第一章　数据库导论

本章主要介绍数据库的基本概念，包括什么是数据库、数据库的系统结构、数据库管理系统、数据库模型、数据模式以及数据库技术的发展趋向，是全书各章的基础。

1.1　什么是数据库

数据库技术仅仅经历 30 余年，已经从简单的文件系统发展到无论在理论上还是在技术上十分成熟的计算机技术，成为全球范围内信息技术的核心，广泛应用于国民经济、国防军事等各个应用领域。

随着人类社会的进步，人们在现代化的生产和生活中不断提出各种新的需要，以及计算机硬、软件技术的飞速发展、新的功能不断增加，使能适应于人们各种需要的新兴数据库系统日新月异地不断研究与开发。

什么是数据库？简单地说，数据库（Database）是存放在计算机文件之中有组织的信息集合，其作用在于存储与产生用户所需要的信息，为他们提供快速简捷的信息服务。

比如，一个装满用户信息的电话本就是一个数据库。一个放置各类书籍的图书馆也可以看成是数据库，甚至我们的教科书也可以看成是数据库。当然，这些“纸张数据库”使用起来比计算机上的数据库麻烦得多。

图 1－1是一个最简化的数据库系统图。

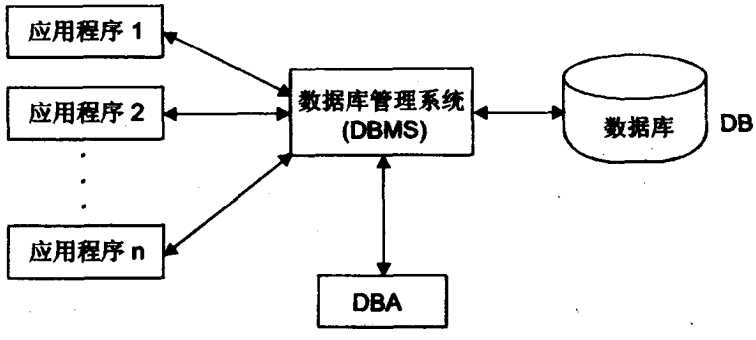


图 1－1　数据库系统图

1.2　数据库系统构成

一般来说，数据库系统是由数据库及其支撑的硬件、软件（包括数据库管理系统(DBMS)，各种应用开发工具等）以及人员（数据库管理员（DBA)，程序员，普通用户）组成。分别介绍如下。

需要全本请访问 www.ertongbook.com

1. 数据库及其支撑的硬件环境

数据库的硬件包括磁盘、磁带以及其它辅设控制器， I/O通道，内存等。通常数据库中的数据是存放在足够大的磁盘上，操作系统、 DBMS 的核心模块；数据库的数据缓冲区和应用程序放置在内存， DBMS 的各种应用开发工具放在磁盘区。

2. 软件

数据库系统的软件主要是 DBMS 及支持它的操作系统，以及开发应用程序所需的各种高级语言及其编译程序。这些高级语言与普通的高级语言的区别在于具有与数据库系统的接口，所以它们原来的编译程序需要修改和扩充。

目前，国内外流行的 DBMS往往均配置各种数据库应用开发工具。如菜单生成器、屏幕表格生成器、报表生成器、图形生成器等第四代语言（4GL），它们为用户开发各种数据库应用程序提供了良好的支撑环境。

3. 人员

数据库系统的用户主要有三类：

(1) 普通用户(End User)。

主要指计算机联机终端存取数据人员。他们使用数据库系统所提供的终端命令语言或菜单驱动等交互方式来存取数据库的数据，获取各种数据表格和图形。他们的操作往往比较简单、易学、易用，特别适合非计算机专业人员使用，不需要特别的专门培训。

(2) 应用程序员 (Application Programmer)。

他们是负责根据需要而设计和调试应用程序的人员，这类用户懂得使用高级语言(C、COBOL、PASCAL 等）和数据库的基本知识。这些应用程序包括通常的批处理应用程序或终端用户应用程序。

(3) 数据库管理员（Database Administrator，即 DBA)。

负责数据库系统的管理和维护的人员。对于大型的数据库系统， DBA 的工作是很重要的，通常由具有较高的技术水平和较好的管理经验人员担当，他们的主要职责和职能在 1.6 节再作介绍。

1.3 数据库的逻辑结构

1. 数据库的抽象结构层次

数据库系统无论建立在哪一种操作系统上，使用什么数据库语言，支持何种数据模型，在逻辑上总的分成三层，即外层（External level），概念层（Conceptual level）和内层（Internet level)。通常称之为数据库的三级结构（见图 1－2)。

内层是接近物理存储的层次，是数据在数据库中的实际存储，又称为存储视图。概念层是指数据库全体数据的逻辑结构和特征，又称为概念视图。外层是靠近用户的一层，又称为用户视图，其用户为联机终端用户或应用程序员。

2. 关于数据库模式

数据库系统需要对其抽象的逻辑层次进行描述，描述的结果称之为模式，故相应以上的三个抽象数据库逻辑层次分别为外模式（或用户模式）、模式 （概念模式）和内模式。

● 内模式，又称为存储模式，是对内层数据的物理结构和存储方式的描述。如记录

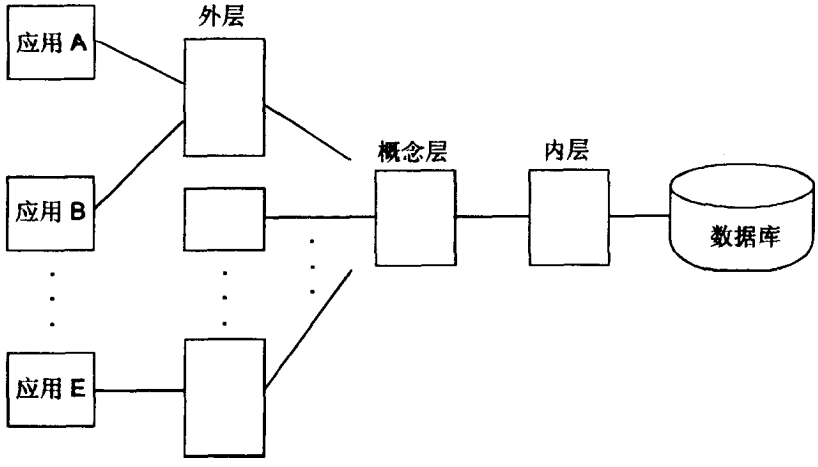


图 1－2　数据库的三个抽象层次

是顺序存储，还是按散列文件（HASH 文件）组织存储，是否需要索引等。内模式通常用内模式数据描述语言（内模式 DDL）来定义。

●模式，是数据库中全部数据的概念层的一个逻辑表示或描述。通常以某一种数据模型为基础，去定义数据的逻辑结构。如记录名，类型，数据项的各类型，长度等，以及记录间数据项之间的联系，还要定义与数据有关的安全保密和完整性约束条件等。

模式是由系统提供的模式描述语言（模式 DDL）来描述的。

●外模式，是个别用户的数据视图，它是外层用户所看到的和使用的数据库。通常个别用户仅需要使用整个数据库的某一部分子集，是用户与数据库的接口。每个用户需要通过一个外模式来使用数据库，但不同的用户可以使用同一个外模式。

外模式既然是各个用户的视图，而不同用户的需要不同，因而不同的用户可以对同一个模式定义出不同的外模式，使它们的外模式在结构、类型、长度、保密程度均与原来的模式不同。

数据库系统提供了外模式描述语言（外模式 DDL）来描述和定义外模式。

3. 数据库模式间的映像

文件系统的主要缺点之一就是用户程序过份地依赖存储数据和数据的全局逻辑结构。当数据存储方式和数据库整体逻辑结构发生改变时，用户不得不重新编写应用程序，从而保证原来的用户要求，这种程序对数据的依赖性的缺点在数据库系统中得到极大的改善。根本原因在于系统提供模式之间的两级映像，一级是在外模式与模式之间，另一级是模式与内模式之间（见图 1－3）。

（1）模式／内模式映像。

这一级映像主要确定概念视图与存储视图之间的对应关系，说明逻辑记录和数据项如何映像成它们所对应的存储结构。如果存储数据库结构发生改变时（如改变数据存储设备、存储位置和数据的存储组织方式等），将模式／内模式也作相应的改变（由 DBA 负责），使其模式保持不变，这使得数据的存储结构和存取方法较高地独立于应用程序。通过映像功能保证数据存储结构的改变不影响到数据库的整体（全局）逻辑结构，从而

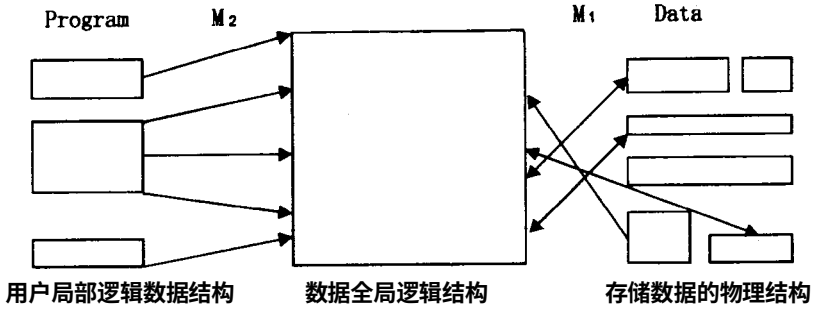


图 1－3　数据库模式之间的两级映像

不用修改程序。这是程序与数据的物理独立性。

(2)外模式/模式映像。

这一级的映像确定的是外部视图与概念视图的对应关系，体现在描述它们的外模式与模式之间的关系。外模式为模式子集，对应同一个模式的不同子模式的记录项的排序、类型均可不同。当模式改变时，只要相应地改变外模式／模式的对应关系，即可以保证外模式的不变，从而使建立在这一外模式上的应用程序保持不变，这就是程序与数据的逻辑独立性。这种通过调整映像使数据的整体逻辑结构的改变时，不改变局部逻辑结构，从而不改变相应的应用程序的特性为数据的逻辑独立性。

通常将数据的逻辑独立性与物理独立性统称为数据的独立性。

4. 小结

数据库系统的三级结构是很重要的概念，大多数的数据库系统的基本结构大体上符合以上结构，它是我们学习数据库原理和技术的基础。

●模式是用户（DBA）对数据库结构的一种描述，是建立数据库的结构框架，而根据这些结构框架装入的数据值为数据库的内容。在设计数据库时，设计者主要考虑模式的设计，而用户使用数据库时，关心的是数据库的内容。数据库的模式（结构框）是相对稳定的，而数据库的内容可千变万化。

● 数据库系统三级结构，将外模式、模式与内模式分开，并提供了它们之间的两级映像，保证了数据的物理独立性与数据的逻辑独立性，在数据库全局逻辑结构以及存储数据发生改变时，用户的应用程序可保持不变，从而大大减轻应用程序员和终端用户的负担。借助于外模式的设计和使用，不同的用户可以不同的观点去使用数据库资源，达到数据共享和减轻数据冗余的目的。另外由于外模式的设置，用户程序只能使用相应子模式的数据，从而有利于数据库的安全保密。

● 有了本节介绍的数据库系统的逻辑结构概念后，可以给数据库一个较为严格的定义。

数据库可以定义为一组存储在一起、具有尽可能小的冗余度的数据集，以最优的方式为多个应用服务，其数据的存储与使用数据的应用程序无关，使用一个公共控制方法来检索和更新现有的数据。

数据库的主要特点是数据的共享性（可为多个应用程序服务），数据独立性（用户程序与数据的逻辑组织与存储方式无关），数据的低冗余度和数据的统一控制。

1.4 数据库管理系统（DBMS）

数据库管理系统是数据库系统的核心，负责数据库的建立、使用和维护，建立在操作系统之上，对数据库实行统一管理、控制，是一个复杂的综合性的大型软件。

1.DBMS 的功能

一般说来，DBMS 的功能主要包括以下四个方面：

（1）数据库定义功能。

DBMS 提供数据定义语言（Data Definition Language，简称 DDL）来定义外模式，概念模式和内模式，各种源模式通过相应的模式翻译程序转换成相应的内部表示，分别为目标外模式、目标模式和目标内模式，它们是一系列的系统表格（作为数据字典的一部分）是 DBMS 存放和管理数据的基本依据，这些目标模式是数据库的描述，是存放数据的数据库框架，而不是数据库的数据本身。

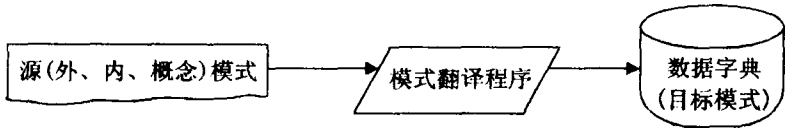


图 1-4

（2）数据存取功能。

DBMS提供数据操纵语言（Data Manipulation Language，简称 DML）实现对数据库数据的基本存取操作，如检索、插入、修改和删除。 DML 有两类：一类为交互式命令语言，其语法简单，可以单独使用，又称为自主型和自含型。另一类为嵌入在主语言中使用，如嵌入在 C，COBOL，PASCAL 语言中使用，这类 DML 语言本身不能单独使用，因此称为宿主型的。目前国内外大多数常用的 DBMS 均具有这两种 DML，供用户选用，如 SQL 语言。

对于交互型的命令语言，DBMS 通常采用解释执行方法，而对于宿主型的语言通常采用预编译程序，对源程序进行扫描时，要识别出所嵌入的 DML 语句，并把它们转换成主语言代码（或主语言调用语句），以便被原主语言编译程序所接受和执行。

（3）运行管理功能。

这是 DBMS 运行控制的核心部分，包括数据的完整性、安全性、并发控制、存储控制、所有数据库的操作都要在这些控制程序统一管理下运行。

（4）数据库的建立与维护功能。

DBMS提供一系列的实用程序来进行数据库的初始数据的装入、数据库的转储、恢复、备份、重组、系统性能监测、分析等。

DBMS 的功能随系统而异，大型的系统功能较强，小型系统功能较弱。如目前不少微机上的单用户 DBMS 的维护功能就较弱。

2.DBMS 对数据的存取过程

存取数据库的数据是 DBMS 最基本，最重要的操作。图 1-5 给出应用程序 P 通过

需要全本请访问 www.ertongbook.com

DBMS 读取一个数据记录的全过程为：

(1) 应用程序 P 向 DBMS 发出读记录的请求；

(2) DBMS 分析命令，进行语义检查，在数据库字典（含模式、子模式、内模式定义表格）中检查是否有记录（关系）和字段与之匹配，如果存取权限不符，则拒绝执行，返回错误信息；

(3) DBMS 执行查询优化，并将该命令转化成有关记录的存取操作命令；

(4) 检查内模式，决定以何方式读出相应的物理记录；

(5) DBMS 向操作系统发出读记录的命令；

(6) 操作系统将数据从数据库的存储区送至系统缓冲区；

(7) DBMS 根据数据字典，导出用户所需要的记录格式；

(8) DBMS 将数据记录从系统缓冲区送到程序 P 的用户缓冲区。

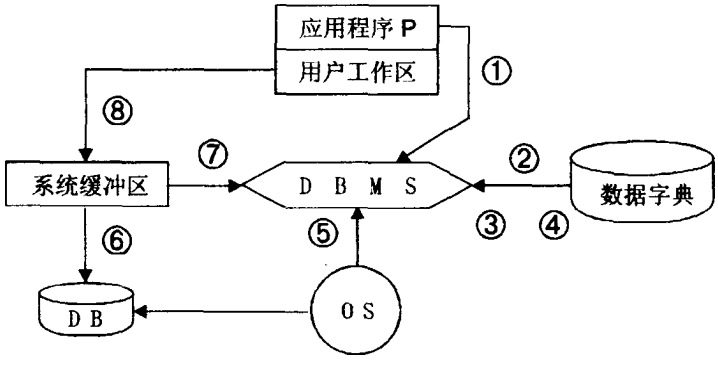


图 1－5　DBMS 访问数据的过程

1.5　数据模型

数据库是信息系统的核心，它不仅表示了数据的本身，而且反映了数据间的联系。一般用数据模型这一工具来抽象表示数据的形式化结构。它是各种计算机上实现 DBMS 的基础。通常从现实世界转换为机器世界，需要两级抽象（见如图 1－6）。

第一级抽象是现实世界中事物在人们头脑中的反映，现实世界的事物在人脑的信息世界中被抽象成实体（Entity），如人、汽车、货物等。也可以是概念性的东西，如考试、销售，甚至可以是事物间的联系，实体有型和值之分，如“学生”实体为型，具体内容为学生值，“张红”为学生实体的值。

实体中所具有的特性称为“属性”（Attribute），如“零件”实体具有名称、重量、产地、出厂日期等属性。属性也具有型与值之分。以上的属性也就是属性的型，而其具体内容如电池、50 克、广州、1997 年 10 月 5 日为相应属性的值。同类型的实体集合组成了“实体集”。

实体彼此区分靠的是标识实体的属性集，能唯一标识实体的属性名称为“码”（关键字），如学生实体中的学号，居民实体中的身份证号码等。

实体之间是相互关联的，实体集之间的关联主要分成三种，即一对一联系，一对多

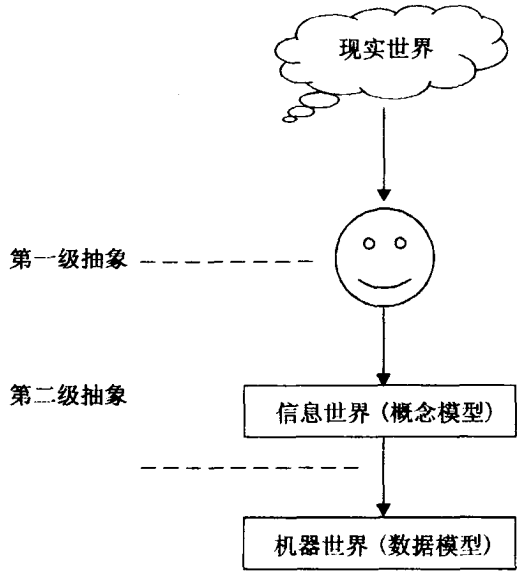


图 1－6　现实世界到机器世界的两级转换

联系，多对多联系以及实体内部的一对多和多对多联系，关于实体－联系的概念模型将在第六章作较详细介绍。

第二级抽象是将人脑中的信息世界中的“实体”以及“联系”，经过加工处理转换成计算机所能接受的“数据”形式，即信息的数据化。如将信息世界的实体数据转变为“记录”，将相应实体的属性的数据转为“数据项”，把相应于实体集的数据转为“文件”。在计算机世界中，原信息世界所抽象的概念模型再次被抽象为数据模型，它是实体（联系）概念模型的数据化，这样的抽象模型才能被计算机上具体的数据库管理系统所接受。

1. 数据模型的特性

数据库的数据模型具有严格的定义，这些概念描述了模型的静态与动态特性。包括三方面的内容：

(1) 数据结构。

数据结构是指数据模型所描述的对象类型间的相互关系集合，包括数据对象本身以及对象之间相关联系，在数据库系统中按数据结构特点分成层次、网状和关系的结构。通常数据结构由数据库子语言（定义模式的 DDL 和数据操纵的 DML）给予定义。

(2) 数据操作。

数据操作是对数据模型中各种对象（包括型与值）执行操作的集合。这些操作包括查询和更新（如修改、删除、插入等）。数据库系统必须给出这些操作的严格定义以及操作规则和实现操作的数据库子语言。

(3) 数据的完整性约束。

在数据库执行操作的过程中，数据模型中的数据及其联系处于不断变动之中。但无论变化有多么大，数据库中的数据，任何时候都要满足数据间的约束条件和一系列的完

整性规则，保证数据库数据的正确性与相容性。

如关系模型中任何关系任何时候都要满足实体的完整性与关联完整性的两个约束条件。

2. 数据模型的类型

当前，国内外流行的 DBMS所支持的主要数据模型有三类，即关系模型（Relational Model）、网状模型（Net Model）、层次模型（Hierarchical Model）。其中，关系模型使用“二维表”表示数据间的关系，而网状模型和层次模型是用图结构来表示数据间关系。

层次与网状模型统称为格式化模型，因为它们均可建立在“图论”表示的基础之上，是较早的数据模型。

这三种数据模型又称为传统数据模型，随着人们生产发展的不断深入和软件技术的不断进步，近年来，数据库理论与技术上又提出多种新的数据模型，如面向对象数据模型，函数式数据模型，语义数据模型等将在第 5 章介绍。

现将三种传统的数据模型简介如下：

(1) 层次模型。

从图论的角度，层次模型的数据模型是一棵树（tree），如图 1－7 所示，结点 A 为根结点，C、E、G 为叶结点，结点 B、F 为兄弟结点。

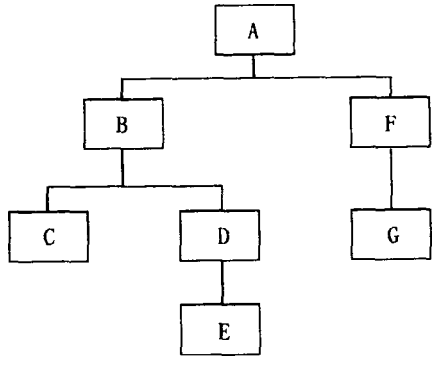


图 1－7　层次模型示意

层次模型的特点是：① 有且仅有一个结点没有双亲，这就是根结点；② 其它结点有且仅有一个双亲。

层次模型中，每一个结点描述一个实体型，对结点的查询，按照先序游历的策略查询，如对图 1－7 中各结点的查询序为 A－B－C－D－E－F－G 。

层次模型结构清晰，每个结点的双亲是唯一的，故只要知道每个结点的双亲结点（根结点除外），就知道整个模型的整体结构。

在现实世界中，具有层次关系的事物及其联系的关系很多，如企事业的行政关系，一个设备及其组成部件间的关系等等，用层次模型描述比较方便。

(2) 网状模型。

如果对层次模型的定义作适当的修改后，层次模型即可转变为网状模型，网状模型

的特征是：① 可以有一个以上的结点没有双亲；②至少有一个结点有多于一个双亲。

图 1－8 所示都是网状模型。

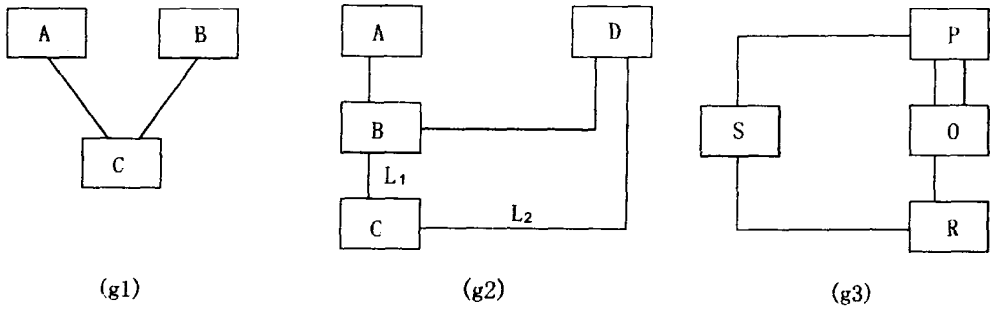


图 1－8　网状数据模型

层次模型与网状模型的主要区别在于：

● 在数据结构上，层次模型中的任一结点（除根结点）只有一个双亲，子女与双亲的联系是唯一的。而网状模型的结点可以有多个双亲，从子女到双亲的联系不唯一，必须靠不同的名字区分。图 1－8（g2）中，C结点的双亲结点为 B 和 D，其间的联系名分别为 L1 和 L2。

●网状模型允许两个结点之间可以有两个或两个以上的联系，这种联系可以用复合链实现（如图 1－8（g3）），而层次模型的结点间只能有一种联系。

●对层次模型中结点的查询总是从根结点开始，而网状模型的检索可以从网络中任一结点开始，经由系统指引，就能在网络中查到所需求的任一结点。

鉴于近年来，国内外对以上两种模型的数据库系统的研究与开发应用较少，故本书以下不再对有关内容作较详细介绍。

(3) 关系模型。

关系模型是三种数据模型中较为重要的模型，当今国内外大多数据库系统都是基于关系模型的。关系模型的数据结构是二维表框架，所以关系模型是“关系框架”所组成的集合。图 1－9 给出职工人事关系框架，如果在二维表框内填上适当的内容就构成如图的关系数据库。

职工人事关系框架

职工号	姓名	性别	出生年月	职务	籍贯	备注

职工人事关系数据库

职工号	姓名	性别	出生年月	职务	籍 贯	备注
G016	刘 明	男	1948.7	厂 长	江苏南京	
GS07	张 红	女	1952.6	高 工	广东佛山	
PH12	李 江	男	1974.8	副主任	广东湛江	
PH25	王 平	男	1970.7	工程师	湖南长沙	

图 1－9　关系模型与关系数据库

关系模型一般由若干个关系框架组成，因而关系模型的实例由若干个关系表格组成。表格中每一列的名称称为“属性”，如“姓名”、“性别”、“职务”等。相当于记录中的数据项名。属性的取值范围称为域（Domain）。表格中的一行称为“元组”（Tuple），相当于通常的记录值。在每一行中对应于某属性的值称为分量，如“王平”为第四个元组对应于“姓名”属性的分量。

关系模型的主要特点是：

（1）关系数据结构简单。在表示复杂信息世界的实体与联系中，无论是实体还是联系都可以用关系表格表示。而关系表格之间的联系是通过表格与表格所具有的相同属性，通过关系运算而实现的，用表格表示概念模型之复杂结构，要比层次和网络的结构简单得多。

（2）关系表格必须是规范化的关系，即关系模型中每一个属性都是不可再分的数据项。如工资项中不应再含有基本工资，扣款工资等项，使关系模型结构简单易操作。

（3）对关系模型的操作，是通过关系表格中的数据进行运算的结果，因而将关系模型中的数据存取路径隐藏起来。用户只需指出查找什么，而无需具体指明如何查找的路线，从而使关系模型的数据子语言具有较高独立性和非过程化，大大减轻用户的编程负担。

（4）关系模型是建立在严格的数学理论基础上，随着数据模型与数据库理论的不断发展，建立在关系模型之上的 DBMS 的研制得到很大的发展，关系数据库产品成为当今数据库市场的主流产品。

1.6 数据库管理员

对于大型的数据库系统，数据库管理员的岗位是极其重要的，可以看成为数据库系统的重要组成部分。下面简介其主要的功能职责。

1. 参加数据库设计

数据库的设计包括概念模式以及内外模式的设计。

（1）定义数据库的概念模式与外模式。

运用 DBMS 提供的模式定义语言 DDL 和外模式定义语言 SDDL 去描述整个数据的信息结构，以及用户所需要的信息结构内容。

（2）定义数据库的内模式。

利用系统提供的内模式定义语言，规定数据在数据库中的存储结构方式，文件的存储方法（顺序文件还是随机文件，文件是否需要索引等）。

2. 参加数据库的维护工作

数据库系统在建立以后，大量的工作在于数据库的安全性、完整性，并发控制以及恢复工作。

- 数据库在设计阶段时，DBA 要根据需要，定义数据的保密程度，规定用户口令以及用户对数据库数据的使用权限。
- 系统在各用户使用时，会产生并行性处理，DBA 则负责监督、检测和控制，以避免数据库系统所产生数据不一致性和死锁等问题。
- 当数据库系统发生意外或操作故障时，DBA 必须熟练地，迅速地进行数据库恢复。同时定期将数据转存到磁带上。根据需要，可以修改数据库模式和重组数据库，

证数据库每日的正常运转业务。

在使用数据库的各类用户中， DBA 具有最高的权限，同时也肩负着最大的责任。一个大型数据库系统的 DBA 由若干个人员分工，协作完成。

1.7 数据库技术发展的历史

数据库技术从 20 世纪 60 年代中至今，仅 30 余年的历史，其发展速度之快，应用范围之广是其他技术难以比拟的。数据库的发展经历了第一代的格式化模型（层次和网状）数据库系统，第二代的关系数据库系统，和发展到第三代的面向新技术和应用的数据库系统。即数据库技术与网络通讯、人工智能、面向对象、多媒体等技术的结合产生了新一代的数据库系统。

数据库技术广泛应用于经济信息管理、政府机关决策部门、工业控制、科学研究、文化教育、军事机关、社会服务等领域，成为信息管理技术的核心。现将数据库技术的发展历程简述如下：

1. 第一代数据库系统

这一时期的数据库系统主要代表是 1969 年 IBM 公司研制成功并已商业化的基于层次模型的数据库管理系统 IMS（Information management System），以及美国数据库系统语言协会 Conference on Data System Language）下属的数据库任务组 DBTG 对数据库方法进行系统的研究后，于 70 年代初提出的 DBTG 报告，确立了数据库系统的许多基本概念、方法和技术。DBTG 所提出的方法是基于网状结构的，是数据库网状模型的代表。

在 DBTG 的方法指引下开发了许多商业化的 DBMS，网状模型系统有 TOTAL，DMS1100，PDM ，基于 VAX－11 上的 DBMS等。

这两种数据库系统的共同特点是：

(1) 支持三级模式结构。

它为用户的应用程序与数据库中的数据提供独立性。

(2) 用存取路径来表示数据之间的联系。

在层次模型中用层次顺序来表示数据间的联系，而在 DBTG 中记录型之间用系（Set）来表示，系一般用指针的方法来实现。显然系值就是一种数据存取的路径。

(3) 独立的数据库语言。

无论是层次还是网状模型的数据库系统均具有独立于高级语言的数据库定义语言(DDL)，用以描述数据库的模式及其映像。

(4) 导航式的数据操作语言（DML)。

层次和网状数据库的数据操作语言是一次一个记录的导航式的过程化语言，它们通常需要嵌入某种高级语言（如 C，COBOL，PL/1 等）中使用。其特点是用户需使用某种高级语言编写程序，按照某条预先定义的存取路径来访问数据库。其优点是存取效率较高，但用户编程负担重，应用程序的移植性较差。

2. 第二代数据库系统（关系数据库）

1970 年，IBM 公司的 E.F.Codd 发表了题为“大型共享数据库数据的关系模型”论文，提出了关系数据库模型的概念，第一次提出了数据库关系方法与关系数据理论的

需要全本请访问 www.ertongbook.com

研究，为关系数据库技术的发展奠定了理论基础，为此他获得了 1981 年的 ACM 图灵奖。

70年代是关系数据库理论迅速发展与原型系统蓬勃开发的时代，主要代表有 IBM 研究室开发的 SYSTEM R 和 Berkeley 大学研制的 INGRES 。这一时期的主要成果在于建立了关系模型的理论，研制开发了关系数据语言，如关系代数、关系演算、 SQL 语言、QBE 等，并研制大量的 RDBMS 原型，解决了关系产品商业化的重大技术课题，如查询优化、并发控制、恢复等数据库保护技术。关系数据库终于从实验室走向社会的实际应用。

80 年代以后，新开发的数据库系统产品几乎都是关系型的。如 DB2, INGRES, ORACLE, Infoxmix, Sybase 等，而且产品均可运行在计算机网络上，以及适合于大、中、小型和微型计算机等环境中。

3. 新一代数据库系统

80年代以后，由于数据库技术在商业领域的巨大经济效益，推动了其他领域对数据库技术发展的需求。这些新的数据库领域包括计算机集成制造系统（CIMS）、计算机辅助设计（CAD）、计算机辅助软件工程（CASE）、办公自动化系统（OAS）、地理信息系统（GIS）、知识库系统（KBS）、图像数据库以及多媒体数据库系统等。

传统的数据库系统在很多方面已无法满足各个领域迅速发展的需求，如对复杂对象的存储管理、长事务与嵌套事务的处理、复杂数据类型的处理（如抽象数据类型、图形、图像、声音、时间等），又如数据库语言与程序语言的无缝集成，异构数据库，以及基于 Internet 网络语言与数据库系统的集成等。

新一代数据库系统的主要特征有：

(1) 建立面向对象的数据模型。

(2) 数据库技术与多学科技术的相互结合和渗透（如多媒体数据库、工程数据库、统计数据库、演绎数据库、时态数据库、专家数据库和模糊数据库等）。

(3) 基于网络环境下的多种同、异构数据库系统的信息集成。

(4) 巨型与超巨型 10^{12}字节）数据库拓展成数据仓库及其基于数据库仓库上的数据挖掘技术及商业智能。

至于新一代数据库系统的技术将在第十章再作进一步介绍。

习题 1

1. 解释下列术语：数据库、数据模型、数据库模式、外模式、概念模式、内模式、数据定义语言、数据操纵语言。

2. 试指出数据库、数据库管理系统和数据库系统三者之间的区别。

3. 试述数据库的物理独立性与逻辑独立性。

4. 试指出数据库系统的主要特点。

5. 试述数据库的三类用户及数据库管理员的主要职责。

6. 回答以下问题：

(1) 数据模式的主要特点是什么？

(2) 指出格式化模型（层次与网状模型）与关系模型的主要区别。

(3) 试指出数据库系统的逻辑结构。

(4) 试述数据库管理系统（DBMS）的主要功能。

(5) 数据库数据模型划分的依据是什么？

第二章　关系数据库基础

关系数据库是当前国内外最流行的数据库，也是本书学习的重点，以下将在第二章、第三章、第四章和第五章中分别作较详细介绍，本章主要介绍关系的数学定义、关系模型的数学基础、关系的运算和关系数据语言。

2.1　关系模型的基础

1. 关系的数学定义

关系数据的理论是建立在集合代数理论的基础上，以下我们用集合代数的方法给关系一个较为形式的定义。

定义 2.1　笛卡尔积

设 D_1，D_2，…，D_n 为 n 个任意集合，这 n 个集合的笛卡尔积（Cartesian Product）为：

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

其中 D_i 为域（Domain），域为值的集合，如字符串，整数，逻辑值，实数等。

每个元素（d_1，d_2，…，d_n）叫做一个 n 元组（n - tuple），简称元组，元组中每个 d_i 叫做元组的分量（component），它来自相应的域（$d_i\in D_i$）。

笛卡尔积是一个元组的集合，具有集合的一切性质，而笛卡尔积可以用二维表来表示，如：

例 2-1　给定域 $D_1=\{X, Y\}$，$D_2=\{0, 1, 2\}$，则笛卡尔积 $D_1\times D_2$ 可表示为：

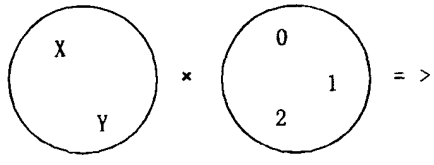


图 2-1　笛卡尔积例

$$D_1\times D_2=\{(X, 0), (X, 1), (X, 2), (Y, 0), (Y, 1), (Y, 2)\}$$

从图 2-1 可知笛卡尔积可表示一个二维表，表的框架由域 D_1，D_2 组成，表的任意一行为元组，它的第一分量来自 D_1，第二分量来自 D_2。

定义 2.2　关系的定义

笛卡尔积 $D_1\times D_2\times \cdots D_n$ 的任一子集称为在域 D_1，D_2，…，D_n 上的关系，表

示成 $R(D_1, D_2, \cdots, D_n)$，R 为关系名，n 为关系的目或度（Degree）。

笛卡尔积子集元素（$d_1, d_2, \cdots, d_n$）是关系中的元组，通常用 r 表示，关系中元组的个数为该关系的基数。

当 $n=1$ 时，称为单元关系（Unary relation）

当 $n=2$ 时，称为二元关系（Binary relation）

……

当 $n=n$ 时，称为 n 元关系。

例 2－2 集合 D_1，D_2 定义同例 2－1，则

$r_1=\{(X, 0), (Y, 1)\}$ 是 D_1，D_2 上的一个二元关系。

$r_2=\{(X, 1), (X, 2), (Y, 0), (Y, 2)\}$ 也是 D_1，D_2 上的二元关系。

r_1

D_1	D_1
X	0
Y	1

r_1

D_1	D_2
X	1
X	2
Y	0
Y	2

图 2－2 $D_1 \times D_2$ 上的关系

同样可将关系看成二维表，表框架为 D_1，D_2，每一行为一个元组，表中每一列对应一个域，每一列起一个名字，称为属性，故 n 目关系必有 n 个属性。属性的取值范围 D_i 为值域。

例 2－3 设一学生关系表如下：

表 2－1

学号	姓名	年龄	性别	系别
97002	林 平	18	男	计算机
97016	李 芳	19	女	计算机
97025	张金良	17	男	数 学
97054	罗宇国	20	男	数 学
97108	王小兰	18	女	电 子

这里，D_1＝学号＝ {97002，97016，…，97108}

D_2＝姓名＝｛林平，李芳，…，王小兰｝

D_3＝年龄＝ {17，18，19，20}

D_4＝性别 ＝ {男，女}

D_5＝所在系＝ {计算机，数学，电子}

显然由 $D_1 \sim D_5$ 五个域的取值，$D_1 \times D_2 \times \cdots D_5$ 共有 $5\times5\times4\times2\times3=600$ 个元素，即由 D_1，D_2，…，D_5 所构成的笛卡尔积集合共 600 个元素（元组），学生关系表仅取其五个元组，事实上，由于这五个属性，受到一定的语义约束，如一个学号仅唯一对应

一个学生姓名，每个学生的性别和系别是唯一。故诸如（97002，李芳，20，女，数学），(97025，张金良，20，男，电子）等这样的元组是违反语义规定的，是没有意义的。所以，一方面从集合代数观点，关系是笛卡尔积的任意子集，但在实际应用上，只有取某些子集才有实际意义。一般来讲，数据库中的关系应是从相应笛卡尔积中所选取的有意义的子集。

从以上关系的形式定义和例子中可以得到关系的特性：

(1) 关系的每一列（属性）都应有不同的名字，而不同列可以具有相同的域；

(2) 关系的列（属性）中属性值是同类型的数据，必须来自同一域；

(3) 关系列的顺序可任意交换；

(4) 关系中的元组不能全同；

(5) 关系中的元组次序可以任意交换；

(6) 关系中的分量（属性值）必须为不可再分的数据单位。

如表 2－2 中的关系，就不是规范关系，属性工资可继续分为基本工资和扣除工资。

表 2－2

职工号	姓名	性别	职务	工　资		电话号码
				基本工资	扣除工资	
A0102	李 平	男	厂 长	1500	200	8220102
A0116	张 三	男	副经理	1200	50	8220116
B0200	王 芳	女	工程师	950	100	8240217

在以下的关系数据库的讨论中，我们都假定关系是规范的，这样可以使得关系的数据结构更简单，使对关系数据模型的操作简化。

2. 关键字

(1) 候选关键字与主关键字。

定义 2.3

在给定的关系中，如果它的某一属性（或属性组）的值可以唯一地标识一个元组，则称该属性组为该关系的候选关键字（Candidate Key)。

一个关系可以具有多个候选关键字（候选码），这时可以根据需要选择其中一个作为主关键字（主码），主关键字中的属性为主属性。如学生档案中，可以将学号和身份证作为候选关键字，在学校部门中，往往取学号为主关键字，而公安部门可以取身份证号作为人事管理的主关键字。

关键字具有两个特点，即唯一性与最小性。即关系中，每当取定一个主关键字值可以唯一地决定一个元组的值。最小性是指关键字上的属性组合 A_1，A_2，… ，A_k 中，任一属性 A_i 都不能删去，否则会破坏唯一性的性质。

(2) 外来关键字。

在日常生活工作中，两个关系相互关联是大量存在着的，如图 2－3 (a)，学生与课程及选课关系，(b) 部门与职工关系。

在图 2－3 (a) 的学生学习成绩关系 SC 中，属性“学号”和“课程号”涉及到了对学生关系与课程关系的调用。显然，关系 SC 中学习成绩中的“学号”中的某个给定

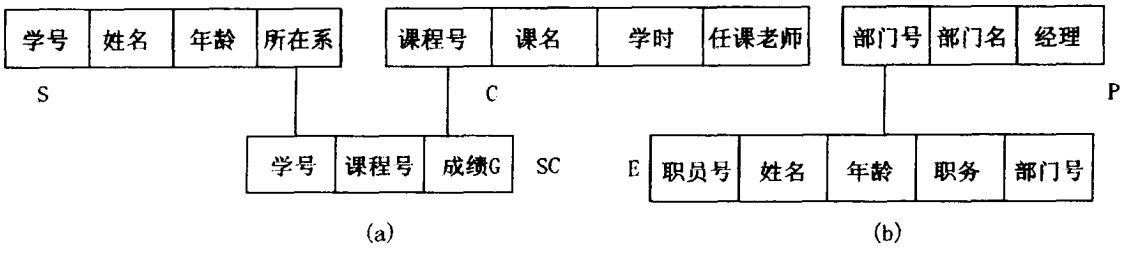


图 2-3　相互引用的关系

值，应能在学生关系的主关键字“学号”值中找到，否则，学生成绩关系 SC 中选课元组是一个不存在的学生的选课成绩，使得学生成绩关系变得无意义，同理对课程关系引用也是这样。我们将学习成绩关系 SC 中的“学号”与“课程号”称之为学习成绩关系的外来键（Foreign Key）。

定义 2.4　（外来关键字）

外来关键字是一关系模型的属性（或属性组合），而这个属性（或属性组合）是另一个关系模型的主关键字。

在图 2-3 (a) 的学生学习成绩关系 SC 中，“学号”是外来关键字，它的值不是该关系 SC 的主关键字值，而是学生关系的主关键字值。同理，“课程号”也是外来关键字。这里学生学习成绩 SC 的主关键字是“学号”加上“课程号”。所以“学号”与“课程号”均为主属性。一般而言，外来关键字不是所在关系的主关键字或主属性。如图 2-3 (b)，关系 E 中的部门号是关系 E 的外关键字，它不是关系 E 的主关键字，而是部门关系 P 的主关键字。

3. 关系数据库

(1) 关系模型的约束。

对关系模型的讨论主要分成三部分，即数据结构、关系的操作和关系完整性约束。关系的结构为简单的二维表，无论是实体或实体之间的联系均可用表格表示。而关系操作主要采用代数和逻辑的方式，即关系代数和关系演算。如关系代数主要有选择、投影、连接以及其它集合操作将在下一节介绍。本节主要介绍关系模型的完整性。

关系模型的完整性，包括实体的完整性、关联完整性与用户定义的完整性等三方面的约束。

1) 实体完整性（Entity Integrity）。

实体完整性是指实体的主关键字的值（主属性值）不能为空值或部分为空值。现实世界中实体是用主关键字来唯一地标识它们的，即不存在一个实体是不用关键字值去标识的，即主关键字值不能取空。但如果主码值可取部分码值，即违反主码的定义（主码值是唯一标识实体的最小属性值）。空值表示“不知道”或“无意义”。

2) 关联完整性（Referential integrity）。

关联完整性是指关系 R_2 的外来关键字与另一关系 R_1 的主关键字相符，即 R_2 的外码的每一值必须与 R_1 的某一元组的主码值相等，或者全为空。如图 2-3 (b) 中，E 的某职工的部门号（外来关键字）的值必须等于部门关系 P 中的某一部门号的值，

需要全本请访问 www.ertongbook.com