

数据库系统概论

An Introduction to Database Systems

关系数据库规范化理论(三)

The Relational Database Normalization Theory

陆嘉恒

www.jiahenglu.net

中国人民大学信息学院

复习：是否符合第三范式？

例：在关系模式STJ (S, T, J) 中，S表示学生，T表示教师，J表示课程。

- 函数依赖：

假设每一教师只教一门课。每门课由若干教师教，某一学生选定某门课，就确定了一个固定的教师。于是有：

$$(S, J) \rightarrow T, (S, T) \rightarrow J, T \rightarrow J$$

5.2.3 第三范式 (3NF)

- 3NF的定义

定义5.9 关系模式 $R\langle U, F \rangle$ 中若不存在这样的码 X , 属性组 Y 及非主属性 Z ($Z \subsetneq Y$) 使得 $X \rightarrow Y$, $Y \not\rightarrow X$, $Y \rightarrow Z$, 成立, 则称 $R\langle U, F \rangle \in 3NF$ 。

是否符合第三范式? **Yes**

例：在关系模式STJ (S, T, J) 中，S表示学生，T表示教师，J表示课程。

- 函数依赖：

假设每一教师只教一门课。每门课由若干教师教，某一学生选定某门课，就确定了一个固定的教师。于是有：

$$(S, J) \rightarrow T, (S, T) \rightarrow J, T \rightarrow J$$

思考：

- 关系R中没有非主属性
- R 是 2 NF?
- R 是 3 NF ?
- R是 4 NF?

思考：

- 关系R中只有两个属性
- R 是 2 NF?
- R 是 3 NF ?
- R是 4 NF?

Question?

- **Welcome question**

5.2.6 规范化

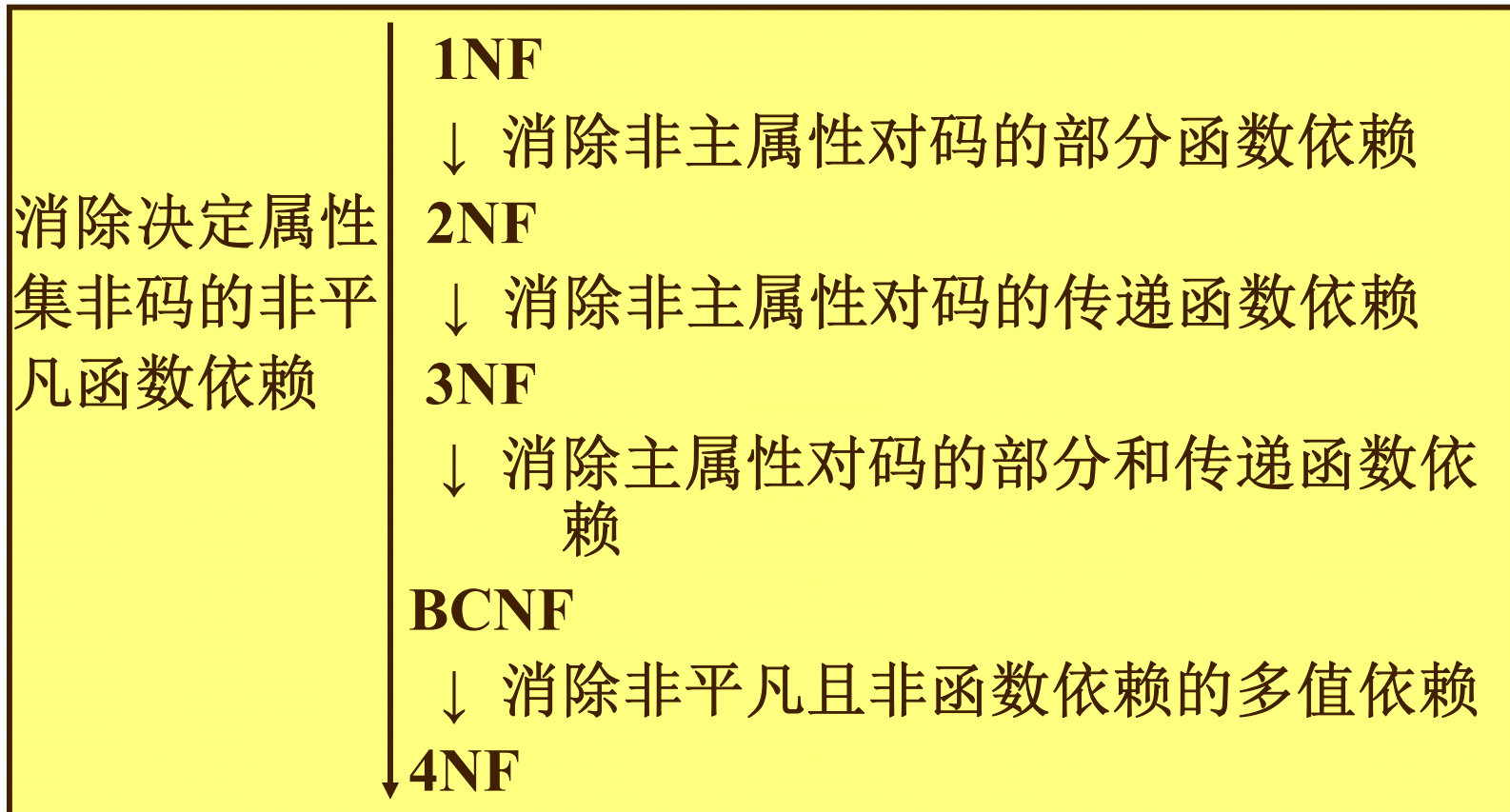
- 关系数据库的规范化理论是数据库逻辑设计的工具。
- 一个关系只要其分量都是不可分的数据项，它就是规范化的关系，但这只是最基本的规范化。
- 规范化程度可以有6个不同的级别，即6个范式。

规范化（续）

- 规范化程度过低的关系不一定能够很好地描述现实世界，可能会存在插入异常、删除异常、修改复杂、数据冗余等问题，解决方法就是对其进行规范化，转换成高级范式。
- 一个低一级范式的关系模式，通过模式分解可以转换为若干个高一级范式的关系模式集合，这种过程就叫关系模式的规范化。

规范化（续）

- 关系模式规范化的基本步骤



1NF

2NF

3NF

BCNF

规范化（续）

- 规范化的基本思想是逐步消除数据依赖中不合适的部分，使模式中的各关系模式达到某种程度的“分离”，即采用“一事一地”的模式设计原则，让一个关系描述一个概念、一个实体或者实体间的一种联系。若多于一个概念就把它“分离”出去。因此所谓规范化实质上是概念的单一化。

规范化（续）

- 不能说规范化程度越高的关系模式就越好。在设计数据库模式结构时，必须对现实世界的实际情况和用户应用需求作进一步分析，确定一个合适的、能够反映现实世界的模式。这也就是说，上面的规范化步骤可以在其中任何一步终止。

模式的分解 与规范化（续）

例：对于关系模式SL（Sno， Sdept， Sloc），
SL中有下列函数依赖：

Sno→**Sdept**

Sdept→**Sloc**

Sno→**Sloc**

思考： SL属于第几范式？

模式的分解 与规范化（续）

例：对于关系模式SL（Sno， Sdept， Sloc），
SL中有下列函数依赖：

Sno→Sdept

Sdept→Sloc

Sno→Sloc

已知SL∈2NF，该关系模式存在插入异常、删除异常、数据冗余度大和修改复杂的问题。

因此需要分解该关系模式，使成为更高范式的关系模式。分解方法可以有很多种。

模式的分解 与规范化（续）

假设下面是该关系模式的一个关系：

SL

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

模式的分解 与规范化（续）

– 第一种分解方法

将SL分解为下面三个关系模式：

SN(Sno)

SD(Sdept)

SO(Sloc)

模式的分解 与规范化（续）

分解后的关系为：

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

模式的分解 与规范化（续）

SN、SD和SO都是规范化程度很高的关系模式（**5NF**）。但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。因此这种分解方法是不可取的。

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。

模式的分解 与规范化（续）

– 第二种分解方法

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

模式的分解 与规范化 (续)

对NL和DL关系进行自然连接的结果为:

NL ⋈ DL

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

模式的分解 与规范化（续）

NL \bowtie **DL**比原来的**SL**关系多了两个元组 **(95002, B, PH)**和**(95005, B, IS)**。因此我们也无法知道原来的**SL**关系中究竟有哪些元组，从这个意义上说，此分解方法仍然丢失了信息。

模式的分解 与规范化（续）

– 第三种分解方法

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：

模式的分解与规范化 (续)

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

模式的分解 与规范化（续）

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	CS	A
95005	PH	B

它与SL关系完全一样，因此第三种分解方法没有丢失信息。

模式的分解与规范化（续）

例：

上面的第三种分解方法虽然具有无损连接性，保证了不丢失原关系中的信息，但它并没有解决插入异常、删除异常、修改复杂、数据冗余等问题。

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。

之所以出现上述问题，是因为分解得到的两个关系模式不是互相独立的。SL中的函数依赖 $Sdept \rightarrow Sloc$ 既没有投影到关系模式ND上，也没有投影到关系模式NL上，而是跨在这两个关系模式上。也就是这种分解方法没有保持原关系中的函数依赖。

模式的分解 与规范化 (续)

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

模式的分解 与规范化（续）

例：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 与规范化（续）

- 判断对关系模式的一个分解是否与原关系模式等价的标准
 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 与规范化（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 与规范化（续）

例：上例中

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。

第二种分解方法保持了函数依赖，但不具有无损连接性。

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。

模式的分解 与规范化（续）

- 规范化理论提供了一套完整的模式分解算法，按照这套算法可以做到：
 - 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。

函数依赖闭包和属性闭包

- 函数依赖闭包
- 属性闭包

函数依赖闭包(1)

Definition: Let F be a set of FDs in R . The **closure** of F is the set of all FDs that are logically implied by F .

- The closure of F is denoted by F^+ .

函数依赖闭包(2)

- A BIG F^+ may be derived from a small F .

Example: For $R(A, B, C)$ and

$$F = \{A \longrightarrow B, B \longrightarrow C\}$$

$$F^+ = \{A \longrightarrow B, B \longrightarrow C, A \longrightarrow C,$$

$$A \longrightarrow A, B \longrightarrow B, C \longrightarrow C,$$

$$AB \longrightarrow AB, AB \longrightarrow A, AB \longrightarrow B, \dots \}$$

$$|F^+| > 30.$$

函数依赖闭包 F^+ 计算

- 利用Armstrong公理

- 一套推理规则，是模式分解算法的理论基础
- 用途
 - 从一组函数依赖求得蕴含的函数依赖

函数依赖闭包 F^+ 计算

- 利用Armstrong公理

- 一套推理规则，是模式分解算法的理论基础
- 用途
 - 从一组函数依赖求得蕴含的函数依赖

Armstrong公理系统

Armstrong's Axioms (1974):

(IR1) Reflexivity rule(自反律):

If $X \supseteq Y$, then $X \longrightarrow Y$.

(IR2) Augmentation rule(增广律):

$\{ X \longrightarrow Y \} \models XZ \longrightarrow YZ$.

(IR3) Transitivity rule(传递律):

$\{ X \longrightarrow Y, Y \longrightarrow Z \} \models X \longrightarrow Z$.

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 。

Armstrong公理系统（续）

Additional rules derivable from Armstrong's Axioms.

(IR4) Decomposition rule(分解规则):

$$\{ X \longrightarrow YZ \} \models \{ X \longrightarrow Y, X \longrightarrow Z \}$$

(IR5) Union rule(合并规则):

$$\{ X \longrightarrow Y, X \longrightarrow Z \} \models X \longrightarrow YZ$$

(IR6) Pseudotransitivity rule(伪传递规则):

$$\{ X \longrightarrow Y, WY \longrightarrow Z \} \models WX \longrightarrow Z$$

Armstrong公理系统的有效性与完备性

● 有效性与完备性的含义

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公理系统的有效性 with 完备性（续）

● 有效性与完备性的证明

定理5.2 Armstrong公理系统是有效的、完备的。

Armstrong公理系统的有效性可由**定理5.1**得到证明。这里给出完备性的证明。

证明完备性的逆否命题，即若函数依赖 $X \rightarrow Y$ 不能由 F 从 Armstrong 公理导出，那么它必然不为 F 所蕴含，它的证明分三步。

属性闭包计算(1)

求属性闭包的算法

算法5.1 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 $B = \{A \mid (\exists V) (\exists W) (V \rightarrow W \in F \wedge V \subseteq X^{(i)} \wedge A \in W)\}$;

属性闭包计算(2)

(3) $X^{(i+1)} = BU X^{(i)}$

(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。

属性闭包计算(3)

[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 由算法5.1, 设 $X^{(0)}=AB$;

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 $X^{(1)}=AB\cup CD=ABCD$ 。

属性闭包计算(4)

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 $C \rightarrow E$ ， $AC \rightarrow B$ ，于是 $X^{(2)} = X^{(1)} \cup BE = ABCDE$ 。

因为 $X^{(2)}$ 已等于全部属性集合，所以 $(AB)_F^+ = ABCDE$ 。

对于算法5.1，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。

属性闭包计算(6)

Example:

$R(A, B, C, G, H, I) = ABCGHI$

$X = AG$

$F = \{A \longrightarrow B, CG \longrightarrow HI, B \longrightarrow H, A \longrightarrow C\}$

Compute $(AG)^+$.

Initialization: $xplus := AG$;

属性闭包计算(7)

1st iteration:

consider $A \longrightarrow B$,

since A is a subset of $xplus$, $xplus = ABG$;

consider $CG \longrightarrow HI$,

since CG is not a subset of $xplus$, $xplus = ABG$;

consider $B \longrightarrow H$,

since B is a subset of $xplus$, $xplus = ABGH$;

consider $A \longrightarrow C$,

since A is a subset of $xplus$, $xplus = ABCGH$;

($xplus$ is changed from AG to $ABCGH$)

属性闭包计算(8)

2nd iteration:

consider $A \twoheadrightarrow B$,

since A is a subset of $xplus$, $xplus = ABCGH$;

consider $CG \twoheadrightarrow HI$,

since CG is a subset of $xplus$, $xplus = ABCGHI$;

consider $B \twoheadrightarrow H$,

since B is a subset of $xplus$, $xplus = ABCGHI$;

consider $A \twoheadrightarrow C$,

since A is a subset of $xplus$, $xplus = ABCGHI$;

($xplus$ is changed from $ABCGH$ to $ABCGHI$)

属性闭包计算(9)

3rd iteration:

**(consider each FD in F again, but there is
no change to xplus, exit)**

Result: $(AG)^+ = ABCGHI$.

下个学期双语课

- 《计算机科学选讲》
- 信息检索与搜索引擎技术
- XML数据管理技术
- 关键词模糊搜索技术

期中考试

- 周一上午10点到12点
- 闭卷

习题五

● 1,2,4,5,9,10,11,12