

第 1 章 数据库基础知识

数据库技术产生于 20 世纪 60 年代中期，它的出现极大地促进了计算机的应用向各行各业的渗透。对于一个国家来讲，数据库的建设规模、数据库信息量的大小和使用频率已成为衡量该国信息化程度的重要指标。目前，各种管理信息系统（MIS）、办公自动化（OA）和决策支持系统（DDS）等系统的核心都离不开数据库技术的支持。因此，掌握数据库的基础知识、了解数据库管理系统的特性、熟悉数据库管理系统的操作是非常重要的。

本章主要介绍数据库的基础知识，包括数据处理、数据库技术、数据描述与数据模型、关系数据库与关系数据库管理系统等。

1.1 数据管理技术概述

早期的计算机主要用于科学计算，尽管计算过程异常复杂，但往往计算对象（数据）比较简单；之后当计算机应用于数据管理时，数据量急剧增加，其中非数值数据所占比例较大，而非数值数据比数值数据复杂得多，不仅要知道各项数据本身的内容，而且还需要知道它们之间的关系。为了有效地管理和利用这些数据，产生了**数据库技术**。

1.1.1 数据、信息与数据处理

数据、信息与数据管理是数据库技术中常涉及的基本概念，它们贯穿于本课程的始终。因此，应正确理解它们的内涵，掌握它们之间的联系与区别。

1. 数据与信息

提起数据，大多数人的第一反应是数学中能够进行加、减、乘、除运算的数字。其实，数字只是一种最常见、最简单的数据，这种理解是对数据的一种传统和狭隘的理解。

按照国际标准化组织（ISO）的定义，数据是对客观事物特征所进行的一种抽象化、符号化的表示。例如，某位同学身高 1.78 米、体重 65 千克、出生于 1986 年 7 月 18 日，这里的 1.78、65、1986 年 7 月 18 日等都是数据，它们描述了该位同学的某些基本特征。

从计算机的角度出发，凡是为了描述客观事物而用到的数字、字符、图形、图像以及声音等能输入到计算机中并能被计算机处理的都可以看成是数据，即凡是能够被计算机处理的对象都称为**数据**。

总之，数据是一种物理符号序列，用来记录事物的属性或状态，用“型”和“值”来表征。型是指数据的类型，不同数据类型，记录事物的性质也不相同。例如，用于表示成绩的数 120，表示得分的多少，其类型是数值型，而房间号 120 并不表示某一个量的大小，仅仅是房间的标号，数据处理中往往将其类型归为字符型。

目前，信息是一个被广泛使用，同时又非常模糊的概念。多数学者认为，**信息**是对原始数据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，数据是信息的载体，而信息是数据中所包含的意义。

当我们把数据输入到计算机中以后，目的并不仅仅是把这些数据原封不动地再取出来，而是想要计算机对这些数据进行处理，为我们提供有用的、新的信息。

例如，某个班级的每位同学的数学期中考试成绩输入并保存在计算机中，这是原始数据，然后教师经过计算、查询等处理得到平均分、最高分和最低分以及优秀率等数据，这些数据反映了该班级数学课程学习情况的信息。

信息是有价值的，其价值取决于它的准确性、及时性、完整性和可靠性。为了提高信息的价值，应采用科学的方法来管理信息，常用的方法就是数据库技术。

2. 数据处理

数据处理又称为信息处理，是指利用计算机将各种类型的数据转换成信息的过程，包括数据的采集、整理、存储、分类、排序、加工、检索、维护、统计、传输等一系列处理过程，其目的就是要从大量的、原始的、杂乱的数据中获得人们所需要的资料并抽取有用的数据成分。

数据处理与数值计算相比，具有计算相对简单、数据量大、数据间逻辑关系复杂等特点。例如，建筑工程预算根据施工图纸的有关数据，参照与之相关的建材价格表等有关数据，计算出工程的费用。预算得出的各项费用便是数据经数据处理后得到的信息，可作为投资者是否投资的依据。

1.1.2 数据管理技术的发展

数据管理是指对数据进行组织、存储、分类、检索和维护等操作，它是数据处理的核心。随着计算机硬件和软件技术的发展，数据管理手段和技术也在不断发展。到目前为止，计算机数据管理技术的发展大致经历了人工管理、文件系统和数据库系统三个阶段。

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。硬件方面，没有磁盘等直接存储设备；软件方面，没有专门的管理数据软件。进行数据处理时，数据与程序往往结合在一起，

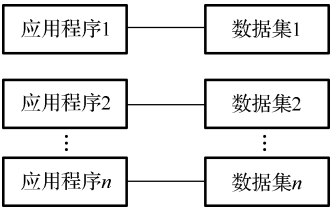


图 1-1 人工管理阶段程序与数据的关系

如图 1-1 所示。

人工管理阶段有以下特点：

- (1) 数据不保存。由于当时计算机主要用于科学计算，一般不需要将数据长期保存，因此只是在计算某一问题时才将数据输入，用完就撤走。
- (2) 程序管理数据。数据主要由程序自己管理，程序设计时不仅要考虑数据的逻辑结构，而且要设计其物理结构，包括存储结构、存取方式、输入方式，因而程序员的负担很重。
- (3) 数据不能共享。数据共享是指多个用户、多个应用程序能够同时或不相互影响地使用同一个数据集。在人工管理阶段，一个程序对应一组数据。即使多个程序涉及相同的数据，也必须各自定义，无法相互利用和参照，造成了程序之间有大量的数据冗余（重复）。
- (4) 数据不具有独立性。当数据的逻辑结构或物理结构发生变化时，必须对应用程序做相应的修改。

2. 文件系统阶段

20 世纪 50 年代后期至 20 世纪 60 年代中期，计算机的硬件和软件系统都有了很大发展。硬件方面出现了可以直接存储的存储器，软件方面出现了高级语言和操作系统。计算机开始大量地用于数据处理工作。这时的数据处理过程中，程序和数据可以分别存储为程序文件和数据文件，因而程序与数据不再是一体，应用程序与数据之间的对应关系如图 1-2 所示。程序员可以把精力集中在数据处理的算法上，而不必再去花更多的时间去考虑数据存储的具体细节。

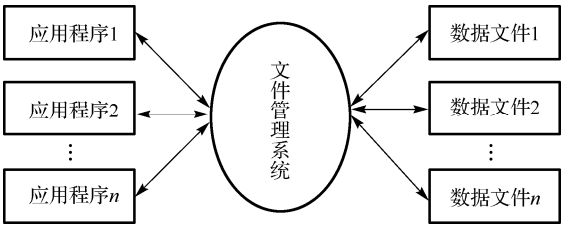


图 1-2 文件系统中程序与数据的关系

- 计算机数据处理的文件系统阶段虽然比人工管理阶段有了很大的进步，但仍有以下缺点：
- (1) 程序与数据间的相互依赖性。数据文件只是为了专门的需要设计的，仅供某一特定应用程序使用。
 - (2) 数据的冗余性。由于文件之间缺乏联系，造成每个处理程序都有相应的文件，有可能同样的数据在多个文件中重复存储。
 - (3) 数据的不一致性。这往往是由数据的冗余造成的，相同的数据在不同文件中存储时，如果处理程序没有同步更新，就会使同样的数据在不同的文件中不一致。
 - (4) 数据的无关性。数据文件无集中管理，各个文件无统一的管理机制，文件之间相互独立，无法相互联系。

3. 数据库系统阶段

从 20 世纪 60 年代后期开始，计算机广泛应用于企业管理，数据量急剧增加，数据管理的规模越来越大，数据共享的要求也越来越高，文件管理远远满足不了人们的需求。为适应多用户、多个应用程序共享大量数据的需要，出现了统一管理数据的专门软件系统，即**数据库管理系统**。数据库管理系统克服了以往所有数据管理方式的缺点，提供了一种更高效的数据管理方式，如图 1-3 所示。

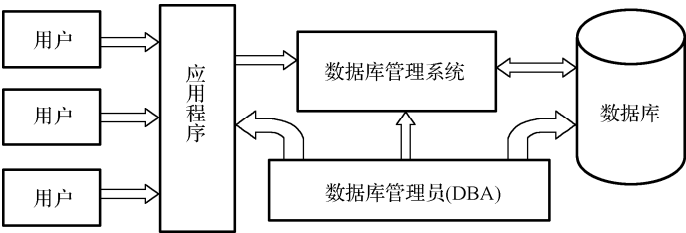


图 1-3 数据库系统中用户、程序与数据之间的关系

数据库系统能对所有的数据进行集中统一的管理，使数据存储独立于使用的应用程序，以实现数据的共享，大大消除了数据冗余，节省了存储空间。

数据库系统管理数据的特点如下：

(1) 数据共享性高，冗余度低，易扩充。在数据库系统阶段，数据不再分属于各个应用程序而是集中存放在数据库中面向整个系统。因此，数据可以被多个用户、多个应用程序共享，减少了不必要的数据冗余，节约了存储空间，还能避免数据之间的不相容性与不一致性。

(2) 数据结构化。在数据库系统中，数据是按照某种数据模型组织的，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。因此，整个应用系统的数据不是孤立的，可方便地表示出数据之间的有机联系。

(3) 提高了数据的独立性。数据的独立性指的是数据与应用程序之间的依赖关系。在数据库系统中，用户的应用程序与数据的逻辑结构及物理存储方式无关，相互独立，不因一方的改变而使另外一方也改变，大大减少了应用程序的修改与维护。

(4) 实现了数据的统一控制。数据库系统中的数据是各用户的共享资源，允许多个用户同时访问，往往还提供必要的保护措施。由于数据库系统提供了相应的各种控制功能，从而确保了数据的并发控制、安全性和完整性。并发控制可以防止多个用户同时访问数据时产生的数据不一致，安全性可以防止非法用户存取数据，完整性可以保证数据的正确性和有效性。

1.2 数据库技术的基本概念

数据库技术是为满足数据管理任务的需要而产生的一种新技术，研究内容主要是如何科学地组织和存储数据，如何高效地获取和处理数据。数据库技术的基本概念有数据库、数据库管理系统、数据库系统和数据库系统的体系结构等。

1.2.1 数据库

通俗地讲，**数据库** (DataBase, DB) 是存放数据的仓库，可以把数据库简单地定义为某一组织或部门为满足其数据处理需要而按一定方式存储在计算机中的相关数据的集合。数据库中的数据并不是杂乱无章的、分散的、孤立的，而是按照某种数据模型组织起来的，存储在计算机外存储器中，可供多个用户共享，具有最小的冗余度和较高的数据独立性与易扩展性。数据库不仅包含数据本身，而且包括相关数据之间的联系。

数据库中数据的存储、管理与使用是通过数据库管理系统软件来实现的。

1.2.2 数据库管理系统

数据库管理系统 (DataBase Management System, DBMS) 是用来帮助用户在计算机上建立、使用和维护数据库而开发的软件系统，是数据库系统的核心。数据库管理系统是在操作系统的支持下工作的，提供了一系列对数据库进行统一的管理和控制功能，能有效地组织、存储、获取和管理数据库中的数据，接收和完成用户、程序发出的各种访问数据的请求。

1. 数据库管理系统的功能

数据库管理系统软件有很多，但无论哪种数据库管理系统软件，都应该提供以下几种基本功能：

(1) 数据定义功能。DBMS 提供数据定义语言 (Data Definition Language, 简称 DDL)，

使用户能够用它来定义数据库的结构，描述数据及数据之间的联系，建立、修改或删除数据库。

例如，关系数据库管理系统的标准语言 SQL，就提供了 CREATE、ALTER 和 DROP 等命令，分别用来创建、修改和删除关系数据库的二维表的结构。

- (2) 数据操作功能。DBMS 提供数据操纵语言 (Data Manipulation Language, 简称 DML)，实现对数据库中数据的增加、修改、删除和检索等操作。
- (3) 控制和管理功能。DBMS 提供数据控制语言 (Data Control Language, 简称 DCL)，用于实现对数据库的控制和管理，以确保数据资源的安全。

2. 常用的数据库管理系统

数据库管理系统软件经过几十年的发展，取得了辉煌的成就，产生了巨大的经济效益。目前，在各种计算机软件中，数据库管理系统占有极为重要的地位。根据管理数据的规模和应用场合，数据库管理系统软件可分为如下两类：

- (1) 大型数据库管理系统。如 DB2、Oracle、Sybase 和 Informix 等，大型数据库管理系统软件需要专人管理和维护，性能比较强，一般应用于大型数据处理场所，如飞机订票系统、银行系统、通信系统等。
- (2) 小型数据库管理系统。主要在小型服务器或微型计算机上运行，例如 SQL Server、Visual FoxPro、Access 等，其中微型机数据库管理系统 (Visual FoxPro、Access) 功能相对简单，集数据定义和数据操作于一体，因其容易掌握，使用也比较方便，目前被广泛地使用。

1.2.3 数据库系统

数据库系统 (DataBase System, 简称 DBS) 是指安装和使用数据库技术的计算机系统。数据库系统实际上是一个具有管理数据库功能的计算机软/硬件综合系统，能够实现有组织地、动态地存储大量的相关数据，具备相应的数据处理与共享手段，为用户提供数据访问和查询服务。数据库系统通常由硬件系统、数据库、数据库管理系统、应用系统、数据库管理员 (DataBase Administrator) 和数据库的终端用户 6 部分组成，如图 1-4 所示。

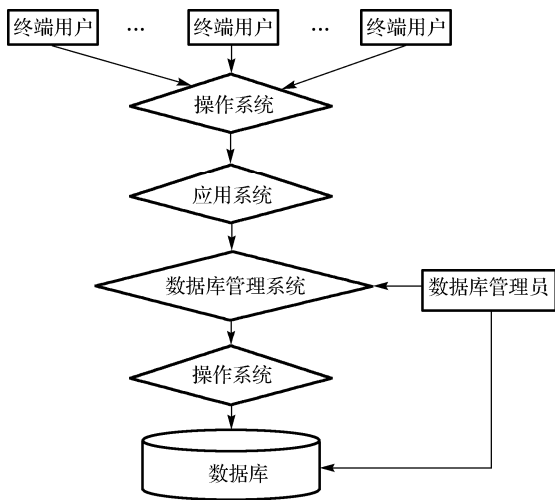


图 1-4 数据库系统的组成

硬件系统是指存储数据库和运行数据库系统的硬件资源。与计算机的其他应用相比，数据库系统的计算机硬件往往要求有容量足够大的内存和外存，用来运行操作系统、数据库管理系统、应用程序及存储数据。另外，对于网络数据库系统，则还需要有相应的网络设备的支持。数据库管理系统软件是数据库系统的核心，帮助用户创建、维护和使用数据库。一个数据库系统若能够稳定运行，还必须有相应人员的支持才行，这些人员包括数据库管理员和用户。负责数据库的建立、维护和管理工作的人员是数据库管理员；使用系统的用户为最终用户或终端用户。

1.2.4 数据库体系结构

为了更加有效地管理数据，提高数据库的逻辑独立性和物理独立性，人们需要为数据库设计一个严谨的体系结构。ANSI（美国国家标准学会）所提出的三层体系结构建议，得到了数据库领域的广泛支持，成为目前公认的**数据库体系结构标准**。

ANSI 将数据库体系结构分为三层，为不同级别的用户提供了三种不同的视图模式。所谓的视图，是指用户观察、认识和理解数据的范围、角度和方法，即数据库在用户“眼中”的反映。这种分层的结构，可以使不同级别的用户“看到”不一样的数据库。在标准的数据库体系结构中，三种不同的视图模式分别称为内模式、模式和外模式，如图 1-5 所示。

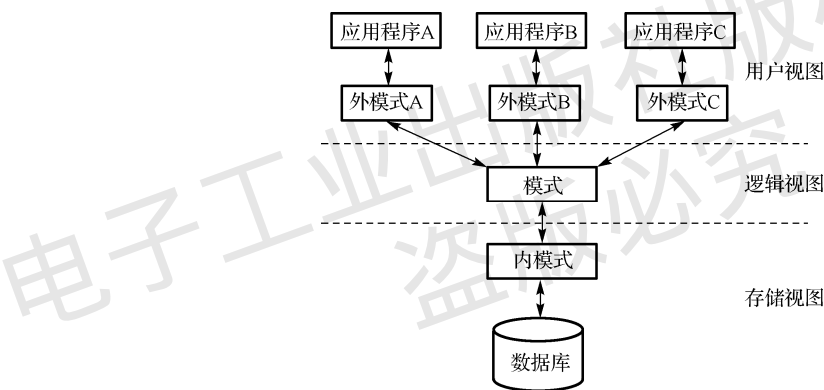


图 1-5 数据库体系结构

1. 内模式

内模式为用户提供了数据库的存储视图，又称为存储模式或物理模式，它与计算机硬件直接进行交互。内模式是对数据库中全体数据的内部描述，反映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器中的数据库。

2. 模式

模式为用户提供了数据库的逻辑视图，又称为逻辑模式或概念模式。模式是由数据库设计人员综合所有的用户数据，按照统一的方式构造的全局逻辑结构。它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，以全局的观念反映了数据库中的数据。

3. 外模式

外模式为用户提供了数据库的用户视图，又称为子模式，是从模式中导出的一个子集。

外模式包含了模式中允许特定用户使用的部分数据，是与某一应用程序相关数据的逻辑表示。外模式为应用程序提供了数据库访问接口，允许应用程序和数据库中的相关数据建立联系。

数据库体系结构中三级模式的划分，是在用户级、概念级和物理级三个层次上的对数据库的抽象表示。虽然只有物理数据库是实际存在的，但概念数据库提供了对物理数据库的逻辑描述，而用户数据库提供了访问物理数据库的接口，这样用户就能够简单地管理数据，而不必关心数据在计算机中的物理表示和存储。

1.2.5 数据库系统结构分类

开发一个数据库系统，往往首先根据其业务需求和使用环境确定数据库系统的体系结构，以便选择相应的数据库管理系统和开发工具。目前，常见的数据库系统体系结构有如下几种。

1. 单用户数据库系统

单用户数据库系统是指数据库、数据库管理系统、应用系统等都安装在同一台计算机上，某时刻只能由单个用户独占地使用数据库中的数据，不同的计算机之间不能共享数据。这是一种最简单的数据库系统。

2. 主从式结构

主从式体系结构的数据库系统是指将数据库、数据库管理系统和数据库应用程序安装在主机上，终端只作为主机的输入输出设备，多个终端用户可使用主机上的数据和程序，如图 1-6 所示。在这种结构中，所有处理任务都由主机完成，用户终端本身没有应用程序和数据。当终端用户数目增加到一定程度时，主机任务会过分繁重，响应缓慢，从而使系统性能大幅度降低。

3. 客户/服务器体系结构（Client/Server，C/S）

C/S 结构是目前流行的数据库系统体系结构之一，如图 1-7 所示。

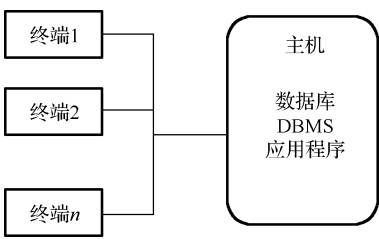


图 1-6 主从式结构

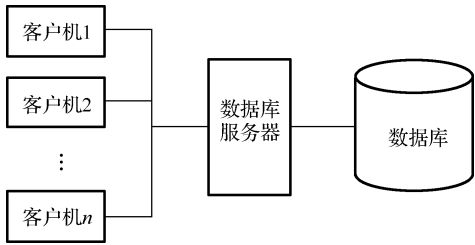


图 1-7 客户/服务器结构

网络上的节点服务器存放数据及执行 DBMS 功能，客户机 DBMS 应用开发工具和应用程序。客户端的用户请求被传送到服务器，服务器处理后，只是将结果返回给用户，因而显著减少了网络上数据的传输量，提高了系统性能。这样不仅可以实现对数据库资源的共享，而且还极大地提高了数据库的安全性能。

4. Web 浏览器/服务器数据库系统（B/S）

随着 Internet 技术的发展，出现了 Web 数据库，随之也产生了 Web 浏览器/服务器体系结构的数据库系统，B/S 体系结构如图 1-8 所示。

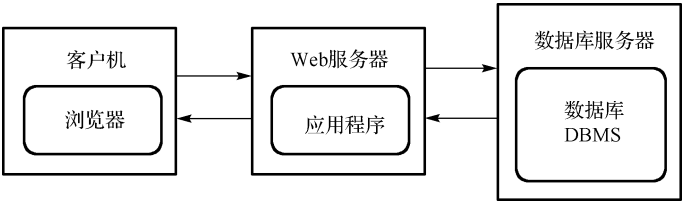


图 1-8 B/S 结构

在 Web 浏览器/服务器数据库系统体系结构中，客户端仅安装浏览器软件，用户通过 URL 向 Web 服务器发出请求，Web 服务器运行脚本程序，向数据库服务器发出请求。数据库服务器执行处理后，将结果返回给 Web 服务器。Web 服务器根据结果产生网页文件，客户端收到网页文件后，将其在浏览器中显示出来。

1.3 数据模型

数据库中的数据源于现实世界的客观事物。由于计算机不能处理这些具体的事物，因此需要把现实世界中的客观事物抽象为能用计算机存储和处理的数据，由现实世界转换为信息世界。

1.3.1 数据模型的概念与分类

数据模型是对现实世界数据特征的抽象，是用于抽象表示、处理现实世界的数据与信息的工具。

1. 数据模型的概念

利用计算机处理现实世界的具体事物时，通常需要对客观事物加以抽象，提取其主要特征，归纳形成一个简单清晰的轮廓，将复杂的问题变得简单且易于处理，这就是建立数据模型的目的。

现实世界中，客观存在着的事物都有一些特征，人们正是利用这些特征来区分事物的，同时各种事物虽然千差万别，但它们之间是相互联系的，而且这种联系有时是比较复杂的。例如，图书馆中的图书与读者之间、高校中教师、学生和课程之间的联系。由于计算机不可能直接处理现实世界中的具体事物，因此必须把现实世界中的具体事物转换成计算机能够处理的对象。在数据库中，这项工作借助于数据模型这一工具来抽象完成的。

在数据库中，**数据模型**是用于提供信息表示和操作手段的形式框架，也是将现实世界转换为数据世界的桥梁，如图 1-9 所示。



图 1-9 数据模型的应用层次

2. 数据模型的分类

在数据库技术中，数据模型按不同的应用层次可划分为概念数据模型（简称概念模型）和逻辑数据模型（也称数据模型）两大类。



(1) 概念模型

概念数据模型是客观世界在人们头脑中的反映,是对客观事物及其联系的一种抽象,用来描述某个特定组织所关心的信息结构,完全独立于计算机系统,并不涉及信息在计算机中如何表示。概念模型是一种直接面向客观世界、面向用户的模型,是用户与数据库程序设计人员进行交流的工具。

(2) 逻辑模型

数据库中的数据是按一定的逻辑结构存放的,这种逻辑结构通常也是用数据模型来表示的,这种模型即逻辑模型。逻辑模型描述了客观事物在计算机数据库中的存储形式以及相互之间的联系,是机器世界的数据库模型。

1.3.2 实体-联系模型

在概念模型中,最著名的是“实体-联系模型”,即 E-R 模型。E-R 模型,是目前用于描述客观世界并建立概念模型的一种常用的抽象方法,并得到了广泛、成功的应用。

1. 概念模型的相关术语

设计概念模型的目的在于有效地模拟现实世界,并不涉及其在计算机中是如何实现的,通常只包含那些对描述现实世界具有普遍意义的抽象概念。概念模型涉及的术语如下。

(1) 实体与实体集

客观存在并且可以相互区别的“事物”称为**实体**。实体可以是可触及的对象,如一个学生、一本书、一台电脑等,也可以是抽象的事件,如一次旅游、一次订货会等。

另外,**实体集**则是同类型实体的集合。例如,一个班所有的学生可以看成是一个学生实体集,而全校的所有学生可以视为一个更大的学生实体集。

(2) 属性

实体通常具有若干特征,这些特征称为**属性**。例如,职工都有“编号”、“姓名”、“出生日期”和“性别”等属性。

属性有“型”和“值”之分。“型”即属性名,如“姓名”和“性别”等都是属性名;“值”即属性的具体内容,如(10001, 李长江, 1971 年 4 月, 男)。这些属性值的集合则表示了一个职工实体。

属性值的取值范围称为该属性的**域**。如职工编号属性的域为 6 个数字符号,性别属性值只能为“男”或“女”等。

(3) 关键字

若某一个属性或几个属性组合的值能够在一个实体集中唯一标识其中每一个实体,则该属性或属性的组合就可以选作**关键字**。例如,职工编号就是职工实体的关键字,而职工实体的姓名属性可能有重名,不能作为职工实体的关键字。

2. 实体集间的联系

实体间的联系通常是指实体集之间的联系。例如,学生与老师实体集之间存在着“授课”联系;学生实体集与课程之间存在“选课”的联系。两个实体集之间的联系可以分为三类:

(1) 一对一联系 (1:1)

实体集 A 中的一个实体至多与实体集 B 中的一个实体相对应,反之,实体集 B 中的一

个实体至多与实体集 A 中的一个实体相对应。例如，学校与校长，观众与座位，病人与病床。

(2) 一对多联系 (1:n)

实体集 A 中的一个实体与实体集 B 中的多个实体相对应，而实体集 B 中的一个实体至多与实体集 A 中的一个实体相对应。例如，班级与学生，公司与员工。

(3) 多对多联系 (m:n)

实体集 A 中的一个实体与实体集 B 中的多个实体相对应，并且实体集 B 中的一个实体也与实体集 A 中的多个实体相对应。例如，一家公司可以经营多种业务，而同一种业务可以被多家公司所经营，因此公司与业务实体之间就存在着多对多的联系。

3. E-R 图

E-R 模型采用图形描述了实体、属性和联系三个要素，具体表达时采用了图形和相关的文字说明来表达，因此也称为 E-R 图。

具体的作图方法是：用矩形框表示实体，并在框内写上实体的名称；用菱形框表示实体间的联系，用线段连接菱形框与矩形，并在线段上注明联系的类型（一对一、一对多、多对多）；用椭圆表示实体的属性，并在框内标明属性的名称。

如图 1-10 所示，用 E-R 图描述了实体之间的三种联系。

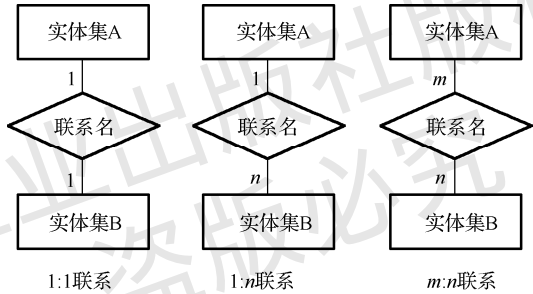


图 1-10 实体之间的三种联系

图 1-11 给出了高校学生实体集与选修课实体集之间的 E-R 图。

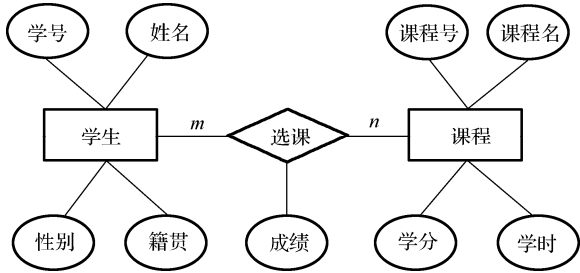


图 1-11 学生实体集与选修课实体集之间的 E-R 模型

目前，E-R 图已经广泛应用于数据库的概念设计中，通过 E-R 图，数据库程序设计人员与其他人员可以进行交流与合作，合理地描述出某一数据处理对象的各个实体及其联系，并将其作为下一步数据库设计的依据。

1.3.3 数据模型

数据模型（逻辑模型）是从数据库角度所看到的数据的组织形式，用图或表的形式描述计算机中数据的逻辑结构。在数据库技术的发展历程中，出现过许多种数据模型，其中最常用的有以下三种。

1. 层次模型

层次模型是最早用于数据库技术的一种数据模型，它按照层次结构的形式来组织数据，利用树型结构表示实体及其之间的联系，该模型体现实体间的一对多联系。图 1-12 是一个层次模型的例子，表示大学、学院、系之间的联系。

2. 网状模型

网状模型是以网状结构表示实体及其之间联系的模型。网状模型体现实体间的多对多联系，具有很大的灵活性。图 1-13 是一个网状模型的例子，表示学校中教师、课程和学生之间的联系。

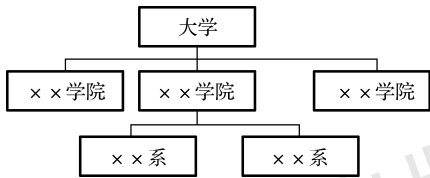


图 1-12 层次模型

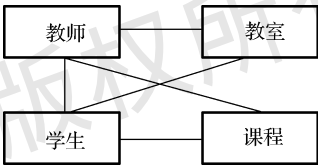


图 1-13 网状模型

3. 关系模型

关系模型是最重要的一种数据模型。1970 年，IBM 公司的 E.F.Codd 首次提出数据库系统的关系模型，开创了关系数据库研究的先河。关系模型有严格的数学基础，抽象级别较高，而且简单清晰，便于理解和使用。目前应用的绝大多数数据库管理系统都是基于关系模型的。

(1) 关系模型

在关系数据库系统中，关系模型由若干二维表格组成，数据及其之间的联系均存放于这些二维表格中。每个表格用于描述一个实体（同类个体的集合），实体由若干个数据项（简称属性）组成，关系模型是以常见的二维表格为基本存储结构，利用二维表格表示实体及其之间联系的模型，表 1-1 就是关系模型数据集合的例子。

表 1-1 学生表

学 号	姓 名	性 别	专 业	出生日期
200401101	王肇博	男	涉外会计	1978-01-25
200402101	马京辉	男	计算数学	1978-12-23
200402102	李尘寰	女	计算数学	1977-9-24
⋮	⋮	⋮	⋮	⋮
200403006	章文晋	男	国际贸易	1979-06-09

(2) 关系的特征

关系数据库中，关系必须规范化，也就是一张二维表若能够成为一个关系模型表示的数据集合必须满足如下几点：

- ① 每一列的数据其数据类型必须相同，即列是同质的，或者来自同一个域。例如，表 1-1 中的“出生日期”列必须都是日期类型数据。
- ② 字段名（列名）必须唯一，即同一个关系中，不能有相同的字段名。
- ③ 同一个关系中，不能有内容完全相同的数据行。
- ④ 一个关系中行或列的次序可以任意交换，不影响关系的实际含义。
- ⑤ 每一列必须是最小的数据项，不能再分，即表中不能包含表，如表 1-2 所示的工资表就不是关系。

表 1-2 工资表

编 号	姓 名	应发工资			扣 税
		基本工资	职务工资	津 贴	
0001	张三	1200	178	238	123
0002	李四	1086	154	365	260
0003	王五	1478	210	434	391

数据模型决定了数据库中数据定义、数据组织、数据操作和数据约束。任何一个数据库管理系统都是基于某种数据模型的，即根据不同的数据模型开发出不同的数据库管理系统。采用上述模型之一构建的数据库则分别称为层次数据库、网状数据库和关系数据库。Visual FoxPro 则是针对关系型数据库而开发的数据库管理系统。

1.4 关系数据库

1970 年，美国 IBM 公司的 E.F.Codd 提出了关系数据模型与一系列数据库方法和理论，为关系数据库技术奠定了理论基础。自 20 世纪 80 年代以来，关系数据库技术发展十分迅速，涌现出许多优秀的商品化的关系数据库管理系统，如著名的 Oracle、DB2、SQL Server 和 Visual FoxPro 等。

1.4.1 关系数据库的基本概念

关系数据模型的最大特点是描述的一致性，既可用表格来表示实体，又可用表格形式来表示实体与实体间的联系。在关系数据库中，涉及的基本概念有关系、记录、字段、关键字、主码以及外码等基本概念。

1. 字段/属性（Field）

关系中的一列称为一个**字段**，或称为一个**属性**。一个关系中往往有多个字段，为了标记不同的字段，引出了字段名或属性名。

例如，表 1-1 学生关系中的姓名、性别、出生日期和专业等都是字段名。

2. 记录/元组（Record）

关系中的一行称为一个**记录**，或称为一个**元组**。实际上，一条记录往往是用于描述同一实体相关属性的属性值的集合。

例如，表 1-1 中的第一行记录（200401101，王肇博，男，涉外会计，1978-01-25）即描述了一位同学的相关信息。

3. 表 (Table)

若干条记录的集合构成了表。表往往由两部分组成，一是表的结构（包含哪些字段），二是表中包含的一行行数据记录。在 Visual FoxPro 中，一个关系被称为一个表 (Table)，对应于一个存储在磁盘上的扩展名为 .DBF 的表文件。

4. 关键字 (Key)

关键字是指某一属性或属性的组合，它能唯一地标识一个记录。关键字又可进一步分为主关键字与候选关键字。若一个关系有多个关键字，选定其中的一个作为当前标识记录的依据，则该关键字为主关键字（主码），其他关键字则为候选关键字。

5. 外码 (Foreign Key)

若某个字段不是本表的主码，但它却是另外一个表的主码，则称这样的字段为表的外部码，简称外码。

例如，图 1-14 所示的“员工”表和“部门”表中，由于员工的“编号”字段可以唯一地标识一个记录，因此它可以作为“员工”表的关键字，同时也可以指定为主关键字，“部门代码”字段在“员工”表中不能作为关键字或主码，但却是“部门”表中的主码，所以“部门代码”字段是“员工”表的外码。

员工					部门		
编号	姓名	性别	部门代码	基本工资	部门代码	部门名称	部门电话
020101	张家瑞	男	01	4500	01	产品研发部	85056995
111103	刘凯政	男	11	2400	02	人力资源部	85052263
041103	孙玉柱	男	11	2335	11	市场营销部	85054708
090102	曹雪慧	女	01	3260			
060203	赵燕华	男	02	2900			
070201	张璇	女	02	2190			

图 1-14 外码

6. 关系模式

对关系的描述称为关系模式。关系模式可以用表的结构来描述，也可以用“关系名（属性名 1，属性名 2，…，属性名 n）”的形式来描述。例如，图 1-14 中的员工表的关系模式可以表示成：

员工（编号，姓名，性别，部门代码，基本工资）

1.4.2 关系的运算

关系数据库中，经常需要对关系进行各种运算，以满足对数据库的各种操作处理的需要。关系运算的对象是关系，运算结果仍然是关系。最基本的关系运算有三种：选择、投影和连接。

1. 选择

选择运算是指从一个关系中选出满足给定条件记录（元组）的操作。选择是从行的角度进行的筛选操作，选择操作的结果会构成原关系中的一个子集，如图 1-15 所示，从表中选择所有性别为“男”的记录。

学号	姓名	性别	专业	出生日期
200401101	王肇博	男	涉外会计	1978-01-25
200402101	马京辉	男	计算数学	1978-12-23
200403010	刘俨如	女	国际贸易	1979-05-16
200401102	张小兵	男	涉外会计	1978-04-12
200402103	吕飞	男	计算数学	1978-09-10
200402102	李尘寰	女	计算数学	1977-9-24
200403006	章文晋	男	国际贸易	1979-06-09

选择

学号	姓名	性别	专业	出生日期
200401101	王肇博	男	涉外会计	1978-01-25
200402101	马京辉	男	计算数学	1978-12-23
200401102	张小兵	男	涉外会计	1978-04-12
200402103	吕飞	男	计算数学	1978-09-10
200403006	章文晋	男	国际贸易	1979-06-09

图 1-15 选择运算

2. 投影

投影运算是指从一个关系中选取若干字段（属性）的操作。投影是从字段的角度进行的运算，结果仍然是一个关系，如图 1-16 所示。

学号	姓名	性别	专业	出生日期
200401101	王肇博	男	涉外会计	1978-01-25
200402101	马京辉	男	计算数学	1978-12-23
200403010	刘俨如	女	国际贸易	1979-05-16
200401102	张小兵	男	涉外会计	1978-04-12
200402103	吕飞	男	计算数学	1978-09-10
200402102	李尘寰	女	计算数学	1977-9-24
200403006	章文晋	男	国际贸易	1979-06-09

投影

学号	姓名	专业
200401101	王肇博	涉外会计
200402101	马京辉	计算数学
200403010	刘俨如	国际贸易
200401102	张小兵	涉外会计
200402103	吕飞	计算数学
200402102	李尘寰	计算数学
200403006	章文晋	国际贸易

图 1-16 投影运算

3. 连接

连接运算是指把两个关系中的记录按一定条件横向结合，生成一个新的关系。最常用的连接运算是自然连接，即利用两个关系中的公共字段，把两个关系中该字段相等的记录连接起来，如图 1-17 所示。

职工编号	姓名	部门
0001	张三	生产部
0002	李四	技术部
0003	王五	销售部
0004	马六	生产部

职工编号	基本工资	津贴	扣所得税	实发工资
0002	1086	365	260	1345
0001	1200	238	123	1493
0004	1378	410	360	1626
0003	1478	434	391	1731

相同职工编号连接

职工编号	姓名	部门	基本工资	津贴	扣所得税	实发工资
0001	张三	生产部	1200	238	123	1315
0002	李四	技术部	1086	365	260	1191
0003	王五	销售部	1478	424	391	1521
0004	马六	生产部	1378	410	360	1428

图 1-17 连接运算

1.4.3 关系的完整性

关系的完整性是指关系中的数据以及与其相关联的数据必须遵循的约束条件或限制规则，以确保数据的正确性、有效性和相容性，防止错误的数据库。关系的完整性通常包括实体完整性、域完整性和参照完整性。

1. 实体完整性

实体完整性是为了保证表中记录唯一的特性，每个表中应该有一个主关键字，主关键字值不允许有重复值且不能取空值。例如，“学生”表中，应该设置“学号”属性为主关键字，用来唯一地标识每位学生。

2. 域完整性

域完整性是指关系中的属性值必须满足某种特定的数据类型和约束规则，即限定某个属性的取值类型和取值范围。域完整性主要包括字段的有效性约束和记录的有效性约束两个方面。例如，“学生”表中，“性别”字段的取值只能是“男”或“女”，某门课考试成绩必须在 0~100 分之间等。

3. 参照完整性

参照完整性指定了关系之间建立联系的约束规则。例如，两个一对多的关系中，在“多”的一方，外码字段允许有重复值，但是要求外码字段的值在“一”方一定存在。例如，图 1-14 中，如果“员工”（多方）一条记录的“部门代码”值为“02”，而“部门”表（一方或父表）中无“部门代码”值为“02”的记录，在这种情况下，关系就不满足参照完整性。

1.4.4 关系数据库示例

关系数据库是由若干张表组成的集合，即关系数据库中至少应有一张表。实际上，关系数据库通常由若干张表有机地组合在一起，每张表由表结构和数据组成，若表之间存在联系，则联系也作为数据库的组成部分一并保存在数据库中，以满足某种应用系统的需要。

在 Visual FoxPro 中，与数据库对应的是数据库文件（.DBC 文件），一个数据库文件包含若干张表（.DBF 文件），而每张表则由表结构与若干数据记录组成。例如，Xsgl.DBC 数据库中包含 STUDENT.DBF、COURSE.DBF 和 SCORE.DBF 共 3 张表，分别用于保存学生的基本信息、课程信息和成绩信息，分别如图 1-18、图 1-19 与图 1-20 所示。

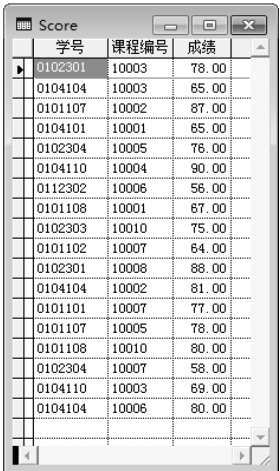
Student									
学号	姓名	性别	专业	出生日期	入学成绩	是否党员	籍贯	简历	照片
0112301	刘长瑜	女	国际贸易	05/06/79	530	F	福建	memo	gen
0101102	张林青	男	英语	04/12/78	512	F	浙江	memo	gen
0102303	吕天昊	男	应用数学	09/10/78	522	T	山东	memo	gen
0102302	李静涵	女	应用数学	09/24/77	535	F	山东	memo	gen
0112302	张文龙	男	国际贸易	06/09/79	541	F	福建	memo	gen
0101108	潘璋	男	英语	11/08/78	601	F	山东	memo	gen
0104101	康采儿	女	应用化学	04/03/77	589	T	山东	memo	gen
0104104	郭树甜	女	应用化学	05/16/78	521	F	浙江	memo	gen
0102304	隋东亮	男	应用数学	07/23/77	544	T	江西	memo	gen

图 1-18 Student 表



课程编号	课程名称	学分	任课教师
10001	大学英语	3.0	许红霞
10003	C语言程序设计	3.0	张功成
10002	哲学	3.0	李天明
10004	大学语文	2.5	万彭
10005	经济学导论	2.0	李天明
10006	数据库原理与应用	3.0	鲁岩
10007	现代教育论	2.0	王英凡
10008	Delphi程序设计	3.0	张守楠
10009	普通物理	4.0	钱英
10010	科学社会概论	2.5	邢景东

图 1-19 Course 表



学号	课程编号	成绩
0102301	10003	78.00
0104104	10003	65.00
0101107	10002	87.00
0104101	10001	65.00
0102304	10005	76.00
0104110	10004	90.00
0112302	10006	58.00
0101108	10001	67.00
0102303	10010	75.00
0101102	10007	64.00
0102301	10008	88.00
0104104	10002	81.00
0101101	10007	77.00
0101107	10005	78.00
0101108	10010	80.00
0102304	10007	58.00
0104110	10003	69.00
0104104	10006	80.00

图 1-20 Score 表

另外，Xsgl 数据库中的 3 张表之间并不是孤立的，而是存在着有机的联系，这种联系在 Visual FoxPro 数据库设计器中如图 1-21 所示，STUDENT.DBF 与 COURSE.DBF 之间存在着一对多的联系，COURSE.DBF 和 SCORE.DBF 之间也存在着一对多的联系。

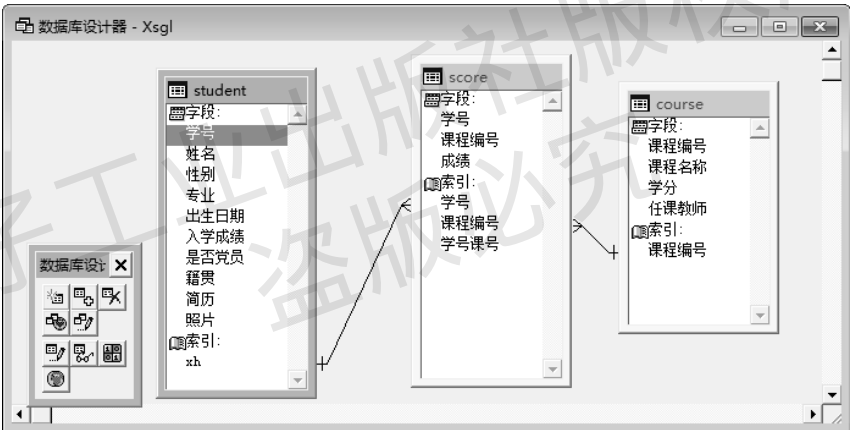


图 1-21 表之间的联系