



数据库应用系列教材



.NET环境下 数据库应用系统 开发技术

刘友华 王贤平 吴伟 编著

 科学出版社
www.sciencep.com

数据库应用系列教材

.NET 环境下数据库应用系统 开发技术

刘友华 王贤平 吴 伟 编著

科 学 出 版 社

北 京

内 容 简 介

ADO.NET 是 .NET 环境用于操作数据库的首选技术之一。本书首先简单介绍 .NET 框架, .NET 环境下数据库应用的系统架构, 基本的数据库原理和 SQL 语言, 以及 .NET 环境下的编程语言 C#; 然后以 ADO.NET 和 SQL Server 2000 为核心, 详细介绍 .NET 环境下数据库应用系统的开发所涉及的几个主要方面, 包括数据库连接、数据获取、数据处理、数据更新、事务和安全等; 最后给出一个典型的开发实例, 使读者能综合运用上述知识, 较为系统地掌握 .NET 环境下的数据库应用开发技术。

本书适合作为高等院校非计算机专业学生的教材; 也适合对关系型数据库有一定了解, 并希望掌握 .NET 环境下使用 ADO.NET 技术的开发人员以及其他相关人员使用。

图书在版编目(CIP)数据

.NET 环境下数据库应用系统开发技术/刘友华等编著. —北京: 科学出版社, 2004

(数据库应用系列教材)

ISBN 7-03-014677-8

I. N… II. 刘… III. 计算机网络—程序设计—教材 IV.TP393

中国版本图书馆 CIP 数据核字 (2004) 第 125220 号

责任编辑: 鞠丽娜 陈砺川 / 责任校对: 柏连海

责任印制: 吕春珉 / 封面设计: 三函设计

科学出版社 出版

北京东黄城根北街16号

邮政编码: 100717

<http://www.sciencep.com>

新蕾印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2004 年 12 月第 一 版 开本: B5 (720×1000)

2004 年 12 月第一次印刷 印张: 19

印数: 1—4 000 字数: 352 000

定价: 25.00 元

(如有印装质量问题, 我社负责调换〈环伟〉)

《数据库应用系列教材》编委会

主 任 王 珊 中国人民大学

徐洁磐 南京大学

编 委 (按姓氏笔画排序)

马玉书 北京石油大学

王能斌 东南大学

孙志挥 东南大学

许龙飞 暨南大学

李庆忠 山东大学

李昭原 北京航空航天大学

邵佩英 中国科学院研究生院

沈钧毅 西安交通大学

邵晓英 宁波大学

单启成 南京大学

郭景峰 燕山大学

唐世渭 北京大学

黄上腾 上海交通大学

序

近年来,我国高等教育事业飞跃发展,在校学生人数突飞猛进,与此同时,高校教育改革逐渐冲破旧的计划经济模式,新的模式也正在建立。在这种形势下,旧的教材体系已不能适应新的需要,因此迫切需要建立新的教材体系。基于此种情况,我们以计算机相关专业中的数据库系统教材为依托,组织了一套适应不同需求、不同层次、不同目标的数据库系列教材,其组织依据是:

1. 在高等学校中随着老校的调整与改革,新校的不断涌现,过去计划经济的一刀切模式已逐渐改变,各校在培养目标、人才市场定位方面已出现多种模式(如研究型、应用型、开发型等),因此需要有多种不同数据库系统教材以适应不同模式的需求,而现有教材大多只能适应少数模式的需求。

2. 近年来计算机应用飞速发展,计算机与其他专业的交叉应用发展很快,如文科中的数量经济、信息管理、电子商务、财政金融等专业,理工科中的机械、建筑、城市规划、遥感遥测等都急需开设计算机及数据库等相应课程,也需相应的教材,而此方面的合适教材目前较为少见。

3. 随着教学改革的深入,数据库课程自身也需要进行改革,它除了需要有主课程外,还需要有若干门配套的辅助性课程与教材,如数据库分析与设计、Web 数据库、数据库应用等课程,以及数据库实验课、实习课以及习题集等配套教材。此外,还需配合使用现代化手段如电子教案及课件等相关音像制品。所有这些教材都需构成一个以数据库主课程为核心的有机组合的系列教材,而此方面的组合教材正是目前所缺少的。

4. 数据库技术本身发展很快,而教材编写相对滞后,同时国内数据库教材又受国外教材影响较大,因此适合国情的本土化教材的建设尤为重要,因此,能编写出既适应目前技术发展水平,又能适应我国经济发展需要的数据库教材是当前之急需。

5. 本系列教材能适应不同模式,不同层次、不同系科(计算机及非计算机专业)的需求,它除追求基本原理的正确性外着重在它的应用性。由于数据库是一门实用性很强的课程,我们希望学生在学了此课程后能在实际应用中发挥作用。

本系列教材正是为适应上面所述的需要而编写的,目前它以计算机及非计算机专业的本科生教材为主,并将逐渐扩充到研究生及大专层次。本系列教材采取开放性组织方式,今后将根据学科发展陆续组织出版数据库领域的优秀图书。

本系列教材的编写人员涉及各个不同层次与专业,有大量实际经验与理论水平,相信这套教材的问世能对数据库教学起一定的促进作用。

《数据库应用系列教材》编委会

2004年9月

前 言

随着计算机技术和网络技术的日益发展,建立数据库应用系统的需求在不断增加。与此同时,用来开发数据库应用系统的开发环境和集成工具也越来越多。.NET 技术为开发网络环境下的数据库应用系统提供了一个良好的开发环境和开发工具,特别是最近几年,在基于企业级数据库的分布式系统的开发中,.NET 已经成为与 J2EE 并列的首选技术之一。ADO.NET 作为.NET 环境下首选的数据访问技术,对原有的 ADO 技术做了较大改进。ADO.NET 使开发人员能够以较为灵活的方式查询和操作数据以及更新数据源,是所有数据驱动的.NET 应用程序或者 Web 服务的核心部件。

本书系统地介绍了.NET 环境下使用 ADO.NET 构建数据库应用系统中的数据访问层的步骤和方法。与其他介绍 ADO.NET 的同类书籍相比,本书着重从应用的角度,详细介绍如何使用 ADO.NET 技术建立一个数据库应用系统,避免了单纯从类库的角度来介绍技术本身的内容,使读者对.NET 环境下如何开发数据库应用系统有一个系统的、感性的认识。本书还从入门和实用的角度考虑,以读者熟悉的简单示例介绍了有关数据库的连接、访问及其他操作等内容。

本书共分 12 章。在介绍.NET 环境下的数据访问之前,第 1~3 章先介绍了.NET 环境下数据库应用开发的基础知识,其中第 1 章从整体的角度介绍了数据库应用系统的软件体系结构,第 2 章和第 3 章分别介绍了关系数据库和 C#编程语言的基础知识。第 4~9 章按照数据访问层的开发过程详细介绍 ADO.NET 的各个组成类以及这些类在数据库应用中的使用方法,其中,第 4 章介绍 ADO.NET 的内部机制,第 5 章介绍数据库连接,第 6 章介绍数据库查询,第 7 章介绍数据读取,第 8 章介绍数据更新,第 9 章介绍数据处理。第 10 章和第 11 章介绍了数据库应用系统的事务处理以及与性能相关的主题。第 12 章用一个完整的开发实例演示了.NET 环境下开发数据库应用系统的全过程。

本书的示例程序均用 C#语言编写,后台数据库使用的是 SQL Server 2000,因此本书简要介绍了 C#和 SQL Server 2000。当然,如果读者熟悉 C#编程语言和 SQL Server 2000 数据库,可以直接跳过这些章节。本书的示例程序均在.NET Framework 1.1 版本上编写并已调试运行。读者在使用本书示例上机实习时,要确保在所使用的计算机上已经正确安装了.NET Framework 1.1 或更高版本,建议读者安装.NET Framework SDK 或者 Visual Studio.NET 2003 作为开发环境和工具。

本书可作为高等院校非计算机专业数据库应用技术课程的教材,也适用于对网络环境,特别是.NET 环境下数据库应用系统开发技术有兴趣的一般读者。在每

一章的后面都有一定数量的习题，作为课后练习并帮助读者巩固重要知识点。

由于作者水平有限，书中疏漏之处在所难免，欢迎读者批评指正。

作 者

2004 年 8 月

目 录

第 1 章 数据库应用系统体系结构	1
1.1 数据库应用系统体系结构	1
1.1.1 以文件服务器为中心的计算模式	1
1.1.2 客户机—服务器计算模式	2
1.2 分布计算的软件环境	6
1.2.1 中间件技术——COBRA 和 COM+	6
1.2.2 开发平台——J2EE 和 .NET	11
1.3 数据库访问技术	13
1.3.1 ODBC	13
1.3.2 DAO	14
1.3.3 RDO	14
1.3.4 OLE DB	15
1.3.5 ADO	15
1.3.6 JDBC	16
1.3.7 ADO.NET	16
1.4 .NET 框架	17
1.4.1 什么是 .NET 框架	17
1.4.2 公共语言运行库	18
1.4.3 .NET 框架类库	19
1.4.4 通用类型系统与公共语言规范	21
1.4.5 中间语言及 JIT 编译	21
1.4.6 编程语言	22
1.4.7 元数据	23
1.4.8 程序集	24
1.5 .NET 环境下数据库应用系统的体系结构	26
1.5.1 N 层系统结构	26
1.5.2 .NET 对 N 层体系结构的影响	27
1.5.3 .NET 环境下数据库应用程序的体系结构	29
习题	30
第 2 章 关系数据库基础	32
2.1 关系数据库管理系统的主要功能	32

2.2 基本 SQL 查询语言	35
2.2.1 SQL 语句概述	35
2.2.2 SELECT 单表查询语句	37
2.2.3 SELECT 语句的查询条件	38
2.2.4 SELECT 中的集合函数与 AS 子句	39
2.2.5 SELECT 中行的分组和排序	40
2.2.6 多表查询	41
2.2.7 嵌套查询	42
2.2.8 Insert、Update 和 Delete 语句	43
2.3 SQL Server 2000 简介	44
2.3.1 关于 SQL Server 2000	44
2.3.2 SQL Server 2000 中可视化创建数据库	44
2.3.3 SQL Server 2000 对标准 SQL 语言的支持	47
习题	48
第 3 章 C#语法描述	50
3.1 数据类型	51
3.2 变量和常量	52
3.2.1 变量	52
3.2.2 常量	55
3.3 类型转换	56
3.4 表达式	57
3.5 流程控制	58
3.5.1 条件语句	59
3.5.2 循环语句	59
3.5.3 编译控制	61
3.6 类	62
3.6.1 类的声明	62
3.6.2 类的成员	63
3.7 继承	74
3.8 异常处理	76
习题	77
第 4 章 .NET 环境下的数据访问——ADO.NET	78
4.1 ADO.NET 工作原理	78
4.2 .NET 数据提供程序简介	79
4.2.1 .NET 的四种数据提供程序简介	79

4.2.2	Connection 类——连接数据库	80
4.2.3	Command 类——执行数据库操作	82
4.2.4	DataReader 类——读取数据	83
4.2.5	DataAdapter 类——数据更新	85
4.3	DataSet 类——数据处理	91
4.4	ADO.NET 和 ADO 的比较	94
	习题	96
第 5 章	连接数据库——Connection 类	97
5.1	.NET 数据提供程序	97
5.1.1	.NET 数据提供程序概述	97
5.1.2	四种 .NET 数据提供程序	98
5.2	用 Connection 类连接数据库	102
5.2.1	Connection 类	102
5.2.2	用 SqlConnection 连接到 SQL Server 数据源	104
5.2.3	用 OleDbConnection 连接到 OLE DB 数据源	109
5.2.4	用 OdbcConnection 连接到 ODBC 数据源	113
5.2.5	用 OracleConnection 连接到 Oracle 数据源	117
5.2.6	关闭数据库连接	118
5.3	各种常见数据源的连接字符串	119
	习题	122
第 6 章	数据库查询——Command 类	124
6.1	创建 Command 对象	124
6.1.1	可视化创建 SqlCommand 对象	124
6.1.2	直接用代码创建 SqlCommand 对象	127
6.2	执行数据库查询	127
6.2.1	ExecuteReader 方法	127
6.2.2	ExecuteNonQuery 方法	128
6.2.3	ExecuteScalar 方法	129
6.2.4	ExecuteXmlReader 方法	130
6.3	执行参数化查询	130
6.4	调用存储过程	133
6.5	用 DataReader 或者 DataSet 读取数据	136
6.5.1	将数据存储在数据集内	136
6.5.2	直接执行数据库操作	137
	习题	144

第 7 章 数据读取——DataReader 类	145
7.1 DataReader 基础	145
7.2 DataReader 操作	147
习题	150
第 8 章 数据更新——DataAdapter 类	151
8.1 DataAdapter 类	151
8.1.1 DataAdapter 类与其他查询类的不同之处	151
8.1.2 DataAdapter 类剖析	152
8.2 用 DataAdapter 类获取数据	156
8.2.1 创建 DataAdapter 类	157
8.2.2 读取数据	158
8.3 用 DataAdapter 类更新数据	162
8.3.1 直接使用 Command 类更新数据	162
8.3.2 使用数据适配器更新数据	165
8.4 使用事务更新数据和处理并发问题	169
8.4.1 使用事务更新数据	169
8.4.2 处理并发问题	170
习题	172
第 9 章 数据处理——DataSet 类	174
9.1 DataSet 类	174
9.1.1 DataSet 的功能和特性	174
9.1.2 DataSet 的结构	176
9.2 使用 DataSet 类	178
9.2.1 创建 DataSet 对象	178
9.2.2 从 DataAdapter 填充 DataSet	179
9.2.3 数据表 (DataTable)	184
9.2.4 数据列 (DataColumn)	190
9.2.5 数据行 (DataRow)	197
9.2.6 约束 (Constraint)	205
9.3 搜索、排序和筛选	207
9.3.1 使用 Find 方法筛选数据行	207
9.3.2 使用 Select 方法筛选数据行	208
9.3.3 数据视图 (DataView) 类	210
9.4 使用 DataRelation 处理 DataSet 中的关系数据	214
9.4.1 关系 (DataRelation) 类	214

9.4.2 创建 DataRelation 对象	214
9.4.3 导航表间数据	216
9.4.4 DataRelation 与约束	218
习题	218
第 10 章 事务处理	220
10.1 事务基础	220
10.1.1 什么是事务	220
10.1.2 事务的属性	221
10.1.3 事务状态	221
10.1.4 隔离级别	222
10.2 数据库应用系统中的事务	224
10.3 ADO.NET 事务支持	224
10.4 ADO.NET 事务对象	225
10.5 使用 ADO.NET 执行事务	226
10.5.1 在事务中执行 Command 对象	226
10.5.2 在事务中用 DataSet 更新数据源	228
10.5.3 Connection 对事务的限制	230
10.6 保存点	231
10.7 隔离级别	232
10.7.1 ADO.NET 和 SQL Server 的隔离级别	232
10.7.2 ADO.NET 中使用隔离级别	232
10.8 事务与性能	233
习题	234
第 11 章 性能	235
11.1 优化数据访问	236
11.1.1 使用 DataReader 还是 DataSet	236
11.1.2 存储过程	237
11.1.3 缓存数据	238
11.2 连接池	240
11.3 消息队列	242
11.3.1 消息队列的类型	242
11.3.2 发送消息	243
11.3.3 接收消息	244
习题	246

第 12 章 .NET 环境下数据库应用实例	247
12.1 个案提出	247
12.2 项目生命周期	247
12.2.1 需求分析	248
12.2.2 执行阶段的其他任务	249
12.3 客户管理案例开发细节	257
12.3.1 建立表结构	257
12.3.2 Windows 程序	258
12.3.3 客户管理项目中各个子模块的详细介绍	263
习题	286
主要参考文献	287

第 1 章 数据库应用系统体系结构

1.1 数据库应用系统体系结构

现代数据库应用系统的运行离不开网络环境，数据库技术的发展和网络技术的发展相互渗透、相互促进，成为计算机科学中发展迅速、应用最为广泛的两大领域。将数据库应用系统的资源分布到计算机网络中有多种方式，应用程序和数据库可以部署在同一个计算机系统中、同一个局域网的不同计算机系统中或不同局域网的不同计算机系统中，应用程序和数据库还可以再细分并且每一个子系统也可以独立分布。实际应用中的数据处理模式决定了数据库系统的应用环境，对这些处理模式的选择将直接影响到应用系统的开发。

1.1.1 以文件服务器为中心的计算模式

到 20 世纪 70 年代中后期，个人计算机（PC 机）因其用户界面友好、操作方便并具有一定的数据处理能力而逐步进入商用领域。单计算机结构将数据库应用系统资源放在单独的一台计算机系统以及与其直连的外围设备上，用户通过与那些直接与计算机相连的简单输入、输出设备与系统进行交互。但自 20 世纪 60 年代后期以来，计算机用于管理的规模越来越大，应用越来越广，数据量急剧增长，同时要求以多种应用、多种语言的方式来共享数据集合的要求越来越强烈。单个计算机的容量有限，因而计算机结构对于大型数据库应用系统来说不切实际甚至不具备可用性，对于这样的系统需要使用群集或多计算机结构，以文件服务器为中心的计算机模式就是这个时代的产物。这种计算机模式的体系结构是基于 LAN 的，由文件服务器（files-server）和工作站（PC 机）构成，而 LAN 的基本目的正是数据通信和资源共享。

在文件服务器结构中，应用程序在工作站上运行，文件服务器只提供了对数据及打印机等资源的集中存储、管理和共享访问途径。应用程序虽是运行在工作站上，也可以和数据库一样集中存放在文件服务器上。由于每次运行时都要访问服务器，时间开销大，也导致网络负荷加重。因此应用程序一般都拷贝到所有的工作站中，这样在执行应用程序时能定位到一台空闲的机器来处理，以便整个系统平衡分担处理负荷。图 1.1 描述了该计算模式的计算过程，应用程序执行处理过程中先向文件服务器发出数据文件请求，服务器响应请求并返回整个数据库文件，然后在工作站中继续进行处理。不难看出，该服务器就相当于一个“大硬盘”，

并对其中的数据库文件集中管理，这也是称之为“文件”服务器的原因。集中管理共享数据资源、应用程序逻辑自治管理是以文件服务器为中心的计算模式的主要特点。

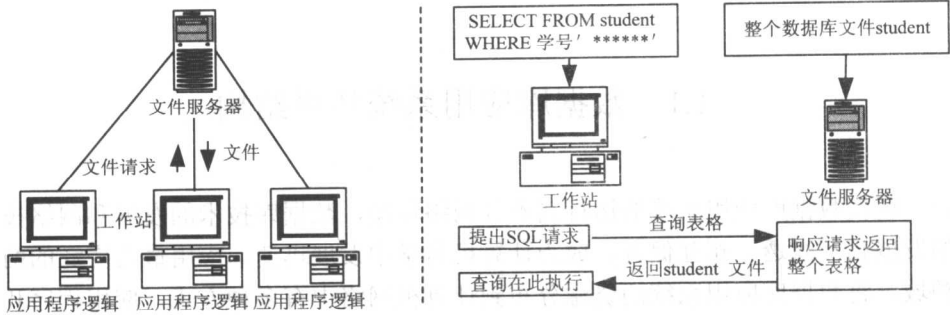


图 1.1 访问文件服务器中数据库的过程

虽然文件服务器结构类似于分布式配置，分散（decentralization）在网络中的计算机都具有可用性，但没能体现出多机之间的合作性，它们的功能和一个单独的大型计算机一样，所以仍将它归为集中式系统中的一种。在此结构中常会出现因工作站的计算和处理能力制约，应用程序功能、性能不能进一步提高，而不得不升级工作站的情形，而服务器高性能 CPU 并没有真正发挥作用。

1.1.2 客户机—服务器计算模式

现代数据库应用系统的部件通常分布于多个计算机系统和不同的物理位置上。例如，公司财务数据可能存储在一个集中式大型计算机上，区域办公室里的 PC 机要根据存储在大型机中的数据定期生成账目和其他报表，分散在很多地点的个人计算机可用于访问和查看定期报表，也可以直接更新中心数据库。这种把部件分布到不同计算机系统和不同位置的计算模式一般叫做分布式计算（distributed computing）或分布式处理，它依赖通信网络来连接地理上分散的计算机设备。和“集中式”系统相比，这种计算环境的本质就是共享资源并协同工作，即由若干个互联的计算机共同分担完成系统指定的某个计算任务，使得系统虽然是一个物理上的松散结合，但也是一个逻辑上的紧密结合。经过近 20 年分布系统的广泛应用，已经产生几种标准资源分布方法，而客户机—服务器结构，或称之为 C/S（Client/Server）结构就是当前分布式计算系统的一种主要应用模式。

客户机—服务器结构由两级构成：位于后台的服务器和位于前台的客户机。服务器用来管理一个或多个系统资源并通过确定的通信接口来提供对这些资源的访问，简单地说服务器就是在网络中为其他计算机提供服务的计算机，作用有管理共享的设备、控制对共享数据库的存取、接受并响应客户机的请求等；客户机则通过通信接口来请求资源，一般由网络中的服务器响应这些请求，客户机是向

网络中的服务器请求服务的计算机，作用有管理用户接口、采集数据、报告请求等。实现通信接口及应用接口的软件通常称为中间件（middleware）。

图 1.2 描述了客户机—服务器结构中最经典的应用程序—数据库应用系统的计算过程。在 C/S 数据库环境中，服务器管理数据库，客户机端运行应用程序。对于一个数据库应用系统来说，大量的数据管理、存取工作由服务器承担，这样可以充分利用服务器速度快、存储量大的特点，性能相对低的客户机只需负责用户方的数据处理和表示工作，可以充分发挥其对界面处理的长处。每当一个用户需要服务时，所有与数据库访问无关的处理部分均在客户机端进行，而与数据库访问有关的部分则由客户机通过应用程序接口发出请求，然后由服务器执行相应的服务，如查询一个数据库中的某条记录、更新数据库等，服务器将结果送回客户机提交给用户。这样，为了完成一个特定的任务，客户机上的程序和服务器上的程序可以协同工作，因此 C/S 计算模式的本质正如分布式计算一样，是双方负载的合理分配和协同处理的紧密结合。

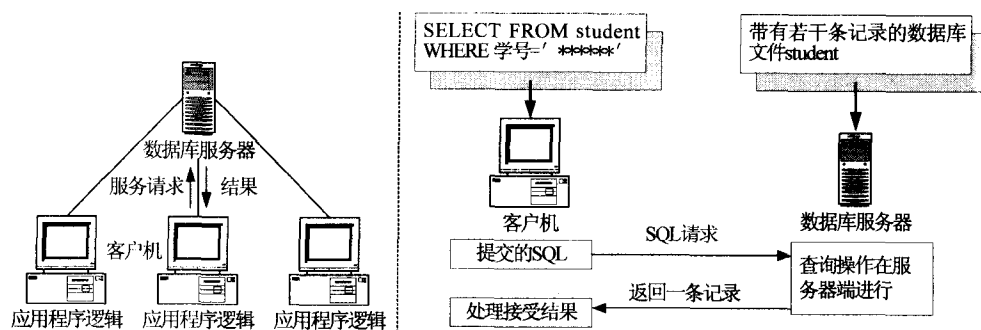


图 1.2 C/S 模式的计算过程

可见，客户机—服务器模式与文件服务器中心模式的最大区别在于服务器控制管理数据的方式由文件管理方式上升到数据库方式，这也是人们把 C/S 中的服务器称为数据库服务器的原因。文件服务器只负责文件的集中管理，并根据用户的请求向工作站发送文件，除此之外文件服务器不执行其他任何程序逻辑，倘若要查询的记录在发送的文件中不存在，则大量数据的网络传输实际上是无用的。而 C/S 模式下只传送处理结果，这样显著地提高了系统的传输效率，特别是当服务器硬件配置功能强大时，将会大大地减少处理请求的时间，从而缩短响应时间。因此，降低网络负荷、缩短响应时间是客户机—服务器计算模式的特点之一。

客户机—服务器计算模式的第二个特点是开放性，也就是客户机平台与服务器的平台无关。服务器上的数据库管理系统等软件集中管理服务器上的所有数据和资源，为各种各样的用户提供快捷、安全、可靠的存取访问，也就是说不管客户机端采用的是什么样的硬件平台和软件环境，都可以通过相应的网络协议和应用程序接口连接到服务器。

客户机—服务器计算模式的第三个特点是可伸缩性。客户机—服务器计算模式下的应用程序与数据库管理系统之间的关系由文件服务器中心模式下的多对一的关系转变到多对多的关系，在开放性原则的指导下，用户可根据应用的需要对系统规模进行增减。

虽然 C/S 模式的优越性已经逐步被人们所广泛承认，但 20 世纪 90 年代初基于 C/S 模式的应用并未大量出现，其主要原因是缺乏有效的开发工具作支持，应用程序对底层网络技术过分依赖，因此导致应用程序的可移植性差。对少数通信领域的专家来说直接针对各种底层（开放的和专用的）网络协议（TCP/IP、IPX/SPX、UNIX、OS/2、SNA 等）编写软件是适宜的，但对大多数开发应用程序的程序员来说，编写这种跨平台的、多协议、多编程语言的应用软件是很困难也很费时的的工作，所以当程序员要在底层网络协议上编程工作时，基于 C/S 模式的应用将很难发展。为了解决应用与网络过分的依赖关系，一个有效的方法就是在客户和服务器之间加一层称之为中间件的通信接口软件，如图 1.3 所示。

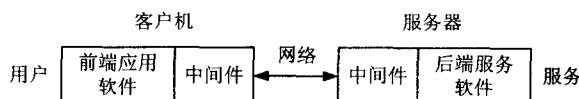


图 1.3 中间件的体系结构

中间件的主要功能就是通过简单的、较高层次的应用程序编程接口（API）将处于高层的应用和处于底层的网络屏蔽开来。从技术角度来看，中间件一般通过相应的支持软件来实现，它是一个逻辑层，并不一定要对应一个独立的物理层。从应用角度来看，中间件对网络的作用就像操作系统对本地计算机资源（硬盘、内存、外设等）的管理作用一样，在编写单机版本应用程序时，程序员不需去关心磁盘寻道、内存换页、I/O 端口等。所以，通过中间件应用程序可以方便有效地访问不同网络中的服务器，使得 C/S 计算模式的应用如火如荼地发展起来，甚至在一个大型系统中可以包括诸如网络服务器、文件服务器、数据库服务器、应用服务器、Web 服务器等专用服务器。

借助中间件技术，可以进一步将用户交互、应用业务处理和数据管理三种功能彻底分离，各自完成其擅长和应该完成的任务，就形成了所谓的多级分布结构模式。客户机—服务器的多级分布体系结构是把应用程序实现的功能划分为三个相互联系的不同层次，即三层（3-layers/3-tiers）体系结构，如图 1.4 所示，这里的三层不能理解成上述的中间件层。第一层是用户表示层，或者称视图层，它面向用户，包括用户界面和其他访问系统的组件。第二层是应用逻辑层，在三层结构中也称作中间层，它面向商业或企业等的业务处理过程，包括应用系统的程序逻辑的实现只在这一层次强调的是组件开发，即将原先客户端的很多处理逻辑