第八章 关系数据库设计

一、使用函数依赖进行分解

1.函数依赖形式化定义：

- 前情提要1：一个大写字母表示一个抽象的关系模式，eg.R
  - 用**大写字母或者希腊字母**来表示**属性集**，eg.α, R
  - 用希腊字母：是指一个有可能是模式也有可能不是模式的属性集
    - 用大写罗马字母：是指这个属性集一定为一个模式
  - 用形如：**小写字母(属性集名称)** 来表示**关系模式**，eg.r®
    - 注意：也可以将关系模式名称省略，即直接用大写字母或者希腊字母表示关系模式
  - 一个关系模式是一个属性集，一个属性集不一定是一个模式
  - 关系实例：一个关系在任意给定时间都有特定的值；我们将那看作一个实例并使用术语“r的实例”。当我们明显在讨论一个实例时，我们可以仅用关系的名字(例如r)
- 前情提要2：$\alpha \subseteq R$表示属性集R中包含属性α
- 前情提要3：$\alpha \rightarrow \beta$的意思是α决定β，这是函数依赖的表示方法

  意思是α和β分别是一个属性集，如果属性集α上的值确定了，
  那么属性集β上的值一定能确定
- 形式化定义：
  - $\alpha \subseteq R$,且 $\beta \subseteq R$
  - 如果我们想要$\alpha \rightarrow \beta$在R上成立的话，当且仅当

    在R上存在着一个合法的关系实例r,在r上存在着两个个元组t_1和t_2，
    如果这两个元组在α属性上的值相等，那么一定能保证这两个元组在β属性上的值相等
    即：$t_1[\alpha] = t_2[\alpha] \Rightarrow t_1[\beta] = t_2[\beta]$

一、使用函数依赖进行分解

2.函数依赖例子：

关系r(A,B)有以下实例：

实例1	1	4
实例2	1	5
实例3	3	7

因为实例1与实例2在A属性上的值相等，但是在B属性上的值不相等，

所以可以推出函数A不能决定函数B

因为在B属性上三个实例的值都不相等，所以可以推出函数B可以决定A，即$B \rightarrow A$

一、使用函数依赖进行分解

这个R是关系模式

这个R是属性集

3.码和函数依赖：

- K是关系模式R的一个**超码** 当且仅当 $K \rightarrow R$
  - 超码(super key)：是一个或多个属性的集合，这些属性的组合可以使我们在一个关系中唯一地标识一个元组
  - 解释：因为超码可以唯一确定一个元组，那么在超码确定的属性集合里，绝对不会有两个元组的值相等，因此函数K可以决定R
- K是关系模式R的一个**候选码** 当且仅当

  $K \rightarrow R$ 且 不存在任何一个K的真子集α,能使$\alpha \rightarrow R$ 成立
  - 候选码(candidate key)：最小超码

一、使用函数依赖进行分解

4.平凡的函数依赖

- 定义：α和β都是属性的集合

  如果$\beta \subseteq \alpha$,则一定有$\alpha \rightarrow \beta$

  - 即如果函数依赖的决定方α的属性包含被决定方β的所有属性，那么恒有$\alpha \rightarrow \beta$成立

- 举例：
  - $ID, name \rightarrow ID$
  - $name \rightarrow name$

一、使用函数依赖进行分解

5.闭包

- 前情提要：函数依赖具有传递性，
  如果$A \to B, B \to C$,我们就能推出$\mathrm{A} \to C$
- 闭包定义：给定关系r(R)上成立的函数依赖集F，F集合的闭包就是能从给定F集合推导出的所有函数依赖的集合，用F^+表示。
- 显然F^+包含F中所有的函数依赖

一、使用函数依赖进行分解

6.Boyce-Codd范式(BCNF,Boyce-Codd Normal Form)（范式分解基本不考）

- 作用：从理论上证明自己关系模式是否正确的标准
- 判别：具有函数依赖集F的关系模式R属于$BCNF$的条件是：

  对$\boldsymbol{F}^+$中所有形如$\alpha \rightarrow \beta$的函数依赖(其中$\alpha \subseteq R$且$\beta \subseteq R$),下面至少有一项成立:
  - $\alpha \rightarrow \beta$是平凡的函数依赖，即$\beta \subseteq \alpha$
  - α是模式R的一个超码
- 一个数据库设计属于BCNF的条件是,构成该设计的关系模式集中的每个模式都属于BCNF
- 分解：

  我们把原来的关系模式R分解为以下两个部分
  - $(\alpha \cup \beta)$
  - $(R-(\beta-\alpha))$

一、使用函数依赖进行分解

7.第三范式(3NF,Third Normal Form)

- 判别：具有函数依赖集F的关系模式R属于$3NF$的条件是：

  对F^+中所有形如$\alpha \rightarrow \beta$的函数依赖(其中$\alpha \subseteq R$且$\beta \subseteq R$),下面至少有一项成立:
  - $\alpha \rightarrow \beta$是平凡的函数依赖，即$\beta \subseteq \alpha$
  - α是模式R的一个超码
  - $\beta - \alpha$中的每个属性A都包含于R的一个候选码中
    - 注意：并没有说单个候选码必须包含$\beta - \alpha$的所有属性；$\beta - \alpha$中的每个属性A可能包含于不同的候选码中
- 分解的方法在之后会介绍

二、函数依赖理论

1.Armstrong公理

- 规定：
  - 用希腊字母($\alpha, \beta, \gamma \ldots$)表示属性集
  - 用大写罗马字母($A, B, C \ldots$)表示单个属性
  - 我们用$\alpha\beta$表示$\alpha \cup \beta$
- 作用：利用Armstrong公理寻找逻辑蕴涵的函数依赖，求出函数依赖集合闭包。
- 公理内容：
  - 自反律：若α为一属性集且$\beta \subseteq \alpha$,则$\alpha \rightarrow \beta$成立
  - 增补律(增广律)：若$\alpha \rightarrow \beta$成立且γ为一属性集，则$\gamma\alpha \rightarrow \gamma\beta$成立
  - 传递律：若$\alpha \rightarrow \beta$和$\beta \rightarrow \gamma$成立，则$\alpha \rightarrow \gamma$成立
- 拓展规则：(进一步简化计算)【考证明】
  - 合并律：若$\alpha \rightarrow \beta$和$\alpha \rightarrow \gamma$成立，则$\alpha \rightarrow \beta\gamma$成立
  - 分解律：若$\alpha \rightarrow \beta\gamma$成立，则$\alpha \rightarrow \beta$和$\alpha \rightarrow \gamma$成立
  - 伪传递律：若$\alpha \rightarrow \beta$和$\gamma\beta \rightarrow \delta$成立，则$\alpha\gamma \rightarrow \delta$成立

二、函数依赖理论

1.Armstrong公理

- **函数依赖集**闭包:

  例子: $R(X,Y), F=\{X\to Y\}$

  $F^+=\{X\to\varphi, X\to X, X\to Y, X\to XY,$

  自反律 自反律 已知 增广律

  $Y\to\varphi, \mathrm{Y}\to\mathrm{Y},$

  自反律 自反律

  $XY\to\varphi, XY\to X, XY\to Y, XY\to XY\}$

  自反律 自反律 自反律 自反律

二、函数依赖理论

1.Armstrong公理

- 求函数依赖集合中相关的属性集的闭包（因为求函数依赖集合的闭包是一个NP问题，我们退而求其次）
  - 前提：给定一个属性集α，再给定一个函数依赖的集合F
  - 我们可以求属性集α在指定函数依赖集合F中的属性集的闭包α^+
  - 算法：

$result := \alpha$

while($result$集合发生了变化)do

  for each 函数依赖$\beta \rightarrow \gamma$ in F do

    begin

      if $\beta \subseteq result$ then $result \coloneqq result \cup \gamma$;

    end

二、函数依赖理论

1.Armstrong公理

- 求函数依赖集合中相关的**属性集的闭包**（因为求函数依赖集合的闭包是一个NP问题，我们退而求其次）
  - 举例：
    $R = (A, B, C, G, H, I), F = \{A \rightarrow B, A \rightarrow C, CG \rightarrow H, CG \rightarrow I, B \rightarrow H\}$
    求$(AG)^+$，并且根据结果判断AG是不是一个super key，是不是一个candidate key
    解：
    1.$result = AG$
    2.$result = ABCG(A \rightarrow \mathrm{C\ and\ A} \rightarrow B\ and\ A \subseteq AG)$
    3.$result = ABCGH(CG \rightarrow \mathrm{H}\ and\ CG \subseteq ABCG)$
    4.$result = ABCGHI(CG \rightarrow I\ and\ CG \subseteq ABCGH)$
    5.虽然$B \rightarrow H$,但是H已经在result中，所以不做操作
    6.再次循环一遍，发现result没有改变，算法结束
  - 判断AG是不是R的一个super key，判断的是AG →R是否成立。因为$(AG)^+ \supseteq R$，所以AG →R成立，即AG是R的一个super key
  - 如果AG是R的一个candidate key，除了要证明AG →R成立，还要证明AG的任何一个子集都不是R的super key,最终结果是AG是R的一个candidate key

二、函数依赖理论

1.Armstrong公理

- 求函数依赖集合中相关的属性集的闭包（因为求函数依赖集合的闭包是一个NP问题，我们退而求其次）
  - 求属性集闭包的作用：
    - 判断某个属性集是否是关系模式的super key或者candidate key
    - 检查函数依赖(检查$\alpha \rightarrow \beta$是否成立，只用求α^+，如果$\alpha^+ \supseteq \beta$，则$\alpha \rightarrow \beta$成立)
    - 求函数依赖集合的闭包(对于每个属性集$\gamma \subseteq R$，我们都可以求γ^+，对于每个$S \subseteq \gamma^+$，我们都能得出$\gamma \rightarrow S$)

二、函数依赖理论

2.候选键(candidate key)计算(课程补充)

- 定义：
  - 左部属性：只出现在F左边的属性
  - 右部属性：只出现在F右边的属性
  - 双部属性：出现在F两边的属性
  - 外部属性：不出现在F中的属性
- 定理：
  - 左部属性和外部属性一定出现在任何候选码中
  - 右部属性一定不出现在任何候选码中

二、函数依赖理论

2.候选键(candidate key)计算(课程补充)

- 举例1：

  $\mathrm{R}=(C,T,H,I,S)\ ,F=\{C\rightarrow T,HI\rightarrow C,HT\rightarrow I,HS\rightarrow I\}$

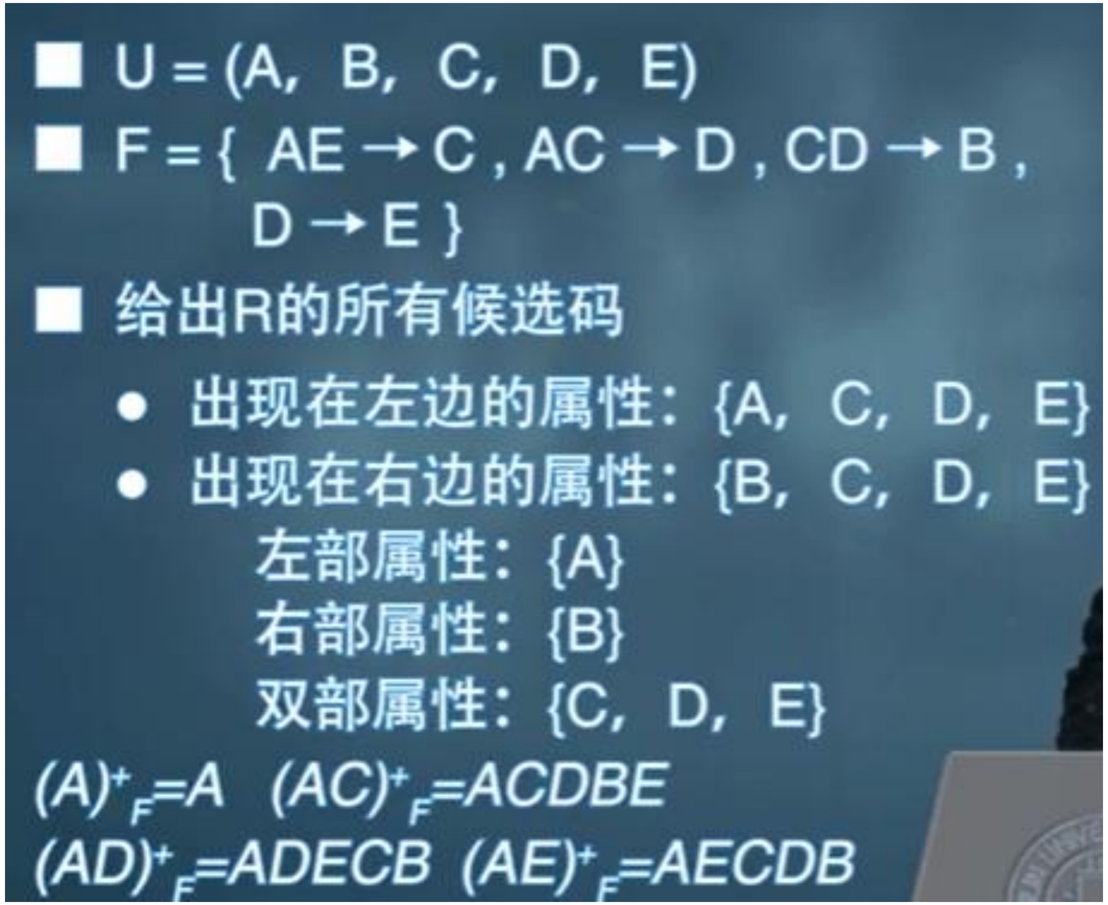
  求R的所有候选码

  - 出现在左边的属性：{C,H,I,T,S}
  - 出现在右边的属性：{C,T,I}
  - 因此可得：
    - 左部属性：{H,S}
    - 右部属性：{}
    - 双部属性：{C,T,I}
  - $(HS)^{+}=HSICT$,因为$(HS)^{+}\supseteq R$，所以$HS\rightarrow R$, HS是R的一个候选键

二、函数依赖理论

2.候选键(candidate key)计算(课程补充)

- 举例2：

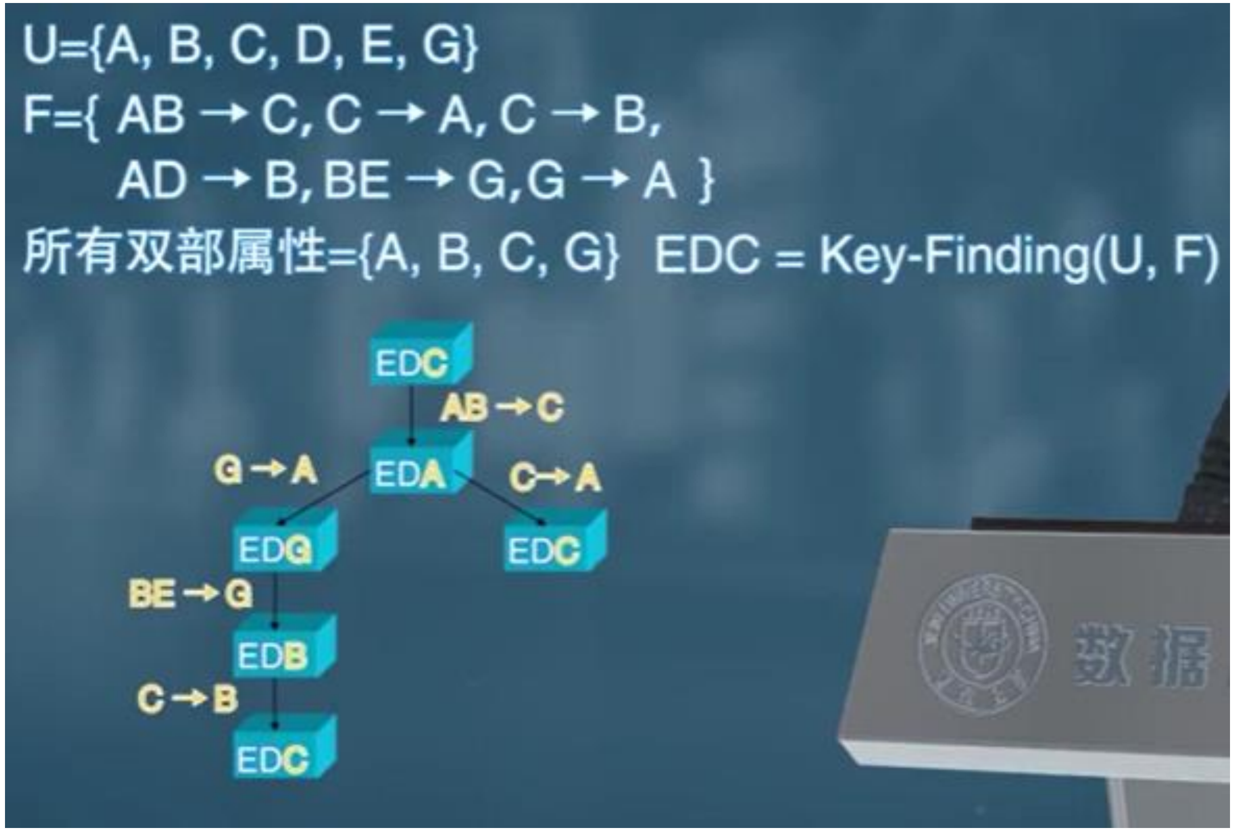


双部属性不为空，则候选键可能是左部属性，以及左部属性与双部属性的组合

二、函数依赖理论

2.候选键(candidate key)计算(课程补充)

- 求全部候选码：替换法



左部属性={E,D}
双部属性={A,B,C,G}
首先通过组合求出一个候选键->EDC
然后以EDC为起点推导
推导过程入左下图

二、函数依赖理论

3.正则覆盖

- 为什么提出这个概念?
  - 冗余的依赖，冗余的属性(即无关属性)

For example: $A \rightarrow C$
is redundant in: $\{A \rightarrow B,\ \ B \rightarrow C\}$

Parts of a functional depe-
ndency may be redundant

右边有无关属性

- E.g.: on RHS:
  $\{A \rightarrow B,\ \ B \rightarrow C,\ \ A \rightarrow CD\}$
  can be simplified to
  $\{A \rightarrow B,\ \ B \rightarrow C,\ \ A \rightarrow D\}$

$A \rightarrow CD$可以拆分成$A \rightarrow C$， $A \rightarrow D$
而$A \rightarrow C$是冗余的，上述的属性可以简化为图中所示

二、函数依赖理论

3.正则覆盖

- 形式化概念：
  - F的正则覆盖F_c应该满足如下条件：
    - F与F_c互相蕴含对方中的所有依赖
    - F_c的任何函数依赖都不包含无关属性
    - F_c中函数依赖的左半部都是唯一的，即不会有两个左半部相同的函数依赖
- 注意：
  - 当检验一个属性是否无关时，检验时用的是F_c当前值中的函数依赖，而不是F中的依赖。（不理解）
  - 如果去除完无关属性之后某个函数依赖的右部为空，应该把这个函数依赖去除

二、函数依赖理论

3.正则覆盖

- **逻辑蕴含**概念：
  - F能推出 原不直观存在于 函数依赖集F 中的函数依赖$\alpha \rightarrow \beta$，则称函数依赖集F逻辑蕴含$\alpha \rightarrow \beta$
- 对于**无关属性**这一概念的形式化定义：
  - 对于决定方的无关属性：
    - $A \in \alpha$并且F逻辑蕴含 $F', F' = (F - \{\alpha \rightarrow \beta\}) \cup \{(\alpha - \mathrm{A}) \rightarrow \beta\}$
    - 解释：如果A是无关属性，那么用函数依赖$(\alpha - A) \rightarrow \beta$来替换原来的$\alpha \rightarrow \beta$，形成的新的函数依赖集$F'$，可以被$F$逻辑蕴含
  - 对于被决定方的无关属性：
    - $A \in \beta$并且$(F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - A)\}$逻辑蕴含$F$
    - 解释：如果A是无关属性，那么用函数依赖$\alpha \rightarrow (\beta - A)$来替换原来的$\alpha \rightarrow \beta$，形成的形的函数依赖集$F'$，可以逻辑蕴含$F$

二、函数依赖理论

3.正则覆盖

- 对于**无关属性**这一概念的形式化定义：
  - 判断某属性是不是无关属性的例子：

题目：判断函数依赖$AB \rightarrow CD$中C是不是无关属性

其中$F = \{AB \rightarrow CD, A \rightarrow E, E \rightarrow C\}$

解答：因为在函数依赖$AB \rightarrow CD$中C在右侧，在被决定侧

所以计算F'使用$F' = \{F - AB \rightarrow CD\} \cup (AB \rightarrow D)$

$= \{AB \rightarrow D, A \rightarrow E, E \rightarrow C\}$

现在就看F'能不能逻辑蕴含F,

在F'中,因为$A \rightarrow E, E \rightarrow C$,所以$A \rightarrow C$;

又因为$AB \rightarrow D$,所以$AB \rightarrow CD$

因此F'能逻辑蕴含F，所以函数依赖$AB \rightarrow CD$中C是无关属性

二、函数依赖理论

3.正则覆盖

- 正则覆盖算法
  - 例题：

    题目：$R=(A,B,C), F=\{A\rightarrow BC, B\rightarrow C, A\rightarrow B, AB\rightarrow C\}$,求$F$的正则覆盖

    解答：1.首先看是否有相同的左半部->有， $A\rightarrow BC$和$A\rightarrow B$，合并为$A\rightarrow BC$

    此时$F'=\{A\rightarrow BC, B\rightarrow C, AB\rightarrow C\}$

    2.判断是否有无关属性，先左部，再右部

    1）左部：$AB\rightarrow C$中的A是否是无关属性?->$\{A\rightarrow BC, B\rightarrow C\}$是否能推出$AB\rightarrow C$

    -> $\mathrm{A}\rightarrow C, B\rightarrow C$则$AB\rightarrow C$

    所以此时$F'=\{A\rightarrow BC, B\rightarrow C\}$

    2）右部：$A\rightarrow BC$中B是否是无关属性？ -> $\{A\rightarrow C, B\rightarrow C\}$是否能推出$A\rightarrow BC$

    ->不行

    $A\rightarrow BC$中C是否是无关属性？ ->$\{A\rightarrow B, B\rightarrow C\}$是否能推出$A\rightarrow BC$

    -> $B\rightarrow \mathrm{C}$可得$B\rightarrow BC$,因此$A\rightarrow BC$

    所以此时$F'=\{A\rightarrow B, B\rightarrow C\}$，此时的$F'$即为$F_c$
  - 注意：判断顺序不同可能导致结果不同

二、函数依赖理论

4.无损连接分解

- 概念：把属性集R分解成R_1和R_2两个集合，如果用关系模式r(R_1)和r(R_2)替代r(R)时没有信息损失，则称该分解为无损连接分解。
- 形式化表示：

$$r = \Pi_{R1}(r) \bowtie \Pi_{R2}(r)$$

- 判断条件：R_1和R_2是R的无损连接分解，如果以下函数依赖中至少有一个属于F^+：
  - $R_1 \cap R_2 \rightarrow R_1$
  - $R_1 \cap R_2 \rightarrow R_2$

  换句话说，如果$R_1 \cap R_2$是R_1或R_2的超码，R上的分解就是无损分解

  （PS：K是关系模式R的一个超码 当且仅当 $K \rightarrow R$）

二、函数依赖理论

4.无损连接分解

- 举例：问题：$R=(A,B,C)$　$F=(A\rightarrow B,B\rightarrow C)$

  把R分解成$R_1=(A,B)$和$R_2=(B,C)$是否是无损连接分解?

  解答：因为$R_1\cap R_2=(B)$,因为函数依赖集F中有$B\rightarrow C$，可以推出$B\rightarrow BC$，所以$R_1\cap R_2\rightarrow R_2$,因此这个分解是无损连接分解

二、函数依赖理论

5.如何在分解的时候保持依赖

- 由定义提出的方法（难实现，一般不用）：

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

- 另一个比较简单的方法：
  - 该算法对F中的每一个$\alpha \rightarrow \beta$使用下面的过程：

$$
\begin{aligned}
& result = \alpha \\
& \textbf{repeat} \\
& \qquad \textbf{for each} \quad \text{分解后的} R_i \\
& \qquad\qquad t = (result \cap R_i)^+ \cap R_i \\
& \qquad\qquad result = result \cup t \\
& \qquad \textbf{until} \quad (result \text{ 没有变化})
\end{aligned}
$$

等价于求解在F_i下的$result$闭包

如果$result$包含β中的所有属性，则函数依赖$\alpha \rightarrow \beta$保持。

分解是保持依赖的当且仅当上述过程中F的所有依赖都保持。

二、函数依赖理论

6.BCNF分解（BCNF判断直接求候选码，但凡左部有一个不包含候选码，那它就不是BCNF）

- 例子：$R=(A,B,C)$ $F=\{A\rightarrow B, B\rightarrow C\}$判断R是否属于BCNF，如果不是，分解直到它属于BCNF，并判断分解的属性集是否是无损连接分解，是否在分解中保持依赖
  - BCNF判别：具有函数依赖集F的关系模式R属于$BCNF$的条件是：
    对F^+中所有形如$\alpha\rightarrow\beta$的函数依赖(其中$\alpha\subseteq R$且$\beta\subseteq R$),下面至少有一项成立：
    - $\alpha\rightarrow\beta$是平凡的函数依赖，即$\beta\subseteq\alpha$
    - α是模式R的一个超码

    注：在第一次判断（即分解之前）的时候可以只判断F中所有的函数依赖就可以。
  - 解答：
    对于$A\rightarrow B$，A是R的一个超码
    对于$B\rightarrow C$，两个条件都不满足，R不属于BCNF

进行分解：我们把原来的关系模式R分解为以下两个部分

$(\alpha\cup\beta)$，即$R_1=(B,C)$

$(R-(\beta-\alpha))$，即$R_2=(A,B)$

再进行判断：

$$F^+=\{A\rightarrow\varphi, A\rightarrow A, A\rightarrow B, A\rightarrow AB,$$
$$B\rightarrow\varphi, B\rightarrow B, B\rightarrow C, B\rightarrow BC,$$
$$C\rightarrow\varphi, C\rightarrow C,$$
$$AB\rightarrow\varphi, AB\rightarrow A, AB\rightarrow B, AB\rightarrow AB,$$
$$BC\rightarrow\varphi, BC\rightarrow B, BC\rightarrow C, BC\rightarrow BC\}$$

符合条件1的函数依赖如下：

$$A\rightarrow\varphi, A\rightarrow A, B\rightarrow\varphi, B\rightarrow B, C\rightarrow\varphi, C\rightarrow C,$$
$$AB\rightarrow\varphi, AB\rightarrow A, AB\rightarrow B, AB\rightarrow AB, BC\rightarrow\varphi,$$
$$BC\rightarrow B, BC\rightarrow C, BC\rightarrow BC$$

我们还需要判断：$A\rightarrow B, A\rightarrow AB, B\rightarrow C, B\rightarrow BC,$

对于$A\rightarrow B$，A是R_2的超码

对于$A\rightarrow AB$，A是R_2的超码

对于$B\rightarrow C$，B是R_1的超码

对于$B\rightarrow BC$，B是R_1的超码

二、函数依赖理论

6.BCNF分解

- 例子：$R=(A,B,C)\quad F=\{A\rightarrow B,B\rightarrow C\}$判断R是否属于BCNF，如果不是，分解直到它属于BCNF，并判断分解的属性集是否是无损连接分解，是否在分解中保持依赖

  $R_1=(B,C)$，$R_2=(A,B)$

  - 接下来判断由R到$R_1\ R_2$的分解是否是无损连接分解

    因为$R_1\cap R_2=B$，B是R_1的一个超码，所以是无损连接分解。

  - 分解是否保持依赖？
    - 对于$A\rightarrow B$,

      初始result=A, result$\cap R_1=\varphi$, 所以$(\text{result}\cap R_1)^{+}=\varphi$, $(\text{result}\cap R_1)^{+}\cap R_1=\varphi, result=A$

      result $\cap R_2=A$,所以$(\text{result}\cap R_2)^{+}=ABC$, $(\text{result}\cap R_2)^{+}\cap R_2=AB$, $\text{result}=\text{AB}$

      下一轮result没有变化，退出循环，因为result包含B，所以$A\rightarrow B$保持依赖
    - 对于$B\rightarrow C$，

      初始result=B, result$\cap R_1$=B,所以$(\text{result}\cap R_1)^{+}=BC$, $(\text{result}\cap R_1)^{+}\cap R_1=BC, result=BC$

      result $\cap R_2$=B,所以$(\text{result}\cap R_2)^{+}=\text{BC}$, $(\text{result}\cap R_2)^{+}\cap R_2=B, result=BC$

      下一轮result没有变化，退出循环，因为result包含C，所以$B\rightarrow C$保持依赖

  因此分解保持依赖

二、函数依赖理论

7.3NF分解

- 判别：具有函数依赖集F的关系模式R属于$3NF$的条件是：

  对F^+中所有形如$\alpha \rightarrow \beta$的函数依赖(其中$\alpha \subseteq R$且$\beta \subseteq R$),下面至少有一项成立：
  - $\alpha \rightarrow \beta$是平凡的函数依赖，即$\beta \subseteq \alpha$
  - α是模式R的一个超码
  - $\beta - \alpha$中的每个属性A都包含于R的一个候选码中
    - 注意：并没有说单个候选码必须包含$\beta - \alpha$的所有属性；$\beta - \alpha$中的每个属性A可能包含于不同的候选码中
- 举例：

  $R = (J, K, L), F = \{JK \rightarrow L, L \rightarrow K\}$

  因为L无法推出R，所以L不是模式R的一个超码，关系模式R不属于BCNF

  接下来判断R是否属于3NF，首先求R的候选码【候选码：A可以推出R且A的子集不能推出R】

  R在左边的属性有：J,K,L　　R在右边的属性有 L,K

  所以R的左部属性有J, R的双部属性有K,L

  可能为R的候选码的组合有：J,JK,JL,JKL

  最后可以得出：R的候选码有：JK,JL

  因为L包含在R的候选码JL中，K包含在R的候选码JK中，所以R属于3NF

二、函数依赖理论

7.3NF分解

- 分解：(分解之后所有新模式都属于3NF且都是无损链接、保持依赖的)
  1. 求F的正则覆盖$\mathrm{F_c}$
  2. i=0,对于F_c中的每一个函数依赖$\alpha \to \beta$，先$i++$,再将$\alpha\beta$作为一个新的模式R_i
  3. 求R的候选码
  4. 如果第二步算出来的所有模式$R_j(j = 1,2 \dots, i)$都不包含R的候选码，那么将R的任意一个候选码也作为一个新模式R_i
  5. 可选步骤：遍历求出来的所有模式，如果一个模式R_j是另一个模式的子集，那么便把R_j去掉

  最终把原模式R分解成新模式组$(R_1, R_2, \dots, R_i)$