

第5

章 数据管理

研究和制造计算机的最初目的是实现快速的科学计算,但是今天,当计算机已经应用于仓储管理、购物结算、档案资料的查询等领域时,它所面对的则是庞大的数据信息和复杂多样的数据类型。为了有效地管理和利用这些数据,就催生了计算机的数据管理技术。

如何收集有用的信息和数据,如何合理地组织信息,如何筛选所需的数据,以及如何高效地利用数据成为我们今天必须面对的问题。传统的纸面数据在存储效率和传播效率上有很大局限性,由此催生了电子数据,“无纸化办公”已经由概念逐渐应用到多个行业领域,增强了人们的环保意识,也促进了各行业办公模式的不断升级。现代化和信息化建设的步伐不断加快,使得利用计算机技术来安全高效地管理信息和数据成为必然。

数据管理是对数据进行收集、分类、组织、存储、检索和维护的一系统列活动。数据管理的目的是为充分有效地发挥数据的作用,实现数据有效管理的关键是数据的组织。数据库技术和数据库系统是伴随数据管理技术的发展而产生的。数据库技术因其扎实的理论基础和广泛的应用前景,已经成为计算机数据处理与信息管理以及信息化基础设施的核心和基础。

5.1 数据库技术

数据库技术能有效解决计算机信息处理中对批量数据的组织和存储问题,实现数据共享和保障数据安全,实现高效地检索数据和处理数据,并提供科学的技术保障和手段支持。伴随计算机技术和互联网技术的发展,数据库技术也经历了一个由低级到高级的发展过程。

5.1.1 数据库技术概述

1. 数据库技术的发展历程

数据库技术诞生至今已有半个多世纪,随着计算机技术的发展,遵循着数据存储冗余不断减少,数据独立性不断增强,数据操作方式日益方便的发展主线,数据库技术经历了人工管理、文件管理和数据库管理 3 个阶段。

(1) 人工管理阶段

20 世纪 50 年代中期以前的数据处理是人工管理阶段,没有管理数据的软件。计算机主要用于科学计算,外部存储器只有纸带、卡片和磁带等。这一时期数据管理的主要特点如下。

① 程序直接管理数据。人工管理阶段,应用程序处理数据时,被处理的数据直接写入程序代码中,由程序管理。

② 数据不共享,冗余度大。由于数据是面向特定的应用程序,一组数据只能对应一个应用程序,如图 5.1 所示。即使几个程序要处理的是同一批数据,也无法实现数据共享,只能重复性地把数据写入每个程序中,因此数据冗余度大。

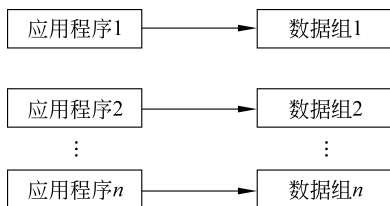


图 5.1 人工管理阶段的应用程序与数据之间的关系

③ 数据缺乏独立性。当数据的逻辑结构和物理结构改变了,应用程序也要改变。

(2) 文件管理阶段

20 世纪 50 年代后期至 60 年代中期,由于磁盘和磁鼓等大容量存储设备的出现,大量数据可以被组织为文件的形式存储在外存储设备中,并由操作系统提供的软件(一般为文件系统)统一管理。文件管理阶段的主要特点如下。

① 应用程序与数据分开存储。应用程序中不带要处理的数据,被处理的数据均独立地组织成数据文件。应用程序与数据文件被分别存入外存储设备,因此,数据和应用程序之间具有了一定的独立性。

② 数据共享性弱,冗余度大。文件管理阶段,数据不再专属于某一特定程序,同一批数据可为多个应用程序使用,实现了数据以文件为单位的共享,如图 5.2 所示。由于文件系统只能简单存取数据,每个数据文件中存在部分的重复数据,因此数据冗余度仍然很大。

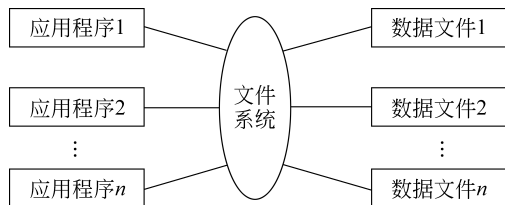


图 5.2 文件管理阶段的应用程序与数据之间的关系

③ 数据不一致。因为数据文件彼此独立和数据的冗余度大,不可避免地会出现数据的不一致,这种不一致性给数据的更新与维护带来困难。

文件管理数据的方式适合处理数据量较小的情况,不适应大规模数据的管理。

(3) 数据库管理阶段

20 世纪 60 年代以后,随着数据处理的规模越来越大,应用范围越来越广,大容量的磁盘出现,数据库技术应运而生,出现了专门对数据库进行统一管理的软件,这就是数据库管理系统。从文件系统阶段到数据库管理阶段,数据管理技术发生了质的飞跃。数据库管理阶段具有以下特点。

① 数据结构化。数据库中的数据是按照一定的数据模型来组织的,可以最大限度减少数据冗余,提高数据的一致性和完整性,降低应用程序的开发和维护代价。

② 应用程序与数据完全独立。应用程序的修改不影响数据,数据的结构及存储位置的改变也不影响应用程序,如图 5.3 所示。

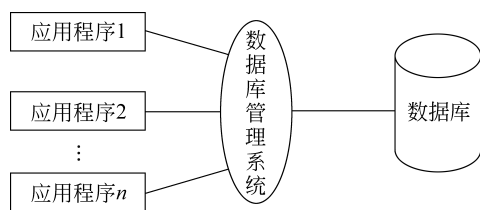


图 5.3 数据库管理阶段应用程序与数据之间的关系

③ 数据共享性好。数据不是面向应用程序,而是面向整个系统,任何应用程序都可以使用数据库中其所需要的数据,因此,数据共享性好。

2. 数据库的基本概念

(1) 数据库

数据库(Database, DB)就是用来存放数据的“仓库”,是将数据有组织地存储于计算机的外存设备,是可共享、可统一管理的数据集合。数据库中的数据具有较小的冗余度、较高的数据独立性和可扩展性,可以为不同的应用程序所共享。

(2) 数据库管理系统

数据库管理系统(Database Management System, DBMS)是一种用来操纵和管理数据库的大型软件,用于建立、使用和维护数据库。它是数据库系统的核心组成部分,是用户与操作系统之间,或者应用程序与操作系统之间的系统软件。DBMS 能对数据库实施有效地组织和存储,对数据库中存储的数据进行操作,如查询、更新、插入、删除以及各种控制等。

DBMS 的基本功能如下。

① 数据定义: DBMS 提供数据定义语言(Data Definition Language, DDL),供用户定义数据库、定义和修改数据库中的表结构以及完整性约束规则等。

② 数据操纵: DBMS 提供数据操纵语言(Data Manipulation Language, DML),供用户对数据的插入、删除、更新、查询等操作。

③ 数据库的建立和维护: 数据库的建立包括数据库初始数据的输入和转换功能,数据库的维护包括数据库的转储、恢复、重新组织以及性能监控等功能,这些功能分别由一

些实用程序完成。

④ 数据库的运行管理：数据库的所有操作都要在控制程序的统一管理下进行，以保证数据的安全性、完整性，实现多个用户对数据库使用的并发性控制，以及安全性检查和存取限制控制。运行管理还包括运行日志的组织管理、事务的管理和自动恢复等功能。这些功能保证了数据库系统的正常运行。

⑤ 通信：DBMS 具有与操作系统的联机处理，具有分时系统及远程作业输入的相关接口，负责处理数据的传送。对网络环境下的数据库系统，还包括 DBMS 与网络中其他软件系统的通信功能以及数据库之间的互操作功能。

在数据库技术的发展过程中，出现了一些具有代表性的数据库管理系统软件，市场上比较流行的数据库管理系统产品主要是 Oracle、DB2、MySQL、SQL Server、Sybase 和 Access 等。

(3) 数据库系统

数据库系统(Database System, DBS)是由硬件系统、数据库管理系统、数据库、应用程序、管理数据库相关人员等构成的综合体。数据库系统可以简单地理解为是在计算机系统中引入数据库后的系统。应用程序是指利用各种开发工具开发的，能满足特定应用环境的程序。数据库管理员(Database Administrator, DBA)，是负责数据库的建立、使用和维护的专门人员。过去的数据库公司提供的产品仅仅是数据库管理系统，随着数据库公司向面向应用的系统集成转型，现在的数据库系统往往是一整套网络数据库应用解决方案，其中包括数据库管理系统、数据库应用服务器、开发工具套件等。常见的数据库系统组成如图 5.4 所示。

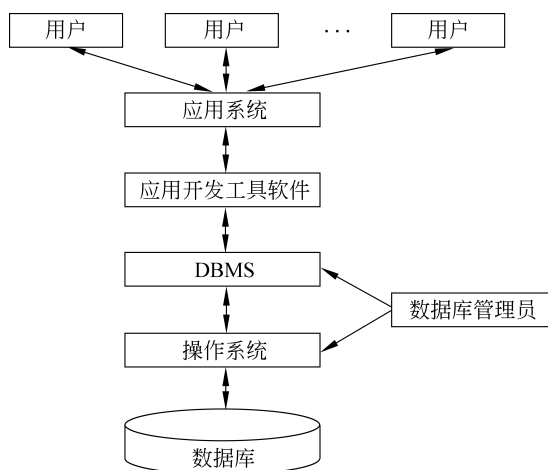


图 5.4 常见的数据库系统组成

5.1.2 数据库体系结构

1975 年，美国国家标准学会/标准计划 and 需求委员会(ANSI/SPARC)为数据库系统

建立了三层体系结构(即三级模式结构):面向用户或应用程序员的用户级、面向数据库设计和维护人员的概念级、面向系统程序员的物理级,分别称为外模式、概念模式和内模式。为了能够在内部实现数据库的三个抽象层次的联系和转换,DBMS在这三级模式之间提供了两层映射:外模式-模式映射、模式-内模式映射。

1. 外模式

外模式又称子模式或用户模式,是数据库用户看见的局部数据的逻辑结构和特征的描述。外模式是应用程序与数据库系统之间的接口,是保证数据库安全性的一个有效措施。用户可使用数据定义语言(DDL)和数据操纵语言(DML)来定义数据库的结构和对数据库进行操纵。对于用户而言,只需要按照所定义的外模式进行操作,而无需了解概念模式和内模式等的细节。一个数据库可以有多个外模式。

2. 概念模式

概念模式又称模式或逻辑模式,是数据库整体逻辑结构的完整描述。概念模式位于数据库系统模式结构的中间层,不涉及数据的物理存储细节和硬件环境,与应用程序、开发工具及程序设计语言无关。一个数据库只能有一个概念模式。

3. 内模式

内模式又称存储模式,是数据库内部数据存储结构的描述。它定义了数据库内部记录类型、索引和文件的组织方式以及数据控制方面的细节。一个数据库只能有一个内模式。

4. 外模式-模式映射

对应于同一个模式可以有多个外模式。对于每一个外模式,数据库系统都有一个外模式-模式映射,它用于保持外模式与概念模式之间的对应性。当数据库的概念模式改变时,只需要对外模式-模式映射进行修改,而使外模式保持不变。从而使应用程序不必修改,保证了数据与程序的逻辑独立性。

5. 模式-内模式映射

在概念模式和内模式之间存在着模式-内模式映射。数据库只有一个模式,也只有一个内模式,所以模式-内模式映射是唯一的,它用于保持概念模式与内模式之间的对应性。当数据库的存储结构改变时,由数据库管理员对模式-内模式映射做相应修改,可以使概念模式保持不变,从而应用程序不必修改,保证了数据与程序的物理独立性。数据库三级模式结构如图 5.5 所示。

5.1.3 数据模型

模型是对现实世界特征的模拟和抽象,就像我们在建造一个公园时,总是先绘出建筑

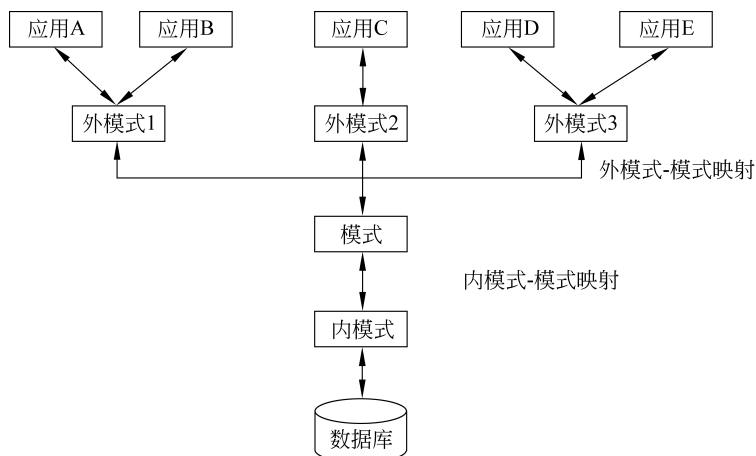


图 5.5 数据库三级模式结构

设计图和制作相应的沙盘一样,需要将想象中的观点和概念用形式化的模型来具体表现。由于计算机不能直接处理现实世界中的具体事物,因此人们必须先把具体事物抽象化为模型,并转换为计算机可以处理的数据形式,从而达到以模拟的方式来表现现实世界和处理实际事物。

数据模型(Data Model,DM)是数据特征的抽象。数据模型所描述的内容有三部分:数据结构、数据操作和数据约束。

数据结构主要描述数据的类型、内容、性质以及数据间的联系等,是目标类型的集合;数据操作主要描述在相应的数据结构上的操作类型和操作方式;数据约束主要描述数据结构内数据间的语法、词义联系、它们之间的制约和依存关系,以及数据动态变化的规则,以保证数据的正确性、有效性和相容性。

数据模型为数据库系统的信息表示与操作提供了一个抽象的框架。数据模型通常满足 3 个基本要求:能够比较真实地反映现实世界的实际情况;能够比较容易地被人们所理解;能够准确和方便地在计算机中实现。

数据模型以现实世界为基础,要经历两个演变过程:一个是建立概念模型(或信息模型),它是以人的观点来模拟现实世界的模型;另一个是在概念模型的基础上生成基本的数据模型,它是按计算机系统的特征,以计算机能够识别和处理的方式建立数据的结构,直接面向数据库中数据的逻辑结构。

1. 概念模型

概念模型是按照人们对客观事物的认识与理解对数据和信息建模,它的建立不依赖于任何数据库管理系统,但又具有较强的语义表达能力,能够方便、直接地表达应用中的各种语义知识,容易向数据库管理系统支持的逻辑数据模型转换,也利于与用户进行交流。概念模型需要相关术语做语义表达,这些术语是概念模型的基本语义单元。相关术语如下。

(1) 实体(Entity)

实体是现实世界中客观存在的、能相互区别的事物。例如,一个学生、一个班级、一门课程、一件商品等,都可以看作是抽象的实体。概念性的事物,例如商品的价格,也可以看作是抽象的实体。

(2) 属性(Attribute)

实体或实体之间的联系所具有的特性称为属性,一个实体可由若干个属性来描述。属性有“型”“值”两个方面的含义。“型”是属性的名字,型的具体赋值就是属性的“值”。例如,表示一个学生实体时,可以选择“学号”“姓名”“性别”“专业”等属性去描述,而“2019121009”“李丽丽”“女”“电子 19-1 班”,这些属性值的集合具体指定了一个学生。实体的属性也被称为字段。

(3) 实体型(Entity Type)

实体型是对具有相同特征和相同性质的实体的总称,它采用实体名及描述该实体的所有属性名的集合来表示。例如,采用“学号”“姓名”“性别”“专业”四个属性来描述一个名为“学生”的实体,则这个实体型可以表示为:学生(学号,姓名,性别、专业)。

(4) 实体集(Entity Set)

某个实体型的全部实体称为实体集。例如,某个班的所有学生组成一个学生实体集。

(5) 键(Key)

键也称关键字或码,是指能够在实体集中唯一标识一个实体的一个属性或一个属性组(即多个属性)。例如,学号可以作为学生实体的关键字,因为每个学号唯一标识一个学生。这种能够在实体集中唯一地标识一个实体的关键字称为主关键字(或主键),它的值不允许重复,也不允许为空。

(6) 域(Domain)

域是属性的取值范围或值域。域的取值可以是整数、实数、字符串、日期、逻辑的真或假等。

(7) 联系(Relationship)

实体内部或实体之间的相互关系称为联系。组成一个实体的各属性之间是相互联系的,也称为实体的内部联系,借助于这种联系可以组建数据表。实体之间的联系通常是指不同实体集之间的联系,被称为实体的外部联系。

2. 实体-联系方法

实体-联系方法(Entity-Relationship Approach)是概念模型表示法的一种,简称为E-R图,利用E-R图可以描述客观事物的实体型、实体所具有的属性以及实体间的联系。其中,实体型用矩形表示;实体的每个属性用椭圆形表示,并用线段与对应的实体型连接;联系用菱形表示,用线段连接相关实体来表示它们之间有联系,并在线段旁注明其联系的类型。实体间的联系通常分为3种类型:一对一联系、一对多联系和多对多联系。

(1) 一对一联系

当前实体集中的每一个实体在另一个实体集中最多只能找到一个可以与它相对应的实体;反之,在另一个实体集中的每一个实体也最多只能当前实体集中找到一个相对应

的实体,那么这两个实体集之间就存在着一对一的联系,记作 $1:1$ 。例如,一个学校只有一个校长,而这个校长只能在这个学校内任职,那么校长与学校之间就是一对一的联系,可以用图 5.6(a)表示。

(2) 一对多联系

当前实体集中的每一个实体在另一个实体集中可以找到多个能够与它相对应的实体;反之,在另一个实体集中的每一个实体,却只能在当前实体集中找到一个与之相对应的实体,那么这两个实体集之间就存在着一对多的联系,记作 $1:n$ 。例如,一个班级里有多个学生,但一个学生只能属于一个班级,那么班级和学生之间就是一对多的联系,可以用图 5.6(b)表示。

(3) 多对多联系

当前实体集中的每一个实体在另一个实体集中可以找到多个能够与它相对应的实体;反之,在另一个实体集中的每一个实体也能在当前实体集中找到多个与之相对应的实体,那么这两个实体集之间就存在着多对多的联系,并记作 $m:n$ 。例如,一门课程允许有多个学生选修;反之,一个学生又可以选修多门课程,那么课程与学生之间就是多对多的联系,可以用图 5.6(c)表示。

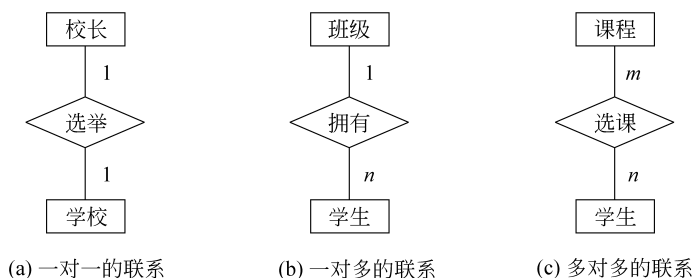


图 5.6 实体间联系的图形表示

图 5.7 所示的是学生实体与课程实体间联系的 E-R 图,学生实体与课程实体之间是多对多的联系。

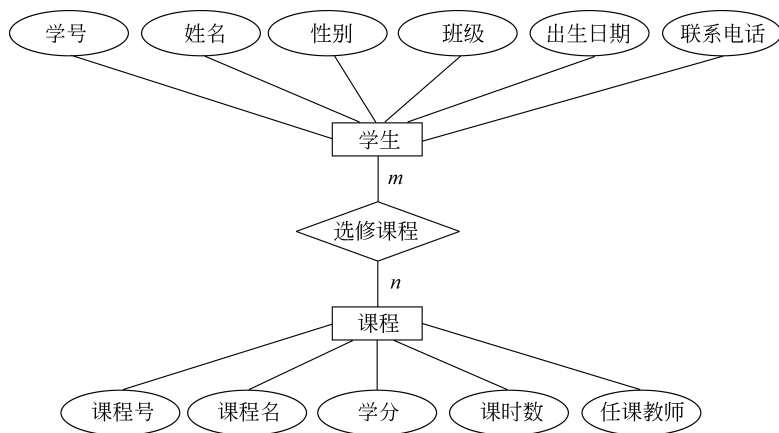


图 5.7 学生实体与课程实体间联系的 E-R 图

3. 数据模型的类型

(1) 层次模型(Hierarchical Model)

```

graph TD
    A[信息工程学院] --> B[计算机科学与技术]
    A --> C[网络工程]
    A --> D[自动化]
    A --> E[数字媒体]
    D --> F[自动控制]
    D --> G[电子工程]
  
```

(2) 网状模型(Network Model)

```

graph TD
    A[学生] --> B[课程]
    A --> C[教室]
    D[教师] --> B
    D --> C
    B --> C

```

(3) 关系模型(Relational Model)

第5章 数据管理

5.1.4 关系模型

关系模型从形式上看就是一张二维表(简称为表)。以下介绍关系模型中的基本概念和基本术语。

1. 关系模型的基本术语

(1) 关系(Relation)

一个关系对应一个二维数据表,换句话说,一个关系的逻辑结构是由行列组成的一张二维表,其中每列称为一个字段,每行称为一条记录,如表 5.1 所示。

表 5.1 学生基本信息

学号	姓名	性别	专业	出生日期	联系电话
190204223	王丽丽	女	电子 19	1999/3/30	136xxxx2502
190834204	张国	男	生物 19	2000/5/8	
190206120	赵欣	女	电子 19	1999/9/23	
180521116	陈玉华	男	会计 18	1999/2/25	139xxxx3505
190322115	李文江	男	数学 19	2000/6/19	138xxxx2406
190122123	宋小兵	男	金融 19	1999/11/28	136xxxx4586

(2) 属性(Attribute)

在二维表中,属性就是字段,字段有名和值的区别,一列中处于表头位置的单元格内容是字段名(或称属性名),除字段名外,该列的其余数据为字段值(或称属性值),每个字段值又被称为分量或数据项。

(3) 域(Domain)

域是字段值(或属性值)的取值范围。

(4) 元组(Tuple)

除表头所在行以外,二维表中每一行称为一个元组或一条记录,它由一组不同的属性值组成。二维表的一条记录相当于一个实体,整个二维表的全部记录相当于一个实体集。

(5) 候选码(Candidate Key)

候选码就是关键字,它是能够唯一标识一个二维表中每个元组的属性或属性组。候选码简称为码。例如,在表 5.1 中,可以将属性“学号”作为该二维表的候选码,因为学号可以唯一确定一个学生,即一个学生的完整记录。又如,表 5.2 展示了一个名为“学生作业成绩”的二维表,该表不能单独使用“学号”属性作为候选码,因为一个学生可以选修多门课程,该表中可能出现两条或多条学号相同但选课不同的记录。此时,学号和选课代码的组合可以唯一确定一个学生的一门课程的成绩,所以应该将“学号”及“选课代码”两个属性组成一个属性组,作为表的候选码。