

信息科学与技术丛书

数据库设计与开发系列

Visual C++

数据库开发基础 及实例解析

郎 锐 编著

- ◎ 企业级信息管理系统开发
- ◎ OLE DB 客户数据库编程
- ◎ ODBC API、MFC ODBC/DAO、ADO 数据库编程
- ◎ 图书借阅管理系统开发实例
- ◎ 企业考勤信息管理系统开发实例
- ◎ 门诊信息管理系统开发实例
- ◎ 考试管理系统开发实例



机械工业出版社
CHINA MACHINE PRESS

信息科学与技术丛书
数据库设计与开发系列

Visual C++数据库开发基础 及实例解析

郎 锐 编著



机械工业出版社

本书从软件开发人员的角度考虑,以 Visual C++ 6.0 开发环境为背景,介绍了软件工程和信息管理系统的相关理论、各种常用的数据库开发技术和若干项目开发实例,使读者能够初步掌握企业级信息管理系统的规划、设计和实现的一般性过程。本书除了重点介绍 VC++ 的数据库开发技术之外,还介绍了信息管理系统、软件工程的相关理论,数据库编程密切相关的其他编程技术(如 CryptoAPI),以及数据的加密、解密处理等。

本书可供各大专院校计算机及其相近专业师生、IT 业的工程技术人员及编程爱好者参考使用。

图书在版编目(CIP)数据

Visual C++数据库开发基础及实例解析/郎锐编著. —北京:机械工业出版社,2005.1
(信息科学与技术丛书 数据库设计与开发系列)

ISBN 7-111-15036-8

I. V... II. 郎... III. C语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 080202 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚

责任编辑: 李馨馨

责任印制: 石 冉

北京中兴印刷有限公司印刷·新华书店北京发行所发行

2005 年 1 月第 1 版·第 1 次印刷

787mm×1092mm¹ 1/16·16.75 印张·409 千字

0001—5000 册

定价: 29.00 元(含 1CD)

凡购本图书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话(010) 68993821、88379646

封面无防伪标均为盗版

出版说明

随着信息科学技术的迅速发展，人类每时每刻都会面对层出不穷的新技术、新概念。毫无疑问，在节奏越来越快的工作和生活中，人们需要通过阅读和学习大量信息丰富、具备实践指导意义的图书，来获取新知识和新技能，从而不断提高自身素质，紧跟信息化时代发展的步伐。

众所周知，在计算机硬件方面，高性价比的解决方案和新型技术的应用一直备受青睐；在软件技术方面，随着计算机软件的规模和复杂性与日俱增，软件技术受到不断挑战，人们一直在为寻求更先进的软件技术而奋斗不止。目前，计算机在社会生活中日益普及，随着因特网延伸到人类世界的层层面面，掌握计算机网络技术和理论已成为大众的文化需求。也正是由于信息科学技术在电工、电子、通信、工业控制、智能建筑、工业产品设计与制造等专业领域中已经得到充分、广泛的应用，所以这些专业领域中的研究人员和工程技术人员将越来越迫切需要汲取自身领域信息化所带来的新理念和新方法。

针对人们对了解和掌握新知识、新技能的热切期待，以及由此促成的人们对语言简洁、内容充实、融合实践经验的图书迫切需要的现状，机械工业出版社适时推出了“信息科学与技术丛书”。这套丛书涉及计算机软件、硬件、网络、工程应用等内容，注重理论与实践相结合，内容实用，层次分明，语言流畅，是信息科学与技术领域专业人员不可或缺的图书。

现今，信息科学技术的发展可谓一日千里，机械工业出版社欢迎从事信息技术方面工作的科研人员、工程技术人员积极参与我们的工作，为推进我国的信息化建设作出贡献。

机械工业出版社

前 言

信息管理系统的数据库编程是目前非常热门的主流编程技术，广泛应用于工程、科研、金融和教育等领域。本书以 Microsoft Visual C++ 6.0 开发环境为背景，对 ODBC、DAO、OLE DB 以及 ADO 等常用的数据库访问技术作了详细介绍。在结构编排上，从基本原理和相关概念入手，逐步对各种编程技术的概况、基本实现流程和具体编程细节作了比较清晰的描述，并在最后通过几个具体实例进一步对上述编程技术进行综合应用。

虽然本书的核心是数据库编程技术，但考虑到信息管理系统设计的具体情况，也对软件工程中的基本知识，尤其是对项目管理作了重点介绍。在本书最后的几种行业信息管理系统的实例开发中，还对 CryptoAPI 数据加密、解密、进程间通信、口令密码保护、数据源的动态配置、通过 COM 技术实现与 MS Word 的通信以及 UDL 文件的使用等与数据库开发关系密切的一些实用技术的实现方法作了简要的阐述。

阅读本书的读者应具有基本的 C++ 编程能力和初步的 Microsoft Visual C++ 6.0 程序开发经验。本书配套光盘提供有各章节出现的所有程序源代码，这些示例代码在 Windows 2000 Professional 下由 Microsoft Visual C++ 6.0 调试、编译通过，具体运行环境请参阅说明文件。

在本书出版之际，笔者要特别感谢我的父母郎益青先生和张连霞女士还有山东大学云昌钦教授、鲁能积成电子系统实验所梁成辉硕士、中国电波传播研究所赵振维研究员、罗发根高工对笔者的支持、帮助与鼓励。书中不妥之处在所难免，诚请读者提出宝贵意见。

作 者

目 录

出版说明	2.2.5 软件风险管理	35
前言	2.3 本章小结	35
第 1 章 数据库开发基础	第 3 章 ODBC API 编程	36
1.1 信息管理系统概述	3.1 ODBC API 函数	36
1.1.1 信息管理系统	3.1.1 环境初始化函数	36
1.1.2 信息管理系统的开发	3.1.2 连接操作函数	38
1.1.3 必要的基础性工作	3.1.3 数据源操作函数	41
1.2 了解数据库	3.1.4 结果集处理函数	47
1.2.1 数据库和数据库管理系统	3.1.5 事务及批处理函数	49
1.2.2 数据库的基本组成	3.1.6 光标函数	50
1.2.3 数据库应用程序体系结构	3.1.7 表列操作函数	52
1.3 数据库开发技术	3.1.8 诊断函数	55
1.3.1 Microsoft Visual C++ 6.0	3.1.9 编目函数	56
开发环境	3.2 使用 ODBC API	57
1.3.2 Microsoft Visual C++ .NET	3.2.1 编程基本步骤	57
开发环境	3.2.2 环境句柄与连接句柄的分配	58
1.3.3 通用数据访问技术	3.2.3 连接、断开数据源	59
1.4 SQL 语法	3.2.4 执行 SQL 语句	61
1.4.1 结构化查询语言	3.2.5 处理结果集	64
1.4.2 SQL 从句	3.2.6 对结果集的编辑	66
1.4.3 SQL 合计函数	3.3 本章小结	68
1.4.4 SQL 命令	第 4 章 MFC ODBC/DAO 数据库	
1.5 本章小结	编程	69
第 2 章 企业级信息管理系统	4.1 ODBC 的 MFC 类	69
的开发	4.1.1 ODBC 的 MFC 编程方式	69
2.1 项目开发流程	4.1.2 CDatabase 类	69
2.1.1 设计目标与设计流程	4.1.3 CRecordset 类	71
2.1.2 软件需求分析	4.1.4 CRecordView 类	73
2.1.3 软件概要设计和详细设计	4.2 记录字段交换处理	74
2.1.4 软件编码、测试与维护	4.2.1 RFX/Bulk RFX 和 DFX 机制	74
2.2 企业级 MIS 系统开发中的	4.2.2 CFieldExchange 类	76
项目管理	4.3 MFC ODBC 数据库访问	77
2.2.1 项目管理概述	4.3.1 创建 MFC ODBC 数据库	
2.2.2 软件项目管理的内容	应用工程	77
2.2.3 软件配置管理	4.3.2 数据库显示	80
2.2.4 软件质量管理	4.3.3 编辑记录集	80

4.4 MFC DAO 数据库访问	82	6.3.1 连接对象	141
4.4.1 使用 DAO	82	6.3.2 命令对象	141
4.4.2 MFC DAO 类	82	6.3.3 记录集对象	142
4.4.3 DAO 应用程序开发	90	6.3.4 域对象	143
4.5 本章小结	96	6.3.5 参数对象	144
第 5 章 OLE DB 客户数据库编程 ..	97	6.3.6 属性对象	144
5.1 OLE DB 原理	97	6.3.7 错误对象	144
5.1.1 OLE DB 基础	97	6.4 建立 ADO 应用程序	145
5.1.2 OLE DB 结构	97	6.4.1 使用 ADO 编程	145
5.1.3 OLE DB 对象	98	6.4.2 建立连接、打开记录集	146
5.1.4 OLE DB 客户模板结构	100	6.4.3 操作、编辑数据	148
5.2 模板与 ATL 基础	104	6.5 本章小结	150
5.2.1 模板	104	第 7 章 图书借阅管理系统	
5.2.2 ATL 概述	105	开发实例	151
5.2.3 ATL COM 对象	106	7.1 需求分析与概要设计	151
5.3 开发 OLE DB 客户程序	107	7.1.1 需求分析	151
5.3.1 使用 MFC 应用程序向导	107	7.1.2 概要设计	151
5.3.2 使用 ATL 对象向导	111	7.2 详细设计	153
5.3.3 使用 ATL 应用程序向导	114	7.2.1 模块功能细化及界面设计	153
5.4 本章小结	117	7.2.2 数据库结构设计	155
第 6 章 ADO 数据库编程	118	7.2.3 程序总体框架设计	158
6.1 理解 ADO	118	7.3 建立程序框架	159
6.1.1 ADO 与 OLE DB	118	7.3.1 使用 MFC AppWizard(exe)	
6.1.2 了解 ADO 的结构体系	118	向导	159
6.1.3 了解 ADO 连接	119	7.3.2 窗体的切换	161
6.1.4 了解 ADO 记录集和字段	119	7.3.3 各窗体的界面实现	164
6.2 COM 编程基础	121	7.4 借书卡管理模块	165
6.2.1 了解 COM	121	7.4.1 新卡的注册	165
6.2.2 COM 对象	122	7.4.2 废卡的注销	167
6.2.3 COM 接口	123	7.5 书目信息管理模块	168
6.2.4 COM 组件	126	7.5.1 书目信息的添加	168
6.2.5 一些基于 COM 的技术	128	7.5.2 对书目信息的查询及其撤消 ..	170
6.2.6 使用代码组件	129	7.6 供应商信息管理模块	172
6.2.7 自动化对象与 IDispatch		7.7 图书借阅信息管理模块	173
接口	136	7.7.1 借阅信息的添加与显示	173
6.2.8 IDispatch 接口使用的数据		7.7.2 借阅信息的删除	178
类型	137	7.7.3 查询借阅信息	179
6.2.9 类型库	139	7.8 本章小结	182
6.3 ADO 对象模型	141	第 8 章 企业考勤信息管理系统	

开发实例	183
8.1 需求分析与概要设计	183
8.1.1 需求分析	183
8.1.2 概要设计	183
8.2 详细设计	184
8.2.1 模块功能细化及界面设计	184
8.2.2 数据库结构设计	187
8.2.3 程序总体框架设计	187
8.3 口令管理模块	189
8.3.1 使用 CryptoAPI	189
8.3.2 CryptoAPI 一般处理流程	190
8.3.3 初始化模块	191
8.3.4 明文口令加密模块	193
8.3.5 密文口令解密模块	194
8.3.6 防止口令在登录系统时 被窃取	195
8.3.7 用户登录及口令管理	196
8.4 动态配置 ODBC 数据源	197
8.5 考勤信息采集模块	198
8.5.1 子进程的创建与退出	198
8.5.2 进程间通信	201
8.6 员工信息管理模块	203
8.7 考勤记录管理模块	204
8.7.1 查看全部记录	204
8.7.2 记录缺考信息	206
8.7.3 考勤信息汇总	207
8.8 本章小结	209
第 9 章 门诊信息管理系统	
开发实例	210
9.1 需求分析与概要设计	210
9.1.1 需求分析	210
9.1.2 概要设计	210
9.2 详细设计	211
9.2.1 模块功能细化及界面设计	211

9.2.2 数据库结构设计	212
9.2.3 程序总体框架设计	212
9.3 表单管理模块	214
9.3.1 使用 DAO 访问数据源	214
9.3.2 记录集字段的动态绑定	215
9.3.3 药剂信息的录入	220
9.3.4 划价信息的录入	222
9.3.5 诊断信息和病人信息的录入	226
9.4 报表管理模块	227
9.4.1 COM 自动化技术	227
9.4.2 与 Word 模板的数据通信	227
9.4.3 处方打印	229
9.4.4 发票打印	232
9.5 本章小结	235
第 10 章 考试管理系统开发实例	236
10.1 需求分析与概要设计	236
10.1.1 需求分析	236
10.1.2 概要设计	236
10.2 详细设计	237
10.2.1 模块功能细化及界面设计	237
10.2.2 数据库结构设计	241
10.2.3 程序总体框架设计	242
10.3 考试成绩管理模块	243
10.3.1 通过 UDL 文件建立 ADO 连接	243
10.3.2 动态子窗体的实现	245
10.3.3 单科成绩录入	248
10.3.4 名次排序	250
10.3.5 考试成绩查询	253
10.3.6 考试成绩汇总	254
10.4 教师信息管理模块	255
10.4.1 教师信息的录入	255
10.4.2 教师信息的删除	256
10.5 本章小结	257

第1章 数据库开发基础

本章首先介绍了信息管理系统及其开发的一些基本概念，同时也对数据库的基本概念及其基本开发技术做了阐述。最后对数据库开发经常用到的 SQL 语句的基本语法作了较为详细的讲解。

1.1 信息管理系统概述

1.1.1 信息管理系统

信息管理系统 (Management Information Systems, MIS) 是一个不断发展的新型学科，通常指由人和计算机设备或其他信息处理技术组成并用于管理信息的系统。信息管理系统早已应用于各行各业，并随着计算机技术的发展和普及而以计算机信息管理系统逐渐取代了以往需要手工处理的各类信息管理系统。本书所讲的“信息管理系统”指的就是计算机信息管理系统。

信息是信息管理系统中的基本对象。所谓信息，指的是经过加工的、有一定利用价值的数据。不同行业的信息管理系统有着不同的信息获取手段，但是所有信息管理系统中待处理的信息均是来自第一手的原始资料，通常来源分散、数据量庞大，而且多有时间局限性。信息管理系统的重要作用之一，是对信息进行管理并在此基础之上为用户提供各种形式的帮助。对信息的管理通常由信息的采集、信息的传输、信息的储存、信息的加工、信息的维护和信息的使用五个方面组成。管理信息的各个组成部分如果按一定的规则相互联系、相互作用并形成具有一定功能的整体，就构成了信息管理系统。这个用来管理信息的系统一般由输入、处理、输出、反馈和控制五个基本要素组成。

一般来说，一个完整的信息管理系统应该由信源、信宿、信息处理、信息用户和信息管理者五个部分构成，并具有以下四个标准：

- 有明确的信息需求。
- 信息的可采集性与可加工性。
- 能够通过计算机软件为管理人员提供信息。
- 能够管理信息。

此外，信息管理系统还是一个交叉性非常强的综合性学科，仅基本组成部分就涉及计算机学科（网络通信技术、数据库技术、软件工程、计算机编程技术等）、数学（统计学、运筹学、线性规划等）、管理学、仿真学等诸多学科，对于涉及不同行业的信息管理系统还要具备相关专业的知识。随着科学技术的高速发展，信息管理系统所要涉及的范围还会不断扩大。

目前，开发高质量信息管理系统的能力普遍落后于计算机软硬件的进展速度，加上社会对信息管理系统发展和完善需求的增加以及对在信息管理系统开发过程中出现的错误认识和

行为而导致系统开发的失败，已极大妨碍了各行业的信息化改造。因此，对信息管理系统的开发进行深入的研究，提高研发成功率是非常必要的。

1.1.2 信息管理系统的开发

根据在众多大型商务信息管理系统的实际开发过程中总结出来的经验和教训，在进行信息管理系统的开发过程中一般应掌握以下原则：

- 创新原则
- 整体原则
- 发展原则
- 经济原则

其中，创新原则指的是在系统开发中使用先进的新技术，使其能够较现有的同类系统有较大的飞跃。整体原则要求系统在功能上必须覆盖该行业的各个方面，能够满足各种不同的需求。发展原则是指在系统开发过程中应具有适当的超前意识，以提高系统的利用率。经济原则是这几个原则中最务实的，考虑的是系统的实用性。在设计系统时应根据实际需求考虑应采取的规模和技术。

尽管如此，在具体的开发过程中，以上几个原则是很难完全贯彻执行的。例如，在开始设计时系统的最终目标及要包含的内容通常难以确定，并且涉及所要管理的内容、预期达到的目标以及运行状况等众多内容，很难仅靠调研确定出所有的内容。这也就决定了信息管理系统的开发不能象普通应用软件的开发一样，在开发前完全确立系统的目标和内容，并根据编写的详细设计说明书实施开发。这也是信息系统开发风险要大于普通软件项目的一个重要原因。

为降低开发风险，对信息管理系统的开发除了尽量遵循上述几个原则外，还要尽可能从实际需求出发，确立适度的开发目标和内容，使企业的决策者和系统的开发者都能够对系统的开发做到心中有数，以有效地安排和部署开发工作，并在开发过程中逐步确立新的工作方式。对于一个应用于某个企业或某个行业的信息管理系统，其规模是比较大的，一般很难从整体把握，可将其分解为多个子系统，从而降低系统的复杂性。但一定要制定好描述各子系统间联系的协议规范，以保证各子系统最终能够形成一个统一的系统。对于各子系统，要全面评估系统开发的目的、内容、效果以及运行后的维护扩展等因素。这些因素关系到系统建设的成败，应从实际需要和可能出发，要从大局着眼进行认真的分析。盲目求大求全，通常会导致项目的失败。最后，在满足协议规范的前提下，用快速原型法开发各子系统。这样就不必在开始就进行详尽的需求分析，而是通过试运行把用户的需求不断吸收进去，让系统直接逼近开发目标。这样既可以降低开发者掌握原有系统的难度，又能够准确地了解用户的需求、确立新的工作模式。

信息管理系统开发的涉及面较大，开发者应具有全局把握和控制开发过程的能力。除了必要的软件设计工作外，开发过程的组织和控制也是开发者所不能忽视的。开发过程中编写的大量程序一次性投入试运行或对开发工作中的问题和困难估计不足，都将导致因开发规模展开过大而失控。而且工作量和难点的过度集中也会妨碍系统建设。通过对系统的分解，根据工作量、用户情况及其在系统中的作用等因素排列开发顺序，并根据子系统的开发情况随时调整和部署开发工作，使系统开发能有一个由小到大的、平稳的发展过程，才能比较顺利地形成完整的信息管理系统。在开发过程中，应尽量避免处于局部利益的考虑而影响系统

的形成。例如，多数用户会在完成了一个功能以后又从孤立的角度临时提出一些实际应该系统考虑的新要求，甚至将系统开发人员简单地当作程序员来使用，安排其马上添加完成这些临时提出的功能。开发者如果不够冷静，很容易陷入到具体的问题中去，头痛医头，脚痛医脚，从而会逐渐失去对开发过程的整体控制。在这种情况下的信息管理系统开发多半是不会成功的。因此，开发者应善于了解和把握系统性、本质性的问题，始终拥有一个明确的系统概念，同时也要采取一些有效的技术和组织措施，确保在开发工作中能够处于主动的位置，以便有效地贯彻系统建设的思想，把系统建设推向成功。

用户的积极参与也是确保系统开发成功的重要因素。有的开发者只着眼于计算机技术本身，没有与用户进行很好的沟通，而是按自己的构想做出了大量程序，这样的系统即使开发出来了用户也会因使用不便而放弃。及时投入试运行是吸引用户参与开发工作并为之进行具体交流的有效途径。为此，子系统的分解要便于开发和试运行。在诸多子系统中，应注意选取那些使用效果明显又易于开发的子系统首先开发，其运行效果会增强用户对系统建设的信心和对开发者的支持。整个系统的形成需要有一个渐进的过程，不可操之过急，而子系统的开发则必须迅速完成，才能确保整体的从容部署，这是系统建设的节奏。如果长时间不能过渡到新的工作方式上去，用户会消极应付，同时也给程序调整带来困难。

1.1.3 必要的基础性工作

信息管理系统的应用大大降低了管理人员的工作强度，使高层管理人员对中层、中层对基层、基层对设备系统逐级加强了管理、监督和控制，业务工作量也因此而减少，但是系统的维护工作量也会因此而增大，大量的业务改进工作将转向系统的改进，为确保系统的正常运转需要选拔、培养出合格人员特别是各层次计算机系统管理员，这也是企业信息管理系统建设的基础。

各层次计算机系统管理员仅具备单纯的计算机知识是远远不够的。分管计算机应用的人员不仅要及时发现、解决计算机本身的问题，更重要的是能够提出适合的系统开发应用规划，并以此作为企业领导对本信息管理系统建设乃至企业改革的决策依据。国内外众多案例的实际经验表明，企业信息管理系统建设的关键是对系统管理人员的选拔与培养。

制定一系列的标准和规范是系统建设的另一个基础性工作。系统的建设与运营涉及数据共享、计算机软硬件、工作模式等多个方面，必须制定统一的规则或约定，以此作为系统各部分联系的准则，使系统能够成为一个有机的整体。这主要包含信息规范、计算机软硬件平台规范和维护管理模式规范三个方面。信息规范指的是代码、事物特性表等所要遵循的标准；计算机软硬件平台规范主要包括计算机硬件、操作系统、数据库、网络通信以及数据处理等内容；维护管理模式规范则对系统运行规程、岗位设置、计算机技术人员的上岗标准等做了约定。上述标准的制定将有助于系统的可靠稳定运行。

1.2 了解数据库

1.2.1 数据库和数据库管理系统

数据库是关于某一特定主题或目标的信息集合。通常将其理解为一个用于存储和操纵数

据全部元素的容器。与文件不同,存放在数据库中的数据是高度结构化的,数据库不仅要考虑记录内数据项之间的联系,还要考虑记录间的联系。这种联系通过数据模型加以描述,主要有层次数据库模型、网状数据库模型、关系数据库模型和实体联系数据库模型等几种。关系数据库模型是应用最为广泛的。采用关系数据库模型的数据库管理系统即为关系型数据库系统。如果一个数据库管理系统是根据自己定义的关系来存储、处理和管理数据库信息的,那么这种数据库管理系统就是一个关系型数据库系统。一个关系数据库可看作是由多个平面数据库连接到一起的组合。在关系数据库中,不仅可以进行平面数据库中的一些操作,还可以在记录集之间建立关系,使用户数据库存储数据的效率更高。例如,用户可以把数据分门别类地进行组织,从而方便地对数据进行跟踪和校验,也可以将不同类型数据之间的联系存储起来,用于产生综合性数据。

如果不采取关系数据库,在管理信息时就要建立一个相当大的表以存储数据。例如,在未采用关系数据库模型的订单管理数据库中,需要用到产品名称、产品资料、供货商资料等信息,每添加一笔订单记录就要同时输入上述信息,如果有多种产品来自同一供货商,那么该供货商的资料将重复出现在表单中。这不仅浪费存储空间,也容易造成信息维护的错误。例如,如果某供应商的资料变更,就需要在所有出现该供应商的位置更改其资料,不仅工作量大,而且容易出现录入错误、遗漏等情况。关系数据库则通过对数据的分类而用多个较小的表来分别存储每一种类型的数据,并通过存储的不同类型数据的联系而产生综合性信息。这样处理后,每条信息只在一处存放,不仅减少数据重复,节约存储空间,而且在供应商资料变更后只需在供应商表单中更改该供应商的资料即可。所有在其他表单出现该供应商的地方将自动更新其资料。

对于一个数据库,在其管理的内容越来越复杂时,信息的收集、存储以及信息之间联系的保持就比较困难了。这就需要有一个数据库管理系统(database management system, DBMS)对数据库信息进行存储、处理和管理。在数据库应用系统中,数据库应用程序从数据库管理系统获取并操作数据,数据库管理系统中的数据管理器负责对数据库的操作。数据库管理系统所具备的功能与具体使用的系统有关,但通常应具备描述、管理、查询、操作以及对数据库的建立与维护等功能。

1.2.2 数据库的基本组成

一个数据库主要由表、窗体、查询和报表等基本对象构成。通过这些对象将可以完成对信息的收集、存储和处理。

表是用户自定义的用于存储数据的结构体,是数据库的基础。数据库里的每一个记录集都是一个表,包含了某一特定主题的信息,由多个字段和记录组成。字段描述了同一实例的不同方面的信息,而每一个实例的所有字段又构成了一条记录。在关系数据库中可以通过在表中定义关键字而在相关表之间建立关联。同一数据库中的表名和在同一表中的字段名都必须是惟一的。表单可以一次查看多条记录,而窗体则是通过窗体界面显示某一条记录的详细信息。窗体要具有良好的人机界面,通过窗体可以方便地录入、修改和显示记录信息。表中存储的是全部的字段和记录,如果需要查看其中一部分关心的信息就要通过查询来进行筛选,查询定义了用户想要操作的一组记录,并检索已输入到表中的符合条件的信息。在进行查询时,要查询的数据记录可以来自多个表,查询结果通常仍以表的形式给出。这些由查询

所确定的记录也被称为动态记录集，不同查询所产生的动态记录集中的记录是不一样的。在修改动态记录集中的某条数据时，其基表中的相应数据也会被修改。与前面几种数据库对象不同，报表的作用主要是显示信息，是为计算、打印和汇总数据而设计的一种数据库对象。虽然表和窗体也能够完成信息的显示和打印，但是报表的功能更为强大，能够对数据进行排序和分组，同时还能给出该组记录的各种统计数据。

1.2.3 数据库应用程序体系结构

一个数据库应用程序在逻辑上通常由数据访问链路和用户界面两部分组成。在设计数据库应用程序时，通常要把数据访问链路构件和用户界面构件分开，数据访问构件均放置在数据模块上。数据模块主要用来实现数据访问链路，也就是引入数据集。用户界面与数据访问链路分离的设计不仅符合模块化程序设计思想，也能够保证应用程序具有一致的用户界面，而且在向更高层次体系结构过渡时，只需修改数据模块的数据集构件，无须更改用户界面。

数据库应用程序的体系结构，取决于使用本地数据库还是远程数据库，取决于同时访问数据库的用户数以及数据库中需要存储信息的类型。如果数据库中的信息不需要在几个用户之间共享，建议使用本地数据库，可以获得较快的访问速度，而且不必购买昂贵的服务器。数据库应用程序可分为单层、双层和多层次的体系结构。在单层数据库应用程序中，应用程序通过数据库引擎从本地数据库获取数据并向用户界面提供数据支持，其体系结构模型如图 1-1 所示。

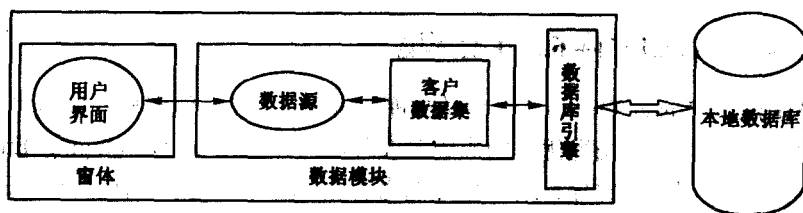


图 1-1 数据库应用程序单层结构体系模型

双层结构体系模型如图 1-2 所示，客户程序提供用户界面，通过数据库引擎从远程数据库服务器获取数据。在这种体系模式下，所有应用程序都被看作是客户。

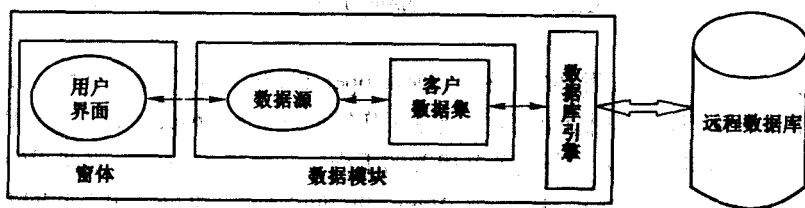


图 1-2 数据库应用程序双层结构体系模型

对于图 1-3 所示的多层次结构体系模型，客户程序、应用服务器和远程服务器分别在不同的机器上。其中，客户程序主要提供用户界面，能够向应用服务器请求数据和申请更新数据。再由应用服务器向远程数据库服务器转发请求。与双层结构相比，多层结构的应

用程序多了一个用于集中处理应用逻辑的中间层。中间层的存在使不同用途的客户程序能够使用相同的数据并且保证数据逻辑的一致。而且，客户程序可以做得非常简单，使之更容易安装、配置和维护。但是数据库应用程序所采用的结构层次越多，其开发难度和费用也就越大，因此，开发数据库应用程序时，最好从单层入手，随着数据和用户的增加，再逐步平滑地过渡到多层体系结构。即使采用的是多层结构的程序，在调试时也可以暂时以本地数据库模拟远程数据库提供数据，在全部调试成功后通过数据库引擎建立到远程数据库的连接即可。

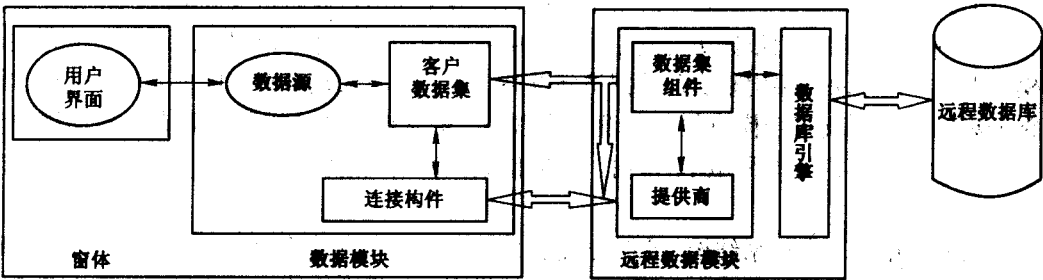


图 1-3 数据库应用程序多层结构体系模型

1.3 数据库开发技术

1.3.1 Microsoft Visual C++ 6.0 开发环境

本书以 Microsoft Visual C++ 为工具对数据库进行开发。同其他编程环境相比，Visual C++ 的效率是比较高的，提供了相当优秀的集成开发环境（Integrated Developing Environment，IDE），集代码编辑、调试、向导、编译和可视化资源编辑等功能于一体，其 MFC 基本类库对 Windows API 函数做了非常好的封装并拓展了功能，可以满足大多数的基本功能需求，程序设计人员只需简单地调用 MFC 类封装的功能函数即可。Visual C++ 对于为 Windows 和万维网(WWW)创建性能最高的应用程序来说是一种效率最高的 C++ 工具。几乎所有世界级的软件(从占主导地位的 Web 浏览器到任务关键的企业应用程序)都是使用 Microsoft Visual C++ 开发系统生成的。Visual C++ 也因其超强的功能而享有“Windows 环境下的外科手术刀”的美誉。

Visual C++ 6.0 的集成开发环境被称作 Developer Studio，从中可以创建工程、源文件、各种资源以及其他文档。对项目的组织是通过工作空间（Workspace）和工程（Project）来完成的，一个工作空间可以包含一个或多个相关的工程。在操作界面上，通过 Workspace 浮动窗口上的三个选项卡 ClassView、ResourceView 和 FileView，可分别用不同的方式来管理、操作工作空间中的各个工程及其内部文档和资源（参见图 1-4）。在用 Visual C++ 6.0 集成开发环境去创建一个新的工作空间时，可通过 File 菜单下的 New 菜单项来弹出如图 1-5 所示的 New 对话框，在 Project 选项卡中可以创建新的工程。Project 选项卡所列举的各个工程项目是针对不同目的的项目开发而设置的，在使用时应根据需求灵活选择，创建相应的工程项目。

下面是对这些项目功能的简要介绍:

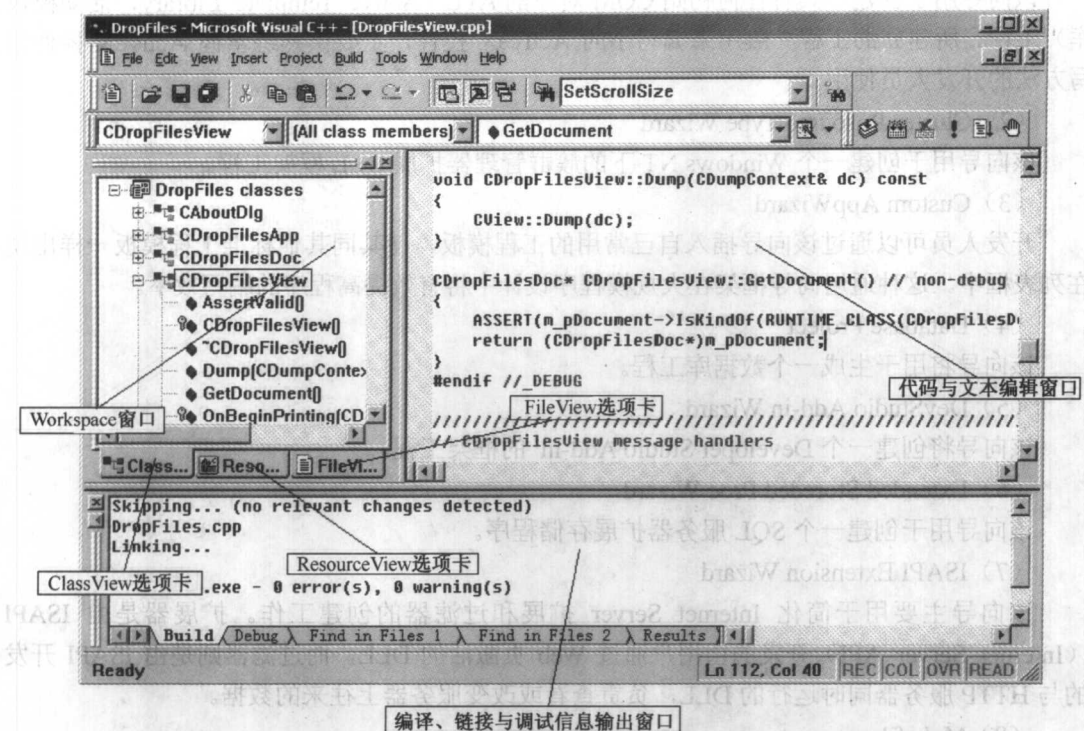


图 1-4 Visual C++ 6.0 的集成开发环境

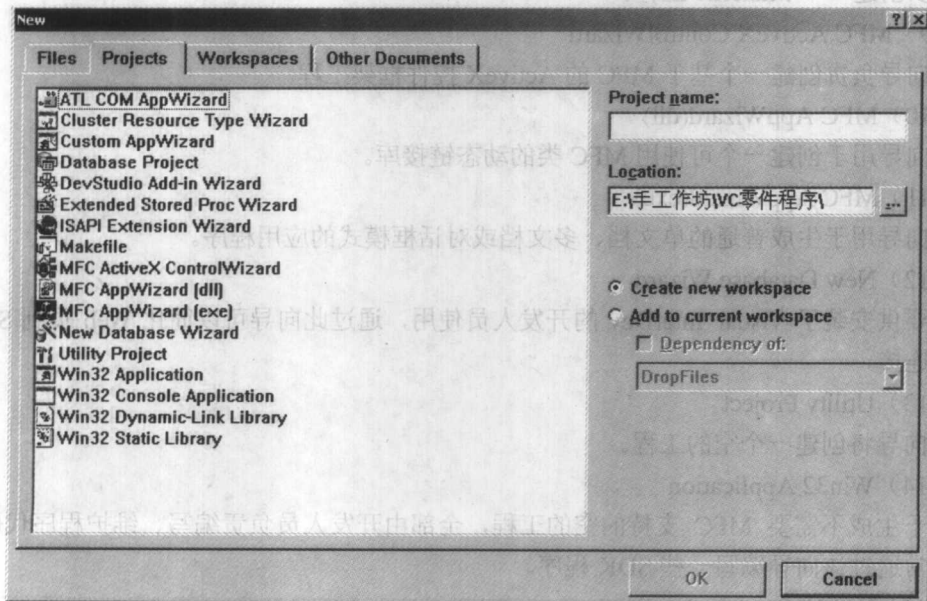


图 1-5 Visual C++ 6.0 的工程向导

(1) ATL COM AppWizard

该向导用于创建不具有任何初始 COM 对象的 ATL (Active Template Library, 活动模板库) 工程。所建立的工程一般用来编写小的 ActiveX 控件, 主要供熟练掌握 ActiveX 控件编写方法的开发人员使用。

(2) Cluster Resource Type Wizard

该向导用于创建一个 Windows NT 下的集群管理器扩展 DLL 框架工程。

(3) Custom AppWizard

开发人员可以通过该向导插入自己常用的工程模板, 使其同其他标准工程模板一样出现在列表框中。这种通用向导框架在大规模程序设计中将有效提高程序的编写效率。

(4) Database Project

该向导将用于生成一个数据库工程。

(5) DevStudio Add-in Wizard

该向导将创建一个 Developer Studio Add-in 的框架工程。

(6) Extended Stored Proc Wizard

该向导用于创建一个 SQL 服务器扩展存储程序。

(7) ISAPI Extension Wizard

该向导主要用于简化 Internet Server 扩展和过滤器的创建工作。扩展器是由 ISAPI (Internet Server API) 开发的由用户通过 Web 页激活的 DLL, 而过滤器则是由 ISAPI 开发的与 HTTP 服务器同时运行的 DLL, 负责查看或改变服务器上往来的数据。

(8) Makefile

该向导主要供那些用可独立应用的工具来取代一部分 Developer Studio 的开发人员使用, 用以创建一个 Makefile 工程。

(9) MFC ActiveX ControlWizard

该向导负责创建一个基于 MFC 的 ActiveX 控件框架工程。

(10) MFC AppWizard(dll)

该向导用于创建一个可使用 MFC 类的动态链接库。

(11) MFC AppWizard(exe)

该向导用于生成普通的单文档、多文档或对话框模式的应用程序。

(12) New Database Wizard

主要供安装了 Visual InterDev 的开发人员使用, 通过此向导可以简化 Web 页到 SQL 数据库的连接。

(13) Utility Project

该向导将创建一个空的工程。

(14) Win32 Application

用于生成不需要 MFC 支持的空的工程, 全部由开发人员负责编写、维护程序代码和资源, 通常通过该向导编写一些 SDK 程序。

(15) Win32 Console Application

该向导用于生成一个简单的控制台应用程序工程。

(16) Win32 Dynamic-Link Library

该向导将创建一个不需要模板和 MFC 支持的空的动态链接库工程。

(17) Win32 Static Library

该向导用于创建静态库工程。

1.3.2 Microsoft Visual C++ .NET 开发环境

Microsoft Visual C++ .NET 是 Microsoft Visual C++ 系列的最新版本（目前包括 2002 和 2003 两个版本），Visual C++ .NET 的出现大大提高了 C++ 的生产效率，没有降低其灵活性、性能和可控制性。以往在 Visual C++ 4.x、5.0 和 6.0 等早期 32 位版本中生成的项目，可以在当前版本中作为项目打开和保存，即将旧版本项目升级到 .NET 版本。其中，可能希望或者需要对新项目中的以下内容进行修改，以确保项目能够顺利编译通过：

- 修改与 WINVER 相关的编译错误；
- 使用标准 C++ iostream 库而不是使用以前的 iostream 库版本；
- 使用当前 MFC 库。

相对于旧版本，Microsoft Visual C++ .NET 新增和改进了许多功能，主要表现在托管代码和面向 .NET 框架、属性化编程、开发环境以及库（包括活动模板库 ATL、ATL Server、C 运行时库、Microsoft 基础类库、OLE DB 模板、共享类库和标准 C++ 库）等。这里所说的属性化编程特性旨在提供一种有效而快捷的方法来简化使用 Visual C++ 的 COM 编程。属性与 C++ 关键字一样，在源文件中使用，并由编译器进行解释。属性可以修改现有代码的行为，甚至可以插入附加框架代码来完成基本任务，例如实现 ActiveX 控件、创建类工厂或者设置数据库命令的格式。属性几乎可以应用到任何 C++ 对象（如类、数据成员以及成员函数）上，还可以作为独立的语句插入到源代码中。

Visual C++ .NET 的集成开发环境如图 1-6 所示。通过选项卡式文档（此功能自动在 IDE 内以选项卡的方式排列文档窗口）、自动隐藏（允许用户沿 IDE 的边缘最小化工具窗口）和“向后定位”和“向前定位”（允许用户在打开的窗口间定位，并在文件内的所选内容和光标历史记录中定位）等技术的使用可以更容易地在屏幕上同时查看更多的代码。新增加的属性窗口提供了一个可自定义的网格，用于查看和修改项目中对象的属性。工具箱则可以显示 Visual Studio 项目中使用的各个不同的项。根据当前正在使用的设计器或编辑器，“工具箱”中可用的选项卡和项会有所变化。可用的项包括 .NET 组件、COM 组件、HTML 对象、代码片段和文本。至于存在于 Visual C++ 早期版本中的类向导 ClassWizard 和向导栏 WizardBar 则已经从 Visual C++ .NET 中移除。如果需要在 Visual C++ .NET 中使用原有的类向导和向导栏功能，可以通过以下方法来实现：

1. 添加新类（向导栏）

在“类视图”或“解决方案资源管理器”中右击项目节点，在快捷菜单上单击“添加”，然后单击“添加类”。

2. 向类添加函数（向导栏）

在“类视图”中右击类，在快捷菜单上单击“添加”，然后单击“添加函数”。