

数据库基础

1.1 数据库系统概述

数据库技术自从 20 世纪 60 年代中期产生以来,无论是在理论还是在应用方面都已变得相当的重要和成熟,成为计算机科学的重要分支。数据库技术是计算机领域发展最快的学科之一,也是应用很广、实用性很强的一门技术。目前,数据库技术已从第一代的网状、层次数据库系统和第二代的关系数据库系统,发展到以面向对象模型为主要特征的第三代数据库系统。

随着计算机技术的飞速发展及其应用领域的扩大,特别是计算机技术和因特网的发展,使得基于计算机网络和数据库技术的管理信息系统、各类应用系统得到了突飞猛进的发展。如事务处理系统(Transaction Processing Systems, TPS)、地理信息系统(Geographic Information Systems, GIS)、联机分析系统(On-Line Analytical Processing, OLAP)、决策支持系统(Decision Support Systems, DSS)、企业资源计划(Enterprise Resource Planning, ERP)、客户关系管理(Customer Relationship Management, CRM)、数据仓库(Data Warehouse, DW)及数据挖掘(Data Mining, DM)等系统都是以数据库技术作为其重要的支撑,因此,数据库技术的基础知识和基本技能已经成为信息社会人们的必备知识。

1.1.1 数据、数据库、数据库管理系统、数据库系统

数据、数据库、数据库管理系统、数据库系统是与数据库技术密切相关的 4 个基本概念。

1. 数据(Data)

(1) 数据的定义

数据是用来记录信息的可识别的符号,是信息的具体表现形式。

(2) 数据的表现形式

数据是数据库中存储的基本对象。数字是最简单的一种数据,是数据的一种传统和

狭隘的理解。从广义上理解,数据的种类很多,文字、图形、图像、声音、视频、语言、学生的档案记录、货物的运输情况等都可以转化为计算机可以识别的标识,并以数字化后的二进制形式存入计算机。

为了了解世界、交流信息,人们需要描述各种事物。在日常生活中直接用自然语言描述,在计算机中,为了存储和处理这些事物,就要抽出人们对这些事物感兴趣的特征组成一个记录来描述。例如,在学生档案中,如果人们最感兴趣的是学生的姓名、性别、年龄、出生年月,那么可以这样来描述一个学生:

(赵一,女,23,1987年5月)

(3) 数据与信息的联系

数据解释的含义称为数据的语义(即信息),数据与其语义是不可分的:数据是信息的符号表示或载体,信息则是数据的内涵,是对数据的语义解释。数据的形式不能完全表达其内容,例如,“小明今年12岁了”,数据“12”被赋予了特定的语义“岁”,它才具有表达年龄信息的功能。

2. 数据库(DataBase,DB)

数据库从字面意思来说就是存放数据的仓库,具体而言是长期存储在计算机内的、有组织的、可共享的大量数据集合,其可供多用户共享,数据库中的数据按一定的数据模型组织、描述和存储,具有尽可能小的冗余度和较高的数据独立性与易扩张性。

数据库具有以下两个比较突出的特点。

(1) 把在特定的环境中与某应用程序相关的数据及其联系集中在一起并按照一定的结构形式进行存储,即集成性。

(2) 数据库中的数据能被多个应用程序的用户所使用,即共享性。

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

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,是数据库系统的核心组成部分。用户在数据库系统中的一些操作,例如,数据定义、数据查询、数据操纵和数据控制都是由数据库管理系统来实现的。DBMS的用途是科学地组织和存储数据、高效地获取和维护数据。

数据库管理系统主要包括以下几个功能。

(1) 数据定义

DBMS提供数据定义语言(Data Definition Language,DDL),用户通过DDL可以对数据库中的数据对象(包括表、视图、索引、存储过程等)进行定义,如表的创建、删除和修改等。

(2) 数据操纵

DBMS提供数据操纵语言(Data Manipulation Language,DML),用户通过DML可

以实现对数据库的一些基本操作,如表信息的查询、插入、删除和修改等。

(3) 数据库的运行管理

数据库的运行管理是数据库管理系统的核心所在。DBMS 通过对数据库的建立、运用和维护提供统一的管理和控制,以保证数据安全、正确、有效地运行。DBMS 通过数据的安全性控制、完整性控制、多用户应用环境的并发控制和数据库数据的系统备份与恢复 4 个方面来实现对数据库的统一控制功能。

(4) 数据库的建立和维护功能

数据库的建立和维护功能包括数据库初始数据的输入、转换功能,数据库的转储、恢复功能,数据库的重组织和性能监视、分析功能等,这些功能通常由一些实用程序完成。

4. 数据库系统(DataBase System, DBS)

数据库系统是指在计算机系统中引入数据库后的系统,其构成主要有数据库及相关硬件、数据库管理系统及开发工具和相关软件、应用系统、数据库管理员和用户。其中,数据库的建立、使用和维护的过程要有专门的人员来完成,这些人员就被称为数据库管理员(DataBase Administrator, DBA)。

在不引起混淆的情况下常常把数据库系统简称为数据库。

数据库系统结构如图 1-1 所示。

数据库系统在整个计算机系统中的地位如图 1-2 所示。

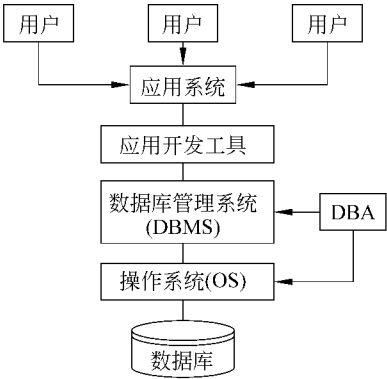


图 1-1 数据库系统结构

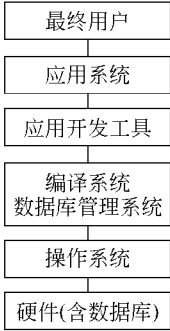


图 1-2 数据库系统在计算机系统中的地位

1.1.2 数据库管理技术的产生和发展

数据处理是对各种数据进行收集、存储、加工和传播的一系列活动的总和,数据管理则是数据处理的中心问题,因此,数据管理是对数据进行分类、组织、编码、存储、检索和维护的管理活动的总称。就用计算机来管理数据而言,数据管理是指数据在计算机内的一系列活动的总和。

人们借助计算机来进行数据管理虽是近 50 多年的事,然而数据管理技术已经历了人工管理、文件系统及数据库系统 3 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。硬件方面:外存只有纸带、卡片、磁带,没有磁盘等直接存取存储设备;软件方面:没有操作系统和管理数据的软件;数据处理方式:批处理。

人工管理阶段的数据具有以下几个特点。

(1) 数据不保存

由于当时计算机主要用于科学计算,数据保存上并不作特别的要求,只是在计算某一个课题时将数据输入,用完就退出,对数据不作保存,有时对系统软件也是这样。

(2) 由应用程序管理数据

数据没有专门的软件进行管理,需要应用程序自己进行管理,应用程序中要规定数据的逻辑结构并设计其物理结构(包括存储结构、存取方法、输入/输出方式等),因此程序员的负担很重。

(3) 数据不共享

数据是面向应用的,一组数据只能对应一个程序。如果多个应用程序涉及某些相同的数据,则必须各自进行定义,从而无法进行数据的参照,因此程序间有大量的冗余数据。

(4) 数据不具有独立性

数据的独立性包括了数据的逻辑独立性和数据的物理独立性。在人工管理阶段,当数据的逻辑结构或物理结构发生变化时,必须对应用程序做相应的修改。

在人工管理阶段,应用程序与数据之间的对应关系如图 1-3 所示。

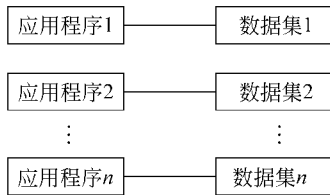


图 1-3 人工管理阶段应用程序与数据之间的对应关系

2. 文件系统阶段

20 世纪 50 年代后期到 60 年代中期,计算机已大量用于数据的管理。硬件方面:有了磁盘、磁鼓等直接存取存储设备;软件方面:操作系统中已经有了专门的管理软件,一般称为文件系统;数据处理方式:批处理、联机实时处理。

文件系统阶段的数据有以下几个特点。

(1) 数据长期保存

由于计算机大量用于数据处理,数据需要长期保留在外存上以反复进行查询、插入、删除及修改等操作。

(2) 由文件系统管理数据

由专门的软件即文件系统进行数据管理,文件系统把数据组织成相互独立的数据文件,利用“按文件名访问,按记录进行存取”的管理技术,可以对文件进行插入、删除及修改

操作。文件系统实现了记录内的结构性,但大量文件之间整体无结构。程序和数据之间由文件系统提供存取方法进行转换,使应用程序与数据之间有了一定的独立性,程序员可以不必过多地考虑物理细节,而将精力集中于应用程序的算法,而且数据在存储上的改变不一定反映在程序上,大大节省了维护程序的工作量,但是,文件系统仍存在缺点。

(3) 数据共享性差,冗余度高

在文件系统中,一个文件基本上对应一个应用程序,即文件仍然是面向应用的。当不同的应用程序具有部分相同的数据时,也必须建立各自的文件,而不能共享相同的数据,因此数据的冗余度高,浪费存储空间。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了困难。

(4) 数据独立性差

文件系统文件是为某一特定应用服务的,文件的逻辑结构对该应用程序来说是优化的,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改变,必须修改应用程序,修改文件结构的定义。应用程序的改变,例如使用不同的高级语言编写应用程序等,也将引起文件的数据结构的改变。因此数据与程序之间仍缺乏独立性。可见,文件系统仍然是一个不具有弹性的、整体无结构的数据集合,即文件之间是孤立的,不能反映现实世界事物之间的内在联系。

在文件系统阶段,应用程序与数据之间的对应关系如图 1-4 所示。

3. 数据库系统阶段

20 世纪 60 年代后期以来,计算机用于管理的规模更为庞大,数据量急剧增长,硬件已有大容量磁盘,硬件价格下降;软件则价格上升,使得编制、维护软件及应用程序的成本相对增加;数据处理方式上,联机实时处理要求更多,分布处理也在考虑之中。鉴于这种情况,文件系统的数据管理已满足不了应用的需求,为解决共享数据的需求,随之从文件系统中分离出了专门的软件系统——数据库管理系统,用来统一管理数据。

在数据库系统阶段,应用程序与数据之间的对应关系如图 1-5 所示。

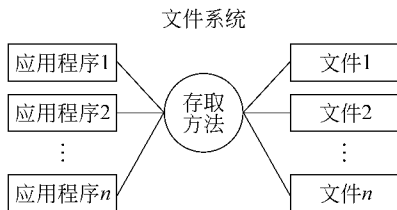


图 1-4 文件系统阶段应用程序与数据之间的对应关系

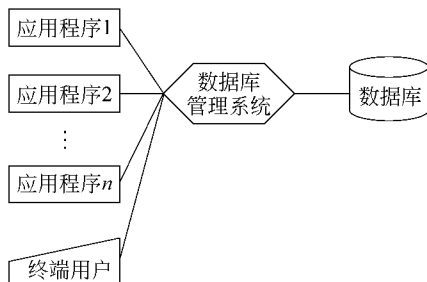


图 1-5 数据库系统阶段应用程序与数据之间的对应关系

以上 3 个阶段应用程序与数据管理的工作任务划分示意图如图 1-6 所示。

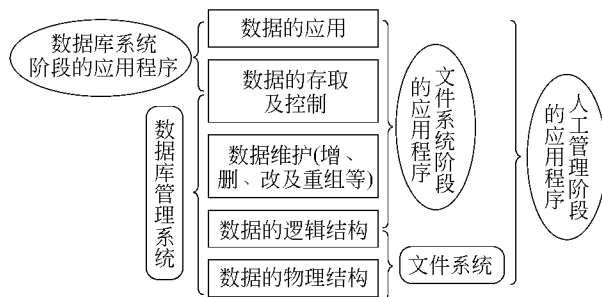


图 1-6 3 个阶段应用程序与数据管理的工作任务划分示意图

综上所述,随着数据管理技术的不断发展,应用程序不断从底层的、低级的、物理的数据管理工作中解脱出来,能独立地、较高逻辑级别地轻松处理数据库数据,从而极大地提高了应用软件的生产力。目前关系数据库系统已逐渐淘汰了网状数据库系统和层次数据库系统,成为当今最流行的商用数据库系统。

1.1.3 数据库系统的特点

与其他两个数据管理阶段相比,数据库系统阶段的数据管理有其自己的特点,主要体现在以下几个方面。

1. 数据结构化

数据结构化是数据库系统与文件系统的根本区别。

在文件系统中,相互独立的文件的记录内部是有结构的。传统文件的最简单形式是等长同格式的记录集合。例如,一个教师人事记录文件的记录格式如图 1-7 所示。

教师人事记录

教师号	姓名	性别	年龄	政治面貌	籍贯	家庭出身	职称	所在系	家庭成员	奖惩情况
-----	----	----	----	------	----	------	----	-----	------	------

图 1-7 教师人事记录格式示例

其中,前 9 项数据是任何教师都必须具有的而且基本上是等长的,而各个教师的后两项数据的信息量大小变化较大。如果采用等长记录形式存储教师数据,为了建立完整的教师档案文件,每个教师记录的长度必须等于信息量最多的教师记录的长度,因而会浪费大量的存储空间,所以最好采用变长记录或主记录与详细记录相结合的形式建立文件。如果将教师人事记录的前 9 项作为主记录,后两项作为详细记录,则教师人事记录变为如图 1-8 所示的记录格式,记录实例教师“王名”的记录如图 1-9 所示。

这样的存储结构可以节省许多存储空间,灵活性也相对提高,但这样建立的文件还有局限性,因为这种结构上的灵活性只针对一个应用而言。一个学校或一个组织涉及许多

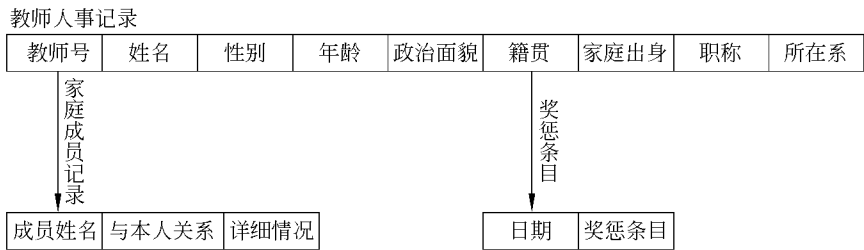


图 1-8 主记录—详细记录格式示例

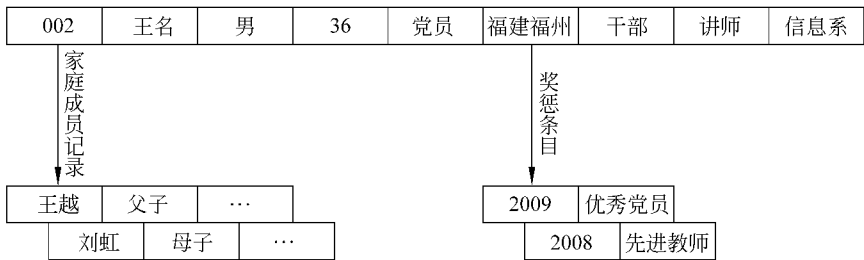


图 1-9 主记录—详细记录的一个实例

应用,在数据库系统中不仅要考虑某个应用的数据结构,还要考虑整个组织中各种应用的数据结构。例如,一个学校的信息管理系统中不仅要考虑教师的人事管理,还要考虑教师的学历情况、任课管理,同时还要考虑教师的科研管理等应用,可按图 1-10 所示的方式为该校的信息管理系统组织其中的教师数据。

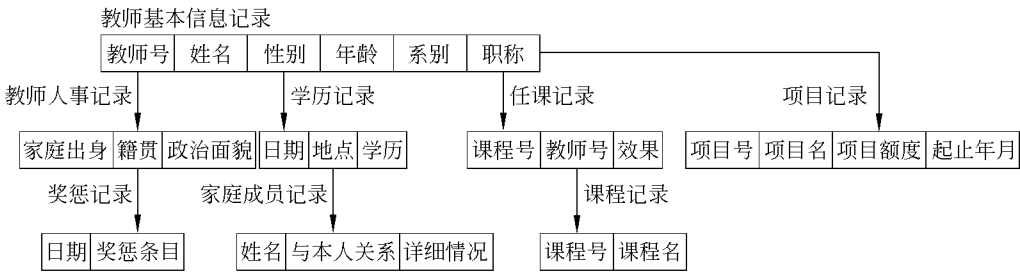


图 1-10 教师数据组织结构

这种数据组织方式为各部分的管理提供了必要的记录,使数据结构化了,这就要求在描述数据时不仅要描述数据本身,还要描述数据之间的联系。

在文件系统中,尽管其记录内已经有了某些结构,但记录之间没有联系。数据库系统实现了整体数据的结构化,故数据结构化是数据库系统的主要特征之一,也是数据库系统与文件系统的本质区别。

在数据库系统中,数据不再针对某一应用,而是面向全组织,是整体结构化的。不仅数据是结构化的,而且存取数据的方式也是很灵活的,可以存取数据库中的某一个数据项(或字段)、一组数据项、一个记录或是一组记录,而在文件系统中,数据的最小单位是记录(一次一记录的读写),粒度不能细到数据项。

2. 数据的共享性高,冗余度低,易扩充

数据库系统从整体角度看待和描述数据,数据不再面向某个应用而是面向整个系统,因此数据可以被多个用户、多个应用共享使用。数据共享可以大大减少数据冗余、节约存储空间,还能够避免数据之间的不相容性与不一致性。

所谓数据的不一致性是指同一数据有不同的复件,而它们的值不完全一致。采用人工管理或文件系统管理时,由于数据被重复存储,当不同的应用使用和修改不同的复件时就容易造成数据的不一致。在数据库系统中,数据能充分共享,减少了由于数据冗余造成的不一致现象。

由于数据面向整个系统,是有结构的数据,其不仅可以被多个应用共享使用,而且容易增加新的应用,这就使得数据系统弹性大、易于扩充,可以适应各种用户的要求,可以取整体数据的各种子集用于不同的应用系统,当应用需求改变或增加时,只要重新选取不同的子集或加上一部分数据便可以满足新的需求。

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两方面。

物理独立性是指用户的应用程序与存储在磁盘上的数据库中的数据是相互独立的,也就是说,数据在磁盘上的数据库中怎样存储是由 DBMS 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改变时,应用程序不用改变。

逻辑独立性是指用户的应用程序与数据库的整体逻辑结构是相互独立的,也就是说,即使数据的逻辑结构改变了,用户程序也可以不修改。

数据独立性是由 DBMS 的三级模式结构与二级映象功能来保证的,将在后面介绍。

数据与程序的独立,把数据的定义从程序中分离出去,加上数据的存取又由 DBMS 负责,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

数据库的共享是并发的共享,即多个用户可以同时存取数据库中的数据,甚至可以同时存取数据库中的同一块数据。

为此,DBMS 还必须提供以下几个方面的数据控制功能。

(1) 数据的安全性控制

数据的安全性是指保护数据以防止不合法的使用造成数据的泄密和破坏。数据的安

全性控制使每个用户只能按规定对某些数据以某些方式进行使用和处理。

(2) 数据的完整性约束

数据的完整性是指数据的正确性、有效性和相容性。完整性约束将数据控制在有效的范围内或保证数据之间满足一定的关系。

(3) 并发控制

当多个用户的并发进程同时存取、修改数据库数据时,可能会发生相互干扰而得到错误结果,使得数据库的完整性遭到破坏,因此必须对多个用户的并发操作加以控制和协调。

(4) 数据库恢复

计算机系统的硬件故障、软件故障、操作员的失误以及故意的破坏等都会影响数据库中数据的安全性与正确性,甚至造成数据库部分或全部数据的丢失。DBMS 必须有将数据库从错误状态恢复到某一已知的正确状态的能力,这就是数据库的恢复功能。

综上所述,数据库是长期存储在计算机内的、有组织的、大量的、可共享的数据集合。它可以提供各种用户共享,具有最小冗余和较高的数据独立性。DBMS 在数据库建立、运用和维护时对数据库进行统一控制,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。

数据库系统的出现使信息系统从以加工数据的程序为中心转向以可共享的数据库为中心的新阶段。这样既便于数据的集中管理,又有利于应用程序的研制和维护,提高了数据的利用率和相容性,提高了决策的可靠性。

目前,数据库已经成为现代信息系统的不可分离的重要组成部分。具有数百万甚至数十亿字节信息的数据库已经普遍存在于科学技术、工业、农业、商业、服务业和政府部门的信息系统中。20 世纪 80 年代后期,不仅在大型机上,而且在多数微型计算机上也配置了 DBMS,使数据库技术得到了更广泛的应用和普及。

数据库技术是计算机领域中发展最快的技术之一,其发展是沿着数据模型的主线展开的。

1.2 数据模型

模型是现实世界特征的模拟和抽象。数据模型也是一种模型,它能实现对现实世界数据特征的抽象,现有的数据库系统均是基于某种数据模型的,因此,了解数据模型的基本概念是学习数据库的基础。

数据模型应该满足 3 个方面的要求:一是能比较真实地模拟现实世界;二是容易为人所理解;三是便于在计算机上实现。

不同的数据模型实际上是提供模型化数据和信息的不同工具。根据模型应用的目的不同,可以将这些模型大致分为两类,分属于两个不同的层次。

第一类模型是概念模型,也称信息模型,它是按用户的观点来对数据和信息建模的,主要用于数据库设计。概念模型一般应具有以下能力。

(1) 具有对现实世界的抽象与表达能力。能对现实世界本质的、实际的内容进行抽象,忽略现实世界中非本质的和与研究主题无关的内容。

(2) 完整、精确的语义表达能力。能够模拟现实世界中本质的、与研究主题有关的各种情况。

(3) 易于理解和修改。

(4) 易于向 DBMS 所支持的数据模型转换。现实世界抽象成信息世界的目的是为了用计算机处理现实世界中的信息。

概念模型作为从现实世界到机器世界(或数据世界)转换的中间模型,它不考虑数据的操作,而只是用比较有效的、自然的方式来描述现实世界的的数据及其联系。

最著名、最实用的概念模型设计方法是 P. P. S. Chen 于 1976 年提出的“实体-联系模型(Entity-Relationship Approach)”,简称 E-R 模型。

另一类模型是数据模型,包括层次模型、网状模型、关系模型等,它是按计算机系统的观点来对数据和信息建模的,主要用于 DBMS 中以实现数据的存储、操纵、控制等。

数据模型是数据库系统的核心和基础,各种机器上实现的 DBMS 软件都是基于某种数据模型的。

为了把现实世界中的具体事物抽象、组织为某一 DBMS 支持的数据模型,人们常常首先将现实世界抽象为信息世界,然后将信息世界转换(或数据化)为机器世界。

1.2.1 数据模型的组成要素

数据模型是模型的一种,是对现实世界数据特征的抽象,它描述了系统的 3 个方面的特性:静态特性、动态特性和完整性约束条件。数据模型一般由数据结构、数据操作和完整性约束 3 个部分组成,是严格定义的一组概念的集合。

1. 数据结构

数据结构用于描述系统的静态特性,是所研究的对象类型的集合。

2. 数据操作

数据操作用于描述系统的动态特性,是对数据库中各种对象及对象的实例允许执行的操作的集合。

3. 完整性约束

完整性约束是一组完整性规则的集合。数据模型应该反映和规定本数据模型必须遵守的基本的、通用的完整性约束条件。数据模型还应该提供自定义完整性约束条件的机制,以反映具体应用所涉及的数据必须遵守的特定的语义约束条件。

1.2.2 概念模型

概念模型是现实世界到机器世界的一个中间层次。现实世界的事物反映到人的头脑中来,人们把这些事物抽象为一种既不依赖于具体的计算机系统又不为某一 DBMS 支持的概念模型,然后再把概念模型转换为计算机上某一个 DBMS 支持的数据模型。概念模型针对于抽象的信息世界,为此先来看信息世界中的一些基本概念。

1. 信息世界中的基本概念

- (1) 实体: 实体是指客观存在并可以相互区别的事物。
- (2) 属性: 属性是指实体所具有的某一特性。
- (3) 码: 码是指唯一标识实体的属性集。
- (4) 域: 域是指属性的取值范围,具有相同的数据类型的数据集合。
- (5) 实体型: 具有相同属性的实体必然具有共同的特征和性质。
- (6) 实体集: 实体集是指同型实体的集合。
- (7) 联系: 在现实世界中,事物内部以及事物之间是有关联的。

两个实体型之间的联系有以下 3 种,如图 1-11 所示。

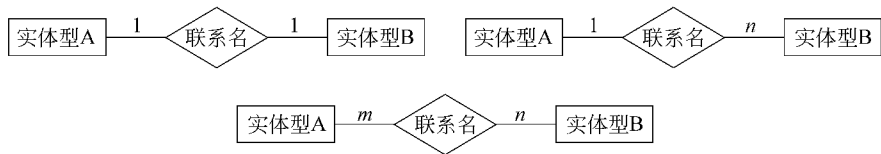


图 1-11 两个实体型之间的 3 种联系表示图

(1) 一对一联系: 如果实体集 A 中的每一个实体,至多有一个实体集 B 与之对应。反之,实体集 B 中的每一个实体,也至多有一个实体集 A 与之对应,则称实体集 A 与实体集 B 具有一对一联系,记作 $1:1$ 。

例如,在一所学校里,一个系只有一个系主任,而系主任只能在某一个系中任职,则系型与系主任型之间(或者说系与系主任之间)具有一对一联系。

(2) 一对多联系: 如果实体集 A 中的每一个实体,实体集 B 中有 n 个实体与之对应。反之,如果实体集 B 中的每一个实体,实体集 A 中至多只有一个实体与之对应,则称实体集 A 与实体集 B 具有一对多联系,记作 $1:n$ 。

例如,一个系中有若干名教师,而每个教师只在一个系中任教,则系与教师之间具有一对多联系。

(3) 多对多联系: 如果实体集 A 中的每一个实体,实体集 B 中有 n 个实体与之对应。反之,如果实体集 B 中的每一个实体,实体集 A 中也有 $m(m \geq 0)$ 个实体与之对应,则称实体集 A 与实体集 B 具有一对多联系,记作 $m:n$ 。

例如,一门课程同时有若干个教师讲授,而一个教师可以同时讲授多门课程,则课程与教师之间具有多对多联系。

3 个实体型之间的一对多联系与多对多联系举例如下。

(1) 例如,对于教师、课程与参考书 3 个实体型,如果一门课程可以有若干个教师讲授,使用若干本参考书,而每个教师只讲授一门课程,每一本参考书只供一门课程使用,则课程与教师、参考书三者间的联系是一对多的,如图 1-12(a)所示。

(2) 又如,有 3 个实体型,项目、零件和供应商,每个项目可以使用多个供应商提供的多个零件,每种零件可由不同供应商供应于不同项目,一个供应商可以给多个项目供应多种零件。为此,这 3 个实体型间的联系是多对多的,如图 1-12(b)所示。

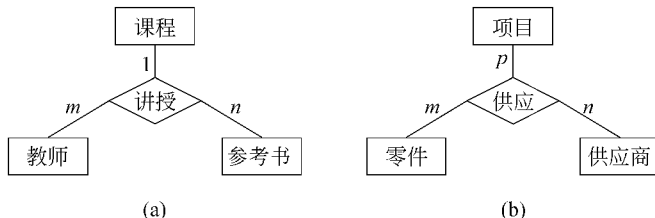


图 1-12 3 个实体型之间的两类联系

2. 概念模型的表示

概念模型的表示方法很多,最常用的是实体-联系方法,该方法用 E-R 图来描述现实世界的概念模型。E-R 图提供了表示实体型、属性和联系的方法。

(1) 实体型:用矩形表示,矩形框内写明实体名。

(2) 属性:用椭圆形表示,椭圆形框内写明属性名,并用无向边将其与相应的实体连接起来。

(3) 联系:用菱形表示,菱形框内写明联系名,并用无向边分别与有关实体连接起来,同时,在无向边旁标上联系的类型(1:1、1:n 或 m:n)。

图 1-13 所示就是一个班级、学生的概念模型(用 E-R 图表示),班级实体型与学生实体型之间很显然是一对多联系。

1.2.3 层次模型

层次模型是数据库系统中最早出现的数据模型,它用树形结构表示各类实体以及实体间的联系。层次模型数据系统的典型代表是 IBM 公司的 IMS (Information Management System, 信息管理系统),这是一个曾经广泛使用的数据库管理系统。

1. 层次模型的数据结构

在层次模型中,有且仅有一个节点无双亲,这个节点称为“根节点”;其他节点有且仅

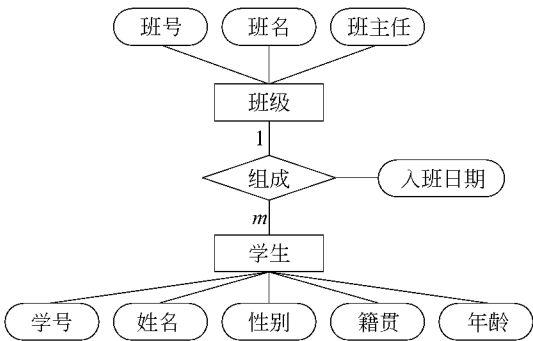


图 1-13 班级、学生关系 E-R 图

有一个双亲。层次模型示例如图 1-14~图 1-16 所示。

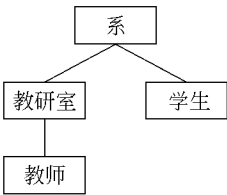


图 1-14 一个层次模型的示例

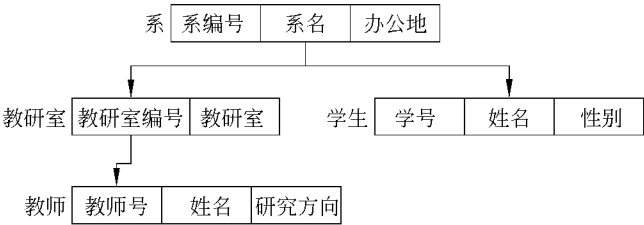


图 1-15 教师、学生数据模型

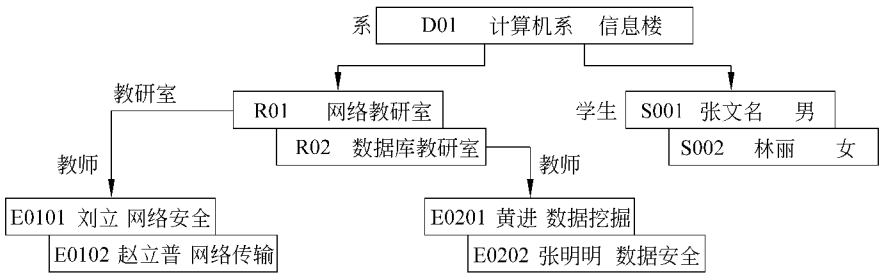


图 1-16 教师、学生数据库的一个值

2. 层次模型的优缺点

(1) 层次模型的优点：层次模型本身比较简单；对于实体间联系是固定的且预先定义好的应用系统，采用层次模型来实现，其性能较优；层次模型提供了良好的完整性支持。

(2) 层次模型的缺点：对于现实世界中非层次的联系，只能通过一些笨拙的方式实现；对节点的插入和删除限制很多，操作起来不方便；对子女节点的查询必须通过双亲节点，缺乏快速定位机制；结构严密，操作命令趋于程序化。

1.2.4 网状模型

网状数据模型的典型代表是 DBTG 系统,也称 COD 系统。它是 20 世纪 70 年代数据系统语言研究会 (Conference On Data Language, CODL) 下属的数据库任务组 (DataBase Task Group, DBTG) 提出的一个系统方案。若用图表示,网状模型是一个网络。

1. 网状模型的数据结构

在数据库中,将满足以下两个条件的基本层次联系集合称为网状模型:①允许一个以上的节点无双亲;②一个节点可以有多于一个的双亲。

它去掉了层次模型的两个限制,此外它还允许两个节点之间有多种联系,因此网状模型可以更直接地去描述现实世界。网状模型的示例如图 1-17 所示。

以教师授课为例,看看网状数据库模型是怎样组织数据的。

按照常规语义,一个教师可以讲授若干门课程,一门课程可以由多个教师讲授,因此教师与课程之间是多对多联系。这里引进一个教师授课的联结记录,它由两个数据项组成,即教师号、课程号,表示某个教师讲授一门课程。

图 1-18 所示为教师、课程、授课的网状数据模型。

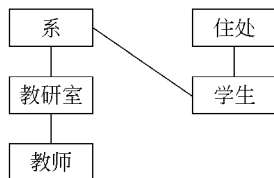


图 1-17 简单网状模型

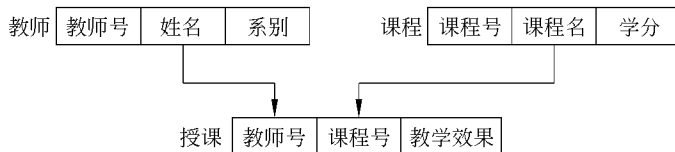


图 1-18 教师、课程、授课的网状数据模型

2. 网状模型的优缺点

(1) 网状模型的优点: 能够更为直接地描述现实世界, 如一个节点可以有多个双亲; 具有良好的性能, 存取效率较高。

(2) 网状模型的缺点: 结构比较复杂, 而且随着应用环境的扩大, 数据库的结构就变得越来越复杂, 不利于最终用户的掌握; 其 DDL、DML 语言复杂, 用户不容易使用。

1.2.5 关系模型

关系模型是目前最重要的一种模型。美国 IBM 公司的研究员 E. F. Codd 于 1970 年发表了题为“大型共享系统的关系数据库的关系模型”的论文, 文中首次提出了数据库系统的关系模型。数据库领域当前的研究工作都是以关系方法为基础的。

关系模型作为数据模型中最重要的一种模型, 也有数据模型的 3 个组成要素。

1. 关系模型的数据结构

关系模型中数据的逻辑结构是一个二维表,它由行和列组成。每一行称为一个元组,每一列称为一个属性(或字段)。通过表 1-1 所示的教师登记表,介绍关系模型中的相关术语。

表 1-1 教师登记表

教师号	姓 名	年 龄	职 称
001	肖正	28	讲师
002	赵琳	40	副教授
003	张立民	44	教授
⋮	⋮	⋮	⋮

- (1) 关系：一个关系对应一个二维表,如表 1-1 所示。
- (2) 元组：二维表中的一行称为一个元组。
- (3) 属性：二维表中的一列称为一个属性,对应每一个属性的名字称为属性名。
- (4) 主码：二维表中的某个属性或是属性组可以唯一确定一个元组,则称为主码,也称为关系键,如图 1-18 中的教师号,可以唯一确定一个元组,则称其为主码,也称为本关系的主码。
- (5) 域：属性的取值范围称为域,如人的年龄一般在 1~120 岁之间,大学生的年龄属性的域是 14~38,性别的域是男和女等。
- (6) 分量：元组中的一个属性值。例如,教师号对应的值 001、002、003 都是分量。
- (7) 关系模式：表现为关系名和属性集的集合,是对关系的具体描述。一般表示为：

关系名 (属性 1,属性 2,⋯,属性 n)

例如上面的关系可以描述为：

教师 (教师号,姓名,年龄,职称)

在关系模型中,实体以及实体间的联系都是用关系来表示的。例如,教师、课程、教师与课程之间的多对多联系在关系模型中可以表示如下：

教师 (教师号,姓名,年龄,职称)

课程 (课程号,课程名,学分)

授课 (教师号,课程号)

关系模型要求关系必须是规范化的,即要求关系必须满足一定规范条件,关系的每一个分量必须是一个不可再分的数据项,不允许表中还有表。例如,表 1-2 中出产日期是可

分的数据项,可以分为年、月、日 3 个子列,因此,该表就是不符合关系模型要求的,必须对其规范化后才能称其为关系。

表 1-2 产品表

产品号	产品名	型 号	出产日期		
			年	月	日
032456	风扇	A134	2006	5	8
⋮	⋮	⋮	⋮	⋮	⋮

2. 关系模型的数据操作与约束条件

关系模型的数据操作主要包括查询、插入、删除和修改数据。这些操作必须满足关系的完整性约束条件,即实体完整性、参照完整性和用户定义完整性。

在非关系模型中,操作对象是单个记录,而关系模型中的数据操作是集合操作,操作对象和操作结果都是关系,即若干元组的集合。关系模型把对数据的存取路径向用户隐蔽起来,从而大大地提高了数据的独立性。

3. 关系模型的存储结构

在关系数据模型中,实体及实体间的联系都用表来表示。在数据库的物理组织中,表以文件形式存储,每个表通常对应一种文件结构,也可以多个表对应一种文件结构。

4. 关系模型的优缺点

关系模型优点如下。

- (1) 有较强的数学理论根据。
- (2) 数据结构简单、清晰,用户易懂易用。
- (3) 关系模型的存取路径对用户透明,具有更高的数据独立性、更好的安全保密性。

关系模型具有查询效率不如非关系模型效率高的缺点。为了提高性能,必须对用户的查询进行优化,增加了开发数据库管理系统的负担。

现实世界、信息世界、机器世界/关系数据库间的术语对照如表 1-3 所示。

表 1-3 现实世界、信息世界、机器世界/关系数据库间术语对照表

现实世界	信息世界	机器世界/关系数据库
事物	实体	记录/元组(或行)
若干同类事物	实体集	记录集(即文件)/元组集(即关系)
若干特征刻画的事物	实体型	记录型/二维表框架(即关系模型)
事物的特征	属性	字段(或数据项)/属性(或列)

续表

现实世界	信息世界	机器世界/关系数据库
事物之间的关联	实体型(或实体)之间的联系	记录型之间的联系/联系表(外码)
事物某特征的所有可能值	域	字段类型/域
事物某特征的一个具体值	一个属性值	字段值/分量
可区分同类事物的特征或若干特征	码	关键字段/关系键(或主码)

1.2.6 面向对象模型

面向对象数据库系统支持面向对象数据模型,即面向对象数据库系统是一个持久的、可共享的对象库的存储者,而一个对象库是一个面向对象模型所定义的对象的可集合体。

1.3 数据库系统结构

可以用多种不同的层次或不同的角度来考察数据库系统的结构。从数据库管理系统内部系统结构看,数据库系统通常采用三级模式结构。从数据库外部的体系结构看,数据库系统的结构分为集中式结构、分布式结构、客户/服务器和并行结构等。

1.3.1 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外模式、模式和内模式,如图 1-19 所示。

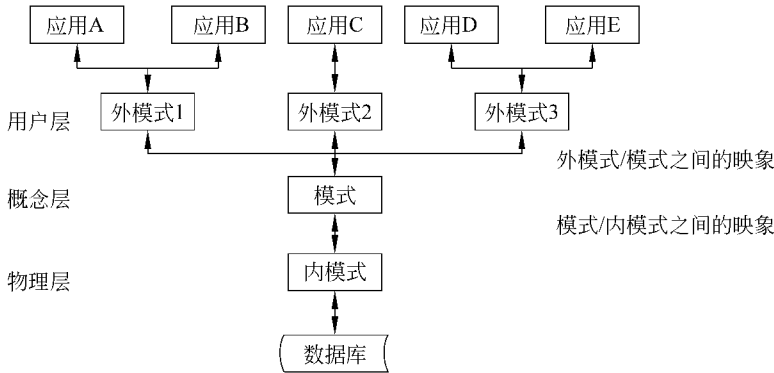


图 1-19 数据库系统的三级模式结构

1. 外模式(External Schema)

外模式也称子模式(Subschema)或用户模式,是三级模式的最外层,它是对数据库用户能够看到和使用的局部数据的逻辑结构与特征的描述。

普通用户只对整个数据库的一部分感兴趣,可根据系统所给的模式,用查询语言或应用程序去操作数据库中的那部分数据,所以,可以把普通用户看到和使用的数据库内容称为视图。视图集也称为用户级数据库,它对应于外模式。外模式通常是模式的子集,一个数据库可以有多个外模式。由于它是各个用户的数据视图,如果不同的用户在不同应用需求、看待数据的方式、对数据保密的要求等方面存在差异,则其外模式描述就是不同的,即使是模式中的同一数据,在外模式中的结构、类型、长度、保密级别都可以被设置成不同的。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用,但一个应用程序只能使用一个外模式。

2. 模式(Schema)

模式又称概念模式,也称逻辑模式,是对数据库中全体数据的逻辑结构和特征的描述,是所有用户的公共数据视图,是数据视图的全部。它是数据库系统模式结构的中间层,既不涉及数据的物理存储细节和硬件环境,也与具体的应用程序、所使用的应用开发工具及高级程序设计语言等无关。

概念模式实际上是数据库数据在逻辑级上的视图,一个数据库只有一个模式。数据库模式以某一种数据模型为基础,统一综合地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。定义模式时不仅要定义数据的逻辑结构,例如数据记录由哪些数据项构成,数据项的名字、类型、取值范围等,而且要定义数据之间的联系,定义与数据有关的安全性、完整性要求。

3. 内模式(Internal Schema)

内模式也称为存储模式,一个数据库只有一个内模式。它是对数据物理结构和存储方式的描述,是数据在数据库内部的表示方式。

1.3.2 数据库的二级映象功能和数据独立性

数据库管理系统在三级模式之间提供了两级映象:外模式/模式映象、模式/内模式映象。这两层映象实现了三个抽象层次的联系和转换以及数据的逻辑独立性和物理独立性。

1. 外模式/模式映象

模式描述的是数据的全局逻辑结构,外模式描述的是数据的局部逻辑结构。数据库系统都有一个外模式/模式映象,它定义了该外模式与模式之间的对应关系。

当模式改变时,由数据库管理员对各个外模式/模式映象作相应改变,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,从而保证了数据与程序的逻辑独立性,简称为数据逻辑独立性。

2. 模式/内模式映象

数据库中只有一个模式,也只有一个内模式,所以模式/内模式映象是唯一的,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了,由数据库管理员对模式/内模式映像作相应改变,可以使模式保持不变,从而应用程序也不必改变,保证了数据与程序的物理独立性,简称为数据物理独立性。

在数据库的三级模式结构中,概念模式即全局逻辑结构是数据库的中心与关键,它独立于数据库的其他层次。

数据库的内模式依赖于它的全局逻辑结构,但独立于数据库的用户视图即外模式,也独立于具体的存储设备。

数据库的外模式面向具体的应用程序,它定义在逻辑模式之上,但独立于内模式和存储设备。

数据库的二级映像保证了数据库外模式的稳定性,从而在根本上保证了应用程序的稳定性,使得数据库系统具有较高的数据与程序的独立性。数据库的三级模式与二级映像使得数据的定义和描述可以从应用程序中分离出去。

1.4 数据库系统组成

数据库系统是指计算机系统中引入数据库后的整个人机系统。为此,数据库系统应由计算机硬件、数据库、计算机软件及各类人员组成。

1. 硬件平台

数据库系统对资源提出了较高的要求:要有足够大的内存存放操作系统、DBMS 的核心模块、数据缓冲区和应用程序;有足够大而快速的磁盘等直接存储设备存放数据库数据,有足够的磁盘空间做数据备份;要求系统有较高的通道能力,以提高数据传送率。

2. 数据库

数据库是存放数据的地方,是存储在计算机内有组织的、大量可共享的数据集合,可以供多用户同时使用,具有尽可能少的冗余和较高的数据独立性,有完整的自我保护能力和数据恢复能力。

3. 软件

数据库系统的软件主要包括以下几个。

- (1) 支持 DBMS 运行的操作系统。
- (2) DBMS: DBMS 可以通过操作系统对数据库的数据进行存取、管理和维护。
- (3) 具有与数据库接口的高级语言及其编译系统。
- (4) 以 DBMS 为核心的应用开发工具,为特定应用环境开发的数据库应用系统。

4. 用户

用户主要有以下几种:数据库管理员(DBA)、系统分析员、数据库设计人员、应用程

序员、最终用户,他们各自的职责如下。

(1) 数据库管理员

- ① 决定数据库中的信息内容和结构。
- ② 决定数据库的存储结构和存取策略。
- ③ 定义数据的安全性要求和完整性约束条件。
- ④ 监控数据库的使用和运行。
- ⑤ 数据库的改进和重组重构。

(2) 系统分析员和数据库设计人员

系统分析员负责应用系统的需求分析和规范说明,要和最终用户及 DBA 相配合,分析确定系统的软、硬件配置,并参与数据库系统的总体设计。

数据库设计人员负责数据库中数据的确定,数据库各级模式的设计。为了合理而良好地设计数据库,数据库设计人员必须深入实践,参加用户需求调查和系统分析。中小型系统中,该人员往往由 DBA 兼任。

(3) 应用程序员

应用程序员负责设计和编写应用系统的程序模块,并进行调试和安装。

(4) 最终用户

最终用户可以分为以下 3 类。

① 偶然用户。这类用户不经常访问数据库,但每次访问数据库时往往需要不同的数据库信息,这类用户一般是企业或组织中的高、中级管理人员。

② 简单用户。数据库的多数用户都是此类,其主要的工作是查询和修改数据库,一般都是通过应用程序员精心设计的且具有良好界面的应用程序存取数据库。银行职员和航空公司的机票出售、预订工作人员都是此类用户。

③ 复杂用户。复杂用户包括工程师、科学家、经济学家、科学技术人员等具有较高科学技术背景的人员。这类用户一般都比较熟悉数据库管理系统的各种功能,能够直接使用数据库语言访问数据库,甚至能够基于数据库管理系统的 API 自己编制具有特殊功能的应用程序。

1.5 数据库技术的研究领域及其发展

1.5.1 数据库技术的研究领域

数据库技术的研究领域十分广泛,概括而言包括以下 3 个方面。

1. DBMS 系统软件的研制

DBMS 是数据库应用的基础,DBMS 的研制包括研制 DBMS 本身及以 DBMS 为核