

C#数据库应用程序 开发技术与案例教程

周洪斌 温一军 主编

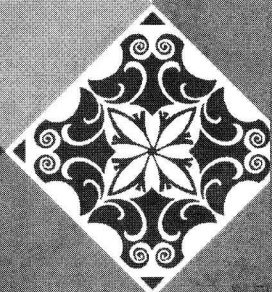


高等院校计

教材(应用型)

C#数据库应用程序 开发技术与案例教程

周洪斌 温一军 主编



机械工业出版社
China Machine Press

C#是微软公司推出的、业界主流的程序设计语言,掌握基于C#的数据库编程技术,对于在校学生、求职应聘者都具有极其重要的意义。本书以Visual Studio 2008、SQL Server 2005以及Access 2007为开发平台,较为全面地介绍了基于C#的数据库编程技术。本书以工作过程为导向,由浅入深,按篇展开,逐步揭开软件开发的神秘面纱。在基础篇中详细讲解了通讯录系统的设计、开发和安装部署的完整过程,把ADO.NET编程技术融入各个任务中,开展手把手教学。学完基础篇,即可开发出一个功能较为完备的通讯录系统。在进阶篇中讲解了企业开发中广泛用到的参数化SQL语句、存储过程使用及自定义数据库操作类,进一步提高读者的数据库编程能力。在高级篇中,首先详细讲解了基于三层架构的通讯录程序的开发过程以及如何开发一个基于Access的通讯录系统,随后讲解了如何实现基于抽象工厂模式的三层架构以及如何采用MD5加密用户密码。本书同时将“毕业生实习与就业跟踪系统”的开发作为独立实践任务安排在每章结尾,有利于读者课外练习,并在附录中给出了一个来源于企业实际项目的课程实训任务书。

本书是校企合作的成果,凝聚了编者近几年来学习.NET技术、开发.NET软件、讲授.NET课程的心得和体会,希望本书能为我国的软件人才培养起到推动作用,同时也希望读者能凭借本书进入.NET开发大门。

本书可作为应用型本科、高职高专院校相关专业C#程序设计的教材,亦可作为软件公司对新员工开展岗前培训的教材,还适合社会培训机构作为培训用书。另外,本书可供编程爱好者学习、参考。

封底无防伪标均为盗版

版权所有,侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目(CIP)数据

C#数据库应用程序开发技术与案例教程 / 周洪斌, 温一军主编. —北京: 机械工业出版社, 2012.7

(高等院校计算机专业人才培养规划教材(应用型))

ISBN 978-7-111-38622-3

I. C… II. ①周… ②温… III. C语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2012)第114649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: 刘 莎

北京诚信伟业印刷有限公司印刷

2012年8月第1版第1次印刷

185mm×260mm·12.25印张

标准书号: ISBN 978-7-111-38622-3

定价: 23.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

客服热线: (010) 88378991; 88361066

购书热线: (010) 68326294; 88379649; 68995259

投稿热线: (010) 88379604

读者信箱: hzjsj@hzbook.com

前言

为什么要写这本书

编者在高校任教多年，多次担任软件企业岗前培训班讲师，发现不少高校学生虽然学过程序设计语言，但没有掌握基本的数据库编程技术，更谈不上设计企业应用软件。究其原因，众多的程序设计教材从基础讲起，对高级部分涉及甚少，对数据库编程的讲解更是“蜻蜓点水”，一带而过，但众多的企业应用软件却需要和数据库“打交道”。学生学完课程后，只会拖拉几个控件，设计几个简单的窗体，根本不会设计真正的应用软件。本书以工作过程为导向，围绕开发基于WinForm的通讯录程序，以典型工作任务为主线，分为基础篇、进阶篇、高级篇，由浅入深介绍基于C#的数据库编程技术，逐步揭开软件开发的神秘面纱，让读者领略编程之美。希望本书能把初学者引入软件开发的大门。

为什么要把通讯录开发作为项目案例

在日常生活中，我们和朋友、亲戚联系，会广泛用到通讯录。通讯录程序的业务流程简单，易于理解。在整个开发过程中，可以使我们专注于程序编写，侧重于具体的数据库编程技术，而不会过多地关注业务流程，避免“细节湮没主题”，防止初学者无从下手。通讯录程序“麻雀虽小，五脏俱全”。通过开发本系统，可以掌握数据库编程的各个要点，体验完整的项目开发流程。

如何学习软件开发

软件开发是一项非常注重实践的工作，不可能只凭看书、看视频就学会软件开发，必须扎扎实实、一行一行地编写代码，不断积累项目经验，才能真正掌握编程技术。所以读者一定要亲自上机编写程序，才能真正掌握软件开发技术。本书将“毕业生实习与就业跟踪系统”的开发作为独立实践任务安排在每章结尾，并在附录中给出了一个来源于企业实际项目的课程实训任务。读者如果能按照要求完成这些项目的开发，编程能力必将有一个质的飞跃！

本书有哪些内容

本书分为基础篇、进阶篇、高级篇，由浅入深地介绍了基于C#的数据库编程技术，所用的开发工具为Visual Studio 2008、SQL Server 2005以及Access 2007。读者只要使用Visual Studio 2005、SQL Server 2000以及Access 2000以上版本，均可顺利完成本书的学习。

在基础篇中，详细讲解了通讯录系统的设计、开发和安装部署这样一个完整过程，把ADO.NET编程技术融入各个任务中，对各个任务进行详细讲解，手把手教学。学完基础篇，即可开发出一个功能较为完备的通讯录系统。读者不仅能够掌握数据库编程的重要知识，提高编程能力，更重要的是能够体会到软件项目开发的乐趣，享受成功进行软件开发的成就感！在整个过程中，注重编码规范，强调能力的培养，在不知不觉中养成企业开发规范，使读

者更快地步入软件开发的大门。

在进阶篇中,介绍了企业项目开发中广泛用到的参数化SQL语句、存储过程使用及自定义数据库操作类,进一步提高读者的数据库编程能力。

在高级篇中,首先详细讲解了基于三层架构的通讯录程序开发过程以及如何开发一个基于Access的通讯录系统。三层架构已在软件项目开发中得到了广泛应用,而Access数据库则在中小型企业中得到了广泛应用。通过学习,读者的知识储备会得到进一步扩展,可以掌握三层架构和基于Access的数据库编程技术。本篇随后讲解了如何实现基于抽象工厂模式的三层架构,本任务的难度较大,要求读者具备较强的面向对象编程的思想,供学有余力的读者参考、学习。最后,介绍了如何采用MD5加密技术对用户密码加密。MD5加密技术也是实际项目开发中广泛用到的。通过MD5加密用户密码,可以提高程序的安全性。

考虑到部分读者的基础可能比较薄弱,本书还在附录A中介绍了如何在SQL Server 2005中启用“SQL Server”身份验证,在附录B中介绍了基本的SQL命令。为帮助读者积累项目开发经验,本书在附录C中提供了一个来源于企业实际项目的课程实训任务。希望本书在内容上的精心安排,能够适应不同层次读者的需求。

本书是校企合作的成果,由沙洲工学院的周洪斌、温一军老师担任主编,参加编写工作的还有沙洲工学院的易顺明、许礼捷、董袁泉、陆国浩、杨佩理老师以及江苏国泰新点软件有限公司高级工程师曹立斌、黄素龙、李强。全书由周洪斌统稿。本书凝聚了编者近几年来学习.NET技术、开发.NET软件、讲授.NET课程的心得和体会,希望本书能为我国的软件人才培养起到推动作用,希望读者能凭借本书进入.NET开发的大门。

尽管我们在编写方面做了很大努力,但由于水平有限,不当之处在所难免,恳请读者批评指正。

电子邮箱: itzhb@qq.com。

技术博客: <http://www.cnblogs.com/zhoubh/>。

编 者

2012年7月

目 录

前言

第一篇 基础篇

任务1 系统分析与设计	2
1.1 通讯录程序简介	2
1.2 数据库设计	3
1.2.1 数据表结构	3
1.2.2 创建数据库	4
小结	7
独立实践任务	7
任务2 用户登录	8
2.1 连接数据库	8
2.1.1 数据库连接程序	8
2.1.2 SqlConnection对象	8
2.1.3 加入异常处理	9
2.1.4 使用using语句	10
2.2 设计用户登录模块	10
2.2.1 设计用户登录界面	10
2.2.2 实现登录功能	10
2.2.3 SqlCommand对象	11
2.2.4 SqlDataReader对象	12
2.2.5 调试数据库应用程序	14
2.3 建立并读取应用程序配置文件	14
2.3.1 建立应用程序配置文件	14
2.3.2 读取应用程序配置文件	15
小结	16
独立实践任务	16
任务3 主窗体设计	17
3.1 设计主窗体	17
3.2 窗体间数据传递	18
3.2.1 新建UserHelper类	18
3.2.2 修改登录代码	18
3.2.3 在主窗体上显示用户名	19
3.3 主窗体其他代码	20
3.3.1 建立各功能模块窗体	20

3.3.2 处理菜单项单击事件	20
3.3.3 处理工具栏按钮单击事件	21
3.3.4 主窗体FormClosed事件代码	21
小结	22
独立实践任务	22
任务4 联系人分组管理	23
4.1 设计联系人分组管理窗体	23
4.2 显示分组信息	24
4.2.1 实现分组信息显示	24
4.2.2 DataSet对象	25
4.2.3 SqlDataAdapter对象	26
4.2.4 DataGridView控件	26
4.3 增加分组	27
4.3.1 设计新增分组窗体	27
4.3.2 实现分组新增功能	27
4.4 删除分组	29
4.5 修改分组信息	30
4.5.1 设计修改分组信息窗体	30
4.5.2 实现分组信息修改功能	31
小结	34
独立实践任务	34
任务5 联系人管理	35
5.1 设计联系人管理窗体	35
5.2 显示联系人信息	36
5.3 增加联系人	38
5.3.1 设计新增联系人窗体	38
5.3.2 实现联系人新增功能	39
5.4 删除联系人	43
5.5 修改联系人信息	44
5.5.1 设计修改联系人信息窗体	44
5.5.2 实现联系人信息修改功能	45
小结	50
独立实践任务	50
任务6 用户密码修改	51
6.1 设计用户密码修改窗体	51

6.2 实现密码修改功能	51	任务11 使用存储过程	83
小结	53	11.1 显示联系人信息	83
独立实践任务	53	11.1.1 编写存储过程	83
任务7 数据库备份与恢复	54	11.1.2 修改Fill方法	84
7.1 设计数据库备份与恢复窗体	54	11.2 新增联系人	85
7.2 实现数据库备份功能	55	11.2.1 编写存储过程	85
7.3 实现数据库恢复功能	57	11.2.2 修改FillGroup方法	85
小结	58	11.2.3 修改新增联系人的方法	86
独立实践任务	59	11.3 删除联系人	87
任务8 设计“关于”窗体	60	11.3.1 编写存储过程	87
8.1 “关于”窗体的设计	60	11.3.2 实现删除功能	87
8.2 修改相关信息	61	11.4 修改联系人	88
8.2.1 修改AssemblyInfo.cs文件	61	11.4.1 编写存储过程	88
8.2.2 程序集简介	62	11.4.2 实现修改功能	88
小结	64	11.5 调用包含输出参数及返回值的存储 过程	90
独立实践任务	64	11.5.1 编写存储过程	90
任务9 系统安装与部署	65	11.5.2 调用存储过程	91
9.1 准备软件部署	65	小结	92
9.1.1 部署前的准备工作	65	独立实践任务	92
9.1.2 了解.NET程序运行条件	66	任务12 使用自定义数据库操作类	93
9.1.3 .NET程序部署方式	68	12.1 编写数据库操作类	93
9.2 制作安装程序	68	12.1.1 读取数据库连接字符串	93
9.3 完成软件部署	72	12.1.2 编写ExecuteDataTable方法	93
小结	73	12.1.3 编写ExecuteReader方法	94
独立实践任务	73	12.1.4 编写ExecuteScalar方法	95
		12.1.5 编写ExecuteNonQuery方法	96
		12.2 使用SqlDbHelper类	97
		12.2.1 用户登录	97
		12.2.2 显示分组信息	98
		12.2.3 删除联系人分组	98
		12.2.4 增加联系人分组	99
		12.2.5 修改分组信息	100
		12.2.6 显示联系人信息	101
		12.2.7 删除联系人信息	102
		12.2.8 增加联系人信息	103
		12.2.9 修改联系人信息	104
		12.2.10 用户密码修改	106
		12.2.11 数据库备份与恢复	107
		小结	108
第二篇 进阶篇			
任务10 使用参数化SQL语句	76		
10.1 修改用户登录代码	76		
10.1.1 SQL注入攻击简介	76		
10.1.2 修改登录代码	76		
10.2 SqlParameter对象	77		
10.2.1 SqlParameter类构造方法	77		
10.2.2 SqlParameter类常用属性	78		
10.3 修改联系人分组管理代码	78		
10.3.1 新增分组	78		
10.3.2 删除分组	80		
10.3.3 修改分组信息	80		
小结	81		
独立实践任务	82		

独立实践任务	108
--------------	-----

第三篇 高级篇

任务13 开发基于三层架构的通讯录

程序	110
----------	-----

13.1 三层架构简介	110
-------------------	-----

13.1.1 常用的三层架构	110
----------------------	-----

13.1.2 三层架构的演变	111
----------------------	-----

13.2 搭建三层架构	112
-------------------	-----

13.2.1 创建整体解决方案	112
-----------------------	-----

13.2.2 添加各层之间的依赖关系	113
--------------------------	-----

13.3 编写三层架构代码	114
---------------------	-----

13.3.1 编写Model类库代码	114
--------------------------	-----

13.3.2 编写SQLDAL类库代码	117
---------------------------	-----

13.3.3 编写Common类库代码	127
---------------------------	-----

13.3.4 编写BLL类库代码	128
------------------------	-----

13.3.5 编写表示层代码	132
----------------------	-----

小结	139
----------	-----

独立实践任务	139
--------------	-----

任务14 开发基于Access的通讯录程序

14.1 数据库设计	140
------------------	-----

14.1.1 创建数据库	140
--------------------	-----

14.1.2 创建数据表和关系	141
-----------------------	-----

14.2 系统实现	143
-----------------	-----

14.2.1 创建新的类库项目AccessDAL	143
--------------------------------	-----

14.2.2 编写Access数据库访问类	144
-----------------------------	-----

14.2.3 编写User.cs	148
------------------------	-----

14.2.4 编写ContactGroup.cs	149
--------------------------------	-----

14.2.5 编写Contact.cs	153
---------------------------	-----

14.2.6 编写BackupAndRestoreDb.cs	157
--------------------------------------	-----

14.2.7 修改业务逻辑层	158
----------------------	-----

小结	159
----------	-----

独立实践任务	159
--------------	-----

任务15 实现基于抽象工厂模式的三层

架构	160
----------	-----

15.1 创建新的类库项目IDAL	160
-------------------------	-----

15.2 修改SQLDAL类库代码	162
-------------------------	-----

15.3 修改AccessDAL类库代码	164
----------------------------	-----

15.4 新建DALFactory类库	165
---------------------------	-----

15.5 修改BLL代码	167
--------------------	-----

15.6 运行程序	168
-----------------	-----

小结	169
----------	-----

独立实践任务	169
--------------	-----

任务16 使用MD5加密用户密码

16.1 加密算法简介	170
-------------------	-----

16.1.1 数据加密简介	170
---------------------	-----

16.1.2 MD5简介	170
--------------------	-----

16.2 加密用户密码	170
-------------------	-----

16.2.1 编写MD5加密类	170
-----------------------	-----

16.2.2 实现用户密码加密	172
-----------------------	-----

小结	173
----------	-----

独立实践任务	173
--------------	-----

附录

附录A 在SQL Server 2005中启用“SQL	
-----------------------------	--

Server”身份验证	176
-------------------	-----

附录B 结构化查询语言SQL简介	180
------------------------	-----

附录C 课程实训任务书	183
-------------------	-----

参考文献	185
------------	-----

不积跬步，无以
至千里；不积小流，
无以成江海

第一篇

基 础 篇

本篇详细讲解了通讯录系统的设计、开发和安装部署这一完整过程，把ADO.NET编程技术融入到各个任务中，对各个任务进行详细讲解，手把手教学。学完基础篇，读者即可开发出一个功能较为完备的通讯录系统。通过对本篇的学习，读者不仅能够掌握数据库编程的重要知识、提高编程能力，更重要的是能够体会到软件项目开发的乐趣，享受成功进行软件开发的成就感！在整个过程中，读者要特别注重编码规范，强调能力的培养，让自己在不知不觉中步入软件开发的大门。

本篇内容

- 任务1 系统分析与设计
- 任务2 用户登录
- 任务3 主窗体设计
- 任务4 联系人分组管理
- 任务5 联系人管理
- 任务6 用户密码修改
- 任务7 数据库备份与恢复
- 任务8 设计“关于”窗体
- 任务9 系统安装与部署

任务1 系统分析与设计

本书所有的内容都是围绕开发通讯录程序来展开的。在日常生活中，我们和朋友、亲戚联系，会广泛用到通讯录。通讯录程序的业务流程简单，易于理解。在整个开发过程中，可以使我们专注于程序编写，侧重于具体的数据库编程技术，而不会过多地关注业务流程，避免“细节湮没主题”，防止初学者无从下手。通讯录程序“麻雀虽小，五脏俱全”。通过开发本系统，读者可以掌握数据库编程的要点，体验完整的项目开发流程。

本章将首先介绍通讯录程序的功能，然后使用SQL Server 2005设计通讯录程序数据库AddressList。

1.1 通讯录程序简介

通讯录程序的业务流程比较简单，主要涉及联系人分组、联系人信息等。下面来介绍一下我们将要开发的通讯录程序的主要功能。

本系统主要功能如图1-1所示。

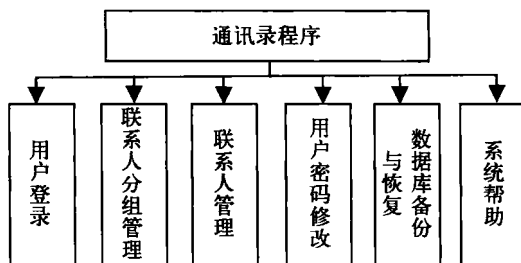


图1-1 系统功能模块图

1. 用户登录

用户登录是数据库应用程序必不可少的功能。用户只有输入正确的用户名和密码后，才能进入系统主界面。

2. 联系人分组管理

为了方便管理，系统提供了对联系人进行分组的功能，可以把联系人归到不同的分组中，如同事、同学、网友。联系人分组管理模块可以显示分组列表，同时可以新增、删除、修改分组信息。

3. 联系人管理

联系人管理模块可以增加、删除、修改、查询联系人信息。联系人信息包括：联系人姓名、手机、电子邮件、QQ、工作单位、办公电话、家庭地址、家庭电话、所属分组等。

4. 用户密码修改

用户可以根据需要，修改自己的密码。

5. 数据库备份与恢复

数据库的备份与恢复对于数据库应用程序而言是至关重要的。一旦数据丢失，将影响程

序的正常运行。因此，本模块提供数据库的备份和恢复功能。

6. 系统帮助

本模块比较简单，主要是通过“关于”窗体显示产品名称、版本号、版权、公司名称等信息。

1.2 数据库设计

本节根据前面对通讯录程序功能的介绍，使用SQL Server 2005设计对应的数据库AddressList。

1.2.1 数据表结构

根据上一节的分析，通讯录程序数据库AddressList只需包含3个表，分别是存放系统用户的用户表User、存放分组信息的联系人分组表ContactGroup和存放联系人信息的联系人表Contact。各数据表结构说明如下。

1. 用户表User

用户表User用于保存系统用户信息，包含用户名、密码。其结构如表1-1所示。

表1-1 用户表User

序号	列名	数据类型	长度	标识	主键	允许空	说明
1	UserName	varchar	50		是	否	用户名
2	Password	varchar	50			否	密码

2. 联系人分组表ContactGroup

联系人分组表ContactGroup用于保存分组信息，包括分组编号、分组名称和备注。其结构如表1-2所示。

表1-2 联系人分组表ContactGroup

序号	列名	数据类型	长度	标识	主键	允许空	说明
1	Id	int	4	是	是	否	分组编号
2	GroupName	nvarchar	50			否	分组名称
3	Memo	nvarchar	200			是	备注

3. 联系人表Contact

联系人表Contact用于保存联系人信息，包括编号、联系人姓名、手机、电子邮件、QQ、工作单位、办公电话、家庭地址、家庭电话、备注、所属分组编号。其结构如表1-3所示。

表1-3 联系人表Contact

序号	列名	数据类型	长度	标识	主键	允许空	说明
1	Id	int	4	是	是	否	编号
2	Name	nvarchar	50			否	联系人姓名
3	Phone	varchar	11			是	手机
4	Email	nvarchar	50			是	电子邮件
5	QQ	varchar	20			是	QQ
6	WorkUnit	nvarchar	200			是	工作单位
7	OfficePhone	varchar	20			是	办公电话

(续)

序号	列名	数据类型	长度	标识	主键	允许空	说明
8	HomeAddress	nvarchar	200			是	家庭住址
9	HomePhone	varchar	20			是	家庭电话
10	Memo	nvarchar	200			是	备注
11	GroupId	int	4			否	分组编号

接下来我们将按照这些表结构来创建数据库AddressList及数据表。

1.2.2 创建数据库

1. 创建AddressList数据库

(1) 打开Microsoft SQL Server Management Studio Express, 在对象资源管理器中右击“数据库”, 在弹出的快捷菜单中选择“新建数据库”, 如图1-2所示。

(2) 在弹出的对话框中输入数据库名称AddressList, 单击“确定”按钮, 即可完成数据库的创建。

2. 建立数据表

(1) 展开AddressList数据库, 右击“表”, 选择“新建表”, 如图1-3所示。

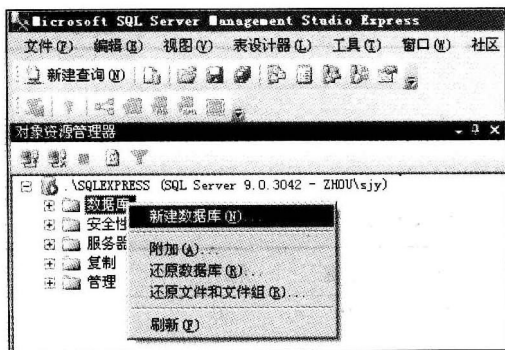


图1-2 新建数据库



图1-3 新建数据表

(2) 在弹出的表设计窗口中按照表1-1设计User表结构, 如图1-4所示。保存时输入表名为“User”。

(3) 按照表1-2设计ContactGroup表结构, 同时要设置Id列为标识列, 标识增量为1, 且设置Id列为主键, 如图1-5所示。保存时输入表名为“ContactGroup”。

(4) 按照表1-3设计Contact表结构, 同时要设置Id列为标识列, 标识增量为1, 且设置Id列为主键, 如图1-6所示。保存时输入表名为“Contact”。

3. 创建关系图

Contact表中的分组编号GroupId必须存在于ContactGroup表中, 所以必须为这两个表创建关系, 以确保数据的完整性。我们可以通过数据库关系图来创建表和表之间的关系。

(1) 展开AddressList数据库, 右击“数据库关系图”, 选择“新建数据库关系图”, 如图1-7所示。

表 - dba.User 摘要			
	列名	数据类型	允许空
1	UserName	varchar(50)	☐
2	Password	varchar(50)	☐

图1-4 User表结构

表 - dbo.ContactGroup 摘要			
列名	数据类型	允许空	
Id	int	☐	
GroupName	nvarchar(50)	☐	
Memo	nvarchar(200)	☒	

列属性	
数据类型	int
允许空	否
表设计器	
RowGuid	否
标识规范	是
(是标识)	是
标识增量	1
标识种子	1

图1-5 ContactGroup表结构

表 - dbo.Contact 摘要		
列名	数据类型	允许空
Id	int	☐
Name	nvarchar(50)	☐
Phone	varchar(11)	☒
Email	nvarchar(50)	☒
QQ	varchar(20)	☒
WorkUnit	nvarchar(200)	☒
OfficePhone	varchar(20)	☒
HomeAddress	nvarchar(200)	☒
HomePhone	varchar(20)	☒
Memo	nvarchar(200)	☒
GroupId	int	☐

列属性	
RowGuid	否
标识规范	是
(是标识)	是
标识增量	1
标识种子	1

图1-6 Contact表结构

如果出现对话框提示“此数据库没有有效所有者”，则可以参照附录A，先启用sa账号，然后在查询分析器中执行如下的SQL语句：

```
use AddressList
EXEC sp_changedbowner 'sa'
```

通过指定AddressList数据库所有者为sa后，即可新建数据库关系图。

(2) 出现“添加表”对话框后，选择需要建立关系的表，如图1-8所示。

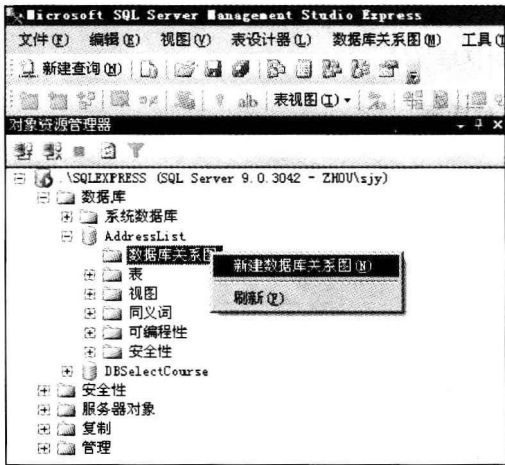


图1-7 新建数据库关系图

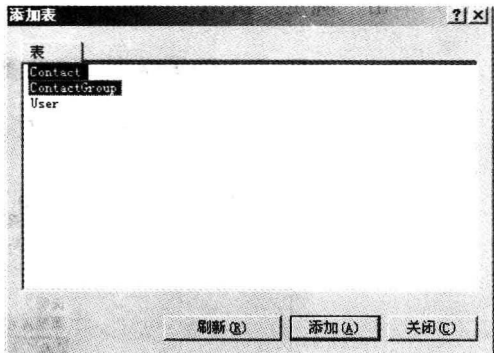


图1-8 选择需要建立关系的表

(3) 在图1-8所示的对话框中单击“添加”按钮，出现如图1-9所示的关系图。

(4) 选择ContactGroup表的Id列，按下鼠标左键不放，拖动至Contact表的GroupId列，弹出如图1-10所示的对话框。

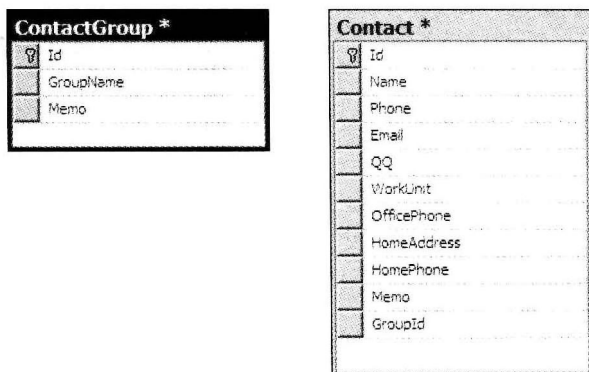


图1-9 关系图

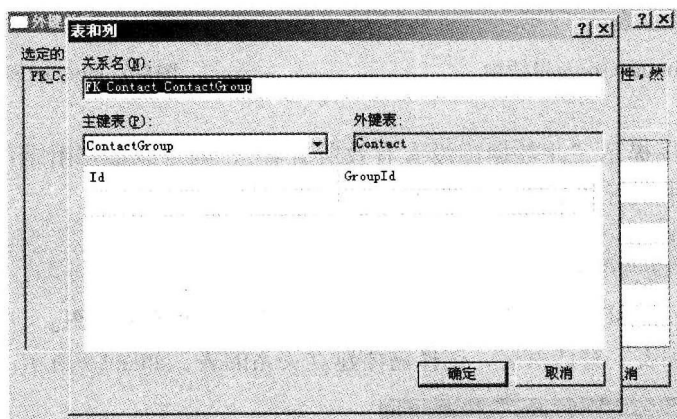


图1-10 创建关系对话框 (1)

(5) 单击“确定”按钮，出现如图1-11所示的对话框。

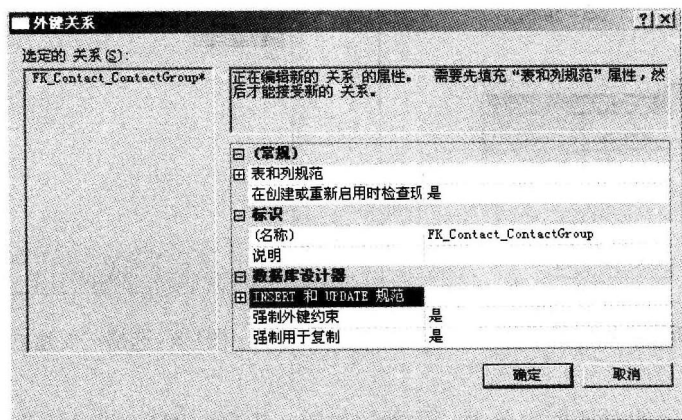


图1-11 创建关系对话框 (2)

(6) 单击“确定”按钮，出现如图1-12所示的最终数据库关系图。

到这里，我们已完成了通讯录程序的数据库设计，可以先在表中输入部分数据，以便在接下来的章节中编写代码访问数据。

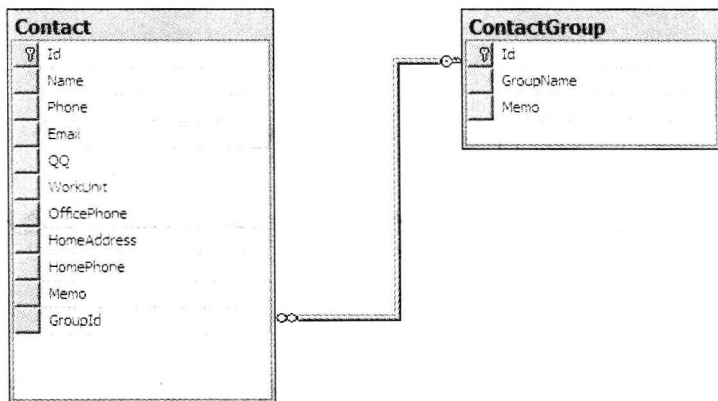


图1-12 最终的数据库关系图

小结

在本章中，我们介绍了通讯录程序的主要功能，并根据这些功能使用SQL Server 2005设计了对应的数据库AddressList及数据表，并创建了数据表之间的关系。在接下来的章节中将使用ADO.NET来访问和操作数据库。

独立实践任务

1. 完成“毕业生实习与就业跟踪系统”的分析与功能设计。

项目背景：高校每年都需要登记、跟踪学生实习与就业情况。目前，多数高校还停留在人工管理阶段。学生实习、就业情况由各班分别以电子表格的形式将信息上报，再由学校的相应部门对表格进行汇总。由于缺乏统一、高效的管理系统，存在着信息查询困难、更新繁琐与滞后等问题。通过开发毕业生实习与就业跟踪系统，可以对学生实习、就业情况进行统一管理，方便信息录入、查询、更新，便于学校不同部门及时掌握学生实习与就业信息，提高管理效率。

2. 根据“毕业生实习与就业跟踪系统”的分析与功能设计，在SQL Server中完成数据库及数据表的建立。

任务2 用户登录

用户登录是每一个数据库应用程序必不可少的功能模块。用户只有输入正确的用户名和密码，才能使用通讯录程序。

2.1 连接数据库

要编写程序操作数据库，首先要连接数据库。使用SqlConnection类可以连接到SQL Server数据库。

2.1.1 数据库连接程序

设计如图2-1所示的Windows程序，单击“连接”按钮，连接AddressList数据库。

(1) 首先在代码中添加对System.Data.SqlClient命名空间的引用：

```
using System.Data.SqlClient;
```

(2) 在“连接”按钮的单击事件中编写代码：

```
string connString = @"Data Source=.\sqlexpress;Initial Catalog=AddressList; Integrated Security=True";
SqlConnection connection = new SqlConnection(connString);
connection.Open();
MessageBox.Show("打开数据库连接成功");
connection.Close();
MessageBox.Show("关闭数据库连接成功");
```

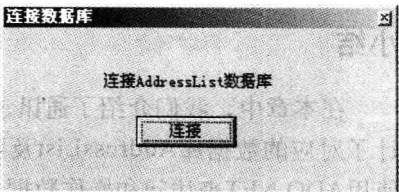


图2-1 数据库连接程序界面

运行程序，如果没有错误，单击“连接”按钮，将先后出现“打开数据库连接成功”对话框、“关闭数据库连接成功”对话框。

上面的代码中用到了SqlConnection对象，下面我们来介绍SqlConnection对象。

2.1.2 SqlConnection对象

使用SqlConnection类可以连接到SQL Server数据库。SqlConnection对象的主要属性和方法如表2-1所示。

表2-1 SqlConnection对象的主要属性和方法

属性和方法		说 明
属性	ConnectionString	连接字符串
方法	Open	打开数据库连接
	Close	关闭数据库连接

连接数据库主要分为以下3步。

(1) 定义连接字符串。

我们可以使用集成的Windows身份验证和SQL Server身份验证这两种不同的方式来连接SQL Server数据库。

使用Windows身份验证的数据库连接字符串如下：

```
Data Source=.\sqlexpress;Initial Catalog=AddressList;Integrated Security=True
```

说明：其中Data Source表示运行SQL Server的服务器，可以用“.”代表本机。由于我们这里用了Microsoft SQL Server Express Edition，所以用“.\sqlexpress”表示服务器；Initial Catalog表示所使用的数据库名，这里为本书所用的通讯录数据库AddressList；设置Integrated Security=True，表示采用集成的Windows身份验证。

使用SQL Server身份验证的数据库连接字符串如下：

```
Data Source=.\sqlexpress;Initial Catalog=AddressList;uid=sa;pwd=sa
```

说明：其中uid为指定的数据库用户名，pwd为指定用户对应的密码。

对启用SQL Server身份验证不熟悉的读者可以参看附录A。

(2) 创建SqlConnection对象。

可以使用定义好的连接字符串创建SqlConnection对象，代码如下：

```
SqlConnection connection = new SqlConnection(connString);
```

(3) 打开数据库连接。

调用SqlConnection对象的Open()方法打开数据库连接，代码如下：

```
connection.Open();
```

注意：数据库连接使用完毕后，要调用SqlConnection对象的Close()方法显示关闭数据库连接。代码如下：

```
connection.Close();
```

2.1.3 加入异常处理

访问数据库的过程中，可能出现数据库服务器没有开启、连接中断等异常情况。为了使应用程序能够处理这些突发情况，需要进行异常处理。.NET提供了try...catch...finally语句块来进行异常处理。由于数据库连接必须显示关闭，所以我们可以把关闭数据库连接的语句写在finally块中。

加入异常处理后的“连接”按钮单击事件代码如下。

```
string connString = @"Data Source=.\sqlexpress;Initial Catalog=AddressList;uid=sa;pwd=sa";
SqlConnection connection = new SqlConnection(connString);
try
{
    connection.Open();
    MessageBox.Show("打开数据库连接成功");
}
catch (Exception ex)
{
    MessageBox.Show(ex.ToString());
}
finally
{
    connection.Close();
}
```