

# 第1章 数据库应用基础——学生管理系统案例分析

本章将以设计实现一个学生管理系统为例，介绍数据库应用系统的设计开发过程，并详细介绍与之相关的数据管理技术的发展、数据与数据模型、数据库系统的结构等相关知识，其中包括数据管理技术的发展、数据库管理系统的发展、数据结构与数据模型、数据库系统结构、数据库设计等内容。

## 1.1 学籍管理系统案例分析

本节将以设计开发学生管理系统为例，着重讲解中小型信息管理系统的设计与实现方法，以及完成学生管理系统的设计开发文档。

### 1.1.1 任务的提出

新学期开始了，学生晓灵被班主任良老师叫到了办公室。

良老师说：“晓灵呀！咱们班的同学学习计算机知识有一段时间了。你作为咱们班的班长，能不能利用所学到的计算机知识开发一个软件来管理咱们班的学生信息。这样一来，你既提高了专业知识水平和解决实际问题的能力，也能够更好地管理咱们班，为同学们提供更好地服务！如果这个软件做得好，我们还可以推广到整个年级、整个系乃至整个学院。”

晓灵说：“做这个软件非常有意义，我非常愿意做这件事。但就凭我目前所掌握的那点计算机知识来做这件事难度很大。”

良老师说：“你只要愿意做这个软件，有困难不怕。这件事学院、系领导都非常支持，需要我解决什么困难尽管说好了。我听说咱们班这学期开设了“数据库原理与应用”这门课，讲授这门课的郝老师水平挺高，这个软件怎么做你先问问他。”

晓灵接受了这个任务，首先为这个软件起了一个很好听的名字——晓灵学生管理系统，寓意为：软件虽小，但很灵！晓灵知道自己虽然学习了一些计算机知识，但要想仅依靠这些知识来做这个软件是远远不够的。于是她就去找了讲授“数据库原理与应用”这门课程的郝老师。

晓灵说：“郝老师，现在良老师让我开发一个以管理学生信息为目的的软件。我需要您的帮助，请您指点一下，我需要从哪方面入手？需要先了解哪些知识？”

郝老师：“要想开发这样的一个软件去管理学生信息，从数据量上看可以称之为一个中小型信息管理系统。当然在开发初期可以做得小一点，在使用过程中可以逐步扩展。这就要求在系统设计之初必须具备前瞻性，功能要适度并尽可能超前。我认为要想实现这样的一个系统，你首先应该考虑好以下几个问题：

第一，要确定这个系统的使用者，及其操作计算机的水平、能力和素质。

第二，要确定系统用户对系统功能的要求并且这些功能是否允许分期实现，从而确定系统的边界。

第三，确定系统的使用环境和运行环境。例如，系统是运行在单机上还是运行在网络中？系统可能在哪些操作系统上运行？

第四，系统用户对系统的性能、稳定性有哪些要求？”

郝老师继续说，“你先把这4个问题回答清楚了，做出一个开发文档。之后我们再讨论”。

晓灵原以为开发这个软件不会太难，一了解发现需要了解 and 掌握的东西还挺多。于是她根据郝老师的指导，通过看书和上网搜索相关的知识并认真学习。

### 1.1.2 解决方案

经过几天的忙碌，晓灵根据郝老师的指导和自己所学的知识完成了“晓灵学生管理系统”的开发准备文档。

#### “晓灵学生管理系统”开发准备文档

某学院对学生信息一直采用手工处理方式，但随着学院的发展，学生日益增多，学校对信息的需求量越来越大，对信息处理的要求也越来越高，手工处理学生信息的弊端日益显现。由于管理方式的落后、处理数据的能力有限、数据的冗余度大导致工作效率低，不能及时为领导和老师提供所需信息、数据共享的程度低使数据得不到充分利用，造成数据的极大浪费。解决这些问题最好的办法是实现学生信息管理的自动化、信息化，用计算机处理来代替手工处理。准备设计实现“晓灵学生管理系统”轻松地完成学生相关信息数据的录入、浏览、查询和统计的操作，方便领导和老师对学生信息的掌控，更好地完成学院教书育人的目的，为社会提供合格的劳动者。

从上述情况可以看出，开发学生信息管理系统、实现学生信息管理的计算机化是非常必要的，也是可行的。因为使用信息化的学生信息管理系统可以彻底改变目前学生信息管理工作的现状，能够提高工作效率，能够提供更准确、及时、适用、易理解的信息，能够从根本上解决手工处理中数据之间联系弱、数据冗余大，信息滞后、资源浪费等问题。

为了降低系统开发初期的开发难度，缩短开发时间，“晓灵学生管理系统”将分两期进行设计实现。

根据郝老师的指导，系统开发准备文档解决了以下4个问题：

第一、确定了这个系统的用户：这个系统的用户是学院的领导、教职员工和学生。这些人对于计算机应用水平、操作的能力和素质参差不齐。为了降低系统开发难度和缩短开发时间，《晓灵学生管理系统》一期暂不提供图形化的用户操作界面，故系统一期的用户是领导、教职员工和学生中计算机应用水平和能力较高的人，让他们经过培训就可以使用该系统。二期目标则是提供图形操作界面，让更多的人享受工作的快捷和高效。

第二、确定了系统的功能：“晓灵学生管理系统”是围绕着学生的日常工作展开的，实现了对学生在校内的日常活动的管理。通过本系统的实施可以实现对学生学习成绩、学生奖惩情况、学费缴纳情况、住宿情况的管理。把上述功能做为该系统的一期建设目标，暂未考虑学生辅修第二专业的情况，即一位学生只能学习一个专业、只能属于一个班级；暂未考虑学生借阅图书的情况。这些功能如果需要可在二期中实现。在“晓灵学生管理系统”一期中只考虑实现这些功能。

第三、确定系统的使用环境和运行环境。由于学院校园网已建成多年，“晓灵学生管理系

统”要充分利用现有条件,减少投入,让现有资源发挥最大的作用。所以该系统一期要实现在校园网上的运行,以实现信息资源充分共享,实现小投入大产出的效果。系统二期可以考虑系统与 Internet 的互联,让更多的用户享受更加便捷的应用。因为学院现有计算机中非 Windows 操作系统的非常少,所以该系统的运行环境是基于 Windows 操作系统下的。非 Windows 平台上的应用,如果有需要可以安排在系统二期开发中实现。

第四、要考虑与原有的部分系统、数据的兼容性问题。由于学院中的某些部门已建立了一部分系统,满足了本部门的应用。没有建立系统的部门也存在本部门数据的应用形式。如在 Excel 中建立文档,管理日常应用的信息和数据。在设计与实现“晓灵学生管理系统”中必须要考虑与当前已使用的系统和数据的兼容问题,最好能够把原有系统的数据集成到本系统内。如不能使用这一目标,最低限度也要实现在本系统中能够使用“晓灵学生管理系统”所提供的数据。

第五、用户对系统的性能和稳定性的要求:用户要求系统的运行速度尽可能地要快,应该能够满足学院日常工作中对信息查询和统计的需要。对数据的稳定性要求高,在系统出现问题时要尽可能的恢复数据,以将损失降到最低。在具体应用中还应考虑当前数据库应用系统的情况,如许多数据可能存放于 Word 文档中,也可能存放于 Excel 或 Access 中,那就需要考虑如何将这此不同形式的数据进行有效地集成,或者进行方便、有效地转换。

另外,数据库应用系统的开发是一个复杂的系统工程,它涉及组织的内部结构、管理模式、经营管理过程、数据的收集与处理、软件系统的开发、计算机系统的管理与应用等多个方面。因此,数据库应用系统的开发应在软件开发理论和方法的指导下进行,否则是很难成功的。因此要成功设计实现“晓灵学生管理系统”要经历以下几个阶段:首先对系统进行需求分析,然后进行概念结构设计,第三对系统进行逻辑结构设计,第四进行系统物理实现,第五进行输入数据、系统运行维护工作,最后整理系统的所有文档。

当然为了更好地完成“晓灵学生管理系统”的开发设计与实现工作,还需要了解数据库系统的基本概念与发展、数据模型、关系模型和数据库体系结构、数据库设计等方面的知识。

## 1.2 数据库系统概述

### 1.2.1 数据库系统的基本概念

#### 1. 数据 (Data)

数据是指存储在某一种媒体上能够识别的物理符号。数据的概念包括两个方面:一是描述事物特性的数据内容;二是存储在某一种媒体上的数据形式。

#### 2. 数据库 (Database, DB)

数据库指长期存储在计算机内有组织的、可共享的数据集合。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,并可为各种用户共享。

#### 3. 数据处理

数据处理是指对各种形式的数据进行收集、存储、加工和传播的一系列活动的总和。其目的之一是从大量的、原始的数据中抽取、推导出对人们有价值的信息以作为行动和决策的依

据；目的之二是为了借助计算机技术科学地保存和管理大量复杂的数据，以便人们能够方便而充分地利用这些宝贵的信息资源。

#### 4. 数据库技术

数据库技术是研究数据库结构、存储、设计、管理和使用的一门软件科学。数据库技术是使数据能按一定格式组织、描述和存储，且具有较小的冗余度，较高的数据独立性和易扩展性，并可为多个用户所共享的技术。

#### 5. 数据库管理系统 (Database Management System, DBMS)

数据库管理系统指位于用户与操作系统之间的一层数据管理软件。数据库在建立、运行和维护时由数据库管理系统统一管理、统一控制。数据库管理系统使用户能方便地定义数据和操纵数据，并能够保证数据的安全性、完整性、多用户对数据的并发使用及发生故障后的系统恢复，它的职能是有效地组织和存储数据、获取和管理数据，接受和完成用户提出的访问数据的各种请求。

#### 6. 数据库系统 (Database System, DBS)

数据库系统指在计算机系统中引入数据库后构成的系统，一般由数据库、数据库管理系统 (及其开发工具)、应用系统、数据库管理员和用户构成。

### 1.2.2 数据库系统的发展

数据模型是数据库技术的核心和基础，因此，对数据库系统发展阶段的划分是以数据模型的发展演变作为主要依据和标志。按照数据模型的发展演变过程，数据库技术从开始到现在短短的 30 年中，主要经历了三个发展阶段：

第一代是层次和网状数据库系统，层次数据库系统的典型代表是 1969 年 IBM 公司研制出的层次模型的数据库管理系统 IMS。而在 60 年代末 70 年代初，美国数据库系统语言协会 CODASYL (Conference on Data System Language) 下属的数据库任务组 DBTG (Database Task Group) 提出了若干报告，被称为 DBTG 报告。DBTG 报告确定并建立了网状数据库系统的许多概念、方法和技术，是网状数据库的典型代表。

第二代是关系数据库系统，1970 年 IBM 公司的 San Jose 研究试验室的研究员 Edgar F. Codd 发表了题为《大型共享数据库数据的关系模型》的论文，提出了关系数据模型，开创了关系数据库方法和关系数据库理论，为关系数据库技术奠定了理论基础。

第三代是以面向对象数据模型为主要特征的数据库系统。从 20 世纪 80 年代以来，数据库技术在商业上的巨大成功刺激了其他领域对数据库技术需求的迅速增长。这些新的领域为数据库应用开辟了新的天地，并在应用中提出了一些新的数据管理的需求，从而推动了数据库技术的研究与发展。面向对象数据模型是第三代数据库系统的主要特征之一，数据库技术与多学科技术的有机结合也是第三代数据库技术的一个重要特征。

## 1.3 信息描述与数据模型

所谓信息是客观事物在人类头脑中的反映。人们可以从现实世界中获得各种各样的信息，从而了解世界并且相互交流。但是信息的多样化特性使得人们在描述和管理这些数据时往往力不从心，因此人们把表示事物的主要特征抽象地用一种形式化的描述表示出来，模型方法就是

这种抽象的一种表示。信息领域中采用的模型通常称为数据模型。

根据模型应用的不同目的,可以将模型分为两类或者说两个层次:一是概念数据模型(也称信息模型),是按用户的观点来对数据和信息建模;一是逻辑数据模型(如网状、层次、关系模型),是按计算机系统的观点对数据建模。

### 1.3.1 数据模型及其三要素

一般地讲,数据模型是严格定义的概念集合。这些概念精确地描述系统的静态特性、动态特性和完整性约束条件。因此,数据模型是由数据结构、数据操作和数据的完整性约束三部分组成。

#### 1. 数据结构

数据结构是研究存储在数据库中对象类型的集合,这些对象类型是数据库的组成部分。数据结构是对系统静态特性的描述。数据库系统是按数据结构的类型来组织数据的,因此数据库系统通常按照数据结构的类型来命名数据模型。如层次结构、网状结构和关系结构的模型分别命名为层次模型、网状模型和关系模型。

#### 2. 数据操作

数据操作是指对数据库中各种对象实例所允许执行操作的集合,包括操作和有关的操作的规则。数据操作是对系统动态特性的描述。例如插入、删除、修改、检索、更新等操作,数据模型要定义这些操作的确切含义、操作符号、操作规则以及实现操作的语言等。

#### 3. 数据的完整性约束

数据的约束条件是完整性规则的集合,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。数据模型中的数据及其联系都要遵循完整性规则的制约。数据模型应该提供定义完整性约束条件的机制以及数据应遵守的语义约束条件。

### 1.3.2 数据模型的分类

在实际应用中,为了更好地描述现实世界中的数据特征,常常针对不同的场合不同的目的,采用不同的方法描述数据特征。一般来说,数据模型有如下三种:

#### 1. 概念数据模型

概念数据模型是面向现实世界的数据库模型,它与具体的 DBMS 无关。该数据模型是独立于计算机系统的数据模型。它完全不涉及信息在计算机中的表示,只是用来描述某个特定组织关心的信息结构。它完全按用户的观点对数据进行建模,强调其语义的表达能力,概念简单,易于用户理解。它是对现实世界的第一次抽象,是用户和数据之间进行交流的工具。

#### 2. 逻辑数据模型

逻辑数据模型直接与 DBMS 有关,它有严格的形式化定义,以便在计算机系统中实现。通常用一组无二义性语法和语义的数据库语言来定义、操纵数据库中的数据。它直接面向数据库的逻辑结构,是对现实世界的第二次抽象,通常由数据库设计开发人员来使用。

逻辑数据模型主要有:层次模型、网状模型、关系模型和面向对象模型。在这里就不介绍了,有兴趣的同学可以参考《数据结构》中的相关内容。

#### 3. 物理数据模型

物理数据模型是描述数据在存储介质上的组织方式的数据模型,它不仅与具体的数据管

理系统有关，而且与操作系统和硬件有关。每一种逻辑数据模型在实现时都有对应的物理数据模型，一般来说都是由 DBMS 自动完成物理数据模型的实现工作。

### 1.3.3 概念模型及其表示方法

概念模型是对现实世界的抽象反映，它不依赖于具体的计算机系统，是现实世界到数据世界的一个中间层次，如图 1-1 所示。

#### 1. 信息实体的概念

在信息领域中，数据库技术涉及的主要概念有：

实体：实体是客观存在并可相互区分的事物。

属性：属性是实体所具有的特性。一个实体可以由若干个属性来描述。

键：能够唯一标识实体的属性集称为键，也叫关键字。

实体集：具有相同属性的实体的集合称为实体集。

联系：现实世界中事物之间的联系必然要在信息世界中加以反映。包括两类联系：一个是实体内部的联系，是指实体各个属性之间的联系；一个是实体之间的联系。

#### 2. 实体之间的联系

实体间的联系是错综复杂的，但就两个实体型的联系来说，主要由以下三种情况：

一对一的联系 (1:1)：如果实体集 E1 中的每一个实体至多和实体集 E2 中的一个实体有联系，反之亦然，那么实体集 E1 与 E2 的联系称为“一对一联系”，记为 1:1。例如，每个学生都有一个学号，每位学生和学号之间具有一对一联系。

一对多联系 (1:M)：如果实体集 E1 中的每个实体可以与实体集 E2 中的任意个（零个或多个）实体间有联系；而实体集 E2 中的每个实体至多与实体集 E1 中一个实体有联系，那么称实体集 E1 与实体集 E2 的联系是“一对多联系”，记为 1:M。例如，一个班级内有多名学生，而一名学生只属于一个班。班级与学生之间具有一对多联系。

多对多联系 (M:N)。如果实体集 E1 中的每个实体可以与实体集 E2 中的任意个（零个或多个）实体间有联系，反之亦然，那么称 E1 与 E2 具有多对多联系，记为 M: N。例如，学生在选课时，一个学生可以选修多门课程，一门课程也可以被多名学生选修，则学生和课程之间具有 M: N 联系。

#### 3. 概念数据模型的表示方法

概念模型的表示方法最常用的是实体联系方法 (Entity-Relationship Approach)，这是 P.P.S.Chen 于 1976 年提出的。用这个方法描述的概念模型称为实体联系模型 (Entity-Relationship Model)，简称为 ER 模型。ER 模型是一个面向问题的概念模型，即用简单的图形方式 (E-R 图) 描述现实世界中的数据。这种描述不涉及数据在数据库中表示和存取方法，非常接近人的思维方式，便于系统的开发者与用户之间的交流。后来又提出了扩展实体联系模型 (Extend Entity-Relationship Model)，简称为“EER 模型”。EER 模型目前已经成为一种使用广泛的概念模型，为面向对象的数据库设计提供了有效的工具。

在 ER 模型中，信息由实体类型、实体属性和实体间的联系三种概念单元来表示。

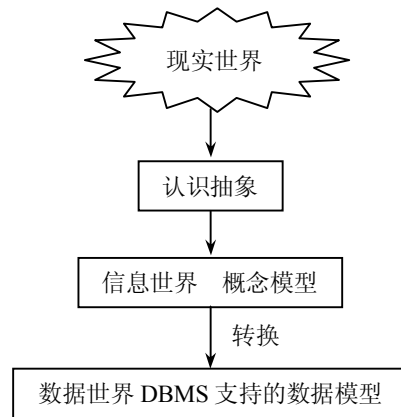


图 1-1 数据抽象过程图

实体类型表示建立概念模型的对象，用长方形表示，在框内写上实体名。

实体属性是实体的说明，用椭圆形表示其属性，并用无向边把实体与其属性连接起来。例如学生实体有学号、姓名、年龄、性别、出生年月等属性，则其 E-R 图如图 1-2 所示。

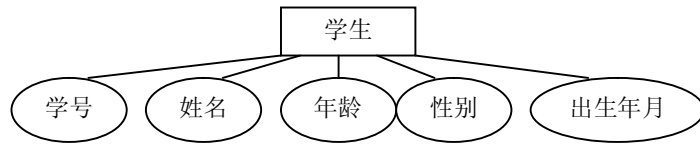


图 1-2 学生实体及其属性

实体间的联系用菱形表示，菱形内要有联系名，并用无向边把菱形分别与有关实体相连接，在无向边旁标上联系的类型，如图 1-3 所示。



图 1-3 实体及其联系图

如果概念模型中涉及的实体带有较多的属性而使实体联系图非常不清晰，我们可以将实体联系图分成两部分，一部分是实体及其属性图，另一部分是实体及其联系图。如图 1-3 所示，我们只给出学生实体与课程实体的联系图，而二者的属性可以单独画出。

## 1.4 关系模型与关系数据库

### 1.4.1 关系模型

关系模型是用规范的二维表结构来表示实体以及实体间联系的模型，由关系数据结构、关系操作集合和关系完整性规则三部分组成。关系数据结构就是由一组关系结构组成的集合，关系操作集合主要包括对表进行查询与更新（插入、修改和删除）数据的操作，关系完整性规则是指对表进行的数据更新操作必须满足一组约束条件。

在关系模型中，每个规范的二维表结构称为关系。

#### 1. 关系模型的数据结构

关系模型的数据结构由规范的二维表结构组成。在关系模型中，将规范的二维表称为关系。每个关系由关系名、关系结构和关系实例组成，对应规范的二维表中的表名、表框架（表头）和表中的行。一个规范的二维表由行和列组成，除第一行（表头）以外，表的每一行称为一个记录（或称为元组）；表中的每一列称为一个字段（或称为属性），每个字段有字段名、字段数据类型和宽度，字段的取值范围称为值域。表头的各列给出了各个字段的名称。

在后面的章节中，为了叙述方便，根据内容不同，将关系称为二维表、表或基本表。

#### 2. 表（关系）的性质

关系模型要求关系数据库中的表必须具有如下性质：

- 表中的每个字段值必须是一个值，不能是值的集合。

- 字段必须是同质的，即同一字段的各个值应是同类型的数据。
- 在同一个表中不能出现相同的字段名。
- 表中不允许有完全相同的记录，即每行记录必须是唯一的。
- 在一个表中记录的次序是任意的。
- 在一个表中字段的次序是任意的。

### 3. 超键、关系键、候选键和主键

在表中能唯一标识记录的字段组合称为该表的超键。

在表中能唯一标识记录且不包括多余字段的字段组合称为该表的关系键。当某些表中具有关系键特性的最小字段组合有多个，即一个表中有多个关系键时，那么这些关系键都称为该表的候选键。

为了唯一地标识表中的每一个记录，保证记录的唯一性，每个表都必须选择一个候选键作为主键。每个表只能有一个主键。对于任意一个表，主键一经选定，通常是不能随意改变的。主键也称为主关系键、键或主码。

## 1.4.2 关系模式和关系数据库

### 1. 关系模式

关系模式是对关系结构（表结构）的描述；关系则是关系模式在某一时刻存储的值，其值是动态的、随时间不断变化的。

在具体的关系数据库管理系统中，使用关系数据库管理系统提供的 SQL 语言的 CREATE TABLE 语句来定义关系模式的名称、关系中的字段、字段类型、宽度、完整性约束等，将定义的语句称为该关系的关系模式。为了便于讨论和描述，关系模式可以表示为：

关系名（字段名 1，字段名 2，…，字段名 n）

其中关系键用下划线标出，n 是关系的目（也可称为度）。

### 2. 关系数据库模式

关系数据库模式是对关系数据库结构的描述，是由一组关系模式组成的集合。一个关系数据库的结构对应一个具体的关系模型。上面给出的学生关系模型中 Student、Course 和 Grade 关系的结构的可用下面的一组关系模式表示：

Student（学号，姓名，年龄，性别，系名）

Course（课程号，课程名，学时数，任课教师）

Grade（学号，课程号，成绩）

### 3. 关系数据库

关系数据库是在一个给定的应用领域关系模型中所有表的集合。

### 4. 关系数据库系统

在关系数据库管理系统支持下，采用关系模型的数据库系统称为关系数据库系统。

## 1.4.3 关系的完整性规则

关系模型的完整性规则是用来约束关系的，以保证数据库中数据的正确性和一致性。关系模型的完整性共有三类：实体完整性、参照完整性和用户定义的完整性。数据完整性由实体完整性和参照完整性规则来维护，实体完整性和参照完整性是关系模型必须满足的完整性约束



条件，由关系数据库管理系统自动支持。

### 1. 实体完整性

实体完整性规则：若属性 A 是基本关系 R 的主键，则属性 A 不能取空值。对于实体完整性的说明如下：

- 一个基本关系对应着一个现实世界的实体集。
- 现实世界中的实体是可区分的，即它们具有某种唯一的标识。
- 关系模型中用主键作为唯一性标识。
- 主码不能取空值，因为主键取空值说明存在某个不可标识的实体，与第二点矛盾。

### 2. 参照完整性

在关系数据库中，关系之间的联系是通过公共属性实现的。这个公共属性是一个表的主键和另一个表的外键。所谓外键是指若一个关系 R 中包含有另一个关系 S 的主键所对应的属性组 F，则称 F 为 R 的外键。外键的值必须是另一个表的主键的有效值或是一个“空值”。

我们先看如表 1-1 和表 1-2 所示的案例 1-1。

#### 【案例 1-1】

键 sID 是表 Student 的主键

表 1-1 学生信息表 (Student) 中的部分数据

sID	sName	sSex	sZhuanYe	sBanji	sRuxueshijian	sSushe	sAddr
040101	温荣奇	男	计算机	z0401	2004-9-1	h1101	天津
040108	高丽华	女	计算机	z0401	2004-9-1	h1201	江苏
040201	高万里	男	信息管理	z0402	2004-9-1	h1101	北京
040203	王向前	男	信息管理	z0402	2004-9-1	h1102	山东
040301	刘常福	女	电子商务	b0403	2004-9-1	h2102	河南

键 payStuID 是表 Pay 的外键

键 payID 是表 Pay 的主键

表 1-2 费用缴纳信息表 (Pay) 中的部分数据

payID	payStuID	payNum	payLeibie	payRiqi	payJingbanren
1	040101	6400	学费	2004-9-1	x002
2	040101	1600	住宿费	2004-9-1	x002
3	040108	6400	学费	2004-9-1	x002
4	040108	3200	学费	2005-9-1	x002
5	040201	6400	学费	2004-9-1	x002
6	040201	800	住宿费	2005-9-1	x002
7	040203	250	其他	2005-12-21	x002
8	040203	6400	学费	2006-9-15	x002
9	040301	1600	住宿费	2004-9-1	x002

在案例 1-1 中，表 Student 和表 Pay 之间通过属性 payStuID 建立了联系。

在使用参照完整性规则时，要注意以下三点：

- 外键和相应的主键可以不同名，只要定义在相同的值域上即可。
- 相关联的两个关系也可以是同一个关系模式，它表示了一个关系中不同元组之间的联系。如课程 Course 关系中的属性先行课 (kcxianxingke) 就是一个外键，其值来源于课程 (Course) 关系中的属性课程编号 (kcID)。
- 外键的值是否为空，应视具体问题而定。如属性先行课号 (kcxianxingke) 的值是可以为空的，可以理解为该课程没有先行课。而属性缴费学生编号 (payStuID) 的值则不能为空。如果为空则会出现语义混淆，即出现“费用到底是谁缴的问题”。

在外键的操作中，还要注意以下两个问题：

- 级联更新 (Cascading Update)：更新主键值的操作，该值由其他表的现有行中的外键列引用。在级联更新中，将更新所有外键值以与新的主键值相匹配。即如果将表 1-1 中 sID 的值 040101 更新成 040102，那么在表 1-12 中的作为外键 payStuID 中值为 040101 的所有数据行中数据项 payStuID 的值也会被自动更新为 040102。
- 级联删除 (Cascading Delete)：删除包含主键值数据行的操作，该值由其他表的现有数据行中的外键列引用。在级联删除中，还将删除其作为外键值被引用的所有数据行。即如果删除表 1-1 中 sID 的值为 040101 的数据行，那么在表 1-12 中的作为外键值被引用的 payStuID 的值为 040101 的所有数据行都会被删除。

### 3. 用户自定义的完整性

用户自定义的完整性规则是针对某一具体数据库的约束条件，由应用环境决定，它反映了某一具体应用所涉及的数据必须满足的语义要求。如学习成绩的取值范围，用户一般会定义在 0~100 之间。数据库管理系统应提供定义和检验这类完整性的机制，以使用统一的方法处理它们而不再由应用程序完成这一任务。

在实际应用中，这类完整性规则一般在建立数据库和基本表的同时进行定义，应用程序的编程人员不需再做考虑。如果某些约束条件没有建立在库表一级，则应用编程人员应在各模块的具体编程中通过程序进行检验和控制。

## 1.5 关系数据库规范化设计

关系数据库的规范化设计理论主要包括三个方面的内容：数据依赖、范式和模式设计方法。数据依赖是关系数据库设计的中心问题，其中函数依赖和多值依赖是最重要的两种形式。数据依赖研究数据之间的联系，范式是关系模式的标准，模式设计方法是自动化设计的基础。从数据依赖的角度出发，在什么是结构合理的关系这一问题上，人们已经做了很多的研究工作。这些工作最终产生了“规范化”理论。在关系数据库的设计实践中，正是通过关系的“规范化”使数据的组织合理化。所谓的“规范化”是把有问题的关系转化成为两个或多个没有这些问题的关系的过程。更重要的是，规范化还可用做检查关系是否合乎需要和正确与否的指南，并且规范化设计理论还对关系数据库结构的设计起着重要的作用。

### 1.5.1 关系模式的设计问题

#### 1. 关系模式的数据冗余的异常问题

在数据管理中，数据冗余一直是影响系统性能的大问题。数据冗余是指同一数据在系统中多次重复出现。在文件系统中，由于文件之间没有联系，引起一个数据在多个文件中重复出现而造成重复存储。数据库系统虽克服了文件系统的这种缺陷，但对于数据冗余问题仍然应予以关注。如果一个关系模式设计得不好，将会产生数据冗余、异常、不一致等问题。为了便于理解，在此举一个“晓灵学生管理系统”中的案例。

【案例 1-2】在“晓灵学生管理系统”的早期设计中，设计有一个关系模式课程情况模式 (kcID, kcName, kcJiaoshi, tTel, kcXuefen, kcxianxingke) 各属性分别表示课程编号（主键）、课程名称、任课老师、联系电话、课程学分和课程先行课，具体数据如表 1-3 所示。

表 1-3 课程情况表

kcID	kcName	kcJiaoshi	tTel	kcXuefen	kcxianxingke
k001	应用英语	刘洪全	80267512	4	
k002	食品加工工艺学	李艳丽	60270604	3.5	
k009	数据库原理与应用	郝亦强	83601316	5	
k010	软件工程	张栋梁	78558388	4.5	k011
k011	软件开发技术	郝亦强	83601316	5	k009
k012	商务网站建设	良易	27085566	5	k010

虽然这个关系模式只涉及 6 个属性，但在使用过程中会出现以下几个问题：

(1) 数据冗余。如果一个教师教几门课程，那么这个教师的联系电话就要重复几次存储。

(2) 操作异常。由于数据的冗余，在对数据进行操作时会引起各种异常：

1) 修改异常。如教师“郝亦强”教两门课程，在关系中就会在属性 tTel 上存在两个元组。如果他的联系电话变了，这两个元组中的联系电话都要改变。若有一个元组中的联系电话未更改，就会造成这个教师的联系电话不唯一：产生不一致现象。

2) 插入异常。如果一个教师刚调来尚未分派教学任务，那么在将教师的姓名和联系电话添加到表中时，在主属性 kcID 上就需要插入空值。而这一操作违反了关系的数据完整性规则中实体完整性规则的要求，无法将这一数据插入数据表中。

3) 删除异常。如果要删除课程编号为 k002 的课程，那么就会把“李艳丽”这个教师的元组删去，同时也把她的联系电话从表中删去了。

因此关系模式 R 的设计不是一个合理的设计。针对上述问题，我们需要将课程情况关系模式分解成教师信息和课程信息两个关系模式，即教师信息 (kcJiaoshi, tTel) 和课程信息 (kcID, kcName, kcJiaoshi, kcXuefen, kcxianxingke)。分别如表 1-4 和表 1-5 所示。

经过这样分解后，前面提到的针对关系模式 R 提到的冗余和操作异常现象就基本消除了。每位教师的姓名及联系电话只存储一次，即使这位教师没有授课任务，其姓名和地址也可存放在教师信息表中。模式分解是解决数据冗余的主要方法，也是规范化理论的一条原则：“如果关系模式中存在数据冗余和操作异常问题，那么就分解这个关系模式”。

表 1-4 教师信息表

kcJiaoshi	tTel
刘洪全	80267512
李艳丽	60270604
郝亦强	83601316
张栋梁	78558388
良易	27085566

表 1-5 课程信息表

kcID	kcName	kcJiaoshi	kcXuefen	kcxianxingke
k001	应用英语	刘洪全	4	
k002	食品加工工艺学	李艳丽	3.5	
k009	数据库原理与应用	郝亦强	5	
k010	软件工程	张栋梁	4.5	k011
k011	软件开发技术	郝亦强	5	k009
k012	商务网站建设	良易	5	k010

但是将课程情况关系模式分解成教师信息和课程信息两个关系模式是否是最佳方案，也不是绝对的。如果要查询某门课程任课教师的联系电话时，就要对关系教师信息和关系课程信息进行连接操作，而连接的代价是很大的，这将影响系统的性能，并且对查询请求的响应速度的影响是致命的，所以说适度、合理的冗余可以提高查询速度。所以在模式设计和进行规范化处理的时候，要根据系统的功能和冗余数据的使用频率来决定。到底什么样的关系模式是合理的？衡量一个关系模式是否合理的标准就是模式的范式（Normal Forms, NF）。

综上所述，规范化的目的可以概括为以下 4 点：

- (1) 把关系中的每一个数据项都转换成一个不能再分的基本项。
- (2) 消除冗余，并使关系的检索简化。
- (3) 消除数据在进行插入、修改和删除时的异常情况。
- (4) 关系模型灵活，易于使用非过程化的高级查询语言进行查询。

### 1.5.2 关系数据库模式的规范化理论

关系数据库设计中，数据库数据合理存储和组织的核心是构造设计一个科学的关系模式，使它能够准确地反映现实世界实体本身以及实体与实体之间的联系，最大限度地减少数据冗余等。这就是关系模式的规范化问题。

#### 1. 关系模式规范化设计

现实世界中的实体可以用关系来描述，但遗憾的是，并非所有的关系都能合理地表示实体。对于某些关系模式，改变其中的数据可能导致一些不希望的结果，我们称之为异常。这些异常可以通过把原有的关系重新定义为两个或多个关系来消除，这种关系重定义的过程即为规范化。

由于某些设计存在着数据不一致性。在使用过程中会发生各种操作异常，显然这样设计

是不行的。一个好的关系模式应该不会发生各种操作异常，数据冗余还应尽可能的小。为了达到这个目标，我们把原有的关系模式分解为符合规范化设计所要求的关系模式。当然这种关系的分解并不是随意的，它必须遵循一定的准则，一般将这些准则称为范式。

范式 (Normal Forms, NF) 的概念和关系模式的规范化问题是由 E.F.Codd 提出的，从 1971 年到 1972 年，E.F.Codd 系统地提出了 1NF、2NF、3NF 的概念，1974 年 Codd 和 Boyce 共同提出了 BCNF，1976 年 Fagin 又提出了 4NF，以后又有人提出 5NF 概念。不同的范式对关系中各属性间的联系提出了不同级别的要求，根据要求级别的高低，一般将关系分为第一范式 (1NF)、第二范式 (2NF)、第三范式 (3NF)、BCNF 范式、第四范式 (4NF) 和第五范式 (5NF)。其中，高级别范式包含在低级别范式中，具体的关系如图 1-4 所示。

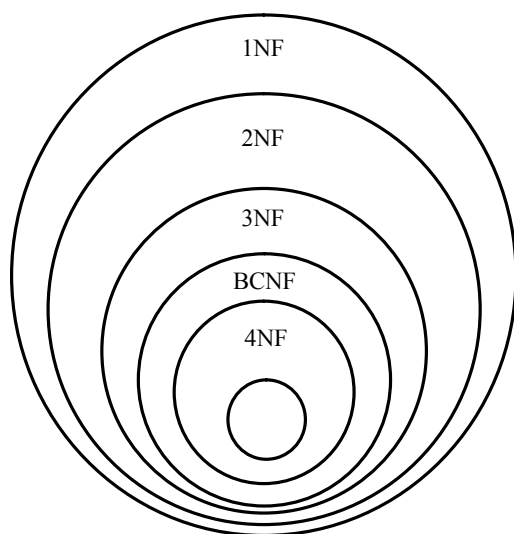


图 1-4 范式关系

## 2. 第一范式 (1NF)

**【定义 1.1】**如果一个关系模式 R 的每个属性的域都只包含单一的值，则称 R 满足第一范式。

通俗地讲，第一范式要求关系中的属性必须是原子值，即为不可再分的基本类型。集合、数组和结构不能作为属性出现，严禁在关系中出现“表中有表”的情况。任何符合关系定义的数据表都满足第一范式的要求。但正如在案例 1-2 中所看到的，符合第一范式中的关系虽然可以使用，但总会有各种操作异常和较大的数据冗余。因此我们必须对此关系进一步规范化，这就导致了第二范式的产生。

## 3. 第二范式 (2NF)

**【定义 1.2】**如果关系模式 R 满足第一范式，且它的所有非主关键字属性完全依赖于整个主关键字（也就是说，不存在部分依赖），则 R 满足第二范式。

根据这一定义，凡是以单个属性作为关键字的关系就自动满足 2NF。因为关键字的属性只有一个，就不可能存在部分依赖的情况。因此，第二范式只是针对主关键字是属性组合的关系。但是，第二范式还远不完美，满足第二范式的关系仍存在着插入、删除和修改的异常。存在这些问题的原因是关系模式中存在传递函数依赖，传递函数依赖是导致数据冗余和操作异常

的另一个原因。所以，满足第二范式的关系模式还需要向第三范式转化，除去非主属性对关键字的传递函数依赖。

#### 4. 第三范式 (3NF)

**【定义 1.3】**如果某关系模式满足第二范式，且它的任何一个非主属性都不传递依赖于任何关键字，则 R 满足第三范式。

换句话说，如果一个关系模式 R 不存在部分函数依赖和传递函数依赖，则 R 满足 3NF。当一个关系模式中存在传递依赖时，应把它分解成两个关系模式，除去传递依赖，从而避免了在第二范式下出现的插入、删除异常，并进一步控制了数据的冗余度。

经过了 1NF、2NF、3NF 的规范化，基本上消除了关系模式中的部分函数依赖、传递函数依赖。但当关系模式具有多个候选键，且这些候选键具有公共属性时，即使该关系满足了第三范式的要求，在操作时仍会出现异常。为了解决这个问题，Boyce 和 Codd 联合提出了一个对第三范式进一步修正的方案，即 BCNF 范式。

#### 5. BCNF 范式

BCNF 范式是由 Boyce 与 Codd 联合提出的。它是对第三范式 (3NF) 的一种改进，通常认为 BCNF 范式是修正的第三范式。

**【定义 1.4】**设一个关系模式  $R(U) \in 1NF$ ，若  $X \rightarrow Y$  且  $Y \not\subset X$  时，X 均包含 R 中的某个关键字，则关系模式  $R \in BCNF$ 。

通俗地讲，关系模式 R 中，若每一个决定因素都包含关键字，则关系模式 R 满足 BCNF 范式。而实际上一个关系模式  $R \in BCNF$ ，则必  $R \in 3NF$ ，但是若  $R \in 3NF$ ，R 未必属于 BCNF。BCNF 比 3NF 的要求更加严格，其原因主要在于第三范式只涉及非主属性与关键字的函数依赖关系，这在关系模式具有多个候选键，且这些候选键具有公共属性时，并不能完全解决数据冗余、更新异常等问题。

一个关系模式如果符合 BCNF，则在函数依赖的范围内已经实现了彻底的分离，消除了插入、删除和修改的异常。但有些异常还会出现，因此人们进一步提出了第四范式 (4NF)、第五范式 (5NF)。由于二者用的较少，甚至第五范式还存在于理论的研究中，所以就不再赘述了。

至此，我们已系统地讨论了关系模式的规范化问题。规范化是通过对已有的关系模式进行分解来实现的。把低一级的关系模式分解为多个高一级的关系模式，使模式中的各关系达到某种程度的分离，让一个关系只描述一个实体或实体间的联系。规范化实质上就是概念的单一化。1NF、2NF、3NF、BCNF 和 4NF 之间逐步深化的过程如图 1-5 所示。

总之，在数据库的设计实践中，关系有时故意保留成非规范化的模式，甚至在规范化后又进行逆规范化处理，这样做通常是为了改善数据库的性能。因此，将关系分解到什么程度要根据实际情况来决定。对大多数的商业系统来说，一般分解到 3NF 就够了，但是有时仍需根据实际情况进一步分解到 BCNF。

最后，从数据库设计实践的角度给出几条经验原则：

(1) 部分函数依赖和传递函数依赖的存在是产生数据冗余、更新异常的重要原因。因此，在关系规范化中，应尽可能消除属性间的这些依赖关系。

(2) 非第三范式的 1NF、2NF 以至非规范化的模式，由于它们性能上的弱点，一般不宜作为数据库模式。

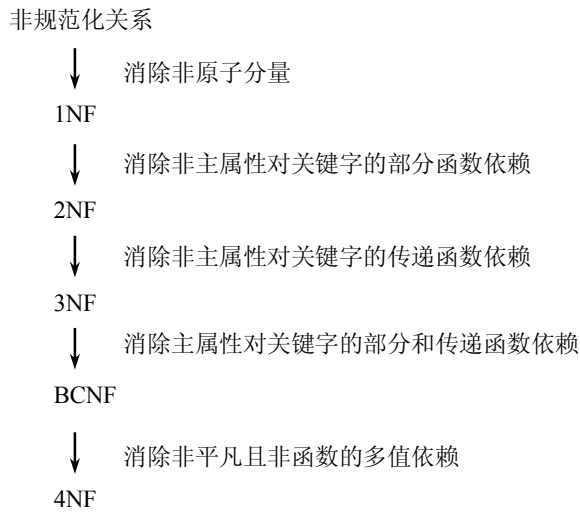


图 1-5 规范化的过程

(3) 由于第三范式的关系模式中不存在非主属性对关键字的部分依赖和传递依赖关系，因而消除了很大一部分冗余和更新异常具有较好的性能，所以一般要求数据库设计达到 3NF。

## 1.6 数据库设计

数据库设计是指对于一个给定的应用环境，提供一个确定最优数据模型与处理模式的逻辑设计，以及一个确定数据库存储结构与存取方法的物理设计，建立起既能反映现实世界中信息和信息联系，满足用户数据要求和加工要求，又能被某个数据库管理系统所接受，同时能实现系统目标，并有效存取数据的数据库。

### 1.6.1 数据库的设计任务与内容

数据库的设计任务是在 DBMS 的支持下，按照应用的要求，为某一部门或组织设计一个结构合理、使用方便、高效的数据库及其应用系统。

数据库设计应包含两方面的内容：一是结构设计，也就是设计数据库框架或数据库结构；二是行为设计，即设计应用程序、事务处理等。

设计数据库应用系统，首先应进行结构设计。数据库结构设计是否合理，直接影响到系统中各个处理过程的性能和质量；另一方面，结构特性又不能与行为特性分离。静态的结构特性设计与动态的行为特性设计相分离，会导致数据与程序不易结合，增加数据库设计的复杂性。

### 1.6.2 数据库的设计方法

目前常用的各种数据库设计方法都属于规范设计法，即都是运用软件工程的思想与方法，根据数据库设计的特点，提出了各种设计准则与设计规范。这种工程化的规范设计方法也是在目前技术条件下设计数据库最实用的方法。

在规范设计法中，数据库设计的核心与关键是逻辑数据库设计和物理数据库设计。逻辑

数据库设计是根据用户要求和特定数据库管理系统的具体特点,以数据库设计理论为依据,设计数据库的全局逻辑结构和每个用户的局部逻辑结构。物理数据库设计是在逻辑结构确定之后,设计数据库的存储结构及其他实现细节。

### 1.6.3 数据库的设计步骤

通过分析、比较和综合各种常用的数据库规范设计方法,我们将数据库设计分为六个阶段,如图 1-6 所示。

#### 1. 需求分析

进行数据库设计首先必须准确了解与分析用户需求(包括数据与处理)。需求分析是整个设计过程的基础,是最困难、最耗费时间的环节。需求分析的结果是否准确地反映了用户的实际要求,将直接影响到后面各个阶段的设计,并影响到设计结果是否合理和实用。

需求分析阶段应该对系统的整个应用情况作出全面的、详细的调查,确定企业组织的目标,收集支持系统总体设计目标的基础数据及其的要求,确定用户的需求,并把这些需求写为用户和数据库设计者都能够接受的文档。

设计人员还应该了解系统将来要发生的变化,收集未来应用所涉及的数据,充分考虑到系统可能的扩充和变动,使系统设计符合未来发展的趋势,并且易于改动,以减少系统维护的代价。进行系统需求分析通常有以下几个环节:

(1) 分析用户活动,产生用户活动图。这一步主要了解用户当前的业务活动和职能,搞清其处理流程(即业务流程),如果一个处理比较复杂,就要把处理分解成若干个子处理,使每个处理功能明确、界面清楚,分析之后画出用户活动图。

(2) 确定系统范围,产生系统范围图。这一步是确定系统的边界。在和用户经过充分讨论的基础上,确定计算机所能进行数据处理的范围,确定哪些工作由人工完成,哪些工作由计算机系统完成。

(3) 分析用户活动所涉及的数据,产生数据流图。深入分析用户的业务处理,以数据流图形式表示出数据的流向和对数据所进行的加工。

数据流图(DFD)是从“数据”和“对数据的加工”两方面表达数据处理系统工作过程的一种图形表示法,具有直观、易于被用户和软件人员双方都能理解的一种系统功能的描述方式。

(4) 分析系统数据,产生数据字典。只有对每个数据都给出确切定义后,才能较完整地描述系统。

数据字典提供对数据描述的集中管理,它的功能是存储和检索各种数据描述并且为 DBA 提供有关的报告。对数据库设计来说,数据字典是进行详细的数据收集和数据分析所获得的主要成果。

数据字典中通常包括数据项、数据结构、数据流、数据存储和处理过程 5 个部分。其中数据项是数据的最小组成单位,若干个数据项可以组成一个数据结构,数据字典通过对数据项

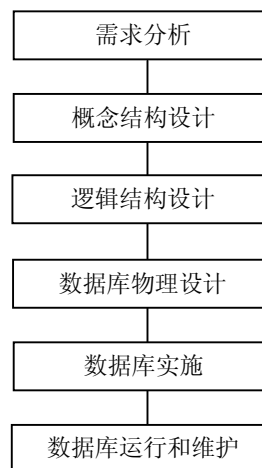


图 1-6 数据库设计的步骤



和数据结构的定义来描述数据流以及数据存储的逻辑内容。

1) 数据项。数据项是数据的最小单位,对数据项的描述,通常包括数据项名、含义、别名、类型、长度、取值范围以及与其他数据项的逻辑关系。

2) 数据结构。数据结构反映了数据之间的组合关系。一个数据结构可以由若干个数据项组成,也可以由若干个数据结构组成,或有若干个数据项和数据结构混合组成。它包括数据结构名、含义及组成该数据结构的数据项名或数据结构名。

3) 数据流。数据流可以是数据项,也可以是数据结构,表示某一加工处理过程的输入或输出数据。对数据流的描述应包括数据流名、说明、流出的加工名、流入的加工名以及组成该数据流的数据结构或数据项。

4) 数据存储。数据存储是处理过程中要存储的数据,它可以是手工凭证、手工文档或计算机文档。对数据存储的描述应包括数据存储名、说明、输入数据流、输出数据流、数据量(每次存取多少数据)、存取频率(单位时间内存取次数)和存取方式(是批处理,还是联机处理;是检索,还是更新;是顺序存取,还是随机存取)。

5) 加工过程。对加工处理的描述包括加工过程名、说明、输入数据流、输出数据流,并简要说明处理工作、频度要求、数据量及响应时间等。

数据字典是在需求分析阶段建立,并在数据库设计过程中不断改进、充实和完善。

## 2. 概念结构设计

概念结构设计是整个数据库设计的关键,其目标是产生反映企业组织信息需求的数据库概念结构,即概念模式。概念模式是独立于计算机硬件结构,独立于支持数据库的 DBMS。

概念设计的任务一般可分为三步来完成:进行数据抽象,设计局部概念模式;将局部概念模式综合成全局概念模式;评审。

利用 ER 方法进行数据库的概念设计,可以分成三步进行:首先设计局部 ER 模式,然后把各局部 ER 模式综合成一个全局 ER 模式,最后对全局 ER 模式进行优化,得到最终的 ER 模式,即概念模式。

(1) 设计局部 ER 模式。为了更好地模拟现实世界,一个有效的策略是“分而治之”,即先分别考虑各个用户的信息需求,形成局部概念结构,然后再综合成全局结构。局部概念结构又称为局部 ER 模式,从如下几方面进行设计:

1) 确定局部结构范围。设计各个局部 ER 模式的第一步,是确定局部结构的范围划分,划分的方式一般有两种:一种是依据系统的当前用户进行自然划分。另一种是按用户要求数据库提供的服务归纳成几类,使每一类应用访问的数据显著地不同于其他类,然后为每类应用设计一个局部 ER 模式。这样做的目的是为了更准确地模仿现实世界,以减少统一考虑一个大系统所带来的复杂性。

局部结构范围的确定要考虑下述因素:

- 范围的划分要自然,易于管理。
- 范围之间的界面要清晰、相互影响要小。
- 范围的大小要适度。

2) 实体定义。每一个局部结构都包括一些实体类型,实体定义的任务就是从信息需求和局部范围定义出发,确定每一个实体类型的属性和键。

实体、属性和联系之间划分的依据通常有以下三点:

- 采用人们习惯的划分。
- 避免冗余，在一个局部结构中，对一个对象只取一种抽象形式，不要重复。
- 依据用户的信息处理需求。

实体类型确定之后，它的属性也随之确定。为一个实体类型命名并确定其键也是很重要的工作。命名应反映实体的语义性质，在一个局部结构中应是唯一的。键可以是单个属性，也可以是属性的组合。

3) 联系定义。ER 模型的“联系”用于描述实体之间的关联关系。一种完整的方式是对局部结构中任意两个实体类型，依据需求分析的结果，考察局部结构中任意两个实体类型之间是否存在联系及确定联系类型。

在确定联系类型时，应注意防止出现冗余的联系（即可从其他联系导出的联系）。如果存在，要尽可能地识别并消除这些冗余联系，以免将这些问题遗留给综合全局的 ER 模式阶段。

联系类型确定后，也需要命名和确定键。命名应反映联系的语义性质，通常采用某个动词命名。联系类型的键通常是它涉及的各实体的键的并集或某个子集。

4) 属性分配。实体与联系都确定下来后，局部结构中的其他语义信息大部分可以用属性描述。这一步的工作有两类：一是确定属性；二是把属性分配到有关实体和联系中去。

确定属性的原则是：属性应该是不可再分解的语义单位；实体与属性之间的关系只能是 1:N 的；不同实体类型的属性之间应无直接关联关系。属性不可分解的要求是为了使模型结构简单化，不出现嵌套结构。

当多个实体类型用到同一属性时，将导致数据冗余，从而可能影响存储效率和完整性约束，因而需要确定把它分配给哪个实体类型。一般把属性分配给那些使用频率最高的实体类型，或分配给实体值少的实体类型。有些属性不宜归属于任一实体，只说明实体之间联系的传递性。

(2) 设计全局 ER 模式。所有局部 ER 模式都设计好以后，接下来就是把它们综合成单一的全局概念结构。全局概念结构不仅要支持所有局部 ER 模式，而且必须合理地表示一个完整、一致的数据库概念结构。

1) 确定公共实体类型。为了给多个局部 ER 模式的合并提供合并的基础，首先要确定各局部结构中的公共实体。公共实体的确定并非一目了然。特别是当系统较大时，可能有很多局部模式，这些局部 ER 模式是由不同的设计人员确定的，因而对同一现实世界的对象可能给予不同的描述。有的作为实体，有的又作为联系或属性。即使都表示成实体，实体名和键也可能不同。在这一步中，我们根据实体名和键来认定公共实体。一般把同名实体作为公共实体的一类候选，把具有相同键的实体作为公共实体的另一类候选。

2) 局部 ER 模式的合并。合并的顺序有时影响处理效率和结果。合并原则的一般顺序是：首先进行两两合并，先合并那些在现实世界中存在联系的局部结构；合并应从公共实体类型开始，最后再加入独立的局部结构。进行二元合并是为了减少合并工作的复杂性，并且使合并结果的规模尽可能小。

3) 消除冲突。由于各类应用不同，且不同的应用通常又是由不同的设计人员设计成局部 ER 模式，因此局部 ER 模式之间不可避免地会有不一致的地方，我们称之为冲突。通常冲突可分为三种类型：

- 属性冲突：属性域的冲突，即属性值的类型、取值范围或取值集合不同。

- 结构冲突：同一对象在不同应用中的不同抽象。同一实体在不同局部 ER 图中属性组成不同，包括属性个数、次序。实体之间的联系在不同的局部 ER 图中呈现不同的类型。
- 命名冲突：包括属性名、实体名、联系名之间的冲突。有同名异义，即不同意义的对象具有相同的名字；有异名同义，即同一意义的对象具有不同的名字。

属性冲突和命名冲突通常采用讨论、协商的方法解决，而结构冲突则需要经过认真分析后才能解决。

设计全局 ER 模式的目的在于把局部 ER 模式在形式上合并为一个 ER 模式，而在于消除冲突，使之成为能够被全系统中所有用户共同理解和接受的统一的概念模型。

(3) 全局 ER 模式的优化。一个好的全局 ER 模式，除了能够准确、全面地反映用户功能需求外，还应满足下列条件：

- 实体类型的个数尽可能少。
- 实体类型所含属性个数尽可能少。
- 实体类型间联系无冗余。

但是，这些条件不是绝对的，要视具体的信息需求与处理需求而定。下面给出几个全局 ER 模式的优化原则。

1) 实体类型的合并。这里的合并不是前面的“公共实体类型”的合并，而是相关实体类型的合并。在公共模型中，实体类型最终转换成关系模式，涉及多个实体类型的信息要通过连接操作获得。因而减少实体类型个数，可减少连接的开销，提高处理效率。一般可以把 1:1 联系的两个实体类型合并。具有相同键的实体类型常常是从不同角度描述现实世界，如果经常需要同时处理这些实体类型，那么也有必要合并成一个实体类型。但这时可能产生大量空值，因此要对存储代价、查询效率进行权衡。

2) 冗余属性的消除。通常在各个局部结构中是不允许冗余属性存在的。但在综合成全局 ER 模式后，可能产生全局范围内的冗余属性。一般同一非键的属性出现在几个实体类型中，或者一个属性值可从其他属性的值导出，此时，应把冗余的属性从全局模式中去掉。

冗余属性消除与否，也取决于它对存储空间、访问效率和维护代价的影响。有时为了兼顾访问效率，有意保留冗余属性，这当然会造成存储空间的浪费和维护代价的提高。

3) 冗余联系的消除。在全局模式中可能存在有冗余的联系，通常利用规范化理论中函数依赖的概念消除冗余联系。

### 3. 逻辑结构设计

逻辑结构设计是将抽象的概念结构转换为所选用的 DBMS 支持的数据模型，并对其进行优化。逻辑设计的目的是把概念设计阶段设计好的全局 ER 模式转换成与选用 DBMS 所支持的数据模型相符合的逻辑结构。这些模式在功能上、完整性和一致性约束及数据库的可扩充性等方面均应满足用户的各种要求。这里只讨论将概念模式转化成关系逻辑数据模型。

转换的一般规则主要有以下三点：

(1) 实体类型的转换：将每个实体类型转换成一个关系模式，实体的属性即为关系模式的属性，实体标识符即为关系模式的键。

(2) 联系类型的转换，要根据以下不同的情况进行不同的处理：

- 若实体间的联系是 1:1 的，可以在两个实体类型转换成的两个关系模式中的任意一个关系模式的属性中加入另一个关系模式的键和联系类型的属性。

- 若实体间的联系是 1:N 的, 则在 N 端实体类型转换成的关系模式中将 1 端实体类型转换成的关系模式的键和联系类型的属性。
- 弱实体: 若实体间的联系是 1:N 的, 而且在 N 端实体类型为弱实体, 转换成的关系模式中将 1 端实体类型 (父表) 的键作为外键放在 N 端的弱实体 (子表) 中。弱实体的主键由父表的主键与弱实体本身的候选键组成, 也可以为弱实体建立新的独立的标识符 ID。
- 若实体间的联系是 M:N 的, 则将联系类型也转换成关系模式, 其属性为两端实体类型的键加上联系类型的属性, 而键则为两端实体键的组合。

(3) 超类和子类的转换: 将超类和子类各转换成一个关系模式, 在子类转换成的关系模式 (子表) 中加入超类转换成关系模式 (父表) 的键, 从而实现父表与子表的联系。由于父表与子表的主键相同, 所以子表的主键也是外键。

在逻辑设计阶段, 仍然要使用关系规范化理论来设计和评价模式。只有这样才能保证所设计的模式不出现数据冗余、更新异常和插入异常, 才能设计出一个好的模式。所以在初始关系模式的基础上还需要进行关系的规范化处理。规范化处理过程分为两个步骤:

1) 确定规范级别: 规范级别取决于两个因素, 一是归结出来的数据依赖的种类, 二是实际应用的需要。首先考察数据依赖集合。在仅有函数依赖时, 3NF 或 BCNF 是适宜的标准, 如还包括多值依赖时, 则应达到 4NF。

2) 实施规范化处理: 确定规范级别之后, 利用模式规范化处理的算法, 逐一考察关系模式, 判断它们是否满足范式要求。若不符合上一步所确定的规范级别, 则利用相应的规范算法将关系模式规范化。在规范化处理过程中, 要特别注意保持函数依赖和无损分解地要求。

最后还需要对规范化处理的结果进行模式评价和模式修正。模式评价的目的是检查已给出的数据库模式是否完全满足用户的功能要求, 是否具有较高的效率, 并确定需要加以修正的部分。模式评价主要包括功能和性能两个方面进行评价。而模式修正是根据模式评价的结果, 对已生成的模式集进行修正。修正的方式依赖于导致修正的原因, 如果因为需求分析、概念设计的疏漏导致某些应用不能得到支持, 则应相应增加新的关系模式或属性; 如果因性能考虑而要求修正, 则可采用合并、分解或选用另外结构的方式进行。如为了提高系统性能适当的增加数据冗余。在经过模式评价及修正的反复多次后, 最终的数据库模式得以确定。

#### 4. 数据库物理设计

数据库物理设计是为逻辑数据模型选取一个最适合应用环境的物理结构 (包括存储结构和存取方法)。数据库的物理结构主要指数据库的存储记录格式、存储记录安排和存取方法。显然, 数据库的物理设计是完全依赖于给定的硬件环境和数据库管理系统。

物理设计可分五步完成, 前三步涉及物理结构设计, 后两步涉及约束和具体的程序设计。

(1) 存储记录结构设计: 包括记录的组成、数据项的类型、长度, 以及逻辑记录到存储记录的映射。

(2) 确定数据存放位置: 可以把经常同时被访问的数据组合在一起。

(3) 存取方法的设计: 存取路径分为主存取路径与辅存取路径, 前者用于主键检索, 后者用于辅助键检索。

(4) 完整性和安全性考虑: 设计者应在完整性、安全性、有效性和效率方面进行分析,

作出权衡。

(5) 程序设计：在逻辑数据库结构确定后，应用程序设计就应当随之开始。物理数据独立性的目的是消除由于物理结构的改变而引起对应用程序的修改。当物理独立性未得到保证时，可能会发生对程序的修改。

#### 5. 数据库的实现

根据逻辑设计和物理设计的结果，设计人员运用 DBMS 提供的数据库语言及其宿主语言在计算机系统上建立起实际数据库结构、装入数据。测试和试运行的过程称为数据库的实现阶段。

数据库实施阶段主要有以下三项工作：

(1) 建立数据库结构。对描述逻辑设计和物理设计结果的程序（即“源模式”），经 DBMS 编译成目标模式和执行后建立的数据库结构。

(2) 装入测试数据对应用程序进行调试。测试数据可以是实际数据，也可由手工生成或用随机数发生器生成。应使测试数据尽可能覆盖现实世界的各种情况。

(3) 装入实际数据，进入试运行状态。测量系统的性能指标是否符合设计目标。如果不符合，刚返回前面几步修改数据库的物理结构，甚至逻辑结构。

#### 6. 数据库运行和维护

数据库应用系统经过试运行后即可投入正式运行。在数据库系统运行过程中必须不断地对其进行评价、调整与修改。

设计一个完善的数据库应用系统，往往是上述六个阶段不断反复的过程。一个数据库设计的过程实际上是设计者对设计经验和设计知识的总结。

在数据库设计过程中要注意以下三个问题：

(1) 数据库设计过程中要注意充分调动用户的积极性。用户的积极参与是数据库设计成功的关键因素之一。用户最了解自己的业务，最了解自己的需求，用户的积极配合能够缩短需求分析的进程，帮助设计人员尽快熟悉业务，更加准确地抽象出用户的需求，减少反复，也能使设计出的系统与用户的最初设想更接近。同时用户参与意见，双方共同对设计结果承担责任也可以减少数据库设计的风险。

(2) 应用环境的改变、新技术的出现等都会导致应用需求的变化，因此设计人员在设计数据库时必须充分考虑到系统的可扩充性，使设计易于变动。一个设计优良的数据库系统应该具有一定的可伸缩性，应用环境的改变和新需求的出现一般不会推翻原设计，不会对现有的应用程序和数据造成大的影响，而只是在原设计基础上做一些扩充即可满足新的要求。

(3) 系统的可扩充性最终都是有一定限度的。当应用环境或应用需求发生巨大变化时，原设计方案可能终将无法再进行扩充，必须推倒重来，这时就会开始一个新的数据库设计的生命周期。但在设计新数据库应用的过程中，必须充分考虑到已有应用，尽量使用户能够平稳地从旧系统迁移到新系统。

数据库系统正式运行标志着数据库设计与应用开发工作的结束和维护阶段的开始。而在数据库系统运行维护阶段，通常有以下四个主要任务：

(1) 维护数据库的安全性与完整性：检查系统安全性是否受到侵犯，及时调整授权和密码，实施系统转储与备份，以便在发生故障后及时恢复数据。

(2) 监测并改善数据库运行性能：对数据库的存储空间状况及响应时间进行分析评价，

结合用户反应确定改进措施，实施再构造或再格式化。

(3) 根据用户要求对数据库现有功能进行扩充。

(4) 及时改正运行中发现的系统错误。

要充分认识到，数据库系统只要在运行，就要不断地进行评价、调整、修改。如果应用变化太大，再组织工作已无济于事，那么表明原数据库应用系统生存期已结束，应该设计新的数据库应用系统了。

#### 1.6.4 晓灵学生管理系统的设计

“晓灵学生管理系统”围绕着学生的日常管理工作展开，涉及了学生在校内的日常活动。通过本系统的实施可以实现对学生学习成绩、奖惩情况、学费缴纳情况、住宿情况的管理。出于某些原因的考虑，本系统未考虑学生辅修第二专业的情况，即一位学生只能学习一个专业、只能属于一个班级。未考虑学生借阅图书的情况以及学生用教材信息的管理。

通过对学生管理系统的分析与抽象，整理设计了“晓灵学生管理系统”的实体联系模型如图 1-7 所示。

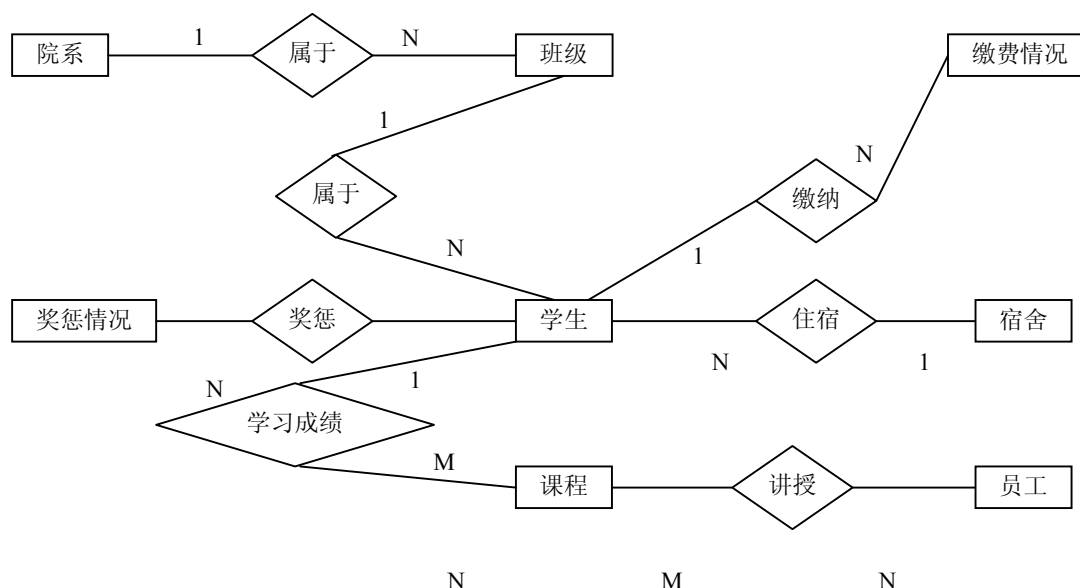


图 1-7 “晓灵学生管理系统”的 E-R 图

在图 1-7 所示的 ER 图，将其中的实体和联系转换成以下关系模式：

1. 院系关系（院系编号，名称，负责人）
2. 教职工关系（教职工号，教师姓名，性别，生日，岗位类别，学历，职称，隶属院系，所学专业，联系电话，家庭地址）
3. 课程关系（课程编号，课程名称，任课教师，学分，课程简介，课程状态，课程先行课）
4. 班级关系（班级编号，班级名称，学制，所属院系，班主任，班长）
5. 宿舍关系（宿舍编号，宿舍电话，宿舍长，宿舍管理员）
6. 学生关系（学号，姓名，性别，生日，所学专业，所属班级，入学时间，所住宿舍，

家庭地址, 联系电话)

7. 学习成绩关系 (学号, 课程号, 成绩)
8. 费用缴纳关系 (缴费学生学号, 缴纳金额, 缴费类别, 缴纳日期, 经办人)
9. 奖惩关系 (奖惩编号, 奖惩人, 奖惩类别, 奖惩内容, 奖惩时间, 是否生效, 相关文件)

针对上述 ER 图, 由于版面的原因未将各实体或联系的属性列出。

根据数据库所在系统硬件的具体情况和 SQL Server 2000 关系数据库系统的特点, 进行了数据库物理模式的设计。设计结果如表 1-6 至表 1-14 所示。

表 1-6 课程信息表 (Course)

序号	字段名	字段类型	说明	备注
1	kcID	char(6)	课程编号, 主键	
2	kcName	varchar(20)	课程名称, 不能为空	
3	kcJiaoshi	char(6)	任课教师	教师工号, 外键约束
4	kcXuefen	smallint	学分	
5	kcZhuangtai	char(4)	课程状态	可开、未开、已开
6	kcJianjie	text	课程简介	超过 8KB 用 text
7	kcxianxingke	char(6)	课程的先行课	

表 1-7 院系信息表 (College)

序号	字段名	字段类型	说明	备注
1	colID	char(6)	院系编号, 主键	
2	colName	varchar(20)	院系名称, 不能为空	
3	colFuzeren	char(6)	院系负责人	员工编号, 外键约束

表 1-8 教师信息表 (Teacher)

序号	字段名	字段类型	说明	备注
1	tID	char(6)	教职工号, 主键	
2	tName	varchar(20)	教师姓名, 不能为空	
3	tSex	char(2)	教师性别	
4	tBirthday	smalldatetime	出生日期	
5	tGangwei	char(4)	岗位类别	教师、教辅、行政、其他
6	tXueli	char(6)	学历	博士、硕士、大本、大专
7	tZhicheng	varchar(10)	职称	
8	tYuanxi	char(6)	隶属院系	院系编号, 外键约束
9	tZhuanye	varchar(20)	所学专业	
10	tTel	varchar(20)	联系电话	
11	tAddr	varchar(40)	家庭住址	

表 1-9 班级信息表 (Class)

序号	字段名	字段类型	说明	备注
1	clsID	char(6)	班级编号, 主键	
2	clsName	varchar(20)	班级名称, 不能为空	
3	clsXuezhzhi	smallint	学制	
4	clsYuanxi	char(6)	所属院系	院系编号, 外键约束
5	clsBanzhuren	char(6)	班主任	教师工号, 外键约束
6	clsBanzhang	char(6)	班长	学生编号, 外键约束

表 1-10 学习成绩表 (Grade)

序号	字段名	字段类型	说明	备注
1	sID	char(6)	学号	学号和课程号共同作为主键
2	kcID	char(6)	课程编号	
3	gradeNum	smallint	成绩	

表 1-11 学生信息表 (Student)

序号	字段名	字段类型	说明	备注
1	sID	char(6)	学生学号, 主键	
2	sName	varchar(20)	学生姓名, 不能为空	
3	sSex	char(2)	学生性别	
4	sBirthday	smalldatetime	出生日期	
5	sZhuanYe	varchar(20)	所学专业	
6	sBanji	char(6)	所属班级	班级编号, 外键约束
7	sRuxueshijian	smalldatetime	入学时间	
8	sSushe	char(6)	所住宿舍	宿舍编号, 外键约束
9	sAddr	varchar(40)	家庭地址	
10	sTel	varchar(20)	联系电话	

表 1-12 费用缴纳信息表 (Pay)

序号	字段名	字段类型	说明	备注
1	payStuID	char(6)	缴费学生学号	
2	payNum	smallmoney	缴纳金额, 不能为空	
3	payLeibie	char(6)	缴费类别	学费、住宿费、其他
4	payRiqi	smalldatetime	缴纳日期	默认系统当前日期
5	payJingbanren	char(6)	经办人	教师工号, 外键约束



表 1-13 宿舍信息表 (Hostel)

序号	字段名	字段类型	说明	备注
1	hosID	char(6)	宿舍编号, 主键	
2	hosTel	char(12)	宿舍电话	
3	hosShezhang	char(6)	舍长	学号, 外键约束
4	hosGuanliyuan	char(6)	宿舍管理员	教师工号, 外键约束

表 1-14 奖惩信息表 (Jiangchen)

序号	字段名	字段类型	说明	备注
1	jcID	int	奖惩编号, 主键	
2	jcRen	char(6)	奖惩人, 不能为空	
3	jcLeibie	char(6)	奖惩类别	奖励、处分、其他
4	jcNeirong	varchar(100)	奖惩内容	
5	jcShijian	smalldatetime	奖惩时间	
6	jcShifoushengxiao	bit	是否生效	
7	jcWenjian	varchar(50)	相关文件	

在物理实现数据库后, 需要装入数据进行调试。装入的数据请参考本书所采用的例库。在这里就不再赘述了。

## 1.7 本章小结

本章以晓灵同学制定“晓灵学生管理系统”开发文档, 为实现“晓灵学生管理系统”奠定基础为线索, 简明扼要地介绍了数据库系统的基本概念和发展历程、数据库系统的结构和组成; 阐述了信息描述与数据模型、数据库规范化设计理论; 重点介绍了关系模型与关系数据库、数据库设计方法; 最后简单地介绍了当前流行的几种关系数据库系统。

通过本章的学习, 学生重点掌握关系模型与关系数据库、数据库的设计方法及相关理论。了解数据库系统的结构与组成和当前流行的关系数据库系统的特点。从实践应用的角度出发, 学生应具备根据应用系统的功能设计数据库的能力。拥有这种能力并不是一蹴而就的, 应该进行大量的设计练习, 并且不断地总结经验才能循序渐进。

## 1.8 课后练习

1. 什么是数据、数据库、数据库技术、数据处理、数据库系统和数据库管理系统?
2. 数据库经历了哪些阶段的发展?
3. 数据模型有几种?
4. 描述概念模型最常用的方法是什么? 实体间的联系又有哪些?
5. 某医院病房计算机管理中需如下信息:  
科室: 科名、科地址、科电话、医生姓名

病房：病房号、床位数、所属科室名

医生：姓名、职称、所属科室名、年龄、工作证号

病人：病历号、姓名、性别、诊断医生、病房号

其中，一个科室有多个病房，多个医生；一个病房只能属于一个科室；一个医生只属于一个科室，但可负责多个病人的诊治；一个病人的主治医生只有一个。设计该系统的 E-R 图。

6. 设计能够表示出房地产交易中客户、业务员和合同三者之间的关系的数据库。

(1) 确定客户实体、业务员实体和合同实体的属性。

(2) 确定客户、业务员和合同三者之间的相互联系，给联系命名并指出联系的类型。

(3) 确定联系本身的属性。

(4) 画出客户、业务员和合同三者关系的 E-R 图。

(5) 将 E-R 图转化为表，写出表的关系模式并表明各自的主键和外键。

7. 一个学生课程数据库，包括学生关系 (Student)、课程关系 (Course) 和选修关系 (SC)，分别求：①机电系的学生②年龄小于 20 的学生③在学生姓名和所在系两个属性上的投影。

学号	姓名	性别	年龄	所在系
001	李勇	男	20	计算机
002	刘晨	女	19	机电
003	王敏	女	18	管理
004	张力	男	19	机电

8. SLC (学号, 系名, 学生住处, 课程号, 成绩) 码为 (学号, 课程号), 每个系的学生住同一地方。判断该模式属于哪一类范式, 并逐步升级到 4NF。

9. 简述关系模型的完整性规则。在参照完整性中, 为什么外键属性的值可以为空? 什么情况下才可以为空?

10. 数据库的设计步骤是什么?

11. 常见的数据库的种类有哪些?

12. Microsoft SQL Server 2000 开发平台有哪些特点?

## 1.9 实验

### 一、实验目的

- (1) 熟悉 E-R 模型的基本概念和图形的表示方法。
- (2) 掌握将现实世界的事物转化为 E-R 图的基本技巧。
- (3) 熟悉关系数据模型的基本概念。
- (4) 熟悉将 E-R 图转化为表。

### 二、实验内容

要开发一个在线考试系统, 该系统能够完成对题库进行管理, 考试新闻发布, 试卷制定、

审核、生成，考场的环境，考试结果公布与查询等具体功能。要求设计出该系统的概念模型、关系模型。

### 三、实验步骤

- (1) 根据需求确定实体、属性和联系。
- (2) 将实体、属性和联系转化为 E-R 图。
- (3) 将 E-R 图转化为基本表。