第 1 章　数据库设计基础

数据库技术是计算机领域的一个重要的分支。在信息技术日益普及的今天，人们的工作、学习和生活都已离不开数据库，数据库技术已经深入到人类社会的各个方面，并且随着计算机技术和互联网的迅猛发展，数据库技术的应用领域也在不断扩大，如企业管理、工程管理、数据统计、多媒体信息系统等领域都在利用数据库技术。

本章主要介绍数据库的基本概念和基本理论知识。

1.1　数据库的基本概念

数据库可以直观地理解为存放数据的仓库，只不过这个仓库是在计算机的大容量存储器上。数据库技术研究的问题就是如何科学地组织、存储和管理数据，如何高效地获取和处理数据。

1.1.1　数据和数据处理

1. 数据和信息

数据是描述现实世界事物的物理符号，包括数值、字符、文字、图形、图像、声音和其他的特殊符号。数据是信息的载体，是信息的具体表示形式。

信息是指现实世界事物存在方式或运动状态的反映。具体地说，信息是一种被加工成特定形式的数据，这种数据形式对接收者来说是有意义的。不经过加工处理的数据只是一种原始材料，它的价值只在于记录了客观世界的事实。只有经过提炼和加工，原始数据才发生了质的变化，产生一定的意义。

数据与信息既有联系，又有区别。一方面，数据是信息的符号表示或载体，信息则是数据的内涵，是数据所包含的意义。另一方面，数据具有任意性，可用多种不同的数据形式来表示一种同样的信息，而信息不随表示它的数据形式不同而改变。例如，一个城市的天气预报情况是一条信息，而描述该信息的数据形式可以是文字、图像或声音等。

2. 数据处理

数据处理是指将数据转换成信息的过程，包括对数据的收集、存储、加工或计算、打印各类报表、传输等的一系列活动。其目的之一是从大量原始数据中抽取、推导出对人们有价值的信息，然后利用信息作为行动和决策的依据；另一目的是为了借助计算机科学地保存和管理复杂的、大量的数据，以使人们能够方便而充分地利用这些宝贵的信息资源。

例如，全校大一学生《大学计算机基础》的考试成绩记录了考生的考试情况（属于原始数据），对考试成绩分班统计（属于数据处理）的结果，可以作为任课教师教学评价的依据之一（属于信息），或者对考试成绩按不同的题型得分进行分类统计（属于数据处理），

可得出试题分布和难易程度的分析报告（属于信息）。

1.1.2 数据管理技术的发展

计算机对数据的管理是指对数据的组织、分类、编码、存储、检索和维护提供操作手段。随着计算机软硬件技术和计算机应用范围的发展，数据管理技术也在不断地改进，大体上经历了 4 个阶段：人工管理阶段、文件系统阶段、数据库系统阶段和分布式数据库系统阶段。

1. 人工管理阶段

20 世纪 50 年代以前，计算机主要用于数值计算。在硬件方面，外存储器只有卡片、纸带和磁带，存储信息容量小，存取速度慢。软件方面没有系统软件和管理数据的软件，程序员不但要负责处理数据还要负责组织数据。程序员直接与物理设备打交道，从而使程序与物理设备高度相关，一旦物理存储发生变化，程序必须全部修改，程序没有任何独立性，如图 1-1（a）所示。

此阶段数据管理的主要特点是：

（1）程序之间不能共享数据。程序代码与数据同处于一个程序中，即一组数据对应一个程序，一个程序不能使用另一个程序中的数据。

（2）程序复杂。由于没有专门的软件对数据进行管理，因此在程序中必须定义数据存储结构，需要编写数据存取方法和输入输出方式等程序。

（3）数据量小且无法长期保存。程序运行时，人工进行数据输入，输入数据和运行结果都保存在内存中，随着程序运行结束，这些数据自动消失，很难实现大数据量处理的任务。

（4）数据重复输入量大。当一个程序用到另一个程序处理结果时，需要重新输入这些数据。一个程序多次运行也可能导致人工重复输入数据。

2. 文件系统阶段

20 世纪 50 年代后期至 60 年代中期，计算机软、硬件有了很大发展。外存储器有磁鼓和磁盘等直接存取设备，存储信息容量和存取速度得到很大改进；软件方面有了操作系统和文件系统，程序通过文件系统访问数据文件，如图 1-1（b）所示。

此阶段数据管理的主要特点是：

（1）程序之间可以共享数据，且易于长期保存。程序代码和数据可以分别存储在各自文件中，在一个程序中输入数据或运行结果可以保存到数据文件中，供其他程序使用。即一组数据可以在多个程序中使用。

（2）程序代码有所简化。数据存储结构、存取方法等都由文件系统负责处理，程序中通过文件名即可存取数据文件中的数据。

（3）数据冗余度大。数据文件通常是非结构化文件，如顺序文件、随机文件等，都没有列标识，不便于多个程序员使用。通常一个数据文件对应一个程序员编写的一组程序，因此，多个程序员的相同数据可能出现重复存储问题。

（4）程序对数据依赖性较强。改变数据文件中数据项的位置或宽度后，可能需要修改程序代码。

（5）专业性较强。对数据文件访问（存取、分类、检索和维护等）通常需要编写程序，因此，使用计算机的人员要具有很强的计算机专业知识。

3. 数据库系统阶段

20 世纪 60 年代后期至 70 年代后期，计算机系统有了进一步发展，外存储器有了大容量磁盘，存取数据的速度明显提高而且价格下降，这就有可能克服文件系统管理数据时的不足，而去满足和解决实际应用中多个应用程序共享数据的要求，从而使数据能为尽可能多的应用程序服务，这就出现了数据库这样的数据管理技术，如图 1-1（c）所示。数据库技术日趋成熟，出现了许多数据库管理系统。例如，在微型计算机上流行 dBASE 系列数据库管理系统，在大、中、小型计算机上使用 Oracle 数据库管理系统等。

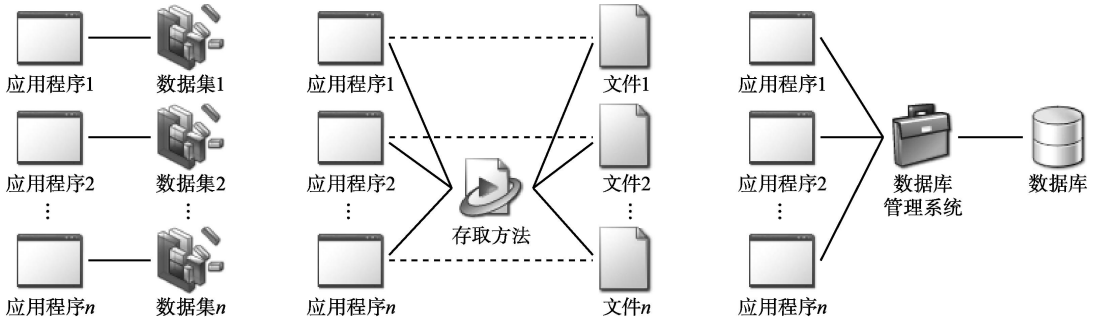


图 1-1　各个数据管理阶段中应用程序和数据之间的对应关系

此阶段数据管理的主要特点是：

（1）数据集中式管理，高度共享。

（2）数据结构化并与程序分离。

（3）数据冗余度小，并具有一定的一致性和完整性等。

数据管理在数据库系统阶段，经历了层次数据库和网状数据库阶段，到了 20 世纪 70 年代，出现了关系数据库系统，并逐渐取代了前两种系统，占据了数据库领域的主导地位，成为当今较为流行的数据库系统。

4. 分布式数据库系统阶段

20 世纪 80 年代初期至今，随着计算机网络技术的发展，一个部门的多台计算机进行连接构成局域网，甚至跨地区、跨国别的多台计算机进行连接构成广域网或 Internet 网，网络技术的发展为分布式数据库系统提供了良好的运行环境。

分布式数据库可以将数据存放在多台计算机上，可以在不同位置访问数据库中的数据。目前支持分布式数据库的数据库管理系统有 Access、SQL Server、Oracle 和 Sybase 等。

分布式数据库比集中式数据库的功能更加强大，主要特点是：

（1）数据局部自治与集中控制相结合，具有很强的可靠性和可用性。

（2）强大数据共享和并发控制能力，使数据的使用价值更高，应用范围更大。

（3）数据一致性和安全性控制措施更加完善。

1.1.3 数据库系统

数据库系统（DataBase System，DBS）是指在计算机系统中引入数据库之后组成的系统，它可以实现有组织地、动态地存储大量相关数据，提供数据处理和信息资源共享服务。它由系统硬件平台、系统软件平台、数据库、数据库管理系统、应用软件、应用界面和相关人员组成，如图 1-2 所示。

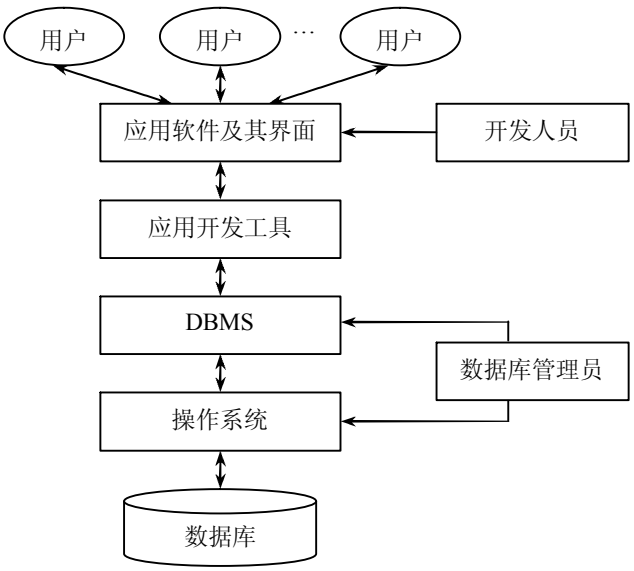


图 1-2 数据库系统的组成

1. 数据库（DataBase，DB）

数据库是数据的集合，是长期存储在计算机内的、有组织的、可共享的数据的集合。数据库中的数据按一定的数据模型组织、描述和存储，具有很小的冗余度、较高的数据独立性和易扩展性，可为各种用户共享。数据库不仅包含描述事物的数据本身，也包含数据之间的联系。

2. 数据库管理系统（DataBase Management System，DBMS）

数据库管理系统是数据库系统的核心，是一种系统软件，数据库中的数据组织、操纵、维护、控制、保护和数据服务等功能都是由数据库管理系统来完成的。数据库管理系统是位于用户与操作系统之间的数据管理软件。

3. 相关人员

数据库系统中的相关人员包括数据库管理员（DataBase Administrator，DBA）、应用程序开发人员和最终用户。

（1）数据库管理员。由于数据库的共享性，因此对数据库的规划、设计、维护和监视等需要有专人管理，他们被称为数据库管理员。数据库管理员主要负责数据库的设计和安装数据库管理系统，通过数据库管理系统建立和维护数据库，制定安全策略，保障计算机软硬件系统的正常运行。

（2）应用程序开发人员。主要工作是按照应用领域的业务人员要求，利用数据库系统资源开发符合业务要求的应用程序。有些简单的应用程序，可以由非计算机专业人员开发，但一些较复杂或者大型的应用程序，通常由计算机专业人员开发。

（3）用户。通常是应用领域的人员，如教务管理系统的教师和学生，电子商务系统的商家和客户等。

1.1.4　数据库管理系统的基本功能

数据库管理系统管理的对象主要是数据库，其基本功能包括：

（1）数据定义。通过 DBMS 提供的数据定义语言（Data Definition Language，DDL）可以定义数据库、数据库表、视图和索引等数据库中的相关信息。

（2）数据操纵。通过 DBMS 提供的数据操纵语言（Data Manipulation Language，DML）可以对数据库中的数据进行插入、修改和删除。

（3）数据查询。通过数据查询语言（Data Query Language，DQL）可以对数据进行查询、排序、汇总和表连接等操作。

（4）数据库运行管理和控制。这是 DBMS 的核心部分，DBMS 提供的数据控制语言（Data Control Language，DCL）负责数据库并发控制（协调多个用户对数据库同时操作，并确保数据一致性），安全性（密码和权限）检查，完整性约束条件（数据的正确性）检查和执行，数据库内部资料（如索引、数据字典）自动维护等。所有数据库的各种操作都要在 DBMS 的统一控制和管理下进行操作，以确保数据的安全性、完整性以及对数据库的并发使用。

（5）数据组织、存储和管理。DBMS 要对数据字典（存放数据库结构的描述信息，如表中字段名和数据类型等）、用户数据和存取路径等信息进行分类组织、存储和管理，确定文件结构和存取方式，实现数据之间的联系，以便节省存储空间和提高数据处理速度。

（6）数据库维护。数据库维护主要包括数据更新和转换（实现与其他软件的数据转换），数据库转存和恢复，数据库重新组织、结构维护和性能监视等。

（7）数据通信。DBMS 要经常与操作系统打交道，进行信息交换，因此，必须提供与操作系统的联机处理、分时处理和远程作业传输接口。

1.1.5　数据库系统的特点

数据库系统脱胎于文件系统，两者都以数据文件的形式组织数据，但数据库系统由于引入了 DBMS，与文件系统相比，它具有以下 4 个特点：

（1）数据的结构化。在数据库系统中，数据是面向整体的，不但数据内部组织有一定的结构，而且数据之间的联系也按一定的结构描述出来，所以数据整体具有结构化的特点。

（2）数据的高共享性，低冗余性，易扩充性。数据库系统是面向整体的，因此数据不但可以被多个用户共享使用，从而大大减少了冗余度，而且可以很容易地增加新的功能，以适应用户新的要求。

（3）数据的高独立性。通过数据库系统提供的二级映像功能，使得数据既具有物理独

立性，又具有逻辑独立性。处理数据时，数据的逻辑结构与物理结构是相互独立的。数据的存储与使用数据的程序之间相互独立，即当改变数据存储结构时程序应尽量不受影响，若程序修改时，也不要求数据结构做较大的改变。

（4）数据的统一管理和控制。数据库管理系统在数据库建立、运用和维护时对数据进行统一控制，以保证数据的完整性、安全性，并在多用户同时使用数据库时进行并发控制，在发生故障后对系统进行恢复。

数据的完整性是指确保数据库中数据的正确性、有效性和兼容性。正确的数据不一定是有效的，因此系统必须能够进行检验，并能够对数据进行统一的控制。安全性是指数据库在使用中，数据要被多个用户共享，就必须对数据加以保护，防止数据的意外丢失与破坏。因此，需要设置用户的使用权限以防止数据的非法使用。并发控制是控制多个应用的并发访问所产生的干扰，以保证其正确性。另外，一旦数据库遭到破坏，系统应有能力将其恢复到可用状态。

1.1.6 数据库系统的内部体系结构

数据库系统在体系结构上通常都具有相同的特征，即采用三级模式结构，并提供二级映像功能。

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外模式、概念模式和内模式三级构成的。数据按外模式的描述提供给用户，按内模式的描述存储在磁盘中，而概念模式提供了一种约束其他两级的相对稳定的中间观点，它使得这两级的任何一级的概念都不受另外一级的牵制。3 个级别的数据结构与应用程序以及数据库的联系如图 1-3 所示。

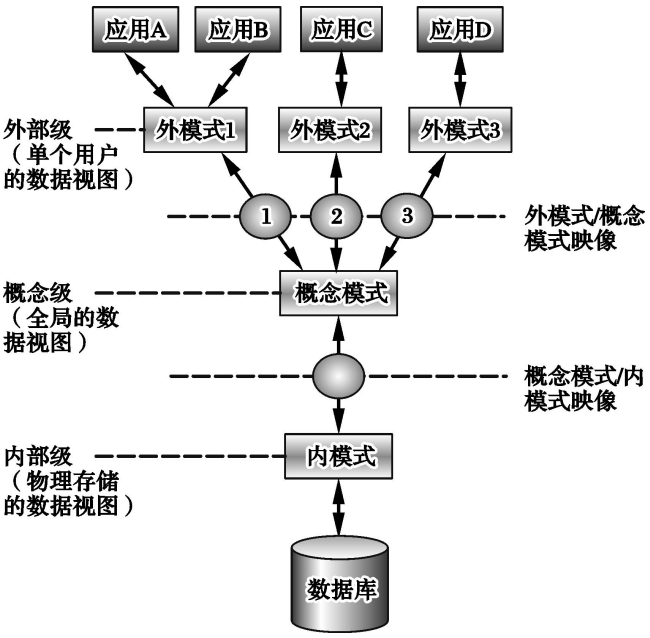


图 1-3　数据库系统三级模式的结构

（1）外模式。

外模式也称为子模式，属用户级数据库，是用户与数据库系统的接口，是用户用到的那部分数据（局部数据）的逻辑结构和特征的描述，又称用户视图，是与某一应用有关的数据的逻辑表示。

一个数据库可以有多个外模式。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，则他们的外模式描述就是不同的。另一方面，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。

（2）概念模式。

概念模式也称为模式，是概念级数据库，是数据库中全部数据的整体逻辑结构的描述，是所有用户的公共数据视图，又称 DBA 视图。它是数据库系统模式结构的中间层，不涉及存储结构、访问技术细节和硬件环境，也不涉及具体的应用程序以及开发工具，由若干个概念记录类型组成。概念模式不但要描述概念记录类型，还要描述记录之间的联系、所允许的操作、数据的一致性、安全性和其他数据控制方面的要求。

一个数据库只有一个概念模式。它以某一种数据模型为基础，统一综合考虑了所有用户的需求，并将这些需求有机地结合成一个逻辑整体。

（3）内模式。

内模式也称为存储模式，属物理级数据库，内模式是数据物理结构和存储结构的描述，是数据在数据库内部的表示方式，又称内部视图。例如，记录的存储方式，索引按何种方式组织，数据是否压缩存储、是否加密，数据的存储记录格式有何规定等。

一个数据库只有一个内模式，它仍独立于具体的存储设备。

2. 数据库系统的二级映像功能与数据独立性

数据库系统的三级模式是数据的三个抽象级别，它把数据的具体组织留给 DBMS 管理，使用户能抽象地处理数据，而不必关心数据在计算机中的表示和存储方式。这三级结构之间往往差别很大，为实现这 3 个抽象级别的转换，DBMS 在这三级结构之间提供了两层映像：外模式/概念模式映像和概念模式/内模式映像。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）概念模式/内模式映像保证了数据的物理独立性。

数据库只有一个概念模式，也只有一个内模式，所以概念模式/内模式映像是唯一的。这个映像存在于概念级和内部级之间，用于定义数据全局结构和存储结构之间的对应关系，该映像一般放在内模式中描述。有时也称为“模式/内模式映像”。

当数据库的内模式要作修改，即数据库的存储设备和存储方法有所变化，那么概念模式/内模式映像也要作出相应的修改，但概念模式很可能仍然保持不变。也就是对内模式的修改尽量不影响概念模式，当然对于外模式和应用程序的影响更小，这样就称数据库达到了数据的物理独立性。

（2）外模式/概念模式映像保证了数据的逻辑独立性。

对应于同一个概念模式可以有任意多个外模式。对于每一个外模式，数据库系统都有一个外模式/概念模式映像，这个映像存在于外部级和概念级之间，用于定义外模式和概念

模式间的对应关系，这些映像定义通常包含在各自的外模式描述中。

如果数据库的整体逻辑结构（即概念模式）要作修改，那么外模式/概念模式映像也要作相应的修改，但外模式很可能仍然保持不变。也就是对概念模式的修改尽量不影响外模式，当然对于应用程序的影响就更小，这样就称数据库达到了数据的逻辑独立性。

1.2 数据模型

模型是对现实世界特征的模拟和抽象。数据模型也是一种模型，只不过它模拟的对象是数据。在数据库技术中，用数据模型这个工具来抽象、表示和处理现实世界的数据和信息。

1.2.1 数据模型的基本概念

怎样把现实世界中的事物及事物之间的联系在数据库中用数据描述出来，是数据库技术中的一个基本问题。由于计算机不可能直接处理现实世界中的具体事物，因此必须把现实世界中存在的具体事物转换成计算机能够处理的数据。数据库中的数据模型可以将复杂的现实世界的要求反映到计算机数据库中的物理世界，这种反映是一个逐步转化的过程，是从现实世界开始，经历信息世界而至计算机世界的过程，三个世界及其关系如图 1-4 所示。

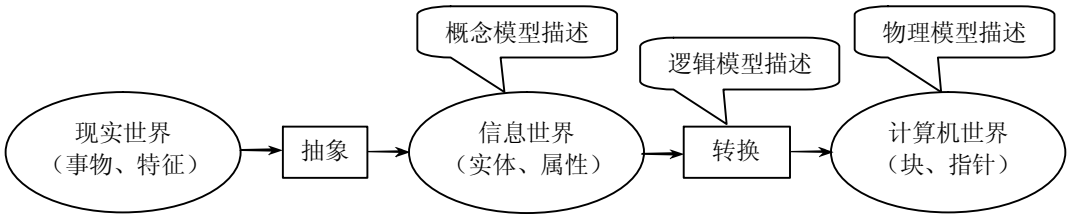


图 1-4　三个世界及其关系

（1）现实世界。客观存在的事物及其相互间的联系构成现实世界。在现实世界中，人们通过事物的特征找出事物之间的差别，以便区分事物，而事物之间又存在着某种必然的联系。例如，出版社、作者和书都是客观事物，出版社与书通过出版相联系；作者与书通过编写相联系。

（2）信息世界。信息世界是客观世界在人们头脑中的反映，是人们对客观事物及其联系的抽象描述和概念化。从本质上来看，信息世界是抽象化的现实世界。在信息世界中，人们将客观事物视为实体，如上述的出版社、作者和书都是实体；将事物的特性视为属性，如出版社的地址是出版社实体的一个属性。

（3）计算机世界。计算机世界是在信息世界基础上致力于其在计算机物理结构上的描述，现实世界的要求只有在计算机世界中才得到真正的物理实现，而这种实现是通过信息世界逐步转化得到的。

表示实体及实体之间联系的模型称为数据模型。数据模型按不同的应用层次分成 3 种类型，分别是概念数据模型、逻辑数据模型和物理数据模型。

（1）概念数据模型。概念数据模型简称概念模型，它是一种面向客观世界、对客观世界进行第一层抽象的模型，用于建立信息世界的数据模型。它与具体的数据库管理系统无关，与具体的计算机平台无关。概念模型着重于对客观世界复杂事物的结构描述及它们之间的内在联系的刻画。概念模型是整个数据模型的基础。目前，较为有名的概念模型有 E-R 模型。

（2）逻辑数据模型。逻辑数据模型又称数据模型，它是一种面向数据库系统的模型，与 DBMS 有关。该模型着重于在数据库系统一级的实现。概念模型只有在转换成数据模型后才能在数据库中实现。目前，逻辑数据模型也有很多种，较为成熟并先后被人们大量使用过的有层次模型、网状模型、关系模型等。

（3）物理数据模型。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。数据库的数据最终必须存储到存储介质上，反映数据的物理存储方式（例如块、指针、索引）的数据模型称为物理数据模型。物理模型不但与 DBMS 有关，而且与操作系统和计算机硬件有关。

1.2.2　概念模型

概念模型是对客观事物及其联系的抽象，用于信息世界的建模。

概念模型中的主要术语如下：

（1）实体。现实世界客观存在且可相互区分的事物叫实体。实体既可以是实际存在的对象，比如一位教师、一本教材、一台机器等，也可以是某种抽象的概念或事件，比如一门课程、一个专业、一次借阅图书、一个运行过程等。

（2）属性。实体所具有的某一方面的特性称为实体的属性。每个实体都具有多个属性，即多个属性才能描述一个实体。如教师实体可用教师编号、姓名、性别、年龄和职称等属性来描述。学生实体有学号、班级、姓名、性别和年龄等属性。

（3）域。属性的取值范围叫域。如性别属性的域为男、女。

（4）关键字。实体的某一属性或属性集合，其取用的值若能唯一标识出某一实体，则称其为关键字，也称码。如学号可以是学生实体集的关键字，因为任何两个学生的学号不会相同，但由于学生的姓名有相同的可能，故不应作为关键字。

（5）实体类型。用实体名及其所有属性名的集合表示一种实体类型，简称实体型，通常一个实体型表示一类实体。因此，通过实体型可以区分不同类型的事物。例如，分别用：教师（编号，姓名，性别，出生日期，职称，联系电话，是否在职）、课程（编号，名称，开课学期，理论学时，实验学时，学分）的形式来描述教师类实体和课程类实体。

（6）实体集。具有相同属性的实体集合称为实体集。如全体学生就是一个实体集。

在关系数据库（如 Access、Visual FoxPro、Oracle 和 Sybase 等）中，通常将同一种实体型的数据存放在一个“表”中，实体属性集合作为表结构，而一个实体属性值的集合作为表中一个“记录”，表示一个实体，用“字段”来表示实体的属性，用“表”来表示实体集，表示一种关系，对应一个文件。显然，字段的集合组成一个记录，记录的集合组成一个表（文件）。

（7）实体之间的联系。现实世界中事物之间是相互关联的，这种关联在事物数据化过程中表现为实体之间的对应关系，通常将实体之间的对应关系称为联系。实体之间的联系归纳起来有 3 种类型：

①一对一联系（1:1）。设 A、B 为两个实体集，一对一联系是指实体集 A 中的实体与实体集 B 中的实体之间存在一一对应关系。例如，一个班级只有一个班长，一个人不会同时在两个（或以上）班级任班长，因此班级与班长之间是一对一的联系。同样，行驶中的汽车与司机之间也是一对一的联系。

②一对多联系（1:n）。一对多联系是指实体集 A 中的每个实体可以与实体集 B 中的多个实体有联系，反过来实体集 B 中的每个实体与实体集 A 中的一个实体有联系。例如，一个班级有多个学生，而某个学生只隶属于一个班级，因此班级与学生之间是一对多的联系。出租车公司与出租车也是一对多的联系。

③多对多联系（m:n）。多对多联系是指实体集 A 中的每个实体可以与实体集 B 中的多个实体有联系，反过来实体集 B 中的每个实体也可以与实体集 A 中的多个实体有联系。例如，一个学生选修多门课程，而一门课程有多名学生选修，因此学生与课程之间是多对多的联系。又如，一个用人单位需要多个专业的学生，而一个专业的学生到多个用人单位工作，因此用人单位与专业之间也是多对多的联系。

需要说明的是，实体之间的联系类型会随着时间、地点的变化而变化。例如，在某一时刻，一个教师只能在一个教室为多个学生上课，此时教师与学生之间是一对多的联系；而在不同的时刻，一个教师可在多个教室为多个学生上课，而一个学生又可能听取多个不同教师的授课，此时教师与学生之间又变成了多对多的联系。

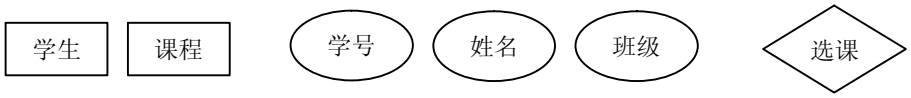
1.2.3 E-R 模型

概念模型有多种，其中较著名的是实体联系模型（Entity Relationship Model），简称为 E-R 模型，它是直接从现实世界中抽象出实体及实体之间的联系，然后用实体联系图（简称为 E-R 图）表示数据模型。E-R 图是描述概念世界、建立概念模型的实用工具。在 E-R 图中：

（1）实体：用矩形框表示，框内写明实体名，如图 1-5（a）表示“学生”和“课程”实体集。

（2）属性：用椭圆表示，椭圆内写明属性名，用实线将其与相应实体连接起来，如图 1-5（b）表示学生属性“学号”“姓名”及“班级”。

（3）联系：用菱形框表示，框内写明联系名，同时标上联系类型，如图 1-5（c）表示学生和课程间的“选课”联系。



（a）实体集　　（b）属性　　（c）联系

图 1-5　E-R 图基本图形

图 1-6、图 1-7、图 1-8 给出了几种实体间联系的 E-R 图表示。

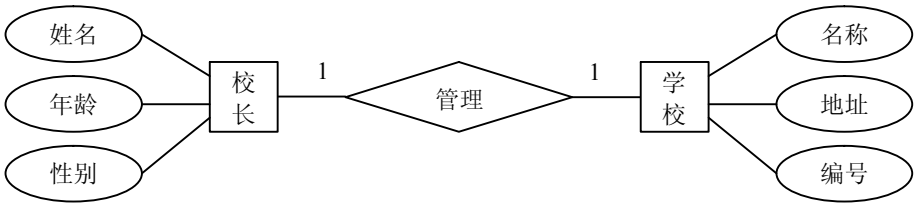


图 1-6　校长与学校间的一对一联系 E-R 图

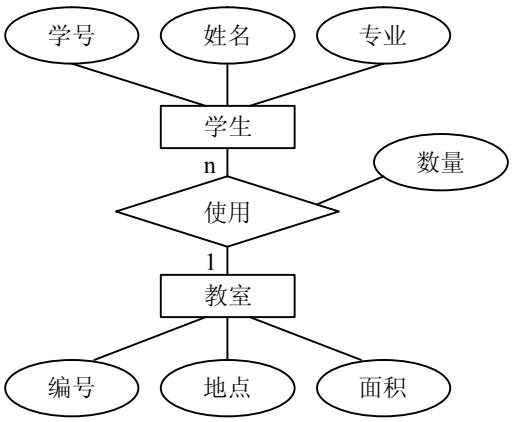


图 1-7　教室与学生间一对多联系 E-R 图

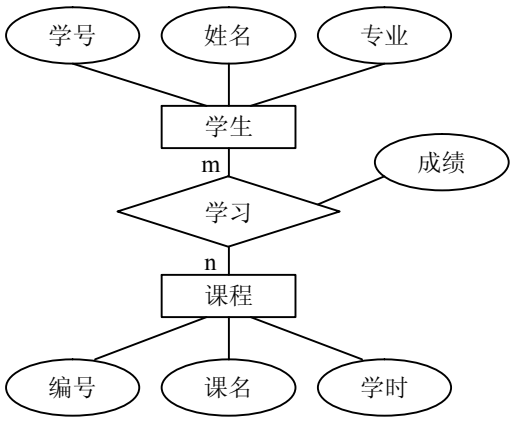


图 1-8　学生与课程间多对多联系 E-R 图

1.2.4　逻辑数据模型

逻辑数据模型也称为数据模型，是面向数据库系统，在数据库系统中表示实体和实体之间联系的模型。

1. 逻辑数据模型

常见的逻辑数据模型有 3 种：层次模型、网状模型和关系模型。

（1）层次模型。

层次模型是通过树形结构表示实体及实体之间联系的数据模型，“树”中每个结点表示一个实体，结点之间的箭头表示实体间的联系（由父到子）。典型的层次模型如图 1-9 所示。

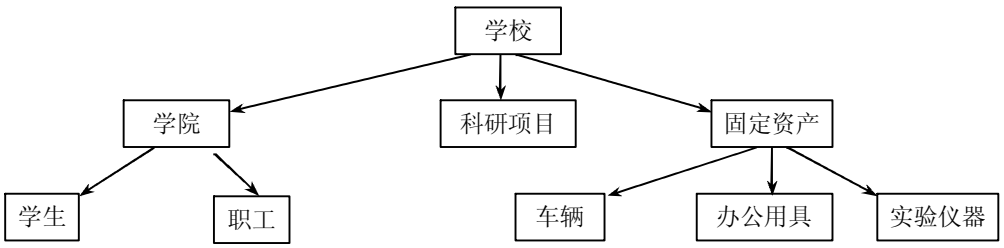


图 1-9　层次模型示例

层次模型的特点是：有且仅有一个结点没有父结点，称之为根结点（如“学校”），每个非根结点有且仅有一个父（直接上层）结点。

在层次模型中，有的结点（如“固定资产”）具有子结点（“车辆”“办公用具”和“实验仪器”），将这样的结点称为父结点，而这些子结点相互称为兄弟结点；有的结点（如“科研项目”“职工”“车辆”等）没有子结点，将这样的结点称为叶子结点。

在数据库技术中，将支持层次数据模型的数据库管理系统称为层次数据库管理系统。

（2）网状模型。

网状模型是通过网状结构表示实体及实体之间联系的数据模型，“网”中每个结点表示一个实体，结点之间的箭头表示实体间的联系（由父到子）。典型的网状模型如图 1-10 所示。

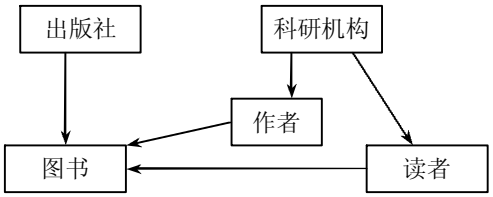


图 1-10　网状模型示例

网状模型的特点是：可能有多个结点（如“出版社”“科研机构”）没有父结点，即有多个根结点，某个非根结点（如“图书”）可能有多个父结点。

在数据库技术中，将支持网状数据模型的数据库管理系统称为网状数据库管理系统。

（3）关系模型。

关系模型是通过二维表的形式描述实体及实体之间联系的数据模型，表中一行数据描述一个实体。

如图 1-11 是“学生”“民族”和“专业”三类实体的关系模型，通过学生表中“民族码”属性值可以分析出来哪些学生属于同一个民族；通过“专业码”属性值可以分析出哪些学生同修一个专业。由此可见，在关系模型中，通过模型中的数据可以反映出实体间的联系。

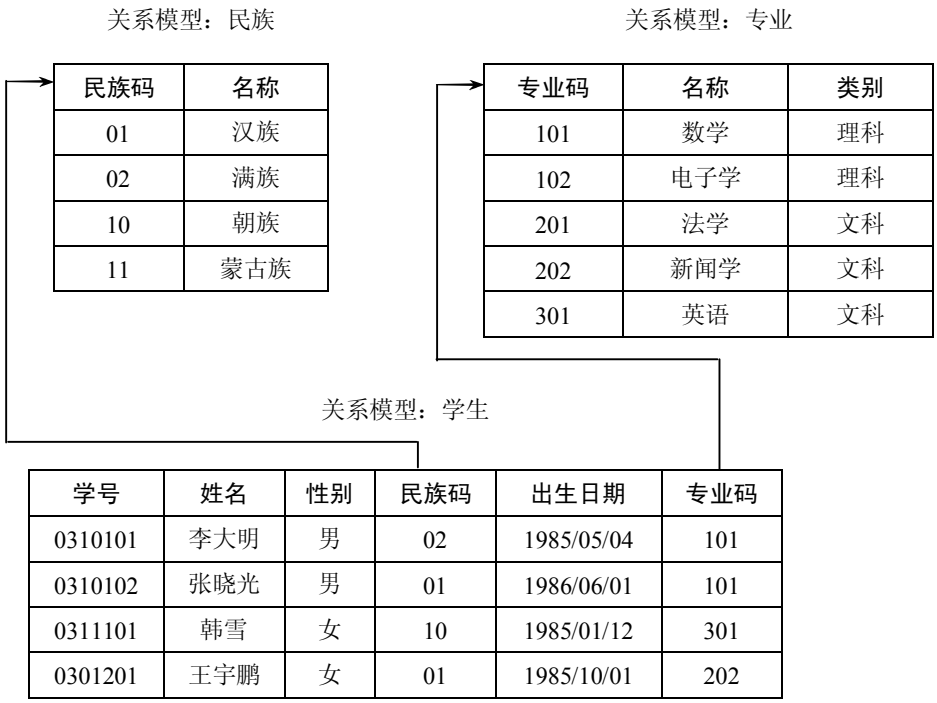
学生表中“民族码”属性与民族表中“民族码”属性具有相同的意义，都表示民族编码，通过这两个具有相同意义的属性将两张表联系起来；同样，通过学生表中“专业码”和专业表中“专业码”两个具有相同意义的属性将学生表和专业表联系起来。由此可见，在关系模型中，可以通过各表中具有相同意义的属性来建立实体之间的联系。

在数据库技术中，将支持关系数据模型的数据库管理系统称为关系数据库管理系统。

在上述三种数据模型中，关系模型是目前使用较多的重要数据模型，常用的数据库管理系统软件都支持该模型。

2. 关系模型中的基本术语

（1）关系：每一张二维表称为一个关系，每个关系有一个关系名。在计算机中，一个关系可以存储为一个数据库文件的表，如上述的“民族表”“专业表”“学生表”等。



民族码	名称
01	汉族
02	满族
10	朝族
11	蒙古族

专业码	名称	类别
101	数学	理科
102	电子学	理科
201	法学	文科
202	新闻学	文科
301	英语	文科

学号	姓名	性别	民族码	出生日期	专业码
0310101	李大明	男	02	1985/05/04	101
0310102	张晓光	男	01	1986/06/01	101
0311101	韩雪	女	10	1985/01/12	301
0301201	王宇鹏	女	01	1985/10/01	202

图 1-11　关系模型示例

（2）元组：表中的行称为元组，一行是一个元组，与文件中的记录相对应。如“学生表”中姓名为“李大明”的学生所在的一行就是一条记录，“学生表”中共有 4 条记录。

（3）属性：表中的列称为属性，一列是一个属性，对应于文件中的一个字段。属性名即字段名。如“民族表”中的“民族码”“名称”等字段。

（4）属性值：行与列的交叉位置上的数据。如“民族表”中的“汉族”，“专业表”中的“数学”，“学生表”中的“0310101”“韩雪”等。

（5）域：属性的取值范围。

（6）表结构：表结构是表中的第一行，由各属性名组成。

（7）候选键：在一个关系中，其值能唯一地标识一个元组（记录）的一个或多个属性，称为候选键。如“专业表”的候选键有“专业码”和“名称”，“学生表”的候选键只有“学号”。

（8）主关键字：一个表中可能有多个候选键，但在实际应用中只能选择一个候选键，将用户选用的候选键称为主关键字，也可简称为主键。主键除了标识元组外，还在建立表之间的联系方面起着重要作用。

（9）外部关键字：如果一个关系 R 的一个或一组属性 F 不是关系 R 的候选键，但 F 与某关系 S 的主键相对应（对应属性含义相同），则 F 是关系 R 的外部关键字，简称外键。例如，“专业码”是“学生表”的一个属性（非候选键），如果将“专业码”选为“专业表”的主键，则“专业码”就是“学生表”的一个外键。

（10）主表和从表：主表和从表是指通过外键相关联的两个表，其中以外键为主键的

表称为主表，外键所在的表称为从表。如若把“民族表”视为主表，则“学生表”便是其从表。

3. 关系模型的特点

关系模型主要具有以下特点：

（1）每一列中的分量是同一类型的数据，来自同一个域。

（2）同一关系中不能有相同的属性名，即字段名不能相同。

（3）任意两个元组不能完全相同。

（4）列的次序可以任意交换。

（5）行的顺序可以任意交换。

（6）每一个分量必须是不可分的数据项。

1.3 关系数据库

尽管数据库领域中存在多种组织数据的方式，但关系数据库是效率最高的一种数据库系统，是当今世界的主流数据库。关系数据库系统（Relation DataBase System，简称 RDBS）采用关系模型作为数据的组织方式，Access 就是基于关系模型的数据库系统。

1.3.1 关系模型的组成

关系模型由关系数据结构、关系操作和关系完整性约束三部分组成。

（1）关系数据结构。关系模型中数据的逻辑结构是一张二维表。在用户看来非常单一，但这种简单的数据结构能够表达丰富的语义，可描述出现实世界的实体以及实体间的各种联系。如一个学校可以有一个数据库，在数据库中建立多个表，其中一个表用来存放教师信息，一个表用来存放学生信息，一个表用来存放课程设置信息等。

（2）关系操作。关系操作采用集合操作方式，即操作的对象和结果都是集合。关系模型中常用的关系操作包括两类。

①查询操作：选择、投影、连接、除、并、交、差等。

②增加、删除、修改等操作。

（3）关系完整性约束。关系模型中的完整性是指数据库中数据的正确性和一致性，关系数据模型的操作必须满足关系的完整性约束条件。关系的完整性约束条件包括实体完整性、参照完整性和用户定义的完整性。其中实体完整性和参照完整性是关系模型必须满足的完整性约束条件，适用于任何关系数据库系统。用户定义的完整性是针对某一具体领域的约束条件，它反映某一具体应用所涉及的数据必须满足的语义要求。

1.3.2 关系代数

1. 传统的集合运算

对关系数据库进行查询时，需要找到用户感兴趣的数据，这就需要对关系进行一定的运算。关系代数是一种抽象的查询语言，它是用对关系的运算来表达查询的。

运算对象、运算符和运算结果是关系代数的三大要素。关系代数的运算对象是关系，运算结果亦为关系，运算主要分为两类：一类是传统的集合运算，另一类是专门的关系运算。前者是从关系的“水平”方向即行的角度来进行的；后者的运算不仅涉及行而且还涉及列。

传统的集合运算都是二目运算。设关系 R 和关系 S 具有相同的目 n，即关系 R 和关系 S 有相同的属性个数 n，且相应的属性取自同一个域，则可以定义如下 4 种运算：

（1）并（Union）。

关系 R 和关系 S 的并记作 R∪S，由属于 R 或属于 S 的元组组成，结果仍为 n 目关系。关系的插入操作可用并运算表示。

（2）交（Intersection）。

关系 R 和关系 S 的交记作 R∩S，由属于 R 且属于 S 的元组组成，结果仍为 n 目关系。

（3）差（Difference）。

关系 R 和关系 S 的差记作 R－S，由属于 R 而不属于 S 的元组组成，结果仍为 n 目关系。

关系的删除操作可用差运算表示。关系的修改操作可用差和并运算表示。

如有两个二目关系 R 和关系 S，关系 R 代表参加项目 1 的员工，关系 S 代表参加项目 2 的员工，如表 1-1 和表 1-2 所示。

表 1-1　关系 R

员工编号	员工姓名
01001	王磊
01003	张晓华
01005	刘洋

表 1-2　关系 S

员工编号	员工姓名
01002	王浩田
01003	张晓华
01004	孟德水

关系 R 和关系 S 的并运算、交运算和差运算的结果如表 1-3 所示。

表 1-3　关系 R 和关系 S 的并运算、交运算和差运算

并运算

员工编号	员工姓名
01001	王磊
01002	王浩田
01003	张晓华
01004	孟德水
01005	刘洋

交运算

员工编号	员工姓名
01003	张晓华

差运算

员工编号	员工姓名
01001	王磊
01005	刘洋

（4）广义笛卡尔积。

两个分别为 m 目和 n 目的关系 R 和 S 的广义笛卡尔积 R×S 是一个(m+n)列的元组的集合。元组的前 m 列是关系 R 的元组，后 n 列是关系 S 的元组。若 R 有 k1 个元组，S 有 k2 个元组，则 R×S 有 k1×k2 个元组。其直观含义是诸集合各元组间一切可能的组合。两个关系的合并操作可用广义笛卡尔积运算表示。

如有学生关系 R 和课程关系 S，如表 1-4 和表 1-5 所示，其广义笛卡尔积的结果如表 1-6 所示。

表 1-4　学生关系 R

学号	姓名	性别
20070501	朱晓强	男
20070502	方根学	男
20070503	欧玲玲	女

表 1-5　课程关系 S

课号	课名
41412051	大学计算机基础
41412061	C 程序设计

表 1-6　关系 R 和关系 S 的广义笛卡尔积运算

学号	姓名	性别	课号	课名
20070501	朱晓强	男	41412051	大学计算机基础
20070501	朱晓强	男	41412061	C 程序设计
20070502	方根学	男	41412051	大学计算机基础
20070502	方根学	男	41412061	C 程序设计
20070503	欧玲玲	女	41412051	大学计算机基础
20070503	欧玲玲	女	41412061	C 程序设计

2. 专门的关系运算

对于关系数据库，已经有了结构化查询语言（Structured Query Language），简称为 SQL 语言，它对表具有很强的操纵能力。在多数关系数据库管理系统中除了支持 SQL 语言外，自身也提供了许多操作表的功能，不同关系数据库管理系统提供的功能可能有些差异，但它们检索数据的操作都是以选择、投影和联接 3 种关系的基本操作为核心。

（1）选择（Selection）。

选择操作是从关系中选取满足某种条件的元组（记录）的操作。通常在命令中加上条件子句和逻辑表达式来完成选择操作。选择运算的结果构成关系的一个子集，是关系中的部分元组，其关系模式不变。

（2）投影（Projection）。

投影是从关系中选取若干个列的操作。通常在命令中加上要选取的各个列名称来完成投影操作，投影运算的结果使关系模式发生改变。

（3）联接（Join）。

在数学上，可以用笛卡尔积建立两个关系间的联接，但这样得到的关系庞大，而且数据大量冗余。在实际应用中，一般两个相互联接的关系往往需要满足一些条件，所得到的结果也较为简单。这样就引入了联接运算和自然联接运算。

联接操作是对两个关系进行联接生成一个新的关系，新关系中所含的列是被联接的两个关系中列的并集或是该并集的子集，新关系中包含的元组（记录）是满足联接条件的所

有元组（记录）的集合。

联接条件中的运算符为比较运算符，当此运算符取“=”时为等值联接。而自然联接是去掉重复属性的等值联接。

下面通过实例说明上述 3 种操作，已知关系 R 代表学生信息表，关系 S 代表学生成绩表，如表 1-7 和表 1-8 所示。

表 1-7　关系 R

学号	姓名	学院
2006010101	刘明	电子工程
2006010102	李海堂	电子工程
2006010103	王小华	电子工程
2006010104	郭磊	电子工程

表 1-8　关系 S

学号	数学	英语
2006010101	95	89
2006010102	85	90
2006010103	88	95
2006010104	89	89

关系 S 中满足“英语成绩大于或等于 90 分”的选择操作，结果如表 1-9 所示。

表 1-9　选择操作

学号	数学	英语
2006010102	85	90
2006010103	88	95

关系 S 在学号、英语两个属性上的投影操作，结果如表 1-10 所示。

表 1-10　投影操作

学号	英语
2006010101	89
2006010102	90
2006010103	95
2006010104	89

关系 R 和关系 S 的自然联接，结果如表 1-11 所示。

表 1-11　自然联接

学号	姓名	学院	数学	英语
2006010101	刘明	电子工程	95	89
2006010102	李海堂	电子工程	85	90
2006010103	王小华	电子工程	88	95
2006010104	郭磊	电子工程	89	89

可以看出，选择运算是在一个关系中进行水平方向的选择，选取的是满足条件的所有

元组。投影运算是在一个关系中进行垂直方向的选择，选取关系中元组的某几列的值。

1.3.3 关系完整性约束

关系完整性约束是为保证数据库中数据的正确性和相容性，对关系模型提出的某种约束条件或规则。关系模型中有三类完整性约束：实体完整性、参照完整性和用户定义完整性，其中实体完整性和参照完整性是关系模型必须满足的完整性约束条件。

1. 实体完整性

关系的主键可以标识关系中的每条记录，而关系的实体完整性要求记录关键字的字段不能为空，不同记录的关键字字段值也不能相同，否则，关键字就失去了唯一标识记录的作用。

例如，学生选课关系“学生选课（学号，课程编号，成绩）”中，“学号、课程编号”为主关键字，则“学号”和“课程编号”都不能取空值，否则无法对应某个具体的学生的课程，这样的表格不完整，对应关系不符合实体完整性规则的约束条件。

2. 参照完整性

参照完整性是相关联的两个表之间的约束，要求关系中“不引用不存在的实体”。

对于具有主从关系的两个表来说，表中每条记录外键的值必须是主表中存在的，如果在两个表之间建立了关联关系，则对一个关系进行的操作要影响到另一个表中的记录。

例如，在“专业表”和“学生表”之间用“专业码”建立了关联关系，“专业表”是主表，“学生表”是从表，“学号”是学生表的主键，是选课表的外键，那么在向从表添加新记录时，系统要检查新记录的学号是否在主表中已存在，如果存在则允许执行输入操作，否则拒绝输入。

3. 用户定义完整性

不同的关系数据库系统根据其应用环境的不同，通常需要针对某一具体字段的数据设置约束条件。

例如，若学生成绩为百分制，则成绩字段的取值必须在 0～100 之间等。

1.4 SQL 基本命令

1.4.1 SQL 简介

SQL（Structrued Query Language）是结构化查询语言，是操作数据库的通用语言。SQL 由数据定义语言、数据操纵语言、数据查询语言和数据控制语言（用于设置用户访问数据库的权限）4 部分组成。在 Access 中，可以使用 SQL 的前 3 种语言。

SQL 语句并不多，但每条语句的功能都非常强大，有些 SQL 语句的结构也比较复杂。目前，各种大、中、小型关系数据库管理系统都支持 SQL，但不同数据库管理系统中支持的 SQL 语句存在差异。

1.4.2　SQL 基本语句

1. 数据定义语言

数据定义语言用于建立（CREATE）、删除（DROP）数据库表及修改（ALTER）数据库表的结构。

（1）建立数据库表。

语句格式：

CREATE TABLE <表名>
(<字段名 1><类型描述>　[[NOT] NULL] [PRIMARY KEY]
……
[,<字段名 n><类型描述>　[[NOT] NULL] [PRIMARY KEY]
[<其他参数>]
[,PRIMARY KEY (<字段名表>)]]
……
);

语句说明：此语句用于建立数据库表，具体说明如下：

- 表名：用于指出新建立的数据库表名称，每个表中可以有多个属性（字段）名。
- 字段名 1，…，字段名 n：表中各个字段的名称，通常是由英文字母或汉字开头，由英文字母、汉字、数字和下划线组成的字符串。字段用于存储表中的各个属性值。
- 类型描述：用于描述字段的数据特征，如存储数据的类型和最大宽度，常用书写格式为“<数据类型符号>[(<最大宽度>)]”。

常用的数据类型符号有——文本型：Char；双精度型：Number；长整型：Long；整型：Integer；日期型：Date；逻辑型：Logical；备注型：Memo。对文本型数据需要说明最大长度。

- [NOT] NULL：在输入数据时，Not Null（默认）表示该字段值不能空，而 Null 表示该字段值可以空。
- PRIMARY KEY：指明对应字段为表的关键字。在一个表中只能有一个主关键字。
- PRIMARY KEY(<字段名表>)：当多个字段组成主关键字时，不能在每个字段后写“PRIMARY KEY”，只能在所有字段描述之后写“PRIMARY KEY (<字段名表>)”。其中，字段名表是逗号分隔的多个字段名。
- 其他参数：用于定义数据有效性规则（Check）等信息。

例如，建立“课程设置”表的 SQL 语句为：

CREATE TABLE 课程设置(课程编号 Char(5)PRIMARY KEY,课程名称 Char(30),开课学期 Integer,理论学时 Integer,实验学时 Integer,学分 Integer);

（2）修改数据库表结构。

语句格式：

ALTER TABLE　<表名>　ADD <字段名><类型描述>|

ALTER <字段名><类型描述>|

DROP <字段名>;

语句说明：此语句可以在表中增加（ADD）新字段，以及修改（ALTER）表中已有字段的类型描述（数据类型和字段宽度）或删除（DROP）已有的字段名。

例如，向“教室”表中增加“负责人”字段的 SQL 语句为：

```
ALTER TABLE 教室 ADD 负责人 Char(6);
```

将“教室”表的“星期”字段的数据类型改为整型的 SQL 语句为：

```
ALTER TABLE 教室 ALTER 星期 Integer;
```

删除“教室”表中“负责人”字段的 SQL 语句为：

```
ALTER TABLE 教室 DROP 负责人;
```

（3）删除表。

语句格式：

DROP TABLE <表名>;

语句说明：此语句用于删除数据库表。

例如，删除数据库表“Test”的 SQL 语句为：

```
DROP TABLE Test;
```

2. 数据操纵语言

数据操纵语言用于完成数据库表中数据记录的增加（INSERT）、删除（DELETE）和修改（UPDATE）操作。

（1）增加数据记录。

语句格式：

INSERT INTO <表名> [(<字段名表>)] VALUES (<表达式表>);

语句说明：此语句在指定表的尾部追加新记录。“字段名表”指出要增加的记录的各个字段名，用“表达式表”中各个表达式值填写对应的字段值，表达式与字段按前后顺序一一对应，并且，表达式值的数据类型必须与对应字段的数据类型一致。如果省略“字段名表”，则表示要增加表中的所有字段值，并按表中字段顺序与表达式一一对应。

例如，向“课程设置”表中增加数据记录的 SQL 语句为：

```
INSERT INTO 课程设置(课程编号,开课学期,理论学时,实验学时,学分)
VALUES (′01004′,2,70,20,4);
```

（2）修改数据记录。

语句格式：

UPDATE <表名> SET <字段名 1>=<表达式 1>

[…,<字段名 n>=<表达式 n>][WHERE <条件>];

语句说明：执行此语句时，用表达式值修改对应的字段值。如果省略 WHERE 选项，则修改表中的全部记录；如果使用“WHERE <条件>”，则仅修改那些使“条件”值为“真”（True）的记录。

例如，为“选课学生”表中的所有记录的“成绩”字段输入数据“0”的 SQL 语句为：

```
UPDATE 选课学生 SET 成绩=0;
```

（3）删除数据记录。

语句格式：

DELETE　FROM　<表名>　[WHERE <条件>]；

语句说明：使用此语句时，如果省略 WHERE 选项，则删除表中的全部记录；如果使用“WHERE <条件>”，则仅删除那些满足“条件”的记录。

例如，删除“选课学生”表中学号的第 3 位和第 4 位（年级）等于 02 的所有记录的 SQL 语句为：

```
DELETE　FROM 选课学生 WHERE　MID(学号,3,2)=′02′;
```

3. 数据查询语言

数据查询语言通过 SELECT 语句对数据进行查询、排序、汇总和表联接等输出操作。

语句格式：

SELECT [DISTINCT] *|<表达式 1>[AS <别名 1>]
[,…,<表达式 n> [AS <别名 n>]]
FROM　<表名 1>[,…,<表名 n>]
[WHERE <条件>]
[ORDER BY <排序关键字>　[ASC|DESC]]
[GROUP BY <分组字段> [HAVING <条件>]];

语句说明：执行此语句时，将数据库表中满足“WHERE <条件>”的数据记录按各个表达式进行计算，形成结果表。

- 表达式

“*|<表达式 1> [,…,<表达式 n>]”用于对表进行投影操作。系统按各个表达式对数据进行计算，得到相关数据项的值，表达式 i 可以是一个字段名。用星号“*”表示输出表中的所有字段。

- 别名

<别名 1> [,…,<别名 n>]为输出表达式值指定的对应列名。如果省略此项，当表达式是一个字段名时，字段名即为列名；当表达式是较复杂的表达式时，系统自动指定列名。

例如，输出“任课教师”表中所有记录的各个字段值的 SQL 语句为：

```
SELECT * FROM 任课教师;
```

- DISTINCT

在默认情况下，输出数据可能有重复行（对应字段值相同）。如果使用 DISTINCT，则对那些重复的数据行仅输出其中一行。

例如，输出“教室”表中的教室编号和名称，且不许出现重复的数据行的 SQL 语句为：

```
SELECT　DISTINCT　教室编号，名称　FROM 教室；
```

- FROM

FROM 之后可以使用多个表名，表名之间用逗号“,”分开，用于指出数据来源，即从哪些表中提取要操作的数据。特别是对多个表联接时，需要在此说明联接的表名。

- WHERE

“WHERE <条件>”不仅用于说明选择数据记录的条件，也用于设置多个表的联接条件。

例如，输出教师编号为“0001”的教师授课信息的 SQL 语句为：

```
SELECT 教室编号,名称,课程编号,教师编号,课节 FROM 教室 WHERE 教师编号＝"0001";
```

- ORDER BY

ORDER BY 用于说明输出结果数据的排序关键字。排序关键字可以是单独字段，也可以是表达式。系统默认输出结果数据按排序关键字的值升序（ASC）排列，也可以使用 DESC 使输出结果数据按排序关键字的值降序排列。

例如，输出“学生成绩”表中所有记录的学号、姓名和成绩，并且按成绩降序排序的 SQL 语句为：

```
SELECT 学号,姓名,成绩 FROM 学生成绩 ORDER BY 成绩 DESC;
```

- GROUP BY

GROUP BY 用于说明数据分组的关键字段，分组字段值相同的数据记录汇总成一行输出。“HAVING <条件>”指出仅输出那些符合“条件”的分组行。

1.5 数据库设计与管理

数据库设计是数据库应用的核心。数据库设计是指对于一个给定的应用环境，构造最优的数据模式，建立数据库及其应用系统，有效地存储数据，满足用户信息要求和处理要求。数据库设计要与整个数据库应用系统的设计开发结合起来，只有设计出高质量的数据库，才能开发出高质量的数据库应用系统。同时，只有统观整个数据库应用系统的功能需求，才能设计出高质量的数据库。针对一个具体问题，应该如何构造一个符合实际的恰当的数据模式，即应该构造几个关系，每个关系应该包括哪些属性，各个元组的属性值应符合什么条件等，这些都是应当全面考虑的问题。在关系数据库设计中要遵守数据库的关系完整性约束和数据库规范化设计。

数据库的关系完整性约束已在 1.3 节中介绍，下面介绍数据库规范化设计。

1.5.1 数据库规范化设计

在数据库设计中如何把现实世界表示成合理的数据库模式，一直是人们非常重视的问题。关系数据库的规范化理论就是进行数据库设计时的有力工具。

关系数据库中的关系要满足一定的条件要求，满足一定条件的关系模式称为范式（Normal Form）。在 1971 年至 1972 年，关系数据模式的创始人 E.F.Codd 系统地提出了第一范式（1NF）、第二范式（2NF）和第三范式（3NF）的概念。1974 年 Codd 和 Boyce 共同提出了 BCNF 范式，为第三范式的改进，还有第四范式和第五范式。一个低级范式的关系模式通过投影分解的方法可以转换成多个高一级范式的关系模式的集合，这个过程称为规范化。

目前遵循的主要范式包括第一范式（1NF）、第二范式（2NF）、第三范式（3NF）和第

四范式（4NF）等。规范化设计的过程就是按不同的范式，将一个二维表不断地分解成多个二维表并建立表之间的关联，最终达到一个表只描述一个实体或者实体间的一种联系的目标。其目的是减少冗余数据，提供有效的数据检索方法，避免不合理的插入、删除、修改等操作，保持数据一致性，增强数据库的稳定性、伸缩性和适应性。下面介绍第一范式（1NF）、第二范式（2NF）和第三范式（3NF）。

1. 第一范式（1NF）

关系中每一个数据项必须是不可再分的，满足这个条件的关系模式就属于第一范式。关系数据库中的所有数据表都必然满足第一范式。

例如，将如表 1-12 所示的学生成绩表规范为满足第一范式的表。

表 1-12　学生成绩表

学号	姓名	课程代码	课程名称	学分	成绩		
					平时成绩	考试成绩	总成绩
20150101	李明	41412076	高数	5	25	60	85
20150102	王丽	41232456	英语	5	26	65	91
…	…	…	…	…	…	…	…

处理方法是处理表头使其成为只有一行表头标题的数据表，如表1-13所示。

表 1-13　处理成满足第一范式的学生成绩表

学号	姓名	课程代码	课程名称	学分	平时成绩	考试成绩	总成绩
20150101	李明	41412076	高数	5	25	60	85
20150102	王丽	41232456	英语	5	26	65	91
…	…	…	…	…	…	…	…

2. 第二范式（2NF）

在一个满足第一范式的关系中，如果所有非主属性都完全依赖于主键，则称这个关系满足第二范式。即对于满足第二范式的关系，如果给定一个主键，则可以在这个数据表中唯一确定一条记录。一个关系模式如果不满足第二范式，就会产生插入异常、删除异常、修改复杂等问题。

例如，在学生选课系统中构造如表 1-14 所示的数据表，表中没有哪一个数据项能够唯一标识一条记录，则不满足第二范式。

表 1-14　学生选课综合信息表

学号	姓名	院系	课程代码	课程名称	学分	成绩	任课教师	职称
20150001	范玉	计算机	11	C 语言	5	78	周伟	教授
20150001	范玉	计算机	12	JAVA	5	86	丁立	副教授
20150001	范玉	计算机	14	数据库技术	4	79	张欣悦	副教授

续表

学号	姓名	院系	课程代码	课程名称	学分	成绩	任课教师	职称
20150002	刘冰	计算机	13	计算机网络	4	90	孙一	教授
20150003	孙贺	计算机	11	C 语言	5	90	周伟	教授
20150003	孙贺	计算机	12	JAVA	5	88	丁立	副教授

该数据表存在如下缺点：

（1）冗余度大。一个学生如果选修 n 门课，则它的有关信息就要重复 n 遍，这就造成数据的极大冗余。

（2）插入异常。在这个数据表中如果要插入一门课程的信息，但此门课程本学期不开设，目前无学生选修，则很难将其插入表中。

（3）删除异常。表中刘冰只选了一门课“计算机网络”，如果他不选了，这条记录就要被删除，那么整个元组都随之删除，使得他的所有信息都被删除了，造成删除异常。

处理表 1-14 使之满足第二范式的方法是将其分解成三个数据表，如表 1-15、表 1-16 和表 1-17 所示。这三个表即为满足第二范式的表。其中“学生信息表”的主键为“学号”，“课程设置表”的主键为“课程代码”，“选课表”的主键为“学号和课程代码”。

表 1-15　学生信息表

学号	姓名	院系
20150001	范玉	计算机
20150002	刘冰	计算机
20150003	孙贺	计算机

表 1-16　课程设置表

课程代码	课程名称	学分	任课教师	职称
11	C 语言	5	周伟	教授
12	JAVA	5	丁立	副教授
13	计算机网络	4	孙一	教授
14	数据库技术	4	张欣悦	副教授

表 1-17　选课表

学号	课程代码	成绩
20150001	11	78
20150001	12	86
20150001	14	79
20150002	13	90
20150003	11	90
20150003	12	88

3. 第三范式（3NF）

对于满足第二范式的关系，如果每一个非主属性都不传递依赖于主键，则称这个关系满足第三范式。传递依赖就是某些数据项间接依赖于主键。在如表 1-16 所示的表中，职称属于任课教师，主键“课程代码”不直接决定非主属性“职称”，“职称”是通过“任课教师”传递依赖于“课程代码”的，则此关系不满足第三范式，在某些情况下，会存在插入异常、删除异常和数据冗余等现象。为将此关系处理成满足第三范式的数据表，可以将其分成“课程信息表”和“任课教师表”，如表 1-18 和表 1-19 所示。

表 1-18　课程信息表

课程代码	课程名称	学分
11	C 语言	5
12	JAVA	5
13	计算机网络	4
14	数据库技术	4

表 1-19　任课教师表

学分	任课教师	职称
5	周伟	教授
5	丁立	副教授
4	孙一	教授
4	张欣悦	副教授

经过规范化处理，满足第一范式的“学生选课综合信息表”被分解成满足第二范式和第三范式的 4 个表（学生信息表、课程信息表、任课教师表和选课表）。

1.5.2　数据库设计的步骤

目前，数据库设计一般采用生命周期法。数据库系统的生命周期是指数据库系统从分析、设计、实现、运行及维护，直到被新的系统所取代而停止使用的整个期间。

从这个角度来讲，数据库设计的步骤一般分为需求分析、概念结构设计、逻辑结构设计、物理结构设计、数据库实施、数据库运行和维护 6 个阶段。

1. 需求分析

设计一个数据库，首先必须准确、全面和深入地了解和分析用户需求，包括数据需求和处理需求。需求分析是整个设计活动的基础，也是最困难、最花时间的一步。需求分析人员既要懂数据库技术，又要对应用环境的业务熟悉，一般由数据库专业人员与业务专家合作进行。

2. 概念结构设计

概念设计是在需求分析基础上，用概念数据模型（如 E-R 模型）表示数据及其相互间

的联系，形成数据库概念结构（如 E-R 图）。概念结构与具体的 DBMS 无关，是对现实世界的可视化描述，容易被用户所理解，因而不但可用于后续的设计，也是与用户交流和数据库移植的重要文档。

3. 逻辑结构设计

逻辑设计是将数据库概念结构转换为某类 DBMS 所支持的数据库逻辑模式，例如，将 E-R 图转换为关系模型所支持的关系数据库模式。逻辑设计也不仅仅是个数据模式的转换问题，还要进一步考虑数据模式的规范化、满足 DBMS 的各种限制等，还得为各类用户或应用设计其各自的局部逻辑模式，即外模式或子模式。逻辑设计的结果，即数据库逻辑模式，是以数据定义语言（DDL）来表示，在 SQL 中，就是编写 CREATE TABLE、CREATE VIEW 等命令。

4. 物理结构设计

物理设计的任务是根据 DBMS 及计算机系统所提供的手段，为数据库逻辑模式选取一个最适合应用环境的物理模式（包括存储结构和存取方法等），以提高数据库访问速度及有效地利用存储空间。

5. 数据库实施

数据库的管理主要是指数据库的实施和维护。数据库实施就是在实际的计算机平台上，真正建立数据库。先运行用 DDL 编写的命令，建立数据库框架，然后通过 DBMS 的实用工具或专门编写的应用程序，将数据实际载入，最终建成数据库。在数据库投入使用之前，要进行测试和试运行。除单独测试之外，还要与数据库应用程序结合起来进行测试。

6. 数据库运行和维护

数据库经过试运行后就可以投入实际运行了。但是，由于应用环境在不断变化，对数据库设计进行评价、调整、修改等维护工作是一项长期的任务，也是设计工作的继续和提高。在数据库运行阶段，由数据库管理员进行数据库的转储和恢复、数据库的安全性和完整性控制、数据库性能的监督和分析、数据库的重组织与重构造等数据库的维护工作。

1.5.3 数据库的管理

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，而实施此项管理的人则称为数据库管理员。数据库管理一般包含如下内容：数据库的建立、数据库的调整、数据库的重组、数据库的安全性控制与完整性控制、数据库的故障恢复和数据库的监控。

1. 数据库的建立

数据库的建立包括两部分内容：数据模式的建立和数据加载。

（1）数据模式的建立。数据模式由 DBA 负责建立，DBA 利用关系数据库管理系统中的 DDL 语言定义数据库名，定义表及相应属性，定义主关键字、索引、集簇、完整性约束、用户访问权限，申请空间资源，定义分区等，此外还需定义视图。

（2）数据加载。在数据模式定义后即可加载数据，DBA 可以编制加载程序将外界数据加载至数据模式内，从而完成数据库的建立。

2. 数据库的调整

在数据库建立并经过一段时间运行后往往会产生一些不适应的情况，此时需要对其作调整，数据库的调整一般由 DBA 完成，调整包括下面一些内容：

（1）调整关系模式与视图使之更能适应用户的需求。

（2）调整索引与集簇使数据库性能与效率更佳。

（3）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。

3. 数据库的重组

数据库在经过一定时间运行后，其性能会逐渐下降，下降的原因主要是由于不断地修改、删除与插入造成的。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。基于这些原因需要对数据库进行重新整理，重新调整存储空间，此种工作叫做数据库重组。一般数据库重组需花大量时间，并做大量的数据搬迁工作。实际中，往往是先做数据卸载，然后重新加载从而达到数据重组的目的。目前一般关系数据库都提供一定手段，以实现数据重组功能。

4. 数据库安全性与完整性控制

数据库是一个单位的重要资源，它的安全性是极端重要的，DBA 应采取措施保证数据不受非法盗用与破坏。此外，为保证数据的正确性，使录入库内的数据均能保持正确，需要有数据库的完整性控制。

5. 数据库的故障恢复

一旦数据库中的数据遭到破坏，需要及时进行恢复，RDBMS 一般都提供此种功能，并由 DBA 负责执行故障恢复功能。

6. 数据库监控

DBA 需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，同时还需监视数据库的性能变化，在必要时对数据库作调整。

1.5.4　Access 数据库应用系统设计实例

按照规范化理论和完整性规则设计出能够正确反映现实应用的数据模型后，还要进行系统功能的设计。对于系统功能设计应遵循自顶向下、逐步求精的原则，将系统必备的功能分解为若干相互独立又相互依存的模块，每一模块采用不同的技术，解决不同的问题，从而将问题局部化，这是数据库设计中的分步设计法。这里以一个学生成绩管理系统为例，简单介绍数据库系统开发的方法。

1. 需求分析

这是数据库应用系统开发的第一步。首先详细调查要处理的对象，明确用户的各种要求，在此基础上确定数据库中需要存储哪些数据及系统需要具备哪些功能等。设计人员必须不断深入地与用户交流，才能逐步确定用户的实际需求，以确定设计方案。对于学生成绩管理系统来说，进行需求分析后，得到以下结果：

（1）用户需要完成数据的录入。学校开设新课程、新生入学、增加新院系、增加新教师、重新选课、统计期末成绩时都需要进行数据录入并提交数据库保存。因此系统要包括以下数据表：院系表、专业表、教师信息表、学生信息表、课程设置表、学生选课表、学生成绩表。

（2）完成数据的修改。当学生、教师、课程等情况发生变化或数据录入错误时，用户要进行数据的修改，以保证表中数据的正确性。

（3）实现信息的查询。包括学生成绩查询、学生信息查询、学生选课查询和课程查询等。

2. 应用系统的数据库设计

这是在需求分析的基础上进行的。首先要弄清需要存储哪些数据，确定需要几个数据表，每一个表中包括几个字段，并确定主键，然后在 Access 中建立数据表。这一过程中要严格遵循关系数据库完整性和规范化设计要求。学生成绩管理系统要创建 7 个数据表：

（1）院系表（院系代码、院系名称）。

（2）专业表（专业代码、专业名称、所属院系代码、所属院系名称）。

（3）教师信息表（教师编号、教师姓名、所属院系名称、所属专业名称）。

（4）学生信息表（学号、姓名、性别、出生日期、民族、政治面貌、院系、专业、班级、籍贯、电话、照片、备注）。

（5）课程设置表（课程代码、课程名称、学时、学分、性质、教师编号、教师姓名、开课学院、开课学期）。

（6）学生选课表（学号、姓名、课程代码、课程名称、学分）。

（7）学生成绩表（学号、姓名、课程代码、课程名称、学分、成绩）。

划线部分为各数据表的主键，各数据表的结构如表 1-20 至表 1-26 所示。

表 1-20　院系表

字段名称	数据类型	字段大小
院系代码	文本	8
院系名称	文本	30

表 1-21　专业表

字段名称	数据类型	字段大小
专业代码	文本	8
专业名称	文本	30
所属院系代码	文本	8
所属院系名称	文本	30

表 1-22　教师信息表

字段名称	数据类型	字段大小
教师编号	文本	8
教师姓名	文本	12
所属院系名称	文本	30
所属专业名称	文本	30

表 1-23　学生信息表

字段名称	数据类型	字段大小
学号	文本	10
姓名	文本	12
性别	文本	2
出生日期	日期/时间	
民族	文本	16
政治面貌	文本	8
院系	文本	30
专业	文本	30
班级	文本	10
籍贯	文本	30
电话	文本	15
照片	OLE 对象	
备注	备注	

表 1-24　课程设置表

字段名称	数据类型	字段大小
课程代码	文本	10
课程名称	文本	30
学时	数字	整型
学分	数字	整型
性质	文本	10
教师编号	文本	8
教师姓名	文本	12
开课学院	文本	30
开课学期	文本	30

表 1-25　学生选课表

字段名称	数据类型	字段大小
学号	文本	10
姓名	文本	12
课程代码	文本	10
课程名称	文本	30
学分	数字	整型

表 1-26　学生成绩表

字段名称	数据类型	字段大小	小数位数
学号	文本	10	
姓名	文本	12	
课程代码	文本	10	
课程名称	文本	30	
学分	数字	整型	
成绩	数字	单精度型	1

3. 应用系统的功能设计

根据需求分析，结合初步设计的数据库模型，设计应用系统的各个功能模块。学生成绩管理系统中有 8 个功能模块。

（1）院系管理：包括院系的设置及相关信息查询。

（2）专业设置：包括专业的设置及相关信息查询。

（3）教师信息：包括教师信息的建立、修改及查询。

（4）学生信息：包括学生信息的建立、修改及查询。

（5）课程管理：包括课程设置及相关信息查询。

（6）选课管理：包括学生选课系统及选课信息查询。

（7）成绩管理：包括学生成绩的录入、修改及查询。

（8）系统管理：包括系统使用权限的设置、系统的说明和退出系统等。

4. 系统的性能分析

软件初步形成后，需要对它进行性能分析，如果有不完善的地方，需要根据分析结果对数据库进行优化，直到应用软件的设计满足用户的需要为止。

5. 系统的发布与维护

系统经过调试满足用户的需要后就可以进行发布，但在使用过程中可能还会存在某些问题，因此在软件运行期间要进行调整，以实现软件性能的改善和扩充，使其适应实际工作的需要。

习题一

一、选择题

1．数据库中存储的是（　　）。

A．数据　　B．数据模型

C．数据之间的联系　　D．数据以及数据之间的联系

2．下列叙述中正确的是（　　）。

A．数据库是一个独立的系统，不需要操作系统的支持

B．数据库设计是指设计数据库管理系统

C．数据库技术的根本目标是要解决数据共享的问题

D．数据库系统中，数据的物理结构必须与逻辑结构一致

3．DB、DBMS 和 DBS 三者之间的关系是（　　）。

A．DB 包括 DBMS 和 DBS　　B．DBS 包括 DB 和 DBMS

C．DBMS 包括 DBS 和 DB　　D．DBS 与 DB 和 DBMS 无关

4．数据库管理系统属于（　　）。

A．系统软件　　B．应用软件

C．编译软件　　D．操作系统

5．单个用户使用的数据视图的描述称为（　　）。

A．外模式　　B．概念模式

C．内模式　　D．存储模式

6．数据库系统体系结构的三级模式间存在二级映射，它们是（　　）。

A．概念模式与子模式间，概念模式与内模式间

B．子模式与内模式间，外模式与内模式间

C．子模式与外模式间，概念模式与内模式间

D．概念模式与内模式间，外模式与内模式间

7．概念模型是描述（　　）的数据模型。

A．现实世界　　B．信息世界

C．计算机世界　　D．网络世界

8．将 E-R 图转换到关系模型时，实体和联系都可以表示为（　　）。

A．属性　　B．关系

C．键　　D．域

9．使用二维表表示实体及实体之间联系的数据模型是（　　）。

A．实体－联系模型　　B．层次模型

C．关系模型　　D．网状模型

10．在关系数据库中，用来表示实体之间联系的是（　　）。

A．树结构　　B．网结构

C．线性表　　D．二维表

11．关系中的每一横行称为一个（　　）。

A．元组　　B．字段

C．属性　　D．键

12．从关系中挑选出指定的属性组成新关系的运算称为（　　）。

A．选择　　B．投影

C．联接　　D．检索

13．数据库设计包括两个方面的设计内容，它们是（　　）。

A．概念设计和逻辑设计　　B．模式设计和内模式设计

C．内模式设计和物理设计　　D．结构特性设计和行为特性设计

14．从关系中选取满足某种条件的元组的操作称为（　　）。

A．选择　　B．投影

C．联接　　D．检索

15．关系数据库中的关系要满足一定的条件要求，满足一定条件的关系模式称为（　　）。

A．概念模式　　B．外模式

C．内模式　　D．范式

二、填空题

1．在数据管理技术的发展过程中，经历了人工管理阶段、文件系统阶段和数据库系统阶段。其中数据独立性最高的阶段是_____。

2．数据库系统中实现各种数据管理功能的核心软件称为_____。

3．数据库系统的三级模式分别为_____模式、内模式与外模式。

4．数据独立性分为逻辑独立性与物理独立性，当数据的存储结构改变时，其逻辑结构可以不变，因此，基于逻辑结构的应用程序不必修改，这称为_____独立性。

5．实现概念模型最常用的表示方法是_____。

6．如果一个工人可管理多个教室，而一个教室只被一个工人管理，则实体“工人”与实体“教室”之间存在_____的联系。

7．在关系数据库中，一个属性的取值范围称为_____。

8．关系的完整性约束包括_____、_____和_____。

9．关系数据库管理系统能实现的专门关系运算，包括选择、_____和联接。

10．设关系 R 有 P 行 M 列，关系 S 有 Q 行 N 列，则 R×S 有_____行_____列。

11．关系数据库的逻辑设计阶段的任务是将 E-R 模型转换为_____模型。

12．满足_____条件的关系模式就属于第一范式。

三、问答题

1．什么是数据库？什么是数据库系统？
2．什么是数据的独立性？
3．什么是实体？两个实体间的联系有哪几种类型？
4．数据模型有几种？
5．什么是联接？什么是自然联接？它和等值联接有什么不同？
6．什么是关系的完整性约束？每种约束的含义是什么？
7．简述第一范式、第二范式和第三范式。
8．数据库的设计分为几个阶段？