

面向对象程序设计系列教材

Visual FoxPro

应用基础与面向对象程序设计教程 (第二版)

李雁翎 编

高等教育出版社

面向对象程序设计系列教材

Visual FoxPro
应用基础与面向对象程序设计教程
(第二版)

李雁翎 编

高等教育出版社

图书在版编目 (CIP) 数据

经济管理计算机基础教程. 下册/王其文主编. —北京: 高等教育出版社, 2000.3

ISBN 7-04-007499-0

I. 经… II. 王… III. 电子计算机-高等学校-教材 IV. TP3

中国版本图书馆 CIP 数据核字 (2000) 第 04278 号

经济管理计算机基础教程(下册)
王其文 主编

出版发行 高等教育出版社

社 址 北京市东城区沙滩后街 55 号

邮政编码 100009

电 话 010-64054588

传 真 010-64014048

网 址 <http://www.hep.edu.cn>

经 销 新华书店北京发行所

印 刷 北京地质印刷厂

开 本 787×1092 1/16

版 次 2000 年 3 月第 1 版

印 张 16.25

印 次 2000 年 3 月第 1 次印刷

字 数 390 000

定 价 20.00 元

凡购买高等教育出版社图书, 如有缺页、倒页、脱页等
质量问题, 请在所购图书销售部门联系

版权所有 侵权必究

第二版前言

《Visual Foxpro 应用基础与面向对象程序设计教程》第一版自 1999 年 9 月出版至今已有三年时间,得到了许多读者的厚爱,发行十余万册。在此期间作者收到了许多读者的信息反馈,对本书提出了许多宝贵的意见和建议,在此深表谢意。

为了适应数据库技术的新发展,根据教育部提出的非计算机专业计算机基础教学三个层次要求,并结合全国计算机等级考试“二级 Visual FoxPro”考试大纲,作者对第一版的部分内容进行了修订,同时增加了部分新的内容。新版教材在整体上基本保持了原书的体系和风格。

新版教材从关系数据库管理系统的基本原理出发,以应用系统开发知识为主线,讲解了 Visual FoxPro 的基本操作方法和操作命令,介绍了面向对象可视化编程、应用系统开发的方法和步骤,并通过大量实例讲述了数据库应用的基本概念,力图做到深入浅出,使读者掌握 Visual FoxPro 基本操作、面向对象编程及应用系统开发。

主要修订内容包括:

- (1) 调整了有关数据库系统基本原理的讲解,增加了关系代数及关系运算等内容。
- (2) 充实了 SQL 语言的内容,介绍了 SQL 语句的基本格式,SQL 语句的使用及如何利用 SQL 语言建立查询及创建视图等实例。
- (3) 将第十章的有关内容加以丰富,并增加相应的实例。
- (4) 修订了第十二章的内容,将以讲解控件属性为主的形式,改为以创建不同类型的表单为主的形式,介绍不同类型的表单设计,以实例介绍控件属性的使用方法、事件驱动程序设计等,特别是增加了 SQL 语句在表单中的应用。
- (5) 修订第十五章的小型系统开发案例,逐一介绍各功能模块的设计方法及系统开发的一般过程,使讲解更深入浅出。
- (6) 在部分章节中增添了数据库操作命令的介绍及使用方法。

为了配合新版教材的推出,作者还对高等教育出版社网站上的免费下载软件进行了更新,根据教材内容修改了原电子课件,并增加了书中所有的实例与系统开发案例,便于教师教学和学生自学使用。

本书主要作为高等学校相关专业数据库课程的教材,同时也可以作为参加计算机等级考试人员的培训或自学用书,也适合应用计算机进行各种数据管理和数据处理的人员使用。

在本书的编写及修订过程中,得到了中国人民大学工商学院周山芙教授,中国人民大学信息学院陈红教授的悉心帮助和指导,以及王连平教授、陈国军教授、陈光老师和高等教育出版社有关同志的大力支持和指导,在此一并表示感谢。

Visual FoxPro 技术日新月异,应用领域日益广泛,本书修订后希望继续得到广大读者的批评指正和帮助。

李雁翎

2002 年 5 月

1577/02

第一版前言

Microsoft Visual FoxPro 6.0 关系数据库系统是新一代数据库管理系统的杰出代表，以强大的性能、完整而又丰富的工具、超高速的速度、极其友好的界面以及完备的兼容性等特点，备受广大用户的欢迎。

Visual FoxPro 6.0 及其中文版，是可运行于 Windows 95 和 Windows NT 平台的 32 位数据库开发系统，它不仅简化数据库管理，而且能使应用程序的开发流程更为合理。Visual FoxPro 6.0 使组织数据、定义数据库规则和建立应用程序等工作变得简单易行。利用可视化的设计工具和向导，用户可以快速创建表单、查询和打印报表。

Visual FoxPro 6.0 还提供了一个集成化的系统开发环境，它不仅支持过程式编程技术，而且在语言方面作了强大的扩充，支持面向对象可视化编程技术，并拥有功能强大的可视化程序设计工具。目前，Visual FoxPro 6.0 是用户收集信息、查询数据、创建集成数据库系统、进行实用系统开发较为理想的工具软件。

本书由李雁翎编写，周山芙主审。在本书编写过程中，得到了陈光、芦华兵的大力支持和指导，在此一并感谢。

由于作者水平有限，难免有错误和不足之处，欢迎广大读者批评指正。

李雁翎
1999 年 8 月

目 录

上篇 Visual FoxPro 基础

第一章 数据库基础理论	3	2.5 Visual FoxPro 向导	35
1.1 信息、数据和数据处理	3	2.6 Visual FoxPro 生成器	36
1.1.1 信息与数据	3	2.7 Visual FoxPro 设计器	37
1.1.2 数据处理	4	思考题	37
1.2 数据模型	4	实验题	37
1.2.1 层次模型	5	第三章 Visual FoxPro 基础	38
1.2.2 网状模型	5	3.1 数据类型	38
1.2.3 关系模型	6	3.1.1 字符型	38
1.3 数据库系统	7	3.1.2 数值型	38
1.3.1 数据库	7	3.1.3 日期型	39
1.3.2 数据库管理系统	7	3.1.4 日期时间型	39
1.3.3 数据库系统的体系结构	8	3.1.5 逻辑型	39
1.3.4 数据库应用系统的构成	9	3.1.6 备注型	39
1.4 关系数据库及其设计	9	3.1.7 通用型	39
1.4.1 关系数据库	9	3.2 数据存储	40
1.4.2 关系的规范化	10	3.2.1 常量	40
1.4.3 表间关联关系的类型	14	3.2.2 内存变量	42
1.4.4 关系的完整性	16	3.2.3 数组变量	42
1.4.5 关系运算	18	3.2.4 字段变量	43
思考题	20	3.2.5 记录	44
第二章 Visual FoxPro 系统概述	21	3.2.6 对象	44
2.1 Visual FoxPro 的特性	21	3.2.7 内存变量的作用域	44
2.2 Visual FoxPro 的安装与启动	22	3.2.8 有关内存变量的操作	45
2.2.1 安装环境	22	3.3 函数	46
2.2.2 Visual FoxPro 的安装	23	3.4 表达式	51
2.2.3 启动 Visual FoxPro	25	3.4.1 算术表达式	51
2.2.4 退出 Visual FoxPro	27	3.4.2 字符表达式	51
2.3 Visual FoxPro 系统环境的配置	27	3.4.3 日期时间表达式	52
2.4 Visual FoxPro 用户界面	28	3.4.4 关系表达式	52
2.4.1 Visual FoxPro 的界面	28	3.4.5 逻辑表达式	53
2.4.2 标题栏	29	3.4.6 名表达式	53
2.4.3 菜单栏	30	3.4.7 类与对象操作符	54
2.4.4 工具栏	33	思考题	54
2.4.5 命令窗口	35	实验题	54
2.4.6 工作区与状态行	35	第四章 表操作	55

4.1 建立表	55	6.2 数据库中的表的使用	113
4.1.1 定义表中的数据类型	55	6.2.1 设置表中字段的显示标题	113
4.1.2 利用表设计器创建表	56	6.2.2 设置表中字段的注释	115
4.1.3 表的打开与关闭	59	6.2.3 设置表中字段的默认值	116
4.1.4 利用表设计器修改表结构	60	6.2.4 设置表中字段的有效规则	117
4.1.5 利用表向导创建表	63	6.3 建立表间的关联关系	119
4.2 向表中加入记录	67	6.3.1 建立表间的一对一关系	119
4.2.1 创建表时立即输入数据	67	6.3.2 建立表间的一对多关系	119
4.2.2 以追加方式输入数据	67	6.3.3 设置参照完整性	121
4.2.3 备注型数据的输入	69	思考题	124
4.2.4 通用型数据的输入	69	实验题	124
4.3 数据的显示和修改方式	73	第七章 视图与查询	125
4.3.1 用编辑方式显示和修改数据	73	7.1 创建视图	125
4.3.2 用浏览方式显示和修改数据	74	7.1.1 利用视图设计器创建视图	125
4.3.3 用 Browse 命令和 List 命令		7.1.2 利用视图向导创建视图	131
浏览、修改数据	76	7.2 利用视图更新数据	136
4.4 控制字段和记录的访问	78	7.3 创建查询	139
4.4.1 限制对字段的访问	78	7.3.1 用查询设计器创建查询	139
4.4.2 限制对记录的访问	80	7.3.2 利用查询向导创建查询	142
4.5 数据表中记录的定位	81	7.4 使用查询	147
4.6 删除和恢复表中的数据	87	思考题	151
4.6.1 逻辑删除表中的记录	87	实验题	152
4.6.2 恢复表中逻辑删除的记录	88	第八章 SQL 关系数据库查询语言	153
4.6.3 物理删除表中的记录	88	8.1 创建查询 SQL 语句	153
4.7 快速修改表中的数据	94	8.2 表定义 SQL 语句	161
思考题	97	8.2.1 创建表	161
实验题	97	8.2.2 修改字段属性	162
第五章 索引及创建索引	98	8.2.3 删除字段	163
5.1 什么是索引	98	8.2.4 增加字段	164
5.2 索引类型	98	8.3 表记录操作 SQL 语句	164
5.3 创建索引	99	8.3.1 插入记录	164
5.4 索引的使用	103	8.3.2 更新记录	165
5.4.1 按索引字段排序	103	8.3.3 删除记录	166
5.4.2 按索引表达式排序	106	思考题	167
思考题	109	实验题	167
实验题	109	第九章 创建项目	168
第六章 创建数据库	110	9.1 创建项目	168
6.1 创建数据库	110	9.2 项目管理器的使用	169
6.1.1 建立数据库	110	9.2.1 用“数据”选项卡管理数据	169
6.1.2 打开数据库	111	9.2.2 用“文档”选项卡管理文档	171
6.1.3 向数据库添加数据表	112		

9.2.3 用“类”选项卡管理类·····	172	9.2.6 用“项目”菜单管理项目 中的文件·····	174
9.2.4 用“代码”选项卡管理 代码文件·····	173	9.3 定制项目管理器·····	174
9.2.5 用“其他”选项卡管理 其他类型的文件·····	173	思考题·····	176
		实验题·····	176

下篇 Visual FoxPro 面向对象可视化编程

第十章 程序设计基础 ·····	142	第十一章 面向对象可视化编程基础 ·····	203
10.1 Visual FoxPro 的工作方式·····	142	11.1 新概念的引入·····	203
10.1.1 命令方式·····	142	11.1.1 对象·····	203
10.1.2 菜单方式·····	142	11.1.2 类·····	204
10.1.3 程序文件方式·····	180	11.1.3 属性·····	206
10.2 程序文件的建立与编辑·····	181	11.1.4 事件与方法·····	206
10.2.1 以命令方式建立、编辑 程序文件·····	181	11.2 类的创建·····	207
10.2.2 以菜单方式建立、编辑 程序文件·····	182	11.2.1 创建类的一般方法·····	207
10.2.3 利用设计工具建立程序代码·····	183	11.2.2 用菜单方式创建类·····	208
10.3 程序文件的调用·····	183	11.2.3 通过编程定义类·····	211
10.3.1 以命令方式调用程序文件·····	183	11.3 对象的操作·····	212
10.3.2 以菜单方式调用程序文件·····	183	11.3.1 由类创建对象·····	213
10.4 程序的基本结构·····	183	11.3.2 设置对象的属性·····	213
10.5 常用的交互输入语句·····	184	11.3.3 调用对象的方法和 触发对象的事件·····	214
10.6 分支结构·····	187	11.4 对象设计实例·····	214
10.6.1 单向分支·····	187	11.4.1 用编程方式设计对象·····	215
10.6.2 双向分支·····	188	11.4.2 用面向对象方式设计对象·····	216
10.6.3 多向分支·····	189	思考题·····	219
10.6.4 使用分支语句应注意的几点·····	190	实验题·····	219
10.7 循环结构·····	190	第十二章 表单设计及运行 ·····	220
10.7.1 “当”型循环控制语句·····	191	12.1 创建表单·····	220
10.7.2 “计数”型循环控制语句·····	192	12.1.1 用表单设计器创建表单·····	220
10.7.3 “指针”型循环控制语句·····	193	12.1.2 用表单向导创建表单·····	221
10.7.4 使用循环语句应注意的几点·····	193	12.2 修改表单·····	227
10.8 过程与过程文件·····	194	12.3 运行表单·····	229
10.9 程序设计实例·····	195	12.4 表单控件·····	230
10.10 程序设计的一般问题·····	198	12.4.1 添加表单控件·····	230
10.10.1 命令和语句的使用效率·····	199	12.4.2 表单控件的属性·····	231
10.10.2 减少和避免程序中的语法错误·····	199	思考题·····	237
10.10.3 提高程序的可维护性·····	200	实验题·····	237
10.11 程序调试工具·····	200	第十三章 实用表单的设计 ·····	238
思考题·····	202	13.1 设计“无控件”表单·····	238
实验题·····	202		

13.2 设计“说明”表单	239	15.4 生成菜单程序	291
13.3 设计“系统登录”表单	248	15.5 运行菜单	291
13.4 设计“数据维护”表单	251	思考题	293
13.5 设计“数据浏览”表单	258	实验题	293
13.6 设计“数据查询”表单	264	第十六章 小型系统开发实例	294
思考题	275	16.1 系统开发的一般过程	294
实验题	275	16.2 系统总体规划	294
第十四章 创建报表	276	16.3 主要功能模块	295
14.1 报表布局	276	16.3.1 数据库及表	295
14.1.1 设置报表页面与布局	276	16.3.2 主程序设计	297
14.1.2 设置报表中数据的显示格式	277	16.3.3 主要工作窗口	298
14.1.3 添加报表控件	277	16.3.4 系统工具条	302
14.1.4 常用的报表布局	277	16.3.5 系统主菜单设计	303
14.2 创建报表	279	16.4 系统部件的组装	304
14.2.1 利用报表设计器创建报表	279	16.5 运行“专门人才信息管理系统”	307
14.2.2 利用报表向导创建报表	281	思考题	310
14.3 修改报表	284	实验题	310
14.4 使用报表	286	附录	311
思考题	286	附录一 数据库操作命令一览表	311
实验题	286	附录二 控件类名称及功能	321
第十五章 菜单设计	287	附录三 对象名称及功能	322
15.1 菜单系统	287	附录四 属性语法与功能	322
15.2 建立菜单系统	288	附录五 事件语法与功能	330
15.2.1 创建主菜单	288	附录六 方法语法与功能	332
15.2.2 创建子菜单	289	附录七 系统内存变量一览表	333
15.3 指定菜单项任务	290	参考文献	335

上 篇

Visual FoxPro 基础

第一章 数据库基础理论

计算机作为具有极高的运算速度、巨大的数据储存能力，以及准确地进行各种算术运算和逻辑运算的现代化计算工具，已被广泛地应用于社会各个领域。计算机技术的高速发展被认为是人类进入信息时代的标志。在信息时代，人们对大批量的信息进行加工处理，在这一过程中应用数据库技术，一方面促进了计算机技术的高度发展，另一方面也形成了专门的信息处理理论及数据库管理系统。从某种意义上说，数据库管理系统软件正是计算机技术和信息时代相结合的产物，它是信息处理或数据处理的核心，是研究数据共享的一门科学，同时，也是计算机科学的一个重要分支。

本章将从数据库基本概念出发，逐一讲解信息、数据、数据处理、数据模型、数据库、数据库设计等基础知识和概念。这些是学习和掌握 Visual FoxPro 技术的基础和前提。

1.1 信息、数据和数据处理

在数据处理这一计算机应用领域，人们首先遇到的基本概念是信息和数据，它们是两个不同的术语，却有着密不可分的联系。

1.1.1 信息与数据

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。数据所反映的事物属性是它的内容，而符号是它的形式。

数据与信息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。

用数据符号表示信息，其形式通常有三种：数值型数据，即对客观事物进行定量记录的符号，如体重、年龄、价格的多少等；字符型数据，即对客观事物进行定性记录的符号，如姓名、单位、地址的标志等；特殊型数据，如声音、视频、图像等。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，而数据如不具有知识性和有用性，则不能称其为信息。

1.1.2 数据处理

数据处理也称为信息处理。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处理。它包括对数据的采集、整理、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份，作为行为和决策的依据。

随着电子计算机软件和硬件技术的发展，数据处理过程发生了划时代的变革，而数据库技术的发展，又使数据处理跨入了一个崭新的阶段。数据管理技术的发展大致经历了从人工管理方式、文件管理方式到数据库系统管理方式三个阶段。

人工管理阶段出现在计算机应用于数据管理的初期。由于没有必要的软件、硬件环境的支持，用户只能直接在裸机上操作。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则随之改变，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，编程效率较差。

文件管理阶段即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，由一个专门的文件管理系统实施统一管理。文件管理系统是一个独立的系统软件，它是应用程序与数据文件之间的一个接口。在这一管理方式下，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。应用程序与数据文件之间具有一定的独立性，因此比手工管理方式前进了一步。但是，数据文件仍高度依赖于其对应的程序，不能被多个程序所共享。由于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。

数据库系统管理阶段即对所有的数据实行统一规划管理，形成一个数据中心，构成一个数据“仓库”。数据库中的数据能够满足所有用户的不同要求，供不同用户共享。在这一管理方式下，应用程序不再只与一个孤立的数据文件相对应，可以取整体数据集的某个子集作为逻辑文件与其对应，通过数据库管理系统实现逻辑文件与物理数据之间的映射。在数据库系统管理的系统环境下，应用程序对数据的管理和访问灵活方便，而且数据与应用程序之间完全独立，使程序的编制质量和效率都有所提高；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。

1.2 数据模型

现实世界中的客观事物是彼此相互联系的。一方面，某一事物内部的诸因素和诸属性根据一定的组织原则相互具有联系，构成一个相对独立的系统；另一方面，某一事物同时也作为一个更大系统的一个因素或一种属性而存在，并与系统的其他因素或属性发生联系。客观事物的这种普遍联系性，决定了作为事物属性记录符号的数据与数据之间也存在着一定的联系性。具有联系性的相关数据总是按照一定的组织关系排列，从而构成一定的结构，对这种结构的描述就是数据模型。

从理论上讲，数据模型是指反映客观事物及客观事物间联系的数据组织的结构和形式。客观事物是千变万化的，表现各种客观事物的数据结构和形式也是千差万别的，尽管如此，它们之间还是有其共同性的。常用的数据模型有如下三种。

1.2.1 层次模型

层次模型（Hierarchical Model）表示数据间的从属关系结构，是一种以记录某一事物的类型为根结点的有向树结构。层次模型像一棵倒置的树，根结点在上，层次最高；子结点在下，逐层排列。其主要特征如下：

- 仅有一个无双亲的根结点。
- 根结点以外的子结点，向上仅有一个父结点，向下有若干子结点。

层次模型表示的是从根结点到子结点的一个结点对多个结点，或从子结点到父结点的多个结点对一个结点的数据间的联系。层次模型的示例如图 1-1 所示。

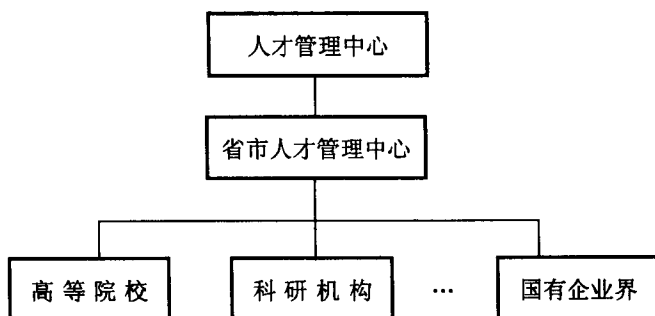


图 1-1 层次模型示例

1.2.2 网状模型

网状模型（Network Model）是层次模型的扩展，它表示多个从属关系的层次结构，呈现一种交叉关系的网络结构。网状模型是以记录为结点的网络结构。其主要特征如下：

- 有一个以上的结点无双亲。
- 至少有一个结点有多个双亲。

网状模型可以表示较复杂的数据结构，即可以表示数据间的纵向关系与横向关系。这种数据模型在概念上、结构上都比较复杂，操作上也有很多不便。网状模型的示例如图 1-2 所示。

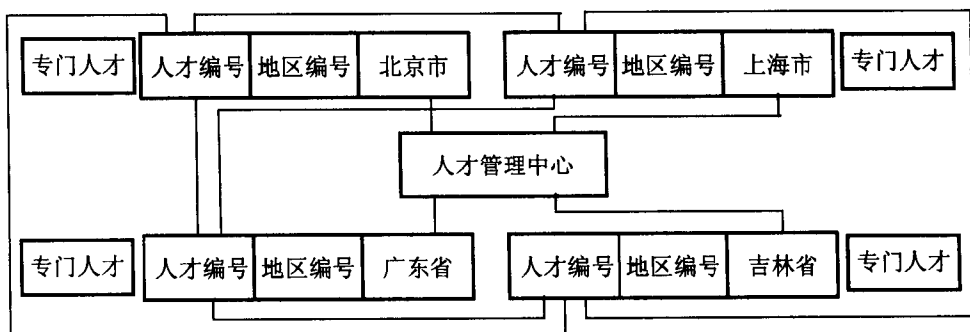


图 1-2 网状模型示例

1.2.3 关系模型

关系模型 (Relational Model) 的所谓“关系”是有特定含义的。广义地说, 任何数据模型都描述一定事物数据之间的关系。层次模型描述数据之间的从属层次关系; 网状模型描述数据之间的多种从属的网状关系。而关系模型的所谓“关系”虽然也适用于这种广义的理解, 但同时又特指那种虽具有相关性而非从属性的平行的数据之间按照某种序列排列的集合关系。

例如, 有数据记录如下: 甲, 女, 40 岁; 乙, 男, 51 岁; 丙, 女, 35 岁; 丁, 男, 45 岁。这四组数据之间是平行的, 从层次从属角度看也是无关系的, 但假如我们知道他们是同一个部门的工作人员, 就可以建立一个关系 (一张二维表), 如表 1-1 所示。

表 1-1 某部门专门人才基本情况表

姓名	性别	年龄
甲	女	40
乙	男	51
丙	男	35
丁	女	45

表中的这些数据虽然是平行的, 不代表从属关系, 但它们构成了某部门工作人员的属性关系结构。同样, 也可以假设上述四组数据不是来自同一个部门, 他们所从事的专业也不同, 由此便构成了某地区专门人才基本情况表, 又可以建立一个关系 (一张二维表), 如表 1-2 所示。

表 1-2 某地区专门人才基本情况表

部门	专业	姓名	性别	年龄
高等院校	信息管理	甲	女	40
研究所	软件工程	乙	男	51
销售公司	计算机	丙	男	35
经贸公司	国际贸易	丁	女	45

以上两表其实就是关系模型。表格中的每一数据都可看成独立的数据项, 它们共同构成了该关系的全部内容。

表格中的每一横行称为一个元组 (Tuple)。用来表示现实世界同一实体型中若干平行的、相对独立的个体事物, 每一元组由若干数据分量 (Component) 组成, 横向排列该事物的诸种属性。如: (高等院校, 信息管理, 甲, 女, 40) 5 个分量数据项描述了人才甲的一些信息。

表格中的每一竖列称为一个属性 (Attribute)。属性表示关系模型中某竖列全部数据项 (属性) 的类型, 每一属性由若干按照某种值域 (Domain) 划分的相同类型的分量组成, 竖向列出其诸种类型 (属性), 每个属性都有一个名字, 称为属性名。

从总体上说, 以属性分类的若干个元组的集合, 构成关系模型中的一个关系 (Relation)。在某种意义上可以说, 关系模型就是一张二维表, 用来描述客观事物以及不同事物间的联系。

关系模型的主要特点有:

- (1) 关系中每一分量不可再分, 是最基本的数据单位;
- (2) 每一竖列的分量是同属性的, 列数根据需要而设, 且各列的顺序是任意的;
- (3) 每一横行由一个个体事物的诸多属性构成, 且各行的顺序可以是任意的;
- (4) 一个关系是一张二维表, 不允许有相同的属性名, 也不允许有相同的元组。

1.3 数据库系统

数据库系统是以数据库为核心的信息处理系统, 它既能集中地进行数据管理, 也能够开发、维护应用系统程序。

1.3.1 数据库

数据库 (Data Base) 是数据库系统的核心和管理对象。

前面介绍的数据模型是对数据库如何组织的一种模型表示, 它不仅包括客观事物本身的信息, 还包括各事物间的联系。数据模型的主要特征是其数据结构, 因此数据模型的确定, 就等于确定了数据间的关系, 即数据库的框架。有了数据间的关系框架, 再把表示客观事物具体特征的数据装入框架中, 就形成了数据库。

所谓数据库, 就是以一定的组织方式将相关的数据组织在一起, 存放在计算机外存储器上形成的, 能为多个用户共享的, 与应用程序彼此独立的一组相关数据的集合。

数据库的性质是由数据模型决定的。在数据库中数据的组织结构如果满足层次模型的特性, 则该数据库为层次数据库; 数据的组织结构如果满足网络模型的特性, 则该数据库为网络数据库; 数据的组织结构如果满足关系模型的特性, 则该数据库为关系数据库。

1.3.2 数据库管理系统

从信息处理的理论角度讲, 如果把利用数据库进行信息处理的工作过程, 或把掌握、管理和操纵数据库的数据资源的方法看作是一个系统的话, 则称这个系统为数据库管理系统 (Data Base Management System, 简称 DBMS)。

数据库管理系统提供对数据库资源进行统一管理和控制的功能, 使数据与应用程序隔离, 数据具有独立性; 使数据结构及数据存储具有一定的规范性, 减少了数据的冗余, 并有利于数据共享; 提供安全性和保密性措施, 使数据不被破坏、不被窃用; 提供并发控制, 在多用户共享数据时保证数据库的一致性; 提供恢复机制, 当出现故障时, 数据恢复到一致性状态。

数据库管理系统通常由三个部分组成: 数据描述语言 (DDL) 及其编译程序、数据操纵语言 (DML) 或查询语言及其编译或解释程序、数据库管理例程序。

数据描述语言用于定义数据库的各级模式 (外模式、概念模式、内模式), 各种模式通过数据描述语言编译器翻译成相应的目标模式, 保存在数据字典中。

数据操纵语言提供对数据库数据存取、检索、插入、修改和删除等的基本操作。数据操纵语言一般有两种类型: 一种是嵌入 COBOL、FORTRAN、C 等高级语言中, 不独立使用, 此类语言称为宿主型语言; 另一种是交互查询语言, 可以独立使用进行简单的检索、更新等

操作，通常由一组命令组成，用于提取数据库中的数据，此类语言称为自主型语言。

数据库管理例程序是数据库管理系统的核心部分，它包括并发控制、存取控制、完整性条件检查与执行、数据库内部维护等，数据库的所有操作都在这些控制程序的统一管理下进行，以确保数据的正确有效。

1.3.3 数据库系统的体系结构

数据库系统的体系结构是数据库系统的一个总的框架。尽管实际的数据库系统的软件产品多种多样（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），但绝大多数的数据库系统在总的体系结构上都具有三级模式结构的结构特征。

从数据管理的角度来看，与数据库打交道的有三类人员：用户、应用程序员和系统管理员。由于他们对数据库的认识、理解和接触范围各不相同，从而形成了各自的数据库视图。所谓视图是指观察、认识和理解数据的范围、角度和方法。根据各类人员与数据库的不同关系，可把视图分为三种：即对应于用户的外部视图、对应于应用程序员的概念视图和对应于系统管理员的内部视图。由此形成数据库系统的三级模式结构：外模式、概念模式和内模式。数据库系统的体系结构如图 1-3 所示。

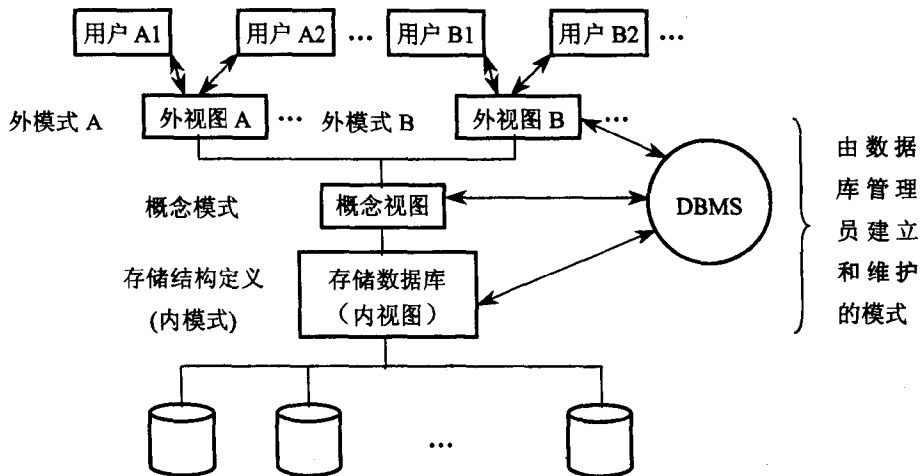


图 1-3 数据库系统结构图

其中，外模式又称子模式或用户模式，对应于用户级，是某个或几个数据库用户所看到的数据库的数据视图。外模式是从概念模式导出的子模式，用户可以通过子模式描述语言来描述用户级数据库的记录，还可以利用数据操纵语言对这些记录进行操作。

概念模式又称逻辑模式，对应于概念级，它是由数据库设计者综合所有用户的数据，按照统一的观点构造的全局逻辑结构，是对数据库中全部数据的逻辑结构和特征的总体描述，是所有用户的公共数据视图。它是用模式描述语言来描述的，是应用程序员所看到的数据库，即应用程序员的数据视图。

内模式又称存储模式，对应于物理级，是全体数据库数据的内部表示或底层描述，它描述了数据在存储介质上的存储方式与物理结构。