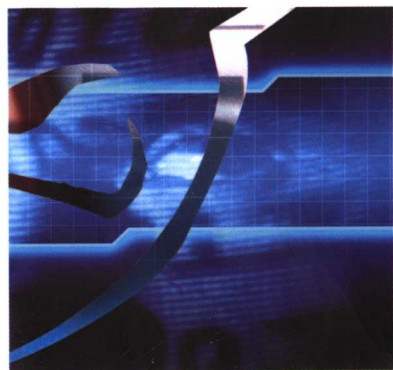
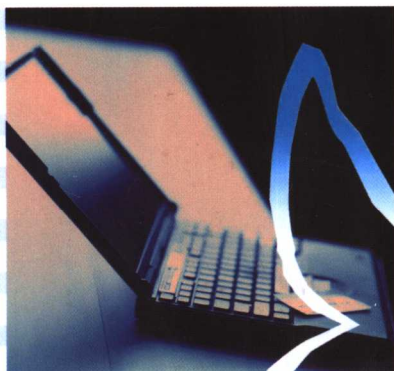




21 世纪高等学校应用型教材

Visual FoxPro 程序设计

□ 李淑华 主编



高等教育出版社
Higher Education Press

图书在版编目(CIP)数据

Visual FoxPro程序设计 / 李淑华主编. —北京: 高等教育出版社, 2004.8 (2005 重印)
ISBN 7-04-015513-3

I.V... II.李... III.关系数据库—数据库管理系统, Visual FoxPro—程序设计 IV.TP311.138

中国版本图书馆CIP数据核字(2004)第075036号

策划编辑 雷顺加 责任编辑 雷顺加 市场策划 韩 飞
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	购书热线	010-58581118
社 址	北京市西城区德外大街 4 号	免费咨询	800-810-0598
邮政编码	100011	网 址	http://www.hep.edu.cn
总 机	010-58581000		http://www.hep.com.cn
		网上订购	http://www.landaco.com
			http://www.landaco.com.cn
经 销	北京蓝色畅想图书发行有限公司		
印 刷	涿州市星河印刷有限公司		
开 本	787×1092 1/16	版 次	2004 年 8 月第 1 版
印 张	23.5	印 次	2005 年 12 月第 4 次印刷
字 数	570 000	定 价	29.00 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

物料号 15513-00

内 容 提 要

本书共 14 章,主要介绍了 Visual FoxPro 6.0 系统的基础知识;讲述了项目管理器的使用;较详细地介绍了数据库的建立和表的有关操作、数据库管理的操作,以及查询和视图、程序设计的常用命令和基本结构、报表和标签设计的方法等;简单介绍了面向对象程序设计的基本概念;详细介绍了常用控件的程序设计、数据表的表单程序设计、菜单等设计方法;结合工资管理系统设计实例介绍了用 Visual FoxPro 开发应用项目的全过程;介绍了 SQL 语言和常用函数的使用。本书有大量测试题与上机操作题,配有电子教案、CAI 课件,并编写了与本教材配套的《Visual FoxPro 程序设计学习指导与实训》。

本书由浅入深、通俗易懂,可作为各类高等学校数据库应用课程的教材,也可供参加全国高等学校计算机考试二级 Visual FoxPro 程序设计的考生学习使用,同时也适于各类管理人员学习参考。

书中所有程序设计例题、程序开发实例的代码全部调试通过,其代码及程序设计相关文件、电子教案均能从高等教育出版社的网站(<http://cs.hep.edu.cn> 或 <http://www.hep-st.com.cn>)下载。要获取本书的 CAI 课件及相关辅助资料,请与作者联系。联系电话:024-24520341。

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)58581118

前 言

随着计算机技术的发展和普及,各行各业的管理部门需要由计算机处理大量的信息。选择一个优秀的数据库管理系统作为开发平台,会给信息处理带来极大的方便。Visual FoxPro 6.0 数据库是一个关系型数据库。利用该数据库可以设计出丰富多彩的用户界面,在用户界面中可以放置各种控件。它能够管理大量复杂的数据信息,同时具有很好的安全性和较强的网络功能,能够实现数据的远程访问和存储加工。为了满足各高校应用型人才培养的教学需要,我们编写了本书及与其配套的《Visual FoxPro 程序设计学习指导与实训》。

本书在编写过程中着重突出了以下特点:

第一,强调应用性。本书在保持知识系统性的同时,突出应用性。在整体结构和素材的选择上,特别注意实际应用,以满足学生学习和工作的需要。书中各章附有大量测试题与上机操作题,可供学生思考和上机操作训练时使用和参考。

第二,提高学生的学习兴趣。本书配有 CAI 课件。该课程充分利用计算机特有的图形、声音、色彩等功能,同时采用动画模拟、局部放大等技术手段,向学生提供图文并茂、有声有色的感性素材,可使抽象的内容具体化,微观的内容形象化,开阔学生的视野,激发学生的学习兴趣,使学生产生学习动力,达到良好的学习效果。

第三,注意培养学生的能力。学生通过屏幕上生动的演示来理解和掌握抽象的概念及一些复杂过程,较好地实现直观感觉与形象思维和抽象思维之间的过渡,有助于培养学生的想象能力、思维能力、自学能力和操作能力。

第四,重点介绍面向对象程序设计方法。从常用控件的一般使用方法入手,到数据表单的设计方法,最后介绍具有代表性的工资管理系统设计实例,体现了知识循序渐进的规律。

第五,兼顾全国计算机等级考试,内容涵盖了等级考试新大纲所要求的基本知识点。

书中程序设计例题、程序开发实例的代码全部调试通过,其代码及程序设计相关文件均能从高等教育出版社的网站(<http://cs.hep.edu.cn> 或 <http://www.hep-st.com.cn>)下载。要获取本书的 CAI 课件及相关辅助资料,请与作者联系。电话:024-24520341。

本书由李淑华主编,负责整体结构的设计,并编写第 1~11 章、第 13 章、第 14 章;张翼英任副主编,并编写第 12 章;张鹏、张丕振负责 CAI 课件的制作。

尽管在编写此书的过程中作者做了许多的努力,但由于水平有限,加之编写时间仓促,书中缺点和疏漏之处难免,敬请读者批评指正。

编 者

2004 年 8 月

目 录

第 1 章 Visual FoxPro 6.0 概述	(1)	2.3.3 使用项目管理器管理项目	(22)
1.1 数据库的基本概念	(1)	2.3.4 在项目管理器中操作文件	(24)
1.1.1 数据与数据处理	(1)	2.3.5 项目管理器的其他操作	(25)
1.1.2 数据库的产生	(1)	2.4 设计器与生成器	(27)
1.1.3 数据库系统	(2)	2.4.1 设计器与工具栏	(27)
1.2 数据模型	(3)	2.4.2 生成器	(28)
1.2.1 基本概念	(3)	2.5 建立工作目录与搜索路径	(28)
1.2.2 实体之间的联系	(4)	习题	(30)
1.2.3 数据模型简介	(4)	第 3 章 数据库的建立和操作	(32)
1.3 关系数据库	(5)	3.1 数据库的概念	(32)
1.3.1 基本概念	(5)	3.1.1 表的概念	(32)
1.3.2 数据完整性	(6)	3.1.2 表的字段	(33)
1.3.3 对关系数据库的要求	(7)	3.2 数据库的建立	(35)
1.3.4 关系运算	(7)	3.2.1 建立数据库的准备	(35)
1.3.5 关系运算的优化	(8)	3.2.2 数据库建立	(35)
1.4 VFP 的发展过程	(8)	3.3 建立自由表	(38)
1.5 VFP 的功能	(9)	3.3.1 创建表	(38)
1.5.1 VFP 的基本功能	(9)	3.3.2 向表中添加数据	(40)
1.5.2 VFP 的特点	(10)	3.3.3 修改表结构	(40)
1.6 VFP 系统的环境与安装	(12)	3.4 维护数据库表中的记录	(41)
1.6.1 VFP 系统环境	(12)	3.4.1 浏览表中的数据	(41)
1.6.2 VFP 系统的安装	(12)	3.4.2 编辑、修改记录中的字段	(42)
1.7 VFP 系统的启动	(13)	3.4.3 删除和还原记录	(43)
1.7.1 启动 VFP	(13)	3.4.4 定制浏览数据窗口	(44)
1.7.2 退出 VFP	(13)	3.5 筛选表记录	(45)
1.8 VFP 环境介绍	(14)	3.5.1 用过滤器限制记录	(45)
1.8.1 主窗口介绍	(14)	3.5.2 用过滤器限制字段	(46)
1.8.2 用 VFP 开发应用程序的方式	(15)	3.6 建立索引	(46)
1.8.3 帮助	(16)	3.6.1 索引概念	(46)
习题	(16)	3.6.2 索引的建立	(47)
第 2 章 VFP 基础	(18)	3.6.3 用索引给表排序	(48)
2.1 VFP 中文版的性能指标	(18)	3.6.4 使用索引应遵循的原则	(49)
2.2 VFP 文件组成	(19)	3.6.5 对多个字段排序	(49)
2.3 项目管理器	(20)	3.6.6 筛选记录	(49)
2.3.1 项目管理器的功能	(21)	习题	(49)
2.3.2 项目管理器的组成	(21)		

第4章 数据库的管理	(51)	5.4.3 为视图添加筛选表达式	(85)
4.1 数据库的设计	(51)	5.4.4 建立远程数据连接	(86)
4.1.1 数据库的设计过程	(51)	5.4.5 建立远程视图	(87)
4.1.2 数据库的简单操作	(52)	5.4.6 用视图更新数据	(87)
4.2 数据库中表的操作	(53)	5.4.7 控制更新数据的条件	(88)
4.2.1 把自由表添加到数据库中	(53)	5.4.8 控制视图更新的方法	(89)
4.2.2 删除数据库中的表	(54)	5.4.9 为视图传递参数	(89)
4.2.3 在数据库中查找表	(55)	习题	(90)
4.3 字段与记录属性设置	(55)	第6章 VFP 应用程序设计结构	(91)
4.3.1 字段与记录属性的常用概念	(55)	6.1 数据及其运算	(91)
4.3.2 设置字段的显示属性	(56)	6.1.1 常量	(91)
4.3.3 输入字段的注释	(57)	6.1.2 变量	(94)
4.3.4 “表达式生成器”对话框	(58)	6.1.3 数组	(97)
4.3.5 字段有效性	(60)	6.1.4 函数	(98)
4.3.6 记录有效性	(62)	6.1.5 表达式	(99)
4.3.7 使用长表名与注释	(64)	6.1.6 命令格式	(102)
4.3.8 使用长字段名、标题与注释	(64)	6.1.7 命令书写的规则	(103)
4.4 创建和编辑关系	(65)	6.2 数据库的操作命令	(103)
4.4.1 建立、编辑永久性关系	(66)	6.2.1 数据库操作命令	(103)
4.4.2 建立、编辑临时性关系	(67)	6.2.2 表的操作命令	(105)
4.5 建立参照完整性	(69)	6.2.3 记录指针定位	(108)
4.6 使用多个数据库	(71)	6.2.4 记录的显示	(109)
习题	(71)	6.3 文件操作	(110)
第5章 查询与视图	(73)	6.3.1 复制文件	(110)
5.1 查询与视图	(73)	6.3.2 显示文件目录	(112)
5.1.1 查询	(73)	6.3.3 修改文件名	(112)
5.1.2 视图	(73)	6.3.4 删除文件	(112)
5.1.3 查询与视图的区别	(73)	6.3.5 表之间的数据传送	(113)
5.2 结构化查询语言 SQL 简介	(74)	6.4 表的修改和维护	(113)
5.2.1 SQL 简介	(74)	6.4.1 记录的修改	(113)
5.2.2 SQL 的格式	(74)	6.4.2 记录的插入与删除	(114)
5.2.3 SQL 命令使用举例	(75)	6.5 表的排序与索引	(116)
5.2.4 SQL 语句在 VFP 中的 使用方法	(76)	6.5.1 表的排序	(116)
5.3 查询数据	(76)	6.5.2 索引文件	(117)
5.3.1 建立查询	(77)	6.5.3 数据检索	(119)
5.3.2 为查询结果排序	(80)	6.5.4 顺序查找命令(LOCATE 与 CONTINUE)	(120)
5.3.3 筛选查询结果	(81)	6.5.5 过滤器命令 (SET FILTER TO)	(121)
5.3.4 查询结果的分组	(81)	6.6 统计命令	(121)
5.4 视图查询	(82)	6.6.1 求和命令(SUM)	(121)
5.4.1 视图文件的建立	(83)	6.6.2 求平均值命令(AVERAGE)	(122)
5.4.2 控制视图字段的显示与输入	(84)		

6.6.3 计数命令(COUNT)	(122)	7.3.3 调整报表带区	(156)
6.6.4 求统计量命令 (CALCULATE)	(122)	7.3.4 报表控件的使用	(157)
6.6.5 分类汇总命令	(123)	7.3.5 定义报表的页面	(159)
6.7 使用多个表	(124)	7.4 标签文件的建立	(161)
6.7.1 工作区	(124)	7.4.1 用“标签向导”创建标签	(161)
6.7.2 设置表间的临时关系	(127)	7.4.2 用“标签设计器”创建标签	(162)
6.8 命令文件的建立与运行	(128)	7.5 预览和打印报表与标签	(162)
6.8.1 命令文件的建立	(128)	7.5.1 预览报表和标签	(162)
6.8.2 命令文件的运行	(129)	7.5.2 打印报表和标签	(163)
6.8.3 调试命令与辅助命令	(129)	习题	(163)
6.9 顺序结构程序设计	(130)	第8章 面向对象程序设计	(165)
6.10 分支结构程序设计	(131)	8.1 面向对象程序设计的基本概念	(165)
6.10.1 简单分支语句 (IF-ENDIF)	(131)	8.1.1 基本概念	(165)
6.10.2 选择分支语句 (IF-ELSE-ENDIF)	(131)	8.1.2 封装性、继承性和多态性	(166)
6.10.3 IF 条件语句的嵌套	(132)	8.1.3 类与对象	(167)
6.10.4 结构分支语句 (DO-CASE-ENDCASE)	(132)	8.1.4 基类与子类	(167)
6.11 循环结构程序设计	(133)	8.2 可视化编程基础	(167)
6.11.1 DO WHILE-ENDDO 循环	(133)	8.2.1 容器类与控件类	(167)
6.11.2 FOR-ENDFOR 循环	(136)	8.2.2 控件与对象	(168)
6.11.3 SCAN-ENDSCAN 循环	(137)	8.2.3 表单对象	(169)
6.12 过程及其调用	(138)	8.2.4 对象的引用	(171)
6.12.1 过程及过程调用的基本概念	(138)	8.2.5 常用事件	(171)
6.12.2 过程文件	(138)	8.2.6 常用方法	(172)
6.12.3 带参数的过程调用	(139)	8.3 常用控件的基本属性	(173)
6.12.4 过程调用的嵌套	(140)	8.4 程序设计的基本方法	(176)
习题	(140)	8.4.1 编程基本方法	(176)
第7章 报表和标签设计	(147)	8.4.2 编程步骤	(176)
7.1 报表的布局	(147)	8.5 修改和定制表单	(180)
7.1.1 创建报表步骤	(147)	8.5.1 选择控件	(181)
7.1.2 报表样式	(147)	8.5.2 控件的操作	(181)
7.1.3 报表布局	(148)	8.5.3 控制网格显示	(183)
7.2 报表文件的建立	(148)	8.6 建立简单的应用程序	(183)
7.2.1 用“报表向导”创建报表	(148)	习题	(185)
7.2.2 用“报表设计器”创建报表	(151)	第9章 控件的使用	(187)
7.2.3 用“快速报表”创建报表	(152)	9.1 标签控件与文本框控件	(187)
7.3 修改报表布局	(154)	9.1.1 标签控件(Label)	(187)
7.3.1 使用“报表设计器”	(154)	9.1.2 文本框控件(TextBox)	(190)
7.3.2 设置报表的数据源	(155)	9.2 命令按钮控件与编辑框控件	(196)
		9.2.1 命令按钮控件 (CommandButton)	(196)
		9.2.2 编辑框控件(EditBox)	(197)
		9.3 计时器控件与容器控件	(199)

9.3.1 计时器控件(Timer)	(199)	第 11 章 菜单设计	(261)
9.3.2 容器控件(Container)	(200)	11.1 用菜单设计器创建菜单	(261)
9.4 选项按钮组与复选框控件	(201)	11.1.1 菜单组成	(261)
9.4.1 选项按钮组控件 (OptionGroup)	(201)	11.1.2 创建菜单栏	(262)
9.4.2 复选框控件(CheckBox)	(207)	11.1.3 创建下拉菜单	(263)
9.5 列表框控件与组合框控件	(209)	11.1.4 创建子菜单	(263)
9.5.1 列表框控件(ListBox)	(209)	11.1.5 创建快捷菜单	(263)
9.5.2 组合框控件(ComboBox)	(214)	11.1.6 设计菜单组的分隔线	(265)
9.6 微调按钮控件与页框控件	(217)	11.1.7 指定热键	(265)
9.6.1 微调按钮控件(Spinner)	(217)	11.1.8 添加快捷键	(265)
9.6.2 页框控件(PageFrame)	(218)	11.1.9 菜单的修改	(266)
9.7 线条控件与形状控件	(222)	11.1.10 保存菜单	(266)
9.7.1 线条控件(Line)	(222)	11.2 用快捷菜单创建菜单	(267)
9.7.2 形状控件(Shape)	(224)	11.3 向菜单添加事件代码	(268)
9.8 表格控件与图像控件	(225)	11.3.1 向菜单添加“清理”代码	(268)
9.8.1 表格控件(Grid)	(225)	11.3.2 向菜单系统添加初始化代码	(269)
9.8.2 图像控件(Image)	(227)	11.3.3 启用和废止菜单项	(269)
9.9 设计简单的动画	(229)	11.3.4 为菜单或菜单项指定任务	(269)
9.10 类设计	(235)	11.3.5 预览菜单系统	(271)
9.10.1 类的设计方法	(235)	11.3.6 运行菜单系统	(271)
9.10.2 类的引用	(236)	11.4 修饰菜单	(272)
习题	(237)	11.4.1 显示状态栏信息	(272)
第 10 章 数据表的表单设计	(241)	11.4.2 定义菜单标题的位置	(272)
10.1 用表单向导设计表单	(241)	11.4.3 为菜单系统创建默认过程	(273)
10.1.1 表单向导	(241)	习题	(273)
10.1.2 一对多表单向导	(244)	第 12 章 工资管理系统设计	(275)
10.2 用表单设计器设计表单	(245)	12.1 开发应用系统的过程	(275)
10.2.1 打开表单设计器	(245)	12.2 工资管理系统主要模块简介	(276)
10.2.2 设置数据环境	(246)	12.2.1 系统的构成	(276)
10.2.3 向表添加字段	(247)	12.2.2 功能模块菜单	(277)
10.3 用表单生成器设计表单	(249)	12.3 项目与数据库的建立	(278)
10.3.1 使用快速表单添加字段	(249)	12.3.1 项目的建立	(278)
10.3.2 快速添加字段	(250)	12.3.2 数据库的建立	(278)
10.4 向表单中添加控件	(251)	12.4 工资管理系统模块设计	(279)
10.4.1 使用生成器向表单添加控件	(251)	12.4.1 系统主菜单设计	(279)
10.4.2 同时添加多个控件	(251)	12.4.2 系统主控表单模块设计	(282)
10.5 修饰表单	(257)	12.4.3 数据库初始化及备份 模块表单设计	(283)
10.5.1 设计具有背景图片和 立体字的表单	(257)	12.4.4 工资录入模块表单设计	(285)
10.5.2 设计具有流动字幕的表单	(258)	12.4.5 工资查找模块表单设计	(288)
习题	(259)	12.4.6 工资浏览模块表单设计	(291)
		12.4.7 修改记录模块表单设计	(293)

12.4.8 统计模块表单设计	(300)	13.2.8 分组与计算查询	(327)
12.4.9 打印模块表单设计	(301)	13.2.9 别名的使用	(328)
12.4.10 系统封面模块表单设计	(303)	13.2.10 内外层互相关嵌套查询	(329)
12.4.11 系统时间模块表单设计	(306)	13.2.11 使用量词和谓词的查询	(329)
12.4.12 系统帮助表单模块设计	(307)	13.2.12 超链接查询	(331)
12.4.13 报表表单设计	(307)	13.2.13 集合的并运算	(333)
12.5 程序的连编	(311)	13.2.14 查询输出去向	(334)
12.5.1 将全部的应用程序添加到 项目管理器中	(311)	13.3 操作功能	(336)
12.5.2 工资系统的主程序设计	(311)	13.3.1 插入	(336)
12.5.3 程序的调试	(312)	13.3.2 更新	(337)
12.5.4 程序的连编	(313)	13.3.3 删除	(338)
习题	(313)	13.4 定义功能	(338)
第13章 关系数据库标准语言 SQL	(316)	13.4.1 表结构的定义	(338)
13.1 SQL 简介	(316)	13.4.2 表的删除	(341)
13.1.1 SQL 语言的主要特点	(316)	13.4.3 表结构的修改	(341)
13.1.2 SQL 语句的执行	(317)	13.4.4 视图的定义	(344)
13.2 查询功能	(317)	习题	(346)
13.2.1 SQL 语法	(317)	第14章 常用函数	(350)
13.2.2 简单查询	(319)	14.1 数值函数	(350)
13.2.3 几个特殊运算符	(320)	14.2 字符函数	(352)
13.2.4 简单的联接查询	(321)	14.3 日期和时间函数	(355)
13.2.5 嵌套查询	(322)	14.4 数据类型转换函数	(356)
13.2.6 排序	(325)	14.5 测试函数	(358)
13.2.7 简单的计算查询	(325)	习题	(362)
		主要参考文献	(364)

第1章

Visual FoxPro 6.0 概述

本章导读

Visual FoxPro 数据库是一种关系型数据库,主要用于 Windows 环境。由于 Visual FoxPro 需要很少编程就可以建立一个面向对象的数据库应用程序,所以在众多的数据库软件中,Visual FoxPro 脱颖而出,成为一种通用的数据库软件。利用 Visual FoxPro 6.0 可以设计出丰富多彩的用户界面,在用户界面中可以放置各种控制部件,如命令按钮、图形、图片、图表等,从而设计出完全图形化的界面,方便用户的操作和使用。

1.1 数据库的基本概念

数据库是按一定方式把相关数据组织、存储在计算机中的数据集合。数据库不仅存放数据,而且还存放数据之间的联系。

1.1.1 数据与数据处理

数据是描述事物的符号。数据的概念有两个方面的涵义:描述事物特性的数据内容以及存储在媒体上的数据形式。数据形式可以是多样的,例如,姓名、电话号码、年龄、工资等都是数据。

数据的概念在数据处理领域中已经大大地拓宽了,数据不仅包括各种文字或字符组成的文本形式的数据,而且包括图形、图像、动画、影像、声音等多媒体数据。

数据处理是指将数据转换成信息的过程,通过数据处理可以获得信息,如通过公司的进货量和销售量,就可以知道库存量,从而为进货提供依据。

1.1.2 数据库的产生

计算机管理数据随着计算机的发展而不断发展,利用计算机对数据进行处理经历了 4 个阶段。

1. 人工管理阶段

计算机诞生之初,外存储器只有纸带、磁带、卡片等,没有像磁盘这样的速度快、存储容量大、

可随机访问、直接存储的外存储器。软件方面,没有专门的数据管理软件,数据包含在计算或处理它的程序之中。这一阶段的数据管理任务,包括存储结构、存取方法、输入输出方式等完全由程序员通过编程实现。这一阶段的数据管理称为人工管理阶段。

2. 文件系统阶段

20 世纪 50 年代后期至 60 年代后期,计算机开始大量地用于各种管理中的数据处理工作。大量数据的存储、检索和维护成为紧迫的需求。此时,在硬件方面,可直接存取的磁盘成为外存储器的主流;软件方面,出现了高级语言和操作系统。

这一段的数据处理采取程序与数据分离的方式,有了程序文件与数据文件的区别。数据文件可以长期保存在外存储器上被多次存取,在操作系统的文件系统的支持下,程序使用文件名访问数据文件,程序员只需关注数据处理的算法,而不必关心数据在存储器上如何存取。这一阶段的数据管理称为文件(系统)管理阶段。

文件系统中的数据文件是为了满足特定的需要而专门设计的,为某一特定的程序而使用,数据与程序相互依赖。同一数据可能出现在多个文件中,这不仅浪费空间,而且由于不能统一更新,容易造成数据的不一致性和数据冗余。

3. 数据库系统阶段

随着社会信息量的迅猛增长,计算机处理的数据量也相应增大,文件系统存在的问题阻碍了数据处理技术的发展,于是数据库管理系统便应运而生。

数据库技术的主要目的是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,使多个用户能够同时访问数据库中的数据;减少数据的冗余度,提高数据的一致性和完整性,提供数据与应用程序的独立性,从而减少应用程序的开发和维护费用。

数据库管理系统从 20 世纪 60 年代末问世以来,一直是计算机管理数据的主要方式。

4. 分布式数据库系统阶段

20 世纪 70 年代以前,数据库多数是集中式的,网络技术的发展为数据库提供了良好的运行环境,使数据库从集中发展到分布式,从主机/终端系统结构发展到客户/服务器系统结构。

1.1.3 数据库系统

1. 基本概念

(1) 数据库

数据库(DB)是存储在计算机存储器中的、结构化的相关数据的集合。它不仅存放数据,而且还存放数据之间的联系。

数据库中的数据面向多种应用,可以被多个应用程序共享。其数据结构独立于使用数据的程序,对于数据的增加、删除、修改和检索由系统软件进行统一的控制。

(2) 数据库管理系统

数据库管理系统(DBMS)是指帮助用户建立、使用和管理数据库的软件系统,主要包括三部分:数据描述语言(DDL)、数据操纵语言(DML)以及其他管理和控制程序。

(3) 数据库应用系统

数据库应用系统(DBAS)是指利用数据库系统资源开发的面向某一类实际应用的应用软件系统。一个 DBAS 通常由数据库和应用程序两部分构成,它们都需要在数据库管理系统 DBMS 支持下开发和工作。

(4) 数据库系统

数据库系统(DBS)是指引进数据库技术后的计算机系统,包括硬件系统、数据库集合、数据库管理系统和相关软件、数据库管理员、用户等五部分。

硬件系统是指运行数据库系统需要的计算机硬件,包括主机、显示器、打印机等。

数据库集合是指数据库系统包含的若干个设计合理、满足应用需要的数据库。

数据库管理系统和相关软件包括操作系统、数据库管理系统、数据库应用系统等相关软件。

数据库管理员是指对数据库系统进行全面维护和管理的人员。

数据库系统最终面对的是用户。

2. 数据库系统的特点

与文件系统相比,数据库系统具有以下特点:

- (1) 数据的独立性强,减少了应用程序和数据结构的相互依赖性。
- (2) 数据的冗余度小,尽量避免存储数据的相互重复。
- (3) 数据的共享高度,即一个数据库中的数据可以为不同的用户所使用。
- (4) 数据的结构化,便于对数据统一管理和控制。

1.2 数据模型

在现实世界中,事物之间是存在联系的,这种联系是客观存在的,是由事物本身的性质决定的。例如,学校教学系统中的教师、学生、课程、成绩等都是相互关联的。通常把表示客观事物及其联系的数据及结构称为数据模型。

1.2.1 基本概念

1. 实体

客观存在并且可以相互区别的事物称为实体。实体可以是实际的事物,如教师、职工、部门、单位等;也可以是抽象的事件,如比赛、订货、选修课程等。

2. 实体集

实体集是具有相同类型及相同性质(或属性)的实体集合,例如,某个学校的所有学生的集合可以被定为实体集 Students。

3. 属性

实体通过一组属性来表示,属性是实体集中每个成员具有的描述性性质。将一个属性赋予某实体集表明数据库为实体集中每个实体存储相似的信息,例如学生可以用学号、姓名、性别、生日

期等属性描述。但对每个属性来说,各实体有自己的属性,即属性被用来描述不同实体间的区别。

4. 联系

实体之间的对应关系称为联系,它反映了现实事物之间的相互联系。例如,一位学生可以选择多门课程;一个部门中可以有多个职工。

1.2.2 实体之间的联系

联系(也称关系)可以归纳为:一对一的联系、一对多的联系和多对多的联系三类。

1. 一对一的联系

若对于实体集 A 中的每一个实体,在实体集 B 中都有惟一的一个实体与之联系,则称实体集 A 与实体集 B 具有一对一的联系。例如,一个部门有一个经理,而每个经理只在一个部门任职,则部门和经理之间具有一对一的联系。

2. 一对多的联系

若对于实体集 A 中的每一个实体,实体集 B 中有 $n(n > 0)$ 个实体与之联系,反之,对于实体集 B 中的每个实体,实体集 A 中至多只有一个实体与之联系,则称实体集 A 与实体集 B 具有一对多的联系。例如,一个部门有若干个职工,而每个职工只在一个部门工作,则部门与职工之间是一对多的联系。

3. 多对多的联系

若对于实体集 A 中的每一个实体,实体集 B 中有 $n(n > 0)$ 个实体与之联系,反之,对于实体集 B 中的每个实体,实体集 A 中也有 $m(m > 0)$ 个实体与之联系,则称实体集 A 与实体 B 具有多对多的联系。例如,学生和选修课程的联系,某个学生可以选修多门课程,某选修课程也可以被多名学生选修。

1.2.3 数据模型简介

数据库中的数据从整体来看是有结构的,即所谓数据的结构化。各实体以及实体间存在的联系的集合称为数据模型,数据模型的重要任务之一就是指出实体间的联系。按照实体集间的不同联系方式,数据库分为三种数据模型,即层次模型、网状模型和关系模型。

1. 层次模型

层次模型的结构是树形结构,树的节(结)点是实体,树的枝是联系,从上到下为一对多的联系。每个实体由“根”开始,沿着不同的分支放在不同的层次上。如果不再向下分支,则此分支中最后的节点称为“叶”。图 1.1 为某学院的机构设置,“根”节点是学院,“叶”节点是各教研室。

支持层次模型的数据库管理系统称为层次数据库管理系统,其中的数据库称为层次数据库。

2. 网状模型

用网形结构表示实体及其之间的联系模型称为网状模型。在网状模型中,每一个节点代表一个实体,并且允许节点有多于一个“父”节点。这样网状模型代表了多对多的联系类型,如图 1.2 所示。

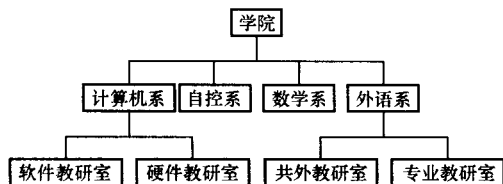


图 1.1 层次模型

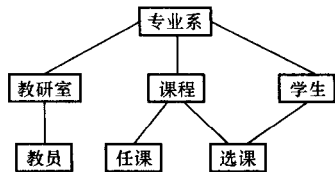


图 1.2 网状模型

支持网状模型的数据库系统称为网状数据库管理系统,其中的数据库称为网状数据库。

3. 关系模型

关系模型是以数学理论为基础构造的数据模型,它用二维表格来表示实体集中实体之间的联系。在关系模型中,操作的对象和结果都是二维表(即关系),表格与表格之间通过相同的栏目建立联系,如图 1.3 所示。

编号	姓 名	性别	年龄	职称	工作时间	婚否	简历	照片
1	张黎黎	女	26	助教	05/24/99	. T.	Memo	Gen
2	李 艳	女	30	助教	09/24/99	. T.	Memo	Gen
3	刘 强	男	38	讲师	12/24/99	. T.	Memo	Gen
4	王秋燕	女	45	讲师	10/09/93	. T.	Memo	Gen
5	姜丽萍	女	45	讲师	10/09/93	. T.	Memo	Gen
6	陈丽丽	女	38	讲师	09/27/95	. T.	Memo	Gen

图 1.3 关系模型

关系模型有很强的数据表达能力和坚实的数学理论,且结构单一,数据操作方便,最易被用户接受,以关系模型建立的关系数据库是目前应用最广泛的数据库。由于关系数据库具有许多优秀功能,层次数据库和网状数据库均已失去其重要性。

1.3 关系数据库

自 20 世纪 80 年代以来,新推出的数据库管理系统几乎都是基于关系模型。Visual FoxPro 就是一种关系数据库管理系统。

1.3.1 基本概念

1. 关系与表

关系的逻辑结构就是一张二维表,如学籍表、课程表等。在 Visual FoxPro 中,一个关系就是一个“表”,每个表对应一个磁盘文件,表文件的扩展名为 .DBF。表文件名即表的名称,也就是关系的名称。

2. 属性与字段

一个关系有很多属性(即实体的属性),对应二维表中的列(垂直方向)。每一个属性有一个名字,称为属性名。对于一张二维表格来说,属性就是表格中的栏(列),同栏的数据应具有相同

的性质,如“姓名”这一栏就只能填入姓名数据,而不能是其他数据。

在 Visual FoxPro 中,属性表示为表中的“字段”,属性名即为字段名。

3. 关系模型与表结构

对关系的描述称为关系模型,一个关系模型对应一个关系的结构。其格式为:

关系名(属性名 1,属性名 2,⋯,属性名 n)

在 Visual FoxPro 中对应的表结构为:

表名(字段名 1,字段名 2,⋯,字段名 n)

4. 元组与记录

在一个表格(一个关系)中,行(水平方向)称为“元组”。在 Visual FoxPro 中,元组表示为表中的“记录”。

一个表中可以有多条记录,也可以没有记录,没有记录的表称为“空表”。

5. 域

域是属性取值的范围,不同的属性有不同的取值范围,即不同的域。例如,成绩的取值范围是 0~100,逻辑型属性的取值只能是 .T.(真)或 .F.(假)。

6. 码与关键字

用来区分不同元组(实体)的属性或属性组合,称为码。在 Visual FoxPro 中对应的概念是关键字,关键字是字段或字段的组合,用于在表中惟一标识记录。如学生成绩表中的学号字段是关键字,因为学号不可能重复,可以用来惟一标识一条记录;性别字段就不是关键字,因为表中性别可能会在不同记录中出现,即有两个或两个以上的记录该属性是相同的。

如果码的任意真子集都不能成为码,这样的“最小码”称为“候选码”。候选码可能有多个,被选中用来区别不同元组的候选码称为主码。在 Visual FoxPro 中,对应的概念是:候选关键字和主关键字。

如果表中的某个字段不是本表的关键字,而是另外一个表中的关键字,则称该字段为外部关键字。

7. 关系模型与数据库

从集合论的观点来看,一个关系模型就是若干个有联系的关系模型的集合,一个关系模型是命名的属性集合,另外,关系是元组的集合,元组是属性值的集合。

在 Visual FoxPro 中,把相互之间存在联系的表放到一个数据库中统一管理。例如,在教工管理数据库中包括职工档案表和职工工资表。数据库文件的扩展名为 .DBC。

1.3.2 数据完整性

数据完整性是指数据库中数据的正确性和一致性(或相容性),数据完整性用来防止数据库中存在不合法的数据,防止错误的数据进入数据库中。

数据完整性可以分为实体完整性、域完整性和参照完整性。

1. 实体完整性

实体完整性是指数据库表的每一行都有一个惟一的标识。实体完整性由实体完整性规则来

定义,完整性规则是指表中的每一行在组成码(关键字)的列上不能有空值或重复值,否则就不能起到惟一标识行的作用。

2. 域完整性

域完整性是指数据库数取值的正确性。它包括数据类型、精度、取值范围以及是否允许空值等。取值范围又可分为静态和动态两种:静态取值范围是指列数据的取值范围是固定的,如年龄小于150;动态取值范围是指列数据的取值范围由另一个列或多列的值决定,或更新列的新值依赖于它的旧值。

3. 参照完整性

参照完整性是指数据库中表与表之间存在码(关键字)与外码(外部关键字)的约束关系,利用这些约束关系可以维护数据的一致性或相容性,即在数据库的多个表之间存在某种参照关系。要实现这种参照关系,首先创建表的码与外码:

(1) 当对含有外码的表进行插入、更新操作时,必须检查新行中外码的值是否在主表中存在,若不存在就不能执行该操作。

(2) 当对主表中的行进行删除、更新操作时,必须检查被删除行或被更新行中主码的值是否正在被一个或多个外码参照,若正被参照,就不能执行该操作。

1.3.3 对关系数据库的要求

通常生活中的二维表格有多种多样,不是所有二维表格都能被当作“关系”而存放 to 数据库。也就是说,在关系模型中对“关系”有一定规范化要求。

(1) 关系中的每个属性(列)必须是不可分割的数据单元。

(2) 同一关系中不应有完全相同的属性名,即在同一个表格中不能出现相同的列(字段)。

(3) 关系中不应用完全相同的元组,即在同一个表格中不能出现相同的行(记录)。

(4) 元组(记录)和属性名(字段)与次序无关,即交换两行或两列的位置不影响数据的实际含义。

1.3.4 关系运算

关系运算对应于 Visual FoxPro 中对表的操作,在对关系数据库进行查询时,为了找到用户感兴趣的数据,需要对关系进行一定的运算。这些运算以一个或两个关系作为输入,运算的结果将产生一个新的关系。关系运算主要指选择、投影、连接三种运算。

1. 选择运算

选择运算是指从关系找出满足给定条件的元组,又称为筛选运算。选择的条件以逻辑表达式给出,使得逻辑表达式的值为真的元组被选取。选择是从行的角度进行的运算,即选择部分行,经过选择运算可以得到一个新的关系,其关系模型不变,但其中的元组是原关系的一个子集。

在 Visual FoxPro 中,选择操作使用命令短语 FOR|WHILE<条件>或设置记录过滤器来实现。

2. 投影运算

从关系模型中指定若干个属性来组成新的关系称为投影。投影是从列的角度进行的运算,