

数据库
程序设计

●刘友霖

●张敏华

●东南大学出版社

前 言

本书介绍关系数据库原理及其程序设计方法。虽然 FoxPro 提供了强有力的可视化编程工具,但并不能完成所有的编程工作,一些较为复杂的功能,仍需通过人工编写的程序才能实现。再则,我们具有了良好的编程技术,也可充分发挥和利用 FoxPro 自身潜在的功能,从而提高应用程序的质量、运行效率和速度。通过编程还可加深对数据库基本原理的理解。

本书原理与应用兼顾,以应用为主。原理部分,深入浅出、简明扼要;应用部分融入了作者多年的数据库编程经验和教学经验,以实例为主线阐述原理及数据库的操作。书中实例丰富,通俗易懂,所有实例均可上机运行,便于读者实际操作。对实例在描述其功能的基础上,说明所采用的算法和实现的技巧,阐明相关的基本概念和基本方法,并在程序的关键处均加注释,而后再进行原理性或引伸性的讨论。在解决某个问题时,我们常给出多种实现方法,并加以比较,以培养读者灵活应用和触类旁通的应变能力。

本书第 1 章为关系数据库引论,第 2 章至第 5 章着重讨论关系数据库的基本概念。第 6 章和第 7 章介绍关系数据库的程序设计方法。第 8 章至第 10 章介绍可视化编程。为了便于教学,还附有四个附录,每章后附有习题。学习顺序建议为:第 2 章、第 3 章、第 4 章、第 5 章、第 6 章;然后,第 7 章、第 8 章、第 9 章、第 10 章结合起来学(第 7.1 节与第 8 章,第 7.2 节与第 9 章,第 7.3 节与第 10.3 节,第 7.4 节与第 10.2 节);最后学习第 1 章。

本书第 6 章、第 7 章由张敏华编写,其余部分由刘友霖完成,王雪参加了第 1 章的修改。

在本书的编写过程中,得到了南京航空航天大学林均海教授的热诚帮助,林教授审阅了全稿并提出了修改意见和建议。南京邮电学院梅杓春副教授也给予编者不少帮助,同时还得到了南京邮电学院教务处、计算机系及计算中心领导的关心和支持。作者在此表示诚挚的感谢。

限于水平与时间,书中难免存在缺点和不妥之处,恳请广大读者批评指正。

编 者

1996.

目 录

1 关系数据库引论	
1.1 信息、数据和数据处理	(1)
1.2 数据库系统	(4)
1.3 关系数据库基础	(11)
1.4 关系数据库的规范化	(14)
习题	(20)
2 数据库	
2.1 数据库文件组成	(21)
2.2 数据库文件的建立	(24)
2.3 数据库文件结构的修改	(26)
2.4 数据库文件的显示	(27)
2.5 数据库文件操作	(31)
习题	(41)
3 数据库记录	
3.1 数据库文件记录的操作	(45)
3.2 数据库文件记录的编辑	(50)
3.3 数据库文件数字字段运算	(55)
3.4 排序文件	(60)
习题	(64)
4 数据库查询	
4.1 顺序查询程序	(66)
4.2 索引文件	(69)
4.3 索引查询	(76)
4.4 模糊查询	(80)
习题	(84)
5 多个数据库文件操作	
5.1 多区数据库文件操作	(88)
5.2 数据库文件的关联操作	(92)
5.3 两个数据库文件之间的记录更新	(93)
5.4 两个数据库文件的连接	(95)
习题	(96)

6 程序设计初步	
6.1 程序设计概述	(98)
6.2 一般程序设计	(100)
6.3 子程序调用	(106)
6.4 库操作程序	(109)
6.5 模块程序设计	(121)
习题	(128)
7 程序设计	
7.1 菜单程序设计	(131)
7.2 数据输入程序	(146)
7.3 查询、修改程序	(159)
7.4 打印、维护程序	(185)
习题	(197)
8 菜单生成器	
8.1 概述	(198)
8.2 直接更改 FoxPro 的系统菜单	(201)
8.3 在 FoxPro 的系统菜单上建立菜单	(205)
8.4 FoxPro 系统菜单程序	(209)
习题	(215)
9 屏幕生成器	
9.1 屏幕菜单选项	(217)
9.2 屏幕设计工具	(222)
9.3 产生屏幕程序	(231)
习题	(232)
10 其它开发工具	
10.1 项目管理器	(233)
10.2 报表生成器	(237)
10.3 关系样本查询(RQBE)	(242)
习题	(248)
附录	
A FoxBASE+ 系统概述	(249)
B FoxBASE 命令一览表	(251)
C FoxBASE 函数一览表	(255)
D 上机实习提要	(257)
参考文献	
.....	(264)

1 关系数据库引论

1.1 信息、数据和数据处理

1) 信息与数据

什么是信息?众说不一,诸如信息是消息,信息是通知,信息是情报、资料、报导和知识。信息是能够带来新内容和新知识的消息。信息是物质和能量在空间和时间中分布的不均匀程度。哈特莱认为,消息、情报、信号、语言等都是信息的“载体”,而信息是它们“载荷着的内容”。C.E. 香农认为,信息是人们对事物了解的不确定性的消除或减少。信息的特点是信息具有可识别性、可转换性、可传递性、可处理性和可存储性。

信息是通过某种形式来表示和传播的,这就是数据。数据是描述客观事物的数字、字母和符号,以及所有能输入计算机(如图像、声音)并被计算机程序加工处理的符号集合。信息和数据都是客观世界的反映,数据反映的是“特定目的尚未做出评价的事实”,而信息是经过加工处理后产生的有助于实现特定目的的数据,信息是消化了的数据。信息依赖于数据而存在,数据组合具体地表现信息。所以,信息与数据既相关又相异。

2) 数据处理

在这里信息处理也称数据处理,最初指在计算机上对商业、企业的信息和数据进行加工,现在常用来泛指计算机在非科技工程方面的所有计算、管理和各种应用。原始数据经过计算机处理(按一定目的、步骤对数据进行处理),转换成有用的信息输出的过程称为信息处理。大量数据在存储器内如何组织、存放,如何检索、分类,如何输入、输出、传输均是数据处理讨论的问题。从数据处理的发展过程看,信息数据处理大致可以分为三个阶段:手工处理阶段、机械处理阶段和电子数据处理阶段。用计算机进行数据处理不仅处理速度快,处理容量大,输入、输出灵活方便,而且把手工操作降低到最小程度。计算机的应用是数据处理领域中的一场革命。随着光盘的出现,声音、图像识别技术的实用化,计算机已经具有综合处理声音、图形、图像、文字以及电视图像的能力,即所谓的多媒体技术。计算机处理信息,起初是数值、文字处理,随后是图形处理,目前是综合数值、文字、图形、动静图像和声音多维信息处理,即将图形、图像、文本、电子数字表、符号数据、声音等混合集成、匹配、排序、演示。这种所谓的人类化信息处理,是多种信息表现形式的集合。其实质是把以自然形式存在的各种信息数字化,再利用计算机对这些数字化的信息进行处理,以一种更符合人类习惯、更容易接受的形式提供给信息使用者。多媒体不仅是信息集成,也是设备集成和软件集成,它们通过逻辑连接形成有机整体,又可以实现交互控制。

3) 数据管理

数据管理是数据处理中的重要和不可分割的部分,它贯穿于数据处理的每一步骤中。

数据管理技术与数据处理方式有密切的关系,并且直接影响着数据处理的效率。数据管理的基本内容是数据收集、数据转换、数据存储、分组和排序、数据筛选、数据检索和数据输出。从数据存储结构 and 处理方式的角度可把计算机数据管理技术的进展分为四个阶段:自由处理阶段、独立文件阶段、共享文件阶段和数据库方法阶段。

① 自由处理阶段,是一组数据对应于一个程序,数据和程序紧密地结合为一个整体。由于当时没有软件工具可以利用,因此程序涉及数据组织和存储的有关问题,如数据的存储结构、存储方式和布局、输入/输出格式等。

② 独立文件阶段,是一组数据按规则组成数据文件,应用程序通过文件管理系统(专门程序)对数据文件进行管理和维护,即应用程序通过文件管理系统建立和存储文件,存取文件中的数据。文件管理系统是应用程序与数据文件之间的接口,数据管理由文件管理系统实行统一管理。用户在应用程序设计时只需考虑数据的逻辑结构和特征,并按规定的组织方式和存储方法,建立并使用相应的数据文件,而较少考虑数据物理存储方面的各种问题,因而简化了应用程序。但应用程序与一组数据文件还成一一对应关系,相互依赖关系。

③ 共享文件阶段,是建立在独立文件基础之上。在一组应用程序和文件管理系统之间插入一个专用程序,任意一个应用程序均可通过专用程序去访问任意一个数据文件,达到数据共享的目的。

④ 数据库方法阶段,是对所有的数据实行统一的、集中的、独立的管理,使数据的存储不依赖于使用数据的应用程序,为数据提供了一种完善的、高级的管理方式(数据库方法见下节讨论)。

4) 信息结构

信息和数据都是客观世界的反映,那么信息怎样进入到计算机和数据库中呢?这里有一个转化过程,这个过程一般地说有三个阶段,也称为三个世界:现实世界、信息世界和计算机世界。在数据处理中,现实世界的事物反映到人脑中,人的脑子对这些事物有个认识过程,经过选择、命名、分类后进入信息世界,在信息世界中将信息经过加工编码进入计算机世界。

① 现实世界,是客观存在的世界,存在千差万别的各种各样的事物,而且每个事物都有一些特征,特征是区分事物的依据。事物又是切切相关的,事物之间是有关联的。

② 信息世界,将现实世界中的事物抽象成实体,事物间的联系也抽象成实体。事物的特征抽象为属性,由属性集来描述实体。属性的值由属性域决定。同类型的实体集合称为实体集。实体集之间的联系可以是实体集中的一个实体也可以是多个实体间的联系。这种联系实际上反映了实体集间的函数关系。这种函数关系可以有:一对一(1:1);一对多(1:M);多对一(N:1);多对多(N:M)的联系。一个学校的校长与学校是1:1联系,校长与教师是1:M联系,学生与教师是N:M的联系。这种联系可以用实体模型(E-R图)表示出来。

③ 计算机世界,是现实世界中的客观现象在计算机中的表现形式,它以数据形式出现,故又称数据世界。数据是计算机世界的对象。数据在计算机内是按文件形式组织的,文件由记录组成,记录由数据项组成。数据项对应于信息世界的属性,对应于现实世界的特征。记录对应于信息世界的实体,文件对应于信息世界的实体集联系。数据项、记录、文

件是计算机世界中的基本的数据存取结构。

5) 数据模型

数据模型是一种描述数据、数据之间的联系以及有关语义约束规则的方法,是模拟和处理信息所采用的模型。E-R 数据模型是 1976 年由美籍华人 P. P. Chen 提出来的,相应数据模式用 E-R 图表示。它已成为软件工程的一个重要设计方法,现简单介绍如下:

(1) 实体的表示

用长方形框表示实体型,框内标注实体名。如实体型教师、学生、课程可表示为图 1.1。

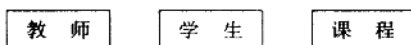


图 1.1 实体

(2) 属性的表示

用椭圆形圈表示属性,属性名写在圈内。属性是依赖于实体的,两者之间用实线连接。如实体型教师有属性职工号(T#)、姓名(TN)、系别(TD)、教研室(TG),则可表示为图 1.2(a)。又如,实体型学生有属性学号(S#)、姓名(SN)、年龄(SA)、系别(SD),表示为图 1.2(b)。课程表示于图 1.2(c)中,C#、CN、PC#、G 分别为课程编号、课程名称、先修课程号、学时数。

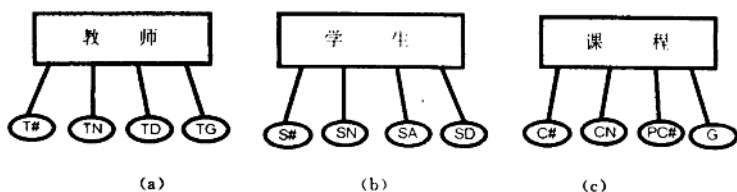


图 1.2 属性

(3) 联系的表示

联系用菱形表示,联系名写在菱形框内,并用无向的实线将菱形与有关的实体相连接,在实线旁标注联系类型或函数对应关系。图 1.3 中表示学生、课程、教师间相互的联系。学生与课程之间有听课的联系,且每门课程可以有若干个学生听课,所以联系类型为 1:M。教师与课程间有讲授的联系,教师与学生间有教与学的联系,这些联系的类型均为 N:M。

用 E-R 图表示的数据模式,需要进行适当的加工才能得到计算机所能够实现的数据模式。目前有三种数据库类型,对应于三种数据模型:层次数据库对应于层次模型,网络数据库对应于网络模型,关系数据库对应于关系模型。关系模型是一种以集合论中的关系概念为基础发展起来的数据模型。关系模型是把一个实体集看成一个二维表,这个表称为关系。如表 1.1 是表示关系 R 的二维表。

表中行称为元组,表中列称为属性。属性的个数是关系的元(或度),表 1.1 中有元组 $(a_1, a_2, a_3, \dots, a_n)$ 或 $(b_1, b_2, b_3, \dots, b_n)$ 等等,关系属于元组的集合,关系中无重复元组。属性的变化范围称为域,域可以有理论上的域与实际上的域,实际上的域是理论上域的子集。表 1.1 中 R 是关系名, A_i 是属性名,关系 R 可记为 $R(A_1, A_2, A_3, \dots, A_n)$ 。

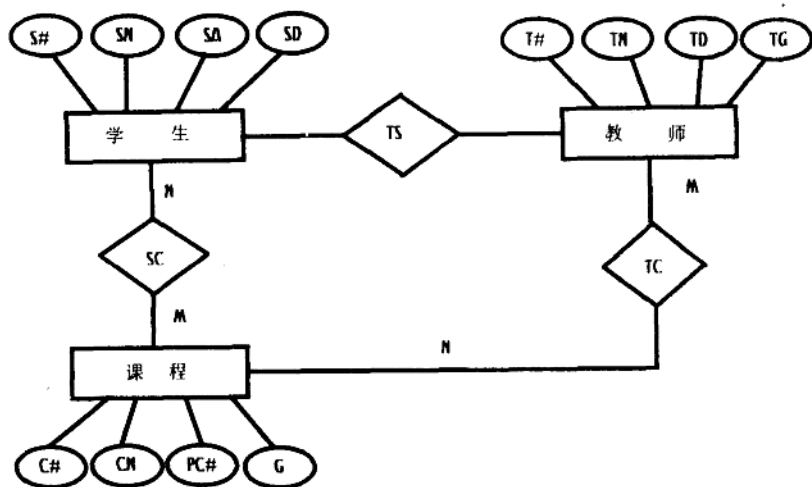


图 1.3 E-R 图

表 1.1 关系 R 的二维表

A_1	A_2	A_3	...	A_n
a_1	a_2	a_3	...	a_n
b_1	b_2	b_3	...	b_n
$\vdots$	$\vdots$	$\vdots$	...	$\vdots$
x_1	x_2	x_3	...	x_n

关系模型的结构方式具有很多优点,例如:

- ① 关系模型的基本结构是二维表,其数据结构简单,易于计算机的实现;
- ② 二维表的表达能力强,不仅能表示实体(以及属性),也能表示联系;
- ③ 由于 E-R 图能用二维表表示,因此二维表的方法不存在 E-R 图的分解问题;
- ④ 二维表有很系统的数学基础——关系理论,研究关系模型具有很好的数学工具(见 1.3 节)。

1.2 数据库系统

1) 数据库

数据库是在计算机存储设备上合理存放的相互关联的数据集合,这些数据具有独立性和最小冗余度,能为不同应用程序所共享,并由一个软件统一管理。数据库的特征有 5 个:

(1) 数据的独立性

在数据处理中,数据库是在传统的文件系统基础上发展起来的,但是数据库中的数据与应用程序之间不存在相互依赖关系。也就是说,数据的逻辑结构、存储结构和存取方法

等,不因应用程序的修改而修改,反之亦然。数据独立性通常分为两级:第一级为物理独立性,数据库的物理性质发生改变时(如更换存储设备、改变文件组织方式和存取策略等),不影响数据的逻辑结构,也不引起应用程序的修改;第二级是逻辑独立性,这是指数据库总体逻辑结构的改变(如修改数据的定义、增加新的数据类型、改变数据间的联系等)无须修改原来的应用程序,同样当用户需要修改应用程序时,也不强求数据结构作相应的改变。由此可见,数据独立性就是数据与应用程序之间的互不依存性,我们把一个满足数据独立性要求的系统,称之为“以数据为中心”的系统。

一个理想的数据库系统应该做到应用程序和数据库结构完全独立,以使得数据资源共享真正成为可能。但是,到目前为止,数据的逻辑独立性尚未能完全地实现,因为要达到这样的目标,将使数据库管理系统的设计变得十分复杂,设计难度和费用增加。因此,实际的数据库系统往往是具有一定程度的数据独立性。

(2) 最小的数据冗余度

最小冗余度是指存储在数据库中的数据的重复尽可能减少。在非数据库系统中,每个应用程序有它自己的数据文件,从而造成存储数据的大量重复,这给数据管理带来下列弊病:①占用了计算机存储设备的大量存储空间;②为了更新某些冗余副本,保持数据的一致性,必须执行多次更新操作,增加不必要的操作时间,不如此则会造成数据的不一致性;③由于数据的不同副本可能处于不同的更新阶段,从而可能给出数据不一致的信息。在数据库系统方式下,则是从整体观点来组织和存储数据的,数据是集成化、结构化的,数据统一存放、集中控制,因而能减少数据的冗余。必须指出,在数据库系统方式下,冗余度并不能完全消除。在实际应用中,有时为了某种原因而使用同一数据,在一定控制条件下多次存储还是必要的。

(3) 最大的数据共享性

数据共享是促成数据库技术发展的重要原因之一,也是数据库最本质的特性。数据共享性可在下面 4 方面得到体现:

① 不同的应用程序可以使用同一个数据库。一个特定的应用总是只存取数据库中的一个数据子集;不同的应用程序对应的数据子集可以相互重叠。

② 不同的应用程序可以在同一时刻去存取同一数据,这在数据库中称作“并发控制”使用。

③ 数据库中的数据不但可供现有的各个应用程序共享,还可以为新开发的应用程序使用,实现新、老应用程序共享数据库中的数据。

④ 应用程序可以用不同的程序设计语言编写,它们可以访问同一数据库。终端命令语言是常用的一种语言,一个商品化的 DBMS 数据库管理系统通常提供了与多种程序设计语言的接口。

(4) 统一管理和控制

将若干不同性质的数据文件统一化,实现数据集成化。数据集成化又称数据结构化,它要求数据库按照一定的数据模型(在下一章中介绍)来组织和存储数据。结构化的数据不但反映了数据之间的自然联系,而且可实现对数据的集中控制和管理。数据结构化实现了数据资源共享、减少冗余等诸方面的要求,同时这种工作方式也带来了文件系统所不具备的一系列特性,如数据的安全性、完整性等。下面,对 5 个概念作一简要说明:

① 安全性保护 数据的安全性主要指数据保密,防止数据的不合法使用。上面提到的数据资源共享并不意味着向一切用户开放所有的数据资源,如对于一个应用用户,数据库系统将对他的权限进行安全性检查,只有合法的用户才能以规定的方式(如检索、修改等)对允许他使用的那一部分数据进行相应的操作。

② 并发控制 并发控制是指不同应用(应用程序、用户终端)在同一时刻期望存取同一数据时发生的事件。例如,在铁路或航空售票系统中,设有多个售票点分别为 X、Y、Z、…。若数据库中有某车次(或航班)余票 n 为 15,当 X、Y 两点同时售票时,他们可以同时检索到有余票 n ,X 点按需要售出 10 张,将修改余票 n 为 5。但尚未将新值送入数据库时,Y 点又按 n 为 15 的情况售出 8 张,此时该售票业务必然出现差错,送入的 n 也成为不可预期的值,从而使 n 值失去了完整性。为了保证数据库的准确性,数据库管理系统的并发控制设施必须采用相应的封锁技术来避免上述错误的发生。

③ 完整性 数据的完整性也就是数据的正确性。数据的不一致性是失去完整性的例子。前已述及数据冗余可能引起数据的不完整性,但是没有冗余的数据项同样可能出现不正确的值。数据库管理系统的集中控制可以避免此类情况出现,它通过 DBA 数据库管理员定义完整性检查,对每一次更新的数据实施相应的检查,保证数据的完整性。必须指出,数据完整性检查对于多用户数据库系统尤为重要,因为多个用户同时使用数据所引起对数据的干扰是经常大量发生的,必须在这种情况下对数据的完整性进行必要的维护。

④ 故障的发现和恢复正常状态 数据库系统在运行过程中,由于硬件/软件的故障及用户操作的失误,随时都有可能使数据库遭到局部性的或全局性的破坏,其所涉及的数据往往是无法用书面复原的。数据库系统应提供一套设施和方法,来发现由于上述各种因素引起的故障,并迅速地把数据库恢复到故障以前的正确状态,使系统正常运行。

⑤ 可修改和可扩充性 一个实用数据库系统建立之后,需要逐步扩充。因此,要求整个数据库系统在结构和组织技术上具有易于修改和扩充的特点。

(5) 可按标准化规定实施数据库系统

随着信息处理自动化技术的发展,信息的标准化越来越得到有关方面的重视。各个团体、行业、部门、国家和国际的标准正在不断制定产生。标准化不仅能够简化信息管理的过程、提高处理速度,而且使信息更具有通用性,能在更广泛的范围内实现通信。不论是在一个组织内部的还是国际性的标准化,都能使数据库系统内集中控制的实施成为可能。当然标准化的程度和层次越高,系统的实用范围越广泛。在一定程度上,数据库技术和标准化技术是相互促进着发展的。

2) 数据模式

美国国家标准局的 ANSI/X3/SPARC 报告把数据模式分为内部模式(简称内模式)、概念模式和外部模式(或称外模式)三级,如图 1.4 所示。数据模式是用给定数据模型对具体数据的描述,反映一个单位的各种事物的结构、属性、联系和约束,是用数据模型对一个单位的模拟。

下面对每一级作简单介绍:

(1) 概念模式

从某个角度看到的数据特征称为数据视图。概念模式涉及到所有用户的数据定义,也就是全局逻辑视图,称为概念视图。概念视图是由若干概念记录型组成。概念视图用数据

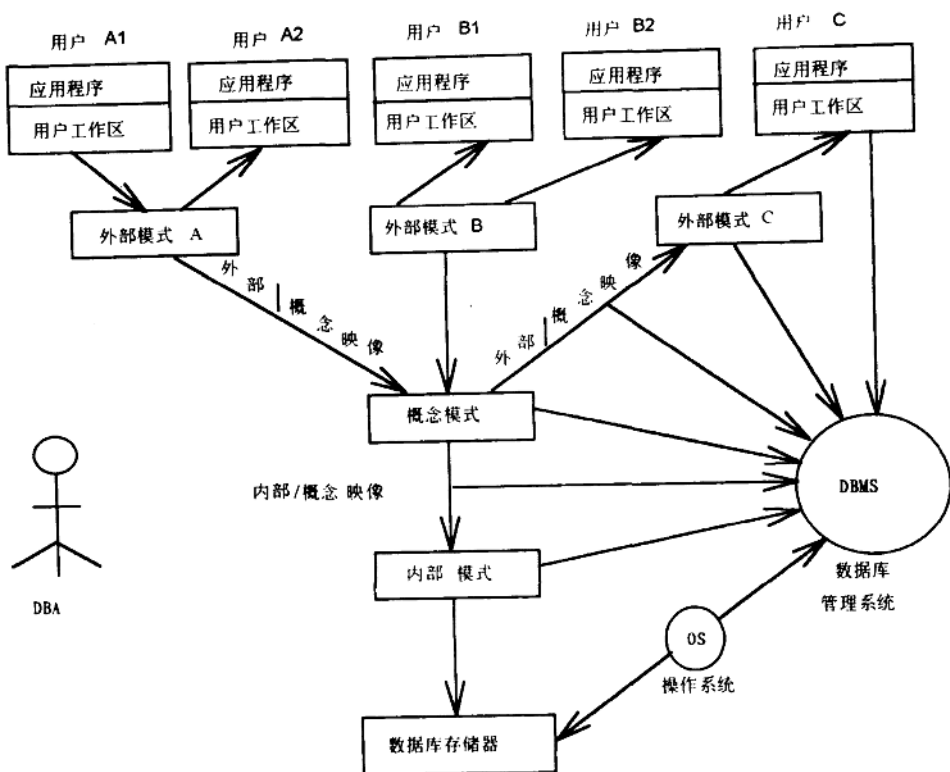


图 1.4 数据库系统三级模式

描述语言 (DDL—Data Description Language) 描述后得到的是概念模式 (又称模式), 描述概念视图的数据定义语言称为“模式 DDL”。

(2) 外模式

外模式是单个用户视图, 称为外部视图。外部视图是由若干外部记录型组成。这种结构用“外模式 DDL”描述后得到的是外模式 (又称子模式)。外模式是外部视图对每个记录型的定义, 因此它是面向一个或多个用户需要的那部分数据的描述。外部记录是用户存取数据库数据的基本存取单位, 对于一个数据库, 只能有一个概念模式, 但可以有多个外模式。外模式是概念模式的子集, 允许多个用户或应用程序共享一个外模式。外模式是用户与数据库系统的接口。

(3) 内模式

内部模式是用物理数据模型对数据的描述。内模式对一般用户是透明的, 它的设计直接影响数据库的性能。内模式是用“内模式 DDL”描述后得到的存储模式。内模式要定义所有的内部记录型, 定义一些索引和数据在存储器的存放位置等。一个概念记录, 可以对应内模式中几个不同类型的内部记录。所以, 内模式是数据在物理存储方面的描述。比内模式更接近于物理存储的软件是文件系统中的文件系统, 例如对磁盘进行的 I/O 操作。

用户对外模式进行操作时, 外模式/概念模式的变换定义了某一外模式和概念模式之间的对应关系。当概念模式改变时, 外模式/概念模式变换要做相应的改变, 以保证与此改

变无关的外模式保持不变。概念模式/内模式变换是定义概念模式和内模式之间的对应关系。当内模式改变时,概念模式/内模式变换也必须作相应的修改,使得概念模式保持不变。正是这二级变换功能连接了数据库的三级结构,数据库管理系统的一项重要工作就是要实现这二级变换功能。

应当注意的是数据库的三种模式只是数据库的一种结构描述,并不是数据库本身,它是一个可以按照给定的某种约束装入具体的数据值的框架,而按这些框架填入的数据才是数据库的内容。通常数据库的内容随时间变化,而模式则是相对稳定的。

3) 数据库语言

(1) 数据描述语言

数据库中的数据是按一定的结构集成的,数据描述语言 DDL 是定义和描述结构的工具,它可以正确地定义和描述数据库中的数据及数据之间的联系,以便数据库管理系统 DBMS 根据这些定义和描述,从内部的存储记录导出整体的概念级记录,又从整体的概念级记录导出应用程序所需要的外部记录。或者相反,从应用程序所需要的外部记录推知其所属的整体的概念级记录及对应的内部存储记录。

由于数据库系统在结构上分为三级,对一个数据库的结构描述也应当分为子模式、模式和存储模式三级,但目前并不是所有的 DBMS 都能独立地提供这三种不同的数据描述语言,往往是把三级描述的语言结合在一起,统称为 DDL。

(2) 数据操纵语言

数据操纵语言(DML—Data Manipulation Language)是用户的应用程序设计语言与 DBMS 之间的接口,它是 DBMS 向用户提供的一组宏指令或调用语句,用户可以使用 DML 向数据库管理系统提出对数据库中数据进行各种操作的请求,如:

① 从数据库中检索满足给定条件的数据,它可以是一个或几个数据项,一个记录或一组记录;

② 把新的数据插入到数据库中的指定位置上;

③ 删除数据库中没有必要保存的原有数据和记录;

④ 更新某些已经发生变化的数据项的值,以便正确反映当前的数据状态。

DML 一般具有操作表达明确,功能齐全,使用方便的特点。

DDL 和 DML 分别说明数据库对象的定义和操纵,它们是数据库的用户设计应用程序时所用的程序设计语言的一个子集,称为数据子语言(DSL)。

4) 数据库管理系统

(1) DBMS 是一种负责数据库定义、建立、操纵、管理和维护的软件系统。其任务是很明确的,即保证数据安全可靠、提高数据库应用的简明性和方便性。一个功能完善的 DBMS 一般应具有如下 5 个功能:

① 定义和建立数据库 具有定义数据库三级视图的能力,并提供相应的工具以便描述三级模式,把各种源模式翻译成计算机目标代码形式,把这些关于数据库中数据的描述信息保存在数据字典之中,并把实际数据库存储到物理存储设备上,完成目标数据库的建立工作;

② 数据库的操纵 是直接面向用户(应用程序)的功能,即接收、分析和执行应用程序对数据库提出的各种操作请求,完成对数据库的检索、插入、删除和修改等操作;

③ 数据库的控制 执行对访问数据库时的安全性、完整性检查和数据共享的并发控制,以保证数据库的可靠性和可用性;

④ 数据库的维护 包括运行时记录工作日志和评价分析数据库的运行性能,为了确保数据库的运行效率而对数据库进行的重新定义和重新组织等;

⑤ 故障恢复 当数据库系统在运行中发生故障时,DBMS 提供相应的恢复功能,把故障造成的影响限制在最小范围,并能以最快的速度排除故障,恢复和重新启动系统进入正常运行状态。

(2) 数据库管理系统通常由三部分组成:语言编译程序系统、系统运行控制程序和服务性程序。

① 语言编译程序

▲ 数据库各级模式 DDL 的编译程序,其作用是将各级源模式编译成各级目标模式,存于目标库中,以供系统随时调用。

▲ DML 编译程序,将应用程序中 DML 语句转换成可执行的程序。在处理时有两种不同的方式:一种方式是对 DML 语句进行预处理,将 DML 语句转换成主语言编译程序能处理的形式(一般以 CALL 调用方式表示);另一种方式是扩充主语言编译程序的功能,使主语言编译程序能处理 DML 语句的编译。

▲ 终端命令解释程序,解释终端命令的意义,决定操作的执行过程。

▲ 数据库控制命令的解释程序,解释每个控制命令的意义,决定怎样执行。

② 系统运行控制程序

▲ 系统总控程序,是 DBMS 运行程序的核心,控制和协调程序的活动;

▲ 存取控制程序,用于检查用户(或应用程序)是否合法;

▲ 并发控制程序,协调各个应用程序对数据库的操作,保证数据库数据的一致性;

▲ 保密性控制程序,实现对数据库数据安全保密控制;

▲ 数据完整性控制程序,检查完整性约束条件,决定是否执行对数据库操作;

▲ 数据存取和更新程序,实施对数据库数据的检索、插入、修改、删除的操作;

▲ 通讯控制程序,实现用户程序与 DBMS 之间的通信。

③ 服务性程序

数据库的日常管理所需的一些程序如下:

▲ 数据装入程序,实现将初始数据装入数据库;

▲ 数据库恢复程序,利用恢复程序将数据库由于软件、硬件的失败而引起的数据库的破坏恢复到正确状态;

▲ 工作日志程序,负责记载对数据库每一记录更新前和更新后的数据;

▲ 性能监测程序,监测操作执行时间和空间的使用情况,为数据库的再组织提供依据;

▲ 数据库的再组织程序,实现对数据库数据的重新组织;

▲ 转存、编辑和打印程序,用于转存数据库的部分或全部数据,编辑数据或按规定格式打印数据。

5) 数据库系统

数据库系统不是指数据库本身,也不是仅指数据库管理系统,而是指计算机系统中引

入数据库后的系统构成。一般认为数据库系统由计算机数据库、数据库管理系统和数据库管理员(DBA)构成,引入数据库后的计算机、软硬件的层次结构如图 1.5 所示。

由图可见,数据库管理系统是通过操作系统使用计算机的,它向操作系统申请所需的软、硬件资源,并接受操作系统的控制和调度;操作系统是数据库管理系统与硬件之间的接口,数据库通过操作系统进行数据的物理存取。所以,数据库是在操作系统的支持下工作的。从用户的角度来看,数据库管理系统扩大了计算机系统原有的功能。用户通过应用程序(或应用程序包)及数据库管理系统使用数据库。

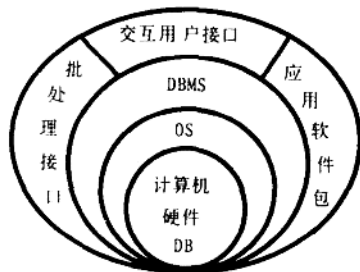


图 1.5 数据库系统

一个计算机系统引入并运行数据库系统,必须满足一定的条件,或者说数据库系统对计算机硬、软件有一定的要求,这些要求是:

- ① 应有足够大的内存,以存放操作系统、DBMS 程序、应用程序和用以建立系统缓冲区、数据表等;
- ② 应有容量足够大的直接存取的辅助存储器,如磁盘等;
- ③ 高的数据通道传输能力;
- ④ 操作系统必须支持 DBMS 的运行。

任何一个单位的数据都可按实际应用的要求使用数据库技术组成一个整体,以供使用。一个数据库系统通常是由一个单位组织建立的,以供本单位或本系统服务。如学校中的学籍管理,企业中的经营管理,政府机关中用于计划统计、资料检索等。这个庞大的公用数据资源需要有专职的管理人员负责管理,这个人就是数据库管理员(DBA),是设计、建立、管理和维护数据库软件人员。其主要职责是:

- ① 决定数据库的信息和结构,确定某现实问题的实体联系模型,建立与 DBMS 有关的数据模型和概念模式。
- ② 决定存储结构和存取策略,建立内模式和概念模式/内模式变换,使数据的存储空间利用率和存取效率两方面都较优。
- ③ 充当用户和数据库系统的联络员,建立外模式和外模式/概念模式变换。
- ④ 定义数据的安全性要求和完整性约束条件,以保证数据库的安全性和完整性。安全性要求是指用户对数据库的存取权限,完整性条件是指对数据进行有效性检验的一系列规则和措施。
- ⑤ 确定数据库的后援支持手段及制定当系统出现故障时数据库的恢复策略。
- ⑥ 监视并改善系统的“时空”性能,提高系统的效率。当系统需要改造时,负责修改和调整外模式、概念模式和内模式。

DBA 有两个很重要的工具。一个是一系列的实用程序(DBMS 的装配、重组、日志、恢复、统计分析等程序),DBA 用此完成一系列的工作。另一个工具是数据字典,它本身也是一个数据库,存储着关于数据的“数据”,其中包括三种模式的定义、记录型及字段的定义、物理文件的结构等。DBA 通过查询数据字典,掌握系统当前的各种工作情况,了解用户

的情况。DBA 还需把对数据库各方面的变化及时地登记到数据字典中去,使之能正确地反映系统的当前值,从而起到控制、管理、维护数据库系统的作用。

1.3 关系数据库基础

1) 集合及其表示方法

集合是一些不同对象的总体,这些对象称为这个集合的元素或成员。例如一个班级中所有的学生就构成了一个集合,其中的每一个学生都是这个集合中的一个成员。通常用大写字母表示集合的名称,用小写字母或数字表示集合的元素。

集合的表示方法有两种,即枚举法与描述法。

枚举法是把一个集合中的所有元素一一列出来并用逗号分隔,再用大括号把它们括起来。例如,集合 S 是由 a, b, c 三个元素组成的,则可表示为

$$S = \{a, b, c\}$$

很多情况下,当一个集合的元素有一些共同性质时,也可用这些性质来描述这个集合。如 $S = \{2, 4, 6, 8, 10\}$,我们可以说集合是不大于 10 的正偶数的集合,这时可表示如下:

$$S = \{x \mid x \text{ 是不大于 10 的正偶数}\}$$

这种方法叫描述法。描述法是一种更为有用的表示法,因为当一个集合有很多元素时,一一列举显得很不方便,有时甚至是做不到的(例如无限集合)。

如果元素 a 是集合 S 的元素,则记作 $a \in S$,读作 a 属于集合 S ,否则记作 $a \notin S$ 。

关于集合,下面的概念是重要的:

- ① 集合中任何二个元素不能相同;
- ② 元素在集合中没有次序概念,如 $\{a, b\}$ 和 $\{b, a\}$ 表示同一个集合;
- ③ 一个集合可以作为另一个集合的元素,如在集合 $A = \{a, \{a, b\}, b, 5\}$ 中,集合 $\{a, b\}$ 是 A 的一个元素。

一个元素也没有的集合称为空集合,或称为空集,记作 $\emptyset$ 。如用枚举法表示则为 $\{\}$ 。如果一个集合 A 包含 m 个元素(m 为一非负的整数),则称集合 A 为有限集合。非有限集合即为无限集合。例如一切自然数的集合,即为无限集合。若有两个集合 A 与 B ,如集合 A 的所有的元素都属于集合 B ,则称集合 A 是集合 B 的子集,记作 $A \subseteq B$ 。若集合 A 是集合 B 的子集,集合 B 也是集合 A 的子集,则两集合相等,即 $A = B$ 。

2) 关系模型的数学概念

关系是一个二维表,对二维表以及表中数据的操作给予严格的数学概念定义,是研究关系数据库的基础。

【定义 1】 设 $D_1, D_2, \dots, D_n$ 为任意集合,定义 $D_1, D_2, \dots, D_n$ 的笛卡尔乘积为:

$$D_1 \times D_2 \times \dots \times D_n = \{(d_1, d_2, \dots, d_n) \mid d_i \in D_i, i = 1, 2, \dots, n\}$$

此定义中应用了集合论中的符号。大括号 $\{\}$ 表示集合,其中小括号中有该集合的定义。上面的集合定义可以读作:该集合由形如 $(d_1, d_2, \dots, d_n)$ 的元素组成,元素中的 d_i 分别属于第 i 个集合 D_i 。

这个集合中的每一个元素叫做一个 n 元组,简称元组,而元组中的 d_i 称为该元组的第 i 个分量,它分别取自相对应的集合 D_i 。

例 1.1 设集合 $D_1 = \{\text{丁一, 张山, 李素}\}$; $D_2 = \{\text{男, 女}\}$; $D_3 = \{23, 22, 21\}$, 则 $D_1 \times D_2 \times D_3 = \{(\text{丁一, 男, 23}), (\text{丁一, 男, 22}), (\text{丁一, 男, 21}), (\text{丁一, 女, 23}), (\text{丁一, 女, 22}), (\text{丁一, 女, 21}), (\text{张山, 男, 23}), (\text{张山, 男, 22}), (\text{张山, 男, 21}), (\text{张山, 女, 23}), (\text{张山, 女, 22}), (\text{张山, 女, 21}), (\text{李素, 男, 23}), (\text{李素, 男, 22}), (\text{李素, 男, 21}), (\text{李素, 女, 23}), (\text{李素, 女, 22}), (\text{李素, 女, 21})\}$ 。

笛卡尔积元素的个数等于构成此乘积的各集合的元素数目之积。例 1.1 中, $D_1 \times D_2 \times D_3$ 应有 $3 \times 2 \times 3 = 18$ 个元素。

【定义 2】 笛卡尔积元素的个数等于构成此乘积的各集合的元素数目之积。例 1 中, $D_1 \times D_2 \times \dots \times D_n$ 的一个子集 R 称之为定义在集合 $D_1, D_2, \dots, D_n$ 上的 n 元关系。集合 $D_1, D_2, \dots, D_n$ 称之为 R 的域。 n 是关系的目或度。

所以关系也是一个集合, 它的元素为元组。关系是一个二维表, 表的每一行对应一个元组, 列为关系的属性。关系中每一个属性都有一个取值范围, 称为属性的值域。每一个属性对应一个值域, 不同的属性也可对应同一个值域。这些属性集相当于记录型, 表中的第一行相当于一个记录值。记录型称为关系模式。

我们从例 1.1 中取出笛卡尔积的一个子集来构造一个关系, 如图 1.2 所示。其关系模式为: $S(\text{学号, 姓名, 年龄, 系别})$, 其中 S 为关系名, 学号、姓名、年龄、系别均为属性名。

在实际应用中, 关系通常是从笛卡尔积中选取的有意义的子集。一个关系应有以下性质:

- ① 任意两行不能全同;
- ② 行的次序可以任意交换;
- ③ 列的次序可以任意交换;
- ④ 每一个属性值是基本的, 不可分裂的。

关系数据库也遵循三级体系结构: 关系子模式、关系模式和内模式。

数据库描述包括若干域的定义以及在这些域上定义的若干关系模式, 称为数据库的型。数据库的值是这些关系模式在某一时刻对应的关系集合。关系模式是稳定的, 而关系是随时间不断变化的。

3) 传统的关系运算

关系代数是以集合代数为基础发展起来的, 它是以关系为运算对象的一组高级运算的集合。关系代数运算可分成两类: 一类是把关系看成一个集合, 集合里的运算(并、差、交、笛卡尔积等)可以引入到关系的运算中来; 另一类是针对数据库环境, 专门设计的关系运算, 包括对关系进行垂直分解(投影运算)、水平分解(选择运算)、关系的结合(连接运算)等。

【定义 3】 若两个关系有相同的度数, 以及它们的第 $j(j=1, 2, \dots, n)$ 个属性分别取值于同一个域, 则称这两个关系是相容的。

(1) 并, 设有两个相容的关系 R 和 S , 定义 R 和 S 的并是由属于 R 或属于 S 的元组组成的集合, 记为 $R \cup S$ 。

(2) 差, 定义两个相容的关系 R 和 S 的差是由属于 R 但不属于 S 的元组组成的集合, 记为 $R - S$ 。

(3) 交, 定义两个相容的关系 R 和 S 的交是同时属于 R 和 S 的元组的集合, 记为

$R \cap S$ 。

【定义4】笛卡尔积。设关系 R 和 S 的元组分别为 r, s 。定义 R 和 S 的笛卡儿积是一个 $(r+s)$ 元组的集合, 每个元组的前 r 个分量来自 R 的一个元组, 后 s 个分量来自 S 的一个元组, 记为 $R \times S$ 。若 R 有 m 个元组, S 有 n 个元组, 则 $R \times S$ 应有 $m \times n$ 个元组。

4) 专门的关系运算

(1) 选择运算(Selection)

该运算是从一个关系中选出所有满足指定条件的元组, 组成新的关系, 这一新建的关系是原来关系的子集, 记为:

$$\sigma_E(R)$$

其中, σ 表示选择命令, R 是作为运算对象的关系, E 是条件表达式。

一般地, 一个简单选择条件是形同 $A \odot C$ 的表达式, A 是属性, C 是常量或属性, $\odot$ 是比较算符, 即 $=, \neq, \leq, \geq, <, >$, 六者之一。

简单选择条件可以通过逻辑运算符非($\neg$)、与($\wedge$)、或($\vee$)的连接组成复杂的选择条件, 在无括号时, 与先乘除后加减的算术运算优先规则类似, 逻辑运算的运算顺序是非、与、或。如果一个元组 t 中的值代入到选择条件 E 中, 能使 E 为真, 则称 t 满足 E 。例如选择条件 E 为(性别="女")。

【定义5】设 R 是一个关系, E 是一个选择条件, 选择运算 $\sigma_E(R)$ 的结果是由 R 中那些满足选择条件 E 的元组组成的关系, 即 $\sigma_E(R) = \{t \in R \mid t \text{ 满足 } E\}$ 。

(2) 投影运算(Projection)

投影运算是从一个关系出发构造其垂直子集的运算。即从一个给定关系的所有属性中选择某些指定属性来组成一个新的关系, 显然新建立的关系是原来关系的子集。投影运算可以表示为

$$\Pi_T(R)$$

其中, Π 是投影命令, R 是作为运算对象的关系, T 是属性表, 它是从关系 R 中投影出来的那些属性, 显然通过投影运算产生的新关系的属性要比关系 R 的属性少, 因此要给它新的关系名。

直观地讲, 投影就是划去那些在 Π 的下标中没有列出的属性值。

【定义6】设关系 R 的关系模式 $(A_1, A_2, \dots, A_n)$, $t = (a_1, a_2, \dots, a_n)$ 是 R 中的元组, t 在属性集 $X = A_1, A_2, \dots, A_k$ 上的投影规定为关系模式 $(A_1, A_2, \dots, A_k)$ 上的元组 $(a_1, a_2, \dots, a_k)$, 记为

$$\Pi_X(t) = (a_1, a_2, \dots, a_k)$$

或 $t[X] = (a_1, a_2, \dots, a_k)$

关系 R 在 X 上的投影为 R 中所有元组在 X 上的投影的集合, 即

$$\Pi_X(R) = \{\Pi_X(t) \mid t \in R\}$$

由于投影运算后取消了某些列, 因而可能出现重复行, 这些重复行应该由实现关系运算的系统予以取消。

(3) 条件连接运算(Join)

该运算是从两个关系的笛卡尔积中选取满足条件的元组组成新的关系, 表示为

$$R \bowtie_F S$$