

21世纪高等教育计算机规划教材

COMPUTER

Visual FoxPro 程序设计教程 (第3版)

Visual FoxPro Programing

■ 陈娟 王丽彬 主编

■ 谢晓艳 李爱华 李越 刘海莎 副主编

- 按知识点讲解、实例说明、课后练习的模式组织内容
- 剖析完整应用系统的开发过程，分解到各章实例中
- 教学网站提供课件、原始文件、操作演示等素材
- 覆盖全国计算机等级考试二级 VFP 的知识要点



21世纪高等教育计算机规划教材

COMPUTER

Visual FoxPro

程序设计教程 (第3版)

Visual FoxPro Programing

■ 陈娟 王丽彬 主编

■ 谢晓艳 李爱华 李越 刘海莎 副主编



人民邮电出版社

北京

Visual FoxPro程序设计教程 / 陈娟, 王丽彬主编

— 3版. — 北京: 人民邮电出版社, 2015.2

21世纪高等教育计算机规划教材

ISBN 978-7-115-38464-5

I. ①V… II. ①陈… ②王… III. ①关系数据库系统
— 高等学校—教材 IV. ①TP311.138

中国版本图书馆CIP数据核字(2015)第025740号

内 容 提 要

本书围绕“岳麓书院图书管理系统”实例,完整地描述了数据库应用系统开发的各个环节,将系统开发的具体步骤详细地贯穿到各个章节的例题中。

全书共10章,内容包括数据库的基础知识、数据与数据运算、数据库和数据表的操作、结构化查询语言、查询和视图、结构化程序设计、表单设计、报表设计、菜单设计、应用程序的生成与发布。每章均按知识点讲解、实例说明、课后练习的模式来组织教学内容。

本书可作为普通高等院校 Visual FoxPro 程序设计或相关课程的教材,也可作为全国计算机等级考试二级 Visual FoxPro 的培训或自学教材。

-
- ◆ 主 编 陈 娟 王丽彬
副 主 编 谢晓艳 李爱华 李 越 刘海莎
责任编辑 邹文波
责任印制 沈 蓉 彭志环
- ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市潮河印业有限公司印刷
- ◆ 开本: 787×1092 1/16
印张: 20.5
字数: 528千字
- 2015年2月第3版
2015年2月河北第1次印刷
-

定价: 45.00 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

第3版前言

Visual FoxPro 既是小型数据库管理系统的杰出代表,又是可视化的面向对象的集成开发工具。它以强大的性能、完整而丰富的工具、较高的处理速度、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。目前,我国很多高校都开设了 Visual FoxPro 程序设计课程。同时, Visual FoxPro 也是全国计算机等级考试的考试科目之一。

我们依据多年的实际教学经验和数据库应用系统的开发经验,在参考和借鉴了多本相关的同类教材后,精心编写了此书。本书具有以下特点。

1. 本书始终围绕着一个数据库应用系统的实例——“岳麓书院图书管理系统”来进行讲解。通过各个章节的例题,完整地描述了开发该系统的各个环节。包括建立项目文件,数据库的操作,表单、报表、菜单、主程序的设计,生成应用程序。

2. 各章按知识点讲解、实例说明、课后练习的模式来组织教学内容。每章先介绍基本概念和基本方法,然后通过大量实例对其进行说明。对于一些细节问题,再通过提示的方式来进行注解。同时,还配有习题和操作题,便于学生巩固所学知识。

3. 作者在个人网站(<http://www.teacherchen.cn>)上提供丰富的教学资源。网站提供有本书的多媒体课件,案例的 Flash 演示,例题、习题素材的下载。

本书的教学内容设计成以下3个部分。

第一部分是基础部分,由第1章、第2章组成。第1章是数据库的基础知识,着重讲解了数据库系统的组成和关系数据库的相关概念,简单介绍了 Visual FoxPro 软件和项目管理器的使用。第2章是数据与数据运算,主要介绍 Visual FoxPro 中常量、变量、运算符和常用函数的概念和使用。通过对该部分的学习,学生可了解数据库的相关理论,还能掌握如何根据条件书写正确的表达式,为后面的学习奠定基础。

第二部分是数据库的操作,由第3章、第4章、第5章组成。第3章是数据库和数据表的操作,详细说明了如何设计、建立、打开、关闭数据库,重点讲解了建立和维护数据表的方法,索引的建立和使用,以及如何在数据表之间建立关联和设置参照完整性。第4章是结构化查询语言,从数据查询、数据操纵和数据定义3个方面介绍 Visual FoxPro 所支持的 SQL 语句。第5章是查询和视图,主要介绍查询和视图的概念、建立和使用,比较了两者的异同。通过对该部分的学习,学生可以掌握如何在 Visual FoxPro 环境中建立和管理数据库。

第三部分是程序设计,由第6章、第7章、第8章、第9章、第10章组成。第6章是结构化程序设计,主要介绍由顺序、选择、循环3种基本结构所构成的传统的程序文件。第7章是表单设计,简单介绍了面向对象的若干基本概念,以及如何通过表单向导和表单设计器来建立表单。其中,详细讲解了一些常用表单控件的

使用。此外,还介绍了表单之间的相互调用和使用自定义类来优化表单。第8章是报表设计,介绍了如何通过报表向导和报表设计器来设计报表。第9章是菜单设计,介绍了下拉式菜单和快捷菜单的设计。第10章是应用程序的生成与发布,介绍了主文件的建立,及如何把项目管理器的各个组件连编成一个完整的应用程序。通过对该部分的学习,学生能了解面向对象的相关概念,掌握程序文件、表单、报表、菜单的设计。

与本书配套的还有实践教材《Visual FoxPro 程序设计实践教程(第3版)》,包含上机指导、习题、样卷和附录4个部分。通过该书的13个上机实验,能够进一步提高学生的实际操作能力,加强对所学理论知识的感性认识。

本书可作为普通高等院校 Visual FoxPro 程序设计或相关课程的教材,也可作为全国计算机等级考试二级 Visual FoxPro 的培训或自学教材。

本书由陈娟、王丽彬担任主编,谢晓艳、李爱华、李越、刘海莎担任副主编。其中,陈娟编写了第1章、第10章,王丽彬编写了第2章、第3章,谢晓艳编写了第4章,李爱华编写了第5章、第6章,李越编写了第7章、第8章,刘海莎编写了第9章。感谢湖南大学信息科学与工程学院李仁发教授对本书的支持和关心,同时感谢陈宝贤、李小英、银红霞、何英、朱理对本书提出的宝贵建议。

由于编者水平有限,加之时间仓促,书中难免存在错误或不足之处,敬请读者批评指正。有任何问题或建议,请与作者联系:cj7428@vip.163.com。

陈娟

于湖南长沙岳麓山

2015年1月

目 录

第1章 数据库基础 1

1.1 数据库系统 1

1.1.1 数据与数据处理 1

1.1.2 计算机数据管理 1

1.1.3 数据库系统的组成 5

1.2 关系数据库 7

1.2.1 概念模型 7

1.2.2 数据模型 9

1.2.3 关系模型 9

1.2.4 关系运算 11

1.3 Visual FoxPro 概述 13

1.3.1 Visual FoxPro 的发展历程 14

1.3.2 Visual FoxPro 的安装 14

1.3.3 Visual FoxPro 的启动和退出 16

1.3.4 Visual FoxPro 的用户界面 17

1.3.5 Visual FoxPro 的选项设置 18

1.4 项目管理器 19

1.4.1 创建项目 19

1.4.2 使用项目管理器 21

1.4.3 定制项目管理器 23

习题 1 24

第2章 数据与数据运算 29

2.1 常量与变量 29

2.1.1 常量 29

2.1.2 变量 31

2.1.3 数组 33

2.2 运算符与表达式 34

2.2.1 数值表达式 34

2.2.2 字符表达式 35

2.2.3 日期表达式 35

2.2.4 逻辑表达式 36

2.3 常用函数 39

2.3.1 数值处理函数 39

2.3.2 字符串处理函数 41

2.3.3 日期和时间函数 44

2.3.4 数据类型转换函数 45

2.3.5 测试函数 47

习题 2 48

第3章 数据库与数据表的操作 53

3.1 设计数据库 53

3.1.1 了解用户需求 53

3.1.2 确定数据库中所需的表 55

3.1.3 设计表的结构 56

3.1.4 确定表的主关键字 59

3.1.5 确定表之间的关系 59

3.2 建立数据库与数据表 59

3.2.1 建立数据库 60

3.2.2 建立数据表 61

3.2.3 定义数据表结构 63

3.2.4 输入数据记录 63

3.2.5 修改数据表结构 66

3.2.6 设置数据字典信息 67

3.2.7 通过浏览窗口新增、修改、

删除数据 72

3.3 数据表的基本操作 75

3.3.1 打开和关闭表 75

3.3.2 显示表的数据记录 77

3.3.3 移动记录指针 78

3.3.4 查找记录 79

3.3.5 新增记录 81

3.3.6 删除记录 82

3.3.7 修改记录 83

3.3.8 筛选数据表 85

3.3.9 表的复制和导入 86

3.3.10 记录与数组的数据交换 89

3.3.11 记录的统计	90	4.2.10 谓词和量词	131
3.4 数据库的基本操作	91	4.2.11 集合的并运算	132
3.4.1 打开数据库及设计器	91	4.2.12 查询结果的输出	132
3.4.2 关闭数据库	92	4.3 数据操纵	133
3.4.3 向数据库添加数据表	93	4.3.1 插入记录	133
3.4.4 从数据库移去数据表	93	4.3.2 更新记录	134
3.4.5 自由表	94	4.3.3 删除记录	134
3.4.6 删除数据库	95	4.4 数据定义	135
3.4.7 数据库的清理与检验	95	4.4.1 建立数据表	135
3.5 索引的建立及使用	96	4.4.2 修改数据表	137
3.5.1 索引的概念	96	4.4.3 删除数据表	139
3.5.2 索引的建立	97	习题 4	139
3.5.3 索引的使用	100		
3.5.4 索引的删除	101	第 5 章 查询与视图	149
3.5.5 物理排序	102	5.1 查询	149
3.6 多表的使用	102	5.1.1 查询的概念	149
3.6.1 工作区	102	5.1.2 查询的建立	149
3.6.2 使用其他工作区的表	104	5.1.3 查询与 SQL 语句的对应	153
3.6.3 数据表之间的临时关联	105	5.1.4 查询的保存、使用和修改	153
3.7 永久联系及参照完整性	107	5.1.5 定义查询去向	154
3.7.1 永久联系	107	5.2 视图	156
3.7.2 参照完整性	108	5.2.1 视图的概念	156
3.7.3 数据完整性	110	5.2.2 视图的建立	157
习题 3	110	5.2.3 视图的修改和使用	159
		5.2.4 视图与数据更新	159
第 4 章 结构化查询语言	117	习题 5	161
4.1 SQL 概述	117		
4.1.1 SQL 的发展	117	第 6 章 结构化程序设计	163
4.1.2 SQL 的特点	117	6.1 程序文件	163
4.2 数据查询	118	6.1.1 程序文件的基本概念	163
4.2.1 SELECT 命令的基本格式	118	6.1.2 程序文件的建立和修改	163
4.2.2 简单查询	119	6.1.3 程序的运行	165
4.2.3 特殊运算符	120	6.1.4 输入命令	166
4.2.4 统计查询	121	6.1.5 其他命令	167
4.2.5 分组查询	122	6.2 程序的基本结构	168
4.2.6 排序查询	125	6.2.1 顺序结构	168
4.2.7 简单连接查询	125	6.2.2 选择结构	168
4.2.8 超连接查询	127	6.2.3 循环结构	172
4.2.9 嵌套查询	129	6.3 多模块程序设计	179

6.3.1 过程的定义和调用	180
6.3.2 参数传递	182
6.3.3 变量的作用域	183
6.3.4 存储过程	184
习题 6	187

第 7 章 表单设计 193

7.1 面向对象基本概念	193
7.1.1 对象	193
7.1.2 Visual FoxPro 基类简介	195
7.1.3 对象的引用	196
7.2 表单的建立与运行	196
7.2.1 使用表单向导创建表单	196
7.2.2 修改表单	199
7.2.3 运行表单	199
7.3 表单设计器	200
7.3.1 启动表单设计器	200
7.3.2 设置数据环境	201
7.3.3 向表单中添加控件	203
7.3.4 为表单及控件设置属性	205
7.3.5 为表单及控件编写代码	208
7.3.6 在表单中快速添加数据 绑定控件	210
7.4 表单控件	212
7.4.1 标签控件	212
7.4.2 线条与形状控件	213
7.4.3 图像控件	214
7.4.4 计时器控件	216
7.4.5 文本框和编辑框控件	217
7.4.6 微调控件	220
7.4.7 选项按钮组控件	221
7.4.8 复选框控件	223
7.4.9 列表框和组合框控件	224
7.4.10 页框控件	228
7.4.11 容器控件	230
7.4.12 表格控件	231
7.4.13 命令按钮和命令按钮组 控件	235
7.4.14 ActiveX 控件和 ActiveX	

绑定控件	243
7.4.15 超级链接控件	248
7.5 多重表单与表单集	249
7.5.1 表单的类型	249
7.5.2 主从表单之间的参数传递	252
7.5.3 表单集	257
7.6 自定义类	258
7.6.1 类的创建	258
7.6.2 类的使用	261
7.6.3 类的编辑	264
习题 7	264

第 8 章 报表设计 271

8.1 报表概述	271
8.1.1 报表组成	271
8.1.2 报表布局	271
8.1.3 创建报表的方法	272
8.2 使用“报表向导”设计报表	272
8.2.1 使用“报表向导”设计 报表	272
8.2.2 修改报表	275
8.2.3 预览和打印报表	275
8.3 使用“快速报表”设计报表	276
8.4 使用“报表设计器”设计报表	278
8.4.1 新建报表	278
8.4.2 报表设计器	278
8.4.3 报表的数据环境	280
8.4.4 在报表中添加控件	281
8.4.5 分组报表	289
8.4.6 多栏报表	292
习题 8	293

第 9 章 菜单设计 297

9.1 菜单概述	297
9.1.1 菜单系统的结构	297
9.1.2 设计菜单系统的原则	298
9.2 下拉式菜单的设计	298
9.2.1 菜单设计的基本过程	298
9.2.2 定义菜单	299

9.3 快捷菜单的设计	306
习题 9	309
第 10 章 应用程序的生成与发布	311
10.1 应用程序的生成与发布	311
10.1.1 主文件	311
10.1.2 连编项目	313
10.1.3 应用程序发布	315

第 10 章 应用程序的生成与发布...311

10.1	应用程序的生成与发布	311
10.1.1	主文件	311
10.1.2	连编项目	313
10.1.3	应用程序发布	315

10.2	数据库应用系统的开发步骤	316
10.2.1	需求分析	316
10.2.2	数据库设计	317
10.2.3	应用程序设计	317
10.2.4	应用程序的生成与发布	318
习题 10		318

参考文献.....320

第 1 章

数据库基础

从 20 世纪 60 年代末,人们开始采用数据库技术来有效地管理数据。随着计算机技术和通信技术的发展,数据库系统已经应用于各行各业。为了使读者更好地理解数据库系统,本章介绍了数据库的一些基本概念,其中着重讲解数据库系统的组成和关系数据库的基础知识。

Visual FoxPro 既是关系数据库管理软件,又是可视化的面向对象的集成开发工具。本章介绍该软件的发展历程及安装、启动、退出等操作,以及 Visual FoxPro 的项目管理文件的创建、使用和定制。

1.1 数据库系统

1.1.1 数据与数据处理

数据是存储在某一媒体上,对客观事物进行描述的物理符号。数据不仅包括数字、字母、汉字等文本形式,也包括图像、声音、视频等多媒体形式。

数据的概念包括内容和形式两个方面。数据的内容指所描述的客观事物的具体特性,即数据的值;数据形式是指存储数据内容的具体形式。例如,描述某人的出生日期可以使用“2000 年 6 月 25 日”,也可以使用“2000/6/25”,采取这两种不同形式所表示的数据的值是相同的。

数据处理是对数据的采集、整理、存储、分类、计算、加工、检索和传输等一系列操作的总和。其目的是从大量原始的数据中,获得有价值的信息,作为人们行为和决策的依据。数据处理是将数据转化为信息的过程。数据是信息的载体,信息是经过加工的数据。例如,在班主任的成绩单中,记录着各位学生的学号、姓名、各科成绩,这些属于数据。由此计算出各位学生的总分,统计成绩排名情况,作为评定奖学金的依据,这就属于信息。

1.1.2 计算机数据管理

早期的计算机主要用于科学计算。随着计算机硬件技术、软件技术和计算机应用范围的不断发展,人们逐渐将计算机用于数据管理。计算机数据管理经历了 3 个阶段:人工管理阶段、文件系统阶段和数据库系统阶段。

1. 人工管理阶段

20 世纪 50 年代中期以前,计算机硬件方面,外存储器只有纸带、卡片、磁带,没有像硬盘一样可以随机访问、直接存取的外部存储设备;软件方面,没有操作系统软件和数据管理软件。

此阶段的数据处理有以下特点。

(1) 数据不保存。用户把应用程序和数据一起输入内存,通过应用程序对数据进行处理,输出处理结果。任务完成后,数据随着应用程序从内存一起释放。

(2) 数据和程序不具有独立性。数据由应用程序自行管理。应用程序中不仅要规定数据的逻辑结构,还要阐明数据在存储器上的存储地址。当数据改变时,应用程序也要随之改变。

(3) 数据不能共享。一个应用程序中的数据无法被其他应用程序所利用。程序和程序之间不能共享数据,因而产生大量重复的数据,称为数据冗余。

例 1.1 图 1.1 所示的是两个 C 语言的计算机程序。程序一的功能是计算学生各科成绩的平均分,程序二的功能是计算并显示每门功课的平均分。这两个程序所要处理的数据作为数组存放在程序中。

```

/*计算每位学生各科成绩的平均分*/
#include <stdio.h>
#define N 5
void main()
{
    struct student{
        char num[4];
        char name[9];
        int score[3];
    }stu[N]={{"101","Mary",80,85,90}, {"102","rose",80,90,95}, {"103","Harry",75,72,65},
            {"104","Peter",65,63,58}, {"105","Richard",95,93,88}}; /*数组记录了五位学生的三门课程的成绩*/
    int i, j, sum; double avg;
    for(i=0; i<N; i++)
    {
        printf("%s %s %s ", stu[i].num, stu[i].name); /*输出学生的学号和姓名*/
        sum=0;
        for(j=0; j<3; j++)
            sum+=stu[i].score[j]; /*计算总分*/
        avg=sum/3; /*计算平均分*/
        printf("%f\n", avg); /*输出平均分*/
    }

    /*计算各门课程的平均分*/
    #include <stdio.h>
    #define M 3
    void main()
    {
        struct course{
            char coursename[10];
            int score[5];
        } s[M]={{"chinese",80,80,75,65,95}, {"maths",85,90,72,63,93}, {"english",90,95,65,58,88}};
        int i, j, sum; double avg;
        for(i=0; i<M; i++)
        {
            printf("%s ", s[i].coursename); /*输出课程名称*/
            sum=0;
            for(j=0; j<5; j++)
                sum+=s[i].score[j]; /*计算总分*/
            avg=sum/5; /*计算平均分*/
            printf("%f\n", avg); /*输出平均分*/
        }
    }

```

图 1.1 使用数组来处理数据的 C 语言程序

2. 文件系统阶段

20 世纪 50 年代后期至 60 年代中后期,随着计算机在数据管理中的广泛应用,大量的数据存储、检索和维护成为紧迫的要求。硬件方面,可直接存取的磁盘成为主要外存,软件方面,出现了高级语言和操作系统。

文件系统(见图 1.2)阶段的数据处理有以下特点。

(1) 数据长期保存。数据项集合为记录,长期保存在磁盘的数据文件中,供用户反复调用和更新。

(2) 程序与数据有了一定的独立性。应用程序和数据分别存储在程序文件和数据文件中,应用程序按文件名访问数据文件,不必关心数据在存储器上的位置、输入/输出方式。

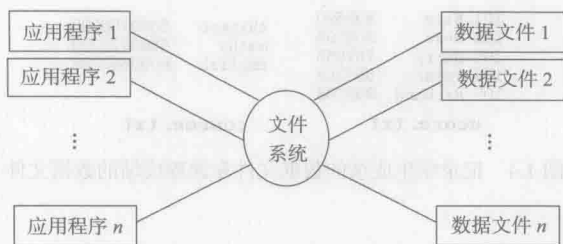


图 1.2 文件系统

(3) 数据的独立性低。由于应用程序对数据的访问基于特定的结构和存取方法，当数据的逻辑结构发生改变时，必须修改相应的应用程序。

(4) 数据的共享性差，存在数据冗余和数据的不一致。大多数情况下，一个应用程序对应一个数据文件。当不同的应用程序所处理的数据包含相同的数据项时，通常是建立各自的数据文件，从而产生大量的数据冗余。当一个数据文件的数据项被更新，而其他数据文件中相同的数据项没有被更新时，将造成数据的不一致。

例 1.2 图 1.3 所示的是两个 C 语言的计算机程序。程序的功能与例 1.1 相同，但是程序所要处理的数据存放在数据文件 score.txt 和 course.txt 中，如图 1.4 所示。

```

/* 计算数据文件score.txt中的学生的平均分*/
#include <stdlib.h>
#define N 5
#define M 3
void main()
{
    struct student{
        char num[4];
        char name[9];
        int score[M];
    };
    struct student s;
    int i, j, sum; double avg;
    FILE *fp;
    fp=fopen("score.txt", "r"); /*只读方式打开数据文件*/
    for(i=0; i<N; i++)
    {
        fread(&s, sizeof(struct student), 1, fp); /*从文件指针读出数据到结构体中*/
        sum=0;
        for(j=0; j<M; j++)
            sum+=s.score[j]; /*计算总分*/
        avg=sum/M; /*计算平均分*/
        printf("%s %s %f\n", s.num, s.name, avg); /*输出学号、姓名和平均分*/
    }
    fclose(fp);
}

/* 计算数据文件course.txt中课程的平均分*/
#include <stdio.h>
#include <stdlib.h>
#define N 3
#define M 5
void main()
{
    struct course{
        char coursename[10];
        int score[M];
    } s;
    int i, j, sum; double avg;
    FILE *fp;
    fp=fopen("course.txt", "r"); /*以只读方式打开课程成绩文件*/
    for(i=0; i<N; i++)
    {
        fread(&s, sizeof(struct course), 1, fp); /*从文件指针读出数据到结构体中*/
        sum=0;
        for(j=0; j<M; j++)
            sum+=s.score[j]; /*计算总分*/
        avg=sum/M; /*计算平均分*/
        printf("%s %f\n", s.coursename, avg); /*输出课程名称和平均分*/
    }
    fclose(fp);
}

```

图 1.3 使用数组来处理数据的 C 语言程序

101 Mary	808590	chinese	8080756595
102 Rose	809095	maths	8590726393
103 Harry	757265	english	9095655888
104 Peter	856358		
105 Richard	959388		

score.txt course.txt

图 1.4 记录学生成绩的数据文件和课程成绩的数据文件

3. 数据库系统阶段

20 世纪 60 年代后期,大容量和快速存储的磁盘相继投入市场,为新型数据管理技术奠定了物质基础。此外,计算机管理的数据量急剧增长,多用户、多程序实现数据共享的要求日益增强。在这种情况下,文件系统的管理已经不能满足需求,数据库技术应运而生。

数据库系统(见图 1.5)阶段的数据处理有以下特点。

(1) 数据的共享性高,冗余度低。建立数据库时,以面向全局的观点组织数据库中的数据。数据可被多个用户、多个应用程序共享使用,大大减少数据冗余。

(2) 采用特定的数据模型。数据库中的数据是以一定的逻辑结构存放的,这种结构由数据库管理系统所支持的数据模型来决定。目前流行的数据库管理系统大多建立在关系模型的基础上。

(3) 数据独立性高。数据与应用程序之间彼此独立。当数据的存储格式、组织方法和逻辑结构发生改变时,不需要修改应用程序。

(4) 统一的数据控制功能。数据库由数据库管理系统来统一管理,并提供对数据的并发性、完整性、安全性等控制功能。

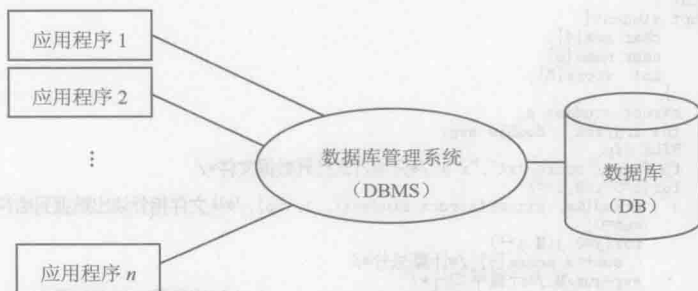


图 1.5 数据库系统

例 1.3 图 1.6 所示的是存放成绩的数据库。在数据库管理系统中使用如图 1.7 所示的两个 SQL 语句,即可完成例 1.1 程序中所要求的计算学生的平均分和课程的平均分的功能。

学号	姓名	课程名称	成绩
101	Mary	chinese	80
101	Mary	maths	85
101	Mary	english	90
102	Rose	chinese	80
102	Rose	maths	90
102	Rose	english	95
103	Harry	chinese	75
103	Harry	maths	72
103	Harry	english	65
104	Peter	chinese	65
104	Peter	maths	83
104	Peter	english	58
105	Richard	chinese	95
105	Richard	maths	93
105	Richard	english	88

图 1.6 存放成绩的数据库

note 计算每位学生的平均分
 select 学号,姓名,avg(成绩) from score group by 学号
 note 计算每门课程的平均分
 select 课程名称,avg(成绩) from score group by 课程名称

图 1.7 计算学生的平均分和课程的平均分的查询语句

1.1.3 数据库系统的组成

数据库系统 (DataBase System, DBS) 是指引入数据库技术的计算机系统。它实现了有组织地、动态地存储大量相关数据, 提供了数据处理和信息资源共享的便利手段。数据库系统通常由 5 部分组成: 硬件系统、数据库、数据库管理系统、相关软件 and 各类人员, 其层次示意图如图 1.8 所示。

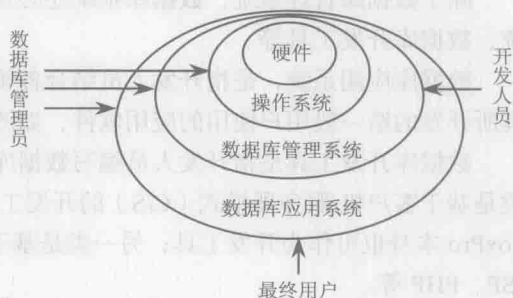


图 1.8 数据库系统层次示意图

1. 硬件系统

硬件系统主要指计算机硬件设备, 包括 CPU、内存、外存、输入/输出设备等。由于要运行操作系统、数据库管理系统的核心程序和应用程序, 要求计算机有足够大的内存; 同时, 由于数据库、系统软件和应用软件都保存在外存中, 对计算机的外存容量的要求也很高。此外, 对于网络数据库系统, 还需要有网络通信设备的支持。

2. 数据库集合

数据库 (DataBase, DB) 可直观地理解为数据的仓库。数据库是指存储在计算机外存中, 结构化的相关数据的集合。它不仅包含了描述事物本身的数据, 还包含了相关数据之间的联系。

数据库以文件的形式存储在外存中, 用户通过数据库管理系统来统一管理和控制数据。

3. 数据库管理系统

数据库管理系统 (DataBase Management System, DBMS) 是对数据实行专门管理的系统软件, 是数据库系统的核心。它在操作系统的基础上运行, 方便用户建立、使用和维护数据库, 提供数据的安全性和完整性等统一控制机制。

目前, 广泛使用的大型数据库管理系统有 Oracle、Sybase、DB2 等, 小型数据库管理系统有 SQL Server、Visual FoxPro、Access 等。

数据库管理系统主要功能如下。

(1) 数据定义: DBMS 提供数据定义语言 (Data Definition Language, DDL), 负责数据库对象的建立、修改和删除等。

(2) 数据操纵: DBMS 提供数据操纵语言 (Data Manipulation Language, DML), 实现数据的基本操作。例如, 对表中数据的查询、插入、删除和修改。

(3) 数据控制: 包括安全性控制、完整性控制和并发性控制等。

安全性控制主要是通过授权机制实现, DBMS 提供数据控制语言 (Data Control Language, DCL), 设置或者更改数据库用户的权限。在访问数据库时, 由 DBMS 对用户的身份进行确认, 只有具有指定权限的用户才能进行相应的操作。

完整性控制是保证数据库中数据的正确性和有效性。例如, 百分制的成绩的值应该是 0~100 之间的数值, 一旦在数据库中定义了这个约束性条件, 在插入和修改成绩时, DBMS 都会进行检查, 保证不符合条件的数据不会存入数据库。

并发控制是指当多个用户同时对同一项数据进行操作时, DBMS 采取一定的控制措施, 防止数据的不一致。例如, 两个终端的应用程序在同时购买车票, 为避免将同号的车票卖给不同的用户, DBMS 可以采取对数据加锁的方法, 以保证当一个用户在存取该数据时其他用户不能修改此

数据。

（4）数据库维护：包括数据库的备份和恢复，数据库的转换、数据库的性能监视和优化等。

4. 相关软件

除了数据库管理系统，数据库系统还必须要有相关软件的支持，包括操作系统、数据库应用系统、数据库开发工具等。

数据库应用系统，是指开发人员结合各领域的具体需求，利用数据库系统资源，使用开发工具所开发的给一般用户使用的应用软件，如图书管理系统、学籍管理系统、商品进销存系统等。

数据库开发工具是指开发人员编写数据库应用系统所使用的软件平台。通常可分为两类：一类是基于客户机/服务器模式（C/S）的开发工具，如 Visual Basic、Visual C++、Delphi 等，Visual FoxPro 本身也可作为开发工具；另一类是基于浏览器/服务器模式（B/S）的开发工具，如 ASP、JSP、PHP 等。

C/S 模式如图 1.9 所示，在服务器结点存放数据及执行 DBMS 功能，客户机安装应用系统。客户端的用户请求被传送到服务器，服务器进行处理后，将处理结果返回给用户。

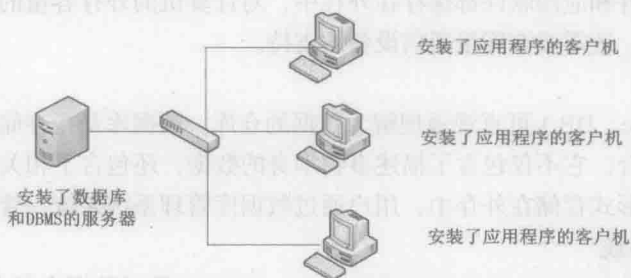


图 1.9 C/S 模式

对于不需要共享使用的数据库系统，通常将数据库、DBMS、数据库应用系统装在一台计算机上，由个人用户独占使用数据。

随着因特网的广泛使用，B/S 模式得到了广泛的应用，如图 1.10 所示。客户端仅安装浏览器软件，用户通过 URL 向 Web 服务器发出请求，Web 服务器运行脚本程序，向数据库服务器发出数据请求。数据库服务器执行处理后，将结果返回给 Web 服务器。Web 服务器根据结果产生网页文件，客户端接收到网页文件后，在浏览器中显示出来。

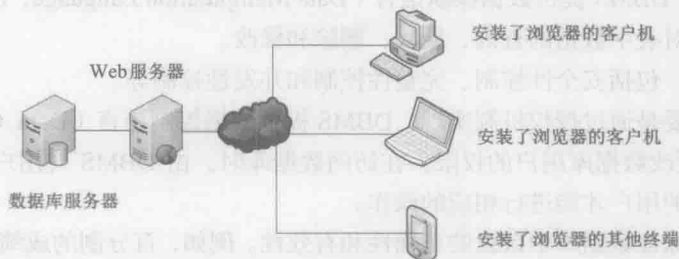


图 1.10 B/S 模式

5. 各类人员

数据库系统中还包括设计、建立、管理、使用数据库的各类人员。

（1）数据库管理员（Database Administrator, DBA）。数据库管理员是负责全面管理和实施数

数据库控制与维护的技术人员,他要参与数据库的规划、设计和建立,负责数据库管理系统的安装和升级;规划和实施对数据库的备份和还原;规划和实施数据库的安全性,控制和监视用户对数据库的存取访问;监督和记录数据库的操作状况,进行性能分析,实施系统优化。

(2) 开发人员。开发人员负责应用系统的需求分析,设计应用系统的功能,使用开发工具实现应用系统。

(3) 最终用户。最终用户只需通过执行数据库应用系统来处理数据,不需要了解数据库的设计、维护和管理等问题。

1.2 关系数据库

数据库中存储和管理的数据都源于现实世界的客观事物。例如,在图书管理系统中的图书和读者,在教学管理系统中的学生、教师、课程;销售管理系统中的商品、客户、员工……由于计算机不能处理这些具体事物,人们必须要将其转换为计算机能够管理的数据。通常,这种转换过程分为两个阶段:首先要将现实世界转换为信息世界,即建立概念模型;再将信息世界转换为数据世界,即建立数据模型。

1.2.1 概念模型

现实世界中事物及联系在人们头脑中的反映,经过人们头脑的分析、归纳、抽象,形成信息世界。对信息世界所建立的抽象的模型,称之为概念模型。由于概念模型是用户与数据库设计人员之间进行交流的语言,因此概念模型一方面应该能够方便、直接地表达应用中的各种语义知识,另一方面它还应该简单、清晰、易于用户理解。目前常用实体联系模型表示概念模型。

1. 实体

实体是客观存在并且可相互区别的事物。它可以是实际的事物,如读者、图书、学生、教师、课程等;也可以是抽象的事件,如借书、选课、订货等活动。

2. 实体属性

实体的特性称为属性,一个实体可以用多个属性来描述。

例如,图书实体可以用条形码、书名、作者、出版社、出版年月、售价等属性来描述。

读者实体可以用读者证号、姓名、身份、性别、电话号码等属性来描述。

3. 实体型和实体集

用实体名及其属性集合描述的同类实体,称为实体型。

例如,图书(条形码、书名、作者、出版社、出版年月、售价)就是一个实体型。读者(读者证号、姓名、身份、性别、电话号码)也是一个实体型。

同类型实体的集合称为实体集。例如,所有的图书构成一个实体集。在图书实体集中,“P0000001 马克思的人学思想袁贵仁北师大 1996/06/06 19.0”表示一本具体的书。所有的读者也构成一个实体集。在读者实体集中,“005 孙建平男研究生 13507317845”表示一个具体的读者。

4. 实体间的联系

实体间的联系就是指实体集与实体集之间的联系。实体间的联系分为一对一、一对多和多对多3种。

(1) 一对一联系

设有实体集 A 和实体集 B , 若实体集 A 中的每个实体仅与实体集 B 中的一个实体联系, 反之亦然, 则两个实体间为一对一联系, 记为 $1:1$ 。例如, 班级和班长是两个实体集, 一个班级只能有一个班长, 而一个班长只能在一个班级任职, 则班级和班长之间为一对一的联系。

(2) 一对多联系

设有实体集 A 和实体集 B , 若对于实体集 A 中的每个实体, 实体集 B 都有多个实体与之对应, 反之, 对于实体集 B 中的每个实体, 实体集 A 中只有一个实体与之对应, 则两个实体间为一对多联系, 记为 $1:n$ 。例如, 班级和学生是两个实体集, 一个班级有多名学生, 而一个学生只能属于一个班级, 则班级和学生之间为一对多的联系。

(3) 多对多联系

设有实体集 A 和实体集 B , 若对于实体集 A 中的每个实体, 实体集 B 都有多个实体与之对应; 反之, 对于实体集 B 中的每个实体, 实体集 A 中也有多个实体与之对应, 则两个实体间为多对多联系, 记为 $m:n$ 。例如, 图书和读者两个实体集, 一个读者可以借阅多本图书, 而一本图书也可以被多位读者相继借阅, 则图书和读者之间为多对多的联系。学生和课程两个实体集, 一个学生可以选修多门课程, 而一门课程也可以被多位学生选修, 则学生和课程之间为多对多的联系。

5. E-R 图

实体-联系模型使用 E-R 图 (Entity-Relationship Diagram) 来描述概念模型。在 E-R 图中, 用矩形表示实体型, 用椭圆表示实体的属性, 用菱形表示实体型之间的联系, 相应的实体名、属性名、联系名写明在对应的框内, 用无向边将各种框连接起来, 并在连接实体型的线段上标上联系的类型。图书与读者的 E-R 图如图 1.11 所示。

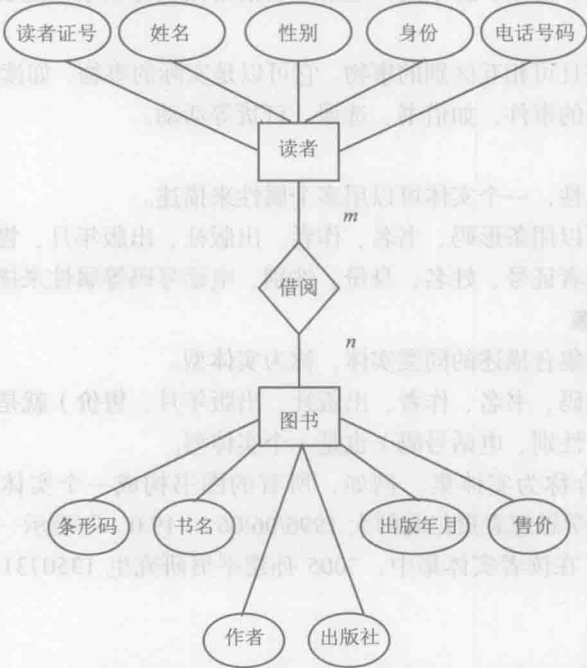


图 1.11 读者与图书的 E-R 图