第1章

数据库的基本概念

学习目标

了解数据库系统的发展历史
掌握在数据库系统中用户看到的数据与计算机数据库中存放的数据两者之间的联系
掌握数据库系统的组成
了解现今数据库系统开发的最新技术

数据库技术是计算机数据管理的最新技术，是计算机科学的重要分支。当计算机的主要应用领域从科学计算转变到数据及事务处理时，数据库技术便应运而生并成为计算机科学的重要领域。今天，数据库技术不仅在企业管理信息系统（MIS）、计算机集成制造系统（CIMS）、办公信息系统(OIS)、地理信息系统(GIS)、Internet 技术等方面得到广泛应用，且越来越多的新应用领域都采用数据库来存储和处理它们的信息资源。对于一个国家来说，数据库的建设规模、数据库信息量的大小和使用频度、数据库的安全性和可靠性已成为衡量一个国家信息化程度的重要标志。因此，“数据库原理及其应用”课程成为计算机科学与技术专业、信息管理专业的重要专业课程之一。

在本章中，我们将从计算机数据管理的历史变迁过程出发，介绍数据库以及数据库系统的组成，并介绍与其相关的一些基本概念，其目的是使读者从数据库技术的总体发展上了解数据库技术的概貌，并为后面章节的学习打下基础。

1.1 数据管理的历史变迁

所谓数据管理是指对数据进行分类、组织、编码、存储、检索和维护的一系列操作，它是数据处理的前提和核心问题。

自 1946 年第一台电子计算机在美国问世以来，伴随着计算机硬件技术和软件技术的发展，计算机的应用领域已从单纯的科学计算逐步渗透到数据管理领域。现在，用于数据管理的计算机数量已远远超过用于科学计算的计算机。在应用需求的推动和计算机硬件、软件发展的基础上，计算机数据管理技术不断更新和完善，主要经历了以下三个阶段：

（1）人工管理阶段（20 世纪 50 年代中期以前）；

（2）文件系统阶段(从 20世纪 50 年代后期到 20世纪 60 年代中期)；

（3）数据库系统阶段（从 20 世纪 60 年代后期至今）。

1.1.1 人工管理阶段

在这一阶段中，从计算机的应用来看，计算机的主要应用领域是科学计算。从硬件上看，这个时期计算机内存空间小，计算速度低，外存只有磁带、卡片和纸带，没有像磁盘这样快速的直接存取的存储设备。从软件上看，计

算机没有操作系统，更没有数据管理软件，数据处理是以批处理方式进行的。

人工管理阶段具有如下特点：

（1）数据不保存。由于当时计算机主要用于科学计算，强调提高计算的速度和侧重于对精度的提高，相对而言处理的数据量较少，一般不需要将数据长期保存。此外，限于内存和外存的空间和速度，其数据也不便于长期保存。因此，只是在计算某一课题时将数据输入计算机，数据用完后就从内存中撤走。不仅对用户数据如此处置，对系统软件涉及的数据有时也是这样。

（2）程序员管理数据。由于当时没有相应的软件系统负责数据的管理工作，应用程序中涉及的数据需要由程序员自己管理，即程序员在程序中不仅要编写规定数据逻辑结构的程序代码，而且还要编写设计数据物理结构的代码，包括数据存储结构、存取方法和输入方式的代码等。

（3）数据不共享。数据共享（Data Sharing）是指多个用户、多种语言、多个应用程序相互覆盖地使用一些共同的数据集合。在人工管理阶段，由于数据是在程序中定义的，一组数据只能对应一个程序。当多个应用程序涉及某些相同的数据时，由于必须在各自的程序中定义，无法或很难互相利用、互相参照，因此程序与程序之间的相同数据无法共享。

（4）数据不具有独立性。由于数据的逻辑结构和物理结构均在程序编码中定义，如果因为某种原因，需要改变数据的逻辑结构或物理结构时，必须对应用程序代码作相应的修改，即数据与程序之间不具有独立性，这不仅加重了程序员的负担，且使用起来也很不方便、费时费力。

在人工管理阶段，应用程序与数据之间的一一对应关系可用图 1.1 表示。其中，左边表示数据从程序外部通过键盘等设备进入程序的情形，右边表示数据集成在程序中的情形。

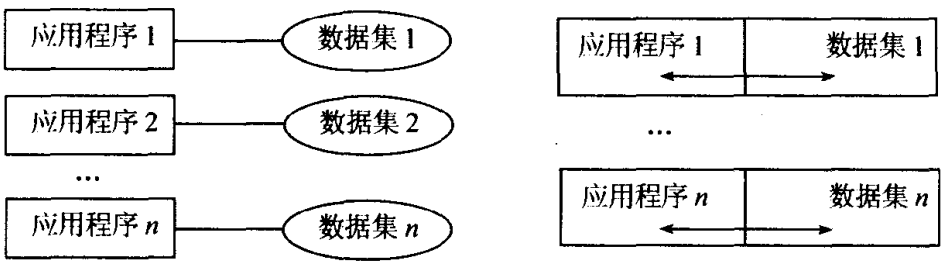


图 1.1　人工管理阶段应用程序与数据之间的对应关系

1.1.2 文件系统阶段

在这一阶段中，计算机应用不仅仅在科学计算，而且还开始大量地应用于数据处理。从硬件上看，计算机内存空间增大，计算速度得到很大的提高，外存有了磁盘、磁鼓等快速的直接存取设备；从软件上看，计算机配备了操作系统，且操作系统中已经有了专门的数据管理软件——文件系统。从数据处理方式上看，不仅有批处理，而且能够联机实时处理。

文件系统阶段有如下特点：

（1）数据可以长期保存。由于计算机大量用于数据处理，数据需要长期保存以便反复使用。把数据以文件的形式保存在磁盘等外存储器上，为数据的长期保存和反复使用（查询、修改、插入和删除等）提供了保证。

（2）文件的多样化和结构化。由于有了直接存取设备，也就出现了顺序文件、索引文件、随机文件等多种文件组织形式。数据文件的存取基本上以记录为单位，而记录则是由一些字段按特定的结构组成的，每个记录的长度是相等的。因此，文件是记录的集合，文件系统实现了记录内部的有结构性，但在整体上数据仍然是无结构的。

（3）文件系统管理数据。操作系统提供的文件管理系统是应用程序与数据文件的专门接口。程序和数据文件之间的存取操作由文件系统根据“按文件名访问，按记录进行存取”的管理技术自动完成。文件的逻辑结构到物理结构的转换由文件系统来自动完成，而不必由程序员设计考虑，从而使得程序和数据分开，数据与程序之间有了一定的独立性。这样，程序员可以集中精力于数据的逻辑结构，而不必或者很少考虑其物理结构。数据在物理结构上的改变，通常不会反映在程序上，从而可以大大地节省维护程序的工作量。

文件系统的上述特点比人工管理阶段有了很大的改进，但仍存在如下缺点：

（1）数据冗余度大。数据冗余（Data Redundancy）是指同一组数据在多个文件中同时出现所引起的数据重复现象。在文件系统中，一个文件基本上对应于一个应用程序，即文件系统中文件仍由应用程序来定义。当不同的应用程序具有部分相同的数据时，也必须建立各自的数据文件，而不能共享相同的数据，例如，在人事文件中包含了企业所有雇员的信息，而销售文件中又包含了属于销售人员的雇员信息。此外由于相同的数据在不同的文件中重复存储、并由各自的程序管理，容易造成数据的不一致性，给数据

的修改和维护带来困难。

（2）数据独立性较差。由于文件的逻辑结构是在应用程序中定义的，文件系统中的文件是为某一特定应用服务的，因此，一旦数据的逻辑结构改变，必须修改应用程序中关于文件结构的定义。反过来，应用程序的改变，例如应用程序改用不同的高级语言编写等，也将引起数据文件结构的改变。因此数据与程序之间独立性较差。

（3）数据联系弱。数据联系（Data Relationship）是指不同文件中的数据之间的联系。在文件系统阶段，不同文件中的数据相互之间是独立的，缺乏相互关联的方法，因此数据联系弱。虽然文件之间的某些数据存在着紧密的逻辑联系，但是由于实现这种逻辑联系的复杂性，很少在文件管理系统中提供这些数据之间的联系方法，因此，各个数据文件之间是孤立的，不能反映现实世界中客观事物之间的内存联系。

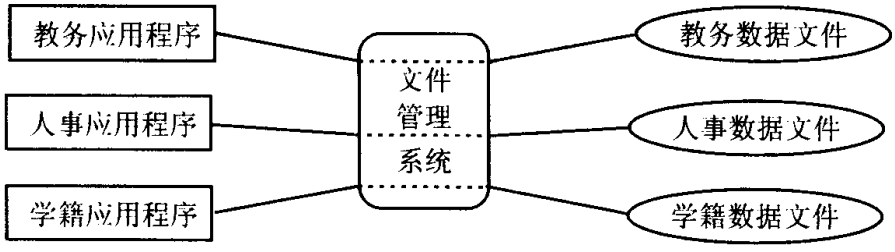


图 1.2　文件系统阶段应用程序与数据文件的对应关系

在文件系统阶段，程序与数据文件之间的关系如图 1.2 所示。其中虚线表示应用程序与数据文件的对应关系是通过文件管理系统实现的。

1.1.3　数据库系统阶段

在数据库系统阶段，计算机应用范围越来越广泛，应用于数据管理的规模越来越大，数据量急剧增长，人们对数据共享的要求越来越强烈。从硬件上看，随着大规模集成电路的发展，计算机硬件价格大大下降，计算速度更快且逐渐出现大容量的磁盘、光盘等直接存取设备。从软件上看，虽然操作系统开始成熟，程序设计语言功能也更加强大，但随着硬件价格的下降，软件价格却不断上升，为编制和维护系统软件及应用程序所需的成本相对增加。此外，在数据处理方式上也有联机实时处理的更大需求，并开始提出和考虑分布式处理方法。在这样的背景下，以文件系统作为数据管理的方法已经不能满足实际应用的需要，为了满足多用户、多应用共享数据的需求，就产生了数据库技术，并出现对数据进行统一管理的专门软件系统——数

据库管理系统(DBMS)。

与文件系统相比，数据库系统有如下特点：

(1) 整体数据的结构化。在文件系统中虽然实现了记录内的结构性，但在整体上数据却是无结构的，即它仅关心记录内部数据项之间的联系，而在不同文件中的记录之间没有联系。在数据库系统中，数据模型不仅描述数据本身的特征，而且还要描述数据之间的联系，且这种联系通过存取路径（指针）来实现整体数据的结构化。通过存取路径表示自然的数据联系，从而实现整体数据的结构化，是数据库系统与传统的文件系统之间的本质差别。数据库系统使数据不再面向特定的某个或某些应用，而是面向整个应用系统，因此大大降低了数据冗余度，实现了数据的共享。

此外，在数据库系统中不仅数据是结构化的，且存取数据的方式也很灵活，可以存取数据库中的某一个数据项、一组数据项、一个记录或一组记录。而在文件系统中，数据的最小存取单位是记录，粒度不能细到数据项。

例 1.1 一个学校的排课管理系统中，假设一个老师可以上多门课，一班要学习多门课程，则排课信息可以用三个记录型——班级记录、教师记录、教室记录来表示(见图 1.3)。

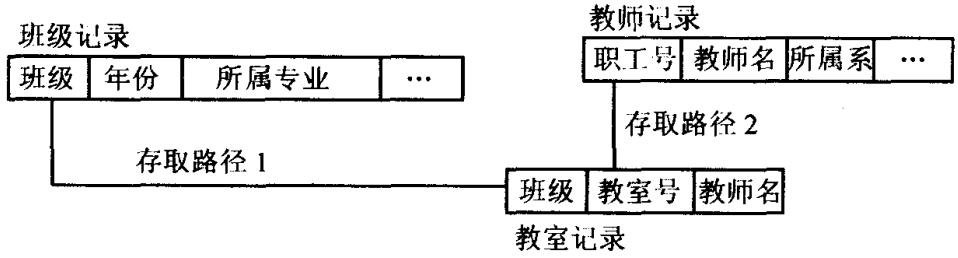


图 1.3 班级和教师在排课管理中的数据联系

为了查询在 2010 班王红老师所上的课程及其所在教室，如果使用文件系统实现排课管理，则由于班级记录、教师记录、教室记录分别存放在三个文件之中且文件之间的记录没有联系，程序员必须编写一个程序，分别从三个文件中找出所需数据。如果采用关系数据库系统进行排课管理，由于班级记录与教室记录可以通过公共字段“班级”（存取路径 1)联系，教师记录与课程表记录可以通过公共字段“教师名”（存取路径 2)联系，数据库系统通过存取路径可以自动建立班级记录、教师记录和教室记录之间的联系。这样，在关系数据库系统中只需一条命令即可查询“2010 班王红老师所上课程及其所在教室”。

(2) 数据独立性高。由于数据库系统中数据与程序的独立性使得数据的定义从程序中分离出去，加上数据的存取也由 DBMS 负责，从而简化了

应用程序的编制，大大减少了应用程序的开发、维护和修改费用。

（3）数据的共享性高，冗余度低，易扩充。数据库系统从整体角度看待和描述数据，数据不再面向某个应用而是面向整个系统，因此数据可以被多个用户、多个应用共享使用。数据共享可以大大减少数据冗余，节约存储空间。数据共享还能够避免数据之间的不相容性与不一致性。

所谓数据的不一致性是指同一数据的不同拷贝（即同一数据在不同文件中）其值不一样。采用人工管理或文件系统管理时，由于数据被重复存储，当不同的应用程序使用和修改不同的拷贝时就很容易造成数据的不一致。在数据库中数据共享，减少了由于数据冗余造成的不一致现象。

由于数据面向整个系统，是有结构的数据，不仅可以被多个应用共享使用，而且容易增加新的应用，这就使得数据库系统弹性大，易于扩充，可以适应各种用户的要求。用户可以存取整体数据的各种子集以用于不同的应用系统，当应用需求改变或增加时，只要重新选取不同的子集或加上一部分数据便可以满足新的需求。

（4）提供了完整的控制功能。数据库的共享是并发（Concurrency）的共享，即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项。为此，DBMS 提供以下几方面的数据控制功能：

1）数据的安全性（Security）。数据的安全性是指保护数据以防止不合法的使用造成数据的泄密和破坏。它使每个用户只能按规定对某些数据以某些方式进行使用和处理。例如，DBMS 使用了检查口令、定义用户保密级或数据库存取权限等机制。当用户对数据进行操作时，系统自动地检查用户是否有权执行此项操作。只有通过安全检查的用户才能执行所允许的操作。

2）数据的完整性（Integrity）。数据的完整性指数据的正确性、有效性和相容性。完整性检查将数据控制在有效的范围内，或保证数据之间满足一定的关系。例如：月份只能是 1 至 12 之间的正整数，学生不能无性别，学号是惟一的，在一般情况下，一个学生不能在两个以上的学院注册学习等。

3）并发控制（Concurrent Control）。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的结果或使得数据库的完整性遭到破坏，因此 DBMS 提供了对多用户的并发操作加以控制和协调的功能。

4）数据库恢复（Recovery）。计算机系统的硬件故障、软件故障、操作员的失误以及故意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。因此，DBMS 提供了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功能。

数据库管理阶段应用程序与数据之间的对应关系可用图 1.4 来表示。

综上所述，数据库是长期存储在计算机外存设备上的、有组织的综合性数据集合，它可以供各种用户共享。 DBMS 是一个数据管理软件系统，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，并在多个用户同时使用数据库时进行并发控制，在发生故障后对系统进行恢复。应用程序是针对某个应用而开发的程序，它完成用户对数据的操作，且这种操作是通过 DBMS 来具体实现的。

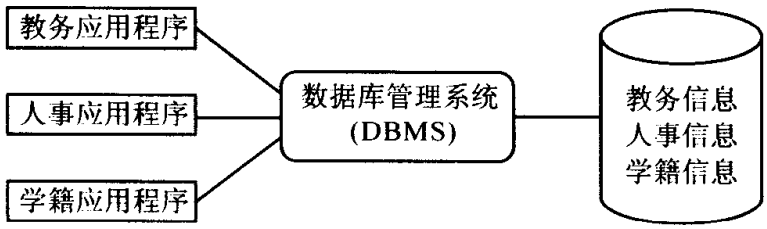


图 1.4 数据库系统阶段应用程序和数据的对应关系

1.2 数据模型

提起模型，大家一定并不陌生。比如汽车模型、建筑模型和飞机模型等，都是我们熟悉的常见模型。这些模型属于实物模型，它们通常是客观事物的外观特征或功能特征的模拟与刻画。此外，还可以用抽象模型来刻画客观事物的某些特征，比如数学模型 $S=\pi R$ 就是一种抽象模型，它抽象地表示了圆的周长与圆的半径之间的数量关系特征。一般地说，模型是现实世界某些特征的模拟和抽象。数据模型(Data Model)也是一种抽象模型。它是客观事物某些特征的数据抽象和模拟。由于计算机只能存储和处理数据而不能直接存储和处理现实世界中的客观事物，也就无法直接用计算机来管理一个企业的设备、产品和职工等，因此，我们必须先把企业的设备、产品和职工等客观事物的某些特征抽象成计算机能够存储和处理的数据，才能用计算机对其进行管理。那么，用什么数据格式来抽象表示客观事物的数据特征呢？那就是数据模型。在数据库中普遍采用数据模型这个工具来抽象表示和处理客观事物的数据特征和信息。因此，数据模型是数据库系统的核心，了解数据模型的基本概念是学习数据库的基础。

1.2.1 数据模型的构成

一般地讲，数据模型是严格定义的一组概念的集合。这些概念精确地描述了数据的静态特性、动态特性和完整性约束条件。因此数据模型一般由数据结构、数据操作和完整性约束三部分构成，通常称为数据模型的三要素。

（1）数据结构。数据结构的概念同信息一样，至今没有一个大家都认可的定义。通常认为，数据结构是计算机数据组织方式和数据之间联系的框架描述。例如，前面介绍的班级记录型，班级（班级名、入学年份、所属专业、班级人数、所在学院）就是一种数据结构，数据文件中的数据就是按照这个数据结构进行组织的。更一般地说，班级记录型其实是对班级这个对象类（Object Type）的抽象描述，因此，可以说数据结构是所研究的对象类（Object Type）的集合。这些对象是数据库的组成成分，一般可分为两类：一类是与数据类型、内容、性质有关的对象，例如网状模型中的数据项、记录，关系模型中的域、属性、关系等；另一类是与数据之间联系有关的对象，例如关系模型中的外键（Foreign Key）。数据结构是刻画一个数据模型的最关键要素，因此在数据库系统中，人们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。数据结构是对数据库静态特性的描述。

（2）数据操作。数据操作是指对数据库中各种对象类的实例（或取值）所允许执行的操作的集合，包括操作方法及有关的操作规则，它是对数据库动态特性的描述。在数据库中，数据操作主要有数据检索和更新（包括插入、删除、修改）两大类。数据模型定义了这些操作的语义、操作符号、操作规则（如优先级）以及实现操作的语言。

（3）完整性约束。完整性约束是指对数据的一组完整性规则（约束条件）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，用以限定数据库中数据的状态以及状态的变化，以保证数据的正确性、有效性和相容性。一个数据模型通常都规定了本数据模型必须遵守的基本和通用的完整性约束条件。例如，在关系模型中，任何关系必须满足实体完整性和参照完整性两个条件（将在后面的章节中做详细介绍）。此外，一个数据模型还应该提供定义完整性约束条件的机制，以反映具体应用所涉及的数据必须遵守的特定的约束条件。例如，某大学的数据库中规定，学生选修某门课程合格所得学分不能为负数，每个学生一学期内成绩不

格的课程门数不得超过 3 门等等。

1.2.2 数据模型的分类

在介绍数据模型的分类之前，先介绍一下数据库系统的用户与数据库设计人员的关系。

（1）用户（User）。用户是指最终用户（End User），他是数据库系统的使用者，通常不是计算机专业人员，他关心的是现实世界中的事物、事物的属性及其相互关系。例如，排课管理数据库的用户可能关心的是教室及其属性，如教室编号、教室名、地点、安排时间的情况等等。用户也关心自己的教学资源，如教师队伍构成、教室的种类与数量等。用户通过数据库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、窗口事件驱动、表格操作 图形显示、报表书写等等，它们给用户提供了更直观、更接近现实的数据表示，尽量让用户使用起来方便。

（2）数据库设计员（DB Designer）。数据库设计员是数据库系统的分析与设计者，通常也称为系统分析员（Analyst），他们通常是数据库方面的专家，负责应用系统的需求分析和规范说明。他们也关心现实世界，但更关心如何对现实世界进行正确的数据抽象，以便输入计算机并对其进行有效的管理。因此，数据库设计人员主要负责数据库的需求调查、分析和规范说明，且与用户及 DBA 合作，确定系统的硬件软件选型和配置，并建立用户的概念模型，完成数据库各级模式的概要设计和数据库设计。作为需求分析的结果，数据库设计人员必须用恰当的数据模型，以文档的形式对需求进行结构化描述。关于数据库需求分析的方法将在第 5 章中介绍。下面介绍数据模型的分类问题。

通常，一个好的数据模型除了应该具备 1.2.1 节中提出的三个要素以外，还应该满足以下三方面的性能要求：一是能比较真实地模拟或抽象表示现实世界；二是容易为人所理解；三是便于在计算机上实现。如果能够找到完全满足这三个要求的数据模型，那么数据库设计和实现就会简单得多。因为我们只要将现实世界抽象成这样的数据模型，然后在计算机中的 DBMS 上实现即可。遗憾的是目前还没有找到这样的数据模型。因此，当前在数据库系统研究和设计开发中常常是采取多步抽象的方法，针对不同的抽象层次和应用目的，分别采用不同的数据模型。根据不同的抽象层次，数据模型有以下三类：

（1）概念数据模型（Conceptual Data Model）。这种数据模型是用户容

易理解的、对现实世界特征的数据抽象，它与具体的 DBMS 无关，是数据库设计员与用户之间进行交流的语言。

数据模型是数据库系统的核心和基础。各种机器上实现的 DBMS 软件都是基于某种数据模型的，如目前最流行的关系数据模型。由于 DBMS 所支持的数据模型常常有诸多规定和限制，一般非计算机专业人员不容易理解。对某个企业的数据库，如果我们的数据库设计员一开始就用 DBMS 所支持的数据模型来设计，就难以得到企业员工理解和参与，也就容易导致数据库系统开发的失败，因为数据库设计员通常对企业的实际情况不够了解，如果没有企业员工参与，所设计的数据库一般难以真实地反映企业实际。因此，概念数据模型主要用来描述现实世界的概念化结构，它能使数据库设计员在设计的初始阶段摆脱计算机系统及 DBMS 的具体技术问题，集中精力分析数据和数据之间的联系，并能与企业员工进行充分交流，从而使其设计能真实地反映企业的客观实际。概念数据模型经转换变成 DBMS 支持的逻辑数据模型后，就能在 DBMS 中实现。最常用的一种概念数据模型是实体一联系(E－R)模型，简称 E－R 模型(在 1.2.3 节中介绍)。

(2) 结构数据模型(Structural Data Model)，又称逻辑数据模型(Logical Data Model)。它是用户从数据库中所看到的数据模型，是具体的 DBMS 所支持的数据模型，比如网状数据模型、层次数据模型、关系数据模型和面向对象数据模型等，它们都是结构数据模型的一种。结构数据模型既要面向用户，也要面向系统，一般由概念数据模型转换得到。一般的 DBMS 支持一种结构数据模型，特殊的 DBMS 可以支持多种结构数据模型。

(3) 物理数据模型(Physical Data Model)。这是描述数据在存储介质上组织结构的数据模型，它不但与具体的 DBMS 有关，而且还与操作系统和硬件有关，是物理层次的数据模型。每一种结构数据模型在实现时都有其对应的物理数据模型。为了保证数据的独立性和可移植性，DBMS 自动完成大部分物理数据模型的实现工作，而设计者只设计索引、聚簇等特殊结构。

图 1.5 表示了以上三类数据模型在数据库设计和实施过程中的地位和顺序。首先，数据库设计员根据用户的实际情况，将现实世界的数据特征抽象为概念数据模型，当确认该模型真实地描述了企业实际之后，数据库设计员将概念数据模型转化为 DBMS 支持的某一种结构数据模型，并利用 DBMS 提供的数据定义语言(DDL)将其书写为程序源码(称为模式)，DBMS 将其转换为数据库目标模式，并在 OS 支持下自动将其转换为物理数据模型，并按物

理数据模型规定的格式将其存放在外存储介质上，形成数据库文件。

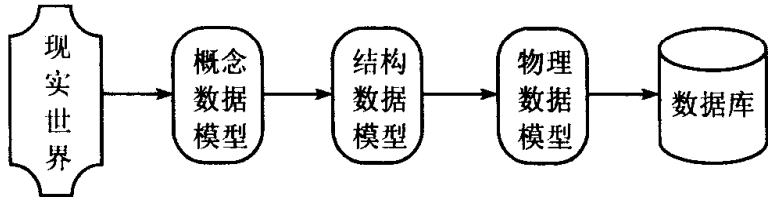


图 1.5　三类数据模型在数据库设计过程中的地位和关系

1.2.3　实体一联系(E－R)模型

由图 1.5 可以看出，概念数据模型是数据库设计员在数据库设计过程中对现实世界特征的第一层次的数据抽象，也是数据库设计员和用户之间进行交流的语言，因此概念数据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、易于用户理解。P. P. S. Chen 于 1976 年提出的实体—联系方法(Entity-Relationship Approach),简称 E－R 方法或 E－R 模型，就具备这些性质，因而是一种概念数据模型。虽然概念数据模型的表示方法很多，但其中 E－R 模型却是最为常用和最为著名的。 E－R 模型用 E－R 图来抽象表示现实世界的特征，是一种语义表达能力强且易于理解的概念数据模型。

1. E－R 模型中的基本概念

(1) 实体 Éntity)。客观存在并可相互区别的事物都称为实体。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,王红、张涛、环境与工程系、离散数学、张涛选修离散数学、环境与工程系购买实验设备、张涛老师在环境与工程系工作等都是实体。实体不能严格地精确定义，就像几何学中的“点”和“线”等概念一样。理解这个概念的关键之处是一个实体能和别的实体相区别。实体的可区分性非常类似面向对象模型中的对象具有可标识的特点，因此，也有人将 E－R 模型归结为面向对象数据模型。

(2) 属性 Áttribute)。实体通常具有若干特征，每个特征称为实体的一个属性。属性是相对实体而言的，是实体所具有的特征。例如，每个教师实体都具有职工号、姓名、年龄、性别、系别、工龄等属性(职工号、姓名等称为属性名),每个属性赋予确定的值,如(10010304,张涛,39,男,计算机,15)就用数据抽象地表示了一个确定的教师实体。

(3) 实体型(Entity Type)。具有相同属性的一类实体必然具有共同

的特征和性质。因此，用实体类名（简称实体名）及其属性名集合来抽象和刻画同类实体，称为实体型。例如，教师(职工号、姓名、年龄、性别、系别、工龄)就是一个实体型，这里的‘教师’是实体名。在数据库设计中，实体名、属性名均可由设计人员根据实际需要自由命名。比如，“教师”这个实体名就可以用 Teacher 等来命名。

(4) 实体集(Entity Set)。若干同型实体的集合称为实体集。例如，信息学院的学生就是一个实体集。实体集的名一般可以使用实体型的名。在实际应用中，一个实体型通常可以被抽象地看作一个实体集。

(5) 关键字(Key)。能惟一地标识实体集中每个实体的属性集合称为关键字（码）。例如，学号可以作为一个学校的学生实体集的关键字；若同一个年级没有重名现象发生，在同一个学校里，姓名与年级也可以作为该校学生实体集的关键字。

(6) 域(Domain)。属性的取值范围称作域。例如，职工号的域为 10 位整数构成的字符串集合，性别的域为集合 {男，女} 。

(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 E-R 模型中反映为实体之间的联系。实体之间的联系分为实体集内部的联系和实体集之间的联系两类。实体集内部的联系是指其内部实体之间的联系。实体集之间的联系是指一个实体集的实体与另一个实体集的实体之间的关联。两个实体集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 1.2 在一个学校里，所有的（正）班长为一个实体集 A，所有的班级为一个实体集 B。一个班级只有一个班长，而一个班长只负责一个班级的工作，则班长实体集 A 与班级实体集 B 之间具有一对一联系。这里的“负责”是两个实体集联系的方式，也称联系名。

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 ($n\geqslant 0$)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 1.3 在一个学校里，所有的班主任为一个实体集 A，所有的学生为实体集 B。一个班主任负责管理若干名学生，而每个学生只有一个班主任，则班主任实体集 A 与学生实体集 B 之间具有一对多联系。同样，这里的“管理”也是两个实体集联系的联系名。

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($n\geqslant 0$)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($m\geqslant 0$)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 1.4 在一个学校里，所有的学生为实体集 A，所有的课程为一个实体集 B。一个学生可以同时选修多门课程，而一门课程同时有若干个学生选修,则学生实体集 A 与课程实体集 B 之间具有多对多联系。这里的“选修”是两个实体集联系的联系名。

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。

图 1.6 按 E-R 模型给出了两个实体集之间的以上三类联系的一般结构和实例,其中矩形表示实体集,框内写明实体集名,菱形表示联系,框内写明联系名。

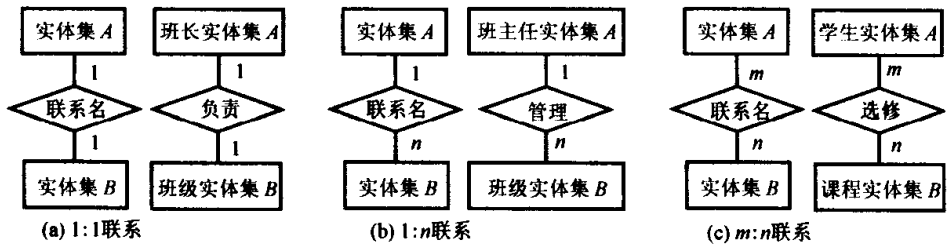


图 1.6 两个实体集之间的三类联系

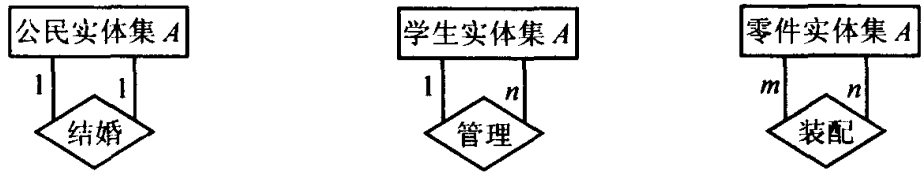
实体集内部的联系也有 $1:1$,$1:n$ 和 $m:n$ 三类联系。

例 1.5 1）我国所有的公民为实体集 A 且假定没有再婚现象，则每个公民都是 A 的一个实体。一个男性公民至多与一个女性公民结婚，一个女性公民至多与一个男性公民结婚。所以，公民这个实体集 A 中的实体之间具有一对一联系。这里的‘结婚’是实体集 A 内部的联系名。

2）在一个学校里，所有的学生为一个实体集 A,班长也是学生,所以是 A 的一个实体。一个班长可以管理若干学生（一个班的学生），而一个学生只由一个班长领导，则学生实体集 A 中的实体之间具有一对多联系。这里的‘管理’是实体集 A 内部的联系名。

3）在一个企业里，所有的零件是一个实体集 A。A 中若干个实体（如螺栓、螺帽等无结构零件）经装配所得的零件（称为有结构零件）也是 A 的实体。反过来，A 的一个实体(有结构的零件)可由 A 的多个实体(零件)装配而成，则零件实体集 A 中的实体之间具有多对多联系。这里的“装配”是实体集 A 内部的联系名。

以上三种联系可用图 1.7 表示。



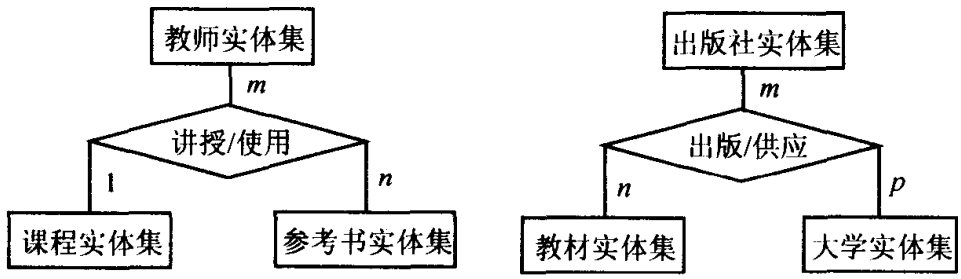
(a) 1:1联系　　(b) 1:*n*联系　　(c) *m*:*n*联系

图 1.7　实体集内部实体之间的三类联系

此外，两个以上的实体集之间也存在着一对一、一对多、多对多联系。下面介绍两个例子，说明三个实体集之间的一对多、多对多联系。

例 1.6　1）对于课程、教师与参考书这三个实体集，如果若干个教师可同时讲授一门课程，每一个教师只讲授一门课程且讲课时可使用若干本参考书，而每一本参考书只供一门课程使用，则教师与课程、参考书之间的联系是一对多的，可记作 $1:m:n$ 联系。这里的‘讲授／使用’是三个实体集联系的联系名，如图 1.8(a)所示。

2）有三个实体型：出版社、教材、大学，一个出版社可以出版多种教材供给多个大学，而每个大学可以使用多个出版社供应的教材，每种教材可由不同出版社供给，由此可知出版社、教材和大学三者之间是多对多的联系，可记作 $m:n:p$ 联系，如图 1.8(b)所示。



(a) 1:*m*:*n*联系　　(b) *m*:*n*:*p*联系

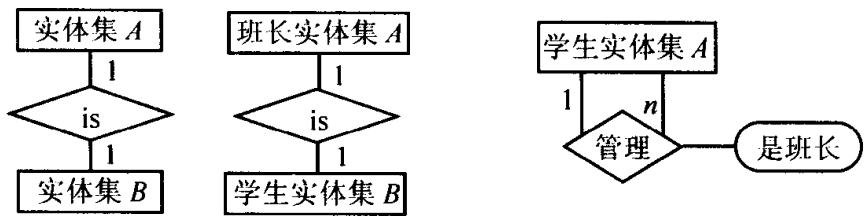
图 1.8　三个实体型之间的联系示例

实体之间的联系除了前面介绍的 $1:1$，$1:n$，$m:n$ 以外，还有一种 is－a 联系，它是实体之间的层次联系。这种联系本质上是 $1:1$，$1:n$ 联系的一种特殊情形，但在面向对象数据模型中使用 is－a 联系更能表示对象（实体）之间的继承关系。

例 1.7　实体集合 A 的实体也是实体集合 B 的实体，A 中实体有某些特殊的特征，需要用自己的特殊属性来描述，这些属性对在 A 中但不在 B 中的实体无意义。也就是说，A 是 B 的特殊类别。其表示方法如图 1.9(a)所示。在我们用 E－R 模型描述概念模型式时，如 A 和 B 存在 is－a 联系，

则 A 的每个实体 a 正好和 B 的一个实体 b 相联系，而 B 的一个实体最多和 A 的一个实体相联系，因此，从这个意义上说它们是一对一的联系。而在现实世界中，a 和 b 实际上是同一事物。A 能继承 B 的属性并可增加别的属性来特别说明 A，A 的关键字实际是 B 的属性，也是 B 的关键字。比如学生实体集合与班长实体集合之间存在着 is－a 联系，即班长是(is－a)学生。例如张三是班长，当将张三作为班长来处理有关他的数据时（如查询他所管理的学生），我们处理 A(班长实体集合)的一个实体 a；当将张三作为一个学生来处理有关他的数据时（如求平均成绩），我们处理 B(学生实体集合)的一个实体 b。对 a 和 b 都有的属性，其值是相同的，实际数据库中只需存储一份。

值得注意的是，上面的实体集合 A 和实体集合 B 也可以指同一个实体集合，此时表示同一实体集合的实体间的联系，A 中特殊实体需要增加的属性作为联系的属性即可，见图 1.9(b)。



(a) 两个实体集的is-a联系　　(b) 实体集内部的is-a联系

图 1.9 is－a 联系的两种情况

2. E－R 图要点

概念数据模型是现实世界特征的第一层次的数据抽象，它应该能够方便、准确地表示出现实世界中的常用概念，且易于用户理解。E－R 方法（模型）就是一种最常用，也是最著名的概念数据模型。

E－R 模型提供了表示实体型（集）、属性和联系的方法，上节已经介绍过实体、实体型、实体集和联系的概念和表示方法，这里再集中介绍一下这些概念的彼此区别以及 E－R 图的要点。关于如何认识和分析现实世界，并从中抽取实体型和实体集之间的联系，以及建立 E－R 模型的方法将在数据库设计一章加以详细介绍。

实体是客观存在并可相互区别的一个事物，比如 205 教室、王红、浙江大学等。实体集则是由若干同类型实体所构成的集合，比如，205 教室的所有学生就构成一个学生的集合。而实体型则是对某个实体集中实体共同特征和性质的抽象描述，比如，205 教室的学生实体集中每个学生都有学号、

姓名、性别、年龄、所在系、入学时间等特征，因此，这个实体集的实体型可描述为 学生(学号，姓名，性别，年龄，所在系，入学时间)。在实际应用中，一个实体型通常可以被抽象地看作是一个实体集，也可以被看作是一个实体。同样，一个实体集有时也可以被看作是一个实体。因此，在以后各个章节中经常不加严格区分地使用实体、实体集和实体型这三个概念，读者可以从上下文之间理解其含义。比如在图 1.10所示的 E-R 图中，左边的课程及课程号、课程名和学分等构成一个实体型，右边的学生及学号、姓名、性别和出生日期等构成另一个实体型。由于它们要通过“选修”进行联系，而实体型与实体型之间是无法联系的，因此，它们实际上表示的是两个实体集，而在实际应用中有时也把它们称为实体。

根据前面的分析，可将 E-R 图中的符号约定如下。

- 实体(集、型)用矩形表示，矩形框内写明实体名；
- 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的实体连接起来，同时在无向边旁标上联系的类型(1∶1，1∶n或 m∶n)。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。
- 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 1.8 图 1.10给出了学生实体集与课程实体集及其联系的 E-R 图。其中学生实体具有学号、姓名、性别和出生日期等属性，课程实体有课程号，课程名和学分等属性。这里的联系‘选修’具有‘成绩’属性。

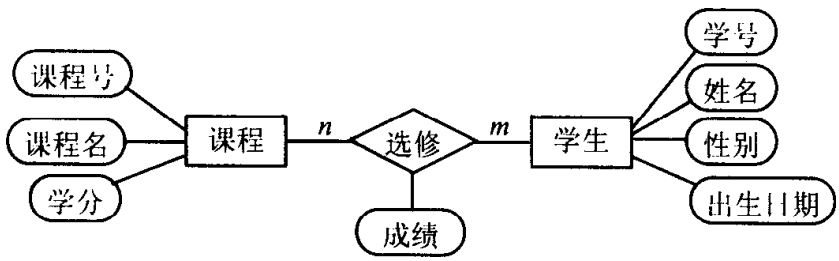


图 1.10 一个简单的 E-R图

E-R 模型是概念模型的一种，是抽象描述现实世界的有力工具。用 E-R 图表示的概念模型独立于具体的 DBMS 所支持的数据模型，比结构数据模型更具有一般意义，更接近现实世界。

下面通过一个简化的学生选课系统例子来介绍 E-R 图的两种画法。

例 1.9 设有教师、学生、班级、课程和参考书五个实体集，则描述学生选课，教师讲课等实体及其联系的 E-R 模型可用图 1.11 或 1.12 来表示。图 1.11 是将实体属性，实体与实体之间的联系全部画在一张图上，构成一个完整的 E-R 图（可称为集成法），这种画法适合规模不太大的问题。图