关系数据理论

数据依赖

函数依赖

> 由s[X]=t[X]能够导致s[Y]=t[Y]，称X函数决定Y，或者Y函数依赖X，记为X->Y

- **平凡与非平凡函数依赖**

  > X→Y，但Y ⊊ X则称X→Y是非平凡的函数依赖。
  >
  > X→Y，但Y⊆ X 则称X→Y是平凡的函数依赖

- **完全函数依赖与部分函数依赖**

  > 完全函数依赖: 是指 X→Y，且对任何X的真子集X'，都有X ' -/-> Y，记作：X -f-> Y。
  >
  > 部分函数依赖: 是指X→Y，且存在X的真子集X '，有X ' →Y， 记作： X-p->Y 。
  >
  > 左部为单属性的函数依赖一定是完全函数依赖。

  - (SNO，CNO) →GRADE 是一个完全函数依赖 SNO-/->GRADE CNO-/->GRADE
  - (SNO，CNO) →ADDRESS 都是部分函数依赖 SNO->ADDRESS CNO->ADDRESS
- **传递函数依赖**

  > 是指若X→Y (Y不包含于 X)，Y -/-> X ， 而Y →Z记作： X -t->Z

多值依赖

> X、Y、Z是U的子集且Z=U-X-Y，当且仅当对于R(U)的任一关系r，
>
> 如果存在元组(x,y1,z1)和(x,y2,z2)时，一定存在(x,y1,z2)和(x,y2,z1)，即交换两元组在Y上的值产生的新元组依然在关系中。则称Y多值依赖于X

关系范式

1. **第一范式 1NF**

   > R的每个属性值都是不可再分的最小数据单位

   - 转化方法
     - 组合属性：拆分 合并
     - 重复组：拆分多个元组
   - 依旧存在问题
     - 数据冗余
     - 插入异常
     - 删除异常
2. **第二范式 2NF**

   > 若R∈1NF，且R中的每一个**非主属性**都**完全函数依赖**于R的**任一候选码**，则R∈2NF

- 转化方法
  - 破坏**部分依赖**的条件 分解为2NF
- 结论
  - 1NF是候选码是单属性 则R一定是2NF
- 依旧存在问题 【原因： 存在传递依赖】
  - 数据冗余
  - 修改复杂
  - 插入异常
  - 删除异常

3. **第三范式 3NF**

> 如果关系R的**任意一个非主属性**都不**传递函数依赖**于它的**任意一个候选码**，则R∈3NF

- 转化方法
  - 破坏传递**函数依赖** 将两个依赖中的属性分解到不同关系中
- 结论
  - 关系R中的属性都是主属性 则R一定是3NF
- 依旧存在的问题 【原因：可能存在主属性对候选码的部分依赖或者传递依赖】
  - 插入异常
  - 删除异常

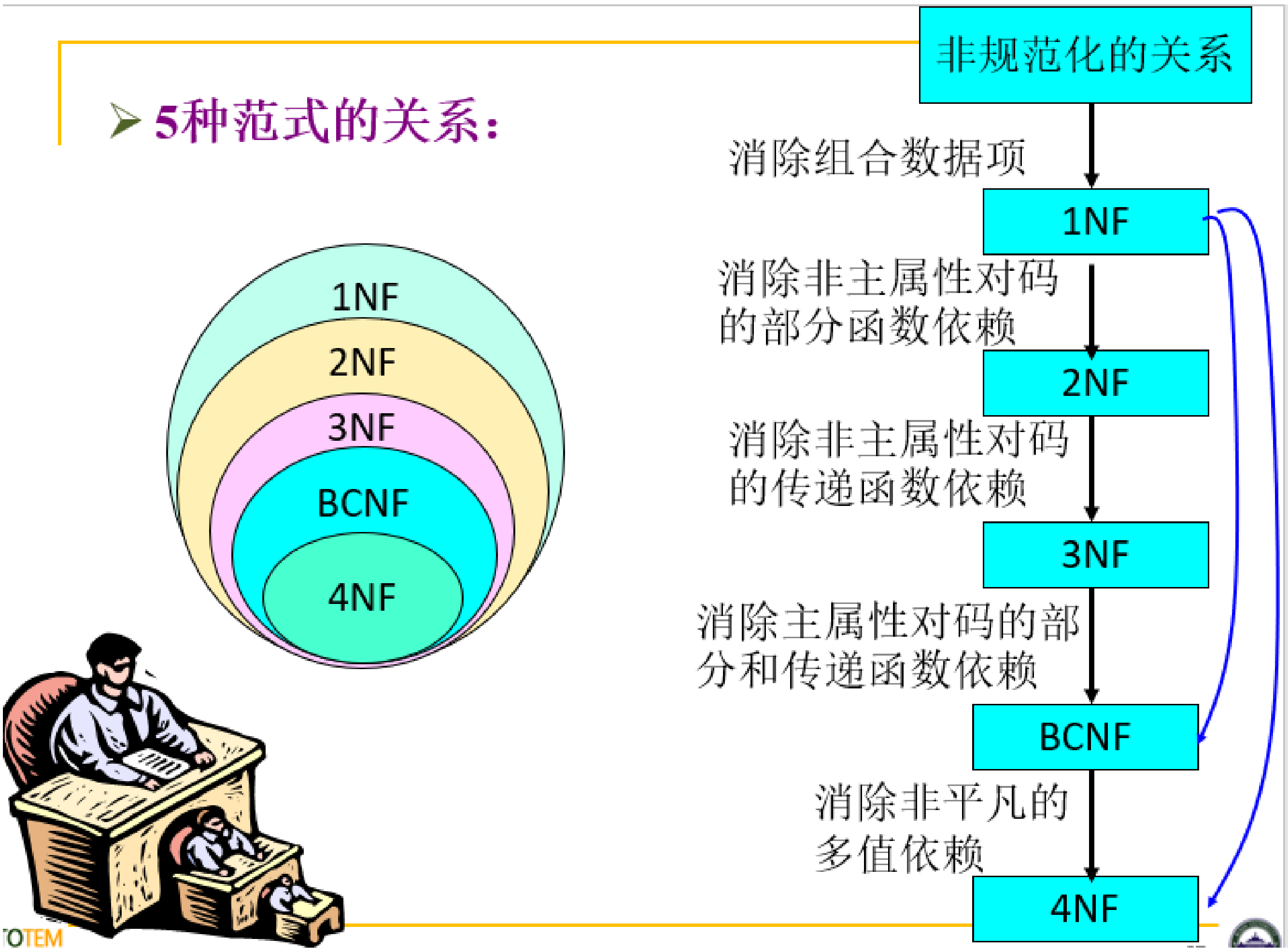
4. **BCNF**

> 关系模式 R<U，F>∈1NF。若函数依赖集合F中的所有函数依赖X→Y（Y不包含于X）的左部都包含R的任一侯选码，则R∈BCNF。
>
> **换言之，BCNF中的所有依赖的左部都必须包含候选码**

- 转化方法
  - BCNF分解
- 结论
  - 任何二元关系必定是BCNF
- 任然存在问题 【原因：可能存在主属性对候选码的部分函数依赖和传递函数依赖】
  - 插入异常
  - 删除异常

5. **第四范式 4NF**

> 若R∈1NF，D是R上的多值依赖集合，如果对于任何一个非平凡多值依赖X →→ Y，X都包含了R的一个候选码，则R∈4NF。



公理系统

阿氏公理

- 逻辑蕴含

  > F:关系模式R的函数依赖集
  >
  > X Y: R的属性子集
  >
  > 如果从F的函数依赖中能够推出X--->Y，则称F逻辑蕴X--->Y

- **Armstrong公理**
  - A1自反律：若Y⊆X，则X→Y;
  - A2增广律：若X→Y，则XZ→YZ;
  - A3传递律：若X→Y，Y→Z，则X→Z;
- **公理的推论**
  - 合并规则：X→Y，X→Z则X→YZ
  - 分解规则：若X→YZ，则X→Y,X→Z。
  - 伪传递规则：若X→Y、WY→Z，则WX→Z
- **闭包的计算**

```
例：设有关系模式R<U,F>，其中U={A,B,C,D,E,I}，
    F={A→D,AB →E,BI →E,CD →I,E →C}
    计算(AE)+

(1)X0={AE}
(2)X1=X0∪{DC}={ACDE}，用了A→D和E→C
(3)X2=X1∪{I}={ACDEI}，用了CD →I
(4)F中没有左部属于X2的函数依赖，X2为(AE)+
```

- **函数依赖集合的等价与覆盖**

> 如果F+=G+，就说函数依赖集F覆盖G或F与G等价

```
例：F={A→B， B→C}，G={A→BC， B→C}，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+。
     ∵AG+=ABC，∴B⊆AG+，∴A→B∈G+ (定理4.6)
       又∵BG+=BC，∴C⊆BG+，∴B→C∈G+
               ∴F⊆G+
  (2) 然后检查G中的每一个函数依赖是否属于F+。
     ∵AF+=ABC，∴BC⊆AF+，∴A→BC∈F+
       又∵BF+=BC，∴C⊆BF+，∴B→C∈F+
                ∴G⊆F+
 由 (1) 和 (2) 可得F和G等价。
```

- **最小函数依赖集**

> 若F满足下列条件，则称其为一个最小函数依赖集Fm。
>
> (1) F中每个函数依赖的右部都是单属性；【右部单属性】
>
> (2) 对于F的任一函数依赖X→A，F-{X→A}与F都不等价，即无多余函数依赖；【无多余函数依赖】
>
> (3) 对于F中的任一函数依赖X→A和X的真子集X′，
>
> (F-(X→A))U{X′→A}与F都不等价,即左部无多余 属性。【左部没有多余属性】

```
例：哪个是最小依赖集？
(1)F1={A →D,BD →C,C →AD}
(2)F2={AB →C,B →A,B →C}
(3)F3={BC →D,D →A,A →D}

F1中第三个依赖的右部不是单属性
F2中第一个依赖左部有多余属性A
F3满足最小依赖集的三个条件

例：设有F={B→C，C→AB，BC→A}，求与F等价的最小函数依赖集。
(1)分解C→AB，F={B→C，C→A，C→B，BC→A}
(2)判断B→C是否冗余，F'={C→A，C→B，BC→A}
     B+ = B，B→C非冗余。 F={B→C，C→A，C→B，BC→A}
   判断C→A是否冗余，F'={B→C，C→B，BC→A}
    C+ = ABC，C→A冗余。 F={B→C，C→B，BC→A}
   判断C→B是否冗余，F'={B→C，BC→A}
    C+ = C，C→B非冗余。 F={B→C，C→B，BC→A}
   判断BC→A是否冗余，F'={B→C，C→B}
```

```
  BC+ = BC, BC→A非冗余。F={B→C, C→B, BC→A}
 判断BC→A。 B+ = ABC, A⊆B+ , 则C在BC→A中是多余的。
(3) Fm={B→C, C→B, B→A}
```

- **一个函数依赖集的最小集不是惟一的**

- **关系模式的分解**
  - 分解等价
    - 无损连接性

      > R<U, F>, 若R的分解ρ={R1, R2....Rk}对R中任何一个关系r, 有:
      >
      > r=ΠR1(r) ⋈ ΠR2(r) ⋈ ... ⋈ ΠRk(r)
      >
      > 称分解ρ具有无损连接性

    - 依赖保持性

      > 设F是关系模式 R的函数依赖集, Z是R的一个属性集合, 则称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ΠZ(F), 有:
      >
      > ΠZ(F)={X→Y|X→Y∈F+且XY⊆Z}
      >
      > 例如F={A→B,C→B,B→D,D→C}, 设Z=CD, 则ΠCD(F)={C→D,D→C}
      >
      > 设关系模式R的一个分解 r={R1,R2,...,Rk}, F是R的函数依赖集, 如果F等价于ΠR1(F)∪ΠR2(F)∪...∪ΠRk(F) , 则称分解r具有函数依赖保持性

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

  - **无损连接性的判断**
    - 多模式

      ```
      (1)构造一个k行n列的表, 第i行对应于关系模式Ri, 第j列对应于属性Aj。如果Aj∈Ri, 则在第i行第j列上放符号aj, 否则放符号bij。 (属于用a代表, 且位置信息用j表示; 不属于用b代表,且位置信息用ij表示。)

      (2)重复考察F中的每一个函数依赖, 并修改表中的元素。
        其方法如下: 取F中一个函数依赖X→Y, 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aj, 则将bij改为aj; 若其中无aj, 则全部改为bij (i是这些行的行号最小值) 。

      (3)如果发现表中某一行变成了a1, a2, …, an, 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
      ```

    - 两个模式

      ```
      设ρ=(R1, R2)是R的一个分解, F是R上的函数
        依赖集, 分解ρ具有无损连接性的充分必要条件是:
                R1∩R2→(R1-R2)∈F+
            或  R1∩R2→(R2-R1)∈F+
      ```

  - **3NF保持无损连接及函数连接分解**

```
3NF的保持无损连接及函数依赖的分解：
   设：  R<U, F>
   1)最小化：求F的最小函数依赖集Fm
   2)排除：对Fm中任一X→A，若XA=U则不分解（R已为3NF），结束。
   2)独立：若R中Z属性在Fm中未出现，则所有Z为一个子模式，令U=U-Z。
   3)分组：对Fm中 X→A1, …. X→An，用合成规则合成一个，再对Fm中每个X→A，令Ri=XA。
   4) R的分解为{R1, R2, ….RK}
   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{R1, R2, ….RK , X}，如果存在Ri属于Rj，则删去Ri

例：设关系模式R<U,F>, U={E,G,H,I,J}, F={E→I,J→I,I→G,GH→I,IH→E}，将其分解为3NF且同时具有无损连接性和函数依赖保持性

(1)求出最小依赖集为Fm={E→I,J→I,I→G,GH→I,IH→E}
   得到分解为：{EI,JI,IG,GHI,IHE}

(2)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(JH)+=IJHGE=U，所以JH是R的唯一候选码

(3)上面的分解中没有子模式含有JH，加上候选码并去重后得到最终的分解：{JI,GHI,IHE,JH}
```

- **BCNF保持无损连接分解**

```
BCNF的保持无损连接的分解：
 (1) 令ρ={R};
 (2) 如果ρ中所有关系模式都是BCNF，则转 (4) ;
 (3) 如果ρ中有一个关系模式Ri<Ui, Fi>不是BCNF，则Ri中必有X→A∈Fi+(A□X)，且X不是Ri的码。设S1=XA, S2=Ui-A，用分解{S1, S2}代替Ri<Ui, Fi>，转 (2) ;
 (4) 分解结束，输出ρ。
```

- **分解为4NF且保持无损连接的方法**

```
分解成4NF且保持无损连接性的方法
(1)令ρ={R}
(2)如果ρ中所有关系模式都是4NF，则转(4)
(3)如果ρ中一个关系模式Ri<Ui,Fi>不是4NF，则Ri中 必存在一个非平凡的多值依赖X→→Y，且X不是Ri的码，Y不属于X，XY不等于Ri。这时令Z=Y-X，于是从X→→Y,X→→X,利用多值依赖的分解规则得到X→→Z。设S1=XZ，S2=Ui-Z，用分解{S1,S2}代替Ri，转(2)
(4)分解结束
```