上篇　数据库基础

第一章　数据库系统概述

20 世纪 60年代末，数据库技术初露头角，随即得到迅速发展，成为数据处理的公用支撑技术。目前全世界已有成千上万个数据库系统在运行，它们几乎触及到人类社会生活的所有方面，无论是企事业内部的信息管理，还是各行业的业务处理，以及一般的信息加工和情报检索无不以数据库技术为基础。近年来随着分布处理、高速网络、多媒体、数据压缩等一批高新技术的发展，数据库技术的应用更加普遍深入。

第一节　数据库系统

数据库系统完成类似于图书馆的工作，主要包括以下功能：

（1）建立数据模型，使用户可以根据数据模型访问数据库中的数据（如检索、插入、删除和修改），而不必关心数据的物理存储位置，就像读者可以按书卡填写借书单而不用顾及书籍存放在书库的什么位置一样。

（2）数据应有组织地存放在存储设备上，并建立数据模型到物理存储位置的映射，它使系统能够按照用户的访问请求，找到被访问数据的存储位置。建立数据模型和设计数据的物理存储（组织）方法，其目的是使用户对数据的应用与数据的存放位置和存储结构无关，后者的变动不影响前者，这称为数据独立性。

（3）数据库系统要为不同用户确定不同的访问权限并进行访问控制。

（4）提供以数据库为基础的各种应用服务。

上述各种功能都是在数据库管理系统（DBMS）的统一管理和控制下实现的。正像一个大型公共图书馆需要有专门的工作人员负责规划、设计、协调、维护和管理一样，数据库系统一般设有数据库管理员（DBA），负责协调和监视数据库的使用，一旦发现违反安全保密或性能下降的现象，立即采取相应对策。因此，一个数据库系统由数据库、数据库管理系统、应用程序和数据库管理员四部分组成，如图 1-1 所示。在不引起混淆的情况下，数据库系统有时也简称数据库。

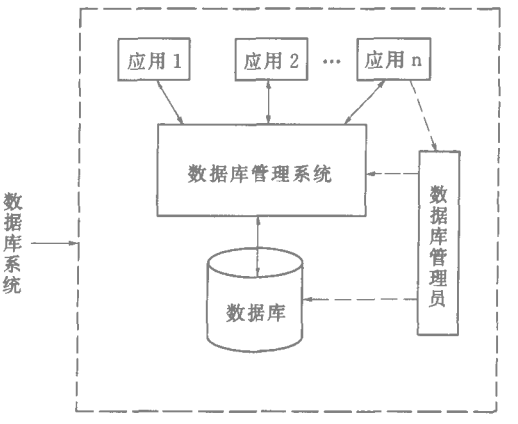


图 1-1　数据库系统环境

需要全本请访问 www.ertongbook.com

概括起来说，数据库系统是实现有组织地、动态地存储大量关联数据，支持多用户访问的计算机软、硬件资源及 DBA 组成的系统。

第二节　现实世界的数据描述

一、信息的三个领域

在现实世界中，信息处于三个领域：现实世界、观念世界和数据世界。①现实世界是存在于人们头脑之外的客观世界，事物及其相互联系就处于这个世界中。②观念世界是现实世界在人们头脑中的反映。客观事物在观念世界中被称为实体，反映事物联系的是实体模型。③数据世界是观念世界中信息的数据化。现实世界中的事物及联系在这里用数据模型描述。三个领域的内容及其联系如图 1-2 所示。

可见，客观事物是信息之源，是设计数据库的出发点，也是使用数据库的最终归宿。实体模型与数据模型是对客观事物及其联系的两级抽象描述。数据库的核心问题是数据模型，数据模型由实体模型导出。

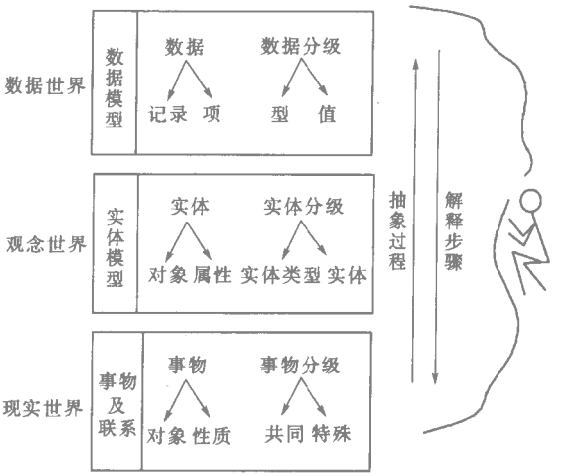


图 1-2　信息的三个领域

二、实体模型

实体模型是设计数据库的先导，首先设计者必须与用户合作，开列用户问题表，而后对现实世界中所要考虑的实体及其联系进行模拟，建立一个正确反映客观事物的实体模型。设计实体模型常用的工具是 E—R 图。

1. 实体与实体类型

实体是客观世界中存在的且可相互区分的事物，实体可以是人也可以是物；可以是具体事物也可以是抽象概念。把具有共性的实体抽象为实体类型（在不引起误解时，实体类型有时也简称实体),在 E—R 图中用矩形框表示。

2. 联系

实体之间往往存在各种关系，这种实体间的关系抽象为联系，在 E—R 图中用菱形框表示。设有 A、B 两个实体类型，其间建立的联系可分为下面三类：

（1）一对一联系（1∶1)。如果 A 中的任一实体至多对应于 B 中的一个实体；反之，B 中的任一实体，至多对应于 A 中的一个实体，则称 A 对 B 是一对一的联系，例如乘客与车票、病人与床位之间的联系。

（2）一对多联系（1∶N)。如果 A 中至少有一个实体对应于 B 中一个以上实体；反之，B 中任一个实体至多对应于 A 中的一个实体，则称 A 对 B 是一对多联系，例如城市对街道、班级对学生之间的联系。

（3）多对多联系（M∶N)。如果 A 中至少有一个实体对应于 B 中一个以上实体：反之，

B 中也至少有一个实体对应于 A 中一个以上实体，则称 A 对于 B 是多对多联系，例如学生与课程、商店与顾客的联系。

3. 属性

属性是实体或联系所具有的性质。每个属性有其取值的范围，称为域。属性又分为原子属性和组合属性，原子属性的域由简单值组成，组合属性的域由结构值组成。属性还可分为单值或多值，单值属性对应域中的元素为单个值，多值属性对应域中的元素为集合，如一个人所获学位可定义为多值属性。在 E—R 图中，属性用椭圆（或圆角矩形）表示，属性与实体（联系）之间用实线连接，组合属性用一个树型结构表示，多值属性用虚椭圆标出。

基于上述概念，现以某校教学管理为例建立实体模型，如图 1-3 所示。

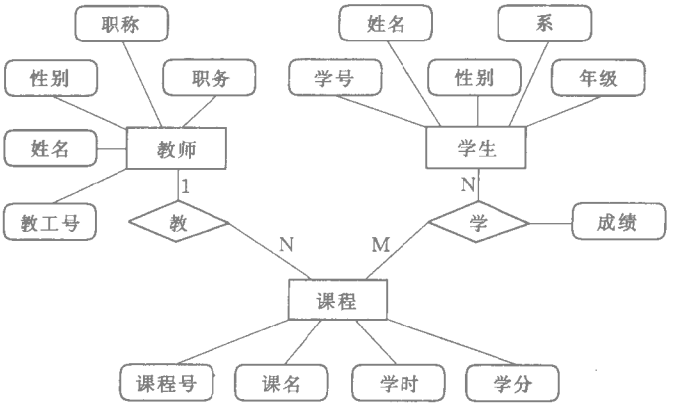


图 1-3 某学校教学管理 E—R 图

三、数据模型

数据模型是对客观事物及其联系的数据描述，即实体模型的数据化。设计一个好的数据模型是数据库设计的核心问题之一。下面简略介绍设计数据模型的有关问题。

1. 记录与数据项（简称项）

数据模型中把描述实体的数据称为记录，把描述属性的数据称为项，记录由若干项组成。项又分成基本项（原子属性）与组合项（组合属性），基本项是具有名称的最小逻辑数据单位，组合项由基本项与组合项构成。

2. 类型与值

在数据模型中将由相同数据项组成的记录抽象为记录类型，称具有某一记录类型的记录为该类型的记录值。直观地说，记录类型是一个框架，只有给它的每个数据项取值后才得到记录。就如一张学生登记表，填写前只是学生记录类型，填写后就得到一个学生记录。数据项也有型与值之分。如数据项“年龄”的型是“名称为年龄，数据特征为三位十进制数”，值是‘1 岁、2 岁、……、 120 岁等”（一般都规定数据项的取值范围，超出值域的值认为无实际意义）。

表示记录类型有图示和使用数据语言描述两种方法，图 1-4 为用图示法描述的学生记录类型。在图示法中，记录类型用方框图

学生



图 1-4 用图示法描述的学生记录类型

表示，它的名称写在方框左上方，各数据项名按顺序写于方框图的小格中。

3. 记录与文件

文件定义为记录型与值的总和，一个记录类型和它的一些当前记录组成文件。在文件中，如果两个记录至少有某个数据项的值不同，则称为不同记录；如果它们对应数据项的值都相同，则称为相同记录。文件中不允许有相同记录。把其值能够惟一标识记录的一个或多个数据项称为记录类型（或文件）的关键字（Key），把用于组织文件的关键字称为主关键字（Primary Key）。图 1-5 就是一个学生文件，其中数据项学号就是关键字。

学生

学号	姓名	出　生			性别
		年	月	日	
98001	王一	1981	8	30	女
98002	李波	1980	6	5	男

图 1-5　一个学生的文件

4. 数据模型

建立了实体模型后，就可以着手建立数据模型了。由于实体之间存在着复杂的联系，所以描述它们的数据之间也存在着复杂的联系。数据模型一般包括数据的基本结构、数据间的联系和数据中的约束。整个模型就像一个框架，填上具体的数据值就得到数据模型的一个实例。

为了使模型能清晰、准确地反映客观事物，并能用于数据库设计，一般应做到以下几点：

（1）给数据模型命名，使不同模型得以区别。

（2）给每个记录类型命名，以标识和说明同一模型中具有的记录类型。

（3）给每个数据项命名，以说明和区分每个记录类型具有的数据项，并确定作为记录类型主关键字的数据项。

（4）说明各个记录类型之间的联系，必要时给这种联系命名。

（5）必要时，指出各数据项的数据特征，即类型、长度、值域。

（6）指出记录类型和联系应满足的约束。

数据模型的设计方法决定数据库的设计方法，传统的数据模型设计方法有关系、层次和网络三种，这里只介绍目前使用最广泛的关系模型（Relational model）。

直观地说，数据的关系模型是若干“关系框架”组成的集合，它的实例由若干“关系”组成，而关系则由关系框架和若干“元组”构成。关系框架相当于前述的记录和联系类型，关系相当于文件，元组相当于记录值。现仍以图 1-3 的教学实体模型为例，来构造其关系数据模型及可能的一个实例，如图 1-6 所示。

关系模型的最大特色是描述的一致性，对象及其联系均用关系描述，或者通过关系之间的连接运算来建立某种联系。根据图 1-6 的关系模型设计的数据库可以回答下列询问：

（1）各个实体的情况。如“学生李波的年龄”一类的询问。

（2）实体之间的联系情况。如“操作系统的开课教员是谁”一类询问。

（3）具有某种属性的实体。如“男同学的人数与姓名”，“某门课程分数 ≥80 分的学生人数与姓名”一类询问。

（4）满足某种条件属性情况。如“李波所学课程的任课教员的职称”一类询问。

（5）某些对象的统计情况。如“学生的平均年龄”，“高等数学最低分数”一类询问。

学生关系框架

学号	姓名	年龄	性别

课程关系框架

课程号	课程名	学时数

教员关系框架

姓名	年龄	职称

学习关系框架

学号	课程号	分数

任课关系框架

课程号	任课教员

学生关系

学号	姓名	年龄	性别
98601	王晓燕	20	女
98602	李　波	23	男
98603	陈志坚	24	男
98604	张　兵	20	男
98605	张　兵	21	女

课程关系

课程号	课程名	学时数
C601	高等数学	200
C602	数据结构	120
C603	操作系统	80
C604	编译原理	80

教员关系

姓名	年龄	职称
陈　华	35	讲　师
刘建平	44	讲　师
王志伟	47	副教授
吴　浩	23	助　教
张为公	56	副教授
周振兴	51	讲　师

学习关系

学号	课程号	分数
98601	C601	90
98601	C602	90
98601	C603	85
98601	C604	87
98602	C601	90
98603	C601	75
98603	C602	70
98603	C604	56
98604	C601	90
98604	C602	85
98605	C601	95
98605	C604	80

任课关系

课程号	任课教员
C601	周振兴
C602	刘建平
C603	刘建平
C604	王志伟

图 1-6　数据的关系模型与实例

第三节　数据库的体系结构

一、数据库的分级结构

数据库分为三级：外模式、概念模式和内模式，如图 1-7 所示，掌握数据库的三级结构及其联系与转换是深入数据库的必由之路。

用户级数据库也称为用户视图（View）或子模式，对应于外模式，它是单个用户看到并获准使用的那部分数据的逻辑结构（称为局部逻辑结构），用户根据系统给出的子模型，用询问语言或应用程序去操作数据库中的数据。

概念级数据库对应于概念模式，简称模式，是对数据库所有用户的数据的整体逻辑描述(故称为数据库的整体逻辑结构),又称为 DBA 视图，即数据库管理员看到的数据库，它是所有用户视图的一个最小并集。设立概念级的目的是为了把用户视图有机地结合成一个逻辑整体，统一地考虑所有用户要求，它涉及的仍然是数据库中所有对象的逻辑关系，而不是它们的物理情况。

物理级数据库对应于内模式，又称为存储模式。它包含数据库的全部存储数据，这些

需要全本请访问 www.ertongbook.com

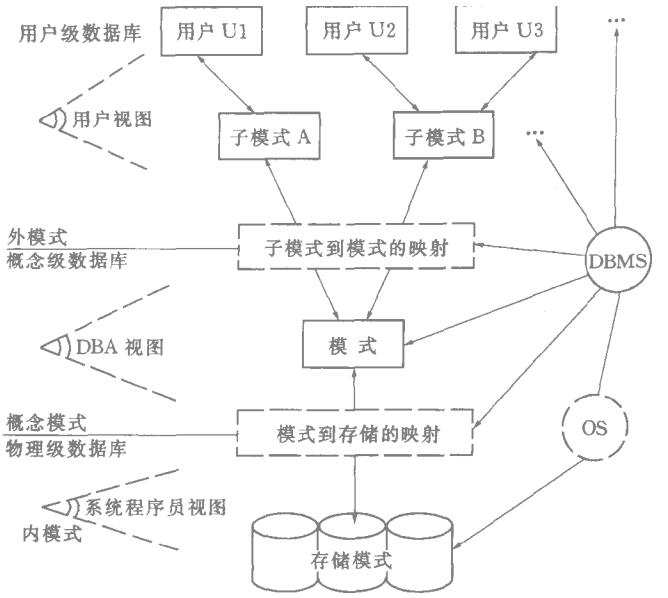


图 1-7　数据库的分级结构

被存储在内、外存介质上的数据也称为原始数据，是用户操作（加工）的对象。从机器的角度看，它们是指令操作处理的位串、字符和字；从系统程序员的角度看，这些数据是他用一定的文件组织方法组织起来的一个个物理文件（或存储文件），系统程序员编制专门的访问程序，实现对文件中数据的访问，所以物理级数据库也称为系统程序员视图。

对一个数据库系统来说，实际上存在的只是物理级数据库，它是数据访问的基础。概念级数据库只不过是物理级数据库的一种抽象（逻辑）描述，用户级数据库是用户与数据库的接口。用户根据子模式进行操作，数据库管理系统通过子模式到模式的映射将操作与概念级联系起来，又通过模式到存储模式的映射与物理级联系起来。这样一来，用户可以在较高的抽象级别上处理数据，而把数据组织的物理细节留给系统。事实上，DBMS 中心工作之一就是完成三级数据库之间的转换，把用户对数据库的操作转化到物理级去执行。

二、模式及映射

1. **模式**

模式是用模式定义语言（DDL）给出的数据库整体逻辑描述，包括逻辑记录类型和记录之间的关系，其主体是数据模型。此外，一般还包括允许的操作、数据完整性和安全保密等方面的控制。

用语言书写的模式称为源模式，机器不能直接使用，必须将此模式用机器代码表示，变为机器书写的模式，称为目标模式。目标模式通常设计成表格形式、树结构或网络结构。

2. **子模式**

子模式是用户与数据库的接口，一般每个用户使用一个子模式，但多个用户（特别是同类型的用户）也可以使用同一个子模式。从逻辑关系上看，子模式是模式的一部分，从

模式用某种规则（如关系方法中的关系运算）可以导出子模式，但子模式也可以做出不同于模式的改变，如在子模式中略去模式的某些记录类型、数据项，改变模式中某些数据项的数据库特征，改变模式中的安全、完整约束条件等。子模式与模式之间的对应关系称为子模式到模式的映射，描述子模式的语言称为子模式 DDL，子模式也有相应的源形式与目标形式。

现以图 1-6 所示教学数据库系统的关系模型为例子，给出两个可能的子模式，如图 1-8。注

学生关系框架

学号	姓名	年龄	性别

教员关系框架

姓名	职称

学生关系

学号	姓名	年龄	性别
98601	王晓燕	20	女
98602	李　波	23	男
98603	陈志坚	21	男
98604	张　兵	20	男
98605	张　兵	21	女

教员关系

姓名	职称
陈　华	讲　师
刘建华	讲　师
王志伟	副教授
梁　浩	助　教
张为公	副教授
周振兴	讲　师

(a)

学生关系框架

学号	姓名	年龄	性别

授课关系框架

课程号	课程名	学时数	任课教员

学习关系框架

学号	课程号	分数

学生关系

学号	课程号	分数
98601	C601	90
98601	C602	95
98601	C603	85
98601	C604	87
98602	C601	90
98603	C601	75
98603	C602	70
98603	C604	56
98604	C601	90
98604	C602	85
98605	C601	95
98605	C604	80

授课关系

课程号	课程名	学时数	任课教员
C601	高等数学	200	周振兴
C602	数据结构	120	刘建平
C603	操作系统	80	刘建平
C604	编译原理	80	王志伟

学生关系

学号	姓名	年龄	性别
98601	王晓燕	20	女
98602	李　波	23	男
98603	陈志坚	21	男
98604	张　兵	20	男
98605	张　兵	21	女

(b)

图 1-8

(a) 子模式 A 及其数据；(b) 子模式 B 及其数据

意,使用子模式 A 的用户只能访问子模式 A 对应的数据，使用子模式 B 的用户只能访问子模式 B 对应的数据。

3. 存储模式

存储模式是数据在物理存储结构方面的描述，它除了定义所有的内部记录类型外，还定义一些索引、存储分配以及恢复等方面的细节。但更具体的物理存储细节，如从磁盘读、写某些数据块等，存储模式一般不予考虑，而是交给操作系统完成。

描述存储模式的语言称为存储模式 DDL，存储模式也有源模式与目标模式之分，其目标模式与模式和子模式的目标形式都存在于系统数据字典中，供系统随时查用。

4. 模式到存储模式的映射

模式到存储模式的映射主要给出概念级数据与物理级数据之间的对应关系。表现在两个方面；一方面是数据结构的变换，即逻辑记录及组成记录的数据项如何对应到内部记录和数据项上去；另一方面是逻辑数据如何在物理设备上定位。在概念级与内部级之间存在这样一个映射，就能保证当存储模式发生变化时（如存储设备、文件组织方法、存储位置等发生变化），只要修改此映射即可，而模式尽量不受影响，从而对子模式和应用程序的影响更小，达到所谓物理数据独立性。通常情况下，模式到存储模式的映射在存储模式中描述。

5. 子模式到模式的映射

子模式到模式的映射主要给出外部级与概念级的对应关系。这两级的数据结构以及数据量纲可能不一致，子模式中某些数据项甚至是由若干数据导出的，在数据库中并不真实存在。因此在映射中需要说明子模式中的记录类型和数据项如何对应着模式中的记录和数据项以及导出的规则步骤（若为导出项）。在外部级与概念级之间存在这样一个映射，就能保证当模式发生变化时，只要修改记录及数据项的对应关系或导出规则，可能不必修改子模式，用户根据子模式设计的应用程序就可继续沿用，达到所谓逻辑数据独立性。通常情况下，子模式到模式的映射在模式中描述。

第四节　数据库管理系统

数据库管理系统(DBMS)是一个非常复杂的软件系统，是使数据库系统具有数据共享、并发访问、数据独立等特性的根本保证，对数据库系统的所有操作和各种运行控制最终都是通过 DBMS 实现的。

一、 DBMS 的功能

DBMS 的主要职责就是有效地实现数据库三级之间的转换，即把用户（或应用程序）对数据库的一次访问，从用户级带到概念级，再导向物理级，转换为对存储数据的操作。因此其功能应包括下面几类。

1. 数据库的定义

DBMS 总是提供数据定义语言 DDL 用于描述模式、子模式和存储模式及其模式之间的映射，描述的内容包括数据的结构、数据的完整约束条件和访问控制条件等，并负责将这些模式的源形式转换成目标形式，存在系统的数据字典中，供以后操作或控制数据时查

用。

2. 数据库的操作及查询优化

DBMS 总是提供数据操作语言 DML 实现对数据库的操作，基本操作包括检索、插入、删除和修改。用户只需根据子模式给出操作要求，其处理过程的确定和优化则由 DBMS 完成。查询处理和优化机制的好坏直接反映 DBMS 的性能。

3. 数据库的控制运行

数据库方法的最大优势在于允许多个用户并发地访问数据库，充分实现共享， DBMS 必须提供并发控制机制、访问控制机制和数据完整性约束机制，从而避免多个读写操作并发执行可能引起的冲突、数据失密或安全性、完整性被破坏等一系列问题。

4. 数据库的恢复和维护

这些维护信息可将数据库恢复到一致状态。此外，当数据库性能下降或系统软硬件设备变化时，也能重新组织或更新数据库。

5. 数据库的数据管理

数据库中物理存在的数据包括两部分：一部分是元数据，即描述数据的数据，主要是前述的三类模式，它们构成数据字典（DD）的主体，DD 由 DBMS 管理、使用；另一部分是原始数据，它们构成物理存在的数据库， DBMS 一般提供多种文件组织方法，供数据库设计人员选用。数据一旦按某种组织方法装入数据库，其后对它的检索和更新都由 DBMS 的专门程序完成。

6. 数据库的多种接口

一个数据库一旦设计完成，可能供多类用户使用，包括常规用户、应用程序的开发者，DBA 等。为适应不同用户的需求， DBMS 常提供各种接口，近年来还普遍增加了图形接口，用户使用起来更直观、方便。

二、 DBMS 的程序组成

从程序的角度看，DBMS 是完成上述各项功能的许多程序模块组成的一个集合，其中一个或几个程序一起完成 DBMS 的一件工作，或一个程序完成几件工作，以设计方便和系统性能良好为原则，所以各个 DBMS 的功能不完全一样，包含的程序也不等。其主要程序如下。

1. 语言处理方面

(1) 模式 DDL 翻译程序，把模式 DDL 源形式翻译成机器可读的目标形式。

(2) 子模式 DDL 翻译程序，把子模式 DDL 源形式翻译成目标形式。

(3) DML 处理程序，把应用程序的 DML 语句转换成主语言的一个过程调用语句。

(4) 终端询问解释程序，解释终端询问的意义，决定操作的执行过程。

(5) 数据库控制命令解释程序，解释每个控制命令的含义，决定怎样执行。

2. 系统运行控制方面

(1) 系统总控程序是DBMS 的神经中枢，它控制、协调 DBMS 各个程序的活动，使其有条不紊地运行。

(2 访问控制程序，其内容包括核对用户标识、口令，核对授权表，检验访问的合法性等，它决定一个访问是否能够进入系统。

（3）并发控制程序，在许多用户同时访问数据库时协调各个用户的访问。

（4）保密控制程序，在执行操作之前，核对保密规定。

（5）数据完整性控制程序，在执行操作前或后，核对数据库完整约束条件，从而决定是否允许操作执行，或清除已执行操作的影响。

（6）数据访问程序，根据用户访问请求，实施对数据的访问。

（7）通信控制程序，实现用户程序与 DBMS 之间的通信。

3. 系统建立、维护方面

（1）数据装入程序，把大批原始数据按某种文件组织方法（顺序、索引、 Hash 等）存储到内外存介质上，完成数据库的装入。

（2）工作日志程序，负责记载进入数据库的所有访问。

（3）性能监督程序，监督操作执行时间与存储空间占用情况，做出系统性能估算，以决定数据库是否需要重新组织。

（4）重新组织程序，当数据库系统性能变坏时需要对数据重新进行物理组织，或者按原组织方法重新装入，或者改变原组织方法，采用新的结构。

（5）系统恢复程序，当软、硬件设备遭到破坏时，该程序把数据库恢复到可用状态。

4. 用户接口方面

目前不少 DBMS 的接口软件的规模往往超过 DBMS 的核心软件。

三、数据语言

数据语言包括数据描述语言 DDL 和数据操作语言 DML 两大部分，前者负责描述和定义数据的各种特性，后者说明对数据进行的操作。

1. 数据描述语言 DDL

数据描述语言用于描述数据库中各种对象的特征，其功能包括以下四个方面。

（1）描述数据的逻辑结构。因为数据库的逻辑描述限于概念级和用户级，所以在数据库中又分模式数据描述语言和子模式数据描述语言。前者描述模式，它必须具备定义数据模型的功能和容易阅读的优点，是一种与现行普通程序语言不同的格式化的陈述性语言，在采用 SQL 标准关系数据库系统中，模式 DDL 都是统一的。后者描述子模式，它可以是用户使用的程序语言（如 C、Fortran 等）的扩充。不论怎样设计模式与子模式 DDL，描述数据的逻辑组织一般要包括如下内容：

1）描述数据模型各个部分（称为数据逻辑单位）的特征，通常是：①给出数据各个逻辑单位的无二义性命名，如数据模型名、记录类型名、数据项名等；②说明各个最小逻辑单位的数据特征，如数据类型是字母、数字还是字符，数据长度及取值范围等；③说明数据单位的自然含义，如数据项是表示姓名还是城市，表示年龄还是分数等。

2）描述各数据逻辑单位之间的联系，即数据表示的对象之间和对象内部的联系，通常是说明：①数据的一个逻辑单位按照什么规则包含哪些更小的逻辑单位，如一个记录类型依次包含哪些数据项；②哪个或哪些数据项组合作关键字使用；③数据项之间的完整约束条件；④各逻辑单位按什么规则形成一个整体的数据结构，如记录类型怎样构成树或网络。

（2）描述数据的物理特征。数据库设计称之为存储结构描述语言，描述的内容包括：①系统中建立了哪些物理文件，它用文件目录说明，文件目录又是系统的一个特殊文件；②

说明每个物理文件数据的文件组织方法，如顺序、索引、 Hash 、树结构、倒排结构等；③说明每个物理文件的文件名称、文件类型、文件的存储设备、开始地址、允许占用的空间等。

（3）描述存储映射即逻辑数据到物理数据的映射。它说明：①每个逻辑单位的数据存放在哪个文件中，如哪个记录类型对应于哪个文件，存放在哪个区域；②逻辑数据到物理数据的转换，如十进制变二进制，数据的存储紧缩过程等。

（4）描述访问规则。它包括：①用户与子模式的对应关系；②用户身份检验，如用户口令、声音、指纹的核对；③对用户的授权；④数据的加密锁与密码。

上述内容只是数据描述的一些基本内容，并不是详尽无遗的，各个数据库系统都根据自己的情况决定包含哪些内容。

2．数据操作语言 DML

数据操作语言是用户与数据库系统的接口之一，是用户操作数据库中数据的工具。在设计数据操作语言时，一般要做到：描述操作准确，无二义性；功能齐全，操作能力强，用户希望使用的操作应尽量满足；语言自然、直观，容易掌握，使用方便。

一般来说，数据操作是一些操作语句组成的集合。 DML 有两类，一类嵌入在宿主语言中使用，称为宿主型 DML，如嵌入在 COBOL、Fortran、C、Ada 等高级语言中；另一类可以独立交互使用，称为自主型或自含型 DML，如 SQL（SQL 也可以嵌在主语言中使用）、QBE等。当前各种数据库系统的数据语言虽然在形式上差别较大，但其功能基本相似，通常包括如下几方面的操作。

（1）从数据库中检索数据。

（2）向数据库中添加数据。

（3）删除数据库中某些已过时、没有保留价值的原有数据。

（4）修改某些属性发生变化的数据项的值，使之能确切反映变化后的情况。

（5）用于并发访问控制的操作。

四、数据字典

数据字典（Data dictionary）的主要任务是描述（或定义）数据库系统中各类对象的性质和属性（包括其自然语言含义），对象之间的交叉联系和它们的使用规则。数据字典描述时给每个对象一个惟一的标识（称为内码），以示区别。数据字典是复杂的，也要用数据模型描述，有时把它称为关于数据的“数据”或“元数据”，它也有源形式与目标形式的问题。它的目标形式是包括模式、子模式表、用户表、物理文件或区域表、内码与自然语言对照表等在内的一组字典表格。

一个数据库系统所涉及的对象大致可分为如下几类：

（1）与数据组织结构有关的对象，包括数据库（一个系统可能容纳多个数据库）、模式、子模式、记录类型（或类）、数据项、物理文件及索引等。

（2）与系统运行、配置有关的对象，包括存储过程、事务、终端、客户机等。

（3）与询问优化有关的对象，包括访问例程、代价估算所用各类信息。

（4）与完整、安全控制有关的对象，包括用户、角色、用户标识、访问授权、密钥及各类完整约束条件等。

需要全本请访问 www.ertongbook.com

（5）与系统监控有关的对象，包括触发子、审计项目及数据字典本身的变化情况等。

（6）数据字典的用处在于：满足 DBMS 快速查找有关对象的要求。

（7）供数据库管理员掌握整个系统运行的下列情况：①系统现有的数据库、用户；②当前具有的模式数目及其名称，每个模式包含的子模式与记录类型；③每个子模式包含的记录类型与用户；④某个记录类型（或数据项）所属的子模式和对应的处理文件或区域等。

五、用户访问数据的过程

现以用户通过应用程序读取一个记录为例，说明用户访问数据过程中的主要步骤：

（1）用户在应用程序中首先要给出所使用的子模式名称，然后在需要读取记录处嵌入一个用数据操作语言书写的读记录语句，当应用程序执行到该语句时，即转入 DBMS 的特定程序或向 DBMS 发出读记录的命令。

（2）DBMS 按照应用程序的子模式名，查找子模式表，确定对应的模式名称。可能还要检验操作的合法性，核对用户的访问权利，如果通不过，则拒绝执行该操作，并向应用程序状态字回送出错状态信息。

（3）DBMS 按模式名查阅模式表，找到对应的目标模式，从中确定该操作所涉及的记录类型，并通过模式到存储映射（往往也在模式中）找到这些记录类型的存储模式。这里还有可能进一步检查操作的有效性、保密性，如果通不过，则拒绝执行该操作并回送出错状态信息。

（4）DBMS 查阅存储模式，确定应从哪个物理文件、区域、设备、存储地址、调用哪个访问程序去读取所需记录。

（5）DBMS 的访问程序找到有关的物理数据块（或页面）地址，向操作系统发出读块（页）操作命令。

（6）操作系统收到该命令后，启动联机 I/O 程序，完成读块（页）操作，把要读取的数据块或页面送到内存的系统缓冲区。

（7）DBMS 收到操作系统 I/O 结束回答后，按模式、子模式定义，将读入系统缓冲区的内容映射为应用程序所需要的逻辑记录，送到应用程序的工作区。

（8）DBMS 向应用程序状态字回送反映操作执行结果的状态信息，如“执行成功”、“数据未找到”等。

（9）记载系统工作日志。

（10）应用程序检查状态字信息。如果执行成功，则可对程序工作区中的数据做正常处理；如果数据未找到或有其它错误，则决定程序下一步如何执行。

如图 1-9 所示，表示了上述操作步骤。

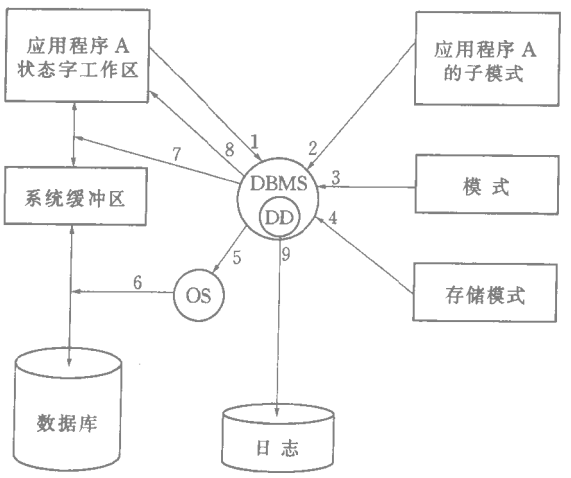


图 1-9　通过 DBMS 访问数据的步骤

习　　题

1. 试述信息三个领域之间的联系。

2. 解释下列术语的含义：实体、对象、属性、实体类型、数据、记录、项、型、值、实体模型、数据模型。

3. 试各举三例说明两个实体类型之间的三种联系。

4. 设有学生、班级、教师、教研室、课程等实体，每个实体可有多种属性，实体之间可有若干联系。试用 E—R 图进行描述。

5. 举例说明记录、项的型与值的区别。

6. 画出数据库的三级结构图。

7. 什么叫模式、子模式和存储模式？

8. 什么是数据独立性？数据库为什么要有独立性？

9. DBMS 的功能是什么？它包含哪些主要程序？

10. 什么是数据字典？它有哪些用途？

11. DBA 的职责主要有哪些？

第二章　关系模型的基本概念

第一节　从格式化模型到关系模型

从 20世纪 60年代到 90年代的短短几十年中数据库理论经历了从格式化模型（即层次模型和网状模型）到关系模型的过程。 1968年 IBM 研制出了层次数据模型系统 IMS。1969年，CODASYL 的下属组织 DBTG 提出了关于网状模型的 DBTG 系统报告。CODASYL 关于数据库的工作和报告澄清了许多概念，建立了若干权威性的观点，为数据库走向成熟奠定了基础。随着 CODASYL 工作的进展，商用网状数据库系统开始走向市场。

1970 年 6 月，IBM 的 E.F.Codd 提出了以关系的数学理论为基础的关系数据模型概念。由他开创的数据库的关系方法和规范化理论研究获得了 1981 年 ACM 图灵奖。从此，一些专用的或研究性的关系数据库系统陆续出现。如 IBM 的 System R 系统。1975年以后，开始出现商用的关系数据库系统，如 SQL/DS、INGRES、ORACLE、SYBASE 等。

关系方法给数据库技术带来了巨大的变革，并把它推向更高级的阶段。从 90 年代开始，关系模型数据库系统已成为商用主流数据库系统，广泛应用于各个领域。

第 二 节　　关 系 的 数 学 定 义

一、域

定义 2.1　域是一组具有相同数据类型的值的集合，又称值域。

【例 2-1】　整数、长度小于 20 的字符串的集合、[0，1]、实数等都可以是域。在关系中用域来表示属性的取值范围。域中所包含的值的个数称域的基数（用 m 表示）。

例如：D_1 = {张清远，刘逸}　　$m_1=3$

D_2 = {计算机专业，信息专业}　　$m_2=2$

D_3 = {李勇，刘晨，王晓}　　$m_3=3$

其中，D_1，D_2，D_3 是域名，“张清远，刘逸”，“计算机专业，信息专业”和“李勇，刘晨，王晓”分别是 D_1，D_2，D_3 各域的值。

二、笛卡尔积

定义 2.2　给定一组任意集合 D_1，D_2，…，D_n（它们可以包括相同的元素），这 n 个集合的笛卡尔积为 $D_1\times D_2\times\cdots\times D_n=\{(d_1,d_2,\cdots,d_n)\mid d_i\in D_i,\ i=1,2,\cdots,n\}$。

笛卡尔积也是一个集合。其中：D_i 称为域；每一个元素，$(d_1,d_2,\cdots,d_n)$ 叫做一个 n 元组（简称元组）；元素中每一个值 d 叫做一个分量，它来自相应的域 $(d_i\in D_i)$。

如果 D_i $(i=1,2,\cdots,n)$ 为有限集，D_i 中的集合元素个数称为 D_i 的基数，用 m_i $(i=1,2,\cdots,n)$ 表示。则笛卡尔积 $D_1\times D_2\times\cdots\times D_n$ 的基数为所有域的基数的累乘乘积 $m=$

$\prod_{i=1}^{n} m_i$。注意：元组不是 d_i的集合，元组的每个分量（d_i）是按序排列的，而集合中的元素是没有排列次序的。例如，元组（a，b，c）≠（b，a，c）≠（c，b，a），而集合（a，b，c）＝（b，a，c）＝（c，b，a）

笛卡尔积也可以用二维表的形式表示。下面我们举一例子。

【例 2-2】设有两个域，D_1＝｛0，1｝和 D_2＝｛a，b，c｝，$D_1 \times D_2$＝｛（0，a），（0，b），（0，c），（1，a），（1，b），（1，c）｝如图 2-1 所示（用二维表示）。

从图 2-1 中可以看出，笛卡尔积实际上是一个二维表，表的框架由域构成。表的任意一行就是一个元组。每一列数据来自同一个域，它的第一个分量来自 D_1，第 2 个分量来自 D_2。笛卡尔积就是所有这样的元组组成的集合。它的基数为 2×3＝6。

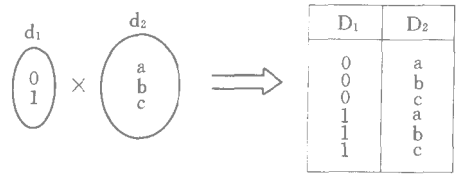


图 2-1 笛卡尔积的二维表示

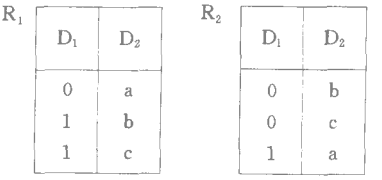


图 2-2 $D_1 \times D_2$ 上的两个子集

三、关系的数学定义

定义 2.3 笛卡尔积 $D_1 \times D_2 \times \cdots \times D_n$ 的任一子集称为定义在 D_1，D_2，…，D_n 域上的 n 元关系，用 R（D_1，D_2，…，D_n）表示。

其中 R 为关系名，n 称为关系的目或度。该子集元素是关系中的元组，通常用 r 表示。关系中的元组个数是关系的基数。当 n＝1 时，称为单元关系；当 n＝2 时，称为二元关系。当 n＝n 时，称为 n 元关系。

同样可把关系看作是一个二维表，表的框架由 D_i（i＝1，2，…，n）构成，每一行对应一个元组，表的每一列对应一个域。由于域可以是相同的，为了加以区别，每个列起一个名字，称为属性。n 目关系必有 n 个属性。属性的名字是惟一的。属性的取值范围 D_i（i＝1，2，…，n）称为值域。

例如，我们在图 2-1 中给出的笛卡尔积中任取两个子集，分别称为关系 R_1 和关系 R_2，如图 2-2 所示，R_1 是 D_1，D_2 上的一个二元关系，R_2 是 D_1，D_2 上的一个二元关系。R_1 和 R_2 有相同的关系框架，来自同一笛卡尔积。我们把具有相同关系框架的关系称为同类关系。因此，R_1 和 R_2 是同类关系。

注意：当 D_i 包含无穷多值时，笛卡尔积是一个无穷集合。其子集可以是无穷集合，也可以是有穷集合。从笛卡尔积中所取的有穷集合称为有限关系；所取的无穷集合称为无限关系。由于机器硬件的限制，在数据库系统中只考虑有限关系。以下例来说明怎样从笛卡尔乘积中构造出一个关系来。

【例 2-3】 设 D_1＝导师集合（SUPERVISOR）＝｛张清远，刘逸｝

D_2＝专业集合（SPECIALITY）＝｛计算机专业，信息专业｝

D_3＝研究生集合（POSTGRADUATE）＝｛李勇，刘晨，王晓｝

笛卡尔乘积 $D=D_1 \times D_2 \times D_3$＝｛（张清远，计算机专业，李勇），（张清远，计算机专

业，刘晨），（张清远，计算机专业，王晓），（张清远，信息专业，李勇），（张清远，信息专业，刘晨），（张清远，信息专业，王晓），（刘逸，计算机专业，李勇），（刘逸，计算机专业，刘晨），（刘逸，计算机专业，王晓），（刘逸，信息专业，李勇），（刘逸，信息专业，刘晨），（刘逸，信息专业，王晓）｝。D 共有 12 个集合元素，即基数 $m=\prod m_i=2\times2\times3=12$。这 12 个集合元素的总体可列成一个二维表。

从笛卡尔乘积中取子集来构造关系。我们知道由于一个研究生只师从与一个导师，学习某一个专业，所以笛卡尔积中的许多关系是无实际意义的。从中取出有意义的元组来构造关系 SAP，属性名取域名即 SUPERVISOR、SPECIALITY、POSTGRADUATE，则这个关系可以表示为 SAP（SUPERVISOR，SPECIALITY，POSTGRADUATE）。假设导师与专业是一对一的，导师与研究生是一对多的，这样 SAP 关系可以包含 3 个元组，见表 2-1.

表 2-1　　SAP 关 系

SUPERVISOR	SPECIALITY	POSTGRADUATE
张清远	计算机专业	李 勇
张清远	计算机专业	刘 晨
刘 逸	计算机专业	王 晓

定义 2.4 定义在域 D_1，D_2，…，D_n（不要求完全相异）上的关系由关系头和关系体组成。关系头由属性（名）A_1，A_2，…，A_n 的集合组成每个属性 A_i 正好对应一个域 D_i（$i=1$，2，3，…，n）；关系体由随时间变化的 n 个元组的集合构成，每个元组依次由一组属性值的集合构成。

关系头是关系的数据结构的描述，或称关系框架。关系头相对固定。关系体是指关系结构中的内容或者说数据（值），其随时间变化（随元组的建立、修改或删除而变化）。

第三节　关系的性质

关系是用集合代数的笛卡尔积定义的，是元组的集合。因此，关系具有如下特性：

（1）关系中没有重复元组，任意一个元组在关系中都是惟一的。因为关系体是元组的集合，而集合是没有重复元素的，所以作为集合元素的元组是不重复的。

（2）元组是非排序的，即元组的次序可以任意交换。因为集合的元素是非排序的，作为集合元素的元组当然是非排序的。因此改变元组次序之间的关系仍然是同一关系。

（3）属性是非排序的。因为关系头被定义为属性的集合，而集合元素不存在排序问题。所以属性是非排序的。属性是用它的名字来标识，而不使用它的位置来标识。

（4）属性必须有惟一的属性名，不同的属性可来自同一个域。属性名和相应域名可以同名，也可不同名，但对于出自相同域的不同属性必须起不同的名字，即一个关系的诸属性必须有不同的名称，以惟一标识元组的分量。

（5）同一属性名下的值是同类型数据，且必须来自同一个域。

（6）所有的属性值都是原子的，每个元组中的每个属性值都是不再可分的数据项（允许是空值）。如果用通俗的话说，就是不能“表中有表”。

满足上述条件的关系称为规范化关系（简称范式），否则称为非规范化关系。我们在关系方法中只用规范化的关系，因为一个规范化关系比一个非规范化关系有更简单的数据结

构，这将导致一系列其他问题的简化（特别是导致 DML 操作的简化）。

第四节　关　系　的　键

一、关系键的定义

在给定的关系中，存在一个或一组属性，它在不同的元组中的相应属性值或组合属性值是不同的，即具有惟一标识特性，这样的属性称为该关系的关系键。关系键也称为码或关键字。例如，学生关系中的属性“学号”，它的每个值能惟一地把每个学生元组区分开，所以属性“学号”是学生关系的关系键。在学生选课关系中组合属性“学号＋课程号”能够惟一地把学生选课关系中的每个元组区别开。所以，组合属性“学号”和“课程号”是它的关系键。下面，我们给出关系键的定义：

定义 2.5　设关系 R 有属性 A_1，A_2，…，A_n，其属性集 $K=(A_i, A_{\ }, \cdots, A_k)$，当且仅当满足下列条件时，K 被称为关系键：

（1）惟一性。在任一给定时间，关系 R 的任意两个不同元组，其属性集 K 的值是不同的；

（2）最小性。组成关系键的属性集（$A_{\ }, A_j, \cdots, A_k$）中，任意一个属性都不能从属性集 K 中删除掉，否则将破坏惟一性的性质。

二、候选键和主关系键

在某些关系中存在关系键特性的属性或属性组有多个。这些属性组都称为该关系的候选关系键，或称为候选关键字或候选码。当在一个关系中有多个候选关系键时，我们从候选关系键中选择一个作为主关系键（也称为主键、主关键字、主码）。主关系键在关系中用来作为插人、删除、检索元组的操作变量。

主关系键在数据库设计中是一个很重要的概念，每个关系都必须选择一个候选关系键作为主关系键。对于任一关系，主关系键一经选定，通常是不能随意改变的。每个关系中都必有且只有一个主关系键。就是说，标识元组惟一存在的主键必定是惟一存在的。

三、外部关系键

定义 2.6　如果关系 R_2 的一个或一组属性 X 不是 R_2 的关系键，而是另一关系 R_1 的关系键，则该属性或属性组 X 称为关系 R_2 的外部关系键（或称外关键字或外部码）。

在学习成绩关系中的“学号”是外部键，因为它不是学习成绩关系中的关系键，而是学生关系中的关系键（主键）。同样，“课程号”也是外部键。

四、关系模型的完整性

1. 实体完整性

主关系键的值不能为空或部分为空的约束条件称为实体完整性。元组是描述现实世界的实体个体。一个元组代表一个实体个体，关系体代表一个实体集。主关系键则实际上惟一标识了这些实体个体。

学生关系中各元组代表着许多不同的学生实体，而主关系键“学号”的属性值不仅标识代表学生实体的元组，而且还标识着学生实体本身。如果一个主关系键的属性值为空或

需要全本请访问 www.ertongbook.com