

21世纪

计算机课程系列教材

主编 郭云飞

Visual FoxPro6.0 程序设计基础教程

Visual FoxPro6.0 Chengxu Sheji Jichu Jiaocheng



北京邮电大学出版社
www.buptpress.com

21 世纪计算机课程系列教材

Visual FoxPro 6.0 程序设计基础教程

(修订版)

主 编 郭云飞



北京邮电大学出版社
www.buptpress.com

内容简介

本书在简要引入关系型数据库技术基本知识的基础上,主要介绍了 Visual FoxPro 系统的基本应用环境,并通过项目管理器的使用、数据库的建立和表的操作、数据库操作、查询程序的设计和常用命令的基本结构、报表和标签的使用等,全面说明了 Visual FoxPro 面向对象程序设计的基本概念。本书配有大量精选的插图,使知识演进手段丰富而不杂乱;在内容组织上理论和应用兼顾,因而适应性较强。

本书可作为高等学校数据库及其应用课程的教材,也可供参加全国计算机等级考试的考生在准备二级 Visual FoxPro 程序设计考试时使用。

图书在版编目(CIP)数据

Visual FoxPro 6.0 程序设计基础教程/郭云飞主编. —北京:北京邮电大学出版社,2007

ISBN 978-7-5635-0596-8

I. V... II. 郭... III. 关系数据库—数据库管理系统, Visual FoxPro—程序设计—教材

IV. TP311.138

中国版本图书馆 CIP 数据核字(2007)第 011888 号

书 名 Visual Fox Pro 6.0 程序设计基础教程
主 编 郭云飞
责任编辑 陈露晓
出版发行 北京邮电大学出版社
社 址 北京市海淀区西土城路 10 号(100876)
电话传真 010-62282185(发行部) 010-62283578(传真)
E-mail ctrd@buptpress.com
经 销 各地新华书店
印 刷 北京忠信诚胶印厂
开 本 787 mm×1 092 mm 1/16
印 张 16.25
字 数 392 千字
版 次 2007 年 1 月第 2 版 2007 年 1 月第 1 次印刷

ISBN 978-7-5635-0596-8/TP·93

定价 28.00 元

如有质量问题请与北京邮电大学出版社高校教材研发中心联系

版权所有 侵权必究

前言

当前,计算机正在以令人难以置信的速度步入普通人的学习、工作和生活。在初见端倪的信息化社会里,不能有效使用计算机的单位和个人,无疑将在其各种活动中面临重重障碍。

随着社会需求的驱动和计算机技术的发展普及,各行各业的管理机构需要用计算机来处理大量的信息。选择一个优秀的数据库管理系统作为开发平台,将给日后的信息处理带来极大的方便。其中,基于微型计算机的数据库系统是应用最广、实用性最强的一种计算机技术。而实践表明:Visual FoxPro 数据库是一个广泛使用、性能/价格比较高、工具完整而又丰富、界面友好、处理速度快的关系型数据库。Visual FoxPro 主要应用于 Windows 环境,利用它可设计出体现 Windows 风格的丰富多彩的图形化用户界面,使用户的操作和使用十分便利。Visual FoxPro 是为数据库应用程序开发而设计的功能强大、面向对象的编程工具。它能够管理大量复杂的数据,同时又具有较强的网络功能,甚至还能够实现数据的远程访问和存储加工。

为了满足上述实际需要,作者编写了《Visual FoxPro 6.0 程序设计基础教程》一书。在编写过程中,作者着重突出了以下特点:

面向实际应用:本书在保证知识的系统性的同时,突出了应用性。在整体结构和素材的选用上,特别注意实际应用,以满足读者实际学习和工作的需要。全书有丰富的示例和习题,并含有适量的技能训练,可供学习思考和上机操作时使用和参考。

合理组织内容:本书收集和制作了大量精选的图文数表,以图文并茂的方式、具体逼真的素材,力图使抽象的内容具体化、实用化,并努力拓宽读者的视野,激发学习的兴趣,加强学习的动力,以求达到良好的学习效果。

注重能力培养:读者可通过作者精心组织的屏幕插图来理解和掌握抽象的概念及一些复杂的过程,从而可较好地实现直观感觉与抽象思维之间的过渡,从而在培养操作能力的同时提高思维能力、在培养应用能力的同时提高自学能力。

循序渐进:本书重点介绍 Visual FoxPro 程序设计方法,但为此准备了必要的数据库理论知识,然后从项目管理器入手介绍 Visual FoxPro 数据库的基本操作,Visual FoxPro 结构化程序设计的基本方法,再引入 Visual FoxPro 面向对象程序设计的概念,以及表单、菜单、报表、常用控件和工具的设计方法,最后介绍具有代表性的、读者便于理解的设计实例,体现了学习数据库及其应用程序设计所必须的由浅入深的过程。

本书修订版对部分章节的顺序做了调整,重写了第 1、3、4、5 章,修改了其它章节的部分内容,参加重写与修改工作的同志有郭云飞(第 1、4、5 章)、曾映兰(第 2 章)、曹江莲(第 3 章)、朱红(第 6 章)和周红球(第 7、8、9 章)。

我们对参加了第一版编写工作的韩忠愿教授、兰艳玲副教授、李宏高级实验师和邱桔博士表示感谢,并对大力支持我们编写工作、提出修改意见的教师与学生表示感谢。

尽管在编写过程中作者作出了许多努力,但由于水平有限,书中难免还有一些错误和疏漏,敬请读者批评指正。

编者

2006 年 12 月

目 录

第1章 数据库系统基础

1.1 数据库系统基础知识	(1)
1.1.1 数据管理的发展	(1)
1.1.2 数据库系统的特点和组成	(3)
1.1.3 数据模型	(4)
1.2 关系数据库	(6)
1.2.1 关系数据模型	(6)
1.2.2 关系运算	(8)
1.2.3 常用关系数据库管理系统及开发工具	(11)

第2章 数据与数据处理

2.1 数据	(13)
2.1.1 常量	(13)
2.1.2 变量	(14)
2.1.3 数组	(17)
2.1.4 系统变量	(18)
2.2 表达式	(18)
2.2.1 数值表达式	(18)
2.2.2 字符表达式	(20)
2.2.3 日期与时间表达式	(20)
2.2.4 逻辑表达式	(21)
2.2.5 表达式生成器	(24)
2.3 常用函数	(25)
2.3.1 数值函数	(26)
2.3.2 字符函数	(28)
2.3.3 日期与时间函数	(31)
2.3.4 数据类型转换函数	(33)
2.3.5 测试函数	(35)

第3章 Visual FoxPro 数据库的基本操作

3.1 Visual FoxPro 项目管理器	(37)
3.1.1 项目管理器及功能	(37)
3.1.2 项目管理器的基本操作	(37)
3.2 数据库的创建与使用	(40)
3.2.1 Visual FoxPro 的数据库与表	(40)
3.2.2 数据库的创建	(41)
3.2.3 数据库的操作	(42)
3.3 数据表的操作	(45)
3.3.1 创建数据表	(45)
3.3.2 表结构的操作	(49)
3.3.3 表文件的打开与关闭	(51)
3.3.4 表文件的数据的输入与添加	(52)
3.3.5 记录的显示与定位	(54)

3.3.6 记录的删除与恢复	(59)
3.3.7 表数据的修改	(61)
3.4 排序与索引	(65)
3.4.1 索引	(65)
3.4.2 排序	(70)
3.5 多表操作	(71)
3.5.1 选择工作区	(71)
3.5.2 使用非当前工作区中的表	(72)
3.5.3 表间关联	(72)
3.6 数据完整性	(77)
3.6.1 设置字段属性	(77)
3.6.2 设置记录规则	(78)
3.6.2 参照完整性(RI)	(79)
第4章 结构化查询语言(SQL)	(82)
4.1 SQL 概述	(82)
4.2 数据定义	(83)
4.3 数据查询	(87)
4.3.1 查询的概念	(87)
4.3.2 SELECT 的基本结构	(88)
4.3.3 简单查询	(88)
4.3.4 联接查询	(91)
4.3.5 嵌套查询	(94)
4.3.6 聚集与分组	(97)
4.3.7 集合的并运算	(99)
4.3.8 其他	(99)
4.4 查询设计器	(101)
4.4.1 Visual FoxPro 查询设计器简介	(101)
4.4.2 查询的定义和功能	(103)
4.4.3 查询设计实例	(107)
4.4.4 查询文件	(108)
4.5 数据操纵	(109)
4.5.1 插入操作	(109)
4.5.2 删除操作	(110)
4.5.3 更新操作	(110)
4.6 视图	(111)
4.6.1 数据库视图的概念	(111)
4.6.2 视图的创建	(112)
4.6.3 Visual FoxPro 视图设计器	(114)
第5章 程序设计基础	(116)
5.1 程序	(116)
5.1.1 命令文件	(116)
5.1.2 简单输入输出命令	(118)
5.1.3 其他功能性命令	(119)
5.2 程序的基本结构	(121)
5.2.1 顺序结构	(121)
5.2.2 选择结构	(122)
5.2.3 循环结构	(127)
5.2.4 程序范例	(132)
5.3 多模块程序	(134)

5.3.1 程序调用	(134)
5.3.2 过程	(136)
5.3.3 自定义函数	(141)
5.3.4 变量的作用域	(142)
第 6 章 表单设计	
6.1 面向对象程序设计	(145)
6.1.1 基本概念	(145)
6.1.2 Visual FoxPro 中的对象	(146)
6.1.3 常用的属性	(147)
6.1.4 事件	(148)
6.1.5 表单常用方法	(148)
6.1.6 对象引用	(148)
6.2 创建表单	(149)
6.2.1 使用表单向导创建表单	(150)
6.2.2 使用表单设计器创建表单	(150)
6.2.3 设置数据环境	(152)
6.3 表单的基本操作	(153)
6.3.1 表单设计用工具栏	(153)
6.3.2 调整 TAB 键次序的命令	(157)
6.3.3 表单的管理	(157)
6.4 常用表单控件	(159)
6.4.1 命令按钮(CommandButton)	(159)
6.4.2 命令按钮组(CommandGroup)	(161)
6.4.3 标签(Label)	(162)
6.4.4 图象(Image)	(163)
6.4.5 文本框(TextBox)	(163)
6.4.6 编辑框(EditBox)	(165)
6.4.7 复选框(CheckBox)	(167)
6.4.8 选项组(Option Group)	(167)
6.4.9 微调控件(Spinner)	(169)
6.4.10 列表框(ListBox)	(169)
6.4.11 组合框(ComboBox)	(173)
6.4.12 计时器(Timer)	(176)
6.4.13 表格(Grid)	(176)
6.4.14 页框(Pageframe)	(180)
6.4.15 ActiveX 控件和 ActiveX 绑定控件	(182)
第 7 章 菜单设计	
7.1 菜单的组成	(184)
7.1.1 菜单结构	(184)
7.1.2 菜单系统的规划	(184)
7.1.3 Visual FoxPro 系统菜单	(185)
7.2 下拉式菜单设计	(186)
7.2.1 菜单生成的基本步骤	(186)
7.2.2 菜单设计器窗口	(187)
7.2.3 快速菜单	(190)
7.2.4 “显示”菜单	(190)
7.3 弹出式菜单设计	(194)
7.3.1 用菜单设计器设计快捷菜单	(194)
7.3.2 菜单编程命令	(195)

7.3.3 在控件中附加快捷菜单	(199)
第8章 报表设计	
8.1 报表向导	(200)
8.2 报表设计器	(204)
8.2.1 报表设计器简介	(204)
8.2.2 设置数据环境	(206)
8.2.3 设计报表带区	(207)
8.2.4 调整报表带区的大小	(208)
8.2.5 预览报表	(210)
8.2.6 打印报表	(211)
8.3 报表控件	(213)
8.3.1 标签、线条、矩形和圆角矩形	(213)
8.3.3 域控件	(214)
8.3.3 报表变量	(216)
8.4 快速报表	(217)
第9章 开发应用程序	
9.1 应用程序开发基本步骤	(218)
9.1.1 需求分析	(218)
9.1.2 数据库设计	(218)
9.2 系统菜单设计	(219)
9.2.1 创建菜单系统的过程	(220)
9.2.2 规划菜单准则	(220)
9.2.3 实现菜单系统	(220)
9.3 设计表单	(221)
9.3.1 数据环境	(221)
9.3.2 表单类型	(222)
9.3.3 创建表单集	(222)
9.3.4 设计表单	(223)
9.4 报表设计	(225)
9.4.1 规划报表	(225)
9.4.2 数据准备	(225)
9.4.3 外观设计	(225)
9.4.4 确定报表变量	(225)
9.4.5 分组和统计	(226)
9.4.6 输出结果类型	(226)
9.5 编译应用程序	(226)
9.5.1 主程序设计	(226)
9.5.2 向项目中添加文件	(228)
9.5.3 连编应用程序	(228)
9.6 打包应用程序	(231)
9.7 发布应用程序	(231)
附录1 Visual FoxPro 6.0 主要命令	(233)
附录2 Visual FoxPro 6.0 主要函数	(242)

第1章 数据库系统基础

1.1 数据库系统基础知识

数据库技术始于20世纪60年代,经历了最初的基于文件的初级系统、20世纪60~70年代流行的层次系统和网状系统,现在广泛使用的是关系数据库系统。随着计算机技术的蓬勃发展,计算机应用已经渗入到人们日常生产生活的各个领域,数据库应用也从简单的事务管理扩展到工程管理、网络支持乃至知识发现,如用于工程设计的工程数据库、用于Internet的Web数据库、用于决策支持的数据仓库、用于多媒体技术的多媒体数据库等。随着时代的进步和发展,数据库的应用领域会越来越广泛,尤其在当今信息社会,计算机已成为人们日常工作中处理数据的得力助手和工具,数据处理与科学计算、过程控制、辅助设计一起,并称为计算机四大应用。本书讲解的Visual FoxPro作为一种目前广泛应用的数据库管理系统,在上述数据处理中扮演着十分重要的角色。本章将从一般数据库知识入手,介绍与数据处理的核心技术—数据库技术相关的概念、知识和技能,为进一步学习Visual FoxPro 6.0及其应用开发技术奠定基础。

1.1.1 数据管理的发展

从某种意义上说,整个人类文明史就是人类进行数据处理的历史。所谓数据处理是指对数据进行收集、存储、加工和传播的一系列活动的总和。而数据管理则是指对数据进行分类、组织、编码、存储、检索和维护,它是数据处理的核心。

自从计算机问世以来,人们便借助计算机这种先进的计算工具进行数据管理。计算机数据管理的发展经历了人工管理、文件系统管理到数据库系统管理三个阶段。

1. 人工管理

数据的人工管理与计算机的产生和应用同步开始,由于早期没有软件系统对数据进行组织,也没有强有力的计算机硬件支持,数据的组织管理全部由人工承担,计算机只能对数据进行计算。其过程是,先将程序和人工组织的数据输入计算机,计算机运行结束后,将结果输出,再由人工解读、分析和保存。

2. 文件系统管理

到了20世纪50年代后期,计算机硬件与软件都有了很大发展,尤其是出现了磁盘、磁鼓等直接存取设备,操作系统中有了专门的数据管理软件,即文件系统。

用文件系统管理数据具有如下特点:

(1) 数据长期保存

由于计算机大量用于数据处理,将数据长期保存在外存中,便于对数据反复进行查询和更新(插入、删除和修改)。

(2) 文件系统管理数据

由专门的软件即文件系统进行数据管理。文件系统把数据组织成相互独立的数据文件,利用“按文件名访问、按记录进行存取”的管理技术,对数据文件进行查询和更新操作。文件系统是数据文件和应用程序的专门接口。

(3) 数据冗余大

数据冗余是指同一数据同时存在多个拷贝这种数据重复现象。在文件系统中,一个文件基本上对应一个应用程序,即文件是面向应用的。当不同的应用程序需要部分相同的数据时,由于不能共享相同的数据,往往需要建立各自的数据文件,从而产生数据冗余。数据冗余不仅浪费存储资源,而且由于各自管理容易造成数据的不一致性,给数据的维护带来困难。数据的不一致性是指同一数据的不同拷贝其值不一样。例如在教务部门的学生成绩数据文件有学生的基本信息,在学工部门的学生管理数据文件也有学生的基本信息,由于各自由不同的应用程序管理,容易造成学生的基本信息在两个部门不一致。

(4) 数据独立性差

由于文件是面向应用的,一个数据文件是为某一特定的应用服务的,文件的逻辑结构是在应用程序中定义的,因此,一旦数据的逻辑结构改变,必须修改应用程序以适应新的结构。反之,应用程序的改变(如用不同的计算机语言改写该应用程序),也可能引起文件的逻辑结构的改变。因此说,程序与数据之间独立性差。

(5) 数据联系弱

这里,数据联系是指不同文件中的数据之间的联系。在文件系统中,不同文件中的数据之间是独立的,缺乏相互关联的方法,因此,数据联系弱。实际上,同一个系统中的不同数据文件之间的数据一般存在着内在的、至关重要的逻辑联系,但由于实现的复杂性,很少在文件管理系统中提供这些数据之间的联系方法。因此,各个数据文件之间是孤立的。

3. 数据库系统管理

上世纪 60 年代后期以来,计算机技术有了迅猛发展,同时,计算机用于数据管理的规模越来越大,应用越来越广泛,数据量急剧增长,数据共享的要求越来越强烈,文件系统作为数据管理手段已经不能满足应用的要求。于是,一种新的、功能强大的数据管理技术——数据库技术便应运而生,这种专门用于数据管理的软件系统称为数据库系统。与文件系统相比,数据库系统具有以下明显的特点:

(1) 数据结构化

数据结构化是数据库系统与文件系统的根本区别。在文件系统中,数据之间的联系弱,即它仅关心记录内部数据项之间的联系,而在不同文件中的记录之间没有联系,也就是在整体上数据是无结构的。而在数据库系统中,数据按照某种数据模型进行组织。数据模型不仅描述了数据本身的特征,而且描述了数据与数据之间的联系,且这种联系通过存取路径(指针)来实现数据的整体结构化。

(2) 数据共享、数据冗余度低

由于数据库系统从整体角度来描述和组织数据,数据不再面向特定的应用程序,而是面向

整个系统,因此数据可以被多个应用程序、多个用户共享,从而大大减少了数据冗余、节约了存储空间,提高了数据的利用率。数据共享还能够避免数据之间的不相容性与不一致性。

(3) 数据独立性高

由于数据库中数据的物理结构与逻辑结构的描述都保存在数据库中,而不是应用程序中。应用程序通过保存在数据库中数据结构描述来存取数据。因而在数据库系统中,数据与程序相互独立,互不依赖,不因一方的改变而改变另一方,这就大大简化了应用程序的设计与维护代价。

(4) 数据的统一控制与管理

在数据库系统中,数据由数据库管理系统(DBMS)统一控制与管理。DBMS 至少应提供数据的安全性保护、完整性检查、并发控制和数据库恢复等功能。数据的安全性保护是指保护数据以防止不合法的使用造成的数据泄密与破坏,即数据必须按规定的权限与方式来使用。数据的完整性检查是将数据控制在有效的范围内,或保证数据之间满足一定的关系。并发控制机制控制和协调多个用户对数据库的并发操作。当数据库系统发生故障时,数据库中的数据可能会遭到破坏或丢失,使数据库处于一个错误的状态,数据库恢复机制将数据库从一个错误的状态恢复到一个正确的状态。

数据库系统从 20 世纪 60 年代中期产生到今天仅仅 30 多年的历史,但其发展速度之快、使用范围之广是其他技术望尘莫及的。短短 30 多年间,数据库系统已从第一代的网状数据库、层次数据库,第二代的关系数据库,发展到第三代以面向对象模型为主要特征的面向对象数据库,数据库技术与网络通信技术、人工智能技术、面向对象程序设计技术、并行计算技术等互相渗透、互相结合,成为当前数据库技术发展的主要特征。

1.1.2 数据库系统的特点和组成

在计算机系统中引入数据库后组成的系统就是数据库系统(DataBase System,简称 DBS)。一般地,数据库系统由数据库、数据库管理系统(及其开发工具)、以数据库管理员(DBA)为代表的数据库用户以及应用系统组成。

1. 数据库(DataBase,简称 DB)

可以说数据库就是存放数据的仓库。所谓数据库是指长期存储在计算机内的、有组织的和可共享的关联数据的集合。数据库是数据库系统的核心和管理对象。

2. 数据库管理系统(DataBase Management System,简称 DBMS)

存储在数据库中的大量数据需要专门的管理软件才能正确、安全、高效地使用,数据库管理系统就是这种系统软件,它是位于操作系统与用户之间的一层数据管理软件。DBMS 具体负责对数据库的数据进行统一的管理和控制。用户或应用程序发出的各种操作数据库及其中数据的命令,都要通过 DBMS 来执行。如数据库创建,数据的定义、查询、更新等都要通过 DBMS 进行。

3. 应用系统

这里所说的应用程序系统特指数据库应用程序系统,它是建立在 DBMS 基础上、针对某个具体的对象(应用)而设计、开发的一个面向用户的软件系统,而且具有良好的交互操作性和用户界面。如学生选课管理系统、人事管理系统、财务管理系统等都是数据库应用程序系统。

4. 数据库用户

数据库用户是指开发、管理和使用数据库系统的人员。根据他们的工作内容可分成以下

三类人员。

(1) 最终用户。这类人员一般是不要求精通计算机的各级管理人员,他们一般通过使用应用程序来访问数据库。

(2) 应用程序员,包括系统分析员和设计人员。这类人员是负责设计和编制应用程序的人员。

(3) 数据库管理员(DataBase Administrator,简称 DBA)。这类人员是指全面负责数据库系统的规划、设计、管理和维护,保证系统能够正常使用的人员。

1.1.3 数据模型

1. 数据模型

数据库的应用是面向现实世界的,而数据库系统是面向计算机的,二者之间存在着很大的差别。如何去认识现实世界、让计算机处理源自现实世界的数据库?在数据库技术中,用数据模型这个工具来抽象、表示和处理现实世界中的数据。数据模型就是现实世界的模拟。数据模型的种类有很多,根据模型应用目的,可以划分为两类,分属不同的层次。一类是概念模型,主要用于数据库设计。它独立于计算机系统,完全不涉及信息在计算机系统的表示。另一类是基本数据模型(也称结构数据模型,简称数据模型),主要用于数据库在 DBMS 上的实现。

数据模型是严格定义的概念的集合,一般包含数据结构、数据操作和完整性约束三部分。

(1) 数据结构是所研究的对象类型的集合。在数据库系统中,通常按照数据结构来命名数据模型。例如层次结构、网状结构和关系结构的数据模型分别命名为层次模型、网状模型和关系模型。

(2) 数据操作是指对数据库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更新(包括插入、删除和修改)两大类。

(3) 完整性约束是指对数据的一组完整性规则的集合。这些规则是数据及其联系所具有的制约和依赖规则,用于限定数据库的状态及状态的变化、以保证数据的正确、有效和安全。

2. 实体—联系模型

概念模型的表示方法有很多,其中最常用的是 P. S. Chen 于 1976 年提出的实体—联系方法(Entity-Relationship Approach)。用该方法来描述现实世界的概念模型也称实体—联系模型(E-R 模型)。E-R 模型涉及的主要概念有:

(1) 实体(Entity):实体是客观存在并可以相互区分的事物。实体既可以是实际的,如人、汽车等,也可以是抽象的,如比赛、演出、学习等。在实际应用中,只要抽象出那些与应用问题密切相关的实体。

(2) 属性(Attribute):属性是实体所具有的某一特性,如姓名、性别、年龄、学号等是学生实体的属性。一个实体可以由若干属性来刻画。在实际应用中,只需选择实体的那些与应用问题密切相关的属性。

(3) 码(Key):码是唯一标识实体的属性集。组成码的属性称为主属性。如学号是学生实体的码,因为每一个学生的学号都不同。

(4) 域(Domain):域是属性的取值范围。

(5) 实体集(Entity Set):具有相同属性的实体的集合。如全体学生。

(6) 实体型(Entity Type):对实体集的抽象与刻画。用实体名及其属性名集合来描述。例如,学生(姓名、性别、年龄、学号)。

(7) 联系(Relationship):现实世界中各事物间联系在概念模型中的反映。任意两个实体集之间的联系可以是以下三种类型之一。

① 一对一的联系(记为 1:1)

实体集 A 中的一个实体至多与实体集 B 中的一个实体相联系,反之亦然。例如,一个班只有一位班长,一个班长只能是一个班的班长,则班长集合与班集合之间是一对一的联系。

② 一对多的联系(记为 1:n)

若实体集 A 中的一个实体可以与实体集 B 中的多个实体相联系,反之,实体集 B 中的一个实体至多与实体集 A 中的一个实体相联系,则实体集 A 到实体集 B 之间的联系是一对多的。例如父亲对子女、学校对系、系对班级、班级对学生都是一对多的联系。

③ 多对多的联系(记为 $n:m$)

实体集 A 中的一个实体可以与实体集 B 中的多个实体相联系,反之亦然。如一个学生可以选修多门课程,而一门课程可以同时有学生选修,学生与课程的联系即为多对多的联系。注意,联系也可以有属性。例如,学生与课程的联系可命名为“选修”,“成绩”就是“选修”的属性。

实体与实体间的联系可以是多方面的,在实际应用中,只要抽象出那些对解决应用问题有帮助的联系。

(8) E-R 图

实体—联系模型的描述工具是 E-R 图,用图形的方式来描述实体—联系模型具有直观、清晰等优点。E-R 图的图例符号主要有:

① 矩形。表示实体型,在矩形框里写上实体名。

② 菱形框。表示联系,用无向边分别与有关实体连接起来,并在菱形框里写上联系名,同时在无向边旁标上联系的类型(1:1,1:n 或 $m:n$)。

③ 椭圆框。表示属性,用无向边与所属实体或联系连接起来,并在椭圆框里写上属性名。若是主属性,则在属性名下划一横线。

例如,在按班级管理学生与班级的模型中,涉及的主要实体有“学生”与“班级”,每个实体的属性根据需要选择,所有与管理无关的属性应该忽略,E-R 图如图 1-1 所示。

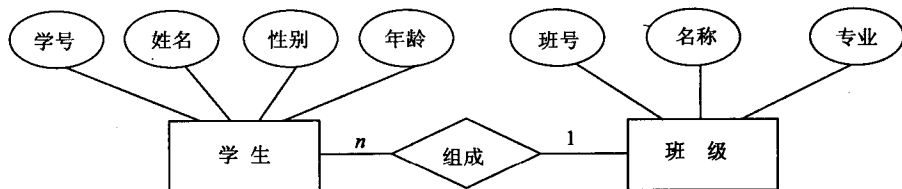


图 1-1 E-R 图

假设我们要解决的问题是学生的成绩与班级管理,并且要记录各班班主任的信息。基本情况是:一个学校有若干教师、若干个班、每班有若干学生、开设有若干课程,一个班有一个班主任,一个教师可以当多个班的班主任,一门课程可以有許多学生选修,一个学生也可以选修多门课程。根据需要,可以抽象出四个实体:“课程”、“学生”、“班级”与“教师”,其 E-R 图如图 1-2 所示。

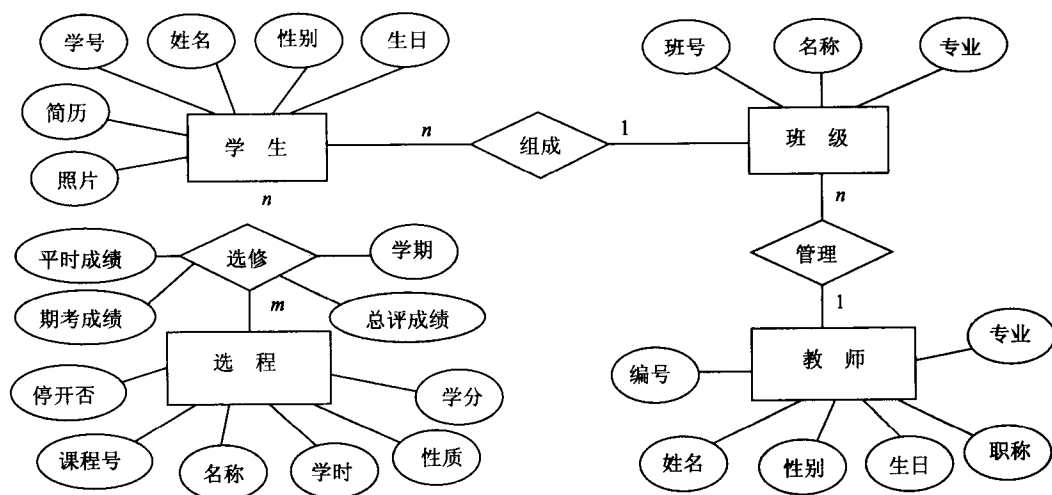


图 1-2 四个实体之间的 E-R 图

1.2 关系数据库

关系数据库采用关系模型作为数据的组织形式。关系数据库具有结构简单、理论基础坚实、数据独立性强以及提供非过程化语言等优点,是使用最广泛的一种数据库。

关系型数据库的真正发展应该归功于 IBM 公司的 E. F. Codd 研究员。从 1970 年开始,他连续地发表大量关于关系理论的论文,定义了关系数据库的基本概念,引进了规范化理论,提出了数据子语言及其完备性问题。此后 30 多年,人们在关系理论及关系型数据库管理系统的实现技术方面又进行了更广泛深入的研究,使得关系数据库得以不断发展。

1.2.1 关系数据模型

关系数据模型是用二维表框架来表示实体及其联系的模型,它的数据结构是一个二维表格结构,因此一个二维表也可以称为一个关系。表 1-1 至表 1-5 是根据图 1-2 给出的 E-R 图转化的关系模型实例。由此可知,关系模型实例是由若干二维表格(关系)组成。正因为关系模型采用了二维表格的数据结构,表格的框架构成了关系模型的结构,它由行和列组成,因而很容易被人们理解分解和采用。

表 1-1 教师

编号	姓名	生日	性别	职称	专业
1001	Mary	1970-1-30	女	教授	经济学
1101	李晓燕	1974-5-12	女	讲师	会计学
2201	张三	1960-5-14	男	副教授	物理学
2101	刘晓明	1950-7-11	男	教授	化学
2301	赵明	1970-4-23	男	副教授	计算机
2202	张三	1976-6-12	女	讲师	计算机

续表

编号	姓名	生日	性别	职称	专业
2102	林晓燕	1955-4-24	女	教授	计算机
1302	李玲玉	1970-1-30	女	讲师	数学
1303	张玉	1978-2-13	女	助教	数学

表 1-2 班 级

班号	名称	专业	班主任
990101	计算机 1	计算机	2101
990102	计算机 2	计算机	2101
990103	计算机 3	计算机	2102
990201	会计学 1	会计学	1101
990202	会计学 2	会计学	1101
990203	会计学 3	会计学	1001
990301	经济学 1	经济学	1001
990302	经济学 2	经济学	1001
990401	化学 1	化学	2301

表 1-3 课 程

课程号	名称	学时	学分	性质	停开否
JSJ001	计算机基础	80	5	公共基础	否
JSJ002	Visual FoxPro	80	5	公共基础	否
JSJ003	BASIC	80	5	公共基础	是
JSJ101	操作系统	80	5	专业基础	否
JSJ102	数据结构	64	4	专业基础	否
JSJ201	C++	48	3	专业主干	否
JSJ202	数据库系统	64	4	专业主干	否
JSJ301	计算机图形学	64	4	选修	否
JSJ302	密码学	48	3	选修	否
JSJ303	软件工程	48	3	选修	否

表 1-4 学 生

学号	姓名	生日	性别	班号	简历	照片
99030101	刘丹	1980-9-22	女	990301		
99030102	张三	1979-6-13	男	990301		
99030201	徐丽	1979-7-29	女	990302		
99030202	周智	1979-8-20	男	990302		
99010101	王明	1981-2-12	男	990101		
99010102	李四	1981-8-17	女	990101		
99010103	王燕	1980-7-13	女	990101		
99020101	傅慧	1980-8-24	女	990201		
99020102	李四	1979-1-24	男	990201		
99020201	王莉莉	1978-5-16	女	990202		
99020202	胡颖	1981-7-21	女	990202		
99010201	狄燕莉	1979-7-15	女	990102		
99010202	陈丽	1981-3-12	女	990102		
99010203	郭剑	1980-8-16	男	990102		

表 1-5 成 绩

学号	课程号	平时成绩	期考成绩	总评成绩	学期
99020201	JSJ001	100	90	95	1
99020202	JSJ001	100	80	85	1
99020201	JSJ002	100	50	60	2
99020202	JSJ002	100	70	80	2
99010201	JSJ101	100	80	85	1
99010202	JSJ101	90	70	75	1
99010203	JSJ101	70	50	55	1
99010201	JSJ102	95	60	65	2
99010202	JSJ102	80	80	80	2
99010203	JSJ102	100	100	100	2
99010101	JSJ201	100	100	100	3
99010102	JSJ201	70	20	30	3
99010103	JSJ201	60	60	60	3
99010201	JSJ201	100	70	70	3
99010202	JSJ201	60	60	60	3

关系模型中的主要概念有：

(1) 关系：也称表，它由行和列组成。在 Visual FoxPro 中，一个关系存储为一个文件，文件扩展名为 .DBF。

(2) 属性：表中的一列即为一个属性，在 Visual FoxPro 中也称字段。

(3) 元组：表中的一行称为关系的一条元组，在 Visual FoxPro 中也称记录。

(4) 分量：元组中的一个属性值。

(5) 关键字：关键字是唯一标识元组的属性集。

如表 1-4 学生关系中，属性“学号”就具有这种特性，它能够唯一地确定一条学生元组，所以学号是学生的关键字。而在表 1-5 选修关系中，“学号”和“课程号”的组合具有唯一确定一条元组的特性，则这两个属性构成选修关系的复合关键字。在具有多个关键字时一般要选择一个作为主关键字。主关键字在插入、删除、检索记录等数据操作起重要作用。

(6) 外部关键字：如果表中的一个属性不是本表的关键字，而是另一个表的关键字，则称该属性为外部关键字。

如属性“班号”在学生关系中是外部关键字，因为它是班级表的关键字，但它不是学生表的关键字。

(7) 关系模式：对关系的描述，一般表示为：

关系名(属性 1, 属性 2, ..., 属性 n)

一个关系有 n 个属性则称为 n 元关系，如二元关系、三元关系等。

关系模型要求关系必须是规范的，即关系必须满足一定的规范条件，其中最基本的条件是：关系的每一个分量必须是一个不可分的数据项。

1.2.2 关系运算

建立关系数据库的目的就是为了利用数据，对关系数据库中的数据进行操作就是关系运算。关系运算可分为传统的集合运算和专门的关系运算两大部分，其中并、差、乘积、选择、投影是基本的关系运算，其他运算可用这五种基本运算合成。

1. 集合运算

在集合运算中,关系是由记录组成的集合。

(1) 并(Union)

两个具有相同关系模式的关系 R_1 和 R_2 的并运算产生一个新关系 R_3 ,它由属于关系 R_1 或 R_2 、或同时属于 R_1 和 R_2 的记录组成,记为:

$$R_3 = R_1 \cup R_2$$

(2) 差(Difference)

两个具有相同关系模式的关系 R_1 和 R_2 的差运算产生一个新的关系 R_4 ,该关系由属于 R_1 但不属于 R_2 的记录组成,记为:

$$R_4 = R_1 - R_2$$

(3) 交(Intersection)

两个具有相同关系模式的关系 R_1 和 R_2 的交运算产生一个新关系 R_5 ,该关系由既属于 R_1 同时又属于 R_2 的记录组成,记为:

$$R_5 = R_1 \cap R_2$$

交运算可用差运算表示为: $R_1 \cap R_2 = R_1 - (R_1 - R_2)$

(4) 乘积(Product)

也称笛卡尔积(Cartesian Product),设 R 为 m 元关系, S 为 n 元关系, R 和 S 的笛卡尔积产生一个新关系 T ,记为:

$$T = R \times S$$

T 由 R 和 S 的所有记录两两联接而成,它具有 $m+n$ 个属性分量和 $m \times n$ 条记录。

若 R 的关系模式为 $R(A_1, A_2, \dots, A_n)$, S 的关系模式为 $S(B_1, B_2, \dots, B_m)$,则 T 的关系模式为 $T(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$ 。

若 $(a_1, a_2, \dots, a_n)$ 是 R 中一条记录, $(b_1, b_2, \dots, b_m)$ 是 S 中一条记录,则 $(a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m)$ 必是 T 中的一条记录。

表 1-6 中给出了一个具体的例子。 S_1 为班级信息, S_2 为课程信息,如 $S_3 = S_1 \times S_2$,则 S_3 为学生班 S_1 选修课程 S_2 的情况。

表 1-6 关系的笛卡尔积运算

S_1			$S_3 = S_1 \times S_2$					
班级编号	班级名称	专业	班级编号	班级名称	专业	课程编号	课程名称	学分
2002110301	信息 1	自动化	2002110301	信息 1	自动化	1011	计算机绘图	2
2002110305	信息 2	计算机	2002110305	信息 2	计算机	1101	FoxPro	4
S_2			2002110301	信息 1	自动化	1108	数据库原理	4
课程编号	课程名称	学分	2002110305	信息 2	计算机	1011	计算机绘图	2
1011	计算机绘图	2	2002110301	信息 1	自动化	1101	FoxPro	4
1101	FoxPro	4	2002110305	信息 2	计算机	1108	数据库原理	4
1108	数据库原理	4						