

# 数据库系统概论

自 20 世纪 50 年代中期开始到 60 年代数据库技术作为一个独立的分支应运而生，计算机的应用也由科学研究逐渐扩展到企业、事业、行政等社会各个领域，数据处理也成为计算机的主要应用；并且随着其不断的发展，数据库的理论和应用取得了巨大的成功，极大地促进了计算机应用向各行各业的渗透。如今，数据库系统已经成为现实生活中不可缺少的一部分。我们每天都会或多或少地、不知不觉地和数据库发生某些联系。我们可能会去邮局订阅一本杂志，可能会去预定旅馆房间或机票，可能会为找一本书而去检索一个由计算机管理的图书馆目录，也可能会去银行取款。所有这些活动都涉及到对数据库的使用。

本章主要介绍数据库中有关的基本概念、数据库系统的组成及主要功能、数据库的发展状况以及数据模型等基本知识，为以后各章的学习打下基础。

## 1.1 信息、数据与数据处理

### 1.1.1 信息与数据

信息和数据是数据处理中的两个基本概念，它们有着不同的含义。

#### 1. 信息(Information)

##### (1)信息的定义

在信息社会，信息是一种重要的资源，它与能源、材料构成了现代社会的三大支柱。信息是目前许多学科广泛使用的概念。在不同的领域中，其含义有所不同。一般认为，信息是关于现实世界事物的存在方式或运动状态反映的综合。例如，我们在多媒体教室上课，用的投影仪屏幕颜色是白色的，形状是正方形的，长、宽尺寸均是 2.6m，材料是塑料的。这些都是关于投影仪屏幕的信息，是投影仪屏幕存在状态的反映。又如，正在进行一场篮球比赛的学生，该时刻学生的运动状态就构成了学生信息的一部分。

##### (2)信息的特征

信息有以下三个重要特征：

信息源于物质和能量。信息的传递需要物质载体，信息的获取和传递需要消耗能量，信息不可能脱离物质而存在。如信息可以通过广播、报纸、电视、网络等进行传递。

信息是可以感知的。人类对客观事物的感知，可以通过感觉器官，也可以通过各种仪

器仪表来获得，不同的信息源有不同的感知方法。如我们从广播上获得的信息是通过听觉器官来感知的；而从报纸上获得信息则通过视觉器官来感知。

信息是可存储、加工、传递和再生的。人类可以用大脑存储少量的信息，用文字长期存储信息。而计算机的发展进一步扩大了信息存储的范围。人类还可对收集到的信息进行取舍整理以及通过各种手段和方法进行传递与再生。

## 2. 数据 (Data)

### (1) 数据的定义

数据是用来记录信息的可识别的符号。

### (2) 数据的表现形式

数据的表现形式多种多样，不仅包括数字和文字，还包括图形、图像和声音等。这些数据可以记录在纸上，也可记录在各种存储器中，如磁盘、磁带、光盘等。如“2008年北京召开奥运会”其中的数据“2008”可以改为汉字的“二零零八”，这样同一信息就有不同的数据表现形式。

## 3. 数据与信息的联系

数据与信息相互联系，数据是信息的物理符号表示或载体，信息是数据的内涵，是对数据的语义解释。另一方面，数据和信息又有区别，某一具体的信息和表示它的数据的这种对应关系又因环境而变化，数据能表示信息，但并非任何数据都能表示信息，有的数据可能完全没有用处，称为数据垃圾。同一信息可能有不同的符号表示，同一数据也可能有不同的解释。因此，信息是人们消化理解了的数据，信息是抽象的，不随数据设备所决定的数据形式而改变，而数据的表示方式及存在方式却是客观具体的。

## 1.1.2 数据处理

数据处理又称信息处理，是将数据转换成信息的过程，包括对数据的收集、存储、加工、检索和传输等一系列活动，其目的是从大量的原始数据中抽取和推导出有价值的信息，作各种应用。

我们可简单地用下列式子表示信息、数据与数据处理的关系：

信息 = 数据 + 数据处理

数据可以形象化地比喻为原料，是输入；信息就像产品，是输出结果；而数据处理是原料变成产品的过程。从这种角度看，“数据处理”的真正含义应该是为了产生信息而处理数据。

# 1.2 数据库技术的产生、发展与研究领域

## 1.2.1 数据库技术的产生与发展

使用计算机后，随着数据处理量的增长，产生了数据管理技术。数据管理技术的发展与计算机硬件（主要是外部存储器）、系统软件及计算机应用的范围有着密切的联系。数据管理技术的发展经历了以下四个阶段：人工管理阶段、文件系统阶段、数据库阶段和高级数据库技术阶段。

### 1. 人工管理阶段（20世纪50年代中期以前）

在这一阶段，计算机主要应用于科学计算，对于数据保存的需求尚不迫切，数据的管理是靠人工进行的，计算机不保存数据，也没有专用的软件对数据进行管理，只有程序（Program）的概念，没有文件（File）的概念，一组数据对应一个应用程序，如图 1.1 所示。数据存在大量重复存储现象。

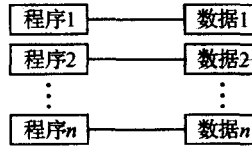


图 1.1 人工管理阶段

### 2. 文件系统阶段（20世纪50年代后期至60年代中期）

在这一阶段，计算机不仅用于科学计算，还用于信息管理。硬件方面出现了可以直接存取的外部存储设备，软件方面有了操作系统中专门管理数据的文件系统。数据的管理是以独立的数据文件形式存放，并可按记录存取。在文件系统阶段，一个应用程序可以处理多个数据文件，文件系统在程序与数据之间起到接口的作用，使程序和数据有了一定的独立性，如图 1.2 所示。这使得程序员可以集中精力于算法，不必过多地考虑物理细节，因此在这一时期各种数据结构和算法得到了充分的发展，大大丰富了计算机科学，今天的数据库也正是在文件系统的基础上发展起来的。但是文件系统的致命缺陷是各数据文件之间缺乏有机的联系，数据与程序之间缺乏独立性，不能有效地共享相同的数据，从而造成数据冗余和不一致，给数据的修改和维护带来了困难。

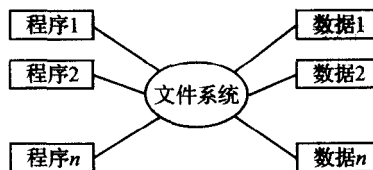


图 1.2 文件系统阶段

### 3. 数据库阶段

随着计算机技术的迅速发展和广泛应用，磁盘技术已取得重要进展，数据管理中数据量也急剧增长，对数据共享和数据管理就提出了更高的要求，此时文件系统已经不能满足应用的需求，数据库技术也就应运而生。

20世纪60年代末的三件大事标志着数据管理技术进入了数据库阶段：

①1968年，美国IBM公司推出的商品化软件IMS数据库管理系统（Information Management System），这是数据库技术的第一次飞跃，为数据库技术的产生揭开序幕，该系统是一种树型结构的层次模型，曾在20世纪70年代商业上广泛应用。

②1969年，美国CODASYL组织发表了网状数据库系统的标准文本：DBTG文本，它是数据库网状模型的基础。

③1970年，IBM公司San Jose现在称Almaden研究室的高级研究员E. F. Codd发表了一篇题为“A Relational Model of Data for Shared Data Base”的重要论文，引进了关系代

数，并首次提出了数据库的关系模型概念，为数据库技术的发展奠定了基础，从而引发数据库技术的第二次飞跃。

这一阶段数据库与应用程序的关系可由图 1.3 表示。

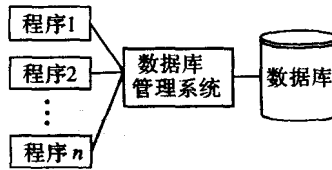


图 1.3 数据库系统阶段

20 世纪 70 年代是数据库技术蓬勃发展的年代。网状系统和层次系统占据了整个商用市场，而关系系统仅处于实验阶段。许多计算机公司研制了各种数据库管理系统，许多教授、专家发表了大量的理论文章，使数据库技术在实践和理论上都得到了飞速发展，日趋成熟。1974 年 IBM 公司在 IBM370 大型机上研制成功第一个关系数据库系统 System R 该公司的 Boyce 和 Chamberlin 提出的 SQL (Structured Query Language 结构化查询语言 在 System R 上得到实现。20 世纪 80 年代，关系数据库技术得到了进一步发展，由于使用方便等原因，关系数据库逐步取代网状系统和层次系统，占领了市场。同时关系数据库的理论也逐步完善。目前流行的关系数据库管理系统有 IBM 公司的 DB2, Oracle 公司的 Oracle, Sybase 公司的 Sybase, Microsoft 公司的 SQL Server, TcX 公司开发的 MySQL 以及面向桌面应用的关系数据库管理系统 Visual FoxPro, Access 等。

现在，数据库技术的应用已经深入到人类社会的各个领域，从工农业生产、商业、行政、科学研究、工程技术到国防军事的各个部门。管理信息系统、办公自动化系统、决策支持系统等都是使用了数据库管理系统或数据库技术的计算机应用系统。

#### 4. 高级数据库技术阶段

这一阶段的主要标志是 20 世纪 80 年代的分布式数据库系统、90 年代的面向对象数据库系统和各种新型数据库系统。

##### (1) 分布式数据库系统

在这一阶段以前的数据库系统是集中式的。在文件系统阶段，数据分散在各个文件中，文件之间缺乏联系。集中式数据库把数据集中在一个数据库中进行集中管理，减少了数据冗余和不一致性，而且数据联系比文件系统强得多。但集中式系统也有弱点；一是随着数据量的增加，系统越来越庞大，操作越来越复杂，开销越来越大；二是数据集中存储，大量的通信都要通过主机，容易造成拥挤。随着小型计算机和微型计算机的普及，以及计算机网络软件和远程通信的发展，分布式数据库系统崛起了。

分布式数据库系统，通俗地说是物理上分散而逻辑上集中的数据库系统。分布式数据库系统使用计算机网络将地理位置分散而又需要不同程度的集中管理和控制的多个逻辑单位（通常是集中式数据库系统）联结起来，共同组成一个统一的数据库系统。其中，被计算机网络联结的每个逻辑单位称为节点（Node）或站点（Site）。所谓逻辑上集中是指各站点之间不是互不关联的，而是一个逻辑整体，并由统一的数据库管理系统进行管理，这个数据库管理系统称为分布式数据管理系统（Distributed Database Management System 简称 DDBMS）。

图 1.4 为分布式数据库系统示意图。

本书 9.2 节将详细介绍分布式数据库系统。

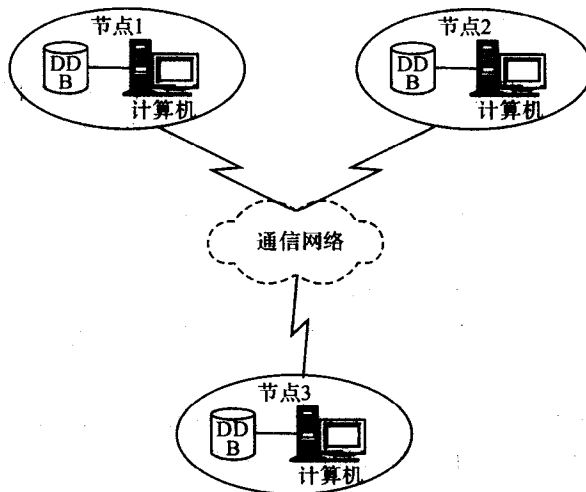


图 1.4 分布式数据库系统

### (2) 面向对象数据库系统

在数据处理领域，关系数据库的使用已相当普遍、相当出色。但是现实世界存在着许多具有更复杂数据结构的实际应用领域，如多媒体、多维表格数据和 CAD（计算机辅助设计）数据等应用问题，需要更高级的数据库技术来表达，以便于管理、构造与维护大容量的持久数据，并使它们能与大型复杂程序紧密结合。而面向对象数据库正是适应这种形势发展起来的，它是面向对象的程序设计技术与数据库技术结合的产物。

面向对象数据库系统主要有以下两个特点：

- 面向对象数据模型能完整地描述现实世界的数据结构，能表达数据间的嵌套、递归的联系。
- 具有面向对象技术的封装性（把数据和操作定义在一起）和继承性（继承数据结构和操作）的特点，提高了软件的可重用性。

本书 9.1 节将介绍面向对象数据库系统的发展状况。

### (3) 各种新型的数据库系统

数据库技术是计算机软件领域的一个重要分支，经过 30 余年的发展，已经形成相当规模的理论体系和实用技术。但受到相关学科和应用领域（如网络、多媒体等）的影响，数据库技术的研究并没有停滞，仍在不断发展，并出现许多新的分支。例如：演绎数据库，主动数据库，时态数据库，模糊数据库，并行数据库，巨型数据库，多媒体数据库，嵌入式数据库，WWW 数据库，数据挖掘，内存数据库，联邦数据库， workflow 数据库，工程数据库，地理数据库等。本书第 9 章对其中部分内容有介绍。感兴趣的读者可以查阅有关的书籍。

## 1.2.2 数据库技术的研究领域

数据库学科的研究范围十分广泛，主要有以下三个领域：

### 1. 数据库管理系统软件的研制

数据库管理系统 (Database Manage System, 简称 DBMS) 是数据库系统的基础。DBMS

的研制包括研制 DBMS 本身以及 DBMS 为核心的一组相互联系的软件系统，包括工具软件和中间件。研制的目标是提高系统性能和用户的生产率。

### 2. 数据库设计

数据库设计的研究范围包括数据库的设计方法、设计工具和设计理论的研究、数据模型和数据建模的研究、计算机辅助数据库设计及其软件系统的研究、数据库设计规范和标准的研究等。数据库设计的目的是在 DBMS 的支持下，按照应用的要求，为某一部门或组织设计一个结构合理、使用方便、效率较高的数据库及其应用系统。

### 3. 数据库理论

数据库理论的研究主要集中于关系规范化理论、关系数据理论等。近年来，随着人工智能与数据库理论的结合以及并行计算技术的发展，数据库逻辑演绎、知识推理和并行算法等都成为新的研究方向。

随着数据库应用领域的不断扩展，计算机技术的迅猛发展，数据库技术与人工智能技术、网络通信技术、并行计算技术等相互渗透、相互结合，不断涌现出新的研究方向。

## 1.3 数据库系统的组成和结构

### 1.3.1 数据库系统的组成

数据库系统(Database System 简称为 DBS)是采用了数据库技术的计算机系统，通常由数据库、硬件、软件、用户四部分组成。数据库系统如图 1.5 所示。

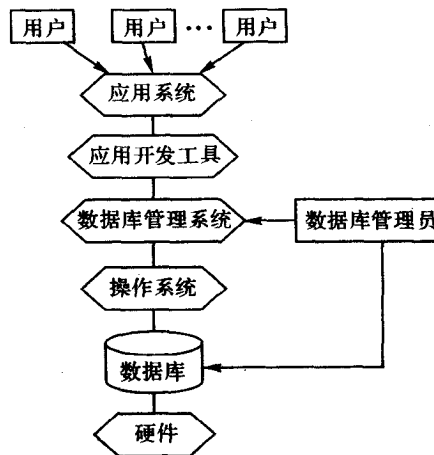


图 1.5 数据库系统

#### 1. 数据库

数据库(Database 简称 DB)是指相互关联的数据集合。一般定义为：长期存储在计算机内的、有组织的、可共享的数据的集合。数据库有以下几个特点：

##### (1) 数据结构化

在数据库系统中，数据不再像文件系统中的数据那样从属于特定的应用，而是面向全组

织的复杂的数据结构，数据的结构化是数据库系统区别于文件系统的根本特征。

### (2) 数据共享

数据库系统中的数据可供多个用户、多种语言和多个应用程序共享，这是数据库技术的基本特征，数据共享大大减少了数据冗余度和不一致性，大大提高了数据的利用率和工作效率。

### (3) 数据独立性

数据独立性包括数据的物理独立性和逻辑独立性。用户的应用程序与存储在磁盘上的数据库的数据是相互独立的，这就是数据的物理独立性；同时用户的应用程序与数据库的逻辑结构是相互独立的，这就是数据的逻辑独立性；它不会因一方的改变而改变，这大大减少了应用程序设计和数据库维护的工作量。

## 2. 硬件

计算机系统的硬件包括中央处理器、内存、外存、输入/输出设备等。在数据库系统中特别要关注内存、外存、I/O存取速度、可支持的结点数和性能稳定性等指标，现在还要考虑支持联网的能力和配备必要的后备存储器等因素。此外，还要求系统有较高的通道能力，以提高数据的传输速度。

## 3. 软件

数据库系统的软件主要包括操作系统（OS）、数据库管理系统（DBMS）、各种宿主语言和应用开发支撑软件。DBMS是在操作系统的文件系统基础上发展起来的，在操作系统支持下工作，是数据库系统的核心软件。为了开发应用系统，需要各种宿主语言，这些语言大部分属于第三代语言（3GL）范畴，例如 COBOL、C、PL/I 等；有些是属于面向对象程序设计语言，例如 C++、Java 等语言。应用开发支撑软件是为应用开发人员提供高效率、多功能的交互式程序设计系统，一般属于第四代语言（4GL）范畴，包括报表生成器、表格系统、图形系统、具有数据库访问和表格 I/O 功能的软件、数据字典系统等。它们为数据库应用系统的开发和应用提供了良好的环境，可提高生产率 20~100 倍。当前比较流行的应用开发工具有 PowerBuilder、Delphi、Visual Basic 等。

## 4. 用户

管理、开发和使用数据库系统的用户主要有数据库管理员、应用程序员和普通用户。数据库系统中不同人员涉及到不同的数据抽象级别，具有不同的数据视图。

### (1) 普通用户

普通用户有应用程序和终端用户两类。它们通过应用系统的用户接口使用数据库，目前常用的接口方式有菜单驱动、表格操作、图形显示、报表生成等，这些接口使得用户的操作简单易学易用，适合非计算机专业人员使用。

### (2) 应用程序员

应用程序员负责设计和调试数据库系统的应用程序。他们通常使用 4GL 开发工具编写数据库应用程序，供普通用户使用。

### (3) 数据库管理员 (Database Administrator, 简称 DBA)

DBA 在数据库管理中是极其重要的，即所谓的超级用户。DBA 全面负责管理、控制和维护数据库，使数据能被任何有使用权限的人有效使用，DBA 可以是一个人，或几个人组成的一个小组。DBA 主要有以下职责：

参与数据库设计的全过程，决定整个数据库的结构和信息内容；

帮助终端用户使用数据库系统 如 培训终端用户，解答终端用户日常使用数据库系统时遇到的问题等；

定义数据的安全性和完整性，负责分配用户对数据库的使用权限和口令管理等数据库访问策略；

监督控制数据库的使用和运行，改进和重新构造数据库系统。当数据库受到损坏时，应负责恢复数据库；当数据库的结构需要改变时，完成对数据库结构的修改。

DBA 不仅要有较高的技术水平和较深的资历，还应具有了解和阐明管理要求的能力。特别对于大型数据库系统, DBA 极为重要。常见的微机系统往往只有一个用户，没有必要设置 DBA, DBA 由应用程序员或终端用户代替。

### 1.3.2 数据库系统的结构

数据库系统通常采用三级结构：模式、外模式和内模式，在这三级模式之间提供了两级映像 外模式 / 模式映像 模式 / 内模式映像 如图 1.6 所示。

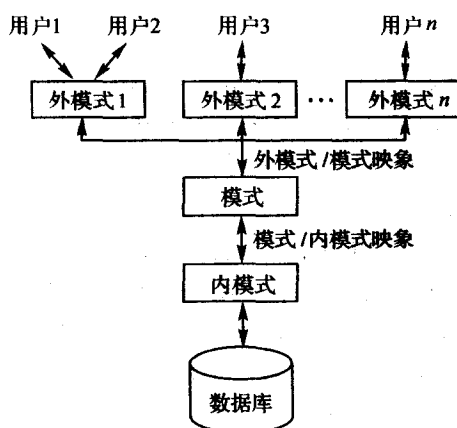


图 1.6 数据库系统的三级模式结构

#### 1. 模式 (Schema)

模式是数据库中全部数据的逻辑结构的描述，是所有概念记录类型的定义，又称概念模式或逻辑模式。模式一般以某一种数据模型为基础，定义数据的逻辑结构，如记录名称、数据项名称、类型、长度等，还要定义数据的安全性和完整性及数据之间的联系。模式是数据库系统三级结构的中间层，它与应用程序和高级语言无关，也与物理结构无关。下面我们将以关系数据库为例。例如，在学生选课数据库中，有如图 1.7 的关系模式集 图 1.8 是这个关系模式的三个具体关系。

数据库系统提供了模式描述语言 (模式 DDL 来定义模式。

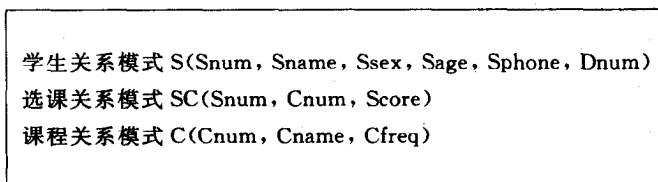


图 1.7 关系模式集

课程表 C

课程号 Cnum	课程名称 Cname	学分 Cfreq
1	数据库系统原理	4
2	C 程序设计	4
3	计算机体系结构	3
4	自动控制原理	2
5	数据结构	4

学生表 S

学号 Snum	姓名 Sname	性别 Ssex	年龄 Sage	电话 Sphone	系编号 Dnum
S001	张山	男	18	45673434	2
S002	李新	男	22	89457321	3
S003	岳岚	女	17		1
S004	王燕	女	19		1
S005	王力	男	24	13098765892	3
S006	赵娟	女	16		1

选修表 SC

学号 Snum	课程号 Cnum	成绩 Score
S001	2	83
S001	4	89
S001	5	65
S002	4	75
S004	1	85
S005	1	92
S005	3	76

图 1.8 学生选课中三个关系

## 2. 外模式 (External Schema)

外模式是指用户所看到和使用的数据库 即局部逻辑结构 又称子模式 或用户视图。一个数据库可以有多个外模式，由于用户的需求、数据的安全等方面的不同，可以有不同的外模式。每个用户都需要通过一个外模式来使用数据库，但不同的用户可以使用同一外模式。外模式是数据库系统保证数据库安全性的一个重要手段。除此之外，还应指出数据与关系模式中相应数据的联系。例如，在学生选课数据库中，用户需要用到外模式成绩 G 如图 1.9 所示。这个外模式的构造过程如图 1.10 所示。

数据库系统提供了外模式描述语言 (Data Definition Language, 简称 DDL 来定义外模式。

成绩外模式 G(Snum, Sname, Cnum, Score)
-----------------------------------

图 1.9 外模式

## 3. 内模式 (Internal Schema)

内模式是对内层数据的物理结构和存储方式的描述，是数据在数据库文件内部的表示

方式，如记录是如何进行存储的（顺序存储方式还是散列方式），如何索引等。内模式是用设备介质语言来定义的，又称存储模式，或物理模式。内模式对一般用户是透明的。

数据库系统提供了内模式描述语言（内模式 DDL）来定义内模式。

#### 4. 两级映像

##### (1) 外模式 / 模式映像

外模式 / 模式映像定义了外模式与模式之间的映像关系。由于外模式和模式的数据结构可能不一致，即记录类型、字段类型的命名和组成可能不一样，因此，需要这个映像说明外部记录和概念记录之间的对应性。

##### (2) 模式 / 内模式映像

模式 / 内模式映像定义了模式与内模式的映像关系。由于模式和内模式的数据结构可能不一致，即记录类型、字段类型的命名和组成可能不一样，因此，需要这个映像说明概念记录和内部记录之间的对应性。

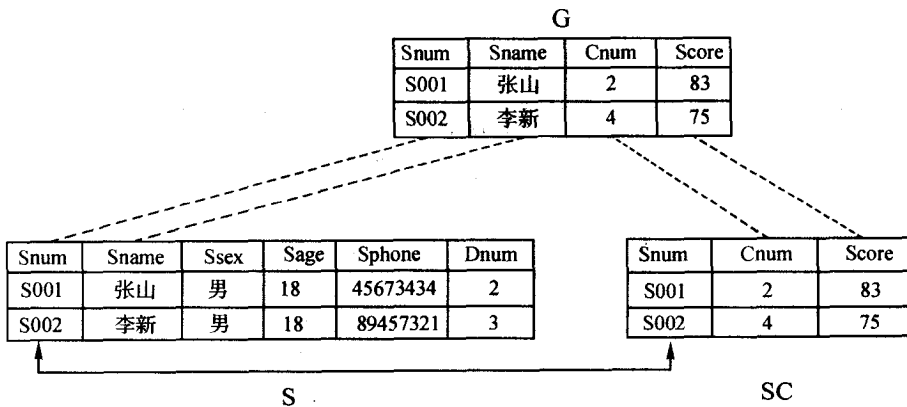


图 1.10 外模式 G 的定义

#### 5. 两级数据独立性

由于数据库系统采用三级模式结构，因此系统具有数据独立性的特点。

数据独立性 (Data Independence) 是指应用程序和数据库的数据结构之间相互独立，不受影响。

数据独立性分成物理数据独立性和逻辑数据独立性两个级别。

##### (1) 物理数据独立性

如果数据库的内模式要修改，即数据库的物理结构有所变化，那么只要对模式 / 内模式映像作相应的修改，可以使概念模式尽可能保持不变。也就是对内模式的修改尽量不影响概念模式，当然对外模式 and 应用程序的影响更小，这样，我们称数据库达到了物理数据独立性（简称物理独立性）。

##### (2) 逻辑数据独立性

如果数据库的概念模式要修改，例如增加记录类型或增加数据项，那么只要对外模式 / 模式映像作相应的修改，可以使外模式尽可能保持不变。这样，我们称数据库达到了逻辑数据独立性（简称逻辑独立性）。

这两个数据独立性是数据库系统的重要特性。

## 1.4 数据库管理系统

数据库管理系统 (DBMS) 是数据库系统中对数据进行管理的一组大型软件系统,它是数据库系统的核心组成部分。数据库系统的一切操作,包括查询、更新及各种控制,都是通过 DBMS 进行的。目前常用的 DBMS 有 Oracle, DB2, Sybase, Microsoft SQL Server, FoxPro 和 Access 等。

### 1.4.1 DBMS 的主要功能

DBMS 的主要功能包括以下几个方面:

#### 1. 数据库定义功能

DBMS 提供数据定义语言 DDL (Data Definition Language) 来定义数据库的三级模式、两级映像,定义数据完整性和保密限制等约束。

#### 2. 数据库操纵功能

DBMS 提供数据操纵语言 DML (Data Manipulation Language) 来实现对数据库的操作,如查询、插入、修改和删除。DML 有两类:一类是嵌入在宿主语言中的,如嵌入在 C, Java, Delphi, PowerBuilder 等高级语言中,这类 DML 称为宿主型 DML;另一类是可以独立地交互使用的 DML,称为自主型或自含型 DML,常用的有 Transact-SQL, SQL Plus 等。目前国内外较流行的 DBMS 都包含有这两种 DML 供用户使用。

#### 3. 数据库保护功能

数据库中的数据是信息社会的战略资源,对数据的保护是至关重要的。DBMS 对数据库的保护主要包括四个方面:数据安全性控制、数据完整性控制、数据并发控制和数据库的恢复。

##### (1) 数据安全性控制

数据安全性控制是对数据库的一种保护措施。它的作用是防止未被授权的用户破坏或存取数据库中的数据。用户首先必须向 DBMS 标识自己,在系统确定有权对指定的数据进行存取时才能存取数据。防止未被授权的用户蓄谋或无意地修改数据是很重要的,否则会导致数据完整性的破坏,从而使企事业单位蒙受巨大的损失。

##### (2) 数据完整性控制

数据完整性控制是 DBMS 对数据库提供保护的另一个重要方面。完整性是数据的准确性和一致性的测度。当数据加入到数据库时,对数据的合法性和一致性的检验将会提高数据的完整性程度。完整性控制的目的是保证进入数据库中的存储数据的语义的正确性和有效性,防止操作对数据造成违反其语义的改变。因此,DBMS 都允许对数据库中各类数据定义若干语义完整性约束,由 DBMS 强制实行。

##### (3) 并发控制

DBMS 一般允许多用户并发地访问数据库,即数据共享,但是多个用户同时对数据库进行访问可能会破坏数据的正确性,或者存储了错误的数据库,或者读取了不正确的数据库即所谓的“脏数据”。因此 DBMS 中必须具有并发控制机制,解决多用户下的并发冲突。

#### (4) 恢复功能

恢复功能是保护数据库的又一个重要方面。数据库在运行中可能会出现各种故障：如停电、软硬件各种错误等，导致数据库的损坏或不一致性。DBMS 必须把处于故障中的数据库恢复到以前的某个正确状态，保证数据库的一致性。

DBMS 的其他保护功能还有系统缓冲区管理以及数据存储的某些自适应调节机制。

#### 4. 数据库维护功能

DBMS 提供一系列的实用程序来完成包括数据库的初始数据的装入、转换功能、数据库的转储、重组、性能监视、分析等维护功能。

#### 5. 数据字典

数据字典 (Data Dictionary, 简称 DD) 是对数据库结构的描述，存放着对实际数据库三级模式的定义，是数据库系统中各种描述信息和控制信息的集合。还存放数据库运行时的统计信息，如记录个数、访问次数等。数据字典是数据库管理的有力工具。数据字典是数据库系统的一部分，但用户通常不能直接访问它，只有 DBMS 才能对它进行访问。

### 1.4.2 DBMS 的组成

DBMS 是许多程序所组成的一个大型软件系统，每个程序都有自己的功能，共同完成 DBMS 的一个或几个工作。一个完整的 DBMS 通常由以下几部分组成。

#### 1. 语言编译处理程序

语言编译程序包括以下两个程序：

##### (1) 数据定义语言 DDL 编译程序

把用 DDL 编写的各级源模式编译成各级目标模式。这些目标模式是对数据库结构信息的描述，它们被保存在数据字典中，供数据操纵控制时使用。

##### (2) 数据操纵语言 DML 编译程序

它将应用程序中的 DML 语句转换成可执行程序，实现对数据库的检索、插入、修改等基本操作。

#### 2. 系统运行控制程序

系统运行控制程序主要包括以下几部分：

##### (1) 系统总控程序

用于控制和协调各程序的活动，它是 DBMS 运行程序的核心。

##### (2) 安全性控制程序

防止未被授权的用户存取数据库中的数据。

##### (3) 完整性控制程序

检查完整性约束条件，确保进入数据库中的数据的正确性、有效性和相容性。

##### (4) 并发控制程序

协调多用户、多任务环境下各应用程序对数据库的并发操作，保证数据的一致性。

##### (5) 数据存取和更新程序

实施对数据库数据的检索、插入、修改和删除等操作。

##### (6) 通信控制程序

实现用户程序与 DBMS 间的通信。

此外，还有事务管理程序、运行日志管理程序等。所有这些程序在数据库系统运行过程中协同操作，监视着对数据库的所有操作，控制、管理数据库资源等。

### 3. 系统建立、维护程序

系统建立、维护程序主要包括以下几部分：

#### (1) 装配程序

完成初始数据库的装入。

#### (2) 重组程序

当数据库系统性能降低时（如查询速度变慢），需要重新组织数据库，重新装入数据。

#### (3) 系统恢复程序

当数据库系统受到破坏时，将数据库系统恢复到以前某个正确的状态。

### 4. 数据字典系统程序

管理数据字典，实现数据字典功能。

## 1.4.3 DBMS 的数据存取过程

在数据库系统中，DBMS 与操作系统、应用程序、硬件等协调工作，共同完成数据各种存取操作 其中 DBMS 起着关键的作用。图 1.11 为 DBMS 对数据的存取过程示意图。从图中可知，DBMS 对数据的存取通常需要以下四步：

①用户对数据库进行操作，使用某种特定的数据操作语言向 DBMS 发出存取请求；

②DBMS 接受请求并解释；

③DBMS 依次检查外模式、外模式 / 模式映像、模式、模式 / 内模式映像及存储结构定义 同时 DBMS 为应用程序在内存开辟一个 DB 的系统缓冲区，用于数据的传输和格式的转变。而三级模式定义存放在数据字典中；

④DBMS 对存储数据执行必要的存取操作。

上述存取过程中还包括安全性控制、完整性控制，以确保数据的正确、有效和一致。

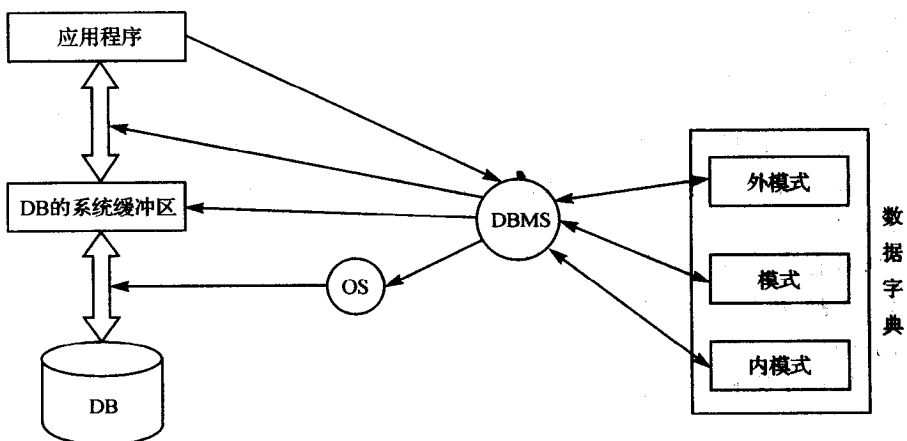


图 1.11 DBMS 对数据存取过程示意图

## 1.5 数据模型

数据模型是理解数据库的基础。模型 (Model) 是对现实世界的抽象。数据模型 (Data Model) 是对现实世界进行抽象的工具, 用于描述现实世界的的数据、数据联系、数据语义和数据约束等方面的内容。

### 1.5.1 数据之间的联系

#### 1. 三个世界的划分

计算机不能直接处理现实世界中的具体事务, 所以人们必须把具体事物抽象并转换成计算机能够处理的数据。数据加工要经历现实世界、信息世界和机器世界三个不同的世界和两级抽象及转换。

##### (1) 现实世界

现实世界 (Real World) 是存在于人们头脑外的客观世界。在现实世界中, 事物的存在不是孤立的, 是相互联系的。而事物及其联系正是数据库最原始的信息。例如, 职工工资管理、学生选修情况、图书馆中对图书和读者的管理、货物的进销存等。

##### (2) 信息世界

信息世界 (Information World) 是现实世界在人们头脑中的反映。现实世界中的事物及其联系经过人们的选择、命名、分类之后进入信息世界, 即把现实世界中的事物及其联系抽象为某一种信息结构, 这种信息结构不依赖于具体的计算机系统, 也不是某一 DBMS 支持的数据模型, 而是概念模型, 是对现实世界的第一层抽象。

##### (3) 机器世界

机器世界 (Machine World) 又称为数据世界, 是数据库的处理对象。信息世界的信息经过加工、编码转换为计算机能接受的数据形式, 即信息的数据化, 这是第二层抽象。

#### 2. 信息世界中的数据描述

在信息世界中, 数据库描述常用到以下一些术语:

##### (1) 实体 (Entity)

客观存在并可相互区别的事物称为实体。实体可以是具体的对象, 如人、事、物, 也可以是抽象的事件或联系, 如学生的一次选课、一次足球比赛、一次借书等。

##### (2) 属性 (Attribute)

实体所具有的某一特性称为属性。实体可以由若干个属性组成。例如, 学生实体有姓名、学号、性别、年龄、电话、系编号等属性。

##### (3) 码 (Key)

能惟一标识实体的属性或属性集称为码。例如, 学生的学号可以作为学生实体的码, 学生的姓名则不一定可以作为学生实体的码, 因为姓名可能有重复。而选课情况则要把学号和课程号的组合作为码。

##### (4) 实体集 (Entity Set)

同一类型的实体的集合称为实体集, 即具有同一类属性的事物的集合。例如所有的学

生、所有的课程、所有的选课情况。

#### (5) 域 Domain

属性的取值范围称为该属性的域或值域。一个属性的值域可以是整数、实数、字符串等。姓名的域为字符串 性别的域为 { 男 女 } 年龄的域为整数。

#### (6) 实体型(Entity Type)

实体型就是用实体名和属性名集合来描述同类实体。例如：学生(学号, 姓名, 性别, 年龄, 电话, 系编号)课程(课程号, 课程名称, 学分)选修情况(学号, 课程号, 成绩)

### 3. 机器世界中的数据描述

机器世界中以下数据描述的术语：

#### (1) 字段 Field

标记实体属性的符号集叫字段，或数据项。它是数据库中可以命名的最小逻辑数据单位，所以又叫数据元素。字段的命名应该体现出属性的具体含义。例如学生表中可用 Snum 来表示学号属性、Sname 来表示姓名属性等。

#### (2) 记录 Record

字段的有序集合称为记录。一般用一个记录描述一个实体，所以记录又可定义为能完整地描述一个实体的符号集。例如一个学生记录由有序的字段集组成：学号、姓名、性别、年龄、电话、系编号。

#### (3) 文件 File

同一类记录的汇集称为文件。文件是描述实体集的，所以它又可以定义为描述一个实体集的所有符号集。例如所有的学生记录组成了一个学生文件。

#### (4) 键(Key)

能惟一标识文件中每个记录的字段或字段集，称为记录的键。这个概念与实体的码概念是一致的。

三个世界的术语对应关系如表 1.1 所列。

表 1.1 三个世界术语的对应关系

现实世界	信息世界	计算机世界
事物总体	实体集	文件
事物个体	实体	记录
特征	属性	字段
事物之间的联系	实体模型	数据模型

## 1.5.2 数据模型概述

不同的数据模型实际上是提供给我们模型化数据和信息不同工具。根据模型应用的不同目的，数据模型大体可分为两类：

第一类是独立于任何计算机系统实现的，如实体—联系模型，这种模型完全不涉及信息在计算机系统上的表示，只是用来描述某个特定组织所关心的信息结构，称为概念数据模型，简称为概念模型。这类模型强调其语义表达能力，概念简单、清晰、易于用户理解，它是现实世界的第一层抽象，是用户和数据库设计人员之间进行交流的语言。概念模型是用于建立信息世界的数据库模型，是按用户的观点来对数据和信息建模，主要用于数据库的设计。

另一类数据模型则是直接面向数据库中的数据逻辑结构 例如有关系、网状、层次、面向对象等模型，这类模型涉及到计算机系统，称为结构数据模型，简称为数据模型。这类模型有严格的形式定义，以便于机器上的实现。它通常有一组严格定义了语法和语义的语言。人们使用它来定义、操纵数据库中的数据，数据模型是现实世界的第二层抽象，是按计算机系统的观点对数据建模，主要用于 DBMS 的实现。

任何一种数据模型都是有严格定义的，包括模型的静态特性和动态特性，通常数据模型有三个要素：

### 1. 数据结构

数据结构用于描述数据库系统的静态特性。数据模型是所描述的对象类型的集合，包括两类：一类是与数据类型、内容、性质有关的对象，例如关系模型中的域、属性、关系等；另一类是与数据之间联系有关的对象，例如网状模型中的系型(Set Type)。

通常，人们按照数据结构的类型来命名数据模型，如层次结构、网状结构、关系结构所对应的数据模型分别命名为层次模型、网状模型和关系模型。

### 2. 数据操作

数据操作用于描述数据库系统的动态特性。数据操作是指对数据库中各种对象执行操作的集合，包括操作及有关的操作规则，主要有检索和更新（包括插入、删除、修改）两大类操作。数据模型必须定义这些操作的确切含义、操作符号、操作规则（如优先级）以及实现操作的数据库语言。

### 3. 数据的约束条件

数据的约束条件是一组完整性规则的集合。完整性规则是数据模型中数据及其联系所具有的制约和存储规则，以保证数据库中的数据的正确性与相容性。一般，数据库都提供定义完整性约束条件的机制。

## 1.5.3 概念数据模型

概念数据模型是用户与数据库设计人员之间进行交流的工具。常见的概念数据模型有实体联系模型(Entity Relationship Model，简称 E-R 模型)，随着数据库应用的深入，将传统的 E-R 模型进行改进，有了扩充 E-R 模型(Enhanced-ER 模型，简称 EER 模型)。另外，将面向对象建模所使用的统一建模语言(Unified Modeling Language，简称 UML)引入，成为新的概念模型(UML 将在第 8 章介绍)。

### 1. E-R 模型

E-R 模型是 P. P. S. Chen 于 1976 年提出的。E-R 模型是用 E-R 图来描述概念模型的一种常用的表示方法。E-R 模型的基本语义单位是实体与联系，它可以形象地用图形表示实体—联系及其关系。E-R 图是直观表示概念模型的有力工具，该方法简单实用。E-R 图有三要素：

#### (1) 实体

用矩形框表示，框内标注实体名称。

#### (2) 属性

用椭圆形表示，并用连线与实体或联系连接起来。

#### (3) 实体间的联系

用菱形框表示，框内标注联系名称，并用连线将菱形框分别与有关实体相连。

实体间的联系有两种方式：一种是同一实体集中实体间的联系，另一种是不同实体集中实体间的联系。我们主要研究第二种。实体间的联系虽然复杂，但可分解到最基本的两个实体间的联系。实体间的联系可归纳为三种类型：

① 一对一联系 (1:1)。如果实体集  $A$  和  $B$  中的每一个实体至多和另一个实体集中的一个实体有联系，那么实体集  $A$  和  $B$  的联系称为一对一联系，记作 1:1。例如，飞机的乘客和座位之间、学校与校长之间等都是 1:1 的联系，要注意的是 1:1 联系不一定是一一对应。如图 1.12 所示。

② 一对多联系 (1:n)。若实体集  $A$  中每个实体与实体集  $B$  中多个任意实体 ( $n \geq 0$ ) 有联系，而实体集  $B$  中每个实体至多与实体集  $A$  中一个实体有联系，那么称从  $A$  到  $B$  是“一对多联系”，记为 1:n。例如，部门与职工之间、班级与学生之间、车间和工人之间、系与学生之间，都是一对多联系。1:1 联系是 1:n 联系的一个特例，即  $n=1$  时的 1:n。如图 1.13 所示。

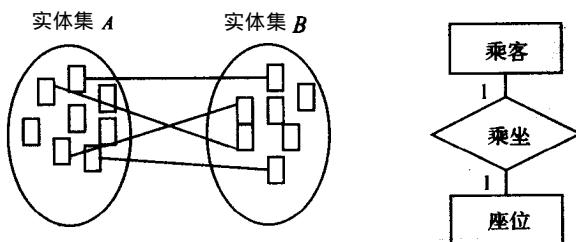


图 1.12 一对一联系

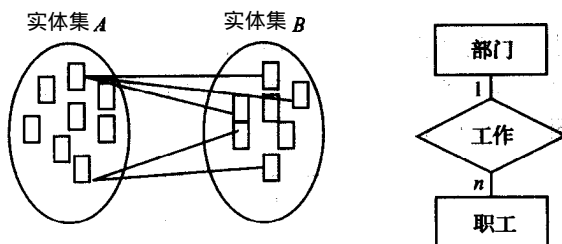


图 1.13 一对多联系

多对多联系 ( $m:n$ )。若实体集  $A$  和实体集  $B$  中允许每个实体都和另一个实体集中多个任意实体有联系，那么称  $A$  和  $B$  为多对多联系，记为  $m:n$ 。例如，图书与读者之间、学生与课程之间、电影院和观众之间、商店和顾客之间都是多对多的关系。如图 1.14 所示。

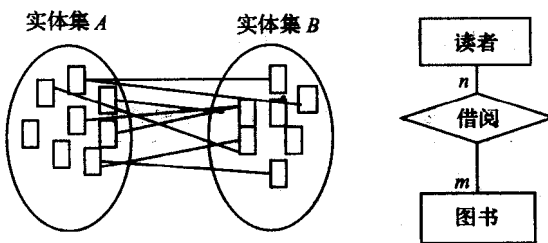


图 1.14 多对多联系