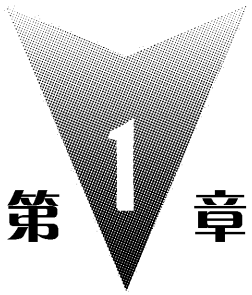


第1章 数据库基本原理

本章以文字阐述和用典型实例说明相结合的方式介绍下列有关数据库的基本原理：

(1) 数据库的产生、发展和它的基本概念；

(2) 数据模型的有关知识、表示方式和基本概念；

(3) 实体集间的三种关系；

(4) 关系的三类完整性；

(5) 关系模型规范化基本知识。

1.1 数据库概述

1.1.1 数据库的产生和发展

数据库技术产生于20世纪60年代中期，几十年来得到了迅速发展。进入21世纪，信息和知识更加迅速膨胀，数据库技术在组织和利用庞大的信息和知识方面将起到越来越重要的作用。

人类活动的整个历史都贯穿着对信息(或数据)的收集、处理、保存和利用。20世纪60年代以来，随着社会生产力的高速发展，信息量急剧膨胀，整个人类社会正成为信息化社会。人们对信息和数据的利用和处理已进入自动化、网络化和社会化阶段，如银行储蓄、股票交易、资料查询、气象预报、机票预订等。这些任务既需要大量数据，又要求快速处理并及时得到结果，是传统的人工方法不可能完成的。飞速发展的计算机技术使上述大规模的数据处理得以实现。即使是很平常的数据处理，借助计算机也可以极大地提高效率。例如，学生的学籍管理是学校的一项重要工作，靠人工查找期末考试有3门或3门以上课程不及格的学生姓名、学号、不及格课程不仅很麻烦，还可能出差错。用计算机管理，就可以快速、准确地完成这项工作。随着计算机和网络技术的迅速发展，现在已经能实现全国几百万考生、几千所学校的高考网上录取工作。至于全国范围内的股票交易、信用卡支付已经是很平常的事了。

数据库(Database)这个名词起源于20世纪50年代。当时，美国为了军事目的将各

种情报集中到一起，揭开了数据库技术的序幕。到了70年代，数据库得到了蓬勃发展，网状系统和层次系统占主导，关系数据库系统处于实验阶段。从80年代起，关系数据库系统逐步取代了网状系统和层次系统。此后，关系数据库得到了长足的发展。70年代中期以后，分布式数据库系统、面向对象的主动数据库系统、智能型数据库系统的相继出现表明数据库技术在不断向更高的水平发展。从目前情况看，关系数据库仍然占绝对的主导地位，并将影响着数据库技术的发展。正因为关系数据库如此重要，像Oracle、SQL Server、Informix、Sybase和Microsoft Access等大型与中小型关系数据库系统都在不断发展。

到现在，数据库技术的发展已经历了4个阶段。

1. 人工管理阶段

20世纪50年代中期以前，计算机主要用于科学计算。由于科学计算的数据量少，数据和应用程序结合在一起，由人工进行管理。当时，还没有磁盘，也没有操作系统。

人工管理数据的特点是：

(1) 数据不保存

数据在运行应用程序时输入，程序执行完释放，不在计算机中保存。

(2) 没有专用软件对数据进行管理

数据的存储结构、存取方法、输入输出方式完全由应用程序确定，数据的改变必然要修改程序。

(3) 数据不共享

数据是面向应用的，即一组数据对应一个程序。各应用程序间很可能存在大量重复数据，即冗余度极大，浪费存储空间。

(4) 数据不具有独立性

当数据的逻辑结构或物理结构发生变化时，必须对应用程序做相应的修改。

2. 文件系统阶段

20世纪50年代后期，计算机开始大量应用于管理方面。由于管理事务存在大量数据，并且这些数据需要长期保留，人们采取文件的方式存储、修改数据，将数据和应用程序分离开来。计算机在硬件方面有了磁盘、磁鼓等直接存取存储设备，软件方面在操作系统中有了专门的数据管理软件。

文件系统管理的特点如下：

(1) 数据可以长期保存

大量的数据保存在计算机的外存设备上，可反复进行查询、修改、插入和删除操作。

(2) 有专用软件对数据进行管理

数据由专门的软件即文件系统进行管理，和程序有一定的独立性。程序的修改受数据改变的影响小，工作效率大大提高。但是，文件系统仍有以下缺点：

(3) 数据共享性差、冗余度大

存放数据的文件是对应一个或几个应用程序的，即文件是面向应用的。不同的应用程序不能共享相同的数据，因此数据的冗余度大，既浪费存储空间，还可能存在不一致性。

(4) 数据独立性差

由于文件系统中的文件是为某一特定应用服务的。所以,一旦数据的逻辑结构改变,必须修改应用程序,修改文件结构的定义。因此,数据和程序之间仍缺乏独立性。

3. 数据库系统阶段

20 世纪 60 年代后期,由于计算机大量应用于数据处理、人工智能和计算机辅助设计等领域。这些领域所处理的数据量非常大,还包含许多非数值数据,而且数据间的联系更加复杂,用文件系统管理数据已不适用。为此,需要有一个高度组织的数据管理系统。另一方面,随着计算机硬件、软件技术的进一步发展,使大量数据集中存储成为可能。数据库系统就是在这样的背景下产生和发展起来的。

数据库系统的特点是:

(1) 数据结构化

数据库在存储数据的同时既描述数据本身的特点,又描述数据间的联系。

(2) 数据冗余度小

数据库存储数据冗余度小,既节约了存储空间,更避免了冗余数据引起的不一致性。

(3) 数据共享性好

数据库中的数据可以做出各种组合,以最优方式满足不同的需要。

(4) 数据独立性高

数据库中的数据既具有物理独立性,又有逻辑独立性。物理独立性是指用户的应用程序与存储在磁盘上的数据是相互独立的。逻辑独立性是指用户的应用程序与数据库的逻辑结构是相互独立的。

(5) 数据有统一管理和控制

数据库系统提供了统一的管理软件,数据由数据库管理系统管理和控制,保证了数据的安全性、完整性和保密性。

4. 高级数据库阶段

数据库技术在商业领域的巨大成功刺激了其他领域对数据库技术的需求。例如,计算机辅助设计/制造(CAD/CAM)、计算机集成制造(CIM)、地理信息系统(GIS)、办公信息系统(OIS)、计算机辅助超大规模集成电路设计(VLSI CAD)等都需要数据库的支持。这些系统在数据类型或数据结构或数据存储方面有特殊要求,传统的数据库系统并不能支持。因此,20 世纪 70 年代中期出现了分布式数据库系统、面向对象的主动数据库系统、智能型数据库系统。目前,通常称它们为高级数据库技术。

1.1.2 现实世界、信息世界与数据世界

1. 现实世界

现实世界存在着大量的事物,这些事物可以是具体的,也可以是抽象的。各个事物都有表征自己的各种特征。例如,某一个人就是一个事物,他的姓名、性别、身高、体重都是他的特征。

起先,人们是通过眼睛等感官接触现实世界的事物的,例如,太阳、月亮、树、鸟、颜色、

声音、气味等具体的事物。随着社会的进步、技术的发展,人们接触现实世界的事物越来越广,包括更多的抽象事物。更为突出的是,人们需要了解各种事物的更为深刻的特征和它们之间更加复杂的关系。例如,还是上面提到的那个人,除了姓名、性别、身高、体重外,人们还需要知道他的其他方面,如身份证号码、年龄、民族、政治面貌、文化程度,甚至需要知道他的专业特长、个人爱好、身体状况等。这些都是这个人的特征。可见,由于考虑问题的不同,同一个事物可用不同的特征来描述它。

现实世界的每一个事物都有反映自身各个方面的特征。每一个事物的全部特征就反映了该事物本身。每一个事物至少有一个特征。

2. 信息世界

人们观察各种事物,在大脑中形成抽象概念,这就是**信息**(Information)。所以说,**信息世界**就是现实世界的事物在人脑中的抽象。例如,还是上面提到的那个人,我们并没有见到其人,但是知道了他的姓名、性别、身份证号码、民族、政治面貌、文化程度等文字材料(也就是抽象出来的信息!),我们就对他有了基本了解。更为重要的是,我们根据他的这些特征可以把他和其他人(另外的事物)区别开来。

3. 数据世界

显然,从现实世界到信息世界的抽象是和计算机完全无关的。为了用计算机处理信息,人们还需要将信息再进一步抽象为计算机所能识别的数据,这种抽象往往和具体的计算机有关。即同样的信息可能因计算机系统的不同抽象出的数据结构不同。

数据世界就是信息世界中信息的数据化。数据世界的数据表示方法不一定和信息世界的描述一致,例如,在数据世界,可能用"1"和"0"分别表示人的性别的"男"和"女",用某种编码表示不同的民族、政治面貌、文化程度等。这种表示方法便于计算机处理。信息和数据是紧密相关的,在许多场合将它们看做同义词。

在数据世界里,将现实世界诸事物中具有有限集合的特征用恰当的编码表示是非常必要的。这样做的结果,既节约了存储空间,又减少了出错的可能,更便于查询和统计。实际上,如果对编码赋予更多的含义,就能发挥更大的作用。例如,用学号的前两位表示学生的入学年份;身份证前 6 位表示登记人户口所在地区的代号,中间 8 位是本人的出生年月日;借书证号的第 1 位用不同的字母表示不同的读者对象(如用 A、B、C 分别表示学生、教师、其他员工)。这样高质量的编码用来进行数据统计是非常方便的。所以说,编码的质量是影响数据库系统的决定性因素。

1.1.3 数据库基本概念

数据、数据库、数据库管理系统和数据库系统是与数据库技术密切相关的 4 个基本概念。

1. 数据

数据(Data)就是描述信息的符号,是数据库中存储的基本对象。随着计算机识别和处理能力的极大提高,现在数据库处理的数据不仅包括数字和文字形式的信息,还包括图像、声音、文件等形式的信息。

数据处理(Data Processing)是将原始数据转换成信息的过程,包括对数据的收集、整理、分类、存储、排序、统计、加工和分析等,数据处理分人工处理和计算机处理两种方式。

2. 数据库

数据库(Database,DB)是在计算机系统中按照一定数据模型组织、存储和应用的相互联系的数据集合。数据库既是存放数据的"仓库",又是一种数据处理技术和方法,它总是与一个信息系统相关联,并作为一个信息系统的核心部件而与之共存。

数据库技术(Database Technique)是一种对数据进行加工以得到有用信息的计算机软件技术。

3. 数据库管理系统

数据库管理系统(Database Management System,DBMS)是一种计算机软件系统。它是数据库系统的核心组成部分。它的主要用途是利用计算机有效地组织数据、存储数据、获取和管理数据。

数据库管理系统是用户和操作系统之间的一层数据管理软件。

数据库管理系统由数据描述语言、数据操纵语言和数据库管理运行程序三部分组成。为了提高数据库的开发效率,除了DBMS,现代数据库还提供了其他一些支持应用开发的工具。

人们通常把以数据库管理系统为核心的应用系统称为**管理信息系统**(Management Information System,MIS)。

4. 数据库系统

数据库系统(Database System,DBS)就是以数据库应用为基础的计算机系统。所以,数据库系统不仅包括必须存储的数据,还包括相应的硬件、软件和各类工作人员。在不引起混淆的情况下常把数据库系统简称为数据库。

(1) 数据

数据是按照需求进行采集并以选定的结构存储在数据库中的,是计算机管理中最重要的资料,它不因硬件的更新、软件的更换而改变。数据库通常由两大部分组成:一部分是有关应用所需的工作数据的集合,称为**物理数据库**,它是数据库的主体;另一部分是各级数据结构的描述,称为**描述数据库**。

(2) 硬件

由于数据库系统存储的数据量很大,DBMS丰富的功能使得自身的规模也很大,还要有各种各样的功能,这就要求硬件必须具有较高的性能,包括足够大的内存和硬盘、较高的数据传送能力等。

(3) 软件

数据库系统软件主要包括:DBMS、支持DBMS的操作系统和以DBMS为核心的应用开发工具等。

(4) 人员

人员是数据库系统的重要组成部分,负责分析、设计、管理和维护数据库。完成这些工作的人员主要是:数据库管理员、系统分析员、应用程序员和最终用户。

1.2 数据模型

随着社会的发展、科技的进步，人们所接触的信息飞速增加，计算机要处理的数据量越来越庞大，相互间的关系越来越复杂。所以，数据库中的大量数据必须按严格的数据模型来组织。数据库中的数据是高度结构化的，它不仅反映数据本身，而且反映数据之间的关系。**数据模型**就是描述这种关系的数据结构形式，在数据库中使用数据模型对现实世界进行抽象。数据模型是数据库系统的核心和基础。

理想的数据模型应能满足三方面的要求：一是能比较真实地描述现实世界；二是容易被人所理解；三是便于在计算机上实现。到目前为止，还没有哪一种数据模型能够满足各种各样的实际需要。在数据库系统中往往针对不同的情况采用不同的数据模型。

根据模型应用的不同目的，可以将模型划分为两类，它们分属于两个不同的层次。第一类模型是概念模型，**概念模型**(Idea Model)是现实世界到信息世界的抽象，又称为**信息模型**。第二类模型是数据模型，**数据模型**(Data Model)是信息世界到数据世界的抽象。在数据库领域最常见的数据模型有 4 种：层次模型、网状模型、关系模型和面向对象模型。其中层次模型和网状模型统称为非关系模型。

目前，关系模型是最常用的数据模型。以关系模型为基础建立的数据库管理系统称为**关系数据库管理系统**(RDBMS)。

1.2.1 数据模型的组成要素

一般地说，任何一种数据模型都是严格定义的一组概念的集合。这些概念精确地描述了系统的静态特征、动态特征和完整性约束条件。因此，数据模型通常都是由数据结构、数据操作和数据的约束条件 3 个要素组成。

1. 数据结构

数据结构是所研究的对象类型(Object Type)的集合。这些对象是数据库的组成成分，例如关系模型中的域、属性、关系等。数据结构是对系统静态特性的描述。

2. 数据操作

数据操作是指数据库中各种对象(型)的实例(值)允许执行的操作的集合，包括操作及有关的操作规则。数据库主要有检索(查询)和更新(包括插入、删除、修改)两大类操作。数据模型必须定义这些操作的确切含义、操作符号、操作规则(如优先级)以及实现操作的语言。数据操作是对系统动态特性的描述[①]。

3. 数据的约束条件

数据的约束条件是一组完整性规则的集合。**完整性规则**是给定的数据模型中数据及其联系所具有的制约和依存规则，用以限定符合数据模型的数据库状态以及状态的变化，

① 这里及本章以后有几处论述文字引自萨师煊、王珊的《数据库系统概论(第三版)》一书，以后不再说明。

以保证数据的正确、有效和相容。

数据模型应该反映和规定本数据模型必须遵守的基本的通用的完整性约束条件。例如，在关系模型中，任何关系必须满足实体完整性和参照完整性两个条件。

此外，数据模型还应该提供定义完整性约束条件的机制，以反映具体应用所涉及的数据必须遵守的特定的语义约束条件。例如，在职工管理系统中规定职工的年龄不能小于18岁；在学生管理系统中规定大专生在校学习时间不得超过6年。

以上3个完整性将在1.3节介绍。

1.2.2 信息世界的基本概念

概念模型是现实世界到数据世界的一个中间层次，用于信息世界的建模，是用户与设计人员之间进行交流的语言，不依赖于具体的计算机硬件和软件。概念模型建立的好坏直接影响到数据模型和整个数据库系统的质量。信息世界的基本概念有：

1. 实体(Entity)

实体就是现实世界中客观存在并可相互区分的事物。实体既可以是看得见摸得着的具体的事物，也可以是抽象的概念或联系。例如，某一本书、某一架飞机、某个学生、某次活动、某种现象、某种理论等都是实体。

2. 属性(Attribute)

实体所具有的特征称为**属性**。一个实体由若干个(至少一个)属性来描述。一个实体的所有属性组成实体本身。例如，学生实体可以由学号、姓名、性别、出生年月日、班级等属性组成。而(0100001、冯东梅、女、1980/12/26、01电子商务1)就是一个学生(实体)的属性值。

3. 码(Key)

惟一标识实体的属性组称为**码**，通常又称为**关键字**。如果实体有多个码，则可以选定其中一个码为**主码**(Primary Key)，通常又称为**主关键字**。如果实体只有一个码，它就是主码。例如，一个学校里，学生实体的学号是肯定不重复的。所以学号可以作为学生实体的码。如果学生实体中含有身份证号属性，则身份证号也是码，可以在学号和身份证号中选定一个作为主码。通常情况下只关注实体的主码，所以，在不会引起混淆时，通常说码、主码、关键字或主关键字，含义都相同。

4. 域(Domain)

属性的允许取值的集合称为该属性的**域**。例如，学号的域是{7位数字}(某校规定)，性别的域是{男、女}，班级的域是该校所有班级的集合。

5. 实体型(Entity type)

具有相同属性的实体必然具有共同的特征。用实体名及其所有属性名集合来抽象和描述同类实体称为**实体型**。例如，学生(学号、姓名、性别、出生年月日、班级)就是一个实体型。

6. 实体集(Entity set)

同型实体的集合称为**实体集**。例如，某个学校(或某个班级)的全体学生就是一个实

体集。

7. 联系(Relationship)

信息世界的不同实体集之间和同一实体集内部都可能存在一定的**联系**。

数据世界的概念是和信息世界的概念相对应的,例如,**数据表**(Data table)是实体集的数据表示,**记录**(Record)是实体的数据表示,**数据项**(Item)是属性的数据表示。记录由若干数据项组成。

为了用计算机解决数据处理问题,人们必须先对现实世界的事物进行分析,将需要的信息及其存在的联系做科学的抽象,建立起能正确反映客观事物的概念模型。然后才能设计出理想的数据模型。

1.2.3 实体的联系

信息世界存在的联系有两种:一是同一个实体集内部的联系;二是不同实体集之间的联系。

1. 实体集内部的联系

实体集内部的联系通常指组成该实体的各属性之间的联系。表 1-1 和表 1-2 分别是"学生情况"和"选课及成绩"两个实体集。在表 1-1 中,不同实体的学号都不重复,即学号与实体间有一一对应关系。而不同实体的姓名(或出生年月日、或家庭所在地等)都有可能重复,即姓名(或出生年月日、或家庭所在地)与实体间没有一一对应关系。所以,学号就是学生情况这个实体集的关键字。在表 1-2 中,不同实体的学号(或课程号、或成绩)都有可能重复。但是,不同实体的学号+课程号则不可能重复。所以,学号和课程号这两个属性组成的属性组是选课及成绩这个实体集的关键字。

表 1-1 学生情况

学 号	姓 名	性别	出生年月日	家庭所在地	家庭人均月收入
0100001	冯东梅	女	1980-12-26	北京	1100
0100002	章 蕾	女	1979-2-18	上海	350
0100007	闻维祥	男	1979-2-24	天津	450
0100008	黎念真	女	1979-8-19	重庆	400
0100009	钟开才	男	1978-8-8	广东	400
0100117	江介敏	女	1978-6-8	湖北	600

表 1-2 选课及成绩

学 号	课程号	考试成绩
0100001	A002	85
0100001	B001	92
0100001	B022	78
0100001	C032	85
0100001	D012	78

续表

学　号	课程号	考试成绩
0100002	A002	90
0100002	B001	80
0100002	B022	98
0100002	C032	92
0100002	D012	89

2. 实体集之间的联系

对于两个不同的实体集 A 和 B,它们之间的联系通常有以下 3 种方式。

(1) 一对一联系(1∶1)

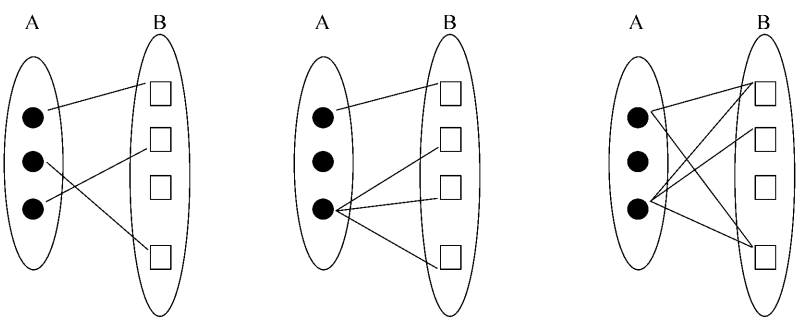
如果实体集 A 中的任一实体仅可能和实体集 B 中的一个实体相对应,并且实体集 B 中的任一实体也仅可能和实体集 A 中的一个实体相对应,则 A 和 B 间的联系就是一对一联系。例如,国家和首都、班级和班长、(飞机)乘客和座位都是一对一联系。

(2) 一对多联系(1∶n)

如果实体集 A 中至少有一个实体和实体集 B 中的两个或两个以上的实体相对应,而实体集 B 中的任一实体仅可能和实体集 A 中的一个实体相对应,则 A 和 B 间的联系就是一对多联系。例如,班级和学生、城市和道路都是一对多联系。

(3) 多对多联系(m∶n)

如果实体集 A 中至少有一个实体和实体集 B 中的两个或两个以上的实体相对应,并且实体集 B 中也至少有一个实体和实体集 A 中的两个或两个以上实体相对应,则 A 和 B 间的联系是多对多联系。以上 3 种联系可以用图 1-1 表示。



(a) 一对一联系　(b) 一对多联系　(c) 多对多联系

图 1-1　实体集间的联系

显然,一对一联系是一对多联系的特例,一对多联系又是多对多联系的特例。多对多联系直接处理起来很困难,通常是将多对多联系转化为两个一对多联系来处理。

概念模型和各种数据模型均不支持多对多联系,只支持一对一联系和一对多联系。

同一实体集内的各实体之间也可以存在一对一、一对多或多对多联系。例如,学生实体集内部存在“领导与被领导”的联系,即某一学生(班长)“领导”多名(全班)学生,而一个

学生仅被一个学生(班长)“领导”,这就是一对多联系。

在复杂问题中,两个以上的实体集之间也往往存在一对一、一对多或多对多联系。

1.2.4 概念模型

对于具体的实际问题,建立正确合理的概念模型是建立数据模型的前提。一个好的概念模型应该考虑和解决的问题是:

(1) 实际问题需要哪些实体集以及各个实体需要哪些属性。

(2) 这些实体集内部和实体集之间有怎样的联系。

(3) 如果存在多对多联系,如何将它转化为一对多联系。

概念模型的表示方法很多,其中最常用的是**实体—联系方法**(entity-relationship approach)。该方法用 **E-R 图**来描述。在 E-R 图中,实体型、属性和联系的表示方法如下:

(1) **实体型**:用矩形表示,矩形框内写实体名。

(2) **属性**:用椭圆表示,并用无向线段与相应的实体连接。

(3) **联系**:用菱形表示,菱形框内写明联系名,并用无向线段与有关的实体连接。同时在无向线段旁标上联系的类型(1∶1,1∶n 或 m∶n)。

应该指出,联系本身也是一种实体型,也可以有属性。如果一个联系有属性,也用无向线段将属性与该联系连接。图 1-2 用 E-R 图描述了两个实体之间的 3 种联系。图 1-3 用 E-R 图描述了“学生”和“课程”两个实体及其属性。

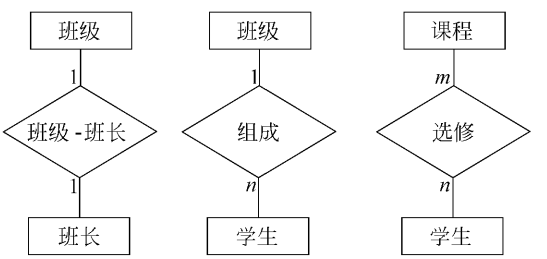


图 1-2 两个实体之间的三类联系

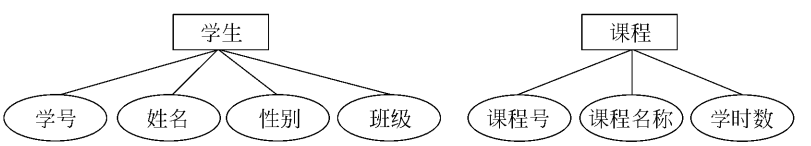


图 1-3 学生实体、课程实体及其属性

下面就学生学习成绩管理这一具体问题说明建立概念模型的方法和步骤,同时说明如何将多对多联系转化为一对多联系。

某高等院校实行学分制,每一个学生都可以选修几门课程,每一门课程也都有多个学生选修,并允许不同的学生选修不同的课程。

对于这个问题，显然要有学生和课程这两个实体集。学生实体集必须有“学号”作为其关键字，其他的属性根据需要确定，例如“姓名”和“班级”等。课程实体集必须有“课程号”作为其关键字，而“课程名称”是必不可少的属性。显然，学生和课程这两个实体集之间是多对多联系。如果只用这两个实体集，问题解决起来很困难。如前所述，这两者之间的联系本身也是一种实体型。现在，增加一个“选课”实体集来代替这个联系，在“选课”实体集中把“学号”和“课程号”集合起来作为关键字，再包括“成绩”属性。这样一来，学生和选课之间、课程和选课之间都是一对多联系。图 1-4 就是这个例子的概念模型。

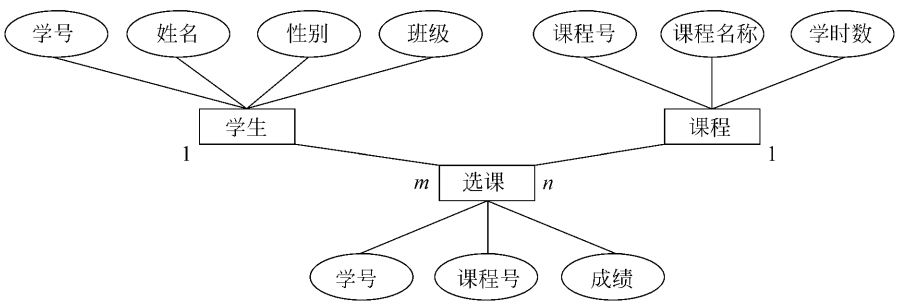


图 1-4　学生学习成绩管理概念模型

这个例子也给出了将多对多联系转化为一对多联系的一般方法，这就是：增加一个新的实体集，并且这个新的实体集和原来的两个实体集之间都是一对多联系。

1.2.5　层次模型

层次模型(Hierarchical Model)是数据库系统中最早出现的数据模型，它用树形结构表示各类实体以及实体间的联系。层次模型满足以下两个条件：

(1) 有且仅有一个结点无双亲结点，称之为**根结点**(简称**根**)；

(2) 根以外的其他结点有且仅有一个双亲结点。

层次模型适合于描述有主次之分的结构关系，对于具有一对多层次联系的描述非常直观，容易理解。用层次模型组织和查询数据也非常方便。但是，层次模型无法描述事物之间复杂的关系，只能反映实体之间一对多的联系。

图 1-5 所示是一个层次模型的典型。

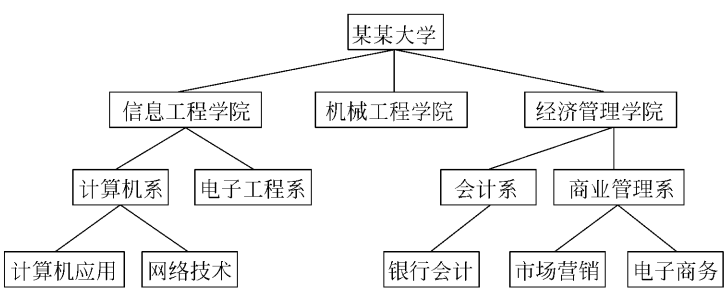


图 1-5　层次模型示意图

1.2.6 网状模型

在数据库中,**网状模型**(Network Model)是比层次模型更具普遍性的一种结构。它去掉了层次模型的两个限制条件,允许有一个以上的结点无双亲结点,允许结点可以有多个双亲结点,此外它还允许两个结点之间有多种联系(称之为复合联系)。因此,层次结构实际上是网状结构的特例。

网状模型有如下优点:

(1) 可以直接描述包括多对多在内的更复杂的关系;

(2) 具有良好的性能,存取效率较高。

但是,网状模型也有缺点:

(1) 对计算机的硬件和软件环境要求较高;

(2) 数据的独立性较差;

(3) 操作比较复杂。

图 1-6 表示零部件的采购—使用关系,这是一个典型的网状模型。

对 1.2.4 小节介绍的学生成绩管理问题,通过建立一个"选课"实体就可以把多对多的联系转化为"学生"与"选课"、"课程"与"选课"之间的两个一对多联系,其网状模型如图 1-7 所示。

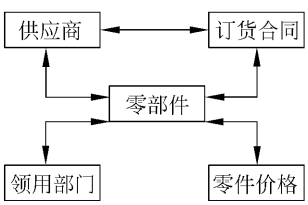


图 1-6 零部件的采购—使用关系

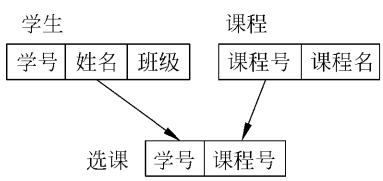


图 1-7 学生—选课关系

1.2.7 关系模型

关系模型(Relational Model)是数据库系统中最重要的一种模型。从用户的观点看,关系模型中数据的逻辑结构是一张二维表(以下简称表),它由行和列组成。例如,表 1-1 所示的学生情况就是一个关系模型。关系模型的主要概念如下:

关系(Relation):关系模型中表示数据的整个一张表就是**关系**。

元组(Tuple):表中的每一行即为一个**元组**,对应概念模型的一个实体。

属性(Attribute):表中的每一列即为一个**属性**,对应概念模型的一个属性。

主码(Primary Key):表中惟一标识元组的某个属性组称为该关系的**主码**(常简称为**码**),对应概念模型的码。

域(Domain):表中任一属性的取值范围称为该属性的**域**,对应概念模型的域。

分量(Component):元组中的每一个属性值称为元组的**分量**。

关系模式(Relation Schema):对关系的描述称为**关系模式**,通常用关系名及其所有属性名集合来表示,类似概念模型中实体型的表示方法。在不至于引起混淆的情况下,往

往将关系模式和关系统称为关系。

表 1-1 就是一个关系，表中的每一行都是一个元组，学号、姓名、性别等每一列都是属性，学号是这个关系的主码，{男、女}、正确的日期集合分别是性别和出生年月日的域，“0100001”、“冯东梅”、“女”、“1980-12-26”、“北京”和“1100”等都是分量。而学生(学号，姓名，性别，出生年月日，家庭所在地，家庭人均月收入)是这个关系的关系模式。

关系模型既可以反映属性之间一对多的联系，也可以反映属性之间多对多的联系，关系模型具有以下特点：

(1) 数据结构简单，概念清楚；

(2) 能够直接反映实体之间一对一、一对多和多对多联系；

(3) 通过公共属性就可以建立表与表之间的联系，从而就建立了实体之间的联系；

(4) 具有严格的数学理论基础。

1.2.8 面向对象模型

面向对象模型(Object Oriented Model)是数据库系统中最近出现的一种模型。面向对象模型中最基本的概念是**对象**(Object)和**类**(Class)。对象与关系模型中的元组的概念相似，但更复杂。每个对象都有惟一的标识符把对象的数据和操作封装在一起。共享同一属性集合和方法集合的所有对象组合在一起构成一个类。类具有嵌套结构。一个类从类层次的祖先那里继承(Inheritance)所有的属性和方法。面向对象模型是正在发展中的模型，具有广阔的前途和生命力。

1.3 关系的完整性

关系模型提供了丰富的完整性控制机制，允许定义三类完整性：**实体完整性**(Entity Integrity)、**参照完整性**(Referential Integrity)和**用户定义的完整性**(User-defined Integrity)。其中实体完整性和参照完整性是关系模型必须满足的完整性约束条件，被称为关系的两个**不变性**，由关系系统自动支持。这里先列出 3 个例题，以便后面引用。

【例 1-1】 学生实体和专业实体可以用下面的关系表示(其中主码用黑体字标识，下同)：

学生(**学号**，姓名，性别，专业号，年龄)
专业(**专业号**，专业名)

【例 1-2】 学生实体、课程实体、学生与课程之间的选课及成绩实体可以用如下 3 个关系来表示：

学生(**学号**，姓名，性别，专业号，年龄)
课程(**课程号**，课程名，学分)
选课及成绩(**学号**，**课程号**，成绩)

【例 1-3】 另一个学生实体的关系是：

学生 2(**学号**，姓名，性别，专业号，年龄，班长学号)

1.3.1 实体完整性

一个**基本关系**通常对应现实世界的一个实体集。例如,学生关系对应学生的集合。现实世界的实体是可区分的,即它们具有某种惟一性标识。相应地,关系模型中以主码作为惟一性标识。主码中的属性即主属性不能取空值。所谓空值就是“不知道”或“无意义”的值。如果主属性取空值,就说明存在某个不可标识的实体,即存在不可区分的实体,这与现实世界的应用环境相矛盾,因此这个实体一定不是一个完整的实体。

实体完整性规则:若属性 A 是基本关系 R 的主属性,则属性 A 不能取空值。

例如,在例 1-1 的学生关系中,学号为主码,则学号不能取空值。

实体完整性规则规定基本关系的所有主属性都不能取空值,而不仅是主码整体不能取空值。在例 1-2 的选课及成绩关系中,学号+课程号为主码,则学号和课程号两个属性都不能取空值。

1.3.2 参照完整性

现实世界中的实体之间往往存在某种联系,在关系模型中实体及实体间的联系都是用关系来描述的。这样就自然存在着关系与关系间的引用。

例 1-1 的两个关系之间存在着属性的引用,即学生关系引用了专业关系的主码“专业号”。显然,学生关系中的专业号值,必须是确实存在的专业的“专业号”,即专业关系中有该专业的记录。这就是说,学生关系中的某个属性的取值需要参照专业关系对应属性的取值。

例 1-2 的 3 个关系之间也存在着属性的引用,即选课及成绩关系引用了学生关系的主码“学号”和课程关系的主码“课程号”。同样,选课及成绩关系中的学号值必须是确实存在的学生的“学号”,即学生关系中有该学生的记录;选课及成绩关系中的课程号值,也必须是确实存在的课程的“课程号”,即课程关系中有该课程的记录。换句话说,选课及成绩关系中某些属性的取值需要参照其他关系对应属性的取值。

不仅两个或两个以上的关系间可以存在引用关系,同一关系内部属性间也可能存在引用关系。

在例 1-3 的关系中,“学号”属性是主码,“班长学号”属性表示该学生所在班级的班长的学号,班长必须是这个班的学生,即“班长学号”必须是确实存在的学生的“学号”。

设 F 是基本关系 R 的一个或一组属性,但不是关系 R 的码,如果 F 与基本关系 S 的主码 K 相对应,则称 F 是基本关系 R 的**外码**(Foreign Key),并称基本关系 R 为**参照关系**(Referencing Relation),基本关系 S 为**被参照关系**(Referenced Relation)或**目标关系**(Target Relation)。关系 R 和 S 不一定是不同的关系。

显然,目标关系 S 的主码 K_S 和参照关系 R 的外码 F 必须定义在同一个(或一组)域上。

在例 1-1 中,学生关系的“专业号”属性与专业关系的主码“专业号”相对应,因此“专业号”属性是学生关系的外码。这里专业关系为被参照关系,学生关系为参照关系。

在例 1-2 中,选课及成绩关系的“学号”属性与学生关系的主码“学号”相对应,“课程

号”属性与课程关系的主码“课程号”相对应。因此，“学号”和“课程号”属性是选课及成绩关系的外码。这里，学生关系和课程关系均为被参照关系，选课及成绩关系为参照关系。

在例 1-3 中，“班长学号”属性与本关系主码“学号”属性相对应，因此“班长学号”是外码。这里，学生 2 关系既是参照关系也是被参照关系。

图 1-8 清楚地表达了例 1-1、例 1-2 和例 1-3 的参照关系。

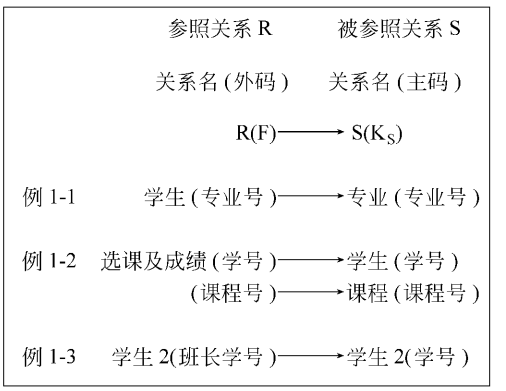


图 1-8　参照关系和被参照关系

这里需要指出，为了便于识别，在可能的情况下给外码与相应的主码取相同的名字。像例 1-1 和例 1-2 这样的情形，外码与相应的主码属于不同的关系，取相同的名字是可能的。但是，在复杂的关系中，即使外码与相应的主码属于不同的关系，要给外码与相应的主码取相同的名字也可能是困难的，甚至是不可能的，就只能取不同的名字。在不可能给外码与相应的主码取相同的名字时，只能取不同的名字，像例 1-3 这样外码与相应的主码属于同一个关系的情形。

参照完整性规则就是定义外码与主码之间的引用规则。

参照完整性规则：若属性(或属性组)F 是基本关系 R 的外码，它与基本关系 S 的主码 K_s 相对应(基本关系 R 和 S 不一定是不同的关系)，则对于 R 中每个元组在 F 上的值必须为：

- 或者取空值(F 的每个属性值均为空值)；
- 或者等于 S 中某个元组的主码值。

对于例 1-1，学生关系中每个元组的专业号属性只能取下面两类值：

- 空值，表示尚未给该学生确定所学专业；
- 非空值，这时该值必须是专业关系中某个元组的专业号值，表示该学生不可能分配到一个不存在的专业中学习。即被参照关系“专业”中一定存在一个元组，它的主码值等于该参照关系“学生”中的外码值。

对于例 1-2 中的选课及成绩关系，按参照完整性规则，学号和课程号属性也可以取两类值：空值或目标关系中已经存在的值。但由于学号和课程号是选课及成绩关系中的主属性，按照实体完整性规则，它们均不能取空值。所以选课及成绩关系中的学号和课程号属性实际上只能取相应被参照关系中已经存在的主码值。

参照完整性规则中,R 与 S 可以是同一个关系。例如,对于例 1-3,按照参照完整性规则,班长属性值可以取两类值:

- 空值,表示该学生所在班级尚未选出班长;
- 非空值,这时该值必须是本关系中某个元组的学号值。

1.3.3 用户定义的完整性

实体完整性和参照性适用于任何关系数据库系统。除此之外,不同的关系数据库系统根据其应用环境的不同,往往还需要一些特殊的约束条件。*用户定义的完整性就是针对某一具体关系数据库的约束条件,它反映某一具体应用所涉及的数据必须满足的实际要求,即取值的限制范围。*例如,某个非主属性不能取空值(如姓名)、某个属性的取值范围限定在 0～100 之间的数(如成绩)、某个属性只能取几个特定的值(如性别)等。关系模型应提供定义和检验这类完整性机制,以便用统一的系统的方法处理它们,而不要由应用程序承担这一功能。

1.4 关系模型的规范化

用关系模型描述现实世界直观、明了。但是,要做到用关系模型很好地描述现实世界却不是一件容易的事情。如前所述,设计一个好的数据库是一项极其复杂的工作,规范化理论就是设计过程中的一个非常有用的辅助工具。这里只简单介绍几个低级别的范式。对规范化理论有兴趣者可参考其他书籍。

一般地说,关系模型的范式级别越高,设计的数据结构质量越高。第一范式是关系模型规范化最基本的要求,第二范式的级别比第一范式高,第三范式的级别又比第二范式高。

1.4.1 第一范式

如果关系模式 R 的每一个关系的属性值都是不可分的原子值,则称 R 属于**第一范式**(1NF)。

属于 1NF 的关系称为规范化关系,不属于 1NF 的关系称为非规范化关系。

不难验证,表 1-1 和表 1-2 所示的实体集都是属于 1NF 的规范化关系。但是实际问题中的报表结构有许多是非规范化的。例如,某高校研究生指导工作规定一位导师可以指导多名研究生,一个研究生只能师从一位导师。表 1-3 清楚地说明了研究生和他的导师之间的关系。但是,表 1-3 是不规范的,因为一位导师的研究生有多名,不是不可分的原子值。不能依这样的结构设计关系模型,必须将这样的表所需要的数据(以及其他数据)进行处理,使之规范化。从规范化的关系模型的数据库是完全能够生成像表 1-3 这样结构的报表的。

一个关系应满足 1NF 是最起码的条件。然而,仅满足 1NF 的关系还可能存在一些问题。

表 1-3 指导研究生

导师姓名	专 业	研究生	
		研究生 1	研究生 2
林知荣	网络系统	田 园	刘 刚
周银华	数据库技术	曲彩燕	
黎 祥	数据库技术	邱罗耕	方 萍

假设有这样一个关系模式：教师任课(教师编号、教师姓名、课程号、课程名称)，它描述教师和他能承担教学的课程的关系。表 1-4 是一个具体实例。

表 1-4 教师任课

教工编号	姓 名	课程号	课程名称
10013	罗开才	C031	计算机应用基础
10013	罗开才	C032	数据库原理及应用
10013	罗开才	C043	操作系统
10036	刘 霞	C031	计算机应用基础
10036	刘 霞	C032	数据库原理及应用
10036	刘 霞	C051	软件工程
10038	欧阳华	D001	统计原理
10038	欧阳华	D012	经济法基础

教师任课的主码为(教师编号、课程号)，根据关系中的教师编号和课程号就确定了教师的姓名和他所能讲授的课程名称。该关系的实质是：

(1) 每个教师有惟一的编号，每一门课程有惟一的代号；

(2) 每个教师可以授多门课程，每一门课程可由多个教师讲授；

(3) 每个教师编号有惟一的教师姓名，每一门课程号有惟一的课程名称。

这个关系是满足 1NF 的，但在操作中可能出现如下的一些问题。

(1) 插入异常

假定有一门新的课程需要开设，但还没有确定讲授教师。如果要插入一行存储该课程的代号和名称时，该行的教师编号和姓名就没有数据可输入，只能设为空值。而教师编号作为该关系的主码的一部分是不允许空值的，因而该记录无法插入，这称为插入异常。

(2) 删除异常

假定教师刘霞调离，不能讲授软件工程等课程。如果要将表 1-4 中刘霞的信息删除，则必然也删除了软件工程的信息。这就将不该删除的信息删除了，这称为删除异常。

(3) 数据冗余

正因为一个教师可以讲授多门课程，一门课程可以由多个教师讲授，因而在表 1-4 所示的关系中，同样的教师姓名或同样的课程名称多次存储。同样数据的多次存储造成了数据存储的冗余。数据的冗余不仅浪费了存储空间，还可能因为多次输入相同数据可能产生的差错引起混乱。当然，系统还要付出很大的代价来维护数据的完整性，比如，某门课的名称变了，就要修改有关此课程的所有元组。

1.4.2 第二范式

如果关系模式 R 属于 1NF,并且每个非码属性都完全依赖于各个码,则称 R 属于**第二范式**(2NF)。如果某个数据库模式中的每个关系模式都是 2NF,则这个数据库模式称为 2NF 的数据库模式。

例如,对于表 1-5 所示的关系模式 R(学号,系名,系办地址,课程号,成绩),主码为学号+课程号。某个学生某门课程的成绩(一个非码属性)是完全依赖于他的学号和该课程的代号(主码)的。也就是说,知道了一个学生的学号和他参加考试的课程号(两者缺一不可),就知道了他这一门课的成绩。例如,学号为 0100001 的学生,参加课程号为 B001 的课程考试,其成绩为 92。而他所在的系名(另一非码属性)实际上仅完全依赖于他的学号(主码中的一个属性),与课程代号(主码中的另一个属性)无关。也就是说,只要知道了一个学生的学号(并不需要知道课程号),就知道了他所在的系名和系办地址。例如,学号为 0100001 的学生所在的系为电子工程系。所以,这个关系模式不属于 2NF(仅属于 1NF)。

表 1-5 关系 R

学 号	系 名	系办地址	课程号	成绩
0100001	经济管理系	经管大楼	A002	85
0100001	经济管理系	经管大楼	B001	92
0100001	经济管理系	经管大楼	B022	78
0100001	经济管理系	经管大楼	C032	85
0100001	经济管理系	经管大楼	D012	78
0100002	经济管理系	经管大楼	A002	90
0100002	经济管理系	经管大楼	B001	80
0100002	经济管理系	经管大楼	B022	98
0100002	经济管理系	经管大楼	C032	92
0100002	经济管理系	经管大楼	D012	89
0100096	电子工程系	电子大楼	A002	80
0100096	电子工程系	电子大楼	B001	75
0100096	电子工程系	电子大楼	D001	45
0100096	电子工程系	电子大楼	D012	28

如果将 R 分解为两个关系模式 R1(学号,系名,系办地址)和 R2(学号,课程号,成绩)(表 1-6 和表 1-7),则 R1 和 R2 都属于 2NF 了。这样分解之后,减少了冗余,节省了存储空间,降低了数据维护的代价。但是,另一方面是增加了查询难度。如果要同时查询某个学生的学号、系名、系办地址、课程号、成绩,在 R 中查最为快捷,在 R1 和 R2 中查就需要在 R1 和 R2 之间做连接操作,比单从 R 中查复杂。这说明,在确定数据模型时,不能简单地说采用第二范式一定比第一范式好,采用第三范式也一定比第二范式好。到底采用哪个范式,要根据具体问题的要求确定。

表 1-6 关系 R1

学 号	系 名	系办地址
0100001	经济管理系	经管大楼
0100002	经济管理系	经管大楼
0100096	电子工程系	电子大楼

表 1-7 关系 R2

学 号	课程号	成绩	学 号	课程号	成绩
0100001	A002	85	0100002	B022	98
0100001	B001	92	0100002	C032	92
0100001	B022	78	0100002	D012	89
0100001	C032	85	0100096	A002	80
0100001	D012	78	0100096	B001	75
0100002	A002	90	0100096	D001	45
0100002	B001	80	0100096	D012	28

1.4.3 第三范式

如果关系模式 R 是 2NF,且每个非主属性都完全直接依赖于各个码(即既不部分依赖于码也不传递依赖于码),则称 R 属于**第三范式**(3NF)。如果某个数据库模式中的每个关系模式都是 3NF,则这个数据库模式称为 3NF 的数据库模式。

如果一个关系不属于第三范式,那么也可能存在一些问题。

在关系 R2 中,主码为学号+课程号,成绩是完全直接依赖于他的学号和该课程的代号(主码)的,故 R2 属于 3NF。

在关系 R1 中,主码为学号,知道了一个学生的学号(主码),就知道了他所在的系名,知道了他所在的系名(非码属性),也就知道了该系办公室的地址。系名是完全直接依赖于他的学号的。但系办公室的地址是通过系名间接(不是直接)依赖于他的学号的,故 R1 不属于 3NF。正因为关系 R1 不属于 3NF,也存在数据冗余和操作异常现象。例如,有多个学生同属于一个系,这个系的地址就多次重复。如果将关系 R1 分解为 R11 和 R12(表 1-8 和表 1-9),则 R11 和 R12 就都是属于 3NF 了。

表 1-8 关系 R11

学 号	系 名
0100001	经济管理系
0100002	经济管理系
0100096	电子工程系

表 1-9 关系 R12

系 名	系办地址
经济管理系	经管大楼
电子工程系	电子大楼

除了上面介绍的三个范式外,还有其他更高级别的范式。一个数据库满足的范式越多,数据的冗余度越小,共享性越高,所占的存储空间越少,并将数据的不一致性减少到最

低程度。但是,高范式的数据库查询起来比较复杂。所以,不应一味追求高范式,对于包含频度很高的查询的数据库,不妨降低些范式。

1.5 小结

要知道数据库发展的四个阶段及其特点、数据、数据库、数据库管理系统、数据库系统等基本概念,关系模型的主要概念和关系模型特点。

要知道问题、事物和事物的特征之间的关系以及编码的重要性。

要深刻理解主码的定义和作用。

要深刻理解实体集之间的三种联系,掌握将多对多联系转化为一对多联系的方法。

要理解关系的三类完整性的实质,尤其是主码和外码的取值限制条件。

要懂得关系模型规范化的三个范式的含义,并能结合实际问题正确运用。

1.6 习题

1.6.1 单项选择题

1-1 计算机数据管理技术发展在几个阶段中,数据独立性由低到高的顺序是_____。

(A) 文件系统阶段、人工管理阶段、数据库系统阶段

(B) 人工管理阶段、文件系统阶段、数据库系统阶段

(C) 数据库系统阶段、人工管理阶段、文件系统阶段

(D) 数据库系统阶段、文件系统阶段、人工管理阶段

1-2 数据库系统与文件系统的主要区别是_____。

(A) 文件系统管理的数据量小,而数据库系统管理的数据量大

(B) 数据库系统中的数据共享性差、冗余度大,而文件系统中的数据共享性好、冗余度小

(C) 文件系统中的数据共享性差、冗余度大,而数据库系统中的数据共享性好、冗余度小

(D) 文件系统中的数据不能长期保存,而数据库系统中的数据能长期保存

1-3 通常都需要对具有有限数据集合的特征进行编码。下面不属于这种编码效果的是_____。

(A) 节约存储空间　　(B) 便于阅读理解

(C) 减少出错的可能　　(D) 便于查询

1-4 数据库系统的核心是_____。

(A) 硬件　　(B) 软件　　(C) 人员　　(D) 数据

1-5 按照一定数据模型组织、存储和应用的相互联系的数据集合称为_____。

(A) 数据库系统　　(B) 数据库管理系统