



Beginning ASP.NET Databases using VB.NET

Written and tested for final release of .NET v1.0

ASP.NET

数据库入门经典

——VB.NET 编程篇

John Kauffman

Fabio Claudio Ferracchiati 等著 康博 译



清华大学出版社
<http://www.tup.com.cn>
<http://www.tup.tsinghua.edu.cn>



ASP.NET 数据库入门经典

——VB.NET 编程篇

John Kauffman

Fabio Claudio Ferracchiati

康 博

等著

译

清华大学出版社

(京)新登字 158 号

北京市版权局著作权合同登记号: 01-2002-3207

内 容 简 介

对于一个 Web 站点而言,最基本的要求就是能够动态地更新自己的内容,能够准确安全地传输和存储数据,这些都需要访问数据库。本书首先概述了什么是数据驱动的 Web 站点以及本书的开发环境;接着介绍了一些创建数据库的理论,并对数据库的连接以及其他主题进行了详细描述;之后讨论了 DataReader 和 DataSet 对象,如何添加、更新和删除记录,如何使用存储过程和类库以及一些性能问题等;最后提供了一个案例,将这些知识都应用到这个实际的应用程序中。

本书适用于具有一定的 Visual Basic .NET 和 ASP.NET 编程经验、熟悉 .NET Framework 和相关技术的开发人员,以及想要学习如何使用 ASP.NET 来编写数据驱动的 Web 应用程序的广大读者。

John Kauffman, Fabio Claudio Ferracchiati et al: Beginning ASP.NET Databases using VB.NET

EISBN: 1-86100619-5

Copyright© 2001 by Wrox Press Ltd.

Authorized translation from the English language edition published by Wrox Press Ltd.

All rights reserved. For sale in the People's Republic of China only.

Chinese simplified language edition published by Tsinghua University Press.

本书中文简体字版由英国乐思出版公司授权清华大学出版社出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

ASP.NET 数据库入门经典——VB.NET 编程篇/(美)考夫曼等著;康博译.—北京:清华大学出版社,2002

书名原文:Beginning ASP.NET Databases using VB.NET

ISBN 7-302-05944-6

I .A... II.①考...②康... III.主页设计-应用软件,ASP.NET IV.TP393.092

中国版本图书馆 CIP 数据核字(2002)第 077182 号

出 版 者:清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑:徐燕萍

封面设计:康博

版式设计:康博

印 刷 者:北京昌平环球印刷厂

发 行 者:新华书店总店北京发行所

开 本:787×1092 1/16 印张:25 字数:640 千字

版 次:2002 年 10 月第 1 版 2002 年 10 月第 1 次印刷

书 号:ISBN 7-302-05944-6/TP·3535

印 数:0001~5000

定 价:46.00 元

出版者的话

随着国际互联网的快速崛起和迅猛发展，计算机之间的互联需求越来越迫切，而目前计算机硬件设备的不兼容性严重束缚了互联网的发展，引发了新一轮的跨平台软件的开发浪潮。软件商纷纷推出新的战略规划和解决方案，Microsoft 提出的 .NET 战略就是其中的经典之作。

在经历这场浪潮的洗涤和考验过程中，全球的软件开发人员都迫切需要了解新的软件技术和开发思路。为了满足国内 IT 从业人员的需要，清华大学出版社从 Wrox 出版公司引进了若干套编程系列丛书，“入门经典”系列是其中不可或缺的入门之作。作为世界著名的编程技术图书的出版公司，Wrox 推出的这套“入门经典”系列丛书主要面向编程的初学者、需要了解 .NET 策略的程序员，以及需要迅速掌握多门编程工具的程序员。该丛书依旧秉承了 Wrox 公司“由程序员为程序员而著 (Programmer to Programmer)”的创作理念，每本书均由世界顶级的编程高手执笔。他们站在资深程序员的高度，循序渐进地为初学者讲述了 .NET 的理念和构架、编程基本思想、编程语言基础、程序的控制和调试、Windows 应用程序的开发、对象编程技术、数据库访问技术、Web 程序开发和 .NET 构架应用等最新的软件开发知识，同时辅以大量操作性强的程序为示例，为读者提供了清晰的编程思路和宝贵的编程经验。

为了保证该系列丛书的质量，清华大学出版社迅速组织了一批位于 IT 开发领域前沿的专家学者进行翻译，经过编辑人员的进一步加工整理后，现陆续奉献给广大读者。

读者可以从 www.wrox.com 网站下载所需的源代码和获取相关的技术支持。同时，也欢迎广大读者参与 p2p.wrox.com 网站上的在线讨论，与世界各地的编程人员交流读书感受和编程体验。

前言

对于所有的 Web 站点而言，最重要的要求是准确而安全地发送和存储信息。这种信息可以是任何形式的，从信贷公司提供的信用卡资料到销售网点的民意投票结果。不管将 ASP.NET Web 页用于哪个方面，您迟早都会发现，访问、读取、写入数据，这些操作一般都与数据库有关。

幸运的是，处理 Web 应用程序的数据库比以前更容易。.NET Framework 改变了开发复杂的 Web 站点的方式。ASP.NET 对如何开发复杂和交互的 Web 站点作了重大的改进，ADO.NET 提供了强大而灵活的数据访问功能，从根本上改变了从 Web 应用程序中检索、处理和存储数据的方式。

本书将介绍如何创建数据支持的 ASP.NET 应用程序。根据一些实践经验进行编写，并列举了演示基础原理的代码示例。本书要求您对 ASP.NET 和 Visual Basic .NET 有一些基本的认识，对处理数据库可以没有任何经验，作者将指导每个过程，如将 ASP.NET 页面与数据库连接，并探讨各种读取、处理和更新数据的方法。接着就是研究所有涉及侧重于数据处理的 ASP.NET 编程问题，包括一些高级的主题，例如使用存储过程和组件。本书还进行案例分析，将前面几章讨论的内容都置于同一个环境中进行运用。

0.1 本书主要内容

第 1 章介绍了创建数据驱动的 Web 站点的一般性概念。特别是讨论了 Web 站点与数据之间的关系，在此过程中介绍了 ADO.NET。在这一章最后，还为本书的其他章节创建了一个开发环境，讨论了一些可能会遇到的困难。

到底什么是数据库呢？这似乎是个奇怪的问题。在第 2 章中，我们介绍了不同类型的数据库，如何设计它们，以及如何用它们存储和检索信息。接下来又概述了 SQL 语言，与 Microsoft 的数据库方案作了一个简单的比较。

了解了数据库后，就需要学习如何将它们与 ASP.NET 代码连接。第 3 章的开始对第 1 章提到的 ADO.NET 作了概述，阐明了需要知道的重要术语。接下来介绍了创建和测试对 Northwind 示例数据库的连接，本书将通篇使用该数据库。本章演示了大量例子，演示了如何连接不同的数据源，例如 Access、SQL Server、Excel 和 XML，以便在任何开发环境中都游刃有余。

第 4 章借助一些例子，运用 DataReader 对象读取和显示数据。首先讨论了在 ASP.NET 和 ADO.NET 中处理数据的原理，ADO.NET 引入了 Command 对象。之后，介绍了如何运用 DataReader 获取数据，接着将那些数据绑定给单选按钮、复选框控件以及最重要的 DataGrid。

第 5 章讨论了一个重要的 ADO.NET 对象：DataSet，它用于存储和检索服务器上任何复杂的数据。在介绍完原理之后，使用大量的例子来演示使用数据集的不同方法。

当然，我们并不总是读取其他人的数据，因此第 6 章介绍了如何在已有的数据库中创建自



己的记录。本章的主题包括 ASP.NET 的验证控件, 和使用 Command 和 DataSet 对象插入记录。

紧接着上一章, 第 7 章演示了如何修改和删除记录。介绍了如何用 DataGrid 和 DataView 控件编辑数据, 如何从 DataSet 对象中更新和删除记录。

接下来的第 8 章首先介绍了存储过程的概念、用途以及它们的一些优缺点。在演示完如何用 Visual Basic .NET 创建存储过程后, 这一章又讲述了如何从 ASP.NET 代码中调用它, 如何提供一些参数去执行特定的任务。

第 9 章介绍了一些更高级的主题, 这在创建商业数据驱动的 ASP.NET 应用程序时会碰到。这些主题包括错误处理、代码结构, 可伸缩性和并发, 以及安全性。

本书后面的第 10 章介绍了组件, 在讨论如何创建它们之前说明了它们的用途。这一章的示例代码主要是创建一个类库, 包括访问数据库的大量方法。

任何一个编程人员的目标之一就是尽可能编写最好的代码, 那么在第 11 章就讨论了性能问题。介绍了连接池和性能计数, 对 DataSet 和 DataReader 对象在不同情况下的相对性能作一些衡量。还重点说明了将旧的 ADO 代码转向 ADO.NET 代码的重要性。

最后, 第 12 章介绍了一个案例分析。它根据本书前面提到的所有内容, 构建了一个完整的应用程序, 使用了多个 ASPX 页面, ASP.NET Web 服务器控件, VB.NET 组件和存储过程。

本书读者对象

本书适合的读者包括:

- 有一定 Visual Basic .NET 和 ASP.NET 编程经验的开发人员
- 熟悉 .NET Framework 和相关技术的开发人员
- 想要学习如何使用 ASP.NET 来创建处理数据的 Web 应用程序

然而, 读者不需要了解任何有关数据库的知识就可以阅读本书。当然, 具备一些基础知识会更好一点。

客户配置

为了能够运行本书中的范例, 您必须在您的 PC 上运行如下软件:

- Windows NT、Windows 2000 或 Windows XP
- .NET Framework
- Visual Studio .NET 中的一个版本, 至少要有 Visual Basic .NET 的标准版

另外, 书中有一些使用 Microsoft Access 和 Microsoft SQL Server 的例子, 但是它们对读者的运行环境没有要求。所有重要的例子都使用 MSDE 作为它们的数据库引擎, 它是与 Microsoft Visual Studio .NET 的所有版本一起发布的。

客户支持

我们一贯重视听取读者的意见, 特别希望听到您对本书的反馈, 例如, 您喜欢什么, 不喜欢什么, 您认为我们下次如何才能做得更好。您可以将您的意见寄给我们, 或者发电子邮件到 feedback@wrox.com。请您在信中务必写清楚书名。

如何下载本书的范例代码

当您访问 Wrox 公司站点(地址为 <http://www.wrox.com/>)时, 通过 Search 工具或书名列表, 可以方便地定位所需要的书。然后, 单击 Code 列中的 Download 超链接, 或者单击本书的具体信息页面中的 Download Code 超链接, 就可以下载相应的范例代码。

从我们的站点上下载的文件都是使用 WinZip 压缩过的文档。将文件存入本地磁盘后, 需要使用解压缩程序(例如 WinZip 或 PKUnzip)来解压缩文件。在解压缩文件时, 通常将代码解压缩到每一章所在的文件夹中。在解压缩的过程中, 应确保软件被设置为使用文件夹名。

勘误表

我们已经尽最大努力确保本书中的文本和代码没有错误, 但是错误仍然在所难免。如果您发现本书存在错误, 例如拼写错误或不正确的代码段, 请给我们发来反馈信息, 我们将不胜感激。勘误表的发送可以节约其他读者学习本书的时间, 而且能够帮助我们提供更高质量的信息。请将您的反馈信息发送到 support@wrox.com。您的反馈信息如果正确, 将被粘贴到本书的勘误页面上, 或者在本书的后续版本中使用。

要在我们的站点上找到勘误表, 请访问 <http://www.wrox.com/>, 并通过 Advanced Search 或者书名列表定位本书页面。然后, 单击 Book Errata 超链接即可。

E-mail 支持

如果您希望直接向了解本书细节的专家咨询书中的问题, 可以发送电子邮件到 support@wrox.com, 您应该在邮件的主题中带上本书的书名和 ISBN 的后 4 位数字。一封典型的电子邮件应包括以下内容:

- 在主题部分中必须有本书的书名、ISBN 的后 4 位数字和所发现问题的页数
- 在信的正文部分应包括读者的名字、联系信息和问题

我们不会给您发垃圾邮件, 为节省大家的时间, 我们需要知道一些具体细节。当您发出电子邮件信息后, 它将经过下面一系列的处理:

- 客户支持: 首先, 您的信息将被转到我们的客户支持人员手中。他们手中有最常见问题的文档资料, 能够立即回答有关本书或者 Web 站点的任何常见问题。
- 编辑处理: 接着, 一些有深度的问题将被送到对本书负责的技术编辑手中, 他们在程序设计语言或者特定的产品上有着丰富的经验, 能够回答相关主题的详细技术问题。
- 作者处理: 最后, 如果编辑不能回答您的问题(这种情况很少发生), 他们将向本书的作者寻求答案。我们将尽量保护作者免受干扰, 以便不影响其写作。然而, 我们也非常高兴转寄给他们一些特殊的问题。所有 Wrox 公司的作者都为他们的书提供技术支持。他们将发送电子邮件给用户和编辑, 进而使所有的读者受益。

Wrox 公司的支持部门仅仅对那些与我们出版的书目内容直接相关的问题提供支持, 对于超出范围的问题, 您可以从 <http://p2p.wrox.com/> 论坛中的公共社区中获得支持信息。

p2p.wrox.com

如果想和作者及同行进行讨论, 请加入 P2P 邮件列表, 而且我们独一无二的系统除了能为



您提供一对一的 Email 支持外，还能通过邮件列表、论坛、新闻组使您实现与程序员之间的联系。如果您向 P2P 发送一个问题，请放心，它一定会得到 Wrox 公司作者和其他相关专家的帮助。无论您是在阅读本书，还是在开发自己的应用程序，都可以在 p2p.wrox.com 站点中找到许多对自己有所帮助的邮件列表。

要想订阅邮件列表，只需遵循以下的步骤：

- (1) 登录到 <http://p2p.wrox.com>/站点。
- (2) 从左边的主菜单栏选择适当的类别。
- (3) 单击希望加入的邮件列表。
- (4) 按照订阅说明填写自己的邮件地址和密码。
- (5) 回复您收到的确认邮件。
- (6) 使用订阅管理器加入更多的邮件列表并设置自己的邮件首选项。

本系统为什么能够为读者提供最好的客户支持

您可以加入邮件列表，也可以只接收每周的邮件摘要。如果您没有时间和工具来接收邮件列表，可以直接查找我们的在线文档。Lyris 系统可以将一些没用的垃圾邮件删除，并保护您的电子邮件地址不被侵扰。当存在加入和离开列表、以及任何有关列表的其他常见问题时，请将邮件发送到 listsupport@p2p.wrox.com。

目 录

第 1 章 在 Web 上显示数据	1
1.1 数据驱动的 Web 站点的优缺点	1
1.1.1 优点	1
1.1.2 缺点	2
1.2 数据源	3
1.2.1 数据库	3
1.2.2 文本文件	4
1.2.3 XML	4
1.2.4 其他数据源	5
1.3 从数据库中检索数据	5
1.4 ADO.NET 概述	7
1.4.1 Windows 平台上数据访问的历史	7
1.4.2 ADO.NET 体系结构	8
1.4.3 数据提供者	10
1.5 数据驱动的应用程序体系结构	11
1.5.1 数据层、业务层和表示层	11
1.5.2 用控件表示数据	13
1.6 Microsoft SQL Server Desktop Engine	14
1.6.1 小型的 SQL Server	14
1.6.2 获得和安装 MSDE	15
1.6.2 使用 MSDE	16
1.7 本章小结	17
第 2 章 关系数据库	18
2.1 数据库术语和概念	18
2.2 数据和实体	20
2.3 关系数据库	21
2.3.1 SQL 概况	23
2.3.2 Codd 的 12 条规则	24
2.4 SQL 简介	27
2.4.1 SELECT 语句	27
2.4.2 INSERT、UPDATE 以及 DELETE 语句	30
2.5 规范化	34



2.5.1 第一范式(1NF)	35
2.5.2 第二范式(2NF)	36
2.5.3 第三范式(3NF)	36
2.6 SQL 与 Access	37
2.7 本章小结	38
第 3 章 连接数据源	39
3.1 什么是连接	39
3.1.1 ADO 与 ADO.NET 连接之间的区别	40
3.1.2 ADO.NET 中可用的数据库连接	41
3.2 连接语法	42
3.2.1 使用 SQL Server .NET 数据提供者进行连接	42
3.2.2 使用 OLE DB .NET 数据提供者连接 MS Access	48
3.2.3 使用 OLE DB .NET 数据提供者连接 Excel	51
3.2.4 有关 Connection 对象的更多内容	56
3.2.5 连接 XML 数据源	57
3.3 获取模式信息(OLE DB)	62
3.4 更好的连接字符串	66
3.5 常见错误	67
3.6 本章小结	68
第 4 章 DataReader 对象、Command 对象和 Web 服务器控件	69
4.1 在 ADO.NET 和 ASP.NET 中处理和显示数据	69
4.1.1 Command 对象	70
4.1.2 DataReader 对象	71
4.2 使用 DataReader	72
4.3 在 ASP.NET 中使用基于数据的控件	76
4.4 列表框和下拉列表控件	78
4.5 使用单选按钮列表	85
4.6 使用复选框和列表框进行多项选择	88
4.6.1 显示多项选择控件	88
4.6.2 使用来自多个选项的数据	88
4.7 DataGrid 控件	95
4.7.1 表格的格式化设置	95
4.7.2 行的格式化设置	97
4.7.3 显示数据的更多方法	101
4.8 常见错误	102
4.9 本章小结	102

第 5 章 使用 DataSet 对象读取数据	104
5.1 DataSet 对象	104
5.1.1 DataSet 和 DataReader	105
5.1.2 DataTable 对象	106
5.1.3 数据传送的 6 个步骤	106
5.1.4 访问 DataSet 表中的行和列	110
5.1.5 处理多个数据源	115
5.1.6 使用 DataView 对象创建数据视图	131
5.2 DataGrid 列的属性	136
5.3 DataGrid 排序	142
5.4 DataGrid 分页	147
5.5 Calendar 控件和数据源	151
5.6 根据下拉列表中的值进行过滤	155
5.7 常见错误	162
5.8 本章小结	162
第 6 章 创建和插入记录	164
6.1 使用 ASP.NET 和 ADO.NET 插入数据	164
6.1.1 关于 INSERT 语句	164
6.1.2 主键码	164
6.1.3 外键码	165
6.1.4 强制性字段	166
6.1.5 格式化语法	166
6.2 使用 ADO.NET 插入新记录	167
6.3 ASP.NET 验证控件	170
6.4 使用 DataSet 对象插入记录	176
6.5 本章小结	190
第 7 章 更新和删除记录	191
7.1 更新数据库	191
7.1.1 SQL UPDATE 语句	191
7.1.2 使用 Command 对象更新记录	193
7.1.3 使用 DataSet 更新记录	197
7.2 删除数据	208
7.2.1 SQL DELETE 语句	208
7.2.2 使用 Command 对象删除记录	208
7.2.3 使用 DataSet 删除记录	209
7.3 验证数据	213
7.4 本章小结	216



第 8 章 使用存储过程	217
8.1 存储过程的优点	217
8.1.1 事务处理	217
8.1.2 速度	217
8.1.3 过程控制	218
8.1.4 安全性	218
8.1.5 减少网络通信	218
8.1.6 模块化	218
8.2 调用存储过程	219
8.3 创建存储过程	220
8.4 将参数传递到存储过程中	226
8.4.1 SQL 变量	227
8.4.2 从 Web 页面传递参数	229
8.5 存储过程输出参数	233
8.6 本章小结	239
第 9 章 数据驱动的 ASP.NET 应用程序	240
9.1 处理数据库错误	240
9.1.1 使用 @@ERROR	240
9.1.2 有效利用 @@ERROR	243
9.1.3 使用 RAISEERROR 命令引发数据库错误	245
9.1.4 在 .NET 中处理错误	248
9.2 事务处理	252
9.2.1 SQL 中的事务处理	254
9.2.2 ADO.NET 中的事务处理	258
9.3 代码的组织结构	261
9.4 数据安全性技巧	266
9.4.1 谨慎使用查询字符串的值	266
9.4.2 不要使用系统管理员(sa)账户	268
9.4.3 根据需要创建多个 SQL 用户	268
9.5 本章小结	269
第 10 章 组件化	270
10.1 组件化的含义	270
10.1.1 对组件的混乱理解	271
10.1.2 类库	271
10.2 使用类库的原因	271
10.2.1 简化编程工作	271
10.2.2 封装	272

10.2.3	重用性	273
10.2.4	增强的持久性	273
10.3	编写类库	273
10.3.1	Northwind 的业务要求	274
10.3.2	构造解决方案	274
10.3.3	在 Web 页中使用类库	281
10.3.4	小结	288
10.4	编写用于数据库访问的类库	289
10.4.1	设计用于数据库访问的类	289
10.4.2	概述 NwtLibrary	289
10.4.3	Category, Product 和 Supplier 类	291
10.4.4	在 Web 应用程序中使用类库	299
10.5	其他注意事项	303
10.5.1	存储过程与类库	304
10.5.2	兼容性	304
10.5.3	复杂性	305
10.5.4	记录文档	306
10.6	本章小结	306
第 11 章	性能	307
11.1	性能确定	307
11.2	连接池	308
11.2.1	OLE DB 和连接池	308
11.2.2	SQL Server 和连接池	309
11.2.3	其他池属性	312
11.3	性能计数器	313
11.4	DataSet、DataReader 和 Recordset	318
11.5	类型化的 DataSet 和后期绑定的 DataSet	321
11.6	微调 DataSet 和 DataReader	322
11.7	本章小结	325
第 12 章	构建数据驱动的 Web 应用程序	326
12.1	Wrox Auction 站点概述	326
12.1.1	基于 Web 的分类系统	326
12.1.2	安装应用程序	328
12.2	ASP.NET 页面流程	330
12.3	创建应用程序	331
12.3.1	创建新的应用程序	331
12.3.2	预加载的代码	332



12.3.3	设置配置信息.....	333
12.4	编写应用程序.....	333
12.4.1	主页.....	333
12.4.2	用户注册和登录.....	335
12.4.3	管理出售的产品.....	352
12.4.4	浏览和绑定.....	369
12.4.5	完成销售.....	379
12.5	增强应用程序.....	383
12.6	本章小结.....	384

第1章 在Web上显示数据

Web 首次出现时，人们必须用一个比喻来说明如何在 Web 上显示数据。如果看一下那个时期的 Web 站点例子，其内容大部分都是在传统媒体上所能看到的内容，如书籍、杂志和报纸。这使得 Web 与其他媒体的用途一样：和它们差不多同时提供了信息。当然，那样做并没有错，但它限制了 Web 的真正用途。

随着时间的推移，支持 Web 的技术日益发展成熟，Web 原来只能提供静态站点，但现在也能提供动态应用程序。这些应用程序允许用户选择他们感兴趣的信息，提供了定制的用户操作，可实时对它们进行修改。

这些应用程序的关键就是它们所包含的数据。不管什么内容——可能是一个产品目录、一组客户资料或是一个档案库——都是数据使得它们变成动态的。在过去，通过 Web 提供数据比通过传统的桌面应用程序更困难，由于使用的开发工具和功能，以及 Web 本身特性的缘故，使得用户不得不放弃这类应用程序和数据。但随着时代的发展，特别是 Microsoft 的 .NET Framework 的出现，这一情形得到了改观。现在，Web 应用程序开发人员可以与桌面应用程序开发人员平起平坐了。

本章将详细介绍数据驱动的 Web 站点，以及如何在 ASP.NET 中实现它们。首先讨论了数据驱动站点的优缺点，接着提供了这类数据的数据源。在这之后介绍了 .NET Framework 的数据访问技术——ADO.NET——包括它的体系结构、它的类以及它是如何适应数据驱动的应用程序的。最后介绍本书将一直使用的数据库服务器的安装。

说明：

数据驱动的 Web 应用程序是个显示动态数据的 Web 站点。用户操作会影响保存在数据存储器中的信息。

1.1 数据驱动的 Web 站点的优缺点

数据驱动系统的一些优点是明显的，但也有一些不是很明显。自然您可能因为某些原因不想将 Web 站点与数据库挂钩。本节将讨论创建基于数据库的 Web 站点的优缺点。

1.1.1 优点

创建数据驱动的 Web 站点有许多间接好处，例如可以重用其他项目中的功能，可以跨系统共享通用信息——这些在创建第二个或第三个应用程序时就可以实现。这里将介绍一些创建数据驱动的 Web 站点的直接好处。

- 内容的质量和及时性——创建数据驱动的站点，最直接的优点就是能及时将新信息显示



在 Web 上,能够放入控件中保证该信息的质量。每次价格改变或是添加新产品时,我们不必用 Web 设计器创建一个包含该信息的页面后将它再次上传,而是可以创建一个工具,通过简单地修改数据库来及时发布新的或更新的信息。这是 Web 优于传统媒体一个主要方面——可以实时查看信息,而不是原有的数据。通过一些强制规则,如谁可以添加和修改数据,如何检查数据以及数据是否被认可,可以在发布之前以更严格的方式对数据进行验证,以确保用户只看到正确的信息。

- 功能——将站点所需要的数据存储于数据库中的另一个主要优点就是得到改进的功能,这是就用户可以在系统上执行的动作而言。它不是生成“目录”(像这本书这样用索引和目录表作为搜索的手段),而是创建窗体以便允许用户指定要查找的内容,让系统从数据库中查询该信息。搜索引擎就是个很好的例子。如果没有数据库,这类站点只能提供其他 Web 站点的人工目录,需要用户在庞大的页面结构中导航。

- 维护——将站点的数据与表示代码分开存储,就不再需要在站点的相关部分间维护 HTML 文件中的静态链接,这样就不需要每次重新设计站点时重新设置许多页面的格式和菜单结构。在数据驱动的系统,Web 页一般是全部页面的模板,而不是为了显示一段信息就用一个页面。

举个例子,可以假定屏幕上显示的页面显示待售产品的信息。在数据驱动的系统,这并不是个单独的 HTML 页面,它是个包含字段和表的页面,而字段和表可以用与任何产品相关的数据填充。这意味着每次重新设计时只要做很少的工作。同样地,各种不同信息之间的关联也可以存储在数据库中(而不是在页面中硬编码),这样可以迅速实现到相关产品和其他信息的链接。

1.1.2 缺点

尽管创建数据驱动的站点有很多优点,但其中一些优点的代价很昂贵,它并不是解决问题的最好办法。为了给终端用户提供更好的服务,还有一些缺点必须克服,在创建之前考虑这些缺点还是很重要的。

- 开发——大部分数据驱动的 Web 站点一开始都是静态的,而如今仍然有许多静态站点要创建。有些需要显示的内容并不适合于数据驱动的站点,创建数据驱动的系统要求额外的时间和技巧,这样使得产品更复杂,更容易出错(不可避免的)。我们必须把这些代价与这类系统的优点作一下衡量。

- 性能——数据驱动的 Web 站点的性能也是个问题。如果站点是完全静态的,那么在组织系统时或是如何满足更多用户的需要时没有任何限制。提高性能的最简单的方法就是购买更快的处理器和更多的内存。当这些条件满足时,就可以创建多种站点,重定向到这些站点的用户的负载很小。这可以以线性形式继续下去,每次添加新的 Web 服务器,性能会得到同样的提高。

有了数据驱动的站点,情况就不是这样,因为整个系统都依赖于一个资源:数据库。如果没有谨慎设计的话,那么数据库会造成系统中的“瓶颈”,因此在等待检索时会使得应用程序停滞不前。而要解决这个瓶颈问题是件很困难的工作——用几个同步的数据库是解决这一问题的一个方法,但这代价很昂贵,而同步过程中的系统开销也是很大的。

● 代价——除了上面提到的技术问题，还有一些相关的商业问题。对于相对静态的站点来说，创建数据库和编写访问它的代码所需的时间比只编辑 HTML 页面要多。而企业级数据库系统很昂贵。仅考虑 Microsoft 的数据存储方案的话，那么众所周知，使用 SQL Server(Microsoft 的企业级数据库服务器)生成方案比 Access(Microsoft 的桌面数据库)的优点更多，例如更高的性能，对产业标准的更好支持，但也更昂贵。

1.2 数据源

读者现在已考虑了上面的一些问题，如果仍想编写数据驱动的 Web 应用程序，首先需要说明的就是最终出现在用户屏幕上的信息的来源。根据数据类型，对数据执行哪些操作以及系统使用量这些因素，信息会有多种来源。本节描述了使用 3 种最常用的数据源类型的原因，还对其他类型作了概述。

1.2.1 数据库

在开始考虑数据源时，最先想到的是数据库，它通常都提供了最可靠、伸缩性最大和最安全的数据存储。在处理大量数据时，数据库也提供了最好的性能。然而也有事实表明在有些情况下，它们并不是最好的选择。

一般来说，数据库存储数据的方式就是以任意的顺序检索任意数量的数据。对于小的数据集合来说，例如一组联系信息，那么创建和访问数据库时所用的时间及其他开销都可能会抹杀数据库所提供的优点。

下一章将更多地讨论数据库结构，但可以举一个很简单的例子，将公司雇员的一些信息存储在数据库中，首先需要创建 Employee 表，这个表可以包含许多雇员的同类型数据，这些信息可以包括他们的 EmployeeID(号码)、LastName、FirstName、BirthDate 和 Country 如图 1-1 所示。



图 1-1

注意：

本章将通过比较和演示来说明如何存储和显示数据。为了前后连贯，将使用同一个例子：那就是存储某公司雇员的资料。

在显示与其他数据源作比较的数据库时，要注意的一点就是它是基于被存储信息的模型，