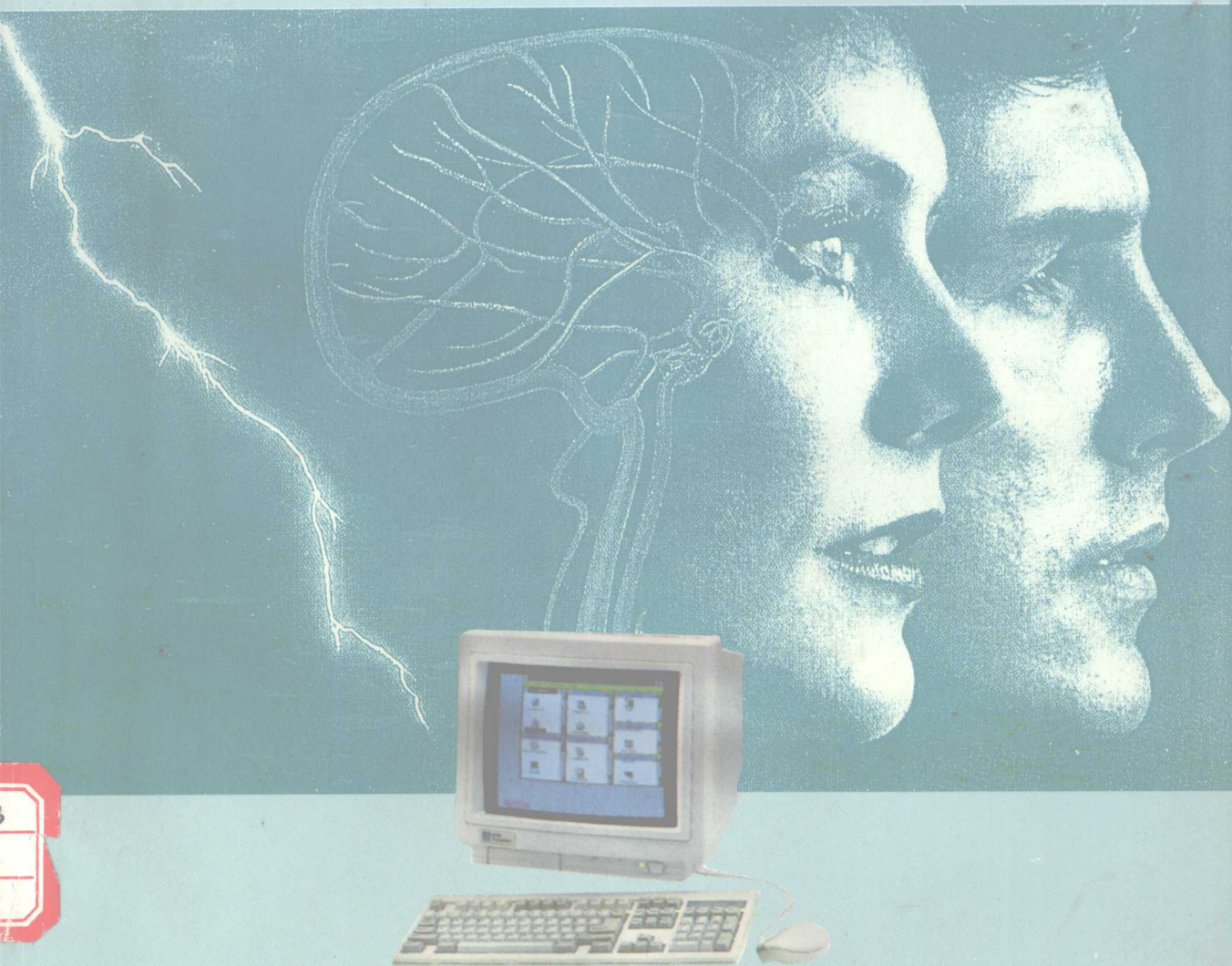


牛允鹏 主编

SHUJUKU JIQI YINGYONG

全国高等教育自学考试教材
计算机及其应用专业(专科)

数据库及其应用



南京大学出版社

全国高等教育自学考试教材
计算机及其应用专业

数据库及其应用

主 编 牛允鹏

副主编 迟成文

主 审 赵振西

南京大学出版社

内

容

简

介

本书是按照全国高等教育自学考试计算机及其应用专业(专科)制定的《数据库及其应用自学考试大纲》编写的。

全书共分十章。前五章着重介绍数据库的基本理论,包括数据库系统的基本概念、数据模型、数据库的存储结构和数据库的设计方法,重点介绍了关系型数据库的基本设计思想。后五章侧重介绍目前广为流行的关系型数据库管理系统 FoxBASE⁺,包括 FoxBASE⁺的使用基础、数据类型、常量、变量、函数、表达式等基本概念,建立与操作数据库的各种命令和程序设计语句,以及程序设计方法和若干技巧,还简要介绍了网络环境下多用户 FoxBASE⁺的程序设计方法。

本书概念清楚、结构合理、层次分明,安排了大量操作实例,理论与应用并重。可作为大专院校专科层次的教材,也可作为数据库应用研究与开发人员的参考书。

(苏)新登字第 011 号

全国高等教育自学考试教材

计算机及其应用专业

数据库及其应用

牛允鹏 迟成文 编

*

南京大学出版社出版发行

(南京大学校内 邮政编码:210008)

中国计算机函授学院照排室排版

安徽省地质印刷厂印刷

开本:787×1092/16¹ 印张:21 字数:504 千

1994 年 4 月第 1 版 1996 年 4 月第 2 次印刷

印数 45001—100000

ISBN: 7-305-02676-X/TP·99

定价:22.80 元

出版前言

高等教育自学考试教材是高等教育自学考试工作的一项基本建设。经国家教育委员会同意,我们拟有计划、有步骤地组织编写一些高等教育自学考试教材,以满足社会自学和适应考试的需要。《数据库及其应用》是为高等教育自学考试计算机及其应用专业组编的一套教材中的一种。这本教材根据专业考试计划,从造就和选拔人才的需要出发,按照全国颁布的《高等教育自学考试计算机及其应用专业(专科)数据库及其应用自学考试大纲》的要求,结合自学考试的特点,组织高等院校一些专家学者集体编写而成的。

计算机及其应用专业《数据库及其应用》自学考试教材,是供个人自学、社会助学和国家考试使用的。现经组织专家审定同意予以出版发行。我们相信,随着高教自学考试教材的陆续出版,必将对我国高等教育事业的发展,保证自学考试的质量起到积极的促进作用。

编写高等教育自学考试教材是一种新的尝试,希望得到社会各方面的关怀和支持,使它在使用中不断提高和日臻完善。

全国高等教育自学考试指导委员会

一九九四年四月

编者的话

本书根据全国高等教育自学考试指导委员会电类专业委员会 93 年 6 月 9 日组织召开的计算机及其应用专业教材工作会议拟定的计划,严格按照经专家评审的、国家教委自学考试办公室批准的自学考试大纲编写的。

全书共分十章。第一至第五章介绍了数据库系统的基本概念,重点介绍了关系型数据库原理;第六至第十章,以关系型数据库管理系统 FoxBASE+2.1 为蓝本,详细介绍了数据库系统的应用;两者相互联系,又各有侧重。在内容安排上,原理在前,应用在后。这是出于系统性考虑而做出的一种安排,并不意味着必须按章节次序阅读。相反地,我们建议初学者先读应用部分,积累一定感性知识后再去读原理,这样做有利于理解理论中的一些抽象概念。

本书编写立足于函授和自学,全书安排了丰富的实例,便于读者效仿,举一反三,起到事半功倍的效果。讲解中力求避免抽象地数学论证,重视基本概念,并对重要概念从不同角度反复进行了论述。编写中力求重点突出,使读者能在繁杂的内容中把握尺度、重点对待。书中带有“*”号部分或用小字排版的内容供选读,它们不属于自学考试大纲要求的范围,略去它们也不会影响后续章节的学习,但了解这部分内容有助于自身水平的提高。

本书第一章到第五章和第十章由牛允鹏教授编写,第六章到第九章由迟成文副教授编写。中国科技大学赵振西教授主审,东南大学周佩德副教授、安徽大学谢荣传副教授副审,在此向审稿的同志表示衷心感谢。

由于编者水平有限、时间仓促,书中疏漏之处请读者批评指正。

主 编

一九九四年四月十日

目 录

第一章 数据库系统基本概念	(1)
§ 1.1 数据与信息、数据处理与数据管理.....	(1)
§ 1.2 数据管理技术的发展	(2)
§ 1.3 数据库系统的组成及其特点	(5)
§ 1.4 数据库体系结构	(9)
§ 1.5 DBMS 的组成	(12)
§ 1.6 数据库保护.....	(14)
习题	(19)
第二章 数据模型	(20)
§ 2.1 什么是数据模型.....	(20)
§ 2.2 实体和属性.....	(21)
§ 2.3 实体间的联系.....	(23)
§ 2.4 数据库类型.....	(25)
§ 2.5 E—R 方法	(30)
§ 2.6 从 E—R 图导出关系数据模型	(34)
习题	(37)
第三章 数据库存储结构	(39)
§ 3.1 磁盘设备和磁盘与主存间的数据交换.....	(39)
§ 3.2 文件的组织方式.....	(41)
* § 3.3 B 树与 B ⁺ 树	(44)
* § 3.4 数据库实现联系的方法.....	(48)
* § 3.5 微型机上的 DBMS	(50)
习题	(54)
第四章 关系数据库理论基础	(55)
§ 4.1 关系的数学定义.....	(55)
§ 4.2 关系的性质.....	(57)
§ 4.3 关系数据库描述.....	(58)
§ 4.4 关系数据操作语言 DML	(60)
* § 4.5 关系数据库标准语言——SQL	(68)
§ 4.6 关系规范化.....	(74)
习题	(85)
第五章 数据库设计	(88)
§ 5.1 数据库设计的内容及特点.....	(88)
§ 5.2 数据库逻辑设计.....	(90)

§ 5.3	数据库物理设计	(96)
§ 5.4	应用程序与系统的运行和维护	(97)
§ 5.5	编写技术文档	(99)
	习题	(101)
第六章	FoxBASE+概述	(102)
§ 6.1	组成、运行环境和安装	(103)
§ 6.2	启动和退出	(105)
§ 6.3	两种使用方式	(106)
§ 6.4	数据类型	(107)
§ 6.5	简单的操作实例	(110)
§ 6.6	常量和变量	(118)
§ 6.7	常用函数	(123)
§ 6.8	表达式	(132)
§ 6.9	命令结构	(136)
§ 6.10	主要技术参数和文件类型	(139)
	习题	(142)
第七章	FoxBASE+对数据库文件的基本操作	(145)
§ 7.1	全屏幕编辑键	(145)
§ 7.2	数据库文件结构的建立和记录的初始输入	(146)
§ 7.3	数据库文件的打开和关闭	(150)
§ 7.4	数据库文件结构的显示和修改	(151)
§ 7.5	记录的显示、定位与顺序查询	(155)
§ 7.6	记录的追加与插入	(161)
§ 7.7	记录的删除	(165)
§ 7.8	记录的修改	(170)
§ 7.9	数据库文件的全屏幕编辑	(172)
§ 7.10	数据库文件的排序、索引和快速查询	(177)
§ 7.11	数据记录的统计处理	(188)
§ 7.12	数据库文件的复制	(192)
§ 7.13	多数据库文件的操作	(198)
§ 7.14	操作过程的记录	(205)
	习题	(208)
第八章	FoxBASE+程序设计基础	(211)
§ 8.1	FoxBASE+程序的建立、运行、调试和修改	(211)
§ 8.2	自定义函数	(216)
§ 8.3	非格式化的输入/输出语句	(219)
§ 8.4	格式化输入/输出语句	(223)
§ 8.5	格式设计	(228)
§ 8.6	程序控制语句	(235)

§ 8.7	内存变量的操作	(248)
§ 8.8	常用环境参数的设置语句	(252)
§ 8.9	磁盘文件的操作语句	(255)
§ 8.10	其它语句	(256)
习题		(258)
第九章	FoxBASE+程序设计技巧	(263)
§ 9.1	菜单程序设计	(263)
§ 9.2	数据库常用操作的程序设计方法	(273)
§ 9.3	保密程序段的设计方法	(283)
§ 9.4	数组在程序设计中的应用	(285)
§ 9.5	宏替换在程序设计中的应用	(288)
习题		(290)
第十章	多用户及网络环境下 FoxBASE+ 的应用程序设计	(294)
§ 10.1	Novell 网简介	(294)
§ 10.2	多用户 FoxBASE+ 在 Novell 网上的安装与启动	(298)
§ 10.3	多用户操作时的并发控制	(300)
§ 10.4	避免死锁的方法	(304)
§ 10.5	多用户命令与函数	(305)
§ 10.6	错误的捕获与处理	(307)
习题		(310)
附录一	FoxBASE+ 函数索引表	(312)
附录二	FoxBASE+ 命令功能分类表	(315)
附录三	FoxBASE+ 命令索引表	(316)
附录四	FoxBASE+ 环境参数设置命令索引表	(321)
附录五	FoxBASE+ 错误提示信息	(323)

第一章

数据库系统基本概念

数据库是一门研究数据管理的技术,始于 60 年代末,经过 30 多年的发展,已自成理论体系,成为计算机软件的一个重要分支。数据库技术体现了当代先进的数据管理方法,赢得社会的广泛承认,并使计算机应用真正渗透到国民经济各部门,在数据处理领域中正在发挥越来越大的作用。

本章从数据管理技术的发展,引出数据库概念,围绕着数据库系统,介绍有关名词术语,从而为读者建立起关于数据库系统的基本概念。

§ 1.1 数据与信息、数据处理与数据管理

数据,是数据库系统研究和处理的对象。但是数据与信息是分不开的,它们既有联系又有区别,因此首先得搞清数据与信息概念上的不同。

计算机的出现,开辟了电子数据处理的新纪元。电子数据处理中的核心问题是数据管理,即怎样有效地管理大量的数据、如何合理地组织、存储、检索和维护数据,这些正是数据库系统所要解决的问题。

一、数据与信息

数据,本质上是对客观事物特征的一种抽象的、符号化的表示,即用一定的符号表示那些从观察或测量中所收集到的基本事实。采用什么符号完全是一种人为的规定,例如在计算机内部用 0 和 1 两个符号描述事物的特征。值得注意的是,不要把数据仅仅理解成“数量”概念,它有着更广泛的含义,除了量化特征描述外,通常还需要一些非量化特征的描述。例如,气象预报中,温度的高低固然可以量化地表示,而“刮风”或“下雨”等特征则需用文字或图形符号进行描述。后者也是数据,不过数据类型不同罢了。

未经处理的数据只是基本素材,仅当对其进行适当地加工处理,产生出有助于实现特定目标的信息对人们才有意义。可见,信息实际上是指经过处理后的数据,是消化了的数

据。例如,“除去物价上涨因素,本市今年生活指数较去年同期提高了8个百分点”,这是一条信息,它的产生是经大量原始数据资料的分析后得出的结论,而其表现形式虽然也是数据,但含义不同了。

综上所述,数据与信息两者密不可分,既有联系,又有区别。数据表示了信息,而信息只有通过数据形式表示出来才能被人理解和接受。尽管数据与信息两者在概念上不尽相同,但通常人们并不严格去区分它们,例如数据处理也可称为信息处理。

信息是资源,可以被利用,并影响人的行为动作。人类进行各项社会活动,不仅要考虑物质条件,而且要认真研究信息,实际上人们的活动总是伴随着数据处理的全过程,人们越来越认识到信息的重要性,因此人们将物质、能量与信息并列为人类社会活动的三个基本条件,是十分正确的。

二、数据处理与数据管理

数据处理也可称为信息处理,因为当把客观事物表示成数据后,这些数据便被人们赋予了特定的含义,从而为人们提供了不必直接观察和度量事物就可以获得有关信息的手段。数据处理的基本含义是从某些已知的数据出发,推导出一些新的数据,这些新的数据又表示了新的信息。具体操作中,涉及到数据收集、管理、加工利用乃至信息输出的演变与推导全过程。

数据处理,通常计算比较简单,但是有其本身固有的特点,表现为数据量大,数据结构复杂,数据之间有复杂的逻辑联系。因此,数据处理业务矛盾焦点不是计算,而是数据管理,后者是指数据的收集、整理、组织、存储、维护、检索、传送等操作,这部分操作是数据处理业务的基本环节,而且是任何数据处理业务中必有的共性部分;而怎样加工和计算,则不同业务处理各不相同。因此,对数据管理部分,理当加以突出,集中精力研制出一个通用、高效而又使用方便的管理软件,把数据有效地管理起来,以便最大限度的减轻程序员的负担;至于处理业务中的加工计算,因不同业务各不相同,要靠程序员根据业务情况编写应用程序加以解决。

所以,数据处理是与数据管理相联系的,数据管理技术的优劣,将直接影响数据处理的效率,数据库技术正是瞄准这一目标研究、发展并逐渐完善起来的专门技术。

§ 1.2 数据管理技术的发展

如同其它科学技术的发展一样,数据管理技术也有一个发展的历程,大体上经历了三个发展阶段:

- 自由管理方式
- 文件管理方式
- 数据库方式

一、自由管理方式

最早的计算机,没有必要的软件支持,因此用户使用计算机进行数据处理不得不自行管理数据,程序员在程序中,既要考虑处理方法,又要管理数据的组织存储,如图 1-1 所示。

存取数据是根据设备的物理地址进行的,这种方式迫使程序员直接与物理设备打交道。管理数据存放地址,导致程序与数据紧密相关,程序高度依赖于数据,数据稍有变动,整个程序就必须全部进行修改,编程效率低,程序不灵活而且容易出错。

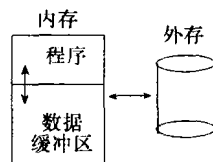


图1-1 自由管理方式

二、文件管理方式

文件管理方式,是把有待于加工处理的数据组织成数据文件,文件可以命名,一旦命名之后,程序中便可以通过文件名逻辑性地存取文件中的数据,解脱了程序员直接与物理设备打交道的沉重负担。这二者间需要一个转换过程,但这是由文件管理系统完成的,程序中不再涉及任何物理细节。由操作系统的知识我们知道,文件是操作系统管理数据的基本单位,文件管理系统负担数据的逻辑组织与物理组织之间的映射。

数据组织成文件后,在一定程度上使程序仅涉及文件中的数据,而与数据的物理存储位置无关,这是我们所追求的,目的是克服数据任何存储位置的变动导致程序全部作废的弊病。

由于文件可以根据数据所代表的含义,分门别类,按一定规则加以组织,所以文件中的数据能描述现实世界事物的特征。

- 起着描述事物各种特征或性质的最小数据单位,称数据项;
- 若干数据项的集合,称记录。记录能表达事物,刻划某种信息;
- 相同性质的数据项构成的记录集合,称文件;

显然,数据有规律地组织成文件形式,含义清楚,逻辑关系非常明确,使数据处理真正体现了信息处理,按名存取数据,既形象,又方便。

已如上述,文件是操作系统管理的基本单位,但是文件怎么组织,则是用户的事。在文件方式下,每个用户都可以建立、维护或者处理一个文件或多个文件。

文件管理方式,是数据管理的一大进步,直到今天,仍被广泛采用。顺便指出,即使是数据库方式,也是以文件方式为基础的。鉴于文件管理方式的重要性,下面举例说明。

[例] 设图 1-2 是假想的汽车运输公司的组织机构,方框代表部门,方框间的连线表示部门之间有业务联系。

假定:

①货运室处理货运业务,建立货运文件如图 1-3 所示。

②调度室处理调度业务,建立调度文件如图 1-4 所示。

③财务科处理工资业务,建立工资文件如图 1-5 所示。

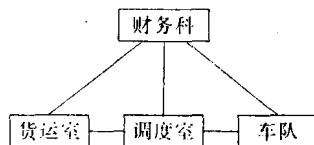


图1-2 汽车运输公司体制

④车队处理派车业务,建立出车文件如图 1-6 所示。

单位	合同号	品名	数量	发地	目的地
机床厂	93001	杉木	15	黄山	合肥
叉车厂	93002	钢管	4.5	马鞍山	南京
⋮	⋮	⋮	⋮	⋮	⋮

图 1-3 货运文件

车队号	品名	数量	发地	目的地	货到日期
01	杉木	15	黄山	合肥	93.9.25
02	钢管	4.5	马鞍山	南京	93.10.8
⋮	⋮	⋮	⋮	⋮	⋮

图 1-4 调度文件

姓名	基本工资	浮动工资	扣款
丁一	125	70	30
王二	110	65	18
张三	150	80	60
⋮	⋮	⋮	⋮

图 1-5 工资文件

出车日期	姓名	派车单号	车型	品名	数量	发地	目的地
93.9.22	丁一	9309221	东风	杉木	5	黄山	合肥
93.9.22	王二	9309222	解放	杉木	5	黄山	合肥
93.9.23	张三	9309231	东风	杉木	5	黄山	合肥
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

图 1-6 出车文件

这样,每个用户都建立自己的文件,处理和维持自己建立的文件,在文件管理系统支持下按名存取数据进行加工计算,这种方式可由图 1-7 示意表示。

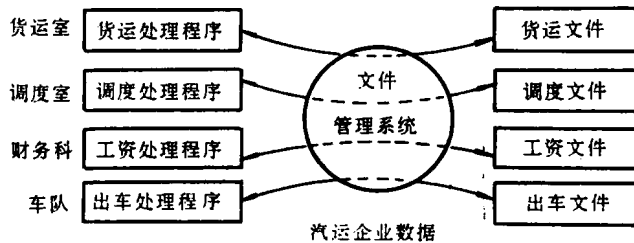


图 1-7 文件管理方式

上例所述,文件组织面向应用,有一个应用,必有一文件(也可能多个)与之对应,文件由用户建立和维护,一般表示形式,如图 1-8 所示。

[分析] 仔细想来,文件管理方式并不合理,因为用户自行其事,互不通气自建文件。然而企业是个整体,各部门互有联系,数据要相互引用,但是由于文件面向应用,导致各部门不得不重复收集并存储大量相同数据。这一点只需留意图 1-3~图 1-6 便知,因而造成诸多弊病,表现为:

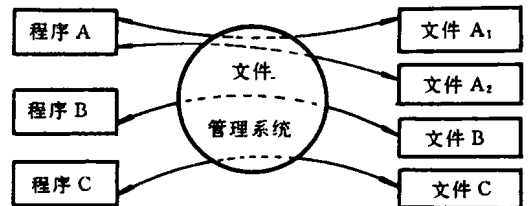


图 1-8 文件管理方式一般表示

①程序与文件相互依存,这说明程序仍与数据相关,这是不希望的,因为文件有所变动,程序得相应修改,而文件离开了使用它的程序便全部失去存在的价值;

②数据冗余大;

③数据可能发生矛盾,因为同一个数据出现在不同文件中,随着数据更新,有可能造成同一个数据在不同文件中具有不同数值,这是数据冗余的恶果;

④不能反映企业间的业务联系。

由于企业各部门有着千丝万缕联系,这种联系至关重要,数据要互相引用,但是文件方式,面向应用组织文件,把至关重要的联系信息丢掉了。

三、数据库方式

针对文件方式的弊端,后来出现了数据库技术。数据库技术的目标,首先是克服程序与文件相互依存,力求数据独立,其次重在表现数据之间的联系,还要尽量克服数据冗余,解决数据安全性和完整性的保护问题。

数据库方式与文件方式根本区别在于,前者面向系统,后者面向应用,即数据库方式综合全体应用,整体观点组织数据,形成一个具有一定数据结构的集成化的数据集合,这个数据集合实际上就是数据库。

图 1-9 所表示的综合,并非数据的简单相加,而是通过一定数据结构形式把各个文件联系起来所形成的统一整体,这个整体如同一个数据仓库,能从其中导出不同用户所需要的数据。

在数据库方式下,某特定用户处理自己业务时,不是自建文件,而是从事先经过严密设计的数据库中导出所需要的部分作为应用程序使用的文件,它并非独立存在,更非实际存储,而是取自库中的某个子集,又因它是从数据库中映射出来的,所以这个文件叫逻辑文件。如图 1-10 所示。

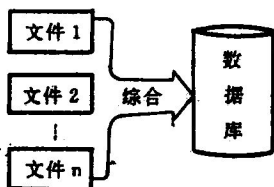


图 1-9 数据库是综合各用户数据形成的数据集合

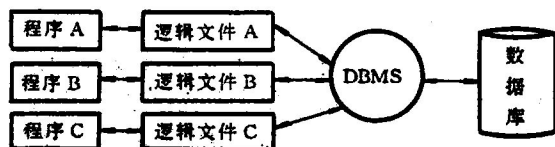


图 1-10 数据库工作方式示意图

值得注意的是,图 1-10 中表明应用程序只能通过 DBMS 与数据库接口,DBMS 是“数据库管理系统”的简称,全文是:Data Base Management System,它是管理、控制数据库的软件,由 DBMS 负责数据库与逻辑文件的映射,用户只需指出映射要求。此外,数据库方式下,逻辑文件的维护是由 DBMS 完成的,不必象文件方式那样程序员自己维护文件,这当然减轻了程序员管理数据的负担,大大提高了编程效率。

有了这种初步认识,下面进一步说明什么是数据库系统。

§ 1.3 数据库系统的组成及其特点

数据库实际上是个系统,它由数据库和支持数据库运行的软、硬件环境以及应用程序和数据库管理员等相互独立而又相互联系的若干部分所组成。

一、系统的组成

1. 数据库

前已说明,数据库是一个结构化的相关数据集合,主要是通过汇总各个数据库用户的文件,除去不必要的冗余,然后使各个文件相互联系,从而形成整体数据结构。联系是数据库的重要特点,至于怎样实现这种联系,取决于数据库类型,类型不同,联系手段也不一样,这个问题在第二章详细讨论。

数据库是集成化的,能为多种应用服务,它独立于应用程序而存在,是数据处理系统的核心和管理对象。

2. 数据库管理系统 DBMS

在数据库系统中,DBMS 是专门用于数据管理的软件。通常,DBMS 提供数据库定义和数据装入功能。提供数据操纵(包括检索与数据存取操作),数据控制(包括安全性、完整性和并发控制)和数据库维护(包括数据库整理、修改与重定义等)等功能。

DBMS 提供应用程序与数据库的接口,允许用户逻辑地访问数据库中的数据,负责逻辑数据与物理地址间的映射,是控制和管理数据库运行的工具。

3. 支持数据库运行的硬、软件环境

(1)硬件

硬件是数据库赖以存在的物理设备,包括 CPU、存储器和其它外部设备。数据库系统要求很大的内存,用来存放系统程序、应用程序和表格缓冲区等,应配备足够容量的内存。数据库系统对外部存储器有特殊要求,一般应配置较大容量的磁盘设备,用来存放需要长期保存和反复使用的大量数据。磁盘是一种能支持直接数据存取的外部设备,它为索引文件、链接文件、直接存取文件和倒排文件等文件组织形式和数据存取方法提供物理支持,是数据库系统不可缺少的设备,一般应配置几十 MB 或几百 MB 以上。

(2)软件

数据库系统是一个计算机应用系统。在计算机应用系统中,基本的系统软件是操作系统,其它任何软件(包括 DBMS)都必须在它的支持下工作。因此当选用某种 DBMS 时,必须选择能对 DBMS 提供支持的操作系统;如果想要处理汉字,还必须配上汉字操作系统和相应的软件和硬件(如汉字系统 UC DOS、汉字打印机等)。

4. 应用程序

数据库是多用户共享的,不同用户有不同业务,他们所需要的数据,已由数据库设计者精心规划在数据库中,但是如何使用库中数据是用户的事,要自编应用程序处理自己业务,其操作权限仅是数据库的一个子集。

5. 数据库管理员 DBA(Data Base Administrator)

在数据库系统中,人的因素起重要作用,应该把 DBA 看成系统不可忽略的一部分。

大型数据库系统,一般配备专职 DBA,他们的职责是对数据库进行管理和维护,保持数据库始终处于最佳工作状态。

数据库的建立,是由计算机专业人员和有关业务人员共同组成的班子,从需求分析开始,汇总所有用户的数据,进行分析、综合、提炼、画出全局逻辑结构图,搞清数据来龙去脉,定义各种数据,包括数据类型、结构、值域、来源、用途、使用权限、保密等级等等,构成数据字典。这个数据字典供系统参考使用。

微型机数据库,DBA 由用户承担。

综上所述,数据库系统以硬件为基础,如图 1-11 所示,在硬件基础上层层套上各种软件。外层依赖于内层支持,可以调用内层但不必考虑内层细节。图中显示了 DBMS 在操作系统支持下与其它语言处理程序可以在一个系统中并存,用户可以根据自己的需要选用 DBMS 或其它语言开发程序。但是一定要注意,如果应用程序访问数据库中的数据,就必须通过 DBMS 才能进行。

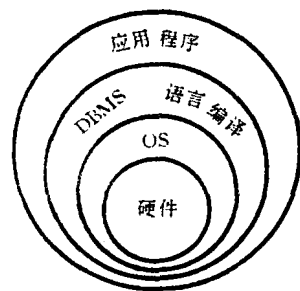


图 1-11 软、硬件关系

二、数据库系统的特点

由于数据库是个复杂的系统,难以用简炼语言概括其全部特征,实际上关于什么是数据库,至今还没有一个公认的严格定义,当我们从不同角度描述数据库时,可能给出的定义出入很大。

例如,C. J. Date 在他的《数据库系统引论》一书中,从以下三方面描述了数据库,如图 1-12 所示:

- ①数据库是存储在磁鼓、磁盘或其它存储介质上的数据集合;
- ②有若干应用程序以数据库为背景进行检索、修改、插入或删除等操作,还可能有一些联机远程终端用户访问数据库;
- ③数据库是集成的,包含许多用户的数据,每个用户只享用其中一部分数据,不同用户所使用的数据可以重叠,并且同一片数据可以为多用户共享。

再如,J. Martin 在《数据库组织》一书中对数据库是如下定义的:

数据库是存储在一起的相关数据的集合,这些数据没有不必要的冗余,能为多种应用服务;数据的存储独立于程序;对数据库的插入、修改和检索均能按一种公用的和可控的方法进行;若在一个系统中,存在着结构上完全分离的多个数据库,则称该系统为一个数据库集合。

如上所述,对什么是数据库,从不同角度描述可能差别很大,但是对数据库所应具有的基本特点,认识大体上是一致的。我们也应从它的基本特点去认识数据库,下面分析一下数据库究竟有哪些特点

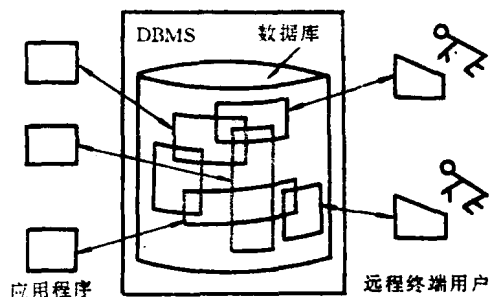


图 1-12 Date 描述的数据库

1. 数据独立

数据独立是数据库方式管理数据的基本优点,也是数据库系统努力追求的目标。数据独立的含义是数据的存储独立于应用程序,以保证数据的改变不致或很少影响应用程序。

数据独立又有两个含义:物理数据独立性和逻辑数据独立性。

所谓物理数据独立性,是指数据库物理结构(包括数据的组织和存储、存取方法、外部存储设备等)发生改变时,不会影响到数据库的逻辑结构,而应用程序使用的是逻辑数据,因此可以不必改动应用程序。

所谓逻辑数据独立性,是指由于某种原因使数据库的全局逻辑结构发生改变时,用户的应用程序不需要改动,似乎数据库并未发生变化一样。这是因为应用程序是根据该用户的数据视图编写的,仅是全局逻辑数据的一个子集,全局结构变化与否与具体用户无关,只要能从全局导出用户的局部视图就行。

以上两种数据独立性,是由 DBMS 所提供的映象功能实现的,在下一节还要进一步说明这个问题。

2. 数据共享

所谓数据共享,是指不同用户可以使用同一个数据库,各自取出他们所需要的子集,而且允许子集任意重迭,这是由于数据集成带来的好处。不过当碰巧多个用户同时对某一数据更新时,要进行控制,避免数据发生矛盾,一般 DBMS 都具有并发控制功能,这个问题在 § 1.6 介绍。

3. 减少了数据冗余

数据库方式,数据是集成化的,应用程序使用的逻辑文件是取自整体的子集,尽管一个数据可能出现在不同的逻辑文件中,但实际上的物理存储只可能出现一次,这就减少了数据冗余。

然而数据冗余有时难以避免,甚至有时为了某种需要有意地重复存储数据而带来其它方便,所以我们只说减少数据冗余而不提“避免”二字。

4. 数据的安全性与完整性

安全性指数据保密,为防止数据被窃或破坏,DBMS 设置严密措施,层层防护。

完整性指数据的正确性和一致性。安全性和完整性保护,不同的 DBMS 有不同的方法,功能有强有弱,这是要以时间开销为代价的。

5. 使用操作方便性

一个数据库系统允许使用多种程序设计语言(如 COBOL、FORTRAN、C 等)与数据库打交道,而有关数据操作命令则由 DBMS 解释,每条操作命令,几乎相当于高级语言的一段子程序,因此在数据库环境下,程序设计效率大为提高。

此外,DBMS 配有非过程化数据语言,使不懂计算机的人,也能在终端上操作自如。

§ 1.4 数据库体系结构

为了有效地管理数据,人们为数据库设计了一个严谨的组织结构。目前,世界上已经有相当数量的数据库正在运行,其种类和复杂程度不尽相同,但是就其体系结构而言却是大体相同的。

一、数据库三级组织结构

美国国家标准学会(ANSI)在 1975 年公布了一个关于数据库标准报告,该报告规定了数据库按三级体系结构组织,这就是有名的 SPARC 分级结构(Standards planning And Requirements Committee)。

三级结构以内层、中间层和外层三个不同层次描述数据库,分别称为内模式、模式和外模式,如图 1-13 所示。三个模式反映了看待数据库的三种不同数据观点,下面作一简要说明。

1. 模式

图 1-13 中的模式,在 ANSI/PSARC 报告中原称概念模式,这里简称模式。它是一种对数据库数据组织的全局逻辑观点,并不涉及数据的物理存储,后者是由内模式定义的。模式对应于总体概念级数据库,故称为 DBA 视图。

2. 内模式

内模式又称存储模式,它具体描述了数据如何组织存入在外部存储器上。内模式是由系统程序员选择一定的文件结构组织起来的,也是由他们编制存取程序实现数据存取的,故称内模式为系统程序员视图。

3. 外模式

外模式面向用户,是用户眼中的数据库,故称外模式为用户视图。通常,外模式是模式的一个子集,所以又称外模式是子模式,它包含了模式中允许特定用户使用的那部分数据。

三个模式中,模式是内模式的逻辑表示;内模式是模式的物理实现;外模式则是模式的部分抽取。

三个模式反映了对数据库的三种不同观点:模式表示了概念级数据库,体现了对数据

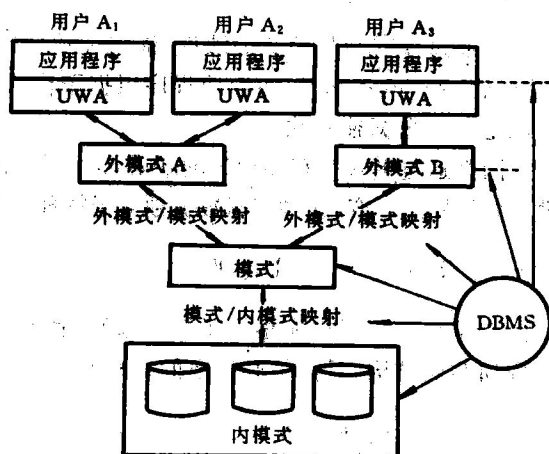


图 1-13 SPARC 分级结构