第 1 章

数据库系统概述

本章重点介绍数据库的有关知识，包括数据管理技术发展简史、关于数据库系统的基本概念、常见的三种数据模型的特点。要求了解数据库的三个发展阶段及各阶段的主要特点，掌握有关数据库的基本概念、数据库系统的组成及各部分的主要功能。重点掌握实体、属性定义和实体型之间的联系类型及特点。了解三种数据模型的特点及区别，为后面各章的学习打下基础。

1.1 数据管理技术发展史

数据库技术是计算机科学中发展最快的领域之一，也是应用最广的技术之一，它已成为计算机信息系统与应用系统的核心技术和重要基础。数据作为表达信息的一种量化符号，正在成为人们处理信息时重要的操作对象。数据处理是对数据进行收集、整理、排序、加工、统计和传输等一系列操作。数据处理的目的是从海量信息中，提取出有用的数据信息，作为决策依据。数据管理是指对数据的组织、编码、分类、存储、检索和维护。计算机的出现及其硬件、软件的迅速发展，加之数据库理论和技术的发展，为数据管理进入一个革命性阶段提供了有力的支持。根据数据和应用程序的相互依赖关系、数据共享以及数据的操作方式，数据管理的发展可以分为三个具有代表性的阶段，即人工管理阶段、文件系统管理阶段和数据库系统管理阶段。

1.1.1 人工管理阶段

20 世纪 50 年代中期以前，计算机硬件和软件发展处于刚刚起步阶段，没有大容量的存储设备，在进行科学计算时，必须把程序和要计算的数据通过打孔的纸带送入计算机中。

人工管理阶段数据管理具有如下特点：

（1）数据由人工保存和管理。当时计算机主要用于科学计算，对于数据保存的需求尚不迫切，没有专用的软件对数据进行管理。

（2）一组数据只能面向一个应用程序，不能实现多个程序共享数据。每个应用程序都包括数据的存储结构、存取方法、输入方式等，程序员不仅要编写程序，还要安排数据的物理存储，因此工作负担重。由于数据是面向程序的，一组数据只能对应一个程序。即使多个应用程序涉及某些相同的数据时，也必须各自定义，因此程序之间有大量的冗余数据。

（3）不同程序间不能直接交换数据，数据没有任何独立性。由于程序依赖于数据，因而如果数据的类型、格式、输入/输出方式等逻辑结构或物理结构发生变化，就必须对应用程序做出相应的修改。

图 1.1 表述了人工管理阶段应用程序与数据之间的对应关系。

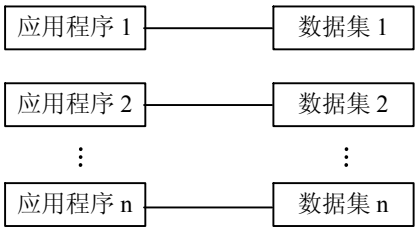

图 1.1 人工管理阶段应用程序与数据之间的对应关系

1.1.2 文件系统管理阶段

20 世纪 50 年代后期到 60 年代中期，计算机硬件的发展出现了磁带、磁鼓等直接存取设备。软件的发展为操作系统提供了文件管理系统。一个应用程序对应一组文件，不同的应用系统之间可

以经过转化程序共享数据。多个应用程序可以共享一组文件，但多个应用程序不能同时访问共享文件组，图 1.2 表述了文件系统管理阶段应用程序与数据之间的对应关系。

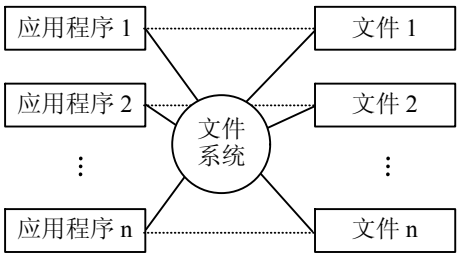

图 1.2　文件系统管理阶段应用程序与数据之间的对应关系

文件系统管理数据具有如下特点：

（1）数据可以文件形式长期保存下来。用户可随时对文件进行查询、修改和增删等处理。

（2）文件系统可对数据的存取进行管理。程序员只与文件名打交道，不必明确数据的物理存储，大大减轻了程序员的负担。

（3）文件形式多样化。有顺序文件、倒排文件、索引文件等，因而对文件的记录可顺序访问，也可随机访问，更便于存储和查找数据。

（4）程序与数据间有一定独立性。由专门的软件即文件系统进行数据管理，程序和数据间由软件提供的存取方法进行转换，数据存储发生变化不一定影响程序的运行。

与人工管理阶段相比，文件系统管理阶段对数据的管理有了很大的进步，但一些根本性问题仍没有得到彻底解决，主要表现在以下三个方面：

（1）数据冗余度大。各数据文件之间没有有机的联系，一个文件基本上对应于一个应用程序，数据不能共享。

（2）数据独立性低。数据和程序相互依赖，一旦改变数据的逻辑结构，必须修改相应的应用程序。而应用程序发生变化，如改用另一种程序设计语言来编写程序，也需要修改数据结构。

（3）数据一致性差。由于相同数据的重复存储、各自管理，在进行更新操作时，容易造成数据的不一致性。

1.1.3　数据库系统管理阶段

主要是指 20 世纪 60 年代后期以后，计算机硬件和软件又有了新的发展，硬件有了大容量的磁盘，软件出现了解决数据共享的数据库管理系统（Database Management System，DBMS）。通过数据库管理系统管理大量的数据，不仅实现了数据的永久保存，而且真正解决了数据的方便查询和一致性维护问题，并且能严格保证数据的安全。图 1.3 表述了数据库管理系统阶段应用程序与数据之间的对应关系。

用数据库系统来管理数据具有如下特点：

（1）具有面向各种应用的数据组织和结构。文件系统中，每个文件面向一个应用程序。而现实生活中，一个事物或实体，含有多方面的应用数据。例如，一个学生的全部信息，包括学生的学籍和成绩信息，还有学生健康方面的信息。这些不同的数据将对应于教务部门和健康部门的不同应用。

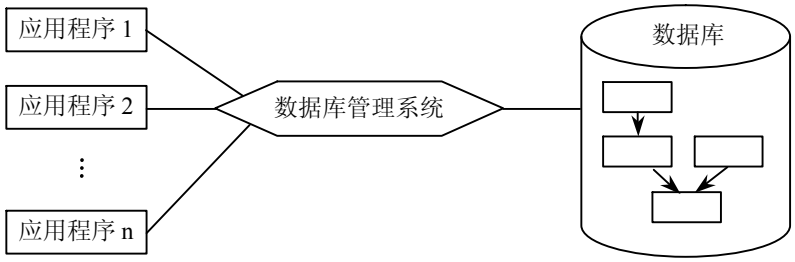

图 1.3 数据库管理系统阶段应用程序与数据之间的对应关系

对学生的全部信息，如果采用文件系统，至少要建立两个独立的文件，都要存储学生的姓名、学号、年龄、性别等学生的基本信息。如果采用数据库系统管理，在数据库设计的时候，就要考虑学生信息的各种应用。设计面向多种应用的数据结构，如学生的学籍数据、学生的健康数据等，使整个实体的多方面应用的数据具有整体的结构化描述，同时也要为数据针对不同应用的存取方式提供各种灵活性。

（2）具有高度的数据独立性。数据结构可分为数据的物理存储结构和数据的逻辑结构。数据的物理存储结构是指数据在计算机物理存储设备（硬盘）上的存储结构。在数据库中，数据在磁盘上的存储结构是由 DBMS 来管理和实现的，用户或应用程序不必关心。

数据的逻辑结构又分为局部逻辑结构和全局逻辑结构。而且不同的应用程序只与自己局部数据的逻辑结构相关。例如，教务部门只关心学生的学习成绩和选课数据，健康部门只关心学生的健康数据。

（3）实现数据的高度共享并保证数据的完整性和安全性。由数据库管理系统管理的数据可以提供多个用户或应用程序同时并发访问同一个数据库中的数据记录或同一个数据项，并要保证数据的安全性、完整性和一致性。

为确保数据库数据的正确有效和数据库系统的正常运行，数据库管理系统提供下述四个方面的数据控制功能：

（1）数据的安全性（Security）控制。防止不合法使用数据造成数据的泄露和破坏，保证数据的安全和机密。

例如，系统提供口令检查或其他手段来验证用户身份，防止非法用户使用系统；也可以对数据的存取权限进行限制，只有通过检查后才能执行相应的操作。

（2）数据的完整性（Integrity）控制。系统通过设置一些完整性规则以确保数据的正确性、有效性和相容性。

1）正确性是指数据的合法性，如年龄属于数值型数据，只能含 0，1，…，9，不能含字母或特殊符号。

2）有效性是指数据是否在其定义的有效范围内，如月份只能用 1~12 之间的正整数表示。

3）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。

（3）并发（Concurrency）控制。多用户同时存取或修改数据库时，防止相互干扰而提供给用户不正确的数据，并使数据库受到破坏。

（4）数据恢复（Recovery）。当数据库被破坏或数据不可靠时，系统有能力将数据库从错误状态恢复到最近某一时刻的正确状态。

综上所述，数据库实现了将有组织的、大量的数据长期存储在计算机内，供多用户共享，具有

最小数据冗余度和较高的数据独立性。DBMS 在数据库建立、运行和维护时对数据库进行统一控制，以保证数据的完整性、安全性，并在多用户同时使用数据时进行并发控制，在发生故障后对系统进行恢复。

1.2　数据库系统

1.2.1　数据库系统概念

数据库系统中的数据是一些已被规格化和结构化且相互关联的数据集合，这些数据中不存在有害的或无意义的冗余；数据的组织与存储结构与使用这些数据的程序相互独立；数据库中的数据可同时为多个应用程序服务；数据库中的数据定义、输入、修改和检索等所有操作均按一种公用的且可控的方式进行。一个数据库系统实际上由三部分内容组成，它们是数据库、多种应用和数据库管理系统。这三部分之间的相互关系如图 1.4 所示。

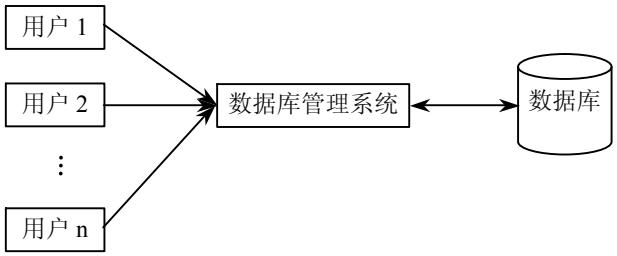

图 1.4　数据库系统组成

（1）数据库。相互关联的且具有最小冗余的数据在其中按照一定物理组织结构存放，并且从用户和数据库管理系统角度来看，这些数据又是按一定逻辑结构组织的。这种物理组织结构和逻辑组织结构在最大程度上与用户所编制的应用程序相互独立。

（2）多种应用。数据库中的数据，在数据库管理系统的控制与管理之下，可以同时为多种不同的应用程序提供服务，即可以为多个不同目的用户服务，各用户所操作使用的数据可以是相互交叉的。用户的操作方式既可以按以数据输入/输出和数据维护为主的数据流量较大的批处理方式进行；也可以按以查询为主的数据流量较小的联机处理方式进行，必要时还可以通过编程来完成对数据库中数据的各种操作。

（3）数据库管理系统。 它一方面负责对数据库中的数据进行管理和维护；一方面为用户操作数据库中的数据提供一种公用的操作方法，接收用户的操作命令，帮助完成有关的对数据库的操作并保障数据库的安全。

根据对数据库的定义以及数据库系统基本组成及作用的描述，一个数据库系统应该具有以下五个基本特点：

（1）由于数据库系统是从整体角度考虑数据的组织，因此它必须有能力描述能够反映客观事物及其相互联系的复杂数据模型，使它能够对数据本身及相互间的各种关系进行充分描述，这也是人们为什么要采用数据库系统来进行数据管理的主要原因之一。目前数据库系统共提供了四种数据模型，它们是层次数据模型、网状数据模型、关系数据模型和对象数据模型，一种类型数据库系统

通常只提供其中一种数据模型描述方法，即只支持其中一种数据模型的数据逻辑组织结构。

（2）数据库中数据的独立性。为了说明这一点，首先介绍两个概念：

1）数据在物理存储设备上的组织结构被称为数据的物理组织。

2）数据在用户或应用程序面前所表现出的组织结构被称为数据的逻辑组织。

一种数据的逻辑组织，可以采用不同的物理组织来实现，物理组织的好坏影响着系统的性能和效率。在运行阶段中，由于性能的要求或存储设备的改变，从而引起数据物理组织的改变，这种改变称为数据的再组织。用户在编制应用程序时，则是根据数据的逻辑组织对数据进行操作的。因此数据物理组织的变化，不会影响数据的逻辑组织，因而也就不会影响已有的应用程序，这种情况被称为数据的物理独立性。

数据的逻辑独立性是指当数据的逻辑组织发生变化时，如数据模型中增加了新的记录类型，某一记录类型中增加了新的数据项等，原有应用程序的执行不受影响或影响最小。数据的独立性，包括物理和逻辑的独立性，都是由数据库管理系统进行维护的。

（3）数据共享。由于数据库是从整体的角度对数据进行组织的，并在保证数据一致性的情形之下，使数据库中的数据为尽可能多的用户提供应用服务。这些用户所使用的数据可以是交叉的，即数据可以共享。如果数据不能共享，数据库中则必然会出现大量的冗余数据，这样不仅造成存储空间的浪费，更主要的是由此可能带来数据不一致的隐患。

（4）数据库系统的安全性、可靠性与完整性。一个数据库系统的可靠性体现在它的软件系统运行故障率很小以及在数据库系统由于各种意外而出现故障时，数据库中的数据损失最小；安全性是指数据库系统对其所存储的数据的保护能力，能够有效地防止数据被有意无意地泄露或篡改，控制数据的授权访问等。而数据库系统的完整性则是指在多用户操作数据情况下，数据能够保持一致性。这些特性可以从以下几个方面进行说明：

1）安全性控制。安全性控制主要指的是数据库的保密性。并不是每个用户都能够存取数据库中所有数据的，负责人和全体工作人员允许掌握的数据范围显然是有区别的，数据库系统把各用户存取数据的权利分成若干等级，如教学人员作为一个用户可以登录学生的成绩，而学生作为一个用户则仅可以查阅成绩而无法对它进行修改或删除。通过对各个用户授予不同的使用权限，以确保数据库免遭损害和被非法使用，通常采用口令密码以及数据库中数据访问授权等方法对使用者操作数据的合法权进行检验，以实现对数据库中数据安全性的保护控制。

2）完整性控制。所谓完整性包括数据的正确性、有效性和相容性。正确的数据不一定是有效的。如若用两位阿拉伯数字来表示月份，当输入 14 来代表月份时显然是无效的。数据库系统应提供尽可能多的检验措施，以确保数据库中的数据满足用户所要求的各种约束条件。

3）并发控制。在多用户操作使用数据库系统的情况下，不同用户并行地操作数据库就可能引起对数据库的干扰，从而使得数据库中的数据发生不一致的问题。如当甲用户从数据库中预定了仅剩的一张机票之后，若在数据库尚来不及将剩余的机票数改为零时，乙用户又请求订票时怎么办？显然对这种并发的操作要采取某种控制措施，最常用的方法是封锁技术，以排除和避免这种错误的发生，保证数据库中数据的操作能够正确执行。

4）故障的发现与恢复。由于数据库系统在运行过程中很难保证不产生故障、出现异议或受到破坏，而且往往这些情况发生的时间都是随机的，如断电、用户误操作等，而重建一个数据库往往要花费很大的精力和代价，有时甚至是不可能的，因此数据库系统应提供应急保护措施，一旦系统的软硬件发生故障或用户误操作导致系统异常时，系统应能够以尽量小的代价，尽快地恢复数据库

的内容和系统的正常运行。

（5）良好的人机接口与性能，任何数据库系统最终都是要和用户打交道，系统所具有的各种功能最终都需要由用户来进行操作使用。简单易学、操作简便和用户界面友好是任何一个数据库系统所必须的。此外系统的响应速度、单位时间内数据的吞吐量也是衡量数据库性能的重要指标。

1.2.2　数据库结构

在数据库技术中，为了提高数据库中数据的逻辑独立性和物理独立性，采用了分级（层）方法，将数据库中数据的组织结构划分成多个级（层）。根据美国国家标准协会（ANSI）所提出的报告，数据库的数据组织结构可以分为三个相互关联的层次，它们分别是物理层、逻辑层和视图层。

（1）物理层抽象。这是最低层次的抽象，描述数据实际上是如何存储的。物理层详细描述复杂的低层数据结构，是开发 DBMS 的数据库供应商应该研究的事情。

（2）逻辑层抽象。这是比物理层稍高层次的抽象，描述数据库中存储什么数据以及这些数据间存在什么关系。因而整个数据库可通过少量相对简单的结构来描述。虽然简单的逻辑层结构的实现涉及到复杂的物理层结构，但逻辑层的用户不必知道这种复杂性。逻辑层抽象是由数据库管理员和数据库应用开发人员使用的，它们必须确定数据库中应该保存哪些信息。

（3）视图层抽象。这是最高层次的抽象，但只描述整个数据库的某个部分。尽管在逻辑层使用了比较简单的结构，但由于数据库的规模巨大，所以仍存在一定程度的复杂性。数据库系统的多数用户并不需要关心所有的信息，而只需要访问数据库的一部分。视图抽象层的定义正是为了使用户与系统的交互更简单。系统可以为同一数据库提供多个视图，而视图又保证了数据的安全性。

如果你是一个最终用户，你根本就不关心数据存储和维护的细节。但是如果你是一个数据库管理员，那么有些细节上的东西你就必须要清楚。数据库管理系统可以为不同的用户提供不同的视图，也就使他们所看到的数据库是不一样的。这就需要进行数据抽象，以形成这些不同的视图。

根据前面讨论的数据抽象层次的不同，数据库模式又可分为：

（1）物理模式，也称为“内模式”。

（2）逻辑模式，通常简称为“模式”。

（3）子模式，也称为“外模式”。

通常，数据库管理系统支持一个物理模式、一个逻辑模式和多个子模式。

由于一个数据库是采用上述的三级模式结构方式对其中的数据组织进行描述的，从而较好地保证了数据的逻辑独立性和物理独立性，方便了用户对数据库中数据的操作使用，减少了数据冗余。这三层模式之间的相互关系如图 1.5 所示。由于数据库中数据实际上是按照物理层数据模式进行存储的，而概念层数据模式和用户层数据模式都只是对物理层数据模式描述数据的一种逐级（层）的逻辑抽象，用户在对数据库进行操作时，都必须通过数据库管理系统来完成从用户层数据模式到概念层数据模式、概念层数据模式到物理层数据模式之间这两种映射，当然这两种映射是由管理系统自动完成的，对用户是透明的。

数据库系统具有三级模式（模式、外模式、内模式）、二级映像（外模式/模式映像和模式/内模式映像）的体系结构。

模式描述的是数据的全局逻辑结构，外模式描述的是数据的局部逻辑结构。对应于同一个模式可以有任意多个外模式。对于每一个外模式，数据库系统都有一个外模式/模式映像，它定义了该外模式与模式之间的对应关系。

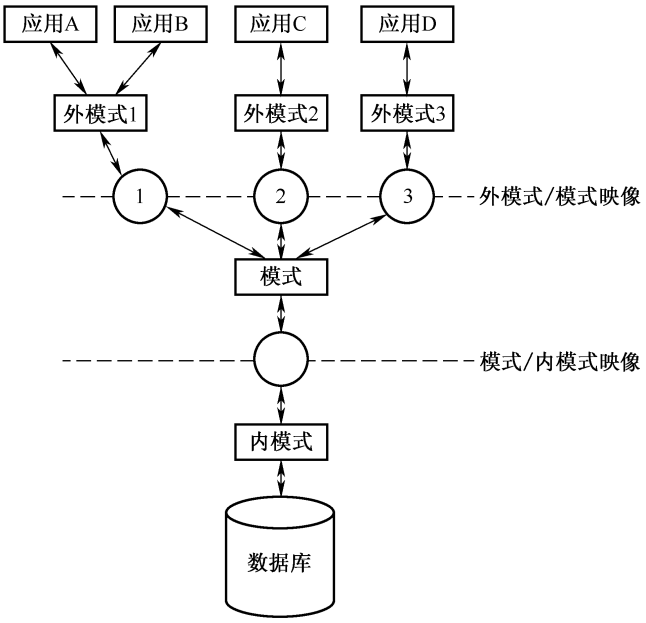

图 1.5 数据库的三级模式结构

数据库中只有一个模式，也只有一个内模式，所以模式/内模式映像是唯一的，它定义了数据全局逻辑结构与存储结构之间的对应关系。

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

数据库的内模式依赖于它的全局逻辑结构，但独立于数据库的用户视图即外模式，也独立于具体的存储设备。

数据库的外模式面向具体的应用程序，它定义在逻辑模式之上，但独立于存储模式和存储设备。

数据库系统三级模式、二级映像的体系结构是实现数据独立性的保障。使得当数据的物理存储设备更新、物理表示及存取方法改变时，数据的逻辑模式可以不改变；当数据的逻辑模式改变时，用户模式可以不改变，因此应用程序也可以不变。另外，对同一数据库的逻辑模式，可以建立不同的用户模式，从而提高数据共享性，使数据库系统有较好的可扩充性，给数据库管理员维护、改变数据库的物理存储提供了方便。

1.2.3 数据库系统

数据库系统的核心是数据库管理系统，在它的控制和帮助下，用户可以建立、使用、修改和维护数据库中的数据。数据库管理系统是建立在操作系统之上的应用软件平台。它一般具有三个主要功能。

（1）提供操作数据库的用户高级接口。

1）提供数据描述语言，供用户对整个数据库中的数据进行各种逻辑和物理组织结构描述，而这些组织结构的具体实现细节，则由 DBMS 完成，用户不必关心。

2）提供数据操纵语言，供用户对数据库中数据按照其定义逻辑组织结构进行各种操作，如插入、删除、修改和查询等，这些操作的的具体实现细节也由 DBMS 完成，用户不必关心。

3）同时还可能提供其他工具，如用户界面生成工具、报表生成工具等，帮助用户更加容易地

对数据库的操纵进行编程。

（2）管理数据库。主要包括以下功能：

1）控制整个数据库系统的运行。

2）控制用户对数据库的并发性操作。

3）执行对数据库中数据的安全、保密、有效性和完整性检验。

4）实施对数据库中数据的检索、插入、删除、修改等操作。

5）维护数据库数据组织结构的完整和一致。

（3）维护数据库。主要包括：

1）初始化时数据库数据的装入。

2）运行时记录下与用户、操作、系统状态和结果等信息的工作日志。

3）监视数据库性能，在性能变坏时，重新组织数据库。

4）在数据库系统的硬件或软件发生故障后，对数据库中受破坏的数据进行恢复。

1.2.4　数据库系统语言

数据库系统语言是用户与数据库系统进行交互操作的主要工具，是连接用户与数据库系统的桥梁。数据库语言功能的强弱直接影响到用户使用数据库系统的方便程度。数据库系统语言通常包括数据库数据描述语言（Data Description Language，DDL）和数据库数据操纵语言（Data Manipulation Language，DML）。数据描述语言用于描述数据库中各种模式的定义；而数据操纵语言则是用于描述对数据库中数据所要进行的各种操作。

1. 数据描述语言

数据描述语言是建立和使用数据库的重要工具，它是用于描述数据库各层数据模式的语言。数据库管理系统将对用户用该语言所描述的各层数据模式进行编译，产生可供数据库系统操作时所使用的目标模式。对应着数据库的模式、子模式和内模式，数据描述语言又可分为模式描述语言、子模式描述语言和内模式描述语言。它们各自的功能如下：

（1）模式描述语言。它是用来描述数据库概念层数据模式的，即用于描述数据库中所有数据以及它们间相互关系的特性。用模式描述语言写出的数据库全体数据的逻辑组织结构的全部语句的集合，通常就被称为一个模式，一个模式的主要内容有：

1）给数据库总体数据的逻辑组织结构命名，即命名模式。

2）描述模式中每个记录类型名称，以及其中各数据项的名称、数据类型（如字符型、数字型等）和数据长度等。

3）描述模式中各记录类型之间的相互联系，如果存在有相互联系的话。

一个模式仅仅是对数据库概念层逻辑数据组织结构的一个描述，并非是概念层逻辑数据本身。与其他程序语言一样，模式描述语言也有自己的一套清晰而又严格的语句和语法规则。对应不同类型的数据库系统，如层次数据库系统和网状数据库系统，它们的模式描述语言也有很大差别，即使是同一类型数据库系统，如关系数据库系统，不同软件商推出的系统，其模式描述语言也不尽相同。但有一点是一致的，这就是它们都必须是可对上面所说明的描述一个模式所需要包括的三个基本方面进行定义说明。

（2）子模式描述语言。它是用来描述数据库用户层数据模式的，即用于描述用户所使用的数据的逻辑数据组织结构的定义。用子模式描述语言写出的用户局部数据逻辑组织结构的全部语句的集合，通常就被称为一个子模式，一个子模式的主要内容有：

1）给用户使用数据库所涉及到局部数据的逻辑组织结构命名，即命名子模式。

2）描述子模式中所包含的每个记录类型及其中的各数据项，这些记录类型的名称以及各数据项的名称和长度，可以与模式中的定义有所不同，但这里主要是描述子模式中的记录类型及其数据项与模式中记录类型及其数据项之间的对应映射关系，子模式中所描述的记录及其中的数据项必须是已在模式中定义过的。

3）描述子模式中各记录类型之间的相互联系。这里同样是描述子模式中的记录间相互联系与模式中记录间相互联系之间的对应映射关系。子模式中所描述的记录间相互联系必须是已在模式中定义过的。

与模式描述语言不同，子模式描述语言有时与编写应用程序所采用的其他程序设计语言相关。

（3）内模式描述语言。它是用来描述数据库中数据在物理存储介质上的组织结构和存放方式等，它与数据库系统所运行的硬件环境特性相关。例如，系统建立了哪些物理文件？文件的存储设备是什么？文件是什么样的组织方式等，这些都是由内模式描述语言来负责描述的。

通常各软件商生产出的数据库系统，往往都要根据自己的具体实现情况，提供出相应的一整套数据描述语言的规范，其中也有一些数据库系统，对上述的数据描述语言的标准进行了一些简化，只给出一种或两种描述语言定义，或干脆将数据描述语言与数据操纵语言归并到了一起，以方便用户使用。

2. 数据操纵语言

数据操纵语言是用户操纵数据库中数据的工具，用户借助它来实现从数据库中检索数据、向数据库中添加数据、删除数据库中没有保留价值的数据或修改某些发生变化的数据等操作。数据操纵语言通常分为两种类型，即宿主式数据操纵语言和自含式数据操纵语言。

自含式数据操纵语言在数据库系统中可独立使用，是一种完整的语言，这类语言使用简单方便，很适合于在终端使用。这类语言的优点是系统运行效率较高且使用简单，缺点是它的应用范围常常受到限制，例如要提取出数据库中的一些数据进行某种复杂运算处理时，单靠数据库系统所提供的这类数据操纵语言有时就很难做到。自含式数据操纵语言通常包含以下基本操作功能：

（1）从数据库中选择满足一定要求的记录或联系。

（2）增加新的记录或联系到数据库中。

（3）修改数据库中的记录或联系。

（4）删除数据库中的记录或联系。

另一种是宿主式数据操纵语言，它不能单独使用，必须嵌入到某种程序设计语言（如 C，COBOL，FORTRAN）之中方能进行数据库操作，这种数据操纵语言语句仅负责对数据库中数据的操作，其他复杂的数据处理工作均由主语言完成，有时这样做会使得用户的应用程序变得相当复杂。由于这样的程序既包含了主语言语句，也包含了数据操纵语言语句，也就使得主语言原有的编译程序不能完全编译应用程序。有两种办法可以解决这一问题：

1）重新设计和实现一个编译程序，使之能编译包括数据操纵语言和主语言的所有语句，这种办法不大现实经济。

2）不修改主语言编译程序，而是设计一个预编程序来对应用程序中的数据操纵语言进行预编译，将其首先转换成用主语言写的程序，然后再用主语言的编译程序来编译，以产生最后的目标程序。这样做办法比较可行，目前已为许多数据库系统所采用。

实际上许多数据库系统除了提供上述两种数据操纵语言之外，还提供了许多编程工具和编程命令，以便帮助用户更加容易地编制数据库的应用程序，如用户界面生成工具、报表生成工具和数据

库应用程序接口（API）等。

1.2.5　数据库系统运行管理与控制软件

数据库系统运行管理与控制软件是数据库管理系统软件的实际组成，它主要包括语言编译处理程序、系统运行控制程序和数据库日常管理程序以及数据库工具等多种软件。下面将概述这几种软件的一些基本功能。

（1）语言编译处理程序。它主要包括：

1）数据库系统中各种数据描述语言的编译处理程序，它们的作用是将各种采用模式描述语言所定义数据模式编译成 DBMS 所使用的内部定义目标模式。

2）数据库系统各种数据操纵语言的处理程序，它们可将应用程序中采用数据操纵语言所写的数据操作语句转换成其宿主语言编译程序所能处理的语句。

3）终端操作命令解释程序，它主要用于解释终端操作命令的意义，完成相应数据库系统命令的执行过程。

4）数据库控制命令解释程序，它负责解释执行每一条数据库控制命令。

（2）系统运行控制程序。它主要包括：

1）数据库系统的总控程序，它用于检查访问的合法性，以决定一个访问是否能使用数据库。

2）并发控制程序。协调多个应用程序对数据库的操作，保证数据库中数据的一致性。

3）保密控制程序。实现对数据库数据的安全保密控制。

4）数据完整性控制程序。核对数据库完整性约束条件，以决定对数据库的操作是否有效。

5）数据库存取访问控制程序。实施对数据库中数据的操作，如执行检索、插入、修改、删除等操作。

6）通信控制程序。实现用户程序与 DBMS 之间的通信。

（3）数据库日常管理程序。这主要包括：

1）数据装入程序。实现将初始数据装入数据库。

2）系统恢复程序。当软硬件出现故障时，利用恢复程序将数据库恢复到正确状态。

3）工作日志程序。负责记载进入数据库的所有访问，其内容包括用户名称、进入系统时间、进行何种操作、数据变更情况等。使每个用户每次访问都留下踪迹。

4）性能监测程序。监测操作执行时间与存储空间占用情况，为数据库的再组织提供依据。

5）重新组织程序。当数据库系统性能变坏时，对数据库重新进行物理组织。

6）转储、编辑、打印程序。用于转储数据库的部分和全部数据，或者编辑打印数据等。

（4）数据库工具软件。它主要是为了方便建立数据库系统具体应用而提供的各种工具软件。其中有数据库系统应用程序界面制作工具、报表制作工具等许多软件工具。

1.3　数据模型

数据模型是对现实世界中各种事物或实体特征的数字化模拟和抽象，用以表示现实世界中的实体及实体之间的联系，使之能存放到计算机中，并通过计算机软件进行处理。

数据模型应能满足三个方面的要求：一是能比较真实地模拟现实世界；二是容易为人所理解；三是便于在计算机上实现。

数据模型的种类很多，最常用的数据模型有实体—联系模型、关系数据模型、层次数据模型和网状数据模型。

1.3.1 实体—联系模型

实体—联系数据模型，即 E-R（Entity-Relationship）数据模型，是 P. P. Chen 于 1976 年首先提出的。它是这样认识现实世界的：现实世界是由一组称作实体的基本对象以及这些对象间的联系构成的。实体是现实世界中可区别于其他对象的一个“事件”或一个“物体”。例如，学生是一个实体，学校开设的课程也是一个实体。数据库中实体通过属性集合来描述。例如，学号和性别属性描述了某个特定的学生。联系是实体间的相互关联。例如选修联系将学生和他选修的课程相关联。

现实世界中，实体之间的联系是多种多样的，实体间的各种联系按照联系所涉及的实体可以划分为两类：一类是不同的实体集中实体之间的联系，另一类是相同的实体集内实体之间的联系。这两种联系都有三种不同的联系情况：

（1）一对一联系。如果实体集 A 中每个实体至多和实体集 B 中的一个实体有联系，反之亦然，就称实体集 A 和实体集 B 的联系为“一对一联系”，记为“1:1”。

例如，班级和班主任之间的联系，一个班级只有一个班主任，一个班主任只能管理一个班级。

（2）一对多联系。如果实体集 A 中每个实体与实体集 B 中的任意多个（零个或多个）实体有联系，而 B 中每个实体至多与实体集 A 中的一个实体有联系，就称实体集 A 对 B 的联系为“一对多联系”，记为“1:N”。

例如，一所学校有多个教师，每个教师只能属于一所学校。

（3）多对多联系。如果实体集 A 中的每个实体与实体集 B 中的任意个（零个或多个）实体有联系，反之，实体集 B 中的每个实体与实体集 A 中的任意个（零个或多个）实体有联系，就称实体集 A 和 B 的联系为“多对多联系”，记为“M:N”。

例如，学生和课程之间的联系，一个学生可以选修多门课程，每门课程有多个学生选修。

实体—联系模型常用实体—联系图（简称 E-R 图）表示，在 E-R 图中，使用的符号如下：

（1）矩形框表示实体类型，单线矩形框表示强实体类型，双线矩形框表示弱实体类型。

（2）菱形表示联系类型。

（3）椭圆框表示属性。

（4）用连线表示实体或实体类型之间的联系。每条连线上附加一对数表示对参与联系的每个角色的约束，即在相关类型的联系中，一个实体作为一个给定的角色参与到该联系中的最大可能性。

E-R 模型支持一对一、一对多和多对多的联系。实体集之间三种联系的表示如图 1.6 所示。

实体—联系方法是抽象和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 所支持的逻辑数据模型，它是各种数据模型的共同基础，因而比逻辑数据模型更一般、更抽象、更接近现实世界。E-R 图的简单化和形象化使得它被广泛使用。

例如，假设学生、班级、课程、教师、参考书五个实体型分别具有下列属性：

学生：学号、姓名、性别、年龄

班级：班级编号、所属专业系

课程：课程号、课程名、学分

教师：职工号、姓名、性别、年龄、职称

参考书：书号、书名、内容提要、价格

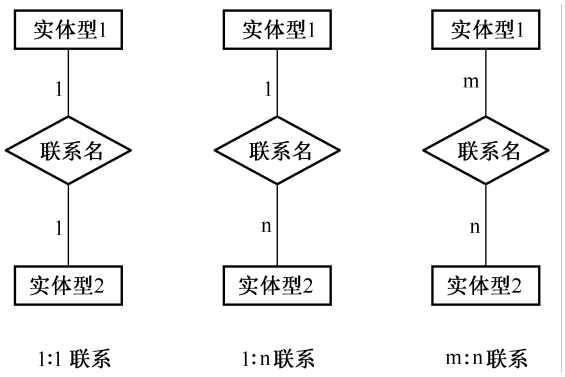

图 1.6　两个实体集之间的三种联系

学生、班级、课程、教师、参考书五个实体属性及其联系的 E-R 图如图 1.7 所示。

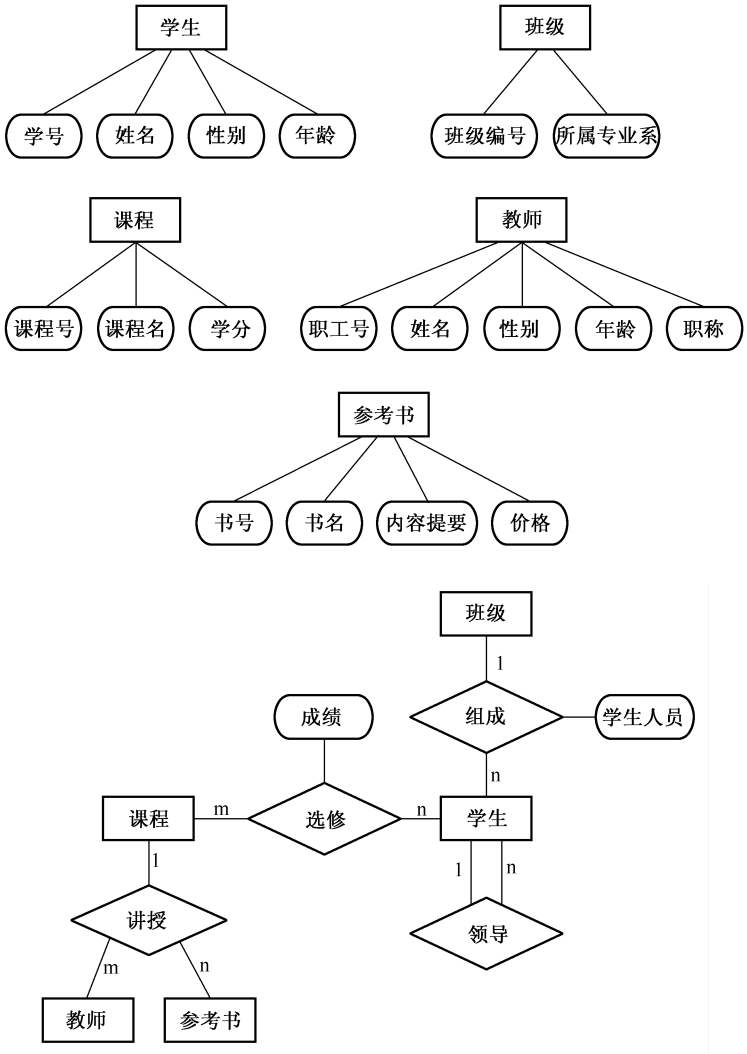

图 1.7　用 E-R 图表示学生、班级、课程、教师、参考书五个实体属性及其联系

1.3.2 关系模型

1. 关系

关系模型是目前最重要的一种数据模型。关系数据库系统采用关系模型作为数据的组织方式。一个关系模型的逻辑结构是一张二维表，它由行和列组成。例如，学生的关系可描述为：

学生（学号，姓名，性别，系别，年龄，籍贯），相应的关系模型如表 1.1 所示。

表 1.1 学生关系模型

学号	姓名	性别	系别	年龄	籍贯
06001	李刚	男	计算机科学	18	江苏
06002	肖明	男	电子科学	19	湖南
06003	王晓平	女	计算机科学	18	湖北
06004	张辉	男	数学	20	浙江
06025	杨洋	女	物理	19	湖北

关系是一种规范化了的二维表中行的集合，为了使相应的数据操作简化，在关系模型中，对关系作了种种限制，关系具有如下特性：

（1）关系中不允许出现相同的元组（即行）。因为数学上集合中没有相同的元素，而关系是元组的集合，所以作为集合元素的元组应该是唯一的。

（2）关系中元组的顺序（即行序）是无关紧要的，在一个关系中可以任意交换两行的次序。因为集合中的元素是无序的，所以作为集合元素的元组也是无序的。根据关系的这个性质，可以改变元组的顺序使其具有某种排序，然后按照顺序查询数据，可以提高查询速度。

（3）关系中属性的顺序是无关紧要的，即列的顺序可以任意交换。交换时，应连同属性名一起交换，否则将得到不同的关系。

（4）同一属性名下的各个属性值必须来自同一个域，是同一类型的数据。

（5）关系中各个属性必须有不同的名字，不同的属性可来自同一个域，即它们的分量可以取自同一个域。

（6）关系中每一分量必须是不可分的数据项，或者说所有属性值都是原子的，即是一个确定的值，而不是值的集合。属性值可以为空值，表示“未知”或“不可使用”，即不可“表中有表”。

满足上述条件的关系称为规范化关系，否则称为非规范化关系。表 1.2 就是非规范化的关系。

表 1.2 不规范的关系

学号	姓名	性别	系别	年龄	籍贯	成绩			
						英语	数学	…	数据库
95001	李勇	男	计算机科学	20	江苏	83.0	78.0	…	90.0
95002	刘晨	女	信息	19	山东	77.0	78.0	…	85.0
95003	王名	女	数学	18	北京	80.0	90.0	…	79.0
95004	张力	男	计算机科学	19	北京	80.0	90.0	…	79.0
…	…	…	…	…	…	…	…	…	…
95700	杨晓冬	男	物理	21	山西	88.0	92.5	…	95.0

2. 与关系有关的概念

（1）关系。对应通常说的表。如学生关系模型。

（2）元组。表中的一行即为一个元组。

（3）属性。表中的一列即为一个属性。如学生关系模型有 6 列，对应 6 个属性（学号、姓名、性别、系别、年龄、籍贯）。

（4）超码（Superkey）。指能够唯一地标识一个实体的一个或多个属性的集合。全部属性的集合总是能够组成一个超码。

（5）候选码（Candidate Key）。包含属性最少的超码。一个实体集可能有多个候选码。

特别要注意，候选码同时具有两个重要性质：

1）唯一性（Uniqueness）。关系 R 的任意两个不同元组，其候选码的值是不同的。

2）最小性（Minimally）。候选码的属性集（Ai，Aj，…，Ak）中，任一属性都不能被删掉，否则将破坏唯一性。

（6）主码（Primary Key）。从候选码中选出，用于在实体集中区分不同实体。如学生关系模型中的学号。

主码是关系模型中的一个重要概念。每个关系必须选择一个主码，选定以后，不能随意改变。每个关系必定有且仅有一个主码，通常用较小的属性组合作为主码。

（7）主属性。把包含在任何一个后选码中的属性叫做主属性（码属性）。

（8）非主属性。不包含在任何候选码中的属性叫做非主属性（非码属性）。

（9）域（Domain）。属性的取值范围。如人的年龄一般在 1～150 岁之间。

（10）分量。元组中的一个属性值。

（11）关系模式。对关系的描述，一般表示为：

关系名（属性 1，属性 2，…，属性 n）

如：学生（学号，姓名，性别，系别，年龄，籍贯）（下划线标注的属性集合为主码）

课程（课程号，课程名，学分）

（12）联系的属性与码。在关系模型中，实体以及实体间的联系都是用关系来表示的。联系在转化为关系时，除了在关系的属性集合中加入该联系自身的属性外，还应加入与该联系相关连的实体的主码集合，并将该集合作为该关系的主码。

例如，学生与课程之间的多对多的选修联系在关系模型中可以表示如下的关系：

选修（学号，课程号，成绩）

3. 关系数据模型的操纵与完整性约束

关系数据模型的操纵主要包括查询、插入、删除和更新数据。这些操作必须满足关系的完整性约束条件。关系的完整性约束条件包括三大类：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（Entity Integrity）规则。若属性 A 是基本关系 R 的主属性，则主属性 A 不能取空值。

例如，学生（学号，姓名，性别，系别，年龄，籍贯）关系中，学号不能为空。

选修（学号，课程号，成绩）关系中学号、课程号均不能为空。

（2）参照完整性（Referential Integrity）规则。外码（Foreign Key）的概念：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 K_s 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（Referencing Relation），基本关系 S 为被参照关系

（Referenced Relation）或目标关系（Target Relation）。

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 K_s 相对应（基本关系 R 和 S 不一定是不同的关系），则对于 R 中每个元组在 F 上的值必须或取空值（F 的每个属性值均为空值）；或等于 S 中某个元组的主码对应值。

例如：学生（学号，姓名，性别，专业号，年龄）

课程（课程号，课程名，学分）

选修（学号，课程号，成绩）

在选修关系中的学号必须或取空值，或等于学生关系中某个元组的学号值；选修关系中的课程号必须或取空值，或等于课程关系中某个元组的课程号值。

（3）用户定义的完整性（User-Defined Integrity）。完整性即合理性。例如，学生的年龄应为大于零、小于 100 的整数。

4. 关系数据模型的存储结构

表以文件形式存储，每一个表通常对应一种文件结构。

5. 关系数据模型的优缺点

关系数据库以关系模型作为数据的组织方式，关系模型是建立在严格的数学概念基础上的，关系数据库的主要优点是概念简单清晰，用户无须了解复杂的存取路径，无须说明“怎么干”，只需说明“干什么”，易懂易学。关系模型的查询效率往往不如非关系数据模型。

1.3.3 层次模型

层次模型是数据库系统中最早出现的数据模型，层次数据库系统的典型代表是 IBM 公司的 IMS（Information Management System）数据库管理系统，这是 1968 年 IBM 公司推出的世界上第一个大型的商用数据库管理系统（DBMS）产品。

层次模型用树型（层次）结构来表示各类实体及实体间的联系。现实世界中许多实体之间的联系本来就呈现出一种自然的层次关系，如行政机构、家族关系等。因此层次模型可自然地表达数据间具有层次规律的分类关系、概括关系、部分关系等，但在结构上有一定的局限性。

在数据库中定义满足下面两个条件的基本层次联系的集合为层次模型：

（1）有且只有一个结点没有双亲结点，这个结点称为根结点。

（2）根以外的其他结点有且只有一个双亲结点。

在层次模型中，每个结点表示一个记录类型，记录类型之间的联系用结点之间的连线表示，这种联系是父子之间的一对多的联系。这就使得层次数据库系统只能处理一对多的实体联系。

例如，用于道路管理的层次模型如图 1.8 所示。

对层次模型数据库的操作主要有查询、插入、删除和修改。进行插入操作的时候，如果没有双亲结点就不能插入子女结点的值。进行删除操作的时候，如果要删除双亲结点值，则相应的子女结点值也同时被删除。当进行修改操作的时候，应当修改所有需要修改的记录，保证数据的一致性。

因此，层次数据库不仅要存储数据本身，还要存储数据间的层次联系。数据间的层次联系用指针实现。

层次数据模型的优点：它本身比较简单，易于实现；对于实体间的联系固定，且预先定义好的应用系统，采用层次数据模型实现性能会较好。

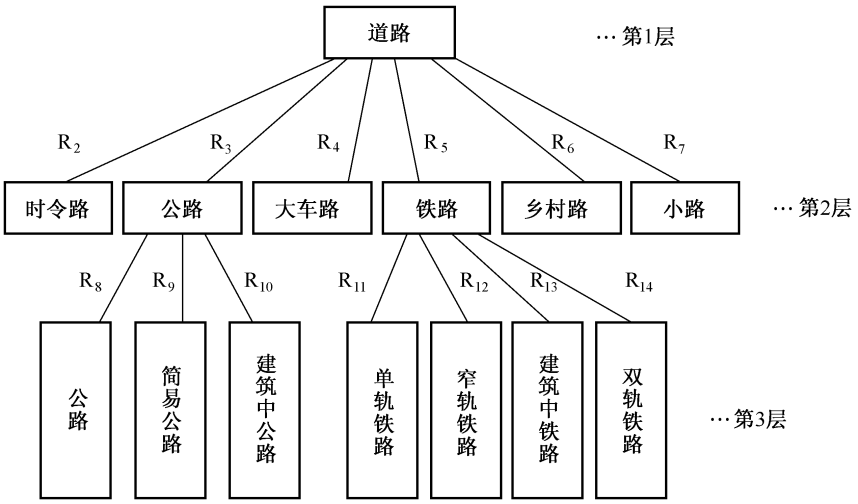

图 1.8　道路层次模型示例

层次数据模型的不足：它支持的联系类型少，只适合支持一对多的联系；对数据的插入和删除操作有较多限制，层次数据模型中对子女结点的存取操作必须通过对祖先结点的遍历才能进行。

层次数据模型的一个基本特点：记录之间的联系通过指针实现。任何一个给定的记录值只有按其路径查看才能显示它的全部意义，没有一个子女记录值能够脱离双亲记录值而独立存在。

1.3.4　网状模型

在现实世界中，事物之间按其联系来看，更多的是非层次的，用层次模型表示非树形结构是很不直接的，采用网状模型则可以克服这一弊病。因为网状模型取消了层次模型的限制，从树的结构变为了图的结构。网状模型中的数据用记录的集合来表示，数据间的联系用指针实现。数据库中的记录可被组织成任意图的集合。

在数据库中，把满足以下两个条件的基本层次联系集合称为网状模型：

（1）允许一个以上的结点无双亲。

（2）一个结点可以有多于一个的双亲。

网状模型是一种比层次模型更具普遍性的结构，它去掉了层次模型的两个限制，允许多个结点没有双亲结点，允许结点有多个双亲结点，此外它还允许两个结点之间有多种联系（称之为复合联系）。因此网状模型可以更直接地去描述现实世界。而层次模型实际上是网状模型的一个特例。

例如，用于人口管理的网状模型如图 1.9 所示。

网状模型存储结构的关键在于实现记录之间的联系，网状模型中记录之间的联系用指针链表实现。

网状模型的优点：它能够更直接地描述现实世界，如一个结点可以有多个双亲结点；存储结构具有良好的导航性能，存取效率较高。

网状模型的缺点：结构比较复杂，而且不能直接处理多对多的关系，必须要把多对多的关系分解为多个一对多的关系才能进行处理。因此，随着应用环境的扩大，数据库的结构就变得越来越复杂，不利于最终用户掌握。

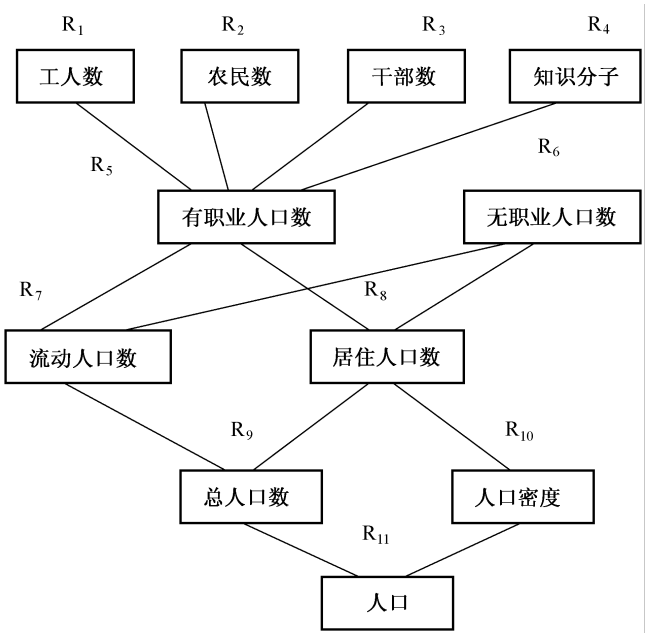

图 1.9　人口网状模型示例

网状数据模型的一个基本特点：由于记录之间的联系是通过存取路径实现的，应用程序在访问数据时必须选择适当的存取路径，因此，用户必须了解系统存储结构的细节，加重了编写应用程序的负担。

思考题与习题

1.1　从程序和数据之间的关系分析文件系统和数据库系统之间的区别和联系。
1.2　使用数据库系统有什么好处？
1.3　什么是数据库的数据独立性？
1.4　什么是数据库管理系统？
1.5　数据库管理系统有哪些功能？
1.6　叙述模型、模式和具体值三者之间的联系和区别。
1.7　简要叙述关系数据库的优点。