

第 5 章

关系数据库的规范化设计

关系数据库的规范化设计是关系数据库原理中的主要理论之一，它与第 3 章的关系运算知识一起，构成了关系数据库最重要的、严密的数学理论基础。

关系数据库的规范化设计理论主要包含数据依赖、范式和规范化设计理论三部分内容。其中，数据依赖是核心，范式是规范化设计标准。数据库设计的一个最基本问题就是如何建立一个好的数据模式，而规范化设计理论则是指导数据模式设计标准。因此，规范化设计对关系数据库的结构设计起着非常重要的作用。本章重点介绍数据依赖中的函数依赖以及范式的判定方法。

5.1 关系模式的设计问题

关系模式是关系数据库的型，是关系数据库最重要的内容之一。设计出一个规范的关系模式，可以尽可能地消除关系数据库中的数据冗余，解决数据库操作中插入、修改和删除异常的问题。

5.1.1 概述

关系数据库的鼻祖——E.F.Codd 从 1971 年起，提出了关系数据库的规范化理论，后经过很多专家和学者的不断研究和发展，规范化理论研究已经取得很多的成果，使数据库设计的方法逐步走向完备。在此理论提出以前，层次和网状数据库的设计没有严密的数学理论依据，只是依照其模型自身的特点和原则来设计，其结果可能会给日后的运行和使用带来一些不可预见的问题。关系数据库中关系模型有严格的数学理论基础，又可以向其他的数据模型转换，因此设计一个好的关系模型需要依托规范化理论这个强有力的设计工具。

数据库设计的一个最基本问题是如何建立一个好的数据库模式，使数据库系统无论是在数据存储方面，还是在数据操作方面都有较好的性能。针对一个具体问题，应该如何构造一个适合于它的数据模式，也就是应该构造几个关系模式，每个关系模式又由哪些属性构成，如何将这些相互关联的关系模式构建成一个适合的关系模型，这是关系数据库逻辑设计所要解决的问题。这就要求关系数据库的设计必须遵循关系数据库的规范化理论。

5.1.2 关系模式存在的问题

想要设计一个相对较好的关系数据库，规范化理论是必须遵循的。关系数据库的设计最重要的是关系模式的设计，那么什么是一个较好的关系模式？一个不好的关系模式又会

存在什么样的问题？

【例 5.1】要求设计学生-课程数据库，其关系模式如下：SDC (SNO, SNAME, AGE, DEPT, DEAN, CNAME, SCORE)，其中，SNO 为学号，SNAME 为姓名，AGE 为年龄，DEPT 为系别，DEAN 为系主任，CNAME 为课程名，SCORE 为成绩。具体内容如表 5.1 所示。

根据实际情况，这些数据有如下语义规定。

- (1) 一个系有若干名学生，但一名学生只属于一个系；
- (2) 一个系只有一名系主任，系主任不可以兼任；
- (3) 一名学生可以选修多门课程，每门课程可被多名学生选修。

表 5.1 关系模式 SDC 对应的部分表内容

学号 SNO	姓名 SNAME	年龄 AGE	系别 DEPT	系主任 DEAN	课程名 CNAME	成绩 SCORE
721011	程民	20	计算机	刘华	计算机应用基础	78
721011	程民	20	计算机	刘华	数据库原理	82
721032	李顺	23	电子	李国义	高频技术	67
722010	王小平	22	自动化	王健	高电压	75
722010	王小平	22	自动化	王健	过程控制	60
722010	王小平	22	自动化	王健	数据库原理	82
722131	刘婷	20	计算机	刘华	C++	77
723011	张小惠	19	自动化	王健	计算机应用基础	74
723011	张小惠	19	自动化	王健	高电压	68
723015	熊民	20	计算机	刘华	计算机应用基础	70
723015	熊民	20	计算机	刘华	C++	50
722017	胡丽文	22	电子	李国义	高频技术	82
722017	胡丽文	22	电子	李国义	通信原理	69
721109	王少国	23	计算机	刘华	数据库原理	65

分析可得，(SNO, CNAME) 属性的组合可以唯一标识一个元组，即每行的 SNO 与 CNAME 组合都是不同的，因此，(SNO, CNAME) 是该关系模式的候选键，且为主键（只有一个候选键）。但在实际操作数据库时，将会出现以下几种问题。

1. 数据冗余

每名学生的信息如姓名和年龄重复存储，选修几门课程就要重复存储几次；每个系的名称和系主任的名字存储的次数等于该系的学生选修课程门数的累加和，也存在重复存储的问题，数据冗余很大，极大地浪费了存储空间。

2. 操作异常

(1) 插入：若一个系里的学生尚未选修课程，则不能进行插入操作。因为 (SNO, CNAME) 是该关系模式的主码，根据关系的实体完整性规则，主码的值要求不能全部或部分为空，而由于主码中的 CNAME 部分为空，所以学生的相关基本信息无法插入到数据库中。

(2) 修改：若某位学生改名了，则所有相关的记录都要逐一修改 SNAME 的值；若某个系的主任改变了，则属于该系的学生记录都要修改 DEAN 的值。由于本身存在数据冗余

的问题，修改量将会特别大，稍有不慎，就很有可能漏改或错改某些内容，造成数据上的不一致，破坏数据的完整性。

(3) 删除：若某个系的学生全部毕业了，本应该只是删除学生的记录，由于 SNO 是主键的一部分，为保证实体完整性，需将整个元组一起删除，这样，系的有关信息也将删除。

由于该关系中包含的内容太多、太杂了，因此会存在上述一些问题。由此得出结论，SDC 不是一个好的关系模式。那么怎样才能得到一个好的关系模式呢？将 TDC 分解为三个关系：学生关系 S (SNO, SNAME, AGE, DEPT)，系关系 D (DEPT, DEAN) 和选修关系 SC (SNO, CNAME, SCORE)，如图 5.1 所示。

S				SC		
学号 SNO	姓名 SNAME	年龄 AGE	系别 DEPT	学号 SNO	课程名 CNAME	成绩 SCORE
721011	程民	20	计算机	721011	计算机应用基础	78
721032	李顺	23	电子	721011	数据库原理	82
722010	王小平	22	自动化	721032	高频技术	67
722131	刘婷	20	计算机	722010	高电压	75
723011	张小惠	19	自动化	722010	过程控制	60
723015	熊民	20	计算机	722010	数据库原理	82
722017	胡丽文	22	电子	722131	C++	77
721109	王少国	23	计算机	723011	计算机应用基础	74
D				723011	高电压	68
系别 DEPT	系主任 DEAN			723015	计算机应用基础	70
计算机	刘华			723015	C++	50
电子	李国义			722017	高频技术	82
自动化	王健			722017	通信原理	69
				721109	数据库原理	65

图 5.1 分解后的三个关系 S 、 D 和 SC

在三个关系中，实现了信息在某种程度上的分离， S 中存储学生的基本信息，与系主任和所选修课程无关； D 中存储系的有关信息，与学生和课程信息无关； SC 中存储学生选修课程的情况，而与学生和系的有关信息无关。它们与 SDC 相比，数据的冗余情况明显减少。即使学生不选修课程，他的信息也能正常插入 S 中，这就避免了插入异常。由于数据的冗余度低，因此也不会引起修改异常的问题。当某位学生只选修了一门课程，而这门课程暂时不开设了，只需在 SC 关系中进行相关的删除，而不会造成其他信息的丢失，这就解决了删除异常的问题。

综上所述，分解后的关系模式是一个较好的数据库模式。三个关系模式极好地降低了数据的冗余程度，也不会发生插入、修改和删除的操作异常问题。但是，一个好的关系模式并不是在任何时候都是最好的，应该根据实际应用系统的需求进行设计。例如，若想知道某位学生所在系的主任和其选修情况，可将三个表进行联接后进行查询，而联接操作所需的系统开销是非常大的。

另一方面，关系模式中的属性之间存在相互制约、相互依赖的关系，它们直接决定着

关系模式的好坏。因此，必须借助理论依据，根据实际情况，从语义上分析属性间的制约和依赖关系，将不好的关系数据库模式转变为较好的关系数据库模式，即进行关系的规范化。

5.2 规范化理论

规范化理论的基本思想是通过合理的分解关系模式消除其中不合适的数据依赖，解决数据冗余、修改异常、插入异常、删除异常的问题，使模式中的各关系模式达到某种程度的分离。

关系数据库中的数据依赖分为函数依赖（Functional Dependency, FD）、多值依赖（Multivalued Dependency, MVD）和联接依赖（Join Dependency, JD）。其中，函数依赖最为重要。

5.2.1 函数依赖

1. 定义

设关系模式 $R(U)$ ， x 、 y 是 U 的子集。若对于 $R(U)$ 上的任何一个可能关系，均有 x 的一个值对应于 y 的唯一具体值，称为 y 函数依赖于 x 或者 x 函数决定 y ，记作： $x \rightarrow y$ 。其中， x 称为决定因素， y 称为依赖因素。进而，若再有 $y \rightarrow x$ ，则称 x 与 y 相互依赖，记作： $x \leftrightarrow y$ 。当 y 不依赖于 x 时，记作： $x \nrightarrow y$ 。

例如，前面讲到的学生-课程数据库中的关系模式 SC (SNO, SNAME, AGE, DEPT, DEAN, CNAME, SCORE)，根据它们的语义定义可以写出该关系模式所有的函数依赖（即函数依赖集 F ）。

$F = \{SNO \rightarrow SNAME, SNO \rightarrow AGE, SNO \rightarrow DEPT, SNO \rightarrow DEAN, DEPT \rightarrow DEAN, (SNO, CNAME) \rightarrow SCORE\}$

由于一个 SNO 有多个 CNAME 的值与之对应，CNAME 不能唯一被确定，也就是 CNAME 不能依赖于 SNO，因此有 $SNO \nrightarrow CNAME$ ，同理有 $SNO \nrightarrow SCORE$ 。

而 (SNO, CNAME) 属性的组合是该关系模式的主键，可以唯一标识一个元组，所以有 $(SNO, CNAME) \rightarrow SCORE$ 。

2. 分类

(1) 部分与完全函数依赖。

设关系模式 $R(U)$ ， x 、 y 是 U 的子集， x' 是 x 的任意一个真子集，若 $x \rightarrow y$ 并且 $x' \nrightarrow y$ ，则称 y 部分函数依赖（Partial Functional Dependency）于 x ，记作 $x \xrightarrow{p} y$ 。若 $x \rightarrow y$ 并且 $x' \nrightarrow y$ ，则称 y 完全函数依赖（Full Functional Dependency）于 x ，记作 $x \xrightarrow{f} y$ 。

例如，在关系模式 SDC 中，因为 $SNO \rightarrow SNAME$ ，所以 $(SNO, CNAME) \xrightarrow{p} TNAME$ 。又因为 $SNO \nrightarrow SCORE$ 且 $CNAME \nrightarrow SCORE$ ，所以 $(SNO, CNAME) \xrightarrow{f} SCORE$ 。

显然，当且仅当决定因素为属性组时，才有可能出现部分函数依赖。完全函数依赖说明在依赖关系的决定因素中没有多余属性，有多余属性就是部分函数依赖。

(2) 传递与直接函数依赖。

设关系模式 $R(U)$ ， x 、 y 、 z 是 U 的子集，若 $x \rightarrow y$ ， y 又不包含于 x ，且 $y \nrightarrow x$ ，但 $y \rightarrow$

z ，则称 z 传递函数依赖 (Transitive Functional Dependency) 于 x ，记作 $x \twoheadrightarrow z$ 。

如果有 $y \rightarrow x$ ，则 $x \leftrightarrow y$ ，此时称 z 直接函数依赖 (Direct Functional Dependency) 于 x ，而不是传递函数依赖。

例如，在关系模式 TDC 中，因为 $SNO \rightarrow DEPT$ ，但 $DEPT \nrightarrow SNO$ ，而 $DEPT \rightarrow DEAN$ ，则有 $SNO \xrightarrow{t} DEAN$ 。若学生不存在同名时，则有 $SNO \leftrightarrow SNAME$ ， $SNAME \rightarrow DEPT$ ，此时 DEPT 对 SNO 是直接函数依赖，而不是传递函数依赖。

(3) 平凡与非平凡函数依赖。

若属性集 y 是属性集 x 的子集，则必有函数依赖 $x \twoheadrightarrow y$ ，称其为平凡函数依赖。如果 y 不是 x 的子集，若有 $x \rightarrow y$ ，则称其为非平凡函数依赖。一般不特别声明，都是指非平凡函数依赖。

$(SNO, CNAME) \rightarrow SNO$ 为平凡函数依赖； $(SNO, CNAME) \rightarrow SCORE$ 则为非平凡函数依赖。

3. 基本性质

(1) 扩张性

两个函数依赖的决定因素与依赖因素分别合并后，依然保持函数依赖关系。假设 $a \rightarrow c$ ，并且 $b \rightarrow d$ ，则 $(a, b) \rightarrow (c, d)$ 。

(2) 投影性

根据平凡函数依赖的定义，一组属性函数决定它的所有子集。 $(a, b) \rightarrow a$ ， $(a, b) \rightarrow b$ 。

(3) 合并性

把决定因素相同的两个函数依赖中的依赖因素进行合并后，依然保持函数依赖关系。假设 $a \rightarrow b$ 且 $a \rightarrow c$ ，则必有 $a \rightarrow (b, c)$ 。

(4) 分解性

决定因素能够决定全部，当然也能够决定全部中的部分。分解性和合并性互为逆过程。假设 $a \rightarrow (b, c)$ ，则 $a \rightarrow b$ 并且 $a \rightarrow c$ 。

属性间的三种联系实际上是属性值之间相互依赖又相互制约的反映，称为属性间的数据依赖。那么函数依赖与属性间的联系类型是有关系的，分别为：

① 如果属性 x 与属性 y 的联系类型是一对一时，则存在函数依赖 $x \rightarrow y$ ， $y \rightarrow x$ ，即 $x \leftrightarrow y$ 。例如，当学生没有同名时，则有 $SNO \leftrightarrow SNAME$ 。

② 如果属性 x 与属性 y 的联系类型是一对多时，则只存在函数依赖 $y \rightarrow x$ 。例如，SNO 与 AGE，DEPT 之间均为多对一的联系类型，所以有 $SNO \rightarrow AGE$ ， $SNO \rightarrow DEPT$ 。

③ 如果属性 x 与属性 y 的联系类型是多对多时，则 x 与 y 之间不存在任何函数依赖关系。例如，一名学生可以选修多门课程，一门课程又可以被多名学生选修，所以 SNO 与 CNAME 之间不存在任何函数依赖关系。

由于函数依赖与属性间的联系类型有关，因此要确定属性间的函数依赖，应该从属性间的联系类型开始分析，进而确定属性间的函数依赖。

要证明一个函数依赖是否成立，必须根据其语义进行分析，而不能只是依照其形式化的定义。例如，在关系模式 SDC 中，只有在学生不存在同名的情况下，才有 $SNAME \rightarrow AGE$ ， $SNAME \rightarrow SNO$ 。否则，这些函数依赖就不成立了。因此，函数依赖是语义范畴的概念，反

映一种语义的完整性约束。

由于函数依赖是关系中的所有元组，而不仅是关系中的某个或某些元组应该满足的约束条件，因此，当关系进行元组的插入、修改或删除操作后都不能违背这种函数依赖。

必须根据语义来确定属性间的函数依赖，而不能仅凭某一时刻的数据来判断。所以说函数依赖的存在与时间没有关系，而只与数据间的语义有关系。

例如，在关系模式 SDC 中，若没有给出“不存在同名的学生”这种语义规定，即使当前关系中没有同名的元组，也只能存在函数依赖 $SNO \rightarrow SNAME$ ，而不能存在函数依赖 $SNAME \rightarrow AGE$ ，因为若增加一名同名的学生，函数依赖 $SNAME \rightarrow AGE$ 必然不成立。

5.2.2 码

1. 候选码

(1) 定义。

设 K 为 $R(U, F)$ 中的属性或属性组，若 $K \xrightarrow{f} U$ ，则 K 称为 R 的候选码（候选键或候选关键字）。若一个关系模式的候选码不只一个，则选择其中的一个为主码（主键）。包含在任何一个候选码中的属性称为主属性。不包含在任何一个候选码中的属性称为非主属性（非码属性）。若候选码包含所有的属性，则称为全码（全键）。第 3 章已经介绍过相关内容，这里就不再举例说明了。

(2) 确定。

① 观察函数依赖集 F ，看哪些属性在依赖因素中没有出现过。设没出现过的属性集为 K' 。若 K' 为空集，转到步骤④；若 K' 不为空集，转到步骤②。

② 根据候选码的定义，其中必定包含 K' ，因为没有其他属性集能决定 K' 。观察 K' ，如有 $K' \xrightarrow{f} U$ ，则 K' 为候选码，转到步骤⑤，否则转到步骤③。

③ K' 可以分别与 $\{U - K'\}$ 中的每一个属性组合成新的属性集，观察哪个属性集能够完全决定 U ，继而找到候选码。若合并一个属性不能找到或者不能找全候选码，可以将 K' 分别与 $\{U - K'\}$ 中的每两个（三个，四个，……）属性组合成新的属性集，进行类似的判断，直到找全所有的候选码。转到步骤⑤。

④ 假如 K' 为空集，则先观察 F 中的每个决定因素。若某个决定因素能够完全决定 U ，则其即为候选码。若不能够完全决定 U ，则将决定因素分为两个或多个组合，观察哪些组合能够完全决定 U ，继而找到其他的候选码。转到步骤⑤。

⑤ 结束。

【例 5.2】 设有关系模式 $R(U, F)$ ，其中， $U = \{A, B, C, D\}$ ， $F = \{AB \rightarrow C, D \rightarrow B, C \rightarrow AD\}$ ，求 R 的所有候选码。

观察函数依赖集 F ，所有属性都在依赖因素中出现。转到步骤④。

观察 F 中的每个决定因素： AB 、 D 、 C ，因为存在 $(A, B) \xrightarrow{f} U$ 和 $C \xrightarrow{f} U$ ，所以 (A, B) 和 C 皆为候选码。转到步骤⑤。结束。

【例 5.3】 设有关系模式 $R(U, F)$ ，其中， $U = \{A, B, C, D, E\}$ ， $F = \{AB \rightarrow C, B \rightarrow DE, D \rightarrow B\}$ ，求 R 的所有候选码。

观察函数依赖集 F ，只有 A 属性没有在依赖因素中出现。转到步骤②。

由于 A 不能完全函数决定 U ，因此转到步骤③。

将 A 分别与 B 、 C 、 D 、 E 组成新的属性集，因为存在 $(A, B) \xrightarrow{f} U$ 和 $(A, D) \xrightarrow{f} U$ ，所以 (A, B) 和 (A, D) 皆为候选码。转到步骤⑤。结束。

2. 外码

外码的概念在第 3 章中已叙述过，这里再总结如下。

F 为关系模式 R 中的属性或属性组，若其不是 R 的主码，但却是另外一个关系模式 S 的主码，则称 F 是 R 的外码或外键。

例如，在学生关系 $S(SNO, SNAME, AGE, DEPT)$ 和选修关系 $SC(SNO, CNAME, SCORE)$ 中， SNO 不是关系 SC 的主码，但它却是 S 的主码，则称 S 中的 SNO 为 SC 的外码。

5.2.3 范式

设计关系数据库中的关系模式必须遵循一定的规则，这种规则就是范式(Normal Form, NF)。关系数据库中的关系必须满足一定的要求，即满足不同的范式。

范式的概念最早是由 E.F.Codd 提出的，从 1971 年起相继提出了关系的三级规范化形式，它们是第 1 范式(1NF)、第 2 范式(2NF)和第 3 范式(3NF)。1974 年，E.F.Codd 和 Boyce 共同提出了一个新的范式概念——Boyce-Codd 范式(BC 范式)。1976 年，Fagin 提出了第 4 范式(4NF)，后来又有人提出了第 5 范式(5NF)。至此，在关系数据库规范中建立了一个范式系列：1NF、2NF、3NF、BCNF、4NF 和 5NF。它们一级比一级有更为严格的要求，满足最低要求的范式是第 1 范式，在第 1 范式的基础上进一步满足要求的称为第 2 范式(2NF)，其余范式以此类推。

范式是符合某一种级别的关系模式的集合。各范式之间的集合关系可以表示为： $1NF \supset 2NF \supset BCNF \supset 3NF \supset 4NF \supset 5NF$ ，如图 5.2 所示。

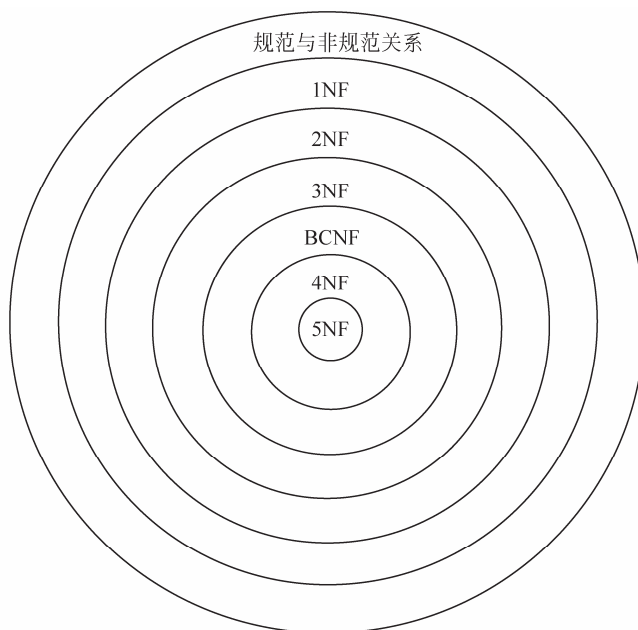


图 5.2 各范式之间的关系

一个较低范式的关系，可以通过关系的分解转换为若干个较高级范式关系的集合，这一过程就叫作关系的规范化。规范化的目的就是使结构更为合理，消除存储异常，使数据冗余度尽量小，便于插入、删除和更新操作。

1. 第 1 范式

若关系模式 R 的每个属性都是不可再分的数据项，也就是每个属性不能有多值或者不能有重复的属性，则称 R 属于第 1 范式，记作 $R \in 1NF$ 。

在任何一个关系数据库中，第 1 范式是对关系模式的基本要求。由于在关系数据库中只讨论规范化的关系，因此所有非规范化的关系模式必须转换成规范化的关系。去掉非规范化关系中的组合项就能将其转换成规范化的关系。每个规范化的关系都是属于 1NF。

【例 5.4】 设计员工信息表的结构，应该有员工号、姓名、教育经历等字段，其中，教育经历分为小学、高中、大学，将它规范成 1NF。

第 1 种方法：员工号为候选码，不要“教育经历”这个属性，直接把三个子属性作为实体的属性。关系模式为：员工信息表（员工号，姓名，小学，高中，大学）。

第 2 种方法：（员工号，教育经历）为候选码，重复存储员工号和姓名，依次填入小学、高中、大学的教育经历。关系模式为：员工信息表（员工号，姓名，教育经历）。

第 3 种方法：员工号为候选码，强制每个员工只填最高学历的教育经历。关系模式为：员工信息表（员工号，姓名，教育经历）。

第 1 范式要求每个属性都是不可再分的数据项，即解决了“表中表”的问题。

2. 第 2 范式

若关系模式 R 属于第 1 范式，并且它的每个非主属性都完全函数依赖于任何一个候选码，则称 R 属于第 2 范式，记作 $R \in 2NF$ 。

第 2 范式是在第 1 范式的基础上建立起来的，根据定义可知，第 2 范式就是不存在非主属性部分依赖于某一候选码。如果 R 的候选码均为单属性，或者 R 的全体属性均为主属性，那么 R 属于 2NF。

【例 5.5】 在关系模式 SDC 中，（SNO，CNAME）为候选码，则 SNO、CNAME 为主属性，SNAME、AGE、DEPT、DEAN 和 SCORE 均为非主属性。

函数依赖关系有：

$SNO \rightarrow SNAME$, $(SNO, CNAME) \xrightarrow{p} SNAME$

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

$SNO \rightarrow DEPT$, $(SNO, CNAME) \xrightarrow{p} DEPT$, $DEPT \rightarrow DEAN$

$SNO \rightarrow DEAN$, $(SNO, CNAME) \xrightarrow{p} DEAN$, $(SNO, CNAME) \xrightarrow{f} SCORE$

它们之间的函数依赖关系用函数依赖图表示，如图 5.3 所示。

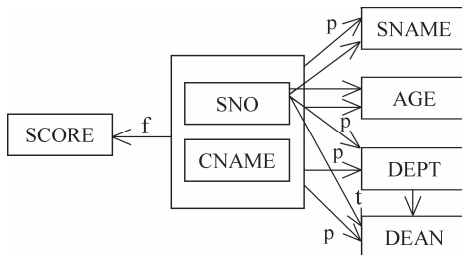


图 5.3 TDC 中的函数依赖关系

经上述分析, 存在非主属性对候选码的部分函数依赖, 所以 $SD \notin 2NF$ 。而且还存在完全函数依赖和传递函数依赖。这种情况正是因为关系中存在着复杂的函数依赖, 所以引起数据冗余和操作异常等问题。

为此将其规范化, 消除非主属性对候选码的部分函数依赖, 使其转换成为 2NF。方法就是要求一个关系只描述一个实体或者实体间的联系, 若多于一个实体或联系, 则将其进行投影分解。

根据此方法, 将 SDC 分解为两个关系模式 SD (SNO, SNAME, AGE, DEPT, DEAN) 和 SC (SNO, CNAME, SCORE), SD 描述学生实体, SC 描述学生与课程的联系。其中, SD 的候选码为 SNO, SNO 是单属性, 不可能存在部分函数依赖; SC 的候选码为 (SNO, CNAME), 也不存在非主属性对候选码的部分函数依赖。因此 SDC 分解后, SD 和 SC 均属于 2NF。

分解后的两个关系通过 TNO 相联系, 需要时可以进行自然联接, 恢复成原来的关系, 这种分解不会丢失原有的信息。

3. 第 3 范式

若关系模式 R 不存在这样的候选码 X 、非主属性 Z , 使得 $X \xrightarrow{f} Z$ 成立, 则称 R 属于第 3 范式, 记作 $R \in 3NF$ 。

若关系模式属于第 3 范式, 则它也属于第 2 范式。但关系模式若属于第 2 范式, 它不一定属于第 3 范式。

【例 5.6】 在关系模式 SD ($SD \in 2NF$) 中, SNO 为候选码, 则 SNO 为主属性, SNAME、AGE、DEPT 和 DEAN 均为非主属性。

函数依赖关系有:

$SNO \xrightarrow{f} SNAME$

$SNO \xrightarrow{f} AGE$

$SNO \xrightarrow{f} DEPT, DEPT \rightarrow DEAN$

$SNO \xrightarrow{t} DEAN$

它们之间的函数依赖关系用函数依赖图表示, 如图 5.4 所示。

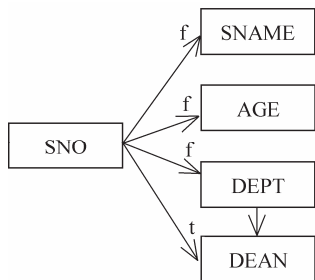


图 5.4 TD 中的函数依赖关系

经上述分析, 存在非主属性对候选码的传递函数依赖, 所以 $SD \notin 3NF$ 。虽然 2NF 的关系模式解决了 1NF 中存在的弊端, 但是 2NF 的关系模式 TD 在进行数据操作时, 仍然会存在数据冗余和操作异常的问题。存在这些问题的关键是由于在 TD 中存在着非主属性对候

选码的传递函数依赖。

为此将其规范化，消除非主属性对候选码的传递函数依赖，使其转换成为 3NF。方法与第 2 范式的规范化一样。

根据此方法，将 SD 分解为两个关系模式 $S(SNO, SNAME, AGE, DEPT)$ 和 $D(DEPT, DEAN)$ ， S 描述学生实体， D 描述系实体。其中， S 的候选码为 SNO， D 的候选码为 DEPT，都不存在非主属性对候选码的传递函数依赖。因此 TD 分解后， S 和 D 均属于 3NF。

如果一个关系数据库中所有的关系模式都属于 3NF，则已经在很大程度上消除了插入异常和删除异常，但由于 3NF 只是限制非主属性对候选码的函数依赖，并没有限制主属性对候选码的函数依赖，因此，关系模式的分离仍然不够彻底，需要对 3NF 进行规范化，向更高一级的 BC 范式进行转换。

4. BC 范式

若关系模式 R 属于第 1 范式，如果对于 R 的每个函数依赖 $X \rightarrow Y$ ($Y \notin X$)， X 都含有候选码，则称 R 属于 BC 范式，记作 $R \in BCNF$ 。

BCNF 通常被认为是修正的 3NF，它是在满足 1NF 的基础上，没有任何属性传递依赖于任意一个候选码。等价于满足第 3 范式且主属性与码之间不存在依赖关系。

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) 所有的主属性对每一个不包含它的候选码都是完全函数依赖。
- (2) 所有非主属性对每一个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。

若关系模式属于 BC 范式，则它也属于第 3 范式。但关系模式若属于第 3 范式，它不一定属于 BC 范式。

【例 5.7】 设有关系模式 SCR ($SNO, SNAME, COMPETITON, TIME, RANKING$)，其中， SNO 为学号， $SNAME$ 为学生姓名， $COMPETITON$ 为比赛名称， $TIME$ 为参加时间， $RANKING$ 为比赛名次。每个学生可以参加若干个比赛，每次比赛有若干名学生参加，学生参加某个比赛有一个名次。若不存在同名的学生，则 ($SNO, COMPETITON, TIME$) 和 ($SNAME, COMPETITON, TIME$) 皆为候选码， SNO 、 $SNAME$ 、 $TIME$ 和 $COMPETITON$ 为主属性， $RANKING$ 为非主属性。

函数依赖关系有：

$SNO \leftrightarrow SNAME$ ，
 $(SNO, COMPETITON, TIME) \xrightarrow{p} SNAME$
 $(SNAME, COMPETITON, TIME) \xrightarrow{p} SNO$
 $(SNO, COMPETITON, TIME) \rightarrow RANKING$
 $(SNAME, COMPETITON, TIME) \rightarrow RANKING$

经上述分析，唯一的非主属性 $RANKING$ 对候选码，既不存在部分函数依赖也不存在传递函数依赖，所以 $SCR \in 3NF$ 。但由于 $SNO \leftrightarrow SNAME$ ，即决定因素 SNO 或 $SNAME$ 不包含候选码，因此 $SCR \notin BCNF$ 。

属于 3NF 的关系模式 SCR 在进行数据操作时，仍然会存在数据冗余和操作异常的问题。存在这些问题的关键是由于在 SCR 中存在着主属性对候选码的部分函数依赖。

为此将其规范化，消除主属性对候选码的部分函数依赖，解决这一问题的办法仍然是

通过投影分解，使其转换为 BCNF。

根据此方法，将 SCR 分解的两个关系模式 $S(SNO, SNAME)$ 和 $SR(SNO, COMPETITON, TIME, RANKING)$ ， S 描述学生实体， SR 描述学生与比赛的联系。其中， S 的候选码为 SNO 和 $SNAME$ ， SR 的候选码为 $(SNO, COMPETITON, TIME)$ ，函数依赖中的所有决定因素都包含一个候选码，即无论是主属性还是非主属性都不存在其对候选码的部分和传递函数依赖。因此 SCR 分解后， S 和 SR 均属于 BCNF。

如果一个关系数据库中所有的关系模式都属于 BCNF，那么在函数依赖的范畴内，已经实现了模式的彻底分解，消除了产生插入异常、修改异常和删除异常的根源，而且数据冗余也减少到极小程度。

5. 多值依赖

前面介绍的范式都是依据函数依赖而定义的。函数依赖只能表示关系模式中属性之间一对一或一对多的联系，而对于多对多的联系，需通过多值依赖来描述。

【例 5.8】 设有关系模式专业必修课管理（专业号，学生，必修课），其中一个专业有若干名学生学习，一名学生只属于一个专业，他们学习所在专业的所有必修课，如表 5.2 所示。

表 5.2 关系模式专业必修课管理对应的部分表内容

专 业 号	学 生	必 修 课
1	赵明	法理学
1	赵明	宪法
1	赵明	法制史
2	李蕾	基础会计
2	李蕾	财务管理
2	刘艳	基础会计
2	刘艳	财务管理
3	钱明建	高等数学
3	钱明建	线性代数
3	钱明建	高等代数
3	熊勇进	高等数学
3	熊勇进	线性代数
3	熊勇进	高等代数
3	罗兰	高等数学
3	罗兰	线性代数
3	罗兰	高等代数

可以得出，该关系模式只有一个函数依赖（学生，必修课）→专业号，候选码为（学生，必修课），因此它属于 BCNF。

由于对于关系中的一个具体专业号来说，有多个学生值与其对应，专业号与必修课也存在着类似的联系；并且对于关系中的一个具体专业号来说，有一组与学生无关的必修课与之对应。因此，进一步分析可以看出，它还存在着数据冗余和操作异常的现象。

通过上述两方面的原因可以看出，专业号与学生之间的依赖关系并不是函数依赖，为此提出多值依赖的概念。

(1) 多值依赖的定义。

设关系模式 $R(U)$, x, y, z 是 U 的子集, $z=U-x-y$ 。若对于 $R(U)$ 的任一关系 r , 给定的一个 (x, z) 值, 存在一组 y 的值与之对应, 并且这组值仅决定于 x 值而与 z 值无关, 称为 y 多值依赖于 x 或者 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。

例如, 在关系模式专业必修课管理中, 对于某一专业号、必修课属性值组合如 (2, 基础会计) 来说, 有一组学生值 {李蕾, 刘艳} 与之对应, 这组值仅由专业号上的值 (2) 决定。也就是说, 对于另一个专业号、必修课属性值组合如 (2, 财务管理), 它对应的一组管理员值仍是 {李蕾, 刘艳}, 尽管这时必修课的值已经改变了。因此必修课多值依赖于专业号, 即: 专业号 \twoheadrightarrow 学生。

(2) 多值依赖与函数依赖的区别。

① 在函数依赖中, $x \rightarrow y$ 的有效性仅由 x, y 这两个属性集决定, 不涉及第三个属性集, 而在多值依赖中, 判定 $x \twoheadrightarrow y$ 在属性集 $U(z=U-x-y)$ 上是否成立, 不仅要检查 x, y 上的值, 而且要检查 U 的其余属性 z 上的值。因此, 多值依赖的有效性与属性集的范围有关。

若 $x \twoheadrightarrow y$ 在 R 上成立, 在属性集 $W(U \supset W)$ 上也成立, 则称 $x \twoheadrightarrow y$ 为 $R(U)$ 的嵌入型多值依赖。

② 若函数依赖 $x \rightarrow y$ 在 $R(U)$ 上成立, 则对于 y 的任一子集 y' 均有 $x \rightarrow y'$ 成立。而多值依赖 $x \twoheadrightarrow y$ 在 $R(U)$ 上成立, 却不能确定 $x \twoheadrightarrow y'$ 成立。

(3) 多值依赖的性质。

① 对称性。如果 $x \twoheadrightarrow y$, 则 $x \twoheadrightarrow z$, 其中 $z=U-x-y$ 。

② 传递性。如果 $x \twoheadrightarrow y, y \twoheadrightarrow z$, 则 $x \twoheadrightarrow (z-y)$ 。

③ 伪传递性。如果 $x \twoheadrightarrow y, wy \twoheadrightarrow z$, 则 $wx \twoheadrightarrow (z-wy)$ 。

④ 合并性。如果 $x \twoheadrightarrow y, x \twoheadrightarrow z$, 则 $x \twoheadrightarrow yz$ 。

⑤ 分解性。如果 $x \twoheadrightarrow y, x \twoheadrightarrow z$, 则 $x \twoheadrightarrow (y \cap z), x \twoheadrightarrow (y-z), x \twoheadrightarrow (z-y)$ 。

⑥ 增广性。如果 $x \twoheadrightarrow y$, 且 $v \in w$, 则 $wx \twoheadrightarrow vy$ 。

⑦ 从函数依赖导出多值依赖: 如果 $x \rightarrow y$, 则 $x \twoheadrightarrow y$ 。

⑧ 从多值依赖导出函数依赖: 如果 $x \twoheadrightarrow y, z \in y, y \cap w = \phi, w \rightarrow z$, 则 $x \rightarrow z$ 。

6. 第4范式

若关系模式 R 属于第1范式, 如果对于 R 的每个非平凡多值依赖 $X \twoheadrightarrow Y, X$ 都含有候选码, 则称 R 属于第4范式, 记作 $R \in 4NF$ 。

在例 5.8 中, 已经分析了专业必修课管理关系模式属于 BCNF, 它的数据依赖有 (学生, 必修课) \rightarrow 专业号和专业号 \twoheadrightarrow 学生。对于专业号 \twoheadrightarrow 学生这个非平凡多值依赖, 决定因素没有包含候选码, 所以专业必修课管理 $\notin 4NF$ 。在对该关系进行数据操作时, 仍然会存在数据冗余和操作异常的问题。存在这些问题的关键是由于在专业必修课管理中存在着非平凡多值依赖。

为此将其规范化, 消除非平凡多值依赖, 使其转换成为 4NF。解决这一问题的办法仍然是通过投影分解, 使其转换成为 4NF。

根据此方法, 将专业必修课分解的两个关系模式专业必修课 1 (专业号, 学生) 和专

业必修课 2（专业号，必修课）。它们分别有一个多值依赖：专业号 \twoheadrightarrow 学生和专业号 \twoheadrightarrow 必修课，这样它们都不存在非平凡多值依赖。因此分解后，专业必修课 1 和专业必修课 2 均属于 4NF。

经过上面的分析可以得知：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 的定义。

当然还有更高级的范式，比如 5NF。如果消除了属于 4NF 的关系模式中存在的联接依赖，则可以进一步达到 5NF。本书将不再讨论 4NF 和 5NF 这方面的内容，有兴趣的读者可以参阅相关书籍。

数据依赖中除了两种最重要的函数依赖和多值依赖，还有联接依赖。如果考虑函数依赖，则属于 BCNF 的关系模式的规范化程度是最高的；如果考虑多值依赖，则属于 4NF 的关系模式的规范化程度是最高的。函数依赖是多值依赖的一种特殊情况，而多值依赖又是联接依赖的一种特殊情况。但联接依赖不像函数依赖和多值依赖那样可以由语义直接导出，而是在关系的联接运算时才反映出来的。

虽然提高数据库的范式级别，有利于在设计的层面上消除数据库操作异常，但是考虑到分解带来数据库表之间的联接代价和可能的联接损失，所以也不必盲目追求高级别的范式，可根据应用的需要而定。

5.3 Armstrong 公理系统

Armstrong 公理系统是有效而完备的公理系统，它其中的一些推理规则是关系模式分解算法的理论基础。本节主要介绍公理系统推理规则、属性集的闭包概念、最小函数依赖集的分析方法和模式设计的原则。

5.3.1 Armstrong 公理系统推理规则

从已知的一些函数依赖，可以推导出另外一些函数依赖，这就需要一系列推理规则。W.W.Armstrong 于 1974 年最早提出了一些函数依赖的推理规则，这些规则常常被称作 Armstrong 公理。

设关系模式 $R(U, F)$ ，其中， U 是属性全集， F 是 U 上的一组函数依赖，有以下的推理规则。

A1 自反律：若属性集 Y 包含于属性集 X ，属性集 X 又包含于 U ，则 $X \rightarrow Y$ 在 R 上成立。

A2 增广律：若 $X \rightarrow Y$ 在 R 上成立，且属性集 Z 包含于属性集 U ，则 $XZ \rightarrow YZ$ 在 R 上成立。

A3 传递律：若 $X \rightarrow Y$ 和 $Y \rightarrow Z$ 在 R 上成立，则 $X \rightarrow Z$ 在 R 上也成立。

A4 伪传性：若 $X \rightarrow Y$ ，且 $YW \rightarrow Z$ ，则 $XW \rightarrow Z$ 。

A5 合成性：若 $X \rightarrow Y$ ，且 $X \rightarrow Z$ ，则 $X \rightarrow YZ$ 。

A6 分解性：若 $X \rightarrow Y$ ，且属性集 Z 包含于属性集 Y ，则 $X \rightarrow Z$ 。

通常把自反律、增广律和传递律称为 Armstrong 公理系统。由于 R 根据 Armstrong 公理系统推导出来的每个函数依赖一定也是在 R 上成立的，因此称 Armstrong 公理系统是有

效的。又由于其他所有函数依赖的推理规则可以使用这三条规则推导出，因此称 Armstrong 公理系统是完备的。总之，Armstrong 公理系统是有效的、完备的。

5.3.2 属性集的闭包

设有关系模式 $R(U, F)$ ，其中， U 为属性全集， X 是 U 的子集， F 为 R 的函数依赖集，则由 Armstrong 公理推导出的所有函数依赖中的依赖因素（右部）所形成的属性集，称为属性集 X 关于函数依赖集 F 的闭包，记作 $(X)_F^+$ 。

下面介绍求解 $(X)_F^+$ 的算法。

(1) 将 X 置入 $(X)_F^+$ 中，即 $(X)_F^+ = X$ 。

(2) 对于 F 中的每一个函数依赖 FD，若决定因素（左部）属于 $(X)_F^+$ ，则将依赖因素（右部）置入 $(X)_F^+$ 中，即 $(X)_F^+ = X \cup$ 依赖因素。

(3) 重复第 2 步，直至 $(X)_F^+$ 不能再扩大。

【例 5.9】 设有关系模式 $R(U, F)$ ， $U = \{A, B, C, D\}$ ， $F = \{A \rightarrow C, AC \rightarrow B, D \rightarrow A, D \rightarrow C\}$ ，分别求属性集 BC、AD 和 AC 的闭包。

(1) $(BC)_F^+$

第一步：将 BC 置入 $(BC)_F^+$ 中，即 $(BC)_F^+ = BC$ 。

第二步：在 F 中再找不到某个函数依赖的决定因素属于 $(BC)_F^+$ ，因此 $(BC)_F^+ = BC$ 。

(2) $(AD)_F^+$

第一步：将 AD 置入 $(AD)_F^+$ 中，即 $(AD)_F^+ = AD$ 。

第二步：由于 $A \rightarrow C$ 的决定因素 A 属于 $(AD)_F^+$ ，则将依赖因素 C 置入 $(AD)_F^+$ 中，即 $(AD)_F^+ = AD \cup C = ACD$ 。

第三步：由于 $AC \rightarrow B$ 的决定因素 AC 属于 $(AD)_F^+$ ，则将依赖因素 B 置入 $(AD)_F^+$ 中，即 $(AD)_F^+ = ACD \cup B = ABCD = U$ ，因此 $(AD)_F^+ = ABCD$ 。

(3) $(AC)_F^+$

第一步：将 AC 置入 $(AC)_F^+$ 中，即 $(AC)_F^+ = AC$ 。

第二步：由于 $A \rightarrow C$ 的决定因素 A 属于 $(AC)_F^+$ ，将依赖因素 C 置入 $(D)_F^+$ 中并不能使其扩大。

第三步：由于 $AC \rightarrow B$ 的决定因素 AC 属于 $(AC)_F^+$ ，则将依赖因素 B 置入 $(AC)_F^+$ 中，即 $(AC)_F^+ = AC \cup B = ABC$ 。

而在 F 中再找不到某个函数依赖的决定因素属于 $(AC)_F^+$ ，因此 $(AC)_F^+ = ABC$ 。

5.3.3 最小函数依赖集

函数依赖集 F 中包含若干个函数依赖，为了得到最为精简的函数依赖集，应该去掉其中平凡的、无关的函数依赖和多余的属性。

如果函数依赖集 F 满足下列条件，那么 F 就是最小的，称为最小函数依赖集或最小覆盖，记作 F_m 。

(1) F 中的每一个函数依赖的依赖因素（右边）只含有单个属性。

(2) 每个函数依赖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。

(3) F 中没有冗余的函数依赖, 即在 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。

下面通过例题介绍求解最小函数依赖集的方法。

【例 5.10】 设有关系模式 $R(U, F)$, 其中, $U = \{A, B, C, D, E\}$, $F = \{AB \rightarrow C, CD \rightarrow BE, A \rightarrow C\}$, 求 F 的最小函数依赖集。

第一步: 将 F 中的所有的依赖因素转换为单个属性。

$$F_0 = \{AB \rightarrow C, CD \rightarrow B, CD \rightarrow E, A \rightarrow C\};$$

第二步: 去掉 F_0 中的所有决定因素的冗余属性。方法是在某个决定因素中去掉其中的一个属性, 看看是否依然能决定依赖因素。

(1) 对于 $AB \rightarrow C$, 若去掉 A , B 的闭包不含 C , 故 A 不是冗余属性, 不能去掉; 若去掉 B , A 的闭包包含 C , 故 B 是冗余属性, 可以去掉。

(2) 对于 $CD \rightarrow B$, 若去掉 D , C 的闭包不含 B , 故 D 不是冗余属性, 不能去掉; 若去掉 C , D 的闭包不包含 B , 故 C 也不是冗余属性, 不可以去掉。

(3) 对于 $CD \rightarrow E$, 若去掉 D , C 的闭包不含 E , 故 D 不是冗余属性, 不能去掉; 若去掉 C , D 的闭包不包含 E , 故 C 也不是冗余属性, 不可以去掉。

因此, $F_1 = \{A \rightarrow C, CD \rightarrow B, CD \rightarrow E, A \rightarrow C\}$ 是当前最为精简的函数依赖集。

第三步: 去掉 F_1 中的冗余函数依赖。

(1) 在 F_1 中去掉 $A \rightarrow C$, 得 $F_2 = \{CD \rightarrow B, CD \rightarrow E, A \rightarrow C\}$, $(A)_{F_2}^+ = AC$, 包含 C , 因此该函数依赖是冗余的, 可以从 F_1 中去掉。

(2) 在 F_2 中去掉 $CD \rightarrow B$, 得 $F_3 = \{CD \rightarrow E, A \rightarrow C\}$, $(CD)_{F_3}^+ = CDE$, 不包含 B , 因此该函数依赖不是冗余的, 不能从 F_2 中去掉。

(3) 在 F_2 中去掉 $CD \rightarrow E$, 得 $F_4 = \{CD \rightarrow B, A \rightarrow C\}$, $(CD)_{F_3}^+ = BCD$, 不包含 E , 该函数依赖不是冗余的, 不能从 F_2 中去掉。

(4) 在 F_2 中去掉 $A \rightarrow C$, 得 $F_5 = \{CD \rightarrow B, CD \rightarrow E\}$, $(A)_{F_5}^+ = A$, 不包含 C , 因此该函数依赖不是冗余的, 不能从 F_2 中去掉。

因此, $F_m = \{CD \rightarrow B, CD \rightarrow E, A \rightarrow C\}$ 。

可以得出, F 与它的最小函数依赖集是等价的。由于在求解过程中对属性和函数依赖的处理顺序的关系, 因此, 每个函数依赖集 F 不一定只有一个最小函数依赖集。

5.3.4 规范化模式设计的三个原则

1. 表达性

表达性涉及两个数据库模式的等价 (数据等价和依赖等价) 问题, 分别用无损联接性和保持函数依赖性来衡量。

关系模式的规范化过程是通过对关系模式的投影分解来实现的。由于投影分解的方法并不只一种, 因此不同的投影分解会得到不同的结果。

只有能够保证分解后的关系模式与原来的关系模式等价的方法才是有意义的。人们判

断对关系模式的一个分解是否与原关系模式等价要符合下面两个条件。

- (1) 分解要具有“无损联接性”。
- (2) 分解要具有“保持函数依赖性”。

如果一个分解具有无损联接性，则能够保证不丢失信息。如果一个分解具有保持函数依赖性，则保证不会破坏原来的语义，减轻或解决各种异常情况。无损联接性的判别方法如下。

$\rho = \{R_1 \langle U_1, F_1 \rangle, R_2 \langle U_2, F_2 \rangle, \dots, R_k \langle U_k, F_k \rangle\}$ 是关系模式 R 的一个分解， $U = \{A_1, A_2, \dots, A_n\}$ ， $F = \{FD_1, FD_2, \dots, FD_p\}$ ，并设 F 是一个最小依赖集，记 FD_i 为 $X_i \rightarrow A_{li}$ ，其步骤如下。

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 属于 U_i ，则在 j 列 i 行交叉处填上 a_{ij} ，否则填上 b_{ij} 。

② 对于每一个 FD_i 做如下操作：找到 X_i 所对应的列中具有相同符号的那些行。考察这些行中 A_{li} 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。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。且分解 ρ 具有无损联接性，否则不具有无损联接性。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。如果发生循环，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 5.11】 若将关系模式 SDC ($SNO, SNAME, AGE, DEPT, DEAN, CNAME, SCORE$)， $F = \{SNO \rightarrow (SNAME, AGE, DEPT), (SNO, CNAME) \rightarrow SCORE\}$ ，分解为三个关系： S ($SNO, SNAME, AGE, DEPT$)、 D ($DEPT, DEAN$) 和 SC ($SNO, CNAME, SCORE$)，判别这个分解是否具有“无损联接性”和“保持函数依赖性”。

(1) 首先构造初始表，如图 5.5 (a) 所示。

(2) 由 $SNO \rightarrow (SNAME, AGE, DEPT)$ ，可以把 b_{32} 改为 a_2 ， b_{33} 改为 a_3 ， b_{34} 改为 a_4 ；对 $(SNO, CNAME) \rightarrow SCORE$ ，因为各元组的第 1、6 列没有相同的分量，所以表不改变，最后结果如图 5.5 (b) 所示。表中没有全 a 行，因此该分解不具有无损联接性。

SNO	SNAME	AGE	DEPT	DEAN	CNAME	SCORE
a_1	a_2	a_3	a_4	b_{15}	b_{16}	b_{17}
b_{21}	b_{22}	b_{23}	a_4	a_5	b_{26}	b_{27}
a_1	b_{32}	b_{33}	b_{34}	b_{35}	a_6	a_7

(a)

SNO	SNAME	AGE	DEPT	DEAN	CNAME	SCORE
a_1	a_2	a_3	a_4	b_{15}	b_{16}	b_{17}
b_{21}	b_{22}	b_{23}	a_4	a_5	b_{26}	b_{27}
a_1	a_2	a_3	a_4	b_{35}	a_6	a_7

(b)

图 5.5 分解不具有无损联接的一个实例

(3) $F = \{SNO \rightarrow (SNAME, AGE, DEPT), (SNO, CNAME) \rightarrow SCORE\}$ 。SDC 分解为 S ($SNO, SNAME, AGE, DEPT$)、 D ($DEPT, DEAN$) 和 SC ($SNO, CNAME, SCORE$)

后，没有丢失某个函数依赖，因此该分解具有“保持函数依赖性”。

【例 5.12】 已知 $R(U, F)$, $U=\{A, B, C, D\}$, $F=\{B \rightarrow C, BC \rightarrow D, A \rightarrow D\}$, 分解为三个关系: $R_1(A, B)$ 、 $R_2(B, C)$ 和 $R_3(A, D)$, 判别这个分解是否具有“无损联接性”和“保持函数依赖性”。

(1) 首先构造初始表，如图 5.6 (a) 所示。

(2) 由 $B \rightarrow C$, 可以把 b_{13} 改为 a_3 ; 对 $BC \rightarrow D$, 因为各元组的第 4 列没有 a 值, 所以表不改变; 由 $A \rightarrow D$, 可以把 b_{14} 改为 a_4 ; 最后结果如图 5.6 (b) 所示。表中第一行为全 a 行, 因此该分解具有无损联接性。

A	B	C	D
a_1	a_2	b_{13}	b_{14}
b_{21}	a_2	a_3	b_{24}
a_1	b_{32}	b_{33}	a_4

(a)

A	B	C	D
a_1	a_2	a_3	a_4
b_{21}	a_2	a_3	b_{24}
a_1	b_{32}	b_{33}	a_4

(b)

图 5.6 分解具有无损联接的一个实例

(3) $F=\{B \rightarrow C, BC \rightarrow D, A \rightarrow D\}$ 。 R 分解为 $R_1(A, B)$ 、 $R_2(B, C)$ 和 $R_3(A, D)$ 后, 丢失了 $BC \rightarrow D$ 这个函数依赖, 因此该分解具有“不保持函数依赖性”。

分解具有无损联接性和保持函数依赖性是两个相互独立的标准。具有无损联接性的分解不一定具有保持函数依赖性。同样, 具有保持函数依赖性的分解也不一定具有无损联接性。

2. 分离性

分离性需要属性之间的“独立联系”使用不同的关系模式表达。这个性质主要是在模式设计中, 要尽可能地消除数据的冗余, 具体来说, 要求模式达到 3NF 或 BCNF。

例如, 前面已经分析了关系模式 $SD(SNO, SNAME, AGE, DEPT, DEAN)$ 属于 2NF, 系和系主任的信息需要重复存储若干次, 存在数据的冗余。而当把 SD 分解成 $S(SNO, SNAME, AGE, DEPT)$ 和 $D(DEPT, DEAN)$ 时, S 和 D 都属于 3NF, 减少了数据冗余的问题。

例如, 前面已经分析了关系模式 $SCR(SNO, SNAME, COMPETITON, TIME, RANKING)$ 属于 3NF, 学生的姓名需要重复存储若干次, 存在数据的冗余。而当把 SCR 分解成 $S(SNO, SNAME)$ 和 $SR(SNO, COMPETITON, TIME, RANKING)$ 时, S 和 SR 都属于 BCNF, 减少了数据冗余的问题。

3NF 消除了非主属性对候选码的传递函数依赖, 而 BCNF 消除了主属性对候选码的部分函数依赖和传递函数依赖。通过模式的分解, 使用不同的关系模式描述属性之间的“独立联系”, 将数据冗余度减少到极小。

3. 最小冗余性

最小冗余性要求在分解后的关系模式能表达原来所有信息的前提下, 实现模式个数和模式中的属性总数达到最少。

例如, 若将关系模式 SDC (SNO, SNAME, AGE, DEPT, DEAN, CNAME, SCORE) 分解为 5 个关系: S_1 (SNO, SNAME)、 S_2 ((SNO, AGE)、 S_3 ((SNO, DEPT)、 D (DEPT, DEAN) 和 SC (SNO, CNAME, SCORE)。由于模式的个数很多, 必定存在一定的数据冗余问题。

若将关系模式 SDC 分解为三个关系: S (SNO, SNAME, AGE, DEPT, DEAN)、 D (SNO, DEPT, DEAN) 和 SC (SNO, CNAME, SNAME, AGE, DEPT, SCORE)。很显然, 其中模式中的属性的总数很多, 也存在一定的数据冗余问题。

而将 SDC 分解为三个关系: S (SNO, SNAME, AGE, DEPT)、 D (DEPT, DEAN) 和 SC (SNO, CNAME, SCORE)。其中模式的个数和模式中的属性的总数均为最少, 数据冗余度也很低。

规范化理论提供了一套完整的模式分解方法, 按照这套算法可以做到: 如果要求分解既具有无损联接性, 又具有保持函数依赖性, 则分解一定能够达到 3NF, 但不一定能够达到 BCNF。

所以在 3NF 的规范化中, 既要检查分解是否具有无损联接性, 又要检查分解是否具有保持函数依赖性。

只有这两条都满足, 才能保证分解的正确性和有效性, 才既不会发生信息丢失, 又保证关系中的数据满足完整性约束。

小 结

(1) 由于关系模式中的属性之间存在着相互制约、相互依赖的关系, 它们直接影响着关系模式的质量, 而关系模式设计的质量决定是否引起数据库中的数据冗余和操作异常等问题, 因此, 在数据库的设计中进行关系的规范化是非常重要的步骤。

(2) 规范化的目的就是使关系模式的结构更加合理, 消除操作中引起的一些异常, 并且使数据的冗余度降低, 便于数据库中的操作。完全和部分函数依赖、传递和直接的函数依赖、码的定义和确定方法, 这些概念是规范化理论的依据和规范化程度的准则要素。在设计关系数据库中的关系模式时, 需要遵循一定的规则。范式的定义就是给出了这样的一些规则。为了满足不同系统的实际要求, 可选择不同的范式级别。

其中, 1NF 解决了表中表问题; 2NF 解决了非主属性与候选码之间的部分函数依赖问题; 3NF 解决了非主属性与候选码之间的传递函数依赖问题; BCNF 解决了主属性与候选码之间的部分函数依赖和传递函数依赖问题。

(3) 为了提高关系模式范式的等级, 可对其进行投影分解。Armstrong 公理系统推理规则是关系模式分解算法的理论基础。属性集的闭包、最小函数依赖集的概念对关系模式的设计的质量有直接的关系。

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。要求具有无损联接性和保持函数依赖性; 模式需要达到 3NF 或 BCNF; 分解后的模式个数最少且模式

中属性总数最少。

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删除异常,这是指()。
A. 该删除的数据不能实现删除 B. 数据删除后导致数据库处于不一致状态
C. 删除了不该删除的数据 D. 以上都不对
2. 若属性 A 函数决定属性 B 时,则属性 A 与属性 B 之间具有()的联系。
A. 一对一 B. 一对多 C. 多对一 D. 多对多
3. 有关系模式 $R(V, W, X, Y, Z)$, 其中, 函数依赖集 $F=\{V \rightarrow W, (X, Y) \rightarrow V, (X, W) \rightarrow Y, (X, Z) \rightarrow Y\}$, 关系模式 R 的候选码是()。
A. (X, Z) B. (X, W) C. (X, Y) D. V
4. 规范化的关系模式中, 所有属性都必须是()。
A. 互不相关的 B. 相互关联的 C. 长度可变的 D. 不可分解的
5. 设关系模式 $R(A, B, C, D, E)$, 其中, 函数依赖集 $F=\{A \rightarrow C, B \rightarrow A, CD \rightarrow B, E \rightarrow D\}$, 则不可导出的函数依赖是()。
A. $AD \rightarrow B$ B. $CD \rightarrow AE$ C. $CE \rightarrow U$ D. $B \rightarrow C$
6. 设关系模式 R 属于第2范式, 若在 R 中消除了传递函数依赖, 则 R 至少属于()。
A. 第1范式 B. 第2范式 C. 第3范式 D. 第4范式
7. 设关系模式 $R(U, F)$, 其中, $U=\{P, S, T\}$, $F=\{PS \rightarrow T, ST \rightarrow P\}$, 则 R 至多属于()。
A. 第2范式 B. 第3范式 C. BC范式 D. 第5范式
8. 下列关于函数依赖的叙述中,()是正确的。
A. 由 $X \rightarrow Y, Y \rightarrow Z$, 有 $X \rightarrow YZ$ B. 由 $XY \rightarrow Z$, 有 $X \rightarrow Z$ 或 $Y \rightarrow Z$
C. 由 $X \rightarrow Y, WX \rightarrow Z$, 有 $WY \rightarrow Z$ D. 由 $X \rightarrow Y$ 及 $Z \subseteq X$, 有 $Y \rightarrow Z$
9. 存在非主属性对候选码的部分函数依赖的关系模式属于()。
A. 第1范式 B. 第2范式 C. 第3范式 D. BC范式
10. 已知 $R(U, F)$, $U=\{A, B, C\}$, $F=\{B \rightarrow A\}$, 有分解 $\rho_1=\{AB, BC\}$, 则 ρ_1 ()。
A. 具有无损联接, 保持函数依赖 B. 不具有无损联接, 保持函数依赖
C. 具有无损联接, 不保持函数依赖 D. 不具有无损联接, 不保持函数依赖

二、填空题

1. 一个不好的关系模式会存在_____和_____等问题。
2. 数据依赖分为_____依赖、_____依赖和联接依赖。
3. 设关系模式 $R(U)$, x, y 是 U 的子集, x' 是 x 的任意一个真子集, 若_____并且_____, 则称 y 部分函数依赖于 x 。
4. 设关系模式 $R(U)$, x, y, z 是 U 的子集, 若 $x \rightarrow y$, _____, 且 $y \twoheadrightarrow x$, 但_____, 则称 z 传递函数依赖于 x 。
5. 设 K 为 $R(U)$ 中的属性或属性组, 若_____, 则 K 称为 R 的候选码(候选键或候

选关键字)。

6. 包含在任何一个候选码中的属性称为_____；包含关系模式中全部属性的候选码称为_____。

7. 一个较低范式的关系，可以通过关系的分解转换为若干个_____范式关系的集合，这一过程就叫作_____。

8. F 与它的最小函数依赖集是_____，每个函数依赖集 F _____只有一个最小函数依赖集。

9. 关系模式的分解是否与原关系等价需要进行_____或者_____的判断。

10. Armstrong 公理系统是_____的和_____的。

三、问答题

1. 设关系模式 $R(U, F)$ ，其中， $U=\{H, I, J, K, L, M\}$ ， $F=\{HI \rightarrow J, IJ \rightarrow K, IL \rightarrow J, JK \rightarrow I, JL \rightarrow HM, JM \rightarrow IK, J \rightarrow H, K \rightarrow LM\}$ ，求出 R 的所有候选码。

2. 设关系模式 $R(U, F)$ ，其中， $U=\{A, B, C, D, E, G\}$ ， $F=\{B \rightarrow G, E \rightarrow A, BE \rightarrow D, A \rightarrow C\}$ ，判断关系模式属于第几范式，若没达到 3NF，则将其分解至 3NF。

3. 关系模式 $R(U, F)$ ， $U=\{COURSE, TEACHER, TIME, CLASSROOM, STUDENT\}$ ，其中，COURSE 代表课程，TEACHER 代表老师，TIME 代表上课时间，CLASSROOM 代表教室，STUDENT 代表学生， $F=\{COURSE \rightarrow TEACHER, (TIME, CLASSROOM) \rightarrow COURSE, (TIME, TEACHER) \rightarrow CLASSROOM, (TIME, STUDENT) \rightarrow CLASSROOM\}$ ，确定关系模式属于第几范式。

4. 关系模式选课(学号，课程号，成绩)，函数依赖集 $F=\{(学号, 课程号) \rightarrow 成绩\}$ 。试问，该关系模式是否为 BCNF，并证明结论。

5. 设关系模式 $R(U, F)$ ，其中， $U=\{A, B, C, D, E, G\}$ ， $F=\{AB \rightarrow C, BC \rightarrow D, BE \rightarrow C, C \rightarrow A, CD \rightarrow B, CE \rightarrow AG, CG \rightarrow BD, D \rightarrow EG\}$ ，求它的最小函数依赖集。

6. 设关系模式 $R(U, F)$ ，其中， $U=\{A, B, C, D, E\}$ ， $F=\{AB \rightarrow C, AC \rightarrow B, B \rightarrow D, C \rightarrow E, CE \rightarrow B\}$ ，求 $(AB)_F^+$ 、 $(CE)_F^+$ 、 $(D)_F^+$ 。

7. 设关系模式 $R(A, B, C, D, E)$ ， $F=\{B \rightarrow A, C \rightarrow B, D \rightarrow C\}$ ，将 R 分解为 $p=\{AB, BDE, CD\}$ 。判断 p 是否具有无损联接性和保持函数依赖性。

8. 设关系模式 $R\{B, O, I, S, Q, D\}$ ， $F=\{S \rightarrow D, Q \rightarrow S, Q \rightarrow B, OS \rightarrow I\}$ ，要求把 R 分解为 BCNF，并且具有无损联接性。

9. 某宾馆的收费管理系统中用关系模式“收费(宾客姓名，性别，年龄，身份证号，地址，客房号，住宿日期，退房日期，押金)”进行记录，语义为：宾客中可能存在同名的现象。一个客人可以有 multiple、不同时间到该宾馆住宿。

(1) 关系模式 R 最高已经达到第几范式？为什么？

(2) 如果 R 不属于 2NF，请将 R 分解成 2NF 模式集。

10. 现在要建立一个关于学科部、系、学生、班级、社团等信息的关系数据库。语义为：一个学科部有若干个系，一个系有若干个专业，每个专业每年可招多个班，每个班有若干名学生，一个系的学生住在同一个宿舍区，每个学生可参加多个社团，每个社团有若干名学生，学生参加某个社团有一个入会年份。

描述学科部的属性：学科部号、学科部名、部主任、部办地点、人数。

描述系的属性：系名、系号、系主任、系办地点、人数、学科部号、宿舍区。

描述学生的属性：学号、姓名、年龄、系号、班号。

描述班级的属性：班号、专业名、入校年份、系号、人数。

描述社团的属性：社团名、成立年份、办公地点、人数。

(1) 请给出所有的关系模式，并写出每个关系模式的最小函数依赖集。

(2) 指出各个关系模式的候选码、外码、全码，若有请指出。