

Mastering C# Database Programming

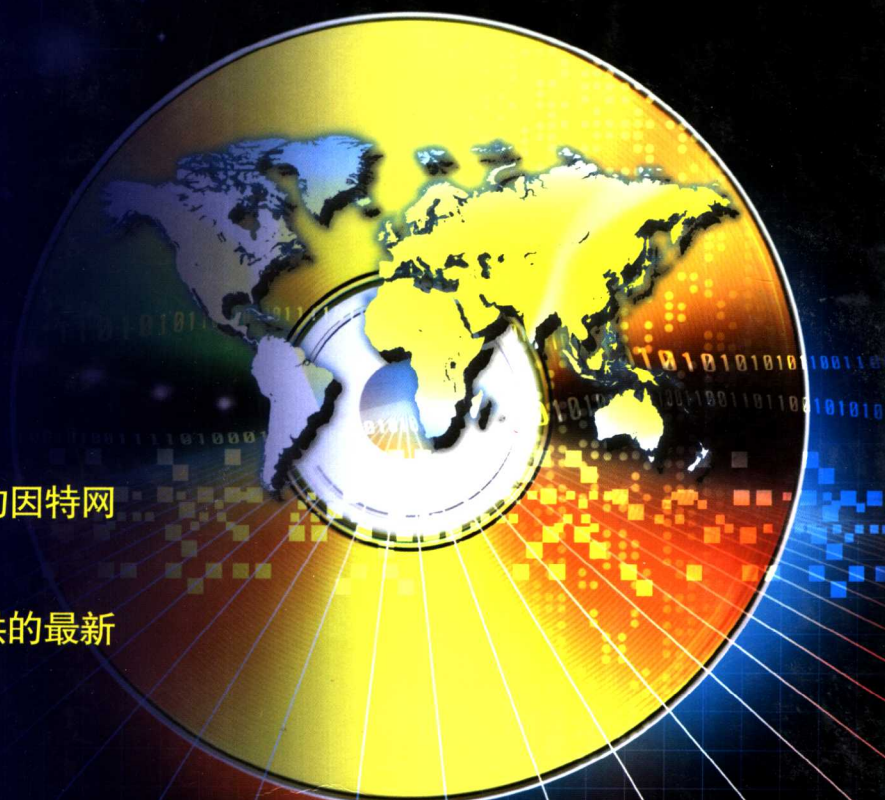
C# 数据库编程

从入门到精通

[美] Jason Price 著

邱仲潘 等译

- 使用C#构建基于XML的因特网数据库应用程序
- 深入介绍ADO.NET提供的最新数据访问功能



电子工业出版社

Publishing House of Electronics Industry

<http://www.phei.com.cn>

Mastering C# Database Programming

C#数据库编程 从入门到精通

〔美〕 Jason Price 著

邱仲潘 等译

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 提 要

.NET已经成为下一波技术应用的热门平台,是编写许多应用程序类型的全新框架,其优点是一开始就面向分布式环境。本书的重点是介绍如何编写与数据库互动的C#程序。全书共分三大部分:第一部分简单介绍了ADO.NET与数据库,包括SQL、Transact-SQL、ADO.NET类和Windows应用程序;第二部分介绍了基本的数据库编程,内容涵盖连接数据库的细节,数据库命令,运用各种对象读取、存储和修改数据;第三部分讲述了ADO.NET高级数据编程,涉及高级事务控制、ASP.NET、SQL Server对XML的支持和Web服务。

本书适用于对C#有所了解的数据库开发人员、应用程序开发人员和Web开发人员。



Copyright©2003 SYBEX Inc., 1151 Marina Village Parkway, Alameda, CA 94501. World rights reserved. No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photograph, magnetic or other record, without the prior agreement and written permission of the publisher.

本书英文版由美国SYBEX公司出版,SYBEX公司已将中文版独家版权授予中国电子工业出版社及北京美迪亚电子信息有限公司。未经许可,不得以任何形式和手段复制或抄袭本书内容。

版权贸易合同登记号: 01-2003-1085

图书在版编目(CIP)数据

C#数据库编程从入门到精通/(美)普赖斯(Price, J.)著;邱仲潘等译.—北京:电子工业出版社, 2003.9

书名原文: Mastering C# Database Programming

ISBN 7-5053-8968-8

I. C… II. ①普… ②邱… III. C语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2003)第065927号

责任编辑: 春 丽

印 刷: 北京天竺颖华印刷厂

出版发行: 电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路173信箱 邮编: 100036

北京市海淀区翠微东里甲2号 邮编: 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 33.625 字数: 860 千字

版 次: 2003年9月第1版 2003年9月第1次印刷

定 价: 55.00元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换,若书店售缺,请与本社发行部联系。联系电话: (010) 68279077

谨将本书献给我已去世的母亲Patricia Anne Price和我的家人，虽然你们已经离我而去，但你们仍在我心里。

致 谢

感谢Sybex公司所有出色的、辛勤工作的人们，包括Tom Cirtin、Denise Santoro Lincoln、Laura Ryan和Erica Yee。

译 者 的 话

本书翻译过程中得到了蔡维斌、陈志斌、洪鹰、杨威、周阳生、刘文红、邹能东、彭振庆、黄志坚、李耀平、江文清等同志的大力帮助，刘文琼、温连英、邱冬金、邱燕明等同志完成了本书的录入工作，刘云昌、刘昌和兄弟帮助进行了书稿与打印稿的校对，在此深表感谢。

邱仲潘

简介

欢迎使用《C#数据库编程从入门到精通》一书。众所周知，.NET已经成为下一波技术应用的热门平台。.NET的优点是一开始就面向分布式环境，换句话说，这个环境包括通过网络连接的计算机和设备。

说明：本书的重点是如何编写与数据库互动的C#程序。C#用ADO.NET与数据库互动，而ADO.NET是ADO的后续产品。本书介绍了与SQL Server数据库互动的细节，SQL Server是Microsoft公司的主要数据库软件。

Microsoft公司投入大量人力和资源，使.NET成为技术社会中的主流组件，谁也不能小视.NET。如果还想在今天和明天的行业中立足，你至少要学习.NET。

简而言之，.NET是编写许多应用程序类型的全新框架。可以用.NET编写的应用程序包括Windows应用程序和Web应用程序。可以用.NET开发由互联的服务构成的系统，通过Internet相互通信。

此外，还可以用.NET生成在手持电脑和移动电话等设备上运行的应用程序。尽管其他语言也可以开发这类应用程序，但.NET的设计考虑了互连网络。

.NET框架包括三大组件：

开发语言和工具 编写.NET程序的开发语言包括C#、Visual Basic.NET (VB.NET) 和 Managed C++。Microsoft公司的RAD（快速应用程序开发）工具Visual Studio.NET (VS.NET)可以在集成开发环境（IDE）中开发程序。本书使用C#和VS.NET。

公共语言运行环境（CLR） 公共语言运行环境管理运行代码并提供各种服务，如内存管理、线程管理（可以并行完成多个任务）和远程功能（使一个应用程序中的对象可以和另一个应用程序中的对象相互通信）。公共语言运行环境还可以强制实行可执行代码的安全性及精确性，保证不发生篡改。

框架基础类库 框架基础类库是Microsoft公司编写的丰富代码集合，可以在自己的程序中使用。例如，除了许多其他功能外，框架基础类库还可以开发Windows应用程序、访问磁盘目录与文件、与数据库互动、在网络上发送与接收数据。

本书适用对象

本书是为已经对C#有所了解的编程人员编写的，包含使用C#进行数据库编程时需要知道的一切。本书不要求读者有数据库方面的经验，但如果已经用过SQL Server与Oracle之类的数据库软件，则能够更快入手。

说明：如果你不熟悉C#，则建议阅读“Mastering Visual C#.NET”一书（Sybex公司，2002年出版，中译本《Visual C#.NET从入门到精通》已由电子工业出版社出版）。

如何使用本书

本书分成三个部分。第一部分“ADO.NET与数据库简介”介绍数据库知识，并介绍ADO.NET如何使C#程序与数据库互动。第二部分“ADO.NET基本数据库编程”介绍ADO.NET与C#程序知识。第三部分“ADO.NET高级数据库编程”介绍专业数据库开发人员所需要的编程技术。

下面几节详细介绍各章内容。

第一部分：“ADO.NET与数据库简介”

第1章“ADO.NET与数据库编程简介”介绍如何用ADO.NET使C#程序与数据库互动，介绍Microsoft公司的RAD工具Visual Studio.NET，最后还介绍如何使用.NET与SQL Server中丰富的Microsoft文档。

第2章“数据库简介”介绍什么是数据库，如何用其存储信息，并介绍了SQL Server数据库Northwind的用法。这个数据库包含假想的Northwind公司的信息，此公司销售食品。这个数据库是通常在SQL Server中安装的一个例子数据库。

第3章“结构化查询语言简介”介绍如何用SQL（结构化查询语言）访问数据库，介绍如何用SQL与Northwind数据库互动，如何取得和修改这个数据库中的信息。

第4章“Transact-SQL编程简介”介绍用Microsoft公司的Transact-SQL语言编程。Transact-SQL可以使编写的程序包含SQL语句和标准编程结构，如变量、条件逻辑、循环、过程与函数。

第5章“ADO.NET类概述”介绍ADO.NET类，还要介绍C#程序如何连接数据库、在本地存储行、切断数据库连接和在切断数据库连接的条件下读取本地行中的内容。这样存储从数据库中检索的行的本地拷贝是ADO.NET的主要强项之一。

第6章“Windows应用程序与ADO.NET简介”介绍Windows程序。Windows应用程序利用显示功能，并用鼠标和键盘进行输入。Windows提供了菜单、文本框、单选钮等图形项目，可以建立易于使用的图形界面。我们介绍如何建立与Northwind数据库互动的Windows应用程序。

第二部分：“ADO.NET基本数据库编程”

第7章“连接数据库”介绍连接数据库的细节。Connection类有三种：SqlConnection、OleDbConnection和OdbcConnection。SqlConnection类的对象连接SQL Server数据库；OleDbConnection类的对象连接支持OLE DB的数据库，如Access与Oracle；而OdbcConnection类的对象连接任何支持ODBC的数据库。与数据库的所有通信最终都是通过Connection对象完成的。

第8章“执行数据库命令”详细介绍执行的数据库命令。Command对象运行SELECT、INSERT、UPDATE或DELETE之类的SQL语句。Command对象还可以调用存储过程或从特定表中取得记录。

第9章“用DataReader对象读取结果”介绍如何用DataReader对象读取从数据库返回的结

果。可以用Command对象和DataReader对象读取从数据库返回的行。

第10章“用DataSet对象存储数据”介绍如何用DataSet对象存储从数据库返回的结果。DataSet对象可以存储数据库中的行拷贝，可以在切断数据库连接时处理这个拷贝。

第11章“用DataSet对象修改数据”介绍如何修改DataSet中的行，然后通过DataAdapter对象将这些改变推到数据库中。

第12章“导航与修改相关数据”介绍如何在表格中导航相关数据，在内存中进行数据改变，最终将这些改变推向数据库中。

第13章“使用DataView对象”介绍如何用DataView对象过滤和排序数据，DataView的优点是可以将其关联到Windows或ASP.NET应用程序中的图形组件。

第三部分：“ADO.NET高级数据库编程”

第14章“高级事务控制”介绍使用SQL Server与ADO.NET的高级事务控制。

第15章“Web应用程序简介：ASP.NET”介绍了ASP.NET基础，以及如何用Visual Studio.NET生成ASP.NET应用程序。

第16章“使用SQL Server的XML支持”介绍了SQL Server对XML的扩展支持，以及如何用XmlDocument与XmlDataDocument对象在C#程序中存储XML。

第17章“Web服务”介绍了如何建立简单Web服务，这是个软件组件，可以在Web上使用。例如，可以建立一个Web服务，让一个公司用XML在Web上向另一公司发订单。

下载例子程序

本书有许多例子程序，演示文中介绍的概念，并且标上了清单号和标题，例如：

清单1.1: FirstExample.cs

文件名就是清单名，例如清单1.1中的文件名为FirstExample.cs。可以从Sybex Web站点 www.sybex.com 下载包含程序的Zip文件。可以用WinZip之类程序打开这个Zip文件的内容。

打开这个Zip文件的内容时，每章建立一个目录，每个目录包含下列子目录：

programs 包含C#程序。

sql 包含SQL脚本。

VS. NET projects 包含Visual Studio.NET项目。

xml 包含XML文件。

说明：并不是每一章都有C#程序、SQL脚本，等等，因此不一定都有上述子目录。

目 录

第一部分 ADO.NET与数据库简介	1
第1章 ADO.NET与数据库编程简介	1
取得所要软件	1
开发第一个ADO.NET程序	2
连接Access与Oracle数据库	8
Visual Studio.NET简介	12
使用.NET文档	18
使用SQL Server文档	20
小结	21
第2章 数据库简介	23
数据库简介	23
使用SQL Server	24
探索Northwind数据库	31
用Enterprise Manager建立查询	39
生成表格	40
小结	46
第3章 结构化查询语言简介	47
使用SQL	47
用Visual Studio.NET访问数据库	73
小结	76
第4章 Transact-SQL编程简介	77
Transact-SQL基础	77
使用游标	84
使用函数	87
创建用户定义函数	97
存储过程简介	101
触发器简介	104
小结	110

第5章 ADO.NET类概述	111
管理提供者与一般性数据集类	111
执行SQL SELECT语句并本地存储行	116
小结	122
第6章 Windows应用程序与ADO.NET简介	124
开发简单的Windows应用程序	124
使用Windows控件	134
如何用DataGrid控件访问数据库	134
用Data Form Wizard生成Windows窗体	140
小结	149
第二部分 ADO.NET基本数据库编程	151
第7章 连接数据库	151
了解SqlConnection类	151
用SqlConnection对象连接SQL Server数据库	152
用Visual Studio.NET生成Connection对象	162
小结	165
第8章 执行数据库命令	167
SqlCommand类	167
生成SqlCommand对象	168
执行SELECT语句和TableDirect命令	170
执行修改数据库信息的命令	186
事务简介	192
向命令提供参数	194
执行SQL Server存储过程	199
用Visual Studio.NET生成Command对象	208
小结	210
第9章 用DataReader对象读取结果	211
SqlDataReader类	211
生成SqlDataReader对象	213
从SqlDataReader对象中取行	214
返回强类型列值	217
读取null值	229
执行多条SQL语句	230
在Visual Studio.NET中使用DataReader对象	235
小结	238

第10章 用DataSet对象存储数据	239
SqlDataAdapter类	239
DataSet类	243
用DataSet对象读取和写入XML	263
映射表与列	269
用强类型DataSet类读取列值	272
用Visual Studio.NET生成DataAdapter对象	277
用Visual Studio.NET生成DataSet对象	280
小结	282
 第11章 用DataSet对象修改数据	283
DataTable类	283
DataRow类	285
DataColumn类	286
在DataTable与DataColumn对象中增加限制	287
在DataTable中寻找、过滤与排序行	301
修改DataTable中的行	307
取得新的标识列值	317
用存储过程增加、修改与删除数据库行	319
自动生成SQL语句	329
DataAdapter与DataTable事件简介	331
处理更新故障	338
在DataSet中使用事务（SQL）	342
用强类型DataSet修改数据	343
小结	344
 第12章 导航与修改相关数据	346
UniqueConstraint类	346
生成UniqueConstraint对象	347
ForeignKeyConstraint类	348
生成ForeignKeyConstraint对象	349
DataRelation类	351
生成与使用DataRelation对象	352
增加、更新与删除相关行	356
更新父行主键的问题	364
嵌套XML	368
用Visual Studio.NET定义关系	373
小结	377

第13章 使用Data View对象	378
Data View类	378
生成与使用Data View对象	380
使用默认排序算法	383
进行高级过滤	384
DataRowView类	384
寻找Data View中的DataRowView对象	385
从Data View中增加、修改与删除DataRowView对象	389
生成子Data View对象	394
Data ViewManager类	396
生成与使用Data ViewManager对象	397
用Visual Studio.NET生成Data View	399
小结	401
第三部分 ADO.NET高级数据库编程	403
第14章 高级事务控制	403
SqlTransaction类	403
设置保存点	404
设置事务隔离级	408
了解SQL Server锁	416
小结	428
第15章 Web应用程序简介: ASP.NET	429
用VS.NET生成ASP.NET Web应用程序	429
Web窗体控件	435
建立更复杂的应用程序	437
用DataGrid控件访问数据库	441
用DataList控件访问数据库	450
维护Web应用程序状态	457
生成简单的购物推车应用程序	464
小结	469
第16章 使用SQL Server的XML支持	470
使用SQL Server FOR XML从句	470
XPath简介	479
XSLT简介	483
用HTTP访问SQL Server	486
使用SQL Server OPENXML()函数	494

用XmlDocument对象存储XML文档	497
用XmlDataDocument对象存储XML文档	502
小结	506
 第17章 Web服务	508
生成Web服务	508
浏览WSDL文件并测试Web服务	511
使用Web服务	516
注册Web服务	520
小结	524



第一部分 ADO.NET与数据库简介

第1章 ADO.NET与数据库编程简介

数据库是由表格组成的有组织的信息集合。每个表格进一步分成行和列，这些列存储实际信息。可以用结构化查询语言（SQL）访问数据库，这是大多数数据库软件都支持的标准语言，包括SQL Server、Access与Oracle。

本章介绍一个C#程序，连接SQL Server数据库、读取并显示表格中一行一列存储的内容，然后切断数据库连接。我们还将介绍连接Access与Oracle数据库的程序。

我们要介绍Microsoft公司的快速应用程序开发（RAD）工具Visual Studio.NET（VS.NET）。VS.NET可以在集成开发环境中开发、运行和调试程序。这个环境使用了Windows的所有奇妙特性，如鼠标和直观菜单，可以提高编程人员的生产率。

本章最后一节介绍如何使用.NET软件开发工具库（SAK）和VS.NET所带的丰富的Microsoft文档。熟悉ADO.NET和C#之后，你会发现这些文档非常有用。我们还将介绍如何使用SQL Server文档。

本章主要内容包括：

- 取得所要软件
- 开发第一个ADO.NET程序
- 连接Access与Oracle数据库
- Visual Studio.NET简介
- 使用.NET文档
- 使用SQL Server文档

取得所要软件

开发第一个C#程序之前，首先要安装.NET软件开发工具库（SDK）或VS.NET。要下载.NET软件开发工具库（SDK），可以到<http://msdn.microsoft.com/downloads>（搜索Microsoft.NET Framework Software Development Kit）；要购买VS.NET的测试版或完整版本，可以到Microsoft公司站点<http://msdn.microsoft.com/vstudio>。

要安装.NET软件开发工具库（SDK），运行下载的可执行文件，并按照屏幕上出现的指令将其安装到计算机上。要安装VS.NET，运行磁盘上的setup.exe文件并按照屏幕上出现

的指令操作。

还要取得SQL Server数据库软件。到编写本书时，可以到Microsoft公司站点<http://www.microsoft.com/sql>下载SQL Server的测试版，也可以到Microsoft公司Web站点购买SQL Server的测试版或完整版本。

本书使用SQL Server 2000软件的开发版，并使用其中的Northwind数据库。这个数据库中包含假想公司Northwind的信息，这家公司向客户销售食品。Northwind是个例子数据库，可以和SQL Server一起安装。Northwind数据库中的客户信息存储在Customers表中，本章稍后的例子程序中会显示这个表格的用法。

如果不想下载或购买SQL Server的测试版，则.NET DSK（和VS.NET）带有一个独立的桌面数据库服务器Microsoft SQL Server 2000 Desktop Engine（MSDE 2000）。MSDE 2000中也有一个Northwind数据库版本，可以代替SQL Server中的Northwind数据库版本，但这时不能得到SQL Server的所有图形化管理工具。如果使用.NET SDK，要安装MSDE 2000，只要选择Start>Microsoft.NET Framework SDK>Samples and QuickStart Tutorials。如果使用VS.NET，要安装MSDE 2000，只要运行安装VS.NET时使用的setup.exe程序，选择MSDE 2000作为要安装的新特性。

说明：关于MSDE 2000的更多信息，参见<http://www.microsoft.com/sql/techinfo/development/2000/msde2000.asp>。

开发第一个ADO.NET程序

本节要开发第一个ADO.NET程序，用一个C#程序完成下列工作：

1. 连接SQL Server Northwind数据库。
2. 从Customers表格中检索一行。
3. 显示这个行中的列。
4. 切断数据库连接。

本节介绍的许多概念将在后面几章详细介绍。目前先不必太过细究这些概念的所有细节，这些概念都将在后面几章中详细介绍。

清单1.1显示的例子程序放在文件FirstExample.cs中。

说明：可以从Sybex Web站点www.sybex.com下载本书所有程序的源代码文件。本书简介中介绍了下载这些文件的方法。下载这些文件之后，即可学习例子而不必亲自输入程序清单。

清单1.1: FirstExample.cs

```
/*  
FirstExample.cs illustrates how to:  
1. Connect to the SQL Server Northwind database.  
2. Retrieve a row from the Customers table using  
   a SQL SELECT statement.  
3. Display the columns from the row.  
4. Close the database connection.  
*/
```

```
using System;
using System.Data.SqlClient;
class FirstExample
{
    public static void Main()
    {
        try
        {
            // step 1: create a SqlConnection object to connect to the
            // SQL Server Northwind database
            SqlConnection mySqlConnection =
                new SqlConnection(
                    "server=localhost;database=Northwind;uid=sa;pwd=sa"
                );

            // step 2: create a SqlCommand object
            SqlCommand mySqlCommand = mySqlConnection.CreateCommand();

            // step 3: set the CommandText property of the SqlCommand object to
            // a SQL SELECT statement that retrieves a row from the Customers table
            mySqlCommand.CommandText =
                "SELECT CustomerID, CompanyName, ContactName, Address " +
                "FROM Customers " +
                "WHERE CustomerID = 'ALFKI'";

            // step 4: open the database connection using the
            // Open() method of the SqlConnection object
            mySqlConnection.Open();

            // step 5: create a SqlDataReader object and call the ExecuteReader()
            // method of the SqlCommand object to run the SELECT statement
            SqlDataReader mySqlDataReader = mySqlCommand.ExecuteReader();

            // step 6: read the row from the SqlDataReader object using
            // the Read() method
            mySqlDataReader.Read();

            // step 7: display the column values
            Console.WriteLine("mySqlDataReader['CustomerID'] = " +
                mySqlDataReader["CustomerID"]);
            Console.WriteLine("mySqlDataReader['CompanyName'] = " +
                mySqlDataReader["CompanyName"]);
            Console.WriteLine("mySqlDataReader['ContactName'] = " +
                mySqlDataReader["ContactName"]);
            Console.WriteLine("mySqlDataReader['Address'] = " +
                mySqlDataReader["Address"]);

            // step 8: close the SqlDataReader object using the Close() method
            mySqlDataReader.Close();

            // step 9: close the SqlConnection object using the Close() method
            mySqlConnection.Close();
        }
    }
}
```

```
}
catch (SqlException e)
{
    Console.WriteLine("A SqlException was thrown");
    Console.WriteLine("Number = " + e.Number);
    Console.WriteLine("Message = " + e.Message);
    Console.WriteLine("StackTrace:\n" + e.StackTrace);
}
}
```

下面逐行介绍FirstExample.cs程序的内容。开始几行是注释语句，表示程序的作用：

```
/*
    FirstExample.cs illustrates how to:
    1. Connect to the SQL Server Northwind database.
    2. Retrieve a row from the Customers table using
       a SQL SELECT statement.
    3. Display the columns from the row.
    4. Close the database connection.
*/
```

接下来的两行用using语句表示程序中引用的名字空间：

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

System名字空间是根名字空间，引用这个名字空间后，就可以在程序中使用Console.WriteLine()调用，而不必使用完全限定System.Console.WriteLine()调用。System.Data.SqlClient名字空间包含SQL Server使用的ADO.NET类，并包括后面程序中用到的SqlConnection、SqlCommand与SqlDataReader类。稍后将介绍这些类，本书将详细介绍ADO.NET类的全部细节。

为了处理代码中可能抛出的异常，我们把代码放在try/catch块中。注意Main()方法在try/catch块中放了九个步骤，用catch块处理try块内的代码中可能抛出的异常对象SqlException。下面先介绍这九个步骤，然后将在“处理异常”一节详细介绍这些异常。

第1步：建立SqlConnection对象以连接数据库

我们用SqlConnection类的对象连接SQL Server数据库。Main()方法中第一步建立SqlConnection对象mySqlConnection以连接SQL Server Northwind数据库：

```
SqlConnection mySqlConnection =
    new SqlConnection(
        "server=localhost;database=Northwind;uid=sa;pwd=sa"
    );
```

传入SqlConnection构造函数的字符串称为连接字符串，包含下列要素：