第 1 章　数据库基本知识

数据库技术是信息社会的重要基础技术之一，是计算机科学技术领域中发展最为迅速的重要分支。数据库技术是一门综合性技术，涉及到操作系统、数据结构、算法设计、程序设计等基础理论知识，因此，在计算机科学中是将其作为专门的学科来学习、研究的，并以之指导和推动应用。对普通计算机用户而言，虽更多注重于学习数据库技术的实际应用方法，但学习、掌握一些必需的、实用的基础知识，也是非常重要的，对数据库技术的应用，特别是在开发应用系统时尤为重要。因此，本章将以一定篇幅介绍数据库技术相关基础知识，使读者在学习、应用数据库技术的过程中，做到既知其然又知其所以然。

本章将简要介绍数据库、数据库系统、数据库管理系统、数据模型等基本概念以及数据库系统的体系结构，并着重介绍关系模式、关系、元组、属性、域等概念。

1.1　信息、数据与数据处理

1.1.1　数据与信息

人们通常使用各种各样的物理符号来表示客观事物的特性和特征，这些符号及其组合就是数据。数据的概念包括两个方面，即数据内容和数据形式。数据内容是指所描述客观事物的具体特性，也就是通常所说数据的“值”；数据形式则是指数据内容存储在媒体上的具体形式，也就是通常所说数据的“类型”。数据主要有数字、文字、声音、图形和图像等多种形式。

信息是指数据经过加工处理后所获取的有用知识。信息是以某种数据形式表现的。

数据和信息是两个相互联系、但又相互区别的概念；数据是信息的具体表现形式，信息是数据有意义的表现。

1.1.2　数据处理

数据处理就是将数据转换为信息的过程。数据处理的内容主要包括：数据的收集、整理、存储、加工、分类、维护、排序、检索和传输等一系列活动的总和。数据处理的目的是从大量的数据中，根据数据自身的规律和及其相互联系，通过分析、归纳、推理等科学方法，利用计算机技术、数据库技术等技术手段，提取有效的信息资源，为进一步分析、管理、决策提供依据。数据处理也称信息处理。

例如，学生各门成绩为原始数据，经过计算得出平均成绩和总成绩等信息，计算处理的过程就是数据处理。

1.1.3　数据处理的发展

伴随着计算机技术的不断发展，数据处理及时地应用了这一先进的技术手段，使数据处理的效率和深度大大提高，也促使数据处理和数据管理的技术得到了很大的发展，其发展过程

大致经历了人工管理、文件管理、数据库管理及分布式数据库管理等四个阶段。

1. 人工管理阶段

早期的计算机主要用于科学计算，计算处理的数据量很小，基本上不存在数据管理的问题。从 20 世纪 50 年代初，开始将计算机应用于数据处理。当时的计算机没有专门管理数据的软件，也没有像磁盘这样可随机存取的外部存储设备，对数据的管理没有一定的格式，数据依附于处理它的应用程序，使数据和应用程序一一对应，互为依赖。

由于数据与应用程序的对应、依赖关系，应用程序中的数据无法被其他程序利用，程序与程序之间存在着大量重复数据，称为数据冗余；同时，由于数据是对应某一应用程序的，使得数据的独立性很差，如果数据的类型、结构、存取方式或输入输出方式发生变化，处理它的程序必须相应改变，数据结构性差，而且数据不能长期保存。

在人工管理阶段，应用程序与数据之间的关系如图 1-1 所示。

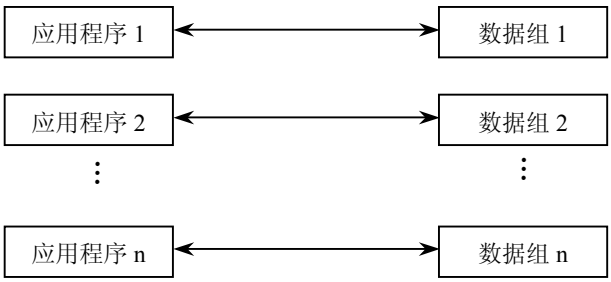

图 1-1　人工管理阶段程序与数据的关系

2. 文件管理阶段

从 20 世纪 50 年代后期开始至 60 年代末为文件管理阶段，应用程序通过专门管理数据的软件即文件系统管理来使用数据。由于计算机存储技术的发展和操作系统的出现，同时计算机硬件也已经具有可直接存取的磁盘、磁带及磁鼓等外部存储设备，软件则出现了高级语言和操作系统，而操作系统的一项主要功能是文件管理。因此，数据处理应用程序利用操作系统的文件管理功能，将相关数据按一定的规则构成文件，通过文件系统对文件中的数据进行存取、管理，实现数据的文件管理方式。

文件管理阶段中，文件系统为程序与数据之间提供了一个公共接口，使应用程序采用统一的存取方法来存取、操作数据，程序与数据之间不再是直接的对应关系，因而程序和数据有了一定的独立性。但文件系统只是简单地存放数据，数据的存取在很大程序上仍依赖于应用程序，不同程序难于共享同一数据文件，数据独立性较差。此外，由于文件系统没有一个相应的模型约束数据的存储，因而仍有较高的数据冗余，这又极易造成数据的不一致性。

在文件管理阶段，应用程序与数据之间的关系如图 1-2 所示。

3. 数据库管理阶段

数据库管理阶段是 20 世纪 60 年代末在文件管理基础上发展起来的。随着计算机系统性价比的持续提高，软件技术的不断发展，人们克服了文件系统的不足，开发了一类新的数据管理软件——数据库管理系统（DataBase Management System，DBMS），运用数据库技术进行数据管理，将数据管理技术推向了数据库管理阶段。

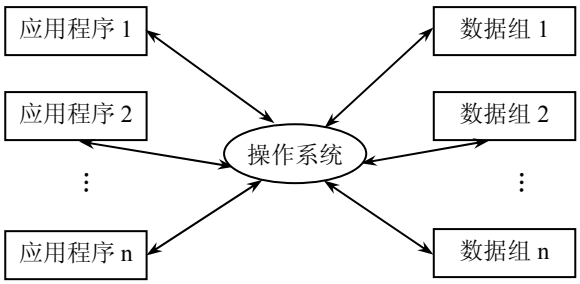

图 1-2　文件管理阶段程序与数据的关系

数据库技术使数据有了统一的结构，对所有的数据实行统一、集中、独立的管理，以实现数据的共享，保证数据的完整性和安全性，提高了数据管理效率。数据库也是以文件方式存储数据的，但它是数据的一种高级组织形式。在应用程序和数据库之间，由数据库管理软件 DBMS 把所有应用程序中使用的相关数据汇集起来，按统一的数据模型，以记录为单位存储在数据库中，为各个应用程序提供方便、快捷的查询、使用。

数据库系统与文件系统的区别是：数据库中数据的存储是按同一结构进行的，不同的应用程序都可直接操作使用这些数据，应用程序与数据间保持高度的独立性；数据库系统提供一套有效的管理手段，保持数据的完整性、一致性和安全性，使数据具有充分的共享性；数据库系统还为用户管理、控制数据的操作，提供了功能强大的操作命令，使用户直接使用命令或将命令嵌入应用程序中，简单方便地实现数据库的管理、控制操作。在数据库管理阶段，应用程序与数据之间的关系如图 1-3 所示。

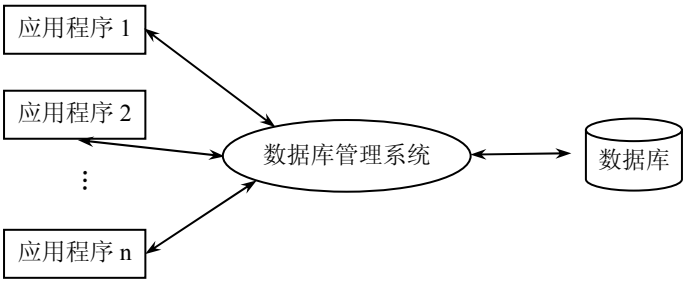

图 1-3　应用程序与数据之间的关系

1.1.4　数据库技术的发展

数据库技术萌芽于 20 世纪 60 年代中期，到 60 年代末 70 年代初出现了三个事件，标志着数据库技术日趋成熟，并有了坚实的理论基础。

（1）1969 年 IBM 公司研制、开发了数据库管理系统商品化软件 IMS（Information Management System），IMS 的数据模型是层次结构的。

（2）美国数据系统语言协会 CODASYL（Conference On Data System Language）下属的数据库任务组 DBTG（Data Base Task Group）对数据库方法进行系统的讨论、研究，提出了若干报告，成为 DBTG 报告。DBTG 报告确定并且建立了数据库系统的许多概念、方法和技术。DBTG 所提议的方法是基于网状结构的，它是网状模型的基础和典型代表。

（3）1970 年 IBM 公司 San Jose 研究实验室的研究员 E.F.Codd 发表了著名的“大型共享系统的关系数据库的关系模型”论文，为关系数据库技术奠定了理论基础。

自 20 世纪 70 年代开始，数据库技术有了很大的发展，表现为：

（1）数据库方法，特别是 DBTG 方法和思想应用于各种计算机系统，出现了许多商品化数据库系统。它们大都是基于网状模型和层次模型的。

（2）商用系统的运行，使数据库技术日益广泛地应用到企业管理、事务处理、交通运输、信息检索、军事指挥、政府管理、辅助决策等各个方面，深入到生产、生活的各个领域。数据库技术成为实现和优化信息系统的基本技术。

（3）关系方法的理论研究和软件系统的研制取得了很大的成果。IBM 公司 San Jose 研究实验室在 IBM 370 系列计算机上研究关系数据库系统 System R 获得成功，1981 年 IBM 公司又宣布了具有 System R 全部特征的新的数据库软件产品 SQL/DS 问世。与此同时，美国加州柏克利分校也研制出 INGRES 关系数据库实验系统，并紧接着推出了商用 INGRES 软件系统，使关系方法从实验室走向社会。

20 世纪 80 年代开始，几乎所有新开发的数据库系统都是关系数据库系统，随着微型计算机的出现与迅速普及，运行于微机的关系数据库系统也越来越丰富，性能越来越好，功能越来越强，应用遍及各个领域，为人类迈入信息时代起到了推波助澜的作用。

1.1.5 数据库新技术

数据库技术发展之快、应用之广是计算机科学其他领域技术无法比拟的。随着数据库应用领域的不断扩大和信息量的急剧增长，占主导地位的关系数据库系统已不能满足新的应用领域的需求，如 CAD（计算机辅助设计）、CAM（计算机辅助制造）、CIMS（计算机集成制造系统）、CASE（计算机辅助软件工程）、OA（办公自动化）、GIS（地理信息系统）、MIS（管理信息系统）、KBS（知识库系统）等，都需要数据库新技术的支持。这些新应用领域的特点是：存储和处理的对象复杂，对象间的联系具有复杂的语义信息；需要复杂的数据类型支持，包括抽象数据类型、无结构的超长数据、时间和版本数据等；需要常驻内存的对象管理以及支持对大量对象的存取和计算；支持长事务和嵌套事务的处理。这些需求是传统关系数据库系统难以满足的。

自 20 世纪 60 年代中期以来，数据库技术与其他领域的技术相结合，出现了数据库的许多新的分支，如：与网络技术相结合出现了网络数据库、与分布处理技术相结合出现了分布式数据库；与面向对象技术相结合出现了面向对象数据库；与人工智能技术相结合出现了知识库、主动数据库；与并行处理技术相结合出现了并行数据库；与多媒体技术相结合出现了多媒体数据库。此外，针对不同应用领域出现了工程数据库、实时数据库、空间数据库、地理数据库、统计数据库、时态数据库、数据仓库等多种数据库及相关技术。

1. 分布式数据库

分布式数据库系统（Distributed DataBase System，DDBS）是在集中式数据库基础上发展起来的，是数据库技术与计算机网络技术、分布处理技术相结合的产物。分布式数据库系统是地理上分布在计算机网络不同结点，逻辑上属于同一系统的数据库系统，它不同于将数据存储在服务器上供用户共享存取的网络数据库系统，分布式数据库系统不仅能支持局部应用，存取本地结点或另一个结点的数据，而且能支持全局应用，同时存取两个或两个以上结点的数据。

分布式数据库系统的主要特点是：

（1）数据是分布的。数据库中的数据分布在计算机网络的不同结点上，而不是集中在一个结点，区别于数据存放在服务器上由各用户共享的网络数据库系统。

（2）数据是逻辑相关的。分布在不同结点的数据，逻辑上属于同一个数据库系统，数据间存在相互关联，区别于由计算机网络连接的多个独立数据库系统。

（3）结点的自治性。每个结点都有自己的计算机软、硬件资源、数据库、数据库管理系统（即 Local DataBase Management System，LDBMS 局部数据库管理系统），因而能够独立地管理局部数据库。局部数据库中的数据可仅供本结点用户存取使用，也可供其他结点上的用户存取使用，提供全局应用。

2. 面向对象数据库

面向对象数据库系统（Object-Oriented DataBase System，OODBS）是将面向对象的模型、方法和机制，与先进的数据库技术有机地结合而形成的新型数据库系统。它从关系模型中脱离出来，强调在数据库框架中发展类型、数据抽象、继承和持久性；它的基本设计思想是，一方面把面向对象语言向数据库方向扩展，使应用程序能够存取并处理对象，另一方面扩展数据库系统，使其具有面向对象的特征，提供一种综合的语义数据建模概念集，以便对现实世界中复杂应用的实体和联系建模。因此，面向对象数据库系统首先是一个数据库系统，具备数据库系统的基本功能，其次是一个面向对象的系统，针对面向对象的程序设计语言的永久性对象存储管理而设计的，充分支持完整的面向对象概念和机制。

3. 多媒体数据库

多媒体数据库系统（Multi-media Database System，MDBS）是数据库技术与多媒体技术相结合的产物。在许多数据库应用领域中，都涉及到大量的文字、图形、图像、声音等多媒体数据，这些与传统的数字、字符等格式化数据有很大的不同，都是一些结构复杂的对象。这主要体现为如下几点：

（1）数据量大。格式化数据的数据量小，而多媒体数据量一般都很大，1 分钟视频和音频数据就需要几十兆数据空间。

（2）结构复杂。传统的数据以记录为单位，一个记录由多个字段组成，结构简单，而多媒体数据种类繁多、结构复杂，大多是非结构化数据，来源于不同的媒体且具有不同的形式和格式。

（3）时序性。文字、声音或图像组成的复杂对象需要有一定的同步机制，如一幅画面的配音或文字需要同步，既不能超前也不能滞后，而传统数据无此要求。

（4）数据传输的连续性。多媒体数据如声音或视频数据的传输必须是连续、稳定的，不能间断，否则出现失真而影响效果。多媒体数据的这些特点，使系统不能像格式化数据一样去管理和处理多媒体数据，也不能简单地通过扩充传统数据库来满足多媒体应用的需求，因此，多媒体数据库需要有特殊的数据结构、存储技术、查询和处理方式。

从实际应用的角度考虑，多媒体数据库管理系统（MDBMS）应具有如下基本功能：

（1）应能够有效地表示多种媒体数据，对不同媒体的数据如文本、图形、图像、声音等能够按应用的不同，采用不同的表示方法。

（2）应能够处理各种媒体数据，正确识别和表现各种媒体数据的特征，各种媒体间的空间或时间关联。

（3）应能够像其他格式化数据一样对多媒体数据进行操作，包括对多媒体数据的浏览、查询检索，对不同的媒体提供不同的操纵，如声音的合成、图像的缩放等。

（4）应具有开放功能，提供多媒体数据库的应用程序接口等。

4. 数据仓库

信息技术的高速发展，数据和数据库在急剧增长，数据库应用的规模、范围和深度不断扩大，一般的事务处理已不能满足应用的需要，企业界需要在大量信息数据基础上的决策支持（Decision Support，DS），数据仓库（Data Warehousing，DW）技术的兴起满足了这一需求。数据仓库可以提供对企业数据的方便访问和强大的分析工具，从企业数据中获得有价值的信息，发掘企业的竞争优势，提高企业的运营效率和指导企业决策。数据仓库作为决策支持系统（Decision Support System，DSS）的有效解决方案，涉及三方面的技术内容：数据仓库技术、联机分析处理（On-Line Analysis Processing，OLAP）技术和数据挖掘（Data Mining，DM）技术。

数据库技术作为数据管理的一种有效手段主要用于事务处理，但随着应用的深入，人们发现对数据库的应用可分为两类：操作型处理和分析型处理。操作型处理也称为联机事务处理（On-Line Transaction Processing，OLTP），它是指对企业数据进行日常的业务处理，这类处理主要是针对企业数据库的一个或一批记录进行查询检索或更新操作。与联机事务处理不同的是，分析型处理主要用于管理人员的决策分析，通过对大量数据的综合、统计和分析，得出有利于企业的决策信息，但若按事务处理的模式进行分析处理，则得不到令人满意的结果，而数据仓库和联机分析处理等技术能够以统一的模式，从多个数据源收集数据提供用户进行决策分析。

数据仓库不是一种产品，而是由软硬件技术组成的环境。它将企业内部各种跨平台的数据，经过重新组合和加工，构成面向决策的数据仓库，为企业决策者方便地分析企业发展状况并做出决策，提供有效的途径。

1.2 数据库系统

1.2.1 数据库系统的组成

数据库应用系统简称为数据库系统（DataBase System，DBS），是一个计算机应用系统。它由计算机硬件、数据库管理系统、数据库、应用程序和用户等部分组成。

1. 计算机硬件

计算机硬件（Hardware）是数据库系统赖以存在的物质基础，是存储数据库及运行数据库管理系统 DBMS 的硬件资源，主要包括主机、存储设备、I/O 通道等。大型数据库系统一般都建立在计算机网络环境下。

为使数据库系统获得较满意的运行效果，应对计算机的 CPU、内存、磁盘、I/O 通道等技术性能指标，采用较高的配置。

2. 数据库管理系统

数据库管理系统（DataBase Management System，DBMS）是指负责数据库存取、维护、管理的系统软件。DBMS 提供对数据库中数据资源进行统一管理和控制的功能，将用户应用程序与数据库数据相互隔离。它是数据库系统的核心，其功能的强弱是衡量数据库系统性能优

劣的主要指标。

DBMS 必须运行在相应的系统平台上，在操作系统和相关的系统软件支持下，才能有效地运行。

3. 数据库

数据库（DataBase，DB）是指数据库系统中以一定组织方式将相关数据组织在一起，存储在外部存储设备上所形成的、能为多个用户共享的、与应用程序相互独立的相关数据集合。数据库中的数据也是以文件的形式存储在存储介质上的，它是数据库系统操作的对象和结果。数据库中的数据具有集中性和共享性。所谓集中性是指把数据库看成性质不同的数据文件的集合，其中的数据冗余很小。所谓共享性是指多个不同用户使用不同语言，为了不同应用目的可同时存取数据库中的数据。

数据库中的数据由 DBMS 进行统一管理和控制，用户对数据库进行的各种数据操作都是通过 DBMS 实现的。

4. 应用程序

应用程序（Application）是在 DBMS 的基础上，由用户根据应用的实际需要所开发的、处理特定业务的应用程序。应用程序的操作范围通常仅是数据库的一个子集，也即用户所需的那部分数据。

5. 数据库用户

用户（User）是指管理、开发、使用数据库系统的所有人员，通常包括数据库管理员、应用程序员和终端用户。数据库管理员（DataBase Administrator，DBA）负责管理、监督、维护数据库系统的正常运行；应用程序员（Application Programmer）负责分析、设计、开发、维护数据库系统中运行的各类应用程序；终端用户（End-User）是在 DBMS 与应用程序支持下，操作使用数据库系统的普通使用者。不同规模的数据库系统，用户的人员配置可以根据实际情况有所不同，大多数用户都属于终端用户，在小型数据库系统中，特别是在微机上运行的数据库系统中，通常 DBA 就由终端用户担任。

如图 1-4 所示是数据库管理系统与计算机硬件及其他软件的层次关系，外层应用依赖于内层资源的支持。

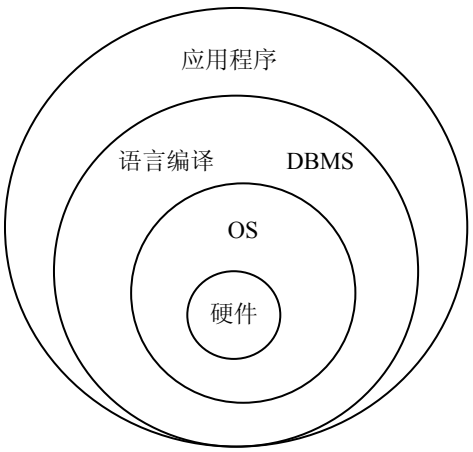

图 1-4　软硬件的层次关系

综上所述，数据库中包含的数据，是存储在存储介质上的数据文件的集合；每个用户均可使用其中的部分数据，不同用户使用的数据可以重叠，同一组数据可以为多个用户共享；DBMS 为用户提供对数据的存储组织、操作管理功能；用户通过 DBMS 和应用程序实现数据库系统的操作与应用。

1.2.2 数据库系统体系结构

为了有效地组织、管理数据，提高数据库的逻辑独立性和物理独立性，人们为数据库设计了一个严谨的体系结构，包括 3 个模式（外模式、模式和内模式）和 2 个映射（外模式－模式映射和模式－内模式映射）。美国 ANSI/X3/SPARC 的数据库管理系统研究小组于 1975 年、1978 年提出了标准化的建议，将数据库结构分为 3 级：面向用户或应用程序员的用户级；面向建立和维护数据库人员的概念级；面向系统程序员的物理级。用户级对应外模式，概念级对应模式，物理级对应内模式，使不同级别的用户对数据库形成不同的视图。所谓视图，就是指观察、认识和理解数据的范围、角度和方法，简而言之，视图就是数据库在用户“眼中”的反映，很显然，不同层次（级别）用户所“看到”的数据库是不相同的。数据库系统的体系结构如图 1-5 所示。

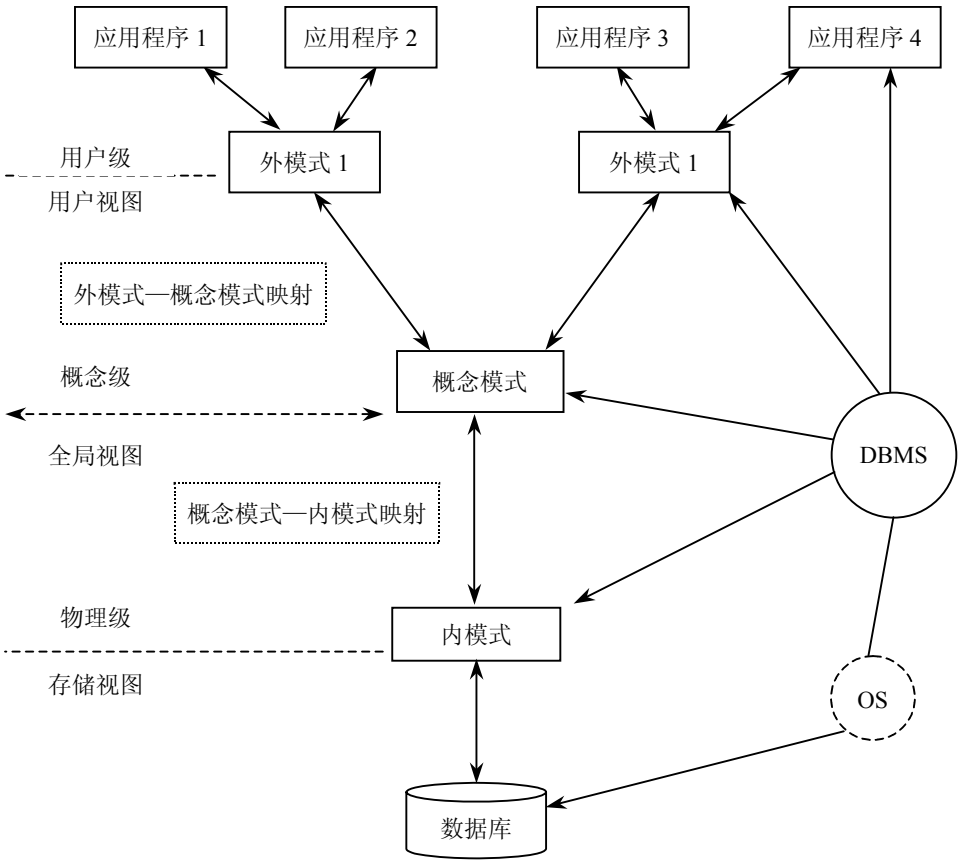

图 1-5 数据库系统的体系结构

1. 模式

模式又称概念模式或逻辑模式，对应于概念级。它是由数据库设计者综合所有用户的数据，按照统一的观点构造的全局逻辑结构。是对数据库中全部数据的逻辑结构和特征的总体描述，是所有用户的公共数据视图（全局视图）。它是由数据库系统提供的数据模式描述语言（Data Description Language，模式 DDL）来描述、定义的，体现、反映了数据库系统的整体观。

2. 外模式

外模式又称子模式，对应于用户级。它是某个或某几个用户所看到的数据库的数据视图，是与某一应用有关的数据的逻辑表示。外模式是从模式导出的一个子集，包含模式中允许特定用户使用的那部分数据。用户可以通过外模式描述语言（外模式 DLL）来描述、定义对应于用户的数据记录（外模式），也可以利用数据操纵语言（Data Manipulation Language，DML）对这些数据记录进行操作。外模式反映了数据库的用户观。

3. 内模式

内模式又称存储模式，对应于物理级。它是数据库中全体数据的内部表示或底层描述，是数据库最低一级的逻辑描述，它描述了数据在存储介质上的存储方式和物理结构，对应着实际存储在外存储介质上的数据库。内模式由内模式描述语言（内模式 DLL）来描述、定义。

4. 数据库系统的二级映射

数据库系统的三级模式是数据在三个级别（层次）上的抽象，使用户能够逻辑地、抽象地处理数据而不必关心数据在计算机中的物理表示和存储。实际上，对于一个数据库系统而言，只有物理级数据库是客观存在的，它是进行数据库操作的基础，概念级数据库中不过是物理数据库的一种逻辑的、抽象的描述（即模式），用户级数据库则是用户与数据库的接口，它是概念级数据库的一个子集（外模式）。

用户应用程序根据外模式进行数据操作，通过外模式－模式映射，定义和建立某个外模式与模式间的对应关系，将外模式与模式联系起来，当模式发生改变时，只要改变其映射，就可以使外模式保持不变，对应的应用程序也可保持不变；另一方面，通过模式－内模式映射，定义建立数据的逻辑结构（模式）与存储结构（内模式）间的对应关系，当数据的存储结构发生变化时，只需改变模式－内模式映射，就能保持模式不变，因此应用程序也可以保持不变。正是通过这两级映射，将用户对数据库的逻辑操作最终转换成对数据库的物理操作，在这一过程中，用户不必关心数据库全局，更不必关心物理数据库，用户面对的只是外模式，因此，换来了用户操作、使用数据库的方便。这两级映射转换是由 DBMS 实现的，它将用户对数据库的操作，从用户级转换到物理级。

1.2.3　数据库管理系统的功能

作为数据库系统核心软件的数据库管理系统 DBMS，通过三级模式间的映射转换，为用户实现了数据库的建立、使用、维护操作，因此，DBMS 必须具备相应的功能。它主要包括如下功能：

1. 数据库定义（描述）功能

DBMS 为数据库的建立提供了数据定义（描述）语言（DDL）。用户使用 DDL 定义数据库结构的子模式（外模式）、模式和内模式；定义各个外模式与模式之间的映射；定义模式与存储模式之间的映射；定义有关约束条件等。

2. 数据库操纵功能

DBMS 提供数据操纵语言（DML）实现对数据库检索、插入、修改、删除等基本操作。DML 通常分为两类：一类是嵌入主语言中的，如嵌入 C、COBOL 等高级语言中，这类 DML 一般本身不能独立使用，称之为宿主型语言；另一类是交互式命令语言，它语法简单，可独立使用，称之为自含型语言。目前 DBMS 广泛采用的就是可独立使用的自含型语言，为用户或应用程序员提供操作使用数据库的语言工具。SQL Server 中 Transact-SQL 既可作为嵌入式语言使用，也是自含型语言。

3. 数据库运行管理功能

对数据库的运行进行管理是 DBMS 运行的核心部分，包括对数据库进行并发控制、安全性检查、存取控制（即存取权限检查）、完整性约束条件的检查及执行数据库内部维护等。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，以保证数据库的安全性、完整性、一致性及多用户对数据库的并发操作，保证数据库的正确有效。

4. 数据组织、存储和管理

数据库中需要存放多种数据，如数据字典、用户数据、存取路径等，DBMS 负责分门别类地组织、存储和管理这些数据，确定以何种文件结构和存取方式物理地组织这些数据。如何实现数据之间的联系，以便提高存储空间的利用率以及提高随机查找、顺序查找、增、删、改等操作的时间效率。

5. 数据库的建立和维护

建立数据库包括数据库的初始数据的输入与数据转换等。维护数据库包括数据库的转存与恢复、数据库的重组织与重构、性能的监视与分析等。

6. 通信功能

作为用户与数据库的接口，用户可以通过交互式或应用程序方式使用数据库。交互式是通过系统提供的实用程序与数据库进行通信，应用程序方式则是应用程序员依据外模式（子模式）编写应用程序模块，实现对数据库中数据的各种操作。另外，DBMS 还提供与其他软件系统进行通信的功能，例如提供与其他 DBMS 或文件系统的接口，从而能够将数据转换为另一个 DBMS 或文件系统能接收的格式，或者接收其他 DBMS 或文件系统的数据。如 SQL Server 可以与 Oracle、Excel 进行数据转换操作。

1.2.4 数据库管理系统的组成

为了提供上述 6 方面的功能，DBMS 通常由以下 4 个部分组成。

1. 数据定义语言及其编译处理程序

DBMS 一般都提供数据定义语言供用户定义数据库的模式、存储模式、外模式、各级模式间的映射、有关约束条件等。用 DDL 定义的模式、存储模式、外模式分别称为源模式、源存储模式、源外模式，各种模式编译程序负责将它们翻译成相应的内部表示，即生成目标模式、目标存储模式、目标外模式。这些目标模式描述的是数据库的框架，而不是数据本身。这些描述存放在数据字典（也称系统目录）中，作为 DBMS 存取和管理数据的基本依据。

2. 数据操作语言及其编译程序

DBMS 提供了数据操纵语言实现对数据库的检索、插入、修改、删除等基本操作。

3. 数据库运行控制程序

DBMS 提供了一些系统运行控制程序负责数据库运行过程的控制与管理，包括系统初启程序、文件读写与维护程序、存取路径管理程序、缓冲区管理程序、安全控制及事务管理程序等。

4. 实用程序

DBMS 通常还提供一些实用程序，包括数据库转存程序、数据库恢复程序、性能监测程序及通信程序等。用户可以利用这些实用程序对系统进行配置、监视和管理。

1.2.5　数据库系统的特点

数据库系统的出现是计算机数据处理技术的重大进步，它具有以下特点。

1. 数据共享

数据共享是指多个用户可以同时存取数据而不相互影响。数据共享包括以下三个方面：所有用户可以同时存取数据;数据库不仅可以为当前的用户服务,也可以为将来的新用户服务;可以使用多种语言完成与数据库的接口。

2. 减少数据冗余

数据冗余就是数据重复，数据冗余既浪费存储空间，又容易产生数据的不一致。在非数据库系统中，由于每个应用程序都有自己的数据文件，所以数据存在着大量的重复。

数据库从全局观念来组织和存储数据，数据已经根据特定的数据模型结构化，从而有效地节省了存储资源，减少了数据冗余，增强了数据的一致性。

3. 具有较高的数据独立性

所谓数据独立是指数据与应用程序之间的彼此独立，它们之间不存在相互依赖的关系。应用程序不必随数据存储结构的改变而变动，这是数据库一个最基本的优点。

在数据库系统中，数据库管理系统通过映像，实现了应用程序对数据的逻辑结构与物理存储结构之间较高的独立性。数据库的数据独立包括两个方面：

（1）物理数据独立：数据的存储格式和组织方法改变时，不影响数据库的逻辑结构，从而不影响应用程序。

（2）逻辑数据独立：数据库逻辑结构的变化（如数据定义的修改，数据间联系的变更等）不影响用户的应用程序。

数据独立提高了数据处理系统的稳定性，从而提高了程序维护的效益。

4. 增强了数据安全性和完整性保护

数据库加入了安全保密机制，可以防止对数据的非法存取。由于实行集中控制，有利于控制数据的完整性。数据库系统采取了并发访问控制，保证了数据的正确性。另外，数据库系统还采取了一系列措施，实现了对数据库破坏的恢复。

1.3　数据模型

1.3.1　现实世界的数据描述

现实世界是存在于人脑之外的客观世界，是数据库系统操作处理的对象。如何用数据来描述、解释现实世界，运用数据库技术表示、处理客观事物及其相互关系，则需要采取相应的

方法和手段进行描述，进而实现最终的操作处理。

1. 信息处理的三个层次

计算机信息处理的对象是现实生活中的客观事物，在对客观事物实施处理的过程中，首先要经历了解、熟悉的过程，从观测中抽象出大量描述客观事物的信息，再对这些信息进行整理、分类和规范，进而将规范化的信息数据化，最终由数据库系统存储、处理。在这一过程中，涉及到三个层次，经历了两次抽象和转换。

（1）现实世界。

现实世界就是存在于人脑之外的客观世界，客观事物及其相互联系就处于现实世界中。客观事物可以用对象和性质来描述；

（2）信息世界。

信息世界就是现实世界在人们头脑中的反映，又称观念世界。客观事物在信息世界中称为实体，反映事物间联系的是实体模型或概念模型。现实世界是物质的，相对而言信息世界是抽象的；

（3）数据世界。

数据世界就是信息世界中的信息数据化后对应的产物。现实世界中的客观事物及其联系，在数据世界中以数据模型描述。相对于信息世界，数据世界是量化的、物化的。

因此，客观事物是信息之源，是设计、建立数据库的出发点，也是使用数据库的最后归宿。概念模型和数据模型是对客观事物及其相互联系的两种抽象描述，实现了信息处理三个层次间的对应转换，而数据模型是数据库系统的核心和基础。

2. 信息世界中的基本概念

（1）实体。

客观事物在信息世界中称为实体（Entity），它是现实世界中任何可区分、识别的事物。实体可以是具体的人或物，如张三同学、天安门城楼，也可以是抽象概念，如一个人，一所学校；

（2）属性。

实体具有许多特性，实体所具有的特性称为属性（Attribute）。一个实体可用若干属性来刻画。例如学生实体可以用学号、姓名、性别、出生年份、入校时间等属性来描述。

（3）域。

属性的取值范围称为该属性的域。例如，规定学生的学号为 8 位整数，性别的域为（男，女）。

（4）实体型和实体值。

实体型就是实体的结构描述，通常是实体名和属性名的集合。具有相同属性的实体，有相同的实体型。如学生实体型可以是：学生（学号，姓名，性别，年龄）；实体值是一个具体的实体，是属性值的集合。如学生李建国的实体值是：（011110，李建国，男，19）；

（5）实体集。

性质相同的同类实体的集合称实体集。如一个班的学生。

（6）实体联系。

建立实体模型的一个主要任务就是要确定实体之间的联系。常见的实体联系有三种。如图 1-6 所示。

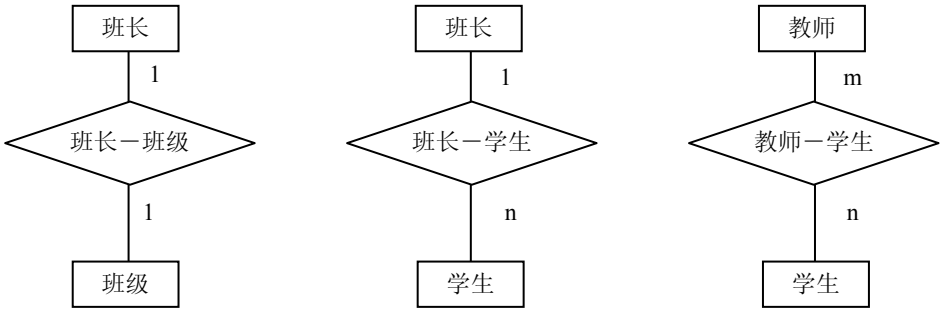

图 1-6　实体间的联系

1）一对一联系（1:1）。

若两个不同型实体集中，任一方的一个实体只与另一方的一个实体相对应，称这种联系为一对一联系。如班长与班级的联系，一个班级只有一个班长，一个班长对应一个班级。如图 1-6（a）所示。

2）一对多联系（1:n）。

若两个不同型实体集中，一方的一个实体对应另一方若干个实体，而另一方的一个实体只对应本方一个实体，称这种联系为一对多联系。如班长与学生的联系，一个班长对应多个学生，而本班每个学生只对应一个班长。如图 1-6（b）所示。

3）多对多联系（m:n）。

若两个不同型实体集中，两实体集中任一实体均与另一实体集中若干个实体对应，称这种联系为多对多联系。如教师与学生的联系，一位教师为多个学生授课，每个学生也有多位任课教师。如图 1-6（c）所示。

3. 实体模型

实体模型又称概念模型，它是反映实体之间联系的模型。数据库设计的重要任务就是建立实体模型，建立概念数据库的具体描述。在建立实体模型时，实体要逐一命名以示区别，并描述它们之间的各种联系。实体模型只是将现实世界的客观对象抽象为某种信息结构，这种信息结构并不依赖于具体的计算机系统，E-R 图是目前常用的概念模型的表示方法。

1.3.2　数据模型

数据模型是指数据库中数据与数据之间的关系。数据模型是数据库系统中一个关键概念，数据模型不同，相应的数据库系统就完全不同，任何一个数据库管理系统都是基于某种数据模型的。数据库管理系统常用的数据模型有下列三种：层次模型、网状模型、关系模型。

1. 层次数据模型（Hierarchical Model）

用树形结构表示实体和实体间联系的数据模型称为层次模型。

树是由结点和连线组成，结点表示数据集，连线表示数据之间的联系，树形结构只能表示一对多联系。通常将表示“一”的数据放在上方，称为父结点；而表示“多”的数据放在下方，称为子结点。树的最高位置只有一个结点，称为根结点。根结点以外的其他结点都有一个父结点与它相连，同时可能有一个或多个子结点与它相连。没有子结点的结点称为叶结点，它

处于分枝的末端。

层次模型的基本特点：

（1）有且仅有一个结点无父结点，称其为根结点。

（2）其他结点有且只一个父结点。

支持层次数据模型的 DBMS 称为层次数据库管理系统，在这种系统中建立的数据库是层次数据库。层次模型可以直接方便地表示一对一联系和一对多联系，但不能用它直接表示多对多联系。

2. 网状数据模型（Network Model）

用网状结构表示实体和实体间的联系的数据模型称为网状模型。网状模型是层次模型的拓展，网状模型的结点间可以任意发生联系，能够表示各种复杂的联系。

网状模型的基本特点：

（1）一个以上结点无父结点。

（2）至少有一结点有多于一个的父结点。

网状模型和层次模型在本质上是一样的，从逻辑上看，它们都是用结点表示数据，用连线表示数据间的联系，从物理上看，层次模型和网络模型都是用指针来实现两个文件之间的联系。层次模型和网状模型的差别在于网状模型中的连线或指针更加复杂，更加纵横交错，从而数据结构更加复杂。

层次模型是网状模型的特殊形式，网状模型是层次模型的一般形式。

支持网状模型的 DBMS 称为网状数据库管理系统，在这种系统中建立的数据库是网状数据库。网状结构可以直接表示多对多联系，这也是网状模型的主要优点，当然在一些已经实现的网状模型 DBMS 中，对这一点做了限制。

3. 关系模型（Relational Model）

用二维表来表示实体和实体间联系的数据模型称为关系模型。例如，在关系模型中可用如表 1-1 所示的形式表示学生对象。关系不但可以表示实体间一对多的联系，也可以方便地表示多对多的联系。

表 1-1　学生基本情况表

学号	姓名	性别	班级名	系别代号	地址	出生日期	是否团员	备注
011110	李建国	男	计 0121	01	湖北武汉	1984-9-28	是	
011103	李宁	女	电 0134	02	江西九江	1985-5-6	否	
011202	赵娜	女	英 0112	03	广西南宁	1984-2-21	否	
011111	赵琳	女	计 0121	01	江苏南京	1985-11-18	是	
021405	罗宇波	男	英 0112	03	江苏南通	1985-12-12	否	

关系模型是建立在关系代数基础上的，因而具有坚实的理论基础。与层次模型和网状模型相比，具有数据结构单一、理论严密、使用方便、易学易用的特点。

自 20 世纪 80 年代以来，新推出的数据库管理系统几乎都支持关系模型。早期许多层次和网状模型系统的产品也加上了支持关系模型的接口。目前，常用的数据库系统基本上都属于关系型数据库系统，如 SQL Server、Oracle、DB2 等都是常用的关系型 DBMS。

1.3.3　关系的基本概念及其特点

1. 关系的基本概念

（1）关系。

一个关系就是一张二维表，通常将一个没有重复行、重复列的二维表看成一个关系，每个关系都有一个关系名。例如表 1-1 的学生基本情况表。

（2）元组。

二维表的每一行在关系中称为元组。

（3）属性。

二维表的每一列在关系中称为属性，每个属性都有一个属性名，属性值则是各个元组在该属性上的取值。例如，表 1-1 中第二列，“姓名”是属性名，“李建国”则为第一个元组在“姓名”属性上的取值。

（4）域。

属性的取值范围称为域。域作为属性值的集合，其类型与范围具体由属性的性质及其所表示的意义确定。如表 1-1 中“性别”属性的域是{男，女}。

2. 关系模型的主要优点

关系模型具有如下优点：

（1）数据结构单一。

关系模型中，不管是实体还是实体之间的联系，都用关系来表示，而关系都对应一张二维数据表，数据结构简单、清晰。

（2）关系规范化，并建立在严格的理论基础上。

关系中每个属性不可再分割，构成关系的基本规范。同时关系是建立在严格的数学概念基础上，具有坚实的理论基础。

（3）概念简单，操作方便。

关系模型最大的优点就是简单，用户容易理解和掌握，一个关系就是一张二维表格，用户只需用简单的查询语言就能对数据库进行操作。

1.4　关系数据库与关系代数

1.4.1　关系数据库概述

所谓关系数据库就是采用关系模型作为数据的组织方式，换句话说就是支持关系模型的数据库系统。

关系模型由三个部分构成：关系数据结构、关系数据操作和完整性约束。

1. 关系数据结构

关系模型的数据结构非常简单，实际上就是一张二维表，但这种简单的二维表却可以表达丰富的语义，可以很方便地描述出现实世界的实体以及实体之间的各种联系。

2. 关系数据操作

关系数据操作采用集合操作方式，即操作的对象和结果都是集合。关系数据操作包括查

询和更新两个部分：

1）查询：选择、投影、连接、除、并、交、差等。

2）更新：增加、删除以及修改。

以上这些操作在本章后面会做详细介绍和说明。

关系模型中的关系操作早期通常是用代数方式或逻辑方式来表示，分别称为关系代数和关系演算。从现代的角度来看，关系数据语言分为三类：

1）关系代数：用关系的运算来表达查询要求的方式；

2）关系演算：用谓词来表达查询要求的方式；

3）SQL 语言：结构化查询语言。

3. 完整性约束

完整性约束条件是关系数据模型的一个重要组成部分，是为了保证数据库中的数据一致性的。

完整性约束分为三类：实体完整性、参照完整性、用户定义的完整性。

1.4.2 关系数据结构

在关系模型中，实体与实体之间的联系都是用二维表（关系）来表示的，而前面已经讲过，关系模型是建立在集合代数基础上的，本节从集合论的角度来讨论关系数据结构的形式化定义。

1.4.2.1 关系

在关系中是用域来表示属性的取值范围。下面我们先来看域的定义。

1. 域

定义 域是一组具有相同数据类型的值的集合。域中所包含的值的个数叫做域的基数。域是需要命名的。例如：

D_1={李国庆 刘娇丽}，表示人名的集合

D_2={清华大学出版社 中国水利水电出版社}，表示出版社的集合，

D_3={数据结构 高等数学}，表示书名的集合

以上三个域的基数都是 2。

2. 笛卡尔积

定义 给定一组域 $D_1,D_2,D_3,\ldots D_n$，则这些域的笛卡尔积为：$D_1\times D_2\times D_3\times\cdots\times D_n=\{(d_1,d_2,d_3,\ldots,d_n)|d_i\in D_j,i=1,2,\ldots,n\}$，其中：

①每一个元组$(d_1,d_2,d_3,\ldots,d_n)$叫做一个 n 元组，简称**元组**。

②元组的每一个值 d_i 叫做一个**分量**。

③笛卡尔积的基数为：

$$m=\prod_{i=1}^{n} m_i$$

说明：①笛卡尔积实际上是一个二维表。

②表的框架由域构成。

③表的每一行对应一个元组。

④每一列数据来自同一个域。

对于上例的三个域 D_1、D_2、D_3，其笛卡积为：

$D_1 \times D_2 \times D_3$={（李国庆，清华大学出版社，数据结构），（李国庆，清华大学出版社，高等数学），（李国庆，中国水利水电出版社，数据结构），（李国庆，中国水利水电出版社，高等数学），（刘娇丽，清华大学出版社，数据结构），（刘娇丽，清华大学出版社，高等数学），（刘娇丽，中国水利水电出版社，数据结构），（刘娇丽，中国水利水电出版社，高等数学）}

其中：（李国庆，清华大学出版社，数据结构），（李国庆，清华大学出版社）等都是**元组**，李国庆，清华大学出版社，数据结构等都是**分量**。$D_1 \times D_2 \times D_3$ 的**基数**为 2×2×2=8。这些元组可以用一张二维表表示，见表 1-2。

表 1-2　D_1,D_2,D_3 的笛卡尔积

作者 Editor	出版社 Publish	图书名 Bookname
李国庆	清华大学出版社	数据结构
李国庆	清华大学出版社	高等数学
李国庆	中国水利水电出版社	数据结构
李国庆	中国水利水电出版社	高等数学
刘娇丽	清华大学出版社	数据结构
刘娇丽	清华大学出版社	高等数学
刘娇丽	中国水利水电出版社	数据结构
刘娇丽	中国水利水电出版社	高等数学

3. 关系

$D_1 \times D_2 \times \cdots \times D_n$ 的子集叫作在域 $D_1,D_2,\cdots,D_n$ 上的关系，用 $R(D_1,D_2,\cdots,D_n)$表示。其中 R 表示关系的名字，n 是关系的目或度（Degree）。

当 n=1 时，关系中仅含一个域，称为单元关系。

当 n=2 时，关系中仅含两个域，称为二元关系。

关系是笛卡尔积的子集，所以关系也是一个二维表，表的每行对应一个元组，表的每列对应一个域。由于域可以相同，为了加以区分，必须对给每列起一个名字，称为属性（Attribute）。n 目关系必有 n 个属性。

4. 码的定义

（1）码（Key）。在关系的各个属性中，能够用来唯一标识一个元组的属性或属性组。

（2）候选码（Candidate Key）。若在一个关系中，某一个属性或属性组的值能唯一地标识该关系的元组，而其真子集不行，则称该属性或属性组为候选码。

（3）主码（Primary Key）。若一个关系有多个候选码，则选定其中一个为主码（也称主键）。

（4）主属性（Prime Attribute）。候选码的诸属性称为主属性。

（5）非主属性（Non-Key Attribute）。不包含在任何候选码中的属性。

在最简单的情况下，候选码只包含一个属性。也就是说一个属性就可以唯一地标识一个元组。在最极端的情况下，关系模式的所有属性是这个关系模式的候选码，也就是说所有的属性加起来才可以唯一地标识一个元组，这种关系称为全码关系（all-key）。

5. 关系的三种类型

基本关系：基本关系通常又称为基本表或基表，指的是实实在在存在的表。

导出表：导出表是从一个或几个基本表进行查询而得到的结果所对应的表。

视图：是由基本表或其他视图表导出的表，是虚表，不对应实际存储的数据。

6. 基本关系的 6 条性质

性质 1 列是同质的，即每一列中的分量是同一类型的数据，来自同一个域。

性质 2 不同的列可出自同一个域，称其中的每一列为一个属性，不同的属性要给予不同的属性名。

性质 3 列的顺序无所谓，即列的次序可以任意交换。

性质 4 任意两个元组不能完全相同。这只是现实中的一般性要求，有些数据库是允许在同一张表中存在两个完全相同的元组的。

性质 5 行的顺序无所谓，即行的次序可以任意交换。

性质 6 分量必须取原子值，也就是说每一个分量都必须是不可分的数据项。

为了更好地说明以上概念，下面举一个图书管理系统的例子来说明。假设有三个关系，一个是图书关系 BOOK，一个是读者关系 READER，另一个是图书借阅关系 BORROW。三个关系分别见表 1-3，表 1-4 和表 1-5。请读者牢记这三个表的结构，本书后续章节对数据库的操作都是以这三个表为例进行讲解。

表 1-3 图书关系 BOOK

图书号 BookId	图书名 Bookname	编者 Editor	价格 Price	出版社 Publish	出版年月 PubDate	库存数 Qty
TP2001--001	数据结构	李国庆	22.00	清华大学出版社	2001-01-08	20
TP2003--002	数据结构	刘娇丽	18.9	中国水利水电出版社	2003-10-15	50
TP2002--001	高等数学	刘自强	12.00	中国水利水电出版社	2002-01-08	60
TP2003--001	数据库系统	汪 洋	14.00	人民邮电出版社	2003-05-18	26
TP2004--005	数据库原理与应用	刘淳	24	中国水利水电出版社	2004-07-25	100

表 1-4 读者关系 READER

借书卡号 CardId	读者姓名 Name	性别 Sex	工作单位 Dept	读者类别 Class
T0001	刘勇	男	计算机系	1
S0101	丁钰	女	人事处	2
S0111	张清蜂	男	培训部	3
T0002	张伟	女	计算机系	1

注：读者类别只有三种取值：1 代表学生；2 代表教师；3 代表临时读者。

对于关系 BOOK 来说，BookId 是能唯一标识元组的属性，所以 BookId 既是唯一候选码，也是主码，也是唯一主属性。每一个属性都是不可再分割的最小数据项。

对于关系 BORROW 来说，（BookId，CardId，Bdate）是可以唯一标识元组的属性组（一个读者在同一时间不能借二本相同的图书），而其真子集不行（一个读者在不同的时间可以借

阅以前曾借阅过的图书，所以 BookId、CardId 不是候选码），所以（BookId，CardId，Bdate）是 BORROW 表的候选码。

表 1-5　借书关系 BORROW

图书号 BookId	借书卡号 CardId	借书日期 Bdate	还书日期 Sdate
TP2003--002	T0001	2003-11-18	2003-12-09
TP2001--001	S0101	2003-02-28	2003-05-20
TP2003--001	S0111	2004-05-06	
TP2003--002	S0101	2004-02-08	

从表 1-5 我们可以看出，借书日期 Bdate 和还书日期 Sdate 都是来自于日期域，但要用不同的列名（属性名）来区分。

1.4.2.2　关系模式

所谓关系模式就是对关系的描述。描述的内容包括：

- 元组集合结构：有那些属性、属性来自那些域，属性与域之间的映像关系（属性的长度和类型）；
- 元组集合的语义；
- 完整性约束条件：属性间的相互关系，属性的取值范围限制。

概括来说，关系模式描述下列五个要素：关系名 R；属性名集合 U；属性来自的域 D；属性向域的映像集合 DOM；属性间数据的依赖关系集合 F。也可以说关系模式是一个五元组。关系模式一般表示为 R（U，D，DOM，F），通常简记为 R（U）或 R（A_1，A_2，…A_n），其中 U 为属性名集合，A_1，A_2，… A_n 为属性名。域名及属性向域的映像常常直接说明为属性类型和长度。

1.4.2.3　关系数据库

所有支持关系数据库模型的实体及实体之间的联系的关系集合就构成了一个关系数据库。

关系数据库有型与值之分，型称为关系数据库的模式，值称为关系数据库的值。关系数据库模式与关系数据库的值通常统称为关系数据库。

1.4.3　关系的完整性

关系模型完整性是为保证数据库中数据的正确性和相容性，关系模型的完整性规则是对关系的某种约束条件。关系的完整性分为三类：实体完整性、参照完整性和用户定义的完整性。

1. 实体完整性

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，关系 BOOK 中的 BookId 是主属性，不可以取空值。关系 BORROW 中的 BookId、CardId、Bdate 都是主属性，它们都不可以取空值。

2. 参照完整性

外码的定义：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 Ks 相对应，则称 F 是基本关系 R 的**外码**，并称基本关系 R 为参照关系，基本

关系 S 为被参照关系或目标关系。

注意：关系 R 和 S 不一定是不同的关系。

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 Ks 相对应，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。

我们来举例说明，假设有下列两个关系“职工”和“部门”，如表 1-6 和表 1-7 所示。

表 1-6 职工

职工姓名 Name	部门编号 DeptNo
刘勇	01
丁钰	02
张清蜂	

外码

表 1-7 部门

部门编号 DeptNo	部门名称 DeptName
01	计算机系
02	人事处
03	电子系

主码

对于上述例子中，职工关系中的 DeptNo 是外码，部门关系中的 DeptNo 是主码。根据参照完整性规则，职工关系中的 DeptNo 属性可以为空，表示暂时没有分配部门，如果不为空值，则其值必须在部门关系中已经存在。也就是说职工关系中的 DeptNo 要么为空值，要么在部门关系中已经存在的值，而不可以为其他值。根据实体完整性规则，部门关系中的 DeptNo 不可以为空值。

3. 用户定义完整性

在关系数据库系统中，用户可以对属性的取值或属性间关系加某种限制条件，这就是用户定义完整性。例如在工资关系中可以定义：应发工资-应扣工资=实发工资，以保正数据的完整性。又如，在上述的读者关系中可以定义：性别只能为男或女。

1.4.4 关系代数

关系代数是用关系的运算来表达查询方式的，它是关系数据操纵语言的一种传统表达方式。其特点是以一个或多个关系作为运算对象，结果为另外一个关系。

关系代数的运算符分为四类：集合运算符、专门的关系运算符、比较运算符、逻辑运算符。常用关系运算符如表 1-8 所示。

表 1-8 关系代数运算符

运算符		含义
集合运算符	∪	并
	—	差
	∩	交
	×	广义笛卡尔积
专门的关系运算符	σ	选择
	Π	投影

续表

运算符		含义
专门的关系运算符	⋈	连接
	÷	除
比较运算符	＞	大于
	≥	大于或等于
	＜	小于
	≤	小于或等于
	＝	等于
	≠	不等于
逻辑运算符	¬	非
	∧	与
	∨	或

关系代数的运算分为传统的集合运算和专门的关系运算。

1.4.4.1　传统的集合运算

传统的集合运算包括并（∪）、交（∩）、差（–）、笛卡尔积（×），它将关系看成元组的集合，从关系的水平方向（行）来进行的。

1. 并∪

设关系 R 和关系 S 具有相同的目，也就是说两个关系的属性个数相同，且相对应的属性取自同一个域，则关系 R 和关系 S 的并由属于 R 或 S 的元组合并而成。其结果关系仍为原来的属性个数，记为：

$$R \cup S = \{t \mid t \in R \vee t \in S\}$$

2. 差–

设关系 R 和关系 S 具有相同的目，且相对应的属性取自同一个域，则关系 R 和关系 S 的差由属于 R 而不属于 S 的元组组成。其结果关系仍为原来的属性个数，记为：

$$R - S = \{t \mid t \in R \wedge \neg t \in S\}$$

3. 交∩

设关系 R 和关系 S 具有相同的目，且相对应的属性取自同一个域（同类属性），则关系 R 和关系 S 的交由既属于 R 又属于 S 的元组组成。其结果关系仍为原来的属性个数，记为：

$$R \cap S = \{t \mid t \in R \wedge t \in S\}$$

4. 广义笛卡尔积×

设关系 R 有 n 目，关系 S 具有 m 目，则 R 和 S 的广义笛卡尔积是 n+m 目关系。元组前 n 列是 R 的一个元组，后 m 列是 S 的一个元组，元组的个数为 R 的元组个数×S 的元组个数。记为：

$$R \times S = \{\widehat{t_r\, t_s} \mid t_r \in R \wedge t_s \in S\}$$

传统的集合运算举例说明如下：设有关系 R 和 S，如表 1-9 所示。其中 A 和 D，B 和 E，C 和 F 都是来自同一个域。

表 1-9 传统集合运算举例

（a）R

A	B	C
a1	b1	c1
a2	b2	c2

（b）S

D	E	F
d1	e1	f1
d2	e2	f2
a1	b1	c1
d1	e1	f1

（c）R∩S

A	B	C
a1	b1	c1

（d）R∪S

A	B	C
a1	b1	c1
a2	b2	c2
d1	e1	f1
d2	e2	f2

（e）R–S

A	B	C
a2	b2	c2

（f）R×S

A	B	C	D	E	F
a1	b1	c1	d1	e1	f1
a1	b1	c1	d2	e2	f2
a1	b1	c1	a1	b1	c1
a2	b2	c2	d1	e1	f1
a2	b2	c2	d2	e2	f2
a2	b2	c2	a1	b1	c1

集合运算实现的数据库操作：

数据库记录的添加、插入----------并运算

删除----------差运算

数据库的修改（先删后插）--------差+并运算

关系的连接----------笛卡尔积

1.4.4.2 专门的关系运算

专门的关系运算包括选择、投影、连接、除等操作。

1. 选择（selection）

选择又称为限制，它是在关系 R 中选择满足给定条件的元组，组成一个新的关系。记作：

$$\sigma_F(R) = \{t \mid t \in R \wedge F(t) = \text{TRUE}\}$$

其中 F 表示选择条件，它是一个逻辑表达式，由属性名、逻辑运算符、比较运算符组成，属性名也可以用它的序号来代替。选择操作是从行的角度进行的运算。

例 1-1 在读者关系中查找男性读者

$$\sigma_{sex='男'}(\text{Reader}) \text{或者} \ \sigma_{3='男'}(\text{Reader})$$

结果如下表所示。

CardId	Name	Sex	Dept	Class
T0001	刘勇	男	计算机系	1
S0111	张清蜂	男	电子系	3

例 1-2　查找计算机系所有读者。

$$\sigma_{dept=\text{‘计算机系’}}(Reader)\text{或者 }\sigma_{4=\text{‘计算机系’}}(Reader)$$

结果如下。

CardId	Name	Sex	Dept	Class
T0001	刘勇	男	计算机系	1
T0002	张伟	女	计算机系	1

2. 投影

从关系 R 上选取若干属性列 A，并删除重复行，组成新的关系。记作：

$$\prod_A (R) = \{ t[A] \mid t \in R\}$$

投影操作是从列的角度进行的运算。

例 1-3　查询关系 BOOK 中所有图书的书名和对应的出版社。

$$\prod_{Bookname,Publish}(Book)$$

结果如下表所示。

Bookname	Publish
数据结构	清华大学出版社
数据结构	中国水利水电出版社
高等数学	中国水利水电出版社
数据库系统	人民邮电出版社
数据库原理与应用	中国水利水电出版社

例 1-4　查询“中国水利水电出版社”出版的所有藏书的书名和库存数量。

$$\prod_{Bookname,\ Qty}(\sigma_{Publish=\text{‘中国水利水电出版社’}}(Book))$$

结果如下表所示。

Bookname	Qty
数据结构	50
高等数学	60
数据库原理与应用	100

3. 连接（join）

连接也称为 θ 连接。它是从两个关系 R 和 S 的笛卡尔积 R×S 中选取属性间满足一定条件的元组，构成新的关系。记作：

$$R \underset{x\theta y}{\bowtie} S\{\widehat{t_r\ t_s} \mid t_r \in R \wedge t_s \in S \wedge X\theta Y\}$$

其中 X 和 Y 分别为 R 和 S 上度数相等且可比的属性组，θ 为比较运算符。当 θ 为“=”时，

称为等值连接，记作：

$$R \underset{x=y}{\bowtie} S=\{\widehat{t_r\ t_s} \mid t_r \in R \wedge t_s \in S \wedge X=Y\}$$

它是从关系 R 与 S 的笛卡尔积中选取 X、Y 属性值相等的那些元组。

自然连接是一种特殊的等值连接，它要求两个关系中进行比较的分量必须是相同的属性组，并且要在结果中把重复的属性去掉。自然连接记作：

$$R \bowtie S=\{\widehat{t_r\ t_s} \mid t_r \in R \wedge t_s \in S \wedge t_r[X]=t_s[X]\}$$

例 1-5 设关系 $R=\prod_{BookId,Bookname,Publish}(Book)$，如表 1-10 的（a）所示。

$S=\prod_{CardId,BookId,}(Borrow)$，如表 1-10（b）所示。

则 R 和 S 的等值连接(R.BookId=S.BookId)的结果如表 1-10（c）所示。

R 和 S 的自然连接的结果如表 1-10（d）所示。

表 1-10

（a）R 关系

BookId	Bookname	Publish
TP2001--001	数据结构	清华大学出版社
TP2003--002	数据结构	中国水利水电出版社
TP2002--001	高等数学	中国水利水电出版社
TP2003--001	数据库系统	人民邮电出版社
TP2004--005	数据库原理与应用	中国水利水电出版社

（b）S 关系

CardId	BookId
T0001	TP2003--002
S0101	TP2001--001
S0111	TP2003--001
S0101	TP2003--002

（c）R 和 S 的等值连接

R．BookId	BookName	Publish	S．BookId	CardId
TP2001--001	数据结构	清华大学出版社	TP2001--001	S0101
TP2003--002	数据结构	中国水利水电出版社	TP2003--002	T0001
TP2003--002	数据结构	中国水利水电出版社	TP2003--002	S0101
TP2003--001	数据库系统	人民邮电出版社	TP312--001	S0111

（d）R 和 S 的自然连接

BookId	BookName	Publish	CardId
TP2001--001	数据结构	清华大学出版社	S0101
TP2003--002	数据结构	中国水利水电出版社	T0001
TP2003--002	数据结构	中国水利水电出版社	S0101
TP2003--001	数据库系统	人民邮电出版社	S0111

4. 除（division）

为了说明除法运算，先得给出象集的概念。

象集的定义：给定一个关系 R（X，Z），X 和 Z 为属性组。定义当 t（X）= x 时，x 在 R 中的象集为：

$$Zx=\{t[Z]|t\in R,t[X]=x\}$$

它表示 R 中属性组 X 上值为 x 的诸元组在 Z 上分量的集合。

除运算定义：给定关系 R（X，Y）和 S（Y，Z），其中 X，Y，Z 为属性组。R 中的 Y 与 S 中的 Y 可以有不同的属性名，但必须取自相同的域集。R 与 S 的除运算得到一个新的关系 P（X），P 是 R 中满足下列条件的元组在 X 属性列上的投影：元组在 X 上分量值 x 的象集 Yx 包含 S 在 Y 上的投影集合。记作：

$$R\div S=\{t_r[X]\mid t_r\in R\wedge Yx\supseteq \Pi y(S)\}$$

其中 Yx 为 x 在 R 中的象集，$x=t_r[X]$。

除操作是同时从行和列角度进行运算的。

例 1-6　设关系 R，S 分别如表 1-11 中的（a）和（b）所示。求 R÷S。

表 1-11

（a）R

A	B	C
a1	b1	c1
a1	b2	c2
a1	b3	c3
a2	b2	c2
a3	b3	c3
a4	b4	c4

（b）S

B	C	D	E
b1	c1	d1	e2
b2	c2	d2	e2

（c）R÷S

A
c1

分析：关系 R 的属性可分为二个组：[X]={A}，[Y]={B，C}

关系 S 的属性可分为二个组：[Y]={B，C}，[Z]={D，E}

在关系 R 中，A 的值为{a1,a3,a4},其中：

a1 的象集为{(b1,c1),(b2,c2),(b3,c3)}。

a2 的象集为{(b2,c2)}。

a3 的象集为{(b3,c3)}。

a4 的象集为{(b4,c4)}。

S 在（B，C）上的投影为Πy(S)={(b1,c1),(b2,c2)}

从上面可以看到，只有 a1 的象集包含了 S 在（B，C）上的投影。

所以 R÷S={a1}

5. 综合举例

例 1-7　查询读者刘勇在 2004 年 4 月 8 号借书的书名。

$$\Pi_{bookname}(\sigma_{name=\text{“刘勇”}}(Reader)\bowtie \sigma_{bdate=\text{“2004.04.08”}}(Borrow)\bowtie Book)$$

例 1-8　查询至今有未还书的读者姓名。

$$\Pi name(Reader\bowtie \sigma_{Sdate=NULL}(Borrow))$$

例 1-9 查询既出版了“高等数学”又出版了“数据结构”的出版社。

先构建临时关系 BK

BOOKNAME
高等数学
数据结构

$\Pi_{\text{bookname, publish}}(\text{Book}) \div \text{Bk2}$

根据除法运算可以得出结果为{中国水利水电出版社}

1.4.5 关系数据库管理系统

关系数据库管理系统（RDBMS）是关系型的具体实现，是指支持关系模型的系统。

如果一个数据库管理系统支持关系数据结构（表结构），且支持选择、投影和连接运算，则可定义为关系数据库系统。依据支持关系模型的程度不同，可以将关系数据库分为如下几个等级：

（1）（最小）关系系统。即满足上面最基本的条件，支持关系数据结构，支持选择、投影和连接操作。这些产品的代表有 Foxbase，FoxPro。

（2）关系完备系统。支持关系数据结构和所有的关系代数操作。一般具有关系完备的数据子语言，在一定程度上实现了数据的独立性，确保用户能够依靠关系名、关键字值和属性名组合，用逻辑方式访问数据库中的每一个数据。目前流行的产品代表有 SQL Server、DB2、Oracle 等。

（3）全关系。这类系统支持关系模型的所有特征，而且支持数据结构中域的概念及实体完整性和参照完整性。1985 年 E.F.Codd 给全关系 DBMS 提出了严格的标准。依照这个标准，目前还没有一个数据库产品达到这个标准，也许新的全关系数据库产品正在研发当中。

习题一

一、选择题

1．（　）是（　）的具体表现形式，（　）是（　）有意义的表现。

A）信息、数据、数据、信息　　B）数据库、信息、信息、数据库

C）数据、信息、信息、数据　　D）数据、信息、数据库、信息

2．数据库管理系统的功能不包括（　）。

A）定义数据库　　B）对已定义的数据库进行管理

C）为定义的数据库提供操作系统　　D）数据通信

3．作为数据库管理系统（DBMS）功能的一部分，（　）被用来描述数据及其联系。

A）数据定义语言　　B）自含语言

C）数据操作语言　　D）过程化语言

4．常见的三种数据模型是（　）、（　）和（　）。

A）链状模型、关系模型、层次模型　　B）关系模型、环状模型、结构模型

C）层次模型、网状模型、关系模型　　D）链表模型、结构模型、网状模型

5．数据库系统的特点不包括（　　）。

A）数据共享　　B）加强了对数据安全性和完整性保护

C）完全没有数据冗余　　D）具有较高的数据独立性

6．数据操纵语言 DML 根据其实现方法可以分为（　　）和（　　）两大类。

A）自含型语言、宿主型语言　　B）自主型语言、高级语言

C）高级语言、宿主型语言　　D）高级语言、低级语言

7．（　　）是（　　）的特殊形式，（　　）是（　　）的一般形式。

A）关系模型、网状模型、网状模型、关系模型

B）层次模型、网状模型、网状模型、层次模型

C）链状模型、关系模型、关系模型、链状模型

D）环状模型、链状模型、网状模型、关系模型

8．关系模型中，一个关系就是一个（　　）。

A）一维数组　　B）一维表　　C）二维表　　D）三维表

9．在数据库系统中，对于现实世界“事物”术语是指（　　）。

A）实际存在的东西　　B）有生命的东西

C）独立存在的东西　　D）一切东西，甚至可以是概念性的东西

10．数据库的三个模式中，真正存储数据的是（　　）。

A）内模式　　B）模式

C）外模式　　D）三者皆存储数据

11．在数据库的三个模式中（　　）。

A）内模式只有一个，而模式和外模式可以有多个

B）模式只有一个，而内模式和外模式可以有多个

C）模式和内模式只有一个，而外模式可以有多个

D）均只有一个

12．关于模式，下列说法中错误的是（　　）。

A）数据库的全局逻辑结构描述　　B）数据库的框架

C）一组模式的集合　　D）数据库中的数据

二、填空题

1．一个完整的数据库系统应包括_____、_____、_____、_____和_____等五个部分。

2．数据的概念包括_____和_____两个方面。

3．DBMS 中数据定义语言的英文缩写是_____，数据操纵语言的英文缩写是_____。

4．在关系模型中，二维表中每一行的所有数据在关系中称为_____。

5．二维表中每一列的所有数据在关系中称为_____。

6．域是指不同元组中在同一属性的_____。

7．迄今为止，数据管理技术经历了_____、_____和_____发展阶段。

8．数据处理中的数据描述实际上经历了_____、_____、_____和三个世界的演

变过程。

9．数据库的三级组织模式分别称为_____、_____和_____。

10．数据库的三级组织模式结构是通过分别称为_____和_____两种映射以保证数据独立性。

三、解释如下名词的概念

1．关系数据库，码，候选码，外码，元组，属性，域。

2．实体完整性，参照完整性，自定义完整性。

3．等值连接，自然连接。

四、计算题

1．设有两个关系 R 和 S，如下表所示，请计算 R∪S，R∩U ，R–S，R×S。

R

A	B	C
a1	b1	c1
a2	b2	c2
a3	b2	c1

S

A	B	C
a1	b2	c2
a2	b2	c2
a3	b2	c1

2．设有如下两个关系 R 和 S，如下表所示，求 R 和 S 的等值连接（R.B=S.B）和自然连接。

R

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3

S

B	D
a1	D1
b2	d2
b4	d5

3．对照本章的表 1-3、表 1-4、表 1-5 三个关系，写出如下操作，并写出结果。

（1）查询人民邮电出版社出版的全部图书的书号和书名。

（2）查询借阅了书号为 TP2003--002 的读者姓名。

（3）查询借过中国水利水电出版社出版的全部图书的读者姓名。