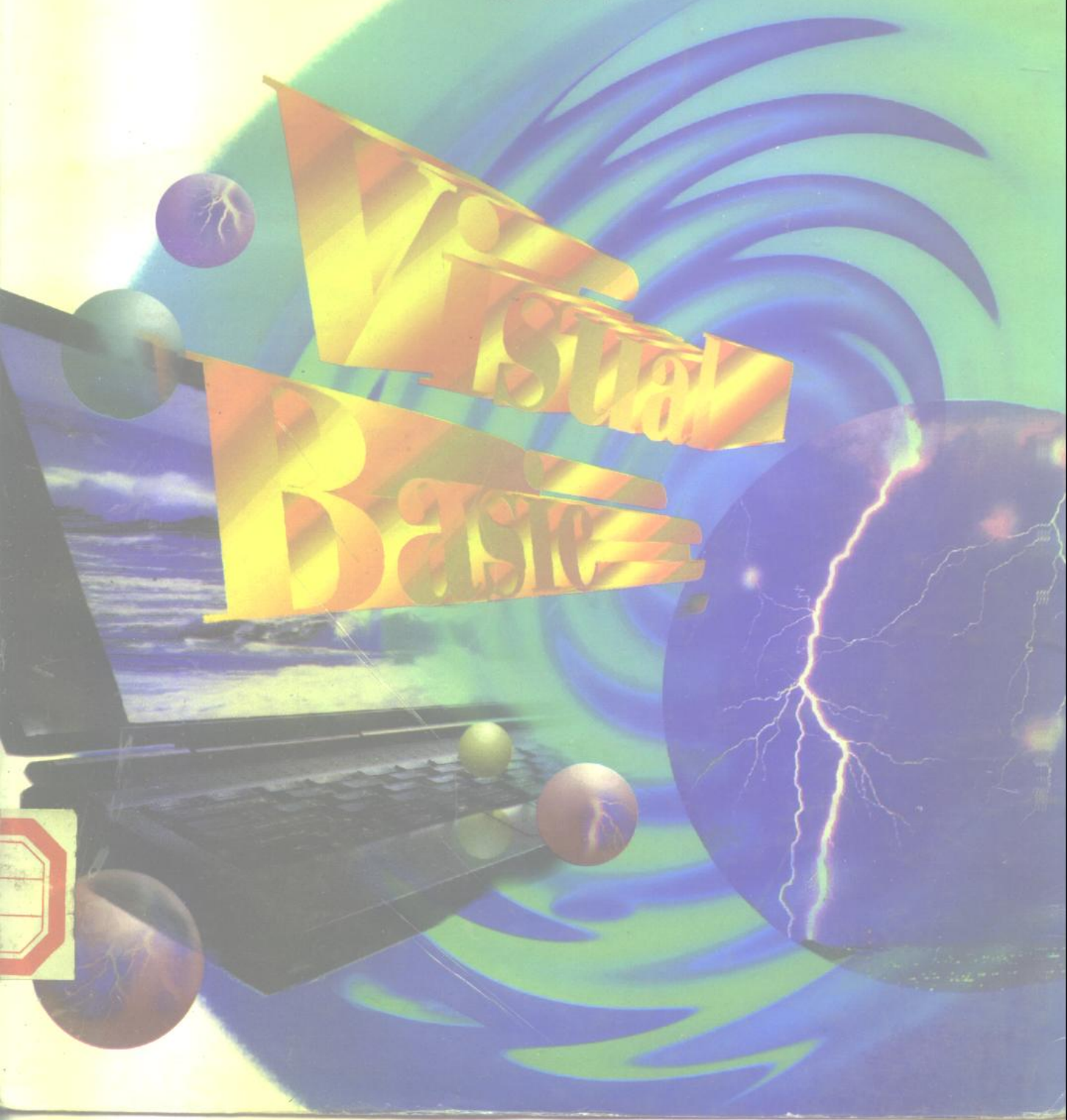


电脑用户丛书

Visual Basic 的 数据库开发应用

邓耘编

重庆出版社



100645

电 脑 用 户 丛 书

Visual Basic 的数据库开发应用

邓 耘 编

重 庆 出 版 社

1997·重庆

(川)新登字 010 号

责任编辑 王 梅
封面设计 邵大维
技术设计 刘忠凤

邓 耘 编

Visual Basic 的数据库开发应用

重庆出版社出版、发行 (重庆长江二路 205 号)
新华书店经销 四川外语学院印刷厂印刷

*

开本 787 × 1092 1/16 印张 13 插页 2 字数 297 千

1997 年 3 月第一版 1997 年 3 月第一版第一次印刷

印数: 1 - 2,000

*

ISBN 7-5366-3407-2/TP · 7

科技新书目 391 - 378

定价: 21.00 元

内 容 简 介

本书是一本专门介绍 Visual Basic 数据存取的书藉,详细讲述了 Visual Basic 中与数据库有关的内容。主要内容包括: Visual Basic 数据存取的构成、 Visual Basic 数据管理器的使用、数据控件和束缚控件用法、数据存取对象的属性和方法、 Visual Basic 使用的 SQL 语言,以及对 Visual Basic 提供的数据库样本程序的详细分析。

本书是使用 Visual Basic 开发 Access 、 Btrieve 、 dBASE 、 FoxPro 、 Paradox 以及 ODBC 等多种格式数据库的实用工具书,也可作为学习 Visual Basic 数据存取的教程使用。

序

数据库管理系统都是用高级语言编写而成的，但是数据库管理系统一旦编译、连接完成后，就与高级语言没有什么联系了。数据库管理系统自成一体，使用自己的命令和方式来实现对数据库的操作。为了开发数据库应用程序，多数数据库管理系统还提供了应用程序开发环境，以一种类似于高级语言的命令集来开发数据库应用程序。

对程序开发人员来说，这无疑是一种灾难。在开发数据库应用程序或基于数据库的其它应用程序时，首先面临的就是一个艰难的选择：是使用数据库管理系统提供的开发环境还是使用高级语言来开发应用程序？

如果您选择了数据库管理系统，也许是一个明智的选择。因为在您作出选择之际，您已拥有三样对开发数据库应用程序至关重要的东西：数据库逻辑模型（通常的数据库管理系统都为关系数据模型）、数据库物理模型（数据库存贮在物理设备上的方法）以及逻辑模型和物理模型的接口。它们提供给您一系列灵活完善的功能和手段来实现对数据库的处理。但是在您体验到这些优越之处以前，您必须花费大量的时间和精力去学习您所使用的数据库管理系统，包括命令的使用、编程方法以及几百个函数；接着您还会发现，数据库管理系统提供的编程环境与高级语言相比，不论是在数据结构、数据处理，还是在用户界面或编程方法上都还有一定的差距。

如果您选择了高级语言，您可以用您最熟悉的语言或市面上最优秀的语言来开发应用程序。您可以运用以往的经验，甚至把以前的程序嵌入新的应用程序中，特别是在屏幕、键盘、鼠标、打印机的控制，以及使用标准算法对数据进行处理等方面，使用高级语言具有绝对的优势，但是您必须自己编写数据库的物理模型，以及逻辑模型和物理模型的接口。从理论上讲，这是数据库厂商的事，实际上，数据库厂商做这个工作会比您要好。当您重新审视您作的努力时，您也许会感到，您所有的努力只是让您学会了一种如何用文件 I/O 来存取某种格式的数据库文件的方法。

高级语言与数据库管理系统的隔阂，降低了程序开发人员的能力，同时也阻碍了高级语言和数据库管理系统的发展。让高级语言和数据库管理系统相互融合，各尽其能，取长补短，不仅是程序开发人员的要求，也是高级语言和数据库管理系统发展的一个趋势。Visual Basic 的数据存取正是顺应了这个趋势。

Access 作为 Microsoft 公司的数据库产品, 是一个优秀的数据库管理系统。为了提高 Visual Basic 的数据处理能力, Microsoft 将 Access 使用的数据库引擎内嵌在 Visual Basic 中。当您拥有 Visual Basic 之际, 您不仅得到一个高级语言编程环境, 同时也得到一个数据库管理系统, 更重要的是它们是从内核层联系在一起的。

除了传统用的文件 I/O 方法, 通过顺序文件、随机文件和二进制文件来存取数据外, Visual Basic 还提供了以下数据存取功能:

- **数据管理器** 这是一个独立的 Windows 应用程序, 您可以在 Visual Basic 开发环境中或开发环境外使用。您可以使用数据管理器创建 Microsoft Access 1.0 或 1.1 格式的数据库, 或者创建 Btrieve、dBASE、FoxPro 和 Paradox 格式数据库的表, 对这些格式的数据库的结构进行维护。通过使用数据管理器, 您可以对 Access、Btrieve、dBASE、FoxPro、Paradox 格式数据库的数据进行存取。

- **数据控件和数据觉察控件** 数据控件用于在 Visual Basic 应用程序中打开数据库, 并与数据库建立连接关系; 数据觉察控件用于查看和修改数据库中的数据。通过使用数据控件和数据觉察控件, 您可以显示、编辑和更新多种格式的数据库中的数据。使用数据控件和数据觉察控件编写应用程序非常简便, 您甚至不必编写一行代码就可以建立一个查看、编辑已有数据库的标准 Windows 应用程序。

- **数据存取对象** Visual Basic 根据关系数据模型, 定义了一套数据存取对象, 在 Visual Basic 中完整地表述了关系数据模型。通过使用像 Database、Table、Field 这些数据存取对象, 您可以方便地编写数据库应用程序, 对数据库的结构进行维护, 对数据库的数据进行编辑。您可以用同样的数据存取对象, 以及相同的代码在 Windows 应用程序中创建和存取多种格式的数据库, 而不必关心这些数据库的物理模型有何不同。

- 对于 Visual Basic 不支持的数据库格式, 可以使用其它的数据库驱动程序, 而减少对源程序的修改。

- 允许使用第三方动态链接库 DDL 和用户开发的控件。

本书的重点放在 Visual Basic 数据存取部分, 有关 Visual Basic 的其它内容, 已有许多书籍作专门介绍, 请参阅。本书的安排如下:

第一章 Visual Basic 数据存取概述 全面介绍数据存取的功能。内容有关系数据库基本概念, 澄清库和表的概念; Visual Basic 如何利用面向对象方法对数据库进行封装, 定义的数据存取对象及其相互间的关系; 以及数据存取使用的语句或函数。本章为 Visual Basic 数据存取必读。

第二章 Visual Basic 数据管理器 一个可以对多种格式的数据库进行管理的实用程序。如何用 Visual Basic 数据管理器创建、修改和访问数据库是本章的主要内容。

第三章 使用数据控件存取数据库 数据控件是 Visual Basic 应用程序和数据库进行连接的重要方法之一。当您使用数据控件在 3 分钟内建立一个 Windows 程序、访问和修改一个数据库的内容, 您也许对 Visual Basic 的数据存取能力有些兴趣了。

第四章 访问样本数据库 BIBLIO 数据控件比 Visual Basic 的其它一些控件还要简单, 但这并不表示它用途有限, 相反它的用途可能要超出我们的想象。当您阅读完本章对访问 BIBLIO 的样本程序的分析后, 也许您会对数据库控件的用法有个新的认识。

第五章 数据存取对象 每个数据存取对象对应于数据库的一个概念, 如您可以定义

一个 Database 对象变量来表示一个数据库, 定义一个 Field 对象变量来表示一个字段。本章介绍了 Visual Basic 对象变量之间的关系, 详细描述了每一个数据存取对象的属性和方法。建议您第一次阅读本章时只需要大致浏览一下, 当您在编程中需要了解对象的某种属性或方法时, 再来详细阅读您所需要的内容。所有对象的属性和方法都按字母顺序进行排列, 并且都可以在目录中找到, 使用非常方便。

第六章 与数据库有关的函数或语句 这些函数和语句都是用于对数据库的操作。本章的阅读方法同上一章, 您只要记住每个函数或语句的用途就行了, 在您编程需要时再来详细学习它。

第七章 通用数据库管理程序 VISDATA 一个能管理多种格式数据库的 Windows 程序是如何用 Visual Basic 编制出来的, 您一定希望知道其中的奥秘。本章对 Visual Basic 的另一个样本程序进行了仔细的分析, 数据库编程常用的方法和技巧都尽在其中。

附录 A 结构化查询语言 SQL 对大多数人来说 SQL 都不是第一次听说, 但大多数人都很少使用 SQL。在几年前 SQL 还只是在许多高档的数据库管理系统中可见, 但今天大多数数据库管理系统都可以使用 SQL。附录 A 从实际应用的角度出发, 通过大量实例为您详细讲述了如何使用 SQL。花点时间学习 SQL 不仅对您编程大有益处, 您也可以掌握一门数据库管理系统的通用语言。

本书所讲述的内容都是在 Visual Basic 3.0 环境下实现的, 但仍然可以在 Visual Basic 4.0 环境下使用。因为在 4.0 版中, 数据存取仍然延续了 3.0 的体系, 只不过是更趋完善。

目 录

序.....	1
--------	---

第一章 Visual Basic 数据存取概述

1.1 Visual Basic 可以访问的数据库.....	1
1.2 Visual Basic 的数据存取.....	2
1.3 关系数据库的结构.....	2
1.4 数据库对象的封装.....	5
1.5 数据存取对象.....	6
1.5.1 Database 对象.....	6
1.5.2 Dynaset、Table、Snapshot 对象.....	7
1.5.3 Field 对象.....	8
1.5.4 Index 对象.....	9
1.5.5 TableDef 对象.....	9
1.5.6 QueryDef 对象.....	10
1.5.7 TableDefs、Fields 和 Indexes 对象.....	10
1.6 与数据存取有关的控件.....	10
1.6.1 数据控件.....	11
1.6.2 束缚控件.....	11
1.7 与数据存取有关的函数和语句.....	12
1.8 专业版和标准版的不同之处.....	12
1.9 样本数据库 BIBLIO.MDB.....	13

第二章 Visual Basic 数据管理器

2.1 Visual Basic 数据管理器的功能.....	15
2.2 新建数据库.....	16
2.3 打开已有数据库.....	16
2.4 创建非 Microsoft Access 格式的数据库.....	18
2.5 创建新表.....	18
2.6 删除旧表.....	20
2.7 向表中添加字段.....	21
2.8 修改表中的数据.....	22
2.8.1 将记录添加到表中.....	22
2.8.2 从表中删除记录.....	22
2.8.3 修改记录中的数据.....	23
2.8.4 查看记录的最新数据.....	23
2.9 修复数据库.....	23
2.10 压缩数据库.....	24

第三章 使用数据控件存取数据库

3.1	快速入门	25
3.2	设置数据控件的属性	30
3.2.1	Connect 属性	31
3.2.2	DatabaseName 属性	31
3.2.3	EditMode 属性	32
3.2.4	Exclusive 属性	33
3.2.5	Option 属性	33
3.2.6	ReadOnly 属性	34
3.2.7	Recordset 属性	34
3.2.8	RecordSource 属性	34
3.3	使用束缚控件	35
3.3.1	束缚控件的类型	35
3.3.2	把一个束缚控件添加到窗体中	36
3.3.3	在程序运行时使用束缚控件	36
3.3.4	使用图片和图像束缚控件	37
3.4	数据控件的事件	37
3.4.1	Error 事件	37
3.4.2	Reposition 事件	38
3.4.3	Validate 事件	39
3.5	数据控件的方法	41
3.5.1	Refresh 方法	41
3.5.2	UpdateControls 方法	43
3.5.3	UpdateRecord 方法	43
3.6	通过代码对记录进行维护	43
3.6.1	管理当前记录	44
3.6.2	改变当前记录	45
3.6.3	添加、编辑和删除记录	47
3.6.4	更新记录集	50
3.7	关闭记录集	51
3.8	处理长数据字段	51
3.9	事务	52
3.9.1	开始一个事务	52
3.9.2	事务的提交	52
3.9.3	事务的回退	52

第四章 访问样本数据库 BIBLIO

4.1	样本程序的组成	54
4.2	BIBLIO 的窗体设计	55
4.3	程序的运行	58
4.3.1	主窗体的装载	58
4.3.2	Titles 表的显示	58

4.3.3	Authors 表的显示	61
4.3.4	Publishers 表的显示	63
4.3.5	Comments 表的显示	65
4.3.6	选择要显示的表	65
4.3.7	同时显示多个表	67
4.3.8	数据的锁定	69

第五章 数据存取对象

5.1	Database 数据库对象	72
5.1.1	Database 数据库对象的属性	72
5.1.2	Database 对象的方法	75
5.2	Dynaset、Table 和 Snapshot 记录集对象	87
5.2.1	记录集变量的属性	88
5.2.2	记录集变量的方法	96
5.3	Field 对象	105
5.3.1	Field 对象的属性	106
5.3.2	Field 对象的方法	112
5.4	Index 对象	116
5.4.1	Fields 属性	116
5.4.2	Name 属性	117
5.4.3	Primary 属性	117
5.4.4	Unique 属性	118
5.5	TableDef 对象	119
5.5.1	Attributes 属性	120
5.5.2	Connect 属性	121
5.5.3	DateCreated、LastUpdated 属性	122
5.5.4	Name 属性	122
5.5.5	SourceTableName 属性	122
5.5.6	Updatable 属性	123
5.6	QueryDef 对象	123
5.6.1	QueryDef 对象的属性	124
5.6.2	QueryDef 对象的方法	125
5.7	TableDefs、Fields 和 Indexes 对象集合	130
5.7.1	TableDefs 集合	130
5.7.2	Fields 集合对象	131
5.7.3	Indexes 集合	132
5.7.4	TableDefs、Fields 和 Indexes 的属性和方法	132

第六章 与数据库有关的函数或语句

6.1	BeginTrans、CommitTrans、Rollback 语句	135
6.2	CompactDatabase 语句	136
6.3	CreateDatabase 函数（专业版）	139

6.4	FreeLocks 语句.....	140
6.5	OpenDatabase 函数 (专业版)	141
6.6	RegisterDatabase 语句	142
6.7	RepairDatabase 语句.....	142
6.8	SetDataAccessOption 语句.....	144
6.9	SetDefaultWorkspace 语句.....	145

第七章 通用数据库管理程序 VISDATA

7.1	窗体的装载	148
7.2	打开数据库	152
7.2.1	打开非 ODBC 数据库	152
7.2.2	打开 ODBC 数据库	157
7.3	创建新数据库	158
7.3.1	创建 Access 数据库	158
7.3.2	创建外部数据库	159
7.3.3	创建 ODBC 数据库	159
7.4	数据库结构设计	160
7.4.1	字段的修改	160
7.4.2	索引	161
7.4.3	表的添加	162
7.5	记录的显示	165
7.5.1	使用数据控件显示记录	166
7.5.2	不使用数据控件显示记录	167
7.5.3	使用网格窗口显示记录	168

附录 A 结构化查询语言 SQL

A.1	SELECT 语句	173
A.1.1	无条件 SELECT 语句	173
A.1.2	条件 SELECT 语句	176
A.1.3	选取结果的排序	179
A.1.4	别名和算术式	180
A.1.5	聚集函数	181
A.1.6	分组查询	182
A.2	表的连接操作	183
A.2.1	内部连接	185
A.2.2	外部连接	186
A.2.3	自身连接	187
A.3	数据的修改	188
A.3.1	插入	188
A.3.2	更新	188
A.3.3	删除	190
A.4	Microsoft Access SQL 保留字	190

A.5 数据类型的对应关系	191
A.6 Microsoft Access SQL 与 ANSI SQL 的区别.....	192
A.6.1 Microsoft Access SQL 的增强	192
A.6.2 Microsoft Access SQL 中不支持的 ANSI SQL 特性	192

第一章 Visual Basic 数据存取概述

Visual Basic 不仅是一个快速高效开发 Windows 应用程序的工具, 同时也是一个开发数据库应用程序的优秀工具。

大多数计算机应用程序, 其核心功能都是对数据的处理。但数据库管理系统和高级语言之间的隔阂由来已久, 给计算机开发人员带来了诸多不便。Visual Basic 的出现, 打破了数据库管理系统和高级语言之间的隔阂, 从此, 在高级语言中我们也可以像在数据库管理系统中一样, 对数据库进行管理。

Visual Basic 通过把 Microsoft Access 使用的数据库引擎内置在 Visual Basic 环境中, 实现了 Visual Basic 的数据存取功能。您可以在应用程序中使用 Visual Basic 的数据存取功能, 访问其它像 Access、FoxPro、dBase 等大多数流行的数据库, 从而使您的程序具有更强的数据处理能力, 而不仅仅限于文件 I/O。

1.1 Visual Basic 可以访问的数据库

在 Visual Basic 应用程序中, 可以对多种格式的数据库进行访问和维护。例如, 您可以在 Visual Basic 应用程序中创建一个 Access 数据库, 通过使用 Microsoft Access 来维护这个数据库; 您也可以使用 Microsoft FoxPro 建立一个数据库, 然后在您的 Visual Basic 应用程序中访问和修改这个数据库的内容。

为了在编程上的方便, 通常把 Visual Basic 可以存取的数据库分为三类:

1. Microsoft Access 格式的数据库 这类数据库可以使用 Visual Basic 或 Microsoft Access 直接进行维护, 这类数据库可以由 Visual Basic 或 Microsoft Access 建立。这种格式的数据库称为 Visual Basic 的自有格式数据库。

2. 外部数据库 外部数据库包括以下格式的数据库: Btrieve、dBASE III、dBASE IV、Microsoft FoxPro 2.0 和 2.5 以及 Paradox。

3. 外部 ODBC 数据库 开放式数据库互连 (Open-DataBase-Connectivity, 即 ODBC), 是 Microsoft 为不同的数据库产品的互相访问提出的解决方案。现在常见的 DBMS

都提供了 ODBC 的驱动接口。Visual Basic 可以访问的外部 ODBC 数据库包括客户机/服务器数据库, 如 Microsoft SQL Server 和 Oracle 数据库管理系统。当访问 ODBC 外部数据库时, 您可以把 SQL 命令直接提交到外部服务器执行。

访问外部 ODBC 格式的数据库比前两种格式的数据库要复杂一些, 需要对数据库文件结构、数据存取方式以及服务器等内容有比较深入细致的了解。

1.2 Visual Basic 的数据存取

在 Visual Basic 中, 把所有与数据库有关的操作统称为数据存取。为了开发数据存取应用程序, Visual Basic 提供了以下功能:

- **数据管理器** 这是一个独立的 Windows 应用程序, 您可以在 Visual Basic 开发环境中或开发环境外使用。您可以使用数据管理器创建 Microsoft Access 1.0 或 1.1 格式的数据库, 或者创建 Btrieve、dBASE、FoxPro 和 Paradox 格式数据库的表, 对这些格式的数据库的结构进行维护。通过使用数据管理器, 您可以对 Access、Btrieve、dBASE、FoxPro、Paradox 格式数据库的数据进行存取。

- **数据控件和数据觉察控件** 数据控件用于在 Visual Basic 应用程序中打开数据库, 并与数据库建立连接关系; 数据觉察控件用于查看和修改数据库中的数据。通过使用数据控件和数据觉察控件, 您可以显示、编辑和更新多种格式的数据库中的数据。使用数据控件和数据觉察控件编写应用程序非常简便, 您甚至不必编写一行代码就可以建立一个查看、编辑已有数据库的标准 Windows 应用程序。

- **数据存取对象** Visual Basic 根据关系数据模型, 定义了一套数据存取对象, 在 Visual Basic 中完整地表述了关系数据模型。通过使用像 Database、Table、Field 这些数据存取对象, 您可以方便地编写数据库应用程序, 对数据库的结构进行维护, 对数据库的数据进行编辑。您可以用同样的数据存取对象, 以及相同的代码在 Windows 应用程序中创建和存取多种格式的数据库, 而不必关心这些数据库的物理模型有何不同。

- 对于 Visual Basic 不支持的数据库格式, 可以使用其它的数据库驱动程序, 而减少对源程序的修改。

- 允许使用第三方动态链接库 DDL 和用户开发的控件。

1.3 关系数据库的结构

Visual Basic 可以存取多种格式的数据库, 虽然 Access、Btrieve、dBASE、FoxPro、Paradox 和 ODBC 产品 (如 SQL Server、Oracle) 等数据库管理系统各具特色, 但它们都有一个相同的特点, 即都是关系数据库, 使用的都是关系数据模型。

数据模型是数据库管理系统的基本结构, 它体现了数据库中数据间的相互联系, 数据模型的好坏直接决定数据库管理系统的质量。数据库管理系统使用的数据模型大致可以分

为三类：层次数据模型、网状数据模型和关系数据模型。

层次数据模型和网状数据模型在 20 世纪 60 ~ 70 年代使用较多,但由于它们表示数据之间的联系过于复杂,数据库的可操作性太差,因此现在使用较少。关系数据模型在 1970 年由 Ted Code 提出,它以关系的数学理论为基础,具有简单严谨的特色,在过去 20 多年中得到了迅速的发展,现在的数据库产品几乎都是以关系数据模型为基础的。

关系数据模型的“关系”,是一个从数学中来的概念。在关系代数中,“关系”是指二维的表,表不仅可以用来表示数据,同时也表示了数据之间的联系。Ted Code 在提出关系数据模型时,是把“关系”作为表同义词使用的。

在关系数据模型中,使用二维表(即关系)来表示数据及数据间的联系。在我们的日常工作中表被大量地使用,月报表、统计表、状况表、调查表等等,我们对表都很熟悉。表具有简单、直观、方便的特点。下面是一个读者状况表,它反映了读者的情况:

表 1-1 读者状况表——Reader

rno	rname	unit
11	张小林	东方电脑公司
18	王军	迪亚软件所
36	李建涛	数据处理公司
15	赵国明	东方电脑公司
10	王军	《软件开发》杂志社

从读者状况表我们可以看出,表具有以下特征:

1. 一个表有一个或多个列。
2. 各列从左到右的顺序不影响表的内容。
3. 一个表可有零个或多个行。
4. 行的顺序不影响表的内容。
5. 一个表的每一行都是唯一的。
6. 任何行和列的交叉点都有一个唯一的值。
7. 同一列的所有数值都具有相同的类型。

在关系数据模型中,表中的列称为字段,每个表所能容纳的字段的数量,在具体的数据库产品中有所不同。例如,在 dBASE III 中,每个表中最多的字段数量不能超过 128。

表中的每个字段的取值都有一定的类型和变化范围,如在读者状况表中,读者编号为数字型,读者姓名为字符类型。字段的类型和变化范围根据使用的数据库管理系统有所不同。在大多数数据库管理系统中,都有整型字段、浮点型字段、字符型字段。

每个表都有一个结构,称为表结构(关系结构),表结构由表名和表中使用的字段的名称构成。在上面的例子中,表的名称为“Reader”,表中使用的字段为“rno”、“rname”和“unit”,分别表示读者的编号、读者姓名和读者的工作单位。

一个表的结构确定了表的二维框架,表是这结构的具体体现,每个表都有一个结构,

表按照结构的要求来存贮数据。

关系数据库使用表（关系）来存贮数据和表示数据之间的类型。在一个数据库中，可以有一个或多个表，所有的表的集合就构成了关系数据库。构成数据库的所有表的结构就构成了关系数据库的结构。

关系数据库的结构可以用图 1-1 来表示：

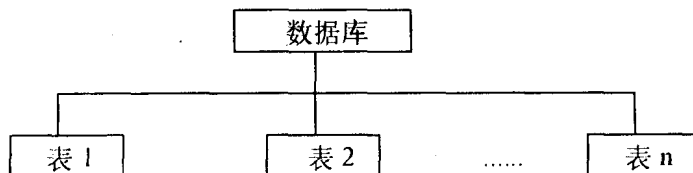


图 1-1 关系数据库的结构

其中，表又可以表示为：

	字段 1	字段 2		字段 n
记录 1	数据 11	数据 21		数据 n1
记录 2	数据 12			
.....				
记录 n	数据 1n			

在大多数数据库管理系统中，一个表就是一个文件。例如，在 dBASE 和 FoxPro 中，一个 DBF 文件就是一个表，所有的 DBF 文件的集合就构成了数据库。这种以文件存贮表的方式在使用上有许多不便之处，一个数据库可能有多个表，这些表分别存放在不同的文件之中，管理比较困难，在使用数据库时，可以打开的表的数量受到操作系统打开文件数量的限制。另外，在数据库只有一个表时，使用上容易形成概念上的误解，把表名与数据库名等同起来，认为库就是表，表就是库。最近，在一些数据库管理系统中，也引进了库文件的概念，如在 FoxPro 3.0 中，除了传统的 DBF 文件外，也增加了 DBC 文件，用于表示库文件。

有些数据库管理系统，如 Microsoft Access、Btrieve，所有的表都存贮在一个文件之中，表不再以单独的文件存在，这个文件也就是库文件。例如，Microsoft Access 的库文件扩展名为 MDB，Btrieve 的库文件扩展名为 DDF，库文件名也即是数据库的名称。

在使用 Visual Basic 对各种格式的数据库进行存取时，为了便于理解和编程，把各种格式的数据库的名称规定为：

- 对 Microsoft Access 和 Btrieve 格式的数据库，数据库名是指库文件的名称，即缺省扩展名为 MDB 和 DDF 的文件。

例如，您使用 Microsoft Access 或 Visual Basic 建立了一个 Access 格式数据库，保存在

MYDB.MDB 文件中, MYDB 就是数据库的名称。MYDB 数据库中的表在操作系统下是不可见的, 只有使用数据库管理系统打开 MYDB 数据库后, 才能知道在 MYDB 数据库中有几个表。

- 对其它格式的数据库, 数据库名是指存放数据库表文件的目录的名称。

例如, 您使用 dBASE 或 FoxPro 建立了一个数据库, 定义了两个表, 分别为 READER.DBF 和 BOOK.DBF, 这两个表放在 C:\USER 目录下, 那么数据库的名称就应该是 USER, READER.DBF 和 BOOK.DBF 只是 USER 数据库下面两个表的名称。

这两个规定对使用 Visual Basic 编写数据库应用程序非常重要, 在对数据库进行存取时一定要分清楚库是指什么, 表是什么, 否则在打开一个数据库时, 您把一个表名 (如 DBF 文件) 当库名使用, 当再要求您指定要使用的表时, 您将会感到迷惑, 不知道表在什么地方。

1.4 数据库对象的封装

虽然 Visual Basic 并不是一个完全的面向对象程序设计 (OOP) 语言 (至少在 Visual Basic 3.0 版中如此), 但它确实使用了 OOP 的几个关键概念。使用 Visual Basic 并不需要了解 OOP 的全部内容, 但它却使我们的编程方法产生了一个质的飞跃, 从过程驱动进入到事件驱动。

在现实中, 对象代表一个实体, 汽车可以是一个对象, 树也可以是一个对象。每一个对象都有自己的特征和行为, 汽车有轮子、发动机, 可以行走和转弯; 树有树根、树干、树叶, 给树浇水可以使树长大。

从程序设计的角度来看, 对象 (在 OOP 中通常称为类) 是一个客观实体的抽象概念, 对象有属性, 它代表了实体的特性; 对象有方法, 它表示实体的行为。把对象的特征和行为集中到一起, 这样比分散开来更安全和方便。当您定义一个对象表示汽车后, 您就不可能编写给汽车浇水让汽车长大的代码, 您也不会编写转动方向盘让树木转弯的代码。而这种类型的错误, 在非 OOP 程序设计时, 一不小心就会出现。

这种把属性 (表示特征的数据) 和方法 (数据处理的操作) 放在一起的作法被称为封装, 它是面向对象的程序设计的最常用的一个概念。在一般的 OOP 程序中, 您可以自己定义新的对象类型, 但是在 Visual Basic 中, 所有的对象类型已经预先定义好了, 您不能再定义新的对象类型, 只能使用 Visual Basic 定义的对象类型。

在 Visual Basic 中窗体 (Form)、打印机 (Printer) 等都是预先定义的对象。Printer 就有 FontName、FontSize、Height、Weight 等属性, 有 Print、Circle、Line 等方法。属性定义了打印机的状态, 如使用的字体、字体的大小、打印的范围等内容, 而 Print、Circle、Line 则表示打印一串文本、画一个圆、画一条线等动作。

Visual Basic 把数据库封装成许多数据存取对象, 当您需要对数据库进行操作时, 需要定义相应的数据存取对象变量, 设置它们的属性, 通过使用它们的方法来实现对数据库的操作。