第 5 章　关系模式的规范化设计

前面为一个学生选课数据库构造了几个关系模式，每个关系模式又由若干属性组成，以此介绍了关系数据库的基本概念、关系模型的组成要素，以及关系数据库的标准语言SQL。但是对一个应用而言，不同设计者可能会设计出不同的数据库模式，那么什么是“好”的数据库模式？如何评价一个“好”的数据库模式？又如何设计一个“好”的数据库模式？这正是本章所要解决的问题。

实际上，设计任何一种数据库应用系统，不论是层次的、网状的还是关系的，都存在着如何构造合适的数据库模式即数据库逻辑结构的问题。由于关系模型有着严格的数学理论基础，并且可以向别的数据模型转换，因此一般以关系数据模型为背景来讨论这个问题。本章介绍关系数据库的规范化设计理论，也是关系数据库逻辑结构设计的理论工具。

5.1　关系模式的设计问题

首先了解什么样的关系模式是“好”的关系模式，或者说不好的关系模式会存在哪些问题。下面举例说明。

【应用实例 5-1】　某学校需要建立一个信息管理系统，要求该信息管理系统能管理各系学生课程学习的相关信息，具体包括学生信息、所在系信息、学生选修课程的信息等。

假设设计单一的关系模式：R（学生学号，学生姓名，所在系，系主任，课程名称，成绩）来存储所有信息。表 5-1 给出了某一时刻关系模式 R 的实例。

表 5-1　关系模式 R 的一个实例

学生学号	学生姓名	所在系	系主任	课程名称	成绩
2020307001	陈昊	计算机系	王林	操作系统	93
2020307001	陈昊	计算机系	王林	计算机网络	87
2020307001	陈昊	计算机系	王林	编译原理	85
2020307002	崔政	计算机系	王林	操作系统	95
2020307002	崔政	计算机系	王林	计算机网络	84
2020308001	冯月	自动化系	赵磊	计算机网络	83
2020308002	黄聪	自动化系	赵磊	软件工程	89

由于一个学生要选修多门课程，而且多名学生会选修相同的课程，这样该关系模式就存在冗余存储的问题，学生姓名和所在系信息、所在系系主任的信息均被重复存储多次，而且该关系模式的候选键只能为（学生学号，课程名称）。

除了存在数据冗余问题，对该关系进行更新操作还会出现下列问题。

（1）插入异常。假设某系新来一名学生，但是该学生还没有选修课程，往数据库中插入新来的学生信息时，由于在主属性“课程名称”上的值为空值，不满足实体完整性约束规

则，所以该学生的信息无法插入关系中。出现了应该插入的信息无法插入的情况。

（2）删除异常。关系 R 中学号为“2020308001”的学生只选修了一门课程，现要将该学生的选课信息删除，若用 UPDATE 语句将属性“课程名称”和“成绩”值置为空值，因“课程名称”是主属性，该操作不能实现，而若用 DELETE 语句将整个元组删除，则该学生的个人信息就消失了。出现了不应该删除的信息被删除的情况。

（3）数据不一致。假设某学生调换了专业系，则需用 UPDATE 语句将该学生的“所在系”属性值进行修改，操作完成后，会发现在关系中存在学生“所在系”的属性值相同，而对应的“系主任”的属性值不同，出现了数据的不一致问题。

由此可见，该关系模式不仅存在数据冗余浪费存储空间问题，还可能出现更新异常和数据不一致问题。

正是由于该关系模式存在上述问题，所以该关系模式不是一个“好”的关系模式。一个“好”的关系模式应当不会出现更新异常和数据不一致问题，数据冗余应尽可能少。

虽然插入和删除操作出现异常是系统完整性约束的限制导致的，但产生问题的根源在于关系模式中的属性间存在数据依赖关系，而属性间的一些不好的数据依赖导致产生上述问题。那么，关系模式的属性间存在哪些数据依赖？怎样消除关系模式属性间不好的数据依赖，从而消除数据冗余以及可能出现的更新异常和数据不一致问题？对于这些问题的讨论形成了关系模式规范化设计理论。

5.2 关系模式的规范化

首先回顾关系模式的形式化定义。一个完整的关系模式应当是一个五元组，形式化地表示为

$$R(U,D,\mathrm{Dom},F)$$

其中，R 为关系名；U 为组成该关系的属性集合；D 为属性集 U 中属性所来自的域的集合；Dom 为属性向域的映射的集合；F 为属性间的一组数据依赖集。

由于 D 和 Dom 可隐含在 U 中，在讨论关系模式的规范化时一般把关系模式简化为用三元组来表示：

$$R(U,F)$$

有时若单独提出数据依赖集 F，$R(U,F)$ 还可简化为 $R(U)$。

数据依赖（data dependency）是关系模式的重要要素，是一个关系的属性与属性之间的一种约束关系，反映的是现实世界事物特征间的一种依赖关系，是数据内在的性质，是语义的体现。

E.F.Codd 在提出关系模型后，基于关系模式属性间的数据依赖，对关系模式的设计问题进行了研究，提出了指导关系模式设计的规范化理论。

属性间的数据依赖有多种类型，下面主要讨论函数依赖（functional dependency，FD），简单介绍广义的函数依赖——多值依赖（multivalued dependency，MVD），而对广义的多值依赖——连接依赖（join dependency，JD）等不进行介绍。

5.2.1 函数依赖

函数依赖是指在一个关系模式中,如果一组属性的取值可以决定另一组属性的取值,就说这两组属性之间存在着函数依赖。

【例 5-1】 在关系模式 R(学生学号,学生姓名,所在系,系主任,课程名称,成绩)中,由于一个学生只属于一个专业系,因此当"学生学号"的值确定了,"所在系"的值也就被唯一确定了,所以"所在系"属性的值依赖于"学生学号"属性的值,类似数学中的函数关系,就称"学生学号"属性和"所在系"属性之间存在着函数依赖。由于一个学生可选修多门课程,因此当元组中"学生学号"的值确定了,并不能确定其对应的"课程名称",所以"学生学号"属性和"课程名称"属性之间不存在着函数依赖。

下面给出函数依赖的形式化定义。

定义 5.1 设有关系模式 R,其属性集为 U,X、Y 是 U 的子集,即 $X\subseteq U$、$Y\subseteq U$。对于 R 的任意可能的关系实例 $r(R)$及其中任意两个元组 $t_1\in r$、$t_2\in r$,若 $t_1[X]=t_2[X]$,则 $t_1[Y]=t_2[Y]$,则称 Y 函数依赖于 X,或 X 函数决定 Y,记为 $X\rightarrow Y$。X 称为决定子,也称决定因素。若 $X\rightarrow Y$,$Y\rightarrow X$,则记作 $X\leftrightarrow Y$。若 Y 不函数依赖于 X,则记作 $X\not\rightarrow Y$。

从定义可知,对于 R 的所有可能的合法值,若两个元组在 X 属性集上的值相等,则两个元组在 Y 属性集上的值也相等,即不可能存在两个元组在属性集 X 上的属性值相等,而在属性集 Y 上的属性值不等。可见,函数依赖是指给定关系模式中一个属性集和另一个属性集间的依赖约束关系,它要求按此关系模式建立的任何关系 $r(R)$都应满足函数依赖集 F 中的约束条件。

在理解函数依赖概念时应注意以下两点。

(1) 函数依赖是关系模式的所有关系实例都应该满足的特性。若在 $R(U)$的某个关系实例 $r(R)$中,存在两个元组在 X 属性上的值相等,而在 Y 属性上的值不等,就可确定 $X\not\rightarrow Y$。

(2) 属性间的函数依赖是数据内在的特性,是数据语义的体现,只能根据数据的语义来确定一个函数依赖成立与否。而不应只是针对某个特定的关系实例,根据函数依赖的定义推断出来。

【例 5-2】 找出关系模式 R(学生学号,学生姓名,所在系,系主任,课程名称,成绩)中存在的函数依赖。学校的实际情况(语义)告诉我们:

(1) 每个学生有唯一的一个学号,只能编制在一个专业系。

(2) 一个专业系有若干学生,只有一名系主任。

(3) 一个学生可以选修多门课程,每门课程可以有多名学生选修。

(4) 每名学生选修每一门课程会有一个成绩。

根据这些语义,可以得到属性集上的一组函数依赖:

$$F=\{\text{学生学号}\rightarrow\text{学生姓名},\text{学生学号}\rightarrow\text{所在系},$$
$$\text{所在系}\rightarrow\text{系主任},(\text{学生学号},\text{课程名称})\rightarrow\text{成绩}\}$$

可用图 5-1 所示的函数依赖图来描述关系模式 R 上的这一组函数依赖。

关系模式上的函数依赖除了来自现实的语义,也可以由数据库设计者对现实世界进

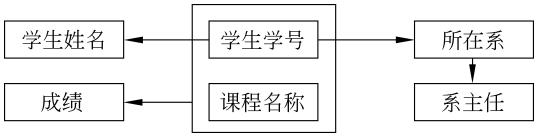


图 5-1 关系模式 R 的函数依赖图

行强制的约束。例如，在关系模式 R（学生学号，学生姓名，所在系，系主任，课程名称，成绩）中，规定“学生姓名→所在系”，则此时要求不允许有同名的学生存在。一旦函数依赖被告知 DBMS，DBMS 就将其作为关系模式的完整性约束条件加以实现，进入数据库中的所有数据都必须严格遵守。

在关系模式 R（学生学号，学生姓名，所在系，系主任，课程名称，成绩）上的一组函数依赖 $F=\{$学生学号→学生姓名，学生学号→所在系，所在系→系主任，（学生学号，课程名称）→成绩$\}$中，哪些函数依赖会引起更新异常和数据不一致问题呢？下面首先介绍函数依赖的相关概念。

定义 5.2 在 $R(U)$ 中，$X\subseteq U$、$Y\subseteq U$，若 $Y\subseteq X$，显然 $X\to Y$ 成立，称为平凡函数依赖，否则称为非平凡函数依赖。

【例 5-3】 在关系模式 R（学生学号，学生姓名，所在系，系主任，课程名称，成绩）中，“（学生学号，学生姓名）→学生学号”是平凡的函数依赖。

对于任一关系模式，平凡函数依赖都是必然成立的，并没有实际的意义，它不反映新的语义，平凡函数依赖一般并不在关系模式的函数依赖集中显式列出。在讨论关系模式规范化时，主要考虑非平凡的函数依赖。

定义 5.3 设 X、Y 是关系 R 的属性集，且 $X\neq Y$，若 $X\to Y$，且不存在 $X'\subset X$，使 $X'\to Y$，则称 Y 完全函数依赖于 X，记为 $X\xrightarrow{f}Y$；否则称 Y 部分函数依赖于 X，记为 $X\xrightarrow{p}Y$。

完全函数依赖 $X\xrightarrow{f}Y$ 是指 Y 不函数依赖于 X 的任何子属性（集），而部分函数依赖 $X\xrightarrow{p}Y$ 则指 Y 函数依赖于 X 的部分属性（集）。当 X 是单属性时，$X\to Y$ 必为完全函数依赖。

【例 5-4】 在关系模式 R（学生学号，学生姓名，所在系，系主任，课程名称，成绩）中，“学生学号→所在系”“所在系→系主任”“（学生学号，课程名称）→成绩”都是完全函数依赖，“（学生学号，课程名称）→所在系”是部分函数依赖，因为“学生学号→所在系”成立。而“学生学号→成绩”不成立，当然也不是部分函数依赖。

定义 5.4 在 $R(U)$ 中，$X\subseteq U$、$Z\subseteq U$，若存在 $Y\subseteq U$，$Y\not\subseteq X$，$Z\not\subseteq Y$，使得 $X\to Y$，$Y\to Z$，且 $Y\not\to X$，则称 Z 传递函数依赖于 X，可记为 $X\xrightarrow{t}Z$。

注意：如果 $Y\to X$，则 $X\leftrightarrow Y$，Z 对 X 是直接函数依赖而不是传递函数依赖。

【例 5-5】 在关系模式 R（学生学号，学生姓名，所在系，系主任，课程名称，成绩）中，由于“学生学号→所在系”“所在系→系主任”，所以“学生学号→系主任”是传递函数依赖。

候选键是关系模式中一个重要的概念，可以用函数依赖的概念定义候选键。

定义 5.5 在关系模式 $R(U)$ 中，若 $K\subseteq U$，且 $K\xrightarrow{f}U$，则 K 为关系 R 的候选键。

【例 5-6】 在关系模式 R(学生学号,学生姓名,所在系,系主任,课程名称,成绩)中,(学生学号,课程名称)$\xrightarrow{f}$(学生学号,学生姓名,所在系,系主任,课程名称,成绩),因而(学生学号,课程名称)是候选键。

但 U 中属性对候选键的函数依赖程度却是不同的,除了构成候选键的主属性"学生学号"和"课程名称"外,其他非主属性对候选键的依赖情况是:

(学生学号,课程名称)$\xrightarrow{f}$成绩

(学生学号,课程名称)$\xrightarrow{p}$学生姓名

(学生学号,课程名称)$\xrightarrow{p}$所在系

(学生学号,课程名称)$\xrightarrow{t}$系主任

各属性对候选键的不同的函数依赖与该关系模式存在的数据冗余和更新异常等问题之间有什么关系呢?

属性组(学生学号,课程名称)作为候选键,用来唯一标识每一个元组,受实体完整性约束限制,这两个属性的值均不能为空,而学生的信息,如姓名、所在系、系主任等只依赖于学生学号,这就使得学生的信息、系的信息冗余存储;且受限于主属性"课程名称"的非空约束,对学生信息的插入和删除操作,以及系的信息插入操作发生异常。

"系主任"属性传递依赖于候选键,而完全依赖于"所在系"属性,当对学生的所在系或系主任进行修改时,由于没有考虑"所在系"属性和"系主任"属性间的依赖关系,导致出现更新带来的数据不一致问题。

因此,关系模式属性间的部分依赖和传递依赖才是导致关系模式存在数据冗余、更新异常及数据不一致问题的根源。为解决这个问题,Codd 提出了范式的概念。

5.2.2 基于函数依赖的范式

范式根据关系模式中属性间的数据依赖来判定关系模式的优劣。

数据依赖满足一定约束(性质)的关系模式称为范式(normal form,NF)。根据约束的程度,范式又有多个级别。满足最低要求的称为第一范式,简称 1NF,在第一范式的基础上满足进一步要求的为第二范式(2NF),其余依次类推。范式主要有 1NF、2NF、3NF、BCNF、4NF、5NF 等。

1971—1972 年,Codd 系统地提出了 1NF、2NF、3NF 的概念。1974 年 Codd 和 Boyce 又共同提出了一个新范式,即 BCNF(也称巴斯范式)。1976 年 Fagin 提出了 4NF,后来又有人提出了 5NF。

"第几范式"表示关系模式的某一种级别,可称某一关系模式 R 为第几范式。也可将"范式"理解成符合某一级别的关系模式的集合,则 R 为第几范式就可以写成 $R \in x$NF。

1. 第一范式(1NF)

在关系数据模型中,一般要求所有的域都是原子数据的集合,这种限制称为第一范式条件。

定义 5.6 对于关系模式 R,当且仅当 R 中的每个属性对应的域是原子的(atomic),

则该关系模式 R 属于第一范式，即 $R \in 1NF$。

第一范式是关系模式需要满足的最低要求，要求属性对应的域中元素是不可分的单元。因此，关系的属性只能是简单属性，不能是复合属性，也不能是多值属性。

在目前的 RDBMS 中，如果关系模式不属于第一范式，则无法创建关系模式对应的基本关系表。

【例 5-7】 对于关系 R（学生学号，学生姓名，所在系，系主任，课程名称，成绩），$R \in 1NF$。如果将 R 中“所在系”属性划分成“所在大学、所在学院、所在系”三个组成部分，或将学生的多个成绩值放在同一元组的同一属性上，则该关系模式 $R \notin 1NF$。

由第 5.1 节中对关系 R 的操作分析，可知满足 1NF 的关系模式 R 存在数据冗余、更新异常和数据不一致等问题，因此仅满足 1NF 的关系模式还不是一个好的关系模式，需要对它进行规范化（normalization）处理。

2. 第二范式（2NF）

定义 5.7 对于关系模式 R，当且仅当 $R \in 1NF$，且 R 中每一个非主属性都完全函数依赖于候选键时，该关系模式 R 属于第二范式，即 $R \in 2NF$。

【例 5-8】 关系 R（学生学号，学生姓名，所在系，系主任，课程名称，成绩）中候选键是（学生学号，课程名称），由于该关系模式中存在函数依赖：学生学号→学生姓名，学生学号→所在系，因此（学生学号，课程名称）$\xrightarrow{P}$学生姓名，（学生学号，课程名称）$\xrightarrow{P}$所在系，即存在非主属性“学生姓名”和“所在系”对于候选键的部分函数依赖，所以该关系模式 $R \notin 2NF$。

但若将关系 R（学生学号，学生姓名，所在系，系主任，课程名称，成绩）中完全依赖候选键的成绩属性和候选键构成一个关系模式 R_1（学生学号，课程名称，成绩），而将只依赖“学生学号”的学生的信息，如“学生姓名”和“所在系”等属性，单独构成一个关系模式 R_2（学生学号，学生姓名，所在系，系主任），R_2 的候选键就是学生学号，此时“学生姓名”属性和“所在系”属性均完全函数依赖于候选键“学生学号”，“系主任”属性仍传递依赖于候选键“学生学号”。这样在两个新的关系模式 R_1 和 R_2 中均不存在着非主属性对于候选键的部分依赖，均属于第二范式。

由表 5-1 中关系 R 分解得到的关系 R_1、R_2 的实例值如表 5-2 所示，关系模式 R_1、R_2 的函数依赖图如图 5-2 所示。

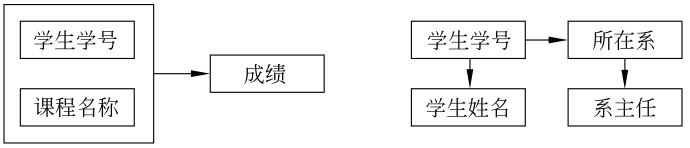
表 5-2 由关系 R 分解得到的关系 R_1 和 R_2 的实例值

（a）R_1 的实例值

学生学号	课程名称	成绩
2020307001	操作系统	93
2020307001	计算机网络	87
2020307001	编译原理	85
2020307002	操作系统	95
2020307002	计算机网络	84
2020308001	计算机网络	83
2020308002	软件工程	89

（b）R_2 的实例值

学生学号	学生姓名	所在系	系主任
2020307001	陈昊	计算机系	王林
2020307002	崔政	计算机系	王林
2020308001	冯月	自动化系	赵磊
2020308002	黄聪	自动化系	赵磊



(a) R_1的函数依赖　　(b) R_2的函数依赖

图 5-2　关系模式 R_1、R_2 的函数依赖图

相对于关系 R，关系 R_1 和 R_2 中数据冗余减少了，并消除关系 R 中由部分函数依赖产生的更新异常问题。例如，新来的学生即使还没有选修课程，也可往关系 R_2 中插入新来的学生信息；在关系 R_1 中，将某学生选修课程的信息删除，不会同时把关系 R_1 中该学生的信息删除。但是在关系 R_2 中仍然存在系主任信息的冗余以及由此带来的数据不一致问题，因此还需要进一步规范化。

3. 第三范式(3NF)

定义 5.8　对于关系模式 R，当且仅当 $R\in 2NF$，且 R 中所有非主属性都不传递函数依赖于候选键时，该关系模式 R 属于第三范式，记为 $R\in 3NF$。

因此，属于 3NF 的关系模式的每一个非主属性既不部分函数依赖于候选键，也不传递函数依赖于候选键。

【例 5-9】　对于由关系 R(学生学号，学生姓名，所在系，系主任，课程名称，成绩)分解得到的满足 2NF 的关系模式 R_1(学生学号，课程名称，成绩)和 R_2(学生学号，学生姓名，所在系，系主任)，根据 3NF 的定义，可知 R_1(学生学号，课程名称，成绩)$\in$3NF，以及 R_2(学生学号，学生姓名，所在系，系主任)中由于存在传递函数依赖“学生学号$\xrightarrow{t}$系主任”，因此 $R_2\notin 3NF$。

可在 R_2(学生学号，学生姓名，所在系，系主任)中，将候选键“学生学号”和完全函数依赖于候选键的“学生姓名”和“所在系”属性单独构成一个关系模式 R_3(学生学号，学生姓名，所在系)，R_3 的候选键仍是“学生学号”；而将“所在系”属性和完全函数依赖于“所在系”属性的“系主任”属性单独构成一个关系模式 R_4(所在系，系主任)，“所在系”属性为候选键。这样在两个新的关系模式 R_3 和 R_4 中均不存在着非主属性对于候选键的传递函数依赖，当然也不存在着非主属性对于候选键的部分函数依赖，均属于第三范式。

由表 5-2 中关系 R_2 分解得到的关系 R_3 和 R_4 的实例值如表 5-3 所示，关系模式 R_3 和 R_4 的函数依赖图如图 5-3 所示。

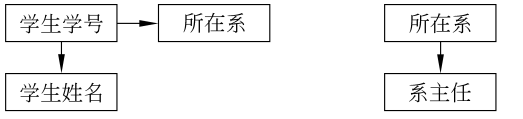
表 5-3　由关系 R_2 分解得到的关系 R_3 和 R_4 的实例值

(a) R_3 的实例值

学生学号	学生姓名	所在系
2020307001	陈昊	计算机系
2020307002	崔政	计算机系
2020308001	冯月	自动化系
2020308002	黄聪	自动化系

(b) R_4 的实例值

所在系	系主任
计算机系	王林
自动化系	赵磊



(a) R_3的函数依赖　　(b) R_4的函数依赖

图 5-3　关系模式 R_3 和 R_4 的函数依赖图

在关系模式 R_3 和 R_4 中，消除了关系模式 R_2 中的传递函数依赖，系主任信息的冗余，以及插入系的信息操作受限等更新异常问题消失，修改学生所在系或系主任的操作不会再产生数据不一致问题。

至此，原有关系模式 R（学生学号，学生姓名，所在系，系主任，课程名称，成绩）中存在的数据冗余、更新异常及数据不一致问题都得到了解决。

原有关系模式 R 之所以存在问题是因为该关系模式中描述的信息太多了，既描述了学生的基本信息，又描述了学生所在系信息，还描述了学生选修课程的信息，这些信息放在同一个关系模式中进行描述，相互依赖，导致了异常的发生。把一个满足低一级范式的关系模式（如 R、R_2）分解为若干满足高一级范式的关系模式（如 R_1、R_2、R_3、R_4）的过程，也就是把这些信息逐渐从原有的关系模式中分离出来的过程，使得每个关系模式描述的信息比较单一，消除关系模式属性间的相互约束。

第三范式是一个可用的关系模式应该满足的最低范式要求。也就是说，如果一个关系模式不满足 3NF 的话，通常是不可用的。那么，满足 3NF 的关系模式是否就不存在问题了呢？请看下面的应用实例。

【应用实例 5-2】　在关系模式 STC(Sno，Tno，Cno，G)中，有学号 Sno、教师编号 Tno、课程编号 Cno、选课成绩 G 这 4 个属性。假设每名学生可以选修多门课程，每门课程可被多名学生选修；每名教师只能讲授一门课程，每门课程可以由多名教师来讲授；某名学生选定了某一门课就确定了一名固定的教师，并有一个选课成绩。

根据数据间的这些语义，可以得到该关系模式的如下函数依赖集，函数依赖图如图 5-4 所示。

$$F=\{\text{Tno}\rightarrow\text{Cno},(\text{Sno},\text{Cno})\rightarrow\text{Tno},(\text{Sno},\ \text{Cno})\rightarrow G,(\text{Sno},\ \text{Tno})\rightarrow G\ \}$$

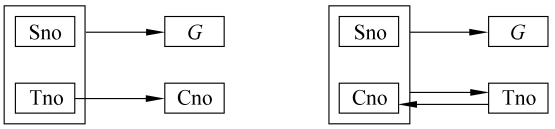


图 5-4　关系模式 STC 的函数依赖图

由关系模式 STC 上的函数依赖集 F 可推断该关系模式的候选键为(Sno，Cno)或(Sno，Tno)，这样在关系模式 STC 中，属性 Sno、Cno 和 Tno 均为主属性，属性 G 为非主属性，非主属性 G 完全函数依赖于两个候选键，不存在非主属性对候选键的部分函数依赖和传递函数依赖，所以 STC∈3NF。表 5-4 给出关系模式 STC 的一个实例。

表 5-4　关系模式 STC 的一个实例

Sno	Tno	Cno	*G*
S01	T01	C01	80
S02	T01	C01	78
S03	T01	C01	90
S04	T02	C01	80
S05	T02	C01	88
S01	T03	C02	98
S02	T04	C02	85
S03	T04	C02	80
S04	T04	C02	69

从表 5-4 给出的关系实例可看到该关系模式仍存在着数据冗余,虽然一个教师只教一门课,但每个选修该教师所授课程的学生选课元组都记录着这一信息。除了数据冗余存储外,还会出现如下问题。

(1) 插入异常:如果要将一名编号为 T10 的教师讲授课程编号为 C02 的课程的信息插入关系 STC 中,但目前还没有学生选修这门课程,则该教师开设这门课程的信息无法插入关系 STC 中。

(2) 删除异常:关系 STC 中教师编号为 T03 的老师所讲授的 C02 课程只有一个学号为 S01 的学生选修,若要将该学生的选课信息删除,如果用 DELETE 语句将整个元组删除,则该教师的讲课信息就丢失了;而如果用 UPDATE 语句将属性 Sno 和 *G* 值置空值,因 Sno 是主属性,操作也不能实现。

(3) 数据不一致:如果要将学号为 S01 的学生选修的课程编号为 C01 的课程改为课程编号为 C02 的课程,就会出现编号同为 T01 的教师讲授的课程不同,而一个老师只能讲授一门课程,出现了数据不一致问题。

产生上述问题的原因是由于该关系模式 STC 存在着函数依赖 Tno→Cno,即存在主属性 Cno 对候选键(Sno,Tno)的部分函数依赖。

因此,一个满足 3NF 的关系模式仍不是一个理想的关系模式,为此引入 BC 范式。

4. BC 范式(BCNF)

定义 5.9　对于关系模式 R,若 $X\rightarrow Y$,$Y\subsetneq X$ 时,X 必含有候选键,即 R 中的所有非平凡的、完全的函数依赖的决定因素是候选键,则 $R\in$BCNF。

根据 BCNF 的定义,可以得到以下结论,若 $R\in$BCNF,则有

(1) R 中所有非主属性都完全函数依赖于候选键。

(2) R 中所有主属性对每一个不包含它的候选键也是完全函数依赖。

(3) R 中没有任何属性完全函数依赖于非候选键的任何一组属性。

【例 5-10】　关系模式 STC(Sno,Tno,Cno,*G*)中,由于存在 Tno→Cno,该函数依赖的决定因素没有包含候选键(Sno,Cno)或(Sno,Tno),因而关系模式 STC∉BCNF。

可将决定因素不是候选键的函数依赖所涉及的属性单独构成一个关系模式,即将 STC 分解为 TC(Tno,Cno)和 ST(Sno,Tno, *G*)两个关系模式,TC 中 Tno→Cno,ST 中

(Sno,Tno)→G,则分解后的两个关系模式均为 BCNF。

由表 5-4 中关系 STC 分解得到的关系 TC 和 ST 的实例值如表 5-5 所示,关系中不再有教师编号和课程编号的数据冗余,以及更新操作异常等问题。例如,可在关系 TC 中插入编号为 T10 的教师讲授课程编号为 C02 的课程信息;可在关系 ST 上删除学号为 S01 的学生选修的编号为 T03 的教师所授课程的选课信息,该编号为 T03 的授课教师的讲课信息仍在关系 TC 中存储;若要将学号为 S01 的学生选修的课程由 C01 改为 C02,可通过在关系 ST 上将学号为 S01 的学生的授课教师的编号由 T01 改为某个讲授课程编号为 C02 课程的教师编号即可。

表 5-5 由关系 STC 分解得到的关系 TC 和 ST 的实例值

(a) ST(Sno,Tno,G)

Sno	Tno	G
S01	T01	80
S02	T01	78
S03	T01	90
S04	T02	80
S05	T02	88
S01	T03	98
S02	T04	85
S03	T04	80
S04	T04	69

TC(Tn0,Cn0)

Tno	Cno
T01	C01
T02	C01
T03	C02
T04	C02

以上在函数依赖的范畴内讨论 1NF、2NF、3NF 和 BCNF,其中 1NF 是关系模式隐含的,2NF 和 3NF 是讨论 BCNF 的基础,本身意义并不大,重要的和广泛应用的是 BCNF,它能满足一般应用的数据处理要求。如果一个关系数据库中的所有关系模式都属于 BCNF,那么在函数依赖范围内,它已实现了数据库中数据的彻底分解,达到了最高的规范化程度。但是,如果考虑其他的数据依赖(如多值依赖),它仍然不是最好的关系模式。

5.2.3 多值依赖与 4NF

1. 多值依赖

【应用实例 5-3】 学校中某一门课程可以由多个教师讲授,他们使用相同的一套参考书,每个教师可以讲授多门课程,每种参考书可以供多门课程使用,可用表 5-6 来直观描述课程名称、教师姓名和参考书名称之间的对应关系。假设用关系模式 CTB(C,T,B)来描述课程、教师和参考书之间的关系,表 5-7 给出了关式模式 CTB 的一个实例。

根据关系模式 CTB(C,T,B)的应用语义以及参考给出的关系的实例,可知关系模式 CTB(C,T,B)的候选键是(C,T,B),是一个全键,则不存在着非主属性及非平凡的函数依赖,所以 CTB∈BCNF。但从给出的关系实例可见,该关系模式 CTB 仍存在数据冗余度大的问题。例如,有多少名任课教师,参考书就要存储多少次,即对于一门给定的课程,所有的教师和参考书的可能组合情况都可能出现。仍然会导致如下问题的出现。