

# 数据库系统概论 高级篇

An Introduction to Database System

王珊

中国人民大学信息学院

School of Information,  
Renmin University of China

2016

# 第六章 关系数据理论

- 6.1 问题的提出—为什么要学习关系数据理论
- 6.2 规范化
- 6.3 数据依赖的公理系统（简单介绍）
- 6.4 模式的分解（简单介绍）
- 6.5 小结



## 6.4 模式的分解

### 6.4.1 模式分解的3个定义

### 6.4.2 分解的无损连接性和保持函数依赖性

### 6.4.3 模式分解的算法



# 模式分解的3个定义（续）

❖ 关系模式的规范化过程是通过对关系模式的分解来实现的

## ❖ 什么是模式分解

定义6.16  $R\langle U, F \rangle$ 的一个分解是指： $\rho = \{R_1\langle U_1, F_1 \rangle, \dots, R_n\langle U_n, F_n \rangle\}$ ，  
其中  $U = U_1 \cup U_2 \cup \dots \cup U_n$ ，并且没有  $U_i \subseteq U_j$ ， $1 \leq i, j \leq n$ ，  
 $F_i$  是  $F$  在  $U_i$  上的投影。

相应地将  $R$  存储在二维表  $r$  中的数据分散到二维表  $r_1, r_2, \dots, r_n$  中去，其中  $r_i$  是  $r$  在属性集  $U_i$  上的投影。



## 6.4.1 模式分解的3个定义

- ❖ 把低一级的关系模式分解为若干个高一级的关系模式并不是唯一的
- ❖ 在这些分解方法中，只有能够保证分解后的关系模式与原关系模式等价的方法才有意义
- ❖ 从不同的角度去观察问题，对“等价”的概念形成了3种不同的定义：
  - 分解具有无损连接性（lossless join）
  - 分解要保持函数依赖（preserve functional dependency）
  - 分解既要保持函数依赖，又要具有无损连接性
- ❖ 这三个定义是实行模式分解的三条准则。





# 例子

[例] 对于关系模式S-L (Sno, Sdept, Sloc), S-L中有下列函数依赖:

$Sno \rightarrow Sdept$

$Sdept \rightarrow Sloc$

传递

$Sno \rightarrow Sloc$

- ❖ 已知  $S-L \in 2NF$ , 该关系模式存在非主属性对码的传递函数依赖。
- ❖ 存在插入异常、删除异常、数据冗余度大和修改复杂的问题。需要分解该关系模式, 使成为更高范式的关系模式。
- ❖ 下面给出几种不同的分解情况, 进行分析, 引出模式分解的3个定义。

# 例子：第一种分解情况

## ❖ 第一种分解情况：

将S-L分解为下面三个关系模式：

SN(Sno)

SD(Sdept)

SO(Sloc)

- ❖ SN、SD和SO都是规范化程度很高的关系模式。
- ❖ 但分解后的关系模式会丢失许多信息



# 例子：第一种分解情况

假设下面是S-L关系模式的一个关系：

S-L

Sno	Sdept	Sloc
95001	CS	A
95002	IS	B
95003	MA	C
95004	IS	B
95005	PH	B

分解后的关系为：

SN	SD	SO
Sno	Sdept	Sloc
95001	CS	A
95002	IS	B
95003	MA	C
95004	PH	
95005		

- ❖ 但分解后的关系，无法查询95001学生所在的系 或 所住的宿舍了！
- ❖ 这种分解是不可取的。丢失了许多信息，丢失了数据之间联系的信息。



# 例子：第二种分解情况

## ❖ 第二种分解情况：

将S-L分解为下面二个关系模式：

NL(Sno, Sloc)

DL(Sdept, Sloc)

分解后的关系为：

NL

Sno	Sloc
950001	A
950002	B
950003	C
950004	B
950005	B

DL

Sdept	Sloc
CS	A
IS	B
MA	C
PH	B

S-L

Sno	Sdept	Sloc
95001	CS	A
95002	IS	B
95003	MA	C
95004	IS	B
95005	PH	B



# 例子：第二种分解情况

对NL和DL关系 进行自然连接的结果为：

NL  $\bowtie$  DL

Sno	Sloc	Sdept
95001	A	CS
95002	B	IS
95002	B	PH
95003	C	MA
95004	B	IS
95004	B	PH
95005	B	IS
95005	B	PH

S-L

Sno	Sdept	Sloc
95001	CS	A
95002	IS	B
95003	MA	C
95004	IS	B
95005	PH	B

NL  $\bowtie$  DL比原来的S-L关系多了三个元组

因此我们也无法知道原来的S-L关系中究竟有哪些元组，

从这个意义上说，此分解仍然丢失了信息。



# 例子：第三种分解情况

## ❖ 第三种分解情况：

将S-L分解为下面二个关系模式：

ND(Sno, Sdept)

NL(Sno, Sloc)

分解后的关系为：

ND		NL	
Sno	Sdept	Sno	Sloc
95001	CS	95001	A
95002	IS	95002	B
95003	MA	95003	C
95004	IS	95004	B
95005	PH	95005	B

对ND和NL关系进行自然连接的结果为：

ND $\bowtie$ NL		
Sno	Sdept	Sloc
95001	CS	A
95002	IS	B
95003	MA	C
95004	CS	A
95005	PH	B

第三种分解情况没有丢失信息，称为分解“具有无损连接性”。



# 例子：第三种分解情况

❖ 但是它存在下面的问题：

- 例如95001学生由CS系转到IS系，ND关系的(95001, CS)元组和NL关系的(95001, A)元组必须同时进行修改，否则会破坏数据库的一致性。

ND		NL	
Sno	Sdept	Sno	Sloc
95001	CS	95001	A
95002	IS	95002	B
95003	MA	95003	C
95004	IS	95004	B
95005	PH	95005	B

IS

B

原因：

- ❖ S-L中的函数依赖  $Sdept \rightarrow Sloc$  既没有投影到关系模式ND上，也没有投影到关系模式NL上。
- ❖ 这种分解没有保持原关系模式中的函数依赖。

# 例子：第四种分解情况

## ❖ 第四种分解情况：

将S-L分解为下面二个关系模式：

ND(Sno, Sdept), Sno→Sdept

DL(Sdept, Sloc), Sdept→Sloc

这种分解保持了函数依赖，称为“具有保持函数依赖性”。





# 例子：第四种分解情况

## ❖ 再看第四种分解情况：

将S-L分解为下面二个关系模式：

ND(Sno, Sdept), Sno→Sdept

DL(Sdept, Sloc), Sdept→Sloc

Sno	Sdept
950001	CS
950002	IS
950003	MS
950004	IS
950005	PH

Sdept	Sloc
CS	A
IS	B
MA	C
PH	B

ND ⋈ DL		
Sno	Sdept	Sloc
95001	CS	A
95002	IS	B
95003	MA	C
95004	IS	B
95005	PH	B

这种分解不仅“具有保持函数依赖性”，还“具有无损连接性”。



# 例子：分解情况分析

在给出的例子中：

第一种情况：既不具有无损连接性，也未保持函数依赖，

第二种情况：既不具有无损连接性，也未保持函数依赖。

第三种情况：具有无损连接性，但未保持函数依赖。

第四种情况：既具有无损连接性，又保持了函数依赖。



## 6.4.1 模式分解

6.4.1 模式分解的3个定义

6.4.2 分解的无损连接性和保持函数依赖性

6.4.3 模式分解的算法



# 1.具有无损连接性的模式分解

## 定义6.18

$\rho = \{R_1 \langle U_1, F_1 \rangle, \dots, R_n \langle U_n, F_n \rangle\}$  是  $R \langle U, F \rangle$  的一个分解，若对  $R \langle U, F \rangle$  的任何一个关系  $r$  均有  $r = r$  在  $\rho$  中各关系模式上投影的自然连接成立，则称分解  $\rho$  具有无损连接性。简称  $\rho$  为无损分解。

- ❖ 只有具有无损连接性的分解才能够保证不丢失信息。
- ❖ 无损连接性不一定能解决插入异常、删除异常、修改复杂、数据冗余等问题。

\* 算法6.2 判别一个分解的无损连接性。 p.197



## 2. 保持函数依赖的模式分解

### 定义6.19

$\rho = \{R_1 \langle U_1, F_1 \rangle, \dots, R_n \langle U_n, F_n \rangle\}$  是  $R \langle U, F \rangle$  的一个分解，若  $F$  所逻辑蕴含的函数依赖一定也为分解后所有的关系模式中的函数依赖  $F_i$  所逻辑蕴含，即  $F^+ = (F_1 \cup F_2 \cup \dots \cup F_n)^+$ ，则称关系模式  $R$  的这个分解是保持函数依赖的（Preserve dependency）

$$F^+ = (F_1 \cup F_2 \cup \dots \cup F_n)^+$$

引理6.3 给出了判别  $R$  的分解是否保持函数依赖的方法。 P.193



### 3. 分解的无损连接性和保持函数依赖性

- ❖ 如果一个分解具有无损连接性，则它能够保证不丢失信息。
- ❖ 如果一个分解保持了函数依赖，则它可以减轻或解决各种异常情况。
- ❖ 分解具有无损连接性和分解保持函数依赖是两个互相独立的标准。
  - 具有无损连接性的分解不一定能够保持函数依赖。
  - 保持函数依赖的分解也不一定具有无损连接性。



## 6.4 模式的分解

### 6.4.1 模式分解的3个定义

### 6.4.2 分解的无损连接性和保持函数依赖

### 6.4.3 模式分解的算法



## 6.4.3 模式分解的算法

- ❖ 算法6.3（合成法）转换为3NF的保持函数依赖的分解
- ❖ 算法6.4 转换为3NF既有无损连接性又保持函数依赖的分解
- ❖ 算法6.5（分解法）转换为BCNF的无损连接分解
- ❖ 算法6.6 达到4NF的具有无损连接性的分解



## 6.4.3 模式分解的算法

❖ 按照不同的分解算法，关系模式所能达到的范式是不相同的。

- 若要求分解保持函数依赖，那么模式分解一定能够达到3NF，但不一定能够达到BCNF。
- 若要求分解既具有无损连接性，又保持函数依赖，则模式分解一定能够达到3NF，但不一定能够达到BCNF。
- 若要求分解具有无损连接性，那么模式分解一定能够达到4NF。

无损连接	保持依赖	达到的范式
	Y	3NF， 不一定BCNF
Y	Y	3NF， 不一定BCNF
Y		4NF



# 第六章 关系数据理论

- 6.1 问题的提出—为什么要学习关系数据理论
- 6.2 规范化
- 6.3 数据依赖的公理系统（简单介绍）
- 6.4 模式的分解（简单介绍）
- 6.5 小结





# 小结

## ❖ 函数依赖

- 平凡函数依赖与非平凡函数依赖
- 完全函数依赖与部分函数依赖
- 传递函数依赖
- 码

## ❖ 范式的概念

## ❖ 关系模式规范化的基本步骤

## ❖ Armstrong公理系统

## ❖ 模式的分解



# 小结

## ❖ 关系模式规范化及基本步骤

消除决定属性  
集非码的非平  
凡函数依赖

**1NF**

↓ 消除非主属性对码的部分函数依赖

**2NF**

↓ 消除非主属性对码的传递函数依赖

**3NF**

↓ 消除主属性对码的部分和传递函数依赖

**BCNF**

↓ 消除非平凡且非函数依赖的多值依赖

**4NF**

# 小结

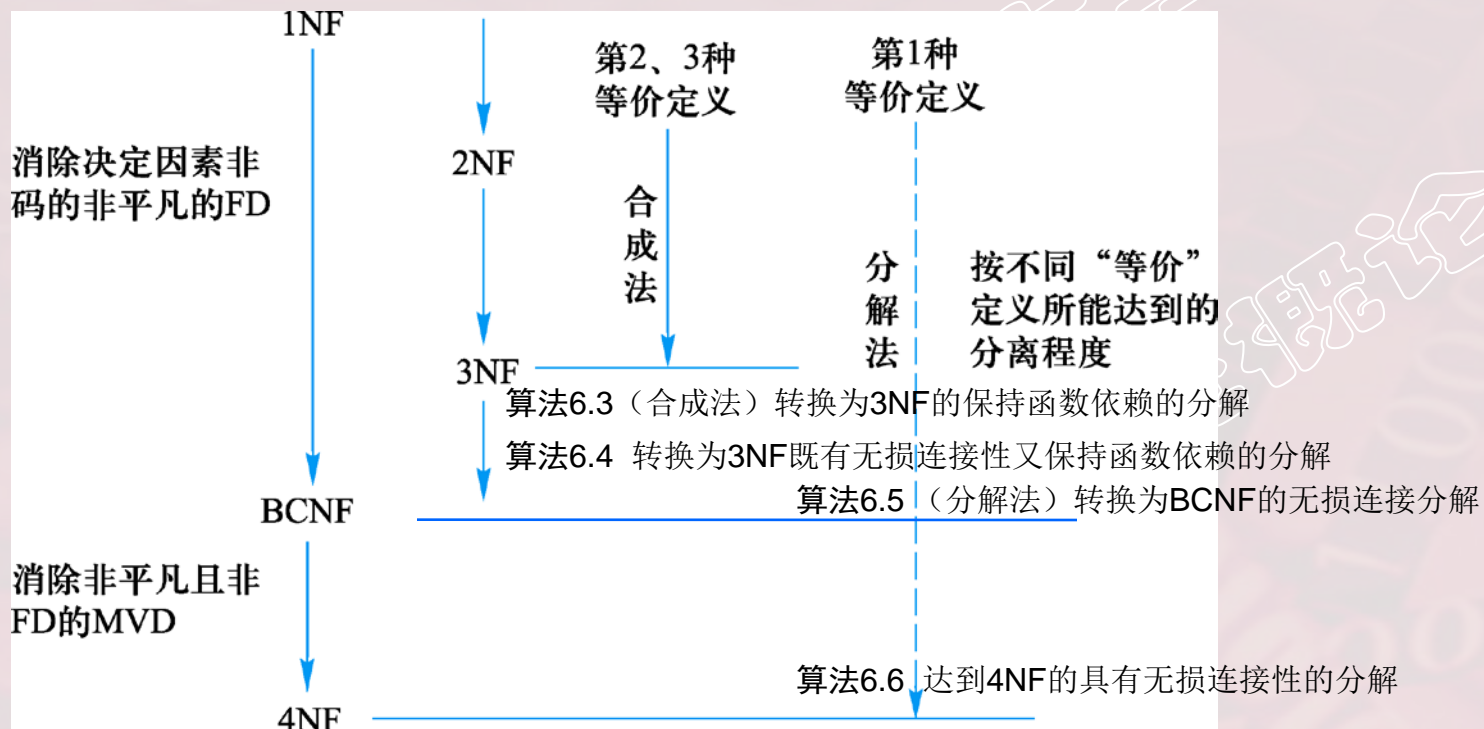
## ❖ Armstrong公理系统

- 自反律；增广律；传递律
- 合并规则；伪传递规则；分解规则
- 函数依赖闭包 $F^+$  求属性集 $X$ 关于 $F$ 的闭包 $X_F^+$
- 最小依赖集



# 小结

关系模式分解：模式分解等价的3个准则、模式分解的算法



# 小结

## 关系模式的规范化基本思想

1. 逐步消除不合适的数据依赖中使模式中的各关系模式达到某种程度的“分离”——模式分解
2. 一事一地的模式设计原则
3. 规范化理论为数据库设计提供了理论指南和算法工具。
4. 结合应用环境的具体情况，合理地设计数据库模式。





# 第六章 关系数据理论

学习关系规范化理论的重要性:

- ❖ 是关系数据库的重要理论基础。
  - ❖ 为数据库设计提供理论指南和算法工具
- 规范化理论研究的实际应用价值。



# 第六章 关系数据理论

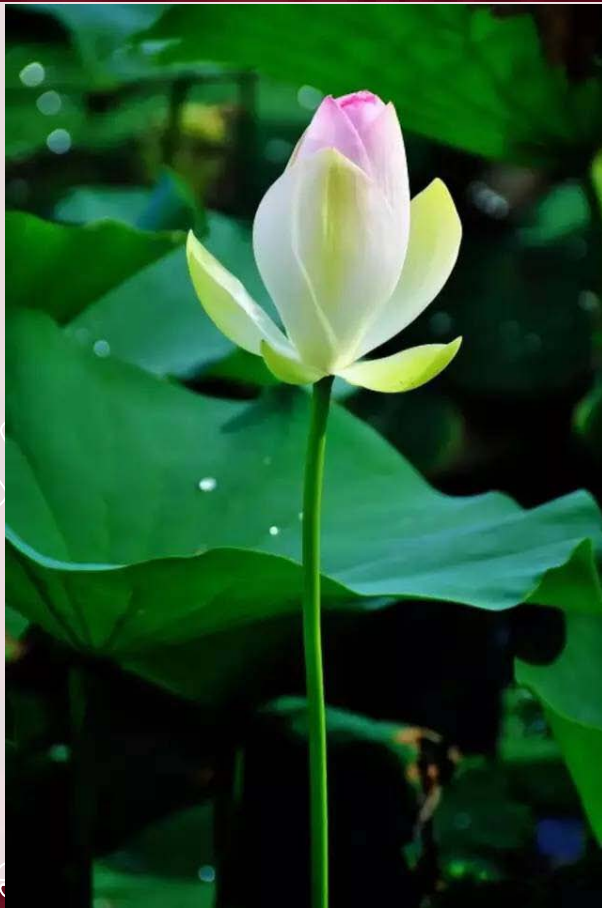
1. 为什么我们不能直接由定义6.18 来判断一个分解的无损连接性。

2. 基于右边表，是否能找出其候选码？如果能，请给出所有候选码。如果不能，请说明为什么。

A	B	C	D	E
a1	1	3	d1	3
a2	1	4	d2	3
a3	2	4	d1	3
a4	2	3	d2	5



# 第六章 关系数据理论



出淤泥而不染  
濯清涟而不妖

