

21世纪

高等院校计算机系列教材

Visual FoxPro 应用基础及基于部件 的系统设计技术

程学先 曾玲 林珊 程传庆 编著



含1CD



中国水利水电出版社

www.waterpub.com.cn

21 世纪高等院校计算机系列教材

Visual FoxPro 应用基础及基于 部件的系统设计技术

程学先 曾 玲 林 珊 程传庆 编著

中国水利水电出版社

内 容 提 要

本书从当代各专业大学生们应掌握的数据库技术与管理信息系统设计技术最基本的知识与实际动手能力出发,以 Visual FoxPro 为主要内容,结合部件技术,介绍关系数据库的基础理论和管理信息系统程序设计的方法。所提供的部件可以即插即用,用于一般管理信息系统设计,当使用该套部件开发管理信息系统时,许多系统都无需编程或只需少量编程就可以快速建成。本书从最基本的语句、语法、面向过程的程序设计方法、面向对象的设计方法,到部件的设计方法,由浅入深,由典型到一般,用例短小精悍,使读者逐步掌握 VFP 的基本知识、基本操作,并具有独立设计小型管理系统的功能。本书内容丰富,有较高的实用价值。

本书所附光盘包括一套软部件、办公自动化系统、一套作为范例的管理信息系统、VFP 命令与函数全文检索系统、与本书配套的试题库及其管理系统、教学幻灯片、学生课程设计精选等。

本书可以作为大专院校计算机基础课程或数据库课程的教材,也可以作为科研人员和管理人员的参考用书和“继续教育”教材。

版权所有,严禁将本书内容及所配光盘中的程序用于任何其他出版物

图书在版编目(CIP)数据

Visual FoxPro 应用基础及基于部件的系统设计技术/程学先等编著. —北京:中国水利水电出版社, 2004

(21 世纪高等院校计算机系列教材)

ISBN 7-5084-1797-6

I. V… II. 程… 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 (2003) 第 120556 号

书 名	Visual FoxPro 应用基础及基于部件的系统设计技术
作 者	程学先 曾 玲 林 姗 程传庆 编著
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@public3.bta.net.cn (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	787×1092mm 16 开本 23 印张 565 千字
版 次	2004 年 1 月第 1 版 2004 年 1 月北京第 1 次印刷
印 数	0001—5000 册
定 价	36.00 元 (含 1CD)

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

前 言

Visual FoxPro (简称 VFP) 关系数据库系统是个人计算机上广泛使用的数据库系统。它将数据库管理与开发系统集成在一起, 提供给我们完整而又丰富的工具、极高的处理速度和友好的界面。它使组织数据、定义数据库结构及建立应用程序等工作变得简单易行。它提供过程式编程语言, 又支持可视化的面向对象的编程技术, 使我们能快速创建功能强大的表单程序、查询程序和报表程序。它既可以用于开发单机上的管理信息系统, 也可作为前台与 SQL Server 数据库或 Oracle 数据库集成, 用于开发局域网上的管理信息系统。它是个人及部门开发小型管理信息系统的首选, 也是开发大型管理信息系统时设计原型系统的优秀工具软件。

本书介绍了数据库系统的基础知识和关系数据库的基本原理, 系统地介绍了 VFP 的语法、句法、面向对象程序设计的方法及 VFP 的开发工具的使用方法。作者总结了管理信息系统程序界面的共同特点, 设计了一套可以通用于基于 VFP 的管理应用系统设计的软部件。软部件是由一个到多个对象或构件组合而成、面向最终系统集成的程序模块。它通过动态定义子模式的方法进一步提高数据的逻辑独立性, 实现数据操作界面对数据结构的独立, 程序有更强的共享性、复用性和自适应性, 一个部件模块可以同时为多个应用服务, 绝大多数的部件模块在构建系统时均无需再编代码或只需编少量代码; 数据在表一级的结构改变可以不要求改变程序, 无需再编代码。它将提高程序设计的效率和质量, 降低软件开发、维护与扩展的成本。没有计算机基础的非计算机专业的学生和专业技术人员也能很容易地设计、使用和维护管理信息系统, 将促使计算机应用更加广泛、普及, 并向更深层次发展。例如, 要设计一套毕业生管理系统, 包括毕业生与用人单位的信息管理, 要求提供多种数据维护界面, 能根据任何字段或字段的组合条件查询信息, 能完成简单的统计并打印报表, 具有系统初始化、代码维护、数据导入与导出等功能。如果已建立好有关数据库的各数据表, 只需如下操作:

- (1) 将本书所附光盘中的部件库拷贝到自己的工作目录中。
- (2) 少数表单需要利用系统提供的生成器生成界面, 或进行装饰、加工和修改。
- (3) 修改菜单并连编成可执行文件。

只需几分钟到十几分钟, 工作就可以完成, 应用该套系统软件开发的一个管理信息系统项目获得了省科技进步三等奖。本书精选典型软部件的主要程序片段作为贯穿全书的范例, 一方面学习 VFP 的语言, 一方面学习管理信息系统的程序设计技术, 使既能加深对理论知识及有关技能的理解和学习, 又能具备开发简单管理系统的能力。结合部件应用, 本书讨论了使用部件开发一般管理系统的方法, 尤其着重介绍了在关系数据库系统平台上开发办公自动化系统的方法, 书中提出了图形符号法以解决 workflow 问题, 方法简单、易行、可靠, 取得较好的效果。本书所附光盘包括: 管理信息系统的软部件选集 (涉及书中范例的软部件提供源程序); VFP 试题库及其管理系统、本书配套教学幻灯片、VFP 命令与函数全文检索系统、大学生课程设计精选 (提供源程序) 等。

必须说明的是, 尽管我们设计了上述软部件, 使得用户只需编写少量代码就能设计出实

用的、功能很强的应用系统，有利于初级读者学习和应用，但它只是基于 VFP 的一个新的开发平台，要在该平台的基础上设计出有特色和有特殊要求的程序的话，对程序员的要求将会更高。因此，本书特别强调培养程序设计的能力，从最基础的内容讲起，突出语法尤其是程序设计方法的教学，由浅入深，逐步深入地组织教学内容。读者一定要完整地、全面地学习 VFP 的程序设计语言，才能进入程序设计的自然王国。我们希望的是读者以我们提供的部件为例，更有效、更深入地学习，真正掌握 VFP 及管理信息系统程序设计的精髓。

本书前 5 章讲述 VFP 与关系数据库的基础知识，建议作为讲授重点。后 6 章介绍管理信息系统程序的基本结构、VFP 程序设计方法和部件技术，可根据学时情况作适当介绍，由学生通过上机学习掌握。本书内容涵盖 VFP 二级考试大纲（2002 年版）所要求的全部内容，可以作为高等学校计算机基础课程或数据库课程的教材，也可以作为科技人员的参考用书和“继续教育”教材。部件库可以作为局域网或单机管理信息系统的开发平台。

本书由程学先、曾玲、林姗、程传庆编著。部件程序由程学先、程传庆、程传慧等设计。参加管理信息系统程序设计的有金波、林姗、彭宏昊、刘伟、曾立平、程传庆、雷飞宇、程传慧、陈永辉、江南等。参加教学幻灯片制作与试题库试题设计的有曾玲、史函、祝苏薇、胡显波、饶昆、杨晓燕、高霞、李芳、余晓艳、周金森、叶品菊、高亮、钱涛、李莎等。参加 VFP 命令与函数全文检索系统设计的有曾玲、程传庆、王莉、程传慧等。课程设计作品由周斌、张德强、杨晏国、晏鹏、李英豪、李娟、徐莉、梁文武、罗品洁、孙敏、邓斯德等设计。

由于时间仓促及作者水平有限，书中错误和疏漏之处在所难免，敬请广大读者批评指正。

作 者

2003 年 9 月

目 录

前言

第 1 章 数据库基础理论	1	习题二	39
本章学习目标	1	第 3 章 Visual FoxPro 基础	40
1.1 数据处理	1	本章学习目标	40
1.1.1 数据、信息	1	3.1 常量、变量和表达式	40
1.1.2 数据处理的发展	2	3.1.1 常量	40
1.1.3 数据库技术概述	2	3.1.2 变量	40
1.1.4 数据库管理系统	6	3.1.3 表达式	45
1.2 数据模型	7	3.2 函数	47
1.2.1 数据模型概念	7	3.2.1 算术运算函数	47
1.2.2 层次数据模型	7	3.2.2 字符运算函数	48
1.2.3 网状数据模型	8	3.2.3 日期和时间运算函数	49
1.2.4 实体-关系模型 (E-R 模型)	8	3.2.4 转换函数	50
1.2.5 关系数据模型	10	3.2.5 宏替换函数	52
1.2.6 实体-关系模型向关系数 据模型的转化	11	3.2.6 状态检验函数	53
1.3 关系数据库概念	13	3.2.7 对话框函数	55
1.3.1 关系数据库结构	13	3.2.8 系统函数	57
1.3.2 索引	14	3.2.9 其他函数	57
1.3.3 范式	16	本章小结	58
本章小结	18	习题三	58
习题一	19	第 4 章 数据表与数据库	60
第 2 章 Visual FoxPro 系统概述	21	本章学习目标	60
本章学习目标	21	4.1 关于建立表及更新表结构的命令	60
2.1 Visual FoxPro 简介	21	4.1.1 建立“表”命令	60
2.1.1 Visual FoxPro 的发展史	21	4.1.2 修改表结构	62
2.1.2 VFP 初步介绍	22	4.2 对数据库数据操纵的基本语句	63
2.1.3 VFP 主屏幕界面	23	4.2.1 打开或选择数据表语句	63
2.1.4 VFP 程序设计工具	27	4.2.2 数据更新语句	64
2.2 Visual FoxPro 的环境配置	32	4.2.3 SQL 数据更新语句	67
2.2.1 使用“选项”对话框	32	4.2.4 索引与排序命令	68
2.2.2 系统内存变量	33	4.2.5 移动指针类命令与查询命令	69
2.3 项目管理器	34	4.2.6 统计类命令	70
2.3.1 建立项目管理器	34	4.2.7 拷贝与导入命令	72
2.3.2 对项目的管理操作	35	4.3 SET...TO 命令群	75
本章小结	38	4.4 SET...ON/OFF 命令群	76
		4.5 加快程序运行速度的技术	77

4.5.1 名称运算式.....	78	对象程序设计.....	124
4.5.2 EVALUATE()函数.....	78	6.2.1 求若干个数组相加求和的程序.....	124
4.5.3 使用 RUSHMORE 技术.....	79	6.2.2 数据录入程序设计.....	127
4.6 关系代数.....	79	6.2.3 修改与删除程序设计.....	129
4.6.1 传统的集合运算.....	79	6.2.4 列表框与组合框.....	130
4.6.2 专门的关系运算.....	82	6.2.5 模糊查询程序设计.....	133
4.7 SQL 查询语言概貌.....	85	6.2.6 使用文件打开对话框 选取文件并浏览该文件.....	134
4.7.1 SQL 数据查询语句.....	85	6.2.7 通用字段录入程序.....	135
4.7.2 实现对单一数据表的简单查询... ..	87	6.2.8 使用进程条.....	136
4.7.3 条件查询.....	88	6.3 多功能数据维护表单的最简设计.....	138
4.7.4 对两个以上表的连接查询.....	89	6.3.1 建立管理项目与数据表.....	138
4.7.5 嵌套查询.....	89	6.3.2 界面设计与建立.....	140
4.7.6 求并、交、差.....	90	6.3.3 方法程序设计.....	142
4.8 视图.....	90	6.3.4 运行表单程序.....	144
本章小结.....	91	6.4 添加单条件查询功能.....	144
习题四.....	91	6.4.1 快速移动指针的方法.....	145
第 5 章 VFP 过程式程序设计.....	94	6.4.2 单条件查询程序设计.....	145
本章学习目标.....	94	6.5 将文本框等控件数据源改为内存变量.....	153
5.1 VFP 应用系统程序的结构.....	94	6.5.1 将文本框、编辑框绑定 的数据源改为内存变量.....	154
5.1.1 VFP 应用系统的模块结构.....	94	6.5.2 自定义方法及其调用.....	158
5.1.2 子程序与过程.....	96	6.5.3 代码表及其使用.....	160
5.2 输入输出语句.....	98	6.5.4 查询结果可能为多条记录的 又一处理方法.....	163
5.2.1 人机交互语句.....	98	6.5.5 使用颜色标志当前控件.....	164
5.2.2 格式化输入输出语句.....	99	6.6 添加成批修改和条件查询等功能.....	165
5.3 程序控制语句.....	101	6.6.1 批修改功能.....	165
5.4 多表关联操作.....	110	6.6.2 条件查询功能.....	167
5.4.1 利用编程处理多表操作 问题(联访).....	111	6.6.3 与 Excel 的联系.....	168
5.4.2 关联命令 (SET RELATION)	112	6.6.4 索引.....	169
5.4.3 批修改命令 (UPDATE)	113	6.7 打印程序设计.....	169
5.5 其他常用命令.....	114	6.7.1 快速建立报表格式文件.....	169
本章小结.....	114	6.7.2 修改报表格式文件.....	170
习题五.....	115	6.7.3 要求分组统计的报表格式文件.....	172
第 6 章 面向对象程序设计方法.....	118	6.7.4 一对多报表设计.....	172
本章学习目标.....	118	6.7.5 执行打印程序.....	173
6.1 对象的基本概念.....	118	6.7.6 复杂报表设计.....	174
6.1.1 对象和容器.....	118	6.8 自定义类库设计.....	177
6.1.2 属性.....	119	6.8.1 实现控件程序对数据的独立.....	177
6.1.3 事件与方法.....	123		
6.2 从过程式程序设计到面向			

6.8.2 使用自定义类库和类.....	178	7.7.2 自制工具条及其使用.....	224
6.8.3 面向对象程序设计的基本概念.....	179	本章小结.....	224
6.8.4 通过编程定义类.....	181	习题七.....	225
6.9 动态定义子模式加强数据逻辑独立性.....	183	第8章 查询与统计程序设计.....	227
6.10 使用英文字段名的数据表的 维护程序设计.....	184	本章学习目标.....	227
6.11 使用工艺表实施对数据 安全性的保护.....	185	8.1 查询向导与查询设计器.....	227
6.12 一对多录改表单的设计.....	187	8.1.1 查询向导.....	227
6.13 自适应单记录录改表单的设计.....	189	8.1.2 查询设计器.....	229
本章小结.....	194	8.2 最简组合查询程序设计.....	231
习题六.....	195	8.2.1 最简组合查询程序设计.....	232
第7章 表格式数据维护表单程序设计.....	197	8.2.2 初始化.....	232
本章学习目标.....	197	8.2.3 条件表达式的生成与 查询程序设计.....	234
7.1 表格的常用属性及其意义.....	197	8.3 通用的具有简单统计功能的 组合查询表单.....	235
7.1.1 表格的主要属性.....	197	8.3.1 通用的具有简单统计功能的 组合查询表单的设计思想.....	235
7.1.2 表格常使用的方法.....	198	8.3.2 通用的具有简单统计功能的 组合查询表单设计.....	236
7.1.3 列(Column)的常用属性.....	198	8.4 涉及多表的组合查询表单.....	243
7.1.4 在表格里添加其他控件的方法.....	199	8.5 多功能查询与统计表单.....	245
7.2 利用表格控件设计单表数据维护表单.....	199	8.5.1 页框.....	245
7.2.1 界面设计.....	200	8.5.2 具有明细、小计、总计分层 格式的查询统计程序的设计.....	246
7.2.2 设置表格控件的属性.....	200	8.6 使用英文字段名的查询程序.....	257
7.2.3 主要方法的程序清单.....	201	本章小结.....	258
7.2.4 查询程序设计.....	203	习题八.....	258
7.3 表格式数据维护部件的设计.....	203	第9章 应用系统工作平台程序设计.....	260
7.4 其他形式的表格式录改表单.....	211	本章学习目标.....	260
7.4.1 含有代码字段的表的 表格式录改表单.....	211	9.1 概述.....	260
7.4.2 实现数据安全与数据 完整性保护.....	213	9.2 数据库结构设计.....	261
7.4.3 采用英文字段名的 数据的表格式录改表单.....	215	9.3 应用系统平台设计.....	261
7.5 单记录-表格录改表单.....	217	9.3.1 主程序设计.....	262
7.6 特殊表格录改程序的设计.....	218	9.3.2 菜单程序设计.....	264
7.6.1 学生成绩修改程序.....	218	9.3.3 建立可执行文件(.EXE 文件).....	267
7.6.2 特殊学生成绩添加与修改程序.....	220	9.4 各种导入程序设计.....	267
7.6.3 “修改存盘”按钮的 Click 方法程序.....	221	9.4.1 从数据表导入.....	268
7.7 表单集.....	222	9.4.2 从文本文件导入.....	270
7.7.1 表单集设计.....	223	9.5 各种导出程序设计.....	273
		9.5.1 ASCII 文本格式(SDF)导出.....	273

9.5.2 自定义格式导出.....	274	本章小结.....	301
9.5.3 从数据表导出到数据表.....	275	习题十.....	301
9.6 其他系统管理程序.....	276	第 11 章 办公自动化程序设计.....	302
9.6.1 代码维护程序.....	276	本章学习目标.....	302
9.6.2 清库程序.....	277	11.1 办公自动化系统简介.....	302
9.6.3 用户表维护程序与 权限维护程序.....	278	11.1.1 办公自动化系统需求分析.....	302
9.6.4 设计.DBF 式帮助程序.....	278	11.1.2 办公过程的特点.....	304
9.7 界面修饰技术.....	279	11.1.3 对象-关系数据库.....	306
9.7.1 利用计时器 Timer1 产生 动画效果.....	279	11.2 通知管理程序设计.....	309
9.7.2 使用形状 Shape1 划分区域、 设置不同底色.....	279	11.3 发文开办程序设计.....	311
9.7.3 使用图像 Image1 设置图形衬底.....	280	11.3.1 发文开办表单界面设计.....	311
9.7.4 使用线条 line1 建立新边框.....	280	11.3.2 工作流定义程序设计.....	312
9.7.5 为表单增加背景音乐.....	280	11.3.3 撰稿与审核程序设计.....	315
9.8 OLE 与 ActiveX 控件.....	280	11.3.4 发文签发与归档程序设计.....	317
9.8.1 ActiveX 控件及其使用.....	280	11.3.5 修改权限与流程序序设计.....	318
9.8.2 拨号通信表单设计.....	281	11.4 办公自动化系统中的目录树菜单设计.....	318
本章小结.....	284	11.4.1 目录树菜单的应用.....	318
习题九.....	284	11.4.2 动态建立目录树表.....	319
课程设计参考题.....	285	11.4.3 根据目录树表建立目录树界面.....	323
第 10 章 基于 C/S 模式的系统设计.....	287	11.4.4 利用目录树程序进行控制.....	324
本章学习目标.....	287	11.5 办公自动化系统与文件管理 系统的联系.....	324
10.1 以升迁方式建立 SQL Server 的数据库与数据表.....	287	11.5.1 在办公自动化系统的表中添加 文件系统中的节点名的方法.....	325
10.1.1 建立指向 SQL Server 表的连接.....	288	11.5.2 设计递归程序从目录树表 中删除一个节点.....	327
10.1.2 以升迁方式建立 SQL Server 数据库与数据表.....	289	11.5.3 通用目录树菜单部件的设计.....	328
10.2 原来指向 VFP 数据表程序的修改.....	290	11.6 设计自己的目录树结构帮助系统.....	330
10.2.1 建立指向 SQL Server 表的远程视图.....	291	11.6.1 目录树结构帮助系统的生成 表单.....	330
10.2.2 应用程序设计.....	294	11.6.2 执行目录树帮助程序表单设计.....	332
10.3 远程单记录录改表单部件.....	295	本章小结.....	333
10.3.1 表单的 INIT 完整程序清单.....	295	习题十一.....	333
10.3.2 录入存盘程序设计.....	298	附录 A 部件库使用说明.....	334
10.3.3 修改存盘程序设计.....	300	附录 B 将部件用于管理信息系统设计.....	345
10.3.4 删除程序设计.....	300	附录 C VFP 命令与函数全文检索系统.....	355
10.4 数据导入远程表.....	300	附录 D VFP 题库管理系统使用说明.....	357
		附录 E VFP 课程设计作品目录.....	359

第1章 数据库基础理论

本章学习目标

本章主要介绍数据库技术的相关概念、数据库系统的组成、数据结构、存储结构、数据模型等基础知识。通过本章学习，读者应该掌握以下内容：

- 数据与信息的概念，数据处理的概念与内容
- 数据库技术相对于文件管理阶段的优越性
- 数据库系统的组成
- 概念模式、子模式、内模式、视图等概念
- 数据共享、数据冗余、数据一致性、数据完整性、数据独立等概念
- 实体的标识属性与关键字的概念
- 实体及相互间联系，一对一、一对多、多对多联系等
- 数据库系统的体系结构
- 实体联系模型，关系数据模型，实体联系模型向关系数据模型的转化
- 范式

1.1 数据处理

当今时代是信息技术飞速发展的时代，社会的一切组织、人类思想和行动的各个领域都离不开信息。在各种组织和个人的活动中都需要决策，而决策正确与否主要依赖于信息是否正确、及时、完整和真实。数据是信息的载体，是表示信息的物理符号。信息是数据的内涵，是数据处理的结果。数据处理是将原始数据转换成信息的过程。其目标就是要在适当的时刻以恰当的形式给不同的人提供所需要的数据。为此必须进行数据的收集、组织、存储、控制、选取、维护，这也正是数据管理的任务。

1.1.1 数据、信息

一般地，人们经常将术语“数据”与“信息”互换使用，但在信息或数据库系统中，区别它们是必要的。

信息（Information）是客观事物属性的反映。它所反映的是关于某一客观系统中某一事物的某一方面属性或某一时刻的表现形式。通俗地讲，信息是经过加工处理并对人类客观行为产生影响的事物属性的表现形式。

数据（Data）是反映客观事物属性的记录，是信息的载体。对客观事物属性的记录是用一定的符号来表达的，因此说数据是信息的具体表现形式。

信息与数据在概念上是有区别的。从信息处理角度看，任何事物的属性都是通过数据来表

示的,数据经过加工处理后,使其具有知识性并对人类活动产生决策作用,从而形成信息。从计算机的角度看,数据泛指那些可以被计算机接受并能够被计算机识别处理的符号。

总之,信息是有用的数据,数据是信息的表现形式,信息是通过数据符号来传播的。

1.1.2 数据处理的发展

由于现实世界的事物往往是相互关联的,基于这一事实,可以从一类数据出发,参照相关数据,进行加工计算,产生出一些新的数据。这新的数据又表示了新的信息,可以作为某种决策的依据,这整个过程就是数据处理的过程。

所谓数据处理,实际上就是利用计算机对各种类型的数据进行采集、整理、组织、存储、分类、排序、检索、维护、加工、统计和传输等一系列操作过程。数据处理的目的是从大量的、原始的数据中获得人们所需要的资料并提取有用的数据成分,作为行为和决策的依据。

在数据处理活动中,计算相对比较简单,很少涉及复杂的数学模型。但是却有其本身固有的特点,表现为数据量大,且数据之间有着复杂的逻辑联系。因此,数据处理任务的矛盾焦点不是计算,而是把数据管理好。

数据管理是指:数据的收集、定义、录入、修改、删除、排序、查询和传送等操作,是数据处理的基本环节,是任何数据处理任务必有的共性部分。应该开发出通用而又方便好用的软件,把数据有效地管理起来。

数据管理技术经历了以下三个阶段:

(1) 人工管理阶段。人工管理阶段出现在 20 世纪 50 年代中期以前,在应用程序中不仅要设计数据的逻辑结构,还要阐明数据在存储器上的存储地址。

在这一管理方式下,应用程序与数据之间相互结合不可分割,当数据有所变动时程序需随之改变,独立性差;另外,各程序之间的数据不能相互传递,缺少共享性,因而这种管理方式既不灵活,也不安全,编程效率较低。

(2) 文件系统阶段。20 世纪 50 年代后期到 60 年代中期,可以把有关的数据组织成一种文件,这种数据文件可以脱离程序而独立存在,由专门的文件管理系统实施统一管理。数据与程序分开在不同文件中存放,应用程序通过文件管理系统对数据文件中的数据进行加工处理,数据与程序分离实现了数据的物理独立。它使各程序之间可以相互传递数据,初步实现了数据共享。

但是,数据文件的结构和程序密切相关,数据文件之间不能建立任何联系,因而数据的通用性仍然较差,冗余量大。

(3) 数据库系统阶段。20 世纪 60 年代后期以来,出现了统一管理数据的专用软件系统:数据库管理系统(DBMS)。它以专门的文件或文件的专门部分存放对数据文件结构的定义,应用程序既可与整体数据集对应,也可与整体数据集的某个子集对应,使数据文件的结构和程序相分离,我们称其实现了数据的逻辑独立。DBMS 还提供对数据管理和访问的工具,使程序的编制效率和质量都大大提高。由于数据文件间可以建立关联关系,数据的冗余大大减少,数据共享性显著增强。

1.1.3 数据库技术概述

数据库是指有组织地动态存储的相互关联的数据的集合。设计一个数据库应用系统,首

先要了解全系统数据和功能的需求；对全系统涉及的所有数据进行分析 and 整理，分析数据之间的联系，用一定的数据模型来表示；设计系统全局的数据结构，用一定语言加以表示并定义；要注意考虑数据的存放位置；数据量大小；对安全保密性、数据正确性、防错纠错措施等方面的要求；在计算机中建立数据库结构；开发维护和使用数据的应用程序。在设计程序时注意要方便用户使用，操作简单，系统高效可靠。

1. 实体与属性

我们将现实世界中客观存在的可标识的事物、概念或事件称为实体。例如学生、部门、课程、商品、记账活动等都是实体。每个实体有不同的特性，用不同的属性来描述。例如，学生可用学号、姓名、性别、出生日期、年龄、所在单位等属性描述，对于具体的学生的属性可以用具体的数据来描述，例如 20000101、吴平、男、83 年 2 月 5 日、20、电计系等，称为属性值。对应一个实体的所有属性值的集合称为记录。有些属性的取值有一定范围，属性的取值范围称为域。具有相同属性描述的实体的集合称为实体集，一个或多个实体集的所有数据可以按一定结构组织存放到文件中，其结构称为全局数据结构。

2. 概念模式

用某一种数据库语言对全局数据结构的定义称为概念模式，简称模式。例如可将“学生”的有关数据组织在一个二维表中，每一行存放一个学生的所有数据，即存放一条记录。这些数据组成学号、姓名、性别、出生日期、年龄、所在单位六列，每列称为数据项或字段。如分别规定为字符型、字符型、逻辑型、日期型、整型、字符型数据类型，所占宽度分别为 6、8、1、8、4、24 个字节。我们用学生（学号 C(6)，姓名 C(8)，性别(L)，出生日期 (D)，年龄 (I)，所在单位 C(24)）表示其结构，这就是关系数据库“学生”表的模式的基本部分。其中，逻辑型、日期型、整型数据存放时默认宽度分别为 1、8 和 4；学号、姓名、性别、出生日期、年龄、所在单位等称为数据项或字段的名称；每一学生的学号、姓名、性别、出生日期、年龄、所在单位的具体内容称为数据项或字段的值。

一个表的关系模式常简单地表示为：表名（数据项 1，数据项 2，...）。其完整表示除了包括数据库名、数据结构方式、记录的构成等内容外，还应包括记录的标识数据项、数据范围及使用权限等。在关系数据库中，“标识数据项”指能惟一标志一条记录的数据项，又叫关键字，例如学生表中的“学号”。

3. 子模式（外模式）与视图

要增强数据的共享性，还需要进一步减少应用程序对全局性数据结构的依赖，让应用程序只和局部数据结构相关，为此可以进一步定义概念模式的逻辑子集，称为子模式。

在子模式定义中允许改变数据项对应的名称、数据类型、宽度，可以加入权限控制的定义。子模式可以在局部范围内加强数据的安全性，对使用数据给予更强的控制。

在一些数据库中，关于数据结构的定义还需要加入所存储的文件名及类型、数据存储形式（顺序文件或随机文件等）、采用的指针、索引文件的结构及存储设备、物理块大小等，这些有关存储方式、物理结构等的描述称为存储模式或内模式，数据库通过存储模式再借助于操作系统实现对数据存储文件的操作。

在关系数据库中，在只和局部数据结构相关这层意义上，视图类似于子模式，是对用户所看到的数据结构的描述。关系数据库中应用程序可以基于模式，也可以基于视图。

例如 VFP 中, 设有学生 (学号, 姓名, 年龄) 和成绩 (学号, 课名, 分数) 两个表, 可以建立视图 SV, 其功能是列举所有不及格的学生姓名、课名和分数。

```
CREATE VIEW SV AS SELECT 学生.姓名,成绩.课名,成绩.分数  
FROM 学生,成绩 (WHERE 学生.学号=成绩.学号 AND 成绩.分数<60)
```

可以将 SV 看成一般表, 通过它对源表“学生”和“成绩”进行查询操作。它实现了两个源表的联系, 相当于源表模式的子集, 它和表的不同之处在于, 它并没有真正地存储数据, 它所存取的数据必须依附于所关联的数据表, 它是一种虚的映射关系。视图使用的字段名可以与源表不同, 其好处是:

(1) 简化用户的操作。视图可以使用户将注意力集中在它所关心的数据上。如果这些数据不是直接来自基本表, 则可以通过定义视图, 使用户眼中的数据库结构简单、清晰, 并且可以简化用户的数据查询操作。例如, 那些定义了若干张表连接的视图, 就将表与表之间的连接操作对用户隐蔽了起来, 用户需要做的只是对一个虚表的简单查询, 而这个虚表是怎样得来的, 用户无须了解。

(2) 使用户能以多种角度看待同一数据。视图机制能使不同的用户以不同的方式看待同一数据, 当许多不同种类的用户使用同一个数据库时, 这种灵活性是非常重要的。

(3) 增强数据的逻辑独立性。数据的逻辑独立性是指当数据库增加新的关系、对原有关系增加新的字段或改变字段的特性时, 用户和用户程序不受影响。

例如, 将“学生”关系学生 (学号, 姓名, 性别, 年龄) 改变字段名为关系 Student(serialno, name, sex, age) 后, 原有程序一般都需要改变。但是如果建立一个学生视图:

```
CREATE VIEW 学生(学号,姓名,性别,年龄)AS SELECT Student.serialno,  
Student.name,Student.sex,Student.age FROM Student
```

意义是建立关于 Student 的视图, 其名字是“学生”, 字段名与原学生表各字段名相同, 分别代表 Student 的 serialno、name、sex、age 字段。这样尽管相应数据库结构改变了, 但用户程序通过视图操作, 表名、字段名都没有改变, 应用程序可以保持不变。

要说明的是, 视图只能在一定程度上提供数据的逻辑独立性, 通过视图对数据表的更新是有条件的, 在某些情况下, 应用程序中修改数据的语句仍会因基本表格结构的改变而改变。

(4) 视图能够对数据提供安全保护。有了视图就可以在设计数据库应用系统时, 对不同的用户定义不同的视图, 使数据不会被无权访问这些数据的用户看到或修改。从而提供了对数据的安全保护功能。例如, 学生表涉及 3 个班的学生数据, 如果在其上定义 3 个视图, 每个视图只包含 1 个班的学生数据, 并只允许每个班的学生通过相应视图查看其所在班的学生情况。这样可以使各班数据相对安全。

(5) 如果以 SQL Server 或 Oracle 数据库为后台, 以 VFP 作为前台开发工具, VFP 提供的远程视图使我们能极其方便地如同对 VFP 数据库一样实现对后台数据库的操作。

综上所述, 最终用户面对的是应用程序, 他所见到的数据部分称为 I/O 视图。程序员面对 I/O 程序及子模式, 所见到的数据库形式称外部视图。数据库管理员 (DBA) 管理数据全局结构, 面对概念模式和内模式。系统管理员关心的是各个数据库及其他文件在系统中的存储和管理, 面对的是各个数据库的存储模式, 他看到的部分除全局数据结构外还关心存取结构和存储方法, 称为内部视图, 如图 1-1 所示。

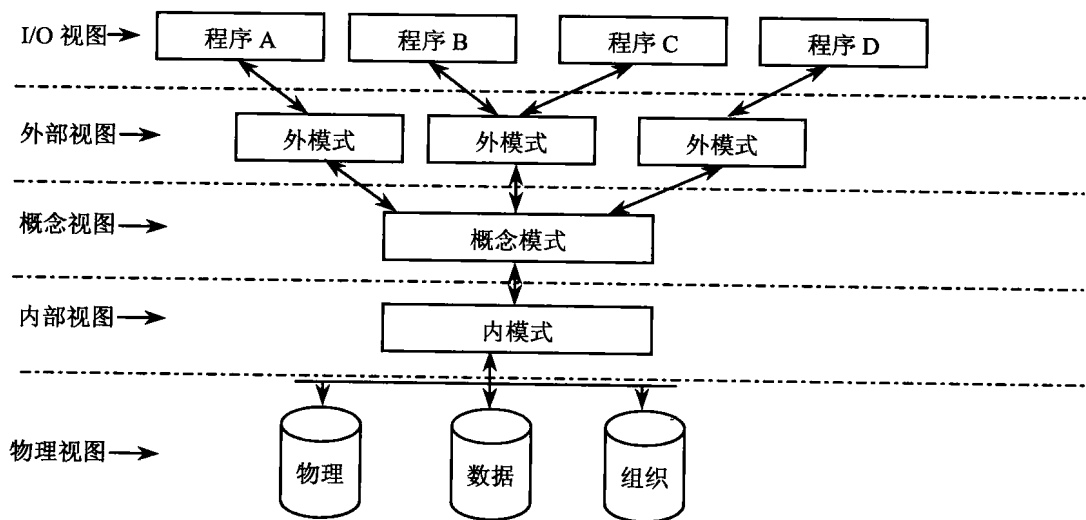


图 1-1 数据库的三级模式与二级映像

4. 数据冗余

所谓数据冗余，是指同一数据在多个不同的地方存放。例如，同一个人的基本情况，在人事管理系统中存放，同时也在财务部门存放。这不仅会导致数据量的增加，使系统处理速度变慢，效率降低，而且易发生错误。多一个数据存储，就需要多一套维护程序，多一些发生错误的可能，从而影响全系统的性能。因此在实际设计中，应尽量减少数据冗余，控制冗余度，这就需要正确地定义全局数据结构。

5. 数据的安全性

数据的安全性是指不同用户各自控制在一定权限范围内使用数据，以防止数据遭到人为破坏或泄密。数据库中数据的安全性通常通过设置用户名和用户口令来实现。系统将用户名、口令及对某种操作的许可权保存在计算机中。用户要登录系统、数据库或做某一种操作时，要先分别按提示输入用户名和口令，如果正确，系统才按照权限使用。

6. 数据的完整性

数据的完整性包括实体完整性、参照完整性和域完整性。数据库提供数据的完整性控制，是为了在数据库的使用过程中防止错误或不当的数据进入数据库。

(1) 实体完整性。为了管理方便，我们利用关键字（码）来惟一区别一个实体，对于具体问题起作用的关键字叫主关键字（主码）。例如，对于学生关系可以用学号来标识一个学生，学号就是学生表的关键字。在限定人名不得相同的条件下，姓名也可以是关键字，这时如果按姓名查询时，姓名就是主关键字。我们将是否存在关键字、关键字在所有数据中是否能惟一标识一条记录等作为数据是否有效的主要标准，称之为实体完整性。

(2) 参照完整性。在不同实体集之间有时有特定的联系。例如学生要学习许多课程，每一门课程可能有许多学生学习，每个学生学习每门课程最终有一成绩。在关系数据库中，可以设计成绩表：成绩（学号，课程名，分数）和学生表：学生（学号，姓名，年龄）来分别管理课程与学生的数据。在这两个表中，都存在“学号”数据，表示同一个实体的这两个数

据应该相同，如果要求成绩表中的数据有意义，就要求对于“成绩”中每一条记录的学号必须在学生表中存在，即必须是某个学生具体的成绩。

在关系数据库中，为了实现表与表之间的联系，有时需要将一个表的关键字作为数据之间联系的纽带放到另一个表中，这些在另外一个表中起联系作用的属性称为外关键字（外码），其每一个值只要不空，都必须在第一个表中存在，称为参照完整性。在学生与课程两个表中，课程表中的学号是学生表的外关键字，它的每一个值都必须在学生表中能够找到，否则我们便认为这个关系无意义，两个表不相容。

（3）域完整性。数据应该客观真实地表现周围事物。每个属性的数据在实际生活中常具有一定范围，属性的取值范围称为域。例如性别只能是男或女，{男，女}是性别域；年龄字段定义成整型，但是作为一般人来讲，不能超过 150 这个范围，也不可能为负数，{0-150}是年龄域等。数据的取值应当在规定的有效范围内，称为域完整性。数据库技术可识别一个数据是否在一定范围内，在限定范围内的数据被认为是正确的，不在限定范围内的数据输入则是错误的。一些数据库可以通过定义数据完整性约束要求，或通过触发器程序的设计及其他方式来确保只允许在正确范围内的数据录入，实现域完整性控制。

7. 并发性控制

当多个用户同时访问同一个数据时，可能会互相干扰而出现数据不一致的情况，常常通过加锁的方法来解决这个问题。

1.1.4 数据库管理系统

数据库管理系统简称 DBMS (Database Management System)，它是数据库系统的关键组成部分。任何数据操作，包括数据库定义、数据查询、数据维护、数据库运行控制等都是在 DBMS 管理下进行的。DBMS 的功能包括如下几方面：

（1）数据库定义功能。DBMS 提供数据描述语言 DDL (Data Description Language) 来定义概念模式、外模式、内模式，并将各种模式翻译成相应的目标代码，保存在数据字典中。这些目标模式反映了数据库的结构。

（2）模式映射功能。虽然实际的数据库软件产品很多，它们支持不同的数据模型，使用不同的数据库语言，建立在不同的操作系统之上，数据的存储结构也各不相同，但从数据库管理系统的角度看，他们在结构上通常都采用三级模式结构，包括：外模式、概念模式和内模式，并提供两级映象功能，通过外模式与概念模式和概念模式与内模式之间的映射，实现数据的独立性。

（3）数据库操纵功能。DBMS 提供数据操纵语言 DML (Data Manipulation Language) 实现对数据库的操作：查询、录入、修改、删除。

（4）程序设计语言。给一般用户使用的数据库系统要求友好的用户界面，使用户能方便、可靠、正确地维护与使用数据库的数据。这就要求有实际应用程序的支持，需要使用程序设计语言开发实际应用系统。程序设计语言可以分为以下类：

- 宿主型 (Host Language)：如 C、Java、Visual Basic 等，用一般的程序设计语言编程，而把 DML 作为主语言的一种扩充嵌入到主语言中。
- 自主型 (Self Contained Language)：DBMS 自含的程序设计语言，可以与 DML 有机

地结合或独立地使用。Visual FoxPro 就属于这类语言, 有自己的编译程序和解释程序。

(5) 数据库运行控制功能。DBMS 主要是通过数据的安全性、完整性、故障恢复和并发控制四方面实现对数据库运行的控制。

(6) 数据库维护功能。包括数据库初始数据装入、数据库转储、数据库重组及记载系统工作日志等功能。

zk2: 2722999990000 01~ 数据模型
MM: xeripnf8f5xc85

求取数据模型常常是设计数据库结构的第一步。 e-mail:

guoxm3@vip.163.com

1.2.1 数据模型概念

数据库存放的是某个企业、组织或部门所涉及的数据的综合, 而且它应能反映数据彼此之间的联系。我们设计数据库系统时, 一般先用图或表的形式抽象地反映数据及数据彼此之间的联系, 称为建立数据模型。常用的数据模型一般可分为两类: 一是语义数据模型, 如实体-关系模型 (E-R 模型)、面向对象模型等; 二是经典数据模型, 如层次模型、网状模型、关系模型。第一类模型强调语义表达能力, 建模容易方便、概念简单清晰, 易于用户理解, 是现实世界到信息世界的第一层抽象, 也是用户和数据库设计人员之间进行交流的语言。第二类模型用于机器世界, 一般和实际数据库对应, 例如层次模型、网状模型和关系模型分别和层次数据库、网状数据库和关系数据库对应, 可以在机器上实现。这类模型有更严格的形式化定义, 常需加上一些限制或规定。我们设计数据库系统时通常利用第一类模型作初步设计, 然后按一定方法转换为第二类模型, 再进一步设计全系统的数据库结构。数据模型通常包括数据结构、数据操作和完整性约束三部分内容。

(1) 数据结构。数据结构描述的是数据库数据的组成、特性及其相互之间的联系。在数据库系统中通常按数据结构的类型来命名数据模型, 例如层次结构、网状结构和关系结构的模型分别命名为层次模型、网状模型和关系模型。

(2) 数据操作。数据操作是指对数据库中各种对象的实例允许执行的操作的集合, 包括操作及有关的操作规则。数据库的操作主要有检索和维护 (包括录入、删除、修改) 两大类。数据模型要定义这些操作的确切含义、操作符号、操作规则及实现操作的语言。数据结构是对系统静态特性的描述。数据操作是对系统动态特性的描述。

(3) 数据的约束条件。数据的约束条件是指数据完整性规则的集合, 它是给定数据模型中数据及其联系所具有的制约和依存规则, 用以限定符合数据模型的数据库状态及其变化, 以保证数据的完整性。数据模型的这三方面中, 数据结构是首要内容, 本节将主要介绍实体-关系 (E-R) 模型和关系数据模型的表示方法及相互之间的关系。

1.2.2 层次数据模型

层次数据模型以实体集为结点, 组成倒置的树型结构, 表示实体集之间的从属关系。上层结点称为父结点, 下层结点称为子结点。其中, 有且仅有一个结点无父结点, 这个结点称为根; 除根以外, 每个子结点有且仅有一个父结点。

一个父结点可以对应多个子结点，表示上层一个机构，与下层有多种业务关系，例如教研室下辖有老师、管理课程，如图 1-2 所示。同时，一个父结点中的一个实体可以对应一个子结点中的多个实体，例如任何一个教研室下辖有多个老师和许多课程，我们称上层实体和下层实体之间是一对多的联系，如图 1-3 所示。

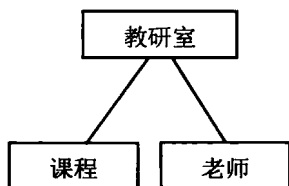


图 1-2 一个实体集对多个实体集

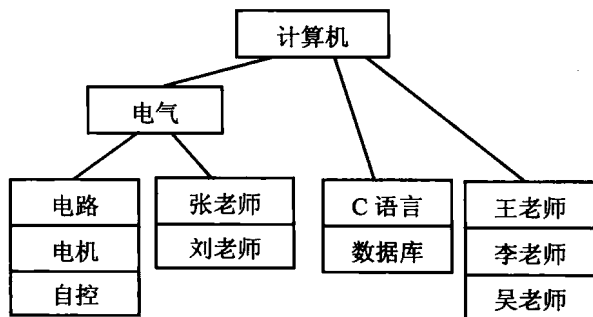


图 1-3 一个实体对多个实体

由此例可见，层次数据模型可以表示组织机构类的数据间的关系。层次结构的优点是结构简单、易于实现。缺点是：支持的联系种类太少，只支持二元一对多联系；数据操纵不方便，子结点的存取只能通过父结点来进行。

1.2.3 网状数据模型

网状数据模型也以实体集为结点，但是可以有一个以上的节点无父节点，有的节点有多于一个的父节点。也就是说，节点间呈现一种交叉关系，每个节点都可和其他多个节点发生关系，如图 1-4 所示。一个节点中的一个实体可以和另一个节点中的多个实体发生关系，另一个节点中的一个实体也可以和前一个节点中的多个实体发生关系，我们称实体和实体之间是多对多的联系。例如：一个老师可以教多门课，有些课有多个老师教，老师和课程间是多对多关系；一个老师可以教多个学生，每个学生都有许多个老师，老师和学生间是多对多关系；一个学生要学多门课，每门课都有许多学生学，学生和课程间是多对多关系等。

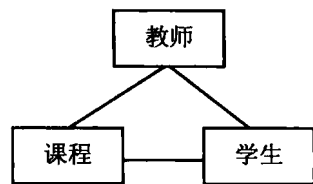


图 1-4 网状数据模型

网状数据模型可以模拟现实世界事物之间的各种复杂关系，但结构过于复杂。

1.2.4 实体-关系模型（E-R 模型）

实体-关系模型（Entity-Relationship Model）是 P.P.S.Chen 于 1976 年提出的一种概念模型，它用 E-R 图来描述一个系统中的数据及其之间的关系。在 E-R 图中，用长方形表示实体集，在长方形框内写上实体名；用菱形表示实体间联系，菱形框内写上联系名；用无向边把菱形和有关实体相连接，在无向边旁标上联系的类型，如 1、M 或 N；用椭圆形表示实体或联系的属性。如果用椭圆形表示，将椭圆形与一个相应实体间以无向边相连。如以表格形式表示，表示格式为：实体名（属性 1，属性 2，...）。