

软件开发技术丛书



Visual Basic 6

数据库访问技术

Database Access with Visual Basic 6

(美) Jeffrey P. McManus 著 赵军锁 龚波 李志 等译



机械工业出版社
China Machine Press

SAMS
PUBLISHING

软件开发技术丛书

Visual Basic 6 数据库访问技术

(美) Jeffrey P. McManus 著

赵军锁 龚波 李志 曹天伟 等译



机械工业出版社
China Machine Press

Microsoft公司最新推出的Visual Studio 6.0提供了众多的先进特性和强大工具,而Visual Basic 6是其核心开发工具之一,它的强大功能使VB数据库开发人员的工作面临较大的改变。本书包含了Visual Basic开发人员可利用的全部数据库访问技术,重点介绍用Visual Basic 6进行数据库程序设计的最新概念,特别详细讲解了最新版本Jet包含的几个数据库引擎新增特性、开放环境中新增的ActiveX Data Objects (ADO)以及与ADO有关的一些工具。

本书虽然是针对Visual Basic 6而著,但是相当多的内容适用于Visual Basic 5。

本书内容丰富、条理清晰,适合用Visual Basic 6进行数据库程序设计的高级开发人员阅读。

Jeffrey P. McManus: Database Access with Visual Basic 6.

Authorized translation from the English language edition published by Sams Publishing, an imprint of Macmillan Computer Publishing. U.S.A.

Copyright 1999 by Sams Publishing.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Chinese Simplified language edition Published by China Machine Press.

Copyright 1999 by China Machine Press.

本书中文简体字版由美国麦克米兰公司授权机械工业出版社独家出版,未经出版者书面许可,不得以任何方式复制或抄袭本书的任何部分。

版权所有,翻印必究。

本书版权登记号:图字:01-1999-1405

图书在版编目(CIP)数据

Visual Basic 6数据库访问技术/(美)麦克曼斯(McManus, J. P.)著;赵军锁等译.-北京:机械工业出版社,1999.10

(软件开发技术丛书)

书名原文:Database Access with Visual Basic 6

ISBN 7-111-07266-9

I.V... II.①麦... ②赵... III.Basic语言-数据库管理系统-程序设计 IV.TP312

中国版本图书馆CIP数据核字(1999)第26320号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码100037)

责任编辑:姜南

北京忠信诚胶印厂印刷·新华书店北京发行所发行

1999年10月第1版第1次印刷

787mm×1092mm 1/16·33印张

印数:0 001-5 000册

定价:65.00元(附光盘)

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

译 者 序

数据库是许多商业软件的核心。由于数据库允许以一种高效、一致的方式对信息进行集中访问，并且易于建立和维护，所以在商业事务处理领域极为流行；而Visual Basic作为一种流行的程序设计语言，深受广大程序设计人员的喜爱。本书主要针对当今用Visual Basic 6进行数据库程序设计的最新概念，介绍如何利用最好的技术和一些高深的技巧，结合现实世界中应用程序、解决方案和项目的应用，对使用Visual Basic 6进行高级数据库访问所需的方方面面进行了详细的介绍。

由于最新版本的Jet包含了几个数据库引擎的新增特征，开发环境中新增了ActiveX Data Objects(ADO)以及与ADO有关的一些工具，使Visual Basic数据库开发人员面临着较大的改变。所以只有及时了解和掌握这些新的技术和先进的开发方法，才能成为一名可以使用Microsoft开发工具进行熟练高效开发的高级程序员。本书正是为了帮你达到这一目的而编写的。

另外，本书包含了几乎Visual Basic开发人员可利用的全部数据库访问技术。其重点不是Visual Basic程序设计，而是介绍那些与数据库访问有关的重要方面。主要有：

- 数据库技术，如何从Visual Basic中访问数据库。
- Visual Basic 6.0的新颖之处。
- client/server数据库应用开发。
- 使用Web浏览器的数据驱动的Internet应用程序。
- 有关的第三方工具的详尽介绍。

Visual Basic 6的许多新的特征和功能都与数据库访问有关；其开发环境也有了较大的改善，如数据视图窗口和数据环境设计器，使得与数据库建立连接和与之一起工作较以往更为容易，但如果没有安装Visual Basic 6，这本书也是适合你的。因为本书95%的主题也适用于Visual Basic 5。

本书是美国专业作家Jeffrey的又一力作。其第1版曾倍受广大Visual Basic程序员的青睐和推崇，由于作者本身不仅是一个专业的开发人员，而且是这个领域的一位资深专家和多本相关书籍的作者。其语言风格浅显易懂，深入浅出，以具体的商业数据库解决方案为主线对Visual Basic中的相关数据库访问技术进行了深刻而详尽的介绍，相信它肯定会使业界人士从中得到有益的启示和帮助。

本书由赵军锁、龚波、李志组织进行翻译，参加本书翻译、录排、校对工作的人员还有：曹天伟、张巧莉、田丽榘、陈曙晖、邓波、邓涛、李林、李卓林、聂宛析、田敏、金光、小光、龚露娜、马军、马丽、田军、田洗县、王小将、高翔、丁建华、崔羽、大军、李节等。本书的出版是集体劳动的结晶，在此特别感谢前导工作室的全体工作人员。

由于译者的水平有限，加之时间仓促，书中难免有不妥之处，请读者批评指正！

4Developers COM Explorer**Spectrum Access Developer Utilities**

这些是试用版的软件。其中有些软件的使用时间是受限的，有些会显示一个屏幕要求你进行注册。如果要注册，请与相应的厂商联系。

所有这些试用软件都有各自的安装程序。它们在本书所带光盘的\3rdParty目录中都有其各自的子目录。

与作者和出版者的联系方式

我很愿意能够收到读者的评论、问题和建议。如果你想和大家共享某些东西，可通过E-mail: jeffreyp@sirius.com与我联系。我或许不能够马上给你答复，但是肯定会认真阅读我收到的每封邮件。

本书在<http://www.redblazer.com/vbdb/>有一个Web站点。在本书的第1版出版之后，我又不时往该站点上登载了一些新的带有代码和过程的补充性文章。对于本书的第2版，我还会这样做，请别忘了光顾一下该站点。

十分感谢你选择了本书，希望能对你有用！

出版者的联系方式：

Fax: 317-817-7070

E-mail: programming@mcpo.com

Mail: Tracy Dunkelberger

Executive Editor

Advanced Programming and

Distributed Architectures Team

Macmillan Computer Publishing

201 West 103rd Street

Indianapolis, IN 46290 USA

目 录

译者序

前言

第1章 数据库基础1

1.1 用数据控件创建用户界面1

1.2 何谓数据库1

1.2.1 何谓数据库引擎1

1.2.2 表和字段2

1.2.3 何谓Recordset5

1.2.4 数据类型5

1.2.5 创建数据库规划6

1.2.6 用Visual Basic创建数据库12

1.2.7 用Microsoft Access创建数据库19

1.2.8 关联21

1.2.9 用更神奇的技术创建数据库24

1.2.10 标准化24

1.3 使用数据视图窗口28

1.4 创建DataEnvironment设计器30

1.5 用数据控件创建用户界面32

1.5.1 连接到数据库并操作记录33

1.5.2 创建基本的用户界面35

1.5.3 用ADO数据控件操作记录36

1.5.4 ADO数据控件的其他重要属性42

1.6 小结43

1.7 问题与解答43

第2章 查询45

2.1 何谓查询45

2.2 SQL用于什么地方45

2.3 用Data View窗口测试查询46

2.3.1 使用Data View窗口创建服务器
方查询50

2.3.2 使用数据环境查询52

2.4 用SELECT子句检索记录53

2.5 使用FROM子句指定记录源54

2.6 用WHERE子句说明条件55

2.7 使用ORDER BY对结果排序58

2.8 使用TOP显示某个范围的第一个记录
或最后一个记录59

2.9 在查询中连接相关的表61

2.9.1 在SQL中表示一个连接62

2.9.2 使用查询设计器创建连接63

2.9.3 使用外部连接以返回更多的
数据65

2.9.4 在连接查询中显示0而不是
Null66

2.10 用AS对字段名进行别名化67

2.11 分组和汇总数据的查询68

2.11.1 SUM函数69

2.11.2 合计函数小结69

2.12 创建功能查询70

2.12.1 更新查询70

2.12.2 删除查询71

2.12.3 追加查询71

2.12.4 创建表的查询72

2.13 合并查询72

2.14 交叉表查询73

2.15 子查询74

2.16 使用存储在数据库中的查询75

2.16.1 使用Visual Data Manager创建
存储查询77

2.16.2 Visual Data ManagerQuery
Builder的使用79

2.16.3 在Visual Data Manager中创建
连接80

2.16.4 使用Microsoft Access创建存储
查询82

2.16.5 在Access中创建其他类型的
查询85

2.17 在运行时创建存储查询85

2.18 使用数据定义语言 (DDL)	86	3.4 小结	153
2.18.1 使用CREATE创建数据库元素	86	3.5 问题与解答	153
2.18.2 向表添加约束	87	第4章 报表输出和数据导出	154
2.18.3 指定外部关键字	88	4.1 前向滚动临时表	154
2.18.4 用CREATE INDEX创建索引	88	4.2 用VB的DataReport设计器进行报表 输出	154
2.18.5 用DROP删除表和索引	89	4.2.1 设计DataReport	155
2.18.6 用Alter修改表的定义	89	4.2.2 浏览和导出DataReport	157
2.19 小结	90	4.3 用Microsoft Access进行报表输出	158
2.20 问题与解答	90	4.3.1 从VB中运行Microsoft Access	158
第3章 数据访问对象	92	4.3.2 使用VideoSoft VSREPORTS运行 Access报表	161
3.1 使用DAO对象模型	93	4.4 使用Crystal Reports进行报表输出	165
3.1.1 用对象编程	94	4.4.1 安装Crystal Reports	166
3.1.2 使用DAO数据控件	98	4.4.2 使用Crystal Reports创建报表	166
3.1.3 使用DAO数据控件的查询属性 来访问外部数据源	98	4.4.3 用Crystal Reports ActiveX 控件在 应用程序中运行报表	169
3.2 使用DAO操作数据	99	4.4.4 获取和使用新版本Crystal Reports	170
3.2.1 使用Database对象连接到数据 库	99	4.5 使用MSChart控件创建图表	170
3.2.2 使用Recordset对象	103	4.6 使用VB的Printer对象	170
3.2.3 用Field对象操作字段	107	4.6.1 建立Printer的坐标系	171
3.2.4 结合Recordset对象使用定位 方法	110	4.6.2 设计报表	171
3.2.5 搜索记录集和表中的数据	117	4.6.3 向Printer对象输出Recordset	171
3.2.6 用Workspace对象访问会话信息	126	4.7 使用VideoSoft VSVIEW创建报表	175
3.2.7 用Errors集合和Error对象处理 错误	130	4.7.1 用vsPrinter控件打印表	176
3.3 创建操作数据库结构的对象	132	4.7.2 使用VideoSoft Data Reporter组件 创建Ad; Hoc报表	179
3.3.1 创建数据库	132	4.8 使用Microsoft Scripting Runtime Library向文本文件导出数据	182
3.3.2 使用TableDef对象操作表	133	4.8.1 使用TextStream对象导出一个 记录集	184
3.3.3 使用Relation对象在表间创建 关联	136	4.8.2 创建分隔数据库输出	185
3.3.4 使用Index对象创建索引	138	4.8.3 导出HTML Web页面	188
3.3.5 使用QueryDef对象操作存储 查询	140	4.9 使用VB的“古典”文件函数向文本 文件导出数据	195
3.3.6 压缩和修复Jet数据库	145	4.9.1 使用Open语句创建文件	195
3.3.7 数据库Documents和Containers 的使用	148	4.9.2 使用FreeFile确定下一个可用的 文件	196
3.3.8 创建和使用数据库对象的自定 义属性	151		

4.9.3 使用Print # 格式化输出	196	6.1.2 用ODBCPING测试数据库连接	248
4.9.4 使用Close语法关闭文件	197	6.1.3 用DAO数据控件和ODBCDirect 访问数据源	249
4.10 向Microsoft Office应用程序导出 数据	197	6.2 使用远程数据控件访问数据	250
4.11 小结	200	6.3 在代码中使用远程数据对象 (RDO)	252
4.12 问题与解答	200	6.3.1 用rdoEngine设置数据库引擎 的属性	252
第5章 SQL Server	202	6.3.2 用rdoEnvironment对象访问 环境	254
5.1 安装和运行Microsoft SQL Server	203	6.3.3 使用rdoConnection对象建立 连接	255
5.1.1 确定SQL Server的安装需求	203	6.3.4 在RDO中响应事件	259
5.1.2 安装SQL Server	203	6.3.5 使用UserConnection设计器创建 连接	260
5.1.3 使用SQL Server Manager启动和 终止SQL Server	204	6.3.6 用UserConnection设计器访问 查询	262
5.1.4 SQL Server入门: 基础知识	205	6.3.7 用rdoResultset对象操作数据	268
5.1.5 使用SQL Enterprise Manager创建 数据库	209	6.3.8 用rdoQuery 对象运行查询	268
5.1.6 在SQL Server数据库中创建表	210	6.3.9 用rdoTable对象访问表	274
5.1.7 使用ISQL/w访问数据库	214	6.3.10 用rdoErrors集合和rdoError对象 处理错误	276
5.1.8 使用数据库视图控制对数据的 访问	224	6.4 编写自己的数据库对象服务器	277
5.1.9 创建和运行存储过程	227	6.5 小结	278
5.1.10 显示现有视图或存储过程的 文本	229	6.6 问题与解答	278
5.1.11 创建触发器	230	第7章 用类进行数据库访问	279
5.1.12 在SQL Enterprise Manager中管理 用户与安全	233	7.1 类和对象的使用	280
5.1.13 在ISQL/w中运用安全属性	237	7.1.1 建立自定义的类	280
5.1.14 从数据库中删除对象	239	7.1.2 创建集合与集合类	285
5.2 由Microsoft Access迁移到SQL Server	239	7.1.3 使用VB Class Builder Utility创建 类分层结构	286
5.2.1 使用Microsoft Access Upsizing Tools	239	7.1.4 把表单作为类使用	291
5.2.2 使用链接表把数据从Access导出 到SQL Server	240	7.1.5 创建多个表单实例	293
5.3 小结	243	7.2 用类和对象进行数据库访问	293
5.4 问题与解答	244	7.2.1 使用单个记录处理类	294
第6章 开放数据库连接与远程数据 对象	245	7.2.2 使用数组处理类	300
6.1 ODBC的配置和使用	245	7.2.3 使用工厂方法	300
6.1.1 创建ODBC数据源	246	7.3 创建数据相关的类	302
		7.4 创建输出数据的类	305

7.5 基于类的ActiveX服务器程序	308	9.1.2 使用记录集层次的锁	348
7.5.1 在标准EXE项目中使用一个ActiveX 组件	308	9.1.3 使用页面级锁	349
7.5.2 在用户的计算机上注册ActiveX 组件	311	9.2 使用Microsoft Jet数据库安全机制	353
7.5.3 把一个Standard EXE项目转化为 一个ActiveX组件项目	311	9.2.1 用代码访问安全Jet数据库	353
7.5.4 远程使用ActiveX组件	311	9.2.2 给数据库分配一个口令	355
7.5.5 创建多线程组件	312	9.2.3 在代码中标识当前用户	357
7.5.6 多线程组件的限制	312	9.2.4 使用Workspace对象登录数据库	358
7.6 小结	313	9.2.5 创建一个工作组信息文件	359
7.7 问题与解答	313	9.2.6 使用DAO代码分配工作组信息 文件	361
第8章 远程数据访问	315	9.2.7 创建和删除用户	361
8.1 关于客户(机)、服务器和代码组件	315	9.2.8 创建和删除用户组	364
8.2 用三层client/server体系结构封装商业 规则	316	9.2.9 往用户组添加用户	366
8.2.1 为DCOM建立硬件体系结构	317	9.2.10 Jet数据库引擎创建的缺省用 户组	368
8.2.2 在其他平台上使用DCOM	318	9.2.11 分配和删除数据库对象的所 有权	368
8.2.3 创建第一个DCOM应用程序	318	9.2.12 给用户和用户组分配权限	369
8.2.4 ActiveX EXE	318	9.2.13 加密Microsoft Jet数据库	374
8.2.5 用Microsoft Transaction Server 来部署ActiveX DLL	319	9.2.14 实现Jet数据库安全的对照表	375
8.3 借助ActiveX组件使数据库访问更加 容易	324	9.3 小结	375
8.3.1 使用GetRows以数组形式返回 数据	325	9.4 问题与解答	376
8.3.2 创建一个类用于解码Variant数组	327	第10章 ActiveX数据对象	377
8.4 利用数据库复制传输数据	331	10.1 用ActiveX数据对象创建Visual Basic 应用程序	377
8.4.1 在设计数据库中考虑复制功能	332	10.1.1 理解OLE DB/ADO结构	377
8.4.2 在Microsoft Access中进行复制	333	10.1.2 在Visual Basic应用程序中安装和 创建对ADO的引用	378
8.4.3 在DAO中进行复制	336	10.1.3 与其他数据访问对象库一起使用 ADO	380
8.4.4 使用部分复制	342	10.1.4 使用ADO的Connection对象连接 到某数据源	382
8.4.5 在Internet上执行数据库复制	344	10.1.5 使用ADO Recordset对象来操作 数据	387
8.5 小结	345	10.1.6 利用Recordset对象插入和更新 记录	390
8.6 问题与解答	345	10.1.7 创建无连接记录集	393
第9章 多用户Jet数据库	346	10.1.8 使用ADO Command 和Parameter 对象执行查询	398
9.1 Microsoft Jet数据库引擎的数据锁	346		
9.1.1 使用数据访问对象锁住整个 数据库	347		

10.1.9 利用ADO的Field对象访问记录集数据	400	12.2.2 用RichTextBox控件显示格式化的数据	445
10.1.10 利用ADO的Errors集合处理错误	401	12.2.3 使用DBCombo和DBList控件显示数据	446
10.1.11 利用Properties集合查看提供者指定的属性	401	12.2.4 使用DataGrid控件	448
10.2 使用ADO远程数据服务	403	12.2.5 使用DataList和DataCombo控件	449
10.3 小结	404	12.2.6 使用DataRepeater控件以定制行的方式查看数据	450
10.4 问题与解答	404	12.2.7 使用MonthView和DateTimePicker控件	452
第11章 基于Internet的数据库应用程序	406	12.2.8 使用MSChart控件图形化地显示数据	453
11.1 为Active Server Pages设置和配置Microsoft Internet Information Server	406	12.2.9 使用DBGrid、MSFlexGrid和Hierarchical FlexGrid控件	454
11.1.1 用ASP编写脚本	408	12.3 创建数据库相关的ActiveX控件	454
11.1.2 在ASP中执行ActiveX Data Object查询	408	12.4 小结	455
11.1.3 在Netscape和其他浏览器中使用ASP	412	12.5 问题与解答	455
11.1.4 利用ASP和HTML表单插入记录	413	第13章 使用DBGrid和Apex True DBGrid控件	456
11.2 小结	423	13.1 使用DBGrid控件	456
11.3 问题与解答	423	13.2 从DBGrid控件开始	457
第12章 用户界面控件	424	13.2.1 在设计时选择要显示的数据列	458
12.1 使用固有的数据相关控件	424	13.2.2 在设计时处理数据列	459
12.1.1 用文本框控件输入数据	424	13.2.3 使网格数据可编辑	460
12.1.2 绑定到DataEnvironment设计器	428	13.2.4 在设计时往网格中插入分隔条	461
12.1.3 利用复选框控件访问布尔值	431	13.2.5 在代码中处理分隔区域	462
12.1.4 利用列表框显示数据	432	13.2.6 在代码中处理网格的数据列	462
12.1.5 使用标准的组合框控件	437	13.2.7 用Bookmark属性在网格中定位	463
12.1.6 用标签控件显示只读数据	437	13.2.8 利用SelBookmarks集合选择记录	464
12.1.7 使用PictureBox控件显示位图图象	438	13.2.9 在非绑定模式下使用DBGrid控件	464
12.1.8 用OLE容器控件显示二进制对象	439	13.3 使用True DBGrid Pro开发数据库相关应用程序	465
12.2 使用数据相关的ActiveX控件	441	13.3.1 从以前的版本迁移到True DBGrid	467
12.2.1 利用MaskedEdit控件控制文本输入	442		

13.3.2 使用Style 对象存储和应用 格式	468	14.4.2 编辑单元格中数据	488
13.3.3 用ValueItem对象更改数据列 的显示方式	471	14.4.3 执行自动搜索	489
13.3.4 使用CellTips为用户提供帮助	473	14.4.4 自动排序	489
13.4 小结	474	14.4.5 访问单元格属性	490
13.5 问题与解答	474	14.4.6 在单元格中显示组合框	492
第14章 使用MSFlexGrid、Hierarchical FlexGrid和VideoSoft VSFLEX 控件	475	14.4.7 保存网格内容到磁盘文件	493
14.1 使用在线决议支持	475	14.4.8 自动地重调整数据列的尺寸	494
14.2 使用MSFlexGrid控件	475	14.4.9 绑定到数组	495
14.2.1 添加数据到MSFlexGrid控件	476	14.5 小结	496
14.2.2 对FlexGrid控件中数据排序	480	14.6 问题与解答	496
14.2.3 合并FlexGrid控件中单元格的 数据	481	第15章 使用Sheridan DataWidgets	497
14.2.4 使用带有数据库的FlexGrid 控件	484	15.1 Sheridan DataWidgets程序组 总览	497
14.3 使用Hierarchical FlexGrid控件	486	15.1.1 DataWidgets 3.1的新功能	497
14.4 使用VideoSoft VSFlexGrid Pro 6.0	487	15.1.2 使用Sheridan SSDBGrid控件	499
14.4.1 绑定vsFlexArray控件到某 数据源	488	15.1.3 使用SSDataCombo控件	508
		15.1.4 使用SSDBDropDown控件	509
		15.1.5 使用SSDataOptSet控件	511
		15.1.6 使用SSDBData控件	512
		15.1.7 使用SSDBCommand按钮	514
		15.2 小结	516
		15.3 问题与解答	516

第1章 数据库基础

1.1 用数据控件创建用户界面

数据库是许多商业软件的核心。由于数据库允许以一种高效、一致的方式对信息进行集中访问，并且易于建立和维护，所以在商业事务处理领域极为流行。本章介绍建立和维护数据库的一些基础知识，包括什么是数据库，为什么数据库如此有用，如何利用数据库创建商业解决方案。

如果你曾经使用过Visual Basic(VB)或进行过数据库程序设计，你可能觉得本章内容十分基础，但是，这些内容会帮你迅速建立起可能有别于其他数据库系统的有关概念术语。

尽管数据库的概念不太可能因数据库系统的不同而有明显的差异，但不同的厂商实现却常常使用它们自己的名称。在某个厂商的系统中，指称某事物的术语可能在另一个厂商的系统中具有完全不同的含义。例如，许多client/server程序员把数据库查询称为视图，Visual Basic和Access程序员却称之为查询或QueryDefs。但两者基本上是一回事。

如果是从以前的版本升级到了Visual Basic 6——特别是升级前的版本为Visual Basic 3——那么需要了解用Visual Basic进行数据库程序设计的几个最新概念。Visual Basic包含最新版本的Jet数据库引擎 (Jet Database engine),Microsoft Access与Visual Basic对该引擎是共享使用的。该版本的Jet包含了几个数据库引擎的新增特征，这将在本章进行介绍并且在本书的其余章节也会经常地进行引用。同时，开发环境中新增的ActiveX Data Objects(ADO)以及与ADO有关的一些工具，使VB数据库开发人员面临着较大的改变。如果你对用32位Visual Basic进行数据库开发十分熟悉，或许能够直接跳到第10章，“ActiveX 数据对象”，看看VB6都新增了哪些东西。

1.2 何谓数据库

数据库是信息的仓库。数据库有几种不同的类型，本书主要论及关系数据库，该数据库是当今世界最常用的数据库类型。关系数据库：

- 在表中存储数据。表由行和列构成，行也称为记录，列也称为字段。
- 用于从表中检索（或查询）数据的子集。
- 能够将不同的表连接（或联系）在一起以便从不同的表中获取相关联的记录。

1.2.1 何谓数据库引擎

数据库的基本函数是由数据库引擎提供的，该引擎即为一个软件系统，用于管理数据如何存储和检索。

本书中涉及的数据库引擎是Microsoft Jet。然而Jet并非一个商业软件产品，而是由几个Microsoft产品共用的一个子系统。Microsoft最初在Visual Basic 3和Microsoft Access 1中应用了该引擎，自从其引入以后便不停地改进和扩展。本书中使用的是Jet 3.51，它与Microsoft

Visual Basic和Microsoft Access一起发行。

注释 除Jet之外还有许多其他的数据库引擎，但是由于Visual Basic对Jet的内在支持，本书便将讨论主要集中在该引擎上了。另外，Jet就象Microsoft Access风格的数据库一样，能够支持其他的数据库引擎，所以关于Jet数据库的大部分讨论也适用于其他数据库引擎。第5章“SQL Server”中讨论了一种完全不同的数据库引擎，即Microsoft SQL Server 6.5的数据库引擎。

商业实例1.1 Jones Novelties公司介绍

许多计算机书籍都是由一个很长的软件功能列表后跟草率的解释构成的。如果你运气好，在关于软件的讨论中可能包含了某些与现实世界有关的软件示例。

但是，本书的任务是根据商业解决方案来解释软件。因此，每一章中都包含了几个商业实例，在实例中，一个假想的公司期望达到完全的办公自动化目标，直接面向的是现实世界中的商业问题。

本书中的商业实例围绕Jones Novelties公司的动人业绩。

公司的首席执行官(CEO)，Brad Jones，意识到商业的成功需要使公司的大部分事务处理实现自动化。Jones必须实现顾客合同(协议)、存货清单和资金系统，使之既与当前的业务相适用，又具有充分的可扩展性，经受得起随时间而引发的诸多变化。

Brad意识到公司的发展和衰败在一定程度上取决于对信息的访问，因此他决定用关系数据库来管理公司的信息。数据库的设计及其功能是本章其余部分讨论的重点。

1.2.2 表和字段

数据库由表构成，表用来表示广泛的数据类别。例如，如果你正在创建一个处理帐目的数据库，你可能会为顾客创建一个表，为各种发票建一个表，而将另外一个表用于雇员。表具有预定义好的结构，它们包含的是适合于该结构的数据。

表由记录组成，记录是表中表示的广泛的数据类中的单条数据。

记录由字段构成，字段代表着记录中的数据子集。例如，一条记录代表地址簿中的一项，可以由许多字段构成，包括姓名、地址、城市、州、邮政编码和电话号码等。

可以用Visual Basic代码来引用和操作数据库、表、记录和字段。

设计数据库

要创建一个数据库，首先要确定它将用于记录什么信息。然后，设计数据库、创建由字段构成的表，字段定义了将要存储的数据是什么类型。在创建了该数据库结构之后，数据库便能够以记录的形式存储数据了。你无法向没有表或字段定义的数据库添加数据，因为数据库不知道把数据放到哪儿。所以说数据库设计是关键，尤其是一旦实现之后便很难改变数据库的设计了。

本书中以标准格式表示表，在图中的最上方是表的名称，以下是字段名列表：

tblMyTable
ID
FirstName
LastName
...

最后一个字段中的省略号（…）表示该表还有一个或多个字段，本书为了简明起见而将它忽略了。

如果你对数据库程序设计不熟悉，但曾使用过其他计算机应用程序，那么可能会对数据库使你在录入数据之前省去了很多麻烦感到惊讶。例如，一个字处理应用程序，只是允许你键入，至于文件如何保存等细节是应用程序本身的事情。事前设计好数据库的原因是为了提高效率。如果一个计算机应用程序确切地知道要存储的数据的种类和数量，它便能够以一种优化的方式存储和检索数据。正如在创建了第一个100 000条记录的多用户数据库之后所了解的那样，在数据库世界中，速度是至关重要的，能够向数据库添加记录或获取信息更为快速的任何尝试都是有价值的。

除效率之外，数据库设计的另一个指导性原则便是将与同一类数据相关的字段放在同一个表中。所有的顾客记录都放在Customer表中，这些顾客的定单都应该放在Order表中，以此类推。

因为把这些数据集放入不同的表单并不意味着你就不能在一起使用它们，而是恰恰相反。当你要使用的数据跨越了关系数据库中的两个或更多的表时，可以使用关联（relationship）来访问该数据。本书在后面将讨论关联，现在，我们重点讨论表的设计。

商业实例1.2 设计表和关联

Brad Jones已经确定Jones Novelties需要一种方式存储顾客的信息。他知道其大部分业务都是重复性工作，所以他希望能够每年向其顾客发送两次价目表以保持联系。

于是Brad在餐巾之上大致地勾勒了一个基本的数据库规划。他指出，“以下是在业务中应该跟踪的东西”：

- 顾客的名字、地址、城市、州、邮政编码以及电话号码。
- 顾客的国内区域（西北、西南，中西部，东北，南或东南）。
- 顾客上一次购买的日期。

Brad指出他应当能够容易地将这些信息放在一个表中，并且可以使数据库简单而精致。

Brad的数据库开发小组大胆地指出虽然这或许是可行的，但这样做最终将导致数据库低效、无组织并且可扩展性极差。

Brad希望包含的信息并非都能直接映射到数据库的字段。例如，因为国内区域是顾客所居住的州的函数，在同一张表中设立State字段和Region字段的意义不大。这样做意味着在输入时要在同一信息上重复两次。而更合理地做法是让数据库在Customer表中存储一个State字段，而将属于区域信息的数据放到Region表中。如果Region表知晓州与区域的映射关系，那么录入人员便无需为每个顾客输入区域信息了。这样，他们只需输入州名，而Customer表可以借助Region表来确定顾客的国内区域。

与此相似，将Name字段分解为FirstName字段和LastName字段，在数据被录入后，对这些字段排序也较为容易。在你看来这些设计技术或许微不足道，但奇怪的是有很多的数据库设计根本没有考虑这些事情——因此在他将系统与数据库集成时便很难掩盖这种设计上的缺陷。

因此Brad和他的开发小组决定Jones Novelties的顾客应该存储在一个表中，叫做tblCustomer，包含的字段如下：

tblCustomer
ID
FirstName
LastName
Company
Address
City
State
Zip
Phone
Fax
Email

属于顾客国内方位的数据应该存储在tblRegion表中。其包含的字段如下：

tblRegion
State
RegionName

两表之间通过State字段建立起一个关联。注意，该字段在两个表中都存在。Region表和Customer表之间的关联为一对多关联，对于tblRegion表中的每一个记录，在tblCustomer中存在一个、多个或零个匹配记录（本章后面关于关联的部分将详细讨论如何在检索记录时利用关联）。

注意数据库的设计者在进行表设计时如何命名表和字段。首先，他在命名每个表时使用了一个前缀“tbl”。这就使得一眼便可看出这是一个表而不是其他能存储记录的数据库对象。其次，每个字段名都使用了全名（而不是简写）并且不包含空格或其他特殊字符如下划线。

尽管Microsoft Access数据库引擎允许你使用空格、下划线和其他特殊字符命名数据库对象，但最好是避免使用它们，因为使用了它们将使日后记忆字段名的确切拼写时产生困难（例如，必须记住字段名是FirstName还是FIRST_NAME）。虽然该规则现在看来微不足道，但当你开始为一个有50张表和300个字段的数据库编写代码时，你就会发现这样做可以较为容易地与最开始的字段命名保持一致。

Brad的期望还有一个没有满足：如何回答问题“顾客最后一次从我处购买物品是什么时候呢？”数据库的开发人员确定该信息可以从存储顾客订单的表中的日期值得出。该表的设计如下：

tblOrder
ID
CustomerID
OrderDate
ItemID
Amount

在该表中，ID字段用于唯一标识每个订单。CustomerID字段，将订单和顾客联系起来。为了将每个订单都与顾客联系起来，顾客的ID被拷贝到Order表的CustomerID字段。这样，查找特定顾客的所有订单便轻而易举了（在后面我们会加以说明）。

1.2.3 何谓Recordset

一旦具备了创建表的能力，便需要一种方法来操作它们。操作表涉及到往表中输入数据和从表中检索数据，同时还要能够检查和修改表的结构。要想操作表的结构，需要使用数据定义命令（见第2章“查询”）或一个TableDef对象（在第3章“数据访问对象”中进行介绍）。要操作表中的数据，需要使用recordset。

Recordset是一个数据结构，通常表示从数据库中提取出来的一个记录子集。从概念上讲，它与表相似，但它还有一些自己的重要特性。

Recordset由对象表示，在概念上与用户界面对象相似（如命令按钮和文本框），在过去你或许在Visual Basic中碰到过它们。与Visual Basic的其他类型的对象相似，Recordset对象有其自己的属性和方法。

更多的信息 可以在三个数据访问库——Data Access Objects(DAO), Remote Data Objects (RDO) 和ActiveX Data Objects (ADO) 的任意一个库中编写代码创建和使用Recordset。DAO的更多信息见第3章“数据访问对象”，RDO见第6章“开放数据库连接与远程数据对象”。关于ADO，见第10章“Active X数据对象”。

1.2.4 数据类型

如果你从前曾编写过程序，你可能已习惯于使用数据类型了。Visual Basic是一种弱类型语言，（在本讨论中）这意味着你通常不用声明所用变量的数据类型，而在强类型语言中必须这样做。如果你选择不去明确定义变量类型，那么其缺省的数据类型是Variant，这是最容易使用的方法（但效率不高）。

下面是弱类型Visual Basic代码不声明变量类型的一个例子，其数据类型使用的是其缺省值：

```
Private Function MySquareLoop( )
```

```
For x = 1 To 10000
```

```
    TheValue = TheValue + (x ^ 2)
```

```
Next x
```

```
MySquareLoop = TheValue
```

```
End Function
```

Visual Basic也允许在代码中进行强制类型定义。下面是MySquareLoop函数的修订版，对其所有的变量都声明了其类型：

```
Option Explicit
```

```
Private Function NewSquareLoop( ) As Single
```

```
Dim x As Integer, TheValue As Single
```

```
For x = 1 To 10000
```

```
    TheValue = TheValue + (x ^ 2)
```

```
Next x
```

```
NewSquareLoop = TheValue
```

End Function

该函数的第1版和第2版并没有什么大的不同，只是NewSquareLoop比MySquareLoop运行快50%。这也是声明变量类型的原因：强类型数据执行更快，尤其是在必须执行重复操作时更是如此。

这同样也适用于数据库。当你设计表时，在建立字段的过程中有一步是声明每个字段的类型，这使得数据库引擎在保存和检索数据时更为高效。传统的Visual Basic程序设计中的数据类型定义与数据库程序设计中的数据类型定义的唯一不同是必须为创建的数据库字段进行强类型定义。

Visual Basic的内在数据库格式，即所谓的Jet，提供了21种不同的数据类型。表1-1列出的是在Visual Basic数据库应用程序中可使用的数据类型。

表1-1 Visual Basic数据库中可使用的数据类型

数据类型	描 述
Binary	用于存储大块数据如图形和数字声音文件
Boolean	具有真值或假值
Byte	单字节整数，从0到255
Currency	数值字段，具有特定属性，可存储精确的货币值
Date / Time	8字节值，表示日期或时间。从100年1月1日到9999年12月31日
Double	8字节，双精度数值数据类型
GUID	128字节数值，也称作全球唯一标识符（Globally Unique Identifier）。可以使用该数字唯一标识一个记录；该数值通常用于复制（见第9章“多用户Jet数据库”）
Integer	两字节有符号数值，范围从-32 768到32 767
Long	四字节有符号整数，范围从-2 147 483 648到2 147 483 647。可以将该字段设为自动增长字段
Long Binary (OLE Object)	极大值字段，可存储二进制数据结构如图形或文件 嵌入到数据库中的OLE对象，最大可为1GB
Memo	极大值字段，可存储65 535个字符。无需事先声明该字段的长度
Single type	4字节，单精度数值数据
Text	固定长度的数值类型，当你声明其数据类型时需指明该字段的大小。 Text字段的长度可从1到255个字符
VarBinary	一段可变长的二进制数据（与ODBCDirect一同使用）

Visual Basic的变量数据类型与数据库字段的数据类型之间不存在一对一的映射关系。例如，不能将数据库字段设为用户自定义类型或Visual Basic风格的Object变量，同样，如果用Microsoft Access创建与VB应用程序一起使用的数据库，要注意有些VB应用程序中的有些数据类型在Microsoft Access表设计器中是不存在的。这是由于Visual Basic支持数据库程序设计而不支持在Microsoft Access中创建的这些表。

1.2.5 创建数据库规划

尽管创建一系列的表和字段是确定数据库结构的一个好的方式，你或许还希望能有一种以图形格式来表示表和字段的方式。这样不但可以看到哪些表和字段是可用的，而且它们如何联系也一目了然。要达到此目的，可以创建一个规划。

规划（Schema）是数据库的导游图。规划以图形方式勾勒出数据库中的所有的表、字段