

C#数据库程序设计

林成春 孟湘来 马朝东 编著

.NET SOFTWARE ENGINEER

定位明确，强调程序设计技能的重要性
内容完备，实例丰富，紧密联系实战
强调程序外的项目开发方法和软件工程概念



清华大学出版社

● 北京交通大学出版社



.NET 软件工程师就业培训系列教程

C#数据库程序设计

林成春 孟湘来 马朝东 编著

清华大学出版社

北京交通大学出版社

• 北京 •

内 容 简 介

基于数据库的网络应用程序开发是当今程序开发的一大特色,本书基于 SQL Server 2005 服务器技术和 .NET 数据库操作技术,全面、详细地介绍了在该框架下的 ADO.NET 数据库访问技术,以及应用 C# 编程语言实现数据库的程序开发。

全书共分为 20 章,主要内容包括:C#数据库设计的基础知识;网络数据库 SQL Server 2005 的相关知识;ADO.NET 数据连接知识,ADO.NET 数据库操作对象 Command,ADO.NET 数据读取对象;以及内存中的数据库 DataSet,离线数据操作与 DataAdapter,事务及其处理机制,BLOB 大型对象的操作原则;同时还介绍了基于在线方式的通讯录设计、基于离线方式的通讯录设计、系统 DAL 组件设计和照片管理器设计等四个典型应用系统的实现过程。

本书主要面向对 C#数据库程序设计感兴趣的读者,按照由基础到高级,并结合具体应用系统的程序设计过程,介绍 C#数据库程序设计技术,使读者通过本书的学习,能够更好地理解和掌握 C#语言和数据库操作的相关技术,能够以专业的方式进行实际数据库应用程序的设计,实现基于网络数据库的应用。

本书可作为高等院校计算机科学与技术、信息管理、电子商务等专业的教学用书,也可以作为 C#数据库程序设计的培训教材。

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 13501256678 13801310933

图书在版编目(CIP)数据

C#数据库程序设计教程 / 林成春编著. —北京:清华大学出版社;北京交通大学出版社, 2010.7

ISBN 978-7-81123-900-3

I. ①C… II. ①林… III. ①C 语言-程序设计-教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2009)第 202997 号

责任编辑:谭文芳

出版发行:清华大学出版社 邮编:100084 电话:010-62776969 <http://www.tup.com.cn>

北京交通大学出版社 邮编:100044 电话:010-51686414 <http://press.bjtu.edu.cn>

印刷者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185×260 印张:29.75 字数:762 千字

版 次:2010 年 7 月第 1 版 2010 年 7 月第 1 次印刷

书 号:ISBN 978-7-81123-900-3/TP·543

印 数:1~4 000 册 定价:46.00 元

本书如有质量问题,请向北京交通大学出版社质监组反映。对您的意见和批评,我们表示欢迎和感谢。

投诉电话:010-51686043, 51686008; 传真:010-62225406; E-mail: press@bjtu.edu.cn。

前 言

基于数据库的网络应用程序开发是当今程序开发的一大特色，本书基于 SQL Server 2005 服务器技术和 .NET 数据库操作技术，全面、详细地介绍了在该框架下的 ADO.NET 数据库访问技术，以及应用 C# 程序设计语言实现数据库的程序开发。

全书共分为 20 章。其中，第 1 章介绍 C# 数据库设计技术概述，包括 .NET 数据库操作技术概述、ADO .NET 数据库操作模型；第 2 章介绍 SQL Server 2005 的特点、系统数据库和表、SQL Server 2005 提供的配置管理器和企业管理控制台的基本功能，以及 SQL Server 2005 的安装；第 3 章 Transact-SQL 语言，重点介绍 SQL Server 2005 的数据类型，并通过例题说明在 SQL Server 2005 查询分析器中如何应用 Transact-SQL 语言进行程序设计；第 4~7 章列举大量例题介绍如何应用 SQL Server 2005 的管理控制台（企业管理器），以及在查询分析器中使用 Transact-SQL 语言创建、管理和维护数据库、数据库表、索引、视图、存储过程、触发器、关系图等数据库和数据库对象；第 8 章介绍 SQL Server 2005 的并发控制机制；第 9 章介绍 ADO.NET 数据连接的相关知识；第 10 章介绍 ADO.NET 数据库操作对象 Command；第 11 章介绍 ADO.NET 数据读取对象 DataReader；第 12 章介绍内存中的数据库 DataSet；第 13 章介绍离线数据操作与 DataAdapter；第 14 章介绍 ADO.NET 的数据连接工厂；第 15 章介绍事务及其处理机制；第 16 章介绍有关数据库大型对象 BLOB 的操作；第 17~20 章介绍基于在线方式的通讯录设计、基于离线方式的通讯录设计、系统 DAL 组件设计和照片管理器设计四个典型应用系统的设计、实现过程。

本书在编写过程中力求文字简练、图表丰富、操作过程清晰，特别是通过大量例题的分析和程序设计能有益于读者理解和掌握所介绍的 C# 数据库程序设计的基本功能、基本技术和基本操作过程。

本书主要面向对 C# 数据库程序设计感兴趣的读者，按照由基础到高级，并结合具体应用系统的程序设计过程，介绍 C# 数据库程序设计技术，使读者通过本书的学习，能够更好地理解和掌握 C# 语言和数据库操作的相关技术，能够以专业的方式进行实际数据库应用程序的设计，实现基于网络数据库的应用。

本书第 1 章、第 9~12 章由马朝东副教授编写；第 2~8 章由林成春教授编写；第 13~18 章由孟湘来副教授编写；第 19 章和第 20 章由熊艺副教授编写；参加本书编写的还有高秀兰、林慧、李娟、林青、李军、张丽、高剑峰等；孟湘来副教授修改了全部书稿，林成春教授审定了全书。

在本书编写过程中天津工业大学李兰友教授给予了具体指导，北京交通大学出版社谭文芳编辑给予了大力支持，在此表示衷心地感谢。

由于时间仓促和编者水平有限，书中疏漏、不妥之处在所难免，敬请读者和同行给予批评、指正。

编 者

2010 年 2 月

目 录

第 1 章 C#数据库设计技术概述	1
1.1 .NET 数据库操作技术概述	1
1.1.1 ADO 技术概述	1
1.1.2 ADO.NET 技术概述	2
1.1.3 .NET ORM 技术映射	2
1.1.4 数据库设计框架概述	3
1.2 ADO.NET 数据操作模型	4
1.2.1 在线方式数据库操作	5
1.2.2 离线方式数据库操作	6
1.3 数据库程序的分类	6
1.3.1 SQL 脚本程序	6
1.3.2 Transact-SQL 程序设计	7
1.3.3 数据库组件设计	7
本章小结	8
习题 1	8
第 2 章 SQL Server 2005 关系数据库管理系统	9
2.1 SQL Server 2005 简介	9
2.1.1 SQL Server 的发展简史	9
2.1.2 SQL Server 2005 的新特性	10
2.2 SQL Server 2005 的版本与安装	12
2.2.1 SQL Server 2005 的版本	12
2.2.2 SQL Server 2005 的运行环境	13
2.2.3 SQL Server 2005 的安装	14
2.3 SQL Server 2005 的系统数据库和表	21
2.3.1 SQL Server 2005 的系统数据库	21
2.3.2 SQL Server 2005 的系统表	22
2.4 SQL Server 2005 的管理工具和实用程序	23
2.4.1 SQL Server 配置管理器	24
2.4.2 SQL Server 管理控制台	28
2.4.3 联机丛书	36
2.5 SQL Server 2005 的安全性管理	37
2.5.1 SQL Server 2005 身份验证模式	37
2.5.2 SQL Server 数据库的安全性	39

2.5.3 SQL Server 数据库对象的安全性	40
本章小结	41
习题 2	41
第 3 章 Transact-SQL 语言	43
3.1 SQL Server 2005 的数据类型	43
3.1.1 数值数据类型	44
3.1.2 字符数据类型	45
3.1.3 日期和时间数据类型	46
3.1.4 文本和图像数据类型	46
3.1.5 货币数据类型	46
3.1.6 二进制数据类型	47
3.1.7 特殊数据类型	47
3.1.8 用户自定义数据类型	48
3.1.9 SQL Server 2005 数据类型的不同分类	48
3.2 数据定义语言 (DDL)	48
3.3 数据操纵语言 (DML)	50
3.4 数据控制语言 (DCL)	54
3.4.1 权限类型和状态	54
3.4.2 GRANT 语句	55
3.4.3 DENY 语句	57
3.4.4 REVOKE 语句	57
3.5 系统存储过程	58
3.6 其他语言元素	58
3.6.1 注释	58
3.6.2 变量	59
3.6.3 运算符	60
3.6.4 函数	61
3.6.5 流程控制语句	66
3.6.6 批处理	73
本章小结	74
习题 3	75
第 4 章 数据库的创建和维护	77
4.1 数据库的存储结构	77
4.1.1 数据库文件	78
4.1.2 数据库文件组	78
4.2 创建、修改和删除数据库	79
4.2.1 创建数据库	79
4.2.2 查看数据库	84
4.2.3 修改数据库	85

4.2.4 删除数据库	87
4.3 备份和还原数据库	87
4.3.1 概述	87
4.3.2 备份数据库的操作	88
4.3.3 还原数据库	92
本章小结	95
习题 4	95
第 5 章 数据库表的创建和维护	97
5.1 定义表结构	97
5.1.1 创建表的设计原则	98
5.1.2 如何定义表的结构	98
5.2 如何修改表结构	101
5.2.1 如何重新命名表	102
5.2.2 如何修改表字段	102
5.3 如何向表中添加记录	103
5.4 如何删除表中记录	105
5.5 如何修改表中的记录值	106
5.6 如何删除表	107
5.7 如何查看表	108
5.7.1 查看表属性	108
5.7.2 查看表中数据——简单查询	108
5.7.3 使用聚合函数的查询	110
5.7.4 查看表中数据——连接查询	115
5.7.5 查看表中数据——嵌套查询	118
本章小结	122
习题 5	123
第 6 章 数据库表索引和约束的建立	125
6.1 创建和管理索引	125
6.1.1 索引的设计原则和索引的类型	126
6.1.2 创建索引	127
6.1.3 查看、修改和删除索引	131
6.2 数据库数据的完整性	133
6.2.1 什么是数据库数据的完整性	133
6.2.2 如何实现数据库数据的完整性	135
6.2.3 主键约束	136
6.2.4 唯一性约束	138
6.2.5 检查性约束	140
6.2.6 外键约束	143
6.2.7 默认约束	147

本章小结	149
习题 6	150
第 7 章 数据库对象的创建和维护	152
7.1 创建和管理视图	152
7.1.1 视图	152
7.1.2 如何创建视图	153
7.1.3 查看视图	157
7.1.4 修改、删除及重命名视图	159
7.1.5 使用视图操作表数据	162
7.2 创建和管理存储过程	165
7.2.1 存储过程的概念	165
7.2.2 创建存储过程	166
7.2.3 查看、修改和删除存储过程	170
7.3 创建和管理触发器	175
7.3.1 触发器的作用和类型	175
7.3.2 创建触发器	176
7.3.3 查看、修改和删除触发器	180
7.4 创建和管理关系图	184
7.5 游标	187
7.5.1 游标的概念	187
7.5.2 游标的使用	187
本章小结	189
习题 7	190
第 8 章 SQL Server 2005 并发控制机制	192
8.1 并发控制理论	192
8.2 并发调度	192
8.2.1 事务	193
8.2.2 数据不一致性	193
8.2.3 可串行化调度	195
8.3 封锁机制	195
8.3.1 基本锁	195
8.3.2 专用锁	196
8.3.3 封锁协议	197
8.3.4 锁的粒度	198
8.4 事务隔离级别	199
8.4.1 事务隔离级别	199
8.4.2 数据库引擎的隔离级别	200
8.5 死锁的预防和处理	200
8.5.1 死锁的概念	200

8.5.2 预防和死锁	202
本章小结	203
习题 8	204
第 9 章 ADO.NET 数据库连接	205
9.1 基本的数据库连接	206
9.2 在线数据库操作模型	207
9.3 数据库连接的基本成员	207
9.4 常用数据库连接方式	207
9.4.1 SQL Server 数据库连接	207
9.4.2 OLE DB 数据库连接	211
9.4.3 ODBC 数据库连接	213
9.4.4 MySQL 数据库连接	215
9.5 连接中的元数据处理	216
9.6 连接的状态与数据处理	218
9.7 使用连接池优化连接	218
9.8 数据库异常处理	221
本章小结	223
习题 9	223
第 10 章 ADO.NET 数据库操作对象 Command	225
10.1 Command 接口和 SqlCommand 对象	225
10.1.1 Command 对象模型	226
10.1.2 Command 对象的常见属性	226
10.2 通过 Command 对象操作数据库	226
10.2.1 查询性操作	228
10.2.2 更改性操作	231
10.2.3 标量行查询	232
10.2.4 Command 参数对象	233
10.3 Command 对象调用存储过程	236
10.3.1 存储过程的调用方法	236
10.3.2 输入参数的设置方法	237
10.3.3 输出参数的设置方法	237
10.3.4 返回参数的设置方法	237
10.4 通过 Visual Studio 2005 设计 Command 对象	237
本章小结	240
习题 10	240
第 11 章 ADO.NET 数据读取对象	242
11.1 DataReader 的基本概念	242
11.1.1 DataReader 的基本模型	242
11.1.2 使用 DataReader 进行数据查询	245

11.1.3	使用 DataReader 进行多指令数据查询	247
11.1.4	DataReader 显示数据	248
11.2	使用 DataReader 操作元数据	250
11.2.1	通过 DataReader 获取表对象信息	250
11.2.2	通过 DataReader 获取字段对象信息	251
本章小结		251
习题 11		251
第 12 章	内存中的数据库 DataSet	252
12.1	DataSet 的基本模型	252
12.2	DataSet 中的表结构	254
12.2.1	表结构的创建	254
12.2.2	表结构的约束	255
12.2.3	表结构的主键	255
12.2.4	表结构的外键	256
12.3	DataSet 表的操作	257
12.3.1	DataSet 表的数据的插入	257
12.3.2	DataSet 表的数据的删除	259
12.3.3	DataSet 表的数据的复制	261
12.3.4	DataSet 表的数据的查询	261
12.3.5	DataSet 表的数据的枚举	261
12.4	DataSet 中的视图对象	262
12.5	DataSet 的 XML 序列化	265
12.6	DataSet 的 XML 操作	267
12.7	通过 Visual Studio 2005 设计 DataSet	269
本章小结		273
习题 12		273
第 13 章	离线数据操作与 DataAdapter	275
13.1	离线数据操作模型	275
13.1.1	DataAdapter 提取数据	276
13.1.2	DataAdapter 更新数据	280
13.2	DataAdapter 数据操作过程中对连接对象的影响	283
13.3	DataAdapter 的基本结构	284
13.3.1	DataAdapter 的 Select 命令	286
13.3.2	DataAdapter 的 Delete 命令	289
13.3.3	DataAdapter 的 Update 命令	292
13.3.4	DataAdapter 的 Insert 命令	294
13.3.5	DataAdapter 的 TableMapping 机制	297
本章小结		298
习题 13		299

第 14 章 ADO.NET 数据连接工厂	300
14.1 ADO.NET 数据连接工厂的概念模型	300
14.2 创建独立于特定数据库的应用程序	301
14.3 数据库连接驱动模型	305
14.3.1 ODBC 数据提供程序	306
14.3.2 OLE DB 数据提供程序	316
14.3.3 SQL Server 数据提供程序	320
14.3.4 Oracle 数据提供程序	323
14.4 ADO.NET 基本结构模型	323
本章小结	324
习题 14	324
第 15 章 事务	326
15.1 事务的基本概念	326
15.2 本地事务	327
15.2.1 事务的处理方法	327
15.2.2 事务对象的属性	329
15.2.3 本地嵌套事务	329
15.3 全局事务	332
15.3.1 全局事务的基本模型	332
15.3.2 基于 Com+服务的全局事务	333
15.3.3 基于 ADO.NET 自身的全局事务处理机制	338
本章小结	340
习题 15	340
第 16 章 BLOB 对象的操作	342
16.1 数据库大对象的类型	342
16.2 二进制数据的操作	343
16.2.1 图片的数据库存储	343
16.2.2 图片的数据库读取	348
16.3 BLOB 数据对象的操作原则	354
本章小结	356
习题 16	356
第 17 章 基于在线方式的通讯录设计	357
17.1 系统的需求概要和项目背景	357
17.2 系统的开发计划	360
17.3 系统的概要设计	361
17.3.1 功能模块分析	361
17.3.2 系统业务流程	361
17.3.3 数据库设计	362
17.4 系统的代码实现	363

17.4.1 创建项目	363
17.4.2 制作“用户登录”窗体	369
17.4.3 制作“主操作”窗体	372
17.4.4 制作“新增联系人”窗体	381
17.4.5 制作“修改联系人”窗体	386
17.4.6 制作“查询联系人”窗体	391
17.5 系统的部署和发布	394
本章小结	398
习题 17	398
第 18 章 基于离线方式的通讯录设计	399
18.1 系统的需求概要和项目背景	399
18.2 系统的开发计划	400
18.3 系统的概要设计	400
18.3.1 功能模块分析	400
18.3.2 系统业务流程	400
18.3.3 数据库设计	400
18.4 系统的代码实现	400
18.4.1 创建项目	400
18.4.2 设计 DataClass.cs 模块	401
18.4.3 制作“用户登录”窗体	404
18.4.4 制作“主操作”窗体	406
18.4.5 制作“新增联系人”窗体	414
18.4.6 制作“修改联系人”窗体	419
18.4.7 制作“查询联系人”窗体	423
18.5 系统的部署和发布	424
本章小结	424
习题 18	425
第 19 章 系统 DAL 组件设计	426
19.1 系统的需求概要和项目背景	426
19.2 系统的概要设计	427
19.3 系统的开发计划	427
19.4 系统的代码实现	428
19.4.1 创建一个 Web 项目, 配置数据库连接	428
19.4.2 创建一个数据访问层	429
19.4.3 给数据访问层添加参数化的方法	435
19.4.4 插入、更新和删除数据	438
19.4.5 完成数据库访问层	440
19.5 系统的部署和发布	440
本章小结	442

习题 19	442
第 20 章 照片管理器设计	443
20.1 系统的概要设计和项目背景	443
20.2 系统的开发计划	443
20.3 系统的代码实现	443
20.3.1 数据库和表的创建	444
20.3.2 数据访问层 (DAL) 的设计与实现	445
20.3.3 表示层 (USL) 的开发	445
20.3.4 业务逻辑层 (BLL) 的设计与实现	450
20.4 系统的部署和发布	459
本章小结	460
习题 20	460
参考文献	461

第 1 章 C#数据库设计技术概述

本章要点:

- ☑ .NET 数据库操作技术概述
- ☑ ADO.NET 数据操作模型
- ☑ SQL Server 2005 技术概述
- ☑ 数据库程序的分类

1.1 .NET 数据库操作技术概述

随着美国 Microsoft 公司不断推出新的技术和新的系统，基于数据库的操作技术也不断获得发展和更新，目前.NET 数据库操作技术是该领域主流技术。

1.1.1 ADO 技术概述

ADO 是微软的一项技术，是 ActiveX Data Objects 的英文缩写。它是微软的 Active-X 组件，随着 IIS 的安装而自动安装，是用来存取数据库中记录的一种程序。

ADO 可以使 ASP 页面从数据库中存取数据。在 ASP 页面中存取数据库中数据的一般步骤是：创建（打开）与数据库的连接、打开数据库、创建 ADO 记录集、打开记录集、从记录集中存取数据、关闭记录集、关闭与数据库的连接。

ADO 的对象包括以下几种。

- (1) Connection 连接对象：用于表示和数据源的连接，以及处理一些命令和事务。
- (2) Command 命令对象：用于执行某些命令来进行诸如查询、修改数据库结构的操作。
- (3) Recordset 记录集对象：用于处理数据源的表格集，它是在表中修改、检索数据的最主要的方法。
- (4) Field 列对象：描述数据集中的列信息。
- (5) Parameter 参数对象：用于对传递给数据源的命令赋参数值。
- (6) Error 错误对象：用于承载所产生错误的详细信息。
- (7) Property 属性对象：通过属性，每个 ADO 对象借此来让用户描述和控制自身的行为。
- (8) Set 集合对象：是一种可以方便地包含其他特殊类型对象的对象类型。ADO 提供 4 种类型的集合：

- ① Connection 对象具有 Error 集合：

- ② Command 对象具有 Parameter 集合;
- ③ Recordset 对象具有 Fields 集合;
- ④ Connection、Command、Recordset、Field 对象都具有 Property 集合。
- (9) Event: 事件模型是异步操作的基础, 这是 ADO 2.0 引进的新特性。

1.1.2 ADO.NET 技术概述

ADO.NET 是微软.NET 平台中的一种新的数据访问技术。它有着全新的设计理念, 并在原有的 ADO 基础上引入了一些重大的变化和革新, 不管数据源是什么数据库, 都可以通过它进行高效访问, 是应用程序和数据库之间的重要桥梁。

ADO.NET 有效地从数据操作中将数据访问分解为多个可以单独使用或先、后使用的不连续组件。ADO.NET 包含用于连接到数据库、执行命令和检索结果的 .NET Framework 数据提供程序。这些结果或者被直接处理, 放在 ADO.NET DataSet 对象中以便以特别的方式向用户公开, 并与来自多个源的数据组合; 或者在层之间传递。DataSet 对象也可以独立于 .NET Framework 数据提供程序, 用于管理应用程序本地的数据或源自 XML 的数据。

ADO.NET 类在 System.Data.dll 中, 并且与 System.Xml.dll 中的 XML 类集成。

ADO.NET 对象可大体分成两大类: 一类是与数据库直接连接的联机对象 (称为 .NET Data Provider), 其中包含了 Command (命令) 对象、DataReader (数据读取器) 对象及 DataAdapter (数据适配器) 对象等, 通过这些类对象, 可以在应用程序里完成连接数据源及数据维护等相关操作; 另一类则是与数据源无关的离线对象, 如 DataSet (数据集) 对象和 DataRelation 对象等。

DataSet 对象是 ADO.NET 的中心概念, 可以把 DataSet 想象成内存中的数据库, 它就像是离线的数据一样。正是由于 DataSet, 才使得程序员在编写程序时可以屏蔽数据库之间的差异获得一致的编程模型。

ADO.NET 的关键设计准则是简单性和高效性, 它的设计原理在于为分布式应用程序提供一种合适的解决方案, 用户可以利用连接对象取得数据源里所需的原始数据, 并且利用断线对象, 一次返回给前端用户, 前端用户在处理变动数据的过程中, 并不需要保持与数据库连接, 当对所有数据完成变动操作之后, 则再一次通过连接对象将数据返回更新到数据库, 因为不需要时时保持与数据库的连接, 所以能够大大降低所消耗的系统资源。

ADO.NET 在 .NET 中为存取任何类型的数据提供了一个统一的框架, 它适用于 WinForms 应用程序、ASP.NET 应用程序和 Web Servers。

1.1.3 .NET ORM 技术映射

ORM (Object Relational Mapping, 对象关系映射) 的实质就是将关系数据 (库) 中的业务数据用对象的形式表示出来, 并通过面向对象 (Object-Oriented) 的方式将这些对象组织起来, 实现系统业务逻辑的过程。在 ORM 过程中最重要的概念是映射 (Mapping), 通过这种映射可以使业务对象与数据库分离。从面向对象来说, 数据库不应该和业务逻辑绑定到一起, ORM 则起到这样的分离作用, 使数据库层透明, 开发人员真正的面向对象。图 1-1 简单说明了 ORM 在多层系统架构中的这个作用。

目前大多数项目或产品都使用关系型数据库实现业务数据的存储, 这样在开发过程

中，常常有一些业务逻辑需要直接用写 SQL 语句实现，但这样开发的结果是：遍地布满 SQL 语句，这些高耦合的 SQL 语句给系统的改造和升级带来很多无法预计的障碍。为了提高项目的灵活性，特别是快速开发，ORM 是一个不错的选择。举个简单的例子：在使用 ORM 的系统中，当数据库模型改变时，不再需要理会逻辑代码和 SQL 语句中涉及该模型的所有改动，只需要将该模型映射的对象稍作改动，甚至不作改动就可以满足要求。

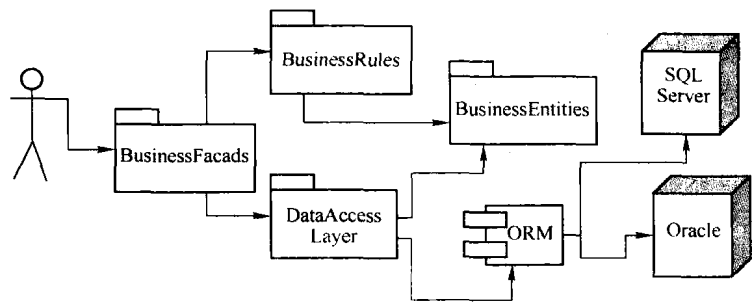


图 1-1 ORM 在多层系统架构中的作用

1.1.4 数据库设计框架概述

随着 .NET Framework 3.5 SP1 和 Visual Studio 2005 SP1 的正式发布，ADO.NET 实体框架正式来到开发人员的面前，它使开发人员可以通过对象模型（而不是逻辑/关系数据模型）专注于数据。实体框架有助于将逻辑数据架构抽象为概念模型，并且允许以多种方式通过对象服务和名为“EntityClient”的新数据提供程序与概念模型交互。

实体框架生成开发人员能够依据其编写代码的概念模型，使用名为“EntityClient”的新数据提供程序和名为“实体 SQL”的新语言（类似于 T-SQL）直接与该模型交互。EntityClient 具有与 ADO.NET 对象类似的模型，使用 EntityConnection 和 EntityCommand 对象返回 DbDataReader。开发人员的另一种方法是通过具有实体 SQL 的 ObjectQuery 对象或 LINQ to Entities 来使用对象服务，对象服务使开发人员可以利用概念模型的生成类，这些生成类提供了强类型化对象和持久性等特性（见图 1-2）。

实体框架使开发人员可以编写更少的数据访问代码，减少维护，将数据结构抽象化为更易于开展业务（标准化程度较低）的方式，并且有利于数据的持久性。当与 LINQ to Entities 结合使用时，由于实体框架从概念模型中生成强类型化类，还有助于降低编译时错误的数量。

实体框架的核心是实体数据模型（EDM）。EDM 定义开发人员通过代码进行交互的实体类型、关系和容器。实体框架将这些元素映射到关系数据库公开的存储架构上。EDM 通

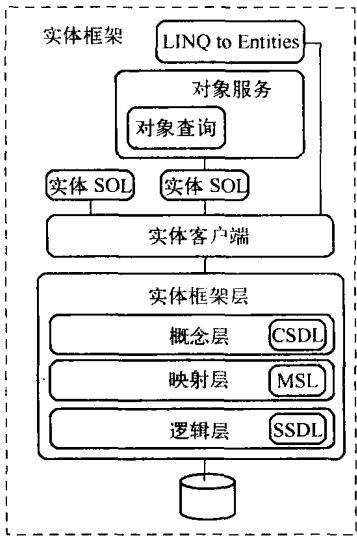


图 1-2 实体框架结构图

过用于定义概念应用程序模型的 XML 向实体框架公开。概念模型可单独定义，也可以与用于定义实际存储架构的 XML 以及用于定义实体框架与关系数据库公开的存储架构两者之间映射的 XML 一起定义。尽管可以（有时也有必要）手动编辑 XML，但使用新的可视化实体数据模型设计器工具来创建和修改实体模型和映射会更加容易。

这些数据访问技术使开发人员可以与 EDM 的概念实体交互，EDM 各层以 XML 文件形式存在；通过这些数据访问技术，可以与实体模型中定义的概念实体进行交互，而不是与物理存储区（如关系数据库）的对象进行交互。

实体框架支持表示数据库中的关系架构的逻辑存储模型，如图 1-2 所示。模型中有三个处于活动状态的层：概念层、映射层和逻辑层。

这三层允许将数据从关系数据库映射到更加面向对象的业务模型。实体框架提供了使用 XML 文件定义这些层的方法。它还基于概念模型的架构生成了一系列类。可以针对这些类进行编程以直接与数据交互。这提供了抽象级别，因此开发人员可以针对概念模型而不是关系模型进行编程。实体框架可将针对概念模型编码的所有命令映射到逻辑模型中。

概念模型是使用概念架构定义语言（Conceptual Schema Definition Language, CSDL）在 XML 文件中定义的。CSDL 定义应用程序业务层所知道的实体和关系。逻辑模型（表示数据库架构）是使用存储架构定义语言（Storage Schema Definition Language, SSDL）在 XML 文件中定义的。例如，在概念模型中有一个实体，该实体实际是从数据库的多个表中派生其数据。概念模型和逻辑模型可按一对一的关系来关联实体。然而，EDM 的功能则不必以一对一的方式来链接实体，而由映射层（使用映射架构语言（Map Schema Language, MSL）定义的）来实现概念层和逻辑层两层彼此之间的映射，此映射使开发人员可以针对概念模型编写代码，并将这些指令映射到逻辑模型。

1.2 ADO.NET 数据操作模型

ADO.NET 是 .NET 框架类库的 System.Data 命名空间和它的子空间中提供的一组类，如图 1-3 所示。

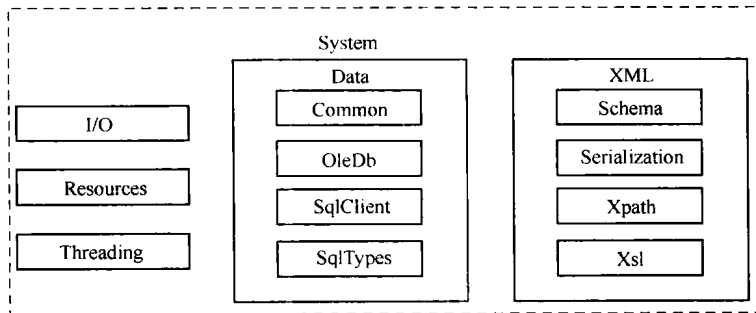


图 1-3 .NET 框架类库命名空间

与前任 ADO 和 OLE DB 不同，ADO.NET 从一开始就是为在 Web 的无连接世界中工作而设计的，并且很容易同 XML 集成，在关系数据和 XML 之间架起桥梁，简化了在它们之间移动数据的任务。