

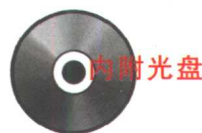
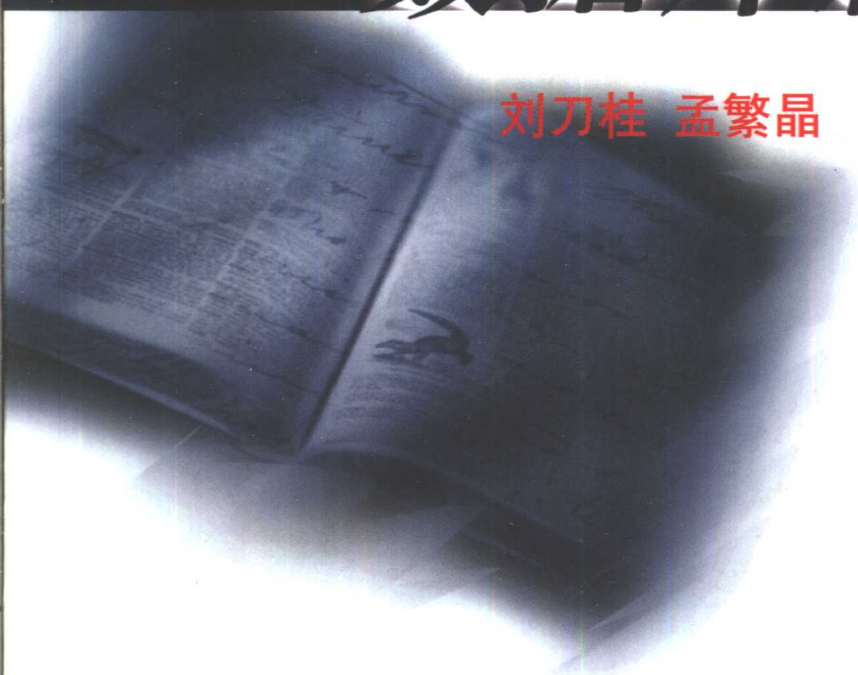


Visual C++

实践与提高

数据库篇

刘刀桂 孟繁晶 编著



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

Visual C++ 实践与提高

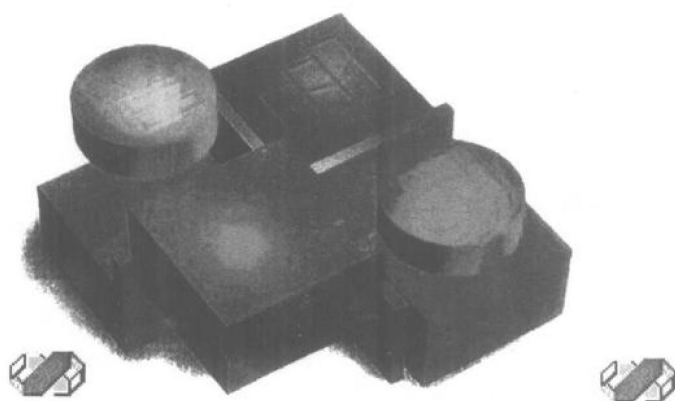
——数据库篇

刘刀桂

编著

孟繁晶

 **Visual C++**



中国铁道出版社

2001 • 北京

(京)新登字 063 号

内 容 简 介

本书是主要介绍如何使用 VC++ 进行数据库开发的各种接口和技术。内容详实全面,重点突出,主次分明。它的最大特色之一为工程应用而写。书中结合实例对读者可能在项目开发中遇到的实际问题进行了详尽细致的剖析;同时,在介绍具体技术时,又力求阐明一种解决问题的思路。特别是书中提供了很多有实用价值的源码,相信对读者开发会大有裨益。

本书适合哪些有一定 VC++ 基础又需要进行数据库工程应用的读者,也可以作为自学或教学选用的参考书。

图书在版编目(CIP)数据

Visual C++ 实践与提高. 数据库篇/刘刀桂, 孟繁晶编著. —北京: 中国铁道出版社, 2001. 3

ISBN 7-113-04083-7

I. V… II. ①刘…②孟… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2001)第 05280 号

书 名: Visual C++ 实践与提高——数据库篇
作 者: 刘刀桂 孟繁晶
出版发行: 中国铁道出版社(100054, 北京市宣武区右安门西街 8 号)
策划编辑: 严晓舟
特邀编辑: 王占清
封面设计: 冯龙彬
印 刷: 北京市兴顺印刷厂
开 本: 787×1092 1/16 印张: 34.75 字数: 844 千
版 本: 2001 年 3 月第 1 版 2001 年 6 月第 2 次印刷
印 数: 5001~10000 册
书 号: ISBN 7-113-04083-7/TP·516
定 价: 56.00 元

MS333/02

版权所有 盗印必究

凡购买铁道版的图书, 如有缺页、倒页、脱页者, 请与本社计算机图书批销部调换。

前言

作为一种功能强大和应用广泛的编程语言，Visual C++（以下简称 VC++）的地位不可动摇。作为一种十分流行且性能稳定的软件开发平台，VC++的应用范围越来越广泛。基于以上两点，从各个方面介绍 VC++开发技术的书简直俯拾皆是，其中介绍 VC++数据库开发技术的书也是如此。

特别是在 VC++数据库开发技术方面，很少能找到这样分量的书——它直接切入工程应用需要并注重解决实际问题中遇到的问题。其实，这也是由数据库应用本身的特点所决定的，因为数据库总是与各种工程问题结合在一起使用。所以，在这些方面作些努力并把我们的经验教训与大家一起分享，是一件很有意义的事情。

但并不是说本书不适合于通常读者使用，因为问题是从实际应用中来的，学会解决这些问题并开发出性能更加优良的程序才是学习的根本目的所在。

本书结构

本书是讨论 VC++数据库各种接口开发技术的工程应用参考书。作者在安排篇章结构时主要从如下两个方面来考虑。

其一，是要在深入与全面间找到平衡。这是一对矛盾，要做到深入就不可能对每个知识点都介绍得非常详细。所以在阐述各个编程接口技术时，对于那些读者一看就知道或者是在其他地方很容易见到的知识，或不着笔墨，或较少提及。

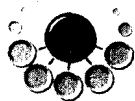
其二，是在介绍具体技术时，力求阐明一种解决问题的思路。这也是本书的特色之一。

根据以上两点，本书内容可分为以下几个部分：

- 关于 VC++数据库编程的基本概念
- 直接使用 ODBC API
- 使用 MFC ODBC 类
- 使用 MFC DAO 技术
- 使用 OLE DB 接口技术
- 使用 ADO 开发数据库技术
- 规划与构建数据库应用系统

另外，在附录部分，还对上述各接口以简明扼要的方式介绍了它们的主要特点和概念。这样做的目的是为了突出重点，特别是哪些对数据库编程接口技术还不是很了解的读者会非常有用。

在介绍各个接口技术时，我们采取“知识点各个击破，整体应用实例剖析”的策略。循序渐进，并力求深入浅出，尽量以最易于理解和接受的方式把有关知识呈现在读者面前。希望会对读者学习带来一定的益处。



如何使用本书

本书是介绍 VC++数据库编程技术比较全面和使用的书。适合于有一定 VC++基础知识但在数据库方面比较薄弱的读者，同时也适合于开发数据库应用的中高级读者。

对于数据库方面比较薄弱的读者，在具体学习各个接口技术之前建议先参考附录 B。附录 B 是关于 VC++数据库编程技术“快速成型”内容。

对于数据库方面有一定基础的读者，可以参看需要用到技术对应的章节。如果读者想了解数据库应用系统的整个过程的话，可以参看第 7 章有关“规划与构建数据库应用系统”以及附录 A 和附录 B 等。

本书中所介绍的各个接口（如 MFC ODBC 和 OLE DB 等）基本上都自成体系，读者不一定要自始至终逐章阅读，完全可以采取跳跃方式阅读。

体例说明

为了使表达方式一致和全书风格统一，我们在本书中对个人编程经验；或者需要提醒读者的注意事项；抑或是在碰到某个问题时要揭示一些关键技术点情况时，采用一些规范的小标题来突出显示。

注意

表示在某个事件或某项操作的进行当中，需要特别提醒读者注意之处。



说文解字

解释或注释某个专有名词或专门术语。



技术点睛

对某个技术点作比较深入全面的阐述，有时对涉及的相关知识也会作简单的解释或说明。总之，是对技术解释的“点睛”之笔。

● 技术精粹：

这是某些章节中的小节标题。在介绍各个具体的编程接口时，有些重要而且比较深入一些的知识点，不便于归入某个小节；或者为了突出重点，没有归入对应章节中，采用此统一标题“纳川归海”。

另外，本书中采用的有些术语，可能与读者在其他参考书中看到的相关术语名称不太一致，这没有谁对谁错的问题。有两个方面的因素，其一这些词基本都是从计算机英语中翻译过来，可能对应译法目前还没有统一。其二是，可能与作者个人的使用习惯等也有一定关系。

例如，在有些书中，把数据库的结构定义和说明信息叫做“纲要”，但我们在这里则称之为“模式”。其实，它们对应的计算机词汇都是同一个词——“Schema”。



这样的情况，本书中有如下这些。

计算机英语	本书称呼	其他可能称呼
Field	列、字段	域
Key	键	关键字
Schema	模式	纲要

关于作者

虽然本书署名作者只有两个，但它完全是集体智慧的共同努力的结晶。此外，邵向前、狄洪超、李彬、陈甬、杜庆柱、付惟龙、李厚福、韦文斌、王正盛、张宁、李隆春、孔莉秦、赵有珍、段海波、赵刚、吴斌、王丽英、毕红利、曾波、邓镐胜、吴移贵、邢蕾、刘艳、李泽、鲍小琳、董海军、孟凯祥、赵庚、董章振等参加了校对和排录等工作。在此对他们的辛勤劳动表示感谢！

在学习过程中，对本书内容有疑问的读者可通过如下方式与作者联系。

E-Mail: ldgsunday@sina.com 或者 yunwinner@21cn.edu.cn

由于时间仓促，加之作者水平有限，书中错漏之处在所难免，欢迎广大读者批评指正！

作者 谨识

2001年1月 雪夜 于北京

目 录

第 1 章 VC++ 数据库开发概述.....	1
1.1 VC++ 开发数据库应用程序的特点和优势	1
1.2 常用开发技术的比较	2
1.3 本书技术思路	3
第 2 章 使用 ODBC API	7
2.1 ODBC 概述	7
2.1.1 ODBC 体系结构	7
2.1.2 ODBC API 应用程序要素	8
2.1.3 句柄	9
2.1.4 ODBC 数据类型	10
2.1.5 ODBC API 函数汇总	13
2.1.6 使用 ODBC API 开发应用程序步骤	15
2.2 配置 ODBC 环境	15
2.2.1 环境句柄	16
2.2.2 分配环境句柄	16
2.2.3 设置环境属性	17
2.2.4 获得环境属性	17
2.3 分配连接句柄	18
2.3.1 分配连接句柄	18
2.3.2 设置连接属性	18
2.4 连接 ODBC 数据源	19
2.4.1 创建 ODBC 数据源	19
2.4.2 建立与数据源的连接	22
2.5 构造和执行 SQL 语句	25
2.5.1 分配语句句柄	25
2.5.2 设置语句属性	26
2.5.3 执行语句	26
2.5.4 使用编目函数	27
2.5.5 释放语句句柄	33
2.6 获得执行结果	33
2.6.1 获得结果集信息	34
2.6.2 获得结果集记录信息	35
2.7 断开并释放数据源连接	36



2.7.1 断开数据源连接	36
2.7.2 释放连接句柄	36
2.8 释放 ODBC 环境	36
2.9 技术精粹	37
2.9.1 深刻理解并使用 ODBC 句柄	37
2.10 ODBC API 实例剖析	37
2.10.1 实现目标	37
2.10.2 技术要点	38
2.10.3 创建过程	38
2.10.4 归纳总结	97
2.11 本章总结	98
第 3 章 使用 MFC ODBC	99
3.1 数据源	99
3.1.1 ODBC 数据源的创建	100
3.1.2 连接数据源	101
3.1.3 数据源的动态加载	103
3.2 记录集	104
3.2.1 用 Wizard 建立从视表到记录集的映射	104
3.2.2 选择和操纵记录	108
3.2.3 一记录集对应多表	109
3.2.4 统计记录数	110
3.3 RFX	111
3.3.1 理解 Wizard 产生的代码	111
3.3.2 RFX 函数	113
3.3.3 CFieldExchange	114
3.4 事务处理	115
3.5 多行存取	116
3.6 技术精粹	118
3.6.1 为 MFC 应用程序添加 ODBC 功能	118
3.6.2 过滤和排序以及参数化查询等	119
3.6.3 利用 SQL 语句删除记录	120
3.6.4 存储过程	121
3.6.5 绑定记录集	123
3.6.6 如何绕过 ODBC 口令提问	127
3.7 实例剖析	128
3.7.1 实现目标	128
3.7.2 技术要点	129



3.7.3 创建过程	129
3.7.4 归纳总结	166
3.8 本章总结	166
第 4 章 使用 MFC DAO	167
4.1 数据源: MFC DAO 与数据库	167
4.1.1 访问 Access 数据库	168
4.1.2 访问 ISAM 数据库	169
4.1.3 访问 ODBC 数据源	170
4.2 操作 DAO 对象	171
4.2.1 DAO 异常处理	171
4.2.2 数据库对象	172
4.2.3 表定义对象	173
4.2.4 记录集对象	177
4.3 DFX 机制	182
4.3.1 理解 Wizard 产生的代码	183
4.3.2 DFX 函数	184
4.3.3 CDaoFieldExchange	185
4.4 事务处理	186
4.4.1 DAO 工作区对象	186
4.4.2 事务处理	186
4.5 DAO 函数直接调用	188
4.5.1 什么情况下直接调用	189
4.5.2 调用接口: 关于身份验证	189
4.6 动态检索数据库	190
4.6.1 动态检索库结构	190
4.6.2 关于 CCrack 类	193
4.7 使用外部数据源	202
4.7.1 附加表	202
4.7.2 直接打开数据源	203
4.8 实例剖析	203
4.8.1 实例目标	203
4.8.2 技术要点	204
4.8.3 实现步骤	204
4.8.4 归纳总结	264
4.9 本章总结	265
第 5 章 使用 OLE DB	267
5.1 概述	267



5.2	预备知识	268
5.2.1	COM 简介	268
5.2.2	OLE DB 组件	270
5.2.3	OLE DB 应用程序基本结构	275
5.3	OLE DB 模板类	278
5.3.1	模板的定义和使用	278
5.3.2	OLE DB 模板结构	279
5.4	数据源	281
5.4.1	直接连接关系数据源	282
5.4.2	间接连接非关系数据源	287
5.4.3	动态加载数据源	289
5.5	记录集	292
5.5.1	存取器和绑定	293
5.5.2	命令类	297
5.5.3	用户记录	300
5.6	技术精粹	300
5.6.1	扩展的 OLE DB 异常处理	300
5.6.2	如何对 MFC 工程添加 OLE DB 支持	302
5.7	OLE DB 应用程序	303
5.7.1	概述	303
5.7.2	直接调用 OLE DB 对象与接口编程	303
5.7.3	OLE DB 数据提供程序	315
5.7.4	OLE DB 数据使用程序	335
5.7.5	OLE DB 综合实例	361
5.8	本章总结	413
第 6 章 使用 ADO		415
6.1	ADO 概述	415
6.2	ADO 结构和对象	416
6.2.1	ADO 应用的结构	416
6.2.2	ADO 对象	416
6.3	到数据源的连接	419
6.3.1	打开连接	420
6.3.2	设置连接超时属性	420
6.3.3	连接访问模式	421
6.3.4	关闭连接	421
6.4	使用记录集	421
6.4.1	移动记录指针	423



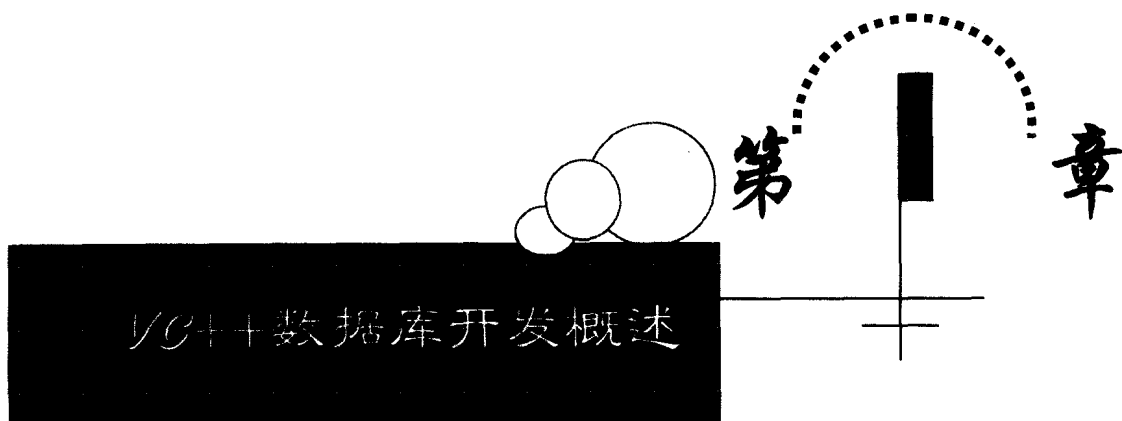
6.4.2 添加记录	424
6.4.3 更新记录	424
6.4.4 删除记录	425
6.5 执行命令	426
6.6 事务处理	426
6.7 ADO 事件	428
6.7.1 ADO 事件概述	428
6.7.2 ADO 事件类型及其参数	429
6.7.3 事件处理程序如何共同工作	430
6.7.4 VC++ ADO 事件实例	432
6.7.5 ADO 事件小结	434
6.8 技术精粹	434
6.8.1 关于#import	434
6.8.2 添加对 ADO 2.0 VC++接口的支持	435
6.8.3 关于记录集对象的特性支持	435
6.9 应用实例剖析	436
6.9.1 ADO VC++ Extensions 综合示例	436
6.9.2 使用 ActiveX 控件开发应用程序	440
6.9.3 使用 ADO 对象开发应用程序	446
6.10 本章总结	502
第 7 章 规划与构建数据库应用系统	503
7.1 规划与建立高性能后台数据库	503
7.1.1 关系数据库设计基础	503
7.1.2 规划数据库	504
7.1.3 使用 E-R 法建立数据库概念模型	504
7.1.4 概念模型转化为关系数据库模型	506
7.1.5 关系数据模型的规范化与优化	506
7.2 选择与优化数据库访问	510
7.2.1 查询方案的选择与优化	510
7.2.2 优化数据库访问	510
7.3 前端数据库应用程序开发	512
7.4 系统测试与试运行	512
7.5 系统运行及维护	513
7.6 本章总结	513
附录 A 数据库基本知识	515
A.1 数据库概述	515
A.1.1 数据库技术的演进及其特点	515



A.1.2	数据模型	516
A.1.3	数据库管理系统	516
A.2	关系数据库	517
A.2.1	关系数据库特点	517
A.2.2	关系数据模型	517
A.3	SQL	518
A.3.1	SQL 简介	518
A.3.2	数据定义语言 DDL	519
A.3.3	数据操纵语言 DML	521
A.3.4	数据控制语言 DCL	523
附录 B	开发接口技术快速进阶	525
B.1	ODBC API 接口	525
B.1.1	分配 ODBC 环境	526
B.1.2	分配连接句柄	526
B.1.3	设置连接属性	526
B.1.4	连接数据源	527
B.1.5	SQL 操作	528
B.1.6	断开同数据源的连接	529
B.1.7	小结	529
B.2	MFC ODBC 接口	529
B.2.1	概述	529
B.2.2	连接数据源	530
B.2.3	增加记录	530
B.2.4	修改记录	530
B.2.5	查询记录	531
B.2.6	删除记录	532
B.2.7	撤销操作	532
B.2.8	SQL 语句的直接执行	532
B.3	DAO 接口	533
B.3.1	概述	533
B.3.2	打开数据库	533
B.3.3	添加记录	533
B.3.4	修改记录	534
B.3.5	删除记录	534
B.3.6	查询记录	534
B.3.7	小结	535
B.4	OLE DB 接口	535



B.4.1	概述	535
B.4.2	OEL DB 应用结构	535
B.4.3	OLE DB 接口 ATL 介绍	536
B.4.4	OLE DB 与 ODBC 的关系	537
B.4.5	小结	537
B.5	ADO 接口	538
B.5.1	概述	538
B.5.2	引入 ADO 库文件	538
B.5.3	初始化 OLE/COM 库环境	539
B.5.4	使用_ConnectionPtr 接口	539
B.5.5	使用_RecordsetPtr 接口	539
B.5.6	使用_CommandPtr 接口	540
B.5.7	关于数据类型转换	540
B.5.8	小结	540
附录 C	本书资源说明	541



数据库技术和互联网（Internet）技术可以说是当今世界范围内最为热门的两大支撑计算（Computing）技术。而且，在很多情况下，数据库是网络计算的后台支撑平台，因此，我们可以认为，数据库技术无论在现在还是将来都将是一项不可或缺的商业应用技术。所以，学习数据库技术和数据库开发技术是一个没有后悔机会的选择！

本书是一本介绍 Visual C++开发数据库应用程序的工具书。本章中我们介绍以下几个方面的内容。

- VC++开发数据库应用的特点是什么
- 常用开发接口技术的比较及选用原则
- 各章的编排思路和编排特点

1.1 VC++开发数据库应用程序的特点和优势

作为数据库开发的前端工具或开发环境有许多种，用得比较多的有 Delphi、Visual Basic 和 PB(PowerBuilder)等。对 Visual C++来说，从 VC++ 4.0 开始就对数据库开发提供了比较好的开发环境；在 VC++ 5 和 VC++ 6 中，对数据库的访问技术更成熟、功能更强大。借助于 VC++可以轻松地开发出功能强、速度快、应用广并且占用资源少的应用程序。

另外，Visual C++提供了多种多样的数据库访问技术——ODBC API、MFC ODBC、DAO、OLE DB、ADO 等。这些技术各有自己的特点，它们提供了简单、灵活、访问速度快、可扩展性好的开发技术。同时，这也是与其他技术相比 VC++开发的优势所在。

1. 简单性

由于 Visual C++中提供了 MFC 类、模板类以及 AppWizard 和 ClassWizard 等一系列“向导”用于产生应用程序这一点会大大简化应用程序的开发。如果借助于某些控件，开发这甚至可以无需编写代码就可以产生一个完整的数据库应用。例如 ODBC 接口和 ADO 接口就可以做到这一点。

2. 灵活性

Visual C++提供的开发环境可以使开发者根据自己的需要设计应用程序的界面和功能，而且，Visual C++提供了丰富的类库和方法，可以使开发者根据自己的应用特点进行选择。例如应用可以是 SDI（单文档）、MDI（多文档）和基于对话框等形式。



3. 访问速度快

为了解决 ODBC 开发的数据库应用程序访问数据库的速度慢的问题, Visual C++ 提供了新的访问技术——OLE DB 和 ADO, OLE DB 和 ADO 都是基于 COM 接口的技术, 使用这种技术可以直接对数据库的驱动程序进行访问, 这大大提高了访问速度。

4. 可扩展性

Visual C++ 提供了 OLE 技术和 ActiveX 技术, 这种技术可以增强应用程序的能力。使用 OLE 技术和 ActiveX 技术可以使开发者利用 Visual C++ 中提供的各种组件、控件以及第三方开发者提供的组件来创建自己的程序, 从而实现应用程序的组件化。使用这种技术可以使应用程序具有良好的可扩展性。

5. 访问不同种类数据源

传统的 ODBC 技术只能访问关系型数据库, 在 Visual C++ 中, 提供了 OLE DB 访问技术, 不仅可以访问关系型数据库, 还可以访问非关系型数据库。

1.2 常用开发技术的比较

Visual C++ 提供了多种访问数据库的技术, 主要包括下面这些数据库访问接口。

- ODBC (Open DataBase Connectivity)
- MFC ODBC (Microsoft Foundation Classes ODBC)
- DAO (Data Access Objects)
- OLE DB (Object Link and Embedding DataBase)
- ADO (ActiveX Data Objects)

这些技术在访问数据库时无论是易用程度还是运行性能都各有千秋, 总结如下:

1. ODBC

ODBC 是客户应用程序访问关系数据库时提供的一个标准的接口, 对于不同的数据库, ODBC 提供了一套统一的 API, 使应用程序可以应用所提供的 API 来访问任何提供了 ODBC 驱动程序的数据库。而且, ODBC 已经成为一种标准, 所以, 目前几乎所有的关系数据库都提供了 ODBC 驱动程序, 这使 ODBC 的应用非常广泛, 基本上可用于所有的关系数据库。

优点是, 由于 ODBC 是一种底层的访问技术, 因此, ODBC API 可以使客户应用程序能够从底层设置和控制数据库, 完成一些高层数据库技术无法完成的功能。

但不足之处是由于 ODBC 只能用于关系数据库, 使得利用 ODBC 很难访问对象数据库及其他非关系数据库。另外, 直接使用 ODBC API 编写应用程序要编制大量代码, 增加了程序编制的复杂程度。

2. MFC ODBC

由于直接使用 ODBC API 编写应用程序要编制大量代码, 在 Visual C++ 中提供了 MFC ODBC 类, 封装了 ODBC API, 这使得利用 MFC 来创建 ODBC 的应用程序非常简便。所以通过 MFC ODBC 类可以简化程序设计。

不足之处是该技术没有提供对数据库的底层控制, 即无法对数据源进行底层操作。

总之, MFC ODBC 技术属于高层访问技术。

3. DAO



DAO 提供了一种通过程序代码创建和操纵数据库的机制。多个 DAO 构成一个体系结构,在这个结构中,各个 DAO 对象协同工作。MFC DAO 是微软提供的用于访问 Microsoft Jet 数据库文件(*.mdb)的强有力的数据库开发工具,它通过 DAO 的封装,向程序员提供了 DAO 丰富的操作数据库手段。

该接口的最大特点是对 Microsoft Jet 数据库的操作很方便,而且是操作 Jet 数据库时性能最好的技术接口之一。并且它并不只能用于访问这种数据库,事实上,通过 DAO 技术可以访问从文本文件到大型后台数据库等多种数据格式。另外,由于 MFC 对 DAO 的封装比较充分,我们可以通过该接口对数据库的结构进行操作。

4. OLE DB

OLE DB 是 Visual C++ 开发数据库应用中提供的新技术,它基于 COM 接口。因此,OLE DB 对所有的文件系统包括关系数据库和非关系数据库都提供了统一的接口。这些特性使得 OLE DB 技术比传统的数据库访问技术更加优越。

直接使用 OLE DB 来设计数据库应用程序需要大量的代码。在 VC 中提供了 ATL 模板,用于设计 OLE DB 数据应用程序和数据提供程序。

与 ODBC 技术相似,OLE DB 属于数据库访问技术中的底层接口。

5. ADO

ADO 技术是基于 OLE DB 的访问接口,它是 VC++ 中提供的面向对象的 OLE DB 技术。它继承了 OLE DB 技术的优点,并且,ADO 对 OLE DB 的接口作了封装,定义了 ADO 对象,使程序开发得到简化,ADO 技术属于数据库访问的高层接口。

为了方便读者更好的理解各种数据库访问接口的优缺点,我们给出了性能比较表 1.1。

表 1.1 各种数据库访问接口的比较

访问接口	易用性	运行性能	可扩展性	突出特点	能否访问非关系数据源	技术层次
ODBC	差	较高	差	可对数据库底层进行控制	否	底层
MFC ODBC	好		一般	通用标准,应用非常广泛	否	高层
MFC DAO	好	较高	一般	访问 Jet 数据库性能最好	能	高层
OLE DB	较难	高	好	能统一访问关系和非关系 DB	能	底层
ADO	最好	高	好	能统一访问关系和非关系 DB; 并有多种编程接口。	能	高层

1.3 本书技术思路

本书的终极目标是能够帮助读者开发出性能优良的数据库应用程序。所谓数据库应用程序,就是能够从数据库管理系统 DBMS (Database Management System) 获取数据并操作数据的程序。



在一般的应用体系结构中，DBMS 处于后台运行，而前端的操作管理等，则借助一些应用程序开发工具开发出符合实际工程需要的、界面友好、简单易用的前台控制系统。一般来说，DBMS（如 ORACLE，DB2，SYBASE 和 MS SQL Server 等）部分都由各专业公司开发和销售。我们所说的数据库程序开发，则是指后一部分——前台控制系统的开发。

下面是一个简单的数据库应用体系结构的示意说明图。如图 1.1 所示。

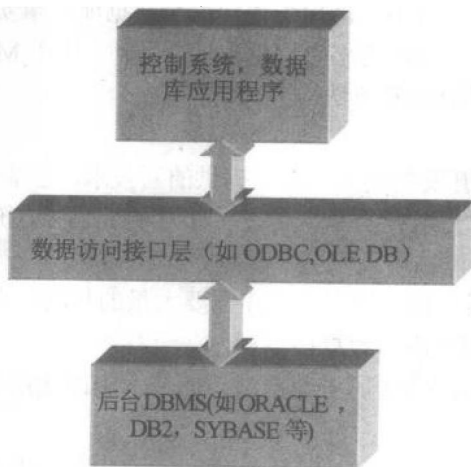


图 1.1 数据库应用体系结构的示意说明图

目前，各种前端开发工具和后台支撑平台系统都非常丰富，各个厂商的产品都各有其特点。虽然，各种模型的数据库系统都有一些业界共同遵守的标准，但相对来说，中间层——数据访问接口层标准性比较强一些，例如几乎所有关系数据库都支持 ODBC 接口。还有 ADO 接口有多种前端开发工具，如 VC++、JAVA 和脚本语言 VBScript 等，只要掌握了一种开发工具，换一种开发工具对涉及访问接口部分变化非常少，只不过根据各种开发语言的风格写法有些差异而已。

本书介绍的前端开发工具为 VC++，后台数据库如没有特别说明指 MS SQL Server。

本书后面介绍的各种数据访问接口虽然具有各自特点，并且实现技术也存在差异，但有一些最本质的概念和原理是相通的。下面对后面将用到的技术编排思路作一简单说明，希望能对读者把握整体脉络有所帮助。

抛开具体的数据访问接口的特性不谈，编写数据库应用程序的过程大概可以分为连接数据源、操作数据源和断开连接等几个步骤。其中，操作数据源部分涉及的内容比较多，是本书论述的除连接数据源之外的另一个重点。



说文解字

——何谓数据源

数据源就是数据的源。一个 ODBC 数据源包含一组特定的数据、访问这些数据所需要的信息以及数据的位置（通常使用数据源名称标识）等。一个数据源可以是桌面应用程序中的单个文件，如 MS Access 或 FoxPro 的文件，或者是一个功能完备的数据库管理系统（DBMS），如 ORACLE，SQL Server 或 Informix 等数据库。从数据源的基本原理看，就是使用户不必知道资源的详细信息，仅通过资源名就可以使用数据。