

西北工业大学

Northwestern Polytechnical University

数据库系统原理

Database System

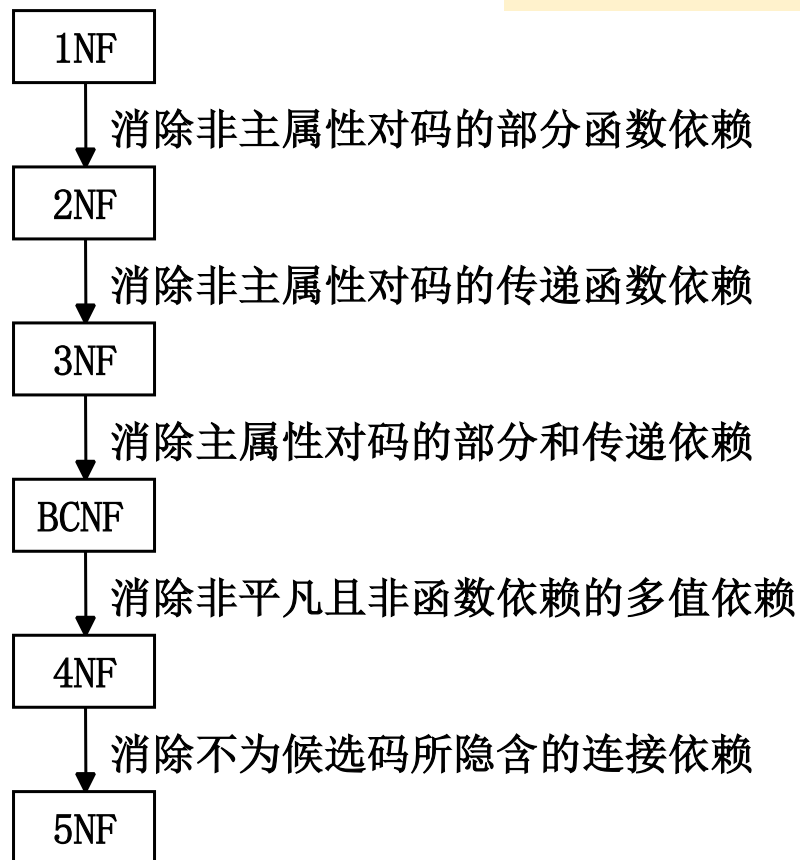
第六章 关系数据理论

赵晓南

2024.10

- 函数依赖
- 范式

数据库模式设计不好引起的四种异常：
更新异常、插入异常、删除异常、数据冗余



规范化的基本原则：

由低到高, 逐步规范, 权衡利弊, 适可而止。
通常, 以满足第三范式
(**3NF**) 为基本要求。

数据库设计满足的范式越高, 其数据处理的开销也越大

- 以下关系模式的设计是否合理？如何改进？

图书：（书号，书名，出版日期，出版社，
书架号，楼层号，楼层志愿者）

参考优化方案：

图书（书号，书名，出版日期，出版社，书架号）

书架（书架，楼层号）

楼层志愿者（楼层号，志愿者）

6.1.问题的提出

6.2.规范化

6.3.数据依赖的公理系统

6.4.保持函数依赖的模式分解

***6.5.无损连接的模式分解**



■ 函数依赖 (Armstrong) 公理系统

➤ 为什么需要这个公理系统？

FD中的部分依赖和传递依赖是不好的，需要消除。
如何知道所有的函数依赖呢？

➤ 主要任务是什么？

如何构造所有的函数依赖，提供一些定义、推理规则等

■ F的闭包

$F = \{X \rightarrow Y, Y \rightarrow Z\}$ $U = \{X, Y, Z\}$

$F^+ = \{$

$X \rightarrow \phi,$	$Y \rightarrow \phi,$	$Z \rightarrow \phi,$	$XY \rightarrow \phi,$	$XZ \rightarrow \phi,$	$YZ \rightarrow \phi,$	$XYZ \rightarrow \phi,$
$X \rightarrow X,$	$Y \rightarrow Y,$	$Z \rightarrow Z,$	$XY \rightarrow X,$	$XZ \rightarrow X,$	$YZ \rightarrow Y,$	$XYZ \rightarrow X,$
$X \rightarrow Y, Y \rightarrow Z,$			$XY \rightarrow Y,$	$XZ \rightarrow Y,$	$YZ \rightarrow Z,$	$XYZ \rightarrow Y,$
$X \rightarrow Z, Y \rightarrow YZ,$			$XY \rightarrow Z,$	$XZ \rightarrow Z,$	$YZ \rightarrow YZ,$	$XYZ \rightarrow Z,$
$X \rightarrow XY,$				$XY \rightarrow XY,$	$XZ \rightarrow XY,$	$XYZ \rightarrow XY,$
$X \rightarrow XZ,$				$XY \rightarrow YZ,$	$XZ \rightarrow XZ,$	$XYZ \rightarrow YZ,$
$X \rightarrow YZ,$				$XY \rightarrow XZ,$	$XZ \rightarrow XY,$	$XYZ \rightarrow XZ,$
$X \rightarrow XYZ,$			$XY \rightarrow XYZ,$	$XZ \rightarrow XYZ,$		$XYZ \rightarrow XYZ \}$

■ 函数依赖 (Armstrong) 公理系统

- **逻辑蕴含**：设关系模式 $R \langle U, F \rangle$ ，其中 F 是属性集 U 上的函数依赖集， $X, Y \subseteq U$ ，对其任何一个关系实例 r ，若函数依赖 $X \rightarrow Y$ 都成立，则称 F **逻辑蕴含** $X \rightarrow Y$
- **属性集的闭包**：设 F 为属性集 U 上的一组函数依赖， $X \subseteq U$ ， X 关于函数依赖集 F 的闭包 $X_F^+ = \{A \mid X \rightarrow A \text{ 能由 } F \text{ 根据 Armstrong 公理导出}\}$ 。
- **函数依赖集的闭包**：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作 F 的闭包 (Closure)，记为 F^+

■ Armstrong公理系统以及推理规则

➤ 公理系统

- A1自反律: 若 $Y \subseteq X \subseteq U$, 则 $X \rightarrow Y$ 为F所蕴含。 $(sno, sage) \rightarrow sno$
- A2增广律: 若 $X \rightarrow Y$ 为F所蕴含, 且 $Z \subseteq U$, 则 $XZ \rightarrow YZ$ 为F所蕴含。
- A3传递律: 若 $X \rightarrow Y$, $Y \rightarrow Z$ 为F所蕴含, 则 $X \rightarrow Z$ 为F所蕴含。

➤ 推理规则:

- B1合并规则: 由 $X \rightarrow Y$, $X \rightarrow Z$, 有 $X \rightarrow YZ$ 。
- B2伪传递规则: 由 $X \rightarrow Y$, $WY \rightarrow Z$, 有 $WX \rightarrow Z$ 。
- B3分解规则: 由 $X \rightarrow Y$, $Z \subseteq Y$, 有 $X \rightarrow Z$ 。

■ Armstrong公理系统以及推理规则

➤ 公理系统证明

重点：根据元组中的值相等关系

证明：设 r 是 $R(U, F)$ 的任一关系，有任意两个元组 $t, s \in r$

<A1正确性证明>：因为若 $t[X]=s[X]$ ， $Y \subseteq X$ ，则可推出 $t[Y]=s[Y]$ ，所以： $X \rightarrow Y$ 。

<A2正确性证明>：因为若 $t[XZ]=s[XZ]$ ，则应有 $t[X]=s[X]$ ， $t[Z]=s[Z]$ ；若 $t[YZ]=s[YZ]$ ，则应有 $t[Y]=s[Y]$ ， $t[Z]=s[Z]$ ；由 $X \rightarrow Y$ 可知若 $t[X]=s[X]$ ，则一定有 $t[Y]=s[Y]$ ；因此，若 $t[XZ]=s[XZ]$ ，则一定可推出 $t[YZ]=s[YZ]$ ，所以 $XZ \rightarrow YZ$ 。

<A3正确性证明>：因为：由 $X \rightarrow Y$ 可知若 $t[X]=s[X]$ ，则一定有 $t[Y]=s[Y]$ ；由 $Y \rightarrow Z$ 可知若 $t[Y]=s[Y]$ ，则一定有 $t[Z]=s[Z]$ ；因此，若 $t[X]=s[X]$ ，则一定可推出 $t[Z]=s[Z]$ ，所以 $X \rightarrow Z$ 。证毕。

➤ 推理规则证明

证明：

(B1) 合并律的正确性证明：由 $X \rightarrow Y$ 和增广律，可以推出 $X \rightarrow XY$ （注：两边都增加一个 X ）；由 $X \rightarrow Z$ 和增广律，可以推出 $XY \rightarrow YZ$ （注：两边都增加一个 Y ）；再由传递律，可得： $X \rightarrow YZ$ 。

(B2) 伪传递律的正确性证明：由 $X \rightarrow Y$ 和增广律，可以推出 $WX \rightarrow WY$ （注：两边都增加一个 W ）；又由 $WY \rightarrow Z$ 和传递律，可以推出 $XW \rightarrow Z$ 。

(B3) 分解律的正确性证明：由 $Z \subseteq Y$ 和自反律，可以推出， $Y \rightarrow Z$ ；再由 $X \rightarrow Y$ 及传递律，可以推出 $X \rightarrow Z$ 。证毕。

证明引：中国大学MOOC 数据库系统（中）：建模与设计（哈工大）



■ Armstrong公理系统的定理:

- 定理6.1: Armstrong推理规则是正确的。
- 定理6.2: Armstrong公理系统是有效的, 完备的。
 - 有效性: 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: 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。
- 定理6.3: 每一个函数依赖集F均等价于一个极小函数依赖集 F_m 。
注意: F_m 不一定是唯一的。



■ 属性闭包的求解算法

算法6.1：求属性集X关于函数依赖F的属性闭包 X^+

输入：R(U, F)中的U, F, X ($X \subseteq U$)

输出：X关于F的闭包 X^+ 。

方法：

- (1) 置初值如 $X(0) = \phi$, $X(1) = X$;
- (2) 如果 $X(0) \neq X(1)$, 置 $X(0) = X(1)$, 否则转4;
- (3) 对F中的每个函数依赖 $Y \rightarrow Z$, 若 $Y \subseteq X(1)$, 置 $X(1) = X(1) \cup Z$, 即将Y的右部并入X(1)中并转(2);
- (4) 输出 $X(1)$, 即为 X^+ ;

常见用途：1) 求超码

2) 通过验证 $B \in A_F^+$, 判断 $A \rightarrow B$ 在F上是否成立

■ 属性闭包的求解例题

例：设有关系模式R (A, B, C, D, E) , F由5个函数依赖组成：
 $F = \{AB \rightarrow C, B \rightarrow D, C \rightarrow E, EC \rightarrow B, AC \rightarrow B\}$, 求 $(AB)^+$.

解：由算法6.1:

	X (0)	X (1)	X (0) = X (1) ?	操作
第1次	ϕ	AB	否	把AB在左边可以导出的属性并入X(1)中
第2次	AB	ABCD	否	把ABCD在左边可以导出的属性并入X(1)中
第3次	ABCD	ABCDE	否	把ABCDE在左边可以导出的属性并入X(1)中
第4次	ABCDE	ABCDE	是	结束，输出结果为{ABCDE}

6.3.数据依赖的公理系统——属性闭包



- $R\langle U, F \rangle$
- $U = \{A, B, C, D, E, G\}$
- $F = \{AC \rightarrow B, CB \rightarrow D, A \rightarrow BE, E \rightarrow GC\}$

AB是 [填空1] BC是 [填空2]
AC是 [填空3] A是 [填空4]

- 1) 超码
- 2) 候选码
- 3) 非候选码

■ 函数依赖 (Armstrong) 公理系统

➤ 定义6.14 覆盖/等价

如果 $G^+ = F^+$ ，就说函数依赖集 F 覆盖 G (或者 F 与 G 等价)

➤ 定义6.15 最小依赖集/最小覆盖 (F_m) :

- (1) F 中任一函数依赖的右部仅含有一个属性;
- (2) F 中不存在这样的函数依赖 $X \rightarrow A$ ，使得 F 与 $F - \{X \rightarrow A\}$ 等价;
- (3) F 中不存在这样的函数依赖 $X \rightarrow A$ ， X 有真子集 Z 使得 $(F - \{X \rightarrow A\}) \cup \{Z \rightarrow A\}$ 与 F 等价。

例: $U = \{sno, school, mname, cno, grade\}$

$F = \{sno \rightarrow school, school \rightarrow mname, (sno, cno) \rightarrow grade\}$

$F_1 = \{sno \rightarrow school, sno \rightarrow mname, school \rightarrow mname, (sno, cno) \rightarrow grade, (sno, school) \rightarrow school\}$



■ 最小函数依赖集求解算法

算法6.2：求函数依赖集F的最小函数依赖 F_m
(定理6.3的证明过程)

(1) 分离每个函数依赖右部属性

对F中每一函数依赖 $X \rightarrow Y$ ，若 $Y = A_1 A_2 \dots A_m (m \geq 2)$ ，则用 $\{ X \rightarrow A_i \mid i=1, \dots, m \}$ 替换 $X \rightarrow Y$ ；

(2) 去掉多余函数依赖

对于F中的每一函数依赖 $X \rightarrow A$ ，若 $A \in X_{F - \{X \rightarrow A\}}^+$ ，则从F中去掉 $X \rightarrow A$ 。

(3) 去掉左部冗余属性

对F中每一函数依赖 $X \rightarrow A$ ，若 $X = B_1 B_2 \dots B_n$ ，逐一考察 B_i ，若 $A \in (X - B_i)_{F^+}$ ，则用 $(X - B_i) \rightarrow A$ 替换 $X \rightarrow A$ ， B_i 称为冗余属性；



■ 最小函数依赖集求解例题

例：设关系模式 $R(ABCDE)$ 上的函数依赖集 $F=\{A\rightarrow BC, BCD\rightarrow E, B\rightarrow D, A\rightarrow D, E\rightarrow A\}$ ，求 F 的最小函数依赖集。

解：由算法6.2的第一步：

对 F 中每一函数依赖 $X\rightarrow Y$ ，若 $Y=A_1A_2\dots A_m (m\geq 2)$ ，则用 $\{X\rightarrow A_i \mid i=1, \dots, m\}$ 替换 $X\rightarrow Y$ ；

第一步：分离每个函数依赖右部属性

需要分解的函数依赖：右部由多个属性构成的函数依赖
本题中： $A\rightarrow BC$

分解之后得：

$F_1=\{A\rightarrow B, A\rightarrow C, BCD\rightarrow E, B\rightarrow D, A\rightarrow D, E\rightarrow A\}$



■ 最小函数依赖集求解例题

前提: $F1 = \{A \rightarrow B, A \rightarrow C, BCD \rightarrow E, B \rightarrow D, A \rightarrow D, E \rightarrow A\}$

由算法6.2的第二步:

对于F中的每一函数依赖 $X \rightarrow A$, 若 $A \in X_{F - \{X \rightarrow A\}}^+$, 则从F中去掉 $X \rightarrow A$

第二步: 去掉多余函数依赖

1) 考察 $A \rightarrow B$:

设 $F0 = F - \{A \rightarrow B\} = \{A \rightarrow C, BCD \rightarrow E, B \rightarrow D, A \rightarrow D, E \rightarrow A\}$

所以 $(A)_{F0}^+ = \{ACD\}$, 而B不属于 $\{ACD\}$, 所以 $A \rightarrow B$ 不多余, 不能删

...

2) 考察 $A \rightarrow D$:

设 $F0 = F - \{A \rightarrow D\} = \{A \rightarrow B, A \rightarrow C, BCD \rightarrow E, B \rightarrow D, E \rightarrow A\}$

所以 $(A)_{F0}^+ = \{ABCED\}$, D属于 $\{ABCED\}$, 所以 $A \rightarrow D$ 多余, 要删除

3) 考察 $E \rightarrow A$

设 $F0 = \{A \rightarrow B, A \rightarrow C, BCD \rightarrow E, B \rightarrow D, E \rightarrow A\} - \{E \rightarrow A\}$

$(E)_{F0}^+ = \{E\}$, A不属于其中, 所以 $E \rightarrow A$ 不多余

最终得: $F2 = \{A \rightarrow B, A \rightarrow C, BCD \rightarrow E, B \rightarrow D, E \rightarrow A\}$

删除了 $A \rightarrow D$



■ 最小函数依赖集求解例题

前提: $F_2 = \{A \rightarrow B, A \rightarrow C, \text{BCD} \rightarrow E, B \rightarrow D, E \rightarrow A\}$

由算法6.2的第三步:

对 F 中每一函数依赖 $X \rightarrow A$, 若 $X = B_1 B_2 \dots B_n$, 逐一考察 B_i , 若 $A \in (X - B_i)_F^+$, 则用 $(X - B_i) \rightarrow A$ 替换 $X \rightarrow A$, B_i 称为冗余属性;

第三步: 去掉左部冗余属性

1) 考察左部为多个属性的函数依赖: 本题只有一个 $\text{BCD} \rightarrow E$

2) 对于 $\text{BCD} \rightarrow E$ 中的每一个属性分别考察

B: $(\text{CD})_{F_2}^+ = \{\text{CD}\}$, E不属于 $\{\text{CD}\}$, 不能代换

C: $(\text{BD})_{F_2}^+ = \{\text{BD}\}$, E不属于 $\{\text{BD}\}$, 不能代换

D: $(\text{BC})_{F_2}^+ = \{\text{BCDEA}\}$, E属于 $\{\text{BCDEA}\}$,

所以用 $\text{BC} \rightarrow E$ 代换 $\text{BCD} \rightarrow E$

所以: $F_m = \{A \rightarrow B, A \rightarrow C, \text{BC} \rightarrow E, B \rightarrow D, E \rightarrow A\}$



■ 最小函数依赖集求解练习题

已知 $U = \{A, B, C, D\}$,

$F = \{A \rightarrow BC, B \rightarrow C, A \rightarrow B, AB \rightarrow D\}$, 求 F_m 。

答案:

$$F_m = \{A \rightarrow B, B \rightarrow C, A \rightarrow D\}$$

注意: F 的最小函数依赖集不是唯一的, 与计算顺序有关。



■ 候选码的基本求解方法

分析：

对于给定的关系模式 $R\langle U, F \rangle$ ，依照函数依赖集 F 将 U 中的属性分为以下四类：

L类属性：在 F 中只出现在函数依赖的左部的属性；

R类属性：在 F 中只出现在函数依赖的右部的属性；

LR类属性：分别出现在 F 中的函数依赖左部和右部的属性；

N类属性：不在 F 中的函数依赖中出现的属性。

结论：

- ① L类属性和N类属性**必包含于**任何候选码中；
- ② R类属性**不包含于**任何候选码中；
- ③ LR类属性**不能确定**是否在候选码中。



■ 候选码的基本求解方法

- (0) 求关系模式 $R \langle U, F \rangle$ 的**最小**函数依赖集 F
- (1) 依照函数依赖集 F 将 R 中的所有属性分为L类、R类、LR类和N类属性，令 X 为L、N类属性的集合， Y 为LR类属性集合；
- (2) 若 $X_F^+ = U$ ，则 X 为 R 的唯一候选码，结束；否则转(3)；
- (3) 逐一取 Y 中的单一属性 A ，若 $(XA)_F^+ = U$ ，则 XA 为候选码，令 $Z = Y - \{A\}$ ，转(4)；
- (4) 依次取 Z 中的任意两个、三个……属性与 X 组成属性组，若 XZ 不包含已求得的候选码，求其关于 F 的闭包 $(XZ)_F^+$ ，若 $(XZ)_F^+ = U$ ，则 XZ 为候选码。直到取完 Z 中的所有属性为止，算法结束。



例：设 $R(A, B, C, D, E)$ ， $F = \{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$ ，求 R 的所有候选码。

解：(1) R 中无 L 、 N 类属性， $ABCDE$ 均为 LR 类属性；

(2) 取 A ，则 $A_F^+ = ABCDE = U$ ，**A**为候选码；

取 B 、 C 、 D ，其闭包均不等于全属性集 U ；

取 E ，则 $E_F^+ = ABCDE = U$ ，**E**为候选码；

(3) 在 $\{BCD\}$ 中任取两个属性判定

$(BC)_F^+ = BCDEA = U$ ，**BC**为候选码；

$(BD)_F^+$ 不等于全属性集 U ， BD 不是候选码；

$(CD)_F^+ = CDEAB = U$ ，**CD**为候选码；

(4) BCD 包含已求得的候选码 BC ， BCD 不是码，结束。

故关系 R 的候选码为：**A, E, BC, CD**。

6.3.数据依赖的公理系统——候选码求解



https://gitee.com/formalization/schema_decomposition



基于Armstrong公理系统的自动模式分解工具

开发人员：李建清
指导教师：蒋建民教授、赵卓宁教授
单位：成都信息工程大学-软件自动生成与智能服务四川省重点实验室

功能按钮

- 计算闭包
- 计算关键字
- 计算主码
- 计算候选码
- 计算最小依赖
- 模式分解

输入框

A->BC, BCD->E, B->D, A->D, E->A

运算结果

【求主码】
结果为：A

功能按钮

- 计算闭包
- 计算增广集
- 计算传递依赖
- 计算闭包
- 计算关键字
- 计算最小依赖
- 模式分解

功能按钮

- 计算闭包
- 计算关键字
- 计算最小依赖
- 分解依赖
- 去除多余依赖
- 去除多余属性
- 计算最小依赖集
- 模式分解

功能按钮

- 计算闭包
- 计算关键字
- 计算最小依赖
- 模式分解
- 分解为2NF
- 分解为3NF
- 分解为BCNF
- 分解过程展示

schema_decomposition-master > js

名称	修改日期
DatabaseTool.js	2021/7/28 21:12
jquery-1.3.2.js	2021/7/28 21:12
mainStyle.js	2021/7/28 21:12

可以进一步改进？

6.1.问题的提出

6.2.规范化

6.3.数据依赖的公理系统

6.4.保持函数依赖的模式分解

***6.5.无损连接的模式分解**

6.4 模式分解



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

原始关系r

NL		DL	
Sno	Sloc	Sloc	Sdept
950001	A	A	CS
950002	B	B	IS
950003	C	C	MA
950004	B	B	PH
950005	B		

分解：2个关系r1和r2

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

连接： $m_p(r) = r1 \bowtie r2$

1. 分解的无损连接性：

数据内容：原始关系r $\neq m_p(r)$: 有损连接（未能保持内容等价）

[示例] R(C, S, Z), C是城市, S是街区, Z是邮政编码

$F = \{ CS \rightarrow Z, Z \rightarrow C \}$

$\rho = \{R_1(SZ), R_2(CZ)\}$

CS \rightarrow Z 走丢了？

2. 分解的保持依赖性：未能保持数据之间的函数依赖

6.4 保持函数依赖的模式分解



将S-L分解为下面二个关系模式：

ND(Sno, School)

NL(Sno, Sloc)

分解后的关系为：

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

存在问题：
若某学生转学院CS-IS，
不得不同步修改NL中
的信息

1. 没有丢失信息=>无损连接性；2. 但是未保持school→sloc函数依赖

ND(Sno, School), Sno→School

DL (School, Sloc), School→Sloc

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

ND ⋈ NL

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



1.无损
2.保持函数依赖

- 如何判断模式分解是否合理？
 - 如何判断分解是否保持依赖？ 引理6.3 (P188)
 - 如何判断分解是否无损连接？ 算法6.3 (P192)

 - 如何进行模式分解：
 - 6.4节： **算法6.2：转换为3NF，且保持依赖**
 - *6.5节
- 算法6.4：转换为3NF，无损连接且保持函数依赖。
算法6.5：转换为BCNF，无损连接。
算法6.6：转换为4NF，无损连接。

6.4 保持函数依赖的模式分解



➤ 如何判断模式分解是否合理？—— 保持依赖判断

Input : $R(A, B, C, D, E)$

$F = \{ A \rightarrow C, B \rightarrow C, C \rightarrow D, DE \rightarrow C, CE \rightarrow A \}$

$\rho = \{ R_1(AC), R_2(BC), R_3(CDE) \}$

Output : ρ 是否是保持依赖的判断

Method : 依据题意

$\pi_{R_1}(F) = \{ A \rightarrow C \}, \pi_{R_2}(F) = \{ B \rightarrow C \}, \pi_{R_3}(F) = \{ C \rightarrow D, DE \rightarrow C \}$

$G = \{ A \rightarrow C, B \rightarrow C, C \rightarrow D, DE \rightarrow C \}$, 显然不保持依赖。

判断F中的每个函数依赖是否被G逻辑蕴含
(判断F和G是否等价)

- 1) 对F中的函数依赖 $A \rightarrow C$, 计算箭头左侧属性的 $(A)^+_G = \{A, C\}$, 由于箭头右侧的C被包含其中, 所以 $A \rightarrow C$ 被G逻辑蕴涵。
- 2) 其他F中的 $B \rightarrow C, C \rightarrow D$ 与上面同理, 都被G逻辑蕴涵。
- 3) 对F中的函数依赖 $DE \rightarrow C$, 计算箭头左侧属性的 $(DE)^+_G = \{D, E, C\}$, 由于箭头右侧的C被包含其中, 所以 $DE \rightarrow C$ 被G逻辑蕴涵。
- 4) 对F中的函数依赖 $CE \rightarrow A$, 计算箭头左侧属性的 $(CE)^+_G = \{C, E, D\}$, 由于箭头右侧的A不被包含其中, 所以 $CE \rightarrow A$ 不被G逻辑蕴涵。

以上分析, 可知G的分解没有保持函数依赖

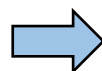
6.4 保持函数依赖的模式分解



➤ 如何判断模式分解是否合理？—— 无损连接判断(算法6.3)

已知 $R = \langle U, F \rangle$, $R = \{A, B, C, D, E\}$, $F = \{AB \rightarrow C, C \rightarrow D, D \rightarrow E\}$,
R的一个分解: $R_1 = (A, B, C)$, $R_2 = (C, D)$, $R_3(D, E)$

A	B	C	D	E
a1	a2	a3	b14	b15
b21	b22	a3	a4	b25
b31	b32	b33	a4	a5



A	B	C	D	E
a1	a2	a3	a4	a5
b21	b22	a3	a4	a5
b31	b32	b33	a4	a5

行i: 分解的R个数
列j: R中的属性个数
元素: 若分解后的 R_i 中包含该属性j, 则对应分量 a_j , 否则为 b_{ij}

出现全a行, 则为无损连接

1.构造初始数据 (如上左表)

2.逐步替换: 判断每个函数依赖

- ① $AB \rightarrow C$: 左侧对应列无相同元素
- ② $C \rightarrow D$: 有两个a3, 右侧替换为相同的元素, 有a则替换为a (a4)
- ③ $D \rightarrow E$: 有两个a4, 右侧替换为相同的元素, 有a则替换为a (a5)

6.4 保持函数依赖的模式分解



➤ 算法6.2（合成法）转换为3NF的保持函数依赖的分解

1. $R(U, F)$: F 进行极小化处理，找到最小函数依赖集。
2. 找出所有不在 F 中出现的属性（记为 U_0 ），构成一个关系模式 $R_0(U_0, F_0)$ 。把这些属性从 U 中去掉，剩余的属性仍记为 U_0 。
3. 若有 $X \rightarrow A \in F$ ，且 $XA = U$ ，则 $\rho = \{R\}$ ，算法终止。
4. 否则，对 F 按具有相同左部的原则分组（假定分 k 组），每一组函数依赖所涉及的全部属性形成一个属性集 U_i 。若 $U_i \subseteq U_j (i \neq j)$ 就去掉 U_i 。由于经过了步骤2，故 $U = \bigcup U_i$ （并集， i 从1到 k ），于是 $\rho = \{R_1(U_1, F_1), \dots, R_k(U_k, F_k)\} \cup R_0(U_0, F_0)$ ，构成 $R(U, F)$ 的一个保持函数依赖的分解。

作业：根据以上算法，请自行举例说明算法的具体运行过程。

6.4 保持函数依赖的模式分解



➤ 算法6.2（合成法）转换为3NF的保持函数依赖的分解

[Algorithm]保持依赖分解成3NF的算法。

Input : 关系模式 $R(U, F)$, F 是函数依赖集最小覆盖。

Output : R 的一个保持依赖分解 ρ , ρ 中的每个关系模式都是 F 在该模式上投影的3NF。

Method : (1)把 R 中不出现在 F 中的属性去掉并单独组成一模式。

(2)对 $\forall X \rightarrow A \in F$, 则以 XA 组成一模式; 若有 $X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_m$ 都属于 F , 则以 $XA_1A_2 \dots A_m$ 组成一模式(即将 n 个模式合并为一个模式)。

(3)取 ρ 为上述模式之集合, 则 ρ 即为所求之分解。

例关系 R : $U = A, B, C, D, E, F, G$

$F = \{A \rightarrow B, A \rightarrow C, C \rightarrow D, C \rightarrow E, E \rightarrow FG\}$

保持函数依赖的分解:

1) $R_1(A, B), R_2(A, C), R_3(C, D), R_4(C, E), R_5(E, F, G)$ 【相同左部为一组】

2) $R_{12}(A, B, C), R_{34}(C, D, E), R_5(E, F, G)$ 【合并相同的左部】

最终的 $\rho = \{R_{12}, R_{34}, R_5\}$ 即为本次分解得到的结果。

可以自行验证以上分解 $\rho = \{R_{12}, R_{34}, R_5\}$, 即保持了函数依赖 (该分解也保持了无损连接) 。

■ 练习

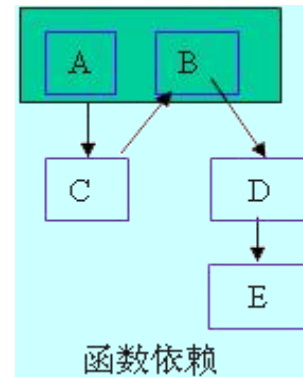
- $U = \{A, B, C, D, E\}$
- $F = \{AB \rightarrow C, B \rightarrow D, D \rightarrow E, C \rightarrow B\}$
 - 哪个是候选码?
 - 是第几范式?为什么?
 - 怎么改进?

■ 练习解答

$$F = \{AB \rightarrow C, B \rightarrow D, D \rightarrow E, C \rightarrow B\}$$

- 候选码是 (A, B), (A, C)
- R最高属于1NF

理由：关系模式R(U, F)中，函数依赖图解如上图，
关系模式R的主属性是ABC，每个分量都是不可再分，因此是1NF；
非主属性存在对码的部分函数依赖 $B \rightarrow D$ ，所以不是2NF；
非主属性对码的传递函数依赖 $B \rightarrow D, D \rightarrow E$ ，所以也不是3NF；
因此最终该关系是1NF。



- 改进方法：分解R为符合第三范式或BCNF范式的关系模式集合如下：
R1 { (A, B, C), $AB \rightarrow C, C \rightarrow B$ }
R2 { (B, D), $B \rightarrow D$ }
R3 { (D, E), $D \rightarrow E$ }

- 属性闭包和最小函数依赖集的求解
- 候选码的求解方法
- 模式分解的思想

作业： 第六章 第2题，第6题，第8题

要求：第2题

- (1)所有的关系名和属性名使用题目所给的中文名称；
- (2)列出的每个关系模式所属范式并说明原因。
- (3)指出各关系的候选码，外部码，有没有全码存在。

补充作业：针对课本算法6.2（保持函数依赖的3NF转换分解算法），用例子说明该算法运行过程。算法6.2参考下页。

➤ 算法6.2（合成法）转换为3NF的保持函数依赖的分解

1. $R(U, F)$: F 进行极小化处理，找到最小函数依赖集。
2. 找出所有不在 F 中出现的属性（记为 U_0 ），构成一个关系模式 $R_0(U_0, F_0)$ 。把这些属性从 U 中去掉，剩余的属性仍记为 U_0 。
3. 若有 $X \rightarrow A \in F$ ，且 $XA = U$ ，则 $\rho = \{R\}$ ，算法终止。
4. 否则，对 F 按具有相同左部的原则分组（假定分 k 组），每一组函数依赖所涉及的全部属性形成一个属性集 U_i 。若 $U_i \subseteq U_j (i \neq j)$ 就去掉 U_i 。由于经过了步骤2，故 $U = \bigcup U_i$ （并集， i 从1到 k ），于是 $\rho = \{R_1(U_1, F_1), \dots, R_k(U_k, F_k)\} \cup R_0(U_0, F_0)$ ，构成 $R(U, F)$ 的一个保持函数依赖的分解。

补充作业：根据以上算法，请自行举例说明算法的具体运行过程。

