数据库高级开发与专业应用系列

Visual Basic 6.0 数据库开发与专业应用

敬　铮　主编
于英杰　编著

国防工业出版社
·北京·

需要全本请访问 www.ertongbook.com

内容简介

Visual Basic 6.0 中文版是 Microsoft 公司最新推出的可视化开发工具，是 Visual Basic 语言的最新版本，它比以前版本增加了许多新的功能和特点。本书以通俗、简明的语言由浅入深地讲解了如何使用 Visual Basic 6.0 开发数据库应用程序。同其他介绍 Visual Basic 的书籍相比较，本书更加全面、系统地介绍了应用 Visual Basic 6.0 访问各种数据库的方法和实际的编程技巧。

本书内容详实、图文并茂、实用性强，能够对数据库编程爱好者提供一定的帮助。

前　言

Visual Basic 6.0 中文版是 Microsoft 公司最新推出的 Windows 应用程序开发工具，是 Microsoft Visual Studio 6.0 系列开发产品之一，它提供了开发 Windows 应用程序最迅速、最简捷的方法。

从目前开发的应用软件的使用范围来看，数据库技术往往在选择开发工具时占据重要的因素。Visual Basic 6.0 给我们提供了用于从数据库访问数据的强大工具集，使用它既可以建立对本地数据库操作的应用程序，也可以建立适应于网络中的分布式数据处理的应用，建立与大型数据库服务器的远程连接。

本书采用最新的 Visual Basic 6.0 中文版，全面介绍了应用 Visual Basic 6.0 进行数据库开发的应用技术与技巧。

本书共分为 13 章。第 1 章介绍 Visual Basic 6.0 开发数据库应用程序的优点以及开发方案；第 2 章介绍如何建立与打开数据库；第 3 章介绍应用中的数据访问；第 4 章：介绍数据环境设计器；第 5 章：介绍 Data 控件和数据绑定控件；第 6 章介绍 ADO、DAO 和 RDO；第 7 章管理 Microsoft Jet 数据库；第 8 章介绍如何建立和管理数据报表；第 9 章开发 DHTML 应用；第 10 章介绍如何建立 ODBC 连接；第 11 章介绍如何访问 SQL Server 数据库；第 12 章介绍多客户环境下 Visual Basic 编程；第 13 章介绍数据库安全方面的有关知识。

本书在内容安排上，本着详实、简练的原则，内容合理、语言通俗、举例丰富，适合对数据库编程方面感兴趣的读者学习。在学习过程中，如果能够结合上机实践，会收到更好的结果。

全书由敬铮主编，于英杰编著，邹忠望、刘朋、阎永军、刘敏、耿永兵、徐宏龙、李明、邓刚、王筑军、邴安大、胡小超、姚俊波、谢石、王小伟、颜林灿等同志参与了部分章节内容的编写和程序调试工作。由于时间仓促，作者水平有限，恳请广大专家读者不吝赐教。

作　者

目　　录

第 1 章　综述 1
1.1　Visual Basic 6.0 的新增功能 1
1.1.1　数据访问 1
1.1.2　控件 2
1.1.3　向导 2
1.1.4　语言 3
1.1.5　Internet 描述 4
1.2　数据库的分类 6
1.2.1　层次数据库 6
1.2.2　网状数据库 8
1.2.3　关系数据库 10
1.3　客户/服务器模式的数据库 11
1.3.1　集中的数据库 11
1.3.2　分布式处理 11
1.4　Visual Basic 6.0 可以访问的数据库 11
1.5　Visual Basic 6.0 的数据库源 12
1.6　数据环境 12
1.7　ADO、DAO 和 RDO 12
1.8　为什么采用 Visual Basic 6.0 开发数据库 13
1.8.1　用 Visual Basic 6.0 作为数据库开发平台 13
1.8.2　Visual Basic 6.0 作为数据库前端 13
1.9　普通的关系数据库管理系统 15
1.9.1　客户/服务器数据库管理系统 15
1.9.2　Microsoft 数据库管理系统 Access 16
1.10　数据库的结构 16
1.11　简单的浏览数据库的程序 17
1.11.1　应用程序(使用数据控件) 18
1.11.2　一点说明 18
1.12　Visual Basic 访问数据库的方案比较 19
1.13　小结 22
第 2 章　建立与打开数据库 23
2.1　数据库的定义 23
2.2　建立数据库 23

2.2.1 建立 Jet 数据库 23
2.2.2 建立 ISAM 数据库 27
2.2.3 建立 ODBC 数据库 28
2.3 打开数据库 30
2.3.1 打开 Jet 数据库 30
2.3.2 打开 ISAM 数据库 31
2.4 打开 ODBC 数据库 31
2.5 数据表操作 32
2.5.1 建立数据表 32
2.5.2 修改数据表结构 35
2.5.3 记录操作 36
2.6 构造不定结构的数据库 38
2.7 小结 41
第 3 章 应用中的数据访问 42
3.1 使用实际程序 42
3.1.1 查询生成器 42
3.1.2 SQL 语句 45
3.1.3 数据窗体设计器 47
3.2 数据窗体向导 49
3.2.1 使用数据窗体向导 49
3.2.2 数据窗体布局介绍 57
3.3 小结 60
第 4 章 数据环境设计器 61
4.1 添加数据环境对象 61
4.2 Connection 对象 62
4.2.1 添加 Connection 对象 63
4.2.2 指定 Connection 对象属性 63
4.2.3 设置登录信息 64
4.2.4 设置连接信息 65
4.3 Command 对象 65
4.3.1 创建一个 Command 对象 66
4.3.2 从存储过程中创建 Command 对象 69
4.3.3 Command 层次结构 70
4.3.4 字段映射 75
4.3.5 在应用中使用数据环境 77
4.3.6 使用数据环境设计器来创建 ActiveX 数据对象 81
4.4 小结 83
第 5 章 Data 控件和数据绑定控件 85
5.1 Data 控件 85
5.1.1 Data 控件能完成的操作 85

需要全本请访问 www.ertongbook.com

5.1.2 Data 控件的属性 85
5.1.3 Data 控件的事件 85
5.2 使用 Data 控件访问数据 86
5.2.1 使用 Data 控件创建一个简单的数据库应用程序 87
5.2.2 使用 DataCombo 和 DataList 控件 87
5.2.3 创建 Northwind 的 OLE DB 数据链接 90
5.2.4 使用 DataGrid 和 ADO Data 控件 90
5.2.5 使用 DataGrid 控件 94
5.3 数据绑定控件 96
5.3.1 DBListBox 和 DBComboBox 控件 97
5.3.2 DBGrid 控件 97
5.4 Data 控件和 DAO 对象编程 97
5.4.1 Recordset 对象的类型 97
5.4.2 Recordset 对象 98
5.4.3 DAO 对象 99
5.5 小结 101
第 6 章 ADO、DAO 和 RDO 103
6.1 数据访问对象（DAO）及 DAO 数据控件 103
6.1.1 数据访问对象（DAO） 103
6.1.2 DAO 数据控件 104
6.2 远程数据对象(RDO)及远程数据控件 104
6.2.1 远程数据对象(RDO) 104
6.2.2 远程数据控件 105
6.3 ActiveX 数据对象(ADO)及 ADO 数据控件 105
6.3.1 ActiveX 数据对象(ADO) 105
6.3.2 ADO 数据控件 106
6.3.3 如何使用数据—约束控件 107
6.4 使用 ActiveX 数据对象（ADO）的技巧 112
6.4.1 如何使用 ADO 压缩或修复 Microsoft Access 文件 112
6.4.2 Visual Basic 中超长 OLE 数据库字段的操纵方法 113
6.4.3 如何使 Data 控件显示的记录与 OLE 字段的内容同步 117
6.5 用 FlexGrid 控件来操纵数据 118
6.5.1 将 FlexGrid 用作通用电子数据表格 118
6.5.2 用 FlexGrid 控件显示数据库记录 124
6.6 小结 129
第 7 章 管理 Microsoft Jet 数据库 131
7.1 Microsoft Access 131
7.2 表的定义及作用 134
7.3 如何建立一个数据库 135
7.4 建立定制数据库应用程序 139

7.5 使用绑定控件显示数据库信息 139
7.6 使用文本框对象显示数据 140
7.7 运行 Courses 程序 141
7.8 使用 Recordset 对象 142
7.8.1 在 Students.mdb 中查找数据 143
7.8.2 向 Students.mdb 数据库添加记录 145
7.8.3 从 Students.mdb 数据库中删除记录 148
7.9 在 Visual Basic 6.0 中使 Access 数据库在网络使用中保持同步 150
7.9.1 数据库同步复制功能 150
7.9.2 基本过程 152
7.9.3 属性与方法 152
7.10 用 Visual Basic 6.0 访问具有用户级安全的 Access 97 数据库 154
7.10.1 访问用户级安全的数据库 155
7.10.2 用 Access 97 创建一个用户级安全的数据库 155
7.10.3 访问此用户级安全的数据库 156
7.11 制作文件备份 159
7.11.1 使用 FileCopy 制作 Students.mdb 的备份 159
7.11.2 运行 MyBackup 程序 160
7.12 Visual Basic 6.0 管理 Access 数据库快速参考 160
7.13 链接 FoxPro 数据库 162
7.14 打开 FoxPro 表 162
7.15 处理 FoxPro 索引 163
7.16 处理备注字段 164
7.17 指定连接信息 164
7.18 方案描述信息文件 165
7.19 程序举例 168
7.20 小结 170
第 8 章 数据报表 173
8.1 建立一个数据报表 173
8.1.1 创建一个简单的数据报表 173
8.1.2 合计字段 177
8.1.3 Function 控件和合计字段 178
8.1.4 分页 178
8.1.5 添加日期、时间、页号和标题 179
8.2 导出报表 180
8.3 报表打印 181
8.3.1 PrintReport 方法 181
8.3.2 举例 182
8.4 小结 182
第 9 章 开发 DHTML 应用 184

9.1 DHTML 应用概述 184
9.1.1 DHTML 应用的概念 184
9.1.2 DHTML 应用程序的结构 184
9.1.3 DHTML 应用程序的优点 185
9.1.4 DHTML 页对象的事件 185
9.1.5 设计 DHTML 应用的注意事项 186
9.1.6 DHTML 应用程序的设计步骤 187
9.2 创建 DHTML 应用 187
9.2.1 工具箱成员 187
9.2.2 新建 DHTML 应用程序工程 190
9.2.3 页面设计 192
9.2.4 添加 DHTML 页面 195
9.2.5 设计代码并运行应用 198
9.3 DHTML 应用中的数据交互 202
9.3.1 创建数据输入 HTML 页面 202
9.3.2 创建查看和更新数据 HTML 页面 205
9.4 小结 207
第 10 章 建立 ODBC 连接 209
10.1 ODBC 简介 209
10.1.1 什么是 DBMS 209
10.1.2 DBMS 与操作系统 210
10.1.3 为什么要创建 ODBC 212
10.1.4 ODBC 是什么 213
10.1.5 ODBC 和标准 CLI 213
10.1.6 ODBC 总体结构 214
10.2 结构化查询语言（SQL） 223
10.2.1 SQL 概貌及其特点 223
10.2.2 SQL 的三种技术 225
10.2.3 处理 SQL 语句 225
10.3 SQL 数据定义功能 226
10.3.1 基本表的定义及修改 227
10.3.2 索引的建立及删除 227
10.3.3 视图 228
10.4 SQL 数据操纵功能 230
10.4.1 SQL 查询语句 230
10.4.2 SQL 更新语句 230
10.4.3 SQL 数据控制功能 232
10.5 嵌入式 SQL 232
10.5.1 嵌入式 SQL 的中心思想及技术 233
10.5.2 嵌入式 SQL 实例 233

10.5.3 编译嵌入式 SQL 程序 234
10.6 静态 SQL 236
10.7 动态 SQL 236
10.8 SQL 模块 237
10.9 调用层接口 238
10.10 数据库访问结构 238
10.10.1 网络数据库访问 238
10.10.2 标准数据库访问结构 239
10.11 ODBC 解决方案 240
10.12 Visual Basic 6.0 的 API 编程精粹 242
10.12.1 API 基础 242
10.12.2 两个实际应用的例子 243
10.13 专业化应用 245
10.14 直接使用 ODBC API 访问数据库的 Visual Basic 程序 247
10.14.1 ODBC API 函数的声明方法 248
10.14.2 使用 ODBC API 的编程方法 248
10.14.3 简单应用实例 250
10.15 实 用 技 巧 252
10.15.1 处理 Select 语句中的单引号 253
10.15.2 非 DSN 的 ODBC 连接 253
10.15.3 建立和删除一个 DSN 253
10.16 小结 254
第 11 章 访问 SQL Server 数据库 257
11.1 SQL Server 简介 257
11.1.1 理解 SQL Server 的安全模式 257
11.1.2 创建用户 257
11.1.3 用户权限的设定 258
11.1.4 备份数据 258
11.2 用 Visual Basic 访问 SQL Server 258
11.2.1 Visual Basic 与 SQL Server 的三种常见接口 259
11.2.2 数据访问对象 259
11.2.3 什么是 RDO 259
11.2.4 RDO 的对象 260
11.2.5 用 RDO 与 SQL Server 连接和断开 261
11.3 SQL Server 数据对象结构的动态建立与访问 262
11.4 小结 264
第 12 章 多客户环境下 Visual Basic 数据库编程 266
12.1 Microsoft Jet 的多客户模型 266
12.1.1 多客户环境下的数据库存储方式 266
12.1.2 静态部件与动态部件 266

12.1.3 Microsoft Jet 多用户模型 267
12.2 数据访问控制 268
12.2.1 互斥地使用数据库 268
12.2.2 拒绝其他用户或程序访问指定的表 269
12.3 共享数据的锁定 269
12.3.1 共享模式与独占模式 270
12.3.2 记录集锁定 272
12.3.3 页面锁定 273
12.4 多用户应用程序中应注意的其他问题 280
12.4.1 数据访问对象集合的刷新 280
12.4.2 再查询与事务 281
12.4.3 用户标识 283
12.5 小结 284
第 13 章 数据库安全 286
13.1 数据库安全与保密 286
13.1.1 数据库安全与保密概述 286
13.1.2 数据库基本安全架构 286
13.1.3 数据库加密 287
13.2 异构平台的数据库安全技术 290
13.2.1 安全策略 290
13.2.2 安全代理模型 290
13.2.3 DM3 的安全体系结构 291
13.2.4 三权分立的安全机制 291
13.2.5 自主访问与强制访问控制 291
13.2.6 隐通道分析技术 292
13.3 应用：在 Visual Basic 中建立处理 Microsoft Access 数据库安全机制 292
13.3.1 Microsoft Access 数据库安全机制 293
13.3.2 如何设置安全机制 293
13.3.3 使用 Visual Basic 编程实现 Jet 安全功能 295
13.3.4 Microsoft Access 数据库的加密与解密 297
13.4 一种简便有效的数据库安全法 298
13.5 数据库恢复 300
13.5.1 故障的种类 300
13.5.2 转储与恢复 302
13.5.3 日志文件 302
13.6 网络数据加密技术浅谈 304
13.6.1 链路加密 304
13.6.2 节点加密 305
13.6.3 端到端加密 305
13.7 小结 306

第1章 综　　述

从目前开发的应用软件的使用范围来看，数据库技术往往在选择开发工具时占据重要的因素。Visual Basic 6.0 给我们提供了用于从数据库中访问数据的强大工具集，使用它既可以建立对本地数据库操作的应用程序，也可以建立适应于网络中的分布式数据处理的应用，建立与大型数据库服务器的远程连接。接下来，我们来看一看 Visual Basic 6.0 较之 VB（Visual Basic）以前的版本有哪些改进。

1.1　Visual Basic 6.0 的新增功能

作为比尔·盖茨以及众多开发人员挚爱的开发工具，Visual Basic 已经经历了很多次的改进，其目的就是使 Visual Basic 能更简单、更快捷地用于设计、编写和调试出优秀的面向对象的应用程序。

VS.net 即将推出，Visual Basic 6.0 作为其中一部分推出。为了适应程序设计目标的要求，Visual Basic 6.0 与其 5.0 版相比，在某些方面有所增加或增强，主要体现在以下几个方面。

1.1.1　数据访问

1. ADO

ADO 是 Microsoft ActiveX Data Objects 的简称，这是一种新兴的数据访问技术，其特性包括：更简单的对象模型，与其他 Microsoft 和非 Microsoft 的技术更好的集成，为本地和远程数据提供通用接口，可以远程访问的和断开的记录集，用户可以访问的数据绑定接口以层次结构的记录集。

2. 数据环境

只在专业版和企业版中才有此内容。数据设计环境器提供了建立 ADO 对象的交互式界面。可作为数据源供窗体或报表上的数据识别对象使用，或者在程序中作为显露在数据环境对象之外的方法和属性进行访问。数据设计环境器不仅支持 Visual Basic 的 UserConnection 设计器的所有功能，还支持一些附加的功能，包括拖放、层次结构、分组及合计等。

3. 增强的数据绑定

在 Visual Basic 6.0 版之前，只允许在一个窗体中绑定控件，而在 Visual Basic 6.0 中，可以将任何 ADO/OLE 数据源绑定到任何 ADO/OLE 数据使用者。在运行期间，可以设置控件的数据源属性，使之动态绑定。该功能只限专业版和企业版。

4. OLE DB 支持

OLE DB 是一组 COM 界面，提供对不同信息源的统一数据访问应用。ADO 便是访问

需要全本请访问 www.ertongbook.com

OLE DB 的一种方法。

5. 可视化的数据工具

可以建立 SQL Server 和 Oracle 数据库表，可以通过拖放创建视图，还可以自动修改表中所列的数据类型。

6. 数据报表

在专业版和企业版中，允许通过拖放快速建立报表。

7. 数据源

允许建立用户控件和类作为数据源，被其他控件绑定。

8. 数据浏览窗口

可以查看已经连接的所有数据库以及它们的表、视图、存储过程等。

9. SQL 编辑器

允许向 SQL Server 和 Oracle 数据库添加存储过程和触发器。

1.1.2 控件

1. ADO Data 控件

这是一种新的对象链接和嵌入数据库（OLE DB）数据源控件，使你可以以最少的代码建立一个数据库应用，其功能极像内部数据和远程数据控件。

2. Collbar 控件

这是一个控件箱，可以用来建立类似 IE 的用户，可配置工作条。

3. DataGrid 控件

这是数据约束网格（DBGrid）的增强，支持 OLE DB，可以用它快速建立浏览和编辑数据表记录的应用。

4. DataList 和 DataCombo 控件

DataList 和 DataCombo 控件是 DBList 和 DBCombo 控件的增强，支持 ADO 数据控件。

1.1.3 向导

1. 增强向导安装

打包和展开向导允许将.cab 文件部署到 Web 服务器、网络共享或其他的文件夹中。这个新的向导集成了对 ActiveX 数据对象（ADO）、对象链接和嵌入数据库（OLE DB）、远程数据对象（RDO）、开放式数据库连接（ODBC）和数据访问对象（DAO）的数据访问的支持，同时也支持新的因特网信息服务（IIS）和动态超文本标记语言（DHTML）应用程序。

2. 工具栏向导

如果加载了应用程序向导，那么窗体上添加工具栏时，工具栏向导将自动打开，然后您就可以创建自定义的工具栏了。

3. 数据对象向导

自动地创建绑定到数据环境或 UserControl 的中层对象。

4. 增强的类生成器实用程序

支持参数列表中的 ParamArray、Optional、ByVal 和 Default 值，并且支持 Enum。

5. 增强的数据窗体向导

允许生成其控件不绑定到数据控件上的纯代码窗体，还允许使用 ADO 代码。该向导

与应用程序向导、图表向导以及 FlexGrid 向导集成在一起，仅适应于专业版和企业版。

6. 外界程序设计器

通过简单指定外界程序的缺省加载行为、名称、目标应用程序和版本以及其他想要的属性，就可以开始进行开发工作。

7. 增强的应用程序向导

可以将设置保存为配置文件，以备将来使用，这就能够使用相同的格式创建多个应用程序。可以直接从应用程序向导内部启动数据窗体向导和工具栏向导，以便创建数据窗体和工具栏，菜单是完全可以自定义的。

1.1.4　语言

1. 公共方法的用户定义类型

用户定义类型可以是公共属性和方法的参数或返回类型。

2. 增强的 StrConv 功能

LCID 参数允许指定一个不同于系统的 LocaleID 的字符串作为一个 LocaleID。

3. 函数可以返回数组

函数和属性过程将可以返回数组。

4. 增强的 CreateObject 功能

CreateObject 可以指定一个机器的名称作为可选的参数，这就可以在远程计算机上创建对象。

5. Dictionary 对象

Dictionary 对象克服了 Collection 对象的一些局限性，而且使用起来也更加方便，在通常的操作过程中，它产生的运行错误更少，并且具有更大的灵活性。

6. 可以分配可变大小的数组

可变大小的数组可以出现在赋值号的左边，而固定大小的数组则不能出现在赋值号的左边，尽管这种数组可以出现在赋值号的右边。

7. CallByName

允许通过指定一个包含属性或方法名称的字符串来访问该属性或方法，而不用在代码中显示指定的属性或方法的名称。

8. 文件系统对象

提供一整套经过优化的、遍历文件系统以及创建文本文件和目录的例程。

9. 新的字符串函数

- Filter 函数
- Formatcurrency 函数
- FormatDateTime 函数
- FormatNumber 函数
- FormatPercent 函数
- InstrRev 函数
- Join 函数
- MonthName 函数
- Replace 函数

- Round 函数
- Split 函数
- StrReverse 函数
- WeekdayName 函数

1.1.5 Internet 描述

1. IIS 应用

IIS（因特网信息服务）是一个 Visual Basic 应用。它将 HTML 和 Visual Basic 代码动态的结合在一起，实现数据浏览。它安装在 Web 服务器上。

2. Web 发布向导

该向导将应用程序包装之后传给 Web 服务器，这样，就可以发布应用了。

3. DHTML 应用

IE 中的动态超文本标记语言（DHTML）技术可以使 Web 页上每一个元素的属性、方法和事件显现出来。DHTML 应用允许编写 Visual Basic 代码响应 HTML 页的动作，而不必将处理转移给服务器。

4. Internet Explore 4.x 数据下载支持 ActiveX 文档

在 Internet Explore 4.x 中下载 ActiveX 文档与下载 ActiveX 控件的操作相同。

另外，Visual Basic6.0 还在以下几方面有所改进：Web Forms、Web Services 和面向对象的语法方面的改进。 Web Forms 使得经验丰富的 Visual Basic 用户可以像现在编写单机程序一样简单地开发网络应用程序。通过 SOAP 接口, Web Services 允许在可以联网的任何地方配置所设计的组件。另外，几个在面向对象的语言方面的关键性的改进使得 Visual Basic 的代码像 C++一样具有灵活性。

(1) Visual Studio Web Forms

Visual Basic 6.0 是 Visual Studio 开发环境的一部分，为网络开发者引入了 Web Form 这个新的概念。引进 Web Form 概念的目的是为了扩展 Visual Basic 的随机存取功能，从而使 Visual Basic 可以应用于影响日益广泛的网络程序的编写。使用 Visual Studio 中的任何一种语言的开发者都可以共享这种 Visual Studio Web Forms。

一个 Web Forms 页包括两部分：实现 Web Forms 页可视界面的一个 HTML 文件和处理 Web Forms 页事件的源文件。既然目前有三分之一基于 Visual Basic 环境的开发是面向网络，Microsoft 计划进一步增强这方面的功能。在 Visual Basic 6.0 中，可以像用 Visual Basic 生成表单一样设计 Web Forms，将拥有一个 Web 控件工具箱，可以直接把控件拖放到 HTML 编辑器中使用，只需要设置一下它们的特性，编写一些适当的代码即可。所以只要知道如何使用 Visual Basic 编写应用程序，Web Forms 就可以让您成为一个网络开发者而丝毫不用改变工作方式。

(2) Web Services

Web Services 是 Visual Studio 开发工具系统采取的第二大改进。就核心而言，一个 Web Services 就是一个通过标准的网络协议发布的中间层的事件处理函数。只要合适地分配 URL，就可以简单地在一个网络应用程序中构造多种 Web Services。在程序运行时，所有这些内部构件之间的调用都会自动打包，通过扩展标记语言（XML）接口进行调用。开发者可以在任何平台上，使用任何语言编写和使用 Web services。如果需要保密，您可以使用安全插口层（SSL，Secure Socket Layer）或标准校检技术。

使用 Visual Basic 6.0，很快就可以把一个具体项目中的函数以 Web Service 的形式发布和实现。您也许很熟悉把一个 Visual Basic 的类定义为 public 的过程。在 Visual Basic 6.0 中将会有一个新的标志，暂时称为 webpublic，这意味着程序将作为 Web Service 发布。它不仅仅可以通过组件对象模型界面（COM）接口为需要它的当地项目所用，而且可以为任何引用了它的 URL 地址的网络程序服务。正如可以把引用加入到一个新项目中的公共对象中一样，也可以把引用加到网络程序中，然后像使用当地程序一样使用它。

当然，运行机制是有些不同的。Visual Basic 6.0 能够通过 COM 接口对当地对象解析引用。当加入一个网络服务的引用到应用程序中时，远程对象将自动生成接口的定义，并使用 SOAP 协议发送到 Visual Studio 开发环境中来。虽然这些将以 XML 形式产生，但不用自己做任何连接的工作，Visual Basic 将自动处理它。在接收到接口定义以后，就可以使用 IntelliSense，如同已经编写了引用该对象的代码一样。

为了方便 Web Services 的开发工作，Visual Basic 将引入一个新的对象类型，即 Web Service。可以像现在创建一个当地的动态链接库（DLL）文件一样简单地设计和发布 Web Service 到远程服务。

(3) 语言改进及更多的语法特性

Visual Basic 6.0 中，Microsoft 加入了面向对象编程的三大特性：继承性、多态性和重载。另外还有一些结构，包括结构化的错误处理和浏览也被引入了 Visual Basic 语言。

目前，Visual Basic 支持 apartment-threaded 模型。虽然这种模型为应用程序的开发提供了真正的高效率，但它还不够理想。Visual Basic 6.0 在这方面有所改进，它采用了 freethreaded 模型，这在编写 scalable 的网络应用程序时很有用处。Visual Basic 6.0 还包括了一些语法结构，可以用来产生多线程。典型的线程发生操作如下：

```
    set t = New Thread(New Threadstart
    (AddressOf(BaseClass.Function1))
```

从该例子中，可以看到 Visual Basic 6.0 有 AddressOf 结构，用它来返回函数的地址，可以不再被迫跳过那些需要函数指针的 API 函数了！

Visual Basic 6.0 的另一项改进是结构化的出错处理。不久以前，Visual Basic 还要求在代码中插入大量的 On Error 声明，而 Visual Basic 6.0 采用了集中处理出错的方式，Visual Basic 6.0 也支持 try-catch-finally 结构。可以在代码的顶端放置一个包含有出错处理的子程序。下面是实现出错处理的一个例子：

```
Sub SafeWrite()
    Try
    Open "Testfile"
    Write #1
    Catch
    Kill "Testfile"
    Finally
    Close #1
    End Try
End Sub
```

在 Visual Basic 6.0 中，可以在变量声明的同时对变量进行初始化：

Dim a as integer = 10

也可以在一个表达式中建立和初始化一个新的对象，并可以通过类来共享变量。不仅仅如此，继承的概念扩展到了项目的用户界面的基础。关于 Visual Basic 6.0 的一个具有代表性的观点是它很难在相同的基础上创建多种不同的表单（在联合开发的环境中，通常有这种要求）。在 Visual Basic 6.0 中，可以通过模板类型来实现。

Visual Basic 6.0 的另外一个重大改进就是使得数据库的操作变得更为方便、快捷，无论是本地数据库，还是远程数据库都是如此！此项改进使得 Visual Basic 6.0 的使用者们不必再为数据库的操纵多元化、复杂化而烦恼了！在接下来的章节里面，将系统、全面地介绍如何运用 Visual Basic 6.0 来操纵数据库。

1.2 数据库的分类

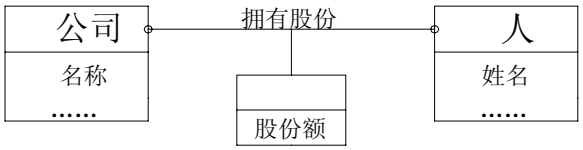
数据库的物理结构方式和逻辑结构方式都会对数据查询和录入的性能产生多方面的影响。特定数据库的性能就是数据结构和数据集大小的反映。开发小组应当花费必要的时间和精力来完成全部数据模型化阶段的工作，以便完成三级规范形式的实体-关系图（有时称为 BNF 及最优化范式（Best Normal Form））。

在数据模型化的数据分析时，应当定义每个实体特性的数据类型及其有效的值域。最好能够用固定长度的数据类型替换掉可变长度的数据类型，这样做需要占用额外的空间，但却是值得的，因为它们的性能高于具有延迟更新问题的数据类型性能。按其结构可以将数据库划分为以下几类。

1.2.1 层次数据库

层次数据库是一种记录倒置树的聚集组织成的数据库。倒置树可以有任意路径，在树顶记录有零个或多个子记录，子记录作为父记录的后继记录。这种父-子关系连续地递归成一棵树。IBM 的 IMS 数据库的产品是最典型的数据库管理系统（DBMS，Data Base ManageMent System）。

多对多关联的强制非对称性是层次数据库的主要缺点。图 1-1 表示对象模型和一些样本数据。



三和厚友	100	鲁日乔
中　石	200	鲁日乔
三和厚友	350	乔立琴
仁　义	512	崔　柯
仁　义	300	乔立琴
仁　义	400	李保林

图 1-1　对象模型和样本数据

图 1-2 左边表示有关公司记录的层次数据库，右边是有关人记录的替换层次数据库。

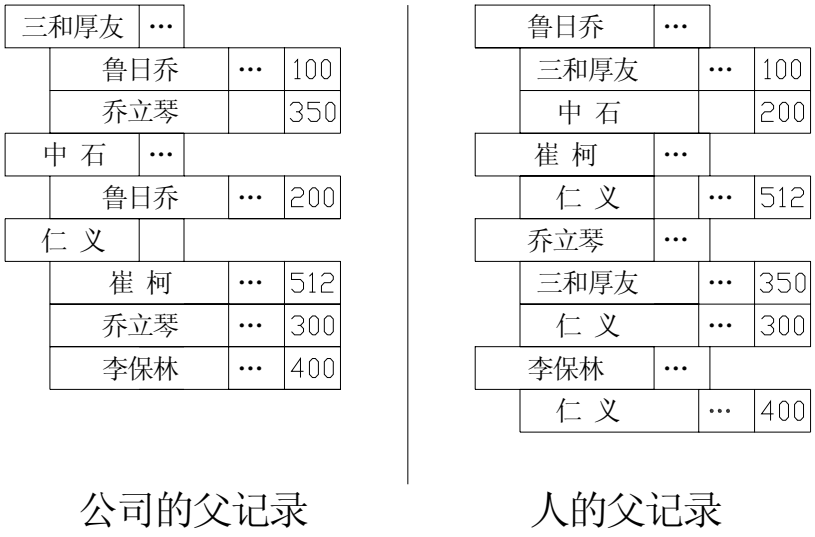


图 1-2　有关公司记录和人记录的层次数据库

数据定义语言（DDL）定义了层次数据库的结构——记录型、字段型、指针和父子关系。

程序必须使用另外一种语言，即数据操纵语言（DML），来存取实际数据。层次数据库有两种方式存取记录：

① 借助于父记录；

② 借助于指针。

通常每个父记录有它的子记录、子孙记录和其他相邻磁盘空间中存储的后代记录。

图 1-3 表示了层次数据库简化的对象元模型。层次数据库包含了不同的记录型，每一个记录型提供了记录实例的格式。某些记录型的实例是树根，应用程序可以导航层次数据库，即从根部开始相继地从父记录导航到子记录，直至找到所需要的子记录为止。应用程序可以用嵌入在记录中的指针进行遍历，并把该指针装订在父-子导航链中。由于邻接存储的连续性，所以搜索层次树是很快的。指针遍历有点慢，但是总体上仍然是非常快的，其他类型的查询要求顺序搜索。

一个记录是字段的列表。每个字段包含一个简单的值，或者包含一个指向记录的指针。一个字段类型定义了值的结构，字段的某些组合可以构成与它的父记录有关的关键字。层次数据库很少支持空值或变长字段。

层次式范例结构在表达复杂信息模型之中是无能为力的。自然的层次体系结构通常是不存在的，并且强制在父记录和子记录之间的层次结构是难以处理的。诚然，部分指针可以对强制性控制进行一些补偿，但是，对大型模型设计一个合适的模型困难重重。程序必须直接地导航到物理数据结构，所以不存在着像关系 DBMS 那样的数据独立性。因此，层次数据库和它的应用在维护和扩展新的意料之外的应用时是困难的。

需要全本请访问 www.ertongbook.com