

● 高等学校教材

Information
Technology

数据库原理及 Visual FoxPro应用

□ 赵忠孝



高等教育出版社
Higher Education Press

高等学校教材

数据库原理及 Visual FoxPro 应用

趙忠孝



高等教育出版社

内容提要

本书从关系数据库的原理出发,以应用程序开发为主线,以学籍管理系统为实例,介绍了数据库的基本原理和 Visual FoxPro 6.0 数据库管理系统的应用。在应用方面,详细介绍了 Visual FoxPro 6.0 的基本命令和具体操作,程序设计的基本结构和方法,面向对象程序设计的理论、方法和步骤。

书中内容可以指导读者完成一个完整的数据库管理系统软件的开发。

本书既可作为高等师范院校数据库及 Visual FoxPro 应用课程教材,也可供以 Visual FoxPro 开发数据库的人员参考。

图书在版编目(CIP)数据

数据库原理及 Visual FoxPro 的应用/赵忠孝. 北京:
高等教育出版社, 2004.8

ISBN 7-04-015390-4

I. 数... II. 赵... III. 关系数据库-数据库管理系统, Visual FoxPro-高等学校-教材
IV. TP311.138

中国版本图书馆 CIP 数据核字 (2004) 第 066739 号

策划编辑 耿 芳 责任编辑 耿 芳 市场策划 刘 茜
封面设计 王凌波 责任印制 陈伟光

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100011
总 机 010-82028899

购书热线 010-64054588
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>

经 销 新华书店北京发行所
印 刷 北京市白帆印务有限公司

开 本 787×1092 1/16
印 张 26.75
字 数 550 000

版 次 2004 年 8 月第 1 版
印 次 2004 年 8 月第 1 次印刷
定 价 27.50 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

郑 重 声 明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其行为人将承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人给予严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

反盗版举报电话：(010) 58581897/58581896/58581879

传 真：(010) 82086060

E - mail: dd@hep.com.cn

通信地址：北京市西城区德外大街 4 号

高等教育出版社打击盗版办公室

邮 编：100011

购书请拨打电话：(010)64014089 64054601 64054588

前 言

数据库技术、网络技术和多媒体技术是当今计算机领域发展最快、应用最广泛的3个分支。本书从应用最广泛的关系数据库的原理出发,以应用开发程序为主线,以学籍管理系统为实例,介绍了数据库的基本原理和 Visual FoxPro 6.0 数据库管理系统的应用。

为使读者设计的数据库管理系统更合理、科学和高效,本书第1章详细介绍了数据库的基本理论,其中包括数据库系统的主要特点和数据模型;关系数据库的基本概念、关系代数、函数依赖、关系完整性和关系范式(1NF、2NF、3NF);数据库系统的结构。在数据库设计方面,介绍了如何进行需求分析、概念设计、逻辑设计、物理设计、数据库的实现和维护。在理论上有一定的深度,对于一般的读者来说,只要掌握了这些基本知识,就能针对各种不同的数据模型,科学地构建数据库管理系统。

Visual FoxPro 6.0 是由 Microsoft 公司推出的,适用于微机的、面向对象的、可视化的关系数据库管理系统。它具有功能强大、操作方便、简单实用和友好的用户界面等特性,是众多微机数据库系统的杰出代表。

本书第2章概括性地介绍了 Visual FoxPro 6.0 的功能、特性和运行环境。通过介绍几个常用的向导,使读者能够尽快地进入编程状态,并可以运用这些工具检验后面讲到的概念和方法。

第3章详细介绍了数据元素,因为这是构成数据库系统最基本的单位。不同的数据类型,具有不同的运算规则,用户在组建数据库系统时必须充分了解和掌握。

第4、5、6章介绍了数据表 and 数据库的关系,如何构建表和数据库以及如何建立表之间的联系等内容。Visual FoxPro 为用户提供了可视化的设计器、生成器和向导等设计工具,使组织数据、定义数据库和建立应用程序等工作变得简单易行,用户可以方便地创建数据表、数据库。使用这些设计工具进行操作,实际是执行 Visual FoxPro 的一系列命令代码。因此,详细地介绍了完成相应操作的命令格式,使得用户在编程时,能用这些命令完成相应的操作,为程序设计打下坚实的基础。

第7章详细介绍了关系数据库查询语言 SQL 语句的各种查询功能。书中的查询实例与《数据库系统概论》(萨师煊、王珊,高等教育出版社)一书的内容相配合,并在 Visual FoxPro 中得到了实现和验证。因此,本书还可作为数据库系统概论课程的上机实践用书。

本书第8章详细介绍了视图的概念和构成。视图和查询的差别在于视图是可更新的,而查询则不行。若想从本地或远程表中提取一组可以更新的数据,就需要使用视图。

一个实用的数据库管理系统是不可能在交互方式下完成所有操作的,只有编写优秀的

程序代码,才能体现计算机自动程序执行的优点并充分发挥计算机的优越性。因此,掌握程序设计技术,是对数据库进行高效管理的有效方法。

第9章首先介绍了结构化程序设计的思想,读者应重点掌握顺序、选择和循环3种基本结构的设计方法,进一步掌握多模块程序设计的技巧。

Visual FoxPro 6.0 还是一个面向对象的可视化程序设计语言,第10、11章介绍了类、对象、事件、方法的基本概念。详细介绍了如何使用系统提供的多种设计器,完成对各种表单和控件的设计工作。

报表和标签是打印输出的主要形式,但在多数教材中都没有进行详细的介绍。本书第12章详细介绍了报表的各种布局,各带区的功能和设计方法,并介绍了一些特殊表格的制作方法,这对于读者设计实用表格是十分必要的。

菜单是系统与用户交互的第一界面,必须方便、友好。本书第13章介绍了制作下拉菜单和快捷菜单的详细方法。

应用程序的发布对于一个实用的软件是必须的,但多数教材介绍得不太详细,没有注意到其中的某些细节,使制作出的安装盘在安装时存在一定的缺陷。本书第14章详细介绍了解决这些问题的方法。

作为一种强大的数据库设计工具,Visual FoxPro 6.0 所涵盖的内容是十分庞杂的。本书仅抓住其中的主要内容,进行了集中的介绍,这是为了满足大多数用户的需求。对于一些很少用到的功能,读者可以参阅 Visual FoxPro 6.0 的有关技术资料,从而解决问题。

为注重应用能力的培养,本书在编写过程中,既注意内容的系统性,也注重实用性;既有通俗易懂的原理介绍,又有具体的操作指导。本书语言浅显易懂,叙述全面,图文并茂,实例生动,代码详细。各种实例均在 Visual FoxPro 6.0 系统下运行通过,运行结果正确。各章后面均配有一定量习题,供读者练习,以便巩固学习的知识。本书的电子教案也是用 Visual FoxPro 系统制作的,可以直接运行各例题中的代码,并显示运行结果。

赵忠孝教授对本书的整体结构及编写思路进行规划,并编写其中的第9、10、12章,参加编写的人员还有刘耀军(第1、3章)、梁晓玲(第5、6、7章,电子教案)、李蕾(第2、4、8章)、赵月爱(第13、14章,电子教案)、赵晔(第11章)等同志,全书最后由赵忠孝教授负责统稿和定稿。

穆晓芳、张朝霞、程小燕老师和部分同学参加了本书的校对工作,在此表示衷心的感谢。由于时间仓促和作者的水平所限,书中难免有不当之处,敬请广大读者指正。

赵忠孝

2004. 6. 25 于福州

E-mail: tysyzzx@public. ty. sx. cn

目 录

第1章 数据库系统概论	(1)
1.1 概述	(1)
1.1.1 信息、数据和数据处理	(1)
1.1.2 数据管理方式	(2)
1.1.3 数据库系统的主要特点	(3)
1.1.4 数据库系统的发展阶段	(5)
1.2 数据模型	(8)
1.2.1 3个世界	(8)
1.2.2 概念模型	(12)
1.2.3 数据模型	(13)
1.3 关系模型	(15)
1.3.1 基本概念	(15)
1.3.2 关系代数	(17)
1.3.3 函数依赖	(21)
1.3.4 关系完整性	(23)
1.3.5 关系范式	(24)
1.4 数据库系统的结构	(27)
1.4.1 数据库系统结构	(27)
1.4.2 数据库系统的模式结构	(28)
1.5 数据库设计	(30)
1.5.1 需求分析	(31)
1.5.2 概念设计	(34)
1.5.3 逻辑设计	(38)
1.5.4 物理设计	(41)
1.5.5 数据库的实现和维护	(43)
思考与练习	(44)
第2章 Visual FoxPro 6.0 系统概述	(48)
2.1 Visual FoxPro 特性	(48)
2.1.1 微机数据库的发展	(48)
2.1.2 Visual FoxPro 的主要特点	(48)
2.2 启动 Visual FoxPro 6.0 系统	(50)
2.2.1 运行环境	(50)

2.2.2 Visual FoxPro 系统的安装、启动和关闭	(50)
2.2.3 Visual FoxPro 的用户界面	(51)
2.3 Visual FoxPro 6.0 的工作方式	(54)
2.3.1 命令格式	(54)
2.3.2 两类工作方式	(56)
2.4 Visual FoxPro 6.0 的辅助设计工具	(56)
2.4.1 向导	(57)
2.4.2 设计器	(58)
2.4.3 生成器	(59)
2.5 项目管理器	(60)
2.5.1 创建一个项目	(60)
2.5.2 项目的组织结构	(61)
2.5.3 使用“数据”选项卡组织数据	(61)
2.5.4 为文件添加说明	(62)
2.5.5 项目间共享文件	(62)
2.6 使用帮助	(62)
2.6.1 一般帮助	(62)
2.6.2 从网络中取得帮助	(63)
思考与练习	(63)
第3章 数据元素	(66)
3.1 数值型数据	(66)
3.1.1 数值型常量	(66)
3.1.2 数值型变量	(67)
3.1.3 算术运算符	(68)
3.1.4 数值计算函数	(69)
3.1.5 算术表达式	(70)
3.2 字符型数据	(71)
3.2.1 字符型常量	(71)
3.2.2 字符型变量	(71)
3.2.3 字符运算符和表达式	(71)
3.2.4 字符串函数	(72)

3.3 日期时间型数据	(74)	4.2.9 成批记录与数组间的数据 传送	(124)
3.3.1 日期型常量和变量	(74)	4.3 打开多个表	(125)
3.3.2 日期运算符和表达式	(76)	4.3.1 工作区的选择	(125)
3.3.3 常用的日期和时间函数	(76)	4.3.2 多个表的打开与关闭	(126)
3.4 逻辑型数据	(77)	4.3.3 与工作区相关的函数	(127)
3.4.1 逻辑常量和变量	(77)	4.4 控制对记录和字段的访问	(128)
3.4.2 比较运算符	(77)	4.4.1 限制对记录的访问	(128)
3.4.3 复合逻辑表达式	(80)	4.4.2 限制对字段的访问	(130)
3.5 数组	(81)	4.5 文件操作	(132)
3.5.1 数组的定义和赋值	(81)	4.5.1 表文件的复制	(132)
3.5.2 数组函数	(83)	4.5.2 表文件与其他格式文件的数据 转换	(133)
3.6 系统变量	(85)	4.5.3 一般文件的复制	(135)
3.7 内存变量的操作	(86)	4.5.4 文件的重命名	(135)
3.7.1 内存变量的显示	(86)	4.5.5 文件的删除	(136)
3.7.2 内存变量的保存	(87)	思考与练习	(137)
3.7.3 内存变量的释放	(88)	第5章 排序与索引	(140)
3.7.4 内存变量的恢复	(88)	5.1 排序	(140)
思考与练习	(88)	5.2 索引的概念	(141)
第4章 表操作	(91)	5.3 索引类型	(142)
4.1 创建表	(91)	5.4 索引文件的创建	(143)
4.1.1 表结构	(91)	5.4.1 使用表设计器	(143)
4.1.2 创建表的方式	(93)	5.4.2 创建索引的命令	(146)
4.1.3 创建表时立即输入数据	(100)	5.5 索引的使用	(148)
4.1.4 数据表的打开与关闭	(103)	5.5.1 索引文件的打开与关闭	(148)
4.1.5 表结构操作	(105)	5.5.2 设置主控索引	(151)
4.1.6 记录指针	(107)	5.5.3 重新索引	(152)
4.1.7 字段变量	(109)	5.6 索引查询	(152)
4.1.8 Visual FoxPro 命令的常用 子句	(109)	5.6.1 顺序查询命令	(152)
4.1.9 Visual FoxPro 环境设置	(112)	5.6.2 索引查询	(154)
4.2 记录的操作	(114)	思考与练习	(156)
4.2.1 插入记录	(114)	第6章 创建数据库	(158)
4.2.2 追加记录	(115)	6.1 数据库设计的过程	(159)
4.2.3 记录的逻辑删除	(116)	6.2 创建和删除数据库	(159)
4.2.4 恢复记录	(117)	6.2.1 建立数据库	(159)
4.2.5 记录的物理删除	(117)	6.2.2 打开数据库	(160)
4.2.6 编辑和浏览记录	(118)	6.2.3 在数据库中创建、添加和 移去表	(161)
4.2.7 替换记录	(122)	6.2.4 删除数据库	(164)
4.2.8 单条记录与数组间的数据 传送	(123)		

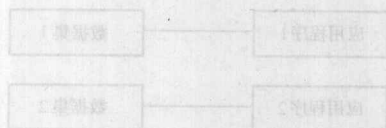
6.3 数据字典·····	(164)	7.7 创建交叉表查询·····	(208)
6.3.1 设置长表名和表的注释·····	(164)	7.8 运行查询·····	(211)
6.3.2 设置字段标题和注释·····	(165)	思考与练习·····	(212)
6.3.3 设置字段的有效性规则·····	(167)	第8章 视图 ·····	(215)
6.3.4 字段值的格式码和输入掩码·····	(168)	8.1 创建视图·····	(215)
6.3.5 数据记录的有效性验证·····	(170)	8.1.1 利用视图设计器创建视图·····	(215)
6.4 建立表间的关系·····	(171)	8.1.2 命令方式创建视图·····	(217)
6.4.1 建立表间的永久关系·····	(171)	8.1.3 视图的修改与删除·····	(219)
6.4.2 建立表间的临时关系·····	(173)	8.2 使用视图·····	(220)
6.4.3 设置参照完整性·····	(174)	8.2.1 打开视图·····	(220)
6.5 引用多个数据库·····	(176)	8.2.2 创建视图索引·····	(221)
6.5.1 打开多个数据库·····	(176)	8.2.3 为视图字段创建默认值·····	(221)
6.5.2 数据库中表的使用·····	(177)	思考与练习·····	(222)
6.5.3 浏览数据库文件·····	(177)	第9章 程序设计基础 ·····	(224)
6.5.4 数据库检查·····	(178)	9.1 程序的编辑·····	(224)
6.5.5 关闭数据库·····	(179)	9.1.1 程序文件的创建、打开·····	(224)
思考与练习·····	(180)	9.1.2 程序文件的保存·····	(225)
第7章 结构化查询语言 ·····	(182)	9.1.3 创建良好的编辑环境·····	(226)
7.1 SQL 语句创建查询·····	(182)	9.2 程序文件的运行·····	(229)
7.2 单表查询·····	(183)	9.2.1 以命令方式运行·····	(229)
7.2.1 选择表中的若干列·····	(183)	9.2.2 以菜单方式运行·····	(230)
7.2.2 选择表中的若干记录·····	(184)	9.2.3 程序执行的终止·····	(231)
7.2.3 嵌套查询·····	(187)	9.3 基本输入语句·····	(231)
7.2.4 对查询结果的排序和分组·····	(190)	9.3.1 字符串输入语句·····	(231)
7.2.5 保存查询结果·····	(191)	9.3.2 表达式输入语句·····	(232)
7.3 多表连接查询·····	(192)	9.3.3 等待输入字符语句·····	(233)
7.3.1 等值连接查询·····	(193)	9.4 程序的基本结构·····	(233)
7.3.2 自身连接查询·····	(194)	9.4.1 程序设计基础·····	(233)
7.3.3 外连接查询·····	(194)	9.4.2 程序的基本结构·····	(234)
7.4 创建查询文件·····	(195)	9.5 选择结构·····	(235)
7.4.1 查询设计器·····	(196)	9.5.1 条件选择结构·····	(235)
7.4.2 使用查询向导建立查询·····	(199)	9.5.2 多分支选择结构·····	(237)
7.5 修改查询文件·····	(203)	9.6 循环结构·····	(238)
7.5.1 使用查询设计器·····	(203)	9.6.1 DO WHILE 循环结构·····	(238)
7.5.2 使用命令·····	(203)	9.6.2 FOR 循环结构·····	(240)
7.6 查询结果的输出·····	(204)	9.6.3 SCAN 循环结构·····	(242)
7.6.1 查询去向对话框·····	(204)	9.6.4 多重循环·····	(243)
7.6.2 将查询结果以图形方式输出·····	(205)	9.6.5 循环小结·····	(245)
7.6.3 在屏幕上输出·····	(207)	9.7 预处理语句·····	(246)
7.6.4 以报表方式输出·····	(208)	9.7.1 常量定义和释放·····	(246)

9.7.2 文件包含	(248)	11.3.2 创建多表表单	(297)
9.7.3 条件编译	(249)	11.4 表单设计器的使用	(299)
9.8 多模块程序设计	(250)	11.4.1 设置 Tab 键次序	(300)
9.8.1 主程序和子程序	(250)	11.4.2 布局工具栏	(300)
9.8.2 参数传递	(251)	11.5 设置数据环境	(301)
9.8.3 过程	(253)	11.5.1 打开数据环境设计器	(301)
9.8.4 自定义函数	(257)	11.5.2 向数据环境设计器添加表或 视图	(302)
9.9 变量的作用域	(260)	11.5.3 数据环境的属性	(302)
9.9.1 公有变量	(260)	11.5.4 在数据环境设计器中设置 关系	(303)
9.9.2 私有变量	(261)	11.5.5 从数据环境设计器中移去表或 视图	(303)
9.9.3 局部变量	(261)	11.6 表单的事件和方法	(304)
9.9.4 应用举例	(262)	11.6.1 表单中常用的事件	(304)
9.10 程序设计一般问题	(263)	11.6.2 方法程序的调用	(305)
9.10.1 程序设计一般步骤	(263)	11.6.3 事件中的参数	(307)
9.10.2 减少程序中的语法错误	(264)	11.7 向表单添加控件	(309)
9.10.3 提高程序的可维护性	(264)	11.8 输出控件	(309)
9.11 程序调试工具	(265)	11.8.1 标签	(309)
9.12 小结	(267)	11.8.2 图像	(310)
思考与练习	(267)	11.8.3 线条	(310)
第 10 章 面向对象的程序设计	(275)	11.8.4 形状	(311)
10.1 面向对象的新概念	(275)	11.9 输入控件	(312)
10.1.1 对象	(275)	11.9.1 文本框	(312)
10.1.2 类	(278)	11.9.2 编辑框	(317)
10.2 类的创建和编辑	(279)	11.9.3 列表框和下拉列表框	(319)
10.2.1 使用菜单创建类	(279)	11.9.4 微调控件	(325)
10.2.2 使用命令创建类	(283)	11.10 控制控件	(327)
10.2.3 编辑属性和方法程序	(283)	11.10.1 命令按钮	(327)
10.3 对象的操作	(285)	11.10.2 命令按钮组	(329)
10.3.1 对象的创建	(285)	11.10.3 复选框	(332)
10.3.2 设置对象的属性值	(286)	11.10.4 选项按钮组	(332)
10.3.3 对象的引用	(287)	11.10.5 计时器	(333)
思考与练习	(288)	11.11 容器控件	(336)
第 11 章 表单设计与应用	(290)	11.11.1 表格	(336)
11.1 表单的构成	(290)	11.11.2 页框	(340)
11.2 使用表单设计器创建表单	(291)	11.11.3 容器	(341)
11.2.1 表单的创建	(291)	11.12 连接控件	(341)
11.2.2 表单的属性设置	(292)	11.12.1 ActiveX 控件	(342)
11.2.3 表单的保存与打开	(294)		
11.3 使用向导创建表单	(294)		
11.3.1 创建单表表单	(294)		

11.12.2 ActiveX 绑定控件	(344)	12.7 预览和打印	(380)
11.12.3 超级链接	(345)	12.7.1 预览	(380)
11.13 表单的运行	(346)	12.7.2 打印报表	(380)
11.14 表单集	(347)	思考与练习	(381)
11.14.1 表单集的创建和打开	(348)	第 13 章 菜单设计	(383)
11.14.2 表单集的操作	(348)	13.1 菜单设计基础	(383)
11.14.3 表单集的运行、释放和 删除	(349)	13.1.1 菜单的基本概念	(383)
思考与练习	(350)	13.1.2 打开菜单设计器窗口	(384)
第 12 章 创建报表和标签	(353)	13.2 设计菜单	(385)
12.1 报表布局	(353)	13.2.1 菜单设计器窗口	(385)
12.1.1 报表布局类型	(353)	13.2.2 “显示”菜单的命令	(389)
12.1.2 报表设计器	(354)	13.3 运行菜单	(393)
12.2 使用报表向导创建报表	(356)	13.3.1 保存菜单定义	(393)
12.2.1 报表向导	(356)	13.3.2 生成菜单程序	(393)
12.2.2 一对多报表向导	(359)	13.3.3 运行菜单程序	(394)
12.2.3 快速报表	(360)	13.4 快捷菜单	(394)
12.3 报表中的数据组成	(361)	思考与练习	(397)
12.3.1 设置报表数据源	(361)	第 14 章 应用程序的管理与发布	(398)
12.3.2 报表变量	(362)	14.1 应用程序管理	(398)
12.3.3 按布局分组数据	(363)	14.1.1 建立项目	(398)
12.4 添加报表控件	(364)	14.1.2 设置应用程序的主文件	(398)
12.4.1 标签控件	(364)	14.1.3 建立一个简单的主程序	(399)
12.4.2 线条控件	(365)	14.1.4 文件的包含和排除	(401)
12.4.3 域控件	(366)	14.1.5 应用程序的连编	(402)
12.4.4 矩形控件	(369)	14.2 应用程序的发布	(403)
12.4.5 圆角矩形和圆形	(369)	14.2.1 发布准备	(403)
12.4.6 图片控件	(369)	14.2.2 创建发布磁盘	(404)
12.4.7 调整控件的位置	(371)	思考与练习	(409)
12.5 页面设置	(373)	附录	(411)
12.6 创建标签	(374)	附录 A 文件扩展名与文件类型	(411)
12.6.1 使用标签向导创建标签	(374)	附录 B Visual FoxPro 系统容量	(412)
12.6.2 标签设计器	(378)	参考文献	(414)

第1章

数据库系统概论



数据库技术、网络技术和多媒体技术是当今计算机科学中最活跃、发展最快的分支。今天,信息资源已成为各国、各个部门的重要财富和资源。占有信息,并能对信息进行有效地管理和加工,将成为一个企业或组织生存和发展的重要条件。因此,作为信息系统核心和基础的数据库技术得到了越来越广泛的应用,从小型单项事务处理系统到大型信息系统,从联机事务处理到联机分析处理,越来越多新的应用领域采用数据库存储和处理信息资源。从近几年来的海湾战争到伊拉克战争,打的就是信息战。数据库的建设规模,数据库信息量的大小和使用频度已成为衡量这个国家信息化程度的重要标志,也是一个国家生存和发展的重要手段。因此,数据库课程是计算机科学与技术专业、信息管理等相关专业的重要课程。

1.1 概述

1.1.1 信息、数据和数据处理

人类赖以生存的世界是一个物质的世界,所有物质形成一个物质流。同时人们也是生活在一个信息的世界中,所有的信息构成一个信息流。信息是关于客观世界各种事物变化和物质特征的反映,是经过加工后提炼形成的可通信的知识。物质的存在伴随着信息的存在,物质的变化会引起信息的变化。

现今的人类社会正在进入信息化的社会,人们在政治、经济、军事、文化、教育、科学、艺术等各种活动中都将产生大量的信息。信息需要被处理、加工、交流和使用,人们用各种物理符号或符号组合来记录信息,记录下来的信息称为数据。因此,信息是观念性的,而数据是物理性的。但是,人们通常认为信息和数据是同一概念,数据处理也称为信息处理。

所谓数据处理包括对数据的收集、记载、分类、排序、存储、计算或加工、传输等工作。人们收集到的各种数据需要经过处理加工才能变为有用的信息。利用计算机进行数据管理经历了自由管理方式、文件管理方式和数据库管理方式3个阶段。

1.1.2 数据管理方式

1. 自由管理方式

早期的计算机主要用于科学计算,当时没有操作系统,也没有管理数据的软件。数据的处理方式是批处理,数据和程序的对应关系如图 1.1 所示。



图 1.1 自由管理方式下程序和数据
的对应关系

自由管理方式的特点如下:

(1) 数据不保存

由于当时的计算机主要用于科学计算,一般不需要将数据长期保存,只是在计算某个课题时输入,程序运行结束,数据也随之撤销。

(2) 应用程序管理数据

应用程序自行管理所使用的数据,没有其他的软件系统来协助管理数据,数据的结构、格式、输入/输出方式、存取方式等必须在程序中设定。程序员需要把大量的时间和精力花费在数据的存储和管理上。

(3) 数据不能共享

由于数据和程序交织在一起,一个程序中使用的数据,当程序运行结束时,数据也就被清除。即使其他程序也使用同样的数据,也不能共享,增加了数据的冗余。

(4) 数据不具有独立性

数据的独立性是指数据和应用程序是相互独立的。也就是说,当数据发生变化时程序不一定要修改,或修改程序后数据也不一定要修改。在自由管理方式下,程序的修改必然引起对数据的修改;反之,数据要修改也必须修改程序。

2. 文件管理方式

由于自由管理方式存在不少缺点,人们总是想把数据和程序分开,分别进行单独管理,这可以说是计算机数据处理史上的第一个里程碑。人们可将程序存储到程序文件中,而将处理的数据存储在数据文件中。数据文件的一个元素称为记录或元组。记录是由一些数据项组成的,每个数据项表示了事物的一个特征。例如,学籍管理中的学号、姓名、年龄、性别等就是数据项,而对应记录格式的一组值就是一条记录,全部记录的集合就是一个文件,即学生文件。

数据组成文件以后就独立存在于程序之外了。每个文件有一个文件名,它存储在外存储器上。

管理文件的软件系统称为文件管理系统,用户可通过文件名在程序中逻辑地调用它,但文件的物理存储用户程序就不用过问了。

文件管理系统是应用程序和数据文件之间的一个接口,如图 1.2 所示。

文件管理方式最大的优点是数据与程序的分离,这才使得以后的数据处理技术得以独立迅速地发展。有了文件系统,省去了用户对数据进行具体存取操作的麻烦,可以逻辑地通过文件名来使用数据。有了文件系统后,数据处理进入了兴旺时期,计算机的应用重点由科学计算转到了数据处理,它推动了管理信息技术的发展。文件管理方式虽然大大提高了数据处理的效率,但它在使用中仍暴露出许多弊病,例如:

(1) 数据与程序间仍然有着十分密切的依赖关系,也就是数据独立性差。

(2) 数据文件的专用性、分散性使得文件不能为多个用户所共享。

(3) 数据重复存储,很难做到数据的一致性。

文件系统本身已无法解决这些问题,必须由一种新技术来取代它,这就是数据库技术。

3. 数据库管理方式

数据库(DataBase, DB)、数据库管理系统(DataBase Management System, DBMS)、数据库系统(DataBase System, DBS)是数据库技术中使用的术语,三者之间有一定的联系和区别。

所谓数据库,一般地说是数据的集合,形象地说是存储数据的“仓库”。但仅仅有了大批数据是没有多少意义的,还应当有一个维护数据并管理用户访问数据的机构。以图书馆作为例子,众所周知,图书馆是存储图书和管理图书借阅的部门。书库是各类图书的集合,不能简单地将图书馆和书库等同起来。图书馆若要很好地为读者服务,首先要有图书馆管理员收集图书并对图书建立完善的书卡。书卡的内容通常包括书号、书名、作者名、出版单位、出版时间、内容摘要和其他项目等;其次,按一定的顺序和规则(物理结构)分别存放不同类别的图书;最后规定图书的借还手续,即管理员对读者访问的响应过程。这一整套图书管理功能就相当于数据库管理系统的功能。

数据库是存储在计算机内的有结构的数据集合。数据库管理系统是个数据库管理软件,它的职能是维护数据库,接受和完成由用户程序或命令提出的访问数据的要求,而数据库系统是指计算机系统引进数据库后的系统结构。一般数据库系统由计算机硬件、数据库、数据库管理系统和用户构成。

1.1.3 数据库系统的主要特点

1. 数据具有最小的冗余度

文件的最简单形式是可以存储等长、相同格式记录的集合。这种形式虽然简单,但存在

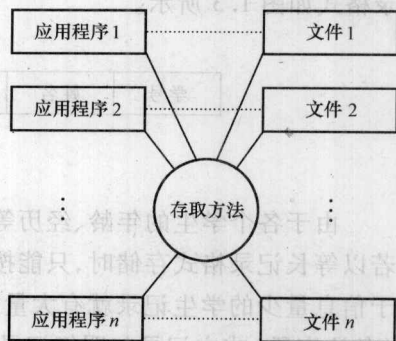


图 1.2 文件管理方式下程序和文件之间的对应关系

大量的数据冗余。例如,一名学生的人事文件,每一记录记载一名学生的全部人事信息,记录格式如图 1.3 所示。

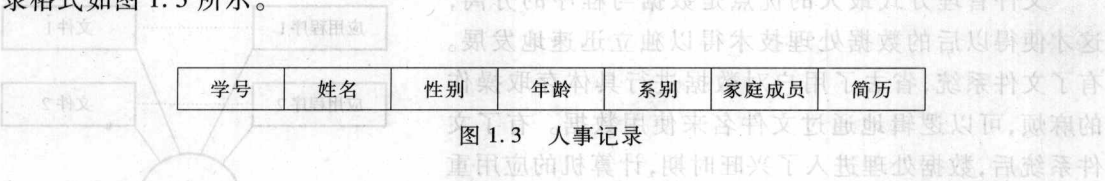


图 1.3 人事记录

由于各个学生的年龄、经历等状况不同,在简历栏内所具有的信息量的多少差异很大。若以等长记录格式存储时,只能按信息量最大的学生需要来设计存储空间的大小。显然,对于信息量少的学生记录就有大量的空白,会造成很大浪费。在数据库中,则改用不等长记录(备注字段)或主记录与明细记录相结合方法。在人事记录中,将任何学生均具有而且是基本上是等长的那部分作为基本信息,称为主记录;其余信息量变较大的部分则按某种原则划分为若干个等长的记录。例如,一个人的简历分为若干个小的等长记录存储在简历文件中,在主文件中只需记下该学生简历在简历文件中位置即可。一条主记录可带上几种各有若干个值的明细记录,该方法增加了灵活性,减少了存储空间浪费,如图 1.4 所示。

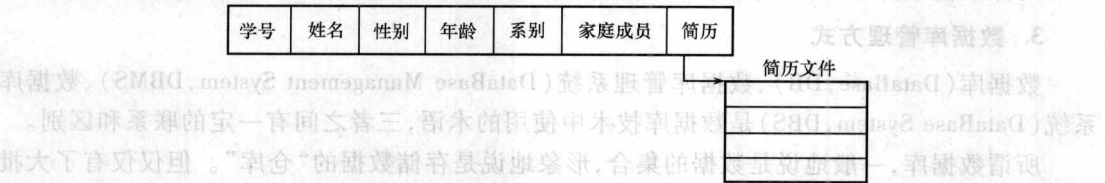


图 1.4 主记录和明细记录之间的联系

另一方面,一个单位往往包括多种应用系统。例如,除人事管理系统外,还有财务、工资、后勤系统等。从整体观点来看,不仅要考虑一个应用系统组的数据结构问题,而且要考虑整个单位的数据结构问题。某些数据是各部门共同使用的,不需要各自存储这部分数据。这样就把整个单位的数据结构化了。这就要求在描述数据时不仅描述数据本身,还要描述数据之间的联系,这种联系是通过存取路径来实现的。通过存取路径来表示自然的数据联系是数据库与传统文件的根本区别。

在对数据结构化和建立相应的联系之后,就减少了每种记录与主记录内容重复的数据,因而大大减少数据的冗余度,节约了空间,减少了存取时间,避免了数据的不相容和数据冗余所引起数据不一致性。此外,由于是从整体观点看数据,数据不再是面向某个应用,而是面向整个系统的。这样,它的弹性大,可以扩充好,应用数据时就可以有很大的灵活方式。

2. 具有共享性

数据库中的数据可以进行各种组合,以最优的方式去满足各个用户应用的需要。

3. 有较高的数据独立性

数据与使用数据的程序是彼此独立的。这种独立性有两个方面的含义:物理独立性和逻辑独立性。所谓物理独立性是指,当数据存放方式改变时,由于不改变数据的全局逻辑结构,因此不用改写程序;而逻辑独立性则是指,当数据的全局逻辑结构改变时,由于不改变某些局部的逻辑结构,一般程序只和局部的逻辑结构有关,因此,程序也不用改写。

4. 有统一管理与控制

为用户提供存储、检索、更新数据的手段,以及并发使用数据库,保证数据的安全性、完整性、保密性,数据库系统提供了统一的管理软件——数据库管理系统进行管理和控制。其功能主要表现在以下几个方面:

(1) 数据的安全性(Security)控制。数据的安全性是指保护数据以防止不合法的使用所造成数据的泄漏和破坏。例如,学生成绩数据库中,只有有修改权的用户才能对此数据库的内容进行修改,其他用户只能读取全部或部分数据。

(2) 数据的完整性(Integrity)约束。数据的完整性是指数据的正确性、有效性和相容性。在输入、修改数据库的过程中,要始终符合原来数据的定义和规定。例如,性别只能是男或女,学号必须是惟一的,一年只能有12个月等。

(3) 并发控制(Concurrency)。当多个用户同时存取、修改数据库中的数据时,可能会发生相互干扰而使数据库的完整性得到破坏。例如,现只有一张去北京的火车票,却同时被两个售票口售出。为防止这种现象发生,必须进行并发控制。

(4) 数据库的恢复(Recovery)。数据库系统虽然在数据管理上有较大地改进,但它仍不能永远正确地工作。因此,当数据库在运行时发生硬件或软件故障时,数据库系统能将数据库恢复到最近某个时刻的正确状态。

综上所述,数据库是个通用化的、综合性的数据集合,可以提供给各用户共享,而且有最小冗余度,并具有较高的程序和数据的独立性。DBMS在数据库建立、运用和维护时对数据库进行统一控制,以保证数据的完整性、安全性,并在多用户同时使用数据库时进行并发控制,在发生故障后对系统进行恢复。

1.1.4 数据库系统的发展阶段

数据模型是数据库技术的核心和基础,故数据库系统的发展阶段的划分通常以数据模型的演变作为主要的依据。按照数据模型的发展和演变过程,数据库技术从开始到现在的30多年中,主要经历了3个发展阶段:层次和网状数据库系统、关系数据库系统、面向对象的数据库系统。数据库技术与网络通信技术、人工智能技术、并行计算技术等相互渗透,互相融合,新一代网络数据库系统也就应运而生。下面分别介绍这些数据库系统的特征。

1. 第一代数据库系统

第一代数据库技术是20世纪70年代研制的层次和网状数据库系统。层次数据库系统

的代表是1969年IBM公司研制的层次数据库管理系统IMS。20世纪60年代末70年代初,美国数据系统语言联合会(Conference on Data System Language, CODASYL)下属的数据库任务组(Data Base Task Group, DBTG)提出了若干报告,被称为DBTG报告。DBTG报告确定并建立了网状数据库系统的许多概念、方法和技术,是网状数据库的典型代表。在DBTG思想和方法的指引下,数据库系统的实现技术不断成熟,并开发出了许多商品化的数据库系统。

可以说,层次数据库是数据库的先驱,而网状数据库则是数据库的概念、方法和技术的基础。

2. 第二代数据库系统

第二代数据库技术是关系数据库系统,1970年IBM公司的San Jose实验室的研究员E. F. Code在美国计算机学会会刊《Communication of the ACM》上发表《大型共享数据库的关系模型》(A Relational Model of Data for Shared Data Banks)的论文,提出了关系数据库模型,为关系数据库理论奠定了基础。为表彰E. F. Code对关系数据库理论的贡献,1981年他被授予ACM图灵奖。

20世纪70年代是关系数据库理论研究发展的时代,IBM公司的San Jose实验室在IBM 370系列机上研制的关系数据库系统System R获得成功。同时,美国加州大学伯克利分校也研制了Ingres关系数据库系统。此间,System R和Ingres关系数据库软件产品被推向市场。

30多年来,关系数据库系统的研究取得了辉煌的成就,涌现了许多性能良好的关系数据库系统,如DB2、Oracle、Ingres、Sybase、Informix、Visual FoxPro等,本书将重点介绍关系数据库系统。

3. 第三代数据库系统

从20世纪80年代以来,数据库技术在商业上应用的巨大成功刺激了其他领域对数据库技术需求的迅速增长。1990年高级DBMS功能委员会发表了《第三代数据库系统宣言》,提出了第三代数据库系统应具有3个基本特征:

- 支持数据管理、对象管理和知识管理。
- 必须保持或继承第二代数据库系统技术。
- 必须对其他系统开放。

面向对象的数据模型是第三代数据库系统的主要特征之一,数据库技术与多学科技术的有机结合也是第三代数据库技术的重要特性。例如,数据库技术与网络技术、Web技术、分布式技术、并行技术的结合,形成了网络数据库系统、Web数据库系统、分布式数据库技术和并行数据库系统。

(1) 网络数据库系统

计算机网络技术的迅速发展,以及地理上分散的用户对数据库应用的需求,使数据库技术从单机环境走向网络环境,从封闭式走向开放式。网络数据库技术是基于客户机/服务器结构的数据库技术。