

Normalización de un Esquema de Relación
Organización de Archivos y Bases de Datos I
Ciclo 2005
<http://www.dirinfo.unsl.edu.ar/~oaybd1/>

OA y BDI
UNSLAño 2005

Normalización en 1FN

Definición:
Sea R un esquema de relación.
Diremos que R está en 1FN si y sólo si para todo atributo $A \in R$ los valores de $\text{dom}(A)$ son atómicos.

- Un atributo no es atómico si la aplicación trata con partes de ese atributo.
- Es una forma normal bastante intuitiva: nadie pone en un solo campo de un registro aquella información que necesita ser accedida individualmente.

OA y BDI
UNSLAño 2005

Normalización en 1FN

Ejemplo
Materias = { CodM , NombreM, Docentes }

CodM	NbreM	Docentes
10	Lógica	{D1, D2, D3}
20	Cálculo	{D2, D4}

No está en 1FN

Materias = { CodM , NombreM, Docente }

CodM	NbreM	Docente
10	Lógica	D1
10	Lógica	D2
10	Lógica	D3
20	Cálculo	D2
20	Cálculo	D4

Está en 1FN

OA y BDI
UNSLAño 2005

Normalización en 2FN

Definición:
Sea: R un esquema de relación.
 F un conjunto de dependencias funcionales
Diremos que R está en 2FN con respecto a F si y sólo si: R está en 1FN y todo atributo no primo depende plenamente de toda clave.

- Si R no está en 2FN respecto de F , entonces existe una dependencia no plena de atributos no primos respecto de alguna de las claves.
- Esto implica que en R hay redundancia de datos.

OA y BDI
UNSLAño 2005

Normalización en 2FN

- Si R no está en 2FN, entonces existe una clave X y existe $Z \subset R$, Z formado sólo por atributos no primo tales que:

$X \xrightarrow{pf} Z$

Esto implica que existe $Y \subset X$ tal que $Y \rightarrow Z$.

- Entonces, descomponemos R en dos subsquemas:
 - $R_1 = R - Z$
 - $R_2 = Y Z$

OA y BDI
UNSLAño 2005

Normalización en 2FN

R
 $X \xrightarrow{pf} Z$ porque existe $Y \subset X$ tal que $Y \rightarrow Z$

$R_1 = R - Z$

$R_2 = Y Z$

- Ahora analizamos si R_1 y R_2 están en 2FN.
- Si no lo están, aplicamos el mismo proceso recursivamente hasta llegar a esquemas que estén en 2FN.

Este algoritmo se conoce con el nombre de **Algoritmo de Descomposición.**

OA y BDI
UNSLAño 2005

Normalización en 2FN

R
Si R no está en 2FN

R₁ **R₂**
Si R₁ no está en 2FN. Si R₂ no está en 2FN.

R₁₁ **R₁₂** **R₂₁** **R₂₂**

⋮ ⋮ ⋮ ⋮

- Se produce un árbol de descomposición, donde los terminales forman una descomposición en 2FN.
- La descomposición encontrada es LLJ.

OA y BDI U.N.S.L. Año 2005

Normalización en 2FN

Lema 1: (demostrado en clases anteriores)

Sea R un esquema de relación, F un conjunto de dependencias funcionales y $\rho = \{R_1, R_2\}$ una descomposición de R. ρ es LLJ si y sólo si:

$$(R_1 \cap R_2) \rightarrow (R_1 - R_2) \text{ o } (R_1 \cap R_2) \rightarrow (R_2 - R_1)$$

Lema 2: (7.6, Ullman)

Sea R un esquema de relación, F un conjunto de dependencias funcionales y $\rho = \{R_1, R_2, \dots, R_n\}$ una descomposición LLJ de R. Sea $\sigma = \{R_{11}, R_{12}\}$ una descomposición LLJ de R₁.
Luego:

$$\rho = \{R_1, R_2, \dots, R_{11}, R_{12}, R_{11}, R_{12}, R_{11}, R_{12}, \dots, R_n\}$$

es una descomposición LLJ de R.

OA y BDI U.N.S.L. Año 2005

Normalización en 2FN

Teorema

El algoritmo de descomposición produce una descomposición LLJ.

Demostración:

La demostración de este teorema se realiza por inducción sobre ρ usando los dos lemas anteriores.

Queda como ejercicio.

OA y BDI U.N.S.L. Año 2005

Normalización en 2FN

Ejemplo:

$R = \{ \text{NroA, NbreA, DirA, CodM, NbreM, Año, Cuat, CodP, NbreP} \}$

$F = \{ \text{NroA} \rightarrow \text{NbreA DirA, CodM} \rightarrow \text{NbreM Año Cuat CodP NbreP, CodP} \rightarrow \text{NbreP} \}$

Clave: NroA CodM

No está en 2FN, porque existen dependencias no plenas de atributos no primos respecto de la clave

$$\underbrace{\text{NroA CodM}}_X \xrightarrow{p1} \underbrace{\text{NbreA DirA}}_Z \text{ porque } \underbrace{\text{NroA} \rightarrow \text{NbreA DirA}}_Y$$

OA y BDI U.N.S.L. Año 2005

Normalización en 3FN

Definición 1 (Maier):

Sea: R un esquema de relación.
F un conjunto de dependencias funcionales

Diremos que R está en 3FN con respecto a F si y sólo si: R está en 1FN y no existen dependencias transitivas de atributos no primos respecto de alguna de las claves

Definición 2 (Ullman):

Sea: R un esquema de relación (se supone ya en 1FN).
F un conjunto de dependencias funcionales

Diremos que R está en 3FN con respecto a F si y sólo si para toda dependencia $X \rightarrow A$ válida en R (con $A \notin X$) se cumple que:

ó X es superclave ó A es primo

OA y BDI U.N.S.L. Año 2005

Normalización en 3FN

- Si R no está en 3FN respecto de F, entonces en R existen redundancia de datos.
- Existen dos algoritmos para normalizar R en 3FN:
 1. Algoritmo de Descomposición (el mismo que vimos para 2FN)
 2. Algoritmo de Síntesis
- Nota: el algoritmo de descomposición sirve para normalizar R en 2FN, 3FN y BCNF.

OA y BDI U.N.S.L. Año 2005

Normalización en 3FN (Descomposición)

Algoritmo de Descomposición

- Si R no está en 3FN respecto de F, entonces existe una dependencia transitiva de atributos no primos respecto de alguna clave.
- En símbolos:
 - existe una clave X
 - existe $Z \subset R$, Z formado sólo por atributos no primos.
 - existe $Y \subset R$
 tales que:

$$X \rightarrow Y \wedge Y \not\rightarrow X \wedge Y \rightarrow Z \text{ con } Z \not\subset XY$$
- Entonces, descomponemos R en dos subesquemas:

$$R_1 = R - Z$$

$$R_2 = YZ$$

OA y BDI U.N.S.L. Año 2005

Normalización en 3FN (Descomposición)

R

$X \rightarrow Y \wedge Y \not\rightarrow X \wedge Y \rightarrow Z \text{ con } Z \not\subset XY$

$R_1 = R - Z$ $R_2 = YZ$

- Ahora analizamos si R_1 y R_2 están en 3FN.
- Si no lo están, aplicamos el mismo proceso recursivamente hasta llegar a esquemas que estén en 3FN.
- Se produce un árbol de descomposición, donde los terminales forman una descomposición de R que está en 3FN y es LLJ.

OA y BDI U.N.S.L. Año 2005

Normalización en 3FN (Descomposición)

Problemas de Normalización a través de Descomposición

Problema 1 :
Se puede producir una descomposición con más esquemas de los necesarios.

Solución:
Cuando planteamos la dependencia transitiva

$$X \rightarrow Y \wedge Y \not\rightarrow X \wedge Y \rightarrow Z \text{ con } Z \not\subset XY$$

Ponemos en Z la máxima cantidad de atributos posibles, es decir:

$$Z = Y^+ - X - Y - AP$$

donde AP es el conjunto de atributos primos,

OA y BDI U.N.S.L. Año 2005

Normalización en 3FN (Descomposición)

Problema 2:
Se pueden introducir dependencias no plenas, lo que produce una descomposición con más esquemas de los necesarios.

Solución:
Cuando planteamos la dependencia transitiva

$$X \rightarrow Y \wedge Y \not\rightarrow X \wedge Y \rightarrow Z \text{ con } Z \not\subset XY$$

Ponemos en Y la mínima cantidad de atributos posibles; esto significa que la dependencia $Y \rightarrow Z$ debe ser reducida a izquierda

Problema 3:
Puede producir una descomposición que no preserve dependencias.

OA y BDI U.N.S.L. Año 2005

Normalización en 3FN (Síntesis)

Algoritmo de Síntesis (Algoritmo 7.5, Ullman)

Entrada:

R esquema de relación.

F un conjunto minimal de dependencias funcionales

Salida:

$\rho = \{R_1, R_2, \dots, R_n\}$ una descomposición en 3FN de R, donde ρ preserva el conjunto de dependencias F.

Método:

1. Si existen atributos que no participan en ninguna dependencia, entonces esos atributos forman un subesquema.
2. Por cada dependencia $X \rightarrow A \in F$ formar un esquema $R_i = X \cup \{A\}$

OA y BDI U.N.S.L. Año 2005

Normalización en 3FN (Síntesis)

Problemas de Normalización a través de Síntesis

Problema 1 :
Se puede producir una descomposición con más esquemas de los necesarios.

Solución:

1. Si existen dependencias de la forma:

$$X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_n$$
 entonces se armar un único subesquema:

$$R_i = X \cup \{A_1, A_2, \dots, A_n\}$$
2. Tratar los anillos de dependencias separadamente, y luego aplicar el algoritmo de síntesis al resto de las dependencias.

OA y BDI U.N.S.L. Año 2005

Problema 2 :

Se puede producir una descomposición que no sea LLJ.

Solución:

La solución a este problema se basa en el siguiente teorema

Teorema (Teorema 7.8, Ullman)

Sea ρ una descomposición 3FN de R obtenida por el algoritmo de síntesis. Sea X una clave de R. Entonces $\sigma = \rho \cup \{X\}$ es una descomposición en 3FN de R que es LLJ y PD.

Sea $\rho = \{ R_1, R_2, \dots, R_n \}$ una descomposición 3FN de R obtenida por el algoritmo de síntesis.

- Si al menos una de las claves de R está incluida en un subesquema $R_i \in \rho$, entonces la descomposición ρ es LLJ (además de 3FN y PD).
- Si ninguna de las claves está incluida en un subesquema, entonces agregamos un subesquema con una de las claves para que la descomposición sea LLJ (además de 3FN y PD).

□