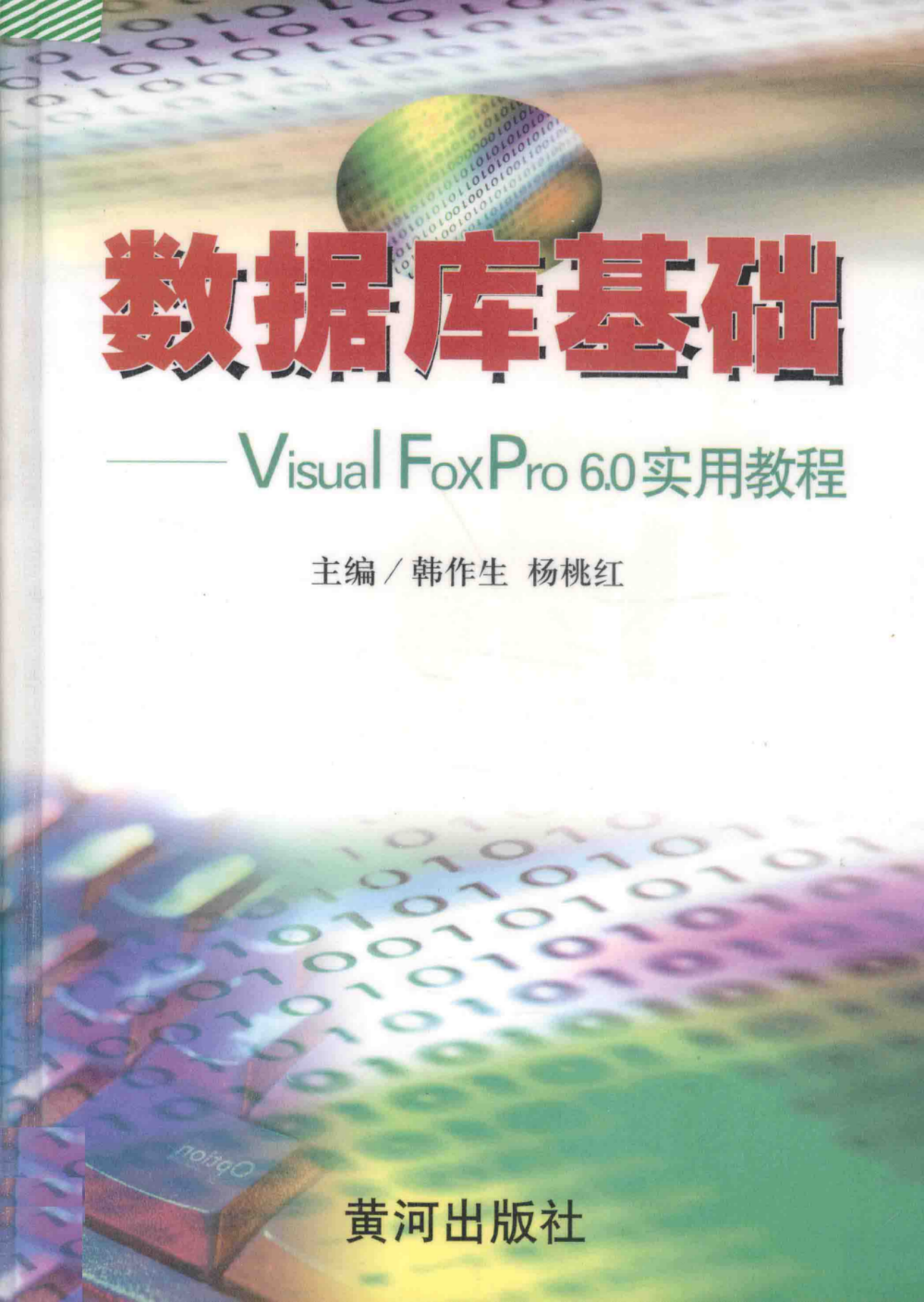




数据库基础

—— Visual FoxPro 6.0 实用教程

主编 / 韩作生 杨桃红

黄河出版社

数 据 库 基 础

——Visual FoxPro 6.0 实用教程

主 编：韩作生 杨桃红

副主编：王飞舟 牟 娟 陈立新

黄 河 出 版 社

2002 年·济南

责任编辑 卢建明
封面设计 徐 宁

图书在版编目(CIP)数据

数据库基础: Visual FoxPro 6.0 实用教程/韩作生,
杨桃红编著. —济南:黄河出版社, 2002.1

ISBN 7-80152-360-1

I. 数... II. ①韩... ②杨... III. 关系数据库-数
据库管理系统, Visual FoxPro 6.0 IV. TP311.138

中国版本图书馆 CIP 数据核字(2001)第 098503 号

书名 数据库基础

——Visual FoxPro 6.0 实用教程

主编 韩作生 杨桃红

出版 黄河出版社

发行 黄河出版社发行部

(济南市英雄山路 19 号 250002)

印刷 济南申汇印务有限责任公司

规格 787×1092 毫米 16 开本

20.5 印张 326 千字

版次 2002 年 1 月第 1 版

印次 2002 年 1 月第 1 次印刷

印数 1—3000 册

书号 ISBN 7-80152-360-1/G·069

定价 39.50 元

前 言

数据库管理系统是计算机应用的一个重要领域，数据库系统以其数据的结构化、低冗余、易扩充、易学易用等特点深受广大用户的欢迎，在当今的社会发展中有着重要的作用。

Visual FoxPro 6.0 是 Microsoft 公司推出的全新的关系型数据库管理系统，适用于 Windows 平台，是目前使用最广、功能最强的微机数据库产品之一。它具有强大的性能、完整而丰富的工具、极其友好的图形用户界面、简单的数据管理方式、良好的兼容性和真正的可编译性，使数据的组织、管理和应用等工作变得简单易行。

Visual FoxPro 提供了多种可视化的工具、向导和生成器，提供了一个集成化的开发环境，可以快速地创建数据库、表单、菜单、查询、视图和报表，并支持面向对象技术和 ActiveX 等新技术，为快速开发功能全面的应用程序提供了极为便利的条件。

全书在结构上力求做到符合教学规律，循序渐进。一方面，通过具有针对性的实例，满足初学者的需要，使读者能很快地熟悉 Visual FoxPro 的应用环境和使用方法；另一方面，深入浅出地介绍了 Visual FoxPro 程序编制的基本概念和方法，并详细介绍了面向对象编程技术的基本方法和技巧，对 Visual FoxPro 的整体面貌作了较为清晰的说明。

本书详细介绍了应用程序的开发步骤和方法，通过一个个具体的实例和具体的操作过程，读者很快就能编制出自己的应用程序。

参加本书编写的有韩作生、杨桃红、王飞舟、牟娟、陈立新、郭军、丰久宽、杨峰、韩新瑞、沙丽云、徐明春、黄涛、王瑞涛。全书由韩作生负责初稿的修改和最后的统稿工作。

由于时间紧迫以及作者的水平有限，书中难免有不足之处，恳请广大读者批评指正。

编者

2001.11

目 录

第一章 数据库基础知识	(1)
1.1 数据库系统概述	(1)
1.1.1 数据管理的发展	(1)
1.1.2 数据库的基本概念	(2)
1.1.3 数据库系统的特征	(3)
1.1.4 数据库系统的结构	(4)
1.2 数据模型	(5)
1.2.1 层次数据模型	(5)
1.2.2 网状数据模型	(5)
1.3 关系数据模型	(6)
1.3.1 关系模型	(6)
1.3.2 关系数据查询语言 SQL	(10)
1.4 数据库安全与恢复	(11)
1.4.1 数据库的安全性	(11)
1.4.2 数据库的恢复	(11)
习题一	(12)
第二章 Visual FoxPro 6.0 概述	(13)
2.1 Visual FoxPro 6.0 功能概述	(13)
2.1.1 Visual FoxPro 6.0 主要功能和特点	(13)
2.1.2 Visual FoxPro 6.0 新增功能和新特点	(14)
2.2 Visual FoxPro 6.0 系统的安装、启动与退出	(16)
2.2.1 系统要求	(16)
2.2.2 Visual FoxPro 6.0 安装过程	(17)
2.2.3 Visual FoxPro 6.0 的启动与退出	(18)
2.3 Visual FoxPro 6.0 环境配置与系统优化	(21)
2.3.1 Visual FoxPro 6.0 的环境配置	(21)
2.3.2 Visual FoxPro 6.0 的系统优化	(24)
2.3.3 Visual FoxPro 6.0 的系统容量	(25)
2.4 Visual FoxPro 6.0 的集成开发环境	(27)
2.4.1 Visual FoxPro 的窗口组成	(28)
2.4.2 Visual FoxPro 系统的菜单操作	(28)
2.4.3 Visual FoxPro 命令窗口简介	(33)
2.4.4 Visual FoxPro 快捷工具栏	(33)
2.4.5 Visual FoxPro 的设计器、向导和生成器简介	(35)

习题二	(35)
第三章 VFP 6.0 基本语言要素和程序设计初步	(36)
3.1 数据类型	(36)
3.2 数据容器	(38)
3.2.1 常量	(38)
3.2.2 变量	(38)
3.2.3 记录	(44)
3.2.4 对象	(44)
3.3 运算符和表达式	(44)
3.3.1 算术表达式	(44)
3.3.2 字符表达式	(45)
3.3.3 日期及日期时间型表达式	(45)
3.3.4 关系表达式	(45)
3.3.5 逻辑表达式	(47)
3.3.6 混合运算	(47)
3.3.7 名表达式	(47)
3.4 函数	(48)
3.4.1 函数的格式和调用	(48)
3.4.2 数值运算函数	(48)
3.4.3 字符操作函数	(50)
3.4.4 日期和时间函数	(53)
3.4.5 类型转换函数	(55)
3.4.6 测试函数	(56)
3.5 命令	(61)
3.5.1 命令格式	(61)
3.5.2 命令使用规则	(62)
3.5.3 输入命令	(62)
3.5.4 输出命令	(63)
3.6 程序	(64)
3.6.1 程序的建立和运行	(64)
3.6.2 顺序结构程序设计	(67)
3.6.3 分支结构程序设计	(67)
3.6.4 循环结构程序设计	(70)
3.6.5 子程序及其调用	(75)
3.7 过程	(76)
3.7.1 过程与过程文件	(76)
3.7.2 过程文件的打开及过程的调用	(77)
3.7.3 过程调用的嵌套与递归	(78)
3.8 自定义函数	(79)
3.8.1 独立自定义函数	(79)

3.8.2 依附自定义函数	(80)
3.9 参数传递	(80)
3.9.1 利用不同属性的变量实现参数传递	(80)
3.9.2 利用参数传递语句传递参数	(83)
习题三	(84)
第四章 数据表的基本操作	(90)
4.1 创建新表	(90)
4.1.1 数据表的概念	(90)
4.1.2 建立数据表的准备工作	(91)
4.1.3 建立数据表的结构	(92)
4.2 数据表的基本操作	(96)
4.2.1 打开表和关闭表	(96)
4.2.2 输入记录	(97)
4.2.3 显示和修改表结构	(99)
4.2.4 表的浏览与编辑	(100)
4.2.5 记录指针及其操作	(101)
4.2.6 直接修改字段值	(102)
4.2.7 记录的删除与恢复	(103)
4.2.8 定制表	(105)
4.3 排序与索引	(108)
4.3.1 排序	(108)
4.3.2 索引	(109)
4.4 记录查询与数据统计	(113)
4.4.1 记录查询	(113)
4.4.2 数据表的统计操作	(115)
4.5 多表操作	(118)
4.5.1 工作区的概念	(118)
4.5.2 数据工作期	(119)
4.5.3 多表操作	(120)
4.6 文件复制和转换	(123)
4.6.1 数据表的复制	(123)
4.6.2 数据表转换为其他数据文件格式	(124)
4.6.3 从其他格式的数据文件中取得数据	(124)
习题四	(126)
第五章 数据库的管理与使用	(127)
5.1 创建数据库	(127)
5.1.1 数据库设计的一般过程	(127)
5.1.2 创建数据库	(127)
5.1.3 数据库的打开和关闭	(128)
5.1.4 向数据库中添加和删除表	(129)

5.1.5	建立表之间的永久关系	(130)
5.2	设置表的字段属性	(131)
5.2.1	设置字段标题	(131)
5.2.2	为字段加入注释	(132)
5.2.3	设置字段的默认值	(133)
5.2.4	设置字段有效性的规则和说明	(133)
5.3	设置表的属性	(134)
5.3.1	设置表的替代名	(135)
5.3.2	设置表的有效性规则和错误信息	(135)
5.3.3	设置触发器	(135)
5.3.4	为表文件加入注释	(136)
5.3.5	使用参照完整性生成器	(136)
5.4	查看和修改数据库结构	(137)
5.4.1	浏览数据库	(137)
5.4.2	扩展数据库文件	(137)
5.4.3	检查数据库	(138)
5.5	项目管理器	(138)
5.5.1	项目的建立和打开	(138)
5.5.2	项目管理器的组成	(139)
5.6	管理数据库	(141)
5.6.1	向项目中添加数据库	(141)
5.6.2	从项目中移去数据库	(142)
5.6.3	从磁盘中删除数据库	(142)
5.7	使用多个数据库	(142)
5.7.1	打开多个数据库	(142)
5.7.2	设置当前数据库	(142)
5.7.3	访问其他数据库中的表	(143)
5.7.4	关闭数据库	(143)
习题五		(143)
第六章	查询与视图	(145)
6.1	查询	(145)
6.1.1	设计查询	(145)
6.1.2	利用“查询向导”建立查询	(146)
6.1.3	利用“查询设计器”建立和修改查询	(151)
6.1.4	运行查询	(159)
6.2	视图	(161)
6.2.1	创建本地视图	(161)
6.2.2	定义连接	(167)
6.2.3	创建远程视图	(170)
6.2.4	设置视图属性和连接属性	(171)

6.2.5	利用视图更新数据	(171)
6.2.6	使用视图	(173)
6.2.7	显示和修改视图结构	(174)
6.2.8	删除视图和创建视图索引	(174)
6.2.9	视图集成	(175)
习题六		(175)
第七章 面向对象程序设计基础		(177)
7.1	基本概念	(177)
7.1.1	对象和类	(177)
7.1.2	VFP的对象和类	(178)
7.2	VFP面向对象程序设计	(182)
7.2.1	VFP的 OOP 工作环境	(182)
7.2.2	VFP对象的创建	(184)
7.3	程序方式创建 VFP 对象	(188)
7.3.1	对象引用规则	(188)
7.3.2	程序方式创建对象	(189)
7.4	类的创建和使用	(193)
7.4.1	类的创建	(193)
7.4.2	将类添加到工具栏	(196)
7.4.3	类的编辑	(197)
7.4.4	为字段设置类	(197)
习题七		(198)
第八章 表单		(199)
8.1	表单向导	(199)
8.1.1	启动表单向导	(199)
8.1.2	表单向导的使用	(199)
8.2	表单设计器	(201)
8.2.1	表单设计的基本步骤	(201)
8.2.2	表单设计器窗口	(203)
8.3	表单控件	(205)
8.3.1	输出类控件	(206)
8.3.2	控制类控件	(209)
8.3.3	输入类控件	(213)
8.3.4	容器类控件	(227)
8.3.5	连接类控件	(237)
习题八		(241)
第九章 报表和标签		(243)
9.1	创建报表和标签	(243)
9.1.1	用快速报表建立报表	(243)
9.1.2	用标签设计器设计标签	(245)

9.2 “报表设计器”的使用	(246)
9.2.1 报表设计器	(247)
9.2.2 “报表设计器”工具栏	(249)
9.2.3 “报表控件”工具栏	(249)
9.2.4 “布局”工具栏	(250)
9.2.5 “调色板”工具栏	(251)
9.2.6 “报表”菜单	(251)
9.2.7 “显示”菜单	(251)
9.3 报表的数据源和布局	(252)
9.3.1 计划报表布局	(252)
9.3.2 设置报表数据环境	(253)
9.4 定制报表布局	(254)
9.4.1 定义报表的页面	(255)
9.4.2 设置域控件格式	(255)
9.4.3 添加线条、矩形和圆形	(256)
9.4.4 添加图片	(257)
9.4.5 为报表控件添加注释	(258)
9.5 在报表带区中添加控件	(258)
9.5.1 添加字段	(258)
9.5.2 添加标签控件	(258)
9.5.3 选择、移动及调整报表控件的大小	(260)
9.5.4 复制和删除报表控件	(261)
9.5.5 对齐控件	(261)
9.5.6 调整控件的位置	(262)
9.6 使用报表变量	(262)
9.7 报表数据分组	(263)
9.7.1 添加单个组	(263)
9.7.2 添加多个数据分组	(264)
9.7.3 更改分组设置	(265)
9.8 报表和标签的预览与打印	(265)
9.8.1 预览	(265)
9.8.2 打印报表和标签	(265)
习题九	(266)
第十章 菜单和自定义工具栏	(267)
10.1 菜单设计	(267)
10.1.1 菜单系统的设计	(267)
10.1.2 菜单系统的规划原则	(268)
10.1.3 创建菜单系统	(268)
10.1.4 定制和修改菜单	(270)
10.1.5 为菜单指定任务	(273)

10.1.6	有关菜单操作的其他内容	(280)
10.1.7	测试与调试菜单系统	(281)
10.1.8	定制菜单系统	(281)
10.1.9	在应用程序中调用菜单	(283)
10.2	自定义工具栏.....	(283)
10.2.1	创建工具栏.....	(283)
10.2.2	协调菜单和自定义工具栏.....	(286)
习题十		(287)
第十一章	创建应用程序的帮助系统.....	(290)
11.1	HTML 帮助系统	(290)
11.1.1	调用 HTML 帮助	(291)
11.1.2	编程实现帮助特性.....	(293)
11.1.3	发布已编译的 HTML 帮助系统	(293)
11.2	图形样式的帮助系统.....	(294)
11.2.1	WinHelp 帮助系统的特性	(294)
11.2.2	访问 WinHelp 帮助系统	(294)
11.2.3	典型的帮助菜单.....	(294)
11.2.4	实现 WinHelp 帮助	(294)
11.2.5	使用 WinHelp 函数	(295)
11.2.6	指定 WinHelp 参数	(295)
11.3	创建 .DBF - 样式帮助	(297)
11.3.1	设计 .DBF - 样式帮助	(297)
11.3.2	一个 .DBF - 样式帮助文件示例	(297)
11.3.3	使用 .DBF - 样式帮助	(300)
11.3.4	自定义 .DBF 形式的帮助.....	(300)
习题十一		(302)
第十二章	生成和发布应用程序.....	(303)
12.1	生成应用程序.....	(303)
12.1.1	设置主文件.....	(303)
12.1.2	生成应用程序.....	(304)
12.2	发布应用程序.....	(305)
12.2.1	创建发布目录树.....	(306)
12.2.2	安装向导的功能.....	(307)
12.2.3	运行安装向导.....	(307)
12.2.3	制作应用程序发布包.....	(313)
习题十二		(314)

第一章 数据库基础知识

1.1 数据库系统概述

数据库技术产生于 20 世纪 60 年代末 70 年代初,其主要目的是为了有效地管理和存取大量的数据资源。数据库技术主要研究如何存储、使用和管理数据,是计算机数据管理技术发展的最新阶段。

1.1.1 数据管理的发展

数据管理是指对数据的组织、分类、编码、存储、检索和维护。与其他任何技术的发展一样,计算机数据管理也经历了由低级到高级的发展过程。随着计算机硬件(尤其是外存储器)、软件技术和计算机应用范围的发展,计算机数据管理大致经历了人工管理、文件系统管理和数据库系统管理三个阶段,其中数据库系统管理阶段又分为集中式数据库系统管理和分布式数据库系统管理两个阶段。

1. 人工管理阶段

1946 年美国发明了世界上第一台电子计算机,这时的计算机主要用于科学计算,外存储器只有卡片、纸带和磁带,存储空间极其有限,没有像磁盘这样的可以随机访问、直接存取的外部存储设备。软件方面,也没有专门管理数据的软件,数据由计算或处理它的程序自行携带,数据处理方式基本是批处理。这种数据处理方式非常简单,数据依赖于程序,不能单独使用,程序关闭后数据也就关闭了,数据的管理依靠人工方式,程序员对数据具有绝对的控制权。

2. 文件系统管理阶段

到了 50 年代末,随着硬件的发展和数据量的增加,计算机开始大量地用于管理中的数据处理工作。在硬件方面,可直接存取的磁鼓和磁盘成为联机的主要外存。在软件方面,出现了高级语言和操作系统。操作系统中的文件系统是管理外存储器的数据管理软件。数据处理方式除了批处理,还有联机实时处理。数据与程序有了一定的独立性,出现了程序文件和数据文件。数据文件可以长期保存在外存储器上多次存取使用,数据的存取以记录为基本单位,并出现了多种文件组织形式,如顺序文件、索引文件和随机文件等。

在文件系统的支持下,数据的逻辑结构与物理结构之间可以有一定的差别,逻辑结构和物理结构之间的转换由文件系统的存取方法来实现。数据与程序之间具有物理独立性,程序只需用文件名访问数据,不必关心数据的物理位置,但是要求应用程序的设计者需要对程序所使用的文件的逻辑结构和物理结构都了解得非常清楚。由于文件格式的差别,程序员还要为程序中所用到的每一个文件都写好相应的接口,而计算机操作系统只提供将文件打开、关闭和保存等非常低级的操作。所以尽管文件系统对数据的管理有了极大的进步,但还是存在数据冗余度大、数据的针对性强以及数据对程序的依赖性强等缺陷,文件系统自身的限制还导致数据共享能力差以及数据的管理缺乏整体性和统一性等。

3. 数据库系统阶段

从 60 年代后期开始,大容量磁盘系统的采用,使计算机联机存取大量数据成为可能;而软件价格的上升和硬件价格的相对下降,使独立开发系统维护软件的成本增加。针对计算机管理的数据量急剧增长和数据的独立性问题,人们逐步发展了以统一管理数据和共享数据为主要特征的系统,这就是数据库系统。它通过有效地管理和存取大量的数据资源,包括提高数据的共享性,使多个用户能够并发地存取数据库中的数据,减小了数据的冗余度,提高了数据的一致性和完整性,并提供了数据与应用程序的独立性,从而减少了应用程序的维护代价。

在此期间出现了两种数据库类型——网状数据库和层次数据库。1964 年,美国通用电气公司成功地开发了世界上的第一个数据库系统——IDS (Integrated Data Store)。IDS 奠定了网状数据库的基础,并且得到了广泛的发行和应用,成为数据库系统发展史上的一座丰碑。1969 年,美国国际商用机器公司 (IBM) 也推出了世界上第一个层次数据库系统 IMS (Information Management System),同样在数据库系统发展史上占有重要的地位。这两种数据库都属于集中式数据库。

计算机技术和网络技术的发展,为远距离数据处理提供了可能,跨区域的数据交流对数据库也有了更高的要求。早在 70 年代, E. F. Codd 就已经在总结层次、网状数据库优点的基础上提出了关系数据模型 (A relation model for large shared data banks) 的概念,并提出了关系代数和关系演算,从理论上确立了完整的关系模型理论、数据依赖理论和关系数据库的设计理论。E. F. Codd 的关系数据理论和网络技术的飞速发展,促使数据库技术和网络技术相互结合,使数据库系统从集中式发展到分布式,形成分布式数据库系统,工作方式也从主机/终端系统结构发展到客户机/服务器结构。

分布式数据库是数据库技术和网络技术紧密结合的产物。分布式数据库是一个逻辑上统一、地域上分散的数据集合,是计算机网络环境中各个节点局部数据库的逻辑集合,同时受分布式数据库管理系统的控制和管理。

数据库管理系统 DBMS (DataBase Management System) 是伴随着数据库的发展,人们开发出的一系列建立、使用和维护数据库的管理软件。它成功地解决了由于数据共享和数据的并发操作所带来的一系列问题。目前较流行的数据库管理系统包括 Oracle、Informix 和 Sybase 等。微机上的数据库管理系统有 dBASE、Paradox、Access、clipper、Fox 系列 (Fox-BASE、FoxPro、FoxPro for Windows 和 Visual FoxPro) 以及大中型数据库系统的微机版本等。

进入 20 世纪 80 年代之后,计算机硬件技术有了飞速的提高。计算机技术的提高促使计算机应用不断深入,产生了许多新的应用领域,例如计算机辅助设计、计算机辅助制造、计算机辅助教学、办公自动化、智能信息处理和决策支持等。这些新的领域对数据库系统提出了新的要求。但是由于应用的多元化,不能设计出一个统一的数据模型来表示这些新型的数据及其相互关系,因而出现了百家争鸣的局面,产生了演绎数据库、面向对象数据库、分布式数据库、工程数据库、时态数据库和模糊数据库等新型数据库的研究和应用。不过到目前为止,在世界范围内应用较多的还是经典的关系数据库系统。

1.1.2 数据库的基本概念

数据库系统是一个实际可运行的存储、维护和向应用系统提供数据的软件系统,是存储

介质、处理对象和管理系统的集合体，是依照某种数据模型组织起来并存放的数据集合。这些数据为多个应用服务，独立于具体的应用程序。数据库由数据库管理系统统一管理，数据的插入、修改和检索均要通过数据库管理系统进行。数据库管理系统是一种系统软件，它的主要功能是维护数据库并有效地访问数据库中任意部分的数据。对数据库的维护包括保持数据的完整性、一致性和安全性。数据管理员负责创建、监控和维护整个数据库，使数据能被任何有权使用的人有效地使用。

数据库系统的个体含义是指一个具体的数据库管理系统软件和用它建立起来的数据库；它的学科含义是指研究、开发、建立、维护和应用数据库系统所涉及的理论、方法和技术所构成的学科。在这一含义下，数据库系统是软件研究领域的一个重要分支，常称为数据库领域。

数据库研究跨越于计算机应用、系统软件和理论研究三个领域，其中应用促进新系统的研制开发，新系统带来新的理论研究，而理论研究又对前两个领域起着指导作用。数据库系统的出现是计算机应用的一个里程碑，它使得计算机应用从以科学计算为主转向以数据处理为主，从而使计算机得以在各行各业乃至家庭普遍使用。在它之前的文件系统虽然也能处理持久数据，但是文件系统不提供对任意部分数据的快速访问，而这对数据量不断增大的应用来说是至关重要的。为了实现对任意部分数据的快速访问，就要研究许多优化技术。这些优化技术往往很复杂，是普通用户难以实现的，所以就由系统软件（数据库管理系统）来完成，而提供给用户的是简单易用的数据库语言。由于对数据库的操作都由数据库管理系统完成，所以数据库就可以独立于具体的应用程序而存在，从而数据库可以为多个用户所共享。因此，数据的独立性和共享性是数据库系统的重要特征。数据共享节省了大量人力物力，为数据库系统的广泛应用奠定了基础。数据库系统的出现使得普通用户能够方便地将日常数据存入计算机并在需要的时候快速访问它们，从而使得计算机走出科研机构进入各行各业，进入家庭。

1.1.3 数据库系统的特征

自 20 世纪 60 年代出现数据库以来，数据管理技术得到长足的发展，形成了极具特色的数据库系统。其主要特征如下：

(1) 实现数据共享，减少数据冗余

在数据库系统中，对数据的定义和描述已经从应用程序中分离出来，通过数据库管理系统来统一管理。数据的最小访问单位是数据项，既可以按数据项的名称存取库中某一个或某一组数据项，也可以存取一条记录或一组记录。

(2) 采用特定的数据模型

数据库中的数据结构由数据模型来表示，不仅表示事物内部各数据项之间的联系，而且反映事物与事物之间的联系。

(3) 数据的独立性好

在数据库系统中，DBMS 提供映像的功能，确保应用和谐，对数据结构和存取方法有较高的独立性。数据的逻辑表现与数据的内部物理管理和存储方式是相互独立的。用户只以简单的逻辑结构来操作数据，无需考虑数据在存储器上的物理位置与结构。

(4) 有统一的数据控制功能

数据库作为多个用户和应用程序的共享资源，对数据的存取往往是并发的，即多个用户

同时使用同一个数据库。数据库管理系统必须提供并发控制功能、数据的安全性控制功能和数据的完整性控制功能。

对于分布式数据库，还具有结构简单、分布的透明性、局部自治与集中控制相结合、可靠性和可用性好、效率高和灵活性好等特征。

1.1.4 数据库系统的结构

数据库系统是指引进数据库技术后的计算机系统，一般是由硬件系统、数据库集合、数据库管理系统及相关软件、数据库管理员和用户五部分构成的，结构如图 1-1 所示。



图 1-1 数据库系统的结构

(1) 硬件系统

运行数据库系统的计算机需要有足够大的内存、足够大容量的磁盘等联机直接存取设备和较高的通道能力以支持对外存的频繁访问，还需要有足够数量的脱机存储介质，如软盘、磁带和光盘等存放数据库备份。

(2) 数据库集合

系统包括若干个设计合理、满足应用需求的数据库。数据库由 DBMS 通过操作系统读写。

(3) 系统软件

数据库管理系统是为数据库的建立、使用和维护而配置的软件。它是数据库系统的核心组成部分，其他还包括支持其运行的操作系统、开发数据库管理及应用系统的其他相关软件。

(4) 数据库管理员

对于较大规模的数据库系统必须有专人全面负责建立、维护和管理数据库系统，承担此任务的人员称为数据库管理员 (DBA, DataBase Administrator)。数据库管理员的职责包括：定义并存储数据库的内容，监督和控制数据库的使用，负责数据库的日常维护，必要时重新组织和改进数据库等。

(5) 用户

数据库的用户分为两类：一类是最终用户，主要对数据库进行联机查询或通过数据库应用系统提供的界面来使用数据库。这些界面包括菜单、表格、图形和报表。最终用户所感觉到的数据库系统除了计算机硬件之外，就是数据库管理系统 DBMS 和数据库应用程序。另一类是专业用户——程序员，他们负责设计应用系统的程序，为最终用户开发适用的数据库应用系统。

对一个具体的数据库系统而言，它的结构取决于它采用的数据模型，下一节中我们将介绍几种最常用的数据模型。

1.2 数据模型

在计算机的世界里，计算机只能认识由 0、1 组成的数据，是一个数据的世界，而我们要处理的是现实世界中的信息，这就需要我们先将现实世界中的信息进行抽象，形成一个中间模型，这个模型要明确各个实体、实体属性、实体之间的联系以及表征每个实体的关键字等，称为实体联系模型（E-R 模型）。由实际的实体联系，我们知道 E-R 模型是由一对一（1:1）、一对多（1:n）和多对多（m:n）三种基本联系构成，从而可形成不同的 E-R 模型。E-R 模型并不能直接存入计算机中，它还需要进一步表示成便于计算机处理的数据模型，因此针对于不同的 E-R 模型产生了不同的数据模型，主要有层次模型、网状模型、关系模型等，本节主要介绍层次模型和网状模型。

1.2.1 层次数据模型

层次数据模型是用树形结构来表示实体及其之间联系的模型，是一个自上而下的结构，与计算机中的目录结构是一样的，如图 1-2 所示。

在这种模型中，数据被组织成由“根”开始的“树”，每个实体由根开始沿着不同的分支放在不同的层次上；如果不再有分支，那么此分支序列中最后的节点称为“叶”；树中的每一个节点代表一个记录类型（对应于实体），它包含了该记录的属性，如在中文系节点中除系名外还应含有系的编号、系主任等属性信息；层与层之间的连线表明它们之间的从属关系即链接指针，图 1-2（a）表示了一种最基本的层次关系。由此可知，在这种数据模型中有且仅有一个节点无父节点，此节点即为树的根，其他节点有且仅有一个父节点。层次数据模型在表现一对一和一对多的联系中非常直接、方便，但是在表现多对多的联系中只有将多对多拆成多个一对多才能进行表示。利用这种数据模型建立的数据库称为层次数据库（HDB）。

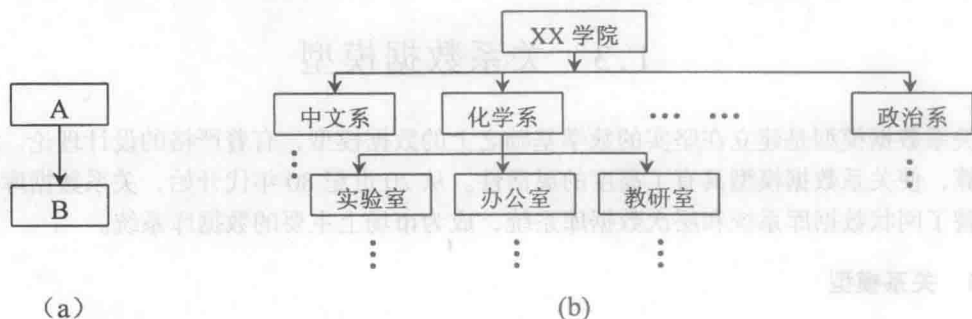


图 1-2 层次数据模型

1.2.2 网状数据模型

在这种数据模型中，实体之间的联系用网状结构来表示，这种结构突破了层次模型的限制，允许节点有多于一个的父节点，也可以有一个以上的节点没有父节点。因此网状模型可以方便地表示各种类型的联系。网中的每一节点代表一个记录类型，联系用链接指针来实现，如图 1-3 所示。

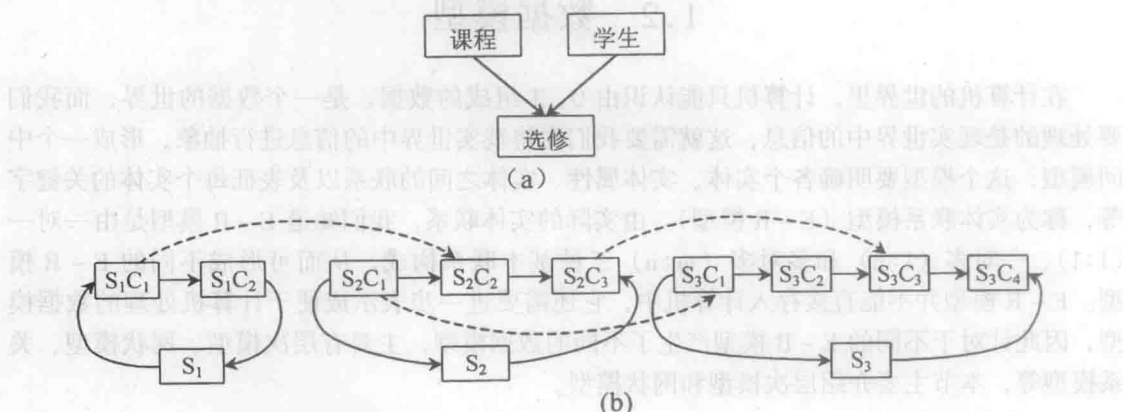


图 1-3 网状数据模型

图 1-3(a)表明了一种基本的对应关系,图 1-3(b)是一种具体的应用关系,可以看出当多对多的关系较多时其结构的复杂性,也就是说网状数据模型是以数据结构的复杂化来换取处理多对多关系的灵活性。

无论是哪一种数据模型,在逻辑上都是基本层次关系的集合,都是用节点表示实体,用箭头(链接指针)表示联系,这种方式实际上是利用指针将所有数据都捆绑在一起,这样模型建立之后,当存储数据时相应的链接指针也就固定了,使用数据时就要求用户对数据的结构或联系比较清楚才行;一旦需要数据改动或数据扩充时就需要调整或增删大量的链接指针,甚至于修改数据的模型结构,重新装载整个系统。

针对层次模型和网状模型的这些缺陷, E. F. Codd 提出了关系数据模型。关系数据模型从根本上抛弃了层次模型和网状模型中采用的父子关联的概念,取而代之的是以关系数据理论为基础的、把数据组织成一种表的结构。在下一节中我们将具体介绍关系数据模型。

1.3 关系数据模型

关系数据模型是建立在坚实的数学基础之上的数据模型,有着严格的设计理论。通过关系运算,使关系数据模型具有了高度的灵活性。从 20 世纪 80 年代开始,关系数据库系统逐步代替了网状数据库系统和层次数据库系统,成为市场上主要的数据库系统。

1.3.1 关系模型

1. 二维表

在关系模型的数据库中,现实世界的的数据被组织成一些二维表格,表格中的数据反映了实体与实体之间的联系,其中每个字段是表中的一列,每个记录是表中的一行,由此构成关系数据模型,这种表格在关系数据库中称为关系。一张二维表格对应一个关系,关系在磁盘上以文件形式存储。其形式如表 1-1 和表 1-2 所示。

表 1-1 是一张订单,显示了订货关系,表 1-2 是库存清单,显示了当前可供商品,这两张单据通过字段“出货号”联系起来,由两张表可以看出,有的商品暂时没有订货,有的商品不止一张订单,而有的商品暂时没有库存,从而反映出实体间的相互关系。