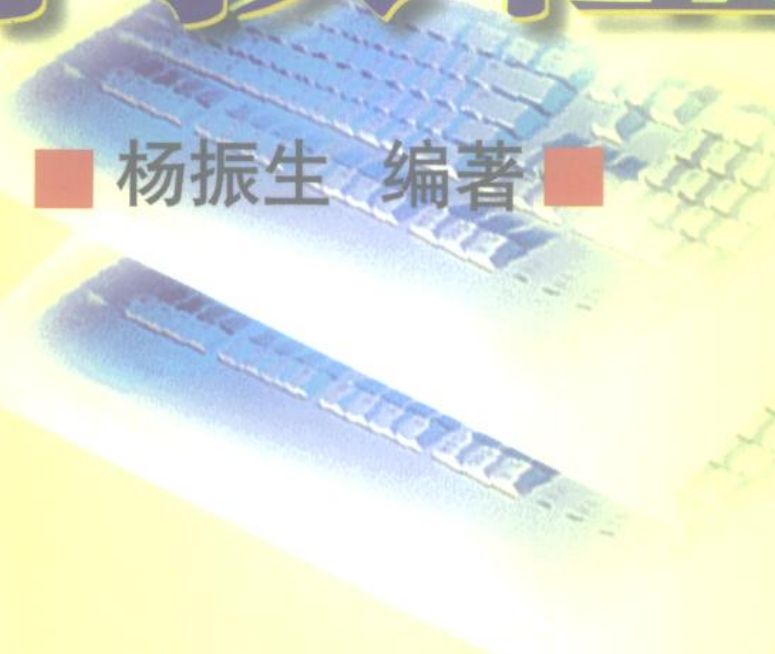


FoxBASE⁺实用教程升级版本

FoxPro 2.6
for **Windows**

实用教程

■ 杨振生 编著 ■



中国科学技术大学出版社

TP311.13

Y31-2

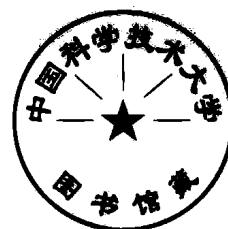
428268

FoxBASE⁺ 实用教程升级版本

FoxPro 2.6 for Windows

实 用 教 程

杨振生 编著



中国科学技术大学出版社

1999·合肥

图书在版编目(CIP)数据

FoxPro 2.6 for Windows 实用教程/杨振生编著. —合肥:中国科学技术大学出版社,1999.2
ISBN 7-312-01056-3

I . F… II . 杨… III . 关系数据库-数据库管理系统,FoxPro2.6-教材 IV . TP311.13

中国版本图书馆 CIP 数据核字 (1999) 第 00245 号

中国科学技术大学出版社出版发行
(安徽省合肥市金寨路 96 号,邮编:230026)
合肥晓星印刷厂印刷
全国新华书店经销

开本:787×1092/16 印张:23.5 字数:570 千

1999 年 2 月第 1 版 1999 年 2 月第 1 次印刷

印数:1—6000 册

ISBN 7-312-01056-3/TP·221 定价:23.00 元

内 容 提 要

本书从关系数据库的基本原理出发,系统地介绍了 FoxPro 2.6 for Windows 的命令和函数的语法、功能及其程序设计方法与技巧。其中包括数据库的各种基本操作、多种生成器的使用方法、SQL 和 RQBE 查询语言以及窗口、菜单、事件驱动、按钮、网络等高级程序设计技术。

该内容丰富、通俗易懂、注重实用。书中含有大量例题,且全部上机通过。每章后附有习题,供读者演练和实习。它可作为大专院校“数据库技术”课程的教材,也适宜计算机培训班和各类管理人员使用,还可作为科技人员的自学读物。

前 言

如今数据库技术已广泛应用于社会生活的各个领域。当今的时代是信息“爆炸”的时代,数据处理和信息管理已成为计算机应用的主流领域,而数据库技术是这个领域的核心。

目前在我国应用范围最广的是微机关系数据库管理系统 dBASE 和 FoxBASE。80 年代初, AshtonTate 公司推出 dBASE 产品,其后版本相继升级,如 dBASEⅢ、dBASEⅢplus 和 dBASEⅣ。dBASE 产品具有功能强、通俗易懂、易学易用和使用方便等优点,受到广大用户的欢迎。但人们很快发现,dBASE 运行速度慢,版本兼容性差,性能较低。因此 Fox 公司于 1984 年推出与 dBASE 全兼容的 FoxBASE,其速度和性能大大提高,特别是多用户产品 FoxBASE+ 的推出,轰动了整个软件市场,成为广大用户的首选产品。从 1987 年起,该公司又相继推出 FoxBASE+ 2.0、FoxBASE+ 2.1、FoxPro 1.0 以及 FoxPro 2.0 等产品。Fox 公司并入 Microsoft 公司后,又陆续推出 FoxPro 2.5 和 FoxPro 2.6。FoxPro 2.6 有两种版本:FoxPro 2.6 for DOS 和 FoxPro 2.6 for Windows。

FoxPro 2.6 for Windows 与已往的版本相比,运行速度更快、功能更强。它拥有优良的 Windows 运行环境、友好的图形窗口界面和丰富的实用工具,并且新增和扩充了许多功能。它采用了 Rushmore 最优化技术,能与 Windows 应用程序实现动态数据交换和对象的嵌入与连接等。因此,倍受广大用户的推崇和青睐。

本书共分十章,第一章介绍数据库的基本概念;第二章介绍 FoxPro 2.6 的初步知识;第三章介绍 FoxPro 2.6 的窗口界面;第四章介绍 FoxPro 的基本命令与操作,这是本书的重点内容之一;第五章介绍 FoxPro 程序设计基础,重点是结构化程序设计方法;第六章介绍 FoxPro 系统的各类内部函数;第七章介绍 FoxPro 的高级命令与操作,也是本书的重点之一;第八章介绍结构查询语言 SQL 和 RQBE;第九章介绍报表与标签的设计和生成;第十章介绍 FoxPro 高级程序设计技术,其中包括窗口程序设计、菜单程序设计、按钮程序设计、事件驱动程序设计和网络程序设计等。第五章和第十章也是本书的重点。

本书可针对不同专业、不同层次的要求,对内容适当取舍。在学习过程中,要注重和强化上机实践环节,以便收到更好效果。

本书在编写过程中,曾得到程玉宝、汪永益等同志的热情支持,在此深表谢意。

书中有不妥之处,敬请专家和读者指正。

作 者

1998 年 7 月于合肥

目 录

前 言	(I)
第一章 数据库基本概念	(1)
1.1 数据、信息和数据处理	(1)
1.2 数据库和数据库系统	(3)
1.3 数据库管理系统	(5)
1.4 数据模型	(7)
1.5 关系的定义和性质	(11)
1.6 关系代数	(14)
习 题	(18)
第二章 FoxPro 2.6 的初步知识	(20)
2.1 FoxPro 的主要特点	(20)
2.2 FoxPro 的性能指标	(21)
2.3 FoxPro 的运行环境	(23)
2.4 FoxPro 系统的安装、启动和退出	(23)
2.5 FoxPro 的语言基础	(27)
2.5.1 数据类型	(28)
2.5.2 常量与变量	(28)
2.5.3 常用函数	(29)
2.5.4 运算符与表达式	(31)
2.6 文件类型	(32)
2.7 文本编辑器	(33)
2.8 FoxPro 命令的语法规则	(36)
习 题	(37)
第三章 熟悉 FoxPro 的窗口界面	(38)
3.1 FoxPro 的菜单结构	(38)
3.2 浏览 FoxPro 的系统菜单	(39)
3.3 FoxPro 的对话框	(44)
3.4 使用 FoxPro 窗口	(46)
3.5 Command 窗口	(48)
3.6 文件管理器	(48)
3.7 在线帮助	(52)
习 题	(54)
第四章 FoxPro 的基本命令与操作	(55)
4.1 数据库文件的建立	(55)
4.2 数据库文件的打开与关闭	(58)

4.3	数据库记录的定位	(61)
4.4	数据库记录的显示	(64)
4.5	数据库记录的添加、插入与筛选	(66)
4.6	数据库记录的删除	(68)
4.7	数据库记录的修改	(70)
4.8	数据库结构的显示、修改和拷贝	(73)
4.9	数据库记录的浏览	(80)
4.10	使用 BROWSE 命令	(87)
4.11	内存变量	(92)
4.12	统计与计算	(99)
4.13	数据的输入与输出	(105)
	习 题	(108)
第五章	FoxPro 程序设计基础	(113)
5.1	应用程序系统的开发过程	(113)
5.2	程序文件的建立与执行	(114)
5.3	顺序结构程序设计	(115)
5.4	分支结构程序设计	(124)
5.5	循环结构程序设计	(131)
5.6	函数与过程的程序设计	(140)
5.7	函数与过程的递归调用	(154)
5.8	数组程序设计	(157)
	习 题	(163)
第六章	FoxPro 系统的内部函数	(166)
6.1	数值函数	(166)
6.2	字符函数	(173)
6.3	日期和时间函数	(177)
6.4	类型转换函数	(178)
6.5	测试函数	(180)
6.6	窗口函数	(184)
6.7	环境函数	(185)
	习 题	(185)
第七章	FoxPro 的高级命令与操作	(188)
7.1	数据库记录的排序	(188)
7.2	数据库的索引	(190)
7.3	数据库记录的分类汇总	(200)
7.4	数据库记录的快速检索	(203)
7.5	多个数据库之间的联合操作	(205)
7.6	VIEW 窗口与多库间的关联	(214)
	习 题	(220)

第八章 结构查询语言 SQL 与 RQBE	(224)
8.1 SELECT 命令	(224)
8.2 基本查询	(226)
8.2.1 简单查询	(226)
8.2.2 SQL 函数的使用方法	(229)
8.2.3 谓词 BETWEEN、IN 和 LIKE 的用法	(229)
8.2.4 排序查询	(232)
8.2.5 分组查询	(233)
8.2.6 合并查询	(235)
8.3 复杂查询	(236)
8.3.1 多表连接查询	(237)
8.3.2 嵌套查询	(240)
8.3.3 关键字 ANY 和 ALL, 谓词 IN 和 EXISTS 的用法	(241)
8.4 关于 RQBE	(243)
8.4.1 RQBE 的进入与退出	(244)
8.4.2 RQBE 窗口	(244)
8.4.3 RQBE 菜单	(245)
8.5 RQBE 的查询方法	(247)
8.5.1 基本查询	(247)
8.5.2 多库查询	(250)
8.6 输出方向	(251)
习 题	(252)
第九章 报表与标签的生成	(254)
9.1 报表的产生过程	(254)
9.2 报表设计	(256)
9.3 操作报表对象	(258)
9.4 对象的对话框	(259)
9.5 报表中的字段表达式	(262)
9.6 报表菜单	(265)
9.7 对象菜单	(271)
9.8 标签的生成	(273)
9.9 报表设计示范	(274)
习 题	(280)
第十章 FoxPro 高级程序设计	(282)
10.1 窗口程序设计	(282)
10.1.1 窗口的定义	(282)
10.1.2 窗口的基本操作	(283)
10.1.3 窗口的其它操作	(285)
10.1.4 窗口的测试	(287)

10.2 菜单程序设计	(288)
10.2.1 菜单的结构与属性	(289)
10.2.2 条形菜单设计	(290)
10.2.3 上弹式菜单设计	(294)
10.2.4 下拉式菜单设计	(299)
10.3 事件驱动程序设计	(302)
10.3.1 事件和事件驱动程序	(302)
10.3.2 键控函数及其程序设计	(303)
10.3.3 键控命令及其程序设计	(307)
10.3.4 错误捕捉与处理	(311)
10.4 按钮控制程序设计	(313)
10.4.1 下压按钮程序设计	(313)
10.4.2 单选按钮程序设计	(315)
10.4.3 复选按钮程序设计	(317)
10.4.4 无形按钮程序设计	(319)
10.4.5 滚数器程序设计	(321)
10.5 网络环境下程序设计	(323)
10.5.1 网络环境下编程的新特点	(323)
10.5.2 文件和记录的加锁与解锁	(324)
10.5.3 文件和记录的自动加锁与解锁	(328)
10.5.4 反复加锁与锁状态检测	(329)
10.5.5 死锁及其预防	(330)
10.6 低级文件输入输出	(332)
10.6.1 低级文件的建立、打开与关闭	(333)
10.6.2 低级文件的输出	(334)
10.6.3 低级文件的输入	(337)
10.6.4 低级文件相关函数	(338)
10.7 其它	(339)
习 题	(341)
附录 A FoxPro 命令汇总表	(343)
附录 B FoxPro 函数汇总表	(353)
附录 C INKEY()的按键返回值	(361)
附录 D ON KEY LABEL 命令识别名称	(363)
附录 E ON KEY 按键代码	(364)
参考文献	(365)

第一章 数据库基本概念

70年代后期,关系数据库理论开始走向成熟。关系数据库技术得到了飞速发展,并取得了卓越成就。关系数据库管理系统的版本不断升级,功能不断扩充和完善,其中 FoxPro 2.6 是最典型的代表,它功能强大、操作方便、效率高,深受广大用户的欢迎和青睐。目前 FoxPro 在数据处理和信息管理领域已成为当今世界的主流。不过,在深入学习 FoxPro 之前,读者必须了解一些关系数据库的基本概念,这正是本章的目的。

1.1 数据、信息和数据处理

一、数据

数据是对客观事物特征的一种符号化表示,用来记录从观察或测量中收集到的基本事实。所采用的符号完全是一种人为的规定,例如,记录数值,既可以用十进制数,也可以用二进制数;记录人的姓名,既可以用中文,也可以用外文。“数据”具有广泛的含义,既有量化的表示,也有非量化表示的。因此,数据有类型和值之分。例如,室内温度可以量化表示,这是数值型数据;“下雨”或“刮风”等可以用文字或图形符号非量化地表示,这是字符型数据。

二、信息

信息和数据密切相关,两者既有联系,又有区别。信息是经过加工的数据,就是说,它是从数据海洋中经过正确地提取和适当地组织而构成的某种数据流。

信息是资源,可以被利用,并影响人们的行为和动作,能为某一特定目的提供决策依据。数据反映信息,而信息必须依靠数据来表达。

现实世界是一个充满信息的世界,人们非常关心对其有用的信息,并以此决定自己的行为 and 决策。例如,人们根据汛情通报制订防汛措施;根据天气预报安排生产和生活;工厂根据市场变化进行产品结构的调整等。

三、数据处理

所谓数据处理泛指对原始数据进行的收集、整理、存储、分类、排序、检索、维护、加工、统计和传输等一系列活动的总称。

当今时代是一个信息“爆炸”的时代,宏大的数据流单靠手工和简单的工具已无法应付,更远远跟不上社会的实际需要。计算机的出现使数据处理发生了划时代的变革,而数据库技术的发展,使数据处理步入了一个崭新的阶段。数据处理的核心是数据管理技术,它的发展大致经历了三个阶段:自由管理方式、文件管理方式和数据库系统方式。

(一)自由管理方式

自由管理方式又称手工管理方式。早期的电子计算机,没有基本的软件支持,用户直接在裸机上作业,虽然有绝对的自由,但必须确定和记住数据存放的物理地址,使得用户程序高度依赖于数据的存储地址。这种管理方式,迫使用户程序直接与物理地址打交道,使用户的负担极重。用这种方式管理数据既不灵活,也不安全,编程效率极低。

(二)文件管理方式

文件管理方式是有关数据组织成文件。这种文件脱离程序独立存在,并容许对文件命名,而应用程序通过文件名来存取文件中的数据。

这些数据文件,由一个专门的软件系统——文件管理系统实施统一管理。文件的组织方式是按照统一的规则和方法来组织和存取的。文件管理系统是一个独立的系统软件,它是应用程序与数据文件之间的一个接口,应用程序通过文件管理系统建立和存取文件。如图 1-1 所示。

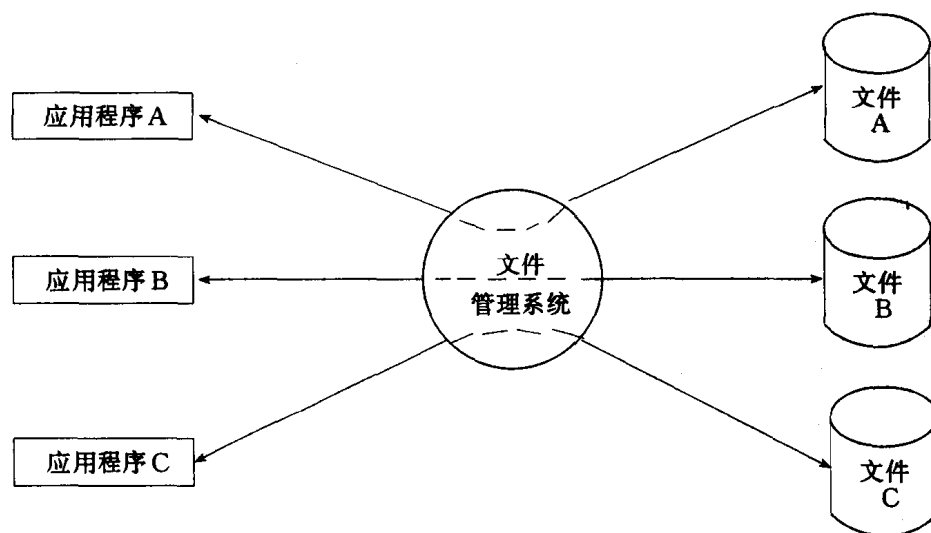


图 1-1 文件管理方式示意图

文件管理方式有以下缺点:

1. 尽管数据以文件方式独立存放,但程序与数据紧密相关,一旦数据文件离开了使用它的应用程序,便失去了存在的价值。
2. 由于不同应用程序各自建立相应的数据文件,造成了数据冗余,使空间利用率大为降低。
3. 由于同一数据存放在不同的数据文件中,很容易造成数据的不一致性。
4. 文件管理方式不能反映信息之间的联系。

为了克服文件管理方式的上述弱点,又产生了数据库系统方式。

(三)数据库系统方式

数据库系统方式与文件管理方式不同,其数据组织是面向整个系统,即用整体观点规划数据,形成一个数据中心,构成一个数据仓库,库中的数据能满足所有用户的不同要求,供不同用户共享。这时,应用程序不再与一个孤立的数据文件相对应,而是取自整体数据集的某个子集,作为逻辑文件与应用程序相对应,通过一个系统软件——数据库管理系统(DBMS)实现逻辑文件与物理数据之间的映射。

1.2 数据库和数据库系统

一、什么是数据库

数据库是以一定的组织方式存贮在一起的、能为多个用户所共享的、与应用程序彼此独立的相互关联的数据集合。如图 1-2 所示。

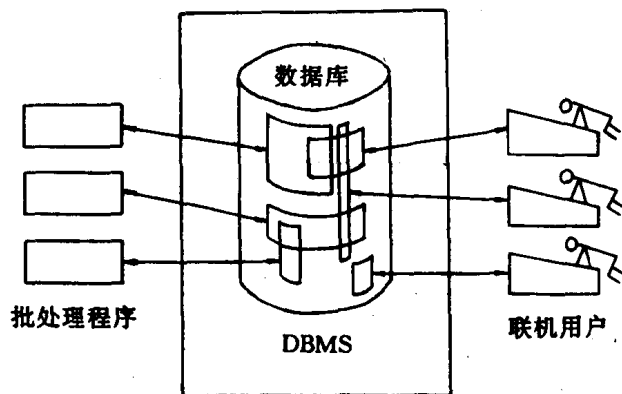


图 1-2 数据库系统简图

数据库具有以下特点：

1. 数据的共享性:数据库中的数据能为多个用户服务;
2. 数据的独立性:用户的应用程序与数据的逻辑组织和数据的物理存贮方式无关;
3. 数据的完整性:数据库中的数据在维护活动中始终保持正确性;
4. 数据库中的冗余数据少。

二、数据库的组织结构

从数据管理的角度来看,与数据库打交道的有三类人员:用户、数据库管理员(DBA)和系统程序员。由于他们对数据库的认识、理解和接触的范围不同,形成了各自的数据库视图。所谓视图是指观察和理解数据的方法与范围。根据各类人员与数据库的不同关系,我们把视图分为三种:用户的外部视图、DBA 的概念视图和系统程序员的内部视图。

因此,数据库的基本结构是一种分层的三级组织结构,即由用户级、概念级和物理级组成。其相互间的关系如图 1-3 所示。

概念级数据库也称为数据库的概念模式,它是由数据库设计者综合所有用户数据,按照统一的观点构造的全局逻辑结构,它是用模式描述语言来描述的、由多种概念记录组成的数据库。概念级数据库是 DBA 所看到的数据库,即 DBA 的数据视图。

用户级数据库也称为数据库的外模式,它是用户所看到和理解的数据库。外模式是从概念模式导出的子模式,用户可以通过子模式描述语言来描述用户级数据库的记录,还可以利用数据操纵语言对这些记录进行操作。

物理数据库也称数据库的内模式或存贮模式。它是由系统程序员设计和组织的、是系统

程序员所看到的和所理解的数据库。物理级数据库就是真正存放在外存贮器上的数据,它描述了数据在存贮介质上的安排与存放方式,实际上是许多物理文件的集合。

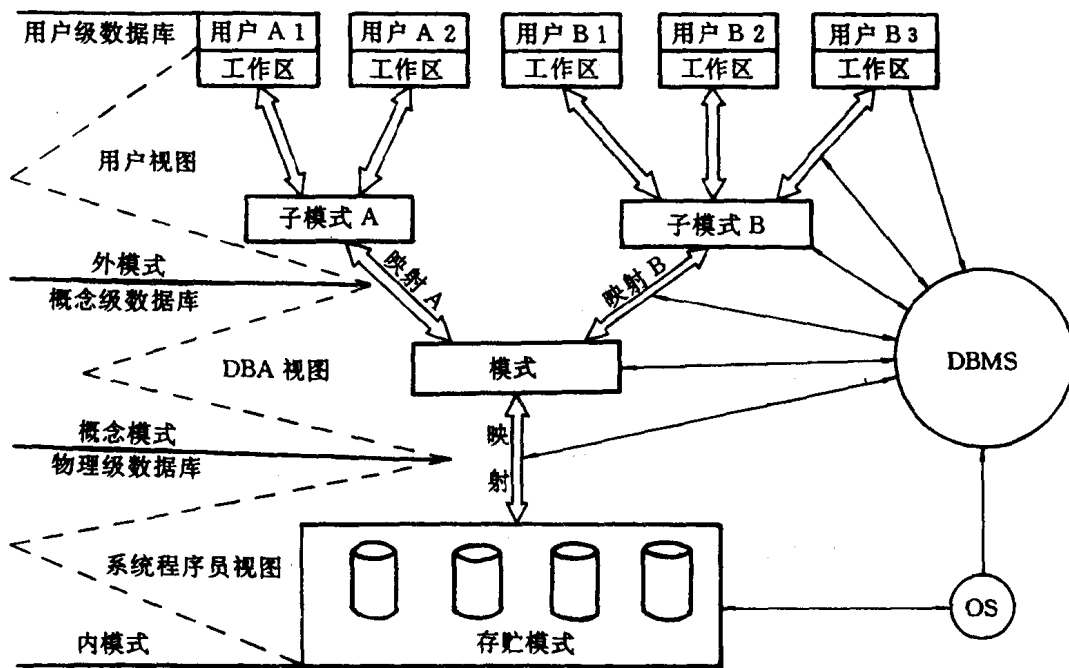


图 1-3 数据库组织结构示意图

从图 1-3 可以看出,三级数据库结构之间的联系是通过两级映射实现的:

模式/存贮模式之间的映射表达了概念数据库与物理数据库之间的对应关系。

子模式/模式之间的映射表达了用户数据库与概念数据库之间的对应关系。

而数据库管理系统(DBMS)的主要工作之一就是完成三级数据库之间的转换,把用户对数据库的操作转化到物理级去执行。

三、什么是数据库系统

数据库系统是指引进数据库技术后的整个计算机系统。它包括四部分:数据、硬件、软件 and 用户。

1. 数据:是数据库系统的管理对象。
2. 硬件:是数据库系统的物理支撑,其中包括 CPU、内存、磁盘及 I/O 设备。
3. 软件:系统软件和应用软件。

系统软件包括操作系统 OS 和数据库管理系统 DBMS。DBMS 是数据库系统中最重要核心软件,它负责数据库的运行控制和管理。

应用软件是在 DBMS 的基础上由用户根据实际需要自行开发的应用程序。

4. 用户:批处理用户、联机用户和系统用户。

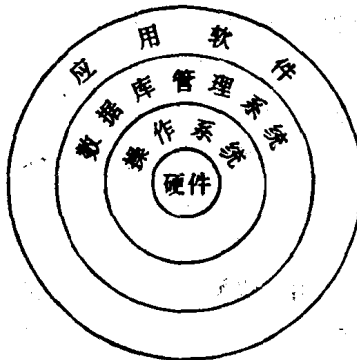


图 1-4 数据库系统的基本结构

批处理用户是指专业程序员,他们能使用程序设计语言编制程序对数据库进行存取操作,并作某种应用处理。联机用户多为非计算机专业人员,他们通常只需要从数据库中获取有关综合性的信息,如统计、查询等,一般使用命令语言,操作相当简单。系统用户对数据库进行整体性的维护,其操作对象是整个数据库。这一特殊用户称为数据库管理员,简称 DBA,他是数据库的责任维护者。

一个数据库系统的基本结构如图 1-4 所示。

数据库技术就是研究在计算机环境下如何合理的组织存放数据、有效地管理数据和高效地进行数据处理。

1.3 数据库管理系统

数据库管理系统的基本功能就是有效地实现数据库三级之间的转换,它的主要功能可以概括为以下几个方面:

1. 定义数据库。DBMS 提供了描述模式、子模式和物理模式的语言以及将各种模式由源形式转换为目标形式的处理能力。定义数据库是建立数据库的第一步工作,这一步的完成将为数据库建立一个“框架”。

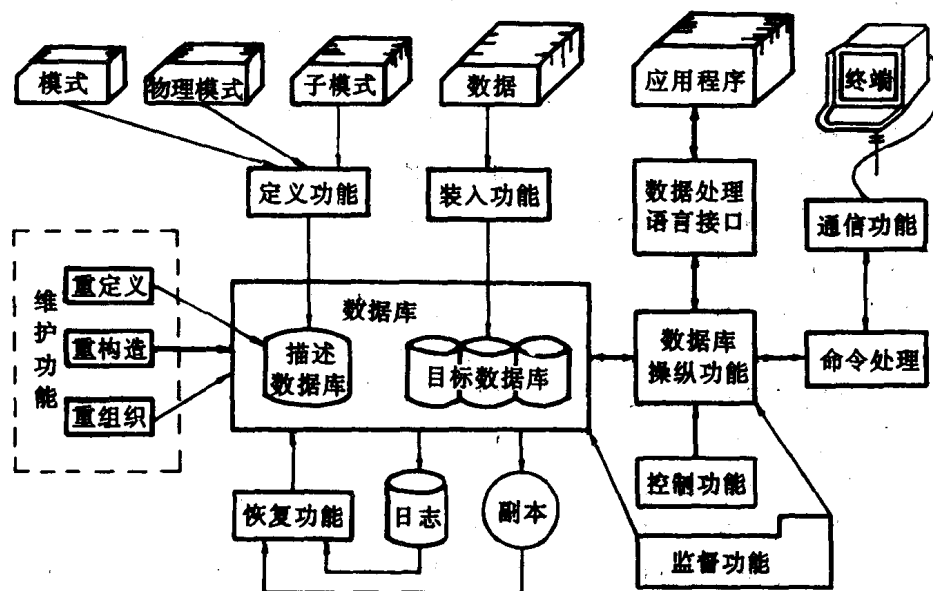


图 1-5 DBMS 功能示意图

2. 数据装入功能。定义数据库还不是数据库本身的建立,因为它只有“框架”,而没有实际内容。为了获得一个实际的数据库,还必须进行数据装入。

3. 数据库操纵功能。主要是接收、分析和执行数据库用户的存取请求,通常包括检索、插入、删除和更新等。因此,DBMS 提供了数据操纵语言及其处理程序。

4. 数据库控制功能。包括控制整个数据库系统运行;控制用户的并发性访问;执行对数据的安全、保密、完整性检验等。

5. 数据库维护功能。数据库维护是系统的例行工作,以保证数据库系统的正常运行,向用户提供有效的数据服务。维护的主要内容是:数据库重定义,数据库重构造,数据库重组织等。

6. 数据库恢复功能。主要包括复制数据库副本的能力、建立系统运行日志的能力,以及恢复和重运行数据库的能力。

7. 数据通信功能。这一功能用来保证系统的联机用户通过远程终端来存取数据库。

8. 数据库监督功能。在数据库系统运行过程中,DBMS 能对各种变化情况进行监督,并随时进行分析,必要时应采取相应措施,以保证系统正常运行。

以上我们只列举了 DBMS 必须具备的一些主要功能,除此之外,DBMS 还应具备许多其他功能。图 1-5 给出了 DBMS 功能示意图。

DBMS 通常由三部分组成:

一、语言翻译处理程序

1. 模式 DDL(Data Description Language 数据描述语言)翻译程序。

为了建立数据库,首先必须正确描述数据与数据之间的关系。模式 DDL 是用来定义数据库全局逻辑结构的,它包括所有数据元素的名字、特征及其相互关系。

用模式 DDL 写出的一个数据库定义的全部语句称为一个模式。模式是数据库所有数据元素类型的一个结构图,它是数据库结构的一种描述,而不是数据库的数据本身,它只是装配数据的一个框架而已。

模式 DDL 和其他程序设计语言一样,有一个定义清楚的词汇表和一套语法规则。由模式 DDL 生成的模式称源模式。源模式由模式 DDL 翻译程序处理,产生一个目标模式,它是一组数据库表(目录和数据字典),DBMS 则借助这些表存取数据。

2. 子模式 SDDL 翻译程序

子模式 SDDL 是用户来定义他所用的局部逻辑数据库。

子模式 SDDL 定义了与此用户有关的数据元素的名字、特征及其相互关系。子模式必须命名。子模式是从模式映射而成的。因此子模式可以在许多方面与模式不同,如一个记录的数据项个数、数据项的名字等。也就是说,子模式只选取模式中的部分记录型。如果数据项的类型或名字不同时,则可以重定义。

同一个子模式可以被多个应用程序使用,但一个应用程序只能使用一个子模式。

源子模式经子模式 SDDL 翻译程序处理后产生目标子模式,它也是一组数据库表。

3. DML(数据操纵语言)处理程序

数据操纵语言 DML 是 DBMS 提供给用户存取、检索、修改、添加、删除数据库中数据的工具,亦称为数据子程序。

数据操纵语言一般有两种类型:一种是嵌入 COBOL、FORTRAN、C 语言等高级语言中,而不独立使用,这类数据操纵语言称为宿主型语言。另一种是查询语言,可以独立使用进行简单的检索、更新等,它通常由一组命令组成,以使用户提取数据库的数据,这一类语言称为自含型语言。

4. 终端查询语言解释程序

它用来解释终端询问的意义,决定操作的执行过程。

5. 数据库控制命令解释程序

它解释每个控制命令的含义,决定如何去执行。

二、系统运行控制程序

1. 系统控制程序。它是 DBMS 的神经中枢,它控制、协调 DBMS 各个程序模块的活动,使其有条不紊地进行工作。

2. 数据访问程序。根据用户的访问请求,实施对数据的访问,从物理文件中查找数据,执行插入、删除、修改等操作。

3. 并发控制程序。在许多用户同时访问数据库时,协调各个用户的访问。例如,按优先级排队,封锁某些访问或某些数据,撤消某种封锁,允许某个访问执行等,并能保证数据的相容性。

4. 合法权检验程序。其作用是核对用户标志、口令、对照授权表检验访问的合法性等。

5. 完整性控制程序。其作用是在操作前后,核对数据库完整性约束条件,从而决定是否允许操作执行,或清除操作执行后的影响。

6. 通信控制程序。它用来实施用户程序与 DBMS 之间的通信。

三、例行维护公用程序

1. 数据装入程序。用于大批原始数据按某种文件组织方式存贮到外部介质上,完成数据库装入。

2. 重新组织程序。当数据库系统的性能变坏时,需要对数据库重新进行物理组织。

3. 系统恢复程序。当软件、硬设备遭到破坏时,该程序可把数据库系统恢复到正常状态。

4. 工作日志程序。它的作用是负责记载进入数据库的所有询问。其内容包括:用户名、进入系统时间、进行何种操作等等,使每个访问都留下踪迹。

总之,DBMS 是一个规模较大的程序系统,它由操作系统支撑,而且又支撑应用系统。

1.4 数据模型

一、树的概念

树是由一个或多个结点组成的有限集 T ,其中有一个特定的结点,称为根;其余的结点为 $m(m>0)$ 个互不相交的有限集 $T_1, T_2, \dots, T_m$,其中每个有限集 $T_i(i=1, 2, \dots, m)$ 又是一棵树,称为根的子树。

由上述定义,树具有两个特点:

- 1) 树中至少有一个结点为根;
- 2) 树中各子树是互不相交的,如图 1-6 所示。

结点的子树数称为该结点的度。如图 1-6 中,A 的度数为 3,B 的度数为 2,……。

度数为 0 的结点称为树叶。如图 1-6 中的结点 K,L,M,F,G,J,I 均为树叶。

结点的层次:从根开始,根为第一层,紧连根的结点为第二层,……。

树中结点的最大层数称为树的深度。如图 1-6 中树的深度为 4。

结点的子树的根称为该结点的子女,而该结点称为子女结点的双亲。如图 1-6 所示的结点 E、F 为结点 B 的子女。结点 B 是结点 E 和 F 的双亲。

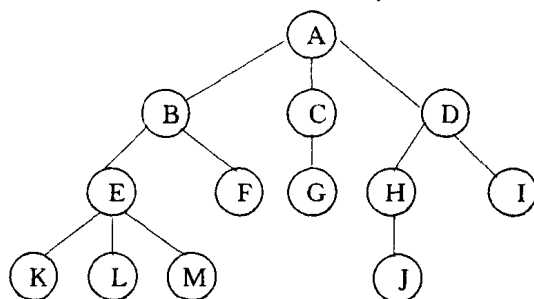


图 1-6 树的示意图

二、实体及其属性

1. 实体

所谓实体是指客观存在的并可互相区别的任何事物。例如,人,工厂,设备,一个规划,一幅画,……。

2. 属性

所谓属性是指实体代表的某一特定事物所具有的某方面的特征。例如,把工厂作为实体,在资产调查中,用厂名、厂长、地址、职工、设备、固定资产、产值等几个属性来刻画工厂的特征。而在产品调查中,用厂名、厂长、地址、产品名、型号、成本、出厂价等几个属性来刻画工厂的特征。

因此,属性的设定是完全根据需要选择的。另一方面,同一个事物因设定了不同的属性而变成了不同的实体。

3. 型与值

实体与属性有型与值之分。所谓型是指结构,值是指在结构约束下的具体取值。

例如,一张只有表头的空表,就叫做实体型,也叫做框架,如表 1-1 所示

表 1-1

姓 名	部 门	年 龄	性 别	级 别

在实体型下,按照栏目要求填入的数据叫做实体值。如表 1-2 所示。

表 1-2

姓 名	部 门	年 龄	性 别	级 别
杨 梅	财务科	32	女	19
江 涛	技术科	45	男	16

同理,属性也有型和值之分。如表 1-2 中,姓名是属性型,属性型下所填的数据杨梅、江涛就称为属性值;部门是型,财务科、技术科是部门(型)下的值;……。