

21世纪大学计算机系列教材

Visual FoxPro 6.0

程序设计教程 (第3版)

丁照宇 孙淑霞 李思明 等编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>



21世纪大学计算机系列教材

Visual FoxPro 6.0 程序设计教程 (第3版)

丁照宇 孙淑霞 李思明 刘焕君 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING



内 容 简 介

本书共 13 章,深入浅出地介绍数据库的有关知识, Visual FoxPro 6.0 的基础知识、基本操作和基本技能,以及数据库设计和开发的有关知识。本书融理论与实验为一体,在编写上力求通俗易懂。书中用大量的实例使读者更快地熟悉 Visual FoxPro 的可视化编程环境;所有操作步骤都按实际操作界面一步一步地讲解,读者可一边学习,一边上机操作。通过本教材的学习,读者将对使用 Visual FoxPro 6.0 进行数据库软件开发有一个较完整的认识,并使读者能掌握开发数据库系统的基本思想和方法,逐步具备数据库管理系统的设计、应用和开发能力。

本书可作为大专院校非计算机专业的教材,对从事数据库应用和开发的读者也有很好的参考价值。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Visual FoxPro 6.0 程序设计教程/丁照宇等编著. —3 版. —北京:电子工业出版社,2010.1

(21 世纪大学计算机系列教材)

ISBN 978-7-121-10094-9

I. V… II. 丁… III. 关系数据库—数据库管理系统, Visual FoxPro 6.0—程序设计—教材
IV. TP311.138

中国版本图书馆 CIP 数据核字(2009)第 232748 号

策划编辑:王昌铭

责任编辑:胡丽华

印 刷:北京丰源印刷厂

装 订:三河市万和装订厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编:100036

开 本:787×1092 1/16 印张:23 字数:601 千字

印 次:2010 年 1 月第 1 次印刷

印 数:4000 册 定价:36.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

第3版前言

本书以 Visual FoxPro 6.0 中文版为平台,结合普通高校非计算机专业数据库程序设计课程的具体要求,深入浅出地介绍 Visual FoxPro 6.0 数据库程序设计的有关知识、方法和具体的实例。

本教材共 13 章。前 12 章分别介绍数据库系统概述, Visual FoxPro 6.0 的基础知识, Visual FoxPro 数据库管理系统所支持的数据类型、函数、表达式、数组、属性、事件、方法、对象、表和数据库的操作, Visual FoxPro 6.0 的编程工具和操作步骤,面向对象的可视化编程、表单操作、菜单与工具栏、数据的检索、视图的更新和报表的设计等。第 13 章在前 12 章的基础上,以“OICQ 号码管理系统”为实例,讲述如何开发数据库应用系统,并给出一个较完整的解决方案。

教材编写的重点是数据库系统的基本概念, Visual FoxPro 6.0 程序设计的基本方法和基本技能。内容的安排强调循序渐进、前后呼应;每章开始有本章要点,章末有上机实验内容和一定量的习题。实验内容有助于读者理论联系实际,提高实际操作和编程的能力,对于有一定难度的实验内容,书中给出了具体的指导,使读者按照书中给出的操作步骤就能够完成规定的实验内容。习题有利于帮助读者自学和检查学习效果。书末给出各章习题的参考答案,以便于读者检查。

本书的作者多年来一直从事计算机基础教学,经过多年的教学实践编写了这本教材。针对初学者和自学读者的特点,本书力求通俗易懂,用大量具体的操作、各种不同的实例让读者进入 Visual FoxPro 的可视化编程环境。所有步骤都按实际操作界面一步一步地讲解,读者可一边学习,一边上机操作,通过一段时间的练习,在不知不觉之中就可逐渐掌握 Visual FoxPro 6.0 数据库程序设计的基础知识、设计思想和方法以及可视化编程的方法和步骤,并提高利用 Visual FoxPro 6.0 解决实际问题的能力。

本教材第 1 版自 2004 年 1 月出版以来,许多读者对该书提出了宝贵的意见。这次,我们结合多年的教学实践,对书中的一些内容进行了修改,对某些部分进行了删减和增添,对部分章节的内容做了调整,具体修改如下:

1. 对部分章节的内容做了调整。
2. 改正了原书的个别错误,在内容上力求更加精练和准确。
3. 在原书的基础上,更加突出了实验和实验指导,增强了编程能力的训练。
4. 有针对性地增加了部分习题,使读者可以更有效地进行复习和测试。

最后,我们衷心地感谢对本书提出宝贵意见的老师和读者。

本书第 1~3 章、第 7~8 章由丁照宇编写,第 4~6 章、第 9~10 章由孙淑霞编写,第 11~12 章由刘焕君编写,第 13 章由李思明、陈晓丹、丁照宇编写。陈佩良、肖阳春、彭舰、黄于鉴等老师也参加了本书的部分编写、校对和程序调试等工作。由于作者的水平有限,书中难免有错误和不妥之处,恳请读者不吝赐教。

编著者电子邮件地址: dzy@cdut.edu.cn

编著者
2009 年 10 月

目 录

第 1 章 概述	(1)
1.1 数据库系统概述	(1)
1.1.1 数据库系统基础知识	(1)
1.1.2 数据库的数据模型	(4)
1.1.3 关系模型	(7)
1.2 Visual FoxPro 6.0 系统概述	(9)
1.2.1 数据库管理系统概述	(9)
1.2.2 Visual FoxPro 6.0 数据库管理系统的特点	(10)
1.2.3 Visual FoxPro 6.0 的技术指标	(10)
1.2.4 Visual FoxPro 6.0 的文件类型	(12)
1.2.5 Visual FoxPro 6.0 的启动与退出	(12)
习题 1	(13)
第 2 章 Visual FoxPro 6.0 基础知识	(16)
2.1 Visual FoxPro 6.0 的用户界面	(16)
2.2 Visual FoxPro 6.0 的工作方式及命令语法规则	(18)
2.2.1 Visual FoxPro 6.0 的工作方式	(19)
2.2.2 命令语法规则	(19)
2.3 Visual FoxPro 的项目管理器	(21)
2.3.1 项目管理器的使用	(21)
2.3.2 项目文件的创建	(26)
2.4 Visual FoxPro 6.0 的设计器	(29)
2.5 Visual FoxPro 6.0 的向导	(29)
2.5.1 启动向导	(30)
2.5.2 向导的使用	(30)
2.6 生成器简介	(31)
习题 2	(32)
本章实验	(33)
第 3 章 Visual FoxPro 的常量、变量、表达式和函数	(35)
3.1 Visual FoxPro 6.0 的数据类型	(35)
3.1.1 数据类型	(35)
3.1.2 常量与变量	(37)
3.2 表达式	(40)
3.2.1 运算符	(40)
3.2.2 Visual FoxPro 6.0 的表达式	(42)
3.3 常用函数	(44)
3.3.1 数学运算函数	(44)
3.3.2 字符和字符串处理函数	(45)
3.3.3 转换函数	(48)
3.3.4 日期函数	(50)
3.3.5 测试函数	(51)

3.3.6 其他函数	(53)
习题 3	(55)
本章实验	(58)
第 4 章 表的基本操作	(62)
4.1 创建自由表	(62)
4.1.1 表的概念	(62)
4.1.2 表结构的设计	(63)
4.1.3 表结构的建立	(64)
4.1.4 表数据的键盘输入	(72)
4.1.5 将已有数据添加到表中	(74)
4.1.6 表结构的修改	(76)
4.2 表记录的基本操作	(77)
4.2.1 表的打开和关闭	(77)
4.2.2 查看表中的数据	(78)
4.2.3 记录指针的定位	(83)
4.2.4 记录的插入和追加	(86)
4.2.5 记录的删除与恢复	(89)
4.2.6 表数据的替换	(91)
4.3 表数据的排序与索引	(93)
4.3.1 排序	(93)
4.3.2 索引	(95)
4.3.3 建立索引	(96)
4.3.4 使用索引	(98)
4.3.5 索引查找	(102)
4.4 计数、求和与汇总	(103)
4.4.1 计数命令 COUNT	(103)
4.4.2 求和命令 SUM	(103)
4.4.3 求平均命令 AVERAGE	(104)
4.4.4 汇总命令 TOTAL	(104)
4.5 多个表的同时使用	(106)
4.5.1 多工作区的概念	(106)
4.5.2 工作区的选择	(106)
4.5.3 建立表的关联	(109)
习题 4	(114)
本章实验	(124)
第 5 章 数据库的基本操作	(131)
5.1 数据库的创建	(131)
5.1.1 创建数据库文件	(131)
5.1.2 数据库的打开和关闭	(133)
5.1.3 在数据库中操作表	(134)
5.2 数据库表属性的设置	(137)
5.2.1 设置字段显示属性	(137)
5.2.2 设置字段输入默认值	(139)
5.2.3 设置有效性规则	(139)

5.2.4	设置触发器	(141)
5.2.5	建立参照完整性	(142)
5.3	数据库的操作	(143)
5.3.1	数据库表的查看	(143)
5.3.2	数据库结构的查看与修改	(144)
5.3.3	在项目中添加数据库	(145)
5.3.4	从项目中移去和删除数据库	(145)
5.3.5	使用多个数据库	(145)
习题 5		(146)
本章实验		(148)
第 6 章	结构化程序设计	(150)
6.1	程序的建立和运行	(150)
6.1.1	源程序的建立和保存	(150)
6.1.2	程序的运行	(151)
6.1.3	程序的书写规则和编辑技巧	(151)
6.2	程序设计中的常用语句	(152)
6.2.1	? ??输出命令	(152)
6.2.2	WAIT 输入命令	(153)
6.2.3	格式输入/输出命令	(153)
6.2.4	CLEAR 清屏命令	(154)
6.2.5	ACCEPT 内存变量接收命令	(154)
6.2.6	INPUT 内存变量输入命令	(155)
6.3	程序的控制结构	(155)
6.3.1	顺序结构	(155)
6.3.2	分支结构	(155)
6.3.3	循环结构	(159)
6.4	过程与用户自定义函数	(163)
6.4.1	过程及过程的调用	(164)
6.4.2	用户自定义函数	(166)
6.4.3	变量的作用域	(167)
6.4.4	程序的调试方法	(169)
习题 6		(172)
本章实验		(182)
第 7 章	面向对象程序设计	(190)
7.1	面向对象编程概述	(190)
7.1.1	从面向过程到面向对象	(190)
7.1.2	深入理解对象	(191)
7.1.3	深入了解类	(193)
7.2	Visual FoxPro 中的类和对象	(195)
7.2.1	Visual FoxPro 的类	(195)
7.2.2	容器类与控件类	(196)
7.3	Visual FoxPro 6.0 的编程工具与编程步骤	(197)
7.3.1	Visual FoxPro 6.0 表单设计器	(197)
7.3.2	Visual FoxPro 6.0 中的事件	(203)

7.3.3	Visual FoxPro 6.0 的方法程序	(205)
7.3.4	Visual FoxPro 6.0 的编程步骤	(205)
7.4	整理表单	(207)
7.4.1	编辑控件	(207)
7.4.2	设置控件的 Tab 键次序	(208)
习题 7		(209)
本章实验		(211)
第 8 章	表单控件的使用	(215)
8.1	线条与形状控件	(215)
8.1.1	使用线条控件	(215)
8.1.2	使用形状控件	(216)
8.2	命令按钮类控件	(216)
8.2.1	创建数据环境	(216)
8.2.2	命令按钮	(218)
8.2.3	命令按钮组	(220)
8.3	标签、文本框和编辑框控件	(223)
8.3.1	标签和文本框	(223)
8.3.2	编辑框	(228)
8.4	选项按钮组和复选框	(230)
8.4.1	选项按钮组	(230)
8.4.2	使用复选框	(231)
8.5	列表框、组合框和页框	(231)
8.5.1	列表框	(232)
8.5.2	组合框	(234)
8.5.3	页框	(235)
8.6	其他常用控件	(237)
8.6.1	容器控件	(237)
8.6.2	微调控件	(238)
8.6.3	图像控件	(239)
8.6.4	计时器控件	(240)
8.6.5	表格控件	(241)
8.7	表单集	(243)
8.7.1	创建表单集	(243)
8.7.2	表单集的应用	(244)
习题 8		(245)
本章实验		(247)
第 9 章	结构化查询语言 (SQL)	(256)
9.1	SQL 语言概述	(256)
9.1.1	SQL 语言的特点	(256)
9.1.2	数据定义语言	(257)
9.1.3	数据操纵语言	(260)
9.1.4	创建临时表	(262)
9.2	SQL 的数据查询功能	(262)
9.2.1	查询语句	(262)

9.2.2 查询分类.....	(263)
习题 9	(270)
本章实验	(273)
第 10 章 查询与视图.....	(275)
10.1 查询.....	(275)
10.1.1 查询的概念.....	(275)
10.1.2 使用向导创建查询.....	(276)
10.1.3 使用查询设计器创建查询.....	(280)
10.2 视图.....	(285)
10.2.1 视图的概念.....	(285)
10.2.2 使用视图设计器建立本地视图.....	(285)
10.2.3 视图与表、视图与查询的比较.....	(289)
习题 10	(290)
本章实验	(291)
第 11 章 菜单设计.....	(293)
11.1 菜单设计概述.....	(293)
11.1.1 创建菜单系统.....	(293)
11.1.2 规划菜单系统.....	(294)
11.2 创建菜单.....	(294)
11.2.1 使用菜单设计器创建菜单.....	(295)
11.2.2 使用快速菜单命令创建菜单.....	(302)
11.2.3 创建快捷菜单.....	(303)
11.2.4 有关菜单的其他操作.....	(304)
习题 11.....	(306)
本章实验	(307)
第 12 章 报表和标签的设计.....	(309)
12.1 创建报表.....	(309)
12.1.1 使用报表向导创建报表.....	(309)
12.1.2 使用报表设计器创建报表.....	(311)
12.1.3 创建快速报表.....	(312)
12.2 设计报表.....	(313)
12.2.1 设置报表数据源.....	(313)
12.2.2 设计报表布局.....	(314)
12.2.3 利用控件设计报表.....	(314)
12.3 设计分组报表.....	(318)
12.3.1 设计报表的记录顺序.....	(318)
12.3.2 设计单级分组报表.....	(319)
12.3.3 设计多级数据分组报表.....	(319)
12.4 设计多栏报表.....	(320)
12.5 报表输出.....	(321)
12.6 标签设计.....	(322)
习题 12	(322)
本章实验	(323)
第 13 章 数据库应用系统开发实例.....	(324)

13.1 数据库应用系统设计	(324)
13.1.1 设计目标	(324)
13.1.2 设计思想	(324)
13.1.3 系统功能分析	(324)
13.1.4 系统功能模块设计	(325)
13.1.5 系统表单设计	(325)
13.2 数据库设计	(326)
13.2.1 数据库需求分析	(326)
13.2.2 数据库的逻辑设计	(326)
13.2.3 数据库的物理设计	(327)
13.3 数据库的实现	(327)
13.3.1 创建系统目录结构	(328)
13.3.2 创建数据库及表结构	(328)
13.4 各功能模块的实现	(329)
13.4.1 设计菜单	(329)
13.4.2 编写主程序	(330)
13.4.3 设计启动画面	(331)
13.4.4 设计系统登录界面	(332)
13.4.5 管理好友分组的实现	(334)
13.4.6 文字信息管理的实现	(335)
13.4.7 图文信息共览的实现	(336)
13.4.8 图像信息管理的实现	(340)
13.4.9 图像信息浏览的实现	(343)
13.4.10 修改密码的实现	(347)
13.5 系统的编译和发布	(348)
13.5.1 设置主文件	(348)
13.5.2 对应用程序进行连编	(348)
13.5.3 发布应用程序	(350)
13.6 最终运行结果的查看	(351)
13.7 小结	(353)
习题参考答案	(354)
习题 1 参考答案	(354)
习题 2 参考答案	(354)
习题 3 参考答案	(354)
习题 4 参考答案	(355)
习题 5 参考答案	(355)
习题 6 参考答案	(355)
习题 7 参考答案	(356)
习题 8 参考答案	(356)
习题 9 参考答案	(356)
习题 10 参考答案	(357)
习题 11 参考答案	(357)
习题 12 参考答案	(357)
参考文献	(358)

第 1 章 概 述

本章要点:

- ☞ 数据库系统基本知识
 - ☞ 数据库的数据模型
 - ☞ Visual FoxPro 6.0 系统概述
-

数据库技术的产生与发展源于对数据的组织和管理,但人们对数据的认识,有狭义和广义两种概念。狭义上的数据,是指一些简单的数字和文字;广义上的数据,不仅包括数字和文字,还包括图形、图像、声音等多种类型和多种表现形式的数据。数据可定义为存储在某种媒体上,且能够被识别的描述事物特征的物理符号或符号记录。描述事物特征的符号可以是数字、文字、图形图像、语音、声音等数据的多种表现形式,它们都可以经过数字化之后存入计算机。

数据库技术是数据管理的技术,它随着数据管理任务的需要而产生于 20 世纪 60 年代中期。这是一门综合性技术,涉及操作系统、数据结构、算法设计和程序设计等知识。

1.1 数据库系统概述

数据库技术的基本思想是对数据实行集中、统一、独立的管理,用户可以最大限度地共享数据资源。本节首先介绍一些与数据库相关的知识,为深入学习打下必备的基础。

1.1.1 数据库系统基础知识

1. 数据管理的基本概念和发展过程

(1) 数据

数据是一种物理符号序列,用来记录事物的情况,用型和值来表征。不同数据类型,记录的事物性质也不同,如数值型数据 1, 2, 3..., 可用来记录事物的多少。

(2) 信息

信息是经过加工的有用数据。这种数据有时能产生决策性的影响。

信息都是数据,但只有经过提炼和抽象之后具有使用价值的数据才能成为信息。加工所得的信息仍以数据形式表现,此时的数据是信息的载体,是人们认识信息的一种媒体。

(3) 数据处理

数据处理是指将基本元数据转换成有用信息的过程,是对数据进行收集、存储、加工和传播的一系列活动的总和。这里的“元数据”就是指描述事物的最基本的属性的数据。从数据处理的角度上看,信息是一种被加工成特定形式的数据,而这种形式的数据对于数

据的接收者来说是有意义的。人们通过处理数据可以获得信息，通过分析和筛选可以产生决策。

数据处理是指对各种类型的数据进行收集、存储、分类、计算、加工、检索及传输的过程，其目的是得到信息。数据处理也称为信息处理或信息技术等。

数据处理的核心是数据管理。数据管理指的是对数据的分类、组织、编码、储存、检索和维护等。计算机数据管理随着计算机硬件、软件技术和计算机应用范围的发展而不断发展，多年来经历了以下由低级到高级的发展过程。

(1) 人工管理

20 世纪 50 年代中期以前采用人工管理数据，其特点是：数据与程序不具有独立性，一组数据对应一组程序；数据不能长期保存；存在着大量的数据冗余。

(2) 文件系统

20 世纪 50 年代后期到 60 年代中后期采用文件系统，其特点是：数据和程序有了一定的独立性；数据文件可以长期保存在外存储器上被多次存取；数据和程序相互依赖，同一数据项可能重复出现在多个文件中，导致数据冗余度大，容易造成数据的不一致性。

(3) 数据库系统

从 20 世纪 60 年代后期开始采用数据库系统，其特点是：能够有效地管理和存取大量的数据资源，提高了数据的共享性，减少了数据的冗余度，提供了数据和程序的独立性。

(4) 分布式数据库系统

分布式数据库系统是数据库技术和计算机网络技术紧密结合的产物，产生于 20 世纪 70 年代后期。

(5) 面向对象的数据库系统

面向对象的数据库系统是面向对象程序设计和数据库技术相结合的产物。面向对象数据库是面向对象方法在数据库领域中的实现和应用，它既是一个面向对象的系统，又是一个数据库系统。

数据库系统的主要特点如下。

- ① 实现数据共享，减少数据冗余。
- ② 采用特定的数据模型。
- ③ 具有较高的数据独立性。
- ④ 有统一的数据控制功能。

数据库系统主要解决了 3 个问题，一是有效地组织数据，对数据进行合理的设计，以便计算机存取；二是将数据方便地输入到计算机中；三是根据用户的要求将数据从计算机中抽取出来（这是人们处理数据的最终目的）。

2. 数据库系统的组成

数据库系统实际上是一个应用系统，它是在计算机硬、软件系统支持下，由用户、数据库管理系统、存储在存储设备上的数据和数据库应用程序构成的数据处理系统。

(1) 数据

这里是指数据库系统中存储在存储设备上的数据，是数据库系统操作的对象。这种数据具有集中性和共享性。

(2) 数据库管理系统

数据库管理系统是指位于用户与操作系统之间、负责数据库存取、维护 and 管理的软件系统。这是数据库系统的核心，其功能强弱是衡量数据库系统性能优劣的主要方面；一般由计算机软件公司提供。

(3) 应用程序

应用程序是指为适合用户操作、满足用户需求而编写的数据库应用程序。

(4) 用户

用户是指使用数据库的人员，主要有 3 类：终端用户、应用程序员和数据库管理员。

① 终端用户是指通过数据库系统所提供的命令语言、表格语言以及菜单等交互式对话手段来使用数据库数据的用户。

② 应用程序员是指为终端用户编写应用程序的软件人员，设计的应用程序的主要用途是使用和维护数据库。

③ 数据库管理员 (DataBase Administrator, DBA) 是指全面负责数据库系统正常运转的高级人员，负责对数据库系统本身的深入研究。

3. 数据库系统的特点

数据库系统是计算机数据处理技术的重大进步，涉及的内容如下。

① 实现数据共享。允许多个用户同时存取数据而互不影响。

② 实现数据独立。应用程序不随数据存储结构的改变而变动。

③ 减少了数据冗余度。逻辑数据文件和物理数据文件存在着“多对一”的重叠关系，有效地节省了存储资源。

④ 避免了数据不一致性。数据只有一个物理备份，故对数据的访问不会出现不同。

⑤ 加强了对数据的保护。数据库加入了安全保密机制，可防止对数据的非法存取。进行集中控制，有利于控制数据的完整性。采取了并发访问控制，保证了数据的正确性。另外，还实现了对数据库破坏后的恢复。

4. 数据库应用系统 (DataBase Application Systems, DBAS)

数据库应用系统是指开发人员利用数据库系统资源开发出来的、面向某实际应用的软件系统，并且可分为如下两类。

① 管理信息系统。这是面向机构内部业务和管理的数据库应用系统，如教学管理系统、财务管理系统等。

② 开放式信息服务系统。这是面向外部、提供动态信息查询功能，以满足不同信息需求的数据库应用系统。例如，大型综合科技情报系统、经济信息系统和专业的证券实时行情、商品信息系统等。

一个数据库应用系统通常由数据库和应用程序两部分组成，它们是在数据库管理系统支持下设计和开发出来的。

5. 数据库设计阶段

数据库设计阶段包括：需求分析、概念设计、逻辑设计、物理设计。数据库设计的每个阶段都有各自的任务。

① 需求分析阶段：这是数据库设计的第一个阶段，主要任务是收集和分析数据，这一阶段收集到的基础数据 and 数据流图是下一步设计概念结构的基础。

② 概念设计阶段：分析数据间内在语义关联，在此基础上建立一个数据的抽象模型，即形成 E-R 图。

③ 逻辑设计阶段：将 E-R 图转换成指定关系型数据库管理系统 (RDBMS) 中的关系模型。

④ 物理设计阶段：对数据库内部物理结构作调整并选择合理的存储结构和存取方法，以提高数据库访问速度及有效利用存储空间。

1.1.2 数据库的数据模型

一般来说，设计数据模型应遵循 3 个原则：一是能较真实地模拟现实世界中的事物；二是容易被人们所理解或接受；三是便于在计算机上实现。

数据模型是数据库应用系统的核心和基础，是 DBMS 用来表示实体及实体间联系的方法，它应该能够反映出数据之间存在的整体逻辑关系。任何一个 DBMS 都是基于某种数据模型的。

从创建数据库技术以来，逻辑数据模型有 3 种类型，即层次模型、网状模型、关系模型。数据模型的另一种表现形式，概念数据类型——E-R 数据模型。本书所讲的 Visual FoxPro 虽然在面向对象技术方面有了很大的发展，但就其本质来讲仍属于关系模型范畴。通常，我们把基于特定的数据模型开发出来的数据库系统相应地称为层次型数据库系统、网状型数据库系统、关系型数据库系统。随着数据库技术的产生与发展，关系模型已经成为目前最为流行且影响最深远的数据库模型。大型的 Oracle, SQL-Server 或 Sybase 等几种 DBMS 系统都采用关系数据模型。

下面简要介绍这些数据模型及其特点。

(1) 层次模型 (hierarchical model)

层次模型是数据库中出现最早的数据模型，它用树型结构来表示各类实体以及实体之间的联系，如图 1-1 所示。现实生活中的许多实体之间的联系都很自然地体现出了这种层次关系，如行政管理机构。层次模型的主要特点如下：

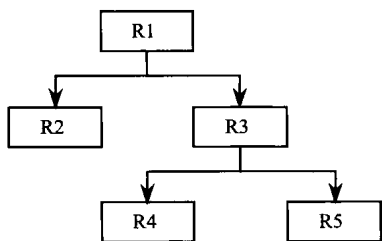


图 1-1 层次模型示例

① 有且只有一个根结点。

② 除根结点以外，其他结点有且只有一个父结点。

③ 在这种树型结构中，每一个结点都表示一个实体类型或称记录类型。结点间的连线反映的是不同实体之间的“一对多”联系。

④ 在这种数据模型中，不能直接体现实体间“多对多”的联系。

层次模型中的记录只能组织成树的集合，对于非层次数据，层次 DBMS 的效率就很低，使用也不方便。随着 DBMS 技术的不断发展，现在已很少使用层次模型来研制 DBMS。

(2) 网状模型 (network model)

网状模型是用网状结构来表示实体及其之间联系的模型,如图 1-2 所示。网状模型的主要特点如下:

① 可以很灵活地表现出实体间的“多对多”联系,而这种灵活性是建立在复杂的结构基础之上的。

② 数据的更新实现起来比较复杂。

③ 在数据的组织实现方面常用链接方法。

对于网状模型,由于这种模型结构复杂,且规律性差,在计算机上实现比较困难。与层次数据模型一样,网状数据模型现在也已很少在 DBMS 的设计中使用。

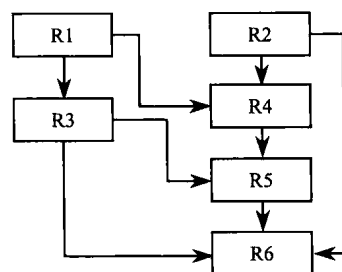


图 1-2 网状模型示例

(3) 关系模型 (relation model)

关系模型是目前最重要的一种数据模型。E.F.Cold 于 1970 年首次提出了数据库系统的关系模型,为关系模型数据库技术奠定了理论基础。由于他的杰出贡献,于 1981 年获得了有计算机界“诺贝尔”奖之称的 ACM 图灵奖。

关系模型不同于层次模型、网状模型,它建立在严格的数学基础之上,它的运算对象和结果都是集合(二维表)。它是用二维表结构来表示实体及实体间联系的数据模型。正是由于关系数据模型具有描述的一致性、模型概念的简单化、完备的数学理论基础、说明性的查询语言和使用方便等优点,从而使它得到了广泛的应用。

表 1-1 是一张记录了学生基本数据的二维表,表的每一行都记录了一名学生的相关信息,在数据库中称为记录。表的每一列称为项,描述了学生属性的同类型数据,如:姓名、性别、学号等。

表 1-1 学生信息

姓名	性别	班级	学号	籍贯	出生年月	入学成绩	专业	简历	相片
李晓红	女	200201101	20020110101	四川	1984-12-11	530	应用数学	memo	gen
王刚	男	200201101	20020110102	四川	1984-10-23	548	应用数学	memo	gen
昭辉	女	200201101	20020110103	四川	1985-10-11	529	应用数学	memo	gen
李琴	女	200201101	20020110104	江苏	1984-12-11	550	应用数学	memo	gen
方芳	女	200201102	20020110205	湖南	1985-6-15	524	计算机	memo	gen
谭新	女	200201102	20020110206	北京	1985-6-23	570	计算机	memo	gen
刘江	男	200201102	20020110207	河南	1985-2-8	539	计算机	memo	gen
王长江	男	200201102	20020110208	山西	1984-3-9	528	计算机	memo	gen
张强	男	200202101	20020210109	江苏	1983-12-1	549	应用化学	memo	gen
江海	男	200202101	20020210110	江苏	1984-10-23	546	应用化学	memo	gen
明天	男	200202101	20020210111	河南	1984-7-8	552	应用化学	memo	gen
希望	男	200202101	20020210112	北京	1983-7-8	560	应用化学	memo	gen

(4) E-R 数据模型

上面介绍了 3 种数据模型,它们都属于逻辑数据模型。这里介绍的 E-R 数据模型(Entity-Relationship data model)即实体联系数据模型,是属于概念数据模型。对于现实世界中的数据,在 E-R 数据模型中就抽象为两个重要的概念:实体和联系。

① 实体 (Entity) 是客观存在并可相互区分的事物, 是对能被人们识别的独立存在的对象的描述, 是对现实世界中事、物、概念等的抽象。如: 一个商店、一本书、老师、学生、图书、火车、商品、公务员、各种机器设备和各种建筑物等都是实体。在一个单位 (或研究的组织事务) 中, 具有共性的一类实体称之为实体集, 可由设计者规定。例如在建立一个学校的学生档案数据库时, 所有该校的学生就构成一个实体集。实体集均要用实体名来对它进行标识, 实体名应能够反映实体集的构成。

实体一般均具有若干特征, 这些特征称为该实体的属性。例如学生这个实体就可以具有下列属性: 学号、姓名、性别、年龄、专业、班级和民族等。每个属性均可以规定一定的取值范围, 在 E-R 模型中称之为值集。例如对某个具体的学生来说, 就可以用 “200371015、张三、男、18、计算机科学与技术、2003710、汉族” 来描述。当实体的某个属性不适用或属性值未知时, 也可用 “NULL” 表示。实体属性与应用系统的要求有关, 因而对于同样的实体集, 其属性的构成可能并不相同。

由于在现实世界中, 抽象成实体的数目要比数据元素的数目少得多, 因此以实体作为研究对象, 就大大地减少了对现实世界分析中所涉及的对象, 简化了分析过程。

② 联系是独立的实体相互之间的关系。现实世界的实体与实体之间通常都有关联, 如: 书店和书的关系是销售关系, 老师与学生这两个实体就有老师给学生上课, 学生选修老师讲授的课程; 工人按操作规程操作机器, 人们为居住而选购建筑物 (住宅楼)。不但不同的实体相互之间有关联, 就是同一实体集中的多个实体之间也存在相互的关联。例如对人这个实体集来说, 人与人之间存在着夫妻关系、父子关系, 等等。为了反映实体与实体之间的联系, 在 E-R 数据模型中, 就把这种实体与实体之间的关联抽象为联系。因此, 联系也是 E-R 模型中的一种描述对象的数据。

能够惟一标识实体的属性或属性组, 称之为该实体的实体键 (Entity Key)。如果一个实体 (集) 有多个实体键存在, 则可选其中最常用的一个作为实体主键 (Entity Primary Key)。

现实世界中可能存在许多实体 (用 E 表示实体或实体集), 例如 $E_1, E_2, \dots, E_n$, 则这些实体之间的联系, 可用一个元组 $(E_1, E_2, \dots, E_n)$ 来表示, 当 $n=2$ 时, (E_1, E_2) 就称之为一个二元联系, 当 $n>2$ 时, 就称为多元联系。

在 E-R 模型中, 如果实体之间的联系是对一个实体集而言的, 这种联系就称为一元联系。

在二元联系中, 实体 E_1 与 E_2 之间的联系 (E_1, E_2) 可归结为如下 3 种类型: $1:1, 1:n, m:n$ 。

① $1:1$ (一对一) 联系。对于任意两个实体 E_1 和 E_2 , 若 E_1 中的任意一个实体最多和 E_2 中的一个实体相联系, E_2 中的每个实体也最多和 E_1 中的一个实体相联系, 则称 E_1 和 E_2 是 “一对一” 联系, 简记为 $1:1$ 联系。例如, 某个班的学生组成一个实体集, 该班指定教室的课桌组成另一个实体集, 假定该班的座位是按固定方式分配的, 则学生与座位之间就构成了一对一联系: 一个学生对应一个座位, 一个座位也对应一个学生。

② $1:n$ (一对多) 联系。对于任意两个实体 E_1 和 E_2 , 若 E_1 中的任意一个实体可以和 E_2 中的任意数目 (可以是 0, 2, 3, $\dots, n$) 的实体相联系, 则称 E_1 到 E_2 的一对多联系, 简记为 $1:n$ 联系。例如, 在某个银行的信贷业务中, 客户与贷款是两个不同的实体, 银行的一笔贷款只能属于一个客户, 但一个客户可以获得多笔贷款, 因此客户与贷款的联系就是一对多的联系。与一对多联系相对应地还有多对一联系 ($n:1$), 所谓多对一联系是指实体 E_1 中的任意一个实体最多同 E_2 中的一个实体相联系, 但 E_2 中的任意一个实体可以同 E_1 中的

任意数目的实体相联系，则称 E1 到 E2 是多对一的联系。E1 到 E2 的多对一联系也可以看成是 E2 到 E1 的一对多联系。

③ $m:n$ (多对多) 联系。对于任意两个实体 E1 和 E2，若 E1 中的每个实体，可以同 E2 中的任意数目 (0, 2, 3, ..., n) 的实体相联系，E2 中的每个实体也和 E1 中的任意数目 (0, 2, 3, ..., m) 的实体相联系，则称 E1 和 E2 是多对多联系，简记为 $m:n$ 联系。

多对多联系在现实生活中是很多的，例如师生关系、学生与图书、供应商与生产厂家、商场与货物等都属于多对多联系。

1.1.3 关系模型

1. 关系模型中的几个术语

(1) 字段 (field)

信息世界中的“属性”，就是数据世界中的“数据项”。从数据库的角度讲，数据项就是字段；从表格的角度讲，数据项称为列。例如学生的学号、姓名、性别和专业就是字段名。

字段和属性一样，也用型和值表示。由此可见，字段、属性、数据项、列这些术语，所描述的对象是相同的，只是从不同角度对对象进行描述而已。

(2) 记录 (record)

字段的有序集合称为记录。在关系模型中，记录称为元组；在表中，记录称为行，在概念模型中称为实体。换句话说，实体、记录、元组和行分别是不同的角度描述同一对象的术语。记录也由型和值来描述：记录型是字段型的集合，记录值是字段值的集合。

(3) 表 (relation)

记录的集合称为表。记录的型和值构成了关系数据库的基本单位，即表。表也分为型和值，表的型也称关系模式，或称表结构，由一系列字段型组成。

(4) 关键字 (keyword)

关键字能够惟一确定记录的字段或字段的集合。有了关键字就可很方便地使用指定的记录。

(5) 关系数据库

这是由若干表组成的集合。也就是说，关系数据库中至少有一个表。在实际应用中，关系数据库通常是由若干有着一定关系的表组成的。在关系数据库系统中，关系是相对稳定的。因数据库中的数据需要不断更新，故关系数据是不断变化的。

2. 关系模型的优点

关系模型是建立在严格的数学概念基础上的，每个关系都用一张表格来描述，字段、记录描述得很清楚。更重要的是可用关系的性质来衡量关系。

- ① 关系规范化，即每个数据项（即字段）都是一个不可分的数据单元。
- ② 概念简单，数据结构简单、清晰，用户易懂易用。
- ③ 存取路径对用户透明，数据的独立性高，安全保密性强。