

- 数据库系统概述
- Visual FoxPro 6.0基础知识
- 数据表的创建和基本操作
- 数据的设计与操作
- 结构化程序设计
- 面向对象程序设计
- 表单的建立与使用
- 结构化查询语言SQL
- 视图与实用表单开发
- 报表与标签技术
- 菜单技术
- 应用系统集成技术

普通高等学校文科专业计算机系列教材

Visual FoxPro 6.0 程序设计基础

卢湘鸿 主编



清华大学出版社

普通高等学校文科专业计算机系列教材

Visual FoxPro 6.0

程序设计基础

卢湘鸿 主编

李吉梅 等编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书是根据教育部高等教育司组织制订的《普通高等学校文科类专业计算机基础课程教学大纲(2000 版)》的要求编写的。

本书以 Visual FoxPro 6.0 为背景,介绍了关系数据库管理系统的基础理论及系统开发技术。主要内容包括数据库系统概述、Visual FoxPro 6.0 基础知识、数据表的创建和基本操作、数据库的设计与操作、结构化程序设计、面向对象程序设计、表单的建立与使用、结构化查询语言 SQL、视图与实用表单开发、报表与标签技术、菜单技术和应用系统集成技术等 12 部分,配有丰富的例题、习题(包括上机练习),并附有解答。

本书可作为普通高等学校各专业计算机公共课的教材,还可作为计算机等级考试培训教材,也可供从事数据库开发的人员学习、参考。

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。

图书在版编目(CIP)数据

Visual FoxPro 6.0 程序设计基础/李吉梅等编著. —北京:清华大学出版社,2002

普通高等学校文科专业计算机系列教材

ISBN 7-302-05646-3

I. V… II. 李… III. 关系数据库—数据库管理系统, Visual FoxPro 6.0—高等学校—教材 IV. TP311.138

中国版本图书馆 CIP 数据核字(2002)第 056669 号

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>

印刷者:北京市清华园胶印厂

发行者:新华书店总店北京发行所

开 本:787×1092 1/16 印张:24.75 字数:568 千字

版 次:2002 年 10 月第 1 版 2002 年 10 月第 1 次印刷

书 号:ISBN 7-302-05646-3/TP·3328

印 数:0001~5000

定 价:30.00 元

《普通高等学校文科专业计算机系列教材》序

21 世纪将是以信息科技和生命科技为核心的科技进步与创新的世纪,也是继续深刻改变人类的生产和生活方式、高速推动世界文明发展的世纪。在信息化社会中,计算机文化知识已成为人类知识结构中不可或缺的重要组成部分;计算机是对人的一生都有重大用处的“通用智力工具”。使用计算机已成为人类必需的技能,那些不能掌握这一技能的现代文盲将会步履艰辛。“人才培养,计算机教育必须先行”早已成为全社会的共识。

因此,我国在中小学中都设置了以计算机使用为主要内容的信息技术课程。但高中毕业生中的多数近期还不能达到中学《计算机教学大纲》(信息技术课指导意见)的要求,所以在他们进入大学后仍然需要在计算机的应用方面进行整合,以便进一步提高。而且在中学阶段教授的计算机知识,也不可能满足大学生对计算机知识的基本需要。因此在相当长的时期内,大学各专业仍有开设计算机公共基础课的必要,当然其起点和内容会有新的变化。

高校专业的涉及面很宽,根据教育部高教司的有关规定,文科类涵盖的学科门类大体上就有 7 门 19 类。繁多的学科门类,自然对计算机知识有着不同的需求。比如,艺术类、体育类学生与经济类学生所需的计算机知识,就有着很大的差别。所以计算机教学还必须充分考虑到信息化社会对不同专业的特殊需要。

有鉴于此,根据教育部高等教育司组织制订的《普通高等学校文科类专业计算机基础课程教学大纲(2000 年版)》的公共基础和专业提高两部分的基本精神和要求,我们组织编写了普通高等学校文科专业计算机系列教材,由教育部高等学校文科计算机教学指导委员会副主任卢湘鸿教授担任主编。该系列教材既包括了可供文科各专业使用的计算机基础课程的教材,还提供了能够满足文史哲、经济、政法、体育、艺术各类专业特殊需要的计算机教材。

计算机科学技术的发展日新月异,文科计算机教学也存在着继续探索和不断积累经验的过程,因此对系列教材中的不足或错误,敬请同行和读者批评指正。

编 者

2002 年 8 月

于北京

前 言

虽然我国在中小学设置了以计算机使用为主要内容的信息技术课程,但高中毕业生的计算机水平不一,多数近期还不能达到中学《计算机教学大纲》(信息技术课指导意见)的要求,他们学过的应用软件进大学后很需要整合。加之大学涉及的学科专业宽广,中学开设的计算机知识自然不能满足大学生的基本需求。因此,大学各专业仍有在中学计算机课基础之上开设计算机公共基础课的必要。此外,还必须立足于信息化社会对大学各专业毕业生在计算机应用方面的需求来考虑对他们的培养。

大学生在毕业后的工作和生活中都离不开在计算机网络平台上对文字、表格、图形、图像、声音、动画等数据的处理,这就是需要具有在某一操作系统的环境下应用办公软件和计算机网络来为工作和生活服务的能力。因此,在网络平台上对计算机的应用,就成为计算机基础课教学的基础或核心内容;网络既是这门课教学的基本手段,也是教学的基本内容。具体地说,根据由教育部高等教育司组织制订的《普通高等学校文科类专业计算机基础课程教学大纲(2000)》(简称《大纲》)的要求,计算机基础课程教学内容总的要求应包括以下模块:

- (1) 计算机及信息处理基础知识;
- (2) 操作系统;
- (3) 办公软件(文字、表格、声音和图像处理);
- (4) 计算机网络基础与应用;
- (5) 多媒体基础;
- (6) 计算机维护基础(包括常用工具软件);
- (7) 数据库系统基础(数据库技术);
- (8) 程序设计基础。

上述内容总体上可分为两部分。第一部分是计算机应用基础,这可以由操作系统、办公应用软件和计算机网络应用组成。第二部分是数据库和程序设计基础。比如各校可酌情选学 Visual FoxPro 数据库技术与程序设计、数据库软件 Access 应用或 Visual BASIC 语言程序设计中的一种。

本书是根据《大纲》的基本要求,以微软 Visual FoxPro 6.0 为背景编写的。Visual FoxPro 6.0 是优秀的小型数据管理系统软件,具有强大的数据库管理系统功能,提供了面向对象程序设计的各类开发工具。该软件不仅可以用来开发小型数据库系统,而且可以广泛用于大型数据库的前端开发,在这些方面它可与 Visual Basic、Power Builder 等软件相媲美。

本书先从数据库基本原理、概念出发,介绍数据表,以及数据库对象的建立、查看、修改、使用与维护等操作,然后在学习结构化程序设计的结构与基本方法后,由浅入深地引入面向对象程序设计思想。

本书充分注意到系统性和完整性。从最初的用户需求分析,经数据库结构设计,各种表单、报表、菜单,软件集成,到最后的软件发布,除给出了每一步的操作细节,还给出了软件清单。此外还从软件工程角度,举例讲述了数据库应用系统的开发过程和软件质量控制方法。各章配有丰富的例题、习题和上机练习,并附有解答。

本书可以满足 72~108 学时的教学需要,其中 1/2~2/3 学时为上机操作。可分 3 个层次安排,第 1 层次安排 36 学时,以掌握数据表、数据库的基本知识、数据表的创建、修改、排序索引和检索等基本使用为基本内容;第 2 层次安排 72 学时,除第 1 层次规定的内容外,还需掌握结构化程序设计和结构化查询语言 SQL 等内容;第 3 层次安排 108 学时,除第 2 层次规定的内容外,还需掌握面向对象程序设计、表单的建立与使用视图与实用表单开发、报表与标签技术、菜单技术和应用系统集成技术。

本书可以满足高等学校各专业(也包括文科类专业)在计算机公共基础课 Visual FoxPro 6.0 数据库技术与程序设计教学方面的基本需要。还可作为全国计算机等级考试二级 Visual FoxPro 6.0 程序设计的培训教材,也可供数据库开发人员参考。

本系列教材由卢湘鸿组织编写。本书初稿主要由李吉梅编写,参加书中某些内容、习题和解答编写的有:卢湘鸿、李亚弟、陈勇军、罗赛杰、何伟红、卢卫、孟先进、刘佳、喻炜、肖磊等。全书最后由卢湘鸿审定。

本书虽然是作者在数据库教学与开发实践的基础上编写的,但难免会有错误与不足之处,敬请同行和读者批评指正。

编 者

2002 年 8 月

目 录

第 1 章 概述	1	2.3.3 变量	37
1.1 数据库系统概述	1	2.3.4 常见函数及其应用	41
1.1.1 数据库系统基础知识	1	2.3.5 表达式	49
1.1.2 现实世界的数据库描述	3	2.4 本章小结	54
1.1.3 关系模型	6	习题 2	54
1.1.4 关系操作	8	第 3 章 数据表的基本操作	59
1.1.5 关系完整性	8	3.1 Visual FoxPro 项目管理器	59
1.2 Visual FoxPro 6.0 系统概述	10	3.1.1 项目文件的建立	59
1.2.1 数据库管理系统概述	10	3.1.2 项目管理器的界面	62
1.2.2 Visual FoxPro 的发展历史	11	3.1.3 项目管理器的使用	64
1.2.3 Visual FoxPro 的性能指标	11	3.1.4 创建自己的应用程序	66
1.2.4 Visual FoxPro 文件类型	12	3.2 创建自由表	67
1.2.5 Visual FoxPro 的安装、卸载 和启动	13	3.2.1 表的概念	67
1.2.6 书写约定	16	3.2.2 表结构设计	68
1.2.7 引例	17	3.2.3 表结构建立	69
1.3 本章小结	19	3.2.4 记录的键盘输入	75
习题 1	19	3.2.5 将已有数据添加到记录中	79
第 2 章 Visual FoxPro 基础知识	22	3.3 表的修改与使用	81
2.1 Visual FoxPro 用户界面	22	3.3.1 表的打开/关闭	81
2.1.1 Visual FoxPro 窗口	22	3.3.2 查看记录	82
2.1.2 操作方式	22	3.3.3 维护表记录	87
2.1.3 菜单系统	23	3.3.4 修改表结构	93
2.1.4 对话框	26	3.3.5 在项目中添加/移去表	94
2.1.5 常用控件	27	3.3.6 使用多个工作区	94
2.1.6 命令语法规则	28	3.4 表的索引与排序	97
2.1.7 系统设置	30	3.4.1 索引的类型	97
2.1.8 获取帮助	30	3.4.2 索引文件的类型	98
2.2 编辑窗口	31	3.4.3 索引的建立、引用与维护	98
2.2.1 编辑菜单	31	3.4.4 排序	102
2.2.2 格式菜单	32	3.5 查询与统计命令	102
2.2.3 文本操作	32	3.5.1 索引查询命令	102
2.2.4 命令窗口	33	3.5.2 统计命令	103
2.3 基本语法与规定	35	3.6 数据共享	104
2.3.1 数据类型	35	3.6.1 数据共享	105
2.3.2 常量	36	3.6.2 数据导入	105
		3.6.3 数据导出	108

3.7 本章小结	111	5.1.3 程序文件的修改	146
习题3	111	5.1.4 程序文件的执行	146
第4章 数据库的设计与操作	116	5.1.5 程序调试	146
4.1 数据库设计概述	116	5.1.6 程序编辑技巧	147
4.1.1 数据库设计的一般步骤	116	5.2 程序设计的一些常用命令	148
4.1.2 确定数据库的用途	117	5.2.1 基本输出命令“? ?”	148
4.1.3 确定需要的表文件	118	5.2.2 基本输入命令	148
4.1.4 确定需要的字段	118	5.2.3 格式输入输出命令	149
4.1.5 确定各表之间的关系	120	5.3 程序的基本控制结构	149
4.1.6 完善数据库	123	5.3.1 顺序结构	149
4.2 创建数据库	123	5.3.2 分支结构	150
4.2.1 建立数据库文件	123	5.3.3 循环结构	151
4.2.2 在数据库中操作表	125	5.4 过程与用户自定义函数	153
4.2.3 在数据库中新建表	126	5.4.1 过程	154
4.2.4 多表关联	128	5.4.2 用户自定义函数	157
4.2.5 表间临时关联	128	5.4.3 变量作用域	157
4.2.6 表的连接	130	5.5 本章小结	159
4.3 数据库表属性的设置	132	习题5	159
4.3.1 字段显示属性的设置	133	第6章 面向对象程序设计	167
4.3.2 字段输入默认值的设置	134	6.1 面向对象编程概述	167
4.3.3 有效性规则	135	6.1.1 从面向过程到面向对象	167
4.3.4 “触发器”设置	136	6.1.2 对象	168
4.3.5 表名和表注设置	137	6.1.3 类	169
4.3.6 参照完整性	137	6.1.4 多态和虚拟方法	170
4.4 数据库操作	138	6.2 Visual FoxPro 中的类和对象	171
4.4.1 数据库的打开/关闭	138	6.2.1 Visual FoxPro 的类概述	171
4.4.2 查看数据库中的表	138	6.2.2 容器与非容器	172
4.4.3 修改与查看数据库结构	140	6.3 面向对象的程序设计方法	173
4.4.4 与数据库操作相关的命令 与函数	140	6.3.1 程序设计方法	173
4.5 在项目中添加/移去数据库	141	6.3.2 对象程序设计	174
4.5.1 添加数据库	141	6.3.3 类与事件程序设计	177
4.5.2 移去或删除数据库	141	6.3.4 类的继承	188
4.6 使用多个数据库	141	6.3.5 事件驱动模型	189
4.6.1 打开多个数据库	142	6.4 本章小结	193
4.6.2 当前数据库的设置	142	习题6	193
4.6.3 作用域	142	第7章 表单的建立与使用	198
4.7 本章小结	142	7.1 表单的设计过程	198
习题4	143	7.2 操作表单	198
第5章 结构化程序设计	145	7.2.1 表单创建与保存	198
5.1 程序的编辑与使用	145	7.2.2 表单属性和方法	199
5.1.1 程序文件的建立	145	7.2.3 表单集	202
5.1.2 程序文件的保存	145	7.2.4 表单文件分析	203

7.2.5 数据环境定义	204	8.2.5 连接查询	252
7.2.6 修改表单	208	8.3 查询结果处理	253
7.2.7 运行表单	208	8.3.1 排序输出	253
7.3 控件应用初步	209	8.3.2 重定向输出(INTO)	254
7.3.1 表单控件	209	8.3.3 输出合并(UNION)	255
7.3.2 创建表单与数据环境	211	8.3.4 分组统计(GROUP)与筛选 (HAVING)	255
7.3.3 标签控件	211	8.4 查询设计器	256
7.3.4 线条与形状控件	211	8.4.1 查询设计器的应用	256
7.3.5 文本框控件	212	8.4.2 交叉表应用	261
7.4 按钮类控件	215	8.4.3 查询结果输出	263
7.4.1 命令按钮和按钮组	215	8.4.4 查询设计器的表单设计	264
7.4.2 单选按钮	219	8.5 本章小结	265
7.4.3 复选框	220	习题 8	265
7.4.4 微调按钮	221	第 9 章 视图与实用表单开发	268
7.5 框类控件	223	9.1 视图设计	268
7.5.1 列表框	223	9.1.1 视图	268
7.5.2 组合框	225	9.1.2 视图设计器概述	269
7.5.3 编辑框	227	9.1.3 视图设计	270
7.5.4 页框	228	9.1.4 视图的应用	276
7.5.5 网格	228	9.1.5 视图的 SQL 语句	277
7.6 其它控件	231	9.1.6 视图与查询、视图与表的比较	278
7.6.1 图像控件	231	9.2 基于视图的表单设计	279
7.6.2 计时(Timer)控件	232	9.2.1 成绩表单	279
7.6.3 超级链接(Hyperlink)控件	232	9.2.2 学生学籍表单	282
7.7 表单的调整和定制	234	9.3 本章小结	283
7.7.1 控件编辑	234	习题 9	284
7.7.2 控件排列	235	第 10 章 报表与标签技术	285
7.7.3 使用自动格式生成器	235	10.1 报表格式设计初步	285
7.7.4 设置 Tab 顺序	235	10.1.1 报表设计器	285
7.8 本章小结	236	10.1.2 页面设置	286
习题 7	236	10.1.3 快速报表	286
第 8 章 结构化查询语言	238	10.1.4 存储与预览报表	287
8.1 SQL 语言概述	238	10.2 报表设计	288
8.1.1 SQL 语言的特点	238	10.2.1 报表格式概述	288
8.1.2 数据定义语言	239	10.2.2 域控件设计	289
8.1.3 数据操纵语言	242	10.2.3 标签控件设计	292
8.1.4 数据控制语言	244	10.2.4 画线设计	293
8.2 SELECT 命令基本用法	244	10.2.5 标题/总结设计	294
8.2.1 基本查询	245	10.2.6 分组设计	297
8.2.2 带条件查询	246	10.2.7 圆角矩形控件设计	298
8.2.3 嵌套查询	249	10.2.8 OLE 控件设计	299
8.2.4 多表查询	250		

10.3 报表输出	301	12.2.2 将文件加入到项目中	331
10.3.1 菜单方式打印报表	301	12.2.3 连编应用程序	333
10.3.2 命令方式报表输出	302	12.3 软件发布	334
10.4 实用报表设计	302	12.3.1 教学管理系统软件发布	335
10.4.1 学生登记卡报表设计	302	12.3.2 软件的安装与删除	336
10.4.2 创建课程表报表	303	12.4 应用程序框架	337
10.4.3 创建成绩单报表	304	12.4.1 应用程序向导	338
10.5 标签设计	305	12.4.2 应用程序生成器	339
10.6 本章小结	305	12.4.3 应用程序框架	340
习题 10	306	12.5 应用软件开发过程概述	346
第 11 章 菜单技术	307	12.5.1 可行性研究阶段	347
11.1 菜单设计的一般步骤	307	12.5.2 需求分析阶段	347
11.2 操作菜单	308	12.5.3 系统设计阶段	350
11.2.1 规划菜单系统	308	12.5.4 实现阶段	356
11.2.2 菜单创建	308	12.5.5 测试运行阶段	360
11.2.3 生成程序	312	12.6 本章小结	361
11.2.4 执行菜单程序	315	习题 12	362
11.3 菜单设计技术	315	附录	363
11.3.1 菜单系统选择逻辑设计	316	附录 1 习题 1 答案	363
11.3.2 将系统菜单引入用户菜单系统	319	附录 2 习题 2 答案	364
11.3.3 顶层表单设计	321	附录 3 习题 3 答案	366
11.3.4 设计应用系统启动程序	322	附录 4 习题 4 答案	369
11.4 本章小结	324	附录 5 习题 5 答案	370
习题 11	324	附录 6 习题 6 答案	373
第 12 章 应用系统集成技术	326	附录 7 习题 7 答案	378
12.1 错误处理技术	326	附录 8 习题 8 答案	379
12.1.1 错误处理	326	附录 9 习题 9 答案	382
12.1.2 对象错误事件处理	328	附录 10 习题 10 答案	382
12.2 编译应用程序	329	附录 11 习题 11 答案	382
12.2.1 构造应用程序框架	329	附录 12 习题 12 答案	383
		参考文献	384

第 1 章 概 述

数据库技术是数据管理的技术,它应数据管理任务的需要而产生于 20 世纪 60 年代中期。这是一门综合性技术,涉及到操作系统、数据结构、算法设计和程序设计等知识。

1.1 数据库系统概述

本章将数据库系统中非常实用的知识通过提取、精炼汇集在一起予以介绍,以便读者能够掌握数据库系统的理论基础。

1.1.1 数据库系统基础知识

1. 数据管理的一些基本概念和发展过程

(1) 数据

数据是一种物理符号序列,用来记录事物情况,用型和值来表征。不同数据类型,记录的事物性质也不同。如数值型数据 1,2,3..., 可用来记录事物的多少。

(2) 信息

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。

信息都是数据,而只有经过提炼和抽象之后具有使用价值的信息才能成为信息。加工所得的信息仍以数据形式表现,此时的数据是信息的载体,是人们认识信息的一种媒体。

(3) 数据处理

数据处理是指对各种类型的数据进行收集、存储、分类、计算、加工、检索及传输的过程。其目的是得到信息。数据处理也称为信息处理或信息技术等。

数据处理的核心是数据管理。数据管理指的是对数据的分类、组织、编码、储存、检索和维护等。数据管理经历了人工管理、文件系统和数据库系统 3 个阶段。

① 人工管理阶段(20 世纪 50 年代中期以前) 其特征是:

- 数据需要由应用程序管理,没有相应的软件系统负责数据的管理工作;
- 数据不具有独立性;
- 数据不保存;
- 数据不能共享;
- 数据冗余度极大。

② 文件系统阶段(20 世纪 50 年代后期到 60 年代中期) 其特征是:

- 数据由专门的软件即文件系统负责管理(图 1.1);
- 数据具有一定的独立性;
- 数据可以长期保存;
- 数据共享性差;

- 数据冗余度大。

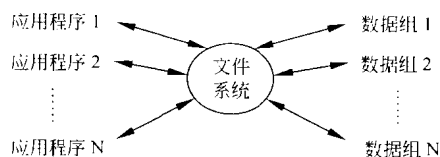


图 1.1 文件系统阶段应用程序与数据之间的关系

③ 数据库管理阶段(20 世纪 60 年代后期以来) 其特征是:

- 数据由数据库管理系统(database management system,DBMS)统一管理和控制, DBMS 还提供了数据的安全性、完整性、并发控制和数据库恢复等功能(参见图 1.2);
- 数据独立性高;
- 数据结构化(这是数据库与文件系统的根本区别);
- 数据充分共享;
- 数据冗余度低。

数据库主要解决了 3 个问题:一是有效地组织数据,对数据进行了合理设计,以便计算机存取;二是将数据方便地输入到计算机中;三是根据用户的要求将数据从计算机中抽取出来(这是人们处理数据的最终目的)。



图 1.2 应用程序与数据库的关系

2. 数据库系统的组成

数据库系统实际上是一个应用系统,它是在计算机硬、软件系统支持下,由用户、数据库管理系统、存储在存储设备上的数据和数据库应用程序构成的数据处理系统。

(1) 数据

这里是指数据库系统中存储在存储设备上的数据,是数据库系统操作的对象。这种数据具有集中性和共享性。

(2) 数据库管理系统

是指位于用户与操作系统之间、负责数据库存取、维护 and 管理的软件系统。这是数据库系统的核心,其功能强弱是衡量数据库系统性能优劣的主要方面。一般由计算机软件公司提供。

(3) 应用程序

是指为适合用户操作、满足用户需求而编写的数据库应用程序。

(4) 用户

是指使用数据库的人员,主要有 3 类:终端用户、应用程序员和数据库管理员。终端用户是指计算机知识不多的一般管理人员,只会通过数据库系统所提供的命令语言、表格

语言以及菜单等交互式对话手段来使用数据库的数据;应用程序员是指为终端用户编写应用程序的软件人员,设计的应用程序主要用途是使用和维护数据库;数据库管理员(database administrator,DBA)是指全面负责数据库系统正常运转的高级人员,负责对数据库系统本身的深入研究。

3. 数据库系统的特点

这是计算机数据处理技术的重大进步,涉及内容如下所示。

(1) 实现数据共享

允许多个用户同时存取数据而互不影响。

(2) 实现数据独立

应用程序不随数据存储结构的改变而变动。

(3) 减少了数据冗余度

逻辑数据文件和物理数据文件存在着“多对一”的重叠关系,有效地节省了存储资源。

(4) 避免了数据不一致性

数据只有一个物理备份,故对数据的访问不会出现不同。

(5) 加强了对数据的保护

数据库加入了安全保密机制,可防止对数据的非法存取。进行集中控制,有利于控制数据的完整性。采取了并发访问控制,保证了数据的正确性。另外,还实现了对数据库破坏后的恢复。

4. 数据库应用系统(database application systems,DBAS)

是指开发人员利用数据库系统资源开发出来的、面向某实际应用的软件系统,分为两类:

- 管理信息系统 这是面向机构内部业务和管理的数据库应用系统,如教学管理系统、财务管理系统等;

- 开放式信息服务系统 这是面向外部、提供动态信息查询功能,以满足不同信息需求的数据库应用系统。例如,大型综合科技情报系统、经济信息系统和专业的证券实时行情、商品信息系统等。

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。有关内容将在后续章节中介绍。

1.1.2 现实世界的描述

这里将介绍信息的3个领域,然后讨论概念模型和数据模型。

1. 数据描述

信息管理的对象是现实生活中的客观事物,描述具体事物的信息经过整理、归类 and 进行规范化后,才能数据化并送入计算机的数据库中保存。这一过程经历了3个领域——现实世界、信息世界和数据世界。

(1) 现实世界

是存在于人脑之外的客观世界、事物及其相互联系。事物可用“对象”与“性质”来描述,又有“特殊事物”与“共同事物”之分。

(2) 信息世界

是现实世界在人们头脑中的反映。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联系。

(3) 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联系在这里用数据模型描述。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联系的两种抽象描述。数据库的核心问题是数据模型,为了得到正确的模型,首先要充分了解客观事物。

2. 实体模型

指的是客观事物在人脑中反映的且经信息世界记载下来的文字和符号。描述事物的常用术语有下列 7 种。

(1) 实体(entity)

客观事物在信息世界中称为实体。它可以是具体的,如一个学生、一本书;也可以是抽象的,如一场足球比赛。实体用型(type)和值(value)来表征。例如学生是一个实体,学生姓名是实体的型的描述,而具体的学生李明是实体值。

(2) 属性(attribute)

实体的特性在信息世界中称为属性。每个属性都有一个值域(domain),其类型可以是整数、实数或字符型,例如:学生的姓名、年龄都是学生这个实体的属性,姓名的类型为字符型,年龄的类型为数值型。

属性用型(type)和值(value)表征,例如:学号、姓名、年龄是属性的型的描述,而具体的值 0107039、赵东、18 等则是属性值。

(3) 实体型

所谓实体型就是对实体的型的描述,通常使用实体名和属性名的集合。如学生实体,其型的描述为:学生(学号,姓名,年龄)就是一个实体型。

(4) 实体值

是实体实例、属性值的集合,如学生赵东的实体值是:0107039、赵东、18。

(5) 实体集(entity set)

性质相同的同类实体的集合称为实体集。如同班的所有学生。

(6) 实体联系

建立实体模型的目的是要找出实体间的联系,基本联系有 3 种(设 A 和 B 为两个实体)。

① 一对一联系(1:1) 若 A 中的任一属性至多对应 B 中的惟一属性,反之亦然,则称 A 与 B 是一对一联系。如观众与座位之间、乘车旅客与车票之间的联系。

② 一对多联系(1:N) 若 A 中至少有一个属性对应 B 中一个以上的属性,且 B 中的任一属性至少对应 A 中的一个属性,则称 A 对 B 是一对多联系,如班级对学生。

③ 多对多联系(M:N) 若 A 中至少有一个属性对应 B 中一个以上的属性,且 B 中也至少有一个属性对应 A 中一个以上的属性,则称 A 与 B 是多对多联系,如学生与课程之间的联系。

原则上,许多实体之间的复杂联系都可用若干组基本联系等价地表示。

(7) 概念模型(即实体模型)

反映实体之间联系的模型称为实体模型。数据库设计的重要任务就是建立实体模型,它是概念数据库的具体描述。在建立实体模型中,实体要逐一命名以示区别,并描述其间的各种联系。现以教学管理为例来建立实体模型。

教学管理可由教师、学生、课程、任课、成绩等方面的实体组成。其中,教师具有的属性为教师代号、姓名、性别、职称;学生具有的属性为学号、姓名、性别、专业;课程具有的属性为课号和课名;任课具有的属性为教师代号和课号;成绩具有的属性为学号、课号和成绩。

学生对课程是多对多联系,因为一个学生可学习多门课程,而一门课程又有多个学生学习。学生对成绩是一对多联系,即一个学生可学多门课程,而每门课程都有一个相应的成绩。同样地,课程对成绩、课程对任课、教师对任课也都是一对多的联系。相应的实体模型见图 1.3。

这是一种静态实体模型,只反映实体的当前状态,而不能反映实体状态的变化过程。

3. 数据模型的分类

数据模型是数据库系统的基石。从创建数据库技术以来,数据模型有 4 种类型,即层次模型、网状模型、关系模型和关系对象模型。

(1) 层次模型(hierarchical model)

使用树型结构来表示实体的类型和实体间的联系,树的结点为记录的类型,记录类型只有简单的层次关系。

(2) 网状模型(network model)

这是层次模型的拓展,广义上讲,任意一个连通的基本层次联系的集合就是一个网状模型。

(3) 关系模型(relation model)

其逻辑结构是一张二维表,由行和列组成。这是数据模型中最重要的模型。1970 年由美国的 E. F. Codd 提出。自 20 世纪 80 年代以来一直使用关系模型来描述现实世界。本书主要介绍关系数据模型的应用。

(4) 关系对象模型

20 世纪 90 年代中期以来,人们发现关系模型有着查询效率不如非关系模型等一些缺陷,所以提出了关系对象模型。关系对象模型一方面对数据结构方面的关系结构进行了改良,另一方面对数据操作引入了对象操作的概念和手段。今天的数据库管理系统基本上都提供了这方面的功能。

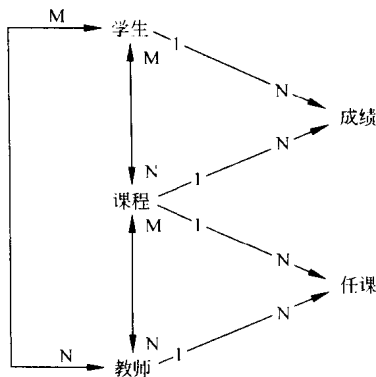


图 1.3 教学实体模型

1.1.3 关系模型

1. 关系模型中的几个术语

(1) 字段(field)

信息世界中的“属性”，就是数据世界中的“数据项”；从数据库的角度讲，数据项就是字段；从表格的角度讲，数据项称为列。例如学生的学号、姓名、性别和专业就是字段名。字段和属性一样，也用型和值表示。由此可见，字段、属性、数据项、列这些术语，所描述的对象是相同的，只是从不同角度对对象进行描述。

(2) 记录(record)

字段的有序集合称为记录。在关系模型中，记录称为元组；在表中，记录称为行；在概念模型中称为实体。换句话说，实体、记录、元组和行分别是不同角度上描述同一对象的术语。记录也由型和值来描述：记录型是字段型的集合，记录值是字段值的集合。

(3) 表(relation)

记录的集合称为表。记录的型和值构成了关系数据库的基本单位，即表。表也分为型和值，表的型也称关系模式，或称表结构，由一系列字段型组成。

(4) 关键字(key word)

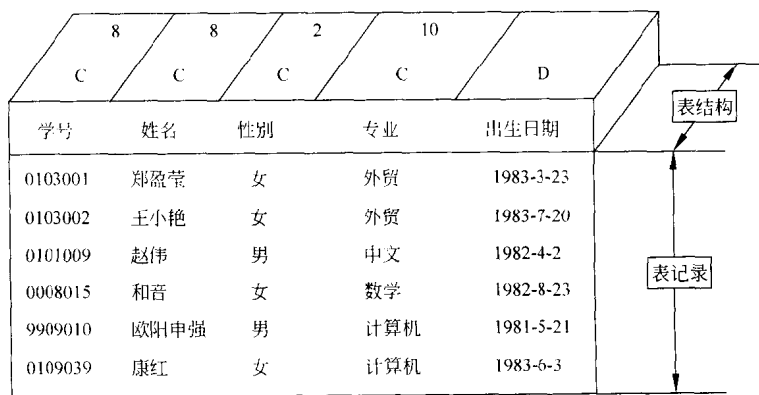
关键字是能够惟一确定记录的字段或字段的集合。有了关键字就可很方便地使用指定的记录。

(5) 关系数据库

这是由若干个表组成的集合。也就是说，关系数据库中至少有一个表。在实际应用中，关系数据库通常是由若干个有着一定关系的表组成的。在关系数据库系统中，关系模式是相对稳定的。因数据库中的数据需要不断更新，故关系数据是不断变化的。

2. 关系的组成与性质

一个关系实际上就是一个表，表是由不同的行和列组合而成的，图 1.4 是个学生关系。从图中可看出，学生关系是由表结构和表记录组成的一个表。在这个表中有 5 列(即



The diagram illustrates a table structure and its records. The table has five columns: 学号 (Student ID), 姓名 (Name), 性别 (Gender), 专业 (Major), and 出生日期 (Date of Birth). The columns are labeled with their data types and lengths: C 8, C 8, C 2, C 10, and D respectively. The table contains six rows of data. To the right of the table, a box labeled '表结构' (Table Structure) points to the column headers, and a box labeled '表记录' (Table Record) points to the data rows.

学号	姓名	性别	专业	出生日期
0103001	郑盈莹	女	外贸	1983-3-23
0103002	王小艳	女	外贸	1983-7-20
0101009	赵伟	男	中文	1982-4-2
0008015	和音	女	数学	1982-8-23
9909010	欧阳申强	男	计算机	1981-5-21
0109039	康红	女	计算机	1983-6-3

注：假设“学号”由年级(2位)+专业(2位)+序号(3位)组成。如“0103001”代表该学生是 01 级外贸专业的 001 号。

图 1.4 学生关系

5 个字段)、6 行(即 6 条记录)。表结构部分的表示是:字段名、字段类型和字段宽度,如学号字段,其名为“学号”、类型为 C(字符型)、宽度为 8。表记录部分表示了一条一条的记录值,即这个表是由 5 个字段和 6 个记录组成的。

从这个关系中可得出一些重要的结论:

- 表中每一列元素是类型相同的数据;
- 列不重名,列的顺序可任意放置;
- 行的顺序也可任意,表中任意两行不能完全相同(即没有重行);
- 表中元素是不可再分的最小数据项(描述对象属性的数据)。

这就是关系的基本性质,也是衡量一个关系的基本要素。在这些性质中,有一点是最关键的,即表中元素是不可再分的最小数据项。最小数据项是基本单元格,不能在表中再套有表。这样规定的目的是将复杂的问题简单化,但是带来的后果却是使得应用复杂化。例如在学生关系中增加一个家庭地址字段,而该字段由邮政编码、地址两个基本项组成;按照关系的性质,该字段就不是最小数据项,因而就不能组成一个关系;必须再细化之后,才能成为一个关系。然而,在实际应用中,使用家庭地址会更方便。也正是由于这一点,才引入了对象数据库的概念。换句话说,对象数据库中去掉了最小数据项的性质。

3. 建立关系结构模型

建立关系结构模型实际上是将概念模型数据化。由于实体之间存在着复杂的联系,因此所有描述实体的数据之间也存在着复杂的联系。为使模型能清晰、准确地反映客观事物,并能用于数据库设计,创建关系结构模型应采取如下步骤:

- ① 给关系结构模型命名,使不同模型得以区别;
- ② 给每个表命名;
- ③ 给每个字段命名,指出数据项特征,即字段的类型、长度、值域;
- ④ 找出各个表之间的联系。

教学管理中的关系结构模型如图 1.6 所示,其中的某一实体(参见图 1.5),表示一个关系表。图 1.5 中的“成绩”是该实体(即关系表)的名称,“学号(C,7)”中的“学号”是字段名,“C”表示字段数据类型为字符型,“7”是指该字段的长度为 7,下划线表示该字段为关键字;“N,5.1”中的“N”表示字段为数值型,其长度为 5,小数位数为 1。在图 1.6 中“L”表示字段数据类型为逻辑型;“M”表示备注型;“G”表示通用型等。更具体的含义请参见 2.3.1 节。

成绩
<u>学号(C,7)</u>
课号(C,6)
平时(N,5.1)
期中(N,5.1)

图 1.5 单一实体

4. 关系模型的优点

• 建立在严格的数学概念基础之上,每个关系都是用一张表格来描述的,字段、记录描述得很清楚。更重要的是可用关系的性质来衡量关系。

- 关系规范化,即每个数据项(即字段)都是一个不可分的数据单元。
- 概念简单,数据结构简单、清晰,用户易懂易用。
- 存取路径对用户透明,数据的独立性高,安全保密性强。