

数据库基础知识

学习目标

1. 了解数据库的发展历程；
2. 理解数据库的基本概念；
3. 掌握数据模型的基本术语和表示方法；
4. 掌握关系数据库；
5. 理解数据库的设计步骤。

1.1 数据库概述

数据库技术产生于 20 世纪 60 年代末,是数据管理的有效技术,也是计算机科学的重要分支。目前,数据库技术已被广泛应用于各种日常管理信息中。数据库技术是信息系统的核心和基础,它是为实现一定目的而按某种规则组织起来的数据集合,比如,学生档案信息管理、酒店客户信息管理、银行客户信息等。

1.1.1 数据库发展历程

数据库技术的发展可以分为人工管理、文件系统和数据库管理系统 3 个阶段。

1. 人工管理

20 世纪 50 年代中期,计算机主要用于科学计算,当时的计算机硬件只有磁带、卡片和纸带等,还没有磁盘等直接存储设备;软件状况是没有操作系统和数据管理,数据处理方式是批处理。

2. 文件系统

20 世纪 50 年代后期到 60 年代中期,随着硬件、软件技术的发展,硬件方面已经有了磁盘、磁鼓等直接存储设备;软件方面已经有了专门的数据管理软件——文件系统;数据处理方式不但有批处理,而且能够联机实时处理。

3. 数据库管理系统

20 世纪 60 年代末,硬件方面有了大容量磁盘。软件方面,为编制和维护系统软件,应用程序所需成本相对增加,有了联机实时处理、分布式处理的应用需求。如果仍然用文件系统来管理数据,已不能满足应用的需求。为解决多用户、多任务共享数据的要求,实现大量的联机实时数据处理,数据库技术应运而生,出现了统一管理数据的专门软件系统——数据库管理系统。

1.1.2 数据库系统的组成

1. 数据

数据(Data)是数据库中存储的基本对象,是描述事物的符号。数据的形式可以是文字、图形、图像、声音等。数据包括两个方面:一是描述事物特征的数据内容;二是存储在某一种媒体上的数据形式。

2. 数据库

数据库(DataBase,DB)是指存储在计算机内、有组织、可共享的数据集合。数据库中的数据是按一定的数据模型组织、描述和存储的,具有较小的冗余度、较高的数据独立性和易扩展性,并且可以被多个用户、多个应用程序共享。

3. 数据库管理系统

数据库管理系统(Data Base Management System,DBMS)是用户与操作系统之间的数据管理软件。数据库在建立、使用和维护时由数据库管理系统统一管理和控制。

数据库管理系统的主要功能有以下 4 个方面。

(1) 数据定义功能。提供数据定义语言(Data Definition Language,DDL),用于定义数据库中的数据对象。

(2) 数据操纵功能。提供数据操作语言(Data Manipulation Language,DML),用于操纵数据,实现对数据库的基本操作,如查询、插入、更新和删除等。

(3) 数据库的运行和管理。保证数据的安全性、完整性、一致性和多用户对数据的并发处理等。

(4) 数据库的建立和维护功能。提供数据库数据输入、批量装载、数据库转储、介质故障恢复、数据库的重组织及性能监视等功能。

4. 数据库应用系统

数据库应用系统(DataBase Application System,DBAS)是指系统开发人员利用数据库系统资源开发的面向某一类实际应用的软件系统,如学生档案信息管理系统、酒店客户信息管理系统、银行客户信息管理系统等,都是以数据库为基础和核心的计算机应用系统。数据库应用系统主要是面向某领域的终端用户。

5. 数据库管理员

数据库管理员(DataBase Administrator,DBA)是指从事管理和维护数据库管理系统的相关工作人员的统称,主要负责业务数据库从设计、测试到部署交付的全生命周期

管理。

6. 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后的系统构成,一般由计算机系统(CS)、数据库、数据库管理系统、数据库管理员和用户(Users)组成的具有高度组织性的整体。

1.1.3 数据库系统的特点

与人工管理和文件管理相比,数据库系统管理数据具有以下特点。

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,也是数据库系统与人工管理和文件管理的本质区别。在数据库系统中,数据不再针对某一应用,而是面向全组织;不仅数据内部是结构化的,整体数据也是结构化的,数据之间存在着联系。

2. 数据的共享性

数据库系统从整体的角度看待与描述数据,使数据面向整体。这些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,避免了数据之间的不相容性与不一致性。

3. 数据的独立性

数据的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存储在磁盘的数据库中,由数据库管理系统进行统一管理,用户的应用程序只需使用简单的逻辑结构操作数据,无须考虑数据在存储器上的物理位置与结构。

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,即数据的逻辑结构发生改变,用户的程序可以不改变。

4. 数据的存储粒度

在文件系统中,数据存储的最小单位是记录,而在数据库系统中,数据存储的粒度可以小到记录中的一个数据项。因此,数据库中数据存取的方式非常灵活,便于对数据的管理。

5. 数据由 DBMS 统一管理和控制

数据库的共享是并发的(Concurrency)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个数据。

6. 为用户提供了友好的接口

用户可以使用交互式的命令语言,如结构化查询语言 SQL(Structured Query Language)对数据库进行操作,也可以把普通的高级语言,如 C# 语言等和 SQL 结合起来,从而把对数据库的访问和对数据的处理有机地结合在一起。

1.1.4 数据模型

模型(Model)是现实世界的特征和抽象,如一架航模飞机、一辆汽车模型都是具体的

事物模型。数据模型也是一种模型,只不过模拟的对象是数据。根据模型应用的不同层次和目的,可将模型分为概念模型和数据模型。概念模型用于数据库的设计,数据模型是对数据的建模,主要包括层次数据模型、网状数据模型和关系数据模型。

1. 概念模型

概念模型用于信息世界的建模,是现实世界到信息世界的第一层抽象。概念模型是数据库设计的有力工具,也是数据库开发人员与用户进行交流的语言,因此概念模型要有较强的表达能力。目前,最常用的概念模型是实体—联系模型。

(1) 概念模型中的常用术语如下。

① 实体。指客观存在并可相互区别的事物。实体可以是具体的人、事或物,也可以是抽象的概念或联系。例如,一辆车、一名学生、一张数据表、一门课程等都是实体。

② 属性。指实体所具有的某一特性。一个实体可以由一个或多个属性构成。例如,一名学生是实体,他的姓名、性别、出生日期等为属性。

③ 关键字。用来唯一标识实体的属性或属性组合。例如,一名学生是实体,学生的学号就是这个实体的关键字。

④ 联系。指实体之间的对应关系,它反映了现实世界中事物之间的相互关联。

⑤ 实体型。属性值的集合表示一个实体,而属性的集合表示一种实体的类型,称为实体型。例如,学生(学号、姓名、出生日期、专业、就读学校)就是一个实体型。

⑥ 实体集。同一类型的实体集合称为实体集。例如,一所学校的全体学生就是实体集。在数据库中用数据表存储同一类实体,即实体集,表中包含的字段就是实体的属性,表中的每一行记录就是一个实体。

⑦ 域。指属性的取值范围。例如,学生的学号属性这个域就是由第一个学生的学号到最后一个学生的学号所构成的集合。

下面以学生表和成绩表为例,对上面的几个术语进行说明,学生表和成绩表信息如表 1.1 和表 1.2 所示。

表 1.1 学生表

学号	姓名	性别	班级
101	小明	女	18 信计
102	小骆	男	18 电商
201	小陈	女	18 信管

表 1.2 成绩表

学号	课程名	成绩	学分
102	网页设计	77	2
101	C 语言	80	2.5
201	数据库	79	2

学生表、成绩表是实体;学生表的属性有学号、姓名、性别和班级,成绩表的属性有学号、课程名、成绩和学分;学生表和成绩表的关键字是学号;学生表、成绩表都是实体型和实体集;学生表的性别属性域就是由男和女构成的集合,成绩表的成绩属性域就是由 0~100 构成的集合。学生表和成绩表之间的联系就是学号属性。例如,要查找小明 C 语言课程的成绩,首先在学生表中查找姓名属性,得到小明的学号是 101,然后通过学号 101 到成绩表查找学号属性,得到 C 语言成绩为 80。可见实体集(数据表)之间是有联系的,学生表依赖于成绩表,而学号是联系两个实体集的纽带,离开了成绩表,学生表的信息将

不完整;同样,离开了学生表,成绩表的信息也不完整。

(2) 概念模型的表示方法。概念模型设计的常用方法是实体关系方法(E-R方法),也称为E-R图或E-R模型,用E-R图对具体数据进行抽象加工,将实体集合抽象成实体类型,用实体间的关系反映现实世界事物间的内在关系。

E-R图有以下3个要素。

实体:用矩形表示,并在矩形框内标注实体名称。

属性:用椭圆表示,并用直线将其与相应的实体型连接起来。

联系:用菱形表示,并在菱形框内标注联系名,通过直线分别将与其有关的实体型连接起来,同时在直线上标注联系的类型(1:1、1:n或m:n)。图1.1所示为学生表和课程表这两个实体以及彼此之间的联系。

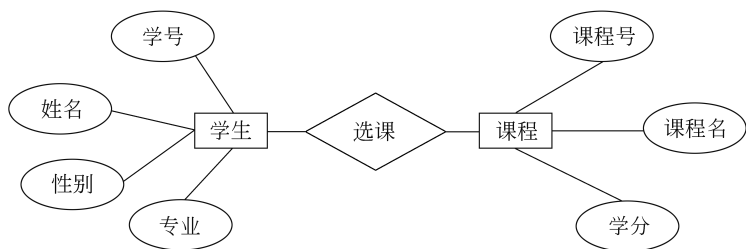


图 1.1 学生表与课程表的 E-R 图

联系的3种类型如下。

① 一对一(1:1)联系。如果对于实体集A中的每一个实体,实体集B中只能有一个实体与之联系;反之亦然,则称实体集A与实体集B具有一对一联系,标记为1:1。例如,人与身份证这两个实体型,一个人只能拥有一张身份证,一张身份证只能属于一个人拥有,因此,人与身份证之间存在一对一联系。

在数据库中,一对一联系表现为主表中的每一条记录只与相关表中的一条记录相关联。例如,人事部门的教师名单表和财务部门的教师工资表之间就是一对一联系,因为一所学校里的一名教师在同一时间只能领取一份工资。

② 一对多(1:n)联系。如果对于实体集A中的每一个实体,实体集B中有n个实体(n大于等于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具有一对多联系,标记为1:n。例如,一个班级拥有多个学生,一个学生只能够属于某个班级。

在数据库中,一对多联系表现为主表中的每一条记录与相关表中的多条记录相关联。即A表中的一条记录在B表中可以有多条记录与之对应,但B表中的一条记录最多只能与A表中的一条记录对应。

③ 多对多(m:n)联系。如果对于实体集A中的每一个实体,实体集B中有n个实体(n大于等于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个实体与之联系,则称实体集A与实体集B具有多对多联系,标记为m:n。例如,一个学生可以选修多门课程,一门课程可以被多个学生选修。

在数据库中,多对多联系表现为主表中的多条记录在相关表中同样可以有多条记录

与之对应。即 A 表中的一条记录在 B 表中可以对应多条记录,而 B 表中的一条记录在 A 表中也可以对应多条记录。

2. 数据模型的组成要素

数据模型由数据结构、数据操作和数据的约束条件 3 个部分组成。

(1) 数据结构。数据结构是所研究对象的集合,这些对象是数据库的组成元素,如表中的字段、记录、名称等。数据结构分为两类,一类是与数据类型、内容、性质有关的对象;另一类是与数据之间的联系有关的对象。

(2) 数据操作。数据操作是指对数据库中各种对象的实例(值)允许执行的操作的集合,包括操作及有关的操作规则。

(3) 数据的约束条件。数据的约束条件是指一组完整性规则的集合。完整性规则是给定的数据模型中数据及其联系所具有的制约和依存规则,以保证数据的正确、有效和保存。

1.1.5 常用的数据模型

目前,常用的数据模型根据其结构的不同可分为层次数据模型、网状数据模型和关系数据模型。

1. 层次数据模型

层次数据模型是指用一棵有向树的数据结构表示各类实体以及实体间的联系。树中每一个节点代表一条记录类型,树状结构表示实体型之间的联系,节点之间用有向线连接。层次数据模型是最早用于商品数据库管理系统的数据模型。

层次数据模型具有以下特征。

(1) 有且仅有一个节点,无父节点,它为树的根。有且仅有一个节点没有双亲,该节点就是根节点。

(2) 其他节点有且仅有一个父节点(根以外的其他节点有且仅有一个双亲节点)。这就使层次数据库系统只能直接处理一对多的实体联系。

在层次数据模型中,同一双亲的子节点称为兄弟节点,没有子节点的节点称为叶节点。

图 1.2 所示为学院、教师、学生之间的层次数据模型。

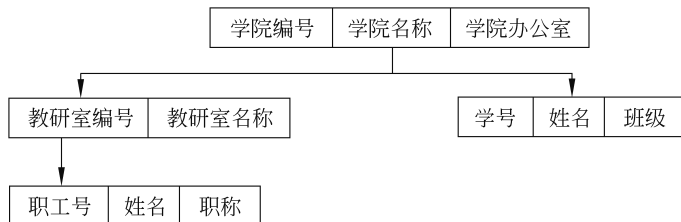


图 1.2 层次数据模型示例

2. 网状数据模型

网状数据模型是指用有向图的网络结构表示实体类型及其实体之间联系的模型。

现实世界中事物之间的联系更多的是非层次关系的联系,用层次数据模型表示这种

关系很不直观,网状数据模型克服了这一弊端,可以清晰地表示这种非层次关系。

网状数据模型具有以下特征。

- (1) 允许有一个以上的节点无双亲。
- (2) 至少有一个节点可以有多个的双亲。

网状数据模型中每个节点表示一个记录型(实体),每个记录型可包含若干个字段(实体的属性),节点间的连线表示记录类型(实体)间的父子关系。

图 1.3 所示为学生与选修课程之间的网状数据模型。

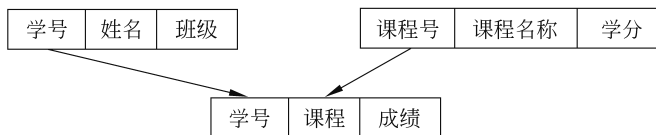


图 1.3 网状数据模型示例

3. 关系数据模型

关系数据模型是指用二维表表示的实体集,用关键字表示实体间联系的数据模型。二维表由行和列组成,通常一张二维表称为一个关系。关系数据模型的具体介绍详见 1.2.1 小节。

1.1.6 数据库系统的体系结构

数据库系统的体系结构从不同的角度可有不同的划分方式。从数据库管理系统的角度可分为三级模式结构,从外到内依次为外模式、模式和内模式,如图 1.4 所示。

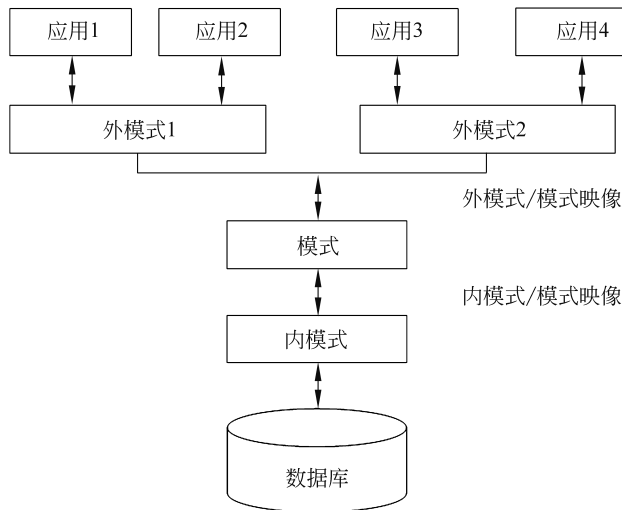


图 1.4 数据库系统的三级模式结构与二级映像

1. 数据库系统的三级模式结构

- (1) 外模式也称为用户模式,是用户与数据库系统的接口,是用户和程序员最后看到

并使用的局部数据逻辑结构和特征。一个数据库可以有若干个外模式。

(2) 模式也称为概念模式或逻辑模式。模式是数据库中全体数据的全局逻辑结构和特征的描述,是所有用户的公共数据视图,用以描述现实世界中的实体及其性质与联系,定义记录、数据项、数据的完整性约束条件及记录之间的联系。

(3) 内模式又称为存储模式,是数据库在物理存储方面的描述,是数据在数据库内部的保存方式。例如,数据是保存在磁盘、磁带还是其他存储介质上,索引按照什么方式组织,数据是否压缩存储,数据是否加密等。

2. 数据库系统的二级映像功能

在数据库系统中,为实现三级模式层次上的联系与转换,数据库管理系统在三级模式之间提供了二级映像功能。二级映像功能保证了数据库系统中的数据具有较高的逻辑独立性和物理独立性。

(1) 外模式/模式映像。模式描述数据的全局逻辑结构,外模式描述数据的局部逻辑结构。对于一个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/模式映像,它定义了外模式与模式之间的映像对应关系。

应用程序是依据数据的外模式编写的,当数据库模式改变时,通过对各个外模式/模式映像作相应改变,可以使外模式保持不变,从而不必修改应用程序,保证了数据与程序的逻辑独立性(数据的逻辑独立性)。

(2) 内模式/模式映像。由于数据库只有一个模式,也只有一个内模式,所以数据库中的内模式/模式映像是一致的。内模式/模式映像定义了数据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发生改变时,如选用了另一种存储结构,数据库管理员通过修改内模式/模式映像,可使模式保持不变,使应用程序不受影响,保证了数据与程序的物理独立性(数据的物理独立性)。

1.2 关系数据库

关系数据库采用了关系数据模型作为数据的组织方式,关系数据模型由关系数据结构、关系操作集合和完整性规则 3 部分组成。

1.2.1 关系数据结构

关系数据结构是由行和列组成的一张二维表格。图 1.5 所示为学生关系表和成绩关系表之间的关系数据模型。

图中两张表分别表示学生关系表和成绩关系表,这两个关系也表示了学生表和成绩表之间的一对多联系,它们之间的联系是由这两个关系的同名属性学号表示的。

关系数据模型具有以下特征。

- (1) 关系可以看成由行和列交叉组成的二维表格,它表示一个实体集。
- (2) 表中的一行称为一个元组,可用来表示实体集中的一个实体。
- (3) 表中的列称为属性,列名即为属性名,表中的属性名不能相同,列的先后顺序无

学生关系表				
学号	姓名	性别	班级	...
101	小明	女	18 信计	...
102	小骆	男	18 电商	...
201	小陈	女	18 信管	...

成绩关系表				
学号	课程名	成绩	学分	...
102	网页设计	77	2	...
101	C 语言	80	2.5	...
201	数据库	79	2	...

图 1.5 关系数据模型示例

关紧要。

(4) 列的取值范围称为域,同列具有相同的域,不同的列也可以有相同的域。如性别的域是男和女,职工年龄和工龄都为数据值域。

(5) 表中的行(元组)不能相同,行的先后顺序无关紧要。

关系数据模型的基本术语如下。

关系:在关系数据模型中,一个关系就是一张二维表格,每个关系都有一个关系名。在数据库中,一个关系存储为一张数据表。

属性:表(关系)中的列称为属性,每一列有一个属性名,对应数据表中的一个字段。

域:表中一个属性的取值范围是该属性的域。

元组:表(关系)中的行称为元组,每一行就是一个元组,对应数据表中的一条具体记录,元组的各分量分别对应关系的各个属性。

候选码:如果表中的某个属性或属性组能唯一地标识一个元组,则称该属性或属性组为候选码(候选关键字)。

主码:若一个表中有多个候选码,可以指定其中一个为主码(主关键字)。

外码:如果表中一个属性(字段)不是本表的主码或候选码,而是另一个表的主码或候选码,则这个属性(字段)称为外码(外部关键字)。

关系模式:一个关系的属性名及其全部属性名的集合简称为关系模式,也就是对关系的描述。关系模式的一般格式如下:

关系名(属性名₁,属性名₂,...,属性名_n)

图 1.5 所示的学生关系表中的每一行就是一条学生的记录,是关系的一个元组。学号、姓名、性别、班级等均是属性名。其中,学号是唯一识别一条记录的属性,是此表的关键字。对于学号属性,其域是 000~999;对于姓名属性,其域是由 2 个汉字组成的字符串;对于性别属性,其域是男或女的取值。

学生关系表的关系模式可记为:学生表(学号,姓名,性别,班级)。

图 1.5 所示的成绩关系表中的学号为成绩关系表的外部关键字。

1.2.2 关系操作集合

在关系数据模型中,常用的关系操作集合包括查询操作和更新操作。查询操作有选择、投影、连接、除、并、交、差等;更新操作有增加、删除、修改等。

1.2.3 完整性规则

关系数据模型的完整性规则是对关系的某种约束条件。关系数据模型有3类完整性约束,即实体完整性、参照完整性和用户自定义完整性。其中,实体完整性和参照完整性是关系数据模型必须满足的完整性约束条件,被称作关系的两个不变性,应该由系统自动支持。

1. 实体完整性

每个关系都有一个主关键字,每行元组主关键字的值是唯一的。主关键字的值不能为空,否则无从识别元组。

例如,在图 1.4 中的学生关系表(学号,姓名,性别,班级)中,学号属性为主关键字,则学号不能取空值。成绩关系表(学号,课程名,成绩,学分)中,学号和课程名属性组合为主关键字,则学号和课程名两个属性不能取空值。

对于实体完整性规则说明如下。

(1) 实体完整性规则是针对基本关系而言。一个基本表通常对应现实世界中的一个实体集。例如,学生关系表对应学生的集合。

(2) 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。例如,学生关系表中的学号,虽然学生的名字相同,但学生的学号却是唯一的。

(3) 在关系数据模型中,以主关键字作为唯一性标识。

(4) 主关键字中的属性值不能为空值或者无意义的值。

2. 参照完整性

现实世界中的实体之间往往存在着某种联系。在关系数据模型中,实体及实体间的联系都是用关系进行描述的,所以存在着关系与关系之间的引用。例如,教师实体和部门实体可以用下面的关系表示,其中主关键字用下划线标识。

教师(教师编号, 姓名, 性别, 年龄, 专业, 部门编号, 职称)

部门(部门编号, 部门名称)

这两个关系之间存在着属性间的引用,即教师关系引用了部门关系的主关键字部门编号。显然,教师关系中的部门编号属性的取值必须是部门关系中确实存在的部门编号。教师关系的部门编号与部门关系的主关键字部门编号相对应,因此,部门编号属性是教师关系的外部关键字。这里的部门关系是被参照关系,教师关系是参照关系。

需要注意的是,外部关键字不一定要与相应的主关键字同名,但在实际应用中,为了便于理解,当外部关键字与相应的主关键字属于不同关系时,往往取相同的名字。

参照完整性规则就是外部关键字与主关键字之间的引用规则。

参照完整性规则:若属性(或属性组)F 是基本关系 R 的外部关键字,且它与基本关