

## 第3章

# 数据库设计

第1章已经从用户需求、业务需求、功能需求等角度对人事管理系统进行了详细的需求分析,本章将在需求分析的基础上,针对需求描述的内容,按照数据库系统的实现过程和步骤,对系统的数据库进行设计。

本章将详细介绍数据库设计的基础知识,包括数据库设计特点、数据库设计基本策略、数据库设计的步骤、数据库设计的主要方法、数据库设计的基本概念,及数据库设计各个阶段的设计方法。

### 学习目标

- (1) 数据库设计的相关概念。
- (2) 数据库设计的方法和步骤。
- (3) 数据库设计过程。

## 3.1 数据库设计

数据库设计(Database Design)是指对于一个给定的应用环境,构造最优的数据库模式,建立数据库及其应用系统,使之能够有效地存储数据,满足各种用户的应用需求,主要包括信息要求和处理要求。

合理的数据库结构是数据库应用系统性能良好的基础和保证,但数据库的设计和开发却是一项庞大而复杂的工程。

从事数据库设计的人员,不仅要具备数据库知识和数据库设计技术,还要有程序开发的实际经验,掌握软件工程的原理和方法。数据库设计人员必须深入应用环境,了解用户具体的专业业务。在数据库设计的前期和后期,与应用单位人员密切联系,共同开发,可大大提高数据库设计的成功率。

### 3.1.1 数据库设计的特点

数据库设计具有以下几个特点。

(1) 数据库建设是硬件、软件和干件的结合,三分技术,七分管理,十二分基础数据。技术与管理的界面称为“干件”。

(2) 数据库设计应该与应用系统设计相结合。主要包括数据结构设计和行为设计。结构设计是指设计数据库框架或数据库结构。行为设计是指设计应用程序、事务处理等。

(3) 结构和行为分离的设计。传统的软件工程忽视对应用中数据语义的分析和抽象,只要有可能就尽量推迟数据库结构设计的决策,早期的数据库设计致力于数据模型和建模方法研究,忽视了对行为的设计。

### 3.1.2 数据库设计的基本策略

数据库设计的一般策略有两种:自顶向下(Top-Down)和自底向上(Bottom-Up)。自顶向下是从一般到特殊的开发策略。它是从一个企业的高层管理着手,分析企业的目标、对象和策略,构造抽象的高层数据模型。然后逐步构造越来越详细的描述和模型(子系统的模型)。模型不断地扩展细化,直到能识别特定的数据库及其应用为止。自底向上的开发采用与抽象相反的顺序进行。它从各种基本业务和数据处理着手,即从一个企业的各个基层业务子系统的业务处理开始,进行分析和设计。然后将各子系统进行综合和集中,进行上一层系统的分析和设计,将不同的数据进行综合,最后得到整个信息系统的分析和设计。

这两种方法各有优缺点。在实际的数据库设计开发过程中,常常把这两种方法综合起来使用。

### 3.1.3 数据库设计的步骤

在确定了数据库设计的策略以后,就需要设计相应的方法和步骤。多年来,人们提出了多种数据库设计方法,多种设计准则和规范。

数据库是某个企业、组织或部门所涉及的数据的综合,它不仅反映数据本身的内容,而且反映数据之间的联系。数据库是用数据模型来抽象、表示、处理现实世界中的数据和信息的。根据模型应用的不同目的,将数据模型分成两个层次:概念模型和具体的(如关系)数据模型。概念模型是用户和数据库设计人员之间进行交流的工具,数据模型由概念模型转化而来,是按照计算机系统的观点来对数据建模。产生具体数据模型的数据库设计即为逻辑设计。

1978年10月召开的新奥尔良(New Orleans)会议提出的关于数据库设计的步骤,简称新奥尔良法,是目前得到公认的、较完整较权威的数据库设计方法,它把数据库设计分为如下4个主要阶段。

(1) 用户需求分析。

(2) 信息分析和定义(概念设计):本阶段包括视图模型化、视图分析和汇总。

(3) 设计实现(逻辑设计):本阶段包括模式初始设计、子模式设计、应用程序设计、模式评价和模式求精。

(4) 物理设计。

数据库设计一般应包括数据库的结构设计和行为设计两部分内容。所谓数据库的结构设计,是指系统整体逻辑模式与子模式的设计,是对数据的分析设计;数据库的行为设计,是指施加在数据库上的动态操作(应用程序集)的设计,是对应用系统功能的分析设计。

计。虽然,数据库行为设计与一般软件工程的系统设计产生模块化程序的过程是一致的,并且从学科划分的范畴来看,它更偏重于软件设计。但是,在系统分析中,过早地将“数据分析”和“功能分析”进行分离是不明智的,也是不可能的。因为数据需求分析是建立在功能分析上的,只有通过功能分析,才能产生系统数据流程图与数据字典,然后才能通过数据分析去划分实体与属性等,最后才能进入结构设计。

目前,较多的数据库设计专家认为,数据库结构设计的基本步骤应如图 3-1 所示。

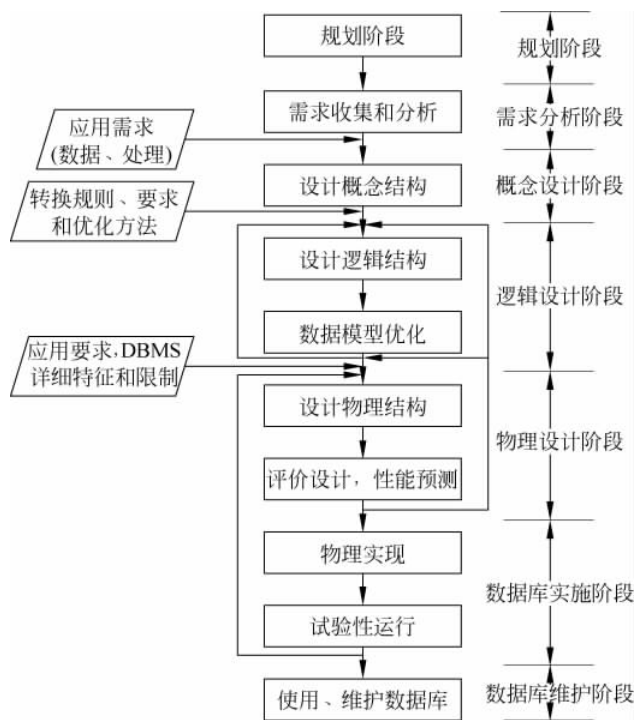


图 3-1 数据库结构设计基本步骤

在任一设计阶段,一旦发现不能满足用户数据需求时,均须返回到前面的适当阶段,进行必要的修正。执行如此的迭代求精过程,直到能满足用户需求为止。在进行数据库结构设计时,应考虑满足数据库中数据处理的要求,将数据和功能两方面的需求分析、设计和实现在各个阶段同时进行,相互参照和补充。

事实上,数据库设计中,每一个阶段设计成果都应该通过评审。评审的目的是确认某一阶段的任务是否全部完成,从而避免出现重大的错误或疏漏,保证设计质量。评审后还需要根据评审意见修改所提交的设计成果,有时甚至要回溯到前面的某一阶段,进行部分重新设计乃至全部重新设计,然后再进行评审,直至达到系统的预期目标为止。

### 1. 规划

数据库设计的规划阶段主要进行建立数据库系统的必要性及可行性分析。规划阶段必须完成以下任务:确定系统的范围、任务,陈述数据库应用程序的主要目标,每个任务标识数据库必须支持的特定任务;确定开发工作所需的资源;估计开发成本;确定项目进度。

系统规划纲要内容主要包括：系统目标与范围的描述，系统运行环境描述，确定计算机系统选型要求，系统开发进度的初步计划。

可行性分析应从经济方面、技术方面、系统运行方面(管理体制、人员的适应性及法律法规)进行分析和评价。

## 2. 需求分析

需求分析是数据库设计的第一阶段，本阶段所得的结果是下一阶段系统的概念结构设计的基础。如果需求分析有误，则以它为基础的整个数据库设计将成为毫无意义的工作。而需求分析也是数据库设计人员感觉最烦琐和困难的一步。

数据库需求分析和一般信息系统的系统分析基本是一致的。但是，数据库需求分析所收集的信息却要详细得多，不仅要收集数据的型(包括数据的名称、数据类型、字节长度等)，还要收集与数据库运行效率、安全性、完整性有关的信息，包括数据使用频率、数据间的联系以及对数据操纵时的保密要求等。

## 3. 概念结构设计

概念结构设计的目标是产生反映系统信息需求的数据库概念结构，即概念模式。概念结构是独立于支持数据库的 DBMS 和使用的硬件环境的。此时，设计人员从用户的角度看待数据以及数据处理的要求和约束，产生一个反映用户观点的概念模式，然后再把概念模式转换为逻辑模式。各级模式之间的关系如图 3-2 所示。

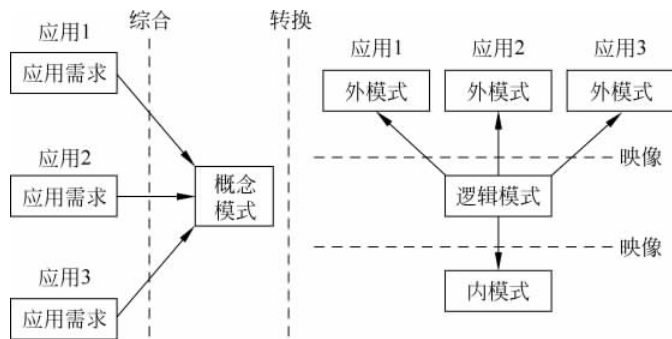


图 3-2 数据库的各级模式之间的关系

描述概念结构的模型应具有以下几个特点。

(1) 有丰富的语义表达能力。能表达用户的各种需求，反映现实世界中各种数据及其复杂的联系，及用户对数据的处理要求等。

(2) 易于交流和理解。概念模型是系统分析师、数据库设计人员和用户之间的主要交流工具。

(3) 易于修改。概念模型能灵活地加以改变，以反映用户需求和环境的变化。

(4) 易于向各种数据模型转换。设计概念模型的最终目的是向某种 DBMS 支持的数据模型转换，建立数据库应用系统。

传统的数据模型(层次、网状、关系模型)，由于缺乏必要的语义表达手段，不适合做概念模型。人们提出了多种概念设计的表达工具，其中最常用、最有名的是 E-R 模型。

在需求分析中已初步得到了有关各类实体、实体间的联系以及描述它们性质的数据

元素,统称数据对象。在这个阶段中,首先要从以上数据对象中确认出:系统有哪些实体,每个实体有哪些属性,哪些实体间存在联系,每一种联系有哪些属性,然后就可以做出系统的局部 E-R 模型和全局 E-R 模型。

从 E-R 模型中可以获得实体、实体间的联系等信息,但不能得到约束实体处理的业务规则。对模型中的每个实体中的数据所进行的添加、修改和删除操作,应该符合预定的规则。特别是删除,往往包含着一些重要的业务规则。

本阶段利用需求分析得到的数据流程图等进行工作,主要的输出文档有系统各子系统的局部概念结构描述、系统全局概念结构描述、修改后的数据字典和概念模型应具有的业务规则。

与本阶段同步,对数据处理的同步分析应产生系统说明书。系统说明书包括新系统的要求、方案和概要设计,反映新系统信息流的数据流程图。

#### 4. 逻辑结构设计

数据库的逻辑设计就是把概念设计得到的数据库模型转化为具体的 DBMS 所能接受的数据库逻辑结构,包括数据库模式和外模式。而目前大多数 DBMS 支持关系数据模型,所以数据库的逻辑设计首先是将 E-R 模型转换为等价的关系模式。

逻辑结构设计阶段主要有以下输入信息。

(1) 概念结构设计阶段的输出信息:所有的局部和全局概念模式。

(2) 处理需求:需求分析阶段产生的业务活动分析结果。包括用户需求、数据的使用频率和数据库的规模。

(3) DBMS 特性:即特定的 DBMS 所支持的数据结构。如 RDBMS 的数据结构是二维表。

本阶段需完成的任务如下所示。

(1) 将 E-R 模型转换为等价的关系模式。

(2) 按需要对关系模式进行规范化。

(3) 对规范化后的模式进行评价。调整关系模式,使其满足性能、存储空间等方面的要求。

(4) 根据局部应用的需要,设计用户外模式。

#### 5. 物理结构设计

数据库物理设计主要指数据库的存储记录格式、存储记录安排和存取方法,包括索引机制、空间大小、块的大小等,物理设计的目的是以合适的存储空间得到用户事务的快速响应(时间和空间的效率),是在计算机的物理设备上确定应采取的数据存储结构和存取方法,以及如何分配存储空间等问题。物理设计与特定硬件系统、DBMS 等相关,难以形成统一的设计方法。

物理设计的过程如下。

(1) 使用 DBMS 的可用功能完成基表的设计和完整性约束和业务规则。

(2) DBMS 为数据提供了许多可选择的文件组织方式,基于对事务的分析,选择合适的文件组织方式,选择合适的索引。

(3) 考虑放宽规范化约束,改善系统性能,怎样处理派生数据,历史数据。考虑重复

列或连接表。

(4) 设计安全措施来避免对数据未经授权的访问,如何实现每个用户的视图,以及表上需要的访问控制。

(5) 不断通过监听操作系统来标识和解决由设计引起的性能问题,并实现新的或改变的需求。

## 6. 数据库运行和维护

数据库应用程序开发即应用程序的设计、编码、调试和试运行过程,其中应用程序设计主要包括事务设计和用户界面设计,是数据库应用程序开发的重要环节。

(1) 事务设计。事务代表了现实世界的事件,事务设计包括事务使用什么数据、事务要做什么、事务的输出、事务的使用频度。事务有检索事务、更新事务、混合事务之分。

(2) 用户界面设计。用户界面设计要易于掌握、操作直观。界面设计可考虑有意义的标题、可视化空间布局与边界、常用字段标签、不可接受值的错误信息提示、便利的游标活动、一致性的术语和缩略语、一致的颜色和字体、一致的操作快捷方式(尽可能地使用非鼠标化操作)、一致的鼠标操作方式、应用程序编写、组织数据入库、应用程序的调试与试运行、整理文档。

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,运行维护阶段包括以下任务。

(1) 维护数据库的安全性和完整性:检查系统安全性是否受到侵犯,及时调整授权和密码,实施系统转储与后备,发生故障及时恢复。

(2) 监测并改善数据库性能: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。

(3) 必要时对数据库进行重新组织和重新构造。

(4) 根据用户要求对数据库现有功能进行扩充。

### 3.1.4 数据库设计的主要方法

从20世纪70年代末以来,众多学者对数据库设计方法进行了深入的探讨和尝试,给出了许多各有优缺点的数据库设计方法,有基于E-R模型的数据库设计方法,基于3NF的设计方法,基于抽象语法规则的设计方法等,较实用的主流方法有以下两种。

#### 1. E-R模型加规范化关系的方法

在数据库结构设计中,主要工作是从需求分析所得到的所有信息以及它们之间的依赖关系出发去构造系统数据模型。在构建模型中,最常用的是E-R模型法。E-R模型中最基本的成分是实体、联系以及它们的属性。而由实体(或联系)与属性构成的关系,因为是否“规范化”而有“好”、“坏”之分,而关系的好坏又直接影响数据库的质量。所以,把E-R模型与规范化关系结合起来,形成了目前最为流行的数据库建模方法。本章介绍概念结构设计时,采用了此方法。

#### 2. 数据元素图加规范化关系的方法

需求分析中得到的每一种信息,称为数据元素。数据元素图指系统所涉及的数据元素信息(包括:数据元素名、类型、字节长度、可能的别名、有效取值范围等)以及数据元素

之间的依赖关系。用数据元素图及数据元素之间的关联类型的分析来归并实体,并且产生部分实体之间的联系,这种方法称为数据元素图法(也称为数据项图法)。

把数据元素图法与规范化关系有机结合起来,也是一种较实用的建模方法。采用这种方法,不必再经过 E-R 模型转换成规范化的关系,而可以直接根据作图法获得规范化的关系,有时候也被称为“扩充数据元素图法”。

### 3.1.5 数据库设计的基本概念

数据库设计中将会用到很多基本概念和术语,下面分别加以介绍。

(1) 规划:在数据库设计的规划阶段主要进行建立数据库系统的必要性及可行性分析。

(2) 需求分析:数据库设计人员采用一定的辅助工具对应用对象的功能、性能、限制性要求所进行的科学分析。

(3) 概念设计:对应用对象精确地抽象、概括而形成的独立于计算机系统的企业信息模型。描述概念模型的最好工具是 E-R 图。

(4) 逻辑设计:将抽象的概念模型转化为与选用的 DBMS 产品所支持的数据模型相符合的逻辑模型,它是物理设计的基础。

(5) 物理设计:逻辑模型在计算机中的具体实现方案。

(6) 数据字典(Data Dictionary,DD):关于数据的信息集合。它包含了应用对象和 DBMS 运行时所需的控制和管理信息的信息库。它既可为用户服务,又可为系统服务。数据字典在需求分析阶段建立,在数据库设计过程中被不断修改、充实和完善。

(7) 数据流程图(Data Flow Diagram,DFD):也称为数据流图,是便于用户理解的系统数据流的图形表示,能精确地在逻辑上描述系统的功能、输入、输出和数据存储。该工具用于需求分析。

## 3.2 实体—联系模型

E-R 图(Entity-Relationship Diagram,实体—联系图)提供了表示实体类型、属性和联系的方法,用来描述现实世界的概念模型。它是表示概念模型的一种方式,用矩形表示实体型,矩形框内写明实体名;用椭圆表示实体的属性,并用无向边将其与相应的实体型连接起来;用菱形表示实体型之间的联系,在菱形框内写明联系名,并用无向边分别与有关实体型连接起来。

### 1. E-R 模型的要素

构成 E-R 图的基本要素是实体型、属性和联系,其表示方法如下。

(1) 实体型(Entity):具有相同属性的实体具有相同的特征和性质,用实体名及其属性名集合来抽象和刻画同类实体;在 E-R 图中用矩形表示,矩形框内写明实体名。例如学生张三、学生李四都是实体,如果是弱实体的话,在矩形外面再套实线矩形。

(2) 属性(Attribute):是实体所具有的某一特性,一个实体可由若干个属性来刻画。在 E-R 图中用椭圆形表示,并用无向边将其与相应的实体连接起来。例如学生的姓名、

学号、性别,这些都是属性。如果是多值属性,则在椭圆形外面再套实线椭圆。如果是派生属性则用虚线椭圆表示。

(3) 联系(Relationship): 联系也称关系,用来反映信息世界中实体内部或实体之间的联系。实体内部的联系通常是指组成实体的各属性之间的联系; 实体之间的联系通常是指不同实体集之间的联系。联系在 E-R 图中用菱形表示,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)。例如老师给学生授课存在授课关系,学生选课存在选课关系。

如果是弱实体的联系则在菱形外面再套菱形。

联系可分为以下 3 种类型。

① 一对一联系(1:1)。

例如,一个部门有一个经理,而每个经理只在一个部门任职,则部门与经理的联系是一对一的。

② 一对多联系(1:N)。

例如,某校教师与课程之间存在一对多的联系“教”,即每位教师可以教多门课程,但是每门课程只能由一位教师来教。

③ 多对多联系(M:N)。

例如,学生与课程间的联系是多对多的,即一个学生可以学多门课程,而每门课程可以有多个学生来学。联系也可能有属性。例如,学生“学”某门课程所取得的成绩,既不是学生的属性也不是课程的属性。由于“成绩”既依赖于某名特定的学生又依赖于某门特定的课程,所以它是学生与课程之间的联系“学”的属性。

## 2. 使用 E-R 图设计数据库

(1) 构造 E-R 模型。

E-R 图作为传统且流行的数据库设计方法,具有接近人的思想、容易理解、与计算机无关等优点。在构造 E-R 模型时,可以采用以下步骤来完成。

① 确定实体。

② 除去重复实体。

③ 列出每个实体的属性。

④ 标记主键。

⑤ 定义联系。

⑥ 检查每个实体类型,以查看它如何与其他实体相连。

⑦ 描述联系的类型(1:1, 1:N, M:N)。

⑧ 除去冗余关系。

建立 E-R 模型是一个反复的过程,所以将会画出几个版本,并对每一个版本进行改进,直到有一个理想的终端出现。这里没有单一的正确答案,只能找出一个相对最优的选择。

(2) 属性的定义规则。

在定义属性时,应符合以下规则。

① 作为“属性”,不能再有需要描述的性质。属性必须是不可再分的数据项,不能含

有其他的属性。

② 属性不能与其他实体有联系。

### 3. 概念模型向关系模型的转换

(1) 实体集的转换规则。

概念模型中的一个实体集转换为关系模型中的一个关系,实体的属性就是关系的属性,实体的键就是关系的键,关系的结构是关系模式。

(2) 实体集间联系的转换规则。

① 1:1 联系的转换方法。

将 1:1 联系转换为一个独立的关系,与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性,且每个实体的键均是该关系的候选键。

将 1:1 联系与某一端实体集所对应的关系合并,则需要和被合并关系中增加属性,其新增的属性为联系本身的属性和与联系相关的另一个实体集的键。

② 1:N 联系的转换方法。

一种方法是将联系转换为一个独立的关系,其关系的属性由与该联系相连的各实体集的键以及联系本身的属性组成,而该关系的键为 N 端实体集的键。

另一种方法是在 N 端实体集中增加新属性,新属性由联系对应的 1 端实体集的键和联系自身的属性构成,新增属性后原关系的键不变。

③ M:N 联系的转换方法。

将 M:N 联系转换为一个关系,转换方法为:与该联系相连的各实体集的键以及联系本身的属性均转换为关系的属性,新关系的键为两个相连实体键的组合(该键为多属性构成的组合键)。

④ 3 个或 3 个以上实体集间的多元联系的转换方法。

对于一对多的多元联系,转换为关系模型的方法是修改 1 端实体集对应的关系,即将与联系相关的其他实体集的键和联系自身的属性作为新属性加入到 1 端实体集中。

对于多对多的多元联系,转换为关系模型的方法是新建一个独立的关系,该关系的属性为多元联系相连的各实体的键以及联系本身的属性,键为各实体键的组合。

## 3.3 关系规范化

规范化是关系数据库逻辑设计的另一种方法,它和 E-R 模型的出发点不一样,但是一个基于规范化的关系设计和一个由 E-R 模型转换成的关系设计几乎可以得到相同的结果。两种方法具有互补性。规范化方法中,从一个将被建模的现实世界的情形出发,分析数据项之间的相互影响,通过无损分解使数据库的逻辑设计趋于合理,把低一级的关系模式分解为若干个高一级的关系模式。

### 1. 关系模式的设计问题

关系模型要求关系必须是规范化的,即要求数据库表中不允许含有多值属性和内部结构,关系的每一个分量必须是一个不可分的数据项,不允许表中还有表。遵守这样规则的表称为第一范式。这是关系数据库设计过程中的一个最基本的约束。满足第一范式还

不能保证数据库模式是良好的数据库模式。所谓良好的数据库模式的标准是什么,如何实现,这就是关系模式的设计问题。

存在大量数据冗余的数据库模式不是良好的数据库模式。因为大量的数据冗余不仅造成存储空间的浪费,存取效率的低下,而且使得数据信息的更新变得烦琐复杂,另外还隐含着破坏数据一致性、完整性的危险。良好的数据库模式不存在数据冗余、插入异常、删除异常和更新异常等问题。重新分配数据项到不同的表中能够消除这些问题,这恰是规范化过程将实现的任务。

## 2. 异常情况说明

(1) 更新异常(Update Anomaly): 如果更改表所对应的某个实体实例或者关系实例的单个属性时,需要将多行更新,那么就说这个表存在更新异常。数据冗余会引起更新异常。

(2) 删除异常>Delete Anomaly): 如果删除表的某一行来反应某个实体实例或者关系实例消失时,会导致另一个不同实体实例或者关系实例的信息丢失,而这是用户不希望发生的,那么就说这个表存在删除异常。

(3) 插入异常(Insert Anomaly): 如果某个实体或者实例信息随着另一个实体或实例信息的存在而存在,在缺少另一个实体或实例信息时,无法表示这个实体或者实例信息,而这是用户不希望看到的,那么就说这个表存在插入异常。

## 3.4 范式和关系模式规范化

范式(Normal Forms, NF)是规范化过程中的一系列逻辑步骤。范式的类型有第一范式(1NF)、第二范式(2NF)、第三范式(3NF)、Boyce Codd 范式(BCNF)。经常称某一关系模式为第几范式,表明该关系模式是符合某一种范式级别的关系模式。

一个低一级范式的关系模式,通过模式分解可以转换为若干个高一级范式的关系模式的集合,这种过程就叫关系模式规范化。

### 1. 第一范式(1NF)

如果关系模式  $R$  的每个关系  $r$  的属性都是不可分的数据项,那么就称  $R$  是第一范式的模式。1NF 是关系模式应具备的最起码的条件。关系数据库设计研究的关系规范化是在 1NF 之上进行的。只符合 1NF 标准的关系模式不是好的数据库模式。

### 2. 第二范式(2NF)

如果关系模式  $R$  是 1NF,且每个非主属性完全函数依赖于候选键,那么就称  $R$  是第二范式。

提示: 1NF 模式通过分解可以达到 2NF,方法是消除非主属性对候选键的部分依赖,把部分依赖的属性单独组建关系。

### 3. 第三范式(3NF)

关系模式  $R$  中不存在这样的候选键  $X$ , 属性组  $Y$  和非主属性  $Z(Z \not\subset Y)$  使得  $X \rightarrow Y$ ,  $(Y! \rightarrow X)Y \rightarrow Z$  成立,即非主属性既不部分依赖也不传递依赖于  $R$  的候选键,则称  $R$  是第三范式。