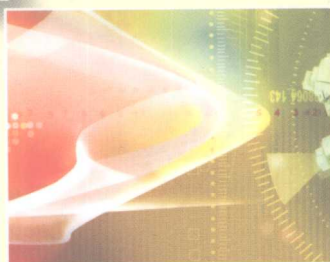
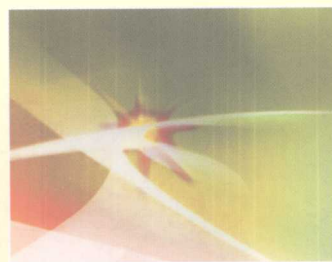




国家级精品课程配套教材
普通高等教育“十一五”国家级规划教材
高职高专计算机系列

Visual FoxPro 程序设计基础教程

刘甫迎 党晋蓉 刘 焱 编著



国家级精品课程配套教材

普通高等教育“十一五”国家级规划教材

高职高专计算机系列

Visual FoxPro 程序设计基础教程

刘甫迎 党晋蓉 刘 焱 编著

人民邮电出版社
北 京

图书在版编目 (CIP) 数据

Visual FoxPro 程序设计基础教程/刘甫迎, 党晋蓉, 刘焱
编著.—北京: 人民邮电出版社, 2008.9

普通高等教育“十一五”国家级规划教材. 高职高专计算机系列

ISBN 978-7-115-18294-4

I. V… II. ①刘…②党…③刘… III. 关系数据库—数据库管理系统, Visual FoxPro—程序设计—高等学校: 技术学校—教材 IV. TP311.138

中国版本图书馆 CIP 数据核字 (2008) 第 085445 号

内 容 提 要

本书是国家级精品课程的配套教材。全书共有 12 章、3 个附录, 涵盖全国计算机等级考试二级 (VFP) 大纲所要求的内容。书中介绍了数据库和表操作、VFP 程序设计、表单和面向对象程序设计、客户/服务器 (C/S) 应用等内容, 并着重加强学生对 SQL 和数据库基本概念的理解与掌握。书中附有教学大纲、实验指导书和精选试卷, 还给出了一个完整的编程案例。

本书内容图文并茂, 强调实用, 可作为高等学校非计算机专业 (VFP 6.0 等级考试) 和计算机类专业 (VFP 9.0) 教材, 也可供广大 VFP 用户参考。

普通高等教育“十一五”国家级规划教材

高职高专计算机系列

Visual FoxPro 程序设计基础教程

- ◆ 编 著 刘甫迎 党晋蓉 刘 焱
责任编辑 张孟玮
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京铭成印刷有限公司印刷
- ◆ 开本: 787×1092 1/16
印张: 17.25
字数: 421 千字 2008 年 9 月第 1 版
印数: 1—3 000 册 2008 年 9 月北京第 1 次印刷

ISBN 978-7-115-18294-4/TP

定价: 29.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154

前 言

微软公司的 Visual FoxPro 是 FoxPro 3.0 版本之后的称谓, 简称 VFP (包括 VFP 6.0、9.0 等)。Fox 数据库管理系统以其强大的功能、很快的运行速度和极好的兼容性, 成为 xBASE 类关系数据库的工业标准之后又一受用户欢迎的产品; 它是既与 SQL 兼容, 又拥有丰富自含命令、函数语言的数据库管理系统, 突出定位于单机和客户/服务器 (C/S) 应用。

20 多年前 (1987 年 5 月), 本书主编刘甫迎教授作为中国科学院公派访问学者从加拿大回国时将 Fox 数据库介绍到中国, 使之取代 dBASE 占微机数据库用户 80% 的地位而普及全国。当时美国每推出一个 Fox 软件新版本, 刘甫迎教授便在国内立即写出一本书来介绍 (从 FoxBASE+1.0、2.0、2.10、FoxGraph 到 FoxPro 1.0、2.0, VFP 3.0、5.0、6.0、8.0)。本书已是作者出版的第 12 部 Fox 方面的书, 并被选为高等教育“十一五”国家级规划教材。刘甫迎教授主持的“Visual FoxPro 程序设计”课程获国家级精品课程称号。

本书的主要特点如下。

1. 全书以 Visual FoxPro 6.0 版本内容为基础, 符合非计算机类专业学生参加全国计算机等级考试的需求。同时, 又在 12.4 等章节具体介绍了使用 Visual FoxPro 9.0 进行 C/S 模式编程的 ADO CursorAdapter 等功能, 适应计算机专业的学习要求。

2. 按全国计算机等级考试二级 (VFP) 大纲的要求, 强化学生对软件工程和数据库基本概念等内容的掌握, 如软件的生存期和开发模型、系统分析与设计及工具 (见第 11 章)、关系、关系运算 (选择、投影、连接)、数据的一致性和完整性等 (见第 1 章)。本书还附有全国计算机等级考试精选参考试题 (见附录 C)。

3. 根据全国计算机等级考试二级 (VFP) 大纲的要求整合有关内容, 本书加强了 SQL 的应用内容 (独立为一章, 见第 6 章), 菜单设计器和报表设计器、VFP 用户界面及可视工具也分别独立写了一章 (第 9 章、第 2 章); 并将项目管理器、共享访问编程及多用户操作函数命令、VFP 的 ActiveX 等内容分别放入第 2 章、第 7 章、第 10 章。

4. 根据学生的学习需要增加了实际案例,如第 11 章“大学生通用习题测试系统”案例,完全是一个使用表单设计器进行面向对象程序设计的实例,学习此系统后学生便能用相同的方法研制实际系统了。

5. 综观全书,作者意图有二:其一,为适合本科非计算机专业或高职学生参加计算机等级考试需要,突出了数据库的基础知识、信息管理的基本方法以及结构化程序设计的三个基本结构编程的内容(见前 9 章),当然这也为面向对象程序设计事件驱动过程中的编程奠定了基础;其二,根据计算机专业的学习需要,本书又突出了面向对象程序设计的内容(见第 10 章等)。由于本书也是作者主持的教育部教改试点专业“计算机信息管理专业”的客户/服务器结构前端(即客户端)数据库的教材,因此还强调了 VFP 与后端(即服务器端)数据库 SQL Server、Oracle 等相连的 ODBC、SQL 视图、ADO 等内容(见第 12 章)。

6. 为了便于学生学习理解,在第 10 章中先介绍表单再引入类、对象、事件、方法的概念,教学实践证明这样安排易于被学生接受。

7. 本书附有教学大纲、实验指导书(见附录 A、附录 B);全书内容图文并茂,强调实用,便于读者参考、使用。

本书由刘甫迎、党晋蓉、刘焱编著。刘甫迎教授编著了第 1、2、3、5、9、10、11、12 章、附录 A、附录 B,并负责全书的统稿。党晋蓉教授编著了第 7 章、第 8 章、4.7 节,并完成全部图稿的工作,刘焱老师编著了第 4 章(不含 4.7 节)、第 6 章、附录 C。

本书内容虽历经多年修改完善,书中难免有不足和不当之处,恳请读者批评指正。

编者

2008 年 4 月

目 录

第 1 章 Visual FoxPro 数据库 管理系统基础.....1	
1.1 数据库的基本概念1	
1.1.1 数据库的定义、体系 结构及数据库系统1	
1.1.2 数据库的由来和发展3	
1.2 关系数据库.....4	
1.2.1 关系数据库的基本概念4	
1.2.2 关系运算6	
1.2.3 数据的一致性和完整性8	
1.3 Visual FoxPro 数据库管理系统9	
1.3.1 Visual FoxPro 功能概述9	
1.3.2 Visual FoxPro 程序结构11	
1.4 Visual FoxPro 的发展历史12	
1.5 Visual FoxPro 的安装与基本操作13	
1.5.1 系统环境要求及安装13	
1.5.2 Visual FoxPro 基本操作14	
1.6 命令窗口16	
习题16	

第 2 章 用户界面、可视工具及 项目管理器18	
2.1 菜单与对话框18	
2.1.1 菜单18	
2.1.2 对话框22	
2.2 工具栏、向导、设计器和生成器25	
2.2.1 工具栏25	
2.2.2 向导27	

2.2.3 设计器.....29	
2.2.4 生成器.....29	
2.3 项目管理器与应用程序.....30	
2.3.1 项目管理器30	
2.3.2 构造一个应用程序33	
2.3.3 添加文件到一个项目34	
2.3.4 生成应用程序与可执 行文件35	
习题36	

第 3 章 Visual FoxPro 的数据37	
3.1 常量37	
3.2 变量38	
3.2.1 字段变量38	
3.2.2 内存变量40	
3.2.3 数组45	
3.2.4 变量的级46	
3.3 表达式47	
3.3.1 运算符47	
3.3.2 表达式分类50	
习题51	

第 4 章 自由表52	
4.1 表文件结构的创建和修改52	
4.1.1 表的结构52	
4.1.2 建立表结构55	
4.1.3 修改表结构56	
4.2 输入表数据58	

4.2.1	CREATE 命令期间 输入数据	58	5.8	其他函数	119
4.2.2	表添加命令	58	5.9	系统设置命令	119
4.2.3	在表中插入记录命令	61	习题		122
4.2.4	追加数据记录命令	62			
4.3	表的基本操作	62	第 6 章 关系数据库标准语言 SQL		123
4.3.1	打开和关闭表文件	62	6.1	SQL 概述	123
4.3.2	表的显示和列表	63	6.2	SQL 的数据定义语言	125
4.3.3	记录定位	65	6.2.1	表结构的建立	125
4.4	修改和删除记录数据	66	6.2.2	表结构的操作	127
4.4.1	修改表数据	67	6.3	SQL 的数据操纵语言	129
4.4.2	删除表记录	70	6.3.1	添加记录	129
4.4.3	删除表文件	72	6.3.2	删除记录	129
4.5	表的排序、索引和查寻	72	6.3.3	更新记录	129
4.5.1	什么是排序与索引	72	6.4	SQL 的数据查询	130
4.5.2	表的排序	73	6.4.1	简单查询	130
4.5.3	表的索引	74	6.4.2	连接查询	132
4.5.4	表的查寻	79	6.4.3	嵌套查询	134
4.6	表的计算、汇总	82	6.4.4	分组计算查询	134
4.6.1	表的计算命令	82	6.4.5	集合的并运算	134
4.6.2	表的汇总命令	83	习题		135
4.7	多表间的基本操作	84	第 7 章 Visual FoxPro 程序设计 基础		137
4.7.1	关联	87	7.1	程序的建立和执行	137
4.7.2	数据工作期窗口与 视图文件	89	7.1.1	程序设计的概念和步骤	137
4.7.3	不同工作区中表的连接	90	7.1.2	程序的建立	138
4.7.4	更新	91	7.1.3	程序的执行	139
习题		92	7.2	程序的交互式命令	139
第 5 章 函数与系统设置命令		95	7.2.1	人机交互式输入语句	139
5.1	数值和数学函数	95	7.2.2	程序设计的辅助命令	141
5.2	三角函数	97	7.3	结构化程序设计	142
5.3	字符串操作函数	97	7.3.1	结构化程序设计的 基本概念	142
5.4	表的操作函数	103	7.3.2	结构化 N-S 流程图	143
5.5	日期函数与 2000 年问题	109	7.3.3	顺序结构	144
5.5.1	日期、时间函数	109	7.3.4	分支结构	145
5.5.2	对 2000 年日期的支持	111	7.3.5	循环结构	148
5.6	显示、打印位置函数	112	7.4	子程序与过程文件及用户	
5.7	系统函数	115			

自定义函数.....151	9.1 菜单程序设计与菜单设计器.....184
7.4.1 子程序与过程的调用.....151	9.1.1 创建菜单、菜单项和子菜单.....184
7.4.2 过程文件的建立.....153	9.1.2 为菜单系统指定任务.....186
7.4.3 用户自定义函数(函数过程).....155	9.1.3 测试与调试菜单系统.....189
7.5 屏幕显示格式程序设计.....156	9.1.4 定制菜单系统.....189
7.6 程序的调试和调试器.....156	9.2 报表程序设计与报表设计器.....190
7.6.1 命令调试.....156	9.2.1 创建报表布局.....191
7.6.2 调试器.....157	9.2.2 一对多报表设计实例.....193
7.7 共享访问编程及多用户操作函数、命令.....158	习题.....199
习题.....162	
第8章 数据库.....166	第10章 表单和面向对象程序设计.....200
8.1 数据库的意义.....166	10.1 表单和表单设计.....200
8.1.1 表和数据库.....166	10.1.1 表单的类型.....201
8.1.2 数据库的优缺点.....167	10.1.2 创建新的表单.....203
8.1.3 数据库的使用范围.....167	10.1.3 添加和操作表单对象.....205
8.2 建立数据库.....167	10.1.4 存储和运行表单.....209
8.2.1 打开、关闭和删除数据库.....168	10.1.5 ActiveX.....210
8.2.2 在数据库中创建新表.....169	10.2 设置表单或表单集模板.....212
8.2.3 添加一个表到数据库.....169	10.3 面向对象的程序设计与对象.....213
8.2.4 使用数据库中的表和视图.....170	10.3.1 对象具有的属性.....214
8.2.5 项目中添加或移去数据库.....170	10.3.2 对象具有的事件和方法.....214
8.2.6 查看和修改数据库结构.....171	10.4 类.....215
8.3 数据字典.....172	10.5 类的层次.....216
8.3.1 字段级和表级数据字典特性.....172	10.5.1 基类和层次.....216
8.3.2 建立存储过程.....180	10.5.2 容器和控件类.....217
8.3.3 建立永久性关系.....180	10.6 对象的处理.....218
8.3.4 生成参照完整性.....182	10.6.1 在容器层次中引用对象.....218
8.3.5 查看和设置数据库属性.....183	10.6.2 设置属性.....219
习题.....183	10.6.3 调用方法.....219
	10.6.4 响应事件.....220
	10.7 编程定义类.....221
	习题.....229
第9章 菜单设计器和报表设计器.....184	第11章 设计应用程序案例——大学生通用习题测试系统.....230
	11.1 软件工程的基本概念及软件开发技术.....230

11.1.1 软件的生存期和开发模型	230	12.4 使用 ADO 数据访问进行 C/S 模式编程	252
11.1.2 系统分析与设计及工具	231	习题	254
11.2 大学生通用习题测试系统	233	附录 A Visual FoxPro 程序设计基础教学大纲	255
11.2.1 系统的表单流程图	233	附录 B 实验指导书	257
11.2.2 系统的实现及表单对象的应用	234	附录 C 计算机等级考试二级 (VFP) 参考试题	262
习题	242	附录 C.1 上机考试样题	262
第 12 章 视图和客户/服务器编程	243	附录 C.2 二级 Visual FoxPro 程序设计试题样卷	263
12.1 客户/服务器结构	243	参考文献	268
12.2 创建和使用视图	244		
12.2.1 创建本地视图	246		
12.2.2 创建多表视图	246		
12.2.3 使用视图	248		
12.3 更新视图数据	250		

第 1 章 Visual FoxPro 数据库 管理系统基础

本章首先介绍数据库的意义、数据库的体系结构、数据库管理系统及数据库系统、数据库的由来和发展、关系模型、关系运算（选择、投影、连接）、数据的一致性和完整性等内容，以便读者对数据库的概念有一个基本的了解。然后介绍 VFP 数据库的发展与特点、VFP 程序的结构，VFP 的安装、启动、窗口及退出，用命令窗口工作等，为后续章节的学习提供了基础。

本章学习重点：数据库概念、VFP 数据库的发展与 VFP 的特点。

本章学习难点：关系运算（选择、投影、连接）。

C 1.1 数据库的基本概念

1.1.1 数据库的定义、体系结构及数据库系统

1. 数据库

数据库（Database，DB）是存储在一起的相关数据的集合，是存储数据的“仓库”。数据库一词源于 20 世纪 50 年代，美国为战争的需要，把各种情报集中在一起，存入计算机，这样就有了数据库的概念。60 年代数据库帮助设计了阿波罗登月火箭，推动了数据库技术的产生。70 年代，数据库蓬勃发展，网状系统和层次系统占领了市场，并且关系数据库理论也日趋完善。80 年代起，关系数据库产品已相当成熟，后来有了分布式数据库系统（Distributed database systems），数据库机（Databasemachines）、知识库（Knowledge base）系统和知识机。

现在，数据库技术已成为计算机领域最重要的技术之一，它是软件学科中一个独立分支（即研究数据库的结构、存储、设计和使用的——门软件学科）。

2. 数据模型

数据库离不开数据模型。数据模型是对现实世界客观事物及其联系的描述，它反映数据项之间和记录之间的联系，在数据库技术中使用模型的概念描述数据库的结构与语义。常用的三种数据模型是：层次模型（hierarchical model）、网状模型（network model）和关系模型

(relational model)。此外，还有面向对象模型 (object-oriented model) 等。

3. 数据库体系结构

数据库的体系结构分成内部级 (internal)、概念级 (conceptual) 和外部级 (external) 三级。图 1-1 所示的结构是 1975 年 2 月美国 ANSI/X3/SPARC (国家标准化委员会/关于计算机和信息处理/系统设计和要求委员会) 提出的。

从某个角度看到的数据特性称为“数据视图”。外部级最接近用户，是单个用户所能看到的数据特征。单个用户使用的数据视图称为“外模型”。概念级是涉及所有用户的数据的定义，也就是全局的数据视图，称为“概念模型”。内部级最接近于物理存储设备，涉及实际数据存储的方式。物理存储的数据视图称为“内模型”。这些模型有两种表达方式，一种是用图示形式，即数据结构图；另一种是用数据定义语言描述。这三种模型用数据定义语言描述分别得到外模式 (亦称子模式)、概念模式 (亦称模式) 和内模式 (亦称存储模式)。

数据库的三级体系结构是数据的三个抽象级别，它把数据的具体组织留给数据库管理系统管理，使用户能抽象地处理数据，而不必关心数据在计算机中的表示和存储。这三级结构之间往往差别很大，为实现这三个抽象级别的转换，数据库管理系统在这三级结构之间提供外模式/模式映像，模式/内模式映像两层映像 (mappings)。

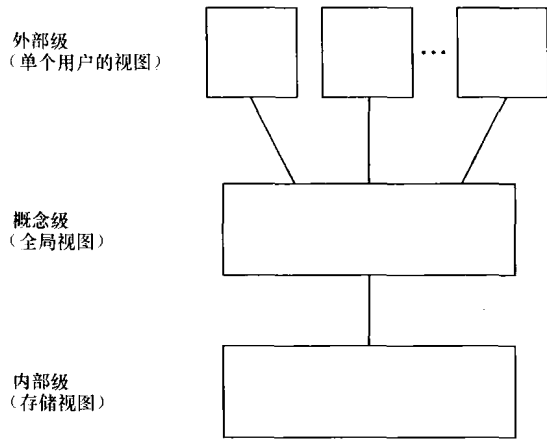


图 1-1 数据库的三级结构

4. 数据库管理系统及数据库系统

数据库加上相应的操作和管理数据库的软件，即为一个数据库管理系统 (DBMS)，它能合理地组织和存储大量数据，并支持对于数据库表中数据的各种操作，如更新、排序、索引、查询、列表、打印等。

数据库系统是现实有组织地、动态地存储大量的相关数据，方便用户访问的计算机软硬件资源组成的系统。它由数据库、数据库管理系统，以及支撑数据库管理系统的软硬件构成。

在计算机的三大主要应用 (科学计算、数据处理与过程控制) 中，数据处理所占比重约为 70% 左右。数据库技术就是作为数据处理中的一门技术发展起来的，今天它已广泛地应用到工业、农业、商业、金融、财贸等行业，以及国家机关、大专院校、科研、国防军事、情报等部门中的管理信息系统 (MIS)、办公自动化系统 (OA)、决策支持系统等领域中。

1.1.2 数据库的由来和发展

数据管理技术大致经过以下4个阶段：人工管理阶段、文件系统阶段、数据库阶段和高级数据库阶段。

1. 人工管理阶段

这一阶段（20世纪50年代中期以前）的计算机主要用于科学计算。硬件中的外存只有卡片、纸带等，软件只有汇编语言，没有数据管理方面的软件。数据处理的方式基本上是批处理，数据不保存；每个应用程序要包括存储结构、存取方法、输入/输出方式等，由于各应用程序处理的数据不会毫无联系，程序之间会有重复；对数据的存取以记录为单位。

2. 文件系统阶段

这一阶段（20世纪50年代后期至60年代中后期）的计算机不仅用于科学计算，还大量用于信息管理。外存已有磁盘、磁鼓等直接存储设备，软件方面出现了高级语言和操作系统。

数据可长期保存在外存的磁盘上；数据的物理结构与逻辑结构有了区别；程序与设备之间有设备独立性（程序只需用文件名与数据打交道，不必关心数据的物理位置）；文件的形式已多样化，有直接存取等，因而对文件的记录可顺序访问，也可随机访问；基本上还是以记录为单位。缺陷：数据冗余性，不一致性，数据联系弱。

3. 数据库阶段

数据管理技术进入数据库阶段的标志是20世纪60年代后期的三大事件：即1969年IBM公司推出的IMS产品（层次数据库系统）和CODASYL研究和建议的DBTG系统（网状数据库系统），以及1970年起，IBM公司E.F.Codd连续发表一系列论文，奠定了关系数据库理论基础。

70年代以来，数据库技术得到迅速发展，层次、网状数据库系统投入使用。70年代末推出了关系数据库一些试验系统，80年代初出现了一批商品化的关系数据库系统，如Oracle、SQL/DS、DB2、INGRES、INFORMIX、UNIFY以及dBASE、FoxBASE等。SQL在1986年被美ANSI和国际标准化组织（ISO）采纳为关系数据库语言的国际标准。

数据库技术的管理方式具有以下特点。

（1）采用复杂的数据模型（结构）。

数据模型不仅要描述数据本身的特点，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。通过一切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。这样，数据不再面向特定的某个或多个应用，而是面向整个应用系统。数据冗余明显减少，实现了数据共享。

（2）有较高的数据独立性。

在数据库系统中，系统提供映像的功能，确保应用程序对数据结构和存取方法有较高的独立性。数据库结构分成用户的逻辑结构、整体逻辑结构和物理结构。在改变物理结构时，由于模式/内模式映像不影响整体逻辑结构、用户的逻辑结构以及应用程序，这样就认为数据库达到了物理数据独立性。在改变整体逻辑时，由于外模式/模式映像不影响用户的逻辑结构

及应用程序，这样就认为数据库达到了逻辑数据独立性。

(3) 数据库系统为用户提供了方便的用户接口。

可使用简单的终端命令操作数据库，也可用程序方式（用数据库操纵语言）操作数据库。

(4) 提供下述 4 方面的数据控制功能。

① 数据完整性：保证数据库始终包含正确的数据。

② 数据安全性：保证数据的安全和机密，防止数据丢失或被窃取。

③ 数据库的并发控制：避免相互干扰，防止数据被破坏，杜绝给用户不正确的数据。

④ 数据的恢复：在数据库被破坏时，系统能把数据库恢复到最近某时刻的正确状态。

(5) 可以按数据项为单位操作。

从文件系统发展到数据库技术，是信息处理领域的一个重大变化。在文件系统阶段程序设计处于主导地位，数据只起着服从程序设计需要的作用；而在数据库方式下，数据开始占据了中心位置，数据的结构设计成为信息系统首先关心的问题，而利用这些程序的应用程序设计则退到以既定的数据结构为基础的外围地位。

4. 高级数据库阶段

这一阶段（20 世纪 70 年代后期开始）的主要标志是分布式数据库系统、面向对象数据库、智能数据库系统的出现。面向对象程序设计是软件工程的重要发展方向之一。为了满足各种新的数据应用的需要，诸如多媒体数据、空间数据、复杂对象、超本文、知识、时序数据等的管理，面向对象数据库引起广泛兴趣。

1.2 关系数据库

1.2.1 关系数据库的基本概念

1. 关系、关系模型

关系模型把数据之间的关系看成一个二维表关系。因二维表关系是建立在集合代数关系理论基础上的，所以把这种建立在关系模型基础上的数据库称为关系数据库。表 1-1 所示的二维表就是关系数据库中的一个表文件，即是一个关系。

表 1-1 数据库中的一个表

姓 名	地 址	城 市	省 名	邮 编	电 话 号 码	编 号
刘先生	成都小狐狸软件公司	成都	四川	610061	15982831806	0003
李红	581 信箱	北京	北京	010000	6775041	0001
郭亮	457 信箱	宜宾	四川	620000	398883	0002

一个数据库的信息通常组织和存储在表（Table）的形式中，各个表具有行和列。在表 1-1 中，各行包括姓名、地址、城市、省名、邮编、电话号码、编号。各行与另外的行相关，因为它们全都包括相同类型的信息，又具有“姓名”列、“地址”列、“城市”列等属性，如“姓名”列中有刘先生、李红、郭亮三个数据（即属性值）。这样由行和列组成一个表，即一个表

文件。一个表文件中，行叫表的记录，列叫表的字段。即在表中，描述对象的项目称为数据项（又称字段），所有数据项组成一个记录，若干个同类记录构成表文件。若干个表以及它们的关系等用来管理数据，也叫数据库文件。

关系有如下性质。

- (1) 一个域中的各项目是同类的，即每一列中的分量是同一类型的数据。
- (2) 不同的列可以出自同一个域，每一列称为属性，需给予不同的名称。
- (3) 列的顺序无所谓，即列的次序可以任意交换。
- (4) 关系中的各个元组是不同的，即不允许有重复的元组。
- (5) 行的顺序无所谓，即行的次序可以任意交换。
- (6) 每一分量必须是不可分的数据项。

关系模型由数据结构、关系操作集合和关系的完整性约束条件集合三部分组成。

2. 数据结构（元组、属性、域、主关键字、外部关键字）

二维表中行和列分别对应记录（也叫元组）和字段（也叫属性），元组中一个属性的值叫分量。每个属性有一个值域，值域的类型可以是整数型、实数型和字符型等（均有长度），例如账户号码属性值域的类型可以是所有的正数。但要注意的是字段又有关键字（段）与一般字段之分：在一个关系中用来区分不同记录的属性（字段）称之为关键字（段），而可以用来唯一标识一条记录的属性称之为主关键字。显然上表中，姓名、编号都可以被指定为关键字，但只有“编号”才可以起到主关键字（Primary key）的作用。外部关键字（Foreign key）是某个关系中的一个属性（可以是一个普通的属性，也可以作为主码的一个属性，也可以作为主码的一部分的一个属性），这个属性在另一个关系中是主关键字。

可见关系模式（关系数据结构）是对关系的描述，它包括关系名、组成该关系的属性名、属性到域的映像。通常记为：

关系名(属性名 1, 属性名 2, ..., 属性名 n)

属性到域的映像通常为属性的类型、长度等。

表 1-1 的关系模式可记为：

通信地址表(姓名,地址,城市,省名,邮编,电话号码,编号)

对关系数据库的描述称为关系数据库的型，包括若干域定义和在这些域上定义的若干关系模式。这些关系模式在某一时刻对应的关系的集合称为关系数据库的值。

图 1-2 所示为一个使用关系数据库的例子。

【例 1.1】 现有两个文件（见图 1-2），通信地址表存储了百货仓库的用户信息，仓库也有一个单独的用户订单数据库表，后者包括用户号、商品号、单价、订货数量和总价字段。现在如果有一个查询：“找出商品号为 14B 的用户的姓名、地址？”

即使这两个表的数据已在磁盘中，文件系统要回答这个查询，一般也要编一个程序，长达几页纸。原因就是在文件系统中，这两个文件之间缺乏联系。如果预先用指针把两个文件的记录联系起来，那么回答上述查询就可以很简单，很迅速，关系数据库解决这类问题是没有困难的。如图 1-2 所示，从这两个文件表可以看到通信地址表 and 用户订单表由于具有相同的用户号字段，而构成了一个关系数据库。通过在通信地址表中查寻用户号和匹配它在订单表中的用户号，数据库管理程序可以从一个表中确定订货者是谁和他的地址，同时从另一个

表确定该订货者订了什么货。本例通过用户号 0002 联系，找寻到订 14B 货物的人是郭亮，地址为 457 信箱（四川宜宾）。关系数据库管理系统就这样通过从一个相同字段连接不同的表中获取了信息。类似地也可定义 3 个以上表之间的各种联系。

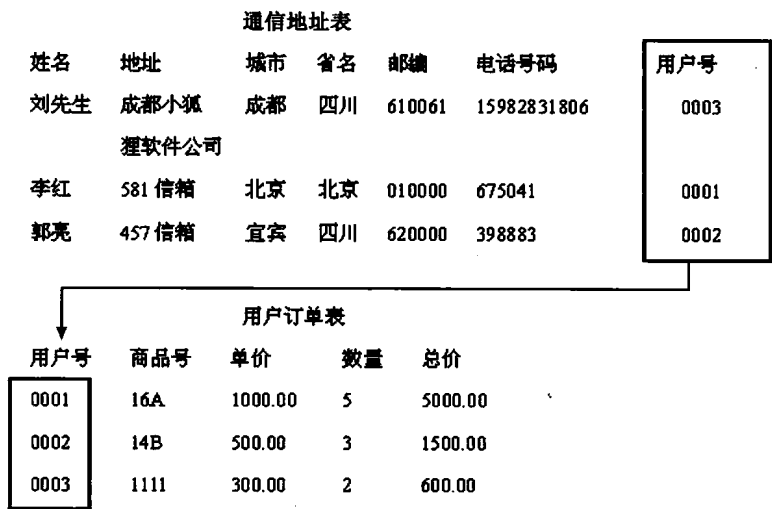


图 1-2 两个有关系的表

可见，数据库与文件系统不同，其数据独立于程序而存在，并可以提供给不同的用户共享使用，其基本思想是对所有的数据实行统一的、集中的、独立于程序的管理。关系数据库管理系统很好地实现了这一基本思想。

1.2.2 关系运算

1. 关系的数据操纵语言

关系数据库的数据操作（关系操作集合）语言包括查询（描述用户要进行的各种检索操作）和非查询（描述用户要进行的插入、修改、删除等操作）两类，查询是最主要的部分。

关系的数据操纵语言按照表达查询的方式可分为以下两大类。

（1）关系代数语言，查询是以集合操作为基础的运算。

（2）关系演算语言，查询是以谓词演算为基础的运算。按谓词变元的基本对象是元组变量（Tuple variable）还是域变量（Domain variable）分为元组关系演算和域关系演算两种。

这两种语言在表达上是彼此等价的。它们都是抽象的查询语言，与具体的数据库管理系统中实现的实际语言并不完全一样。实际的查询语言还提供了许多附加的功能，如算术运算、库函数、系赋值等功能。

2. 关系代数运算

（1）关系代数运算的分类

关系代数的运算对象是关系，运算的结果也是关系。关系代数的运算可分为如下两类。

① 传统的集合运算（如并、交、差、广义笛卡儿积）把关系看作元组的集合，其运算是从行的角度进行的，较简单不详述。

② 专门的关系运算，如选择、投影、连接、除。这类运算不仅涉及行，而且涉及列。

(2) 选择 (Select)

也称为限制 (Restriction)，是根据某些条件对关系作水平分解，选择符合条件的元组。条件可用命题公式 F 表示， F 取值为‘真’或‘假’。运算符有算术比较运算符 ($<$, $\leq$, $>$, $\geq$, $\neq$, $=$) 和逻辑运算符 ($\wedge$, $\vee$, $\neg$)。逻辑运算符 $\neg$ 可定义为：如果逻辑表达式 P 为真，则 $\neg P$ 为假。关系 R 关于公式 F 的选择运算用 $\sigma_F(R)$ 表示，定义如下：

$$\sigma_F(R) = \{t | t \in R \wedge F(t) = \text{'真'}\}$$

设有贷款关系，如表 1-2 所示。

表 1-2 贷款情况关系 (LOAN)			
分行名	贷款编号	顾客名	余额
Branch_A	17	Customer_A	1 000
Branch_A	18	Customer_B	2 000
Branch_A	19	Customer_C	15 000
Branch_B	15	Customer_D	1 500
Branch_B	16	Customer_E	1 300
Branch_C	20	Customer_F	1 000
Branch_D	21	Customer_G	3 000

【例 1.2】求 Branch_B 分行贷款金额大于 1 350 元的所有元组。

$$\sigma_{\text{分行名}=\text{'Branch_B'} \wedge \text{金额}>1350}(\text{LOAN})$$

结果如表 1-3 所示。

表 1-3 结果			
分行名	贷款编号	顾客名	余额
Branch_B	15	Customer_D	1 500

(3) 投影 (Projection)

这个操作是对关系作垂直分解，消去关系中某些列，并重新排列次序，删去重复元组。

设关系 R 是 k 元关系， R 在其分量 $A_{i1}, \dots, A_{im}$ ($m \leq k, i_1, \dots, i_m$ 为 1 到 k 之间互不相同的整数) 上的投影，用 $\pi_{i1, \dots, im}(R)$ 表示，它是一个 m 元元组组成的集合。形式定义如下：

$$\pi_{i1, \dots, im}(R) = \{t | t = \langle t_{i1}, \dots, t_{im} \rangle \wedge \langle t_1, \dots, t_k \rangle \in R\}$$

【例 1.3】求贷款关系 LOAN 在贷款编号和余额这两个属性上的投影。

$$\Pi_{\text{贷款编号, 余额}}(\text{LOAN})$$

投影的结果如表 1-4 所示。

表 1-4 结果	
贷款编号	余额
17	1 000
18	2 000
19	15 000
15	1 500
16	1 300
20	1 000
21	3 000

投影之后不仅取消了某些列，而且还可能取消某些元组。因为取消了某些属性之后，就可能出现重复行，应该取消这些相同的行。

(4) 连接 (Join)

也称为 θ 连接，连接是从两个关系的笛卡儿积中选取属性间满足一定条件的元组。记作： $R \bowtie_{\theta} S$ ，这里 i 和 j 分别是关系 R 和 S 中的第 i 、第 j 个属性， θ 是算术比较运算符。连接的定义如下：

$$R \bowtie_{\theta} S = \{ \overline{t_r \ t_s} \mid t_r \in R \wedge t_s \in S \wedge t_r[i] \theta t_s[j] \}$$

这里， R 是 n 目关系， S 是 m 目关系， $t_r \in R, t_s \in S, \overline{t_r \ t_s}$ 称为元组的连串 (Concatenation)。这是一个 $(n+m)$ 元组，前 n 个分量为 R 中的一个 n 元组，后 m 个分量是 S 中的一个 m 元组。

连接运算从 $R \bowtie S$ 的广义笛卡儿积中选取 R 关系在 i 属性组上的值与 S 关系在 j 属性组上满足比较关系 θ 的元组。当 θ 为 “=” 时称为等值连接。

自然连接 (Natural join) 是一种特殊而常用的连接。若 R 和 S 具有相同的属性组 i ，则自然连接定义如下：

$$R \bowtie S = \{ \overline{t_r[i] t_s[i]} \mid t_r \in R \wedge t_s \in S \wedge t_r[i] = t_s[i] \}$$

自然连接要求两个关系中相等的分量必须是相同属性组，而等值连接不必。自然连接要在结果中把重复的属性去掉。

【例 1.4】 设关系 R, S 分别如表 1-5 中的 (a)、(b) 所示， $R \bowtie_{\theta} S$ 的结果如表 1-6 (a) 所示， $R \bowtie S$ 自然连接的结果如表 1-6 (b) 所示。

表 1-5

R	A	B	C
	a_1	b_1	15
	a_1	b_2	16
	a_2	b_3	18
	a_2	b_4	22
	a_3	b_5	30

(a)

关系 R, S

S	B	D
	b_1	13
	b_2	16
	b_3	20
	b_3	12
	b_6	12

(b)

表 1-6

A	$R.B$	C	$S.B$	D
a_1	B_1	15	b_2	17
a_1	b_1	15	b_3	20
a_1	b_2	16	b_2	17
a_1	b_2	16	b_3	20
a_2	b_3	18	b_3	20

(a)

结果

A	B	C	D
a_1	b_1	15	13
a_1	b_2	16	17
a_2	b_3	18	20
a_2	b_3	18	12

(b)

1.2.3 数据的一致性和完整性

关系模型的完整性是实体完整性 (Entity Integrity)、参照完整性 (Referential integrity)