第1篇

技　术　篇

第1章 数据库系统概述

数据库技术是计算机科学的重要分支，主要用于对数据进行管理。今天，信息资源已成为各个企业的重要财富，企业的成功与否很大程度上依赖于它是否能够及时准确地获取有关企业运营的各项数据，对它们进行有效的管理和分析，并据此来引导企业的决策和行为。随着企业数据容量的急剧增长和内容的迅速变化，建立一个满足信息处理要求的行之有效的数据管理系统已成为一个企业生存和发展的重要条件。因此，数据库技术得到了越来越广泛的应用，越来越多新的应用领域采用数据库来存储和管理它们的信息资源。本章介绍数据库系统的一些基本概念和常用术语，作为后面各章节的准备和基础。

1.1 数据库技术的发展

数据库技术是应数据管理任务的需要而产生的。它最初主要用来处理一类被称为数据密集型的应用，例如飞机订票系统、银行信息系统、部门财务系统、情报检索系统等。这类应用具有的特点是：涉及的数据量大，数据需要长期保存并可以被许多应用程序所共享。如何对这种大量的、持久的、共享的数据进行管理，从20世纪50年代末以来就一直成为计算机科学技术领域中的重要研究课题。

早期的数据管理都采用文件系统。在文件系统中，数据根据其内容、结构和用途被组织成相互独立的文件，利用“按文件名访问，按记录进行存取”的管理技术，可以对文件进行增、删、改操作。但文件系统存在以下缺点：

① 数据共享性差，冗余度大。在文件系统中，一个文件基本上对应于一个应用程序，即文件是面向应用的。当不同的应用程序具有部分相同的数据时，也必须建立各自的文件，而不能共享相同的数据，因此数据的冗余度大，浪费存储空间。同时由于相同数据的重复存储、各自管理，容易造成数据的不一致性，给数据的修改和维护带来了困难。

② 数据独立性差。文件系统中的文件是为某一特定应用服务的，文件结构的修改将导致应用程序的修改。随着应用环境和需求的变化，修改文件的结构是常有的事，例如扩充某些字段的长度，改变某些字段的表示格式等。另一方面，应用程序的改变，也将引起文件结构的改变，例如应用程序改用不同高级语言等。可见，文件系统中应用程序和数据之间缺乏独立性。

③ 由于数据缺乏统一管理，在数据的结构、编码、表示格式、命名以及输出格式等方面不容易做到规范化、标准化。在数据的安全和保密方面也难以采取有效的措施。

针对文件系统的上述特点，人们逐步开发了以统一管理和共享数据为主要特征的数据库系统（database system）。在数据库系统中，数据不再针对某一应用，而是面向全组织，具有整体的结构。数据库系统从整体角度看待和描述数据，数据不再面向某个应用，而是面向整个系统，因此数据可以被多个用户、多个应用程序共享使用。数据共享可以大大减少数据冗余，节约存储空间，同时还能够避免数据之间的不相容性与不一致性。数据库系统中的数据由一个称为数据库管理系统（database management system，DBMS）的软件统一管理。由于有 DBMS 的统一管理，用户不必关心数据存储和其他实现的细节，可以在更高的抽象级别上观察和访问数据。文件结构的一些修改也可以由 DBMS 屏蔽，使用户看不到这些修改，从而减少应用程序的维护工作量，提高数据的独立性。由于数据的统一管理，人们可以从全单位着眼，合理组织数据，减少数据冗余；还可以更好地贯彻规范化和标准化，从而有利于数据的转移和更大范围内的共享。

世界上第一个通用的 DBMS 诞生于 20 世纪 60 年代，由通用电气公司的 Charles Bachman 设计，称为集成数据存储系统（integrated data store，IDS）。该系统奠定了网状数据模型的基础，极大地影响了数据库系统的发展。Charles Bachman 也因此于 1973 年成为图灵奖（相当于计算机科学界的诺贝尔奖）得主。

1969 年，IBM 公司推出了第一个商品化的数据库管理系统（information management system，IMS），它基于层次数据模型。与此同时，美国航空公司和 IBM 公司联合开发了用于预订机票的 SABRE 系统，该系统允许多个人通过计算机网络同时访问相同的数据。

层次和网状数据模型是从过去应用程序处理数据时所用的数据结构概括和发展来的。层次数据模型基于树，而网状数据模型基于图。这两种数据模型具有一定的通用性，但其中保留了不少实现的细节，例如指针，使得用户观察和访问数据的抽象级别还不够高，数据独立性还不够好，数据库的使用也不够方便。但它们为数据库技术奠定了基础，搭起了框架，打开了应用局面，至今还有许多这样的系统在运行。

数据库技术最有意义的成就是关系数据库的发展。1970 年，IBM 公司 San Jose 实验室的 Edgar Frank Codd 提出了关系数据模型，以关系或表作为描述数据的基础。在其后的几年中，Codd 又发表了一系列文章，奠定了关系数据库的理论基础。20 世纪 70 年代是关系数据库理论研究和原型开发的时代，其中以 IBM San Jose 实验室开发的 System R 和 Berkeley 大学研制的 INGRES 为典型代表。这两个原型系统差不多都在 1977 年前后研制成功并开始运行。它们提供了先进的关系 DBMS（简记为 RDBMS）技术，为研制商品化的 RDBMS 完成了技术上的准备。IBM 公司在 System R 的基础上先后推出了 SQL/DS 和 DB2 两个商品化的 RDBMS。INGRES 也由 INGRES 公司商品化。20 世纪 80 年代，几乎所有新开发的系统均是关系型的。这些商用数据库系统的运行，特别是微机 RDBMS 的使用，使数据库技术日益广泛地应用到企业管理、情报检索、辅助决策等各个方面，成为实现和优化信息系统的实用技术。Edgar Frank Codd 因此于 1981 年获得了图灵奖。

20 世纪 80 年代后期直至 90 年代，数据库技术经过大量高层次的研究和开发，在查

询语言、数据模型、复杂的数据分析等方面取得了一系列的研究成果。许多数据库提供商(如 IBM,Oracle,Sybase,Microsoft 等)纷纷扩展它们的数据库系统,增加其对复杂数据类型(如图像或文本)的存储能力和对复杂查询的处理能力等。有的提供商(如 NCR 公司)还专门开发了数据仓库(data warehouse)系统,对来自多个数据库的数据进行联机分析处理。

进入 21 世纪,数据库管理系统也不可避免地跨入了因特网时代。早期的 Web 站点只访问 HTML 文件或 XML 文件,现在则需要访问数据库中的数据。人们通过特定的 Web 格式发送查询,查询结果以某种容易被浏览器所显示的格式返回。

现在,随着计算机网络的发展,数据访问变得更加容易,对数据进行联机处理的需求迅速增长,数据管理变得越来越重要。一些新的应用领域,如计算机辅助设计/管理(CAD/CAM)、计算机集成制造(CIM)、办公信息系统(OIS)、地理信息系统(GIS)、知识库系统和实时系统、决策支持和数据挖掘等,均需要数据库的支持,它们为数据库应用开辟了新的天地。另一方面,数字图书馆、多媒体、交互式视频等新技术的出现也直接推动了数据库技术的研究和发展。

1.2　数据库的基本概念

在系统地介绍数据库之前,首先介绍一些数据库最常用的术语和基本概念。

1. 数据

数据在大多数人头脑中的第一反应就是数字。其实数字只是最简单的一种数据,是数据的一种传统和狭义的理解。实际上,数据的种类很多,可以是文字、图形、图像、声音、学生档案记录、货物的运输情况等。

为了了解世界,交流信息,人们需要描述各种事物。在日常生活中直接用自然语言(如汉语)描述。在计算机中,为了存储和处理各种事物,就要抽出对这些事物感兴趣的特征组成一个记录来描述。例如:在学生档案中,如果人们最感兴趣的是学生的姓名、性别、年龄、出生年份、籍贯、所在系别、入学时间,那么可以这样描述:(陆鸣,男,21,1982,江苏,计算机系,2000)。这样的记录,一般人可能不解其义,但是了解这个记录含义的人会得到如下信息:陆鸣是个大学生,1982 年出生,男,江苏人,2000 年考入计算机系。这种对事物描述的符号记录称为数据。数据有一定的格式,例如上例中,姓名栏最多允许 4 个汉字,性别栏允许 1 个汉字,等等。这些格式的规定是数据的语法;而数据的含义是数据的语义。人们通过解释、归纳、分析、综合等方法,从数据所获得的有意义的内容称为信息。因此,数据是信息存在的一种形式,只有通过解释或处理才能成为有用的信息。

2. 数据库(database,DB)

数据库是长期存储在计算机内、有组织的、可共享的数据集合。以前,人们把数据存放在文件柜里,或存放在电子文件里,现在人们借助于计算机技术,将大量复杂的数据科学地保存在数据库里,以便能方便而充分地利用这些宝贵的资源。

数据库中的数据是按一定的格式存放的，具有较小的冗余度，较高的数据独立性和易扩展性，并可为各种用户共享。例如，有关一个大学的数据库可能包含如下信息：

- 实体　如学生、系或学院、课程、教师、教室等；
- 实体间的关系　如学生选课、教室使用情况等。

3. 数据库管理系统（database management system，DBMS）

数据库管理系统是对数据库进行管理的软件。它位于用户和操作系统之间，主要任务是按一定的格式组织数据，将其存放在数据库中并进行高效处理，同时负责对数据库中的数据进行维护。数据库管理系统使用户能方便地定义数据和操纵数据，并能够保证数据的安全性、完整性、多用户对数据的并发使用及发生故障后的系统恢复。

4. 数据库系统（database system，DBS）

数据库系统是指在计算机系统中引入数据库后的系统构成，一般由数据库、数据库管理系统（及其开发工具）、应用系统、数据库管理员构成。应当指出的是，数据库的建立、使用和维护等工作只靠一个 DBMS 是不够的，还要有专门的人员来完成，这些人称为数据库管理员（database administrator，DBA）。

在不引起混淆的情况下，人们常把数据库系统简称为数据库。

1.3　数据库系统

数据库系统的结构如图 1-1 所示。图中的数据库是数据的汇集，它们以一定的组织形式存于存储介质上，一般是磁盘。DBMS 是管理数据库的软件，它实现数据库系统的各种功能。数据库管理员负责数据库的规划、设计、协调、维护和管理等工作。应用系统是指以数据库为基础的各种应用程序，应用程序必须通过 DBMS 访问数据库。这里的用户指最终用户，通过应用系统的用户接口使用数据库，常用的接口方式有浏览器、菜单驱动、表格操作、图形显示、报表书写等，给用户提供简明直观的数据表示。

从图 1-1 可以看出，DBMS 是数据库系统的核心。选择 DBMS 是设计数据库系统的关键一步。但是，有了 DBMS 并不等于建立了数据库系统，还要设计和装入数据，开发以数据库为基础的应用程序，进行数据库系统的调整、维护和管理等。一般来说，在数据库领域有两个方面的工作要做：一是实现 DBMS，这是系统软件研制者的任务；二是用 DBMS 构成数据库系统，这是数据库应用开发人员的任务。本书的重点是数据库技术与应用，故在这里只对 DBMS 的功能和组成作一简单介绍，后续章节将主要介绍应用 DBMS 来设计、建立、使用、管理和维护数据库及其应用系统的技术。

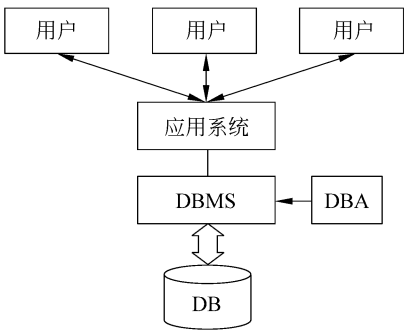

图 1-1　数据库系统的组成

1.3.1　DBMS 的功能

DBMS 是一个通用的平台软件，也称为系统软件、基础软件，由厂家提供。由于 DBMS 实现的硬件资源、软件环境不同，所以 DBMS 的功能和性能便有差异。但不管有多少差异，它们通常都具有如下几个方面的基本功能。

1. 数据库定义

DBMS 提供数据定义语言（data definition language，DDL），通过它可以方便地对数据库的模式结构、数据库的完整性、数据库的安全性等进行定义。这些定义存储在数据字典（也称为系统目录）中，是 DBMS 运行的基本依据。

2. 数据存取

DBMS 提供数据操纵语言（data manipulation language，DML），用户可以使用 DML 操纵数据，实现对数据库的查询、插入、删除和修改。

3. 数据库运行管理

数据库在运行时由 DBMS 统一管理、统一控制，以保证数据的安全性、完整性、多用户对数据的并发使用及发生故障后的系统恢复。

4. 数据组织、存储和管理

DBMS 要分类组织、存储和管理各种数据，包括数据字典、用户数据、存取路径等。要确定以何种文件结构和存取方式在存储介质上组织这些数据，如何实现数据之间的联系。数据组织和存储的基本目标是提高存储空间利用率和方便存取，提供多种存取方法（如索引查找、HASH 查找、顺序查找等），提高存取效率。

5. 数据库的建立和维护

DBMS 的功能包括数据库初始数据的输入、转换功能，数据库的转储、恢复功能，数据库的重组、重构功能和性能监视、分析功能等。这些功能通常是由一些实用程序完成的。

6. 其他功能

DBMS 的功能还包括 DBMS 与网络中其他软件系统的通信功能；一个 DBMS 与另一个 DBMS 或文件系统的数据转换功能；异构数据库之间的互访和互操作功能等。

1.3.2　DBMS 的组成

作为一个庞大的系统软件，DBMS 由众多程序模块组成，它们分别实现 DBMS 复杂而繁多的功能。正如在前面所说的，不同的 DBMS 功能并不完全相同。大型系统功能完备，小型系统则常常对系统功能做了裁减。但大致来说，DBMS 的程序模块按功能划分如图 1-2 所示。

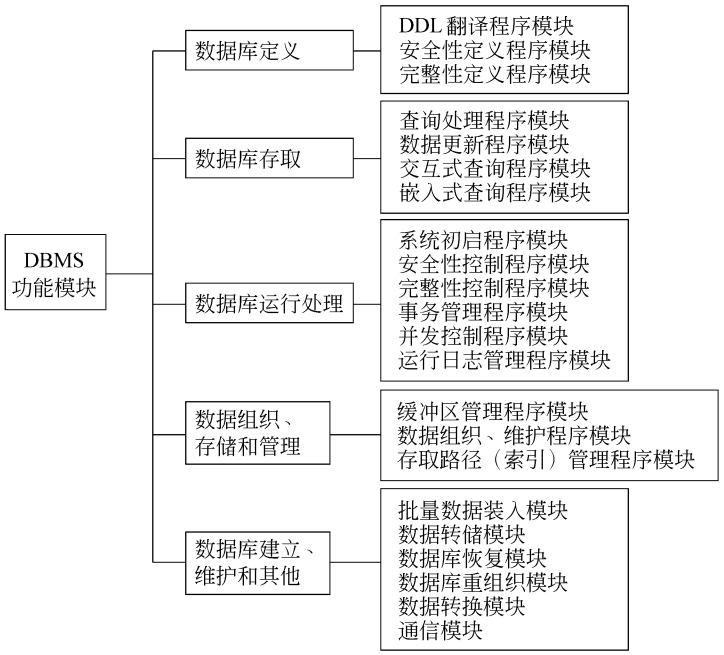

图 1-2　DBMS 程序模块结构

1. 数据定义方面的程序模块

数据定义的程序模块主要包括：

- 模式、外模式、存储模式的定义模块，在 RDBMS 中就是创建数据库、创建表、创建视图、创建索引等定义模块；
- 安全性定义，如授权定义及处理模块；
- 完整性定义，如主码、外码、其他完整性约束定义及处理模块。

这些 DDL 程序模块接收相应的定义，进行语法、语义检查，把它们翻译为内部格式存储在数据字典中。创建数据库的模块还根据定义建立数据库的框架（即形成一个空库），等待装入数据。

2. 数据操纵方面的程序模块

数据操纵的程序模块主要包括：

- 查询（SELECT 语句）处理程序模块；
- 数据更新（增、删、改）程序模块；
- 交互式查询程序模块；
- 嵌入式查询程序模块。

这些程序模块对用户的数据操纵请求进行语法分析、语义检查，生成某种内部表示，通常是语法树。对于查询语句，要由查询优化器（模块）进行优化，如根据一定的等价变换规则把语法树转换成标准（优化）形式；对于语法树中的每一个操作根据存取路径、数据的存储分布、数据的聚簇等信息来选择具体的执行算法；最后生成查询计划（生成代

码）交查询执行模块执行，完成对数据库的存取操作。

3. 数据库运行管理方面的程序模块

数据库运行管理方面的程序模块主要有系统初启程序，负责初始化 DBMS，建立 DBMS 的系统缓冲区、系统工作区、打开数据字典等。还有安全性控制、完整性检查、并发控制、事务管理、运行日志管理等程序模块，在数据库运行过程中监视对数据库的所有操作，控制管理数据库资源，处理多用户的并发操作等。它们一方面保证用户事务的正常运行，另一方面保证数据库的安全性和完整性。

4. 数据库组织、存储和管理方面的程序模块

数据库组织、存储和管理方面的程序模块有文件读写与维护程序、存取路径（如索引）管理和维护程序、缓冲区管理程序（包括缓冲区读、写、淘汰模块）等。这些程序负责维护数据库的数据和存取路径，提供有效的存取方法。

5. 数据库建立、维护和其他方面的程序模块

其他模块有数据库初始装入程序、转储程序、恢复程序、数据库重构造程序、数据转换程序、通信程序等。

DBMS 的这些组成模块互相联系，互相依赖，共同完成 DBMS 的复杂功能。

1.4　数据模型

模型是现实世界特征的模拟和抽象。例如，建筑上使用的沙盘、军事上使用的地图等，都是具体的模型。一眼望去，就会使人联想到真实生活中的事物。数据模型也是一种模型，它是现实世界数据特征的抽象。

数据库是某个企业、组织或部门所涉及的数据的综合，它不仅要反映数据本身的内容，而且要反映数据之间的联系。由于计算机不可能直接处理现实世界中的具体事物，所以人们必须事先把具体事物转换成计算机能够处理的数据。在数据库中用数据模型这个工具来抽象、表示和处理现实世界中的数据和信息。

一般来说，数据模型应满足两方面的要求：一是能比较真实地模拟现实世界，容易被人所理解；二是便于在计算机上实现。但是，很难有一种数据模型能同时满足这两个方面的要求。从用户的角度来讲，总是希望数据模型尽可能自然地反映现实世界和接近人对现实世界的观察和理解，也就是说数据模型要面向现实世界，面向用户；从实现的角度来看，又希望数据模型接近数据在计算机中的物理表示，以便于实现和减小开销，也就是说数据模型还不得不在一定程度上面向实现、面向计算机。这两方面的要求显然是矛盾的。数据库中解决这个矛盾的途径有点类似于程序设计语言。在程序设计语言中，有面向用户的高级程序设计语言，也有面向计算机的汇编语言，有时还有介于两者之间的中间语言，各有各的用途，而由编译系统完成高级到低级程序设计语言的转换。在数据库中，也是针对不同的使用对象和应用目的，采用不同的数据模型。一般可分为下面 3 级：

（1）概念数据模型

概念数据模型独立于计算机系统，它完全不涉及信息在计算机系统中的表示，只是用来描述某个特定组织所关心的信息结构，是按用户的观点来对数据和信息建模，是对企业主要数据对象的基本表示和概括性描述，主要用于数据库设计。这类模型强调其语义表达能力，概念应该简单、清晰，易于用户理解，是数据库设计人员和用户之间进行交流的工具。概念数据模型与 DBMS 无关，这使得数据库设计人员可以在设计的开始阶段，把主要精力用于了解和描述现实世界上，而把涉及 DBMS 的一些技术性问题推迟到设计阶段去考虑。

（2）逻辑数据模型

逻辑数据模型直接面向数据库的逻辑结构，通常是有一组严格定义、无二义性的语法和语义的数据库语言，人们可以用这种语言来定义、操纵数据库中的数据。逻辑数据模型与 DBMS 有关，DBMS 以所支持的逻辑数据模型来分类。用概念数据模型表示的数据必须转化为逻辑数据模型表示的数据，才能在 DBMS 中实现。逻辑数据模型既要面向用户，也要面向实现。

（3）物理数据模型

物理数据模型是对数据最低层的抽象，它描述数据在磁盘或磁带上的存储方式和存取方法，是面向计算机系统的。每种逻辑数据模型在实现时，都有其对应的物理数据模型。物理数据模型的实现不但与 DBMS 有关，还与操作系统和硬件有关。

在设计一个数据库时，首先需要将现实世界抽象成概念数据模型，然后将概念数据模型转换为逻辑数据模型，最后将逻辑数据模型转换为物理数据模型。前两步是由数据库设计人员完成的，后一步是由 DBMS 完成的。本书将重点介绍概念数据模型、逻辑数据模型以及它们之间的转换，有关物理数据模型的细节以及逻辑数据模型到物理数据模型的转换不在本书讨论范围之列，感兴趣的读者可以参考有关 DBMS 实现方面的文献。

下面首先介绍数据模型的共性——数据模型的组成要素，然后分别介绍两类不同的数据模型——概念数据模型和逻辑数据模型。

1.4.1 数据模型的组成要素

一般地讲，数据模型是严格定义的一组概念的集合。这些概念精确地描述了系统的静态特性、动态特性和完整性约束条件。因此数据模型通常由数据结构、数据操作和完整性约束 3 部分组成。

1. 数据结构

数据结构是所研究的对象类型的集合。这些对象是数据库的组成成分，它们包括两类，一类是与数据类型、内容、性质有关的对象，例如网状数据模型中的数据项、记录，关系数据模型中的域、属性、关系等；一类是与数据之间联系有关的对象，例如网状数据模型中的系型（Set Type）。

数据结构是刻画一个数据模型性质的最重要的方面。因此在数据库系统中，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数据模型、网状数据模型和关系数据模型。

数据结构是对系统静态特性的描述。

2. 数据操作

数据操作是指对数据库中各种对象(指数据的型)的实例(指数据的值)允许执行的操作的集合,包括操作及有关的操作规则。数据库主要有检索和更新(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)以及实现操作的语言。数据操作是对系统动态特性的描述。

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。完整性规则是给定的数据模型中数据及其联系应满足的制约和依存规则,用以限定符合数据模型的数据库状态以及状态的变化,以保证数据的正确、有效、相容。

数据模型应该反映和规定本数据模型必须遵守的基本的、通用的完整性约束条件。例如,在关系数据模型中,任何关系必须满足实体完整性和参照完整性两个条件(第 2 章将详细讨论这两个完整性约束条件)。

此外,数据模型还应该提供定义完整性约束条件的机制,以反映具体应用所涉及数据必须遵守的特定的语义约束条件。例如,在学校的数据库中规定大学生入学年龄不得超过 30 岁,硕士研究生入学年龄不得超过 38 岁,学生累计成绩不得有 3 门以上不及格等。

需要注意的是,有个别数据模型,并不完全包含上述 3 个部分。例如下面将要介绍的 E-R 数据模型,只有描述数据静态特性和数据约束的手段,还缺少操作的定义。

1.4.2　概念数据模型

为了把现实世界中的具体事物抽象、组织为某一 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,然后将信息世界转换为机器世界。也就是说,首先把现实世界中的客观对象抽象为某一种信息结构,这种信息结构并不依赖于具体的计算机系统,不是某一个 DBMS 支持的数据模型,而是概念级的模型。然后再把概念模型转换为计算机上某一 DBMS 支持的数据模型。最典型的概念数据模型是实体联系数据模型(Entity-Relationship data model, E-R 数据模型)。下面以其为例对概念数据模型进行介绍。

1. 基本概念

信息世界涉及的基本概念主要有:

- 实体(Entity)　客观存在并可相互区别的事物称为实体。实体可以是具体的人、事、物或抽象的概念。例如:玩具,玩具店、玩具店店主等都是实体。
- 属性(Attribute)　实体所具有的某一特性称为属性。一个实体可以由若干个属性来刻画。例如:雇员实体可以由雇员号、姓名、部门、性别、年龄、工作岗位等属性组成。(10001,王平,市场部,男,38,销售经理)这些属性组合起来表征了一个雇员。
- 实体型(Entity Type)　具有相同属性的实体必然具有共同的特征和性质。用实