第1章 数据库入门必学

内容导读

本章将对数据库系统的基本概念、数据库的体系结构、数据模型、数据库系统设计以及主流的关系型数据库进行介绍。通过本章的学习，读者可以全面了解数据库的基础知识，为后续章节的学习打下坚实基础。

学习目标

- 了解数据库技术的发展历史
- 熟练掌握数据库系统的组成
- 掌握数据模型的基本概念
- 熟练掌握关系数据模型的相关知识
- 掌握数据库设计的步骤和方法
- 了解常见的关系型数据库

课时安排

- 理论学习 4 课时

1.1 数据库系统简介

在系统介绍数据库知识之前，首先介绍数据库的一些基本概念。

1.1.1 数据管理技术的发展

数据是现实世界中实体（或客体）在计算机中的符号表示，数据不仅可以是数字，还可以是文字、图形、图像、音频和视频。现实生活中我们需要管理大量的数据，比如，教务相关的学生、教工、课程和成绩等方面的数据，医院有关病历、药品、医生、处方等方面的数据，银行有关存款、贷款、信用卡和投资理财业务等方面的数据。因此，对各种数据实现有效的管理具有重要意义。

自计算机产生以来，人类社会进入了信息化时代，数据处理的速度及规模远远超出了过去人工或机械方式的能力范围，计算机以其快速准确的计算能力和海量的数据存储能力在数据处理领域得到了广泛的应用。但是数据库技术并不是最早的数据管理技术，总的来说，数据管理的发展经历了人工管理、文件系统管理和数据库系统管理三个发展阶段。

1．人工管理阶段

20 世纪 50 年代中期以前，计算机主要用于科学计算。当时的外存只有纸带、卡片、磁带等，并没有磁盘等直接存取的存储设备。计算机系统没有操作系统，没有管理数据的软件，数据管理方式为人工管理。

人工管理数据具有如下特点。

（1）数据不被保存。

当时的计算机主要用于科学计算，一般不需要将数据长期保存，只是在计算某一课题时将数据输入，用完就撤走。

（2）应用程序管理数据。

数据需要由应用程序管理，没有相应的软件系统负责数据的管理工作。应用程序中不仅要规定数据的逻辑结构，而且要设计其物理结构，包括存储结构、存取方法、输入方式等，因此程序员的负担很重。

（3）数据不能共享。

数据是面向应用的，一组数据只能对应一个程序。当多个应用程序涉及某些相同的数据时，由于必须各自定义，无法互相利用、互相参照，因此程序与程序之间有大量的冗余数据。

（4）数据不具有独立性。

数据的逻辑结构或物理结构改变后，必须对应用程序做相应的修改，这就进一步加重了程序员的负担。

在人工管理阶段，程序与数据之间的一一对应关系如图 1-1 所示。

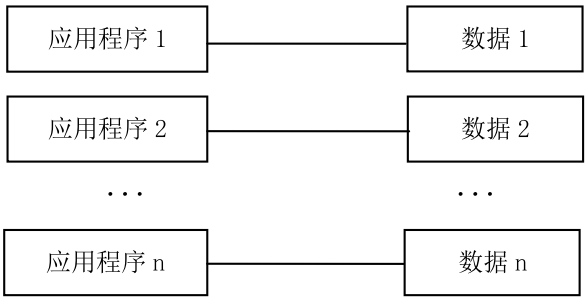


图 1-1　人工管理阶段程序和数据的关系

2．文件系统管理阶段

20 世纪 50 年代后期到 60 年代中期，已有磁盘、磁鼓等直接存储设备。在计算机系统方面，不同类型的操作系统的出现极大地增强了计算机系统的功能。操作系统中用来进行数据管理的是文件系统。这时可以把相关的数据组织成一个文件存放在计算机中，在需要的时候只要提供文件名，计算机就能从文件系统中找出所要的文件，把文件中存储的数据提供给用户处理。因为这时数据的组织仍然是面向程序，所以存在大量的数据冗余，无法有效地进行数据共享。

文件系统管理数据具有如下优点。

（1）数据可以长期保存。

数据可以组织成文件长期保存在计算机中反复使用。

（2）由文件系统管理数据。

文件系统把数据组织成内部有结构的记录，实现“按文件名访问，按记录进行存取”的管理技术。

文件系统使应用程序与数据之间有了初步的独立性，程序员不必过多地考虑数据存储的物理细节。例如，文件系统中可以有顺序结构文件、索引结构文件、Hash 等。数据在存储方式上的不同不会影响程序的处理逻辑。如果数据的存储结构发生改变，应用程序的改变很小，那么可节省程序的维护工作量。但是，文件系统仍存在以下缺点。

（1）数据共享性差，冗余度大。

在文件系统中，一个（或一组）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）使用部分相同的数据时，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，浪费了存储空间。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，文件的逻辑结构对该应用来说是优化的，因此要想对现有的数据增加一些新的应用会很困难，系统不容易扩充。一旦数据的逻辑结构发生改变，就必须修改应用程序、文件结构的定义。因此数据与程序之间仍缺乏独立性。文件系统管理阶段程序与数据之间的关系如图 1-2 所示。

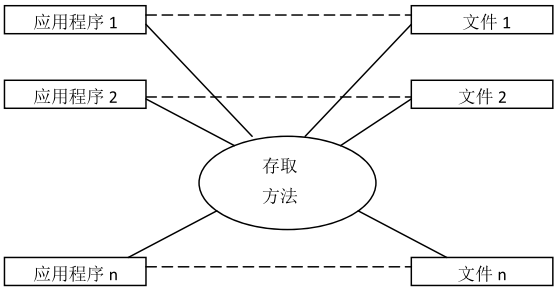


图 1-2　文件系统阶段程序和数据的关系

3．数据库系统管理阶段

20 世纪 60 年代后期，计算机用于管理的规模越来越大，应用越来越广泛，数据量急剧增长，同时多种应用、多种语言互相覆盖的共享数据集合的要求也越来越强烈。这时已有大容量磁盘，硬件价格下降，软件价格则上升，为编制和维护系统软件及应用程序所需的成本相对增加。在这种背景下，以文件系统作为数据管理手段已经不能满足应用的需求，于是为解决多用户、多应用共享数据的需求，使数据为尽可能多的应用服务，数据库技术应运而生，出现了统一管理数据的专用软件系统——数据库管理系统。

用数据库系统来管理数据比文件系统具有明显的优点，从文件系统到数据库系统，标志着数据管理技术的飞跃。

数据库系统具有如下特点。

（1）数据结构化。

数据库系统实现了整体数据的结构化，这是数据库最主要的特征之一。这里所说的“整体”结构化，是指在数据库中的数据不再仅针对某个应用，而是面向全组织；不仅数据内部是结构化，而且是整体式结构化，数据之间有联系。

（2）数据的共享性高，冗余度低，易扩充。

因为数据是面向整体的，所以数据可以被多个用户、多个应用程序共享使用，可以大大减少数据冗余，节约存储空间，避免数据之间的不相容性与不一致性。

（3）数据独立性高。

数据独立性包括数据的物理独立性和逻辑独立性。物理独立性是指数据在磁盘上的数据库中如何存储是由 DBMS（Database Management System）管理的，用户程序不需要了解，应用程序要处理的只是数据的逻辑结构，这样，当数据的物理存储结构改变时，用户的程序不用改变。逻辑独立性是指用户的应用程序与数据库的逻辑结构是相互独立的，也就是说，数据的逻辑结构改变了，用户程序可以不改变。

数据与程序的独立，把数据的定义从程序中分离出去，加上存取数据由 DBMS 负责提供，从而简化了应用程序的编制，大大减少了应用程序的维护和修改。

（4）数据由 DBMS 统一管理和控制。

数据库系统中的数据由 DBMS 进行统一的控制和管理，所有应用程序对数据的访问都要交给 DBMS 来完成。

DBMS 主要提供以下控制功能：

◎ 数据的安全性保护（security）；

◎ 数据的完整性检查（integrity）；

◎ 数据库的并发访问控制（concurrency）；

◎ 数据库的故障恢复（recovery）。

在数据库系统管理阶段程序与数据之间的对应关系如图 1-3 所示。

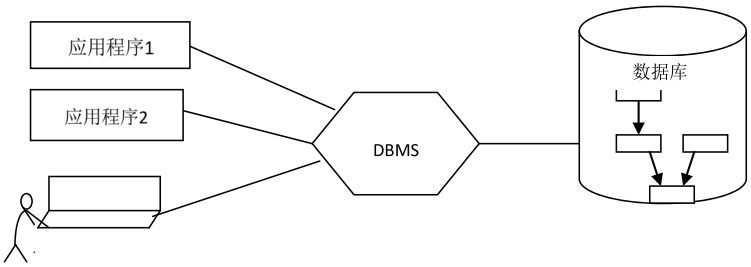


图 1-3　数据库系统阶段程序与数据的关系

上述三个阶段的比较如表 1-1 所示。

表 1-1　数据管理三个阶段的比较

背景与处理特点	人工管理阶段	文件系统管理阶段	数据库系统管理阶段
应用背景	科学计算	科学计算、管理	大规模管理
硬件背景	无直接存取设备	磁盘、磁鼓	大容量磁盘
软件背景	没有操作系统	有文件系统	有数据库管理系统
处理方式	批处理	联机实时处理、批处理	联机实时处理、分布处理、批处理
数据库的管理者	用户（程序员）	人工、文件系统	数据库管理系统
数据的共享程度	某一应用程序	某一应用	现实世界
数据面向的对象	无共享，冗余度极大	共享性差，冗余度大	共享性高，冗余度小
数据的独立性	不独立，完全依赖于程序	独立性差	具有高度的物理独立性和一定的逻辑独立性
数据的结构化	无结构	记录内有结构、整体无结构	整体结构化，用数据模型描述
数据控制能力	应用程序控制	应用程序控制	由数据库管理系统提供数据安全性、完整性、并发控制和恢复能力

知识拓展

数据库管理技术的发展过程实际上也是应用程序和数据逐步分离的过程，人工管理阶段程序和数据不分家，而在数据库系统管理阶段程序和数据具有了高度的独立性。

1.1.2 数据库系统的组成

数据库系统，（Database System，DBS）是指在计算机系统中引入数据库后的系统，一般由数据库、数据库管理系统（及其开发工具）、应用系统和数据库管理员构成。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 是远远不够的，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）。

在一般不引起混淆的情况下，人们常常把数据库系统简称为数据库。数据库系统的组成如图 1-4 所示。数据库系统在计算机系统中的地位如图 1-5 所示。

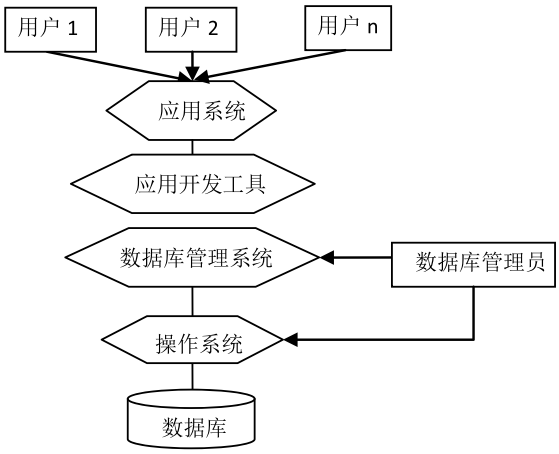


图 1-4　数据库系统的组成

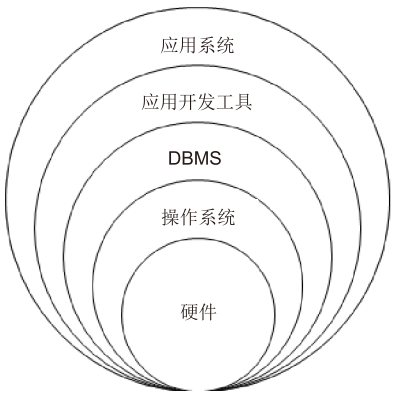


图 1-5　数据库在计算机系统中的地位

下面介绍数据库系统的组成。

1．硬件

硬件主要指计算机的各个组成部分。鉴于数据库应用系统的需求，特别要求数据库主机或数据库服务器外存要足够大，I/O 存取效率要高，主机的吞吐量大、作业处理能力强。对分布式数据库而言，计算机网络也是基础环境。具体指：

（1）要有足够大的内存，存放操作系统和 DBMS 的核心模块、数据库缓冲区和应用程序。

（2）有足够大的磁盘等直接存取设备存放数据库，有足够的光盘、磁盘、磁带等作为数据备份介质。

（3）要求连接系统的网络有较高的数据传输速率。

（4）有较强处理能力的中央处理器（CPU）来保证数据处理的速度。

2. 数据库

过去人们把数据存放在文件柜里，当数据越来越多时从大量的文件中查找数据显得十分困难。现在人们借助计算机和数据库科学地保存和管理大量复杂的数据，能方便而充分地利用这些宝贵的信息资源。

数据库（Database），顾名思义，就是存放数据的仓库。只不过这个仓库是在计算机的存储设备上，而且数据是按照一定的数据模型组织并存放在外存上的一组相关数据集合，通常这些数据是面向一个组织、企业或部门的。例如，学生成绩管理系统中，学生的基本信息、课程信息、成绩信息等都是来自学生成绩管理数据库的。

严格地讲，数据库是长期存储在计算机内，有组织的、大量的、可共享的数据集合。数据库中的数据按一定的数据模型组织、描述和存储，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。简单来说，数据库中的数据具有永久存储、有组织和可共享三个基本特点。

3．软件

数据库系统的相关软件主要包括：

（1）DBMS。

在建立数据库之后，下一个问题就是如何科学地组织和存储数据，如何高效地获取和维护数据，完成这个任务的就是系统软件——数据库管理系统（Database Management System，DBMS）。DBMS 是指数据库系统中对数据进行管理的软件系统，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。

如果用户要对数据库进行操作，是由 DBMS 把操作从应用程序带到外部级、概念级，再导向内部级，进而操纵存储器中的数据。一个 DBMS 的主要目标是使数据作为一种可管理的资源来处理。DBMS 应使数据易于为各种不同的用户所共享，应该增进数据的安全性、完整性及可用性，并提供高度的数据独立性。

DBMS 的主要功能如下：

◎ 数据的定义功能；

◎ 数据的操纵功能；

◎ 数据的控制功能；

◎ 其他功能。

（2）支持 DBMS 运行的操作系统。

（3）与数据库通信的高级程序语言及编译系统。

（4）为特定应运环境开发的数据库应用系统。

4．数据库管理员及相关人员

数据库有关人员包括数据库管理员、系统分析员和数据库设计人员、应用程序员以及普通用户等。

（1）数据库管理员。

数据库管理员（Database Administrator，DBA）负责管理和监控数据库系统，为用户解决应用中出现的系统问题。为了保证数据库能够高效正常地运行，大型数据库系统都设有专人负责数据库系统的管理和维护。其主要职责如下：

① 决定数据库中的信息内容和结构。数据库中要存放哪些信息，DBA 要参与决策。因此 DBA 必须参加数据库设计的全过程，并与用户、应用程序员、系统分析员密切合作共同协商，做好数据库设计工作。

② 决定数据库的存储结构和存取策略。

③ 监控数据库的运行（系统运行是否正常，系统效率如何），及时处理数据库系统运行过程中出现的问题。比如系统发生故障时，数据库会因此遭到破坏，DBA 必须在最短的时间内把数据库恢复到正确状态。

④ 安全性管理，即通过对系统的权限设置、完整性控制设置来保证系统的安全性。DBA 要负责确定各个用户对数据库的存取权限、数据的保密级别和完整性约束条件。

⑤ 日常维护，如定期对数据库中的数据进行备份，维护日志文件等。

⑥ 对数据库有关文档进行管理。

数据库管理员在数据库管理系统的正常运行中起着非常重要的作用。

（2）系统分析员和数据库设计人员。

系统分析员和数据库设计人员负责应用系统的需求分析和规范说明，和用户及 DBA 交流、合作，确定系统的硬件、软件配置，并参与数据库系统概要设计。

（3）应用程序员。

应用程序员是负责设计、开发应用系统功能模块的软件编程人员，他们根据数据库结构编写特定的应用程序，并进行调试和安装。

（4）普通用户。

这里的用户是指最终用户。最终用户通过应用程序的用户接口使用数据库。

1.2 数据库的体系结构

考察数据库系统的结构可以从多种不同的层次或不同的角度，主要分为内部结构和外部结构。内部结构是从数据库管理系统角度看，数据库系统通常采用三级模式结构；外部结构是从数据库最终用户角度看，数据库系统分为单用户结构、主从式结构、客户端 / 服务器结构（C/S 结构）、浏览器 / 服务器结构（B/S 结构）和分布式结构。

本节分别从以上两个方面介绍数据库的系统结构。

1.2.1 数据库系统的内部结构

虽然实际的数据库系统软件产品种类很多，它们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，但从数据库管理系统的角度看，它们的体系结构都具有相同的特征，即采用三级模式结构。

1．数据库系统的三级模式结构

数据库系统的三级模式结构是指数据库系统是由概念模式、外模式和内模式三级构成，如图 1-6 所示。

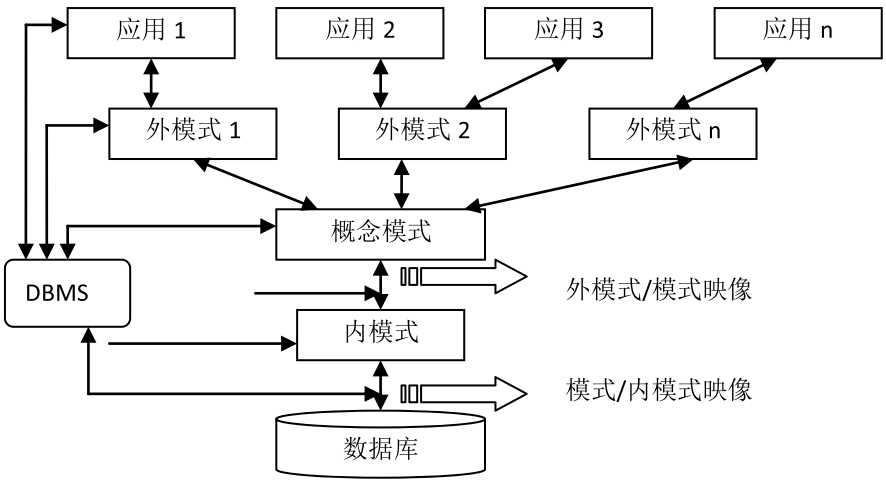


图 1-6　数据库系统的三级模式结构

（1）概念模式。

概念模式（Schema）也称为逻辑模式，是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共视图，它仅仅涉及型（Type）的描述，不涉及具体的值（Value）。逻辑模式的定义中主要包含数据的逻辑结构（数据项的名字、类型、取值范围等）、数据之间的联系以及数据有关的安全性要求等方面。一个数据库只有一个模式。

（2）外模式。

外模式也称子模式（Subschema）或用户模式，它是对数据库用户（包括应用程序员和最终用户）能够看见和使用的局部数据的逻辑结构和特征的描述，是数据库用户的数据视图，也是与某一应用有关的数据的逻辑表示。

外模式通常是模式的子集。一个模式可以有多个外模式。由于它是各个用户的数据视图，如果不同的用户在应用需求、看待数据的方式、对数据保密的要求等方面存在差异，则其外模式描述就可能不同。即使是模式中的同一数据，在外模式中的结构、类型、长度、保密级别等都可以不同。另外，同一外模式也可以为某一用户的多个应用系统所使用，但一个应用程序只能使用一个外模式。

外模式是保证数据库安全性的一个有力措施。每个用户只能看到和访问所对应的外模式中的数据，数据库中的其他数据是看不到的。

设立外模式的好处如下：

◎ 方便用户使用，简化了用户接口。用户只要依照模式编写应用程序或在终端输入命令，无须了解数据的存储结构。

◎ 保证数据的独立性。由于在三级模式之间存在两级映像，使得物理模式和概念模式的变化都反映不到子模式一层，从而不用修改应用程序，提高了数据的独立性。

◎ 有利于数据共享。从同一模式产生不同的子模式，减少了数据的冗余度，有利于为多种应用服务。

◎ 有别于数据的安全和保密。用户程序只能操作其子模式范围内的数据，从而与数据库中的其余数据隔离开来，缩小了程序错误传播的范围，保证了其他数据的安全。

（3）内模式。

内模式（Internal Schema）也称存储模式（Storage Schema），一个模式只有一个内模式。它是数据物理结构和存储方式的描述，它定义所有的内部记录类型、索引和文件的组织形式，以及数据控制方面的细节。

内部记录并不涉及物理记录，也不涉及设备的约束。比内模式更接近于物理存储和访问的那些软件机制是操作系统的一部分（即文件系统），例如从磁盘读数据或写数据到磁盘上的操作等。

2．数据库系统的二级映像和数据独立性

为了让用户不必考虑存取路径等细节，同时减少了应用程序的维护和修改工作，需要保证程序和数据之间的独立性，即当数据改变时程序不需要改变，反之，程序改变时数据也不需要改变。为了使程序和数据之间具有一定的独立性，DBMS 提供了两层映像：外模式 / 模式映像和模式 / 内模式映像。

映像实质上是一种对应关系，是指映像双方如何进行数据转换，并定义转换规则。有了这两层映像，用户在处理数据时不必关心数据在计算机中的具体表示方式与存储方式。正是这两层映像保证了数据库系统中的数据能够具有较高的逻辑独立性和物理独立性。

（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对应关系。如果模式需要进行修改，例如数据重新定义，增加新的关系、新的属性、改变属性的数据类型等，那么只需对各个外模式 / 模式的映像做相应的修改，使外模式尽量保持不变，而应用程序一般是依据外模式编写的，因此应用程序也不必修改，从而保证了数据与程序的逻辑独立性，这就是数据的逻辑独立性。

（2）模式 / 内模式映像。

模式 / 内模式映像定义了模式和内模式之间的对应关系，即数据全局逻辑结构与存储结构之间的对应关系。模式 / 内模式映像一般是在模式中描述的。当数据库的存储结构改变时（例如采用了另外一种存储结构），由数据库管理员对模式 / 内模式映像做相应改变，可以使模式保持不变，因此应用程序也不必改变。这就保证了数据与程序的物理独立性，简称数据的物理独立性。

1.2.2 数据库系统的外部结构

从数据库管理系统的角度看，数据库系统是一个三级模式结构，但数据库的这种模式结构对最终用户和程序员是透明的，他们见到的仅是数据库的外模式和应用程序。从最终用户角度来看，数据库系统的结构分为单用户结构、主从式结构、分布式结构、客户端 / 服务器结构（C/S 结构）、浏览器 / 服务器结构（B/S 结构）。

1．单用户数据库系统

单用户数据库系统是早期最简单的数据库系统。在单用户系统中，整个数据库系统，包括应用程序、DBMS、数据等都安装在一台计算机上，由一个用户独占，不同的机器间不能共享数据，如图 1-7 所示。

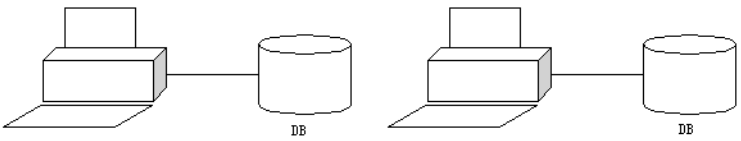


图 1-7　单用户数据库系统

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，各个部门的机器是独立的。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．主从式结构的数据库系统

主从式结构是指一个主机带多个终端的多用户结构。在这种结构中，数据库系统，包括应用程序、DBMS、数据等集中存放在主机上，所有任务都由主机完成，各个用户通过主机的终端并发地存取数据库，共享数据资源，如图 1-8 所示。

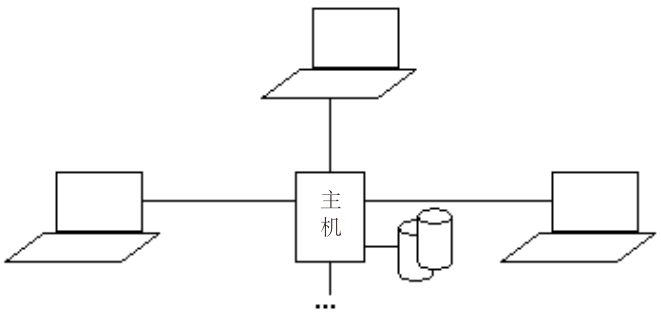


图 1-8　主从式数据库系统

主从式结构的优点是结构简单，数据易于维护和管理。缺点是当终端用户增加到一定程度后，主机的任务过于繁重，成为瓶颈，从而使系统性能大幅度下降。另外，当主机出现故障后，整个系统不能使用，因而系统的可靠性不高。

3．分布式结构的数据库系统

分布式结构的数据库系统是指数据库中的数据在逻辑上是一个整体，但物理分布在计算机网络的不同节点上，如图 1-9 所示。网络的每一个节点都可以独立处理本地数据库中的数据，执行局部应用；也可以同时存取和处理多个异地数据库中的数据，执行全局应用。

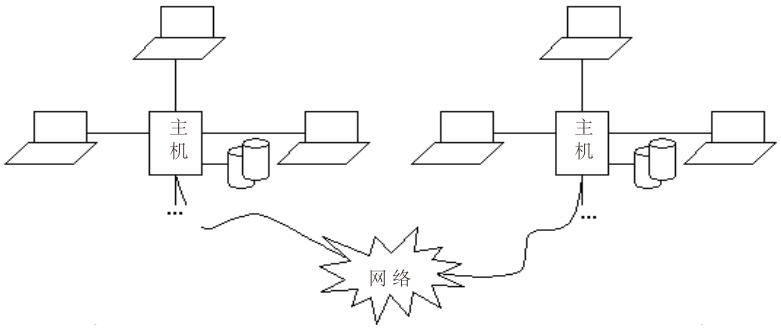


图 1-9　分布式数据库系统

分布式结构的数据库系统是计算机网络发展的必然产物，它适应了地理上分散的公司、团体和组织对于数据库应用的需求。但数据的分布存放给数据的管理、维护带来困难。此外，当用户需要经常访问远程数据时，系统效率明显地受网络速率的制约。

4．客户端 / 服务器结构的数据库系统

主从式数据库系统中的主机和分布式数据库系统中的每个节点均是通用计算机，既执行 DBMS 功能，又执行应用程序。随着工作站功能的增强和广泛使用，人们开始把 DBMS 功能和应用分开。网络中某些节点上的计算机专门执行 DBMS 功能，称为数据库服务器，简称服务器，其他节点上的计算机安装 DBMS 外围应用开发工具，支持用户的应用，称为客户机，这就是客户端 / 服务器（Client/Server）结构，简称 C/S 结构，如图 1-10 所示。

在客户端 / 服务器结构中，客户端的用户请求被传送到数据库服务器，数据库服务器进行处理后，只将结果返回给用户 (而不是整个数据)，从而显著减少了网络数据的传输量，提高系统的性能、吞吐量和负载能力。

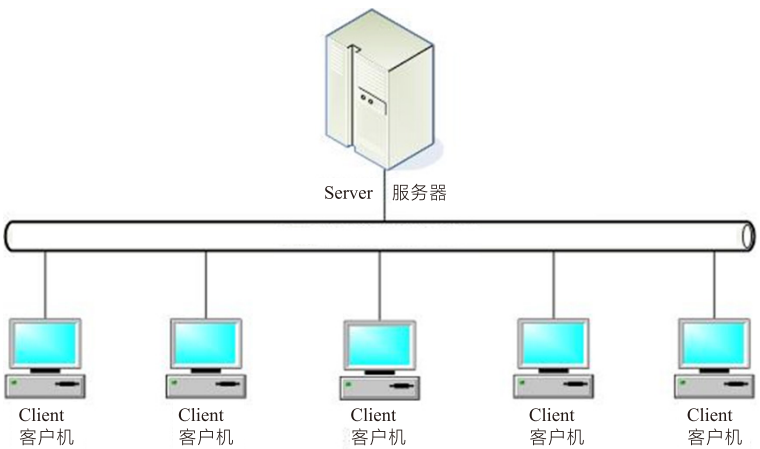


图 1-10　客户端 / 服务器结构

5．浏览器 / 服务器结构的数据库系统

随着互联网的飞速发展，移动办公和分布式办公越来越普及，客户端 / 服务器（C/S）结构的缺点就逐渐暴露出来，特别是客户端需要安装专用的客户端软件，一旦客户端软件

升级，那么所有的客户计算机上的客户端软件均需要更新，因此需要对 C/S 结构进行改进，浏览器 / 服务器（Browser/Server）结构应运而生，简称 B/S 结构，如图 1-11 所示。

在浏览器 / 服务器结构下，用户工作界面是通过浏览器来实现，极少部分事务逻辑在前端 (Browser) 实现，主要事务逻辑在服务器端 (Server) 实现。这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。客户机上只要安装一个浏览器（Browser），浏览器通过 Web Server 同数据库进行数据交互。这样就大大简化了客户端电脑载荷，减轻了系统维护与升级的成本和工作量，降低了用户的总体成本。

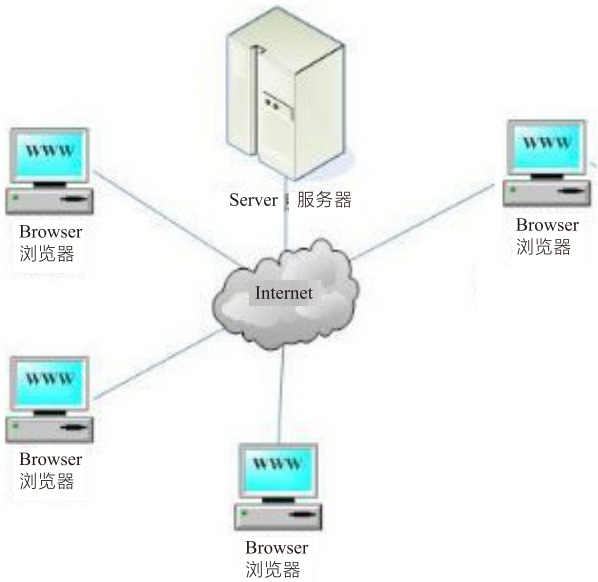


图 1-11　浏览器 / 服务器结构

知识拓展

数据库系统的结构有很多，目前主流的数据库系统结构是 C/S 结构和 B/S 结构，而且很多系统都是二者相结合的。

1.3　数据模型

模型，是现实世界特征的模拟与抽象。比如一组建筑规划沙盘、精致逼真的飞机航模，都是对现实生活中的事物的描述和抽象，见到它就会让人们联想到现实世界中的实物。

1.3.1 数据模型的概念

数据模型（data model）也是一种模型，它是现实世界数据特征的抽象。由于计算机不可能直接处理现实世界中的具体事物，因此人们必须事先把具体事物转换成计算机能够处理的数据，即首先要数字化，把现实世界中的人、事、物、概念用数据模型这个工具来抽象、表示和加工处理。数据模型是数据库中用来对现实世界进行抽象的工具，是数据库中用于提供信息表示和操作手段的形式构架，是现实世界的一种抽象模型。

数据模型按不同的应用层次分为三种类型，分别是概念数据模型（conceptual data model）、逻辑数据模型（logic data model）和物理数据模型（physical data model）。

概念数据模型又称概念模型，是一种面向客观世界、面向用户的模型，与具体的数据库管理系统无关，与具体的计算机平台无关。人们通常先将现实世界中的事物抽象到信息世界，建立所谓的“概念模型”，然后再将信息世界的模型映射到机器世界，将概念模型转换为计算机世界中的模型。因此，概念模型是从现实世界到机器世界的一个中间层次。

逻辑数据模型又称逻辑模型，是一种面向数据库系统的模型，它是概念模型到计算机之间的中间层次。概念模型只有在转换成逻辑模型之后才能在数据库中得以表示。目前，逻辑模型的种类很多，其中比较成熟的有层次模型、网状模型、关系模型、面向对象模型等。

这四种数据模型的根本区别在于数据结构不同，即数据之间联系的表示方式不同。

（1）层次模型用“树结构”来表示数据之间的联系。

（2）网状模型用“图结构”来表示数据之间的联系。

（3）关系模型用“二维表”来表示数据之间的联系。

（4）面向对象模型用“对象”来表示数据之间的联系。

物理数据模型又称物理模型，它是一种面向计算机物理表示的模型，此模型是数据模型在计算机上的物理结构表示。

数据模型通常由三部分组成，分别是数据结构、数据操纵和完整性约束，也称为数据模型的三大要素。概念数据模型非常多，在这里我们只介绍最经典的模型 E-R 模型。同样，逻辑数据模型包括层次模型、网状模型和关系模型，在这里我们只介绍目前主流的逻辑数据模型——关系模型。

1.3.2 ER 模型

概念模型中最著名的是实体联系模型（Entity Relationship Model，E-R 模型）。E-R 模型是 P．P．Chen 于 1976 年提出的。这个模型直接从现实世界中抽象出实体类型及实体间联系，然后用实体联系图（E-R 图）表示数据模型。设计 E-R 图的方法称为 E-R 方法。E-R 图是设计概念模型的有力工具。下面先介绍有关的名词术语及 E-R 图。

1. 实体（Entity）

现实世界中客观存在并可相互区分的事物叫作实体。实体可以是一个具体的人或物，如王伟、汽车等，也可以是抽象的事件或概念，如购买一本图书等。

2. 属性（Attribute）

实体的某一特性称为属性。如学生实体有学号、姓名、年龄、性别、院系等方面的属性。属性有“型”和“值”之分，“型”即为属性名，如姓名、年龄、性别是属性的型；“值”即为属性的具体内容，如（990001, 张立 ,20, 男 , 计算机）这些属性值的集合表示一个学生实体。

3. 实体型（Entity Type）

若干属性型组成的集合可以表示一个实体的类型，简称实体型，如学生（学号 , 姓名 , 年龄 , 性别 , 院系）就是一个实体型。

4. 实体集（Entity Set）

同型实体的集合称为实体集，如所有的学生、所有的课程等。

5. 码（Key）

能唯一标识一个实体的属性或属性集称为实体的码，如学生的学号。学生的姓名可能有重名，不能作为学生实体的码。

6. 域（Domain）

属性值的取值范围称为该属性的域，如学号的域为 6 位整数，姓名的域为字符串集合，年龄的域为小于 40 的整数，性别的域为（男 , 女）。

7. 联系（Relationship）

在现实世界中，事物内部以及事物之间是有联系的，这些联系同样要抽象和反映到信息世界中来。在信息世界中将被抽象为实体型内部的联系和实体型之间的联系。

实体内部的联系通常是指组成实体的各属性之间的联系；实体之间的联系通常是指不同实体集之间的联系。

两个实体型之间的联系有如下三种类型。

（1）一对一联系（1 ： 1）。

实体集 A 中的一个实体至多与实体集 B 中的一个实体相对应，反之亦然，则称实体集 A 与实体集 B 为一对一的联系，记作 1 ： 1。例如：班级与班长，观众与座位，病人与床位。

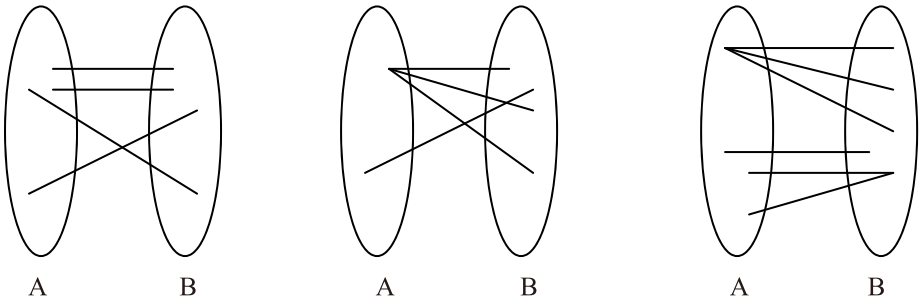
（2）一对多联系（1 ： n）。

实体集 A 中的一个实体与实体集 B 中的多个实体相对应，反之，实体集 B 中的一个实体至多与实体集 A 中的一个实体相对应，记作 1 ： n。如班级与学生、公司与职员、省与市。

（3）多对多（m ： n）。

实体集 A 中的一个实体与实体集 B 中的多个实体相对应，反之，实体集 B 中的一个实体与实体集 A 中的多个实体相对应，记作（m ： n）。例如 ：教师与学生 , 学生与课程，工厂与产品。

实际上，一对一联系是一对多联系的特例，而一对多联系又是多对多联系的特例。可以用图形来表示两个实体型之间的这三类联系，如图 1-12 所示。



（a） 1：1 联系　　（b） 1：n 联系　　（c） m：n 联系

图 1-12　三种联系示意图

在 E-R 图中有下面 4 个基本成分：

① 矩形框，表示实体类型（研究问题的对象）。

② 菱形框，表示联系类型（实体间的联系）。

③ 椭圆形框，表示实体类型和联系类型的属性。

相应的命名均记入各种框中。对于实体标识符的属性，在属性名下面画一条横线。

④ 直线，联系类型与其涉及的实体类型之间以直线连接，用来表示它们之间的联系，并在直线端部标注联系的种类（1 ： 1、1 ： n 或 m ： n）。

E-R 模型有两个明显的优点：一是接近于人的思维，容易理解；二是与计算机无关，用户容易接受。因此 E-R 模型已成为软件工程中的一个重要设计方法。但是 E-R 模型只能说明实体间语义的联系，不能进一步说明详细的数据结构。一般遇到一个实际问题，总是先设计一个 E-R 模型，然后再把 E-R 模型转换成计算机已实现的数据模型。

1.3.3　关系模型

目前，数据库领域中常用的逻辑数据模型有四种：

◎ 层次模型（Hierarchical Model）；

◎ 网状模型（Network Model）；

◎ 关系模型（Relational Model）；

◎ 面向对象模型（Object Oriented Model）。

其中，层次模型和网状模型统称为非关系模型。非关系模型的数据库系统在 20 世纪 70 年代至 80 年代初非常流行，在数据库系统产品中占据了主导地位，现在已逐渐被关系模型的数据库系统取代，但在美国等一些国家里，由于早期开发的应用系统都是基于层次数据库或网状数据库系统，因此目前仍有不少层次数据库或网状数据库系统在继续使用。

1970 年美国 IBM 公司 San Jose 研究室的研究员 E.F.Codd 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理论基础。20 世纪 80 年代以来，计算机厂商新推出的数据库管理系统几乎都支持关系模型，非关系系统的产品也大都加上了对关系模型接口。数据库领域当前的研究工作也都是以关系方法为基础。

面向对象的方法和技术在计算机各个领域，包括程序设计语言、软件工程、信息系统设计、计算机硬件设计等方面都产生了深远的影响，也促进了数据库中面向对象数据模型的研究和发展。

关系模型是目前最重要的一种数据模型。关系数据库系统采用关系模型作为数据的组织方式，下面主要介绍关系模型。关系数据库系统与非关系数据库系统的区别是，关系系统只有“表”这一种数据结构，而非关系数据库系统还有其他数据结构，对这些数据结构还有其他操作。

1．关系模型的基本术语

在关系模型中，用单一的二维表结构来表示实体及实体间的关系。

（1）关系（Relationship）。一个关系对应一个二维表，二维表名就是关系名。

（2）关系模式（Relationship Schema）。二维表中的行定义（表头）、记录的类型，即对关系的描述称为关系模式，关系模式的一般形式为：

```
关系名（属性 1, 属性 2,…, 属性 n）
```

例如学生关系模式为：

```
学生（学号 , 姓名 , 年龄 , 性别 , 院系）
```

（3）属性（Attribute）及值域（Domain）。二维表中的列（字段）称为关系的属性。属性的个数称为关系的元数，又称为度。度为 n 的关系称为 n 元关系，度为 1 的关系称为一元关系，度为 2 的关系称为二元关系。关系的属性包括属性名和属性值两部分，其列名即为属性名，列值即为属性值。属性值的取值范围称为值域，每一个属性对应一个值域，不同属性的值域可以相同。

（4）元组（Tuple）。二维表中的一行，即每一条记录的值称为关系的一个元组。其中，每一个属性的值称为元组的分量。关系由关系模式和元组的集合组成。

（5）键（Key），也称为码，由一个或多个属性组成。在实际使用中，有下列几种键。

① 候选键（Candidate Key）：若关系中的某一属性组的值能唯一标识一个元组，则称该属性组为候选键。

② 主键（Primary Key）：若一个关系有多个候选键，则选定其中一个为主键。主键中包含的属性称为主属性，不包含在任何候选键中的属性称为非键属性（Non-Key attribute）。关系模型的所有属性组是这个关系模式的候选键，称为全键（All-key）。

③ 外键（Foreign Key）：设 F 是关系 R 的一个或一组属性，但不是关系 R 的键。如果 F 与关系 S 的主键相对应，则称 F 是关系 R 的外键，关系 R 称为参照关系，关系 S 称为被参照关系。

（6）主属性与非主属性。关系中包含在任何一个候选键中的属性称为主属性，不包含在任何一个候选键中的属性称为非主属性。

2．关系的性质

我们用集合的观点来定义关系。也就是说，把关系看成一个集合，集合中的元素是元组，每个元组的属性个数均相同。如果一个关系的元组个数是无限的，称为无限关系；反之，

称为有限关系。

在关系模型中对关系做了一些规范性的限制，可通过二维表格形象地理解关系的性质。

性质 1：关系中每个属性值都是不可分解的，即关系的每个元组分量必须是原子的。从二维表的角度讲，不允许表中嵌套表。表 1-2 就出现了这种表中再嵌套表的情况，在“成绩”下嵌套“平时”和“期末”两项。虽然类似的表在实际生活中司空见惯，但却不符合关系的基本定义。因为关系是从域出发定义的，每个元组分量都是不可再分的，不可能出现表中套表的现象。遇到这种情况，可对表格进行简单的等价变换，使之成为符合规范的关系。例如，可把表 1-2 改成表 1-3。这里把“成绩”分成两列——“平时成绩”和“期末成绩”，两个属性都取自同一个域“成绩”。

表 1-2　不符合规范的成绩表

学　号	姓　名	成　绩	
		平时	期末
201804001	陈家乐	90	88
201804002	王静茹	98	95

表 1-3　符合规范的成绩表

学　号	姓　名	平时成绩	期末成绩
201804001	陈家乐	90	88
201804002	王静茹	98	95

性质 2：关系中不允许出现相同的元组。从语义角度看，二维表中的一行即一个元组，代表着一个实体。现实生活中不可能出现完全一样、无法区分的两个实体，因此，二维表不允许出现相同的两行。同一关系中不能有两个相同的元组存在，否则将使关系中的元组失去唯一性，这一性质在关系模型中很重要。

性质 3：在定义一个关系模式时，可随意指定属性的排列次序，因为交换属性顺序的先后，并不改变关系的实际意义。例如，在定义表 1-3 所示的关系模式时，可以指定属性的次序为（学号 , 姓名 , 平时成绩 , 期末成绩），也可以指定属性的次序为（学号 , 姓名 , 期末成绩 , 平时成绩）。

性质 4：在一个关系中，元组的排列次序可任意交换，并不改变关系的实际意义。由于关系是一个集合，因此不考虑元组间的顺序问题。在实际应用中，常常对关系中的元组排序，这样做仅仅是为了加快检索数据的速度，提高数据处理的效率。

对于性质 3 和性质 4，需要再补充一点。判断两个关系是否相等，是从集合的角度来考虑的，与属性的次序无关，与元组次序无关，与关系的命名也无关。如果两个关系仅仅是上述差别，在其余各方面完全相同，就认为这两个关系相等。

性质 5：关系模式相对稳定，关系却随着时间的推移不断变化。这是由数据库的更新操作（包括插入、删除、修改）引起的。

3．关系的完整性。

关系模型的完整性规则是对关系的某种约束条件。关系模型中可以有三类完整性约束：实体完整性、参照完整性和用户定义的完整性。

（1）实体完整性（Entity Integrity）。

一个基本关系通常对应现实世界的一个实体集，如银行关系对应于银行的集合。现实世界中的实体是可区分的，即它们具有某种唯一性标识。相应地，关系模型中以主键作为唯一性标识。主键的属性即主属性，不能取空值。所谓空值就是“不知道”或“无意义”的值。如果主属性取空值，就说明存在某个不可标识的实体，即存在不可区分的实体，这与现实世界的应用环境相矛盾，因此这个实体一定不是一个完整的实体。

实体完整性规则：若属性 *A* 是基本关系 *R* 的主属性，则属性 *A* 不能取空值。

（2）参照完整性（Referential integrity）。

现实世界中的实体之间往往存在某种联系，在关系模型中实体及实体间的联系都是用关系来描述的。这样就自然存在着关系与关系间的引用。

设 F 是基本关系 R 的一个或一组属性，但不是关系 R 的键，如果 F 与基本关系 S 的主键 Ks 相对应，则称 F 是基本关系 R 的外键（Foreign Key），并称基本关系 R 为参照关系（Referencing Relation），基本关系 S 为被参照关系（Referenced Relation）或目标关系（Target Relation）。关系 R 和 S 不一定是不同的关系。

参照完整性规则就是定义外码与主码之间的引用规则。

参照完整性规则：若属性（或属性组）F 是基本关系 R 的外键，它与基本关系 S 的主键 Ks 相对应（基本关系 R 和 S 不一定是不同的关系），则对于 R 中每个元组在 F 上的值必须为：

◎ 或者取空值（F 的每个属性值均为空值）；

◎ 或者等于 S 中某个元组的主键值。

（3）用户定义的完整性（User-defined Integrity）。

实体完整性和参照完整性均适用于任何关系数据库系统。除此之外，不同的关系数据库系统根据其应用环境的不同，往往还需要一些特殊的约束条件。

用户定义的完整性就是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。关系模型应提供定义和检验这类完整性的机制，以便用统一的系统的方法处理，而不是由应用程序承担这一功能。

1.4 关系数据库设计

有人说：一个成功的管理信息系统，是由 50% 的业务 +50% 的软件所组成，而成功软

件所占的 50% 又由 25% 的数据库 +25% 的程序所组成。笔者认为非常有道理。要开发管理信息系统，数据库设计的好坏是关键。

数据库设计是指在给定的环境下，创建一个性能良好，能满足不同用户使用要求，又能被选定的 DBMS 所接受的数据模式。

从本质上讲，数据库设计乃是将数据库系统与现实世界相结合的一种过程。

人们总是力求设计出一个好用的数据库系统。设计数据库时既要考虑数据库的框架和数据结构，又要考虑应用程序存取数据和处理数据的效率。因此，最佳设计不可能一蹴而就，只能是一个反复探寻的过程。

大体上可以把数据库设计划分成以下几个阶段：需求分析、概念结构设计、逻辑结构设计、数据库物理结构设计、数据库实施、数据库运行和维护，如图 1-13 所示。

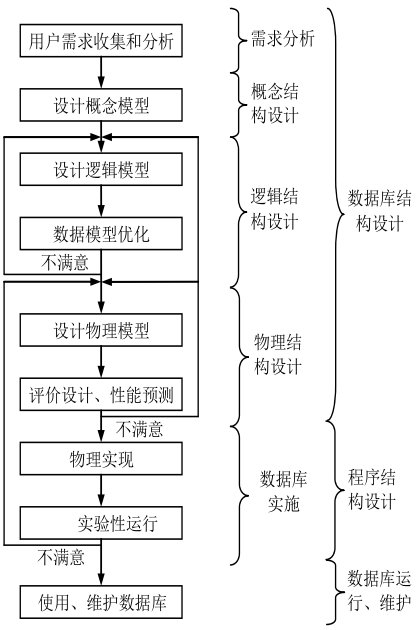


图 1-13　数据库系统设计流程

下面详细介绍数据库设计过程。

1.4.1　需求分析

准确地了解清楚用户需求，乃是数据库设计的关键。需求分析的好坏，决定了数据库设计的成败。

确定用户的最终需求是一件很困难的事情。一方面用户缺少计算机知识，开始时无法确定计算机究竟能为自己做什么，不能做什么，因此无法一下子准确地表达自己的需求，他们所提出的需求往往不断地变化。另一方面设计人员缺少用户的专业知识，不易理解用户的真正需求，甚至误解用户的需求。此外，新的硬件、软件技术也会使用户需

求发生变化。因此设计人员必须与用户不断深入地进行交流，才能逐步确定用户的实际需求。

需求分析阶段的成果是系统需求说明书，主要包括数据流图、数据字典、各种说明性表格、统计输出表、系统功能结构图等。系统需求说明书是以后设计、开发、测试和验收等过程的重要依据。

需求分析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）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的需求来设计数据库。

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。

需求分析阶段的主要任务有以下几个方面。

（1）确认系统的设计范围，调查信息需求，收集数据。分析需求调查得到的资料，明确计算机应当处理和能够处理的范围，确定新系统应具备的功能。

（2）综合各种信息包含的数据，各种数据间的关系，数据的类型、取值范围和流向。

（3）建立需求说明文档、数据字典、数据流程图。将需求调查文档化，文档既要为用户所理解，又要方便数据库的概念结构设计。需求分析的结果应及时与用户进行交流，反复修改，直到得到用户的认可。

在数据库设计中，数据需求分析是对有关信息系统现有数据及数据间联系的收集和处理，当然也要适当考虑系统在将来的需求。需求分析一般包括数据流分析及功能分析。功能分析是指系统如何得到事务活动所需要的数据，在事务处理中如何使用这些数据进行处理（也叫加工），以及处理后数据流向的全过程的分析。换言之，功能分析是对所建数据模型支持的系统事务处理的分析。

数据流分析是对事务处理所需的原始数据的收集以及处理后所得数据及其流向的分析，一般用数据流程图（DRP）来表示。在需求分析阶段，应当用文档形式整理出整个系统所涉及的数据、数据间的依赖关系、事务处理的说明和所需产生的报告，并且尽量借助数据字典加以说明。除了使用数据流程图、数据字典外，需求分析还可使用判定表、判定树等工具。

1.4.2 概念结构设计

概念结构设计是数据库设计的第二阶段，其目标是对需求说明书提供的所有数据和处理要求进行抽象与综合处理，按一定的方法构造反映用户环境的数据及其相互联系的概念模型，即用户数据模型或企业数据模型。这种概念数据模型与 DBMS 无关，是面向现实世界的数据模型，极易为用户所理解。为保证所设计的概念数据模型能正确、完全地反映用户（一个单位）的数据及其相互联系，便于进行所要求的各种处理，在本阶段设计中可吸收用户参与和评议设计。在进行概念结构设计时，可设计各个应用的视图（View），即各个应用所看到的数据及其结构，然后再进行视图集成（View Integration），以形成一个单位

的概念数据模型。形成的初步数据模型还要经过数据库设计者和用户的审查和修改，最后才能形成所需的概念数据模型。

1.4.3 逻辑结构设计

逻辑结构设计阶段的设计目标是把上一阶段得到的不被 DBMS 理解的概念数据模型转换成等价的，能为某个特定的 DBMS 所接受的逻辑模型所表示的概念模式，同时将概念结构设计阶段得到的应用视图转换成外部模式，即特定 DBMS 下的应用视图。在转换过程中要进一步落实需求说明，并使其满足 DBMS 的各种限制。逻辑结构设计阶段的结果是 DBMS 提供的用数据定义语言（DDL）写成的数据模式。逻辑结构设计的具体方法与 DBMS 的逻辑数据模型有关。

1.4.4 物理结构设计

物理结构设计阶段的任务是把逻辑结构设计阶段得到的逻辑数据库在物理上加以实现。其主要内容是根据 DBMS 提供的各种手段，设计数据的存储形式和存取路径，如文件结构、索引的设计等，即设计数据库的内模式或存储模式。数据库的内模式对数据库的性能影响很大，应根据处理需求及 DBMS、操作系统和硬件的性能进行精心设计。

1.4.5 数据库的实施

数据库实施主要包括以下工作：

◎ 用 DDL 定义数据库结构；

◎ 组织数据入库；

◎ 编制与调试应用程序；

◎ 数据库试运行。

1．定义数据库结构

确定了数据库的逻辑结构与物理结构后，就可以用选好的 DBMS 提供的数据定义语言（DDL）来严格描述数据库结构。

2．数据装载

数据库结构建立好后，就可以向数据库中装载数据了。组织数据入库是数据库实施阶段最主要的工作。对于数据量不大的小型系统，可以用人工方式完成数据入库，其步骤如下：

01 筛选数据。需要装入数据库中的数据通常都分散在各个部门的数据文件或原始凭证中，所以首先必须把需要入库的数据筛选出来。

02 转换数据格式。筛选出来的需要入库的数据，其格式往往不符合数据库要求，还需要进行转换。这种转换有时可能很复杂。

03 输入数据。将转换好的数据输入计算机。

04 校验数据。检查输入的数据是否有误。

对于大型系统，由于数据量大，用人工方式组织数据入库将会耗费大量人力物力，而且很难保证数据的正确性，因此应该设计一个数据输入子系统由计算机辅助数据入库工作。

3．编制与调试应用程序

数据库应用程序的设计应该与数据入库并行进行。在数据库实施阶段，当数据库结构建立好后，就可以开始编制与调试数据库的应用程序。调试应用程序时由于数据入库尚未完成，可先使用模拟数据。

4．数据库试运行

应用程序调试完成，并且已有小部分数据入库后，就可以开始数据库的试运行。数据库试运行也称为联合调试，其主要工作包括以下两项内容。

（1）功能测试。实际运行应用程序，执行对数据库的各种操作，测试应用程序的各种功能。

（2）性能测试。测量系统的性能指标，分析是否符合设计目标。

1.4.6 数据库的运行和维护

数据库试运行结果符合设计目标后，数据库就可以真正投入运行了。数据库投入运行标志着开发任务的基本完成和维护工作的开始，但并不意味着设计过程的终结。由于应用环境在不断变化，数据库运行过程中物理存储也会不断变化，对数据库设计进行评价、调整、修改等维护工作是一个长期的任务，也是设计工作的继续和提高。

在数据库运行阶段，对数据库经常性的维护工作主要是由 DBA 完成的。维护工作包括故障维护，数据库的安全性、完整性控制，数据库性能的监督、分析和改进，数据库的重组织和重构造。

1.5 常见关系型数据库

纵观当今的商用关系型数据库管理系统，自 20 世纪 70 年代关系模型的概念提出后，由于其突出的优势，迅速被商用关系型数据库管理系统所采用。据统计，在新发展的 DBMS 中，关系型数据库管理系统已占到 90%。其中涌现出了大量性价比高、功能强大的关系型数据库管理系统产品。本节将主要介绍 6 大常见关系型数据库管理系统。

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 Internet 应用领域数据库的佼佼者。

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公司的需要，并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 公司推出的信息管理系统（Information Management System，IMS）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特性融入 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索体系结构，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 或 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的应用。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

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件。SybaseASE 是其主要的数据库

产品，可以运行在 UNIX 和 Windows 平台。移动数据库产品有 Adaptive Server Anywhere。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 办公软件中一个重要的成员。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库，还是部门或中小公司使用的数据库，在需要管理和共享数据时，都可以使用 Access 作为数据库平台，提高个人的工作效率。例如，可以使用 Access 处理公司的客户订单数据，管理自己的个人通讯录，进行科研数据的记录和处理等。Access 只能在 Windows 系统下运行。

Access 最大的特点是界面友好、简单易用，和其他 Office 成员一样，极易被一般用户所接受。因此，在许多低端数据库应用程序中，经常使用 Access 作为数据库平台；在初次学习数据库系统时，很多用户也是从 Access 开始的。

6. MySQL

MySQL 是由瑞典 MySQL AB 公司开发的开源关系型数据库产品，目前属于 Oracle 公司。MySQL 是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就提升了速度并提高了灵活性。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库，因此 MySQL 是目前最流行的关系型数据库管理系统之一，特别是在 Web 应用方面，MySQL 是最好的 RDBMS (Relational Database Management System，关系型数据库管理系统) 应用软件。

强化训练

本章初步讲解了数据库的基本概念，并通过对数据管理技术进展情况的介绍，阐述了数据库技术产生和发展的背景，讲解了数据系统的结构，同时对数据模型特别是 E-R 模型和关系模型进行了详细的介绍，并对数据库设计步骤也做了一些介绍，使读者能够对数据库相关的基础知识有一个系统的了解，为读者学习后续课程打下良好的理论基础。

现需要设计一个成绩管理数据库系统，数据需求如下：需要存储学生的学号、姓名、年龄、性别和院系信息；需要存储课程的课程号、课程名、学分和课程性质信息；需要存储学生选课信息，包括学号、课程号和成绩；其中一个学生可以选修多门课程，一门课程也可以被多个学生选修。

请完成以下任务：

（1）请确定该系统中所涉及的实体和属性。

（2）请设计该系统的 E-R 模型。

（3）请将 E-R 模型转化为关系模式。

（4）请指出每个关系模式的主键和外键。

（5）请设计每个关系模式的完整性约束。

常见疑难解答

Q：数据库技术的发展经历了哪三个阶段？

A：人工管理阶段、文件系统阶段和数据库系统阶段。

Q：数据模型有三个要素，其中用于描述系统静态特性的是什么？

A：数据结构。

Q：学生与课程的选课关系是几对几的关系？

A：多对多的关系。

Q：键和主键的区别和联系是什么？

A： 键（key）是能够唯一确定一个元组（一条记录）的属性或者属性集合，一个表可以有多个键，所以键也称为候选键；主键是从键中选出一个作为主键，所以主键是键中的一个，主键是唯一的。

Q：数据库设计的六个阶段都有哪些？

A：需求分析、概念结构设计、逻辑结构设计、物理结构设计、数据库实施、数据库的运行和维护。