

第1章 数据库基础

在远古时代,人类就学会了用数据描述现实世界,通过数据的运算来表征现实世界的变化。随着社会活动及生产活动的发展,数据及其运算越来越复杂,从而逐渐产生了运算的机械化与自动化的需要,中国古代的算盘和西方十九世纪的机械计算装置反映了人类的这种追求。本世纪四十年代,由于电子技术的发展,为满足二次世界大战战术数据处理的需要,开始了电子自动计算工具的研究,1946年诞生了第一台电子计算机。四十多年来,电子计算机获得了飞速发展,在社会活动的各个领域中获得广泛的应用,其中电子数据处理领域逐步成为计算机应用的主流领域。

本章首先讨论数据处理的基本概念,然后介绍 FoxPro 2.5 for Windows 的运行环境及各项技术指标,最后介绍系统的安装、启动及联机帮助。

§ 1.1 信息、数据和数据处理

用计算机存贮和管理现实世界的信息,必须研究信息与数据的关系,研究数据在计算机中的存贮方式和处理方式。

信息是对各种事物的存在方式、运动状态和相互联系特征的一种表达和陈述。信息是自然界、人类社会和人类思维活动中普遍存在的一切物质和事物的属性。从计算机信息处理的观点出发可对信息作如下定义:

信息是现实世界在人脑中的抽象反映,是通过人的感官(眼、耳、鼻、舌、身)感知出来并经过人脑的加工而形成的反映现实世界中事物的概念。这些概念不仅为人们所认识和理解,而且把它作为知识来进行推理、加工和传播,从而达到认识世界、最终改造世界的目的。

这里所说的“事物”要比“物质”有着更广泛的含义。它不仅可指那些看得见、摸得着的具体物体,如汽车、房子,而且也包含那些不可触及的抽象概念,如产量、质量等等。因此,信息可以被看成是现实世界的真实反映。

在我们的学习、生活和工作中,经常要接触到各式各样的信息,而且频繁地传播、加工和利用这些信息。可以说,现实世界里到处都充满了信息。信息可以用人工或自动化装置进行记录、解释和处理。在用计算机来处理信息时,必须要将现实世界中的信息转换为计算机能够识别的符号,符号就是通常所说的数据。

如果说数据是反映客观的记录符号,那末“信息”则是潜在于数据的意义。信息反映了客观世界中各种事物的状态、特征与规律,它能增长人们的知识,并影响接受者的行为。从这种意义上说,数据是用来载荷信息的物理符号,是信息的载体。

数据和信息是两个互相联系、互相依赖但又互相区别的概念。信息以数据为载体而表现,数据则是表示信息的一种手段。同一信息可以有不同的数据表示方式,在计算机数据处理中,数据的格式往往与具体的计算机系统有关。

事物——信息——数据,实际上贯穿了三个世界,即现实世界、信息世界和计算机世界。

现实世界存在多种不同的事物类,事物类可以是实际的物体的集合,如人、书、机器零件,

也可以是抽象的概念的集合。每一事物类由隶属于该类的具体事物组成,如对于“人”这一事物类而言,它可以是包含了张某、李某…这些具体的人,每一个事物又具有它自己的内涵,如对张某来说,有姓名、性别、身高、体重…这些内涵。从而构成三个层次的概念。

信息世界是描述现实世界的,现实世界中的事物类,在信息世界中就叫实体集,与事物相对应的就是实体,与内涵相对应的称为属性。

而在数据世界(计算机世界)中,与三个层次概念相对应的分别是文件、记录和字段。

三个世界的类比关系可由图 1.1 表现。

现实世界	信息世界	数据世界
事物类	实体集	文件
事物	实体	记录
内涵	属性	字段

图 1.1 三个世界的类比关系

数据处理是指对信息进行收集、整理、存贮、加工及传播等一系列活动的总和,其基本目的是从大量的、杂乱无章的甚至是难于理解的数据中,提炼、抽取出人们所需要的有价值、有意义的数 据,借以作为决策的依据。数据的组织、存贮、检查和维护等工作是数据处理的基本环节,这些工作一般统称为数据管理。

§ 1.2 数据管理技术的发展

计算机数据管理技术经历了由低级到高级的发展过程,这一过程大致可分为三个阶段:

无管理— 文件系统——数据库系统

1. 无管理阶段

这一阶段的特点是:①数据不保存;②数据不能独立,它是程序的组成部分,即数据和程序完全结合成一个不可分割的整体。程序员对数据的存贮结构、存取方法及输入输出的格式拥有绝对的控制权,要修改数据必须修改程序;③数据是面向应用的,不同应用的数据之间是相互独立、彼此无关的,即使两个不同应用涉及到相同的数据,也必须各自定义,无法互相利用,互相参照。数据不仅高度冗余,而且不能共享。

2. 文件系统阶段

随着计算机应用进入数据处理领域,数据管理技术从五十年代后期进入文件系统阶段。这一阶段又可以分为两个阶段。

(1) 初等文件系统

它的特点是:①文件以顺序方式组织,适用于批处理方式;②数据的物理结构基本上等同于数据的逻辑结构,因此数据结构的改变必然导致应用程序的修改;③计算机的文件管理系统仅涉及数据的输入和输出;④数据可以共享,但数据高度冗余。

(2) 成熟的文件系统

它的特点是:①文件的组织方式既可以是顺序的,也可以是随机的,因此既能适用于批处理方式,也能适用于实时处理方式;②数据的物理结构不再等同于数据的逻辑结构,物理结构和逻辑结构之间有了简单的变换,数据和程序可以相互独立;③计算机的文件管理系统提供了

数据存取方法;④数据可以共享,但数据仍存在相当程度的冗余。

3. 数据库阶段

现实世界是复杂的,反映现实世界的各类数据之间必然存在错综复杂的联系。为反映这种复杂的数据结构,让数据资源能为多种应用需要服务,并为多个用户所共享,同时为让用户能更方便地使用这些数据资源,在计算机科学领域中逐步形成了数据库技术这一独立分支。计算机中的数据及管理统一由数据库系统来完成。

数据库系统的目标是:解决数据冗余问题、实现数据独立性,实现数据共享并解决由于数据共享而带来的数据完整性、安全性及并发控制等一系列问题。为实现这一目标,数据库的运行必须由一个软件系统来控制,这个系统软件称为数据库管理系统(Data Base Management System, DBMS)。

这一阶段的特点是:①具有面向全组织的复杂的数据结构;②数据冗余度小,易扩充;③程序独立于数据;④统一的数据管理功能,包括数据的安全性控制、数据的完整性控制及并发控制。

在详细讨论数据库系统之前,首先介绍数据库系统的三级模式结构。

尽管实际的数据库系统软件多种多样;支持不同的数据模型、使用不同的数据库语言、建立在不同的操作系统之上,但是绝大多数数据库系统在总的体系结构上都具有三级模式的结构特征。三级模式是指外模式、模式和内模式。

外模式:亦称子模式或用户模式。是数据库用户看到的数据视图,是与某一应用有关的数据的逻辑表示。不同用户的外模式可以互相覆盖,同一外模式可以为某一用户的多个应用所启用,一个应用只能启用一个外模式。

模式:亦称逻辑模式。是数据库中全体数据的逻辑结构和特性的描述,是所有用户的公共数据视图。它与具体的应用程序及使用的程序设计语言无关,模式不仅仅是数据的逻辑结构的定义,而且还要定义与数据有关的安全性、完整性,必要时还需定义数据项之间的联系及记录之间的联系。

内模式:亦称存储模式。是数据在数据库系统内部的表示,即对数据的物理结构和存储方式的描述。

模式描述的是全局逻辑结构。外模式涉及的是数据的局部的逻辑结构,外模式通常是模式的子集。

为了实现三级模式之间的转换,数据库系统在三级模式之间提供了两个层次的映像:

外模式到模式的映像,定义了外模式和模式之间的对应关系;

模式到内模式的映像,定义了数据的逻辑结构和物理结构之间的对应关系。

图 1.2 表示了三级模式和模式间两个层次的映像。

由于在三级模式之间存在这两种层次的映像,所以通过数据库系统可以实现数据的逻辑独立性及物理独立性。所谓数据的逻辑独立性是指:当模式发生变化时,只要外模式不变,应用程序就可以维持不变。物理独立性是指:由于存储数据的硬件设备变化或存储方法变化,引起内模式发生变化,但是数据的逻辑结构可以维持不变。

对数据库系统进行进一步分析,数据库系统是由计算机系统、数据库、数据库管理程序、应用程序集合及数据库管理员组成的。

计算机系统:指的是用于进行数据管理的计算机硬件资源和基本软件资源。硬件资源包括中央处理机,用来存放操作系统、数据库管理系统、数据库、应用程序、系统缓冲区等的大容量

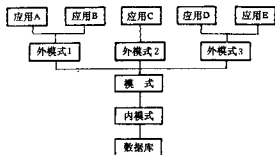


图 1.2 模式及映像

内存,以及包括可以直接存取的存取设备和必要的输入输出设备。软件资源包括操作系统及各种语言处理系统。

数据库:数据库是数据的仓库。可为数据库下这样一个定义: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。在计算机中,数据库是由很多数据文件及相关的辅助文件所组成的。

数据库管理系统:简称 DBMS,是一个以统一的方式管理、维护数据库中的数据的一系列软件的集合。DBMS 在操作系统的支持与控制下运行,按功能 DBMS 可分为三大部分:①语言处理部分。包括数据描述语言(DDL),用以描述数据模型;数据操纵语言(DML),是 DBMS 提供给用户的操纵数据的工具。当 DML 嵌入计算机的某种程序设计语言中作为子语言使用时,我们称这个程序设计语言为主语言。当 DML 可以单独使用时,我们称该 DML 为自含语言。有些数据库管理系统提供自己的程序设计语言。语言处理部分通常还包括数据库控制命令解释程序。②系统运行控制部分。包括系统总控程序;并发、数据安全性及数据完整性等控制程序;数据访问程序;数据通讯程序。③系统维护部分。包括数据装入程序、性能监督程序、系统恢复程序、重新组织程序及系统工作日志程序等。

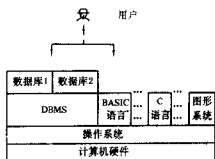


图 1.3 用户、数据库与计算机

用户一般不能直接加工或使用数据库中的数据,而必须通过数据库管理系统。DBMS 的主要功能是维持数据库系统的正常活动,接受并响应用户对数据库的一切访问要求,包括建立及删除数据文件、检索、统计、修改和组织数据库中的数据及为用户提供对数据库的维护手段等等。通过使用 DBMS,用户可以逻辑地、抽象地处理数据,而不必关心这些数据在计算机中的存放方式以及计算机处理数据的过程细节。这样,把一切处理数据的具体而繁杂的工作

交给 DBMS 去完成。

对大型数据库系统,用户一般是通过应用程序操作数据库,数据库的维护和管理工作由数据库管理员负责。而对 FoxPro 这类桌面型(Desktop)数据库系统而言,用户除通过应用程序访问数据库之外,可通过即席输入数据库操作命令的方式操作数据库,这时往往不再去设数据库管理员,数据库的维护及管理工作由用户直接负责。

用户、数据库管理系统、数据库及整个系统之间的关系如图 1.3 所示。

§ 1.3 数据模型

现实世界五彩缤纷,目前任何一种科学技术手段都不可能将现实世界按原样进行复制和管理。在计算机处理现实世界的信息时,只能根据需要,首先选择某个局部世界,并抽取这个局部世界的主要特征,特别是数据之间的结构关系,构造一个能反映这个局部世界的数据库模型。在数据库领域中,目前广泛应用的数据模型主要有三类:层次模型、网状模型和关系模型。

层次模型将反映现实世界的实体集抽象为一个严格的自上而下的层次关系。例如,在进行工业企业管理时,由于现实世界中管理体制基本上是个层次关系,所以可以构造如图 1.4 所示的模型。

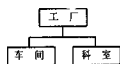


图 1.4 工业企业的数据模型

通过这个模型可以反映某个具体工厂的组织体制。(图 1.5)

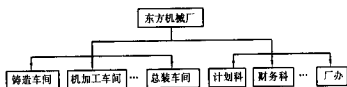


图 1.5 工厂的组织体制

在层次模型中,每个构造单元称为记录型。例如:工厂、车间、科室就是工业企业数据模型的记录型。上层记录型可对应一个或多个下层记录型(工厂对应车间、科室两个记录型)。每个记录型可以有一个或多个记录值,上层记录值对应下层一个或多个记录值,而下层每个记录值只能对应上层一个记录值。例如东方机械厂的下层记录值有铸造车间...、计划科...,而该计划科只属于东方机械厂。

对于现实世界的另一些问题,用层次模型不能有效地反映这些问题中各类实体型的现实联系,例如在讨论学校中教师、学生和开设课程这类问题时,可以构造出图 1.6 的模型。在这个模型中,教师和课程之间存在开课这种联系,教师和学生间存在管理联系,学生和课程之间还存在着选课联系。

这种模型称为网状模型,网状模型中每个记录型对应一个或多个其它记录型,每个记录型也存在一个或多个记录值,而每个记录值可能对应一个或多个其它记录型的记录值。



图 1.6 网状模型

网状模型和层次模型都是成功的数据模型,基于这些模型构造了一些成功的数据库管理系统。但是,这两种模型除各自固有的一些不足外,其共同的缺点是在用户处理数据库中的数据时,必须非常清楚数据之间的网状(或层次)联系,如果我们把数据库操作看作是在数据库的数据海洋中航行的话,用户必须时刻注意自己在数据模型结构中的位置及航向,所以基于这两类数据模型的数据库管理系统都称为“导航”式系统。由于数据之间的联系是根据应用的需要而对现实世界抽象的结果,一旦用户的应用需求发生变化,就可能要修改数据的模型结构,严重时可能危及整个应用系统。

针对层次模型和网状模型的这些缺陷,七十年代初提出了关系模型。在关系模型的数据库中,现实世界的数据库组织成一些二维表格,在关系模型中表格称为关系,用户对数据的操作抽象为对关系的操作。

在一个单位的建筑物管理工作中,可以构造建筑物一览表(简称建筑物关系)以反映单位建筑物的各项情况(图 1.7)。

建筑物名	建筑日期	造价(美元)	设计单位	教学用	备注	附图
大礼堂	01/31/31	400000.00	杨氏营造所	.F.		
中大院	01/31/29	300000.00	江南营造所	.F.		
老图书馆	01/31/29	200000.00	孝氏营造所	.F.		
新图书馆	01/31/85	300000.00	南工设计院	.F.		
中心楼	01/31/23	500000.00	南工设计院	.F.		
体育馆	01/31/23	200000.00	江南营造所	.T.		
五四楼	01/31/54	250000.00	南工设计院	.T.		
前上院	01/31/29	200000.00	江南营造所	.T.		
中山院	01/31/22	200000.00	江南营造所	.T.		
专家楼	01/31/86	300000.00	东大设计院	.F.		

图 1.7 建筑物一览表(关系)

使用这些建筑物的部门的数据可用一个部门关系来反映(图 1.8)。

名 称	所在建筑物	建立日期
计算机系	中心楼	1972
建筑系	中大院	1927
外语系	中心楼	1987
哲学系	新图书馆	1982

图 1.8 部门关系

建筑物设计单位的数据也可用一个设计单位关系来反映(图 1.9)。

名 称	管 理 者	电 话
杨氏营造所	杨先生	
孝氏营造所	孝先生	
江南营造所		
南工设计院	齐康	(025)7715599
东大设计院	齐康	(025)7715599

图 1.9 设计单位关系

粗略地说,关系对应于现实世界的表格,每个关系有一个关系名,关系在横向由若干个数据项(或称字段)组成,从纵向看,关系中每行数据称为一个记录。

再深入分析就会发现,建筑物关系不仅反映了每栋建筑的基本情况,还反映了建筑物和设计单位之间的联系,即反映了各建筑物由哪个设计单位设计。所以在关系数据库中,无论是数据还是数据之间的联系都是通过一些关系来反映的。

§ 1.4 关系数据库及其操作

关系数据库的数据组织形式是关系,关系直观上是二维表,但不是任何一种表格都可以作为关系数据库的一个关系,关系必须是规范化的。规范化最基本的要求是关系的每个数据项都必须是一个单纯域,或者说每个数据项都必须是一个不可分的数据项,而不允许表中有表。

例如建筑物关系按建筑物的设计单位组合在一起,构成的图 1.10 的表格就不是关系,因为不满足规范化的要求。

设计单位	建 筑 物					
	建筑物名	建筑日期	造价(美元)	教学用	备注	附图
杨氏营造所	大礼堂	01/31/31	400000.00	.F.		
茅氏营造所	老图书馆	01/31/29	200000.00	.F.		
南工设计院	新图书馆	01/31/85	300000.00	.F.		
	中心楼	01/31/23	500000.00	.F.		
	五四楼	01/31/54	250000.00	.T.		
江南营造所	体育馆	01/31/23	200000.00	.T.		
	前工院	01/31/29	200000.00	.T.		
	中山院	01/31/22	200000.00	.T.		
	中大院	01/31/29	300000.00	.F.		
东大设计院	专家楼	01/31/86	300000.00	.F.		

图 1.10 非关系的表格

上述规范化要求是最基本要求,满足这个要求的关系称为第一范式的关系。随着规范化要求的提高,还可以描述关系的第二范式、第三范式等等。但关系数据库最基本的要求就是每个关系都必须满足第一范式的规范化要求。

每个关系有一个关系名,关系中每个字段有一个字段名。每个字段根据其存储数据的不同特性而具有不同的类型。例如建筑物关系的名称为 building,每个建筑物由建筑物名、建造日期、造价、是否教学用、设计单位、备注及附图这七个字段的值进行描述。建筑物名和设计单位由字符组成,是否教学用为“是”或者为“不是”,从而可以用逻辑的真假值表示,建造日期字段中是 10/10/49 这种日期型数据,造价字段中填入数值型的值,备注中是一些说明信息,每个建筑物的备注可长可短,附图是一个通用型字段。由此也说明,在数据处理中数值类数据只是数据类型中的一种,字符串、日期等都作为某种类型的数据进行处理。本书介绍的 FoxPro 2.5 是关系数据库管理系统,其中的关系称为数据文件(或表),设计单位数据文件和部门数据文件中加入数据后分别构成图 1.7、图 1.8 和图 1.9 的数据文件。用户对该单位的建筑物管理可以抽象成对建筑物数据文件、设计单位数据文件和部门数据文件的管理。

数据管理工作中要对数据文件进行各种操作,这些操作以下列三种操作为基础:选择操作、投影操作和连结操作。

选择操作是选取数据文件中满足给定条件的那些记录,从直观上看,选择是水平地抽取二维表中满足条件的那些行。例如选取用于教学(.T.)的那些记录,就构成图 1.11 的二维表。

建筑物名	建筑日期	造价(美元)	设计单位	教学用	备注	附图
体育馆	01/31/23	200000.00	江南营造所	.T.		
五四楼	01/31/54	250000.00	南工设计院	.T.		
前工院	01/31/29	200000.00	江南营造所	.T.		
中山院	01/31/22	200000.00	江南营造所	.T.		

图 1.11 选择操作

所谓投影操作是抽取文件某几个字段的数据,从直观上看是垂直抽取二维表中某几列的数据。例如将建筑物数据文件在建筑物名和设计单位上投影,就构成图 1.12 的二维表。

建筑物名	设计单位
大礼堂	杨氏营造所
中大院	江南营造所
老图书馆	李氏营造所
新图书馆	南工设计院
中心楼	南工设计院
体育馆	江南营造所
五四楼	南工设计院
前工院	江南营造所
中山院	江南营造所
专家楼	东大设计院

图 1.12 投影操作

为说明连结操作,先考察下面这个例子:找出计算机系所在建筑物的设计单位。由于部门数据文件中只有部门和建筑物的联系,关于建筑物及设计单位的信息存放在建筑物数据文件中。为找出部门所在建筑物的设计单位,就必须同时涉及这两个数据文件,为此,关系数据库提供了连结操作。连结操作可以设想为先将两个文件在“建筑物名”字段的值相等的记录连成一个记录,从而构造成一个更“宽”的二维表,这样就可以找出各部门所在建筑物的设计单位(图 1.13)。

建筑物名	建筑日期	造价(美元)	设计单位	***	名 称	日 期
大礼堂	01/31/31	400000.00	杨氏营造所	***		
中大院	01/31/29	300000.00	江南营造所	***	建筑系	1927
老图书馆	01/31/29	200000.00	李氏营造所	***		
新图书馆	01/31/85	300000.00	南工设计院	***	哲学系	1982
中心楼	01/31/23	500000.00	南工设计院	***	计算机系	1972
中心楼	01/31/23	500000.00	南工设计院	***	外语系	1987
体育馆	01/31/23	200000.00	江南营造所	***		
五四楼	01/31/54	250000.00	南工设计院	***		
前工院	01/31/29	200000.00	江南营造所	***		
中山院	01/31/22	200000.00	江南营造所	***		
专家楼	01/31/85	300000.00	东大设计院	***		

图 1.13 连结操作

前面结合关系数据库讨论了数据模型的两个方面:数据结构及数据操作,下一节还要讨论数据模型的第三个方面——完整性控制。数据结构、数据操作及完整性控制是数据模型的三个要素。

§ 1.5 数据库的保护

数据库是数据管理活动的基础。为了保证数据的安全可靠和正确有效,数据库管理系统必须提供统一的数据保护功能。数据库的保护主要指数据库的安全性、完整性和并发控制。

数据库的安全性是指防止非法用户使用数据库而提供的保护。数据库的安全保密有多种方式,最常见的保护方式是口令。口令可以设定有若干级:在进入数据库管理系统时,系统要求用户应答口令,并由系统进行校核;在进入数据库乃至进入某些敏感数据时,都可设计这种口令的应答,以便确认用户身份。为防止他人窃取口令,用户对口令的应答常被隐蔽起来而不在屏幕上显示。数据库中的数据也可转换成密码形式,以防止通过通讯线路窃取数据,或盗窃数据的物理存储介质。

数据库的完整性是指数据的正确性和相容性。数据库管理系统必须提供一定的功能来保证数据库中的数据满足规定的约束条件。这些条件在数据库中称为完整性约束条件。最常见的完整性约束条件是对数据值规定一定的限制,即对数据库中数据的类型、数据的取值范围及数据精度进行规定。完整性约束条件在系统设计时定义,由数据库管理系统进行控制。例如规定职工的工资额必须是正数,且不得小于 50 元。在用户输入数据或修改数据时,如工资值违反了上述约定,数据库管理系统认为该操作无效,并向用户发出出错信息。

数据库的并发控制是多用户数据库管理系统必须解决的问题。由于数据库是个共享的数据资源,多个用户可能同时对数据库进行存取,这种操作称为并发操作。若对数据库的并发操作不加控制,就会检索或存贮不正确的数据,造成数据库的破坏或用户访问数据的失效。

为说明并发操作对数据的影响,必须首先明确这样一个概念:数据库是存贮在外存(一般是磁盘)上的数据资源,而用户对数据的操作则是内存操作,即用户在读取数据时,首先由系统将外存上的有关数据读到系统为该用户在内存开辟的工作区中,用户对数据的修改也是首先在内存工作区中修改读入的数据副本,在适当的时候再由系统将该副本写回到外存的数据库中。如果有多个用户并发操作数据库,则系统为每个用户在内存中开辟各自独立的工作区,多个用户若同时对同一个数据进行操作,则这个数据值在内存中可能有多个副本。

设想有一个飞机订票系统,某数据文件的字段 seat 存贮了某航班当前的机票余额。两个用户通过不同工作站同时申请预订该航班的一张飞机票。在系统内存中给这两用户各开辟了一个工作区。用户 A 工作中的 seat' 和用户 B 工作区中的 seat'' 均和 seat 对应(图 1.14)。

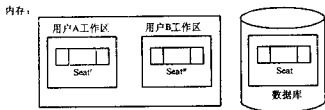


图 1.14 数据库及内存工作区

如果系统将这两个用户的操作分解为一些基本步骤,并按下列调度次序执行:

- ①用户 A 读出该航班机票余额 seat, 放入 seat' 中, 设 seat=6, 则 seat'=6;
- ②用户 B 读出该航班机票余额 seat, 放入 seat'' 中, 所以 seat''=6;
- ③用户 A 预订一张飞机票, 在用户 A 工作区中修改 seat' =seat' -1=5;
- ④用户 B 预订一张飞机票, 在用户 B 工作区中 seat''=seat''-1=5;
- ⑤系统调度将 B 修改后的余额送回数据库中, 所以 seat=5;
- ⑥系统再将 A 修改后的余额送回数据库中, seat=5;

尽管售出了两张飞机票, 但并发操作的结果数据库中仅售出一张飞机票。用户 B 的修改活动被用户 A 的修改所掩盖, 这种情况称为“丢失修改”。丢失修改造成数据的不一致。

为保证数据库的完整性, 必须由数据库管理系统对多用户的并发操作进行一定的限制。由上例可见, 数据库的不一致是由于多个用户对某一个共同感兴趣的数据“同时”修改的结果。换言之, 在某个用户执行修改操作时, 如果数据库管理系统不允许其它用户对同一数据进行修改, 而将这一修改延迟至当前用户修改完成以后进行, 就可以保证数据库的一致性, 即通过适当限制多用户的并发程度而保证数据库的一致性。多用户数据库管理系统必须提供这种控制功能, 由系统自动封锁或由用户封锁数据库的有关区域, 基于这种功能的操作称为加锁操作。

并发控制问题是个相当复杂的问题, 上面仅讨论了封锁的必要性。至于封锁的其它问题, 例如封锁的范围可以是整个数据文件, 也可以是数据文件的个别记录; 封锁的性质可以是写封锁也可以是读封锁, 即在读取数据时用户要求封锁有关区域, 防止其它用户修改操作的干扰, 等等。这些问题准备在第 6 章“多用户功能”中结合 FoxPro 系统作进一步的讨论。

§ 1.6 Windows 3.1 和 FoxPro 2.5 系统简介

本书介绍 FoxPro 2.5 for Windows。

Windows 是美国 Microsoft 公司为个人计算机而设计的操作系统平台。

Windows 提供的是图形化的用户界面, 用户以鼠标作为操作手段在菜单和图标上进行选择从而与计算机交互, 彻底改变了 DOS 操作系统的字符命令行操作方法。

Windows 系统包含了丰富的应用软件, 由于 Windows 的图形环境提供了图标、下拉菜单和对话框三个关键特性, 使得 Windows 下的各种应用程序具有类似的应用界面和学习方法。

Windows 在保护模式下运行, 彻底改革了 DOS 的实模式方式, 从而突破了 DOS 的 640K 内存限制。

Windows 实现了自动内存管理、模块动态连接和多任务处理, 根据程序大小分配适当数量的内存, 也可简单地改变每个虚拟机所分配的时间片。

Windows 的各个应用程序作为任务运行于多任务环境下, 系统提供剪贴板、动态数据交换 (DDE) 及对象链接和嵌入 (OLE) 等多种手段, 使这些程序可以交换及共享信息, 特别通过 OLE 技术, 正文、数字、图形、图象、声音, 甚至活动视频图象都可作为对象在不同应用程序之间进行交换, 从而实现了所谓整合式的工作环境。

Windows 的优点还包括图文输出的“所见即所得”的特性、设备无关技术、True Type 字型等等。

FoxPro 2.5 是一个多用户关系型数据库管理系统。所谓多用户关系型数据库管理系统是指这样一个数据管理的系统软件, 它提供一系列的手段, 使用户能以二维表(关系)的形式存贮

数据与管理数据,而且提供数据保护的有效措施,允许多个用户对数据进行并发操作而不致产生数据的不一致。FoxPro 对数据的描述和操纵的功能均以命令或函数的形式提供给用户,即用户和数据库之间的界面就是一组内容极其丰富的命令和函数。同时,FoxPro 还提供一个功能很强的程序设计语言,所有命令和函数都可以作为这个程序设计语言中的合法的成分,用该程序设计语言可以设计出功能相当复杂的应用系统。

FoxPro 2.5 通过采用 Rushmore 存取技术成为最快的 DBMS,特别是在多数据文件处理和多作业环境中运行时速度优越性更能得到充分体现。从而使得基于 FoxPro 2.5 建立操作数百万条记录的应用系统成为可能。

FoxPro 2.5 是 32 位跨平台运行的产品,通过表格驱动技术,使得基于 FoxPro 开发的应用程序不经修改就可在 Windows、DOS、UNIX 和 Macintosh 环境之间移植,为程序的编码、调试、运行及推广提供了足够的便利。FoxPro 2.5 作为 Windows 运行环境中的一个应用,可利用 Windows 多任务运行的特性及 Windows 的可视化设计方法与其它应用进行动态数据交换 (DDE)及进行数据对象的链接和嵌入 (OLE)。

FoxPro 2.5 具有充分的向下兼容性,可以将绝大多数用 dBASE II Plus、dBASE IV 或 FoxBASE 开发的应用程序当作 Windows 的应用程序来运行,从而保护了用户的软件投资和智力投资。同时,通过 FoxPro 2.5 for Windows 能开发出“纯”Windows 应用程序,从而充分发挥 Windows 环境的威力。

FoxPro 2.5 具有若干第四代语言的特性,提供应用程序生成器、屏幕生成器、报表生成器等生成系统,提供项目管理器等工程化管理系统,直到最后生成 .EXE 执行文件。

由于 FoxPro 2.5 具有多平台一致的特性,尽管本书介绍的是 Windows 上的 FoxPro 2.5,但其内容同样适用于 DOS、Macintosh 及 UNIX 操作系统上的 FoxPro 2.5。

FoxPro 的其它相关产品还包括:

- Library Construction Kit,是一个附加的软件包,允许软件开发者用 C 语言或汇编语言创建 FoxPro 的函数库,该函数库被连入 FoxPro 系统后,其中的函数将作为 FoxPro 的内部函数以供调用。这个软件包还包括最新的 Watcom C 编译器、Watcom 链接器及各种 Watcom 实用程序和 Fox 软件代码,供创建者使用。

- Distribution Kit, FoxPro 的一个附加软件包,它使开发者可以三种方式分销其产品:运行版、单个可执行文件、一个需要链接 FoxPro 库的可执行文件,这些选项都可由开发者根据应用、硬件和软件环境的需要以及其它因素作最优选择。

- FoxGraph, Fox 软件的商用和研究用图形程序,能用于 FoxPro、FoxBase+ 和兼容产品如 dBASE I、dBASE II plus、dBASE IV、Clipper 等以产生全色图和二维、三维图。

- Connectivity Kit,允许开发者存取基于 SQL 端(包括 SQL SERVER)的数据,该产品包括了 MS-DOS 和 Windows 两者都可用的库。

FoxPro 2.5 for Windows 对软、硬件环境的要求(不包括多用户操作系统或计算机网络对硬件的要求):

- 80386SX 及更高档次 CPU
- 不设虚拟内存时需 6M 内存,设置了临时(或永久)虚拟内存后需 4M 内存
- MS-DOS 3.1 或更高版本
- Microsoft Windows 3.1 或更高版本
- 网络操作需要支持 NetBIOS 的网络操作系统,服务器必须有硬盘

- VGA 或更高分辨率的显示器
- 鼠标一只
- 可配置一台打印机

FoxPro 的技术指标:

数据文件和索引文件

• 每数据文件记录数	1,000,000,000
• 每个记录长度	65,000 字节
• 每个记录字段数	225
• 每个字段长度	254 字节
• .DBF 数据文件打开个数	225
• .IDX 索引文件索引码	100 字节
• .CDX 索引文件索引码	240 字节
• 每数据文件打开的索引文件	无限制 *
• 每工作区中打开的索引文件	无限制 *
• 关联个数	无限制 *
• 关联表达式长度	无限制 *

* : 受限于内存大小及 MS-DOS 的文件描述字(handler)个数。

字段

• 字符型字段长	254 字节
• 数值型字段长	20 位
• 数值计算精度	16 位

内存变量

• 内存变量个数	65000
• 数组组个数	65000
• 数组中数组元素个数	65000

程序和过程文件

• 源文件行数	无限制
• 经编译后单个过程(函数)长度	64K
• 每文件含过程数	无限制
• DO 嵌套深度	32
• READ 嵌套深度	5
• 流程控制命令(如 DO WHILE)嵌套深度	64
• 过程的参数个数	24

报表书写能力

• 报表定义对象数	无限制
• 报表定义行数	255
• 分组层次	20

窗口支持

• 打开窗口数	无限制
• 浏览窗口打开数	225

其它

· 每字符串字符数	2G 字节
· 每命令行字符	2048
· 每宏字符长	2048
· 打开文件数	受限于 MS-DOS
· 每键盘宏的击键个数	1024
· SQL SELECT 语句中选择字段数	255

色彩支持

· 每色彩集中色彩模式(scheme)	24
· 色彩集个数	无限制
· 每色彩模式中色彩对数	10
· 用户定义可用的色彩模式	8

FoxPro 使用的文件类型:

- .DBF 数据文件,亦称数据库文件或表。
- .FPT 备注型字段存贮文件。
- .IDX 单项索引码索引文件。为保证 FoxPro 2.5 的向下兼容性,系统保留了这种文件格式。
- .CDX 组合索引文件。组合索引文件利用 Rushmore 技术,效率比 .IDX 索引有很大提高。
- .BMP 位图文件。位图是一种图形的点阵存贮方法,每个位图文件存贮了一幅图形。
- .PRG 程序文件。由 FoxPro 的命令构成的命令文件或由若干个命令文件集成的过程文件。
- .FXP 编译结果的文件。
- .MEM 内存变量文件。内存变量存贮在文件中,可长期保存内存变量的值。
- .FRX 报表文件。
- .LBX 标签文件。
- .SCX 屏幕文件。
- .SPR 屏幕代码文件(编译后生成 .SPX 文件)。
- .MNX 菜单文件。
- .MPR 菜单代码文件(编译后生成 .MPX 文件)。
- .QPR SQL 查询代码(编译后生成 .QPX 文件)。
- .VUE 观察文件,保存系统环境。
- .FMT 格式文件(编译后生成 .PRX 文件)。
- .TXT 文本文件。
- .APP 项目(Project)程序文件,FoxPro 2.5 中编译的项目可执行文件。
- .PJX 项目文件,FoxPro 2.5 中项目的定义及控制信息文件。
- .BIN 汇编语言的二进制目标文件。
- .BAK 后备文件。
- .EXE 执行文件。
- .ERR 出错信息文件。

.TMP 系统工作用临时文件。

同时 FoxPro 2.5 通过数据复制、DDE (动态数据交换) 及 OLE (对象链接和嵌入), 还支持 Microsoft 公司其它产品支持的文件格式。

§ 1.7 FoxPro 系统的组成、安装及启动

FoxPro 2.5 for Windows 以压缩格式存放在七张软盘上。

系统安装分三种类型:

- (1) 正常安装: FoxPro 2.5 for Windows 安装在单机(或网络工作站)上。
- (2) 管理组安装: FoxPro 2.5 for Windows 安装在网络服务器上。
- (3) 工作站安装: 在网络服务器上安装后必须在工作站上执行工作站安装, 以共享网络上的 FoxPro 系统。

安装时首先必须进入 Windows 系统, 在程序管理器的 File 菜单中通过 Run 菜单项, 执行(驱动器): \SETUP, 然后根据系统提示, 逐张插入 FoxPro 2.5 系统盘。

在正常安装情况下, FoxPro 2.5 for Windows 在根目录下建立 FoxProw 子目录, 其文件的目录结构和主要文件作用为:

FOXPROW	FOXAPP	FOXAPP 系统
	LANGUAGE	示例
	TUTORIAL	示例
	SAMPLE	示例
	GOODIES	示例
	READM. TXT	问题及解答
	CONFIG. FPW	系统配置文件
	FOXUSER. DBF	系统资源文件
	FOXUSER. FPT	用户环境信息文件
	FOXHELP. DBF	联机帮助文件
	FOXHELP. FPT	联机帮助备注文件
	FOXPROW. EXE	系统执行文件
	FOXDOC. FLL	文档生成工具
	BEAUTIFY. APP	文档生成工具
	FOXSTART. APP	FOXPRO 启动时执行的应用程序
	GENMENU. PRG	菜单生成工具
	GENSCREEN. PRG	屏幕生成工具
	GENGRAPH. APP	图形生成工具
	GENPD. APP	FOXPRO 的打印驱动程序
	GENXTAB. PRG	FOXPRO 输出 RQBE 结果的程序
	TRANSPRT. PRG	将应用程序从 2.0 转化到 2.5 的工具程序
	SPELLCHK. APP	FOXPRO 的拼音检查程序

FoxPro 的启动和退出:

启动:

进入 Windows 系统后,在程序管理器菜单中,选择 FoxPro 窗口中的 FoxPro for Windows 图标,进入 FoxPro。

退出:

数据库操作完成后,在交互方式下返回 FoxPro 的命令窗口。为退出 FoxPro 系统而返回 Windows 系统,必须执行 QUIT 命令。

QUIT <CR>

如果用关机方式强行退出 FoxPro 就可能破坏当前处于打开状态的文件,造成数据丢失。

在程序执行方式中,程序流程执行完成后,也可以通过 QUIT 语句,退出 FoxPro 系统而直接返回 Windows 系统。

像 Windows 的其它软件包一样,FoxPro 也能用鼠标来完成许多操作。鼠标有三个基本操作:指向(Pointing)、选择(Clicking)和拖曳(Dragging)。鼠标用来控制一特殊光标的位置,这个光标称为鼠标指针。在 FoxPro 中鼠标的指针通常是指向斜上方的小箭头。

为了用鼠标指向一个目标,只要向目标方向移动鼠标,这时鼠标指针也相应地往这个方向移动,选择(Clicking)的动作有两种:一种按鼠标左键一次,另一种是连续击鼠标左键两次。本书分别称这两种动作为单击鼠标及双击鼠标。指针指向一目标并选择相应功能后,就可执行 FoxPro 提供的相应操作。拖曳(Dragging)是指在移动鼠标时一直按住左按钮,这种操作常在选择 FoxPro 菜单的菜单项时使用。

在使用鼠标前必须安装鼠标驱动程序,关于如何安装鼠标驱动程序,请参阅 Windows 的安装说明。

§ 1.8 FoxPro 的联机帮助

FoxPro 2.5 提供了联机帮助手段。联机帮助实际上是用联机的方式提供了一本详尽的用户手册,对熟悉 MS-DOS 的用户,可以选择 DOS 型的帮助;对熟悉 Windows 系统的用户可以选择 Windows 型的帮助。联机帮助中包括各个主题详细的说明及有关使用范例,还包括对主题之间的相互参照查询。

在 Windows 型的帮助方式下,无论何时都可以通过 F1 键显示有关当前窗口或对话框的帮助信息,若想等到有关菜单的菜单项的帮助,先按 Shift+F1 键,然后选择该菜单的菜单项;如果要找到与某一单词或短语相关主题的帮助信息,在 HELP 菜单中选择 Search for Help on ...

在 HELP 窗口中,通过 BookMark 菜单项可以定义当前主题的书签名,从而能更快地访问该主题。Print Topic 可以将帮助主题打印出来;Edit 菜单中 Copy 可以将有关帮助的内容复制到 Windows 的剪辑板中,剪辑板中的内容可以粘贴到用户的任何一个文件中,包括将帮助信息中的使用范例剪辑和粘贴到用户的程序文件中。

FoxPro 系统还提供了求助命令 HELP。

语法:HELP [<主题>]

[IN[WINDOW]<窗口名>|IN SCREEN]

[NOWAIT]

执行 HELP 命令相当于按 F1 键。命令中的其它选择项可以指定求助的主题,可以指定系统对该主题的帮助信息的显示窗口等等。

第2章 FoxPro 的基本命令 (一)

FoxPro 有两种执行方式,一种是命令执行方式,用户通过计算机逐条输入命令,系统解释执行每条命令,然后将该命令的执行结果(或出错信息)返回给用户。在命令执行方式下,FoxPro 在屏幕上开辟有两个窗口,FoxPro 的主窗口及覆盖在主窗口上的一个命令窗口(Command 窗口)。用户在 Command 窗口中输入命令,FoxPro 的执行结果显示在主窗口中。为满足排版需要,本书中命令和命令的执行结果顺序排列,而未组织成分窗口的形式。另一种方式是程序方式,用户将要执行的一系列命令组织在程序中,由系统完成该程序中规定的一系列动作。命令方式是一种人与计算机的对话方式,用户输入的每条命令计算机都会立即做出响应。所以本书首先以命令方式作为学习 FoxPro 的入门。

本章讨论 FoxPro 命令方式中的最基本的命令。首先讨论命令的语法。

§ 2.1 命令、表达式和运算符

FoxPro 的每条命令都要求计算机完成一个确定的动作,所以每条命令都有确定的格式。假设与建设物有关的信息已经组织在 building.dbf 文件中,为寻找造价满足某种条件的楼房可用定位命令。定位命令的语法格式是:

LOCATE [(<范围>)] [FOR <逻辑表达式>] [WHILE <逻辑表达式>]

命令的主体是一个命令动词,命令中还可能包含若干选择项,本书将由单个单词构成的选择项称为短语,一般选择项称为子句。在 LOCATE 命令中,LOCATE (<定位>)是标识该命令的动词,<范围>短语在数据文件中限定处理范围,FOR/WHILE 子句中的(<逻辑表达式>)确定被定位的记录必须满足的条件。

为找到造价(buil_price)在 30 万美元至 40 万美元之间的所有楼房,定位命令具体化为:

LOCATE ALL FOR buil_price>=300000 .AND. buil_price<=400000

命令语法格式中各符号的含义如下:

大写:英文的大写形式表示 FoxPro 的关键字,命令的主体动词、子句的主体词和短语等都是关键词,关键词是 FoxPro 系统规定且由 FoxPro 系统使用的,所以关键词又称保留字。用户实际操作中,不一定输入关键词的全部字母,只要输入开始的几个字母(至少 4 个)即可,但这几个字母应能将输入的关键词与 FoxPro 的其它关键词有所区别(例 LOCATE 只要输入 LOCA)。

< :命令中的用户定义项

[] :可选菜单项,视具体使用要求由用户选择

| :前后两项只择其一

... :表示命令中的一部分可以类似方法重复多次

表达式:表达式是操作数(变量、常量及函数)与运算符的结合。表达式经运算产生一个结果,根据结果值的类型,表达式可再分为字符表达式、数值表达式、逻辑表达式及日期表达式。多个表达式构成表达式表,表达式表中的各表达式应使用逗号分隔

在 FoxPro 中还有一种称为名表达式的特殊表达式,关于名表达式的用法见本书第 4 章。
范围:在数据文件中限定处理的范围,有四种范围选择:

- ALL 数据文件的所有记录
- NEXT (n) 数据文件从当前记录开始的连续 n 个记录
- RECORD (n) 数据文件的第 n 号记录
- REST 从当前记录开始到文件结束的所有记录

提供的运算符包括:

运算符			
类别	符号	说明	优先级
算术运算符	()	括号	高 ↑ 低
	+, -	正、负号	
	** 或 ^	乘幂	
	*, /, %, +, -	乘除、求余、加、减	
	+, -	加、减	
字符串运算符	+	字符串联接	同类优先级相等
	-	字符串联接	
关系运算符	<	小于	同类优先级相等
	>	大于	
	=	等于	
	<> 或 !=	不等于	
	<=	小于等于	
	>=	大于等于	
逻辑运算符	==	精确等于	低
	()	括号	
	.NOT. 或 !	逻辑非	
	.AND.	逻辑与	
逻辑运算符	.OR.	逻辑或	低

在一个表达式中可以使用多个运算符,运算从左到右根据运算符优先级的高低依次进行,各类运算符之间的优先关系如上表长箭头所示;同类运算符中,优先关系如短箭头所示。在构造表达式时可用圆括号重新组合运算次序,圆括号层数不限。

命令行可行书写,行末用“;”作为当前行未结束的续行符。命令的关键字可用大写也可用小写。本书为了清楚起见,关键字一律用大写,用户定义名(如数据文件名等)用小写。

命令格式中其它的一些约定将结合具体命令逐步介绍。

§ 2.2 数据文件的创建

FoxPro 2.5 是一个数据库管理系统,数据管理功能主要体现在对数据文件的管理。管理一个数据文件的第一步是创建该文件的结构,这项工作由 CREATE 命令完成。

语法:CREATE [(文件)]?]

每个数据文件有一个文件名。系统规定每个文件名由长度不超过 8 个字符的字母、数字及下划线字符组成。文件最好取与文件内容有关的名字以助记忆,各数据文件不能重名。计算机系统中有多种类型的文件存在,文件类型由文件扩展名标明,数据文件的扩展名为 .DBF,该扩展名由系统自动生成。