



泛关系模式

总述

内容：关系数据库设计的核心是关系模式设计，关系模式设计的规范化理论是关系数据库的重要理论基础，是数据库设计的有力工具。本知识点学习规范化理论的产生背景和研究内容。

- 一、关系模式的形式化定义
- 二、泛关系模式
- 三、泛关系模式存在的问题

一、关系模式的形式化定义

回顾几个概念

- **关系**：用来描述实体、属性和实体间的联系，从形式上看，是一张二维表；
- **关系模式**：用来定义关系；
- **关系数据库**：一个给定应用领域的所有关系的集合构成一个关系数据库，从形式上看由一组二维表组成；
- **关系数据库模式**：定义这组关系的关系模式的全体。

STUDENT		GRADE		
sno	sname	sno	cno	grade
S1	李立	S1	C1	78
S2	丁惠	S1	C3	87
		S2	C1	90
		S2	C2	83
		S2	C3	77

cno	cname
C1	DB
C2	DS
C3	OS

一、关系模式的形式化定义

关系模式的一般形式: $R(U, D, DOM, F)$

R : 关系名

U : 全部属性集合

D : 属性所来自的域的集合

DOM: 属性向域的映像集合

F : 属性间数据的依赖关系集合

STUDENT		GRADE		
sno	sname	sno	cno	grade
S1	李立	S1	C1	78
S2	丁惠	S1	C3	87
COURSE		S2	C1	90
		S2	C2	83
		S2	C3	77
cno	cname			
C1	DB			
C2	DS			
C3	OS			

规范化理论主要讨论 $R \langle U, F \rangle$ 。

二、泛关系模式

如何设计一个合理的关系数据库模式

例：设计一个关系数据模型存放学生各门课考试成绩。

有若干答案：

1) STUDENT关系：(sno,sname)

COURSE关系：(cno,cname)

GRADE关系：(sno,cno,grade)

2) GRADE关系：(sno,sname,cno,grade)

COURSE关系：(cno,cname)

二、泛关系模式

3) GRADE关系: (sno,sname,cno,cname,grade)

$F = \{sno \rightarrow sname, cno \rightarrow cname, (sno, cno) \rightarrow grade\}$

我们称3) 为**泛关系模式**，它把现实问题的所有属性组成一个关系模式。

泛关系:

sno	sname	cno	cname	grade
s1	李立	c1	DB	78
s1	李立	c3	OS	87
s2	丁惠	c1	DB	90
s2	丁惠	c2	DS	83
s2	丁惠	c3	OS	77

三、泛关系模式存在的问题

(1) 数据冗余

(2) 更新异常

(3) 插入异常

(4) 删除异常

sno	sname	cno	cname	grade
s1	李立	c1	DB	78
s1	李立	c3	OS	87
s2	丁惠	c1	DB	90
s2	丁惠	c2	DS	83
s2	丁惠	c3	OS	77

产生问题的原因？

$F = \{sno \rightarrow sname, cno \rightarrow cname, (sno, cno) \rightarrow grade\}$

属性间约束关系（即数据间的依赖关系）太强。

三、泛关系模式存在的问题

解决问题的方法：
分解关系

STUDENT

sno	sname
S1	李立
S2	丁惠

GRADE

sno	cno	grade
S1	C1	78
S1	C3	87
S2	C1	90
S2	C2	83
S2	C3	77

COURSE

cno	cname
C1	DB
C2	DS
C3	OS

sno	sname	cno	cname	grade
s1	李立	c1	DB	78
s1	李立	c3	OS	87
s2	丁惠	c1	DB	90
s2	丁惠	c2	DS	83
s2	丁惠	c3	OS	77

练习1

以下哪条不是泛关系模式存在的问题（ ）？

- A. 数据冗余
- B. 数据查询异常
- C. 数据增加异常
- D. 数据删除异常

解答

以下哪条不是泛关系模式存在的问题（ B ）？

- A. 数据冗余
- B. 数据查询异常
- C. 数据增加异常
- D. 数据删除异常

小结

关系数据库设计的核心是关系模式设计。即按照一定的原则从数量众多而又相互关联的数据中，构造出一组既能较好地反映现实世界，而又有良好的操作性能的关系模式。

规范化要解决的主要问题是：

- (1) 怎样评价关系模式的优劣；
- (2) 怎样将关系模式分解为一组较理想的关系模式。



谢谢！



函数依赖

总述

内容：函数依赖是判断关系模式规范化程度的基础理论。本知识点学习数据库的函数依赖的定义、性质等基本概念，以及如何根据语义确定函数依赖关系。

- 一、函数依赖的定义和性质
- 二、完全函数依赖和部分函数依赖
- 三、平凡函数依赖和非平凡函数依赖

一、函数依赖的定义和性质

函数依赖 (Functional Dependency, F D)

是数据依赖的一种，它反映一个关系中属性或属性组之间相互依存、互相制约的关系，即反映现实世界的约束关系。

一、函数依赖的定义和性质

定义：设 $R(U)$ 是属性 U 上的一个关系模式， X 和 Y 均为 $U=\{A_1, A_2, \dots, A_n\}$ 的子集， r 为 R 的任一关系，如果对于 r 中的任意两个元组 u, v ，只要有 $u[X]=v[X]$ ，就有 $u[Y]=v[Y]$ ，则称 X 函数决定 Y ，或称 Y 函数依赖于 X ，记为 $X \rightarrow Y$ 。

一、函数依赖的定义和性质

例1：具有函数依赖关系

$sno \rightarrow sname$

$cno \rightarrow cname$

$(sno, cno) \rightarrow grade$

例2：不具有函数依赖关系

$sno \not\rightarrow grade$

$cno \not\rightarrow tname$

只能根据语义来确定函数依赖的存在。

一、函数依赖的定义和性质

属性间的联系决定函数依赖关系。

设 X 、 Y 均是 U 的子集，则：

(1) X 和 Y 间联系是 $1:1$ ，则 $X \rightarrow Y, Y \rightarrow X$ ；

(2) X 和 Y 间联系是 $M:1 (M > 1)$ ，则 $X \rightarrow Y$ ；

(3) X 和 Y 间联系是 $M:N (M, N > 1)$ ，则 X 、 Y 间不存在函数依赖。

一、函数依赖的定义和性质

自反性、增广性和传递性

设 U 是关系模式 R 的属性集， F 是 R 上成立的只涉及到 U 中属性的函数依赖集，则：

自反性：若 $Y \subseteq X \subseteq U$ ，则 $X \rightarrow Y$ 在 R 上成立。

增广性：若 $X \rightarrow Y$ 在 R 上成立，且 $Z \subseteq U$ ，则 $XZ \rightarrow YZ$ 在 R 上成立。

传递性：若 $X \rightarrow Y$ 和 $Y \rightarrow Z$ 在 R 上成立，则 $X \rightarrow Z$ 在 R 上成立。

一、函数依赖的定义和性质

自反性、增广性和传递性

例3：已知函数依赖集

$F = \{\text{学号} \rightarrow \text{姓名}; \text{课程号} \rightarrow \text{教师名}; \text{教师名} \rightarrow \text{工资}\}$ ，则：

$(\text{学号}, \text{姓名}) \rightarrow \text{姓名}$ —— 自反性

$(\text{学号}, \text{性别}) \rightarrow (\text{姓名}, \text{性别})$ —— 增广性

$\text{课程号} \rightarrow \text{工资}$ —— 传递性

二、完全函数依赖和部分函数依赖

定义：在 $R(U)$ 中，如果 $X \rightarrow Y$ ，并且对于 X 的任何真子集 X' 都不存在 $X' \rightarrow Y$ ，则称 Y 完全函数依赖于 X ，记作 $X \twoheadrightarrow Y$ ；否则，如果 $X \rightarrow Y$ ，且 X 中存在一个真子集 X' ，使得 $X' \rightarrow Y$ 成立，则 Y 部分函数依赖于 X ，记作 $X \twoheadrightarrow Y$ 。

二、完全函数依赖和部分函数依赖

例4: $(sno, cno) \xrightarrow{P} cname$

$(sno, cno) \xrightarrow{F} tname$

$cno \xrightarrow{F} cname$

$(sno, cno) \xrightarrow{F} grade$

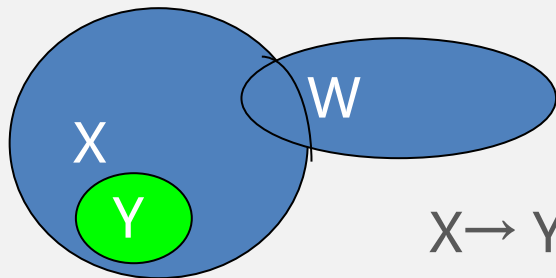
三、平凡函数依赖和非平凡函数依赖

设 X 、 Y 均为某关系上的属性集，且 $X \rightarrow Y$ ，则：

- (1) 若 Y 包含于 X ，则称 $X \rightarrow Y$ 为：平凡函数依赖；
- (2) 若 Y 不包含于 X ，则称 $X \rightarrow Y$ 为：非平凡函数依赖。（一般都指的是非平凡函数依赖）

三、平凡函数依赖和非平凡函数依赖

例5：设X、Y、W为关系R中的三个属性组，属性关系如下图所示，问 $X \rightarrow Y$ ， $X \rightarrow W$ ， $W \rightarrow Y$ 各属于平凡函数依赖还是非平凡函数依赖。



$X \rightarrow Y$ 为：平凡函数依赖

$X \rightarrow W$ 为：非平凡函数依赖

$W \rightarrow Y$ 为：非平凡函数依赖

练习1

已知关系R有属性ABCD，包括下表所示记录，
确定下列函数依赖在R上是否成立。

A	B	C	D
2	4	3	3
2	3	5	3
1	4	3	2
3	1	2	2

- (1) $A \rightarrow B$
- (2) $A \rightarrow BC$
- (3) $B \rightarrow C$

解答

已知关系R有属性ABCD，包括下表所示记录，
确定下列函数依赖在R上是否成立。

A	B	C	D
2	4	3	3
2	3	5	3
1	4	3	2
3	1	2	2

(1) $A \rightarrow B$ 不成立

(2) $A \rightarrow BC$ 不成立

(3) $B \rightarrow C$ 不确定

小结

函数依赖是关系模式规范化设计的核心。

函数依赖反映的是现实世界的一种语义，它不是关系模式R在某个时刻的关系（值）满足的约束条件，而是指R在任何时刻的一切关系均要满足的约束条件。



谢谢！



低级范式

总述

内容：关系模式的规范化程度可以用范式来区分，关系模式的规范化设计是指把一个低一级范式的关系模式分解为高一级范式的关系模式的过程。本知识点学习范式的基本概念以及第一范式和第二范式。

- 一、范式的概念
- 二、第一范式 (1NF)
- 三、第二范式 (2NF)

一、范式的概念

关系模式满足的确定约束条件称为**范式**，根据满足约束条件的级别不同，范式由低到高分为1NF、2NF、3NF、BCNF等。

满足不同程度的要求即为不同范式，满足最低要求的称为第一范式。

各种范式之间的关系可以表示为：

$$\text{BCNF} \subseteq 3\text{NF} \subseteq 2\text{NF} \subseteq 1\text{NF}$$

二、第一范式 (1NF)

关系模式的所有域为简单域，其元素不可再分，
即属性不能再分，每个属性是数据项而不是属性组。

例1: $R(A_1, A_2, A_3, \dots, A_k, \dots, A_n)$



例2: 工资(工号, 姓名, 工资(基本工资, 年绩
津贴, 煤电补贴))

二、第一范式 (1NF)

不满足1NF的关系称为非规范化关系。关系数据库模型不能存储上述两个例子。转化方法：

例1: $R (A_1, A_2, A_3, \dots, \underline{A_{k1}}, \underline{A_{k2}}, \dots, A_n)$

例2: 工资(工号, 姓名, 基本工资, 津贴, 奖金)

三、第二范式 (2NF)

概念回顾

超键：能唯一标识元组的属性集。

候选键(码)：不含有多余属性的超键。

主属性：包含在任何一个候选键中的属性。

非主属性：不包含在任何一个候选键中的属性。

三、第二范式 (2NF)

给定关系模式R及其上的函数依赖集F，如果R的任何一个非主属性都完全依赖于它的每个候选键，则称R是第二范式，简记为2NF。

定理：一个2NF的关系模式必定是1NF的。
($2NF \subseteq 1NF$)

三、第二范式 (2NF)

例3: 已知 $R\langle U, F \rangle$, $U=\{ABCD\}$,
 $F=\{AB\rightarrow C, C\rightarrow D\}$, 判断R属于第几范式。

R中的候选键为: **AB**

R为: **2NF**

原因: $AB\rightarrow C$, $AB\rightarrow D$, 不存在非主属性C和D对候选键AB的部分函数依赖。

三、第二范式 (2NF)

若给出关系模式的函数依赖集，如何确定关系模式中的候选键？

- (1) 如果有属性不在函数依赖集中出现，那么它必须包含在候选键中；
- (2) 如果有属性不在函数依赖集中任何函数依赖的右边出现，那么它必须包含在候选键中；
- (3) 如果该属性或属性组能唯一标识元组，则它就是候选键。

三、第二范式 (2NF)

例4: SCT (sno, cno, cname, grade, tname, salary)

$$F = \{(sno, cno) \rightarrow grade, cno \rightarrow cname, \\ cno \rightarrow tname, tname \rightarrow salary\}$$

候选键: (sno, cno) SCT为: 1NF

存在非主属性cname对候选键 (sno, cno)
的部分函数依赖 $cno \rightarrow cname$

异常: 冗余度高、插入问题、删除问题、修改困难

分解: SC (sno, cno, grade)

CT (cno, cname, tname, salary)

三、第二范式 (2NF)

分解后: SC (sno, cno, grade)

CT (cno, cname, tname, salary)

若R中所有候选键均为单属性, 此时1NF必为2NF。因为不可能存在某个非主属性部分函数依赖于候选键。

结论: 若 $R \in 1NF$, 且R中所有候选键均为单属性, 则 $R \in 2NF$ 。

练习1

判断对错。

(1) 在关系数据库中，对关系模式的基本要求是满足1NF。 ()

(2) 关系模式的规范化是把一个高级范式的关系模式分解为低一级范式的关系模式的过程。
()

解答

判断对错。

(1) 在关系数据库中，对关系模式的基本要求是满足1NF。 (对)

(2) 关系模式的规范化是把一个高级范式的关系模式分解为低一级范式的关系模式的过程。

(错)

小结

范式是评价关系模式好坏的标准。关系模式满足的最低要求是1NF，如果消除了非主属性对候选键的部分函数依赖，1NF就转化为2NF。

通过逐步消除不合适的函数依赖，使数据库中的各个关系模式达到某种程度的分离。



谢谢！



高级范式

总述

内容：关系模式的规范化程度可以用范式来区分，关系模式的规范化设计是指把一个低一级范式的关系模式分解为高一级范式的关系模式的过程。本知识点学习第三范式和BC范式。

- 一、低级范式回顾
- 二、第三范式 (3NF)
- 三、BC范式 (BCNF)

一、低级范式回顾

1NF：关系模式R的所有域为简单域，其属性不能再分。

2NF：关系模式R的任何一个非主属性都完全依赖于它的每个候选键。

二、第三范式 (3NF)

给定关系模式R及其上的函数依赖集F，如果R的任何一个非主属性都不传递函数依赖于它的任何一个候选键，则称R是第三范式，简记为3NF。(非主属性也不部分依赖于键)

定理：一个3NF的关系模式必定是2NF的。
($3NF \subseteq 2NF \subseteq 1NF$)

二、第三范式 (3NF)

例1: CT (cno, cname, tname, salary)

$F = \{cno \rightarrow tname, cno \rightarrow cname, tname \rightarrow salary\}$,

判断R属于第几范式。

存在非主属性salary传递函数依赖于候选键cno, 故CT \in 2NF, 不是3NF。

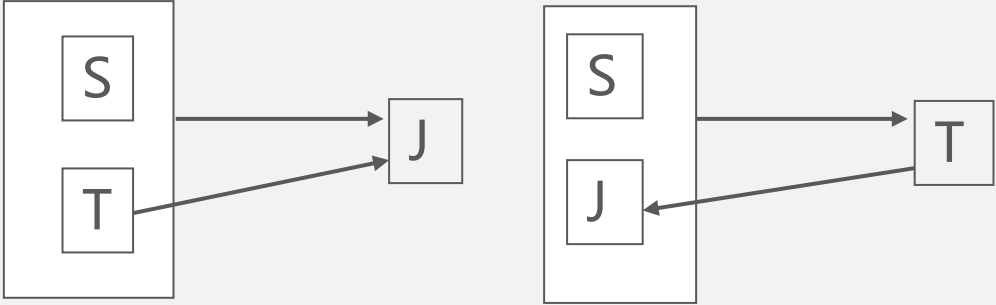
可分解为:

C (cno, cname, tname),

T (tname, salary) 均是3NF

二、第三范式 (3NF)

例2: $R=\{ S, T, J \}$ $F=\{ T \rightarrow J, SJ \rightarrow T, ST \rightarrow J \}$



候选键为: (S, T) 和 (S, J) 。

R中没有非主属性, 故R为3NF。

仍然存在插入异常等问题。

三、BC范式 (BCNF)

给定关系模式R及其上的函数依赖集F，如果F中每个非平凡函数依赖 $X \rightarrow Y$ 的左部（决定因素）X中必含有候选键，则称R是Boyce/Codd 范式，简记为BCNF（改进的3NF）。

$R \in 1NF$ ，若 $X \rightarrow Y$ 且 $Y \not\subseteq X$ ，X中必含有候选键，
则 $R \in BCNF$ 。

三、BC范式 (BCNF)

BCNF 含义

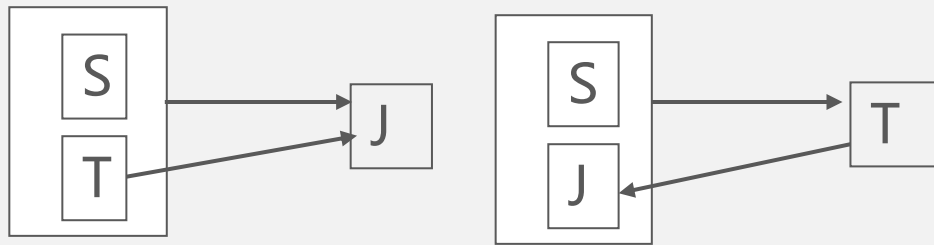
- (1) 非主属性对候选键完全函数依赖;
- (2) 非主属性不传递函数依赖于任何一个候选键;
- (3) 主属性对不含它的候选键完全函数依赖;
- (4) 主属性不传递函数依赖于任何一个候选键。

定理: BCNF满足3NF, 即:

($BCNF \subseteq 3NF \subseteq 2NF \subseteq 1NF$)

三、BC范式 (BCNF)

例3: $R=\{S, T, J\}$ $F=\{T \rightarrow J, SJ \rightarrow T, ST \rightarrow J\}$



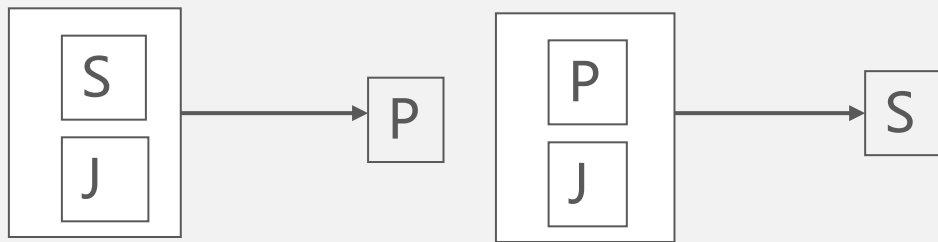
候选键为: (S, T) 和 (S, J) 。R中没有非主属性, 故R为3NF。

(1) 函数依赖 $T \rightarrow J$ 的左部不是候选键, 判断出R不是BCNF。

(2) 也可由 $T \rightarrow J$ 知, 存在主属性J对不包含它的候选键 (S, T) 的部分函数依赖, 判断R不是BCNF。

三、BC范式 (BCNF)

例4: $R = \{ S, P, J \}$ $F = \{ SJ \rightarrow P, JP \rightarrow S \}$



候选键为：(S, J) 和 (P, J)。不存在主属性对不包含它的候选键的部分和传递函数依赖，则 R 不但是 3NF，也是 BCNF。

或者说，每个函数依赖的左部都是候选键，则 R 满足 BCNF。

练习1

已知关系模式 $R(A,B,C,D)$ ，当满足下列函数依赖时，判断 R 是否达到BCNF。

(1) $BCD \rightarrow A, A \rightarrow C$ ()

(2) $ABD \rightarrow C$ ()

解答

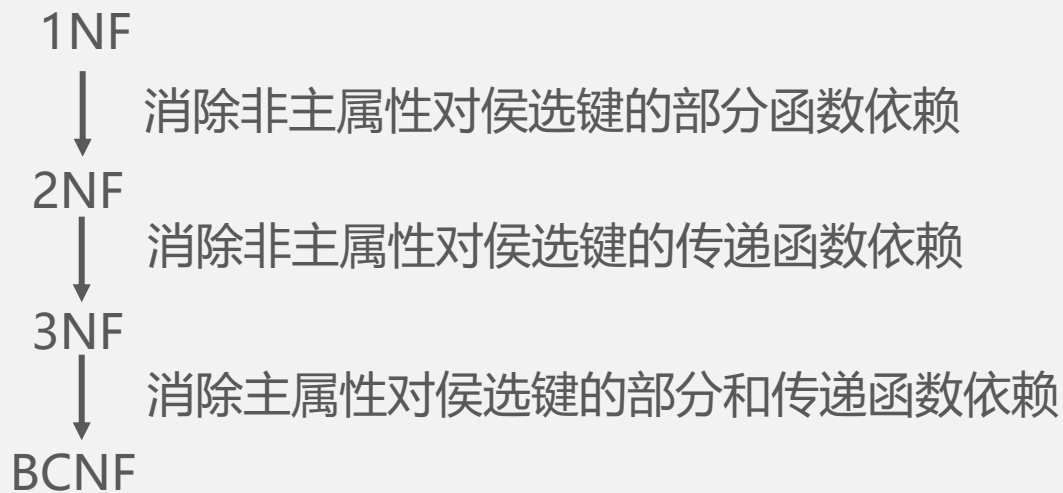
已知关系模式 $R(A,B,C,D)$ ，当满足下列函数依赖时，判断 R 是否达到BCNF。

(1) $BCD \rightarrow A, A \rightarrow C$ (3NF)

(2) $ABD \rightarrow C$ (BCNF)

小结

范式是评价关系模式好坏的标准。



逐步消除不合适的函数依赖，使数据库中的各个关系模式达到某种程度的分离。



谢谢！



数据依赖的公理系统

总述

内容：关系的规范化就是将低级范式的关系模式分解为高级范式的关系模式的过程，数据依赖的公理系统是模式分解算法的理论基础。本知识点学习数据依赖的公理系统。

- 一、函数依赖的逻辑蕴涵
- 二、Armstrong公理
- 三、属性集闭包
- 四、函数依赖集的等价
- 五、最小函数依赖集

一、函数依赖的逻辑蕴涵

定义1: F 逻辑蕴涵 $X \rightarrow Y$

对于关系模式 $R \langle U, F \rangle$, 其任何一个具体关系 r , 若函数依赖 $X \rightarrow Y$ 都成立 (即 r 中任意两元组 s, t , 若 $t[X]=s[X]$, 则 $t[Y]=s[Y]$), 则称函数依赖集 F **逻辑蕴涵** $X \rightarrow Y$ 。

例1: 关系模式 $R=(A,B,C)$, 函数依赖集 $F=\{A \rightarrow B, B \rightarrow C\}$, F 逻辑蕴涵 **$A \rightarrow C$** ...

一、函数依赖的逻辑蕴涵

定义2：函数依赖集F的闭包

在关系模式R $\langle U, F \rangle$ 中为F所逻辑蕴含的函数依赖的全体叫做**F的闭包**，记作 F^+ 。

例2：R=ABC, $F=\{A \rightarrow B, B \rightarrow C\}$, 求 F^+ 。

$F^+ = \{A \rightarrow \Phi, AB \rightarrow \Phi, AC \rightarrow \Phi, ABC \rightarrow \Phi, B \rightarrow \Phi, C \rightarrow \Phi$
 $A \rightarrow A, AB \rightarrow A, AC \rightarrow A, ABC \rightarrow A, B \rightarrow B, C \rightarrow C,$
 $A \rightarrow B, AB \rightarrow B, AC \rightarrow B, ABC \rightarrow B, B \rightarrow C,$
 $A \rightarrow C, AB \rightarrow C, AC \rightarrow C, ABC \rightarrow C, B \rightarrow BC,$
 $A \rightarrow AB, AB \rightarrow AB, AC \rightarrow AB, ABC \rightarrow AB, BC \rightarrow \Phi,$
 $A \rightarrow AC, AB \rightarrow AC, AC \rightarrow AC, ABC \rightarrow AC, BC \rightarrow B,$
 $A \rightarrow BC, AB \rightarrow BC, AC \rightarrow BC, ABC \rightarrow BC, BC \rightarrow C, \Phi \rightarrow \Phi$
 $A \rightarrow ABC, AB \rightarrow ABC, AC \rightarrow ABC, ABC \rightarrow ABC, BC \rightarrow BC \}$

二、Armstrong公理

定理：若 U 为关系模式 R 的属性全集， F 为 U 上的一组函数依赖，设 X 、 Y 、 Z 、 W 均为 U 的子集，对 $R(U, F)$ 有：

F1(自反律)：若 $Y \subset X$ (表示 X 包含 Y)，则
 $X \rightarrow Y$ 在 R 上成立；

F2(增广律)：若 $X \rightarrow Y$ 为 F 所蕴涵，则
 $XZ \rightarrow YZ$ 在 R 上成立；

F3(传递律)：若 $X \rightarrow Y, Y \rightarrow Z$ 为 F 所蕴涵，则
 $X \rightarrow Z$ 为 F 所蕴涵。

二、Armstrong公理

F4(伪增律): 若 $Z \subset W$, $X \rightarrow Y$ 为F所蕴涵,
则 $XW \rightarrow YZ$ 在R上成立;

F5(伪传律): 若 $X \rightarrow Y$, $YW \rightarrow Z$ 为F所蕴涵,
则 $XW \rightarrow Z$ 在R上成立;

F6(合并律): 若 $X \rightarrow Y, X \rightarrow Z$ 为F所蕴涵,
则 $X \rightarrow YZ$ 在R上成立;

F7(分解律): 若 $Z \subset Y$, $X \rightarrow Y$ 为F所蕴涵,
则 $X \rightarrow Z$ 在R上成立。

定理: $X \rightarrow A_1 A_2 \dots A_k$ 成立的充分必要条件是
 $X \rightarrow A_i$ 成立 ($i=1, 2, \dots, k$)

二、Armstrong公理

Armstrong公理是有效和完备的。

有效性指的是由F出发根据Armstrong公理推导出来的每一个函数依赖一定在 F^+ 中。

完备性指的是 F^+ 中的每一个函数依赖必定可以由F出发根据Armstrong公理推导出来

二、Armstrong公理

例3：已知关系模式R中

$U = \{A, B, C, D, E, G\}$,

$F = \{AB \rightarrow C, C \rightarrow A, BC \rightarrow D, ACD \rightarrow B, D \rightarrow EG,$

$BE \rightarrow C, CG \rightarrow BD, CE \rightarrow AG\}$,

判断 $BD \rightarrow AC$ 是否属于 F^+

解一：

(1) 由 $D \rightarrow EG$ 知 $D \rightarrow E$, $BD \rightarrow BE$;

(2) 由 $BE \rightarrow C$, $C \rightarrow A$ 所以 $BE \rightarrow A$, $BE \rightarrow AC$;

由(1)、(2)知 $BD \rightarrow AC$, 所以

$BD \rightarrow AC$ 被 F 所蕴涵, 即 $BD \rightarrow AC \in F^+$ 。

三、属性集闭包

定义：若 F 为关系模式 $R(U)$ 的函数依赖集， X 是 U 的子集，则由Armstrong公理推导出的所有 $X \rightarrow A_i$ 所形成的属性集 $\{A_i | i=1,2,\dots\}$ 称为 X 关于 F 的闭包，记为 X^+ 。

例4：

设 $R=ABC, F=\{A \rightarrow B, B \rightarrow C\}$ ，当 X 分别为 A, B, C 时，求 X^+ 。

解：1)当 $X=A$ 时， $X^+ = ABC$

2)当 $X=B$ 时， $X^+ = BC$

3)当 $X=C$ 时， $X^+ = C$

三、属性集闭包

定理： $X \rightarrow Y$ 能根据Armstrong推理规则导出的充要条件是 $Y \subseteq X^+$ 。

结论:

判定函数依赖 $X \rightarrow Y$ 是否能由 F 导出的问题，可转化为求 X^+ 并判定 Y 是否是 X^+ 子集的问题。即求闭包问题可转化为求属性集问题。

三、属性集闭包

例3：已知关系模式R中， $U=\{A, B, C, D, E, G\}$,
 $F=\{AB\rightarrow C, C\rightarrow A, BC\rightarrow D, ACD\rightarrow B, D\rightarrow EG,$
 $BE\rightarrow C, CG\rightarrow BD, CE\rightarrow AG\}$,

判断 $BD\rightarrow AC$ 是否属于 F^+ (AC 是否包含在 $(BD)^+$)

解二：

(1) $(BD)^+ = \{ BD \}$ 初值

(2) 找左部是B,D或BD的将右部并起来

由 $D\rightarrow EG$, 则 $(BD)^+ = \{ BDEG \}$

(3) 由 $BE\rightarrow C$, 则 $(BD)^+ = \{ BDEGC \}$

(4) 由 $CE\rightarrow AG$, 则 $(BD)^+ = \{ BDEGCA \}$

直到等于全集时算法终止, 即 $BD\rightarrow AC \in F^+$ 。

四、函数依赖集的等价

已知: $R1=ABC, F1=\{A \rightarrow B, B \rightarrow C\}$

$R2=ABC, F2=\{A \rightarrow B, B \rightarrow C, AC \rightarrow B\}$

问 $F1$ 和 $F2$ 是否等价?

定义: 设 F 和 G 是关系模式 $R(U)$ 上的两个函数依赖集,
如果 $F^+ = G^+$, 则称 **F 等价于 G** 。

定理: $F^+ = G^+$ 的充分必要条件是 $G \subset F^+$ 且 $F \subset G^+$, 则
 F 和 G 等价的充分必要条件是 $G \subset F^+$ 且 $F \subset G^+$ 。

五、最小函数依赖集

已知: $R1=ABC, F1=\{A \rightarrow B, B \rightarrow C\}$

$R2=ABC, F2=\{A \rightarrow B, B \rightarrow C, AB \rightarrow C, A \rightarrow C, A \rightarrow BC\}$

问 $F1$ 和 $F2$ 哪个是最小函数依赖集?

设 F 为函数依赖集, 如果 F 满足:

- (1) F 中每个函数依赖 $X \rightarrow Y$ 的右边 Y 均为单个属性;
- (2) F 中任何一个函数依赖 $X \rightarrow A$, 其 $F - \{X \rightarrow A\}$ 与 F 都不等价;
- (3) F 中任何一个函数依赖 $X \rightarrow A$, Z 为 X 的真子集,
($F - \{X \rightarrow A\}$) \cup $\{Z \rightarrow A\}$ 与 F 都不等价;

则称 F 为最小函数依赖集。

小结

规范化是将低级范式的关系模式分解为高级范式的关系模式的过程，模式分解要保持函数依赖关系，即两个函数依赖集要求等价。

函数依赖集等价的充要条件是其闭包相同。利用Armstrong公理及推论可以求出函数依赖集 F 的闭包，为使分解算法变得简单可用，将求函数依赖集 F 的闭包转化为求属性集的闭包。

数据依赖的公理系统是模式分解算法的理论基础。



谢谢！



模式的分解

总述

内容：关系的规范化就是将低级范式的关系模式分解为高级范式的关系模式的过程。本知识点学习模式分解的准则。

模式的分解

- 一、分解的几种情况
- 二、模式分解的准则

一、分解的几种情况

模式分解是把一个关系模式分解成若干个关系模式的过程。一个模式的分解是多种多样的，但分解后产生的模式应与原模式等价。

等价有三个不同的定义：

- (1) 仅具有无损联接性：信息不失真（不增减信息）；
- (2) 仅保持函数依赖集：不破坏属性间存在的依赖关系；
- (3) 既具有无损联接性，又保持函数依赖集。

一、分解的几种情况

例1: $R(ABC)$, $F=\{A \rightarrow B, B \rightarrow C\}$

候选键是A, R为2NF。

关系模式 $R(ABC)$ 的分解有四种情况。

A	B	C
a1	b1	c1
a2	b1	c1
a3	b2	c2
a4	b3	c1

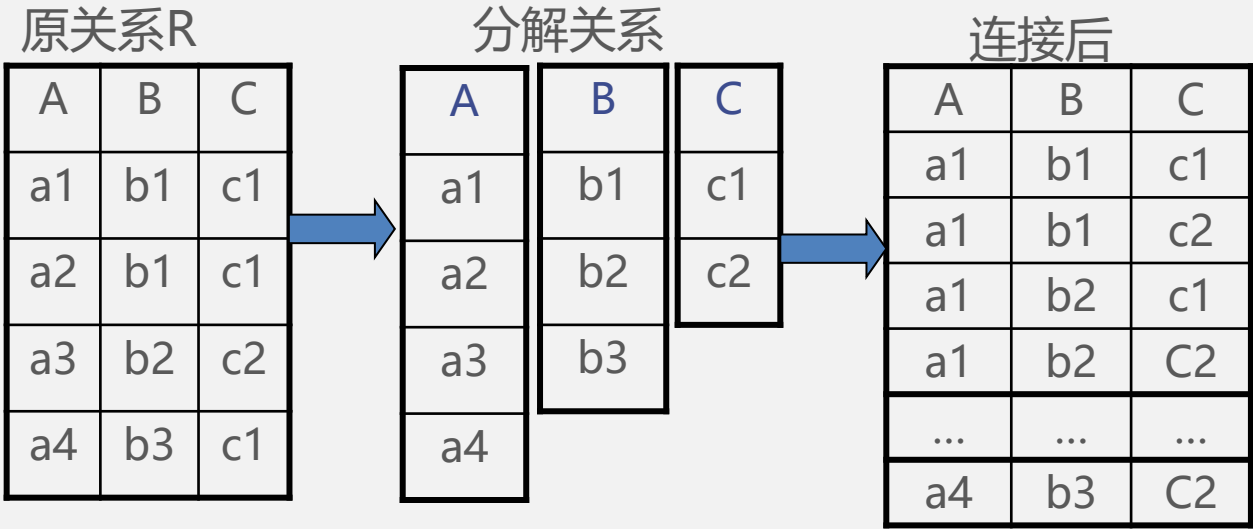
模式的分解

一、分解的几种情况

例1: $R(ABC)$, $F=\{A \rightarrow B, B \rightarrow C\}$

(1) 将R分解为 $P=\{R1(A), R2(B), R3(C)\}$

它不能保持函数依赖集F,
也不能恢复到原关系上。



模式的分解

一、分解的几种情况

例1: $R(ABC)$, $F=\{A \rightarrow B, B \rightarrow C\}$

(2) 将R分解为 $P=\{R1(AB), R2(AC)\}$

它不能保持函数依赖 $B \rightarrow C$,
但能恢复到原关系上。

原关系R

A	B	C
a1	b1	c1
a2	b1	c1
a3	b2	c2
a4	b3	c1

分解关系

A	B
a1	b1
a2	b1
a3	b2
a4	b3

A	C
a1	c1
a2	c1
a3	c2
a4	c1

连接后

A	B	C
a1	b1	c1
a2	b1	c1
a3	b2	c2
a4	b3	c1

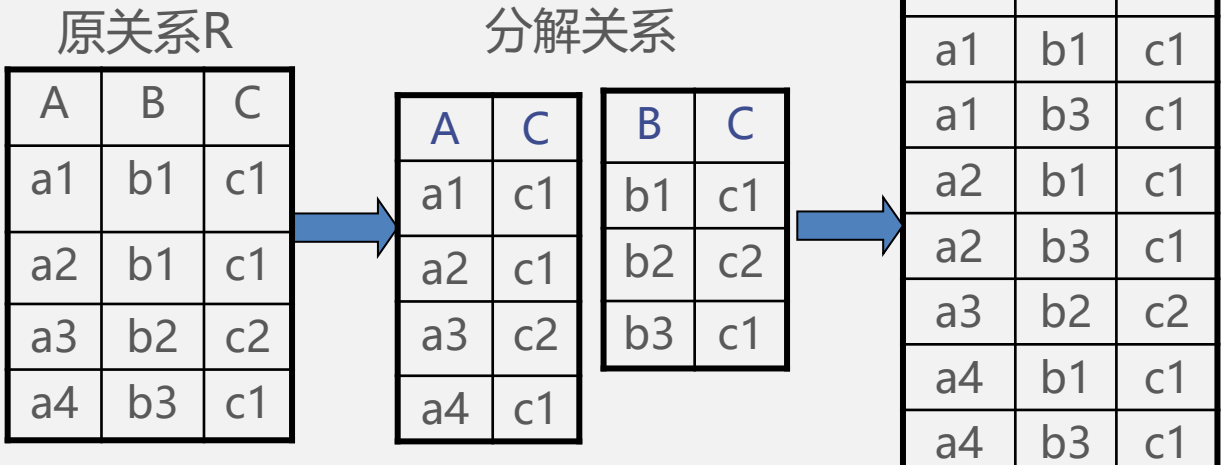
模式的分解

一、分解的几种情况

例1: $R(ABC), F=\{A \rightarrow B, B \rightarrow C\}$

(3) 将R分解为 $P=\{R1(AC), R2(BC)\}$

它不能保持函数依赖 $A \rightarrow B$,
不能恢复到原关系上。



模式的分解

一、分解的几种情况

例1: $R(ABC)$, $F=\{A \rightarrow B, B \rightarrow C\}$

(4) 将R分解为 $P=\{R1(AB), R2(BC)\}$

它不仅能保持函数依赖集F,
也能恢复到原关系上。

原关系R

A	B	C
a1	b1	c1
a2	b1	c1
a3	b2	c2
a4	b3	c1

分解关系

A	B
a1	b1
a2	b1
a3	b2
a4	b3

B	C
b1	c1
b2	c2
b3	c1

连接后

A	B	C
a1	b1	c1
a2	b1	c1
a3	b2	c2
a4	b3	c1

二、模式分解的准则

- (1) **仅具有无损联接性**：信息不失真（不增减信息）；
- (2) **仅保持函数依赖集**：不破坏属性间存在的依赖关系；
- (3) **既具有无损联接性，又保持函数依赖集。**

二、模式分解的准则

例2：分解关系模式

$R(\text{学号}, \text{课程号}, \text{成绩}, \text{成绩等级})$ 。

$F = \{ (\text{学号}, \text{课程号}) \rightarrow \text{成绩}, \text{成绩} \rightarrow \text{成绩等级} \}$

(1) $R_1(\text{学号}, \text{课程号}, \text{成绩})$,
 $R_2(\text{学号}, \text{课程号}, \text{成绩等级})$

(2) $R_1(\text{学号}, \text{课程号}, \text{成绩})$,
 $R_2(\text{成绩}, \text{成绩等级})$

小结

规范化就是将低级范式的关系模式分解为高级范式的关系模式的过程，分解后产生的模式应与原模式等价。

等价包括无损连接性和保持函数依赖两个含义。无损连接性和保持函数依赖集是两个互相独立的标准，具有无损连接性的分解不一定能保持函数依赖，反过来也是。



谢谢！