

第5章

关系数据库设计理论

前面的章节已经介绍了数据库中涉及的基本概念,关系模型的3个组成部分(关系数据结构、关系操作集合和关系完整性)以及关系数据库的标准语言,但是还有一个很基本的问题没有提及,就是针对一个具体的问题应该构造几个关系模式,每个关系由哪些属性组成,各属性之间的依赖关系及其对关系模式性能的影响等。关系数据库的规范化理论为用户设计出合理的数据库提供了有利的工具。

本章将介绍如何有效地消除关系中存在的冗余数据和更新异常等现象,从而设计出优秀的关系数据库模式,即主要讨论关系数据库规范化的理论,这是数据库逻辑设计的理论依据。

5.1 问题的提出

数据库的逻辑设计为什么要遵循一定的规范化理论?什么是好的关系模式?某些“不好”的关系模式会导致哪些问题?对于这些问题可以通过一个现实生活中普遍存在的例子加以分析。

【例 5.1】 假设有一个用于电力设备存放管理的数据库,其关系模式如下:

WAE(仓库号,所在区域,区域主管,设备号,数量)

并且有语义如下:

- 一个区域有多个仓库,一个仓库只能属于一个区域;
- 一个区域只有一个区域主管;
- 一个仓库可以存放多种设备,每种设备可以存放在多个仓库中;
- 每个仓库的每种设备都有一个库存数量。

在此关系模式中填入一部分具体的数据,则可得到 WAE 关系模式的实例,如图 5.1 所示。

这个关系模式存在以下几个方面的问题。

- 数据冗余:每个区域主管的姓名重复出现,这将浪费大量的数据空间。
- 更新异常:如果某区域主管更换,则该主管对应的所有记录都要逐一修改,稍有不慎,就有可能漏改某些记录,这就造成了数据的不一致性,破坏了数据的完整性。
- 插入异常:如果某个仓库刚刚建成,尚未有设备存入,则仓库的信息无法插入到数据库中。因为在这个关系模式中,(仓库号,设备号)是主键。根据关系的实体完整性约束,主键的值不能为空,而这时没有设备,设备号为空,因此不能完成插入操作。

仓库号	所在区域	区域主管	设备号	数量
WH1	A 区	赵龙	P1	100
WH1	A 区	赵龙	P2	150
WH1	A 区	赵龙	P3	120
WH2	B 区	张立	P2	200
WH2	B 区	张立	P4	500
WH2	B 区	张立	P5	300
WH3	A 区	赵龙	P1	200
WH3	A 区	赵龙	P4	300

图 5.1 关系 WAE

- 删除异常：如果某个仓库的设备全部出库，仓库已空，要删除该仓库全部设备的记录，这时仓库号、所在区域、区域主管的信息也将被删掉，事实上这个仓库依然存在，但在数据库中却无法找到该仓库的信息。

鉴于以上存在的种种问题，可以得出结论：上述 WAE 关系模式不是一个好的关系模式。

假如把这个单一的模式改造一下，分解为 3 个关系模式：

- W(仓库号,所在区域);
- A(区域,区域主管);
- WE(仓库号,设备号,数量)。

这时图 5.1 可以分解为如图 5.2 所示的 3 个关系模式，这 3 个模式都不会发生插入异常、删除异常的问题，数据的冗余也得了控制。

W		WE		
仓库号	所在区域	仓库号	设备号	数量
WH1	A 区	WH1	P1	100
WH2	B 区	WH1	P2	150
WH3	C 区	WH1	P3	120
		WH2	P2	200
		WH2	P4	500
		WH2	P5	300
		WH3	P1	200
		WH3	P4	300

A	
区域	区域主管
A 区	赵龙
B 区	张立

图 5.2 把关系 WAE 分解成 W、A 和 WE 共 3 个关系

一个“不好”的模式会有哪些不好的性质？如何改造一个“不好”的模式？这是下面几节将要讨论的问题。

5.2 基本概念

为了使数据库设计的方法走向完备，人们研究了规范化理论。从 1971 年起，E. F. Codd 就提出了这一理论，规范化理论的研究已经取得了很多成果。这一理论主要致力于解决关系模式中不合适的数据依赖问题，而函数依赖与多值依赖是最重要的数据依赖。

5.2.1 函数依赖

1. 函数依赖

对于数学中形如 $Y=f(X)$ 的函数大家十分熟悉,它代表 X 和 Y 数值上的一个对应关系,即给定一个 X 值,都有一个 Y 值和它对应, X 函数决定 Y 或 Y 函数依赖于 X 。

在关系数据库中同样存在函数依赖的概念,例如,在学生关系——学生(学号,姓名,性别,出生年月)中,给定一个同学的学号,一定能找到唯一一个与之对应的同学姓名,姓名= f (学号),也一定能找到唯一一个对应的性别和出生年月。这里,学号是自变量 X ;姓名、性别、出生年月是因变量 Y ,并且把 Y 函数依赖于 X 或 X 函数决定 Y 表示为 $X \rightarrow Y$ 。

再举一下电力抢修工程数据库的例子,在配电抢修物资领料出库表 Out_stock(prj_num, mat_num, amount, get_date, department) 中,只有确定了工程项目编号 prj_num 和物资编号 mat_num,才能知道领取物资的数量 amount、领料日期 get_date 及领料部门 department,即 prj_num 和 mat_num 函数决定 amount、get_date 和 department,可以表示为 $(prj_num, mat_num) \rightarrow amount, (prj_num, mat_num) \rightarrow get_date, (prj_num, mat_num) \rightarrow department$ 。

下面对函数依赖给出严格的形式化定义。

定义 5.1 设 $R(U)$ 是属性集 U 上的一个关系模式, X, Y 是 U 的子集。对于 $R(U)$ 上的任何一个可能的关系 r ,如果 r 中不存在两个元组,它们在 X 上的属性值相同,而在 Y 上的属性值不同,则称“ X 函数决定 Y ”或“ Y 函数依赖 X ”,记作 $X \rightarrow Y$ 。

对于函数依赖,需要说明以下几点:

- 函数依赖不是指关系模式 R 的某个或某些关系实例满足的约束条件,而是指 R 的所有关系实例均要满足的约束条件。
- 用户只能根据语义来确定一个函数依赖,不能按照其形式化定义来证明一个函数依赖是否成立。例如,对于上述关系模式 WAE,在区域主管不存在重名的情况下,可以得到:

区域主管 $\rightarrow$ 所在区域

这种函数依赖成立的前提条件是区域主管无重名,否则就不存在函数依赖了,所以函数依赖反映了一种语义完整性约束。

- 函数依赖存在的时间无关性:由于函数依赖是指关系中的所有元组应该满足的约束条件,而不是指关系中某个或某些元组所满足的约束条件。关系中元组的增加、删除或更新都不能破坏这种函数依赖。因此,用户必须根据语义来确定属性之间的函数依赖,而不能单凭某一时刻关系中的实际数据值来判断。例如,对于上述关系模式 WAE,根据语义,只能存在函数依赖所在区域 $\rightarrow$ 区域主管,而不应该存在区域主管 $\rightarrow$ 所在区域,因为如果新增加一个重名的区域主管,这个函数依赖必然不存在。
- 若 $X \rightarrow Y$,则称 X 为这个函数依赖的决定因素(determinant)。
- 若 $X \rightarrow Y$,并且 $Y \rightarrow X$,则记为 $X \leftrightarrow Y$ 。
- 若 Y 不函数依赖于 X ,则记为 $X \nrightarrow Y$ 。

2. 平凡函数依赖与非平凡函数依赖

定义 5.2 设 $R(U)$ 是属性集 U 上的一个关系模式, X, Y 是 U 的子集。如果 $X \rightarrow Y$, 并且 Y 不包含于 X , 则称 $X \rightarrow Y$ 是**非平凡函数依赖**。若 $Y \subseteq X$, 则称 $X \rightarrow Y$ 是**平凡函数依赖**。

很显然, 对于任一关系模式, 平凡函数依赖都是必然存在的, 它不反映新的语义, 因此若不特别声明, 我们总是讨论非平凡函数依赖。

3. 完全函数依赖与部分函数依赖

定义 5.3 在 $R(U)$ 中, 如果 $X \rightarrow Y$, 并且对于 X 的任何一个真子集 X' 都有 $X' \not\rightarrow Y$, 则称 Y 对 X **完全函数依赖**, 记作 $X \xrightarrow{f} Y$ 。

如果 $X \rightarrow Y$, 但不完全函数依赖于 X , 则称 Y 对 X **部分函数依赖**, 记作 $X \xrightarrow{p} Y$ 。

4. 传递函数依赖

定义 5.4 在 $R(U)$ 中, 如果 $X \rightarrow Y, Y \rightarrow Z$, 且 Y 不包含于 $X, Y \not\rightarrow X$, 则称 Z **传递函数依赖于 X** , 记作 $X \xrightarrow{t} Z$ 。

在这里加上条件 $Y \not\rightarrow X$, 是因为如果 $Y \rightarrow X$, 则 $X \leftrightarrow Y$, 实际上就是 $X \xrightarrow{\text{直接}} Y$, 是直接函数依赖, 而不是传递函数依赖。

例如, 在上述关系 WAE(仓库号, 所在区域, 区域主管, 设备号, 数量) 中, 存在非平凡函数依赖 仓库号 $\rightarrow$ 所在区域, 所在区域 $\rightarrow$ 区域主管, (仓库号, 设备号) $\rightarrow$ 数量。其中有完全函数依赖 (仓库号, 设备号) $\xrightarrow{f}$ 数量, 另外, 根据 仓库号 $\rightarrow$ 所在区域, 所在区域 $\rightarrow$ 区域主管, 可以得到传递函数依赖 仓库号 $\xrightarrow{t}$ 区域主管。

5.2.2 码

前面章节给出了关系模式的码的非形式化定义, 这里使用函数依赖的概念来严格定义关系模式的码。

定义 5.5 设 K 为 $R \langle U, F \rangle$ 中的属性或属性组合。若 $K \xrightarrow{f} U$, 则 K 为 R 的**候选码**(Candidate Key)。若候选码多于一个, 则选定其中的一个作为**主键**(Primary Key)。

包含在任何一个候选码中的属性称为**主属性**(Prime Attribute), 不包含在任何码中的属性称为**非主属性**(Nonprime Attribute)或**非码属性**(Non-key Attribute)。最简单的情况是, 单个属性是码。最极端的情况是, 整个属性组是码, 称为**全码**(All-key)。

例如, 在上述关系 WAE 中, (仓库号, 设备号) $\xrightarrow{f} U$, 所以 (仓库号, 设备号) 是该关系的码。

定义 5.6 关系模式 R 中属性或属性组 X 并非 R 的码, 但 X 是另一个关系模式 S 的码, 则称 X 为 R 的**外键**(Foreign Key)。

5.3 规范化

规范化的理论是 E. F. Codd 于 1971 年提出的,目的是要设计“好的”关系数据库模式,其基本思想是消除关系模式中的数据冗余,消除数据依赖中不合适的部分,以解决数据插入、删除时发生的异常现象,这就要求在关系数据库中设计出来的关系模式要满足一定的条件。

通常把关系数据库的规范化过程中为不同程度的规范化要求设立的不同标准称为范式(Normal Form)。根据关系模式满足的不同性质和规范化的程度,把关系模式分为第一范式(1NF)、第二范式(2NF)、第三范式(3NF)、BC 范式(BCNF)、第四范式(4NF),直到第五范式(5NF)。各种范式存在着以下联系:

$$1NF \supset 2NF \supset 3NF \supset BCNF \supset 4NF \supset 5NF$$

通常情况下,把某一模式 R 的第 n 范式简记为 $R \in nNF$ 。一个低一级别的关系模式通过模式分解可以转化为若干个高一级别范式的关系模式的集合,这种过程称为规范化。

5.3.1 第一范式

定义 5.7 如果关系模式 R 的所有属性均为简单属性,即每个属性都是不可再分的,则称 R 属于**第一范式**,简称 1NF,记作 $R \in 1NF$ 。

目前,世界上几乎所有商用关系数据库管理系统都规定关系的属性是原子性的,也就是要求关系都为第一范式。因此,数据库语言的语法决定了关系必须是第一范式。不满足第一范式的数据库模式不能称为关系数据库。

然而,一个关系模式仅仅满足属于第一范式是不够的。前面探讨的关系模式 WAE 属于第一范式,但它具有大量的数据冗余和插入异常、删除异常、更新异常等弊端。为什么会存在这种问题呢?我们来分析一下 WAE 中的函数依赖关系,它的码是(仓库号,设备号)这一属性集,所以有:

$$\begin{aligned} &(\text{仓库号}, \text{设备号}) \xrightarrow{f} \text{数量} \\ &\text{仓库号} \rightarrow \text{所在区域}, (\text{仓库号}, \text{设备号}) \xrightarrow{p} \text{所在区域} \\ &\text{仓库号} \xrightarrow{t} \text{区域主管}, (\text{仓库号}, \text{设备号}) \xrightarrow{t, p} \text{区域主管} \end{aligned}$$

由此可见,在关系 WAE 中既存在非主属性对码的完全函数依赖,又存在非主属性对码的部分函数依赖和传递函数依赖。正是由于关系中存在着复杂的函数依赖,才导致数据操作中出现了种种弊端,出现了例 5.1 中提到的 4 个问题。因此有必要用投影运算将关系分解,去掉过于复杂的函数依赖,向高一级的范式转化。

5.3.2 第二范式

定义 5.8 如果关系模式 $R \in 1NF$,且每个非主属性都完全函数依赖于 R 的码,则称 R 属于**第二范式**,简称 2NF,记作 $R \in 2NF$ 。

在上述关系 WAE 中,仓库号、设备号为主属性;所在区域、区域主管和数量为非主属

性。经过分析,我们发现存在着非主属性对码的部分函数依赖,故 $WAE \notin 2NF$ 。

为了消除部分函数依赖,采用投影分解法,把 WAE 分解为两个关系模式:

WE (仓库号,设备号,数量)

WA (仓库号,所在区域,区域主管)

其中, WE 的码为(仓库号,设备号),函数依赖为:

$(\text{仓库号}, \text{设备号}) \xrightarrow{f} \text{数量}$

WA 的码为仓库号,非主属性为所在区域和区域主管,函数依赖为:

$\text{仓库号} \rightarrow \text{所在区域}, \text{所在区域} \rightarrow \text{区域主管}, \text{仓库号} \xrightarrow{t} \text{区域主管}$

显然,在分解后的关系模式中,非主属性都完全函数依赖于码。例 5.1 中提到的 4 个问题在一定程度上得到了解决:

- 如果某个仓库刚刚建成,尚未有设备存入,该仓库的记录可以插入到 WA 中。
- 如果某个仓库被清空,仍不会影响该仓库信息在 WA 中的记录。
- 由于仓库的存储情况与仓库的基本情况分开存储在两个关系中,因此无论该仓库中存储多少种设备,仓库信息在 WA 中都只存储一次,这就大大减少了数据冗余。

但同时我们看到, WA 关系中也存在着一定的异常:

- 若某个区域刚刚设立还没有仓库,则所在区域和区域主管的值无法插入,造成插入异常。
- 有一定的数据冗余,当多个仓库处于同一个区域时,区域主管的值被多次存储。
- 若某区域要更换区域主管,要逐一修改该区域的所有区域主管记录,稍有不慎,就有可能漏改某些记录,造成更新异常。

因此, WA 仍不是一个好的关系模式。

5.3.3 第三范式

定义 5.9 如果关系模式 $R \in 2NF$,且每个非主属性都不传递函数依赖于 R 的候选码,则称 R 属于**第三范式**,简称 $3NF$,记作 $R \in 3NF$ 。

对于关系模式 WA (仓库号,所在区域,区域主管),存在的函数依赖为 $\text{仓库号} \rightarrow \text{所在区域}, \text{所在区域} \rightarrow \text{区域主管}, \text{仓库号} \xrightarrow{t} \text{区域主管}$,主码为仓库号,主属性为仓库号,非主属性为所在区域及区域主管。由于存在着非主属性区域主管对码为仓库号的传递函数依赖,故 $WA \notin 3NF$ 。

同样,采用投影分解法把 WA 分解为两个关系模式:

W (仓库号,所在区域)

A (所在区域,区域主管)

在 W 关系中,存在函数依赖 $\text{仓库号} \rightarrow \text{所在区域}$,码为仓库号,在 A 关系中存在函数依赖 $\text{所在区域} \rightarrow \text{区域主管}$,码为所在区域。这两个关系模式均满足 $3NF$,原关系中的某些数据冗余也不存在了。

WAE 规范到 $3NF$ 后,所存在的异常现象已经全部消失。但是, $3NF$ 只规定了非主属

性对码的依赖关系,没有限制主属性对码的依赖关系。如果发生了这种依赖,仍有可能存在数据冗余、插入异常、删除异常和更新异常的情况。

为了消除主属性对码的依赖关系,1974年,Boyce和Codd共同提出了一个新范式的定义,这就是 Boyce-Codd 范式,通常简称 BCNF 或 BC 范式。

5.3.4 BC 范式

定义 5.10 如果关系模式 $R \in 1NF$,且对于所有的函数依赖 $X \rightarrow Y (Y \notin X)$,决定因素 X 都包含了 R 的一个候选码,则称 R 属于 BC 范式(Boyce-Codd Normal Form),简称 BCNF,记作 $R \in BCNF$ 。

作为一个满足 BCNF 的关系模式,它具有如下 3 个性质:

- 所有非主属性都完全函数依赖于每个候选码。
- 所有主属性都完全函数依赖于每个不包含它的候选码。
- 没有任何属性完全函数依赖于非码的任何一组属性。

由上面的定义可知 BC 范式既检查非主属性,又检查主属性,显然比第三范式的限制更要严格。当我们只检查非主属性而不检查主属性时,就成了第三范式。因此可以说任何满足 BC 范式的关系都必然满足第三范式。

在上述关系 W 和 A 中都只有一个主键,以作为唯一的候选码,且都只有一个函数依赖,为完全函数依赖,符合 BC 范式的条件,所以 W 和 A 都满足 BC 范式。

【例 5.2】 关系模式 SPJ(学号,课程号,名次),假设每一个学生选修每门课程的成绩都有一个名次,每门课程中每一个名次只有一个学生(即没有并列名次)。由语义可以得到下面的函数依赖:

$(\text{学号, 课程号}) \rightarrow \text{名次}, (\text{名次, 课程号}) \rightarrow \text{学号}$

所以, (学号, 课程号) 与 (名次, 课程号) 都可以作为候选码。在这个关系模式中显然没有非主属性,因此也不存在非主属性对码的部分或传递函数依赖,所以 $SPJ \in 3NF$ 。而且除了 (学号, 课程号) 与 (名次, 课程号) 外没有其他的决定因素,所以 $SPJ \in BCNF$ 。

再举一个属于 3NF 但不属于 BCNF 的例子。

【例 5.3】 假设有关系模式电力设备管理 WES(仓库号,设备号,职工号),它所包含的语义如下:

- (1) 一个仓库可以有多个职工。
- (2) 一名职工仅在一个仓库工作。
- (3) 在每个仓库,一种设备仅由一名职工保管,但每个职工可以保管多种设备。

根据以上语义有函数依赖 $\text{职工号} \rightarrow \text{仓库号}, (\text{仓库号, 设备号}) \rightarrow \text{职工号}$,该关系模式的码是 (仓库号, 设备号),根据范式的定义,该模式是 3NF,因为不存在非主属性对码的传递函数依赖,但不是 BCNF,因为职工号是决定因素,职工号不包含候选码。

给出 WES 的一个关系实例,如图 5.3 所示,我们仍可发现一些问题,例如,某位职工刚分配到一个仓库工作,但尚未负责具体设备,这样的信息就无法插入。另外,如果插入记录 ('WH3', 'E7', 'S4'),这样职工 S4 将同时属于 WH3 和 WH2,这是违背第(2)条语义的,但却无法防止。

解决以上问题的方法仍然是模式分解,如分解为 WS(仓库号,职工号),WE(职工号,设备号),这样可以解决上述提到的问题,而且 WS、WE 都是 BCNF。但是这样的分解破坏了第(3)条语义,即函数依赖关系(仓库号,设备号)→职工号在模式分解后丢失。

5.3.5 多值依赖与第四范式

上面讨论的都是函数依赖范畴内的关系模式的范式问题。一个关系模式达到 BCNF 以后是否很完美了呢?看一看下面对例 5.3 的进一步分析。

对例 5.3 中提到的关系模式电力设备管理 WES(仓库号,设备号,职工号),给它赋予新的语义如下:

(1) 一个仓库可以有多个职工,每个职工可以管理一个仓库中的多种设备。

(2) 一名职工可以管理多个仓库的设备。

(3) 每种设备可以存放在多个仓库。

可以用一个非规范化的关系来表示三者之间的关系,如图 5.4 所示。

把图 5.4 转化成规范化的关系,如图 5.5 所示。

很显然,该关系模式具有唯一的候选码(仓库号,设备号,职工号),即全码,因而属于 BCNF,但该模式仍存在着一些问题。

仓库号	设备号	职工号
WH1	E1	S1
WH1	E2	S1
WH1	E3	S1
WH1	E4	S2
WH1	E5	S2
WH1	E6	S2
WH1	E7	S2
WH2	E4	S3
WH2	E1	S3
WH2	E3	S4
WH2	E2	S4
WH2	E7	S4
WH3	E5	S5
WH3	E6	S5
WH3	E8	S5

图 5.3 WES 的一个实例

仓库	职工	设备
WH1	S1	E1
	S2	E2
WH2	S1	E2
	S3	E3

图 5.4 规范前的关系

仓库	职工	设备
WH1	S1	E1
WH1	S1	E2
WH1	S1	E3
WH1	S2	E1
WH1	S2	E2
WH1	S2	E3
WH2	S1	E2
WH2	S1	E3
WH2	S3	E2
WH2	S3	E3

图 5.5 规范后的关系

(1) 数据冗余: 仓库号、设备号、职工号的信息被多次存储。

(2) 插入异常: 例如,职工 S4 被分配到 WH1 仓库工作,这时必须插入('WH1','S4','E1'),('WH1','S4','E2'),('WH1','S4','E3')3 个元组。

(3) 更新异常: 职工 S1 换成职工 S6,则要修改多行记录。

(4) 删除异常: 如果 E3 不再存放在 WH1 仓库中,这时要删除元组('WH1','S1','E3'),('WH1','S2','E3'),('WH1','S4','E3')。

BCNF 的关系模式 WES 之所以会产生上述问题,主要是因为以下两个原因:

- (1) 对于一个“仓库”值,例如 WH1,有多个“设备”值与之对应。
- (2) 仓库与设备的对应关系与职工无关。

从上述两个方面可以看出,仓库与设备之间的联系显然不是函数依赖,在此称为多值依赖(Multivalue Dependence, MVD)。

定义 5.11 设 $R(U)$ 是属性集 U 上的一个关系模式。 X, Y, Z 是 U 的子集,并且 $Z = U - X - Y$ 。关系模式 $R(U)$ 中多值依赖 $X \twoheadrightarrow Y$ 成立,当且仅当对 $R(U)$ 的任一关系 r ,给定的一对 (x, z) 值有一组 Y 的值,这组值仅仅决定于 x 值而与 z 值无关。

如果 $X \twoheadrightarrow Y$, 而 $Z = \Phi$, 则称 $X \twoheadrightarrow Y$ 为平凡的多值依赖,否则称 $X \twoheadrightarrow Y$ 为非平凡的多值依赖。

结合上面的电力设备管理关系模式,很显然存在着非平凡的多值依赖,即仓库 $\twoheadrightarrow$ 设备,仓库 $\twoheadrightarrow$ 职工。

多值依赖具有如下性质。

- 对称性: 若 $X \twoheadrightarrow Y$, 则 $X \twoheadrightarrow Z$, 其中 $Z = U - X - Y$ 。
- 传递性: 若 $X \twoheadrightarrow Y, Y \twoheadrightarrow Z$, 则 $X \twoheadrightarrow Z - Y$ 。
- 合并性: 若 $X \twoheadrightarrow Y, X \twoheadrightarrow Z$, 则 $X \twoheadrightarrow YZ$ 。
- 分解性: 若 $X \twoheadrightarrow Y, X \twoheadrightarrow Z$, 则 $X \twoheadrightarrow (Y \cap Z), X \twoheadrightarrow Z - Y, X \twoheadrightarrow Y - Z$ 均成立。

即如果两个相交的属性子集均多值依赖于另一个属性子集,则这两个属性子集因相交而分割成的三部分也都多值依赖于该属性子集。

- 函数依赖可看作多值依赖的特例: 即若 $X \rightarrow Y$, 则 $X \twoheadrightarrow Y$ 。这是因为当 $X \rightarrow Y$ 时,对 X 的每一个值 x , Y 都有一个确定的值 y 与之对应,所以 $X \twoheadrightarrow Y$ 。

有了多值依赖,是否意味着不需要函数依赖了呢? 恰恰相反,一般来讲,不仅要找出关系模式中的所有多值依赖,而且要找出关系模式中的所有函数依赖,这样,一个完整的关系模式就可能既包含一个函数依赖集,又包含一个多值依赖集。

从上面的例子可以看出,一个存在多值依赖的关系模式存在着严重的数据异常现象,如果把它分解为 WS(仓库,职工)和 WE(仓库,设备)两个关系,它们的数据异常情况会得到很好的解决。

在职工关系模式中,虽然也有仓库 $\twoheadrightarrow$ 职工,但它是平凡的多值依赖,同样设备关系模式中也存在平凡的多值依赖仓库 $\twoheadrightarrow$ 设备,为此引入第四范式(4NF)的概念。

定义 5.12 关系模式 $R \langle U, F \rangle \in 1NF$, 如果对于 R 的每个非平凡多值依赖 $X \twoheadrightarrow Y (Y \not\subseteq X)$, X 都含有码,则称 $R \langle U, F \rangle \in 4NF$ 。

通过 4NF 的定义可知,4NF 就是限制关系模式的属性之间不允许有非平凡且非函数依赖的多值依赖。

在前面讨论的 WES 关系模式中,仓库 $\twoheadrightarrow$ 职工,仓库 $\twoheadrightarrow$ 设备,它们都是非平凡的多值依赖。而仓库不是码,关系模式的码是(仓库,职工,设备),故该模式不是 4NF。分解后的关系 WS 和 WE 中虽然分别有仓库 $\twoheadrightarrow$ 职工,仓库 $\twoheadrightarrow$ 设备,但它们都是平凡的多值依赖,且都不是函数依赖,因此都是 4NF。

5.3.6 关系模式的规范化

在关系数据库中,对关系模式的最基本的规范化要求就是每个分量不可再分,在此基础上逐步消除不合适的数据依赖。

关系模式的规范化具体可分为以下几步:

(1) 对 1NF 关系进行投影,消除原关系中非主属性对码的部分函数依赖,将 1NF 关系转换为若干个 2NF 关系。

(2) 对 2NF 关系进行投影,消除原关系中非主属性对码的传递函数依赖,将 2NF 关系转换为若干个 3NF 关系。

(3) 对 3NF 关系进行投影,消除原关系中主属性对码的部分函数依赖和传递函数依赖(即决定因素都包含一个候选码),得到一组 BCNF 关系。

上述 3 步可以概括为一步:对原关系进行投影,消除决定属性不是候选码的任何函数依赖。

(4) 对 BCNF 关系进行投影,消除原关系中非平凡且非函数依赖的多值依赖,得到一组 4NF 关系。

其基本步骤如图 5.6 所示。

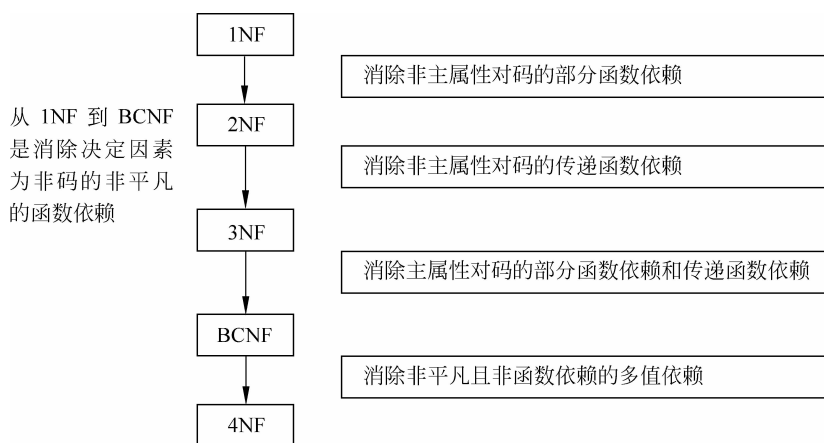


图 5.6 规范化过程

在数据库规范化过程中,如果范式过低可能会存在插入异常、删除异常、更新异常、数据冗余等问题,需要转换为高一级的范式。高范式的优点是避免数据冗余,提高数据完整性,减少数据库的空间;缺点是操作难,经常需要多表连接,数据库性能影响极大。所以并不意味着规范化程度越高的关系模式就越好,例如,当我们对数据库的操作主要是查询而更新较少时,为了提高查询效率,可能宁愿保留适当的数据冗余,让关系模式中的属性多一些,而不愿把模式分解得太小,否则为了查询一些数据,常常要做大量的连接运算,这样会花费大量时间,或许得不偿失,因此保留适量冗余,达到以空间换时间的目的。目前的计算机技术,空间不是问题,但对查询速度要求极高。因此,在充分利用关系数据库的 3 个完整性约束条件保证数据完整性的基础上,达到 3NF 已经足够了,这也是模式分解的一个重要原则。

5.4 函数依赖的公理系统

W. W. Armstrong 在 1974 年提出了函数依赖的一套推理规则,即 Armstrong 公理系统。数据依赖的一个有效而完备的公理是模式分解算法的理论基础。

5.4.1 Armstrong 公理系统

定义 5.13 对于满足一组函数依赖 F 的关系模式 $R\langle U, F\rangle$,其任何一个关系 r ,若函数依赖 $X\rightarrow Y$ 都成立(即 r 中的任意两元组 t, s ,若 $t[X]=s[X]$,则 $t[Y]=s[Y]$),则称 F 逻辑蕴涵 $X\rightarrow Y$ 。

Armstrong 公理: 设关系模式 $R\langle U, F\rangle$,其中, U 为属性集, F 是 U 上的一组函数依赖,那么有如下推理规则。

- A1 自反律(Reflexivity): 若 $Y\subseteq X\subseteq U$,则 $X\rightarrow Y$ 为 F 所蕴涵。
- A2 增广律(Augmentation): 若 $X\rightarrow Y$ 为 F 所蕴涵,且 $Z\subseteq U$,则 $XZ\rightarrow YZ$ 为 F 所蕴涵。
- A3 传递律(Transitivity): 若 $X\rightarrow Y, Y\rightarrow Z$ 为 F 所蕴涵,则 $X\rightarrow Z$ 为 F 所蕴涵。

根据上述 3 条推理规则又可推出下述 3 条推理规则。

- 合并规则: 若 $X\rightarrow Y, X\rightarrow Z$,则 $X\rightarrow YZ$ 为 F 所蕴涵。
- 伪传递律: 若 $X\rightarrow Y, WY\rightarrow Z$,则 $XW\rightarrow Z$ 为 F 所蕴涵。
- 分解规则: 若 $X\rightarrow Y, Z\subseteq Y$,则 $X\rightarrow Z$ 为 F 所蕴涵。

5.4.2 闭包

根据上述合并规则和分解规则,我们很容易得到这样一个重要事实:

引理 5.1 $X\rightarrow A_1 A_2 \cdots A_k$ 成立的充分必要条件是 $X\rightarrow A_i$ 成立($i=1, 2, \cdots, k$)。

由该引理可以得出函数依赖的闭包 F^+ 和属性的闭包 X_F^+ 的定义。

定义 5.14 关系模式 $R\langle U, F\rangle$ 中为 F 所蕴涵的函数依赖的全体称为 F 的闭包,记为 F^+ 。

定义 5.15 设 F 为属性集 U 上的一组函数依赖, $X\subseteq U, X_F^+ = \{A | X\rightarrow A \text{ 能由 } F \text{ 根据 Armstrong 公理导出}\}$,则称 X_F^+ 为属性集 X 关于函数依赖集 F 的闭包。

引理 5.2 设 F 为属性集 U 上的一组函数依赖, $X\subseteq U, Y\subseteq U, X\rightarrow Y$ 能由 F 根据 Armstrong 公理导出的充分必要条件是 $Y\subseteq X_F^+$ 。

这样,判定 $X\rightarrow Y$ 是否由 F 根据 Armstrong 公理导出的问题就转化为求 X_F^+ ,判定 Y 是否为 X_F^+ 的子集的问题。这一问题可由以下算法解决。

算法 5.1 求属性集 $X(X\subseteq U)$ 关于 U 上的函数依赖集 F 的闭包 X_F^+ 。

输入: X, F

输出: X_F^+

步骤:

(1) 令 $X^{(0)} = X, i=0$;

- (2) 求 B , 这里 $B = \{A | (\exists v)(\exists w)(VW \in F \wedge V \subseteq X^{(i)} \wedge A \in W)\}$;
- (3) $X^{(i+1)} = B \cup X^{(i)}$;
- (4) 判断 $X^{(i+1)} = X^{(i)}$ 是否成立;
- (5) 若相等, 或 $X^{(i)} = U$, 则 $X^{(i)}$ 为属性集 X 关于函数依赖集 F 的闭包, 且算法终止;
- (6) 若不相等, 则 $i = i + 1$, 返回第(2)步。

【例 5.4】 已知关系模式 $R < U, F >$, $U = \{A, B, C, D, E\}$, $F = \{A \rightarrow B, D \rightarrow C, BC \rightarrow E, AC \rightarrow B\}$, 求 $(AE)_F^+$ 和 $(AD)_F^+$ 。

解: 先求 $(AE)_F^+$ 。

由上述算法, 设 $X^{(0)} = AE$ 。

计算 $X^{(1)}$: 逐一扫描 F 中的各个函数依赖, 找到左部为 A, E 或 AE 的函数依赖, 得到 $A \rightarrow B$, 故有 $X^{(1)} = AE \cup B = ABE$ 。

因为 $X^{(0)} \neq X^{(1)}$, 继续。

计算 $X^{(2)}$: 逐一扫描 F 中的各个函数依赖, 找到左部为 ABE 或 ABE 子集的函数依赖, 因为找不到这样的函数依赖, 所以 $X^{(1)} = X^{(2)}$, 算法终止, $(AE)_F^+ = ABE$ 。

再求 $(AD)_F^+$: 设 $X^{(0)} = AD$ 。

计算 $X^{(1)}$: 逐一扫描 F 中的各个函数依赖, 找到左部为 A, D 或 AD 的函数依赖, 得到 $A \rightarrow B, D \rightarrow C$ 两个函数依赖, 故有 $X^{(1)} = AD \cup BC = ABCD$ 。

计算 $X^{(2)}$: 逐一扫描 F 中的各个函数依赖, 找到左部为 $ABCD$ 或 $ABCD$ 子集的函数依赖, 得到 $BC \rightarrow E, AC \rightarrow B$ 两个函数依赖, 故有 $X^{(2)} = ABCD \cup E$, 所以 $X^{(2)} = ABCDE = U$, 算法终止, $(AD)_F^+ = ABCDE$ 。

5.4.3 函数依赖集的等价和最小化

从蕴涵的概念出发, 可以引出两个函数依赖集等价和最小函数依赖集的概念。

定义 5.16 一个关系模式 $R < U, F >$ 上的两个函数依赖集 F 和 G , 如果 $F^+ = G^+$, 则称 F 和 G 是等价的, 记作 $F \equiv G$ 。

若 $F \equiv G$, 则称 G 是 F 的一个覆盖, 反之亦然。两个等价的函数依赖集在表达能力上是完全相同的。

引理 5.3 $F^+ = G^+$ 的充分必要条件是 $F \subseteq G^+, G \subseteq F^+$ 。

定义 5.17 如果函数依赖集 F 满足下列条件, 则称 F 为最小函数依赖集或最小覆盖。

- (1) F 中的任何一个函数依赖的右部仅含有一个属性;
- (2) F 中不存在这样一个函数依赖 $X \rightarrow A$, 使得 F 与 $F - \{X \rightarrow A\}$ 等价;
- (3) F 中不存在这样一个函数依赖 $X \rightarrow A$, X 有真子集 Z 使得 $F - \{X \rightarrow A\} \cup \{Z \rightarrow A\}$ 与 F 等价。

定理 5.1 每个函数依赖集 F 均等价于一个极小函数依赖集 F_m , 此 F_m 称为 F 的依赖集。求最小函数依赖集, 可用分解的算法。

算法 5.2 求最小函数依赖集。

输入: 一个函数依赖集。

输出: F 的一个等价的最小函数依赖集 G 。

步骤:

- (1) 用分解的规则使 F 中的任何一个函数依赖的右部仅含有一个属性。
- (2) 去掉多余的函数依赖: 从第一个函数依赖 $X \rightarrow Y$ 开始将其从 F 中去掉, 然后在剩下的函数依赖中求 X 的闭包 X^+ , 看 X^+ 是否包含 Y , 若是, 则去掉 $X \rightarrow Y$; 否则不能去掉, 依次做下去, 直到找不到冗余的函数依赖。
- (3) 去掉各依赖左部多余的属性: 逐个检查函数依赖左部非单个属性的依赖 $X \rightarrow Y$, 设 $X = B_1 B_2 \cdots B_m$, 逐一考查 $B_i (i = 1, 2, \dots, m)$, 若 $Y \in (X - B_i)_F^+$, 则以 $X - B_i$ 取代 X 。

【例 5.5】 已知关系模式 $R \langle U, F \rangle$, $U = \{A, B, C, D, E, G\}$; $F = \{AB \rightarrow C, C \rightarrow A, CG \rightarrow BD, ACD \rightarrow B\}$, 求 F 的最小函数依赖集。

解:

- (1) 利用分解规则将所有的函数依赖变成右边都是单个属性的函数依赖。

$$F = \{AB \rightarrow C, C \rightarrow A, CG \rightarrow B, CG \rightarrow D, ACD \rightarrow B\}$$

- (2) 去掉 F 中多余的函数依赖, 具体可分解为以下几步。

- ① 设 $AB \rightarrow C$ 为多余的函数依赖, 则去掉 $AB \rightarrow C$ 得:

$$F_1 = \{C \rightarrow A, CG \rightarrow B, CG \rightarrow D, ACD \rightarrow B\}$$

因为从 F_1 中找不到左部为 AB 或 AB 子集的函数依赖, 则 $(AB)_{F_1}^+ = \Phi$, 所以 $AB \rightarrow C$ 为非多余的函数依赖, 不能去掉。

- ② 设 $CG \rightarrow B$ 为多余的函数依赖, 则去掉 $CG \rightarrow B$ 得:

$$F_2 = \{AB \rightarrow C, C \rightarrow A, CG \rightarrow D, ACD \rightarrow B\}$$

则 $(CG)_{F_2}^+ = ABCDG$ 。

由于 $B \subset ABCDG$, 所以 $CG \rightarrow B$ 为多余的函数依赖, 应从 F 中去掉。去掉 $CG \rightarrow B$ 后的函数依赖集仍记为 F , $F = \{AB \rightarrow C, C \rightarrow A, CG \rightarrow D, ACD \rightarrow B\}$ 。

同理, 可以推导出函数依赖 $C \rightarrow A$ 、 $CG \rightarrow D$ 及 $ACD \rightarrow B$ 均不能从 F 中去掉。

故有 $F = \{AB \rightarrow C, C \rightarrow A, CG \rightarrow D, ACD \rightarrow B\}$ 。

- (3) 去掉 F 中各依赖左部多余的属性。

因为存在函数依赖 $C \rightarrow A$, 故函数依赖 $ACD \rightarrow B$ 的属性 A 是多余的, 得到新的函数依赖集为:

$$F^1 = \{AB \rightarrow C, C \rightarrow A, CG \rightarrow D, CD \rightarrow B\}$$

考查 $AB \rightarrow C$, 由于 $A_{F^1}^+ = A, B_{F^1}^+ = B, A_{F^1}^+$ 和 $B_{F^1}^+$ 中均不包含 C , 所以 $AB \rightarrow C$ 的左边无多余属性。

同理, 可以推出 $CG \rightarrow D$ 、 $CD \rightarrow B$ 的左边均无多余属性。

所以, 极小函数依赖集 $F_{\min} = \{AB \rightarrow C, C \rightarrow A, CG \rightarrow D, CD \rightarrow B\}$ 。

需要注意的是, F 的最小函数依赖集 $F_{\min}$ 不一定是唯一的, 它与对各函数依赖 FD_i 及 $X \rightarrow A$ 中 X 各属性的处理顺序有关。

5.5 模式分解

在数据库规范化过程中, 人们为了获得操作性能较好的关系模式, 通常把一个关系模式分解为多个关系模式。模式的分解涉及属性的划分和函数依赖集的划分。

5.5.1 模式分解的准则

定义 5.18 设 F 是关系模式 $R\langle U, F \rangle$ 的函数依赖集, $U_1 \subseteq U, F_1 = \{X \rightarrow Y \mid X \rightarrow Y \in F^+ \wedge X, Y \subseteq U_1\}$, 称 F_1 是 F 在 U_1 上的投影, 记为 $F(U_1)$ 。

由上述定义可以看出, F 投影的函数依赖的左部和右部都在 U_1 中, 这些函数依赖可在 F 中出现, 也可不在 F 中出现, 但一定可由 F 推出。

【例 5.6】 已知关系模式 $R\langle U, F \rangle, U_1 = \{A, D\} \subseteq U, F = \{A \rightarrow B, B \rightarrow C, C \rightarrow D, BC \rightarrow A\}$, 求 F 在 U_1 上的投影。

解: 在 F 中没有左部和右部都在 U_1 中的函数依赖。但由 $A \rightarrow B, B \rightarrow C, C \rightarrow D$ 可以得出 $A \rightarrow D \in F^+$, 所以 $F(U_1) = \{A \rightarrow D\}$ 。

定义 5.19 关系模式 $R\langle U, F \rangle$ 的一个分解是指 $\rho = \{R_1(U_1, F_1), R_2(U_2, F_2), \dots, R_n(U_n, F_n)\}$, 其中 $U = \bigcup_{i=1}^n U_i$, 并且没有 $U_i \subseteq U_j, 1 \leq i, j \leq n, F_i$ 是 F 在 U_i 上的投影, $F_i = \{X \rightarrow Y \mid X \rightarrow Y \in F^+ \wedge X, Y \subseteq U_i\}$ 。

对于模式分解有以下说明:

- (1) 分解是完备的, U 中的属性全部分散在分解 ρ 中。
 - (2) 在分解中, 由于 U_i 的属性构成不同, 可能使某些函数依赖消失, 即不能保证分解对函数依赖集 F 是完备的, 但应尽量保留 F 所蕴涵的函数依赖, 所以对每一个子模式 R_i 均取 F 在 U_i 上的投影。
 - (3) 分解是不相同的, 不允许在 ρ 中出现一个子模式 U_i 被另一个子模式 U_j 包含的情况。
 - (4) 当需要对若干个关系模式进行分解时, 可分别对每个关系模式进行分解。
- 对一个给定的模式进行分解, 使得分解后的模式与原来的模式等价, 其判定的准则有以下 3 种:

- 分解要保持函数依赖;
- 分解具有无损连接性;
- 分解既要具有无损连接性, 又要保持函数依赖。

按照不同的分解准则, 模式所能达到的分离程度各不相同, 各种范式就是对分离程度的测度。

5.5.2 分解的函数依赖保持性和无损连接性

首先看一个模式分解的例子。

【例 5.7】 已知关系模式 $R\langle U, F \rangle$, 其中, $U = \{\text{仓库号}, \text{所在区域}, \text{区域主管}, \text{设备号}, \text{数量}\}$; $F = \{\text{仓库号} \rightarrow \text{所在区域}, \text{所在区域} \rightarrow \text{区域主管}, (\text{仓库号}, \text{设备号}) \rightarrow \text{数量}\}$ 。

$R\langle U, F \rangle$ 的一个分解 $\rho_1: U_1 = \{\text{仓库号}, \text{所在区域}\}, F_1 = \{\text{仓库号} \rightarrow \text{所在区域}\}; U_2 = \{\text{所在区域}, \text{区域主管}\}, F_2 = \{\text{所在区域} \rightarrow \text{区域主管}\}; U_3 = \{\text{仓库号}, \text{设备号}, \text{数量}\}, F_3 = \{(\text{仓库号}, \text{设备号}) \rightarrow \text{数量}\}$ 。此时 $F_1 \cup F_2 \cup F_3 = F$, 该分解没有丢失函数依赖。

$R\langle U, F \rangle$ 的另一个分解 $\rho_2: U_1 = \{\text{仓库号}, \text{所在区域}, \text{区域主管}\}, F_1 = \{\text{仓库号} \rightarrow \text{所在}$

区域,所在区域→区域主管}; $U_2 = \{\text{设备号,数量}\}$, $F_2 = \Phi$ 。此时 $F_1 \cup F_2 \neq F$, F 中每个仓库的每种设备都有一个库存数量的语义丢失了。

由此可见,不会因模式分解而丢失函数依赖是分解的一个重要标准。

定义 5.20 设 $\rho = \{R_1 \langle U_1, F_1 \rangle, R_2 \langle U_2, F_2 \rangle, \dots, R_n \langle U_n, F_n \rangle\}$ 是关系模式 $R \langle U, F \rangle$ 上的一个分解。若 $\bigcup_{i=1}^n F_i^+ = F^+$, 则称分解 ρ 具有函数依赖保持性。

如对上述例 5.7 的分解 ρ_1 就是一个保持函数依赖的分解。在分解 ρ_1 中,保留了 $R \langle U, F \rangle$ 中的所有语义,即所有函数依赖。函数依赖的保持性反映了模式分解的依赖等价原则,依赖等价保证了分解后的模式与原有的模式数据语义上的一致性。

在模式分解时,除了希望保持函数依赖外,还希望分解后的关系再连接时能恢复到分解前的状态,这就是所谓的无损连接分解。

定义 5.21 设 $\rho = \{R_1 \langle U_1, F_1 \rangle, R_2 \langle U_2, F_2 \rangle, \dots, R_n \langle U_n, F_n \rangle\}$ 是关系模式 $R \langle U, F \rangle$ 上的一个分解。若任何属于 $R \langle U, F \rangle$ 的关系 r , 令 $r_1 = \pi_{R_1}(r)$, $r_2 = \pi_{R_2}(r)$, $\dots$, $r_n = \pi_{R_n}(r)$, 有 $r = r_1 \bowtie r_2 \bowtie \dots \bowtie r_n$ 成立,则称分解 ρ 具有无损连接性。

在这里, $\pi_{R_i}(r)$ ($i=1, 2, \dots, n$) 是 r 在 U_i 上的投影。

一个分解可能只满足函数依赖或只满足无损连接,或同时满足二者,最理想的情况是同时满足二者,次之是满足无损连接性。看下面的例子:

【例 5.8】 设关系模式 $R \langle U, F \rangle$ 中 $U = \{S, T, U, V\}$, $F = \{S \rightarrow T, U \rightarrow V\}$, 请分析下列分解 ρ 的函数依赖保持性和无损连接性:

$\rho: U_1 = \{S, T\}, F_1 = \{S \rightarrow T\}$ 。

$U_2 = \{U, V\}, F_2 = \{U \rightarrow V\}$ 。

解: 因为 $F_1 \cup F_2 = F$, 故 ρ 具有函数依赖保持性,但 ρ 不具有无损连接性,例如可以对 r 进行如下分析:

r :	S	T	U	V
	s ₁	t ₁	u ₁	v ₁
	s ₂	t ₂	u ₂	v ₂

r ₁ :		r ₂ :	
S	T	U	V
s ₁	t ₁	u ₁	v ₁
s ₂	t ₂	u ₂	v ₂

r ₁ ⋈ r ₂ :	S	T	U	V
	s ₁	t ₁	u ₁	v ₁
	s ₁	t ₁	u ₂	v ₂
	s ₂	t ₂	u ₁	v ₁
	s ₂	t ₂	u ₂	v ₂

很显然 $r \neq r_1 \bowtie r_2$, 故 ρ 不是无损连接分解。

采用定义 5.21 来鉴别一个分解的无损性是比较困难的,定理 5.2 和算法 5.3 将给出判别的方法。

定理 5.2 设 $R\langle U, F \rangle$, $\rho = \{R_1\langle U_1, F_1 \rangle, R_2\langle U_2, F_2 \rangle\}$ 是 R 的一个分解, F 是 R 上的函数依赖, ρ 具有无损连接性的充要条件为:

$$(U_1 \cap U_2) \rightarrow (U_1 - U_2) \in F^+$$

或

$$(U_1 \cap U_2) \rightarrow (U_2 - U_1) \in F^+$$

【例 5.9】 设 $R\langle U, F \rangle$, $U = \{A, B, C\}$, $F = \{A \rightarrow B, C \rightarrow B\}$, 分解 $\rho_1 = \{AB, BC\}$, 则:

$$U_1 \cap U_2 = B, \quad U_1 - U_2 = A, \quad B \rightarrow A \notin F^+$$

$$U_2 - U_1 = C, \quad B \rightarrow C \notin F^+$$

所以, 分解 ρ_1 不具有无损连接性。

分解 $\rho_2 = \{AC, BC\}$, 则:

$$U_1 \cap U_2 = C, \quad U_1 - U_2 = A, \quad C \rightarrow A \notin F^+$$

$$U_2 - U_1 = B, \quad C \rightarrow B \in F^+$$

所以, 分解 ρ_2 具有无损连接性。

算法 5.3 判别一个分解的无损连接性。

设 $\rho = \{R_1\langle U_1, F_1 \rangle, R_2\langle U_2, F_2 \rangle, \dots, R_k\langle U_k, F_k \rangle\}$ 是关系模式 $R\langle U, F \rangle$ 上的一个分解, $U = \{A_1, A_2, \dots, A_n\}$, $F = \{FD_1, FD_2, \dots, FD_p\}$, 设 F 是一个最小函数依赖集, 记函数依赖 FD_i 为 $X_i \rightarrow A_{ij}$, 则判定步骤如下:

(1) 建立一张 n 列 k 行的表, 每一列对应一个属性, 每一行对应分解中的一个关系模式。若属性 $A_j \in U_i$, 则在 j 列 i 行上填上 a_j , 否则填上 b_{ij} 。

(2) 对于每一个 FD_i 做如下操作: 找到 X_i 对应的列中具有相同符号的行。考察这些行中 l_i 列的元素, 若其中有 a_{li} , 则全部改为 a_{li} , 否则全部改为 b_{mli} , m 是这些行的行号最小值。

应当注意的是, 若某个 b_{li} 被更改, 那么该表的 l_i 列中凡是 b_{li} 的符号 (不管它是开始找到的哪些行) 均应做相应的更改。

如果在某次更改后有一行成为 $a_1, a_2, \dots, a_n$, 则算法终止, ρ 具有无损连接性, 否则 ρ 不具有无损连接性。

对 F 中 p 个 FD 逐一进行一次这样的处理, 称为对 F 的一次扫描。

(3) 比较扫描前后表有无变化, 如有变化, 则返回第 (2) 步, 否则算法终止。若发生循环, 那么前次扫描至少应使该表减少一个符号, 表中符号有限, 因此, 循环必然终止。

【例 5.10】 设有关系模式 $R\langle U, F \rangle$, 其中:

$$U = \{A, B, C, D, E\}, F = \{A \rightarrow D, E \rightarrow D, D \rightarrow B, BC \rightarrow D, DC \rightarrow A\}.$$

判断 $\rho = \{AB, AE, CE, BCD, AC\}$ 是否为无损连接分解。

解: (1) 构造 ρ 的无损连接性的初始判断表如图 5.7(a) 所示。

(2) 逐一考查 F 中的函数依赖 $F = \{A \rightarrow D, E \rightarrow D, D \rightarrow B, BC \rightarrow D, DC \rightarrow A\}$:

考查 $A \rightarrow D$, 因为属性列 A 上的 1、2、5 行的值都为 a_1 , 因此可使属性列 D 上对应的值全相同, 例如将 b_{24} 、 b_{54} 改为 b_{14} , 如图 5.7(b) 所示。

考查 $E \rightarrow D$, 因为属性列 E 的第 2、3 行的值为 a_5 , 因此 D 属性列上对应的值也相等, 可将 b_{34} 改为 b_{14} , 如图 5.7(c) 所示。

R_i	A	B	C	D	E
AB	a1	a2	b13	b14	b15
AE	a1	b22	b23	b24	a5
CE	b31	b32	a3	b34	a5
BCD	b41	a2	a3	a4	b45
AC	a1	b52	a3	b54	b55

(a)

R_i	A	B	C	D	E
AB	a1	a2	b13	b14	b15
AE	a1	b22	b23	b14	a5
CE	b31	b32	a3	b34	a5
BCD	b41	a2	a3	a4	b45
AC	a1	b52	a3	b14	b55

(b)

R_i	A	B	C	D	E
AB	a1	a2	b13	b14	b15
AE	a1	b22	b23	b14	a5
CE	b31	b32	a3	b14	a5
BCD	b41	a2	a3	a4	b45
AC	a1	b52	a3	b14	b55

(c)

R_i	A	B	C	D	E
AB	a1	a2	b13	b14	b15
AE	a1	a2	b23	b14	a5
CE	b31	a2	a3	b14	a5
BCD	b41	a2	a3	a4	b45
AC	a1	a2	a3	b14	b55

(d)

R_i	A	B	C	D	E
AB	a1	a2	b13	b14	b15
AE	a1	a2	b23	b14	a5
CE	b31	a2	a3	a4	a5
BCD	b41	a2	a3	a4	b45
AC	a1	a2	a3	a4	b55

(e)

R_i	A	B	C	D	E
AB	a1	a2	b13	b14	b15
AE	a1	a2	b23	b14	a5
CE	a1	a2	a3	a4	a5
BCD	a1	a2	a3	a4	b45
AC	a1	a2	a3	a4	b55

(f)

图 5.7 无损连接分解的判定

考查 $D \rightarrow B$, 因为 D 列上第 1、2、3、5 行的值相同, 均为 b14, 因此 B 列上对应行的值也应相同, 将 b22、b32、b52 全改为 a2, 如图 5.7(d) 所示。

考查 $BC \rightarrow D$, BC 列的第 3、4、5 行值相同, 均为 a2、a3, 可使得 D 列的第 3、4、5 行的值全为 a4, 如图 5.7(e) 所示。

考查 $DC \rightarrow A$, DC 列上第 3、4、5 行的值相同, 都为 a3、a4, 所以可将 b31、b41 全改为 a1, 如图 5.7(f) 所示。

(3) 因为第 3 行已出现了 a1, a2, ..., a5 这样的行, 因此此分解具有无损连接性。

5.5.3 模式分解的算法

针对上述模式分解的准则, 规范化理论提供了一套完整的模式分解的算法, 按照这套算法可以做到以下几点:

(1) 若要求分解保持函数依赖, 那么模式分解一定能够达到 3NF, 但不一定能够达到 BCNF。

(2) 若要求分解既具有无损连接性, 又保持函数依赖, 则模式分解一定能够达到 3NF, 但不一定能够达到 BCNF。

(3) 若要求分解具有无损连接性, 那么模式分解一定能够达到 4NF。

它们分别由算法 5.4、算法 5.5 和算法 5.6 来实现。

算法 5.4 转换为 3NF 的保持函数依赖的分解。

给定关系模式 $R\langle U, F \rangle$, 求分解 $\rho, \rho = \{R_1\langle U_1, F_1 \rangle, R_2\langle U_2, F_2 \rangle, \dots, R_n\langle U_n, F_n \rangle\}$,

使得 ρ 中的关系模式 R_i 满足 3NF, 且 $\bigcup_{i=1}^n F_i^+ = F^+$ 成立。该算法可分为 4 步:

(1) 求 F 的最小函数依赖集 $F_{\min}$ 。

(2) 分组。

对 $F_{\min}$ 中的函数依赖按左边相同原则进行分组。设可分为 m 组, 即 $U_1, U_2, \dots, U_m$, 它们分别是以 $X_1, X_2, \dots, X_m$ 为左部的函数依赖集分组后得到的属性集。

(3) 吸收。

在 $\{U_1, U_2, \dots, U_m\}$ 中, 若存在 $U_i \subseteq U_j (i \neq j, i, j = 1, 2, \dots, m)$, 则用 U_j 吸收 U_i , 去掉 U_i , 经过吸收后, 得到 $U_1, U_2, \dots, U_k (k \leq m)$, 使得在 $\{U_1, U_2, \dots, U_k\}$ 中不存在 $U_i \subseteq U_j (i \neq j, i, j = 1, 2, \dots, k)$ 。

(4) 对不在 F 中出现的属性, 把它们单独作为一组, 记为 U_0 。

通过以上 4 步, 所得到的分解为保持函数依赖的 3NF 分解。

【例 5.11】 对于给定的关系模式 $R\langle U, F \rangle, U = \{A, B, C, D, E\}, F = \{AB \rightarrow C, A \rightarrow B, D \rightarrow BC, C \rightarrow B\}$, 求 R 保持函数依赖的 3NF 分解。

解:

(1) 求 F 的最小函数依赖集 $F_{\min}$ 。

① 将所有的函数依赖变成右边都是单个属性的函数依赖:

$F_1 = \{AB \rightarrow C, A \rightarrow B, D \rightarrow B, D \rightarrow C, C \rightarrow B\}$ 。

② 去掉多余的函数依赖得:

$F_2 = \{AB \rightarrow C, A \rightarrow B, D \rightarrow C, C \rightarrow B\}$ 。

③ 去掉 F_2 中各依赖左边多余的属性得:

$$F_3 = \{A \rightarrow C, A \rightarrow B, D \rightarrow C, C \rightarrow B\}.$$

对 F_3 进行分析,发现还存在多余的函数。

④ 去掉 F_3 中多余的函数依赖得:

$$F_{\min} = \{A \rightarrow C, D \rightarrow C, C \rightarrow B\}.$$

(2) 分组: $U_1 = \{A, C\}, U_2 = \{D, C\}, U_3 = \{B, C\}$ 。

(3) 吸收: $\{U_1, U_2, U_3\}$ 中不存在需要吸收的子模式。

(4) 因为 $E \notin \{A, B, C, D\}$, 所以 $U_0 = \{E\}$ 。

由此得到 R 的一个保持函数依赖的分解 ρ 为:

$$R_0: U_0 = \{E\}, F_0 = \Phi;$$

$$R_1: U_1 = \{A, C\}, F_1 = \{A \rightarrow C\};$$

$$R_2: U_2 = \{D, C\}, F_2 = \{D \rightarrow C\};$$

$$R_3: U_3 = \{B, C\}, F_3 = \{C \rightarrow B\}.$$

算法 5.5 转换为 3NF 的保持函数依赖和无损连接的分解。

其步骤也分为 4 步:

(1) 求 F 的最小函数依赖集 $F_{\min}$ 。

(2) 分组: 与算法 5.4 相同。

(3) 吸收: 与算法 5.4 相同。

(4) 若存在 R_i , 使得 U_i 中包含 R 的码, 则算法结束。否则, 令 X 是 R 的码, 把 $R(X, F_x)$ 添加到分解 ρ 中, 算法结束。

【例 5.12】 对于给定的关系模式 $R \langle U, F \rangle$, $U = \{A, B, C, D, E\}$, $F = \{AB \rightarrow D, C \rightarrow B, B \rightarrow C, BD \rightarrow A, D \rightarrow A\}$ 。试将 R 保持函数依赖且无损连接地分解到 3NF。

解:

(1) 求 F 的最小函数依赖集 $F_{\min}$ 。

$$F_{\min} = \{AB \rightarrow D, C \rightarrow B, B \rightarrow C, D \rightarrow A\} \quad (\text{过程略})$$

(2) 分组:

$$U_1 = \{B, C\}, \quad F_1 = \{C \rightarrow B, B \rightarrow C\}$$

$$U_2 = \{A, B, D\}, \quad F_2 = \{AB \rightarrow D, D \rightarrow A\}$$

$$U_3 = \{A, D\}, \quad F_3 = \{D \rightarrow A\}$$

(3) 吸收:

由于 $U_3 \subseteq U_2$, 去掉 U_3 , 得到

$$\rho = \{R_1 \langle U_1, F_1 \rangle, R_2 \langle U_2, F_2 \rangle\}$$

(4) ABE 是 R 的码, 因为 ρ 中各子模式无 R 的码, 将 $U_3 = \{A, B, E\}$ 加入 ρ 中, 最后得:

$$\rho: U_1 = \{B, C\}, \quad F_1 = \{C \rightarrow B, B \rightarrow C\}$$

$$U_2 = \{A, B, D\}, \quad F_2 = \{AB \rightarrow D, D \rightarrow A\}$$

$$U_3 = \{A, B, E\}, \quad F_3 = \Phi$$

算法 5.6 转换为 BCNF 的无损连接分解算法。

其步骤为:

(1) 求 F 的最小函数依赖集 $F_{\min}$, 令 $\rho = \{R\}$ 。

(2) 二项分解:

① 若 ρ 中所有 R_i 是 BCNF 范式, 则算法结束, ρ 为所要求的分解, 否则转②。

② 若存在 $R_i \in \rho$, R_i 不是 BCNF 范式, 则在 R_i 的 F_i 上存在函数依赖集 $X \rightarrow Y$, 且 X 不是 R_i 的码。

③ 对②中已找到的 $X \rightarrow Y$ 做如下二项分解:

$R_{i1}: U_{i1} = X \cup Y, F_{i1} = F(u_{i1});$

$R_{i2}: U_{i2} = U - Y, F_{i2} = F(u_{i2})。$

④ 去掉 R_i , 并将 R_{i1} 和 R_{i2} 添加到 ρ 中, 即 $\rho = \rho \cup \{R_{i1}, R_{i2}\} - \{R_i\}$, 转到①执行。

【例 5.13】 对于给定的关系模式 $R \langle U, F \rangle$, 设 $U = \{A, B, C, D, E\}$, $F = \{ABE \rightarrow C, BC \rightarrow E\}$, 求 R 的 BCNF 无损连接的分解。

解:

(1) 求 F 的最小函数依赖集 $F_{\min}$ 。

F 已经是最小函数依赖集, 有 $F_{\min} = F$ 。

(2) 二项分解:

由于 R 的候选码为 $ABCD$ 及 $ABDE$, 所以函数依赖 $ABE \rightarrow C$ 及 $BC \rightarrow E$ 的左边均不包含码, 所以 $R \notin \text{BCNF}$ 。选择 $BC \rightarrow E$ 做二项分解:

$R_1: U_1 = \{B, C, E\}, F_1 = \{BC \rightarrow E\};$

$R_2: U_2 = \{A, B, C, D\}, F_2 = \Phi。$

R_1 的码为 BC , $R_1 \in \text{BCNF}$, R_2 的码为 $ABCD$ (全码), $R_2 \in \text{BCNF}$ 。

故得 $R \langle U, F \rangle$ 的一个 BCNF 分解 $\rho: U_1 = \{B, C, E\}, F_1 = \{BC \rightarrow E\}; U_2 = \{A, B, C, D\}, F_2 = \Phi。$

小结

本章主要讨论了关系数据库中的模式设计问题, 首先由关系模式的存储异常问题引出了函数依赖的概念, 其中包括完全函数依赖、部分函数依赖和传递函数依赖, 这些概念是规范化理论的依据和规范化程度的准则。

一个关系只要其分量不可再分, 则满足 1NF; 消除 1NF 关系中非主属性对码的部分函数依赖, 得到 2NF; 消除 2NF 关系中非主属性对码的传递函数依赖, 得到 3NF; 消除 3NF 关系中主属性对码的部分依赖和传递函数依赖, 可得到一组 BCNF 关系。这 4 种范式讨论的都是函数依赖范畴内的关系模式的范式问题。

对 BCNF 关系进行投影, 消除原关系中非平凡且非函数依赖的多值依赖, 得到一组 4NF 关系。函数依赖和多值依赖是数据依赖的重要组成部分。数据依赖的理论基础是 Armstrong 公理系统, 该系统是正确的、完备的。

关系模式在分解时应保持“等价”, 有数据等价和语义等价两种, 分别用无损连接分解和保持函数依赖两个特征来衡量。前者能保持关系在投影操作以后, 经过连接操作仍能恢复回来, 保持不变; 后者能保证数据在投影或连接中语义不变, 即不违反函数依赖的语义。由

此而确定的判定模式分解前后是否等价的准则有保持函数依赖、具有无损连接性,以及既要具有无损连接性,又要保持函数依赖 3 种。

习题

一、选择题

1. 关系数据库中的二维表至少是()。
A. 1NF B. 2NF C. 3NF D. BCNF
2. 各级范式之间的关系为()。
A. $1NF \subset 2NF \subset 3NF \subset BCNF$ B. $1NF \subset 2NF \subset BCNF \subset 3NF$
C. $1NF \supset 2NF \supset 3NF \supset BCNF$ D. $1NF \supset 2NF \supset BCNF \supset 3NF$
3. 属于 1NF 的关系模式消除了部分函数依赖,则必定属于()。
A. 3NF B. 2NF C. 1NF D. BCNF
4. 关系数据库规范化设计是为了解决关系数据库中()问题而引入的。
A. 插入、删除异常和数据冗余 B. 提高检索速度
C. 减少数据操作的重复性 D. 保证数据的安全性和完整性
5. 二维表的候选码可以有(),主键有()。
A. 1 个,1 个 B. 0 个,1 个
C. 1 个或多个,1 个或多个 D. 多个,1 个
6. 设关系模式 $R\langle U, F \rangle$, U 为 R 的属性集合, F 为 U 上的一种函数依赖,则对 $R\langle U, F \rangle$ 而言,如果 $X \rightarrow Y$ 为 F 所蕴涵,且 $Z \subseteq U$,则 $XZ \rightarrow YZ$ 为 F 所蕴涵,这是函数依赖的()。
A. 传递性 B. 合并规则 C. 自反律 D. 增广律
7. $X \rightarrow A_i (i=1, 2, \dots, k)$ 成立是 $X \rightarrow A_1 A_2 \dots A_k$ 成立的()。
A. 充分条件 B. 必要条件
C. 充要条件 D. 既不充分也不必要条件
8. 设一关系模式为运货路径(顾客姓名,顾客地址,商品名,供应商姓名,供应商地址),则该关系模式的主键是()。
A. 顾客姓名,供应商姓名 B. 顾客姓名,商品名
C. 顾客姓名,商品名,供应商姓名 D. 顾客姓名,顾客地址,商品名
9. 设有关系模式 $R\langle U, F \rangle$, U 是 R 的属性集合, X, Y 是 U 的子集,则多值函数依赖的传递律为()。
A. 如果 $X \twoheadrightarrow Y$, 且 $Y \twoheadrightarrow Z$, 则 $X \twoheadrightarrow Z$
B. 如果 $X \twoheadrightarrow Y, Y \twoheadrightarrow Z$, 则 $X \twoheadrightarrow (Z - Y)$
C. 如果 $X \twoheadrightarrow Y$, 则 $X \twoheadrightarrow (U - Y - X)$
D. 如果 $X \twoheadrightarrow Y, V \subseteq W$, 则 $WX \twoheadrightarrow VY$
10. 下列有关范式的叙述正确的是()。
A. 如果关系模式 $R \in 1NF$, 且 R 中主属性完全函数依赖于主键, 则 R 是 2NF

- B. 如果关系模式 $R \in 3NF, X, Y \in U$, 若 $X \rightarrow Y$, 则 R 是 BCNF
- C. 如果关系模式 $R \in BCNF$, 若 $X \twoheadrightarrow Y$ 是平凡的多值依赖, 则 R 是 4NF
- D. 一个关系模式如果属于 4NF, 则一定属于 BCNF, 反之不成立

二、简答题

1. 解释下列术语的含义:

函数依赖、平凡函数依赖、非平凡函数依赖、部分函数依赖、完全函数依赖、传递函数依赖、1NF、2NF、3NF、BCNF、多值依赖、4NF、最小函数依赖集、函数依赖保持性、无损连接性。

2. 什么叫关系模式分解? 模式分解要遵循什么准则?
3. 3NF 与 BCNF 的区别和联系是什么?
4. 试证明全码(All-Key)的关系必是 3NF, 也必是 BCNF。
5. 设一关系为学生(学号, 姓名, 年龄, 所在系, 出生日期), 判断此关系属于第几范式, 为什么?
6. 关系规范化一般应遵循的原则是什么?

三、综合题

1. 设关系模式 $R(A, B, C, D)$, 如果规定关系中 B 值与 D 值之间是一对多联系, A 值与 C 值之间是一对多联系, 试写出相应的函数依赖。

2. 设关系模式 $R(A, B, C)$, F 是 R 上成立的函数依赖集, $F = \{A \rightarrow C, B \rightarrow C\}$, 求 R 的码和 F 在模式 AB 上的投影。

3. 设关系模式 $R(A, B, C, D)$, F 是 R 上成立的函数依赖集, $F = \{A \rightarrow B, C \rightarrow D, D \rightarrow B\}$, 分析 $\{AD, BC, BD\}$ 相对于 R 是否具有无损连接性?

4. 下列关系模式最高属于第几范式? 候选码是什么? 解释原因。

- (1) $R(A, B, C, D, E)$, 函数依赖为 $AB \rightarrow CE, E \rightarrow AB, C \rightarrow D$;
- (2) $R(A, B, C, D)$, 函数依赖为 $B \rightarrow D, D \rightarrow B, AB \rightarrow C$;
- (3) $R(A, B, C)$, 函数依赖为 $A \rightarrow B, B \rightarrow A, A \rightarrow C$;
- (4) $R(A, B, C, D)$, 函数依赖为 $A \rightarrow C, D \rightarrow B$ 。

5. 关系 $R(A, B, C, D, E, F, G, H, I, J)$ 满足下列函数依赖:

$$\{ABD \rightarrow E, AB \rightarrow G, B \rightarrow F, C \rightarrow J, CJ \rightarrow I, G \rightarrow H\}$$

该函数依赖集是最小函数依赖集吗? 给出该关系的候选码。

6. 已知学生关系模式 $S(Sno, Sname, SD, Sdname, Course, Grade)$, 其中 Sno 表示学号; $Sname$ 表示姓名; SD 表示系名; $Sdname$ 表示系主任名; $Course$ 表示课程; $Grade$ 表示成绩。

- (1) 写出关系模式 S 的基本函数依赖;
 - (2) 求 S 的主键;
 - (3) 将关系模式无损并保持函数依赖地分解成 3NF。
7. 在一个订货数据中存有顾客、货物、订货单的信息。

每个顾客数据库包含顾客号(唯一的)、收货地址(一个顾客可以有几个地址)、赊购限

额、余额以及折扣。

每个订货单包含顾客号、收货地址、订货日期、订货细则(每个订货单有若干条)。

每条订货细则内容为货物号和订货数量。

每一种货物包含货物号、制造厂商、每个厂商的实际存货量、规定的最低存货量和货物描述。

由于处理上的要求,每个订货单的每一细则中还有一个未发量(此值初始时为订货数量,随着发货将减为0)。

请设计一个数据库模式,并给出一个合理的数据依赖。