# 第6章 关系数据库模式设计



金培权（jpq@ustc.edu.cn）

# 本章主要内容

- 关系模式的设计问题
- 函数依赖
- 关系模式的分解
- 关系模式的范式

# 三、模式分解

- 概念
- 无损连接(Lossless Join)
- 保持函数依赖(Preserve Dependency)

# 1、模式分解的概念

- 设有关系模式R（U）和R1(U1)，R2(U2)，…，Rk(Uk)，其中U＝U1 ∪ U2 … ∪ Uk，设ρ＝{R1, R2,…, Rk}，则称ρ为R的一个分解
- 模式分解的含义
  - 属性集的分解
  - 函数依赖集的分解
    - R(A,B,C), F={A→B, C→B}, 则分解为R1(A,B), R2(A,C)丢失了C→B

# 2、模式分解的标准

- 具有无损连接
- 要保持函数依赖
- 既具有无损连接，又要保持函数依赖

# 3、无损连接

- 动机
- 概念
- 无损连接的测试

# （1）动机

| S＃ | Status | City |
| --- | --- | --- |
| S3 | 30 | Paris |
| S5 | 30 | Athens |

| S＃ | Status |
| --- | --- |
| S3 | 30 |
| S5 | 30 |

| Status | City |
| --- | --- |
| 30 | Paris |
| 30 | Athens |

$\bowtie$

| S＃ | Status | City |
| --- | --- | --- |
| S3 | 30 | Paris |
| S3 | 30 | Athens |
| S5 | 30 | Paris |
| S5 | 30 | Athens |

- 模式分解的过程应是可逆的，**R**的所有数据在分解后应没有丢失

信息丢失

分解后要不能得到**S3**的**City**是**Paris**

# （2）概念

- 设R是关系模式，分解成关系模式 ρ＝{R1,R2,…,Rk}，F是R上的一个FD集，若对R中满足F的每个关系r，都有：
  $r=\pi_{R1}(r) \bowtie \pi_{R2}(r) \bowtie \dots \bowtie \pi_{Rk}(r)$ ,则称这个分解ρ相对于F是“无损连接分解”
  - R的每个关系r是它在$R_i$上的投影的自然连接
  - 无损连接保证R分解后还可以通过$R_i$恢复

# （2）概念

- **我们记** $m_{\rho}(r)=\mathop{\infty}\limits_{i=1}^{k}\pi_{Ri}(r)$

- **则对于关系模式R关于F的无损连接条件是r＝$m_{\rho}$(r)**

# （2）概念

R

| S# | Status | City |
|---|---|---|
| S3 | 30 | Paris |
| S5 | 30 | Athens |

R1

| S# | Status |
|---|---|
| S3 | 30 |
| S5 | 30 |

R2

| Status | City |
|---|---|
| 30 | Paris |
| 30 | Athens |

$\bowtie$

| S# | Status | City |
|---|---|---|
| S3 | 30 | Paris |
| S3 | 30 | Athens |
| S5 | 30 | Paris |
| S5 | 30 | Athens |

$$m_\rho(r) = \pi_{R1}(r) \infty \pi_{R2}(r)$$

$r \neq m_p(r)$

所以不是无损连接

(1)Select * From R

(2)Select * From R1,R2 where R1.Status=R2.Status

返回结果不一致

# （3）无损连接的测试

- 方法1：Chase
  - 输入：关系模式R(A1,A2,…,An)，R上的函数依赖集F，R的一个分解p={R1,…,Rk}
  - 输出：判断p相对于F是否具有无损连接性
  - 算法：Chase

# 1）Chase过程

- 构造一个$k$行$n$列的表格，每行对应一个模式$R_i$ (1≤$i$ ≤ $k$)，每列对应一个属性$A_j$（1≤$j$≤ $n$），若$A_j$在$R_i$中，则在表格的第$i$行第$j$列处填上$a_j$，否则填上符号$b_{ij}$
- 检查F的每个FD，并修改表格中的元素，方法如下：
  - 对于F中的函数依赖X→Y，若表格中有两行在X分量上相等，在Y分量上不相等，则修改Y：
    - 若Y的分量中有一个$a_j$，则另一个也修改为$a_j$；
    - 如果没有$a_j$，则用其中一个$b_{ij}$替换另一个符号（i是所有b中最小的行数），一直到表格不能修改为止
- 若修改后，表格中有一行是全a，即$a_1a_2…a_n$，则ρ相对于F是无损连接的分解，否则不是

# 1）Chase过程

- 扫描一次F后，若表格中未出现全a的行，则进行下一次扫描
  - 由于每次扫描F至少能减少一个符号，而符号有限，因此算法最后必然终止
  - 终止条件
    - 全a行
    - 表格扫描后不再发生任何修改

# 2）Chase示例

- R(A,B,C,D,E)
  - R1(A,D), R2(A,B), R3(B,E), R4(C,D,E), R5(A,E)
  - F={A→C, B→C, C→D, DE→C, CE→A}
- 判断R分解为p＝{R1,R2,R3,R4,R5}是否是无损连接的分解

# 2）Chase示例

## 1、构造初始表格

R1(A,D), R2(A,B), R3(B,E), R4(C,D,E), R5(AE)

| | A | B | C | D | E |
|---|---|---|---|---|---|
| AD | a1 | b12 | b13 | a4 | b15 |
| AB | a1 | a2 | b23 | b24 | b25 |
| BE | b31 | a2 | b33 | b34 | a5 |
| CDE | b41 | b42 | a3 | a4 | a5 |
| AE | a1 | b52 | b53 | b54 | a5 |

## 2、处理表格

A→C：将b23，b53改为b13

B→C：将b33改为b13

C→D：将b24，b34，b54改为a4

DE→C：将第3行和第5行的C改为a3

CE→A：将第3行和第4行的A改为a1

# 2）Chase示例

## 1、构造初始表格

R1(A,D), R2(A,B), R3(B,E), R4(C,D,E), R5(AE)

| | A | B | C | D | E |
|---|---|---|---|---|---|
| AD | a1 | b12 | b13 | a4 | b15 |
| AB | a1 | a2 | b13 | a4 | b25 |
| BE | a1 | a2 | a3 | a4 | a5 |
| CDE | a1 | b42 | a3 | a4 | a5 |
| AE | a1 | b52 | a3 | a4 | a5 |

## 2、处理表格

A→C：将b23，b53改为b13

B→C：将b33改为b13

C→D：将b24，b34，b54改为a4

DE→C：将第3行和第5行的C改为a3

CE→A：将第3行和第4行的A改为a1

因此是无损连接的分解

# 3）方法2

- 当R分解为两个关系模式R1和R2时，有一种简便的方法可以测试无损连接性
  - p＝{R1,R2}
  - p是无损连接的分解当且仅当下面之一满足
    - （R1∩R2）→（R1－R2）
    - （R1∩R2）→（R2－R1）
    - 其中R1∩R2指模式的交，返回公共属性
    - R2－R1表示模式的差集，返回属于R2但不属于R1的属性集
- 例 R(A,B,C), F={A→B}
  - ρ1={R1(A,B),R2(B,C)}, ρ2={R1(A,B),R2(A,C)}
  - ρ2是无损连接，ρ1不是

# 4、保持函数依赖

- 关系模式R的FD集在分解后仍在数据库模式中保持不变
  - 给定R和R上的一个FD集F， ρ＝{R1,R2,…, Rk}的分解应使F被Ri上的函数依赖逻辑蕴含
- 定义：设F是属性集U上的FD集，Z是U的子集，F在Z上的投影用$\pi_Z$（F）表示，定义为： $\pi_Z(F)=\{X\to Y \mid X\to Y\in F^+ \wedge XY\subseteq Z\}$。对于R(U)上的一个分解ρ ＝{R1,R2,…, Rk},若满足下面条件，则称分解ρ保持函数依赖集F：

$$\left(\bigcup_{i=1}^{k}\pi_{R_i}(F)\right)^+ = F^+$$

# （1）例子

- $R(city, street, zip)$, $F=\{(city,street)\rightarrow zip,\ zip\rightarrow city\}$
- 分解为$\rho=\{R1(street,zip), R2(city,zip)\}$
- 是否无损连接？
  - $R1\cap R2=\{zip\}$, $R2-R1=\{city\}$, $zip\rightarrow city$
  - 无损连接
- 是否保持函数依赖？
  - $\pi_{R1}(F)=\{$按自反律推出的平凡FD$\}$
  - $\pi_{R2}(F)=\{zip\rightarrow city$，以及按自反律推出的平凡FD$\}$
  - $\pi_{R1}(F)\cup\ \pi_{R2}(F)=\{zip\rightarrow city\}^+\neq F^+$
  - 不保持函数依赖

# （2）不保持函数依赖带来的问题

- R(city, street, zip), F={(city,street)→zip, zip→city}
- 分解为p={R1(street, zip)，R2(city, zip)}
- 在R1中插入('a','100081')和('a','100082')
- R2中插入('Beijing','100081')和('Beijing','100082')
- R1 $\bowtie$ R2：得到

| City | Street | Zip |
|---|---|---|
| Beijing | a | 100081 |
| Beijing | a | 100082 |

- 违反了(city,street)→zip，因为它被丢失了，语义完整性被破坏

# 模式分解小结

- 三种准则
  - 无损连接
    - 若R分解为n(n>2)个关系模式，使用Chase方法判断是否无损连接
    - 若R分解为R1和R2，使用(R1 ∩ R2) → (R1－R2)或(R1 ∩ R2) → (R2－R1) 判断
  - 保持函数依赖

$$\left(\bigcup_{i=1}^{k} \pi_{R_i}(F)\right)^{+} = F^{+}$$

  - 既无损连接，又保持函数依赖