

# 第6章

## 关系数据理论

- 6.1 问题的提出
- 6.2 规范化
- 6.3 数据依赖的公理系统

数据库逻辑设计的工具——关系数据库的规范化理论

# 6.1 问题的提出

- 关系模式五元组  $R(U, D, DOM, F)$

- 关系名  $R$  是符号化的元组语义

- $U$  为一组属性

- $D$  为属性组  $U$  中的属性所来自的域

- $DOM$  为属性到域的映射

- $F$  为属性组  $U$  上的一组数据依赖

$R\langle U, F \rangle$

## • 数据依赖

- 是一个关系内部属性与属性之间的一种约束关系
  - 通过属性间值的相等与否体现出来的数据间相互联系
- 是现实世界属性间相互联系的抽象
- 是数据内在的性质
- 是语义的体现

函数依赖

多值依赖



- 函数依赖普遍存在于现实生活中

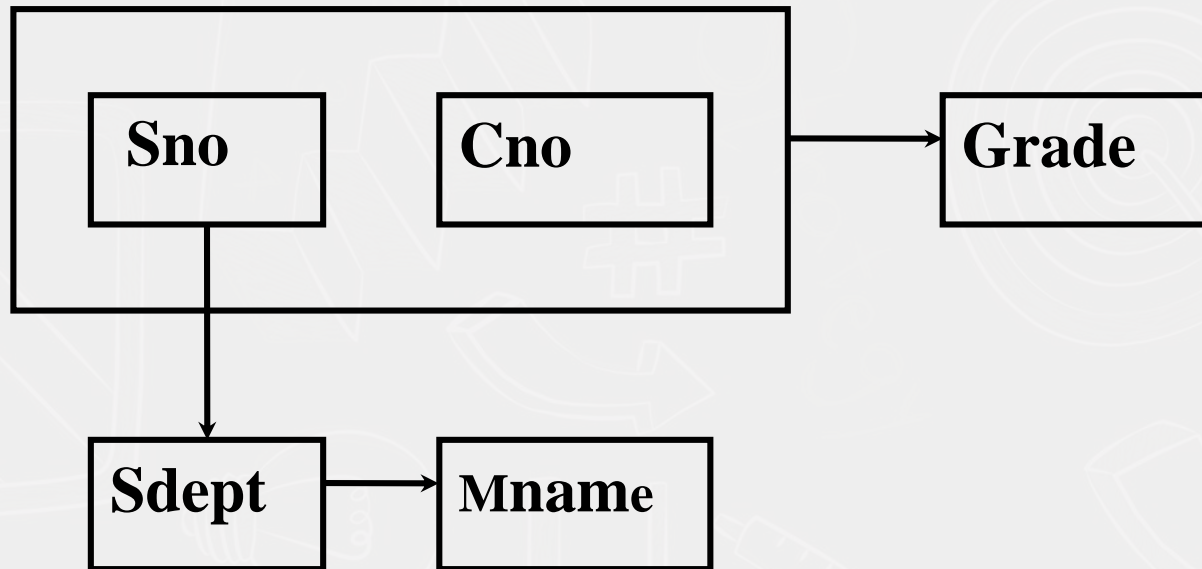
- 描述一个学生关系，可以有学号、姓名、系名等属性。

- 一个学号只对应一个学生，一个学生只在一个系中学习
    - “学号”值确定后，学生的姓名及所在系的值就被唯一确定。

- $Sname=f(Sno)$ ,  $Sdept=f(Sno)$

- 即Sno函数决定Sname
    - Sno函数决定Sdept
    - 记作 $Sno \rightarrow Sname$ ,  $Sno \rightarrow Sdept$

- $U = \{Sno, Sdept, Mname, Cno, Grade\}$
- $F = \{Sno \rightarrow Sdept, Sdept \rightarrow Mname, (Sno, Cno) \rightarrow Grade\}$



- 关系模式Student<U, F>中存在的问题：

## (1) 数据冗余

### ■浪费大量的存储空间

- 每一个系主任的姓名重复出现，重复次数与该系统所有学生的所有课程成绩出现次数相同。

## (2) 更新异常 (Update Anomalies)

- 数据冗余，更新数据时，维护数据完整性代价大。
  - 某系更换系主任后，必须修改与该系学生有关的每一个元组。



### (3) 插入异常 (Insertion Anomalies)

- 如果一个系刚成立，尚无学生，则无法把这个系及其系主任的信息存入数据库。

### (4) 删除异常 (Deletion Anomalies)

- 如果某个系的学生全部毕业了，则在删除该系学生信息的同时，把这个系及其系主任的信息也丢掉了。

## ❖ 结论

- Student关系模式不是一个好的模式。
- 一个“好”的模式应当不会发生插入异常、删除异常和更新异常，数据冗余应尽可能少。

## ❖ 原因

- 由存在于模式中的某些数据依赖引起的。

## ❖ 解决方法

- 用规范化理论改造关系模式来消除其中不合适的数据依赖

❖把这个单一的模式分成三个关系模式：

■ $S(Sno, Sdept, Sno \rightarrow Sdept)$ ;

■ $SC(Sno, Cno, Grade, (Sno, Cno) \rightarrow Grade)$ ;

■ $DEPT(Sdept, Mname, Sdept \rightarrow Mname)$ ;

❖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。