

● 高等学校21世纪计算机教材

Visual Basic.NET

数据库 程序设计与实例

卢智勇 谭小丹 李光明 编著

冶金工业出版社

高等学校 21 世纪计算机教材

Visual Basic.NET 数据库 程序设计与实例

卢智勇 谭小丹 李光明 编著

北 京

冶金工业出版社

内 容 简 介

本书以中文版 Visual Basic.NET 为背景,从数据库编程基础、SQL Server、可视化数据库工具、ADO.NET 对象、数据适配器 DataAdapter、数据集 DataSet、数据绑定控件、Web 数据访问等方面详细介绍了 Visual Basic.NET 数据库编程的基本操作和使用方法。本书最后以进销存管理系统的设计与开发和 Web 教学日历管理系统的开发两个实例阐述了数据库的开发思路和步骤。

本书内容丰富、重点突出、操作简练、简明易懂,理论介绍与实例演示并重,图文并茂,具有很强的实用性。另外,本书按教材体例编写,循序渐进,配有适当的练习题。因此,本书适合于使用 Visual Basic.NET 开发数据库的初学者,也可作为高等学校计算机专业的教学用书,同时,对从事数据库开发的人员也是一本很好的参考书。

图书在版编目(CIP)数据

Visual Basic.NET 数据库程序设计与实例 / 卢智勇等

编著. —北京:冶金工业出版社, 2005.7

ISBN 7-5024-3773-8

I. V... II. 卢... III. BASIC 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 065346 号

出版人 曹胜利 (北京沙滩嵩祝院北巷 39 号, 邮编 100009)

责任编辑 戈兰

佛山市新粤中印刷有限公司印刷; 冶金工业出版社发行; 各地新华书店经销

2005 年 7 月第 1 版第 1 次印刷

787mm × 1092mm 1/16; 22.25 印张; 658 千字; 348 页

35.00 元

冶金工业出版社发行部 电话: (010) 64044283 传真: (010) 64027893

冶金书店 地址: 北京东四西大街 46 号 (100711) 电话: (010) 65289081

(本社图书如有印装质量问题, 本社发行部负责退换)

前 言

一、关于数据库和 Visual Basic.NET

随着计算机被应用于管理领域范围的不断扩展,需要计算机处理的数据量急剧增长。为了解决多用户、多应用程序共享数据的需求,出现了数据库技术。

数据库无论对于公司还是个人而言,都是十分重要的。现代生活和数据库息息相关,例如:银行依靠数据库来管理支票账目,从而提高工作效率,而且在整个处理过程中不会出现错误。

随着数据库的应用开发,Visual Basic 成为很多程序员的首选编程语言,因为 Visual Basic 易学、易于调试、可以比其他编程语言更迅速的建立复杂的应用程序。此外,Visual Basic 中内置了许多用于开发数据库应用程序的专业化支持。程序员使用 Visual Basic 不仅可以开发传统的面向表格的数据库应用程序,还可以开发应用于 Web Server 的后台数据库,比使用传统的 Perl 脚本和 Active Server Pages 的效率都要高。现在 Visual Basic.NET 对数据库编程的强大支持,使得开发人员可以像在数据库管理系统中一样在 Visual Basic 中进行数据库应用程序的开发,既简单方便又快速高效。

二、本书内容结构

本书分为 10 章,各章内容如下:

第 1 章: Visual Basic.NET 数据库编程基础。主要介绍了数据库的发展、目前的几种数据库、数据库的一些基本概念和术语、设计数据库的规则、SQL 语言、Visual Basic.NET 访问数据库的几种方案以及 XML,并建立了一个关系数据库 sales 的实体模型。

第 2 章: 使用 SQL Server。主要介绍了 Client/Server 基础、SQL Server、使用 SQL Server 建立数据库对象,并给出了一个编程示例。

第 3 章: 可视化数据库工具。主要介绍了服务器资源管理器、数据库设计器、查询和视图设计器、解决方案资源管理器、XML 设计器、报表设计器等可视化数据库工具,并给出了相应的编程实例。

第 4 章: ADO.NET 概述。主要介绍了 ADO.NET 的工作原理、ADO.NET 对象以及 ADO.NET 和 XML 之间的关系,最后给出一个 ADO.NET 的应用程序。

第 5 章: ADO.NET 数据适配器。主要介绍了如何创建数据适配器、配置数据适配器参数、数据适配器中的表映射、数据适配器的事件,并给出一个实例。

第 6 章: ADO.NET 数据集。主要介绍了 DataSet 的结构、如何使用 DataSet 和 DataView 以及并发性控制与处理,最后给出一个设计实例。

第 7 章: 数据绑定控件。主要介绍了用于数据库访问的常用数据绑定控件: ListBox 控件、ComboBox 控件、CheckedListBox 控件、DataGrid 控件、ErrorProvider 控件和用户控件,最后给出了一个实例。

第 8 章: Web 应用。主要介绍 ASP.NET、Web 窗体、Web 服务,并给出一个集成 Web 服务的订单查询系统的应用实例。

第 9 章: 综合实例 1——进销存管理系统的设计与开发。通过一个进销存管理系统的开发,介绍了使用 Visual Basic.NET 开发数据库的全过程。

第 10 章: 综合实例 2——Web 教学日历管理系统的开发。通过一个 Web 教学日历管理系统的开发,介绍了如何使用 Visual Basic.NET 开发 Web 数据库。

三、本书特点

(1) 组织合理,内容全面。

本书以中文版 Visual Basic.NET 为背景,通过建立一个关系数据库 sales,并以此数据库为基础详细介绍了 Visual Basic.NET 数据库编程的基本操作和使用方法。

(2) 实例丰富,通俗易懂。

本书除介绍 Visual Basic.NET 数据库编程基础理论外,还备有丰富的实例,这些实例涉及了程序设计的各个方面,既有简单的理论阐述,又有详细的制作过程,使读者能轻松快速地使用 Visual Basic.NET 编写数据库应用程序。为巩固对每章的学习,书中给出了一定数量的练习题,以便使读者在较短的时间内掌握 Visual Basic.NET 数据库的编程。

四、本书适用对象

本书适合于使用 Visual Basic.NET 开发数据库的初学者,也可作为高等学校计算机专业的教学用书,同时,对从事数据库开发的人员也是一本很好的参考书。

本书主要由卢智勇、谭小丹、李光明编写,刘国庆也为本书的写作付出了辛勤的劳动。由于编者水平有限,编写仓促,书中不妥之处,恳请读者批评指正,编者将不胜感激。

虽然经过严格的审核、精细的编辑,本书在质量上有了一定的保障,但我们的目标是力求尽善尽美,欢迎广大读者和专家对我们的工作提出宝贵建议,联系方法如下:

电子邮件: service@cnbook.net

网址: www.cnbook.net

本书电子教案、所有源代码和练习题参考答案可在本网站免费下载,此外,该网站还有一些其他相关书籍的介绍,可以方便读者选购参考。

编 者
2005 年 5 月

目 录

第 1 章 Visual Basic.NET 数据库编程基础..... 1

1.1 数据库概述.....	1
1.1.1 数据库的发展.....	1
1.1.2 桌面数据库.....	2
1.1.3 分布式数据库.....	2
1.1.4 基本概念和术语.....	3
1.2 关系数据库设计.....	5
1.2.1 规划关系数据库.....	5
1.2.2 设计一个关系数据库.....	6
1.3 SQL 语言.....	8
1.3.1 SQL 的特点和基本功能.....	8
1.3.2 SQL 数据类型.....	9
1.3.3 SQL 语句.....	10
1.4 Transact-SQL.....	11
1.4.1 T-SQL 语言.....	11
1.4.2 T-SQL 函数.....	15
1.5 存储过程.....	24
1.5.1 设计存储过程.....	24
1.5.2 创建、执行存储过程.....	25
1.6 Visual Basic 的数据访问技术.....	27
1.6.1 Visual Basic 支持的数据库.....	27
1.6.2 Visual Basic 的数据访问方式.....	28
1.6.3 Visual Basic 数据访问中相关 缩写词.....	29
1.7 从 ADO 到 ADO.NET.....	30
1.7.1 ADO.NET 简介.....	30
1.7.2 ADO.NET 的优点.....	32
1.8 XML.....	32
1.8.1 XML 的元素、属性和类型.....	33
1.8.2 XML 架构.....	34
小结.....	34
综合练习一.....	35
一、选择题.....	35
二、填空题.....	35
三、思考题.....	35

四、上机实践..... 35

第 2 章 使用 SQL Server..... 36

2.1 Client/Server 基础.....	36
2.1.1 基本概念.....	36
2.1.2 双层 Client/Server 模型.....	37
2.1.3 三层 Client/Server 模型.....	37
2.1.4 三层 Client/Server 的优点.....	38
2.2 SQL Server.....	38
2.2.1 SQL Server 概述.....	38
2.2.2 SQL Server 的 Client/Server 特性.....	39
2.2.3 SQL Server 数据库结构.....	41
2.2.4 SQL Server 数据类型.....	41
2.3 使用 SQL Server 建立数据库对象.....	45
2.3.1 Enterprise Manager 简介.....	45
2.3.2 建立数据库.....	46
2.3.3 建立数据表.....	48
2.3.4 管理安全性.....	50
2.4 实例演练——建立数据库.....	51
小结.....	55
综合练习二.....	55
一、选择题.....	55
二、填空题.....	55
三、思考题.....	55
四、上机实践.....	55

第 3 章 可视化数据库工具..... 56

3.1 可视化数据库工具概述.....	56
3.2 使用“服务器资源管理器”.....	56
3.2.1 “服务器资源管理器”主窗口介绍.....	57
3.2.2 使用“服务器资源管理器”建立 数据连接举例.....	59
3.3 使用数据库设计器.....	60
3.3.1 表设计器.....	61
3.3.2 表的“属性页”对话框.....	62
3.3.3 数据库关系图.....	63

3.4 使用查询和视图设计器	65	5.2.1 使用服务器资源管理器创建 数据适配器	107
3.4.1 查询和视图设计器	65	5.2.2 使用工具箱创建数据适配器	109
3.4.2 建立查询	67	5.3 配置数据适配器参数	111
3.5 使用“解决方案资源管理器”	69	5.4 数据适配器中的表映射	113
3.6 使用 XML 设计器	70	5.5 数据适配器的事件	114
3.7 使用报表设计器	74	5.5.1 事件处理	114
3.7.1 Crystal Report Designer 的组成	75	5.5.2 FillError 事件	115
3.7.2 Crystal Reports 专家	76	5.5.3 RowUpdated 事件	116
3.7.3 报表控件	77	5.5.4 RowUpdating 事件	117
3.7.4 设计并运行新报表	77	5.6 实例演练——SqlDataAdapter 控件 的使用	118
3.8 实例演练——可视化数据库工具 的使用	80	小结	121
小结	83	综合练习五	121
综合练习三	84	一、选择题	121
一、选择题	84	二、填空题	122
二、填空题	84	三、思考题	122
三、思考题	84	四、上机实践	122
四、上机实践	84		
第 4 章 ADO.NET 概述	85	第 6 章 ADO.NET 数据集	123
4.1 ADO.NET 工作原理	85	6.1 DataSet 的结构	123
4.2 ADO.NET 对象	86	6.1.1 DataTable 对象	123
4.2.1 Connection 对象	86	6.1.2 DataColumn 对象	126
4.2.2 Command 对象	89	6.1.3 DataRow 对象	127
4.2.3 DataReader 对象	91	6.1.4 DataRelation 对象	129
4.2.4 DataAdapter 对象	93	6.1.5 约束	131
4.2.5 DataSet 对象	96	6.1.6 创建和使用数据表	132
4.3 ADO.NET 和 XML	99	6.2 使用 DataSet	138
4.4 实例演练——建立一个简单的 ADO.NET 应用程序	100	6.2.1 创建 DataSet	138
小结	105	6.2.2 使用 DataSet 更新数据库	143
综合练习四	105	6.2.3 DataSet 的处理	144
一、选择题	105	6.3 并发性	146
二、填空题	106	6.3.1 并发控制	146
三、思考题	106	6.3.2 处理并发错误	148
四、上机实践	106	6.4 使用 DataView	149
第 5 章 ADO.NET 数据适配器	107	6.5 实例演练——销售员清单管理	153
5.1 概述	107	小结	162
5.2 创建数据适配器	107	综合练习六	163
		一、选择题	163

二、填空题.....	163	综合练习八	210
三、思考题.....	163	一、选择题.....	210
四、上机实践.....	163	二、填空题.....	210
第7章 数据绑定控件.....	164	三、思考题.....	210
7.1 应用数据控件访问数据库.....	164	四、上机实践.....	210
7.2 ListBox 控件.....	164	第9章 综合实例1——进销存管理系统的	
7.3 ComboBox 控件.....	167	设计与开发.....	211
7.4 CheckedListBox 控件	169	9.1 进销存管理系统简介	211
7.5 DataGrid 控件.....	171	9.2 数据库设计.....	212
7.6 ErrorProvider 控件.....	172	9.2.1 建立实体模型	212
7.7 用户控件.....	174	9.2.2 创建数据库.....	214
7.8 实例演练——DataGrid 控件的使用	174	9.3 主界面和主菜单	215
小结	180	9.3.1 主界面设计.....	215
综合练习七	181	9.3.2 模块编程.....	217
一、选择题.....	181	9.3.3 frmMain 窗体代码	220
二、填空题.....	181	9.4 用户菜单.....	227
三、思考题.....	181	9.4.1 注销子菜单.....	227
四、上机实践.....	181	9.4.2 登录子菜单.....	228
第8章 Web 应用	182	9.4.3 退出子菜单.....	230
8.1 ASP.NET 概述	182	9.5 数据库菜单.....	230
8.2 Web 窗体	182	9.5.1 业务员数据表子菜单	230
8.2.1 Web 窗体简介	182	9.5.2 客户数据表子菜单	239
8.2.2 创建基本的 Web 窗体	183	9.5.3 厂商数据表子菜单	244
8.2.3 验证 Web 窗体的数据输入.....	186	9.5.4 商品数据表子菜单	249
8.3 Web 窗体的数据访问	190	9.5.5 进货数据表子菜单	253
8.3.1 Web 窗体数据访问简介	190	9.5.6 销售数据表子菜单	257
8.3.2 Web 数据访问策略	191	9.5.7 退货数据表子菜单	262
8.3.3 在 Web 窗体中通过 DataGrid 控件		9.5.8 库存数据表子菜单	266
显示数据.....	192	9.6 交易菜单.....	270
8.3.4 在 Web 窗体中创建		9.6.1 进货登记子菜单	270
只读数据访问	195	9.6.2 销售登记子菜单	283
8.3.5 在 Web 窗体中更新数据	197	9.6.3 退货登记子菜单	293
8.4 Web 服务	200	9.7 统计菜单.....	300
8.4.1 Web 服务简介	200	9.7.1 统计结果显示窗体	300
8.4.2 创建 Web 服务	201	9.7.2 各子菜单的编程实现	304
8.4.3 使用 Web 服务.....	203	9.8 报表菜单.....	304
8.5 实例演练——订单查询系统.....	204	9.8.1 报表设计.....	305
小结	210	9.8.2 报表编程实现	308

9.9 窗口和帮助菜单	309	10.2.2 数据库概念结构设计	317
9.9.1 窗口菜单	309	10.2.3 数据库逻辑结构设计	317
9.9.2 帮助菜单	309	10.3 数据库结构的实现	318
9.10 系统运行	310	10.3.1 创建数据库	318
小结	313	10.3.2 网络登录界面	319
第 10 章 综合实例 2——Web 教学日历		10.3.3 用户注册页面	323
管理系统的开发	314	10.3.4 注册成功界面	324
10.1 系统设计	314	10.3.5 用户信息管理页面	325
10.1.1 系统功能分析	314	10.3.6 用户基本信息修改页面	331
10.1.2 功能演示	315	10.3.7 新建教学日历界面	333
10.2 数据库设计	316	10.3.8 教学日历操作页面	336
10.2.1 数据需求分析	316	10.3.9 报表输出界面	345
		小结	348

第 1 章 Visual Basic.NET 数据库编程基础

在 Visual Basic.NET 推出之前 Visual Basic 是 6.0 版本。用 Visual Basic 6.0 编写的数据库应用程序在当时的时间和环境下是没有什么问题的,但是它们不适合 Internet 时代,因为它们没有真正的伸缩性。而 Visual Basic.NET 克服了 Visual Basic 6.0 这一缺陷,能编写真正适合 Internet 的数据库应用程序。

在使用 Visual Basic 创建数据库应用程序之前,必须对数据库有很好的了解,并且知道 Visual Basic 如何访问数据库。本章主要介绍数据库的发展、目前的几种数据库、数据库的一些基本概念和术语、设计数据库的规则、Visual Basic 访问数据库的几种方案以及 ADO.NET 的优点,最后简单介绍了 XML。

1.1 数据库概述

有多种多样的方法可以在计算机上存储信息,但是最终都归结为把信息存储在文件中,而文件放在磁盘驱动器上。根据存储方式的不同,有两种类型的文件:顺序存取文件和随机存取文件。顺序存取文件把文件当作一个单元存取,随机存取文件允许用户访问文件的某一部分。

对顺序文件中的数据进行操作,必须读写整个文件,传统的文件系统基本上属于这种方式。在大多数的应用程序中,仅仅需要及时存取特定点的数据子集,没有必要为了存取一小部分数据而对整个文件进行操作。为了实现对文件的随机存取,随机存取文件把文件分成许多称为记录的块,每个记录都有惟一的记录号,记录号是根据记录在文件中的相对位置决定的,可以根据记录号来读取或者更新一条记录,于是有了数据库的雏形。

1.1.1 数据库的发展

20 世纪 60 年代后期以来,随着计算机被应用于管理领域范围的不断扩展,需要计算机处理的数据量急剧增长。为了解决多用户、多应用程序共享数据的需求,出现了数据库技术。

数据库技术解决了在使用随机存取文件时所面临的几个问题:

- (1) 并发。
- (2) 数据安全。
- (3) 进行备份和使用事务日志。

并发是指保证两个以上同时运行的程序在对数据进行操作时不会互相干扰。

数据安全是指某一用户只能读写特定的文件,而不能存取权限以外的其他文件。

进行备份可保证在出现意外事故时数据的损失最小。事务日志是记录所有对数据的改变操作的文件,仅包含对文件的修改内容。

数据库的出现使信息系统的研制从以加工数据的程序为中心转变到以共享的数据库为中心来进行。数据库开始是在大公司或机构中用来进行大规模事务处理的,后来,伴随着微机的普及,数据库技术被应用到微机上,出现了像 FoxBase、Access、FoxPro 这样的个人数据库应用。这类数据库系统也叫桌面数据库,一般在单机环境下使用。

随着计算机数量的增加,人们对软件和数据资源的共享越来越迫切,传统的单机环境下的数据库系统已不能满足用户的需求,用户希望能通过网络方便地访问跨部门、跨产业、甚至跨地区的数据资源。网络技术的迅猛发展,为网络数据库的产生奠定了基础。

网络数据库是以网络环境作为下层支撑的数据库系统的总称,一般有以下特点:

(1) 数据库或数据库的处理可以被分布在网络中不同节点上完成。数据库的处理包括数据库的应用和数据库的服务两部分。

- (2) 数据资源能够在网络的各个节点中方便的传输和访问。
- (3) 能支持远程数据的同步查询和更新。
- (4) 具有必要的数据恢复能力。
- (5) 多用户系统。

网络数据库的发展离不开分布式数据库理论的提出,正是分布式数据库理论首次把计算机网络技术和数据库技术结合起来,才使得网络数据库的发展成为可能,并标志着网络数据库的产生。从七十年代初分布式数据库技术的兴起,到如今基于客户/服务器体系结构的数据库、数据库的互连、互操作概念的相继提出,数据库一刻也离不开网络,并将一直和网络技术结合在一起。

1.1.2 桌面数据库

桌面数据库是一类数据库软件,因为这些数据库都使用 ISAM 文件,所以有时也叫 ISAM(Indexed Sequential Access Method,索引顺序存取方法)数据库。目前市场上有很多种桌面数据库,例如 Microsoft Access、Microsoft Foxpro、Borland Paradox 等。这些数据库产品在很多方面都彼此不同,但作为桌面数据库,还是有一些共同点:

- (1) 数据存放在 ISAM 文件中。
- (2) 数据文件是可自描述的。
- (3) 有自己的语言和数据类型。
- (4) 有自己的解释程序,可以运行自己语言编写的程序。
- (5) 提供标准的 DBMS(数据库管理系统)功能。
- (6) 是专门为在个人计算机上运行而设计的。
- (7) 可以处理记录封锁,允许多个线程和应用程序同时访问数据。
- (8) 提供对记录中的数据类型检查功能,保证数据的完整性。
- (9) 提供基于集合的操作,一条命令可完成对成千上万条记录的处理。

桌面数据库的数据文件可以通过局域网供远程计算机访问,此时运行应用程序的计算机叫客户机,数据驻留的计算机叫服务器。当数据文件通过局域网访问时,数据要在客户机上进行处理,所有数据和索引文件都要从服务器一端通过网络传送到客户机进行处理,因此不适合建立高性能的客户/服务器应用系统。另外桌面数据库是为个人计算机设计的,其能力和数据吞吐量都是有限的,用户数量和数据文件的大小也有一定限制。

1.1.3 分布式数据库

分布式数据库是在集中式数据库的基础上,由计算机网络的引入而发展起来的新技术。可以简单的理解为:集中式的数据库是建立在一个单一计算机系统上的数据库,而分布式数据库是建立在计算机网络各个节点上的数据库的集成。分布式数据库一般指数据资源(物理库)在物理上存储在一个含有多个数据库管理系统的计算机网络的多个节点中,但是逻辑上把它们视为一个统一的逻辑数据库进行处理。即物理上的分布性和逻辑上的统一性。

与集中式数据库比较,分布式数据库有它本身的特点:

(1) 节点自治:由于在分布式数据库中存在局部和全局两种应用,因此对数据库的数据操作有两种控制,产生了全局和局部两种管理员。分布式数据库强调局部管理员的作用,即节点的数据由其局部管理员进行管理,节点间的协调工作也可由局部管理员来完成。

(2) 数据独立性:分布式数据库使应用程序所面对的分布存放的数据库就像集中数据库一样,不必考虑它的分布性。

(3) 数据冗余:由于节点自治概念的出现,分布式数据库的局部应用需要有一定的数据冗余度用来提高系统可靠性,减少数据损失,以及增加数据复制,减少运行时间和通信费用。

(4) 完整性、可恢复性、并发控制。

分布式数据库从体系结构上讲,通常可分为综合型和联合型两种。综合型分布式数据库体系结构是指分布式数据库建立之前还没有建立独立的局部 DBMS,系统设计人员在综合权衡用户的需求之后,设计出一个完整的数据库管理系统,然后把系统的功能按照确定的方案选择,分散配置在这个分布式环境中。其中,每一个节点的数据库软件都谙熟整个网络的情况,也了解其他各节点的情况。从整体上看,它们组成了一个完整的系统,各个软件包之间的联系是靠网络存取进程间的通信服务来维持。这种体系结构如图 1-1 所示。

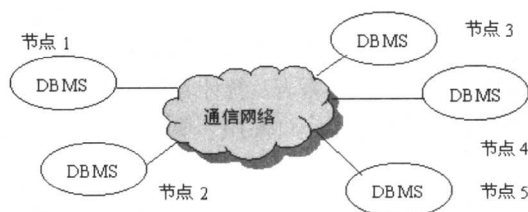


图 1-1 综合型体系结构

联合型的分布式数据库体系结构是以现存的集中式 DBMS 为基础,构成一个用分布式调制机制来完成分布处理功能的分布式数据库管理系统。联合型体系结构按各节点所保存的原数据库管理系统的异同,分为两种体系。如果每个节点的局部 DBMS 支持同一种数据模型、命令语言和查询语言,称为同构联合型体系结构,如图 1-2 所示。

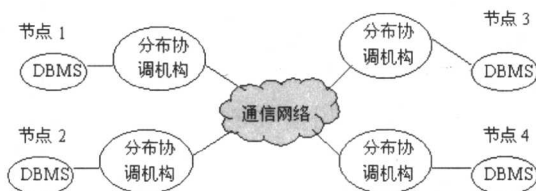


图 1-2 同构联合型体系结构

如果不同节点的局部 DBMS 支持的数据模型、命令语言和查询语言各不相同,称为异构联合型体系结构,如图 1-3 所示。

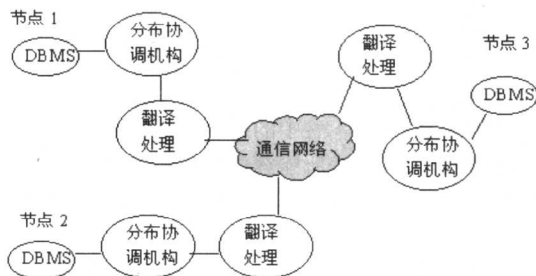


图 1-3 异构联合型体系结构

分布式数据库的查询处理和集中式数据库的查询处理相比较有两个特点:第一是由于查询所涉及的数据可能分布于网络不同的节点上,而节点间的数据要花费较多的时间,因此查询处理应考虑数据传送的开销;第二是由于各节点上的数据库可以独立的做一些查询处理,因此查询处理应考虑如何让位于各节点的数据库做并行处理。

1.1.4 基本概念和术语

为便于对数据库编程有个全面的了解,在进入实际编程之前,先介绍一下在数据库编程中经常使

用的一些概念和术语:

1. 数据库

对于早期的数据库系统,一个数据库文件就是一个二维数据表,随着数据库技术的发展,数据库所包含的内容不断得到扩展和延伸。对于现在人们所接触的大多数数据库系统来说,它已不仅仅是一张二维数据表的概念了,它存储的是若干二维数据表、数据表与数据表之间的关系和视图、触发器等众多内容。

2. 数据表

数据表用于存储数据,它以行/列方式组织,可以使用 SQL 从中获取、修改和删除数据。数据表是数据库的基本元素,数据库中一般有若干张表,而数据库本身则是管理这些表的。数据表有直观、方便和简单的特点,在生活中随处可见,如学籍表、统计表、月报表。数据表是以一组相关的数据行来排列,可以把它想象成行列整齐的表格,每一行称作一条记录,每一列称作一个字段。如表 1-1 所示为一个学生成绩表。

表 1-1 学生成绩表

学号	院系	班号	姓名	成绩
20000101	计算机系	计 2000	刘明	85
20000102	计算机系	计 2000	李敏	91
20000103	计算机系	计 2000	王文	88
20000104	计算机系	计 2000	赵辉	93

3. 临时表

临时表与数据表有着相似的性质,也可以看成是一张二维表格,对它的操作和数据表一样,不过它是只读的。临时表也是由一条条记录组成,而每条记录由若干个字段组成。临时表是程序根据用户的需求查询相关数据表后生成的,往往在运行中存在于内存缓冲区中或以临时文件(.tmp 文件名)存在于磁盘上,运行完后自行释放。

4. 记录

记录指表中的一行,在一般情况下,记录和行的意思是相同的,它惟一地描述了一个有意义的事物。在表 1-1 中,每个学生所占据的一行是一条记录,描述了一个学生的情况。可以看出,在一个表中不允许有重复的记录,即每个记录都是惟一的。

5. 字段

字段是表中的一列,在一般情况下,字段和列所指的内容是相同的,它的每一个元素描述了该元素所在记录的某一方面性质。数据表是由其包含的各种字段定义的,每个字段描述了所包含的数据。在表 1-1 中,如“学号”一列即为一个字段。表中的每个字段都有一定的数据类型和变化范围,具体情况要视具体的数据库管理系统而定。每个表所能容纳的字段的数量,在不同的数据库管理系统中是不同的。字段可以包含各种字符、数字,甚至图形、声音、动画等多媒体信息。

6. 域

域是指字段的取值范围。如表 1-1 中对于“成绩”字段的取值范围为 0~100,任何一个学生记录的成绩信息都在这个范围之内。

7. 索引

索引是建立在表上的单独的物理数据库结构,基于索引的查询将使数据的获取更为快捷。索引可以非惟一的,也可以是惟一的,主要看这些字段是否允许重复。主索引是表中的一列或多列的组合,作为表中记录的惟一标识。外部索引是相关联的表的一列或多列的组合,通过这种方式来建立多个表之间的联系。

8. 键

键就是数据表中为快速检索而被索引的一个字段或多个字段,键可以是惟一的,也可以是不惟一

的, 主要看这些字段是否允许重复。惟一键可以指定为主键, 用来惟一标识数据表的每一行。如表 1-1 中的“学号”字段惟一标识了一个学生, 因此可以作为主键。

9. 关系

数据库可以由多个数据表组成, 表与表之间可以用不同的方式相互关联。在数据库中, 关系是建立在两个表之间的链接, 以表的形式表现其间的链接, 使数据的处理和表达有更大的灵活性。关系的概念虽然很简单但却很重要。在目前流行的数据库系统中, 有三种关系: 一对一关系、一对多关系和多对多关系。

10. 视图

视图是一个与真实表相同的虚拟表, 用于限制用户可以看到和修改的数据量, 从而简化数据的表达; 是从一个或几个基本表导出的表, 其本身不独立存储在数据库中。数据库中只存储视图的定义, 不存储对应的数据, 运行时系统自动将视图链接到相关的数据表。

11. 触发器

触发器是一组 SQL 命令的集合, 当用户对表进行插入、删除或更新时触发自动执行。

12. 存储过程

存储过程是一个编译过的 SQL 程序, 在该过程中, 可以嵌入条件逻辑、传递参数、定义变量和执行其他编程任务。

13. 数据库引擎

它是一种管理数据怎样被存储和检索的软件系统。数据库的基本功能由数据库引擎提供。

1.2 关系数据库设计

DBMS 从产生直至发展到现在, 出现了多种类型。按其数据模型来分, 主要有层次数据库、网状数据库和关系数据库。前两者在六七十年代较为流行, 然而它们表示数据之间的联系太过于复杂, 现在已经很少使用, 只是在讲解数据库的时候才有所提及。

关系数据库模型是由 Ted Codd 于 1970 年提出的, 他认为数据库应当以表的形式 (即关系) 组织数据。关系数据库以关系的数学理论为基础, 是数据库技术的一项重大突破。在关系数据库中, 用户不必关心数据的存储结构, 与此同时, 关系数据库的查询可以用高级语言来描述, 这大大提高了查询效率。正是由于关系数据库的简单严谨的特点, 它在过去二十多年来有了长足的发展, 现在已经成为事实上的标准, 当今几乎所有的数据库产品都是基于关系数据库的。无论是 Access、FoxPro、dBase, 还是开放数据库互连 (ODBC) 产品如 SQL Server、Oracle 等数据库管理系统都是关系数据库系统。其中的开放数据库互连产品如 SQL Server、Oracle 还可以和计算机网络一起组成分布式数据库系统。

数据库设计是数据库应用的一个关键因素, 设计结构合理、功能齐全的数据库对于提高数据库应用程序的开发效率和程序的性能都是非常重要的。数据库设计是指对于一个给定的应用环境, 构造最优的数据模式, 建立数据库, 使其能够有效地存储数据记录, 并能满足各种应用需求。数据库的设计通常是以一个已经存在的数据库管理系统为基础的, 在设计一个数据库时, 应该注意将数据库的设计与应用系统的设计结合起来, 也就是说, 要注意把结构 (数据) 设计和行为 (处理) 设计结合起来。数据库设计质量将直接影响到系统中各个处理过程的质量和运行性能。一个设计失败的数据库往往到了应用程序的开发阶段还要不断地修改。

在本节中将建立一个简单的关系数据库表 Sales, 这个数据库存储的是某公司的销售订单情况等内容。数据库 Sales 有三张表 Customers、Orders 和 Salers, Customers 表存储客户的信息, Salers 表存储销售员的信息, Orders 表存储客户的订单。

1.2.1 规划关系数据库

进行数据库的规划是所有数据库编程的第一步, 也是最重要的一步。一个经过精心设计的数据库

可非常方便的访问它，可保证在后期的程序设计中能够随心所欲的存取任何关心的数据。因此，在进行正式编程之前，一定要花足够的精力来进行数据库的规划。这里所说的规划主要指以下几件事情：

首先，列出客户所关心的所有数据和建立这个数据库的主要目的。这可以通过收集各种相关的数据报表和表格来完成。在此期间，一定要经常和客户保持沟通，收集的数据一定要完整，要尽可能的理解每个数据的含义。需求分析是数据库设计的第一阶段，不断的调查与研究，了解组织机构的情况，了解部门的业务流程等系统需求，对于设计好数据库是非常重要的。在需求分析中，应该对现实世界要处理的对象进行详细地调查，在确定系统功能的过程中，收集支持系统功能目标的数据以及分析数据之间的关系和处理流程。

然后，就可以进行实质性的数据表规划工作了。一般来说，进行数据库的数据表规划是整个数据库设计中技巧性最强的一个步骤。因为此时虽然已经有了一些数据库的轮廓，但所搜集的数据只是反映了客户自己所关心的数据格式，而在数据库中只能存放数据库特有的数据格式。所以要把收集到的数据信息转化为与所采用的数据库管理系统所支持的数据模型相符合的数据形式。对于数据表的规划，一种行之有效的方法是对各个数据信息进行仔细研究，确定每种数据信息的实际内容，然后把具有同一个主题的数据规划到一个表中。不相干的信息不要放到同一个表中，以免造成数据库存储信息的冗余。

在确定了数据表后就可以对数据表中的字段进行规划，确定每个数据表中包含哪些字段。对于一个数据表来说，一般有一个主题，那么该数据表的字段就围绕这个主题来确定。

规划数据库要完成以下三个目标：

- (1) 消除包含在表中的冗余信息。
- (2) 适应表结构在未来发生的变化。
- (3) 在处理数据的前端应用程序的数据库结构发生变化时，使可能发生的冲突最小。

规划数据库有以下三条基本原则：

(1) 数据库中的记录个数应该反映现实世界对象的个数。如果一个对象的实例在现实世界中存在，则在数据库中有且只有一条记录存在。例如要储存客户的信息，则对应每一个客户，在数据库中都有有一条记录，且仅有一条。一条记录不能包括多个客户的信息，也不能将一个客户的信息分解在多条记录中存放。该原则是设计关系数据库的基础，可以使应用程序更有价值，因为它可以保证数据库的开放性和可访问性。

(2) 每行记录的字段应该表达现实世界对象的属性。

(3) 现实世界对象之间的关系应该反映在数据库记录之间的关系上。

规划数据库应该应用以下三个范式：

(1) 在表中没有重复的组。

(2) 表中非键字段都必须完全依赖于主键，如果主键是一个符号键，就必须依赖于主键中的每一个字段。

(3) 索引非键字段都必须依赖主键，且非键字段间彼此要独立。

1.2.2 设计一个关系数据库

前面学习了一些在设计数据库时用到的一些原则和方法，现在将结合一个实例来介绍如何设计一个数据库。一般设计一个数据库可分以下几个步骤进行：

1. 陈述问题

陈述问题看起来简单，实际做起来却很难。陈述问题要表现出对体系结构的理解，同时着重于商业需要。如果是替换现有的数据库，就要把现有的数据库作为解决问题的基础，如果是第一次建立数据库，就要理解所要解决的问题。如果连问题都提不出来，就谈不上解决问题。陈述问题应当尽可能简单。对于将要讨论的实例，可以这样描述问题：对某公司的销售员、客户的基本情况和订单情况进

行管理。

2. 分析问题

一旦对要解决的问题有了基本的了解，就可以建立数据表来存储相关的信息。对提出的问题进行分析，可以得到具体的信息。首先要管理客户的基本情况，就必须建立一个客户信息表，在这个数据表中存储客户的基本信息，如客户姓名、性别、客户编号、年龄、电话、地址等信息。其次要管理销售人员的基本资料，就要建立一个销售人员的信息表存储销售人员的基本信息，如销售员姓名、性别、编号、年龄、电话和地址等。最后，要对订单情况进行管理，还要建立一个订单信息表，记录公司的每一笔订单信息，如订单编号、订单金额、客户编号、销售员编号、订单日期等。

3. 建立实体和关系模型

实体是可以被惟一确定的，例如一个客户、一个销售员或者一份订单就是一个实体。每个实体都有一系列属性用来描述实体。例如每个客户都有名字、住址等属性。关系产生在两个实体之间，例如客户和订单的关系。通过前面的分析，可以建立客户、销售员和订单的实体模型如表 1-2、表 1-3 和表 1-4 所示。

表 1-2 客户信息实体模型

字段名	数据类型	长度	描述
Customer_id	Text	12	客户编号
Customer_name	Text	10	客户姓名
Customer_sex	Text	2	客户性别
Customer_age	Integer	2	客户年龄
Customer_phone	Text	12	客户联系电话
Customer_addr	Text	40	客户联系地址

表 1-3 销售员信息实体模型

字段名	数据类型	长度	描述
Saler_id	Text	12	销售员编号
Saler_name	Text	10	销售员姓名
Saler_sex	Text	2	销售员性别
Saler_age	Integer	2	销售员年龄
Saler_phone	Text	12	销售员联系电话
Saler_addr	Text	40	销售员联系地址

表 1-4 订单信息实体模型

字段名	数据类型	长度	描述
Order_id	Text	12	订单编号
Customer_id	Text	12	客户编号
Saler_id	Text	12	销售员编号
Order_amount	Currency	-	订单金额
Order_date	Date	8	订单日期

创建了实体模型之后，接下来是确定实体间的关系。前面说过有三种基本的关系：一对一、一对多和多对多。这些关系指一个实体中与另外实体中的实例相关的实例数目。在一对一关系中，一个实体中只有一个实例与另一个实体中的一个实例相关。在一对多关系中，一个实体中的实例与另一个实体中的多个实例相关。在多对多关系中，一个实体的多个实例和另一个实体的多个实例相关。例如客户信息实体中的一个客户实例可以和订单实体中的多个订单实例相关，销售员信息实体的一个销售员实例也可以和订单实体的多个实例相关。

4. 创建数据库

创建了实体模型后,创建数据库就是很简单的事情了。只要将每个实体转换成数据表,将实体的属性转换成数据表的字段,再将实体的每个实例转换成数据表的记录就可以了。在后面的介绍中,将创建这个数据库,在编程中将对这个数据库进行操作。

1.3 SQL 语言

关系数据库之所以成为当今数据库事实上的标准,除了具有简单的关系模型以外,更为重要的是,基于关系模型的结构化查询语言(SQL)为简单高效的操作数据库提供了强大的支持,这使得关系数据库深入人心。

SQL 已经成为数据库语言的标准,使数据库程序设计人员和数据库管理人员在创建和维护工作上进入了一个新的领域,并且随着客户机/服务器结构的应用程序的兴起,使 SQL 的概念延伸到了个人计算机。对于 Visual Basic 数据库编程来说,SQL 语言可以提供高效灵活的查询,大大弥补了 DAO 数据库编程的不足,使得编程简单有效。

SQL 其英文全称是 Structure Query Language,即结构化查询语言。SQL 语言最初是由 E.F.Codd 博士在 1970 年提出的。E.F.Codd 博士也是关系数据库模型的创立者,然而在他提出的关系数据库模型的理论中,只是对如何设计关系数据库提出了理论性的描述,并没有关于使用数据库的描述。对于创建表,指定表包含的字段,建立表间的关系以及对数据库中记录的添加、删除和修改等操作,需要一种数据库应用程序语言来实现。正是在这种背景下,由 Codd 博士所成立的实验室研究针对 IBM 的称为 System R 的一种原型关系数据库定义了一套结构化查询语言,称为 SEQUEL,这就是现在使用的 SQL 语言的前身。SEQUEL 的第二个版本是 SEQUEL/2,后来改名为 SQL。因此,SQL 的发音习惯上仍是 Sequel 的读音。

尽管 SQL 已有通用标准,然而对不同的系统,它们支持的 SQL 的功能也各有不同,语法也有差异。Visual Basic 使用的基本上是 Microsoft Jet 数据库引擎,因此它支持的 SQL 就是 Jet SQL 语法。尽管各种系统的 SQL 语言功能和使用语法各不相同,但是都是以 ANSI SQL-92 为基础的,因此其整体结构和功能都相似,所以掌握了一个系统的 SQL 语言,对于其他系统的 SQL 语言也可以触类旁通。

标准的 SQL 语言可以分为以下几类:

- (1) 数据查询语言 DQL。
- (2) 数据操作语言 DML。
- (3) 事务处理语言 TCL。
- (4) 数据控制语言 DCL。
- (5) 数据定义语言 DDL。
- (6) 游标定义语言 CCL。

在本书中,主要着重于 Visual Basic 中支持的数据查询语言(DQL)、数据定义语言(DDL)和数据操作语言(DML)。

1.3.1 SQL 的特点和基本功能

SQL 出现后,由于其突出的优点,迅速成为数据库语言的标准。SQL 的优点可以归纳为操作方式的改变、简单易用、兼容性和客户机/服务器结构的应用四点:

1. 操作方式的改变

以往的数据库语言总是针对一个记录或字段进行各种操作,这使数据库的维护受到很大限制并导致效率低下,而 SQL 则完全改变了操作的方式,摒弃了以往面向记录或字段的查询方式,转而面向整个表查询,甚至还可以通过建立表与表间的关系而进行表间的数据查询。这种方式的改变使程序员从使用程序控制数据库中一行行的记录的工作中解放出来,使用 SQL 语言,控制数据库进行排序、删除、