



普通高等教育“十二五”应用型本科规划教材
上海市“085工程”资助出版精品教材

数据库应用系统开发过程、 技术及案例详解

主编 吴爱华

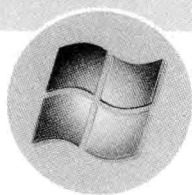


中国人民大学出版社



普通高等教育“十二五”应用型本科规划教材
上海市“085工程”资助出版精品教材

数据库应用系统开发 过程、技术及案例详解



主 编 吴爱华

中国人民大学出版社
• 北京 •

图书在版编目 (CIP) 数据

数据库应用系统开发过程、技术及案例详解/吴爱华主编. —北京: 中国人民大学出版社, 2015. 7
普通高等教育“十二五”应用型本科规划教材 上海市“085 工程”资助出版精品教材
ISBN 978-7-300-21716-1

I. ①数… II. ①吴… III. ①数据库系统-课程设计-高等学校 IV. ①TP311.13

中国版本图书馆 CIP 数据核字 (2015) 第 167262 号

普通高等教育“十二五”应用型本科规划教材
上海市“085 工程”资助出版精品教材

数据库应用系统开发过程、技术及案例详解

主编 吴爱华

Shujuku Yingyong Xitong Kaifa Guocheng Jishu ji Anli Xiangjie

出版发行 中国人民大学出版社

社 址 北京中关村大街 31 号

邮政编码 100080

电 话 010-62511242 (总编室)

010-62511770 (质管部)

010-82501766 (邮购部)

010-62514148 (门市部)

010-62515195 (发行公司)

010-62515275 (盗版举报)

网 址 <http://www.crup.com.cn>

<http://www.ttrnet.com> (人大教研网)

经 销 新华书店

印 刷 北京密兴印刷有限公司

规 格 185mm×260mm 16 开本

版 次 2015 年 8 月第 1 版

印 张 26.5

印 次 2015 年 8 月第 1 次印刷

字 数 615 000

定 价 55.00 元

版权所有 侵权必究 印装差错 负责调换

内容简介



本书以数据库应用系统的开发为核心，通过丰富的案例，系统地讲解了其开发过程、分析方法和技术细节。主要包括：软件过程模型和项目组织方法，数据库设计，Windows 窗体程序开发技术细节，如数据库访问技术、事件驱动机制、ActiveX 控件及其使用、数据库编程。本书特别注意系统开发中的原理解释，以便读者可以将数据库原理、面向对象思想和数据库应用系统开发的现实结合起来，达到全面实训的目的。

本书结构清晰、内容翔实、语言朴实、示例丰富，力求用容易理解和激发思考的方式使读者掌握数据库应用系统开发的基本原理及技术。本书可作为高等院校计算机各相关专业学生的“数据库课程设计”教材，同时也适合对数据库应用系统、Windows 窗体开发，特别是对数据库设计感兴趣的读者。



随着计算机技术的普及,为了提高效率、增加竞争力,越来越多的企事业单位,甚至普通大众都趋向于将其工作和生活中的主要业务由人工管理转换为计算机管理,过程中产生的各类数据也由纸质文件档案转变为计算机内的数据库,这样的系统就是数据库应用系统。毫不夸张地说,在目前涉及计算机软件开发的所有应用系统中,80%属于数据库应用系统,它们广泛存在于工业、农业、交通运输、生活娱乐、体育卫生、文化教育等几乎所有的社会领域。由此,数据库应用系统的开发已经成为计算机类专业学生必须具备的基础知识。

“数据库课程设计”或类似课程在我国绝大多数高校的计算机、信息、软件工程和网络专业都已开设多年,是本科生的一门异常重要的必修专业课。这门课一般要求学生实际业务调研之后,综合运用数据库、面向对象、信息系统分析和设计等多门课的知识,通过编写代码完成一个可以实际使用的数据库应用系统,是非常有价值且重要的课程。

但由于课时的限制,大部分高校在“数据库课程设计”这门课之前,仅安排了C/C++、面向对象程序设计、数据结构、操作系统、数据库原理等基础课,这些课虽然也要做实验、编程序,但只需要编写小型函数、测试算法或语言,学生缺少编写大型程序的实际训练,更没接触过可视化编程,也缺少开发数据库系统的理论指导。这导致很多学生拿到课程设计题目时无从下手,特别是没选修过可视化编程及信息系统分析和设计(软件工程)类课程的同学,有的学生一开始甚至连窗体是什么都不知道,更别提怎样才能组团完成一个系统的开发。常见的现象是,由于设计不充分,急忙编码,或者组内沟通不足,各忙各的子模块,系统开发最后进行不下去,要推翻重来,因此迫切需要针对零基础的数据库应用系统开发的指导教材。

本教材主要讲述数据库应用系统的分析、设计、编码实现等整个过程中涉及的理论基础、实践技术方法以及项目过程管理等,是作者近十年来从事数据库原理与面向对象程序设计课程教学、科研与项目开发的经验总结。编写目标是让读者能了解、学会并思考数据库应用系统开发中的技术要点。

基于这个目的,编者在编写过程中,遵循了以下原则:

1. 定位清晰,实用性强。作为“数据库原理”的配套课程设计教材,本教材默认读者具有一定的理论基础,没有过多地阐述数据库理论,而是主要阐述数据库应用系统开发

的问题。围绕这个中心,详细阐述了开发过程中最常碰到的各种问题,并按照顺序将内容组织为三个层次:软件过程、开发技术和实际案例。从宏观思路到技术细节都有涵盖,非常实用。

2. 结合实际,难度适中。本教材针对零基础的学生,用语通俗,尽量采用简单易懂的语言去解释概念。采用的案例、提出的思考问题,很多都来自实际项目。整本书的内容从宏观到微观,阐述的案例由浅入深,重点突出了数据库设计这部分内容。

3. 案例丰富,理论联系实际。本教材通过大量实例来阐述抽象的概念和复杂的理论,深入浅出,便于理解。不单在前5章讲解知识点时,给出了很多课堂实例,还有2章是专门的数据库设计案例,以及2章的完整系统开发案例,让读者充分了解数据库应用系统的分析、设计和开发细节,并启发读者抽象分析问题的思维。这些案例大部分都来自实际项目,由简单到复杂,可以作为实际开发工作的参考。

本教材共有9章:第一章概述了数据库应用系统的基本概念及其开发所需的主要技术,还介绍了两类数据库应用系统开发的环境配置,主要包括:SQL Server 2008 的安装,Visual Studio 2010 的安装,IIS 的安装和配置,JDK、Tomcat 和 Eclipse 的安装和配置过程。第二章介绍了数据库应用系统及其开发步骤,详细说明了系统开发的每一个环节,包括需求分析、系统设计和数据库设计的方法与问题讨论,说明了常见需求文档中的图形,如数据流图、业务流图、各类 UML 图。第三章说明了数据库开发的详细步骤,详细介绍了其中的调研、概念设计、逻辑设计和物理设计等步骤需要完成的工作,讨论了一些常见的复杂案例的数据库设计方案。第四章讲解事件驱动程序的运行机制,并详细介绍了 Windows 窗体中常用的控件,以及其中蕴含的面向对象编程思想。第五章以 ADO、ADO.net、JDBC 等为例,介绍了常见环境下的数据库访问技术。第六章和第七章分别以一个较简单的实际应用系统为例,详细说明了系统的分析设计过程,以及系统的编程实现技术细节。第八章和第九章则分别以一个较复杂的实际项目为例,说明了数据库的分析设计过程。

本书适合作为高等院校非计算机专业学生学习数据库技术和 .NET 程序设计的教材,也适合具有初步编程知识并对数据库编程感兴趣的读者自学。

本教材的第一、三、八、九章由吴爱华编写;第二章由王洪编写;第四、六章由王卫华编写;第五、七章由宋森编写。另外,王洪也参与编写了第 3.3.1.2 节。每个编者都曾经担任过相关课程的教学工作,对所负责编写的内容均有一定的经验。在此向他们致以诚挚的谢意。

由于时间仓促和编者水平有限,书中错误和不妥之处在所难免,敬请广大读者批评指正。

编者

2015 年 5 月

目 录



第一章 概述	1
1.1 数据库应用系统	1
1.2 开发数据库应用系统	4
1.3 环境准备	8
1.4 本章小结	34
第二章 数据库应用系统开发步骤和方法	35
2.1 数据库应用系统的生命周期模型	35
2.2 信息系统的开发方法	36
2.3 需求分析	40
2.4 系统设计	50
2.5 面向对象分析和设计	54
2.6 软件过程管理	65
第三章 数据库设计	68
3.1 简述	68
3.2 引例——大学水电费管理信息系统	69
3.3 数据库的生命周期与各阶段任务	80
3.4 案例分析——疑难数据库设计	113
3.5 本章小结	121

第四章 事件驱动和 Windows 窗体应用程序	125
4.1 事件驱动的编程机制	125
4.2 Windows 窗体控件技术	128
4.3 常用控件	135
4.4 本章小结	162
第五章 数据库引擎及其数据库访问技术	163
5.1 数据库访问技术概述	163
5.2 SQL Server 数据库简介	166
5.3 ADO.NET 数据库访问技术基础——以 C# 为例	173
5.4 ADO.NET 数据库访问技术进阶——以 C# 为例	186
5.5 JDBC 及其数据库访问技术——以 Java 为例	215
5.6 本章小结	221
第六章 学生选课管理系统	223
6.1 需求分析	223
6.2 概念结构设计	230
6.3 逻辑结构设计	231
6.4 学生选课管理系统的实现	237
6.5 本章小结	297
第七章 化妆品销售信息管理系统	298
7.1 需求分析	298
7.2 系统设计	301
7.3 环境准备	303
7.4 基础支撑类的实现	307
7.5 用户登录模块的实现	318
7.6 销售管理模块的实现	322
7.7 后台管理模块的实现	328
7.8 代码合并	348
7.9 本章小结	349

第八章 数据库设计案例——1 元抢购网站 350

8.1 系统概述 350

8.2 调研结果 351

8.3 需求分析 360

8.4 数据库设计 374

8.5 本章小结 385

第九章 数据库设计案例——某高校教材管理系统 386

9.1 概述 386

9.2 调研结果 387

9.3 需求分析结果 399

9.4 数据库设计 406

9.5 本章小结 412



第一章 概述



1.1 数据库应用系统

什么叫数据库应用系统？也许你不熟悉这个名词，但你一定对某个具体的数据库应用系统并不陌生。生活中，工作中，数据库应用系统比比皆是：走进大学图书馆，你要借助**图书管理系统**查询意向图书的位置和借用情况，而管理员则要借助这类软件管理图书及其借阅信息；自习前，你可能要在教学楼下的某台机器前查询空闲教室信息，使用的是**教室查询系统**；超市收银台电脑上工作的**超市销售系统**；订火车票的**12306 网上订票系统**；看病挂号的**医联预约平台**；**电子商务网站**……所有需要使用数据库来存储信息的应用系统都叫数据库应用系统。

严格来讲，数据库应用系统是一类软件的总称，它们都以数据库为信息存储平台，以数据库管理系统（DBMS）为数据管理者，以数据的增、删、改、查询和统计为主要功能，面向具体业务，通常采用可视化编程技术编写而成。无论是面向内部业务的管理信息系统，还是面向外部提供信息服务的开放式信息系统，都以数据库为基础和核心。

简单地，可以将数据库应用系统理解为帮助人们管理单位或个人业务数据的软件。比如在图书馆业务中，工作人员需要记录并管理每一本图书、杂志和电子刊物的信息以及它们的借阅信息。没有计算机的时代，工作人员把信息记录在卡片和档案本里，新书采购之后要在这些纸面上增加新书信息，图书销毁以后要删除相应数据，图书借走以后，要登记本次图书借阅信息，归还以后又要记录好归还信息……有了电脑以后，上述所有数据都可以存在数据库中，日常工作就变成了数据库的增、删、改和查询。可是，第一，不是所有工作人员都学过计算机，可能很多人不懂如何操作数据库管理系统（如 Oracle）；第二，每个人都直接操作数据库，不利于数据安全控制和敏感数据保护；第三，复杂功能不能通过简单查询实现。因此，程序员们开发了可视化应用程序，以便图书管理员们可以在这些应用程序的界面上，通过简单的鼠标操作查询、统计和管理数据。这样的可视化应用程序就是数据库应用系统。

数据库应用系统是应用最广且开发最多的一类软件，也是计算机、信息管理与信息系统专业本科阶段必须深入了解并掌握其开发技巧的一类软件。

1.1.1 数据库应用系统的组成

典型的数据库应用系统由数据库、数据库管理系统、应用软件及其界面组成。其中，数据库占据了核心地位，它是所有业务数据的集合，也是整个系统服务的对象；但数据库不能凭空出现在计算机中，人们需要使用数据库管理系统这样的软件来创建、查询和管理数据库；另外，数据库管理系统提供了各类编程接口，应用软件必须通过这些接口才能操作数据库；而应用软件则负责为用户提供操作界面，并实现用户的操作诉求。

另外，也有人认为数据库应用系统由软件、硬件和人员组成。数据库应用系统开发完成以后，要部署到使用单位，就需事先配建好计算机及其网络（硬件），安装好已完成的系统及其支撑软件，如操作系统、数据库管理系统及其他必备组件。而使用单位各业务部门的工作人员或服务对象、日常管理该系统的 IT 部门职员，就是运行阶段系统的人员构成。一般单位都会指定一个 IT 部门员工担任系统的数据库管理员（Database Administrator, DBA），全权负责数据库的日常工作，如：其他用户访问数据库的权限分配；日常数据备份和还原；监控数据库使用情况，并适时调整系统软硬件配置，以提供流畅的数据访问服务；审计并追踪非法用户访问；修补系统 bug 等。

图 1—1 从软件环境角度，以图书馆管理系统为例，描述了一个典型的数据库应用系统的构成，而图 1—2 则描述了其软硬件部署。从图中可以看出，数据库以某种文件的形式储存在计算机硬盘上，当数据量太大时，则存放在磁盘阵列上。一般企业会配一个专门的服务器，专门处理用户的数据查询和管理请求，具体由 SQL Server、Oracle 这样的 DBMS 实现。DBMS 一般由服务器和客户端两部分组成，顾名思义，客户端安装在各个用户终端上，负责收集数据请求；服务器端则安装在数据库服务器上，负责处理数据请求。一旦终端用户（用户可以是某个工作人员，也可以是某个应用程序）提交了某个查询请求，服务器将解析该请求，通过操作系统的文件系统到硬盘/磁盘阵列上取得该数据，并反馈给用户。因而，读者或管理员们需要借助图书管理系统这样的软件来提出增、删、改、查询数据的请求。这类软件采用高级语言编写而成，需运行在一定的编译系统和工具包之上。终端用户多数据请求频繁的系统，会专门设置应用程序服务器，处理非数据请求。

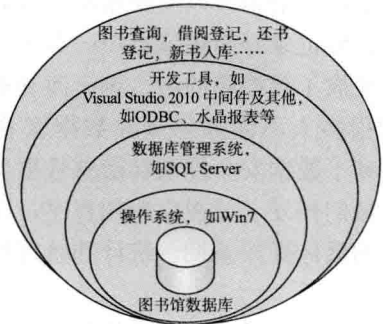


图 1—1 典型的数据库应用系统的构成

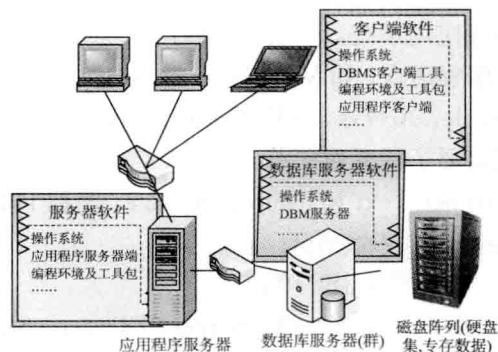


图 1—2 数据库应用系统的典型软硬件框架

1.1.2 数据库应用系统的类型

数据库应用系统可以有很多种分类方式，但提及最多的是依照架构方式将其分为 B/S (Browser/Server) 结构和 C/S (Client/Sever) 结构的数据库应用系统。开发这两类数据库应用系统的技术要求是不同的。

简单来说，图 1—2 中，如果服务器和客户端采用局域网连接，且系统界面是传统的 Windows 窗体，那么该数据库应用系统是 C/S 结构的。而如果服务器和客户端采用 Internet 连接，且系统界面是浏览器，那么该系统是 B/S 结构的。

C/S 结构，是一种基于客户机/服务器模式的软件架构，系统由前台客户端+后台服务器组成。一般将软件分为客户端和服务端两层，数据量大的也可分成图 1—2 所示的数据库服务器、应用程序服务器和客户端三层，而数据量巨大的，甚至将每个服务器以一群服务器代替。以现在流行的“胖客户端”两层 C/S 结构为例，客户端负责实现界面显示和核心业务功能。在数据处理完成后，将结果通过网络存储到服务器上运行的数据库管理系统中。这种构架的不足是前端负载较重。为了减少前端负载，可以将核心业务功能转移到专门的应用程序服务器上。

B/S 结构，也叫浏览器/服务器机构，是随着 Internet 技术的兴起，对 C/S 结构的一种改进。不同于 C/S 结构的“胖客户端”模式，B/S 结构只有极少部分的业务逻辑在浏览器前端实现，而其主要的业务逻辑完全在应用服务器端实现，甚至用户表现也全部在 Web 服务器上实现，客户端基本只实现数据输入和结果呈现，也被称为“瘦客户端”或“灵敏的客户端”结构。

C/S 结构和 B/S 结构各有彼此的优势和劣势，并不能说哪一个结构更加优秀。就 C/S 结构来说，它的优点是界面和操作较为丰富，安全性较好，而且由于业务操作都是在客户端完成的，响应速度很快。它的缺点在于通常只适用于局域网中，升级成本较高，因为每一次升级都要求修改所有客户端程序。B/S 结构则刚好相反，它的升级很方便，如果业务逻辑发生变化，只需修改服务器代码即可；安装部署也很方便，只要安装有 Web 浏览器即可使用。B/S 结构的问题在于速度和安全性，若要达到 C/S 结构的同等性能，需要巨大的设计成本。并且 B/S 结构是基于请求—响应模式，在使用过程中经常需要刷新页面，这

也是一个缺点。

开发 C/S 结构的数据库应用系统，一般需掌握：

- Windows 窗体类的编程语言，如 Visual C++，C#，VB.net，Java 等；
- 数据库技术，如 Oracle，Access，SQL Server，DB2 等。

而开发 B/S 结构，则需要了解下列技术：

- 超文本标记语言（Hyper Text Markup Language，简称 HTML）；
- 脚本语言，如 Javascript，VBScript；
- 服务器端程序语言，如 ASP，JSP 等；
- 数据库技术，如 Oracle，Access，SQL Server，DB2，MySQL 等。

本书后面的案例将以 C# + SQL Server 为例详细介绍 C/S 结构的数据库应用系统开发技术细节。



1.2 开发数据库应用系统

1.2.1 数据库应用系统开发过程及注意事项

开发数据库应用系统是一个系统性的工作，需要完成调研、需求分析、系统分析、系统设计、编码实现、测试和部署实施等多个阶段任务。过硬的代码能力只是保证项目成功的因素之一，扎实的理论基础、分析并抽象复杂问题的能力、沟通和团队合作等也是完成一个质量合格的软件成品必不可少的能力。

刚接触项目开发的新人，往往着急编写代码，编着编着发现 bug 越来越多，很多问题没法解决，再返工修改，然后可能涌现更多问题，最后陷入泥潭，这些都是因为开发过程没有遵从良好的开发模型，前期调研、分析和设计做得不够导致的。优良的开发模型是一个数据库应用系统成功的重要保障，好的开发模型有助于控制项目进度、降低风险。因此，开发数据库应用系统的第一个注意事项就是不要盲目开展工作，工作安排要遵从一定的开发模型。

典型的开发模型包括：瀑布、迭代、快速原型、螺旋和敏捷开发模型。

瀑布模型，也叫结构化开发，是一种经典的计算机软件开发方法。它将软件的开发过程严格地划分为立项、调研、需求分析、系统设计、编码、集成、测试、部署运行与维护等阶段，要求只能依次序进行，每个阶段的工作完成后才能开展下一阶段的工作。

迭代式开发将整个软件开发组织为一系列小项目，即一系列的迭代，起初只完成一个迭代，在得到客户的反馈认可后，再在上一次的成果上开展下一次迭代，直到完成。每一次迭代都包括了需求分析、设计、编码与测试。

快速原型模型一般根据客户需要，在很短的时间内实现最迫切的需求，完成一个可以演示的产品，即原型。原型只是达到功能的实现，充当抛砖引玉的角色，以确定用户的真正需求，能激发业务人员的潜在需求。需求确定之后，原型可能被抛弃，也可能作为软件的一部分被保留下来。

螺旋模型是瀑布模型和快速原型的结合，其核心在于项目起始阶段无需将所有事情都定义得清清楚楚，而是把最重要的功能定义出来，实现它，然后听取客户的意见，之后才进入下一个阶段。如此不断重复，直到得到最终产品。螺旋模型的每一个阶段都经过：(1) 制订计划：确定软件目标，选定实施方案，弄清项目开发的限制条件；(2) 风险分析：分析评估所选方案，考虑项目风险的识别和消除；(3) 实施工程：实施软件开发和验证；(4) 客户评估：评价开发工作，提出修正建议，制订下一步计划。

敏捷开发，是 20 世纪 90 年代出现的新型软件开发方法，目前备受关注。敏捷开发更强调程序员团队与业务人员之间的协作，以及面对面的沟通，也更注重软件开发中人的作用。敏捷开发思想认为：人和交互重于过程和工具；可以工作的软件重于求全而完备的文档；客户协作重于合同谈判；随时应对变化重于循规蹈矩。由于项目规模会增加面对面沟通的难度，因此敏捷方法更适用于较小的队伍。

传统的瀑布式开发，要求每个开发阶段都做到最好，特别是前期阶段，设计得越完美，提交后的成果损失就越少。迭代式开发不要求阶段工作成果完美，即便发现了不足之处，也不急于完善它，而是以搭建主要功能为目的，争取以最短的时间、最少的代价先完成一个“不完美的作品”，再通过客户反馈，逐步完善它。螺旋开发很大程度上是一种风险驱动的方法体系，因为在每个阶段之前及经常发生的循环之前，都必须首先进行风险评估。敏捷开发和迭代式开发一样，强调在较短甚至更短的开发周期内提交软件，并更强调队伍中的高度协作，是一种适应性的方法。

本书第二章将重点介绍结构化开发方法、原型方法和面向对象的开发方面的内容。

另外，沟通不足也是一个常见的导致项目失败的原因。沟通不足包括内外两个方面。对外沟通，即开发人员和业务人员之间的沟通，这层沟通越顺利，双方越能对系统功能和系统目标达成共识，这将大大降低因需求不符而导致的设计重做、代码重写的风险，原型则是一个很好的对外沟通工具。对内沟通，即需求分析、系统设计和代码实现人员之间的沟通，团队内部需要经常沟通，以达到对解决方案的共识，避免各自按自己的理解开展工作，最后矛盾冲突，导致项目推翻重来。

因此，在数据库应用系统的开发过程中，特别是团队合作时，要对各阶段任务认真对待，有疑问或不确定之处一定要和团队内其他成员商量，切不可闭门造车。首先，有改变的代码细节，如变量取值含义、数据库表结构等，只要是可能影响到别人的，都要及时告知。其次，事先应制定统一的编码规约，比如变量命名、全局变量和全局函数的设定等，编码时一定要遵从统一约定。再次，要养成及时给代码写注释的习惯，这样就不会发生自己看不懂自己写的代码的情况，日后差错调优时，就不用重新整理思路；而对整个函数或程序文件也要进行注释，说明函数的功能、输入输出参数的函数、和其他函数之间的关系，以及历次修改和版本。最后，程序开发需要一个团队的力量，项目开展中，团队要定时开会，协调、解决好整个程序编写过程中遇到的问题，并讨论下一步的工作安排。

在做课程设计时，建议同学们先充分分析问题并讨论解决方案，小组一起完成需求分析、系统总体设计和数据库设计，然后再各自设计和编码实现个人部分。不要一上来就简单地分任务，然后各自分析、设计、实现自己那一小部分，而不和其他同学沟通，这样出来的成品多半会失败。

1.2.2 数据库编程接口

编写数据库应用系统，不可避免地需要访问数据库。比如：“用户登录”是常见的软件模块，可以用下面的伪码来描述其代码思想：

用户登录模块的伪码
1) 验证用户是否输入了“用户名”、“密码”等必要字段 2) 构建一个查询语句，用以查询数据库的用户表中是否存在该用户 3) 链接数据库 4) 向数据库提交已构建的查询语句，并获得查询结果 5) 如果查询结果为空，该用户不存在，拒绝登录系统 否则，转到相应的功能界面

我们来分析一下，要把这段伪码转换成真实代码，会碰到什么困难。
第一步很简单，“用户名”和“密码”一般以文本框的形式给出，该文本框的 text 属性代表的就是其内容，代码只要判断这两个文本框的 text 属性值是否为空字符串，是的话，就弹出消息框，提醒用户正确输入“用户名”和“密码”。

第二步也不难，就是构建一个以此查询为内容的字符串，如：
`Sql = "select * from USER where uname = '" + 用户名 .text + "' and pwd = '" + 密码 .text + "'";`

但是，第三步和第四步有点迷惑，C# 等编程语言，如何指定要链接的数据库，又如何连接上数据库？如何向 DBMS 提交查询语句，又如何获取查询结果？有什么语句或者函数可以使用吗？

数据库编程接口就是由各个编程环境定义的专门用于连接、访问和使用数据库的类及其函数和属性成员组成的一个工具包，可以供程序员调用。不同的数据库，不同的编程环境，其编程接口可能不尽相同。

比如，微软提供的 ADO.net，就是用于 .NET 体系架构内的数据库编程接口，VB.NET、C#、Visual C++ 2010 等语言均通过这个接口连接、访问和使用数据库，下面是 ADO.net 提供的几个类，具体用法请阅读第五章：

- SqlConnection: 用于连接 SQL Server 数据库的类，其属性 `ConnectionString` 用于说明要连接的数据库服务器和具体数据库；而其方法 `open ()`，用于实际连接上数据库；方法 `close ()` 则用于关闭数据库连接。
- SqlCommand: 执行 SQL 查询语句或者调用存储过程的类，其属性 `CommandText` 用于说明要执行的 SQL 查询语句或要调用的存储过程；其属性 `Connection` 用于说明 SQL 语句将在哪个 `SqlConnection` 对象上执行，即向哪个数据库提交查询语句；其方法 `ExecuteReader` 用于执行一个“select”命令，并得到查询结果；其方法 `ExecuteNonQuery` 用于执行一个“insert”、“update”或“delete”查询。
- SqlDataReader: 表示查询结果的类。

早期的程序员在程序中要连接数据库是非常困难的，每种 DBMS 产生的数据库文件

的格式都不一样，程序员要对他们访问的 DBMS 的底层 API 非常了解，才能访问该 DBMS。这就产生了一个问题，当 DBMS 发生改变，或用户习惯使用的 DBMS 与开发程序使用的 DBMS 不符合时，应用软件便无法正常访问 DBMS。因此，能处理各种数据文件的 API 便产生了，这就是 ODBC。ODBC 基于 SQL 查询语言，是一个接口标准，大多数 DBMS 都提供了 ODBC 驱动程序。虽然 ODBC 统一了对多种常用 DBMS 的访问，但仍需要很多低级调用才能访问数据库。

微软先后还提出过多个数据库编程接口，依次序包括：DAO (Data Access Objects)，RDO (Remote Database Object)，OLE DB (Object Linking and Embedding Database)，ADO (ActiveX Data Objects) 和 ADO.NET。目前，除了 DAO 和 RDO，其他接口仍被经常使用。另外，Java 程序也有自己的数据库编程接口，即 JDBC。

OLE DB，数据库链接和嵌入对象，是微软提出的基于 COM 思想且面向对象的一种数据接口，除了关系型数据之外，它还可以用于访问邮件数据、Web 文本或图形、目录服务及主机系统中的文件等。OLE DB 的核心内容就是对以上各种数据都提供相同的访问接口，使得数据的使用者可以使用同样的方法访问各种数据，而不用考虑数据的具体存储地点、格式或类型。OLE DB 通过 OLE DB Provider 访问数据，比如，调用 OLE DB Provider for SQL Server 就可以直接访问 SQL Server 数据库。

ADO，ActiveX 数据对象，也是微软推出的一种面向对象的数据库访问接口。由于 OLE DB 标准的 API 只能供 C++ 语言调用，为了使流行的各种编程语言都可以编写符合 OLE DB 标准的应用程序，微软在 OLE DB API 之上，提供了一种面向对象、与语言无关的应用编程接口，这就是 ADO。实质上，ADO 是将 OLE DB 面向 C++ 的复杂接口封装起来，提供便于操作的接口。

本书第五章将更详细地讲解各类数据库接口技术，并将结合例子具体讲解 ADO.NET 的数据库访问技术细节。

1.2.3 开发数据库应用系统的技术背景

要想成功开发一个数据库应用系统，除了熟练掌握一门具体的编程语言之外，至少还要具备以下五个方面的基础知识：面向对象编程，事件驱动的程序运行机制，Windows 窗体编程，数据库及其设计的理论，以及 SQL 语言。

一般来说，数据库应用系统，不管是 BS 结构，还是 CS 结构，编程语言不管是 C#、VB.NET、Visual C++ 还是 Java，都需要用到数据库编程接口来访问数据库，需要采用 ActiveX 控件做界面。而每个控件都是一个类，数据库编程接口提供的也是类，这就要求大家一定要明白面向对象编程技术，知道什么是类，什么是对象，对象的属性和方法，它们的访问机制，类的派生、多态性、模板等基本概念。本书默认大家都已经具备这些知识，如果有读者还有迷惑，可以从专门的面向对象编程教材中了解到具体知识。

CS 结构的程序，一般都采用 Windows 窗体和 ActiveX 控件做界面。Windows 窗体大家非常熟悉，几乎所有大家熟知的程序都使用窗体做界面，而窗体内的按钮、文本框、单复选、列表等都是 ActiveX 控件。要从界面上获取用户输入的信息，并把处理结果呈现在界面上，都要求程序员熟知窗体和控件的各个属性的含义、各个方法的作用和调用方式。

另外，事件驱动的程序，为窗体和控件定义了很多触发事件（一种函数），用于响应用户在界面上对这些控件或窗体的鼠标或键盘动作，比如按钮，有点击事件，事件内的代码用于响应用户在该控件上的单击鼠标左键行为。这些都要求程序员了解事件驱动的程序和 Windows 窗体编程。BS 结构的程序，也要采用 ActiveX 控件做界面，也需要在脚本程序中编写一些事件响应程序，甚至很多网站采用 C# 做界面，ASP 做数据处理，这样一来，BS 结构的程序也一样要求程序员了解事件驱动的程序和 Windows 窗体编程。

数据库设计是系统的核心内容，因而数据库设计绝不能简单处理。好的数据库设计对系统的正确性和效率至关重要，这要求大家具有一定的数据库理论，了解范式理论，知道关系模式的分解方法，明白如何得到并分析数据流图和数据字典，掌握数据库设计的步骤和各个步骤的工作方法。本书第三章完整地介绍了数据库设计的流程和各阶段的工作，至于范式理论和模式分解，读者可以参考相关书籍。本书第八章和第九章各通过一个综合例子，说明了数据库设计的一般流程和方法。

本书第四章将为大家介绍事件驱动的程序和面向对象的关系，以及 Windows 窗体编程中常用的控件，本书没有介绍的控件，大家可以参看 MSDN 在线帮助（<http://msdn.microsoft.com/zh-cn/default.aspx>）。

本书第 5.2 节介绍了开发数据库应用系统中常用的 SQL 语言，读者可以从其他专门介绍 SQL 语言的资料中学习到复杂 SQL 查询的写法。



1.3 环境准备

DBMS 和编译环境的安装，以及必要的系统环境变量配置，是编码之前必须完成的准备工作。

1.3.1 数据库管理系统及其安装

数据库管理系统，即 DBMS，是一款专用于数据库管理的系统软件。依据其底层数据模型，可以划分为三个主要类别：层次、网状和关系型。目前大多数数据库应用系统都是基于关系型数据的，因此，本小节将简单介绍几种常用的关系型 DBMS，并详细说明 SQL Server 2008 的安装和配置。

目前市面上常用的关系型 DBMS，主要有 Oracle，DB2，SQL Server，Access 和 MySQL。Oracle 由甲骨文公司发布，是目前市场占有率最高的 DBMS；DB2 是 IBM 的产品，一般在小型机服务器上使用。这两者都能管理大规模的数据库，性能优越。SQL Server 和 Access 是微软的产品，前者适合管理中等规模的数据库，后者适合管理 2 GB 以下的小型数据库。MySQL 一般和 Java 配合，适用于编写 BS 结构的数据库应用系统。它们支持的 SQL 语言虽然都遵循同一个标准，但语法和技术细节方面又有所不同，编写代码时可以参考其帮助系统查看具体语法。Oracle 的查询语言叫 PLSql（SQL Developer 自带），SQL Server 的叫 Transact SQL，它们都有在线帮助系统。

基于做课程设计、入门学习的目的，我们推荐大家使用 SQL Server。SQL Server 有