

# 第 1 章 数据库基础知识

在信息社会，数据已经成为重要的资源。大数据时代改变了人类原有的生活和发展模式，也改变了人类认识世界和价值判断的方式。以数据库技术为基础的数据管理技术，可以对数据进行有效的收集、加工、分析与处理，使得释放更多数据价值，充分发挥数据的作用。数据库技术是计算机学科的一个重要分支，是各类计算机信息系统的核心技术和重要基础，所有与数据信息有关的业务及应用系统都需要数据库技术的支持。数据库知识是当今大学生信息素养的重要组成部分，Microsoft 公司推出的 Access 数据库管理系统是集成在 Office 套装软件中的一个组件，它具有自身的特点和应用场合。从教学的角度讲，Access 很适合于初学者理解和掌握数据库的概念与操作方法。本书以常用的 Access 2010 作为实践环境。

本章围绕数据库系统的基础知识而展开。通过本章的学习，要掌握数据管理的概念、数据管理技术的发展、数据库的概念、数据库系统的组成与特点、数据模型的概念、数据库的体系结构、关系数据库的基本知识及数据库的设计方法等内容，这些内容对学习掌握 Access 2010 数据库操作及应用开发是十分必要的。

## 1.1 数据管理技术

人类在长期的社会实践中会产生大量数据，如何对数据进行高效的处理和应用成为迫切的现实需要。只有在计算机成为数据处理的工具之后，才使数据处理现代化成为可能。

### 1.1.1 数据与数据管理

#### 1. 数据和信息

数据 (Data) 和信息 (Information) 是数据处理中的两个基本概念，有时可以混用，如平时讲数据处理就是信息处理，但有时必须分清。一般认为，数据是人们用于记录事物情况的物理符号。为了描述客观事物而用到的数字、字符，以及所有能输入到计算机中并能被计算机处理的符号都可以看成数据。例如，章达明老师的基本工资为 2750 元，职称为教授，这里的“章达明”、“2750”、“教授”就是数据。在实际应用中，有两种基本形式的数据，一种是可以参与数值运算的数值型数据，如表示工资、成绩的数据；另一种是由字符组成、不能参与数值运算的字符型数据，如表示姓名、职称的数据。此外，还有图形、图像、声音等多媒体数据，如人的照片、商品的商标等。

信息是数据中所包含的意义。通俗地讲，信息是经过加工处理并对人类社会实践和生产活动产生决策影响的数据。不经过加工处理的数据只是一种原始材料，对人类活动产生不了决策作用，它的价值只是在于记录了客观世界的事实。只有经过提炼和加工，原始数据才发生了质的变化，给人们以新的知识和智慧。例如，收到一条微信，“有聚划算活动，商品 5 折”，就会根据数据“5 折”，再参照商品原价格（原始数据），计算出打折后的商品价格（新的数据），新的数据包含了新的信息，即商品降价了，可作为是否购买商品的依据。



数据与信息既有区别，又有联系。一方面，数据是信息的载体，但并非任何数据都能成为信息，只有经过加工处理之后具有新的内容的数据才能成为信息。另一方面，信息不随表示它的数据形式的改变而改变，它是反映客观现实世界的知识，而数据则具有任意性，用不同的数据形式可以表示同样的信息。例如，一个城市的天气预报情况是一条信息，而描述该信息的数据形式可以是文字、图像或声音等。

### 2. 数据处理和数据管理

数据处理是指将数据转换成信息的过程，它包括对数据的收集、存储、分类、计算、加工、检索和传输等一系列活动。其基本目的是从大量的、杂乱无章的、难以理解的数据中整理出对人们有价值、有意义的数据（信息），从而作为决策的依据。长期以来，人类主要用人工进行数据处理工作，使用计算机才实现了数据处理的自动化，才使数据处理的速度更快、效率更高。没有计算机，就不会有现代数据处理技术的形成和发展。从这个意义上讲，计算机是一个具有程序执行能力的数据处理工具，如图 1-1 所示。计算机数据处理所得到的输出数据（信息），除了取决于输入数据外，还取决于程序。程序不同，完成的数据处理方法不同，得到的结果也不同，即包含的信息也就不同。计算机通过不同的程序完成不同的数据处理任务。

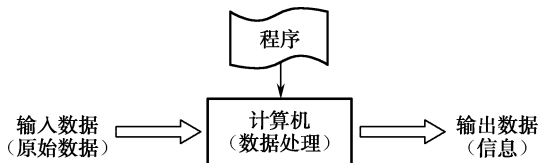


图 1-1 计算机数据处理

以高考成绩处理系统为例，全体考生各门课程的考试成绩记录了考生的考试情况，属于原始数据，对考试成绩进行分析和处理，如按成绩从高到低顺序排列、统计各分数段的人数等，进而可以根据招生人数确定录取分数线，输出的数据即包含了丰富的信息。

数据管理是指数据的收集、组织、存储、检索和维护等操作，这些操作是数据处理的中心环节，是任何数据处理业务中不可缺少的部分。数据管理的基本目的是为了实现数据共享、降低数据冗余、提高数据的独立性、安全性和完整性，从而能更加有效地管理和使用数据资源。

### 1.1.2 数据管理技术的变迁

计算机技术的发展和数据处理的现实需要，促使数据管理技术得到了很大发展，从而有效地提高了数据处理的应用水平。计算机数据管理技术经历了人工管理、文件管理和数据库管理 3 个发展阶段。

#### 1. 人工管理阶段

20 世纪 50 年代中期以前，计算机主要应用于科学计算，虽然当时也有数据管理的问题，但当时的数据管理是以人工管理方式进行的。在硬件方面，外存储器只有磁带、卡片和纸带等，没有磁盘等直接存取的外存储器；在软件方面，只有汇编语言，没有操作系统，没有对数据进行管理的软件。在人工管理阶段，数据管理有以下特点。

##### (1) 数据不保存

人工管理阶段处理的数据量较少，一般不需要将数据长期保存，只是在计算时将数据随应用程序一起输入，计算后将结果输出，数据和应用程序一起从内存中被释放。若要再次进行计算，则需重新输入数据和应用程序。

## (2) 由应用程序管理数据

系统没有专用的软件对数据进行管理，数据需要由应用程序自行管理。每个应用程序不仅要规定数据的逻辑结构，而且要设计数据的存储结构及输入/输出方法等，程序设计任务繁重。

## (3) 数据有冗余，无法实现共享

应用程序与数据是一个整体，一个应用程序中的数据无法被其他应用程序使用，因此，应用程序与应用程序之间存在大量的重复数据，数据无法实现共享。

## (4) 数据对应用程序不具有独立性

由于应用程序对数据的依赖性，数据的逻辑结构或存储结构一旦有所改变，则必须修改相应的应用程序，这就进一步加重了程序设计的负担。

以一所学校的信息管理为例，在人工管理阶段，应用程序和数据之间的关系如图 1-2 所示。图中各类数据对应于相应的应用程序，而且不同种类的数据之间会有许多重复的数据，如教务数据可能包含教师和学生的一部分数据。

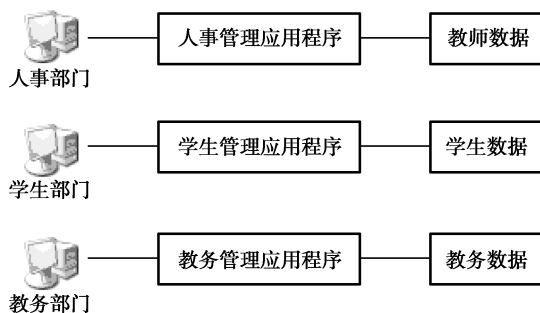


图 1-2 人工管理阶段应用程序和数据之间的关系

## 2. 文件管理阶段

20 世纪 50 年代后期至 60 年代后期，计算机开始大量用于数据管理。在硬件方面，出现了直接存取的大容量外存储器，如磁盘、磁鼓等，这为计算机数据管理提供了物质基础；在软件方面，出现了高级语言和操作系统。操作系统的出现为数据管理提供了技术支持。

应用程序利用操作系统的文件管理功能（由文件系统实现），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和管理，实现数据的文件管理方式。其特点有以下两个方面。

### (1) 数据可以长期保存

文件系统为应用程序和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和操作数据。数据可以组织成文件，能够长期保存、反复使用。

### (2) 数据对应用程序有一定的独立性

应用程序和数据不再是一个整体，而是通过文件系统把数据组织成一个独立的数据文件，由文件系统对数据的存取进行管理。程序员只需通过文件名来访问数据文件，不必过多考虑数据的物理存储细节，因此，程序员可集中精力进行算法设计，并大大减少了应用程序维护的工作量。

文件管理使计算机在数据管理方面有了长足的进步，时至今日，文件系统仍是高级语言普遍采用的数据管理方式。然而，当数据量增加、使用数据的用户越来越多时，文件管理便不能适应更有效地使用数据的需要了，其症结表现在以下 3 个方面。

① 数据的共享性差、冗余度大，容易造成数据不一致。由于数据文件是根据应用程序的需要而建立的，当不同的应用程序所使用的数据有相同部分时，也必须建立各自的数据文件，即数据不能共享，



造成大量数据重复。这样不仅浪费存储空间,而且使数据修改变得非常困难,容易产生数据不一致等问题,即同样的数据在不同的文件中所存储的数值不同,造成矛盾。

② 数据独立性差。在文件系统中,尽管数据和应用程序有一定的独立性,但这种独立性主要是针对某一特定应用而言的,就整个应用系统而言,文件系统还未能彻底体现数据逻辑结构独立于数据存储的物理结构的要求。在文件系统中,数据和应用程序是互相依赖的,即应用程序的编写与数据组织方式有关,如果改变数据的组织方式,就必须修改应用程序;而应用程序发生变化,如改用另一种程序设计语言来编写应用程序,也必须修改文件的数据结构。

③ 数据之间缺乏有机的联系,缺乏对数据的统一控制和管理。文件系统中各数据文件之间是相互独立的,没有从整体上反映现实世界事物之间的内在联系,因此,很难对数据进行合理的组织以适应不同应用的需要。在同一个应用项目中的各个数据文件没有统一的管理机构,数据完整性和安全性很难得到保证。

在文件管理阶段,学校信息管理中应用程序和数据文件之间的关系如图 1-3 所示。显然,教务数据中有教师、学生和课程等数据,数据冗余是难免的。各应用程序通过文件系统对相应的数据文件进行存取和处理,但各个数据文件基本上对应于相关的应用程序,而且各数据文件之间是孤立的,数据之间的联系无法体现,例如,因某个学生转学而在学生数据文件中删除了其数据,此时无法在教务数据文件中实现联动操作,因此仍然存在缺陷。

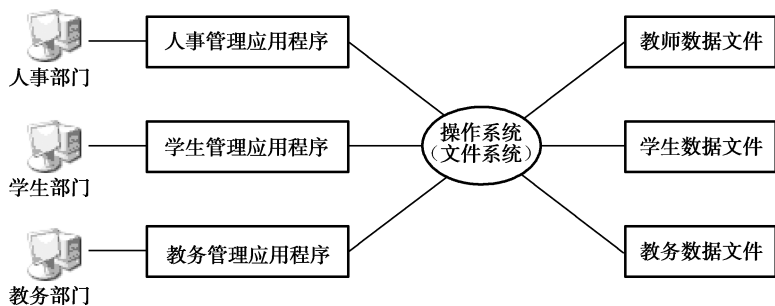


图 1-3 文件管理阶段应用程序和数据文件之间的关系

### 3. 数据库管理阶段

20 世纪 60 年代后期,计算机用于数据管理的规模更加庞大,数据量急剧增加,数据共享性要求更加强烈。同时,计算机硬件价格下降,而软件价格上升,编制和维护软件所需的成本相对增加,其中维护成本更高。这些成为计算机数据管理技术发展的原动力。

数据库(Database, DB)是按一定的组织方式存储起来的、相互关联的数据集合。在数据库管理阶段,由一种称为数据库管理系统(Database Management System, DBMS)的系统软件来对数据进行统一的控制和管理。数据库管理系统把所有应用程序中使用的相关数据汇集起来,按统一的数据模型存储在数据库中,为各个应用程序所使用。在应用程序和数据库之间保持较高的独立性,数据具有完整性、一致性和安全性高等特点,并且具有充分的共享性,有效地减少了数据冗余。

在数据库管理阶段,学校信息管理中应用程序和数据库之间的关系如图 1-4 所示。有关学校信息管理的数据都存放在一个统一的数据库中,数据库不再面向某个部门的应用,而是面向整个应用系统,实现了数据共享,并且数据库和应用程序之间保持较高的独立性。

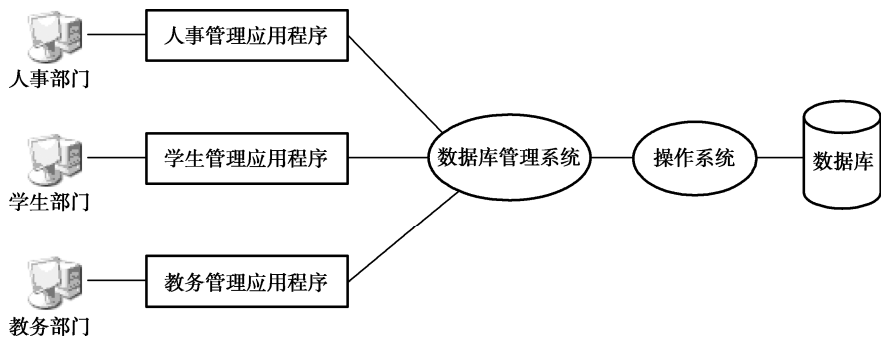


图 1-4 数据库管理阶段应用程序和数据库之间的关系

#### 4. 数据管理技术新发展

数据库技术的发展先后经历了层次数据库、网状数据库和关系数据库。层次数据库和网状数据库可以看作第 1 代数据库系统，关系数据库可以看作第 2 代数据库系统。自 20 世纪 70 年代提出关系数据库模型和关系数据库后，数据库技术得到了蓬勃发展，应用也越来越广泛。但随着应用的不断深入，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。例如，在实际应用中，除了需要处理数字、字符数据的简单应用之外，还需要存储并检索复杂的复合数据（如集合、数组、结构）、多媒体数据、计算机辅助设计绘制的工程图纸和地理信息系统（Geographic Information System, GIS）提供的空间数据等，对于这些复杂数据，关系数据库无法实现对它们的管理。正是实际应用中涌现出的许多问题，促使数据管理技术不断向前发展，出现了许多不同类型的新型数据管理技术。下面概要性地做一些介绍。

##### (1) 分布式数据库系统

分布式数据库系统（Distributed Database System, DDBS）是数据库技术与计算机网络技术、分布式处理技术相结合的产物。分布式数据库系统是系统中的数据地理上分布在计算机网络的不同结点上，但逻辑上属于一个整体的数据库系统。分布式数据库系统不同于将数据存储在服务上供用户共享存取的网络数据库系统，它不仅支持局部应用（访问本地数据库），而且能支持全局应用（访问异地数据库）。

分布式数据库系统的主要特点有以下 3 个方面。

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的数据库系统。

③ 结点的自治性：每个结点都有自己的计算机软硬件资源，包括数据库、数据库管理系统等，因而能够独立地管理局部数据库。局部数据库中的数据可以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) 面向对象数据库系统

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。面向对象数据库系统的基本设计思想是：一方面把面向对象的程序设计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,充分支持完整的面向对象概念和机制。

### (3) 多媒体数据库系统

多媒体数据库系统(Multimedia Database System, MDBS)是数据库技术与多媒体技术相结合的产物。随着信息技术的发展,数据库应用从传统的企业信息管理扩展到计算机辅助设计(Computer Aided Design, CAD)、计算机辅助制造(Computer Aided Manufacture, CAM)、办公自动化(Office Automation, OA)、人工智能(Artificial Intelligent, AI)等多种应用领域。这些领域中要求处理的数据不仅包括传统的数字、字符等格式化数据,还包括大量多种媒体形式的非格式化数据,如图形、图像、声音等。这种能存储和管理多种媒体数据的数据库称为多媒体数据库。

多媒体数据库的结构和操作与传统格式化数据库的结构和操作有很大差别。现有数据库管理系统无论是在模型的语义描述能力、系统功能、数据操作上,还是在存储管理、存储方法上,都不能适应非格式化数据的处理要求。综合程序设计语言、人工智能和数据库领域的研究成果,设计支持多媒体数据管理的数据库管理系统已成为数据库领域中一个新的重要研究方向。

在多媒体信息管理环境中,不仅数据本身的结构和存储形式各不相同,而且不同领域对数据处理的要求也比一般事务管理复杂得多,因而对数据库管理系统提出了更高的功能要求。

### (4) 数据仓库技术

随着信息技术的高速发展,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,企业界需要在大量数据基础上的决策支持,数据仓库(Data Warehouse, DW)技术的兴起满足了这一需求。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3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。

数据仓库、联机分析处理和数据挖掘是作为3种独立的数据处理技术出现的。数据仓库用于数据的存储和组织,联机分析处理集中于数据的分析,数据挖掘则致力于知识的自动发现。它们都可以分别应用到信息系统的设计和实现中,以提高相应部分的处理能力。但是,由于这3种技术内在的联系性和互补性,将它们结合起来就是一种新的决策支持系统架构。这一架构以数据库中的大量数据为基础,系统由数据驱动。

### (5) 大数据技术

大数据(Big Data)是规模非常巨大和复杂的数据集,传统数据库管理工具对其处理起来面临很多困难,如对数据库高并发读/写要求、对海量数据的高效率存储和访问需求、对数据库高可扩展性和高可用性的需求。

大数据有4个基本特征:数据规模大(Volume)、数据种类多(Variety)、要求数据处理速度快(Velocity)、数据价值密度低(Value),即所谓的4V特性。这些特性使得大数据区别于传统的数据概念。大数据的概念与海量数据不同,后者只强调数据的量,而大数据不仅用来描述大量的数据,还进一步指出数据的复杂形式、数据的快速处理特性,以及对数据分析处理后最终获得有价值信息的能力。

① 数据规模大:大数据聚合在一起的数据量是非常大的,根据国际数据公司IDC的定义至少要有超过100TB的可供分析的数据,数据量大是大数据的基本属性。

② 数据种类多:数据类型繁多、格式复杂是大数据的重要特性。数据来自多种数据源,非结构化数据大量涌现。由于非结构化数据没有统一的结构属性,难以用表结构来表示,在记录数据数值的同时还需要存储数据的结构,增加了数据存储和处理的难度。

③ 要求数据处理速度快：要求数据的快速处理是大数据区别于传统海量数据处理的重要特性之一。对于大数据而言，许多应用都要求能够实时处理，例如，有大量在线交互的电子商务应用，就具有很强的时效性。在这种情况下，大数据要求快速、持续的实时处理。

④ 数据价值密度低：数据价值密度低是大数据关注的非结构化数据的重要属性。传统的结构化数据，依据特定的应用，对事物进行了相应的抽象，每一条数据都包含该应用需要考虑的信息，而大数据为了获取事物的全部细节，不对事物进行抽象、归纳等处理，直接采用原始的数据，保留了全部数据，从而可以分析更多的信息，但也引入了大量没有意义的信息，这样就使得一方面数据的绝对数量激增，另一方面数据包含有效信息量的比例不断减少，数据价值密度偏低。

大数据处理技术就是从各种类型的数据中快速获得有价值信息的技术。大数据本质也是数据，但它包括非结构化数据，为此给数据管理技术带来了以下新的挑战。

① 大数据的表示面临挑战。用统一的模型对非结构化数据进行分析处理非常困难，传统的数据表示方法不能直观地展现数据本身的含义。为了有效利用数据并挖掘其中的知识，必须寻找最合适且有效的数据表示方法。目前使用的方法是数据标识，标识方法可减轻数据识别和分类的困难，但却给用户增添了预处理工作量。研究既有效又简易的数据表示方法是进行大数据处理首先面临的技术难题之一。

② 数据量的成倍增长给数据存储能力带来了挑战。传统的数据库追求高度的数据一致性和容错性，缺乏较强的扩展性和较好的系统可用性，不能有效存储视频、音频等非结构化和半结构化的数据。大数据及其潜在的商业价值要求使用专门的数据处理技术和专用的数据存储设备。目前，数据存储能力的增长远远赶不上数据的增长，设计最合理的分层存储架构成为信息系统的关键。

③ 数据融合技术面临挑战。数据不整合就发挥不出大数据的巨大价值，大数据面临的一个重要问题是个人、企业和政府机构的各种数据和信息能否方便地融合。如同人类有许多种自然语言一样，作为网络空间中唯一客观存在的数据难免有多种格式，但为了清除大数据处理的障碍，应研究推广不与平台绑定的数据格式。大数据已成为联系人类社会、物理世界和网络空间的纽带，需要通过统一的数据格式构建融合人、机、物三元世界的统一信息系统。

④ 数据跨越组织边界传播，给信息安全带来了巨大挑战。随着技术的发展，大量信息跨越组织边界传播，信息安全问题相伴而生，不仅使没有价值的信息大量出现，保密数据、隐私数据也成倍增长。国家安全、知识产权、个人信息等都面临着前所未有的安全挑战。大数据时代，犯罪分子获取信息更加容易，人们防范、打击犯罪行为更加困难，这对数据存储的物理安全性及数据的多副本与容灾机制提出了更高的要求。想要应对瞬息万变的安全问题，最关键的是算法和特征，如何建立相应的强大安全防御体系来发现和识别安全漏洞是保证信息安全的重要环节。

目前，围绕大数据，一批新兴的数据挖掘、数据存储、数据处理与分析技术不断涌现，使得人们能够将隐藏于海量数据中的信息和知识挖掘出来，从而为人类的社会经济活动提供决策依据。大数据将在商业智能、政府决策、公共服务等领域得到广泛应用。

## 1.2 数据库系统

数据库系统（Database System, DBS）是指基于数据库的计算机应用系统。和一般的应用系统相比，数据库系统有其自身的特点，它将涉及一些相互联系而又有所区别的基本概念。



## 1.2.1 数据库系统的组成

数据库系统是一个用来为用户提供信息服务的计算机应用系统。它通常由计算机硬件、软件、数据库和有关人员 4 部分组成，它们的逻辑关系如图 1-5 所示。

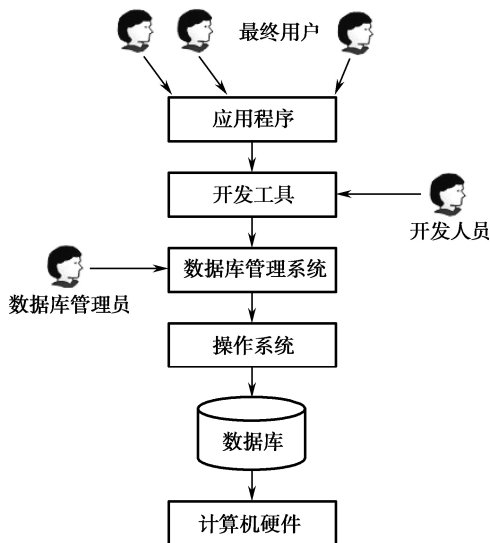


图 1-5 数据库系统

### 1. 计算机硬件系统

计算机硬件系统是数据库系统的物质基础，是存储数据库及运行相关软件的硬件设备，主要包括中央处理器（CPU）、存储设备、输入/输出设备及计算机网络环境。

### 2. 计算机软件系统

计算机软件系统包括操作系统、数据库管理系统、系统开发工具及数据库应用程序等。

#### (1) 操作系统

操作系统是所有软件的核心和基础，是其他软件运行的环境和平台。在计算机硬件层之上，由操作系统统一管理计算机的资源。

#### (2) 数据库管理系统

数据库管理系统在操作系统的支持下工作，是数据库系统的核心软件。常见的数据库管理系统有 Access、Visual FoxPro、SQL Server、Oracle、Sybase 等。

数据库管理系统是用户与数据库的接口，它可以实现数据的组织、存储和管理，提供访问数据库的方法，包括数据库的建立、查询、更新及各种数据控制等。数据库管理系统具有以下几个方面的功能。

① 数据定义功能。数据库管理系统提供数据定义语言（Data Definition Language, DDL），通过它可以方便地对数据库中的数据对象进行定义。

② 数据操纵功能。数据库管理系统提供数据操纵语言（Data Manipulation Language, DML），使用它可以实现对数据库中数据的基本操作，如修改、插入、删除和查询等。

③ 数据库运行管理功能。数据库管理系统通过对数据库的控制以适应共享数据的环境，确保数据库数据正确有效和数据库系统的正常运行。对数据库的控制主要通过以下 4 个方面实现。

- 数据的安全性控制：是指防止数据库的非法使用造成数据的泄露、更改或破坏。例如，系统提



供口令检查来验证用户身份，以防止非法用户使用系统。还可以对数据的存取权限进行限制，用户只能按所具有的权限对指定的数据进行相应的操作。

- **数据的完整性控制：**是指防止合法用户在使用数据库时向数据库加入不符合语义的数据，保证数据库中数据的正确性、有效性和相容性。正确性是指数据的合法性，如成绩只能是数值，不能包含字符；有效性是指数据是否在其定义的有效范围，如月份只能用 1~12 之间的正整数表示；相容性是指表示同一事实的两个数据应相同，否则就不相容，如一个人不能有两个性别。
- **多用户环境下的并发控制：**在多用户共享的系统中，多个用户可以同时存取数据库中的数据，甚至可以同时存取数据库中的同一个数据，并发控制负责协调并发事务的执行，保证数据库的完整性不受破坏。
- **数据的恢复：**是指在因某种故障引起数据库中的数据不正确或数据丢失时，系统能将数据库从错误状态恢复到最近某一时刻的正确状态。

④ **数据库的建立和维护功能。**它包括数据库初始数据的装入和转换功能、数据库的备份和恢复功能、数据库的重组功能，以及系统性能监视和分析功能等。

⑤ **其他功能。**如数据库管理系统与网络中其他软件系统的通信功能。

### (3) 系统开发工具

数据库系统开发工具是指各种数据库应用程序的编程工具。随着计算机技术的不断发展，各种数据库编程工具也在不断发展。目前，比较常用的数据库系统开发工具有 Visual Basic、C++、C#、Java 等通用程序设计语言。

### (4) 数据库应用程序

数据库应用程序是指系统开发人员利用某种开发工具开发出来的、面向某一类实际应用的软件系统。例如，人事管理系统、教学管理系统、证券实时行情分析系统等。

## 3. 数据库

数据库是指数据库系统中按照一定的方式组织的、存储在外部存储设备上的、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。它不仅包括描述事物的数据本身，而且还包括相关事物之间的联系。

数据库中的数据往往不是像文件系统那样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。其数据结构独立于使用数据的应用程序，对于数据的增加、删除、修改和检索由数据库管理系统进行统一管理和控制，用户对数据库进行的各种操作都是由数据库管理系统实现的。

### 4. 数据库系统的有关人员

数据库系统的有关人员主要有 3 类：最终用户（End User）、数据库应用系统开发人员和数据库管理员（Database Administrator, DBA）。最终用户指通过应用程序界面使用数据库的人员，他们一般对数据库知识了解不多。数据库应用系统开发人员包括系统分析员、系统设计员和程序员。系统分析员负责应用系统的分析，他们和最终用户、数据库管理员相配合，参与系统分析；系统设计员负责应用系统设计和数据库设计；程序员则根据设计要求进行编码。数据库管理员是数据管理机构的一组人员，他们负责对整个数据库系统进行总体控制和维护，以保证数据库系统的正常运行。

综上所述，数据库中包含的数据是存储在外部存储介质上的数据的集合。每个用户均可使用其中的数据，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。数据库管理系统为用户提供数据的存储组织、操作管理功能，用户通过数据库管理系统和应用程序实现数据库系统的操作与应用。



### 1.2.2 数据库系统的特点

数据库系统的出现是计算机数据管理技术的重大进步，它克服了文件系统的缺陷，提供了对数据更高级、更有效的管理。

#### 1. 数据结构化

在文件系统中，文件的记录内部是有结构的。例如，学生数据文件的每条记录是由学号、姓名、性别、出生日期、籍贯、简历等数据项组成的。但这种结构只适用于特定的应用，对其他应用并不适用。

在数据库系统中，每一个数据库都是为某一应用领域服务的。例如，学校信息管理涉及多个方面的应用，包括对学生的学籍管理、课程管理、学生成绩管理等，还包括教工的人事管理、教学管理、科研管理、住房管理和工资管理等，这些应用彼此之间都有着密切的联系。因此，在数据库系统中不仅要考虑某个应用的数据结构，还要考虑整个组织（多个应用）的数据结构。这种数据组织方式使数据结构化了，这就要求在描述数据时不仅要描述数据本身，还要描述数据之间的联系。而在文件系统中，尽管其记录内部已有了某些结构，但记录之间没有联系。数据库系统实现整体数据的结构化，这是数据库的主要特点之一，也是数据库系统与文件系统的本质区别。

#### 2. 数据共享性高、冗余度低

数据共享是指多个用户或应用程序可以访问同一个数据库中的数据，而且数据库管理系统提供并发和协调机制，保证在多个应用程序同时访问、存取和操作数据库数据时，不产生任何冲突，从而保证数据不遭到破坏。

数据冗余既浪费存储空间，又容易产生数据不一致等问题。在文件系统中，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冗余。

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，在数据库中用户的逻辑数据文件和具体的物理数据文件不必一一对应，从而有效地节省了存储资源，减少了数据冗余，保证了数据的一致性。

#### 3. 具有较高的数据独立性

数据独立性是指应用程序与数据库的数据结构之间相互独立。在数据库系统中，因为采用了数据库的三级模式结构，保证了数据库中数据的独立性。在数据存储结构改变时，不影响数据的全局逻辑结构，这样保证了数据的物理独立性。在全局逻辑结构改变时，不影响用户的局部逻辑结构及应用程序，这样就保证了数据的逻辑独立性。

#### 4. 有统一的数据控制功能

在数据库系统中，数据由数据库管理系统进行统一控制和管理。数据库管理系统提供了一套有效的数据控制手段，包括数据安全性控制、数据完整性控制、数据库的并发控制和数据库的恢复等，增强了多用户环境下数据的安全性和一致性保护。

## 1.3 数据模型

数据库是现实世界中某种应用环境（一个单位或部门）所涉及的数据的集合，它不仅要反映数据本身的内容，而且要反映数据之间的联系。由于计算机不能直接处理现实世界中的具体事物，所以必须将这些具体事物转换成计算机能够处理的数据。在数据库技术中，用数据模型（Data Model）对现实世界中的数据进行抽象和表示。

### 1.3.1 数据模型的组成要素

一般而言，数据模型是一种形式化描述数据、数据之间的联系及有关语义约束规则的方法。这些规则分为 3 个方面：描述实体静态特征的数据结构、描述实体动态特征的数据操作规则和描述实体语义要求的数据完整性约束规则。因此，数据结构、数据操作及数据的完整性约束也被称为数据模型的 3 个组成要素。

#### 1. 数据结构

数据结构研究数据之间的组织形式（数据的逻辑结构）、数据的存储形式（数据的物理结构）及数据对象的类型等。存储在数据库中的数据对象类型的集合是数据库的组成部分。例如，在教学管理系统中，要管理的数据对象有学生、课程、选课成绩等，在课程对象中，每门课程包括课程号、课程名、课程类别和学时等信息，这些基本信息描述了每门课程的特性，构成在数据库中存储的框架，即对象类型。

数据结构用于描述系统的静态特性，是刻画一个数据模型性质最重要的方面。因此，在数据库系统中，通常按照其数据结构的类型来命名数据模型。例如，层次结构、网状结构和关系结构的数据模型分别命名为层次模型、网状模型和关系模型。

#### 2. 数据操作

数据操作用于描述系统的动态特性，是指对数据库中的各种数据所允许执行的操作的集合，包括操作及有关的操作规则。数据库主要有查询和更新（包括插入、删除和修改等）两大类操作。数据模型必须定义这些操作的确切含义、操作符号、操作规则（如优先级）及实现操作的语言。

#### 3. 数据的完整性约束

数据的完整性约束是一组完整性规则的集合。完整性规则是给定的数据模型中数据及其联系所具有的约束和依存规则，用以限定符合数据模型的数据库状态及状态的变化，以保证数据的正确、有效和相容。

数据模型应该反映和规定数据必须遵守的完整性约束。此外，数据模型还应该提供定义完整性约束条件的机制，以反映具体所涉及的数据必须遵守的、特定的语义约束条件。例如，在学生信息中的性别的值只能为“男”或“女”，学生选课信息中的课程号的值必须取自学校已开设课程的课程号等。

### 1.3.2 数据抽象的过程

从现实世界中的客观事物到数据库中存储的数据是一个逐步抽象的过程，这个过程经历了现实世界、观念世界和机器世界 3 个阶段，对应于数据抽象的不同阶段，采用不同的数据模型。首先将现实世界的客观事物及其联系抽象成观念世界的概念模型，然后再转换成机器世界的数据库管理系统支持的数据模型。概念模型并不依赖于具体的计算机系统，它不是数据库管理系统所支持的数据模型，它是现实世界中客观事物的抽象表示。概念模型经过转换成为计算机上某一数据库管理系统支持的数据模型。所以说，数据模型是对现实世界进行抽象和转换的结果，这一过程如图 1-6 所示。

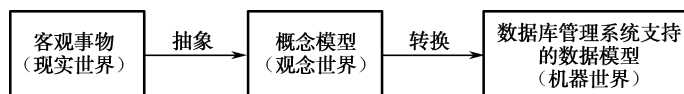


图 1-6 数据抽象的过程



### 1. 对现实世界的抽象

现实世界就是客观存在的世界，其中存在着各种客观事物及其相互之间的联系，而且每个事物都有自己的特征或性质。计算机处理的对象是现实世界中的客观事物，在对其实施处理的过程中，首先应了解和熟悉现实世界，从对现实世界的调查和观察中抽象出大量描述客观事物的事实，再对这些事实进行整理、分类和规范，进而将规范化的事实数据化，最终实现由数据库系统存储和处理。

### 2. 观念世界中的概念模型

观念世界是对现实世界的一种抽象，通过对客观事物及其联系的抽象描述，构造出概念模型。概念模型的特征是按用户需求观点对数据进行建模，表达了数据的全局逻辑结构，是系统用户对整个应用项目涉及的数据的全面描述。概念模型主要用于数据库设计，它独立于实现时的数据库管理系统。也就是说，无论选择何种数据库管理系统，都不会影响概念模型的设计。

概念模型的表示方法很多，目前较常用的是实体联系模型 (Entity Relationship Model)，简称 E-R 模型，用 E-R 图来表示。

### 3. 机器世界中的逻辑模型和物理模型

机器世界是指现实世界在计算机中的体现与反映。现实世界中的客观事物及其联系，在机器世界中以逻辑模型描述。在选定数据库管理系统后，就要将 E-R 图表示的概念模型转换为具体的数据库管理系统支持的逻辑模型。逻辑模型的特征是按计算机实现的观点对数据进行建模，表达了数据库的全局逻辑结构，是设计人员对整个应用项目数据库的全面描述。逻辑模型服务于数据库管理系统的应用实现。通常，也把数据的逻辑模型直接称为数据模型。数据库系统中主要的逻辑模型有层次模型、网状模型和关系模型。

物理模型是数据库底层的抽象，用以描述数据在物理存储介质上的组织结构，与具体的数据库管理系统、操作系统和硬件有关。

从概念模型到逻辑模型的转换是由数据库设计人员完成的，从逻辑模型到物理模型的转换是由数据库管理系统完成的，一般人员不必考虑物理实现细节，因而逻辑模型是数据库系统的基础，也是应用过程中要考虑的核心问题。

## 1.3.3 概念模型

当分析某种应用环境所需的数据时，总是首先找出涉及的实体及实体之间的联系，进而得到概念模型，这是数据库设计的先导。

### 1. 实体

实体 (Entity) 是现实世界中任何可以相互区分和识别的事物，它可以是能触及的客观对象，如一位教师、一名学生、一种商品等；还可以是抽象的事件，如一场演出、一次考试等。

### 2. 属性

每个实体都具有一定的特征或性质，这样才能区分不同的实体。例如，学生的学号、姓名、性别等都是学生实体具有的特征，考试的时间、地点、考生人数等都是考试实体的特征。实体的特征称为属性 (Attribute)，一个实体可用若干属性来刻画。能唯一标识实体的属性或属性集称为实体标识符。例如，学号可以作为学生实体的标识符。

性质相同的同类实体的集合称为实体集 (Entity Set)，例如，一个学院的所有学生，一个学院的全部期末考试等。在很多时候，实体集也常常简称为实体，例如，学生实体是指全部学生的集合。

### 3. 类型与值

属性和实体都有类型（Type）和值（Value）之分。属性类型就是属性名及其取值类型，属性值就是属性所取的具体值。例如，教师实体中的“姓名”属性，属性名“姓名”和取字符类型的值是属性类型，而“刘强希”、“章达明”等是属性值。每个属性都有特定的取值范围，即值域（Domain），超出值域的属性值则认为无实际意义的。例如，“性别”属性的值域为男、女，“职称”属性的值域为助教、讲师、副教授、教授等。由此可见，属性类型是个变量，属性值是变量所取的值，而值域是变量的取值范围。

实体类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，有相同的实体类型。实体值是一个具体的实体，是属性值的集合。

例如，教师实体类型是：

教师（编号，姓名，性别，出生日期，职称，基本工资，研究方向）

教师“章达明”的实体值是：

（T6，章达明，男，09/21/65，教授，2750，数据库技术）

由此可见，属性值所组成的集合表征一个实体，相应的这些属性名的集合表征一个实体类型，同类型实体的集合称为实体集。

在 Access 2010 中，用表来表示同一类实体，即实体集；用记录来表示一个具体的实体；用字段来表示实体的属性。显然，字段的集合组成一条记录，记录的集合组成一个表，实体类型则代表了表的结构。

### 4. 实体之间的联系

实体之间的联系（Relationship）是指一个实体集中可能出现的每一个实体与另一实体集中多少个具体实体存在联系。实体之间有各种各样的联系，归纳起来有 3 种类型。



#### （1）一对一联系

如果对于实体集  $A$  中的每一个实体，实体集  $B$  中至多只有一个实体与之联系，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，一个乘客只能坐一个座位，而一个座位只能被一个乘客占有，乘客与座位之间的联系是一一对一的联系，如图 1-7 所示。

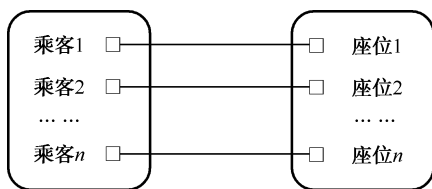


图 1-7 一对一联系

#### （2）一对多联系

如果对于实体集  $A$  中的每一个实体，实体集  $B$  中可以有多个实体与之联系，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  $B$  有一对多的联系，记为  $1:n$ 。例如，一个公司有许多员工，但一个员工只能在一个公司就职，所以公司和员工之间的联系是一对多的联系，如图 1-8 所示。

#### （3）多对多联系

如果对于实体集  $A$  中的每一个实体，实体集  $B$  中可以有多个实体与之联系，而对于实体集  $B$  中的每一个实体，实体集  $A$  中也可以有多个实体与之联系，则称实体集  $A$  与实体集  $B$  之间有多对多的联系，



记为  $m:n$ 。例如，一个供应商可以提供多种货物，任何一种货物可以由多个供应商供应，所以供应商与货物之间的联系是多对多的联系，如图 1-9 所示。

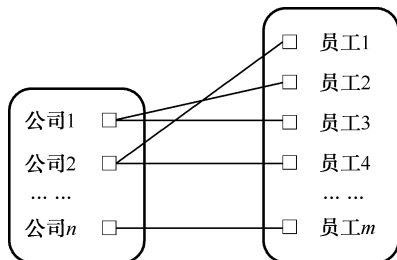


图 1-8 一对多联系

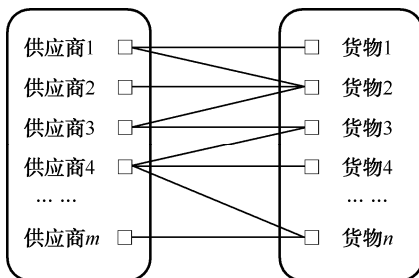


图 1-9 多对多联系

### 5. E-R 图

概念模型是反映实体及实体之间联系的模型。在建立概念模型时，要逐一给实体命名以示区别，并描述它们之间的各种联系。E-R 图是用一种直观的图形方式建立现实世界中实体及其联系模型的工具，也是数据库设计的一种基本工具。

E-R 图用矩形框表示现实世界中的实体，用菱形框表示实体间的联系，用椭圆形框表示实体和联系的属性，实体名、属性名和联系名分别写在相应框内。对于作为实体标识符的属性，在属性名下画一条横线。实体与相应的属性之间、联系与相应的属性之间用线段连接。联系与其涉及的实体之间也用线段连接，同时在线段旁标注联系的类型 ( $1:1$ 、 $1:n$  或  $m:n$ )。



前述乘客和座位的 E-R 图如图 1-10 所示，其中“身份证号”属性作为乘客实体的标识符（不同乘客的身份证号不同），“座位号”属性作为座位实体的标识符。联系也可以有自己的属性，如乘客实体和座位实体之间的“乘坐”联系具有“乘坐日期”属性。

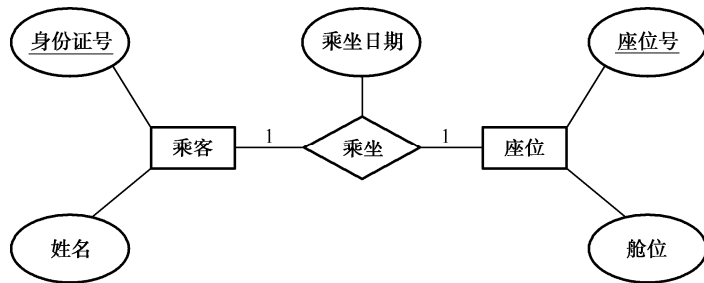


图 1-10 乘客和座位的 E-R 图

公司和员工的 E-R 图如图 1-11 所示，其中“公司名称”属性作为公司实体的标识符，“员工编号”属性作为员工实体的标识符。公司实体和员工实体之间的“拥有”联系具有“数量”属性。

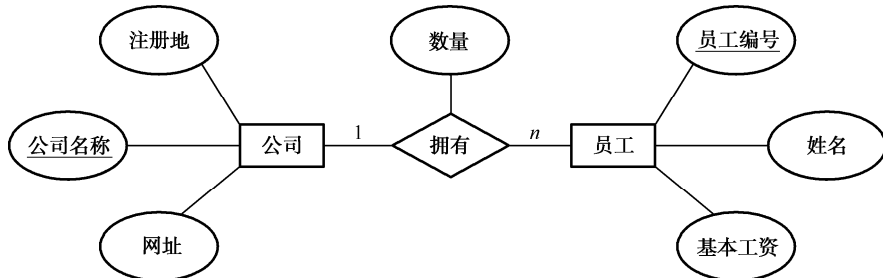


图 1-11 公司和员工的 E-R 图

供应商和货物的 E-R 图如图 1-12 所示, 其中“供应商号”属性作为供应商实体的标识符, “货物代码”属性作为货物实体的标识符。供应商实体和货物实体之间的“采购”联系可以有“采购日期”属性。

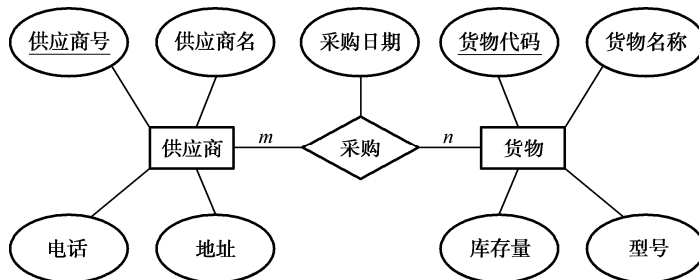


图 1-12 供应商和货物的 E-R 图

### 1.3.4 逻辑模型

概念模型只能说明实体间语义的联系, 还不能进一步说明详细的数据结构。在进行数据库设计时, 总是先设计概念模型, 然后再把概念模型转换成计算机能实现的逻辑模型, 如关系模型。逻辑模型不同, 描述和实现的方法也不同, 相应的支持软件即数据库管理系统也不同。在数据库系统中, 常用的逻辑模型有层次模型、网状模型和关系模型 3 种。

#### 1. 层次模型

层次模型 (Hierarchical Model) 用树形结构来表示实体及其之间的联系。在这种模型中, 数据被组织成由“根”开始的“树”, 每个实体由“根”开始沿着不同的分支放在不同的层次上。树中的每一个结点代表一个实体类型, 连线则表示它们之间的联系。根据树形结构的特点, 建立数据的层次模型需要满足如下两个条件。

- (1) 有一个结点没有父结点, 这个结点即根结点。
- (2) 其他结点有且仅有一个父结点。

如图 1-13 所示的层次模型就像一棵倒置的树, 根结点在最上面, 其他结点在下, 逐层排列。事实上, 许多实体间的联系本身就是自然的层次关系, 如一个单位的行政机构、一个家庭的世代关系等。

层次模型的特点是各实体之间的联系通过指针来实现, 查询效率较高。但由于受到如上所述的两个条件的限制, 层次模型可以比较方便地表示出一对一和一对多的实体联系, 而不能直接表示出多对多的实体联系, 对于多对多联系, 必须先将其分解为几个一对多联系, 才能表示出来。因此, 对于复杂的数据关系, 实现起来较为麻烦, 这就是层次模型的局限性。

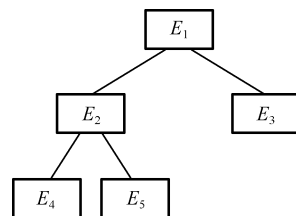


图 1-13 层次模型示例

采用层次模型来设计的数据库称为层次数据库。层次模型的数据库管理系统是最早出现的, 它的典型代表是 IBM 公司在 1968 年推出的 IMS (Information Management System), 这是世界上最早出现的大型数据库管理系统。

#### 2. 网状模型

网状模型 (Network Model) 用以实体类型为结点的有向图来表示各实体及其之间的联系, 其特点如下。

- (1) 可以有一个以上的结点无父结点。
- (2) 至少有一个结点有多于一个的父结点。

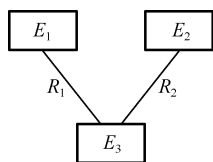


图 1-14 网状模型示例

在网状模型中,子结点与父结点的联系可以不唯一,因此要为每个联系命名,并指出与该联系有关的父结点和子结点。在图 1-14 中,  $E_3$  有  $E_1$  和  $E_2$  两个父结点,把  $E_1$  和  $E_3$  之间的联系命名为  $R_1$ ,  $E_2$  和  $E_3$  之间的联系命名为  $R_2$ 。

网状模型要比层次模型复杂,但它可以直接用来表示多对多联系。然而由于技术上的困难,一些已实现的网状数据库管理系统(如 Database Task Group 系统)中仍然只允许处理一对多联系。

网状模型的特点是各实体之间的联系通过指针实现,查询效率较高,多对多联系也容易实现。但是当实体集和实体集中实体的数目都较多时(这对数据库系统来说是理所当然的),众多的指针使得管理工作相当复杂,对用户来说使用也比较麻烦。

### 3. 关系模型

关系模型 (Relational Model) 用二维表格来表示实体及其相互之间的联系。在关系模型中,把实体集看成一个二维表格,每一个二维表格称为一个关系,每个关系均有一个名字,称为关系名。表 1-1 是一个教师关系,教师关系的每一行代表一个教师实体,每一列代表教师实体的一个属性。

表 1-1 教师关系

编号	姓名	性别	出生日期	职称	基本工资	研究方向
T1	刘强希	女	09/24/56	教授	3 200	软件工程
T2	罗佳旺	男	11/27/73	讲师	1 960	数据库技术
T3	黎达仁	男	12/23/81	助教	1 450	网络技术
T4	顾秋高	男	01/27/63	副教授	2 100	信息系统
T5	黄丹浩	女	07/15/79	助教	1 600	信息安全
T6	章达明	男	09/21/65	教授	2 750	数据库技术

#### (1) 元组

二维表格的每一行在关系中称为元组 (Tuple), 相当于表的一条记录 (Record)。二维表格的一行描述了现实世界中的一个实体。例如, 在表 1-1 中, 每行描述了一个教师的基本信息, 共包含有 6 位教师的信息, 即 6 个元组。

#### (2) 属性

二维表格的每一列在关系中称为属性 (Attribute), 相当于记录中的一个字段 (Field) 或数据项。每个属性有一个属性名, 一个属性在其每个元组上的值称为属性值, 因此, 一个属性包括多个属性值, 只有在指定元组的情况下, 属性值才是确定的。同时, 每个属性有一定的取值范围, 称为该属性的值域, 如表 1-1 中的第 3 列, 属性名是“性别”, 取值是“男”或“女”, 不是“男”或“女”的数据应被拒绝存入该表, 这就是数据约束条件。同样, 在关系数据库中, 列是不能重复的, 即关系的属性名不允许相同。属性必须是不可再分的, 即属性是一个基本的数据项, 不能是几个数据项的组合。

#### (3) 关系模式

关系模型是由若干个关系组成的, 关系用关系模式 (Relational Schema) 来描述。关系模式就相当于前面提到的实体类型, 它代表了关系的结构, 也就是二维表格的框架 (表头)。对于教师关系可以表示为:

教师 (编号, 姓名, 性别, 出生日期, 职称, 基本工资, 研究方向)

#### (4) 关键字

关系中能唯一区分、确定不同元组的单个属性或属性组合, 称为该关系的一个关键字。关键字又称为键或码 (Key)。单个属性组成的关键字称为单关键字, 多个属性组合的关键字称为组合关键字。



需要强调的是，关键字的属性值不能取空值，因为空值无法唯一地区分、确定元组。所谓空值，就是不知道或不确定的值，通常记为 Null。

在如表 1-1 所示的关系中，“性别”属性无疑不能充当关键字，“职称”属性也不能充当关键字，从该关系现有的数据分析，“编号”和“姓名”属性均可单独作为关键字，但“编号”属性作为关键字会更好一些，因为可能会有教师重名的现象，而教师的编号是不会相同的。这也说明，某个属性能否作为关键字，不能仅凭对现有数据进行归纳确定，还应根据该属性的取值范围进行分析判断。

关系中能够作为关键字的属性或属性组合可能不是唯一的。凡在关系中能够唯一区分、确定不同元组的属性或属性组合，称为候选关键字 (Candidate Key)。例如，表 1-1 所示关系中的“编号”和“姓名”属性都是候选关键字 (假定没有重名的教师)。

在候选关键字中选定一个作为关键字，称为该关系的主关键字或主键 (Primary Key)。关系中主关键字的取值是唯一的。

#### (5) 外部关键字

如果关系中某个属性或属性组合是另一个关系的关键字，则称这样的属性或属性组合为本关系的外部关键字或外键 (Foreign Key)。在关系数据库中，用外部关键字表示两个表之间的联系。例如，在表 1-1 的教师关系中，增加“部门代码”属性，则“部门代码”属性就是一个外部关键字，该属性是部门关系的关键字，该外部关键字描述了教师和部门两个实体之间的联系。

虽然关系模型比层次模型和网状模型发展得晚，但它的数据结构简单、容易理解，而且它建立在严格的数学理论基础之上，所以是目前比较流行的一种数据模型。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。本书讨论的 Access 2010 就是一种关系数据库管理系统。

## 1.4 数据库的体系结构

为了有效地组织和管理数据，提高数据库的逻辑独立性和物理独立性，数据库的体系结构采用三级模式和二级映射结构。三级模式包括外模式、概念模式和内模式，二级映射则分别是概念模式到内模式的映射及外模式到概念模式的映射，如图 1-15 所示。

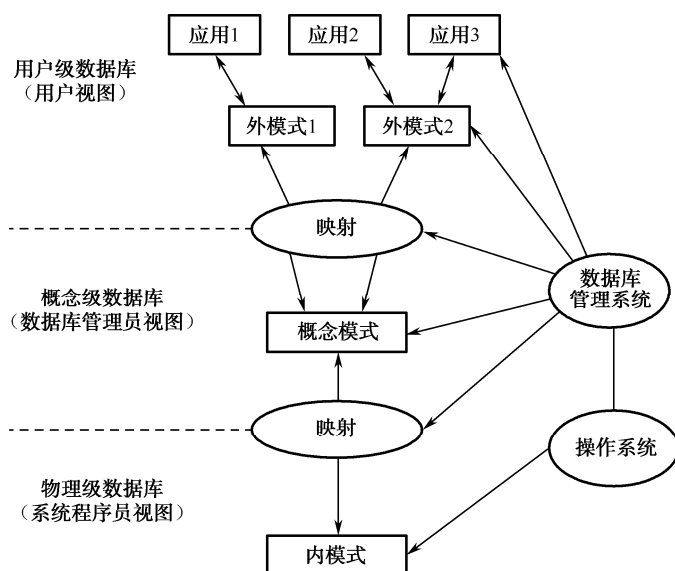


图 1-15 数据库的三级模式及二级映射



## 1.4.1 数据库的三级模式

三级模式使不同级别的用户对数据库形成不同的视图 (View)。所谓视图, 就是指观察、认识和理解数据的范围、角度和方法, 是数据库在用户中的反映。很显然, 不同层次 (级别) 用户所看到的数据库是不同的, 由此形成了面向用户或应用程序的用户级数据库、面向建立和维护数据库人员的概念级数据库及面向系统程序员的物理级数据库。用户级数据库对应外模式 (External Schema), 概念级数据库对应概念模式 (Conceptual Schema), 物理级数据库对应内模式 (Internal Schema)。

### 1. 概念模式

概念模式又称逻辑模式, 或简称为模式, 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)。概念模式是由数据库系统提供的数据库定义语言 (Data Definition Language, DDL) 来描述、定义的, 体现并反映了数据库系统的整体观。

### 2. 外模式

外模式又称子模式或用户模式, 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。外模式是从概念模式导出的一个子集, 包含概念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定义语言 (外模式 DDL) 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或物理模式, 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 (数据文件的结构)、存取方法 (主索引和辅助索引机制) 及外存的空间分配, 对应着实际存储在外存储介质上的数据库。内模式由内模式定义语言 (内模式 DDL) 来描述、定义, 反映了数据库的存储观。

在一个数据库系统中, 只有唯一的数据库, 因此作为定义、描述数据库存储结构的内模式和定义、描述数据库逻辑结构的概念模式, 也是唯一的, 但建立在数据库系统之上的应用则是非常广泛、多样的, 所以对应的外模式不是唯一的, 也不可能唯一。

#### 【例 1-1】数据库三级模式举例。

以货物供应管理为例, 将图 1-12 所示的 E-R 图转换为如下关系模型 (转换方法将在 1.6.2 节中介绍)。

供应商 ( 供应商号, 供应商名, 地址, 电话)

货物 ( 货物代码, 货物名称, 型号, 库存量)

采购 ( 供应商号, 货物代码, 采购日期)

由此得到货物供应管理数据库的三级模式如图 1-16 所示。

概念模式对应的是数据抽象过程中的逻辑模型 (如关系模型), 而不是概念模型 (E-R 模型)。外模式是针对具体用户应用需要的数据而设计的, 与该用户无关的数据就不必放入, 外模式中的数据可以从逻辑模型的数据库中得到。内模式涉及数据文件的组织方式及为了提高查询速度而引入的索引机制。在应用数据库的过程中, 主要强调数据库的逻辑结构, 而不太关注数据库的存储结构和数据存储的具体实现方式。

