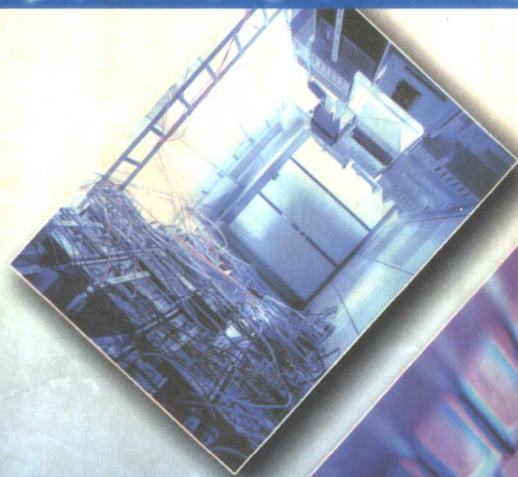


Visual FoxPro

数据库应用与程序设计

崔巍 吴秋霞



内容提要

本书以 Visual FoxPro 为背景介绍数据库系统的概念、使用、管理和开发,内容包括数据库基础知识、Visual FoxPro 数据库的建立与使用、关系数据库标准语言 SQL、查询和视图、界面设计、报表设计、建立应用程序等。涉及到数据库的知识也力求更加准确,为日后学生学习和使用其他(特别是大型)数据库打下良好基础。同时本书覆盖了全国计算机等级考试(二级 Visual FoxPro)大纲,从取材、习题等各方面既适应于普通高等学校的教学,也适应于计算机等级考试。

本书可作为高等院校各专业相关课程的教材,还可作为全国计算机等级考试(二级)等方面的培训教材,也可供广大数据库应用系统开发人员阅读参考。

图书在版编目(CIP)数据

Visual FoxPro 数据库应用与程序设计/崔巍,吴秋霞. —北京:高等教育出版社,2004.7

ISBN 7-04-015472-2

I. V… II. ①崔…②吴… III. 关系数据库—数据库管理系统, Visual FoxPro—高等学校—教材
IV. TP311.138

中国版本图书馆 CIP 数据核字(2004)第 058229 号

策划编辑 刘 英 责任编辑 刘 英 市场策划 陈 振
封面设计 杨立新 责任印制 宋克学

出版发行	高等教育出版社	购书热线	010-64054588
社 址	北京市西城区德外大街 4 号	免费咨询	800-810-0598
邮政编码	100011	网 址	http://www.hep.edu.cn
总 机	010-82028899		http://www.hep.com.cn

经 销 新华书店北京发行所
印 刷 北京印刷集团有限责任公司印刷二厂

开 本	787×1092 1/16	版 次	2004 年 7 月第 1 版
印 张	19	印 次	2004 年 7 月第 1 次印刷
字 数	460 000	定 价	23.10 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 是 Microsoft 公司推出的关系数据库管理系统及面向对象和可视化的数据库应用系统开发工具,与早期的 FoxBASE 或 FoxPro 相比,它引入了面向对象程序设计的思想,可以进行面向对象和可视化的程序设计,无论是从数据库的概念、使用和管理,还是从数据库应用系统的开发、速度、能力和灵活性等方面,都是早期的软件无法比拟的。

Visual FoxPro 具有功能强大、直观易用、工具完善而又丰富、友好的用户界面和完备的兼容性等特点。它提供的向导(Wizards)、设计器(Designer)、生成器(Builder)工具,使得对数据库的设计和管理变得十分容易。

本书以 Visual FoxPro 为背景介绍数据库系统的概念、使用、管理和开发。内容包括数据库基础知识、Visual FoxPro 数据库的建立与使用、关系数据库标准语言 SQL、查询和视图、程序设计基础、界面设计、报表设计及建立应用程序等。

现在几乎所有专业(包括文科)都开设数据库的课程,也都学习程序设计,而以 Visual FoxPro 作为学习环境是最佳的选择。本书还针对很多院校组织学生参加全国计算机等级考试(二级),所以在编写时特别兼顾、覆盖计算机等级考试(二级 Visual FoxPro)大纲,从取材、习题等各方面适应于一般的教学,也适应于计算机等级考试。同时涉及到数据库的知识也力求更加准确,为学生日后学习和使用其他(特别是大型)数据库打下良好基础。其中标注星号“*”的章节为超出计算机等级考试(二级 Visual FoxPro)大纲的内容。

每章配有大量的习题,题型一般分为思考题、选择题、填空题和上机操作题,其中选择题、填空题和上机操作题适合全国计算机等级考试(二级)的题型,也是对本章知识点的总结。所以,不管是否参加全国的计算机等级考试,都应该认真做好这些题。

对于广大非计算机专业的学生来说,Visual FoxPro 相对来说更容易学习。Visual FoxPro 可以说是一个入门级的数据库应用平台和开发工具,学生需要理解相关的数据库概念、掌握关系数据库标准语言 SQL 的使用和一般的程序设计基础、能够在 Visual FoxPro 中完成建立项目和数据库等一些基本操作、会建立和使用视图及查询、基本掌握表单、报表及菜单的应用等。

Visual FoxPro 的功能非常丰富,无法在一本 40 多万字的教科书中面面俱到,因此,在编写本书时尽量将基本的概念、核心的技术、必需的技能包含进来。而对 Visual FoxPro 全面地掌握和提高,还需要读者通过实践不断提高。笔者希望通过学习此书,使读者能够真正理解数据库,了解数据库设计的基本步骤,掌握数据库应用系统的基本开发方法,具有利用 Visual FoxPro 建立、开发数据库应用系统的基本能力,并为日后在工作、学习、生活中使用计算机进行数据管理打下一个良好的基础。

由于时间仓促和水平有限,书中疏漏之处在所难免,欢迎广大读者批评指正。

作 者

2004 年 4 月

目 录

第 1 章 Visual FoxPro 和数据库基础	(1)	2.3.5 查询定位记录	(52)
1.1 数据库基础	(1)	2.3.6 显示数据	(53)
1.1.1 什么是数据库	(1)	2.3.7 修改数据	(53)
1.1.2 数据库管理系统和数据 库系统	(3)	2.3.8 删除数据	(54)
1.1.3 数据管理的发展历程	(3)	2.3.9 使用浏览器操作数据	(55)
1.1.4 数据模型	(8)	2.3.10 与数据库操作相关的常用 函数	(56)
1.1.5 传统的三大数据模型	(9)	2.4 Visual FoxPro 数据库的基本概念	(57)
1.1.6 关系数据库	(10)	2.5 项目管理器简介	(57)
1.1.7 关系代数	(13)	2.5.1 新项目文件的创建	(58)
* 1.1.8 关系的规范化	(15)	2.5.2 已有项目文件的打开	(59)
1.1.9 数据库的设计	(19)	2.5.3 项目管理器窗口	(59)
1.2 Visual FoxPro 基础	(21)	2.5.4 项目管理器的使用	(62)
1.2.1 Visual FoxPro 6.0 的特点	(21)	2.6 向导、设计器、生成器简介	(64)
1.2.2 Visual FoxPro 6.0 的安装和 运行	(22)	2.6.1 向导简介	(64)
1.2.3 Visual FoxPro 6.0 的主要界面和 开发环境	(23)	2.6.2 设计器简介	(65)
1.2.4 Visual FoxPro 6.0 的配置	(25)	2.6.3 生成器简介	(66)
1.3 本章小结	(26)	2.6.4 表单向导的使用	(67)
习题与练习	(27)	2.6.5 报表向导的使用	(71)
第 2 章 Visual FoxPro 及使用入门	(29)	2.7 应用程序向导和应用程序生成器	(72)
2.1 数据类型、运算符和表达式	(29)	2.7.1 应用程序向导	(73)
2.1.1 数据类型	(29)	2.7.2 应用程序生成器	(74)
2.1.2 变量和常量	(31)	2.8 本章小结	(78)
2.1.3 运算符及表达式的构成	(34)	习题与练习	(78)
2.2 常用函数	(37)	第 3 章 Visual FoxPro 数据库及其操作	(81)
2.2.1 数值函数	(37)	3.1 Visual FoxPro 数据库的基本操作	(81)
2.2.2 字符函数	(40)	3.1.1 建立数据库	(81)
2.2.3 日期和时间函数	(43)	3.1.2 打开数据库	(82)
2.2.4 类型转换函数	(45)	3.1.3 打开数据库设计器	(85)
2.2.5 测试函数	(46)	3.1.4 删除数据库	(85)
2.3 数据的存储、查询和修改	(48)	3.2 建立数据库表	(86)
2.3.1 建立表	(49)	3.2.1 在数据库中建立表	(87)
2.3.2 打开表	(50)	3.2.2 修改表结构	(91)
2.3.3 输入数据	(51)	3.3 数据库表与自由表	(92)
2.3.4 记录定位	(51)	3.3.1 基本概念	(92)
		3.3.2 将自由表添加到数据库	(93)
		3.3.3 从数据库中移出表	(94)

3.4 使用 SQL 查询数据	(95)	4.1.1 视图的概念	(145)
3.4.1 SQL 查询概述	(95)	4.1.2 使用命令建立视图	(146)
3.4.2 简单查询	(96)	4.1.3 使用视图设计器建立视图	(148)
3.4.3 简单的连接查询	(98)	4.1.4 连接和远程视图	(152)
3.4.4 嵌套查询	(99)	4.1.5 视图与数据更新	(155)
3.4.5 几个特殊运算符	(100)	4.1.6 使用视图	(156)
3.4.6 利用空值查询	(101)	4.2 查询	(157)
3.4.7 排序	(102)	4.2.1 查询的概念	(157)
3.4.8 简单的计算查询	(103)	4.2.2 建立查询	(157)
3.4.9 分组与计算查询	(104)	4.2.3 使用查询	(160)
3.4.10 别名与自连接查询	(105)	4.3 本章小结	(163)
3.4.11 内外层互相关嵌套查询	(106)	习题与练习	(163)
3.4.12 使用量词和谓词的查询	(107)	第 5 章 Visual FoxPro 程序设计基础	(167)
3.4.13 超连接查询	(108)	5.1 概述	(167)
3.4.14 集合的并运算	(112)	5.1.1 Visual FoxPro 程序的基本概念	(167)
3.4.15 Visual FoxPro SQL SELECT 的几个特殊选项	(112)	5.1.2 程序文件的建立和执行	(167)
3.5 使用 SQL 操作数据	(114)	5.2 程序的基本结构和语句	(169)
3.5.1 插入	(115)	5.2.1 简单的输入输出命令	(169)
3.5.2 更新	(116)	5.2.2 顺序结构	(170)
3.5.3 删除	(116)	5.2.3 条件语句	(170)
3.6 索引	(116)	5.2.4 循环语句	(172)
3.6.1 基本概念	(117)	5.2.5 等待语句	(174)
3.6.2 在表设计器中建立索引	(118)	5.3 宏替换	(174)
3.6.3 用命令建立索引	(119)	5.4 数组及其应用	(175)
3.6.4 使用索引	(121)	5.4.1 基本概念与数组的定义	(175)
3.7 数据完整性与表之间的关联	(122)	5.4.2 数组的使用	(176)
3.7.1 实体完整性与主关键字	(122)	5.4.3 与数组相关的两条语句	(176)
3.7.2 域完整性与约束规则	(122)	5.5 多模块程序和过程	(177)
3.7.3 参照完整性与表之间的关联	(123)	5.5.1 过程文件与过程调用	(177)
3.7.4 数据完整性对操作的影响	(126)	5.5.2 参数的类型和参数传递	(178)
3.8 使用 SQL 语句定义表和完整性	(127)	5.5.3 变量的作用域	(179)
3.8.1 表的定义	(127)	5.5.4 过程应用实例	(180)
3.8.2 表的删除	(130)	5.6 程序调试	(182)
3.8.3 表结构的修改	(130)	5.6.1 调试器环境	(182)
3.9 其他	(132)	5.6.2 调试器菜单	(184)
3.9.1 多工作区的使用	(132)	5.6.3 断点的设置	(185)
3.9.2 表之间的临时关联	(133)	5.7 面向对象程序设计的基本概念	(187)
3.9.3 排序命令	(134)	5.7.1 对象与类	(187)
3.10 本章小结	(134)	5.7.2 子类与继承性	(188)
习题与练习	(135)	5.7.3 Visual FoxPro 中的类	(190)
第 4 章 视图与查询	(145)	5.7.4 可视化和面向对象开发方法的基本概念	(192)
4.1 视图	(145)		

5.7.5 控件、属性与事件	(193)	7.1.2 系统菜单的结构和组成	(241)
5.8 本章小结	(194)	7.2 设计菜单	(244)
习题与练习	(194)	7.2.1 设计菜单的一些原则	(244)
第6章 表单设计与应用	(201)	7.2.2 使用菜单设计器创建菜单	(244)
6.1 表单设计器及表单的设计	(201)	7.2.3 从用户菜单返回到系统菜单	(248)
6.1.1 表单设计器与创建表单	(201)	7.2.4 重新配置系统菜单	(249)
6.1.2 数据环境	(203)	7.2.5 修改菜单	(249)
6.1.3 控件的操作与布局	(204)	7.2.6 菜单的应用	(250)
6.1.4 表单的常用属性、事件和方法	(208)	7.2.7 使用编程方式建立菜单	(252)
6.1.5 表单的种类	(211)	7.3 快捷菜单	(253)
6.1.6 完成一个简单的程序	(211)	* 7.4 设计工具栏	(254)
6.2 控件与数据的绑定	(213)	7.4.1 建立工具栏	(254)
6.3 常用表单控件	(213)	7.4.2 将工具栏添加到表单集	(255)
6.3.1 标签控件	(213)	7.4.3 定义工具栏的操作	(256)
6.3.2 文本框控件	(213)	7.4.4 协调菜单和工具栏	(256)
6.3.3 编辑框控件	(215)	7.5 本章小结	(257)
6.3.4 命令按钮控件	(216)	习题与练习	(258)
6.3.5 命令按钮组控件	(217)	第8章 报表设计与应用	(260)
6.3.6 选项按钮控件	(218)	8.1 快速建立报表	(260)
6.3.7 复选框控件	(219)	8.1.1 快速报表	(260)
6.3.8 列表框控件	(220)	8.1.2 报表向导	(261)
6.3.9 组合框和下拉列表框控件	(224)	8.2 使用报表设计器	(265)
6.3.10 微调控件	(225)	8.2.1 报表设计器中的带区	(265)
6.3.11 表格控件	(226)	8.2.2 报表工具栏	(265)
6.3.12 图像控件	(228)	8.2.3 报表的数据源或数据环境	(266)
6.3.13 计时器控件	(229)	8.2.4 修改布局	(267)
6.3.14 页框控件	(230)	8.2.5 增添控件	(268)
* 6.4 表单与表单集	(231)	8.3 数据分组报表	(269)
6.4.1 表单集的作用	(231)	8.3.1 记录顺序与添加分组	(269)
6.4.2 表单集的建立	(231)	8.3.2 编辑组标头和组注脚带区	(270)
6.4.3 表单集的操作	(232)	8.4 多栏目标表	(270)
6.4.4 表单集的删除	(232)	8.5 报表的预览和打印	(271)
6.5 表单应用举例	(232)	8.5.1 预览报表	(271)
6.5.1 带输入和表格的表单应用程序	(232)	8.5.2 打印报表	(271)
6.5.2 含有页框控件的表单综合应用	(233)	8.6 报表应用举例	(273)
6.6 本章小结	(234)	8.6.1 含有单一分组的报表	(273)
习题与练习	(235)	8.6.2 含有多层分组的报表	(276)
第7章 菜单设计与应用	(240)	8.7 本章小结	(279)
7.1 Visual FoxPro 的系统菜单	(240)	习题与练习	(279)
7.1.1 基本概念	(240)	第9章 项目管理与应用程序连编	(281)
7.1.2 系统菜单的结构和组成	(241)	9.1 管理项目中的文件	(281)
7.2 设计菜单	(244)	9.1.1 文件的添加和移去	(281)
7.2.1 设计菜单的一些原则	(244)	9.1.2 文件的包含与排除	(282)
7.2.2 使用菜单设计器创建菜单	(244)		
7.2.3 从用户菜单返回到系统菜单	(248)		
7.2.4 重新配置系统菜单	(249)		
7.2.5 修改菜单	(249)		
7.2.6 菜单的应用	(250)		
7.2.7 使用编程方式建立菜单	(252)		
7.3 快捷菜单	(253)		
* 7.4 设计工具栏	(254)		
7.4.1 建立工具栏	(254)		
7.4.2 将工具栏添加到表单集	(255)		
7.4.3 定义工具栏的操作	(256)		
7.4.4 协调菜单和工具栏	(256)		
7.5 本章小结	(257)		
习题与练习	(258)		
第8章 报表设计与应用	(260)		
8.1 快速建立报表	(260)		
8.1.1 快速报表	(260)		
8.1.2 报表向导	(261)		
8.2 使用报表设计器	(265)		
8.2.1 报表设计器中的带区	(265)		
8.2.2 报表工具栏	(265)		
8.2.3 报表的数据源或数据环境	(266)		
8.2.4 修改布局	(267)		
8.2.5 增添控件	(268)		
8.3 数据分组报表	(269)		
8.3.1 记录顺序与添加分组	(269)		
8.3			

9.2 主程序与初始化	(283)	* 9.4 生成可发布的应用程序	(287)
9.2.1 设置应用的主程序	(283)	9.4.1 发布应用程序的步骤	(287)
9.2.2 初始化环境	(283)	9.4.2 准备要发布的应用程序	(287)
9.2.3 控制事件循环	(284)	9.4.3 定制要发布的应用程序	(289)
9.3 连编应用程序	(285)	9.4.4 创建发布磁盘	(290)
9.3.1 测试项目	(285)	9.5 本章小结	(291)
9.3.2 生成应用程序	(285)	习题与练习	(291)
9.3.3 生成可执行文件	(286)	参考文献	(294)
9.3.4 生成动态链接库	(286)		

第1章 Visual FoxPro 和数据库基础

在接触过计算机的人中,很多人都知道 FoxPro,知道 FoxPro 是数据库管理系统,人们对数据库已经不再陌生。

1.1 数据库基础

数据库是数据管理的工具。数据管理经历了从手工管理阶段、文件管理阶段到数据库管理阶段的变迁。数据库技术自产生以来便为广大用户所接受,并获得了广泛的应用。

1.1.1 什么是数据库

数据库,顾名思义就是存放数据的仓库,这种想当然的理解是不准确的。数据库对应的英文单词是 DataBase,如果直译则是数据基地;而数据仓库则另有其词——DataWarehouse。事实上数据库和数据仓库也不是同义词,数据仓库是在数据库技术的基础上发展起来的又一新的应用领域,是为辅助决策支持提供数据服务的技术。

由于近 20 多年 dBASE、FoxBASE 和 FoxPro 的应用非常普及,也有不少计算机用户认为 dBASE、FoxBASE 和 FoxPro 就是数据库,这当然也是一种错误的理解,dBASE、FoxBASE 或 FoxPro 是管理数据库的软件,而不是数据库。

那么,什么是数据库呢?虽然数据库不像数学定律那样有严格的定义,但仍然有大家公认的基本特征:数据库是相互关联的数据的集合。下面一段话全面概括了数据库应该具备的一些特征,也可以把它作为数据库的定义:

数据库是相互关联的数据的集合,它用综合的方法组织数据,具有较小的数据冗余,可供多个用户共享,具有较高的数据独立性,具有安全控制机制,能够保证数据的安全、可靠,允许并发地使用数据库,能有效、及时地处理数据,并能保证数据的一致性和完整性。

1. 相互关联的数据的集合

数据库中的数据不是孤立的,数据与数据之间是相互关联的。也就是说,在数据库中不仅要能够表示数据本身,还要能够表示数据与数据之间的联系。

比如在学籍管理中,有学生和课程两类数据,在数据库中除了要存放这两类数据之外,还要存放哪些学生选修了哪些课程或哪些课程由哪些学生选修这样的信息,这就反映了学生数据和课程数据之间的联系。

2. 用综合的方法组织数据

数据库能够根据不同的需要按不同的方法组织数据,比如可以有顺序组织方法、索引组织方法等。

3. 低冗余与数据共享

由于在数据库技术之前,数据文件都是独立的,所以任何数据文件都必须含有满足某一应

用的全部数据。比如,人事部门有一个职工文件,教育部门也有一个职工文件,人事部门的职工文件的记录格式是:

职工基本情况	有关人事管理的数据
--------	-----------

教育部门的职工文件的记录格式是:

职工基本情况	有关教育培训的数据
--------	-----------

这样在两个部门的职工文件中都有“职工基本情况”的数据,也就是说这一部分数据是重复存储的,如果还有第三、第四个部门也有类似的职工文件,那么重复存储所造成的空间浪费是很大的。在数据库中,可以共享类似“职工基本情况”这样的共用数据,从而降低数据的冗余度。

4. 数据具有较高的独立性

数据独立性是指数据的组织和存储方法与应用程序互不依赖、彼此独立的特性。在数据库技术之前,数据文件的组织方式和应用程序是密切相关的,当改变数据结构时相应的应用程序也必须随之修改,这样就大大增加了应用程序的开发代价和维护代价。而数据库技术却可以使数据的组织和存储方法与应用程序互不依赖,从而大大降低应用程序的开发代价和维护代价。

5. 保证数据的安全、可靠

数据库技术能够保证数据库中的数据是安全、可靠的。数据库有一套安全机制,可以有效地防止数据库中的数据被非法使用或非法修改;数据库还有一套完整的备份和恢复机制,保证当数据遭到破坏时(软件或硬件故障引起的),能立刻将数据完全恢复,从而保证系统能够连续、可靠地运行。

6. 最大限度地保证数据的正确性

保证数据正确的特性在数据库中称之为数据完整性。在数据库中通过建立一些约束条件来保证数据库中的数据是正确的。比如,某学生的年龄是20岁,当误输入为2岁或200岁时,数据库能够主动拒绝这类错误。

7. 数据可以并发使用并能同时保证数据的一致性

数据库中的数据是共享的,并且允许多个用户同时使用相同的数据,这就要求数据库能够协调一致,保证各个用户之间对数据的操作不发生矛盾和冲突,即在多个用户同时使用数据库时,能够保证数据的一致性和正确性。

以上概括介绍了数据库的主要特性和特征,在后续的章节中还会对它们做出更详细的解释,会介绍 Visual FoxPro 在这些方面的功能,使读者能够明确一些“为什么”、“什么是”、“如何做”等方面的问题,从而真正理解数据库、掌握 Visual FoxPro 的应用技术等。

需要说明的是,Visual FoxPro 毕竟是从个人计算机环境发展过来的小型数据库管理系统,虽然比以前的 dBASE、FoxBASE 和 FoxPro 完善了很多,但是与严格意义的大型数据库管理系统(如 SQL Server、Oracle 等)相比仍然不够完备。Visual FoxPro 只是具备了数据库管理系统的基本数据定义、数据管理和数据操作等功能,但是仍然缺乏数据可靠性和安全性等方面的功能。

1.1.2 数据库管理系统和数据库系统

数据库、数据库管理系统和数据库系统三个不同的概念。

1. 数据库管理系统

1.1.1 小节提到的数据库的各种功能和特性,并不是数据库中的数据固有的,它是靠管理或支持数据库的系统软件——数据库管理系统(DataBase Management System, DBMS)提供的。一个完备的数据库管理系统应该具备前面提到的各种功能,它的任务就是对数据资源进行管理,并且使之能为多个用户共享,同时还能保证数据的安全性、可靠性、完整性、一致性,以及数据的高度独立性。

具体来说,一个数据库管理系统应该具备如下功能。

(1) 数据库定义功能。可以定义数据库的结构和数据库的存储结构,可以定义数据库中数据之间的联系,可以定义数据的完整性约束条件和保证完整性的触发机制等。

(2) 数据库操纵功能。可以完成对数据库中数据的操纵,可以装入、删除、修改数据,可以重新组织数据库的存储结构,可以完成数据库的备份和恢复等操作。

(3) 数据库查询功能。可以以各种方式提供灵活的查询功能,使用户可以方便地使用数据库中的数据。

(4) 数据库控制功能。可以完成对数据库的安全性控制、完整性控制、多用户环境下的并发控制等方面的控制。

(5) 数据库通信功能。在分布式数据库或提供网络操作功能的数据库中还必须提供数据库的通信功能。

从以上诸功能来看,Visual FoxPro 主要欠缺的是数据库控制功能。

2. 数据库系统

前面介绍了数据库的各种特征和管理数据库的系统软件 DBMS,那么什么是数据库系统呢?

简单地讲,数据库系统就是基于数据库的计算机应用系统。这样一个系统首先要包括:

- 以数据为主体的数据库;
- 管理数据库的系统软件 DBMS。

此外还要包括:

- 支持数据库系统的计算机硬件环境和操作系统环境;
- 管理和使用数据库系统的人,特别是负责设计、维护数据库的技术人员——数据库管理员;
- 方便使用和管理系统的各种技术说明书和使用说明书。

至此可以看出,数据库、数据库管理系统和数据库系统三个不同的概念,数据库强调的是数据,数据库管理系统是系统软件,而数据库系统强调的是系统。

1.1.3 数据管理的发展历程

数据库的核心任务是数据管理,它包括数据的分类、组织、编码、存储、检索和维护等。但并不是一开始就有数据库技术,在计算机诞生的初期,计算机主要用于科学计算,虽然此时同

样有数据管理的问题,但这时的数据管理是以人工的方式进行的,后来发展到文件系统,再后来才是数据库。也就是说,数据管理经历了人工管理阶段、文件系统阶段和数据库系统阶段。

1. 人工管理阶段

人工管理阶段是指计算机诞生的初期(20世纪50年代中期以前)。这个时期的计算机技术,从硬件看还没有磁盘这样的可直接存取的存储设备,从软件看没有操作系统,更没有管理数据的软件。所以这个时期数据管理有如下特点:

(1) 数据不保存。因为计算机主要用于科学计算,一般也不需要长期保存数据,只是在完成某一个计算或课题时才将数据输入,然后不仅原始数据不保存,计算结果也不保存。

(2) 还没有文件的概念。这个时期的数据组织必须由每个程序的程序员自行组织和安排。

(3) 一组数据对应一个程序。每组数据只对应一个应用,即便两个程序用到相同的数据,也必须各自定义、各自组织,数据无法共享、无法相互利用和互相参照。因此,程序和程序之间有大量的数据重复。

(4) 没有形成完整的数据管理的概念。由于以上几个特点及没有对数据进行管理的软件系统,所以这个时期的每个程序都要包括数据存取方法、输入/输出方法和数据组织方法等。因为程序是直接面向存储结构的,所以存储结构的任何一点修改,都会导致程序的修改,程序与数据不具有独立性。

人工管理阶段的特点可用图1-1表示。

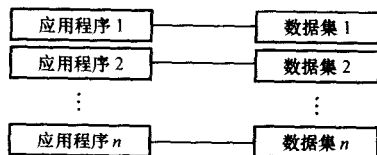


图1-1 数据的人工管理阶段

2. 文件系统阶段

文件系统阶段是指20世纪50年代后期到60年代中期这一阶段。从那时起,计算机不仅大量应用于科学计算,也开始应用于信息管理。像磁盘这样的直接存取存储设备也已经出现,在软件方面也有了操作系统和高级语言,也有了专门用于数据管理的软件——即文件系统(或操作系统的文件管理部分)。这个阶段的数据管理有以下一些特点:

(1) 数据可以长期保存在磁盘上,也可以反复使用,即可以经常对文件进行查询、修改、插入和删除等操作。

(2) 操作系统提供了文件管理功能和访问文件的存取方法,程序和数据之间有了数据存取的接口,程序开始通过文件名和数据打交道,可以不再关心数据的物理存放位置。因此,这时也有了数据的物理结构和数据的逻辑结构的区别。程序和数据之间有了一定的独立性。

(3) 文件的形式已经多样化。由于有了磁盘这样的直接存取存储设备,文件也就不再局限于顺序文件,也有了索引文件、链表文件等。因而,对文件的访问可以是顺序访问,也可以是直接访问。但文件之间是独立的,它们之间的联系要通过程序去构造,文件的共享性还比较差。

(4) 有了存储文件以后,数据就不再仅仅属于某个特定的程序,而可以由多个程序反复使用。但文件结构的设计仍然是基于特定的用途,程序仍然是基于特定的物理结构和存取方法编制的。因此,数据的存储结构和程序之间的依赖关系并未根本改变。

(5) 数据的存取基本上以记录为单位。

图 1-2 示意了文件系统阶段的特点及程序和数据之间的关系。

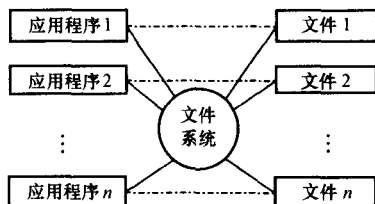


图 1-2 数据管理文件系统阶段的特点

虽然文件系统比人工管理有了长足的进步,但是文件系统所能提供的存取方法和数据管理仅仅是初级水平。无论如何,文件系统阶段是数据管理技术发展中的一个重要阶段,在这个阶段中得到充分发展的各种数据结构和算法等,都大大丰富了计算机科学,今天的数据库技术也正是在文件系统的基础上发展起来的。

现在,站在数据库的观点上反观文件系统,可以看出文件系统有以下几方面明显的缺陷:

(1) 数据冗余大。这是因为每个文件都是为特定的用途设计的,因此就会造成同样的数据在多个文件中重复存储。

(2) 数据不一致性。这往往是由数据冗余造成的,在进行更新时,稍不谨慎就会造成同一数据在不同文件中的不一致。

(3) 程序和数据之间的独立性差。应用程序依赖于文件的存储结构,使得若修改文件的存储结构则必须修改程序。

(4) 数据联系弱。文件与文件之间是独立的,文件之间的联系必须通过程序来构造。因此,文件系统是一个不具有弹性的、无结构的数据集合,不能反映现实世界事物之间的联系。

3. 数据库系统阶段及其发展过程

数据库系统阶段从 20 世纪 60 年代后期开始,数据库技术的诞生既有计算机技术的发展做依托,又有数据管理的需求做动力。数据库的数据不再是面向某个应用或某个程序,而是面向整个企业(组织)或整个应用的,图 1-3 示意了这种特点。

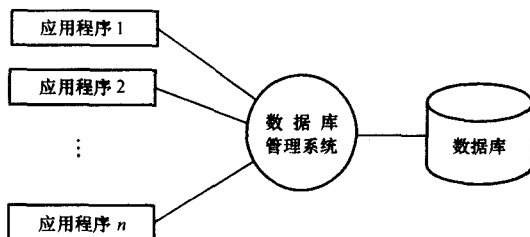


图 1-3 数据管理的数据库系统阶段的特点

关于数据库的特点和优点前面已经做了比较详细的叙述,这里不再重复。

数据库技术的诞生可以以 20 世纪 60 年代末和 70 年代初的三个事件做标志,它们是:

(1) 1968 年美国 IBM 公司研制成功、1969 年形成产品的信息管理系统 IMS (Information Management System),它曾是历史上最大的数据库系统之一,该系统支持的是层次数据模型。

(2) 美国数据系统语言协会 CODASYL (Conference On DATA SYstem Language) 下属的数据库任务组 DBTG (DataBase Task Group) 对数据库方法进行了系统的研究,在 20 世纪 60 年代末和 70 年代初发表了若干个报告(称为 DBTG 报告),该报告建立了数据库技术的很多概念、方法和技术。DBTG 所提议的方法是基于网状数据模型的。

(3) 从 1970 年起,IBM 的研究员 E.F.Codd 发表了一系列的论文,提出了数据库的关系模型,开创了数据库关系方法和关系数据理论的研究,为关系数据库的发展和理论研究奠定了基础。

可以说,20 世纪 70 年代是以 IMS 为代表的层次数据库和以 DBTG 为代表的网状数据库的鼎盛时期。这些数据库都有很高的效率,特别是 DBTG 报告包含了很多完备的概念和技术,这大大推动了计算机在信息管理领域的应用,也使许多商品化的数据库开始出现,数据库技术日益广泛地应用到企业管理、交通运输、情报检索、军事指挥、政府管理和辅助决策等各个方面。层次数据库和网状数据库是面向专业人员的,要求使用人员要有较高的技术水平和专业水平,使用起来比较困难。

20 世纪 70 年代也是关系数据库的萌芽时期。与层次数据库和网状数据库相比,研究关系数据库的出发点之一就是简单、易用,这样就要尽可能减少人为操作,让计算机自动地完成更多的工作。受当时计算机硬件和软件技术的约束,关系数据库经历了比较长时间的研究过程。尽管在 20 世纪 60 年代末和 70 年代初就提出了关系数据库的概念,但是关系数据库真正得到广泛的应用,则是进入 20 世纪 80 年代以后的事情。

在我国,20 世纪 70 年代使用数据库的单位主要集中在国家部委、国防军事、气象预报、石油勘探等一些特殊的行业和部门。而数据库技术真正得到广泛推广和使用,可以说是从 20 世纪 80 年代初的 dBASE II 开始的,尽管 dBASE II、甚至其后的 xBASE 系列都不能称为是完备的数据库管理系统,但是,它们支持基本的关系数据模型,使用起来非常方便,一般也能满足规模不大或中、小规模的信息管理的应用。现在人们使用的各种数据库系统几乎都是基于关系数据模型的,层次数据库和网状数据库由于自身的缺点均已被关系数据库所取代。

从 20 世纪 80 年代算起,又经过了 20 多年,关系数据库系统的技术也在不断完善和提高,有关数据库的新的研究课题不断取得进展,如分布式数据库已经成为实用技术,基于关系模型、支持面向对象方法的关系对象模型数据库也已经问世等。数据库技术还会朝着支持更大规模、更快速度、更广泛的应用等方向发展。

4. 分布式数据库系统

传统的数据库系统是集中式数据库,也就是说,整个数据库是存放在一台计算机或服务器上的。这种系统的数据采取集中管理的方式,要求主机或服务器有比较大的容量,这样比较容易实现和管理。但是,随着数据库应用的不断发展和数据库应用规模的不断扩大,越来越感到集中式数据库有很多缺陷和不便。

随着计算机技术和网络技术的发展,人们希望数据能分布存储在网络上的各台计算机上,为此产生了分布式数据库系统。

分布式数据库是一个物理上分布于计算机网络的不同地点、而逻辑上又属于同一系统的

数据集合。网络上每个地点的数据库都有自治能力,能够完成局部应用;同时每个地点的数据库又属于整个系统,通过网络也可以完成全局应用。

物理上分布、逻辑上集中是分布式数据库的最明显特点,也就是说,用户使用的是一个整体数据库,他们不需要知道哪些数据存放在什么地方,在分布式数据库中把这种特性称为位置透明性。

5. 客户/服务器数据库结构

客户/服务器(Client/Server)数据库结构是一种数据库应用的体系结构,它强调客户端和服务器的分工与合作,使整个系统达到最高的效率。

客户端是面向最终用户的,所以它的主要任务是提供友好的用户界面,提交数据访问请求以及接收和处理数据库的返回结果,组织返回数据的输出(如生成数据浏览窗口、生成数据报表和图形等),提供初步的数据验证功能等。

数据库服务器则完成数据管理、信息共享、安全管理以及一些更高级的管理。数据库服务器实际由一个支持客户/服务器结构的数据库管理系统来支持。

客户/服务器结构是一种开放的体系结构,可以接受来自各种应用程序和开发工具的客户端的连接。其中最具代表性的连接协议是 ODBC(Open DataBase Connectivity),中文含义是开放式数据库连接。

Visual FoxPro 不仅是一个独立的数据库管理系统,它还可以作为其他数据库管理系统的客户端开发工具。例如,可以在 Visual FoxPro 中通过 ODBC 连接到 SQL Server 对 SQL Server 数据库进行操作等。

6. 面向对象数据库系统

面向对象数据库是面向对象技术和数据库技术相结合的产物。面向对象数据库是指对象的集合、对象的行为、状态和联系是以面向对象数据模型来定义的。面向对象数据库系统是支持定义和操作面向对象数据库的数据库系统。

一个面向对象数据库系统应该具备如下的基本内容和特征:

- (1) 在数据模型方面支持对象、复合对象、封装、类、继承、重载、多态性等基本概念。
- (2) 数据库管理系统除了具有传统的数据库管理系统所具有的功能(如并发控制、故障和恢复)外,还支持永久对象、长事务处理和嵌套事务,能维护数据完整性,适合在分布式环境下工作。
- (3) 数据库访问界面要支持消息传递,提供计算能力完备的数据库程序设计语言,能解决数据库语言与宿主语言的某些不匹配问题,提供类似 SQL 的非过程化查询功能。

Visual FoxPro 虽然不支持面向对象的数据模型,但是对程序设计语言进行了扩充,以支持面向对象的一些程序设计思想。

7. 数据库技术的广泛应用领域

数据库技术的最初应用领域主要是信息管理领域,如政府部门、工商企业、图书情报、交通运输、银行金融、科研教育等各行各业的信息管理和信息处理。事实上,只要有大量的数据要管理、需要有大量数据支持的工作,都可以使用数据库。下面再介绍几个比较有代表性的应用领域:

- (1) 因特网上的 Web 数据库。因特网用一种统一的网络协议将全世界的计算机连成网

络,最初它只能提供事先制作好的静态信息。后来,人们为了从因特网上得到动态的、实时的信息,将数据库技术引入因特网,从而有了 Web 数据库,Web 数据库是因特网上数据库的一种称呼。

(2) 多媒体数据库。这种数据库不仅可以存储和管理文字和报表数据,还可以存储和管理如声音、图片、动画等各种媒体的数据,这种多媒体数据库可以支持广泛的应用。

(3) 工程数据库。主要用于管理工程数据,如计算机辅助设计(CAD)和计算机辅助制造(CAM)过程中使用的数据,以及设计的图纸、工艺流程等数据的存储和管理。

(4) 辅助决策支持。早期的模型库、方法库和现在广泛讨论的数据仓库技术都是用于辅助决策支持的。

(5) 人工智能领域的知识库。人工智能是从 20 世纪 60 年代开始发展起来的研究机器智能和智能机器的高科技学科,它需要大量的演绎和推理规则的支持,这无疑又为数据库提供了一个用武之地。通过将人的知识抽象化、条理化,利用数据库技术建立知识库,从而使数据库智能化。

以上只是概括介绍了一些数据库的应用领域,实际上还远远不止这些,这些技术也不只是用于单一领域,通过将这些技术结合可以有更广阔的领域,或者说一些应用领域需要综合数据库技术的支持。

1.1.4 数据模型

数据库是相互关联的数据的集合。为了理解这种关联,需要通过信息结构和数据模型来理解这一层的含义。

1. 信息结构与实体之间的联系

数据需要人们的认识、理解、整理、规范和加工,然后才能存放到数据库中。

例如在日常的学生成绩管理中,首先涉及的是学生、课程以及学生课程的考试成绩等,这种管理称为现实世界管理。在现实世界管理中这些被管理的对象称为实体(Entity)。实体可以定义为:客观存在并可以相互区分的客观事物或抽象事件。

每个实体肯定具有一定的特征(性质),这样人们才能根据实体的特征来区分一个个实体。例如学生的学号、姓名、年龄等都是学生实体具有的特征;足球赛的比赛时间、地点、参赛队等都是足球赛实体的特征。

把具有相同特征的一类实体的集合称为实体集。比如,所有的学生、所有的教师、所有的订货都构成各自的实体集。

实体的特征可以用来区分实体,但并不是所有的特征都能达到区分实体的目的。比如,职工的工资特征就不能区分一个个职工实体。所以要强调区分对象的特征或区分实体的特征——标志特征。比如,职工号就可以用来区分一个个职工,职工号是职工的标志特征,而工资就不是职工的标志特征。

人们不仅关心实体,还关心实体与实体之间的联系,并且根据实体与实体之间的对应关系将实体与实体之间的联系分为三种类型,它们是一对一联系、一对多联系和多对多联系。

(1) 一对一联系。如果实体集 A 与实体集 B 之间存在联系,并且对于实体集 A 中的任意一个实体,实体集 B 中最多只有一个实体与之对应;而对实体集 B 中的任意一个实体,在实体集 A 中也最多只有一个实体与之对应,则称实体集 A 到实体集 B 的联系是一对一的,记为

1:1。

(2) 一对多联系。如果实体集 A 与实体集 B 之间存在联系,并且对于实体集 A 中的任意一个实体,实体集 B 中可以有多个实体与之对应;而对实体集 B 中的任意一个实体,在实体集 A 中都最多只有一个实体与之对应,则称实体集 A 到实体集 B 的联系是一对多的,记为 $1:n$ 。

例如有考场和考生两个实体,并且规定:一个考场可以有多名考生,但是一个考生只能在一个考场参加考试。那么考场和考生之间的联系是一对多的。

(3) 多对多联系。如果实体集 A 与实体集 B 之间存在联系,并且对于实体集 A 中的任意一个实体,实体集 B 中可以有多个实体与之对应;而对实体集 B 中的任意一个实体,在实体集 A 中也可以有多个实体与之对应,则称实体集 A 到实体集 B 的联系是多对多的,记为 $m:n$ 。

例如有学生和课程两个实体,并且规定:一个学生可以选修多门课程,一门课程可以由多个学生选修。那么学生和课程之间的联系就是多对多的。

在数据库中不仅要存放表示实体的数据,还要存放表示实体之间联系的数据。在 Visual FoxPro 中实体和联系都用表(Table)来表示。

2. 实体 - 联系模型

描述现实世界的模型称为概念数据模型,它描述的是现实世界中实体以及实体与实体之间的联系。目前描述概念数据模型所使用的最广泛的方法就是实体 - 联系方法,简称 E - R (Entity - Relationship)方法,该方法使用的主要工具是实体 - 联系图(即 E - R 图),也将用 E - R 方法描述的模型称为 E - R 模型。

E - R 方法的要素主要包括实体和联系的表示方法。以前用手工绘制 E - R 图,一般用矩形框表示实体,用连接实体的连线表示实体之间的联系,用椭圆框表示属性(用连线连到相关的实体)。现在有各种数据库设计和建模工具可以使用,一般不再手工绘制 E - R 图,但是最重要的要素仍然是实体和联系的表示方式。

用 E - R 方法描述的概念数据模型是对现实世界信息结构的描述,它不依赖于具体的数据库管理系统。

1.1.5 传统的三大数据模型

上一小节讨论的概念数据模型是“概念上”的、抽象的,它与具体的数据库管理系统无关。本节要介绍的数据模型将与具体的数据库管理系统有关,与数据库管理系统支持的数据和联系的表示或存储方法有关。

可能读者听过或见过“关系数据库”或“关系数据库管理系统”这样的词语(通常 Visual FoxPro 就被称为关系数据库管理系统),读者可能会问这里的“关系”是什么意思?这是指这个数据库或数据库管理系统所支持的数据模型是基于关系方法的。除此之外,传统的数据模型还有层次模型和网络模型。

前面提到过,数据库中不仅要存放数据本身,还要存放数据与数据之间的联系,可以用不同的方法表示数据与数据之间的联系,把表示数据与数据之间联系的方法称之为数据模型。

1. 层次数据模型

用树形结构来表示实体之间联系的模型称为层次模型。

构成层次模型的树是由结点和连线组成的,结点表示实体集(文件或记录型),连线表示相

连两个实体之间的联系,这种联系只能是一对多的。通常把表示“一”的实体放在上方,称为父结点;而把表示“多”的实体放在下方,称为子结点。

层次模型表示一对多的联系是直接而方便的。但层次模型有以下两点限制:

- 有且仅有一个结点无父结点,这个结点即为树的根;
- 其他结点有且仅有一个父结点。

这样就使得多对多联系不能直接用层次模型表示。

支持层次模型的数据库管理系统称为层次数据库管理系统。

2. 网状数据模型

如果取消层次模型中的两点限制,即允许每一个结点可以有多个父结点,便形成了网状结构。

用网状结构来表示实体之间联系的数据模型称为网状数据模型或网络数据模型。

由于网状模型没有层次模型的两点限制,所以可以直接表示多对多的联系。但是,多对多的联系实现起来太复杂了,所以在一些实际的支持网状模型的数据库管理系统上,对多对多联系还是做了限制。

网状模型和层次模型在本质上是一样的,从逻辑上看它们都是用连线表示实体之间的联系,用结点表示实体集;从物理上看,层次模型和网状模型都是用指针来实现两个文件之间的联系,其差别仅在于网状模型中的连线或指针更加复杂,更加纵横交错,从而使数据结构更复杂。

在网状模型中同样使用父结点和子结点这样的术语,并且同样把父结点安排在子结点的上方。

支持网状数据模型的数据库管理系统称为网状数据库管理系统。

层次数据模型和网状数据模型盛行于 20 世纪 70 年代,由于这种模型对用户来说太复杂,所以从 20 世纪 80 年代开始逐步被关系数据模型所取代。

3. 关系数据模型

关系数据模型源于数学,它把数据看成是二维表中的元素,而这个二维表就是关系。

用关系(表格数据)表示实体和实体之间联系的模型称为关系数据模型。

通俗地讲,关系就是二维表格,表格中的每一行称为是一个元组,它相当于一个记录值,每一列是一个属性值集,列可以命名,称为属性名。这里的属性与前面讲到的实体的属性(特征)或记录的字段意义相当。由此可以说,关系是元组的集合,如果表格有 n 列,则称该关系是 n 元关系。

在关系数据模型中实体本身以及实体与实体之间的联系都用关系来表示,实体之间的联系不再通过指针来实现。

1.1.6 关系数据库

Visual FoxPro 就是基于关系模型的,所以说 Visual FoxPro 是关系数据库管理系统。

1. 关系术语

在关系模型和关系数据库中使用关系表示实体以及实体和实体之间的联系,这里给出在关系数据库中使用的一些术语。