

21世纪高等院校教材

Visual FoxPro

面向对象程序设计教程

黎升洪 杨波 沈波 编著
徐升华 主审



科学出版社
www.sciencep.com

21 世纪高等院校教材

Visual FoxPro 面向对象程序设计教程

黎升洪 杨 波 沈 波 编著

徐升华 主审

科学出版社

北 京

内 容 简 介

本书是 Visual FoxPro 6.0 的初、中级教程,共分 11 章,介绍了关于数据、信息、关系数据库、结构化程序编程和面向对象编程的基本概念、知识,主要针对非计算机专业学生的特点,以应用为目的,从简单实例出发,强调对 Visual FoxPro 操作的实用性。本书试图注重概念的严谨和清晰,同时又注意采用读者容易理解的方法阐明看似深奥难懂的问题,力图做到例题丰富,通俗易懂,便于自学。

本书可供高校或培训班用作非计算机类学生教材,也可作为 Visual FoxPro 计算机等级考试的参考资料,或供广大计算机爱好者自学用。

图书在版编目(CIP)数据

Visual FoxPro 面向对象程序设计教程/黎升洪,杨波,沈波编著. - 北京:科学出版社, 2004

(21 世纪高等院校教材)

ISBN 7-03-014834-7

I. V... II. ①黎...②杨...③沈... III. 关系数据库 - 数据库管理系统, Visual FoxPro - 程序设计 - 高等学校 - 教材 IV. TP311.138

中国版本图书馆 CIP 数据核字 (2004) 第 143035 号

责任编辑:杨瑰玉/责任印制:高 嵘

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

湖北北京山德新印刷有限公司印刷

科学出版社发行 各地新华书店经销

*

2004 年 12 月第 一 版 开本: 787×1092 1/16

2004 年 12 月第一次印刷 印张: 18 1/4

印数: 1~7 000

字数: 448 000

定价: 30.00 元

(如有印装质量问题,我社负责调换)

前 言

随着当今科技的迅猛发展,网络化、数字化等各种信息技术的变化层出不穷,21世纪将以信息为基础的知识经济时代。掌握计算机知识和应用无疑是现代大学生应该具备的基本技能。本书是 Visual FoxPro 6.0 的初、中级教程,介绍了关于数据、信息、关系数据库、结构化程序编程和面向对象编程的基本概念、知识。本书针对非计算机专业学生的特点,以应用为目的,从简单实例出发,强调对 Visual FoxPro 操作的实用性。本书试图注重概念的严谨和清晰,同时又注意采用读者容易理解的方法阐明看似深奥难懂的问题,力图做到例题丰富,通俗易懂,便于自学。书中不但举了许多小例子,而且以教学管理为贯穿全书的实例。

全书总共分 11 章。第一章介绍数据库的基本概念和基础知识,包括关系、数据库的概念,数据库的应用模式等基础知识。第二章介绍 Visual FoxPro 安装、使用界面和项目管理器的使用。第三章介绍了 Visual FoxPro 6.0 有关常量、变量、函数和表达式的基础知识。第四章以教学管理数据库为样本,介绍如何对数据库进行检索、创建索引以及如何建立数据库和数据表等操作,介绍了“表设计器”、“数据库设计器”等操作方法。第五章对 SQL 中关于 SELECT 语句进行了详细的介绍,在介绍数据库多表操作和数据库模式上有独到的例子,同时介绍了数据操纵语言以及“查询设计器”编写 SQL 中 SELECT 和 VIEW 的方法。第六章介绍了程序设计的流程和流程图等基本概念,重点介绍了在 Visual FoxPro 中编写结构化程序的方法,同时给出了过程和子程序等编程实例,并通过实例介绍了程序调试方法。第七章介绍面向对象中的核心概念,给出了 Visual FoxPro 中所提供的对象及如何引用这些对象的属性和方法,通过实例展示面向对象编程一次编写多次使用的优点。第八章介绍了 Visual FoxPro 中图形化界面下的程序设计方法,介绍 Visual FoxPro 6.0 “表单设计器”和常见的控件使用方法及实例。第九章介绍了在 Visual FoxPro 下设计菜单系统的方法。第十章介绍在 Visual FoxPro 下利用“报表设计器”设计报表的方法。第十一章结合教学管理数据库,给出了一个应用系统的开发全过程。

本书根据作者多年从事计算机应用基础教学、研究和 Visual FoxPro 软件开发实践的经验编写而成。杨波编写了第 8、9、10、11 章;沈波编写了第 2、4 章;黎升洪完成了本书初始的规划,并编写了第 1、3、5、6、7 章及附录,最后对全书进行统稿。本书荣幸地由徐升华教授担任主审,徐教授对本书初稿提出了许多宝贵的意见和建议。同时感谢万常选教授对本书编写提出了许多有益的建议;江腾蛟、罗清胜等对本书的编写给予了大力的支持和帮助。本书所有例子均有电子文档,有需要者请与 lish@jxufe.edu.cn 联系。

本书可供高校或培训班用作非计算机类学生教材,也可作为 Visual FoxPro 计算机等级考试的参考资料,或供广大计算机爱好者自学用。

由于编者水平有限,书中难免有许多不足或疏漏之处,恳请专家和广大读者批评指正。

编 者

2004 年 11 月

目 录

第一章 数据库系统概论.....	1
1.1 数据库技术.....	1
1.1.1 信息与数据.....	1
1.1.2 数据处理.....	1
1.1.3 数据库系统.....	2
1.2 数据模型.....	3
1.2.1 数据模型的三要素.....	4
1.2.2 概念模型与实体-联系方法.....	5
1.2.3 数据模型.....	6
1.3 关系数据库.....	9
1.3.1 关系模型.....	9
1.3.2 关系完整性约束.....	12
1.4 数据库系统应用模式.....	14
本章关键词.....	15
习题.....	15
第二章 Visual FoxPro 操作基础.....	17
2.1 Visual FoxPro 简介.....	17
2.2 Visual FoxPro 的安装与启动.....	18
2.2.1 Visual FoxPro 的安装.....	18
2.2.2 Visual FoxPro 的启动与退出.....	19
2.3 Visual FoxPro 的用户界面.....	20
2.3.1 Visual FoxPro 系统主界面.....	20
2.3.2 Visual FoxPro 命令窗口.....	21
2.3.3 Visual FoxPro 系统菜单.....	21
2.3.4 Visual FoxPro 工具栏.....	23
2.4 Visual FoxPro 操作概述.....	23
2.4.1 Visual FoxPro 操作方式.....	23
2.4.2 Visual FoxPro 可视化设计工具.....	23
2.4.3 Visual FoxPro 系统环境的设置.....	26
2.4.4 Visual FoxPro 帮助系统的使用.....	27
2.5 Visual FoxPro 命令概述.....	28
2.5.1 Visual FoxPro 命令的结构.....	28
2.5.2 Visual FoxPro 命令中的常用短语.....	28
2.5.3 Visual FoxPro 命令的书写规则.....	28
2.5.4 Visual FoxPro 保留字.....	29

2.6 Visual FoxPro 项目管理器	29
2.6.1 项目文件的建立与打开	29
2.6.2 Visual FoxPro 项目管理的使用	30
本章关键词	32
习题	33
第三章 Visual FoxPro 语言基础	34
3.1 数据类型	34
3.2 常量	34
3.3 变量	35
3.3.1 变量定义与特性	35
3.3.2 内存变量	36
3.4 函数	39
3.4.1 函数概述	39
3.4.2 数值处理函数	40
3.4.3 字符函数	41
3.4.4 日期和时间处理函数	44
3.4.5 数据类型转换函数	44
3.4.6 数据表处理函数	45
3.5 Visual FoxPro 的表达式	48
3.5.1 算术表达式	48
3.5.2 字符表达式	48
3.5.3 日期和时间表达式	49
3.5.4 关系表达式	49
3.5.5 逻辑表达式	50
本章关键词	51
习题	51
第四章 Visual FoxPro 数据库操作基础	53
4.1 表的基本操作	53
4.1.1 工作区和表的打开与关闭	54
4.1.2 表的复制	57
4.2 表记录的基本操作	58
4.2.1 表记录指针的定位	58
4.2.2 表记录的显示与浏览	60
4.2.3 表记录的修改	64
4.2.4 数据表记录的增加	66
4.2.5 表记录的删除	68
4.3 表的索引与排序	70
4.3.1 表的索引	70
4.3.2 表的排序	75
4.4 表记录的查找	76

4.4.1	顺序查找	76
4.4.2	快速查找	77
4.5	表的统计与计算	78
4.5.1	统计记录个数	78
4.5.2	统计累加和	79
4.5.3	统计平均值	79
4.5.4	财务统计	80
4.5.5	分类汇总	80
4.6	表的临时关联	81
4.6.1	建立表的临时关联	81
4.6.2	取消表的临时关联	82
4.6.3	建立记录的一对多联系	82
4.7	数据表的建立与修改	82
4.7.1	数据表的建立	82
4.7.2	数据表结构的修改	85
4.8	数据库的基本操作	86
4.8.1	利用数据库设计器建立数据库	88
4.8.2	数据库的打开与关闭	89
4.8.3	数据库的修改与删除	90
4.8.4	向数据库中添加表	91
4.8.5	建立数据表间的外键约束	93
4.8.6	数据库建立实例	97
	本章关键词	98
	习题	99
第五章	Visual FoxPro 中 SQL 语言的应用	99
5.1	SQL 语言概述	99
5.1.1	SQL 的历史和标准	99
5.1.2	SQL 的优点	100
5.1.3	基本表和视图	100
5.2	数据查询	101
5.2.1	单表查询	101
5.2.2	多表查询	109
5.2.3	创建和使用视图	115
5.3	数据操纵	117
5.3.1	插入记录	117
5.3.2	删除记录	118
5.3.3	更新记录	118
5.4	数据定义	118
5.4.1	建立表结构	118
5.4.2	删除表	121

5.4.3 修改表结构·····	121
5.5 Visual FoxPro 查询设计器·····	121
5.5.1 查询的建立·····	121
5.5.2 查询的保存、修改和输出重定向·····	126
5.5.3 使用视图设计器设计视图·····	127
本章关键词·····	128
习题·····	129
第六章 Visual FoxPro 程序设计基础 ·····	131
6.1 Visual FoxPro 程序的建立与执行·····	131
6.1.1 Visual FoxPro 程序的基本构成·····	131
6.1.2 程序文件的建立和编辑·····	131
6.1.3 程序文件的编译和执行·····	133
6.1.4 Visual FoxPro 命令方式的输入输出语句·····	134
6.2 结构化程序设计基础·····	137
6.2.1 程序设计过程·····	137
6.2.2 算法·····	138
6.2.3 程序流程图及示例·····	138
6.3 Visual FoxPro 程序的控制语句结构·····	142
6.3.1 顺序结构程序·····	142
6.3.2 分支结构程序·····	142
6.3.3 循环结构程序·····	147
6.3.4 程序设计举例·····	153
6.4 数组的应用·····	155
6.4.1 数组中常用的语句·····	155
6.4.2 数组中常用的函数·····	157
6.5 程序的模块化方法·····	158
6.5.1 子程序·····	158
6.5.2 过程·····	162
6.5.3 自定义函数·····	164
6.6 Visual FoxPro 程序调试方法·····	165
6.6.1 程序错误的种类·····	165
6.6.2 使用 Visual FoxPro 调试器调试程序·····	167
本章关键词·····	173
习题·····	173
第七章 面向对象程序设计基础 ·····	177
7.1 面向对象的基本概念·····	177
7.1.1 面向对象核心概念·····	177
7.1.2 类或对象的特性·····	178
7.2 Visual FoxPro 6.0 对面向对象方法的支持·····	179
7.3 Visual FoxPro 面向对象的程序设计中的其他问题·····	185

7.3.1 控件焦点与 Tab 序	185
7.3.2 Visual FoxPro 中关于类的操作	186
本章关键词	190
习题	191
第八章 表单设计与应用	192
8.1 表单的建立与运行	192
8.1.1 创建表单	192
8.1.2 修改表单	197
8.1.3 表单的运行	198
8.2 表单的操作	199
8.2.1 表单设计器环境	199
8.2.2 表单的属性	201
8.2.3 表单的事件与方法	202
8.2.4 数据环境	204
8.2.5 表单控件的操作与布局	206
8.3 常用表单控件	207
8.3.1 命令按钮控件	207
8.3.2 命令按钮组控件	209
8.3.3 文本框控件	211
8.3.4 编辑框控件	213
8.3.5 复选框控件	214
8.3.6 单选按钮控件	216
8.3.7 标签控件	217
8.3.8 列表框控件	218
8.3.9 组合框控件	220
8.3.10 计时器控件	221
8.3.11 微调控件	222
8.3.12 页框控件	223
8.3.13 表格控件	225
本章关键词	230
习题	230
第九章 菜单设计	234
9.1 菜单设计概述	234
9.1.1 菜单的结构	234
9.1.2 菜单设计的基本过程	235
9.1.3 系统菜单的控制	236
9.2 下拉式菜单设计	236
9.2.1 创建主菜单	237
9.2.2 创建子菜单	237
9.2.3 指定菜单项任务	238

9.2.4 生成和运行菜单·····	239
9.2.5 为顶层表单添加菜单·····	239
9.3 快捷菜单设计·····	241
本章关键词·····	243
习题·····	243
第十章 报表设计 ·····	246
10.1 利用报表向导创建报表·····	246
10.2 利用快速报表创建报表·····	248
10.3 利用报表设计器创建设计报表·····	250
10.3.1 报表数据源和布局·····	251
10.3.2 在报表中使用控件·····	253
10.4 数据分组和多栏报表·····	259
10.4.1 设计数据分组报表·····	259
10.4.2 设计多栏报表·····	262
10.5 报表输出·····	263
本章关键词·····	263
习题·····	263
第十一章 小型系统开发实例 ·····	266
11.1 系统开发的一般过程·····	266
11.2 系统总体规划·····	266
11.2.1 设计系统规划方案·····	266
11.2.2 主程序设计·····	267
11.2.3 系统登录表单的设计·····	268
11.2.4 系统主菜单设计·····	269
11.2.5 学生信息登记表单的设计·····	270
11.2.6 学生信息查询表单的设计·····	270
11.2.7 课程信息维护表单的设计·····	270
11.2.8 学生选课管理表单的设计·····	271
11.3 系统部件的组装·····	271
11.4 运行选课管理系统·····	274
习题·····	276
附录一: Visual FoxPro 常用函数一览表 ·····	277
附录二: Visual FoxPro 文件扩展名的含义一览表 ·····	279
主要参考文献 ·····	280

第一章 数据库系统概论

随着计算机技术的蓬勃发展,计算机应用已经涉足到人们日常生活、工作的各个领域。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、科学计算、过程控制和辅助设计是计算机四大应用,而数据处理的主要技术是数据库技术。本章讲解数据库技术的基本知识和概念,重点是关系数据库和数据库模式的概念。

1.1 数据库技术

在信息社会中,信息是一种资源。对企业来说,各种必需的信息是其赖以生存和发展的根本;对一个国家来说,信息决定其如何建设和发展;对一个人来说,信息是其决定如何发展才能适应社会的基本要素。信息是维持生产活动、经济活动和社会活动必不可少的基本资源,它是有价值的,是构成客观世界的三大要素(信息、能源和材料)之一。因此,人们为了获取有价值的信息用于决策,就需要对信息和用于表示信息的数据进行处理和管理。人们用计算机对数据进行处理的应用系统称为计算机信息系统,而计算机信息系统的核心是数据库。

1.1.1 信息与数据

信息和数据是数据处理中的两个基本概念。在一些不是很严格的场合下,对它们没有做严格的区分,甚至当作同义词来使用。这里,数据是记录现实世界中各种信息并可以识别的物理符号,是信息的载体,是信息的具体表现形式。数据的表示形式不仅仅只是数字,还包括字符(文字和符号)、图表(图形、图像和表格)及声音等形式。数据以格式化的形式来表示事实和概念,这种形式有助于通信、解释和处理。数据有两方面的特征:一是客体属性的反映,这是数据的内容;二是记录信息的符号,这是数据的形式。

信息是数据所包含的意义。信息具有如下重要特征:

- (1) 信息具有表征性。它能够表达事物的属性、运动特性及状态。
- (2) 信息具有可传播性。信息可以进行获取、存储、传递、共享。
- (3) 信息具有可处理性。信息可以进行压缩、加工、再生。
- (4) 信息具有可用性、可增值性、可替代性。

数据与信息是密切关联的。信息是向人们提供有关现实事物的知识,数据则是载荷信息的物理符号,二者是不可分离而又有一定区别的两个相关的概念。信息可以用不同形式的数据来表示,也不随它的数据形式不同而改变。例如,张平同学的高考成绩总分为630分。这里符号630就是数据;630解释为高考成绩总分,表示的是630的含义——即信息。

总之,数据形式是信息内容的表现方式,信息内容是数据形式的实质,即“数据是信息的载体,信息是数据的内涵。”

1.1.2 数据处理

要使获得的信息能够充分地发挥作用,就必须对其进行处理。这种处理称为信息处理,

常常又称为数据处理。严格地说,信息处理中包含了数据处理,而数据处理是信息处理最主要的内容。数据处理实际上是指利用计算机对各种形式的数据进行一系列的存储、加工、计算、分类、检索、传输等处理。如果稍加扩展就包括数据的采集、整理、编码、输入和输出等数据组织,这一数据组织过程也应属于数据处理的内容,只不过这一过程主要是由人对其进行有效的处理,并把数据组织到计算机中。

1.1.3 数据库系统

一、数据库的概念

在日常工作中,需要处理的数据量往往都很大,为便于计算机对其进行有效的处理,我们可以将采集的数据存放在建立于磁盘、光盘等外存媒介的“仓库”中,这个“仓库”就是数据库(database 或 data base,简称 DB)。数据集中存放在数据库中,便于对其进行处理,提炼出对决策有用的数据和信息。这就如同一个工厂生产出产品要先存放在仓库中,既便于管理,又便于分期分批地销售;一个学校采购大量的图书存放在图书馆(书库),供学生借阅。因此,数据库就是在计算机存储器中用于存储数据的仓库。

与货仓、书库需要管理员和一套管理制度一样,数据库的管理也需要一个管理系统,这个管理系统就称为数据库管理系统(database management system,简称 DBMS)。以数据库为核心,并对其进行管理的计算机系统称为数据库系统(data base system,简称 DBS)。

那么,什么是数据库呢?数据库是一个复杂的系统,给它下一个确切的定义是困难的,目前还没有一个公认的、统一的定义。但具体地,对一个特定数据库来说,它是集中、统一地保存、管理着某一单位或某一领域内所有有用信息的系统,这个系统根据数据间的自然联系结构而成,数据较少冗余,且具有较高的数据独立性,能为多种应用服务。

二、数据库的发展

数据管理的发展经历了人工管理、文件系统到数据库系统三个阶段。

在人工管理阶段,由于没有软件系统对数据进行管理和计算机硬件的限制,数据的管理是靠人工进行的,而计算机只能对数据进行计算。当时对数据处理的过程是,先将程序和数据输入计算机,计算机运行结束后,将结果再输出,由人工保存,计算机并不存储数据。

20 世纪 50 年代后期到 60 年代中期,由于计算机外存得到发展,软件又有了操作系统,对数据管理便产生了文件系统。在文件系统阶段,人们是按照数据文件的形式来存放数据的,在一个文件中包含若干个“记录”,一个记录又包含若干个“数据项”,用户通过对文件的访问实现对记录的存取。这种数据管理方式称为文件管理系统。文件管理系统的一个致命的不足是数据的管理没有实现结构化组织,数据与数据之间没有联系,文件与文件之间没有有机的联系,数据不能脱离建立其数据文件的程序,从而也使文件管理系统中的数据独立性和一致性差,冗余度大,限制了大量数据的共享和有效的应用。

20 世纪 60 年代末期,随着计算机技术的发展,为了克服文件管理系统的缺点,人们对文件系统进行了扩充,研制了一种结构化的数据组织和处理方式,即数据库系统。数据库系统建立了数据与数据之间的有机联系,实现了统一、集中、独立地管理数据,使数据的存取独立于使用数据的程序,实现了数据的共享。从 20 世纪 90 年代至今,数据库技术得到了飞速的发展。

三、数据库的特征

作为信息管理中的核心技术,数据库技术在计算机应用中得到迅速的发展,目前已经成

为信息管理中最重要、最新的技术。数据库有以下明显特点：

(1) 数据结构化。数据库中的数据不再像文件系统中的数据那样从属特定的应用，而是按照某种数据模型组织成为一个结构化的数据整体。它不仅描述了数据本身的特性，而且描述了数据与数据之间的种种联系，这使数据库具备了复杂的内部组织结构。

(2) 实现数据共享。这是数据库技术先进性的重要体现。由于数据库中的数据实现了按某种数据模型组织为一个结构化的数据，实现了多个应用程序、多种语言及多个用户共享一个库中的数据，甚至在一个单位或更大的范围内共享，大大提高了数据的利用率，提高了工作效率。

(3) 减少数据冗余度。在数据库技术产生之前，许多应用系统都需要建立各自的数据文件，即使相同的数据也都需要在各自的系统中保留，造成大量的数据重复存储，这一现象称为数据的冗余。由于数据库实现了数据共享，从而减少了存储数据的重复，节省了存储空间，减少了数据冗余。

(4) 数据独立性。数据库技术中的数据与操作这些数据的应用程序相互独立，互不依赖，不因一方的改变而改变另一方，这大大简化了应用程序设计与维护的工作量，同时数据也不会随应用程序的结束而消失，可长期保留在计算机系统中。

(5) 统一的数据安全保护。数据共享在提供多个用户共享数据资源的同时，还需解决数据的安全性、一致性和并发性问题。这里安全性是指只有合法授权的用户才能对数据进行操作；一致性是指当多个用户对同一数据操作时不能互相干扰，以免出现操作结果不确定或不一致的情况；在保证一致性的前提下，数据库系统提供并发功能，使多用户对数据库同时操作有一致的正确结果。

四、数据库的构成

一个数据库系统至少应该包括以下三个部分：

(1) 数据库。一个结构化的相关数据的集合，包括数据本身和数据间的联系。它独立于应用程序而存在，是数据库系统的核心和管理对象。

(2) 物理存储设备。这是保存数据的硬件介质，一般为磁盘等大容量的存储器。

(3) 数据库软件。负责对数据库管理和维护的软件，具体完成对数据定义、描述、操作、维护等功能，接受并完成用户程序及终端命令对数据库的不同请求，并负责保护数据免受各种干扰和破坏。数据库软件的核心是数据库管理系统。

另外，数据库系统需要有专门的人员负责数据库的设计、建立、执行和维护，这些人员称为数据库管理员(database administrator—DBA)。

1.2 数据模型

提到模型我们自然会联想到建筑模型、飞机模型等具体事物。广义地说，模型是现实世界特征的模拟和抽象。在数据库中，用数据模型(data model)这个工具来对现实世界进行抽象，数据模型是数据库系统中用于提供信息表示和操作手段的形式构架。数据模型应满足三方面要求：一是能比较真实地模拟现实世界；二是容易为人所理解；三是便于在计算机上实现。数据模型要很好地满足这三方面的要求在目前尚很困难。

在数据库系统中针对不同的使用对象和应用目的，可采用不同的数据模型。不同的数据模型是提供给我们模型化数据和信息的不同工具。根据模型应用的目的，可以将数据模型分

为两种类型：第一类是概念模型，也称信息模型，它是独立于计算机之外的模型，如实体-联系模型，这种模型不涉及信息在计算机中如何表示，而是用来描述某一特定应用范围内人们所关心的信息结构，它是按用户的观点来对数据和信息建模，主要用于数据库设计；另一类模型是直接面向计算机的，是按计算机系统的观点对数据进行建模，主要用于 DBMS 的实现，常称为基本数据模型或数据模型，数据库中常用的基本数据模型有网状模型、层次模型和关系模型。

数据模型是数据库系统的核心和基础。各种机器上实现的 DBMS 软件都是基于某种数据模型的。

图 1.1 显示了把现实世界中的具体事物抽象、组织为某一 DBMS 支持的数据模型的过程。在概念上我们常常首先将现实世界抽象为信息模型(也称为概念模型)，然后将信息模型(概念模型)转换为计算机实现的世界。具体来说，现实世界的现实事务经过信息抽象转化为信息模型，信息模型使用实体-联系模型表示，再通过数据抽象，将信息模型转化为计算机实现，这里计算机实现使用数据模型表示。

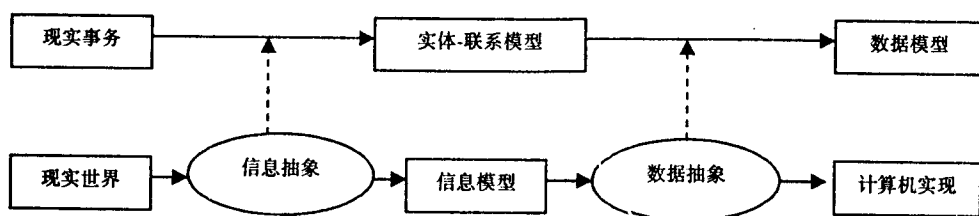


图 1.1 数据抽象过程

1.2.1 数据模型的三要素

数据模型是现实世界中的各种事物及其相互联系用数据和数据间的联系来表示的一种方法。一般地讲，数据模型是严格定义的概念的集合，这些概念精确地描述系统的静态特性、动态特性和完整性约束条件。因此，数据模型通常由数据结构、数据操作和完整性约束三部分组成。

一、数据结构

数据结构是所研究对象和对象具有的特性、对象间的联系的集合，它是对数据静态特性的描述。这些对象是数据库的组成部分，如网状模型中的数据项、记录、系型，关系模型中的域、属性、关系等。

在数据库系统中，通常按照数据结构的类型来命名数据模型，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。

二、数据操作

数据操作是指对数据库中各种对象(型)的实例(值)允许执行的操作的集合，包括操作及有关的操作规则。通常对数据库的操作有检索和更新(包括插入、删除和修改)两大类，这些操作反映了数据的动态特性，因为现实世界中的实体及实体间的联系是在不断变化的，数据模型应能反映出这种变化。

三、数据的约束条件

数据的约束条件是完整性规则的集合。完整性规则是指给定的数据模型中数据及其联系必须满足给定的要求。例如，年龄的数据取值不能大于 150 岁。

1.2.2 概念模型与实体-联系方法

由图 1.1 可以看出,信息模型(概念模型)实际上是现实世界到机器世界的一个中间层次。概念模型用于信息世界的建模,是现实世界到信息世界的第一层抽象,是数据库设计人员进行数据库设计的有力工具,也是数据库设计人员和用户之间进行交流的语言。因此概念模型一方面应该具有较强的语义表达能力,能够方便、直接地表达应用中的各种语义知识,另一方面它还应该简单、清晰,易于用户理解。

一、信息世界中的基本概念

信息世界涉及的概念主要有:

(1) 实体(entity)。客观存在并可相互区别的事物称为实体。实体可以是具体的人、事、物,也可以是抽象的概念或联系,例如,一个具体学生、一门具体课程等都是实体。

(2) 实体集(entity set)。性质相同的同类实体的集合称为“实体集”,也称为实体整体,如所有的(全体)学生、所有的汽车、所有的学校、所有的课程、所有的零件都称为实体集。

(3) 实体型(entity type)。具有相同属性的实体必然具有共同的特征和性质。用实体名及其属性名集合来抽象和刻画同类实体,称为实体型。例如,学生(学号,姓名,性别,出生年份,系,入学时间)就是一个实体型。

(4) 属性(attribute)。实体所具有的某一特性称为属性。一个实体可以由若干个属性来刻画。例如学生实体可以由学号、姓名、性别、出生年份、系、入学时间等属性组成。属性有“型”和“值”的区分,如学生实体属性的名称:姓名、性别、年龄等是属性的型,而属性的值是其型的具体内容,如王源、男、18 分别是姓名、性别、年龄的值。由此可以看到,事物的若干属性值的集合可表征一个实体,而若干个属性型所组成的集合可表征一个实体的类型,简称为“实体型”。同类型的实体集合组成实体集。

(5) 关键字(key)。能惟一标识实体的属性或属性集称为关键字(或码)。例如学号是学生实体的关键字(码)。本书将混用关键字和码这两个概念。

(6) 域(domain)。属性的取值范围称为该属性的域。例如,学号的域为 8 位数字符号,年龄的域为小于 150 的整数,性别的域为(男,女)。

(7) 联系(relationship)。在现实世界中,事物内部以及事物之间是有联系的,这些联系在信息世界中反映为实体(型)内部的联系和实体(型)之间的联系。实体内部的联系通常是指组成实体的各属性之间的联系。实体之间的联系通常是指不同实体集之间的联系。

两个实体型之间的联系可以分为三类:

1) 一对一联系(1:1)。如果实体集 A 中的一个实体至多与实体集 B 中的一个实体相对应(相联系),反之亦然,则称实体集 A 与实体集 B 的联系为一对一的联系。如一个学校只能有一个校长,一个校长也只能在一个学校任职,则学校与校长的联系即为一对一的联系;还有班长与班、学生与座位之间也都是一对一的联系。

2) 一对多联系(1:n)。如果实体集 A 中的一个实体与实体集 B 中的多个实体相对应(相联系),反之,实体集 B 中的一个实体至多与实体集 A 中的一个实体相对应(相联系),则称实体集 A 与实体集 B 的联系为一对多的联系。如一个班级可以有多个学生,而一个学生只会有一个班级,班级与学生的联系即为一对多的联系。

3) 多对多联系(m:n)。如果实体集 A 中的一个实体与实体集 B 中的多个实体相对应(相联系),而实体集 B 中的一个实体也与实体集 A 中的多个实体相对应(相联系),则称实体集 A

与实体集 B 的联系为多对多的联系。如一门课程可以有多个学生选修，而一个学生同时可以选修多门课程，课程与学生的联系即为多对多的联系。

实际上，一对一联系是一对多联系的特例，而一对多联系又是多对多联系的特例。可以用图形来表示两个实体型之间的这三类联系，如图 1.2 所示。一般地，两个以上的实体型之间也存在着一对一、一对多、多对多联系。

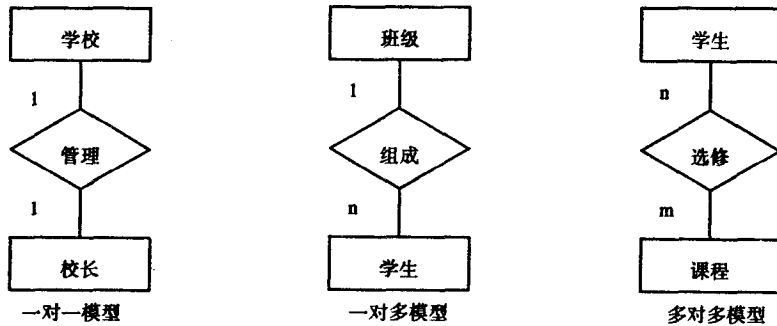


图 1.2 实体间的联系

二、概念模型的表示方法

为了在信息世界中简洁、清晰地描述现实世界的实体模型，通常使用 E-R 图描述，E-R 图是 P. P. S. Chen 于 1976 年提出的实体联系模型(entity-relationship model)。E-R 图提供了实体、属性与联系的方法，其图元符号如图 1.3 所示。在 E-R 图中：

- 实体集用矩形框表示，并在矩形框里写上实体名
- 属性用椭圆框表示，并在椭圆框里写上属性名
- 联系用菱形框表示，并在菱形框里写上联系名



图 1.3 E-R 使用的符号

如图 1.2 所示，分别是学校与校长(一对一)、班级与学生(一对多)、学生与课程(多对多)的 E-R 实体模型图。班级、学生和课程对应的 E-R 图如图 1.4 所示，由于实体班级、学生和课程属性太多，我们这里只画出部分属性。

1.2.3 数据模型

目前，数据库领域中最常用的数据模型有四种，它们是：

- 层次模型(hierarchical model)
- 网状模型(network model)
- 关系模型(relational model)
- 面向对象模型(object oriented model)

关系模型是本章的重点，我们将在 1.3 中加以介绍。

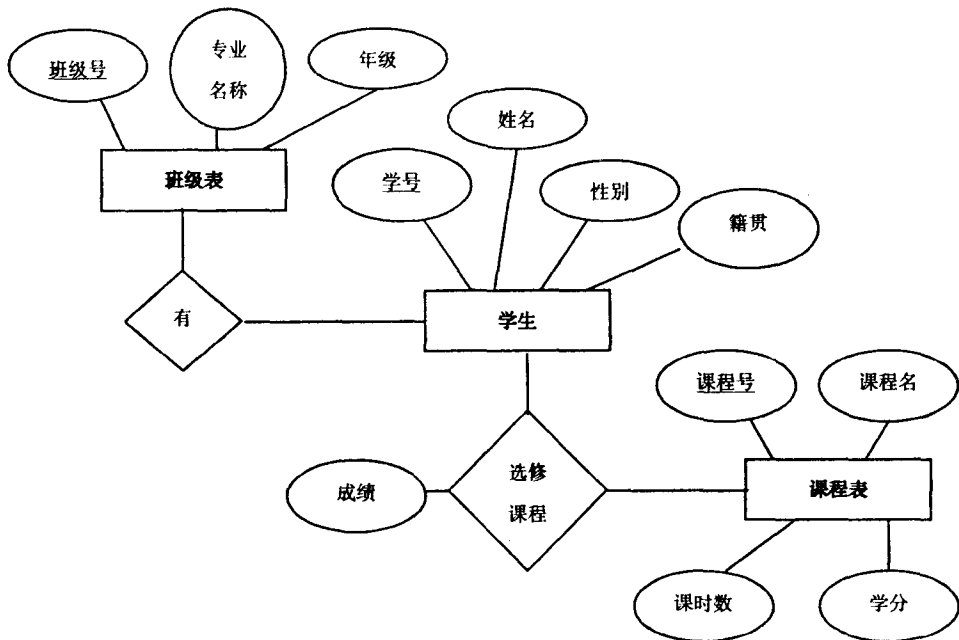


图 1.4 班级、学生和课程对应的 E-R 图(省略了部分属性)

一、层次模型

层次模型用树形结构来表示各类实体以及实体间的联系。现实世界中许多实体之间的联系本来就呈现出一种很自然的层次关系，如行政机构、家族关系等。

在层次模型中，每个结点表示一个记录类型，记录(类型)之间的联系用结点之间的连线(有向边)表示，这种联系是父子之间的一对多的联系。这就使得层次数据库系统只能处理一对多的实体联系。

每个记录类型可包含若干字段，这里，记录类型描述的是实体，字段描述的是实体的属性。各个记录类型及其字段都必须命名。各个记录类型、同一记录类型中各个字段不能同名。每个记录类型可以定义一个排序字段，也称为码字段，如果定义该排序字段的值是惟一的，则它能惟一地标识一个记录值，如图 1.5 所示为一个层次模型。这里，结点 R_1 是结点 R_2 、 R_3 的双亲结点，结点 R_2 、 R_3 是结点 R_1 的子女结点。

层次模型的主要优点有：

- (1) 层次数据模型本身比较简单。
- (2) 对于实体间联系是固定的且预先定义好的应用系统，采用层次模型来实现，其性能优于关系模型，不低于网状模型。
- (3) 层次数据模型提供了良好的完整性支持。

层次模型的主要缺点有：

- (1) 现实世界中很多联系是非层次性的，如多对多联系、一个结点具有多个双亲等，层次模型表示这类联系的方法很笨拙，只能通过引入冗

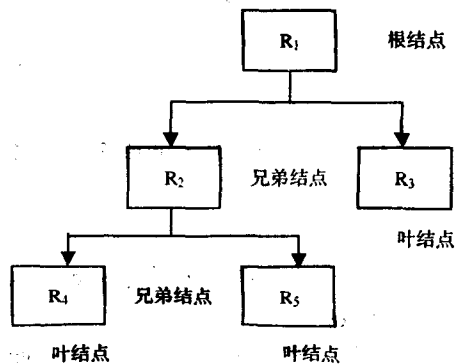


图 1.5 一个层次模型示例