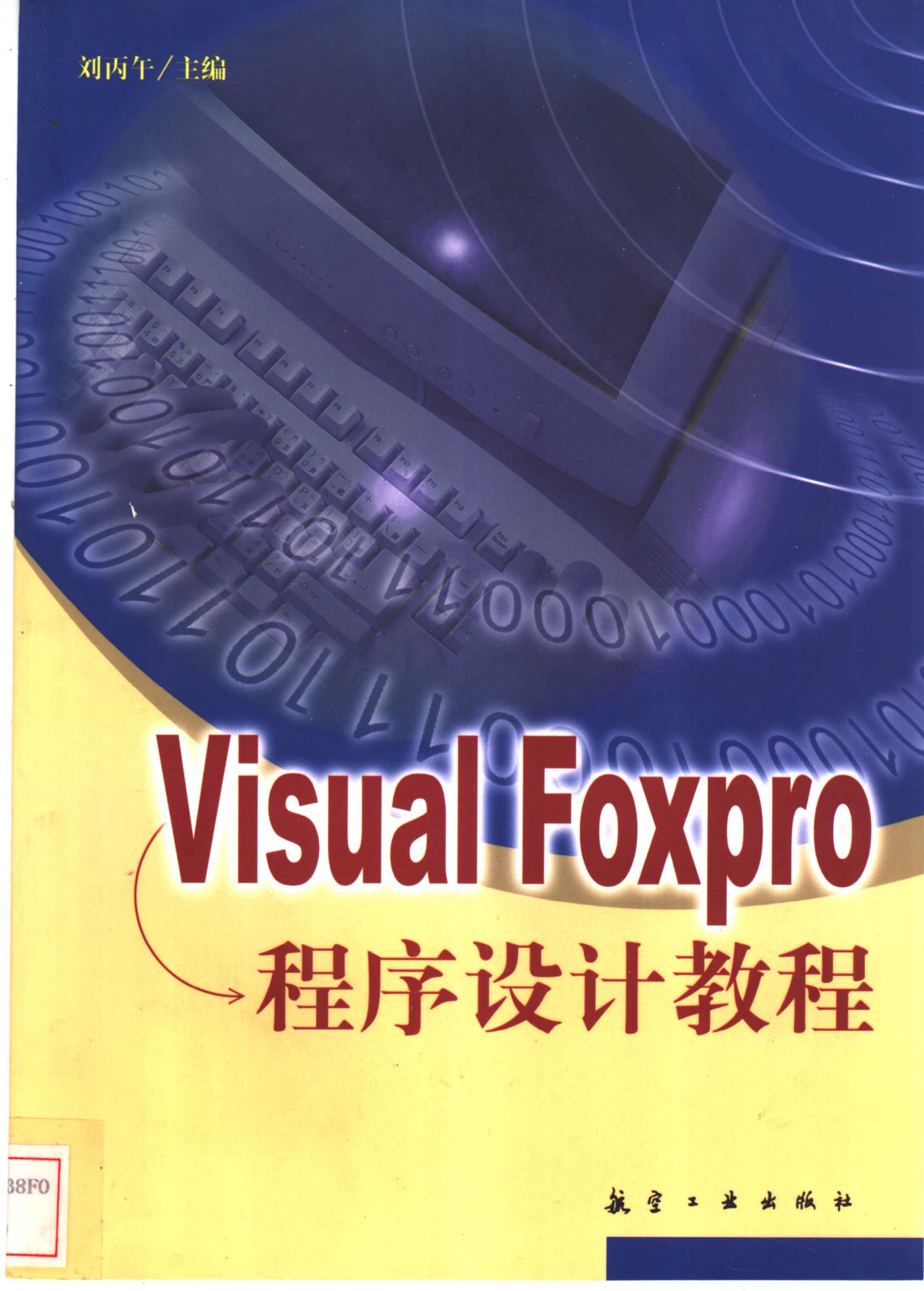


刘丙午/主编



Visual Foxpro

程序设计教程

航空工业出版社

38FO

Visual FoxPro 程序设计教程

刘丙午 主编

航空工业出版社

内 容 提 要

本书以 Visual FoxPro 6.0 为主详细介绍了数据库程序设计方法, 内容主要包括数据库基础知识、Visual FoxPro 语法基础、Visual FoxPro 开发环境、数据库和表操作、查询和视图的使用、程序设计基础、表单设计、报表和标签设计、菜单设计、数据的导入与导出等。此外, 本书还简要介绍了另外两种数据库开发系统: Access 2000 与 SQL Server 2000。

全书实例众多、语言流畅, 并且每章都附有思考与练习, 特别适合大专院校作为教材, 也可供各类培训班和管理系统开发、使用与维护人员使用。

图书在版编目 (CIP) 数据

Visual FoxPro 程序设计教程 / 刘丙午主编. —北京:
航空工业出版社, 2004. 2
ISBN 7-80183-328-7

I. I... II. 刘... III. 关系数据库—数据库管理系统,
Visual FoxPro—程序设计—教材
IV. TP311. 138

中国版本图书馆 CIP 数据核字 (2004) 第 005683 号

航空工业出版社出版发行

(北京市安定门外小关东里 14 号 100029)

北京市金顺印刷厂印刷

全国各地新华书店经售

2004 年 2 月第 1 版

2004 年 2 月第 1 次印刷

开本: 787×1092

1 / 16

印张: 18.5

字数: 453 千字

印数: 1 ~ 5000

定价: 25.00 元

本社图书如有缺页、倒页、脱页、残页等情况, 请与本社
发行部联系调换。联系电话: 010-84917422 或 82822965

编者的话

Visual FoxPro 是目前微机上最流行的关系数据库语言之一，它以卓越的数据库处理性能、良好的开发环境赢得了广大用户的喜爱。用户可通过 Visual FoxPro 的开发环境方便地设计数据库结构、管理数据库、设计应用程序界面、设计查询、设计报表、设计菜单，并可利用项目管理器对数据库和程序进行管理、生成可执行文件，以及进行发布等。

基于 Visual FoxPro 所具有的各种优点，目前绝大部分学校都以该软件为例，向学生介绍有关数据库方面的知识。与目前已经出版的相关教材相比，本书主要具有以下一些特点：

- 将理论与实践进行了很好的结合。本书首先介绍了与数据库程序开发相关的一些背景知识，然后详细介绍了使用 Visual FoxPro 进行数据库程序设计的方法。
- Visual FoxPro 开发环境介绍与程序设计并重。由于 Visual FoxPro 本身即具备强大的数据库管理功能，用户可直接借助其开发环境创建、管理和应用数据库。
- 本书除了详细介绍了 Visual FoxPro 外，还简要介绍了另外两种流行的数据库开发平台：Access 2000 与 SQL Server 2000。
- 每章都有精心设计的思考与练习题，从而便于教学。
- 本书作者均为一线教学人员，具有长期的数据库教学经验，从而确保了本书的质量。

本书各章内容安排如下：

- 第 1 章首先介绍了数据库理论知识，从而便于读者对数据库技术的现状、基本概念，数据库管理系统的主要功能和组成有一个大致的了解，进而便于后面各章的学习。
- 第 2 章主要介绍了 Visual FoxPro 的基本语法，如数据类型、运算符、表达式、函数以及数据存储方式等，这是学习 Visual FoxPro 语言必须了解的。
- 第 3 章介绍了 Visual FoxPro 的开发环境，具体包括项目管理器的使用、文本编辑和命令窗口使用方法，以及 Visual FoxPro 设计器、向导和生成器分类解析。值得指出的是，Visual FoxPro 不仅可以用于开发数据库管理系统，而且可以直接创建和管理数据库。
- 第 4 章主要介绍了使用 Visual FoxPro 开发环境和相关命令创建和管理数据库和表的方法。
- 第 5 章介绍了数据查询与视图应用。其中，视图是 Visual FoxPro 为应用程序和用户提供了一种定制的、且可更改的数据集合，它具备了表和查询的特点并被保存在数据库中。
- 第 6 章介绍了 Visual FoxPro 程序设计，即如何借助 Visual FoxPro 开发面向特定用途的数据库管理系统。
- 第 7 章介绍了表单设计方法。所谓表单 (Form)，就是应用程序中最常见的交互式操作界面，各种对话框和窗口都是表单不同的外观表现形式。
- 第 8 章与第 9 章分别介绍了如何借助报表设计器 (Report Designer) 设计报表与标签，以及如何借助菜单设计器和命令设计菜单。
- 第 10 章介绍了如何向 Visual FoxPro 数据库中导入数据，以及从 Visual FoxPro 数据

库中导出数据。

- 第 11 章和第 12 章简要介绍了另外两种流行的数据库开发系统: Access 2000 和 SQL Server 2000。

尽管作者已尽了很大努力, 但错误可能在所难免, 热情欢迎广大读者批评指正。

此外, 本教材的出版得到了北京物资学院教材建设基金的资助, 在此对相关领导及人员深表谢意。

编 者
2004 年 1 月

本书编委会

主 编 刘丙午

副 主 编 秦惠林 董萍萍

参编人员 申贵成 朱韶红 张 博 岳溥庥

刘玉海 刘 冲 郭 风

目 录

第1章 数据库基础知识	1	2.3.5 逻辑运算符及逻辑表达式	28
1.1 数据库概述	1	2.3.6 名称表达式	28
1.1.1 数据管理技术的产生与发展	1	2.4 函数	29
1.1.2 数据库基本概念	4	2.4.1 数值处理函数	29
1.2 数据库系统结构	5	2.4.2 字符处理函数	31
1.2.1 数据库系统的三级模式	5	2.4.3 日期和时间处理函数	33
1.2.2 数据库的二级映像功能与数据 独立性	7	2.4.4 数据类型转换函数	34
1.3 数据模型	7	2.4.5 测试函数	36
1.3.1 数据模型的要素	7	2.4.6 其他函数	38
1.3.2 实体联系模型	8	思考与练习	40
1.3.3 常用的数据模型	10	第3章 Visual FoxPro 开发环境	42
1.4 关系数据库	11	3.1 Visual FoxPro 的启动和环境设置	42
1.4.1 关系数据结构	11	3.1.1 Visual FoxPro 的启动	42
1.4.2 关系操作	11	3.1.2 Visual FoxPro 的环境设置	42
1.4.3 关系完整性约束	11	3.2 项目管理器应用	44
1.5 Visual FoxPro 6.0 简介	12	3.2.1 使用项目组织文件	45
1.5.1 Visual FoxPro 6.0 的特点	12	3.2.2 打开项目管理器	48
1.5.2 Visual FoxPro 6.0 的工作方式	13	3.2.3 项目管理器窗口概述	48
1.5.3 系统主界面	13	3.2.4 项目管理器窗口操作	48
1.5.4 Visual FoxPro 6.0 命令的一般形式	14	3.2.5 组成文件列表	49
1.5.5 系统文件类型	15	3.2.6 项目管理器的按钮	50
思考与练习	16	3.2.7 查看或填充项目信息	51
第2章 Visual FoxPro 语法基础	18	3.2.8 从一个项目文件中向另一个项目 文件中加入文件	52
2.1 数据类型	18	3.2.9 建立自己的项目文件	52
2.2 数据存储	19	3.3 Visual FoxPro 中的文本编辑	54
2.2.1 常量	20	3.3.1 光标移动控制键	54
2.2.2 变量	21	3.3.2 文本编辑	54
2.2.3 其他数据存储容器	25	3.3.3 改变字体	57
2.3 运算符和表达式	25	3.3.4 命令续行	57
2.3.1 数值运算符及数值表达式	25	3.3.5 增加和撤销注释	58
2.3.2 字符串运算符及字符表达式	26	3.3.6 行缩进	58
2.3.3 日期运算符及日期表达式	26	3.3.7 改变编辑器属性	58
2.3.4 关系运算符及关系表达式	27	3.4 命令窗口使用方法	58

3.4.1 命令输入和编辑	59	4.5 数据表的维护	96
3.4.2 出错处理	60	4.5.1 修改表结构	96
3.4.3 执行带有续行符的命令	60	4.5.2 从已有的文件中向表中添加记录	97
3.5 使用键盘宏	60	4.5.3 编辑修改记录	99
3.5.1 键盘宏的建立	61	4.5.4 删除记录	102
3.5.2 编辑宏	62	4.6 建立数据字典	105
3.5.3 宏的使用	63	4.6.1 为数据库表设置字段属性	105
3.5.4 使用宏时应注意的问题	63	4.6.2 设置表的属性	107
3.6 Visual FoxPro 设计器、向导和 生成器分类解析	64	4.6.3 创建永久关系	108
3.6.1 设计器	64	4.6.4 设置参照完整性	110
3.6.2 向导	65	4.7 数据排序与索引	111
3.6.3 生成器	68	4.7.1 数据表的排序	111
3.7 Visual FoxPro 系统键盘定义	69	4.7.2 创建索引	112
3.7.1 菜单快捷键	69	4.7.3 数据表查询	117
3.7.2 窗口操作组合键	70	4.7.4 数据表的统计和汇总	118
3.7.3 帮助键	70	4.8 使用多个表	120
3.7.4 属性窗口组合键	70	4.8.1 工作区的选择	120
3.7.5 系统缺省键盘宏定义	71	4.8.2 数据表的关联和使用关联	122
思考与练习	71	思考与练习	124
第 4 章 数据库和表操作	73	第 5 章 查询与视图	129
4.1 数据库设计	73	5.1 设计查询	129
4.1.1 数据库设计步骤	73	5.1.1 启动查询设计器或查询向导	130
4.1.2 分析数据需求	74	5.1.2 设置表间联接条件	131
4.1.3 将需求分类放入表	74	5.1.3 选定查询输出字段	133
4.1.4 确定所需字段	75	5.1.4 设置联接条件和筛选条件	133
4.1.5 确定关系	76	5.1.5 排序查询结果	136
4.1.6 设计求精	78	5.1.6 分组查询结果	136
4.2 数据库创建与操作	79	5.1.7 限制查询结果	137
4.2.1 数据库的创建	79	5.1.8 定向输出查询结果	138
4.2.2 数据库的基本操作	80	5.1.9 运行查询	141
4.3 创建数据表	81	5.1.10 修改查询	142
4.3.1 数据表的结构	81	5.1.11 关于 SQL	142
4.3.2 创建自由表并输入数据	82	5.1.12 举例	143
4.3.3 创建数据库表	86	5.2 使用视图更新数据	146
4.4 数据表的基本操作	88	5.2.1 创建视图	146
4.4.1 数据表的浏览	88	5.2.2 设置更新条件	148
4.4.2 记录的定位	93	5.2.3 定制视图	151
		5.2.4 举例	152

5.3 结构化查询语言 SQL	154	7.4 控件使用技巧	219
5.3.1 SQL 的主要特点	154	7.4.1 使控件更容易使用的方法	219
5.3.2 SQL 语言的基本概念	155	7.4.2 允许拖放操作	220
5.3.3 SQL 的数据定义功能	156	思考与练习	221
5.3.4 SQL 的数据查询功能	158	第 8 章 报表和标签	223
5.3.5 SQL 的数据更新功能	160	8.1 报表设计的步骤与方法	223
思考与练习	161	8.1.1 报表设计的步骤	223
第 6 章 程序设计基础	163	8.1.2 创建报表布局的方法	223
6.1 程序文件的建立与执行	163	8.1.3 报表类型	223
6.1.1 基本概念	163	8.2 报表设计示例	224
6.1.2 程序文件的建立与修改	163	8.2.1 启动报表向导或打开报表设计器	224
6.1.3 程序文件的执行	164	8.2.2 使用报表向导创建初步报表	224
6.2 程序设计常用命令	165	8.2.3 利用报表设计器调整报表	227
6.2.1 输入/输出命令	165	8.3 报表设计器使用详解	229
6.2.2 程序流程图	168	8.3.1 报表设计器中的带区	229
6.2.3 分支命令	168	8.3.2 设置报表设计器的数据环境	230
6.2.4 循环命令	172	8.3.3 报表控件	231
6.3 子程序、过程与自定义函数	177	8.3.4 修改报表布局	232
6.3.1 子程序	177	8.3.5 更改页面	236
6.3.2 自定义函数	178	8.3.6 在布局上分组数据	237
6.3.3 过程与过程文件	179	8.3.7 定义报表变量	238
6.3.4 变量的作用域	182	8.3.8 使用报表带区事件	239
思考与练习	184	8.3.9 控制报表控件的打印	239
第 7 章 表单设计	187	8.3.10 预览结果	241
7.1 关于面向对象程序设计	187	8.3.11 打印报表	241
7.1.1 关于面向对象程序设计	187	8.4 报表设计技巧和快速报表	242
7.1.2 对象、属性、事件和方法	189	8.4.1 创建一对多报表	242
7.1.3 对象操作	191	8.4.2 运用快速报表方法生成报表的初步布局	243
7.1.4 类、子类、父类和封装	193	8.4.3 暂时更改报表的打印机设置	244
7.2 表单设计	200	8.5 标签文件的创建与使用	244
7.2.1 创建表单对象	201	8.6 向应用程序中添加报表和标签	245
7.2.2 创建数据环境	202	8.6.1 控制报表和标签的输出	246
7.2.3 向表单中添加控件	205	8.6.2 集成查询和报表	246
7.2.4 使用表单集	207	思考与练习	246
7.3 表单设计与应用	207	第 9 章 菜单设计	247
7.3.1 创建表单	207	9.1 菜单系统	247
7.3.2 保存表单	208	9.1.1 菜单系统的类型	247
7.3.3 表单与常用控件的设计	208		

9.1.2 创建菜单系统的过程	248	11.2 创建表	269
9.1.3 菜单设计器的使用	249	11.2.1 字段的数据类型和属性	269
9.2 下拉式菜单的设计	253	11.2.2 使用“表设计器”来创建表	271
9.2.1 设计自定义菜单	253	11.3 在数据表视图中输入数据	274
9.2.2 设计 SDI 菜单	256	11.3.1 输入文本类型的数据	275
9.3 快捷菜单的设计	257	11.3.2 输入“是 / 否”类型的数据	276
思考与练习	260	11.3.3 输入“日期 / 时间”类型的数据	276
第 10 章 导入和导出数据	262	11.3.4 输入 OLE 对象类型的数据	276
10.1 理解导入和导出	262	11.3.5 输入超级链接类型的数据	278
10.2 导入数据	262	11.4 导入、导出数据	278
10.2.1 可导入的文件类型	262	11.4.1 导入 Access 数据库中的表	279
10.2.2 使用“导入向导”导入数据到新表	263	11.4.2 导出 Access 数据库中的表	280
10.2.3 使用“导入向导”导入数据到现有的表	264	思考与练习	281
10.3 导出数据	264	第 12 章 SQL Server 数据库系统	282
10.3.1 导出数据的步骤	265	12.1 SQL Server 2000 的特点	282
10.3.2 导出数据命令	265	12.2 SQL Server 2000 版本	282
10.4 追加记录	265	12.3 SQL Server 2000 安装	283
10.4.1 追加记录步骤	265	12.3.1 硬件及操作系统要求	283
10.4.2 追加记录命令	266	12.3.2 安装前的 Windows 准备	283
思考与练习	266	12.3.3 安装 SQL Server 2000 SP3	285
第 11 章 Access 2000 数据库系统	267	12.4 SQL Server 2000 工具和实用程序	285
11.1 创建与打开数据库	267	思考与练习	287

第1章 数据库基础知识

数据库技术产生于20世纪60年代,是数据管理的最新技术,也是计算机科学的重要分支。近半个世纪以来,正是由于数据库技术的出现,极大地促进了计算机应用向各行各业的渗透,成为应用最广泛的领域之一。

本章将从数据管理的发展过程简要地阐明什么是数据库,以及为什么要发展数据库技术;概括地介绍数据库涉及的基本概念,包括数据模型、数据库系统的体系结构、数据库管理系统的主要功能和组成部分等,作为后面各章学习的基础。

1.1 数据库概述

1.1.1 数据管理技术的产生与发展

数据是描述事物的符号记录。对数据进行有效管理是人们正常社会生活的需求。数据管理是对数据的收集、分类、组织、编码、存储、检索和维护等一系列活动的总和。

人们借助计算机进行数据管理是近半个世纪的事情,在应用需求的推动之下,数据管理技术不断发展,这与计算机软、硬件(主要是外存)的发展以及计算机应用的深入有着密切的联系。其发展大致经过了以下几个阶段。

1. 人工管理阶段

20世纪50年代中期以前为人工管理阶段。在这一阶段,计算机主要应用于科学计算,还没有运用于其他领域;还没有出现可以直接存取数据的存储设备,只有卡片、纸带等;软件方面只有汇编语言,没有关于数据管理的软件;数据处理方式基本上是批处理。这一阶段的数据管理有以下特点:

(1) 数据不保存

由于没有直接存取数据的存储设备,计算机只用于科学计算,一般不保存数据。需要计算时,将数据与程序一同输入内存,运算后将结果输出即可。

(2) 没有专门的软件管理数据

编制程序时,程序员不但要设计程序本身,还要设计数据的存储结构、存取方式和输入输出形式等。程序和数据之间没有独立性,因而程序员负担很重。

(3) 数据不能共享

数据是面向应用的,一组数据只对应一个程序。当多个程序需要使用相同的一组数据时,只能各自定义,造成大量的重复。数据与程序之间的关系如图1-1所示。

2. 文件系统阶段

从20世纪50年代后期至60年代中期为文件系统阶段。在这一阶段,计算机应用于

科学计算和信息管理；出现了磁盘、磁鼓等可直接存取数据的存储设备；软件方面出现了操作系统和高级语言；数据处理方式除了以往的批处理还有联机实时处理方式。这一阶段的数据管理有以下特点：

(1) 数据可以长期保存

这一阶段的计算机大量用于数据处理，由于出现了可直接存取设备，因而数据能保存在外存上进行反复操作。

(2) 有专门的软件管理数据

操作系统中的文件系统就是专门管理数据的软件。文件系统将数据组织成各自独立的文件存储在磁盘上，利用“按文件名访问、按记录读写”的技术，对文件进行浏览、编辑和存储等操作。由文件系统提供存取方法实现数据与程序之间的转换，数据可以反复使用。

(3) 文件组织形式多样化

数据文件由记录组成，有索引文件、直接存取文件和链接文件等多种形式，但各文件之间相互独立，数据间的联系必须由应用程序进行构造。

(4) 数据独立性差

文件系统阶段只是实现了对数据的初级管理，未体现出数据的逻辑结构独立于数据存储中的物理结构的要求。因此，一旦数据的逻辑结构发生改变，就必须修改相应的应用程序。同样，应用程序由于使用不同的高级语言，也会导致数据文件的结构定义。可见，此阶段的数据文件只是一个没有弹性的数据集合，不能反映出数据之间的内在联系，数据结构与程序间的依赖关系并未根本改变，这促使人们研究新的数据管理技术。文件系统阶段数据与程序之间的关系如图 1-2 所示。

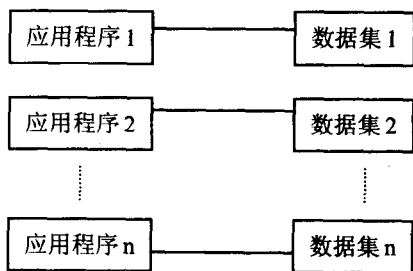


图 1-1 人工管理阶段数据与程序之间的关系

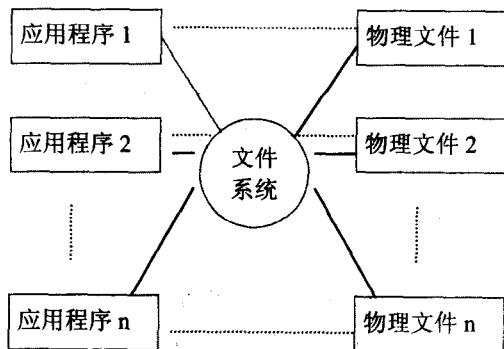


图 1-2 文件系统阶段数据与程序之间的关系

3. 数据库系统阶段

从 20 世纪 70 年代开始进入数据库系统阶段。数据库系统克服了文件系统的不足，提供了更为有效的数据管理方法。随着计算机的普及与发展，人们共享数据的要求越来越强烈。为满足多用户、多应用共享数据的需求，数据库技术应运而生，出现了统一管理数据的软件——数据库管理系统 (DBMS)。与文件系统相比，数据库系统有以下特点：

(1) 数据结构化

数据库系统的根本特征是数据结构化。数据由 DBMS 按一定结构组织存储在计算机

中, 并进行统一管理, 如图 1-3 所示。数据结构有物理结构和逻辑结构之分, DBMS 既要考虑数据本身的定义, 还要考虑数据之间以及文件之间的相互联系。

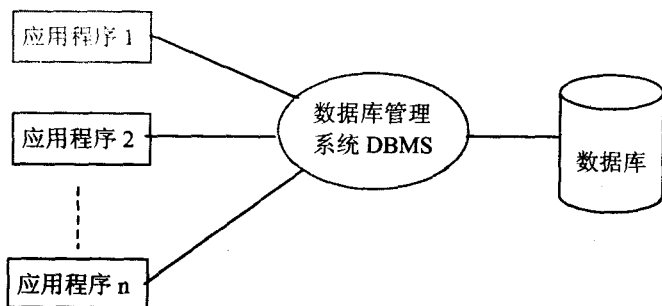


图 1-3 数据库系统阶段数据与程序之间的关系

(2) 数据独立性高

数据独立性指的是应用程序与磁盘存储数据之间的相互独立。在磁盘中存储的数据由 DBMS 进行管理, 用户一般不需要了解其存储结构, 只是处理数据的逻辑结构。这样, 当数据在存储设备上的物理存储结构发生改变时, 由 DBMS 处理这种变化, 而应用程序保持不变, 这称为数据的物理独立性。此外, DBMS 还可以提供逻辑独立性功能, 当数据的逻辑结构改变时, 用户程序不变。这一特性大大减少了程序员编制和维护应用程序的工作量。

(3) 数据共享性高, 冗余度低

数据库系统改变了文件系统中应用程序面对自己专用数据文件的缺陷, 从全局角度描述数据, 使数据面向整体而不是仅仅面对某个应用, 从而大大减少了数据冗余。由于数据由 DBMS 进行统一管理, 有灵活的处理方式, 可以适应不同用户的要求, 因而实现了多个应用程序对数据的共享。

(4) 统一的数据控制

为了适应数据共享的环境, DBMS 还提供了数据控制功能, 包括数据安全性、完整性、并发控制和数据库恢复控制。

① 数据安全性

数据的安全性是指保护数据, 防止非法使用所造成的数据丢失、泄露或破坏, 保证用户使用数据库的安全与机密。如: 用口令检查用户的合法身份。

② 数据完整性

数据的完整性指的是数据的正确性、有效性和兼容性, 即数据控制在有效范围内, 或要求数据之间满足一定的关系, 使数据库中始终包含正确的数据。用户可设计完整性规则保证数据的正确性。

③ 数据库并发控制

当多个用户并发访问数据库时, 可能会发生相互干扰。为解决冲突, 保护数据库, 数据库系统应采取措施对多用户的并发操作加以控制、协调。

④ 数据库恢复

计算机系统的软件、硬件故障, 操作员的失误及故意破坏, 病毒等均可能影响数据库中数据的正确性, 使部分甚至全部数据丢失。DBMS 必须具有将数据库从错误状态恢复到

某一已知的正确状态的功能，这就是数据库的恢复功能。

4. 高级数据库技术发展阶段

从 20 世纪 80 年代开始，相继出现了分布式数据库系统和面向对象数据库系统，标志着进入高级数据库技术发展阶段。

(1) 分布式数据库系统

随着计算机网络和远程通信的发展，出现了分布式数据库系统。分布式数据库由一组数据组成，这些数据物理上分布于网络的不同结点，而逻辑上属于同一个系统。网络中的每个结点具有独立的处理能力，可以实现局部应用，同时，各结点也可以通过网络通信相互连接，实现全局应用。

(2) 面向对象数据库系统

面向对象数据库系统是面向对象的程序设计与数据库相结合的产物，它具有面向对象的封装性（将数据本身和数据操作定义在一起）和继承性（子类数据继承父类数据的特点），提高了软件的应用效率，降低了系统开发工作量。面向对象数据模型可以更真实地描述现实世界的数据结构，更好地表达数据之间的联系。

1.1.2 数据库基本概念

与数据库技术密切相关的有四个概念，它们是数据、数据库、数据库管理系统和数据库系统。下面介绍这四个基本概念。

1. 数据 (Data)

数据是描述事物的符号记录。在日常生活中人们用自然语言描述事物，在计算机中，为了存储和处理这些事物，就要抽出对这些事物感兴趣的特征组成一个记录来描述，这些描述符号被人们赋予特定的语义，所以它们就具有了刻画事物、传递信息的功能。

数据处理领域中数据概念比科学技术中的数据概念要广，它不仅包含数字符号（数值型数据），更可包含文字、图像和其他特殊符号（非数值型）。

数据与其语义是不可分的，对于一条记录，了解其语义的人即可得到相应的信息，而不了解语义的则无法理解其中含义。如：在高校中可用“1、2、3、4”（或 A、B、C、D）代表职称信息（教授、副教授，讲师，助教），也可代表不同年级的学生信息。由此可见，数据的形式本身并不能完全表达其内容，须经过语义解释。

2. 数据库 (DataBase, DB)

数据库是长期存储在计算机内、有组织的、可共享的数据集合。数据库中的数据按一定的数据模型组织、描述和存储，具有较小的冗余度、较高的数据独立性和易扩充性，并为各种用户所共享。

3. 数据库管理系统 (DataBase Management System, DBMS)

在收集抽取所需的大量数据后，由软件系统 DBMS 将这些数据进行科学组织，存储于数据库中，并进行统一管理。DBMS 是位于用户与操作系统之间的一层数据管理软件。数据库在建立、运用和维护时由 DBMS 进行统一管理和控制，使用户方便地定义和操纵数

据, 保护数据的安全性、完整性、多用户对数据的并发使用以及发生故障后的系统恢复。

4. 数据库系统 (DataBase System, DBS)

数据库系统是指在计算机系统中引入数据库后的系统构成, 一般由数据库、数据库管理系统、应用系统、数据库管理员 (DBA) 和一般用户构成, 如图 1-4 所示。

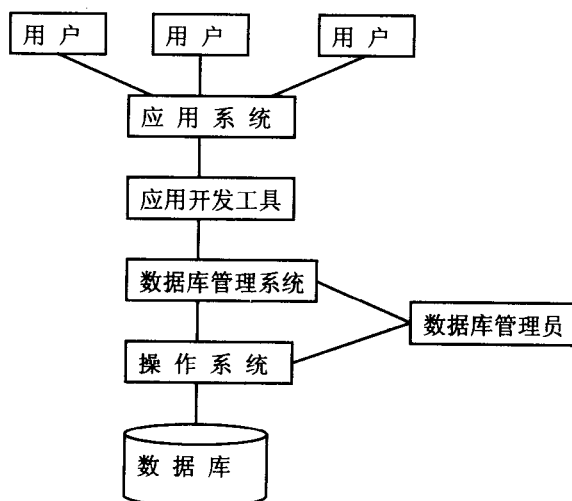


图 1-4 数据库系统的组成

数据库的建立、使用和维护仅靠一个 DBMS 远远不够, 还需有专业人员, 即 DBA (DataBase Administrator) 负责为存取数据库的用户授权, 协调监督用户对数据库和 DBMS 的使用, 同时也负责系统安全性保护和系统性能的监督和改善。大多数情况下, DBA 即为数据库的设计者。

1.2 数据库系统结构

1.2.1 数据库系统的三级模式

数据库的体系结构由三级构成: 内部级、概念级和外部级。虽然数据库管理系统的产品多种多样, 但大多数系统的总体结构采用三级结构的形式。外部级最接近用户, 是用户看到的数据特征; 概念级是关于整体数据的定义, 即全局的数据特征; 内部级则最接近存储设备, 定义数据的存储方式。

数据库的三级体系结构是数据的三个抽象级别。三个级别之间的转换和联系由 DBMS 完成, 这就是两层映像: 外模式/模式映像与模式/内模式映像, 如图 1-5 所示。

1. 模式 (也称逻辑模式或概念模式)

- 模式是数据库中全体数据的逻辑结构和特征的描述, 是所有用户的公共数据视图 (注: 从某个角度看到的数据特性称为“数据视图”)。它是数据库系统模式结构的

中间层次, 不涉及数据的物理存储细节和硬件环境, 与具体的应用程序、开发工具及程序设计语言无关。

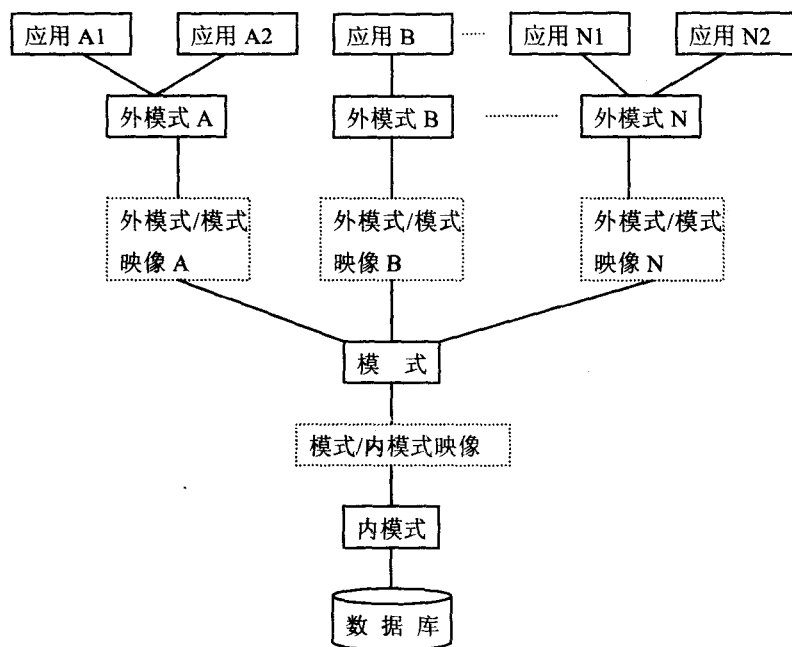


图 1-5 数据库系统结构

实际上, 模式是数据库数据在逻辑级的视图。一个数据库只有一个模式, 模式以某种数据模型为基础, 综合考虑用户要求, 并有机地结合成一个逻辑整体。定义模式时不仅要定义数据的逻辑结构 (记录由哪些数据项组成, 数据项的名称、类型和取值范围等), 还要定义与数据有关的安全性、完整性要求, 定义这些数据之间的联系。

2. 外模式 (也称子模式或用户模式)

外模式是数据库用户看见和使用的局部数据的逻辑结构和特征的描述, 是数据库用户的数据视图与某一应用有关的数据的逻辑表示。

外模式通常是模式的子集, 一个数据库可以有多个外模式。由于它是各用户的数据视图, 若不同用户在需求、看待数据的方式、保密程度等方面存在差异, 则其外模式描述也可不同。即使对模式中的同一数据, 在外模式中的结构、类型和长度等均可不同, 一个外模式可以为某一用户的多个应用系统使用, 但一个应用程序只能使用一个外模式。

外模式是保证数据库安全的重要手段, 每个用户只能看见和访问所对应的外模式中的数据, 其余数据是不可见的。

3. 内模式 (也称存储模式)

内模式是数据物理结构和存储结构的描述, 是数据在数据库内部的表示方式。例如: 记录的存储方式、索引的组织方式、数据是否压缩存储、是否加密; 数据的存储记录结构有何规定等。一个数据库系统中只有一个内模式。

1.2.2 数据库的二级映像功能与数据独立性

数据库系统的三级模式是对数据的三个抽象级别,它把数据的具体组织留给 DBMS 管理,使用户能够抽象地处理数据,而不必关心数据在计算机中的具体表示方式与存储方式。为了在内部实现这三个抽象层次的联系与转换,数据库系统在这三个模式之间提供了两层映像:外模式/模式映像、模式/内模式映像。正是这两层映像保证了数据库系统中的数据能具有较高的逻辑独立性和物理独立性。

1. 外模式/模式映像

模式描述的是数据的全局逻辑结构。外模式描述的是数据的局部逻辑结构。对应于同一个模式可以有任意多个外模式。对于每个外模式,数据库系统都有一个外模式/模式映像,它定义了外模式与模式之间的对应关系。这些映像定义通常包含在各自外模式的描述中。当模式改变时(如增加新的数据类型、数据项和关系等),由数据库管理员对各个外模式/模式的映像做相应改变,可以使外模式保持不变,从而应用程序不必修改,保证了数据的逻辑独立性。

2. 模式/内模式映像

数据库中只有一个模式,也只有一个内模式,因此,该映像是惟一的。它定义了数据全局逻辑结构与存储结构之间的对应关系,如说明逻辑记录 and 字段在内部是如何表示的。该映像定义通常包含在模式描述中。当数据库的存储结构改变了,由数据库管理员(DBA)对模式/内模式映像做相应改变,可以使模式保持不变,从而保证了数据的物理独立性。

1.3 数据模型

建立数据库系统的目的是为了应用,如何根据现实世界的要求构造数据模型是使用数据库系统的关键。数据模型决定数据库系统的结构、数据定义语言、操纵语言、数据库设计方法及数据库管理系统软件的设计与实现。本节简单介绍有关的概念。

数据模型:一组描述数据库的概念。这些概念精确描述数据、数据之间的联系、数据的语义和完整性约束,有的模型还包括一个操作集合,这些操作用来说明对数据库的存取和更新。

设计数据模型时,应满足三方面的要求:一是能真实地模拟现实世界;二是易于理解;三是便于在计算机上实现。目前已有很多数据模型。

1.3.1 数据模型的要素

一般地,数据模型均由以下三部分组成:数据结构、数据操作和数据约束条件。

1. 数据结构:用于描述系统的静态特性

数据结构是所研究的对象类型的集合,它们是数据库的组成部分,包括两类:一类是与数据类型、内容和性质有关的对象,如关系型中的域、属性和关系等;另一类是与数据

之间联系有关的对象,如网状型中的系型。

数据结构是刻画一个数据模型性质最重要的方面。因此,在数据库系统中,通常按数据结构的类型来命名数据模型,如层次结构、网状结构、关系结构的数据模型分别命名为层次模型、网状模型、关系模型。

2. 数据操作:用于描述系统的动态特性

数据操作指对数据库中各种对象(型)的实例(值)允许执行的操作的集合,包括操作及其规则。数据库主要有检索和更新两大类操作。数据模型必须定义这些操作的含义、符号、规则及实现的语言。

3. 数据的约束条件:一组完整性规则的集合

完整性规则是给定的数据模型中数据及其联系所具有的制约和储存规则,用以限定符合数据模型的数据库状态及状态变化,保证数据的正确性。

数据模型还应提供定义完整性约束条件的机制,以反映具体应用所涉及的数据必须遵守的特定的语义约束条件。

1.3.2 实体联系模型

实体联系模型(ER 模型)是直接将现实世界的客观对象抽象出实体类型和实体之间的联系,用实体联系图(ER 图)描述。ER 模型是数据库设计者与用户之间交流的语言,它能准确表达应用中的语义,便于用户理解,同时,它不依赖于具体的计算机系统,只有将 ER 模型转换成计算机上某一 DBMS 支持的数据模型后才能在计算机上运行。

1. 实体联系模型的基本概念

(1) 实体(entity)

客观存在并可以相互区别的事物叫实体,可以是人、事、物,也可以是事物本身或事物间的联系,如一名雇员,一种产品,一份订单。

(2) 属性(attribute)

实体有许多特征,某个特征被称为属性,一个实体可以由多个属性来进行刻画。如雇员号,姓名,头衔,出生日期等都能作为属性来刻画雇员实体。

(3) 码(key)

又叫做“键”,能惟一标识实体的属性(集)叫“码”。如雇员号是雇员的码,在无姓名相同者时,姓名也可以是码。

(4) 域(domain)

属性的取值范围称为该属性的域。

(5) 实体型(entity type)

有相同属性的实体必有共同的特性。用实体名及其属性集合来抽象和刻画同类实体,即为实体型,如雇员(雇员号,姓名,头衔,出生日期……)

(6) 实体集(entity set)

同型实体的集合为实体集,如全体雇员是一个实体集。