2026年春季学期

数据库系统概论

An Introduction to Database Systems

第六章 关系数据理论

中国科学技术大学
人工智能与数据科学学院
黄振亚，huangzhy@ustc.edu.cn

第六章 关系数据理论

6.3 数据依赖的公理系统

- 回顾规范化理论
  - 函数依赖，部分函数依赖，传递函数依赖，成为判断关系模式是好是坏的因素
  - 问题：如何找到全部的函数依赖？
    - 现实世界的语义约束
    - 有没有别的办法？

1. Armstrong公理系统

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

- **Armstrong公理系统**
  - **一套推理规则，是模式分解算法的理论基础**
  - **用途**
    - **求给定关系模式的码**
    - **从一组函数依赖求得蕴涵的函数依赖**

已知: R<U，F>, U={X,C,W,Y,Z}, F={X →YZ, Z→CW}
问： X →CWYZ是否为F逻辑蕴含？R的码？

6.3 数据依赖的公理系统

- 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

- **Armstrong公理系统　（推理系统）**

设U为属性集总体，F是U上的一组函数依赖，于是有关系模式R <U,F >。对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(并集)**
  - **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

A1. 自反律: 若$Y \subseteq X \subseteq U$，则$X \rightarrow Y$为$F$所蕴含

证明:

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

注意：由自反律所得到的函数依赖均是平凡的函数依赖，自反律的使用并不依赖于***F***。

定理 6.1 Armstrong推理规则是正确的（续）

A2. 增广律: 若$X \rightarrow Y$为F所蕴含, 且$Z \subseteq U$，则$XZ \rightarrow YZ$ 为F所蕴含

证明：设$X \rightarrow Y$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$\boldsymbol{t[Z]=s[Z]}$；

因为$X \rightarrow Y$，则$t[Y]=s[Y]$，所以$\boldsymbol{t[YZ]=s[YZ]}$，

所以：$XZ \rightarrow YZ$为F所蕴含

增广律得证。

定理 6.1 Armstrong推理规则是正确的（续）

A3. 传递律：若$X\rightarrow Y$及$Y\rightarrow Z$为F所蕴含，则 $X\rightarrow Z$为 F所蕴含。

证明：设$X\rightarrow Y$及$Y\rightarrow Z$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$X\rightarrow Y$，有 $t[Y]=s[Y]$；

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

传递律得证。

2. 导出规则

- 1. 根据A1，A2，A3可以得到下面三条推理规则：
  - 合并规则：由$X \to Y$，$X \to Z$，有$X \to YZ$
  - 伪传递规则：由$X \to Y$，$WY \to Z$，有$XW \to Z$
  - 分解规则：由$X \to Y$及 $Z \subseteq Y$，有$X \to Z$

- 2. 根据合并规则和分解规则，可得引理6.1

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

证明课后思考

2. 导出规则

- 1. 根据A1，A2，A3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）X →YX，XY →ZY
  - 伪传递规则：由$X \rightarrow Y$，$WY \rightarrow Z$，有$XW \rightarrow Z$

    （A2，A3）XW →YW
  - 分解规则：由$X \rightarrow Y$及 $Z \subseteq Y$，有$X \rightarrow Z$

    （A1，A3）$Z \subseteq Y$，$Y \rightarrow Z$
- 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，…，$k$）

F的闭包

- **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^+。

- Armstrong公理系统（A1,A2,A3）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

Armstrong公理系统

- 证明有效性：显然
- 证明完备性，就首先要解决如何判定一个函数依赖是否属于由F根据Armstrong公理系统推导出来的函数依赖的集合
  - 如果能够求出这个集合，问题就解决了
  - 但是，这个是一个NP完全问题

F的闭包

R<U,F>, U=(X, Y, Z), F={X→Y, Y→Z}

F^+={

X→φ,	Y→φ,	Z→φ,	XY→φ,	XZ→φ,	YZ→φ,	XYZ→φ,
X→X,	Y→Y,	Z→Z,	XY→X,	XZ→X,	YZ→Y,	XYZ→X,
X→Y,	Y →Z,		XY→Y,	XZ→Y,	YZ→Z,	XYZ→Y,
X→Z,	Y→YZ,		XY→Z,	XZ→Z,	YZ→YZ,	XYZ→Z,
X→XY,			XY→XY,	XZ→XY,		XYZ→XY,
X→XZ,			XY→YZ,	XZ→XZ,		XYZ→YZ,
X→YZ,			XY→XZ,	XZ→XY,		XYZ→XZ,
X→XYZ,			XY→XYZ,	XZ→XYZ,		XYZ→XYZ

}

F={X→A1, …… , X→An}，求F的闭包F^+计算是一个NP完全问题

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**

关于闭包的引理

- 引理6.2 （引理6.1得出）

  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$Y \subseteq X_F^+$

- 引理6.2的用途
  - 1. 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
  - 2. 如果X_F^+ =U，则X是R<U, F>的候选码
- X_F^+ 可以用算法求得

求属性集X关于F的闭包X_F^+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^+

迭代

输入：X，F　　　　输出：X_F^+

对$X^{(i)}$中的每个元素，依次检查相应的函数依赖,将依赖它的属性加入*B*

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$X^{(i+1)} = X^{(i)}$相等或$X^{(i)}=U$，则$X^{(i)}$就是X_F^+，算法终止。

（6）若否，则 i=i+l，返回第（2）步。

函数依赖闭包

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

[例6.11] 已知关系模式R<U，F>，其中

U={A，B，C，D，E}；

F={AB→C，B→D，C→E，EC→B，AC→B}。

求（AB）$_F^+$。

解 设$X^{(0)}$=AB；

(1) $X^{(1)}$ = AB∪CD = ABCD。

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

$X^{(2)}=X^{(1)}$∪BE=ABCDE。

(3) $X^{(2)}$=U，算法终止

→（AB）$_F^+$ = ABCDE。

计算$X^{(1)}$：逐一的扫描F集合中各个函数依赖，找左部为A、B或AB的函数依赖。得到两个：AB→C，B→D

于是$X^{(1)}$=AB∪CD=ABCD。

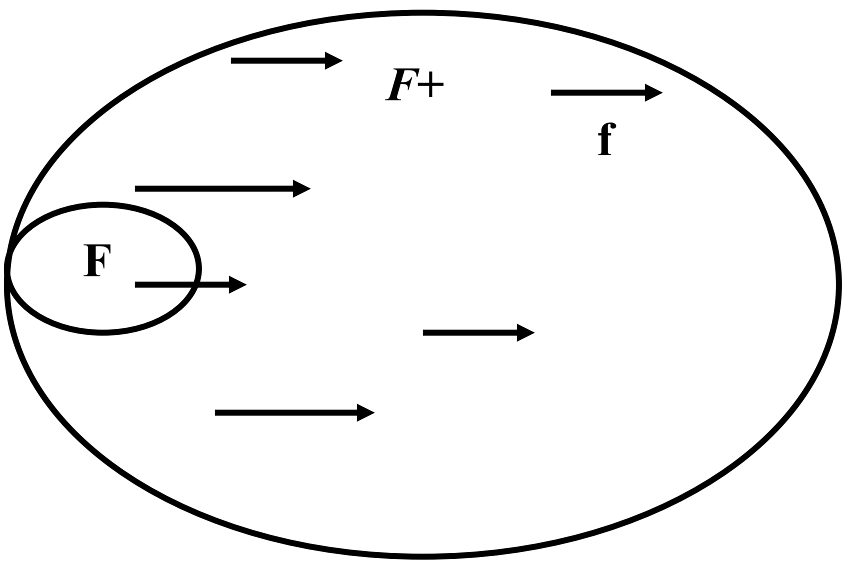
找出左部为ABCD子集的那些函数依赖，又得到C→E，AC→B，于是$X^{(2)}=X^{(1)}$∪BE=ABCDE

算法6.1

- 算法6.1 一定会终止
- 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。

4. Armstrong公理系统的有效性与完备性

- 定理6.2 Armstrong公理系统（A1,A2,A3）是有效的、完备的
  - 明确：1.公理系统推导出来的 f 正确？

    2. F^+中的每一个 f 都能推导出来？
- 

f 不能由F 导出, f $\notin$ F+

4. Armstrong公理系统的有效性与完备性

- 定理6.2 Armstrong公理系统（A1,A2,A3）是有效的、完备的
- 证明（了解）：

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

   完备性：所有不能用Armstrong公理推导出来f, 都不为真

   若 f 不能用Armstrong公理推导出来， f $\notin$ ***F***+

4. Armstrong公理系统的有效性与完备性

- 证明完备性：*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$X \to Y$不能由*F*从Armstrong公理导出，那么它必然不为*F*所蕴含
  - 证明分3步
  - (1) 若$V \to W$成立，且$V \subseteq X_F^+$，则$W \subseteq X_F^+$

证 因为 $V \subseteq X_F^+$，所以有$X \to V$成立；（?）

因为$X \to V$，$V \to W$，于是$X \to W$成立

所以$W \subseteq X_F^+$

- 引理6.2（引理6.1得出）

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

4. Armstrong公理系统的有效性与完备性

- 证明完备性逆否命题: 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
  - (2) 构造一张二维表r，它由下列两个元组构成，

可证明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

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

若r 不是$R<U,F>$ 的关系，则必由于F中有某一个函数依赖$V \rightarrow W$在r上不成立所致。由r 的构成可知，V 必定是X_F^+ 的子集，而W 不是 X_F^+ 的子集，可是由第（1）步，$W \subseteq X_F^+$，矛盾。

所以r 必是$R<U,F>$的一个关系。

4. Armstrong公理系统的有效性与完备性

- 证明完备性逆否命题: 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
  - (3) 若X→Y 不能由F从Armstrong公理导出，则Y 不是XF+ 的子集。（引理6.2）
    - 因此必有Y 的子集Y’ 满足 Y’ $\subseteq$ U-XF+,
    - 则X→Y在 r 中不成立,
    - 即X→Y必不为 R<U，F> 蕴含

  /* 因为 F+中的全部函数依赖在 r上成立。

- 引理6.2 （引理6.1得出）

  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$Y \subseteq X_F^+$

Armstrong公理系统完备性证明

- **Armstrong公理的完备性及有效性说明:**
  - **“导出”与“蕴涵”是两个完全等价的概念**
  - **F^+：为F所逻辑蕴涵的函数依赖的全体（定义6.12）**

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^+。

  - **F^+：可以说成由F出发借助Armstrong公理导出的函数依赖的集合**

5. 函数依赖集等价

- 引理6.2（引理6.1得出）

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$Y \subseteq X_F^+$

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**。

- **两个函数依赖集等价是指它们的闭包等价**

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

证明： 必要性显然：因为$F \subseteq F^+$， $F^+ = G^+$，所以$F \subseteq G^+$

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$F^+ = G^+$

如何判定$F \subseteq G^+$？

只需逐一对F中的函数依赖$X \rightarrow Y$考察 Y 是否属于X_{G+}^+

6. 最小依赖集

亦称为极小函数依赖集或最小覆盖

定义6.15 如果函数依赖集F满足下列条件，则称F为一个最小依赖集。

(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

即F中的函数依赖均不能由F中其他函数依赖导出

(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

F中各函数依赖左部均为最小属性集（X不存在冗余属性）

最小依赖集

[例6.12] 关系模式$S<U, F>$，其中：

U = { Sno，School，Mname，Cno，Grade }，

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

设F′ = {Sno→School，Sno→Mname，School→Mname，

(Sno，Cno)→Grade，(Sno，School)→School}

F是最小覆盖，而F'不是。

因为：F' - {Sno→Mname}与F'等价

F' - {(Sno，School)→School}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(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

极小化过程（续）

证明（了解），但需要掌握构造求解方法:

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

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

引理6.l $X \to A_1 A_2 \ldots A_k$成立的充分必要条件是$X \to A_i$成立（i=l，2，…，k）。

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

若$A \in X_G^+$，则从F中去掉此函数依赖。

引理6.2

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

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

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

则以$X-B_i$取代X。

定义6.15 (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。

例题—求F的最小函数依赖集

- R<U, F>，U=ABCD,
- 函数依赖集 F = {A→BD，AB→C，C→D}。
- 求：F最小函数依赖集
- 解法步骤：
  - 1. 将F中的所有函数依赖的右边化为单一属性
  - 2. 去掉F中的所有函数依赖左边的冗余属性
  - 3. 去掉F中所有冗余的函数依赖

例题图解（1）

- （1）将F中的所有函数依赖右边化为单一属性

$$F = \{\boxed{A \rightarrow BD},\ AB \rightarrow C,\ C \rightarrow D\}$$

$$F = \{\boxed{A \rightarrow B},\ \boxed{A \rightarrow D},\ AB \rightarrow C,\ C \rightarrow D\}$$

例题图解（2）

- （2）去掉F中的所有函数依赖左边的冗余属性

$$F = \{A \to B,\ A \to D,\ \mathbf{AB \to C},\ C \to D\}$$

$$A^+ = \{A, B, C, D\} \qquad B^+ = \{B\}$$

$$F = \{A \to B,\ A \to D,\ A \to C,\ C \to D\}$$

例题图解（3）

- （3）去掉F中所有冗余的函数依赖关系

$$F = \{ A \to B,\ \boxed{A \to D,}\ A \to C,\ C \to D \}$$

$$F = \{ A \to B,\ \quad ,\ A \to C,\ C \to D \}$$

$$A^{+} = \{A, B, C, D\}$$

$$Fm = \{ A \to B,\ A \to C,\ C \to D \}$$

极小化过程（续）

F的最小依赖集Fm是否唯一？

[例6.13] $F = \{A\to B，B\to A，B\to C，A\to C，C\to A\}$

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，它与对各函数依赖FDi 及X→A中X各属性的处置的顺序有关

候选码的求解(补充)

- 候选码如何求解？
  - **如果XF+ =U，则X是R<U, F>的候选码**

- 给定的关系R(A1,A2,…,An)和函数依赖集F，可以将其属性分为4类：
  - L类：仅出现在F的函数依赖左边的属性
  - R类：仅出现在F的函数依赖右边的属性
  - N类：在F的函数依赖左右两边均未出现的属性
  - LR类：在F的函数依赖左右两边均出现的属性

候选码的求解(补充)

- **快速求解候选码**
- 定理：对于给定的关系模式R(U,F)，若X（ $X \in R$ ）是**L类属性**，则X必为R的任一候选码的成员
- 定理：对于给定的关系模式R(U,F)，如果X（ $X \in R$ ）是**R类属性**，则X不在任何候选码中
- 定理：设有关系模式R(U,F)， X（ $X \in R$ ）是**N类属性**，则X必包含在R的任一候选码中
  - 推论：对于给定的关系模式R(U,F)，如果X是R的**N类和L类组成的属性集**，且 **X_F^+包含了R的全部属性U，则X是R的唯一候选码**

候选码的求解(补充)

- 例：关系模式R（A,B,C,D,E,P），函数依赖集F={$A\rightarrow D$, $E\rightarrow D$, $D\rightarrow B$, $BC\rightarrow D$, $DC\rightarrow A$}

- 解：
  - 1. F中，C、E均是L类属性，则C、E在R的任何候选码中。
  - 2. P是N类属性，则P在R的任何候选码中
  - 3. (CEP)+=ABCDEP，则CEP是R的候选码

Armstrong公理系统—总结

- 问题引入：判断R<U,F>的好坏的关键是找到部分F，传递F
- 目标需求：找出R<U,F>的全部函数依赖F_R^+
  - 但，直接通过语义找到F很难
  - 因此，建立一套推理系统，给定R<U,F>和已知的F，推导所有F_R^+
- Armstrong公理系统引入
  - 第一步：给出逻辑蕴含F，
  - 第二步：函数依赖闭包 F^+
  - 第三步：Armstrong公理系统的导出规则。证明$F_R^+ = F^+$
  - 第四步：如何方便的求出F^+：求属性集X（X ⊆ U）关于F的闭包X_F^+
  - 第五步：如何计算X_F^+（算法6.1）？如何求R的码？
  - Armstrong公理系统的有效性与完备性
  - 函数依赖集等价的概念
  - 最小依赖集或最小覆盖

第六章 关系数据理论

6.4 模式的分解

- 6.4.1 模式分解的3个定义
- 6.4.2 分解的无损连接性和保持函数依赖性
- 6.4.3 模式分解的算法

6.4 模式的分解

- 关系模式的规范化过程是通过对关系模式的分解来实现的
- 什么是模式分解

（定义6.16）关系模式R<U,F>的一个分解是指：ρ={ R_1<U_1,F_1>，R_2< U_2, F_2 >， …， R_n< U_n, F_n >}， 其中U= $U_1 \cup U_2 \cup .. \cup U_i$，且不存在 $U_i \subseteq U_j$，F_i 为 F在 U_i 上的投影

- 相应地，将 R 存储在二维表 r 中的数据分散到二维表r_1 , r_2 , … ,r_n中去，其中 r_i是 r在属性集 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**

关系模式分解的标准

- 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
- 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

（这三个定义是实行模式分解的三条准则）

例子

例子：S-L(Sno, Sdept, Sloc) 有下列函数依赖：

$$F = \{ Sno \to Sdept,$$

$$Sdept \to Sloc,$$

$$Sno \xrightarrow{\text{传递}} Sloc \}$$

- 已知S-L∈2NF，该关系模式存在非主属性对码的传递函数依赖
- 存在插入异常、删除异常、数据冗余度大和修改复杂的问题。需要分解该关系模式，使成为更高范式的关系模式
- 下面**对4种不同的分解**情况进行分析，引出模式分解的3个定义

例子：第一种分解情况

例：S-L(Sno, Sdept, Sloc) F={Sno→Sdept, Sdept→Sloc, Sno→Sloc}

- 分解方法一. S-L分解为三个关系模式:

  SN(Sno) 、SD(Sdept)、 SO(Sloc)

S-L

Sno	Sdept	Sloc
95001	CS	A
95002	IS	B
95003	MA	C
95004	IS	B
95005	PH	B

分解

SN

Sno
95001
95002
95003
95004
95005

SD

Sdept
CS
IS
MA
PH

SO

Sloc
A
B
C

- 分解后的关系，无法查询95001学生 所在的系 或 所住的宿舍了！
- 这种分解是不可取的。丢失了许多信息，丢失了数据之间联系的信息

例子：第二种分解情况

例：S-L(Sno, Sdept, Sloc) F={Sno→Sdept, Sdept→Sloc, Sno→Sloc}

- 分解方法二： SL分解为下面二个关系模式：

NL(Sno, Sloc)，DL(Sdept, Sloc)

S-L

Sno	Sdept	Sloc
95001	CS	A
95002	IS	B
95003	MA	C
95004	IS	B
95005	PH	B

NL

Sno	Sloc
95001	A
95002	B
95003	C
95004	B
95005	B

DL

Sdept	Sloc
CS	A
IS	B
MA	C
PH	B

例子：第二种分解情况（续）

- 对NL和DL关系 进行自然连接的结果为

ND ⋈ DL

Sno	Sloc	Sloc
95001	A	CS
95002	B	IS
95002	B	PH
95003	C	MA
95004	B	IS
95004	B	PH
95005	B	IS
95005	B	PH

S-L

Sno	Sdept	Sloc
95001	CS	A
95002	IS	B
95003	MA	C
95004	IS	B
95005	PH	B

自然连接比原来的S-L关系多了三个元组，因此，也无法知道原来的S-L关系中究竟有哪些元组，从这个意义上说， 此分解仍然丢失了信息

例子：第三种分解情况

例：S-L(Sno, Sdept, Sloc) F={Sno→Sdept, Sdept→Sloc, Sno→Sloc}

- 分解方法三：将SL分解为下面二个关系模式：
  ND(Sno, Sdept)，NL(Sno, Sloc)

分解后的关系为：

ND

Sno	Sdept
95001	CS
95002	IS
95003	MA
95004	IS
95005	PH

NL

Sno	Sloc
95001	A
95002	B
95003	C
95004	B
95005	B

对ND和NL关系进行自然连接的结果为：

ND ⋈ NL

Sno	Sdept	Sloc
95001	CS	A
95002	IS	B
95003	MA	C
95004	IS	B
95005	PH	B

第三种分解情况没有丢失信息，称为分解“具有无损连接性”

例子：第三种分解情况（续）

- 第3种分解方法具有无损连接性，但是它存在下面的问题
  - 例如，95001学生由CS系转到IS系，ND关系的(95001, CS)元组和NL关系的(95001, A)元组必须同时进行修改，否则会破坏数据库的一致性。

ND

Sno	Sdept
95001	CS
95002	IS
95003	MA
95004	IS
95005	PH

IS

NL

Sno	Sloc
95001	A
95002	B
95003	C
95004	B
95005	B

B

原因：

- S-L中的函数依赖 Sdept→Sloc 既没有投影到 关系模式ND上，也没有投影到关系模式NL上。
- 这种分解没有保持原关系模式中的函数依赖

例子：第四种分解情况

例：S-L(Sno, Sdept, Sloc) F={Sno→Sdept, Sdept→Sloc, Sno→Sloc}

- 分解方法四：　将SL分解为下面二个关系模式：

  ND(Sno, Sdept)，Sno→Sdept

  DL(Sdept, Sloc)，　Sdept→Sloc

这种分解方法就保持了函数依赖，称为"具有保持函数依赖性"

例子：第四种分解情况（续）

继续分析第四种分解情况

ND(Sno, Sdept)， Sno→Sdept

DL(Sdept, Sloc)， Sdept→Sloc

Sno	Sdept
95001	CS
95002	IS
95003	MA
95004	IS
95005	PH

Sdept	Sloc
CS	A
IS	B
MA	C
PH	B

ND ⋈ DL

Sno	Sdept	Sloc
95001	CS	A
95002	IS	B
95003	MA	C
95004	IS	B
95005	PH	B

这种分解不仅“具有保持函数依赖性”，还“具有无损连接性”

例子：分解情况分析

在给出的例子中：

- 第1种分解：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：既不具有无损连接性，也未保持函数依赖
- 第3种分解：具有无损连接性，但未保持函数依赖
- 第4种分解：既具有无损连接性，又保持了函数依赖

6.4 模式的分解

- 6.4.1 模式分解的3个定义
- 6.4.2 分解的无损连接性和保持函数依赖性
- 6.4.3 模式分解的算法

1. 具有无损连接性的模式分解

[定义6.18]： $\rho=\{R_1, \ldots, R_n\}$是 R的一个分解，

- 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>}

若对 R的任何一个关系 r均有 **r = r在ρ中各关系模式上投影的自然连接**成立，则称分解ρ具有无损连接性。简称ρ为无损分解。

- **只有具有无损连接性的分解才能够保证不丢失信息。**
- **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。**

*** 算法6.3 判别一个分解的无损连接性。 p.192**

2. 保持函数依赖的模式分解

[定义6.17]： ρ={R_1, …, R_n}是 R的一个分解，

- 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F所逻辑蕴含的函数依赖一定也为分解后所有的关系模式中的函数 依赖 F_i 所逻辑蕴含，即 $F^+ = (F_1 \cup F_2 \cup \ldots \cup F_n)^+$，则称关系模式R的这个分解是保持函数依赖的（Preserve dependency）

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

引理6.3 给出了判别R的分解是否保持函数依赖的方法。 P.188

3. 分解的无损连接性和保持函数依赖性

- 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。
  - 保持函数依赖的分解也不一定具有无损连接性

6.4 模式的分解

- 6.4.1 模式分解的3个定义
- 6.4.2 分解的无损连接性和保持函数依赖性
- 6.4.3 模式分解的算法

6.4.3 模式分解的算法

- 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

6.4.3 模式分解的算法

- 按照不同的分解算法，关系模式所能达到的范式是不相同的
  - 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
  - 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

无损连接	保持依赖	达到的范式
	Y	3NF，不一定BCMF
Y	Y	3NF，不一定BCMF
Y		4NF

第六章 关系数据理论

6.1 问题的提出

6.2 规范化

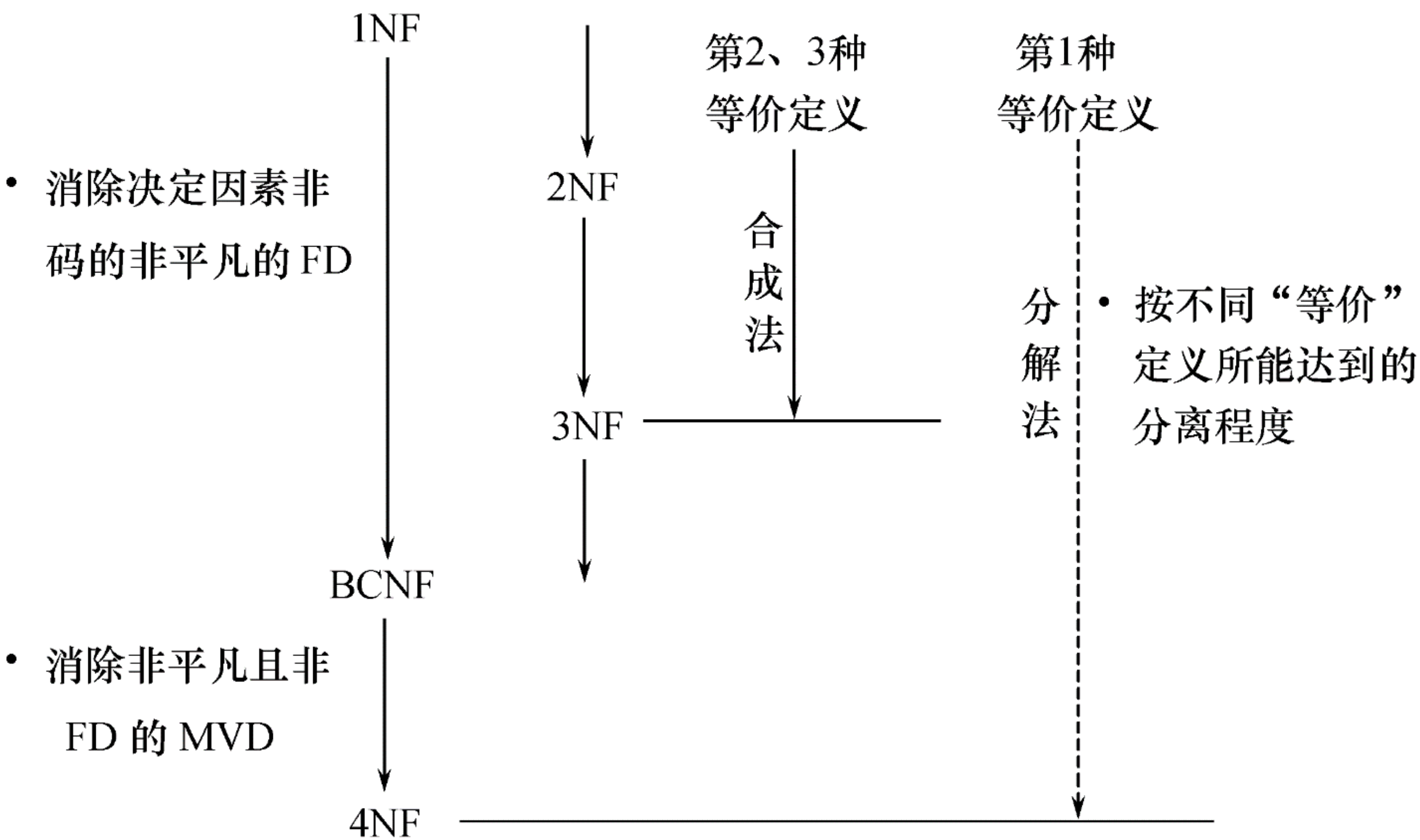
6.3 数据依赖的公理系统

*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：



规范化小结（续）

- 关系模式规范化的基本步骤

消除一些非平凡函数依赖，这些非平凡函数依赖的决定属性集不是码

	1NF
	↓ 消除非主属性对码的部分函数依赖
消除决定属性	2NF
集非码的非平	↓ 消除非主属性对码的传递函数依赖
凡函数依赖	3NF
	↓ 消除主属性对码的部分和传递函数依赖
	BCNF
	↓ 消除非平凡且非函数依赖的多值依赖
	4NF

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