



第六章 关系数据理论（续）



知识回顾

1、关系模式的形式化表示

关系模式由五部分组成，即它是一个五元组：

$R(U, D, DOM, F)$

R: 关系名

U: 组成该关系的属性名集合

D: 属性组**U**中属性所来自的域

DOM: 属性向域的映象集合

F: 属性间数据的依赖关系集合



- 2、关系模式的简化表示

简化为一个三元组： $R(U, F)$



3、数据依赖的类型

- **函数依赖** (Functional Dependency, 简记为FD)
- **多值依赖** (Multivalued Dependency, 简记为MVD)



一、函数依赖

二、平凡函数依赖与非平凡函数依赖

三、完全函数依赖与部分函数依赖

四、传递函数依赖

五、平凡多值依赖和非平凡的多值依赖



□ 4、关系模式规范化的基本步骤

1NF

↓ 消除**非主属性**对码的部分函数依赖

消除决定属性
集非码的非平
凡函数依赖

2NF

↓ 消除**非主属性**对码的传递函数依赖

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**



第六章 关系数据理论

6.1 数据依赖

6.2 规范化

6.3 数据依赖的公理系统

6.4 模式的分解



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 $R \langle U, F \rangle$, 其任何一个关系 r , 若函数依赖 $X \rightarrow Y$ 都成立, 则称 F 逻辑蕴含 $X \rightarrow Y$



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含的函数依赖



1、Armstrong公理系统

关系模式 $R \langle U, F \rangle$ 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。

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



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 $X \rightarrow YZ$ 。

伪传递规则：

由 $X \rightarrow Y$ ， $WY \rightarrow Z$ ， 有 $XW \rightarrow Z$ 。

分解规则：

由 $X \rightarrow Y$ 及 $Z \subseteq Y$ ， 有 $X \rightarrow Z$ 。



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



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， $X \subseteq U$ ，

$X_F^+ = \{ A | X \rightarrow A \text{ 能由 } F \text{ 根据Armstrong公理导出} \}$ ， 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

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) $X^{(i+1)} = B \cup X^{(i)}$



算法6.1

(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。



函数依赖闭包

[例1] 已知关系模式 $R\langle U, F \rangle$ ，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) 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
：

$$AB \rightarrow C, B \rightarrow D.$$

于是 $X^{(1)} = AB \cup CD = ABCD$ 。



函数依赖闭包

(2) 因为 $X^{(0)} \neq X^{(1)}$ ，所以再找出左部为 $ABCD$ 子集的那些函数依赖，又得到 $AB \rightarrow C$, $B \rightarrow D$, $C \rightarrow E$, $AC \rightarrow B$,
于是 $X^{(2)} = X^{(1)} \cup BCDE = ABCDE$ 。

(3) 因为 $X^{(2)} = U$ ，算法终止

所以 $(AB)_F^+ = ABCDE$ 。



练习：已知关系模式 $R(U, F)$ ， $U = \{A, B, C, D, E\}$ ， $F = \{A \rightarrow B, D \rightarrow C, BC \rightarrow E, AC \rightarrow B\}$ 。求 $(AE)_{F^+}$ ， $(AD)_{F^+}$

(1) $X^{(0)} = AE$ ，得到 A ， E 或 AE 的函数依赖，只有一个 $A \rightarrow B$

(2) $X^{(1)} = AE \cup B$ ，判断 $X^{(0)}$ 不等于 $X^{(1)}$ 。

(3) $X^{(1)} = AEB$ ，得到函数依赖， $A \rightarrow B$

(4) $X^{(2)} = AEUB$ ，判断 $X^{(2)}$ 等于 $X^{(1)}$

那么 $(AE)_{F^+} = ABE$

$(AD)_{F^+} = ABCDE$



练习：

1. 关系R (A, B, C, D, E) 满足下列函数依赖： $F = \{A \rightarrow C, C \rightarrow D, B \rightarrow C, DE \rightarrow C, CE \rightarrow A\}$ ，候选关键字？ **BE**
2. 在上述关系模式中，求 $(BE)_F^+$



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。



函数依赖集等价的充要条件

引理6.3 $F^+ = G^+$ 的充分必要条件是

$$F \subseteq G^+, \text{ 和 } G \subseteq F^+$$



函数依赖集等价

- 要判定 $F \subseteq G^+$ ，只须逐一对 F 中的函数依赖 $X \rightarrow Y$ ，考察 Y 是否属于 X_{G^+} 就行了。因此引理6.3 给出了判断两个函数依赖集等价的可行算法。



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



最小依赖集

[例2] 对于6.1节中的关系模式 $S\langle U, F\rangle$ ，其中：

$U=\{ \text{SNO}, \text{SDEPT}, \text{MN}, \text{CNAME}, \text{G} \}$,

$F=\{ \text{SNO} \rightarrow \text{SDEPT}, \text{SDEPT} \rightarrow \text{MN},$
 $(\text{SNO}, \text{CNAME}) \rightarrow \text{G} \}$

设 $F'=\{ \text{SNO} \rightarrow \text{SDEPT}, \text{SNO} \rightarrow \text{MN},$
 $\text{SDEPT} \rightarrow \text{MN}, (\text{SNO}, \text{CNAME}) \rightarrow \text{G},$
 $(\text{SNO}, \text{SDEPT}) \rightarrow \text{SDEPT} \}$

F 是最小覆盖，而 F' 不是。

因为： $F' - \{ \text{SNO} \rightarrow \text{MN} \}$ 与 F' 等价

$F' - \{ (\text{SNO}, \text{SDEPT}) \rightarrow \text{SDEPT} \}$ 也与 F' 等价

$F' - \{ (\text{SNO}, \text{SDEPT}) \rightarrow \text{SDEPT} \}$

$\cup \{ \text{SNO} \rightarrow \text{SDEPT} \}$ 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:构造性证明, 依据定义分三步对 F 进行“极小化处理”, 找出 F 的一个最小依赖集。

(1)逐一检查 F 中各函数依赖 $FD_i: X \rightarrow Y$,

若 $Y = A_1 A_2 \dots A_k$, $k > 2$,

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



极小化过程

(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。

由于 F 与 $G = F - \{X \rightarrow A\}$ 等价的充要条件是

$A \in X_G^+$



极小化过程

(3) 逐一取出 F 中各函数依赖 $FD_i: X \rightarrow A$,
设 $X = B_1 B_2 \dots B_m$,
逐一考查 B_i ($i=1, 2, \dots, m$),
若 $A \in (X - B_i)_F^+$,
则以 $X - B_i$ 取代 X 。



极小化过程

由定义，最后剩下的 F 就一定是极小依赖集。

因为对 F 的每一次“改造”都保证了改造前后的两个函数依赖集等价，因此剩下的 F 与原来的 F 等价。

□ 这也是求 F 极小依赖集的过程。



极小化过程

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

求 F 的最小依赖集。

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

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

F_{m1} 、 F_{m2} 都是 F 的最小依赖集

- F 的最小依赖集 F_m 不一定是唯一的它与对各函数依赖 FD_i 及 $X \rightarrow A$ 中 X 各属性的处置顺序有关



极小化过程

- 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
 - ◆ 若改造后的 F 与原来的 F 相同, 说明 F 本身就是一个最小依赖集



第六章 关系数据理论

6.1 数据依赖

6.2 规范化

6.3 数据依赖的公理系统

6.4 模式的分解



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并不是唯一的
- 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

三种模式分解的等价定义

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 $R\langle U, F \rangle$ 的一个分解：

$$\rho = \{ R_1\langle U_1, F_1 \rangle, R_2\langle U_2, F_2 \rangle, \dots, R_n\langle U_n, F_n \rangle \}$$

$U = U_1 \cup U_2 \cup \dots \cup U_n$ ，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: **SL (Sno, Sdept, Sloc)**

$F = \{ Sno \rightarrow Sdept, Sdept \rightarrow Sloc, Sno \rightarrow Sloc \}$

$SL \in 2NF$

存在插入异常、删除异常、冗余度大和修改复杂等问题

分解方法可以有多种



模式的分解（续）

SL

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



模式的分解（续）

1. **SL**分解为下面三个关系模式：

SN(Sno)

SD(Sdept)

SO(Sloc)





分解后的关系为：

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



模式的分解（续）

分解后的数据库**丢失了许多信息**

例如无法查询95001学生所在系或所在宿舍。

如果分解后的关系可以通过自然连接恢复为原来的关系，那么这种分解就没有**丢失信息**



2. SL分解为下面二个关系模式:

NL(Sno, Sloc)

DL(Sdept, Sloc)

分解后的关系为:

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

DL	
Sdept	Sloc
CS	A
IS	B
MA	C
PH	B





NL \bowtie DL

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



模式的分解（续）

NL \bowtie DL比原来的SL关系多了3个元组

无法知道95002、95004、95005

究竟是哪个系的学生

元组增加了，信息丢失了



第三种分解方法

3. 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:





模式的分解（续）

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



模式的分解（续）

ND \bowtie NL

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

与SL关系一样，因此没有丢失信息



具有无损连接性的模式分解

- 关系模式 $R\langle U, F \rangle$ 的一个分解 $\rho = \{ R_1\langle U_1, F_1 \rangle, R_2\langle U_2, F_2 \rangle, \dots, R_n\langle U_n, F_n \rangle \}$

若 R 与 R_1 、 R_2 、...、 R_n 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）

第三种分解方法具有无损连接性

问题：

这种分解方法没有保持原关系中的函数依赖

SL中的函数依赖 $Sdept \rightarrow Sloc$

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 $R\langle U, F \rangle$ 被分解为若干个关系模式

$$R_1\langle U_1, F_1 \rangle, R_2\langle U_2, F_2 \rangle, \dots, R_n\langle U_n, F_n \rangle$$

（其中 $U = U_1 \cup U_2 \cup \dots \cup U_n$ ，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模式分解（Preserve dependency）。



第四种分解方法

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）

第一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

第二种分解方法既不具有无损连接性，也未保持函数依赖，它也不是原关系模式的一个等价分解

第三种分解方法具有无损连接性，但未保持函数依赖

第四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 转换为BCNF的无损连接分解（分解法）
- ❑ 算法6.6 达到4NF的具有无损连接性的分解



分解算法

- 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

- 在这一章，我们首先由关系模式的存储异常问题引出了函数依赖的概念，其中包括完全函数依赖、部分函数依赖和传递函数依赖，这些概念是规范化理论的依据和规范化程度的准则。
- 在规范化过程中，逐渐消除存储异常，使数据冗余尽量小，便于插入、删除和更新。



- 规范化的基本原则就是遵从概念单一化“**一事一地**”的原则，即一个关系只描述一个实体或者实体间的联系。
- 规范化的投影分解方法不是唯一的，对于3NF的规范化，分解既要具有无损连接性，又要具有函数依赖保持性。



练习:

设有关系模式R(课程号, 教师姓名, 学号, 学生姓名, 成绩), 规定: 每一门课由一名教师讲授, 每个学生每门课只有一个成绩, 学生的学号决定学生的姓名。请回答下列问题:

- 1) 请列出该关系模式中包含的所有函数依赖;
- 2) 指出该关系模式的候选码;
- 3) 关系模式R符合第几范式?说明理由
- 4) 将R分解符合第三范式, 并说明理由
- 5) R分解后是否满足BCNF



1) 函数依赖

学号 \rightarrow 学生姓名, (学号, 课程号) \rightarrow 成绩, 课程号 \rightarrow 教师姓名

2) 候选码: (学号, 课程号)

3) 确定范式:

\therefore 课程号 \rightarrow 教师姓名

\therefore (学号, 课程号) 与教师姓名 部分函数依赖

\therefore R不属于2NF, R属于1NF

同理 \therefore 学号 \rightarrow 学生姓名

\therefore (学号, 课程号) 与 学生姓名部分函数依赖

\therefore R不属于2NF, R属于1NF



把R分解为如下三个关系模式：

R1 (学号, 课程号, 成绩)

R2 (课程号, 教师姓名)

R3 (学号, 学生姓名)

∵ R1的候选码为：(学号, 课程号)，非主属性：成绩
又∵ (学号, 课程号) → 成绩 ∴ R1属于2NF

∵ R1不存在非主属性对码的传递函数依赖， ∴ R1属于3NF

∵ R2的候选码为：课程号，非主属性：教师姓名
又∵ 课程号 → 教师姓名 ∴ R2属于2NF

∵ R2不存在非主属性对码的传递函数依赖， ∴ R2属于3NF



∴ R3的候选码为：学号，非主属性：学生姓名

又 ∴ 学号 \rightarrow 学生姓名 \therefore R3属于2NF

∴ R3不存在非主属性对码的传递函数依赖， \therefore R3属于3NF

5) 全都属于BCNF。