

项目一

数据库的基础知识

项目导入

某高校日常管理工作的效率低下，教务科工作人员每学期在烦琐的纸质表格中更新、查询学生课程数据。为提高学校日常管理工作的效率，技术部门成立了信息管理工作小组，要求信息管理员王明设计出一个系统，使学生和教师可以通过该系统完成信息的查询和修改。

项目描述

- (1) 你听说过“数据库”吗？在日常生活和工作中，你是否用到了数据库？
- (2) 20 世纪的人类是如何管理数据的？现在又是如何管理数据的呢？
- (3) 人类是通过什么定义、操纵数据库中海量数据的？
- (4) 本书将研究哪种类型的数据库？它能给工作和生活带来什么样的变化？
- (5) 从不同的角度看，数据库系统结构各有什么特征？

教学导航

- (1) 掌握：数据库、数据库系统和数据库管理系统的概念，以及数据模型的组成要素。
- (2) 理解：概念结构设计、逻辑结构设计和数据库物理设计。
- (3) 了解：信息、数据与数据处理的概念，数据库系统的产生和发展，以及关系数据库的理论。

知识准备

1.1 数据库的基本概念

随着计算机技术的发展，信息技术的应用日益广泛，管理信息资源的数据库技术也得到迅速发展，应用范围涉及管理信息系统、专家系统、过程控制、联机分析处理等各个领域。数据库技术已成为计算机信息系统与应用系统的核心技术和重要基础，以及衡量社会信息化程度的重要标志。

1.1.1 数据、信息与数据处理

数据是数据库中存储的基本对象,是可以被计算机接受并能够被计算机处理的符号。数据的表现形式多样化,可以是数字、文字、图形、图像、声音等信息。例如,定义学生的姓名为“王明”,性别为“女”,年龄为“20”,“王明”“女”和“20”都是数据。

信息是对数据的解释,是经过加工处理后具有一定含义的数据集合,它具有超出事实数据本身之外的价值,能提高人们对事物认识的深刻程度,对决策或行为有现实或潜在的价值。

数据与信息既有联系又有区别。数据是信息的表现形式,信息是加工处理后的数据,是数据表达的内容。相同的数据可以因载体的不同表现出不同的形式,信息则不会随信息载体的不同而改变。

将数据转换成信息的过程称为数据处理,是指利用计算机对原始数据进行科学的采集、整理、存储、加工和传输等一系列活动,从繁杂的数据中获取所需的资料和有用的数据。

1.1.2 数据库、数据库系统、数据库管理系统

1. 数据库

数据库可以理解为存放数据的仓库,它是以一定的方式将相关数据组织在一起并存储在外存储器上,能为多个用户共享、与应用程序彼此独立的一组相互关联的数据集合。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。

2. 数据库系统

数据库系统(database system, DBS)是由数据库及其管理软件组成的系统,它是一种为适应数据处理需要而发展起来的较为理想的数据处理的核心机构,能够有组织、动态地存储大量数据,提供数据处理和数据共享机制,是存储介质、处理对象和管理系统的集合体。

3. 数据库管理系统

数据库管理系统(database management system, DBMS)是处理数据访问的软件系统,它位于用户与操作系统之间,用户必须通过数据库管理系统统一管理和控制数据库中的数据。

数据库管理系统的功能主要包括以下6方面。

1) 数据定义功能

数据定义功能是数据库管理系统面向用户的功能,数据库管理系统提供数据定义语言(data definition language, DDL)定义数据库中的数据对象,包括三级模式及其相互之间的映像等,如数据库、基本表、视图的定义、数据完整性和安全控制等约束。

2) 数据操纵功能

数据操纵功能是数据库管理系统面向用户的功能，数据库管理系统提供数据操纵语言（data manipulation language, DML），用户可以使用 DML 对数据库中的数据进行各种操作，如存取、查询、插入、删除和修改等。DML 有两类：一类可以独立交互使用，不依赖任何程序设计语言，称为自主型或自含型 DML；另一类必须嵌入宿主语言中使用，称为宿主型 DML。在使用高级语言编写的应用程序中，需要使用宿主型 DML 访问数据库中的数据。

3) 数据库运行管理功能

数据库运行管理功能是数据库管理系统的运行控制和管理功能，包括多用户环境下的并发控制、安全性检查和存取限制控制、完整性检查和执行、运行日志的组织管理、事务的管理和自动恢复。这是数据库管理的核心部分，所有数据库的操作都要在这些控制程序的统一管理和控制下进行，这些功能保证了数据库系统的正常运行。

4) 数据维护功能

数据维护功能包括数据库数据的导入功能、转储功能、恢复功能、重新组织功能、性能监视和分析功能等，这些功能通常由数据库管理系统的应用程序提供给数据库管理员。

5) 数据库的传输功能

数据库管理系统实现数据的传输，实现用户程序与数据库管理系统之间的通信，通常与操作系统协调完成。

6) 数据通信接口

数据库管理系统需要提供与其他软件系统进行通信的功能。例如，提供与其他数据库管理系统或文件系统的接口，从而能够将数据转换为另一个数据库管理系统或文件系统能够接受的格式，或者接收其他数据库管理系统或文件系统的数据库。

1.2 数据库管理技术及发展

数据管理是指数据的收集、整理、组织、存储、检索、维护和传送等各种操作，是数据处理中的基本环节，是任何数据处理任务必须具有的共同部分。

1.2.1 数据管理技术的发展阶段

随着社会的不断进步，人类社会积累的信息正以几何级数的速度增长。人们过去传统、落后的数据处理方法已经远远适应不了形势发展的需要，人们对数据处理的现代化的要求日益迫切。计算机的数据管理技术大致经历了三个阶段。

1. 人工管理阶段

在计算机出现之前，人们运用常规的手段记录、存储和加工数据，也就是利用纸张



视频讲解

3

项目
一

来记录,利用计算工具进行计算,并主要利用人的大脑管理和利用这些数据。20世纪50年代中期以前,计算机主要用于数值计算,数据量较少,一般不需要长期保存。在硬件方面,外存只有卡片、磁带和纸带,还没有磁盘等直接存取的存储设备;在软件方面,没有专门管理数据的软件,数据的处理方式基本是批处理。人工管理阶段的数据与应用程序之间的关系是一一对应的,如图1-1所示。

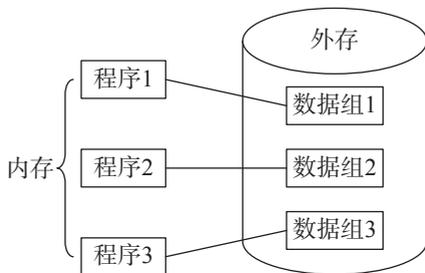


图 1-1 人工管理阶段数据与应用程序之间的关系

一组数据只对应一个应用程序。即使两个应用程序都涉及了某些相同的数据,也必须各自定义,无法相互利用和参照,数据无法共享,从而导致程序与程序之间存在大量的冗余。

2. 文件系统阶段

20世纪50年代后期至60年代中后期,计算机不仅用于科学计算,还用于信息管理。在硬件方面,外存有磁盘、磁鼓等直接存取的存储设备;在软件方面,操作系统中已经有了专门的数据管理软件,称为文件系统。数据处理方式有批处理和联机实时处理两种。文件系统阶段的数据与应用程序之间的关系如图1-2所示。

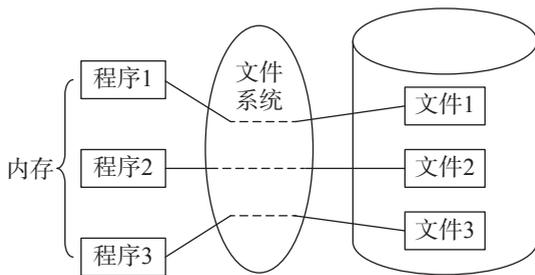


图 1-2 文件系统阶段数据与应用程序之间的关系

虽然文件系统阶段较人工管理阶段有了很大的改进,但仍显露出很多缺点。例如,由于应用程序的依赖性,因此导致编写应用程序很不方便;存储在文件中的数据如何存放由程序员自己定义,不统一,难于共享;数据冗余大,浪费了存储空间;不支持对文件的并发访问;文件间联系弱,必须通过应用程序来实现;难以按最终用户视图表示数据;无安全控制功能等。

3. 数据库系统阶段

20 世纪 60 年代后期, 计算机管理的范围越来越广泛, 数据量也急剧增加。在硬件方面, 计算机性能得到进一步提高, 出现了大容量磁盘, 存储容量大大增加且价格下降; 在软件方面, 操作系统更加成熟, 程序设计语言的功能更加强大。在此基础上, 数据库技术应运而生, 主要克服文件系统管理数据时的不足, 满足和解决实际应用中多用户、多种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。也因此出现了统一管理数据的专门软件系统, 即数据库管理系统。数据库系统阶段的数据与应用程序之间的关系如图 1-3 所示。

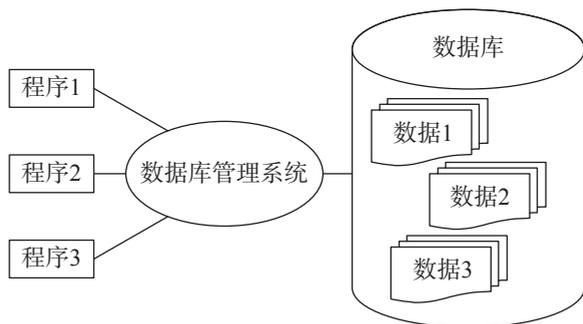


图 1-3 数据库系统阶段数据与应用程序之间的关系

1.2.2 数据库系统的特点

数据库系统的特点体现在以下 5 方面。

1. 数据共享

这是数据库系统区别于文件系统的最大特点之一, 也是数据库系统技术先进性的重要体现。共享是指多用户、多种应用、多种语言互相覆盖的共享数据集合, 所有用户可以同时存取数据库中的数据。数据库是面向整个系统的, 以最优的方式服务于一个或多个应用程序, 实现数据共享。

2. 数据结构化

在数据库中, 数据不再像文件系统那样从属于特定的应用, 而是按照某种数据模型组织成为一个结构化的整体。它不仅描述了数据本身的特性, 还描述了数据与数据之间的种种联系, 使数据库具备复杂的结构。

数据结构化有利于实现数据共享, 数据实现集中统一的存储与管理, 各种应用存取各自相关的数据子集, 满足各种应用要求。

3. 数据独立性

在文件系统管理中, 应用程序较依赖数据文件。如果把应用程序使用的磁带顺序文件改为磁盘索引文件, 则必须对应用程序进行修改。而数据库技术的重要特征就是数据



视频讲解

与程序相互独立，互不依赖，不因一方的改变而改变，这大大简化了应用程序的设计与维护的工作量。

4. 冗余度低、易扩充

在数据库中，数据共享减少了数据冗余造成的不一致现象。由于数据面向整个系统，它是有结构的数据，因此它不但可以被多个应用共享使用，而且增加新的应用变得容易，使数据库系统弹性大，易于扩充。

5. 统一数据控制功能

数据库是系统中各用户的共享资源，因而计算机的共享一般是并发的，即多个用户同时使用数据库。因此，数据库管理系统必须提供以下 4 方面的数据控制功能，保证整个系统的正常运转。

1) 数据安全性控制

数据安全性控制是指采取一定的安全保密措施以确保数据库中的数据不被非法用户存取。

2) 数据完整性控制

数据的完整性是指数据的正确性、有效性与相容性。系统要提供必要的功能，保证数据库中的数据在输入、修改过程中始终符合原来的定义和规定。

3) 并发控制

当多个用户并发进程同时存取、修改数据库中的数据时，可能会互相干扰而得到错误结果，使数据库完整性遭到破坏，因此必须对多用户的并发操作加以控制和协调。

4) 数据恢复

当系统出现故障或操作数据发生错误时，系统能进行应急处理，把数据库恢复到正确状态。

1.3 数据模型

模型是对现实世界中某个对象特征的模拟和抽象。数据模型与具体的数据库管理系统相关，可以说它是概念模型的数据化，是现实世界的计算机模拟。

1.3.1 数据模型的组成要素

数据模型通常有一组严格定义的语法，人们可以使用它定义、操纵数据库中的数据。数据模型的组成要素包括以下三部分。

1. 数据结构

数据结构是对系统静态特性的描述，是研究的对象和对象类型的集合。这些对象和对象类型是数据库的组成部分，一般可分为两类：一类是与数据类型、内容和其他性质有关的对象；另一类是与数据之间的联系有关的对象。

在数据库领域中，通常按照数据结构的类型命名数据模型，进而对数据库管理系统进行分类。例如，层次结构、网状结构和关系结构的数据模型分别称为层次模型、网状模型和关系模型，相应地，数据库分别称为层次数据库、网状数据库和关系数据库。

2. 数据操作

数据操作是对系统动态特性的描述，是指对各种对象类型的实例或值允许执行的操作的集合，包括操作及有关的操作规则。在数据库中，主要的操作有检索和更新（包括插入、删除、修改）两大类。数据模型定义了这些操作的定义、操作符号、操作规则和实现操作的语言。

3. 数据的完整性约束条件

数据的完整性约束条件是完整性规则的集合。完整性规则是指在给定的数据模型中，数据及其联系具有的制约条件和依存条件。它们用来限制符合数据模型的数据库的状态以及状态的变化，确保数据的正确性、有效性和一致性。

数据模型应该反映和规定符合本数据模型的必须遵守的基本、通用的完整性约束条件，还应该提供定义完整性约束条件的机制，用以反映特定的数据必须遵守特定的语义约束条件。

以上三个要素完整地描述了一个数据模型。数据模型不同，描述和实现方法也不同。

1.3.2 数据模型的类型

数据模型按不同的应用层次可分成概念模型、逻辑模型和物理模型三种类型。在概念模型中，最常用的是 E-R 模型、扩充的 E-R 模型、面向对象模型及谓词模型。在逻辑模型中，最常用的是层次模型、网状模型和关系模型。

1.3.3 概念模型

为了把现实世界中的具体事物抽象、组织为某一个数据库管理系统支持的数据模型，人们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具体的计算机系统，它不是某一个数据库系统支持的数据模型，而是概念级的模型，称为概念模型；然后再把概念模型转换为某一个计算机系统上的某一个数据库管理系统支持的数据模型。因此，概念模型是从现实世界到机器世界的一个中间层次。在概念模型中，经常使用一些概念或名词来描述数据结构，如实体、属性、域、实体型、实体集等。

(1) 实体：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，也可以是抽象的概念和联系。例如，一本书、一名学生、一台计算机等都是实体。

(2) 属性：实体具有的各个特性称为实体的属性。例如，学生的学号、姓名、性别、身高等都是学生实体的属性。



视频讲解

(3) 域：属性的取值范围称为该属性的域。例如，性别的域为（男，女）。

(4) 实体型：具有相同属性的实体称为同型实体，对于同型实体，可以用实体名及其属性名的集合来描述，称为实体型。例如，每名学生都具有学号、姓名、性别和身高属性，他们是同型实体，“学生（学号、姓名、性别、身高）”描述了学生这些同型实体，它是一个实体型。

(5) 实体集：同型实体的集合称为实体集。例如，所有学生就是一个实体集。

(6) 码：能够唯一标识实体集中每个实体的属性或属性集称为实体的码。例如，学号就是学生实体的码。

(7) 联系：在现实世界中，事物内部及事物之间存在普遍联系，这些联系在信息世界中表现为实体型内部各属性之间的联系以及实体型之间的联系。两个实体型之间的联系可分为以下三类。

- 一对一联系（1 : 1）：若对于实体集 A 中的每一个实体，实体集 B 中至多有一个（也可以没有）实体与之联系，反之亦然，则称实体集 A 与实体集 B 具有一对一联系，记为 1 : 1。例如，一个部门只有一个经理，而每个经理只在一个部门任职，则部门和经理之间具有一对一联系。
- 一对多联系（1 : n）：若对于实体集 A 中的每一个实体，实体集 B 中有 n 个实体（ $n \geq 0$ ）与之联系；反之，对于实体集 B 中的每一个实体，实体集 A 中至多只有一个实体与之联系，则称实体集 A 与实体集 B 有一对多联系，记为 1 : n 。例如，一个部门有若干职工，而每个职工只在一个部门上班，则部门与职工之间具有一对多联系。
- 多对多联系（ m : n ）：若对于实体集 A 中的每一个实体，实体集 B 中有 n 个实体（ $n \geq 0$ ）与之联系；反之，对于实体集 B 中的每一个实体，实体集 A 中也有 m 个实体（ $m \geq 0$ ）与之联系，则称实体集 A 与实体集 B 具有多对多联系，记为 m : n 。例如，一门课程同时有若干学生选修，而一个学生可以同时选修多门课程，则课程与学生之间具有多对多联系。

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。

概念模型描述实体、实体的属性、实体间的联系，它是现实世界的第一级抽象，反映现实世界客观事物及事物间的联系。概念模型的表示方法很多，最常用的是实体 - 联系（entity-relationship, E-R）方法，该方法用 E-R 图表示概念模型。

构成 E-R 图的基本要素是实体型、属性和联系，表示方法分别为：

- (1) 实体型：用矩形表示，矩形框内写明实体名。
- (2) 属性：用椭圆表示，椭圆内写明属性名，用无向边将属性与实体相连。
- (3) 联系：用菱形表示，菱形框内写明联系名，用无向边与有关实体连接起来，同时在无向边上注明联系类型。

E-R 图示例如图 1-4 所示。

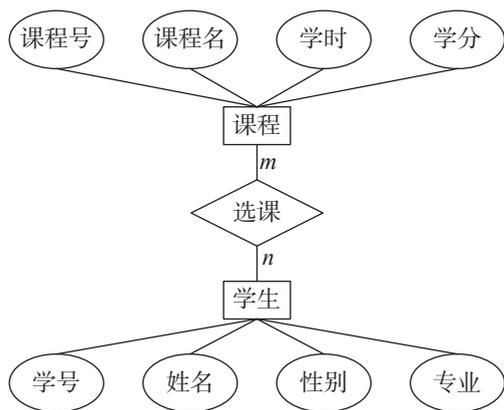


图 1-4 E-R 图示例

1.3.4 层次模型

层次模型是数据库系统中最早出现的数据模型，用树状结构表示实体之间的联系，这种结构方式反映了现实世界中数据的层次结构关系。

在现实世界中，许多实体之间的联系本身就是一种自然的层次结构关系，图 1-5 为某学院按层次模型组织的数据示例。

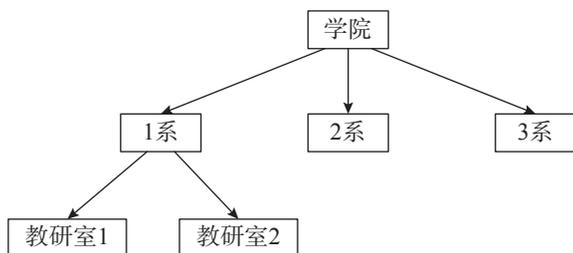


图 1-5 某学院按层次模型组织的数据示例

树中的每一个节点表示一个记录类型，连线表示双亲-子女关系。因此，层次模型实际上是以记录类型为节点的有向树。层次模型满足三个条件：有且仅有一个节点无双亲节点，称为根节点；根以外的其他节点有且仅有一个双亲节点；没有子女节点的节点，称为叶节点。

在层次模型中，由于是通过指针实现记录之间的联系，因此查询效率较高，其层次分明，结构清晰，不同层次间的数据关联直接简单。但其也存在着一定的缺点，由于从属节点有且只有一个双亲节点，因此它只能表示 $1:n$ 联系，虽然有各种辅助手段实现 $m:n$ 联系，但较复杂，用户不易掌握；数据将不得不纵向向外扩展，节点之间很难建立横向的关联；由于严格和复杂的层次顺序，导致数据的查询和更新操作很复杂，因此应用程序的编写也比较复杂。

1.3.5 网状模型

每一个数据用一个节点表示,每个节点与其他节点都有联系,这样,数据库中的所有数据节点就构成了一个复杂的网络。用网状结构表示实体及其联系的模型称为网状模型。

网络中的每一个节点表示一个记录类型,联系用链接指针来实现。网状模型满足两个条件:允许有一个以上的节点无双亲节点;一个节点可以有多个双亲节点。这样,在网状模型中任何两个节点都可以有联系,从而可以方便地表示各种类型之间的联系。图 1-6 为一个简单的城市之间的铁路交通联系的网状模型。

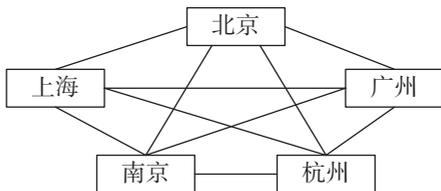


图 1-6 按网状模型组织的数据示例

在网状模型中,由于是通过指针实现记录之间的联系,因此查询效率较高,而且能表示多对多联系,直接描述复杂的关系。但其应用程序编写比较复杂,程序员必须熟悉数据库的逻辑结构;数据的独立性比较差,程序和数据没有完全独立;由于数据间的联系使用指针表示,指针数据项的存在使数据量大大增加,如果数据关系复杂,则指针部分会占用大量的数据库存储空间,修改数据库中的数据,指针也必须随之变化,因此网络型数据库中的指针的建立和维护成为相当大的额外负担。

1.3.6 关系模型

关系模型以关系数学理论为基础,它是用二维表结构表示实体及实体之间联系的模型。在关系模型中,把数据看成是二维表中的元素,操作的对象和结果都是二维表,一张二维表就是一个关系。经常使用一些概念或名词来描述关系模型的数据结构,关系、元组、属性、域、码(候选码或候选键)、主码(或主键)、主属性、关系模式等。

(1) 关系(或表):一个关系就是一张表,如教师信息表和课程表等。

(2) 元组:表中的一行称为一个元组(不包括表头),一个元组对应现实世界的一个实体。

(3) 属性:表中的一列称为一个属性,它对应实体的属性,一个表会有多个属性,每个属性有一个属性名,同一个表中不能有相同的属性名。

(4) 域:属性的取值范围。

(5) 分量:元组中的一个属性值。

(6) 码：如果表中的某个属性或属性组的值可以唯一地确定一个元组，这样的属性或属性组称为关系的码（候选码或候选键）。

(7) 主码：如果表中存在多个码，那么只能选择其中的一个码来区分元组，被选定的码称为主码或主键，其他候选码或候选键则称为备选键。

(8) 主属性：被定义为主码的属性称为主属性，其他属性则称为非主属性。

(9) 关系模式：对关系的描述，一般表示为“关系名（属性 1，属性 2，…，属性 n ）”。关系模型中没有层次模型中的链接指针，记录之间的联系是通过不同关系中的同名属性来实现的。

例如，在学生成绩管理系统中，有一个学生信息表，学生信息表的结构和部分数据见表 1-1。

表 1-1 学生信息表

学 号	姓 名	性 别	籍 贯	专 业
1101001	张三	女	南京	计算机
1101002	李四	男	徐州	网络
1101003	王五	男	无锡	通信工程
...

在这个表中，有 5 个不同的属性，分别是学号、姓名、性别、籍贯和专业，“1101001、张三、女、南京、计算机”描述的是一个实体（一个学生）的信息，称为一个元组。在这个关系的 5 个属性中，学号属性具有唯一识别每个学生的特性，是该关系的码。学生信息关系可以描述为“学生（学号，姓名，性别，籍贯，专业）”。

关系模型的特点为：建立在关系数据理论之上，有可靠的数据基础；可以描述一对一、一对多和多对多联系；表示的一致性，即实体本身和实体间联系都使用关系描述；关系的每个分量的不可分性，也就是不允许表中表。

关系模型概念清晰，结构简单，格式唯一，理论基础严格，实体、实体间联系和查询结果都采用关系表示，用户比较容易理解。另外，关系模型的存取路径对用户是透明的，程序员无须关心具体的存取过程，减轻了程序员的工作负担，具有较好的数据独立性和安全保密性。但在某些实际应用中，关系模型的查询效率有时不如层次模型和网状模型。因此，为了提高查询效率，有时需要对查询进行一些特别的优化。

1.4 关系数据库理论

关系数据库是建立在关系模型基础上的数据库，它是一种利用数据库进行数据组织的方式，是最有效率的数据组织方式之一。

1.4.1 关系操作

关系操作是一种集合操作方式,即操作的对象和结果都是集合。该方式也称为一次一集合的方式。关系操作主要是对关系数据的查询操作和更新操作。查询操作包括选择、投影、连接、除、并、交和差,更新操作包括对记录的增加、删除和修改。其中,以查询操作为核心。

1.4.2 关系运算

关系操作能力可以用关系代数来表示。关系代数直接用对关系的运算来表达操作目的。本书只介绍专门的关系运算。

1. 选择

选择是从指定关系中选取满足给定条件的若干元组组成一个新的关系。选择运算可表示为 $\sigma_F(R)$ 。其中, R 是关系名, F 是一个逻辑表达式,表示选择条件。

例如,从表 1-1 中查询所有女生的数据,表示为 $\sigma_{\text{性别}='女'}(\text{学生})$ 。其运算结果为关系 $R1$,见表 1-2。

表 1-2 关系 $R1$

学 号	姓 名	性 别	籍 贯	专 业
1101001	张三	女	南京	计算机

选择运算是针对元组的运算。这种运算从水平方向抽取数据,从行的角度得到新的关系,新关系的关系模式不变,其元组是原关系元组的一个子集。

2. 投影

投影是从指定的关系中选取指定的若干属性组成一个新关系。投影运算可表示为 $\pi_A(R)$ 。其中, R 为关系名, A 为 R 中被投影的属性列。

例如,从表 1-1 中查询学生的学号、姓名和专业信息,表示为 $\pi_{\text{学号,姓名,专业}}(\text{学生})$ 。其运算结果为关系 $R2$,见表 1-3。

表 1-3 关系 $R2$

学 号	姓 名	专 业
1101001	张三	计算机
1101002	李四	网络
1101003	王五	通信工程

投影运算是针对属性的运算。这种运算是从垂直方向抽取数据,对关系中的属性进行选择或重组得到新关系。新关系的关系模式包含的属性个数一般比原关系少,或者属性的排列顺序与原关系不同,其内容是原关系的一个子集。

需要注意的是，经过投影运算后，属性减少，元组也可能减少，因为取消了某些属性列后，有可能出现重复元组。按照关系的定义，应取消重复元组。新关系和原关系不是同类关系。

3. 连接

连接是从两个关系中选取属性满足给定条件的元组连接在一起组成一个新关系。连接运算可以表示为 $R \bowtie_{A\theta B} S$ 。其中， R 和 S 是两个关系的名字， A 是 R 中的属性， B 是 S 中的属性， θ 代表比较运算符（ $>$ 、 \geq 、 $<$ 、 \leq 、 $=$ 、 \neq ）， $A\theta B$ 是一个逻辑表达式，表示给定的条件。

当比较运算符 θ 为“ $=$ ”，且进行连接运算的两个关系 R 和 S 中用于比较的两个属性 A 和 B 相同时，称为自然连接，记作 $R \bowtie S$ 。在自然连接得到的新关系中，保持了原来两个关系中的所有属性，并且原来两个关系中用于比较的相同属性只出现一次。

例如，表 1-3 所示的关系 R_2 和表 1-4 所示的关系 R_3 进行自然连接运算，其结果为关系 R_4 ，见表 1-5。

表 1-4 关系 R_3

学 号	成 绩
1101001	80
1101002	77
1101003	67
1101005	76

表 1-5 关系 R_4

学 号	姓 名	专 业	成 绩
1101001	张三	计算机	80
1101002	李四	网络	77
1101003	王五	通信工程	67

1.4.3 关系完整性

关系完整性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。完整性通常包括实体完整性、域完整性、参照完整性和用户定义完整性。

1. 实体完整性

若属性 A 是基本关系 R 的主属性，则属性 A 不能取空值。例如学生关系中，学号是主属性，因此，学号的值不能取空值。

一个关系对应现实世界中一个实体集，关系中的一个元组对应一个实体。实体是可区分的，它们具有某种唯一性标识。如果关系中某一个元组的某个主属性值为空值，则

这个元组不可标识,这就意味着这个元组对应的实体没有其唯一性标识,即存在不可区分的实体,这与客观事实矛盾,这样的实体不是一个完整实体。按照完整性规则要求,主属性不能取空值,如果主关键字是多个属性的组合,则所有主属性均不能取空值。

2. 域完整性

域完整性是指属性被有效性约束,要求关系中的属性值必须具有正确的数据类型、格式及有效的范围,保证输入值的有效性。例如,性别为字符数据类型,只能是“男”或“女”;成绩是数值类型,并且不能为负数。

3. 参照完整性

参照完整性是指定义建立关系之间联系的主关键字与外部关键字引用的约束条件。对于两个关系 R 和 S , R 中存在属性 F , 它是基本关系 R 的外键,与基本关系 S 的主键对应,则对于 R 中每个元组在 F 上的值必须为空值或者等于 S 中某个元组的主键值。

例如,如果在学生表和选修表之间用学号建立关联,学生表是主表,选修表是从表,那么,在向从表中输入一条新记录时,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,这就是参照完整性。

参照完整性还体现在对主表中的删除操作和更新操作。例如,如果删除主表中的一条记录,则从表中外键值与主表的主键值相同的记录也会被同时删除,称为级联删除;如果修改主表中主关键字的值,则从表中相应记录的外键值也随之被修改,称为级联更新。

4. 用户定义完整性

用户定义完整性是根据应用环境的要求和实际需要,针对具体的应用提出的约束条件。这些约束不是关系模型自身要求的,而是具体应用要求的。这样的完整性约束需要用户自己定义,故称为用户定义完整性。用户定义完整性可以通过定义列之间的有效性约束来实现。

1.5 数据库系统结构

数据库系统结构可以有多种不同的层次或不同的角度。从数据库管理系统角度,数据库系统通常采用三级模式结构,这是数据库系统内部的体系结构,通常称为数据库系统的模式结构;从数据库最终用户角度,数据库系统的结构可以分为单机结构、主从式结构、分布式结构、客户机/服务器结构和浏览器/服务器结构等,这是数据库系统外部的体系结构,简称为数据库系统的体系结构。

1.5.1 数据库系统的模式结构

尽管实际的数据库管理系统使用的环境、内部数据的存储结构、使用的语言不同,但它们都采用了三级模式结构,并提供两级映像功能。



1. 三级模式结构

数据库系统的三级模式结构包括外模式、模式和内模式，它们是对数据的三个抽象层次，其结构如图 1-7 所示。三级模式结构把对数据的具体组织留给数据库管理系统管理，使用户能逻辑、抽象地处理数据，而不必关心数据在计算机中的具体表示与存储。

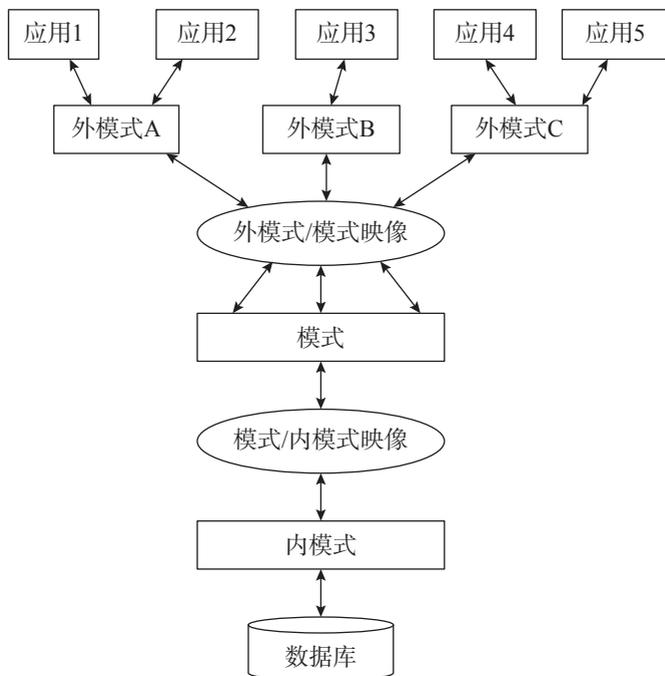


图 1-7 数据库系统的三级模式结构

外模式（也称子模式或用户模式）是三级模式结构的最外层，也是面向具体用户或应用程序的数据视图，即特定用户或应用程序涉及的数据的逻辑结构。外模式是模式的子集，不同用户使用不同的外模式。一个数据库可以有多个外模式，每一个外模式都是为不同的用户建立的数据视图。由于各用户的需求和权限不同，各个外模式的描述也是不同的。即使是对模式中的同一数据，其在不同外模式中的结构、密级等都可以不同。每个用户只能调用对应的外模式涉及的数据，其余数据是无法访问的。数据库管理系统提供外模式描述语言来定义外模式。

模式（也称逻辑模式）是数据库中全部数据的逻辑结构和特征的描述，也是所有用户的公共数据视图。它通常以某种数据模型为基础，定义数据库全部数据的逻辑结构，如数据记录的名称、数据项的名称、类型、值域等。还要定义数据项之间的联系、不同记录之间的联系，以及与数据有关的安全性、完整性等要求。一个数据库系统只能有一个逻辑模式，它不涉及硬件环境和物理存储细节，也不与任何计算机语言有关。数据库管理系统提供模式描述语言来定义模式。

内模式（也称存储模式或物理模式）既定义了数据库中全部数据的物理结构，也定义了数据的存储方法、存取策略等。内模式的设计目标是将系统的逻辑模式组织成最优

的物理模式，以提高数据的存取效率，改善系统的性能指标。数据库管理系统提供内模式描述语言来描述和定义内模式。

2. 两级映像功能

为了能够在内部实现这三个抽象层次的联系和转换，数据库系统在这三级模式结构之间提供了两层映像：外模式 / 模式映像和模式 / 内模式映像。

外模式 / 模式映像实现了从外模式到模式之间的相互转换。对于每一个外模式，数据库系统都有一个外模式 / 模式映像，它定义了该外模式与模式之间的对应关系。这些映像定义通常包含在各自外模式的描述中。当模式改变时，只要相应改变外模式 / 模式映像，就可以使外模式保持不变。应用程序是依据数据的外模式编写的，外模式不变，应用程序就没必要修改。这种用户数据独立于全局的逻辑数据的特性叫作数据的逻辑独立性，外模式 / 模式映像功能保证了数据的逻辑独立性。

模式 / 内模式映像实现了从模式到内模式之间的相互转换。模式 / 内模式映像是唯一的，它定义了数据库全局逻辑结构与存储结构之间的对应关系。当数据库的存储结构改变时，只要相应改变模式 / 内模式映像，就可使模式保持不变。这种全局的逻辑数据独立于物理数据的特性叫作数据的物理独立性。模式不变，建立在模式基础上的外模式就不会变，与外模式相关的应用程序也就不需要改变，模式 / 内模式映像功能保证了数据的物理独立性。

数据库的三级模式结构是数据库组织数据的结构框架，依照结构框架组织的数据才是数据库的内容。在设计数据库时，主要是定义数据库的各级模式；而用户使用数据时，关心的只是数据库的内容。数据库的模式通常是稳定的，而数据库中的数据经常是变化的。

3. 三级模式结构的优点

三级模式结构有以下 4 个优点。

1) 保证数据的独立性

将模式和内模式分开，保证了数据的物理独立性；将外模式和模式分开，保证了数据的逻辑独立性。

2) 简化了用户接口

按照外模式编写应用程序或输入命令，不需要了解数据库的逻辑结构，更不需要了解数据库内部的存储结构，方便用户的使用。

3) 有利于数据共享

不同的外模式为不同的用户提供不同的数据视图，从而实现不同的用户对数据库中全部数据的共享，减少了数据冗余。

4) 有利于数据的安全保密

在外模式下根据要求进行操作，只能对限定的数据进行限定的操作，保证了其他数据的安全性与保密性。



1.5.2 数据库系统的体系结构

一个数据库应用系统通常包括数据存储层、应用层与用户界面层三个层次。数据存储层一般由数据库管理系统承担对数据库的各种维护操作；应用层是使用某种程序设计语言实现用户要求的各项工作的程序；用户界面层提供用户的可视化图形操作界面，便于用户与数据库系统之间的交互。

从最终用户角度看，数据库系统可分为以下 5 种。

1. 单机结构

单机结构是一种比较简单的数据库系统。在单机系统中，整个数据库系统包括的应用程序、数据库管理系统和数据库都安装在一台计算机上，由一个用户独占，不同机器之间不能共享数据。这种数据库系统也称桌面系统。在这种桌面型数据库管理系统中，数据的存储层、应用层和用户界面层的所有功能都存储在单机上，容易造成大量的数据冗余。

2. 主从式结构

主从式系统是指一台大型主机带着若干终端的多用户结构。在这种结构中，全部数据都集中存放在主机中，数据库管理系统和应用程序也存放在主机中，所有处理任务都由主机完成。各终端用户可以并发地访问主机上的数据库，共享其中的数据。

在主从式结构的数据库管理系统中，数据的存储层和应用层都放在主机中，用户界面层放在各个终端上。当终端用户数目增加到一定程度后，主机的任务将十分繁重，常处于超负荷状态，这样会使系统性能大大降低。

主从式结构的优点是简单、可靠、安全；缺点是主机的任务很重，终端数目有限，当主机出现故障时，会影响整个系统的使用。

3. 分布式结构

分布式结构是指地理上或物理上分散而逻辑上集中的数据库系统。每台计算机上都装有分布式数据库管理系统和应用程序，可以处理本地数据库中的数据，也可以处理异地数据库中的数据。在分布式数据库系统中，大多数处理任务由本地计算机访问本地数据库完成局部应用。对于少量本地计算机不能胜任的处理任务，通过网络同时存取和处理多个异地数据库中的数据执行全局应用。分布式数据库系统适应了地理上分散的组织对于数据库应用的需求。

分布式结构的优点是体系结构灵活，能适应分布式管理和控制，经济性能好，可靠性高，在一定的条件下，响应速度快，可扩充性好；其缺点是系统开销较大，存取结构复杂，数据的安全性和保密性难以解决等。

4. 客户机 / 服务器结构 (client/server 结构, C/S 结构)

随着工作站功能的增强和广泛使用，人们开始把数据库管理系统的功能和应用分开。网络中专门用于执行数据库管理系统的功能的计算机称为数据库服务器，简称为服务器 (server)，其他安装数据库应用程序的计算机称为客户机 (client)，这种结构称为客户机 /

服务器结构。

在 C/S 结构的数据库系统中,数据存储层位于服务器上,而应用层和用户界面层位于客户机上。服务器的任务是完成数据管理、信息共享、安全管理等,它接受并处理来自客户机的数据访问请求,然后将结果返回给用户;客户机的任务是提供用户界面,提交数据访问请求,接收和处理数据库的返回结果。由于服务器对数据服务请求进行处理后只返回结果,而不是返回整个系统,因此减少了网络上的数据传输量,提高了系统的性能和负载能力。

C/S 结构的优点有两个:可以减少网络流量,提高系统的性能、吞吐量和负载能力;使数据库更加开放,客户机和服务器可以在多种不同的硬件和软件平台上运行。C/S 结构的缺点是系统的客户机程序的更新升级有一定困难。

5. 浏览器 / 服务器结构

浏览器 / 服务器结构 (browser/server 结构, B/S 结构) 是随着互联网技术的兴起,对客户机 / 服务器结构的变化或改进的结构。

浏览器 / 服务器结构由浏览器 (browser)、Web 服务器、数据库服务器三层结构组成。在这三层结构中,Web 服务器担任中间层应用服务器的角色,它是连接数据库服务器的通道。在浏览器 / 服务器结构中,用户通过浏览器向 Web 服务器发出请求,服务器对浏览器的请求进行处理,将用户所需的信息返回浏览器。

B/S 结构的优点是具有分布性特点,可以随时随地进行查询、浏览等业务处理;业务扩展简单方便,通过增加网页便可增加服务器功能;维护简单方便,只需要改变网页即可实现所有用户的同步更新;开发简单,共享性强。

项目拓展训练

1. 拓展训练内容

设计一个学生管理数据库,涉及学生、系部、课程、班级等的相关信息。

2. 拓展训练目的

掌握概念模型和逻辑模型的设计。

项目小结

本项目对数据、信息和数据处理的定义,数据库、数据库系统和数据库管理系统的概念、特点,数据模型的定义、组成要素和类型,概念模型、层次模型、网状模型和关系模型的定义、特点,关系数据库的理论和数据库系统结构等,均做了详细的讨论。

数据是数据库中存储的基本对象,是可以被计算机接受并能够被计算机处理的符号。信息是对数据的解释,是经过加工处理后具有一定含义的数据集合。将数据转换成信息的过程称为数据处理。

数据库是以一定的方式将相关数据组织在一起并存储在外存上，能为多个用户共享、与应用程序彼此独立的一组相互关联的数据集合。数据库系统是由数据库及其管理软件组成的系统，它能够有组织、动态地存储大量数据，提供数据处理和数据共享机制，是存储介质、处理对象和管理系统的集合体。数据库管理系统是处理数据访问的软件系统，它位于用户与操作系统之间。数据库管理系统的功能体现在数据定义、数据操纵、数据库运行管理、数据维护、数据库的传输和数据通信接口 6 个方面。

计算机数据管理技术大致经历了人工管理、文件系统和数据库系统三个阶段。

数据模型按不同的应用层次分成概念模型、逻辑模型和物理模型三种类型。

关系数据库是建立在关系模型基础上的数据库，是利用数据库进行数据组织的一种方式。完整性通常包括实体完整性、域完整性、参照完整性和用户定义完整性。

数据库系统结构可以有多种不同的层次或不同的角度。从数据库管理系统角度，数据库系统通常采用三级模式结构；从数据库最终用户角度，数据库系统的结构可以分为单机结构、主从式结构、分布式结构、客户机 / 服务器结构和浏览器 / 服务器结构，简称为数据库系统体系结构。