

第 1 章 绪 论

在介绍数据库的基本概念前，我们首先应该搞清楚什么是数据库。

在计算机技术的应用中，很重要的一条就是对数据进行整理、加工和存储，比如对一个公司的员工资料、商店的进货出货状况资料、网站的用户资料等。这些数据在经过前端的收集后，我们不可以把它们零散地存放在计算机中。即使我们集中存放这些数据，也不可以简单地将其排成一行，否则会在使用这些数据时产生困难。因此，必须对收集的数据进行处理和加工，然后存放在一个便于管理的地方，才能达到数据收集的目的。

数据库指的是某企业、组织或部门等所涉及的某些种类的数据的集合。它不仅反映了数据本身的内容，还反映了数据间的联系。它与数据库管理系统和数据库应用程序共同组成数据库系统。通俗地说，数据库系统提供了一种把与我们的工作和生活紧密相关的信息集合在一起的方法，它还提供了在某个集中的地方存储和维护这些信息的方法。

数据库技术就是科学地组织和储存数据以及高效地获取和处理数据的技术。数据库技术是本世纪 60 年代以后发展起来的数据管理技术。数据库系统是当代计算机系统的重要组成部分。

1.1 数据管理技术的发展

本节主要介绍：

- 1) 数据管理技术的发展历史；
- 2) 数据库系统的特点。

在读完本节后，用户可以了解数据管理技术的三个阶段和数据库系统的特点。

在当今的信息社会里，人们的全部社会活动（生产、交流、生活等）都离不开信息。

信息总是以数据为载体来反映和记录的。而对数据进行的采集、储存、分析加工、检索使用和维护，更是我们每天自觉地进行着的繁琐工作。

我们使用数据库就是要进行数据处理，数据处理是指对数据进行收集、存储、加工和传播的一系列活动的综合。其目的是对所收集的大量数据进行有规律的集中和组织，借助计算机系统科学地保存和管理大量的复杂数据，以便于人们更加有效地利用这些数据。

数据处理的中心问题就是数据管理。数据管理指的是对数据的分类、组织、存储、查询和维护。过去半个多世纪以来，数据管理技术经历了人工管理阶段（50年代中期以前）、文件系统阶段（50年代中期到60年代中期），在60年代以后，由于计算机在管理方面的应用越来越广，数据量也急剧增长，对数据共享的要求也越来越高，数据管理技术进入到数据库系统阶段。

1.1.1 人工管理阶段

50 年代中期以前，对数据的管理使用的是人工管理。那时计算机是少数科学家手中的珍品，主要用于科学计算。数据的载体只有磁带、卡片、纸带，没有操作系统、管理数据的软件，数据处理的主要方式是批处理。

这一阶段，因为数据主要用于科学计算，所以不需要长期保存。同时，由于没有软件对数据进行管理，程序员要负担设计数据的逻辑结构、物理结构等，数据的管理十分复杂繁琐。

1.1.2 文件管理阶段

在 50 年代中期以后出现了文件管理系统，它把数据组织在一个个独立的数据文件中。每个文件都有完整的体系结构，文件内全部数据由若干记录构成，而记录又由若干数据项组成。对数据的操作是按文件名访问，按记录进行存取的。但是在文件之间没有任何联系，是孤立的，即不是结构化的。各个应用程序都对应自己的文件，数据冗余度大，浪费存储空间。

1.1.3 数据库系统阶段

随着人类社会的不断发展、进步，必然引起数据信息量的飞速膨胀，对数据处理的精度、速度的要求不断提高。除了要求在一个数据集合（文件集合）内部建立数据之间的联系外，在各数据集合之间也需要建立这样或那样的联系，以实现多用户、多应用共享数据的要求。这样，传统的文件管理系统已经无法满足需求。为了适应这种不断增长的要求，迫切需要更先进的数据组织与管理技术。于是在 60 年代后期，数据管理方法进入数据库系统阶段。

早期的数据库管理系统以集中应用为基础，所有的应用都局限于某一固定的计算机系统上集中运行。这种系统资源消耗大，对硬件系统依赖性强，使得大型数据库系统只能在大型机上运行，限制了数据库系统的应用发展。

80 年代以后，微型计算机的发展很快，使以往只能在大型机上运行的数据库管理系统同样可以在微机上运行。由于大型数据库系统开销大，早期微机设备能力低，所以像 dbase、foxbase 等一些简单的 PC 数据库管理系统得以迅速的发展，占领了微机的广大市场。到目前为止，我国 PC 机数据库系统仍然以这些传统的数据库管理系统为主，大多数的用户也是从此步入数据库系统的大门的。直到近几年，一些先进的关系数据库系统如 MS SQLServer、Sybase、Oracle 等才开始大规模地出现在各行各业的应用系统中。

1.1.4 数据库系统的特点

数据库系统有以下几个特点。

1. 数据结构化

在文件系统中，文件内部的数据一般是有结构的，但文件之间不存在联系，因此从数据的整体来说是没有结构的。数据库系统虽然也常常分成许多单独的文件，并且文件的内部也具有完整的数据结构，但是它更注意同一数据库中各文件之间的相互联系，故而特别能适应大量数据管理的客观需要。

2. 数据共享和冗余控制

共享是数据库系统的目的，也是它的重要特点。一个数据库中的数据，不仅可以为同一企业或组织内部的各部门共享，而且在文件系统中，数据总是由特定用户专用的。

在文件系统中，由于每个应用都拥有并使用自己的数据，各数据文件中难免有许多数据相互重复，这就是冗余，它造成了存储空间巨大浪费。数据库系统是面对整个系统的数据共享而建立的，各个应用的数据集中存储，共同使用，因而尽可能地避免了数据的重复存储，减少和控制了数据的冗余。

3. 数据独立性

在文件系统中，数据结构和应用程序是相互依赖的，任何一方的改变总是要影响另一方的改变。而在数据库系统中，这种相互依赖性是很小的，数据和程序具有相对的独立性。

数据库系统提供了两方面的映像功能。一个是数据的存储结构（或物理结构）和逻辑结构之间的映像（或转换）功能；另一个是数据的总体逻辑结构和某类应用所涉及的局部逻辑结构之间的映像（或转换）功能。第一种映像使数据存储结构（或物理结构）改变时逻辑结构不变，因而相应的程序也不变，这就是它们的物理独立性；第二种映像使逻辑结构改变时，用户结构不变，应用程序也不用改变，这是数据和程序的逻辑独立性。

由于物理和逻辑的独立性，使得应用程序的编写再不用考虑数据的描述和存取路径等细节问题，简化了应用程序的编制，从而大大减少了应用程序的修改和维护工作。

1.2 关系模型和关系数据库

本节主要讲述：

- 1) 数据模型的演变；
- 2) 关系模型的概念；
- 3) 关系数据库理论；

通过本章的学习，用户应对数据库系统中最重要的关系模型以及关系数据库理论有一定的了解。

1.2.1 数据模型

数据库中数据从整体来看是有结构的，即所谓数据的结构化。按照实现结构化所采取的不同联系方式，数据库的整体结构可区分为三类数据模型：层次模型、网络模型、关系模型。其中前两类又可称为“格式化模型”。

早期出现的是层次数据库，如 1969 年 IBM 公司开发的 IMS（Information Management System）数据库管理系统。层次数据库的特点是数据实体之间按层次关系来定义。由于在实用中它不能很好地表达实体间的复杂关系，又产生了网状数据库，如 HP 的 Turboimage 数据库管理系统。它很好地解决了实体间复杂关系的表达问题，但是它也有致命的弱点，这就是当需求扩展时，对原有数据结构及应用程序的修改工作量非常大。

1970 年，美国 IBM 公司 San Joes 研究实验室的研究员 E F Codd 发表了题为《大型共享数据库数据的关系模型》的论文，提出了关系模型的概念，首次运用数学方法来研究数据库的结构和数据操作，将数据库的设计从以经验为主提高到以理论为指导。不仅如此，关系模型运用人们通常使用的表格形式为存储结构，易学易用，使它从一开始就吸引了公众的注意，成为广大用户特别是微机用户乐于接受的数据模型。

1.2.2 关系模型和关系数据库

关系模型是三种数据模型中最重要的模型。自 80 年代以来，计算机厂商所推出的数据库管理系统几乎都是支持关系模型的，许多非关系系统的产品也都有专用的关系接口。本节的重点是介绍关系数据库。

关系模型是一种新的数据模型。它是建立在严格的数学理论、集合论和演算公式基础上的。它提供的逻辑结构简单，数据独立性强，存取具有对称性，操纵灵活。新设计的数据库系统，特别是微型计算机数据库管理系统，绝大部分都是关系（式）数据库管理系统。

关系模型将数据元素（文件）内部各数据项间的联系（局部逻辑结构）和各数据元素间的联系（整体逻辑结构）都表示成满足一定条件的二维关系表形式。即数据在用户观点下的逻辑结构是一张二维表。下面我们以某公司的员工登记表（如图 1-1 所示）为例介绍关系模型中的主要术语。

员工号	姓名	年龄	性别	部门	学历	职位
1176578	李强	27	男	研发部	硕士	工程师
1176579	林涛	32	男	客服部	本科	经理
1176580	张锦涛	29	男	研发部	本科	工程师
1176581	赵娜	24	女	行政部	大专	助理
1176582	许玮	25	男	行政部	本科	经理
1176721	刘建明	27	男	研发部	本科	工程师
1176722	赵逸民	26	男	市场部	本科	营销员
1176723	陈志军	33	男	市场部	硕士	经理
1176724	沈丽	25	女	研发部	本科	工程师
1176725	张宇	26	女	客服部	本科	助理

图 1-1 关系名：员工登记表

- 关系：一个关系对应一张二维表。
- 元组：表中的一行称为一个元组。
- 属性：表中的一列称为属性，每列的名称即属性名。
- 主码：表中的某个属性组，它们惟一的标识是一个元组。
- 域：属性的取值范围。

用这种关系模型设计的数据库系统就是关系数据库系统。关系数据库系统一般都由许多不同的关系构成，其中每个关系就是一个数据元素，可以用一张二维表表示。数据元素间的每一组联系也是一个关系，也用一张二维表表示，取消了路径的概念。

关系数据模型的理论是集合论，每一个关系就是一个集合论的子集。在关系数据库系统中，对数据的各种处理是以集合运算为根据的。基本的关系运算有 3 种：选择、投影、和链接。完善的关系数据库管理系统以数据操纵语言及结构化查询语言（SQL）来实现各种关系运算。

对于关系数据库系统的详细而严格的定义，涉及到集合论、关系代数等，我们就不在此讲述，请用户参阅相关专业书籍。

1.3 数据库系统结构

本节主要介绍：

- 1) 数据库系统的三级模式结构；
- 2) 数据库系统的三个组成部分；
- 3) 带有数据库系统的计算机系统的构成；
- 4) 数据库管理系统（DBMS）；
- 5) DBMS 与操作系统；
- 6) 数据库应用程序。

通过学习，用户可以了解数据库系统的概况和 DBMS 这一重要概念。

数据库系统主要由三大部分组成：数据库管理系统（Data Base Management System，它专门负责组织和管理数据信息的程序）、数据库应用程序（Data Base Application，它使我们能够获取、显示和更新由 DBMS 存储的数据）、数据库（Data Base，按一定结构组织在一起的相关数据的集合）。

1.3.1 数据库系统的三级模式结构

我们先来描述一下数据库系统的体系结构，它是数据库系统的一个总的框架。尽管实际的数据库系统软件产品多种多样，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，数据存储的结构也各有不同，但是绝大多数的数据库系统在体系结构上都具有三级模式的结构特征。

数据库系统的三级模式结构由外模式、模式和内模式组成，如图 1-2 所示。

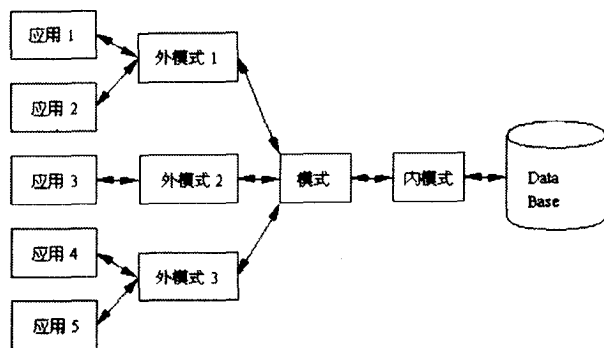


图 1-2 数据库系统的三级模式

● 外模式：

亦称子模式或用户模式，是数据库用户所看到的数据视图。

外模式是个别用户的数据视图，即与某一应用有关的数据的逻辑表示。

数据库的用户可能是应用程序员，也可能是一个联机的终端用户。不同用户的外模式可以互相覆盖，同一外模式可以为某一用户的任意多个应用程序所启用；一个应用（程序）只能启用一个外模式。

外模式通常是模式的子集，它是各个用户的数据视图，而不同的用户因其需求不同，看待数据的方式可以不同，对数据的保密要求也可以不同，编写应用程序所用的语言也可能不同。因此，不同用户的外模式的描述是有差异的。即使对模式中同一数据，在外模式中的结构、类型、长度、保密级别等都可以不同。

数据库系统提供外模式描述语言描述用户数据视图，用外模式描述语言写出的一个用户数据视图的逻辑定义的全部语句称为此用户的外模式。

外模式描述语言和用户选用的程序设计语言具有相容的语法。

● 模式：亦称逻辑模式，是数据库中所有数据的逻辑结构和特性的描述，是所有用户的公共数据视图。

模式是数据库中全部数据的一个逻辑表示或描述，它不同于内模式，比内模式抽象，不设计数据库的物理存储环境和硬件环境；它不同于外模式，与具体的应用程序和程序事件的程序语言也无关。

模式是数据库数据在逻辑级上的视图。它通常以某种数据模型为基础，模型不仅仅定义数据的逻辑结构，而且要定义与数据有关的安全性、完整性要求。不仅要定义数据记录内部的结构，而且要定义这些数据项之间的联系，进一步表示不同记录之间的联系。

数据库系统采用模式描述语言来严格地定义这些内容。用模式描述语言来写出一个数据库逻辑定义的全部语句，称为某一个数据库的模式。模式是对数据库的一种描述，而不是数据库本身，只是装配数据的一个框架。

● 内模式：亦称存储模式，是数据在数据库系统内部的表示，即对数据的物理结构和存储方式的描述。

内模式是全体数据库数据的内部表示或者低层次描述，用来定义数据的存储方式和物理结构。例如，数据是树形存储还是顺序存储，如何组织索引，数据是否压缩，是否加密

等。

内模式通常用内模式数据描述语言来描述和定义。数据库系统的三级模式结构是数据库的三个数据抽象级别，至于数据的具体组织是由数据库管理系统（DBMS）管理的，使用户可以逻辑地处理数据，而不必关心数据在计算机系统内部的表示和存储。

为了实现三个抽象层次的联系和转换，数据库系统在这三级层次里提供了两层映像：

- 外模式／模式映像。
- 模式／内模式映像。

对应于同一个模式。可有任意多个外模式。外模式／模式映像定义某一个外模式和模式之间的对应关系。这些映像定义通常包含在各自的外模式中。当模式改变时，外模式／模式的映像要相应改变，以保证外模式保持不变。

模式／内模式的映像定义数据逻辑结构和存储结构之间的对应关系。例如，说明逻辑记录的表示形式。当数据库的存储结构改变了，模式／内模式的映像也必须作出相应的改变，使得模式保持不变。

正是由于这二级映像的功能，使得数据库系统中的数据具有较高的逻辑独立性和物理独立性。

1.3.2 带有数据库系统的计算机系统的构成

带有数据库系统的计算机系统的软硬件层次如图 1-3 所示。

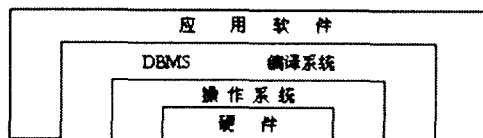


图 1-3 带有数据库系统的计算机系统的软硬件层次

实际的数据库系统由以下几部分构成：

(1) 硬件和数据库：一般数据库系统对硬件的要求是：足够大的内存（一般都有上千兆字节）来存放操作系统，还要有相应的 DBMS 核心模块、数据缓冲区和应用模块。由于数据库十分庞大，因此需要很大的磁盘（一般有几个至数百 TB 的容量不等，1TB=1000GB），有的还应有足够的磁盘作为数据库的备份。

(2) 软件 主要包括支持 DBMS 运行的操作系统和 DBMS 本身，还有用于开发应用系统的高级语言及其编译系统

(3) 人员：包括数据库管理员（Data Base Administrator，简称 DBA），系统分析员、应用程序员和用户。DBA 是数据库系统的总管，数据库的信息内容和结构、存储结构和策略、运行的监督和控制、完整性和安全性定义都由 DBA 来完成。

数据库系统中不同的人员涉及不同的数据抽象级别，具有不同的数据视图，如图 1-4 所示。

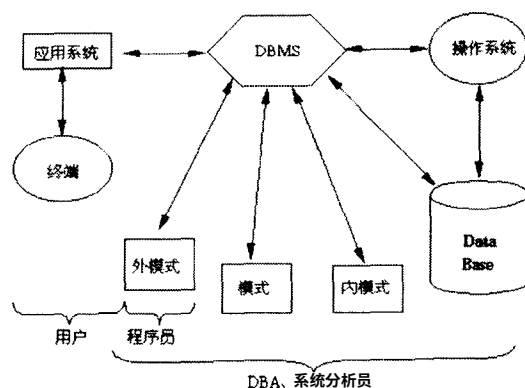


图 1-4 不同人员的数据视图

1.3.3 数据库管理系统（DBMS）

数据库管理系统（DBMS）是用于描述、管理和维护数据库的程序系统，是数据库系统的核心组成部分。它建立在操作系统的基础上，对数据库进行统一的管理和控制。其主要功能有如下所述的几个方面。

- （1）描述数据库：描述数据库的逻辑结构、存储结构、语言信息和保密要求等。
- （2）管理数据库：控制整个数据库系统的运行，控制用户的并发性访问，检验数据的安全、保密与完整性，执行数据检索、插入、删除、修改等操作。
- （3）维护数据库：控制数据库初始数据的装入，记录工作日志，监视数据库性能，修改更新数据库，重新组织数据库，恢复出现故障的数据库。
- （4）数据通信：组织数据的传输。

应用程序、DBMS、数据库的关系如图 1-5 所示。

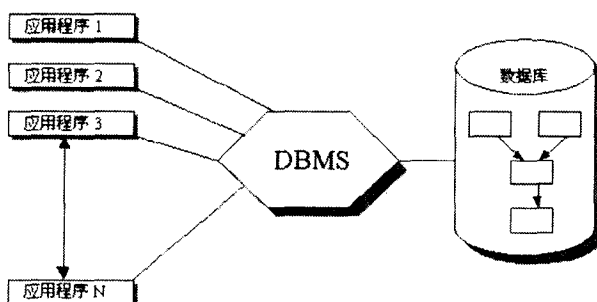


图 1-5 应用程序、DBMS、数据库的关系

关系数据库管理系统（RDBMS）是关系数据库系统的管理系统。作为一个庞大的系统软件，DBMS 由众多的程序模块组成，他们分别实现 DBMS 复杂而繁多的功能。DBMS 在不同规模的系统中的实现算法是不尽相同的。即使在同样规模的系统里，因为涉及的实际情况不同，也有很大的差异。我们下面用层次结构的观点介绍 DBMS 体系结构。DBMS

的程序模块可按功能划分为如图 1-6 所示的层次结构。

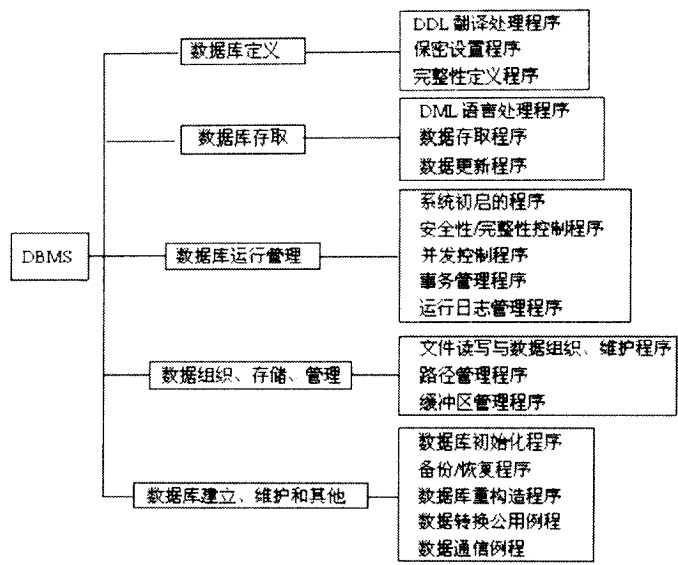


图 1-6 DBMS 的程序模块

DBMS 的这些组成模块相互联系，相互依赖，共同组成 DBMS 系统。

1.3.4 DBMS 与操作系统

DBMS 作为一个软件通常是运行在操作系统之上的。根据不同操作系统的特点，DBMS 可以利用操作系统的基本功能来实现 DBMS 的功能，一般用共享模块法、分离进程法和操作系统结合。

(1) 共享模块法：

此方法针对的是面向过程的操作系统。用户在使用过程中，根据需要把 DBMS 模块加入到自己的进程中。在 DBMS 与应用程序间预先建立链接后，用户进程按子程序调用 DBMS 模块，用户程序的并行服务借助操作系统调度完成。其图解如图 1-7 (a)所示，DBAP 是指数据库系统应用程序或用户，AP 指非数据库系统应用程序或用户。从用户角度看，每个用户进程对应一个虚拟 DBMS。

(2) 分离进程法：

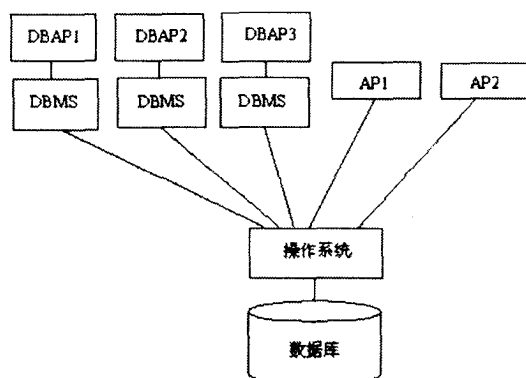
此方法针对的是面向消息的操作系统。用户进程和 DBMS 进程间是平行的关系，它们之间用消息进行交流。用户进程的请求通过消息发给 DBMS 进程，挂到 DBMS 进程的消息链上。DBMS 进程根据消息链完成用户进程的请求，把结果作为消息发回给用户进程。其关系如图 1-7 (b) 所示。

(3) 和操作系统结合：

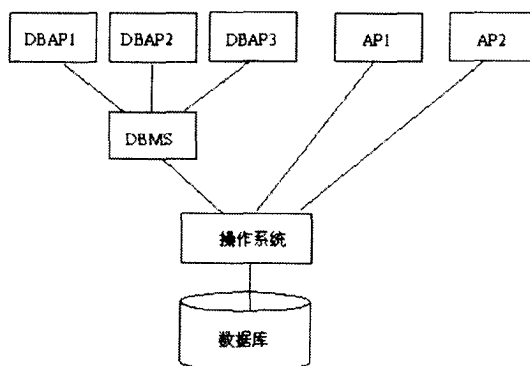
我们知道，操作系统是直接控制硬件工作的软件，它在调用系统硬件时具有最高的效率。一般的 DBMS 系统，都是作为应用软件工作在操作系统之上的，在需要使用硬件资源时，要先向操作系统发出请求，由操作系统分配硬件资源，这样必然会使效率的降低，

而且会在一定程度上降低系统的稳定性。为了让 DBMS 达到最高的运行效率，我们把 DBMS 与操作系统结合，使之成为效率高，运行稳定的数据库系统。如图 1-7 下 (c) 所示。

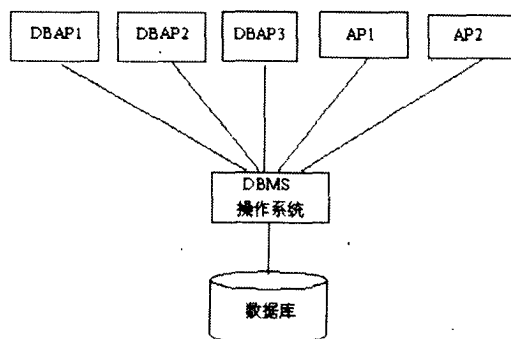
与操作系统结合的数据库系统的典型代表是 IBM AS/400 数据库服务器，对于 IBM AS/400 来说，DBMS 即是操作系统，操作系统即是 DBMS。它牺牲了兼容性，换来的是数据库系统运转的极高效率和远远优于其他数据库系统的稳定性。



(a)



(b)



(c)

图 1-7 DBMS与操作系统的三种关系

1.3.5 数据库应用程序

DBMS 中存储了大量的数据信息，其目的是为用户提供数据信息服务。而数据库应用程序则与 DBMS 进行通信，并访问 DBMS 中的数据。它是 DBMS 实现其对外提供数据信息服务这一目的的惟一途径。

简单地说，数据库应用程序是一个允许用户插入、修改、删除并报告数据库中的数据计算机程序。数据库应用程序在传统上是由程序员用一种或多种通用或专用的程序设计语言编写的。但是近年来出现了多种面向用户的数据库应用程序开发工具，这些工具可以简化使用 DBMS 的过程，并且不需要专门编程，例如 Delphi 等。

用来生成数据库应用程序的语言主要分为如下所述的三大类型。

1. 过程化语言

标准的计算机程序设计语言如 Pascal、Basic 和 C 都是过程化语言。这些语言可以通过某种“应用程序接口”(API)来创建数据库应用程序。这种 API 由一组标准的函数(或调用)组成，这些函数和调用则扩展了语言的功能，使之能访问数据库中的数据。当程序设计人员用过程化语言创建数据库应用时，必须把应用的代码编写成一系列的过程。每个过程执行应用的某一部分的工作。如一个过程查询数据库，而另一过程更新数据库中的数据，然后不同的过程通过其他的用户界面过程(例如菜单系统)联系在一起，并且在应用程序中的适当地方运行。

上述这些过程化语言一般用来创建非数据库应用程序，它们通常被称为“第三代语言”(3GL)。还有一些过程化程序设计语言是某种特定的 DBMS 专用的，这些语言一般被称为“第四代语言”(4GL)，即数据库专用语言。常见的数据库专用的过程化语言如 dBASE 语言，Paradox 数据库的 PAL 语言等等。

2. 结构化查询语言 (SQL)

结构化查询语言 (Structure Query Language) 是基于关系模型的数据库查询语言，它是一种非过程化的程序语言。也就是说，没有必要写出将如何做某事情，只需写出做到什么就可以了。写出的语句可看作是一个问题，称为“查询”(Query)，针对这个查询，得到所需的查询结果。下面是一个例子：

```
Select Name from Info where Age>30
```

这个查询意为从数据库表 Info 中将年龄 (Age) 大于 30 的所有人选出来，并列出他们的姓名 (Name)。

把 SQL 描述为子语言更适当一些，因为它没有任何屏幕处理或用户输入/输出的能力。它的主要目的是为了提供访问数据库的标准方法，而不管数据库应用的其余部分是用什么语言编写的。它既是为数据库的交互式查询而设计的(因此被称为动态 SQL)，也可在过程化语言编写的数据库应用程序中使用(因此被称为嵌入式 SQL)。

3. 其他语言

用于开发数据库应用程序的语言中，还可以使用目前非常常见的“面向对象程序设计”（OOP）语言，如 C++ 等。OOP 代表了一种完全不同的程序设计方法，在这种程序设计方法中，活动被定义为在“对象”上发生的操作，而不是作为一系列过程来定义的。在数据库应用程序中使用 OOP 语言的情况正在不断增加。

开发数据库应用程序使用的另一种语言是“宏”语言。宏语言不是一种完全的程序设计语言。它实际上是一个用户手工输入的表，这个表被输入到应用程序中，以便自动执行一定的任务。对于某个特定应用的高级语言，宏语言通常可以在低档 DBMS 软件中或数据库服务器的前端中找到。

最后，还有一种“Query-By-Example”（QBE，范例查询）语言。严格地讲 QBE 不是一种语言，它是面向用户提供了一个或多个空表的界面，这些空表对应于数据库中的表。用户可以通过键盘选择需要查询的列，并在适当的列中填入条件从而定义查询的检索条件，然后 DBMS 就把 QBE 转换成相应的动作，以完成用户要求的查询任务。

1.4 数据库系统的使用

本节主要介绍：

数据库系统的一般使用。

现行引用较广泛的关系数据库系统有 IBM DB2、MS SQL Server、Informix、Oracle、MySQL 等，它们都可以运行在多种操作系统上

不同数据库系统的安装和配置工作一般都比较复杂，尤其是 IBM DB2 或 Oracle 这样的大型数据库系统。这些活动多数由系统管理员去完成，不属于应用程序员的范畴，因此就不在此做介绍。我们要讲述的是在一个数据库系统配置完成后怎样去使用它。

首先，我们要进入 DBMS 所在的操作系统，如下就是通过 Telnet 方式登录 Linux 操作系统的情况：

```
Red Hat Linux release 6.1 (Cartman)
Kernel 2.2.12-20 on an i586
login: ProgMan
Password:

Last login: Sat Aug 19 07:30:29 from 202.31.44.82
No directory /home/ProgMan
Logging in with home = "/".
bash: /home/ProgMan/.bash_profile: Permission denied
[ProgMan@ nic/]$
```

进入操作系统后就可以执行相应的 DBMS 的执行文件，例如在 Linux 系统中执行

MySQL:

```
[ProgMan@ nic/]$ cd usr/bin
[ProgMan@ nic/]$ MySQL -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9450 to server version: 3.22.26a

Type 'help' for help.
```

MySQL >

这表明 MySQL 已经执行，但是还未链接任何数据库。

接下来，我们可以使用 DBMS 提供的一些指令。我们在上述环境中输入 “?;”，结果出现了 MySQL 的指令列表：

MySQL > ?;

MySQL commands:

```
help      (\h)    Display this text
?         (\h)    Synonym for `help`
clear     (\c)    Clear command
connect   (\r)    Reconnect to the server. Optional arguments are db and host
edit      (\e)    Edit command with $EDITOR
exit      (\q)    Exit MySQL. Same as quit
go        (\g)    Send command to MySQL server
ego       (\G)    Send command to MySQL server; Display result vertically
print     (\p)    Print current command
quit      (\q)    Quit MySQL
rehash    (\#)    Rebuild completion hash
status    (\s)    Get status information from the server
use       (\u)    Use another database. Takes database name as argument
```

Connection id : 12473 (Can be used with MySQLadmin kill)

MySQL>

我们要介绍这些指令之外的两条常用命令： Show 和 Desc。

Show 用于列出当前数据库系统中的数据库名称或当前数据库中的表名称，使用格式是 show databases 或 show tables，例如下面例子：

```
MySQL> use Information;
Database changed
MySQL> show databases;
```

Database
Information
Normal
Test
CS
HR
MK

6 rows in set (0.01 sec)

MySQL> use Information;

Database changed

MySQL> show tables;

Tables in Information
User

1 row in set (0.02 sec)

MySQL>

先列出的是该数据库系统中的所有数据库名称，可见共有 6 个数据库。然后使用 `use database name` 的命令进入了 Information 数据库，列出了该数据库中的表名。

现在就可以执行 SQL 命令了，例如：查询当前基本表 User 的全部数据。

MySQL> select * from User;

Num	Name	Sex	Age
1	Zhao	男	18

1 rows in set (0.03 sec)

Desc 用于列出基本表的描述，使用该命令可以查看所选表名的各属性列的数据类型等信息：格式：`Desc table_name, table_name` 参数，即表名。

此外，其他一些命令如 `\s` 是列出当前数据库系统的信息：

MySQL> \s

MySQL Ver 9.35 Distrib 3.22.26a, for pc-linux-gnu (i686)

```

Connection id:      87339
Current database:    Information
Current user:        root@localhost
Server version       3.22.26a
Protocol version     10
Connection           Localhost via UNIX socket
UNIX socket          /var/lib/MySQL/MySQL.sock
Uptime:              15 hours 44 min 24 sec

```

Threads:3 Questions:517732 Slow queries:0 Opens:157 Flush tables:1 Open tables:63

MySQL>

如果使用 quit，即退出了 MySQL 回到 Linux:

```
MySQL> quit
```

```
Bye
```

```
[ProgMan@ nic/]$
```

其他各种数据库系统的操作各有不同，详情可查阅各数据库系统的操作手册。

1.5 本章小结

1.5.1 主要内容

- 数据管理经历了人工管理阶段、文件管理阶段到了现在的数据库系统阶段。
- 数据库系统具有数据结构化、数据共享、可控冗余和数据独立的特点，因此得到了广泛的应用。
- 关系数据库是数据库系统中最重要的重要组成部分，现行的数据库系统多数是关系数据库系统或是支持关系模型的数据库系统。
- 数据库系统主要由三大部分组成：数据库管理系统（DBMS）、数据库应用程序（DBA）数据库（DB）。
- 结构化查询语言 SQL 是关系数据库系统的主要操作语言。

1.5.2 课后习题

- (1) 数据库管理技术经历了几个阶段，各有什么特点？
- (2) 对于下面的二维基本表，指出：
 - 1) 主码；
 - 2) 所有属性名；
 - 3) 各属性的域。

学 号	姓 名	系 别	年 龄
01	A	CS	20
02	B	CS	21
03	C	MA	19
04	D	CS	19
05	E	CI	21
06	F	MA	19
07	G	CI	19

- (3) 简述数据库系统的构成。
- (4) 什么是数据库系统的三级模式结构？模式间映像功能实现了数据库的什么特性？
- (5) 简述实际数据库系统的构成。
- (6) 什么是 DBMS？DBMS 的主要功能有哪些？

第 2 章 SQL 概 述

在了解了数据库系统和关系数据库的基本概念后，我们进一步对数据库的具体使用进行探讨。

在一个数据库系统已经完全建立以后，我们就可以利用现有的硬件资源做我们需要做的事，那就是数据的收集、整理、存储和利用。

对于目前应用最为广泛的关系数据库，最常用的方法就是用 SQL 语言进行操作。作为关系数据库系统之一的 SQL Server 7.0 也是如此。通过 SQL 语言，我们可以告诉数据库该做什么操作来满足我们的需求。

本章将介绍 SQL 语言的历史与现状以及 SQL 语言的分类，SQL 的操作符和库函数等内容，使读者建立一个对 SQL 语言的整体认识。

2.1 SQL 的历史

本节主要介绍：

SQL 语言的由来和发展。

SQL (Structure Query Language) 语言是关系数据库的核心语言。

在 70 年代初，E.E.Codd 首先提出了关系模型。70 年代中期，IBM 公司在研制 SYSTEM 关系数据库管理系统中研制了 SQL 语言。最早的 SQL 语言（叫 SEQUEL2）是在 1976 年 11 月的 IBM Journal of R&D 上公布的。1979 年 ORACLE 公司首先提供商用的 SQL，IBM 公司在 DB2 和 SQL/DS 数据库系统中也实现了 SQL。

1986 年 10 月，美国 ANSI (美国国家标准协会) 采用 SQL 作为关系数据库管理系统的标准语言 (ANSI X3.135-1986)，后为国际标准化组织 (ISO) 采纳为国际标准。1989 年，美国 ANSI 采纳在 ANSI X3.135-1989 报告中定义的关系数据库管理系统的 SQL 标准语言，称为 ANSI SQL 89，该标准替代 ANSI X3.135-1986 版本。该标准为下列组织所采纳：

- 国际标准化组织 (ISO)，为 ISO9075-1989 报告“Database Language SQL With Integrity Enhancement”；

- 美国联邦政府，发布在 The Federal Information Processing Standard Publication (FIPS PUB) 127。

目前，最新的 SQL 标准为 SQL92 标准。

SQL 成为国际标准后，数据库产品的各个厂家纷纷推出各自支持 SQL 的软件或 SQL 接口软件。这样就形成了一种趋势：各种类型的计算机和数据库系统都采用 SQL 作为其存取语言和标准接口，从而使数据库世界有可能链接为一个统一的整体。这个前景是意义十分重大的。

SQL 标准的影响超出了数据库领域。SQL 成为国际标准后，它在数据库以外的其他

领域也受到重视和采用。把 SQL 的数据检索功能和图形功能、软件工程工具、软件开发工具等结合的产品也越来越多。因此，在未来很长的一段时间里，SQL 将是关系数据库领域的主流语言。在软件工程、人工智能领域，SQL 也有相当大的潜力。

SQL 是一个标准的数据库语言，是面向集合的描述性非过程化语言。它功能强、效率高、简单易学易维护。然而 SQL 语言由于以上优点，同时也出现了这样一个问题：它是非过程性语言，即大多数语句都是独立执行的，与上下文无关。而绝大部分应用都是一个完整的过程，显然用 SQL 完全实现这些功能是很困难的。所以大多数数据库公司为了解决此问题，做了如下两方面的工作：

- (1) 扩充 SQL，在 SQL 中引入过程性结构；
- (2) 把 SQL 嵌入到高级语言中，以便一起完成一个完整的应用。

2.2 SQL 概貌及特点

本节主要介绍：

- 1) SQL 语言的特点；
- 2) SQL 语言的分类。

通过本节的学习，用户可以初步了解 SQL 语言。

2.2.1 SQL 概貌

SQL 是关系数据库的结构化查询语言，主要功能就是同各种数据库建立联系，进行沟通。按照 ANSI（美国国家标准协会）的规定，SQL 被作为关系型数据库管理系统的标准语言。SQL 语句可以用来执行各种各样的操作，例如，更新数据库中的数据，从数据库中提取数据等。目前，绝大多数流行的关系型数据库管理系统，如 Oracle、Sybase、Microsoft SQL Server、Access 等都采用了 SQL 语言标准。虽然很多数据库都对 SQL 语句进行了再开发和扩展，但是包括 Select、Insert、Update、Delete、Create 以及 Drop 在内的标准的 SQL 命令仍然可以被用来完成几乎所有的数据库操作。

SQL 是结构化语言，因此与现行的计算机语言——过程化语言有很大不同。过程化语言（Basic 和 C 等）通过某种 API（应用程序接口）实现对数据库的操作，必须把应用的代码编写成一系列的过程，每个过程执行应用某一部分的工作，然后不同的过程通过其他的用户界面联系在一起，并且在应用中的适当地方运行。而结构化语言 SQL 语言没有必要写出将如何做某事情，只需写出做到什么就可以。写出的语句可看作是一个问题，称为“查询”（Query），针对这个查询，得到所需的查询结果。

2.2.2 SQL 语言的特点

SQL 广泛地被采用正说明了它的优点。它使全部用户，包括应用程序员、DBA 管理员和终端用户获益匪浅。