

第4章 关系数据库设计理论

本章内容

- 4.1 问题的提出
- 4.2 规范化
- 4.3 数据依赖的公理系统*
- 4.4 小结
- 习题

4.1 问题的提出

这一章我们讨论关系数据库设计理论。即关于：如何使用关系模型设计关系数据库？也就是面对一个现实问题，如何选择一个比较好的关系模式的集合？其中每个关系模式又由哪些属性组成？这就是数据库逻辑设计主要关心的问题。

这一节有两方面的内容：

4.1.1 规范化理论概述

4.1.2 不合理的关系模式存在的问题

[BACK](#)

4.1.1 规范化理论概述

关系数据库设计理论主要包括三个方面的内容：函数依赖、范式（Normal Form）和模式设计。其中函数依赖起着核心作用，是模式分解和模式设计的基础，范式是模式分解的标准。

BACK

4.1.2 不合理的关系模式存在的问题

[例1] 要求设计学生-课程数据库，其关系模式SDC如下：
SDC (SNO, SN, AGE, DEPT, MN, CNO, SCORE)

其中：SNO 表示学生学号

SN 表示学生姓名

AGE 表示学生年龄

DEPT 表示学生所在的系别

MN 表示系主任名

CNO 表示课程号

SCORE 表示成绩。

SNO	SN	AGE	DEPT	MN	CNO	SCORE
S1	赵红	20	计算机	张文斌	C1	90
S1	赵红	20	计算机	张文斌	C2	85
S2	王小明	17	外语	刘伟华	C5	57
S2	王小明	17	外语	刘伟华	C6	80
S2	王小明	17	外语	刘伟华	C7	
S2	王小明	17	外语	刘伟华	C4	70
S3	吴小林	19	信息	刘伟华	C1	75
S3	吴小林	19	信息	刘伟华	C2	70
S3	吴小林	19	信息	刘伟华	C4	85
S4	张涛	22	自动化	钟志强	C1	93

图4.1 关系SDC

在进行数据库的操作时，会出现以下几方面的问题。

(1) 数据冗余。

系名，学生姓名、年龄等等都要重复存储多次

(2) 插入异常。

(SNO, CNO) 是主键。缺少一个都无法插入数据另外，若学

生未选课，同样也不能进行插入操作。

(3) 删除异常。

删去学生数据，导致课程及教师信息丢失。如果某个学生不

再选修某课程，有关该学生的其他信息也随之丢失。

(4) 修改异常。

如果某学生改名，则该学生的所有记录都要逐一修改SN的值；稍有不慎，就有可能漏改某些记录。

我们把关系模式SDC分解为：

学生关系 S (SNO, SNO, AGE, DEPT)

系关系 D (DEPT, MN)

选课关系 SC (SNO, CNO, SCORE)

• S

SNO	SN	AGE	DEPT
S1	赵红	20	计算机
S2	王小明	17	外语
S3	吴小林	19	信息
S4	张涛	22	自动化

- D

DEPT	MN
计算机	张文斌
外语	刘伟华
信息	刘伟华
自动化	钟志强

- SC

SNO	CNO	SCORE
S1	C1	90
S1	C2	85
S2	C5	57
S2	C6	80
S2	C7	
S2	C4	70
S3	C1	75
S3	C2	70
S3	C4	85
S4	C1	93

图4.2 关系SDC经分解后的三关系S、D与SC

分解后的关系模式集是一个好的关系数据库模式。这三个关系模式都不会发生插入异常、删除异常的毛病，数据冗余也得到了尽可能地控制。

但要注意，一个好的关系模式并不是在任何情况下都是最优的，比如查询某个学生选修课程名及所在系的系主任时，要通过连接操作来完成（即由图4.2中的三张表，连接形成图4.1中的一张总表），而连接所需要的系统开销非常大，因此要以实际设计的目标出发进行设计。

[BACK](#)

4.2 规范化

本节将讨论下述内容：首先讨论一个关系属性间不同的依赖情况，讨论如何根据属性间的依赖情况来判定关系是否具有某些不合适的性质。通常按属性间依赖情况来区分关系规范化的程度为第一范式、第二范式、第三范式、BC范式和第四范式等。然后直观地描述如何将具有不合适性质的关系转换为更合适的形式。

[**BACK**](#)

4.2.1 函数依赖

1. 函数依赖

定义4.1 设关系模式 $R(U, F)$ ， U 是属性全集， F 是 U 上的函数依赖集， X 和 Y 是 U 的子集，如果对于 $R(U)$ 的任意一个可能的关系 r ，对于 X 的每一个具体值， Y 都有唯一的具体的值与之对应，则称 X 函数决定 Y ，或 Y 函数依赖于 X ，记 $X \rightarrow Y$ 。我们称 X 为决定因素， Y 为依赖因素。当 Y 不函数依赖于 X 时，记作： $X \not\rightarrow Y$ 。当 $X \rightarrow Y$ 且 $Y \rightarrow X$ 时，则记作： $X \leftrightarrow Y$ 。

对于关系模式SDC:

$U = \{SNO, SN, AGE, DEPT, MN, CNO, SCORE\}$

$F = \{SNO \rightarrow SN, SNO \rightarrow AGE, SNO \rightarrow DEPT, DEPT \rightarrow MN, SNO \rightarrow MN, (SNO, CNO) \rightarrow SCORE\}$

一个SNO有多个SCORE的值与之对应，因此SCORE不能唯一地确定，即SCORE不能函数依赖于SNO，所以有： $SNO \twoheadrightarrow SCORE$,

同样有： $CNO \twoheadrightarrow SCORE$ 。

但是SCORE可以被(SNO, CNO)唯一地确定。所以可表示为： $(SNO, CNO) \rightarrow SCORE$ 。

函数依赖有几点需要说明。

(1) 平凡的函数依赖与非平凡的函数依赖

当属性集Y是属性集X的子集时，则必然存在着函数依赖 $X \rightarrow Y$ ，这种类型的函数依赖称为平凡的函数依赖。如果Y不是X子集，则称 $X \rightarrow Y$ 为非平凡的函数依赖。若不特别声明，我们讨论的都是非平凡的函数依赖。

(2) 函数依赖与属性间的联系类型有关

① 在一个关系模式中，如果属性X与Y有1:1联系时，则存在函数依赖 $X \rightarrow Y, Y \rightarrow X$ ，即 $X \leftrightarrow Y$ 。例如，当学生没有重名时， $SNO \leftrightarrow SN$ 。

② 如果属性X与Y有m:1的联系时，则只存在函数依赖 $X \rightarrow Y$ 。例如，SNO与AGE，DEPT之间均为m:1联系，所以有 $SNO \rightarrow AGE$ ， $SNO \rightarrow DEPT$ 。

③ 如果属性X与Y有m:n的联系时，则X与Y之间不存在任何函数依赖关系。例如，一个学生可以选修多门课程，一门课程又可以为多个学生选修，所以SNO与CNO之间不存在函数依赖关系。

由于函数依赖与属性之间的联系类型有关，所以在确定属性间的函数依赖时，可以从分析属性间的联系入手，便可确定属性间的函数依赖。

(3) 函数依赖的语义范畴的概念

我们只能根据语义来确定一个函数依赖，而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，对于关系模式S，当学生不存在重名的情况下，可以得到：

$SN \rightarrow AGE$ 、 $SN \rightarrow DEPT$

这种函数依赖关系，必须是在没有重名的学生条件下才成立，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，是语义的要求。

(4) 函数依赖关系的存在与时间无关

(5) 函数依赖可以保证关系分解的无损连接性

设 $R(X, Y, Z)$ ， X 、 Y 、 Z 为不相交的属性集合，如果 $X \rightarrow Y$ 或 $X \rightarrow Z$ 则有 $R(X, Y, Z) = R[X, Y] \bowtie R[X, Z]$ ，其中 $R[X, Y]$ 表示关系 R 在属性 (X, Y) 上的投影，即 R 等于两个分别含决定因素 X 的投影关系（分别是 $R[X, Y]$ 与 $R[X, Z]$ ）在 X 上的自然连接，这样便保证了关系 R 分解后不会丢失原有的信息，称作关系分解的无损连接性。

例如，对于关系模式SDC，有

$$\text{SNO} \rightarrow (\text{SNO}, \text{SN}, \text{AGE}, \text{DEPT}, \text{MN}),$$
$$\text{SDC}(\text{SNO}, \text{SN}, \text{AGE}, \text{DEPT}, \text{MN}, \text{CNO}, \text{SCORE}) = \text{SDC}[\text{SNO}, \text{SN}, \text{AGE}, \text{DEPT}, \text{MN}] \bowtie \text{SDC}[\text{SNO}, \text{CNO}, \text{SCORE}],$$

也就是说，用其投影在SNO上的自然连接可复原关系模式SDC。这一性质非常重要，在后面的关系规范化中要用到。

2. 函数依赖的基本性质

(1) 投影性

根据平凡的函数依赖的定义可知，一组属性函数决定它的所以子集。例如，在关系SDC中， $(SNO, CNO) \rightarrow SNO$ 和 $(SNO, CNO) \rightarrow CNO$ 。

说明：投影性产生的是平凡的函数依赖，需要时也能使用的。

(2) 扩张性

若 $X \rightarrow Y$ 且 $W \rightarrow Z$ ，则 $(X, W) \rightarrow (Y, Z)$ 。例如， $SNO \rightarrow (SN, AGE)$ ， $DEPT \rightarrow MN$ ，则有 $(SNO, DEPT) \rightarrow (SN, AGE, MN)$ 。

说明：扩张性实现了两函数依赖决定因素与被决定因素的分别合并作用。

(3) 合并性

若 $X \rightarrow Y$ 且 $X \rightarrow Z$ 则必有 $X \rightarrow (Y, Z)$ 。例如，在关系SDC中， $SNO \rightarrow (SN, AGE)$ ， $SNO \rightarrow DEPT$ ，则有 $SNO \rightarrow (SN, AGE, DEPT)$ 。

说明：决定因素相同的两函数依赖被决定因素的可以合并。

(4) 分解性

若 $X \rightarrow (Y, Z)$ ，则 $X \rightarrow Y$ 且 $X \rightarrow Z$ 。很显然，分解性为合并性的逆过程。

说明：决定因素能决定全部，当然也能决定全部中的部分。

由合并性和分解性，很容易得到以下事实：

$X \rightarrow A_1, A_2, \dots, A_n$ 成立的充分必要条件是
 $X \rightarrow A_i$ ($i=1, 2, \dots, n$) 成立。

3. 完全/部分函数依赖和传递/非传递函数依赖

定义4.2 设有关系模式 $R(U)$ ， U 是属性全集， X 和 Y 是 U 的子集， $X \rightarrow Y$ ，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（Full Functional Dependency），记作 $X \xrightarrow{f} Y$ 。如果对 X 的某个真子集 X' ，有 $X' \rightarrow Y$ ，则称 Y 对 X 部分函数依赖（Partial Functional Dependency），记作 $X \xrightarrow{p} Y$ 。

例如，在关系模式SDC中，因为 $SNO \twoheadrightarrow SCORE$ ，且 $CNO \twoheadrightarrow SCORE$ ，所以有： $(SNO, CNO) \xrightarrow{f} SCORE$ 。而因为有 $SNO \rightarrow AGE$ ，所以有 $(SNO, CNO) \xrightarrow{p} AGE$ 。

定义4.3 设有关系模式 $R(U)$ ， U 是属性全集， X, Y, Z 是 U 的子集，若 $X \rightarrow Y$ ，但 $Y \not\rightarrow X$ ，而 $Y \rightarrow Z$ ($Y \not\subseteq X$)，则称 Z 对 X 传递函数依赖 (Transitive Functional Dependency)，记作： $X \xrightarrow{t} Z$ 。

注意：如果有 $Y \rightarrow X$ ，则 $X \leftrightarrow Y$ ，这时称 Z 对 X 直接函数依赖，而不是传递函数依赖。

4.2.2 码

在第2章中已给出有关码的概念。这里用函数依赖的概念来定义码。

定义4.4 设 K 为 $R(U, F)$ 中的属性或属性集合，若 $K \xrightarrow{f} U$ 则 K 为 R 的**候选码**（或**候选关键字**或**候选键**）（Candidate key）。若候选码多于一个，则选定其中的一个为**主码**（或称**主键**，Primary key）。

包含在任何一个候选码中的属性，叫做**主属性** (Prime attribute)。不包含在任何一个候选码中的属性称为**非主属性** (Nonprime attribute) 或非码属性 (Non-key attribute)。在最简单的情况，单个属性是码。最极端的情况，整个属性组U是码，称为全码 (All-key)。如在关系模式S (SNO, DEPT, AGE) 中SNO是码，而在关系模式SC (SNO, CNO, SCORE) 中属性组合 (SNO, CNO) 是码。

下面举个全码的例子。

关系模式TCS (T, C, S) , 属性T表示教师, C表示课程, S表示学生。一个教师可以讲授多门课程, 一门课程可有多个教师讲授, 同样一个学生可以选听多门课程, 一门课程可被多个学生选听。教师T, 课程C, 学生S之间是多对多关系, 单个属性T、C、S或两个属性组合 (T, C) 、 (T, S) 、 (C, S) 等均不能完全决定整个属性组U, 只有 $(T, C, S) \rightarrow U$, 所以这个关系模式的码为 (T, C, S) , 即All-key。
全码关系只有主属性。

找出已知关系模式R (U, F) 的所有候选键的方法:

1、查看函数依赖集F中的每个形如 $X_i \rightarrow Y_i$ (要确认每个函数依赖 $X_i \rightarrow Y_i$ 均为非平凡的完全的函数依赖) 的 ($i=1, \dots, n$) 函数依赖关系。看哪些属性在所有 Y_i ($i=1, \dots, n$) 中没有出现过,

设没出现过的属性集为P ($P=U-Y_1-Y_2-\dots-Y_n$)。

则当 $P=\phi$ (表示空集) 时, 转4

当 $P \neq \phi$ 时, 转2。

2、根据候选键的定义, 候选键中应必含P (因为没有其它属性能决定P)。考察P:

若有 $P \xrightarrow{f} U$ 成立, 则P为候选键, 并且候选键只有一个P, 转5结束

若 $P \xrightarrow{f} U$ 不成立, 则转3。

3、P可以分别与{U-P}中的每一个属性合并，形成 P_1, P_2, \dots, P_m 。再分别判断 $P_j \xrightarrow{f} U$ ($j=1, \dots, m$)是否成立？能成立则找到了一个候选键，没有则放弃。合并一个属性若不能找到或不能找全候选键，可进一步考虑P与{U-P}中的两个（或三个，四个，……）属性的所有组合分别进行合并，继续判断分别合并后的各属性组对U的完全函数决定情况……；如此直到找出R的所有候选键为止。转5结束。（需要提醒的是：如若属性组K已有 $K \xrightarrow{f} U$ ，则完全不必去考察含K的其它属性组合了，显然它们都不可能再是候选键了）。

4、若 $P = \phi$ ，则可以先考察 $X_i \rightarrow Y_i$ ($i=1, \dots, n$) 中的单个 X_i ，判断是否有 $X_i \xrightarrow{f} U$ ？若成立则 X_i 为候选键。剩下不是候选键的 X_i ，可以考察它们两个或多个的组合，查看其是否能完全函数决定 U ，从而找出其它还有可能的候选键。转5结束。

5、本方法结束。

定义4.5

关系模式R中属性或属性组X并非R的主码，但X是另外一个关系模式S的主码，则称X是R的**外部码**或**外部关系键**（Foreign Key），也称**外码**。

如在SC（SNO，CNO，SCORE）中，单SNO不是主码，但SNO是关系模式S（SNO，SN，SEX，AGE，DEPT）的主码，则SNO是SC的外码。

主码与外码提供了一个表示关系间联系的手段。如关系模式S与SC的联系就是通过SNO这个在S中是主码又在SC中是外码的属性来体现的。

4.2.3 范式

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的标准或准则称为**范式 (Normal Form)**。满足最低要求的叫**第一范式,简称1NF**。在第一范式中满足进一步要求的为**第二范式(2NF)**,其余以此类推。**R**为第几范式就可以写成 **$R \in xNF$** (**x**表示某范式名)。

各个范式之间的集合关系可以表示为：

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

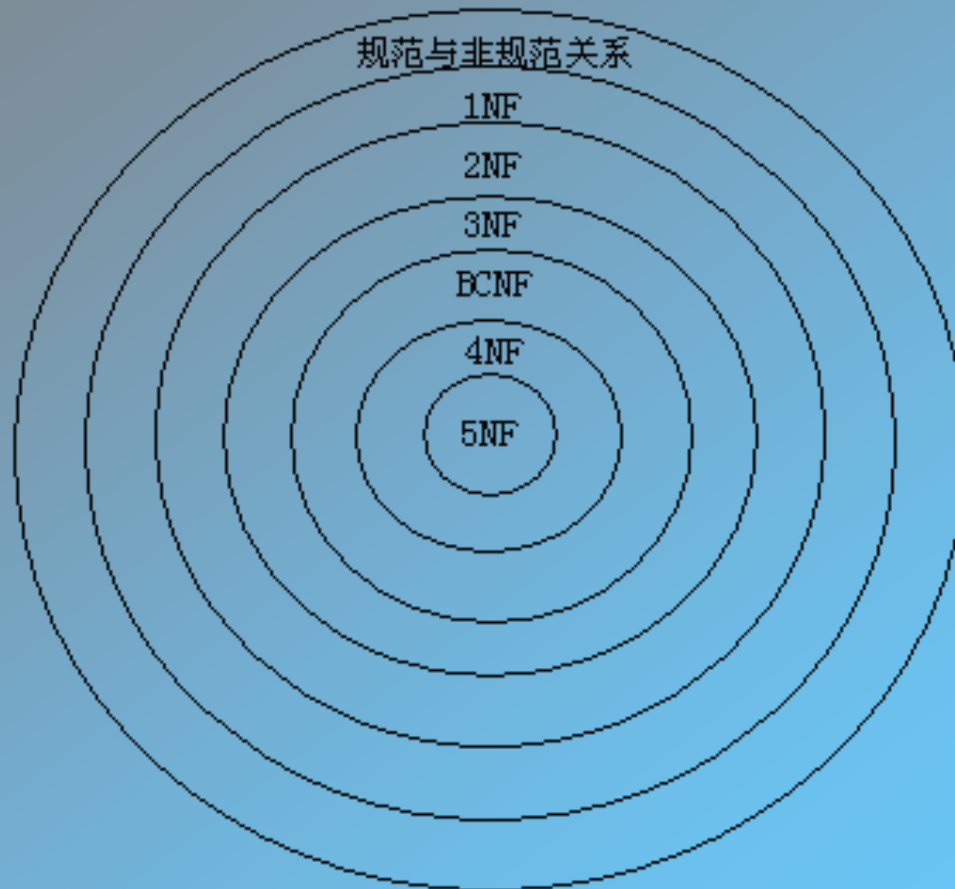


图4.3 各范式之间的关系

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**规范化**。

4.2.4 第一范式

第一范式 (First Normal Form) 是最基本的规范化形式, 即关系中每个属性都是不可再分的简单项。

定义4.6 如果关系模式R所有的属性均为简单属性, 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 $R \in 1NF$ 。

在关系数据库系统中只讨论规范化的关系, 凡是非规范化的关系模式必须转化成规范化的关系。在非规范化的关系中去掉组合项就能转化成规范化的关系。每个规范化的关系都是属于1NF。

[例2] 如职工号，姓名，电话号码组成一个表（一个人可能有一个办公室电话和一个家里电话号码）规范成为1NF有四种方法：

一是重复存储职工号和姓名。这样关键字是职工号与电话号码的组合。关系模式为：职工(职工号，姓名，电话号码)。

二是职工号为关键字，电话号码分为单位电话和住宅电话两个属性。关系模式为：职工(职工号，姓名，单位电话，住宅电话)。

三是职工号为关键字，但强制每条记录只能有一个电话号码。关系模式为：职工(职工号，姓名，电话号码)。

四分析设计成两个关系，关系模式分别为：职工(职工号，姓名)，职工电话(职工号，电话号码)，两关系的关键字分别是职工号，职工号与电话号码的组合。

以上四个方法，可按实际情况选取使用。

4.2.5 第二范式

- 1. 第二范式的定义

定义4.7

如果关系模式 $R \in 1NF$ ， $R(U, F)$ 中的所有非主属性都完全函数依赖于任意一个候选关键字，则称关系 R 是属于第二范式（**Second Normal Form**），简称**2NF**，记作 $R \in 2NF$ 。

从定义可知，满足第二范式的关系模式 R 中，不可能有某非主属性对某候选关键字存在部分函数依赖。下面让我们来分析下4.1.2节中给出的关系模式SDC。

在关系模式SDC中，它的关系键是（SNO，CNO）的属性组合，函数依赖关系有：

$(SNO, CNO) \xrightarrow{f} SCORE$

$SNO \rightarrow SN, (SNO, CNO) \xrightarrow{p} SN$

$SNO \rightarrow AGE, (SNO, CNO) \xrightarrow{p} AGE$

$SNO \rightarrow DEPT, (SNO, CNO) \xrightarrow{p} DEPT, DEPT \rightarrow MN$

$SNO \xrightarrow{t} MN, (SNO, CNO) \xrightarrow{p} MN$

我们可以用函数依赖图表示以上函数依赖关系，如图4.4所示。

由此可见，在SDC中，既存在完全函数依赖，又存在部分函数依赖和传递函数依赖，所以SDC2NF，这种情况往往在数据库中是不允许的，也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数据冗余、插入异常、删除异常、修改异常等弊端。

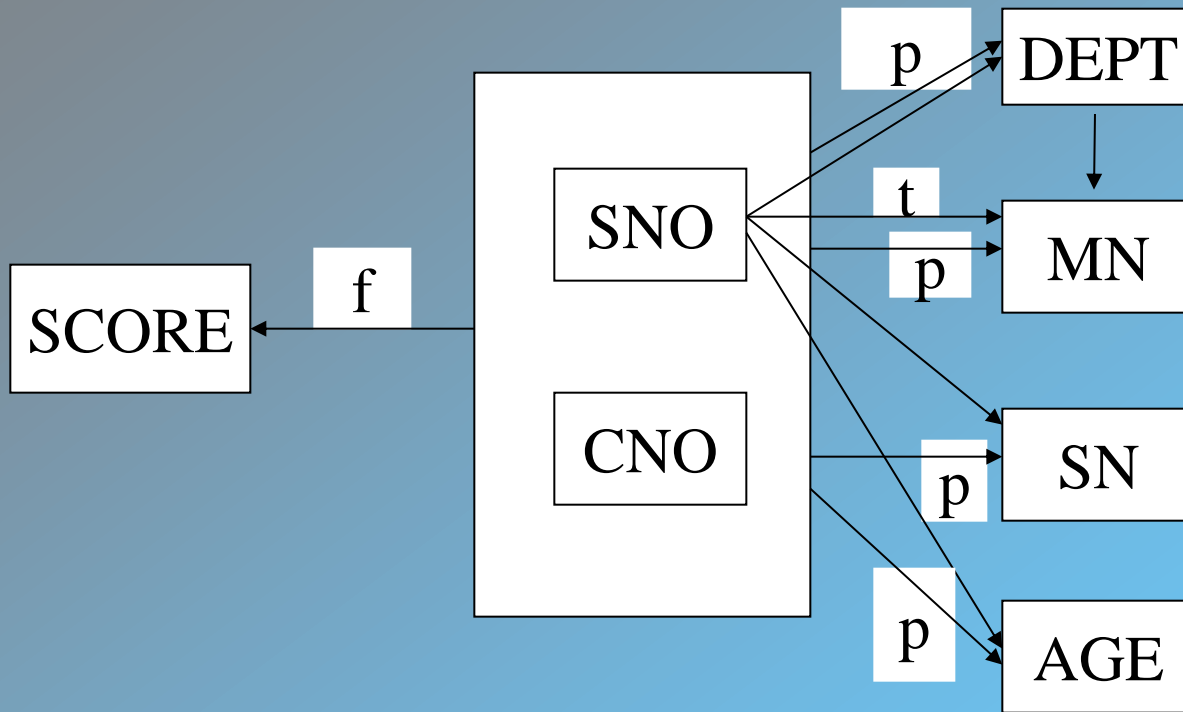


图4.4 SDC中函数依赖图

2. 2NF的规范化

2NF规范化是指把1NF关系模式通过投影分解，消除非主属性对候选关键字的部分函数依赖，转换成2NF关系模式的集合的过程。

分解时遵循的原则是“一事一地”，让一个关系只描述

一个实体或实体间的联系。如果多于一个实体或联系，则进行投影分解。

据此我们可以将关系模式SDC分解成两个关系模式：
SD (SNO, SN, AGE, DEPT, MN)，描述学生实体
SC (SNO, CNO, SCORE)，描述学生与课程的联系。

对于分解后的关系模式SD的码为SNO，关系模式SC的候选关键字为（SNO，CNO），非主属性对候选关键字均是完全函数依赖的，这样就消除了非主属性对候选关键字的部分函数依赖。即 $SD \in 2NF$ ， $SC \in 2NF$ ，它们之间通过SC中的外键SNO相联系，需要时再进行自然联接，能恢复成原来的关系，这种分解不会丢失任何信息，具有无损连接性。

分解后的函数依赖图如图4.5和4.6所示。

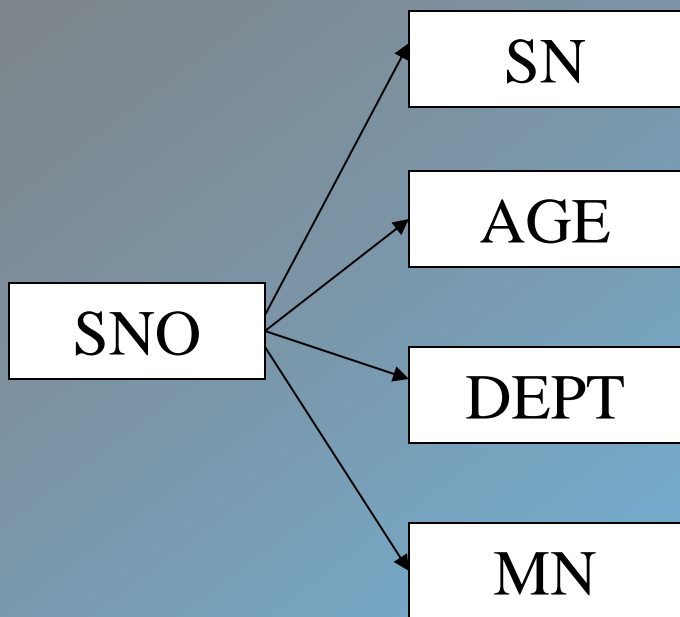


图4.5 SD中的函数依赖关系图

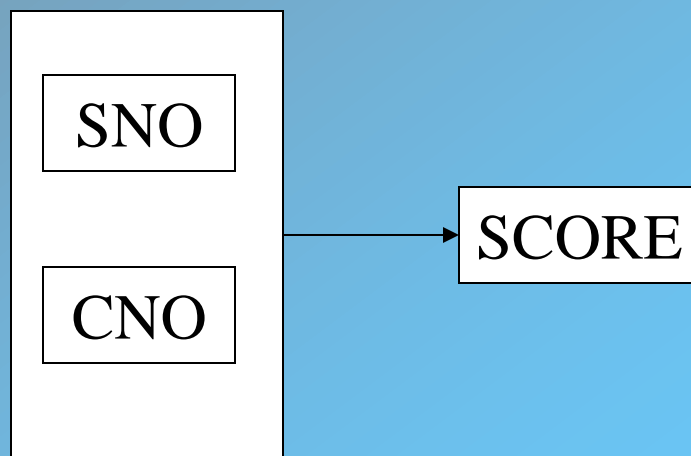


图4.6 SC中的函数依赖关系

注意：如果R的候选关键字均为单属性，或R的全体属性均为主属性，则 $R \in 2NF$ 。

例如，在讲述全码的概念时给出的关系模式TCS (T, C, S)，(T, C, S)三个属性的组合才是其唯一的候选关键字即关系键，T, C, S均是主属性，不存在非主属性，所以也不可能存在非主属性对候选关键字的部分函数依赖，因此 $TCS \in 2NF$ 。

4.2.6 第三范式

1. 第三范式的定义

定义4.8

如果关系模式 $R \in 2NF$ ， $R(U, F)$ 中所有非主属性对任何候选关键字都不存在传递函数依赖，则称 R 是属于第三范式(Third Normal Form)，简称3NF，记作 $R \in 3NF$ 。

第三范式具有如下性质：

(1) 如果 $R \in 3NF$ ，则 R 也是 $2NF$ 。

证明：采用反证法。设 $R \in 3NF$ ，但 $R \notin 2NF$ 。则根据判定 $2NF$ 的定义知，必有非主属性 A_i ($A_i \in U$ ， U 是 R 的所有属性集)，候选关键字 K 和 K 的真子集 K' (即 $K' \subset K$)存在，使得有 $K' \rightarrow A_i$ 。由于 A_i 是非主属性，所以 $A_i - K \neq \Phi$ (代表空)， $A_i - K' \neq \Phi$ 。由于 $K' \subset K$ ，所以 $K - K' \neq \Phi$ ，并可以断定 $K' \twoheadrightarrow K$ 。这样有 $K \rightarrow K'$ 且 $K' \twoheadrightarrow K$ ， $K' \rightarrow A_i$ ，且 $A_i - K \neq \Phi$ ， $A_i - K' \neq \Phi$ ，即有非主属性 A_i 传递函数依赖于候选键 K (若觉得 $K' \subset K$ ，则可以在 K' 上合并一个 A_j ，设 A_j 亦为非主属性，此时仍有 $K \rightarrow K'A_j$ 且 $K'A_j \not\subseteq K$ ， $K'A_j \twoheadrightarrow K$ ， $K'A_j \rightarrow A_i$)，所以 $R \notin 3NF$ ，与题设 $R \in 3NF$ 相矛盾。从而命题得证。

(2) 如果 $R \in 2NF$ ，则 R 不一定是 $3NF$ 。

例如：前面讲的关系模式 SDC 分解为 SD 和 SC ，其中 SC 是 $3NF$ ，但 SD 就不是 $3NF$ ，因为 SD 中存在非主属性对候选关键字的传递函数依赖： $SNO \rightarrow DEPT$ ， $DEPT \rightarrow MN$ ，即 $SNO \xrightarrow{t} MN$ 。

$2NF$ 的关系模式解决了 $1NF$ 中存在的一些问题，但 $2NF$ 的关系模式 SDC 在进行数据操作时，仍然存在下面一些问题：

- (1) 数据冗余。如果每个系名和系主任的名字存储的次数等于该系学生的人数。
- (2) 插入异常。当一个新系没有招生时，有关该系的信息无法插入。
- (3) 删除异常。如某系学生全部毕业而没有招生时，删除全部学生的记录也随之删除了该系的有关信息。
- (4) 修改异常。如更换系主任时仍需要改动较多的学生记录。

之所以存在这些问题，是由于在 SDC 中存在着非主属性对候选关键字的传递依赖。消除这种依赖就转换成了 $3NF$ 。

2. 3NF的规范化

3NF规范化是指把2NF关系模式通过投影分解，消除非主属性对候选关键字的传递函数依赖，而转换成3NF关系模式集合的过程。

3NF规范化同样遵循“一事一地”原则。我们将只属于2NF的关系模式SD规范为3NF。根据“一事一地”原则可SD分解为：

S (SNO, SN, AGE, DEPT)，描述学生实体； D (DEPT, MN)，描述系的实体。

分解后S和D的主键分别为SNO和DEPT，不存在传递函数依赖。所以 $S \in 3NF$ ， $D \in 3NF$ 。S和D的函数依赖分别如图4.7和图4.8所示。

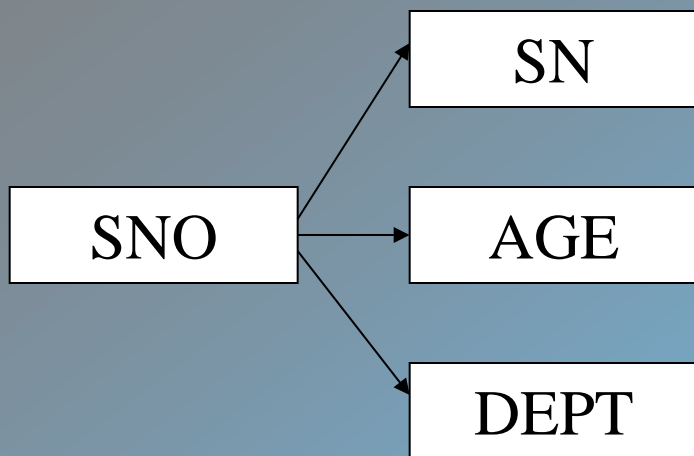


图4.7 S中的函数倚赖关系图 4.8 D中的函数依赖关系图

由以上两图可以看出，关系模式SD由2NF分解为3NF后，函数依赖关系变得更加简单，既没有非主属性对码的部分依赖，也没有非主属性对码的传递依赖，解决了2NF中存在的四个问题，因此，分解后的关系模式S和D具有以下特点：

(1) 数据冗余度降低了。如系主任的名字存储的次数与该系的学生人数无关，只在关系D中存储一次。

(2) 不存在插入异常。如当一个新系没有学生时，该系的信息可以直接插入到关系D中，而与学生关系S无关。

(3) 不存在删除异常。如当要删除某系的全部学生而仍然保留该系的有关信息时，可以只删除学生关系S中的相关记录，而不影响系关系D中的数据。

(4) 不存在修改异常。如更换系主任时，只需修改关系D中一个相应元组的MN属性值，从而不会出现数据的不一致现象。

SDC规范化到3NF后，所存在的异常现象已经全部消失。但是，3NF只限制了非主属性对码的依赖关系，而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、删除异常和修改异常。这时，则需对3NF进一步规范化，消除主属性对码的依赖关系，向更高一级的范式BCNF转换。

4.2.7 BC范式

1. BC范式的定义

定义4.9

如果关系模式 $R \in 1NF$ ，且所有的函数依赖 $X \rightarrow Y$ （ Y 不包含于 X ，即 $Y \not\subseteq X$ ），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），记作 $R \in BCNF$ 。

由BCNF的定义可以得到以下结论，一个满足BCNF的关系模式有：

- (1) 所有非主属性对每一个候选码都是完全函数依赖。
- (2) 所有的主属性对每一个不包含它的候选码都是完全函数依赖。
- (3) 没有任何属性完全函数依赖于非码的任何一组属性。

由于 $R \in BCNF$ ，按定义排除了任何属性对候选码的传递依赖与部分依赖，所以 $R \in 3NF$ 。证明留给读书完成。但若 $R \in 3NF$ ，则 R 未必属于 $BCNF$ 。下面举例说明。

例设有关系模式 $SCS(SNO, SN, CNO, SCORE)$ ，其中 SNO 代表学号， SN 代表学生姓名，并假设不重名， CNO 代表课程号， $SCORE$ 代表成绩。可以判定， SCS 有两个候选键 (SNO, CNO) 和 (SN, CNO) ，其函数依赖如下：

$$SNO \leftrightarrow SN$$

$$(SNO, CNO) \rightarrow SCORE$$

$$(SN, CNO) \rightarrow SCORE$$

唯一的非主属性SCORE对键不存在部分函数依赖，也不存在传递函数依赖。所以SCS \in 3NF。但是，因为SNO \longleftrightarrow SN，即决定因素SNO或SN不包含候选键，从另一个角度说，存在着主属性对键的部分函数依赖： $(SNO, CNO) \xrightarrow{p} SN$ ， $(SN, CNO) \xrightarrow{p} SNO$ ，所以SCS不是BCNF。正是存在着这种主属性对键的部分函数依赖关系，造成了关系SCS中存在着较大的数据冗余，解决这一问题的办法仍然是通过投影分解进一步提高范式的等级，将其规范到BCNF。

- 2. BCNF规范化

BCNF规范化是指把3NF的关系模式通过投影分解转换成BCNF关系模式的集合。

下面以3NF的关系模式SCS为例，来说明BCNF规范化的过程。

例 将SCS (SNO, SN, CNO, SCORE) 规范到BCNF。
SCS产生数据冗余的原因是因为在这个关系中存在两个实体，一个为学生实体，属性有SNO, SN；另一个为选课实体，属性有SNO, CNO和SCORE。根据分解的原则，我们可以将SCS分解成如下两个关系：

S (SNO, SN) ， 描述学生实体

SC (SNO, CNO, SCORE) ， 描述学生与课程的联系。

对于S，有两个候选码SNO和SN；对于SC，主码为（SNO，CNO）。在这两个关系中，无论主属性还是非主属性都不存在对码的部分函数依赖和传递依赖， $S \in BCNF$ ， $SC \in BCNF$ 。

分解后，S和SC的函数依赖分别如图4.9和图4.10所示。



图 4.9 S中的函数依赖关系图

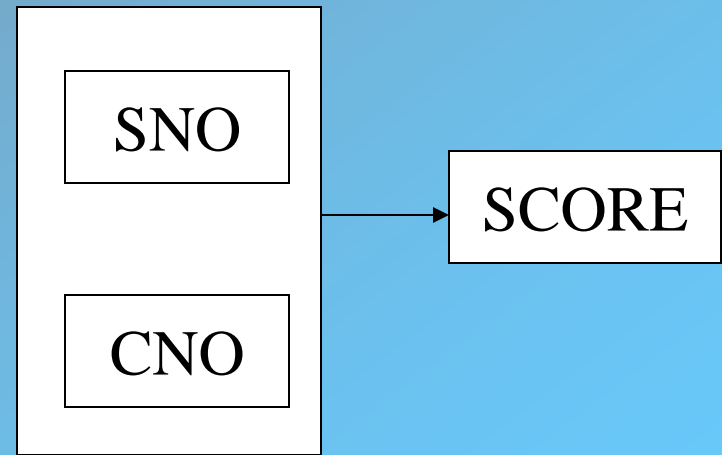


图4.10 SC中的函数依赖关系图

关系SCS转换成BCNF后，数据冗余度明显降低。学生的姓名只在关系S中存储一次，学生要改名时，只需改动一条学生记录中相应的SN值即可，从而不会发生修改异常。

下面再举一个有关BCNF规范化的实例。

[例5] 设有关系模式STK (S, T, K)，S表示学生，T表示教师，K表示课程。语义假设是，每一位教师只讲授一门课程；每门课程由多个教师讲授；某一学生选定某门课程，就对应一个确定的教师。

根据语义假设，STK的函数依赖是： $(S, K) \rightarrow T$ ， $(S, T) \rightarrow K$ ， $T \rightarrow K$ 。

函数依赖图如图4.11所示。

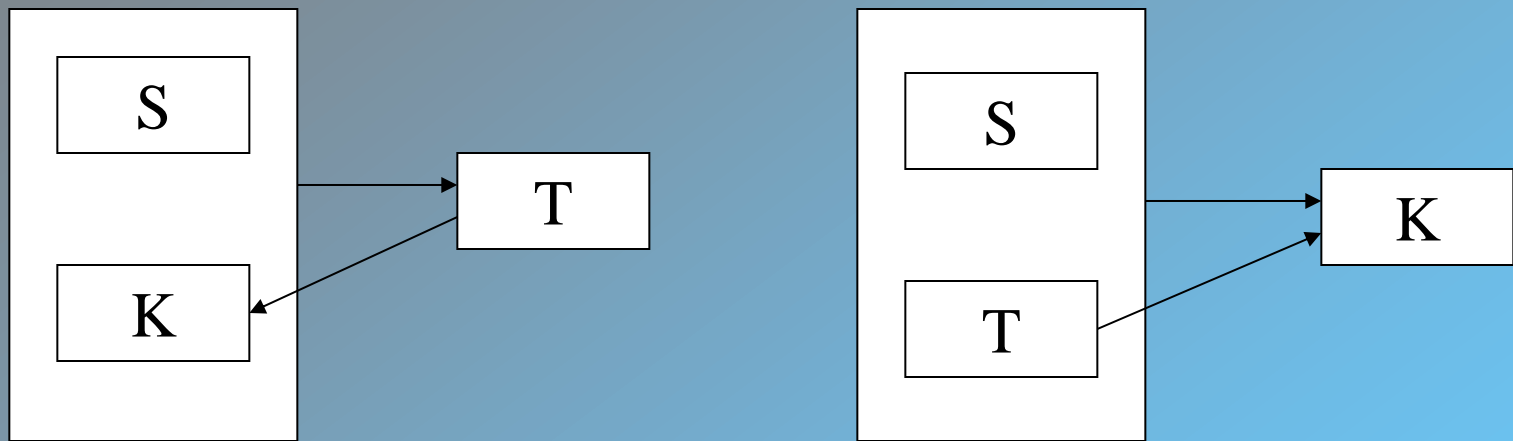


图4.11 STK中的函数依赖关系

这里 (S, K) , (S, T) 都是候选码。

STK是3NF，因为没有任何非主属性对码的传递依赖或部分依赖（因为STK中没有非主属性）。但STK不是BCNF关系，因为有 $T \rightarrow K$ ， T是决定因素，而T不包含候选码。

对于不是BCNF的关系模式，仍然存在不合适的地方。非BCNF的关系模式STK可分解为ST (S, T) 和TK (T, K) ，它们都是BCNF。

3NF和BCNF是在函数依赖的条件下对模式分解所能达到的分离程度的测度。一个模式中的关系模式如果都属于BCNF，那么在函数依赖范畴内，它已实现了彻底的分离，已消除了插入和删除异常。3NF的“不彻底”性表现在可能存在主属性对候选码的部分依赖和传递依赖。

4.2.8 多值依赖与4NF

- 1. 多值依赖

- (1) 多值依赖的定义

一个关系属于BCNF范式，是否就已经很完美了呢？为此，我们先看一个例子。

[例6] 假设学校中一门课程可由多名教师教授，教学中他们使用相同的一套参考书，这样我们可用如图4.12的非规范化的关系来表示课程C、教师T和参考书R间的关系。

如果我们把图4.12的关系CTR转化成规范化的关系，如图4.13所示。

课程C	教员T	参考书R
数据库系统概论	萨师焯 王珊	数据库原理与应用 数据库系统 SQL Server 2000
计算数学	张平 周峰	数学分析 微分方程

图4.12 关系CTR

课程C	教师T	参考书R
数据库系统概论	萨师煊	数据库原理与应用
数据库系统概论	萨师煊	数据库系统
数据库系统概论	萨师煊	SQL Server 2000
数据库系统概论	王珊	数据库原理与应用
数据库系统概论	王珊	数据库系统
数据库系统概论	王珊	SQL Server 2000
计算数学	张平	数学分析
计算数学	张平	微分方程
计算数学	周峰	数学分析
计算数学	周峰	微分方程

图4.13 规范后的关系CTR

由此可以看出，规范后的关系模式CTR，只有唯一的一个函数依赖 $(C, T, R) \rightarrow U$ (U即关系模式CTR的所有属性的集合)，其码显然是 (C, T, R) ，即全码，因而CTR属于BCNF范式。但是进一步分析可以看出，CTR还存在着如下弊端：

- ① 数据冗余大。课程、教师和参考书都被多次存储。
- ② 插入异常。若增加一名教授“计算数学”的教师“李静”时，由于这个教师也使用相同的一套参考书，所以需要添加两个元组，即：（计算数学，李静，数学分析）和（计算数学，李静，微分方程）。
- ③ 删除异常。若要删除某一门课的一本参考书，则与该参考书有关的元组都要被删除，如：删除“数据库系统概论”课程的“数据库系统”，则需要删除（数据库系统概论，萨师焯，数据库系统）和（数据库系统概论，王珊，数据库系统）两个元组。

产生以上弊端的原因主要有以下两方面：

① 对于关系CTR中的 C的一个具体值来说，有多个T值与其相对应；同样，C与R间也存在着类似的联系。

② 对于关系CTR中的一个确定的C值，与其所对应的一组T值与R值无关。如：与“数据库系统概论”课程对应的一组教师与此课程的参考书毫无关系。

从以上两个方面可以看出，C与T间的联系显然不是函数依赖，在此我们称之为多值依赖（Multivalued Dependency, MVD）。

定义4.10

设有关系模式 $R(U)$ ， U 是属性全集， X, Y, Z 是属性集 U 的子集，且 $Z=U-X-Y$ ，如果对于 R 的任一关系，对于 X 的一个确定值，存在的一组值与之对应，且 Y 的这组值仅仅决定于 X 的值而与 Z 值无关，此时称 Y 多值依赖于 X ，或 X 多值决定 Y ，记作：

$X \twoheadrightarrow Y$ 。

在多值依赖中，若 $X \twoheadrightarrow Y$ 且 $Z=U-X-Y \neq \phi$ ，则称 $X \twoheadrightarrow Y$ 是非平凡的多值依赖，否则称为平凡的多值依赖。

如：在关系模式 CTR 中，对于某一 C, R 属性值组合（数据库系统概论，数据库系统）来说，有一组 T 值{萨师焯，王珊}，这组值仅仅决定与课程 C 上的值（数据库系统概论）。也就是说，对于另一个 C, R 属性值组合（数据库系统概论，SQL

Server 2000），它对应的一组 T 值仍是{萨师焯，王珊}，尽管这时参考书 R 的值已经改变了。

因此 T 多值依赖于 C ，即： $C \twoheadrightarrow T$ 。

下面是多值依赖的另一形式化定义：

设有关系模式 $R(U)$ ， U 是属性全集， X 、 Y 、 Z 是属性集合 U 的子集，且 $Z=U-X-Y$ ， r 是关系模式 R 的任一关系， t, s 是 r 的任意两个元组，如果 $t[X]=s[X]$ ， r 中必有的两个元组 u 、 v 存在，使得：

① $s[x]=t[X]=u[X]=v[X]$

② $u[Y]=t[Y]$ 且 $u[Z]=s[Z]$

③ $v[Y]=s[Y]$ 且 $v[Z]=t[Z]$

则称 X 多值决定 Y 或 Y 多值依赖于 X 。

- (2) 多值依赖与函数依赖的区别

① 在关系模式R中，函数依赖 $X \rightarrow Y$ 的有效性仅仅决定与X、Y这两个属性集，不涉及第三个属性集，而在多值依赖中， $X \twoheadrightarrow Y$ 在属性集U ($U=X+Y+Z$) 上是否成立，不仅要检查属性集X、Y上的值，而且要检查属性集U的其余属性Z上的值。因此，如果 $X \twoheadrightarrow Y$ 在属性集W ($W \subset U$) 上成立，而在属性集U上不一定成立，所以，多值依赖的有效性与属性集的范围有关。

如果在R (U) 上有 $X \twoheadrightarrow Y$ ，在属性集W ($W \subset U$) 上也成立，则称 $X \twoheadrightarrow Y$ 为R (U) 的嵌入型多值依赖。

② 如果在关系模式R上存在函数依赖 $X \rightarrow Y$ ，则任何 Y' 包含于Y均有 $X \rightarrow Y'$ 成立，而多值依赖 $X \twoheadrightarrow Y$ 在R上成立，但不能断言对于任何 Y' 包含于Y有 $X \twoheadrightarrow Y'$ 成立。

(3) 多值依赖的性质

- ① 多值依赖具有对称性。即若 $X \twoheadrightarrow Y$ ，则 $X \twoheadrightarrow Z$ ，其中 $Z = U - X - Y$ 。
- ② 多值依赖具有传递性。即若 $X \twoheadrightarrow Y$ ， $Y \twoheadrightarrow Z$ ，则 $X \twoheadrightarrow Z - Y$ 。
- ③ 函数依赖可看作是多值依赖的特殊情况。即若 $X \rightarrow Y$ ，则 $X \twoheadrightarrow Y$ 。
- ④ 函数依赖合并性。即若 $X \rightarrow Y$ ， $X \rightarrow Z$ ，则 $X \rightarrow YZ$ 。
- ⑤ 多值依赖分解性。即若 $X \twoheadrightarrow Y$ ， $X \twoheadrightarrow Z$ ，则 $X \twoheadrightarrow (Y \cap Z)$ ， $X \twoheadrightarrow Y - Z$ ， $X \twoheadrightarrow Z - Y$ 均成立。这说明，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。

1. 第四范式 (4NF)

- (1) 第四范式 (4NF) 的定义

在4.2.8节中我们分析了关系CTR虽然属于BCNF，但还存在着数据冗余、插入异常和删除异常的弊端，究其原因就是CTR中存在非平凡的多值依赖，而决定因素不是码。因而必须将CTR继续分解，如果分解成两个关系模式CTR1 (C, T) 和CTR2 (C, R)，则它们的冗余度会明显下降。从多值依赖的定义分析CTR1和CTR2，它们的属性间各有一个多值依赖 $C \twoheadrightarrow T$ ， $C \twoheadrightarrow R$ ，都是平凡的多值依赖。因此，含有多值依赖的关系模式中，减少数据冗余和操作异常的常用方法是关系模式分解为仅有平凡的多值依赖的关系模式。

定义4.11

设有一关系模式 $R(U)$ ， U 是其属性全集， X 、 Y 是 U 的子集， D 是 R 上的数据依赖集。如果对于任一多值依赖 $X \twoheadrightarrow Y$ ，此多值依赖是平凡的，或者 X 包含了 R 的一个候选码，则称关系模式 R 是第四范式的，记作： $R \in 4NF$ 。

由此定义可知：关系模式 CTR 分解后产生的 $CTR1(C, T)$ 和 $CTR2(C, R)$ 中，因为 $C \twoheadrightarrow T$ ， $C \twoheadrightarrow R$ 均是平凡的多值依赖，所以 $CTR1$ 和 $CTR2$ 都是 $4NF$ 。

经过上面分析可以得知：一个 $BCNF$ 的关系模式不一定是 $4NF$ ，而 $4NF$ 的关系模式必定是 $BCNF$ 的关系模式，即 $4NF$ 是 $BCNF$ 的推广， $4NF$ 范式的定义涵盖了 $BCNF$ 范式的一定。

- (2) 4NF的分解

把一个关系模式分解为4NF的方法与分解为BCNF的方法类似，就是当把一个关系模式利用投影的方法消去非平凡且非函数依赖的多值依赖，并具有无损连接性。

[例7] 设有关系模式R (A, B, C, E, F, G)，数据依赖集 $D = \{A \twoheadrightarrow BGC, B \rightarrow AC, C \rightarrow G\}$ ，将R分解为4NF。

解：利用 $A \twoheadrightarrow BGC$ ，可将R分解为 $R_1 (\{ABCG\}, \{A \twoheadrightarrow BGC, B \rightarrow AC, C \rightarrow G\})$ 和 $R_2 (\{AEF\}, \{\})$ ，其中 R_2 既无函数依赖又无多值依赖，其已是4NF的关系模式。而 R_1 根据4NF的定义还不是4NF的关系模式。

再利用 $B \rightarrow AC$ 对 R_1 再分解为 $R_{11} (\{ABC\}, \{B \rightarrow AC\})$ 和 $R_{12} (\{BG\}, \{\})$ ，显然 R_{11} 、 R_{12} 都是4NF的关系模式了。

由此对R分解得到的三个关系模式R11 ({ABC}, {B→AC})、R12 ({BG}, { }) 和R2 ({AEF}, { })，它们都属于4NF，但此分解丢失了函数依赖{C→G}。若最后一次分解利用到函数依赖C→G来做，则由此得到R的另一分解的三个关系模式R11 ({ABC}, {B→AC})、R12 ({CG}, {C→G}) 和R2 ({AEF}, { })，它们同样都是属于4NF的关系模式，且保持了所有的数据依赖。这说明，4NF的分解结果不是唯一的，结果与选择数据依赖的次序有关。任何一个关系模式都可无损分解成一组等价的4NF关系模式，但这种分解不一定具有依赖保持性。

4.2.9 连接依赖与5NF*

- (1) 连接依赖的定义
- **定义4.12** 设有关系模式 $R(U)$ 、 $R_1(U_1)$ 、 $R_2(U_2)$ 、...、 $R_n(U_n)$ ，且 $U=U_1 \cup U_2 \cup \dots \cup U_n$ ， $\{R_1, \dots, R_n\}$ 是 R 的一个分解， r 为 R 的一个任意的关系实例，若（表示 r 在 $R_i(U_i)$ 上的投影，即， $i=1, 2, \dots, n$ ）则称 R 满足连接依赖（Join Dependency，简称JD），记作 $\infty(R_1, \dots, R_n)$ 。
- (2) 平凡连接依赖和非平凡连接依赖：
设关系模式 R 满足连接依赖，记 $\infty(R_1, \dots, R_n)$ 。若存在 $R_i \in \{R_1, R_2, \dots, R_n\}$ ，有 $R=R_i$ ，则称该连接依赖为平凡的连接依赖，否则称为非平凡连接依赖。

4.2.9 连接依赖与5NF*

- (3) 第五范式(5NF)
- **定义4.13** 设有关系模式 $R(U)$ 、 $R_1(U_1)$ 、 $R_2(U_2)$ 、...、 $R_n(U_n)$ ，且 $U=U_1 \cup U_2 \cup \dots \cup U_n$ ， D 是 R 上的函数依赖、多值依赖和连接依赖的集合。若对于 D^+ （称为 D 的闭包，是 D 所蕴含的函数依赖、多值依赖和连接依赖的全体，可参阅4.3节中的相关概念）中的每个非平凡连接依赖 $\infty(R_1, \dots, R_n)$ ，其中的每个 R_i 都包含 R 的一个候选键，则称 R 属于第五范式，记 $R \in 5NF$ 。

4.2.9 连接依赖与5NF*

- 举例：设关系模式SPJ({S,P,J})的属性分别表示供应商、零件、项目等含义，表示三者间的供应关系。如果规定模式R的关系是三个二元投影（SP({S,P}、PJ({P,J} JS({J,S}）的连接，而不是其中任何两个的连接。例如设关系中有<S1,P1,J2>、<S1,P2,J1>两个元组，则SPJ满足投影分解为SP、PJ、SJ后，SPJ一定是SP、PJ、SJ的连接而非它们间的两两连接，那么模式SPJ中存在着一个连接依赖 ∞ (SP,PJ,JS)。
- 在模式SPJ存在这个链接依赖时，其关系将存在冗余和异常现象。元组在插入或删除时就会出现各种异常，如插入一元组必须连带插入另一元组，而删除一元组时必须连带删除另外元组等等，因为只有这样才能不违反模式SPJ存在的连接依赖。

4.2.9 连接依赖与5NF*

- 例如，在上面SPJ中有两个元组的情况下，再插入元组 $\langle S2, P1, J1 \rangle$ ，读者会发现，有三个元组的SPJ，分解后的3个二元关系SP、PJ、SJ连接后产生的SPJ不等于分解前的SPJ，而是多了一个元组 $\langle S1, P1, J1 \rangle$ ，这就表明，根据语义的约束（或为了保证SPJ中连接依赖的存在），在插入 $\langle S2, P1, J1 \rangle$ 时，必须同时插入 $\langle S1, P1, J1 \rangle$ 的。读者还可以验证，在SPJ中有以上4个元组后，再删除 $\langle S2, P1, J1 \rangle$ 或 $\langle S1, P1, J1 \rangle$ 时，也有需要连带删除其余某些元组的现象。这就是SPJ中存在非平凡连接依赖后，存在操作异常的想象。

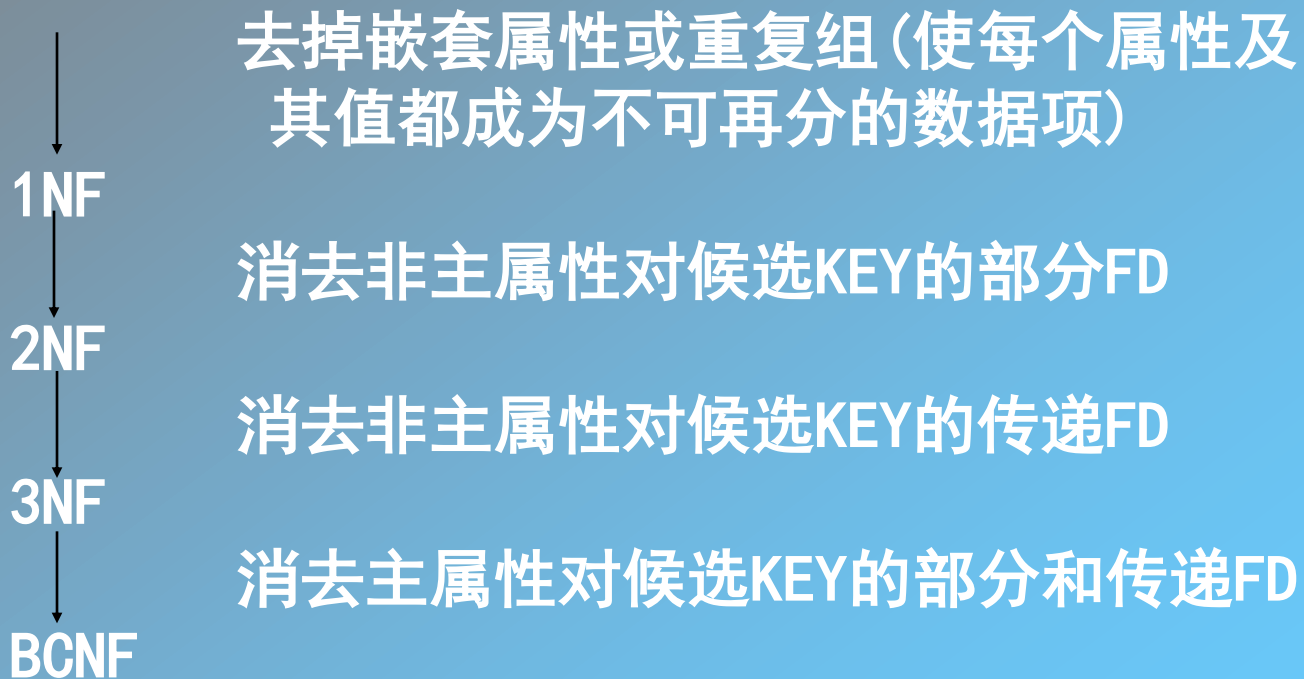
4.2.9 连接依赖与5NF*

- 关系SPJ，其一个连接依赖 $\infty(\text{SP}, \text{PJ}, \text{JS})$ 是非平凡的连接依赖，显然不满足5NF定义要求，它达不到5NF。应该把SPJ分解成SP($\{S, P\}$)、PJ($\{P, J\}$) JS($\{J, S\}$)三个模式，这样这个分解是无损分解，并且每个模式都是5NF，各模式已清楚了冗余和异常操作现象。
- 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。但它不像FD和MVD的语义那么直观，要判断一个模式是否是5NF的往往也比较困难。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的模式集。

4.2.9 规范化小结

1、关系规范化过程

非规范关系



2、结论

- 1) 3NF必定为2NF和1NF，反之不一定；
- 2) BCNF必为3NF，反之不一定；
- 3) 3NF已在很大程度上控制了数据冗余；
- 4) 3NF已在很大程度上消去了插入和删除操作异常；
- 5) 3NF分解仍不够彻底（可能存在主属性对候选码的部分fd和传递fd）；
- 6) 在fd范围内，BCNF下已完全消去了插入删除异常；
- 7) 范式并非越高越好；范式越高，异常越少，但查询操作越麻烦；
- 8) 适可而止：
理论上：一般到3NF
应用：存取垃圾；连接运算。
- 9) 分解不唯一。

[BACK](#)

4.3 数据依赖的公理系统*

数据依赖的公理系统是模式分解算法的理论基础，下面先讨论函数依赖的一个有效而完备的公理系统——Armstrong公理系统。

定义4.12

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

为了求得给定关系模式的码，为了从一组函数依赖求得蕴涵的函数依赖，例如已知函数依赖集 F ，要问 $X \rightarrow Y$ 是否为 F 所蕴含，就需要一套推理规则，这组推理规则是1974年首先由Armstrong提出来的。

Armstrong公理系统 设U为属性集总体，F是U上的一组函数依赖，于是有关系模式R (U, F)。对R (U, F) 来说有以下的推理规则：

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

注意：由自反律所得到的函数依赖均是平凡的函数依赖，自反律的使用并不依赖于F。

定理4.1 Armstrong推理规则是正确的。
下面从定义出发证明推理规则的正确性。

证： (1) $Y \subseteq X \subseteq U$ 。

对 $R(U, F)$ 的任一关系 r 中的任意两个元组 t, s ：
若 $t[X]=s[X]$ ，由于 $Y \subseteq X$ ，有 $t[Y]=s[Y]$ ，所以 $X \rightarrow Y$ 成立，自反律得证。

(2) $X \rightarrow Y$ 为 F 所蕴含，且 $Z \subseteq U$ 。

设 $R(U, F)$ 的任一关系 r 中的任意两个元组 t, s ：
若 $t[XZ]=s[XZ]$ ，则有 $t[X]=s[X]$ 和 $t[Z]=s[Z]$ ；由 $X \rightarrow Y$ ，于是有 $t[Y]=s[Y]$ ，所以 $t[YZ]=s[YZ]$ ，所以 $XZ \rightarrow YZ$ 为 F 所蕴含，增广律得证。

(3) 设 $X \rightarrow Y$ 及 $Y \rightarrow Z$ 为 F 所蕴含。

设 $R(U, F)$ 的任一关系 r 中的任意两个元组 t, s ：
若 $t[X]=s[X]$ ，由 $X \rightarrow Y$ ，有 $t[Y]=s[Y]$ ；再由 $Y \rightarrow Z$ ，有 $t[Z]=s[Z]$ ，所以 $X \rightarrow Z$ 为 F 所蕴含，传递律得证。

根据A1, A2, A3这三条推理规则可以得到下面很有用的推理规则:

- 合并规则: 由 $X \rightarrow Y, X \rightarrow Z, X \rightarrow YZ$ 。
- 伪传递规则: 由 $X \rightarrow Y, WY \rightarrow Z$, 有 $XW \rightarrow Z$ 。
- 分解规则: 由 $X \rightarrow Y$ 及 $Z \subseteq Y$, 有 $X \rightarrow Z$ 。

根据合并规则和分解规则, 很容易得到这样一个重要事实:

引理4.1 $X \rightarrow A_1 A_2 \dots A_k$ 成立的充分必要条件是 $X \rightarrow A_i$ 成立 ($i=1, 2, \dots, k$)。

定义4.13

在关系模式 $R(U, F)$ 中为 F 所蕴含的函数依赖的全体叫做 F 的**闭包**, 记为: F^+ 。

人们把自反律，传递律和增广律称为Armstrong公理系统。Armstrong公理系统是有效的、完备的。

Armstrong公理的**有效性**指的是：由F出发根据Armstrong公理推导出来的每一个函数依赖一定在 F^+ 中；**完备性**指的是 F^+ 中的每一个函数依赖，必定可以由F出发根据Armstrong公理推导出来。

要证明完备性，就首先要解决如何判定一个函数依赖是否属于由F根据Armstrong公理推导出来的函数依赖集合。当然，如果能求出这个集合，问题就解决了。但不幸的是，这是个NP完全问题。比如从 $F = \{ X \rightarrow A_1, \dots, X \rightarrow A_n \}$ 出发，至少可以推导出 2^n 个不同的函数依赖，为此引出了下面的概念：

定义4.14

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

由引理4.1容易得出：

引理4.2

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

于是，判定 $X \rightarrow Y$ 是否能由 F 根据Armstrong公理推导出的问题，就转化为求出 X_F^+ 的子集的问题。这个问题由算法4.1解决了。

算法4.1 求属性集 X (XU) 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

$$B = \{A \mid (\exists V) (\exists W) (V \rightarrow W \in F \wedge V \subseteq X^{(i)} \wedge A \in W)\};$$

(3) $X^{(i+1)} = B \cup X^{(i)}$

(4) 判断 $X^{(i+1)} = X^{(i)}$ 吗?

(5) 若相等或 $X^{(i+1)} = U$, 则 $X^{(i+1)}$ 就是 X_F^+ , 算法终止。

(6) 若否, 则 $i=i+1$, 返回第(2)步。

[例8] 已知关系模式R (U, F) ,

其中 $U = \{A, B, C, D, E\}$; $F = \{AB \rightarrow C, B \rightarrow D, C \rightarrow E, EC \rightarrow B, AC \rightarrow B\}$ 。 求 $(AB)_F^+$ 。

解 由算法4.1, 设 $X^{(0)} = AB$;

计算 $X^{(1)}$; 逐一扫描F集合中各个函数依赖, 找左部为A, B或AB的函数依赖。得到两个: $AB \rightarrow C, B \rightarrow D$ 。于是 $X^{(1)} = AB \cup CD = ABCD$ 。

因为 $X^{(0)} \neq X^{(1)}$, 所以再找出左部为ABCD子集的那些函数依赖, 又得到 $C \rightarrow E, AC \rightarrow B$, 于是 $X^{(2)} = X^{(1)} \cup BE = ABCDE$ 。

因为 $X^{(2)}$ 已等于全部属性的集合, 所以 $(AB)_F^+ = ABCDE$ 。

定理4.2 Armstrong公理系统是有效的、完备的。

Armstrong公理系统的有效性可由**定理4.1**得到证明。这里给出完备性的证明。

证明完备性的逆否命题，即若函数依赖 $X \rightarrow Y$ 不能由F从Armstrong公理导出，那么它必然不为F所蕴含，它的证明分三步。

(1) 若 $V \rightarrow W$ 成立，且 $V \subseteq X_F^+$ ，则 $W \subseteq X_F^+$ 。

证：因为 $V \subseteq X_F^+$ ，所以有 $X \rightarrow V$ 成立；于是 $X \rightarrow W$ 成立（因为 $X \rightarrow V$ ， $V \rightarrow W$ ），所以 $W \subseteq X_F^+$ 。

(2) 构造一张二维表r，它由下列两个元组构成，可以证明r必是R(U, F)的一个关系，即F中的全部函数依赖在r上成立。

X_F^+	$U - X_F^+$
11.....1	00.....0
11.....1	11.....1

若 r 不是 $R(U, F)$ 的关系, 则必由于 F 中有函数依赖 $V \rightarrow W$ 在 r 上不成立所致。由 r 的构成可知, V 必定是 X_F^+ 的子集, 而 W 不是 X_F^+ 的子集, 可是由第(1)步, $W \subseteq X_F^+$, 矛盾。所以 r 必是 $R(U, F)$ 的一个关系。

(3) 若 $X \rightarrow Y$ 不能由 F 从Armstrong公理导出, 则 Y 不是 X_F^+ 的子集, 因此必有 Y 的子集 Y' 满足 $Y' \subseteq U - X_F^+$, 则 $X \rightarrow Y$ 在 r 中不成立, 即 $X \rightarrow Y$ 必不为 $R(U, F)$ 蕴含。

Armstrong公理的完备性及有效性说明了“导出”与“蕴含”是两个完全等价的概念。于是 F^+ 也可以说成是由 F 发出借助Armstrong公理导出的函数依赖集合。

从蕴含(或导出)的概念出发, 又引出了两个函数依赖集等价和最小依赖集的概念。

定义4.15 如果 $G^+=F^+$ ，就说函数依赖集F覆G（F是G的覆盖，或G是F的覆盖），或F与G等价。

引理4.3 $F^+=G^+$ 的充分必要条件是 $F \subseteq G^+$ ，和 $G \subseteq F^+$ 。

证 必要性显然，只证充分性。

(1) 若 $F \subseteq G^+$ ，则 $X_F^+ \subseteq X_{G^+}^+$ 。

(2) 任取 $X \rightarrow Y \in F^+$ ，则有 $Y \subseteq X_F^+ \subseteq X_{G^+}^+$ 。

所以 $X \rightarrow Y \in (G^+)^+ = G^+$ 。即 $F^+ \subseteq G^+$ 。

(3) 同理可证 $G^+ \subseteq F^+$ ，所以 $F^+=G^+$ 。

而要判定 $F \subseteq G^+$ ，只需逐一对F中的函数依赖 $X \rightarrow Y$ ，考察Y是否属于 $X_{G^+}^+$ 就行了。因此**引理4.3**给出了判定两个函数依赖集等价的可行算法。

定义4.16 如果函数依赖集F满足下列条件，则称F为一个极小函数依赖集。亦称为最小依赖集或最小覆盖。

(1) F中任一函数依赖的右部仅含有一个属性。

(2) F中不存在这样的函数依赖 $X \rightarrow A$ ，使得F与 $F - \{X \rightarrow A\}$ 等价。

(3) F中不存在这样的函数依赖 $X \rightarrow A$ ，X有真子集Z使得 $F - \{X \rightarrow A\} \cup \{Z \rightarrow A\}$ 与F等价。

定理4.3 每一个函数依赖集F均等价一个极小函数依赖集 F_m 。此 F_m 称为F的最小依赖集。

证：这是个构造性的证明，分三步对F进行“极小化处理”，找出F的一个最小依赖集来。

(1) 逐一检查F中个函数依赖 $FD_i: X \rightarrow Y$ ，若 $Y = A_1 A_2 \dots A_k$ ， $k \geq 2$ ，则用 $\{X \rightarrow A_j \mid j = 1, 2, \dots, k\}$ 来取代 $X \rightarrow Y$ 。

(2) 逐一检查F中各函数依赖 $FD_i: X \rightarrow A$ ，令 $G = F - \{X \rightarrow A\}$ ，若 $A \in X_G^+$ ，则从F中去掉此函数依赖（因为F与G等价的充要条件是 $A \in X_G^+$ ）。

(3) 逐一取出 F 中各函数依赖 $FD_i: X \rightarrow A$, 设 $X=B_1B_2 \dots B_m$, 逐一考察 B_i ($i=1, 2, \dots, m$), 若 $A \in (X - B_i)_{F^+}$, 则以 $X - B_i$ 取代 X (因为 F 与 $F - \{X \rightarrow A\} \cup \{Z \rightarrow A\}$ 等价的充要条件是 $A \in Z_{F^+}$ 其中 $Z = X - B_i$)。

最后剩下的 F 就一定是极小依赖集, 并且与原来的 F 等价。因为对 F 的每一次“改造”都保证了改造前后的两个

函数依赖集等价。这些证明很显然, 请读者自己补上。

应当指出, F 的最小依赖集 F_m 不一定是唯一的, 它与对各函数依赖 FD_i 及 $X \rightarrow A$ 中 X 各属性的处置顺序有关。

[例9] $F = \{ A \rightarrow B, B \rightarrow A, B \rightarrow C, A \rightarrow C, C \rightarrow A \}$

$F_{m1} = \{ A \rightarrow B, B \rightarrow C, C \rightarrow A \}$

$F_{m2} = \{ A \rightarrow B, B \rightarrow A, A \rightarrow C, C \rightarrow A \}$

这里给出了F的两个最小依赖 F_{m1} , F_{m2} 。

若改造后的F与原来的F相同, 说明F本身就是一个最小依赖集, 因此**定理4.3**的证明给出的最小化过程也可以看成是检查F是否为极小依赖集的一个算法。

两个关系模式 (U, F) , (U, G) , 如果F与G等价, 那么 R_1 的关系一定是 R_2 的关系。反过来, R_2 的关系也一定是 R_1 的关系。所以在 $R(U, F)$ 中用与F等价的依赖集G取代F是允许的。

函数依赖保持性

定义4.17 设有关系模式 $R(U, F)$ ， $Z \subseteq U$ ，则 Z 所涉及到的 F 中所有函数依赖为 F 在 Z 上的投影，记为 $\Pi_Z(F)$ ，有 $\Pi_Z(F) = \{X \rightarrow Y \mid (X \rightarrow Y) \in F^+ \text{ 且 } XY \subseteq Z\}$ 为函数依赖集 F 在 Z 上的投影。

定义4.18 设 $R(U, F)$ 的一个分解 $\rho = \{R_1, R_2, \dots, R_k\}$ ，如果 F 等价于 $\Pi_{R_1}(F) \cup \Pi_{R_2}(F) \cup \dots \cup \Pi_{R_k}(F)$ ，则称分解 ρ 具有函数依赖保持性。一个无损连接的分解不一定具有函数依赖保持性；同样地，一个具有函数依赖保持性的分解也不一定具无损连接性。

检验一个分解是否具有依赖保持性，实际上是检验 $\Pi_{R_1}(F) \cup \Pi_{R_2}(F) \cup \dots \cup \Pi_{R_k}(F)$ 是否覆盖 F 。

[BACK](#)

4.4 关系分解保持性*

关系模式的规范化就是要通过对模式进行分解，将一个属于低级范式的关系模式转换成若干个属于高级范式的关系模式，从而解决或部分解决插入异常、删除异常、修改复杂、数据冗余等问题。

4.4 关系分解保持性*

- 4.4.1 关系模式的分解
- (1) 关系模式分解的定义：设有关系模式R，U为R的属性集。R的一个分解定义为 $\rho=\{R_1, R_2, \dots, R_n\}$ ，其中 R_i 的属性集为 U_i ($i=1,2,\dots,n$)，且 $U=U_1 \cup U_2 \cup \dots \cup U_n$ 。
- (2) 函数依赖集的投影：设有关系模式R，U为R的属性集，F为R上的函数依赖集， $\rho=\{R_1, R_2, \dots, R_n\}$ 为R的一个分解，其中 R_i 的属性集为 U_i ($i=1,2,\dots,n$)，则 $F_i=\{X \rightarrow Y \mid X \rightarrow Y \in F^+, X \subseteq U_i, Y \subseteq U_i\}$ 称为函数依赖集F在属性集 U_i 上的投影。
- (3) 对模式分解的要求：一个模式可以有多种分解方法，但要使分解有意义，就应当保证在分解过程中不丢失原有模式中的信息。模式分解的无损连接性和函数依赖保持性就是用以衡量一个模式分解是否导致原有模式中部分信息丢失的两个标准。

4.4 关系分解保持性*

4.4.2 模式分解的无损连接性

(1) 基于模式分解的关系连接: 设有关系模式 $R(U)$, $\rho = \{R_1, R_2, \dots, R_n\}$ 是 R 的一个分解, 其中 R_i 的属性集为 U_i ($i=1, 2, \dots, n$) ($U_i \subseteq U$), r 是 R 的一个关系实例, 则将 r 在 ρ 中各关系模式上投影的连接 $\Pi_{R_1}(r) \bowtie \Pi_{R_2}(r) \bowtie \dots \bowtie \Pi_{R_n}(r)$ (意即为 $\Pi_{U_1}(r) \bowtie \Pi_{U_2}(r) \bowtie \dots \bowtie \Pi_{U_n}(r)$) 记作 $m_\rho(r)$ 。

可以证明, 对 $m_\rho(r)$ 有: ① $r \subseteq m_\rho(r)$; ② 若 $s = m_\rho(r)$, 则 $\Pi_{R_i}(s) = \Pi_{R_i}(r)$; ③ $m_\rho(m_\rho(r)) = m_\rho(r)$ 。

(2) 无损连接性的定义

定义 4.19 设有关系模式 R , $\rho = \{R_1, R_2, \dots, R_n\}$ 是 R 的一个分解, 若对 R 的任意关系实例 r 都有 $r = m_\rho(r)$, 则称 ρ 为具有无损连接性的分解。

4.4 关系分解保持性*

- (3) 无损连接性的判定算法
- 算法4.2 设有关系模式R, $U=\{A_1, A_2, \dots, A_m\}$ 为R的属性集, F为R上的函数依赖集, $\rho=\{R_1, R_2, \dots, R_n\}$ 为R的一个分解。
- ①建立如下判定表T, 不妨以 $T(R_i, A_j)$ 表示T中第i行与第j列交叉处的单元格, 填表要求为: 若 R_i 中包含属性 A_j 则在 $T(R_i, A_j)$ 中填入 a_j , 否则填入 b_{ij} 。
- $A_1 A_2 \dots A_m R_1 R_2 \dots R_n$ ②对F中的每个函数依赖 $X \rightarrow Y$, 若存在 ρ 中关系模式 $R_{s_1}, R_{s_2}, \dots, R_{s_k}$ ($1 \leq s_1 < s_2 < \dots < s_k \leq n$), 使对任意 $A_i \in X$, 都有 $T(R_{s_1}, A_i) = T(R_{s_2}, A_i) = \dots = T(R_{s_k}, A_i)$, 则对每一 $A_j \in Y$, 应修改单元格中的内容, 使 $T(R_{s_1}, A_j) = T(R_{s_2}, A_j) = \dots = T(R_{s_k}, A_j)$ 。
- 具体修改方法为: 如果存在 $R_{s_i} \in \{R_{s_1}, R_{s_2}, \dots, R_{s_k}\}$, 满足 $T(R_{s_i}, A_j) = a_j$, 则令 $T(R_{s_1}, A_j) = \dots = T(R_{s_i}, A_j) = \dots = T(R_{s_k}, A_j) = a_j$, 否则令 $T(R_{s_1}, A_j) = \dots = T(R_{s_k}, A_j) = b_{s_1 j}$ 。
- 此外, 当对表中某个 b_{ij} 按上述规则进行了修改时, 应对表T的第j列中所有符号 b_{ij} 进行同样的修改, 而不管这些符号所在的行是否与关系模式 $R_{s_1}, R_{s_2}, \dots, R_{s_k}$ 相对应。
- 如果在某次修改之后, 表中有一行成为 a_1, a_2, \dots, a_n , 则可判定分解 ρ 具有无损连接性, 算法终止。
- ③若第②步中未对表T进行任何修改, 则可判定 ρ 不具有无损连接性, 算法终止, 否则至少使表T中减少了一个符号, 应返回第②步, 进行下一轮处理。由于表T中符号的个数是有限的, 这样的循环一定能够终止。

4.4 关系分解保持性*

- (4) 无损连接性的判定定理
- 定理4.4 设有关系模式R，U为R的属性集，F为R上的函数依赖集， $\rho=\{R_1, R_2, \dots, R_n\}$ 为R的一个分解，用算法4.2对 ρ 进行判定，则 ρ 具有无损连接性的充分必要条件是：该算法终止时，并且表T中有一行为 a_1, a_2, \dots, a_n 。
- 定理4.5 设有关系模式R，U为R的属性集，F为R上的函数依赖集， $\rho=\{R_1, R_2\}$ 为R的一个分解， R_1 的属性集为 U_1 ， R_2 的属性集为 U_2 ，则 ρ 具有无损连接性的充分必要条件是： $(U_1 \cap U_2) \rightarrow (U_1 - U_2) \in F^+$ 或 $(U_1 \cap U_2) \rightarrow (U_2 - U_1) \in F^+$ 。

4.4 关系分解保持性*

- (1) 函数依赖保持性的定义
- **定义4.20** 设有关系模式R，U是R的属性集，F为R上的函数依赖集， $\rho=\{R_1, R_2, \dots, R_n\}$ 为R的一个分解， U_i 为 R_i 的属性集， F_i 是F在 U_i 上的投影（ $i=1,2,\dots,n$ ）。如果 $F^+= (F_1 \cup F_2 \cup \dots \cup F_n)^+$ ，则称分解 ρ 具有函数依赖保持性。

4.4 关系分解保持性*

- (2) 函数依赖保持性的判定算法
- 算法4.3 设有关系模式R，U是R的属性集，F为R上的函数依赖集， $\rho=\{R_1, R_2, \dots, R_n\}$ 为R的一个分解， U_i 为 R_i 的属性集， F_i 是F在 U_i 上的投影（ $i=1,2,\dots,n$ ）。令 $G=F_1 \cup F_2 \cup \dots \cup F_n$ ，则必有 $G \subseteq F$ ，即 $G^+ \subseteq F^+$ ，要判断 $G^+=F^+$ 是否成立，只需判断 $F^+ \subseteq G^+$ 是否成立。算法如下：①对F中的每个函数依赖 $X \rightarrow Y$ ，求X关于G的闭包 XG^+ ，若 $Y \notin XG^+$ ，则可判定 ρ 不具有函数依赖保持性，算法终止；② ρ 具有函数依赖保持性，算法终止。

4.4 关系分解保持性*

- 4.4.4 模式分解算法
- 在现有的模式分解算法中，分解到3NF的算法可以达到具有无损连接性和函数依赖保持性，分解到BCNF和4NF的算法只能达到具有无损连接性，还不能达到具有函数依赖保持性。

4.4 关系分解保持性*

- (1) 分解到3NF, 具有函数依赖保持性的模式分解算法
- 算法4.4 设有关系模式R, U为R的属性集, F为R上的函数依赖集。①令模式分解 $\rho=\varnothing$; ②若存在函数依赖 $X\rightarrow Y\in F$, 满足 $X\cup Y=U$, 则令 $\rho=\{R\}$, 转步骤⑥; ③令 $U_0=\varnothing$, 对U中的每个属性 A_i , 如果 A_i 既不出现在F中任一函数依赖的左端, 也不出现在F中任一函数依赖的右端, 则令 $U_0=U_0\cup\{A_i\}$; 以 U_0 为属性集, 构造关系模式 R_0 , 并令 $\rho=\rho\cup\{R_0\}$, $U=U-U_0$; ④若F中存在左端相同的函数依赖 $X\rightarrow Y_1, X\rightarrow Y_2, \dots, X\rightarrow Y_n$, 则对这些函数依赖进行合并, 即令 $F=(F-\{X\rightarrow Y_1, X\rightarrow Y_2, \dots, X\rightarrow Y_n\})\cup\{X\rightarrow (Y_1\cup Y_2\cup\dots\cup Y_n)\}$; 重复执行步骤④, 直至F中不存在左端相同的函数依赖; ⑤对F中的每个函数依赖 $X_i\rightarrow Y_i$, 令 $U_i=X_i\cup Y_i$, 以 U_i 为属性集构造关系模式 R_i , 再令 $\rho=\rho\cup\{R_i\}$; ⑥算法终止。

4.4 关系分解保持性*

- (2) 分解到3NF，既具有无损连接性又具有函数依赖保持性的模式分解算法
- 算法4.5 设有关系模式R，U为R的属性集，F为R上的函数依赖集。①用算法4.4对关系模式R进行分解，生成具有函数依赖保持性的模式分解 ρ ，此时 ρ 中的每个关系模式都属于3NF；②设X是R的一个候选键，以X为属性集构造关系模式RX，令 $\rho=\rho \cup \{RX\}$ ；③ ρ 为R的一个既具有无损连接性又具有函数依赖保持性的模式分解，且 ρ 中的每个关系模式都属于3NF，算法终止。
- 定理4.7 设有关系模式R，U为R的属性集，F为R上的函数依赖集， $\rho=\{R_1, R_2, \dots, R_n, R_x\}$ 是由算法4.5生成的R的一个模式分解，则 ρ 既具有无损连接性，又具有函数依赖保持性，且 ρ 中每个关系模式都属于3NF。

4.4 关系分解保持性*

- (3) 分解到BCNF，具有无损连接性的模式分解算法
- 算法4.6 设有关系模式R，U是R的属性集，F为R上的函数依赖集。①令 $\rho=\{R\}$ ；②如果 ρ 中各关系模式都属于BCNF，则转步骤④，否则继续；③任选 ρ 中不属于BCNF的关系模式 R_i ，设 R_i 的属性集为 U_i ，F在 U_i 上的投影为 F_i ，由于 R_i 不属于BCNF，则必存在函数依赖 $X \rightarrow Y \in F_i$ ，其中X不是 R_i 的候选键，且 $Y \not\subseteq X$ 。分别以属性集 $U_i - Y$ 和 $X \cup Y$ 构造关系模式 R_i' 和 R_i'' ，令 $\rho = (\rho - \{R_i\}) \cup \{R_i', R_i''\}$ ，转步骤②；④算法终止。
- 由于U中的属性个数有限，该算法经有限次循环后必能终止。

4.4 小结

本章讨论如何设计关系模式问题。关系模式设计有好与坏之分，其设计好坏与数据冗余度和各种数据异常问题直接相关。

本章在函数依赖、多值依赖的范畴内讨论了关系模式的规范化，在整个讨论过程中，只采用了两种关系运算——投影和自然连接。

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。前者能保持泛关系在投影联接以后仍能恢复回来，而后者能保证数据在投影或联接中其语义不会发生变化。

范式是衡量关系模式优劣的标准，范式表达了模式中数据依赖应满足的要求。要强调的是，规范化理论主要为数据库设计提供了理论的指南和参考工具，并不是关系模式规范化程度越高，实际应用该关系模式就越好，实际上必须结合应用环境和现实世界的具体情况合理地选择数据库模式的范式等级。

本章最后还简介了模式分解相关的理论基础——数据依赖的

习题

一、选择题

- 1、关系模式中数据依赖问题的存在，可能会导致库中数据插入异常，这是指()。
A. 插入了不该插入的数据 B. 数据插入后导致数据库处于不一致状态
C. 该插入的数据不能实现插入 D. 以上都不对
- 2、若属性X函数依赖于属性Y时，则属性X与属性Y之间具有()的联系。
A. 一对一 B. 一对多 C. 多对一 D. 多对多
- 3、关系模式中的候选键()。
A. 有且仅有一个 B. 必然有多个
C. 可以有一或多个 D. 以上都不对
- 4、规范化的关系模式中，所有属性都必须是()。
A. 相互关联的 B. 互不相关的
C. 不可分解的 D. 长度可变的
- 5、设关系模式R{A,B,C,D,E}，其上函数依赖集F={AB→C, DC→E, D→B}，则可导出的函数依赖是()。
A. AD→E B. BC→E C. DC→AB D. DB→A

BACK

6、设关系模式R属于第一范式，若在R中消除了部分函数依赖，则R至少属于()。

- A. 第一范式 B. 第二范式 C. 第三范式 D. 第四范式

7、若关系模式R中的属性都是主属性，则R至少属于()。

- A. 第三范式 B. BC范式 C. 第四范式 D. 第五范式

8、下列关于函数依赖的叙述中，哪一个是不正确的。

- A. 由 $X \rightarrow Y$, $X \rightarrow Z$, 有 $X \rightarrow YZ$ B. 由 $XY \rightarrow Z$, 有 $X \rightarrow Z$ 或 $Y \rightarrow Z$
C. 由 $X \rightarrow Y$, $WY \rightarrow Z$, 有 $XW \rightarrow Z$ D. 由 $X \rightarrow Y$ 及 $Z \subseteq Y$, 有 $X \rightarrow Z$

9、在关系模式R(A,B,C)中，有函数依赖集 $F = \{AB \rightarrow C, BC \rightarrow A\}$ ，则R最高达到()。

- A. 第一范式 B. 第二范式 C. 第三范式 D. BC范式

10、设有关系模式R(A, B, C)，其函数依赖集 $F = \{A \rightarrow B, B \rightarrow C\}$ ，则关系R最高达到()。

- A. 1NF B. 2NF C. 3NF D. BCNF

[BACK](#)

二、填空题

- 1、数据依赖主要包括_____依赖、_____依赖和连接依赖。
- 2、一个不好的关系模式会存在_____、_____和_____等弊端。
- 3、设 $X \rightarrow Y$ 为R上的一个函数依赖，若_____，则称Y完全函数依赖于X。
- 4、设关系模式R上有函数依赖 $X \rightarrow Y$ 和 $Y \rightarrow Z$ 成立，若_____且_____，则称Z传递函数依赖于X。
- 5、设关系模式R的属性集为U，K为U的子集，若_____，则称K为R的候选键。

- 6、包含R中全部属性的候选键称_____。不在任何候选键中的属性称_____。
- 7、Armstrong公理系统是_____的和_____的。
- 8、第三范式是基于_____依赖的范式，第四范式是基于_____依赖的范式。
- 9、关系数据库中的关系模式至少应属于_____范式。
- 10、规范化过程，是通过投影分解，把_____的关系模式“分解”为_____的关系模式。

三、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、范式、无损连接性、依赖保持性。
2. 给出2NF、3NF、BCNF的形式化定义，并说明它们之间的区别和联系。
3. 什么叫关系模式分解？为什么要做关系模式分解？模式分解要遵循什么准则？
4. 试证明全码的关系必是3NF，也必是BCNF。

5. 要建立关于系、学生、班级、研究会等信息的一个关系数据库。规定：一个系有若干专业、每个专业每年只招一个班，每个班有若干学生，一个系的学生住在同一个宿舍区。每个学生可参加若干研究会，每个研究会有若干学生。学生参加某研究会，有一个入会年份。

描述学生的属性有：学号、姓名、出生年月、系名、班号、宿舍区。

描述班级的属性有：班号、专业名、系名、人数、入校年份。

描述系的属性有：系号、系名、系办公室地点、人数。

描述研究会的属性有：研究会名、成立年份、地点、人数。

试给出上述数据库的关系模式；写出每个关系的最小依赖集（即基本的函数依赖集，不是导出的函数依赖）；指出是否存在传递函数依赖；对于函数依赖左部是多属性的情况，讨论其函数依赖是完全函数依赖还是部分函数依赖，指出各关系的候选键、外部键。

6. 设有关系模式 $R(A, B, C, D, E, F)$ ，函数依赖集 $F=\{(A, B) \rightarrow E, (A, C) \rightarrow F, (A, D) \rightarrow B, B \rightarrow C, C \rightarrow D\}$ ，求出 R 的所有候选关键字。

7. 设有关系模式 $R(X, Y, Z)$ ，函数依赖集为 $F=\{(X, Y) \rightarrow Z\}$ 。请确定 SC 的范式等级，并证明。

[**BACK**](#)

8. 设有关系模式R (A, B, C, D, E, F), 函数依赖集 $F=\{A \rightarrow (B, C), (B, C) \rightarrow A, (B, C, D) \rightarrow (E, F), E \rightarrow C\}$ 。试问: 关系模式R是否为BCNF范式, 并证明结论。

9. 设有关系模式R (E, F, G, H), 函数依赖 $F=\{E \rightarrow G, G \rightarrow E, F \rightarrow (E, G), H \rightarrow (E, G), (F, H) \rightarrow E\}$

- (1) 求出R的所有候选关键字;
- (2) 根据函数依赖关系, 确定关系模式R属于第几范式;
- (3) 将R分解为3NF, 并保持无损连接性和函数依赖保持性;
- (4) 求出F的最小函数依赖集。

10. 试问下列关系模式最高属于第几范式, 并解释其原因。

- (1) R (A, B, C, D), $F=\{B \rightarrow D, AB \rightarrow C\}$ 。
- (2) R (A, B, C, D, E), $F=\{AB \rightarrow CE, E \rightarrow AB, C \rightarrow D\}$ 。
- (3) R (A, B, C, D), $F=\{B \rightarrow D, D \rightarrow B, AB \rightarrow C\}$ 。
- (4) R (A, B, C), $F=\{A \rightarrow B, B \rightarrow A, A \rightarrow C\}$ 。
- (5) R (A, B, C), $F=\{A \rightarrow B, B \rightarrow A, C \rightarrow A\}$ 。
- (6) R (A, B, C, D), $F=\{A \rightarrow C, D \rightarrow B\}$ 。
- (7) R (A, B, C, D), $F=\{A \rightarrow C, CD \rightarrow B\}$ 。

[BACK](#)