

DEPT(Sdept, Mname, Sdept→Mname)

这三个模式都不会发生插入异常、删除异常的问题，数据的冗余也得到了控制。

一个模式的数据依赖会有哪些不好的性质，如何改造一个不好的模式，这就是下一节规范化要讨论的内容。

6.2 规范化

本节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程度为第一范式、第二范式、第三范式和第四范式等；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。

6.1 节关系模式 $Student \langle U, F \rangle$ 中有 $Sno \rightarrow Sdept$ 成立，也就是说在任何时刻 $Student$ 的关系实例（即 $Student$ 数据表）中，不可能存在两个元组在 Sno 上的值相等，而在 $Sdept$ 上的值不等。因此，表 6.2 的 $Student$ 表是错误的。因为表中有两个元组在 Sno 上都等于 S_1 ，而 $Sdept$ 上一个为计算机系，一个为自动化系。

表 6.2 一个错误的 $Student$ 表

Sno	Sdept	Mname	Cno	Grade
S_1	计算机系	张明	C1	95
S_1	自动化系	张明	C1	90
S_3	计算机系	张明	C1	88
S_4	计算机系	张明	C1	70
S_5	计算机系	张明	C1	78
\vdots	\vdots	\vdots	\vdots	\vdots

6.2.1 函数依赖

定义 6.1 设 $R(U)$ 是属性集 U 上的关系模式， X, 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 $X \rightarrow Y$ 。

函数依赖和别的数据依赖一样是语义范畴的概念，只能根据语义来确定一个函数依赖。例如，姓名 \rightarrow 年龄这个函数依赖只有在该部门没有同名人的条件下成立。如果允许有同名入，则年龄就不再函数依赖于姓名了。

设计者也可以对现实世界作强制性规定，例如规定不允许同名入出现，因而使姓名 \rightarrow 年龄函数依赖成立。这样当插入某个元组时这个元组上的属性值必须满足规定的函数依赖，若发现有同名入存在，则拒绝插入该元组。

注意：函数依赖不是指关系模式 R 的某个或某些关系满足的约束条件，而是指 R 的一切关系均要满足的约束条件。

下面介绍一些术语和记号。

- $X \rightarrow Y$ ，但 $Y \not\subseteq X$ ，则称 $X \rightarrow Y$ 是非平凡的函数依赖。