



国家“十一五”规划教材

# 数据库原理与应用教程(第4版)



机械工业出版社  
China Machine Press



# 第6章 关系数据库规范化理论



- 6.1 函数依赖
  - 6.1.1 函数依赖基本概念
  - 6.1.2 一些术语和符号
  - 6.1.3 为什么要讨论函数依赖
- 6.2 关系规范化
  - 6.2.1 关系模式中的码
  - 6.2.2 范式

## 6.1 一些数据语义问题



- 数据的语义不仅表现为完整性约束，对关系模式的设计也提出了一定的要求。
- 如何构造一个合适的关系模式，应构造几个关系模式，每个关系模式由哪些属性组成等，都是数据库设计问题，确切地讲是关系数据库的逻辑设计问题。

# 6.1.1 函数依赖基本概念



- 省 =  $f$ (城市): 只要给出一个具体的城市值, 就会有唯一的一个省值和它对应,
- 如“武汉市”在“湖北省”, 这里“城市”是自变量  $X$ , “省”是因变量或函数值  $Y$ .
- 把  $X$  函数决定  $Y$ , 或  $Y$  函数依赖于  $X$  表示为:

$$X \rightarrow Y$$

- 如果有关系模式  $R(A_1, A_2, \dots, A_n)$ ,  $X$  和  $Y$  为  $\{A_1, A_2, \dots, A_n\}$  的子集, 则对于关系  $R$  中的任意一个  $X$  值, 都只有一个  $Y$  值与之对应, 则称  $X$  函数决定  $Y$ , 或  $Y$  函数依赖于  $X$ .

# 示例



- 例1：对学生关系模式

Student (Sno, Sname, Sdept, Sage)

- 有以下依赖关系：

$Sno \rightarrow Sname, Sno \rightarrow Sdept, Sno \rightarrow Sage$

例2： SC (Sno, Cno, Grade)

$(Sno, Cno) \rightarrow Grade$

# 函数依赖定义



- 设有关系模式  $R(A_1, A_2, \dots, A_n)$ ,  $X$  和  $Y$  均为  $\{A_1, A_2, \dots, A_n\}$  的子集,  $r$  是  $R$  的任一具体关系,  $t_1$ 、 $t_2$  是  $r$  中的任意两个元组;
- 如果由  $t_1[X] = t_2[X]$  可以推导出  $t_1[Y] = t_2[Y]$ , 则称  **$X$  函数决定  $Y$** , 或  **$Y$  函数依赖于  $X$** , 记为  $X \rightarrow Y$ 。
- 在以上定义中特别要注意, 只要
$$t_1[X] = t_2[X] \quad t_1[Y] = t_2[Y]$$
- 成立, 就有  $X \rightarrow Y$ 。也就是说只有当  $t_1[X] = t_2[X]$  为真, 而  $t_1[Y] = t_2[Y]$  为假时, 函数依赖  $X \rightarrow Y$  不成立; 而当  $t_1[X] = t_2[X]$  为假时, 不管  $t_1[Y] = t_2[Y]$  为真或为假, 都有  $X \rightarrow Y$  成立。



## 6.1.2 一些术语和符号



1. 如果  $X \rightarrow Y$ , 但  $Y$  不包含于  $X$ , 则称  
 $X \rightarrow Y$  是非平凡的函数依赖。

2. 如果  $X \rightarrow Y$ , 但  $Y$  包含于  $X$ , 则称  
 $X \rightarrow Y$  是平凡的函数依赖。

若无特别声明, 我们讨论的都是非平凡的函数依赖。

# 术语和符号（续）



3. 如果  $X \rightarrow Y$ , 则  $X$  称为决定因子。
4. 如果  $X \rightarrow Y$ , 并且  $Y \rightarrow X$ , 则记作  $X \leftrightarrow Y$ 。



# 术语和符号（续）



- 如果  $X \rightarrow Y$ ，并且对于  $X$  的一个任意真子集  $X'$  都有  $X' \not\rightarrow Y$ ，则称  $Y$  完全函数依赖于  $X$ ，并记作： $X \xrightarrow{f} Y$
- 如果  $X \rightarrow Y$  成立，并且对于  $X$  的某个真子集  $X'$  有  $X' \rightarrow Y$  成立，则称  $Y$  部分函数依赖于  $X$ ，并记作： $X \xrightarrow{p} Y$
- 如果  $X \rightarrow Y$ （非平凡函数依赖，并且  $Y \not\rightarrow X$ ）、 $Y \rightarrow Z$ ，则称  $Z$  传递函数依赖于  $X$ ，记作：

$$X \xrightarrow{\text{传递}} Z$$

# 示例



- 例1：有关系模式SC (Sno, Sname, Cno, Credit, Grade)，主码为 (Sno, Cno)，则函数依赖关系有：

$Sno \rightarrow Sname$

姓名函数依赖于学号

$(Sno, Cno) \xrightarrow{p} Sname$

姓名部分函数依赖于学号和课程号

$(Sno, Cno) \xrightarrow{f} Grade$

成绩完全函数依赖于学号和课程号

# 示例



- 例2：假设有关系模式S（Sno, Sname, Dept, Dept\_master），其中各属性分别为：学号、姓名、所在系和系主任（假设一个系只有一个主任），主码为Sno，则函数依赖关系有：

$Sno \xrightarrow{f} Sname$

姓名完全函数依赖于学号

由于： $Sno \xrightarrow{f} Dept$

所在系完全函数依赖于学号

$Dept \xrightarrow{f} Dept\_master$

系主任完全函数依赖于系

所以有： $Sno \xrightarrow{\text{传递}} Dept\_master$

系主任传递函数依赖于学号

## 6.1.3 为什么要讨论函数依赖?



有关系模式：

S-L-C(Sno,Sdept,SLOC,Cno,Grade)

Sno	Sname	Ssex	Sdept	Sloc	Cno	Grade
9512101	李勇	男	计算机系	2公寓	c01	90
9512101	李勇	男	计算机系	2公寓	c02	86
9512101	李勇	男	计算机系	2公寓	c06	NULL
9512102	刘晨	男	计算机系	2公寓	c02	78
9512102	刘晨	男	计算机系	2公寓	c04	66
9521102	吴宾	女	信息系	1公寓	c01	82
9521102	吴宾	女	信息系	1公寓	c02	75
9521102	吴宾	女	信息系	1公寓	c04	92
9521102	吴宾	女	信息系	1公寓	c05	50
9521103	张海	男	信息系	1公寓	c02	68
9521103	张海	男	信息系	1公寓	c06	NULL
9531101	钱小平	女	数学系	1公寓	c01	80
9531101	钱小平	女	数学系	1公寓	c05	95

# 存在问题



- 数据冗余问题
- 数据更新问题
- 数据插入问题
- 数据删除问题



# 结论



- Student关系模式不是一个好的模式。
- “好”的模式：
  - 不会发生插入异常、删除异常、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
- 解决方法：通过分解关系模式来消除其中不合适的数据依赖

## 6.2 关系规范化



- 关系规范化是指导将有“不良”函数依赖的关系模式转换为良好的关系模式的理论。
- 这里涉及到范式的概念，不同的范式表示关系模式遵守的不同的规则。



## 6.2.1 关系模式中的码



**候选码**：设 $K$ 为 $R(U, F)$ 中的属性或属性组，若 $K$ 完全决定 $U$ ，则 $K$ 为 $R$ 的**候选码**。

**主码**：关系 $R(U, F)$ 中可能有多个候选码，则选其中一个作为主码。

**全码**：候选码为整个属性组。

包含在任一候选码中的属性称为**主属性**，不包含在任一候选码中的属性称为**非主属性**。

# 示例



- 例1：学生（学号，姓名，性别，身份证号，年龄，所在系）
- 候选码：学号，身份证号。
- 主码：“学号”或“身份证号”。
- 主属性：学号，身份证号。
- 非主属性：姓名，性别，年龄，所在系。

# 示例



- 例2．有关系模式：选课（学号，课程号，考试次数，成绩）
- 设一个学生对一门课程可以有多次考试，每一次考试有一个考试成绩。
- 候选码：（学号，课程号，考试次数），也为主码。
- 主属性：学号，课程号，考试次数
- 非主属性：成绩。

# 示例



- 例3 . 有关系模式：授课（教师号，课程号，学年）
- 语义：一个教师在一个学年可以讲授多门不同的课程，可以在不同学年对同一门课程讲授多次，但不能在同一个学年对同一门课程讲授多次。一门课程在一个学年可以由多个不同的教师讲授，同一个学年可以开设多门课程，同一门课程可以在不同学年开设多次。
- 候选码：（教师号，课程号，学年。
- 主码：同候选码。
- 主属性：教师号，课程号，学年。
- 非主属性：没有
- 是全码表。

# 外码



**外码：**用于关系表之间建立关联的属性（组）。

定义：关系模式R中属性（组）X不一定是R中的码，但X是另一个关系模式S的候选码（通常为主码），则称X为R的外码。

## 6.2.2 范式



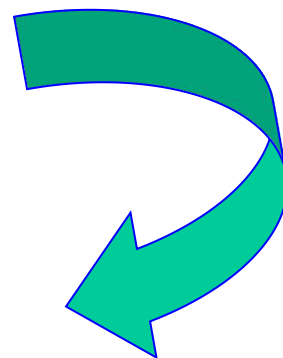
- 关系数据库中的关系要满足一定的要求，满足不同程度要求的为不同的**范式**（Normal Form）。
- **范式的种类：**
  - 第一范式(1NF)
  - 第二范式(2NF)
  - 第三范式(3NF)

# 第一范式



第一范式 (1NF): 不包含重复组的关系

系名称	高级职称人数	
	教授	副教授
计算机系	6	10
信息管理系	3	5
电子与通讯系	4	8



系名称	教授人数	副教授人数
计算机系	6	10
信息管理系	3	5
电子与通讯系	4	8



# 第二范式



**第二范式 (2NF)** : 如果  $R(U, F) \in 1NF$ , 并且  $R$  中的每个非主属性都完全函数依赖于主码, 则  $R(U, F) \in 2NF$ 。

**例: S-L-C(Sno, Sdept, SLOC, Cno, Grade)**

因为有:  $Sno \rightarrow SLOC$ , 因此存在部分函数依赖, 该表不是 2NF。

# 分解方法



- 首先，对于组成主码的属性集合的每一个子集，用它作为主码构成一个表。
- 然后，将依赖于这些主码的属性放置到相应的表中。
- 最后，去掉只由主码的子集构成的表。

# 分解S-L-C表



- 对S-L-C (Sno, Sdept, SLOC, Cno, Grade)
- 首先分解为如下形式的三张表：
  - S-L (Sno, ...)
  - C (Cno, ...)
  - S-C (Sno, Cno, ...)

## 分解S-L-C表（续）



- 然后，将依赖于这些主码的属性放置到相应的表中，形成如下三张表：
  - S-L（**Sno**, Sdept, Sloc）
  - C（**Cno**）
  - S-C（**Sno**, **Cno**, Grade）

## 分解S-L-C表（续）



- 最后，去掉只由主码的子集构成的表。
- S-L-C关系模式最终分解的形式为：
  - S-L ( Sno, Sdept, Sloc )
  - S-C ( Sno, Cno, Grade )

# 分解后的函数依赖关系



- 分解后的关系模式的函数依赖关系：
- S-L :  $\text{Sno}^f \rightarrow \text{Sdept}$  ,  $\text{Sno}^f \rightarrow \text{SLOC}$   
是2NF
- S-C:  $(\text{Sno}, \text{Cno})^f \rightarrow \text{Grade}$  : 是2NF

# S-L(Sno,Sdept,SLOC) 存在问题



- **数据冗余**：有多少个学生就有多少个重复的Sdept和SLOC;
- **插入异常**：当新建一个系时，若还没有招收学生，则无法插入;



# 第三范式



第三范式 ( 3NF ) : 如果  $R(U, F) \in 2NF$ , 并且所有非主属性都不传递依赖于主码, 则  $R(U, F) \in 3NF$ 。

- S-L(Sno, Sdept, SLOC)
- $\because$  Sno  $\xrightarrow{\text{传递}}$  SLOC,  $\therefore$  不是 3NF

# 分解方法



- （1）对于不是候选码的每个决定因子，从表中删去依赖于它的所有属性；
- （2）新建一个表，新表中包含在原表中所有依赖于该决定因子的属性；
- （3）将决定因子作为新表的主码。

# 分解S-L关系模式



- S-L分解后的关系模式为：

S-D ( Sno, Sdept )

S-L ( Sdept, Sloc )

对 S-D, 有:  $Sno \xrightarrow{f} Sdept$ , 因此 S-D 是 3NF 的。

对 S-L, 有:  $Sdept \xrightarrow{f} Sloc$ , 因此 S-L 也是 3NF 的。



- **S-L-C** ( Sno, Sname, Ssex, Sdept, Sloc, Cno, Grade )  
共分解为如下三个关系模式，每个关系模式都是3NF的。
- **S-D** ( Sno, Sname, Ssex, Sdept ) , Sdept为引用S-L关系模式的外码。
- **S-L** ( Sdept, Sloc ) , 没有外码。
- **S-C** ( Sno, Cno, Grade ) , Sno为引用S-D关系模式的外码。