

第 1 章 数据库系统概述

数据库 (Database) 是存储在一起的相关数据的集合, 它不仅包括数据本身, 而且包括关于数据之间的联系。在当今信息社会中, 数据库无时不与我们的日常工作、生活和学习相关, 如国家人口信息数据库, 国家地理信息数据库, 银行数据库, 电信数据库, 工业企业中产品信息数据库, 企业管理信息数据库, 学生档案数据库, 学生成绩数据库等等都与我们的日常工作、生活和学习有着直接或间接的联系, 起着越来越重要的作用。

数据库技术是研究数据库的结构、存储、设计和使用的—门软件学科。它产生于 20 世纪 60 年代中期。根据计算机应用的调查, 在计算机的三大主要应用领域 (科学计算、数据处理与过程控制) 中, 数据处理约占 70% 左右的比重。正因为如此, 为适应数据处理需求的不断增长, 满足各类信息系统对数据管理的要求, 随着数据库技术的发展, 数据管理在文件系统的基础上不断发展, 产生了目前普遍采用的数据库管理系统。现在, 数据库技术已成为计算机领域中最重要—的技术之一, 是软件学科的一个独立分支。数据库方法原本是针对事务处理中大量数据管理需求的, 但是它的应用范围不断扩大, 不仅应用于事务处理, 而且进一步应用到情报检索、人工智能、专家系统和计算机辅助设计等, 涉及到非数值计算各方面的应用。应用范围的扩大又进一步促进了数据库技术的深入发展, 可以说数据库系统已成为当代计算机系统的重要组成部分。

近半个世纪以来, 数据库在理论上和实现技术上均得到很大的发展, 出现了许多商用数据库管理系统, 使得计算机应用渗透到各行各业的各类管理工作中。例如, 管理信息系统、办公自动化系统和决策支持系统等都是使用了数据库管理系统或数据库技术的典型的计算机应用系统。

我国自从第一次全国人口普查使用了数据库技术以来, 数据库应用逐渐普及, 深受欢迎。随着我国国民经济与社会信息化工作的不断深入, 数据库将成为我们日常工作、学习和生活的重要组成部分。作为数据库应用开发人员, 只有了解数据库系统的基本原理, 掌握数据库设计的基础理论, 熟悉数据库管理系统特点, 才能应用和开发好数据库应用系统。

为了使读者建立数据库和数据库系统的相关概念, 并对数据库和数据库系统有初步了解, 本章首先介绍数据、信息和数据处理的概念, 回顾数据管理技术的发展过程, 然后介绍数据库、数据库系统和数据库管理系统的基本概念、基本组成和基本功能。

1.1 信息、数据与信息系统

1.1.1 信息、数据与数据处理

在计算机应用中, 数据处理和以数据处理为基础的信息系统所占的比重最大, 一个国家现代化水平越高, 科学管理、自动化服务的需求就越大。在数据处理这一领域中, 首先应

该掌握的基本概念是信息 (information) 和数据 (data)。

信息是目前许多学科广泛使用的概念。人类的一切活动都离不开数据，离不开信息。在不同的领域中，其含义有所不同。但普遍承认信息是与能源、材料科学并列地构成现代社会的三大支柱之一。在数据处理领域，通常把信息理解为关于现实世界事物的存在方式或运动状态的反映的组合。例如，一所大学里的一名学生，他的学号 050201024，姓名是关雨，身高是 1.78 米，民族是汉族，专业是机械制造及其自动化，这些都是关于 050201024 号学生的信息，都是关于这位学生的存在状态的反映，从不同角度“反映”或“刻画”了这位学生的基本状况。

信息有如下一些重要特征：

(1) 信息源于物质和能量。一切事物，包括自然界和人类都产生信息。信息是物质和能量形态的反应，它不可能脱离物质而存在。信息传递需要物质载体，信息的获取和传递要消耗能量。

(2) 信息是可以感知的。人类对客观事物的感知，可以通过感觉器官，也可以通过各种仪器仪表和传感器等，不同的信息源有不同的感知方式。

(3) 信息是可存储的。动物用大脑存储信息，叫做记忆。文字的发明，提供了长期存储信息的可靠手段。近年来，计算机存储器、录音、录像等技术的发展，进一步扩大了信息存储的范围。

(4) 信息是可以加工、传递和再生的。电子计算机是信息处理领域中最先进的工具之一，人类对收集到的信息可以进行取舍整理。从客观世界收集到或加工后的信息，可以通过各种形式传递，例如语言、表情、动作、文字、电报、电话、电视、计算机通信和卫星通信等。

上述这些特点，构成了“信息”的最重要的自然属性。作为信息的社会属性，信息已经成为社会上各行各业不可缺少的重要资源之一。人类获取、积累并利用信息是认识和改造客观世界的必要过程。借助信息，人类才能获得知识，才能有效地组织各种社会活动。因此，信息是人类维持正常活动的不可缺少的资源。

几乎和信息同样广泛使用的另一个概念是“数据”。所谓数据，通常指用符号记录下来的可加以鉴别的信息。数据的概念包括两个方面：其一，数据内容是事物特性的反映或描述；其二 数据是符号的集合。例如 为了描述一位学生的信息 我们可以用一组数据‘ 050201024、关 雨、1.78、汉族、机械工程及自动化 ’ 来表示，数据“ 050201024”、“ 关 雨 ”、“ 汉 族 ”、“ 机 械 制 造 及 其 自 动 化 ” 分别描述了这位学生的学号、姓名、民族、所学专业，他们分别是数字和汉字字符的集合。这里，这些符号已经被人们赋予了特定的语义，所以，它们就具有了传递信息的功能。

从这个例子中，我们可以看到信息和数据之间的固有联系：数据是信息的符号表示或称为载体，信息则是数据的内涵，是对数据的语义解释。

但另一方面，某一具体的信息与表示它的数据的这种对应关系又因环境而异。同一信息可能有不同的符号表示，同一数据也可能有不同的解释。

数据处理领域中的数据概念较之科学计算领域中的数据概念已经大大地拓宽了。定义中所说的符号，不仅包含数字符号，而且包含文字、图像和其他特殊符号；而所谓“记录下来”也不仅是指用笔写在纸上，它还包括磁记录、纸带穿孔、光刻等各种记录形式。近年来

在国际上出现的关于多媒体数据库技术的研究，正是这种数据概念拓宽的自然结果。

由于信息是现实世界中事物的存在方式和运动状态的反映，而现实世界的事物常常是相互关联的，这就使得人们在了解、掌握事物之间的固有联系和运动规律的基础上，可以从一些已知的信息出发，经过演绎推理，导出新的信息，为人类社会生活的各种需要服务，这就是我们常说的“信息处理”。例如，学号“050201024”，前两位（“05”）表示所在学院，第三、四位（“02”）表示入学年份，第五、六位（“01”）表示专业编号，最后三位（“024”）表示顺序号，这样，根据学生的学号，我们可以推算出学生所在学院、入学年份、所学专业。

数据能表示信息，但并非任何数据都能表示信息，正如人们常说的“如果计算机输入的是垃圾，输出的也会是垃圾”。同一数据也可能有不同的解释。由此可见，信息只是人们消化理解了的数据。信息是抽象的，不随数据设备所决定的数据形式而改变；数据的表示方式具有可选择性。

一般，人们将原始信息（源信息）表示成源数据，然后对这些源数据进行综合推导加工，得出新的数据。这些结果数据表示了新的信息，可以作为某种决策的依据（或用于新的推导加工）这个过程通常称为“数据处理”。图 1-1 描述了信息处理的基本流程。实际上，对数据处理存在广义和狭义两种定义。广义地讲，数据处理包括对数据的收集、表示、存储、加工、分类、检索、传播等一系列活动。狭义地讲，它是指对所输入的数据进行加工整理。数据处理的基本目的是从大量的、已知的数据出发，根据事物之间的固有联系和运动规律，通过分析归纳、演绎推导等手段，萃取出对人们有价值、有意义的信息，作为决策的依据。由此可见，信息是一种被加工成特定形式的的结果，这种数据形式对于数据接收者来说是有意义的。对数据的加工可以相对比较简单也可以相当复杂。简单加工包括组织、编码、分类、排序等；复杂加工可以复杂到使用统计学方法、数学模型等对数据进行深层次的加工。

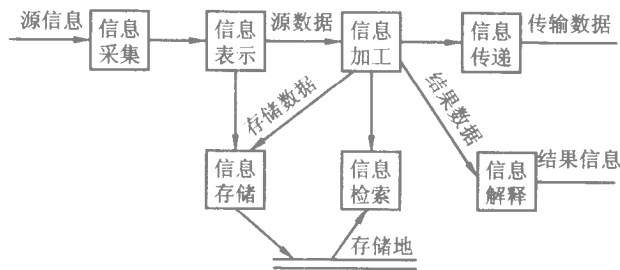


图 1-1 信息处理活动流程示意图

电子计算机使大规模的数据处理成为可能，它和通信技术的发展一起，进一步推动了信息处理和利用的社会化，极大地增强了人类社会信息处理的能力。今天，除了人的思维活动之外，大量信息处理活动都采用将信息表示为数据，然后对这类符号进行演算和推理的方式。因此，在大多数场合下，我们将不再区分信息处理和数据处理这两个概念。

计算机数据处理的方式，有批处理（Batch-Processing）和联机系统（On-Line Systems）两种。

在批处理方式下，用户将程序和数据组成作业（可能包含若干事务）提交给计算机系统，计算机把一批作业汇集起来统一处理，输出结果或报表。在计算机处理期间，人工不能进行干预。这种方式，计算机利用率高，但缺点是结果输出不及时，如果程序中有错误，或

发生了某些意外情况，用户需要中止程序的执行是不可能的，必须在该批作业处理完毕后才能干预，这往往会延误工作

联机方式是每当输入一个事务后立即进行处理，源数据通过终端直接进入计算机，在计算机处理过程中，一旦发现错误，立即通过终端提示用户，用户也可随时中止程序的执行。因此，联机方式不需要对作业进行额外的组织工作，可以立即修正错误，缩短处理周期。但由于输出和对话往往采用低速设备，耗费时间，使 CPU 使用效率降低；另一方面，为了适应资源统一管理及各用户并发地使用，使系统相对复杂化，这些都是联机方式的不足之处。

1.1.2 信息系统及其分类

信息已经受到社会的广泛重视，被看作社会和科学技术发展的三大支柱之一（材料、能源、信息）。在一个组织里，信息已作为一种资源，占有重要地位。当然，信息是一种可以管理和控制的，有别于基本资源（指人力、物力、财力、时间）的新形式的资源。当今社会已进入信息时代，管理人员在工作中多半时间花费在记录、检索和获取信息上。

1. 信息系统

信息系统是指为了某些明确的目的而建立的，由人员、设备、程序和数据集合构成的统一整体。信息系统的主要功能是提供信息，以支持一个组织机构的运行、管理和决策。更确切地说，信息系统将不适用的数据形式加工成可利用的形式。

一个信息系统的质量取决于它是否能及时地为用户提供所需要的信息。在一个组织机构中，不同阶层的管理人员因其管理的目标不同，所需要信息亦不相同。信息系统针对各个层次的需求，通过计算机实现信息支持，达到辅助管理的目的。

2. 信息系统分类

信息系统可分为三类：数据处理系统，管理信息系统和决策支持系统。

数据处理系统：用计算机代替繁杂的手工处理事务工作，其目的是提高数据处理的准确性、及时性，节约人力，提高工作效率。例如，计算机运行会计核算软件，对会计的“簿记”事务进行常规处理、提供数据查询、会计报表等功能，使会计部门的日常工作自动化。

管理信息系统：由若干子系统构成的一个集成的人机系统，从组织的全局出发，实现数据共享，提供分析、计划、预测、控制等方面的综合信息。其主要目的是发挥系统的综合效益，提高管理水平。

例如，某企业管理信息系统由技术管理子系统、人事管理子系统、财务管理子系统、物资管理子系统、生产管理子系统、设备管理子系统和销售管理子系统组成。实现计算机管理能够迅速、准确地提供有关信息，不仅有力地支持各个职能部门的组织管理，并且通过信息共享加强了各子系统之间的协同，使整个系统有机地联系起来，同时为企业领导制定计划、确定经营目标、指挥生产提供信息支持，从而大大提高企业的综合效益，增强市场竞争能力。

应该指出，管理信息系统的结构和组织机构的影响是可逆的。现存的组织结构可以影响管理信息系统的设计，管理信息系统也可以影响未来组织机构的调整。由于计算机的引进使工作和管理环境为之一新，原来行之有效的制度、方法和监督机制在新的环境可能不再适用了，必须依据新环境的特点制定新制度、采用新方法，以至实施新的业务流程。

决策支持系统：为决策过程提供有效的信息和辅助决策手段的人机系统。其主要目的

是帮助决策者提高决策的科学性及有效性。

计算机辅助决策必须积累大量的数据、案例、方法和模型，更进一步还可以利用知识库系统和专家系统。决策支持系统的服务对象是面向某种决策问题的管理人员，它协助决策者在求解问题的过程中方便地检索出相关数据，对多种可选方案进行测试比较，然后作出决定。

这里需要强调指出，决策支持系统只能对决策提供支持，并不能由计算机代替人自动地作出决定，人是决策行动的主体。例如，不同的管理人员运行同一套决策支持系统软件，结果可能作出不同的决策来。

1.2 计算机数据管理的发展

通过上一节对信息系统的简单介绍可以了解到，各类信息系统都需要处理大量的数据。在这些信息系统中数据处理的中心问题是数据管理。数据管理是指对数据的组织、分类、编码、存储、检索和维护等工作。

数据处理工作由来已久，早在 1880 年美国进行人口统计，就采用了穿孔卡片存储信息，称为机器卡片文件。这样用机械方法进行数据处理，称之为机械的数据处理系统。后来由于电子计算机的出现，在数据处理中采用计算机，数据处理就进入了电子数据处理系统时代。随着计算机技术的发展，由于采用了磁鼓、磁心、磁带、半导体存储器和磁盘等存储技术，数据管理也得到了迅速发展。它主要围绕着提高数据独立性、降低数据冗余度、数据共享、提高数据的安全性和完整性等方面来进行改进，让使用者能方便地运用这些数据资源和管理这些资源。

数据在物理存储设备上的组织形式称之为数据的物理结构，一般以文件形式组织。根据其组织结构的特点可分为顺序文件、索引文件和随机文件等。数据在使用者面前呈现的组织形式，称之为数据的逻辑结构。对于数据的逻辑结构，可以用不同的物理结构来实现。物理结构的好坏影响系统的性能和效率，所以一种数据的逻辑结构，在运行阶段中，由于性能要求或存储数据的设备更新，就引起数据的物理结构的改变，这种改变称之为数据的再组织。用户（User）在编制应用程序时，是根据数据的逻辑结构对数据进行操作。应用程序对数据的物理结构的依赖程度称之为物理数据的独立性，依赖程度低，则物理数据的独立性越高。如果在数据处理环境中，数据具有物理独立性，则在数据再组织时不会影响原来应用程序的执行。

与任何其他技术的发展一样，计算机数据管理也经历了由低级到高级的发展过程。计算机数据管理随着计算机硬件（主要是外存储器）、软件技术和计算机应用范围的发展而不断发展。根据数据的独立性、数据的冗余度、数据间相互联系及数据的安全性、完整性等特点，数据管理方法可划分成四个阶段：人工管理阶段，文件系统阶段，数据库系统阶段，分布式数据库系统阶段。

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。当时在硬件方面，外存储器只有卡片、纸带、磁带，没有像磁盘这样的可以随机访问、直接存取的外部存储设备。软件方面，没有专门管理数据的软件，数据由计算或处理它的程序自行携带。数据处理方式基本是批处理。数据与应用程序之间的关系如图 1-2 所示。这一时期数据管理的特点是：

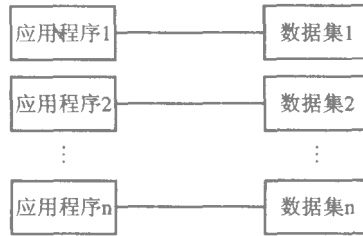


图 1-2 人工管理数据与程序的关系

(1) 数据不保存。因为计算机主要用十科学计算，对于数据保存的需求尚不迫切。一般的做法是，在进行某一课题计算时将原始数据随程序一起输入主存，运算处理后将结果数据输出。随着计算任务的完成，用户作业退出计算机系统，数据空间随同程序空间一起被释放。

(2) 没有专用的软件对数据进行管理。每个应用程序都要包括存储结构、存取方法和输入输出方式等，这些完全由应用程序员编程实现，从而要求应用程序员必须对各种数据存储结构的性能、实现和维护有较深的了解。程序中的存取子程序随着存储结构的改变而改变，因而数据与程序不具有独立性，即一组数据对应一组程序。这就使得程序依赖于数据，如果数据的类型、格式或者数据量、存取方法、输入输出方式等改变了，程序必须作相应的修改。此外，由于程序是直接面向存储结构的，因此不存在逻辑结构与物理结构的区别。

(3) 只有程序 (Program) 的概念，没有文件 (File) 的概念。即使有文件，也大多是顺序文件，其他组织方式必须由程序员自行设计与安排。

(4) 数据面向于应用。即一组数据对应于一个程序。由于各应用程序处理的数据不会毫无联系，程序之间可能有重复数据。

(5) 对数据的存取以记录为单位，灵活性差。

2. 文件系统阶段

20 世纪 50 年代后期至 60 年代中后期，计算机开始大量地用于管理中的数据处理工作。大量的数据存储、检索和维护成为紧迫的需求。在硬件方面，可直接存取的磁鼓、磁盘成为联机的主要外存。在软件方面，出现了高级语言和操作系统。操作系统中的文件系统 (有的也称为信息处理模块) 是专门管理外存的数据管理软件。数据处理方式有批处理，也有联机实时处理。这一阶段数据管理的特点是：

(1) 数据可长期保存在外存的磁盘上。用户经常随时通过程序对文件进行查询、修改和增删等处理。由于计算机应用从计算转向管理，数据处理的工作量增大。

(2) 数据的物理结构与逻辑结构有了区别，但较简单。程序与数据之间有设备独立性 (程序只需用文件名与数据打交道，不必关心数据的物理位置)，由操作系统的文件系统提供存取方法 (读 / 写)

(3) 文件的形式已多样化。有索引文件、链接文件和直接存取文件等，因而对文件的记录可顺序访问，也可随机访问。但文件之间是独立的，联系要通过程序去构造，文件的共享性差。

(4) 有了存储文件以后，数据不再仅仅属于某个特定的程序，而可以重复使用。但文件结构的设计仍然是基于特定的用途，程序仍然是基于特定的物理结构和存取方法编制的，

因此，数据结构与程序之间的依赖关系并未根本改变。

(5) 对数据的存取基本上还是以记录为单位。

文件系统的上述特点可用图 1-3 表示。

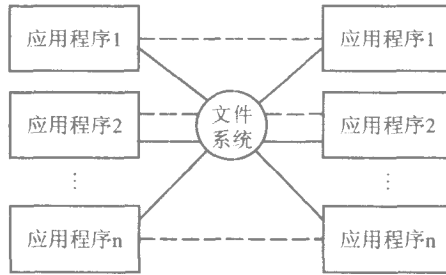


图 1-3 文件系统数据与程序的关系

在文件系统中，改变存储设备，不必改变应用程序。虽然文件系统提供了存取方法，但这只是初级的数据管理。这种文件系统，还未能彻底体现用户观点下的数据逻辑结构独立于数据在外存的物理结构要求。因此，数据的物理结构修改时，仍然需要修改用户的应用程序。

文件系统阶段是数据管理技术发展中的一个重要阶段。在这一阶段中，得到充分发展的各种数据结构和算法大大丰富了计算机科学，今天的数据库技术也正是在文件系统的基础上发展起来的。

但是，现在看来文件系统有三大缺陷：

(1) 数据冗余度大。数据冗余是指不必要的重复存储，同一数据项重复出现在多个文件中。在文件系统下，数据文件基本上与各自的应用程序相对应，数据不能以记录和数据项为单位共享。即使有部分数据相同，只要逻辑结构不同，用户必须各自建立自己的文件，这不仅浪费存储空间，增加更新开销。更严重的是，由于不能统一修改，容易造成数据的不一致性。

(2) 缺乏数据独立性。文件系统中的数据文件是为了满足特定业务领域某部门的专门需要而设计的，服务于某一特定应用程序。数据和程序相互依赖，如果改变数据的逻辑结构或文件的组织方法，必须修改相应的应用程序。同样道理，如果修改应用程序，如改用另一种程序设计语言来编写程序，也将影响数据文件的结构。

(3) 数据无集中管理。除了对记录的存取由文件系统承担以外，文件没有统一的管理机制，其安全性与完整性无法保障。数据的维护任务仍然由应用程序来承担。

文件是无弹性、无结构的数据集合。所谓无弹性是指由于记录的内部结构由应用程序自己定义，而不是由系统来统一管理，因此对现有数据文件的应用不易扩充，不易移植，也难以增删数据项以适用新的应用要求。无结构是指各个数据文件之间是独立的，缺乏联系，不能反映现实世界事物之间的联系。

这些问题阻碍了数据处理技术的发展，不能满足日益增长的信息需求，这是数据库技术产生的原动力，也是数据库系统产生的背景。应用需求和计算机技术的发展促使人们研究一种新的数据管理技术——数据库技术。

3. 数据库系统阶段

从 20 世纪 60 年代中后期开始，计算机应用于管理的规模更加庞大，需要计算机管理的

数据量急剧增长，并且对数据共享的需求日益增强。大容量磁盘（数百兆字节以上）系统的采用，使计算机联机存取大量数据成为可能；软件价格上升，硬件价格相对下降，使独立开发系统维护软件的成本增加。文件系统的管理方法已无法适应开发应用系统的需要。为解决数据的独立性问题，实现数据的统一管理，达到数据共享的目的，发展了数据库技术。

数据库（Database）是通用化的相关数据集合，它不仅包括数据本身，而且包括关于数据之间的联系。数据库中的数据不只面向某一项特定应用，而是面向多种应用，可以被多个用户、多个应用程序共享。例如，某个企业、组织或行业所涉及的全部数据的汇集。其数据结构独立于使用数据的程序，对于数据的增加、删除、修改和检索由系统进行统一的控制，而且数据模型也有利于将来应用的扩展。

为了让多种应用程序并发地使用数据库中具有最小冗余度的共享数据，必须使数据与程序具有较高的独立性。这就需要有一个软件系统对数据实行专门管理，提供安全性和完整性等统一控制机制，方便用户以交互命令或程序方式对数据库进行操作。

为数据库的建立、使用和维护而配置的软件称为数据库管理系统 DBMS（Database Management System），它是在操作系统支持下运行的。目前流行的数据库管理系统包括 Oracle, Informix, Sybase 等，主要应用在微机上的数据库管理系统有 DBASE, FoxBASE, FoxPro, FoxPro for Windows, ACCESS, CLIPPER, WINBASE 等。

现在，数据库已成为各类信息系统的核心基础。在数据库管理系统支持下数据与程序的关系如图 1-4 所示。

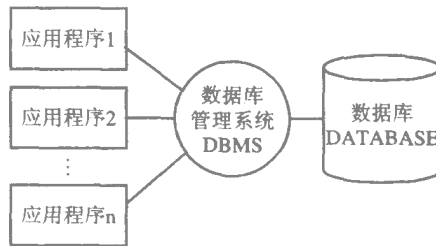


图 1-4 数据库系统数据与程序的关系

数据库系统的主要特点是：

(1) 实现数据共享，减少数据冗余。在数据库系统中，对数据的定义和描述已经从应用程序中分离开来，通过数据库管理系统来统一管理。数据的最小访问单位是数据项，既可以按数据项的名称存取库中某一个或某一组数据项，也可以存取一条记录或一组记录。

建立数据库时，应当以面向全局的观点组织库中的数据，而不能像文件系统那样仅仅考虑某一部门的局部应用。数据库中存放全组织（如企业）通用化的综合性的数据，某一类应用通常仅使用总体数据的子集，这样才能发挥数据共享的优势。

(2) 采用特定的数据模型。整个组织的数据不是一盘散沙，必须表示出数据之间所存在有机的关联，才能反映现实世界事物之间的联系。也就是说，数据库中的数据是有结构的，这种结构由数据模型表示出来。

文件系统只表示记录内部的联系，类似于属性之间的联系，而不涉及不同文件记录之间的联系。要想在不同文件中查询相关的数据，必须编写一个程序。

例 有三个文件，图书（总编号、分类号、书名、作者、出版单位、单价）；读者（借

书证号、姓名、性别、单位、职称、地址)；借阅(借书证号、总编号、借阅日期、备注)，要想查找某人所借图书的书名、出版社及借阅者的职称，则必须编写一段并不简单的程序来实现。

数据库系统不仅表示属性之间的联系，而且表示实体之间的联系。只要定义好数据模型，上述询问可以非常容易地联机查到。关于数据模型将在第 2 章中详细介绍。

(3) 具有较高的数据独立性。在数据库系统中，DBMS 提供映像的功能，确保应用程序对数据结构和存取方法有较高的独立性。数据的物理存储结构与用户看到的逻辑结构可以有很大差别。用户只以简单的逻辑结构来操作数据，无需考虑数据在存储器上的物理位置与结构。

(4) 有统一的数据控制功能。数据库作为多个用户和应用程序的共享资源，对数据的存取往往是并发的，即多个用户同时使用同一个数据库。数据库管理系统提供如下四方面的控制功能(将在第 6 章中详细介绍)：

1) 数据库并发控制功能：避免并发程序之间的相互干扰，防止数据库被破坏，杜绝提供给用户不正确的数据。

2) 数据库的安全性控制功能：保证数据的安全和机密，防止数据丢失或被窃取。

3) 数据库的完整性控制功能：保证数据库始终包含正确的数据。用户可设计一些完整性规则以确保数据值的正确性。例如，可把数据值限制在某个范围内，并对数据值之间的联系进行各种检验。

4) 数据库的恢复功能：在数据库被破坏时或数据不可靠时，系统有能力把数据库恢复到最近某个时刻的正确状态。

综上所述，数据库是一个存储起来为某个特定组织的多种应用服务并具有尽可能小的冗余度的互相关联的数据集合。其数据结构独立于使用数据的程序，对数据的增添、删除、修改和检索由系统进行统一的控制，而且数据模型也有利于将来应用的扩展。

从文件系统发展到数据库技术是信息处理领域的一个重大变化。在文件系统阶段，信息处理的传统观点如图 1-5a 所示，人们关注的中心问题是系统功能的设计，因此程序设计处于主导地位，数据只起着服从程序设计需要的作用。而在数据库方式下，信息处理的传统观念已为图 1-5b 所示的新体系所取代，数据开始占据了中心位置。数据的结构设计成为信息系统首先关心的问题，而利用这些数据的应用程序设计则退居到以既定的数据结构为基础的外围地位。

目前数据库系统的应用已经深入到人类社会生活的各个领域，从企业管理、银行业务、资源分配、经济预测一直到情报检索、档案管理、普查、统计等，并在通信网络基础上，建立了许多国际性的联机检索系统(如科技文献的检索系统)

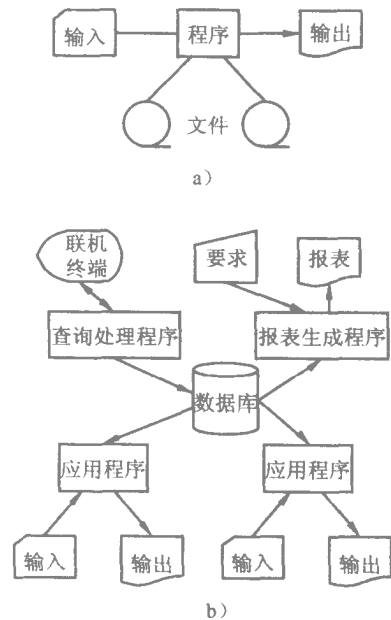


图 1-5 信息处理方式的演变

a) 传统方式 b) 数据库方式

4. 高级数据库系统阶段

这一阶段的主要标志是分布式数据库系统、面向对象的数据库系统的出现。

(1) 分布式数据库系统 (Distributed Database Systems)。20 世纪 70 年代后期之前, 数据库系统多数是集中式的。分布式数据库系统是数据库技术和计算机网络技术相结合的产物, 在 20 世纪 80 年代中期已有商品化产品问世, 如 ORACLE、SYBASE 等。分布式数据库是一个逻辑上统一、地域上分布的数据集合, 是计算机网络环境中各个结点局部数据库的逻辑集合, 同时受分布式数据库管理系统的控制和管理, 如图 1-6 所示。

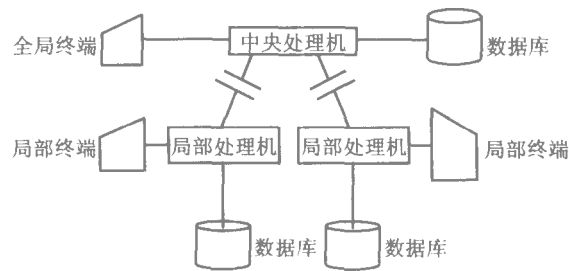


图 1-6 分布式数据库系统

分布式数据库在逻辑上像一个集中式数据库系统, 实际上数据存储在处于不同地点的计算机网络的各个结点上。每个结点有自己的局部数据库管理系统, 它有很高的独立性。用户可以由分布式数据库管理系统 (网络数据库管理系统) 通过网络通信相互传输数据。分布式数据库系统有高度透明性, 每台计算机上的用户并不需要了解他所访问的数据究竟在什么地方, 就像在使用集中式数据库一样。其主要特点有:

1) 局部自主: 网络上每个结点的数据库系统都具有独立处理本地事务的能力 (大量的), 而且各局部结点之间也能够互相访问、有效地配合处理更复杂的事务。因此, 分布式数据库系统特别适合各个部门的地理位置分散的组织机构, 例如银行业务、飞机订票、企业管理等。

2) 可靠性和可用性: 分布式系统比集中式系统有更高的可靠性, 在个别结点或个别通信链路发生故障的情况下可以继续工作。一个局部系统发生故障不至于导致整个系统停顿或破坏, 只要有一个结点上的数据备份可用, 则数据是可用的。可见, 支持一定程度的数据冗余是充分发挥分布式数据库系统优点的先决条件之一。

3) 效率和灵活性: 分布式系统分散了工作负荷, 缓解了单机容量的压力。数据可以存储在临近的常用结点上, 如果本结点的数据子集包含了要查询的全部内容, 显然比集中式数据库在全集上查找节省时间。

4) 系统易于实现扩展: 例如, 一个单位要增加新的机构, 分布式数据库系统能够在对现有系统影响较小的情况下实现扩充。由此, 扩大系统规模比集中式系统更加方便、经济、灵活。

(2) 面向对象的数据库系统。进入 20 世纪 80 年代, 在数据库领域, 虽然关系数据库的使用已相当普遍, 但在新的应用领域 (例如, 现代集成制造系统 (CIMS)、计算机辅助软件工程 (CASE)、办公自动化系统 (OA)、地理信息系统 (GIS)、知识库系统 (KBS)) 内显得力不从心。为了支持新的应用, 人们一般采用的方法是, 在传统 DBMS (主要是关系

型 DBMS) 的基础上针对不同应用进行不同功能扩充, 以满足应用需要, 但结果常常不能令人满意。在面向对象程序设计方法的影响下, 把面向对象的技术与数据库技术相结合, 出现了面向对象数据库系统 (Object-Oriented Database System, 简称 OODBS)。面向对象的数据库系统在逻辑上和物理上都从面向记录上升为面向具有复杂结构的对象, 它允许以自然方法并结合数据抽象机制在结构和行为上对复杂对象建立模型, 从而大幅度地提高了管理效率, 降低了用户使用复杂性, 为版本管理、动态模式修改等功能的实现创造了条件。面向对象的数据库系统主要特点如下:

1) OODBS 中基本结构是对象而不是记录。在 OODBS 中, 一切事物、概念等都是对象。一个对象可以由多个对象复合而成, 一个对象不仅包括描述它的数据, 而且还包括对它操作的方法, 因而能对现实世界的实体进行自然而直接的模拟, 这是传统数据库系统所不能比拟的。在传统数据库系统中, 记录与实体不一定相对应, 一个记录中可以包括几个实体的信息, 一个实体型也可以划分成多个记录类型存储, 记录结构带有随意性, 不一定反映人们对现实世界的认识。

2) 传统数据库系统只提供简单的数据类型, 如整数、实数及字符串等, 而在 OODBS 中, 用户可以通过构造一个对象类定义自己需要的数据类型。一个对象类定义中, 不仅定义了数据结构, 还定义了其上的操作, 体现了抽象数据类型的特点。因此, 与传统数据库系统相比, OODBS 具有容纳不同类型数据对象的能力, 具有对不同类型对象进行管理的机制。

3) OODBS 比传统数据库系统具有更多的语义。虽然关系模型中键的概念, 层次模型中的父子关系及网状模型中的系统都提供了一定的语义信息, 但总的来说, 语义比较贫乏。记录中各个属性的关系和作用是不明确的, 数据的语义要由用户来掌握。在数据模型中, 属性类、子类、超类的概念, 可以直接表达现实世界的对象与联系, 可表示诸如某个对象由“哪些对象组成”、有“什么性质”、处在“什么状态”、哪些对象类具有共同的特征等等, 具有丰富的语义信息。

4) OODBS 允许用户扩充数据类型。新数据类型可定义为原有类型的子类或超类, 系统扩充通过复用已有模式中的信息进行。

此外, OODBS 还支持版本演化、模式演变等, 这些都是传统数据库系统所不具备的。

1.3 数据库的体系结构

1. 数据库的三级体系结构

数据库的体系结构分成三级, 也称为三级模式: 内部级 (Internal)、概念级 (Conceptual) 和外部级 (External)。图 1-7 所示的结构是 1975 年 2 月美国 ANSI / X3 / SPARC (国家标准化委员会 / 关于计算机和信息处理 / 系统设计和要求委员会) 提出的。虽然现在数据库管理系统的产品多种多样, 在不同的操作系统支持下工作, 但是绝大多数系统在总的数据体系结构上仍具有三级结构的特征。

从某个角度看到的数据特性称为“数据视图”。外部级最接近用户, 是单个用户所能看到的数据特性。单个用户使用的数据视图称为“外模型”。概念级是涉及到所有用户的数据的定义, 也就是全局的数据视图, 称为“概念模型”。内部级最接近于物理存储设备, 涉及到实际数据存储的方式。物理存储的数据视图称为“内模型”。这些模型有两种表达方式,

一种是用图示形式表示的数据结构图（第 2 章将详细介绍）；另一种是用数据定义语言描述。这三种模型用数据定义语言描述分别得到外模式（亦称为子模式）、概念模式（亦称为模式）和内模式（亦称为存储模式）。

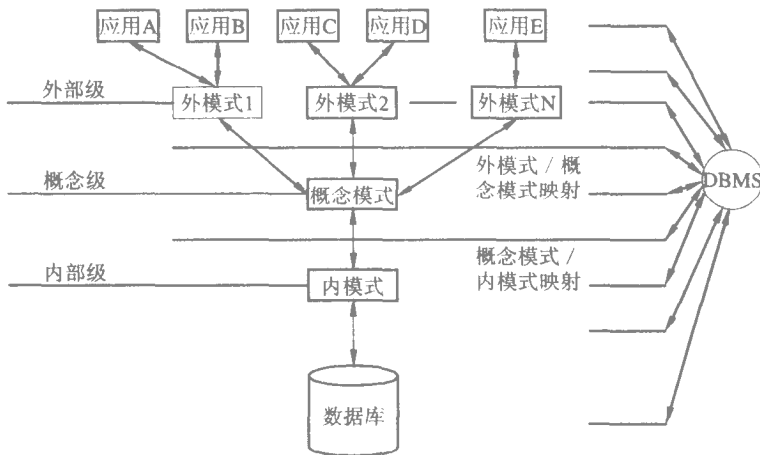


图 1-7 数据库系统体系结构

需要指出的是：这三层都不表示数据的实际存在，而仅表示了数据的抽象方法，即定义了数据库中每一个记录的类型（称为记录类型）。数据实际存放在物理设备上。不同层次的数据视图表示称为模式（Schema），因此，这里模式是一个非常重要的概念，通常意义上讲，模式是一个统称，是数据库每一层的数据结构的描述，包括外模式（亦称为子模式）、概念模式（亦称为模式）和内模式（亦称为存储模式）。

数据库的三级体系结构是数据的三个抽象级别，它把数据的具体组织留给数据库管理系统管理，使用户能抽象地处理数据，而不必关心数据在计算机中的表示和存储。这三级结构之间往往差别很大，为实现这三个抽象级别的转换，数据库管理系统在这三级结构之间提供了两层映像（Mappings）：外模式/模式映像，模式/内模式映像。

数据库的三级结构到底是如何描述的呢？不妨先看一个简单的例子。如图 1-8 所示，假设有一单位的职员数据库表 STAFF，它有两个用户 U1 和 U2，U1 负责处理有关人事的数据，U2 负责处理电话和电子邮件的数据，两个用户各自根据需从表中存取相关字段的数据，但不是表中的全部字段的数据。作为数据库系统，其目的是维护和管理 STAFF 表，并提供对两个用户（U1 和 U2）的服务。根据假设，这个数据库的体系结构有一个内模式，一个概念模式，两个外模式。在概念级，数据库有一个记录类型 STAFF，其概念模式用数据库定义语言描述为：

```
CREATE TABLE STAFF (( STAFF_ID CHAR(6), NOT NULL)
                     STAFF_NAME VARCHAR2 (12),
                     SEX NUMBER (1),
                     BIRTHDAY DATE,
                     SALARY NUMBER (5),
                     TELEPHONE VARCHAR2 (15),
                     EMAIL VARCHAR2(50));
```

它有七个字段，标明了相应的字段名、数据类型和长度，其中 STAFF_ID 为非空字段。对于内部级（内模式，亦称为存储模式），用数据库定义语言描述为：

```
CREATE INDEX STAFF# ON STAFF (STAFF_ID)
```

它描述了在 STAFF 模式的职员号上建立索引。在外部级，一个外模式可以用数据定义语言描述为：对于 U1 的应用程序：

```
CREATE VIEW PERSONAL AS
  SELECT STAFF_ID, STAFF_NAME, SEX, BIRTHDAY, SALARY
  FROM STAFF
```

对于 U2 的应用程序：

```
CREATE VIEW ADDRESS AS
  SELECT STAFF_ID, STAFF_NAME, TELEPHONE, EMAIL
  FROM STAFF
```

外模式中用到的记录名、字段名可以与概念级不同（一般也可以用相同的命名）。在这三级描述中还应该说明这些命名的变化（详细说明见第 3 章相关内容）。

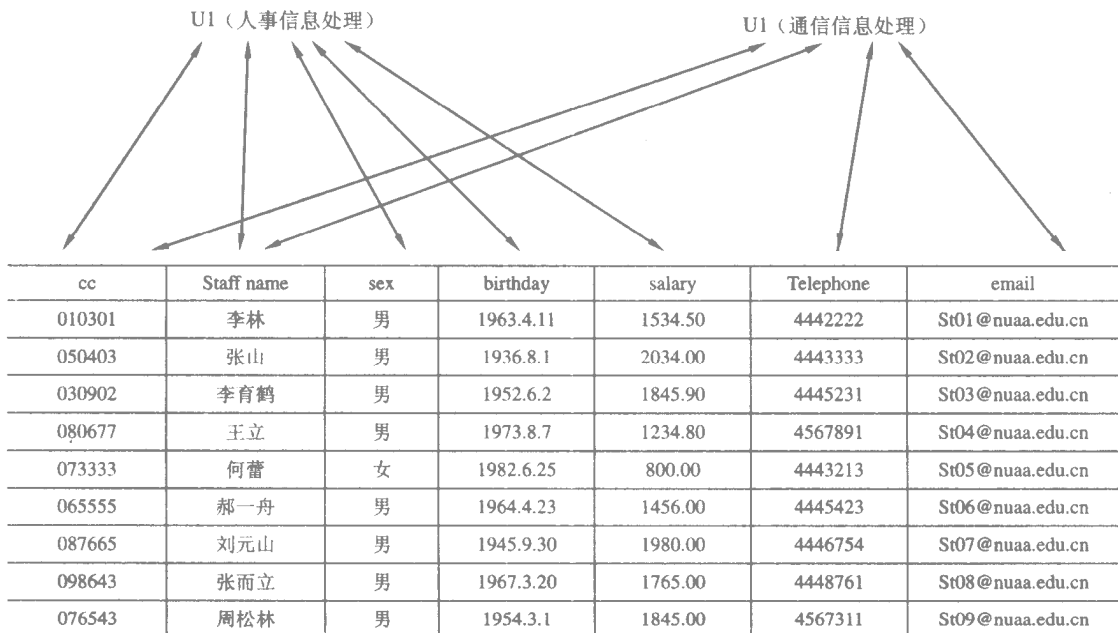


图 1-8 一个数据库的简单例子

2. 概念模式 (Conceptual Schema)

数据库的全局逻辑视图称为概念视图，由若干个概念记录类型组成。概念视图用 DDL (Data Definition Language；数据定义语言) 描述后得到的是概念模式（通常简称为模式）。描述概念视图的 DDL 称为“模式 DDL”。模式是所有概念记录类型的定义，因此它是数据库中全部数据逻辑结构的描述，即数据库中所有记录类型的整体描述。实际上，模式就是数据库每一层的数据结构（或数据模型）的描述。

模式还要描述记录之间的联系、所允许的操作、数据的一致性、有效验证、安全和其

他管理控制方面的要求。数据按外模式的描述提供给用户，按内模式的描述存储在磁盘中。而概念模式提供了一种约束其他两级的相对稳定的中间观点，它使得这两级的任何一级的改变都不受另一级的牵制。

在模式中还必须达到数据独立性。模式描述中必须不涉及到存储结构、访问技术等细节，这样，模式描述中就不会遇到存储字段的表达、记录的物理顺序、索引方式等存储/访问细节问题。只有这样，模式才算做到了数据独立，而在模式基础上定义的外模式才能做到数据独立。

3. 外模式 (External Schema)

外模式是用户与数据库系统的接口。

单个用户的视图称为外部视图。一个用户往往只用到数据库的一部分（如图 1-8 所示），外部视图由若干外部记录类型组成，这些外部记录类型和概念记录类型、内部记录类型可能不一样。用户使用数据操纵语言语句对数据库进行操作，实际上是对外部视图的外部记录进行操作。例如 GET 操作是读一个外部记录值（实际上是逻辑记录值），而不是数据库的内部记录值；每个外部视图用 DDL 描述后得到的是外模式。描述外部视图的 DDL 称为“外模式 DDL”。外模式是外部视图中每个外部记录类型的定义，因此，它是面向一个或多个用户需要的那部分数据的描述。在外模式中还必须定义外模式和概念模式间数据结构的差异。

外模式与概念模式的分离，使数据库系统的灵活性大大增加。外模式是从模式导出的，它必须是导出它的模式的逻辑子集。但在外模式里，不仅那些与用户无关的数据可以略去，而且数据项可以按照用户使用的习惯重新命名，记录类型也可以重新组合和命名。应用程序员不必再关心整个数据库的全局逻辑结构，它只与外模式发生直接联系，按照外模式存储和操纵数据。

对于一个数据库系统而言，任何一个应用程序都必须使用而且只能使用一个外模式，才能对数据库中的数据进行操作。一个模式可以支持若干个外模式，但每一个外模式只能属于一个模式。

在网状系统中，外模式的描述方法和应用程序所使用的程序设计语言（通常称作宿主语言）有关，例如，C 语言环境下外模式的描述与 FORTRAN 环境下的描述就不同，但它们完全可能对应同一个外模式。

4. 内模式 (Internal Schema)

内部视图是数据库结构中最低一级的逻辑表达，它由若干内部记录类型组成。内部记录也称为存储记录。内部视图用 DDL 描述后得到的是内模式，描述内部视图的 DDL 称为“内模式 DDL”。内模式要定义所有的内部记录类型，定义一些索引、数据在存储器的安排以及安全性、恢复和其他管理方面的细节。所以内模式是数据在物理存储结构方面的描述。

由于内部记录不涉及物理记录和物理块，也不涉及设备的约束，因此内部级与物理级是不同的。比内模式更接近于物理存储的那些软件机制是属于操作系统的一部分（即文件系统），例如从磁盘上读一部分数据或写一部分数据到磁盘上等操作。

外模式、模式、内模式都有“源”与“目标”之分。用 DDL 书写的都是源模式，系统不能直接使用，必须由数据库管理系统提供编译程序，将源模式编译成二进制的目标码，成为机器可使用的模式，即目标模式，放在目标库（一般称为数据字典）中，以供系统随时调用。

5. 模式 / 内模式映像

这个映像存在于概念级和内部级之间，用于定义模式和内模式间的对应性。由于这两级的数据结构可能不一致，即记录类型和字段类型的组成可能不一样，因此需要说明概念记录和字段怎样对应到内部记录和字段。如果内模式要作修改，即数据库的存储设备和存储方法有所变化，那么模式 / 内模式映像也要做出相应的修改，但模式很可能仍然保持不变。也就是对内模式的修改尽量不要涉及到模式，当然对于外模式和应用程序的影响就更小，这样我们就说数据库达到了物理数据独立性。

模式 / 内模式映像一般是在内模式中描述的，但有的数据库系统把映像的部分内容放在模式中描述。

6. 外模式 / 模式映像

这个映像存在于外部级和概念级之间，用于定义外模式和模式之间的对应性。由于这两级的数据结构可能不一致，因此需要说明外部记录和字段怎样对应到概念记录和字段。一个模式可能有多个外模式，每个用户只使用一个外模式，但不同的用户可共享同一个外模式，不同的外模式可以重叠。如果数据库的整体逻辑结构（即模式）要作修改，那么外模式 / 模式映像也要作相应的修改，但外模式很可能仍然保持不变。也就是对模式的修改尽量不要涉及到外部级的外模式，当然对于应用程序的影响就更小，这样我们就说数据库达到了逻辑数据独立性。外模式 / 模式映像都是在外模式中描述。

7. 用户

用户是指使用数据库的应用程序或联机终端用户。编写应用程序的语言仍然是高级程序设计语言，例如，FORTRAN、C 等，也可以是汇编语言。在数据库技术中，这些语言称为宿主语言，或简称为主语言。

数据库系统还提供数据操纵语言（Data Manipulation Language，简记为 DML）让用户使用。通常，DML 可自成系统在终端上直接对数据库进行操作。DML 也可嵌入在主语言中使用，此时主语言是经过扩充的能处理 DML 语句的主语言。

8. 用户界面（User Interface）

这是用户和数据库系统的一条分界线，在界线下，用户是不可知的。在数据库技术中，用户界面定义在外部级上，用户对于外模式还是可知的。模式和内模式对用户是透明的，不可知的。

需要指出，数据库模式和数据库是有区别的。数据库模式只是数据库的一种结构描述，还不是数据库本身，它是一个可以按照给定的某种约束装入具体数据值的框架，但它本身并不涉及任何具体数据值。我们常说“定义了一个数据库”，实际上只是指定义了一个数据库模式，只有按照该模式装入数据以后，这个数据库才算真正建立起来。通常，把一个数据库称为它的模式的一个“实例”（Instance）。一般，数据库的内容可随时变化，而模式则是相对稳定的。

从以上讨论可知，由于数据库这种多层次的体系结构，提供了高度的数据独立性。我们可将数据库技术的这一重要特征进一步归纳如下：

（1）一个数据库的全局逻辑描述（模式）是独立于数据库的其他所有结构描述的。在定义一个数据库的各层次结构时，全局逻辑结构（模式）应首先定义。

（2）内模式是依赖于全局逻辑结构的，其目的是具体地将概念模式中所定义的全部数据

及其联系进行适当的组织并加以存储，以实现较好的运行效率。但内模式独立于任何一个用户的局部逻辑结构模式（外模式）。如果采用段-页地址方式，那么内模式也可以独立于具体的存储设备。

(3) 用户的局部逻辑结构（外模式）是独立于存储模式和存储设备的，它必须在数据库的全局逻辑结构描述的基础上定义。但是，一个外模式一旦被定义，除非全局逻辑结构的变化使得外模式中的某些数据项无法再从数据库中导出，外模式将是不必改变的。我们总可以通过修改外模式 / 模式映像实现这一点，这就是外模式对于模式的相对独立性，即逻辑数据独立性。

(4) 应用程序是在外模式的数据结构上编制的，因此，它必须依赖于特定的外模式，但在一个完善的数据库系统中，它是独立于存储设备和存储模式的，并且只要数据库全局逻辑结构的变化不导致其对应的外模式的改变，应用程序也是独立于概念模式的。

上述结论都是对完善的数据库管理系统而言的，在目前现有的数据库商品软件中，三级结构的划分常常是不清晰的，有的系统仅仅支持一级数据结构。因此，不同的系统的数据独立性程度也是不同的，一般说来，关系数据库系统在支持数据独立性方面要优于其他类型的系统。

1.4 数据库系统

数据库系统 (Database Systems 简记为 DBS) 是一个复杂的系统，它是采用了数据库技术的计算机系统。因此数据库系统的含义已经不仅仅是一组对数据进行管理的软件 (即通常称为数据库管理系统) ，也不仅仅是一个数据库。一个数据库系统是一个实际可运行的，按照数据库方式存储、维护和向应用系统提供数据或信息支持的系统。它是存储介质、处理对象和管理系统的集合体，通常由数据库、硬件、软件和数据库用户四部分组成。

1. 数据库

数据库是与一个特定组织的各项应用相关的全部数据的汇集，通常由两大部分组成：一部分是有关应用所需要的工作数据的集合，称作物理数据库，它是数据库的主体；另一部分是关于各级数据结构的描述，称作描述数据库，通常是由一个数据字典系统管理。

2. 硬件支持系统

这一部分包括中央处理机、主存储器、外部存储设备、数据通道等各种存储、处理和传输数据的硬件设备。计算机系统一般是从市场上选购的，但对数据库系统来说，要特别关注内、外存储容量，I/O 存取速度，可支持终端数和性能稳定性等指标，在许多应用中还要考虑系统支持联网的能力和配备必要的后备存储设备等因素。

在数值计算中，计算机主机的运算速度决定着整个程序的运行速度。但在数据处理中，数据的存取速度主要是 I/O 所占的时间，因此 CPU 速度就不是数据库运算速度的决定因素。

数据库要存储大量的数据，外存是存放数据的介质。磁盘在计算机存储设备中目前占统治地位，磁盘容量已达每轴 5000 兆字节，还在继续增加。另外，随着磁泡存储器、电荷耦合器件、向光盘等存储技术的发展，大容量数据库的物质条件越来越优越。此外还要求系统有较高的 I/O (输入 / 输出) 能力，以提高数据的传输速度。

3. 软件支持系统

数据库系统的软件主要包括操作系统、各种宿主语言、实用程序以及数据库管理系统等。数据库管理系统 (Database Management Systems , 简称 DBMS) 是管理数据库的软件系统, 它是在操作系统的文件系统基础上发展起来的, 而且 DBMS 是在操作系统支持下工作的。为了开发应用系统, 还要有各种宿主语言 (如 FORTRAN、C 等) 及其编译系统, 这些语言应与数据库有良好的接口。

应用开发支撑软件是系统为应用开发人员和最终用户提供的高效率、多功能的交互式程序设计系统, 一般属于第四代语言 (4GL) 范畴。包括报表生成器、表格系统、图形系统、具有数据库存取和表格 I/O 功能的软件、数据字典等。它们为数据库应用系统的开发和应用提供了良好的环境, 使用户提高生产率 20~100 倍。使用第四代语言的人主要是终端用户, 从初学者到专家都可以快速简便地用它编写程序解决用户的问题。这些软件系统均以 DBMS 为核心。

4. 数据库用户

管理、开发和使用数据库系统的用户主要有数据库管理员 (Database Administrator , 简称 DBA)、系统分析员、应用程序员和用户。数据库系统中不同人员涉及到不同的数据抽象级别, 具有不同的数据视图。

(1) 用户。有应用程序和终端用户两类。它们通过应用系统的用户接口使用数据库, 目前常用的接口方式有菜单驱动、表格操作、图形显示、报表生成等, 这些接口给用户提供了简明直观的数据表示。

(2) 应用程序员。负责设计应用系统的程序模块, 根据外模式编写应用程序和对数据库的操作过程。

(3) 系统分析员。负责应用系统的需求分析和规范说明。他们要和用户及 DBA 相配合, 确定系统的软硬件配置并参与数据库各级模式的概要设计。

(4) DBA 要想成功地运转数据库, 就要在数据处理部门配备数据库管理人员——DBA。DBA 必须熟悉企业全部数据的性质和用途, 因此他是面向应用的, 对用户的需求有充分的认识。他对系统性能也非常关切, 因而要求他兼有系统程序员和运筹学专家的品质和知识。DBA 是控制数据整体结构的人, 负责保护和控制数据, 使数据能被任何有权使用的人有效使用。DBA 可以是一个人, 但一般是由几个人组成的一个小组, 其主要职责是:

1) 决定数据库的信息内容和结构, 确定某现实问题的实体联系模型, 建立与 DBMS 有关的数据模型和概念模式。

2) 决定存储结构和存取策略, 建立内模式和模式/内模式映像。使数据的存储空间利用率和存取效率两方面都较优。

3) 充当用户和 DBS 的联络员, 建立外模式和外模式/模式映像。

4) 定义数据的安全性要求和完整性约束条件, 以保证数据库的安全性和完整性。安全性要求是用户对数据库的存取权限; 完整性约束条件是对数据进行有效性检验的一系列规则和措施。

5) 确定数据库的后援支持手段及制订系统出现故障时数据库的恢复策略。

6) 监视并改善系统的性能, 提高系统的效率。

7) 当系统需要扩充和改造时, 负责修改和调整外模式、模式和内模式。