第 1 章　数据库原理

本章简要介绍数据库的发展，重点讲授数据库的基本概念，数据模型（主要针对关系模型）的组成要素、关系模型的数据结构、关系模型的基本术语、关系操作、关系模型的完整性约束，数据库系统的三级结构，数据库管理系统的基本功能、系统结构及运行过程。

1.1　数据库发展概述

人类社会正处于信息社会，人类知识以惊人的速度增长。如何组织和利用这样庞大的知识，成为急需解决的问题。在 20 世纪 60 年代的‘软件危机’中，数据库技术作为一门软件科学应运而生。

据查阅资料，数据库（Database，简称 DB）一词起源于 20 世纪 60 年代，当时美国为了战争需要，把各种情报集中在一起，存储在计算机中，将其称为 Information Base 或 Database。1963 年，美国 Honeywell 公司的 IDS（Integrated Data Store）系统投入运行，揭开了数据库技术的序幕。1965 年至 1968 年间，美国社会上人们可看到形形色色的 Database，但基本上都是文件系统的扩充或是倒排文件系统。1968 年，美国 IBM 公司推出了层次模型的 IMS 数据库系统；1969 年，美国 CODASYL（Conference On Data System Language，数据系统语言协会）组织的数据库任务组（DBTG）发表关于网状模型的 DBTG 报告（1971 年正式通过）；1970 年，IBM 公司的 E. F. Codd 发表论文提出关系模型，由此奠定了数据库技术发展的基础。

20 世纪 70 年代是数据库蓬勃发展的年代。网状系统和层次系统占据了整个商用市场，而关系数据库系统正处于实验阶段，许多学者倾注了大量的研究，使数据库技术在实践中和理论上得到了飞速发展，数据库技术也日趋成熟。

20 世纪 80 年代，关系数据库系统以绝对的优势取代网状系统和层次系统占据了市场。同时，关系数据库的理论日趋完善，走向更高级的阶段。关系数据库从集中式处理走向分布式处理。

现在，数据库技术已成为计算机领域中最重要的技术之一，它是软件科学中的一个独立分支。数据库技术使得计算机的应用渗透到各行各业乃至社会的每一角落，正在改变着社会的未来。尤其是 21 世纪，全球信息化浪潮滚滚而来，各个国家乃至每个地区都把推动信息化进程、加快信息产业的发展、开发和利用信息资源作为战略规划的重点。数据库是实现信息化的核心和基础。

1.2　数据库基本概念

1.2.1　数据、数据库

1. 数据（Data）

数据是数据库中存储的基本对象。数据按通常的理解表现为数字形式，这是对数据的一种传统和狭义的理解。广义的理解：数字只是数据的一种表现形式，在计算机中可表示数据的种类很多，文字、图形、图像、声音都可以数字化，所以这些都是数据。为了了解世界、交流信息，人们在计算机中需要并且也能够描述、存储和处理这些表现形式多样和内容复杂的数据。

需要全本请访问 www.ertongbook.com

可以对数据做如下定义 描述事物的符号记录称为‘ 数据 ”。因此,根据上面的解释,描写事物的符号可以是数字，也可以是文字、图形、图像和声音等，即有多种表现形式，但它们都是经过数字化后存入计算机的。

2. 数据库(Data Base,缩写为 DB)

数据库可以直观地理解为存放数据的仓库，只不过这个仓库是在计算机的大容量存储器上，如硬盘就是一类最常见的计算机的大容量存储设备。数据必须按一定的格式存放，因为它不仅需要存放，而且还要便于查找。所以可以认为，数据是被长期存放在计算机内的、有组织的、可以表现为多种形式的、可共享的数据集合。数据库技术使数据能按一定格式组织、描述和存储，且具有较小的冗余度、较高的数据独立性和易扩展性，并可为多个用户所共享。

1.2.2 数据库管理系统

了解了数据和数据库的概念，就应该研究如何利用计算机有效地组织和存储数据、获取和管理数据，完成这个任务的是数据库管理系统(Data Base Management System ,缩写为 DBMS),它属于一类计算机系统软件。

数据库管理系统是位于用户与操作系统之上的一层数据管理软件，它的主要功能包括以下几个方面：

1. 数据定义功能

DBMS 提供数据定义语言 (Data Definition Language ,缩写为 DDL)，用户通过它可以方便地对数据库中的相关内容进行定义，如对数据库、基本表、视图和索引进行定义。

2. 数据操纵功能

DBMS 向用户提供数据操纵语言 (Data Manipulation Language ,缩写为 DML)实现对数据库的基本操作,如对数据库中数据的查询、插入、删除和修改。

3. 数据库的运行管理

这是 DBMS 的核心部分，它包括并发控制即处理多个用户同时使用某些数据时可能产生的问题、安全性检查、完整性约束条件的检查和执行、数据库的内部维护 (如索引、数据字典的自动维护)等。所有数据库的操作都要在这些控制程序的统一管理下进行，以保证数据的安全性、完整性以及多个用户对数据库的并发使用。

4. 数据库的建立和维护功能

包括：数据库初始数据的输入、转换功能，数据库的转储、恢复功能，数据库的重新组织功能和性能监视、分析功能等，这些功能通常是由一些实用程序完成的，它是数据库管理系统的一个重要组成部分。

5. 数据字典

数据字典(Data Dictionary ,缩写为 DD) 是存放数据库各级模式结构的描述，也是访问数据库的接口。在大型系统中,DD 也可单独成为一个系统。

6. 数据通信功能

包括与 OS 的联机处理、分时处理和远程作业传输的相应接口等，这一功能对分布式数据库系统尤为重要。

1.2.3 数据库系统

数据库系统(Data Base System ,缩写为 DBS)通常是指带有数据库的计算机应用系统，因此，数据库系统不仅包括数据库本身，即实际存储在计算机中的数据，还包括相应的硬件、软件和各类人员。数据库系统是一个由使用和维护人员、加工设备和数据资源等构成的完整计算机应用系统。图 1.1 为 数据库系统组成示意，现分别介绍其相关内容。

1. 硬件

由于一般数据库系统数据量很大，加之 DBMS 丰富、强有力的功能使得自身的体积就很大，因此整个数据库系统对硬件资源提出了较高的要求：

（1）有足够大的内存以存放操作系统、DBMS 的核心模块、数据缓冲区和应用程序。

（2）有足够大的直接存储设备存放数据（如磁盘），有足够的磁带或其他存储设备来进行数据备份。

（3）要求计算机有较高的数据传输能力，以提高数据传送率。

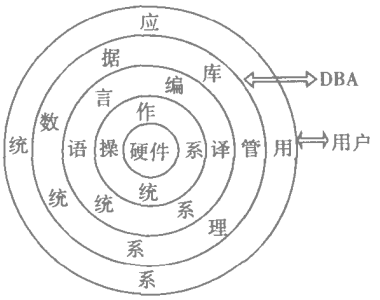


图 1.1 数据库系统组成示意图

2. 软件

数据库系统的软件主要包括：

（1）DBMS 是数据库系统的核心软件，它实现数据库的建立、使用和维护。

（2）支持 DBMS 运行的操作系统。通常，DBMS 运行时都是基于某一操作系统，并通过操作系统来实现对数据的存取。

（3）一般来讲，数据库管理系统的数据处理能力较弱，所以需要提供与数据库接口的高级语言及其编译系统，以便于开发应用程序。这种高级语言称为“数据库的主语言”。

（4）以 DBMS 为核心的应用开发工具。应用开发工具是系统为应用开发人员和最终用户提供的高效率、多功能的应用生成器、第四代语言等各种软件工具，如报表生成系统、表格软件、图形系统等，它们为数据库系统的开发和应用提供了有力的支持。当前，开发工具已经成为数据库软件的有机组成部分。

（5）为特定应用开发的数据库应用软件。数据库软件为数据的定义、存储、查询和修改提供支持，而数据库应用软件是对数据库中的数据进行处理和加工的软件，它面向特定应用，如基于数据库的各种管理软件、管理信息系统（MIS）、决策支持系统（DSS）和办公自动化（OA）等都属于数据库应用软件。

3. 数据

数据是数据库的基本组成内容，是对客观世界所存在事物的一种表征，也是数据库用户操作的对象。数据是数据库系统也是企业或组织的真正财富。数据应按照需求进行采集，并有结构地存入数据库。由于数据类型的多样性，数据的采集方式和存储方式也会不同。数据作为一种资源是数据库系统最稳定的成分，即硬件可能更新，甚至软件也可以更换，但只要企业或组织的性质不改变，数据将是可以长期使用的财富。

4. 人员

参与分析、设计、管理、维护和使用数据库中数据的人员都是数据库系统的组成成分。他们都在数据库系统的开发、维护和应用中起重要作用。分析、设计、管理和使用数据库系统的人员主要是：数据库管理员、系统分析员、应用程序员和最终用户。下面介绍他们各自的职责，并着重介绍数据库管理员的任务和职责。

（1）数据库管理员（Data Base Administrator，缩写为 DBA）：数据库是整个企业或组织的数据资源，因此企业或组织设立了专门的数据资源管理机构来管理数据库，DBA 则是这个机构的一组人员，负责全面地管理和控制数据库系统。具体的职责包括：

①决定数据库中的数据内容和结构：数据库中要存放哪些数据，是由系统需求来决定的。为了更好地对数据库系统进行有效的管理和维护，DBA 应该参加或了解数据库设计的全过程，并与用户、应用程序员、系统分析员密切合作、共同协商，搞好数据库设计。

②决定数据库的存储结构和存取策略：DBA 综合各用户的应用要求，和数据库设计人员共同决

定数据库的存储结构和存取策略，以求获得较高的存取效率和存储空间利用率。

③定义数据的安全性要求和完整性约束条件： DBA 的重要职责是保证数据库的安全性和完整性，即数据不被非法用户所获取，且保证数据库中数据的正确性和数据间的相容性。因此， DBA 负责确定各个用户对数据库的存取权限、数据的保密级别和完整性约束条件。

④监控数据库的使用和运行：DBA 还有一个重要职责就是监视数据库系统的运行情况，及时处理运行过程中出现的问题。当系统发生某些故障时，数据库中的数据会因此遭到不同程度的破坏，DBA 必须在最短时间内将数据库恢复到某种一致状态，并尽可能不影响或少影响计算机系统其他部分的正常运行。为此，DBA 要定义和实施适当的后援和恢复策略，如采用周期性的转储数据和维护日志文件等方法。

⑤数据库的改进和重组 :DBA 还负责在系统运行期间监视系统的存储空间利用率、处理效率等性能指标，对运行情况进行记录、统计分析，依靠工作实践并根据实际应用环境，不断改进数据库设计。不少数据库产品都提供了对数据库运行情况进行监视和分析的实用程序，DBA 可以方便地使用这些实用程序来完成这些工作。

另外，在数据库运行过程中，大量数据不断插入、删除、修改，随着运行时间的延长，在一定程度上会影响系统的性能，因此，DBA 要定期对数据库进行重新组织，以提高系统的性能。

当用户的需求增加和改变时，DBA 还要对数据库进行较大的改造，包括修改部分设计、实现对数据库中数据的重新组织和加工。

（2）系统分析员：是数据库系统建设初期主要的参与人员，负责应用系统的需求分析和规范说明。他们要和用户相结合，确定系统的基本功能、数据库结构和应用程序的设计以及硬软件配置，并组织整个系统的开发。所以，系统分析员是一类具有领域业务知识和计算机知识的专家，他们在很大程度上影响数据库系统的质量和成败。

（3）应用程序员：根据系统的功能需求负责设计和编写应用系统的程序模块，并参与对程序模块的测试。

（4）用户：这里用户是指最终用户（End User）。数据库系统的用户是有不同层次的，不同层次的用户其需求的信息以及获得信息的方式也是不同的。一般可将用户分为操作层、管理层和决策层，他们通过应用系统的用户接口使用数据库。常用的接口方式有菜单驱动、表格操作、图形显示、随机查询和对数据库中的数据进行统计，分析时使用专用的软件和分析、决策模型。

1.2.4 数据库系统特征

与手工操作和文件系统相比，数据库系统的优点是明显的：

1. 与手工操作相比较，其查询迅速、准确，而且可以省去大量的纸面文件

以一个大型仓库管理为例：用手工操作，如要查找“某地区生产的商品的名称、规格、数量”，就必须要翻阅账本，费时、费力。使用数据库系统，数据由 DBMS 按一定结构组织、存放在计算机中，可以迅速地查找到所要求的数据，而且不会出现错误。

2. 数据结构化且统一管理

在数据库中，数据是有结构的，并且由 DBMS 统一管理。DBMS 既管理数据的物理结构，也管理数据的逻辑结构；既考虑数据本身，也考虑数据之间以及文件之间的联系，即 DBMS 管理的是结构化的数据。

3. 数据冗余度小

在文件系统中，一个应用程序面对自己专用的一个或几个数据文件，会有许多数据相重复。数据库系统是从整体上即从全局看待和描述数据的，数据不仅面向某个应用而且面向整体应用，从而大大减少数据冗余，节约了存储空间，避免了数据之间的不一致性。所谓数据的不一致性是指同一数据不同拷贝的值不一样，这是由于文件系统中的数据重复存储，不同的应用程序使用和修改不同的拷贝所

造成的。

4. 具有较高的数据独立性

数据独立性是指用户应用程序与存储在磁盘上数据库中数据的相互独立性。也就是说，数据在磁盘上数据库中的存储是由 DBMS 管理的，用户程序一般不需要了解，应用程序要处理的只是数据的逻辑结构。这样，当数据在计算机存储设备上的物理存储改变时，应用程序可以不改变，而由 DBMS 来处理这种改变,这称之为“物理独立性”。有的 DBMS 还提供了一些功能使得某种程度上数据库的逻辑结构改变了,用户程序也可以不变,这称之为“逻辑独立性”。所以说,数据独立性是数据库的一种特征和优点，它有利于在数据库结构修改时能保持应用程序尽可能地不改变或少改变，这样就大大地减少了应用人员的开发工作量。

5. 数据的共享性好

在数据库应用中，数据是共享的，这不仅使某些应用程序的编写更加方便，而且冗余度小，系统易维护,易扩充。

在数据库中数据冗余度小，减少了由于数据冗余造成的同一数据重复存储而导致修改时的困难和可能造成数据的不一致。

由于数据采用数据库统一管理，且有结构，在使用数据时就有很灵活的方式，可以适应各种用户的要求，而且数据易于维护和扩展。数据库中数据共享的优点和作用使多种应用、多种语言、多种用户可以相互覆盖地使用数据集合。

6. 数据控制功能

为了适应共享数据的环境， DBMS 还提供了数据控制功能，控制功能包括对数据库中数据的安全性、完整性、并发和恢复的控制。

(1) 数据的安全性：数据的安全性是指保护数据，以防止不合法的使用造成的数据的泄密和破坏，使每个用户只能按规定对某些数据以某种方式进行使用和处理。

(2) 数据的完整性：数据的完整性是对数据的正确性、有效性和相容性的要求，即控制数据在一定的范围内有效，或要求数据之间满足一定的关系。

(3) 并发控制：当多个用户的并发进程同时存取、修改数据库时，可能会发生相互干扰而得到错误结果并使得数据库的完整性遭到破坏，因此必须对多用户的并发操作加以控制和协调。

(4) 数据库恢复：计算机产生的硬件故障、操作员的失误以及人为的破坏都会影响数据库中数据的正确性，甚至造成数据库部分或全部数据的丢失。 DBMS 必须具有将数据库从错误状态恢复到某一已知的正确状态，亦称为完整状态或一致状态的功能，这就是数据库的恢复。

综上所述，数据库是长期存储在计算机内有组织的大量的共享数据集合。它可以供各种用户共享，具有最小冗余度和较高的数据独立性。DBMS 在数据库建立、运用和维护时对数据库进行统一控制，以保证数据的完整性、安全性，并在多用户对数据并发使用时进行并发控制和发生故障后进行系统恢复。

1.3 数据模型

定义 模型是对现实世界的抽象。在数据库技术中,用模型的概念描述数据库的结构与语义,对现实世界进行抽象。把表示实体类型及实体间联系的模型称为数据模型，数据模型是现实世界数据特征的抽象。

1.3.1 数据模型的组成要素

如果抽象出数据模型共性，并加以归纳，数据模型可严格定义成一组概念的集合，这些概念精确描述了系统的静态特性、动态特性和完整性约束条件。因此，数据模型通常由数据结构、数据操作和完

需要全本请访问 www.ertongbook.com

整性约束三部分组成。

1. 数据结构

数据结构是所研究的对象类型的集合，这些对象是数据库的组成成分，一般可以分为两类：一类是与数据类型、内容、性质有关的对象，如网状模型中的数据项、记录，对应于现实应用环境中的数据项、记录；关系模型中的关系，对应于现实世界中的实体等；另一类是与数据之间联系有关的对象。

在数据库系统中通常按照数据结构的类型来命名数据模型，如将层次结构、网状结构和关系结构的模型分别命名为层次模型、网状模型和关系模型。数据结构是对系统静态特性的描述。

2. 数据操作

数据操作是指对数据模型中各种对象值允许执行的操作的集合。如关系模型中的关系的值所允许执行的所有操作即操作的集合，包括操作及有关的操作规则。数据库中主要有检索和更新（包括插入、删除、修改）两类操作。数据模型要定义这些操作的确切含义、操作符号、操作规则（如操作优先级别）以及实现操作的语言。

数据操作是对系统动态特性的描述。

3. 数据的完整性约束条件

数据的完整性约束条件是完整性规则的集合。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，这些规则用来限定基于数据模型的数据库状态及状态的变化，以保证数据库中数据的正确、有效和相容。

1.3.2 概念模型

概念模型实际上是现实世界到机器世界的一个中间层次。其目的是用于信息世界的建模，是现实世界到信息世界的第一层抽象，是数据库设计人员进行数据库设计的有力工具，也是数据库设计人员和用户之间进行交流的语言。因此，概念模型一方面应该具有较强的语义表达能力，能够方便、直接地表达应用中的各种语义知识，另一方面它还应该简单、清晰、易于被用户理解。

1.信息世界中的基本概念

信息世界涉及的概念主要有：

(1)实体(Entity).

客观存在并可相互区别的事物称为实体。实体可以是具体的人、事、物，也可以是抽象的概念或联系，例如，一个职工、一个学生、一个部门、一门课、学生的一次选课、部门的一次订货等都是实体。

(2)属性(Attribute)。

实体所具有的某一特性称为属性。一个实体可以由若干个属性来刻画，例如学生实体可以由学号、姓名、性别、出生年月、系号、高考成绩等属性组成(9801011 ,张莹,女,1979.01,01,550) ,这些属性组合起来表征了一个学生。

(3) 码(Key).

惟一标识实体的属性集称为码。例如，学号是学生实体的码。

(4)域 (Domain).

属性的取值范围称为该属性的域。例如，学号的域为 7位，姓名的域为字符串集合。

(5)实体型 (Entity Type)。

具有相同属性的实体必然具有共同的特征和性质。用实体名及其属性名集合来抽象和刻画同类实体，称为实体型。例如(学号，姓名，性别，出生年月，系号，高考成绩)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，全体学生就是一个实体集。

(7)联系 Relationship).

在现实世界中，事物内部以及事物之间是有联系的，这些联系在信息世界中反映为实体（型）内部

的联系和实体（型）之间的联系。实体内部的联系通常是指组成实体的各属性之间的联系，实体之间的联系通常是指不同实体集之间的联系。

两个实体型之间的联系可以分为三类：

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（也可以没有）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，记为 1∶1。

例如，学校里面，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。

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≥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 与实体集 B 具有一对多联系,记为 1∶n。

例如，一个系中有若干名学生，而每个学生只在一个系学习，则系与学生之间具有一对多联系。

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≥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≥0)与之联系,则称实体集 A 与实体集 B 具有多对多联系,记为 m∶n。

例如，一门课程同时有若干个学生选修，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。

实际上，一对一联系是一对多联系的特例，而一对多联系又是多对多联系的特例。

可以用图形来表示两个实体型之间的这三类联系，如图 1.2 所示。

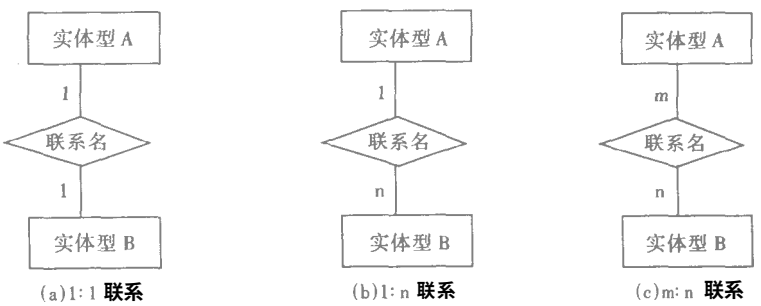


图 1.2　两个实体型之间的三类联系

例如，学生、教师、课程三个实体型，一门课可以有多个教师讲，一门课可以有多个学生选，则课程和教师之间存在 1∶N 的联系，课程和学生之间也存在 1∶N 的联系。

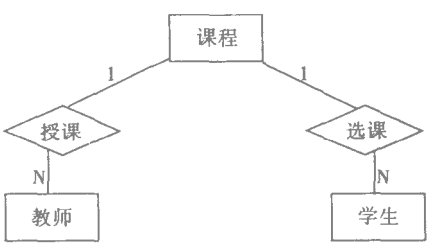


图 1.3　三个实体型之间的联系示例

2. 概念模型的表示方法

概念模型是对信息世界建模，所以概念模型应该能够方便、准确地表示出上述信息世界中的常用概念。概念模型的表示方法很多，其中最为著名、最为常用的是 P. P. S. Chen 于 1976 年提出的实体—联系方法(Entity—Relationship Approach)。该方法用 E—R 图来描述现实世界的概念模型，E—R 方法也称为 E—R 模型。

E—R 图提供了表示实体型、属性和联系的方法：

- 实体型：用矩形表示，矩形框内写明实体名。
- 属性：用椭圆形表示，并用无向边将其与相应的实体连接起来。

例如，学生实体具有学号、姓名、性别、出生年月、系号、高考成绩等属性，用 E—R 图描述如图 1.4 所示。

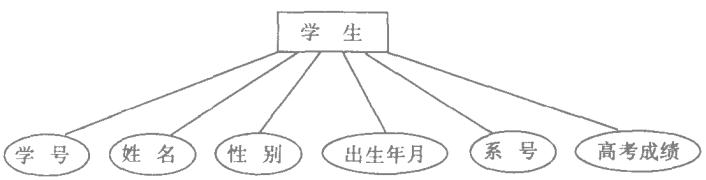


图 1.4 学生实体及属性

联系：用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体连接起来，同时在无向边旁标上联系的类型(1∶1，1∶n 或 m∶n)。

需要注意的是，如果一个联系具有属性，则这些属性也要用无向边与该联系连接起来，如图 1.5 所示。

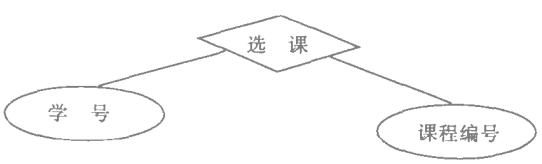


图 1.5 联系的属性

1.3.3 关系模型

1. 关系模型的数据结构

关系模型的数据结构单一，是一种二维表格结构，如图 1.6 所示。

关系名：学生(Student)

2. 关系模型的基本术语（见图 1.6 中的解释）

(1)关系模型：用二维表结构来表示实体及实体间联系的模型称为关系模型（Relational Model）。

(2)属性和值域 在二维表中的列(字段、数据项)称为属性(Attribute) 列值称为属性值 属性值的取值范围称为值域(Domain)。

(3)关系模式：在二维表中，行定义（记录的型）称为关系模式。

(4)元组(Tuple)与关系 在二维表中的行(记录的值)称为元组 元组的集合称为关系 关系模式和关系常常通称为关系。

(5)关键字(Key)或码 在关系的诸属性中，能够用来惟一标识元组的属性(或属性组合)称为关键字或码，即关系中的元组由关键字的值来惟一确定。

关键字是一个语义概念。一个关系的关键字是什么是没有办法从数学上证明的。同时，关键字的

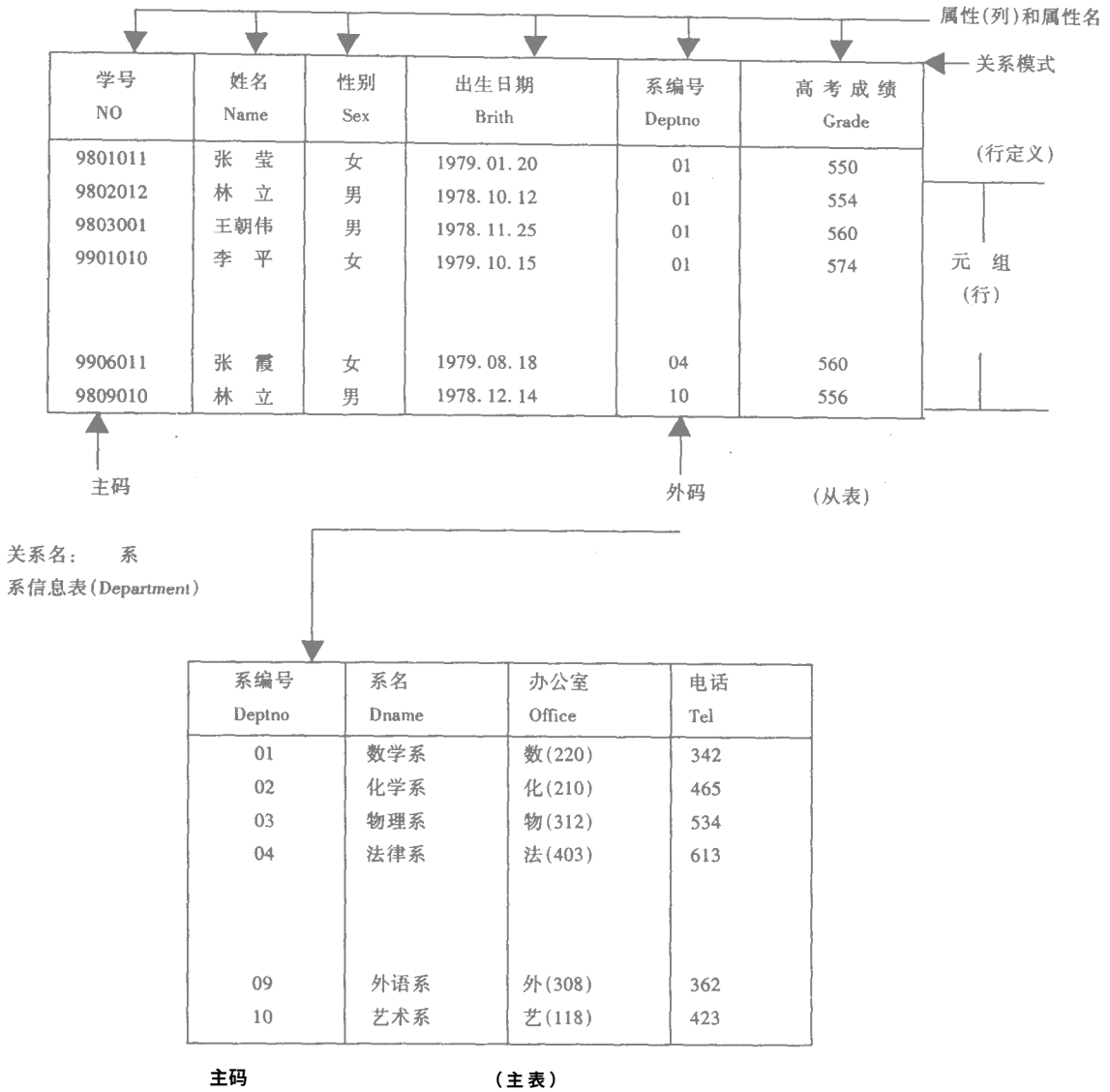


图 1.6 关系模型的数据结构

惟一性不是只对关系的当前元组构成来确定的。对图 1.6 学生关系和系关系，关键字分别是学号和系号。学生关系中一般不能用姓名作关键字，因为在一个学校中不可避免地有同名学生。

在一个关系中，关键字的值不能为空，即关键字的值为空的元组在关系中是不允许存在的。但在一些桌面型的系统中,如 xBase 系列却没有这种严格的限制。

有些关系的关键字是由单个属性组成的，还有一些关系的关键字常常是由若干个属性组合而构成的，即这种关系中的元组不能由任何一个属性惟一标识，必须由多个属性的组合才能惟一标识。

(6)候选关键字(Candidate Key) 或候选码：如果在一个关系中，存在多个属性或属性组合都能用来惟一标识该关系的元组，这些属性或属性组合都称为该关系的候选关键字或候选码。

(7)主关键字 (Primary Key) 或主码：在一个关系的若干个候选关键字中指定作为关键字的属性或属性组合，称为该关系的主关键字或主码。

(8) 非主属性(Non Primary Attribute) 或非码属性：关系中不组成码的属性均为非主属性或非码属性。

(9)外部关键字(Foreign Key)或外键：当关系中的某个属性或属性组合虽不是该关系的关键字或只是关键字的一部分，但却是另一个关系的关键字时，称该属性或属性组合为这个关系的外部关键字或外键。如学生关系中的系号。

(10)主表与从表：主表和从表是指以外键相关联的两个表，以外键作为主键的表称为主表，外键所在的表称为从表。如学生关系是从表，系信息表是主表。

3. 关系操作

关系数据库所使用关系语言的特点是高度非过程化，即用户只需要说明“做什么”而不必说明“怎样做”。用户不必请求数据库管理员为其建立特殊的存取路径，存取路径的选择是由 DBMS 自动完成的。这也是关系数据库的主要优点之一。

早期，关系操作有两种表示方式：关系代数与关系演算。理论上，关系代数和关系演算被证明是完全等价的。

关系代数通过对关系的运算来表达查询，其操作对象是关系，操作结果亦为关系。关系代数的运算可包括传统的集合操作和专门的关系操作两类。

以下通过关系代数来说明关系操作是如何实现的：

(1)传统的集合操作：传统的集合操作包括并、交、差、广义笛卡尔积等。这类操作将关系看成元组的集合。其操作是从关系的水平方向，即是对关系的行来进行的。

设关系 R 和关系 S 具有相同数目的属性列(n 列属性),并且相应的属性取自同一个域,则可定义以下四种集合运算：

①并(Union)关系 R 与关系 S 的并,它是属于 R 或属于 S 的元组组成的集合,结果为 n 列属性的关系。

②交(Intersection)关系 R 与关系 S 的交,它是既属于 R 又属于 S 的元组组成的集合，结果为 n 列属性的关系。

③差(Difference)关系 R 与关系 S 的差,它是属于 R 而不属于 S 的元组组成的集合，结果为 n 列属性的关系。

④广义笛卡尔积(Extended Cartesian Product)关系 R(假设为 n 列)和关系 S(假设为 m 列)的广义笛卡尔积是一个(n+m)列元组的集合,每一个元组的前 n 列是来自关系 R 的一个元组，后 m 列是来自关系 S 的一个元组。若 R 有 k1 个元组,S 有 k2 个元组,则关系 R 和关系 S 的广义笛卡尔积有 k1·k2 个元组。

图 1.7 中(a)、(b)分别为具有三个属性列的关系 R、S,(c)为关系 R 与 S 的并,(d)为关系 R 与 S 的交,(e)为关系 R 和 S 的差,(f)为关系 R 和 S 的广义笛卡尔积。

(2)专门的关系操作:包括选择、投影、连接等。这类操作不仅涉及行,而且也涉及列。

①选择(Selection)：选择操作是指在关系中选择满足某些条件的元组。例如：要对学生信息表中找出所有女生的数据，则可对学生信息表做选择操作，条件是 Sex='女'。

②投影(Projection)：投影操作是在关系中选择某些属性列。例如：要对学生信息表中找出所有学生的高考成绩，则可对学生信息表做投影操作，将表数据投影到学号、姓名和高考成绩列上。

③连接(Join)：连接操作是从两个关系的笛卡尔积中，选取属性间满足一定条件的元组。

R

A	B	C
a1	b1	c1
a1	b2	c2
a2	b2	c1

(a)

S

A	B	C
a1	b2	c2
a1	b3	c2
a2	b2	c1

(b)

R 与 S 的并

A	B	C
a1	b1	c1
a1	b2	c2
a2	b2	c1
a1	b3	c2

(c)

R 与 S 的交

A	B	C
a1	b2	c2
a2	b2	c1

(d)

R 与 S 的差

A	B	C
a1	b1	c1

(e)

R 和 S 的广义笛卡尔积

A	B	C	A	B	C
a1	b1	c1	a1	b2	c2
a1	b1	c1	a1	b3	c2
a1	b1	c1	a2	b2	c1
a1	b2	c2	a1	b2	c2
a1	b2	c2	a1	b3	c2
a1	b2	c2	a2	b2	c1
a2	b2	c1	a1	b2	c2
a2	b2	c1	a1	b3	c2
a2	b2	c1	a2	b2	c1

(f)

R 和 S 的等值连接

A	B	C	A	B	C
a1	b2	c2	a1	b2	c2
a1	b2	c2	a2	b2	c1
a2	b2	c1	a1	b2	c2
a2	b2	c1	a2	b2	c1

(g)

R 和 S 的自然连接

A	B	C	A	C
a1	b2	c2	a1	c2
a1	b2	c2	a2	c1
a2	b2	c1	a1	c2
a2	b2	c1	a2	c1

(h)

图 1.7 集合运算示例

例如,对图 1.7 中关系 R 和 S 做连接操作，连接条件是 S.B=R.B ,即在图 1.7(f)所示的两个关系的笛卡尔积中选取 S.B=R.B 的元组，得到的结果关系如图 1.7(g)。连接条件中的属性称为连接属性，两个关系中的连接属性应该是可比的，即是同一类的数据类型，如：都是数字型或都是字符型。连接条件中的运算符为算术比较运算符，当此算符取“ = ”时,为等值连接。图 1.7(g)则是关系 R 和 S 做等值连接后得到的结果关系。若等值连接中连接属性为相同属性（或属性组），且在结果关系中去掉重复属性（或属性组），则此等值连接为自然连接。如图 1.7(h)即是关系 R 和 S 做自然连接得到的结果关系,其连接条件为 S.B=R.B。自然连接是最常用的连接操作。

由此可见，关系操作的特点是集合操作，即操作的对象是元组的集合—关系，而操作的结果也是元组的集合—关系。

4. 关系模型的完整性约束

关系模型中的完整性是指数据库中数据的正确性和一致性。数据完整性是由数据完整性规则来维护，包括实体完整性和参照完整性规则。当然，任何给定数据库中还要包括与应用有关的完整性规则，这类完整性规则并不是关系模型特有的，将其称为用户定义的完整性规则。

(1) 实体完整性规则(Entity Integrity Rule).

实体完整性是指：关系的主属性即主码的组成不能为空，也就是关系的主属性不能是空值(Null)，在关系系统中一个关系通常对应一个表。在机器上实际存储数据的表称为基本表，除此之外的查询结果表是临时表，视图表是虚表，是不实际存储数据的表，而实体完整性是针对基本表的。因此，具体地讲，实体完整性是指在实际存储数据的基本表中主属性不能取空值。

定义实体完整性的必要性是：关系对应于现实世界中的实体，而现实世界中实体是可区分的。也就是说每个实体具有惟一性标识，在关系模型中，是由主码做惟一标识的。若主码取空值，则说明这个实体不可标识，即不可区分，这显然是错误的，与现实世界应用环境矛盾，因此不存在这样的无标识实

需要全本请访问 www.ertongbook.com

体，从而引入实体完整性的概念。

(2) 参照完整性规则(Reference Integrity Rule)。

参照完整性的定义：如果基本关系 R2 的属性中，包含与另一个基本关系 R1 的主码 K 相对应的属性组 F,那么 F 称为外部码。对 R2 每个元组在 F 上的取值只允许两种可能：一是空值；二是等于 R1 中某个元组的主码值。这里 R1 与 R2 不一定是不同的关系；K 与 F 是定义在同一域中。在图 1.6 中学生关系的‘系号’属性与系关系的主码‘系号’相对应,因此‘系号’属性是学生关系的外码,系关系是被参照关系，学生关系为参照关系。对于学生关系中每个元组的“系号”属性只能取下面两类值：

①空值：表示尚未给该学生分配系。

②非空值：该值必须是系关系中某个元组的“系号”值，表示该学生不可能分配到一个不存在的系中。即被参照关系‘系号’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。

实体完整性与参照完整性是由系统自动支持的,即在建立关系(表)时只要说明了“谁是主码”、“谁参照于谁”,系统将自动进行此类完整性的检查。

(3)用户定义的完整性规则。

用户定义的完整性规则是针对某一应用环境的完整性约束条件，它反应了某一具体应用所涉及的数据应满足的要求。系统提供定义和检验这类完整性规则的机制，其目的是用统一的方式由系统来处理它们，不再由应用程序来完成这项工作。

在实际系统中，这类完整性规则一般在建立库表的同时进行定义，应用编程人员不需再作考虑。如果某些约束条件没有建立在库表一级，则应用编程人员应在各模块的具体编程中通过程序进行检查和控制。

1.4 数据库系统结构

数据库系统的结构是数据库系统的一个总框架。尽管实际数据库软件产品种类繁多，使用的数据库语言各异，基础操作系统不同，采用的数据结构模型相差甚大，但是绝大多数数据库系统在总体结构上都体现三级模式的结构特征。

1.4.1 数据库系统的三级模式结构

1. 数据库系统模式的概念

在数据模型中有‘型’(Type)和‘值’(Value)的概念。型是指对某一类数据的结构和属性的说明，值是型的一个具体赋值。

模式是数据库中全体数据的逻辑结构和特征的描述，它仅仅涉及到型的描述，不涉及到具体的值。模式的一个具体值称为模式的一个实例（Instance)。同一个模式可以有很多实例。模式是相对稳定的，而实例是相对变动的，因为数据库中的数据是在不断更新的。模式反映的是数据的结构及其联系，而实例反映的是数据库某一时刻的状态。

虽然实际的数据库管理系统产品种类很多，它们支持不同的数据模型，使用不同的数据库语言，建立在不同的操作系统之上，数据的存储结构也各不相同，但它们在体系结构上通常都具有相同的特征，即采用三级模式结构（早期微机上的小型数据库系统除外）并提供两级映象功能。

2. 数据库系统的三级模式结构

数据库系统的三级模式结构是指数据库系统由外模式、模式和内模式三级构成。对应这三种模式，我们把数据库分为用户级、概念级和物理级，它们之间的相互关系如图 1.8 所示。

(1)外模式(External Schema).

用户级数据库对应外模式，外模式也称子模式(Subschema)或用户模式，它是数据库用户（包括应用程序员和最终用户）能够看到和使用的局部数据的逻辑结构和特征的描述，是数据库用户的数据

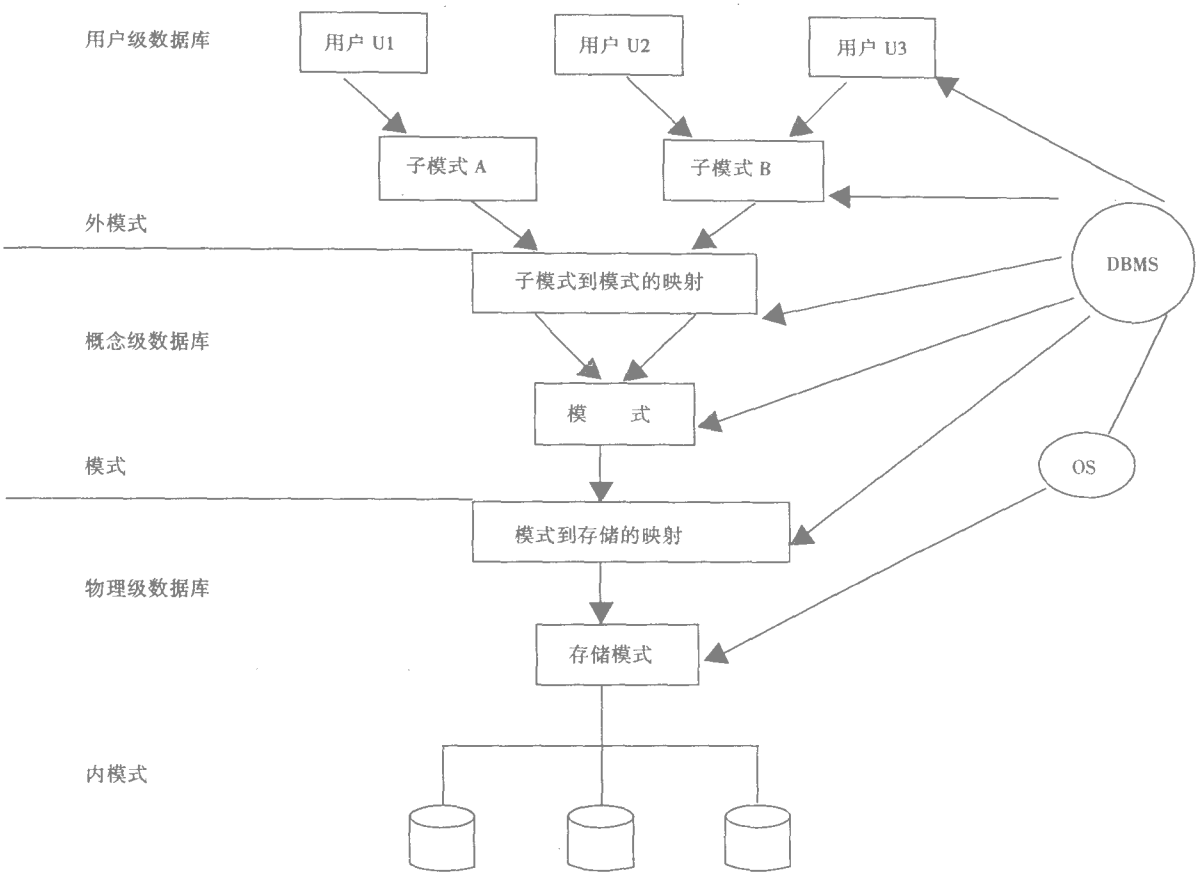


图 1.8　数据库系统的三级模式结构

视图，是与某一应用有关的数据的逻辑表示。用户根据系统给他的视图，用询问语言或应用程序去操作数据库中的数据。

外模式通常是模式的子集，一个数据库可以有多个外模式。外模式描述了用户使用的视图，是用户与数据库的接口，因此每个用户都必须使用一个外模式，但多个用户也可以使用同一个模式。如果不同的用户在应用需求、看待数据的方式、对数据保密的要求等方面存在差异，则其外模式描述就是不同的。即使对模式中同一数据，在外模式中的结构、类型、长度、保密级别等都可以不同。另一方面，同一外模式也可以为某一用户的多个应用系统所使用，但一个应用程序只能使用一个外模式。

外模式是保证数据安全性的一个有利措施，是数据的局部逻辑结构，也是数据库用户看到的数据视图。每个用户只能看见和访问所对应的外模式中的数据，数据库中的其余数据是不可见的。

（2）**模式**(Schema)。

概念级数据库对应概念模式，简称模式，也称逻辑模式，是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。它是数据库系统模式结构的中间层，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和与所使用的应用开发工具及高级程序设计语言无关。模式的主体就是数据库的数据模型。

模式实际上是数据库数据在逻辑级上的视图。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，统一综合地考虑了所有用户的需求，并将这些需求有机地结合成一个逻辑整体。定义模式时不仅要定义数据的逻辑结构，例如数据记录由哪些数据项构成以及数据项的名字、类型、取值范围等，而且要定义数据之间的联系，定义与数据有关的安全性、完整性要求。

（3）**内模式**（Internal Schema）。

物理级数据库对应内模式。物理级数据库包括数据库的全部存储数据，是用户操作的对象。

内模式也称存储模式(Storage Schema)，一个数据库只有一个内模式。它是数据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。例如，记录的存储方式是顺序存储、按照 B 树结构存储还是按 hash 方法存储；索引按照什么方式组织；数据是否压缩存储，是否加密；数据的存储记录结构有何规定等。

对于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。概念级数据库是物理级数据库的一种抽象描述，用户级数据库是用户与数据库的接口。用户根据外模式进行操作，通过外模式到模式的映射就把用户级和概念级联系起来，通过模式到存储的映射又把概念级与物理级联系起来。

图 1.9 是一个图书馆系统的三级关系，由此我们可以对数据库系统的三级模式结构得到进一步的比较形象的理解。

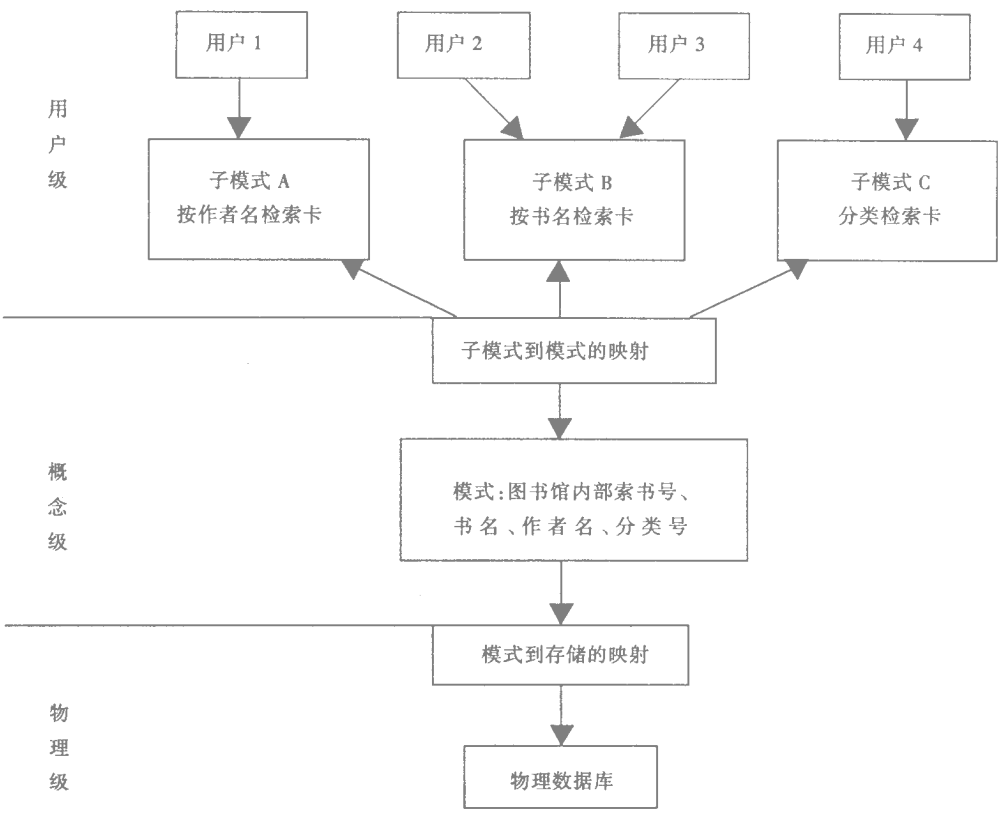


图 1.9 图书馆系统的三级关系

1.4.2 数据库的二级映象功能与数据独立性

数据库系统的三级模式是对数据的三个抽象级别，它把数据的具体组织留给 DBMS 管理,使用户能逻辑地、抽象地处理数据，而不必关心数据在计算机中的具体表示方式与存储方式。为了能够在内部实现这三个抽象层次的联系和转换，数据库管理系统在这三级模式之间提供了两层映象：

- 外模式／模式映象
- 模式／内模式映象

正是这两层映象保证了数据库系统中的数据能够具有较高的逻辑独立性和物理独立性。

1. 外模式／模式映象

模式描述的是数据的全局逻辑结构，外模式描述的是数据的局部逻辑结构。对应于同一个模式可以有任意多个外模式。对于每一个外模式，数据库系统都有一个外模式／模式映象，它定义了该外模式与模式之间的对应关系。这些映象定义通常包含在各自外模式的描述中。

当模式改变时（例如增加新的关系、新的属性、改变属性的数据类型等），由数据库管理员对各个外模式／模式的映象作相应改变，可以使外模式保持不变。应用程序是依据数据的外模式编写的，从而应用程序不必修改，保证了数据与程序的逻辑独立性，简称数据的逻辑独立性。

2. 模式／内模式映象

数据库中只有一个模式，也只有一个内模式，所以模式／内模式映象是惟一的，它定义了数据库全局逻辑结构与存储结构之间的对应关系，例如说明逻辑记录和字段在内部是如何表示的。该映象定义通常包含在模式描述中。当数据库的存储结构改变了（例如选用了另一种存储结构），由数据库管理员对模式／内模式映象作相应改变，可以使模式保持不变，从而应用程序也不必改变。保证了数据与程序的物理独立性，简称数据的物理独立性。

在数据库的三级模式结构中，数据库模式即全局逻辑结构是数据库的中心与关键，它独立于数据库的其他层次。因此，设计数据库模式结构时，应首先确定数据库的逻辑模式。

数据库的内模式依赖于它的全局逻辑结构，但独立于数据库的用户视图即外模式，也独立于具体的存储设备。它是将全局逻辑结构中所定义的数据结构及其联系按照一定的物理存储策略进行组织，以达到较好的时间与空间效率。

数据库的外模式面向具体的应用程序，它定义在逻辑模式之上，但独立于存储模式和存储设备。当应用需求发生较大变化，相应外模式不能满足其视图要求时，该外模式就得做相应改动，所以设计外模式时应充分考虑到应用的扩充性。

特定的应用程序是在外模式描述的数据结构上编制的，它依赖于特定的外模式，与数据库的模式和存储结构独立。不同的应用程序有时可以共用同一个外模式。数据库的二级映象保证了数据库外模式的稳定性，因此从底层保证了应用程序的稳定性，除非应用需求本身发生变化，否则应用程序一般不需要修改。

数据与程序之间的独立性，使得数据的定义和描述可以从应用程序中分离出去。另外，由于数据的存取由 DBMS 管理，用户不必考虑存取路径等细节，从而简化了应用程序的编制，大大减少了应用程序的维护和修改。

1.4.3 数据库系统的操作过程

为了体现数据库三级模式的作用，现以一个应用程序从数据库中读取一个数据记录为例，说明用户访问数据时数据库管理系统的操作过程，同时也具体反映了数据库各部分的作用以及它们之间的相互关系。图 1.10 表示了用户访问数据库的主要步骤。

图中各标号的说明：

1——应用程序 A 向 DBMS 发出读取数据的请求，同时给出记录名及要读取记录的关键字值；

2——DBMS 接到请求之后,利用应用程序 A 所用的外模式来分析这一请求；

3——DBMS 调用模式，进一步分析请求，根据外模式与模式之间变换的定义，决定应读入哪些模式记录；

4——DBMS 通过内模式，将数据的逻辑记录转换为实际的物理记录；

5——DBMS 向操作系统发出读所需物理记录的请求；

6——操作系统对实际的物理存储设备启动读操作；

7——读出的记录从保存数据的物理设备送到系统缓冲区；

8——DBMS 根据模式和外模式的规定，将记录转换为应用程序所需的形式；

9——DBMS 把数据从系统缓冲区传送到应用程序 A 的程序工作区；

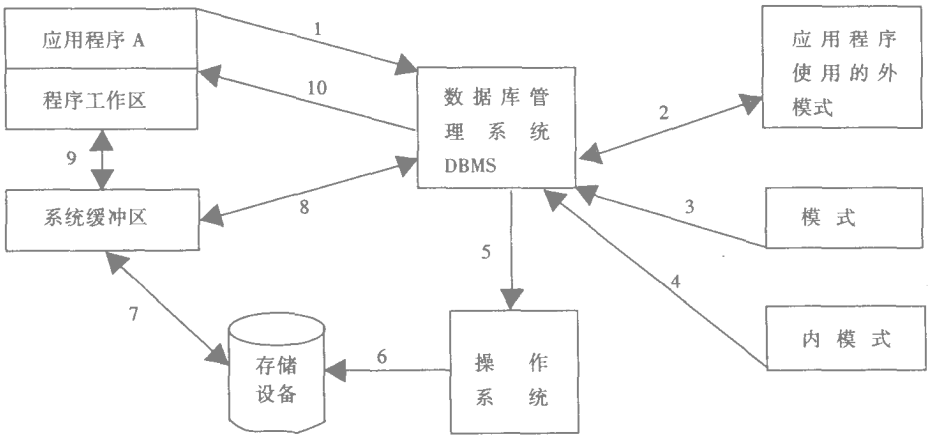


图 1.10 数据库的操作过程

10——DBMS 向应用程序 A 发出请求执行情况的信息。

以上给出应用程序 A 读取数据库中数据的一般步骤和过程，并体现了三级模式的作用。不同的数据库管理系统其操作细节可能存在差异，但其基本过程会大致一致。至于其他的数据操作，如写入数据、修改数据、删除数据等，其步骤会有增加或变化，但总体上是十分相近的。

1.5 数据库管理系统(DBMS)

1.5.1 DBMS 的基本功能

DBMS 主要是实现对共享数据有效的组织、管理和存取。围绕数据 DBMS 应具有如下几个方面的基本功能：

1. 数据库定义

对数据库的结构进行描述，包括外模式、模式、内模式的定义；数据库完整性的定义；安全保密定义，如用户口令、级别、存取权限等 存取路径，如索引的定义。这些定义存储在数据字典(亦称为系统目录)中，是 DBMS 运行的基本依据。

2. 数据存取

提供用户对数据的操纵功能，实现对数据库数据的检索、插入、修改和删除。一个好的 DBMS 应该提供功能强并易学易用的数据操纵语言（DML）、方便的操作方式和较高的数据存取效率。DML 有两类：一类是宿主型语言，一类是自含型语言。前者的语句不能独立使用而必须嵌入某种主语言，如在 C 语言、COBOL 语言中使用，而后者可以独立使用，通常供终端用户交互使用。

3. 数据库运行管理

这是指 DBMS 运行控制、管理功能。包括多用户环境下的并发控制、安全性检查和存取权限控制、完整性检查和执行、运行日志的组织管理、事务的管理和自动恢复即保证事务的原子性，这些功能保证了数据库系统的正常运行。

4. 数据组织、存储和管理

DBMS 要分类组织、存储和管理各种数据，包括数据字典、用户数据、存取路径等等，要确定以何种文件结构和存取方式在存储器上组织这些数据，以及如何实现数据之间的联系。数据组织和存储的基本目标是提高存储空间利用率，选择合适的存储方法提高存取(如随机查找、顺序查找、增、删、改)效率。

5. 数据库的建立和维护

包括数据库的初始建立、数据的转换、数据库的转储和恢复、数据库的重组织和重构造以及性能监测分析等功能。

6. 其他功能

包括 DBMS 与网络中其他软件系统的通信功能；一个 DBMS 与另一个 DBMS 或文件系统的数据转换功能等。

1.5.2 DBMS的系统结构

1. DBMS 程序模块的组成

作为一个庞大的系统软件，DBMS 由众多程序模块组成，它们分别实现 DBMS 复杂而繁多的功能。正如前面所说的，不同的 DBMS功能并不完全相同，其差别主要表现在大型系统功能完备，实现这些功能时对各种情况考虑周全，涉及算法比较复杂，大多为通用系统；而小型系统则常常对各种情况作简化处理，涉及算法比较简单，对系统功能作必要的剪裁。但大致来说，DBMS 的程序模块可按功能划分，如图 1.11 所示。

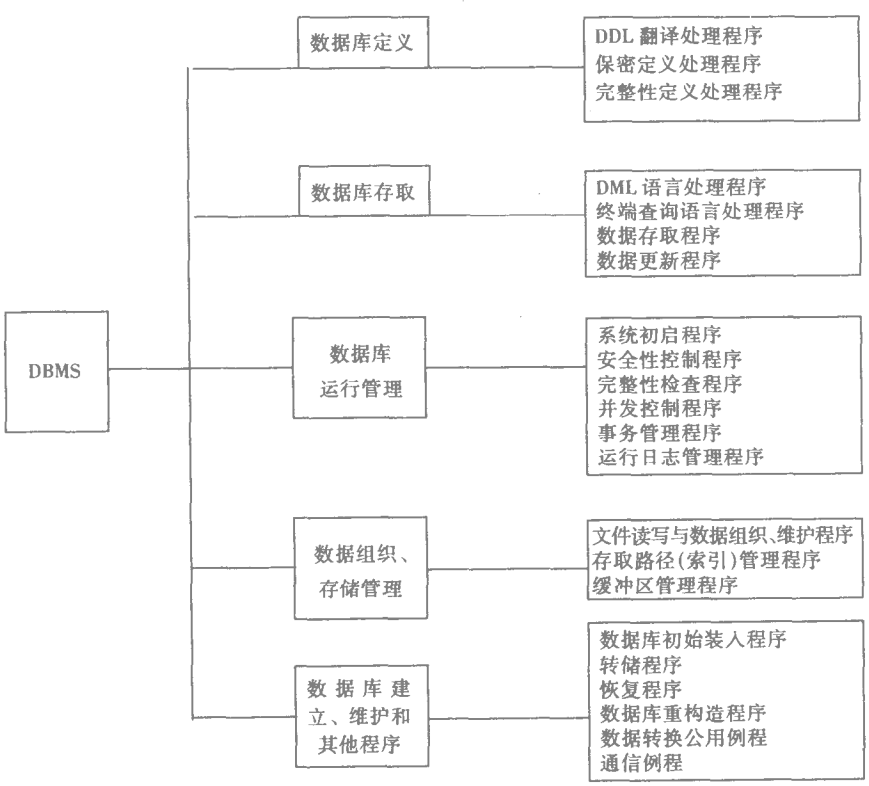


图 1.11 DBMS 的程序模块划分

(1) 数据定义方面 有 DDL 翻译处理程序(包括外模式、模式、存储模式处理程序)、保密定义处理程序(如授权定义处理程序)、完整性约束定义处理程序等，这些程序接收相应的定义，进行语法、语义检查，把它们翻译为内部格式存储在数据字典中。DDL 翻译程序还根据模式定义负责建立数据库的框架（即形成一个空库），等待装入数据。

(2) 数据操纵方面 有 DML 语言处理程序、终端查询语言解释程序、数据存取程序、数据更新程序等，DML 语言处理程序或终端查询语言解释程序对用户数据操纵请求进行语法、语义检查，由数据存取或更新程序完成对数据库的存取操作。

需要全本请访问 www.ertongbook.com