

第3章 计算机软件技术基础



练一练

- 什么是计算机软件、计算机软件的分类。
- 操作系统的作用、常用操作系统介绍。
- 程序设计语言的介绍、组成和处理。
- 算法的定义、特征以及控制结构。
- 数据库的基本概念。
- 关系模型与关系代数。
- SQL 语言的基本知识。

计算机系统通常由硬件系统和软件系统组成。硬件是组成计算机的物理设备，软件是完成数据处理任务所需的各种程序的集合，两者相互依存，是构成计算机系统不可或缺的两个部分。

计算机解决现实世界中的问题，必须先提出一个算法，然后依据算法设计出程序，如图3-1所示。实际上，程序是解决问题算法的具体体现。计算机执行程序中规定的各种操作，便完成了数据处理任务。

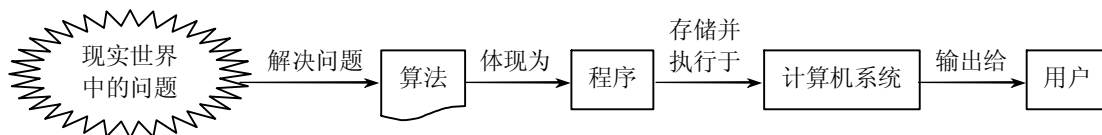


图3-1 计算机解决问题的过程

从计算机底层的角度来看，程序是机器指令的一个序列。从程序设计语言的角度来看，程序就是用更容易理解和表达的语言记号对数据和数据加工过程的描述。

软件是计算机的灵魂，如同人类大脑里的思想和知识。本章先介绍计算机软件，再介绍程序设计语言、算法、数据结构和数据库等知识。

3.1 概述

3.1.1 什么是计算机软件

按照传统的观点，计算机软件就是计算机系统上完成数据处理任务所需的各种程序的集合。即使是专业人员，提到软件首先想到的也是计算机程序。但随着计算机科学的不断发展，

再把软件等同于程序就不准确了。

对于计算机软件的概念，目前尚没有一个统一的定义。我们一般认为计算机软件（Computer Software）是指计算机系统上的程序、数据和文档的集合。其中，程序当然是软件的主体，单独的数据或文档不能认为是软件；数据是程序运行过程中需要处理的对象和必须使用的一些参数（如函数、英汉字典等）；文档指的是与程序开发、维护及操作有关的一些资料（如设计报告、维护手册和使用指南等）。通常，软件（特别是商品软件和大型软件）必须有完整、规范的文档作为支持。

软件具有与硬件不同的特点：

（1）表现形式不同。硬件看得见，摸得着。而软件无形，看不见，摸不着。软件大多存在人们的脑袋里或纸面上，它的正确与否，是好是坏，一直要到程序在机器上运行才能知道。这就给设计、生产和管理带来许多困难。

（2）生产方式不同。软件开发是人的智力的高度发挥，不是传统意义上的硬件制造。尽管软件开发与硬件制造之间有许多共同点，但这两种活动是根本不同的。

（3）要求不同。硬件产品允许有误差，而软件产品却不允许有误差。

（4）维护不同。硬件是要用旧用坏的，在理论上，软件是不会用旧用坏的，但实际上，软件也会变旧变坏。因为在软件的整个生存周期中，一直处于改变（维护）状态。

3.1.2 软件分类

从应用的角度出发，通常将计算机软件分为系统软件和应用软件。

1. 系统软件

系统软件（System Software）指那些为了有效使用计算机系统、给应用软件开发与运行提供支持、能为用户管理与使用计算机提供方便的软件。一般分为两大类。

一类软件负责管理计算机系统的资源，与计算机硬件紧密地结合，使计算机系统的硬件部件、相关的软件和数据相互协调地工作，同时支持用户很方便地使用计算机，高效率地共享计算机系统的资源。

操作系统（Operating System）是这类系统软件的代表。常见的操作系统有 Windows、UNIX、Linux。计算机要完成的任务虽然各不相同，但会涉及一些所有用户都共同需要的基础性操作。例如都要通过输入设备取得数据，向输出设备送出数据；从磁盘读取数据，向磁盘写入数据；把程序装载到内存中，启动这个程序等。这些操作也要由一系列指令来完成。因此可以把这些指令集中起来，组合成为一个操作系统，对其他程序提供统一的支持。此外，操作系统还要负责管理硬件、软件和外存数据，使得在一台计算机上运行的各个程序有条不紊地共享有限的硬件设备，共享系统里存放的软件和数据。例如，两个程序都要向硬盘存入各自的数据，如果没有操作系统作为一个协调管理机构来为它们划定区域的话，怎么避免可能出现的互相破坏对方数据的情况呢？

另一类系统软件通常称为实用程序或实用软件。它们负责提供几乎是所有用户都会需要的各种各样的公共应用服务，例如基本输入/输出系统（BIOS）、程序设计语言的各种处理程序、数据库管理系统（DBMS）、数据备份程序、数据恢复程序、磁盘清理程序等。

2. 应用软件

应用软件是指用户为解决某些应用领域中的各类问题而开发的程序，这种应用程序五花

八门、极其丰富，很多通用的程序可以根据其功能组成不同的软件包，供用户下载使用。按照应用软件的开发方式和适用范围，应用软件可以分为通用应用软件和定制应用软件。

(1) 通用应用软件。通用应用软件就是几乎人人都需要使用的应用软件。现代社会不论从事什么职业，无论是学习还是娱乐，人们所做的活动计算机几乎都有相应的应用软件来提供服务。

通用应用软件有很多种。例如文字处理软件、信息检索软件、媒体播放软件、网络通信软件等，如表 3-1 所示。这些软件易学易用，是当代学习生活必不可少的软件。

表 3-1 通用应用软件的主要类别和功能

类别	功能	流行软件举例
文字处理软件	文本编辑、文字处理、桌面排版等	WPS、Word 等
电子表格软件	表格制作、数值计算和统计等	Excel 等
图形图像软件	图像处理、几何图形绘制、动画制作等	AutoCAD、Photoshop、3ds max、CorelDRAW 等
媒体播放软件	播放各种数字音频和视频文件	Windows Media Player、Real Player、Winamp 等
网络通信软件	电子邮件、聊天、IP 电话等	Outlook Express、MSN、QQ、ICQ 等
演示软件	幻灯片制作等	PowerPoint 等
游戏软件	游戏和娱乐	联众、QQ 游戏等
信息检索软件	在数据库和 Internet 中查找需要的信息	万方数据、超星浏览器等

(2) 定制应用软件。定制应用软件就是按照不同领域用户的特定应用要求而开发的软件。如大学教务管理系统、医院门诊挂号系统、酒店客房管理系统等。这类软件专用性强，设计和开发成本相对比较高，只有相应的机构用户需要购买，价格也比通用应用软件高。

没有配置任何软件，只包含硬件系统的计算机称为“裸机”。安装了操作系统后的计算机，原来的硬件并没有发生变化，但功能和运行效率得到极大的增强，一般称为“虚计算机”。裸机、操作系统、实用程序、应用软件之间的层次关系如图 3-2 所示。大多数情况下，用户是在和一台安装了操作系统的计算机打交道。用户和操作系统交互的方式属于“软件用户界面”问题。

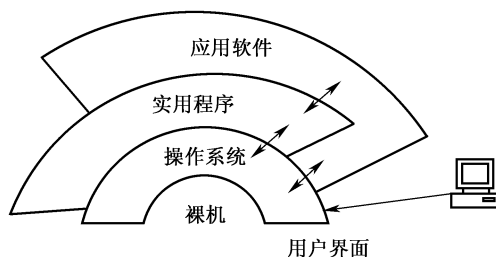


图 3-2 计算机的软件层次关系

3.1.3 软件版权保护

提到买软件，很多人可能第一反应就是买盗版或者找朋友复制一份！如果这样做，你可

能已经触犯了法律。如同其他出版物一样，软件产品具有智力产品的特性，是受到知识产权法律保护的对象。

版权是一种排他性的法律权利，没有得到版权持有人的许可，复制有版权的产品就触犯了知识产权法。书籍、文章、音乐、电影、计算机软件都是有版权的产品。从法律的观点看，复制一个软件而没有给软件的版权持有人付费，性质和在超市不付钱就拿走商品是一样的。这是一种软件侵权行为。同样，未经唱片公司、电影公司的许可，就在网络上提供其产品的影音下载服务，是一种网络侵权行为。

当然，不是所有的软件都是要收费的。公开软件就没有版权，软件作者只是把他的作品与大家分享。Linux 操作系统原版就是一个典型例子。免费软件有版权，但在一段时间或者某个范围里免费发行。常见的一种情况是，版权人想看看市场反应，推出一个免费的试用版。共享软件则是用户可以免费得到的软件，有时在需要技术支持或者软件升级时要收费。

至于拥有专利的软件，用户只能在购买许可证（即购买软件的使用权，而不是软件产品本身）后才可使用。



练一练

- 1.（判断题）计算机软件通常是指用于指示计算机完成特定任务的，以电子格式存储的程序、数据和相关的文档。
- 2.（判断题）软件使用说明是软件产品不可缺少的一部分。
- 3.（单选题）如果你购买了一个商品软件，通常就意味着得到了它的（ ）。
A. 修改权 B. 拷贝权 C. 使用权 D. 版权
- 4.（填空题）从应用的角度出发，通常将计算机软件分为_____和_____。其中我们经常使用的 Excel 软件是一种_____软件。
- 5.（填空题）计算机软件由程序、数据和文档组成，其中主体是_____。

3.2 操作系统

操作系统（Operating System, OS）是管理计算机硬件与软件资源的程序，同时也是计算机系统的内核与基石。操作系统负责诸如管理与配置内存、决定系统资源供需的优先次序、控制输入与输出设备、操作网络与管理文件系统等基本事务。

3.2.1 概述

1. 操作系统的作用

操作系统的作用是统一调度、统一分配和统一管理所有的硬件设备和软件系统，使各个部分之间协调一致、有条不紊地工作，使计算机系统的所有资源最大限度地发挥作用，为用户提供方便、有效、友好的服务界面。

操作系统的作用主要体现在以下 3 个方面：

（1）操作系统作为用户与计算机硬件系统之间的接口。

操作系统作为用户与计算机硬件系统之间的接口的含义是操作系统处于用户与计算机硬

件系统之间，用户通过操作系统来使用计算机系统。操作系统作为接口的示意如图 3-3 所示。

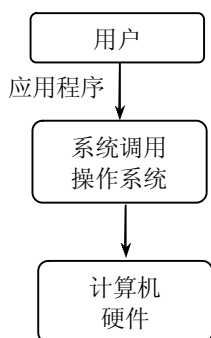


图 3-3 操作系统与硬件的关系

(2) 操作系统作为计算机系统的资源管理者。

在操作系统中，能分配给用户使用的硬件和软件设施统称为资源，包括两大类：硬件资源和信息资源。硬件资源又分为处理器、存储器、I/O 设备等；信息资源又分为程序和数据等。

操作系统作为计算机系统的资源管理者的重要任务之一是有序地管理计算机中的硬件、软件资源，跟踪资源使用情况，监视资源的状态，满足用户对资源的需求，协调各程序对资源的使用冲突；让用户简单、有效地使用资源，最大限度地实现各类资源的共享，提高资源利用率，从而提高计算机系统的效率。

(3) 操作系统为用户提供虚拟计算机。

操作系统是紧靠硬件的第一层软件，计算机上安装操作系统后，可扩展其基本功能，为用户提供一个功能显著增强、使用更加方便、安全可靠性好、效率明显提高的机器，称为虚拟计算机或操作系统虚拟机 (Virtual Machine)。

由此可知，裸机装上操作系统后，它将磁盘抽象成一组命名的文件，用户通过文件操作，按文件名来存取信息，不必涉及诸如数据物理地址、磁盘记录命令、移动磁头臂、搜索物理块及设备驱动等物理细节，便于使用且效率高。

2. 常用操作系统简介

下面简单介绍一下影响极为广泛的操作系统。

(1) DOS 和 Windows 系列。

DOS 系统是 1981 年 Microsoft 公司为 IBM 个人计算机开发的，即 MS-DOS。它是一个单用户单任务的操作系统，用户界面为命令行形式。在一段时间里 DOS 是个人计算机上使用最广泛的一种操作系统，功能集中在磁盘管理和其他外设的管理方面。

Windows 是 Microsoft 公司研发的另一个操作系统。Windows 版本的发展历史如表 3-2 所示。

表 3-2 Windows 版本的发展历史

操作系统名称	发布日期	类型
Windows 1.0	1983.10	桌面操作系统
Windows 2.0	1987.10	桌面操作系统
Windows 3.0	1990.5	桌面操作系统

续表

操作系统名称	发布日期	类型
Windows 3.1	1992.4	桌面操作系统
Windows NT workstation 3.5	1994.7	桌面操作系统
Windows NT 3.5x	1994.9	服务器操作系统
Windows 95	1995.8	桌面操作系统
Windows NT workstation 4.x	1996.7	桌面操作系统
Windows NT Server 4.0	1996.9	服务器操作系统
Windows 98	1998.6	桌面操作系统
Windows 2000	2000.2	桌面操作系统
Windows 2000 Server	2000.2	服务器操作系统
Windows XP	2001.10	桌面操作系统
Windows Vista	2007.1	桌面操作系统

2001年10月微软推出操作系统 Windows XP。Windows XP 有家庭版、专业版、媒体中心版、平板 PC 版和 64 位版本等多种，它有丰富的音频、视频和网络通信功能，工作更加可靠，最大可支持 4GB 内存和两个 CPU。此外，它还增强了防病毒功能，增加了系统安全措施（如 Internet 防火墙、文件加密等）。

（2）UNIX 和 Linux。

UNIX 是使用比较广泛、影响比较大的主流操作系统之一。1969 年在 AT&T 的贝尔实验室诞生。UNIX 操作系统几乎全部使用 C 语言编写，是第一个主要用高级语言编写的系统软件。以功能强大、简洁、极其稳定、易于移植等优点，迅速得到学术界和业界的一致肯定。

Linux 是可以运行在 PC 机上的免费的 UNIX 操作系统。它是由芬兰赫尔辛基大学的学生 Linus Torvalds 在 1991 年开发的，作为自己的操作系统课程设计成果，在互联网上发布。由于 Linux 是个免费软件，源代码完全公开，加上互联网的传播作用，世界各地有相同爱好的人们纷纷加入到后续的发展进程中。

UNIX 和 Linux 主要安装在巨型机、大型机上作为网络操作系统使用，也可用于工作站和嵌入式系统。

3.2.2 操作系统的功能

操作系统的主要任务是有效地管理系统资源、提供友好便捷的用户接口。为实现其主要任务，操作系统具有以下五大功能：处理机管理、存储器管理、设备管理、文件系统管理和接口管理。下面简单介绍一下处理机管理和存储器管理。

1. 处理机管理

由于存在多个程序共享系统资源的事实，必然会引发对处理机（CPU）的争夺。如何有效地利用处理机资源，如何在多个请求处理机的进程中选择取舍，这都是进程调度要解决的问题。处理机是计算机中宝贵的资源，能否提高处理机的利用率，改善系统性能，在很大程度上取决于调度算法的好坏。因此，进程调度成为操作系统的核心。在操作系统中负责进程调度的程序称为进程调度程序。

(1) 进程调度程序的功能。在进程调度过程中, 由于多个进程需要循环使用 CPU, 所以进程调度是操作系统中最频繁的工作。调度程序一般采用按时间片(如 1/20s)轮转的策略, 即每个程序都能轮流得到一个时间片的 CPU 时间, 在时间片用完之后, 调度程序再把 CPU 交给下一个程序, 就这样一遍遍循环下去。只要时间片结束, 不管程序多么重要, 正在执行的程序就会被强行暂时终止。

(2) 进程调度方式。调度方式分为非剥夺式和剥夺式(抢占式)两种。非剥夺式调度是让正在执行的进程继续执行, 直到该进程完成或发生其他事件, 才移交 CPU 控制权; 剥夺式调度是当“重要”的或“系统”的进程出现时, 便立即暂停正在执行的进程, 将 CPU 控制权分配给“重要的”或“系统”的进程。剥夺式调度反映了进程优先级的特征及处理紧急事件的能力。

2. 存储器管理

存储管理基本技术包括分区法、可重定位分区法、覆盖技术、交换技术和虚拟存储技术, 主要内容包括内存的分配和回收、内存的共享和保护、内存自动扩充等。现在, 操作系统一般都采用虚拟存储技术(也称为虚拟内存技术, 简称虚存)进行存储管理。

(1) 虚拟存储器的引入。

用户程序运行时, 常常因为内存容量不足, 致使程序无法运行。考虑从物理上增加内存容量会受到机器自身的限制, 而且要增加系统成本, 所以应考虑从逻辑上扩充内存容量。这正是虚拟存储技术所要解决的主要问题。

虚拟存储技术是相对于“实存”的另一种存储管理技术。它使用软件方法扩充存储器, 20世纪70年代以后这一技术被广泛采用。虚拟存储器是指一种实际上并不存在的虚拟存储器, 它能提供给用户一个比实际内存大得多的存储空间, 使用户在编制程序时可以不考虑存储空间的限制。

在虚拟管理中, 把程序访问的逻辑地址称为“虚拟地址”, 而把处理器可直接访问的主存地址称为“实在地址”。虚拟地址的集合称为“虚拟地址空间”, 把计算机主存称为“实在地址空间”。程序和数据所在的虚拟地址必须放入主存的实在地址中才能运行。因此要建立虚拟地址和实在地址的相应关系, 这种地址转换由动态地址映像机构来实现。

当把虚拟地址空间与主存地址空间分开以后, 这两个地址空间的大小就独立了, 也就是说虚拟地址空间可以远大于主存的实在地址空间。另一个相关的问题是作业运行时其整个虚拟地址空间是否必须全部调入主存中, 如果必须的话, 那么实在地址空间仍必须大于虚拟地址空间。但实际情况是程序中有些部分是不用的(如错误处理程序), 有些程序用得很少(如程序中启动和终止处理部分), 即使经常使用的部分也可以只将最近要执行的部分装入内存, 其他部分到要用到的时候再调入内存, 而这时又可以把暂时不用的部分调出内存, 这一情况使虚拟存储管理技术有实现的可能。

操作系统把各级存储器统一管理起来, 把一个程序当前正在使用的部分放在磁盘上, 就启动执行它。操作系统根据程序执行时的要求和内存的实际使用情况随机地对每一个程序进行换入/换出。这样, 就给用户提供了一个比真实的内存空间大得多的地址空间, 这就是虚拟存储器, 也就是用户能作为可编址内存对待的存储空间, 在这种计算机系统中虚拟地址被映像成实在地址。

(2) 虚拟存储器受到的限制。

机器语言很难记忆, 极难使用, 而且不同系列的 CPU 具有不同的机器语言, 这样程序就无法在使用不同种类 CPU 的计算机上面运行, 也就是没有可移植性。

(2) 汇编语言。为了使程序设计变得容易一些, 人们定义了汇编语言。主要特征是机器指令符号化, 即用助记符的形式来表示机器指令的成分, 这就比使用二进制数的表示形式好得多。如用 ADD 表示加法, SUB 表示减法, MOV 表示传送数据等。

(3) 高级语言。所谓“高级”是指语言记号形式完全脱离机器指令, 很接近人们已经习惯的自然语言和数学语言, 看上去像英语句子和算术式子。这样不但易于理解, 更为重要的是高级语言和特定的 CPU 指令集在形式上不再关联。高级语言不但和具体的计算机无关, 甚至和计算机内部的基本技术概念都无关。

程序设计语言是人和机器的信息交流, 因此高级语言要围绕一组人为的规则来构造, 语法规则数量很少而且必须十分严格地定义, 所以是一种形式语言。

2. 常用的程序设计语言介绍

(1) C/C++语言。C语言是在20世纪70年代初问世的。1978年美国电话电报公司(AT&T)贝尔实验室正式发表了C语言。早期的C语言主要是用于UNIX系统。C语言是一种结构化语言。它层次清晰, 便于按模块化方式组织程序, 易于调试和维护。C语言的表现能力和处理能力极强。它具有丰富的运算符和数据类型, 便于实现各类复杂的数据结构。由于C语言实现了对硬件的编程操作, 因此C语言集高级语言和低级语言的功能于一体。既可用于系统软件的开发, 也适合于应用软件的开发。此外, C语言还具有效率高、可移植性强等特点。因此广泛地移植到了各种类型的计算机上, 从而形成了多种版本的C语言。

在C语言的基础上, 1983年又由贝尔实验室的Bjarne Stroustrup推出了C++。C++进一步扩充和完善了C语言, 成为一种面向对象的程序设计语言。C++提出了一些更为深入的概念, 它所支持的这些面向对象的概念容易将问题空间直接映射到程序空间, 为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法。C是C++的基础, C++语言和C语言在很多方面是兼容的。因此, 掌握了C语言, 再进一步学习C++就能以一种熟悉的语法来学习面向对象的语言, 从而达到事半功倍的目的。

(2) Java语言。Java是Sun公司1995推出的一种面向对象的编程语言。它是一种通过解释方式来执行的语言, 语法规则和C++类似。同时, Java也是一种跨平台的程序设计语言。用Java语言编写的程序叫做Applet(小应用程序), 用编译器将它编译成类文件后, 将它存在WWW页面中, 并在HTML文档上作好相应标记, 用户端只要装上Java的客户软件就可以在网上直接运行Applet。Java非常适合于企业网络和Internet环境, 现在已成为Internet中最受欢迎、最有影响的编程语言之一。Java有许多值得称道的优点, 如简单、面向对象、分布式、解释性、可靠、安全、结构中立性、可移植性、高性能、多线程、动态性等。Java摒弃了C++中各种弊大于利的功能和许多很少用到的功能。Java可以运行在任何微处理器上, 用Java开发的程序可以在网络上传输, 并运行于任何客户机上。

(3) FORTRAN语言。FORTRAN是英文FORmula TRANslator的缩写, 译为“公式翻译器”, 它是世界上最早出现的计算机高级程序设计语言, 广泛应用于科学和工程计算领域。FORTRAN语言以其特有的功能在数值、科学和工程计算领域发挥着重要作用。

FORTRAN语言开始是为解决数学问题和科学计算而提出的, 多年来的应用表明: 由于FORTRAN本身具有标准化程度高、便于程序互换、较易优化、计算速度快等优点, 因此这种

高级语言目前已广泛流行。国外几乎所有的计算机厂商都能向用户提供 FORTRAN 的编译程序和应用程序的版本。

除了上面介绍的几种常用程序语言外，具有影响的程序语言还有 LISP 语言（适用于符号操作和表处理，主要用于人工智能领域）、PROLOG 语言（一种逻辑式编程语言，主要用于人工智能领域）、Ada 语言（一种模块化语言，且易于控制并行任务和处理异常情况）、MATLAB（一种面向向量和矩阵运算的提供数据可视化等功能的数值计算语言）等，请读者参阅相关书籍。

3.3.2 程序设计语言中的控制成分

高级语言种类千差万别，但是基本成分可以归纳为 4 种：

- （1）数据成分，用以描述程序所涉及的数据，如对数据类型和数据结构进行说明。
- （2）运算成分，用以描述程序中所包含的运算，如算术表达式和逻辑表达式等。
- （3）控制成分，用以描述程序中所包含的流程控制，如条件语句和循环语句等。
- （4）传输成分，用以表达程序中数据的传输，如 I/O 语句。

下面着重介绍控制成分，另 3 种成分请参阅相关资料。

程序设计语言中控制成分的作用是提供一个基本框架，在此基本框架下，可以将数据和运算对数据的运算组合成程序。3 种基本控制结构是顺序结构、条件选择结构和循环结构。

1. 顺序结构

顺序结构表示操作步骤按时间顺序依次执行。如图 3-4 所示，首先执行 S1，然后再执行 S2。图 3-5 表示了由 4 个顺序执行的动作构成的烧水喝茶过程。

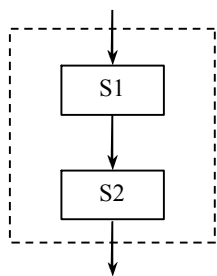


图 3-4 顺序结构

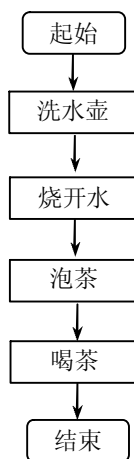


图 3-5 烧水喝茶流程图

2. 条件选择结构

条件选择结构表示对一个条件的取值进行判断，然后选择执行哪一个操作。如图 3-6 所示，当条件 P 的结果为“真”时，选择执行 S1；结果为“假”时，选择执行 S2。条件选择结构里，往往以逻辑表达式来表示一种判断条件。在 C 语言中，if 语句是典型的条件选择结构，它的表示形式为：

```
if(P) S1; else S2;
```

3. 循环结构

循环结构又称为重复结构。在这种结构里，一组操作反复地执行若干次。重复执行的操作叫循环体，通过循环控制条件的设定来控制循环体的重复执行次数。循环结构有多种形式，这里以 while 型重复结构为例。在 C 语言中，while 结构的一般形式为：

```
while(P) S;
```

如图 3-7 所示，当 P（循环控制条件）成立时，就重复执行 S（循环体）；直到 P 为假时，结束重复操作。

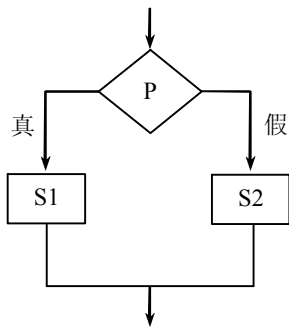


图 3-6 条件选择结构

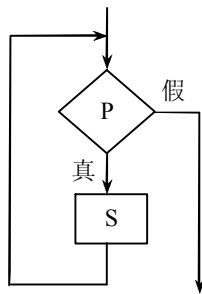


图 3-7 while 型循环结构

3.3.3 程序设计语言的处理过程

显然，除了机器语言之外，任何其他形式的程序设计语言都要经历一个翻译过程，被翻译的对象称为源程序，结果称为目标程序。而翻译是由一些系统软件来完成的。

把用汇编语言编写的一个源程序转变为机器语言表示的目标程序所使用的翻译程序叫做汇编程序。汇编的结果是可以直接在 CPU 上运行的机器程序。不少人习惯把汇编语言写的程序也叫做“汇编程序”，读者一定要把它和起翻译作用的汇编程序区别理解。

高级语言的通常处理过程是：用一个叫做编译程序的系统软件来进行翻译，把得到的机器语言目标程序再交给另一个语言处理软件——连接程序处理，连接上一些标准的程序段，才能产生完整的、可以在 CPU 上运行的机器程序。有时，可执行的目标程序还要在运行之前交给装入程序处理，以决定最后的运行细节。

高级语言的另一种可能的翻译方式称为解释。一个叫解释程序的系统软件可以逐句地分析源程序，随即产生对应的机器指令序列并执行它。高级语言的这种处理方式称为解释执行。

比较这两种语言处理方法，采用编译方法时程序的执行时间效率要高得多。表面上，编译和连接要花费“额外的”时间才能得到目标程序。但可执行的目标程序一旦产生，就可以在磁盘上保存起来，以后每当需要，就装入内存直接执行。而采用解释方法时，没有产生目标程序，每次执行程序都要重复地对源程序进行解释，耗费的时间要多得多。



练一练

1. (判断题) 任何高级语言编写的程序都必须经过转换后才能由计算机执行。

2. (判断题) 汇编语言是面向计算机指令系统的, 因此汇编语言程序可以由计算机直接执行。
3. (单选题) 以下关于高级程序设计语言中数据成分的说法中, 错误的是 ()。
 - A. 数据的名称用标识符来命名
 - B. 数组是一组相同类型数据元素的有序集合
 - C. 指针变量中存放的是某个数据对象的地址
 - D. 程序员不能自己定义新的数据类型
4. (单选题) 高级程序设计语言的编译程序和解释程序属于下列 () 类程序。
 - A. 语言编辑
 - B. 程序连接
 - C. 文本编辑
 - D. 语言处理
5. (填空题) 高级程序设计语言种类繁多, 但其基本成分可归纳为数据成分、控制成分等 4 种, 其中算术表达式属于_____成分。

3.4 算法与数据结构基础

瑞士计算机科学家尼克劳斯·沃思(Niklaus Wirth)提出了“数据结构+算法=程序”这一著名公式。数据结构就是计算机存储数据的结构, 算法是用来操作这些数据结构的, 即各种数据结构的设计是以算法的实现为依据的。在以前的学习中, 我们知道“程序是软件的核心”, 实际上“算法是程序的核心”。因为要使计算机解决某个问题, 首先必须针对该问题设计一个解题步骤, 然后再据此编写程序并交给计算机执行。这里所说的解题步骤就是“算法”, 采用某种程序设计语言对问题的对象和解题步骤进行的描述就是程序。

3.4.1 算法

1. 算法的基本概念

算法(Algorithm)是在有限步骤内求解某一问题所使用的一组定义明确的规则。通俗地说, 就是计算机解题的过程。在这个过程中, 无论是形成解题思路还是编写程序都是在实施某种算法。前者是推理实现的算法, 后者是操作实现的算法。一个算法应该具有以下 5 个重要的特征:

- (1) 有穷性: 算法必须在有限时间内做完, 即算法必须在执行有限步骤之后结束。
- (2) 确定性: 算法的每一步骤必须有确切的定义, 不允许有模棱两可的解释和多义性。
- (3) 输入: 一个算法有 0 个或多个输入, 以刻画运算对象的初始情况, 所谓 0 个输入是指算法本身给定了初始条件。
- (4) 输出: 一个算法有一个或多个输出, 以反映对输入数据加工后的结果。没有输出的算法是毫无意义的。
- (5) 能行性: 算法中有待实现的操作都是可执行的, 即在计算机的能力范围之内, 且在有限的时间内能够完成。

2. 算法表示

算法的表示一般有三种形式: 自然语言、伪代码和流程图。

- (1) 自然语言: 是指人们日常使用的语言, 如汉语、英语或其他语言。
- (2) 伪代码: 是用介于自然语言和计算机语言之间的文字和符号(包括数学符号)来描述算法。

(3) 流程图：是广泛使用的一种算法表示工具，用规定的图形符号来表示要执行的各种操作步骤、用流线表示操作步骤的转移次序，从而描述出算法过程。因为图形符号主要是各种不同形状的图线框，所以业界又习惯把流程图称为“框图”。流程图也可以用来描写程序的操作过程，所以有时也叫做程序流程图。

一个问题的解决往往可以有多种不同的算法。算法的好坏，除考虑正确性外，还应该考虑以下因素：

- 执行算法所要占用的计算机资源是否最少，包括时间资源和空间资源两个方面。
- 算法是否容易理解、是否容易调试和测试等。

3.4.2 数据结构

1. 数据结构的定义

数据是对客观事物的符号表示，在计算机科学中是指所有能输入到计算机中并被计算机程序处理的符号的总称。

数据结构是指相互之间存在一种或多种特定关系的数据元素的集合，即数据的组织形式。

数据结构作为计算机的一门学科，主要研究和讨论以下3个方面的内容：

- (1) 数据集中各个数据元素之间所固有的逻辑关系，即数据的逻辑结构。
- (2) 在数据元素进行处理时，各数据元素在计算机中的存储关系，即数据的存储结构。
- (3) 对各种数据结构进行的运算。

讨论以上问题的目的是为了提提高数据处理的效率，所谓提高数据处理的效率是指提高数据处理的速度和尽量节省在数据处理过程中所占用的计算机存储空间。

一般情况下，在具有相同特征的数据元素集合中，各个数据元素之间存在有某种关系（即连续），这种关系反映了该集合中数据元素所固有的一种结构。在数据处理领域中，通常把数据元素之间这种固有的关系简单地用前后件关系（或直接前驱与直接后继关系）来描述。

2. 数据的逻辑结构

数据结构是指反映数据元素之间关系的数据元素集合的表示，更通俗地讲，数据结构是指带有结构的数据元素的集合。所谓结构实际上就是指数据元素之间的前后件关系。

一个数据结构应该包含两方面信息：表示数据元素的信息和表示各数据元素之间的前后件关系的信息。

数据的逻辑结构是对数据元素之间的逻辑关系的描述，它可以用一个数据元素的集合和定义在此集合中的若干关系来表示。

数据的逻辑结构包括集合结构、线性结构、树型结构和图型结构4种，如图3-8所示。

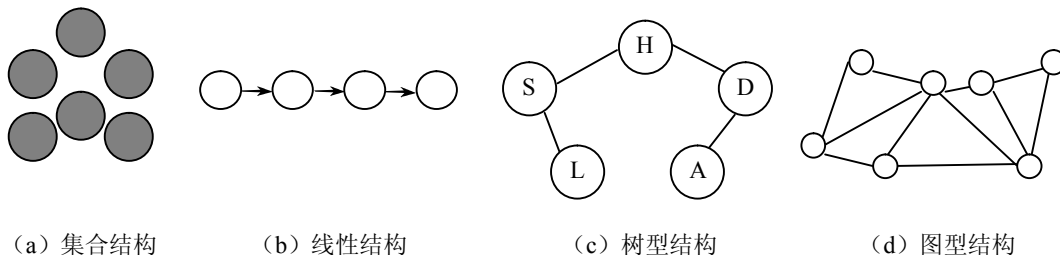


图 3-8 4 种基本数据结构

3. 数据的存储结构

数据的逻辑结构在计算机存储空间中的存放形式称为数据的存储结构（也称数据的物理结构）。

数据元素在计算机存储空间中的位置关系可能与逻辑关系不同，为了表示存放在计算机存储空间中的各数据元素之间的逻辑关系（即前后件关系），在数据的存储结构中，不仅存放各数据元素的信息，还需要存放各数据元素之间的前后件关系的信息。

一种数据的逻辑结构根据需要可以表示成多种存储结构，常用的存储结构有顺序、链接、索引等存储结构。而采用不同的存储结构，其数据处理的效率是不同的，因此，在进行数据处理时，选择合适的存储结构是很重要的。

4. 数据的运算

为数据处理的需要，需要在数据上进行各种运算。数据的运算是定义在数据的逻辑结构上的，但运算的具体实现要在存储结构上进行。数据的各种逻辑结构有相应的各种运算，每种逻辑结构都有一个运算的集合。下面列举几种常用的运算。

(1) 检索。在数据结构里查找满足一定条件的节点。

(2) 插入。往数据结构里增加新的节点。

(3) 删除。把指定的节点从数据结构里去掉。

(4) 更新。改变指定节点的一个或多个域的值。

(5) 排序。保持线性结构的节点序列里的节点数不变，把节点按某种指定的顺序重新排列。

例如，按节点中某个域的值由小到大对节点进行排列。

数据的运算是数据结构的一个重要方向。讨论任何一种数据结构时都离不开对该结构上的数据运算及其实现算法的讨论。



练一练

1. (判断题) 数据结构一般包括 3 个方面的内容，即数据的逻辑结构、数据的存储结构及在这些数据上定义的运算。

2. (判断题) 程序设计的核心是算法设计。

3. (单选题) 一个算法应包含至少 () 输出。

A. 零个 B. 一个 C. 一个以上 D. 多个

4. (单选题) 抽象地反映数据元素之间的约束关系而不考虑其在计算机中的存储方式，称为数据的 ()。

A. 存储结构 B. 层次结构 C. 物理结构 D. 逻辑结构

5. (填空题) 若求解某个问题的程序要反复多次执行，则在设计求解算法时应重点从 _____ 代价上考虑。

3.5 数据库设计基础

数据库技术是 20 世纪 60 年代末在文件系统基础上发展起来的数据管理技术，是计算机

科学的重要分支。经过近 40 年的发展，现在已经形成相当规模的理论体系和应用领域，不仅应用于事务处理，而且进一步应用到了情报检索、人工智能、专家系统、计算机辅助设计等各个领域。本节将介绍数据库的基本概念、SQL 等基础知识。

3.5.1 数据库系统的基本概念

数据库技术所研究的问题就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。数据库系统由数据库、数据库管理系统、数据库管理员、数据库应用程序、用户 5 个部分组成。

(1) 数据库 (DataBase, DB): 是数据的集合，具有统一的结构形式并存放于统一的存储介质内，是多种应用数据的集成，并可被各个应用程序共享。数据库存放数据是按数据所提供的数据库模式存放的，具有集成与共享的特点。

(2) 数据库管理系统 (DataBase Management System, DBMS): 是一种系统软件，负责数据库中的数据组织、数据操纵、数据维护、控制及保护和数据服务等，是数据库的核心。

(3) 数据库管理员 (DataBase Administrator, DBA): 是对数据库进行规划、设计、维护、监视等的专业管理人员。

(4) 数据库应用程序: 数据库应用程序是使用数据库语言开发的、能够满足数据处理需求的应用程序。

(5) 用户: 用户可以通过数据库管理系统直接操纵数据库，或者通过数据库应用程序来操纵数据库。

数据库系统的 5 个部分及其相互关系如图 3-9 所示。

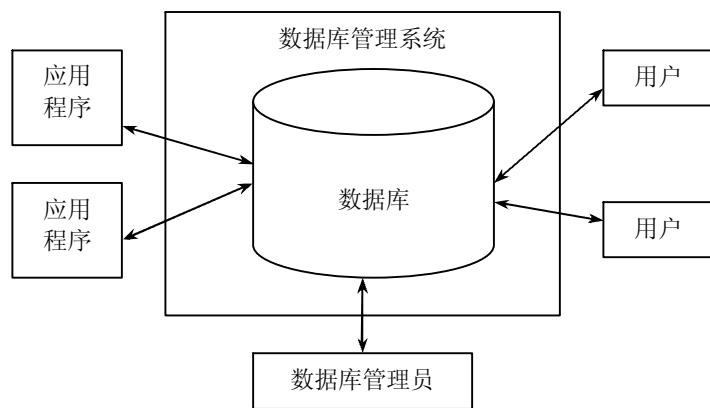


图 3-9 数据库系统的组成

3.5.2 数据模型

模型是对现实世界的抽象。在数据库技术中，用模型来描述数据库的结构和语义，对现实世界进行抽象。表示实体类型及实体之间关系的模型称为“数据模型”。

1. 数据模型的基本概念

在数据库中用数据模型来抽象、表示和处理现实世界中的数据和信息。通俗地讲，数据模型就是现实世界的反映，它包括把现实世界中的客观对象抽象为概念模型；把概念模型转换

为某一 DBMS 支持的数据模型这样两个阶段。

用户为了某种需要，需要将现实世界中的部分需求用数据库实现，这样我们所见到的是在客观世界中划定边界的一部分环境，它称为现实世界。通过抽象为现实世界进行数据库级上的刻画所构成的逻辑模型叫信息世界。在信息世界基础上致力于其在计算机物理结构上的描述，从而形成的物理模型叫计算机世界。

数据模型所描述的内容包括数据结构、数据操作和数据约束，即数据模型的三要素。

(1) 数据模型中的数据结构主要描述数据的模型、内容、性质以及数据间的联系等。数据结构是刻画数据模型性质最重要的方面。因此，在数据库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。数据结构是对系统静态特性的描述。

(2) 数据模型中的数据操作主要描述在相应数据结构上的操作类型与操作方式。数据库主要有检索和更新（包括插入、删除、修改）两大类操作。

(3) 数据模型中的数据约束主要描述数据结构内数据间的语法、语义联系，它们之间的制约与依存关系，以及关系动态变化的规则保证了数据的正确性、有效性和相容性。数据模型按不同的应用层次分成 3 种类型：概念数据模型、逻辑数据模型、物理数据模型。

2. E-R 模型

概念模型是面向现实世界的，它的出发点是有效和自然地模拟现实世界，给出数据的概念化结构。长期以来被广泛使用的概念模型是 E-R 模型（Entity Relationship Model，实体关系模型），它于 1976 年由 Peter Chen 首先提出。该模型将现实世界的要求转化成实体、联系、属性等几个概念，以及它们间的两种基本联结关系，并且可以用一种图非常直观地表示出来。

(1) E-R 模型的基本概念。

①实体：现实世界中的事物可以抽象成实体，实体是概念世界中的基本单位，它们是客观存在的且又能相互区别的事物。由具有共同属性的实体组成的集合称为实体集。

②属性：现实世界中的事物均有一些特性，这些特性可以用属性来表示，属性刻画了实体的特征。一个实体往往可以有若干属性，每个属性可以有值，一个属性的取值范围称为该属性的值域或值集。

③码：唯一标识实体的属性（集）称为该属性的码。

④域：属性的取值范围称为该属性的域。

⑤实体型：具有相同属性的实体必然具有共同的特征和性质。用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。

⑥实体集：同型实体的集合称为实体集。

⑦联系：在现实世界中事物间的关联称为联系。在概念世界中联系反映了实体集间的一定关系。

两个实体集间的联系实际上是实体集间的函数关系，这种函数关系有以下几种（如图 3-10 所示）：

- 一对一的联系，简记为 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。

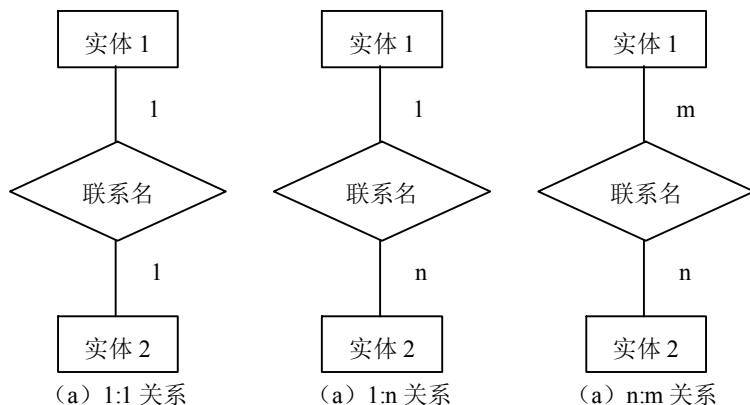


图 3-10 两个实体间的函数关系

- 一对多或多对一联系，简记为 $1:N$ ($1:n$) 或 $N:1$ ($n:1$)。如果对于 A 中的每一个实体，实体集 B 中有 n 个实体 ($n \geq 0$) 与之联系，反之，对于实体集 B 中的每一个实体，实体集 A 中至多只有一个实体与之联系，则称实体集 A 与实体集 B 有一对多联系。反之即为多对一联系。
- 多对多联系，简记为 $M:N$ 或 $m:n$ 。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 ($n \geq 0$) 与之联系，反之，对于实体集 B 中的每一个实体，实体集 A 中有 m 个实体 ($m \geq 0$) 与之联系，则称实体集 A 与实体集 B 具有多对多联系。

(2) 实体、联系、属性之间的联结关系。

① 实体(集)与属性间的联结关系。实体是概念世界中的基本单位，属性附属于实体，它本身并不构成独立单位。一个实体可以有若干个属性，实体以及它的所有属性构成了实体的一个完整描述。

属性有属性域，每个属性可取属性域内的值。

实体有型与值之别，一个实体的所有属性构成了这个实体的型，相同的实体构成了实体集。

② 实体(集)与联系。实体(集)间可通过联系建立联结关系，一般而言，实体(集)间无法建立直接关系，它只能通过联系才能建立起联结关系。

(3) E-R 模型的图示法。

E-R 模型用 E-R 图来表示。

① 实体集表示法。在 E-R 图中用矩形表示实体集，在矩形内写上该实体集的名字。如实体集学生 (student)、课程 (course) 可用如图 3-11 所示来表示。

② 属性表示法。在 E-R 图中用椭圆形表示属性，在椭圆形内写上属性的名称。如学生有姓名 (S_n)、年龄 (S_a) 等属性，它们可以通过如图 3-12 所示来表示。

③ 联系表示法。在 E-R 图中用菱形表示联系，菱形内写上联系名。如学生与课程间有联系 SC，可用如图 3-13 所示来表示。

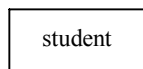


图 3-11 实体集表示法

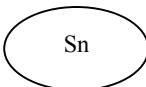
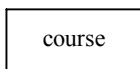


图 3-12 属性表示法

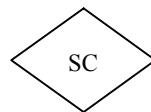
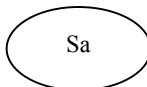


图 3-13 联系表示法

④实体集（联系）与属性间的联结关系。在 E-R 图中，实体集与属性间的联系可用联结这两个图形间的无向线段表示。如实体集 student 有属性姓名（Sn）和年龄（Sa）；实体集 course 有属性 Cn（课程号）和 Cname（课程名），此时它们可用如图 3-14 所示来表示。

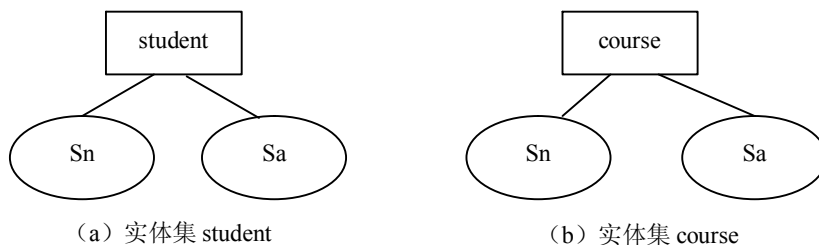


图 3-14 实体集的属性间的联结

⑤实体集与联系间的联结关系。在 E-R 图中实体集与联系间的联结关系可用联结这两个图形间的无向线段表示。实体集 student 与联系 SC 间有联结关系，实体集 course 与联系 SC 间也有联结关系，因此它们之间可用无向线段相联，构成一个如图 3-15 所示的图。

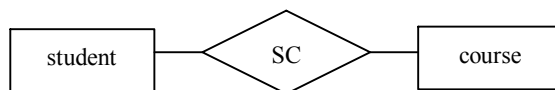


图 3-15 实体集与属性间的联结关系

有时为了进一步刻画实体间的函数关系，还可在线段边上注明其对应的函数关系，如图 3-16 所示。

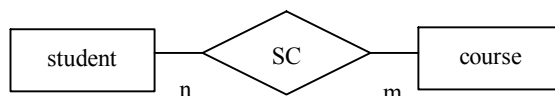


图 3-16 实体集间的关系标示图

3. 关系模型

(1) 关系模型数据结构。

关系模型采用二维表来表示，简称表，如表 3-3 所示。二维表由表框架及表的元组组成。表框架由 n 个命名的属性组成， n 称为属性元数。每个属性有一个取值范围，称为值域。

表 3-3 关系模型数据结构

学号	姓名	性别	年龄
2001001	张三	男	18
2001002	李霞	女	19
2001003	王强	男	19
2001004	生紫烟	女	20

在关系模型中，实体与实体间联系的表示方法是，实体型直接用关系表表示；属性用属性名表示；一对一联系隐含在实体对应的关系中；一对多联系隐含在实体对应的关系中；多对多联系直接用关系表示。

二维表的特性是，二维表中元组个数是有限的——元组个数有限性；二维表中元组均不相同——元组的唯一性；二维表中元组的次序可以任意交换——元组的次序无关性；二维表中元组的分量是不可分割的基本数据项——元组分量的原子性；二维表中属性名各不相同——属性名唯一性；二维表中属性与次序无关，可任意交换——属性的次序无关性；二维表属性的分量具有与该属性相同的值域——分量值域的统一性。

关系模型的优点是，由于建立在严格数学概念的基础上，它的概念单一，数据结构简单、清晰，用户易懂易用；实体和各类联系都用关系表示；对数据的检索结果也是关系；关系模型的存取路径对用户透明；具有更高的数据独立性，更好的安全保密性；简化了程序员的工作。

关系模型的缺点是，存取路径对用户透明导致查询效率往往不如非关系模型，关系数据库模型为提高性能，必须对用户的查询请求进行优化，增加了开发数据库管理系统的难度。

(2) 关系操纵。

关系操纵包括数据查询、数据删除、数据插入和数据修改。

对一个关系内查询的基本单位是元组分量，其基本过程是先定位后操作；对多个关系间的数据查询则可分为3步，即将多个关系合并成一个关系，对合并后的一个关系作定位，最后进行操作。

数据删除可分为一个关系内的元组选择与和系中元组删除两个基本操作。

数据插入是指在指定关系中插入一个或多个元组。

数据的修改是在一个关系中修改指定的元组与属性。它可以分解为删除需要修改的元组和插入修改后的元组两个更基本的操作。

关系模型的基本操作包括关系的属性指定、关系的元组选择、两个关系合并、一个或多个关系的查询、关系中元组的插入和关系中元组的删除。

(3) 关系中的数据约束。

关系中的数据约束包括实体完整性约束、参照完整性约束和用户定义的完整性约束等。

实体完整性约束要求关系主键中的属性值不能为空，这是数据库完整性的最基本要求。

参照完整性约束是关系之间相关联的基本约束，它不允许关系引用不存在的元组，即在关系中的外键要么是所关联关系中实际存在的元组，要么就为空值。

用户定义的完整性约束针对某一具体关系数据库的约束条件，它反映某一具体应用所涉及的数据必须满足的语义要求。

任何关系数据库系统都应该支持实体完整性和参照完整性。

关系模型的二维表简单易懂，用户只需要用简单的查询语句就可以对数据进行操作，并不涉及存储结构、访问技术等细节。关系模型是数学化的模型，要用到集合论、离散数学等知识。SQL语言是关系数据库的代表性语言，现在已经成为关系数据库的标准查询语言。

3.5.3 关系代数

关系代数是一种抽象的查询语言，是关系数据操纵语言的一种传统表达方式，它用对关系的运算来表达查询。任何一种运算都是将一定的运算符作用于一定的运算对象上，得到预期

的运算结果，所以运算对象、运算符和运算结果是运算的三大要素。

关系代数操作简称为关系操作。它们可分为两类：一类是传统的集合操作，另一类是关系专用的操作。关系操作的结果仍为关系，可以再参与其他关系操作，构成关系代数表达式，由此描述对关系的各种复杂操作。

1. 传统的集合操作

传统的集合运算是二目运算，包括并、交、差、广义笛卡尔积四种运算。

(1) 并 (Union)。设关系 R 和关系 S 具有相同的目 n (即两个关系都有 n 个属性)，且相应的属性取自同一个域，则关系 R 与关系 S 的并由属于 R 或属于 S 的元组组成。其结果关系仍为 n 目关系，记作：

$$R \cup S = \{t | t \in R \vee t \in S\}$$

(2) 差 (Difference)。设关系 R 和关系 S 具有相同的目 n，且相应的属性取自同一个域，则关系 R 与关系 S 的差由属于 R 而不属于 S 的所有元组组成。其结果关系仍为 n 目关系。记作：

$$R - S = \{t | t \in R \wedge t \notin S\}$$

(3) 交 (Intersection Referential Integrity)。设关系 R 和关系 S 具有相同的目 n，且相应的属性取自同一个域，则关系 R 与关系 S 的交由既属于 R 又属于 S 的元组组成。其结果关系仍为 n 目关系，记作：

$$R \cap S = \{t | t \in R \wedge t \in S\}$$

(4) 广义笛卡尔积 (Extended Cartesian Product)。两个分别为 n 目和 m 目的关系 R 和 S 的广义笛卡尔积是一个 (n+m) 列的元组的集合。元组的前 n 列是关系 R 的一个元组，后 m 列是关系 S 的一个元组。若 R 有 k1 个元组，S 有 k2 个元组，则关系 R 和关系 S 的广义笛卡尔积有 k1 × k2 个元组，如图 3-17 所示。

行标号	行号
row1	1
row2	2
row3	3

列标号	列号
arrange1	1
arrange2	2

R
S

行标号	行号	列标号	列号
row1	1	arrange1	1
row1	1	arrange2	2
row2	2	arrange1	1
row2	2	arrange2	2
row3	3	arrange1	1
row3	3	arrange2	2

R × S

图 3-17 广义笛卡尔积操作示例

2. 专门的关系操作

专门的关系运算包括选择、投影、连接、除等。

为了叙述上的方便，我们先引入几个记号。

- 设关系模式为 $R(A_1, A_2, \dots, A_n)$ 。它的一个关系设为 R 。 $t \in R$ 表示 t 是 R 的一个元组。 $t[A_i]$ 则表示元组 t 中相应于属性 A_i 的一个分量。
- 若 $A = \{A_{i1}, A_{i2}, \dots, A_{ik}\}$ ，其中 $A_{i1}, A_{i2}, \dots, A_{ik}$ 是 A_1, A_2, \dots, A_n 中的一部分，则 A 称为属性列或域列。 A 则表示 $\{A_1, A_2, \dots, A_n\}$ 中去掉 $\{A_{i1}, A_{i2}, \dots, A_{ik}\}$ 后剩余的属性组。 $t[A] = (t[A_{i1}], t[A_{i2}], \dots, t[A_{ik}])$ 表示元组 t 在属性列 A 上诸分量的集合。
- R 为 n 目关系, S 为 m 目关系。设 $t_r \in R, t_s \in S$, 则 $\widehat{t_r t_s}$ 称为元组的连接 (Concatenation)。

它是一个 $(n+m)$ 列的元组，前 n 个分量为 R 中的一个 n 元组，后 m 个分量为 S 中的一个 m 元组。

- 给定一个关系 $R(X, Z)$ ， X 和 Z 为属性组。我们定义，当 $t[X]=x$ 时， x 在 R 中的象集 (Images Set) 为：

$$Z_x = \{t[Z] \mid t \in R, t[X]=x\}$$

它表示 R 中属性组 X 上值为 x 的诸元组在 Z 上分量的集合。

(1) 选择 (Selection)。选择又称为限制 (Restriction)。它是在关系 R 中选择满足给定条件的诸元组，记作：

$$\sigma_F(R) = \{t \mid t \in R \wedge F(t) = \text{真}\}$$

其中 F 表示选择条件，它是一个逻辑表达式，取逻辑值“真”或“假”。

逻辑表达式 F 的基本形式为：

$$X_1 \theta Y_1 [\varphi X_2 \theta Y_2]$$

θ 表示比较运算符，它可以是 $>$ 、 \geq 、 $<$ 、 \leq 、 $=$ 或 \neq 。 X_1 、 Y_1 等是属性名或常量或简单函数。属性名也可以用它的序号来代替。 φ 表示逻辑运算符，它可以是 \neg 、 \wedge 或 \vee 。 $[\]$ 表示任选项，即 $[\]$ 中的部分可以要也可以不要， \dots 表示上述格式可以重复下去。

因此选择运算实际上是从关系 R 中选取使逻辑表达式 F 为真的元组。这是从行的角度进行的运算。

(2) 投影 (Projection)。关系 R 上的投影是从 R 中选择出若干属性列组成新的关系，记作：

$$\Pi_A(R) = \{t[A] \mid t \in R\}$$

其中 A 为 R 中的属性列。

(3) 连接 (Join)。连接包括 θ 连接、自然连接、外连接、半连接。它是从两个关系的笛卡尔积中选取属性间满足一定条件的元组。

连接运算从 R 和 S 的笛卡尔积 $R \times S$ 中选取 (R 关系) 在 A 属性组上的值与 (S 关系) 在 B 属性组上的值满足比较关系 θ 的元组。

连接运算中有两种最为重要也最为常用的连接，一种是等值连接 (equi-join)，另一种是自然连接 (Natural join)。

θ 为 “=” 的连接运算称为等值连接。它是从关系 R 与 S 的笛卡尔积中选取 A 、 B 属性值

相等的那些元组。 θ 连接示例如图 3-18 所示。

A	B	C
1	2	3
4	5	6
7	8	9

R

D	E
4	5
6	4

S

A	B	C	D	E
1	2	3	4	5
1	2	3	6	4
4	5	6	6	4

$R \bowtie S$
2<1

图 3-18 θ 连接示例

自然连接（Natural join）是一种特殊的等值连接，它要求两个关系中进行比较的分量必须是相同的属性组，并且要在结果中把重复的属性去掉。自然连接示例如图 3-19 所示。

学生姓名	奖励等级
张雷	2
王宁	1
周光明	1
李霞霞	3

R1

奖励等级	奖学金
1	3000
2	2000
3	1000
4	500

R2

学生姓名	奖励等级	奖学金
张雷	2	
王宁	1	
周光明	1	
李霞霞	3	

$R1 \bowtie R2$

图 3-19 自然连接示例

一般的连接操作是从行的角度进行运算，但自然连接还需要取消了重复列，所以是同时从行和列的角度进行运算。

（4）除（Division）。给定关系 $R(X,Y)$ 和 $S(Y,Z)$ ，其中 X 、 Y 、 Z 为属性组。 R 中的 Y 与 S 中的 Y 可以有不同的属性名，但必须出自相同的域集。 R 与 S 的除运算得到一个新的关系 $P(X)$ ， P 是 R 中满足下列条件的元组在 X 属性列上的投影：元组在 X 上分量值 x 的象集 Yx 包含 S 在 Y 上投影的集合。除法操作的示例如图 3-20 所示。

3.5.4 SQL 概述

SQL 是结构化查询语言（Structure Query Language）的英文缩写，它是一种基于关系运算理论的数据库语言。由于 SQL 所具有的特点，目前关系数据库系统大都采用 SQL 语言，使其成为一种通用的国际标准数据库语言。

SQL 是非过程化语言，所谓非过程化语言是指用户只需了解数据的逻辑模式，不必关心数据的物理存储细节，用户只要指出“做什么”，而无须指出“怎么做”，从而免除了用户描述操作过程的麻烦。由于 SQL 具有功能丰富、语言简洁、使用灵活等优点，因而受到广泛的欢

迎。众多的数据库产品的厂家纷纷推出了支持 SQL 的软件，并很快得到了应用和推广。目前，无论是哪一种数据库管理系统大都采用 SQL 作为共同的数据库存取语言和标准接口，如 DB2、Oracle、SQL Server、Access 等。

学号	姓名	课程号	课程名
C005	张雷	W001	英语
C005	张雷	W002	日语
C005	张雷	W003	德语
C008	王宁	W001	英语
C008	王宁	W003	德语
R098	钱欣	W001	英语
A041	周光明	W002	日语
A041	周光明	W003	德语
M038	李霞霞	W001	英语
A041	周光明	W001	英语

R: 学生—外语课程选修表

课程号	课程名
W001	英语
W002	日语
W003	德语

S: 外语课程开设表

学号	姓名
C005	张雷
A041	周光明

R ÷ S: 选修全部外语课程学生表

图 3-20 除法操作的示例

1. 常用的 SQL 命令

SQL 语言包含 9 个命令动词，分别用于数据定义、数据查询、数据操纵、数据控制，如表 3-4 所示。

表 3-4 SQL 语言的功能及常用命令

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

本节中只讲解数据定义、数据查询、数据操纵功能的使用，其他请参见 SQL 语言工具书。在 SQL 语言的语法格式中，<>表示必选项，[]表示可选项。所有标点符号均应为半角。每个 SQL 语句以分号 (;) 结束。

2. SQL 的数据定义

SQL 的数据定义语言具有创建表 (Table) 结构、修改表结构、删除表、建立与删除索引等功能，如表 3-5 所示。

表 3-5 SQL 的数据定义语言

SQL 语句	功能
CREATE TABLE	创建表结构
ALTER TABLE	修改表结构：在已有的表中添加新字段或限制条件
DROP	删除表，或从字段或字段组中删除一项索引
CREATE INDEX	创建字段或字段组的索引

SQL 语言用 CREATE TABLE 语句创建表，其语法格式如下：

```
CREATE TABLE <表名> (<字段名 1> <数据类型> [完整性约束条件]
[,<字段名 2> <数据类型>[完整性约束条件]...]);
```

说明：

(1) 表名：所创建表的名称。

(2) 表中定义字段的每一语句之间用逗号（,）隔开，最后一个语句不用逗号。

SQL 语言中的数据类型如表 3-6 所示（Access 2000 适用，在其他数据库软件中会有差别，请查阅相应的数据库软件使用指南）。

表 3-6 SQL 语言中的数据类型

数据类型	说明
Integer 或 Long	长整型
Single	单精度型
Double 或 Float	双精度型
Date, datetime	日期型
Currency	货币型
Text(n)、char(n)、String(n)	文本型，n 为字段大小
Logical	是/否类型
Memo	备注型
OLEObject	OLE 对象类型

(3) 完整性约束条件，指输入该字段值必须要遵守的规则，如设置字段不能为空或设置年龄字段值必须在 16~25 之间等。

3. SQL 的数据查询

建立数据库的主要目的是进行数据库查询，SQL 语言用 SELECT 语句实现数据库查询，其语法格式如下：

```
SELECT [ALL|DISTINCT] <目标字段表达式> [,<目标字段表达式>]
FROM <表名>[,<表名>]...
[WHERE <条件表达式 1>]
[GROUP BY <字段名 1> [HAVING<条件表达式 2>]]
[ORDER BY <字段名 2> [ASC|DESC]];
```

SELECT 语句的含义是：根据 WHERE 子句的条件表达式，从 FROM 子句指定的表中筛

选符合条件的记录，按 SELECT 子句中列出的目标字段表达式输出结果。

4. SQL 的数据操纵

数据库中的数据不是一成不变的，而是每时每刻都在发生着变化。例如，在某网站中，每当有新用户登录网站进行注册时，都需要将新用户的注册数据插入到“用户注册数据”表中，或当已注册的用户的联系方式变了，我们需要对已存在的数据进行修改（或称为更新）等。这些使数据表中的数据发生改变的操作称为数据操纵。SQL 语言中数据操纵包括插入数据、更新数据和删除数据 3 条语句。

(1) 插入数据。插入数据的语法格式如下：

```
INSERT INTO <表名> [( <字段 1>[, <字段 2>, ...])
```

```
VALUES (<常量 1>[, <常量 2>, ...)
```

功能：INSERT 语句用于将新记录插入指定表中。

(2) 更新数据。更新数据的语法格式如下：

```
UPDATE <表名>
```

```
SET <字段 1>=<表达式 1>[, <字段 2>=<表达式 2>], ...
```

```
[WHERE <条件>];
```

功能：UPDATE 语句用于修改指定表中满足 WHERE 子句条件的记录。特别注意，如果省略 WHERE 子句，将更新数据表内的全部记录。

(3) 删除数据。删除数据的语法格式如下：

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

功能：DELETE 语句用于从指定表中删除满足 WHERE 子句条件的记录。特别注意，如果省略 WHERE 子句，将删除数据表内的全部记录。

5. SQL 的视图

视图是 DBMS 所提供的一种由用户观察数据库中数据的重要机制。视图可由基本表或其他视图导出。它与基本表不同，视图只是一个虚表，在数据字典中保留其逻辑定义，而不作为一个表实际存储数据。

定义视图可用 CREATE VIEW 语句实现，其一般形式为：

```
CREATE VIEW <视图名> [( <属性名 1>[, <属性名 2>, ...])
```

```
AS <SELECT 语句> [WITH CHECK OPTION]
```

CREATE VIEW 语句的功能仅是定义一个视图，该语句执行的结果是将有关视图的定义存储在数据字典中，只有对视图进行操作时才根据定义从基本表中形成实际数据供用户使用。

视图定义后，对于视图的查询（SELECT）操作与基本表的查询相同，即对基本表的各种查询形式对视图同样有效。因为视图是一个“虚表”，它并不存储数据，因而对于视图的修改有很多限制。

3.5.5 数据库的设计与管理

数据库已广泛应用于各类应用系统中，如办公自动化系统、管理信息系统、决策支持系统等。实际上，数据库已成为现代信息技术的三大核心技术之一，数据库设计得好坏直接影响整个应用系统的效率和质量。

1. 数据库设计概述

数据库设计是在一个给定的应用环境里构造最优的数据库模式，建立数据库及其应用系统，使之能够有效地存储数据，满足各种用户的应用需求（信息要求和处理要求）。在数据库领域内，常常将使用数据库的各类系统称为数据库应用系统。

数据库是信息系统的核心和基础。数据库可以把信息系统中大量的数据按照一定的模型组织起来，提供存储、维护、检索数据的功能，这样信息系统的使用者就可以方便、及时、准确地从数据库中获得所需的信息。数据库设计是信息系统开发和建设的重要组成部分。

数据库设计有两种方法：一种方法是以信息需求为主，兼顾处理需求，称为面向数据的方法；另一种方法是以处理需求为主，兼顾信息需求，称为面向过程的方法。

数据库设计目前主要采用生命周期法，即将整个数据库应用系统的开发分解成目标独立的需求分析、概念设计、逻辑设计、物理设计、编码、测试、运行、进一步修改等阶段，如图 3-21 所示。

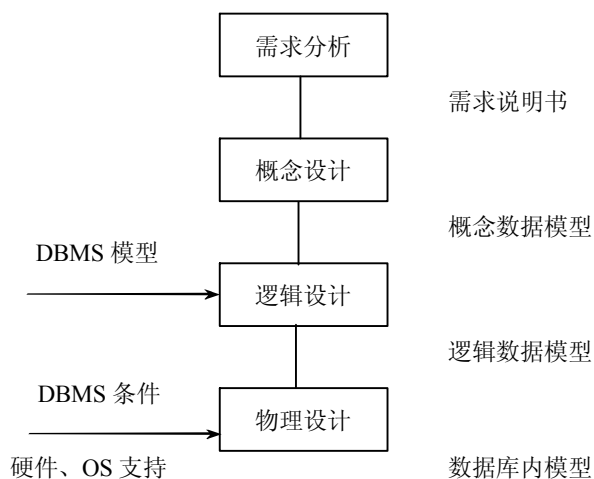


图 3-21 数据库设计的 4 个主要阶段

数据库的设计人员应该具备数据库的基本知识、数据库设计技术、计算机科学的基础知识、程序设计的方法和技巧、软件工程的原理和方法、相关应用领域的知识。

数据库设计过程主要有以下 6 个阶段：

(1) 需求分析阶段：综合各个用户的应用需求。

(2) 概念结构设计阶段：形成独立于机器特点，独立于各个 DBMS 产品的概念模式（E-R 图）。

(3) 逻辑结构设计阶段：首先将 E-R 图转换成具体数据库产品支持的数据模型，如关系模型，形成数据库逻辑模式；然后根据用户处理的要求、安全性的考虑，在基本表的基础上再建立必要的视图，形成数据的外模式。

(4) 数据库物理设计阶段：根据 DBMS 特点和处理的需要进行物理存储安排，建立索引，形成数据库内模式。

(5) 数据库实施阶段：运用 DBMS 提供的数据库语言、工具及宿主语言，根据逻辑设计和物理设计的结果，建立数据库、编制与调试应用程序、组织数据入库并进行调试运行。

(6) 数据库运行和维护阶段：数据库应用系统经过调试运行后即可投入正式运行。

设计一个完善的数据库应用系统往往是上述6个阶段的不断反复。

2. 数据库设计的需求分析

(1) 需求分析任务。需求分析阶段的任务是通过详细调查现实世界要处理的对象(组织、部门、企业等),充分了解原系统(手工系统或计算机系统)的工作概况,明确用户的各种需求,在此基础上确定新系统的功能。新系统必须充分考虑今后可能的扩充和改变,不能仅仅按当前应用需求来设计数据库。

需求分析的重点是调查、收集与分析用户在数据管理中的信息要求、处理要求、安全性与完整性要求。信息是指用户需要从数据库中获得信息的内容与性质,由用户的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。处理要求是指用户对处理功能的要求、对处理响应时间的要求、对处理方式的要求。安全性与完整性要求是为了很好地完成调查的任务,设计人员必须不断地与用户交流,与用户达成共识,以便逐步确定用户的实际安全性与完整性要求,然后通过分析来准确表达这些需求。

新系统的功能必须能够满足用户的信息要求、处理要求、安全性与完整性要求。

(2) 需求分析步骤。进行需求分析首先是调查清楚用户的实际需求,与用户达成共识。调查用户需求的步骤如下:

①调查组织机构情况。包括了解组织部门的组成情况、职责等。

②调查各部门的业务活动情况。包括各个部门输入和使用什么数据、如何加工处理这些数据、输出什么信息、输出到什么部门、输出结果的格式是什么。

③在熟悉业务活动的基础上,协助用户明确对新系统的各种要求。包括信息要求、处理要求、完全性与完整性要求。

④对前面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) 数据字典。数据字典是各类数据描述的集合,是进行详细的数据收集和数据分析所获得的主要结果。数据字典在数据库设计中占有很重要的地位。数据字典通常包括数据项、数据结构、数据流、数据存储和处理过程5个部分。数据项是数据的最小组成单位,若干个数据项可以组成一个数据结构。数据字典通过数据项和数据结构的定义来描述数据流、数据存储的逻辑内容。

数据项是不可再分的数据单位;数据结构反映了数据之间的组合关系,一个数据结构可以由若干个数据项组成,也可以由若干个数据结构组成,或由若干个数据项和数据结构混合组成;数据流是数据结构在系统内传输的路径,数据流来源是说明该数据流来自哪个过程,数据流去向是说明该数据流将到哪个过程去,平均流量是指在单位时间(每天、每周、每月等)里的传输次数,高峰期流量则是指在高峰时期的数据流量;数据存储是数据结构停留或保存的地方,也是数据流的来源和去向之一;处理过程的具体处理逻辑一般用判定表或判定树来描述,数据字典中指需要描述处理过程的说明性信息。

3. 数据库概念设计

数据库概念设计的目的是分析数据间内在语义的关联,在此基础上建立一个数据的抽象模型。可以采用集中式模式设计法和视图集成设计法进行数据库的概念设计。

集中式模式设计法是根据需求由一个统一机构或人员设计一个综合的全局模式。它强调统一与一致,适用于小型或并不复杂的单位或部门,而对大型的或语义复杂的单位则并不适合。视图集成设计法是将一个单位分解成若干个部分,先对每个部分作局部模式设计,建立各个部分的视图,然后以各视图为基础进行集成。视图集成设计法是一种由分散到集中的方法,它的设计过程复杂但它能较好地反映需求,适合于大型与复杂的系统。

使用 E-R 模型与视图集成法进行设计时,需要先选择局部应用,再进行局部视图设计,最后对局部视图进行集成得到概念模式。

4. 数据库的逻辑设计

概念结构是各种数据模型的共同基础,为了能够用某一 DBMS 实现用户需求,还必须将概念结构进一步转化为相应的数据模型,这正是数据库逻辑结构设计所要完成的任务。

数据库的逻辑结构设计一般包括以下设计步骤:

(1) 从 E-R 图向关系模式转换。E-R 图由实体、实体的属性和实体之间的联系 3 个要素组成,关系模型的逻辑结构是一组关系模式的集合,将 E-R 图转化为关系模型,将实体、实体的属性和实体之间的联系转化为关系模式。

(2) 逻辑模式规范化、调整与实现。在逻辑设计中还需要对关系做规范化验证,需要对逻辑模式进行调整以满足 RDBMS 的性能、存储空间等要求,同时对模式做适应 RDBMS 限制条件的修改,设计关系视图以便提供数据的逻辑独立性,使得应用程序不受逻辑模式变化的影响,能够适应用户对数据的不同需求,具有一定的数据保密功能。

5. 数据库的物理设计

数据库在物理设备上的存储结构与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。为一个给定的逻辑数据模型选取一个最适合应用环境的物理结构的过程就是数据库的物理设计。

在设计物理数据库的结构之前,必须充分了解应用环境,详细分析要运行的事务,以获得选择物理数据库设计所需的参数,必须充分了解所用 RDBMS 的内部特征,特别是系统提供的存取方法和存储结构。

数据库物理设计的内容主要包括关系模式选择存取方法,设计关系、索引等数据库文件的物理存储结构。

数据库物理设计的步骤包括确定数据库的物理结构,确定数据的存放位置和存储结构,包括关系、索引、日志、备份等的存储安排和存储结构,确定存储配置等;对物理结构进行评价,评价的重点是时间和空间效率。对数据库物理设计过程中产生的多种方案进行细致的评价,从中选择一个较优的方案作为数据库的物理结构。

如果评价结果满足原设计要求则可进入到物理实施阶段,否则就需要重新设计或修改物理结构,有时甚至要返回逻辑设计阶段修改数据模型。

6. 数据库管理

数据库是一种共享资源,它需要维护与管理,这种工作称为数据库管理,实施管理的人称为数据库管理员。数据库管理一般包括数据库的建立、数据库的调整、数据库的重组、数据库的安全控制与完整性控制、数据库的故障恢复和数据库的监控。

数据库的建立包括数据库模式的建立和数据加载。

数据库的调整一般由 DBA 完成,它主要包括调整关系模式与视图使之更能适应用户的需

求, 调整索引与集簇使数据库性能与效率更佳, 调整分区、数据库缓冲区大小以及并发度使数据库物理性能更好。

数据库运行一定时间后, 其性能会逐渐下降, 下降主要是由于不断地修改、删除与插入所造成的。由于不断地删除而造成盘区内废块的增多而影响 I/O 速度, 由于不断地删除与插入而造成集簇的性能下降, 同时也造成了存储空间分配的零散化, 使得一个完整表的空间分散, 从而造成存取效率下降。对数据库进行重新整理, 重新调整存储空间, 这种工作叫做数据库重组。

数据库的最大特点是数据可以共享, 但数据共享必然带来数据库的安全性问题。数据库系统中的数据共享不能是无条件的共享。数据库中数据的共享是在 DBMS 统一、严格控制之下的共享, 即只允许有合法使用权限的用户访问允许用户存取的数据。数据库系统的安全保护措施是否有效是数据库系统主要的性能指标之一。

数据库的完整性是指数据库的正确性和相容性。

计算机系统中硬件的故障、软件的错误、操作员的失误以及恶意的破坏是不可避免的, 这些故障轻则造成运行事务非正常中断, 影响数据库中数据的正确性, 重则破坏数据库, 使数据库中的数据全部或部分丢失。数据库管理系统具有把数据库从错误状态恢复到某一已知的正确状态的功能, 这就是数据库的恢复。

DBA 需要随时观察数据库的动态变化, 并在发生错误、故障或产生不适应情况时随时采取措施, 如数据库死锁、对数据库的误操作等, 同时还需要监视数据库的性能变化, 在必要时对数据库作调整。

3.5.6 常用数据库管理系统简介

目前市场上比较流行的数据库管理系统产品主要是 Oracle、IBM、Microsoft 和 Sybase 等公司的产品, 下面对常用的几种系统进行简要介绍。

1. Oracle

Oracle 数据库被认为是业界目前比较成功的关系型数据库管理系统。Oracle 公司是世界第二大软件供应商, 是数据库软件领域的第一大厂商 (大型机市场除外)。Oracle 的数据库产品被认为是运行稳定、功能齐全、性能超群的贵族产品。这一方面反映了它在技术方面的领先, 另一方面也反映了它在价格定位上更着重于大型的企业数据库领域。对于数据量大、事务处理繁忙、安全性要求高的企业, Oracle 无疑是比较理想的选择 (当然用户必须在费用方面做出充足的考虑, 因为 Oracle 数据库在同类产品中是比较贵的)。随着 Internet 的普及, 带动了网络经济的发展, Oracle 适时地将自己的产品紧密地和网络计算结合起来, 成为在 Internet 应用领域数据库厂商的佼佼者。

Oracle 数据库可以运行在 UNIX、Windows 等主流操作系统平台上, 完全支持所有的工业标准, 并获得最高级别的 ISO 标准安全性认证。Oracle 采用完全开放策略, 可以使客户选择最适合的解决方案, 同时对开发商提供全力支持。

2. DB2

DB2 是 IBM 公司的产品, 是一个多媒体、Web 关系型数据库管理系统, 其功能足以满足大中型公司的需要, 并且可灵活地服务于中小型电子商务解决方案。DB2 系统在企业级的应用中十分广泛, 目前全球 DB2 系统用户超过 6000 万, 分布于约 40 万家公司。

1968年IBM公司推出的IMS (Information Management System) 是层次数据库系统的典型代表, 是第一个大型的商用数据库管理系统。1970年, IBM公司的研究员首次提出了数据库系统的关系模型, 开创了数据库关系方法和关系数据理论的研究, 为数据库技术奠定了基础。目前IBM仍然是最大的数据库产品提供商(在大型机领域处于垄断地位), 财富100强企业中的100%和财富500强企业中的80%都使用了IBM的DB2数据库产品。DB2的另一个非常重要的优势在于基于DB2的成熟应用非常丰富, 有众多的应用软件开发商围绕在IBM的周围。2001年, IBM公司兼并了世界排名第四的著名数据库公司Informix, 并将其所拥有的先进性融入到DB2当中, 使DB2系统的性能和功能有了进一步提高。

DB2数据库系统采用多进程多线程体系结构, 可以运行于多种操作系统之上, 并分别根据相应平台环境作了调整和优化, 以便能够达到较好的性能。DB2目前支持从PC到UNIX, 从中小型机到大型机, 从IBM到非IBM (HP及SUN UNIX系统等) 的各种操作平台, 可以在主机上以主/从方式独立运行, 也可以在客户机/服务器环境中运行。其中服务平台可以是OS/400、AIX、OS/2、HP-UNIX、SUN-Solaris等操作系统, 客户机平台可以是OS/2、Windows、DOS、AIX、HP-UX、SUN Solaris等操作系统。

3. SQL Server

SQL Server是微软公司开发的大型关系型数据库系统。SQL Server的功能比较全面, 效率高, 可以作为大中型企业或单位的数据库平台。SQL Server在可伸缩性与可靠性方面做了许多工作, 近年来在许多企业的高端服务器上得到了广泛应用。同时, 该产品继承了微软产品界面友好、易学易用的特点, 与其他大型数据库产品相比, 在操作性和交互性方面独树一帜。SQL Server可以与Windows操作系统紧密集成, 这种安排使SQL Server能充分利用操作系统所提供的特性, 不论是应用程序开发速度还是系统事务处理运行速度, 都能得到较大的提升。另外, SQL Server可以借助浏览器实现数据库查询功能, 并支持内容丰富的扩展标记语言(XML), 提供了全面支持Web功能的数据库解决方案。对于在Windows平台上开发的各种企业级信息管理系统来说, 不论是C/S(客户机/服务器)架构还是B/S(浏览器/服务器)架构, SQL Server都是一个很好的选择。SQL Server的缺点是只能在Windows系统下运行。

4. Sybase 系列

Sybase公司成立于1984年11月, 产品研究和开发包括企业级数据库、数据复制和数据访问。主要产品有: Sybase的旗舰数据库产品Adaptive Server Enterprise、Adaptive Server Replication、Adaptive Server Connect及异构数据库互连选项。Sybase ASE是其主要的数据库产品, 可以运行在UNIX和Windows平台上。Sybase Warehouse Studio在客户分析、市场划分和财务规划方面提供了专门的分析解决方案。Warehouse Studio的核心产品有Adaptive Server IQ, 其专利化的从底层设计的数据存储技术能快速查询大量数据。围绕Adaptive Server IQ有一套完整的工具集, 包括数据仓库或数据集市的设计、各种数据源的集成转换、信息的可视化分析, 以及关键客户数据(元数据)的管理。

5. Access

Access是微软Office办公组件中的一个重要成员。自从1992年开始销售以来, Access已经卖出了超过6000万份, 现在它已经成为世界上最流行的桌面数据库管理系统。

Access简单易学, 一个普通的计算机用户即可掌握并使用它。同时, Access的功能也足以应付一般的小型数据管理及处理需要。无论用户是要创建一个个人使用的独立的桌面数据

10. 理论上已经证明, 求解问题的计算机程序的3种控制结构是()。
- A. 转子(程序)、返回、处理 B. 输入、输出、处理
C. 顺序、选择、重复 D. I/O、转移、循环
11. 下列操作系统产品中, () 是一种“共享软件”, 其源代码向世人公开。
- A. DOS B. Windows
C. UNIX D. Linux
12. () 不是程序设计语言。
- A. FORTRAN B. C++
C. Java D. Flash
13. 假定学生关系模式是 S (学号, 姓名, 性别, 年龄), 课程关系模式是 C (课程号, 课程名, 学时数), 选课关系模式是 SC (学号, 课程号, 成绩), 要查找选修课程名为“信息技术”的所有女学生的姓名, 将涉及关系()。
- A. S B. C、SC
C. S、SC D. S、C、SC

三、填空题

1. 算法是对问题求解过程的一种描述, “算法中描述的每个操作都是可以由计算机执行的, 且能够在有限时间内完成”, 这句话所描述的性质被称为算法的_____性。
2. 解决某一个问题的算法也许有多种, 但它们都必须满足确定性、有穷性、能行性等性质, 其中输出的个数 n 应大于等于_____。(填一个数字)
3. Java 是一种面向_____的程序设计语言, 特别适用于网络环境的软件开发。
4. 使用计算机求解问题的主要步骤是: 先要理解和确定问题, 然后寻找其解决方法并将其表示成_____, 接着再进行编程、调试和运行。
5. 程序设计语言中的控制成分包括顺序结构、条件选择结构和_____。
6. SQL 语言中用_____语句创建视图。
7. 算法和_____的设计是程序设计的主要内容。