物的内部结构，分析变量之间的相互关系，进行统计推断和预测。统计推断分析一般借助统计数学模型完成，它用已有信息推断未知信息的工作过程，如用过去的资料来推测未来，利用局部资料推断总体，利用相关总体的资料进行变量间关系的推断等等。推断统计是描述统计的继续，是统计研究的深入和发展。由于各方面条件的约束，不可能也没有必要对每项统计调查，全面、系统地认识总体的全部单位，而只需要抽取少量单位的信息资料，对总体状况进行推断或估计。这就可以有效地发挥统计的作用。统计研究中的抽样推断方法、相关与回归分析方法，统计推算与预测、统计假设检验等方法，都是模型推断方法的具体表现形式。这些方法主要从样本调查的结果推算总体，包括在一定的把握条件下，对总体的数量特征做出一定区间内的推测；也可用于推断两个不同总体之间某一数量特征是否具有明显的差异，在统计假设检验中，可以得到具体的应用。

2. 统计类数据挖掘技术

统计作为数据挖掘中的一种技术应用是成功的，其原因就是因为统计技术是对同样类型问题的在同样情况下的应用，例如预测、分类和发现。

统计分析工具可以用于一系列的商业活动，例如使用统计工具进行数据分析，以寻求最佳机会，增加市场份额和利润，提高产品和服务的质量使顾客更加满意（利用全面质量管理程序），通过流水线产品制造和后勤服务的协调来增加利润。统计类数据挖掘技术已经成为目前最成熟的数据挖掘技术。这些技术可以成功地减少分析时间，有助于更好地进行决策分析。

作为统计类的数据挖掘技术还涉及一般数据库中的聚集函数、数据的度量、数据分布的图形、数据的趋势、数据的最近邻和数据的聚类等。

聚集函数中的 count, sum, avg, max, min 等，以及对数据进行度量的中心趋势、

数据的离散度和统计类的图形表示工具都已成为一些成熟的数据挖掘技术。

10.1.2 数据的聚集与度量技术

在许多数据库中都包含常用的聚集函数,例如 `count()`, `sum()`, `avg()`, `max()`, `min()` 等。这些函数在数据挖掘中主要发挥概要统计作用。`count()` 用于统计对象的个数, `sum()` 用于统计对象的总值, `avg()` 用于统计对象的平均值, `max()` 用于统计对象的最大值, `min()` 用于统计对象的最小值。

为数据进行中心趋势的度量,可以采用算术平均值。这就是一般数据库的 `avg()` 函数,在大部分的数据立方体的预计算中都保存了 `count()` 和 `sum()` 函数。此时,算术平均值就可以用 `sum()/count()` 来导出。

如果数据对象的值与某个权重有关,即值的大小需要考虑值的意义、重要性或频率,就不能简单地用算术平均值来度量数据对象的中心趋势,而需要采用加权算术平均值。

在数据对象是倾斜情况下,数据中心的度量最好采用中位数。如果数据对象已经排好序,当数据对象的个数为奇数时,中位数就是有序数列的中间值。如果数据对象的个数为偶数时,中位数就是中间两个数的平均值。

10.1.3 柱状图数据挖掘技术

总结数据的最好方法是提供数据的柱状图。在一个简单的样本数据库中,通过计算数据库中信用评价的不同发生次数,就可创建信用评价的一个柱状图。对于只有 10 个记录的简单客户信用数据库(见表 10-1),这相当容易做到;对于一个有许多条记录的数据库,例如,对一个超过 100 万数据记录的数据库,柱状图将是一种非常有用的方法,可以获得对数据库中数据的更高层次理解。

表 10-1 简单客户信用数据库表

序号	姓名	年龄	收入	信用评价	性别
1	王平	62	一般	一般	女
2	李力	53	一般	差	男
3	高洁	47	高	一般	女
4	李强	32	一般	差	男
5	李玲	21	高	优良	女
6	曾前	27	高	一般	男
7	武颖	50	低	优良	女
8	程勇	46	高	优良	男
9	牛兰	27	低	优良	女
10	高程	68	低	优良	男

利用数据挖掘工具 SPSS 的柱状图分析技术描绘表 10-1 中数据的信用评价预测属性 (见图 10.1), 该柱状图给出了各种信用评价的顾客数目。这些概要信息直观地显示该数据库中的一些重要信息, 例如信用优良的人最多。在数据库中无论有 100 条客户记录还是数亿条客户记录, 它都仅有很少的不同值。但是其他的预测属性可能有更多不同的值, 且能创建一个复杂得多的柱状图。

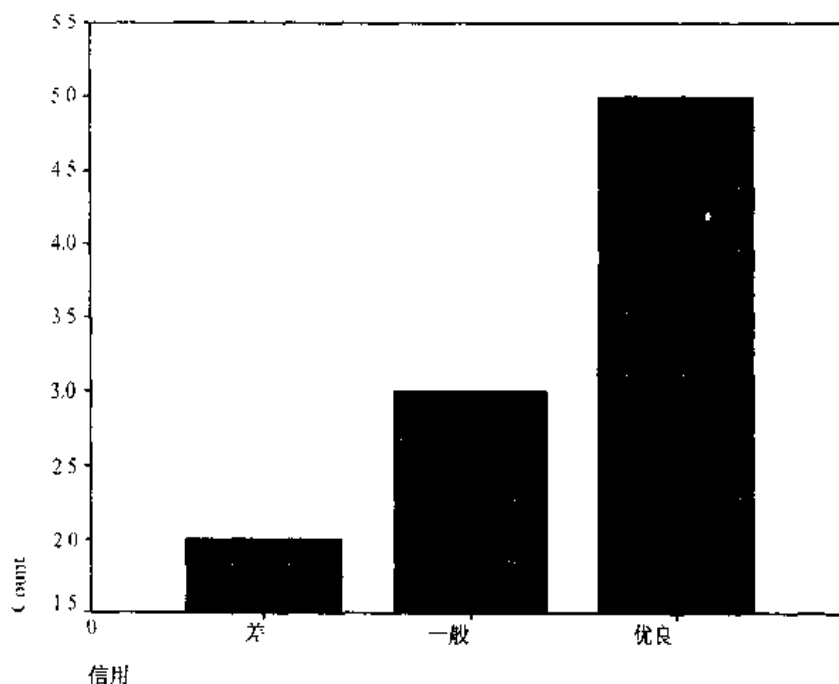


图 10.1 信用评价预测属性柱状图

10.1.4 线性回归数据挖掘技术

“回归”最初是遗传学中的一个名词, 是由英国生物学家兼统计学家高尔登 (Galton, 1822 年~1911 年) 首先提出的。他在研究人类的身高时, 发现高个子父母的子女身高有低于其父母身高的趋势; 而矮个子父母的子女身高往往有高于其父母的趋势。从整个发展趋势看, 高个子回归于人口的平均身高, 而矮个子则从另一方向回归于人口的平均身高。“回归”这一名词, 从此便一直为生物学和统计学所沿用。

回归的现代涵义与过去大不相同。一般而言, 回归是研究自变量与因变量之间关系的分析方法。其目的在于根据已知自变量来估计和预测因变量的总平均值。例如, 企业的盈利与客户数、客户购买能力和销售成本有着依存关系。通过对这一依存关系的分析, 在已知有关客户数、客户购买能力和销售成本的条件下, 可以预测企业的平均盈利水平。

在统计中有许多不同类型的回归, 但是它们的基本思想是创建的模型能够匹配预测属性中的值, 这样做预测时就会犯很少的错误。回归最简单的形式是仅包含一个预测目

标和一个预测属性的简单线形回归, 这两者之间的关系可以绘制一个二维空间: 沿着 Y 轴绘制表示预测值的记录值, 沿着 X 轴绘制预测属性值。这样的回归模型可被视为一条曲线, 该曲线用于最小化实际预测值和线上点 (从模型上得到的预测值) 之间的错误发生率, 在经过数据所画的许多曲线中, 曲线和数据点距离最小的那条曲线被选为预测模型。

一般情况下, 若要猜测曲线上的值, 在那些所有存在冲突的数据中应有一个可被接受的折中值。同样的, 如果对一特定的输入数值没有所得的数据值, 那么基于相似的数据曲线能够提供一合理答复的统计值。

线形回归是最简单的回归形式。双变量回归将一个随机变量 Y (称做响应变量) 看做为另一个随机变量 x (称为预测变量) 的线形函数, 即

$$Y = \alpha + \beta x \quad (10.1)$$

其中, 假定 Y 的方差为常数, α 和 β 是回归系数, 分别表示直线在 Y 轴的截距和直线的斜率, 这些系数可用最小二乘法求解, 这使得实际数据与该直线的估计之间误差很小。给定 s 个样本或形如 $(x_1, y_1), (x_2, y_2), \dots, (x_s, y_s)$ 的数据点, 回归系数 α 和 β 可用公式 (10.2) 和 (10.3) 计算

$$\beta = \frac{\sum_{i=1}^s (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^s (x_i - \bar{x})^2} \quad (10.2)$$

$$\alpha = \bar{y} - \beta \bar{x} \quad (10.3)$$

其中, \bar{x} 是 x_1, x_2, \dots, x_s 的平均值, 而 \bar{y} 是 y_1, y_2, \dots, y_s 的平均值。系数 α 和 β 通常给出在其他情况下复杂回归方程的较好的近似。

表 10-2 中给出一组年薪数据。其中 x 表示大学毕业生毕业后工作的年数, 而 Y 表示对应的收入。这些数据之间的线形关系可用 SPSS 的图形描述工具表现出来 (参见图 10.2)。我们用方程 $Y = \alpha + \beta x$ 表示年薪和工作年数之间的关系。

根据给定数据计算出 $\bar{x} = 9.1$, $\bar{y} = 55.4$, 将这些值代入公式 (10.2) 和 (10.3) 方程中, 得到

$$\beta = \frac{(3-9.1)(30-55.4) + (8-9.1)(57-55.4) + \dots + (16-9.1)(83-55.4)}{(3-9.1)^2 + (8-9.1)^2 + \dots + (16-9.1)^2} = 3.5$$

$$\alpha = 55.4 - (3.5)(9.1) = 23.6$$

这样, 得到方程式 $Y = 23.6 + 3.5x$ 。使用此方程, 可以预计出有 10 年工作经验的大学毕业生的年薪为人民币 58 600 元。

表 10-2 年薪数据表

x 工作年数	y 年薪 (单位: 1000 元)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83

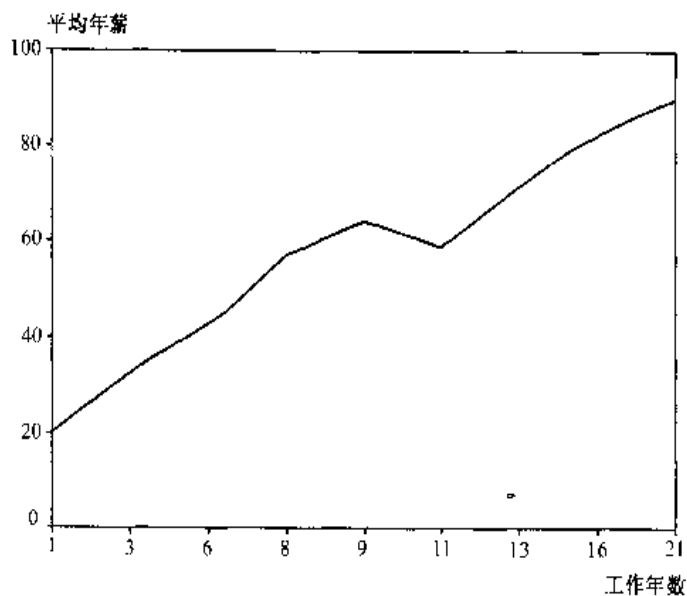


图 10.2 工作年数与年薪关系分析图

10.1.5 非线性回归数据挖掘技术

当判定变量间的关系大致是一条直线时, 可以拟合一条直线反映其变动关系, 然而在很多情况下, 变量间的关系呈曲线形式, 即非线形的, 这时就应拟合一条曲线来反映变量间的关系。例如, 给定的响应变量和预测变量间的关系可用多项式函数表示。通过对基本线形模型添加多项式项, 多项式回归可以用于建模。通过对变量进行变换, 可将非线性模型转换成线形的, 然后用最小二乘法求解。

非线性回归主要有以下 7 种模型。

1. 双曲线模型

$$y_i = \beta_1 + \beta_2 \frac{1}{x_i} + \varepsilon_i \quad (10.4)$$

2. 二次曲线模型

$$y_i = \beta_1 + \beta_2 x_i + \beta_3 x_i^2 + \varepsilon_i \quad (10.5)$$

3. 对数模型

$$y_i = \beta_1 + \beta_2 \ln x_i + \varepsilon_i \quad (10.6)$$

4. 三角函数模型

$$y_i = \beta_1 + \beta_2 \sin x_i + \varepsilon_i \quad (10.7)$$

5. 指数模型

$$y_i = \alpha \beta^{x_i} + \varepsilon_i \quad (10.8)$$

$$\ln y_i = \beta_0 - \beta_1 - \beta_2 x_i - \varepsilon_i \quad (10.9)$$

6. 幂函数模型

$$y_i = \alpha x_i^b + \varepsilon_i \quad (10.10)$$

7. 修正指数增长曲线

$$y_i = \alpha + \beta x_i^{\lambda} + \varepsilon_i \quad (10.11)$$

根据非线性回归模型线形化的不同性质, 上述模型一般可细分成如下 3 种类型。

第 1 类: 直接换元法。这类非线性回归模型通过简单的变量换元, 可以直接化为线形回归模型, 如双曲线模型, 二次曲线模型, 对数模型和三角函数模型。由于这类模型的因变量没有变形, 可以直接采用最小平方方法估计回归系数并且进行检验和预测。

第 2 类: 间接代换法。这类非线性回归模型经常通过对数变形的代换, 间接地化为线形回归模型, 如指数模型, 幂函数模型。由于这类模型在对数变形代换过程中改变了因变量的形态, 使得变形后模型的最小平方估计失去了原模型的残差平方和为最小的意义, 从而估计不到原模型的最佳回归系数, 造成回归模型与原数列之间的较大偏差。

第 3 类: 非线形型。这类非线性回归模型属于不可线形化的非线性回归模型, 如修正指数增长曲线。

10.1.6 聚类数据挖掘技术

聚类 (clustering) 是将数据对象分组为多个类或簇 (cluster) 的数据挖掘技术。聚类分析方法作为统计学的分支, 在其多年的研究中主要集中在距离的聚类分析上。这些方法已经在许多统计软件包中得到应用, 例如, SPSS 和 SAS 统计软件包中均有聚类方法。在数据挖掘中, 聚类分析主要集中在聚类方法的可伸缩性, 对聚类复杂形状和类型的数据有效性, 高维聚类分析技术以及针对大型数据库中混合数值和分类数据的聚类方法上。

1. 聚类分析原理

在进行聚类分析时, 必须用到 n 维“空间”。该空间用来定义聚类中必须解决的计量距离问题。例如, 某房产开发商对其客户数据进行聚类分析时发现, 如果按照数据中的“年龄”和“收入”两个字段值进行聚类处理 (参见图 10.3), 客户群可以分成三个主要的类别: 类别 1 是中低收入但是已经退休的老年人、类别 2 是较高收入的中年人, 类别 3 是高收入的年轻人。除此以外, 还有一部分数据散落在这三个类以外: 高收入的中年人和低收入的年轻人。

这些散落在外, 不能归并到任一类中的数据称为“孤立点”或“奇异点”。“孤立点”的数据与数据库中其他部分数据不同或不一致, 在这些“孤立点”数据中就可能隐藏着一些重要的信息。例如在“欺诈分析”中, 这些“孤立点”可能意味着有欺诈行为存在。在市场分析中则用来分析极低或极高收入客户的消费行为。“孤立点”的确定需要通过“孤立点”与类别中心距离来判断。凡是落入半径范围以内的点都归属于该类, 否则就是孤立点。

在 n 维空间中应用聚类数据挖掘时, 需要对数据之间的距离进行测量, 这种距离的测量可以采用“欧几里得距离”、“曼哈顿距离”和“明考斯基距离”。“明考斯基距离”定义为

$$d(i, j) = (|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \cdots + |x_{ip} - x_{jp}|^q)^{1/q} \quad (10.12)$$

其中 $i = (x_{i1}, x_{i2}, \dots, x_{ip})$, $j = (x_{j1}, x_{j2}, \dots, x_{jp})$ 是两个 p 维的数据对象, 即数据库中有 p 个字段的第 i 条记录与第 j 条记录。显然在利用该公式计算距离时必须对某些字段进行标准化处理, 当然有时对某些应用, 这些字段可能不需要进行标准化的处理。如果公式 (10.12) 中的 q 值为 1 时, 所得结果就是“曼哈顿距离”, q 值为 2 时, 就是“欧几里得距离”。在聚类分析中, 有的数据值根据聚类需要给予较大的权重。例如, 对流失客户进行聚类分析, 就需要对客户最后一次购买商品到现在的时间给予较大的权重。此时加权“明考斯基距离”计算公式为

$$d(i, j) = (w_1 |x_{i1} - x_{j1}|^q + w_2 |x_{i2} - x_{j2}|^q + \cdots + w_p |x_{ip} - x_{jp}|^q)^{1/q} \quad (10.13)$$

其中的 w_p 为对应的 $|x_{ip} - x_{jp}|$ 权重, 其值在 0, 1 之间, 但是所有的权重之和应为 1。同样, 加权处理也可用于“欧几里得距离”和“曼哈顿距离”的计算。

目前, 聚类方法主要有分层聚类、划分聚类、密度聚类、网格聚类和模型聚类等。

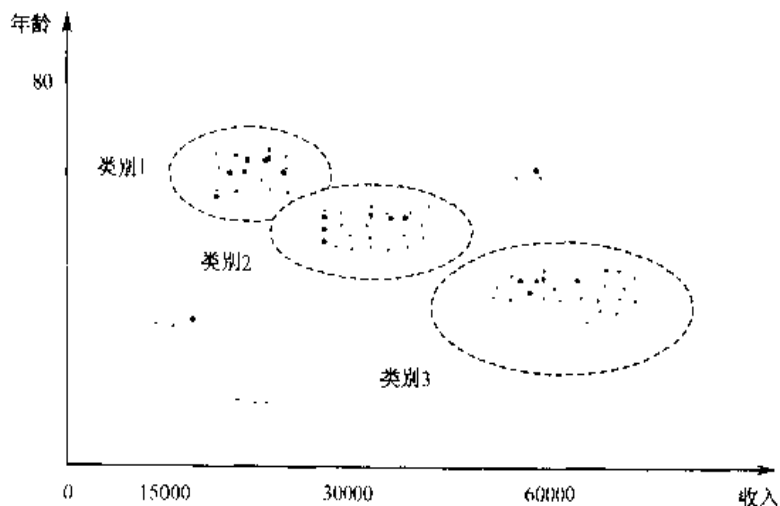


图 10.3 客户聚类分析图

2. 分层聚类

分层聚类主要有创建一个层次的聚类和另外一些部分层次的聚类两种类型。分层聚类技术是从小到大创建一个聚类的层次。这样做的主要原因是, 聚类是一种无教师学习数据挖掘技术, 因此可能没有确定的、一致的正确答案。因为这个缘故, 并且基于聚类的特定应用, 就可设计出较少或较多数量的簇。定义了一个聚类层次, 就可以选择希望数量的簇。在极端情况下, 可能有与数据库中记录数量一样多的簇。在这种情况下, 簇内的记录之间极为相似 (因为只有一个簇), 并且确实不同于其他的簇。当然, 这种聚类技术就丧失了意义, 因为聚类的目的就是发现数据库中有用的模式并且概括它, 使其更易于理解; 任何和记录一样多的簇的聚类算法都不能帮助用户更好地理解数据。这样, 关于聚类的主要一点就是应该比原先记录数量更少的簇。应当恰好形成多少簇则是解释的事情。分层聚类的好处是它们允许最终用户从许多簇或某些簇中做出选择。

分层聚类通常被看成一棵树, 其中最小的簇合并在一起创建下一个较高层次的簇。这一层次的簇再合并在一起, 就创建了再下一层次的簇。图 10.4 为客户新增与流失分层聚类图, 表明一些簇形成了一定层次。当创建像这样的一种分层簇时, 用户就可决定用多少数量的簇可以概括这些数据, 并且仍然能够提供有用的信息。在其他极端情况下, 包含所有记录的单一簇是一个很大的概括, 但并不包含足够的有用信息。

例如, 在考察不同地区的客户新增率与流失率时, 收集到了不同地区的新增与流失客户数据资料 (见表 10-3), 现利用 SPSS 的层次聚类分析工具进行层次聚类分析。在聚

类分析中, 聚类法采用了最小距离法, 距离计算准则采用欧几里得距离。在图 10.4 中 SPSS 层次聚类分析工具给出了每一步合并的是哪两类以及并类的距离, 为用户根据要求选择聚类数提供了便利。

表 10-3 新增与流失客户数据

Num	Label	新增率	流失率	num	Label	新增率	流失率
1	江苏	0.52	0.30	9	内蒙古	0.16	0.08
2	山东	0.12	0.12	10	陕西	0.36	0.10
3	广东	0.31	0.11	11	广西	0.34	0.10
4	海南	0.39	0.13	12	吉林	0.14	0.11
5	辽宁	0.10	0.12	13	湖北	0.16	0.09
6	黑龙江	0.18	0.12	14	新疆自治区	0.26	0.05
7	江西	0.46	0.14	15	浙江	0.36	0.15
8	上海	0.50	0.14	16			

***** HIERARCHICAL CLUSTER ANALYSIS *****

Dendrogram using Single Linkage

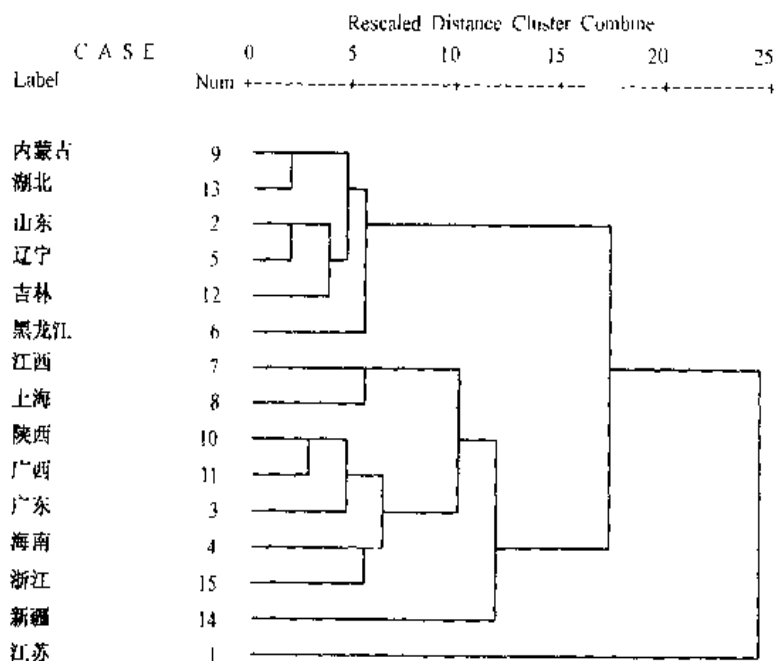


图 10.4 客户新增与流失分层聚类图

3. 划分聚类

划分聚类方法是给定一个 n 个对象或元组的数据库构建 k 个划分的方法。每个划分为一个聚簇, 并且 $k \leq n$, 该方法将数据划分为 k 个组, 每个组至少有一个对象, 每个对

象必须属于而且只能属于一个组（在有的模糊划分技术中对此要求不很严格）。该方法的划分采用给定的 k 个划分要求，先给出一个初始的划分，再用迭代重定位技术，通过对象在划分之间的移动来改进划分。

为达到划分的全局最优，划分的聚类可能穷举所有可能的划分。但实际操作中，采用比较流行的 k -平均算法和 k -中心点算法。前者，每个簇用该簇中对象的平均值表示。后者，每个簇用接近聚类中心的一个对象表示。划分的最后认可，要求同一个类中的对象之间尽可能接近或相关，而不同类之间尽可能远离或不同。

4. 密度聚类

密度聚类的思想基于距离的划分方法，只能发现球状的簇，而不能发现其他形状的簇。密度聚类则只要邻近区域的密度（对象或数据点的数目）超过某个阈值，就继续聚类。也就是说，对给定类中的每个数据点，在一个给定范围的区域中必须至少包含某个数目的点。这样，密度聚类方法就可用于过滤“噪声”孤立点数据，发现任意形状的簇。

5. 网格聚类

网格聚类方法是将对对象空间量化为有限数目的单元，形成一个网格结构。所有的聚类都在这个网格结构（即量化的空间）上进行。这种方法的优点是它的处理速度很快，其处理时间独立于数据对象的数目，只与量化空间中每一维的单元数目有关。

6. 模型聚类

基于模型的聚类方法为每个簇假定一个模型，寻找数据对给定模型的最佳拟合。一个基于模型的算法，可能通过构建反映数据点空间分布的密度函数来定位聚类。它也是基于标准的统计数字自动决定聚类的数目，考虑“噪声”数据或孤立点，从而产生健壮的聚类方法。

实际应用中的聚类工程可能包含多种聚类算法，而不是单一的聚类算法。

聚类分析的数据挖掘应用涉及市场研究，客户群区分，模式识别，数据分析和图像处理等领域。

10.1.7 最近邻数据挖掘技术

最近邻数据挖掘工具是数据挖掘技术中最容易理解的技术之一。因为它用与人们思维方式相似的方法进行分析——检测最接近的匹配样本。例如，在预测某些人的收入时，常要了解他目前处于什么阶层或获得什么学位。因为人们的收入高低往往与其所相处的人群、与他的文化程度有关，因此需要检测与其最相邻的人群。

用最近邻方法进行预测的基本概念是相互之间“接近”的对象具有相似的预测值。如果知道其中一个对象的预测值后，可以预测其最近的邻居对象。这种最近邻的概念往往和人们能将对象进行合理排序的能力有关，例如在判断汽车价格时，人们往往认为一汽的奥迪汽车与上海大众汽车的关系要比上海大众与美国大众汽车的关系更密切。这种不同对象的排序能力可以帮助人们在时空方面更好地理解现实世界。

用最近邻技术预测产品销售的市场变化时，可以采用创建训练记录的方法。例如，选择连续 10 个产品销售数据，用前面的 9 个作为预测属性值，第 10 个作为预测值。如果在产品销售的时间序列数据中有 100 个数据点，就只能创建 10 种不同的训练记录。但是采用从每个数据点开始的数据序列，就有可能创建更多的训练记录。即首先取出最前面 10 个数据点创建一个训练记录，然后从第二个数据开始选取连续的 10 个数据，创建第二个训练记录。这样，就一共可以创建 90 个不同的训练记录。

最近邻的预测需要根据最近的历史记录值进行预测，在进行预测之前必须确定数据之间的距离。数据之间的距离定义对最近邻的预测结果影响很大（见图 10.5）。如图 10.5 中记录 A、B 和 C 的信用评价情况。记录 A 的最近邻都是一些未及时支付贷款的客户记录，因为记录 A 与最近邻的记录在收入和年龄水平上有相似的特征，因此就可以将记录 A 预测为信用较差的可能拖欠贷款的客户。至于记录 B 的周围邻居既有未及时支付贷款的客户，也有及时支付贷款的客户。由于离 B 最近的是一个未及时支付贷款的客户，可以将其归并于未及时支付贷款的客户一类。

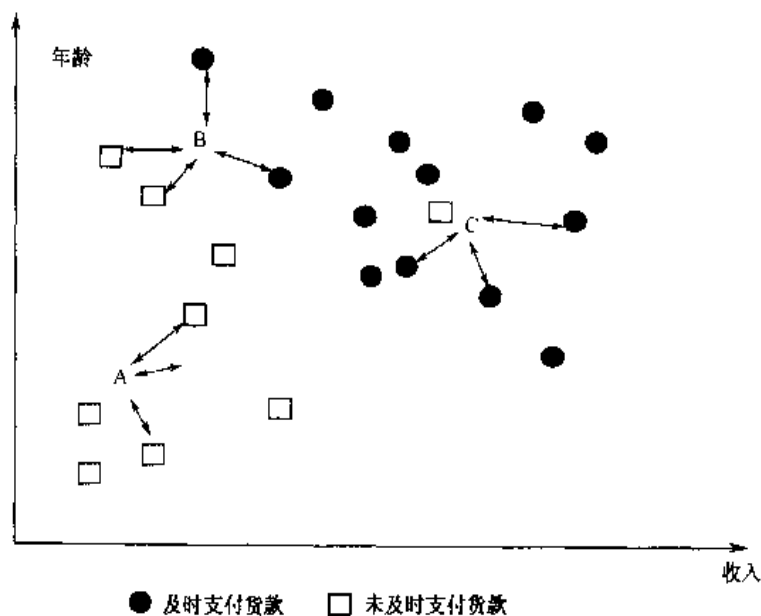


图 10.5 最近邻的预测

在有些情况下单纯依靠最近点预测，可能出现偏差。例如，图 10.5 中离记录 C 最近

的客户历史记录是一个未及时支付货款客户，按照最近邻预测就要将其归入未及时支付货款客户之中，但是 C 周围的其余客户信用记录都很良好。因此，与记录 C 最近的未及时付款点可能是一个奇异点，C 是一个良好信用客户的可能性远大于信用不良客户的可能性。针对这种情况，需要采用 9 到 15 个最近点来确定未分类记录 C 的信用状况。这种最近邻预测就称为 k 近邻方法。如果 k 个最近邻的预测值是二元的逻辑值，就按照 k 个记录的多数取值。如果 k 个最近邻预测值是多元的分类值，就可以取它们的平均值作为未分类记录的预测值。

10.2 统计分析类工具

10.2.1 统计类数据挖掘工具与商业分析员

直到最近，统计分析工具还主要为技术和工程应用中的统计员以及技术上的专家服务。但是，许多企业已经开始应用开发统计分析工具进行管理决策分析，使企业的决策分析获得成功，使统计分析工具开始为商业分析人员所采纳和应用。这些商业分析员是其业务领域的专家，但却不是程序员或统计员。他们要从数据仓库中选择恰当的数据，将它抽取出来并且进行分析。商业分析员不可能将其有限的时间和精力投入学习如何编写计算机程序、操作数据库，而构造形式化的统计分析方法和策略可能更适合他们的决策分析。商业分析员所需要的主要技巧包括某个商业领域的专门知识，具备如何进行分析以解决所遇到的商业问题的能力。

数据挖掘中的统计分析工具，是一种处于知识发现工具和信息处理工具之间的数据挖掘工具。信息处理工具的应用一般需由商业分析员来发动（参见图 10.6），大多由信息管理员来完成，而且其应用领域一般只限于信息的处理，只能对业务操作提供帮助。而统计分析工具的应用只需要对商业分析员做有限的、恰当的指导，就可启动数据的分析处理。此时，统计类数据挖掘工具可以完成信息的分析处理，并且能够进一步进行商业活动的统计分析，这比单纯的信息处理功能增强许多。当然统计类数据挖掘工具与其他的知识发现类工具相比，还缺乏更强大的知识挖掘功能。而且还不能像其他数据挖掘工具那样，完全依靠数据的驱动进行数据挖掘，还需要用户的指导。

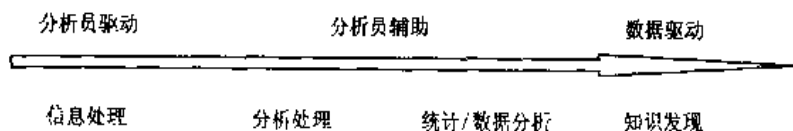


图 10.6 商业分析人员的作用

在利用统计分析工具时，商业用户必须从数据仓库或数据集市选取恰当的数据。当

商业用户抽取到合适的数据库以后，就可执行统计分析工具中的可视化功能和分析功能来寻找数据间的关系，并且构造统计模型和数学模型来解释这些数据之间的商业模式。在分析过程中，商业用户可以利用交互式过程或迭代过程对模型进行求精。其目标是开发最适应商业环境的模型，并且将数据转化成信息。在选择适应性商业环境的模型时，商业分析人员在商业领域的专业知识和解决问题的技巧是极其重要的。

10.2.2 统计类数据挖掘工具的功能

考虑到许多统计分析任务的复杂性，统计分析类工具应当提供可视化功能、探索功能、统计功能、数据管理功能、显示功能、数据挖掘描述功能和开发功能。

1. 可视化功能

数据可视化功能有助于查找大量数据之间的关系。例如，可以识别时间序列数据中的模式，也可进行曲线匹配，以发现数据中的“商业规则”或“商业模式”，还可以通过自动成组化离散值，或者通过改变直方图的始点和尺寸来操作数据。

2. 探索功能

数据挖掘工具的探索功能有助于选择适用于数据的恰当统计功能和模型。这些功能包括多维表，面向分析的求助信息；细剖，排序和数据子集；分割文件并且做示例；指明极值和冗余。工具应能动态生成恰当的图表、图形或表格，并且将其作为探索过程的一部分自动提供给商业分析员。

3. 统计和操作功能

应该提供丰富的数据统计和操作功能。例如，线性、非线性回归分析，时间序列分析（包括自动关联），快速傅里叶变换和预测；多变量分析，ANOVA，CHAID，非参数化测试和多响应分析。

4. 数据管理功能

利用数据的管理功能可为用户提供查找细节信息，浏览数据的子集，删除冗余，比较子集和数据存储格式的转换等数据操作。

5. 显示功能

这些功能可以记录分析的步骤，将记录传送给商业分析员，然后显示整个分析任务过程。记录功能应该包括分析步骤，数据集选择过程，所选图表和图形的调色板或演示功能，以及其他信息间的通信。这些功能在多用户的网络数据挖掘过程中，向用

户提供共享统计分析任务的中间结果和分析过程，是很重要的。

6. 挖掘结果描述功能

数据挖掘结果描述功能需要提供较为简单的商业图表、图形和表格形式，将数据挖掘结果表示出来，以方便复杂的数据分析和通信。这种功能应该能够很快地从图表类型中转化成数据，且可按照需要将数据显示成不同的图表；还能够将各种图表、图形和表的类型以合适的形式显示给商业用户，以使用户很容易地选择合适的表示方法。其中包括一些基本的图表和图形，例如， x - y 线以及散点绘图、框架绘图、直方图，条形图、饼图、面积图、区间绘图、三维图形和轮廓绘图、统计图表以及类似的报表。

7. 开发工具

用户利用这些工具可以很容易插入桌面应用程序和构件，以便进行统计分析，制作图表、图形和报表。面向对象编程语言以及通过类似对象链接与嵌入（OLE）技术的数据交换功能，将会增强商业分析员的能力，可将统计分析与桌面决策支持应用恰当地组合起来。

8. 可接受的响应时间

统计分析类数据挖掘工具的操作可能花上几分钟，甚至几个小时，对商业决策来说都是可接受的。当然也存在例外，例如在遇到紧急市场分析处理时，几天之后的响应将是无法接受的，因为当数据不能反映当前状况时，有可能无法进行相关分析，通往市场的大门往往也就被关闭了。

10.2.3 统计类数据挖掘工具——SPSS

SPSS（Statistical Program for Social Sciences）作为通用的统计软件包在数据挖掘领域中得到广泛的应用，其应用范围涉及经济、管理、工业、心理和教育等领域。

SPSS 11.0 在数据接口方面有了很大的改进，使数据文件大小不再受到限制。而且，利用数据库向导数据接口和连接 ODBC 功能，使 SPSS 与其他数据库的接口得到很大的改进，更加容易对数据库与数据仓库中的数据进行数据挖掘。

SPSS 11.0 在通过“开始”→“程序”→“SPSS 11.0 for Windows”命令启动后，将出现 SPSS 的主对话框（见图 10.7），也即文件打开对话框和变量定义对话框。利用文件打开对话框，可以打开 SPSS 格式文件、Excel 的表格数据文件、后缀为*.dbf 的数据库文件以及文本文件*.txt 等文件。

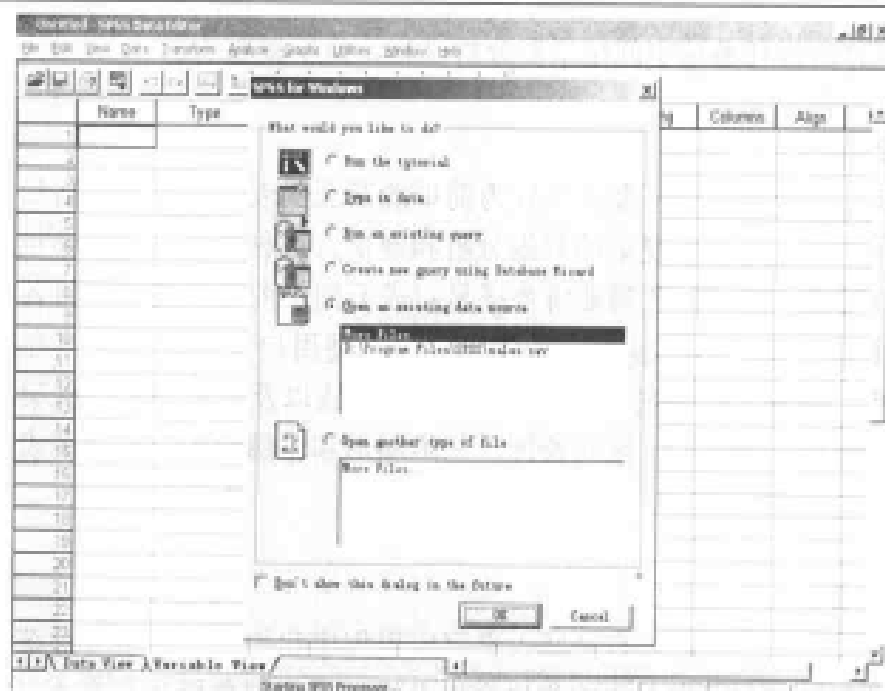


图 10.7 SPSS 的主对话框

其他格式的数据可以利用数据库向导将其格式转换为 SPSS 格式文件，数据库向导可以采用“File”→“Open Database”→“New Query”命令启动。数据库向导启动后，可在数据库向导的欢迎对话框的数据源选择框中选择有关的数据库（见图 10.8）。如果未发现所需要的数据库，可以单击“Add Data Source”按钮，进入“ODBC 数据源管理器”对话框（见图 10.9），设置所需要的数据源。

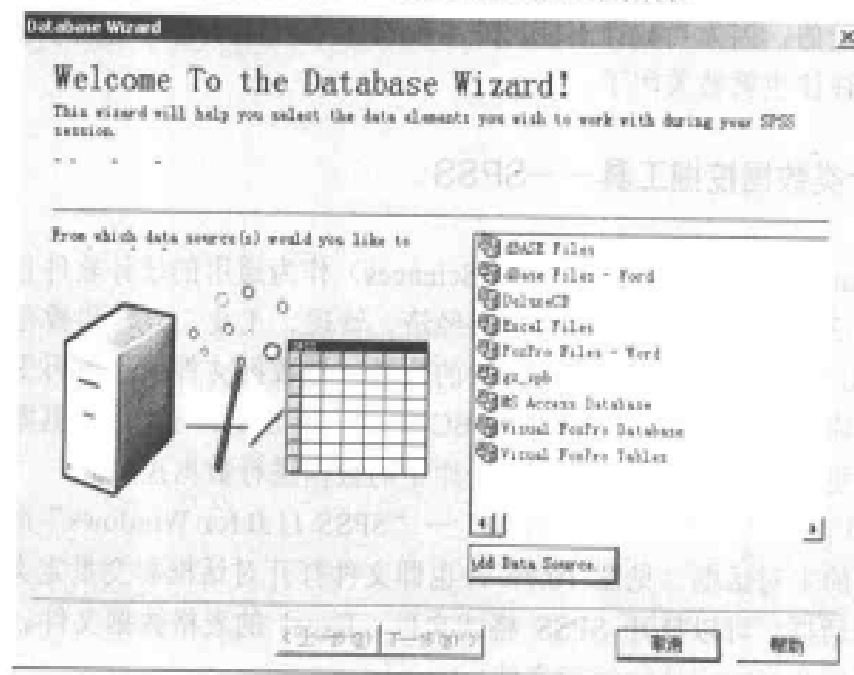


图 10.8 SPSS 的数据库向导欢迎对话框



图 10.9 “ODBC 数据源管理器”对话框

在 SPSS 中可以完成基本统计分析、回归分析、相关分析、分类分析、因子分析和非参数分析等数据挖掘工作。

1. 基本统计分析

SPSS 的基本统计分析由 Analyze 菜单下的报告分析 (Report) 和描述性统计分析 (Descriptive Statistics) 两项功能组成。利用基本统计分析，可以了解所分析数据对象的许多统计学指标，例如均数、方差、标准差、标准误差、最大值、最小值、范围、偏差、峰值以及标准误差等，并且能对数据进行正态分析、独立性检验，分析单变量数据的特性和多变量数据的相互关系。

报告分析通过命令“Analyze”→“Report”，可以启动联机分析处理 (OLAP Cubes)、观察值摘要分析 (Cases Summary)、行式摘要报告 (Report Summaries in Rows) 和列式摘要报告 (Report Summaries in Columns) 等分析。

描述性统计分析可以通过“Analyze”→“Descriptive Statistics”，启动频数分析 (Frequencies)、描述型统计量分析 (Descriptives)、探索分析 (Explore) 和多维频数分布列联表 (Crosstabs)。

2. 回归分析

在 SPSS 中可以完成线性回归分析 (Linear)、曲线回归分析 (Curve Estimation)、二维 logistic 回归分析 (Binary Logistic)、多维 logistic 回归分析 (Multinomial Logistic)、Ordinal 回归分析 (Ordinal)、概率单位回归分析 (Probit) 和非线性回归分析 (Nonlinear)

等统计分析。这些回归分析均在“Analyze”→“Regression”菜单项下启动。

3. 相关分析

在 SPSS 中的相关分析包括相关分析 (Bivariate)、偏相关分析 (Partial) 和距离分析 (Distances) 等数据分析功能。相关分析主要通过数据变量之间的密切程度, 根据样本资料推断总体是否相关。这些相关分析的启动需要使用命令“Analyze”→“Correlate”。

4. 分类分析

SPSS 中的分类分析主要有快速样本聚类 (K-Means Cluster)、层次聚类 (Hierarchical Cluster) 和判别分类 (Discriminant)。这些分类方法均在命令“Analyze”→“Classify”下。

5. 因子分析

SPSS 中的因子分析主要用于研究若干个变量 (因素) 中每个变量对某些响应的的作用。对这些因素的研究可以是单因素也可以是多因素的。在 SPSS 中用“Analyze”→“Data Reduction”→“Factor”命令可以进行因子分析。因子分析的目的是用少数几个因子去描述许多指标或因素之间的联系, 即将相互关系比较密切的几个变量归纳在同一个类别中, 每个类别就成为一个因子, 就可以用少数几个因子反映数据中的大部分信息。

10.3 统计分析类工具的用途

在数据挖掘过程中, 有时需要对对序数据库和序列数据库进行数据挖掘, 时序数据库中的数据是一些反映随时间变化的序列值或事件组成的数据库, 这些值一般是等时间间隔采集的数据, 例如证券市场每天的波动、产品加工过程的质量变化等。序列数据库也是包含序列数据的数据库, 但是它可能有时间标记, 也可能没有。例如, Web 页面遍历序列是一种序列数据, 但未必就是时序数据。统计类数据挖掘工具可以在时序数据和序列数据的挖掘中发挥重要作用, 主要是趋势分析、相似性搜索、与时间有关数据的序列模式挖掘和周期性模式的挖掘。

10.3.1 趋势分析

发生时序变化的数据通常可能出现长期的趋势变化、循环变化、季节变化以及随机

变化的倾向。

趋势变化的数据序列可以反映一般的变化方向，它的时序图是一种较长时间间隔上的数据变化。这种变化反映一种趋势，确定这种趋势的方法可以采用加权平均或最小二乘法。

循环变化数据的趋势线在一个较长的时期内呈现一种摆动变化迹象。这种摆动可能是一种完全周期性的，也可能不是周期性的，即在时间间隔之间循环不按同样的模式演变。

季节变化数据反映每年都重复出现的事件，例如，在春节前，各种商品的销售量会有一个较大幅度的增长。这种时序变化是以同一或类似同一模式，在连续几年的有关月份中重复出现。

随机变化的倾向数据时序，反映由于随机事件所引发的数据时序变化。例如，洪水对市场的一个较长时间的影响。

对于给定的一组时序数据 (y_1, y_2, \dots, y_n) 趋势的确定，可以采用移动平均值计算

$$\frac{y_1 + y_2 + \dots + y_n}{n}, \frac{y_2 + y_3 + \dots + y_{n+1}}{n}, \frac{y_3 + y_4 + \dots + y_{n+2}}{n}, \dots \quad (10.14)$$

如果在序列中使用加权算术平均，则得到加权移动平均序列值。一般对中间数据给予较大的权重，以抵消平滑的效果。趋势线的确定还可采用最小二乘法。

对于季节性波动的趋势确定，需要采用季节指数来处理，即用一组数字表示一年中某些月份某变量的相关值。例如，12 月、1 月和 2 月的销售量分别是全年平均月销售量的 110%，115%，150%，那么 110，115 和 150 就是本年度的季度指数。如果原始的每月数据由对应的季节指数去除，结果数据就是反季节变化的。利用反季节变化数据可对趋势做进一步的调整，即按照对应的趋势值去除这些数据。

对于呈现周期或类似周期的变化趋势，可以按照引入季节性指数的方法引入循环指数处理。至于随机变化数据的趋势可以针对趋势、季节、循环变化的数据调整加以估计。一般情况下，小偏差出现的频率较高，大偏差出现的频率较低，应该服从正态分布。

在数据挖掘中，可以通过对趋势、循环、季节变动或随机变动的系统分析，制定比较合理的长期或短期预测。

10.3.2 时序分析

时序分析是指在时序数据中应用所谓的相似搜索，找出与给定查询序列最接近的数据序列，主要找出与给定序列相似的所有数据序列的子序列匹配或找出彼此间相似的整体序列匹配。这些相似搜索可以用于对市场数据的分析中。

时序的相似搜索需要经过数据变换，将时序数据从时间域转换到频率域，转换方法

主要采用傅里叶变换 (DFT) 和离散小波变换 (DWT)。一旦数据完成变换, 就可提交系统, 由系统根据索引检索出与查询序列保持最小距离的数据序列。然后, 通过计算时间序列和未满足查询的序列间的实际距离, 进行必要的后处理。

在相似搜索中不一定要要求匹配的子序列在时间轴上完全一致。也就是说, 子序列只要具有同样的形状, 即使存在间隙或在偏移或振幅中存在差异, 也可以认为是匹配的。

为提高相似搜索效率, 在数据转换以后需要建立一些索引, 这些索引主要有 R-树、R*-树以及后缀树等。

10.3.3 周期分析

周期分析是针对周期模式的挖掘, 即在时序数据库中找出重复出现的模式。周期模式挖掘可以看成以一组分片序列为持续时间的序列模式挖掘, 例如, 在每年春节销售这一事件出现前后的每一天销售等。

周期模式的挖掘问题可以分成挖掘全周期模式、挖掘部分周期模式和挖掘周期关联规则 3 种。

挖掘全周期模式是指在周期中的每一时间点都影响时序上的循环行为, 例如一周中的每一天销售量都会对一周中的销售量发挥作用 (参见图 10.10)。

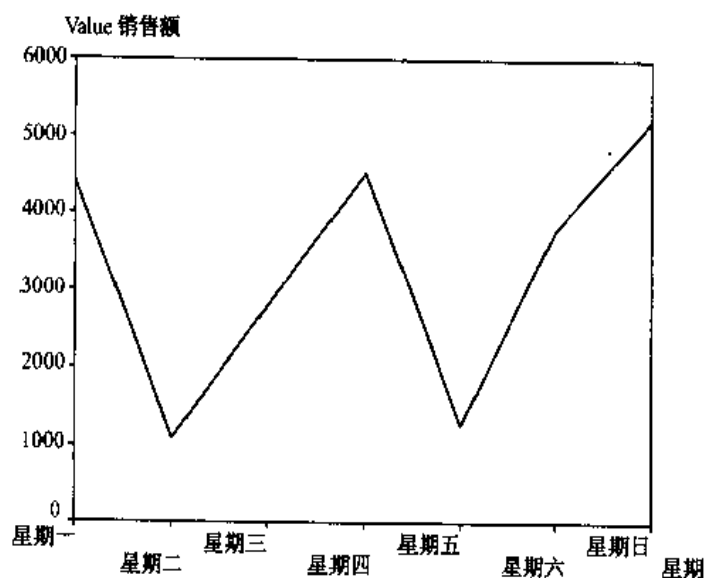


图 10.10 超市销售情况的周变化趋势分析图

挖掘部分周期模式是一种比较松散的全周期模式。这种模式在现实中是常见的, 它主要描述部分时间点的时序周期。例如, 企业的销售高峰发生在每天的上午 10:30 至 11:00, 其他时间中的销售量则没有什么规律。

挖掘周期关联规则是指周期性出现的事件的关联规则。即在某个周期中, 某个事件发生后, 将会导致另一事件的发生。例如, 某临近体育场的快餐店根据长期的营业记录可以发现, 如果每周的周末下午 3:00 至 5:00 体育场有球赛, 则快餐店的最佳营业时间在 5:30 至 7:30。

10.4 统计分析类工具应用中的问题

10.4.1 统计类数据挖掘的预处理问题

在现实世界中的数据仓库极易受噪声、空缺数据和不一致性数据的影响, 因为数据仓库太大, 常常多达数千兆字节, 甚至更多。这时就会出现这样的问题: 如何进行预处理数据才能提高数据质量, 从而提高挖掘结果的质量, 或怎样预处理数据才能使得挖掘过程更加有效, 更加容易?

存在不完整的、含噪声的和不一致的数据是大型的、现实数据库或数据仓库的共同特点。不完整数据的出现可能有多种原因。有些感兴趣的属性, 如销售事务数据中的顾客的信息, 并非总是可用的。其他数据没有包括在内, 可能输入时认为是不重要的。相关数据没有记录是由于理解错误, 或者因为设备故障。同其他记录的数据不一致可能由于被删除。此外, 记录历史或修改的数据可能被忽略。空缺的数据, 特别是某些属性上缺少值的元组可能需要推导。

数据含噪声(具有不正确的属性值)可能有多种原因: 收集数据的设备可能出故障, 人为的或计算机的错误可能在数据输入时出现, 数据传输中的错误也可能出现。这些可能是由于技术的限制, 如用于数据传输同步的缓冲区大小的限制。不正确的数据可能是由命名或所用的数据代码不一致而造成的。重复元组也会造成数据噪声, 对此也需要进行清理。

数据清理例程通过填写空缺的值, 平滑噪声数据; 识别、删除孤立点, 并且解决不一致来“清理”数据。脏数据能使挖掘过程陷入混乱, 导致不可靠的输出。尽管大部分挖掘例程都有一些过程, 处理不完整或噪声数据, 但它们并非总是强壮的。相反, 它们更致力于避免数据过分适合所建的模型。这样, 需要一个预处理步骤, 清理数据中的各种问题。

1. 空缺值处理

如果一个数据库中许多元组的一些属性值没有记录值, 可以采用以下的方法为该属性添上空缺的值。

(1) 忽略元组

如果挖掘任务涉及分类或描述,但是缺少类标号时可以忽略元组。该方法应用时,要求元组有多个属性缺少值,否则该方法不是很有效的。当每个属性缺少值的百分比变化很大时,它的性能就非常差。

(2) 人工填写空缺值

一般地讲,该方法很费时,且当数据集很大、缺少很多值时,该方法可能行不通。

(3) 使用一个全局变量填充空缺值

将空缺的属性值用同一个常数替换。

(4) 使用属性的平均值填充空缺值

例如,顾客的平均收入为 2 800 元,则用它来替换“收入”字段中的空缺值。使用与给定元组属同一类的所有样本的平均值。

(5) 使用最可能的值填充空缺值

可以用回归、基于推导的使用贝叶斯形式化方法的工具或判定树归纳确定最可能的值,将其填充到空缺值中。

2. 噪声数据处理

噪声是一个测量变量中的随机错误或偏差。给定一个数值属性的噪声,可以将其平滑掉或剔除掉噪声。

(1) 分箱

分箱方法用来平滑噪声。该方法主要通过考察“邻居”(即周围的值),平滑存储数据的值。存储值被分布到一些“桶”或箱中。由于分箱方法参考相邻的值,因此它进行局部平滑。图 10.11 是一个分箱技术应用的过程示例,价格数据首先被划分并且存入等深的箱中(深度 3)。然后,按箱平均值平滑,箱中每一个值被箱中的平均值替换。类似地,可以使用按箱中值平滑,此时箱中的每一个值被箱中的中值替换。对于按箱边界替换,箱中的最大或最小值被视为箱边界,箱中的每一个值被最近的边界值替换。一般而言,宽度越大,平滑效果越大。箱也可以是等宽的,每个箱值的区件范围是个常量。分箱也可以作为一个离散化技术使用。

(2) 聚类

数据中的孤立点噪声可用聚类检测出来。聚类将类似的值组织成群或“聚类”。直观地看,落在聚类集合之外的值被视为孤立点。孤立点值作为噪声值处理,将其删除或用“聚类”中心值代替。

(3) 计算机和人工检查结合

可以通过计算机和人工检查相结合的方法来识别孤立点。例如在一种应用中,使用信息理论度量,帮助识别手写体字符数据库中的孤立点。度量值反映被判断的字符和已

知的符号相比的“差异”的程度。孤立点模式可能是提供的信息（例如，识别有用的数据异常）或“垃圾”（例如，错误的字符）。其差异程度大于某个值的模式，输出到一个表中。人们可以审查表中的模式，识别真正的垃圾。这比人工搜索整个数据库快得多。在其后的数据挖掘应用中，垃圾模式将从数据库清除。

价格排序后的数据：4, 8, 15, 21, 21, 24, 25, 28, 34

划分为等深的箱（每箱装3个数据）：

箱1：4, 8, 15

箱2：21, 21, 24

箱3：25, 28, 34

用箱平均值平滑：

箱1：9, 9, 9

箱2：22, 22, 22

箱3：29, 29, 29

用箱边界平滑：

箱1：4, 4, 15

箱2：21, 21, 24

箱3：25, 25, 34

图 10.11 分箱技术的应用过程示例

（4）回归

可以通过让数据适合一个函数（如回归函数）来平滑噪声数据。线性回归涉及找出适合两个变量的“最佳”直线，使得一个变量能够预测另一个。多线形回归是线形回归的扩展，它涉及多于两个变量，适合多维面数据。使用回归找出适合数据的数学方程式，能够帮助消除噪声。

3. 不一致数据处理

对于有些事务，所记录的数据可能存在不一致性。有些数据不一致可以使用其他材料人工地加以更正，例如数据输入时的错误可以使用纸上的记录加以更正。这可以与帮助纠正编码不一致的例程一起使用。知识工程工具也可用来检测违反限制的数据。例如，知道属性间的函数依赖关系，可以查找违反函数依赖的不一致值。

10.4.2 统计分析遵循的基本原则

统计分析的科学依据在于事物发展的规律性。具体来说，应该遵循以下3个基本原则。

1. 与定性分析相结合原则

统计分析是一种定量分析，但不是抽象的量，而是具有一定质的量。首先，必须对现象的性质有足够的认识，在管理理论指导下对现象进行详细的分析，找到事物的内在

联系和主要的数量关系。这样,才能用恰当的数学模型进行分析。对分析的结果也应根据有关专业理论进行分析和修正。

2. 连贯和类推原则

这是进行模型外推分析所要遵循的两条重要原则。连贯指的是过去和现在的状况将会依某种规律延续到将来。它有两方面的含义:一是时间的连贯性,即分析对象在较长时间所呈现的主要数量特征保持相对稳定,以时间序列为代表的趋势外推分析正是利用时间连贯性的假定。二是结构的连贯性,即分析对象系统的结构基本上不随时间而变。各变量间相互影响的关系基本稳定,因果关系分析则是以这一假定为前提。类推原则指客观事物的结构和变化都有一定模式。同一性质、同一类型的事物,其结构变化应该有同一模式。这种模式可由数学模型模拟,将过去的情况类推到将来。类推原则是建立统计模型的理论基础。

3. 统计资料的可靠性和分析公式的适应性原则

必须保证统计资料准确、可靠和合理,才能利用观测数据找到真正的统计规律,从而建立可靠的分析模型。对于同一目的、同一批数据的分析问题来说,可以有不同的分析模型和不同的分析方法,这时要根据事物的特点及其统计规律,确定使分析误差达到最小的分析模型和分析方法,即建立最合适的分析公式。

10.4.3 统计分析的步骤

1. 确定分析目标

“凡事预则立”。这句古话说明对未来状况的分析是行动成功的关键。对社会经济现象的未来前景的估计,怎样尽可能正确,尽量减少行动决策中的风险,这正是分析所要研究的问题。每次分析之前,先要明白分析的对象是什么,解决什么问题,达到什么要求以及分析的时间、范围等。这些问题解决了,才能明确分析的具体任务。

2. 收集、审核及分析统计资料

根据分析目的,广泛收集所需资料,对资料认真审核,保证数据真实准确,且对资料进行分析、归纳和选择,剔除非正常因素的数据,找出事物发展的统计规律。确保指标口径一致可比,数据资料正确是保证分析结果准确的基础。事实上,统计数据不可靠往往会造成分析结果的偏差,甚至对分析方法的误解,这是十分重要的一环。

3. 确定分析模型、选择分析方法

统计模型用于分析时,称为分析模型。分析模型有很多种,必须根据分析的要求及

事物本身的特点,选择恰当的模型。还要选择正确的估计模型参数值的方法,即分析方法。一个分析模型可有不同的估计方法;同样,一个分析方法也适用于不同的模型。应当根据分析的目的、占有资料的数量和可靠程度、分析精度要求、分析费用等项要求,选择恰当的分析模型和分析方法。

4. 进行分析

根据选定的模型,用选定的分析方法计算出参数后,就有了据以分析的分析公式。根据分析公式对数据进行分析。

5. 误差分析

统计分析是对未来情况的估计值,由于分析模型的理论解释和假定中,考虑定量因素不完整,加之客观现象的变化,所以分析误差是不可避免的。这就是说,所求出的分析值与实际值是有一定差异的。分析模型建立并且获得分析结果后,一般要经过误差分析,如果误差过大,要从各方面分析误差产生的原因,再进行模型或参数的修正,以建立可靠的分析公式,提高分析水平。

10.4.4 统计类数据挖掘的性能问题

统计方法的优点是精确、易理解并且已经被广泛应用。许多人认为统计方法是数据挖掘最准确的形式,事实上许多数据挖掘技术都用存在已久的统计技术。一种很流行的决策树方法 CHAID 用卡方度量;关联算法使用了支持度和置信度;聚类技术使用了 K 均值算法之类的统计尺度;贝叶斯网使用 1763 年就存在的统计技术“贝叶斯概率理论”。

统计学受到的最大责难是很难有效使用,数据挖掘是从数据中抽取有价值的信息的过程,而统计学是一个完整的研究领域,包括从数据中抽取有价值信息。统计学家与想利用分析模型的其他商业人员间总是存在隔阂,许多商业人员经常无法搞清楚如何将商业问题与统计处理联系在一起。因此,有人认为数据挖掘与统计学不同,商业人员更加容易掌握数据挖掘。IBM, SPSS 和 SAS 公司一直在为打破这种观点而努力,它们将标准的统计模型和神经元、决策树以及其他与数据挖掘有关的技术结合在一起,取得了较好的效果。

统计分析是一种有力的技术,用它可以了解客户、市场、产品和其他关键商业参数。但也存在一些问题:

- 它是劳动力密集的,需要相当一部分统计分析员和商业分析员的分析劳动;
- 成功的可能性很大程度上依赖于商业分析员解决问题的能力,不能自行查找隐

藏在数据背后的知识:

- 在许多情况下, 商业分析员并不知道需要查找什么, 或者无法选择离散的变量来启动分析处理, 统计分析工具就难以承担重任;
- 在进行市场细分时, 很难集成和分析非数字化数据 (例如地理数据), 只适合数字化的数据处理;
- 一般很难以合理的成本获得可接受的响应时间。



本章小结

统计类数据挖掘技术是数据挖掘技术中比较成熟的一种, 主要包括数据的聚集与度量技术、各种回归技术、聚类挖掘技术和最近邻数据挖掘技术等。

在统计类数据挖掘技术的应用中, 需要商业分析人员给予必要的辅助指导, 因此统计类数据挖掘技术应用的成败往往取决于商业分析人员的专业水平。

在统计类数据挖掘工具中, 主要是 SPSS 和 SAS 统计软件。SPSS 统计软件利用 ODBC 可将许多数据库中的数据转换成 SPSS 文件, 可使 SPSS 的数据源直接来自各种数据库。在 SPSS 中提供了基本统计分析、回归分析、相关分析、分类分析、因子分析等数据挖掘模式。

统计类数据挖掘技术可以用于趋势分析、时序分析、周期分析等领域。在使用统计分析类数据挖掘工具时要注意防止数据受到噪声的影响, 并且遵循统计分析技术应用的一些基本原则。

在应用统计类数据挖掘工具时, 按照确定挖掘对象、收集统计数据、选择合适的统计分析模型、分析处理和分析结果等步骤进行。



习题

10-1 在某个数据库中有不同元组值是: 12, 13, 13, 15, 16, 16, 16, 19, 19, 22, 22, 25, 25, 25, 25, 28, 28, 28, 29, 31, 31, 32, 32, 32, 35, 35, 36, 36, 36, 37, 37, 39, 39, 39, 40, 41, 44, 45, 45。该系列数据的 count, sum, avg, max, min 分别是多少? 另外给出其他三个本章没有介绍的常用数据统计度量值。

10-2 给定两个对象分别用元组 (22, 1, 42, 10) 和 (20, 0, 36, 8) 描述, 计算这两个对象之间的曼哈顿距离、欧几里得距离和明考斯基距离, 明考斯基距离的 q 值为 4。

10-3 现有职工生产情况数据库, 其中包含职工两个月的生产记录 (见表 10-4)。绘制数据

图观察两者之间是否具有线性关系？利用最小二乘法，求由职工前一个月生产情况预测后一个月生产情况的公式，并且预测前一个月产量为 86 的职工下一个月的产量为多少？

表 10-4 职工两个月的生产记录

723	842
502	636
816	772
747	783
944	904
866	756
592	492
831	795
650	773
337	539
887	747
818	901

10-4 聚类是一种重要的数据挖掘技术，请讨论将聚类作为主要数据挖掘方法应用的情况，将聚类作为其他数据挖掘的数据准备情况。

10-5 在靠近一个新数据点的 20 个相邻点中，8 个属于 m 类，11 个属于 n 类，1 个属于 k 类，这个新数据点最有可能属于哪一类？

10-6 对第 1 题中的数据按照箱平滑方法进行数据平滑处理，箱的深度为 3。除本章所介绍的对数据平滑方法外，还有哪些数据平滑方法？

第 11 章

知识类数据挖掘技术

引言

在用统计分析类数据挖掘技术进行数据分析时, 企业管理人员或管理顾问必须在分析开始之前就知道变量是什么, 他们需要分析什么。如果他们不知道所分析的对象或对所分析的变量不清楚, 那就很难对数据仓库中如此众多的数据和对象采用统计类数据挖掘技术进行商业分析。但是他们往往凭直觉, 感到在数据背后隐藏着某些市场规律和商业知识。此时, 统计类数据挖掘工具就难以承担重任, 人们就不会对手中的数据挖掘技术感到满意。他们需要一些功能更加强大的, 依靠数据驱动的, 而不是依靠商业分析人员驱动的数据挖掘技术来完成这些数据挖掘分析任务。

通过本章学习, 可以掌握:

- ◆ 知识发现系统的一般体系结构
- ◆ 关联规则、神经网络、遗传算法和粗糙集等各种知识型数据挖掘技术概念
- ◆ 知识型数据挖掘技术的基本应用
- ◆ 知识型数据挖掘工具的系统结构
- ◆ 知识型数据挖掘应用中的问题

11.1 知识发现系统的一般结构

11.1.1 知识发现的定义

知识发现技术(KDD)是随着数据库开始存储了大量业务操作数据,开始使用机器学习的方法,分析这些数据、挖掘这些数据背后的知识而开始发展起来的。随着 KDD 研究的进展,出现了有关 KDD 的一些定义:

- 知识发现是用一种简洁的方式从大量数据中抽取信息的一种技术,而这些信息是隐含的、未知的,并且具有潜在应用价值;
- 知识发现可以看成一种有价值信息的搜寻过程,它不必预先假设或提出问题,但仍能找到那些非预期的令人关注的信息;这些信息表示数据元素之间的关系和模式,它也能通过完整的、全面的信息发现和数据分析,找到有价值的商业规则;
- 知识发现可能意味着在数据仓库或数据集市的上千兆字节数据中,寻找预先未知的商业事实。

通过这些定义可以看出, KDD 是从数据仓库中识别有效的、新颖的、有潜在应用价值的,以及最终可以理解知识的复杂数据处理过程。这个数据处理过程是一个多步骤过程,这些步骤之间相互影响、反复调整,形成一种螺旋式的上升过程。数据挖掘是 KDD 最核心的部分,是采用机器学习等方法进行知识挖掘的阶段。数据挖掘算法的好坏将直接影响所发现知识的质量。目前大多数的研究都集中在数据挖掘算法和应用上。人们往往不严格区分数据挖掘和数据仓库中的知识发现,把两者混淆使用。一般在科研领域中称为 KDD,而在工程领域则称为数据挖掘。KDD 是一门交叉学科,涉及人工智能、机器学习、模式识别、统计学、智能数据库、知识获取、数据可视化、专家系统等多个领域。

在现实生活中,商业经理和商业分析员总在寻找相关的和最新的商业信息,以便做出更好的商业决策,这些决策对企业战略发展具有重要的影响。商业分析员没有时间,也没有过多的注意力和精力从数据仓库中发现所有隐含的关系和模式。使用传统的数据查询和分析技术时,要求所提出的问题是恰当的。知识发现技术则是自身决定所要提出的问题,然后不断对问题进行深入询问,持续地进行深入探索,直到找到用户所感兴趣的知识。它能够从数据仓库的大量数据中筛选信息,寻找市场经营中经常出现的特有模式,检查市场的发展趋势并且发掘事实。知识发现系统只需分析员最少的指导(在有分析员参加的情况下),就可以在最短时间内找到事实或知识。在知识发现过程中,需要有数据仓库或数据集市的大量数据支持;在找到事实或知识后,这些事实或知识将发送给商业分析员,用于管理决策分析。

知识发现系统用于发现预先并不具有的知识,即那些数据中隐含的知识,或在其应

用领域中没有明显表示的知识。这里的知识主要限于那些能以数据元素间的关系或数据模式所表示的知识，这些知识往往与特定的领域和任务有关，是一些令人感兴趣的、有实际使用价值的知识。在知识发现的定义中隐含着某种程度的自治性——不依靠那些有监督的学习。知识发现的自治性是一个现实问题，许多知识发现系统需要分析员输入或指导，这就需要将商业分析员作为知识发现系统的一部分来考虑。

11.1.2 知识发现系统的结构

知识发现系统的结构由知识发现系统管理器、知识库、商业分析员、数据仓库的数据库接口、数据选择、知识发现引擎、知识发现评价、知识发现描述等部分组成（参见图 11.1）。知识发现系统依靠这些构件可从数据仓库或数据集市的数据存储中，抽取分析人员关注的、对于商业分析员有用的模式和关系。

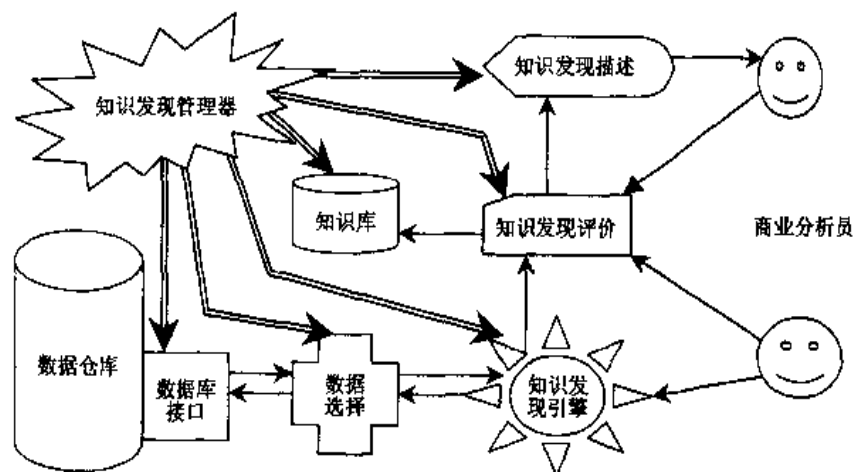


图 11.1 知识发现系统结构

在知识发现系统中，各部分之间的区分并不很清晰。知识发现系统中主要的输入是源于数据仓库的数据、商业分析员的指导，以及存储在知识库中的先验知识和经验。从数据仓库中选择的数据在知识发现引擎里处理，引擎中提供大量抽取算法，以便生成辅助模式和关系。然后对这些辅助模式和关系进行评价，它们中的一些被认为感兴趣的，被发送给商业分析员。有些发现可能还要加入知识库中，以便为后继知识发现提供先验知识。

1. 知识发现系统管理器

知识发现系统管理器控制并且管理整个知识发现过程。商业分析员的输入和知识库中的信息用于驱动以下三个过程：数据选择过程，抽取算法的选择及使用过程，发现的评价过程。系统管理器也帮助生成需要发送给商业分析员的发现结果描述，它还帮助将

合适的发现结果存储于知识库，作为下一步知识发现的先验知识。知识发现引擎的管理层由知识发现系统管理器管理。

2. 知识库和商业分析员

知识库包含源于各方面的知识。商业分析员可将元数据输入数据仓库，描述数据仓库的数据结构。商业分析员还要在知识库中输入其他相关的数据知识，例如应当注意的关键数据字段、分析中用于产生数据需求的商业规则、任何数据层次等。其目的是按一种有效的方式指导对关注性信息的发现。使用指导带来的风险是可能丢失潜在有用的模式和关系，商业分析员必须对此做出权衡。通过存储新的发现结果以驱动且增强后继使用，可以提高知识发现工具的能力。

知识库还可存储大量数据信息，以便在发现达到所需的可信级别时，抽取这些信息。有时只需一定数量的示例就足够了，有时却要处理整个数据库。不同模式抽取算法所需的数据类型和数据，存储于知识库中。

3. 数据仓库的数据库接口

知识发现系统利用数据库的查询机制，从数据仓库中抽取数据。对于关系数据库，可以使用 SQL 查询语言。知识库中的数据仓库元数据指导数据库接口正确组织数据结构，并且正确组织数据结构在数据仓库中存储的方式。为了提高效率，知识发现系统的数据库接口可以直接与数据仓库通信。

4. 数据选择

此构件可以确定从数据仓库中需要抽取的数据及数据结构。知识库指导数据选择构件选择需要抽取的数据以及抽取方式。如果只需示例数据，数据选择构件必须有能力选择并且抽取恰当的随机事例。此外，它还要选择算法所需要的数据类型，且将数据类型输入算法。

5. 知识发现引擎

知识发现引擎将知识库中的抽取算法提供数据选择构件抽取的数据，其目的是抽取数据元素间的模式和关系。存储在知识库中的经验对发现抽取有重要的作用。

许多数据挖掘算法可与知识发现系统结合，作为知识发现引擎，例如数据依赖、分类规则、聚类、概括数据、偏差检查、归纳和模糊推理等。

6. 发现评价

商业分析员需要寻找关注性的数据模式，以便了解顾客、产品、市场等等。数据仓

库潜在地具有宿主模式。评价构件或过滤构件有助于商业分析员筛选模式，选出关注性的信息。用于分析关注性模式的技术包括统计的重点、覆盖级别的置信度因子，以及可视化分析。

7. 发现描述

此构件提供两种必须的功能。一种是以发现评价辅助商业分析员，在知识库中保存关注性的发现结果，以备引用和使用。另一种是保持发现与商业经理（或商业总经理）的通信。其目的是利用知识发现来理解业务模式，将此理解转化成可执行的建议。知识发现系统中的描述技术包括可视化导航和浏览、自然语言文本报告以及图表和图形。

11.2 知识发现技术

数据挖掘中的知识发现技术按照其不同的技术特点，可以分成规则型知识挖掘技术、神经网络型知识挖掘技术、遗传算法型知识挖掘技术和粗糙集型数据挖掘技术。这些不同类型的知识挖掘技术在数据挖掘中占有重要的地位。

11.2.1 规则型知识挖掘技术

规则归纳是数据挖掘的一种主要形式，而且是无教师学习系统中最普遍的知识发现形式。它也是与大多数人想象的数据挖掘过程最为相似的一种数据挖掘形式，即在大型数据库中“淘金”。这里的金子指人们感兴趣的规则——能够提供一些原先不知道，或者不能明确表达出来的有关数据库的信息。

1. 关联规则的基本概念

在关联规则系统中，规则本身是“如果条件怎么样、怎么样、怎么样，那么结果或情况就怎么样”的简单形式。可以表示为“ $A \Rightarrow B$ ”关联规则，它包括两个部分：左部 A 称为前件，右部 B 称为后件。前件可以包括一个或多个条件，在某个给定的正确率中，要使后件为真，前件中的所有条件必须同时为真。后件一般只包含一种情况，而不是多种情况。

例如，购买计算机有购买财务软件趋向的关联规则，以及年龄在 30 至 40 岁之间并且年收入在 42 000 元至 50 000 元之间的客户购买高清晰度彩色电视机趋向的关联规则可以分别表示为

$$\text{buys}(x, \text{"computer"}) \Rightarrow \text{buys}(x, \text{"financial_management_software"}) \quad (11.1)$$

$$\text{age}(\text{"30...40"}) \wedge \text{income}(\text{"42 000...50 000"}) \Rightarrow \text{buys}(x, \text{"high_resolution_TV"}) \quad (11.2)$$

其中 x 为表示客户的变量。

关联规则在实际应用中根据值类型、数据维、层次的不同，可以分成各种类型的规则。

根据规则中所处理的值类型可以分成布尔关联规则和量化关联规则两种。例如，上述的关联规则 (11.1) 就是布尔关联规则，而关联规则 (11.2) 则是量化规则，其量化属性值是离散值。

如果规则中的项或属性只涉及到一个维，那就是单维规则。例如关联规则 (11.1) 只涉及 buys 维。而关联规则 (11.2) 则涉及三个维 age, income 和 buys 数据维，因此是多维关联。

如果关联规则集涉及不同的抽象层次，那么关联规则集就是多层关联规则；反之，就是单层关联规则。例如，规则 (11.1) 和 (11.2) 都是单层规则。而关联规则集

$$\text{age} ("30 \cdots 40") \Rightarrow \text{buys} (x, "IBM \text{ computer} ") \quad (11.3)$$

$$\text{age} ("30 \cdots 40") \Rightarrow \text{buys} (x, "computer ") \quad (11.4)$$

涉及的购买商品有较低抽象层次 “IBM computer” 和较高抽象层次的 “computer”。因此，规则集 (11.3) 和 (11.4) 是多层关联规则。

关联规则在实际应用中用 SQL 语言就可以很好地处理，例如对于关联规则 (11.2) 可以用下面的 SQL 查询语句完成。

```
Select Cust.name, P.item_name
from Purchases, P
group by Cust.ID
having (Cust.age>=30.and.Cust.age<=40).and. (Cust.income>=42000 and Cust.income
<= 50000) and (p.item_name='high_resolution_TV')
```

2. 关联规则的应用目标

关联规则的应用须有应用目标，在实际应用中可以以前件为目标、以后件为目标、以正确率为目标、以覆盖率为目标或者以“兴趣度”为目标。

以前件为目标的关联规则是将前件等于某值的所有规则收集起来显示给用户。例如：一个五金店可能需要前件为钉子、螺栓或螺钉的所有规则，以了解对这些低利润的商品打折是否能够促进其他高利润商品的销售。

以后件为目标的关联规则是查找后件等于某值的所有规则，用来了解什么因素与后件有关或对后件有什么影响。例如，得到后件为“咖啡”的所有准则对于咖啡的销售就十分重要，可以从中了解哪些商品的销售会导致咖啡销售的增加。咖啡店就可以将这些商品放到咖啡附近，以同时提高两者的销售额。或者，咖啡厂商可以根据这条准则决定

下次把他们的优惠券放到哪些杂志上。

以正确率为目标的关联规则，主要是以正确率表示前件为真时，后件为真的可能性。正确率高表示规则比较可靠。正确率有时亦称为置信度，对于“ $A \Rightarrow B$ ”关联规则，其置信度或正确率可以定义为

$$\text{置信度}(A \Rightarrow B) = \frac{\text{包含}A\text{和}B\text{的元组数}}{\text{包含}A\text{的元组数}} \quad (11.5)$$

有时，对用户来说最重要的是规则的正确率。正确率达到 80% 或 90% 以上的规则，表明发现的关系是很强的。即使它们对数据库的覆盖率较低，出现的次数有限，只要抓住这些规则，成功的可能性就比较高。

以覆盖率为目标的关联规则表示数据库中适用于规则的记录数量。其覆盖率可以定义为

$$\text{覆盖率}(A \Rightarrow B) = \frac{\text{包含}A\text{和}B\text{的元组数}}{\text{元组总数}} \quad (11.6)$$

覆盖率高表示规则经常被使用，由取样技术或数据库性质得到某种现象的可能性也比较大。有时，用户想知道哪些是最普通的规则或哪些规则最容易应用。将规则按覆盖率排序，用户就能很快知道哪些情况是数据库中经常出现的。

以“兴趣度”为目标的关联规则是评价人们使用关联规则以后，对所产生规则的兴趣程度，它与正确率和覆盖率有关。如果覆盖率一定时，兴趣度随着正确率的增大而增大。在正确率一定时，兴趣度随着覆盖率的增大而增大。可以采用各种方法将规则按某种兴趣排序，就能在覆盖率和正确率的评价目标选择中做出平衡。

11.2.2 神经网络型知识挖掘技术

1. 神经网络及其学习方法

神经网络(Neural Net)是指由大量神经元互联而成的网络，有点像服务器互联而成的因特网。它主要由“神经元”的互联，或按层组织的结点构成。通常，神经网络模型由三个层次组成：输入层，中间层（亦称隐层）和输出层（参见图 11.2）。在每个神经元求得输入值后，再汇总计算总输入值；由过滤机制比较总输入值，确定网络的输出值。可以通过连接一组神经元来模型化复杂行为。当修改连接层的“接度”或权值时，神经网络就进行了学习或“训练”。

神经网络将每个连接看做一个处理单元(PE)，试图模拟人脑神经元的功能。处理单元(PE)采用一系列数学函数，通过汇总和转换，对数据进行处理。一个 PE 的功能有限，

但是多个 PE 连接成系统后,就可创建一个智能模型。PE 能够以多种方式进行连接,为了精确地拟合建模数据,可能需要反复训练多次,甚至成百上千次。处理单元 PE 需要与输入/输出层的单元进行连接。在网络的训练过程中,对输入单元和输出单元的连接强度(权值)进行修改,其修改的值变化是根据它对所产生结果的重要性来确定的。连接的强度依赖于在反复训练过程中赋予它的权值。训练过程采用称为学习规则的数学方法调节权值。神经网络的训练是根据历史样本数据反复进行的,训练过程中 PE 对数据进行汇总和转换,它们之间的连接被赋予不同的权值。为对一个样本进行预测,要对网络尝试各种不同的方案。当输出结果与已知结果的吻合达到一定的精度或满足其他的结束准则时,就停止对网络的训练。此时,网络就可以用来对所需要分析的数据进行预测。

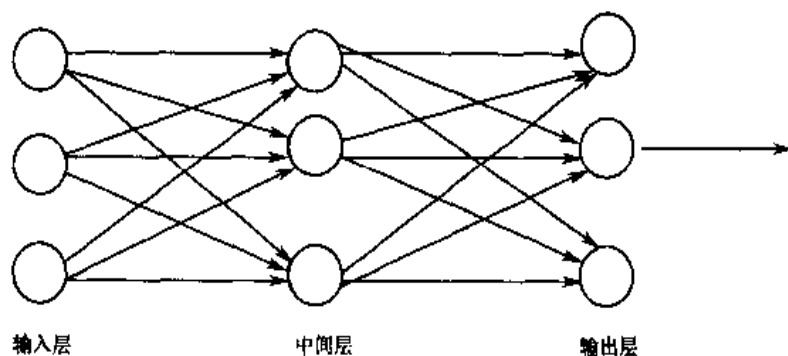


图 11.2 神经网络模型

所有神经网络的工作过程主要分两个阶段:学习阶段和工作阶段。学习方式则有三种:有监督(教师)学习、无监督(教师)学习和强化学习。有教师学习过程需要给定结果,并且给出反馈值;无教师学习则不用指定结果,不给出期望反馈值,按照网络自认为合适的方式组织数据。在强化学习过程中,外部环境对系统输出结果给出奖、惩信息,学习系统通过强化受奖的动作来改善系统性能。

神经网络在学习过程中必须依靠学习算法,矫正学习过程中的误差或偏离。这些算法有误差纠正法、竞争学习法等。

神经网络从经验中学习,经常用于发现一组输入数据和一个结果之间的未知联系,与其他方法一样,神经网络先要检测数据中存在的模式,再对从数据中发现的规则进行概括,最后给出结果。由于神经网络能对许多复杂的过程进行预测,得到人们的关注。

2. 基于神经网络的数据挖掘

基于神经网络的数据挖掘技术数以百计,较常使用的有基于自组织神经网络的数据挖掘技术和模糊神经网络类型的数据挖掘技术。

(1) 基于自组织神经网络的数据挖掘技术

自组织过程是一种无教师学习过程。通过学习,可以提取一组数据中的重要特征或

某种内在知识,如分布的特征或按某种特征聚类。

芬兰学者 T.Kohonen 认为,神经网络中邻近的各单元如同组成大脑的神经元一样,所发挥的作用是不同的,通过相互作用,可以自适应地发展成检查不同性质信号的特殊检测器。因为处于不同大脑空间部位的神经元分工是不同的,它们各自对输入不同的模式很敏感。T.Kohonen 还提出一种学习方式,使输入信号映射到低维空间,并且保持相同特征的输入信号在空间上对应临近区域,这就是所谓的自组织特征映射(SOFM)。

(2) 模糊神经网络类型的数据挖掘技术

尽管神经网络具有较强的学习、分类、联想与记忆等功能,但是在将神经网络用于数据挖掘时最大的难度是无法对输出结果给出直观的说明。将模糊处理功能引进神经网络后,不仅可以增加神经网络的输出表达能力,而且使系统变得更加稳定。

经常用于数据挖掘的模糊神经网络有模糊 BP 网络、模糊 Kohonen 聚类网络、模糊推理网络和模糊 ART 模型等。

模糊 BP 网络是基于传统的 BP 网络发展而来的。在传统的 BP 网络中,如果样本 x 属于第 k 类,除第 k 个输出节点为 1 以外,其他输出节点的输出值均为 0。即传统 BP 网络的输出值非 0 即 1,一点也不含糊。但是在模糊 BP 网络中,样本的希望输出值改为样本相对各类的希望隶属度。模糊 BP 网络在学习阶段将样本及其相对于各类的希望隶属度经过训练后,模糊 BP 网络具有了反映训练集内输入与输出隶属关系的能力,在数据挖掘时能够给出待识别模式的隶属度。

模糊 Kohonen 聚类网络不仅在输出表达方面实现了模糊化,而且将样本的隶属度引入权系数的修正规则中,使权系数的修正规则也实现了模糊化。

3. 基于神经网络的数据挖掘技术特点

神经网络可按管理模式或非管理模式来学习。在管理模式中,神经网络需要预测现有示例可能带来的结果,它将预测结果与目标答案相比较且从错误中进行学习。管理模式的神经网络可用于预测、分类和时间序列模型。非管理模式的学习在描述数据时很有效,却不用于预测结果。非管理模式的神经网络创建自己的类描述、合法性验证和操作,它与数据模式无关。神经网络可能需要经历很长的学习时间。由于它们的行为像黑盒,有时无法满足商业分析员的置信度要求。

11.2.3 遗传算法型知识挖掘技术

1. 遗传算法的基本原理

遗传算法是模拟生物进化过程的计算模型,是自然遗传学与计算机科学相互结合、相互渗透而形成的新的计算方法。

遗传是一种生物从其亲代继承特性和性状的现象。继承的信息由基因携带，多个基因组成染色体，基因在染色体中的位置为基因座。同一基因座可能有的全部基因为等位基因，等位基因和基因座决定了染色体的特征，也就决定了生物个体的特性。

从染色体的表现形式看，有两种相应的表示模式，分别是基因型和表现型。表现型是指生物个体表现出来的性状，而基因型则是指与表现密切相关的基因组成。同一基因型的生物个体在不同的环境条件下有不同的表现型。因此，表现型是基因型与环境相互作用的结果。

在遗传算法中染色体对应的是一系列符号序列，通常用 0, 1 的位串表示。串上各个位置对应上述的基因座，各位置上所取的值对应等位基因。遗传算法对染色体进行处理，称其为基因个体。一定数量的基因个体组成基因群体。群体中个体的数目为群体的规模，各个体对环境的适应程度称适合度。

生物在其生存过程中，能够通过自然选择逐渐向适应生存环境的方向转化，即进行生物的遗传进化。在这一过程中包括三种演化操作：在父代基因群中的双亲选择操作，两个父代双亲为产生子代基因的交叉操作，在子代基因群体中的变异操作。

遗传算法为模拟生物的遗传进化操作，必须完成两种数据转换：一是从表现型到基因型的转换，即将搜索空间中的参数或可行解转换成遗传空间中的染色体或个体，完成编码操作；另一种是从基因型到表现型的转换，是前者的反方向操作，为译码操作，将遗传空间中的染色体或个体转换成解空间中的最优解。

遗传算法实质上是一种繁衍、检测和评价的迭代算法。该算法以所有个体为对象，通过选择、交叉和变异算子实现群体的换代演化（参见图 11.3），使新生代的基因群体具有更高的适应环境的能力。

遗传算法的最大优点是问题的最优解与初始条件无关，而且搜索最优解的能力极强。遗传算法可以将各种数据挖掘技术进行优化，例如，神经网络、最近邻规则等。解决这些问题的关键是如何将复杂的现实问题解决方案转换成计算机中的模拟遗传物质（一系列的计算机符号）。

2. 遗传算法的处理过程

(1) 编码并生成祖先群体

要用遗传算法解决问题，先要定义有待解决的问题

$$F=f(a, b, c), F \in R, (a, b, c) \in \Omega$$

$F=f(a, b, c)$ 是属于实数域 R 的一个实数，也是每一组解 $(a_i, b_i, c_i) \in \Omega$ 的适应度的度量，算法的目标是找一个 (a_0, b_0, c_0) ，使 $F=f(a_0, b_0, c_0)$ 取最大值。

然后，将问题转换为遗传空间中的数串结构。其中主要是对各种自变量进行编码，

一般用一定位数的二进制编码表示。将所有的自变量编码连接成一串,得到表示自变量的一组取值所决定的一个可行解。例如,有自变量 a, b, c , 每个自变量用 4 位表示, 可以取得 12 位的二进制代码, 如: 100011100101。如果将每个解看成生物群体中的一个个体, 那这个代码串就是个体遗传特性的基因码链。

从遗传算法的初始计算考虑, 要为遗传计算准备若干初始解的祖先群体, 这些祖先群体由随机数生成。每个码串构成祖先群体中的一个祖先个体, 具有一定数量的祖先个体就构成了原始的祖先群体。遗传算法就从这些原始祖先群体开始。

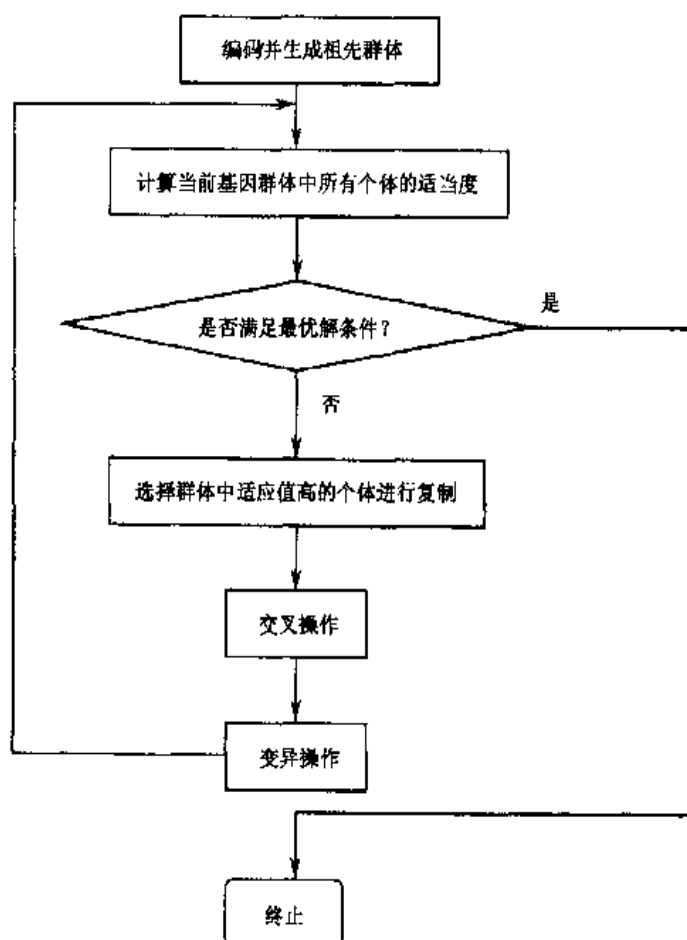


图 11.3 遗传算法处理流程图

(2) 计算当前基因群体中所有个体的环境适合度

对当前基因群体中的所有个体分别计算环境适应度函数。

(3) 用适应函数评价每个个体对环境的适应度

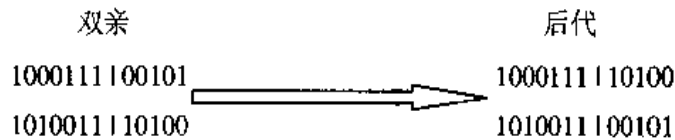
评价每个个体的适应度函数, 也就是计算目标函数, 按照编码规则将群体中的每个基因个体的基因码所对应的自变量取值代入目标函数, 算出相应的函数值 F_i 。 F_i 越大, 第 i 个基因的适应度越好。

(4) 选择适应度好的生物个体进行复制

用各种算法选择适应度好的个体作为优先配对繁殖的个体,使适应度好的基因有更多的机会繁殖后代,使优良生物的基因得以遗传和保留。这些适应度好的基因个体通过复制操作被送进配对库。

(5) 选择适应度好的生物个体进行复制交叉配对繁殖

随机地从配对库中选择双亲进行位串处理,得到新一代基因个体。位串处理时,对随机选择的双亲的位串随机地取一个截断点,将双亲的基因码链在截断点处切开,交换双亲的位串尾部,得到两个后代的基因个体。例如,从配对库中随机选择了双亲 100011100101 和 101001110100,截断点随机定于第 7 位于第 8 位之间。则交叉操作的结果是



经过这样处理后,得到新一代的群体,其对环境的平均适应度要比上一代好。

(6) 新生代的变异操作

在完成新生代的繁殖以后,还要根据某种概率,随机地从新生代中选择一些个体进行变异操作。一般这种变异的概率只在 1%至 2%之间。变异操作的目的是避免由于选择交叉过程中引起某些信息的永久性丢失,而降低变异操作的有效性,即避免出现只能获取局部解的弊端。在完成新生代的变异操作后,可以返回第(3)步重新生成新的下一代。这样持续地迭代下去,使群体的平均适应度和最优个体的适应度不断上升,直至获取满意解或群体的适应度不再提高为止。

11.2.4 粗糙集型知识挖掘技术

现在所使用的大多数数据挖掘工具都是基于集合论开发的,其中应用最多的是粗糙集(RS)。粗糙集是波兰学者 Pawlak Z 在 1982 年提出的,这是一种研究不确定性问题的数学工具。粗糙集(RS)作为集合论的扩展,主要用于研究不完全和不完整信息描述的数据挖掘技术。它能够在缺少关于数据先验知识的情况下,以考察数据的分类能力为基础,解决模糊或不确定数据的分析和处理。同时粗糙集算法简单,易于操作。目前,以其为基础构造的数据挖掘工具也比较多。

粗糙集根据已有的给定问题的知识,对问题论域进行划分。然后对划分后的每个组成部分确定其对某个概念的支持程度:肯定支持、肯定不支持 and 可能支持三种。在粗糙集中用正域、负域和边界三个近似集合,表示这三种情况。粗糙集中的不精确概念用所有对象一定被包含在集合中的下近似和所有对象可能被包含的上近似的表示。

粗糙集用于从数据库中发现分类规则的基本思想是将数据库中的属性分为条件属性和结论属性。对数据库中的元组根据各个属性不同的属性值分成相应的子集，然后对条件属性划分的子集与结论属性划分的子集之间上下近似关系生成判定规则。

粗糙集的理论出发点是假定所研究的每个对象涉及一些信息（数据、知识）。例如，对象是某些流失客户的案例，那么这些流失客户的特征就构成流失客户的信息。如果对象由相同的信息描述，那么它们就是相似的或不可分辨的，也就是是一些可能流失的客户。

所有相似对象的集合称为初等集合，形成知识的基本成分，任何初等集合的并集称为精确集。否则，一个集合就是粗糙的（不精确的）。每个粗糙集都具有边界元素，也就是那些既不能确定为集合元素也不能确定为集合补集元素的元素，而精确集没有边界元素。

11.3 知识发现技术的运用

11.3.1 关联规则的应用

通常，关联规则用于值域的基数很高或有多个二值属性列的数据库。经典的例子是超级市场扫描记录的销售数据，它包括单个商品的名称和数量，几万种商品组合后可能产生几十万个 SKU（库存单元标识 Stock Keeping Units）。

在这些数据库中，有时很难定义记录的概念——把很多数据仓库中用于存储超市交易的典型的星型模式看做事实表中不同的入口。事实表的列包括销售记录的惟一标志（所有的商品在同一个销售记录中）、购物时间、商品是否涨价（特价或使用优惠券）。因此，销售记录中的每个商品是位于事实表的不同行。这样的数据排布对大多数数据挖掘算法来说不是最优的，它们倾向于用一行表示一个销售记录、用一列表示某个给定商品出现与否的数据结构。

然而，这样存储数据的代价是相当昂贵的。如果商店有 60 000 个 SKU，即结账柜台上出现的不同的商品数，这样的记录结构将产生一个极高维的空间（有 60 000 维，每一维都是二维的），经典的数据挖掘算法如神经网络和决策树很难对其进行处理。

关联规则则可对其进行预测，但关联规则更多地用于无监督学习的知识发现。系统能够给出数据的详细信息，包括一些出现次数较少、只能从细节数据中得到的重要模式。系统也能给出数据总的情况，有些系统向用户提供一个包含数据库中所有模式的总的视图。正是这样，关联规则系统把微观视图和宏观视图很好地结合起来了。其中，宏观视图显示那些覆盖多种情况的模式，它们经常被使用，置信度较高，可以用来概括整个数据库；微观视图是系统能够找到那些覆盖少数情况，但非常有用的规则，且把它们提供给最终用户。如果它们覆盖的情况非常有价值（规则只适用于那些利润较高的顾客），那么这些规则也是很有价值的。如果规则仅代表一小部分顾客，但这部分顾客在不断增长，

它们也是有价值的，因为这可能表示销售的转换或一个新竞争对手的出现（例如，国内某个特定地区出现了新的竞争对手，因此那几个地区的顾客不断减少）。

产生规则并且计算规则的兴趣度后，还希望用规则进行预测。每条规则本身就是一种预测——后件是目标，规则的正确率就是预测的正确率。因为关联规则系统是有多条规则对应于一个给定的前件或后件，所以可能得到相互冲突的预测，正确率也互不相同。把不同的规则结合起来，可能提高整个系统的性能。还有多种做法，如把正确率作为权值累加起来，或用正确率最高的规则作为预测。

表 11-1 显示了前件和后件规则中的正确率和覆盖率，它们都包含某个给定的后件或前件，但正确率和覆盖率各不相同。在这个例子中，是根据购物篮中的其他商品预测顾客是否购买牛奶。如果购物篮中只有面包，那么从表中可以看出同时购买牛奶的可能性为 35%，如果顾客同时买了面包、黄油和鸡蛋，情况又该如何？是否是 65%。因为在这三种商品中，黄油和牛奶的关联度最大，为 65%，或者，同时购买这三种商品会增加购买牛奶的可能性，使它超过 65%。在用规则做预测时，如何把多个规则的信息结合起来是算法的关键。

表 11-1 前件和后件规则中的正确率和覆盖率

前件	后件	正确率	覆盖率
百吉饼	奶油干酪	80%	5%
百吉饼	橙汁	40%	3%
百吉饼	咖啡	40%	2%
百吉饼	鸡蛋	25%	2%
面包	牛奶	35%	30%
黄油	牛奶	65%	20%
鸡蛋	牛奶	35%	15%
奶酪	牛奶	40%	8%

从商业的角度来看，正确的规则很重要，因为它们表示数据库中可以发现有用的起预测作用的信息——即前件和后件之间的某种联系。正确率越低，规则就越可能是随意的猜测。如果正确率比随意猜测还低的多，则前件的否定可能会很有用。

覆盖率则是表示有用规则被使用的频率。如果企业有一条规则的正确率为 100%，但仅仅适用于 10000 条销售记录中的一条。根据这一事实，企业可以重新安排货架，但这并不一定能让企业获利，因为这一事件发生的可能性只有万分之一。表 11-2 表示规则覆盖率和正确率的平衡。在应用规则时，需要根据实际情况进行综合平衡。

表 11-2 规划覆盖率和正确率的平衡表

覆盖率	正确率	平衡
覆盖率高	规划很少是正确的，但可以使用	规划多数情况下是正确的，而且可以经常使用
覆盖率低	规划很少是正确的，一般不被使用	规划多数情况下是正确的，但很少被使用

11.3.2 神经网络的应用

图 11.4 是一个非常简单的预测贷款拖欠情况的神经元网络图。圆圈表示节点，圆圈之间的连线表示连接。神经网络是这样工作的，它从左边的节点获得预测属性值，对于这些值进行计算后，在最右边的节点产生新值，最右节点的值表示神经元网络模型做出的预测。在这里，神经网络把年龄和收入作为输入的预测属性，预测一个人是否会拖欠银行贷款。

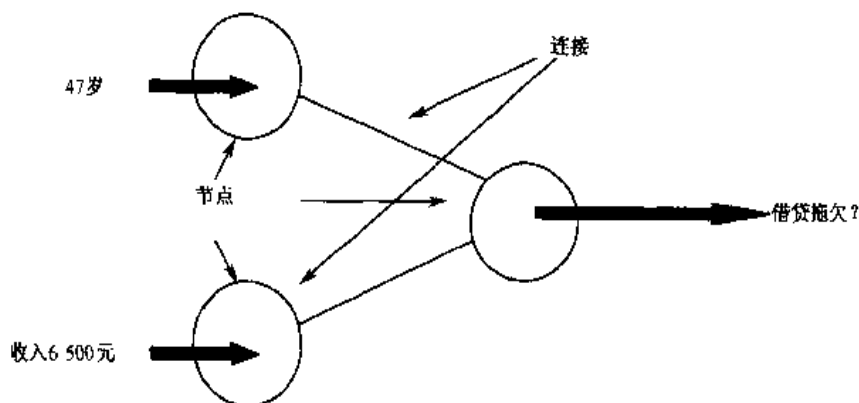


图 11.4 一个简单的预测贷款拖欠情况的神经元网络

为了进行预测，神经网络从输入节点获得预测属性的值，这些值称为节点的值。节点与连接中存储的值相乘，得到的值在最右节点相加，再进行指定的阈值运算，得到的数值就是预测值。在这里，如果得的值是零，就认为这条记录的信用风险较低（无拖欠情况发生）如果得到的值为 1，就认为这条记录的信用风险较高（很可能拖欠贷款）。对图 11.4 的计算进行标准化，得到图 11.5。这里，年龄值 47 被标准化到 0.0 和 1.0 之间，变成了 0.47，而收入值被标准化为 0.65。这个简化后的神经网络做出的预测是，收入为 6 500 元，年龄为 47 岁的顾客是否会拖欠贷款，连接权值分别为 0.7 和 0.1，节点值与连接权值相乘后得到的结果为 0.39。经过训练后网络用输出 1.0 表示拖欠，输出 0.0 表示不拖欠。这里得到的输出值 0.39 更接近于 0.0，因此对这条记录做出的预测是不拖欠。

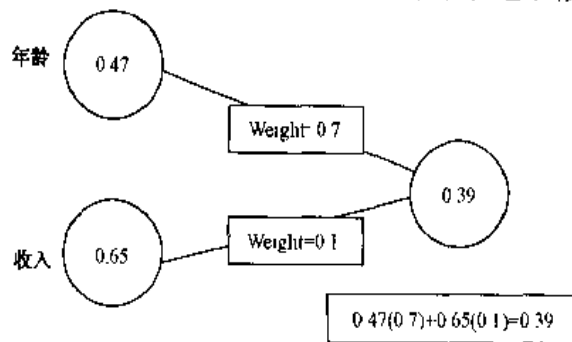


图 11.5 预测树结果

由于人工神经网络模型较多,在进行数据挖掘之时,必须根据数据挖掘目的,选择合适的神经网络模型和算法(参见表 11-3),才能获取良好的数据挖掘效果。

表 11-3 神经网络模型的比较

神经网络模型	神经网络模型	神经网络模型
分类	ARTMAP 模型	监督、反馈
	概率网络、LVQ 模型	监督、前馈
聚类	(模糊) ART 模型	无监督、反馈
	Kohonen 特征映射网络	无监督、前馈
	模糊 ARTMAP 模型	监督、反馈
分类、建模、时序分析	(模糊) BP 模型、RBF 模型、模糊推理网络、ANFIS	监督、前馈
优化、联想、分类	Hopfield	无监督、反馈
联想、分类、聚类	模糊 Hopfield	无监督、反馈

神经网络在使用时需要很长的训练时间,因而对有足够长训练时间的应用更为合适。此外,神经网络对噪声数据具有较高的承受能力。

对于神经网络来说,也要注意过适应数据问题。神经网络很可能对训练样本过适应,从而可能对新的数据记录的预测效果很差。一个解决的办法就是限制连接的数量。因为连接的数量越多,神经网络就越复杂,而复杂的神经网络更容易出现过适应数据问题。因此,限制连接的数量是解决神经网络过适应数据的一种方法。还有一种方法是将训练集的数据记录分为两部分,一部分数据用来构造神经网络模型,另一部分用来对构造的神经网络模型进行测试。在神经网络模型成长的过程中,对模型进行测试,且把正确率记录下来。最后,选择正确率最高的神经网络模型作为最后的模型。

11.3.3 遗传算法的应用

遗传算法在知识挖掘中能够发挥重要的作用,尤其因为它的优化处理技术,将遗传算法与其他数据挖掘技术结合在一起应用,以提高数据挖掘的处理效率。

表 11-4 是一个客户的信息组成数据表,可以利用遗传算法在客户群中预测最佳客户的类型。企业的最佳客户群,即可以从客户处获取最大利润的特征应该由客户的收入水平、客户的家庭人口、客户的年龄所构成。而从客户处所获取的利润则是从客户的累计购买商品金额乘以 2%,减去每次购买商品的手续费 10 元。这里的手续费约束条件在其他数据挖掘中是较难考虑的。

根据客户数据中的条件,可用如下 4 个染色体来定义客户类型。

基因 1: 客户的年龄下限;

表 11-4 客户的信息组成数据

客 户 ID	年 龄	累计购买金额	收 入	家 庭 人 口	性 别
10985	46	1843	中等	4	女
18595	49	0	中等	2	男
47382	61	3628	低	5	男
74912	36	18463	高	6	女
95623	29	8463	高	3	男
85526	32	274	中等	2	男
58753	52	1846	低	2	女
64957	48	0	中等	3	女
76957	27	21634	高	5	男
65839	45	842	低	1	女

基因 2: 客户的年龄上限;

基因 3: 客户的收入水平;

基因 4: 客户的人口状况, 分成少 (1~2 人)、一般 (3~4 人) 和多 (5 人以上) 三种状况。

这样就可以用: “40|55|中等|一般” 的基因位串表示年龄在 40~55 岁之间、中等收入、家庭人口一般的客户群。环境适应函数则从客户群中的购买收益中扣除手续费 (可以用 SQL 语句实现)。在适应函数中还要增加一个限制条件: 客户的年龄下限必须小于客户的年龄上限, 以防止在遗传计算过程中出现客户年龄下限超过客户年龄上限的情况。

在这里给出的基因分组会使最理想的客户群出现只有一种家庭人口和一种收入水平, 显然, 这与现实不相符合。为此, 对基因重新进行设计。按照家庭收入的高、中、低和人口的少、一般、多分别设计三个基因。用“是”、“否”值来确定某个值是否存在于某个客户群中。表 11-5 给出客户群的新基因组成的例子。在对这些染色体进行遗传计算时, 通常要将染色体的等位基因转换成二进制数, 例如, 用“1”表示“是”, “0”表示“否”。

表 11-5 客户群的新基因组成

	年龄下限	年龄上限	高收入	中等收入	低收入	人口少	人口一般	人口多
客户群 1	38	64	是	是	否	否	是	是
客户群 2	26	50	是	否	是	是	否	是
客户群 3	20	40	否	是	是	是	否	否

在完成遗传编码的定义后, 就可以产生一群随机生物个体; 然后, 计算这些生物个体的环境适应性, 并且利用遗传算法中的各种竞争、杂交算法进行生物的繁殖、杂交、异变的处理。为简化讨论, 这里采用一种简单的竞争和杂交算法, 该算法将竞争和杂交

只限于局部范围内进行。首先将所有的生物个体按顺序排放在一张二维表格上,使每个生物体的上、下、左、右都与其他生物体相邻接。然后,按照以下算法进行生物的遗传进化处理。

1. 竞争复制

每个生物与它左边邻居比较环境的适应度,如果左边邻居的环境适应度大,就用左边邻居的遗传信息替代自己的遗传特性,否则不做改变。这样就将导致适应度小的生物死亡,和适应度大的生物个体复制。

2. 杂交繁殖

从复制完成后的生物个体某个随机位置开始,将其与上方的邻居进行杂交繁殖,相互交换部分遗传信息,完成所有生物的交叉繁殖。

3. 异变处理

对新生代生物进行随机异变处理,即以 1%至 2%的概率随机选择新生代生物个体,并且随机改变被选中生物位串上某个位置的值。将原来是 1 的改变为 0,原来是 0 的改变为 1。

随机改变竞争和杂交方向,重复 1 至 3 的过程。即下一代的竞争、杂交方向可与右边邻居进行,而不是以前的与左边邻居的杂交处理。

在本例中,由于适应性函数是从利润角度定义的,系统应该逐渐收敛于客户收益最大的客户特征群。即使在计算过程中可能会使某类客户收益不是最大的客户群在其中占据多数,也会由于系统的突变功能,使系统继续探索新的客户特征群。

遗传算法在客户的分类中的应用是比较成功的。在应用遗传算法解决信息应用问题时,如果问题的解决方案价值可以详细说明;或要解决的问题很复杂而且没有其他的直接解决方法;或需要解决的是一种新问题,由于认识不足,无法使用其他工具;或问题涉及许多相互影响的变量,需要测试这些变量的相互影响效果。此时,遗传算法可以发挥极佳效果。

11.3.4 粗糙集的应用

粗糙集在商业应用中也具有很大的应用范围。例如从客户的忠诚度评价看,客户的忠诚意味着客户不断地回来找你,购买你的产品或服务,也许你没有最好的产品、最低的价格或最快速的交付手段。这种现象看上去很不合情理,但是基于良好的客户关系,可为企业带来客户价值和明显的收益。客户获得的全部价值不仅包括他们获得的产品或

服务, 也包括获得该产品/服务的方式。那些能将两方面都做到最好的公司常常是其专业领域的佼佼者, 他们将获得更多的市场份额和利润。

表 11-6 是流失客户的信息, 包括六个客户数据的属性值, 表的列为对象的属性, 行标识为对象(客户), 表中的数据记录了属性值。表中的每行都可看成有关流失客户的信息。例如, 流失的客户 970230 由表中的下列属性描述: { (赞扬竞争对手的产品, 是), (挑选产品时间很长, 否), (距最后一次销售时间, 长), (客户流失, 是) }。

表 11-6 流失客户的信息

客户ID	赞扬竞争对手的产品	挑选产品时间很长	距最后一次销售时间	客户流失
970102	否	是	长	是
970230	是	否	长	是
980304	是	是	很长	是
980625	否	是	正常	否
990211	是	否	长	否
990327	否	是	很长	是

表中客户 970230, 980304, 990211 相对属性“赞扬竞争对手的产品”是相似的; 客户 980304, 990327 相对属性“挑选产品时间很长”和“客户流失”是相似的; 客户 970230, 990211 相对属性“赞扬竞争对手的产品”、“挑选产品时间很长”和“距最后一次销售时间”是相似的; 这样, 属性“赞扬竞争对手的产品”产生两个初等集合: {970230, 980304, 990211}和{970102, 980625, 990327}; 而属性“赞扬竞争对手的产品”和“挑选产品时间很长”生成三个初等集合: {970102, 980625, 990327}, {970230, 990211}和{980304}。同样, 可以确定由任意子集所生成的初等集合。

因为客户 970230 已经流失, 而客户 990211 没有流失, 对于属性“赞扬竞争对手的产品”、“挑选产品时间很长”和“距最后一次销售时间”是相似的。因此, 客户流失不能以属性“赞扬竞争对手的产品”、“挑选产品时间很长”和“距最后一次销售时间”作为特征进行描述, 而 970230, 990211 就是边界实例, 即它们不能根据有效知识进行适当的分类。余下的客户 970102, 980304 和 990327 所显示的特征, 可以将他们确定为已经流失的客户。当然, 也不能排除 970230 和 990211 已经流失, 而 980625 毫无疑问没有流失。所以客户集合中“流失”的下近似集合是{970102, 980304, 990327}, 上近似集合是{970102, 970230, 980304, 990211, 990327}。同样, 980625 没有流失, 而 970230 和 990211 不能排除流失。因此, 客户“没有流失”概念的下近似是{980625}, 上近似是{970230, 980625, 990211}。实际上, 为了确定客户是否流失, 不必使用表中的所有属性。如果一个客户距最后一次购买的时间很长, 那该客户就一定流失了。如果距最后一次购买的时间正常, 那就一定没有流失。

由于知识发现所研究的对象大多是关系型数据库关系表, 可以将其看成粗糙集理论

中的决策表,这就给粗糙集的应用提供了便利的条件。在现实研究中的规则有确定性的和不确定性的,而粗糙集可从数据库中发现不确定性知识。同时粗糙集可从数据中发现异常数据,排除知识发现过程中的噪声干扰。粗糙集在实际应用中还可以和其他数据挖掘技术结合起来,提高数据挖掘的效率,例如,神经网络不能自动选择合适的属性集,但是利用粗糙集进行预处理,就可以去掉多余的属性,提高知识发现的效率。而且粗糙集在数据挖掘中所获得的决策规则与推理过程,要比神经网络或模糊集方法更容易验证。因此,粗糙集在数据挖掘中的应用将越来越广泛。

11.4 知识发现工具的应用

11.4.1 知识发现工具的系统结构

随着知识发现的流行和扩散,可以预计未来的几年中,将会设计、开发出各种知识发现系统。尽管丰富和强大的知识挖掘功能形成了知识挖掘的核心,像大部分软件系统一样,知识发现系统的结构和设计也是至关重要的。一个好的系统结构将有利于系统更好地利用数据库环境,可以有效地、及时地完成数据挖掘任务,有利于与其他信息系统协调和交换信息,有利于系统适应用户的种种要求。

“知识发现系统的期望结构是什么?”,数据库和信息产业界研究和开发历经数十年,使数据库和数据仓库系统已经成为主流信息系统。海量数据和信息已经存储、集成在这些系统中。此外,全面的信息处理 and 数据分析基础已经或将持续不断地、系统地围绕数据库系统 and 数据仓库构造。这包括多个异构数据库的访问、集成、统一和转换,ODBC/OLE DB 连接,Web 访问和服务工具,报告和 OLAP 分析工具。

在这种情况下,知识发现系统设计的一个重要问题是:是否应当将数据挖掘(DM)系统与数据库(DB)系统和数据仓库(DW)系统耦合或集成?如果应当,应该怎样正确地进行?为此,需要观察耦合或集成 DM 系统和 DB/DW 系统的可能途径。基于不同的结构设计,用不耦合、松散耦合、半紧密耦合和紧密耦合模式可将 DM 系统和 DB/DW 系统集成。

1. 无耦合

无耦合(no coupling)意味 DM 系统不利用 DB 或 DW 系统的任何功能。它可能由特定的源(如文件系统)提供数据,使用某些数据挖掘算法处理数据,再将挖掘结果存放在另一个文件中。

这种系统尽管简单,但有不少缺点。首先,DB/DW 系统在存储、组织、访问和处理数据立方体方面提供了很大的灵活性和有效性。不使用 DB/DW 系统,DM 系统可能要花

大量的时间查找、收集、清理和转换数据。在 DB 和 DW 系统中,数据已经被很好地组织、索引、清理、集成或合并,使得找出与任务相关的、高质量的数据成为一件容易的事情。其次,在 DB/DW 系统中,有许多被测试的、可伸缩的算法和数据结构。使用这种系统开发有效的、可伸缩的实现是切实可行的。此外,大部分数据已经或将要存放在 DB/DW 系统中。不与这些系统耦合,DM 系统就需要使用其他工具提取数据,这种系统今后将很难集成到信息处理环境中。因此,不耦合体系的数据挖掘系统是一种很糟糕的设计。

2. 松散耦合

松散耦合 (loose coupling) 意味 DM 系统将使用 DB/DW 的某些工具,从这些系统管理的数据存储中提取数据,进行数据挖掘,然后将挖掘的结果存放在文件中,或者存放在数据库或数据仓库中的指定位置。

松散耦合比不耦合好,因为它可以使用查询处理、索引和其他工具,提取存放在数据库或数据仓库中任意部分的数据。这就可以利用数据库或数据仓库系统所提供的灵活性、有效性等优点。然而,许多松散耦合的系统是基于内存的。挖掘本身不能使用 DB/DW 提供的数据结构和查询优化方法,对于大量的数据集,松散耦合系统就很难获得可伸缩性和良好的性能。

3. 半紧密耦合

半紧密耦合 (semitight coupling) 意味着除了将 DM 系统连接到一个 DB/DW 系统外,一些基本数据挖掘原语可以在 DB/DW 系统中实现。这些原语可能包括排序、索引、聚集、直方图分析和一些基本的统计度量(如求和、计数、最大、最小、标准差等)的预处理。此外,一些频繁使用的中间结果也可以预计算,并且存放在 DB/DW 系统中。由于这些中间挖掘结果或者预计算,或者可以有效地计算,这种半紧密耦合设计将提高 DM 系统的性能。

4. 紧密耦合

紧密耦合 (tight coupling) 意味着 DM 系统平滑地集成到 DB/DW 系统中。数据挖掘系统被视为信息系统的一部分。数据挖掘查询和功能根据 DB/DW 系统的查询分析、数据结构、索引模式和查询处理方法优化。随着技术进步,DM 和 DB/DW 将进一步集成在一起,成为一个具有多种功能的信息系统。这将提供一个一致的信息处理环境。这种方法是人们高度期望的,因为它有利于数据挖掘功能的有效实现,提高系统性能,实现集成的信息处理环境。

从这些分析中可以看到知识发现系统应当与一个 DB/DW 系统实现某种程度的耦合。

松散耦合尽管不太有效,也比不耦合好,因为它可使用 DB/DW 的数据和系统工具。紧密耦合是高度期望的,但实现并非易事,在此领域需要进行更多的研究。半紧密耦合是松散和紧密耦合之间的折衷。知识挖掘系统的结构最重要的是识别常用的数据挖掘原语,提供这些原语在 DB/DW 系统中实现的有效方法。

11.4.2 知识发现工具运用中的问题

许多知识发现技术源于人工智能和机器学习的研究。这些技术正在逐渐地从大学和研究中心推广至商业用户。在不断的实际应用中,知识发现技术正在不断吸取各种领域的经验而逐渐成熟。从目前情况来看,在运用知识挖掘技术时还需要注意一些问题,有的问题是所有知识挖掘技术在应用中都会遇到的公共问题,有的是不同数据挖掘技术自身所特有的一些问题。

1. 数据挖掘技术应用中的共性问题

在应用数据挖掘技术时,所遇到的共性问题有:数据质量、数据可视化、极大数据库、性能与成本、分析人员的技能、数据噪声和模式评价等问题。

(1) 数据质量

由于知识发现是数据驱动的,而且不易管理,因此知识发现很容易遇到数据质量的问题。许多数据库都是动态的、有错误而且不完整的、有冗余的和稀疏的,当然也是巨大的。因此在使用恰当的知识发现功能和技术的同时,必须小心地分析异常情况,不能将异常数据所造成的结果作为普遍的模式加以应用。

(2) 数据可视化

数据仓库包含大量数据,其中隐藏着各种业务模式。如果只对这些海量数据进行分析,其结果可能会使分析员变得不知所措。因此在知识发现过程中需要通过设定有效的探索始点,能够按适当的隐语来表示数据,使知识挖掘分析人员能够得到有力的帮助。数据的可视化是一种帮助知识挖掘人员了解数据、获取知识的有力工具。但是在知识挖掘中所遇到的数据大多是一些复杂的海量数据,要将其可视化,须有复杂的数据可视化工具支持。数据可视化是一种新兴的技术,它可提高商业分析员分析数据、获取知识的能力,尤其是在数据维数较低的时候,其效果更加明显。

(3) 极大数据库 (vLDB) 的问题

在数据仓库设计时,力图使数据库足够小,以便满足用户信息分析处理的需要。数据仓库一般只要有可能,就对数据概括化,以减少存储空间并且改善查询和报表的响应时间。但是对于知识发现,需要事务数据或细节数据,否则无法了解客户的行为方式和商业模式。这样,用于知识挖掘的数据仓库实际上是一个极大数据库,需要既保持查询

分析的概括数据,也要提供进行知识挖掘的细节数据。极大数据库除了在系统管理时存在问题外,许多知识发现技术也会由于极大数据库的尺寸过大而发生应用问题。例如,过大的查询数据尺寸会对一些特定技术(例如,神经网络训练)造成困难。这样,对极大数据库往往需要使用其他的数据抽取技术,生成一个知识挖掘数据库,便于知识挖掘技术的应用。

(4) 性能和成本

为了满足许多知识发现系统的计算要求,需要在硬件、操作系统软件上采用并行技术。这些性能要求大大增加了知识挖掘的成本。

(5) 商业分析员的技能

商业分析员需要丰富的业务知识,并且具有极强的调查能力,同时还应有创造性。创造性允许商业分析员试验各种知识发现技术,以便发现大量潜在的模式和关系。然后分析并且了解它,最后生成预测模型且按用户容易理解的形式发布。

(6) 处理噪声和不完全数据

存放在数据库中的数据可能包含噪声、异常情况或不完整的数据对象。这些对象可能使分析过程混乱,导致数据与所构造的知识模型过分适应。其结果是,所发现的模式精确性可能很差,不能满足实际的需要。因此,在知识挖掘系统中需要有处理数据噪声的数据清理方法和数据分析方法,以及发现和分析异常情况的孤立点挖掘方法来解决这些问题。

(7) 模式评估——兴趣度问题

知识发现系统可能发现数以千计的模式。对于给定的用户,许多模式不是有趣的。这表明不是知识模式缺乏吸引用户的新颖性,就是用户缺乏对知识模式的理解能力。因此,一种评价模式兴趣度的技术是解决这一问题的关键。关于开发模式兴趣度的评估技术,特别是关于给定用户类,基于用户的信赖或期望,评估模式价值的主观度量,仍然存在一些挑战。使用兴趣度度量来指导发现过程的压缩搜索空间,将是知识挖掘中一个研究活跃的领域。

2. 数据挖掘技术应用中的个性问题

(1) 规则归纳应用中的问题

规则归纳应用主要用于显式描述数据抽取的规则,常常是对带有属性或描述的数据项应用规则算法。使用规则归纳技术,数据库中所有可能的模式都要被系统地抽取出来,然后估计它们的正确性和重要性,以判断模式可以使人们相信的程度有多高,再次出现的可能性有多大。这样,它就可能得到数据库中所有可能的有趣模式,不会漏掉任何一种情况。从另一个角度讲这也是它的缺点,因为用户会淹没在数量繁多的规则中,把所有规则看一遍是很困难的。进行系统的规则归纳,找到所有的规则,工作量是巨大的,

这给系统分析人员带来很大的挑战。

(2) 神经网络应用中的问题

神经网络的研究内容相当广泛,反映了多学科交叉技术领域的特点。迄今为止,在人工神经网络研究领域中,有代表性的网络模型已达数十种,而学习算法的类型更难以统计其数量。它的最大优点就是能够精确地对复杂问题进行预测。

神经网络方法也有一些缺点。第一,神经网络易于受训练过度的影响。如果对具有很强学习功能的神经网络,用支持这种功能的少量数据进行训练,开始时正如希望的那样,网络学习的是数据中的一般趋势,此后网络却不断地学习训练数据中非常具体的特征,这不是所希望的。这样的网络由于记住了训练数据,缺乏概括能力。如今的商用神经网络已经有效地解决了这个问题。通过定期检查测试数据集的结果,可以检测训练过度问题。训练过程初期,训练和测试数据的误差都比较小。如果网络的功能超过预定功能或者训练数据太少,这种情况就不会继续下去。在训练过程中,如果测试数据开始产生错误结果,而训练数据的结果仍然在不断提高,这就说明,出现了训练过度问题。第二,神经网络的训练速度问题。构造神经网络时要求对其训练许多遍,这意味着获得精确的神经网络,需要花费许多时间。因此,神经网络的模型构建、数据训练可能花费过多的时间。

(3) 遗传算法应用中的问题

遗传算法能够解决许多其他技术难以解决的问题。它在问题解决过程中不是针对参数本身,而是通过对参数集进行编码的基因个体。使遗传算法可对一些复杂的结构对象,例如集合、序列、树、图、表等进行操作。利用对所有个体进行处理的方法,可以探索空间中的多个解,使遗传算法具有较好的全局搜索特性。

在数据挖掘领域,目前主要用于增强其他数据挖掘技术,例如与神经网络技术的结合,可以提高神经网络的可理解性。从遗传算法自身的角度考察,遗传算法实际上是一种最难以理解和开发难度最大的算法。

11.4.3 知识发现的价值

在目前极其复杂而且竞争激烈的商业环境中,组织中高层管理者的职责是制定商业或市场策略,定出销售计划,确定如何分配有限的资源。他们常要依据特定信息来了解与评价企业的特性,这些信息包括企业数据(企业历史)、产业和经济数据,以及最终影响其企业收益的未来事件。他们需要将数据仓库中的原始数据转化为简洁的商业信息,以便指导他们的市场营销、运作、投资策略和决策。知识发现工具有助于了解商业活动,发现商业异常和预测未来趋势。

1. 了解商业活动

知识发现工具及技术可以帮助了解商业活动的细节,有助于寻找重要的,但是不可见的和未知的商业事实。客户行为模式可按多种方式分析:从近似分析到市场细分,从侧面生成到购买序列模型。无论显式规则还是隐式规则都可生成并且进行分析。可以使用多种知识发现技术,以适合于不同数据类型,从数字到描述、到图像。

组合技术或组合不同技术的序列应用,有助于了解商业事实并且生成可以采取行动的建议。

2. 发现商业异常

知识发现工具也有助于异常检查和异常分析。检查异常可以使决策更佳,消除那些异常数据带来的影响。检查可使人们对数据建立未曾意识到的更深刻的商业认识。

3. 预测模型

在了解商业活动中发生了什么及其发生的原因后,知识发现技术可以帮助解决“现在该怎么办”的问题。也就是说,用户可以从过去预测未来并且做出计划。预测模型系统可以帮助用户了解市场变化,它与商业分析员专业领域知识相结合,构成可靠决策的最佳组合。

预测模型可以用于多个重要的商业领域,例如交叉销售、相关市场营销、产品包装(由单个或多个制售商包装其产品)、信用风险分析和商店位置设置分析等。

11.4.4 知识类数据挖掘工具简介

知识类数据挖掘工具有 IBM 公司的 IM 智能挖掘器,加拿大 Simon Fraser 大学智能数据库系统研究室创建的 DBMiner,SGI 公司与美国 Standford 大学所开发的 Mineset 等。在 MS SQL Server2000 中的 Analysis Services 中也提供了决策树和聚类两种挖掘技术。

1. DBMiner 的体系结构

DBMiner 是加拿大 Simon Fraser 大学智能数据库系统研究室所创建,由 DBMiner Technology 公司进一步开发的联机数据挖掘系统,目前已经有 3.0 版本。2.0 版本可从 <http://db.cs.sfu.ca/DBMiner> 或 <http://www.dbminer.com> 免费下载,有 90 天的试用期,有关单用户、教育用户的许可证可以从 <http://www.dbminer.com> 获取。

DBMiner 可从关系数据库或数据仓库中抽取数据,通过集成、转换装入多维数据库,例如,可从 SQL Server OLAP 的数据立方体中抽取数据,并且根据用户的要求进行联机

数据挖掘处理。在数据的挖掘处理过程中,可以进行钻取、切片、切块等灵活的操作。数据挖掘结果可用多种形式表示。数据的汇总、特征化等统计结果可用 MS Excel 的图形工具输出,关联规则可用关联表、关联规则图表示,分类结果可用决策树或决策表表示。

2. DBMiner 的数据挖掘类型

DBMiner 能够支持的数据挖掘功能包括分析、关联、分类、聚类、预测和时间序列分析。

DBMiner 的分析功能主要利用钻取、切片、切块等 OLAP 操作多维度,展示数据立方体中的内容,输出结果可以是各种可视化图形,并且可用统计分析工具计算最大值、最小值、标准差以及其他数据分布情况。

DBMiner 的关联功能可从多维数据库中挖掘一系列的关联规则,用户可以指定元模式限制对规则搜索,也可沿着任一维在多个抽象层次上挖掘规则,并且可用图形表示挖掘的关联规则。

DBMiner 的分类功能可对一组训练数据进行分析,根据数据特性对每个分类构造一个模型,再根据测试数据对模型进行调整。用决策树或决策表表示模型,且用该模型对其他数据分类。

DBMiner 的聚类功能可将一组选定的数据对象分成若干簇,使簇内的数据相似度高,不同簇中的数据相似度低。聚类结果可用不同颜色的图形输出。

DBMiner 的预测与时间序列分析在 DBMiner 2.0 版本中未实现,只在 DBMiner 3.0 以上的版本中才具备这些功能。



本章小结

知识挖掘技术是一种依赖数据驱动的、从数据中挖掘业务模式的知识发现技术。知识发现系统的结构在知识挖掘过程中各自承担自己的任务,相互协调才能使知识挖掘技术发挥其特有的价值。

在知识挖掘技术中,使用较多的有关联规则、人工神经网络、遗传算法和粗糙集等。关联规则是知识挖掘中一种主要的挖掘技术,通过关联规则在数据仓库中的应用,可使人们了解各种事物发生的前因后果。使企业利用挖掘的各种商业规则在市场竞争中获取优势。

人工神经网络是一种有效的预测模型。其模型比较复杂,许多人都难以理解,但是在聚类分析、奇异点分析、特征抽取中可以得到较大的应用,例如在信用卡欺诈、信贷风险、客户分类、盈利客户特征分析商业模式的识别上应用。

遗传算法作为基于生物进化过程的组合优化方法，在数据挖掘中主要用于分类系统中，并且经常与神经网络等数据挖掘技术综合应用。

粗糙集在数据挖掘应用中，经常用于处理不确定问题，而且在处理过程中可以不需要关于问题的先验知识，可以自动找出问题的内在规律。因此，在模式识别、决策分析、知识发现等方面得到较广泛的应用。

在应用知识挖掘工具时，需要关注其系统结构。只有与数据源结合良好的挖掘工具，才能发挥其真正的价值。

使用知识发现技术时会遇到数据质量、可视化数据的能力、极大数据库尺寸，以及商业分析员（也是数据发掘者）技能的一些问题，这些都会影响到数据挖掘工具的应用成败。



习题

11-1 知识挖掘系统的结构包括哪几个部分？它们是如何相互配合完成知识发现的？

11-2 现有某企业的员工数据库，数据已经概括处理，其中的合计数为对应所给定的部门、职务、年龄和工资值的人数（参见表 11-7）。

表 11-7 某企业员工数据库

部 门	职 务	年 龄 (岁)	工 资 (元)	合 计 (人)
销售	高级管理	31~35	4600~5000	30
销售	低级管理	26~30	2600~3000	40
销售	低级管理	31~35	3100~3500	40
生产	低级管理	21~25	4600~5000	20
生产	高级管理	31~35	6600~7000	5
生产	低级管理	26~30	4600~5000	3
生产	高级管理	41~45	6600~7000	3
财务	高级管理	36~40	4600~5000	10
财务	低级管理	31~35	4100~4500	4
行政	高级管理	46~50	3600~4000	4
行政	低级管理	26~30	2600~3000	6

(1) 针对表 11-7，设计一个遗传算法，分析员工的年龄、部门与工资的关系。

(2) 利用粗糙集技术对表 11-7 的数据进行分析，讨论可能得到什么结论。

11-3 在超市中的商品价格都是大于等于零的，超市的总经理只关心如何利用送一件免费商品而带来 1000 元以上的总销售量。讨论如何挖掘这种商业模式。

11-4 现在需要购买一个商品化的数据挖掘工具，从多角度对其进行分析，例如可以处理的数据类

型、系统的体系结构、数据源、数据挖掘功能、数据挖掘方法、与数据仓库的耦合情况、用户的图形界面等。对该系统进行一个实际的评价，并且描述其具体的实现方法。

11-5 遗传算法的主要思路是什么？其中的变异操作有什么作用？

第12章

其他数据挖掘技术和工具

引言

统计类数据挖掘工具与知识类数据挖掘工具主要面向以结构化数据为主的数据库和数据仓库。近年来,随着各种数据处理工具、先进的数据库技术与因特网技术的迅速发展,出现了大量结构各异的、形式复杂的数据,例如文本结构数据、超文本结构数据、非结构化数据和多媒体数据。面对这些数据,数据挖掘技术遇到了新的挑战,需要采用各种与常规数据挖掘技术相异的数据挖掘技术,解决这些结构相异的数据挖掘问题。

通过本章学习,可以了解:

- ◆关于文本数据挖掘的概念与处理方法
- ◆关于 Web 挖掘的基本概念与处理方法
- ◆关于各种挖掘技术在分类处理中的应用
- ◆关于可视化数据挖掘、地理信息系统与空间数据挖掘的概念
- ◆关于分布式数据挖掘的处理技术

12.1 文本挖掘技术

从原则上讲,可在任何类型的信息存储上进行数据挖掘,从而提取知识。采用数据挖掘技术从关系数据表等格式化的数据中获取知识,是一种很常见的行为。然而在现实世界中,知识不仅以传统数据库中的结构化数据的形式出现,也以各种各样形式表现,诸如书籍、研究论文、新闻文章、Web 页面和电子邮件等。由于在这些非结构化数据源中存在着大量的知识,因此可从这些数据源上进行数据挖掘,提取知识成为数据挖掘中的一个研究热点。文本挖掘技术就是解决这个问题方法。

大部分的数据挖掘工具是在集成的、一致的和清理过的、也就是“二手”的数据上运行。如关系数据库的数据挖掘主要与存储在经过净化和一定处理的关系数据库中进行,并且实质上与结构化的数据协同工作。文本挖掘则与存储在文本文档的非结构化集合中的信息——这些信息通常是未经过处理的“一手”数据——协同工作。联机文本挖掘,就是将网上的非结构化文本数据仔细搜索一遍,从中获得知识。

信息检索技术是一项成熟的处理文本数据的技术。随着网络和联机出版的发展,文本数据出现爆炸性增长,传统的信息检索技术已经不能适应现在数据处理的需要。现在所能获得的文本信息集合是如此巨大,以至于不能进行简单的阅读和分析。在现今海量的文本数据中,只有很少的一部分与某个用户相关。不清楚文档的内容,就很难形成有效的查询。文本挖掘可以完成不同文档的比较,以及文档重要性和相关性排列,或者找出多文档的模式及趋势。

12.1.1 信息检索系统

信息检索领域与数据库领域是并行发展的领域。信息检索领域中所用的传统模型是信息被组织成文档,且是信息量巨大的文档,而这些文档数量又十分庞大。信息检索的过程就是根据用户的输入,例如关键词或示例文档,查找相关文档的过程。

信息检索系统的典型例子是联机图书目录和联机文档管理系统,比如存储报纸上文章的文档管理系统。在这种系统中,数据被组织成一系列文档,例如,报纸上的文章或图书馆目录中的条目。系统使用者可以检索特定的一个或一类文档。所要查找的文档通常由一组关键词描述,例如关键词“数据挖掘”可能用来标识有关数据挖掘的图书。文档与一组关键词相关联,当用户输入关键词后,那些包含用户输入关键词的所有文档将被检索出来。

信息检索系统和数据库系统处理的是不同类型的数据。数据库系统处理的是在相对复杂的数据模型基础上组织起来的结构化信息,而信息检索系统一直使用的是一种简单

的模型。在这种模型中,数据库中的信息被组织成一系列非结构化的文档。这样,数据库中的一些常见问题并不出现在信息检索系统中,如并发控制、恢复、事务管理和更新。同样,信息检索系统处理的某些问题在数据库系统中也未得到充分的重视。例如,信息检索领域中处理管理非结构化文档的问题,比如用关键词进行模糊查询;以及处理基于查询文档的相关程度检索文档的问题。

信息检索领域一般用查全率和查准率,对检索的效果进行量化评价。设与查询相关的所有文档集合记为 A , 系统检索出来的所有文档集合记为 B , 既相关又被系统检索出来的文档集合记为 C (参见图 12.1), 则查准率 (precision) 为度量系统检索出来的相关文档与系统检索出来的所有文档百分比

$$\text{Precision} = C/B \quad (12.1)$$

查全率 (recall) 则是系统检索出来的相关文档和与查询相关的所有文档的百分比

$$\text{Recall} = C/A \quad (12.2)$$

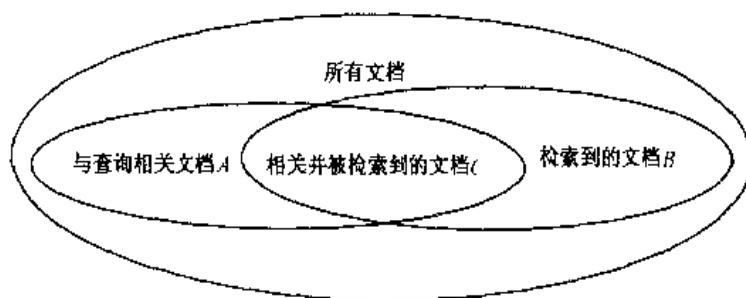


图 12.1 相关文档集和检索到的文档集之间的关系

1. 基于关键字和基于相似性的检索

(1) 基于关键字的检索

在基于关键字的信息检索系统中,文档被看成字符串,可用一组关键字加以识别。用户提供一个关键字或一组由关键字构成的表达式,如“计算机 and 制造”,由关键字进行查询。这样,用户可以找出包含关键字“计算机 and 制造”的全部文档。在基于关键字的信息检索系统中,还要考虑“同义词问题”。如在用关键字“计算机 and 制造”查找有关计算机制造的相关文档时,如果一个文档的关键字是“计算机 and 生产”,那么,虽然它也是和计算机制造有关的文档,但是,它不会被检索出来。可以采用同义词的方法解决这个问题。对每个词都定义一个同义词,比如定义“制造”的同义词为“生产”。那么,再用关键字“计算机 and 制造”进行检索时,关键字为“计算机 and 制造”和“计算机 and 生产”的文档都将被找出来。基于关键字的信息检索系统还有一个难题,就是“多义词问题”,即同一个关键字,在不同的上下文中可能有不同的含义。遗憾的是,目前这种基于上下文确定关键字含义的检索系统还不成熟。

(2) 基于相似性的检索

某些信息检索系统允许基于相似性的检索。这时,用户可给系统一个文档 A,然后要求系统找出与 A “相似”的文档。两个文档的相似性可以自定义,例如根据一组共同的关键词作为相似性。此类检索的输出应当基于相关度,其中相关度的度量是根据关键词的近似性,关键词的出现频率等。如果与 A 相似的文档非常多,系统可以只呈现给用户其中几个,并且允许用户从中选择最相关的那些文档,然后根据选出的文档和文档 A 的相似性开始一个新的检索。

2. 文档的索引

一个高效的索引结构,对于信息检索系统查询的高效处理是十分重要的。系统可以采用倒排索引定位,包含关键词的文档。倒排索引是一种索引结构,它包含两个索引表:文档表和词表。其中文档表由一组文档记录组成,它包含两个字段: doc_id 和 posting_list,其中 doc_id 是文档的标识;posting_list 是出现在文档中的词(或指向词的指针)的列表,按一定的相关度排序。词表由一组词记录组成,也包含两个字段: term_id 和 posting_list。其中 term_id 是词的标识,posting_list 是包含该词的文档标识的列表。通过这种组织,可以很容易地回答类似查询“找出与给定词集相关的所有文档”,或“找出与指定文档相关的所有的词”。例如,找出与一组词相关的所有文档,可先找出每个词在 term_table 中的文档标识列表,然后取其交集,结果是一组相关文档。实际中倒排索引被广泛地使用。它易于实现,但是不能适应对同义词和多义词的处理。posting_list 可能非常长,使得存储开销很大。

文档的全文索引用文档中的每个词作为关键词。文本文档中的“a”,“the”,“for”,“with”之类的词,或中文中的“的”、“地”、“得”、“是”、“和”等词虽然出现的频率都很高,但是没有索引价值。所以在建索引时,要从文档中去掉这些词,并且这些词也不能作为查询中的关键词。

12.1.2 文本分析和语义网络

1. 文本分析

文本分析是实现文本挖掘的关键技术。几十年来,科学家一直努力使计算机理解自然语言,文本分析就是一个努力的方向。文本分析就是文本信息的自动分析,是让计算机从自然语言文档中获得语义。文本分析可以用于如下 4 个方面。

(1) 为一个大型文本集合提供内容概况

文本分析可为一个大型文本集合提供内容概况,例如发现一个客户反馈集中文档的显著簇。这样,可能发现公司的产品或服务在哪里需要改进。

(2) 指出对象间的隐藏结构

在组织一个企业内部网站时,文本分析可以找出对象间的隐藏结构。这样,有关联的文档就能被超链接连接起来。

(3) 提高发现相似或相关信息搜索过程的效率和有效性

可从一个新闻服务机构搜索文章和发现独有的文档,这些文档含有到现在为止在别的文章中没有提到过的新趋势或技术的线索。

(4) 侦察存档中的重复文档

文本分析可以用于大量文本需要分析的地方。虽然自动处理不能达到人类阅读分析的深度,但它可以用来抽取关键点、产生总结或分类文档等。

2. 语义网络

一个有效文本分析的第一步,是创建该文本的一个语义网络。一个语义网络是一系列来自分析的文本的最重要概念(词与词的组合),以及文本中这些概念间的语义联系。一个语义网络为分析的文本提供一个简明和非常准确的总结。与人工神经网络一样,语义网络的每个元素——概念都被它的权重和一组与此网络其他元素的联系所标识——一个上下文结点。一旦为调查研究的文本构造的一组准确的语义网络建立起来,所有文本分析任务就可执行。

语义网络的建立使许多任务的自动执行成为可能,如文本提炼、聚集一个文档集、分类新来的文档、在一组文本或网上“聪明”地搜索语义信息、创建一个易于操纵且定制的知识库、为电子书籍提供一个对用户友好的浏览机制等。

在现有的大部分算法中,一个语义网络是在一些已定义的规则和概念的基础上建立起来的,不过,也存在一些比较强大的算法。这些算法不需要任何关于主题的预先背景知识,可以仅在一个调查研究文本的基础上完全自动地建立起一个语义网络。

12.1.3 文本挖掘

在当今世界,一个人或一个组织所能获得的文本信息集合十分巨大,而且文本信息集合还在不断地更新和增加(特别是基于网络的信息)。这样,信息检索等技术就不能适应现今文本信息处理的需要。所以,必须用文本挖掘技术来解决这一难题。文本挖掘可对大量文档集合的内容进行总结、关联分析、分类和聚类分析等。

1. 文本总结

文本总结是从文档中抽取关键信息,用简洁的形式摘要或解释文档内容。这样,用户不需要浏览全文就可了解文档或文档集合的总体内容。文本总结在有些场合十分有用,

例如搜索引擎在向用户返回查询结果时,可以给出文档的摘要,以便用户的理解。

2. 基于关键字的关联分析

这类分析首先收集经常一起出现的关键字或词汇,然后找出其关联或相互关系。在这类分析中,每个文档被视为一个事务,文档中的关键字组可视为事务中的一组事务项。这样,这种基于关键字的关联分析就变成事务数据库中事务项的关联挖掘问题。

一组经常连续出现或紧密相关的关键字可以形成一个词或词组。关联分析有助于找出复合关联,即领域相关的词或词组,如[中国,长江,三峡]。还有助于找出非复合关联,即领域不相关的词或词组,如[人民币,交易,总额,证券,佣金,参股]。基于这些词或词组关联的挖掘被称为“词级关联挖掘”。利用这种词和词组的识别,词级挖掘可以找出词或关键字之间的关联。

3. 文档分类分析

分类的概念是在已有的数据基础上学会一个分类函数或构造一个分类模型。对文档进行分类有利于对文档的检索和分析。在文档的分类分析中,一般的做法是先把一组预先分类过的文档作为训练集,然后分析训练集,以便得出分类模式。这种分类模式一般需要经过一定的测试过程,不断地细化。最后,用分类模式对其他文档加以分类。

常用的一种对文档分类的有效方法基于关联的分类。这种分类方法基于一组相关联的、经常出现的文本模式对文档加以分类。

(1) 提出关键字

通过简单的信息检索技术或关联分析技术,提出关键字或词汇。

(2) 生成关键字和词的概念层次

使用已有的词类,或基于专家知识,或用关键字分类系统,生成关键字和词的概念层次。训练集中的文档可以分类为类层次结构。

(3) 使用词级关联挖掘方法发现一组关联词

它可以最大化地区分一类文档和另一类文档。这样,每类文档由相关一组关联规则表示。这些分类规则可以基于其出现频率和识别能力,加以排序,并且用于对新的文档进行分类。

在对 Web 文档进行分类时,由于超链接包含有关页面内容的高质量信息,可用这些信息帮助对 Web 文档进行分类。

4. 文档聚类分析

文档聚类是把文档集分成不同组的完全自动的过程。文档聚类与分类的不同之处在于,聚类没有预先定义好的主题类别,它的目标是将文档集合分成若干个组,要求同一

组内文档内容的相似度尽可能大,而不同组间的相似度尽可能小。当文档的内容作为聚类的基础时,不同组是对应于集合中讨论的不同主题或论题。因此,聚类是找出集合所含内容的一条途径。为了帮助识别一组主题,聚类工具可以识别在此组文档中频繁出现的术语或词的列表。聚类也能根据文档的属性集实施,例如它们的长度、日期等进行聚类。

5. 文本挖掘的应用

(1) 电子邮件的管理

电子邮件管理是文本挖掘应用的成功案例。利用文本挖掘构造的电子邮件路由,可对电子邮件进行文本挖掘以后,确定由哪个部门、哪个人来处理这些电子邮件,并且可以根据电子邮件的内容进行相关统计。

(2) 文档管理

文档管理是许多组织中十分烦琐而又重要的工作,通过文本挖掘可以帮助组织对成千上万的文档实现有效的管理,可使组织很快地了解需要查找文档的所在位置,以及其包含的主要内容。

(3) 客户自动问答系统

企业可用文本挖掘建立一个客户自动问答系统,对客户所邮寄的信件、电子邮件进行文本挖掘以后,根据其反映的主要问题,能够确定客户的需求置信度后,就可以自动给客户发送合适的回信。

(4) 市场研究

企业可用联机文本挖掘系统对因特网上所出现的特定词、概念和主题进行挖掘统计,可对市场进行客观的统计分析。

(5) 情报收集

企业可用一些具有文本挖掘功能的自动智能网络爬虫,收集与企业有关的市场、竞争对手和市场环境的信息,并且给出总结性的分析报告。

6. 文本挖掘工具

目前在市场上已经出现许多文本挖掘工具,例如 Automony 的 Agentware, IBM 的 Intelligent Miner for Text 和 Megaputer 的 TextAnalyst 等。

Automony 的 Agentware 可为用户提供一个完全自动和精确的分类、交叉验证和表示信息的方法,能够监视特定的因特网和企业内部网、新闻网站和内部文档库,并且生成关于这些被监视对象的变化报告。

IBM 的 Intelligent Miner for Text 是一个软件开发工具包,用其开发出的应用程序能从网页和联机新闻服务等文本来源获得信息,并且具有从文本中抽取模式、按照主题组

织文档和搜索匹配给定题目文档的能力。

Megaputer 的 TextAnalyst 是实现一种独特神经网络技术的智能文本挖掘和语义信息搜索系统, 可以实现自然语言文本的结构化处理, 可以用于创建知识库、搜索语义信息和自动抽取文本。

12.2 Web 挖掘技术

随着因特网/内联网技术的发展, 尤其是 Web 的全球普及, 使得 Web 上的信息量无比丰富, 越来越多的机构和个人在网络上发布信息、查找信息。网络成为人们获得信息的必要途径和重要手段。网络在给人们带来方便的同时, 也带来了许多问题。Web 上的数据是海量的, 同时, Web 是无结构的、动态的, 页面极其复杂。这样, 使人们从成千上万的 Web 站点中找到有用的数据变得比较困难。

人们越来越关注如何开发和利用 Web 上的数据资源, 开发了各种搜索引擎。但是, 基于关键字的搜索引擎目前存在着许多缺陷: 其覆盖面有限、误差率和漏查率高、检索速度也不理想。目前功能最完善的搜索引擎也只能找到 Web 网址的 1/3 网页, 而且无论怎么选择关键词, 都会返回大量并不需要的结果。同时, 由于许多与话题有关的文档并不包含关键词, 所以搜索引擎不能找出它们。这表明目前 Web 搜索引擎对 Web 资源的查找还存在缺陷。

Web 挖掘 (Web mining) 是解决上述问题的一个途径。Web 挖掘就是利用数据挖掘技术从 Web 文档和 Web 活动中抽取人们感兴趣的、潜在的有用模式和隐藏的信息。

12.2.1 Web 的特点

了解 Web 挖掘, 先要了解 Web 的特点, Web 是一个非常成功的基于超文本的分布式信息系统。Web 目前涉及新闻、广告、消费信息、金融管理、教育、政府、电子商务等许多信息服务。Web 还包含丰富和动态的超链接信息, 以及 Web 页面的访问和使用信息, 这为数据挖掘提供丰富的资源。从 Web 的特点分析, 可以发现 Web 挖掘是一项很有挑战性的工作。Web 具有以下 4 个特点。

1. 庞大性

Web 为在全球范围发布和传播信息提供机会, 它允许任何人在任何地方、任何时间传播和获取信息。由于 Web 的开放性, 使得 Web 上的信息与日俱增, 爆炸性增长。到 1999 年年底, 至少有 1600 万台主机连入因特网, 网上的网页数量达到 10 亿, 而且正在以每月近千万的数量增长, 甚至有人预言 Web 页面的数量每隔 100~120 天要翻一番。

2. 动态性

Web 不仅以极快的速度增长, 而且其信息还在不断地发生更新。新闻、公司广告、股票市场、Web 服务中心等都在不断地更新着各自的页面。链接信息和访问记录也在频繁更新之中。

3. 异构性

从数据库研究的角度出发, Web 网站上的信息也可看做一个数据库, 一个更大、更复杂的数据库。Web 上的每个站点就是一个数据源, 每个数据源都是异构的, 这就构成了一个巨大的异构数据库环境。

4. 半结构化的数据结构

Web 上的数据与传统数据库中的数据不同。传统数据库都有一定的数据模型, 可以根据模型来具体描述特定的数据。Web 上的数据非常复杂, 没有特定的模型描述, 每个站点的数据都各自独立设计, 并且数据本身具有自述性和动态可变性。因而, Web 上的数据应具有一定的结构性, 但因其自述层次的存在, 是一种非完全结构化的数据, 这也可以称为半结构化数据。半结构化是 Web 数据的最大特点。

Web 面对的用户群体是形形色色的, 用户有各种背景、兴趣和使用目的。对于一个用户而言, 只关心他想要的 Web 上的很小一部分信息。

由于 Web 的这些固有特点, 从这些分散的、异构的、没有统一管理的海量信息中快速、准确地获取信息, 成为 Web 挖掘所要解决的一个难点, 用于 Web 的数据挖掘技术不能照搬用于数据库的数据挖掘技术。

12.2.2 Web 内容挖掘

Web 挖掘是一个具有挑战性的课题, 它实现对 Web 存取模式、Web 结构和规则, 以及动态的 Web 内容的查找。Web 挖掘可以分为 Web 内容挖掘 (Web content mining)、Web 结构挖掘 (Web structure mining) 和 Web 使用记录挖掘 (Web usage mining) 3 类。

Web 内容挖掘可以说是将数据挖掘技术在网络信息处理上的应用。Web 内容挖掘主要针对各种非结构化的数据, 如文本数据、音频数据、视频数据和图形图像数据等各种数据相融合的多媒体数据挖掘。又可将其分为基于文本信息挖掘和基于多媒体信息挖掘两种数据挖掘方式。

1. 基于文本信息的挖掘

Web 上的内容挖掘多为基于文本信息的挖掘, 它和通常的平面文本挖掘功能和方法

比较类似。平面文本挖掘的方法也可用于 Web 文本的挖掘。Web 文档多为 HTML, XML 等语言, 因此可用 Web 文档中的标记, 如<Title>, <Heading>等额外信息, 利用这些信息提高 Web 文本挖掘的性能。

在对 Web 文档进行分类分析中, 可以基于一组预先分好类的文档, 从预定义分好类目录中为每个文档赋予一个类标签。例如, Yahoo! 的文档及其相关文档可以作为训练集, 用于导出 Web 文档的分类模式, 这一模式可以分类新的 Web 文档。由于超链接包含有关页面内容的高质量信息, 可以利用这些信息对 Web 文档进行分类。这种分类比基于关键字的分类方法更准确、更完美。

2. 基于多媒体信息的挖掘

随着网络带宽的不断加大, 多媒体信息在网上迅速增加, 这就对多媒体信息的挖掘提出了要求。多媒体信息的挖掘主要是指基于音频的挖掘、基于图片的静态图像的挖掘和基于视频的动态图像挖掘。

12.2.3 Web 结构挖掘

Web 结构挖掘是从 WWW 的组织结构和链接关系中推导知识。Web 结构挖掘通过分析一个网页链接和被链接数量, 以及对象建立 Web 自身的链接结构模式。这种模式可以用于网页归类, 且可由此获得有关不同页面间相似度和关联度的信息。Web 结构挖掘有助于用户找到相关主题的权威站点, 并且可以指向众多权威站点的相关主题站点。

搜索某个给定话题的 Web 页面时, 不仅希望得到相关的 Web 页面, 而且希望检索到的 Web 页面是权威 Web 页面。也就是说, 检索到的页面具有高质量, 或对该主题具有权威性。

怎样才能自动找出权威 Web 页面呢? 信息检索领域根据杂志论文的引用情况来评估论文的质量。这种方法的原理是一个作者引用另一篇论文, 可以看做该作者对这篇论文的认可, 从这一点可以得到启发。因为, Web 不仅由页面构成, 且还包含从一个页面指向另一个页面的超链接。超链接包含大量人类潜在的语义, 它有助于自动分析出权威性语义。当一个 Web 页面的作者建立指向另一页面的指针时, 可以看做作者对另一页面的注解, 也就是对另一页面的认可。把一个页面的来自不同作者的注解收集起来, 可以用来反映页面的重要性。可以通过这种方法, 寻找权威 Web 页面。

与杂志的引用率不同, Web 链接结构具有特殊的特征。首先, 不是每个超链接都代表寻找的认可。有些链接是为其他目的而创建的, 例如, 为了导航或为了付费广告。总体上, 若大部分超链接具有认可性质, 就可用于权威判断。其次, 基于商业或竞争的考虑, 很少有 Web 页面指向其竞争领域的权威页面, 如可口可乐不会链接到其竞争对手百

事可乐的 Web 页面。第三,权威页面很少具有特别的描述,如 Yahoo!主页面不会明确给出“Web 搜索引擎”之类的自描述信息。

由于 Web 链接结构的这些局限性,人们提出另外一种重要的 Web 页面,称为 Hub 页面。Hub 页面是指一个或多个 Web 页面,它提供指向权威页面的链接集合。对于一个 Hub 页面来说,它本身可能并不突出,但是却提供了指向某个话题的权威页面的链接。Hub 页面起到隐含说明某话题权威页面的作用。通常,好的 Hub 指向许多好的权威页面;好的权威页面则指有好的 Hub 页面指向的页面。这样,可用 Hub 页面和权威页面之间的这种相互作用,用于权威页面的挖掘和高质量 Web 结构和资源的自动发现。

算法 HITS (Hyperlink—Induced Topic Search) 是 Hub 的搜索算法。

(1) 提交查询词给搜索引擎

将查询词提交普通的基于相识度的搜索引擎,搜索引擎返回 n 个页面,把这 n 个页面作为根集 S 。由根集进一步扩展,加入所有由根集中的页所指的页,以及所有指向根集页的页,扩展为一个更大的集合——基本集 T 。基本集中的所有 Hub 页面为集合 V_1 ,所有权威 Web 页面为集合 V_2 。

(2) 计算页面的权威权重和 Hub 权重

给基本集中的每个页面赋予一个非负的权威权重 a_p 和非负的 Hub 权重 h_p ,且将所有的 a 和 h 值初始为同一常数,如初始值为常数 1。Hub 权重 h_p 和权威权重 a_p 的计算公式为

$$a_p = \sum_{(q \text{ 满足 } q \rightarrow p)} h_q \quad (12.3)$$

$$h_p = \sum_{(q \text{ 满足 } q \leftarrow p)} a_q \quad (12.4)$$

每次计算后,都要对 a_p 和 h_p 做规范化处理

$$a_p = \frac{a_p}{\sqrt{\sum_{q \in V_2} (a_q)^2}} \quad (12.5)$$

$$h_p = \frac{h_p}{\sqrt{\sum_{q \in V_1} (h_q)^2}} \quad (12.6)$$

公式(12.3)表示一个页面的权威权重为所有指向它的页面的现有 Hub 权重之和,公式(12.4)的含义是一个页面的 Hub 权重为该页面链接的所有页面的现有权威权重之和。这两个公式还反映了一个页面若有许多好的 Hub 页面所指,则其权威权重会相应地增加;一个页面若指向许多好的权威页面,则其 Hub 权重也会相应地增加。

(3) 比较、确定页面的权威权重和 Hub 权重

对页面的权威权重和 Hub 权重进行计算并且输出一组具有较大权威权重的页面和具有较大 Hub 权重的页面。

HITS 算法对许多查询具有非常好的搜索结果。基于 HITS 算法的系统由于纳入 Web 链接信息,其查询效果十分明显。

12.2.4 Web 使用记录的挖掘

Web 内容的挖掘和 Web 结构挖掘的对象是网上的原始数据。Web 使用记录的挖掘则不同于前两者,它面对的是在用户和网络交互的过程中抽取出来的第二手数据。Web 使用记录挖掘通过挖掘 Web 日志文件和相关数据,发现用户访问 Web 页面的模式。

因特网的用户一旦连接到一个在线的服务器上,就在这个服务器上留下了一个“脚印”,这就是服务器上的日志文件。它包括所请求的 URL,发出请求的 IP 地址和时间戳。对于基于 Web 的电子商务服务器,保存了大量的 Web 访问日志记录。这些日志记录提供了有关 Web 动态的丰富信息。可以通过对用户留下的这些日志文件进行 Web 数据挖掘,提取有关用户的知识,对用户的访问行为、频度、内容等进行分析,得到关于用户的行为和方式模式。从而改进站点的结构,或为用户提供个性化服务。这方面的研究主要有一般的访问模式追踪和个性化的使用记录追踪两个方向。

一般的访问模式追踪通过分析使用记录了解用户的访问模式和倾向,从而改进站点的组织结构。个性化的使用记录追踪则倾向于分析单个用户的偏好,其目的是根据不同用户的访问模式,为每个用户提供个性化的页面,开展有针对性的服务以满足用户的需求。

Web 使用记录的挖掘通常需要经过数据预处理、模式识别、模式分析这三个阶段。

1. 数据预处理阶段

它主要包括数据清洗和事务识别两个部分。数据清洗主要是对无关记录的删除,判断是否有重要的访问没有被记录,用户的识别等。事务识别是指将页面访问序列划分为代表 Web 事务或用户会话的逻辑单元。

2. 模式识别阶段

这个阶段采用统计法、机器学习等成熟技术,从 Web 使用记录中挖掘知识。实现的算法可以是统计分析、聚类、分类、关联规则和序列模式识别等。对 Web 使用记录的挖掘,早期大多采用统计方法进行。当用户通过浏览器对 Web 站点进行访问时,建立统计模型对用户访问模式进行多种简单的统计,如频繁访问页、单位时间访问数、访问数据量的时间分布图等。

3. 模式分析阶段

这个阶段的任务是采用合适的成熟的技术和工具,进行模式的分析,从而辅助分析人员的理解,使采用各种工具挖掘出的模式得到很好的利用。目前通常采用的方法有两种:一种采用 SQL 查询语句进行分析;另外一种是将数据导入多维数据立方体中,利用 OLAP 工具进行分析并且提供可视化的结果输出。

通过对 Web 日志文件和相关数据的挖掘,可以进行系统设计、Web 页面交换、Web 页面预取,认识 Web 信息访问的本质,理解用户的反映和动机,发现用户的行为模式,从而改进 Web 页面的设计和 Web 应用程序,发现潜在的客户、用户和市场等。

12.2.5 Web 数据挖掘的应用

数据挖掘技术已经广泛应用于金融业、零售业、远程通信业、政府管理、制造业、医疗服务和体育等行业中,而它在网络中的应用,即 Web 挖掘也正在成为一个热点。Web 挖掘的应用涉及电子商务、网站设计和搜索引擎服务等多方面。

1. 电子商务

(1) 客户分类和客户聚类

对 Web 的客户访问信息进行挖掘,对客户进行分类分析,例如根据国家或类型(.com, .edu, .gov)进行分类分析。应用聚类分析对客户进行分组,并且分析组中客户的共同特征。这样,就可以让销售商更好地了解自己的客户,向客户提供更有针对性的服务。

(2) 找到潜在的客户

在对 Web 的客户访问信息的挖掘中,利用分类技术可在因特网上找到未来的潜在客户。通常,获得这些潜在客户的市场策略是先对已经存在的访问者进行分类。对于一个新的访问者,通过在 Web 上的分类发现,识别这个访问者与已经分类的访问者的一些公共的描述,从而对这个访问者进行正确分类,以判断这个新的访问者是否是一个潜在的客户。

(3) 客户的驻留

对于客户而言,传统客户与销售商之间的空间距离在电子商务中已经不复存在,在网上,每个销售商对于客户来说都是一样的。那么,销售商就要尽量使客户在自己的网站上驻留更长的时间。利用 Web 挖掘,就可知道客户的行为模式,了解客户的兴趣及需要,从而根据客户的兴趣及需要动态调整 Web 页面,以更好地满足客户的需要。因为站点上的页面内容的安排和连接如同传统商店中物品在货架上的摆设一样,可以利用 Web

挖掘,找出具有一定支持度和信任度的相关联的物品,并且针对客户的动态变化调整站点的结构,使客户访问关联信息的连接更直接。

2. 网站设计的应用

通过对网站内容的挖掘,主要是对文本内容的挖掘,可以有效地组织网站信息,例如采用自动归类技术实现网站信息的层次性组织。可以结合对用户访问日志记录信息的挖掘,把握用户的兴趣,有助于开展网站信息推送服务以及个人信息的定制服务。例如,有些研究提出了可适应站点的概念,即可通过用户访问模式改进 Web 站点内容。

3. 搜索引擎的应用

通过对网页内容的挖掘,可以实现对网页的聚类 and 分类,实现网络信息的分类浏览与检索;通过用户使用的提问式(query)历史记录分析,可以有效地进行提问扩展,提高用户的检索效果(查全率、查准率);运用 Web 挖掘技术改进关键词加权算法,提高网络信息的标引准确度,改善检索效果。Web 挖掘是目前网络信息检索发展的一个关键。

由于 Web 上存在大量信息,并且 Web 在当今社会经济生活中扮演越来越重要的角色,发挥越来越大的作用。Web 挖掘的应用将越来越广泛,用户对高品质、个性化信息的需求也将进一步推动 Web 挖掘这项技术的研究和开发。Web 挖掘将成为数据挖掘中一个重要和繁荣的研究领域。

12.3 分类分析技术

分类分析的输入集是一组记录集合和几种类别的标记。这个输入集又称示例数据库或训练集。训练集中的记录称为样本。在这个训练集中,每个记录都被赋予一个类别的标记。分类分析就是通过分析训练集中的数据,为每个类别建立分类分析模型。然后用这个分类分析模型对数据库中的其他记录进行分类。分类分析方法的一个典型例子是信用卡核准过程。信用卡公司根据信誉程度,将一组持卡人记录分为良好、一般和较差三类,且把类别标记赋给每个记录。分类分析就是分析该组记录数据,对每个信誉等级建立分类分析模型。如“信誉良好的客户是那些收入在 5 万元以上,年龄在 40~50 岁之间的人士”。得出这个分类分析模型之后,就可根据这个分类分析模型对新的记录进行分类,从而判断一个新的持卡人的信誉等级是什么。

分类分析可以分为以下两个步骤。

第一步:分析训练集中的数据,构造一个分类分析模型。通常,模型用分类规则、决策树或数学公式的形式提供。这个阶段是学习阶段。由于在训练集中每个记录的类别标记都是已知的,分类又称为有监督学习。相应地,聚类称为无监督学习,因为聚类分

析的训练集中记录的类别标记是未知的。

第二步：使用分类分析模型进行分类。在使用分类分析模型对新的记录进行分类之前，要先评估模型的预测准确率。如果认为模型的预测准确率可以接受，就可用它对新的类别标号未知的数据记录或对象进行分类。

分类分析的技术很多，如决策树归纳、贝叶斯分类和贝叶斯网络、神经网络、最近邻分类、遗传算法、基于关联的分类和模糊逻辑技术等。下面讨论决策树与贝叶斯分类在分类分析中的应用。

1. 用决策树归纳分类

决策树(decision tree)是能够被看成一棵树的预测模型。树的每个分支都是一个分类问题，内部节点表示在一个属性上的测试，树叶代表类或类分布。图 12.2 是一个“是否购买计算机”的决策树。它预测×××公司的顾客是否购买计算机。将顾客分为两类：“会购买”和“不会购买”。图中的矩形代表一个内部节点，即代表在一个属性上的测试；椭圆形代表一个树叶，也就是代表着一类(“会购买”或“不会购买”)。

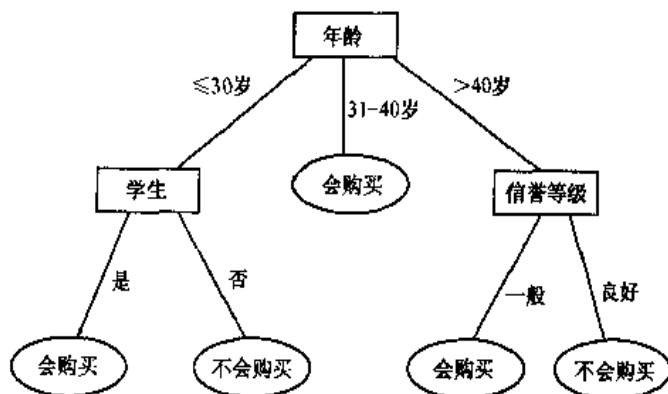


图 12.2 “是否购买计算机”的决策树

从这个决策树中，可以看出什么样的顾客会购买计算机，什么样的顾客不会购买计算机：年龄在31~40岁之间的顾客，年龄小于等于30岁并且是学生的顾客和年龄大于40岁并且信誉等级为“一般”的顾客类别标号是“会购买”，也就是说这类顾客会购买计算机。年龄小于等于30岁并且不是学生的顾客和年龄大于40岁并且信誉等级为“良好”顾客的类别标号是“不会购买”，这类顾客不会购买计算机。也可以用这个决策树，判断一个新的顾客会不会购买计算机。比如，一个新顾客的数据记录是“年龄=21，学生=是，信誉等级=一般”，由这个决策树可以看出，这个新顾客是会购买计算机的。那么公司就给这个顾客发放些促销信息，以吸引这个新顾客购买计算机。

由决策树可以很容易地得到“IF-THEN”形式的分类规则。方法是沿着由根节点到树叶节点的路径，路径上的每个属性-值对形成“IF”部分的一个合取项，树叶节点包含

类预测，形成“THEN”部分。一条路径创建一个规则。图 12.2 的决策树可以转化为以下形式的分类规则

IF 年龄=“≤30” AND 学生=“是”	THEN 类别标记=“会购买”
IF 年龄=“≤30” AND 学生=“否”	THEN 类别标记=“不会购买”
IF 年龄=“31~40”	THEN 类别标记=“会购买”
IF 年龄=“>40” AND 信誉等级=“一般”	THEN 类别标记=“会购买”
IF 年龄=“>40” AND 信誉等级=“良好”	THEN 类别标记=“不会购买”

怎样才能得到这样的一个决策树呢？下面以 ID3 算法为例来说明决策树的创建。

ID3 是最早的决策树一个重要的算法，它是建立在推理系统和概念学习系统的基础上的算法。ID3 算法的基本策略见如下 4 点。

(1) 创建一个节点。如果样本都在同一类中，则算法停止，把该节点改成树叶节点，且用该类标记。

(2) 否则，选择一个能够最好地将训练集分类的属性，该属性作为该节点的测试属性。

(3) 对测试属性中的每个值，创建相应的一个分支，据此划分样本。需要注意的是，在 ID3 算法中，属性的值都是离散的。如果属性值是连续的，需要通过数据变换，把属性值化为离散的。有的时候虽然属性值是离散的，但是离散的值太多；且为每个离散的属性值创建一个分支，对分类分析没有什么明显的改善，那么也把属性值进行变换，如属性“年龄”，一般把年龄的属性值划分为几个区间。

(4) 使用同样的过程自顶向下递归，直到满足下面的三个条件中的一个时，就停止递归：

- (a) 给定节点的所有样本都属于同一类；
- (b) 没有剩余的属性可以用来进一步划分；
- (c) 继续划分得到的改进不明显。

在 (b)，(c) 这两种情况下，以在该节点下的样本中的大多数的类别标号作为该节点的类别标号，创建一个树叶节点。

决策树算法的一个重要问题是在树的各个内部节点处寻找一个属性，该属性最好能将训练集进行分类。ID3 通过划分提供的信息增益选择测试属性。信息增益被定义为原始分割的熵与划分以后各分割的熵累加得到的总熵之间的差。也就是说，信息增益是指划分前后进行正确预测所需的信息量之差。信息增益越大，则划分后预测所需的信息越少，这就表明较好地降低了划分前的无序度。因此，需要选择具有最高信息增益的属性作为当前节点的测试属性。

如何根据 ID3 算法创建出图 12.2 这样的决策树？首先对训练集的数据进行预处理，

得到表 12-1 的数据记录。

表 12-1 经过数据预处理的数据记录

编号	年龄/岁	学 生	身体素质	是否会买
1	≤ 30	是	良好	会购买
2	≤ 30	是	一般	会购买
3	> 40	否	一般	会购买
4	> 40	否	良好	不会购买
5	> 40	否	一般	会购买
6	31~40	是	一般	会购买
7	≤ 30	否	良好	不会购买
8	> 40	是	一般	会购买
9	≤ 30	否	良好	不会购买
10	> 40	否	良好	不会购买
11	≤ 30	否	一般	不会购买
12	31~40	是	一般	会购买
13	31~40	否	一般	会购买
14	31~40	是	良好	会购买

先计算用每个属性进行划分的信息增益。“年龄”在各个属性中具有最大的信息增益，所以选择“年龄”属性作为第一个测试属性，创建一个节点，用“年龄”标记，且对每个属性值引出一个分支，将“年龄”的属性值划分为三个区间：“ ≤ 30 ”，“31~40”“ > 40 ”。共引出三个分支：“年龄”在“31~40”区间的顾客属于同一类，即“会购买”，所以要在该分支的端点创建一个树叶节点；对于“年龄”在“ ≤ 30 ”区间的分支，继续进行计算，计算剩余各个属性的相应的信息增益，选择信息增益最大的属性作为测试属性，这时信息增益最大的是“学生”属性，创建一个节点，用“学生”标记，且对每个属性值引出一个分支，应当引出两个分支。“学生”属性值等于“是”的这个分支的顾客属于同一类：“会购买”。因此，要创建一个树叶节点，用“会购买”标记。另一个分支的顾客都属于同一类：“不会购买”，也要创建一个树叶节点，用“不会购买”标记。对于“年龄”属性值在“ > 40 ”区间的分支也同样处理。最后得出如图 12.2 所示的决策树。

建好决策树并不意味着决策树的完成，还要检查模型是否过适应数据。也就是说，模型是否过度贴近训练集的特性了。如果模型过适应数据的话，对其他的数据集数据就不一定奏效了。因为在决策树构造时，有些分支反映的是训练集中的噪声或孤立点。如果模型过适应数据的话，可以用剪枝方法加以处理。树剪枝方法使用统计度量，检测和剪去这种不可靠的分支，这样，可以提高在未知数据记录上分类的准确性。

2. 贝叶斯分类

贝叶斯分类是统计学的分类方法。贝叶斯分类基于贝叶斯公式，即后验概率公式。

贝叶斯分类的分类过程是，首先令每个数据样本用一个 n 维特征向量 $X=\{x_1, x_2, \dots, x_n\}$ 表示，其中 x_k 是属性 A_k 的值。所有的样本分为 m 类： C_1, C_2, \dots, C_m 。对于一个类别的标记未知的数据记录而言，若

$$P(C_i|X) > P(C_j|X) \quad 1 \leq j \leq m, j \neq i$$

也就是说，如果在条件 X 下，数据记录属于 C_i 类的概率大于属于其他类的概率的话，贝叶斯分类将把这条数据记录归类为 C_i 类。

由贝叶斯公式可知：

因为 $P(X)$ 是常数，那么 $P(X|C_i) \cdot P(C_i)$ 最大的话， $P(X|C_i)$ 就最大。一般来讲，计算 $P(X|C_i)$ 的开销很大。为了降低计算的开销，可以做个假定。假定属性值相互条件独立，也就是说，假定属性之间不存在依赖关系。这样一来，概率 $P(x_1|C_i), P(x_2|C_i), \dots, P(x_n|C_i)$ 由训练样本估值

$$P(x_k|C_i) = s_{ik}/s_i, \quad 1 \leq k \leq n$$

$$P(C_i) = s_i/s$$

s_{ik} 是在属性 A_k 上具有值 x_k 的类 C_i 的样本数， s_i 是 C_i 中的样本数， s 是总样本数。

以表 12-1 的数据作为训练样本，用贝叶斯分类给一个新的数据记录分类。设类 C_1 对应的类别标号为“会购买”，类 C_2 对应的类别的标号为“不会购买”。已知新的数据记录为：“年龄”=“21”，“学生”=“是”，“信誉等级”=“一般”，则

$$X = (\text{“年龄”} = \text{“21”}, \text{“学生”} = \text{“是”}, \text{“信誉等级”} = \text{“一般”})$$

看一下 $P(X|C_1) \cdot P(C_1)$ 和 $P(X|C_2) \cdot P(C_2)$ 哪一个比较大，那么新数据记录就属于哪一类。

$$P(C_1) = 9/14 = 0.64$$

$$P(C_2) = 5/14 = 0.36$$

$$P(\text{“年龄”} = \text{“≤30”} | C_1) = 2/9 = 0.22$$

$$P(\text{“年龄”} = \text{“≤30”} | C_2) = 3/5 = 0.60$$

$$P(\text{“学生”} = \text{“是”} | C_1) = 6/9 = 0.67$$

$$P(\text{“学生”} = \text{“是”} | C_2) = 1/5 = 0.20$$

$$P(\text{“信誉等级”} = \text{“一般”} | C_1) = 6/9 = 0.67$$

$$P(\text{“信誉等级”} = \text{“一般”} | C_2) = 2/5 = 0.40$$

则

$$P(X|C_1) \cdot P(C_1) = 0.22 \times 0.67 \times 0.67 \times 0.64 = 0.06$$

$$P(X|C_2) \cdot P(C_2) = 0.60 \times 0.20 \times 0.40 \times 0.36 = 0.02$$

因为 $P(X|C_1) \cdot P(C_1) > P(X|C_2) \cdot P(C_2)$, 也就是 $P(C_1|X) > P(C_2|X)$, 在 X 的条件下, 属于 C_1 的概率大于属于 C_2 的概率, 所以要把 X 归类为 C_1 类。这样, 由贝叶斯分类得出这个顾客类别标号是“会购买”。

12.4 可视化数据挖掘技术

可视化数据挖掘是用数据可视化技术, 从大的数据集中将数据转换为图形或图像, 在屏幕上显示出来, 并且进行交互处理的理论、方法和技术。利用可视化可以很清楚地发现隐含的和有用的知识, 可视化数据挖掘是从大量数据中发现知识的有效途径。可视化数据挖掘可以看做数据可视化和数据挖掘这两个学科的融合, 它也和计算机图形学、图像处理、计算机视觉、多媒体系统、人机接口、模式识别及高性能计算紧密相关。

12.4.1 数据可视化技术

数据可视化的概念来自科学计算可视化。随着计算机技术的发展, 数据可视化的概念大大扩展。可视化技术可分为体可视化技术和信息可视化技术两类。体可视化技术是空间数据场的可视化。科学计算数据的可视化、工程数据的可视化和测量数据的可视化都是体可视化。信息可视化则是指非空间数据的可视化。在科学计算可视化中, 显示的对象涉及标量、矢量及张量等不同类别的空间数据, 研究的重点放在如何真实、快速地显示三维数据场等方面。在信息可视化中, 显示的对象主要是多维的标量数据, 研究的重点是设计和选择什么样的显示方式, 才能便于用户了解庞大的多维数据以及它们相互之间的关系, 其中很多地方涉及心理学和人机交互技术等问题。

数据可视化技术的主要特点是可视性、多维性和交互性。可视性是指数据可以用图像、曲线、二维图形、三维图形和动画来显示。多维性是指可以看到表示对象或事件的数据的多个属性或变量, 可将数据每一维的值分类、排序、组合和显示。交互性是指用户可以方便地以交互的方式管理和开发数据。

数据可视化的应用非常广泛, 可以用于自然科学、工程技术、金融、商业和通信等各种领域。体可视化技术已经成功地应用于医学、油气勘探、气象预报和工程等领域。信息可视化在商务、金融和通信等领域, 有着十分广阔的应用前景。

随着社会信息化的推进和网络应用的日益广泛, 信息源越来越庞大。数据量呈爆炸趋势, 人类的视觉系统和大脑不具备处理这么多数据的能力, 海量的数据只有通过可视化变成形象, 才能激发人的形象思维。利用数据可视化技术有助于加深人类对数据含义的理解, 可以大大加快数据的处理速度; 可在人与人、人与数据之间实现图像通信, 从而使人类能够发觉用其他方式不能发现的数据中隐含的模式; 可对计算、编程或其他过

程实现引导和控制, 通过交互式手段改变过程所依据的条件, 并且观察影响程度。

12.4.2 可视化数据挖掘技术

数据挖掘技术是从大量的、不完整的、有噪声的和不一致的数据中提取隐含的、潜在的、有用的信息和知识的过程。可视化数据挖掘是可视化技术和数据挖掘技术的融合。人类的大脑可以看做一个强有力并且高度并行的处理和推理引擎, 它带有一个大的知识库。可视化数据挖掘可以有效地利用人类的大脑。把可视化技术应用到数据挖掘之中, 有助于人类更好、更方便地理解数据的含义, 使用户能在较高的抽象层次上观察数据, 方便用户找出潜在模式, 更好地理解数据挖掘的结果等。

可视化数据挖掘可以分为以下 4 个方面。

1. 数据可视化

数据库和数据仓库中的数据可以看做具有不同的粒度或不同抽象级别, 也可看做由不同属性和维组合起来的。数据能够用多种可视化方式进行描述, 如盒状图、三维立方体、曲线、曲面、数据分布图表、连接图等等。这种数据的可视化显示能把数据库或数据仓库中数据特性的总体印象提供给用户, 并且可让用户明白从哪里开始挖掘。例如, RightPoint 公司生产的 DataCruncher 数据挖掘工具具有数据可视化功能 (见图 12.3), 当用户选择左边数据框内有关数据后, 其右边可用直方图、折线图等方式显示数据的图形表示。

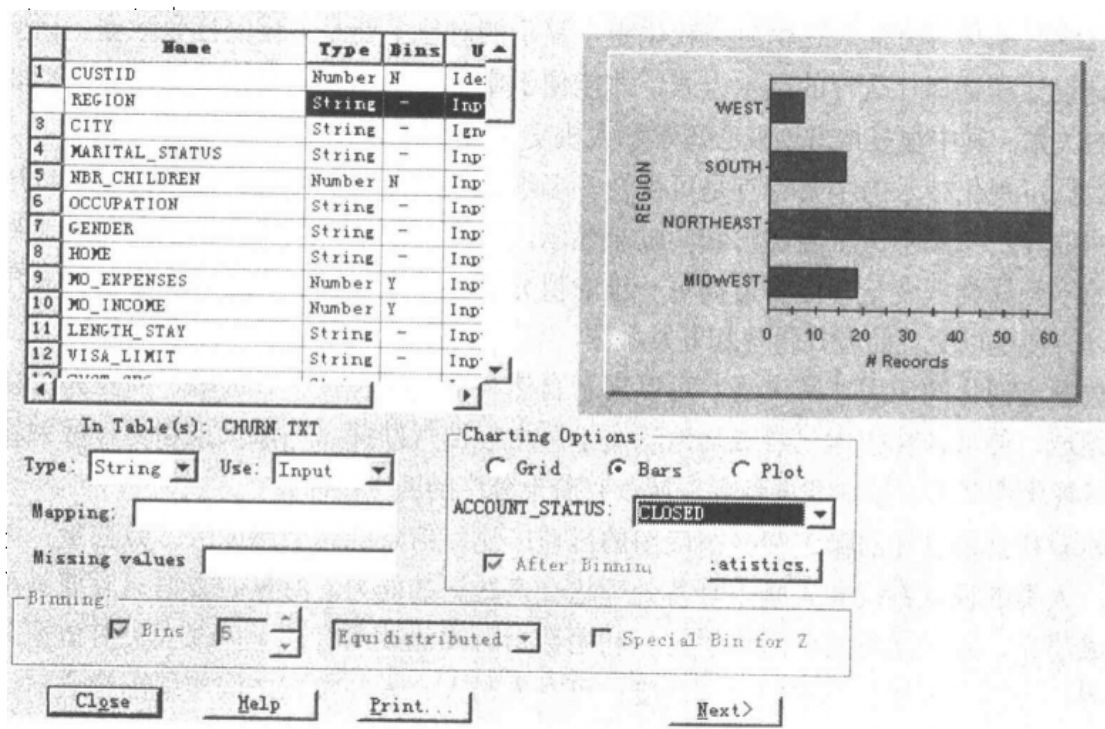


图 12.3 数据的可视化

2. 数据挖掘结果可视化

数据挖掘结果可视化是将数据挖掘后得到的结果，用可视化的形式表示出来，如表示为散列图、盒状图等形式。决策树、关联规则、概化规则等也可通过可视化描述。这样便于用户理解数据挖掘的结果。图 12.4 为可视化的数据挖掘结果，是 DataCruncher 数据挖掘工具中的饼图。图 12.5 为数据挖掘结果的 3D 图形，即用 3D 图形表示 DataCruncher 数据挖掘工具进行数据挖掘的结果。

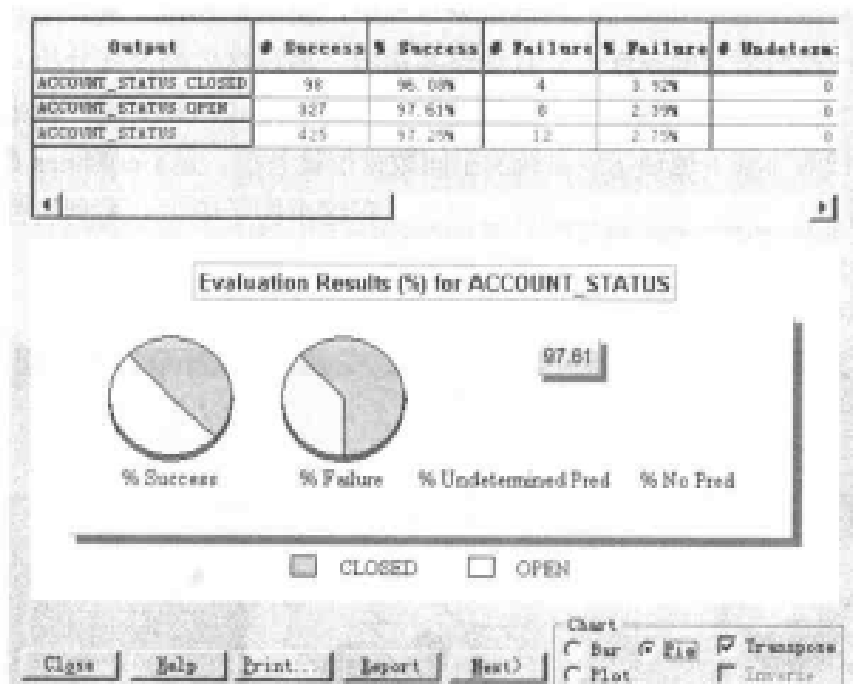


图 12.4 可视化的数据挖掘结果

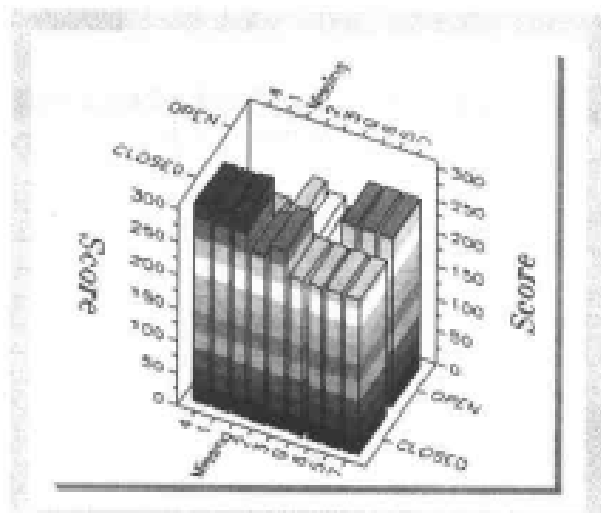


图 12.5 数据挖掘结果的 3D 图形

3. 数据挖掘过程可视化

数据挖掘过程可视化是用可视化过程描述数据的挖掘过程。这样，用户可以看出数据是从哪个数据库或数据仓库中取出来的，怎么抽取的，以及怎样清理、集成、预处理的，怎样挖掘的，甚至还可看到数据挖掘采用的方法，结果存储的地址及显示方式。

4. 交互式可视化数据挖掘

可视化数据挖掘是在交互式的数据挖掘过程中，使用可视化工具。用户可以通过交互式手段改变过程所依据的条件，并且观察其影响。通过这种勘探式分析，在不使用自动数据挖掘技术的情况下，允许用户高效地寻找和发现模式，帮助用户做出正确的数据挖掘决策。图 12.6 为基于地理信息系统的空间数据挖掘系统，是 GeoMiner 数据挖掘工具在数据挖掘过程中的交互式可视化挖掘的例子。在这个例子中，一系列属性的数据分布用彩色扇区表示。这样可以帮助用户决定哪个作为分类的扇区首先被选中，哪个地方是最好的扇区分割点。

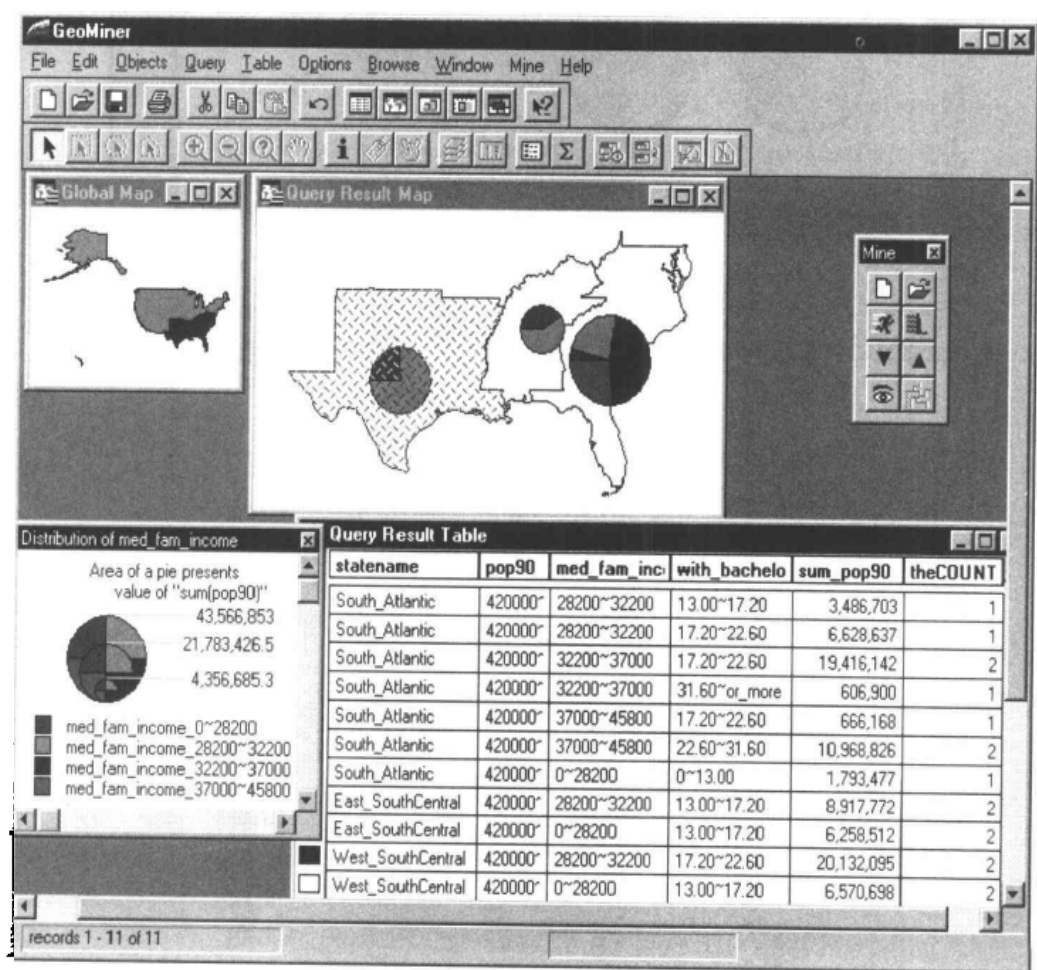


图 12.6 基于地理信息系统的空间数据挖掘系统

总之,数据可视化技术和数据挖掘的结合,有助于解决日益显著的“数据超载”问题,使人类可以方便、快速地从海量的、动态的数据中提取潜在的、有用的知识和信息。

12.5 地理信息系统与空间数据挖掘

12.5.1 地理信息系统

在当今信息化社会中,谁能更有效地利用资源,掌握更全面、更准确的信息,更快做出准确的决策,谁就能够顺应社会的潮流,在激烈的市场竞争中站稳脚跟。对于全球化的大企业或者国内大型企业来说,因为其本身是跨地域的机构,有遍布全球或全国的客户,其生产经营活动遍布世界各地或全国各地,不同的企业资源之间有着复杂的网络和层次关系。所以大企业面临的一个问题是如何简单、有效地管理庞大的资源?对于企业来说,面对激烈的市场竞争,需要巩固已有的市场,寻找新的市场。这就要求对市场进行全面的分析。在建设企业内部网体系的时候,企业面临的问题是如何把分布在各地以至全世界的有关数据和信息统一到内部网中?面对庞大的分布的信息数据如何做出正确的决策?对于政府机关,各级政府机关的业务本身就是分布性的,面对下级各机关的各种统计数据,如何直观地显示对比关系,获取隐藏在数据后面的事实?地理信息系统是解决上述问题的一条有效途径。

1. 地理信息系统概念

(1) 地理信息

地理信息是与研究对象的空间地理分布有关的信息,它表示地表物体及环境固有的数量、质量、分布特征、联系和规律。地理信息属于空间信息,其位置的识别与数据联系在一起的,具有区域性。地理信息又具有多维结构的特征,即在同一个位置上具有多个专题和属性的信息结构。例如在一个地面点位上,可取得高度、噪声和污染和交通等多种信息。地理信息还有很强的时序特征,也就是具有动态变化的特征。

(2) 地理信息系统

地理信息系统(GIS, Geographical Information System)过去被认为是一项专门技术,仅用于测绘、环境及资源管理等领域。它被定义为一种特定而又十分重要的空间信息系统,以采集、存储、管理、分析和描述整个或部分地球表面(包括大气层在内)与空间和地理分布有关的数据的空间信息系统。计算机制图、计算机辅助设计、数据库管理系统、遥感图像处理技术奠定了地理信息系统的技术基础。地理信息系统是这些学科的综合。

随着计算机技术的迅速发展和社会需求的不断扩大,地理信息系统的应用领域不断

扩大, 扩展到航天、电信、电力、交通运输、商业、市政基础设施管理、公共卫生及安全、油气及其他矿产资源的勘测等诸多领域。特别是地理信息系统 (GIS) 与管理信息系统 (MIS) 相结合, 其应用几乎可以覆盖人类生活的各个方面。

地理信息系统分为两种: 地理地图 GIS 和 GIS 应用系统。前者以解决地理学领域的应用问题为研究目标; 后者是一种基于地理信息的设备和生产管理的计算机图文交互系统, 也是一种将图文技术和数据库管理技术相结合的计算机应用系统。对于 GIS 应用系统而言, 地理信息只是作为背景, 提供地理位置参照体系。

2. 地理信息系统的特点

地理信息系统和其他信息系统的差异在于: 它将空间和属地信息有机的结合起来, 从空间和属性两个方面对现实对象进行查询和分析, 且将结果以各种直观的形式准确、形象地表达出来。结合空间数据与属性数据, 可把数据存储与管理一体化。这样, 可以降低数据结构的复杂性, 减少开发和维护费用, 且能显著地提高工作效率。

地理信息系统中的数据量一般很大, 需要数据库来管理, 因此它是一个数据库系统。

地理信息系统一般地理范围跨越广, 系统运行的环境一般是广域网 (城域网), 因此它是一个分布式系统。

地理信息系统可让大量枯燥的分布式数据变成直观的图表和对比关系, 因此它又是一个可视化系统。

3. 数据挖掘技术和地理信息系统相结合

地理信息系统本身的特点, 决定了它很适合地理位置跨越广的企业、政府机构或其他机构的数据资源管理。当今社会的全球化, 使得地理信息系统越来越重要。运用地理信息系统可以提高资源的管理水平, 改善各部门之间的交流, 增加和社会的接触, 使管理走向现代化, 大大提高信息的可视化水平和决策支持能力。数据挖掘技术是从海量的数据中提取或“挖掘”知识的技术。那么, 数据挖掘技术能否与地理信息系统相结合呢? 答案是肯定的。这样可以丰富数据挖掘技术及工具的功能和性能。采用数据挖掘技术和地理信息系统相结合的技术, 有助于用户对分布式的海量的信息及数据挖掘结果的理解, 加快数据的处理速度, 可以较充分地利用人的视觉系统和大脑, 可使用户高效地寻找和发现模式, 有助于用户发现其他方式所不能发现的潜在模式。

地理信息系统的数据挖掘有如下 4 个特点。

(1) 图形化数据挖掘

可对地图数据逐层钻取, 同时将数据以图形形式表现出来。如观察全国销售增长情况, 知道某一省份取得很高的销售增长; 观察该省的数据, 发现某个地区有很高的销售增长。这样逐层深入, 可以挖掘隐藏在数据下的真正原因。

(2) 图形化统计查询

提供丰富的统计查询方式,如将不同地区某个指标的比较结果,以颜色、柱状图、饼图、点密度图等诸多方式反映在地图上。

(3) 图形化报表输出

可将数据统计结果以图形报表形式打印。

(4) 专业的地理分析功能

具有网络分析,三维分析,路径计算等功能。

12.5.2 空间数据挖掘

空间数据挖掘是对空间数据库中非显式存在的知识、空间关系或其他有意义模式的提取。空间数据挖掘需要综合数据挖掘与空间数据库技术的支持。利用空间数据挖掘可以加强对数据的理解,空间关系与非空间数据间关系的发现,空间知识库的构造、空间数据库的重组和空间数据查询的优化。

1. 空间数据挖掘

地理信息数据库是空间数据库的特定应用。加拿大 Simon Fraser 大学的计算机系基于关系数据库挖掘系统 DBMiner, 开发的空间数据挖掘系统 GeoMiner (参见图 12.6), 能在地理空间数据库中挖掘特征规则、比较规则、分类规则和数据聚类等。该系统拥有空间数据库模型、空间数据立方体、空间 OLAP 等模块 (参见图 12.7), 并且设计了专门用于空间数据挖掘的语言 GMQL。GeoMiner 的设计思想和体系结构, 也成为开发空间数据挖掘系统的参考依据。

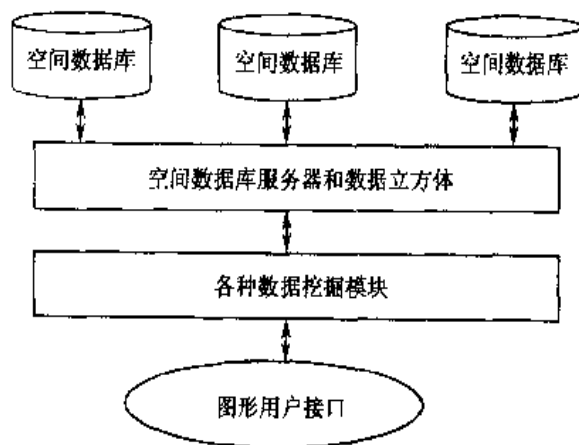


图 12.7 GeoMiner 系统结构

作为基于关系数据库挖掘系统 DBMiner 基础之上的空间数据挖掘系统 GeoMiner, 将

各种非空间数据挖掘任务直接由 DBMiner 完成, 而 GeoMiner 只完成空间数据的挖掘以及空间数据与非空间数据之间的关联挖掘。由于空间数据库中的数据和对象包含初级抽象层次上的详细信息, 而用户又常希望对这些数据进行概括, 且在某个较高层次上抽象表示出来。在空间数据挖掘时, 就可能沿概念树向下挖掘特殊的性质或地区, 分析一些细节数据。因此空间数据挖掘不仅要提供泛化处理, 而且还需要具有灵活描述概念的特化功能。

2. 空间数据挖掘用途

空间数据挖掘主要是对存储了大量与空间有关数据的空间数据库进行数据挖掘, 例如, 地图、预处理后的遥感数据、医学图像数据和 VLSI 芯片设计等数据。空间数据挖掘主要是对空间数据库中非显式的知识、空间关系和其他有意义的模式的提取。由于空间数据库包含大量的拓扑/距离信息, 需要按照复杂的多维空间索引结构组织数据。在访问这些数据时, 需要采用空间推理、地理计算和空间知识的表示技术。这些技术一般比较复杂, 需要效率很高的空间数据挖掘技术来处理。

空间数据挖掘方法目前主要有空间数据分类、空间数据关联分析和空间趋势分析等。

空间数据分类——企业在确定销售地点分布时, 往往需要根据人口的家庭收入、受教育水平等因素对全国、全球进行地区分类, 就需要找出决定地区分类的空间因素, 例如大中学校、高速公路、服务设施等特性。通过对这些特性的分析找出有意义的分类模式, 为企业寻找新的销售点提供合适的模式。

空间趋势的分析——在分析一个地区经济发展与周边空间环境的变化趋势时, 需要用到空间数据挖掘。例如, 在进行地区经济发展规划时, 希望寻找合适的模式: 地区经济发展的不同模式与大中城市的联系、与交通的关联等。这种空间趋势分析, 一般需要在空间数据结构和空间数据访问的基础上, 使用回归或相关分析方法进行。

12.6 分布式数据挖掘

12.6.1 概述

分布式数据挖掘是一种应用分布式算法, 从分布式数据库中挖掘知识的过程, 是一种用途广泛的数据挖掘技术。

许多大型国际企业往往希望了解企业所生产的某种产品与不同国家经济发展的关系, 这就需要抽取位于不同国家的产品销售数据库和经济环境数据库。如果使用集中式数据挖掘工具, 必然要在某个地点将分布在不同国家的这两个数据库组合在一起, 显然这是不可能的。在这种情况下, 就需要利用分布式数据挖掘技术。

分布式数据挖掘技术通常用于拥有分布式数据资源，或将集中式数据库按照水平方式或垂直式划分后，分布在不同的站点上。在水平划分情况下，各站点上的数据是同质（同构）的，即各个站点上数据具有相同的属性集。在垂直划分的情况下，各个站点上的数据是异质（异构）的，即各个站点上的数据有不同的属性集。现实中的分布数据库大多是垂直划分的。

典型的分布式数据挖掘算法涉及两个步骤：首先完成各个站点的局部数据分析，构建局部数据模型；最后，组合不同数据站点上的局部数据模型，获得全局数据模型。

对于水平划分的分布式数据挖掘，由于各个站点的数据模式的同构，挖掘算法简单，只要将通常的集中式数据挖掘方法稍微改造，然后按照上述方法，就能挖掘出合适的全局数据模型。对于垂直划分的数据库，就不能利用集中式的数据挖掘方法，构造合适的局部数据模型，而需要采用汇集型数据挖掘方法。

12.6.2 适合水平式数据划分的分布式挖掘方法

Kargupta 在 1996 年提出了使用数据挖掘代理的分布式数据挖掘代理系统（PADMA）（参见图 12.8）。该系统利用数据挖掘代理、协调器和用户接口，实现分布式数据挖掘工作。

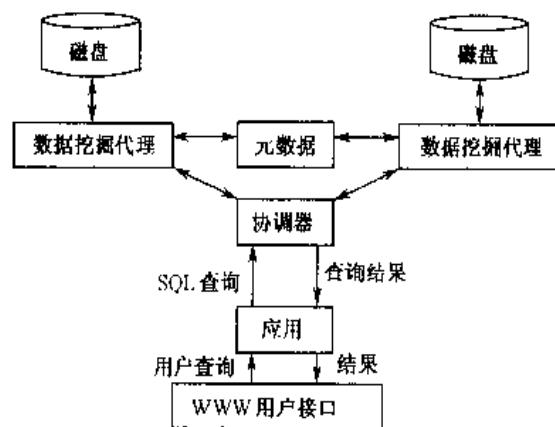


图 12.8 PADMA 体系结构

PADMA 中的数据挖掘代理独立维护各个数据站点的磁盘系统，进行局部的数据 I/O 操作，为整个系统提供并行的数据访问功能。对于分布式文档，各个数据挖掘代理使用简化的模块化数据分布算法。为了访问关系数据库，在 PADMA 中将各个文本文档组成一个文本文件，作为关系表存放在系统中。PADMA 的数据分析主要由各代理以分布式并行方式完成，并将聚类层次结构图、决策树或类似相关矩阵的统计分析结果等“概念图”返回协调器。

协调器接受用户以标准 SQL 表示的查询,且以广播方式通知各代理。而各代理在完成数据分析后,将各自的“概念图”返回协调器后,协调器将其汇集起来,再提供给用户。

PADMA 的用户接口是基于 Web 的图形接口,可以满足用户的远程数据分布式挖掘的需要。

12.6.3 适合垂直式数据划分的分布式挖掘方法

Kargupta 在 1998 年提出使用正交基函数进行局部分析的“汇集型数据挖掘框架”(CDM),解决了在垂直型数据划分中,采用局部数据分析方法不能正确生成构造全局数据模型所需要的局部模型问题。

CDM 的基本思想是将待学习的函数用一组合适的函数,按照分布式方式表示,允许各个数据站点选择不同的学习算法。CDM 能够生成整个数据集的全局分布模型,不必依照各个数据站点的特征空间的特殊划分方法将整个模型的创建分解。CDM 为各个数据站点提供由局部观察变量定义并且用于局部分析和计算基函数的程序,通过各个数据站点对学习算法、通信方式和处理方法的选择,给每个程序分配一个自治度。这个程序就是数据挖掘代理,数据挖掘代理也通过协调器相互协作。

CDM 的数据挖掘主要步骤为:在每个数据站点上产生近似正交基函数及其系数:将选择好的数据样本从各个站点传送到某个站点,生成与非线性交叉项相对应的近似基函数系数:组合局部模型,将模型转化成用户所希望的表示方式并且输出模型。

在 CDM 的体系结构中(见图 12.9),所有的数据挖掘代理在学习阶段,根据各自的局部数据进行学习,一旦代理识别局部基函数及其系数,就将每个代理的不正确预测与对应的索引,以及预测某个类别的强度或可信度,发送到协调器。针对数据库中同一类别标识的输出总数,计算正确预测的百分比。然后,由协调器标识所有代理不正确预测输出的公共数据集,且从所有代理那里得到这个数据子集。协调器利用这个子集运行它的学习算法,以确定不同数据站点上的特征变量定义的基函数。

在测试阶段中,每个代理独立分析和预测,将预测结果及有关的可信度发送到协调器,由协调器根据可信度对预测结果排序,并且根据用户定义的可信度阈值确定各个代理预测结果的可靠性。如果所有代理预测结果都不可信,协调器就自己承担学习任务,从各个代理那里获得相应的考察特征值。协调器利用这些值运行自己的模型,作出最终的预测。否则,如果某个代理的预测结果有较高的可信度,就将其作为最终预测结果。

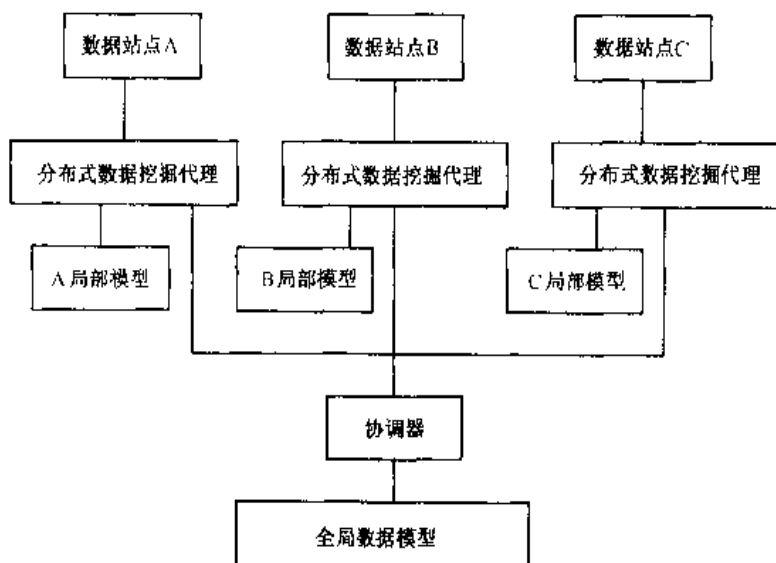


图 12.9 CDM 的体系结构



本章小结

文本数据挖掘技术主要用于解决书籍、研究论文、新闻文章、Web 页面和电子邮件等非结构化数据源中的数据挖掘问题。文本挖掘是基于信息检索系统发展而来的，基于关键字、相似性的检索技术和文档索引技术，在文本挖掘中得到应用。文本分析和语义网络是实现文本挖掘的关键。利用这些文本挖掘技术可以实现文本的总结、关联分析、分类分析和聚类分析等。

Web 挖掘技术可从 Web 文档和 Web 活动中抽取人们感兴趣的、潜在的有用模式和隐藏的信息。在 Web 挖掘中主要有 Web 内容挖掘、Web 结构挖掘和 Web 使用记录挖掘。通过这些挖掘技术，可为电子商务、网站建设等因特网应用提供有益的知识。

分类分析是数据挖掘中一种常用的数据挖掘应用，可以用于分类分析的数据挖掘技术有决策树归纳、贝叶斯分类和贝叶斯网络、神经网络、最近邻分类、遗传算法、基于关联的分类和模糊逻辑技术等。

数据可视化技术包含数据可视化、数据挖掘结果可视化和数据挖掘过程可视化等技术。利用可视化技术可以很清楚地发现隐含的有用知识，这是一个从大量数据中发现知识的有效途径。

空间数据挖掘是对空间数据库中非显式存在的知识、空间关系或其他有意义模式的提取。利用空间数据挖掘可以加强对空间关系与非空间数据间关系的发现，空间知识库

的构造, 空间数据库的重组和空间数据查询的优化。

分布式数据挖掘是应用分布式算法, 从分布式数据库中挖掘知识的过程。在分布式数据挖掘中, 主要有适合水平式数据划分的分布式挖掘方法和适合垂直式数据划分的分布式数据挖掘方法。



习题

12-1 确良电子邮件数据库中包含了大量的电子邮件信息, 可以将其看成主要包含文本数据的半结构化数据库。如何将其转换成结构化的数据库, 以便支持多维检索? 例如按照发送者、接收者、主题和时间等检索。从这个 E-mail 数据库中挖掘什么信息?

12-2 每个科学的学科都有其自身的主题索引分类标准, 用于对学科的文档进行分类。试设计一个 Web 文档分类方法, 可用学科的主题索引标准对 Web 文档进行自动分类。如何利用 Web 链接信息改进分类的质量? 如何利用 Web 使用信息来改进分类的质量?

12-3 由于因特网的动态性和海量存储数据, 开发一个基于因特网的数据仓库是很困难的。但是某因特网信息服务公司希望开发一个基于因特网的数据仓库, 以帮助旅游者选择当地旅馆和餐厅。请设计一个能够提供汇总的、局部的和多维信息的数据仓库, 以帮助旅游者选择旅馆和餐厅。假设每个旅馆和餐厅都有一个自己的 Web 页面, 讨论如何使基于 Web 的旅游数据仓库大众化? 如何查询这些页面? 用什么方法从这些页面中抽取信息? 能否实现一种数据挖掘方法, 提供关联信息, 例如, “去市郊旅游景点的旅游者有 80% 会在中心餐厅就餐一次”。

12-4 可视化挖掘有哪些挖掘类型? 它们在数据挖掘中发挥什么作用?

12-5 分布式数据挖掘在现实管理决策支持中有什么作用?

第 13 章

数据仓库的应用与管理

引 言

在数据仓库建成以后，管理人员开始跃跃欲试地准备将数据仓库用于自己的决策分析中。因为他们都了解美国国际数据公司（IDC）在1996年进行了有名的“数据仓库的冲击”研究，这项研究以62家开发应用了数据仓库的公司为基础，得到数据仓库开发应用的3年ROI为401%，有25%以上公司的ROI大于600%，平均投资回收期为2.3年，平均年盈利220万美元等令人兴奋不已的结果。

管理者都希望在自己的应用领域中领略利用数据仓库辅助决策分析的喜悦。一旦他们在管理决策分析中成功地使用数据仓库以后，往往感到由衷的兴奋。但是，在管理决策人员成功使用数据仓库的同时，一些客户却控告企业侵犯了客户的隐私权。

在数据仓库的应用调查中，显示数据仓库应用所特有的高投入（一个数据仓库的开发，动辄需要数十万甚至数百万元）、高风险（有相当大的一部分数据仓库开发不能取得圆满成功）是数据仓库开发者必须慎重对待的问题。看来，数据仓库的开发、应用，并不是一个简单的技术问题，而更多地还是一个管理问题和社会问题。

通过本章学习，可以了解：

- ◆数据仓库在信息管理中的实际应用
- ◆数据仓库与数据挖掘技术在学习中所涉及到的法律问题与处理
- ◆数据仓库在实际开发与学习中的成本处理
- ◆数据仓库在实际学习中的投资回报问题
- ◆数据仓库在开发与学习中的其他各种管理问题

13.1 数据仓库在信息管理中的实际应用

建立在较全面和完善的信息应用基础之上的数据仓库，主要用于支持高层决策分析。随着信息技术的发展，对信息处理的要求也越来越复杂。管理者希望通过使用数据进行各种各样的分析，以发现有价值的信息，用于辅助决策。但是，管理决策所遇到的问题是不同的，数据仓库的应用也各有特色，数据仓库的开发应用应该根据具体的实际情况采取正确的对策。

13.1.1 分层决策体系

某家用电器制造 / 销售跨国公司在世界许多地区拥有子公司，各子公司拥有相当大的自主权。它们的商务行为各不相同，执行相对独立的制造和销售策略。为了加强管理，执行公司的全球战略规划，公司总部需要集成世界各地子公司的经营信息。由于这上百家的子公司不但数量多，而且经营模式也很不相同，都要符合当地的经济生活习惯。这样，造成各子公司的业务处理系统各式各样，不但硬件环境完全异质（如有的子公司的系统建在主机系统上，有的则建在小型机上，而有的建在分布式系统上），而且事务的处理过程、步骤，生成的报表内容、格式也完全不同。

在公司总部希望加强统一领导的同时，各子公司也希望能对各自的经营状况进行分析，以便根据市场变化灵活地采取相应的策略。于是在决定建造数据仓库取得一致、集成的数据时，面临着两种方案的抉择：一种是先建成公司总部的数据仓库，然后各子公司从该数据仓库中抽取自己关心的部分建立局部的数据仓库；另一种则是各子公司先建立各自局部的数据仓库，形成一个数据集市（Data Mart），公司总部再在这个数据集市的基础上逐步建造全局的数据仓库。前者的困难在于把各局部的杂乱的事务处理一级的数据集成到全局数据仓库是极困难的，所需的投资多，建设周期长，风险大，万一失败损失是巨大的；优点则是一旦建成全局的数据仓库，各子公司的局部数据仓库便可按它的模式建立，数据从全局数据仓库到局部数据仓库的抽取也非常容易、方便。后者的问题在于各子公司都要先建设自己的数据仓库，这种投入的总和会大于前者一次建成全局数据仓库的成本；各局部数据仓库的数据模式很可能复杂多样，而且之间的数据格式的差距也会很大；这样数据从各局部数据仓库到全局数据仓库还需要再进行一次格式转换，不过困难较前者已小得多，而优点却是显著的：先建立局部的数据仓库所需的投资分散，由各子公司承担，建设周期短，风险小，能够较快取得经济效益。对公司总部来说，在建立起各局部数据仓库，各子公司取得局部的经济效益之后，再逐步建设全局的数据仓库则是很便利的。

在考虑两种方案各自的利弊之后，公司总部决定采用第二种方案，即数据集市方案。也就是说，各子公司先建立局部的数据仓库，形成数据集市，然后公司总部再在此基础上建设全局的数据仓库。它的数据来源是各局部数据仓库中经过格式转换的轻度综合数据，而细节数据仍保留在子公司级的局部数据仓库中。特别需要注意的是，各子公司在建造局部数据仓库时应尽量协调各局部数据仓库之间的数据模式和硬件环境（比如一家子公司先建起局部数据仓库，其他子公司根据自身情况采取模仿、改造的方法建立各自的局部数据仓库），公司总部根据将来建造全局数据仓库的要求来加以指导，特别是要尽量保证各局部数据仓库在数据模式和数据格式方面的统一，以便数据易于向全局数据仓库转换。

这个数据仓库体系结构，既方便各子公司根据自身的经营状况制定局部市场策略，又方便了公司总部制定全球战略规划，加强对各子公司的统一领导。例如，在各子公司的局部数据仓库中存放着公司的电器销售信息，各子公司可对这些细节数据进行分析、综合，提炼出有用信息供决策之用。表 13-1 是设在英国子公司的销售记录。

表 13-1 英国子公司的销售记录

日期	产品号	产品名称	订货单位	购买数量(台)	单价(英镑)
.....
2002.1.12	0021-11	2952R 型彩电	威尔士电器行	95	485
2002.1.30	0021-11	2952R 型彩电	伦敦约翰电器行	120	480
2002.2.05	0021-12	2550G 型彩电	威尔士电器行	230	410
2002.3.21	0112-02	1396 型复印机	曼彻斯特办公用品公司	400	960
2002.3.22	0420-16	243M 型电话机	伦敦琼斯电器公司	320	90
....

这家设在英国的子公司在其局部数据仓库的销售记录的基础上进行综合、统计，充分分析公司近期的销售业绩，并且结合客户记录、市场调查结果和竞争对手的销售状况等公司的内外部信息，迅速调整了市场营销策略：

- 根据伦敦市民普遍喜欢读报和近期伦敦市电话装机量有所上升的情况，在伦敦的主要报纸上刊登电话机的广告；
- 根据威尔士的高速激光打印机市场处于上升阶段，竞争开始激烈的情况，对威尔士进行激光打印机的电视广告“轰炸”，以便在当地树立起品牌形象；
- 根据曼彻斯特的录像机市场已趋饱和的情况，在该市停止录像机的电视广告等。

同时存在各子公司的局部数据仓库中的细节数据和某些综合数据，需要定期进行集成、综合和格式转换，生成“过渡”的数据仓库记录文件。在每次生成这种全局数据仓库格式的记录文件之后，便可将其送入全局数据仓库。表 13-2 就是全局数据仓库中的彩电销售汇总记录。

表 13-2 全局数据仓库中的彩电销售汇总记录

年份	地区	彩电型号	数量(台)	总金额	销售国家
2002-1	0021-11	2952R 型彩电	447	248 000 美元	英国
2002-1	0021-10	2950R 型彩电	2011	1 230 000 美元	中国
2002-1	0021-12	2550G 型彩电	210	102 000 美元	法国
2002-2	0021-14	3249H 型彩电	340	219 000 美元	泰国
.....

这样,公司总部就能很容易得到各地子公司汇总的信息。例如,公司总部可在各地子公司销售信息的基础上,结合全球市场竞争状况、各地区经济发展程度和市场调查结果等公司内外部信息,对彩电市场进行分析。根据分析结果,公司总部认为普通大屏幕彩电的生命周期在欧美发达国家已过成熟期,高清晰度彩电市场开始萌芽;而在亚洲由于一些发展中国家经济发展迅速,居民购买力上升很快,特别是像中国这样的国家,普通大屏幕彩电的生命周期还处于成长期,市场潜力极大。在很快分析出大屏幕彩电市场需求转移的情况之后,该公司抢在竞争对手之前,从欧美的子公司抽取资金,加大对亚洲子公司的投入,扩大其生产规模并采取适当营销策略,迅速抢占市场。

该公司的数据仓库结构是一个分层鲜明的决策体系,其独特性有如下 3 个方面。

1. 一个全面的综合性的数据仓库

该数据仓库体系不仅有全局的数据仓库,而且有局部的数据仓库;不但公司总部可以根据全局数据仓库进行分析、决策,而且各子公司也可参照本地的局部数据仓库采取灵活的市场策略。

2. 开发方法的独特性

在建造这个数据仓库体系时,采用自下而上和自上而下相结合的方式。所谓“自下而上”是指各子公司先建立自己的局部数据仓库,在形成数据集市之后再在其上建造全局的数据仓库。所谓“自上而下”是指在各子公司建立自己的局部数据仓库时,公司总部必须在数据模式上给予全局性的指导,以便在将来建设全局数据仓库时,局部数据仓库的数据能够顺利向全局数据仓库过渡。建造这个数据仓库体系采用自下而上和自上而下相结合的方式是比较合适的。因为各子公司大体上位于同一业务级别,它们之间的相似之处很多,使得各局部数据仓库的数据模式实际上是基本一致的。便于公司总部在各子公司自下而上地建造局部数据仓库时,自上而下地进行指导,尽量保证各局部数据仓库之间以及它们与将来的全局数据仓库之间的一致。

3. 数据按需要进行综合

由于公司总部在进行分析决策时，一般并不需要细节数据，所以通常细节级数据存在各子公司的局部数据仓库中，实际上被送入全局数据仓库的数据已经过转换，且是轻度综合级以上的数据。

13.1.2 数据抽样分析

某家化学公司想在不增加投资、不购进设备的条件下，采用挖掘现有生产潜力的办法来增加产量，提高效益。他们用产出率来评价每次化学产品生产的效率，目的是想通过提高产出率来提高每炉化学产品的产量。产出率是每炉中所有优质产品数量占总数量的比率。若一炉中的所有产品均为优质，则该炉的产出率为 100%。

每炉产品有 50 000 个变量，如化学反应时的温度和一些化学催化剂、化学配料等参数。希望分析哪些因素与高产出率有关？哪些因素与低产出率有关？这是一个统计方面的问题。但所要处理的数据量太大，每炉化学反应有 50 000 个变量，又要考虑多座化学反应炉。于是，需要建立数据仓库来容纳大量的数据。如果每次都要收集所有数据，那么其工作量是极大的，所花时间要数以天计。实际上这 50 000 个变量并不都是要分析的，有的对产品质量影响很大，而有的却不显著。根据这种情况，公司采用了较特殊的数据仓库建设方法，化学公司数据仓库的存储层次结构如图 13.1 所示。最高层按照专家意见，选出最重要的 200 个变量，存放在高速存储设备上。通常，分析人员使用该层变量分析比较化学反应的情况。这样，一般的分析工作可在几分钟内完成。下一层是 4 000 个变量，存放在中速存储设备上。最低一级存储了所有 50 000 个变量的数据，因为很少存取，便可存放在低速的大容量存储器上。分析总是从高层开始，若有必要才逐层向下进行。按变量的重要程度分别选择不同的存储介质和存取方式，既能保证日常分析的高效率，又大大降低了硬件成本。

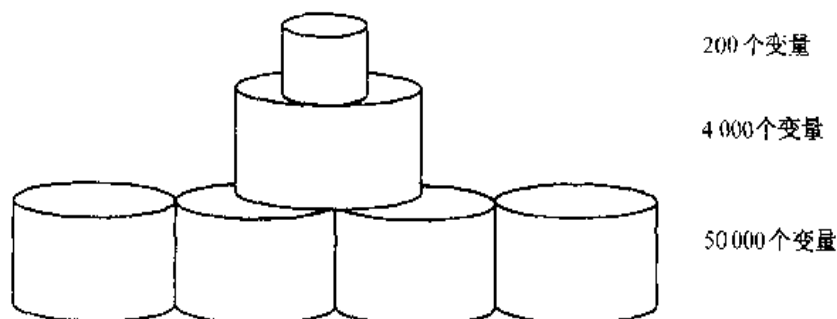


图 13.1 化学公司数据仓库的存储层次结构

由于该数据仓库的应用环境的限制，数据仓库表现出如下两个特点。

1. 一个专门的小型数据仓库

该数据仓库用于解决专门问题，没有对数据进行粒度划分，整个数据仓库只有一个粒度——细节级数据。其细节数据不能被综合或聚集，数据按重要程度进行分级存储，而没有像一般数据仓库那样对数据进行汇总和统计，并且按照综合程度进行数据的粒度划分。

2. 数据按实际需要进行分层

在许多情况下，随着时间的延续，对细节数据的需求也就减少了。但在这个应用中，半年前一次化学反应的细节数据与昨天一次化学反应的细节数据同样重要。因此，该数据仓库并没有将细节数据再分成当前细节数据和早期细节数据，没有把早期细节数据转储到海量存储设备上，而是按照数据的重要程度：或者说，按照分析时对数据的读取频率对数据进行分层。

13.1.3 发挥历史数据的经济效益

远程销售商为了开展远程函购销售，定期向外散发商品目录单。消费者接到目录单后，若对某种商品感兴趣，可打电话询问更多的信息。为此，公司建立了一个数据仓库。数据仓库中包含许多对该零售商有用的集成历史信息，如客户信息、供应商信息、产品信息和销售信息等。例如，客户信息包括的内容如表 13-3 所示。

表 13-3 客户信息

时 间	客 户 号	客 户 名	购 买 商 品	购 买 数 量	购 买 商 品
.....
2001.12.23	001213	Mark	黑色西服	1 件	笔记本电脑
2001.12.23	003902	Inmon	蓝色高尔夫 T 恤衫	2 件	高尔夫球
2001.12.25	012902	Smith	CANON 相机	1 部	SONY 摄像机
.....

该数据仓库的数据来自该公司多年的业务活动的记录。分析工具可以根据该数据仓库进行多方面的分析，提取有用的信息，且把这些信息存回业务处理系统的某个表中。例如，其中的一种分析报表（见表 13-4）从数据仓库中提取了有关客户的几条信息，包括：

- 零售商最后一次与该客户接触是什么时候？
- 客户上次购买的物品是什么？
- 该客户喜欢哪类商品？

表 13-4 经分析得出的客户报表

客 户 号	客 户 名	上次购货时间	上次所购买商品	喜欢的商品	推 荐 商 品
.....
001213	Mark	1996.12.23	黑色西服	电脑	AST 电脑
003902	Innon	1996.12.23	蓝色高尔夫 T 恤衫	高尔夫球用具	高尔夫球
012902	Smith	1996.12.25	CANON 相机	摄影器材	SONY 摄像机
.....

根据上面的一条信息可以推测出该客户的一些背景信息。根据这些分析, 销售商便把相应的商品目录附在上述的分析报表之后。

这样当一个客户打来电话时, 远程销售商便可迅速调出该客户的报表。根据报表就可有针对性地说: “先生, 非常高兴再次接到你的电话。我想我们自去年 12 月以来还未收到过你的消息。” 然后又可以接着问, “你去年买的那件蓝色高尔夫 T 恤衫如何? 够暖和吗? 运动后有没有撑大?” 这样, 前两条信息所产生的效果会使消费者觉得自己是唯一受重视的客户。之后, 远程销售商又可以向客户推销他喜欢的那类产品, 如高尔夫球。而在没有数据仓库之前, 要取得这种信息简直不可想像。由此可见, 简单的信息可以大大增强竞争力并且带来巨大的经济效益, 而数据仓库的投入较由此取得的长远的经济效益相比是很微小的。

这个数据仓库的特点有如下两点。

1. 无专门数据仓库体系

该数据仓库应用方式比较特殊, 即从数据仓库中提取的信息又重新存回原有的业务处理系统当中, 有关客户的分析结果从数据仓库中提取出来, 存入一个相对独立的表中, 公司的业务处理系统便可以对此表进行快速查询。

2. 建于业务处理系统的数据仓库

这是一个在业务处理环境中使用历史信息例子, 其投资额较低。这说明对存储的大量历史数据只要充分利用, 善于分析, 提炼出有用信息。就能用“活”数据仓库, 提高经济效益。

13.1.4 回扣分析

航空公司与各旅行社约定, 旅行社每订出一张机票将获得一定的回扣。航空公司发现: 如果所给回扣太低, 旅行社就转向别的航空公司进行交易; 而所给的回扣太高, 又会降低航空公司的利润。如何计算最优回扣, 需要两条信息——该航班的当前订票情况

和历史订票情况。如果此航班有较多空座位,则可以给较高的回扣,以吸引旅行社代订机票。如果航班的空座位较少,则可以给较低的回扣。若要查看的两条信息,第一条在当前业务处理系统中,即一般的航班订票系统,通过快速存取、统计容易取得;第二条在数据仓库中。在数据仓库里的历史记录数以百万计,如果每次分析历史订票情况都要进行存取和统计,响应速度之慢可想而知。折衷方案是周期地扫描数据仓库,然后进行统计、汇总,把结果存入数据仓库以备分析之用。

当航空公司想要制定某个合理回扣时,就可方便而快捷地调出当前汇总数据和历史汇总数据进行比较和计算,以使公司在保证航班满载的条件下尽量降低成本。

此例说明历史信息的重要性,因为航班当前订票情况的好与坏是相对于历史订票情况而言的。一般在没有建立数据仓库的飞机订票系统中,历史的订票情况都被不断刷新而丢失了,只存有当前的或近前的订票信息,从而无法进行客观的比较。由于数据仓库的数据量较大,每次从大量的细节级数据开始分析是不可想象的,所以必须对细节级数据进行综合,以便为分析工作提供方便。

13.1.5 客户关系管理

在新型的客户关系管理策略中,数据仓库中存储的客户各类数据,能够提供客户的详细信息,从而增进对客户管理的关切程度,改进高层次销售方式。增加对客户实际情况的了解以及客户所有要求的了解,可以用来指导市场部门与客户保持恰当的联系。数据仓库在客户关系管理中可以发挥以下5个方面的作用。

1. 维护客户基础

在开放的市场上,公司所面临的最大的挑战之一是客户的流失。在市场竞争中客户的流失现象是必然发生的,但是必须了解哪些客户的流失是不必介意的,哪些是必须尽力避免的。应该不断进行客户的细分工作,发现谁是最好的客户,谁是最好的潜在客户,这是客户关系管理中最重要也是最艰难的事。基于数据仓库并结合数据挖掘的解决方案可以知道正在失去哪些客户,哪些客户有可能流失。通过数据仓库上的深层次数据挖掘,发现和确定离开的客户。从统计的角度,能够发现竞争对于在哪些地区很活跃,在哪些层次上很敏感。这些信息可以帮助制定非常成功的客户忠诚计划,针对客户的具体情况提供新的服务协议来赢回客户。

2. 管理的收益

在数据仓库系统实施以后,决策者可以简便地了解大量的信息,以回答关于客户和服务方面的特别问题,同时发现隐藏在数据中的潜在趋势,以求更接近一对一销售的市

场营销理念。

利用数据仓库可以统一掌握客户信息,数据仓库集中管理客户的所有数据——包括旧有信息和因特网信息,能够提供一个统一的客户信息管理系统。同时还能保证信息在企业内部是统一的、一致的,这样就可迅速准确地预期客户需求,提高盈利能力。在数据仓库中还包含大量的经营信息,数据仓库中的信息可以通过客户和市场调查来提供。利用数据仓库帮助分析、采集和记录用户信息,目的是为了定向销售和服务,为客户提供符合他们需求和行为的正确信息,帮助企业进行有效的客户鉴别和定位自己的目标客户。统一的数据仓库可将客户服务系统和 CRM 进行有机集成,为客户提供多层次、个性化和多样化的服务:针对客户的消费心理、价值取向与消费行为,实现对客户关系与资源的挖掘、分析与管理,以保持现有的客户、发现潜在客户,实现营销的个性化服务与企业利益的最大化,且对积累的客户信息,进行深度分析挖掘,产生客户分类模式及行为模式,企业的经营管理决策提供业务预测及相关业务评价。

3. 利润的增长分析

企业的销售利润与企业所提供的产品和服务有关。在产品销售与服务收益没有明显的联系时,可以通过历史趋势分析来了解这种关系。

4. 企业的战略管理

现有的客户可能带来取得新收益的机会,利用交叉销售或提升销售可使企业获得销售的增长。通常,企业的业务处理数据是一种特定的信息源,一般仅适用于本企业。如果将业务处理数据与统计数据结合,可以产生一个特定的信息库,利用它可以更好地了解客户,例如客户的购买方式、产品包装、服务需求等方面。具有这些源于数据仓库的经验后,可以制定带来大量利润或吸引顾客的市场营销策略。

5. 改变竞争的基础

从数据仓库的历史数据中收集关于客户的知识,且能通过对实际运行结果的快速反馈得到增强。这些知识可能造就一种可行的、更快和更新的业务运营方式,以便更精确、更全面地满足顾客的需要。例如,市场定位系统如果有更适当的产品、合适的特征,且在恰当时间推出,将使客户更加满意。

13.2 数据仓库应用与数据挖掘中的法律问题

在数据仓库与数据挖掘中所涉及的法律问题,主要指客户的隐私权保护与处理问题。由于数据仓库与数据挖掘所具有的强大数据处理功能,可将原本分散在各系统中、隐藏

在数据背后的客户信息集中在一起，很清晰地表现出来。此时，数据仓库的拥有者就遇到了客户隐私扩散的危险。这种客户隐私的扩散将带来客户的不满，使原本希望利用数据仓库争取市场的企业，意外地遭遇到客户的声讨，甚至被客户告上法庭，带来与开发数据仓库初衷相违的后果。因此，客户的隐私保护问题就成为数据仓库开发、应用中的一个重要法律问题。

13.2.1 数据的隐私权问题

客户的隐私问题是全世界所有企业都关注的问题。虽然这在数据仓库管理中只是一个背景问题，但已经引起各方面的关注。因此，在建立和管理数据仓库的过程中，对于收集、运用、分发和管理客户信息以及对信息的选择，都需要建立一系列明确的政策、措施和指导方针。客户的隐私在数据仓库中的应用，意味着对大量客户信息使用的控制和保护。当大量的客户信息存储在数据仓库中时，数据仓库的拥有者就有义务保护这些客户数据免遭滥用。这不仅是保护客户隐私权的义务，更重要的是不要使企业陷入因保护客户隐私不当而带来的诉讼泥潭中。

1980年由OECD（经济合作与开发组织）颁布的《隐私保护管理指导方针》，为个人数据保护确立了基本准则。这项指导方针鼓励成员国颁布关于用户数据标准方面的法律。这些法律能够赋予个人一些权利，监督数据收集者合法使用个人数据。这些权利主要从个人和信息收集者的角度，对个人隐私数据的收集与处理做一些规定。

个人有权知道所收集数据的使用目的。在收集个人数据时，数据收集者应该将数据的用途和处理过程以浅显易懂的方式告诉消费者。个人数据必须符合实际情况，能够准确、完整地反映个人实际情况。个人应该可以访问自己的数据，可以就有关自己数据的真实性提出质疑。如果质疑是对的，可以进行删除或修改关于自己的数据。应该能够知道关于个人数据的发展、使用状况和有关政策等信息。应该提供方法来确定个人数据的存在和属性，主要的用途，以及数据收集者的身份证明和实际住址。

数据收集者在收集数据时必须有一个数据的收集限度，任何个人数据都要通过合法、公平的手段获得，并且在数据的使用上也有限制。在获得客户准许或经法律批准以前，个人数据不能以任何形式披露、出售、使用，或者用于不同于收集阶段所阐明的用途。要对个人数据采取必要的安全措施，以防丢失、未授权访问、破坏、使用、修改或泄露。数据收集者还须对个人数据使用中所产生的后果负责。

13.2.2 数据隐私权的处理

一般来说，在收集和使用客户数据之前，公司应该向客户通知他们所收集的数据，

解释为什么要收集这些数据。他们也应该给予客户选退（拒绝）收集和使用他们数据的权利，除非法律有明确的规定，或者已和个人签订了合同或保护了个人的权利。数据收集方还应该有控制收集的机制，有对数据使用以及对数据访问或其他类似操作的管理方法。

从数据仓库角度来讲，隐私既会带来成功又会带来威胁。滥用隐私不仅破坏公司在客户心目中树立的良好形象，也会将数据仓库和数据挖掘推入灰暗的前景中。公众会对这种有目的性的滥用隐私行为感到厌烦，将阻碍数据新技术的采纳及应用。另一方面，采纳了数据隐私政策的公司将会在客户中建立起信任关系和良好的公众形象，并在遵守法律要求方面也提供了信心。而且，一个运行良好的隐私程序将帮助公司收集到关于客户的更准确、更详细的数据（但须经过客户的合作和同意）。为此，需要在数据仓库中进行有效的数据隐私处理。数据隐私处理主要从以下两个方面实现。

1. 数据隐私的处理

为解决数据仓库的隐私问题，要让客户了解客户数据收集的目的，客户数据应用的权利，客户数据的认可与客户数据的安全保护。

（1）应该使客户知道如下的信息

应让客户知道所收集或使用的个人信息的存在性及本质，数据收集的政策；任何类型处理的预期目的，例如，数据的收集、应用或披露等；“数据控制员”以及其他接收数据人员的身份；任何自动处理中涉及的逻辑。

（2）收集和使用限制

应将收集和使用限定为明确、具体和合法目的。相对于初始目的来说，数据必须是“适当的、相关的而且不过分的”。数据必须以可识别的形式存在，存在的期限不应长于初始目的所确定的必要期限。

（3）选退和选进

客户应能选退将个人数据用于直接营销，并且可以选退将个人数据披露给第三方。客户也可通过明确选进，表示同意数据的使用目的。

（4）数据质量、访问、精确性和更正

应给客户提供一种能力，使他们能对不精确或不完全的个人数据加以检查和纠正，客户有权删除或阻截那些与地方法律规定不相符合的个人数据收集。

（5）数据安全

确保个人数据不丢失，以及不发生未经授权的访问、破坏、改动、使用或泄露数据。

（6）义务、强制和求助

支持现存的法律和补充规则的强制执行，支持国家隐私管理部门所认定的应该达到的隐私控制要求。

2. 数据隐私控制框架

为了达到上述隐私处理要求，需要建立一个隐私控制框架。

(1) 增强逻辑数据模型

为了强调隐私，应该首先检查公司已发展的逻辑数据模型，且将所有与“客户”相关的数据进行实体确认，包括显示身份、提供个人信息。首先应该检查客户轮廓（也就是与“客户”有关的当前数据实体），因为这样可以确定是否应该增加一些有价值的数据实体，以便获得更深入的关于客户喜好的信息。

选退的表或列应该反映到逻辑数据模型中，以支持客户对有些个人信息使用的选退。至少应该有四个“选退”的存在：“直接营销”、“向第三方披露”、“自动决策”和“敏感数据的使用”。

(2) 用隐私视图支持限制性访问、选退和匿名

建立增强的逻辑数据模型之后，就可识别个人数据字段、敏感数据字段、显示身份字段和适当细节的选退结构。这时应对整个数据仓库应用进行一次复查，包括那些在模糊查询和其他为各种级别数据用户而建立的数据查询形式。建立视图的目的主要有限制访问个人数据，允许完全访问个人数据，使个人数据匿名化，为了某项特定目的客户进行选退使用他们的数据，选择性地使个人数据匿名化。应该将应用分类应用于下述 5 个类型的视图中。

- 分析应用：“匿名化”视图。
- 采取行动应用：“直接营销选退”视图。
- 披露应用：“选择性匿名化”视图。
- 特别管理应用和用户：“个人数据”视图。
- 所有其他应用：“标准”视图。

(3) 为个人数据管理提供交互式客户服务界面

在建立基于拓展模式的数据仓库，并且加进额外个人数据字段和“选退”标志之后，需要一种方法为这些额外的列加入特殊的客户数据。客户档案信息可以通过客户访问的网站，或呼叫中心界面，或电话和邮件活动来获得。这些活动应该以隐私为中心，还要提供一份关于隐私保护的声明。一个交互式客户界面对于客户访问他们的私人信息是有用的。他们可以以此来检查、更新、更正那些信息，这种交互界面对在客户之间产生良好关系是很重要的，也会带来安全和性能方面的问题。

(4) 提供报告验证是否遵守隐私

隐私问题的另一个方面是需要对遵守情况进行验证。验证可由一个独立组织、政府部门或自我验证来完成。报告中应该包括具体的逻辑数据模型、数据仓库概括、不同的隐私视图以及相关的优先级别和通过视图访问数据仓库的应用类型。

13.3 数据仓库开发与应用的成本/效益分析

数据仓库同所有的信息技术一样，都是通过投资来提高企业的竞争能力和盈利水平。因此，企业需要制定数据仓库计划（商业的和技术的），并且进行成本/效益分析。在数据仓库的开发应用中还需要对可能遇到的风险进行分析，这些分析不仅涉及有形的风险和效益，还涉及无形的风险和效益。

数据仓库的投资可能巨大而且具有相当大的风险。因此，在数据仓库的实际应用中能否获取这些巨大投资的回报，成为数据仓库开发者十分关心的问题。

与各独立的 DSS 维护费用总和相比，一个数据仓库的维护成本相对要低。数据仓库的主要目标是使构造信息库的过程自动化，从而使支持费用相对较低。随着软件、硬件和存储费用的不断下降，大型数据仓库的维护费用可以得到持续的降低。使用数据仓库可以减少对管理决策人员的聘用人数，进一步减少组织的经营管理成本。

数据仓库的效益还表现在对决策的实质支持上。数据仓库通常是支持决策的惟一数据源，能为决策者观察问题提供崭新的视角，帮助他们弄清数据的最原始来源，发现数据的变化规律，并且以此为依据进行决策。因为数据仓库中既有细节数据，又有汇总数据，使企业能够同时进行宏观和微观决策分析管理。数据仓库自动对细节数据进行各种层次抽象的能力，不仅使企业节约了大量的手工劳动，而且避免信息不全所带来的决策错误后果。

为了明确地显示数据仓库建立以后所产生的实质性经济效益，需要具体分析数据仓库的投资回报。具体分析数据仓库投资回报的方法很多，主要包含定量分析与定性分析两个方面。

13.3.1 数据仓库投资回报的定量分析

评估投资机会的方法有很多，主要有投资回报率（ROI, Return On Investment）、回报周期（Payback Period）、净现值（Net Present Value）和内部回报率（Internal Rate of Return）等。

企业判断一个投资机会是否具有吸引力，最常用的方法是计算该项目投资的投资回报率 ROI。在数据仓库建设中，ROI 应用的具体操作步骤有：首先识别数据仓库影响哪些重要的商业过程，然后计算使用该数据仓库造成的费用增长和获得的利润。假设一个企业在第一年投资，在第三年获得回报，计算 ROI 时不能简单地直接比较这两个值。为使投资和收益具有可比性，两者必须用同一时期的价格表示，即转换为同一时期的货币时间价值。ROI 的数学表达式如下：

ROI=收益现值/成本现值

$$\text{收益现值} = \sum_{t=0}^n \frac{B_t}{(1+i)^t}$$

$$\text{成本现值} = \sum_{t=0}^n \frac{C_t}{(1+i)^t}$$

公式中 B_t 为第 t 年的收益、 i 为实际利率（或行业平均投资收益率）， C_t 为第 t 年的投资额（或成本）， n 为计算截止时期。

对数据仓库进行 ROI 分析，能够反映数据仓库的使用是否带来价值。当数据仓库的信息能够有效地用于制定决策时，ROI 结果就会很高。在这些情况下，管理人员往往以数据仓库中的信息作为其赖以制定决策的的惟一依据，如果没有数据仓库的支持，他们就没法进行决策。而且数据仓库给他们带来的好处并不是一次性的，而会长期延续下去。

除了 ROI 分析投资回报外，度量投资回报的另一个常用的方法是投资回收期。投资回收期是指一个企业收回所有投资需要的全部时间。许多重视投资产生效益所需时间的企业往往很关心这一个指标。投资回报期的计算公式如下

$$T = t + \frac{I}{B - C}$$

其中： T 为投资回收期，单位为年；

t 为资金投入至开始产生效益所需要的时间，单位为年；

I 为投资额，单位为万元；

B 为数据仓库运行后，每年新增加的效益，单位为万元/年；

C 为数据仓库每年运行费用，单位为万元/年；

如果要计算更精确些，对投资额、效益和运行费用都要计算现值。

13.3.2 数据仓库投资回报的定性分析

数据仓库无疑会给组织带来许多无形收益，其中最重要的一点是为决策者解决商业过程中存在的问题提供良好的技术手段。数据仓库提供深入了解商业模式的能力，能够准确地指出如何利用商业模式。因此，数据仓库是一种工具，管理人员可以用它获取存在于企业数据内部的信息。数据仓库的投资回报的定性分析可从以下 5 个方面考虑。

1. 为客户提供更好的服务

数据仓库可为企业建立一个关于客户与产品种类、地区和销售渠道之间关系的集成的视图。这种视图有助于一个企业更好地为已有客户服务，使这些客户更加满意，从而增加客户再次购买商品的可能性。数据仓库中的信息是组织中很有价值的财富。组织利

用这些信息可以向客户提供更好的相关服务,以加强企业与客户之间的关系,创造更高的销售收入,延长客户的购买寿命,同时也使客户获得更好的服务。在一些企业中,数据仓库已成为产品生产和销售总体策略中的一个核心部分。

2. 建立企业内部的合作关系

企业中各个部门之间的合作失调,往往是困扰企业的主要问题,它严重地妨碍企业的发展。数据仓库为各个独立的视图与企业的最终目标联合提供基础,各个部门的每个人都可看见自己在企业中的角色,更好地了解自己的任务与同事所承担的任务之间关系,使员工知道应该如何与他人合作,获得工作的成功。

3. 对市场机会快速反应

数据仓库能够及时为决策提供需要的丰富信息,包括当前的细节信息,各个不同时间点的历史信息,以及日、周、月、年的各种汇总信息。将当前信息放在过去的相关环境中,能够反映一个历史的变化过程,可使决策者更清楚地认识当前的事实,正确地把握转瞬即逝的市场机会。

4. 既能够管理宏观数据也能够管理微观数据

如何在维护企业的宏观视图与维护企业内部的细节数据之间进行权衡,一直是管理人员难以解决的问题。数据仓库为管理人员同时高效地管理两极视图提供手段,因为数据仓库本身正是基于大量的细节数据建立的,同时又被集成为多种不同级别的视图。

5. 改善管理能力

当许多管理人员面对大量的企业数据时,无法从大量的数据中发现事物发展的趋势,他们往往仅凭直觉来主观预测事情的发展。数据仓库能够妥善地处理这些大量数据,为管理人员提供监视与测量事件状况的能力,提供对事情发展进行控制的能力,使经营管理者能够得出仅凭直觉难以获得的结论,有效地提高管理能力。

13.4 数据仓库的开发与运行管理

数据仓库的管理应该涉及数据仓库的整个生命周期,即包含从无到有,从小到大的数据仓库开发管理,也包含数据仓库投入运行后的管理和对数据仓库的评价等。这些管理涉及数据仓库开发与应用的人员组织结构与管理,数据仓库开发项目的管理,数据仓库的运行管理和数据仓库评价4个方面。

13.4.1 数据仓库开发与应用的组织结构

数据仓库的开发与运行管理是一个系统工程。建立数据仓库的复杂性和数据仓库本身的丰富性需要大量的专业人员参与。而数据仓库日常运行中的维护与管理也涉及大量的用户以及数据仓库运行和维护人员。

为了做好数据仓库的开发和运行，需要组织大量具有不同技能的成员，形成一个数据仓库项目的开发管理组织（见图 13.2）。一旦开始规划数据仓库，就可根据数据仓库的开发进度，管理整个数据仓库开发过程。项目开发组中包含数据仓库开发组、用户工具开发组、数据仓库管理组和数据仓库用户组。这些小组需要在数据仓库项目主管与数据仓库技术主管的领导与指导下，展开数据仓库的开发应用工作。



图 13.2 数据仓库的项目开发管理组织

为了适应数据仓库的开发与应用，数据仓库项目组织需要由不同的人员组成。这些人员在数据仓库开发应用中都具有自己的职责和技能。这不仅是数据仓库项目开发应用的保障，而且也是组织整体结构有效管理的需要。

1. 数据仓库项目主管

数据仓库项目主管对整个数据仓库的开发应用负责，促进组织中数据仓库的开发与应用。要求其对数据仓库具有基本概念的了解，能够同数据仓库技术主管密切合作，组织领导数据仓库开发应用的所有业务活动。数据仓库的主管应该建立与高级管理层的良好关系，向他们介绍建立数据仓库所带来的益处，增加高级管理层开发应用数据仓库的信心。在高级管理层认可构建数据仓库的价值之后，数据仓库的主管还负责将有限的资金分配到不同的数据仓库主题区域构建上。数据仓库的主管还执行一些其他的典型职能，如进行预算、计划安排以及对数据仓库开发应用中与其他管理部门、业务处理系统的协调。

2. 数据仓库技术主管

数据仓库技术主管负责数据仓库开发应用中的所有技术工作，负责为整个组织设计数据仓库的标准和整体体系结构。负责将各个部门的数据仓库开发要求转变为技术解决

方案,包括数据建模、设计和实施数据仓库结构,并且管理所有的界面。数据仓库技术主管还要负责选定必要的数据库组件,例如硬件、网络、数据库管理软件、中间件、数据提取和转换及清理软件、最终用户工具等。

3. 数据仓库开发组

数据仓库开发组主要负责数据仓库的开发设计,其成员包括数据仓库分析员、数据仓库设计员等。

(1) 数据仓库分析员

数据仓库分析员主要从事各种各样的数据模型和数据仓库元数据模型的分析工作,包括数据仓库、数据集市规范数据模型和特殊数据模型的分析创建。为了做好这项工作,工作人员要在客户结构和业务流程方面具有渊博的知识,能够与用户代表和数据仓库设计员紧密合作。

(2) 数据仓库设计员

数据仓库设计员主要负责根据数据模型设计数据仓库的物理模型,确定物理数据仓库的各种设计指标,实现数据的转换以及将数据整合到数据仓库中。数据仓库设计员的其他职责还包括对数据仓库的性能进行优化,负责标识数据源并监视数据的质量,执行数据仓库数据库管理任务。

4. 用户工具开发组

用户工具开发组主要由应用程序员组成,他们的主要职责是将用户信息请求转换为报表、转换为用户所需要的其他表现形式。测试这些应用程序,以确保其正确性。应用程序员还需要为用户提供分析应用程序。为了做到这一点,他们要分析公司各个部门的用户需求,与数据仓库分析员合作,建立针对具体应用的数据模型。在分析数据来源时需要和数据仓库设计员合作,以确保所设计的用户工具能使用户很好地应用数据仓库。

5. 数据仓库管理组

数据仓库管理组主要由数据仓库系统管理员组成,他们的工作主要是负责完成所有与系统直接相关的任务,包括安装软硬件、建立用户文档和实施安全策略等。每项工作通常都由这个方面的专家来负责完成。数据仓库系统管理员将负责一些典型的系统管理任务,如用户数据资源分配、用户安全管理、用户记账信息收集、系统管理软件管理和软件补丁应用程序的管理等。

6. 数据仓库用户组

数据仓库用户组主要由业务顾问,数据仓库用户管理员,数据仓库用户培训员和最

终用户代表组成。

(1) 业务顾问

业务顾问师是业务专家，他们能够从部门的角度定义数据仓库项目的目标和收益。还要确定并分析数据仓库所涉及的业务领域，根据正确的决策分析流程与数据仓库分析员一起建立概念模型。他们的活动聚焦于分析业务的决策管理活动上。一旦项目得以实施，他们还要持续关注数据仓库对管理决策的影响，对此做出正确的评价。

(2) 数据仓库用户管理员

管理用户系统的用户管理员是最终用户了解数据仓库信息的主要支持。用户管理员需要负责指导用户在大量的报表中寻找对自己管理业务适合的信息。有时，用户管理员还要帮助用户创建或测试已经开发的报表。这就要求用户管理员熟悉用户所使用的业务处理系统以及对应的数据仓库主题域，因为数据仓库要用的数据将从这里获取，且使所需的数据变得可供用户使用。

(3) 数据仓库用户培训员

数据仓库用户培训员负责培训用户使用数据仓库的能力。在数据仓库项目的工作中，培训的概念包括对最终用户就数据仓库有关的内容进行培训。这些内容包括数据库、或者引导用户使用 OLAP 工具以及这些工具所开发的应用报表。

(4) 最终用户代表

至少在每个主题域中有一个最终用户代表，他们的职责在于向数据仓库开发人员反映用户对数据仓库的期望，在管理决策分析中所遇到的信息短缺问题，在数据仓库应用中发现的技术难题。

13.4.2 数据仓库的项目开发管理

1. 数据仓库项目需求管理

当用户意识到数据仓库对管理决策分析的重要性后，就会纷纷提出数据仓库开发项目的请求。这些开发请求可能包括客户关系管理，产品质量控制，信用欺诈，市场开发和供应链的管理等。

面对蜂拥而来的数据仓库开发请求，数据仓库开发者必须做好数据仓库开发项目的管理。从中选择对组织经营战略影响较大的，或是收益比较明显的，或是容易开发的项目，或是收益部门较多的项目作为优先开发项目。

为此，首先要求各个部门的业务分析员，有关领域的专家，最终用户代表和部门经理等管理人员定义他们各自有关业务的关键问题，即他们最需要迅速而准确地知道答案的问题。

再将所有需要解决的问题合并起来，考虑它们之间的共同之处，列出具有公共主题

的数据仓库开发项目,按照数据仓库开发的公共主题,从共享的到独占的,对所有的数据仓库开发项目进行排序。

对每个数据仓库开发项目可能产生的收益和在组织经营战略中的地位,从大到小进行排序;按照所需开发力量、开发资源的大小,从小到大进行排序;对项目开发的难易程度,按照从易到难进行排序。

最后,根据项目的综合评估情况,结合数据仓库的开发力量和开发资源确定数据仓库开发项目的优先次序。

应该注意到用户的数据仓库开发需求是会变化的,对开发项目的评估一定要估计到项目上马后,可能产生的变化趋势与所带来的成本变化。在对项目需求管理中,开发人员与用户应该彼此信任,相互协商,用良好的愿望定义项目的开发和开发风险的管理。

2. 数据仓库的设计管理

在数据仓库的设计过程中,必须充分考虑所涉及的各种业务领域,在设计时从实际出发,在技术能够支持的情况下,对可行的计划折衷权衡。

数据仓库设计管理的主要任务,是在软件组件和数据仓库系统及需要解决的决策问题之间提供一种结合方式,还应提供与需要回答的高层次问题之间的结合方式。利用灵活的体系结构,提供回答这些问题的手段。在数据仓库系统中,设计是以数据为中心的,因此,商业实体及其行为领域的概念是关键所在。在设计过程中,要对数据维、事实的细节粒度级别、组件表达形式的完整性进行识别、定义和精练;同时进行比较,找出候选实体的不一致性。在设计数据仓库时,要用大量的时间来标识、定义和精练维的最小集合,以使它能体现数据的特征。对于每一种维,其连续的和一致的表达可以作为一种约束,强制项目始终指向正确的方向——客户、产品、时间、地点、供应商、服务、提供商、诊断和渠道等。这种应用的核心是用技术将商业规则与商业系统联系起来。为此,在建立实际系统的开始就要识别和定义维,尽可能建立完整的客户、产品维结构。对于维的定义,要求在维的交集中体现粒度级别。客户在特定的时间和地点买了一件商品,这是基本的事务。如果客户一周内3次做了同样的事件,这就是每周时间的一次聚合。对于回答业务问题的细节来说,重要的是获得、跟踪和体现客户的信息。同时,系统的运作细节也有重要的作用。应该规定如何在数据仓库元数据系统组件或储存库中获得和维护的这一信息。

3. 数据仓库项目开发进度管理

在数据仓库项目开发中,关键是使各个开发步骤紧密相连,一旦延期,必将导致整个计划的推迟。所以,应将数据仓库的开发作为一个工程项目来管理,其主要目的是运用系统工程方法制定工作计划,对计划的落实进行组织、监督与控制,保证按质按时开

发出预定目标的数据仓库。

(1) 数据仓库开发项目工作计划的编制

编制工作计划首先要确定:

- 开发阶段、子项目与工作步骤的划分;
- 子项目间的依赖关系与系统的开发顺序;
- 各个开发阶段、子项目与工作步骤的工作量。

在此基础上,根据项目的总进度要求,用某种或多种工程项目计划方法制定具体工作内容与要求,落实到具体人员,限定完成时间的行动方案——项目工作计划。

开发阶段是项目开发过程中的大段落,每个段落都要求有明确的成果。开发阶段的划分与采用的开发策略、开发方法有关,当综合性地采用多种开发策略与方法时,可以存在并列的开发阶段。数据仓库的开发阶段有规划分析、设计实施及应用支持和增强三个阶段。由于这三个阶段在实际开发过程中显得过于庞大,要对每个阶段进行子项目的划分。子项目可按系统的构成来划分。例如,数据仓库中的各个应用子系统、支持系统的平台、人员培训等。子项目确定后,还要分析它们之间的相互依赖关系,以便能在时间上安排先后开发顺序。

工作步骤是开发阶段的进一步细分,每个工作步骤需要确定完成一项具体的工作内容。

数据仓库的各个开发阶段、子项目及工作步骤工作量的核定,一般只能依据经验统计数给出估计数。数据仓库的系统分析与系统设计阶段的工作量在开发总工作量中占有很高的比率,这也表明系统开发的前期工作是非常重要的。大量的实践证明,在这些阶段工作做得仔细所付出的代价,将在系统的实施与运行阶段得到补偿。反之,将会付出很高的代价。

(2) 数据仓库开发项目进度的控制

在数据仓库实际开发中几乎没有一个数据仓库开发项目能按计划进度完成的,因此有必要对数据仓库开发项目的进度实施必要的控制。进度的控制主要通过计划执行的监督和检查、计划的延误的分析和解决等活动实现。

当计划发生延误时,需要进行具体原因的分析。一般讲,数据仓库开发进度的延期,除与其他工程项目同样存在的环境变化、资金不到位、人员变动等原因外,还有一些特殊的原因:

- 估计各项开发活动的工作量与实际工作量发生了较大的误差;
- 开发过程中产生不少未曾估计到的活动,使工作量增加;
- 由于需求或其他情况发生变化,完成的成果要做局部修改,造成返工。

上述导致计划不能如期进行的原因往往是不可避免的,但哪些活动延误,什么原因造成的延误,必须分析清楚。只有在明确问题的前提下,才能选取对策、解决问题或修

改计划。在总体上要把握项目开发进度，以使延误造成的损失减到最小。

针对不同的原因，可能采取的解决措施有：

- 减少开发中的不确定性——可以事先在工作计划中留有一定的时间宽裕度，例如开发工作量的估计值取上限，预设机动时间等；
- 开发过程中经常与用户交流，随时掌握企业的发展动向，及时明确遗留的不确定问题，减少返工现象；
- 当关键路线上的活动延误时，集中人力予以重点解决；
- 在上述措施难以有效解决时间延误时，可以调整原定计划，例如子项目先后次序的调整，部分工作步骤的提前或延迟。

4. 数据仓库项目开发的其他管理

(1) 数据仓库项目的质量管理

数据仓库开发项目工作计划除要控制开发进度外，还有质量控制问题。用户对开发进度延误有时可以容忍，但对质量上的欠缺是不能允许的。因此，质量上的控制往往显得更重要。质量的定义是从通用商业代码（UCC，universal commercial code）中借用的，质量就是“非常适用”的含义。高质量意味着它适用于工作。在对软件体系进行质量评估时，则要涉及各种不同的质量因素。表 13-5 对软件系统的质量特性进行了描述，而高质量的软件则意味着系统中的所有组件均能完美和谐地一起运作。

表 13-5 软件系统质量特性

正确性	系统符合使用说明，达到用户目标
可靠性	系统在异常情况下能够正常运行
有效性	消耗资源与产生结果的比率是定义好的并且能够达到
完整性	结果是一致的、有意义的，在控制范围内
可用性	操作和解释系统行为的资源经过定义，能够满足要求
可维护性	适应新的要求，或可以改进，以适应已有要求或其他质量特性
可测试性	测试系统符合定义的规范说明或功能
安全性	阻止、监督、报告或控制未经授权的个人或过程对系统的访问
可移植性	可将系统移植到不同的软硬件环境中
可重用性	以有用的、新的方式组合系统组件，适应新的要求或条件
互用性	系统可与其他系统通信或连接

根据系统质量的特性，可对数据仓库的质量进行定义。数据仓库的质量问题实质上应从数据仓库总体数据的质量方面考虑，具有良好质量的总体数据可使数据仓库具有良好的质量基础。总体数据的质量衡量标准包括数据的准确性，客观性，一致性，数据的二义性，数据的及时性，数据在时间上的一致性，数据的安全性和数据的可信度等。

数据的准确性、客观性与一致性是指数据仓库所提供的数据与现实情况是一致的,反映现实世界中的客观情况。如果数据与某人所相信的事实一致,而此人所相信的并不是实际的情况,那么数据只是准确的,而不是客观的。

在数据仓库中的数据不应有二义性,例如某个客户编号“03898”只能指某个具体的客户,而不是指多个客户。数据的无二义性对数据仓库中数据质量的影响是关键。

数据的及时性是指数据仓库中的数据应该及时反映客观现实。这样才能对管理决策者的决策分析活动提供有益的帮助。如果每日的销售分析数据是基于一个月前的销售数据,尽管这些数据是准确的、一致的和客观的,但是缺乏及时性,对决策分析也是无益的。

数据在时间上的一致性则是在决策分析活动中,要求数据仓库必须提供相同时段的数据。如果在销售分析中所提供的客户数据是1月份的,而销售数据却是3月份的,管理者据此所做的决策分析,可能要比没有数据做的决策分析效果更差。

数据的安全性要求数据仓库能够防止数据不受未经授权的访问和修改,能够防止偶然或恶意的破坏,但是又能够允许开发人员、最终用户有效完成工作的安全措施建设,是数据仓库开发中富有挑战性的工作。

数据的可信度则是用户对数据仓库的信任问题,如果数据仓库的可信度低,将导致用户不再使用数据仓库来进行决策分析,使数据仓库最终成为“货架数据”,用户将其搁置一边,不再使用,使数据仓库的开发应用陷入失败。数据仓库的可信度是建立在数据的准确性、一致性与及时性等基础上的。

(2) 数据仓库项目的风险管理

对于控制成本和保持数据仓库项目进度来说,实施项目风险管理是有效的方法。项目风险管理的主要责任是维护一个风险清单,合理地分配资源,以避免项目开发风险。数据仓库的风险包括数据质量的风险,系统草率集成或延期的风险,分布式网络系统中的连通失效风险,数据概括不适合而导致数据仓库应用效率降低的风险,以及在理解、准备和维护元数据方面的风险。

在进行风险管理过程中,可以通过资源配置,让风险出现,处理它,并且实施风险解决方案。由于数据仓库本身受到商业、技术和市场不确定性的限制,风险清单不可能是完整的;但是,通过风险清单可以提高系统处理风险的能力,加深对不确定的理解,提高对风险的防范和控制。

在开发数据仓库的过程中,总伴随有风险。不可能消除它们,但能控制它们。集中开发一个灵活的数据体系结构,将有助于在开发过程中较早地发现风险。在数据仓库的风险处理中主要对计划(成本)、质量和交付产品三个不确定因素处理。如果强调其中的一个,便将导致其余的因素有所变化。例如,缩短计划将导致质量或功能的降低;增强质量将导致计划改变,因为这将要执行更多的测试。必须在协调诸因素的情况下,管理

所有项目风险因素,实现真正的风险管理。

(3) 数据仓库项目的文档管理

数据仓库的文档是描述数据仓库从无到有的整个发展与演变过程以及演变过程中,各阶段不同状态的文字资料。这些文字资料可能保存在纸质文件上,也可能保存在系统中。数据仓库的开发需要以文档描述为依据,数据仓库的运行与维护更需要文档的支持。

系统文档不可能一次性形成,它是在数据仓库的开发、实施、运行和维护过程中通过编写、修改、完善与积累而形成的。如果没有系统文档或没有规范的系统文档,数据仓库的开发、运行和维护就会处于一种混沌状态,将严重影响数据仓库的整体质量。当开发人员发生变化时,问题就显得更加严重。可以说,系统文档是数据仓库的生命线,没有文档就没有数据仓库。文档的重要性决定了文档管理的重要性,文档管理是有序地、规范地开发与运行数据仓库所必须做好的重要工作。数据仓库文档的管理工作主要有:

- 文档标准与规范的制定;
- 文档编写的指导与督促;
- 文档收存、保管与借用手续的办理等。

文档的标准与规范要按行业或国际标准组织所指定的标准规定,结合数据仓库的特点在系统开发前或至少在所产生阶段前制定,用于指导与督促系统开发人员及系统使用人员及时编写有关的文档资料。为了保存文档的一致性与可跟踪性,所有文档需要集中管理。文档管理的好坏对系统的质量至关重要,而且必须有专人负责,形成制度化。

在数据仓库系统中,必须具备的文档有:需求文档,它详细指出有关数据仓库设计对决策分析的支持问题;数据仓库体系结构文档,其中包括数据仓库体系结构、各类数据模型和事实表、维表、元数据以及粒度的说明;设计文档,包括所有应用程序提取、转换、聚合和数据加载公式与测试计划;用户手册,应当详细说明对客户或最终用户的界面;工作文档,其中包括问题日志和解决方法、问题报告和解决方法、风险报告以及操作备忘录等。

13.4.3 数据仓库应用的阶段性

数据仓库的应用过程可以分成数据仓库的初步应用、熟悉应用和熟练提高三个阶段。

1. 数据仓库的初步应用阶段

数据仓库开发的失败,常常是由于企业中的许多管理人员不能很快地熟悉数据仓库和数据挖掘工具的应用,使其不能正常发挥应有的效果所致。因此在数据仓库实施的第一阶段——初步应用阶段,要通过对管理人员的有计划引导,使其对数据仓库、挖掘工具逐步熟悉,了解这些工具的性能和使用方法以及使用效果。管理人员在对这些信息工

具的逐步熟悉应用过程中,还可逐步加深对信息管理的认识,扩大数据仓库的应用成果,提高管理人员对数据仓库的应用信心。

为使管理人员逐步了解、熟悉数据仓库的操作,首先要让管理人员在数据仓库使用的第一个阶段,熟悉设计人员事先所设计的定制报表,进行用户的确认操作。这些查询通常涉及一些基本数据,例如企业商品销售的总收入、总销售量、总费用、总数量或产品总量等汇总数据;企业经营活动最大的销售量、收入、配送和服务发生的时间和地点等峰值;与过去相比,企业的当前销售存在哪些差异能反映企业销售变化的数据等;反映企业在市场竞争中的长处和短处的数据:企业最丰富或最缺乏的资金、产品、交通、人员等有效资源是什么。

利用这些报表体系,可以解释管理人员过去经常遇到的一些业务处理问题,反映企业的业务状况、企业所面对的市场及客户状况。这些状况的反映在这一阶段主要依靠综合性数据的描述,而不是今后阶段从数据仓库中深入挖掘时所使用的详细数据。

在初始应用阶段中,系统除提供这些定制的查询以外,还应该提供模糊查询手段,让管理人员进行一些事先无法确定但是具有创新的信息查询和使用。数据仓库还应提供一些容易访问数据的途径,使管理人员将数据仓库应用焦点逐步转移到以前从未被重视、从未被访问过的数据元素上,使管理人员逐步产生对一些客户行为或市场问题进行深层次了解的冲动和意识。因此在这一阶段,数据仓库应该满足管理人员对定制报表进行延伸查询,或提出对更多数据或将更多数据转变为信息的要求。

在数据仓库初始应用阶段中,忌讳对系统应用期望过高。避免由于管理人员在初步使用过程中,未能达到期望的目标而丧失对系统的使用信心。管理人员必须清楚,只有随着时间的推移,他们才能熟练掌握使用系统的能力,才能逐步获取他们所需要的数据,也才能逐步了解他们应该从数据仓库中获取哪些数据。

在数据仓库的初始阶段一定避免只允许管理人员利用定制查询对数据仓库进行操作,而禁止管理人员对历史数据进行操作的限制。这种限制企业信息资源应用的做法,也许能够保证数据的安全、质量、性能和成本,但是却给数据仓库的应用带来了严重的不良后果。这样不仅无法使企业能找到新的商业模式,而且使管理人员永远无法取得正确使用数据仓库的经验,使数据仓库永远停留在初始应用阶段。管理人员只有学会如何对数据仓库中的详细数据提出问题,学会如何对数据仓库中大量的、复杂的信息进行组合访问,才能使企业在数据仓库上的巨大投资迅速获得回报,给企业带来丰厚的高额利润。

2. 数据仓库的熟悉应用阶段

当管理人员在初始阶段学会利用模糊查询对业务本质进行深入查询后,就可进入数据仓库应用的第二阶段——数据的分析应用,即熟悉应用阶段。在初始阶段,当管理人

员利用数据仓库了解围绕业务所发生实质情况后,往往希望进一步了解为什么会发生这些情况。这就需要使用数据挖掘工具对数据仓库中的详细数据进行分析来完成,这就是数据仓库应用第二阶段的主要任务。第二阶段是对数据仓库价值认识的一个重要阶段。通过第二阶段的应用,企业才能真正感受到数据仓库的价值。在这个阶段中,管理人员可以利用数据仓库的信息对业务过程进行细分和分析。管理人员可向数据仓库提出这样一些查询:为什么我们没有达到或超过原先的预定销售目标?为什么销售数量如此低?为什么会带来如此好的销售结果或高额利润?我们在哪些地方取得了良好的管理成果?这种查询必须基于各种模型和详细的具有数学关系的数据挖掘,并且利用数据仓库向下挖掘数据,以获得精细信息,同时要能基于这些数据演绎出商业模式。这就使得业务管理人员发现一些在报表查询中难以发现的市场规律、客户变化趋势和商业模式。

在这个阶段中,人们开始注重于理解业务过程,开始关注:为什么客户的平均收益率会下降?年度客户变化为什么如此之大?企业的商业活动为什么没有达到预定计划要求?商品的销售为什么低于预期的计划?为什么客户会从我这里购买?为什么销售渠道的成本会下降?为什么客户的响应率比以前下降了?为什么不同商品之间的收益率会有如此大的差别?为什么在某个特定渠道中的需求成本会上升?如果单纯利用报表,需要大量的各种报表才能揭示这些规律及其内在的原因。利用数据挖掘工具,则比较容易获取这些信息。在第二阶段,管理人员开始逐步地利用数据挖掘工具对“为什么会发生”的一些问题进行深究。这种对过去现象进行深入研究的目的在于了解以往管理过程中未曾注意到的一些规律和因素,其目的是将其应用到企业目前的市场运作中,细分业务过程,对不同的客户采用正确的营销策略,提高企业的市场竞争能力。企业只有掌握了理解过去的的能力,才能清楚造成现状的原因,避免重蹈覆辙。

企业在数据仓库应用的第二阶段,才开始逐步熟悉各种挖掘工具,对大规模数据进行各种分析,发现市场中潜在的、未来的机遇。

3. 数据仓库的熟练提高阶段

数据仓库应用的第三阶段,也是数据仓库应用的最高境界——熟练提高阶段。在这一阶段能对未来做出知识性较强、可靠度较高的预测。只有掌握这种技术的企业才能在市场竞争中真正赢得主动,才能获取最大的利润,获得数据仓库投资的高额回报。要实现这个目标,就需要企业的数据仓库拥有“分析、建模”的能力,管理人员具有熟练应用数据仓库的能力。

企业管理部门和员工的数据仓库应用能力,是管理人员及数据仓库设计人员在数据仓库的阶段应用中逐步掌握,在数据仓库应用的第三个阶段中熟练应用,得到创新发展而形成的。

管理人员需要借助其运作数据仓库的熟练能力,对数据仓库进行各种查询,以达到

预定的目标。例如查询“什么样的客户正处于流失的危险中？”，以解决客户的流失问题；查询“客户将会购买什么样的商品和服务？”，用于达到市场细分的目的；查询“与客户取得联系的最好方式是什么？”，优化企业的销售渠道；查询“新产品怎样销售？”，以预测未来的商品销售需求。在这个阶段中实现这些预测性应用，需要系统具有先进的决策支持技术、并行查询功能、海量的详细数据，能够跨部门形成关于客户的整体信息，能够了解客户知识且对客户进行评分。这些评分范围包含客户的信用、支付能力、行为方式、购买倾向等。系统这些能力的形成，需要数据仓库设计人员在管理人员的积极参与之下才能完成。

数据仓库应用成功的企业，应该能够同时了解过去，分析现状和潜力，能以较高的精度预测未来。在数据仓库的应用中，从应用程度来说经历了学习、熟练和应用发展。如果能够达到应用发展，那企业的数据仓库就处在成功阶段，企业就能在市场竞争中获取优势。

13.4.4 数据仓库的运行管理

在建立数据仓库后，经过调试和必要的修改就进入了数据仓库系统的运行阶段。数据仓库在运行阶段，必然出现许多只有在实际运行中才能发现的缺陷。随着企业内部与外部环境的变化，数据仓库系统也会暴露出不足之处或与实际不相适应之处，这些都是在数据仓库运行管理过程中始终存在的需要解决的问题。数据仓库系统中的基本数据及历史数据与信息同样是企业的重要信息资源，为了充分利用这些资源，数据与信息的存储、维护、安全与保密也是运行管理中的重要工作。数据仓库运行阶段的管理工作目的与开发阶段有根本的区别，开发阶段要求经济地、按质按时地完成系统的开发，而运行阶段的管理目的是使数据仓库系统正常发挥其应有的作用，产生应有的效益。

数据仓库的日常运行管理是为了保证数据仓库长期有效地正常运转而进行的管理活动，具体有数据仓库运行情况的记录、日常维护及适应性维护等工作。

1. 数据仓库系统运行情况的记录

数据仓库的运行情况有正常、不正常与无法运行等，后两种情况应将所发生的现象、发生的时间及可能的原因做详细的记录。系统运行情况的记录对系统问题的分析与解决，有重要的参考价值。对此，一般在数据仓库系统中设置自动记录功能，但对一些重要的运行情况及所遇到的问题仍应做好书面记录。

数据仓库运行情况的记录应事先制定尽可能详细的规章制度，具体工作主要由使用人员完成。数据仓库的运行情况无论自动记录还是人工记录，都应作为基本的系统文档长期保管，作为数据仓库维护与评价的参考。

2. 数据仓库的日常维护

根据数据仓库的维护目的,可以分为日常维护和适应性维护两种。日常维护是定时内容地重复进行有关数据与硬件的维护,以及对突发事件的处理等。在数据方面,日常维护工作有备份、存档和整理等。

为安全考虑,每次在数据加载完毕后,要对更改过的或新增加的数据做备份。数据的正本与备份应分别存储在不同的磁盘或不同存储介质上。数据存档或归档是当数据积累到一定数量或经过一定时间间隔后,转入档案数据库的处理。数据的整理是关于数据文件或数据表的索引、记录顺序的调整等。数据整理可使数据的查询与引用更为快捷与方便,提高数据的完整性与正确性。数据仓库运行的初始化主要是指在进行数据加载以后所进行的概括数据处理,以及以月度或年度为时间单位的数据文件或数据表的概括数的预置等。

在数据仓库的系统硬件方面,日常维护主要有各种设备的保养与安全管理、简单故障的诊断与排除和易耗品的更换与安装等。

数据仓库系统的突发事件的发生,会给系统的运行带来严重的打击。突发事件应由企业的专业人员处理,对发生的现象、造成的损失、引起的原因及解决的方法等必须做详细的记录。

3. 数据仓库系统的适应性维护

由于数据仓库的外部环境总是处在一个不断变化的过程中,数据仓库系统也要不断地改进与提高。同时,数据仓库的初始建立难免存在一些缺陷与错误,随着数据仓库的运行逐步暴露出来。对这些暴露出的问题应该及时予以解决。为了适应环境的变化及克服自身存在的不足,还需要对数据仓库进行适当的调整、修改与扩充。

数据仓库系统的适应性维护是一项长期的有计划的工作,且以系统运行情况记录与日常维护记录为基础。其内容有:

- 数据仓库发展规划的研究、制定与调整;
- 数据仓库缺陷的记录、分析与解决方案的设计;
- 数据仓库结构的调整、更新与扩充;
- 数据仓库功能的增设与修改;
- 数据仓库数据结构的调整与扩充;
- 每个工作站点应用工具的功能重组;
- 系统硬件的维修、更新与添置;
- 数据仓库维护的记录及维护手册的修订等。

数据仓库的维护不仅为系统正常运行所必须,也是使系统始终适应系统环境的重要

保证。

13.4.5 数据仓库的评价

由于数据仓库的开发是一种螺旋形的逐步开发模式，开发人员往往围绕某几个主题开发数据仓库以后，需要根据用户的评价，对原有的数据仓库进行改进或开发新的数据仓库主题。这就离不开对已经开发应用的数据仓库的评价，只有正确地对开发应用的数据仓库评价，才能使数据仓库的开发应用不断地螺旋式上升，使数据仓库的应用越来越符合用户的需求，才能使巨额投资得到丰厚的回报。对数据仓库的评价还能检查数据仓库系统是否达到预期的目标，技术性能是否达到设计要求，系统的各种资源是否得到充分利用，经济效益是否理想，指出数据仓库的长处与不足，为改进与扩展提出建议。数据仓库的有利与不利之处，体现在定性与定量两个方面，目前一般采用多指标评价体系进行。

1. 数据仓库的评价内容

对数据仓库的评价可从技术和经济两方面进行。

(1) 技术上的评价内容

- 数据仓库的总体水平。例如系统的总体结构、网络规模、所采用技术的先进性等。
- 数据仓库功能的范围与层次。例如，功能的多少、难易程度或所对应的管理层次高低等。
- 数据仓库所包含的信息资源开发与利用的范围与深度。例如，企业内部信息与外部信息的比例、外部信息的利用率等。
- 数据仓库的质量。例如，数据仓库的及时性、正确性、可扩展性、可维护性与健壮性等。

- 数据仓库的安全与保密性。

- 数据仓库文档的完备性。

(2) 经济上的评价内容

主要是系统的效果和效益，包括直接的与间接的两个方面。

直接评价内容有：

- 数据仓库开发投资额；
- 数据仓库运行费用；
- 数据仓库运行所带来的新增效益；
- 数据仓库的投资回收期。

间接的评价内容有：

- 数据仓库对企业竞争能力的提高、客户关系的改善;
- 数据仓库对企业的体制与组织机构的改革、管理流程优化的影响;
- 数据仓库对企业战略决策支持、员工决策分析能力的影响。

数据仓库在运行与维护过程中会不断地发生变化,因此评价工作不是一次性的工作。数据仓库的评价应定期进行或每当系统有较大改进后进行。数据仓库的评价工作由数据仓库开发人员,数据仓库管理、维护人员和数据仓库用户等共同参与;评价方式可以是鉴定或评审等方式。

2. 数据仓库的评价指标

数据仓库的评价是一项难度较大的工作,它往往属于多目标评价问题。目前大部分的数据仓库评价只能就部分评价内容列出可度量的指标,不少内容还只能用定性方法做出叙述性的评价。数据仓库的评价指标可由数据仓库性能指标,与直接经济效益有关的指标和与间接经济效益有关的指标 3 个部分组成。

(1) 数据仓库性能指标

- 数据仓库人机交互的灵活性与方便性;
- 数据仓库的查询响应时间与决策分析时间;
- 数据仓库输出信息的正确性与精确度;
- 数据仓库在单位时间内的故障次数,以及故障时间与工作时间的比率;
- 数据仓库结构与功能的调整、改进及扩展,与其他业务处理系统交互或集成的难易程度;
- 数据仓库故障的诊断、排除和恢复的难易程度;
- 数据仓库安全保密措施的完整性、规范性和有效性;
- 数据仓库文档资料的规范、完整与正确程度。

(2) 与直接经济效益有关的指标

● 数据仓库的投资额,包括系统硬件、系统软件的购置和安装,应用系统的开发或购置所投入的资金,企业内部所投入的人力、材料等也应计入。对验收评价后所做的阶段评价,还要包括对数据仓库维护所投入的资金。

● 数据仓库运行费用,包括消耗性材料费用、系统投资折旧费用及硬件日常维护费等。

● 系统运行所增加的效益,主要反映在销售利润增加。新增效益可以采用总括性的,在同等产出或服务水平下有无数据仓库所致的可盈利客户数的增加来表示。具体衡量时,可以利用企业的客户流失率下降、优质客户数的上升、客户新价值的增加、促销成本的下降等有关数据来计算新增效益。

● 投资回收期。投资回收期为新增效益,逐步收回投入的资金所需要的时间,它是

反映数据仓库系统经济效益好坏的重要指标。

(3) 与间接经济效益有关的指标

间接经济效益是通过战略决策的改进, 客户流失的减少, 产品质量的提高等方式, 促使组织运行成本下降、利润增加而间接地获得的效益。对这些增加的利润只能是定性分析, 所以间接经济效益也称为定性效益。尽管间接经济效益难以估计, 但其对企业的生存与发展所发挥的作用往往大于直接经济效益。数据仓库的间接经济效益可体现在以下 4 个方面。

- 对组织为适应环境所做的结构、管理制度与管理模式等的变革产生的推动作用。
- 显著地改善战略决策质量, 提高新市场的开拓能力、提高优质客户的比率和客户的利用价值所发挥的作用。
- 促使管理人员获得新知识、新技术与新方法, 提高管理素质所起的促进作用。
- 数据仓库所导致的围绕主题的信息共享, 密切部门之间、管理人员之间的联系, 增强企业各部门的协作精神, 提高企业的凝聚力等方面的影响。



本章小结

数据仓库在信息管理中应用时, 需要根据实际情况分别对待。本章所介绍的几个应用案例, 分别涉及比较全面的综合性数据仓库, 小型的专用数据仓库, 基于业务处理系统的数据仓库等。

由于数据仓库与数据挖掘工具的配合使用, 可使公司了解客户非常详细的隐私资料。公司在使用这些客户隐私时, 一定要注意取得客户的认可, 且对客户隐私进行必要的处理, 控制客户的隐私外泄, 避免给公司带来不必要的麻烦。

数据仓库开放的巨大投资, 使组织对数据仓库的开发投资必须做好必要的定量与定性的投资和效益分析, 以取得对数据仓库投资开发的信心。

数据仓库的开发运行管理是数据仓库开发工作的关键, 首先需要组织适当的数据仓库开发组织, 其次要做好数据仓库的项目管理工作, 根据数据仓库应用的阶段性做好数据仓库的运行管理, 最后还需要根据实际应用情况对数据仓库进行评价。



习题

13-1 对 13.1.4 节中的航空公司对旅行社的机票回扣案例进行分析, 这个数据仓库的体系结构应该具备什么特点?

13-2 某个银行有个数据仓库与数据挖掘系统，该系统通过对信用卡使用模式研究，注意到你与一家房地产公司有数额较大的交易，银行主动向你提供关于房屋装修特别贷款的信息。讨论这种行为是否与你的隐私权发生冲突？能否给出其他的关于数据仓库应用中所发生的隐私权侵犯的问题。能否给出一个在数据仓库应用中既能锁定促销目标，又不侵犯客户隐私权的处理模式。

13-3 数据仓库的开发应用成本/效益分析应该如何进行？

13-4 数据仓库的开发管理除本章所介绍的一些内容以外，你认为还应该包含哪些管理？

13-5 数据仓库的应用可以分几个阶段进行？为什么要将数据仓库的应用分成若干阶段？是否有这个必要？

13-6 对于数据仓库的运行管理，你认为应该包含哪些内容？为什么？

- 1 Inmon W H.. Building Data Warehouse. Second Edition. John Wiley, 1996
- 2 R. Groth. Data Mining: Building Competitive Advantage. Prentice-Hall, 1997
- 3 ric Sperley. The Enterprise Data Warehouse, Volume I: Planning Building And mplementation. Prentice Hall PTR, 1999
- 4 Alex Berson, Stephen Smith, Kurt Thearling. Building Data Mining Applications for CRM. McGraw-Hill Companies, 2000
- 5 Lou Agosta. The Essential Guide to Data Warehousing. Prentice-Hall, 2000
- 6 William A Giovinazzo. Object-Oriented Data Warehouse Design. 2000
- 7 Ronald Swift. Accelerating Customer Relationships: Using CRM and Relationship Technologies. Prentice-Hall, 2001
- 8 Jiawei Han, Micheline Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2001
- 9 R. Kimball. The Data Warehouse Toolkit. John Wiley & Sons, 1996
- 10 E. Thomsen. OLAP Solutions: Building Multidimensional Information Systems. John Wiley & Sons, 1997
- 11 W. Ziarko, Rough Sets. Fuzzy Sets and Knowledge Discovery. Springer-Verlag, 1994
- 12 S. M. Weiss, N. Indurkha. Predictive Data Mining. Morgan Kaufmann, 1998
- 13 R. S. Michalski, I. Brakto, M. Kubat. Machine Learning and Data Mining: Methods and Applications. John Wiley & Sons, 1998
- 14 S. Chaudhuri, U. Dayal. An overview of data warehousing and OLAP technology. ACM SIGMOD Record, 1997
- 15 G. Piatetsky-Shapiro, W. J. Frawley. Knowledge Discovery in Database. Cambridge: MA: AAAI/MIT press, 1991
- 16 U. M. Fayyyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy. Advances in Knowledge Discovery and Data Mining. Cambridge: MA: AAAI/MIT Press, 1996
- 17 D. A. Keim. Visual techniques for exploring datasets, In Tutorial Notes. 3rd Int. Conf. Knowledge Discovery and Data Mining (KDD'97), Newport Beach, 1997(8)

- 18 A. Berson, J. Smith. Data Warehousing, Data Mining and OLAP. McGraw-Hill, 1997
- 19 R. Agrawal, R. Srikant. Privacy-preserving data mining. In Proc. 2000 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'00), 2000 (5): 439~450
- 20 M. J. A. Berry, G. Linoff. Mastering Data Mining: The Art Science of Customer Relationship Management. John Wiley & Sons, 1999
- 21 M. Berthold, D. J. Hand. Intelligent Data Analysis: An Introduction. Springer-Verlag, 1999
- 22 刘同明等. 数据挖掘技术及其应用. 北京: 国防工业出版社, 2001
- 23 Harjinder S. GILL. 数据仓库——客户/服务器计算指南. 王仲谋, 刘书舟译. 北京: 清华大学出版社, 1997
- 24 Joyce Bischoff, Ted Alexander. 数据仓库技术. 成栋, 魏立源译. 北京: 电子工业出版社, 1998
- 25 Michael J. Corey, Michel Abbey. Oracle 8 数据仓库分析、构建使用指南. 陈越等译. 北京: 机械工业出版社, 2000
- 26 W. H. Inmon. 数据仓库管理. 王天佑等译. 北京: 电子工业出版社, 2000
- 27 罗纳德. S. 史威福特. 客户关系管理——加速利润和优势提升. 杨东龙等译. 北京: 中国经济出版社, 2001
- 28 王珊等. 数据仓库技术与联机分析处理. 北京: 科学出版社, 1998
- 29 黄梯云. 管理信息系统. 北京: 高等教育出版社, 1999
- 30 罗运模等. SQL Server 2000 数据仓库应用与开发. 北京: 人民邮电出版社, 2001
- 31 陈国良等. 遗传算法及其应用. 北京: 人民邮电出版社, 1996
- 32 朱爱群. 客户关系管理与数据挖掘. 北京: 中国财政经济出版社, 2001
- 33 李纪华, 王珊. OLAP 的两种支持技术. 计算机世界, 1996-7-15
- 34 陈京民. 数据析取技术在市场营销中的作用. 商业研究, 2000 (3)
- 35 陈京民. IT/IS 战略管理若干问题探索. 电子与信息化, 2000 (9)
- 36 陈京民. 数据仓库开发的规划研究. 计算机与网络, 2000 (5)
- 37 陈京民等. 企业的客户关系管理 (CRM) 实施过程. 企业经济, 2002 (2)
- 38 邹雯、陈文伟. 数据开采中的遗传算法. 计算机世界, 1997-6-30
- 39 马建军, 陈文伟. 数据开采中的集合论方法. 计算机世界, 1997-6-30
- 40 王珊, 罗力. 从数据库到数据仓库. 中国人民大学数据与知识工程研究所, 1997
- 41 李德毅. 从数据库中发现知识的策略和方法. 计算机世界报, 1995-3-22
- 42 胡侃, 夏绍伟. 基于大型数据仓库的数据挖掘. 软件学报, 1998 (1)