

高校非计算机专业计算机基础教育改革型教材

Visual FoxPro

教程(第4版)

GAOXIAOFEIJISUANJIZHUANYE
JISUANJICHUJIAOYUGAIGEXINGJIAOCAI

©卢雪松 主编



东南大学出版社
SOUTHEAST UNIVERSITY PRESS

高校非计算机专业计算机基础教育改革型教材

Visual FoxPro 教程

(第4版)

卢雪松 主编



东南大学出版社

• 南京 •

内 容 提 要

本书融数据库基本理论、数据库管理系统和数据库应用程序开发为一体,既注重基础理论,又强调实践应用。在突出面向对象概念的同时又兼顾了传统的面向过程的程序设计方法。本书图文并茂、示例翔实,便于读者阅读和上机操作。

本书可供各类高等院校本、专科学生使用,也可供参加江苏省计算机等级考试和全国计算机等级考试的考生复习迎考。

图书在版编目(CIP)数据

Visual FoxPro 教程 / 卢雪松主编. —4 版. —南京:东南大学出版社, 2012. 11

ISBN 978-7-5641-3841-7

I. ①V… II. ①卢… III. ①关系数据库系统—数据库管理系统—高等学校—教材 IV. TP311.138

中国版本图书馆 CIP 数据核字 (2012) 第 251956 号

Visual FoxPro 教程(第 4 版)

主 编 卢雪松
责任编辑 张 煦
装帧设计 毕 真

出版发行 东南大学出版社
出 版 人 江建中
社 址 江苏省南京市四牌楼 2 号(210096)
经 销 江苏省新华书店
印 刷 南京京新印刷厂
版 次 2012 年 11 月第 4 版 2012 年 11 月第 1 次印刷
开 本 787×1092 mm 1/16
印 张 20
字 数 512 千字
书 号 ISBN 978-7-5641-3841-7
定 价 32.00 元

凡因印装质量问题,请直接向东大出版社读者服务部调换。电话:025—83791830

再版前言

随着计算机及其应用的推广普及,各行各业纷纷建立了以数据库为核心的信息系统。数据库技术在当今信息社会的地位和作用显得尤为重要。Visual FoxPro 是当前微型计算机上普遍使用的一种可视化的、面向对象的关系型数据库管理系统。

Visual FoxPro 程序的运行既可以采用解释方式、又可以采用编译方式。其自带的语言系统涵盖了面向过程的程序设计、面向问题的程序设计、面向对象的程序设计以及计算机自动编程等程序设计语言的发展过程。运用 Visual FoxPro 开发一个应用系统还运用到了软件工程的思想和方法。所有这些恰好对其前趋课程《计算机应用基础》中的有关概念进行了进一步的诠释和印证,更有利于读者理解和掌握计算机程序设计的概念和方法,这是其他程序设计语言无可比拟的。

随着信息技术的快速发展和计算机的高度普及,社会对新型人才的计算机应用能力与水平提出了更高的要求,而运用计算机进行数据处理是新型人才必须具备的基本能力。Visual FoxPro 具有强大的数据处理能力,能满足非理工类专业计算机应用的需求。Visual FoxPro 操作性强,数据处理的概念直观明了,相对于算法要求较强的其他程序设计语言而言,更适合非理工类专业学生修读。

本书融数据库基本理论、数据库管理系统和数据库应用程序开发为一体,既注重基础理论,又强调实践应用。与一般教材不同的是,本书即突出了面向对象的概念,同时又兼顾到了传统的面向过程的程序设计方法。本书图文并茂、示例翔实,便于读者阅读和上机操作。

本书在总结作者多年的 VFP 教学经验的基础上,根据《中国高等院校计算机基础教育课程体系 2008》的精神和计算机等级考试大纲的要求组织编写,可供各类高等院校本、专科学生使用,也可供参加江苏省计算机等级考试和全国计算机等级考试的考生复习迎考。

本书由卢雪松主编。参加编写工作的有:杨晓秋(第 1 章)、徐晶(第 2 章)、方益明(第 3~4 章)、张平(第 5~6 章)、潘钧(第 7 章)、卢雪松(第 8~9 章)、沈启坤(第 10~11 章)、楚红(第 12 章)。

在本书的编写过程中,参考了许多同类书籍,同时得到了扬州大学教务处有关同志的关心和支持,在此一并表示衷心的感谢。

本教材由扬州大学出版基金资助。

由于编者水平有限,不妥之处敬请广大读者批评指正。

编 者

2012 年 10 月于扬州大学

目 录

第 1 章 数据库系统及 Visual FoxPro 概述	(1)
1.1 数据库系统的基本概念	(1)
1.1.1 数据管理技术的发展	(1)
1.1.2 数据库的特点	(2)
1.1.3 数据库系统	(3)
1.1.4 数据库系统体系结构	(4)
1.2 数据模型	(5)
1.2.1 概念模型	(5)
1.2.2 结构数据模型	(7)
1.3 关系数据库理论基础	(8)
1.3.1 关系模型的数据结构和基本术语	(8)
1.3.2 关系模型的完整性约束条件	(9)
1.3.3 关系数据操纵与关系代数	(10)
1.3.4 关系规范化理论	(11)
1.4 关系数据库系统简介	(13)
1.4.1 Visual FoxPro 6.0 简介	(13)
1.4.2 VFP 集成环境的使用	(14)
1.4.3 VFP 命令	(15)
1.4.4 文件类型与创建	(16)
1.5 项目管理器	(17)
1.5.1 创建、打开项目	(17)
1.5.2 项目管理器的使用	(18)
1.5.3 定制项目管理器	(20)
练习题	(22)
 第 2 章 Visual FoxPro 的数据类型	(24)
2.1 数据类型	(24)
2.2 数据存储	(26)
2.2.1 VFP 中的命名规则	(27)
2.2.2 常量	(27)
2.2.3 变量	(28)

2.3 函数.....	(29)
2.3.1 数据类型函数.....	(29)
2.3.2 数据库类函数.....	(39)
2.3.3 其他类函数.....	(43)
2.4 表达式.....	(46)
2.4.1 字符表达式.....	(46)
2.4.2 日期表达式.....	(47)
2.4.3 算术表达式.....	(47)
2.4.4 关系表达式.....	(48)
2.4.5 逻辑表达式.....	(49)
2.4.6 名称表达式.....	(50)
2.4.7 宏替换.....	(51)
2.4.8 NULL 值	(51)
练习题	(52)
 第 3 章 表的使用	(54)
3.1 表结构的创建和使用.....	(54)
3.1.1 字段的基本属性.....	(54)
3.1.2 创建新表.....	(55)
3.1.3 表结构的修改.....	(56)
3.2 表记录的编辑修改.....	(57)
3.2.1 记录的追加.....	(57)
3.2.2 查看表的内容.....	(58)
3.2.3 记录的定位.....	(59)
3.2.4 修改记录.....	(61)
3.2.5 删除记录.....	(63)
3.2.6 筛选记录.....	(64)
3.2.7 筛选字段.....	(64)
3.3 表的使用.....	(65)
3.3.1 工作区.....	(65)
3.3.2 打开和关闭表.....	(66)
3.3.3 表的独占与共享使用.....	(67)
3.3.4 利用缓冲访问表中的数据.....	(68)
3.4 表的索引.....	(69)
3.4.1 索引文件的种类.....	(69)
3.4.2 索引文件的创建.....	(70)
3.4.3 索引的修改和删除.....	(71)
3.4.4 索引的使用.....	(72)

3.5 建立表之间的临时关系	(74)
3.5.1 临时关系	(74)
3.5.2 临时关系的建立	(74)
3.5.3 解除临时关系	(75)
练习题	(76)
第4章 数据库的创建和使用	(78)
4.1 VFP 数据库	(78)
4.2 数据库的设计过程	(78)
4.2.1 确定建立数据库的目的	(78)
4.2.2 确定需要的表	(79)
4.2.3 确定表中的字段	(79)
4.2.4 确定表之间的关系	(80)
4.2.5 设计的改进	(81)
4.3 建立数据库文件	(81)
4.4 数据字典	(83)
4.5 数据库表字段的扩展属性	(84)
4.5.1 字段的显示属性	(84)
4.5.2 字段的有效性规则	(85)
4.6 数据库表的表属性	(86)
4.6.1 长表名	(86)
4.6.2 表记录的有效性规则	(87)
4.6.3 表的触发器(Trigger)	(88)
4.6.4 约束机制及其激活时机	(89)
4.7 表之间的关系	(89)
4.8 参照完整性	(90)
4.8.1 参照完整性概念	(90)
4.8.2 设置参照完整性	(91)
4.9 使用多个数据库	(92)
4.9.1 打开多个数据库	(92)
4.9.2 设置当前数据库	(92)
4.9.3 数据库中表的使用	(93)
4.9.4 关闭数据库	(93)
4.10 查看和设置数据库的属性	(93)
4.10.1 DBGETPROP()函数	(93)
4.10.2 DBSETPROP()函数	(94)
练习题	(95)

第 5 章 查询和视图	(97)
5.1 SQL 语言和查询技术	(97)
5.1.1 SQL 语言概述	(97)
5.1.2 SELECT - SQL 查询命令	(97)
5.1.3 SELECT - SQL 应用举例	(98)
5.2 查询的创建和使用	(103)
5.2.1 查询的本质	(103)
5.2.2 查询设计器的使用	(103)
5.2.3 查询向导的使用	(113)
5.3 视图的创建和使用	(115)
5.3.1 视图的实质	(115)
5.3.2 创建本地视图	(116)
5.3.3 创建参数化视图	(117)
5.3.4 使用视图更新基表数据	(118)
5.3.5 视图的操作	(119)
练习题	(122)
第 6 章 报表和标签	(125)
6.1 报表类型	(125)
6.2 创建报表	(126)
6.2.1 用报表向导创建报表	(126)
6.2.2 创建快速报表	(128)
6.2.3 利用报表设计器创建报表	(128)
6.2.4 创建一对多报表	(135)
6.3 报表的预览与打印	(136)
6.3.1 页面设置	(136)
6.3.2 打印预览	(136)
6.3.3 打印报表	(136)
6.4 设计标签	(137)
6.4.1 标签类型	(137)
6.4.2 标签向导	(137)
6.4.3 标签设计器	(137)
练习题	(138)
第 7 章 VFP 程序设计基础	(139)
7.1 创建、修改和运行应用程序	(139)
7.1.1 创建应用程序	(139)
7.1.2 修改程序	(140)

7.1.3 运行程序	(140)
7.2 Visual FoxPro 基本语法结构	(140)
7.2.1 程序中常用的命令	(140)
7.2.2 顺序结构程序设计	(142)
7.2.3 分支结构程序设计	(143)
7.2.4 循环结构程序设计	(145)
7.3 数组	(150)
7.3.1 数组的声明	(150)
7.3.2 数组元素的赋值	(151)
7.3.3 数组与表之间的传送数据	(151)
7.4 过程和用户自定义函数	(152)
7.5 变量的作用域	(157)
7.6 综合应用	(157)
练习题	(163)
第 8 章 表单	(167)
8.1 创建表单	(167)
8.1.1 用表单向导创建表单	(167)
8.1.2 用表单设计器创建表单	(170)
8.1.3 创建快速表单	(175)
8.1.4 创建表单集	(176)
8.2 对象的属性、事件和方法	(177)
8.3 常用的事件和方法	(180)
8.4 添加属性和方法程序	(186)
8.5 多文档界面与表单类型	(188)
练习题	(189)
第 9 章 控件	(191)
9.1 标签	(191)
9.2 命令按钮	(192)
9.3 文本框	(194)
9.4 编辑框	(198)
9.5 微调框	(199)
9.6 列表框	(200)
9.7 组合框	(202)
9.8 复选框	(203)
9.9 选项按钮组	(204)
9.10 命令按钮组	(205)

9.11	表格	(207)
9.12	页框	(209)
9.13	计时器	(211)
9.14	图形控件	(212)
9.15	图像控件	(214)
9.16	OLE 控件	(215)
	练习题	(216)
第 10 章	类	(218)
10.1	VFP 的类层次	(218)
10.1.1	类	(218)
10.1.2	VFP 的类及其层次	(219)
10.2	类的创建与修改	(221)
10.2.1	何时使用类	(221)
10.2.2	子类的创建	(221)
10.2.3	类的修改	(228)
10.3	扩展基类	(228)
10.4	为类指定外观	(229)
10.5	类的应用	(231)
10.5.1	添加类到表单	(231)
10.5.2	覆盖默认属性设置	(232)
10.5.3	调用父类方法代码	(233)
10.5.4	给子类增加功能	(233)
10.5.5	在嵌套容器中向对象添加功能	(234)
10.5.6	调用通用的类代码	(234)
10.5.7	防止基类代码被执行	(234)
	练习题	(235)
第 11 章	菜单和工具栏设计	(237)
11.1	设计菜单	(237)
11.1.1	菜单设计器的使用	(237)
11.1.2	使用菜单设计器创建普通菜单系统	(240)
11.1.3	使用菜单设计器创建快捷菜单系统	(247)
11.1.4	调试与运行菜单程序	(248)
11.1.5	使用菜单设计器创建 SDI 菜单系统	(249)
11.2	创建自定义工具栏	(249)
11.2.1	定制 VFP 工具栏	(250)
11.2.2	创建自定义工具栏	(251)

练习题	(254)
第 12 章 数据库应用系统开发	(256)
12.1 开发 VFP 数据库应用系统的一般步骤	(256)
12.1.1 需求分析	(256)
12.1.2 数据库设计	(257)
12.1.3 应用程序设计	(258)
12.1.4 软件测试	(260)
12.1.5 应用程序发布	(260)
12.1.6 系统运行与维护	(261)
12.2 教学管理系统的开发	(261)
12.2.1 需求分析	(261)
12.2.2 数据库设计	(261)
12.2.3 应用程序设计	(263)
12.2.4 系统测试	(276)
12.3 应用程序的管理和发布	(276)
12.3.1 管理应用程序	(276)
12.3.2 应用程序发布	(281)
附录 1 ASCII 码表	(287)
附录 2 VFP 常用函数	(288)
附录 3 VFP 常用命令	(293)
附录 4 控件的主要属性、事件和方法	(300)
参考文献	(308)

第 1 章 数据库系统及 Visual FoxPro 概述

数据库是数据处理的重要工具,也是管理信息系统(MIS)、办公自动化(OA)系统和决策支持系统(DSS)等应用系统的核心部分。它能够有效、合理地存储各种数据,为有关应用准确、快速地提供有用的信息。数据库技术是计算机领域中最重要技术之一,其应用已渗透到人类社会各个领域,并正在改变着人们的生活方式和工作方式。因此,我们有必要学习和掌握数据库系统的原理和技术,用以解决各种计算机应用中的实际问题。

1.1 数据库系统的基本概念

1.1.1 数据管理技术的发展

数据管理技术是指对数据进行分类、组织、编码、存储、检索和维护的技术。在计算机环境下,数据管理技术经历了从低级到高级的 3 个发展阶段:

1. 人工管理阶段(20 世纪 50 年代中期以前)

该阶段尚无统一的数据管理软件,对数据的管理完全由各个程序员在其程序中进行。程序员在编制其课题程序时,必须考虑数据的逻辑定义和组织、数据存放的存储设备、物理存储方式和地址分配,并通过物理地址来存取数据。表示处理流程的程序与其处理对象——数据相互结合成一个整体,两者紧密地相互依赖。数据的管理仍然是分散的,计算机在数据管理中还没有发挥应有的作用。因此,严重地影响了计算机的使用效率。

2. 文件系统管理阶段(20 世纪 50 年代后期至 60 年代中期)

在这一阶段,由于计算机技术的发展,出现了磁带、磁鼓和磁盘等容量较大的存储设备。软件方面有了操作系统,计算机的应用范围也由科学计算领域扩展到数据处理领域。

这一阶段的特点是:

① 数据可以以操作系统的文件形式长期保存在计算机中,文件的组织方式由顺序文件逐步发展到随机文件。

② 操作系统的文件管理系统提供了对数据的输入和输出操作接口,进而提供数据存取方法。

③ 一个应用程序可以使用多个文件,一个文件可为多个应用程序使用,数据可以共享。

④ 数据仍然是面向应用的,文件之间彼此孤立,不能反映数据之间的联系,因而仍存在数据的大量冗余和不一致性。

3. 数据库系统管理阶段(20 世纪 60 年代后期开始)

针对数据的文件管理方式存在的上述缺点,计算机软件工作者经过长期不懈的努力,提出了数据库的概念。数据库技术为数据管理提供了一种较完善的高级管理方式,它克服了文件系统方式下分散管理的弱点,对所有的数据实行统一、集中的管理,使数据的存储独立于使用它的程序,从而实现数据共享。这样也就克服了文件管理方式的缺点。如图 1-1 所示。

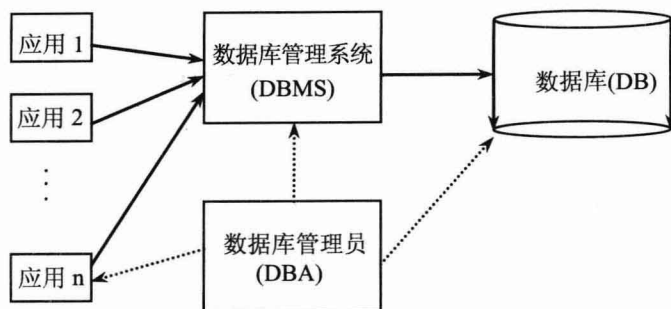


图 1-1 数据库系统

1.1.2 数据库的特点

现在数据库已经成为各种计算机应用系统的核心部分。其所以如此,是因为数据库有着许多独特的优点,它们主要是:

一、数据被集成化

所谓集成化就是按照一定的数据模型来组织和存放数据,故而又称数据的结构化。数据的集成化或结构化是数据库系统管理和文件系统管理之间的一个本质区别,是实现对数据的集中控制和减少数据冗余的前提和保证。

由于数据库是从一个组织的全部应用来全盘考虑并集成数据的结构,所以数据库中的数据不再是面向个别应用,而是面向系统。各个不同的应用系统所需的数据只是整体模型的一个子集。

二、具有数据独立性

所谓数据独立性就是数据与应用程序之间不存在相互依赖关系,也就是数据的逻辑结构、存储结构和存取方法等,不因应用程序的修改而修改,反之亦然。这是数据库系统与文件系统之间的另一个重要区别,也是使得数据库系统结构复杂的一个重要原因。数据独立性分为物理独立性和逻辑独立性两级。

1. 物理独立性

即数据的物理结构(或存储结构)的改变,如物理存储设备的更换、物理存储位置的变更、存取方法的改变等等,不影响数据库的逻辑结构,从而不致引起应用程序的修改。

2. 逻辑独立性

数据库总体逻辑结构的改变,如修改数据的定义、增加新的数据类型、改变数据间的联系等等,无需修改原来的应用程序,这就是数据的逻辑独立性。遗憾的是,到目前为止,数据的逻辑独立性还没有完全彻底地实现。

三、实现数据共享

数据共享是促成发展数据库技术的重要原因之一,也是数据库技术先进性的一个重要体现。数据库中数据的共享性体现在两个方面:

(1) 数据库中的数据可供多个应用程序用于不同的目的,每个应用程序各有其自己的局部数据逻辑结构。数据库中的数据不但可供现有的各个应用程序共享,还可开发新的应用程序而无需附加新的数据,实现新、老应用程序共享数据库中的数据。

(2) 数据库可以为批处理用户和终端的多个用户同时共享。

四、数据的冗余度小

数据冗余是指相同的数据重复出现。在非数据库系统中,每个应用有它自己的文件,从而造成了存储数据的大量重复。大量的数据冗余带来的弊病是:

(1) 浪费了存储空间。

(2) 为了更新某些冗余副本,保持数据的一致性,必须执行多次更新操作,从而增加不必要的更新阶段和机器时间。

(3) 由于数据的不同副本可能处于不同的更新阶段,从而可能给出不一致的信息。

数据库则是从整体观点来组织和存储数据的,数据是集成化、结构化的。它是为多种应用所共享的,从而大大减少了数据冗余。

五、具有统一的数据控制功能

由于数据库是多用户的共享资源,而计算机的共享一般是并发的,即多用户同时使用数据库,因此,数据的安全性和可靠性是一个数据库系统能否实用的关键问题,所以必须采用有效措施保护数据。

1. 安全性控制

这里的安全性,主要是指数据的保密性控制,例如用户的身份号,验明身份后才能进入系统,对某些特定的数据限定使用权限;对不同的操作采用不同的保护级别(如有的用户只允许查询,有的用户有修改权);采用密码方式存放数据等。

2. 完整性控制

指利用某些条件判断来排除不正确的数据,它包括正确性(如数值型数据中不应含有字母)、有效性(如表示月份必须为1至12的正整数)、相容性(如一人的工资在两个地方不应出现不同的数值,数据不相容主要是由于数据冗余引起的)的控制。

3. 并发控制

避免一个用户读出另一个用户正在修改的错误数据。

4. 故障的发现与恢复

防止硬件和软件故障以及用户操作上的错误所造成的对数据的破坏。

1.1.3 数据库系统

数据库系统(Data Base System, 简称 DBS)是指在计算机环境下引进数据库技术后构成的整个计算机系统。除计算机软、硬件环境外,数据库系统包括以下几个组成部分:

一、数据库(Data Base, 简称 DB)

数据库被定义为是在计算机存储设备上合理存放的、互相关联的数据集合。

二、数据库管理系统(Data Base Management System, 简称 DBMS)

数据库管理系统是数据库系统中专门用于数据管理的软件,是用户与数据库的接口。提供有:

1. 数据库定义功能

提供数据描述语言(Data Defined Language, 简称 DDL)及其翻译程序,用于定义数据库结构(模式及模式间映射)、数据完整性和保密性约束等。

2. 数据库操作功能

提供数据操纵语言(Data Manipulation Language, 简称 DML)及其翻译程序,用于实现对数据库数据的查询、插入、更新和删除等操作。

数据操纵语言亦称数据库语言。一般有两种使用方式:一种是交互式命令语言,它语法简明,可以独立使用,称为自含型语言;另一种是嵌入到某种程序设计语言中,如 C 语言、FORTRAN 语言、COBOL 语言、PASCAL 语言等,称为宿主型语言。

3. 数据库运行和控制功能

包括数据安全性控制、数据完整性控制、多用户环境的并发控制等。

4. 数据库维护功能

包括数据库数据的载入、转储和恢复,数据库的维护和数据库的功能及性能分析和监测等。

三、应用程序

应用程序是数据库中特定用户的数据处理业务,利用 DBMS 支持的程序设计语言编写的程序。

四、人员

参与分析、设计、管理、维护和使用数据库中数据的人员都是数据库系统的组成部分,分析、设计、管理和使用数据库系统的人员主要是:

1. 数据库管理员(DataBase Administrator, 简称 DBA)

数据库管理员是管理和维护数据库系统正常运转的专职人员。具体的职责包括:①决定数据库中的数据内容和结构;②决定数据库的存储结构和存取策略;③定义数据的安全性要求和完整性约束条件;④监控数据库的使用和运行;⑤数据库的重构和重组。

2. 系统分析员

系统分析员是数据库系统建设期主要的参与人员,负责应用系统的需求分析和规范说明,确定系统的基本功能、数据库结构和应用程序的设计以及软、硬件配置,并组织整个系统的开发。

3. 应用程序员

应用程序员根据应用系统的功能需求负责设计和编写应用系统的程序模块,并参与对程序模块的测试。

4. 用户

这里用户是指最终用户(End User)。

目前,微型机到大型计算机都有数据库系统的应用。一般而言,大型机上的系统是多用户系统,系统提供的设施较完整,功能也较强。微型机上的系统大多是单用户系统,这种系统提供的设施往往是大型多用户系统的一个子集,即使设施相同,也没有大型多用户系统那么强的功能。

1.1.4 数据库系统体系结构

数据库的体系结构是数据库系统的一个总框架。尽管实际数据库软件产品种类繁多,使用的数据库语言各异,基础操作系统不同,采用的数据结构模型相差甚大,但是绝大多数数据库系统在总体结构上都具有三级模式的结构特征。数据库的三级模式结构由外模式、模式和内模式组成,如图 1-2 所示。

(1) 外模式:又称子模式或用户模式,是模式的子集,是数据的局部逻辑结构,也是数据库用户看到的数据视图。

(2) 模式:又称逻辑模式或概念模式,是数据库中全体数据的全局逻辑结构和特性的描

述,也是所有用户的公共数据视图。

(3) 内模式:又称存储模式,是数据在数据库系统中的内部表示,即数据的物理结构和存储方式的描述。

数据库系统的三级模式是对数据的三级抽象。为了实现 3 个抽象层次的转换,数据库系统在三级模式中提供了两次映像:外模式/模式映像和模式/内模式映像。所谓映像就是存在某种对应关系。

外模式到模式的映像,定义了外模式与模式之间的对应关系。模式到内模式的映像,定义了数据的逻辑结构和物理结构之间的对应关系。正是由于这两级映像,使数据库管理的数据具有两个层次的独立性:物理独立性和逻辑独立性。

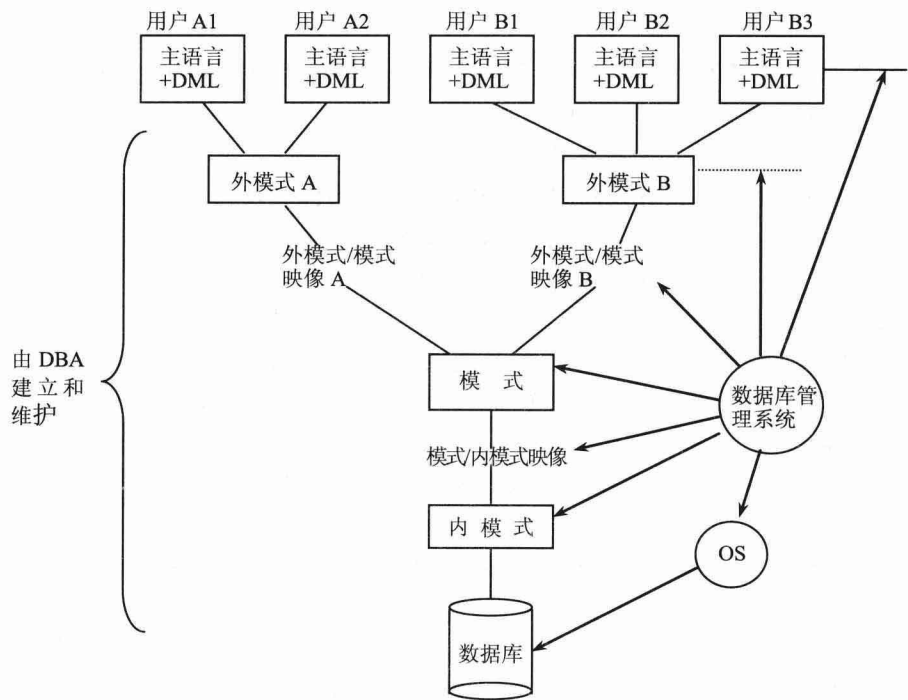


图 1-2 数据库系统的体系结构

1.2 数据模型

模型一词本身含意是现实世界事物的一种抽象,例如汽车模型、船舶模型等。

数据模型是数据库结构和语意的一种抽象,通俗地说,数据模型是现实世界事物及事物之间联系的数据表示,从数据组织形式而言,数据库的数据结构形式就是数据模型。

1.2.1 概念模型

从现实世界到数据世界,实际上经历了“现实世界”→“信息世界”→“数据世界”3 个世界的演变过程,信息世界是现实世界事物间的内在联系在人们头脑中的反映。其认识、理解、抽象和表达的结果,用概念模型来表示,又称“信息模型”。

一、几个术语

信息世界涉及的主要概念有:

1. 实体(Entity)

客观存在并可相互区分的事物。实体可以指实际的对象,也可以指某些概念。例如一个职工、一个学生、一个部门、一门课、一次定货等。

2. 属性(Attribute)

实体所具有的某一特性。一个实体可以由若干个属性来刻画。例如,某学生实体可以由学号、姓名、年龄、性别、班级等属性组成。(970103331,王平,21,男,计 9701)这些属性值组合起来表征了一个学生。

3. 关键字(Key)

唯一标识实体的属性集称为关键字,又称码、键。例如,学号是学生实体的关键字。

4. 域(Domain)

某个(些)属性的取值范围。例如,学号的域为 9 位数字,姓名的域为字符串集合,年龄的域为小于 35 的整数,性别的域为(男,女)。

5. 实体型(Entity type)与实体集(Entity Set)

具有相同属性的实体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、年龄、性别、班级)是一个实体型。同型实体的集合称为实体集,例如,全体学生就是一个实体集。

6. 联系(Relationship)

现实世界的事物之间是有联系的,这种联系必然要在信息世界中加以反映,一般存在两类联系:一是实体内部的联系,如组成实体的属性之间的联系;一是实体之间的联系。

二、联系的类型

这里我们讨论的是实体型之间的联系。两个实体型之间的联系可分为 3 类。

1. 一对一联系(1:1)

若对于实体集 A 中的每一个实体,实体集 B 中至多有一个实体与之联系,反之亦然,则称实体集 A 与实体集 B 具有一对一联系,记为 1:1。

2. 一对多联系(1:n)

若对于实体集 A 中的每一个实体,实体集 B 中有 $n(n \geq 0)$ 个实体与之联系,反之,对于实体集 B 中的每一个实体,实体集 A 中至多只有一个实体与之联系,则称实体集 A 与实体集 B 具有一对多的联系。记为 1:n。

3. 多对多联系(m:n)

若对于实体集 A 中的每一个实体,实体集 B 中有 $n(n \geq 0)$ 个实体与之联系。反过来,对于实体集 B 中的每一个实体,实体集 A 中也有 $m(m \geq 0)$ 个实体与之联系,则称实体集 A 与实体集 B 具有多对多联系。记为 m:n。

例如:一个部门有一个经理,而每个经理只在一个部门任职,则部门和经理之间具有一对一联系;一个部门有若干职工,而每个职工只在一个部门工作,则部门与职工之间是一对多的联系;一个项目有多个职工参加,而一个职工可以参加若干项目的工作,则项目和职工之间具有多对多的联系。

三、实体联系模型(Entity Relationship Model)

概念模型的表示方法最常用的是实体联系方法。这个方法是用 E-R 图来描述某一组织的概念模型,这种模型也称为实体联系模型(E-R 模型)。

在 E-R 图中,实体用“方框”表示,属性用“椭圆框”表示,联系用“菱形框”表示,再用线