

高等院校计算机基础教育教材



2004版

新编

中文 Visual FoxPro 6.0

实用教程

- 数据库与 Visual FoxPro 概述
- 数据与数据运算 数据表的操作
- 索引与查询 报表与标签 表单
- 结构化查询语言 SQL
- Visual FoxPro 6.0 程序设计基础
- 面向对象的程序设计
- 应用程序的连编和发布
- 多用户与数据共享 综合实例

本书编委会 编

高等院校计算机课程教材

新 编

中文 Visual FoxPro 6.0 实用教程

本书编委会 编



西北工业大学出版社

155121/04

【内容提要】本书是为计算机基础教学和广大电脑爱好者自学而编写的教材。Visual FoxPro 6.0 是由美国 Microsoft 最近推出的程序开发工具软件,主要用于小型数据库系统的开发。本书是一本通用的关系型数据库管理系统实用教材,由理论知识、本章小结和课后练习三部分组成。

本书主要内容有:数据库与 Visual FoxPro 概述、数据与数据运算、数据表的操作、索引与查询、报表与标签、表单、Visual FoxPro 6.0 程序设计基础、结构化查询语言 SQL、面向对象的程序设计、应用程序的连编和发布、多用户与数据共享、综合实例等内容。

本书可作为高等院校本科生的教材,也可作为职业技术学院和高等专科学校在校生的教科书,同时也适合具有同等文化程度的读者自学之用。

图书在版编目(CIP)数据

新编中文 Visual FoxPro 6.0 实用教程/《新编中文 Visual FoxPro 6.0 实用教程》编委会编.—西安:西北工业大学出版社,2002.12

ISBN 7-5612-1559-2

I.新… II.新… III.关系数据库-数据库管理系统, Visual FoxPro 6.0-教材 IV.TP311.138

中国版本图书馆 CIP 数据核字(2002)第 068954 号

出版发行:西北工业大学出版社

通信地址:西安市友谊西路 127 号 邮编:710072 电话:029-8493844

网 址: <http://www.nwpup.com>

印 刷 者:兴平市印刷厂

开 本:787 毫米×1 092 毫米 1/16

印 张:20

字 数:540 千字

版 次:2002 年 12 月第 1 版 2004 年 1 月第 3 次印刷

定 价:24.00 元

前 言

数据库应用是计算机应用最重要的领域之一。微机数据库管理系统已由 dBASE 经 FoxBase、FoxPro 发展到 Visual FoxPro 6.0。程序设计方法,已由面向过程的结构化程序设计方法,发展到面向对象由事件驱动的程序设计方法。这是一次程序设计思想和程序设计方法的突破性进展。当前在开发数据库应用软件时,已广泛地使用了 Visual FoxPro 6.0。高等学校的计算机应用教学,也已进入了面向对象的应用开发时代。然而从目前的图书市场看,还没有一本面向对象并适应于高等院校的数据库应用数据。虽然国内已经出版了几种这方面的书籍,但都限于资料性的或操作方面的,不适于用作教程。因为对教程的要求是:概念清晰,说理透澈,循序渐进,逻辑严谨,例题丰富。为此,作者编写了此书以满足教学需要。

该教程共划分为十二章,第一章介绍了数据库与 Visual FoxPro 概述;包括数据系统基本概念,关系数据库系统,数据库应用系统等内容。第二章讲述了数据与数据运算;主要包括数据类型、常量、变量、表达式和常用函数。第三章讲述了数据表的操作;主要包括项目管理器,数据库的创建,数据表的建立与编辑和维护。第四章讲述了索引与查询;主要讲述了索引与查询命令在数据库和表中的使用以及视图。第五章讲述了报表与标签;包括创建简单报表、快速创建报表、创建标签。第六章讲述了表单;主要内容有:简单表单的创建、快速创建表单、添加表单控件以及美化表单。第七章讲述了 Visual FoxPro 6.0 程序设计基础;主要包括程序设计的几个基础命令和程序的控制结构。第八章讲述了结构化查询语言 SQL;主要讲述了 SQL 语言的使用。第九章讲述了面向对象的程序设计;包括基本概念,定义类的程序方式,利用类设计器定义类,创建和使用类库,面向对象程序设计与实例。第十章讲述了应用程序的连编和发布;包括调试器、应用程序的连编与发布。第十一章讲述了多用户与数据共享;包括多用户问题以及实现数据共享。第十二章讲述了综合实例,列举了一个固定资产管理系统的开发。

该书是一本最新的通俗而完整的实用数据库应用教程。所谓最新,指书的内容采用了面向对象由事件驱动的程序设计方法,全书都是围绕这一中心内容介绍的。通俗,指把类、对象、事件、方法以及事件驱动程序等深奥的概念,采用通俗易懂的语言,通过应用实例介绍,使学生易于理解和掌握。所谓完整,指该书的内容,既包括了面向对象由事件驱动的数据库应用程序设计所需要的基本知识,如常量、变量、表达式、函数、常用的 FoxPro 6.0 命令、程序结构等内容,也包括了类与对象、属性、方法、事件和可视化编程以及程序方式开发 Visual FoxPro 6.0 应用软件等内容。因而,在讲授或自学该教程时不再需要设计的基础,然后再通过实例,使读者较全面地掌握面向对象的事件驱动的数据库应用程序开发方法和设计技巧。

本书由《新编中文 Visual FoxPro 6.0 实用教程》编委会编写,版式设计为张延,编委会主任为张延、张军安,编委会成员主要有罗虹、吕红敏、李粉珠等。由于时间仓促和作者水平有限,书中难免存在缺点和不足之处,敬请广大读者和同行批评指正。

编 者

目 录

第一章 数据库与 Visual FoxPro 概述	1
第一节 数据库系统基本概念	1
一、数据、信息和数据库	1
二、数据库系统	2
三、数据模型	4
四、数据库管理系统	6
第二节 关系数据库系统	6
一、关系术语	7
二、关系的特点	7
三、关系设计	7
四、关系运算	8
第三节 数据库应用系统	9
第四节 Visual FoxPro 概述	10
一、Visual FoxPro 的特点	10
二、Visual FoxPro 的运行环境和安装	12
三、Visual FoxPro 的启动与退出	13
四、Visual FoxPro 的用户界面及工作方式	13
五、Visual FoxPro 的辅助设计工具	19
第五节 Visual FoxPro 的命令格式及文件类型	21
一、Visual FoxPro 的命令格式	21
二、Visual FoxPro 的文件类型	22
本章小结	23
习 题	24
第二章 数据与数据运算	25
第一节 数据类型	25
第二节 常量与变量	26
一、常量	27
二、变量	27
三、内存变量赋值命令	28
第三节 表达式	29
一、运算符	29
二、表达式	32

第四节 常用函数	33
一、字符及字符串处理函数	33
二、数学运算函数	34
三、转换函数	35
四、日期函数	36
五、测试函数	37
六、键值函数	40
七、数组函数	41
八、其他函数	42
本章小结	44
习 题	44
 第三章 数据表的操作	45
第一节 项目管理器	45
一、项目管理器简介	45
二、管理项目的数据库	47
三、管理项目的文档	48
四、管理项目的各种操作	48
第二节 使用工具栏	52
一、工具栏简介	52
二、创建新工具栏	53
三、修改 Visual FoxPro 6.0 工具栏	54
第三节 创建数据库与表	54
一、表和数据库的概念	54
二、建立工作目录	55
三、建立数据库文件	56
四、建立表文件	58
第四节 编辑数据表	62
一、显示表记录	62
二、浏览窗口的分隔	64
三、添加表记录	66
四、不同类型文件的复制	68
五、查找与替换	69
第五节 维护数据表	73
一、修改表文件结构	73
二、输入“照片”字段中的记录	75
三、表文件记录删除	77
四、数据表字段特性	79

本章小结	82
习 题	82
第四章 索引与查询	83
第一节 索引数据表	83
一、索引关键字	83
二、建立索引	84
三、查看和修改索引	87
四、使用索引	87
第二节 信息查询	89
一、查询含义	89
二、创建查询文件	89
三、建立数据表间的联接	91
四、设定查询条件	92
五、设置数据排序	93
六、设置分组查询	93
七、杂项设置	94
第三节 视 图	96
一、视图含义	96
二、创建本地视图	96
三、创建远程视图	101
本章小结	104
习 题	104
第五章 报表与标签	105
第一节 创建简单报表	105
一、报表布局	105
二、利用向导创建报表	107
三、利用一对多向导创建报表	111
第二节 快速创建报表	115
一、快速创建报表	115
二、设置报表布局	118
三、报表控件的使用	120
四、美化报表设计	125
第三节 创建标签	134
一、标签概念	134
二、利用标签向导创建	135

三、利用标签设计器创建	138
四、打印报表或标签	140
本章小结	140
习 题	140
第六章 表 单	141
第一节 简单表单的创建	141
一、表单创建方法与种类	141
二、利用向导创建表单	141
三、保存及运行表单	145
四、创建一对多表单	147
第二节 快速创建表单	150
一、启动表单设计器	150
二、快速创建表单	152
三、利用命令创建表单	154
第三节 添加表单控件	155
一、表单控件简介	155
二、添加文本框和标签控件	156
三、添加组合框和列表框控件	157
四、添加命令按钮控件	160
五、添加页框控件	162
六、添加表格控件	163
第四节 美化表单	167
一、完善控件按钮	167
二、改变控件的格式	168
三、改变控件的颜色	169
四、布局表单中的对象	170
五、设置对象的 Tab 键次序	170
本章小结	171
习 题	172
第七章 Visual FoxPro 6.0 程序设计基础	173
第一节 结构化程序设计纲要	173
一、对问题进行说明	173
二、分解问题	174
三、编制模块	174
四、示例程序	175

五、测试模块	176
六、组装全部模块	177
七、整体测试	178
第二节 基本语句和状态设置命令	178
一、输出命令	178
二、基本输入 (ACCEPT/INPUT/WAIT)	181
第三节 程序的控制结构	184
一、顺序结构	184
二、选择结构	185
三、循环结构	191
第四节 过程与函数	194
一、函数	194
二、过程和过程设计	196
本章小结	199
习 题	199
 第八章 结构化查询语言 SQL	 200
第一节 SQL 概要	200
第二节 SQL 数据定义功能	200
一、建立表结构	201
二、修改表结构	204
三、建立临时表	206
四、删除表	207
第三节 SQL 的数据修改功能	207
一、插入数据	207
二、删除数据	208
三、更新数据	209
第四节 SQL 的数据查询功能	209
一、简单查询	211
二、嵌套查询	215
三、连接查询	217
四、分组及使用库函数查询	219
本章小结	221
习 题	222

第九章 面向对象的程序设计	223
第一节 基本概念	223
一、对象与类	223
二、类的特征和基类	226
三、事件	229
第二节 定义类的程序方式	230
一、定义类的命令	231
二、向容器类中添加对象	232
三、指定方法和事件程序	234
四、保护和隐藏类成员	236
五、按类层次调用方法或事件代码	237
第三节 利用类设计器定义类	239
第四节 创建和使用类库	243
一、创建类库的两种方法	243
二、修改库中的类定义	243
三、使用类库的命令	244
第五节 面向对象程序设计 with 实例	245
一、设计一个应用系统的具体实现过程	245
二、一个数据库应用系统实例	246
本章小结	260
习 题	260
第十章 应用程序的连编和发布	261
第一节 调试器	261
一、调试器的设置	261
二、调试器的使用	262
第二节 应用程序的连编	266
一、设置主文件	266
二、连编应用程序	268
第三节 应用程序的发布	269
本章小结	274
习 题	274

第十一章 多用户与数据共享	276
第一节 多用户环境中的数据访问	276
一、多用户环境中数据访问的方式	276
二、数据锁定的方法	278
三、数据解锁的方法	281
四、数据工作期	281
第二节 数据缓冲技术	282
一、缓冲方法与锁定方式	282
二、使用缓冲技术更新数据	284
第三节 事务处理	285
一、事务处理的控制命令	285
二、使用事务	286
三、事务处理的嵌套	287
第四节 冲突处理	287
一、死锁	287
二、出错处理例程	287
三、冲突检测的函数	288
本章小结	288
习 题	288
第十二章 综合实例	289
第一节 课程设计的目的和意义	289
第二节 系统功能	289
第三节 创建项目管理文件	289
一、数据表设计	291
二、主控界面设计	292
三、录入表单的制作	294
四、制作浏览表单	298
五、制作固定资产报表系统	301
六、用主窗体调用各子窗体	306

第一章 数据库与 Visual FoxPro 概述

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, 为促进祖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 Visual FoxPro 的特点和基础知识, 为学习其余各章打下良好的基础。

第一节 数据库系统基本概念

一、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 1985 年 6 月 28 日, 身高 1.72 m, 体重 66 kg, 其中 1985 年、6 月、28 日、1.72 m 和 66 kg 等就是数据, 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 “1985.6.28”、“{06/28/85}” 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大大地拓宽了, 不仅包括数字、字母、文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、对决策有价值的数据。请看一个简单例子: 某排球队中, 每个队员的身高数据为 1.85 m, 1.97 m, 1.86 m, …, 经过计算得到平均身高为 1.88 m, 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, 它反映出本届学生整体的年龄状况。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, 是数据的语义解释。信息是有价值的。信息的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, 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、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，这就是所谓的“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。例如，学生成绩数据库中存储的是学号、姓名、性别及其各门课程的成绩等，这些数据是相关的。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。数据以什么形式存储，什么是结构化，何谓相关数据，这几个问题明确了，你对数据库也就有了一个全面而深入的理解，而不是简单地将它看作是存储数据的仓库。

二、数据库系统

1. 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，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，各程序之间存在大量的重复数据，称为数据冗余。

（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管理中的数据处理工作。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的数据管理软件。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，各自为政，其安全性与完整性无法保证。诸如此类的问题造成了文件系统管理的低效率、高成本，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急剧增长，文件系统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。

2. 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, DBS)。数据库系统由以下几部分组成:

(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: 存储在计算机外存设备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 (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。比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。最终用户侧重对数据库的使用, 主要是对数据库进行联机查询, 或者通过数据库应用系统提供的界面使用数据库。

3. 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。数据冗余是指数据的重复。

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,以防止不合法的使用所造成数据的泄密和被破坏。

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、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。事务是一系列操作,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。

三、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。

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。联系分为两种,一种是实体内部各属性之间的联系,另一种是实体之间的联系。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:例如,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。

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。数据模型主要有 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 1.1.1 是一个层次模型的例子,它体现出实体之间一对多的联系。

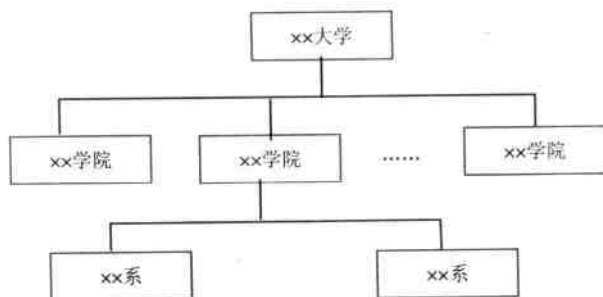


图 1.1.1 层次模型

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 1.1.2 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。

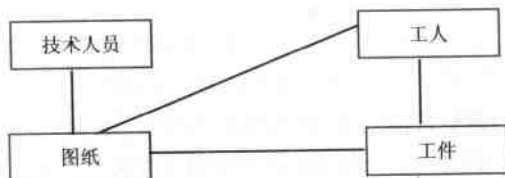


图 1.1.2 网状模型

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 学生关系实例

学号	姓名	性别	出生日期	个人简历	照片
0205001	道明寺	男	02/25/81	memo	Gen
0205002	花泽类	男	05/23/80	memo	Gen
0205003	林杉菜	女	03/23/78	memo	Gen
0205004	西门	男	05/04/79	memo	Gen
0205005	美作	男	08/24/79	memo	Gen
0205006	李真	女	07/21/80	memo	Gen
0205007	腾堂静	女	01/14/80	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，即是说，关系模型很容易被用户所接受，并且关系模型以关系数学理论为基础，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。这就是本书重点介绍关系型数据库管理系统的根本原因。

经过 30 多年的发展，基于不同数据模型的数据库系统先后经历了第一代和第二代，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 ORDBS (Object Relational DataBase Systems) 于 20 世纪 80 年代中期开始研究,虽然目前尚处于发展过程之中,但已经显示出其光明的前景。

四、数据库管理系统

在数据库系统中,数据成为多个用户或多个应用程序的共享资源,它已经从应用程序中独立出来,由数据库管理系统 (DataBase Management System, DBMS) 来统一管理。一般地说,数据库管理系统具有下列功能。

(1) 数据定义功能:数据库管理系统向用户提供数据定义语言 (Data Definition Language, DDL), 用于描述数据库的结构。例如,关系数据库的标准语言 SQL 的 DDL 语言,一般设置有 create table/index, alter table, drop table/index 等语句,可以分别用于建立、修改或删除关系数据库的二维表结构,或者定义或删除数据库表的索引。

(2) 数据操作功能:数据库管理系统向用户提供数据操作语言 (Data Manipulation Language, DML), 支持用户对数据库中的数据进行查询、追加、插入、删除、修改等操作。在不同的数据库管理系统中,数据操作语言的语法格式也不同。按其实现方法可分为两类:一类数据操作语言可以独立使用,不依赖于任何其他程序设计语言,称为自含型或自主型语言;另一类是宿主型数据操作语言,需要嵌入到宿主语言 (例如 FORTRAN, COBOL, C 等) 中使用。在使用高级语言编写应用程序时,如果需要调用数据库中的数据,则需要用宿主型数据操作语言的语句来实现。因此,数据库管理系统必须包含数据操作语言的编译或解释程序。

(3) 控制和管理功能:数据库中的数据是宝贵的共享资源,必须有一定的控制手段来保障资源不受破坏。因此,数据库管理系统必须具有控制和管理功能。如前所述,在多用户使用数据库时对数据进行的“并发控制”,对用户权限实施监督的“安全性检查”,数据的备份、恢复和转储功能等。数据库系统的规模越大,这类功能也就越强。需要指出的是,一种数据库管理系统只能支持一种模型的数据库系统。目前,微机上使用的数据库管理系统都是关系型数据库管理系统,简称 RDBMS (Relational DBMS)。它们提供的数据库语言都具有“一体化”的特点,即集数据定义语言 DDL 和数据操作语言 DML 于一体,在数据库管理系统 DBMS 的统一管理下完成上述各种功能。

(4) 数据字典:数据字典 (Data Dictionary) 中存放着系统中所有数据的定义和设置信息。例如,字段属性、记录规则、表间关系等,提供了对数据库数据描述的集中管理手段。它使用户在设计和修改数据库时有更大的灵活性。例如,用户可以设置字段默认值,创建数据库表间的永久关系等。

第二节 关系数据库系统

目前,关系数据库理论日趋成熟,在微机数据库系统中得到普遍的应用。关系数据库可分为单用户数据库和多用户数据库。早期的微机数据库都是单用户系统,只能使一人使用。随着局域网应用的发展,供网络用户共享的多用户数据库开始流行。Visual FoxPro 就是一种多用户数据库系统。在它以前,已有 FoxBASE+、FoxPro 等多用户数据库供微机用户选用。