

计算机等级考试丛书

谭浩强 主编

Visual FoxPro 程序设计
(二级) 教程

周山芙 编著

清华大学出版社

内 容 简 介

Visual FoxPro 是 FoxPro 系列中新一代数据库应用系统的代表,它一改 FoxPro 面向过程的程序设计方式为面向对象的程序设计方式,成为小型数据库管理系统的杰出代表。Visual FoxPro 有许多不同版本,3.0 是最早的版本,比较成熟的版本是 5.0,而 6.0 是目前最高的版本。本书则以 Visual FoxPro 6.0 版本作为讲述的内容。

本书是根据教育部考试中心新制定的《全国计算机等级考试 考试大纲》中关于二级考试数据库语言程序设计部分的要求编写的。本书以初学者为主要对象,以 Visual FoxPro 的基本内容为基础,详细介绍了 Visual FoxPro 系统的基本概念、基本操作及基本编程方法。本书既适合参加等级考试的读者进行系统的学习,也适合初学者自学。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: Visual FoxPro 程序设计(二级)教程

作 者: 周山芙 编著

出版者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

[http:// www .tup .tsinghua .edu .cn](http://www.tup.tsinghua.edu.cn)

印刷者: 北京通州区大中印刷厂

发行者: 新华书店总店北京发行所

开 本: 787 × 1092 1/16 印张: 21.75 字数: 497 千字

版 次: 2002 年 5 月第 1 版 2002 年 9 月第 4 次印刷

书 号: ISBN 7-302-05177-1/ TP · 3037

印 数: 15001 ~ 20000

定 价: 28.00 元

序

跨入 21 世纪,我国已掀起了第三次计算机普及高潮。在这次高潮中将向一切有文化的人普及计算机知识和应用。随着社会主义市场经济的发展,近年来面向社会和面向学校的各种计算机考试如雨后春笋般涌现。许多人认为,学历是从整体上反映了一个人的知识水平,而证书则反映了一个人在某一方面的能力。证书制度是学历制度必要的补充,符合人才市场的需要,因而受到各方面的欢迎。

在众多的计算机考试中,由国家教育部考试中心主办的“全国计算机等级考试”是最权威、影响最广、最受欢迎的一种社会考试。自 1994 年推出了“全国计算机等级考试”以来,至 2003 年底,累计已有 1000 多万人报名参加考试,其中 300 多万人获得了等级证书。不少单位已经把通过全国计算机等级考试作为任职或晋升的条件。

全国许多地区和部门也组织了本地区或本系统的计算机统一测试。考试内容和方法大多与全国计算机等级考试类似。

随着计算机应用技术的发展,教育部考试中心进行了广泛调查研究和征求意见,适时调整了考试大纲。全国计算机等级考试(2002 大纲)分为四个等级(将执行到 2004 年底):

一级,一级 **B**(Windows 环境):要求具有计算机的初步知识和使用微机办公软件的初步能力。

二级:要求具有计算机软件、硬件的基础知识和使用一种高级语言编制程序、上机调试的能力。包括以下内容:

二级 QBASIC

二级 FORTRAN

二级 C

二级 FoxBASE

二级 Visual Basic

二级 Visual FoxPro

(可从中任选一种应试)

三级:要求具有计算机应用基础知识和计算机硬件系统或软件系统开发的初步能

力。包括以下内容:

- 三级 PC 技术
- 三级 信息管理技术
- 三级 网络技术
- 三级 数据库技术

四级: 要求具备深入而系统的计算机知识和较高的计算机应用能力。

2005 年, 教育部考试中心将对一、二级科目进行调整, 取消三个科目:

- 二级 QBASIC
- 二级 FORTRAN
- 二级 FoxBASE

新增四个科目:

- 一级 WPS Office
- 二级 Java
- 二级 Access
- 二级 C++

并对原有的一级和一级 B 科目统一规范为一级 MS Office 和一级 B, 其环境改为 Windows 2000。二级所有科目对基础知识作统一要求, 使用统一的基础知识大纲。二级上机考试中将取消对 DOS 部分的考核。

为了帮助广大应考者准备考试, 我们于 1998 年和 2002 年分别根据当时的新大纲编写和出版了一套“计算机等级考试丛书”, 由清华大学出版社出版, 很受读者欢迎。根据考试内容的变化, 我们对本丛书进行了必要的调整和补充。该丛书由以下四个系列构成:

(1) 计算机等级考试教程: 全面而系统地介绍考试大纲所规定的内容。

(2) 计算机等级考试辅导: 用来帮助已学过该课程的读者复习和准备考试, 每本书的内容均包括各章要点、各章难点、例题分析和思考题, 并附有模拟试题。

(3) 计算机等级考试样题汇编: 按照计算机等级考试的内容和试题形式, 提供了 800~1000 道样题, 供应试者选用。

(4) 计算机等级考试上机指导: 帮助考生熟悉上机考试环境、分析上机试题并提供模拟上机试题供考生练习。

本丛书不仅适用于全国计算机等级考试, 也适用于内容相似的其他计算机统一考试, 对大中学生和其他计算机学习者也有一定的参考价值。

丛书中各书的作者都是高等学校或计算机应用部门中具有丰富教学经验并对计算机等级考试有较深入研究的教授、专家。相信该丛书的出版一定会受到广大准备参加计算机等级考试的读者的欢迎。

欢迎读者对本丛书提出宝贵意见, 以便不断完善。

“计算机等级考试丛书”主编

谭浩强

2004 年 1 月

前 言

当今的时代是一个信息的时代,计算机日益成为人类生活、工作、学习必备的工具和助手。利用计算机进行一些简单的文字处理等操作,还不是计算机应用的主流。利用计算机进行大量数据信息的处理才是计算机应用的主流。怎样才能对大量的数据信息进行及时地、高效率地采集、存储、传输、分析统计、共享和调用呢?必须有先进的数据组织和管理技术支持。这个先进的数据管理方法就是数据库技术。特别是在当前的网络信息时代,以爆炸形式产生的大量数据信息,更加突出了这一重要性。因此学习和掌握基本的数据库原理和技术,自然是每一个人必不可少的学习内容。

随着计算机技术的发展,计算机的应用也更加深入、广泛。新推出的计算机软件不断地更新换代,特别是计算机的基本操作平台——操作系统的变化更是改朝换代。广大计算机用户早就从 DOS 时代进入了 Windows 时代。当然所有在 DOS 平台上使用的软件也必然变成了必须在 Windows 平台上使用的软件。

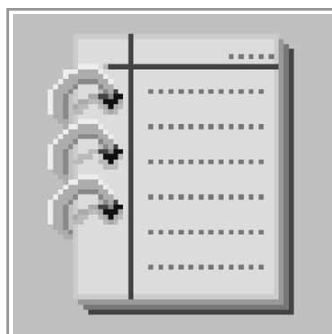
本书正是根据教育部考试中心公布的最新的二级(数据库语言程序设计部分)考试大纲的要求编写的。除了可以供参加二级数据库考试的读者使用外,也适用于高校或各种专业学校非计算机类专业的学生学习使用。本书还适合作为一般管理人员或技术人员的自学用书。

本书力图深入浅出、由浅入深、循序渐进,使读者在最短的时间内掌握 Visual FoxPro 的基本内容,解决一些实际工作中的问题。为了适应考生参加等级考试的需要,本书已经将初学者较难理解的概念,复杂程序的设计方法、设计技巧,以及特殊功能的使用等都尽量简化或排除。有兴趣提高自己实际编程能力的读者,可以在掌握这些基本知识和能力的基础上,再学习和参考其他书籍。

全书由谭浩强教授主持和指导,由中国人民大学周山芙老师具体编写。由于时间紧迫,本人的水平和能力有限,书中一定会存在许多不足之处,敬请广大读者及专家指正。

作者

2002 年 1 月



第 1 章

Visual FoxPro 基础知识

Visual FoxPro 是 FoxPro 系列中新一代数据库管理系统的代表,它一改 FoxPro 面向过程的程序设计方式为面向对象的方式,成为小型数据库管理系统的杰出代表。Visual FoxPro 有许多不同版本,3.0 版本是市场上最早见到的,比较成熟的是 5.0 版本,而 6.0 版本是目前最高的版本。本书将以 Visual FoxPro 6.0 版本作为讲述的内容。

1.1 数据库的基本概念

1.1.1 微机数据库技术的发展

在当今的信息社会里,信息资源已经成为最重要的社会资源之一。人们的社会活动(生产、交流、生活等)都离不开数据信息。而对数据的采集、储存、分析加工、检索使用和维护工作更是我们每天自觉或不自觉地进行着的大量繁琐工作。例如:每天的工作、活动日程安排,个人(或家庭)的收支账目、股票的升落变化等等,更不用说生产管理、办公管理中大量事务性处理工作,无处不存在,无时不进行着大量的对数据的处理和管理工作。因此建立相应的信息系统是社会、企业、部门,甚至是家庭或个人生存和发展的重要支柱。

随着人类社会的不断发展、进步,必然引起数据信息量的飞速膨胀、发展,并且对数据处理的精度、速度也会不断提出更高的要求。为了满足这种不断增长的要求,我们不但需要更先进的计算机,而且需要更先进的数据组织与管理技术。于是在 20 世纪 60 年代后期,一种新的数据管理方法——数据库技术出现了。

数据库系统是在文件系统的基础上发展起来的。由于数据库具有数据结构化、高度共享、冗余度低、程序和数据相互独立、易于扩充、易于编制应用程序等优点,所以一出现便得到了迅速的发展。目前国内开发使用的绝大多数管理信息系统都是以数据库为基础的。数据库的应用范围已经从一般的事务处理扩展到计算机辅助设计、人工智能、软件工程、电子设计自动化(EDA)、办公室自动化、多媒体等计算机应用的各个领域。

早期的数据库管理系统以集中式应用为基础,所有的应用都局限于某一固定的计算



机系统上集中运行。这种系统资源消耗大,对硬件系统依赖性强,使得大型数据库系统只能在大型机上运行,限制了数据库系统的应用发展。20世纪80年代以后,微型计算机的迅猛发展很快使以往只能在大型机上运行的数据库管理系统同样可以在微机上运行,其中最有代表性的是 Oracle、Sybase、Informix 等。但是像 FoxBASE、FoxPro、Access 等一些简单的 PC 数据库管理系统仍然是我国各种传统的数据库管理系统的主体,大多数的用户也是从此步入数据库系统的大门的。

随着计算机技术特别是网络技术的发展和用户对数据库应用系统要求的变化,一种以网络为基础的新型计算机体系结构——Client/Server(客户/服务器)体系结构已经使用得越来越普遍了。这是一种由网络联结的多台硬件组成的协同工作环境。该系统巧妙地将硬件做了分工,服务器专门用来存储共享数据及事务处理过程,客户机用来实现用户的应用程序;这种分工充分发挥了不同硬件的特点,有助于用户利用微机系统建立一个分布式的、既支持联机事务处理又具有友好用户图形界面和良好可扩充性的应用系统。在这里,大型数据库管理系统的微机版适合于做数据库服务器,而传统的 PC 数据库管理系统更适合于做客户机的本地数据库系统。

1.1.2 数据库的基本术语

1. 数据和信息

信息至今尚未有一个统一的、确切的定义。在不同的研究领域中对信息的内涵有不同的理解,于是形成了不同的定义和描述。

从广泛意义上说,信息是对事物运动状态和特征的描述,它反映的是某一客观系统中某一事物的属性或表现形式。例如:一辆飞驰的汽车,是一个客观事物。为了描述这个事物,人们会说:看见一辆银灰色捷达轿车在高速行驶。这种描述就是一般意义的信息。为了在计算机中存储和处理这个事物,必须把它的属性和特征抽出,当然,人们会根据处理的需要每次只提取部分必要的特征。例如:车牌、车名、型号、颜色、车速,并且可以描述成如下形式:

(京 A111111,捷达,GTX,银灰,80km/h)

这些记录信息的物理符号就是数据,这样一组数据的集合叫做一条记录。数据所反映的客观事物的属性是它的内容,而表示数据内容的符号则是它的形式。

数据形式通常有三种:数字型数据,即对客观事物进行定量记录的符号,如体重、年龄、价格的多少等;字符型数据,即对客观事物进行定性记录的符号,如姓名、单位、地址的标志等;特殊型数据,如声音、视频、图形、图像等。从计算机的角度看,数据泛指那些可以被计算机接受,并能够被计算机处理的(数字化)符号。在数据库中数据是存储和管理的基本对象。

同时,信息也定义为是人类认识了的数据,是数据的含义;通俗地讲,信息是经过加工处理并能对人类客观行为产生影响的数据表现形式。

2. 数据处理

所谓数据处理实际上是对各种类型数据进行加工的处理操作,及把处理过的数据合理组织、存储,随时为用户服务的管理操作。数据处理包括对数据进行采集、整理、存储、



加工、传输等;数据管理包括对数据进行分类、编码、组织、存储、检索、维护等。

数据处理的目的是不但要从大量的、原始的数据中获得我们所需要的资料并提取有用的数据成分,而且要管理好这些数据信息,以便人们能够随时提取和使用它们。所以,数据采集是基础,数据管理是核心。

随着电子计算机软件和硬件技术的发展,数据处理过程发生了划时代的变革,而数据库技术的发展,又使数据处理跨入了一个崭新的阶段。数据处理技术的发展经历了以下三个阶段。

(1) 阶段 1: 人工管理方式

在人工管理方式下,没有专门的软件对数据进行管理,在程序中既要考虑处理过程,又要考虑数据的定义和组织,程序和数据总是联系在一起。它只适合早期主要应用于数据量很少的科学计算。其中程序和数据之间的对应关系如图 1.1 所示。



图 1.1

(2) 阶段 2: 文件系统方式

文件系统方式将数据从程序中分离出来,组成相互独立的数据文件。文件系统建立了数据文件内部的数据结构,每个程序都通过自己的文件系统和相应的数据联系,每个文件系统都管理着某个程序需要的数据。但是数据文件之间没有联系,或者说文件系统在整体上是无结构的。如果需要这种联系,只能靠建立专门的程序来实现。它适用于科学计算及简单数据管理。其中程序和数据之间的对应关系如图 1.2 所示。

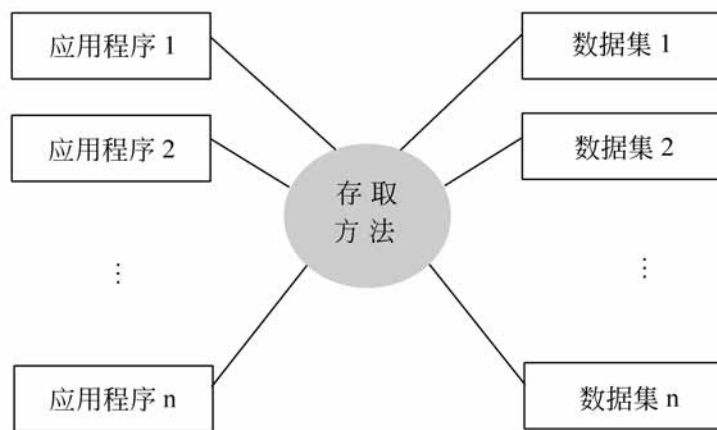


图 1.2

(3) 阶段 3: 数据库系统方式

在数据库系统方式下,一方面可以实现数据与程序的完全独立,另一方面又实现了数据的统一管理。众多程序或应用需要的各种数据,全部交给数据库系统管理,大大压缩了冗余数据,实现了多用户、多应用的数据共享。其中程序和数据之间的对应关系如图 1.3 所示。

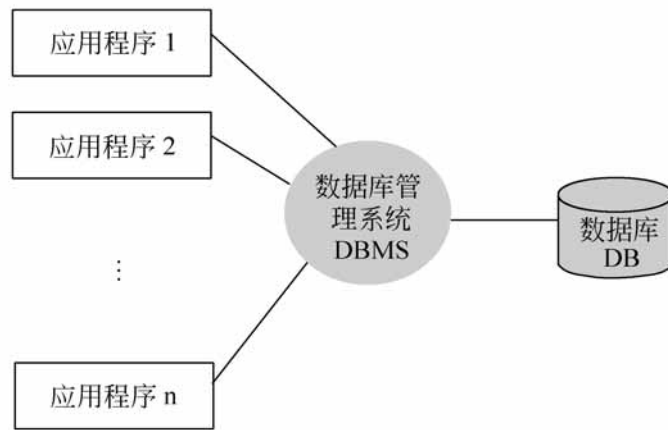


图 1.3

3. 数据库

数据库(data base,简称 DB)可以定义为:以一定的组织方式将相关数据组织在一起并存储在外存储器上所形成的、能为多个用户共享的、与应用程序彼此独立的一组相互关联的数据集合。

数据库不是根据某个用户的需要,而是按照信息的自然联系构造数据;它能以最佳的方式、最少的冗余,为多个用户或多个应用共享服务。

4. 数据库管理系统

数据库管理系统是指帮助用户建立、使用和管理数据库的软件系统,简称为 DBMS(data base management system)。数据库管理系统是数据库系统的核心,DBMS 通常由下列三个基本部分组成。

(1) 数据描述语言 DDL(data description language)

用来描述数据库、表的结构,供用户建立数据库及表。

(2) 数据操作语言 DML(data manipulation language)

供用户对数据表进行数据的查询(包括检索与统计)和存储(包括增加、删除与修改)等操作。

(3) 其他管理和控制程序

实现数据库建立、运行和维护时的统一管理、统一控制,从而保证数据的安全、完整,及多用户并发操作。同时完成初始数据的输入、转换、转存、恢复、监控、通讯,以及工作日志等管理控制的实用程序。

在小型的数据库管理系统(DBMS)中,DDL 和 DML 通常合二为一,成为一体化的语言。在比较完善的数据库管理系统中还会增加一些内容,例如在后面要介绍的 Visual FoxPro 数据库管理系统中,还包含 SQL 语言和向导。

· 结构化查询语言 SQL(structure query language)

这是美国国家标准化组织(ANSI)的标准数据语言,是一种非过程化的语言,一般用来建立查询和视图。

· 向导 Wizards

向导是以逐步提示的方式,指导操作者正确、快速地创建操作对象的工具。

5. 数据库系统

数据库系统 DBS(data base system)是指以计算机系统为基础,以数据库方式管理大



量共享数据的综合系统。它一般应当由数据库、计算机软硬件系统、数据库管理系统和用户(最终用户、应用程序设计员和数据库管理员)四个部分构成。人们习惯上常常把数据库系统简称为数据库,但是应当注意和前面说的数据库(相关数据集合)概念相区别。

6. 数据库应用系统

数据库应用系统是在数据库管理系统(DBMS)支持下建立的计算机应用系统,简称为 DBAS。实际上就是一个具体的数据库系统,所以数据库系统和数据库应用系统经常不加以细分。

在应用系统前面加上“数据库”三字,是为了区别于由普通文件管理系统支持的应用系统。一个数据库应用系统通常由:计算机软硬件系统、数据库系统、应用程序和反映实际情况的数据构成。在微机上建立数据库应用系统一般都使用通用的数据库管理系统 DBMS(如 Visual FoxPro),用户只要开发相应的数据库和应用程序就可以了。

随着计算机应用从单机到网络的发展,数据库技术也朝网络应用的方向有了新的发展。其中最主要的一是从单用户应用扩展到多用户应用;二是从集中式应用扩展到分布式应用。早期的微机 DBMS 都是单用户系统,比如 FoxBASE 等,而 Visual FoxPro 是应用于微机网络上的多用户数据库管理系统。

1.1.3 数据库系统的特点

数据库系统和其他数据管理系统相比,有如下一些基本特点。

1. 数据结构化

在文件系统中,独立文件内部的数据一般是有结构的,但文件之间不存在联系,因此从数据的整体来说是没有结构的。数据库系统虽然也常常分成许多单独的文件,并且文件内部也具有完整的数据结构,但是它更注意同一数据库中各文件之间的相互联系,特别能适应大量数据管理的客观需要。

例如,要实现对学生的基本信息的管理,可以建立一个学生档案记录文件。它有完整的内部数据结构,可以完成学生基本数据的存取操作。

学生档案: XSDA

学号	姓名	年龄	性别
----	----	----	----	-------

如果要实现学校开设课程的管理,应当建立一个课程档案记录文件。它的内部也有完整的数据结构,可以方便地完成对课程信息的存取操作。

课程档案: KCDA

课程号	课程名	学分	教师
-----	-----	----	----

同样要实现学生考试成绩的管理,就应该建立一个学期成绩记录文件;为了方便地完成考试成绩的存取,当然它也有完善的内部数据结构。

学期成绩: XQCJ

学号	课程号	成绩
----	-----	----



我们看到,对于每个应用来说,由于数据记录文件提供了完善的数据结构,所以能够很好地实现对数据的操作。但是从某个学校学生信息管理的整体来说,经常还需要综合操作各数据记录文件。比如要查询本学期某学生的外语成绩,单独从哪个文件中也不能得到结果,这是因为各文件之间没有联系。

在数据库系统中,不仅要考虑某个应用的数据结构,还要考虑整个单位各个应用之间的数据结构。在数据库中,上述三个文件被描述成三个数据元素,它们之间的联系被描述为相应的路径,其结构如图 1.4 所示。

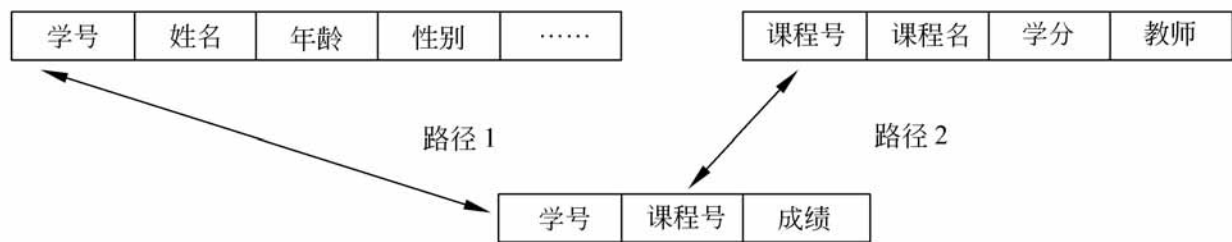


图 1.4

数据元素之间的联系通过存取路径实现,各种复杂的多元素间查询直接用存取路径来完成。所以说数据库系统和数据文件相比,最大的区别就是实现了整体数据的结构化。

2. 数据共享

共享是数据库系统的目的,也是它的重要特点。一个数据库中的数据,不仅可以为同一企业或组织的内部各部门共享,还可以为不同组织、地区、甚至不同国家的用户所共享。而在文件系统中,数据总是由特定用户专用的。

3. 可控冗余度

在文件系统中,由于每个应用都拥有并使用自己的数据,各数据文件中难免有许多数据相互重复,这就是冗余。数据库系统是面对整个系统的数据共享而建立的,各应用的数据集中存储,共同使用;尽可能地避免了数据的重复存储,减少了数据的冗余。为了建立各数据文件之间的联系,不可能消除所有的数据冗余,但是冗余度可以由设计者主动控制。

4. 统一的管理和控制

数据库通过数据库管理系统软件包统一管理数据。由于多用户共享数据,数据库还具有安全性、完整性、并发性控制和数据库恢复功能。

5. 数据独立性

在文件系统中,数据结构和应用程序是相互依赖的,任何一方的改变总是要影响另一方的改变。在数据库系统中,这种相互依赖性是很小的,数据和程序具有相对的独立性。数据库的建立独立于程序,是通过模式来描述的。如图 1.5 所示。

从图 1.5 可以看出,在数据库模式中数据结构具有物理结构和逻辑结构两个方面。描述物理结构的称为物理数据库描述(或存储模式、内模式),它直接与操作系统或硬件联系。一个数据库系统只有一个内模式。

描述逻辑结构的称为模式(或概念模式、全局模式),它是数据库数据的完整表示,是所有用户的公共数据视图。一个数据库系统只有一个模式,它总是以某一种数据模型为基础,统一考虑所有用户的要求,并有机地综合成一个逻辑整体。模式仅仅是数据型的描

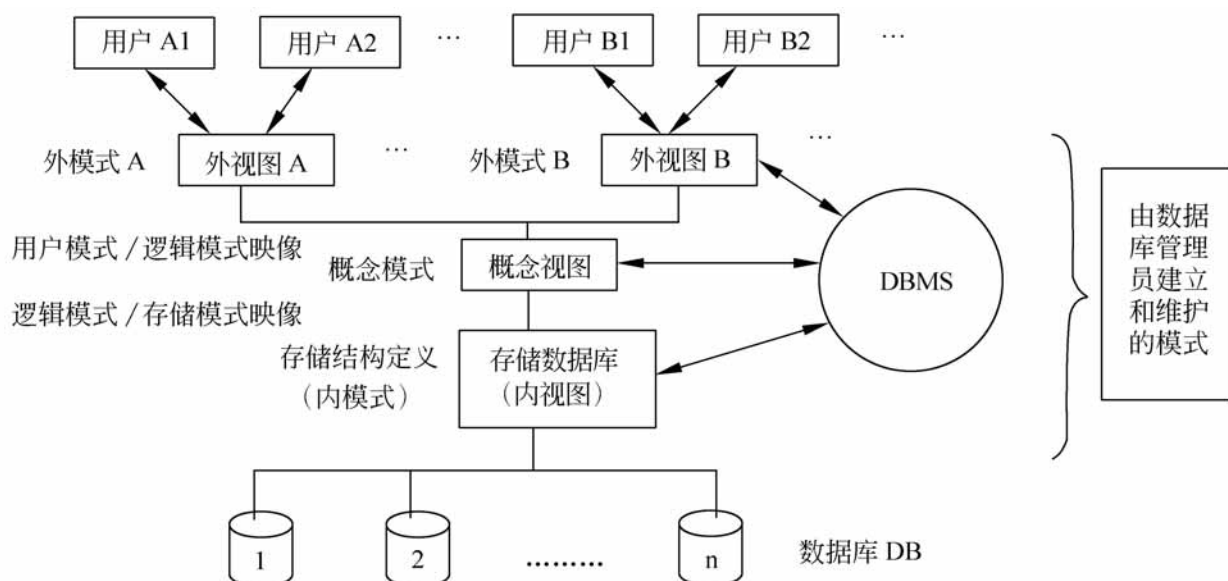


图 1.5

述,不涉及具体数据值。模式的一个具体值称为模式的一个实例,一个模式往往有许多的实例。模式是相对稳定的,而实例是不断变动的。因为模式反映的是数据库的结构,一旦定义好基本就不再变动;而实例反映的是数据库某个时刻的状态,数据库的数据是在不断更新变化的。

针对每一个用户或应用,又由模式导出若干个子模式(或叫外模式)。子模式是直接面向用户的,是用户能够看见并使用的局部数据的逻辑结构描述。每一个子模式都是模式的一个子集;也可以把它看成是模式的一个窗口。一个数据库系统可以有多个子模式。

数据库系统的三级模式中还提供了两个映像功能。一个是在存储结构与逻辑结构之间的映像(转换)功能,另一个是在逻辑结构与用户结构之间的映像(转换)功能。第一种映像使得数据存储结构改变时逻辑结构不变,因而相应的程序也不变,这就是数据库的物理独立性;第二种映像使得逻辑结构改变时,用户结构不变,应用程序也不用改变,这是数据和程序的逻辑独立性。由于这种独立性,使得应用程序的编写再不用考虑数据的描述和存取问题,从而大大减少了应用程序的修改和维护工作。

1.2 数据模型

1.2.1 模型的概念

我们常常会说:“嘿!这个汽车模型太逼真了。”这里说明了两个问题:首先我们看到的不是真实的汽车,其次它又真实地反映了实际汽车的属性和特征。

这种对现实世界事物特征的模拟和抽象就是这个事物的模型。人们对现实世界事物的研究,总是通过对它的模型研究实现的。特别是计算机不能直接处理现实世界的具体事物,所以人们必须先把具体事物转换为抽象的模型,进而转换为计算机可以处理的数据。从而以模拟的方式实现对现实世界事物的处理。所以说在数据库中数据模型是抽象地表示和处理现实世界中数据的工具。

模型应当满足的要求是:一要真实反映现实世界,二要容易被人理解,三要便于在计



计算机上实现。能够真实地反映(模拟)现实世界是根本要求,但既要人容易理解,同时又要计算机便于理解实现,就很不容易了。因此采用逐步抽象的方法,把数据模型划分为两类,以人的观点模拟现实世界的模型叫做概念模型(信息模型),以计算机系统的观点模拟现实世界的模型叫做数据模型。

概念模型和数据模型属于两个层次,为了把现实世界中的具体事物抽象成计算机能够处理的数据和信息,首先按人们的认识观点将现实世界抽象为属于信息世界的概念模型,然后再通过数据模型将信息世界的认识转换为机器(计算机)世界的认识。属于机器世界的数据库模型还可以进一步细分为逻辑数据模型和物理数据模型,它们的层次如图1.6所示。

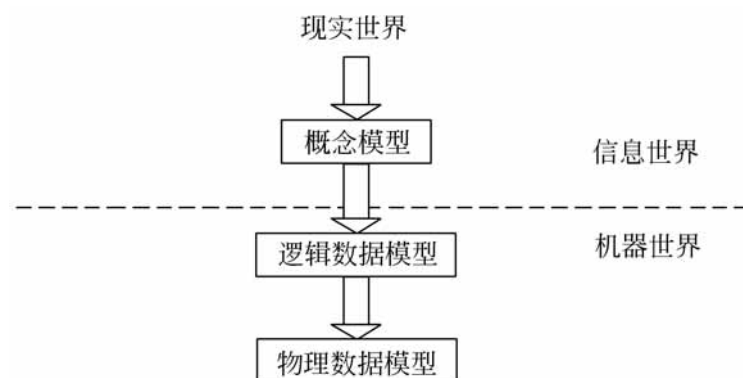


图 1.6

1. 概念模型

概念模型是概念层次的数据模型。概念模型按用户的观点对现实世界建模,是缺乏计算机知识的基本用户最容易理解的,便于和数据库设计人员进行交流的语言。它独立于任何数据库管理系统,但是又很容易向数据库管理系统支持的逻辑数据模型转换。最常见的是实体-联系模型(ER模型)。

2. 数据模型

(1) 逻辑数据模型

逻辑数据模型是用户通过数据库管理系统看到的现实世界,它描述了数据库数据的整体结构。不同的数据库管理系统提供了不同的逻辑数据模型,如层次模型、网状模型、关系模型和面向对象模型等。

(2) 物理数据模型

物理数据模型用来描述数据的物理存储结构和存储方法。不但受数据库管理系统控制,而且与计算机存储器、操作系统密切相关。一般用户是看不到物理数据结构的,所以无需过多了解。

1.2.2 概念模型

概念模型是从现实世界到信息世界的抽象,是数据库设计人员设计数据库的工具,同时又是与用户交流的语言。

1. 基本术语

在概念模型中涉及的主要术语如下。



(1) 实体

客观存在,并且可以互相区别的事物称为实体。它可以是具体的物件,例如,张三、中国人民大学、计算机。也可以是抽象的概念,例如,交通法规。还可以是某种联系,例如,学生的选课、顾客的购物。

(2) 属性

实体具有的每一个特性都称为一个属性。选择的实体属性越多,刻画出的实体越清晰。属性有“型”和“值”的概念,属性的名称(说明)就是属性的“型”;对“型”的具体赋值就是属性的“值”。例如一个计算机实体可以由编号、品名、规格型号、单价、出厂日期等属性型的序列来描述。而属性值(1091,台式机,实达梦飞 II 代,6998.00,2001/5/4)的集合则表征了一个计算机实体的值。

(3) 码

在众多属性中能够惟一标识(确定)实体的属性或属性组称为实体的码。比如计算机实体的码应当是“编号”。

(4) 域

属性的取值范围称为该属性的域。例如,计算机的单价属性的域可以是 0~20000。

(5) 实体型

用实体名及描述它的各属性名,可以刻画出全部同质实体的共同特征和性质,它被称为实体型。例如,计算机(编号、品名、规格型号、单价、出厂日期)就是一个实体型。

(6) 实体集

某个实体型下的全部实体,叫做实体集。例如,某单位目前拥有的全部计算机,就是一个计算机实体集。

(7) 联系

构成实体型的各属性之间是互相联系的,叫做实体内部联系。在实体集之间也存在着联系,称为实体的外部联系。

2. 实体之间的联系

实体之间的联系常常叫做实体型之间的联系,实质上是指实体集之间的联系。这种联系可以分为三类。

(1) 一对一联系

当前实体集中的每一个实体,在另一个实体集中最多只能找到一个可以与它相联系的实体;反过来说,在另一个实体集中的每一个实体,也只能在当前实体集中最多找到一个能够相联系的实体。那么这两个实体集之间就存在着一对一的联系,并记做 1:1。

例如,假定每个教师只能开出一门课程,每门课程只能由一个教师开设;那么“教师”和“课程”的联系,就是一对一的联系。

(2) 一对多联系

当前实体集中的每一个实体,在另一个实体集中可以找到多个能够与它相联系的实体;反过来说,在另一个实体集中的每一个实体,却只能在当前实体集中找到一个能够相联系的实体。那么,这两个实体集之间就存在着一对多的联系,并记做 1:n。

例如,实际上每个教师一般都能开出几门课程,若每门课程只能由一个教师开设;那



么“教师”和“课程”的联系,就是一对多的联系。

(3) 多对多联系

当前实体集中的每一个实体,在另一个实体集中可以找到多个能够与它相联系的实体;反过来说,在另一个实体集中的每一个实体,也能够当前实体集中找到多个能够相联系的实体。那么这两个实体集之间就存在着多对多的联系,并记做 $m \ n$ 。

例如,实际上每个教师一般都能开出几门课程,而每门课程也总是可以由多个教师开设;那么“教师”和“课程”的联系,就是多对多的联系。

实际上一对一联系是一对多联系的特例,而一对多联系又是多对多联系的特例。在 Visual FoxPro(VFP)中总是将一个多对多联系分解为两个一对多联系。例如,增加一个“开课”实体集,将所有教师开出的全部课程实体都集中在这里。因为虽然每个教师可以开多门课程,而一个课堂只能由一个教师上课,所以“教师”和“开课”之间就是一对多联系。同样,一门课程虽然可以由多个教师开设,但同一门课程会有多个课堂,所以“课程”和“开课”也是一对多联系。这样,通过“开课”这个新实体的加入,“教师”和“课程”之间多对多的联系就被分解成两个一对多联系(如图 1.9 所示)。在这个转换的过程中,“开课”实体起到了一种纽带的作用,所以就把它叫做纽带实体。

1.2.3 数据模型

数据模型是数据库系统的核心和基础(指逻辑数据模型),它通常由数据结构、数据操作和数据完整性约束三部分概念组成。数据结构是对系统静态特性的描述,它是数据模型中最重要的部分,所以人们一般以数据结构的类型来命名数据模型。常见的数据模型有层次、网状、关系和面向对象四类模型。

1. 层次模型

层次数据库是最早出现的,如 IBM 的 IMS 数据库管理系统。层次数据库的特点是实体之间按层次关系来定义。实体用记录(类型)表示,实体的属性对应记录的数据项;实体之间的联系用有向连线表示。图 1.7 是一个简单的例子。



图 1.7

层次模型以每个实体为结点,上层结点称为父结点,下层结点称为子结点。层次模型像一棵倒置的树,父结点在上,子结点在下,逐层排列。最上层的结点没有双亲,称为根结点;最下层的结点没有子女,称为叶结点。所以层次模型是一种以记录类型为结点的有向树结构。层次模型表示的是从根结点到子结点的一个结点对多个结点,或从子结点到父结点的多个结点对一个结点的数据间的联系。其主要特征如下:

仅有一个无双亲的根结点。

根结点以外的子结点,向上仅有一个父结点,向下有若干子结点。

2. 网状模型

由于层次模型还不能很好地表达实体间的复杂关系(多对多联系),于是又产生了网状模型,如 HP 的 Turboimage 数据库管理系统。它很好地解决了实体间复杂关系的表达问题,但是它也有致命的弱点,就是当需求扩展时,对原有数据结构及应用程序的修改会产生严重的后果。图 1.8 是网状模型的示意图。

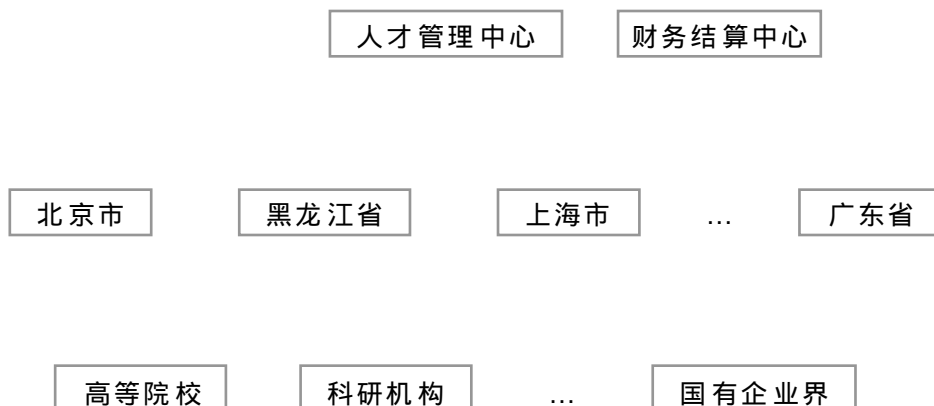


图 1.8

像上面讲到的“教师”和“课程”的多对多联系,用网状模型就可以清楚地表示出来。当然还是应当先增加一个纽带实体“开课”。三个实体之间的关系用网状模型表示如图 1.9 所示。

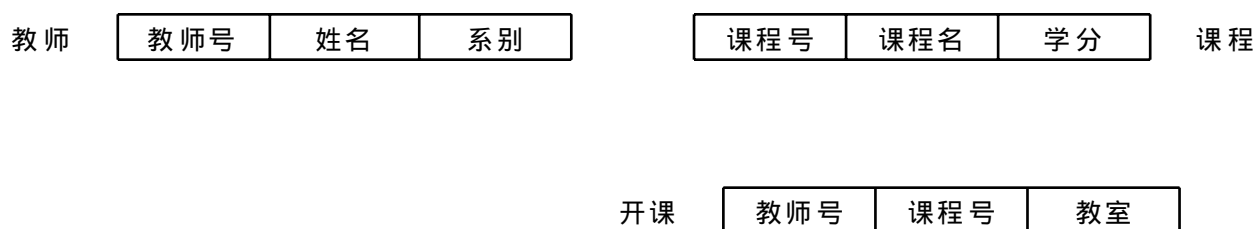


图 1.9

网状模型是层次模型的扩展,表示多个从属关系的层次结构,可以描述比较复杂的数据关系,即可以表示数据间的纵向关系与横向关系,呈现一种交叉关系的网络结构。

其主要特征如下:

有一个以上的结点无双亲。

至少有一个结点有多个双亲。

3. 关系模型

1970年,美国的 E.F.Codd 提出了关系模型的理论,首次运用数学方法来研究数据结构和数据操作,将数据库的设计从以经验为主提高到以理论为指导。不仅如此,关系模型以人们经常使用的表格形式作为基本的存储结构,通过相同关键字段实现表格间的数据联系。关系模型简单明了、易学易用,使它从一开始就吸引了公众的注意,成为广大用户特别是微机用户乐于接受的数据模型。

4. 面向对象模型

不同于层次、网状、关系这些传统的数据模型,面向对象数据模型是非传统的数据模型。将面向对象程序设计方法与数据库技术相结合就产生了面向对象数据库系统。



1.3 关系模型与关系数据库

1.3.1 关系模型

关系模型是一种新的数据模型。它建立在严格的数学理论(集合论和谓词演算公式)的基础上。它提供的逻辑结构简单,数据独立性强,存取具有对称性,操纵灵活。新设计的数据库系统,特别是微型计算机数据库管理系统绝大部分都是关系数据库管理系统。

在数据库中的数据结构如果依照层次模型定义,则该数据库为层次数据库;如果依照网状模型定义,则该数据库为网状数据库;如果依照关系模型定义,数据库系统就是关系数据库系统。关系数据库系统一般都由许多不同的关系构成,其中每个关系就是一个实体,可以用一张二维表表示。例如一张“学生”数据表就是一个关系,它可以如图 1.10 所示。

系别	专业	姓名	性别	年龄
财经系	财经	丙	男	19
信息系	信息管理	甲	女	20
国经系	国际贸易	丁	女	20
计算机系	软件工程	乙	男	21

图 1.10 中的表格标注如下：
 - 属性名：指向表头中的“系别”、“专业”、“姓名”、“性别”、“年龄”等列。
 - 属性值：指向表体中的具体数据，如“财经”、“甲”、“男”等。
 - 框架：指向表头部分。
 - 元组：指向表体中的每一行数据。

图 1.10

其中的关系术语如下。

关系(Relation): 一张二维表对应一个关系。

属性(Attribute): 表中每一列叫做一个属性,属性有名和值的区别。

元组(Tuple): 由属性值组成的每一行叫做一个元组(记录值)。

框架(Framework): 由属性名组成的表头称为框架(关系型)。

分量: 表中每一个属性值称为分量。

域(Domain): 每个属性的取值范围。

候选码(Candidate Key): 可以唯一地确定一个元组的属性或属性组(可简称码),例如表中“姓名”属性。

主码(Primary Key): 一个关系中往往有多个候选码,当前被指定的一个候选码称为主码。

主属性(Primary Attribute): 包含在主码中的属性。

非码属性(Non-key Attribute): 不包含在任何候选码中的属性(也叫非主属性)。

关系模式: 对关系的描述称为关系模式,关系模式常常记做:

关系名(属性 1, 属性 2, 属性 3, ..., 属性 n)

上面的学生关系可以表示为:

学生(系别, 专业, 姓名, 性别, 年龄)