

高职高专计算机系列教材
Gaozhi Gaozhuan Jisuanji Xilie Jiaocai

Visual FoxPro 程序设计教程

柳 青 李拥军 谭小球 编

138

高等教育出版社

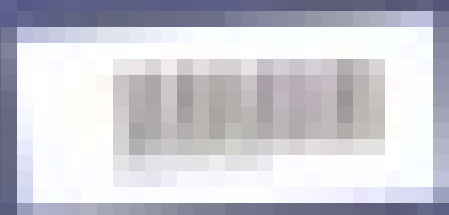
HIGHER EDUCATION PRESS



THE UNIVERSITY OF CHICAGO PRESS

THE UNIVERSITY OF CHICAGO PRESS

THE UNIVERSITY OF CHICAGO PRESS



THE UNIVERSITY OF CHICAGO PRESS

1194947

高职高专计算机系列教材

Visual FoxPro 程序设计教程

柳 青 李拥军 谭小球 编

高等教育出版社

内容提要

本书以 Visual FoxPro 6.0 为例,针对其将面向对象程序设计技术与传统的过程化程序设计模式相结合的特点,详细介绍了数据库管理系统的使用方法和应用程序的开发技能。全书共 11 章,主要包括:数据库原理概述、Visual FoxPro 基础、表的建立与操作、程序设计基础、面向对象程序设计基础、表单、数据库的建立与操作、查询与视图、菜单设计、报表和标签、应用程序设计实例等。

全书配有大量的实例,每章均有小结和适量的习题,具有较强的实用性,可作为高等职业技术学校 and 高等专科学校相关专业数据库课程的教材和教学参考书,也可供各类计算机培训班和个人自学使用。

图书在版编目(CIP)数据

Visual FoxPro 程序设计教程/柳青等编. —北京:高等教育出版社,2001

ISBN 7-04-008655-7

I. V... II. 柳... III. 关系数据库-数据库管理系统, Visual FoxPro 6.0-教材 N. TP311.138

中国版本图书馆 CIP 数据核字(2001)第 07807 号

Visual FoxPro 程序设计教程

柳 青 李拥军 谭小球 编

出版发行 高等教育出版社

社 址 北京市东城区沙滩后街 55 号

邮政编码 100009

电 话 010-64054588

传 真 010-64014048

网 址 <http://www.hep.edu.cn>

<http://www.hep.com.cn>

经 销 新华书店北京发行所

排 版 高等教育出版社照排中心

印 刷 北京印刷二厂

开 本 787×1092 1/16

版 次 2001 年 6 月第 1 版

印 张 21

印 次 2001 年 6 月第 1 次印刷

字 数 500 000

定 价 17.90 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

前 言

Visual FoxPro 6.0 提供了为数据库结构和应用程序开发而设计的面向对象环境,具有功能强大、开发工具丰富、数据处理速度快、用户界面友好等特点,易于学习和使用,受到广大用户的欢迎。

Visual FoxPro 6.0 及其中文版是可运行于 Windows 95/98 和 Windows NT、Windows 2000 平台的 32 位数据库开发系统,该系统可以简化数据库的管理,使应用程序的开发流程更为合理。Visual FoxPro 6.0 利用可视化的设计工具和向导,使组织数据、定义数据库和建立应用程序等工作变得简单易行,使用户可以方便地创建数据表、数据库、表单、查询、视图、报表、标签和程序。Visual FoxPro 6.0 为用户提供了集成化的系统开发环境,不仅支持面向过程的程序设计技术,而且支持面向对象可视化程序设计技术,并拥有功能强大的可视化程序设计工具和客户/服务器应用程序的开发能力,而且支持 OLE 和 ActiveX,为用户开发功能全面的应用程序创造了良好的条件。

本书以 Visual FoxPro 6.0 为例,较详细地介绍了数据库管理系统的使用方法和应用程序的开发技能。

为贯彻以能力为本位的思想,本书注重应用能力的培养。在教材的编写中,既注意内容的系统性,也注重实用性;既有通俗易懂的原理介绍,又有具体的操作指导。全书配有大量的实例,且每章后面附有小结和适量的习题,突出了较强的实用性,可作为高等职业技术学校 and 高等专科学校相关专业数据库课程的教材或教学参考书,也可供各类计算机培训班和个人自学使用。

本书由柳青副教授主编。其中第一、二、三、七、九、十章由柳青编写,第四、五、六章由李拥军编写,第八、十一章由谭小球编写。全书由柳青负责统稿和定稿。

本书的编写得到广州航海高等专科学校的大力支持和帮助,在此表示衷心的感谢。

限于作者的水平,书中难免有不当之处,敬请指正。

作者

2000.12

目 录

第一章 数据库原理概述	1	习题	51
1.1 数据库基本概念	1	第三章 表的建立与操作	53
1.1.1 数据处理的概念	1	3.1 表结构的建立与操作	53
1.1.2 数据库技术的发展	2	3.1.1 表结构的建立	53
1.1.3 数据库系统	3	3.1.2 表结构的修改与显示	61
1.1.4 数据模型	4	3.2 表的编辑和显示	63
1.2 关系数据库	6	3.2.1 表文件的打开与关闭	63
1.2.1 关系数据库的结构	6	3.2.2 向表中输入记录数据	64
1.2.2 关系运算	6	3.2.3 数据的显示和编辑	69
1.2.3 SQL 语言简介	11	3.2.4 记录指针与定位	74
1.2.4 关系数据库的规范化	11	3.2.5 记录的删除和恢复	77
习题	12	3.2.6 控制字段和记录的访问	80
第二章 Visual FoxPro 基础	13	3.3 表的索引	82
2.1 Visual FoxPro 6.0 中文版简介	13	3.3.1 索引概述	82
2.1.1 Visual FoxPro 6.0 中文版的特点与 功能	13	3.3.2 索引的建立和使用	84
2.1.2 Visual FoxPro 6.0 中文版的安装、 配置与优化	16	3.3.3 改变记录的物理顺序	87
2.1.3 Visual FoxPro 6.0 中文版的用户 界面	20	3.4 表内容的查找	87
2.1.4 Visual FoxPro 的设计工具	28	3.5 使用多个数据表	90
2.2 Visual FoxPro 语言基础	30	3.5.1 使用数据工作期	90
2.2.1 数据类型	30	3.5.2 在工作区中打开和关闭工作表	90
2.2.2 数据存储方式	32	3.5.3 引用工作区和使用表别名	91
2.2.3 表达式	36	本章小结	92
2.2.4 函数	39	习题	93
2.2.5 Null 值	39	第四章 程序设计基础	95
2.2.6 数据库和表的组成	39	4.1 结构化程序设计概述	95
2.2.7 命令格式	41	4.2 程序文件的建立与执行	96
2.3 项目管理器	42	4.2.1 程序文件的建立	97
2.3.1 项目管理器的功能和界面	42	4.2.2 程序文件的执行	98
2.3.2 项目管理器的操作	45	4.3 程序设计的基本语句	99
2.3.3 项目管理器能访问的工具	48	4.3.1 程序调试语句	99
本章小结	51	4.3.2 简单的输入语句	99
		4.3.3 简单的输出语句	101
		4.4 程序控制的基本结构	101
		4.4.1 顺序结构	101

4.4.2 选择分支结构	102	6.4.1 用表单模板创建新表单	188
4.4.3 循环结构	107	6.4.2 用表单集扩充表单	189
4.4.4 多种结构的嵌套	113	6.4.3 表单的显示类型	189
4.5 过程与过程文件	115	6.4.4 管理表单	191
4.5.1 子程序和过程的概念	115	6.5 程序设计举例	192
4.5.2 参数传递与作用域	119	本章小结	196
4.5.3 自定义函数	122	习题	197
4.5.4 子程序的嵌套与递归调用	123	第七章 数据库的建立与操作	198
4.6 应用程序设计的基本方法	125	7.1 Visual FoxPro 数据库简介	198
本章小结	129	7.2 创建数据库	199
习题	129	7.2.1 建立数据库	199
第五章 面向对象程序设计基础	132	7.2.2 打开数据库	201
5.1 面向对象程序设计基本概念	132	7.2.3 在数据库中添加和移去数据表	201
5.1.1 面向对象技术概述	132	7.3 建立表间的关联关系	202
5.1.2 面向对象技术的特点	132	7.3.1 建立关联关系前的准备	202
5.2 Visual FoxPro 中的对象与类	133	7.3.2 建立数据表间的关联关系	203
5.2.1 Visual FoxPro 中的类	133	7.4 设置数据库表与数据库的特性	206
5.2.2 Visual FoxPro 中的对象	135	7.4.1 设置字段标题	206
5.3 创建类	136	7.4.2 设置字段的注释	206
5.3.1 类与任务的匹配	136	7.4.3 设置字段的默认值	207
5.3.2 创建新类	138	7.4.4 设置有效性规则和有效性说明	208
5.3.3 将属性和方法添加到类	141	7.4.5 建立数据库的参照完整性	209
5.3.4 使用类库文件	143	本章小结	210
5.4 面向对象程序设计	145	习题	210
5.4.1 将类添加到表单中	145	第八章 查询和视图	212
5.4.2 在容器类中添加对象	146	8.1 Visual FoxPro 中的查询与视图	212
5.4.3 设置属性	148	8.2 利用向导建立查询	212
5.4.4 调用方法	149	8.3 利用查询设计器建立查询	217
本章小结	150	8.3.1 启动查询设计器	218
习题	151	8.3.2 定义查询结果	218
第六章 表单	152	8.3.3 定向输出查询结果	221
6.1 设计和创建表单	152	8.3.4 运行查询	222
6.1.1 用表单“向导选取”设计表单	152	8.3.5 分组查询	223
6.1.2 用表单设计器设计表单	159	8.4 用表达式筛选查询结果	223
6.1.3 用“快速表单”设计表单	160	8.5 视图的创建与运行	227
6.1.4 表单的修改和运行	161	8.5.1 创建视图	227
6.2 设置表单的数据环境	162	8.5.2 保存视图	231
6.3 表单控件	165	8.5.3 修改视图	231
6.3.1 表单控件概述	165	8.5.4 运行视图	231
6.3.2 表单控件	166	8.6 用视图更新数据	232
6.4 表单的设计	187	8.6.1 设置关键字段和可更新字段	232

8.6.2 用视图更新数据与检查更新冲突	234	10.3.1 用“标签向导”创建标签	254
本章小结	234	10.3.2 用“标签设计器”创建标签	256
习题	235	本章小结	258
第九章 菜单设计	236	习题	258
9.1 规划菜单系统	236	第十一章 应用程序设计实例	259
9.2 创建菜单和子菜单	237	11.1 设计目标	259
9.2.1 启动菜单设计器	237	11.2 创建项目	260
9.2.2 用菜单设计器创建菜单系统	238	11.3 创建数据库	260
9.2.3 创建菜单项或子菜单	239	11.4 建立应用程序的过程	262
9.2.4 为菜单指定热键和快捷键	240	11.4.1 创建“启动”表单	262
9.2.5 为菜单系统指定任务	241	11.4.2 创建“功能选择”表单	263
9.2.6 生成菜单程序	243	11.4.3 创建“职工信息”表单	264
9.3 创建快捷菜单	243	11.4.4 创建“工资信息”表单	275
本章小结	244	11.5 项目连编	280
习题	244	11.5.1 设置主程序	280
第十章 报表和标签	245	11.5.2 项目连编	281
10.1 报表的建立	245	本章小结	281
10.1.1 用“报表向导”创建报表	245	习题	282
10.1.2 用“报表设计器”建立快速报表	248	附录	283
10.2 报表的设计	251	附录 A Visual FoxPro 6.0 函数	283
10.2.1 确定报表的布局	252	附录 B Visual FoxPro 6.0 命令	297
10.2.2 报表的设计	252	附录 C Visual FoxPro 6.0 类与对象	310
10.3 标签	254	附录 D Visual FoxPro 6.0 属性、事件和方法	311

第一章 数据库原理概述

随着计算机技术的发展,计算机的应用逐渐由数值计算向非数值计算的各个领域扩展,已经进入社会的各个领域乃至人们的家庭生活中。计算机技术的高速发展推动人类社会进入了信息时代。

数据库技术是数据信息管理技术的最新成果,被广泛地应用于国民经济、文化教育、企业管理以及办公自动化等方面,为计算机的应用开辟了广阔的天地。从某种意义上来说,数据库管理系统是计算机技术和信息时代相结合的产物,是信息处理和数据处理的核心,是研究数据共享的科学,也是计算机科学的一个重要分支。

本章主要介绍数据库的基础知识和基本概念,为后面学习和掌握 Visual FoxPro 程序设计技术打下基础。

1.1 数据库基本概念

1.1.1 数据处理的概念

数据(Data)和信息(Information)是数据处理中的两个基本概念。数据是人们为了进行交流、通信、解释,用来表示主客观对象的一种形式。或者说,数据是用来表示数量、活动、事务等情况的一组符号,这些符号可以是文字、符号、数字、表格、图形、声音等。在数据库中,把计算机能识别并能处理的一切符号,都称为数据。这里的符号包括数字、文字(英文字母、汉字等)、各种符号和图形等。通常可将数据分为许多类,如表示工资、物价、考试成绩等具有量的多少的数据,称为数值型数据;表示姓名、商品名称、课程名称等的的数据,称为字符型数据。

信息是客观事物属性的反映,是经过加工并对人类社会实践和生产经营活动产生影响的数据表现形式。或者说,信息是对原始数据加工后得到的,对于某个目的来说有用的知识或数据。数据是信息的原始资料,数据经过解释并赋予一定的意义后,便成为信息。

信息和数据在概念上既有联系,又有区别。它们之间的关系可以看成是原料和成品之间的关系,不经过加工的数据只是一种原始材料,不能影响人们的社会实践和生产经营活动,这种数据只能记录客观世界的事实。只有经过提炼和加工,使数据发生质的变化,才能成为信息。因此,信息来源于数据,是对数据进行加工处理的产物。数据经过加工后,被赋予一定的含义,使其具有知识性并对人类活动产生决策作用,从而形成信息。经过加工后得到的信息,仍然以数据的形式出现,此时的数据是信息的载体,是供人们认识和利用信息的一种媒介。例如,某百货公司

各月份的商品销售量是该公司商品销售情况的反映。商品销售量可以用数字表示为一组数据,管理人员难以从该组数据中直接得到该公司的销售情况分析。但是,把商品销售数据按商品进行分组,并统计出各种商品的销售量后,就可以知道该公司各种商品的销售情况了。这组反映商品销售情况的数据就是信息,它们是在原始销售数据的基础上经加工后得到的。

数据处理又称为信息处理,即利用计算机对各种类型的数据进行处理,包括对数据的采集、存储、整理、分类、排序、统计、加工、检索、维护和传输等一系列操作过程。数据处理的目的是从大量的、原始的数据中获得我们所需要的资料并提取有用数据成分,作为行为和决策的依据。随着人类社会的发展和进步,在人类社会活动(包括政治、经济和文化活动)中,信息居于非常重要的地位。可以说,人类的全部活动都是在信息的支配和指导下进行的。随着信息量的急剧增长,人们对数据处理提出了更高的要求。电子计算机已经成为数据处理的强有力工具,尤其是数据库技术的发展,将数据处理提高到一个更高的水平。

1.1.2 数据库技术的发展

随着计算机硬件与软件的发展,利用计算机进行数据处理经历了人工管理、文件系统和数据库系统三个阶段。

人工管理阶段是数据处理的初期阶段,采用程序管理的方式。这个阶段由于计算机技术还很落后,数据量不是很大,数据处理是一个程序对应一组数据,数据和程序在一起。因此,数据不能长期保存,没有软件系统对数据进行管理,基本上没有文件的概念,数据本身不具有独立性而且有大量的数据冗余。

文件系统管理阶段把数据组织成文件的形式,所有的文件由一个称为文件管理系统的软件进行管理。文件中的数据以“记录”的形式存放,记录由某些相关的数据项组成,若干个具有相同性质的记录的集合构成了文件。每一个用户都可以建立、维护和处理一个或几个文件,每个文件都有特定的文件名存在外部存储器上。数据可以离开处理它的程序而独立存在。应用程序通过文件系统对文件中的数据进行操作。文件系统作为应用程序和数据之间的桥梁,使得一批数据在一个专门的文件系统管理之下,可以被多个应用程序使用。因此,文件系统可以使数据具有独立性,在一定程度上减少了数据的冗余。但是,文件系统的方法还存在着冗余度大、空间浪费、文件不易扩充等缺点,数据还缺乏充分的独立性,数据间的联系及数据共享还不理想,对数据的统一管理和控制还比较困难。

数据库系统阶段在文件系统的基础上形成了数据库技术。数据库是把数据按一定的结构和形式组织起来的记录的集合,这些数据存储在计算机外存储器上,具有以下特点:尽可能不重复(即最少冗余);以最优方式服务于一个或多个应用程序(应用程序对数据资源的共享);数据的存放应尽可能地独立于使用它的应用程序(数据的独立性);用一个软件统一管理这些数据,如维护、增加、变更和检查数据等。数据库的基本思想是要对所有的数据实行统一、集中和独立的管理,使数据独立于程序而存在并可提供给各类不同的用户共享使用。

由此可以看出:

(1) 数据库中的数据必须是关联的。实际上这种关联就是把数据按一定的结构存放,因此,数据库不但存放数据,而且存入数据之间的关系。

(2) 数据库中的数据都与应用程序无关地存放在外存储器上。

目前,数据库系统已经得到了广泛的应用,大型复杂的信息管理系统大多数以数据库为核心,数据库系统已经成为现代管理信息系统强有力的工具,在计算机应用中起着越来越重要的作用。随着计算机应用从单机发展到网络,数据库技术也向着网络应用方向发展。

1.1.3 数据库系统

数据库是指以一定方式存储在计算机存储设备上的、相互关联的数据的集合,供数据处理时使用。

数据库系统(数据库应用系统)由相关的硬件环境、软件系统、数据和人员四个部分组成。其中:

1. 硬件环境

硬件环境是数据库系统的物理支撑。数据库系统需要在操作系统的支持下工作,而且本身包含着数据库管理系统和应用程序等,因而需要有较大的内存容量。同时,由于用户的数据、系统软件和应用软件都要保存在外部存储器上,也需要较大的外存容量。

2. 软件系统

软件系统包括系统软件和应用软件。系统软件主要包括支持数据库管理系统运行的操作系统、数据库管理系统、开发应用系统的程序设计语言及其编译系统、应用系统开发工具等。系统软件为开发应用系统提供了良好的环境。应用软件是指在数据库管理系统的基础上,由用户根据自己的实际需要自行开发的应用程序。

数据库管理系统(Data Base Management System,简称 DBMS)负责对数据库进行统一管理和控制,是用户应用程序和数据库中数据之间的接口。数据库管理系统的职能是管理和维护数据库,接受用户提出的访问,处理数据库的各种请求。数据库只有在数据库管理系统的管理下,才能实现对数据的各种操作。数据库管理系统提供一系列与高级语言类似的命令,用户可以使用这些命令来实现对数据库的各种操作,也可以直接用这些命令编写应用程序。

数据库管理系统一般具有以下特征:

- (1) 支持整个数据的共享存取。
- (2) 可以灵活定义特定应用领域的数据结构。
- (3) 有为处理和分析数据提供广泛通用工具的能力。

在数据库系统中,用户对数据库进行的各种操作都通过数据库管理系统实现,因而使数据库中的数据具有较大的独立性。

3. 数据

数据是数据库系统的管理对象,是为用户提供数据的信息源。数据库是把数据按一定的结构和形式组织起来的记录的集合。

4. 人员

数据库系统的人员是指管理、开发和使用数据库的主要人员,主要包括数据库管理员、系统分析员、应用程序员和用户。数据库管理员负责管理和控制数据库系统;系统分析员负责应用系统的需求分析和规范说明,确定系统的软硬件配置、系统的功能及数据库概念设计;应用程序员

负责设计应用系统的程序模块,编写应用程序;最终用户通过应用系统提供的用户接口界面使用数据库。常用的接口方式有菜单驱动、图形显示、表格操作等,这些接口为用户提供了简明直观的数据表示方式和方便快捷的操作方法。

1.1.4 数据模型

数据库系统研究的对象是客观事物以及反映这些客观事物间相互联系的数据。数据库把数据按一定结构和形式组织起来,各个数据对象以及它们之间存在的相互关系的集合,称为数据模型。

数据模型反映了客观世界中各种事物的联系,是用来反映数据与数据之间联系的静态结构与行为的一种手段。由于数据库中的数据都是结构化的数据,因此,讨论数据模型对数据库设计来说很重要的。

根据数据所描述实体对象的属性特征以及人们所采用的描述方法,数据模型可分为三种:层次模型(Hierarchical Model)、网状模型(Network Model)和关系模型(Relational Model)。相应地,数据库也可分为层次型数据库、网络型数据库和关系型数据库。

1. 层次模型

在层次模型中,每个数据元素可以与下面任何一层的多个数据元素相联系,但只能与它上面一层中的一个数据元素相联系。层次模型最简单的结构及约束描述,可以用数据结构中的树来描述。层次模型像一棵由根元素出发向下逐层辐射枝叶的倒置的树,最高一层的数据元素称为根元素,根元素是惟一的,而且只有下属元素。树的结点表示数据元素,枝表示数据元素之间的联系。在层次模型中,有且仅有一个结点无上层结点(称为根);其他结点有且仅有一个上层结点。如一个企业的组织机构就是层次模型(见图 1-1)。

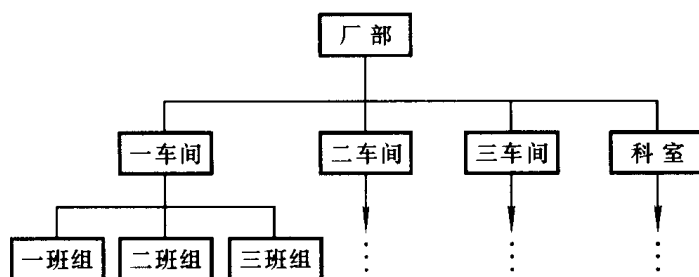


图 1-1 层次模型示意图

层次模型的特点是层次分明,结构清晰,适于表达事物之间“一对多”型的联系,如一个单位的行政组织关系、生产组织关系等。但是,由于层次模型结构的严格性,它不能直接地表示所有的事实,并且对数据存储和操作都不是理想的。

2. 网状模型

在网状模型中,可以有一个以上的数据元素没有上层联系;至少有一个以上结点有多于一个的上层结点。因此,网状模型有更一般的表示事实的能力,可以用来描述事物之间的较复杂的联系。图 1-2 是网状模型的一个例子,表示 4 种零件可由 3 个不同的厂家供应以及它们之间的联

系。在学校模型中,校部、系、教师和学生之间也具有网状联系,如图 1-3 所示。

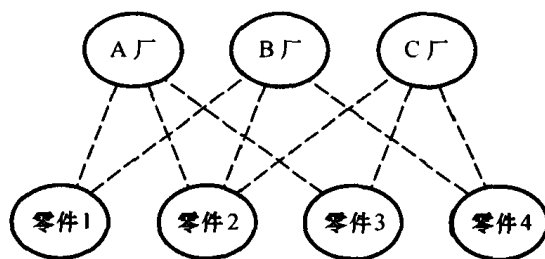


图 1-2 网状模型示意图

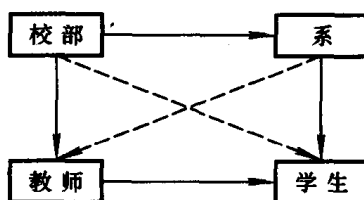


图 1-3 学校模型中的网状结构

网状模型适于表达事物之间的“多对多”型的联系。

3. 关系模型

在关系模型中,不同数据之间的联系用关系来表示,其实质是将数据的逻辑结构归结为满足一定条件的二维表形式,每个二维表称为一个关系。在二维表中,每一行称为一个记录(元组),每一列称为一个数据项或字段(属性),数据项名称为字段名或属性名,整个表表示一个关系。这种模型既可用于表达事物之间“一对一”和“一对多”的联系,也可以用来表达“多对多”的联系。

以表 1-1 所示的学生成绩关系作为例子来分析关系数据模型。

表 1-1 某班级学生成绩表

学号	姓名	英语	高等数学	计算机基础	总分	平均分
10101	王涛	90.00	80.00	95.00	265.00	88.33
10102	李冰	80.00	56.00	75.00	211.00	70.33
10103	谢红	55.00	75.00	52.00	182.00	60.67
10104	郑洁	62.00	67.00	88.00	217.00	72.33
10105	袁明	50.00	70.00	60.00	180.00	60.00
10106	张莉	52.00	49.00	58.00	159.00	53.00
10107	罗娟	86.00	78.00	92.00	256.00	85.33

关系模型的主要特点有:

(1) 关系中每一列具有相同的属性,每个属性(列)被指定一个不同的属性名(字段名),且属性名不能重复,列数根据需要设置;

(2) 关系中的每一个数据项必须是简单的数据项,而不是组合数据项;

(3) 关系中每一行记录由一个个体事物的诸多属性构成;

(4) 行和列的排列顺序是任意的;

(5) 一个关系是一张二维表,不允许有相同的字段名,也不允许有相同的记录。

利用关系模型构筑的关系型数据库,可以提供数据库系统的全部功能。

关系型数据库除了功能强大以外,还具有简单灵活,操作方便,易学易懂,数据独立性强等优点,所以近年来得到了迅速的发展和推广普及。

1.2 关系数据库

1.2.1 关系数据库的结构

关系数据库(Relation database)是若干个按照关系模型设计的数据表文件的集合,也可以说,关系数据库是由若干张按关系模型设计的二维表组成的。

一般将一张二维表称为一个数据表,在数据表中包含了数据和数据之间的关系。一个关系数据库由若干个数据表组成,数据表由若干个记录构成,而每一条记录又由若干个字段(数据项)组成。在关系数据库中,每一个数据表都具有相对独立性,该独立性的惟一标志是数据表名,又称为表文件名。一个表文件名代表一个独立的表文件。在关系数据库中,对数据表中数据的访问首先通过表文件名实现,因而不允许有数据表同名。关系数据库中各个数据表的独立性,可以方便对数据表中的数据进行操作、存取和传输。

对于关系数据库中具有相关性的数据表,其相关性依靠各个独立数据表内具有相同属性的字段而建立。一般情况下,一个关系数据库中可以有多个相关的独立数据表,可为数据资源的共享和充分利用提供极大的方便。由于关系数据库中各数据表之间具有独立性,而若干个数据表之间又具有相关性,因而使其具有极大的优越性,得到迅速的普及。

1.2.2 关系运算

关系数据库采用关系模型作为数据的组织方式,因此,关系数据库的基本数据结构是满足一定条件的二维表。

1. 关系模型中的几个基本概念

(1) 关系(Relation):一个二维表称为一个关系。

(2) 属性(Attribute):二维表中的一列称为一个属性,又称为字段(Field)。

(3) 域(Domain):一个属性的取值范围称为一个域。

(4) 元组(Tuple):二维表中的一行称为一个元组,又称为记录(Record)。

(5) 键(Key):二维表中的某个属性。例如,假定在一个班级中的学号无重复,则学号就是学籍关系的键。一个关系的所有键称为“候选键”;若一个关系有多个“候选键”,可选定其中一个为“主键”,这个属性称为主属性。

(6) 分量:元组中的一个属性值称为元组的一个分量。

(7) 关系模式:对关系的描述,包括关系名、组成该关系的属性名、属性到域的映像。通常简记为:

关系名(属性名₁,属性名₂,...,属性名_n)

属性到域的映像通常直接说明为属性的类型、长度等。一个关系模式在某一时刻的内容(状态)是元组的集合,称为关系。

如前所述,关系数据库是采用关系模式作为数据组织方式的数据库。对关系数据库的描述,称为关系数据库的型,它包括若干域的定义以及在这些域上定义的若干关系模式。这些关系模式在某一时刻对应的关系的集合,称为关系数据库的值。

例如,表1-1中的关系是一个学生成绩表。表中每一行是关系的一个元组,“学号”、“姓名”、“英语”、“高等数学”、“计算机基础”等是属性。其中“学号”是识别一条记录的主属性,称为“主键”。表1-1中,“学号”属性的域是“10101~10107”;“姓名”属性的域是2~4个汉字组成的字符串;“性别”属性的域是“男”或“女”等。因此,学生成绩表的关系模式可表示为:

学生成绩表(学号,姓名,英语,高等数学,计算机基础,总分,平均分)

2. 关系运算

关系之间可以进行运算,关系运算可以用两种方式表示:关系代数和关系演算。已经证明,这两种方式是等价的。这里仅以关系代数来表示关系运算。

关系代数是把关系作为“集合”来进行运算的。常用的关系运算有两类:传统的集合运算和专门的关系运算。

(1) 传统的集合运算如集合的并、交、差、广义笛卡尔积等。这类运算将关系看作元组的集合,其运算是从关系的“水平”方向,即二维表的行的角度来进行的。

① 集合的并运算

两个关系作集合的并运算,是将这两个关系合并,并消去冗余元组。例如,表1-2和表1-3是两个特殊的关系R和S(学生成绩处理系统中有关的部分关系),将这两个关系作集合的并运算时,要求这两个关系的所有属性相同。并运算的结果是将这两个关系合并,且消去冗余元组,运算结果如表1-4所示。

表1-2 关系 R

学号	姓名	平均分
10101	王涛	88.33
10102	李冰	70.33
10103	谢红	60.67
10104	郑洁	72.33

表 1-3 关 系 S

学号	姓名	平均分
10101	王涛	88.33
10102	李冰	70.33
10105	袁明	60.00
10106	张莉	53.00

表 1-4 R 与 S 并运算

学号	姓名	平均分
10101	王涛	88.33
10102	李冰	70.33
10103	谢红	60.67
10104	郑洁	72.33
10105	袁明	60.00
10106	张莉	53.00

② 集合的交运算

两个关系作集合的交运算时,同样要求关系中的所有属性相同。两个关系的交运算,是取两个关系中的公共元组。例如,表 1-2 和表 1-3 所表示的关系 R 和 S 作交运算的结果如表 1-5 所示。

表 1-5 R 与 S 交运算

学号	姓名	平均分
10101	王涛	88.33
10102	李冰	70.33

③ 集合的差运算

两个关系作集合的差运算,要求关系中的所有属性相同。两个关系 R 和 S 差运算的结果是取属于关系 R 而不属于关系 S 的元组。例如,表 1-2 和表 1-3 所表示的关系 R 和 S 作差运算的结果如表 1-6 所示。

表 1-6 R 与 S 差运算

学号	姓名	平均分
10103	谢红	60.67
10104	郑洁	72.33

(2) 专门的关系运算如选择、投影、连接等。这类运算不仅涉及关系的水平方向(二维表的

行),而且涉及关系的竖直方向(二维表的列)。

① 选择运算(Selection)

选择运算是在关系中选择满足某些条件的元组,即选择满足条件的行。例如,在学生成绩表中选取所有女学生的元组,可用选择运算实现,选择的条件是性别等于“女”。选择运算常用于对指定关系按条件进行检索。

② 投影运算(Projection)

投影运算是在关系中选择某些属性的列,其余属性列都不在运算结果中。例如,在表 1-1 所示学生成绩表中找出平均分在 70 分至 85 分之间所有学生的学号、姓名和平均分,则可以用投影运算实现,运算结果如表 1-7 所示。投影运算是用来检索指定关系中用户所关心属性列的值,运算结果是一个新的关系。

表 1-7 投影运算

学号	姓名	平均分
10101	王涛	88.33
10102	李冰	70.33
10104	郑洁	72.33
10107	罗娟	85.33

③ 连接运算(Join)

连接运算是将两个关系的笛卡尔积中选取属性间满足一定条件的元组。例如,表 1-8(关系 X)和表 1-9(关系 Y)的广义笛卡尔积为表 1-10 中的关系 Z。

对关系 X 和关系 Y 作条件为“X.专业号 = Y.专业号”的连接操作,所得关系 U 是关系 Z 的一个子集,如表 1-11 所示。

表 1-8 关系 X

学号	姓名	性别	专业号
10101	王涛	男	1
10102	李冰	女	1
10105	袁明	男	2

表 1-9 关系 Y

专业号	专业名
1	电子
2	财会
3	机电