

.NET开发经典名著

Beginning ASP.NET 4.5 Databases, Third Edition

ASP.NET 4.5

数据库入门经典(第3版)

//

[美] Sandeep Chanda 著
Damien Foggon 译
王榕 蔡松伯



Apress®

清华大学出版社

014017959

TP393.092.2
50

.NET 开发经典名著

ASP.NET 4.5 数据库 入门经典 (第3版)

[美] Sandeep Chanda 著
Damien Foggon
王 榕 蔡松伯 译



清华大学出版社

北 京



北航

C1705317

TP393.092.2

50

020511310

Beginning ASP.NET 4.5 Databases, Third Edition

By Sandeep Chanda, Damien Foggon

EISBN: 978-1-4302-4380-9

Original English language edition published by Apress Media. Copyright © 2013 by Apress Media.
Simplified Chinese-Language edition copyright © 2014 by Tsinghua University Press. All rights reserved.

本书中文简体字版由 Apress 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。

北京市版权局著作权合同登记号 图字：01-2013-8600

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目(CIP)数据

ASP.NET 4.5 数据库入门经典(第3版)/(美)查恩达(Chanda, S.), (美)福根(Foggon, D.) 著;
王榕, 蔡松伯 译. —北京: 清华大学出版社, 2014
(.NET 开发经典名著)
书名原文: Beginning ASP.NET 4.5 Databases, Third Edition

ISBN 978-7-302-34232-8

I. ①A… II. ①查… ②福… ③王… ④蔡… III. ①网页制作工具—程序设计
IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2013)第 247725 号

责任编辑: 王 军 韩宏志

装帧设计: 牛艳敏

责任校对: 邱晓玉

责任印制: 刘海龙

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 清华大学印刷厂

装 订 者: 北京市密云县京文制本装订厂

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 13.75 字 数: 370 千字

版 次: 2014 年 1 月第 1 版 印 次: 2014 年 1 月第 1 次印刷

印 数: 1~3000

定 价: 48.00 元

产品编号: 054979-01

译者序

ASP.NET 数据库编程的优势在于允许程序员便捷地将数据集成到页面，使程序员不必了解连接详情、命令以及 `DataReader` 和 `DataAdapter` 对象就能执行普通数据任务。对于初学者来说，ASP.NET 使得基本数据操作变得简单易学，而且初学者可借助 ASP.NET 完成较为复杂的数据应用任务。

从诞生至今，ASP.NET 的每个版本几乎都会推出十分重要的实用功能。例如，从 1.0 升级到 2.0 版本时，将创建 ASP.NET 页面所需的代码行数减少约 70%，大幅提高了开发效率；3.5 版本则新增了数据访问层设计技术 ADO.NET Entity Framework，方便了用户将对象模型转化为数据库对象；4.0 版本借鉴了开源阵营众多 MVC 框架思想，提供了 ASP.NET MVC 框架；在 ASP.NET 4.5 中，对数据模型绑定方式做了重大改进。

我们非常高兴能通过翻译本书，为国内读者带来关于使用 ASP.NET 4.5 开发工具建立数据驱动的 Web 应用程序所需的最新知识和技巧。书中介绍了如何使用 ASP.NET、ADO.NET 和 C# 建立数据驱动的 Web 应用程序。除了介绍构建 Web 窗体和将控件放入页面的基础知识外，还阐述了 asp.net 开发人员应如何构建能够处理数据库中数据的 Web 站点，如何将 Web 站点连接到各种不同数据源(不只是数据库)并使用数据动态生成页面内容。另外，本书还介绍了关系数据库的创建以及如何使用 SQL 与数据源通信。本书面向初学 ASP.NET、ADO.NET 和 C# 的开发人员和用户。对于想学习新版 .NET Framework 4.5 框架中的数据库知识的开发人员，本书也是极佳选择。

在翻译本书过程中，我们参阅了最新专业文档词典及著作，并求证于很多业内专家，专业术语尽量遵循约定俗称的标准译法，在仔细斟酌的基础上力求译文准确并忠于原著。由于自身水平有限，难免会有翻译不妥之处，敬请读者批评指正。蔡松伯翻译了本书的 1~7 章，王榕翻译了本书的第 8~13 章，并负责全书的审稿，参与翻译活动的还有孔祥亮、陈跃华、杜思明、熊晓磊、曹汉鸣、陶晓云、王通、方峻、李小凤、曹晓松、蒋晓冬、邱培强、洪妍、李亮辉、高娟妮、曹小震和陈笑。

王榕 蔡松伯

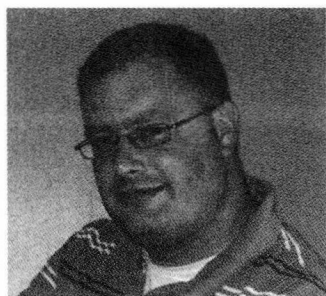
作者简介



Sandeep Chanda 是 Neudesic 的解决方案总监, Neudesic 是 Microsoft 的全美系统集成商和金牌合作伙伴。在过去 10 年中, Sandeep 一直在使用多种 Microsoft 技术(包括但不限于 .NET、Azure、BizTalk、SharePoint 和 Dynamics CRM), 构建跨多个行业的大型企业应用程序。他热爱技术, 在各种公司会议和公共网络研讨会上发言。Sandeep 在一个流行的在线开发杂志上就 Microsoft Dynamics CRM 4.0 发表了多篇文章, 他还是 *Microsoft Windows Identity Foundation Cookbook* (Packt Publishing, 2012) 一书的作者。最近一段时间, 他致力于推广应用程序生命周期管理(ALM)和使用 Team Foundation Server 2012 进行开发协作, 并就这一主题在 Great Indian

Developer Summit 2012 上发表了演讲。他还用了不少时间去各地为不同的团队培训 .NET Framework 4.5 的新功能和 Windows 8 Metro 应用程序开发。Sandeep 从 Pilani 的 BITS 大学获得了软件系统硕士学位。他关注的领域包括面向服务的计算、普适计算和触觉设备。他不定期在 <http://vstslive.wordpress.com> 上发表博客。可通过 sandeep.chanda@neudesic.com 联系他。

技术编辑简介



Michael Mayberry 使用 Microsoft 技术开发软件已经超过了 13 年。在此期间，他不断地采用新的解决方案和工具，解决日益复杂的问题。

Michael 目前是达拉斯沃斯堡区一家非盈利组织的软件架构师。他是一名集成专家，提供的解决方案能够让新软件与已有系统交互。他具有多个领域的经验，包括内容管理系统、数据仓库以及 CRM 系统等。

Michael 一直很注重团队组建，总是与他人分享知识。最近，写作和评审也成了他关注的领域。

在业余时间，Michael 喜欢与他美丽的妻子和四个孩子呆在一起。

如果有疑问或者想要对 Michael 说些什么，那么可以通过此地址联系他：michaelmayberry@hotmail.com。

前言

欢迎阅读本书。本书并不是上一版 *Beginning ASP.NET 2.0 Databases* 的简单修订版本。自上一版出版之后，.NET Framework 已经发布了多个版本，构建数据驱动的应用程序涉及的概念也发生了巨大变化。随着 .NET Framework 引入了 Entity Framework 5、WCF 数据服务和 ASP.NET 4.5 中的一些新功能(如强类型化控件和模型绑定)，开发人员在构建从各种数据源获取数据的 Web 应用程序时，有了更好的选择。本书将指导你使用 .NET Framework 4.5 中的新功能构建数据驱动的网站，还将介绍现代 Web 应用程序开发中的领先技术 ASP.NET MVC 4。

目标读者

对于仍在使用 .NET Framework 2.0，想要学习 .NET Framework 4.5 中新的数据访问功能的开发人员，本书是一个极佳选择。对于初学者，本书也是一本非常实用的指南，带领他们进入使用 ASP.NET 4.5 开发数据驱动网站的世界。本书不只介绍构建数据访问层的新方式，还介绍相关的最佳实践。

本书结构

本书共分为 13 章。

第 1 章是简介章节，概述了新的数据访问模式。该章还讨论了 ASP.NET 4.5 中可以使用的不同数据源，并介绍了 ADO.NET 的新功能。

第 2 章探讨了关系数据库的未来。该章介绍了 SQL Server 2012，并讨论了一些数据可视化功能。

第 3 章介绍了非关系数据库的概念，以及它们与在 .NET 中构建数据驱动应用程序有什么关系。该章将通过 MongoDB 介绍 NoSQL 概念。

第 4 章详细介绍了 ADO.NET，并使用 ADO.NET 执行一些数据库操作。该章讨论了一个重要的新功能：异步 CRUD 操作。

第 5 章概述了本书中将一直用到的 LINQ 功能。

第 6 章概述了实体数据模型。不仅第三方 ORM 工具会用到其基本概念，Entity Framework 也会使用它。该章还详细讨论了概念模式定义语言。

第 7 章探讨了 Entity Framework 使用实体数据模型的基本概念生成数据模型的方式。

该章还介绍了使用 LINQ 和实体 SQL 查询模型的不同方式, 以及一些数据访问接口。

第 8 章在 Entity Framework 中使用第 1 章讨论过的数据访问模式。该章还概述了在 ASP.NET Web Forms 中执行验证的数据注释。

第 9 章详细叙述了如何使用 REST 风格的架构, 通过 WCF 数据服务和 Web API 获取数据。该章还讨论了 OData 协议。

第 10 章介绍了 ASP.NET 4.5 和 ASP.NET MVC 4 中新的数据绑定功能。该章还讨论了数据表示功能的增强之处。

第 11 章讨论了 ASP.NET 4.5 中的动态数据概念, 以及如何使用动态数据快速构建一个原型以操作存储库中的数据。

第 12 章探讨了构建没有耦合的数据访问层的最佳实践, 并针对构建数据驱动网站时可能遇到的实际问题提供了一些有用的提示。

第 13 章介绍的知识可帮助更高效地开发数据驱动的应用程序。该章还讨论了如何有效利用 Team Foundation Server(TFS) 2012 对 Visual Studio 2012 所做的增强。

本书的出版需要感谢整个编辑团队, 特别是 Jonathan Hassell、Kevin Shea 和 Gary Schwartz。

特别感谢 Michael Mayberry 审校本书。并提出了见解深刻的反馈意见。

目 录

第 1 章 ASP.NET 4.5 数据源	1
1.1 新的数据访问模式	2
1.2 ASP.NET 中的数据源	4
1.2.1 ADO.NET 数据源	4
1.2.2 NoSQL	5
1.2.3 Web 服务	5
1.2.4 WCF 数据服务	5
1.2.5 联合服务	6
1.2.6 Azure 存储	6
1.2.7 HTML 5 本地存储	6
1.3 .NET Framework 4.5 中的 ADO.NET 简介	7
1.3.1 ADO.NET 的新变化	7
1.3.2 Entity Framework 5.0	7
1.3.3 SqlClient 数据访问接口	7
1.3.4 ADO.NET 和 LINQ	8
1.3.5 数据访问接口	8
1.3.6 异步数据访问	9
1.3.7 .NET Framework 4.5 中的 异步编程模型	10
1.4 开发第一个示例	12
1.4.1 试一试: 使用 ASP.NET 4.5 创建 第一个数据驱动的面页	12
1.4.2 示例说明	20
1.5 小结	20
第 2 章 关系数据库的未来发展	21
2.1 关系数据库的演化	22
2.1.1 关系数据库的缺点	22
2.1.2 Microsoft SQL Server 如何摆脱限制	22
2.2 Microsoft SQL Server 2012 简介	23

2.3 空间数据	25
2.4 数据库引擎和 T-SQL 增强	29
2.4.1 支持分页	29
2.4.2 序列	30
2.4.3 在 Exec 命令中使用 WITH RESULT SET	31
2.4.4 FileTable	32
2.5 数据可视化	32
2.5.1 Power View 和 Power Pivot	33
2.5.2 自助警报	33
2.6 SQL Server 2012 云	34
2.6.1 私有云	35
2.6.2 SQL Azure——公有云	35
2.7 小结	38
第 3 章 非关系数据库简介	39
3.1 理解 NoSQL 数据库	39
3.1.1 概述	40
3.1.2 关键优势	40
3.1.3 流行的 NoSQL 技术	40
3.1.4 面向文档的数据库	41
3.1.5 使用 MongoDB 进行编程	42
3.1.6 MongoDB 的设置步骤	42
3.1.7 使用 MongoDB 客户端	43
3.1.8 使用 C# 进行 MongoDB 编程	44
3.1.9 MongoDB 的工作原理	47
3.2 小结	48
第 4 章 使用 ADO.NET 访问数据	49
4.1 ADO.NET 架构	49
4.1.1 使用 ADO.NET 数据访问接口 读取和修改数据库	50
4.1.2 DataSet 与 DataReader	50

4.1.3 连接数据库.....	51	6.1.2 存储模式.....	84
4.1.4 CRUD 操作的命令.....	53	6.1.3 映射模式.....	85
4.1.5 使用存储过程操作数据.....	53	6.1.4 EDM 生成工具.....	86
4.1.6 异步数据流.....	55	6.2 ADO.NET 元数据.....	87
4.1.7 访问接口工厂.....	55	6.2.1 EDM 类型和关系.....	88
4.1.8 多重活动结果集.....	57	6.2.2 DataSpace 枚举器.....	89
4.2 DataSet、DataAdapter 和 DataSource 控件.....	58	6.2.3 元数据工作区.....	89
4.3 数据类型映射.....	60	6.2.4 规范和存储函数.....	92
4.4 安全的数据访问.....	61	6.3 实体 SQL 简介.....	92
4.4.1 访问 SQL 数据库的权限.....	61	6.4 小结.....	93
4.4.2 加密 Web 配置.....	62		
4.4.3 防止注入攻击.....	63	第7章 使用 Entity Framework 设计数据 访问应用程序.....	95
4.4.4 异常处理.....	64	7.1 Entity Framework 架构.....	95
4.4.5 在内存中处理敏感信息的 安全字符串.....	64	7.2 生成实体模型.....	96
4.4.6 Microsoft SQL Server 2012 扩展保护.....	64	7.2.1 实体关联.....	99
4.5 小结.....	64	7.2.2 使用 LINQ to Entities 查询 EDM.....	100
第5章 LINQ 简介.....	65	7.3 Entity Framework 中使用 Text Template 的代码生成.....	101
5.1 使用 LINQ 编写查询.....	65	7.3.1 DbContext 生成器和 POCO 类.....	103
5.1.1 LINQ 数据源.....	66	7.3.2 POCO 代理.....	105
5.1.2 使用 LINQ 构建查询.....	69	7.4 预先加载实体.....	106
5.2 查询运算符.....	70	7.5 复杂类型.....	106
5.2.1 查询运算符的分类.....	70	7.6 枚举类型.....	110
5.2.2 查询运算符的执行.....	70	7.7 实体客户端数据访问接口.....	110
5.3 将 LINQ 用于 ADO.NET 数据访问.....	71	7.8 对数据访问层进行 单元测试.....	110
5.3.1 LINQ to SQL.....	71	7.9 小结.....	112
5.3.2 延迟加载.....	74		
5.3.3 LINQ to DataSet.....	75	第8章 Entity Framework 的数据 访问模式.....	113
5.3.4 LINQ to Entities.....	76	8.1 模型优先的方法.....	113
5.4 LINQ 和并行性.....	76	8.2 代码优先的方法.....	120
5.5 查询 WCF 数据服务.....	77	8.2.1 数据库初始值.....	122
5.6 小结.....	80	8.2.2 使用 Fluent API 的 自定义映射.....	124
第6章 ADO.NET 实体数据模型.....	81	8.2.3 代码优先迁移.....	125
6.1 EDM 架构.....	81	8.2.4 代码优先方法的逆向工程.....	125
6.1.1 概念模式.....	82		

8.3 使用数据注释进行验证	126	第 11 章 使用动态数据构建可扩展的 数据驱动 Web 应用程序	165
8.4 小结	127	11.1 动态数据层	165
第 9 章 使用 WCF 数据服务	129	11.2 动态数据基架	166
9.1 REST 语义概述	130	11.2.1 自定义 Scaffolding Template	170
9.2 WCF 数据服务简介	130	11.2.2 模板行为	172
9.3 将数据公开为服务	132	11.3 动态数据的工作原理	173
9.3.1 WCF 数据服务访问接口	136	11.3.1 MetaModel	173
9.3.2 将 ASP.NET MVC WebAPI 服务公开为 OData 源	137	11.3.2 路由	174
9.4 在客户端应用程序中使用 数据服务	140	11.3.3 已有控件中的动态 数据	174
9.4.1 通过 jQuery Ajax 使用 OData 源	140	11.4 小结	175
9.4.2 通过 WCF 数据服务客户端库 使用 OData 源	141	第 12 章 构建数据驱动网站的最佳 实践和技术	177
9.5 小结	143	12.1 数据访问层模式	177
第 10 章 ASP.NET 4.5 中的数据 绑定	145	12.1.1 隔离数据层和实体 模型	177
10.1 数据源控件	145	12.1.2 存储库模式	179
10.1.1 LinqDataSource	146	12.1.3 工作单元	183
10.1.2 EntityDataSource	148	12.2 处理问题的技术	188
10.2 数据绑定控件	153	12.2.1 并发性	188
10.2.1 DataList	153	12.2.2 事务	189
10.2.2 DetailsView	154	12.2.3 调试问题	189
10.2.3 QueryExtender	155	12.3 小结	192
10.3 强类型化数据控件	156	第 13 章 应用程序生命周期管理	193
10.3.1 ItemType 特性	156	13.1 为企业创建架构	194
10.3.2 HTML 编码的 表达式	157	13.1.1 设计新应用程序	194
10.4 模型绑定	157	13.1.2 分析已有的应用程序 代码库	195
10.5 ASP.NET MVC 4 中的 数据绑定	159	13.2 提高工作效率	196
10.5.1 Razor 视图引擎	159	13.2.1 更好地访问文件和 代码	196
10.5.2 非介入式 JavaScript	161	13.2.2 访问数据	197
10.5.3 使用 Ajax 进行异步 显示	163	13.2.3 暂停和恢复	199
10.6 小结	164	13.2.4 代码评审	199

13.2.5	分析和配置应用程序.....	200	13.4.2	团队生成定义 触发器.....	203
13.3	测试和调试 ASP.NET 应用程序.....	201	13.4.3	团队生成过程.....	203
13.3.1	单元测试.....	201	13.4.4	使用团队生成进行 部署.....	204
13.3.2	使用 IntelliTrace 进行 历史调试.....	201	13.4.5	部署 SQL Server 数据库.....	205
13.4	生成和部署.....	202	13.5	小结.....	206
13.4.1	团队生成的定义.....	202			

ASP.NET 4.5 数据源

过去十年间，计算设备影响我们生活的方式发生了非同寻常的变化。它们的渗透程度和环境适应程度远超出我们的想象。World Wide Web 已经从显示一些链接页面的集合转变为基于上下文和用户动作显示动态行为。如今的网站能够显示响应性设计，基于目标设备的分辨率修改布局，基于用户位置显示内容，不需要用户下载安装任何媒体特定的插件就能显示内容等，不一而足。在未来十年，推动创新的力量将会是令人着迷的、流畅的用户体验。

由于可以使用 Visual Studio 作为首选的快速应用程序开发工具来快速开发动态网站，因此可以说 ASP.NET 是技术上的一大跃进。我们十分确信，你仍然喜欢用拖放方式将现成控件添加到 Web 应用程序的设计界面上，然后通过几次单击把它们绑定到数据源。自问世以来，ASP.NET 已经发生了显著的变化，ADO.NET 也相应地做了多次改进。目前，ADO.NET 支持多种数据访问模式，包括使用 WCF(Windows Communication Foundation) Data Services 来进行 REST(Representation State Transfer)风格的创建、读取、更新和删除(Create、Read、Update、Delete, CRUD)操作。

本章将概述现代数据访问模式，这有助于学习数据访问组件的设计方式，以及使用它们来满足应用程序的需求。这里只是从一个比较高的层面上进行概述，并没有介绍具体的实现细节。本书其余部分将使用这里讨论的概念来帮助你理解如何将模式的不同组件组合起来，以生成数据驱动的网站。

注意：

第 8 章将详细讨论数据访问模式。

本章主要内容：

- 新的数据访问模式有哪些？
- ASP.NET 4.5 网站可以使用哪些不同的数据源？
- ADO.NET 增强了哪些地方？
- 如何使用 ASP.NET 4.5 和 ASP.NET MVC 4 创建第一个数据驱动的面页？

如果你不确定是否需要创建数据驱动的网站，那么我们强烈建议你阅读Damien Foggon撰写的 *Beginning ASP.NET 2.0 Databases: From Novice to Professional*(Apress, 2006)。该书是本书内容的前驱，很容易把其中的内容与本书讨论的新概念联系起来。

1.1 新的数据访问模式

在开发流程中，如何确定使用哪种方法来构建数据访问层？在设计过程的一个较高层次上，做出这种决定还是很容易的，要考虑的场景只有以下两种：

数据库已经存在：此时，可以选择由已有数据库生成实体模型，然后使用实体模型来创建数据访问层。

注意：

实体模型是一个实体关系图，通过借助概念模型生成器从数据库模式生成。Microsoft .NET Framework 4.5 在 ADO.NET Entity Framework 中内置了一个模型生成器。

数据库是新建的：此时，也有两个选项。可首先创建实体模型，然后根据它来创建数据库。另一种方法是首先创建数据库，然后如上所示，自动生成实体模型。

提示：

从使用实体模型开始是一个好主意，这样就有机会在最终确定概念模型前自由调整模型，而不需要每次都调整数据库模式，让团队中的 DBA 感到厌烦。

在应用程序开发过程中采用以设计为中心的方法时，会出现前面的场景。另外还有一种以代码为中心的方法，许多开发人员更喜欢这种方法，因为操作代码要比在设计界面上修改模型更容易。另一种支持采用以代码为中心的方法的观点是这种方法更易于部署。但是使用这种方法要小心，否则可能需要编写大量代码。

把以代码为中心的方法用于新数据库要比用于已有数据库更简单。有一些工具能够生成已有数据库的代码，但生成的代码很容易出错，而且代码的结构也可能无法满足你的需求。

注意：

以代码为中心的方法提供了更大的灵活性和控制权，所以它在开发社区中越来越流行。开发人员感觉自己有了更大的权力，而且当数据库(或其子集)不只存储在数据库服务器，还存储在手持设备上的独立存储器中时，这种方法也可简化多目标部署。

下面总结了当代应用程序开发中采用的三种数据访问模式：

数据库优先：从数据库模式逆向工程得出实体模型。使用工具从实体模型自动生成实体类。图 1-1 演示了这种数据访问模式。

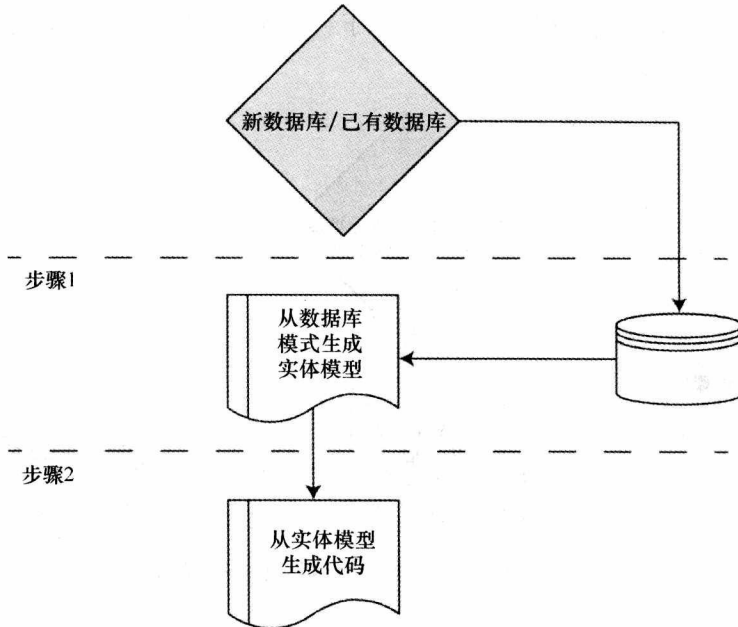


图 1-1 数据库优先数据访问模式的开发流程

模型优先：在设计器中创建实体模型，然后从实体模型生成数据库模式和实体类。图 1-2 演示了这种数据访问模式。

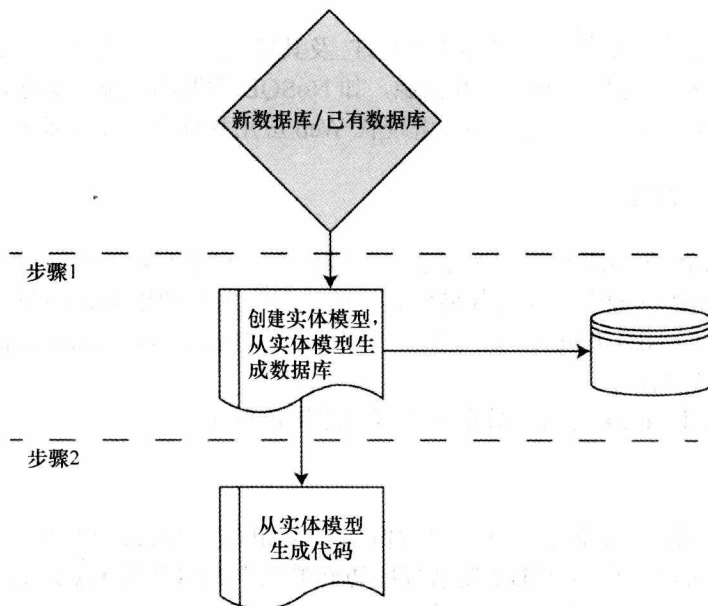


图 1-2 模型优先数据访问模式的开发流程

注意：

对于模型优先的方法，应用于已有数据库要比应用于新数据库更加复杂。图 1-2 的图示忽略了一些额外的细节以保持简单。第 8 章将进一步讨论相关内容。

代码优先：在代码中定义模型和关系。对于新数据库，它将在运行时自动创建。图 1-3 演示了这种数据访问模式。

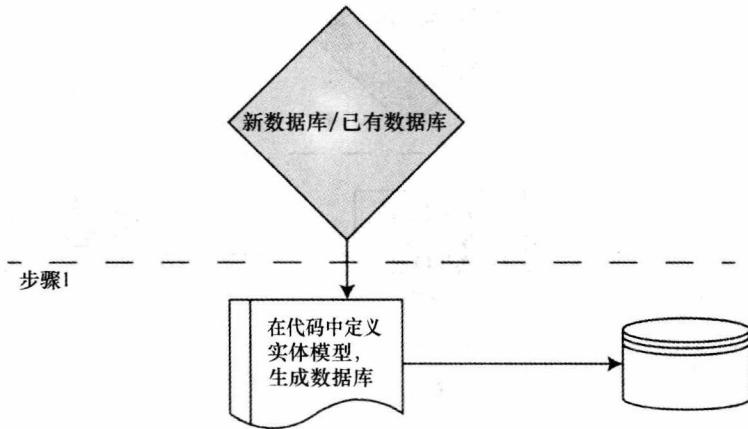


图 1-3 代码优先数据访问模式的开发流程

1.2 ASP.NET 中的数据源

ASP.NET 中对多种数据源的支持都有了显著改进，不仅支持标准的关系数据库，也支持服务和源(feed)。

在很长时间内，关系数据库是 ADO.NET 及其前代 ADO 使用的主要数据源。虽然 ADO.NET 仍占据统治地位，但是新数据源，如 NoSQL 数据库、源和服务，正在逐渐被越来越多的人所接受。接下来，我们将了解现代 Web 应用程序开发中最流行的一些数据源。

1.2.1 ADO.NET 数据源

ADO.NET 依然是 .NET 世界的主宰者。一般认为，它之所以能够如此流行，是因为它可以把数据访问和数据操作干净地分离到离散的、可以独立使用的组件中。

ADO.NET 类是 .NET Framework 4.5 的一部分，包含在 System.Data 名称空间中。它们完成以下三种不同的任务：

- 使用 .NET Framework 4.5 数据访问接口连接数据库
- 执行命令
- 检索结果

ADO.NET 可用的数据源相当多，包含像 Microsoft SQL Server 和 Oracle 这样的关系数据库、可以使用 ODBC 和 OLE DB 驱动程序访问的数据源以及 Entity Data Model(EDM)这样的概念模型。EDM 是新近引入的，第 6 章将详细讨论。

从 ADO.NET 数据源检索的结果可被直接处理，或者存储到 DataSet 中以便进一步处理。ADO.NET DataSet 很受欢迎，因为它提供了从多个数据源获取数据和在显示数据前聚合它们等数据处理功能。

1.2.2 NoSQL

开发社区作为一个整体(Microsoft 等),正在致力于构建解决方案,使非定制的通用软件能够满足企业的实际需求。Ted Neward 在他的文章 *The Working Programmer—Going NoSQL with MongoDB*(<http://msdn.microsoft.com/en-us/magazine/ee310029.aspx>)中提到的“**NoSQL movement**”(NoSQL 运动)在某种程度上可以说是一种革命。这篇文章对所有数据类型都适合放到关系数据库中这一观点提出质疑,并推广面向文档的数据库这一概念,用这种数据库来存储非结构化、但需要频繁访问的数据。博客是使用 NoSQL 数据库作为数据存储的一个很好的例子。网站经常更新没有关联的博客文章,虽然在更新文章和提交评论时需要进行并发控制,但并没有对原子事务操作的需求。在内容管理系统中, NoSQL 数据库也比传统的关系数据库管理系统(Relational Database Management System, RDBMS)更加适合。

现在有很多种 NoSQL 数据库,其中最流行的是 MongoDB 和 CouchDB。虽然它们很相似,但侧重点不同。MongoDB 更关注高性能,而 CouchDB 更关注高并发性。

注意:

.NET 没有直接提供访问 NoSQL 数据库(如 MongoDB)的方式,但是,.NET 开发社区针对它们开发出了 .NET Framework 驱动程序。

1.2.3 Web 服务

Web 服务允许通过 HTTP 访问存储在远程服务器上的数据。如今,“Web 服务”这个术语基本上就是指使用 SOAP(Simple Object Access Protocol)规范通过 HTTP 交换数据。Web 服务也有一段演化史。经历了多次标准修订后,现在 Web 服务是所有现代应用程序架构的组成部分。事实上,新近出现了一种面向服务架构(Service-Oriented Architecture, SOA),这是完整的一层架构,根据多年实现服务的模式构建而成。在 .NET 中,服务已经从使用 WS*规范的 web service enhancements(WSE)转变为一个完善的框架功能:WCF,它最早在 .NET Framework 3.0 中引入。

ASP.NET 4.5 Web 应用程序可被配置为使用代理消费 Web 服务,或者也可以使用 WCF Rich Internet Application(RIA) Services 域服务访问数据。WCF RIA Services 域服务会自动为 CRUD 操作生成对应的方法。

1.2.4 WCF 数据服务

WCF 数据服务允许使用开放数据协议(Open Data Protocol, OData)对数据执行 CRUD 操作。Microsoft 已经将 OData 深入融合到其数据访问策略中。现在,Microsoft 的所有新一代的平台都支持 OData,包括 SQL Server 2008 R2、SQL Server 2012、Windows Azure 存储、SharePoint 2013 和 PowerPivot。

数据服务允许以 REST 风格访问数据模型,且具有灵活查询和关联遍历能力,因而能够与 ASP.NET 这样的平台很自然地集成。数据服务允许以流行的格式表示数据,如 JSON 和 ATOM。