

<<ADO.NET数据访问技术>>

图书基本信息

书名：<<ADO.NET数据访问技术>>

13位ISBN编号：9787302275084

10位ISBN编号：7302275084

出版时间：2012-4

出版时间：清华大学出版社

作者：龚根华，王炜立 主编

页数：269

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<ADO.NET数据访问技术>>

内容概要

本书是一本专门介绍ADO.NET基本功能和新特性的技术参考书。全书共分10章，主要介绍了ADO.NET的数据连接、数据集、事务、LINQ基础以及ADO.NET与XML的交互等一系列重要问题。全书结构严密、层次清晰，以浅显易懂的语言进行描述，并结合大量实例向读者阐述了较为复杂的数据访问问题，帮助读者理解掌握。

本书的读者对象是对Visual Studio.NET和Visual Basic有一定了解的中级用户。本书可作为大学本科计算机专业软件方向的教材以及高等职业院校的实训教材，还可以作为广大用户自学的参考书。

<<ADO.NET数据访问技术>>

书籍目录

第1章 关系数据库基础

- 1.1 表、行、列、约束和表间关系
- 1.2 在Microsoft SQL Server 2008中创建表
- 1.3 视图和索引
 - 1.3.1 视图
 - 1.3.2 索引
- 1.4 SQL语言
 - 1.4.1 SQL概述
 - 1.4.2 SQL语言的组成
- 1.5 存储过程
 - 1.5.1 存储过程的概念
 - 1.5.2 存储过程的优点
 - 1.5.3 SQL Server 2008中存储过程的使用
- 1.6 小结
- 1.7 习题

第2章 ADO.NET概述

- 2.1 ADO.NET简介
- 2.2 ADO.NET数据提供程序
- 2.3 ADO.NET对象模型
- 2.4 ADO.NET命名空间
- 2.5 ADO.NET与XML
- 2.6 小结
- 2.7 习题

第3章 ADO.NET 演示程序

- 3.1 构建演示程序的数据源
- 3.2 创建数据驱动的应用程序：拖放的方法
 - 3.2.1 ASP.NET 3.5 中的拖放
 - 3.2.2 Windows窗口应用程序中的拖放
- 3.3 数据驱动的应用程序：自己编写代码的方法
 - 3.3.1 ASP.NET 3.5 中的代码编写
 - 3.3.2 Windows窗口应用程序中的代码编写
- 3.4 小结
- 3.5 习题

第4章 数据源连接和连接模式数据操作

- 4.1 数据源连接
 - 4.1.1 创建连接对象
 - 4.1.2 生成提供程序特定的连接串
 - 4.1.3 编写连接串的简便方法
 - 4.1.4 增强连接串的安全性
- 4.2 IDbConnection接口
- 4.3 DbConnection类
- 4.4 连接池
 - 4.4.1 工作原理
 - 4.4.2 使用连接字符串关键字控制连接池
- 4.5 关闭连接

<<ADO.NET数据访问技术>>

4.6 Command 对象

4.6.1 创建SqlCommand

4.6.2 SqlCommand属性

4.6.3 SqlCommand常用方法

4.7 小结

4.8 习题

第5章 数据集DataSet

5.1 非连接模型的案例

5.2 DataSet对象模型

5.2.1 DataTable

5.2.2 DataColumn

5.2.3 DataRow

5.2.4 设置主键：PrimaryKey属性

5.2.5 Constraint

5.2.6 动态构建DataTable

5.2.7 DataTable的事件

5.2.8 DataTable事件的实际用法

5.3 关系数据

5.4 综合运用

5.5 强类型DataSet

5.5.1 DataSet架构

5.5.2 构建强类型DataSet

5.5.3 类型化DataSet的性能

5.6 小结

5.7 习题

第6章 非连接模式下获取数据

6.1 什么是DataAdapter

6.2 使用DataAdapter

6.2.1 创建SqlDataAdapter

6.2.2 查询数据表

6.2.3 填充DataSet中的多个数据表

6.2.4 查询数据库架构

6.3 映射

6.3.1 表映射和字段映射

6.3.2 ADO.NET的数据类型映射

6.4 小结

6.5 习题

第7章 更新数据

7.1 更新数据表：简单拖放方法

7.2 使用命令构建器对象

7.3 DataRow中的状态管理以及在更新数据时状态管理的使用

7.4 移动大量的数据

7.5 编辑非连接数据

7.5.1 添加新行

7.5.2 修改现存行记录

7.5.3 删除现存行记录

7.6 GetChanges和Merge

<<ADO.NET数据访问技术>>

- 7.6.1 使用GetChanges()方法的时机
- 7.6.2 在DataSet中获得更改的示例
- 7.6.3 将更改合并到DataSet
- 7.7 使用映射名称更新记录
- 7.8 小结
- 7.9 习题
- 第8章 事务
- 8.1 什么是事务
- 8.1.1 ACID属性
- 8.1.2 数据库事务
- 8.1.3 事务词表
- 8.2 ADO.NET的事务支持
- 8.3 编写事务性数据库应用程序
- 8.3.1 实现事务
- 8.3.2 隔离级别
- 8.4 单数据库的高级技术
- 8.4.1 保存点
- 8.4.2 嵌套事务
- 8.4.3 使用DataSet和DataAdapter的事务处理
- 8.5 分布式事务
- 8.5.1 分布式事务中的关键技术
- 8.5.2 两阶段提交
- 8.5.3 实现分布式事务
- 8.5.4 可提升登记：简单基础
- 8.5.5 System.Transactions：手动登记和多线程环境
- 8.6 明智地使用事务
- 8.6.1 事务和性能
- 8.6.2 事务的默认行为
- 8.6.3 事务和用户确认
- 8.6.4 同时发生的ADO.NET和RDBMS事务
- 8.7 小结
- 8.8 习题
- 第9章 LINQ技术基础
- 9.1 LINQ简介
- 9.1.1 LINQ的基本概念
- 9.1.2 一个简单实例
- 9.2 LINQ查询运算符
- 9.2.1 建立测试项目
- 9.2.2 from子句
- 9.2.3 where子句
- 9.2.4 select子句
- 9.2.5 join子句
- 9.2.6 orderby子句
- 9.2.7 group子句
- 9.3 LINQ to DataSet
- 9.3.1 DataSet操作简介
- 9.3.2 使用LINQ to DataSet查询数据

<<ADO.NET数据访问技术>>

- 9.3.3 DataSet扩展的作用
- 9.3.4 LINQ查询的其他用法
- 9.4 LINQ to SQL
 - 9.4.1 一个LINQ to SQL的简单实例
 - 9.4.2 强类型的DataContext
 - 9.4.3 自动生成实体类
 - 9.4.4 使用自动生成的实体类编程
- 9.5 LINQ to XML
 - 9.5.1 LINQ to XML框架
 - 9.5.2 XElement类
 - 9.5.3 XAttribute类
 - 9.5.4 XDocument类
 - 9.5.5 加载XML文档
 - 9.5.6 遍历XML文档
 - 9.5.7 操纵XML文档
- 9.6 小结
- 9.7 习题
- 第10章 ADO.NET和XML
 - 10.1 XML和DataSet的交互
 - 10.1.1 将DataSet保存为XML
 - 10.1.2 从XML加载DataSet
 - 10.1.3 数据集与XmlDataDocument同步
 - 10.2 XML文档的处理
 - 10.3 顺序读取XML数据
 - 10.3.1 创建对象
 - 10.3.2 XmlReader的常用属性和方法
 - 10.3.3 XmlReader读取XML实例
 - 10.4 顺序写入XML数据
 - 10.4.1 创建对象
 - 10.4.2 XmlWriter的常用属性和方法
 - 10.4.3 利用XmlWriter写元素
 - 10.4.4 利用XmlWriter写属性
 - 10.4.5 XmlWriter处理命名空间
 - 10.4.6 XmlWriter生成XML实例
 - 10.5 使用DOM处理XML数据
 - 10.5.1 DOM
 - 10.5.2 DOM节点和相关的.NET节点类型
 - 10.5.3 将XML文档读入DOM
 - 10.5.4 利用XmlDocument读取XML数据
 - 10.5.5 利用XmlDocument添加XML数据
 - 10.5.6 利用XmlDocument更新XML数据
 - 10.6 小结
 - 10.7 习题

章节摘录

版权页：插图：当今社会是一个信息化的社会，信息已经成为社会上各行各业的重要资源。

数据是信息的载体，数据库是互相关联的数据集合。

数据库能利用计算机保存和管理大量复杂的数据，快速而有效地为多个不同的用户和应用程序提供数据，帮助人们有效利用数据资源。

严格地讲，数据库是长期存储在计算机内、有组织的、可共享的大量数据的集合。

数据库中的数据按一定的数据模型组织、描述和存储，具有较小的冗余度、较高的数据独立性和易扩展性，并可为各种用户共享。

关系模型（Relational Model）是目前最重要的一种数据模型。

关系模型用二维表（即集合论中的关系）来表示实体和实体之间的联系，它是经典数据模型中建模能力最强的一种，对于各种类型数据联系都可以描述。

关系模式以关系理论为基础，有严密的数学理论支持，是当今实用数据库系统的主流数据模型。

且该模型对数据的检索和更新结果也是关系，所以其数据结构简单、清晰，用户易懂易用。

关系数据库系统采用关系模型作为数据的组织方式。

本章概要介绍关系数据库的基本知识和结构化查询语言（Structured Query Language, SQL），并在Microsoft SQL Server 2008环境中创建一个数据库实例，能让读者更深刻地体会如何在一个数据库中组织数据。

1.1 表、行、列、约束和表间关系 从用户观点来看，关系模型中数据的逻辑结构就是一张二维表，每张表都有一个唯一的名字；从计算机存储表示来看，一张表就是一个数据文件。

表由行和列组成，表中的一行代表的是一系列值之间的联系，一张表就是这种联系的一个集合。

表和数学上的关系是密切相关的，这正是关系数据库名称的由来。

表1—1所示的是一个快餐店网上预订配送系统中的客户表（Customer），该客户表中保存了快餐店注册客户的信息，包含有客户ID、注册名、密码、客户姓名、客户地址、客户电话、电子邮件和备注等。

下面以表1—1为例介绍关系数据库的基本概念。

<<ADO.NET数据访问技术>>

编辑推荐

《21世纪高等学校计算机专业实用规划教材:ADO.NET数据访问技术》使用的开发环境是Visual Studio 2010+SQL Server 2008,《21世纪高等学校计算机专业实用规划教材:ADO.NET数据访问技术》各知识点的叙述由浅入深,并辅以大量的实例对各知识点进行说明,在大部分的实例中,数据都来源于一个完整的数据库案例。

《21世纪高等学校计算机专业实用规划教材:ADO.NET数据访问技术》在叙述上简洁明了,内容涵盖了ADO.NET技术的大部分知识点。

《21世纪高等学校计算机专业实用规划教材:ADO.NET数据访问技术》适合于有一定编程基础、想要学习和扩展数据库开发技术的读者,也可作为大学以及高职、高专等学校的相关专业或培训教材和程序设计编程人员的参考书。

<<ADO.NET数据访问技术>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>