

# 数据库范式应用实例剖析

## The Analysis of Application Case of Normal Form

涂云杰 TU Yun-jie

(呼伦贝尔学院计算机科学与技术学院, 呼伦贝尔 021008)

(Computer Science and Technology Department, Hulunbeier College, Hulunbeier 021008, China)

**摘要:** 范式是符合某一种级别的关系模式的集合。最基本的数据库范式有三个, 第一范式(1NF), 第二范式(2NF)和第三范式(3NF)。文章介绍了范式相关概念、1范式和2范式存在的问题和模式分解的方法。文章首先给出一个达到1范式的关系模式, 分析里面存在的函数依赖, 并分析了该模式存在的诸多问题。然后将此1范式分解为若干2范式。进而将2范式转换成若干3范式。通过模式分解在一定程度上解决了1NF存在的数据冗余大和插入异常、删除异常及其修改异常等问题。

**Abstract:** Normal form is the set of relational mode suitable for a kind of level. Database to be built must abide by a certain rule. The most basic normal forms includes three kinds, namely 1NF, 2NF and 3NF. The related concepts of normal form, the problems of 1NF and 2NF and the ways of mode decomposition were introduced in this paper. In the first place, an example of 1NF was given. The functional dependency in the 1NF and the problems in it were listed. Then the 1NF was decomposed into several 2NF. Furthermore, a 2NF was decomposed into several 3NF. By means of mode decomposition, such problems as a lot of data redundancy, insertion anomalies, deletion anomalies and modification anomalies are dealt with to a certain extent.

**关键词:** 范式; 规范化; 函数依赖; 模式分解

**Key words:** normal form; normalization; functional dependency; mode decomposition

中图分类号: TP392

文献标识码: A

文章编号: 1006-4311(2012)26-0213-02

### 0 引言

范式是符合某一种级别的关系模式的集合。构造数据库必须遵循一定的规则。在关系数据库中, 这种规则就是范式。

目前关系数据库有六种范式: 第一范式(1NF)、第二范式(2NF)、第三范式(3NF)、第四范式(4NF)、第五范式(5NF)和第六范式(6NF)。满足最低要求的范式是第一范式(1NF)。在第一范式的基础上进一步满足更多要求的称为第二范式(2NF), 其余范式以次类推。最基本的数据库范式有三个, 第一范式(1NF), 第二范式(2NF)和第三范式(3NF)。并不是规范化程度越高的关系就越优, 一般说来, 第三范式就足够了。<sup>[1]</sup>并不是任何一个关系模式都是合理的, 那么如何才能设计出合理的关系模式呢?

### 1 范式有关概念

**1.1 第一范式(1NF)** 如果一个关系模式  $R$  的所有属性都是不可再分的数据项, 则称  $R \in 1NF$ 。<sup>[2,4]</sup>即实体中的某个属性不能有多值或者不能有重复的属性。这个单一属性由基本类型构成, 包括整型、实数、字符型、逻辑型、日期型等。简而言之, 第一范式就是无重复的列。

在任何一个关系数据库中, 第一范式(1NF)是对关系模式的基本要求, 不满足第一范式(1NF)的数据库就不是关系数据库。很显然, 在当前的任何关系数据库管理系统(DBMS)中, 傻瓜也不可能做出不符合第一范式的数据库, 因为这些DBMS不允许你把数据库表的一列再分成二列或多列。因此, 你想在现有的DBMS中设计出符合第一范式的数据库都是不可能的。

**1.2 第二范式(2NF)** 如果关系模式  $R \in 1NF$ , 且每一个非主属性完全函数依赖于码, 则  $R \in 2NF$ 。<sup>[2]</sup>即在2NF中不存在非主属性对码的部分函数依赖。

第二范式(2NF)是在第一范式(1NF)的基础上建立起来的, 即满足第二范式(2NF)必须先满足第一范式(1NF)。第二范式(2NF)要求数据库表中的每个实例或行必须可以被唯一地区分。为实现区分通常需要为表加上一个列, 以存储各个实例的惟一标识。这个惟一属性列被称为主关键字或主键、主码。

第二范式(2NF)要求实体的属性完全依赖于主关键字。所谓完全依赖是指不能存在仅依赖主关键字一部分的属性, 如果存在, 那么这个属性和主关键字的这一部分应该分离出来形成一个新的实体, 新实体与原实体之间是一对多的关系。为实现区分通常需要为表加上一个列, 以存储各个实例的惟一标识。

**1.3 第三范式(3NF)** 如果关系模式  $R(U) \in 1NF$ , 且  $R(U)$  中每一个非主属性都不传递函数依赖于  $R$  的候选码, 则  $R(U) \in 3NF$ 。<sup>[5]</sup>

数据库的设计范式是数据库设计所需要满足的规范, 满足这些规范的数据库是简洁的、结构明晰的, 同时, 不会发生插入(insert)、删除(delete)和更新(update)操作异常。反之则是乱七八糟, 不仅给数据库的编程人员制造麻烦, 而且面目可憎, 可能存储了大量不需要的冗余信息。

### 2 范式分解实例

下面我们举例介绍第一范式(1NF)、第二范式(2NF)和第三范式(3NF)。

**2.1 1NF 关系模式存在的问题** 下面我们以模式 SCG(Sno, Sname, Sage, Sdep, DT, Cno, Cname, Peno, Grade)为例来说明该模式存在的问题。(Sno 代表学号, Sname 代表学

生姓名, Sage 代表年龄, Sdep 代表系别, DT 代表系办电话, Cno 代表课号, Cname 代表课程名称, Peno 代表先行课, Grade 代表学生选修的课程成绩。)这个关系的某些语义如下:

每个学生属于一个且仅属于一个系;

每个学生选修的每门课程有且仅有一个成绩;

一个系只有一部系办电话。

模式 SCG 候选码为组合关键字 (Sno, Cno), 存在如下决定关系:

$(Sno, Cno) \rightarrow (Sname, Sage, Sdep, DT, Cname, Peno, Grade)$

$Sno \rightarrow Sname, Sage, Sdep$

$Cno \rightarrow Cname, Peno$

$Sdep \rightarrow DT$

这个关系不满足第二范式, 因为存在非主属性 Sname, Sage, Sdep, Cname, Peno 对码 (Sno, Cno) 的部分函数依赖。

表 1 是其一个实例。

表 1 模式 SCG

Sno	Sname	Sage	Sdep	DT	Cno	Cname	Peno	Grade
0001	何为	19	数学	8259176	C01	离散数学	C03	78
0001	何为	19	数学	8259176	C02	线性代数	C03	88
0001	何为	19	数学	8259176	C03	高等数学		76
0002	马立	18	计算机	8259175	C02	线性代数	C03	67
0002	马立	18	计算机	8259175	C03	高等数学		88
0003	李立	18	计算机	8259175	C04	操作系统	C11	76

根据定义判定该模式不符合 2NF, 这个模式存在如下问题:

**2.1.1 冗余度大** 同一门课程由  $n$  个学生选修, Cname 和 Peno 就重复  $n-1$  次; 同一个学生选修了  $m$  门课程, 姓名 Sname、年龄 Sage 和系别 Sdep 就重复了  $m-1$  次。一方面浪费存储, 另一方面系统要付出很大代价维护数据库完整性。

**2.1.2 操作异常** 由于数据的冗余, 在对数据操作时会引起各种异常:

①插入异常。假设要开设一门新的课程, 暂时还没有人选修。这样, 由于主属性 Sno 为空, 违反了关系的实体完整性, 该条记录无法插入到关系中。所以无法把课程的信息存入数据库。

②删除异常。在删除某学生选课记录的同时, 会将该学生及其课程的基础信息在系统中一起删除。

③修改异常。如某个学生年龄修改后, 必须注意修改有关的每一个元组。否则会出现同一学生年龄不同的情况, 造成数据的不一致。

一个关系模式之所以会产生这些问题, 是由于模式中存在的某些不好的数据依赖引起的。解决方法是通过分解关系模式来消除其中不合适的数据依赖, 以解决插入异常、删除异常、更新异常和数据冗余。<sup>[2]</sup>

**2.2 1NF 到 2NF 的分解** 按照定义判断, SCG 最高达到了第一范式, 而没有达到第二范式, 因此要进行分解。分解方法如下:

①把关系模式中对键完全函数依赖的非主属性与决定它们的键放在一个关系模式中。

②把对键部分函数依赖的非主属性和决定它们的主属性放在一个关系模式中。

③检查分解后的新模式, 如果仍不是 2NF, 则继续按照前面的方法进行分解, 直到达到要求。

概括来讲就是部分依赖归子集, 完全依赖随键。<sup>[3]</sup>

依据以上方法可以将 SCG 分解为如下关系:

$SD(Sno, Sname, Sage, Sdep, DT)$

$C(Cno, Cname, Peno)$

$SC(Sno, Cno, Grade)$

分解后的关系模式 SD、C 和 SC 中不再存在非主属性对码的部分函数依赖, 所以  $SD \in 2NF, C \in 2NF, SC \in 2NF$ 。

采用投影分解法将一个 1NF 的关系分解为多个 2NF 的关系, 可以在一定程度上减轻原 1NF 关系中存在的插入异常、删除异常、数据冗余度大、修改异常等问题。<sup>[2]</sup>

以上分解出的各个关系中, SD 中存在非主属性 DT 对码 Sno 的传递函数依赖, 所以 SD 最高达到了第二范式, 而没有达到第三范式。其他两个关系 C 和 SC 都达到了第三范式。在 SD 中还存在数据冗余问题。

**2.3 2NF 到 3NF 的分解** 一个范式如果满足第二范式但不满足第三范式, 可以通过模式分解将其分解为若干个模式, 使得分解后的模式能够满足第三范式。分解方法如下:

①把关系模式中对键直接函数依赖的非主属性与决定它们的键放在一个关系模式中。

②把造成传递函数依赖的决定因素连同被它们决定的属性放在一个关系模式中。

③检查分解后的新模式, 如果仍不是 3NF, 则继续按照前面的方法进行分解, 直到达到要求。

概括来讲就是基本依赖为基础, 中间属性作桥梁。<sup>[3]</sup>

依据以上方法将 SD 分解为如下关系:

$S(Sno, Sname, Sage, Sdep)$

$D(Sdep, DT)$

分解后的关系模式 S 与 D 中不再存在非主属性对码的传递函数依赖, 所以  $S \in 3NF, D \in 3NF$ 。

基于投影分解法将最初的一个模式 SCG 分解为 S, D, C, SC 这四个关系模式, 成功解决了 SCG 中存在的冗余、插入异常、删除异常和修改异常问题。

**参考文献:**

[1]李冬芸. SQL Server 数据库应用技术实例教程[M]. 北京: 中国水利水电出版社, 2010.

[2]萨师煊, 王珊. 数据库系统概论(第四版)[M]. 北京: 高等教育出版社, 2006.

[3]马玉华. 关系数据库模式分解方法分析[J]. 漯河职业技术学院学报, 2007, (4).

[4]褚洪波. 关系数据库设计范式应用实例剖析[J]. 鸡西大学学报, 2008, (1).

[5]崔巍. 数据库系统及应用(第二版)[M]. 北京: 高等教育出版社, 2006.