

# 第5章

## 数据库概述

程序处理的对象是大量的数据,数据又是如何组织和存储的呢?在信息社会,信息系统越来越显现出重要性,数据库技术作为信息系统的基础与核心技术更加引人注目。数据库技术是计算机应用技术中的一个重要组成部分,是数据管理的技术,它所研究的问题是如何科学地组织和存储数据,使对大量数据的管理比用文件管理具有更高的效率。

### 5.1 数据库的发展

数据库系统已经融入人们的日常工作和生活之中,扮演着非常重要的角色。例如,一个消费者去书店购买图书,就仿佛处在一个数据库系统之中,购买图书的过程就是访问数据库的过程。再如,现在在网上购物的人越来越多,每天浏览各种购物网站的顾客不计其数,人们所知道的是浏览器的功能非常强大,但是却忽略了其强大原因是基于后台数据库的支持。大家平时所看到的购物记录以及与购物相关的信息都是以调用数据库来实现的。还有很多资源管理、人事管理、客户管理等都可以通过数据库系统来建立并记录数据,进行统计分析。数据库系统可以带来很高的效率,为企业节约成本,提高企业整体运行效率。

#### 5.1.1 数据与信息的关系

数据与信息有多种解释。一般而言,数据是对客观事物描述与记载的物理符号,而信息则是数据的集合、含义与解释,是事物变化、相互作用、特征的反映。例如,当对一个企业当前各类生产经营指标(即数据)分析时,就可以得出该企业生产经营状况的若干信息。

数据库技术是信息系统的一个核心技术。它研究如何组织和存储数据、如何高效地获取和处理数据。

##### 1. 数据

数据是人们用来反映客观世界而记录下来的可以鉴别的数字、字母、符号、图形、声音、图像、视频信号。它不仅指狭义上的数字,还可以是文字、图形和声音等,也是客观事物的属性、数量、位置及其相互关系等的抽象表示。例如,“0、1、2”以及阴、雨、下降、气温等都是数据。

数据按运算的特性可分为数值型数据和非数值型数据。数值型数据以数字表示,可以进行算术运算;非数值型数据以字符(含数字)等表示,不能进行算术运算。例如,字符、文字、图表、图形、图像、声音、视频等均属于非数值型数据。

##### 2. 信息

信息是人类社会最重要的战略资源之一,20世纪40年代后期建立起来的信息科学已经对科学的发展产生了广泛而深远的影响。信息一般通过数据形式来表示,而计算机能够实际

处理的就是各种各样的数据。

信息是对原始数据进行加工或解释之后得到的、对客观世界产生影响的数据。

### 3. 信息与数据的关系

信息与数据既有联系,又有区别。数据是信息的载体,而信息是数据的内涵,信息加载在数据之上,是数据的语义解释。信息依赖数据来表达,数据则生动具体地表现出信息。数据是符号,是物理性;信息则是对数据进行加工处理之后所得到的,并对决策产生影响的数据,是逻辑性的。

例如,对于一幅计算机中 Microsoft Office 的图像,数据就是彩色位图点阵,信息就是 Microsoft Office 产品 Logo。再如,在运输货物时,对司机来说,运输单就是信息,因为司机可以从运输单上知道收货地址和收件人等信息。而对于负责经营的管理者来说,运输单只是数据,因为一张运输单只提供了这一单的数据,并不能了解本月的整体经营情况。

## 5.1.2 数据管理技术的发展

计算机的发展历史表明,应用领域的不断拓展与深化是推动计算机学科发展的内在动力,数据管理技术也是如此。随着计算机硬件、软件技术的发展和应用领域的改变,数据管理经历了由低级到高级的发展过程,数据管理技术的发展可以大体归为 3 个阶段:人工管理阶段、文件系统阶段和数据库系统阶段。

### 1. 人工管理阶段

20 世纪 50 年代以前,计算机主要应用于科学计算。由于计算机软硬件技术发展水平的限制,外部存储设备只有卡片、纸带和磁带等硬件,也没有专门的数据管理软件和操作系统,计算机的主要任务是数值计算。数据的管理者就是程序的设计者和使用者,数据和程序编写在一起,每个程序都有自己的数据,程序之间无法进行数据共享,数据完全依赖于程序,数据冗余度大。人工管理阶段的数据管理具有以下 3 个特点。

(1) 应用程序直接管理数据。程序和数不可分割,一起输入、输出,数据不保存。计算机系统不提供用户数据的管理功能,应用程序中只包含自己要用到的全部数据。用户编写程序时,必须全面考虑好相关的数据,包括数据的定义、存储结构以及存取方法等。程序和数是一个不可分割的整体,数据脱离了程序就无任何存在的价值。

(2) 数据缺乏独立性。数据属于应用程序,应用程序要规定数据的物理结构和存取方式,修改数据必须修改程序。由于数据与程序是一个整体,因此数据只能为本程序所使用。数据只有与相应的程序一起保存才有价值,否则就毫无用处。所以,所有程序的数据均不能单独保存。

(3) 数据无法共享。一组数据对应一个应用程序,多个程序涉及部分相同数据时,无法互相参照、利用,存在大量的冗余。不同的程序均有各自的数据,这些数据对不同的程序通常是

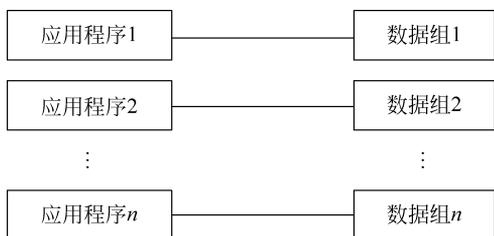


图 5-1 人工管理阶段应用程序与数据的关系

不相同的,也不可共享。即使不同的程序使用了相同的一组数据,这些数据也不能共享,程序中仍然需要各自加入这组数据。基于这种数据的不可共享性,必然导致程序与程序之间存在大量的重复数据,浪费存储空间。

另外,人工管理阶段数据的输入输出方式、存取方式等都由程序设计者自行设计,人工管理阶段应用程序与数据的关系如图 5-1 所示。

## 2. 文件系统阶段

20世纪50年代后期,计算机开始大量地应用于数据处理工作。计算机不仅用于科学计算,也大量应用于企事业单位的管理,数据管理进入文件系统阶段。用户通过操作系统对文件进行打开、读写、关闭等操作,既可批处理,又可联机实时处理。文件系统阶段的数据管理具有以下3个特点。

(1) 数据由文件系统管理。不同应用程序可以共享一组数据,实现了数据以文件为单位的共享。文件系统利用“按文件名访问,按记录进行存取”的管理技术,可对文件进行修改、插入和删除操作。

(2) 数据和程序间具有独立性。数据可以重复使用,不再专属于某一特定程序。而应用程序的变化,如使用不同的高级语言编写应用程序,也将引起数据文件结构的改变。

(3) 数据共享性差、冗余度依旧很高。程序和数据依然相互依赖,相同数据重复存储,使修改和维护工作进行起来仍然比较困难。文件系统仍然是面向应用的,当数据完全相同时,可通过同一数据文件共享,但当不同的应用程序具有部分相同的数据时,仍必须建立各自的数据文件,而不能共享相同的数据。

综上所述,文件系统仍然是无弹性的、无结构的数据集合,即数据文件之间是孤立的,不能反映现实世界事物之间的内在联系。文件系统管理阶段应用程序与数据的关系如图5-2所示。

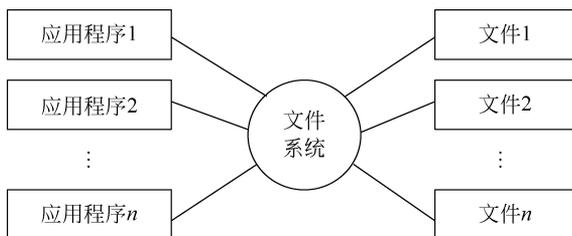


图 5-2 文件系统管理阶段应用程序与数据的关系

## 3. 数据库系统阶段

20世纪60年代后期,进入集成电路计算机时代。计算机磁盘存储技术取得重大进展,大容量和快速存取的磁盘相继投入市场,为新型数据管理技术的开发提供了良好的物质基础。此外,随着计算机应用于管理的规模的不断扩大和数据量的急剧增长,联机实时处理的要求日渐迫切。文件系统作为数据管理的手段已不能满足用户的需求。为了满足多用户、多应用共享数据的要求,使数据为尽可能多的应用服务,数据库技术随之出现。数据库技术是数据管理的最新技术,是计算机科学的重要分支,它的出现极大地促进了计算机应用向各行各业的渗透。数据库系统克服了文件管理方式的缺陷,具有以下4个特点。

(1) 具有很高的数据独立性。数据独立于应用程序,减少了程序维护和修改的工作量。在处理数据时,用户所面对的是简单的逻辑结构,而不涉及具体的物理存储结构。数据的存储和使用数据的程序彼此独立,数据存储结构的变化应尽量不影响用户程序的使用,用户程序的修改也不要求数据结构做较大的改动。

(2) 数据结构化。数据按照数据模型来组织,实现数据整体结构化。数据的存取单位可以小到一个数据项,大到一组记录。数据反映了客观事物间的本质联系,而不是着眼于面向某个应用,是结构化的数据。这是数据库系统的主要特征之一,是与文件系统的根本区别。

(3) 对数据进行统一管理。数据所有操作均由数据库管理系统统一控制和管理。通过设

置用户的使用权限,防止数据被非法使用。可采用完整性检验,以确保数据符合某些规则,保证数据库中的数据始终是正确的。

(4) 数据共享性高、冗余度低、易扩充。数据不再面向某个应用程序,而是面向整个系统。一个用户可面对的数据资源是多样化的,一部分数据资源可被多种需求的用户访问。

数据库系统管理阶段应用程序与数据的关系如图 5-3 所示。

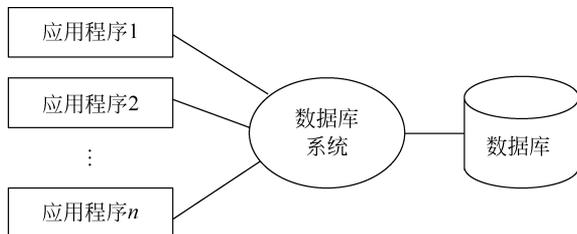


图 5-3 数据库系统管理阶段应用程序与数据的关系

数据库系统经历了 3 个发展阶段。

第一代数据库系统。20 世纪 70 年代,以层次数据库和网状数据库为代表,第一代数据库系统得到广泛应用。它们基本实现了数据管理中的“集中控制与数据共享”这一目标。

第二代数据库系统。20 世纪 80 年代出现了以关系数据库为代表的第二代数据库系统。如 Oracle 等关系数据库系统已广泛用于大型信息管理系统。

第三代数据库系统。20 世纪 80 年代末到 90 年代初,新一代数据库技术的研究和开发已成为数据库领域学术界和工业界的研究热点,关系数据库已成为数据库技术的主流,如多媒体数据库、时态数据库、空间数据库、面向对象数据库、分布式数据库、并行数据库系统、数据仓库、移动数据库、XML 数据管理技术等。

进入 21 世纪以后,随着市场需求的日益增加和技术条件的逐渐成熟,出现了对象数据库、网络数据库、嵌入式数据库等技术。

## 5.2 数据库系统与体系结构

数据库系统是为适应数据处理需要而发展起来的一种较为理想的数据处理的核心机构。为理解数据库系统,需要对数据库系统的结构有足够的了解。数据库系统的体系结构是数据库系统的一个总的框架。

### 5.2.1 数据库系统

数据库系统(Database System, DBS)是指在计算机系统中引入数据库后构成的计算机应用系统,包括数据库、数据库管理系统、数据库管理员、软件(主要包括操作系统、各种宿主语言、实用程序)。用户可以通过应用程序系统的用户接口使用数据库。数据库系统的结构如图 5-4 所示。

#### 1. 数据库

数据库(Database, DB)是指长期存储在计算机内,有组织的、统一管理的相关数据的集合。数据库能为各种用户所共享,具有较小的冗余度,数据间联系紧密,但数据又有较高的独立性等。数据库的特点如下所述。

(1) 数据库是相互关联的数据的集合。数据库中的数据不是孤立的,数据与数据之间是

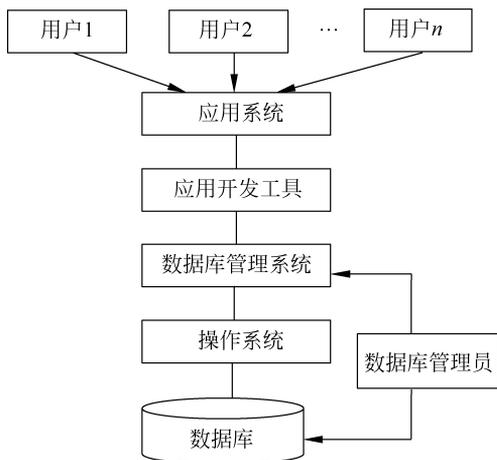


图 5-4 数据库系统的结构

相互关联的。也就是说,在数据库中不仅要能够表示数据本身,还要能够表示数据与数据之间的联系。

(2) 低冗余与数据共享。数据共享是指数据库中的数据可为多个不同的用户所共享,即多个不同的用户可以使用多种不同的语言,为了不同的目的同时存取数据。

(3) 能保证数据的安全、可靠,有效地防止数据库中的数据被非法使用或非法修改。当数据遭到破坏时,能立刻将数据恢复。

(4) 能最大限度地保证数据的正确性。保证数据正确的特性在数据库中称为数据完整性。

(5) 能用综合的方法组织数据。数据库能够根据不同的需求、按不同的方法组织数据。

(6) 数据库中的数据具有较高的独立性。数据的组织和存储方法与应用程序互不依赖、彼此独立。

(7) 允许并发地使用数据库,能及时、有效地处理数据。在多个用户同时使用数据库时,能够保证不产生冲突和矛盾,保证数据的一致性和完整性。

## 2. 数据库管理系统

数据库管理系统(Database Management System, DBMS)是位于用户和操作系统之间的一层数据管理软件,是用户和数据库的接口。DBMS 是数据库系统的核心软件,它的主要功能有以下 4 点。

(1) 数据定义功能。DBMS 提供数据定义语言(Data Definition Language, DDL),用户通过它可以描述数据库结构,定义数据库的完整性约束和安全性控制,并对数据库中的数据对象进行定义。通过使用 DDL 将数据库的结构和数据的特性通知相应的 DBMS,从而生成存储数据的框架。

(2) 数据操纵功能。DBMS 提供数据操纵语言(Data Manipulation Language, DML),用户可以使用 DML 操纵数据,实现对数据库的基本操作,包括对数据库数据的检索、插入、修改和删除等。

(3) 数据运行管理功能。当数据库系统运行时,DBMS 执行管理功能,包括数据安全性控制、完整性控制和并发控制等。

(4) 数据组织和存储功能。DBMS 确定数据的组织方式、文件结构和物理存取方式,并实现数据之间的联系。

### 3. 数据库管理员

数据库管理员(Database Administrator, DBA)指负责设计、建立、管理和维护数据库的人员或部门。DBA 应熟悉计算机的软件和硬件,具有较全面的数据处理知识,熟悉本单位的业务、数据及各项流程。DBA 对保障数据库系统的正常运行具有十分重要的作用。其具体职责如下所述。

- (1) 决定数据库的内容和结构。
- (2) 决定数据库的存储结构和存取策略。
- (3) 定义数据库的安全性要求和数据完整性约束条件。
- (4) 监督数据库的使用和运行。
- (5) 整理和重新构造数据库。

## 5.2.2 常见的数据库管理系统

目前,数据库管理系统软件有很多,如 Oracle、Microsoft SQL Server、MySQL、Access 等,虽然这些产品的功能不完全相同,操作上的差别也很大,但是,这些软件都是以关系模型为基础的,因此都属于关系数据库管理系统(Relational Database Management System, RDBMS)。

下面将介绍常用的 4 个基于关系模型的数据库管理系统。

### 1. Oracle

Oracle 是甲骨文公司的一个关系数据库管理系统,是目前最流行的客户机/服务器或浏览器/服务器体系结构的数据库管理系统之一。

Oracle 被广泛应用于各种操作环境,如 Windows NT、基于 UNIX 系统的小型机与大型机以及一些专用的硬件操作系统平台,并且在管理信息系统、企业数据处理、Internet 及电子商务等领域有着广泛的应用。它具有以下突出特点。

(1) 支持大数据库、多用户的高性能事务处理。Oracle 能支持多用户、大数据量的工作负荷;可充分利用硬件设备;支持大量用户同时在同一数据上执行各种数据应用,并使数据争用达到最小,从而保证数据一致性。

(2) Oracle 具有良好的硬件环境独立性,能支持各种类型的大型、中型、小型和微型计算机系统。

(3) 遵守数据存取语言、操作系统、用户接口和网络通信协议的工业标准。Oracle 是一个开放的系统,又能够有效地保护用户的资源。

(4) 具有良好的安全性和完整性控制。Oracle 实施数据审计,追踪和监控数据存取,有效地保证了数据的安全性。Oracle 通过约束或触发器等机制,实现对数据完整性的控制。

(5) 支持分布式数据库和分布处理。Oracle 可以将物理上分布在不同地点的数据库或不同地点的不同计算机上的数据看作一个逻辑数据库。数据的物理结构对应用程序是隐藏的,数据是否驻留在数据库中对应用程序是透明的。数据可被全部网络用户存取,就好像所有数据都是物理地存储在本地数据库中一样。

(6) 具有可移植性、可兼容性和可连接性。由于 Oracle 可以在许多不同的操作系统上运行,因此在 Oracle 上所开发的应用系统只需做很少的修改甚至不需要修改就可以移植到任何操作系统上。Oracle 同工业标准相兼容,包括许多工业标准的操作系统也都兼容。可连接性是指 Oracle 允许不同类型的计算机和操作系统通过网络共享信息。

## 2. Microsoft SQL Server

Microsoft SQL Server 是由 Microsoft 公司研制的网络数据库管理系统,只能在 Windows 平台上运行,适用于大中型数据库应用系统。

Microsoft SQL Server 是一个典型的 RDBMS,诞生于 20 世纪 80 年代后期,是 Microsoft 品牌中的一个重要产品。Microsoft 公司在 Microsoft SQL Server 产品方面投入了巨大的开发力量,持续不断地研发新技术,以满足用户不断增长的需求,从而使得该产品功能越来越强大,用户使用起来越来越方便,系统的可靠性也越来越高,应用范围更加广泛。

Microsoft SQL Server 是一个全面的、集成的、端到端的数据解决方案,它为企业中的用户提供了一个安全、可靠和高效的平台,服务于企业数据管理和商业智能应用。Microsoft SQL Server 有以下 9 个特点。

- (1) 具有真正的客户机/服务器体系结构。
- (2) 支持 Windows 图形化管理工具,使系统管理和数据库管理更加直观、简单,也可支持本地和远程的系统管理和配置。
- (3) 丰富的编程接口工具,为用户进行程序设计提供了更大的选择余地。
- (4) Microsoft SQL Server 与 Windows 操作系统完全集成,可以使用操作系统的许多功能,如发送和接收消息、管理登录安全性等。
- (5) 强大的事务处理功能,采用各种方法保证数据的完整性。
- (6) 具有良好的伸缩性,可以在 Windows 的各种版本平台上使用。
- (7) 对 Web 技术的支持,使用户能够很容易地将数据库中的数据发布到 Web 页面上。
- (8) 能提供数据仓库功能,这个功能只在 Oracle 和其他更昂贵的 DBMS 中才有。
- (9) 内置的数据复制功能、强大的管理工具、与 Internet 的紧密集成以及开放的系统结构,为广大用户、开发人员和系统集成商提供了一个出众的数据库平台。

## 3. MySQL

MySQL 是一种开放源代码的关系数据库管理系统,开发者为瑞典 MySQL AB 公司。2008 年 1 月,MySQL 被美国的 Sun 公司收购。2009 年 4 月,Sun 公司又被美国的 Oracle 公司收购。

MySQL 是一个单进程、多线程,支持多用户,基于客户机/服务器体系结构的关系数据库管理系统。目前 MySQL 被广泛地应用在 Internet 上的中小型网站中。由于其体积小、速度快、总体成本低、开源等特点,许多中小型网站选择了 MySQL 作为网站数据库管理系统。MySQL 进入 Oracle 产品体系后,获得了更多的研发投入。

MySQL 具有以下突出特点。

- (1) MySQL 运行速度很快,开发者声称 MySQL 可能是目前最快的数据库管理系统。
- (2) MySQL 是一个高性能且相对简单的数据库管理系统,与一些更大数据库管理系统的设置和管理相比,其复杂程度较低。
- (3) MySQL 对大部分的个人用户来说是免费的。MySQL 的价格随平台和安装方式而变化,MySQL 的 Windows 版本在任何情况下都不是免费的。而任何 UNIX 变种(包括 Linux)的 MySQL,如果由用户自己或系统管理员而不是第三方安装,则是免费的,第三方安装则必须支付许可费。
- (4) MySQL 可以利用 SQL(结构化查询语言),优化 SQL 查询算法,有效地提高查询速度。

(5) MySQL 支持多个客户机同时连接到服务器,多个客户机可同时使用多个数据库,可利用几个输入查询并查看结果的界面来交互式地访问 MySQL。这些界面可以是命令行客户机程序、Web 浏览器等。此外,还可以用各种语言(如 C、Perl、Java、PHP 和 Python)编写应用程序界面。因此,可以选择编写客户机程序或使用已编写好的客户机应用程序。

(6) MySQL 是完全网络化的,其数据库可在 Internet 上的任何地方访问。因此,可以和任何地方的任何人共享数据库。而且 MySQL 还能进行访问控制,可以控制哪些人能看到数据,哪些人不能看到数据。

(7) MySQL 可运行在各种版本的 UNIX 以及其他非 UNIX 的系统(如 Windows 和 OS/2)上。MySQL 可运行在从家用微型计算机到高级的服务器上。

#### 4. Access

Access 是由 Microsoft 公司研制的随 Microsoft Office 一起发行的桌面数据库管理系统,具有网络功能相对简单,使用方便的优点,可以满足日常的办公需要,可以用于中小型数据库应用系统。

Access 把数据库引擎的图形用户界面和软件开发工具结合在一起。作为 Microsoft Office 的一部分,具有与 Word、Excel 和 PowerPoint 等相同的操作界面和使用环境,操作简单,易学易懂。和其他关系数据库管理系统相比,Access 具有以下特点。

(1) Access 管理的对象有表、查询、窗体、报表、页、宏和模块,以上对象都存放在扩展名为 .mdb 的数据库文件中,存储方式单一,便于用户进行操作和管理。

(2) Access 是一个面向对象的开发工具,利用面向对象的方式将数据库系统中的各种功能对象化,将数据库管理的各种功能封装在各类对象中。通过对象的方法、属性完成数据库的操作和管理,极大地简化了用户的开发工作。同时,这种基于面向对象的开发方式使开发应用程序变得更为简便。

(3) Access 是一个可视化工具,其风格与 Windows 完全一样。用户想要生成对象并应用,只要使用鼠标进行拖放即可,非常直观方便。系统还提供了表生成器、查询生成器、报表设计器以及数据库向导、表向导、查询向导、窗体向导、报表向导等工具,使操作更加简便、易用和掌握。

(4) Access 基于 Windows 操作系统下的集成开发环境,该环境集成了各种向导和生成器工具,极大地提高了开发人员的工作效率,使得建立数据库、创建表、设计用户界面、设计数据查询、报表打印等可以方便有序地进行。

(5) Access 支持开发数据库互联(Open Data Base Connectivity, ODBC),利用 Access 强大的 DDE(动态数据交换)和 OLE(对象的链接和嵌入)特性,可以在一个数据表中嵌入位图、声音、Excel、Word,还可以建立动态的数据库报表和窗体等。而且 Access 可以将程序应用于网络,并与网络上的动态数据相链接。利用数据库访问页对象生成 HTML 文件,轻松地构建 Internet/Intranet 的应用。

### 5.2.3 数据库系统的结构

从数据库管理系统角度来看,数据库系统通常采用三级模式结构;从数据库最终用户角度看,数据库系统的体系结构分为单用户结构、主从式结构、分布式结构、客户机/服务器结构和浏览器/服务器结构。其中,第一种主要考虑数据管理的特性,第二种主要考虑数据库系统所使用的计算机系统的环境。

### 1. 数据库的三级模式结构

尽管实际的数据库系统的类型和规模可能相差很大,软件产品多种多样(支持不同的数据模型,使用不同的数据库语言,建立在不同的操作系统之上,数据的存储结构也各不相同),但绝大多数的数据库系统一般都遵循美国国家标准委员会下属的标准规划和要求委员会(Standards Planning And Requirements Committee,SPARC)于1975年公布的数据库体系结构标准,即SPARC分级结构。SPARC分级结构将数据库的组织从内到外分为3个层次描述,分别为内模式、概念模式和外模式。数据库的三级模式两级映像结构如图5-5所示。

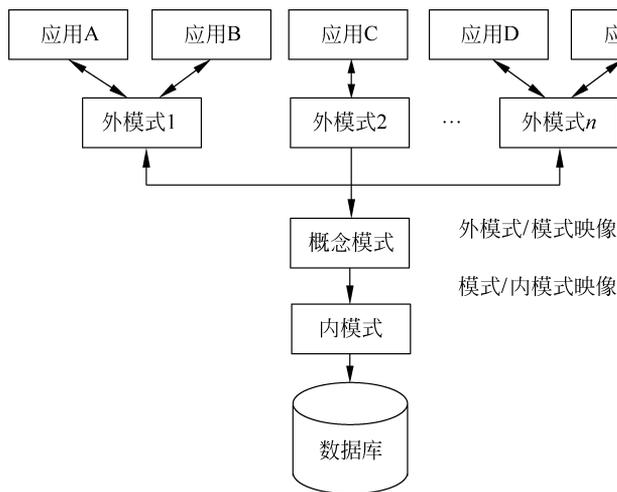


图 5-5 数据库的三级模式两级映像结构

#### 1) 内模式

内模式(Internal Schema,也称为存储模式)是数据库中全体数据的底层描述,描述了数据在存储介质上的存储方式和物理结构,是数据在数据库内部的表示方式。一个数据库只有一个内模式。例如,记录是顺序存储还是按B树结构存储、数据是否加密等。内模式是由系统程序员设计实现的,故称为“系统程序员”视图,由内模式描述语言DSDL描述和定义。

#### 2) 概念模式

概念模式(Schema,简称为模式,也称为逻辑模式)是数据库的总框架,是对数据库中数据的逻辑结构和特征的总体描述。概念模式不涉及数据的物理存储,故称为DBA视图,一个数据库只有一个概念模式。定义概念模式一方面要定义数据的逻辑结构,如数据记录由哪些数据项构成,数据项的名称、类型、取值范围等;另一方面还要定义数据项之间的联系、数据记录之间的联系以及数据的完整性和安全性等要求。由概念模式描述语言DDL描述和定义。

#### 3) 外模式

外模式(External Schema,也称为子模式)通常是模式的一个子集,外模式面向用户,故称为用户视图。一个数据库可以有多个外模式。外模式是数据库用户的数据视图。它属于概念模式的一部分,用来描述用户数据的结构、类型、长度等。所有的应用程序都是根据外模式中对数据的描述编写的。在同一个外模式中可以编写多个应用程序。但一个应用程序只能对应一个外模式。外模式由外模式描述语言SDDL进行具体描述。

这三种模式体现了对数据库的3种不同的观点。内模式表示了物理级数据库,反映了数据库的存储观;概念模式表示了概念级数据库,反映了数据库的整体观;外模式表示用户级数据库,体现了数据库的用户观。

## 2. 数据库的两级映像功能

事实上,三级模式中只有内模式是真正用于存储数据的模式,而概念模式和外模式仅是一种表示数据的逻辑方法,但却可以放心大胆地使用它们,这是由 DBMS 的映像功能实现的。在这三级模式之间存在两级映像。

(1) 模式/内模式映像,用于将概念数据库与物理数据库联系起来。

模式/内模式映像是唯一的,它定义了数据库全局逻辑结构与存储结构之间的对应关系,该映像的定义通常包含在模式描述中。当数据库的存储结构改变时,DBA 对模式/内模式映像做相应的改变,可以使模式保持不变,从而应用程序也不必改变,保证了数据与程序的物理独立性,简称为数据的物理独立性。

(2) 外模式/模式映像,用于将用户数据库与概念数据库联系起来。

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。这些映像定义通常包含在各自外模式的描述中。当模式改变时(如增加新的关系、属性,改变属性的数据类型等),DBA 对各个外模式/模式映像做相应的改变,可以使外模式保持不变,应用程序是依据数据的外模式编写的,从而使应用程序不必修改,保证了数据与程序的逻辑独立性,简称为数据的逻辑独立性。

通过映像可以使数据库有较高的数据独立性,也可以使逻辑结构和物理结构得以分离,给用户的使用带来了方便,最终把用户对数据库的逻辑操作转为对数据库的物理操作。

## 3. 数据库系统的体系结构

在信息高速发展的时代,数据信息同样是宝贵的资产,应该妥善地使用、管理并加以保护。根据数据库存放位置的不同,数据库系统的外部体系结构如下所述。

### 1) 单用户结构

单用户结构的整个数据库系统装在一台计算机上,被一个用户独占,不同机器之间不能共享数据,数据冗余度大,是早期的最简单的数据库系统。例如,一个企业的各个部门都使用本部门的机器来管理本部门的数据,各个部门间的机器是相互独立的。由于不同部门之间不能共享数据,因此企业内部存在大量的冗余数据。

### 2) 主从式结构

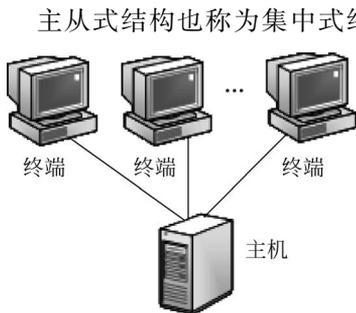


图 5-6 主从式结构的数据库系统

主从式结构也称为集中式结构,所谓集中式结构就是一个主机带多个终端用户结构。在这种结构中,包括应用程序、DBMS、数据都集中地存放在主机上,所有处理任务都由主机来完成。各个用户通过主机的终端可并发地存取数据库中的数据,共享数据资源。主从式结构的数据库系统如图 5-6 所示。

主从式结构数据库系统具有以下 3 个优点。

- (1) 集中控制处理效率高,可靠性好。
- (2) 数据冗余少,数据独立性高。

(3) 易于支持复杂的物理结构,从而获得对数据的有效访问。

其缺点是当终端用户数目增加到一定程度后,主机的任务会变得繁重,使系统性能下降。系统的可靠性依赖于主机,当主机出现故障时,整个系统都不能使用。

随着数据库应用范围的不断拓展,人们也逐渐意识到在处理数据时,过分集中化的系统有许多局限性。例如,不在同一地点的数据无法共享;系统过于庞大、复杂,显得不灵活且安全

性较差；存储容量有限，不能完全适应信息资源存储要求等。为了克服这种系统的缺点，人们采用数据分散的办法，即把数据库分成多个，建立在多台计算机上，这种系统称为分布式数据库系统。

### 3) 分布式结构

分布式数据库是数据库技术、计算机网络技术与分布处理技术相结合的产物，它是物理上分布在计算机网络的不同节点上，逻辑上属于同一系统的数据库系统。

分布式数据库系统的体系结构是在原来集中式数据库系统的基础上增加了分布式处理功能，常常采用集中和自治相结合的控制机构。各局部的数据库管理系统可以独立地管理局部数据库，具有自治的功能。同时，系统又设有集中控制机制，协调各局部数据库管理系统的工作，执行全局应用。

分布式数据库由一组数据组成，这些数据在物理上分布在计算机网络的不同节点上，逻辑上属于同一个系统。这些节点由通信网络连接在一起，每个节点都是一个独立的数据库系统，它们都拥有各自的数据库、中央处理机、终端以及各自的局部数据库管理系统。因此分布式数据库系统可以看作一系列集中式数据库系统的联合。它们在逻辑上属于同一系统，但在物理结构上是分布式的。分布式结构的数据库系统如图 5-7 所示。

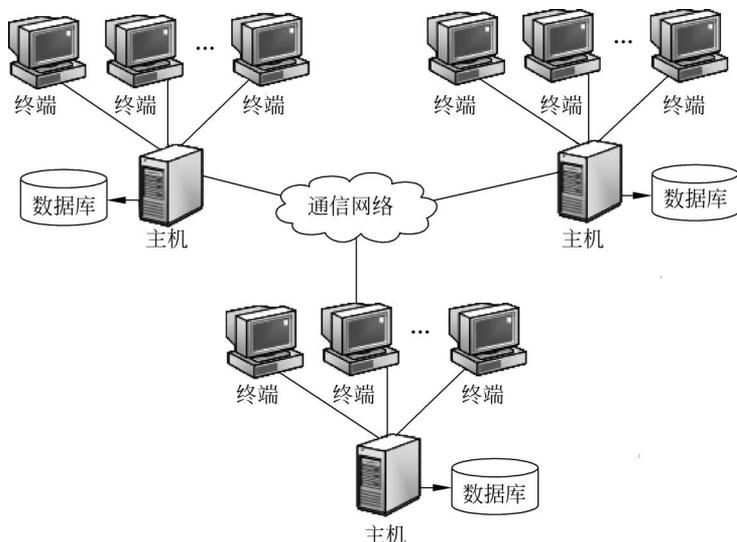


图 5-7 分布式结构的数据库系统

分布式数据库系统具有以下 6 个优点。

- (1) 具有灵活的体系结构。
- (2) 能适应分布式的管理和控制结构。
- (3) 经济性能优越。
- (4) 系统的可靠性高，可用性好。
- (5) 局部应用的响应速度快。
- (6) 可扩展性好，易于集成现有系统。

分布式数据库系统的缺点是系统开销较大，主要花费在通信部分。复杂的存取结构在集中式 DBS 中是有效存取数据的重要技术，但在分布式系统中不一定有效。数据的安全性和保密性较难处理。

### 4) 客户机/服务器结构

客户机/服务器(Client/Server, C/S)结构是一种比较熟悉的数据库系统体系结构，它将数

数据库系统看作由两个非常简单的部分组成：一个服务器(后端)和一组客户机(前端)。前端的客户机上安装专门的应用程序,完成接收、处理数据的工作;后端的数据库服务器主要完成数据的管理工作。客户机和服务器两者都参与一个应用程序的处理,可以有效地降低网络通信量和服务器运算量,从而降低系统的通信开销,可以称为一种特殊的协作式处理模式。

在该体系结构中,客户机向服务器发送请求,服务器响应客户机发出的请求并返回客户机所需要的结果。客户机/服务器结构的数据库系统如图 5-8 所示。

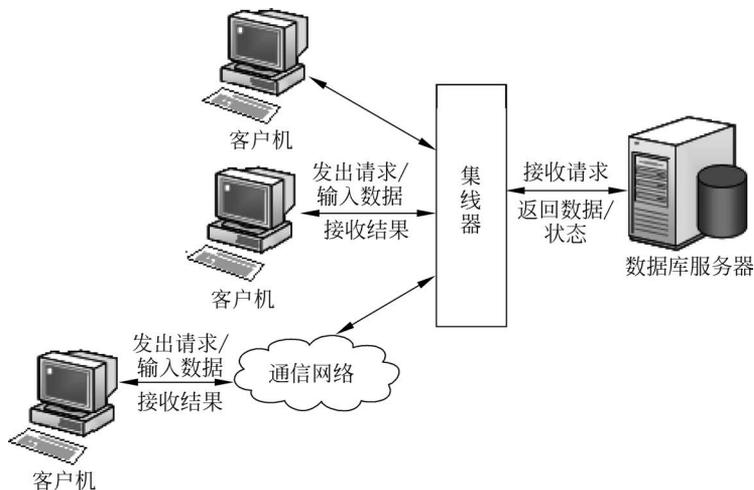


图 5-8 客户机/服务器结构的数据库系统

从应用的角度来说,客户机/服务器结构主要应用在基于行业的数据库应用系统中,如 Outlook Express、QQ、股票信息接收系统等。

客户机/服务器结构的优点是充分利用两端硬件环境的优势,发挥客户机的处理能力,很多工作可以在客户机处理后再提交给服务器,有效降低系统的通信开销。

其缺点是只适用于局域网,客户机需要安装专用的客户机软件,升级维护不方便,并且对客户机的操作系统一般也会有一定限制。

#### 5) 浏览器/服务器结构

浏览器/服务器(Browser/Server, B/S)结构是随着 Internet 技术的兴起,对客户机/服务器结构的一种变化或者改进的结构,是一种以 Web 技术为基础的新型数据库应用系统体系结构。它把传统客户机/服务器模式中的服务器分解为一个数据服务器和多个应用服务器(Web 服务器),统一客户机为浏览器。

浏览器/服务器结构具有三层结构,作为浏览器并非直接与数据库相连,而是通过应用服务器(Web 服务器)与数据库进行交互。这样减少了与数据库服务器的连接数量,而且应用服务器(Web 服务器)分担了业务规则、数据访问、合法校验等工作,减轻了数据库服务器的负担。浏览器/服务器结构的数据库系统如图 5-9 所示。

从应用的角度来说,浏览器/服务器模式特别适合非特定的用户。典型的例子是网上的购物系统、订票系统以及收发电子邮件。

浏览器/服务器结构的优点:首先是简化了客户机,客户机只要安装通用的浏览器软件即可。因此,只要有一台能上网的计算机,就可以在任何地方进行操作而不用安装专门的应用软件,节省客户机的硬盘空间与内存,实现客户机零维护。其次是简化了系统的开发和维护,使系统的扩展非常容易。系统的开发者无须再为不同级别的用户设计开发不同的应用程序,只需把所有的功能都实现在应用服务器(Web 服务器)上,并就不同的功能为各个级别的用户设置权限即可。

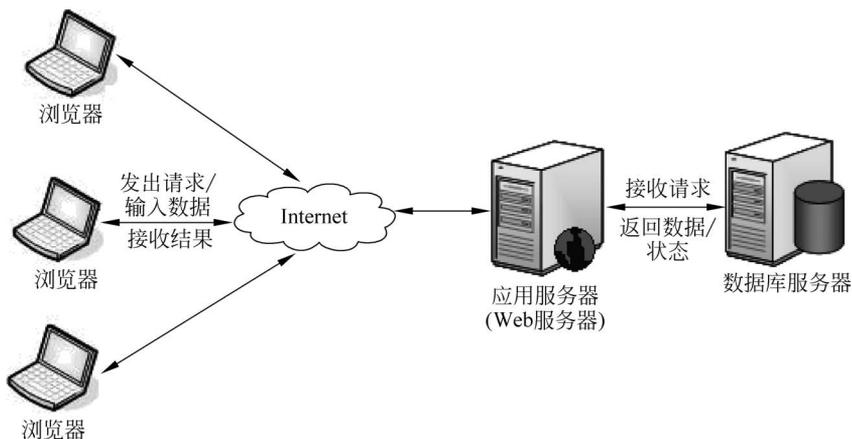


图 5-9 浏览器/服务器结构的数据库系统

浏览器/服务器结构的缺点：首先是应用服务器(Web 服务器)端处理了系统的绝大部分事务逻辑,从而造成应用服务器(Web 服务器)运行负荷较重;其次是浏览器功能简单,许多功能不能实现或实现起来比较困难。

基于上述三层浏览器/服务器结构存在的问题,又提出多层浏览器/服务器体系结构。多层浏览器/服务器体系结构是在三层浏览器/服务器体系结构中间增加了一个或多个中间层,来提高整个系统的执行效率和安全性。

## 5.3 数据模型

数据库系统操作处理的对象来自现实世界中的具体事物,如何用数据来描述、解释现实世界,运用数据库技术表示、处理客观事物及其相互关系呢?这就是数据库的数据模型,在数据库中用数据模型来抽象、表示和处理现实世界中的信息和数据。

对于模型,大家并不陌生,如一架精致的航模飞机、一张地图等。一般而言,模型是现实世界某些特征的模拟和抽象。模型可以分为实物模型与抽象模型。建筑模型、汽车模型、飞机模型等都是实物模型,它们通常是客观事物的某些外观特征或者功能的模拟与刻画;数学模型是一种抽象模型,如 $s = \pi r^2$ ,它抽象地描述了圆的面积和圆的半径之间的数量关系,揭示客观事物的某些本质的、内部的特征。

数据模型(Data Model)也是一种模型,是现实世界数据特征的抽象,它是用来描述数据、组织数据以及对数据进行操作的。

现有的数据库系统均是基于某种数据模型来操作的。数据模型是数据库系统的核心和基础,因此,了解数据模型的基本概念是学习数据库的基础。

根据数据模型应用的不同目的,数据模型可分为3类:概念模型、逻辑模型和物理模型。本节主要介绍概念模型和逻辑模型。客观对象的抽象过程如图5-10所示。

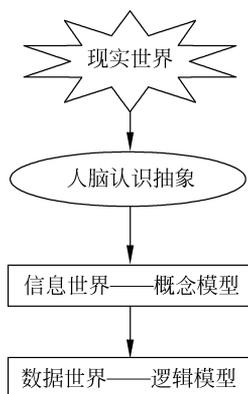


图 5-10 客观对象的抽象过程

### 5.3.1 信息的3个世界

如何抽象地表示与处理现实世界中的数据和信息呢?这就需要利用数据模型工具,即数据库中用于提供信息表示和操作手段的形式框架,是将现实世界转换为数据世界的桥梁。

### 1. 现实世界

现实世界是存在于人脑之外的客观世界,事物及其相互联系就存在于这个世界中,即人们所能看到的、接触到的世界。现实世界存在无数事物(个体),每个个体都有属于自己的特征。想要求解现实问题,就要研究它们的性质及其内在规律,从而找到求解方法。

### 2. 信息世界

信息世界就是现实世界在人们头脑中的反映,又称为观念世界。客观事物在信息世界中称为实体,反映事物间联系的是概念模型。现实世界是物质的,相对而言信息世界是抽象的。在信息世界中,有以下 5 个重要概念。

(1) 实体。客观存在并且可以相互区别的事物称为实体。可以触及的客观对象,如客观存在并且可以相互区别的事物称为实体。客观存在的抽象事件,如一堂课、一次比赛。

(2) 属性。实体的某一特性称为属性。例如,学生实体有学号、姓名、年龄、性别、专业等方面的属性。

(3) 实体集。同类型实体的集合成为实体集。例如,所有的学生、所有的课程等。

(4) 键。能唯一标志一个实体的属性或属性集称为实体的键。例如,学生的学号。而学生的姓名可能出现重名,不能作为学生实体的键。

(5) 域。属性值的取值范围称为该属性的域。例如,学号的域为 8 位整数,性别的域为男、女。

### 3. 数据世界

数据世界是数据在观念世界中信息的数据化,现实世界中的事物及联系在数据世界中用逻辑模型来描述。逻辑模型反映的是数据间的联系,它是对客观事物及其联系的两级抽象的描述,数据库的核心问题是逻辑模型。

信息世界中的实体抽象为数据世界中的数据,存储在计算机中。在数据世界中,有以下 3 个重要概念。

(1) 字段。对应于属性的数据称为字段,也称为数据项。字段的命名往往和属性名相同。例如,学生的学号、姓名、年龄、性别、专业等字段。

(2) 记录。对应于每个实体的数据称为记录。例如,一个学生(990001,张立,20,男,计算机)为一个记录。

(3) 文件。对应于实体集的数据称为文件。例如,所有学生的记录组成了一个学生文件。

对用户来说数据及求解建立模型,即概念模型;对计算机系统而言计算机内部的数据及求解建立模型,即逻辑模型。

### 4. 3 个世界之间的联系

在数据世界中,概念模型被抽象为逻辑模型,现实世界是设计数据库的出发点,也是使用数据库的最终归宿。

(1) 现实世界是信息之源,是设计数据库的出发点。概念模型和逻辑模型是对现实客观事物的两极抽象描述。

(2) 在信息世界中,反映事物间联系的是概念模型;数据库设计的重要任务就是建立概念模型,建立数据库的具体描述。

(3) 现实世界中的事物及联系在数据世界中用逻辑模型来描述,它是实现数据库系统的

根据。

可以得出,要得到正确的逻辑模型,必须首先充分了解客观事物的结构。通过以上介绍,可总结出3个世界中各术语及对应关系如图5-11所示。

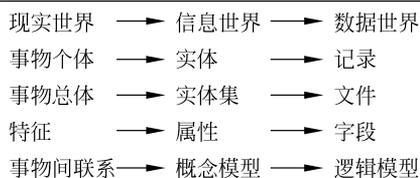


图 5-11 3 个世界中各术语及对应关系

### 5.3.2 概念模型

为了把现实世界中的具体事物抽象为某一 DBMS 支持的数据模型,人们常常首先将现实世界抽象为信息世界,再将信息世界转换为数据世界。概念模型就是用于信息世界的建模,是现实世界到信息世界的第一层抽象。也就是说,概念模型不依赖于具体的计算机系统,而是概念级的模型。

#### 1. 基本概念

概念模型是面向用户的数据模型,它是用户所容易理解的现实世界特征的数据抽象。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。下面介绍其基本概念。

(1) 实体(entity): 例如,一名员工、一件商品等。

(2) 属性(attribute): 例如,员工实体具有员工编号、姓名、性别、年龄、民族、电话、住址、简历、部门编号等属性,用来描述员工的个人信息;商品实体具有品名、单价、产地等属性。

(3) 联系(relationship): 现实世界中事物内部以及事物之间的联系可以用实体集之间的关联关系加以描述。实体之间的联系分为一对一联系(1:1)、一对多联系(1:n)和多对多联系(m:n)。

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 中最多有一个实体与之有联系,并且反之亦然,则称为实体集 A 与实体集 B 之间是一对一联系,记作 1:1,如图 5-12(a)所示。

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 中有 n 个实体与之有联系,反之对于实体集 B 的每一个实体,实体集 A 中最多只有一个实体与之有联系,则称为实体集 A 与实体集 B 之间是一对多的联系,记作 1:n,如图 5-12(b)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 中有 n 个实体与之有联系,反之对于实体集 B 中的每一个实体,实体集 A 中有 m 个实体与之有联系,则称为实体集 A 与实体集 B 之间是多对多联系,记作 m:n,如图 5-12(c)所示。

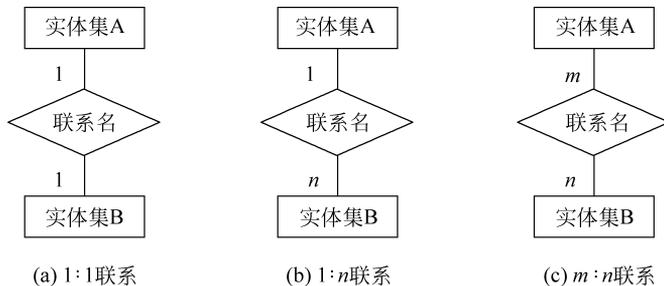


图 5-12 两个实体集之间的 3 种联系

## 2. 实体联系的方法

最常用和最著名的概念模型是实体-联系模型,简称为 E-R 模型。在 E-R 模型中,用 E-R 图来抽象和表示现实世界的数据库特征,是一种语义表达能力强、易于理解的概念模型。

E-R 图提供了用图形表示实体型、属性和联系的方法。

(1) 实体:用矩形表示实体集,矩形内标明实体名,如图 5-13(a)所示。

(2) 属性:用椭圆形表示属性,并用无向边将其与相应的实体连接起来,如图 5-13(b)所示。

(3) 联系:用菱形表示联系,菱形内写出联系名,如图 5-13(c)所示。用无向边分别与有关实体连接起来,同时,在无向边旁边标上联系的类型(1:1、1:n、m:n),如图 5-13(d)~图 5-13(f)所示。

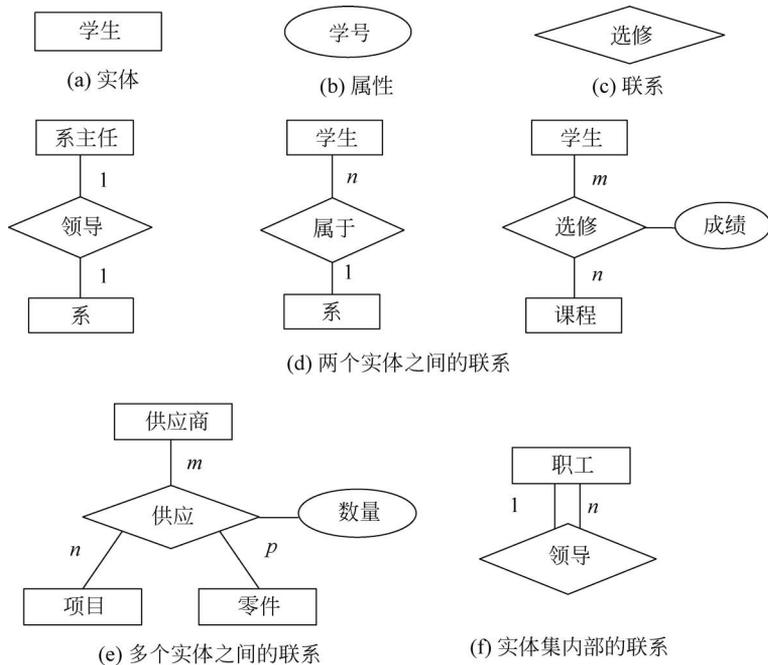


图 5-13 E-R 图的 3 种基本成分及其图形的表示方法

### 5.3.3 逻辑模型

逻辑模型是按计算机系统的观点对数据建模,它是与具体的计算机系统密切相关并直接面向数据库中数据的逻辑结构,主要有层次模型、网状模型、关系模型。

#### 1. 层次模型

层次模型是数据库系统中最早出现的数据模型,其实质是一种有根节点的定向有序树。树的节点是记录类型,根节点只有一个,其余节点有且仅有一个父节点。上一层记录与下一层记录的关联关系只能是一对多联系,即每个记录至多有一个父记录。层次模型如图 5-14 所示。

层次模型的数据操作包括数据记录的插入、删除、修改和检索。

层次模型的数据模型本身比较简单,对于实体间联系是固定的,且预先定义好了应用系

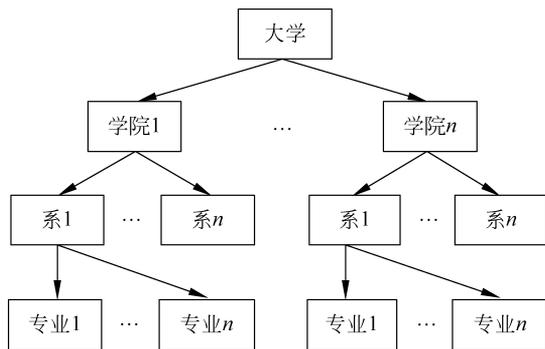


图 5-14 层次模型

统,采用层次模型来实现,其性能优于关系模型,不低于网状模型。层次模型提供了良好的完整性支持。

由于现实世界中很多联系是非层次性的,层次模型难以直接体现这些联系。对插入、删除和检索操作的限制比较多,如查询子节点必须通过双亲节点。由于结构严密,层次命令趋于程序化。

## 2. 网状模型

用网状结构表示实体类型及实体之间联系的数据模型称为网状模型。在网状模型中,一个子节点可有多个父节点。在两个节点之间可以有一种或多种联系,其实体间的联系为多对多联系,记录之间的联系是通过指针实现的。例如,课程实体在两个节点之间可以有一种或多种联系。网状模型如图 5-15 所示。

网状模型的数据操作包括数据记录的插入、删除、修改和检索,支持记录与联系的连入、断开和转移操作。

网状数据模型的主要优点是能够更为直接地描述现实世界,如一个节点可以有多个双亲,具有良好的性能,存取效率高。

其主要缺点为结构比较复杂,而且随着应用环境的扩大,数据库的结构就变得越来越复杂,不利于最终用户的掌握。其 DDL、DML 复杂,给用户的使用带来不便。

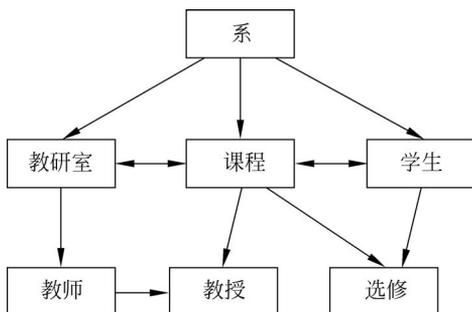


图 5-15 网状模型

## 3. 关系模型

1970年,IBM公司的研究人员 E. F. Codd 发表论文,提出了关系数据模型。随后又发表一系列论文,阐述了关系规范化的概念。

用表格形式表示实体类型及实体之间联系的数据模型称为关系模型。关系数据模型把一些复杂的数据结构归结为简单的二维表格形式。记录是表中的行,属性是表中的列。关系模型如图 5-16 所示。

作为一个关系的二维表,必须满足以下条件。

(1) 表中的每一列必须是基本数据项。

学号	姓名	性别	出生日期	所在院系
20230001	张丽	女	2005-03-11	会计
20230002	周佳睿	男	2005-07-15	会计
20230003	赵晶晶	女	2005-05-08	计算机
20230004	李明	男	2005-11-12	计算机

图 5-16 关系模型

- (2) 表中的每一列必须具有相同的数据类型。
- (3) 表中的每一列名字必须是唯一的。
- (4) 表中不应该有内容完全相同的行。
- (5) 行的顺序与列的顺序不影响表格中所表示的信息的含义。

在关系数据库中,对数据的操作一般建立在一个或多个表格上,并通过对这些关系表格的分类、合并、连接或选取等运算来实现数据的管理。关系模型具有以下优点。

- ① 关系模型与非关系模型不同,它是建立在严格的数学概念的基础上的。
  - ② 关系模型的概念单一。无论实体还是实体之间的联系都用关系表示。对数据的检索结果也是关系(即表)。
  - ③ 关系模型的存取路径对用户透明,从而具有较高的数据独立性和更好的安全保密性,同时也简化了程序员的开发工作和数据库的建立工作。
- 关系模型的主要缺点是由于存取路径对用户透明,查询效率往往不如非关系模型。

## 5.4 结构化查询语言

关系数据库系统的数据语言有多种,但经过多年的使用、竞争和更新后,结构化查询语言(SQL)已成为国际标准化组织所确认的关系数据库系统所使用的唯一数据语言。用该语言所书写的程序可以在任何关系数据库系统上运行。

### 5.4.1 结构化查询语言的特点及功能

SQL 允许对数据库进行复杂的查询,同时也提供了创建数据库和维护数据库的方法,是数据库应用程序开发的重要组成部分,应用范围非常广泛。

#### 1. SQL 的特点

SQL 的特点是集数据定义(definition)、数据查询(query)、数据操纵(manipulation)、数据控制(control)功能于一体的、综合的、通用的、功能极强同时又简洁易学的语言。

(1) 在关系模型中实体和实体间的联系均用关系表示,查找、插入、删除、更新等操作都只需一种操作符号。SQL 的命令动词如表 5-1 所示。

表 5-1 SQL 的命令动词

SQL 功能	命令动词
数据操纵	SELECT、INSERT、UPDATE、DELETE
数据定义	CREATE、DROP、ALTER
数据控制	GRANT、REVOKE

(2) 在采用 SQL 进行数据操作时,只要提出“做什么”,而不必指明“怎么做”,其他工作由

系统完成,具有高度的非过程化。

(3) 以同一种语法结构提供两种使用方式:一种是联机交互使用方式,在此种方式下,SQL可以独立使用,称为自含式语言;另一种是嵌入式使用方式,在此种方式下,它以某些高级程序设计语言(如Java、C等)为宿主语言,而SQL则被称为嵌入式语言。

(4) SQL支持关系数据库的三级模式结构。其中,视图对应的是外模式;大多数基本表对应的是概念模式;数据库的存储文件、索引文件构成关系数据库的内模式。

## 2. 结构化查询语言的功能

下面以学生成绩管理系统为例。在该系统中,学生信息包含每名学生的学号、姓名、性别、出生日期等信息;学校各门课程信息包括课程编号、课程名称、学时、学分等信息;成绩管理包含每名学生的每门课程成绩情况。学生信息表如表5-2所示,课程信息表如表5-3所示,成绩信息表如表5-4所示。

表 5-2 学生信息表

学 号	姓 名	性 别	出 生 日 期	家 庭 住 址
20230001	张丽	女	2005-03-11	黑龙江省哈尔滨市
20230002	周佳睿	男	2005-07-15	北京市
20230003	赵晶晶	女	2005-05-08	黑龙江省牡丹江市
20230004	李明	男	2005-11-12	天津市
20230005	姜晓海	男	2005-08-25	黑龙江大庆市

表 5-3 课程信息表

课 程 编 号	课 程 名 称	学 时	学 分	开 课 院 系	考 核 方 式
010101	线性代数	48	3	数学	考试
010102	高等数学	64	4	数学	考试
010103	外语	64	4	外语	考试
010104	软件工程	48	3	计算机	考试
010105	数据库	48	3	计算机	考查

表 5-4 成绩信息表

学 号	课 程 编 号	成 绩
20230002	010101	77
20230002	010103	98
20230003	010103	86
20230004	010104	91
20230005	010102	63
20230005	010105	89

SQL的主要功能包括数据定义、数据查询、数据操纵、数据控制、与主语言的接口以及存储过程等。

(1) 数据定义。数据定义功能包括基本表的定义和删除、视图的定义和删除、索引的建立和删除。

(2) 数据查询。数据查询功能包括单表查询、多表连接查询、分组、排序等。

(3) 数据操纵。数据操纵功能包括数据查询、数据删除、数据插入、数据修改。

(4) 数据控制。数据控制功能包括数据的完整性约束、数据的安全性及存取权限、数据的触发、数据的并发控制及故障恢复。

(5) 与主语言的接口。SQL 提供游标语句以解决 SQL 与主语言间因数据不匹配所引起的接口。

(6) 存储过程。SQL 还提供远程调用功能。在客户机/服务器模式下,客户机中的应用可以通过网络调用服务器数据库中的存储过程。存储过程是一个由 SQL 语句所组成的过程,该存储过程在被应用调用后执行 SQL 语句序列,最终将结果返回给应用。存储过程可以供多个应用所共享。

## 5.4.2 数据定义功能

关系数据库系统支持三级模式结构,其内模式、模式和外模式中的基本对象有表、视图和索引。因此,SQL 的数据定义功能包括基本表、视图和索引等数据对象的定义。本节只介绍基本表的定义。

### 1. 定义基本表

生成新的表要使用 CREATE TABLE 命令。语句格式:

```
CREATE TABLE <表名>(<列名><数据类型>[列级完整性约束条件]
[,<列名><数据类型> [列级完整性约束条件]]...
[,<表级完整性约束条件>])
```

其中,<表名>是所要定义的基本表的名字,它可由一个或多个属性(列)组成。

功能: 建立一个新的基本表,指明基本表的表名与结构,包括组成该表的每个字段名、数据类型等。

**【例 5-1】** 以创建学生成绩管理系统为例,使用 SQL 语句建立一个学生信息表,它由学号、姓名、性别、出生日期、家庭住址这 5 个属性组成。其中,学号不能为空,并且值是唯一的。

```
CREATE TABLE 学生信息
(学号 CHAR(5) NOT NULL UNIQUE,
姓名 CHAR(8),
性别 CHAR(2),
出生日期 DATETIME,
家庭住址 CHAR(10))
```

### 2. 修改基本表

在创建了一个基本表以后,可以使用 ALTER TABLE 语句对表进行修改。语句格式:

```
ALTER TABLE <表名>
ADD <列名><数据类型>[<列级完整性约束>]
ALTER COLUMN <列名><数据类型>[<列级完整性约束>]
DROP <完整性约束>
```

其中,<表名>是要修改的基本表,ADD 子句用于增加新列和新的完整性约束条件,ALTER COLUMN 用于修改列名、数据类型、列级完整性约束条件,DROP 子句用于删除指定的完整性约束条件。

**【例 5-2】** 在学生信息表中,增加备注字段,数据类型为 CHAR(10);修改家庭住址字段,数据类型为 CHAR(50);删除备注字段。

```
ALTER TABLE 学生信息
ADD 备注 CHAR(10)
ALTER TABLE 学生信息
ALTER COLUMN 家庭住址 CHAR(50)
```

```
ALTER TABLE 学生信息  
DROP COLUMN 备注
```

### 3. 删除基本表

语句格式:

```
DROP TABLE <表名>
```

功能: 删除指定表及其数据, 释放相应的存储空间, 同时系统也自动删除在此表上建立的各种索引, 也删除了在该表上授予的操作权限。虽然删除表时并未删除定义在该表上的视图, 但这些视图已无效, 不能再使用了。

**【例 5-3】** 删除已存在的学生信息表。

```
DROP TABLE 学生信息
```

## 5.4.3 数据查询功能

数据库查询是数据库的核心操作。SQL 提供了 SELECT 语句进行数据库的查询, 功能非常强大, 其选项也非常丰富, 同时 SELECT 语句的完整句法也非常复杂。其一般格式为:

```
SELECT[ ALL|DISTINCT]<目标列表表达式>[, <目标列表表达式>]  
FROM <参与查询的表名或视图名>  
[ WHERE <查询选择的条件> ]  
[ GROUP BY <分组表达式> ] [ HAVING <分组查询条件> ]  
[ ORDER BY <排序表达式> [ ASC | DESC ] ]
```

**说明:**

SELECT 子句指定要显示的属性列, 它可以是星号(\*)、表达式、列表、变量等。

FROM 子句指定要查询的基本表或者视图。

WHERE 子句用来限定查询的范围和条件。

GROUP BY 子句对查询结果按指定列的值分组, 将属性列值相等的元组分为一个组。通常会在每组中作用聚合函数。

HAVING 短语跟随 GROUP BY 子句使用, 筛选出满足指定条件的组。

ORDER BY 子句对查询结果按指定列值的升序或降序排序。ASC 表示升序排列, DESC 表示降序排列。

整个语句的含义为根据 WHERE 子句中的条件表达式, 从基本表(或视图)中找出满足条件的元组, 按 SELECT 子句中的目标列, 选出元组中的分量形成结果表。

这些子句中 SELECT 子句和 FROM 子句为必需的, 其他子句为可选项。SELECT 语句既可完成简单的单表查询, 又可完成复杂的连接查询和嵌套查询。

下面以学生成绩管理系统为例, 对常用的查询方法进行说明。在学生成绩管理系统数据库中有学生信息表、课程信息表和成绩信息表, 结构如表 5-2~表 5-4 所示。

### 1. 单表查询

在很多情况下, 用户只对单一表中的一部分属性行或列感兴趣, 这时可以通过 SELECT 子句在一个表中进行查询。

**【例 5-4】** 查询全体学生的学号和姓名。

```
SELECT 学号, 姓名
```

FROM 学生信息

**【例 5-5】** 查询选修课程编号为“010101”的学生的学号和成绩。

```
SELECT 学号, 成绩
FROM 成绩信息
WHERE 课程编号 = '010101'
```

**【例 5-6】** 查询所有课程信息,并按学时降序排列。

```
SELECT *
FROM 课程信息
ORDER BY 学时 DESC
```

**【例 5-7】** 查询选修两门及两门以上课程的学生学号。

```
SELECT 学号
FROM 成绩信息
GROUP BY 学号
HAVING COUNT(*) > 2
```

注意,此处应先分组,再对每一组计数,选出统计结果大于或等于 2 的组的学号,WHERE 子句作用于基本表或视图,HAVING 短语作用于组。

## 2. 多表连接查询

前面的查询都是针对一个表进行的。如果一个查询同时涉及两个或两个以上的表,则称为多表查询。多表查询是关系数据库中最主要的查询。

**【例 5-8】** 查询出成绩高于 90 分的学生的学号和姓名。

```
SELECT 学生.学号, 姓名
FROM 学生信息, 成绩信息
WHERE 学生信息.学号 = 成绩信息.学号 AND 成绩 > 90
```

**【例 5-9】** 查询出所有学生的学号、姓名、课程名称和成绩。

```
SELECT 学生信息.学号, 姓名, 课程名称, 成绩
FROM 学生信息, 课程信息, 成绩信息
WHERE 学生信息.学号 = 成绩信息.学号 AND
成绩信息.课程编号 = 课程信息.课程编号
```

## 5.4.4 数据操纵功能

SQL 数据操纵功能是指用来插入、更新和删除数据库中数据的功能,这些语句包括 INSERT、UPDATE、DELETE 等。

### 1. INSERT 语句

插入数据语句用于向数据库中添加一行新记录,并给新记录的字段赋值。其一般格式为:

```
INSERT INTO <表名> [( <列名 1> [, <列名 2> ] ... )]
VALUES( <表达式 1> [, <表达式 2> ] ... )
```

**【例 5-10】** 向学生信息表中添加新记录。

```
INSERT INTO 学生信息
VALUES('20230006', '张红', '女', '2005-05-10', '黑龙江省哈尔滨市')
```

### 2. UPDATE 语句

更新数据语句用于更新数据库表中特定记录或者字段的数据,其一般格式为:

```
UPDATE <表名>  
SET <列名> = <表达式>[, <列名> = <表达式>] ...  
[ WHERE <条件表达式>]
```

功能：修改指定表中满足 WHERE 子句条件的元组，如省略 WHERE 子句，则表示要修改表中的所有元组。SET 子句指出将被更新的列及其新值。

**【例 5-11】** 将学生信息表中学号为“20230004”的学生姓名改为“赵鑫”。

```
UPDATE 学生信息  
SET 姓名 = '赵鑫'  
WHERE 学号 = '20230004'
```

**【例 5-12】** 将所有课程的学分加 1。

```
UPDATE 课程信息  
SET 学分 = 学分 + 1
```

### 3. DELETE 语句

使用删除数据语句可以删除表中的一行或多行记录，其一般格式为：

```
DELETE FROM <表名>  
[ WHERE <条件>]
```

功能：从指定表中删除满足 WHERE 子句条件的所有元组。若省略 WHERE 子句，则表示删除表中全部元组，但表依然存在。

**【例 5-13】** 删除学生信息表中姓名为“赵鑫”的学生记录。

```
DELETE FROM 学生信息  
WHERE 姓名 = '赵鑫'
```

**【例 5-14】** 删除学生信息表中所有学生记录。

```
DELETE FROM 学生信息
```

## 5.5 数据库系统的开发过程

数据库设计是综合运用计算机软硬件技术，结合应用系统领域的知识和管理技术的系统工程。它不是凭借个人经验和技巧就能够设计完成的，而是须遵守一定的规则实施设计而成。

在现实世界中，信息结构十分复杂，应用领域千差万别，而设计者的思维也各不相同，所以数据库设计的方法和路径多种多样。

和其他软件一样，数据库的设计过程可以使用软件工程中的生存周期（称为数据库设计的生存期，它是指从数据库研制到不再使用它的整个时期）的概念来说明。

按规范设计法，可将数据库设计分为 6 个阶段。

- (1) 需求分析阶段。
- (2) 概念结构设计阶段。
- (3) 逻辑结构设计阶段。
- (4) 物理结构设计阶段。
- (5) 数据库实施阶段。
- (6) 数据库运行与维护阶段。

数据库设计中，前两个阶段是面向用户的应用要求，面向具体的问题；中间两个阶段是面

向数据库管理系统；最后两个阶段是面向具体的实现方法。前4个阶段可统称为分析和设计阶段，后两个阶段称为实现和运行阶段。

### 5.5.1 需求分析

需求分析是整个数据库设计过程中的第一步，也是最重要的一步。整个数据库开发活动从对系统的需求分析开始。系统需求包括对数据的需求和对应用功能的需求两方面内容。该阶段应与系统用户相互交流，了解他们对数据的要求及已有的业务流程，并把这些信息用数据流程图或文字等形式记录下来，最终获得处理需求。

从数据库设计的角度来看，需求分析的任务是对现实世界要处理的对象（组织、部门、企业）等进行详细的调查，通过对原系统的了解，收集支持新系统的基础数据并对其进行处理，在此基础上确定新系统的功能。

具体地说，需求分析阶段的任务包括以下3项。

#### 1. 调查分析用户的活动

这个过程通过对新系统运行目标的研究、对现行系统所存在的主要问题的分析以及制约因素的分析，明确用户总的需求目标，确定这个目标的功能域和数据域。具体做法如下所述。

(1) 调查组织机构情况，包括该组织的部门组成情况、各部门的职责和任务等。

(2) 调查各部门的业务活动情况，包括各部门输入和输出的数据与格式、所需的表格与卡片、加工处理这些数据的步骤、输入输出的部门等。

#### 2. 收集和分析需求数据，确定系统边界

在熟悉业务活动的基础上，协助用户明确对新系统的各种需求，包括用户的信息需求、处理需求、安全性和完整性的需求等。

(1) 信息需求指目标范围内涉及的所有实体、实体的属性以及实体间的联系等数据对象，也就是用户需要从数据库中获得信息的内容与性质。由信息要求可以导出数据要求，即在数据库中需要存储哪些数据。

(2) 处理需求指用户为了得到需求的信息而对数据进行加工处理的要求，包括对某种处理功能的响应时间、处理的方式（批处理或联机处理）等。

(3) 安全性和完整性的需求。在定义信息需求和处理需求的同时必须相应地确定安全性和完整性。

在收集各种需求数据后，对调查的结果进行初步分析，确定新系统的边界，确定哪些功能由计算机完成或将来准备让计算机完成、哪些活动由人工完成。由计算机完成的功能就是新系统应该实现的功能。

#### 3. 编写数据需求分析说明书

系统分析阶段是在调查与分析的基础上，依据一定的规范要求编写数据需求分析说明书。数据分析需求说明书一般依据一定规范要求编写。我国有相关的国家标准与部委标准，也有企业标准，其制定是为了规范说明书编写及需求分析的内容，同时也为了统一编写格式。数据需求分析说明书一般用自然语言和表格书写。

数据需求分析说明书是对需求分析阶段的一个总结。编写数据需求分析说明书是一个不断反复、逐步深入和逐步完善的过程，数据需求分析说明书应包括如下内容。

- (1) 系统概况,系统的目标、范围、背景、历史和现状。
- (2) 系统的原理和技术,对原系统的改善。
- (3) 系统总体结构与子系统结构说明。
- (4) 系统功能说明。
- (5) 数据处理概要、工程体制和设计阶段划分。
- (6) 系统方案及技术、经济、功能和操作上的可行性。

完成数据需求分析说明书后,在项目单位的领导下,要组织有关技术专家对数据需求分析说明书进行评审,这是对需求分析结构的再审查。审查通过后,由项目方和开发方的领导签字认可。数据需求分析说明书应提供下列附件。

- (1) 系统的硬件、软件支持环境的选择及规格要求(所选择的数据库管理系统、操作系统、汉字平台、计算机型号及其网络环境等)。
- (2) 组织机构图、组织之间联系图及各机构功能业务一览图。
- (3) 数据流程图、功能模块图和数据字典等图表。

如果用户同意数据需求分析说明书和方案设计,在与用户进行详尽商讨的基础上,最后签订技术协议书。数据需求分析说明书是设计者和用户一致确认的权威性文献,是今后各阶段设计和工作的依据。

## 5.5.2 概念结构设计

在需求分析阶段,设计人员充分调查并描述用户的需求,但这些需求只是现实世界的具体要求,应把这些需求抽象为信息世界的结构,这样才能更好地实现用户的需求。概念结构设计就是将由需求分析得到的用户需求抽象为信息结构,即概念模型。

概念结构的主要目的就是分析数据之间的内在语义关联,在此基础上建立一个数据的抽象模型。描述概念结构的工具是 E-R 图。

### 1. 概念结构设计的方法

设计概念结构的 E-R 模型可采用 4 种方法。

- (1) 自顶向下。先定义全局概念结构 E-R 模型的框架,再逐步细化。
- (2) 自底向上。先定义各局部应用的概念结构 E-R 模型,然后将它们集成,得到全局概念结构 E-R 模型。
- (3) 逐步扩张。先定义最重要的核心概念 E-R 模型,然后向外扩充,以滚雪球的方式逐步生成其他概念结构 E-R 模型。
- (4) 混合策略。该方法采用自顶向下和自底向上相结合的方法,先自顶向下定义全局框架,再以它为骨架集成自底向上方法中设计的各个局部概念结构。

### 2. 概念设计设计的步骤

这里只介绍自底向上的设计方法的步骤。在概念结构设计过程中使用 E-R 方法的基本步骤包括设计局部 E-R 图、综合成初步 E-R 图、优化成基本 E-R 图。

下面举例说明从局部 E-R 图到全局 E-R 图的设计步骤。

**【例 5-15】** 在简单的教务管理系统中,有如下语义约束:一个学生可选修多门课程,一门课程可被多个学生选修,因此学生和课程是多对多的联系;一个教师可讲授多门课程,一门课程可被多个教师讲授,因此教师和课程也是多对多的联系;一个系可有多个教师,一个教师只

能属于一个系,因此系和教师是一对多的联系,同样系和学生也是一对多的联系。

### 1) 设计局部 E-R 图

设计局部 E-R 图的任务是根据需求分析阶段产生的各个部门的数据流程图和数据字典中相关数据,设计出各项应用的局部 E-R 图。具体操作如下所述。

#### (1) 确定实体和属性。

实体和属性之间在形式上并无可以明显区分的界限,通常是按照现实世界中事物的自然划分来定义实体和属性,将现实世界中的事物进行数据抽象,得到实体和属性。

#### (2) 确定联系类型。

依据需求分析结果,考察任意两个实体类型之间是否存在联系。若有联系,则要进一步确定联系的类型。在确定联系时应特别注意两点:一是不要丢掉联系的属性;二是尽量取消冗余的联系,即取消可以从其他联系导出的联系。

#### (3) 画出局部 E-R 图。

根据上述约定,可以得到如图 5-17 所示的学生选课局部 E-R 图和如图 5-18 所示的教师任课局部 E-R 图。形成局部 E-R 模型后,应该再去征求用户意见,以求完善,使之如实地反映现实世界。

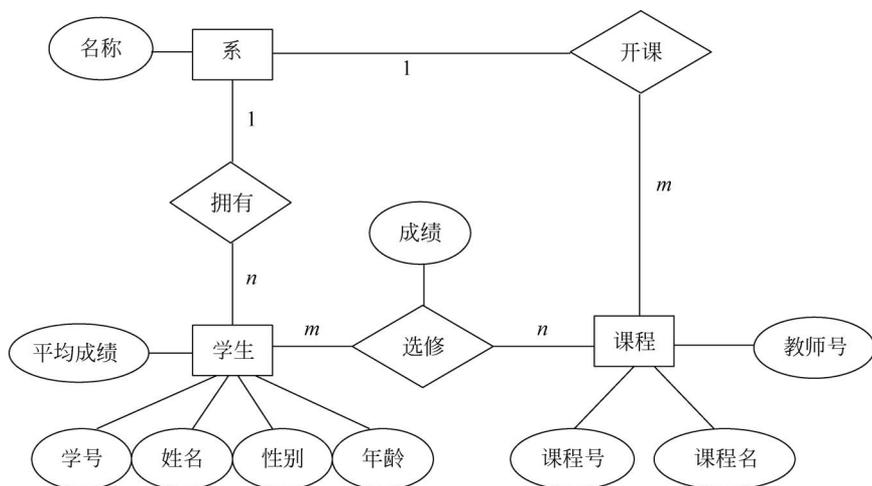


图 5-17 学生选课局部 E-R 图

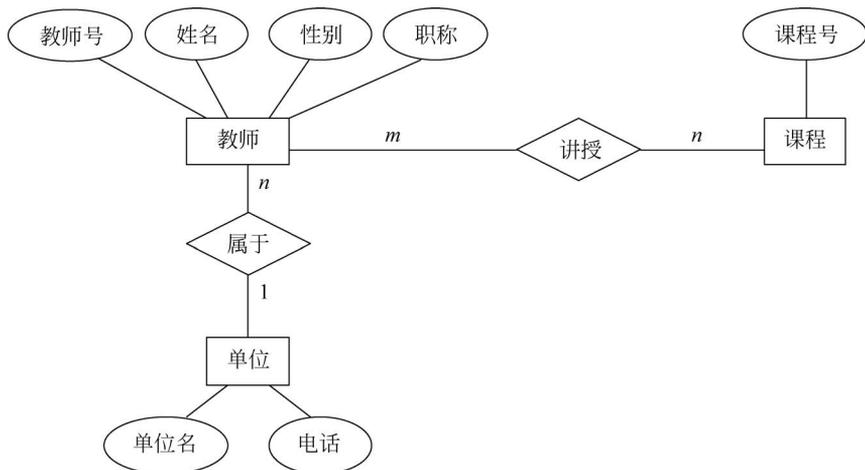


图 5-18 教师任课局部 E-R 图

## 2) 综合成初步 E-R 图

### (1) 局部 E-R 图的合并。

为了减小合并工作的复杂性,先两两合并。合并从公共实体类型开始,最后再加入独立的局部结构。

### (2) 消除冲突。

一般有 3 种类型的冲突:属性冲突、命名冲突和结构冲突。具体调整手段可以考虑以下 3 种:对同一个实体的属性取各个分 E-R 图相同实体属性的并集;根据综合应用的需要,把属性转换为实体或者把实体变为属性;实体联系要根据应用语义进行综合调整。

下面以教务管理系统中的两个局部 E-R 图为例说明如何消除各局部 E-R 图之间的冲突,进行局部 E-R 模型的合并,从而生成初步 E-R 图。

首先,这两个局部 E-R 图中存在着命名冲突,学生选课局部 E-R 图中的实体“系”与教师任课局部 E-R 图中的实体“单位”,都是指“系”,即所谓的异名同义,合并后统一改为“系”,这样实体中的属性“名称”和实体“单位”中的属性“单位名”即可统一为“系名”。

其次,还存在着结构冲突,实体“系”和实体“单位”在两个不同应用中的属性组成不同,合并后这两个实体的属性组成为原来局部 E-R 图中的同名实体属性的并集。解决上述冲突后,合并两个局部 E-R 图,生成教务管理系统的初步 E-R 图如图 5-19 所示。

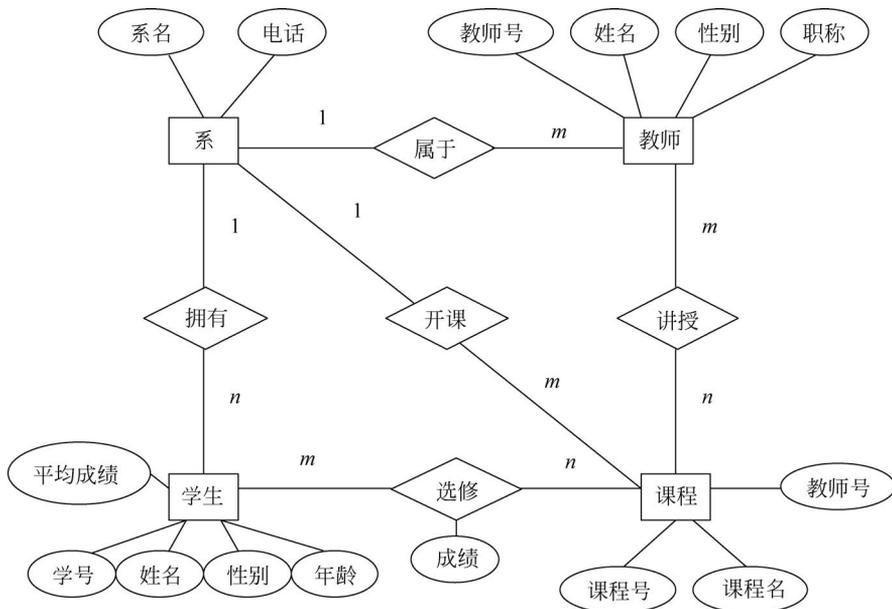


图 5-19 教务管理系统的初步 E-R 图

### (3) 初步 E-R 图的优化。

E-R 图的优化主要包括消除冗余属性和消除冗余联系两步。

图 5-19 所示的初步 E-R 图中,“课程”实体中的属性“教师号”可由“讲授”这个“教师”与“课程”之间的联系导出,而学生的平均成绩可由“选修”联系中的属性“成绩”中计算出来,所以“课程”实体中的“教师号”与“学生”实体中的“平均成绩”均属于冗余数据。另外,“系”和“课程”之间的联系“开课”,可以由“系”和“教师”之间的“属于”联系与“教师”和“课程”之间的“讲授”联系推导出来,所以“开课”属于冗余联系。

这样,图 5-19 所示的初步 E-R 图在消除冗余数据和冗余联系后,便可得到基本的 E-R 模型。教务管理系统的基本 E-R 图如图 5-20 所示。

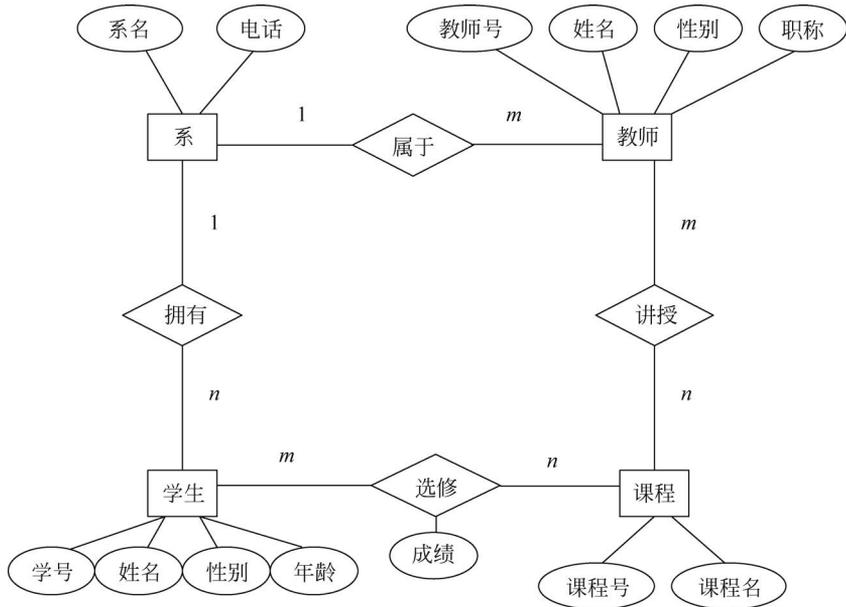


图 5-20 教务管理系统的基本 E-R 图

最终得到的基本 E-R 模型是企业的概念模型,它代表了用户的数据要求,是沟通要求和设计的桥梁。它决定数据库的总体逻辑结构,是成功建立数据库的关键。如果设计不好,就不能充分发挥数据库的功能,无法满足用户的处理要求。

因此,用户和数据库人员必须对这一模型反复讨论,在用户确认这一模型已正确无误地反映了他们的要求后,才能进入下一阶段的设计工作。

### 5.5.3 逻辑结构设计

概念结构设计阶段得到的 E-R 模型是用户的模型,它独立于任何一种数据模型,独立于任何一个具体的 DBMS。为了建立用户所要求的数据库,需要把上述概念模型转换为某个具体的 DBMS 所支持的数据模型。数据库逻辑结构设计的任务是将概念结构转换为特定 DBMS 所支持的数据模型的过程。从此开始便进入了实现设计阶段,需要考虑具体的 DBMS 的性能、具体的数据模型特点。

E-R 图所表示的概念模型可以转换为任何一种具体的 DBMS 所支持的数据模型,如网状模型、层次模型和关系模型。这里只讨论关系数据库的逻辑设计问题,所以只介绍 E-R 图如何向关系模型进行转换。逻辑结构设计过程可分为初始关系模式设计、关系模式规范化、模式的评价与改进。

#### 1. 初始关系模式设计

概念结构设计中得到的 E-R 图是由实体、属性和联系组成的,而关系数据库逻辑结构设计的结果是一组关系模式的集合。所以将 E-R 图转换为关系模型实际上就是将实体、属性和联系转换为关系模式。在转换中要遵循以下原则。

(1) 一个实体集转换为一个关系模式,实体的属性就是关系的属性,实体的关键字就是关系的关键字。

(2) 一个 1:1 的联系转换为一个关系模式,每个实体的关键字都是关系的候选关键字。

(3) 一个  $1:n$  的联系转换为一个关系模式多个实体的关键字是关系的关键字。

(4) 一个  $m:n$  的联系转换为一个关系模式,联系中各实体关键字组成关系的关键字(组合关键字)。

(5) 具有相同关键字的关系需要合并。

## 2. 关系模式规范化

规范化理论在数据库设计中有如下 3 方面的应用。

(1) 在需求分析阶段,用数据依赖概念分析和表示各个数据项之间的联系。

(2) 在概念结构设计阶段,以规范化理论为指导,确定关键字,消除初步 E-R 图中冗余的联系。

(3) 在逻辑结构设计阶段,从 E-R 图向数据模型转换的过程中,用模式合并与分解方法达到规范化级别。

**【例 5-16】** 以图 5-20 所示的 E-R 模型为例,E-R 图向关系模型转换的具体做法如下所述。

(1) 把每一个实体转换为一个关系模式。

首先分析各实体的属性,从中确定其主键,然后分别用关系模式表示。

4 个实体分别转换为 4 个关系模式:

学生(学号,姓名,性别,年龄)

课程(课程号,课程名)

教师(教师号,姓名,性别,职称)

系(系名,电话)

其中,有下画线者表示是主键。

(2) 把每一个联系转换为关系模式。

由联系转换得到的关系模式的属性集中,包含两个发生联系的实体中的主键以及联系本身的属性,其关系键的确定与联系的类型有关。

4 个联系也分别转换为 4 个关系模式:

属于(教师号,系名)

讲授(教师号,课程号)

选修(学号,课程号,成绩)

拥有(系名,学号)

## 3. 模式的评价与改进

模式的评价主要包括功能和性能两方面。经过反复多次的模式评价和修正之后,最终的数据库模式得以确定。逻辑结构设计阶段的结果是全局逻辑数据库结构。对于关系数据库系统来说,就是由一组符合一定规范的关系模式组成的关系数据库模型。

### 5.5.4 物理结构设计

数据库最终要存储在物理设备上,并为物理设备实现数据的处理和输出提供理论依据。数据库在物理设备上的存储结构和存取方法称为数据库的物理结构,它依赖于给定的计算机系统。

一方面设计人员必须深入了解给定的 DBMS 的功能: DBMS 提供的环境和工具、硬件环

境特别是存储设备的特征。另一方面也要了解应用环境的具体要求。只有“知己知彼”才能设计出较好的物理结构。决定存储结构的主要因素包括存取时间、存储空间和维护代价 3 方面。设计时应当根据实际情况,对这 3 方面进行综合权衡。一般情况下,DBMS 也具有一定的灵活性可供选择。

确定了数据库的物理结构之后,要对其进行评价,重点是时间和空间的效率。如果评价结果满足设计要求,则可进行数据库实施。实际上,往往需要经过反复测试才能优化物理结构设计。

### 5.5.5 数据库实施

数据库实施是指根据逻辑结构设计和物理结构设计的结果,在计算机上建立起实际数据库结构、装入数据、进行测试和试运行的过程。该阶段是建立数据库的实质性阶段,需要完成装入数据、编码、测试等工作。完成以上工作后,即可投入试运行,即把数据库连同有关的应用程序一起装入计算机,从而考察它们在各种应用中能否达到预定的功能和性能要求。

(1) 数据库加载。由于数据库的数据量都很大,加载一般是通过系统提供的实用程序或自编的专门录入程序进行的。在真正加载数据之前,有大量的数据整理工作要做。应当建立严格的数据录入和检验规范,设计完善的数据检验与校正程序,才能确保数据的质量。

(2) 数据库运行和维护数据库投入运行标志着数据库设计与应用开发工作基本结束,运行与维护阶段开始。

### 5.5.6 数据库运行与维护

完成了部署数据库系统后,用户开始使用系统,但这并不标志着数据库开发周期的结束。要保持数据库持续稳定的运行,需要数据库管理员具备特殊的技能,同时要付出更多的劳动。而且,由于数据库环境是动态的,随着时间的推移,用户数量和数据库事务不断扩大,数据库系统必然发展。因此,数据库管理员必须持续地关注数据库管理,并在必要时对数据库进行升级。

数据库运行与维护阶段的主要任务包括以下 3 点。

- (1) 维护数据库的安全性和完整性。
- (2) 监测并改善数据库性能。
- (3) 必要时对数据库进行重新组织。

只要数据库系统在运行,就需要不断地进行修改、调整和维护。一旦应用变化太大,数据库重新组织也无济于事,这就表明数据库应用系统的生命周期结束,应该建立新系统,重新设计数据库。从头开始数据库设计工作,这也标志着一个新的数据库应用系统生命周期的开始。

## 5.6 区块链简介

互联网的出现极大地加快了信息传递的速度,深刻地改变了人们的生活方式,但互联网技术并不关心人与人之间的协作模式和信任构建方法。而区块链在信息互联网的基础上构建了一种新的可信的大规模协作方式,以解决数字经济发展的信息问题,被誉为下一代互联网的重要特征。很多人对区块链的理解仅止步于比特币类的加密数字货币,那么区块链究竟是什么?

### 5.6.1 区块链的概念

最早的区块链技术雏形出现在比特币项目中,作为比特币背后的分布式记账平台,在无集中式管理的情况下,比特币网络稳定运行了近8年时间,支持了海量的交易记录,并未出现严重的漏洞。

公认的最早关于区块链的描述性文献是《比特币:一种点对点的电子现金系统》,但该文重点在于讨论比特币系统,实际上并没有明确提出区块链的定义和概念,其中区块链被描述为用于记录比特币交易的历史账目。复式记账法能对每一笔账目同时记录来源和去向,将对账验证功能引入记账过程,提高了记账的可靠性。区块链则是首个自带对账功能的数字记账技术。狭义来讲,区块链是一种按照时间顺序将数据区块以顺序相连的方式组合成的一种链式数据结构,并以密码学方式保证的不可篡改和不可伪造的分布式账本。从更广泛的意义来看,区块链属于一种去中心化的记录技术。

### 5.6.2 区块链的特征

区块链是多种已有技术的集成创新,主要用于实现多方信任和高效协同。通常,一个成熟的区块链系统具备去中心化、透明可信、防篡改、可追溯、隐私安全保障以及系统高可靠这6个特性。

#### 1. 去中心化

由于使用分布式核算和存储,不存在中心化的硬件或管理机构,任意节点的权利和义务都是均等的,系统中的数据块由整个系统中具有维护功能的节点来共同维护。

#### 2. 透明可信

(1) 人人记账保证人人获取完整信息,从而实现信息透明化。在去中心化的系统中,网络中的所有节点均是对等节点,大家平等地发送和接收网络中的消息。所以,系统中的每个节点都可以完整地观察系统中节点的全部行为,并将观察到的这些行为在各个节点进行记录,即维护本地账本,整个系统对于每个节点都具有透明性。

(2) 节点间决策过程共同参与,共识保证可信性。区块链系统是典型的去中心化系统,网络中的所有交易对所有节点均是透明可见的,所以整个系统对所有节点均是透明、公平的,系统中的信息具有可信性。

#### 3. 防篡改

防篡改是指交易一旦在全网范围内经过验证并添加至区块链,就很难被修改或者抹除。一方面,当前区块链所使用的共识算法,从设计上保证了交易一旦写入即无法被篡改;另一方面,基于共识算法的区块链系统的篡改难度及花费都是极大的,若要对此类系统进行篡改,攻击者需要控制全系统超过51%的节点。

#### 4. 可追溯

可追溯是指区块链上发生的任意一笔交易都是有完整记录的,可以针对某一状态在区块链上追查与其相关的全部历史交易。防篡改特性保证了写入的区块链上的交易很难被篡改,这为可追溯特性提供了保证。

## 5. 隐私安全保障

区块链的去中心化特性决定了区块链的“去信任”特性。因为区块链系统中的任意节点都包含了完整的区块校验逻辑,所以任意节点都不需要依赖其他节点完成区块链中交易的确认过程,也就是无须额外地信任其他节点。“去信任”的特性使得节点之间不需要互相公开身份,因为任意节点都不需要根据其他节点的身份进行交易有效性的判断,这为区块链系统保护用户隐私提供了前提保障。

## 6. 系统高可靠

每个节点对等地维护一个账本并参与整个系统的共识。也就是说,如果其中某一个节点出故障了,整个系统也依然能够正常运转,这就是为什么可以自由地加入或者退出比特币系统网络,而整个系统依然可以正常工作。

### 5.6.3 区块链平台发展历程

区块链的诞生最早可以追溯到密码学和分布式计算,区块链的发展先后经历了加密数字货币、企业应用和价值互联网 3 个阶段。

#### 1. 区块链 1.0: 加密数字货币

2009 年 1 月,在比特币系统论文发表两个月之后,比特币系统正式运行并开放了源码,这标志着比特币网络的正式诞生。通过其构建的一个公开透明、去中心化、防篡改的账本系统,比特币开展了一场规模空前的加密数字货币实验。在区块链 1.0 阶段,区块链技术的应用主要聚集在加密数字货币领域,典型代表即比特币系统以及比特币系统代码衍生出来的多种加密数字货币。比特币系统在全球范围内只能支持每秒 7 笔交易,交易记账后追加 6 个区块才能比较安全地确认交易,追加一个区块大约需要 10min,意味着大约需要 1h 才能确认交易,不能满足实时性等较高的应用需求。

#### 2. 区块链 2.0: 企业应用

针对区块链 1.0 存在的专用系统问题,为了支持如众筹、溯源等功能,区块链 2.0 阶段为支持用户自定义的业务逻辑,引入了智能合约,从而使区块链的应用范围得到了极大的拓展,开始在各个行业迅速落地,极大地降低了社会生产消费过程中的信任和协作成本,提高了行业内和行业间协同效率。典型的代表是 2013 年启动的以太坊系统。针对区块链 1.0 阶段存在的性能问题,以太坊系统在共识算法方面也进行了提升。

以太坊项目为其底层的区块链账本引入了智能合约的交互接口,这对区块链应用进入 2.0 时代发挥了巨大作用。智能合约是一种通过计算机技术实现的,旨在以数字化方式达成共识、履约、监控履约过程并验证履约结果的自动化合同,极大地扩展了区块链的功能。有了智能合约系统的支持,区块链的应用范围开始从单一的货币领域扩大到涉及合约共识的其他金融领域。区块链技术首先在股票、清算、私募股权等众多金融领域崭露头角。传统的通过交易所的股票发行完全可以被区块链分布式账本所取代。这样,企业就可以通过分布式自治组织协作运营,借助用户的集体行为和集体智慧获得更好的发展。在投入运营的第一天就能实现募资,而不用经历复杂的流程,进而产生高额费用。

随着区块链 2.0 阶段智能合约的引入,其开放透明、去中心化及不可篡改的特性在其他领

域逐渐受到重视。各行业专业人士开始意识到,区块链的应用也许不仅局限于金融领域,还可以扩展到任何需要协同共识的领域中。于是,在金融领域之外,区块链技术又陆续应用到了公证、仲裁、审计、域名、物流、医疗、邮件、签证、投票等其他领域,应用范围逐渐扩大到各个行业。

### 3. 区块链 3.0: 价值互联网

进入 21 世纪以来,全球科技创新进入空前活跃的时期,新一轮科技革命和产业革命正在重构全球创新版图,重塑全球经济结构。以人工智能、量子信息、移动通信、物联网、区块链为代表的新一代信息技术加速突破应用。

价值互联网是一个可信赖的,能实现各个行业协同互连,实现人和万物互连,实现劳动价值高效、智能流通的网络,主要用于解决人与人、人与物、物与物之间的共识协作、效率提升问题,将传统的依赖于人或中心的公正、调节、仲裁功能自动化,按照协议交给可信赖的机器来自动执行。通过对现有互联网体系进行改革,区块链技术将与 5G 网络、机器智能、物联网等技术创新一起承载着智能化、可信赖的梦想飞向更具价值的互联网时代。

在未来,区块链将渗透到生活和工作的方方面面,充分发挥审计、监控、仲裁和价值交换的作用,确保技术创新向着让世界更加美好的方向发展。

## 5.6.4 区块链的关键技术

从技术角度讲,区块链涉及的领域比较复杂,包括分布式存储、密码学、心理学、经济学、博弈论、网络协议等。由于区块链主要解决的是交易的信任和安全问题,针对这个问题提出了以下 3 个有待解决或改进的关键性技术。

### 1. 分布式账本

分布式账本就是交易记账由分布在不同地方的多个节点共同完成,而且每个节点都记录着完整的账目。因此,所有节点都可以合法地参与监督交易,同时也可以共同为其作证。

与传统的分布式存储有所不同,区块链的分布式存储的独特性主要体现在两方面:一是区块链的每个节点都按照区块链式结构存储完整的数据,而传统分布式存储一般是将数据按照一定的规则分成多份进行存储;二是区块链每个节点存储都是独立的、地位等同的,依靠共识机制保证存储的一致性,而传统分布式存储一般是通过中心节点往其他备份节点同步数据。

没有任何一个节点可以单独记录账本数据,从而避免了单一记账人被控制或者被贿赂而记假账的可能性。也由于记账节点足够多,理论上讲除非所有的节点被破坏,否则账目就不会丢失,从而保证了账目数据的安全性。

### 2. 密码学技术

存储在区块链上的交易信息是公开的,如何防止交易记录被篡改?如何证明交易方的身份?如何保护交易双方的隐私?密码学正是解决这些问题的有效手段。将账户身份信息高度加密,只有在数据拥有者授权的情况下才能访问到,从而保证了数据的安全和个人的隐私。

传统方案包括哈希算法、加解密算法、数字证书和签名等。区块链技术的应用将可能促进密码学的进一步发展,包括随机数的产生、安全强度、加解密处理的性能等。随着量子计算机等新技术的出现,RSA 密码算法等已经无法提供足够的安全性,这些问题的解决将依赖于数学科学的进一步发展和新一代计算技术的突破。

### 3. 分布式共识机制

共识机制就是所有记账节点之间如何达成共识,去认定一个记录的有效性。这既是认定的手段,也是防止篡改的手段。共识机制的核心在于如何解决某个变更在网络中是一致的,是被大家都承认的,同时这个信息是被确定的、不可推翻的。

区块链提出了工作量证明机制(PoW)、股权证明机制(PoS)、授权股权证明机制(DPoS)和实用拜占庭容错算法(PBFT)4种不同的共识机制,适用于不同的应用场景,在效率和安全性之间取得平衡。

区块链的共识机制具备“少数服从多数”以及“人人平等”的特点,其中“少数服从多数”并不完全指节点个数,也可以是计算能力、股权数或者其他的计算机可以比较的特征量。“人人平等”是指当节点满足条件时,所有节点都有权优先提出共识结果,直接被其他节点认同后并最后有可能成为最终共识结果。

以比特币为例,采用的是工作量证明机制,只有在控制了全网超过51%的记账节点的情况下,才有可能伪造出一条不存在的记录。当加入区块链的节点足够多时,伪造记录基本上是不可能的,从而杜绝了造假情况的发生。

## 5.6.5 区块链的应用

### 1. 金融管理

自有人类社会以来,金融交易就是必不可少的经济活动。交易本质上交换的是价值的所属权(如房屋、车辆的所属权)。现在为了完成交易,往往需要一些中间环节,特别是中介担保角色。这是因为交易双方往往存在着不充分信任的情况,要证实价值所属权并不容易,而且往往彼此的价值不能直接进行交换。合理的中介担保确保了交易的正常进行,提高了经济活动的效率,但已有的第三方中介机制往往存在成本高、时间周期长、流程复杂、容易出错等缺点。正是因为这些原因,金融服务成为区块链最为火热的应用领域之一。

区块链技术可以为金融服务提供有效的所属权证明和相当强的中介担保机制。

### 2. 银行交易

银行的活动包括发行货币,完成存款、贷款等大量的交易内容。银行必须能够确保交易的正确性,必须通过诸多手段确立自身的信用地位。传统的金融系统为了完成上述功能,开发了极为复杂的软件和硬件方案,不仅消耗了昂贵的成本,还需要大量的维护成本。即便如此,这些系统仍然存在诸多缺陷,如很多交易都不能在短时间内完成,每年都会发生大量的利用银行相关金融漏洞进行的犯罪。此外,在目前金融系统流程情况下,大量商家为了完成交易,还常常需要额外的组织(如支付宝)进行处理,这些实际上都增加了金融交易的成本。

区块链技术被认为是有可能促使银行交易发生革命性变化的技术。

### 3. 资源共享

当今社会,共享经济进行得如火如荼,而资源共享目前面临的问题主要包括共享过程成本过高、用户身份评分难、共享服务管理难等。区块链可以作为解决这些问题的一个有效的途径。

例如,大量提供短租服务的公司已经开始尝试用区块链来解决共享中的难题。一份报告

中指出: Airbnb 等 P2P 住宿平台已经开始通过利用私人住所打造公开市场,来变革住宿行业,但是这种服务的接受程度可能因人们对人身安全以及财产损失的担忧而受到限制。如果引入安全且无法篡改的数字化资质和信用管理系统,区块链就能有助于提升 P2P 住宿的接受度。报告还指出,可能采用区块链技术的企业包括 Airbnb、HomeAway 以及 OneFineStay 等,市场规模为 30 亿~90 亿美元。

#### 4. 证券交易

证券交易包括交易执行和确认环节。交易本身相对简单,主要是由交易系统(极为复杂的软件、硬件系统)完成电子数据库中内容的变更。但中心的验证系统极为复杂和昂贵,交易指令执行后的结算和清算环节也十分复杂,往往需要较多的人力成本和时间成本,并且容易出错。

目前来看,基于区块链的处理系统还难以实现海量交易系统所需要的性能。但在交易的审核和清算环节,区块链技术存在诸多的优势,可以避免人工的参与。

2015 年 10 月,美国纳斯达克证券交易所推出区块链平台,实现主要面向一级市场的股票交易流程。通过该平台进行股票发行的发行者将享有数字化的所有权。

### 5.6.6 区块链的价值和前景

区块链是人类迄今为止去中心化和解决信任问题的一次革命性的探索,具备去中心化、透明、防篡改、高效率、低成本等特性。区块链从一开始就致力于解决人类信任问题,将人与人的信任转变为人与机器的信任。如果说现代社会处在契约时代,区块链将使人类进入自动契约时代。通过编码让机器代替第三方中介委托监管各种契约履约情况,既提高了效率,又避免了第三方违约的情况。

随着区块链技术的快速发展,其必将在更多领域更深层次地影响和改变商业社会的发展。区块链技术对商业社会的影响具体表现在以下 3 方面。

#### 1. 降低社会交易成本

区块链网络中的所有信息都是经过多方共识、可信、不可篡改的。这将极大地简化传统交易模型中所要面对的冗长的交易审查、确认等流程,甚至不再需要重复的账目核对、价值结算、交易清算等操作,从而大幅度降低社会交易成本。

#### 2. 提升社会效率

随着区块链技术在经济领域的应用,必将优化各领域内的业内流程,降低运营成本,提高协同效率。以金融领域内的场景为例,当前金融系统是一个复杂庞大的系统,跨行交易、跨国汇兑往往需要依赖各类“中介”组织来实现,漫长的交易链条,加之缺乏统一的监管方式,使得交易效率低下,大量资产在交易过程中被锁定或延时冻结。而借助区块链系统实现去中心化体系,社会中的投资和交易将可以实现实时结算,这将有助于大幅度提升投资和交易效率。

#### 3. 交易透明可监管

信息的实时性和有效性是监管效率的关键。除了涉及个人隐私或商业机密等情况外,区块链技术可以实现有效的交易透明、不可篡改特性。监管机构还可以实现实时的透明监管,甚至可以通过智能合约对交易实现自动化的合规检查、欺诈甄别等。

## 5.7 本章小结

本章主要讲述了数据库的基本概念,介绍了数据库系统与体系结构,又介绍了数据模型、结构化查询语言和数据库系统的开发过程,并简单介绍了区块链知识。

数据管理技术经历了人工管理、文件系统和数据库管理系统阶段。这 3 个阶段是随着计算机软件、硬件技术和应用领域的发展而发展的。数据库、数据库管理系统和数据库系统的基本概念和基本知识是学习数据库技术的基础。数据库系统三级模式和两级映像是本章的难点。希望读者通过本章的学习,能够了解信息的 3 个世界和数据模型;掌握概念模型(用于信息世界的建模)、逻辑模型(用于数据世界的建模);掌握 SQL 的功能和常用语句;学会简单 SQL 语句查询功能;了解数据库设计过程和区块链技术。

## 5.8 赋能加油站与扩展阅读

在物资管理中,一个供应商为多个项目供应多种零件,一种零件只能保存在一个仓库中,一个仓库中可保存多种零件,一个仓库有多名员工值班,由一个员工负责管理。画出该物资管理系统的 E-R 图。

请说出 Access、SQL Server 和 Oracle 数据库管理系统各自的优点和缺点。

练习使用 Windows 操作系统中的 ODBC 创建一个数据源,连接到 Access 数据库,数据库名称自定。



扩展阅读



思维小筑

## 5.9 习题与思考

1. 什么是数据库、数据库管理系统和数据库系统?
2. 试述数据库系统的三级模式结构。
3. 试述数据库系统的组成。
4. 常用的数据库管理系统有哪些?
5. 简述常用的数据模型及其特点。
6. 简述 SQL 的特点。
7. 简述 SQL 的功能。
8. 简述数据库设计的步骤。
9. 试述数据库设计的意义。