

数据库基础

本章主要介绍与数据库有关的基本概念和知识,包括数据、信息和数据处理、数据管理技术的发展、数据库系统、数据模型以及关系数据库等。这些基本概念和知识是学习和使用 Visual FoxPro 的基础。

1.1 数据、信息与数据处理

1.1.1 数据与信息

1. 数据

数据是客观事物属性的取值,是信息的具体描述和表现形式,是信息的载体。在计算机系统中,凡是能为计算机所接受和处理的各种字符、数字、图形、图像及声音等都可称为数据。因此,数据泛指一切可被计算机接受和处理的符号。数据可分为数值型数据(如产量、价格、成绩等)和非数值型数据(如姓名、日期、文章、声音、图形、图像等)。数据可以被收集、存储、处理(加工、分类、计算等)、传播和使用。

2. 信息

信息是事物状态及运动方式的反映(表现形式),需经过加工、处理后才能进行交流和使用的。人们往往用数据去记载、描述和传播信息,因此数据是描述或表达信息的具体表现形式,是信息的载体。

信息与数据既有联系又有区别,它们之间的关系可描述为:信息是对客观现实世界的反映,数据是信息的具体表现形式。注意,可以用不同的数据形式表示同样的信息,信息不随它的数据形式的不同而改变。例如,某个部门要召开会议,可以把“开会”这样一个信息通过广播(声音形式的数据)、文件(文字形式的数据)等方式通知给有关单位,在这里,声音或文字是不同的反映方式(表现形式),可以表示同一个信息。

1.1.2 数据处理

数据处理也称为信息处理。所谓数据处理,是指利用计算机将各种类型的数据转换成信息的过程。它包括对数据的采集、整理、存储、分类、排序、加工、检索、维护、统计和传输等

一系列处理过程。数据处理的目的是从大量的、原始的数据中获得人们所需要的资料并提取有用的数据成分,从而为人们的工作和决策提供必要的数据库基础和决策依据。

1.2 数据管理技术的发展

数据管理是指对数据进行组织、存储、分类、检索和维护等操作,是数据处理的核心。随着计算机硬件技术和软件技术的发展,数据管理水平的不断提高,管理方式也发生了很大的变化。数据管理技术的发展主要经历了人工管理、文件管理和数据库系统管理三个阶段。

1.2.1 人工管理阶段

人工管理阶段始于 20 世纪 50 年代,出现在计算机应用于数据管理的初期。这个时期的计算机主要用于科学计算。从硬件看,由于当时没有磁盘作为计算机的存储设备,数据只能存放于卡片、纸带、磁带上。在软件方面,既没有操作系统,也没有专门管理数据的软件,数据由计算生成或由处理它的程序自行携带。

在人工管理阶段,数据管理存在的主要问题如下:

- 数据不能独立。编写的程序直接针对程序中的数据,程序的运行依赖于数据的逻辑格式和物理格式。当数据修改时,程序也得修改;而程序修改后,数据的格式、类型也得变化,以适应处理它的程序。
- 数据不能长期保存。数据被包含在程序中,程序运行结束后,数据和程序一起从内存中释放。
- 没有专门进行数据管理的软件。人工管理阶段不仅要设计数据的处理方法,而且还要说明数据在存储器中的存储地址。应用程序和数据是相互结合且不可分割的,各程序之间的数据不能相互传递,数据不能被重复使用。



图 1-1 人工管理阶段程序与数据之间的关系

因而这种管理方式既不灵活,也不安全,编程效率低下,程序维护和数据管理困难。

- 一组数据对应于一个程序。一个程序中的数据不能被其他程序利用,数据无法共享,从而导致程序和程序之间有大量重复的数据存在。

在人工管理阶段,程序与数据之间的关系如图 1-1 所示。

1.2.2 文件管理阶段

在 20 世纪 60 年代,计算机软、硬件技术得到快速发展,硬件方面有了磁盘、磁鼓等大容量且能长期保存数据的存储设备;软件方面有了操作系统,操作系统中有专门的文件系统用于管理外部存储器上的数据文件,数据与程序分开,且数据能长期保存。

在文件管理阶段,把有关的数据组织成一个文件,这种数据文件能够脱离程序而独立存

储在外存储器上,由一个专门的文件管理系统对其进行管理。在这种管理方式下,应用程序通过文件管理系统对数据文件中的数据进行加工处理。应用程序与数据文件之间具有一定的独立性。与早期人工管理阶段相比,使用文件系统管理数据的效率和数量都有很大提高,但仍存在以下问题:

- 数据没有完全独立。虽然数据和程序被分开,但所设计的数据依然是针对某一特定的程序,所以无论是修改数据文件还是程序文件,二者都要相互影响。也就是说,数据文件仍然高度依赖于其对应的程序,不能被多个程序所共享。
- 存在数据冗余。文件系统中的数据没有合理和规范的结构,使得数据的共享性极差,即使是不同程序使用部分相同的数据,数据结构也完全不同,同样需要创建各自的数据文件。这便造成数据的重复存储,即数据的冗余。
- 数据不能被集中管理。文件系统中的数据文件没有集中的管理机制,数据的安全性和完整性都不能得到保障。各数据、数据文件之间缺乏联系,给数据处理造成不便。

在文件系统阶段,程序与数据之间的关系如图 1-2 所示。

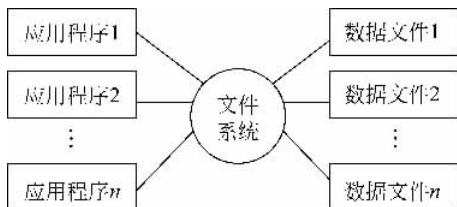


图 1-2 文件系统阶段程序与数据之间的关系

1.2.3 数据库系统管理阶段

由于用文件系统管理数据存在缺陷,所以迫切需要一种新的数据管理方式,把数据组成合理结构,进行集中、统一管理。数据库技术始于 20 世纪 60 年代末,到了 20 世纪 80 年代,随着计算机的普遍应用和数据库系统的不断完善,数据库系统在全世界范围内得到广泛的应用。

在数据库系统管理阶段,将所有数据集中到一个数据库中,形成一个数据中心,实行统一规划,集中管理,用户通过数据库管理系统来使用数据库中的数据。

数据库系统的主要特点如下:

- 实现了数据结构化。在数据库中采用了特定的数据模型组织数据。数据库系统把数据存储在有一定结构的数据库文件中,实现了数据的独立和集中管理,克服了人工管理和文件管理的缺陷,大大方便了用户的使用和提高了数据管理的效率。
- 实现了数据共享。数据库中的数据能被多个应用程序共享,为多个用户服务。数据共享可以减少数据冗余,节约存储空间,还能够避免数据之间的不一致性。
- 实现了数据独立。用户的应用程序与数据的逻辑结构及数据的物理存储方式无关。数据独立可以简化应用程序的编制,减少应用程序的维护和修改。
- 实现了数据统一控制。数据库系统提供了各种控制功能,保证了数据的并发控制、

安全性和完整性。数据库作为多个用户和应用程序的共享资源,允许多个用户同时访问。并发控制可以防止多用户并发访问数据时产生的数据不一致。安全性可以防止非法用户存取数据。完整性可以保证数据的正确性和有效性。

在数据库系统管理阶段,应用程序和数据完全独立,应用程序对数据管理和访问更加灵活。一个数据库可以为多个应用程序共享,使得程序的开发和运行效率大大提高,减少了数据冗余,实现了数据资源共享,提高了数据的完整性、一致性以及数据的管理效率。

在数据库系统管理阶段,程序与数据之间的关系如图 1-3 所示。

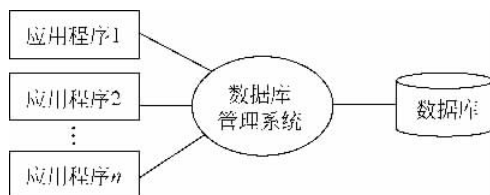


图 1-3 数据库系统阶段程序与数据之间的关系

1.3 数据库系统基本概念

在数据库技术中,人们常常接触到数据库、数据库管理系统、数据库系统、数据库应用系统这些名词,它们之间有着一定的联系和区别。

1.3.1 数据库

数据库(Data Base,DB)就是按一定的组织形式存储在一起的相互关联的数据的集合。实际上,数据库就是一个存放大量业务数据的场所,其中的数据具有特定的组织结构。所谓“组织结构”,是指数据库中的数据不是分散的、孤立的,而是按照某种数据模型组织起来的,不仅数据记录内的数据之间是彼此相关的,而且数据记录之间在结构上也是有机地联系在一起。数据库具有数据的结构化、独立性、共享性、冗余量小、安全性、完整性和并发控制等基本特点。在数据库系统中,数据库已成为各类管理系统的核心基础,为用户和应用程序提供了共享的资源。

1.3.2 数据库管理系统

数据库管理系统(Data Base Management System,DBMS)是负责数据库的定义、建立、操纵、管理和维护的一种计算机软件,是数据库系统的核心部分。数据库管理系统是在特定操作系统的支持下工作的,它提供了对数据库资源进行统一管理和控制的功能,使数据结构和数据存储具有一定的规范性,提高了数据库应用的简明性和方便性。DBMS 为用户管理数据提供了一整套命令,利用这些命令可以实现对数据库的各种操作,如数据结构的定义,数据的输入、输出、编辑、删除、更新、统计和浏览等。

1.3.3 数据库系统

数据库系统(Data Base System,DBS)是由计算机系统引入数据库后的系统构成,它是一个具有管理数据库功能的计算机软硬件综合系统。具体地说,它主要包括计算机硬件、操作系统、数据库(DB)、数据库管理系统(DBMS)和相关软件、数据库管理员及用户等组成部分。数据库系统具有数据的结构化、共享性、独立性、可控冗余度以及数据的安全性、完整性和并发控制等特点。

- 硬件系统:是数据库系统的物理支持,包括主机、外部存储器、输入/输出设备等。
- 软件系统:包括系统软件和应用软件。系统软件包括支持数据库管理系统运行的操作系统(如 Windows 2000)、数据库管理系统(如 Visual FoxPro 6.0)、开发应用系统的高级语言及其编译系统等;应用软件是指在数据库管理系统基础上,用户根据实际问题自行开发的应用程序。
- 数据库:是数据库系统的管理对象,为用户提供数据的信息源。
- 数据库管理员(DBA):是负责管理和控制数据库系统的主要维护管理人员。
- 用户:是数据库的使用者,利用数据库管理系统软件提供的命令访问数据库并进行各种操作。用户包括专业用户和最终用户。专业用户即程序员,是负责开发应用程序的设计人员。最终用户是对数据库进行查询或通过数据库应用系统提供的界面使用数据库的人员。

1.3.4 数据库应用系统

数据库应用系统(Data Base Application System,DBAS)是在 DBMS 支持下根据实际问题开发出来的数据库应用软件。一个 DBAS 通常由数据库和应用程序两部分组成,它们都需要在 DBMS 支持下开发。

由于数据库的数据要供不同的应用程序共享,因此在设计应用程序之前首先要对数据库进行设计。数据库的设计是以“关系规范化”理论为指导,按照实际应用的报表数据,首先定义数据的结构,包括逻辑结构和物理结构,然后输入数据形成数据库。开发应用程序也可采用功能分析,总体设计,模块设计,编码调试等步骤来实现。

1.3.5 数据库系统的数据模式

从数据库管理系统的角度看,数据库系统可分为三级模式,从外到内依次为外模式、模式和内模式。

1. 外模式

外模式又称子模式或用户模式,它是数据库用户和数据库系统的接口,是数据库用户看到的数据视图,是对数据库中局部数据的逻辑结构和特征的描述,是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。同一个外模式可

以被某一个用户的多个应用程序所使用,但一个应用程序只有一个外模式。

2. 模式

模式也称逻辑模式或概念模式,它是对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。一个数据库只有一个模式。数据库模式以某一种数据模型为基础。

模式是在数据库模式结构的中间层中,既不涉及数据的物理存储细节和硬件环境,也与具体的应用程序、应用开发工具以及高级程序设计语言无关。DBMS 提供模式定义语言 DDL 来描述模式。定义模式时要定义数据的逻辑结构,包括记录由哪些数据项构成;数据项的名称、类型、取值范围;数据之间的联系;与数据有关的安全性、完整性要求等。

3. 内模式

内模式又称为存储模式,它是对数据库物理结构和存储方式的描述,是数据在数据库内部的表示方式。它规定了数据在存储介质上的物理组织方式,记录了寻址技术、物理存储块的大小、溢出处理方法等。一个数据库只有一个内模式。

为了实现三级模式的联系和转换,数据库管理系统在三级模式之间提供了两层映像:外模式/模式映像和模式/内模式映像。映像是一种对应规则,指出映像双方应如何进行转换。数据库的三级模式通过这两层映像联结起来,从而为各类用户提供操纵数据库的手段。

- 外模式/模式映像:定义外模式与模式之间的对应关系。当数据库的全局逻辑结构改变时,只需要修改外模式与模式之间的对应关系,而不必修改局部逻辑结构,即保证外模式不变,相应的应用程序也不必修改,从而实现数据和程序的逻辑独立性。
- 模式/内模式映像:定义数据全局逻辑结构与存储结构之间的对应关系。当数据库的物理存储结构改变时,只需要修改模式与内模式之间的对应关系,即可保持模式不变,从而实现数据和程序的物理独立性。

1.4 数据模型

数据模型是对现实世界数据特征的抽象,是用来描述数据的一组概念和定义。数据模型按不同的应用层次可划分为概念数据模型和逻辑数据模型两类。概念数据模型又称为概念模型,是一种面向客观世界、面向用户的模型,主要用于数据库的设计。而逻辑数据模型常称为数据模型,是一种面向计算机系统的模型,主要用于实现数据库管理系统。

1.4.1 数据模型概述

数据模型是对现实世界数据特征的抽象,是用来描述数据的结构和联系的一组概念和定义,是数据库的核心内容。

由于计算机不能直接处理现实世界中的具体事物,所以必须把具体事物转换成计算机能够处理的数据。在数据库系统中,实现转换的过程通常是先把现实世界中的客观事物抽

象为概念数据模型(简称概念模型),然后再把概念数据模型转换为某一数据库管理系统所支持的逻辑数据模型(简称数据模型),如图 1-4 所示。

概念数据模型和逻辑数据模型是数据模型的不同应用层次。概念数据模型是从现实世界到数据世界的一个中间层次,是一种面向客观世界、面向用户的模型,是数据库设计人员进行数据库设计的重要工具,也是数据库设计人员和用户之间进行交流的语言,E-R 模型、扩充的 E-R 模型等是常用的概念模型。逻辑数据模型是一种面向数据库系统的模型,即依赖于某种具体的数据库管理系统 DBMS,主要用于 DBMS 的实现,常见的逻辑数据模型包括层次模型、网状模型和关系模型等。

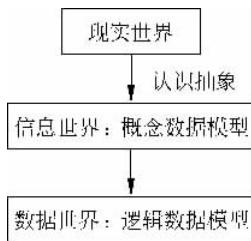


图 1-4 事物到数据的抽象过程

1.4.2 E-R 数据模型

E-R 数据模型(Entity-Relationship Data Model)即实体-联系数据模型,用来描述现实世界,具有直观、自然、语义丰富及便于向逻辑数据模型转换等优点。

设计 E-R 模型的目标是有效和自然地模拟现实世界,而不是关心它在计算机中如何实现,因此 E-R 模型中只应包含那些对描述现实世界具有普遍意义的抽象概念。E-R 模型中的基本概念有实体、联系、属性等。

1. 实体(entity)

客观存在并可相互区分的事物称为实体。它是信息世界的基本单位。实体既可以是人,也可以是物;既可以是实际对象,也可以是抽象对象;既可以是事物本身,也可以是事物与事物之间的联系。例如,一个学生、一个教师、一门课程、一支铅笔、一部电影、一个部门等都是实体。

同类型的实体的集合称为实体集(entity set)。例如,一个学校的全体学生是一个实体集,而其中的每个学生都是实体集的成员。

2. 联系(relationship)

联系是实体集之间关系的抽象表示,是对现实世界中事物之间关系的描述。例如,公司实体集与职工实体集之间存在“聘任”联系。实体集之间的联系可分为以下 3 类。

(1) 一对一联系(1:1): 如果对于实体集 A 中的每一个实体,实体集 B 中至多有一个实体与之联系,反之亦然,则称实体集 A 与实体集 B 具有一对一联系。例如,在一个学校中,一个班级只有一个正班长,而一个班长只在一个班中任职,则班级与班长之间具有一对一联系。又如职工和工号的联系是一对一的,每一个职工只对应于一个工号,不可能出现一个职工对应于多个工号或一个工号对应于多名职工的情况。

(2) 一对多联系(1:n): 如果对于实体集 A 中的每一个实体,实体集 B 中有 n 个实体($n \geq 0$)与之联系,反之,对于实体集 B 中的每一个实体,实体集 A 中至多只有一个实体与之联系,则称实体集 A 与实体 B 有一对多联系。

考查系和学生两个实体集,一个学生只能在一个系里注册,而一个系可以有很多学生,

所以系和学生是一对多联系。又如单位的部门和职工的联系是一对多的,一个部门对应于多名职工,多名职工对应于同一个部门。

(3) 多对多联系($m:n$): 如果对于实体集 A 中的每一个实体,实体集 B 中有 n 个实体($n \geq 0$)与之联系,反之,对于实体集 B 中的每一个实体,实体集 A 中也有 m 个实体($m \geq 0$)与之联系,则称实体集 A 与实体 B 具有多对多联系。例如,一门课程同时有若干名学生选修,而一个学生可以同时选修多门课程,则课程与学生之间具有多对多联系。又如在单位中,一个职工可以参加若干个项目的工作,一个项目可有多个职工参加,则职工与项目之间具有多对多联系。

3. 属性(attribute)

描述实体的特性称为属性。一个实体可由若干个属性来刻画。属性的组合表征了实体。例如,商品有商品代码、商品名称、单价、生产日期、进口否、商品外形等属性;铅笔有商标、软硬度、颜色、价格、生产厂家等属性。

唯一标识实体的一个属性集称为码,例如,学号是学生实体的码。属性的取值范围称为域,例如,学生实体中,性别属性的域为(男,女),年龄的域可定为 18~60。

这里要注意区分属性的型与属性的值,例如,学生实体中的学号、姓名等属性名是属性的型,而某个学生的“0001”、“张三”等具体数据则称为属性值。

相应地,实体也有型和值之分,实体的型用实体名及其属性名的集合来表示。例如,学生以及学生的属性名集合构成学生实体的型,可以简记为:学生(学号,姓名,性别,出生日期,籍贯,专业,是否团员),而(“0001”,“张三”,“女”,1982-12-23,“成都”,“信息”,. T.)是一个实体值。实体集实际上就是同类型实体的集合,例如,全体学生就是一个实体集。

1.4.3 几种主要数据模型

数据模型一般分为 3 种,即:层次模型、网状模型和关系模型。如果数据库中的数据是依照层次模型存储的数据,则该数据库称为层次数据库;如果是依照网状模型进行存储,则该数据库称为网状数据库;如果是依照关系模型进行存储,则该数据库称为关系数据库。

1. 层次模型

层次模型是数据库系统最早使用的一种模型。层次模型表示数据间的从属关系结构,它是以树形结构表示实体(记录)与实体之间联系的模型。层次模型的主要特征如下:

- 层次模型像一棵倒立的树,有且仅有一个无双亲的根结点。
- 除根结点以外的子结点,有且仅有一个父结点。

层次数据模型只能直接表示一对多(包括一对一)的联系,但不能表示多对多联系。例如,学校的行政机构(如图 1-5 所示)、企业中的部门编制等都是层次模型。支持层次模型的数据库管理系统称为层次数据库管理系统。

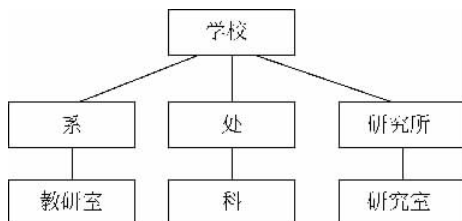


图 1-5 学校行政机构的层次模型

2. 网状模型

网状模型是以网状结构表示实体与实体之间联系的模型,使用网状模型可表示多个从属关系的层次结构,也可表示数据间的交叉关系,是层次模型的扩展。网状模型的主要特征如下:

- 允许有一个以上的结点无双亲。
- 一个结点可以有多个双亲。

网状数据模型的结构比层次模型更具普遍性,它突破了层次模型的两个限制,允许多个结点没有双亲结点,允许一个结点具有多个双亲结点。此外,它还允许两个结点之间有多种联系。因此网状数据模型可以更直接地描述现实世界。图 1-6 中给出了一个简单的网状模型。

网状模型是以记录为结点的网络结构。支持网状数据模型的数据库管理系统称为网状数据库管理系统。

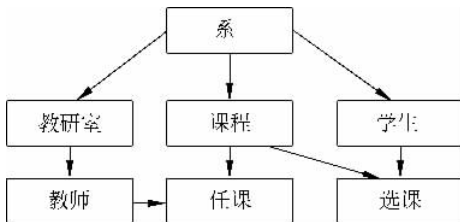


图 1-6 网状模型示例

3. 关系模型

关系模型是一种以关系(二维表)的形式表示实体与实体之间联系的数据模型。关系模型不像层次模型和网状模型那样使用大量的链接指针把有关数据集合到一起,而是用一张二维表来描述一个关系。关系模型的主要特点如下:

- 关系中的每一分量不可再分,是最基本的数据单位。
- 关系中每一列的分量是同属性的,列数根据需要而设,且各列的顺序是任意的。
- 关系中每一行由一个具体实体或者联系的一个或多个属性构成,且各行的顺序可以是任意的。
- 一个关系是一张二维表,不允许有相同的列(属性),也不允许有相同的行(元组)。

表 1-1 所示的是一张商品情况表。在二维表中,每一行称为一个记录,用于表示一组数据项;表中的每一列称为一个字段或属性,用于表示每列中的数据项。表中的第一行称为字段名,用于表示每个字段的名称。

表 1-1 商品情况表

商品代码	商品名称	单价	生产日期	进口否	商品外形	备注
s1	笔记本电脑	9380.00	2006.08.12	T	(略)	(略)
s2	彩色激光打印机碳粉	52.00	2006.07.23	F	(略)	(略)
s3	POS 机色带	1.50	2006.07.03	F	(略)	(略)
s4	笔记本电脑内存条	320.00	2006.04.15	F	(略)	(略)
s5	液晶电脑一体机	3288.00	2006.06.19	T	(略)	(略)
s6	指纹 U 盘	845.00	2006.09.10	T	(略)	(略)
s7	MP3 手表	228.00	2006.05.28	F	(略)	(略)
s8	GPS 车载电脑	5850.00	2006.08.20	F	(略)	(略)
s9	15 寸液晶显示器	1780.00	2006.07.24	T	(略)	(略)

关系模型对数据库的理论和实践产生了极大的影响,它与层次模型和网状模型相比有明显的优势,是目前最流行的数据库模型。支持关系模型的数据库管理系统称为关系数据库管理系统。Visual FoxPro 采用的数据模型是关系模型,因此它是一个关系数据库管理系统。

1.5 关系数据库

关系数据库是依照关系模型设计的若干个二维数据表文件的集合。在 Visual FoxPro 中,一个关系数据库由若干个数据表组成,每个数据表又是由若干个记录组成,每个记录由若干个数据项组成。一个关系的逻辑结构就是一张二维表。这种用二维表的形式表示实体和实体之间联系的数据模型称为关系数据模型。

1.5.1 关系术语

关系是建立在数学集合概念基础之上的,是由行和列表示的二维表。

关系:一个关系就是一张二维表,每个关系有一个关系名。在 Visual FoxPro 中,一个关系就称为一张数据表,如表 1-1。

元组:二维表中水平方向的行称为元组,每一行是一个元组。在 Visual FoxPro 中,一行称为一个记录,如表 1-1 中的一行数据项。

属性:二维表中垂直方向的列称为属性,每一列有一个属性名。在 Visual FoxPro 中,一列称为一个字段,如表 1-1 中的商品代码、商品名称、单价等对应的列。

域:表中属性的取值范围称为域。Visual FoxPro 中,一个字段的取值范围通过一个字段的宽度定义。

分量:元组中的一个属性值,如表 1-1 中的“笔记本电脑”。

候选码(候选关键字):表中的某个属性或属性组合,其值可唯一确定一个元组。一个关系可以有多个候选码。例如,表 1-1 中,商品名称不重复的情况下,商品代码、商品名称是候选码。

主码(主关键字):从候选码中,选择一个作为主码。一个关系只能有一个主码。如表 1-1 中的商品代码。

外码(外关键字):如果关系中的一个属性不是本关系的主码或候选码,而是另外一个关系的主码或候选码,则该属性称为外码。例如,“销售表”中的商品代码不是“销售表”的主码或候选码,而是“商品销售表”的主码,则商品代码是“销售表”的外码。

主属性:包含在任何一个候选码中的属性。例如表 1-1 中的商品代码、商品名称属性是主属性。

非主属性:不包含在任何一个候选码中的属性。例如表 1-1 中的单价、生产日期、进口否等属性是非主属性。

关系模式:对关系的描述。一个关系模式对应一个关系的结构。其格式如下:

关系名(属性名 1,属性名 2,属性名 3,⋯,属性名 n)