

21世纪网络平台大学计算机系列教材



Visual FoxPro 6.0

# 与面向对象的程序设计

C H E N G X U S H E J I

朱 寄 田 爱 景 主 编



科学出版社

[www.sciencep.com](http://www.sciencep.com)

·21 世纪网络平台大学计算机系列教材·

# Visual FoxPro 6.0 与面向对象的 程序设计

朱 寄 田爱景 主编

科 学 出 版 社

北 京

## 内 容 简 介

本书为 Visual FoxPro 课程教材,内容按照用户使用数据库的逻辑顺序编排,从数据库的基本原理和基本概念出发,介绍数据库的基本操作。在介绍结构化程序设计方法后,通过介绍 Visual FoxPro 可视化程序设计工具的使用方法,引导读者在潜移默化中循序渐进地理解面向对象程序设计的思想。

本书汲取了非计算机专业计算机教育研究的最新成果,以微软公司的 Visual FoxPro 6.0 为背景组织编写。本教材配有实验教材和教学光盘,力求做到有机结合,在内容上形成互补,尽可能满足不同个体的学习需求和不同教学计划的要求,为任课教师和学生提供更多的选择空间,为大幅度提高教学效率奠定基础。

本教材既可作为高校非计算机专业数据库应用课程教材,也可供计算机等级考试应试者以及社会各类计算机应用人员参考。

### 图书在版编目 ( CIP ) 数据

Visual FoxPro 6.0 与面向对象的程序设计/朱寄,田爱景主编. - 北京: 科学出版社, 2004

(21 世纪网络平台大学计算机系列教材)

ISBN 7-03-014127-x

I . V … II . ①朱…②田… III . 关系数据库 - 数据库管理系统, Visual FoxPro 6.0 - 程序设计 IV . TP311.138

中国版本图书馆 CIP 数据核字 (2004) 第 081908 号

责任编辑: 王雨舸

责任印制: 高 嵘 / 封面设计: 张 琴

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

武汉大学出版社印刷总厂印刷

科学出版社发行 各地新华书店经销

2004 年 8 月第 一 版 开本: 787×1092 1/16

2004 年 8 月第一次印刷 印张: 17 1/4

印数: 1 - 8 000 字数: 419 000

定价: 28.80 元 ( 含光盘 )

( 如有印装质量问题, 我社负责调换 )

**《Visual FoxPro 6.0 与面向对象的程序设计》编委会**

**主 编：朱 寄 田爱景**

**副主编：郑德义 张国庆 石 澜**

## 前 言

数据库技术是事务处理和信息管理领域中采用的主要技术手段,随着社会信息化进程的加快,它已经成为现代管理中必须具备的知识与技能。

Visual FoxPro 6.0 是微软公司推出的小型数据库管理系统开发平台,是目前 PC 机上广泛流行的数据库管理系统软件,它以强大的功能、完整而丰富的工具、较高的处理速度、友好的操作界面以及完备的兼容性等特点,备受广大用户的欢迎。Visual FoxPro 6.0 提供了一个集成化的系统开发环境,它不仅支持传统的结构化编程技术,而且支持面向对象的可视化编程技术;Visual FoxPro 6.0 还提供了功能丰富的可视化程序设计工具,从而使得数据的组织、数据库规则的定义和应用程序的开发等工作变得简单方便。正因为 Visual FoxPro 6.0 具有上述特色,所以它不仅是目前较为理想的小型数据库开发平台,同时也成为微机数据库的首选教学软件。

本教材汲取了非计算机专业计算机教育研究的最新成果,以 Visual FoxPro 6.0 为背景组织编写。在教材的编写过程中,作者力图实现如下目标:

第一,教材结构符合教学的认知规律。本教材的内容按照用户使用数据库的逻辑顺序编排,从数据库的基本原理和基本概念出发,介绍数据库的基本操作,在介绍结构化程序设计方法后,通过介绍 Visual FoxPro 可视化程序设计工具的使用方法,引导读者在潜移默化中循序渐进地理解面向对象程序设计的思想。

第二,叙述通俗易懂,方便自学。在教材的写法上既注重概念的严谨和清晰,又尽量避免过多地引入概念,力求采用读者容易理解的叙述方法阐明数据库的基本理论和面向对象的基本概念。

第三,教材形式多样,便于组织教学。本教材包括文字教材以及配套的实验教材和教学光盘,力求作到实验教材与文字教材有机结合,在内容上形成互补,尽可能满足不同个体的学习需求和不同教学计划的要求,为任课教师和学生提供更多的选择空间,为大幅度提高教学效率奠定基础。

本教材既可作为大专院校非计算机专业数据库应用课程教材,也可供计算机等级考试应试者以及社会各类计算机应用人员参考。

为了适应计算机技术高速发展的形势,全面提升在校大学生的信息素养和计算机应用水平,我们对教材从事形式到内容都进行了一些探索,至于是否圆满地实现了上述目标,还有待广大读者作出评判。

本书由朱寄、田爱景任主编,郑德义、张国庆、石澜任副主编。其中第一章、第三章的 3.4~3.5 节和第四章由田爱景编写,第二章、第三章的 3.1~3.3 节以及第十章由张国庆编写,第七章由郑德义编写,第五章、第六章、第八章、第九章、第十一章由朱寄编写,石澜参与了部分章节的编写与编辑工作。全书由朱寄统稿。

由于作者水平有限,对于书中出现的不足和疏漏,恳请广大读者提出宝贵意见。

编者

2004 年 5 月

# 目 录

第一章 概述	1
1.1 数据库基础	1
1.1.1 计算机数据管理的基本概念和发展	1
1.1.2 数据库系统	3
1.1.3 数据模型	4
1.2 关系数据库	9
1.2.1 关系模型	9
1.2.2 关系运算	10
1.2.3 关系的设计	11
1.3 Visual FoxPro 6.0 系统概述	15
1.3.1 Visual FoxPro 6.0 的特点	15
1.3.2 Visual FoxPro 6.0 的运行环境和安装	16
1.3.3 Visual FoxPro 6.0 的运行	19
1.4 Visual FoxPro 6.0 的用户界面	20
1.4.1 Visual FoxPro 6.0 的主界面	20
1.4.2 Visual FoxPro 6.0 的工作方式	20
习题	21
第二章 Visual FoxPro 的基础知识	23
2.1 Visual FoxPro 的数据类型	23
2.2 Visual FoxPro 的常量与变量	23
2.2.1 常量	23
2.2.2 变量	25
2.3 运算符与表达式	29
2.3.1 数值运算符及数值表达式	29
2.3.2 字符串运算符及字符表达式	30
2.3.3 日期型   日期时间型运算符及日期型   日期时间型表达式	31
2.3.4 关系运算符及关系型表达式	31
2.3.5 逻辑运算符及逻辑型表达式	32
2.4 Visual FoxPro 的常用函数	33
2.4.1 数值处理函数	33
2.4.2 字符处理函数	34
2.4.3 日期和时间函数	35
2.4.4 数据类型转换函数	36
2.4.5 测试函数	37
2.5 Visual FoxPro 的命令结构和书写规则	38

2.5.1	Visual FoxPro 的命令结构	38
2.5.2	Visual FoxPro 命令和短语的书写规则	40
2.6	Visual FoxPro 的文件类型	41
	习题二	42
<b>第三章</b>	<b>数据表的基本操作</b>	<b>44</b>
3.1	数据表的建立	44
3.1.1	表结构的设计	44
3.1.2	表结构的建立	45
3.1.3	记录数据的输入	47
3.1.4	通用型字段的操作	47
3.2	表的维护与使用	49
3.2.1	表的打开和关闭	49
3.2.2	显示表的结构和表的数据	50
3.2.3	修改表结构	53
3.2.4	修改记录数据	54
3.2.5	表记录的插入与追加	54
3.2.6	记录的删除与恢复	56
3.2.7	记录指针的移动	57
3.2.8	表结构和数据记录的复制命令	58
3.2.9	使用多个表的操作	60
3.3	项目管理器	63
3.3.1	启动项目管理器	63
3.3.2	项目管理器的操作界面	63
3.3.3	使用项目管理器	64
3.4	表的排序与索引	65
3.4.1	基本概念	65
3.4.2	数据表的排序	65
3.4.3	数据表的索引	66
3.5	查询与统计	73
3.5.1	顺序查询命令	73
3.5.2	索引查询命令	74
3.5.3	统计命令	76
	习题三	78
<b>第四章</b>	<b>数据库操作</b>	<b>81</b>
4.1	创建数据库	81
4.1.1	数据库设计概述	81
4.1.2	数据库的建立与打开	83
4.1.3	向数据库添加或移去数据表	89
4.1.4	创建数据库表	92
4.2	建立表间的关系	94

4.2.1	基本概念	94
4.2.2	表间的临时关联	94
4.2.3	表的连接	96
4.3	数据完整性	98
4.3.1	实体完整性	98
4.3.2	域完整性	99
4.3.3	参照完整性	101
	习题四	103
<b>第五章</b>	<b>结构化查询语言 SQL 基础</b>	<b>105</b>
5.1	SQL 概述	105
5.2	数据定义功能	105
5.3	数据操纵功能	111
5.4	查询功能	112
	习题五	129
<b>第六章</b>	<b>查询与视图</b>	<b>130</b>
6.1	查询	130
6.1.1	查询的概念	130
6.1.2	查询向导	130
6.1.3	查询设计器	131
6.2	视图	139
6.2.1	视图的概念	139
6.2.2	使用视图设计器建立视图	140
6.2.3	视图的打开与关闭	144
6.2.4	建立视图的 SQL 命令	145
	习题六	146
<b>第七章</b>	<b>结构化程序设计基础</b>	<b>147</b>
7.1	VFP 程序文件	147
7.1.1	程序的概念	147
7.1.2	程序文件的建立和编辑	147
7.1.3	程序文件的执行	149
7.1.4	常用环境参数设置命令	149
7.1.5	应用举例	151
7.2	VFP 程序设计中常用的基本命令	152
7.2.1	交互方式命令	152
7.2.2	程序的注释、清屏与基本输出命令	155
7.2.3	格式输入输出命令	155
7.3	VFP 程序的基本结构	158
7.3.1	顺序结构	158
7.3.2	判断选择结构	158
7.3.3	循环结构	162



7.4 数组及其应用	173
7.4.1 数组的概念、定义与赋值	173
7.4.2 数组与数据表记录之间的数据交换	175
7.5 过程与自定义函数	178
7.5.1 子程序及其调用	178
7.5.2 过程与过程调用	180
7.5.3 参数传递	182
7.5.4 自定义函数	183
7.5.5 内存变量的作用域	184
习题七	186
<b>第八章 面向对象程序设计</b>	<b>190</b>
8.1 面向对象程序设计的基本概念	190
8.1.1 概述	190
8.1.2 面向对象的基本概念	190
8.2 Visual FoxPro 中的基类	191
8.3 Visual FoxPro 中对象的属性、方法和事件	193
习题八	196
<b>第九章 表单设计</b>	<b>197</b>
9.1 利用表单向导创建简单的表单	197
9.2 表单设计器	200
9.2.1 启动表单设计器	200
9.2.2 表单设计器的操作界面	200
9.2.3 利用表单设计器设计表单	202
9.2.4 表单的保存、修改和运行	204
9.2.5 表单的常用属性、事件与方法	204
9.2.6 设置数据环境	205
9.3 常用控件的设计	208
9.3.1 控件的相关操作	208
9.3.2 标签	209
9.3.3 文本框与编辑框	209
9.3.4 命令按钮与命令按钮组	212
9.3.5 列表框与组合框	215
9.3.6 复选框与选项按钮组	218
9.3.7 表格	220
9.3.8 页框与页	222
9.3.9 容器	224
9.3.10 计时器	224
9.3.11 微调按钮	225
9.3.12 ActiveX 控件	225
9.4 表单的高级设计	226

9.4.1 表单集.....	226
9.4.2 多重表单.....	228
9.4.3 用户自定义属性和方法.....	230
9.4.4 用户自定义类.....	231
习题九.....	234
<b>第十章 菜单设计.....</b>	<b>236</b>
10.1 菜单设计的基本步骤.....	236
10.1.1 菜单的基本结构.....	236
10.1.2 菜单设计的基本步骤.....	236
10.1.3 菜单设计器.....	237
10.1.4 “显示”菜单.....	239
10.2 下拉式菜单设计实例.....	240
10.3 快捷菜单设计实例.....	243
习题十.....	244
<b>第十一章 报表与标签设计.....</b>	<b>245</b>
11.1 报表设计.....	245
11.1.1 报表设计的基本内容.....	245
11.1.2 利用报表向导创建报表.....	245
11.1.3 利用报表设计器创建报表.....	246
11.1.4 创建分组报表.....	251
11.1.5 创建多栏报表.....	252
11.2 标签设计.....	253
习题十一.....	254
<b>参考答案.....</b>	<b>256</b>

# 第一章 概 述

数据库技术是计算机科学的一个重要分支。数据库具有冗余度较低、数据结构化与独立性高、容易扩充和操纵等特点。较大的信息系统都是建立在数据库基础之上的。无论在大、中、小型计算机,还是在微型计算机上,数据库管理系统作为通用的系统软件都已成为必备的基本配置。本章简单地介绍数据库系统的基本概念。

## 1.1 数据库基础

### 1.1.1 计算机数据管理的基本概念和发展

#### 1. 信息、数据和数据处理

(1) 信息 (information)。信息是现实世界中各种事物的特征、形态和不同事物之间的联系在人类头脑中的抽象反映。换言之,信息是经过加工的有用数据。这种数据有时能产生决策性的影响。例如,一个企业如果不组织人力进行市场调查,收集、整理和分析信息,企业的决策者就无法做出正确的决策,久而久之,该企业就会在竞争中被淘汰。

(2) 数据 (data)。数据是存储在某种媒体上的、能够被识别的符号。数据用型(存储在媒体上的数据形式)和值(描述数据特性的数据内容)来表征与记录事物。从广义上讲,数据是一切文字、符号、图形、图像、动画、影像、声音等元素的有意义的组合。例如,数字0~9、大小写英文字符、汉字等都是符号,“电脑桌长 1.7 米”就是这些符号中元素的一个有意义的组合。所以说,数据是信息的载体,信息是数据的内涵。信息依赖数据来表达,数据则具体地表现信息,即“信息与数据是既紧密联系又相互区别的一对术语”。同一信息可以有不同的数据表示形式(例如,某种统计信息既可用表格表示,也可用直方图表示);而同一数据也可能有不同的解释(例如,同样是数据“1”与“0”,在计算机中它们有时被解释成一个二进制位,有时被解释为逻辑“真值”与“假值”)。

尽管信息与数据两个术语严格地讲是有区别的,但在很多场合下,不严谨地区分它们也不致引发误解。因此,使用中很多时候都不严格区分这两个术语。

(3) 数据处理 (data processing)。数据处理是指对数据的收集、记载、整理、组织、存储、检索、计算/加工、维护、传送等一系列活动的总和,其目的是从已知数据出发,参照各种相关数据,经过计算/加工,得到新的数据。

新的经过精炼的数据表示新的信息,反映事物的本质和特征,以及事物间的内在联系,是有价值、有意义的信息,成为人们决策的依据,这一过程就叫做信息处理。由于信息是用数据表示的,所以信息处理具体地体现在数据处理中。通过处理数据可以获得信息,通过分析和筛选信息可以产生决策。

数据处理的特点是数据量大,数据间的逻辑关系错综复杂;但计算相对地比较简单,较少涉及复杂的数学模型和计算方法。因此,数据处理过程中的主要矛盾是如何把数据组织管理好。数据库技术正是针对这一目标,为有效地管理数据、最大限度地减轻程序员的负担、提高数据处理的效率而逐渐完善起来的计算机软件技术。

## 2. 数据管理技术的发展

数据处理的中心问题是数据管理。数据管理包括对数据的收集、组织、编目、定位、存储、检索、统计和维护等。计算机数据管理技术的发展大体上经历了三个阶段：人工管理阶段、文件系统阶段、数据库系统阶段。

(1) 人工管理阶段。20 世纪 50 年代中期以前，计算机主要应用于科学计算。硬件方面，计算机的外存只有磁带、卡片和纸带等；软件方面，只有汇编语言，没有操作系统，没有数据管理的软件。其管理方式是：

① 数据不保存在计算机中：在程序运行时输入数据，结果输出后数据与程序都不存入计算机。

② 没有专用的软件对数据进行管理：数据管理由人工操作，程序员除编写程序外还要组织数据。

③ 数据不具有独立性：数据在存储上稍有改变，如增、删、改数据，都必须修改程序。

④ 数据不能共享，冗余度大：程序与数据一一对应，即使两个程序使用同一组数据，也要各自输入，造成大量的重复数据。

(2) 文件系统阶段。20 世纪 50 年代后期至 60 年代中期，硬件方面有了磁盘、磁鼓，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代替了人工管理数据，其特征如下：

① 数据有专门的软件——文件系统管理：数据的变更由文件系统负责维护，不一定修改程序。

② 数据具有一定的独立性，可长期保存：文件系统把数据组织成文件形式在磁盘上保存。

③ 数据共享性差、冗余度高：数据文件仍然与程序对应，同一组数据在多个文件中重复存储。

④ 数据的不一致性：同一数据出现在多个文件中，修改时易造成同一数据在不同文件间的差异。

⑤ 数据联系弱：文件间缺乏联系，因而就不能反映现实世界中事物间的联系。

(3) 数据库系统阶段。该阶段始于 20 世纪 60 年代中后期，硬件方面有了成本低、容量大、速度快的磁盘，软件方面，1968 年美国 IBM 公司推出层次模型的 IMS 系统，1969 年美国数据系统语言协会 (CODASYL) 的数据库任务组 (DBTG) 发表了网状模型的 DBTG 报告，1970 年，E. F. Codd 发表的多篇论文奠定了关系数据库的理论基础，使数据库技术迅速发展。该阶段的特征如下：

① 用数据模型表示数据的结构，使数据结构化：数据库中的文件相互联系，遵循一定的结构形式，通过文件间的联系，较好地反映了现实世界事物间的自然联系。

② 数据充分共享：数据库中的数据不是面向某个应用，而是面向所有用户的数据需求和整个应用系统，同一数据可供多个用户共享。

③ 数据冗余度低：在数据库系统中，用户使用数据库中的数据。该数据以逻辑文件的形式提供给用户，同一数据物理上只存储一次，使用时被映射到不同的逻辑文件里，从而减少了冗余。

④ 数据独立性高：数据独立性就是使数据与程序无关，数据存储方式的改变不影响应用程序。数据库的结构分三级：用户的逻辑结构、整体逻辑结构和物理结构。数据独立性分两级：物理数据独立性和逻辑数据独立性。数据库三级结构的转化过程是：

数据库的物理结构  $\xrightarrow{\text{通过映射}}$  数据库的整体逻辑结构  $\xrightarrow{\text{再次映射}}$  用户的逻辑结构  
当数据库物理结构发生改变时，修改映射，使数据库整体逻辑结构不受影响，进而使用户的逻

辑结构及应用程序不用改变,达到了物理数据的独立性;当数据库的整体逻辑结构发生改变时,修改映射,使用户的逻辑结构及应用程序不必改变,达到数据库的逻辑数据的独立性。

⑤ 方便的用户接口:数据库管理系统作为用户与数据库的接口,提供数据库的定义、运行、维护等功能。用户可使用查询语言或终端命令操作数据库,也可用程序方式操作数据库,如图 1.1 所示。

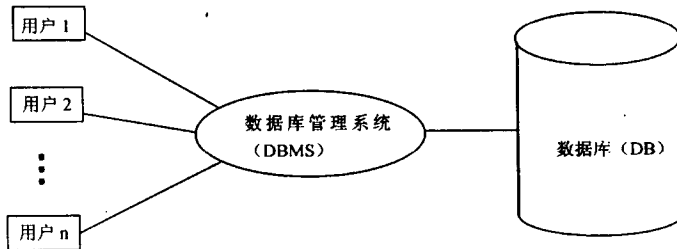


图 1.1 DBMS 作为用户与数据库接口的示意图

⑥ 数据的统一管理:数据库系统可提供数据控制功能,包括数据库的恢复、并发控制、数据安全性和数据完整性。此外,对数据的操作既可按记录(行)进行,也可按数据项(列)进行,增加了操作的灵活性。

从文件系统发展到数据库系统,使信息处理的一些传统观点发生了重大变化。在文件系统阶段,程序处于主导地位,数据只起着服从程序设计需要的作用;在数据库系统下,数据占据了中心位置,而使用这些数据的程序则退到以数据库中既定的数据结构为基础的地位。

现在,数据库的应用已经深入到人类社会活动的各个领域。从 20 世纪 70 年代后期至今,数据库技术与网络技术结合,采用 client/server 系统结构,产生了分布式数据库系统。本书所介绍的 Visual FoxPro 6.0 就是在 ODBC(开放式数据库互连)支持下与数据库相连。近些年来,数据库技术与面向对象程序设计相结合,产生了面向对象数据库系统,是面向对象方法在数据库领域中的实现和应用。Visual FoxPro 6.0 既支持标准的结构化程序设计,也提供了面向对象程序设计的强大功能和更大的灵活性。

## 1.1.2 数据库系统

### 1. 数据库、数据库管理系统和数据库系统

(1) 数据库(DB, database)。数据库是以一定的格式存放在计算机存储设备上的、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。数据库中的数据实行统一管理,按一定的数据模型描述和储存,具有较少的冗余度,数据间既密切联系又具有较高的数据独立性和易扩充性,并可为各种用户共享。

(2) 数据库管理系统(DBMS, database management system)。DBMS 是数据管理的核心软件,是介于操作系统和用户之间的系统软件,用于对数据库中的数据实行专门管理,提供安全性和完整性等统一控制机制,方便用户以交互命令或程序方式对数据库进行操作;是用户与数据库的接口,提供数据库的定义、建立、检索、更新、维护及各种数据控制。按照所处理的数据模型,可将 DBMS 分为层次型、网状型、关系型和面向对象型数据库管理系统。

(3) 数据库应用系统(DBAS, database application system)。DBAS 是利用数据库资源开发出来的、面向某一类实际应用的软件系统。一个数据库应用系统通常由数据库和应用程序

两部分组成,它们是在数据库管理系统支持下设计和开发出来的。

(4)数据库系统(DBS, database system)。数据库系统是引进数据库技术后,实现有组织、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,方便多用户访问的计算机软件、硬件和数据资源的计算机系统。一个数据库系统通常由5部分组成,包括计算机硬件系统、数据库集合、相关软件、数据库管理系统、数据库管理员(DBA, database administrator)和用户。数据库系统中各层次之间的相互关系如图1.2所示。

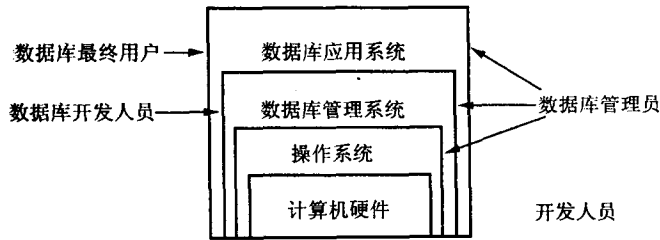


图 1.2 数据库系统层次示意图

(5)数据库技术。这是一门研究数据库的结构、存储、管理的软件学科。深入学习需要离散数学、数据结构、操作系统、软件工程等理论知识,所以是一门综合性较强的学科。

(6)数据库理论与技术研究的方向。当前数据库理论与技术研究的主要方向是:数据仓库、演绎数据库系统、多媒体数据库、图形数据库系统和多数据库系统集成技术。

## 2. 数据库系统的特点

(1)实现数据共享。允许多个用户同时存取数据而不互相影响。

(2)实现数据独立。组织数据库中的数据时面向全局,程序员编制程序时,只以简单的逻辑结构来操作数据,使得应用程序不随数据存储结构的改变而变动。

(3)减少了数据的冗余度。在数据库中,同一个数据在物理上只存储一次,使用时再把它映射到不同的逻辑文件里,有效地节省了存储资源。

(4)避免了数据的不一致性。数据只有一个物理存储,对数据的访问、修改不会出现差异。

(5)加强了对数据的保护。数据库的保密机制,可防止对数据的非法存取;进行集中控制有利于数据的完整性;采取并发访问控制保证了数据的正确性;实现了对数据库破坏后的恢复。

### 1.1.3 数据模型

数据模型(data model)是对客观事物及其关系的数据描述。数据模型不仅要表示存储了哪些数据,更重要的是用某种结构形式表示各种不同数据之间的关系。为了反映事物本身及事物之间的各种关系,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。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,即数据模型是数据库的框架。

根据数据模型描述数据的能力(是描述一般的文字、数值,还是声音、图形以至对象)和描述数据间关系的结构方式(层次、网状、关系)将其分类。数据库管理系统支持的比较成熟的数据模型有层次模型、网状模型、关系模型以及面向对象模型。

#### 1. 数据与数据间关系的描述

从客观事物到抽象概念再到计算机的存储方式,实际上涉及三个领域:现实世界→信息世

界→机器世界。在不同的领域里,人们使用不同的名词术语对数据进行描述。

(1) 现实世界。现实世界是客观存在的世界,事物及其相互关系是数据的发源地。例如,学生管理分为在校生人数、毕业生人数、学生成绩等,这些都是现实世界中最原始的数据。

(2) 信息世界。信息世界指现实世界中的事物在人类头脑中的抽象反映。客观事物在信息世界中称为实体,实体模型反映了事物的相互关系。信息世界中的常用术语如下:

① 实体(entity):实体是客观存在的可以相互区别的事物。实体可以是实际的人或物(如一名学生、一张桌子),也可为概念性的东西(如一场球赛)以及事物间的关系(如一次聚会)。

② 实体型(entity type)与实体值(entity value):实体用型(type)和值(value)来表征。型是概念的内涵,值是概念的实例。通常使用实体名和属性名的集合来描述实体型。如学生实体的实体型可描述为:学生(学号,姓名,出生日期)。实体值是实体的实例,是属性值的集合。如张三是一个学生实体,张三实体的值是“04008026,张三,86-01-18”。

③ 属性(attribute):实体所具有的特征称为实体的属性。例如,学生的学号、姓名、性别等都是学生实体的属性。每个属性都有一个值域(domain),属性的取值类型和取值范围称为值域,取值类型可为数值型、字符型等。例如,性别属性的值域,就是字符型的“男”或“女”。属性也用型和值表征,“性别”就是该属性的型,具体的值“男”则是属性值。

④ 实体集(entity set):具有相同性质的同类实体的集合称为实体集。例如,一个班的所有学生,一个实验室里的所有计算机等。

⑤ 实体标识符(identifier):能够惟一标识每个实体的属性或属性集的符号称为实体标识符。例如,学生的学号可以作为学生的实体标识符,一个学号惟一地对应一个学生。

(3) 机器世界。信息在机器世界中以数据形式存储,因此,机器世界又称数据世界。数据世界是信息世界中信息的数据化,现实世界中的事物及其相互关系在机器世界用数据模型描述。因此,客观事物是信息之源,是设计数据库的出发点,也是使用数据库的最终归宿。实体模型与数据模型是对客观事物及其相互关系的两种抽象描述。数据库的核心问题是数据模型,为了得到正确的数据模型,首先要充分了解客观事物。

## 2. 实体联系

实体的联系有两种:一是实体内的联系,反映在数据上是同一记录中各字段间的联系;二是实体间的联系,反映在数据上是记录与记录间的联系。在数据库系统中要解决如何描述联系、实现联系、处理联系等问题。这里主要讨论实体间最基本的联系方式。

两个不同实体集的实体间有三种关系:一对一关系、一对多关系、多对多关系。设有两个实体集E1与E2,下面分别用E1和E2为例说明三种关系。

(1) 一对一关系(one to one relationship)。若E1中的每个实体至多和E2中的一个实体有联系;反之,E2中的每个实体至多和E1中的一个实体有联系,则称E1与E2的关系是一对一关系,记为1:1。例如,一个学校只有一个校长,一个校长不能同时在其他学校再兼任校长,学校和校长之间就是1:1关系。

(2) 一对多关系(one to many relationship)。若E1中的一个实体与E2中多个实体有联系,而E2中的每个实体至多和E1中的一个实体有联系,则称E1对E2的关系是一对多关系,记为1:N。例如,一个学校同时可有許多学生注册,一个学生同一时间只能在一个学校注册,学校和学生之间就是1:N关系。

(3) 多对多关系(many to many relationship)。若E1中的每个实体与E2中的多个实体有联系;反之,E2中的每个实体和E1中的多个实体有联系,则称E1与E2的关系是多对多关系,

记为 M:N。例如,一个学生可以选多门课程,每一门课程可以有多个学生选修,学生和课程之间就是 M:N 关系。

### 3. 实体联系模型

数据常用的模型有两类:信息模型(又称概念数据模型)和结构数据模型。信息模型不涉及信息在计算机内的表示和处理等问题,纯粹用来描述信息的结构。这类模型要求表达的意思清晰,应当正确地反映出数据之间存在的整体逻辑关系,即使不是计算机专业人员也很容易理解。在实际的数据库系统开发过程中,信息模型是用户和数据库设计人员之间的交流工具。

典型的信息模型是 P. P. Chen 于 1976 年提出的实体关系模型(entity relationship model, 简称 ER 模型)。ER 模型通过 ER 图表示实体及实体间的联系。设计数据库时,先用 ER 图准确地反映信息,再把 ER 图与计算机系统和数据库管理系统

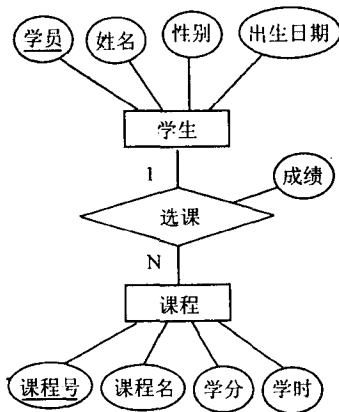


图 1.3 学生与课程联系的 ER 图

结合,构造实际的数据模型。ER 图有 4 个基本成分:矩形框表示实体;菱形框表示实体间的联系;椭圆形框表示实体或实体间联系的属性;直线用来连接上述 3 种图形。

画图时,把相应的名称填入框中,对组成关键字的属性加下划线,在菱形框的引出线上标出联系的方式(即注明是 1:1、1:N 或 M:N 关系)。图 1.3 就是一个学生实体与课程实体联系的 ER 图。

ER 图不是惟一的,根据强调的侧面不同做出的 ER 图可能有很大差别。ER 图只描述实体间的联系,不能在计算机中直接实现,是现实世界的第一层抽象。结构数据模型则是第二层抽象。结构数据模型直接面向数据库的逻辑结构,有严格的形式化定义,便于在计算机系统中实现。

### 4. 结构数据模型

结构数据模型是能够在计算机中实现的模型,它有严格的形式化定义。其模型一般有 3 个组成部分:数据结构(解决数据和实体及实体间的联系如何表达,怎样实现等问题);数据操作(实现对数据库的检索和更新操作,其中更新包括插入、删除、修改等);数据完整性约束(数据模型必须给出数据及联系应具有制约和依赖规则,从而保证数据库中的数据是正确的)。

在计算机中实现结构数据模型,需用某种计算机语言来描述。模型的 3 个组成部分都有具体的语句表达,经过编译,数据模型将以计算机能够接受的形式存于计算机内。数据模型中的数据结构部分就是数据库的框架。该框架形式化地描述数据库的数据组织形式,在框架的约束下填入具体数据就形成了数据库。下面介绍结构数据模型的数据结构特性。

(1) 层次模型(hierarchical model)。用树形结构表示实体及实体间联系的模型称为层次模型。树的节点是实体。节点间是一种上下级关系或称父子关系。树上有且仅有一个节点无父节点,称为根节点,其他节点均有且只有一个父节点;上下层实体是 1:N 关系;整个模型就像一棵倒立的树,按照层次模型的结构框架,填入具体数据就组成层次数据库。层次数据库中,与实体对应的部分称为记录。记录值之间的联系在模型的约束下依然保持树形。图 1.4 所示的就是一个层次模型的示例。层次模型有如下的特点:

- ① 记录间的联系通过指针实现:检索效率较高,但查找记录时要指定存取路径。
- ② 层次分明:适合表示现实世界中有级别、分层次、相互间有 1:N 关系的事物。
- ③ 不能直接表示 M:N 关系:为使层次模型能处理 M:N 关系,需要提供多种辅助手段,但



很复杂,不易掌握,加之层次顺序严格,使得编写操纵数据的应用程序十分复杂。层次模型的典型代表是 1968 年美国 IBM 公司研制的 IMS (information management system) 数据库管理系统。

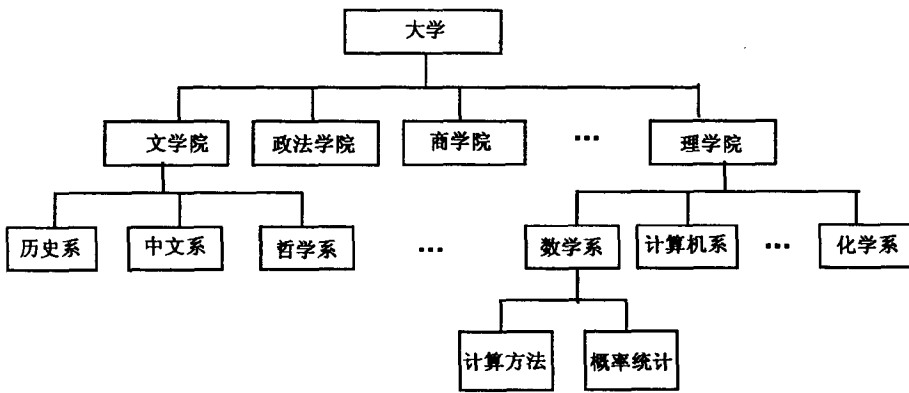


图 1.4 层次模型示例

(2) 网状模型 (network model) 用网状结构的有向图表示实体及实体间联系的模型称为网状模型。有向图中的节点是实体,有向边 (即箭头) 表示从箭尾的实体到箭头的实体是 1:N 关系。网状模型的结构是节点的连通图。与层次模型相比,有向图中可能有多个节点无父节点,有些节点又可能有多个父节点。如图 1.5 所示,节点“计科系”和“外文系”都没有父节点;而节点“教师”、“课程”、“学生”、“任课”和“选课”都有两个父节点。网状模型具有表达各种复杂关系的能力。

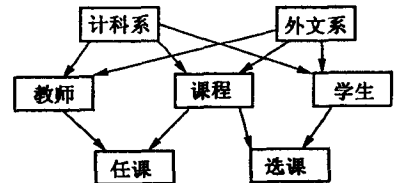


图 1.5 网状模型示意图

网状模型的特点是:实体间的联系通过指针实现,检索效率较高,表示多对多关系比较灵活。但这种灵活性是以数据结构复杂化为代价的。程序员必须非常熟悉数据库的逻辑结构,编写应用程序也很复杂。网状模型和层次模型都存在难以实现系统的修改与扩充等缺陷。关系模型具有概念简单、容易理解、编程方便等特点。因此,从 20 世纪 80 年代中期开始,关系模型的产品很快就占领了市场。

(3) 关系模型 (relation model)。用二维表结构表示实体及实体间关系的模型称为关系模型。在关系模型中,操作的对象和结果都是二维表,这种二维表称为关系。数据库技术中常用的术语之一是关系模式 (relation schema, 关系的描述称为关系模式)。关系可用表格或者数学式子来描述。当用表格描述时,关系模式相当于二维表格的框架。在这个框架下填入数据,就组成了关系模式的一个实例。一般地讲,关系可形式化地表示为:关系名 (属性名 1, 属性名 2, 属性名 3, ..., 属性名 i, ...), 一个关系可具有多个属性。例如,在图 1.3 的 ER 图中,包含两个关系:学生 (学号, 姓名, 性别, 出生日期) 和课程 (课程号, 课程名, 学分, 学时)。对于学生关系,“学生”是关系名,“学号、姓名、性别、出生日期”是关系的属性名;对于课程关系,“课程”是关系名,“课程号、课程名、学分、学时”是关系的属性名。填入实体数据后用表格表示如表 1-1 和表 1-2 所示。

显然,一个关系就是一张二维表。关系名对应表名,关系的属性名对应表中的栏目名,在数据库表中称为字段名;表中的一行对应一个实体,在数据库表中称为记录;表中的列对