

数据库
混合编程
丛书

Visual Basic + SQL Server 数据库应用系统 开发与实例

李晓黎 张 巍 编著

- 包含大量实用小技巧，重点突出，便于灵活掌握
- 包含数据库的管理、开发和实例，内容全面，一步到位
- 分析详细，实用性强



人民邮电出版社
POSTS & TELECOMMUNICATIONS PRESS

数据库混合编程丛书

Visual Basic + SQL Server 数据库 应用系统开发与实例

李晓黎 张 巍 编著

人民邮电出版社

图书在版编目 (CIP) 数据

Visual Basic+SQL Server 数据库应用系统开发与实例 / 李晓黎, 张巍编著.

—北京: 人民邮电出版社, 2003.8

(数据库混合编程丛书)

ISBN 7-115-11228-2

I. V... II. ①李... ②张... III. ①BASIC 语言—程序设计 ②关系数据库—数据库管理系统, SQL Server IV. ①TP312 ②TP311.138

中国版本图书馆 CIP 数据核字 (2003) 第 027356 号

内 容 提 要

本书介绍了 Visual Basic 和 SQL Server 2000 的必备知识, 并完整地介绍了几个使用 Visual Basic 开发 SQL Server 2000 数据库应用程序的实例, 包括学生档案管理系统、软件项目开发管理系统、旅游资源及线路管理系统和企业库存管理系统, 既可以作为独立的系统运行, 也可以作为应用软件的一个模块, 具有很强的实用性和可移植性。本书最后还给出了程序优化和发布方法。

本书适合 SQL Server 2000 数据库管理员和应用程序开发人员, 以及对相关技术感兴趣的读者阅读。读者只要具备基本的 Visual Basic 编程知识, 无需有任何经验, 就可以按照书中的实例开发数据库应用程序。

数据库混合编程丛书

Visual Basic+SQL Server 数据库应用系统开发与实例

◆ 编 著 张晓黎 张 巍

责任编辑 刘 浩

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

读者热线 010-67132692

北京汉魂图文设计有限公司制作

北京顺义振华印刷厂印刷

新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 26.5

字数: 640 千字

2003 年 8 月第 1 版

印数: 1-6 000 册

2003 年 8 月北京第 1 次印刷

ISBN7-115-11228-2/TP·3423

定价: 42.00 元 (附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

前 言

随着数据库应用技术的发展,越来越多的软件开发人员和爱好者希望了解并掌握管理开发数据库应用程序的方法。而目前市场中,偏重于数据库管理的图书居多,数据库应用系统开发方面的图书所占的比例不大。另外,很多数据库开发类图书的内容偏重于程序设计方法的介绍,所列举的示例程序实用性不强。这些都会影响读者的学习效果,在实际开发过程中,遇到问题仍然会感到无所适从。

本书在编写过程中,考虑到以上问题,作者把多年管理和开发 SQL Server 2000 数据库的经验编写成本书,详尽介绍了使用 Visual Basic 开发 SQL Server 2000 数据库应用程序所必备的管理及开发技术。

本书最大的特点在于实用性。书中选择的 4 个实例都是作者在多年的应用系统开发过程中,积累的有很强实用价值的程序。它们都是当前应用比较广泛的数据库应用系统,既可以作为独立的系统使用,也可以作为大型综合数据库管理系统的一个模块。

在应用系统开发过程中,经验是非常重要的。通过阅读本书实例,读者可以分享作者在开发过程中的技巧和经验教训,避免在学习过程中走不必要的弯路,轻松拥有自己开发的数据库应用程序。对于急需开发应用程序,而又没有时间系统学习的读者,则可以对实例稍加修改,直接使用。

本书最后还介绍了开发数据库应用系统的几个非常常用的技术,包括界面优化、权限控制、日志管理、制作帮助文件和安装程序等。通过本书,读者将对数据库应用系统开发有一个全面系统地认识。

随书附赠的光盘中,提供了本书介绍实例程序的全部代码。为了便于读者使用光盘中的实例,作者在光盘中按照章名创建每一章中实例的目录。例如,第 1 章中出现的实例将被存储在 X:\Chp1 目录下(X: 是您的计算机中光盘驱动器号)。如果在一章中存在多个实例程序,它们将被存储在不同的目录下。目录名使用简单的数字 1, 2.....表示。例如,第 1 章的第一个实例存储在 X:\Chp1\1 目录下,第二个实例存储在 X:\Chp1\2 目录下。在相关章节介绍到一个实例时,会明确地说明此实例的存储目录。

需要说明的是,光盘中并不包括 SQL Server 2000 数据库的安装程序。

本书在编写过程中,充分考虑了初学者可能遇到的问题,以程序开发的实用技术为中心,注意提高读者的系统设计和开发能力。读者只要具备基本的 Visual Basic 编程知识,无需有任何编程经验,即可轻松掌握书中内容。

参加本书编写的还有:李树有、刘熙芳、张志祥、刘熙凤、张晓辉、李晓喆等。由于作者水平有限,本书中难免有不足之处,敬请广大读者批评指正。

编者

2003.3

目 录

第 1 章 SQL Server 2000 数据库对象管理	1
1.1 数据库管理	1
1.1.1 系统数据库	1
1.1.2 创建数据库	2
1.1.3 删除数据库	4
1.2 表管理	4
1.2.1 SQL Server 2000 数据类型	4
1.2.2 表设计器	8
1.2.3 数据库设计器	9
1.2.4 查看表中的数据	12
1.2.5 设置表中的键	16
1.2.6 删除表	17
1.3 视图管理	18
1.3.1 视图的基本概念	18
1.3.2 创建视图	19
1.3.3 删除视图	22
1.4 服务器与客户端配置	22
1.4.1 启动、暂停和停止 SQL Server	23
1.4.2 配置 ODBC 数据源	24
第 2 章 Transact-SQL 语言简介	28
2.1 Transact-SQL 概述	28
2.1.1 Transact-SQL 的语法规则	28
2.1.2 Transact-SQL 的语法元素	29
2.1.3 osql 实用工具	30
2.2 数据库对象操作语句	32
2.2.1 创建数据库语句 CREATE DATABASE	32
2.2.2 删除数据库语句 DROP DATABASE	37
2.2.3 创建表语句 CREATE TABLE	37
2.2.4 删除表语句 DROP TABLE	41
2.2.5 创建视图语句 CREATE VIEW	41
2.2.6 删除视图语句 DROP VIEW	42
2.3 数据查询语句	42
2.3.1 SELECT 语句语法简介	43

2.3.2	检验查询结果的方法	44
2.3.3	最基本的 SELECT 语句	47
2.3.4	设置查询条件	50
2.3.5	对查询结果排序	50
2.3.6	使用聚合函数	51
2.3.7	对查询结果分组	52
2.3.8	指定组或聚合的搜索条件	54
2.3.9	连接查询	54
2.3.10	子查询	56
2.3.11	合并查询	57
2.3.12	保存查询结果	57
2.4	数据更新语句	58
2.4.1	插入数据语句 INSERT	58
2.4.2	修改数据语句 UPDATE	60
2.4.3	删除数据语句 DELETE	62
第 3 章	Visual Basic 数据库工程实用管理	63
3.1	Visual Basic 访问 SQL Server 数据库概述	63
3.1.1	Visual Basic 数据访问技术综述	63
3.1.2	ADO 数据访问技术简介	64
3.1.3	ODBC 数据库访问技术简介	67
3.2	Visual Basic 工程管理	71
3.2.1	工程的文件组成和管理	72
3.2.2	设置工程选项	73
3.3	Visual Basic 数据库访问控件	73
3.3.1	Data 控件	73
3.3.2	Data 控件的示例程序	75
3.3.3	ADO Data 控件	78
3.3.4	ADO Data 控件的示例程序	79
3.3.5	DataList 控件 / DataCombo 控件	81
3.3.6	DataList 控件 / DataCombo 控件的示例程序	82
3.3.7	DataGrid 控件	86
3.3.8	DataGrid 控件的示例程序	91
3.3.9	MSChart 控件	95
3.3.10	MSChart 控件的示例程序	98
3.4	Visual Basic 功能模块规划	104
3.4.1	窗体、模块和类模块的功能划分	104
3.4.2	Const 模块内容介绍	107
3.4.3	DbFunc 模块内容介绍	107
3.4.4	GeneralFunc 模块内容介绍	113

3.5	Visual Basic 数据库工程开发过程	118
3.5.1	系统功能分析和总体设计	118
3.5.2	数据库准备	119
3.5.3	工程管理	120
3.5.4	添加类模块	120
3.5.5	添加模块	121
3.5.6	添加并编辑窗体	121
第 4 章	学生档案管理系统	122
4.1	系统设计	122
4.1.1	系统功能描述	122
4.1.2	功能模块划分	123
4.1.3	系统流程分析	125
4.2	数据库设计	126
4.2.1	创建数据库	126
4.2.2	数据库逻辑结构设计	127
4.2.3	创建表的脚本文件	131
4.3	设计工程框架	134
4.3.1	创建工程	134
4.3.2	添加模块	134
4.3.3	添加类模块	135
4.3.4	创建数据源	145
4.4	系统主界面设计	145
4.4.1	设计主界面	145
4.4.2	在主界面中添加代码	146
4.5	登录模块设计	147
4.5.1	设计登录窗体	147
4.5.2	在登录窗体中添加代码	147
4.6	院系管理模块设计	149
4.6.1	使用 TreeView 控件管理院系数据的方法	149
4.6.2	设计编辑院系信息的窗体	151
4.6.3	设计院系管理窗体	154
4.6.4	设计选择院系班级窗体	158
4.6.5	在主界面中增加院系管理代码	158
4.7	学生基本信息管理模块设计	159
4.7.1	设计选择日期窗体	159
4.7.2	设计选择学生窗体	160
4.7.3	设计编辑学生基本信息的窗体	163
4.7.4	设计学生信息管理窗体	169
4.8	学生照片管理模块设计	172

4.8.1	ImageFunc 模块	172
4.8.2	设计照片管理窗体	174
4.8.3	在学生信息管理窗体中添加照片管理代码	176
4.9	学生教育经历管理模块设计	176
4.9.1	设计学生教育经历编辑模块	177
4.9.2	设计学生教育经历管理模块	178
4.10	学生奖惩管理模块设计	181
4.10.1	设计编辑奖惩信息的窗体	181
4.10.2	设计奖惩信息管理窗体	181
4.11	学籍变动管理模块设计	183
4.11.1	设计编辑学籍变动信息的窗体	183
4.11.2	设计学籍变动管理窗体	185
4.12	课程设置管理模块设计	187
4.12.1	设计课程信息编辑窗体	188
4.12.2	设计课程信息管理窗体	189
4.13	学生成绩管理模块设计	190
4.13.1	设计学生成绩信息编辑窗体	190
4.13.2	设计学生成绩信息管理窗体	192
4.14	用户管理模块设计	193
4.14.1	设计编辑用户信息的窗体	193
4.14.2	设计用户管理窗体	195
4.14.3	在主界面中增加用户管理代码	197
第 5 章	软件项目开发管理系统	198
5.1	系统设计	198
5.1.1	系统功能描述	198
5.1.2	功能模块划分	200
5.1.3	系统流程分析	201
5.2	数据库设计	202
5.2.1	创建数据库	202
5.2.2	数据库逻辑结构设计	204
5.2.3	创建表的脚本文件	207
5.3	设计工程框架	210
5.3.1	创建工程	211
5.3.2	添加模块	211
5.3.3	添加类模块	212
5.3.4	创建数据源	220
5.4	系统主界面设计	220
5.4.1	设计主界面	220
5.4.2	在主界面中添加代码	222

5.5	登录模块设计	222
5.5.1	设计登录窗体	222
5.5.2	在登录窗体中添加代码	222
5.6	项目管理模块设计	222
5.6.1	设计编辑项目信息的窗体	223
5.6.2	设计编辑项目分配信息的窗体	225
5.6.3	设计项目管理窗体	226
5.6.4	在主界面中增加项目管理代码	232
5.6.5	设计选择项目的窗体	232
5.7	项目实施管理模块设计	234
5.7.1	设计添加项目组成员信息的窗体	234
5.7.2	设计编辑功能模块信息的窗体	235
5.7.3	设计编辑工作报告信息的窗体	238
5.7.4	设计项目实施管理窗体	239
5.8	文档管理模块设计	241
5.8.1	DocFunc 模块	241
5.8.2	设计文档基本信息编辑窗体	244
5.8.3	设计文档管理窗体	245
5.8.4	在主界面中添加文档管理代码	247
5.9	测试管理模块设计	247
5.9.1	设计报错问题编辑窗体	248
5.9.2	设计报错问题管理窗体	250
5.10	项目总结管理模块设计	251
5.10.1	设计项目总结信息编辑窗体	252
5.10.2	设计项目总结信息编辑窗体	252
5.11	用户管理模块设计	254
5.11.1	设计编辑用户信息的窗体	254
5.11.2	设计用户管理窗体	256
5.11.3	在主界面中增加用户管理代码	258
第 6 章	旅游资源及线路管理系统	260
6.1	系统设计	260
6.1.1	系统功能描述	260
6.1.2	功能模块划分	261
6.1.3	系统流程分析	263
6.2	数据库设计	263
6.2.1	创建数据库	263
6.2.2	数据库逻辑结构设计	265
6.2.3	创建表的脚本文件	270
6.3	设计工程框架	275

6.3.1	创建工程	276
6.3.2	添加模块	276
6.3.3	添加类模块	277
6.3.4	创建数据源	290
6.4	系统主界面设计	290
6.4.1	设计主界面	290
6.4.2	在主界面中添加代码	292
6.5	登录模块设计	292
6.5.1	设计登录窗体	292
6.5.2	在登录窗体中添加代码	293
6.6	旅游地域资源管理模块设计	293
6.6.1	设计旅游地域资源编辑窗体	293
6.6.2	设计旅游地域资源管理窗体	294
6.6.3	在主界面中增加地域管理代码	298
6.7	景点资源管理模块设计	299
6.7.1	设计景点资源编辑窗体	299
6.7.2	设计景点资源管理窗体	300
6.8	宾馆资源管理模块设计	303
6.8.1	设计宾馆资源编辑窗体	303
6.8.2	设计宾馆资源管理窗体	304
6.9	餐厅资源管理模块设计	304
6.9.1	设计餐厅资源编辑窗体	305
6.9.2	设计餐厅资源管理窗体	305
6.10	娱乐资源管理模块设计	305
6.10.1	设计娱乐资源编辑窗体	306
6.10.2	设计娱乐项目管理窗体	306
6.10.3	设计娱乐资源管理窗体	308
6.11	火车资源管理模块设计	308
6.11.1	设计火车资源编辑窗体	309
6.11.2	设计火车资源管理窗体	309
6.12	飞机资源管理模块设计	309
6.12.1	设计飞机资源编辑窗体	310
6.12.2	设计飞机资源管理窗体	313
6.13	旅游线路管理模块设计	313
6.13.1	设计旅游线路管理窗体	314
6.13.2	设计旅游线路编辑窗体	315
6.13.3	设计线路中旅游资源信息的编辑窗体	322
6.14	用户管理模块设计	325
6.14.1	设计编辑用户信息的窗体	326

6.14.2	设计用户管理窗体	327
6.14.3	在主界面中增加用户管理代码	329
第 7 章	企业库存管理系统	330
7.1	系统设计	330
7.1.1	系统功能描述	330
7.1.2	功能模块划分	331
7.1.3	系统流程分析	332
7.2	数据库设计	333
7.2.1	创建数据库	333
7.2.2	数据库逻辑结构设计	334
7.2.3	创建表的脚本文件	338
7.3	设计工程框架	341
7.3.1	创建工程	341
7.3.2	添加模块	341
7.3.3	添加类模块	342
7.3.4	创建数据源	352
7.4	系统主界面设计	352
7.4.1	设计主界面	352
7.4.2	在主界面中添加代码	354
7.5	登录模块设计	354
7.5.1	设计登录窗体	354
7.5.2	在登录窗体中添加代码	355
7.6	客户管理模块设计	355
7.6.1	设计客户信息编辑窗体	355
7.6.2	设计客户信息管理窗体	358
7.7	仓库管理模块设计	361
7.7.1	设计仓库信息编辑窗体	361
7.7.2	设计仓库信息管理窗体	363
7.8	用户管理模块设计	364
7.9	产品管理模块设计	364
7.9.1	设计产品类目管理模块	364
7.9.2	设计产品信息管理模块	367
7.10	库存操作管理模块设计	368
7.10.1	设计入库操作管理模块	368
7.10.2	设计出库操作管理模块	371
7.10.3	设计库存盘点操作管理模块	372
7.11	库存警示管理模块设计	372
7.11.1	设计产品数量报警管理模块	372
7.11.2	设计产品失效报警管理模块	374

7.12 统计查询管理模块设计	375
7.12.1 设计产品出入库统计表窗体	375
7.12.2 设计库存产品流水统计表窗体	377
第 8 章 应用程序的优化和发布	379
8.1 应用程序界面优化	379
8.1.1 为应用程序添加皮肤	379
8.1.2 应用程序中的动画处理	382
8.2 应用程序的权限控制	384
8.2.1 权限设置实现方法	385
8.2.2 学生档案管理系统权限控制	385
8.3 应用程序的日志管理	389
8.3.1 日志管理实现方法	389
8.3.2 学生档案管理系统日志管理	390
8.4 创建帮助文件	394
8.4.1 制作 CHM 帮助文件	394
8.4.2 在 Visual Basic 中访问 CHM 帮助文件	400
8.5 制作安装程序	401
8.5.1 打包和展开向导	401
8.5.2 安装程序制作过程	402
8.5.3 应用程序安装过程	405
附录 A 下载 SQL Server 2000 评估版	407
附录 B 推荐网站	409

第 1 章 SQL Server 2000 数据库对象管理

SQL Server 2000 继承了数据库软件设计和引擎方面的优势，已经成为新一代网络应用程序的最佳数据库和分析解决方案之一。作为 Microsoft .NET Enterprise Servers 的核心组件，它可以有效地减少将电子商务、在线商务和数据仓储应用程序推向市场的时间，同时对大多数需要的环境提供可靠的支持。SQL Server 2000 对 XML 和 HTTP 提供充分的支持，提供分割装载和确保时间的特性，提供先进的管理和协调机制，可以自动管理线程任务，降低总体开销。

1.1 数据库管理

在 SQL Server 2000 中，数据库由包含数据的表集合和其他对象（如视图、索引等）组成，目的是为执行与数据有关的活动提供支持。在这一节中，将介绍如何管理 SQL Server 2000 的数据库。

1.1.1 系统数据库

SQL Server 2000 系统有 4 种系统数据库：

- master 数据库

master 数据库记录 SQL Server 系统的所有系统级别信息，它记录所有的登录账户和系统配置设置。master 数据库是这样一个数据库，它记录所有其他的数据库信息，其中包括数据库文件的位置。master 数据库记录 SQL Server 的初始化信息，它始终提供一个可用的最新 master 数据库备份。

- tempdb 数据库

tempdb 数据库保存所有的临时表和临时存储过程。它还满足任何其他的临时存储要求，例如存储 SQL Server 生成的工作表。tempdb 数据库是全局资源，所有连接到系统的用户的临时表和存储过程都存储在该数据库中。tempdb 数据库在 SQL Server 每次启动时都重新创建，因此该数据库在系统启动后总是干净的。临时表和存储过程在连接断开时自动除去，而且当系统关闭后将没有任何连接处于活动状态，因此 tempdb 数据库中没有任何内容会从 SQL Server 的一个会话保存到另一个会话。

默认情况下，在 SQL Server 在运行时 tempdb 数据库会根据需要自动增长。不过，与其他数据库不同，每次启动数据库引擎时，它会重置为其初始大小。如果为 tempdb 数据库定义的大小较小，则每次重新启动 SQL Server 时，将 tempdb 数据库的大小自动增加到支持工作负荷所需的大小，这一工作可能会成为系统处理负荷的一部分。为避免这种开销，可以使用 ALTER DATABASE 增加 tempdb 数据库的大小。

- model 数据库

model 数据库用于在系统上创建的所有数据库的模板。当发出 CREATE DATABASE 语句时,新数据库的第一部分通过复制 model 数据库中的内容创建,剩余部分由空页填充。由于 SQL Server 每次启动时都要创建 tempdb 数据库,model 数据库必须一直存在于 SQL Server 系统中。

- msdb 数据库

msdb 数据库供 SQL Server 代理程序调度警报和作业时使用。

1.1.2 创建数据库

对于使用 SQL Server 2000 的用户来说,创建数据库是最基本的操作了。在创建数据库之前,必须先确定数据库的名称、所有者(创建数据库的用户)、大小,以及用于存储该数据库的文件和文件组。

可以使用 3 种文件类型来存储数据库:

- 主文件 主文件包含数据库的启动信息。主文件还用于存储数据。每个数据库都包含一个主文件。
- 次要文件 次要文件含有不能置于主要数据文件中的所有数据。如果主文件足够大,能够容纳数据库中的所有数据,则该数据库不需要次要数据文件。有些数据库可能非常大,因此需要多个次要数据文件,或可能在各自的磁盘驱动器上使用次要文件,以便在多个磁盘上存储数据。
- 事务日志 这些文件包含用于恢复数据库的日志信息。每个数据库必须至少有一个事务日志文件(但是可以有多个)。日志文件最小为 512 KB。

打开 SQL Server 企业管理器,可以按照下面的方法来创建数据库。

1. 展开服务器组,然后展开服务器实例。

2. 右击“数据库”,然后单击“新建数据库”命令,打开“数据库属性”窗口,如图 1.1 所示。

3. 键入新数据库的名称,假定数据库名为 newdb。默认情况下,系统自动地使用指定的数据库名作为前缀创建主数据库和事务日志文件,例如: newdb_Data.mdf 和 newdb_Log.ldf。

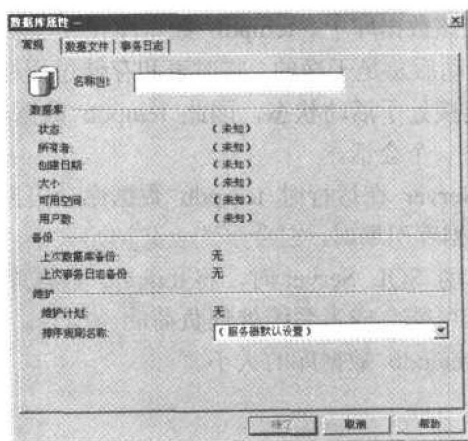


图 1.1 数据库属性

4. 要更改新建主数据库文件的默认值，单击“数据文件”选项卡，如图 1.2 所示。要更改“文件名”、“位置”、“初始大小 (MB)”和“文件组”等列的默认值，单击要更改的适当单元格，再键入新值。

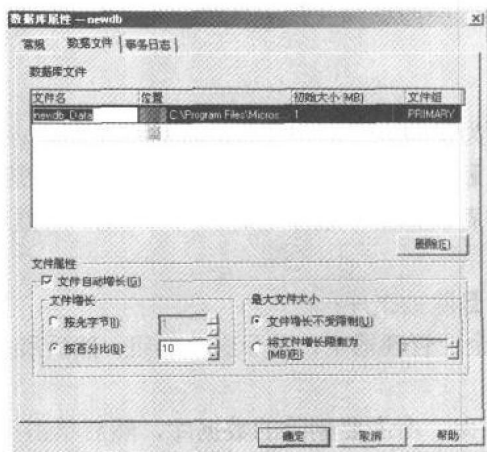


图 1.2 修改主数据库文件的默认值

要指定文件如何增长，请从以下选项中选择：

- 当需要更多的数据空间时，若要允许当前选定的文件增长，则选择“文件自动增长”命令；
- 要指定文件按固定步长增长，选择“按兆字节”命令并指定一个值；
- 若要指定文件按当前大小的百分比增长，选择“按百分比”命令，并且指定一个值。

建议设置数据库文件的最大文件大小。这样做可以防止文件在添加数据时无限制增大，以至用尽整个磁盘空间。选择“将文件增长限制为 (MB)”命令，并设置文件大小的限度。

5. 若要更改新建事务日志文件的默认值，单击“事务日志”选项卡，方法与第 4 步相同。

一切都设置完成后，单击“确定”按钮，可以看到数据库 newdb 出现在左侧的数据库列表中。单击选中 newdb，就可以看到 SQL Server 2000 数据库的组成，如图 1.3 所示。

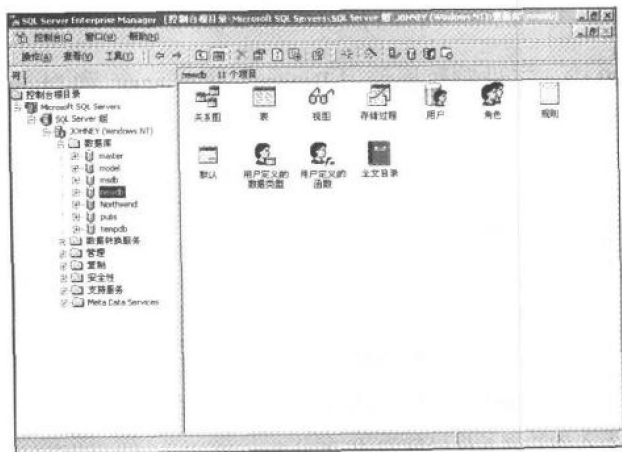


图 1.3 新建的数据库 newdb

在本书的后面部分，数据库 newdb 将作为主要的示例数据库。

创建数据库之后，建议创建一个 master 数据库的备份。因为创建数据库将更新 master 中的系统表，如果将原有的 master 数据库备份还原，则从上次备份 master 之后新建的所有数据库信息都将从系统表中丢失，因而可能导致出现错误信息。

1.1.3 删除数据库

当不再需要数据库时，为了节省空间，可以将该数据库删除。数据库删除之后，文件及其数据都从服务器上的磁盘中删除。一旦删除数据库，它即被永久删除，并且不能进行检索，除非使用以前的备份。建议读者在数据库删除之后备份 master 数据库，因为删除数据库将更新 master 中的系统表。

读者可以按照以下步骤删除数据库：

1. 在 SQL Server 2000 企业管理器中，展开服务器组，然后展开要删除数据库所在的服务器实例。
2. 展开“数据库”文件夹，右击要删除的数据库，然后单击“删除”命令。
3. 在“确认删除”对话框中，单击“是”按钮。

1.2 表管理

表是包含数据库中所有数据的数据库对象。表定义为列的集合。与电子表格相似，数据在表中是按行和列的格式组织排列的。每行代表惟一的一条记录，而每列代表记录中的一个域。

1.2.1 SQL Server 2000 数据类型

设计数据库的关键就是设置表。首先应确定需要什么样的表，各表中都有哪些数据以及各个表的存取权限等。

设计表时首先要为每列指派数据类型。数据类型定义了各列所允许的数据值。下面介绍 SQL Server 2000 的基本数据类型，数据类型不区分大小写。

1. 二进制数据

在 SQL Server 2000 中，二进制数据使用 Binary、Varbinary 和 Image 数据类型存储。具体说明如表 1.1 所示。

表 1.1 二进制数据类型说明

数据类型	描述
Binary	固定长度的二进制数据类型，Binary [(n)] 表示固定长度的 n 个字节二进制数据。 n 必须从 1~8 000，存储空间大小为 $n+4$ 字节
Varbinary	变长的二进制数据类型，Varbinary [(n)] 表示 n 个字节变长二进制数据。 n 必须从 1~8 000。存储空间大小为实际输入数据长度+4 个字节，而不是 n 个字节。输入的数据长度可能为 0 字节
Image	可以用来存储超过 8 KB 的可变长度的二进制数据



小技巧

在设计表结构时,如果需要存储的二进制数据长度小于 8KB,则使用 Binary 或 Varbinary。使用 Varbinary 数据类型可以按照数据的实际长度进行存储,从而节省存储空间。当数据长度超过 8KB 时,使用 Image 数据类型进行存储。Image 是一个非常有用的数据类型,很多读者可能会以为 Image 是专门用来存储图像数据的,其实它可以存储如 Microsoft Word 文档、Microsoft Excel 电子表格、包含位图的图像、GIF 文件和 JPEG 文件等多种格式的数据。本书第 4 章将介绍如何使用 Image 数据类型存储图像数据。

2. 字符数据

字符数据由字母、符号和数字组成。例如,有效字符数据包括“123”、“Johney”和“(0*&(%B99nh jkJ”。在 SQL Server 2000 中,字符数据使用 Char、Varchar 和 Text 数据类型存储。具体说明如表 1.2 所示。

表 1.2

字符数据类型说明

数据类型	描述
Char	固定长度的字符数据类型。Char(<i>n</i>)表示长度为 <i>n</i> 个字节的固定长度且非 Unicode 的字符数据。 <i>n</i> 必须是一个介于 1~8 000 之间的数值。存储大小为 <i>n</i> 个字节
Varchar	是非固定长度的字符数据类型。Varchar(<i>n</i>)表示长度为 <i>n</i> 个字节的可变长度且非 Unicode 的字符数据。 <i>n</i> 必须是一个介于 1~8 000 之间的数值。存储大小为输入数据的字节的实际长度,而不是 <i>n</i> 个字节。所输入的数据字符长度可以为零
Text	数据类型的列可用于存储大于 8KB 的 ASCII 字符



小技巧

在设计表结构时,如果需要存储的字符数据长度小于 8KB,则使用 Char 或 Varchar。使用 Varchar 数据类型可以按照数据的实际长度进行存储,从而节省存储空间。当数据长度超过 8KB 时,使用 text 数据类型进行存储。例如,由于 HTML 文档均由 ASCII 字符组成,而且一般长于 8KB,所以通常把 HTML 文档在 SQL Server 中存储在 text 列中。

3. Unicode 数据

在 SQL Server 2000 中,传统上非 Unicode 数据类型允许使用由特定字符集定义的字符。字符集是在安装 SQL Server 时选择的,不能更改。使用 Unicode 数据类型,列可存储由 Unicode 标准定义的任何字符,包含由不同字符集定义的所有字符。Unicode 数据类型需要相当于非 Unicode 数据类型两倍的存储空间。

Unicode 数据使用 SQL Server 中的 Nchar、Nvarchar 和 Ntext 数据类型进行存储。具体说明如表 1.3 所示。

表 1.3

Unicode 数据类型说明

数据类型	描述
Nchar	Nchar 是固定长度 Unicode 数据的数据类型, Nchar(<i>n</i>)表示包含 <i>n</i> 个字符的固定长度 Unicode 字符数据。 <i>n</i> 的值必须介于 1~4 000 之间。存储大小为 <i>n</i> 字节的两倍