第 1 章

数据库基本原理

本章以文字阐述和典型实例说明相结合的方式介绍下列有关数据库的基本原理。

(1) 数据库的产生、发展及其基本概念；

(2) 数据模型的有关知识、表示方式和基本概念；

(3) 实体集间的三种关系；

(4) 关系的三类完整性；

(5) 关系模型规范化基本知识。

1.1 数据库概述

1.1.1 数据库的产生和发展

数据库技术产生于 20 世纪 60 年代中期，几十年来得到了迅速发展。进入 21 世纪，信息和知识迅速膨胀，数据库技术在组织和利用庞大的信息和知识方面将起越来越重要的作用。

人类活动的整个历史都贯穿着对信息(或数据)的收集、处理、保存和利用。20 世纪 60 年代以来，随着社会生产力的高速发展，信息量急剧膨胀，整个人类社会正成为信息化社会。人们对信息和数据的利用和处理已进入自动化、网络化和社会化阶段，如银行储蓄、股票交易、资料查询、气象预报、机票预订等。这些任务既需要大量数据，又要求快速处理并及时得到结果，是传统的人工方法不可能完成的。飞速发展的计算机技术使上述大规模的数据处理得以实现。即使是很平常的数据处理，借助计算机也可以极大地提高效率。例如，学生的学籍管理是学校的一项重要工作，靠人工查找期末考试有 3 门或以上课程不及格的学生姓名、学号、不及格课程不仅很麻烦，还可能出差错。用计算机管理，就可以快速、准确地完成这项工作。随着计算机和网络技术的迅速发展，现在已经能实现全国几百万名考生、几千所学校的高考网上录取工作。至于全国范围内的股票交易、信用卡支付已经是很平常的事了。

数据库(DataBase)这个名词起源于 20 世纪 50 年代。当时，美国为了军事目的将各种情报集中到一起，揭开了数据库技术的序幕。20 世纪 70 年代，数据库得到了蓬勃发展，网状系统和层次系统占主导，关系数据库系统处于实验阶段。从 20 世纪 80 年代起，

关系数据库系统逐步取代了网状系统和层次系统。此后,关系数据库得到了长足的发展。20 世纪 70 年代中期以后,分布式数据库系统、面向对象的主动数据库系统、智能型数据库系统的相继出现表明数据库技术在不断向更高的水平发展。从目前情况看,关系数据库仍然占绝对的主导地位,并将影响着数据库技术的发展。正因为关系数据库如此重要,像 Oracle、SQL Server、Informix、Sybase 和 Microsoft Access 等大型与中小型关系数据库系统都在不断发展。

到现在,数据库技术的发展已经历了 4 个阶段。

1. 人工管理阶段(20 世纪 50 年代中期以前)

20 世纪 50 年代中期以前,计算机主要用于科学计算。由于科学计算的数据量少,数据和应用程序结合在一起,由人工进行管理。当时,硬件也还没有磁盘,软件也还没有操作系统。

人工管理数据的特点如下。

(1) 数据不保存。数据在运行应用程序时输入,程序执行完释放,不在计算机中保存。

(2) 没有专用软件对数据进行管理。数据的存储结构、存取方法、输入输出方式完全由应用程序确定,数据的改变必然要修改程序。

(3) 数据不共享。数据是面向应用的,即一组数据对应一个程序。各应用程序间很可能存在大量重复数据,所以冗余度极大,浪费存储空间。

(4) 数据不具有独立性。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。

2. 文件系统阶段(20 世纪 50 年代后期至 20 世纪 60 年代后期)

20 世纪 50 年代后期,计算机开始大量应用于管理方面。由于管理事务存在大量数据,并且这些数据需要长期保留,人们采取文件的方式存储、修改数据,将数据和应用程序分离开来。计算机的硬件方面有了磁盘、磁鼓等直接存取存储设备,软件方面在操作系统中有了专门的数据管理软件。

文件系统管理的特点如下。

(1) 数据可以长期保存。大量的数据保存在计算机的外存设备上,可反复进行查询、修改、插入和删除操作。

(2) 有专用软件对数据进行管理。数据由专门的软件即文件系统进行管理,和程序有一定的独立性。程序的修改受数据改变的影响小,工作效率大大提高。

但是,文件系统仍有以下缺点。

(1) 数据共享性差、冗余度大。存放数据的文件是对应一个或几个应用程序的,即文件是面向应用的。不同的应用程序不能共享相同的数据,因此数据的冗余度大,既浪费存储空间,还可能存在不一致性。

(2) 数据独立性差。由于文件系统中的文件是为某一特定应用服务的。所以,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。因此,数据和程序之间仍缺乏独立性。

3. 数据库系统阶段(20 世纪 60 年代后期开始)

20 世纪 60 年代后期以来,计算机大量应用于数据处理、人工智能和计算机辅助设计等领域。这些领域所处理的数据量非常大,还包含许多非数值数据,而且数据间的联系更加复杂,用文件系统管理数据已不适用。为此,需要有一个高度组织的数据管理系统。另外,随着计算机硬件、软件技术的进一步发展,使大量数据集中存储成为可能。数据库系统就是在这样的背景下产生和发展起来的。

数据库系统的特点如下。

(1) 数据结构化。数据库在存储数据的同时既描述数据本身的特点,又描述数据间的联系。

(2) 数据冗余度小。数据库存储数据冗余度小,既节约了存储空间,更避免了冗余数据引起的不一致性。

(3) 数据共享性好。数据库中的数据可以做出各种组合,以最优方式满足不同的需要。

(4) 数据独立性高。数据库中的数据既具有物理独立性,又有逻辑独立性。物理独立性是指用户的应用程序与存储在磁盘上的数据是相互独立的。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。

(5) 数据有统一管理和控制。数据库系统提供了统一的管理软件,数据由数据库管理系统管理和控制,保证了数据的安全性、完整性和保密性。

4. 高级数据库阶段(20 世纪 70 年代中期开始)

数据库技术在商业领域的巨大成功刺激了其他领域对数据库技术的需求。例如,计算机辅助设计/制造(CAD/CAM)、计算机集成制造(CIM)、地理信息系统(GIS)、办公信息系统(OIS)、计算机辅助超大规模集成电路设计(VLSI CAD)等都需要数据库的支持。这些系统在数据类型、数据结构或数据存储方面有特殊要求,传统的数据库系统并不能支持。因此,20 世纪 70 年代中期以来出现了分布式数据库系统、面向对象的主动数据库系统、智能型数据库系统。目前,通常称它们为高级数据库技术。

1.1.2　现实世界、信息世界与数据世界

1. 现实世界

现实世界存在着大量的事物,这些事物可以是具体的,也可以是抽象的。各个事物都有表征自己的各种**特征**。例如,某一个人就是一个事物,他的姓名、性别、身高、体重都是他的特征。

最初,人们是通过眼睛等感官接触现实世界的事物的,如太阳、月亮、树、鸟、颜色、声音、气味等具体的事物。随着社会的进步、技术的发展,人们接触现实世界的事物越来越广,包括更多的抽象事物。更为突出的是,人们需要了解各种事物的更为深刻的特征和它们之间更加复杂的关系。例如,要了解一个人除了姓名、性别、身高、体重外,还需要知道他的其他信息,如身份证号、年龄、民族、政治面貌、文化程度,甚至需要知道他的专业特长、个人爱好、身体状况等。这些都是这个人的特征。可见,考虑问题的不同,同一个事物可能用不同的特征来描述它。

现实世界的每一个事物都有反映自身各个方面的特征。每一个事物的全部特征就反映了该事物本身。每一个事物至少有一个特征。

2. 信息世界

人们观察各种事物,在大脑中形成抽象概念,这就是**信息**(Information)。所以说,**信息世界**就是现实世界的事物在人脑中的抽象。例如,有时我们并没有见到某人,但是通过文字材料知道了他的姓名、性别、身份证号、民族、政治面貌、文化程度等(也就是抽象出来的信息),我们就对他有了基本了解。更为重要的是,我们根据他的这些特征可以把他和其他人(另外的事物)区别开来。

3. 数据世界

显然,从现实世界到信息世界的抽象是和计算机完全无关的。为了用计算机处理信息,人们还需要将信息再进一步抽象为计算机所能识别的数据,这种抽象往往和具体的计算机有关,即同样的信息可能因计算机系统的不同抽象出不同的数据结构。

数据世界就是信息世界中信息的数据化。数据世界的数据表示方法不一定和信息世界的描述一致,例如,在数据世界,可能用"1"和"0"分别表示人的性别的"男"和"女",用某种编码表示不同的民族、政治面貌、文化程度等。这种表示方法便于计算机处理。信息和数据是紧密相关的,在许多场合将它们看作同义词。

在数据世界里,将现实世界诸事物中凡属有限数据集合的特征都用恰当的编码表示是非常必要的。这样做的结果,既节约了存储空间,又减少了出错的可能,更便于查询和统计。实际上,如果对编码赋予更多的含义,就能发挥更大的作用。例如,用学号的前两位表示学生的入学年份;身份证前 6 位表示登记人户口所在地区的代号,中间 8 位是本人的出生年月日;借书证号的第 1 位用不同的字母表示不同的读者对象(例如,用 A、B、C 分别表示学生、教师、其他员工)。这样高质量的编码用来进行数据统计是非常方便的。所以说,编码的质量是影响数据库系统的决定性因素。

1.1.3 数据库基本概念

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

数据(Data)就是描述信息的符号,是数据库中存储的基本对象。随着计算机识别和处理能力的极大提高,现在的数据库处理的数据不仅包括数字和文字形式的信息,还包括图像、声音、文件等形式的信息。

数据处理(Data Processing)是将原始数据转换成信息的过程,包括对数据的收集、整理、分类、存储、排序、统计、加工和分析等,分为人工处理和计算机处理。

2. 数据库

数据库(DataBase,DB)是在计算机系统中按照一定数据模型组织、存储和应用的相互联系的数据集合。数据库既是存放数据的"仓库",又是一种数据处理技术和方法,它总是与一个信息系统相关联,并作为一个信息系统的核心部件而与之共存。

数据库技术(DataBase Technique)是一种对数据进行加工以得到有用信息的计算机软件技术。

3. 数据库管理系统

数据库管理系统(DataBase Management System,DBMS)是一种计算机软件系统。它是数据库系统的核心组成部分。它的主要用途是利用计算机有效地组织、存储、获取和管理数据。

数据库管理系统是用户和操作系统之间的一层数据管理软件。

数据库管理系统由数据描述语言、数据操纵语言和数据库管理运行程序三部分组成。为了提高数据库的开发效率,除了 DBMS,现代数据库还提供了其他一些支持应用开发的工具。

人们通常把以数据库管理系统为核心的应用系统称为**管理信息系统**(Management Information System,MIS)。

4. 数据库系统

数据库系统(DataBase System,DBS)就是以数据库应用为基础的计算机系统。所以,数据库系统不仅包括必须存储的数据,还包括相应的硬件、软件和各类工作人员。在不引起混淆的情况下常把数据库系统简称为数据库。

(1) 数据

数据是按照需求进行采集并以选定的结构存储在数据库中的,是计算机管理中最重要的资料,它不因硬件的更新、软件的更换而改变。数据库通常由两大部分组成:一部分是有关应用所需的工作数据的集合,称为**物理数据库**,它是数据库的主体;另一部分是各级数据结构的描述,称为**描述数据库**。

(2) 硬件

由于数据库系统存储的数据量很大,还要有各种各样的功能,这就要求硬件必须具有较高的性能。

(3) 软件

数据库系统软件主要包括:DBMS、支持 DBMS 的操作系统、与数据库接口的高级语言及其编译系统、以 DBMS 为核心的应用开发工具和为特定环境开发的数据库应用系统等。

(4) 人员

人员是数据库系统的重要组成部分,负责分析、设计、管理和维护数据库。完成这些工作的人员主要有:数据库管理员、系统分析员、应用程序员和最终用户。

1.2　数据模型

随着社会的发展、科技的进步,人们所接触的信息飞速增加,计算机要处理的数据量越来越庞大、相互间的关系越来越复杂。所以,数据库中的大量数据必须按严格的数据模型来组织。数据库中的数据是高度结构化的,它不仅反映数据本身,而且反映数据之间的

关系。**数据模型**就是描述这种关系的数据结构形式,在数据库中使用数据模型对现实世界进行抽象。数据模型是数据库系统的核心和基础。

理想的数据模型应能满足三方面的要求:一是能比较真实地描述现实世界;二是容易被人所理解;三是便于在计算机上实现。到目前为止,还没有哪一种数据模型能够同时满足不同的应用需求。在数据库系统中往往针对不同的情况采用不同的数据模型。

根据模型应用的不同目的可以将模型分为两类,它们分属于两个不同的层次。第一类模型是概念模型,**概念模型**(Idea Model)是现实世界到信息世界的抽象,又称为**信息模型**。第二类模型是数据模型,**数据模型**(Data Model)是信息世界到数据世界的抽象。在数据库领域最常见的数据模型有 4 种:层次模型、网状模型、关系模型和面向对象模型。其中层次模型和网状模型统称为非关系模型。

目前,关系模型是最常用的数据模型。以关系模型为基础建立的数据库管理系统称为**关系数据库管理系统**(RDBMS)。

1.2.1 数据模型的组成要素

一般地说,任何一种数据模型都是严格定义的一组概念的集合。这些概念精确地描述了系统的静态特征、动态特征和完整性约束条件。因此,数据模型通常都是由数据结构、数据操作和数据的约束条件 3 个要素组成。

1. 数据结构

数据结构是所研究的**对象类型**(Object Type)的集合。这些对象是数据库的组成成分,例如关系模型中的域、属性、关系等。数据结构是对系统静态特性的描述。

2. 数据操作

数据操作是指数据库中各种对象(型)的实例(值)允许执行的操作的集合,包括操作和有关的操作规则。数据库主要有检索(查询)和更新(包括插入、删除、修改)两大类操作。数据模型必须定义这些操作的确切含义、操作符号、操作规则(如优先级)以及实现操作的语言。数据操作是对系统动态特性的描述[①]。

3. 数据的约束条件

数据的约束条件是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,用以限定符合数据模型的数据库状态以及状态的变化,以保证数据的正确、有效和相容。

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。例如,在关系模型中,任何关系必须满足实体完整性和参照完整性两个约束条件。

此外,数据模型还应该提供定义完整性约束条件的机制,以反映具体应用涉及的数据必须遵守的特定的语义约束条件。例如,在职工管理系统中规定职工的年龄不能小于 18 岁;在学生管理系统中规定大专生在校学习时间不得超过 6 年。

---

① 萨师煊,王珊. 数据库系统概论[M]. 3 版. 北京:高等教育出版社,2000.

1.2.2　信息世界的基本概念

概念模型是现实世界到数据世界的一个中间层次，用于信息世界的建模，是用户与设计人员之间进行交流的语言，不依赖于具体的计算机硬件和软件。概念模型建立的好坏直接影响到数据模型和整个数据库系统的质量。信息世界的基本概念如下。

1. 实体(Entity)

实体就是现实世界中客观存在并可相互区分的事物。实体既可以是看得见、摸得着的具体的事物，也可以是抽象的概念或联系。例如，某一本书、某一架飞机、某个学生、某次活动、某种现象、某种理论等都是实体。

2. 属性(Attribute)

实体所具有的特征称为**属性**。一个实体由若干个(至少一个)属性来描述。一个实体的所有属性组成实体本身。例如，学生实体可以由学号、姓名、性别、出生年月日、班级等属性组成。而(0100001，冯东梅，女，1980/12/26，01 电子商务 1)就是一个学生(实体)的属性值。

3. 码(Key)

唯一标识实体的属性组称为**码**，通常又称为**关键字**。如果实体有多个码，则可以选定其中一个码为**主码**(Primary Key)，通常又称为**主关键字**。如果实体只有一个码，它就是主码(主关键字)。例如，一个学校里，学生实体的学号是肯定不重复的，所以学号可以作为学生实体的码。如果学生实体中含有身份证号属性，则身份证号也是码。可以在学号和身份证号中选定一个作为主码(主关键字)。通常情况下只关注实体的主码，所以，在不会引起混淆时，通常说码、主码、关键字或主关键字，含义都相同。

4. 域(Domain)

属性的允许取值的集合称为该属性的**域**。例如，学号的域是{7 位数字}(某校规定)，性别的域是{男，女}，班级的域是该校所有班级的集合。

5. 实体型(Entity Type)

具有相同属性的实体必然具有共同的特征。用实体名和其所有属性名集合来抽象并描述同类实体称为**实体型**。例如，学生(学号，姓名，性别，出生年月日，班级)就是一个实体型。

6. 实体集(Entity Set)

同型实体的集合称为**实体集**。例如，某个学校(或某个班级)的全体学生就是一个实体集。

7. 联系(Relationship)

信息世界的不同实体集之间和同一实体集内部都可能存在一定的**联系**。

数据世界的概念和信息世界的概念是相对应的，例如，**数据表**(Data Table)是实体集的数据表示，**记录**(Record)是实体的数据表示，**数据项**(Item)是属性的数据表示。记录由若干数据项组成。

为了用计算机解决数据处理问题，人们必须先对现实世界的事物进行分析，将需要的

信息及其存在的联系做科学的抽象,建立起能正确反映客观事物的概念模型,然后才能设计出理想的数据模型。

1.2.3 实体的联系

信息世界存在的联系有两种:一是同一个实体集内部的联系;二是不同实体集之间的联系。

1. 实体集内部的联系

实体集内部的联系通常指组成该实体的各属性之间的联系。表 1-1 和表 1-2 分别是"学生情况"和"选课及成绩"两个实体集。在表 1-1 中,不同实体的学号都不重复,即学号与实体间有一一对应关系。而不同实体的姓名(或出生年月日、或家庭所在地等)都有可能重复,即姓名(或出生年月日、或家庭所在地)与实体间没有一一对应关系。所以,学号就是"学生情况"这个实体集的关键字。在表 1-2 中,不同实体的学号(或课程号、或成绩)都有可能重复。但是,不同实体的学号加课程号则不可能重复。所以,学号和课程号这两个属性组成的属性组是"选课及成绩"这个实体集的关键字。

表 1-1 学生情况

学 号	姓 名	性别	出生年月日	家庭所在地	家庭人均月收入
0100001	冯东梅	女	1980-12-26	北京	1100
0100002	章 蕾	女	1979-2-18	上海	350
0100007	闻维祥	男	1979-2-24	天津	450
0100008	黎念真	女	1979-8-19	重庆	400
0100009	钟开才	男	1978-8-8	广东	400
0100117	江介敏	女	1978-6-8	湖北	600

表 1-2 选课及成绩

学 号	课程号	考试成绩
0100001	A002	85
0100001	B001	92
0100001	B022	78
0100001	C032	85
0100001	D012	78
0100002	A002	90
0100002	B001	80
0100002	B022	98
0100002	C032	92
0100002	D012	89

2. 实体集之间的联系

对于两个不同的实体集 A 和 B,它们之间的联系通常有以下 3 种方式。

(1) 一对一联系(1∶1)

如果对于实体集 A 中的每一个实体,实体集 B 中至多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。例如,国家和首都、班级和班长、(飞机)

乘客和座位都具有一对一联系。

(2) 一对多联系(1∶n)

如果对于实体集 A 中的每一个实体,实体集 B 中有 $n(n\geqslant 0)$个实体与之联系,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。例如,班级和学生、城市和道路都有一对多联系。

(3) 多对多联系(m∶n)

如果对于实体集 A 中的每一个实体,实体集 B 中有 $n(n\geqslant 0)$个实体与之联系,反之,对于实体集 B 中的每一个实体,实体集 A 中也有 $m(m\geqslant 0)$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。

以上 3 种联系可以用图 1-1 表示。

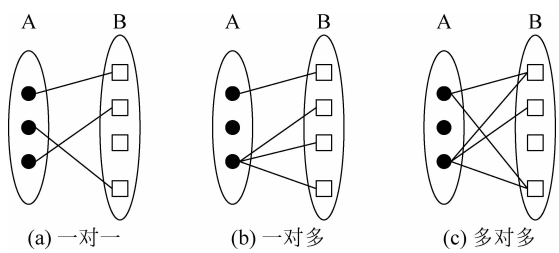

图 1-1　实体集间的联系

显然,一对一联系是一对多联系的特例,一对多联系又是多对多联系的特例。多对多联系直接处理起来很困难,通常是将多对多联系转化为两个一对多联系来处理。

概念模型和各种数据模型均不支持多对多联系,只支持一对一联系和一对多联系。

同一实体集内的各实体之间也可以存在一对一、一对多或多对多联系。例如,学生实体集内部存在“领导与被领导”的联系,即某一学生(班长)“领导”多名(全班)学生,而一个学生仅被一个学生(班长)“领导”,这就是一对多联系。

在复杂问题中,两个以上的实体集之间也往往存在一对一、一对多或多对多联系。

1.2.4　概念模型

对于具体的实际问题,建立正确合理的概念模型是建立数据模型的前提。一个好的概念模型应该考虑和解决的问题有以下几个方面。

(1) 实际问题需要哪些实体集以及各个实体需要哪些属性。

(2) 这些实体集内部和实体集之间有怎样的联系。

(3) 如果存在多对多联系怎样将它转化为一对多联系。

概念模型的表示方法很多,其中最常用的是**实体—联系方法**(Entity-Relationship Approach)。该方法用 **E-R 图**来描述。在 E-R 图中,实体型、属性和联系的表示方法如下。

(1) 实体型:用矩形表示,矩形框内写实体名。

(2) 属性:用椭圆表示,并用无向线段与相应的实体连接。

(3) 联系:用菱形表示,菱形框内写明联系名,并用无向线段与有关的实体连接。同时在无向线段旁标上联系的类型(1∶1,1∶n 或 m∶n)。

应该指出,联系本身也是一种实体型,也可以有属性。如果一个联系有属性,也用无向线段将属性与该联系连接。如图 1-2 所示,用 E-R 图描述了两个实体之间的三类联系。图 1-3 用 E-R 图描述了“学生”和“课程”两个实体及其属性。

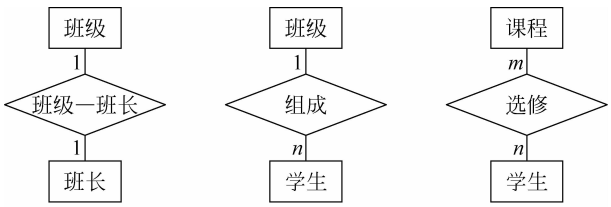

图 1-2 两个实体之间的三类联系

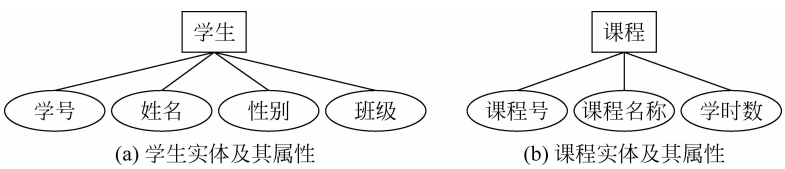

图 1-3 学生实体、课程实体及其属性

下面就“学生学习成绩管理”这一具体问题说明建立概念模型的方法和步骤,同时说明如何将多对多联系转化为一对多联系。

某高等院校实行学分制,每一个学生都可以选修几门课程,每一门课程也都有多个学生选修,允许不同的学生选修不同的课程。

对于这个问题,显然要有“学生”和“课程”这两个实体集。“学生”实体集必须有“学号”作为其关键字,其他的属性根据需要确定,例如“姓名”和“班级”等。“课程”实体集必须有“课程号”作为其关键字,而“课程名称”是必不可少的属性。显然,“学生”和“课程”这两个实体集之间是多对多联系。如果只用这两个实体集,问题解决起来很困难。如前所述,这两者之间的联系本身也是一种实体型。现在,增加一个“选课”实体集来代替这个联系,在“选课”实体集中把“学号”和“课程号”集合起来作为关键字,再包括“成绩”属性。这样一来,“学生”和“选课”之间、“课程”和“选课”之间都是一对多联系。图 1-4 就是这个例子的概念模型。

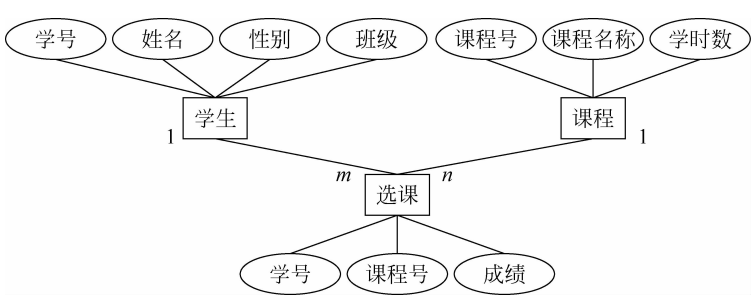

图 1-4 学生学习成绩管理概念模型