

计 / 算 / 机 / 应 / 用 / 基 / 础

(根据教育部对计算机基础教学的三个层次的要求编写)

Visual FoxPro

程序设计教程

沈琴婉 崔宝深 王娟 编著

南开大学出版社

计算机应用基础

(根据教育部对计算机基础教学的三个层次的要求编写)

Visual FoxPro

程序设计教程

沈琴婉 崔宝深 王 娟 编著

南开大学出版社

天津

图书在版编目(CIP)数据

Visual FoxPro 程序设计教程/沈琴婉编著. —天津:
南开大学出版社, 2002. 10
ISBN 7-310-01773-0

I. V... II. 沈... III. 关系数据库—数据库管理系
统, Visual FoxPro 6.0—高等学校—教材
IV. TP311.138

中国版本图书馆 CIP 数据核字(2002)第 058158 号

出版发行 南开大学出版社
地址:天津市南开区卫津路 94 号 邮编:300071
营销部电话:(022)23508339 23500755
营销部传真:(022)23508542
邮购部电话:(022)23502200

出版人 肖占鹏
承印 河北昌黎人民胶印厂印刷
经销 全国各地新华书店
版次 2002 年 10 月第 1 版
印次 2002 年 10 月第 1 次印刷
开本 787mm×1092mm 1/16
印张 22.75
字数 597 千字
印数 1—5000
定价 32.00 元

内 容 提 要

本教程是根据教育部对非计算机专业的学生提出的计算机基础教学三次（即计算机文化基础、技术基础和应用基础）的要求，由多年从事计算机基础教学的教师编写的。

本书以中文 Visual FoxPro 6.0 为开发平台。全书共分十三章，内容包括：数据库系统基础知识、Visual FoxPro 数据库基础、项目管理器、数据库的创建、数据表的操作、记录的检索、视图、报表设计、程序设计基础、面向对象的程序设计、表单设计、菜单设计和一个应用系统开发实例。该书每一章后面都有大量的习题，读者可以练习，并从实践中学习掌握。

在编写中特别注意遵循由浅入深、循序渐进、繁简适当的原则。全书采用图文并茂的形式，结合大量实例，深入浅出地讲解了面向对象编程的基本概念和方法，使读者逐步掌握中文 Visual FoxPro 6.0 的基本操作和面向对象编程技术。

本书不仅适合高等院校、职大职专等非计算机专业的学生使用，同时，对正在学习 Visual FoxPro 程序设计的读者也是一本颇有价值的教材。

前 言

计算机知识更新的速度之快令人目不暇接,新理论、新技术、新工具更是层出不穷。50年来,计算机的变化虽然很快,但有规律可循。有些基础知识在常用的计算机中基本没有改变,发生较大变化的是计算机的应用模式和人机交互模式。所以,我们一方面应注意让学生了解并掌握当前主流软件的应用,增加网络和有关信息处理的知识,使学生了解信息化社会对大学生的要求;另一方面,让学生掌握比较成熟的、稳定的、适用面广的理论和应用知识,在毕业以后,能够尽快适应信息化社会的需求,又有一定的后劲适应信息化社会的突飞猛进的发展。因为对当代大学生来说,只知道用还不够,更新教学内容应着眼于加强基础。程序设计代表了计算机文化的基本特征,除了结构化程序设计以外,要引进面向对象程序设计的概念和方法等新内容。

中文 Visual FoxPro 6.0 是 Microsoft 公司推出的最新可视化关系数据库开发平台,是新一代微机版数据库管理系统的突出代表。其特点是功能强大、处理速度快,具有完整而丰富的工具和良好的用户界面,并且兼容了 FoxBASE 和 FoxPro,因此本教材以之作为讲授平台。

本教材对程序设计的重点和难点,从不同角度、不同层次由浅入深地进行讲解,使读者通过阅读典型例题受到一定的启发,深入而全面地理解结构化程序设计和面向对象程序设计的基本概念,并通过独立完成一定数量的习题,牢固掌握基础知识并能将其灵活运用,独立完成一个数据库应用系统的设计,使学生增强创新意识,提高应用能力。

本书第1章、第2章、第3章、第4章、第8章及附录由崔宝深老师编写,第5章、第6章、第7章由王娟老师编写,第9章、第10章、第11章、第12章和第13章由沈琴婉老师编写。在编写过程中得到南开大学出版社李正明等同志的热情支持和指导,提出了许多很好的意见,作者在此致以衷心的感谢。书中错误与不妥之处敬请读者批评指正。

作 者
2002 年 5 月

E-mail 地址: shenqinwan@eyou.com

目 录

第 1 章 数据库系统基础知识

1.1 数据库的基本概念	1
1.1.1 数据、信息与数据库	1
1.1.2 数据管理技术的发展	2
1.1.3 数据库系统的组成及特点	3
1.1.4 基本的数据模型	4
1.2 数据库管理系统	6
1.3 关系型数据库系统	7
1.3.1 关系术语	7
1.3.2 关系的特点	8
1.3.3 关系的设计	8
1.3.4 关系运算	10
习题 1	11

第 2 章 Visual FoxPro 数据库管理系统

2.1 Visual FoxPro 概述	13
2.1.1 Visual FoxPro 的特点	13
2.1.2 Visual FoxPro 的运行环境和安装	15
2.1.3 Visual FoxPro 的启动与退出	15
2.2 Visual FoxPro 的界面风格	16
2.2.1 Visual FoxPro 的用户界面	16
2.2.2 菜单结构及其操作	17
2.2.3 工具栏的使用	20
2.2.4 对话框操作	21
2.2.5 Visual FoxPro 的工作方式	22
2.3 Visual FoxPro 的常量、变量和数组	23
2.3.1 常量	23
2.3.2 变量和数组	24
2.4 Visual FoxPro 的表达式和函数	25
2.4.1 表达式	25
2.4.2 函数	28
2.5 Visual FoxPro 的命令格式及文件类型	33
2.5.1 Visual FoxPro 的命令格式	33
2.5.2 Visual FoxPro 的文件类型	35

习题 2	36
------------	----

第 3 章 Visual FoxPro 的辅助设计工具和项目管理器

3.1 Visual FoxPro 的辅助设计工具	39
3.1.1 向导	39
3.1.2 设计器	41
3.1.3 生成器	42
3.2 选项对话框的使用	43
3.3 项目管理器	46
3.3.1 启动项目管理器	46
3.3.2 项目管理器的功能	47
3.3.3 项目管理器的界面操作	47
3.3.4 项目管理器的使用	48
习题 3	51

第 4 章 创建数据库

4.1 数据库设计	53
4.2 创建自由表	56
4.2.1 设计表的结构	56
4.2.2 建立表的结构	58
4.2.3 数据的输入与修改	60
4.3 创建数据库	61
4.3.1 创建数据库的两种方式	61
4.3.2 向数据库中添加或移去表	62
4.4 有效性规则与参照完整性	63
4.4.1 长表名、长字段名与注释	63
4.4.2 设置字段属性	64
4.4.3 设置记录规则	66
4.4.4 主索引与永久关系	67
4.4.5 参照完整性	70
4.4.6 创建存储过程	72
4.5 表的索引	73
4.5.1 索引的类型	73
4.5.2 创建索引	74
4.5.3 索引的使用	76
习题 4	78

第 5 章 数据表的维护与浏览

5.1 数据表结构的维护	81
5.1.1 菜单交互方式修改表结构	81

5.1.2 命令方式修改表结构	83
5.1.3 命令方式复制表结构	83
5.2 表中数据的维护	83
5.2.1 记录指针的定位	84
5.2.2 在表中添加记录	86
5.2.3 在表中修改记录	91
5.2.4 删除或还原表中的记录	93
5.2.5 表中数据的备份	95
5.3 数据表的浏览窗口	96
5.3.1 改变数据表浏览窗口的大小	96
5.3.2 改变列宽和行高	96
5.3.3 移动字段显示的位置	97
5.3.4 打开和关闭网格线	97
5.3.5 拆分“浏览”窗口	98
5.3.6 限定数据表的显示内容	99
习题 5	101

第 6 章 记录的检索

6.1 查看特定的记录	105
6.1.1 使用指针定位命令查看记录	105
6.1.2 按照索引顺序定位检索	107
6.1.3 使用筛选进行检索	109
6.2 使用查询工具检索记录	110
6.2.1 使用查询向导创建简单的查询	111
6.2.2 利用查询设计器创建查询	115
6.2.3 建立复杂的条件查询	127
6.2.4 建立交叉表查询	130
6.3 常用的 SQL 查询语句介绍	133
6.3.1 简单查询	134
6.3.2 嵌套查询	138
6.3.3 连接查询	140
6.3.4 分组及使用组函数查询	141
习题 6	143

第 7 章 视图

7.1 视图的功能以及与查询的区别	146
7.2 使用视图向导创建视图	147
7.3 使用视图设计器建立视图	152
7.4 利用视图更新数据	157
7.4.1 指定可更新的字段	157

7.4.2 设置更新方法	159
7.4.3 检查更新冲突	159
习题 7	160

第 8 章 报表和标签的设计

8.1 创建报表	161
8.1.1 报表布局	161
8.1.2 使用报表向导创建报表	161
8.1.3 使用“快速报表”创建报表	165
8.1.4 使用“报表设计器”创建和修改报表	166
8.2 创建分组报表	172
8.2.1 设置报表的记录顺序	172
8.2.2 设计单级分组报表	173
8.2.3 设计多级数据分组报表	175
8.3 创建一对多报表	177
8.4 设计多栏报表	180
8.5 创建标签	182
8.6 报表和标签的打印	183
习题 8	186

第 9 章 Visual FoxPro 程序设计基础

9.1 程序文件	188
9.1.1 程序文件的建立与执行	189
9.1.2 不同工作方式下的环境设置	190
9.1.3 输入输出命令	191
9.2 程序的基本结构	194
9.2.1 顺序结构	194
9.2.2 选择结构	195
9.2.3 循环结构	199
9.3 过程与过程调用	209
9.3.1 子程序、函数与过程	209
9.3.2 内存变量和数组的作用范围	214
9.3.3 主程序与子程序或过程之间的参数传递	217
9.3.4 结构化程序设计	218
9.4 应用程序的编译和调试	219
9.4.1 常见的错误类型	219
9.4.2 常用的调试技术	220
9.4.3 使用“调试器”进行调试	220
习题 9	224

第 10 章 面向对象的程序设计

10.1 基本概念	232
10.1.1 对象与类	233
10.1.2 类的特征和基类	236
10.1.3 事件	238
10.2 定义类的程序方式	241
10.2.1 定义类的命令	241
10.2.2 向容器类中添加对象	243
10.2.3 指定方法和事件程序	245
10.2.4 保护和隐藏类成员	247
10.2.5 按类层次调用方法或事件代码	248
10.3 利用类设计器定义类	251
10.4 创建和使用类库	254
10.4.1 创建类库的两种方法	254
10.4.2 修改库中的类定义	255
10.4.3 使用类库的命令	255
习题 10	256

第 11 章 表单设计

11.1 表单设计概述	263
11.1.1 使用表单向导创建新表单	263
11.1.2 使用“快速表单”命令创建新表单	268
11.1.3 表单设计器简介	269
11.1.4 保存和运行表单	272
11.1.5 表单的参数传递	274
11.2 设置数据环境	275
11.3 基本控件简介	278
11.3.1 控件使用要点	278
11.3.2 设计直线 (Line) 控件	279
11.3.3 设计形状 (Shape) 控件	279
11.3.4 设计标签 (Label) 控件	280
11.3.5 设计文本框 (Text Box) 与编辑框 (Edit Box) 控件	280
11.3.6 设计命令按钮 (Command Button) 控件	283
11.3.7 设计命令按钮组 (Command Group) 控件	285
11.3.8 设计复选框 (Check Box) 与选项按钮组 (Option Group) 控件	287
11.3.9 设计列表框 (List Box) 与组合框 (Combo Box) 控件	288
11.3.10 设计微调 (Spinner) 控件	291
11.3.11 设计计时器 (Timer) 控件	291
11.3.12 设计页框 (Page Frame) 控件	292

11.3.13 设计容器 (Container) 控件	293
11.3.14 设计表格 (Grid) 控件	294
11.3.15 设计图像 (Image) 控件	296
11.3.16 设计 ActiveX 控件与 ActiveX 绑定控件	297
习题 11	299

第 12 章 菜单设计

12.1 创建菜单系统	303
12.1.1 建立主菜单栏	303
12.1.2 创建菜单项或子菜单	305
12.2 预览及执行菜单	309
12.3 菜单系统的初始化代码和清理代码	310
12.4 为菜单系统插入系统菜单项	310
习题 12	313

第 13 章 一个应用系统开发实例

13.1 设计一个应用系统的具体实现过程	314
13.1.1 规划应用程序	314
13.1.2 创建数据库	314
13.1.3 定义和引用类	315
13.1.4 精心设计用户界面	315
13.1.5 提供具有交互能力的输出信息	315
13.1.6 测试和调试	315
13.2 一个数据库应用系统实例	315
习题 13	335
附录 Visual FoxPro 6.0 命令一览表	339

第1章 数据库系统基础知识

随着社会信息化进程的加快, 计算机应用已经渗透到各行各业的管理工作之中。以数据库系统为核心的办公自动化系统、管理信息系统、决策支持系统等得到广泛应用, 数据库技术和计算机网络技术相互渗透、相互促进, 已成为当前计算机理论和应用中发展极为迅速、应用非常广泛的两大领域。

目前, 数据库技术的应用范围已不仅仅是事务管理, 而是扩大到专家系统、情报检索、人工智能和计算机辅助设计等非数值计算的各个方面。作为计算机应用人员, 只有掌握数据库系统的基础知识, 熟悉数据库管理系统的特点, 才能开发出适用的、水平较高的数据库应用系统, 为促进我国经济发展和人类社会的进步做出积极的贡献。

在学习 Visual FoxPro 程序设计之前, 首先要建立一些有关数据库系统的基本概念, 了解数据库技术的基础知识, 为学习其余各章打下坚实基础。

1.1 数据库的基本概念

1.1.1 数据、信息和数据库

数据(Data)在一般意义上被认为是对客观事物特征所进行的一种抽象化、符号化表示。例如, 某人出生日期是1986年6月28日, 身高1.72m, 体重66kg, 其中1986年、6月、28日、1.72m、66kg等都是数据, 它们描述了该人的某些特征。数据可以有不同的形式。例如, 出生日期也可以表示成“1986.6.28”“{06/28/86}”等形式。需要明确的是, 这里数据的概念比在科学计算领域中已经大大地拓宽了, 不仅包括数字、字母、汉字及其他特殊字符组成的文本形式的数据, 而且还包括图形、图像、声音等多媒体数据。总之, 凡是能够被计算机处理的对象都称为数据。

信息(Information)通常被认为是有一定含义的、经过加工处理的、对决策有价值的数。请看一个简单例子: 某排球队中, 每个队员的身高数据为1.85m, 1.97m, 1.86m, ..., 经过计算得到平均身高为1.88m, 这便是该排球队的一条重要信息。又如, 今年入学的所有新生中, 每个人的出生日期为原始数据, 用今年年份减去出生日期中的年份, 得到每个人的年龄(可视为二次数据), 再由每个人的年龄求出平均年龄, 即得到有用的信息, 它反映出新生整体的年龄状况。数据与信息之间的关系可以表示为:

信息 = 数据 + 处理

其中, 处理是指将数据转换成为信息的过程, 包括数据的收集、存储、加工、排序、检

索等一系列活动。数据处理的目的是从大量的现有数据中,提取对人们有用的信息,作为决策的依据。可见,信息与数据是密切相关的。数据是信息的载体,它表示了信息;信息是数据的内涵,即数据的语义解释。信息是有价值的,其价值取决于它的准确性、及时性、完整性和可靠性。为了提高信息的价值,就必须用科学的方法来管理信息,这种方法就是数据库技术。

数据库(DataBase,简称DB)是指存储在计算机存储设备上、结构化的相关数据的集合。请注意,这些数据是以二进制形式存储在磁盘、光盘或半导体存储器等存储介质上的。那么,它们又是如何存储的呢?众所周知,图书馆书库中的图书是按一定规则(即藏书模型)分门别类整齐地排列在书架上的,读者查阅起来十分方便。试想,如果数以百万计的图书杂乱无章地堆放在一起,要从中找出一本所需要的书,那简直如同大海捞针!同理,为了便于检索和使用数据,数据库中的大量数据也必须按照一定的规则(即数据模型)来存放,这就是所说的“结构化”。此外,存储在数据库中的数据彼此之间是有一定联系的,而不是毫不相干的。可见,数据库不仅包括描述事物的数据,而且反映了相关事物之间的联系。

1.1.2 数据管理技术的发展

计算机数据管理技术随着计算机硬件和软件技术的发展而不断发展,大致经历了人工管理、文件系统和数据库系统三个发展阶段。

1. 人工管理阶段

20世纪50年代中期以前,计算机主要用于科学计算,还没有专门用于管理数据的软件。数据与计算或处理它们的程序在一起。如果数据的类型、格式、数量或输入输出方式改变了,则程序也必须作相应的修改,数据与程序不具有独立性。一个程序中的数据,其他程序不能使用。因此,各程序之间存在着大量的重复数据,称之为数据冗余。

2. 文件系统阶段

20世纪50年代后期至60年代,计算机开始大量地用于数据处理工作。在软件方面,出现了高级语言和操作系统。操作系统中的文件系统是专门管理外存储器中数据的软件。程序和数据可以分别存储为程序文件和数据文件,因而程序与数据有了一定的独立性。常用的高级语言FORTRAN、BASIC、C等都支持使用数据文件。这个阶段称为文件系统阶段。

文件系统阶段对数据的管理虽然有了长足的进步,但是一些根本性问题并没有得到解决。例如,数据冗余度大,同一数据项在多个文件中重复出现;缺乏数据独立性,数据文件只是为了满足专门需要而设计的,供某一特定应用程序使用,数据和程序相互依赖;数据无集中管理,各个文件没有统一管理机制,无法相互联系,其安全性与完整性得不到保证。诸如此类的问题造成了文件系统管理的低效率、高成本,这就促使人们研究新的数据管理技术。

3. 数据库系统阶段

从20世纪60年代后期开始,随着社会信息量的迅速增长,需要计算机管理的数据量也在急剧增长,文件系统采用的一次存取一个记录的访问方式,以及不同文件之间缺乏相互联系的结构,越来越不能适应管理大量数据的需要。同时,人们对数据共享的需求日益增强。计算机技术的迅猛发展,特别是大容量磁盘开始使用,在这种社会需求和技术成熟的条件下,数据库技术应运而生,使得数据管理技术进入崭新的数据库系统阶段。

数据库系统克服了文件系统的种种弊端,它能够有效地储存和管理大量的数据,使数据

得到充分共享,使数据冗余大大减少,使数据与应用程序彼此独立,并提供数据的安全性和完整性统一机制。数据的安全性是指防止数据被窃取和失密,数据的完整性是指数据的正确性和一致性。用户可以以命令方式或程序方式对数据库进行操作,方便而高效。数据库系统的优越性使其得到迅速发展和广泛应用。从大型机到微型机,从 UNIX 到 Windows,推出了许多成熟的数据库管理软件,如 ORACLE、SYBASE、FoxBASE、FoxPro 和 Visual FoxPro 等等。今天,数据库系统已经成为计算机数据管理的主要方式,而由文件系统支持的数据文件,仅在数据量较小的场合下使用。

计算机网络技术的迅速发展为数据库提供了更好的运行环境,使数据库系统从集中式发展到分布式。所谓“集中”和“分布”是对数据存放地点而言的。分布式数据库把数据分散存储在网络的多个结点上,各个结点上的计算机可以利用网络通信功能访问其他结点上的数据库资源。例如,一个银行有众多储户,如果所有储户的数据都存放在一个集中式数据库中,所有储户存款、取款时都要访问这个数据库,数据传输量必然很大。如果使用分布式数据库,将众多储户的数据分散存储在离各自住所较近的储蓄所,则大多数储户就可以就近存取,仅有少量数据需要远程调用,从而大大减少了网上的数据传输量,提高了运行效率。

值得一提的是,近年来,智能数据库的研究取得了可喜的进展。传统数据库存储的数据都是已知的事实,智能数据库除了存储已知的事实外,还能存储用于逻辑推理的规则,故又称为“基于规则的数据库”(rule-based database)。例如,某智能数据库中存有“科长领导科员”的规则,如果同时存有“甲是科长”“乙是科员”等数据,它就能够推理得出“甲领导乙”的新事实。随着人工智能逐步走向实用化,对智能数据库的研究日趋活跃。演绎数据库、专家数据库和知识库系统等都属于智能数据库的范畴。

1.1.3 数据库系统的组成及特点

1. 数据库系统的组成

通常把引进了数据库技术的计算机系统称为数据库系统(DataBase System,简称 DBS)。数据库系统主要由以下几部分组成:

(1) 计算机硬件系统 需要有容量足够大的内存和外存,用来运行操作系统、数据库管理系统核心模块和应用程序,以及存储数据库。

(2) 数据库集合 存储在计算机外存上的若干个设计合理、满足应用需要的数据库。

(3) 数据库管理系统(DataBase Management System,简称 DBMS)这是帮助用户创建、维护和使用数据库的软件系统,是数据库系统的核心。较流行的微机数据库管理系统有 FoxBASE、FoxPro for Windows、Visual FoxPro、Access 等。

(4) 相关软件 包括操作系统、编译系统、应用开发工具软件和计算机网络软件等。较大型的数据库系统,通常是建立在多用户系统或网络环境中的。

(5) 人员 包括数据库管理员和用户。在大型数据库系统中,需要有专人负责数据库系统的建立、维护和管理的工作,承担该任务的人员称为数据库管理员。用户可分为两类:专业用户和最终用户。专业用户侧重于设计数据库、开发应用系统程序,为最终用户提供友好的用户界面。最终用户侧重于对数据库的使用。例如,通过数据库应用系统提供的界面对数据库进行联机查询。

2. 数据库系统的特点

数据库系统的主要特点是：数据结构化、数据共享、数据独立性和统一的数据控制功能。

(1) 数据结构化

数据库中的数据是有结构的，这种结构是由数据库管理系统所支持的数据模型表现出来的。数据库系统不仅可以表示事物内部各数据项之间的联系，而且可以表示事物与事物之间的联系。

(2) 数据共享

数据共享就是说数据库中的数据可以被多个用户、多种应用访问，这是数据库系统最重要的特点。由于数据库中的数据被集中管理、统一组织、定义和存储，可以避免不必要的冗余，因而也避免了数据的不一致性。与此同时，还带来了对数据的灵活应用方式，可以取整体数据的各种合理子集用于不同的应用系统。正因为如此，才使得多用户、多应用共享数据成为现实。

(3) 具有较高的数据独立性

在数据库系统中，数据与应用程序之间的相互依赖大大减小，数据的修改对程序不会产生大的影响或没有影响，具有较高的数据独立性。

(4) 具有统一的数据控制功能

数据共享必然伴随着并发操作，即多个用户同时使用同一个数据库。为此，数据库系统必须提供必要的保护措施，主要是以下 3 种数据控制功能：

① 安全性控制 数据库系统设置了一套安全保护措施，只有合法用户才能进行指定权限的操作，以防止非法使用造成数据的泄密和数据的破坏。

② 完整性控制 数据的完整性是指数据的正确性、有效性和相容性。系统提供必要的功能来保证数据库中的数据在输入或修改过程中始终符合原来的定义和规定。此外，当计算机系统发生故障而破坏了数据或对数据的操作发生错误时，系统能提供相应机制，将数据恢复到正确状态。

③ 并发控制 当多个用户的并发进程同时存取、修改数据库时，可能会相互干扰而得到错误的结果，并使数据库的完整性遭到破坏，因此必须对多用户的并发操作予以控制和协调。并发控制中有一概念称为事务（Transaction），它是并发控制的基本单位与控制对象。事务是一操作系列，这些操作要么都做，要么都不做。两事务的并发操作可能造成数据的错误，通常采用封锁措施来保证数据的正确性。例如，事务 T1 要修改数据 A，首先封锁它，执行完读写操作之后才解锁 A。在事务 T1 的执行过程中，如果事务 T2 也提出对数据 A 的封锁要求，则必须等待，直到事务 T1 解锁数据 A 后，T2 才能获得对数据 A 的控制权。

1.1.4 基本的数据模型

客观存在并且可以相互区别的事物称为实体。实体可以是实际事物（例如，一个学生，一个部门等），也可以是抽象事件（如借阅若干本书，一场足球比赛等）。同类型实体的集合构成一个实体集。例如，学生、教师、课程和图书等都是实体集。

实体的特性称为属性。例如，学生实体可以用学号、姓名、性别、出生日期等属性来描述；图书实体可以用编号、书名、作者、出版社、定价等属性来描述。

实体之间的对应关系称为联系，它反映现实世界事物之间的相互关联。联系分为实体内

部各属性之间的联系和实体之间的联系。实体之间的联系有以下3种类型：

(1) 一对一联系

例如，如果一所学校只有一位校长，他不能同时在其他学校兼任校长，那么学校与校长这两个实体之间存在一对一的联系。

(2) 一对多联系

例如，一所学校有多名学生，而一名学生只能在一所学校里注册，学校与学生两个实体之间便存在一对多的联系。

(3) 多对多联系

我们来看学生和课程两个实体，一名学生可以选修多门课程，而一门课程可以被多名学生选修，可见，学生与课程之间存在多对多的联系。又如，科研课题与科研人员两个实体之间也存在多对多的联系。

为了叙述方便，我们把一对一联系记作“1-1”，一对多联系记作“1-m”，多对多联系记作“m-n”。

在一个数据库系统中，为了反映事物本身及事物之间的各种联系，数据库中的数据必须具有一定的结构，这种结构用数据模型来表示。任何一个数据库管理系统都是基于某种数据模型的。基本的数据模型有3种：层次模型、网状模型和关系模型。

1. 层次模型

利用树型结构表示实体及其之间联系的模型称为层次模型。图1-1是一个层次模型的例子，它体现出实体之间一对多的联系。

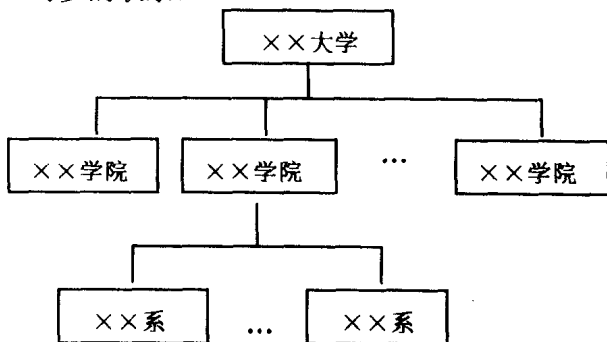


图 1-1 层次模型

2. 网状模型

利用网状结构表示实体及其之间联系的模型称为网状模型。该模型体现多对多的联系，具有很大的灵活性。图1-2给出了一个用网状模型表示某汽车制造厂中技术人员、图纸、工人和工件之间的联系。

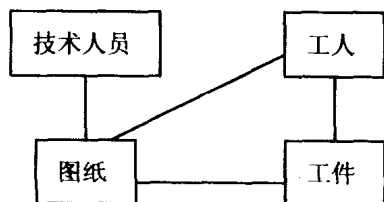


图 1-2 网状模型

3. 关系模型

用二维表结构表示实体以及实体之间联系的模型称为关系模型。关系模型把各种联系都统一描述成一些二维表，即由若干行和若干列组成的表格。每一个这样的二维表格就称为一个关系。例如，表 1-1 就是一个关系。

表 1-1 学生关系

学号	姓名	性别	出生日期	简历	照片
0101011	李晓明	男	02/26/85	memo	gen
0100112	王民	男	11/05/84	memo	gen
0100136	马玉红	女	12/15/83	memo	gen
0100108	王海	男	03/31/83	memo	gen
0100123	李建中	男	06/27/85	memo	gen
0100156	田爱华	女	08/20/83	memo	gen
0100168	马萍	女	04/25/86	memo	gen

对我们来说，无论是浏览还是设计这样一张二维表格都没有什么困难，可见，关系模型很容易被用户所接受。此外，关系模型有严格的理论基础（关系数学理论），因此，基于关系模型的关系型数据库管理系统成为当今最为流行的数据库管理系统。

经过 30 多年的发展，基于不同数据模型的数据库系统经历了第一代和第二代，而现在正向第三代前进。

第一代数据库系统是非关系型数据库系统，包括层次型和网状型数据库系统。第二代数据库系统是关系型数据库系统。与第一代数据库系统相比，第二代数据库系统的突出优点有两个：一是采用二维表作为数据结构，简单明了、易学易用；二是查询效率高，仅用一条命令即可访问整个二维表，而第一代数据库每次仅能访问一条记录。此外，通过多表联合操作还能对有联系的若干个二维表实现“关联”查询。

随着多媒体技术的发展，要求数据库能够存储图形、声音等复杂对象。将数据库技术与面向对象技术相结合的第三代数据库系统，即对象—关系数据库系统（Object-Relational DataBase Systems，简称 ORDBS）于 20 世纪 80 年代中期开始研究。第三代数据库系统除了包含第二代数据库系统的功能外，还支持图像、声音等新的数据类型，支持类、继承、函数/方法等丰富的对象机制，并能提供高度集成的、可支持客户机/服务器应用的用户接口。目前，一些厂商已经推出了可供实用的 ORDBS 产品，充分显示出其光明前景。

1.2 数据库管理系统

数据库的建立和使用，都是通过特定的数据库语言进行的。关系数据库使用的语言称为“关系数据语言”。由于查询是数据库语言的最主要的功能，故数据库语言有时也称为查询语言。被国际标准化组织（ISO）确定为关系数据语言标准的 SQL 语言，就是“结构化查询语言”（Structured Query Language）英文名的缩写。

正如使用高级语言需要编译程序或解释程序的支持一样，使用数据库语言也需要一个特