数据库系统概论
An Introduction to Database System

第六章 关系数据理论

刘 淇
Email: qiliuql@ustc.edu.cn
课程主页：
http://staff.ustc.edu.cn/~qiliuql/DB2020HF.html

第六章 关系数据理论

- 逻辑蕴含

定义**6.11** 对于满足一组函数依赖 F 的关系模式R <U，F>，其任何一个关系r，若函数依赖$X \rightarrow Y$都成立,（即**r**中任意两元组t，s，若$t[X]=s[X]$，则$t[Y] = s[Y]$），则称F逻辑蕴含$X \rightarrow Y$

1. Armstrong公理系统

从已知的一些函数依赖，可以推导出其蕴含的另外一些函数依赖，这就需要一系列推理规则，这些规则常被称作“**Armstrong 公理**”。

判断一个关系模式属于哪个范式，严谨的方式就是根据**armstrong**公理求出其函数依赖集的正则覆盖，然后根据各个范式的定义来判断。

正则覆盖定义：F的正则覆盖Fc是一个依赖集，使得F与Fc相互逻辑蕴涵。

1. Armstrong公理系统

从已知的一些函数依赖，可以推导出其蕴含的另外一些函数依赖，这就需要一系列推理规则，这些规则常被称作“**Armstrong 公理**”。

判断一个关系模式属于哪个范式，严谨的方式就是根据armstrong公理求出其函数依赖集的正则覆盖，然后根据各个范式的定义来判断。

具体地，对关系模式$R<U，F>$来说有以下的推理规则：

- A1.自反律（Reflexivity）：若$Y \subseteq X \subseteq U$，则$X \rightarrow Y$为$F$所蕴含。
- A2.增广律（Augmentation）：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（Transitivity）：若$X \rightarrow Y$及$Y \rightarrow Z$为F所蕴含，则$X \rightarrow Z$为F所蕴含。

定理 6.1 Armstrong推理规则是正确的

（1）自反律: 若$Y \subseteq X \subseteq U$，则$X \rightarrow Y$为$F$所蕴含

证: 设$Y \subseteq X \subseteq U$

对R <U，F> 的任一关系r中的任意两个元组t，s：

若$t[X]=s[X]$，由于$Y \subseteq X$，有$t[y]=s[y]$，

所以$X \rightarrow Y$成立，自反律得证

定理 6.1 Armstrong推理规则是正确的（续）

(2)增广律: 若$X \to Y$为F所蕴含，且$Z \subseteq U$，则$XZ \to YZ$ 为F所蕴含。

证：设$X \to Y$为F所蕴含，且$Z \subseteq U$。

设R<U，F> 的任一关系r中任意的两个元组t，s：

若$t[XZ]=s[XZ]$，则有$t[X]=s[X]$和**$t[Z]=s[Z]$**；

由$X \to Y$，于是有$t[Y]=s[Y]$，所以**$t[YZ]=s[YZ]$**，所以

$XZ \to YZ$为F所蕴含，增广律得证。

定理 6.1 Armstrong推理规则是正确的（续）

(3) 传递律：若$X \to Y$及$Y \to Z$为F所蕴含，则 $X \to Z$为 F所蕴含。

证：设$X \to Y$及$Y \to Z$为F所蕴含。

对R<U，F> 的任一关系 r中的任意两个元组 t，s：

若$t[X]=s[X]$，由于$X \to Y$，有 $t[Y]=s[Y]$；

再由$Y \to Z$，有$t[Z]=s[Z]$，所以$X \to Z$为F所蕴含，传递

律得证。

2. 导出规则

1.根据A1，A2，A3这三条推理规则可以得到下面三条推理规则：

- 合并规则：由$X \rightarrow Y$，$X \rightarrow Z$，有$X \rightarrow YZ$。
  （A2， A3）
- 伪传递规则：由$X \rightarrow Y$，$WY \rightarrow Z$，有$XW \rightarrow Z$。
  （A2， A3）
- 分解规则：由$X \rightarrow Y$及 $Z \subseteq Y$，有$X \rightarrow Z$。
  （A1， A3）

导出规则

2.根据合并规则和分解规则，可得引理6.1

引理6.1 $X \rightarrow A_1 A_2 \ldots A_k$成立的充分必要条件是$X \rightarrow A_i$成立（$i$=1，2，…，$k$）

Armstrong公理系统

- 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

定义6.I2　在关系模式*R*<*U*，*F*>中为*F*所逻辑蕴含的函数依赖的全体叫作*F*的闭包，记为F^+。

3. 函数依赖闭包

定义6.12 在关系模式R<U，F>中为F所逻辑蕴含的函数依赖的全体叫作 F的闭包，记为F^+。

定义6.13 设F为属性集U上的一组函数依赖，$X \subseteq U$，$X_F^+ = \{ A \mid X \rightarrow A$能由$F$ 根据Armstrong公理导出$\}$，X_F^+称为属性集X关于函数依赖集F 的闭包

F的闭包

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

$F^{+}=\{$

$X \rightarrow \varphi,\quad Y \rightarrow \varphi,\quad Z \rightarrow \varphi,\quad XY \rightarrow \varphi,\ XZ \rightarrow \varphi,\ YZ \rightarrow \varphi,\ XYZ \rightarrow \varphi,$

$X \rightarrow X,\quad Y \rightarrow Y,\quad Z \rightarrow Z,\quad XY \rightarrow X,\ XZ \rightarrow X,\ YZ \rightarrow Y,\ XYZ \rightarrow X,$

$X \rightarrow Y,\quad Y \rightarrow Z,\quad XY \rightarrow Y,\ XZ \rightarrow Y,\ YZ \rightarrow Z,\ XYZ \rightarrow Y,$

$X \rightarrow Z,\quad Y \rightarrow YZ,\quad XY \rightarrow Z,\ XZ \rightarrow Z,\ YZ \rightarrow YZ,\ XYZ \rightarrow Z,$

$X \rightarrow XY,\quad XY \rightarrow XY,\ XZ \rightarrow XY,\quad XYZ \rightarrow XY,$

$X \rightarrow XZ,\quad XY \rightarrow YZ,\ XZ \rightarrow XZ,\quad XYZ \rightarrow YZ,$

$X \rightarrow YZ,\quad XY \rightarrow XZ,\ XZ \rightarrow XY,\quad XYZ \rightarrow XZ,$

$X \rightarrow ZYZ,\quad XY \rightarrow XYZ,\ XZ \rightarrow XYZ,\quad XYZ \rightarrow XYZ\ \}$

$F=\{X \rightarrow A1, \ldots\ldots, X \rightarrow An\}$的闭包$F^{+}$计算是一个NP完全问题

关于闭包的引理

- **引理6.2**

  设F为属性集U上的一组函数依赖，X，$Y \subseteq U$，$X \rightarrow Y$能由F 根据Armstrong公理导出的充分必要条件是$Y \subseteq X_F^+$

- 用途

  将判定$X \rightarrow Y$是否能由F根据Armstrong公理导出的问题，转化为求出X_F^+、判定Y是否为X_F^+的子集的问题

求闭包的算法

算法6.1 求属性集X（$X \subseteq U$）关于U上的函数依赖集F 的闭包X_F^+

输入：X，F　　　　输出：X_F^+

步骤：

（1）令$X^{(0)}=X$，$i=0$

（2）**求B，这里$B=\{A \mid (\exists V)(\exists W)(V \rightarrow W \in F \wedge V \subseteq X^{(i)} \wedge A \in W)\}$;**

（3）$X^{(i+1)}=B \cup X^{(i)}$

（4）判断$X^{(i+1)}=X^{(i)}$吗?

（5）若相等或$X^{(i)}=U$，则$X^{(i)}$就是X_F^+，算法终止。

（6）若否，则 $i=i+1$，返回第（2）步。

算法6.1

对于算法6.1， 令$a_i = |X^{(i)}|$，$\{a_i\}$形成一个步长大于1的严格递增的序列，序列的上界是$|U|$，因此该算法最多$|U| - |X|$次循环就会终止。

函数依赖闭包

[例1] 已知关系模式$R<U, F>$，其中

$U=\{A, B, C, D, E\}$;

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

求（AB）$_F^+$。

解 设$X^{(0)}=AB$;

(1) $X^{(1)}=AB\cup CD=ABCD$。

(2) $X^{(0)}\neq X^{(1)}$

$X^{(2)}=X^{(1)}\cup BE=ABCDE$。

(3) $X^{(2)}=U$，算法终止

→（AB）$_F^+=ABCDE$。

4. Armstrong公理系统的有效性与完备性

- 定理6.2 Armstrong公理系统是有效的、完备的
- 证明：

1. 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）

   可由定理6.1得证

定理6.1：Armstrong推理规则是正确的

2. 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）

   只需证明逆否命题: 若函数依赖$X \rightarrow Y$不能由*F*从Armstrong公理导出，那么它必然不为*F*所蕴含

Armstrong公理系统完备性证明

只需证明逆否命题：若函数依赖$X \rightarrow Y$不能由F从Armstrong公理导出，那么它必然不为F所蕴含

分以下三步：

(1) 引理：若$V \rightarrow W$成立，且$V \subseteq X_F^+$，则$W \subseteq X_F^+$

(2) 构造一张二维表r，它由下列两个元组构成，可以证明r必是R（U，F）的一个关系，即F^+中的全部函数依赖在 r上成立。

$$\begin{array}{cc} \overbrace{\begin{array}{c} 11\ldots\ldots1 \\ 11\ldots\ldots1 \end{array}}^{X_F^+} & \qquad \overbrace{\begin{array}{c} 00\ldots\ldots0 \\ 11\ldots\ldots1 \end{array}}^{U\text{-}X_F^+} \end{array}$$

(3) 若$X \rightarrow Y$ 不能由F从Armstrong公理导出，则Y 不是X_F^+的子集。

5. 函数依赖集等价

定义6.14 如果$G^+=F^+$，就说函数依赖集F覆盖G（F是G的覆盖，或G是F的覆盖），或F与G等价。

引理6.3 $F^+ = G^+$的充分必要条件是$F \subseteq G^+$，和$G \subseteq F^+$

证: 必要性显然，只证充分性。

（1）若$F \subseteq G^+$，则$X_F^+ \subseteq X_{G+}^+$。

（2）任取$X \rightarrow Y \in F^+$ 则有 $Y \subseteq X_F^+ \subseteq X_{G+}^+$。

所以$X \rightarrow Y \in (G^+)^+ = G^+$。即$F^+ \subseteq G^+$。

（3）同理可证$G^+ \subseteq F^+$，所以$F^+ = G^+$。

6. 最小依赖集

定义6.15 如果函数依赖集F满足下列条件，则称F为一个极小函数依赖集。亦称为最小依赖集或最小覆盖。

(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→A，使得F与F-{X→A}等价。

(3) F中不存在这样的函数依赖X→A， X有真子集Z使得F-{X→A}∪{Z→A}与F等价。

最小依赖集

[例2] 关系模式*S*<*U*，*F*>，其中：

U={ Sno，Sdept，Mname，Cno，Grade }，

F={ Sno→Sdept，Sdept→Mname，(Sno，Cno)→Grade }

设F′={Sno→Sdept，Sno→Mname，Sdept→Mname，

(Sno，Cno)→Grade，(Sno，Sdept)→Sdept}

*F*是最小覆盖，而*F*′不是。

因为：*F* ′ - {Sno→Mname}与*F* ′等价

F ′ - {(Sno，Sdept)→Sdept}也与*F* ′等价

7. 极小化过程

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。此F_m称为F的最小依赖集。

证明: 构造性证明，找出F的一个最小依赖集。

极小化过程（续）

(1)逐一检查F中各函数依赖FD_i：$X\rightarrow Y$，若$Y=A_1A_2\ldots A_k$，$k>2$，

则用 $\{X\rightarrow A_j \mid j=1，2，\ldots，k\}$ 来取代$X\rightarrow Y$。

(2)逐一检查F中各函数依赖FD_i：$X\rightarrow A$，令$G=F-\{X\rightarrow A\}$，

若$A\in X_G^+$， 则从F中去掉此函数依赖。

(3)逐一取出F中各函数依赖FD_i：$X\rightarrow A$，设$X=B_1B_2\ldots B_m$，

逐一考查B_i（i=1，2，…，m），若$A\in (X-B_i)_F^+$，

则以$X-B_i$取代X。

极小化过程（续）

[例3] $F = \{A \to B, B \to A, B \to C, A \to C, C \to A\}$

F_{m1}、F_{m2}都是F的最小依赖集：

$$F_{m1} = \{A \to B, B \to C, C \to A\}$$

$$F_{m2} = \{A \to B, B \to A, A \to C, C \to A\}$$

- F的最小依赖集F_m不唯一
- 极小化过程(定理6.3的证明)也是检验F是否为极小依赖集的一个算法

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 模式的分解

6.5 小结

6.4 模式的分解

- 把低一级的关系模式分解为若干个高一级的关系模式的方法不是唯一的
- 只有能够保证分解后的关系模式与原关系模式等价，分解方法才有意义

关系模式分解的标准

三种模式分解等价的定义：

1. 分解具有无损连接性

2. 分解要保持函数依赖

3. 分解既要保持函数依赖，又要具有无损连接性

模式的分解（续）

定义6.16 关系模式R<U,F>的一个分解：

ρ={ R_1<U_1,F_1>，R_2<U_2,F_2>，…，R_n<U_n,F_n>}

$U=\bigcup_{i=1}^{n} U_i$，且不存在 $U_i \subseteq U_j$，F_i 为 F在 U_i 上的投影

定义6.17 函数依赖集合$\{X\rightarrow Y \mid X\rightarrow Y \in F^+ \wedge XY \subseteq U_i\}$ 的一个覆盖 F_i 叫作 ***F* 在属性 U_i 上的投影**

模式的分解（续）

例：S-L（Sno， Sdept， Sloc）

F={ Sno→Sdept,Sdept→Sloc,Sno→Sloc}

S-L∈2NF

分解方法可以有多种：

1. S-L分解为三个关系模式：SN(Sno)
   SD(Sdept)
   SO(Sloc)
2. SL分解为下面二个关系模式： NL(Sno, Sloc)
   DL(Sdept, Sloc)
3. 将SL分解为下面二个关系模式： ND(Sno, Sdept)
   NL(Sno, Sloc)

具有无损连接性的模式分解

- 关系模式R<U,F>的一个分解 ρ={ R_1<U_1,F_1>，R_2<U_2,F_2>， …，R_n<U_n,F_n>}

  若**R**与**R1**、**R2**、**…**、**Rn**自然连接的结果相等，则称关系模式**R**的这个分解ρ具有无损连接性（**Lossless join**）

- 具有无损连接性的分解保证不丢失信息
- 无损连接性不一定能解决插入异常、删除异常、修改复杂、数据冗余等问题

模式的分解（续）

第2、3种分解方法具有无损连接性

1. S-L分解为三个关系模式：SN(Sno)
   SD(Sdept)
   SO(Sloc)
2. SL分解为下面二个关系模式： NL(Sno, Sloc)
   DL(Sdept, Sloc)
3. 将SL分解为下面二个关系模式： ND(Sno, Sdept)
   NL(Sno, Sloc)

问题:第2、3种分解方法没有保持原关系中的函数依赖

- 例如， 对于第三种分解方法，SL中的函数依赖Sdept→Sloc没有投影到关系模式ND、NL上

保持函数依赖的模式分解

设关系模式R<U,F>被分解为若干个关系模式

$R_1<U_1,F_1>$，$R_2<U_2,F_2>$，…，$R_n<U_n,F_n>$

（其中$U=U_1 \cup U_2 \cup \ldots \cup U_n$，且不存在$U_i \subseteq U_j$，$F_i$为F在$U_i$上的投影），**若F所逻辑蕴含的函数依赖一定也由分解得到的某个关系模式中的函数依赖F_i所逻辑蕴含，则称关系模式R的这个分解是保持函数依赖的（Preserve dependency）**

模式的分解（续）

4. 将SL分解为下面二个关系模式：

ND(Sno, Sdept)

DL(Sdept, Sloc)

这种分解方法就保持了函数依赖

模式的分解（续）

- 如果一个分解具有无损连接性，则它能够保证不丢失信息
- 如果一个分解保持了函数依赖，则它可以减轻或解决各种异常情况
- 分解具有无损连接性和分解保持函数依赖是两个互相独立的标准。具有无损连接性的分解不一定能够保持函数依赖；同样，保持函数依赖的分解也不一定具有无损连接性。

模式的分解（续）

第1种分解方法既不具有无损连接性，也未保持函数依赖，

它不是原关系模式的一个等价分解

第2种分解方法保持了函数依赖，但不具有无损连接性

第3种分解方法具有无损连接性，但未持函数依赖

第4种分解方法既具有无损连接性，又保持了函数依赖

分解算法

- 算法6.2 判别一个分解的无损连接性
- 算法6.3（合成法）转换为3NF的保持函数依赖的分解。
- 算法6.4 转换为3NF既有无损连接性又保持函数依赖的分解
- 算法6.5 （分解法）转换为BCNF的无损连接分解
- 算法6.6 达到4NF的具有无损连接性的分解

6.1 问题的提出

6.2 规范化

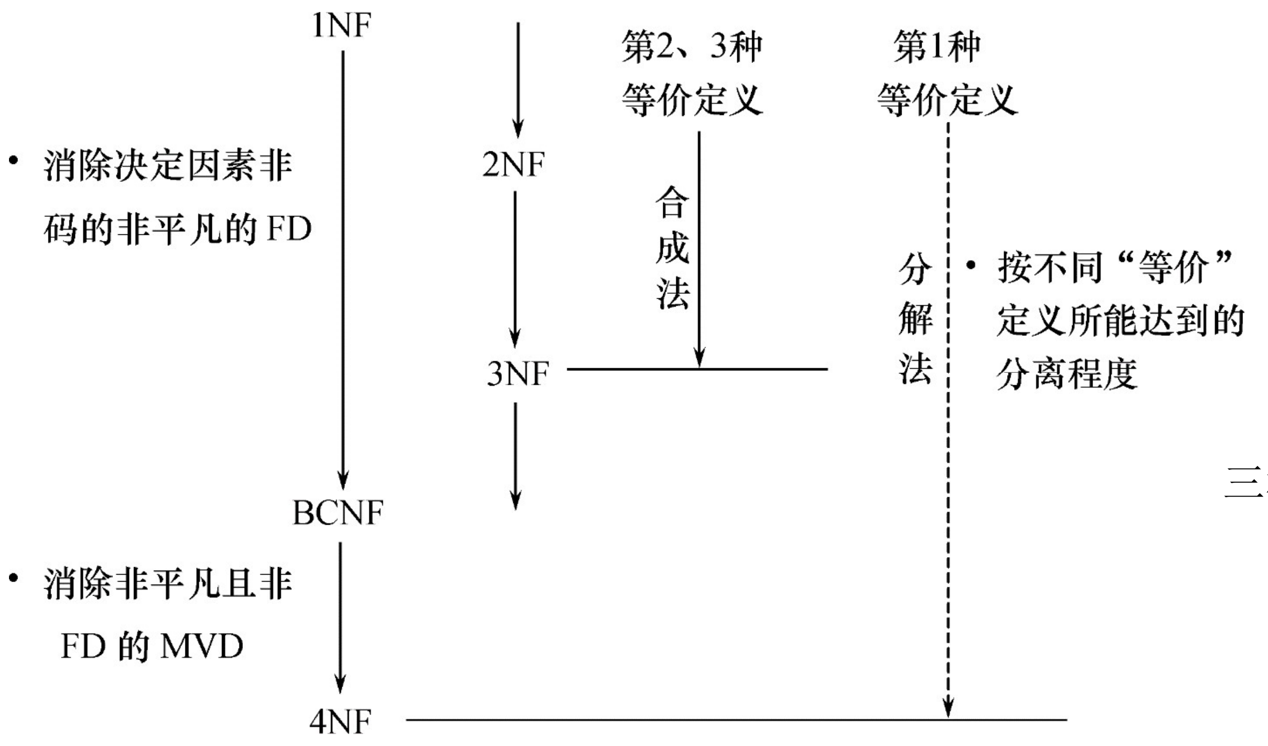
6.3 数据依赖的公理系统

*6.4 模式的分解

6.5 小结

6.5 小结

关系模式的规范化，其基本思想：



三种模式分解等价的定义：

1. 分解具有无损连接性

2. 分解要保持函数依赖

3. 分解既要保持函数依赖，又要具有无损连接性

小结(续)

- 若要求分解具有无损连接性，那么模式分解一定能够达到4NF
- 若要求分解保持函数依赖，那么模式分解一定能够达到3NF，但不一定能够达到BCNF
- 若要求分解既具有无损连接性，又保持函数依赖，则模式分解一定能够达到3NF，但不一定能够达到BCNF

小结(续)

- 规范化理论为数据库设计提供了理论的指南和工具
  - 也仅仅是指南和工具

- 并不是规范化程度越高，模式就越好
  - 必须结合应用环境和现实世界的具体情况合理地选择数据库模式