

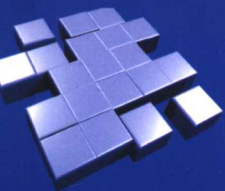


# Delphi + SQL Server

## 数据库应用实例

# 完全解析

伊文敏 刘峰 王颖 编著



- 小区物业管理系统
- 学生公寓管理系统
- 学生学籍管理系统
- 车辆调度管理系统

- 餐饮业收费管理系统
- 毕业生就业管理系统
- 计件工资管理系统

数据库、源代码、PPT演示文件



人民邮电出版社  
POSTS & TELECOM PRESS



# Delphi + SQL Server

数据库应用案例

## 完全解析

作者：陈国平 编著



本书主要介绍 Delphi 与 SQL Server 数据库的应用案例，包括：Delphi 与 SQL Server 数据库的连接、Delphi 与 SQL Server 数据库的交互、Delphi 与 SQL Server 数据库的查询、Delphi 与 SQL Server 数据库的更新、Delphi 与 SQL Server 数据库的备份与恢复等。

本书适合 Delphi 与 SQL Server 数据库的初学者阅读，也可作为 Delphi 与 SQL Server 数据库的进阶参考书。



清华大学出版社



# Delphi + SQL Server 完全解析

数据库应用实例

伊文敏 刘峰 王颖 编著

人民邮电出版社

## 图书在版编目 (CIP) 数据

Delphi+SQL Server 数据库应用实例完全解析 / 伊文敏, 刘峰, 王颖编著.

—北京: 人民邮电出版社, 2006.12

ISBN 7-115-15356-6

I. D... II. ①伊... ②刘... ③王... III. ①软件工具—程序设计②关系数据库—数据库管理系统, SQL Server IV. TP311.56 ②TP311.138

中国版本图书馆 CIP 数据核字 (2006) 第 118420 号

## 内 容 提 要

本书详尽介绍了使用 Delphi 和 SQL Server 开发数据库应用程序所必备的管理及开发技术, 包括 SQL Server 数据库的管理、SQL 语句和 ADO 数据库访问技术。这些技术会在本书的 7 个实例中得到应用。通过技术与实例相结合的方式, 读者可以很快掌握数据库应用系统开发的过程和方法, 并将所学技术应用到实际工作中。与其他实例类图书相比, 本书更注重系统开发过程的完整性, 还为每个实例制作了一个 PPT 文件, 详细介绍和演示实例所需的实用技术和开发过程, 更加方便读者学习和理解。

本书案例贴近实际, 内容精炼, 适合大中专院校的师生和软件开发人员参考使用。

### Delphi+SQL Server 数据库应用实例完全解析

- ◆ 编 著 伊文敏 刘 峰 王 颖  
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>  
北京密云春雷印刷厂印刷  
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16  
印张: 22.75  
字数: 555 千字 2006 年 12 月第 1 版  
印数: 1—5 000 册 2006 年 12 月北京第 1 次印刷

ISBN 7-115-15356-6/TP • 5743

定价: 38.00 元 (附光盘)

读者服务热线: (010) 67132692 印装质量热线: (010) 67129223

# 前 言

数据库应用系统开发是当前最流行的实用技术之一，大多数大型应用系统都需要后台数据库的支持。使用 SQL Server 作为后台数据库，能够降低成本，方便管理。而 Delphi 是非常流行的可视化开发语言，具有简单易学、灵活方便和易于扩充等特点。本书详尽介绍了使用 Delphi 和 SQL Server 开发数据库应用程序所必备的管理及开发技术。

本书介绍了开发数据库应用系统必须掌握的技术，包括 SQL Server 数据库的管理、SQL 语句和 ADO 数据库访问技术。这些技术会在本书的 7 个实例中得到应用。通过技术与实例相结合的方式，读者可以很快掌握数据库应用系统开发的过程和方法，并将所学技术应用到实际工作中。书中选择的 7 个实例都是有很强实用价值的程序，是当前应用比较广泛的数据库应用系统。

与其他实例类图书相比，本书更注重系统开发过程的完整性。笔者在介绍数据库应用系统实例的过程中，并不是简单讲解实例功能和代码，而是把每个实例作为一个项目来实现。书中的每个实例都按照开发项目的步骤进行介绍，包括系统需求分析、系统体系结构设计、开发工具的选择、开发实例所需的实用技术、数据库结构设计、工程框架设计以及各功能模块的设计等。按照此步骤阅读本书实例，读者可以体验一个项目从无到有、逐步实现的过程。而在实际系统开发过程中，认真完成每个步骤的工作，是项目能够顺利完成的保障。

在本书附赠的光盘中，除了提供各实例的源代码外，还为每个实例制作了一个 PPT 文件，详细介绍和演示实例所需的实用技术和开发过程，更加方便读者学习和理解。

本书以程序开发的实用技术为中心，注重提高读者的系统设计和开发能力。读者只要具备基本的 Delphi 编程知识，即可轻松掌握书中内容。

本书第 1 章由王颖编写；第 2 章至第 4 章由伊文敏编写；第 5 章和第 6 章由刘峰编写；第 7 章和第 8 章由张巍编写。参加资料整理和编写的还有全杉、李敏业、高英、董婷、翁文勇、李斐、张巍、李晓黎、侯志荣、许云锴、赵亮、江贺、孙卫强、吴明桥、李响、黄志斌、刘蔚、洪宝珠、冯冲、胡萍、李宁、冯静芳、田文永、原岳等。由于作者水平有限，书中难免有不足之处，敬请广大读者批评指正（E-mail: [book\\_better@sohu.com](mailto:book_better@sohu.com)）。

编者

2006 年 11 月

## 光 盘 说 明

随书附赠的光盘中，提供了本书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节实例的目录。例如，第 2 章中出现的实例将被存储在“X:\第 2 章”目录下（X: 是光盘驱动器号）。在第 1 章的目录下，包括一个实例代码目录和数据库备份文件。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。

随书附赠的光盘中，还为每个实例制作了一个 PPT 文件，详细介绍和演示实例所需的实用技术和开发过程。例如，第 2 章实例的 PPT 文件名为“第 2 章.PPT”，依此类推。

光盘中不包括 SQL Server 的安装程序。

# 目 录

|                              |    |
|------------------------------|----|
| 第 1 章 数据库编程实用技术.....         | 1  |
| 1.1 SQL Server 数据库管理.....    | 1  |
| 1.1.1 企业管理器.....             | 1  |
| 1.1.2 创建数据库.....             | 2  |
| 1.1.3 创建表.....               | 2  |
| 1.1.4 创建视图.....              | 5  |
| 1.1.5 备份和还原数据库.....          | 6  |
| 1.2 常用 SQL 语句介绍.....         | 9  |
| 1.2.1 执行 SQL 语句的环境.....      | 9  |
| 1.2.2 SELECT 语句.....         | 9  |
| 1.2.3 INSERT 语句.....         | 13 |
| 1.2.4 UPDATE 语句.....         | 13 |
| 1.2.5 DELETE 语句.....         | 13 |
| 1.3 Delphi 中 ADO 技术的使用.....  | 14 |
| 1.3.1 ADO 编程的基本方法.....       | 15 |
| 1.3.2 ADO 连接数据库.....         | 17 |
| 1.3.3 读取数据库记录.....           | 18 |
| 1.3.4 存取字段数据的方法.....         | 19 |
| 1.4 Delphi 中数据库开发常用组件介绍..... | 20 |
| 1.4.1 TDataSource 组件.....    | 21 |
| 1.4.2 TDBGrid 组件.....        | 22 |
| 1.4.3 TADOConnection 组件..... | 23 |
| 1.4.4 TADOQuery 组件.....      | 26 |
| 1.4.5 TADOCommand 组件.....    | 27 |
| 1.5 数据库应用系统工程规划.....         | 28 |
| 1.5.1 数据库项目开发实施过程.....       | 28 |
| 1.5.2 功能模块划分.....            | 30 |
| 第 2 章 小区物业管理系统.....          | 34 |
| 2.1 概述.....                  | 34 |
| 2.2 系统分析与总体设计.....           | 34 |
| 2.2.1 系统需求分析.....            | 34 |
| 2.2.2 系统体系结构设计.....          | 34 |
| 2.2.3 开发工具的选择.....           | 37 |
| 2.2.4 开发实例所需的实用技术.....       | 37 |
| 2.2.5 数据库结构设计.....           | 37 |
| 2.3 设计工程框架.....              | 40 |
| 2.3.1 工程中的模块.....            | 40 |
| 2.3.2 添加类模块.....             | 42 |
| 2.4 系统主界面和登录窗体设计.....        | 46 |
| 2.4.1 设计主界面.....             | 47 |
| 2.4.2 设计登录窗体.....            | 49 |

|        |               |     |
|--------|---------------|-----|
| 2.5    | 楼宇信息管理模块设计    | 51  |
| 2.5.1  | 使用 DBGrid 控件  | 51  |
| 2.5.2  | 设计添加窗体        | 52  |
| 2.5.3  | 设计楼宇信息窗体      | 55  |
| 2.6    | 业主基本信息管理模块设计  | 59  |
| 2.6.1  | 设计业主基本信息编辑窗体  | 59  |
| 2.6.2  | 设计业主信息管理窗体    | 62  |
| 2.7    | 费用设置管理模块设计    | 68  |
| 2.7.1  | 设计费用设置窗体      | 68  |
| 2.7.2  | 添加功能代码        | 69  |
| 2.8    | 缴费登记管理模块设计    | 72  |
| 2.8.1  | 设计缴费管理窗体      | 73  |
| 2.8.2  | 添加功能代码        | 76  |
| 2.9    | 车位管理模块设计      | 78  |
| 2.9.1  | 设计车位信息管理窗体    | 78  |
| 2.9.2  | 添加功能代码        | 79  |
| 2.10   | 系统管理模块设计      | 83  |
| 2.10.1 | 设计修改密码窗体      | 83  |
| 2.10.2 | 设计用户管理窗体      | 84  |
| 2.10.3 | 在主界面中增加权限限制代码 | 87  |
| 第3章    | 学生公寓管理系统      | 89  |
| 3.1    | 概述            | 89  |
| 3.2    | 系统分析与总体设计     | 89  |
| 3.2.1  | 系统需求分析        | 89  |
| 3.2.2  | 系统体系结构设计      | 89  |
| 3.2.3  | 开发工具的选择       | 92  |
| 3.2.4  | 开发实例所需的实用技术   | 92  |
| 3.2.5  | 数据库结构设计       | 93  |
| 3.3    | 设计工程框架        | 95  |
| 3.3.1  | 工程中的模块        | 95  |
| 3.3.2  | 添加类模块         | 97  |
| 3.4    | 系统主界面和登录窗体设计  | 103 |
| 3.4.1  | 设计主界面         | 103 |
| 3.4.2  | 设计登录窗体        | 105 |
| 3.5    | 楼房管理模块设计      | 107 |
| 3.5.1  | 设计功能界面的父窗体    | 107 |
| 3.5.2  | 设计楼房管理窗体      | 110 |
| 3.6    | 寝室管理模块设计      | 114 |
| 3.6.1  | 设计寝室管理窗体      | 115 |
| 3.6.2  | 设计空床统计窗体      | 118 |
| 3.7    | 学生基本信息管理模块设计  | 119 |
| 3.7.1  | 设计学生信息窗体      | 119 |
| 3.7.2  | 添加功能代码        | 121 |
| 3.8    | 交费管理模块设计      | 121 |
| 3.8.1  | 设计交费管理窗体      | 121 |
| 3.8.2  | 添加功能代码        | 122 |

|                            |            |
|----------------------------|------------|
| 3.9 来访登记管理模块设计 .....       | 123        |
| 3.9.1 设计来访登记窗体 .....       | 123        |
| 3.9.2 添加功能代码 .....         | 124        |
| 3.10 系统管理模块设计 .....        | 125        |
| 3.10.1 设计更改密码窗体 .....      | 125        |
| 3.10.2 设计操作员管理窗体 .....     | 126        |
| 3.10.3 在主界面中增加权限限制代码 ..... | 128        |
| <b>第4章 餐饮业收费管理系统 .....</b> | <b>129</b> |
| 4.1 概述 .....               | 129        |
| 4.2 系统分析与总体设计 .....        | 129        |
| 4.2.1 系统需求分析 .....         | 129        |
| 4.2.2 系统体系结构设计 .....       | 129        |
| 4.2.3 开发实例所需的实用技术 .....    | 131        |
| 4.2.4 数据库结构设计 .....        | 132        |
| 4.3 设计工程框架 .....           | 135        |
| 4.3.1 工程中的模块 .....         | 135        |
| 4.3.2 添加类模块 .....          | 136        |
| 4.4 系统主界面和登录窗体设计 .....     | 144        |
| 4.4.1 设计主界面 .....          | 144        |
| 4.4.2 设计登录窗体 .....         | 146        |
| 4.5 菜谱信息管理模块设计 .....       | 148        |
| 4.5.1 设计功能界面的父窗体 .....     | 148        |
| 4.5.2 设计菜谱信息窗体 .....       | 149        |
| 4.6 酒水信息管理模块设计 .....       | 153        |
| 4.6.1 设计酒水信息管理窗体 .....     | 154        |
| 4.6.2 添加功能代码 .....         | 155        |
| 4.7 采购入库管理模块设计 .....       | 158        |
| 4.7.1 设计入库信息窗体 .....       | 158        |
| 4.7.2 添加功能代码 .....         | 159        |
| 4.8 费用支出管理模块设计 .....       | 160        |
| 4.8.1 设计费用支出窗体 .....       | 160        |
| 4.8.2 添加功能代码 .....         | 161        |
| 4.9 点菜服务管理模块设计 .....       | 161        |
| 4.9.1 设计桌号信息窗体 .....       | 162        |
| 4.9.2 添加功能代码 .....         | 164        |
| 4.10 系统管理模块设计 .....        | 171        |
| 4.10.1 设计更改密码窗体 .....      | 171        |
| 4.10.2 设计操作员管理窗体 .....     | 172        |
| 4.10.3 权限限制代码 .....        | 173        |
| <b>第5章 学生学籍管理系统 .....</b>  | <b>174</b> |
| 5.1 概述 .....               | 174        |
| 5.2 系统分析与总体设计 .....        | 174        |
| 5.2.1 系统需求分析 .....         | 174        |
| 5.2.2 系统体系结构设计 .....       | 174        |
| 5.2.3 开发实例所需的实用技术 .....    | 177        |

|        |              |     |
|--------|--------------|-----|
| 5.2.4  | 数据库结构设计      | 177 |
| 5.3    | 设计工程框架       | 180 |
| 5.3.1  | 工程中的模块       | 180 |
| 5.3.2  | 添加类模块        | 182 |
| 5.4    | 系统主界面和登录窗体设计 | 189 |
| 5.4.1  | 设计主界面        | 190 |
| 5.4.2  | 设计登录窗体       | 191 |
| 5.5    | 院系信息管理模块设计   | 193 |
| 5.5.1  | 设计功能界面的父窗体   | 194 |
| 5.5.2  | 设计院系管理窗体     | 195 |
| 5.6    | 学生基本信息管理模块设计 | 199 |
| 5.6.1  | 设计学生基本信息管理窗体 | 199 |
| 5.6.2  | 添加功能代码       | 202 |
| 5.7    | 学生教育经历管理模块设计 | 207 |
| 5.7.1  | 设计学生教育经历管理窗体 | 207 |
| 5.7.2  | 添加功能代码       | 208 |
| 5.8    | 学生奖惩信息管理模块设计 | 209 |
| 5.8.1  | 设计学生奖惩信息管理窗体 | 209 |
| 5.8.2  | 添加功能代码       | 210 |
| 5.9    | 学籍变动信息管理模块设计 | 211 |
| 5.9.1  | 设计学籍变动信息管理窗体 | 211 |
| 5.9.2  | 添加功能代码       | 213 |
| 5.10   | 系统管理模块设计     | 214 |
| 5.10.1 | 设计更改密码窗体     | 215 |
| 5.10.2 | 设计操作员管理窗体    | 215 |
| 5.10.3 | 权限限制代码       | 217 |
| 第 6 章  | 毕业生就业管理系统    | 218 |
| 6.1    | 概述           | 218 |
| 6.2    | 系统分析与总体设计    | 218 |
| 6.2.1  | 系统需求分析       | 218 |
| 6.2.2  | 系统体系结构设计     | 218 |
| 6.2.3  | 开发实例所需的实用技术  | 221 |
| 6.2.4  | 数据库结构设计      | 221 |
| 6.3    | 设计工程框架       | 224 |
| 6.3.1  | 工程中的模块       | 224 |
| 6.3.2  | 添加类模块        | 226 |
| 6.4    | 系统主界面和登录窗体设计 | 233 |
| 6.4.1  | 设计主界面        | 233 |
| 6.4.2  | 设计登录窗体       | 235 |
| 6.5    | 院系信息管理模块设计   | 237 |
| 6.5.1  | 设计功能界面的父窗体   | 237 |
| 6.5.2  | 设计院系管理窗体     | 238 |
| 6.6    | 学生基本信息管理模块设计 | 242 |
| 6.6.1  | 设计学生基本信息管理窗体 | 243 |
| 6.6.2  | 添加功能代码       | 245 |
| 6.7    | 学生就业记录管理模块设计 | 250 |

|              |                      |            |
|--------------|----------------------|------------|
| 6.7.1        | 设计学生就业记录窗体.....      | 250        |
| 6.7.2        | 添加功能代码.....          | 251        |
| 6.8          | 就业单位信息管理模块设计.....    | 252        |
| 6.8.1        | 设计单位信息管理窗体.....      | 252        |
| 6.8.2        | 添加功能代码.....          | 253        |
| 6.9          | 招聘信息管理模块设计.....      | 254        |
| 6.9.1        | 设计招聘信息管理窗体.....      | 254        |
| 6.9.2        | 添加功能代码.....          | 256        |
| 6.10         | 系统管理模块设计.....        | 256        |
| 6.10.1       | 设计更改密码窗体.....        | 256        |
| 6.10.2       | 设计操作员管理窗体.....       | 257        |
| 6.10.3       | 权限限制代码.....          | 258        |
| <b>第 7 章</b> | <b>车辆调度管理系统.....</b> | <b>259</b> |
| 7.1          | 概述.....              | 259        |
| 7.2          | 系统分析与总体设计.....       | 259        |
| 7.2.1        | 系统需求分析.....          | 259        |
| 7.2.2        | 系统体系结构设计.....        | 259        |
| 7.2.3        | 开发实例所需的实用技术.....     | 261        |
| 7.2.4        | 数据库结构设计.....         | 262        |
| 7.3          | 设计工程框架.....          | 266        |
| 7.3.1        | 工程中的模块.....          | 266        |
| 7.3.2        | 添加类方法.....           | 267        |
| 7.4          | 系统主界面和登录窗体设计.....    | 268        |
| 7.4.1        | 设计主界面.....           | 268        |
| 7.4.2        | 设计登录窗体.....          | 270        |
| 7.4.3        | 添加数据模块.....          | 273        |
| 7.5          | 驾驶员信息管理模块设计.....     | 273        |
| 7.5.1        | 使用数据库控件.....         | 274        |
| 7.5.2        | 设计驾驶员列表窗体.....       | 274        |
| 7.5.3        | 设计驾驶员详细信息窗体.....     | 277        |
| 7.6          | 车场信息管理模块设计.....      | 279        |
| 7.6.1        | 使用数据库控件.....         | 279        |
| 7.6.2        | 设计车场列表窗体.....        | 279        |
| 7.6.3        | 设计车场详细信息窗体.....      | 280        |
| 7.7          | 车辆信息管理模块设计.....      | 281        |
| 7.7.1        | 使用数据库控件.....         | 282        |
| 7.7.2        | 设计车辆列表窗体.....        | 282        |
| 7.7.3        | 设计车辆详细信息窗体.....      | 283        |
| 7.8          | 车辆预约管理模块设计.....      | 284        |
| 7.8.1        | 使用数据库控件.....         | 284        |
| 7.8.2        | 设计车辆预约信息列表窗体.....    | 284        |
| 7.8.3        | 设计车辆预约详细信息窗体.....    | 285        |
| 7.9          | 车辆派遣管理模块设计.....      | 287        |
| 7.9.1        | 使用数据库控件.....         | 287        |
| 7.9.2        | 设计车辆派遣信息窗体.....      | 288        |
| 7.9.3        | 设计车辆派遣详细信息窗体.....    | 297        |

|                          |            |
|--------------------------|------------|
| 7.10 系统管理模块设计.....       | 300        |
| 7.10.1 使用数据库控件.....      | 300        |
| 7.10.2 设计用户管理窗体.....     | 301        |
| 7.10.3 设计密码修改窗体.....     | 303        |
| <b>第8章 计件工资管理系统.....</b> | <b>305</b> |
| 8.1 概述.....              | 305        |
| 8.2 系统分析与总体设计.....       | 305        |
| 8.2.1 系统需求分析.....        | 305        |
| 8.2.2 系统体系结构设计.....      | 306        |
| 8.2.3 开发实例所需的实用技术.....   | 308        |
| 8.2.4 数据库结构设计.....       | 308        |
| 8.3 设计工程框架.....          | 313        |
| 8.3.1 工程中的模块.....        | 313        |
| 8.3.2 添加类方法.....         | 314        |
| 8.4 系统主界面和登录窗体设计.....    | 315        |
| 8.4.1 设计主界面.....         | 315        |
| 8.4.2 设计登录窗体.....        | 317        |
| 8.4.3 添加数据模块.....        | 320        |
| 8.4.4 添加通用数据处理类.....     | 320        |
| 8.5 公司信息管理模块设计.....      | 325        |
| 8.5.1 使用数据库控件.....       | 325        |
| 8.5.2 设计公司详细信息窗体.....    | 326        |
| 8.6 部门信息管理模块设计.....      | 328        |
| 8.6.1 使用数据库控件.....       | 328        |
| 8.6.2 设计部门列表窗体.....      | 329        |
| 8.6.3 设计部门详细信息窗体.....    | 331        |
| 8.7 员工信息管理模块设计.....      | 333        |
| 8.7.1 使用数据库控件.....       | 333        |
| 8.7.2 设计员工列表窗体.....      | 333        |
| 8.7.3 设计员工详细信息窗体.....    | 335        |
| 8.7.4 设计员工固定工资窗体.....    | 337        |
| 8.8 产品信息管理模块设计.....      | 338        |
| 8.8.1 使用数据库控件.....       | 338        |
| 8.8.2 设计产品列表窗体.....      | 339        |
| 8.8.3 产品工序详细信息窗体.....    | 341        |
| 8.9 计件录入模块设计.....        | 342        |
| 8.9.1 使用数据库控件.....       | 342        |
| 8.9.2 设计计件列表信息窗体.....    | 342        |
| 8.9.3 计件详细信息窗体.....      | 344        |
| 8.10 工资管理模块设计.....       | 345        |
| 8.10.1 使用数据库控件.....      | 345        |
| 8.10.2 设计工资管理窗体.....     | 346        |
| 8.11 系统管理模块设计.....       | 350        |
| 8.11.1 使用数据库控件.....      | 350        |
| 8.11.2 设计用户管理窗体.....     | 350        |
| 8.11.3 设计密码修改窗体.....     | 353        |

# 第 1 章 数据库编程实用技术

本章将介绍 Delphi 数据库编程实用技术。使用 Delphi 开发数据库应用程序需要掌握如下基础技术：

- 基本的 Delphi 编程方法。
- 基本的后台数据库管理方法，例如创建数据库、创建表、创建视图、备份和还原数据库等。
- 常用 SQL 语句的使用。
- 在 Delphi 使用 ADO 数据库访问技术。

本书假定读者已经掌握了基本的 Delphi 编程方法，因此只对后面 3 项技术进行介绍。

## 1.1 SQL Server 数据库管理

本书实例采用 SQL Server 2000 作为后台数据库，这一小节将介绍管理 SQL Server 数据库的基本方法，为读者理解实例的实现过程奠定基础。

### 1.1.1 企业管理器

企业管理器是 SQL Server 2000 数据库系统中最重要管理工具，可以在企业管理器中对 SQL Server 数据库进行各种操作。

依次点击“开始”\“程序”\“Microsoft SQL Server”\“企业管理器”，运行 SQL Server 2000 企业管理器。企业管理器的主界面如图 1.1 所示。

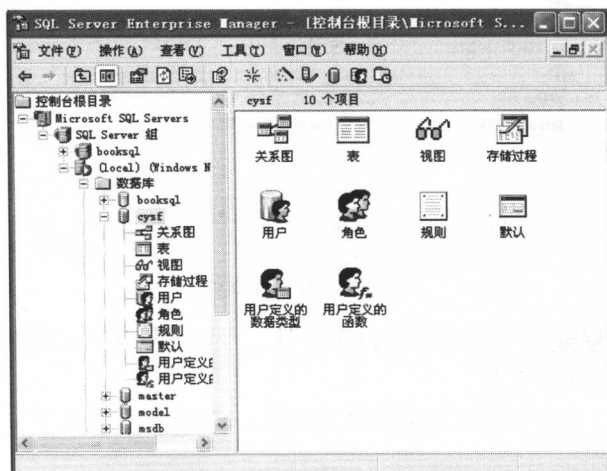


图 1.1 SQL Server 企业管理器

企业管理器的窗口分为两个部分，左侧窗口使用树状结构描述了 SQL Server 2000 的逻辑结构，右侧窗口显示左侧窗口中选择对象的具体内容。SQL Server 组可以包含多个 SQL Server 服务器，右键单击“SQL Server 组”，并选择菜单中的“新建 SQL Server 注册”，就可以将 SQL Server 服务器添加到树中，从而实现对网络中所有 SQL Server 数据库的管理。

一个 SQL Server 服务器下可以创建多个数据库，展开“数据库”节点，将显示指定服务器中的所有 SQL Server 数据库。数据库又由表、视图、用户等数据库对象组成，在企业管理器中，可以对这些数据库对象进行添加、修改、删除和查看等操作。

### 1.1.2 创建数据库

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。除了数据库名称外，其他参数都可以由系统默认设置。

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 1.2 所示。

(3) 键入新数据库的名称，假定数据库名为 UserMan。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：UserMan\_Data.mdf 和 UserMan\_Log.ldf。

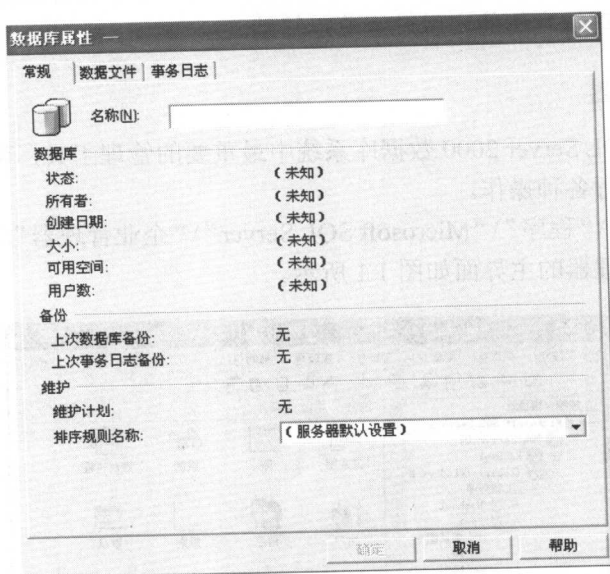


图 1.2 数据库属性

(4) 如果需要设置数据文件和事务日志文件，可以分别单击“数据文件”和“事务日志”选项卡。建议读者采用默认设置。

(5) 单击“确定”按钮完成创建数据库的操作。在企业管理器中可以看到新建的数据库。

### 1.1.3 创建表

表是包含数据库中所有数据的数据库对象。在设计数据库结构时，关键在于决定数据库由

哪些表组成，然后定义每个表的结构。

表的结构由若干个列（字段）组成。在定义表结构时，需要指定列名、数据类型、长度和其他列属性。

SQL Server 2000 的常用数据类型如表 1.1 所示。

**表 1.1 SQL Server 2000 常用数据类型说明**

| 数据类型              | 描述                                                                                                                                                                                                                                                                                       |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Image             | 可以用来存储超过 8 KB 的可变长度的二进制数据                                                                                                                                                                                                                                                                |
| Char              | 固定长度的字符数据类型。Char[(n)]表示长度为 $n$ 个字节的固定长度且非 Unicode 的字符数据。 $n$ 必须是一个介于 1~8 000 之间的数值。存储大小为 $n$ 个字节                                                                                                                                                                                         |
| Varchar           | 非固定长度的字符数据类型。Varchar[(n)]表示长度为 $n$ 个字节的可变长度且非 Unicode 的字符数据。 $n$ 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 $n$ 个字节。所输入的数据字符长度可以为零                                                                                                                                                      |
| Datetime          | 从 1753 年 1 月 1 日到 9999 年 12 月 31 日的日期和时间数据，精确度为 0.03s(等于 3.33ms 或 0.00333s)                                                                                                                                                                                                              |
| Int               | 从 $-2^{31}$ ( $-2\ 147\ 483\ 648$ ) $\sim 2^{31}-1$ ( $2\ 147\ 483\ 647$ ) 的所有整型数据。存储大小为 4B                                                                                                                                                                                              |
| Smallint          | 从 $-2^{15}$ ( $-32\ 768$ ) $\sim 2^{15}-1$ ( $32\ 767$ ) 的整型数据。存储大小为 2B                                                                                                                                                                                                                  |
| Tinyint           | 从 0~255 的整型数据。存储大小为 1B                                                                                                                                                                                                                                                                   |
| Decimal 和 Numeric | 带定点精度和小数位数的 Numeric 数据类型。Decimal[(p[,s])] 和 Numeric[(p[,s])] 表示定点精度和小数位数。使用最大精度时，有效值从 $-10^{38}+1 \sim 10^{38}-1$ 。 $p$ 表示精度，指定小数点左边和右边可以存储的十进制数字的最大个数。精度必须是从 1 到最大精度之间的值。最大精度为 38。 $s$ 表示小数位数，指定小数点右边可以存储的十进制数字的最大个数。小数位数必须是从 0~ $p$ 之间的值。默认小数位数是 0，因而 $0 \leq s \leq p$ 。最大存储大小基于精度而变化 |
| Float [(n)]       | 从 $-1.79\text{E}+308 \sim 1.79\text{E}+308$ 之间的浮点数字数据。 $n$ 为用于存储科学记数法 Float 数尾数的位数，同时指示其精度和存储大小。 $n$ 必须为从 1~53 之间的值                                                                                                                                                                      |
| Real              | 从 $-3.40\text{E}+38 \sim 3.40\text{E}+38$ 之间的浮点数字数据。存储大小为 4B                                                                                                                                                                                                                             |
| Bit               | bit 数据类型只能包括 0 或 1。可以用 bit 数据类型代表 TRUE 或 FALSE、YES 或 NO。例如，询问客户是否为初次访问的问题可存储在 bit 列中，类似于其他数据类型中的布尔型                                                                                                                                                                                      |

在定义 char、varchar、decimal、numeric 和 float 数据类型时，需要指定数据的长度。

下面在数据库 UserMan 中创建一个表 Users，用于保存用户信息，结构如表 1.2 所示。

**表 1.2 表 Users 的结构**

| 字段名      | 数据类型 | 长度 | 具体说明 |
|----------|------|----|------|
| UserId   | int  |    | 用户编号 |
| UserName | Char | 10 | 用户名  |
| UserPwd  | Char | 10 | 密码   |

在 SQL Server 2000 企业管理器中，展开服务器组，然后展开服务器实例。展开要添加表的数据库，右击“表”目录。选择菜单中“新建表”命令，打开表设计器，如图 1.3 所示。

首先输入列名。列名必须惟一，而且要符合 SQL Server 的标识符规范。

选择某一列，在表设计器的上半部分可以编辑列名、数据类型、长度和允许空等属性。如果选中“允许空”复选框，则该列数值可以为空，否则在输入数据时，必须输入该列数据。在表设计器的下半部分，可以编辑列的其他属性，包括描述、默认值、精度、小数位数和标识等。

输入完成后，单击“保存”图标，将弹出“输入表名”对话框，如图 1.4 所示。

输入表名后单击“确定”按钮，可以在企业管理器中看到新建的表。右键单击该表，在弹

出菜单中选择“设计表”命令，可以打开表设计器对其进行修改。

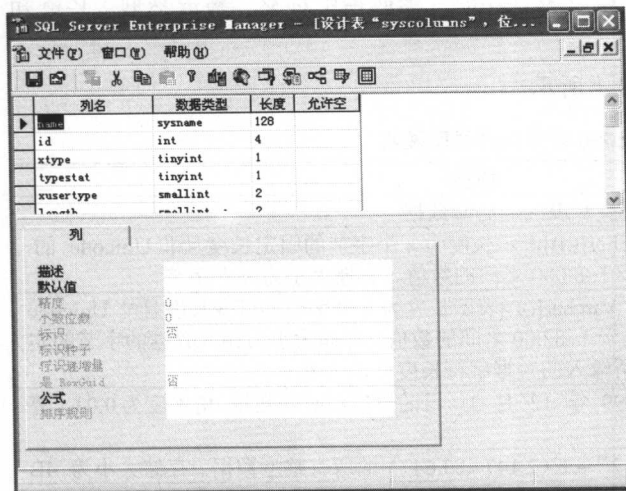


图 1.3 表设计器

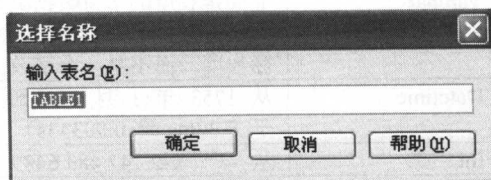


图 1.4 输入表名

右键单击表，在弹出菜单中选择“打开表”/“返回所有行”，可以以表格的形式查看表中的数据，如图 1.5 所示。

| id | status  | first    | indid | root     |   |
|----|---------|----------|-------|----------|---|
| 1  | 18      | <Binary> | 1     | <Binary> | 4 |
| 1  | 2       | <Binary> | 2     | <Binary> | 7 |
| 1  | 0       | <Binary> | 3     | <Binary> | 9 |
| 2  | 18      | <Binary> | 1     | <Binary> | 6 |
| 2  | 0       | <Binary> | 255   | <Binary> | 0 |
| 3  | 16402   | <Binary> | 1     | <Binary> | 4 |
| 3  | 2       | <Binary> | 2     | <Binary> | 9 |
| 4  | 16402   | <Binary> | 1     | <Binary> | 2 |
| 4  | 2       | <Binary> | 2     | <Binary> | 3 |
| 6  | 16402   | <Binary> | 1     | <Binary> | 1 |
| 8  | 0       | <Binary> | 0     | <Binary> | 7 |
| 9  | 16402   | <Binary> | 1     | <Binary> | 1 |
| 10 | 2113554 | <Binary> | 1     | <Binary> | 2 |

图 1.5 查询表中数据

在表中，有的列可以被设置作为行的惟一标识，也就是说在表的所有行中，此列的数据是惟一的。通常情况下，可以把编号列设置为惟一标识列。为了描述这种惟一标识列，数据库中引入了键（KEY）的概念。在 SQL Server 中，有两种类型的键：

- **主键（PRIMARY KEY）** 表中的一组列，对表中的每一行这些列都有惟一的值。定义主键可以强制在指定列中不允许输入空值，如果要插入行的主键值已经存在，则此行不允许被插入。表只能有一个主键。
- **外键（FOREIGN KEY）** 是用于建立和加强两个表数据之间的链接的一列或多列。有时称这两个参与表为外键表和主表。外键可以约束表中的行，因此也称为外键约束。外键确保添加到外键表中的任何行在主表中有相应的行。

在表设计器中单击要定义为主键的数据列，如果要选择多个列，可同时按下 CTRL 键。

右击该列，然后选择“设置主键”命令。此列的前面出现了主键符号，同时“允许空”的标记被去掉。

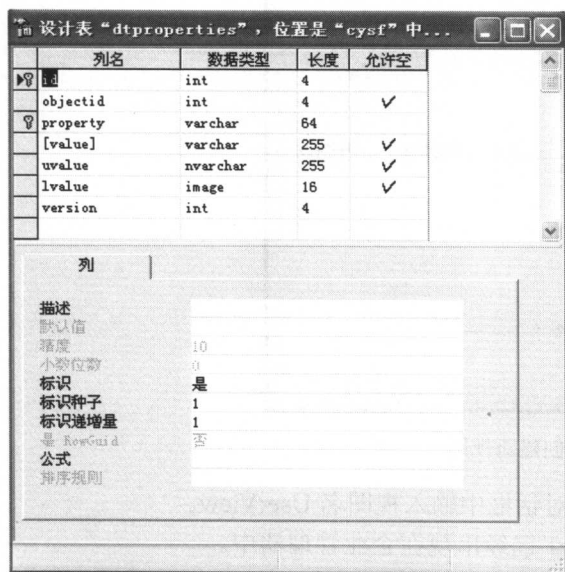


图 1.6 设置标识列

例如，可以从用户表 Users 中提取 UserName 和 UserPwd 创建一个视图，方法如下：

(1) 在企业管理器中，展开指定实例的数据库，右键单击“视图”项，在弹出菜单中选择“新建视图”命令，打开“视图设计器”窗口。此窗口由 4 部分窗格组成，由上到下分别为关系图窗格、网格窗格、SQL 窗格和结果窗格，如图 1.7 所示。

(2) 右击关系图网格，选择“添加表”命令，打开“添加表”对话框，如图 1.8 所示。

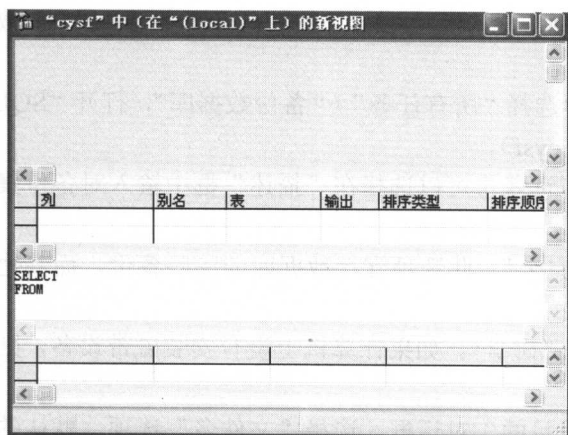


图 1.7 视图设计器

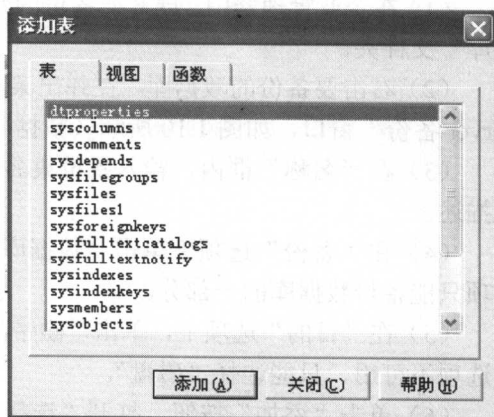


图 1.8 “添加表”对话框

(3) 选择表 Users，单击“添加”按钮，将表 Users 添加到关系图网格中。

(4) 在关系图网格中，选择表 Users 中的 UserName 和 UserPwd 字段。

(5) 设置完成后，单击工具栏中的“运行”按钮，运行视图。运行结果出现在“结果窗格”中，如图 1.9 所示。