



中国计算机函授学院图书编写中心 组编
国家教育部电教办计算机培训基地指定培训用书

电脑工程师丛书

Visual Basic网络开发

罗喆帅 丁凉 葛艳 叶勇 石竹 编

Visual Basic WANGLUO KAIFA

实例

●北京伯通电子出版社

中国计算机函授学院图书编写中心组编
国家教育部电教办计算机培训基地指定培训用书



电脑工程师丛书

Visual Basic 网络开发实例

罗喆帅 丁凉 葛艳 叶勇 石竹 编

北京伯通电子出版社

内
容
简
介

Visual Basic 中文版是微软公司推出的可视化快速开发工具。基于 Windows 平台的 *Visual Basic*, 不但简单易用, 而且在开发信息管理系统、网络应用、通信控制等方面效率十分显著。因此, 有越来越多的程序员开始使用 *Visual Basic* 编程语言。

本书详细介绍了 *Visual Basic* 在网络方面的应用, 语言通俗易懂、结构清晰、立意新颖, 各章均以实例为中心展开介绍, 使读者迅速掌握所需知识。所举实例全部经上机调试通过。

本书适合使用该软件的专业编程人员, 也适合于广大计算机爱好者。

丛 书 名: 电脑工程师丛书

书 名: *Visual Basic* 网络开发实例

编 者: 罗喆帅 丁凉 葛艳 叶勇 石竹

出版发行: 北京伯通电子出版社

社 址: 北京市朝阳区慧忠北里 107 楼 1911 号 邮政编码 100101

电 话: 010-64868172

印 刷: 合肥学苑印刷厂印刷

经 销: 全国新华书店经销

开 本: 787×1092(mm) 1/16 印张: 14 字数: 331.5 千字

版 次: 2003 年 1 月第 1 版 2003 年 1 月第 1 次印刷

印 数: 1~5000

书 号: ISBN 7-900112-44-8/TP·23

定 价: 25.00 元(盘配书)

* 版权所有 侵权必究

前 言

当前市场上计算机类图书浩如烟海,仅某一类型的书就有几十种乃至上百种之多,竞争之激烈可说为图书业之最。在这种情况下,中国计算机函授学院图书编写中心把眼光定位于计算机的中级用户和对计算机在各个应用领域有兴趣的计算机爱好者,经过了周密的策划,为他们量身而作了“电脑工程师”丛书。这套丛书秉承中国计算机函授学院图书编写中心历来“出好书、出精品书”的意识,组织了大量具有实际工作经验的专家、学者和长期从事计算机应用的工程技术人员,将他们平时工作和教学中用到的精彩范例,加以精炼、提高,编写了这套丛书。

丛书内容侧重于实际操作,使已掌握某一方面入门知识的读者,通过对应用实例的学习,能够提高其自身的动手能力和实际应用能力,从而很快地将所学知识应用到具体工作中,能独当一面地进行工作。

丛书中行文力求通俗易懂,以操作为主,突出实用性。为使读者在较短时间内掌握计算机应用知识,在详细给出实例制作过程之前,我们先给出实例的创意分析及素材准备,在描述过程中尽量多给出一些制作技巧和典型说明,以期进一步提高学习效果。

最后衷心感谢参与本套丛书协作的全体老师和创作人员,衷心祝愿本套丛书的读者早日成为电脑高手。

编者的话

当今时代是互联网时代,随着网络的快速发展和迅速普及,它已对我们的日常生活、工作产生了广泛的影响,能编写基于网络的应用程序已经成为广大程序员的基本要求。

Visual Basic 以它易于学习、功能强大、编程效率高等特点,赢得了最广泛的程序员加盟。Visual Basic 6.0 中文版是目前程序员使用得最多的 Visual Basic 中文版本,它在对集成开发环境(IDE)作了较大改动的同时,最显著的变化是更加易于开发基于网络的应用程序。

本书共分为六章:

第一章是编写基于客户/服务器(C/S)模式的应用程序,在本章中介绍了 C/S 结构的特点、远程数据库的常见访问方法、ADO 知识简介,最后结合学生成绩管理系统这个应用实例来讲述开发基于 C/S 模式应用程序的开发步骤、知识点和注意事项。

第二章介绍了 ASP 的基本知识,着重结合实例讲解 ASP 常见的内置对象的使用方法。

第三章力求使读者能够应用 ASP 的知识开发 WEB 数据库应用程序,在本章节中较详细地介绍了三个应用程序实例:BBS 应用程序、调查投票系统应用程序、网上人才与劳务市场应用程序。

第四章介绍了 Visual Basic 在文件传输方面的应用。

第五章介绍 Visual Basic 在电子邮件方面的应用。

第六章介绍了 Visual Basic 在 WinSock 方面的应用。

限于编者水平的局限性,书中不可避免存在一些问题和不足,因此真诚欢迎广大读者批评指正。

编者

二〇〇二年十一月

目 录

1

基于客户/服务器模式的数据库应用程序实例

1.1 客户/服务器结构的特点	(1)
1.2 远程数据库的常见访问方法	(2)
1.3 ADO 简介	(3)
1.3.1 ADO 对象的优点	(3)
1.3.2 基本的 ADO 编程模型和对象模型	(3)
1.4 学生成绩管理系统	(4)
1.4.1 系统的环境	(4)
1.4.2 系统的功能介绍	(4)
1.4.3 数据库的结构和数据库的建立	(5)
1.4.4 ODBC 数据源的注册方法	(7)
1.4.5 程序代码及分析	(10)

2

ASP 内置对象应用实例

2.1 ASP 编程的准备知识	(42)
2.1.1 ASP 语言概况	(43)
2.1.2 IIS 的安装和日常管理	(45)
2.2 ASP 内置对象简介	(50)
2.3 Response 对象	(51)
2.3.1 Cookies 集合实例	(51)
2.3.2 Write 方法实例	(52)
2.4 Request 对象	(53)
2.4.1 Cookies 集合实例	(53)
2.4.2 POST 和 GET 方法	(54)
2.4.3 Form 集合实例	(54)
2.4.4 QueryString 集合实例	(56)
2.4.5 ServerVariables 集合实例	(57)
2.5 Server 对象	(58)
2.5.1 ScriptTimeout 属性实例	(59)
2.5.2 Execute 方法实例	(59)

2.5.3	Transfer 方法实例	(60)
2.5.4	MapPath 方法实例	(61)
2.6	Session 对象	(61)
2.6.1	变量的作用域	(62)
2.6.2	Timeout 属性实例	(63)
2.6.3	Session 对象实例	(64)
2.6.4	SessionID 属性实例	(66)
2.6.5	Abandon 方法实例	(66)
2.7	Application 对象	(67)
2.7.1	Application 对象实例	(67)
2.7.2	Lock 和 Unlock 方法实例	(68)
2.7.3	GLOBAL.ASA 文件	(69)
2.8	ObjectContext 对象	(70)
2.9	ASPErrors 对象	(71)
2.10	Scripting.FileSystemObject 对象	(74)
2.10.1	Drives 集合的实例	(74)
2.10.2	有关驱动器的各种方法的实例	(76)
2.10.3	有关文件夹的各种方法的实例	(77)
2.10.4	有关文件的各种方法的实例	(79)
2.11	Scripting.TextStream 对象	(80)
2.11.1	建立文本文件的实例	(81)
2.11.2	打开文本文件的实例	(82)

3

利用 ASP 开发 Web 数据库应用程序实例

3.1	应用程序开发环境及必备的知识	(83)
3.2	开发 BBS 应用程序实例	(84)
3.2.1	程序组成	(84)
3.2.2	数据表结构	(84)
3.2.3	程序代码及分析	(85)
3.3	调查投票系统应用程序实例	(92)
3.3.1	程序组成	(92)
3.3.2	数据库结构	(92)
3.3.3	程序代码及分析	(93)
3.4	网上人才与劳务市场应用程序实例	(98)
3.4.1	程序组成	(98)
3.4.2	数据表结构	(99)
3.4.3	程序代码及分析	(100)

4

利用 Internet Transfer 控件开发程序实例

4.1	Internet Transfer 控件简介	(128)
4.2	利用 Internet Transfer 控件设计程序实例	(128)
4.3	利用 Internet Transfer 控件设计 FTP 程序实例	(133)

5

利用 MAPI 控件设计电子邮件收发程序实例

5.1	MAPI 控件简介	(151)
5.2	电子邮件的基本概念	(152)
5.3	邮件缓冲区	(153)
5.4	程序设计	(154)

6

利用 Winsock 控件设计程序实例

6.1	Winsock 控件简介	(169)
6.2	TCP 方式与 UDP 方式	(169)
6.2.1	TCP 方式	(169)
6.2.2	UDP 方式	(170)
6.2.3	选择通讯协议	(170)
6.2.4	协议的设置	(170)
6.3	利用 Winsock 控件设计聊天程序实例	(171)
6.4	利用 Winsock 控件设计文件通信程序实例	(179)
6.4.1	服务器端程序	(180)
6.4.2	客户端的程序	(190)

附录 A	VBScript/Visual Basic 常用函数	(204)
-------------	----------------------------------	-------

附录 B	ADO 常用对象的方法和属性	(211)
-------------	----------------------	-------

基于客户/服务器模式的数据库 应用程序实例

1

众所周知,计算机网络的主要功能就是实现计算机资源的共享。现在单位内部的局域网已经十分普及了,采用客户/服务器模式来访问数据库的信息,不仅可以实现数据库信息的共享,而且也可以对数据库系统进行集中管理,从而提高数据库的安全性。

本章主要讲述以下几个内容:

- 客户/服务器结构的特点
- 远程数据库的常见访问方法
- ADO 简介
- 学生成绩管理系统应用实例

1.1 客户/服务器结构的特点

客户/服务器结构(Client/Server 简称 C/S 结构)是在专用服务器的基础上发展起来的,包括处于同一网络环境中的其他计算机,其中专用服务器称为服务器端,其他计算机称为客户端。数据库系统位于服务器端,目前比较流行的基于 C/S 结构的数据库系统有 Microsoft 的 SQL Server, Sybase 的 Adaptive Server 和 Oracle 等。应用程序常位于客户端,负责接收用户的指令以及向用户显示数据等。客户/服务器结构的示意图如图 1.1 所示。

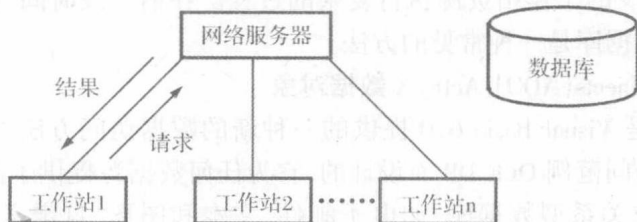


图 1.1 客户/服务器结构

位于客户端的应用程序接受用户的操作指令,并向数据库提出服务请求。数据库所在的服务器提供大容量的存储设备,管理数据资源并运行数据库引擎,处理客户端的请求,并将执行的结果返回给客户端的应用程序。

客户/服务器结构的优点:

①分布式处理模式:应用程序提交的任务分别由客户端和服务端承担,因而速度快,机器档次要求不高。

②系统可扩充性好:当系统规模扩大时,可以不必重新设计整个系统,只是简单地加挂服务器或客户机,因此可以更有效地充分利用现有的系统资源。

③数据安全性好:由于数据库在客户/服务器结构中是集中式、多用户的管理模式,因而对保证数据库的完整性、数据安全保护和封锁机制是极为有利的。

④跨平台:运用标准的通讯协议,应用程序能够从不同地方,用不同的硬件结构来进行通讯。

1.2 远程数据库的常见访问方法

Visual Basic 6.0 是开发基于 C/S 模式的数据库应用程序的比较好的一种工具。它提供了多种访问远程数据库的方法,因此在设计应用程序时,用户拥有比较大的灵活性。常用的方法有如下几种:

(1) 基于 API 的数据库接口

Visual Basic 所使用的基于 API 的数据库应用程序开发接口包括 ODBC API 和 DB - Library API 两种,其中 DB - Library API 只能访问 SQL Server 数据库系统。在使用 DB - Library API 时,VBSQL 控件还需要其他声明文件、驱动程序和动态连接库等,而这些文件不包含在 VB 中,它可以从 SQL WorkStation 等产品中获得。基于 API 编写的应用程序代码执行效率高,但编写代码的难度较大。

(2) Data Access Objects(DAO) 数据访问对象

DAO 既可以和 Microsoft Jet 数据库引擎一起使用,也可以通过设置 ODBC Direct 选项而单独使用。使用 DAO 访问远程数据库时,可以通过 Data 控件的 Connect 属性来建立和 ODBC 数据源的连接,其他的操作方法和访问本地数据库十分类似。

(3) Remote Data Access Objects(RDO) 远程数据访问对象

远程数据访问对象提供了一系列的对象用来满足对远程数据库访问的特殊要求。在 ODBC API 和驱动程序管理器之上,RDO 通过一个很薄的代码层,用来建立连接,创建结果集和游标,并且使用尽可能少的工作站资源执行复杂的过程。在前一段时间,使用 RDO 和 Remote Data 控件来访问远程数据库是一种常见的方法。

(4) ActiveX Data Objects(ADO) ActiveX 数据对象

ActiveX 数据对象是 Visual Basic 6.0 提供了一种新的数据访问方法。ADO 是为 Microsoft 最新和最强大的数据访问范例 OLE DB 而设计的,它为任何数据源提供了高性能的访问,这些数据源包括关系型和非关系型数据库,如电子邮件、文本和图形、自定义业务对象等。通过 ADO 访问数据库比使用 DAO 和 RDO 方法访问数据库,无论是在代码的编写难易度上,还是程序的执行效率上都是有很大的优势。因此,建议使用 ADO 对象去编写基于 C/S 模式的数据库应用程序。

1.3 ADO 简介

1.3.1 ADO 对象的优点

- ①ADO 的大部分对象是相对孤立的,用户可以根据应用程序的实际情况来创建不同的对象,对象之间的相对独立性使得用户在编写应用程序时对数据的访问手段更加灵活和有效。
- ②ADO 对象对数据的更新有立即模式和批处理模式,其中批处理模式的运用可以减轻网络的通信流量。
- ③ADO 对象可以根据用户对数据库记录的不同操作要求来决定采用何种方式的游标类型。ADO 对象支持的四种游标类型分别是 Forward-only, Static, Keyset 和 Dynamic。在程序中选最佳的游标类型,能够提高程序的执行效率,减少程序占用的系统资源。
- ④ADO 对象有完善的错误处理机制,这为用户编写应用程序的错误陷阱提供了极大的方便。
- ⑤支持服务器端的存储过程和返回多个记录集的查询。

1.3.2 基本的 ADO 编程模型和对象模型

ADO 提供了以下的操作方式:

- ①连接到数据源,同时可确定对数据源的操作是否成功。
- ②指定访问数据源的命令,同时可以优化参数。
- ③执行指定的命令。
- ④如果这个命令使数据按表中的行的形式返回,则将这些行存储在易于操作、检查和更改的缓存中。
- ⑤在适当的情况下,可使用缓存行的更改内容来更新数据源。
- ⑥提供多种常规方法来检测错误。

在大多数情况下,需要在编程模型中采用以上的所有步骤,但由于 ADO 有很强的灵活性,所以有时只需执行部分步骤就可以完成工作。

ADO 对象包括有六个常用对象,它们分别是 Connection, Command, Parameter, RecordSet, Field, Error。它们的示意图如图 1.2 所示。

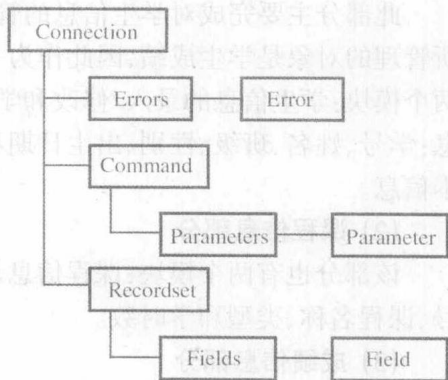


图 1.2 ADO 对象模型

在六大对象中最主要的对象是 Connection, Command 和 Recordset 这三个对象。Connection 负责打开与数据源的连接,Command 对象指定了对数据库的执行命令,Recordset 对象又是这三个主要对象中的最重要对象,任何基于 ADO

编写的应用程序几乎都不能缺少 Recordset 对象,它负责存储来自基本表或命令执行结果的记录全集。在任何时候,Recordset 对象所指定的当前记录均为集合内的单个记录。在以上模型中,每个 Connection, Command, Recordset 和 Field 对象都有 Properties 集合。有关 ADO 常用对象的方法和属性请读者查阅附录。

1.4 学生成绩管理系统

前面简单地介绍了开发 C/S 模式应用程序的基本知识,本节将通过一个具体的实例——学生成绩管理系统,阐述开发此类应用程序常见步骤以及其他的必备知识。

1.4.1 系统的环境

本系统对计算机的硬件要求并不高,但是计算机最好能够运行 Windows 2000 Professional。本实例的开发工具为中文 Visual Basic 6.0 企业版。在开发基于 C/S 模型的数据库应用程序时,数据库的选择十分重要。常见的数据库管理系统(DBMS)平台有 Access 数据库、SQL Server 以及 Oracle 等。理论上,后台选择哪种数据库平台并不影响技术的实现,但是从实际应用的角度以及从真正体现 C/S 模式的特点来说,后台常常选择 SQL Server 和 Oracle 等大型数据库。本实例中,选择的数据库系统是 Microsoft SQL Server 7.0。因此,读者必须在开发机器中安装此数据库系统软件。

1.4.2 系统的功能介绍

学生成绩管理系统主要分为三个部分:学生信息,课程信息和成绩信息,其中每部分又包括了若干个模块。

(1) 学生信息部分

此部分主要完成对学生信息的管理,是这个系统的最基本部分。因为学生成绩管理系统所管理的对象是学生成绩,因此作为一个系统必须对学生的最基本信息有所反映。该部分有两个模块:学生信息的录入/修改和学生信息浏览。前一模块用于录入或修改学生的最基本信息:学号、姓名、班级、性别、出生日期和生源。第二个模块按班级的顺序浏览所有学生的最基本信息。

(2) 课程信息部分

该部分也有两个模块:课程信息录入/修改和课程信息浏览。它们管理的内容包括课程编号、课程名称、类型和学时数。

(3) 成绩信息部分

此部分分为三个模块:成绩录入/修改、成绩按班级浏览和成绩查询。这个部分为整个系统的核心。成绩录入/修改部分是按班级和课程的分类来输入学号和成绩信息,成绩按班级浏览是按班级为单位查看每个学生的成绩信息,成绩查询是根据学号来查询对应的学生成绩信息。

1.4.3 数据库的结构和数据库的建立

本实例中需要建立一个数据库,并在这个数据库下建立三个表,这三张表分别为学生信息表、课程表和成绩表。

①学生信息表:表名为 Student_info,本表记录了学生最基本信息。表中字段具体属性如表 1-1 所示。

表 1-1 学生信息表

字段名	数据类型	长度	是否允许空
学号	Varchar	50	否
姓名	Varchar	50	否
班级	Varchar	50	否
性别	Char	2	否
出生日期	Datetime	8	是
生源	Varchar	50	是

②课程表:表名为 Course,本表记录了有关课程的信息。表中字段具体属性如表 1-2 所示。

表 1-2 课程表

字段名	数据类型	长度	是否允许空
课程编号	Varchar	50	否
课程名称	Varchar	50	否
类型	Char	1	是
学时	Varchar	3	是

“类型”字段的长度仅为 1,字符 A 代表课程类型是必修课,字符 B 代表的是指定选修课,字符 C 代表的是自由选修课。

③成绩表:表名为 Grade,本表记录了每个学生的成绩信息。表中字段具体属性如表 1-3 所示。

表 1-3 成绩表

字段名	数据类型	长度	是否允许空
学号	Varchar	50	否
班级	Varchar	50	否
课程编号	Varchar	50	否
成绩	Decimal	9(小数位 1)	否

在了解了三个表的结构之后就应该设法在 SQL Server 环境中创建数据库和三个表以及权限分配。Student_info、Course 和 Grade 三个表的创建方法雷同,此处仅以 Student_info 为例来说明创建表的步骤。

(1)创建数据库

① 执行“开始/程序/Microsoft SQL Server 7.0/Enterprise Manager”命令。

② 在 Console Root/Microsoft SQL Servers 界面中依次展开 Microsoft SQL Servers、SQL Server Group 和 SQL Server 服务器名(本实例假设是以超级用户 sa 登录的,sa 的默认密码为空。)

③ 选择其中的 DataBase 选项,单击鼠标右键在弹出的快速单菜单中选择“New Database”选

项,如图 1.3 所示。

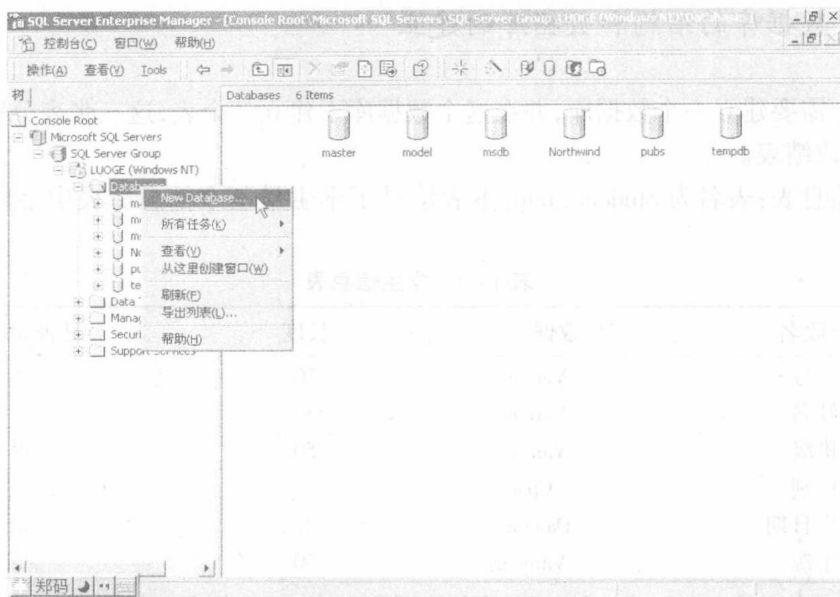


图 1.3 执行“New Database”选项

当然也可以通过菜单命令“操作/New Database...”选项创建数据库。

④ 系统打开“Database Properties”对话框,开始创建所需要的数据库结构,如图 1.4所示。在“Name”文本框中输入数据库名 student,其余选项保持默认值。

单击对话框中的“确定”按钮,数据库 student 就创建完毕了。

(2) 创建数据表

① 展开数据库 student,选中 tables,单击鼠标右键,在弹出的快捷菜单中选择“New Table...”选项。

系统打开数据表设计工具,输入要创建的表的名称 student_info,如图 1.5 所示。

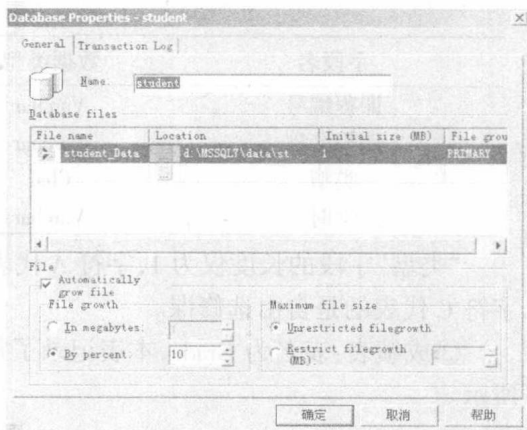


图 1.4 创建一个数据库 student

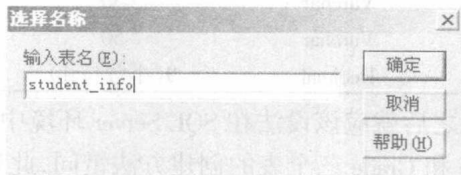


图 1.5 填入数据表名称

② 在数据表设计工具中逐一填入要创建的字段名称与类型等设置,创建结果如图 1.6 所示。

(3) 设置数据表权限

① 单击 Student 数据库的 Tables 项,此时窗口右边出现所有表的列表。



图 1.6 Student_info 表结构

② 选择 Student_info 表,单击鼠标右键,在弹出的快捷菜单中选择“属性”选项。

③ 系统弹出“Table Properties”对话框,单击“Permissions”按钮,系统弹出对象属性对话框,依次选中 SELECT, INSERT, UPDATE, DELETE, EXEC 和 DRI,显示结果如图 1.7 所示。

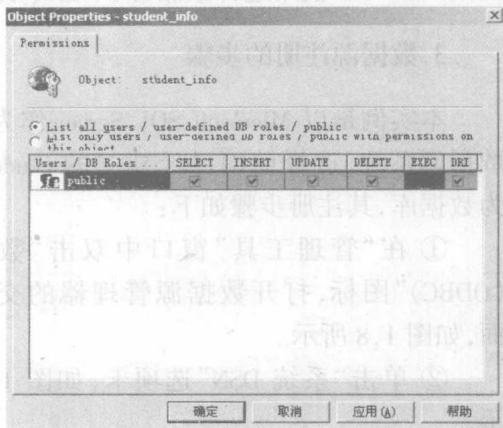


图 1.7 数据表 Student_info 权限设置

1.4.4 ODBC 数据源的注册方法

ODBC(Open DataBase Connectivity 开放式数据库连接)是一种应用程序接口规范。数据源是数据的提供者,它在应用程序和数据库之间起到桥梁的作用。从应用程序角度来看,应用程序是通过数据源来存取数据库中的数据;从数据库角度来看,数据库中的数据是通过数据源来提供给用户的应用程序。ODBC 技术的出现,为前台的应用程序是统一的编程界面而后台的数据库是不同的数据库管理系统(DBMS)这种方式提供了实现的可能性,也就是说,当后台的 DBMS 发生了变化,应用程序仅改动数据源的相关属性就可以了。在应用程序中可以使用 API 函数直接调用 ODBC,或者利用 DAO、ADO 来调用 ODBC。但是无论使用哪种方法,都必须首先进行 ODBC 数据源的注册,这样才能对数据源进行连接、访问和操作,对数据库进行增加、删除、修改和检索等操作。

1. ODBC 数据源管理器

在进行数据库开发时,为了达到配置 ODBC、进行 DSN 定义注册的目的,微软给出了一个手工操作的解决方法。在 Windows 95/98 操作系统的控制面板中,有一个名为“ODBC 数据源(32 位)”的图标,可以通过它激活专门为用户设置 ODBC 环境的程序(ODBC Data Source Administrator, ODBC 数据源管理器)。在 Windows 2000 操作系统中,上述图标被放置在控制面板的“管理工具”里面。这个用于设置 ODBC 环境的程序叫做桌面驱动程序,它支持数种 DBMS。当用户想增加一个数据源和一个所需要的驱动程序时,可以通过 ODBC 数据源管理器配置特定类型的数据库。大多数情况下,在编写对数据库操作的程序时,至少需要知道诸如数据库文件

名、系统(本地或远程)、文件夹等信息,同时要给数据源命名(本实例在 Win 2000 环境下实现)。

2. 数据源的类型

用户可以定义以下三种类型的数据源:

① 用户 DSN:作为位于计算机本地的用户数据源而创建的,并且只能被创建这个数据源的用户所使用,只能应用在本机中。

② 系统 DSN:作为属于计算机或系统而不是特定用户的系统数据源而创建的,用户必须有访问权才能使用,系统数据源对当前机器上的所有用户可见。

③ 文件 DSN:指定到文件中作为文件数据源而定义的,任何已经正确地安装了驱动程序的用户皆可以使用这种数据源,不必专属于某一用户或本地计算机。

3. 数据源注册的步骤

本实例是以 Microsoft SQL Server 作为后台数据库系统,选择前面已经建立的 Student 作为数据库,其注册步骤如下:

① 在“管理工具”窗口中双击“数据源(ODBC)”图标,打开数据源管理器的交互界面,如图 1.8 所示。

② 单击“系统 DSN”选项卡,如图 1.9 所示。

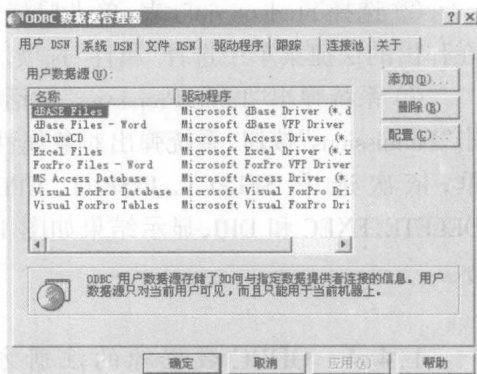


图 1.8 数据源管理器

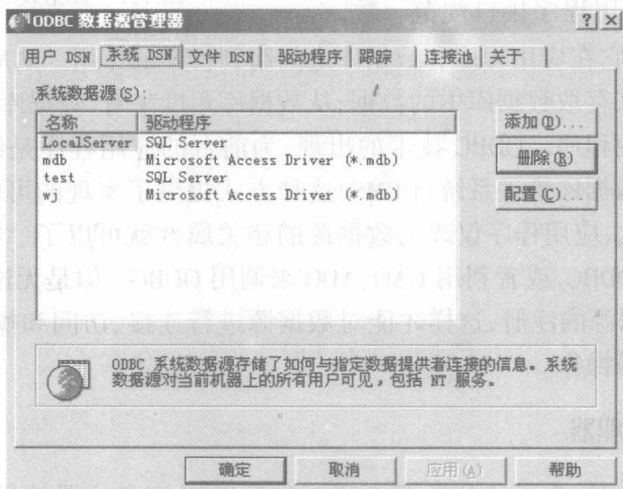


图 1.9 系统 DSN

③ 单击“添加”按钮,创建一个系统数据源。列表框中列出了许多数据源的驱动程序,选择 SQL Server,如图 1.10 所示。

④ 单击“完成”按钮,进入“建立新的数据源到 SQL Server”对话框。在“名称”框中输入新的数据源名称 dns_student,在“服务器”框中输入想连接的 SQL Server 服务器名或 IP 地址。如果要连接的 SQL Server 服务器是安装在本机的,那么就从下拉框中选择 local 选项,表明是连接

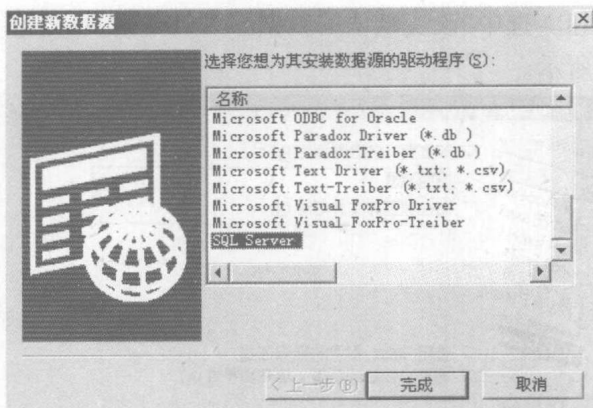


图 1.10 创建 SQL Server 数据源

到本地服务器。本实例选择 local 选项,如图 1.11 所示。

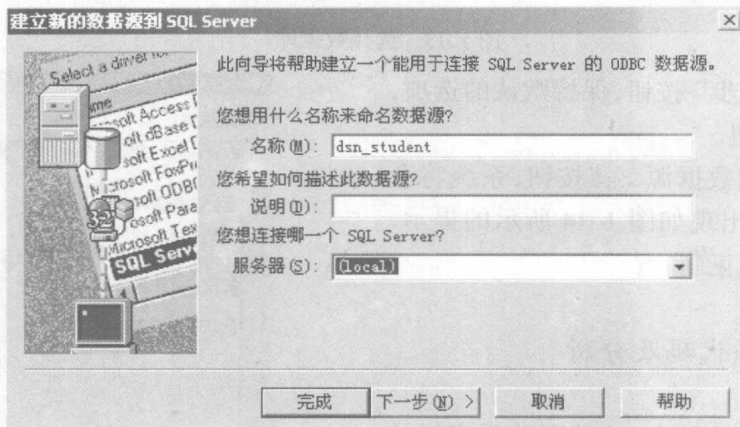


图 1.11 填写数据源名称和服务器名称

⑤ 单击“下一步”按钮。选择“使用用户输入登录 ID 和密码的 SQL Server 验证(S)”。在下面的“登录 ID”中输入 sa,密码为空(SQL Server 安装完毕后,超级用户 sa 的密码默认为空)。当然,在这两个文本框中也可以输入 SQL Server 服务器的其他用户名和对应的密码,如图 1.12 所示。

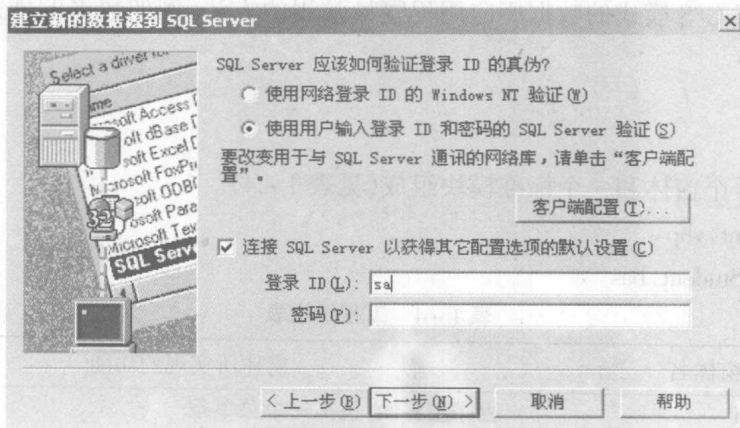


图 1.12 用户验证