北京林业大学

数据库原理与应用

北京林业大学 信息学院

函数依赖

函数依赖的定义

关系模式中的各属性之间相互依赖、相互制约的联系称为数据依赖。

函数依赖 **多值依赖**

函数依赖（FD，Functional Dependency）是关系模式中属性之间的一种逻辑依赖关系。

SCD (SNo，SN，Age，Dept，MN，CNo，Score)

SNo ▶ **一个学生** ▶ **SN，Age，Dept**

SNo决定函数（SN，Age，Dept）

（SN，Age，Dept）函数依赖于SNo

函数依赖的定义

定义

设关系模式$R(U，F)$，U是属性全集，F是U上的函数依赖所构成的集合，X和Y是U的子集，如果对于$R(U)$的任意一个可能的关系r，对于X的每一个具体值，Y都有唯一的具体值与之对应，则称X决定函数Y，或Y函数依赖于X，记作$X \rightarrow Y$。我们称X为决定因素，Y为依赖因素。当Y不函数依赖于X时，记作：$X \not\rightarrow Y$。当$X \rightarrow Y$且$Y \rightarrow X$时，则记作：$X \leftrightarrow Y$。

函数依赖的定义

U={SNo，SN，Age，Dept，MN，CNo，Score}

F={SNo→SN，SNo→Age，SNo→Dept，

(SNo，CNo)→Score}

Sno↛Score

Sno↛CNo

函数依赖的逻辑蕴涵定义

定义

设F是在关系模式$R(U)$上成立的函数依赖集合，X，Y是属性集U的子集，$X\rightarrow Y$是一个函数依赖。如果从F中能够推导出$X\rightarrow Y$，即如果对于R的每个满足F的关系r也满足$X\rightarrow Y$，则称$X\rightarrow Y$为F的逻辑蕴涵（或F逻辑蕴涵$X\rightarrow Y$），记为$F|=X\rightarrow Y$。

闭包（Closure）的定义

定义

设F 是函数依赖集，被F 逻辑蕴涵的函数依赖的全体构成的集合，称为函数依赖集F 的闭包（Closure），记为F^+。即：

$F^+=\{ X\rightarrow Y \mid F \models X\rightarrow Y\}$

函数依赖的推理规则及正确性

设有关系模式R（U），U是关系模式R 的属性集，F 是R上成立的只涉及U 中属性的函数依赖集。X，Y，Z，W 均是U 的子集，r是R 的一个实例。

函数依赖的推理规则及正确性

Armstrong 公理及正确性

A1：自反律(Reflexivity)

如果$Y \subseteq X \subseteq U$，则$X \rightarrow Y$在$R$上成立。

即一组属性函数决定它的所有子集。

如：（SNo，CNo）$\rightarrow SNo$

函数依赖的推理规则及正确性

A2：增广律(Augmentation)

若$X \rightarrow Y$在R上成立，且$Z \subseteq U$，则$XZ \rightarrow YZ$在R上也成立。

如：$SNo \rightarrow Age$，$(Sno，SN) \rightarrow (Age，SN)$

A3：传递律(Transitivity)

若$X \rightarrow Y$和$Y \rightarrow Z$在R上成立，则$X \rightarrow Z$在R上也成立。

如：$SNo \rightarrow Dept$，$Dept \rightarrow MN$，$SNo \rightarrow MN$

函数依赖的推理规则及正确性

定理：如果$X \rightarrow Y$是从F用Armstrong公理推理导出，那么$X \rightarrow Y$在F^+中。

函数依赖的推理规则及正确性

Armstrong 公理推论及正确性

- 合并律（Union rule）

若$X \rightarrow Y$和$X \rightarrow Z$在R上成立，则$X \rightarrow YZ$在R上也成立。

函数依赖的推理规则及正确性

- 伪传递律（Pseudotransitivity rule）

若$X \rightarrow Y$和$YW \rightarrow Z$在R上成立，则$XW \rightarrow Z$在R上也成立。

- 分解律（Decomposition rule）

若$X \rightarrow Y$和$Z \subseteq Y$在R上成立，则$X \rightarrow Z$在R上也成立。

函数依赖的推理规则及正确性

定理：如果$A_1A_2\ldots A_n$是关系模式R的属性集，那么$X \rightarrow A_1A_2\ldots A_n$成立的充分必要条件是$X \rightarrow A_i$（$i=1,2,\ldots n$）成立。

- 复合律（Composition）

若$X \rightarrow Y$和$W \rightarrow Z$在R上成立，则$XW \rightarrow YZ$在R上也成立

函数依赖推理规则的完备性

正确性：
从函数依赖集F使用推理规则推出的函数依赖必定在F^+中

完备性：
F^+中的函数依赖都能从F集使用推理规则集推出

完全函数依赖与部分函数依赖

定义

设有关系模式$R(U)$，U是属性全集，X和Y是U的子集

◆ 如果$X \to Y$，并且对于X的任何一个真子集X'，都有$X' \not\to Y$，则称Y对X完全函数依赖，记作$X \xrightarrow{f} Y$。

◆ 如果对X的某个真子集X'，有$X' \to Y$，则称Y对X部分函数依赖，记作$X \xrightarrow{p} Y$。

◆ 关系模式SCD中，因为SNo $\not\to$Score，且CNo$\not\to$Score，所以有：(SNo，CNo) $\xrightarrow{f}$Score。而SNo$\to$Age，所以(SNo，CNo) $\xrightarrow{p}$Age 。

完全函数依赖与部分函数依赖

只有当决定因素是组合属性时，讨论部分函数依赖才有意义；

当决定因素是单属性时，只能是完全函数依赖。

传递函数依赖

定义

设有关系模式R(U)，U是属性全集，X，Y，Z是U的子集

若$X \to Y$，但$Y \not\to X$，而$Y \to Z$（$Y \notin X$，$Z \notin Y$），则称Z对X传递函数依赖，记作：$X \xrightarrow{t} Z$。

如果$Y \to X$，则$X \leftrightarrow Y$，这时称Z对X直接函数依赖，而不是传递函数依赖。

传递函数依赖

例如，在关系模式SCD中，SNo $\longrightarrow$ Dept ，但Dept $\not\longrightarrow$ SNo，而Dept $\longrightarrow$ MN，则有SNo $\xrightarrow{t}$ MN

当学生不存在重名的情况下，有SNo $\longrightarrow$ SN，SN $\longrightarrow$ SNo，SNo $\longleftrightarrow$ SN，SN $\longrightarrow$ Dept，这时Dept对SNo是直接函数依赖，而不是传递函数依赖。

属性集的闭包及其算法

定义

设有关系模式$R(U)$，U是属性集，F是R上的函数依赖集，X是U的子集（$X \subseteq U$），用函数依赖推理规则可从F推出函数依赖$X \rightarrow A$中所有A的集合，称为属性集X关于F的闭包，记为X^+。

$X^+ = \{\text{属性A} | X \rightarrow A \text{在} F^+ \text{中}\}$

属性集的闭包及其算法

定理

$X \rightarrow Y$能用函数依赖推理规则推出的充分必要条件是$Y \subseteq X^{+}$中

属性集的闭包及其算法

设有关系模式$R(U)$，U是属性集，F是R上的函数依赖集，X是U的子集（$X \subseteq U$）

算法

result=X
do
{
if F中有某个函数依赖$Y \rightarrow Z$满足
$Y \subseteq$ result
then result=result $\cup$ Z
}
while (result有所改变);

属性集的闭包及其算法

例子：$U=\{XYZW\}$，$F=\{X \rightarrow Y$，$Y \rightarrow Z$，$W \rightarrow Y\}$，计算X^+，$(XW)^+$和$(YW)^+$

- 根据$X \rightarrow Y$，$X^+=\{XY\}$；

  根据$Y \subseteq \{XY\}$，$Y \rightarrow Z$，$X^+=\{XYZ\}$

- 根据$X \rightarrow Y$，$W \rightarrow Y$，$(XW)^+=\{XWY\}$；

  根据$Y \subseteq \{XWY\}$，且$Y \rightarrow Z$，$(XW)^+=\{XWYZ\}$

- 根据$Y \rightarrow Z$和$W \rightarrow Y$，$(YW)^+=\{WYZ\}$；

候选码的求解理论和算法

候选码的定义

◆ 设关系模式R的属性集是U，X是U的一个子集，F是在R上成立的函数依赖集。

◆ 如果X→U在R上成立（即X→U在F^+中），那么称X是R的一个超码。

◆ 如果X→U在R上成立，但对X的任一真子集X′都有X′→U不成立（即X′→U不在F^+中，或者$X \xrightarrow{f} U$），那么称X是R上的一个候选码。

候选码的求解理论和算法

快速求解候选码的一个充分条件

对于给定的关系模式R(A_1…，A_n)和函数依赖集F，可将其属性分为以下四类：

- L类
- R类
- N类
- LR类

候选码的求解理论和算法

定理

(1) 若X（X∈R）是L类属性，则X必为R的任一候选码的成员。

(2) 若X（X∈R）是L类属性，且X^+包含了R的全部属性，则X必为R的唯一候选码。

(3) 若X（X∈R）是R类属性，则X不在任何候选码中。

候选码的求解理论和算法

(4) 若X（X∈R）是N类属性，则X必为R的任一候选码的成员。

(5) 若X（X∈R）是R的N类和L类属性组成的属性集，且X^+包含了R的全部属性，则X是R的唯一候选码。

(6) 若X（X∈R）是LR类属性，则X可能为R的任一候选码的成员，也可能不为R的任一候选码的成员。

例：设有关系模式R（A,B,C,D）与它的函数依赖集F={D→B，B→D，AD→B，AC→D}，求R的所有候选码。

候选码的求解理论和算法

多属性函数依赖集候选码的求解算法

设有关系模式R，F是R上的函数依赖集，求R的所有候选码

输入：关系模式R及其函数依赖集F

输出：关系模式R的所有候选码

(1) 属性分类（L、R、N和LR），X代表L 类和N类属性，Y代表LR类属性。

(2) 若X^{+}包含了R的全部属性，转（5）；否则，转（3）。

候选码的求解理论和算法

(3) 在Y中取一个属性A，求$(XA)^+$，若它包含了R的全部属性，则转（4）；否则，调换一属性反复进行这一过程，直到试完所有Y中的属性。

(4) 如果已找出所有候选码，则转（5）；否则在Y中依次取两个属性、三个属性、…，求它们的属性集的闭包，直到其闭包包含R的全部属性。

(5) 停止，输出结果。

候选码的求解理论和算法

例：设有关系模式R（A,B,C,D,E）与它的函数依赖集$F=\{A \rightarrow BC,\ CD \rightarrow E,\ B \rightarrow D,\ E \rightarrow A\}$，求$R$的所有候选码。

- 属性A、B、C、D、E都是LR类属性
- 依次取一个属性求解属性集闭包

$A^+=ABCDE$，$B^+=BD$，$C^+=C$，$D^+=D$，

$E^+=ABCDE$

- 取出两个属性计算属性集闭包

$(BC)^+=ABCDE$，$(CD)^+=ABCDE$，$(BD)^+=BD$

- R的候选码为A、E、BC和CD

函数依赖集的等价、覆盖和最小函数依赖集

定义

关系模式$R(U)$的两个函数依赖集F和G，如果满足$F^+ = G^+$，则称F和G是等价的函数依赖集。记作：$F \equiv G$。如果F和G等价，就说F覆盖G，或G覆盖F。

函数依赖集的等价、覆盖和最小函数依赖集

定义

设F是属性集U上的函数依赖集，$X \rightarrow Y$是F中的函数依赖。

函数依赖中无关属性、无关函数依赖的定义如下：

（1）如果$A \in X$，且F逻辑蕴涵$(F-\{X \rightarrow Y\}) \cup \{(X-A) \rightarrow Y\}$，则称属性$A$是$X \rightarrow Y$左部的无关属性。

（2）如果$A \in X$，且$(F-\{X \rightarrow Y\}) \cup \{X \rightarrow (Y-A)\}$逻辑蕴涵$F$，则称属性$A$是$X \rightarrow Y$右部的无关属性。

（3）如果$X \rightarrow Y$的左右两边的属性都是无关属性，则函数依赖$X \rightarrow Y$称为无关函数依赖。

函数依赖集的等价、覆盖和最小函数依赖集

定义

设F是属性集U上的函数依赖集。如果F_{min}是F的一个最小函数依赖集，那么F_{min}应满足下列四个条件：

(1) $F_{min}^{+}=F^{+}$；

(2) 每个函数依赖的右边都是单属性；

函数依赖集的等价、覆盖和最小函数依赖集

(3) F_{min}中没有冗余的函数依赖（即在F_{min}中不存在这样的函数依赖$X \rightarrow Y$，使得F_{min}与F_{min}-$\{X \rightarrow Y\}$等价），即减少任何一个函数依赖都将与原来的F不等价；

(4) 每个函数依赖的左边没有冗余的属性（即F_{min}中不存在这样的函数依赖$X \rightarrow Y$，X有真子集W使得F_{min}-$\{X \rightarrow Y\} \cup \{W \rightarrow Y\}$与$F_{min}$等价），减少任何一个函数依赖左部的属性后，都将与原来的F不等价。

函数依赖集的等价、覆盖和最小函数依赖集

[例] 设有函数依赖集F_1，F_2，判断是否为最小函数依赖集

(1) $F_1=\{AB\rightarrow CD，BE\rightarrow C，C\rightarrow G\}$

(2) $F_2=\{A\rightarrow D，B\rightarrow A，A\rightarrow C，B\rightarrow D，D\rightarrow C\}$

函数依赖集的等价、覆盖和最小函数依赖集

算法——计算函数依赖集F的最小函数依赖集G

(1) 对F中的任一函数依赖$X\rightarrow Y$，如果$Y=Y_1,Y_2,\cdots,Y_k$（k≥2）多于一个属性，就用分解律，分解为$X\rightarrow Y_1$，$X\rightarrow Y_2$，…，$X\rightarrow Y_k$，替换$X\rightarrow Y$，得到一个与F等价的函数依赖集G，G中每个函数依赖的右边均为单属性。

(2) 去掉G中各函数依赖左部多余的属性。即一个一个检查G左边是非单属性的依赖。如，$XY\rightarrow A$，现在要判断Y是否为多余的，则以$X\rightarrow A$代替$XY\rightarrow A$是否等价？只要在G中求X^+，若X^+包含A，则说明$X\rightarrow A$可以代替$XY\rightarrow A$，即Y是多余的属性；否则，Y不是多余的属性。

函数依赖集的等价、覆盖和最小函数依赖集

算法——计算函数依赖集F的最小函数依赖集G

(3) 在G中消除冗余的函数依赖。具体做法是：从第一个函数依赖开始，在G中去掉它（假设该函数依赖是$X \rightarrow Y$），然后在剩下的函数依赖中求X^+，看X^+是否包含Y，若是，则去掉$X \rightarrow Y$；若不包含Y，则不能去掉$X \rightarrow Y$。依次进行下去。

函数依赖集的等价、覆盖和最小函数依赖集

[例] 设F是关系模式$R(A，B，C)$的函数依赖集，$F=\{A\rightarrow BC，B\rightarrow C，A\rightarrow B，AB\rightarrow C\}$，求其最小函数依赖集$F_{min}$

(1) 右边单属性

将F中每个函数依赖的右部变成单属性。$F=\{A\rightarrow B，A\rightarrow C，B\rightarrow C，AB\rightarrow C\}$

函数依赖集的等价、覆盖和最小函数依赖集

[例] 设F是关系模式$R(A，B，C)$的函数依赖集，$F=\{A\rightarrow BC，B\rightarrow C，A\rightarrow B，AB\rightarrow C\}$，求其最小函数依赖集$F_{min}$

(2) 左部多余属性

在$AB\rightarrow C$，验证属性B是不是多余属性。计算$A^+=(ABC)$，A^+包含属性C，因此，B是左部多余的属性可以去掉。$AB\rightarrow C$简化为$A\rightarrow C$，$F=\{A\rightarrow B，A\rightarrow C，B\rightarrow C\}$

函数依赖集的等价、覆盖和最小函数依赖集

[例] 设F是关系模式$R(A，B，C)$的函数依赖集，$F=\{A\rightarrow BC，B\rightarrow C，A\rightarrow B，AB\rightarrow C\}$，求其最小函数依赖集$F_{min}$

(3) 冗余函数依赖

$A\rightarrow C$可以通过$A\rightarrow B$和$B\rightarrow C$推出，
因此可以去掉$A\rightarrow C$。

所以$F_{min}=\{A\rightarrow B，B\rightarrow C\}$