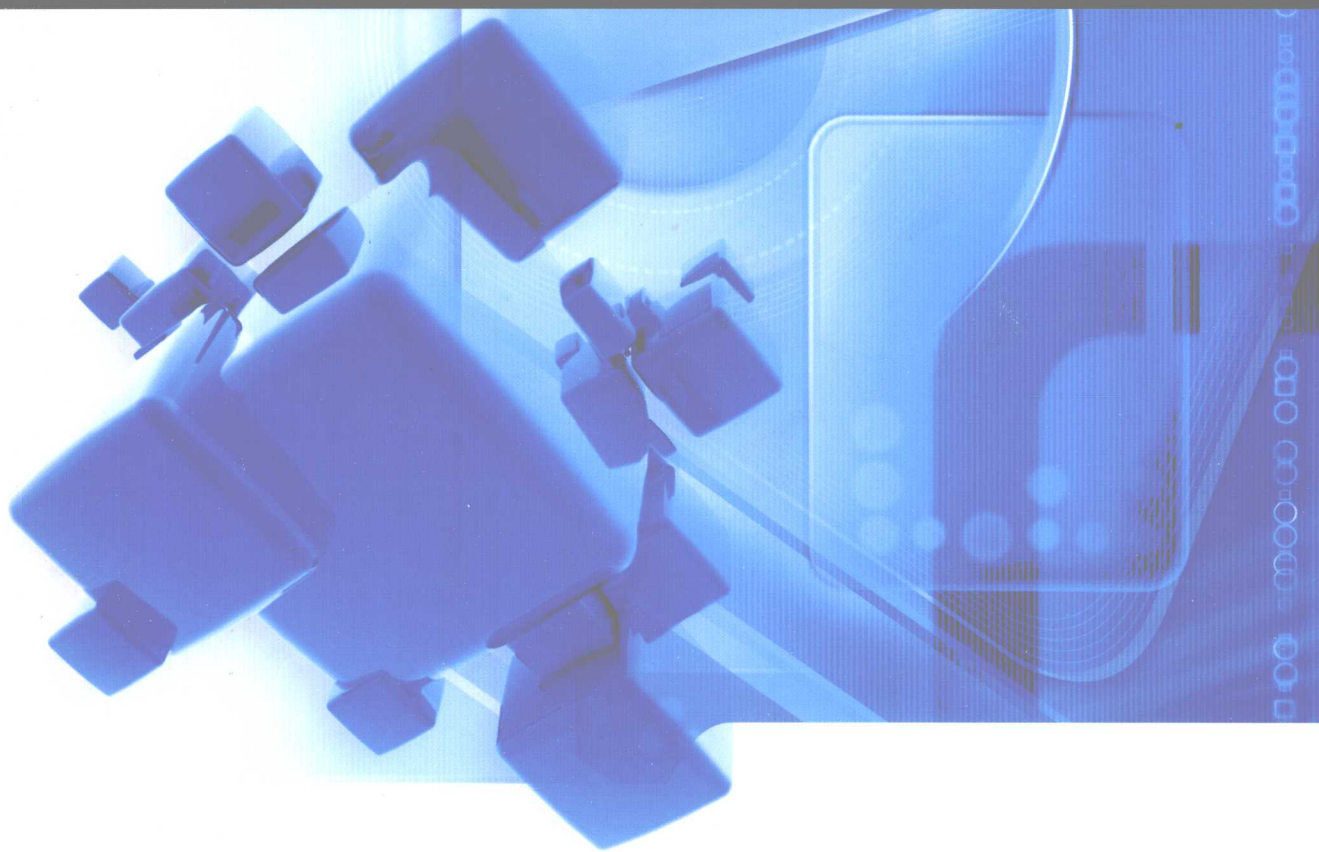


21世纪高等学校
计算机应用技术教材



彭国星 陈芳勤 主编

Visual FoxPro

程序设计



化学工业出版社

21 世纪高等学校计算机应用技术教材

Visual FoxPro 程序设计

彭国星 陈芳勤 主编



化学工业出版社

· 北京 ·

本书根据目前我国高等院校非计算机专业计算机课程开设的实际情况,以及全国计算机等级考试数据库考试大纲及湖南省非计算机专业等级考试大纲的要求,并结合多年从事数据库教学和数据库程序开发的实践经验编写而成。从熟悉 Visual FoxPro 的开发环境以及基本操作入手,结合了大量的数据库使用、实例,深入浅出,系统地介绍了 Visual FoxPro 数据库基础,系统初步,表、索引及数据库的操作、查询,视图,程序设计,表单,报表,工具栏及菜单栏。本书可作为高等院校非计算机专业学习数据库程序设计用教材,也可作为计算机应用人员学习 Visual FoxPro 的教材和参考用书,也可作为广大电脑学习者的参考资料。

图书在版编目 (CIP) 数据

Visual FoxPro 程序设计/彭国星, 陈芳勤主编. —北京:
化学工业出版社, 2009.1
21 世纪高等学校计算机应用技术教材
ISBN 978-7-122-04370-2

I. V… II. ①彭…②陈… 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 (2008) 第 213665 号

责任编辑: 旷英姿
责任校对: 郑捷

文字编辑: 陈元
装帧设计: 史利平

出版发行: 化学工业出版社 (北京市东城区青年湖南街 13 号 邮政编码 100011)
印 装: 大厂聚鑫印刷有限责任公司
787mm×1092mm 1/16 印张 17 字数 432 千字 2009 年 3 月北京第 1 版第 1 次印刷

购书咨询: 010-64518888 (传真: 010-64519686) 售后服务: 010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书, 如有缺损质量问题, 本社销售中心负责调换。

定 价: 29.00 元

版权所有 违者必究

21 世纪高等学校计算机应用技术教材

《Visual FoxPro 程序设计》

主 编	彭国星	陈芳勤	
编 者	(按姓名笔画排序)		
	刘承宗	许赛华	李 欣
	陈芳勤	周 浩	唐黎黎
	黄国辉	彭国星	童 启

前 言

高级语言程序设计已由面向过程的程序设计逐步向面向对象的程序设计过渡,各种可视化程序设计语言越来越受到广大计算机应用工作者的喜爱。在数据库应用技术领域中,Visual FoxPro 6.0 中文版是适用于微型计算机系统的最优秀的小型关系型数据库管理系统之一。

本书是由多年从事高校计算机基础教学的专职教师参照《全国计算机等级考试考试大纲》(二级 Visual FoxPro)和《湖南省普通高等学校非计算机专业学生计算机应用水平考试大纲》(一、二级)的要求编写而成的。书中不少内容就是这些具有丰富的理论知识和教学经验的教师们对实践经验的总结。

本书从熟悉 Visual FoxPro 的开发环境以及基本操作入手,结合了大量的数据库使用、实例,深入浅出,系统地介绍了数据库基础,Visual FoxPro 系统初步,表、索引及数据库的操作、查询,视图,程序设计,表单,报表,工具栏及菜单栏。

同时,本书还兼顾了全国计算机等级考试的相关内容,重点突出了湖南省普通高等学校非计算机专业学生计算机应用水平考试的有关内容,从而提高学生的获证能力。为了方便教师教学和学生自学使用,编者精心制作了课件。本书适用于大中专院校非计算机专业进行数据库应用技术的教学,也可作为广大电脑学习者的参考资料。

本书由彭国星、陈芳勤主编,童启、许赛华、黄国辉、周浩、刘承宗、李欣、唐黎黎参编。第1章由童启编写,第2章由许赛华编写,第3章由黄国辉编写,第4、5章由周浩编写,第6章由陈芳勤编写,第7章由刘承宗编写,第8章由李欣编写。第9章由唐黎黎编写。最后全书由陈芳勤统稿。

由于编者水平有限,编写时间较紧,书中难免有不妥之处,恳请读者批评指正。

编 者
2008 年 12 月

目 录

第 1 章 Visual FoxPro 基础	1	2.1.11 通用型	31
1.1 数据库系统的基本概念	1	2.1.12 字符型 (二进制)	31
1.1.1 计算机数据处理的发展	1	2.1.13 备注型 (二进制)	31
1.1.2 数据库系统的组成	2	2.2 常量和变量	31
1.1.3 数据库系统的结构	4	2.2.1 常量	31
1.2 数据模型	6	2.2.2 变量	32
1.2.1 实体相关概念	6	2.2.3 内存变量的操作	33
1.2.2 实体间联系及联系的种类	6	2.3 运算符和表达式	38
1.2.3 实体联系的表示方法	7	2.3.1 计算和显示命令	38
1.2.4 数据模型简介	7	2.3.2 运算符	38
1.3 关系数据库	9	2.3.3 表达式	43
1.3.1 关系模型	9	2.3.4 运算优先级	43
1.3.2 关系运算	11	2.4 函数	44
1.4 Visual FoxPro 系统初步认识	12	2.4.1 数值处理函数	44
1.4.1 Visual FoxPro 系统概述	12	2.4.2 字符处理函数	46
1.4.2 Visual FoxPro 6.0 的用户界面	13	2.4.3 日期时间函数	48
1.4.3 Visual FoxPro 6.0 的工作方式	16	2.4.4 类型转换函数	49
1.4.4 Visual FoxPro 的配置	17	2.4.5 测试函数	51
1.4.5 Visual FoxPro 设计工具	19	习题	53
1.4.6 项目管理器	22	第 3 章 表与数据库	55
习题	26	3.1 表的创建	55
第 2 章 数据与数据运算	29	3.1.1 表的概念	55
2.1 Visual FoxPro 的数据类型	29	3.1.2 表结构的设计	55
2.1.1 字符型	29	3.1.3 表结构的创建	58
2.1.2 数值型	29	3.1.4 表数据的输入	60
2.1.3 货币型	29	3.2 表的维护	61
2.1.4 逻辑型	30	3.2.1 表文件的打开与关闭	61
2.1.5 日期型	30	3.2.2 表结构的显示与修改	63
2.1.6 日期时间型	30	3.2.3 记录的显示	65
2.1.7 整型	30	3.2.4 记录的修改	68
2.1.8 双精度型	31	3.2.5 记录指针的定位	69
2.1.9 浮点型	31	3.2.6 记录的增加	71
2.1.10 备注型	31	3.2.7 记录的删除与恢复	73

3.3 排序与索引	75	4.4.5 查询集合的并运算	109
3.3.1 排序	75	4.5 定义视图	110
3.3.2 索引	76	4.5.1 视图的定义和删除	110
3.4 多表操作	79	4.5.2 视图查询及操作	111
3.4.1 多工作区	79	4.5.3 关于视图的说明	111
3.4.2 表之间的关联	81	习题	112
3.4.3 表之间的联接	82	第5章 查询与视图	113
3.5 表文件的复制	83	5.1 创建查询	113
3.5.1 复制任何文件	83	5.1.1 基本概念	113
3.5.2 表内容复制	83	5.1.2 使用“查询向导”建立查询	114
3.5.3 表结构复制	84	5.1.3 使用“查询设计器”创建查询	118
3.5.4 文件重命名	84	5.1.4 定项输出查询结果	125
3.6 数据库的创建及其基本操作	84	5.1.5 利用查询结果建图形	126
3.6.1 基本概念	84	5.2 创建视图	129
3.6.2 数据库设计的一般步骤	85	5.2.1 视图的概念	129
3.6.3 创建数据库	86	5.2.2 使用向导创建视图	129
3.6.4 数据库操作命令	86	5.2.3 利用“视图设计器”创建视图	132
3.7 在数据库中添加和移去表	88	5.2.4 远程视图与连接	134
3.7.1 在数据库中直接创建表	88	5.2.5 视图与数据更新	135
3.7.2 向数据库中添加表	91	5.2.6 使用视图	136
3.7.3 从数据库中移去表	92	习题	137
3.8 数据的完整性	92	第6章 程序设计基础	138
3.8.1 实体完整性	92	6.1 程序文件的建立与运行	138
3.8.2 域完整性	93	6.1.1 程序的概念	138
3.8.3 参照完整性	93	6.1.2 程序文件的建立与运行	139
习题	95	6.1.3 程序中的辅助命令	141
第4章 结构化查询语言 SQL	98	6.1.4 简单的输入输出命令	142
4.1 SQL 概述	98	6.2 程序的基本结构	146
4.2 SQL 的数据定义命令	99	6.2.1 顺序结构	146
4.2.1 定义基本表	99	6.2.2 选择结构	146
4.2.2 表的删除	101	6.2.3 循环结构	150
4.2.3 表结构的修改	101	6.3 多模块程序	158
4.3 SQL 的数据操作命令	102	6.3.1 模块的定义和调用	158
4.3.1 插入	102	6.3.2 参数传递	160
4.3.2 更新	103	6.3.3 变量的作用域	163
4.3.3 删除	103	6.4 程序的调试	165
4.4 SQL 的数据查询命令	103	6.4.1 程序中常见的错误	165
4.4.1 简单查询	104	6.4.2 调试器环境	165
4.4.2 联接查询	107	6.4.3 设置断点	166
4.4.3 嵌套查询	108	6.4.4 调试菜单	167
4.4.4 分组与聚合函数	108	6.5 常用算法实例	168

6.5.1 累加、连乘	168	习题	220
6.5.2 求素数	169	第 8 章 报表设计	223
6.5.3 穷举法	170	8.1 创建报表	223
6.5.4 递推法	172	8.1.1 利用报表向导创建报表	223
6.5.5 求最大值或最小值	172	8.1.2 利用快速报表创建报表	227
6.5.6 有关数据库的简单程序	173	8.1.3 利用报表设计器创建报表	228
习题	174	8.2 设计报表	229
第 7 章 表单设计	180	8.2.1 报表工具栏	230
7.1 面向对象程序设计	180	8.2.2 设置报表数据源	231
7.1.1 基本概念	180	8.2.3 设计报表布局	232
7.1.2 Visual FoxPro 中的类	181	8.3 修改和输出报表	238
7.1.3 Visual FoxPro 中类的操作	183	8.3.1 修改报表	238
7.1.4 为控件或容器类添加对象	185	8.3.2 输出报表	240
7.1.5 为类添加成员和定义事件	185	习题	241
7.2 表单的基本知识	186	第 9 章 菜单与工具栏设计	243
7.2.1 建立表单	186	9.1 菜单系统及其规划	243
7.2.2 表单操作	191	9.1.1 菜单系统的结构	243
7.2.3 表单的数据环境	192	9.1.2 系统菜单	243
7.2.4 表单中对象的属性设置	194	9.1.3 菜单系统的规划	245
7.2.5 创建单文档和多文档界面	198	9.2 菜单设计	245
7.2.6 用表单集扩充表单	201	9.2.1 菜单设计的基本过程	245
7.2.7 表单的常用事件	202	9.2.2 下拉式菜单设计	246
7.2.8 向表单添加控件	202	9.2.3 快捷菜单设计	250
7.3 常用表单控件简介	204	9.3 菜单的常规选项和菜单选项	253
7.3.1 标签	204	9.3.1 常规选项	253
7.3.2 文本框	204	9.3.2 菜单选项	253
7.3.3 命令按钮	206	9.3.3 定制菜单系统	254
7.3.4 命令按钮组	208	9.4 顶层表单的菜单加载	255
7.3.5 选项按钮组	209	9.5 用编程方式定义菜单	256
7.3.6 复选框	211	9.5.1 条形菜单定义	256
7.3.7 组合框	212	9.5.2 弹出式菜单定义	256
7.3.8 列表框	216	9.6 设计工具栏	258
7.3.9 微调按钮	217	9.6.1 定制 Visual FoxPro 工具栏	258
7.3.10 表格控件	217	9.6.2 定制工具栏类	259
7.3.11 图像控件	218	9.6.3 在表单集中添加自定义工具栏	261
7.3.12 计时器控件	219	9.6.4 协调菜单和用户自定义工具 栏的关系	262
7.3.13 页框控件	219	习题	263
7.3.14 形状和线条	219	参考文献	264

第1章 Visual FoxPro 基础

数据处理是指对数据的收集、整理、传输、加工、存储、更新和维护等活动。数据处理技术随着计算机技术的发展而发展,数据库技术应运而生,经过 40 多年的迅速发展,取得了辉煌的成就,已成为计算机应用领域中一个重要的分支。

Visual FoxPro 是目前微机上优秀的数据库管理系统之一,正如其名中冠之的“Visual”一样,它采用了可视化的、面向对象的程序设计方法,大大简化了应用系统的开发过程,并提高了系统的模块性和紧凑性。

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,才能开发出实用的、水平较高的数据库应用系统。通过本章学习我们应了解数据库系统的基本概念、数据模型、关系数据库基本理论,数据库设计基础和 Visual FoxPro 系统初步知识。

1.1 数据库系统的基本概念

1.1.1 计算机数据处理的发展

1.1.1.1 数据与数据处理

数据(data)是对客观事物的符号表示,是用于表示客观事物的未经加工的原始素材,如图形符号、数字、字母等。在计算机科学中,数据是指所有能输入到计算机并被计算机程序处理的符号的介质的总称,是用于输入电子计算机进行处理,具有一定意义的数字、字母、符号和模拟量等的通称。数据不仅包括数字、字母、文字和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、动画、影像、声音等多媒体数据。但目前使用最多、最基本的仍然是文字数据。

现实世界中的数据往往是原始的、非规范的,但它是数据的原始集合,通过这些原始数据的处理,才能产生新的数据(信息)。这一处理包括对数据的收集、记录、分类、排序、存储、计算/加工、传输、制表和递交等操作,这就是数据处理的概念。从数据处理的角度而言,信息是一种被加工成特定形式的数据,这种数据形式对于数据接收者来说是有意义的。因此,人们有时说的“信息处理”,其真正含义应该是为了产生信息而处理数据。

1.1.1.2 计算机数据管理的发展

数据管理经历了从低级到高级的发展过程,这一过程大致可分为三个阶段:手工管理阶段、文件系统阶段、数据库系统阶段。

(1) 手工管理阶段

在 20 世纪 50 年代中期以前,计算机主要用于科学计算,计算机上没有操作系统,没有管理数据的专门软件,也没有像磁盘这样的设备来存储数据。这个时期数据管理的特点是:

- ① 数据不保存;
- ② 数据和程序一一对应,即一组数据对应一个程序,不同应用程序的数据之间是相互独立、彼此无关的;
- ③ 没有软件系统对数据进行管理,程序员不仅要规定数据的逻辑结构,而且还要在程序中设计物理结构,包括存储结构、存取方法及输入输出方式等;也就是说数据对程序不具有

独立性，数据是程序的组成部分，一旦数据在存储上有所改变，必须修改程序。

(2) 文件系统阶段

数据管理从 20 世纪 50 年代后期进入文件系统阶段。操作系统中已经有了专门的管理数据的软件，一般称为文件系统。所谓文件系统是一种专门管理数据的计算机软件。在文件系统中，按一定的规则将数据组织成为一个文件，应用程序通过文件系统，对文件中的数据进行存取和加工。

文件系统数据管理的特点如下。

- ① 文件的逻辑结构与存储结构的转换由系统进行，使程序与数据有了一定的独立性。
- ② 文件系统文件基本上对应于某个应用程序，即数据还是面向应用的。不同的应用程序可以实现以文件为单位的共享，但是当所需要的数据有部分相同时，也必须建立各自的文件。
- ③ 文件系统文件是为某个应用服务的，文件的逻辑结构对该应用程序来说是优化的。因此，要想对现有的数据再增加一些应用很困难，系统不易扩充。一旦数据的逻辑结构改变，必须修改程序。而应用程序的改变，也将影响文件的数据结构的改变。数据和程序缺乏独立性。

(3) 数据库系统阶段

从 20 世纪 60 年代后期开始，需要计算机管理的数据量急剧增长，并且对数据共享的需求日益增强。文件系统的数据管理方法已无法适应开发应用系统的需要。为了实现计算机对数据的统一管理，达到数据共享的目的，发展了数据库技术。

数据库技术的主要目的是有效地管理和存取大量的数据资源，包括：提高数据的共享性，使多个用户能够同时访问数据库中的数据；减少数据的冗余度，提高数据的一致性和完整性；提供数据与应用程序的独立性，从而减少应用程序的开发和维护代价。

为数据库建立、使用和维护而配置的软件称为数据库管理系统 (DataBase Management System, DBMS)。数据库管理系统利用了操作系统提供的输入/输出控制和文件访问功能，因此它需要在操作系统的支持下运行。Visual FoxPro 就是一种在微机上的数据库管理系统软件。数据库系统中数据与程序的关系如图 1-1 所示。

1.1.2 数据库系统的组成

1.1.2.1 数据库系统的组成

数据库系统是指引进数据库技术后的计算机系统，实现有组织地、动态地存储大量相关数据，提供数据处理和信息资源共享的便利手段。一般来说，数据库系统由五部分组成：计算机系统、数据库、数据库管理系统、数据库管理员和用户。它们之间的层次如图 1-2 所示。

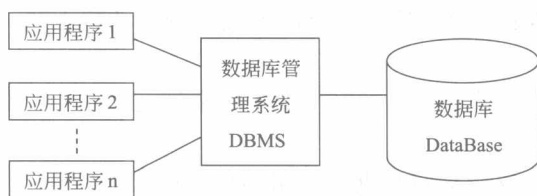


图 1-1 数据库系统中数据与程序的关系

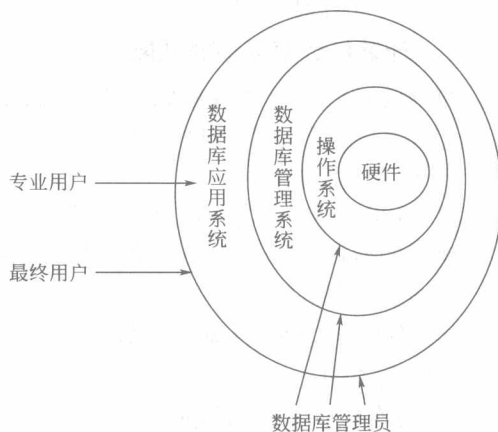


图 1-2 数据库系统层次示意图

(1) 计算机系统

计算机系统是指提供数据库系统运行的软、硬件平台。硬件平台一般包括计算机中央处理器、足够大的内存、足够大容量的磁盘等联机直接存取设备和较高的通道能力,以支持对外存的频繁访问,同时还包括足够多的联机存储介质。软件平台指计算机操作系统提供的运行环境及开发工具。计算机系统提供的运行环境不恰当,将会影响数据库系统的运行效率或者根本无法运行。

(2) 数据库

数据库(DataBase, DB)是指存储在计算机外存设备或网络存储设备上的、结构化的相关数据集合。它不仅包括描述事物的数据本身,而且还包括相关事务之间的联系。

数据库中的数据面向多种应用,可以被多个用户、多个应用程序共享。例如,某企业、组织或行业所涉及的全部数据的汇集。其数据结构独立于使用数据的程序,对于数据的增加、删除、修改和检索由系统软件统一控制。

(3) 数据库管理系统

为了让多种应用程序并发地使用数据库中具有最小冗余度的共享数据,必须使数据与程序具有较高的独立性。这就需要一个软件系统对数据实行专门管理,即数据库管理系统。数据库管理系统(DataBase Management System, DBMS)是管理数据库的工具,是应用程序与数据库之间的接口,是为数据库的建立、使用和维护而配置的软件。

它是数据库系统的核心,建立在操作系统的基础上,实现对数据库的统一管理和控制。它需要解决两个问题:科学地组织和存储数据;高效地获取和维护数据。

它的主要功能包括以下四个方面。

① 数据定义功能:DBMS 提供数据定义语言(Data Definition Language, DDL),用户通过它可以方便地对数据库中的数据对象进行定义。

② 数据操作功能:DBMS 提供数据操作语言(Data Manipulation Language, DML),用户可以使用 DML 操作数据实现对数据库的基本操作,如查询、插入、删除和修改等。

③ 数据库的运行管理功能:它包括并发控制、存取控制(安全性检查)、完整性约束条件的检查与执行等。所有数据库的操作都要在这些控制程序的统一管理下进行,以保证事务的正确运行和数据库数据的正确有效。

④ 数据库的建立和维护功能:包括数据库初始数据的输入、转换功能、数据库的转储、恢复功能,数据库的重组组织功能和性能监视、分析功能等。

(4) 数据库管理员

数据库管理员(DataBase Administrator, DBA)是负责建立、维护和管理数据库系统的操作人员,他们应有丰富的计算机应用经验,对业务数据的性质、结构、流程有较全面的了解。DBA 的职责包括定义并存储数据库的内容、监督和控制数据库的使用、负责数据库的日常维护、必要时重新组织和改进数据库。

(5) 用户

数据库系统的用户(User)分为两类。一类是最终用户,主要对数据库进行联机查询或通过数据库应用系统提供的界面来使用数据库。如操作员、企业管理人员、工程技术人员,他们不必了解数据库系统的结构和模式。另一类为专业用户即程序员,他们负责设计应用系统的程序模块,对数据库进行操作。他们有较强的计算机专业知识,可对所授权使用到的数据库(或视图)进行查询、插入、删除、修改操作,因此他们要了解数据库的外模式。

1.1.2.2 数据库系统的特点

数据库系统有如下主要特点。

(1) 数据共享

在数据库系统中，对数据的定义和描述已经从应用程序中分离出来，通过数据库管理系统来统一管理。数据库中的数据不仅可为同一企业或结构之内的各个部门所共享，也可可为不同单位、地域甚至不同国家的用户所共享。

(2) 数据结构化

数据库中的数据是有结构的，这种结构由数据库管理系统所支持的数据模型表现出来，任何数据库管理系统都支持一种抽象的数据模型。数据库中的数据文件是有联系的，在整体上服从一定的结构形式。关于数据模型将在 1.2.4 节中具体介绍。

(3) 较高的数据独立性

数据独立性是指数据独立于应用程序而存在。在文件系统中，数据结构和应用程序相互依赖、相互影响。数据库系统则力求减少这种依赖，实现数据的独立性。在数据库系统中，数据库管理系统提供映像功能，实现了应用程序对数据的总体逻辑结构、物理存储结构之间较高的独立性。用户只以简单的逻辑结构来操作数据，无需考虑数据在存储器上的物理位置与结构。

(4) 冗余度可控

文件系统中数据专用，每个用户拥有和使用自己的数据，造成许多数据重复，这就是数据冗余。在数据库系统实现共享后，不必要的重复将删除，但为了提高查询效率，有时也保留少量重复数据，其冗余度可由设计人员控制。

(5) 数据统一控制

为保证多个用户能同时正确地使用同一个数据库，数据库系统提供以下三方面的数据控制功能。

① 安全性控制：数据库设置一套安全保护措施，保证只有合法用户才能进行指定权限的操作，防止非法使用所造成的数据泄密和破坏。

② 完整性控制：数据库系统提供必要措施来保证数据的正确性、有效性和相容性，当计算机系统出现故障时，提供将数据恢复到正确状态的相应机制。

③ 并发控制：当多用户并发进程同时存取、修改数据库时，可能会发生相互干扰使数据库的完整性遭到破坏，因此，数据库系统提供了对并发操作的控制功能，对多用户的并发操作予以控制和协调，保证多个用户的操作不相互干扰。

1.1.3 数据库系统的结构

1972 年，美国国家标准协会计算机与信息处理委员会 ANSI/X3 成立了一个 DBMS 研究组，试图规定一个标准化的数据库系统结构，规定总体结构、标准化数据库系统的特征，包括数据库系统的接口和各部分所提供的功能，这就是有名的 SPARC (Standards Planning And Requirement Committee) 分级结构。这三级结构以内模式、概念模式和外模式三个层次来描述数据库。

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名目繁多，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环境之上、各有不同的存储结构，但数据库系统在总的体系结构上都具有三级模式的结构特征。

它们之间的联系经过两次转换，把用户所看到的数据变成计算机存储的数据，即三级模式两级映像。如图 1-3 所示。

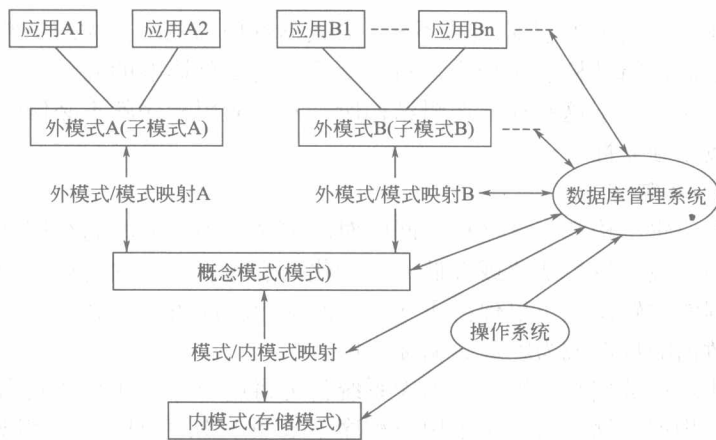


图 1-3 数据库的三级体系结构

1.1.3.1 三级模式

(1) 外模式

外模式也称子模式或用户模式，它是用户（包括应用程序员和最终用户）看见和使用的局部数据的逻辑结构和特征的描述，是用户的数据视图，是与某一应用有关的数据的逻辑表示。一个应用只能启动一个外模式，一个外模式可以为多个应用启用，如图中的外模式 A 被应用 A1 和应用 A2 启用。对于同一个对象，因不同的需求，使用不同的程序设计语言，不同的用户外模式的描述可能各不相同，同一数据在外模式中的结构、类型、长度、保密级别都可能不同。

外模式属于模式的子集。数据库系统提供外模式数据定义语言（Data Definition Language, DDL），用外模式 DDL 写出的一个用户数据视图的逻辑定义的全部语句称为此用户的外模式。

(2) 概念模式

概念模式简称为模式，是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图。

概念模式不同于外模式，它与具体的应用程序及高级语言无关，也不同于内模式，比内模式抽象，不涉及数据的物理存储结构和硬件环境。数据库系统提供概念模式描述语言（模式 DDL）来严格地定义模式所包含的内容，用模式 DDL 写出的一种数据库逻辑定义的全部语句，称为数据库的模式。模式是对数据库结构的一种描述，而不是数据库本身，它是装配数据库的一个框架。

(3) 内模式

内模式又称为存储模式，是全部数据在数据库系统内部的表示或底层描述，即为数据的物理结构和存储方法的描述。

内模式具体描述了数据如何组织并存入外部存储器上，一般由系统程序员根据计算机系统的软硬件配置决定数据存储方法，并编制程序实现存取，因而内模式对用户是透明的。内模式是用内模式描述语言（内模式 DDL）来描述或定义。

1.1.3.2 二级映像

(1) 外模式/模式映像

它定义了某个外模式和模式之间的对应关系，是数据的全局逻辑结构和数据的局部逻辑结构之间的映像，这些映像定义通常包含在各自的外模式中。当系统要求改变模式时，可改

变外模式/模式的映射关系而保持外模式不变。如数据管理的范围扩大或某些管理的要求发生改变后，数据的全局逻辑结构发生变化，对不受该全局变化影响的局部而言，最多改变外模式与模式之间的映像，基于这些局部逻辑结构所开发的应用程序就不必修改。这种特性称为用户数据的逻辑数据独立性。

(2) 模式/内模式映像

它定义了数据逻辑结构和存储结构之间的对应关系，当数据库的存储结构发生了改变，如存储数据库的硬件设备发生变化或存储方法变化，引起内模式的变化，由于模式和内模式之间的映像使数据的逻辑结构可以保持不变，因此应用程序可以不必修改。这种全局的逻辑数据独立于物理数据的特性成为物理数据独立性。

由于有了上述两种数据独立性，数据库系统就可将用户数据和物理数据结构完全分开，使用户避免烦琐的物理存储细节。由于用户程序不依赖于物理数据，也就减少了应用程序开发和维护的难度。

1.2 数据模型

数据库需要根据应用系统中数据的性质、内在联系，按照管理的要求来设计和组织。人们把客观存在的事物以数据的形式存储到计算机中，经历了对现实生活中事物特性的认识、概念化到计算机数据库里的具体表示的逐级抽象过程。

1.2.1 实体相关概念

1.2.1.1 实体

客观存在并且可以相互区别的事物称为实体。实体可以是实在的事物，也可以是抽象事件。例如：商品、客户等属于实体，它们是实际事物，而销售、订购、比赛等活动也是实体，它们是比较抽象的事件。

1.2.1.2 实体的属性

描述实体的特征称为属性。例如，学生实用体（学号、姓名、性别、出生日期、政治面貌）等若干属性来描述。

1.2.1.3 实体集和实体型

属性值的集合表示一个实体，而属性的集合表示一种实体的类型称为实体型。同类型的实体的集合称为实体集。

例如，在学生实体集当中，（060201，吴大伟，男，88/06/20，团员）表征学生名册中的一个具体人。

在 Visual FoxPro 中，用“表”来存放同一类实体，即实体集。例如学生登记表。Visual FoxPro 中一个“表”包含若干个字段，“表”中所包含的“字段”就是实体的属性。字段值的集合组成表中的一条记录，代表一个具体的实体，即每一条记录表示一个实体。

1.2.2 实体间联系及联系的种类

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。如，一个学生可以有多个任课老师；一个任课老师可以教多个学生。实体间联系的种类是指一个实体型中可能出现的每一个实体与另一个实体型中多少个具体实体存在联系。两个实体间的联系可以归结为三种类型。

(1) 一对一联系

一对一联系是指实体集 A 中的每一个实体与实体集 B 中的一个实体对应，反之亦然。记为 1:1。例如，学校和校长两个实体型，不包括副校长的情况下，学校和校长之间存在一对一的联系。

(2) 一对多联系

实体集 A 中的每一个实体与实体集 B 中的多个实体对应，反之不然。记为 1:n。例如班级和学生两个实体型，一个班级可以有多名学生，而一名学生只属于一个班级。

一对多联系是最普遍的联系。也可以把一对一的联系看作一对多联系的一个特殊情况。

(3) 多对多联系

实体集 A 中的每一个实体与实体集 B 中的多个实体对应，反之亦然。记为 m:n。例如，学生和课程两个实体型，一个学生可以选修多门课程，一个课程由多个学生选修。因此，学生和课程间存在多对多的联系。

1.2.3 实体联系的表示方法

E-R 图又被称为实体-联系图，它提供了表示实体、属性和联系的方法，用来描述现实世界的概念模型。

构成 E-R 图的基本要素是实体、属性和联系，其表示方法如下。

① 实体：用矩形表示，矩形框内写明实体名；

② 属性：用椭圆形表示，椭圆形框内写明属性的名称并用无向边将其与相应的实体连接起来；

③ 联系：用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体连接起来，同时在无向边旁标上联系的类型（1:1，1:n 或 m:n）。

用 E-R 图表示的教学实体模型如图 1-4 所示。

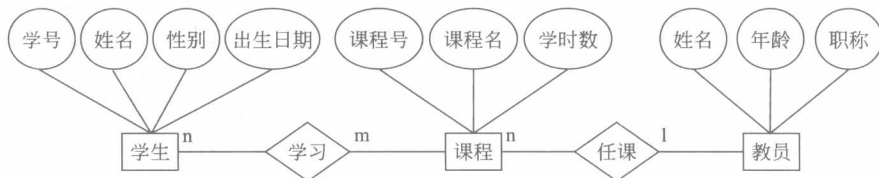


图 1-4 教学实体 E-R 图

1.2.4 数据模型简介

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有一定的结构，这种结构用数据模型来表示。数据库不仅管理数据本身，而且要使用数据模型表示出数据之间的联系。可见，数据模型是数据库管理系统用来表示实体及实体间联系的方法。一个具体的数据模型应当正确地反映出数据之间存在的整体逻辑关系。数据模型应满足三方面的要求：能比较真实地模拟现实世界；容易为人所理解；便于在计算机上实现。

任何一个数据库管理系统都是基于某种数据模型的。数据库管理系统所支持的数据模型分为三种：层次模型、网状模型、关系模型。因此，使用支持某种特定数据模型的数据库管理系统开发出来的应用系统相应地称为层次数据库系统、网状数据库系统、关系数据库系统。

关系模型对数据库的理论和实践产生很大的影响，成为当前最流行的数据库模型。为了使读者对数据模型有一个全面的认识，进而更深刻地理解关系模型，这里先对层次模型和网状模型作一个简单的介绍，再详细地介绍关系数据模型。

1.2.4.1 层次模型

层次数据模型用树状结构来表示各类实体及实体间的联系。满足下面两个条件的基本层次联系的集合为层次模型：

- ① 有且只有一个结点没有父结点，这个结点称为根结点；
- ② 根结点以外的其他结点有且只有一个双亲结点。

在层次模型中，每个结点表示一个实体集，实体集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。同一父结点的子结点称为兄弟结点，没有子结点的结点称为叶结点。若需要子结点有很多父结点或不同父结点的子结点间联系，则无法使用层次模式，必须改用其他模式。

层次数据模型结构的优点是：结构简单，易于操作；从上而下寻找数据容易；与日常生活的数据类型相似。其缺点是：寻找非直系的结点非常麻烦，必须通过多个父结点由下而上，再向下寻找，搜寻的效率太低。

图 1-5 给出一个层次模型的例子。其中“专业系”为根结点；“教研室”和“课程”为兄弟结点，是“专业系”的子结点；“教师”是“教研室”的子结点；“教师”和“课程”为叶结点。

1.2.4.2 网状模型

用网状结构表示实体及其之间的联系的模式称为网状模型。网状模型是层次模型的扩张，去掉了层次模型的两个限制：允许结点有多于一个的父结点；可以有一个以上的结点没有父结点。因此，网状模型可以方便地表示各种类型的联系。

图 1-6 给出了一个简单的网状模型。每一个联系都代表实体之间一对多的联系，系统用单向或双向环形链接指针来具体实现这种联系。

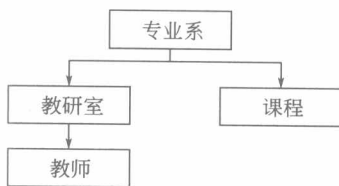


图 1-5 层次模型示例

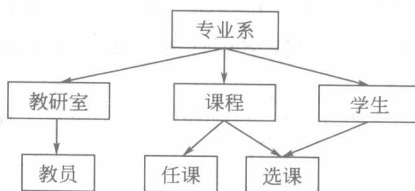


图 1-6 网状模型示例

网状模型的优点是表示多对多的联系具有很大的灵活性，这种灵活性是以数据结构复杂化为代价的。缺点是路径太多，当增加或删除数据时，牵动的相关数据很多，重建和维护数据比较麻烦。

网状模型与层次模型在本质上是一样的。从逻辑上看，它们都是用结点表示实体，用有向边表示实体间的联系，实体和联系用不同的方法来表示；从物理上看，每一个结点都是一个存储记录，用链接指针来实现记录之间的联系。这种用指针将所有数据记录都“捆绑”在一起的特点，使得层次模型和网状模型存在难以实现系统修改与扩充等缺陷。

1.2.4.3 关系模型

针对层次模型和网状模型的这些缺陷，20 世纪 70 年代初提出了关系模型。关系模型是用二维表结构来表示实体以及实体之间联系的模型。在关系模型中，操作的对象和结果都是二维表，这种二维表就是关系。

关系模型与层次模型、网状模型的本质区别在于数据描述的一致性，模型概念单一。在关系型数据库中，每一个关系都是一个二维表，无论实体本身还是实体间的联系均用称为“关系”的二维表来表示，使得描述实体的数据本身能够自然地反映它们之间的联系。而传统的

层次和网状模型是使用链接指针来存储和体现联系的。支持关系模型的数据库管理系统称为关系数据库管理系统，Visual FoxPro 系统就是一种关系数据库管理系统。

1.3 关系数据库

1.3.1 关系模型

1.3.1.1 关系术语

在用户观点下，关系模型中数据的逻辑结构是一张二维表，它由行和列组成。

以表 1-1 学生基本情况表为例，介绍关系模型中的一些术语。

表 1-1 学生基本情况表

学号	姓名	性别	出生年月	政治面貌	籍贯	入学时间
33006101	赵玲	女	1986.8.6	团员	黑龙江省哈尔滨	2006.9.6
33006102	王刚	男	1985.6.5	党员	四川省自贡市	2006.9.6
33006104	李广	男	1986.3.12	团员	山东省菏泽市	2006.9.6
.....						

① 关系 (Relation)：一个关系对应通常说的是一张二维表。例如，表 1-1 中这张学生基本情况表就是一个关系，可以命名为学生关系。

② 元组 (Tuple)：表中的一行即为一个元组。例如，表 1-1 有 3 行，对应 3 个元组。

③ 属性 (Attribute)：表中的一列即为一个属性，给每一个属性起一个名称即属性名。例如，表 1-1 中有 7 列，对应的 7 个属性分别为学号、姓名、性别、出生年月、政治面貌、籍贯、入学时间。

④ 关键字：属性或属性的组合，其值能够唯一地标识一个元组。例如，表 1-1 中学号可以唯一确定一个学生，也就可以作为本关系的关键字。

⑤ 外部关键字：如果表中的一个字段不是本表的主关键字或候选关键字，而是另外一个表的主关键字或候选关键字，这个字段（属性）就称为外部关键字。

⑥ 域 (Domain)：属性的取值范围。例如，性别的域是（男，女）。

⑦ 分量：元组中一个属性值。例如表 1-1 中第一个元组在学号属性上的取值为 33006101，则 33006101 就是第一个元组的一个分量。

⑧ 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）

例如，表 1-1 的学生关系可描述为：学生（学号，姓名，性别，出生年月，政治面貌，籍贯，入学时间）。在关系模型中，实体以及实体间的联系都是用关系来表示的。

从集合论的观点来定义关系，可以将关系定义为元组的集合。关系模式是命名的属性集合。元组是属性值的集合。一个具体的关系模型是若干个有联系的关系模式的集合。

1.3.1.2 关系的特点

关系模型看起来简单，但并不能把日常手工管理所用的各种表格，按照一张表一个关系直接存放到数据库系统中。在关系模型中对关系有一定的要求，关系必须具有以下特点。

① 关系必须规范化。规范化是指关系模型中每个关系模式都必须满足一定的要求，最基本的要求是关系必须是一张二维表，每个属性值必须是不可分割的最小数据单元，即表中不能再包含表。