



6.2 规范化(函数依赖)

规范化(Normalization)是指定义关系模式应该符合的条件(范式)。符合规范化的关系模式就不存在某些操作异常，冗余也会减小。

☞ 函数依赖(Functional Dependencies, 简称为FD)

定义：设 $R(U)$ 是一个属性集 U 上的关系模式， X 和 Y 是 U 的子集。若对于 $R(U)$ 的任意一个可能的关系 r ， r 中不可能存在两个元组在 X 上的属性值相等，而在 Y 上的属性值不等，则称“ X 函数确定 Y ”或“ Y 函数依赖于 X ”，记作 $X \rightarrow Y$ (读作 X 决定 Y)。 X 称为这个函数依赖的决定因素(Determinant)。

例：学生关系 $Student(Sno, Sname, Ssex, Sage, Sdept)$ 假设不允许重名，则有：

$Sno \rightarrow Ssex, Sno \rightarrow Sage, Sno \rightarrow Sdept,$

$Sno \rightarrow Sname, Sname \rightarrow Sno,$

$Sname \rightarrow Ssex, Sname \rightarrow Sage, Sname \rightarrow Sdept.$



6.2 规范化(函数依赖)

说明：

1. 函数依赖不是指关系模式R的某个或某些关系实例满足的约束条件，而是指R的**所有关系实例**均要满足的约束条件。
2. 函数依赖是**语义范畴**的概念。只能根据数据的语义来确定函数依赖。
例如“姓名 \rightarrow 年龄”这个函数依赖只有在不允许有同名人的条件下成立。
3. 现实世界中实体的很多属性间都有函数依赖关系，如“学生学号决定学生的姓名”，即如果知道了学生的学号，就能确定该学生的姓名。
4. 数据库设计者也可以对现实世界作强制的规定。例如规定不允许同名人的出现，函数依赖“姓名 \rightarrow 年龄”成立。所插入的元组必须满足规定的函数依赖，若发现有同名人的存在，则拒绝插入该元组。



6.2 规范化(函数依赖)

几种特殊的函数依赖:

在关系模式R(U)中, 对于U的子集X和Y,

- ①若 $X \rightarrow Y$, 但 $Y \not\subseteq X$, 则称 $X \rightarrow Y$ 是**非平凡的函数依赖**
- ②若 $X \rightarrow Y$, 但 $Y \subseteq X$, 则称 $X \rightarrow Y$ 是**平凡的函数依赖**
- ③若 $X \rightarrow Y$, 并且 $Y \rightarrow X$, 则记为 $X \leftrightarrow Y$ 。(X与Y相互决定)
- ④若Y不函数依赖于X, 则记为 $X \not\rightarrow Y$ 。

例: 在关系SC(Sno, Cno, Grade)中,

非平凡函数依赖: $(Sno, Cno) \rightarrow Grade$

平凡函数依赖: $(Sno, Cno) \rightarrow Sno, (Sno, Cno) \rightarrow Cno$

注: 对于任一关系模式, 平凡函数依赖都是必然成立的, 它不反映新的语义, 因此若不特别声明, 我们总是讨论非平凡函数依赖。



6.2 规范化(函数依赖)

定义：在关系模式R(U)中，如果 $X \rightarrow Y$ ，并且对于X的任何一个真子集 X' ，都有 $X' \not\rightarrow Y$ ，则称Y完全函数依赖于X，记作 $X \xrightarrow{F} Y$ 。若 $X \rightarrow Y$ ，但Y不完全函数依赖于X，则称Y部分函数依赖于X，记作 $X \xrightarrow{P} Y$ 。

例：在关系SC(Sno, Sname, Cno, Grade)中，

由于： $Sno \rightarrow Grade$ ， $Cno \rightarrow Grade$ ，

因此： $(Sno, Cno) \xrightarrow{F} Grade$

$(Sno, Sname, Cno) \xrightarrow{P} Grade$

定义：在关系模式R(U)中，如果 $X \rightarrow Y$ ，($Y \not\subseteq X$)， $Y \rightarrow X$ ， $Y \rightarrow Z$ ，则有 $X \rightarrow Z$ ，称Z传递函数依赖于X，记为： $X \xrightarrow{T} Z$ 。

注：①如果 $Y \rightarrow X$ ，即 $X \leftarrow Y$ ，则Z直接依赖于X；

②如果 $Y \subseteq X$ ，则 $X \xrightarrow{P} Z$ 。

例：在关系Std(Sno, Sdept, Mname)中，有：

$Sno \rightarrow Sdept$ ， $Sdept \rightarrow Mname$ ，则 $Sno \xrightarrow{T} Mname$



6.2 规范化(函数依赖)

定义：设K为关系模式R<U, F>中的属性或属性组。若 $K \xrightarrow{F} U$ ，则K称为R的一个**候选码**(Candidate Key)。若关系模式R有多个候选码，则选定其中的一个做为**主码**(Primary key)。候选码常常简称为码。

主码的两个性质：

① 决定性： $K \rightarrow U$

② 最小性： $\neg \exists K' \subset K$ ，使得 $K' \rightarrow U$

- 主属性(Prime Attribute)：所有候选码中出现的属性
- 非主属性(Nonprime Attribute)：不出现在任何候选码中的属性
- 全码(All Key)：由关系模式的所有属性构成码

例：关系模式S(Sno, Sname, Sdept, Sage)无重名，则Sno, Sname是候选码，Sno, Sname是主属性，Sdept, Sage是非主属性。



6.2 规范化(函数依赖)

定义：关系模式 R 中属性或属性组 X 并非 R 的码，但 X 是另一个关系模式的码，则称 X 是 R 的**外部码**(Foreign key)，也称外码。

例：在关系 $SC(Sno, Cno, Grade)$ 中，码为 (Sno, Cno) ， Sno 是另一关系 $S(Sno, Sname, \dots)$ 的码，而非 SC 的码，故 Sno 是 SC 的外码，同样 Cno 也是 SC 的外码。

注：① 主码又和外部码一起提供了表示关系间联系的手段。

② 给出多个候选码时，以 “,” 分隔不同的候选码；当候选码为多个属性时，候选码用 $()$ 括起来，属性间用 “,” 隔开。

如：关系模式 $S(Sno, Sname, Sdept, Sage)$ 的候选码为： Sno , $Sname$ 。

关系模式 $SC(Sno, Cno, Grade)$ 的候选码为 (Sno, Cno) 。