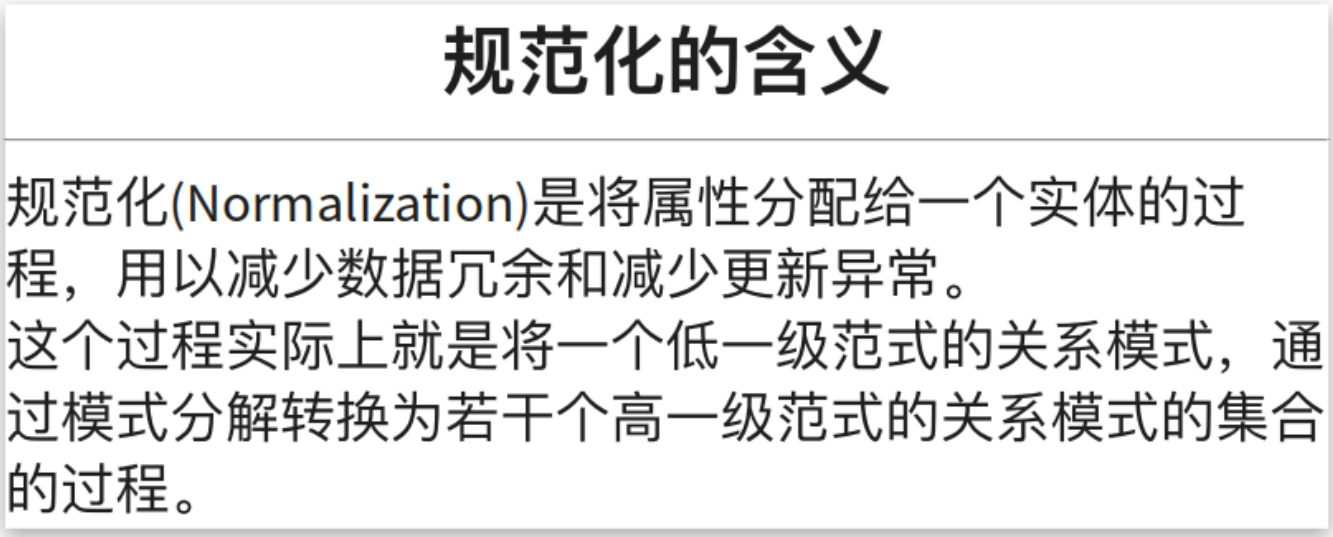
- 规范化的含义

> 
>

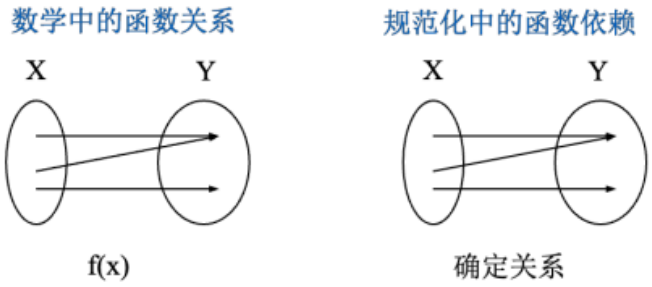
函数依赖

函数依赖定义

- 定义

  函数依赖(Functional Dependency)定义

  设$R(U)$是属性集U上的关系模，X、Y是U的子集。
  若对于$R(U)$的任意一个可能的关系r，r中不可能存在两个元组在X上的属性值相等，而在Y上的属性值不等，则称X函数确定Y或Y函数依赖于X，记作$X \rightarrow Y$。

- 一些通俗的理解

  函数依赖

  函数依赖与属性间的关系

  - 若X、Y是$1:1$关系，则存在$X \rightarrow Y$或$Y \rightarrow X$
    如 学号与借书证号
  - 若X、Y是m:1关系，则存在$X \rightarrow Y$但$Y \nrightarrow X$
    如 学号与姓名
  - 若X、Y是m:n关系，则X、Y间不存在函数依赖关系
    如 姓名与课程

函数依赖

函数依赖另一种定义

若R的任意关系有：对X中的每个属性值，在Y中都有惟一的值与之对应，则称Y函数依赖于X, 记作$X \rightarrow Y$, X叫做决定因素。

若$X \rightarrow Y, Y \rightarrow X$, 则记作$X \leftrightarrow Y$。

如果Y不函数依赖于X, 则记做$X \nrightarrow Y$。

- 函数依赖是语义范畴的概念，只能通过语义确定
- 函数依赖要求对关系模式的**所有关系**都成立

- 例子

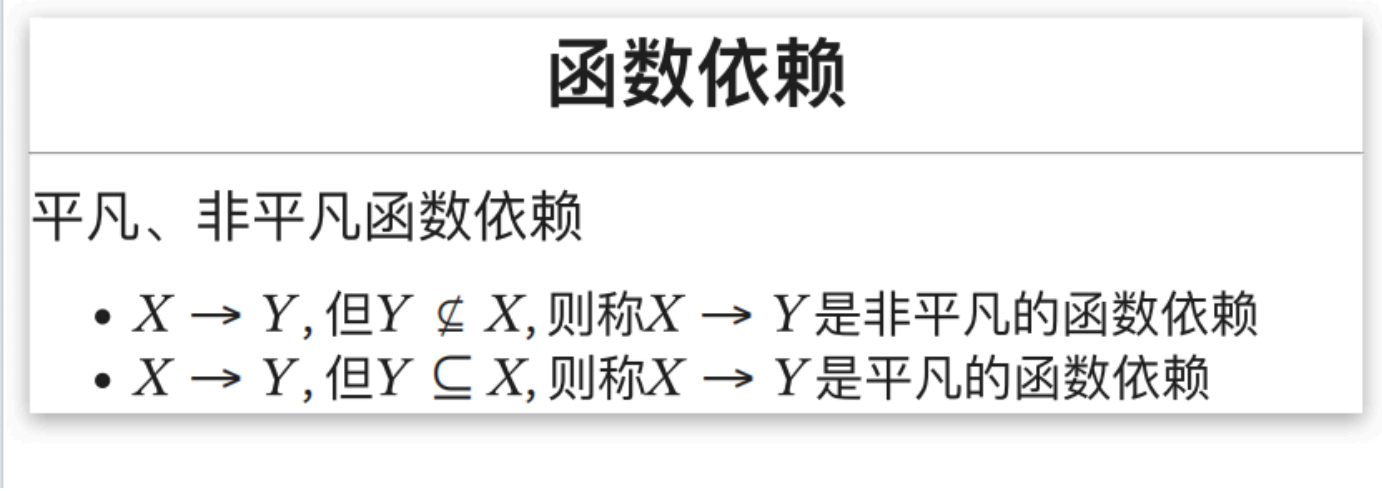
函数依赖

例 试指出学生关系模式 Student(sno, sname, class, cno, tno, tname, tage, taddr, grade)中存在的函数依赖。

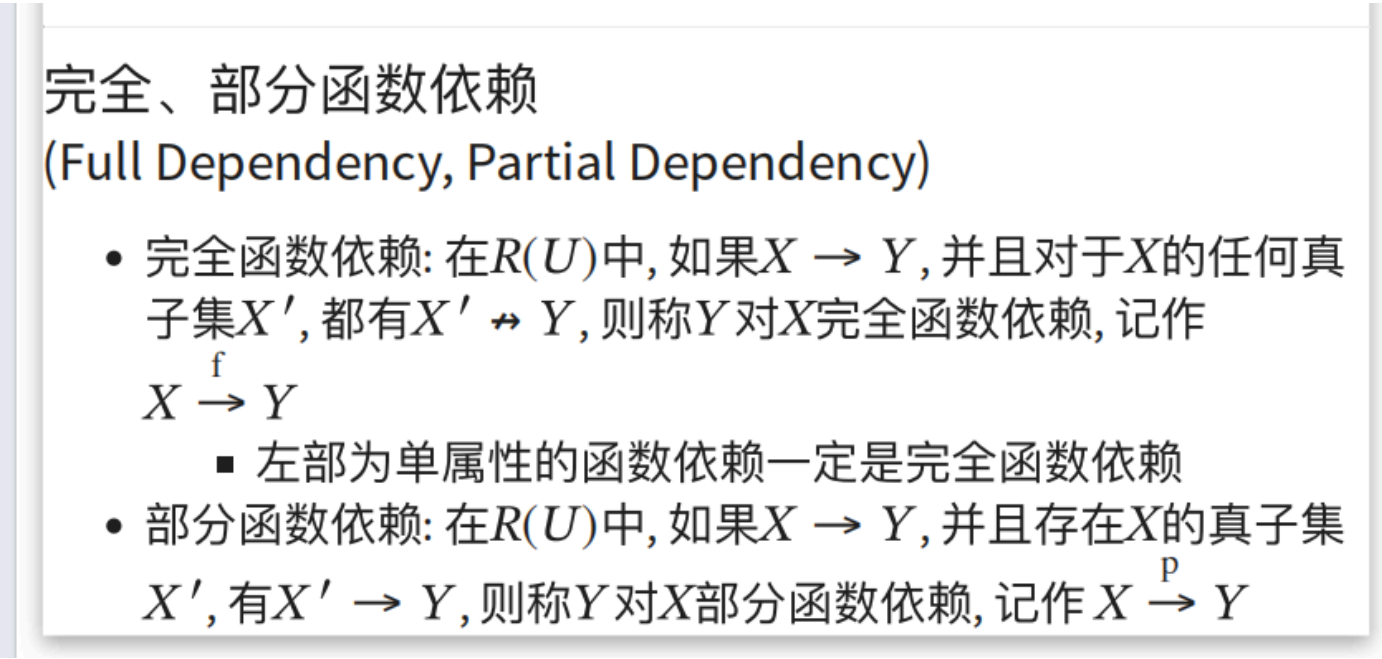
- sno → sname, 每个学号只能有一个学生姓名
- sno → class, 每个学号只能有一个班级
- cno → tno, 设每门课只有一个教师任教，而一个教师可教多门课
- tno → tname, 每个教师只能有一个姓名
- tno → tage, 每个教师只能有一个年龄
- tno → taddr, 每个教师只能有一个地址
- (sno, cno) → grade, 每个学生学习一门课只能有一个成绩
- (sno, cno) → sname
- (sno, cno) → class
- (sno, cno) → cno
- (sno, cno) → tname
- (sno, cno) → tage
- (sno, cno) → taddr

一些特别的函数依赖

- 平凡与非平凡函数依赖



- 完全函数依赖与部分函数依赖 **我的理解：完全函数依赖就是对于X的所有子集都不能函数确定Y，也就是X是个最小单元，此时为完全函数依赖。如果X的子集有函数确定Y的，也就是X不是最小单元，那就是部分函数依赖。**



- 一个例子

函数依赖

左部为多属性函数依赖，如何判断其是否为完全函数依赖?
方法：取真子集，看其能否决定右部属性

例 试指出学生关系模式 Student(sno, sname, class, cno, tno, tname, tage, taddr, grade)中存在的完全函数依赖和部分函数依赖

sno → sname、 sno → class、 tname → tage、
tname → taddr、 cno → tname 都是完全函数依赖

(sno, cno) → grade是一个完全函数依赖,
因为 sno ↛ grade、 sno ↛ grade

函数依赖

左部为多属性函数依赖，如何判断其是否为完全函数依赖?
方法：取真子集，看其能否决定右部属性

例 试指出学生关系模式 Student(sno, sname, class, cno, tno, tname, tage, taddr, grade)中存在的完全函数依赖和部分函数依赖

(sno, cno) → sname、(sno, cno) → class、(sno, cno) → tname、
(sno, cno) → tage、(sno, cno) → taddr 都是部分函数依赖, 因为
sno → sname、 sno → class、 cno → tname、 cno → tage、
cno → taddr

- 传递函数依赖

函数依赖

传递函数依赖

定义：在$R(U)$中，如果$X \to Y(Y \subsetneq X)$，$Y \nrightarrow X$，且$Y \to Z$，则称Z对X传递函数依赖，记作：$X \xrightarrow{t} Z$。

例 试指出学生关系模式 Student(sno, sname, class, cno, tno, tname, tage, taddr, grade)中存在的传递函数依赖。

因为cno → tno，tno → tname，
所以cno → tname是一个传递函数依赖。
类似地，cno → taddr也是一个传递函数依赖。

候选键

- 抽象思维地想，一个候选键组唯一确定sql中的每一行，也就是X完全函数依赖U

函数依赖

候选键的形式定义

设X为$R(U, F)$中的属性或属性组，若$X \xrightarrow{f} U$, 则X为R的候选键。

- 存在性与不唯一性
- 主属性: 包含在任何一个侯选键中的属性
- 非主属性：不包含在任何一个候选键中的属性
- 全键：整个属性组为键
  例 R(customer, commodity, pdate)

范式

- 定义

范式(NORMAL FORMS)

定义：关系数据库中符合某一级别的关系模式的集合。
所谓"第几范式"，是表示关系模式的某一种级别。
R为第几范式可以写成$R \in \mathrm{xNF}$。

各范式之间的包含关系为
$5NF \subset 4NF \subset BCNF \subset 3NF \subset 2NF \subset 1NF$

并不总是需要达到最高范式

第一范式

不存在多值属性和内部结构

- 多值属性就是一个属性可以有多种取值，这时候一般新建一张表，引入外键（之前就学过）。内部结构就是比如first name和last name这种。

第1范式(FIRST NORMAL FORM, 1NF)

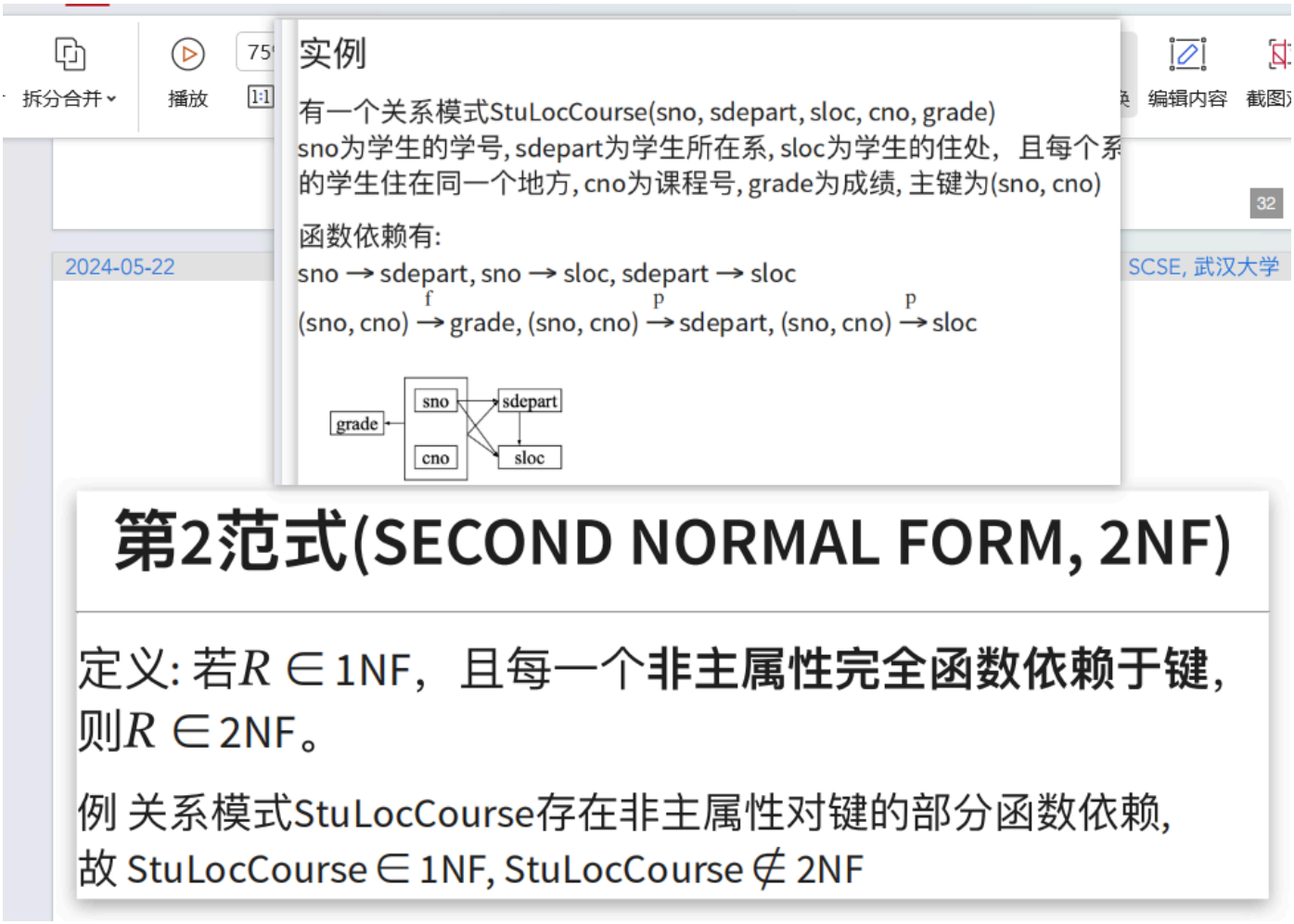
定义: 如果一个关系模式不含有多值属性(表里为重复字段)和内部结构(如记录类型)属性, 则称该模式为第1范式。

- 关系模式StuLocCourse $\in$ 1NF
- 关系模式StuLocCourse存在更新异常(插入异常和删除异常)

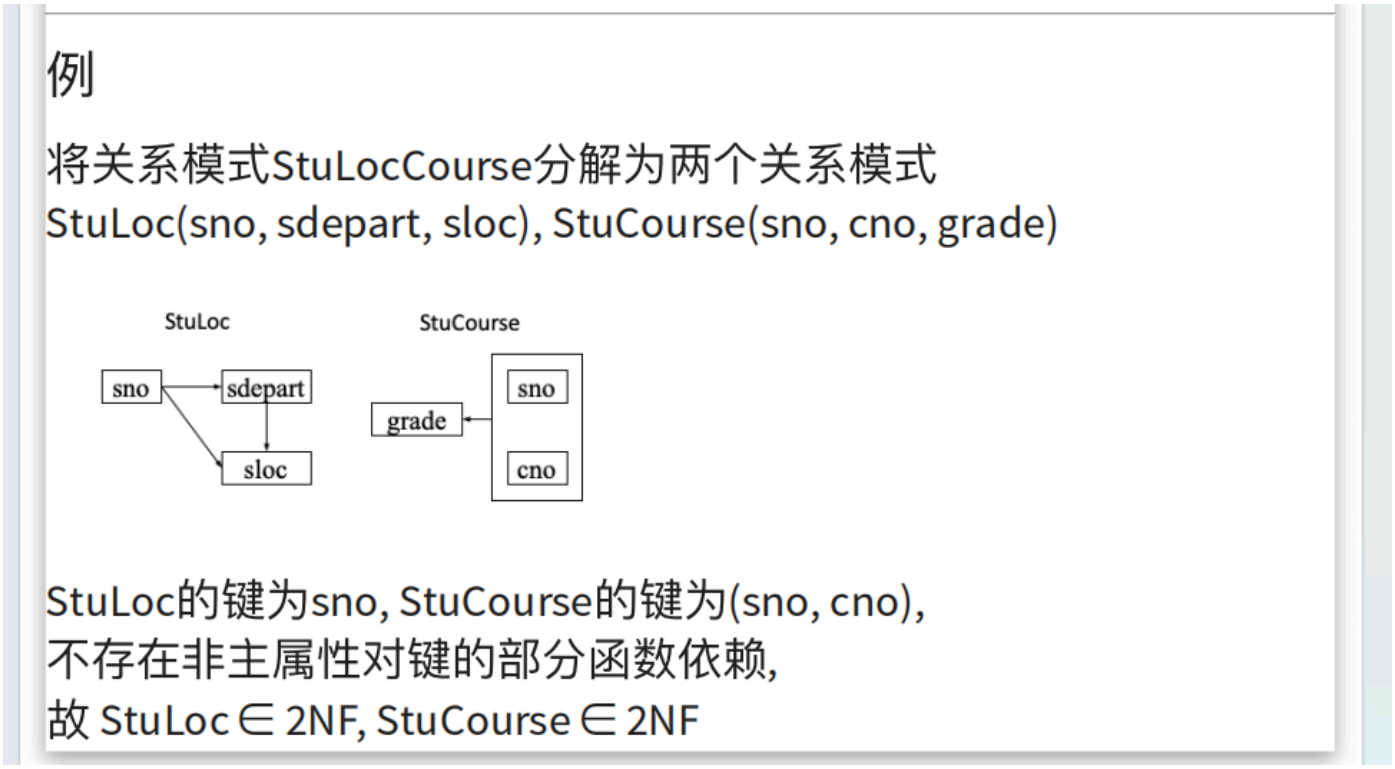
第二范式

每一个非主属性完全函数依赖于键

- 每一个非主属性完全函数依赖于键，此例子中有sno->sdepart,又有(sno,cno)->sdepart，sno是(sno,cno)的一个子集，显然这不满足完全函数依赖

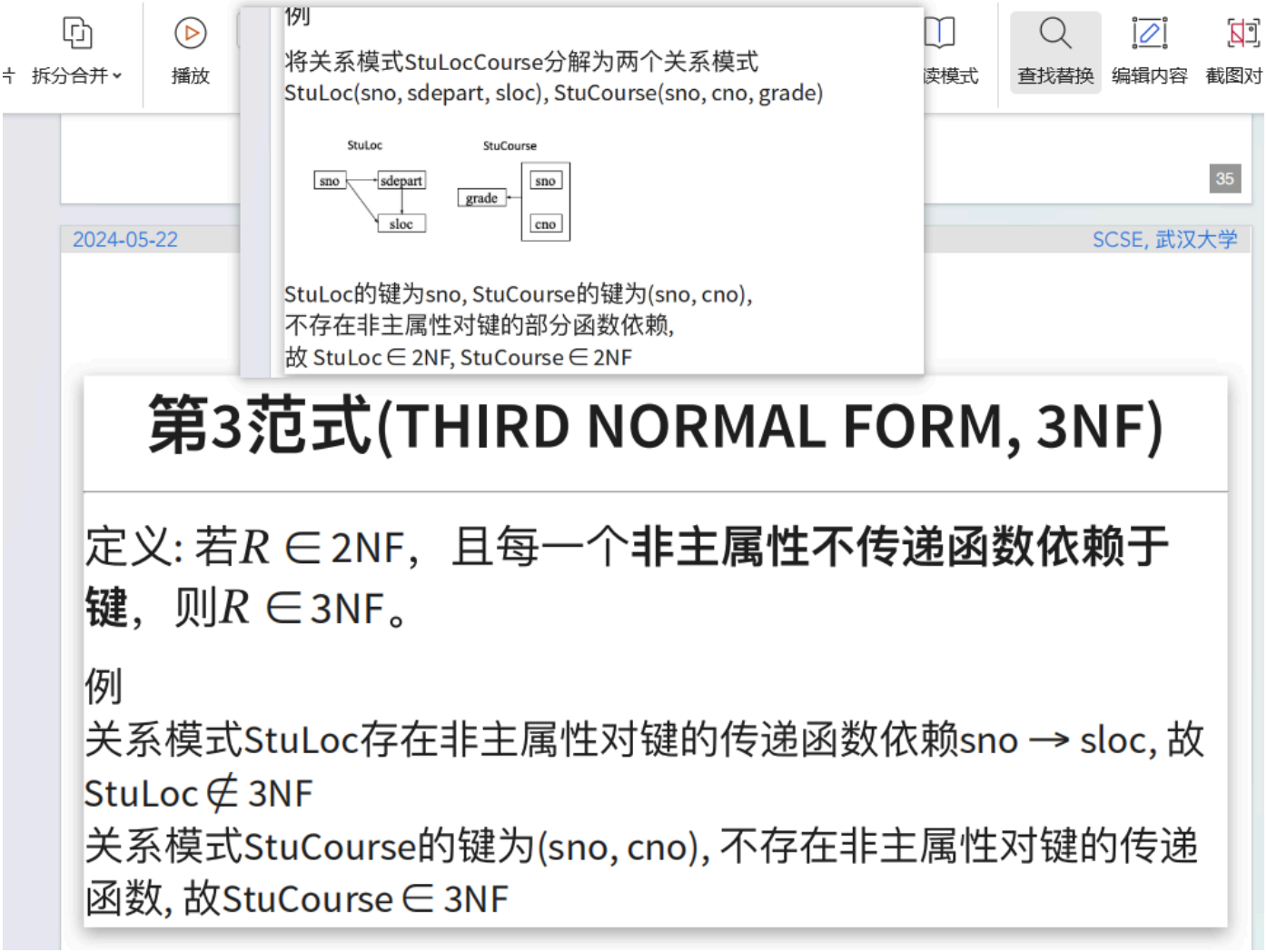


- 改为这样就满足第二范式了



第三范式

- 每一个非主属性不传递函数依赖于键，此例子中sno->sdepart,sdepart->sloc，又有sno->sloc，这就是个传递函数依赖



- 再拆分一次，消除传递函数依赖就行了

第3范式(THIRD NORMAL FORM, 3NF)

例

将关系模式StuLoc分解为两个关系模式
StuDept(sno, sdepart), DeptLoc(sdepart, sloc)

StuDept的键为sno, DeptLoc的键为sdepart,
不存在非主属性对键的传递函数依赖,
故 StuDept ∈ 3NF, DeptLoc ∈ 3NF

BC范式

- BC范式更像是一个补丁，弥补前三个范式的不足，要求每一个决定因素都包含键

BOYCE-CODD范式

Boyce-Codd范式(Boyce-Codd Normal Form, BCNF) 定义: 若 $R \in 1NF$，且**每一个决定因素都包含键**，则$R \in BCNF$。

例
关系模式 StuCourse(sno, cno, grade) 有函数依赖(sno, cno) → grade, 其决定因素包含键, StuCourse ∈ BCNF
关系模式 StuDept(sno, sdepart) 有函数依赖sno → sdepart, 其决定因素包含键, StuDept ∈ BCNF
关系模式 DeptLoc(sdepart, sloc) 有函数依赖sdepart → sloc, 其决定因素包含键, DeptLoc ∈ BCNF

- 并非所有属于3NF的关系模式都属于BCNF

并非所有属于3NF的关系模式都属于BCNF

例

有关系模式StuTeachCourse(sno, tno, cno), sno表示学生学号, tno表示教师编号, cno表示课程编号。 每一位教师只教一门课, 每门课有若干教师, 某一学生选定某门课， 就对应一个固定的教师。

由语义可以得到函数依赖:
tno → cno, (sno, cno) → tno, (sno, tno) → cno, (sno, tno) → (sno, tno, cno)

StuTeachCourse的键有(sno, cno)、(sno, tno)

BOYCE-CODD范式

并非所有属于3NF的关系模式都属于BCNF

所有属于BCNF的关系模式都属于3NF (可反证)

例

StuTeachCourse 没有非主属性, 不存在非主属性对键的部分与传递依赖， 故StuTeachCourse ∈ 3NF

StuTeachCourse有函数依赖tno → cno, 其决定因素tno不包含键, 故StuTeachCourse ∉ BCNF

数据依赖公理

- 逻辑蕴涵

逻辑蕴涵

定义 设F是关系模式R的函数依赖集,X、Y是R的属性子集,如果从F的函数依赖中能够推出$X \to Y$,则称F逻辑蕴涵$X \to Y$。

在进行函数依赖的检查时，基于已有函数依赖集判断另外一些函数依赖是否成立，通过函数依赖的逻辑蕴涵即可做到。

阿氏公理及其推论

ARMSTRONG公理(阿氏公理)

公理 对$R(U, F)$有

- 自反律: 若$Y \subseteq X$,则$X \to Y$。
- 增广律: 若$X \to Y$,则$XZ \to YZ$。
- 传递律: 若$X \to Y$、$Y \to Z$，则$X \to Z$。

理解

- 自反律：所有的平凡函数依赖都是成立的
- 增广律：函数依赖两边增加相同属性也成立
- 传递律：由已知函数依赖可以推导出新依赖

ARMSTRONG公理(阿氏公理)推论

推论

- 合并规则：若$X \to Y$、$X \to Z$，则$X \to YZ$。
- 分解规则：若$X \to YZ$，则$X \to Y$、$X \to Z$。
- 伪传递规则：若$X \to Y$、$WY \to Z$，则$WX \to Z$。

阿氏公理系统的完备性

建立阿氏公理系统的目的在于有效而准确地从已知的函数依赖推出未知的函数依赖, 公理系统满足两个特性:
(1) 正确性
按阿氏公理推出的依赖都是正确的
(2) 完备性
能推出所有的依赖
等价于所有不能用公理系统推出的依赖都不成立,
即不能由已知函数依赖集F逻辑蕴涵

函数依赖集闭包

- 大部分闭包的概念都大同小异，函数依赖集的闭包就是由已有的函数依赖可以通过一步或多步阿氏公理推导得到的所有函数依赖总的集合

函数依赖集闭包

定义 设有关系模式$R(U,F)$,X、Y为U的子集,函数依赖集F的闭包为:$F^+ = \{X \to Y | X \to Y$基于阿氏公理推出$\}$,即$F^+$为$F$所逻辑蕴含的函数依赖全体.

函数依赖集F的闭包F^+包括
① F中的函数依赖,由属性语义决定;
② 由F推出的平凡的函数依赖:
$A \to \phi$、$A \to A$、$AB \to A$、…这一类函数依赖与F无关,对R中任何属性都成立;
③ 由F推出的非平凡的函数依赖

- 但是函数依赖的闭包会非常多，很不好求

例

有关系模式$R(A,B,C)$,它的函依赖集$F = \{A \to B, B \to C\}$,计算F的闭包.

解:
(1) F中的函数依赖:$A \to B, B \to C$
(2) 由F推出的非平凡函数依赖:$A \to C, AC \to B \ldots$
(3) 由F推出的平凡函数依赖:$A \to \phi, A \to A \ldots$

F^+的计算很麻烦,可能会非常大

属性集闭包

- 因此引入属性集闭包，可以看出这个闭包是把范围所有缩小了的。F为U上某一个函数依赖集，X+是从X作为左部，通过F中的函数依赖和阿氏定理可以推导出的所有属性的集合

属性集闭包

定义 设有关系模式$R(U,F)$,F是属性集合U上的一个函数依赖集,$X \subseteq U$,属性集X关于F的闭包
$X_F^+ = \{A | X \to A$能由F用阿氏公理导出$\}$,可简写为X^+.

X_F^+是由X从F中推出的所有函数依赖右部的集合.

例

$R(A,B,C)$中,$F = \{A \to B, B \to C\}$,则$A_F^+ = \{A,B,C\}$,
$B_F^+ = \{B,C\}$,$C_F^+ = \{C\}$

函数依赖集闭包转变成属性集闭包计算

- 这一点很重要，正是因为求函数依赖集闭包不好求，因此把问题化简为求属性集闭包
- **判断X->Y不在F+中，只用判断Y是否属于X+**

> ## 函数依赖集闭包转变成属性集闭包计算
>
> 定理
> $X \to Y$能从F中用阿氏公理导出的充要条件是: $Y \subseteq X_F^+$.
>
> 判断$X \to Y$在不在F^+中, 只要判断Y是否属于X^+.
>
> 因此计算F^+的问题可以变换成计算X^+的问题,
> X^+的计算相对简单.

属性集闭包计算

- 特别留意核心操作第二步，注意要对使用过的Y->Z做已使用的标记

算法 有关系模式$R(U,F)$, F是属性集合U上的一个函数依赖集, $X \subseteq U$, 求属性集X关于F的闭包X_F^+ (X^+)

(1) $X^0 = X$
(2) $X^{i+1} = X^i \cup A$,
A由如下所有Z_j构成:
F中没有使用过的$Y_j \to Z_j$且$Y_j \subseteq X^i$;
求得X^{i+1} 后, 对$Y_j \to Z_j$做已使用的标记;
如果找不到这样的A, 则转(4)
(3) 若$X^{i+1} = X^i$或$X^{i+1} = U$则转(4), 否则转(2)
(4)结束, 输出X^{i+1}, 即为X^+

算法会终止吗? 最多$|U| - |X|$步

- 一些例子，这种其实给了题自然而然就会写，没必要硬看算法

例

有关系模式$R\langle U, F\rangle$, 其中 $U = \{A, B, C, D, E, I\}$,
$F = \{A \to D, AB \to E, BI \to E, CD \to I, E \to C\}$, 请计算$(AE)^+$.

解
(1) $X^0 = \{A, E\}$
(2) $X^1 = X^0 \cup \{D, C\} = \{A, C, D, E\}$, 用了$A \to D$和$E \to C$
(3) $X^2 = X^1 \cup \{I\} = \{A, C, D, E, I\}$, 用了$CD \to I$
(4) F中没有左部属于X^2的函数依赖, X^2为$(AE)^+$

有关系模式$R\langle U, F\rangle$, 其中
$U = \{A, B, C, D, E, G\}$,
$F = \{AB \to C, C \to A, BC \to D, ACD \to B, D \to EG, BE \to C,$
$CG \to BD, CE \to AG\}$,
请计算$(BD)^+$.

解
(1) $X^0 = \{B, D\}$
(2) $X^1 = \{B, D, E, G\}$, 用了$D \to EG$
(3) $X^2 = \{B, C, D, E, G\}$, 用了$BE \to C$
(4) $X^3 = \{A, B, C, D, E, G\}$, 用了$C \to A$
(5) $X^3 = U$, 所以$(BD)^+ = X^3 = \{A, B, C, D, E, G\}$

函数依赖集

函数依赖集等价

- 直接求出函数依赖集是不现实的，因此引出此判断方法

等价定义
关系模式R有两个函数依赖集F、G, 如果$F^+ = G^+$, 就说函数依赖集F覆盖G或F与G等价.

有关性质
(1) 若$G \subseteq F$, 则$G^+ \subseteq F^+$
(2) $(F^+)^+ = F^+$

判断方法

定理 关系模式R有两个函数依赖集$F = G$的充分必要条件是$F \subseteq G^+$和$G \subseteq F^+$.

- 对每个属于F形如$X \to Y$的属性依赖, 有$X \to Y \in G^+$, 即$Y \subseteq X_G^+$
- 对每个属于G形如$X \to Y$的属性依赖, 有$X \to Y \in F^+$, 即$Y \subseteq X_F^+$

- 例子

函数依赖集等价

例

关系模式R上有两个函数依赖集, $F = \{A \to B, B \to C\}$,
$G = \{A \to BC, B \to C\}$, 请判断$F$和$G$是否等价.

解

(1) 先检查F中的每一个函数依赖是否属于G^+

$\because A_G^+ = \{A, B, C\}, \therefore \{B\} \subseteq A_G^+, \therefore A \to B \in G^+$

$\because B_G^+ = \{B, C\}, \therefore \{C\} \subseteq B_G^+, \therefore B \to C \in G^+$

$\therefore F \subseteq G^+$

(2) 然后检查G中的每一个函数依赖是否属于F^+

$\because A_F^+ = \{A, B, C\}, \therefore \{B, C\} \subseteq A_F^+, \therefore A \to BC \in F^+$

$\because B_F^+ = \{B, C\}, \therefore \{C\} \subseteq B_F^+, \therefore B \to C \in F^+$

$\therefore G \subseteq F^+$

由(1)和(2)可得F和G等价.

最小函数依赖集

- 要紧扣概念最小，右部要是单属性，无多余函数依赖，左部无多余属性

最小函数依赖集

定义

关系模式R上有函数依赖集F, 若F满足下列条件,
(1) F中每个函数依赖的右部都是单属性;
(2) 对于F的任一函数依赖$X \to A, F - \{X \to A\}$与F都不等价,
即无多余函数依赖;
(3) 对于F中的任一函数依赖$X \to A$和X的真子集X',
$(F - (X \to A)) \cup \{X' \to A\}$与$F$都不等价,
即左部无多余属性
则称F为一个最小函数依赖集F_{m}.

最小
(1) F中每个函数依赖的右部没有多余的属性;
(2) F中不存在多余的函数依赖;
(3) F中每个函数依赖的左部没有多余的属性

- 例子

最小函数依赖集

例

如下函数依赖集, 哪个是最小依赖集?

$$F_1 = \{A \to D, BD \to C, C \to AD\}$$
$$F_2 = \{AB \to C, B \to A, B \to C\}$$
$$F_3 = \{BC \to D, D \to A, A \to D\}$$

F_1中第三个依赖的右部不是单属性
F_2中第一个依赖左部有多余属性A
F_3满足最小依赖集的三个条件

最小函数依赖集求法

- 第一步先让右边全部为单属性
- 第二步判断每一个函数依赖是否多余，也就是在F中减去这个X->A得到F',在F'限制中求X+，判断A是否属于X+，如果属于则X->A多余
- 第三步最小左边的多余属性，即有XY->A,在F中求X+，如果A属于X+，则Y多余

最小函数依赖集

定理 关系模式R上的函数依赖集F与一个最小函数依赖集F_{m}等价.

算法 计算最小函数依赖集F_{m}

(1) 分解: 使F中任一函数依赖的右部仅含有单属性.
(2) 删除冗余的函数依赖:
对F中任一$X \to A$, 在$F-\{X \to A\}$中求X^{+},
若$A \subseteq X^{+}$, 则$X \to A$为多余的.
(3) 最小化左边的多余属性:
对F中任一$XY \to A$, 在F中求X^{+},
若$A \subseteq X^{+}$, 则Y为多余的.

- 例题

例

关系模式R上有函数依赖集$F=\{B \to C, C \to AB, BC \to A\}$, 求与$F$等价的最小函数依赖集.

(1) 分解$C \to AB, F=\{B \to C, C \to A, C \to B, BC \to A\}$
(2) 判断$B \to C$是否冗余
$F'=\{C \to A, C \to B, BC \to A\}, B_{F'}^{+}=\{B\}, B \to C$非冗余
(3) 判断$C \to A$是否冗余
$F'=\{B \to C, C \to B, BC \to A\}, C_{F'}^{+}=\{A, B, C\}$,
$C \to A$冗余, $F=\{B \to C, C \to B, BC \to A\}$
(4) 判断$C \to B$是否冗余
$F'=\{B \to C, BC \to A\}, C_{F'}^{+}=\{C\}, C \to B$非冗余
(5) 判断$BC \to A$是否冗余
$F'=\{B \to C, C \to B\}, BC_{F'}^{+}=\{B, C\}, BC \to A$非冗余
(6) 判断$BC \to A$左边是否有多余属性
$B_{F}^{+}=\{A, B, C\}, \{A\} \subseteq B_{F}^{+}, C$在$BC \to A$中是多余的.
故$F_{\mathrm{m}}=\{B \to C, C \to B, B \to A\}$.

等价模式分解

等价模式分解

定义 关系模式$R\langle U, D, \mathrm{Dom}, F\rangle$ 的分解就是将其分解成一组等价的子关系子模式.

包括三个方面
(1) 属性的分解
(2) 关系的分解
(3) 函数依赖的分解

等价模式分解

(1) 属性等价
(2) 保持无损连接
(3) 保持函数依赖

- 属性等价性

  模式分解的属性等价性

  关系模式$R\langle U\rangle$被分解为关系模式$R_1\langle U_1\rangle$、$R_2\langle U_2\rangle$、…、$R_n\langle U_n\rangle$, 若$U = U_1 \cup U_2 \cup \ldots \cup U_n$, 则该分解是属性等价的分解.

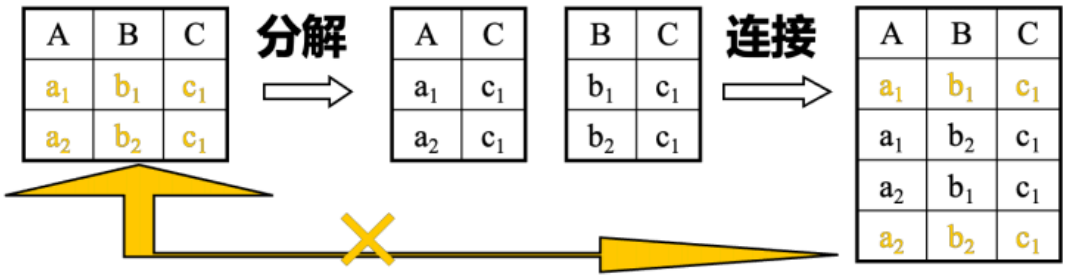
- 无损连接性

模式分解的无损连接性

关系模式$R\langle U\rangle$一个分解为
$\rho=\{R_1\langle U_1\rangle,R_2\langle U_2\rangle,\ldots,R_n\langle U_n\rangle\}$,若对$R$中任一个关系$r$,有$r=\Pi_{U_1}(r)\bowtie\Pi_{U_2}(r)\bowtie\ldots\bowtie\Pi_{U_n}(r)$,则该分解是保持无损连接的分解.

满足属性等价且有冗余属性的分解一定具有无损连接性吗?

等价模式分解



满足属性等价且有冗余属性的分解不一定具有无损连接性

- 函数依赖保持性，Z所涉及的F+中的所有的函数依赖为F在Z上的投影，注意要有X,Y都属于F+，那么X->Y才是一个投影

模式分解的函数依赖保持性

定义 关系模式$R\langle U,F\rangle$,F是R的函数依赖集,Z是R的一个属性子集,$Z\subseteq U$,称Z所涉及到的F^+中所有的函数依赖为F在Z上的投影,记为$\Pi_Z(F)$,有
$\Pi_Z(F)=\{X\to Y|X\to Y\in F^+且XY\subseteq Z\}$.

例 关系模式$R\langle U,F\rangle$,$F=\{A\to B,C\to B,B\to D,D\to C\}$,设$Z=CD$,则$\Pi_{CD}(F)=\{C\to D,D\to C\}$.

模式分解的函数依赖保持性

关系模式$R\langle U, F\rangle$被分解为关系模式$R_1\langle U_1\rangle$、$R_2\langle U_2\rangle$、…、$R_k\langle U_k\rangle$, 若F等价于$\left(\Pi_{U_1}(F) \cup \Pi_{U_2}(F) \cup \ldots \cup \Pi_{U_k}(F)\right)$, 则该分解是具有函数依赖保持性的分解.

保持函数依赖的分解就是指: 当一个关系模式R被分解后, 无语义丢失, 原来的函数依赖关系都分散在分解后的子模式中.

- 例题，说实话我没太懂这一块啥意思

例

有关系模式$R(A, B, C)$, 存在函数依赖集$F = \{A \rightarrow B, B \rightarrow C\}$, 下面的几个分解中, 哪一个最好?

$\rho_1 = \{R_1(A), R_2(B), R_3(C)\}$

$\rho_2 = \{R_4(A, B), R_5(A, C)\}$

$\rho_3 = \{R_4(A, B), R_6(B, C)\}$

$\rho_4 = \{R_5(A, C), R_6(B, C)\}$

例

分解	属性等价	保持无损连接	保持函数依赖
ρ_1	✔	✗	✗
ρ_2	✔	✔	✗
ρ_3	✔	✔	✔
ρ_4	✔	✗	✗

无损连接性检验

- 这个地方只要知道算法怎么用就行了，具体证明不做要求
- 重复考察X->Y中每一个函数依赖，并修改表中的元素:在X的分量中寻找相同的行，然后将这些行中Y的分量改为相同的符号。

无损连接性检验

(2) 重复考察F中每一个函数依赖$X \to Y$，并修改表中的元素:
在X的分量中寻找相同的行，然后将这些行中Y的分量改为相同的符号，如果其中有a_j，则将b_{ij}改为a_j；若其中无a_j，则全部改为b_{ij} (i是这些行的行号最小值)

- 如果发现表中某一行变成了$a_1, a_2, \ldots, a_n$，则分解ρ具有无损连接性.
- 如果F中所有函数依赖都不能再修改表中的内容，且没有发现这样的行，则分解ρ不具有无损连接性.

无损连接性检验 举例

关系模式$R\langle U, F\rangle$, $U = \{A, B, C, D, E\}$, $F = \{AB \to C, C \to D, D \to E\}$. R的一个分解为 $\rho = \{R_1(A, B, C), R_2(C, D), R_3(D, E)\}$. 请判断$\rho$是否具有无损连接性.

	A	B	C	D	E
ABC	a_1	a_2	a_3	b_{14}	b_{15}
CD	b_{21}	b_{22}	a_3	a_4	b_{25}
DE	b_{31}	b_{32}	b_{33}	a_4	a_5

AB→C

	A	B	C	D	E
ABC	a_1	a_2	a_3	b_{14}	b_{15}
CD	b_{21}	b_{22}	a_3	a_4	b_{25}
DE	b_{31}	b_{32}	b_{33}	a_4	a_5

C→D

	A	B	C	D	E
ABC	a_1	a_2	a_3	a_4	b_{15}
CD	b_{21}	b_{22}	a_3	a_4	b_{25}
DE	b_{31}	b_{32}	b_{33}	a_4	a_5

D→E

	A	B	C	D	E
ABC	a_1	a_2	a_3	a_4	a_5
CD	b_{21}	b_{22}	a_3	a_4	a_5
DE	b_{31}	b_{32}	b_{33}	a_4	a_5

- 特殊情况: 只有两个子模式的无损分解判定,有公式但是感觉不如上面的算法

分解为3NF子模式且具有无损连接性和函数依赖保持性的算法

- 这个地方感觉看看就行，不必深究

分解为3NF子模式且具有无损连接性和函数依赖保持性的算法

输入: 关系模式$R\langle U, F\rangle$
输出: 子关系模式集
(1) 最小化： 求F的最小函数依赖集F_{m}.
(2) 排除: 对F_{m}中任一$X \to A$, 若$XA = U$, R已为3NF, 则不分解, 结束.
(3) 独立: 若R中Z属性在F_{m}中未出现, 则所有Z为一个子模式的属性集, 令$U = U - Z$.
(4) 分组:
对F_{m}中$X \to A_1$、…、$X \to A_n$, 用合成规则合成一个函数依赖,
再对F_{m}中每个$X \to A$, 令$U_i = XA$.
R的分解$\rho = \{R_1(U_1), R_2(U_2), \dots, R_k(U_k)\}$.
(5) 添键: 如果分解中没有一个子模式含R的候选码X, 则将分解变成
$\rho = \{R_1(U_1), R_2(U_2), \dots, R_k(U_k), R_{k+1}(X)\}$; 如果存在$U_i \subseteq U_j$, 则删去$R_i$.

等价模式分解

例

有关系模式$R\langle U, F\rangle, U = \{E, G, H, I, J\}$,
$F = \{E \to I, J \to I, I \to G, GH \to I, IH \to E\}$, 将其分解为3NF子模式且同时具有无损连接性和函数依赖保持性.

求出F的最小函数依赖集为$F_{\mathrm{m}} = \{E \to I, J \to I, I \to G, GH \to I, IH \to E\}$.
得到分解为: $\{R_1(E, I), R_2(J, I), R_3(I, G), R_4(G, H, I), R_5(I, H, E)\}$
由候选码的定义和属性闭包的求解算法可以得到R的候选码中至少包含J和H, 且
$(J, H)^+ = \{I, J, H, G, E\} = U$, $\{J, H\}$是R的候选码.
上面的分解中没有子模式含有属性集$\{J, H\}$, 加上候选码并去重后得到最终的分解:
$\{R_1(J, I), R_2(G, H, I), R_3(I, H, E), R_4(J, H)\}$.