前　言

随着我国社会经济的迅猛发展，计算机在各行各业的应用得到了迅速推广。社会上需要大量既熟悉自己所从事的专业，又掌握相关计算机应用技术的新型人才。社会发展对人才提出新的要求，作为新型人才培养主要力量的高等院校，都纷纷加强了对非计算机专业学生的计算机知识的教育。不仅要求各专业学生能够掌握计算机的基本知识和应用，还要求学生能够掌握基本的程序设计方法，提高学生在实际工作中应用计算机解决问题的能力，增强学生在社会中的竞争力。

非计算机专业的计算机教育，在教学目的、教学内容、教学方法等各方面都与计算机专业有很大的不同。因此，不能照搬计算机专业的教学模式。目前，衡量非计算机专业学生掌握计算机知识和能力的方法，主要是通过各级的“计算机等级考试”。根据“计算机等级考试”的要求和将来工作的需要，对于非计算机专业特别是农、医、商、经、文和管理专业的学生，要学习和掌握程序设计的基本方法，选择 Visual FoxPro 是比较适合的。

Visual FoxPro 9.0 系统是 Microsoft 公司最新发布的一款数据库应用系统，它在以往版本的基础上有了很大的改进，功能更加强大，提供了可视化界面的设计方法，支持面向对象的程序设计技术，并且新增了许多 Internet 的功能。同时作为具备自开发语言的数据库管理系统，Visual FoxPro 9.0 系统是进行中小型数据库应用系统开发的优秀工具。

本书根据 Visual FoxPro 9.0 系统既具有数据库管理系统的功能，又具有强大的可视化程序开发功能这一特点，首先介绍了如何使用可视化的方式和命令方式操作数据库、表和记录，然后介绍了 Visual FoxPro 9.0 程序设计的基本知识以及面向对象程序设计的方法，接下来介绍了表单、表单控件、菜单、工具栏和报表等概念，然后介绍了如何测试、调试和编译程序，最后通过一个小型管理系统开发实例“教学管理系统”介绍了程序设计的完整过程和使用 Visual FoxPro 9.0 开发数据库程序时的一些常用技巧。

本教材目标明确、重点突出、内容翔实、范例精确、条理清晰。在内容选材上主要针对非计算机专业的学生，各章节在组织和内容表述上条理性好，清晰度高，语言表达通俗易懂。并注重采用案例教学的方法，从设计实例引出理论知识，由浅入深、循序渐进，注重实践性和应用性。本书除了教授基本知识和技能外，突出体现知识点间的关联性，培养学生融会贯通的自主学习能力。在每一章后面，都配有一定数量的习题。为了便于进一步理解和掌握教材内容的需要，还配套出版了《Visual FoxPro 上机实验与二级考试指导》一书。

在使用本教材时最好采用多媒体教学方法，建议安排 60 教学课时和 30 上机课时。本教材由余益负责全书内容的安排和统稿，并编写第 10～14 章；甘九如编写第 1～5 章；姚怡编写第 6～9 章。

在本教材的编写过程中，得到广西大学计算机与信息学院李陶深教授的大力支持和帮助。广西师范大学出版社的编辑也为本书的出版付出了辛勤劳动。编者谨对他们表示诚挚的感谢！

由于时间和水平所限，书中存在错误与纰漏之处在所难免，恳请专家和读者斧正。

编者

2007 年 5 月

目　录

第 1 章 数据库基础知识……（1）
1.1 数据和数据处理……（1）
1.1.1 数据与数据处理 ……（1）
1.1.2 数据管理技术的发展 ……（2）
1.2 数据库系统……（4）
1.2.1 数据库、数据库管理系统、数据库系统 ……（4）
1.2.2 数据库系统的特点 ……（6）
1.3 数据库的类型……（6）
1.3.1 数据模型……（6）
1.3.2 数据库的类型……（9）
1.4 关系数据库……（9）
1.4.1 基本概念……（9）
1.4.2 关系操作……（10）
1.5 数据库设计步骤……（11）
习题 ……（12）
第 2 章 Visual FoxPro 9.0 概述……（14）
2.1 数据库管理系统发展的历史沿革 ……（14）
2.2 Visual FoxPro 9.0 的功能简介 ……（15）
2.3 Visual FoxPro 9.0 的安装与启动 ……（16）
2.3.1 Visual FoxPro 9.0 的运行环境 ……（16）
2.3.2 Visual FoxPro 9.0 的安装和卸载 ……（17）
2.3.3 Visual FoxPro 9.0 的启动与退出 ……（20）
2.4 Visual FoxPro 9.0 系统环境的配置……（21）
2.5 Visual FoxPro 9.0 的主界面……（23）
2.6 Visual FoxPro 9.0 的工作方式……（26）
2.7 项目管理器 ……（27）
2.7.1 创建项目 ……（27）
2.7.2 使用项目管理器……（30）
2.7.3 定制项目管理器……（33）
2.8 Visual FoxPro 9.0 的命令概述……（34）
2.8.1 命令的结构……（34）
2.8.2 命令的书写规则……（35）
2.8.3 命令中的常用子句……（35）
2.9 Visual FoxPro 9.0 的文件组成……（36）
习题……（37）

第 3 章 Visual FoxPro 9.0 基础……（39）
3.1 数据类型 ……（39）
3.2 常量 ……（40）
3.3 变量 ……（42）
3.3.1 字段变量……（42）
3.3.2 内存变量……（43）
3.3.3 系统变量……（45）
3.3.4 数组变量……（46）
3.4 函数 ……（47）
3.4.1 数值函数……（47）
3.4.2 字符函数……（48）
3.4.3 日期和时间函数……（48）
3.4.4 数据类型转换函数……（49）
3.4.5 测试函数……（49）
3.4.6 其他函数……（51）
3.5 表达式 ……（53）
3.5.1 数值表达式……（53）
3.5.2 字符表达式……（53）
3.5.3 日期和时间表达式……（53）
3.5.4 关系表达式……（54）
3.5.5 逻辑表达式……（55）
3.5.6 表达式运算的优先级……（56）
习题……（56）
第 4 章 表的创建与使用 ……（58）
4.1 创建表……（58）
4.1.1 定义表的结构……（58）
4.1.2 创建表 ……（59）
4.2 打开和关闭表 ……（65）
4.3 修改表结构 ……（66）
4.4 向表中加入记录 ……（67）
4.4.1 浏览方式输入数据……（67）
4.4.2 编辑方式输入数据……（68）
4.4.3 追加方式输入数据……（68）
4.4.4 备注型数据的输入……（70）
4.4.5 通用型数据的输入……（70）
4.5 记录的显示和修改方式 ……（71）
4.5.1 用浏览方式显示和修改数据 ……（71）
4.5.2 用编辑方式显示和修改数据……（74）
4.6 删除和恢复表中的记录 ……（75）
4.6.1 逻辑删除表中的记录 ……（75）

4.6.2 恢复表中逻辑删除的记录 ……（76）
4.6.3 物理删除表中的记录 ……（77）
习题……（77）
第5章 表的基本操作 ……（79）
5.1 记录的定位……（79）
5.2 排序与索引……（81）
5.2.1 排序 ……（81）
5.2.2 索引的概念 ……（82）
5.2.3 索引的类型 ……（83）
5.2.4 索引的建立 ……（83）
5.2.5 索引的使用 ……（88）
5.2.6 删除索引 ……（90）
5.3 查询与统计……（90）
5.3.1 数据表查询……（90）
5.3.2 记录数统计……（92）
5.3.3 求和……（92）
5.3.4 求平均值……（93）
5.3.5 分类汇总……（94）
习题……（95）
第6章 数据库的创建与操作……（96）
6.1 创建数据库……（96）
6.1.1 数据库设计的一般步骤 ……（96）
6.1.2 创建数据库 ……（97）
6.1.3 打开与关闭数据库 ……（99）
6.2 修改数据库……（101）
6.2.1 向数据库添加表……（101）
6.2.2 从数据库中移去表……（102）
6.3 创建数据库表……（103）
6.3.1 设置表中字段的显示标题 ……（104）
6.3.2 设置字段的显示格式和输入掩码……（105）
6.3.3 设置表中字段的注释……（105）
6.3.4 设置表中字段的默认值……（107）
6.3.5 设置表中字段的有效性规则……（107）
6.3.6 数据库表属性的设置 ……（109）
6.4 建立表间的关联关系 ……（110）
6.4.1 表之间的关系 ……（110）
6.4.2 创建表间的永久关系 ……（111）
6.4.3 创建表间的临时关系 ……（112）
6.4.4 设置参照完整性 ……（114）
6.5 删除数据库 ……（116）

6.6 多表操作……（116）
6.3.1 多工作区的概念……（117）
6.3.2 使用不同工作区的表……（117）
6.3.3 数据工作期……（118）
习题……（119）
第 7 章 查询与视图……（121）
7.1 查询……（121）
7.1.1 利用“查询向导”创建查询……（121）
7.1.2 利用“查询设计器”创建查询……（124）
7.1.3 运行查询……（127）
7.1.4 查看查询的 SQL 语句……（128）
7.1.5 利用查询进行统计和计算……（128）
7.2 视图……（130）
7.2.1 视图的概念……（130）
7.2.2 利用“视图向导”创建本地视图……（130）
7.2.3 利用“视图设计器”创建本地视图……（132）
7.2.4 视图的运行、修改和删除……（135）
7.2.5 利用视图更新数据表数据……（136）
7.2.6 创建参数视图……（138）
7.2.7 用命令方式处理视图……（139）
7.2.8 远程视图与连接……（140）
习题……（143）
第 8 章 关系数据库标准语言 SQL ……（145）
8.1 SQL 概述……（145）
8.1.1 SQL 语言的规则……（145）
8.1.2 SQL 语言的特点……（145）
8.2 SQL 的数据定义……（146）
8.2.1 基本表的定义……（146）
8.2.2 修改表的结构……（147）
8.2.3 删除表……（149）
8.3 SQL 的数据查询……（149）
8.3.1 简单查询……（150）
8.3.2 联接查询……（152）
8.3.3 嵌套查询……（152）
8.3.4 计算与分组查询……（153）
8.4 SQL 的数据操作……（154）
8.4.1 插入……（155）
8.4.2 更新……（156）
8.4.3 删除……（156）
习题……（157）

第 9 章 程序设计基础……（159）
9.1 程序文件……（159）
9.1.1 程序的概念……（159）
9.1.2 程序文件的建立与运行……（160）
9.2 程序的控制结构……（164）
9.2.1 顺序结构……（165）
9.2.2 选择结构……（165）
9.2.3 循环结构……（169）
9.3 多模块程序……（173）
9.3.1 子程序……（174）
9.3.2 自定义函数……（175）
9.3.3 过程……（176）
9.3.4 变量的作用域……（179）
9.3.5 程序的调试……（180）
习题……（184）
第 10 章 面向对象的程序设计……（186）
10.1 面向对象的概念……（186）
10.1.1 对象与类……（187）
10.1.2 面向对象程序设计的特点……（188）
10.1.3 事件与方法……（188）
10.1.4 Visual FoxPro 的基类……（189）
10.2 创建表单……（192）
10.2.1 用表单向导创建表单……（192）
10.2.2 利用表单设计器创建表单……（197）
10.3 设置表单的数据环境……（198）
10.4 表单的基本属性……（200）
10.5 表单中的控件……（201）
10.5.1 向表单添加控件……（201）
10.5.2 控件的操作和布局……（202）
10.5.3 设置控件的属性……（203）
10.6 管理表单……（204）
10.6.1 表单的保存……（205）
10.6.2 表单的修改……（205）
10.6.3 表单的运行……（205）
10.7 常用表单控件……（205）
10.7.1 标签……（205）
10.7.2 文本框……（207）
10.7.3 命令按钮……（208）
10.7.4 编辑框……（209）
10.7.5 命令按钮组……（210）

10.7.6 列表框……（212）
10.7.7 选项按钮组……（215）
10.7.8 复选框……（217）
10.7.9 组合框……（219）
10.7.10 表格……（220）
10.7.11 页框……（224）
10.7.12 微调控件……（229）
10.7.13 计时器……（230）
10.7.14 形状……（232）
10.7.15 图像……（233）
10.7.16 容器……（234）
10.8 子类的创建与应用……（236）
10.8.1 子类的创建方法……（236）
10.8.2 子类的应用实例……（238）
10.9 用户自定义属性和方法……（239）
10.9.1 用户自定义属性……（239）
10.9.2 用户自定义方法……（240）
10.9.3 自定义属性和方法的编辑……（241）
10.9.4 自定义属性和方法的应用实例……（242）
10.10 表单集……（243）
10.10.1 表单集……（243）
10.10.2 表单集应用实例……（244）
10.11 多重表单……（245）
10.11.1 表单的类型定义……（245）
10.11.2 隐藏 Visual FoxPro 的系统窗口……（246）
10.11.3 在顶层表单中运行子表单……（247）
10.11.4 表单间的参数传递……（248）
习题 ……（253）
第 11 章 菜单与工具栏的设计……（255）
11.1 菜单系统及其规划……（255）
11.1.1 菜单系统……（255）
11.1.2 菜单系统的规划……（255）
11.2 创建菜单……（256）
11.2.1 创建菜单的基本步骤……（256）
11.2.2 “菜单设计器”窗口……（257）
11.3 为顶层表单的菜单加载……（264）
11.4 创建快捷菜单……（268）
11.5 创建工具栏……（271）
11.6 添加工具栏到表单集中……（274）
习题……（274）

第 12 章 创建输出报表与标签……（276）
12.1 创建报表……（276）
12.1.1 利用报表向导创建报表 ……（276）
12.1.2 利用快速报表创建报表 ……（281）
12.1.3 利用报表设计器创建报表 ……（283）
12.2 报表控件 ……（285）
12.3 数据分组报表 ……（288）
12.4 多栏报表……（293）
12.5 修改和打印报表……（294）
12.5.1 修改报表……（294）
12.5.2 打印报表……（294）
12.6 标签……（295）
12.6.1 创建标签……（296）
12.6.2 编辑标签……（297）
12.6.3 打印标签……（297）
习题……（298）
第 13 章 创建用户的应用系统……（299）
13.1 设置主文件……（299）
13.2 初始化环境……（300）
13.3 控制事件循环……（300）
13.4 建立一个简单的主程序……（301）
13.5 连编项目……（302）
习题……（303）
第 14 章 小型管理系统开发实例 ……（304）
14.1 系统开发的一般过程……（304）
14.2 系统开发实例：教学管理系统……（305）
14.2.1 系统总体设计……（305）
14.2.2 数据库设计……（306）
14.2.3 主控程序设计……（308）
14.2.4 启动界面设计……（309）
14.2.5 系统登录表单设计……（312）
14.2.6 系统菜单设计……（314）
14.2.7 学生管理模块设计……（315）
14.2.8 教师管理模块设计……（321）
14.2.9 基本资料管理模块设计……（322）
14.2.10 系统管理模块设计……（322）
14.3 连编生成应用系统……（325）
习题答案……（327）
参考文献……（328）

第 1 章　数据库基础知识

随着计算机技术的不断发展，计算机的应用已渗透到人类社会的各个领域，从科学计算到信息处理，过程控制到人工智能。据统计，目前全世界 80％以上的计算机主要用于信息处理。

何谓信息处理，其真正含义应该是：为了产生信息而处理数据，通过处理数据而获得信息，通过分析和筛选信息可以产生决策。

可以说任何单位和部门都有数据需要处理。民航、火车站、汽车站有售票记录，图书馆有藏书目录和借书登记，银行有存取款记录，仓库有出入库登记，工厂有订货合同与生产计划，学校有教学计划与成绩管理等，这些都与数据处理息息相关。如不能及时收集和更新这些数据，必然会使某些单位的日常工作很快陷于混乱，譬如图书馆找不到书，售票处能否继续售票等。

现代信息技术的发展，使得各单位各部门涉及的信息量日益剧增，需要掌握和处理的数据量越来越大。目前企事业单位的管理、金融部门的通存通兑业务、电子商务、生产过程控制等，没有哪项不用计算机来处理数据。

数据库管理系统是利用计算机处理数据的有效工具。利用这一工具可以有效地组织和处理数据，使数据更好地为用户所共享，为人们提供有用的信息。

1.1 数据和数据处理

数据库是数据处理的手段，数据库管理系统是数据处理的有效工具。在学习和掌握这一有效工具之前，有必要先了解数据、数据处理的概念和计算机数据管理技术的发展历程。

1.1.1 数据与数据处理

数据，广义来说是指存储在某一种媒体上能够识别的物理符号。数据的概念包括两个方面，其一是描述事物特征的数据内容，如人的姓名、性别、出生日期等；其二是存储在某一种媒体上的数据形式，如某人的出生日期是“1986 年 7 月 1 日”，也可以表示为"07/01/86"，其含义并没有改变，只是表示的格式不同而已。

由于计算机数据处理领域的拓宽，数据的概念已经大大地拓宽了。它已不再是传统概念上的数字、字母、文字和其他特殊字符组成的文本型数据，还包括图形、图像、动画、视频、声音等多媒体数据。但使用最多、最广泛的还是数字和文字数据。

数据处理是指将数据转换成信息的过程。包括对各种类型的数据进行收集、存储、加工、

检索和传输的过程。从数据处理的角度而言，信息是被加工成一种特殊形式的数据，这种数据对数据的接收者来说往往就成为决策的依据之一。例如，在人事管理中，一个职工的“出生日期”属于原始数据，通过现年与出生日期相减得到的年龄是被加工后的数据，根据有关此人的年龄、性别、职称和职务的信息，就可以知道此人何时应当办理退休手续。又如，在学籍管理中，一个学生各门必修课、限定选修课和任意选修课的学分是原始数据，各种学分之和就是被加工后的数据，根据这些信息以及相关规定，就可判断该生是否修完规定学分，具备毕业资格。

利用计算机处理数据，就是采用外存储器存储数据，通过数据库管理系统管理数据，通过应用程序对数据进行加工处理。

1.1.2 数据管理技术的发展

数据处理的中心问题是数据管理。计算机对数据的管理是指对数据的组织、编码、存储、分类、检索和维护提供操作手段。

计算机数据管理随着计算机硬件、软件技术和计算机应用范围的发展而不断发展。按照发展过程划分方法，多年来经历了人工管理、文件系统、数据库系统、分布式数据库系统和面向对象数据库系统等几个阶段。如果高度概括起来，也可以简单地分为人工管理、文件系统和数据库系统三个阶段。

1. 人工管理阶段

20 世纪 50 年代中期，计算机硬件方面没有随机访问、直接存取的磁盘，软件方面没有专门管理数据的软件，数据由处理它的程序自行携带，一个程序附带一组数据。数据管理完全由程序设计人员自负其责。

这一时期计算机管理数据的特点是，数据密切配合程序，数据与程序不具独立性；一组数据对应一个程序，一个程序中的数据无法被其他程序利用，程序与程序之间存在大量的重复数据，这种数据称为数据冗余。

2. 文件系统阶段

20 世纪 50 年代后期至 60 年代中后期，计算机硬件方面，可直接存取数据的硬盘成为主要外存储器。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数据的管理软件。

在文件系统阶段，程序和数据分开存储，数据按一定的规则组成文件，可以长期保存在外存储器上被多次存取。在文件系统的支持下，程序通过文件名访问数据文件。如图 1.1 所示。

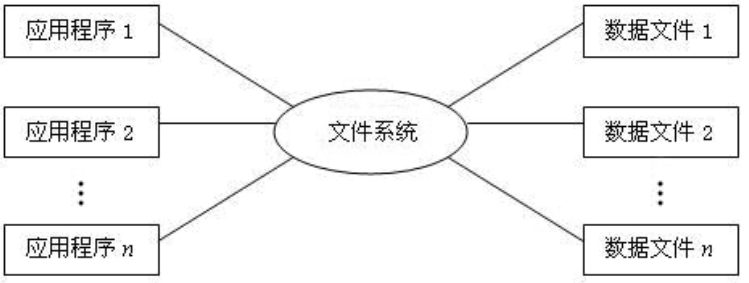


图 1.1 文件管理方式下应用程序与数据文件的关系

但这一阶段仍是以程序为中心，根据程序的需要组织数据文件，如工资文件、学生成绩文件都是根据专门的需要而设计的，形成数据和程序相互依赖。由于每个数据文件与一个或几个应用程序相对应，同一个数据项可能重复出现在多个文件中，导致数据冗余度大。不仅浪费存储空间，更严重的是，由于不能统一修改数据，容易造成数据的不一致性。

文件系统管理数据存在的问题阻碍了数据处理技术的发展，不能满足日益增长的信息需求。计算机软件工作者经过长期不懈的努力，于 20 世纪 60 年代末提出了数据库的概念。

3. 数据库系统阶段

这一阶段不是以程序为中心，而是以数据为中心的。先考虑各种应用的需要综合设计数据库，再设计各种应用程序来操作、使用这些数据。由于数据库是考虑到各种应用的需要而综合设计的，自然可以避免同一数据项的重复，数据冗余度小；数据库技术的主要功能就是有效管理和存取大量的数据资源，除了减少数据冗余以外，还可以提高数据的一致性和完整性；提高数据的共享性，使多个用户能同时访问数据库中的数据；提高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的代价。

为数据库的建立、使用和维护而配置的软件称为数据库管理系统 DBMS（DataBase Management System）。数据库管理系统在操作系统的平台上，利用操作系统提供的输入/输出控制和文件访问功能，对数据库的创建、使用和维护进行管理。还对数据增加了安全机制，防止非法访问，当数据库一旦被破坏时，还可以采取措施，能恢复到最近时刻的状态。而用户又利用数据库管理系统提供的语句编写操作数据库的应用程序，以程序控制的方式操作数据库。如图 1.2 所示。

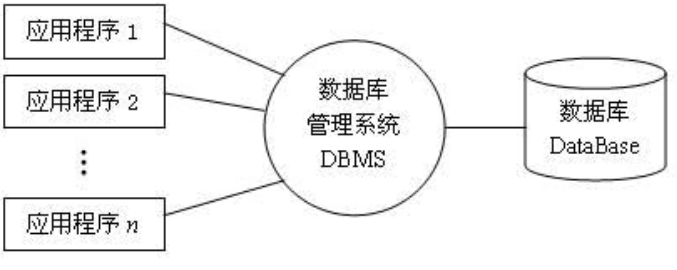


图 1.2 数据库管理方式下数据与程序的关系

4. 分布式数据库系统阶段

分布式数据库系统是数据库技术和计算机网络技术紧密结合的产物。在 20 世纪 70 年代

后期以前，数据库系统多数是集中式的。网络技术的发展为数据库提供了分布式运行环境，使数据库的运行方式从主机／终端体系结构发展到客户／服务器系统结构。

数据库技术和计算机网络技术的结合分为紧密结合与分散结合两大类。因此，分布式数据库就分为物理上分布、逻辑上集中的分布式数据库结构和物理上分布、逻辑上分布的分布式数据库结构两种。

物理上分布、逻辑上集中的分布式数据库结构是一个逻辑上统一、地域上分布的数据的集合，是计算机网络环境中各个节点局部数据库的逻辑集合，同时受分布式数据库管理系统的统一控制和管理，使大部分数据可以就地或者就近存取，而用户不会感到数据的分布。

物理上分布、逻辑上分布的分布式数据库结构是把多个集中式数据库系统通过网络连接起来，各个节点上的计算机可以利用网络通信功能访问其他节点上的数据库资源。在这种运行环境中，各个数据库系统的数据库由各自独立的数据库管理系统集中管理，节点间的数据共享则由双边协商确定。这种数据库结构有利于数据库的集成、扩展和重新配置。

5. 面向对象数据库系统阶段

面向对象的方法起源于程序设计语言的一种认识、描述事物的方法论。面向对象的程序设计是 20 世纪 80 年代引入计算机科学领域的，属于一种新型的程序设计技术，它的发展十分迅猛，影响和涉及计算机科学及其应用的各个领域。

按照哲学上的方法论和认识论的观点，面向对象的方法就是按照人们认识世界和改造世界的习惯方法，对客观事物进行最自然的、最有效的抽象和表达，同时又以各种严格高效的行为规范和机制实施对客观事物的有效模拟和处理，以对象属性描述来表达客观事物，以对象行为特征的方式来表达对客观事物的操作处理，并且把这两者结合成为一个有机的整体，从而把事物完整的内部结构和外部行为机制刻画得淋漓尽致。

面向对象数据库是数据库技术与面向对象程序设计相结合的产物。面向对象数据库是面向对象方法在数据库领域的实现和应用，它既是一个面向对象的系统，又是一个数据库系统。Visual FoxPro 就是一种面向对象的数据库程序设计语言，它不仅仍然支持标准的面向过程的程序设计，而且提供了面向对象程序设计的强大功能和更大的灵活性，是一次程序设计思想和程序设计方法的革命。

1.2 数据库系统

本节将介绍数据库、数据库管理系统、数据库系统等几个相互关联又有区别的基本概念，对于初学者是必须认真加以区分的。

1.2.1 数据库、数据库管理系统、数据库系统

1. 数据库

数据库（DataBase）是存储在计算机存储设备上，具有一定组织形式的、结构化的相关数据的集合。它不仅包括描述事物的数据本身，而且还包括相关事物之间的联系。

数据库中的数据自然不像文件系统管理那样简单，只是面向某一特定的应用，而是面向

多种应用，可以被多个用户、多个应用程序所共享。例如，就一个学校而言，基于一个数据库就可以服务于人事管理系统、学生管理系统、图书管理系统、教务管理系统、财务管理系统等多个应用程序系统。数据结构独立于使用数据的应用程序，数据的增加、删除、修改和检索都由数据库管理系统进行统一控制和管理。

2. 数据库管理系统

为了使数据库中的数据成为具有最小冗余度的共享数据，必须使数据与应用程序具有较高的独立性，这就需要一个专门的软件系统对数据进行管理，提供安全性和完整性的统一控制机制，方便用户对数据库进行操作和维护。

数据库管理系统（DataBase Management System，简称 DBMS）是实现对数据库资源进行有效组织、管理和操作的系统软件。数据库管理系统提供一套数据库应用系统开发的高级语言和相关的开发支持环境，所以 DBMS 也是一种数据库程序设计语言。它的主要功能包括：数据定义、数据库的建立和维护、数据库操作、数据库的运行管理和数据通信等。

目前较有影响的数据库管理系统有 Oracle、Informix、Sybase、SQL Server、Visual FoxPro、Access 等。Oracle、Sybase 等适用于大型的数据库应用系统，Visual FoxPro、Access 适用于中小型数据库应用系统。

3. 数据库应用系统

数据库应用系统是指系统开发人员在数据库管理系统的支持下，利用数据库系统资源开发出来的，面向某一类实际应用的应用软件系统。例如财务管理系统、人事管理系统、教学管理系统、图书管理系统、售票系统、生产管理系统，等等。无论是面向内部业务和管理的管理信息系统，还是面向外部提供信息服务的开放式信息系统，从实现技术角度而言都是以数据库为基础和核心的计算机应用系统。

4. 数据库系统

数据库系统（DataBase System，简称 DBS），它是一个“人—机”系统，由硬件系统、数据库、数据库管理系统、数据库应用系统、数据库管理员和用户组成。其层次关系如图 1.3 所示。

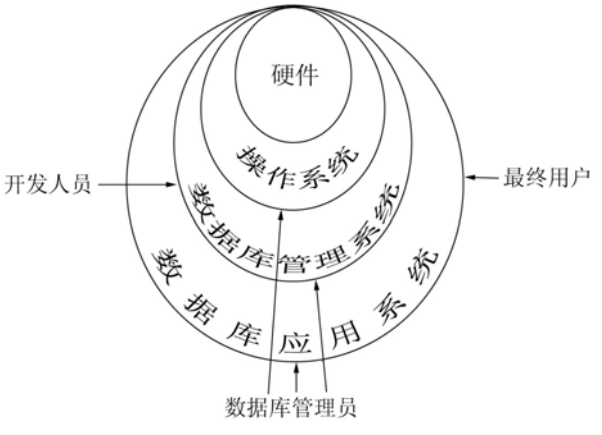


图 1.3　数据库系统的层次关系

从上面的层次关系图可以看出，在数据库管理系统的支持下，创建的数据库存储在计算

机外存中，用户一般通过数据库应用系统操作数据库。开发人员在数据库管理系统的支持下，创建数据库应用系统。数据库管理员通过数据库管理系统实现对数据库的更新和备份、数据库系统的维护和用户的管理。没有数据库管理员，数据库系统创建成功以后，也很难正常运行。

另外，从软件平台上也体现出三个层次：数据库应用系统在数据库管理系统的支持下运行，而数据库管理系统又必须在操作系统的平台上才能运行。

可见，数据库、数据库管理系统、数据库系统是既相互联系又有区别的三个不同的概念。简而言之，数据库是存储数据的仓库，数据库管理系统是创建数据库、操作数据库、管理数据库的工具，数据库系统是一个“人—机”系统，包括装载有数据库、数据库管理系统、数据库应用系统的计算机系统以及使用操作数据库系统的人。

1.2.2 数据库系统的特点

数据库系统具有如下主要特点：

1. 实现数据共享，减少数据冗余

在数据库系统中，数据与应用程序完全独立，由数据库管理系统统一管理。由于数据的最小访问单位是字段，既可存取某一字段或某一组字段的数据，也可以存取一条或一组记录的数据，加上建库时以全局的观点组织数据，自然就实现了数据共享，减少了数据冗余。

2. 采用特定的数据类型

在数据库中，事物内部各数据项之间的联系由数据库结构体现，而这种结构是由数据管理系统所支持的数据模型表现出来的。数据库系统还可以表示事物与事物之间的联系，从而反映出现实世界事物之间的联系。因此，任何数据库管理系统都支持一种抽象的数据模型。

3. 具有较高的数据独立性

在数据库系统中，利用数据库管理系统的映射功能，采用三级模式结构管理数据库，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，使得用户能逻辑地处理数据，而不必关心数据在存储器上的物理位置与结构。

4. 有统一的数据控制功能

在数据库被多个用户或应用程序共享时，往往是多个用户同时使用同一数据库。此时，数据库管理系统对数据提供了必要的保护措施，包括并发访问控制功能、数据的安全控制功能和数据的完整性控制功能。

1.3 数据库的类型

数据库都是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的。人们把客观存在的事物以数据的形式存储到计算机中，这就要把对客观事物特性的认识，概念化到数据库中的具体表示。要完成将事物及其特征转换到数据库中的概念，有必要讨论实

体、实体属性、实体间的联系和反映实体间联系的数据模型。

1.3.1 数据模型

1. 实体及其属性

（1）实体

客观存在的并且可以相互区别的事物称为实体。实体可以指实际的事物，也可以指某些概念和抽象的事件；可以指事物本身，也可以指事物之间的联系。

例如，单位的职工，学校的学生，图书馆的图书等属于实际中的事物实体；而借阅图书、比赛活动等就是比较抽象的事件实体。

（2）实体属性

描述实体的特性称为实体属性。这些特性是实体所固有的，它因描述的角度不同而有所选择。

例如，职工实体可以用职工号、姓名、性别、籍贯、出生日期、职称或职务、简历等若干属性来描述；图书实体可以用编号、分类号、书名、作者、出版社、单价等属性来描述；学生实体可以用学号、姓名、性别、籍贯、出生日期、院系等属性描述。

2. 实体间的联系及联系的种类

实体之间存在着对应关系，这种对应关系称之为联系。它反映现实世界事物之间的相互关联，是客观存在的。例如，一位读者可以借阅若干本图书，而同一本书可以相继被几个读者借阅；一位学生可以选修若干门课程，而一门课程往往有几位教师讲授等。

两个实体间的联系通常可以归结为三种类型：一对一联系、一对多联系和多对多联系。

（1）一对一联系（one-to-one relationship）

考察学校和校长两个实体，如果一个学校只有一个校长，并且一个校长不能同时在其他学校再兼任校长。在这种情况下，学校和校长两个实体之间就存在一对一（1:1）的联系。

（2）一对多联系（one-to-many relationship）

现在，一个大学都有若干学院，一个学院设若干个系，一个系有许多学生。这样大学与学院、学院与系、系与学生都是一对多(1:m)联系。同样，一个部门有多名职工，而一名职工只能在一个部门任职，或者说只占一个部门的编制，则这个部门与职工两个实体之间也是一对多(1:m)联系。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形。

（3）多对多联系（many-to-many relationship）

一个学生可以选修多门课程，一门课程由多个学生选修，学生与课程两个实体之间就存在多对多(m:m)联系。同样一个读者可以借若干本图书，同一本书可以相继被多个读者借阅，因此图书与读者也是多对多联系。

3. 数据模型

为了反映实体本身及实体之间的各种联系，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呈现出数据之间的联系。

可见，数据模型是数据库中数据组织的结构和形式。它是数据库管理系统用来表示实体及实体间联系的方法。自然，一个具体的数据模型要能正确地反映出数据之间存在的整体逻辑关系。

数据库采用三种数据模型来反映实体之间的各种联系：层次模型、网状模型、关系模型。目前最常用的是关系模型。

（1）层次数据模型

用树型结构表示实体及其之间联系的模型称为层次数据模型。它适用于反映实体间一对多(1:m)联系。在这种模型中，数据被组织成由“根”开始的一棵倒挂的“树”，由根开始沿着不同的分支把每个实体放在不同的层次的结点上。树的结点是实体。上级结点与下级结点的连线就是树枝，体现实体之间的联系。如果一个结点不再向下分支，那么此分支序列中最后的结点称为“叶”。如图 1.4 所示。

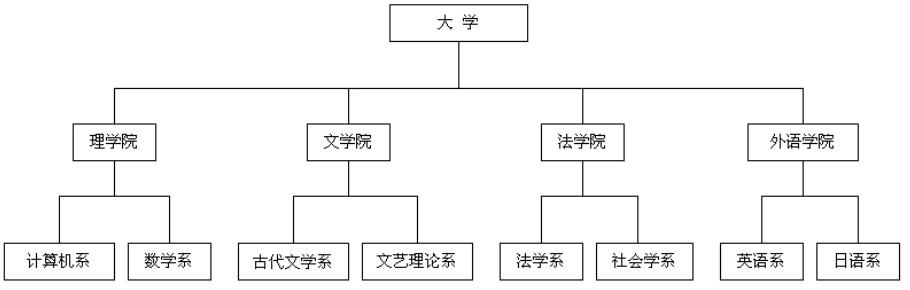


图 1.4 层次模型示意

从图中的大学与院系的层次结构可以看出，大学是根结点，系结点没有再向下分支，就是叶结点，其余都是中间结点。大学与学院、学院与系都是一对多联系。

（2）网状数据模型

用网状结构表示实体及其之间联系的模型称为网状数据模型。它适合于反映实体间的多对多(m:m)联系。在这种模型中，网中的每个结点代表一个实体，允许结点有多于一个的父结点，也可以有一个以上的结点没有父结点。如图 1.5 所示。

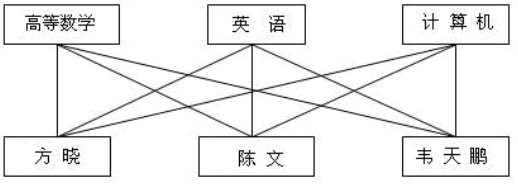


图 1.5 网状模型示意

上图是一个课程与学生选课的示意图，一门课程有多个学生选修，而一名学生要选修多门课程，它体现了课程与学生这两个实体之间的多对多联系。

（3）关系数据模型

用二维表结构来表示实体及其实体之间联系的模型称为关系模型。它把实体间的联系以及实体内部的联系都用一张二维表来表示，这种表就称为“关系”。在二维表中，描述实体的数据本身能够自然地反映实体之间的联系，而层次模型与网状模型是用链接指针来存储和体现实体间联系的，如表 1.1 所示。

表 1.1 关系模型示意

学号	姓名	英语	高等数学	大学计算机	马克思主义
0501101	方晓	80	85	90	90
0501112	陈磊	85	86	95	85
0502205	韦天鹏	78	80	85	80
0503106	冯思齐	85	85	90	90
0502318	陈文	88	85	88	90
0502318	王振兴	82	80	85	90
0501225	何峰	86	85	90	88
0503109	叶博文	88	86	95	95
0505226	王文涛	86	87	89	80
0505231	陈华娟	88	86	92	95

表 1.1 是一个学生成绩表，它反映了学生与课程成绩之间的联系。表中的一行数据用来描述学生实体的课程成绩的属性，而成绩的属性值又反映了学生与课程成绩的联系。

关系数据模型是以关系数学理论为基础的，它对数据库的理论和实践产生很大的影响，成为当今最流行的数据模型。

1.3.2 数据库的类型

根据构建数据库所使用的数据模型，数据库也就相应地分为层次型数据库、网状型数据库和关系型数据库。

由于关系型数据库有严格的关系数学理论基础，简单灵活，数据独立性高，与其他数据库相比有突出的优点。所以，数据库技术发展至今，绝大多数数据库都是关系型的。

任何一个数据库管理系统都是基于某一种数据模型的，相应地就有层次型数据库管理系统、网状型数据库管理系统和关系型数据库管理系统。

使用上述三种数据库管理系统开发出来的应用系统相应地称为层次型数据库系统、网状型数据库系统和关系型数据库系统。

1.4 关系数据库

自 20 世纪 80 年代以来，所推出的数据库管理系统几乎都支持关系数据模型。Visual FoxPro 就是一种关系型数据库管理系统。本节将结合 Visual FoxPro 集中介绍关系数据库的基本概念和基本操作。

1.4.1 基本概念

1. 关系数据库

关系数据库就是用关系数据模型创建的数据库。

在关系数据库中，一个关系就是一个二维表。换句话说，在 Visual FoxPro 中，一个“表”就是一个关系，一个学生表即学生关系，一个成绩表就是成绩关系。表 1.2 就是一个学生情况表，简称学生表。