

本章介绍关系数据库的规范化设计理论(即“模式设计理论”)。这个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。其中数据依赖起着核心作用,数据依赖研究数据之间的联系,范式是关系模式的标准。

规范化设计理论对关系数据库结构的设计起着重要的作用。

本章介绍最重要的一种数据依赖——函数依赖,关系模式分解的两大特性——无损分解和保持依赖以及范式、模式分解方法,最后介绍模式进一步规范化的内容。

5.1 关系模式的设计问题

5.1.1 关系模型的外延和内涵

一个关系模型包括外延(Extension)和内涵(Intension)两个方面的内容。

外延就是通常所说的关系、表或当前值。由于用户经常对关系进行插入、删除和修改操作,因此外延是与时间有关的,随着时间的推移在不断变化。

内涵是对数据的定义以及数据完整性约束的定义。对数据的定义包括对关系、属性、域的定义和说明。对数据完整性约束的定义涉及面较广,主要包括以下两个方面。

静态约束:涉及数据之间的联系(称为“数据依赖”(Data Dependences))、主键和值域的设计。

动态约束:定义各种操作(插入、删除、修改)对关系值的影响。一般,把内涵称为关系模式,所以关系模式应包括这些内容。

5.1.2 泛关系模式与数据库模式

本章讨论问题的框架如图 5.1 所示。

对于一个现实问题,它有一个属性集 U ,其中每个属性 A_i 对应一个值域,而不同的属性可以有相同的值域。现实问题的所有属性组成的关系模式记为 $R(U)$ 。关系 r 是关系模式 $R(U)$ 的当前值,是元组的集合。这样的关系模式和关系一般称为泛关系模式(Universal Relation Schema)和泛关系(Universal Relation)。

但在实际使用时,往往 $R(U)$ 和 r 不是恰当的形式,而必须用一个关系模式的集合 $\rho = \{R_1, R_2, \dots, R_k\}$ 代替 $R(U)$,其中每个 R_i 的属性是 U 的子集。有时就用 R_i 表示其属性集,

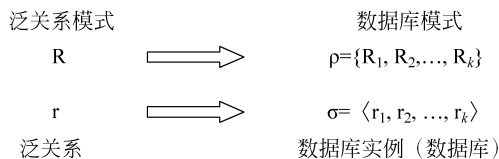


图 5.1 泛关系模式与数据库模式

因此有 $R_1 \cup R_2 \cup \dots \cup R_k = U$ 。这里 ρ 称为数据库模式(Database Schema)。对数据库模式的每一个关系模式 R_i 赋予一个当前值,就得到数据库实例(简称为数据库)。

因此,在计算机中数据并不是存储在泛关系 r 中,而是存储在数据库 σ 中。

本章主要介绍如何把泛关系模式分解成规范的、较优的数据库模式。

5.1.3 关系模式的冗余和异常问题

在数据管理中,数据冗余一直是影响系统性能的大问题。数据冗余是指同一个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出现。数据库系统克服了文件系统的这种缺陷,但对于数据冗余问题仍然应加以关注。如果一个关系模式设计得不好,仍然会出现像文件系统一样的数据冗余、异常、不一致等问题。

【例 5.1】 设有一个关系模式 $R(SNO,CNO,CNAME,TNAME)$,其属性分别表示学生学号、选修课程的课程号、课程名、任课老师姓名,具体实例如图 5.2 所示。

SNO	CNO	CNAME	TNAME
S2	C4	Pascal	WEN
S4	C4	Pascal	WEN
S6	C4	Pascal	WEN
S6	C2	ADA	LIU
S4	C2	ADA	LIU
S8	C6	BASIC	MA

图 5.2 关系模式 R 的实例

虽然这个模式只有四个属性,但在使用过程中会出现以下问题。

① 数据冗余。如果一门课程有多个学生选修,那么在关系中要出现多个元组,也就是这门课程的课程名和任课老师姓名要重复多次。

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。

- 修改异常。例如 C4 课程有三个学生选修,在关系中就会有三个元组。如果这门课程的教师改为 CHEN 老师,那么这三个元组的教师姓名都要改为 CHEN 老师。若有一个元组的教师姓名未改,就会造成这门课程的任课老师不唯一,产生不一致现象。
- 插入异常。如果需安排一门新课程(C8,DELPHI,CHEN),在尚无学生选修时,要把这门课程的数据值存储到关系中去时,在属性 SNO 上就会出现空值。在数据库技术中空值的语义是非常复杂的,对带空值元组的检索和操作也十分麻烦。
- 删除异常。如果在图 5.2 中要删除学生 S8 选课元组,那么就要把这门课程的课程名和教师姓名一起删除,这也是一种不合适的现象。

因此,关系模式 R 的设计不是一个合适的设计。

在例 5.1 中,关系模式 R 存在数据冗余和操作异常现象。如果用下面两个关系模式 R1 和 R2 代替 R: $R1(SNO,CNO)$ 和 $R2(CNO,CNAME,TNAME)$,其关系实例如图 5.3 所示。

SNO	CNO		CNO	CNAME	TNAME
S2	C4		C4	Pascal	WEN
S4	C4		C2	ADA	LIU
S6	C4		C6	BASIC	MA
S6	C2				
S4	C2				
S8	C6				

(a) 关系模式R1的实例

(b) 关系模式R2的实例

图 5.3 关系模式 R2 的实例

这样分解后,例 5.1 中提到的冗余和异常现象基本消除了。每门课程的课程名和教师姓名只存放一次,即使这门课程还没有学生选修,其课程名和教师姓名也可存放在关系 R2 中。

但是将 R 分解成 R1 和 R2 两个模式是否是最佳分解,也不是绝对的。如果要查询学生所学课程的任课教师,就要对两个关系做连接操作,连接的代价是很大的。而在原来模式 R 的关系中,就可直接找到上述结果。到底什么样的关系模式是最优的,标准是什么,如何实现,都是本章要讨论的问题。

5.1.4 本章的符号规定

- 为了便于阅读,本章对使用的符号有如下规定:
- (1) 英文字母表首部的大写字母“A,B,C,…”表示单个的属性。
 - (2) 英文字母表尾部的大写字母“…,U,V,W,X,Y,Z”表示属性集。
 - (3) 大写字母 R 表示关系模式,小写字母 r 表示其关系。为叙述方便,有时也用属性名的组合写法表示关系模式。若模式有 A、B、C 三个属性,就用 ABC 表示关系模式。
 - (4) 属性集{A₁,A₂,…,A_n}简写为 A₁A₂…A_n。属性集 X 和 Y 的并集 X∪Y 简写为 XY。X∪{A}简写为 XA 或 AX。

5.2 函数依赖

在数据库中,数据之间存在着密切的联系。在数据库技术中,把数据之间存在的联系称为“数据依赖”。设计人员的一个职责就是要把数据依赖找出来。在数据库规范化设计中,数据依赖起着关键的作用。其中,函数依赖是基本的一种依赖,它是关键码概念的推广。

5.2.1 函数依赖的定义

在数据库中,属性值之间会发生联系。例如每个学生只有一个姓名,每门课程只有一个任课教师,每个学生学一门课程只能有一个总评成绩等。这类联系称为函数依赖,其形式定义如下。

定义 5.1 设有关系模式 R(U),X 和 Y 是属性集 U 的子集,函数依赖(Functional Dependency,FD)是形为 X→Y 的一个命题,只要 r 是 R 的当前关系,对 r 中任意两个元组 t 和 s,都有 t[X]=s[X]蕴涵 t[Y]=s[Y],那么 FD X→Y 在关系模式 R(U)中成立。

这里 $t[X]$ 表示元组 t 在属性集 X 上的值,其余类同。 $X \rightarrow Y$ 读作“ X 函数决定 Y ”,或“ Y 函数依赖于 X ”。FD 是对关系模式 R 的一切可能的关系 r 定义的。对于当前关系 r 的任意两个元组,如果 X 值相同,则要求 Y 值也相同,即有一个 X 值就有 Y 值与之对应,或者说 Y 值由 X 值决定。因而这种依赖称为函数依赖。

【例 5.2】 有一个关于学生选课、教师任课的关系模式:

$R(SNO, SNAME, CNO, GRADE, CNAME, TNAME, TAGE)$

属性分别表示学生学号、姓名、选修课程的课程号、成绩、课程名、任课教师姓名和年龄等意义。

如果规定,每个学号只能有一个学生姓名,每个课程号只能决定一门课程,那么可写成下列 FD 形式:

$SNO \rightarrow SNAME$

$CNO \rightarrow CNAME$

每个学生每学一门课程,有一个成绩,那么可写出下列 FD:

$(SNO, CNO) \rightarrow GRADE$

还可以写出其他一些 FD:

$CNO \rightarrow (CNAME, TNAME, TAGE)$

$TNAME \rightarrow TAGE$

【例 5.3】 设关系模式 $R(ABCD)$,在 R 的关系中,属性值间有这样的联系: A 值与 B 值有一对多联系,即每个 A 值有多个 B 值与之联系,而每个 B 值只有一个 A 值与之联系; C 值与 D 值之间有一对一联系,即每个 C 值只有一个 D 值与之联系,每个 D 值只有一个 C 值与之联系。试根据这些规则写出相应的函数依赖。

【解】 从 A 值与 B 值有一对多联系,可写出函数依赖 $B \rightarrow A$ 。

从 C 值与 D 值有一对一联系,可写出两个函数依赖 $C \rightarrow D$ 和 $D \rightarrow C$ 。

5.2.2 FD 的逻辑蕴涵

由于函数依赖是用命题形式定义的,因此函数依赖之间存在着逻辑蕴涵的关系。比如 $A \rightarrow B$ 和 $B \rightarrow C$ 在关系模式 R 中成立,那么 $A \rightarrow C$ 在 R 中是否成立? 这个问题就是 FD 之间的逻辑蕴涵问题。

定义 5.2 设 F 是在关系模式 $R(U)$ 上成立的函数依赖集, X 和 Y 是属性集 U 的子集。如果从 F 推导出 $X \rightarrow Y$ 也在 $R(U)$ 上成立,那么称 F 逻辑蕴涵 $X \rightarrow Y$,记为 $F \models X \rightarrow Y$ 。

定义 5.3 设 F 是函数依赖集,被 F 逻辑蕴涵的函数依赖全体构成的集合,称为函数依赖集 F 的闭包(Closure),记为 F^+ 。即:

$$F^+ = \{X \rightarrow Y \mid F \models X \rightarrow Y\}$$

5.2.3 FD 的推理规则

从已知的一些 FD,可以推导出另外一些 FD,这就需要一系列推理规则。FD 的推理规则最早出现在 1974 年 W. W. Armstrong 的论文里,这些规则常被叫作“Armstrong 公理”。

下面的推理规则是他人于 1977 年提出的改进形式。

设 U 是关系模式 R 的属性集, F 是 R 上成立的只涉及 U 中属性的函数依赖集。FD 的推理规则有以下三条:

A1(自反性, Reflexivity): 若 $Y \subseteq X \subseteq U$, 则 $X \rightarrow Y$ 在 R 上成立。

A2(增广性, Augmentation): 若 $X \rightarrow Y$ 在 R 上成立, 且 $Z \subseteq U$, 则 $XZ \rightarrow YZ$ 在 R 上成立。

A3(传递性, Transitivity): 若 $X \rightarrow Y$ 和 $Y \rightarrow Z$ 在 R 上成立, 则 $X \rightarrow Z$ 在 R 上成立。

定理 5.1 FD 推理规则 A1、A2 和 A3 是正确的。也就是, 如果 $X \rightarrow Y$ 是从 F 用推理规则导出, 那么 $X \rightarrow Y$ 在 F^+ 中。

证明: 根据 FD 的定义和使用反证法来证明。

(1) A1 是显然的。因为不可能在一个关系中存在两个元组在 X 上是相等的, 而在 X 的某个子集 Y 上不相等。

(2) 假设 R 的某个关系 r 中存在两个元组 t 和 s 违反 $XZ \rightarrow YZ$, 即 $t[XZ] = s[XZ]$, 但 $t[YZ] \neq s[YZ]$ 。

从 $t[YZ] \neq s[YZ]$ 可知 $t[Y] \neq s[Y]$ 或 $t[Z] \neq s[Z]$ 。如果 $t[Y] \neq s[Y]$, 则与已知的 $X \rightarrow Y$ 矛盾; 如果 $t[Z] \neq s[Z]$, 则与假设的 $t[XZ] = s[XZ]$ 矛盾。因此, 假设不成立, 从而得出 A2 是正确的。

(3) 假设 R 的某个关系 r 中存在两个元组 t 和 s 违反 $X \rightarrow Z$, 即 $t[X] = s[X]$, 但 $t[Z] \neq s[Z]$ 。

如果 $t[Y] \neq s[Y]$, 则与已知的 $X \rightarrow Y$ 矛盾; 如果 $t[Y] = s[Y]$, 则与已知的 $Y \rightarrow Z$ 矛盾。因而 A3 是正确的。

【例 5.4】 已知关系模式 $R(ABC)$, $F = \{A \rightarrow B, B \rightarrow C\}$, 求 F^+ 。

根据 FD 的推理规则, 可推出 F 的 F^+ 有 43 个 FD。

例如, 据规则 A1 可推出 $A \rightarrow \emptyset$ ($\emptyset$ 表示空属性集), $A \rightarrow A$, ... 据已知的 $A \rightarrow B$ 及规则 A2 可推出 $AC \rightarrow BC$, $AB \rightarrow B$, $A \rightarrow AB$, ... 据已知条件及规则 A3 可推出 $A \rightarrow C$ 等。读者可自行推出这 43 个 FD。

定义 5.4 对于 FD $X \rightarrow Y$, 如果 $Y \subseteq X$, 那么称 $X \rightarrow Y$ 是一个“平凡的 FD”; 否则称为“非平凡的 FD”。

正如名称所示, 平凡的 FD 并没有实际意义, 根据规则 A1 就可推出。人们感兴趣的是非平凡的 FD。只有非平凡的 FD 才和“真正的”完整性约束条件相关。

已经证明, $\{A1, A2, A3\}$ 是函数依赖的一个正确的和完备的推理规则集。推理规则的正确性是指“从 FD 集 F 使用推理规则集推出的 FD 必定在 F^+ 中”, 完备性是指“ F^+ 中的 FD 都能从 F 集使用推理规则集导出”。也就是正确性保证了推出的所有 FD 是正确的, 完备性保证了可以推出所有被蕴涵的 FD。这就保证了推导的有效性和可靠性。

除了上述 A1、A2、A3 三条规则外, FD 还有几个实用的推理规则, 这些规则可从上面三条规则导出。这些规则如下:

A4(合并性, Union): $\{X \rightarrow Y, X \rightarrow Z\} \vdash X \rightarrow YZ$ 。

A5(分解性, Decomposition): $\{X \rightarrow Y, Z \subseteq Y\} \vdash X \rightarrow Z$ 。

A6(伪传递性): $\{X \rightarrow Y, WY \rightarrow Z\} \vdash WX \rightarrow Z$ 。

A7(复合性, Composition): $\{X \rightarrow Y, W \rightarrow Z\} \vdash XW \rightarrow YZ$ 。

A8 $\{X \rightarrow Y, W \rightarrow Z\} \vdash X \cup (W - Y) \rightarrow YZ$ 。

其中 A8 是 1992 年由 Darwer 提出的,称为“通用一致性定理”(General Unification Theorem)。

从 A4 和 A5,立即可得到下面的定理。

定理 5.2 如果 $A_1 A_2 \cdots A_n$ 是关系模式 R 的属性集,那么 $X \rightarrow A_1 A_2 \cdots A_n$ 成立的充分必要条件是 $X \rightarrow A_i (i=1,2,\cdots,n)$ 成立。

5.2.4 FD 和关键码的联系

有了 FD 概念后,可以把关键码和 FD 联系起来。实际上,函数依赖是关键码概念的推广。

定义 5.5 设关系模式 R 的属性集是 U, X 是 U 的一个子集。如果 $X \rightarrow U$ 在 R 上成立,那么称 X 是 R 的一个超键。如果 $X \rightarrow U$ 在 R 上成立,但对于 X 的任一真子集 X_1 都有 $X_1 \rightarrow U$ 不成立,那么称 X 是 R 上的一个候选键。本章的键都是指候选键。

【例 5.5】 在学生选课、教师任课的关系模式中:

$R(SNO, SNAME, CNO, GRADE, CNAME, TNAME, TAGE)$

如果规定:每个学生每学一门课只有一个成绩,每个学生只有一个姓名,每个课程号只有一个课程名,每门课程只有一个任课教师,那么根据这些规则,可以知道(SNO,CNO)能函数决定 R 的全部属性,并且是一个候选键。虽然(SNO,SNAME,CNO,TNAME)也能函数决定 R 的全部属性,但相比之下,只能说是一个超键,而不能说是候选键,因为其中含有多余属性。

5.2.5 属性集的闭包

在实际使用中,经常要判断能否从已知的 FD 集 F 推导出 FD $X \rightarrow Y$,那么可先求出 F 的闭包 F^+ ,然后再看 $X \rightarrow Y$ 是否在 F^+ 中。但是从 F 求 F^+ 是一个复杂且困难的问题(NP 完全问题,指数级问题)。下面引入属性集闭包概念,将使判断问题化为多项式级时间问题。

定义 5.6 设 F 是属性集 U 上的 FD 集, X 是 U 的子集,那么(相对于 F)属性集 X 的闭包用 X^+ 表示,它是一个从 F 集使用 FD 推理规则推出的所有满足 $X \rightarrow A$ 的属性 A 的集合:

$$X^+ = \{ \text{属性 } A \mid X \rightarrow A \text{ 在 } F^+ \text{ 中} \}$$

从属性集闭包的定义,立即可得出下面的定理。

定理 5.3 $X \rightarrow Y$ 能用 FD 推理规则推出的充分必要条件是 $Y \subseteq X^+$ 。

从属性集 X 求 X^+ 并不太难,花费的时间与 F 中全部依赖的数目成正比,是一个多项式级时间问题。

【例 5.6】 属性集 U 为 ABCD, FD 集为 $\{A \rightarrow B, B \rightarrow C, D \rightarrow B\}$ 。据属性集闭包的定义,可求出 $A^+ = ABC, (AD)^+ = ABCD, (BD)^+ = BCD$, 等。

5.2.6 FD 集的最小依赖集

如果关系模式 $R(U)$ 上的两个函数依赖集 F 和 G,有 $F^+ = G^+$,则称 F 和 G 是等价的函数依赖集。

函数依赖集 F 中的 FD 很多,应该从 F 中去掉平凡的 FD、无关的 FD、FD 中无关的属性,以求得 F 的最小依赖集 $F_{\min}$ 。形式定义如下:

定义 5.7 设 F 是属性集 U 上的 FD 集。如果 $F_{\min}$ 是 F 的一个最小依赖集,那么 $F_{\min}$ 应满足下列四个条件:

- (1) $F_{\min}^+ = F^+$;
- (2) 每个 FD 的右边都是单属性;
- (3) $F_{\min}$ 中没有冗余的 FD(即 F 中不存在这样的函数依赖 $X \rightarrow Y$,使得 F 与 $F - \{X \rightarrow Y\}$ 等价);
- (4) 每个 FD 的左边没有冗余的属性(即 F 中不存在这样的函数依赖 $X \rightarrow Y$, X 有真子集 W 使得 $F - \{X \rightarrow Y\} \cup \{W \rightarrow Y\}$ 与 F 等价)。

显然,每个函数依赖集至少存在一个最小依赖集,但并不一定唯一。

【例 5.7】 设 F 是关系模式 $R(ABC)$ 的 FD 集, $F = \{A \rightarrow BC, B \rightarrow C, A \rightarrow B, AB \rightarrow C\}$,试求 $F_{\min}$ 。

【解】 ① 先把 F 中的 FD 写成右边是单属性形式:

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

显然多了一个 $A \rightarrow B$,可删去。得 $F = \{A \rightarrow B, A \rightarrow C, B \rightarrow C, AB \rightarrow C\}$ 。

② F 中 $A \rightarrow C$ 是冗余的,可删去。得 $F = \{A \rightarrow B, B \rightarrow C, AB \rightarrow C\}$ 。

③ F 中 $AB \rightarrow C$ 可从 $B \rightarrow C$ 推出,因此 $AB \rightarrow C$ 也可删去。最后得 $F = \{A \rightarrow B, B \rightarrow C\}$,即所求的 $F_{\min}$ 。

5.3 关系模式的分解特性

5.3.1 模式分解问题

定义 5.8 设有关系模式 $R(U)$, $R_1, R_2, \dots, R_k$ 都是 R 的子集(这里把关系模式看成是属性的集合), $U = R_1 \cup R_2 \cup \dots \cup R_k$ 。关系模式 $R_1, R_2, \dots, R_k$ 的集合用 ρ 表示, $\rho = \{R_1, R_2, \dots, R_k\}$ 。用 ρ 代替 R 的过程称为关系模式的分解。这里 ρ 称为 R 的一个分解, ρ 也称为数据库模式。

实际上,这个定义已在图 5.1 中表示。在 5.1 节中已提到, R 分解成 ρ 的目的是为了消除数据冗余和操作异常现象。那么 σ 和 r 是否表示同一个数据库? 如果两者表示不同的内容,那么这个分解就没有什么意义了。

可以从两个角度来考虑分解。

- (1) σ 和 r 是否等价,即是否表示同样的数据。这个问题用“无损分解”特性表示。
- (2) 在模式 R 上有一个 FD 集 F ,在 ρ 的每一个模式 R_i 上有一个 FD 集 F_i ,那么 $\{F_1, F_2, \dots, F_n\}$ 与 F 是否等价。这个问题用“保持依赖”特性表示。

5.3.2 无损分解

【例 5.8】 设有关系模式 $R(ABC)$,分解成 $\rho = \{AB, AC\}$ 。

① 图 5.4(a)是 R 上的一个关系 r ,图 5.4(b)、图 5.4(c)是 r 在模式 AB 和 AC 上的投影 r_1 和 r_2 。显然,此时有 $r_1 \bowtie r_2 = r$ 。也就是在 r 投影、连接以后仍然能恢复成 r ,即未丢失信息,这正是大家所希望的。这种分解称为“无损分解”。

R	A	B	C	r ₁	A	B	r ₂	A	C
	1	1	1		1	1		1	1
	1	2	1		1	2			

(a) (b) (c)

图 5.4 未丢失信息的分解

② 图 5.5(a)是 R 上的一个关系 r,图 5.5(b)和图 5.5(c)是 r 在模式 AB 和 AC 上的投影 r_1 和 r_2 ,图 5.5(d)是 $r_1 \bowtie r_2$ 。此时 $r_1 \bowtie r_2 \neq r$ 。也就是 r 在投影、连接以后比原来 r 的元组还要多(增加了噪声),把原来的信息丢失了。这种分解是我们不希望产生的。这种分解称为“损失分解”。

r	A	B	C	r ₁	A	B	r ₂	A	C	r ₁ ⋈ r ₂	A	B	C
	1	1	4		1	1		1	4		1	1	4
	1	2	3		1	2		1	3		1	1	3
											1	2	4
											1	2	3

(a) (b) (c) (d)

图 5.5 丢失信息的分解

定义 5.9 设 R 是一个关系模式, F 是 R 上的一个 FD 集。R 分解成数据库模式 $\rho = \{R_1, R_2, \dots, R_k\}$ 。如果对 R 中满足 F 的每一个关系 r, 都有

$$r = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \dots \bowtie \pi_{R_k}(r)$$

那么称分解 ρ 相对于 F 是“无损连接分解”(Lossless Join Decomposition), 简称为“无损分解”, 否则称为“损失分解”(Lossy Decomposition)。其中符号 $\pi_{R_i}(r)$ 表示关系 r 在模式 R_i 属性上的投影。r 的投影连接表达式 $\pi_{R_1}(r) \bowtie \dots \bowtie \pi_{R_k}(r)$ 用符号 $m_\rho(r)$ 表示, 即 $m_\rho(r) = \bigbowtie_{i=1}^K \pi_{R_i}(r)$ 。

r 和 $m_\rho(r)$ 之间的联系有下面三个性质。

设 $\rho = \{R_1, R_2, \dots, R_k\}$ 是关系模式 R 的一个分解, r 是 R 的任一关系, $r_i = \pi_{R_i}(r) (1 \leq i \leq k)$, 那么有下列性质:

- (1) $r \subseteq m_\rho(r)$;
- (2) 若 $s = m_\rho(r)$, 则 $\pi_{R_i}(s) = r_i$;
- (3) $m_\rho(m_\rho(r)) = m_\rho(r)$, 这个性质称为幂等性(Idempotent)。

上述三个性质可用图 5.6 表示。

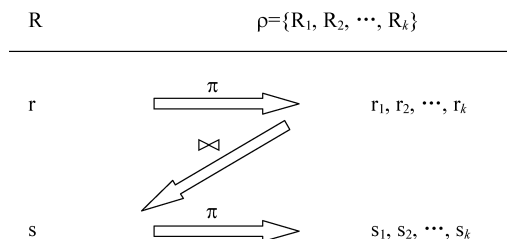


图 5.6 r 的投影连接变换示意图

应注意,上述性质有一个先决条件,即 r 是 R 的一个关系。也就是在先存在 r (泛关系)的情况下,再去谈论分解,这是关系数据库理论中著名的“泛关系假设”(Universal Relation Assumption)。在存在泛关系情况下,泛关系的投影连接变换的示意图可修改为图 5.7。

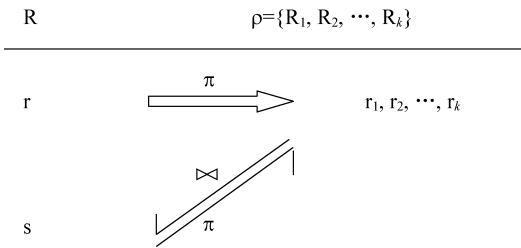


图 5.7 泛关系假设下的示意图

如果谈论模式分解时,先不提泛关系 r 的存在性,而先说存在一个数据库实例 $\sigma(r_1, r_2, \dots, r_k)$,再设 $\bigotimes_{i=1}^K \pi_{r_i}(r) = s$,那么 $\pi_{R_i}(s)$ 就未必与 r_i 相等了。示意图见图 5.8。原因就是 r_i 中可能有“悬挂”元组(Dangling Tuple,破坏泛关系存在的元组)。下面以例 5.9 为例对这种情况进行说明。

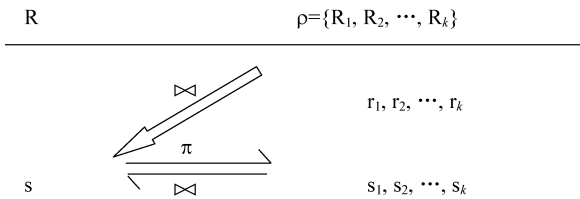


图 5.8 无泛关系假设时的示意图

【例 5.9】 设关系模式 $R(ABC)$ 分解成 $\rho = \{AB, BC\}$ 。图 5.9(a)和图 5.9(b)分别是模式 AB 和 BC 上的值 r_1 和 r_2 ,图 5.9(c)是 $r_1 \bowtie r_2$ 的值。显然 $\pi_{BC}(r_1 \bowtie r_2) \neq r_2$ 。这里 r_2 中元组 (b_2, c_2) 就是一个悬挂元组,由于它的存在,使得 r_1 和 r_2 不存在泛关系 r 。

r_1	A	B
	a_1	b_1

(a) 关系 r_1

r_2	B	C
	b_1	c_1
	b_2	c_2

(b) 关系 r_2

$r_1 \bowtie r_2$	A	B	C
	a_1	b_1	c_1

(c) $r_1 \bowtie r_2$

图 5.9 关系 r_1 和 r_2 不存在泛关系

模式分解能消除数据冗余和操作异常现象,并能使数据库中存储悬挂元组,即存储泛关系中无法存储的信息。但是分解以后,检索操作需要做笛卡儿积或连接操作。这将付出时间代价。一般认为,为了消除冗余和异常现象,对模式进行分解是值得的。

5.3.3 无损分解的测试方法

在把关系模式 R 分解成 ρ 以后,如何测试分解 ρ 是否是无损分解? 已有人提出一个“追踪”(Chase)算法,用于测试一个分解是否是无损分解。

算法 5.1 无损分解的测试。

输入：关系模式 $R=A_1A_2\cdots A_n$ ， F 是 R 上成立的函数依赖集， $\rho=\{R_1,R_2,\cdots,R_k\}$ 是 R 的一个分解。

输出：判断 ρ 相对于 F 是否具有无损分解特性。

方法：(1) 构造一张 k 行 n 列的表格，每列对应一个属性 $A_j(1\leq j\leq n)$ ，每行对应一个模式 $R_i(1\leq i\leq k)$ 。如果 A_j 在 R_i 中，那么在表格的第 i 行第 j 列处填上符号 a_j ，否则填上 b_{ij} 。

(2) 把表格看成模式 R 的一个关系，反复检查 F 中每个 FD 在表格中是否成立，若不成立，则修改表格中的值。修改方法如下：

对于 F 中一个 FD $X\rightarrow Y$ ，如果表格中有两行在 X 值上相等，在 Y 值上不相等，那么把这两行在 Y 值上也改成相等的值。如果 Y 值中有一个是 a_j ，那么另一个也改成 a_j ；如果没有 a_j ，那么用其中一个 b_{ij} 替换另一个值(尽量把下标 ij 改成较小的数)。一直到表格不能修改为止(这个过程称为 chase 过程)。

(3) 若修改的最后一张表格中有一行是全 a ，即 $a_1a_2\cdots a_n$ ，那么称 ρ 相对于 F 是无损分解，否则称损失分解。

【例 5.10】 设关系模式 $R(ABCD)$ ， R 分解成 $\rho=\{AB,BC,CD\}$ 。如果 R 上成立的函数依赖集 $F_1=\{B\rightarrow A,C\rightarrow D\}$ ，那么 ρ 相对于 F_1 是否是无损分解？如果 R 上成立的函数依赖集 $F_2=\{A\rightarrow B,C\rightarrow D\}$ 呢？

【解】

① 相对于 F_1 ，chase 过程的示意图如图 5.10 所示。

	A	B	C	D		A	B	C	D
AB	a_1	a_2	b_{13}	b_{14}	AB	a_1	a_2	b_{13}	b_{14}
BC	b_{21}	a_2	a_3	b_{24}	BC	a_1	a_2	a_3	a_4
CD	b_{31}	b_{32}	a_3	a_4	CD	b_{31}	b_{32}	a_3	a_4

(a) 初始表格

(b) 修改后的表格

图 5.10 相对于 F_1 ，chase 过程的示意图

据 $B\rightarrow A$ ，可把 b_{21} 改成 a_1 ；据 $C\rightarrow D$ ，可把 b_{24} 改成 a_4 。此时第二行已是全 a 行，因此相对于 F_1 ， R 分解成 ρ 是无损分解。

② 相对于 F_2 ，chase 过程的示意图如图 5.11 所示。

据 $C\rightarrow D$ ，可把 b_{24} 改成 a_4 ；据 $A\rightarrow B$ ，不能修改表格。此时表格没有一行是全 a 行，因此相对于 F_2 ， R 分解成 ρ 是损失分解。

	A	B	C	D		A	B	C	D
AB	a_1	a_2	b_{13}	b_{14}	AB	a_1	a_2	b_{13}	b_{14}
BC	b_{21}	a_2	a_3	b_{24}	BC	b_{21}	a_2	a_3	a_4
CD	b_{31}	b_{32}	a_3	a_4	CD	b_{31}	b_{32}	a_3	a_4

(a) 初始表格

(b) 修改后的表格

图 5.11 相对于 F_2 ，chase 过程的示意图

在 chase 过程中,如果把 b 改成 a,则表示从其他模式和已知的 FD 使得该模式可以增加一个属性。如果改成另一个 b_{ij} ,表示模式相应关系中该属性值虽然还没有,但其值应与其他关系中的值相等。

当最后一张表格中存在一行全 a 时,这行表示的模式中可以包含 R 的所有属性,也就回到原来的表格,即 $m_\rho(r)=r$ 。因此,分解是无损分解。

当最后一张表格中不存在全 a 行时,也就是回不到原来的表格,即 $m_\rho(r) \neq r$ 。因此,分解是损失分解。

当 ρ 中只包含两个关系模式时,存在一个较简单的测试定理。

定理 5.4 设 $\rho=\{R_1, R_2\}$ 是关系模式 R 的一个分解, F 是 R 上成立的 FD 集,那么分解 ρ 相对于 F 是无损分解的充分必要条件是:

$$(R_1 \cap R_2) \rightarrow (R_1 - R_2) \quad \text{或} \quad (R_1 \cap R_2) \rightarrow (R_2 - R_1)$$

这个定理的证明可以用算法 5.1 的 chase 过程来实现。

5.3.4 保持函数依赖的分解

分解的另一个特性是在分解的过程中能否保持函数依赖集,如果不能保持 FD,那么数据的语义就会出现混乱。

定义 5.10 设 F 是属性集 U 上的 FD 集, Z 是 U 的子集, F 在 Z 上的投影用 $\pi_Z(F)$ 表示, 定义为:

$$\pi_Z(F) = \{X \rightarrow Y \mid X \rightarrow Y \in F^+, \text{ 且 } XY \subseteq Z\}$$

定义 5.11 设 $\rho=\{R_1, R_2, \dots, R_k\}$ 是 R 的一个分解, F 是 R 上的 FD 集,如果有 $\bigcup_{i=1}^K \pi_{R_i}(F) \models F$, 那么称分解 ρ 保持函数依赖集 F。

从定义 5.10 可知 $F \models \bigcup_{i=1}^K \pi_{R_i}(F)$, 从定义 5.11 可知 $\bigcup_{i=1}^K \pi_{R_i}(F) \models F$, 因此, 在分解 ρ 保持函数依赖情况下有 $\left(\bigcup_{i=1}^K \pi_{R_i}(F)\right)^+ = F^+$ 。

根据定义 5.11, 测试一个分解是否保持 FD, 比较可行的方法是逐步验证 F 中每个 FD 是否被 $\bigcup_{i=1}^K \pi_{R_i}(F)$ 逻辑蕴涵。

如果 F 的投影不蕴涵 F, 而用 $\rho=\{R_1, R_2, \dots, R_k\}$ 表达 R, 很可能会找到一个数据库实例 σ 满足投影后的依赖, 但不满足 F。对 σ 的更新也有可能使 r 违反 FD。下面的例子说明了这种情况。

【例 5.11】 设关系模式 R(WNO, WS, WG) 的属性分别表示职工的工号、工资级别和工资数目。FD 有 $WNO \rightarrow WS, WS \rightarrow WG$ 。

R 分解成 $\rho=\{R_1, R_2\}$, 其中 $R_1=\{WNO, WS\}, R_2=\{WNO, WG\}$, 可以验证这个分解是无损分解。

R_1 上的 FD 是 $WNO \rightarrow WS$, R_2 上的 FD 是 $WNO \rightarrow WG$ 。但从这两个 FD 推导不出在 R 上成立的 FD $WS \rightarrow WG$, 因此分解 ρ 把 $WS \rightarrow WG$ 丢失了, 即 ρ 不保持 F。图 5.12(a) 和图 5.12(b) 是两个关系 r_1 和 r_2 , 图 5.12(c) 是 $r_1 \bowtie r_2$ 。 r_1 和 r_2 分别满足 $\pi_{R_1}(F)$ 和 $\pi_{R_2}(F)$ 。但 $r_1 \bowtie r_2$ 违反了 $WS \rightarrow WG$ 。

WNO	WS	WNO	WG	WNO	WS	WG
W1	8级	W1	2000	W1	8级	2000
W2	6级	W2	1600	W2	6级	1600
W3	6级	W3	1400	W3	6级	1400

(a) 关系 r_1 (b) 关系 r_2 (c) $r_1 \bowtie r_2$

图 5.12 丢失 FD 的分解

如果某个分解能保持 FD 集,那么在数据输入或更新时,只要每个关系模式本身的 FD 约束被满足,就可以确保整个数据库中数据的语义完整性不受破坏。显然这是一种良好的特性。

5.3.5 本节小结

本节讨论的关系模式分解的两个特性实际上涉及两个数据库模式的等价问题,这种等价包括数据等价和依赖等价两个方面。数据等价是指两个数据库实例应表示同样的信息内容,用“无损分解”衡量。如果是无损分解,那么对泛关系反复的投影和连接都不会丢失信息。依赖等价是指两个数据库模式应有相同的依赖集闭包。在依赖集闭包相等的情况下,数据的语义是不会出差错的。违反数据等价或依赖等价的分解很难说是一个好的模式设计。

但是要同时达到无损分解和保持 FD 的分解也不是一件容易的事情,需要认真对待。下面的例子表示关系模式 $R(ABC)$ 在不同函数依赖集上即使对同样的分解也会产生不同的结果。

【例 5.12】 设关系模式 $R(ABC)$, $\rho = \{AB, AC\}$ 是 R 的一个分解。试分析分别在 $F_1 = \{A \rightarrow B\}$, $F_2 = \{A \rightarrow C, B \rightarrow C\}$, $F_3 = \{B \rightarrow A\}$, $F_4 = \{C \rightarrow B, B \rightarrow A\}$ 情况下, ρ 是否具有无损分解和保持 FD 的分解特性。

【解】

- ① 相对于 $F_1 = \{A \rightarrow B\}$, 分解 ρ 是无损分解且保持 FD 的分解。
- ② 相对于 $F_2 = \{A \rightarrow C, B \rightarrow C\}$, 分解 ρ 是无损分解, 但不保持 FD 集。因为 $B \rightarrow C$ 丢失了。
- ③ 相对于 $F_3 = \{B \rightarrow A\}$, 分解 ρ 是损失分解但保持 FD 集的分解。
- ④ 相对于 $F_4 = \{C \rightarrow B, B \rightarrow A\}$, 分解 ρ 是损失分解且不保持 FD 集的分解。

从上例可以看出分解的无损分解与保持 FD 的分解两个特性之间没有必然的联系。

5.4 关系模式的范式

关系模式的好与坏,用什么标准衡量? 这个标准就是模式的范式(Normal Forms, NF)。范式有 1NF、2NF、3NF、BCNF 等多种。范式的种类与数据依赖有着直接的联系。

5.4.1 第一范式

定义 5.12 如果关系模式 R 的每个关系 r 的属性值都是不可分的原子值,那么称 R 是

第一范式(First Normal Form, 1NF)的模式。

满足 1NF 的关系称为规范化的关系,否则称为非规范化的关系。关系数据库研究的关系都是规范化的关系。例如关系模式 $R(\text{NAME}, \text{ADDRESS}, \text{PHONE})$, 如果一个人有两个电话号码(PHONE), 那么在关系中至少要出现两个元组, 以便存储这两个号码。

1NF 是关系模式应具备的最起码的条件。

5.4.2 第二范式

即使关系模式是 1NF, 但很可能具有不受欢迎的冗余和异常现象。因此需把关系模式做进一步的规范化。

如果关系模式中存在局部依赖, 就不是一个好的模式, 需要把关系模式分解, 以排除局部依赖, 使模式达到 2NF 的标准。具体定义如下所述。

定义 5.13 对于 $\text{FD } W \rightarrow A$, 如果存在 $X \subset W$ 有 $X \rightarrow A$ 成立, 那么称 $W \rightarrow A$ 是局部依赖 (A 局部依赖于 W); 否则称 $W \rightarrow A$ 是完全依赖。

完全依赖也称为“左部不可约依赖”。

定义 5.14 如果 A 是关系模式 R 的候选键中的属性, 那么称 A 是 R 的主属性; 否则称 A 是 R 的非主属性。

定义 5.15 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, 那么称 R 是第二范式(2NF)的模式。如果数据库模式中每个关系模式都是 2NF, 则称数据库模式为 2NF 的数据库模式。

【例 5.13】 设关系模式 $R(\text{SNO}, \text{CNO}, \text{GRADE}, \text{TNAME}, \text{TADDR})$ 的属性分别表示学生学号、选修课程的编号、成绩、任课教师姓名和教师地址等意义。 (SNO, CNO) 是 R 的候选键。

R 上有两个 FD: $(\text{SNO}, \text{CNO}) \rightarrow (\text{TNAME}, \text{TADDR})$ 和 $\text{CNO} \rightarrow (\text{TNAME}, \text{TADDR})$, 因此前一个 FD 是局部依赖, R 不是 2NF 模式。此时 R 的关系就会出现冗余和异常现象。例如某一门课程有 100 个学生选修, 那么在关系中就会存在 100 个元组, 因而教师的姓名和地址就会重复 100 次。

如果把 R 分解成 $R_1(\text{CNO}, \text{TNAME}, \text{TADDR})$ 和 $R_2(\text{SNO}, \text{CNO}, \text{GRADE})$ 后, 局部依赖 $(\text{SNO}, \text{CNO}) \rightarrow (\text{TNAME}, \text{TADDR})$ 就消失了。 R_1 和 R_2 都是 2NF 模式。

在关系模式 R 中消除非主属性对候选键的局部依赖的方法如下所述。

算法 5.2 分解成 2NF 模式集的算法。

设关系模式 $R(\text{WXYZ})$, 主键是 WX , R 上还存在 $\text{FD } X \rightarrow Z$ (也就是 $\text{WX} \rightarrow Z$ 是一个局部依赖)。此时应把 R 分解成两个模式:

$R_1(\text{XZ})$, 主键是 X ;

$R_2(\text{WXY})$, 主键是 WX , 外键是 X (REFERENCES R_1)。

利用外键和主键的连接可以从 R_1 和 R_2 重新得到 R 。

如果 R_1 和 R_2 还不是 2NF, 则重复上述过程, 一直到数据库模式中每一个关系模式都是 2NF 为止。

5.4.3 第三范式

定义 5.16 如果 $X \rightarrow Y, Y \rightarrow A$, 且 $Y \not\rightarrow X$ 和 $A \notin Y$, 那么称 $X \rightarrow A$ 是传递依赖 (A 传递依

赖于 X)。

定义 5.17 如果关系模式 R 是 1NF,且每个非主属性都不传递依赖于 R 的候选键,那么称 R 是第三范式(3NF)的模式。如果数据库模式中每个关系模式都是 3NF,则称其为 3NF 的数据库模式。

【例 5.14】 例 5.13 中的 $R_1(\text{CNO}, \text{TNAME}, \text{TADDR})$ 是 2NF 模式。如果 R_1 中存在 FD: $\text{CNO} \rightarrow \text{TNAME}$ 和 $\text{TNAME} \rightarrow \text{TADDR}$,那么 $\text{CNO} \rightarrow \text{TADDR}$ 就是一个传递依赖,即 R_1 不是 3NF 模式。此时 R_1 的关系中也会出现冗余和异常操作。例如一个教师开设五门课程,那么关系中就会出现五个元组,教师的地址就会重复五次。

如果把 R_1 分解成 $R_{11}(\text{TNAME}, \text{TADDR})$ 和 $R_{12}(\text{CNO}, \text{TNAME})$ 后, $\text{CNO} \rightarrow \text{TADDR}$ 就不会出现在 R_{11} 和 R_{12} 中。这样 R_{11} 和 R_{12} 都是 3NF 模式。

在关系模式 R 中消除非主属性对候选键的传递依赖的方法如下所述。

算法 5.3 分解成 3NF 模式集的算法。

设关系模式 $R(\text{WXY})$,主键是 W, R 上还存在 FD $X \rightarrow Y$ 。这样 $W \rightarrow Y$ 就是一个传递依赖。此时应把 R 分解成两个模式:

$R_1(\text{XY})$,主键是 X;

$R_2(\text{WX})$,主键是 W,外键是 X(REFERENCES R_1)。

利用外键和主键相匹配机制, R_1 和 R_2 通过连接可以重新得到 R。

如果 R_1 和 R_2 还不是 3NF,则重复上述过程,一直到数据库模式中每一个关系模式都是 3NF 为止。

从定义 5.13 和定义 5.16 可以知道,局部依赖的存在必定蕴涵着传递依赖的存在。也就是,如果 R 是 3NF 模式,那么 R 也是 2NF 模式。

局部依赖和传递依赖是模式产生冗余和异常的两个重要原因。由于 3NF 模式中不存在非主属性对候选键的局部依赖和传递依赖,因此消除了很大一部分存储异常,具有较好的性能。而对于非 3NF 的 1NF、2NF,甚至非 1NF 的关系模式,由于它们性能上的弱点,一般不宜作为数据库模式,通常需要将它们变换成 3NF 或更高级的范式,这种变换过程,称为“关系的规范化处理”。

3NF 有个等价的定义,如下所述。

定义 5.18 设 F 是关系模式 R 的 FD 集,如果对 F 中每个非平凡的 FD $X \rightarrow Y$,都有 X 是 R 的超键,或者 Y 的每个属性都是主属性,那么称 R 是 3NF 的模式。

这个定义表明,如果非平凡的 FD $X \rightarrow Y$,X 不包含超键(并且 Y 不是主属性),那么 Y 必传递依赖于候选键,因此 R 不是 3NF 模式。

5.4.4 BCNF

在 3NF 模式中,并未排除主属性对候选键的传递依赖,仍有可能有冗余和异常现象。因此有必要提出更高一级的范式。Boyce 和 Codd 分别在 1972 年提出了这种范式,故称为“BCNF”

定义 5.19 如果关系模式 R 是 1NF,且每个属性都不传递依赖于 R 的候选键,那么称 R 是 BCNF 的模式。如果数据库模式中每个关系模式都是 BCNF,则称为 BCNF 的数据库模式。

【例 5.15】 设关系模式 $R(BNO, BNAME, AUTHOR)$ 的属性分别表示书号、书名和作者名。如果规定,每个书号只有一个书名,但不同书号可以有相同书名;每本书可以有多个作者,但每个作者参与编著的书名应该互不相同。这样的规定可以用下列两个 FD 表示:

$$BNO \rightarrow BNAME \text{ 和 } (AUTHOR, BNAME) \rightarrow BNO$$

R 的关键码为 $(BNAME, AUTHOR)$ 或 $(BNO, AUTHOR)$, 因而模式 R 的属性都是主属性, R 是 3NF 模式。但从上述两个 FD 可知, 属性 $BNAME$ 传递依赖于关键码 $(AUTHOR, BNAME)$, 因此 R 不是 BCNF 模式。例如一本书由多个作者编写时, 其书名与书号间的联系在关系中将多次出现, 带来冗余和操作异常现象。

如果把 R 分解成 $R_1(BNO, BNAME)$ 和 $R_2(BNO, AUTHOR)$, 能解决上述问题, 且 R_1 和 R_2 都是 BCNF。但有可能引起新的问题, 例如这个分解把 $(AUTHOR, BNAME) \rightarrow BNO$ 丢失了, 数据语义将会引起新的矛盾。

从定义 5.17 和定义 5.19 可以知道, 如果 R 是 BCNF 模式, 那么 R 也是 3NF 模式。

BCNF 也有个等价的定义, 如下所述。

定义 5.20 设 F 是关系模式 R 的 FD 集, 如果对 F 中每个非平凡的 FD $X \rightarrow Y$, 都有 X 是 R 的超键, 那么称 R 是 BCNF 的模式。

这个定义表明, 如果非平凡的 FD $X \rightarrow Y$ 中 X 不包含超键, 那么 Y 必定传递依赖于候选键, 因此 R 不是 BCNF 模式。

5.4.5 分解成 BCNF 模式集的方法

这个方法的基本思路如下所述。

对于关系模式 R 的分解 ρ (初始时 $\rho = \{R\}$), 如果 ρ 中有一个关系模式 R_i 相对于 $\pi_{R_i}(F)$ 不是 BCNF, 据定义 5.20 可知, R_i 中存在一个非平凡 FD $X \rightarrow Y$, 有 X 不包含超键。此时把 R_i 分解成 XY 和 $R_i - Y$ 两个模式。重复上述过程, 一直到 ρ 中每一个模式都是 BCNF。

上述方法能保证把 R 无损分解成 ρ , 但不一定能保证 ρ 能保持 FD (例 5.15 说明了这种情况)。

5.4.6 分解成 3NF 模式集的方法

这个方法的基本思路如下所述。

(1) 对于关系模式 R 和 R 上成立的 FD 集 F , 先求出 F 的最小依赖集, 然后再把最小依赖集中那些左部相同的 FD 用合并性合并起来。

(2) 对最小依赖集中每个 FD $X \rightarrow Y$ 去构成一个模式 XY 。

(3) 在构成的模式集中, 如果每个模式都不包含 R 的候选键, 那么把候选键作为一个模式放入模式集中。

这样得到的模式集是关系模式 R 的一个分解, 并且这个分解既是无损分解, 又能保持 FD, 并且每个模式都是 3NF。

【例 5.16】 设关系模式 $R(ABCDE)$, R 的最小依赖集为 $\{A \rightarrow B, C \rightarrow D\}$ 。从依赖集可知 R 的候选键为 ACE 。

先根据最小依赖集, 可知 $\rho = \{AB, CD\}$ 。然后再加入由候选键组成的模式 ACE 。因此

最后结果 $\rho = \{AB, CD, ACE\}$ 是一个 3NF 模式集, R 相对于该依赖集是无损分解且保持 FD。

5.4.7 模式设计方法的原则

至此,读者对关系模式的分解有了较全面的了解。关系模式 R 相对于函数依赖集 F 分解成数据库模式 $\rho = \{R_1, R_2, \dots, R_k\}$, 一般应具有三个特性:

- (1) ρ 是 BCNF 模式集, 或 3NF 模式集;
- (2) 无损分解, 即对于 R 上任何满足 F 的泛关系应满足 $r = m_\rho(r)$;
- (3) 保持函数依赖集 F, 即 $\left(\bigcup_{i=1}^k \pi_{R_i}(F)\right) \models F$ 。

数据库设计者在进行关系数据库设计时, 应作权衡, 尽可能地使数据库模式保持最好的特性。一般尽可能设计成 BCNF 模式集。如果设计成 BCNF 模式集时达不到保持 FD 的特点, 那么只能降低要求, 设计成 3NF 模式集, 以求达到保持 FD 和无损分解的特点。

模式分解并不单指把泛关系模式分解成数据库模式, 也可以把数据库模式转换成另一个数据库模式, 分解和转换的关键是要“等价”地分解。一个好的模式设计方法应符合三条原则: 表达性、分离性和最小冗余性。

表达性涉及两个数据库模式的等价问题, 即数据等价和语义等价, 分别用无损分解和保持依赖来衡量。

分离性是指在关系中只存储有直接联系的属性值, 而不要把有间接联系的属性值放在一张表中。应该把有间接联系的属性值放在不同的表中。实际上“分离”就是清除冗余和异常现象。如能达到这个目的, 就分离, 否则将更糟。分离的基准是一系列范式。在分解成 BCNF 模式集时, 分离与依赖等价有时是不兼容的。

最小冗余性要求分解后的模式个数和模式中属性总数应最少。目的是节省存储空间, 提高操作效率, 消除不必要的冗余。但要注意, 实际使用时并不一定要达到最小冗余, 有时带点冗余对提高查询速度是有好处的。

5.5 节讨论另外两种对数据库设计有一定用处的数据依赖和范式, 但基本的模式设计思想还是这些。

* 5.5 模式的进一步规范

前面提到的 FD 有效地表达了属性值之间的多对一联系, 它是现实世界中广泛存在也是最基本的一种数据依赖。但 FD 还不足以刻画现实世界中属性值之间的一对多联系, 本节介绍的多值依赖和联系依赖能刻画一部分一对多联系。与这两种依赖有关的范式是 4NF 和 5NF。

5.5.1 多值依赖的定义

【例 5.17】 模式 $R(DNAME, TNAME, SNAME)$ 的属性分别表示学校的系名、教师名和学生名, 存储每个系里的教师和学生。例如系 d1 有两个教师 t1 和 t2, 三个学生 s1、s2 和 s3, 那么在关系中就要出现六个元组, 否则数据就不完整如图 5.13 所示。显然, 这里存在着

数据冗余和操作异常现象。如果一个系有 20 个教师和 100 个学生,那么关系中就要出现 2000 个元组。

DNAME	TNAME	SNAME
d1	t1	s1
d1	t1	s2
d1	t1	s3
d1	t2	s1
d1	t2	s2
d1	t2	s3
d2	t3	s4

图 5.13 模式 R 的实例

产生问题的原因是教师和学生没有直接的联系。教师与系有直接的联系,学生与系有直接的联系,但教师与学生之间的联系是间接的联系。把有间接联系的属性放在一个模式中就会产生冗余和异常现象。

在模式 R 中,一个系有很多教师(一对多联系),一个系有很多学生(一对多联系),但教师与学生间没有直接联系。这种属性间的一对多联系称为多值依赖。

多值依赖的形式定义如下。

定义 5.21 设 U 是关系模式 R 的属性集, X 和 Y 是 U 的子集, $Z=U-X-Y$,小写的 xyz 表示属性集 XYZ 的值。对于 R 的关系 r ,在 r 中存在元组 (x, y_1, z_1) 和 (x, y_2, z_2) 时,也就存在元组 (x, y_2, z_1) 和 (x, y_1, z_2) ,那么称多值依赖(Multivalued Dependency, MVD) $X \twoheadrightarrow Y$ 在模式 R 上成立。

在例 5.17 中,模式 R 中的属性值间的一对多联系可用 MVD 表示:

$DNAME \twoheadrightarrow TNAME$ 和 $DNAME \twoheadrightarrow SNAME$

5.5.2 关于 FD 和 MVD 的推理规则集

关于 FD 和 MVD,已经找到一个完备的推理规则集,这个集合有八条规则,三条是关于 FD 的(即 5.2 节中提到的 $A1$ 、 $A2$ 、 $A3$ 规则),三条是关于 MVD 的,还有两条是关于 FD 和 MVD 相互推导的规则。具体如下:

设 U 是关系模式 R 上的属性集, W 、 V 、 X 、 Y 、 Z 为 U 的子集,关于 FD 和 MVD 的推理规则有以下八条。

- $A1$ (FD 的自反性): 若 $Y \subseteq X$,则 $X \rightarrow Y$ 。
 - $A2$ (FD 的增广性): 若 $X \rightarrow Y$,且 $Z \subseteq U$,则 $XZ \rightarrow YZ$ 。
 - $A3$ (FD 的传递性): 若 $X \rightarrow Y$, $Y \rightarrow Z$,则 $X \rightarrow Z$ 。
 - $A4$ (MVD 的补规则,Complementation): 若 $X \twoheadrightarrow Y$,则 $X \twoheadrightarrow U - XY$ 。
 - $A5$ (MVD 的增广性): 若 $X \twoheadrightarrow Y$,且 $V \subseteq W$,则 $WX \twoheadrightarrow VY$ 。
 - $A6$ (MVD 的传递性): 若 $X \twoheadrightarrow Y$, $Y \twoheadrightarrow Z$,则 $X \twoheadrightarrow Z$ 。
 - $A7$ (复制性,Replication): 若 $X \rightarrow Y$,则 $X \twoheadrightarrow Y$ 。
 - $A8$ (接合性,Coalescence Rule): 若 $X \twoheadrightarrow Y$, $W \rightarrow Z$,并且 $Z \subseteq Y$, $W \cap Y = \emptyset$,那么 $X \rightarrow Z$ 。
- 像 $A1 \sim A3$ 的证明一样,可以用反证法证明规则 $A4 \sim A8$ 的正确性。

已经证明,推理规则 A1~A8 对于 FD 和 MVD 是完备的。和 FD 一样,也存在着平凡的 MVD。

定义 5.22 对于属性集 U 上的 MVD $X \twoheadrightarrow Y$,如果 $Y \subseteq X$ 或者 $XY = U$,那么称 $X \twoheadrightarrow Y$ 是一个平凡的 MVD,否则称 $X \twoheadrightarrow Y$ 是一个非平凡的 MVD。

这是因为从 $Y \subseteq X$ 可根据 A1 和 A7 推出 $X \twoheadrightarrow Y$,从 $XY = U$ 可根据 A1、A7 和 A4 推出 $X \twoheadrightarrow Y$ 。

根据规则 A1~A8,还可以推出另外的推理规则。

A9(MVD 的并规则): 若 $X \twoheadrightarrow Y, X \twoheadrightarrow Z$,则 $X \twoheadrightarrow YZ$ 。

A10(MVD 的交规则): 若 $X \twoheadrightarrow Y, X \twoheadrightarrow Z$,则 $X \twoheadrightarrow Y \cap Z$ 。

A11(MVD 的差规则): 若 $X \twoheadrightarrow Y, X \twoheadrightarrow Z$,则 $X \twoheadrightarrow Y - Z, X \twoheadrightarrow Z - Y$ 。

A12(MVD 的伪传递): 若 $X \twoheadrightarrow Y, WY \twoheadrightarrow Z$,则 $WX \twoheadrightarrow Z - WY$ 。

A13(混合伪传递): 若 $X \twoheadrightarrow Y, XY \rightarrow Z$,则 $X \rightarrow Z - Y$ 。

在有 FD 和 MVD 的情况下,也可以用 chase 过程来测试关系模式 R 相对于已知的 FD 和 MVD 集分解成 ρ 是否为无损分解。

另外,也可以用无损分解概念来定义 MVD。

定义 5.23 若 U 是关系模式 R 的属性集, X, Y, Z 是 U 的一个分割。若对 R 的每一个关系 r ,都有 $r = \pi_{XY}(r) \bowtie \pi_{XZ}(r)$,则称 MVD $X \twoheadrightarrow Y$ 在 $R(U)$ 上成立。

这个定义说明,如果一个模式可以无损分解成两个模式,那么蕴涵着一个多值依赖。

5.5.3 第四范式

定义 5.24 设 D 是关系模式 R 上成立的 FD 和 MVD 集合。如果 D 中每个非平凡的 MVD $X \twoheadrightarrow Y$ 的左部 X 都是 R 的超键,那么称 R 是 4NF 的模式。

【例 5.18】 在例 5.17 的模式 $R(DNAME, TNAME, SNAME)$ 中,键是 $(TNAME, SNAME)$,在多值依赖 $DNAME \twoheadrightarrow TNAME$ 和 $DNAME \twoheadrightarrow SNAME$ 的左部未包含键,因此 R 不是 4NF。若把 R 分解成 $R_1(DNAME, TNAME)$ 和 $R_2(DNAME, SNAME)$ 以后,则 R_1 和 R_2 都是 4NF。

从 4NF 的定义可知,是 4NF 的模式肯定是 BCNF 模式。

5.5.4 连接依赖

在定义 5.23 中,MVD 定义为一个模式无损分解为两个模式。类似地,对于一个模式无损分解成 n 个模式的数据依赖,称为连接依赖,形式定义如下所述。

定义 5.25 设 U 是关系模式 R 的属性集, $R_1, R_2, \dots, R_n$ 是 U 的子集,并满足 $U = R_1 \cup R_2 \cup \dots \cup R_n$, $\rho = \{R_1, R_2, \dots, R_n\}$ 是 R 的一个分解。如果对于 R 的每个关系 r 都有 $m_\rho(r) = r$,那么称连接依赖(Join Dependency, JD)在模式 R 上成立,记为 $*$ ($R_1, R_2, \dots, R_n$)。

定义 5.26 如果 $*$ ($R_1, R_2, \dots, R_n$) 中某个 R_i 就是 R ,那么称这个 JD 是平凡的 JD。

【例 5.19】 设关系模式 $R(SPJ)$ 的属性分别表示供应商、零件、项目等含义,表示三者之间的供应联系。如果规定,模式 R 的关系是三个二元投影 (SP, PJ, JS) 的连接,而不是其中任何两个的连接,如图 5.14 所示,那么模式 R 中存在着一个连接依赖 $*$ (SP, PJ, JS)。

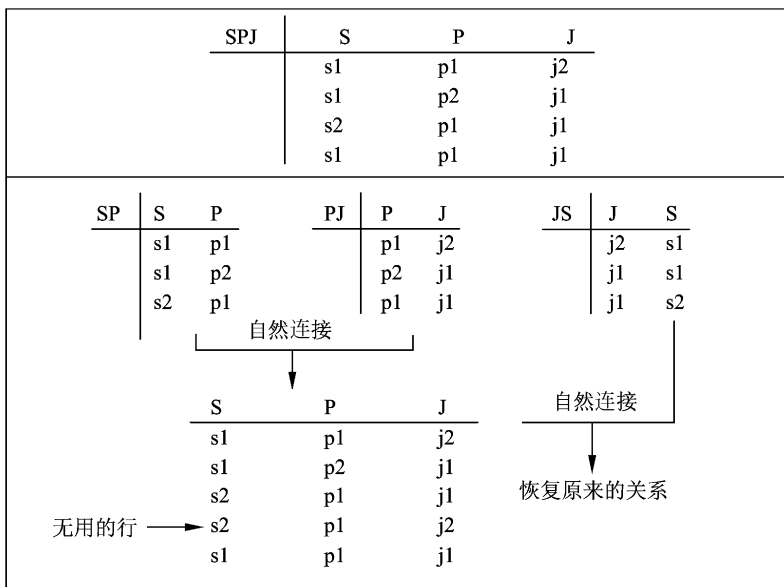


图 5.14 关系 SPJ 是三个二元投影的连接而不是其中任何两个的连接

在模式 R 存在这个连接依赖时,其关系将存在冗余和异常现象。例如在元组插入或删除时就会出现各种异常(见图 5.15)。

SPJ	S	P	J
	s1	p1	j2
	s1	p2	j1

在插入元组(s2, p1, j1)时,

必须再插入元组(s1, p1, j1);

否则将违反连接依赖*(SP, PJ, JS)

SPJ	S	P	J
	s1	p1	j2
	s1	p2	j1
	s2	p1	j1
	s1	p1	j1

- 元组(s2, p1, j1)可直接删除
- 元组(s1, p1, j1)被删除时, 必须再删除其他三个元组中的一个, 才能不违反连接依赖*(SP, PJ, JS)

图 5.15 在 SPJ 中更新问题的例子

5.5.5 第五范式

定义 5.27 如果关系模式 R 的每个 JD 均由 R 的候选键蕴涵,那么称 R 是 5NF 的模式。在有的文献中,5NF 也称为投影连接范式(Project-Join NF,PJNF)。

这里 JD 可由 R 的键蕴涵,是指 JD 可由键推导得到。如果 $JD * (R_1, R_2, \dots, R_n)$ 中某个 R_i 就是 R,那么这个 JD 是平凡的 JD; 如果 JD 中某个 R_i 包含 R 的键,那么这个 JD 可用 chase 方法验证。

【例 5.20】 在例 5.19 中提到的 R(SPJ)中, $*(SP, PJ, JS)$ 是非平凡的 JD,因此 R 不是

5NF。应该把 R 分解成 SP、PJ、JS 三个模式,这个分解是无损分解,并且每个模式都是 5NF,清除了冗余和异常现象。

连接依赖也是现实世界属性间联系的一种抽象,是语义的体现。但是它不像 FD 和 MVD 的语义那么直观,要判断一个模式是否 5NF 也比较困难。

对于 JD,已经找到一些推理规则,但尚未找到完备的推理规则集。可以证明,5NF 的模式也一定是 4NF 的模式。根据 5NF 的定义,可以得出一个模式总是可以无损分解成 5NF 模式集。

不同级别的范式之间的关系如图 5.16 所示。

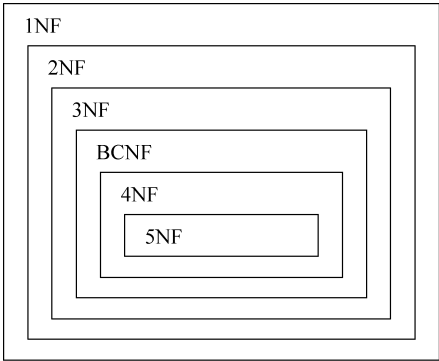


图 5.16 不同级别的范式间的关系

小 结

本章讨论如何设计关系数据库的模式问题。关系模式设计得好与不好,直接影响数据库中数据冗余度、数据一致性问题。要设计好的数据库模式,必须有一定的理论为基础。这就是模式设计理论。

函数依赖是对关系中属性值之间多对一联系描述,也是对关系中值的一种约束。它是对关键码概念的扩充。

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,分别用无损分解和保持依赖两个特征来衡量。前者能保证泛关系在投影连接以后仍能恢复回来,而后者能保证数据在投影或连接中其语义不会发生变化,也就是不会违反 FD 的语义。

范式是衡量模式优劣的标准。范式的级别越高,其数据冗余和操作异常现象就越少。

关系模式的规范化过程实际上是一个“分解”过程:把逻辑上独立的信息放在独立的关系模式中。

习 题 5

1. 解释下列名词。

- | | | | |
|-------|-----------|---------|-------------------|
| 函数依赖 | 函数依赖的逻辑蕴涵 | 平凡的函数依赖 | 函数依赖集 F 的闭包 F^+ |
| 最小依赖集 | 无损分解 | 保持函数依赖 | 1NF |

2NF

3NF

BCNF

4NF

5NF 推理规则的正确性和完备性

多值依赖

连接依赖

2. 设关系模式 R 有 n 个属性, 在模式 R 上可能成立的函数依赖有多少个? 其中平凡的 FD 有多少个? 非平凡的 FD 有多少个?

3. 对函数依赖 $X \rightarrow Y$ 的定义加以扩充, X 和 Y 可以为空属性集, 用 $\emptyset$ 表示, 那么 $X \rightarrow \emptyset$, $\emptyset \rightarrow Y$, $\emptyset \rightarrow \emptyset$ 的含义是什么?

4. 已知关系模式 $R(ABC)$, F 是 R 上成立的 FD 集, $F = \{A \rightarrow B, B \rightarrow C\}$, 试写出 F 的闭包 F^+ 。

5. 设关系模式 $R(ABCD)$, 如果规定, 关系中 B 值与 D 值之间是一对多联系, A 值与 C 值之间是一对一联系。试写出相应的函数依赖。

6. 试举出反例说明下列规则不成立:

(1) $\{A \rightarrow B\} \models \{B \rightarrow A\}$

(2) $\{AB \rightarrow C, A \rightarrow C\} \models \{B \rightarrow C\}$

(3) $\{AB \rightarrow C\} \models \{A \rightarrow C\}$

7. 设关系模式 $R(ABCD)$, F 是 R 上成立的 FD 集, $F = \{A \rightarrow B, C \rightarrow B\}$, 则相对于 F , 试写出关系模式 R 的关键码。并说明理由。

8. 设关系模式 $R(ABCD)$, F 是 R 上成立的 FD 集, $F = \{A \rightarrow B, B \rightarrow C\}$ 。

(1) 试写出属性集 BD 的闭包 $(BD)^+$ 。

(2) 试写出所有左部是 B 的函数依赖(即形为“ $B \rightarrow ?$ ”)。

9. 设关系模式 $R(ABC)$ 分解成 $\rho = \{AB, BC\}$, 如果 R 上的 FD 集 $F = \{A \rightarrow B\}$, 那么这个分解是损失分解。试举出 R 的一个关系 r , 不满足 $m_\rho(r) = r$ 。

10. 试解释数据库“丢失信息”与“未丢失信息”两个概念。“丢失信息”与“丢失数据”有什么区别?

11. 设关系模式 $R(ABC)$, F 是 R 上成立的 FD 集, $F = \{A \rightarrow C, B \rightarrow C\}$, 试分别求 F 在模式 AB 和 AC 上的投影。

12. 设关系模式 $R(ABC)$, F 是 R 上成立的 FD 集, $F = \{B \rightarrow A, C \rightarrow A\}$, $\rho = \{AB, BC\}$ 是 R 上的一个分解, 那么分解 ρ 是否保持 FD 集 F ? 说明理由。

13. 设关系模式 $R(ABC)$, F 是 R 上成立的 FD 集, $F = \{B \rightarrow C, C \rightarrow A\}$, 那么分解 $\rho = \{AB, AC\}$ 相对于 F , 是否无损分解和保持 FD? 说明理由。

14. 设关系模式 $R(ABCD)$, F 是 R 上成立的 FD 集, $F = \{A \rightarrow B, B \rightarrow C, A \rightarrow D, D \rightarrow C\}$, $\rho = \{AB, AC, BD\}$ 是 R 的一个分解。

(1) 相对于 F , ρ 是无损分解吗? 为什么?

(2) 试求 F 在 ρ 的每个模式上的投影。

(3) ρ 保持 F 吗? 为什么?

15. 设关系模式 $R(ABCD)$, R 上的 FD 集 $F = \{A \rightarrow C, D \rightarrow C, BD \rightarrow A\}$, 试说明 $\rho = \{AB, ACD, BCD\}$ 相对于 F 是损失分解的理由。

16. 设关系模式 $R(ABCD)$, F 是 R 上成立的 FD 集, $F = \{AB \rightarrow CD, A \rightarrow D\}$ 。

(1) 试说明 R 不是 2NF 模式的理由。

(2) 试把 R 分解成 2NF 模式集。

17. 设关系模式 $R(ABC)$, F 是 R 上成立的 FD 集, $F = \{C \rightarrow B, B \rightarrow A\}$ 。

(1) 试说明 R 不是 3NF 模式的理由。

(2) 试把 R 分解成 3NF 模式集。

18. 设有一个记录各个球队队员每场比赛进球数的关系模式

$R(\text{队员编号}, \text{比赛场次}, \text{进球数}, \text{球队名}, \text{队长名})$

如果规定每个队员只能属于一个球队, 每个球队只有一个队长, 每个队员每场比赛只有一个进球数。

(1) 试写出关系模式 R 的基本 FD 和关键码。

(2) 说明 R 不是 2NF 模式的理由, 并把 R 分解成 2NF 模式集。

(3) 进而把 R 分解成 3NF 模式集, 并说明理由。

19. 设有关系模式:

$R(\text{职工名}, \text{项目名}, \text{工资}, \text{部门名}, \text{部门经理})$

如果规定每个职工可参加多个项目, 各领一份工资; 每个项目只属于一个部门管理; 每个部门只有一个经理。

(1) 试写出关系模式 R 的基本 FD 和关键码。

(2) 说明 R 不是 2NF 模式的理由, 并把 R 分解成 2NF 模式集。

(3) 进而把 R 分解成 3NF 模式集, 并说明理由。

20. 设关系模式 $R(ABC)$ 上有一个 MVD $A \twoheadrightarrow B$ 。如果已知 R 的当前关系存在三个元组 (ab_1c_1) 、 (ab_2c_2) 和 (ab_3c_3) , 那么这个关系中至少还应该存在哪些元组?

21. 试撰写 2000 字短文, 论述泛关系假设、无损分解和保持依赖间的联系。