Bases de Datos I

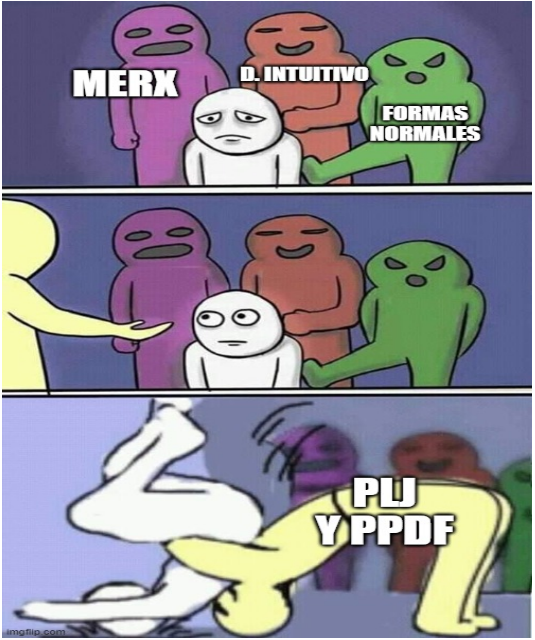
Teoría del diseño: PLJ, PPDF y BCFN

Lic. Andy Ledesma García
Lic. Víctor M. Cardentey Fundora
Dra. C. Lucina García Hernández

Departamento de Computación
Facultad de Matemática y Computación
Universidad de La Habana

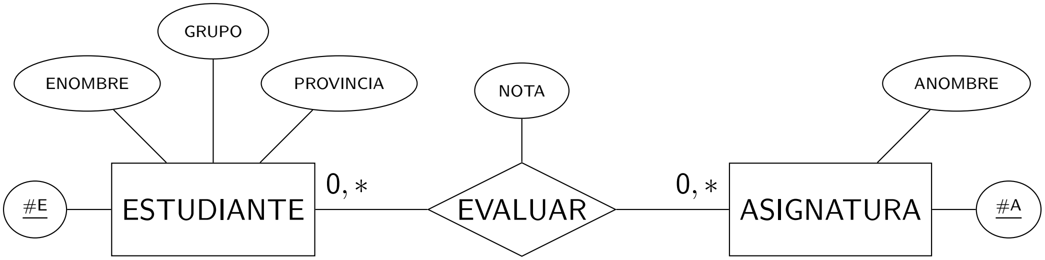
11 de agosto de 2024

Anteriormente...



Anteriormente en Bases de Datos I

Partimos de un diseño intuitivo obtenido a partir de la especificación...



Anteriormente en Bases de Datos I

Para obtener un universo de atributos y las dependencias funcionales que se establecen entre estos...

1. $U = \{$#E, ENombre, Grupo, Provincia, #A, ANombre, Nota$\}$
2. $F = \{$
   #E $\rightarrow$ ENombre, Grupo, Provincia
   #A $\rightarrow$ ANombre
   #E,#A $\rightarrow$ #E,#A
   #E, #A $\rightarrow$ Nota
   Provincia $\rightarrow$ Grupo
   $\}$
3. Definimos el esquema relacional **Evaluaciones**(U, F) con llave #E, #A

Calculamos un cubrimiento minimal de F para deshacernos de dependencias "problemáticas"...

#E $\rightarrow$ ENombre
#E $\rightarrow$ Grupo
#E $\rightarrow$ Provincia
#A $\rightarrow$ ANombre
#E, #A $\rightarrow$ #E, #A
#E, #A $\rightarrow$ Nota
Provincia $\rightarrow$ Grupo

Anteriormente en Bases de Datos I

Calculamos un cubrimiento minimal de F para deshacernos de dependencias "problemáticas"...

#E → ENombre
#E → Grupo
#E → Provincia
#A → ANombre
#E, #A → #E, #A
#E, #A → Nota
Provincia → Grupo

$$\#E \rightarrow \text{Provincia} \wedge \text{Provincia} \rightarrow \text{Grupo} \models \#E \rightarrow \text{Grupo}$$

Anteriormente en Bases de Datos I

Calculamos un cubrimiento minimal de F para deshacernos de dependencias "problemáticas"...

#E $\rightarrow$ ENombre
#E $\rightarrow$ Grupo
#E $\rightarrow$ Provincia
#A $\rightarrow$ ANombre
#E, #A $\rightarrow$ #E, #A
#E, #A $\rightarrow$ Nota
Provincia $\rightarrow$ Grupo

#E $\rightarrow$ ENombre
#E $\rightarrow$ Provincia
#A $\rightarrow$ ANombre
#E, #A $\rightarrow$ #E, #A
#E, #A $\rightarrow$ Nota
Provincia $\rightarrow$ Grupo

Anteriormente en Bases de Datos 1

Y normalizamos para evitar anomalías y asegurar el cumplimiento de las dependencias funcionales.

$R_1(U_1, F_1)$:
$U_1 = \{\#\text{E, NombreE, Provincia}\}$
$F_1 = \pi_{U_1}(F)$

$R_2(U_2, F_2)$:
$U_2 = \{\text{Provincia, Grupo}\}$
$F_2 = \pi_{U_2}(F)$

$R_3(U_3, F_3)$:
$U_3 = \{\#\text{A, NombreA}\}$
$F_3 = \pi_{U_3}(F)$

$R_4(U_4, F_4)$:
$U_4 = \{\#\text{E}, \#\text{A, Nota}\}$
$F_4 = \pi_{U_4}(F)$

¿Qué es un diseño teóricamente correcto?

- Todos los esquemas relacionales de la descomposición están en una forma normal aceptable (3FN o superior).
- Se cumple la propiedad de join sin pérdida de información (PLJ).
- Se cumple la propiedad de preservación de dependencias funcionales (PPDF).

¿Qué es un diseño teóricamente correcto?

- Todos los esquemas relacionales de la descomposición están en una forma normal aceptable (3FN o superior).
- Se cumple la propiedad de join sin pérdida de información (PLJ).
- Se cumple la propiedad de preservación de dependencias funcionales (PPDF).

Un diseño correcto no garantiza que sea el mejor

¿PLJ y PPDF?

¿Al reunir las relaciones normalizadas la relación resultante será la original?

Propiedad del Join sin Pérdida de Información

Si para un esquema relacional $R(U, F)$ se tiene la descomposición $\rho = (R_1, R_2, ..., R_k)$, se dice que dicha descomposición ρ cumple con la propiedad de join ($\bowtie$) sin pérdida de información con respecto al conjunto de dependencias funcionales F si para toda instancia r de R que satisfaga a F, se cumple que:

$$r = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie ... \bowtie \pi_{R_k}(r) = \mathop{\bowtie}_{i=1}^{k} R_i(r)$$

Algoritmo para determinar si se cumple la PLJ

Entrada:

- Un esquema relacional $R(U, F)$ con $U = \{A_1, A_2, ..., A_n\}$
- Una descomposición $\rho = (R_1(U_1, F_1), R_2(U_2, F_2), ..., R_k(U_k, F_k))$

Salida: Una decisión de si ρ cumple con la PLJ.

Algoritmo para determinar si se cumple la PLJ

1. Construir una tabla T de n columnas y k filas donde:
   - Si A_j pertenece a U_i entonces $T_{ij} = a_j$
   - En caso contrario $T_{ij} = b_{ij}$

	E	EN	P	G	A	AN	N
(E, EN, P)	a_1	a_2	a_3	b_{14}	b_{15}	b_{16}	b_{17}
(P,G)	b_{21}	b_{22}	a_3	a_4	b_{25}	b_{26}	b_{27}
(A,AN)	b_{31}	b_{32}	b_{33}	b_{34}	a_5	a_6	b_{37}
(E,A,N)	a_1	b_{42}	b_{43}	b_{44}	a_5	b_{46}	a_7

Algoritmo para determinar si se cumple la PLJ

2. Por cada dependencia funcional $X \rightarrow A_j$ en F se buscan todas las filas en que coincidan los valores de los símbolos de todas las columnas que correspondan a los atributos de X, si se encuentran dos o más filas que cumplan tal condición, se igualan los símbolos de la columna que corresponden al atributo A.
   - Si uno de los símbolos es a_j los otros se igualan a a_j.
   - Si todos son b_{ij} los otros se igualan a alguno de ellos tomado arbitrariamente.

	E	EN	P	G	A	AN	N
(E, EN, P)	a_1	a_2	a_3	b_{14}	b_{15}	b_{16}	b_{17}
(P,G)	b_{21}	b_{22}	a_3	a_4	b_{25}	b_{26}	b_{27}
(A,AN)	b_{31}	b_{32}	b_{33}	b_{34}	a_5	a_6	b_{37}
(E,A,N)	a_1	b_{42}	b_{43}	b_{44}	a_5	b_{46}	a_7

E $\rightarrow$ EN, P
P $\rightarrow$ G
A $\rightarrow$ AN
E, A $\rightarrow$ N

Algoritmo para determinar si se cumple la PLJ

2. Por cada dependencia funcional $X \rightarrow A_j$ en F se buscan todas las filas en que coincidan los valores de los símbolos de todas las columnas que correspondan a los atributos de X, si se encuentran dos o más filas que cumplan tal condición, se igualan los símbolos de la columna que corresponden al atributo A.
   - Si uno de los símbolos es a_j los otros se igualan a a_j.
   - Si todos son b_{ij} los otros se igualan a alguno de ellos tomado arbitrariamente.

	E	EN	P	G	A	AN	N
(E, EN, P)	a_1	a_2	a_3	b_{14}	b_{15}	b_{16}	b_{17}
(P,G)	b_{21}	b_{22}	a_3	a_4	b_{25}	b_{26}	b_{27}
(A,AN)	b_{31}	b_{32}	b_{33}	b_{34}	a_5	a_6	b_{37}
(E,A,N)	a_1	b_{42}	b_{43}	b_{44}	a_5	b_{46}	a_7

E $\rightarrow$ EN, P

P $\rightarrow$ G

A $\rightarrow$ AN

E, A $\rightarrow$ N

Algoritmo para determinar si se cumple la PLJ

2. Por cada dependencia funcional $X \to A_j$ en F se buscan todas las filas en que coincidan los valores de los símbolos de todas las columnas que correspondan a los atributos de X, si se encuentran dos o más filas que cumplan tal condición, se igualan los símbolos de la columna que corresponden al atributo A.
   - Si uno de los símbolos es a_j los otros se igualan a a_j.
   - Si todos son b_{ij} los otros se igualan a alguno de ellos tomado arbitrariamente.

	E	EN	P	G	A	AN	N
(E, EN, P)	a_1	a_2	a_3	b_{14}	b_{15}	b_{16}	b_{17}
(P,G)	b_{21}	b_{22}	a_3	a_4	b_{25}	b_{26}	b_{27}
(A,AN)	b_{31}	b_{32}	b_{33}	b_{34}	a_5	a_6	b_{37}
(E,A,N)	a_1	a_2	a_3	b_{44}	a_5	b_{46}	a_7

E $\to$ EN, P
P $\to$ G
A $\to$ AN
E, A $\to$ N

Algoritmo para determinar si se cumple la PLJ

2. Por cada dependencia funcional $X \rightarrow A_j$ en F se buscan todas las filas en que coincidan los valores de los símbolos de todas las columnas que correspondan a los atributos de X, si se encuentran dos o más filas que cumplan tal condición, se igualan los símbolos de la columna que corresponden al atributo A.
   - Si uno de los símbolos es a_j los otros se igualan a a_j.
   - Si todos son b_{ij} los otros se igualan a alguno de ellos tomado arbitrariamente.

	E	EN	P	G	A	AN	N
(E, EN, P)	a_1	a_2	a_3	b_{14}	b_{15}	b_{16}	b_{17}
(P,G)	b_{21}	b_{22}	a_3	a_4	b_{25}	b_{26}	b_{27}
(A,AN)	b_{31}	b_{32}	b_{33}	b_{34}	a_5	a_6	b_{37}
(E,A,N)	a_1	a_2	a_3	b_{44}	a_5	b_{46}	a_7

E $\rightarrow$ EN, P

P $\rightarrow$ G

A $\rightarrow$ AN

E, A $\rightarrow$ N

Algoritmo para determinar si se cumple la PLJ

2. Por cada dependencia funcional $X \rightarrow A_j$ en F se buscan todas las filas en que coincidan los valores de los símbolos de todas las columnas que correspondan a los atributos de X, si se encuentran dos o más filas que cumplan tal condición, se igualan los símbolos de la columna que corresponden al atributo A.
   - Si uno de los símbolos es a_j los otros se igualan a a_j.
   - Si todos son b_{ij} los otros se igualan a alguno de ellos tomado arbitrariamente.

	E	EN	P	G	A	AN	N
(E, EN, P)	a_1	a_2	a_3	a_4	b_{15}	b_{16}	b_{17}
(P,G)	b_{21}	b_{22}	a_3	a_4	b_{25}	b_{26}	b_{27}
(A,AN)	b_{31}	b_{32}	b_{33}	b_{34}	a_5	a_6	b_{37}
(E,A,N)	a_1	a_2	a_3	a_4	a_5	b_{46}	a_7

E $\rightarrow$ EN, P

P $\rightarrow$ G

A $\rightarrow$ AN

E, A $\rightarrow$ N

Algoritmo para determinar si se cumple la PLJ

2. Por cada dependencia funcional $X \to A_j$ en F se buscan todas las filas en que coincidan los valores de los símbolos de todas las columnas que correspondan a los atributos de X, si se encuentran dos o más filas que cumplan tal condición, se igualan los símbolos de la columna que corresponden al atributo A.
   - Si uno de los símbolos es a_j los otros se igualan a a_j.
   - Si todos son b_{ij} los otros se igualan a alguno de ellos tomado arbitrariamente.

	E	EN	P	G	A	AN	N
(E, EN, P)	a_1	a_2	a_3	a_4	b_{15}	b_{16}	b_{17}
(P,G)	b_{21}	b_{22}	a_3	a_4	b_{25}	b_{26}	b_{27}
(A,AN)	b_{31}	b_{32}	b_{33}	b_{34}	a_5	a_6	b_{37}
(E,A,N)	a_1	a_2	a_3	a_4	a_5	b_{46}	a_7

E $\to$ EN, P

P $\to$ G

A $\to$ AN

E, A $\to$ N

Algoritmo para determinar si se cumple la PLJ

2. Por cada dependencia funcional $X \rightarrow A_j$ en F se buscan todas las filas en que coincidan los valores de los símbolos de todas las columnas que correspondan a los atributos de X, si se encuentran dos o más filas que cumplan tal condición, se igualan los símbolos de la columna que corresponden al atributo A.
   - Si uno de los símbolos es a_j los otros se igualan a a_j.
   - Si todos son b_{ij} los otros se igualan a alguno de ellos tomado arbitrariamente.

	E	EN	P	G	A	AN	N
(E, EN, P)	a_1	a_2	a_3	a_4	b_{15}	b_{16}	b_{17}
(P,G)	b_{21}	b_{22}	a_3	a_4	b_{25}	b_{26}	b_{27}
(A,AN)	b_{31}	b_{32}	b_{33}	b_{34}	a_5	a_6	b_{37}
(E,A,N)	a_1	a_2	a_3	a_4	a_5	a_6	a_7

E $\rightarrow$ EN, P
P $\rightarrow$ G
A $\rightarrow$ AN
E, A $\rightarrow$ N

Algoritmo para determinar si se cumple la PLJ

2. Por cada dependencia funcional $X \to A_j$ en F se buscan todas las filas en que coincidan los valores de los símbolos de todas las columnas que correspondan a los atributos de X, si se encuentran dos o más filas que cumplan tal condición, se igualan los símbolos de la columna que corresponden al atributo A.
    - Si uno de los símbolos es a_j los otros se igualan a a_j.
    - Si todos son b_{ij} los otros se igualan a alguno de ellos tomado arbitrariamente.

	E	EN	P	G	A	AN	N
(E, EN, P)	a_1	a_2	a_3	a_4	b_{15}	b_{16}	b_{17}
(P,G)	b_{21}	b_{22}	a_3	a_4	b_{25}	b_{26}	b_{27}
(A,AN)	b_{31}	b_{32}	b_{33}	b_{34}	a_5	a_6	b_{37}
(E,A,N)	a_1	a_2	a_3	a_4	a_5	a_6	a_7

E $\to$ EN, P

P $\to$ G

A $\to$ AN

E, A $\to$ N

Algoritmo para determinar si se cumple la PLJ

3. Si al concluir una pasada por todas las DF en F, una fila contiene solo símbolos a_j entonces la descomposición ρ cumple la PLJ. Si no, en caso de que se haya producido un cambio en la tabla se repite el paso 2 en caso contrario la descomposición no cumple la PLJ.

	E	EN	P	G	A	AN	N
(E, EN, P)	a_1	a_2	a_3	a_4	b_{15}	b_{16}	b_{17}
(P,G)	b_{21}	b_{22}	a_3	a_4	b_{25}	b_{26}	b_{27}
(A,AN)	b_{31}	b_{32}	b_{33}	b_{34}	a_5	a_6	b_{37}
(E,A,N)	a_1	a_2	a_3	a_4	a_5	a_6	a_7

La descomposición cumple la PLJ

Propiedad de Preservación de Dependencias Funcionales (PPDF)

Si para un $R(U, F)$ se tiene la descomposición $\rho = (R_1, R_2, ..., R_k)$, se dice que ρ cumple la Propiedad de Preservación de Dependencias Funcionales (PPDF) con respecto al conjunto de dependencias funcionales F si:

$$F \equiv \bigcup_{i=1}^{k} \Pi_{R_i}(F)$$

Recordando...

Entrada: Un esquema relacional $R(U, F)$, F es un conjunto irreducible de dependencias funcionales.
Salida: Una descomposición $\rho = (R_1, R_2, ..., R_n)$, tal que los esquemas relacionales $R_i(U_i, F_i)$ están en 3FN con respecto a $\Pi_{R_i}(F)$, $\forall i = 1, ..., n$.
Método:

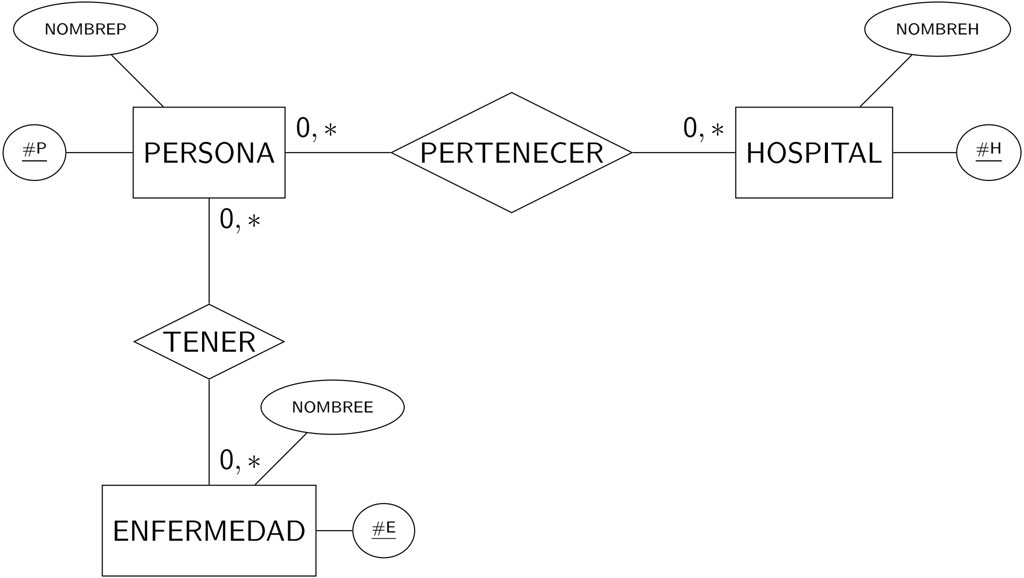
1. Eliminar cada $X \to Y$ en F y añadir $X \to A_i$ para todo $A_i \in Y$.
2. Por cada dependencia funcional $X \to A_i$ en F crear el esquema relacional $R_i(U_i, F_i)$ tal que $U_i = X \cup \{A_i\}$ y $F_i = \Pi_{R_i}(F)$. Si en F se tiene $X \to A_1$, $X \to A_2$,..., $X \to A_k$ se puede utilizar un esquema relacional de la forma $R_j(U_j, F_j)$ con $U_j = X \cup \{A_1, A_2, ..., A_k\}$ y $F_j = \Pi_{R_j}(F)$.
3. Si en U existe algún atributo que no está contenido en ninguna dependencia funcional de F, este atributo puede formar un esquema relacional por sí mismo.
4. Luego, $\rho = (R_1, R_2, ..., R_n)$

Entonces...

El algoritmo para obtener una descomposición en 3FN siempre cumple la PPDF

¿Ocurrirá lo mismo con la PLJ?

Supongamos el siguiente escenario



Definiendo el esquema universal

1. $U = \{\#\text{P, NombreP, }\#\text{H, NombreH, }\#\text{E, NombreE}\}$
2. $F = \{$

   $\#\text{P} \to \text{NombreP}$

   $\#\text{H} \to \text{NombreH}$

   $\#\text{E} \to \text{NombreE}$

   $\#\text{P,}\#\text{H} \to \#\text{P,}\#\text{H}$

   $\#\text{P,}\#\text{E} \to \#\text{P,}\#\text{E}$

   $\}$
3. Definimos el esquema relacional **Hospital**(U, F) con llave #P, #H, #E

Definiendo el esquema universal

1. $U = \{$#P, NombreP, #H, NombreH, #E, NombreE$\}$
2. $F = \{$

   $$\begin{array}{l} \#\text{P} \rightarrow \text{NombreP} \\ \#\text{H} \rightarrow \text{NombreH} \\ \#\text{E} \rightarrow \text{NombreE} \\ \#\text{P},\#\text{H} \rightarrow \#\text{P},\#\text{H} \\ \#\text{P},\#\text{E} \rightarrow \#\text{P},\#\text{E} \end{array}$$

   $\}$
3. Definimos el esquema relacional **Hospital**(U, F) con llave #P, #H, #E

Ya F es un cubrimiento minimal

Obteniendo una descomposición en 3FN que cumple la PPDF

$R_1(U_1, F_1)$:
$U_1 = \{\#\text{P}, \text{NombreP}\}$
$F_1 = \Pi_{U_1}(F)$

$R_2(U_2, F_2)$:
$U_2 = \{\#\text{H}, \text{NombreH}\}$
$F_2 = \Pi_{U_2}(F)$

$R_3(U_3, F_3)$:
$U_3 = \{\#\text{E}, \text{NombreE}\}$
$F_3 = \Pi_{U_3}(F)$

$R_4(U_4, F_4)$:
$U_4 = \{\#\text{P}, \#\text{H}\}$
$F_4 = \Pi_{U_4}(F)$

$R_5(U_5, F_5)$:
$U_5 = \{\#\text{P}, \#\text{E}\}$
$F_5 = \Pi_{U_5}(F)$

Comprobando la PLJ

	P	NP	H	NH	E	NE
R_1	a_1	a_2	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	a_3	a_4	b_{25}	b_{26}
R_3	b_{31}	b_{32}	b_{33}	b_{34}	a_5	a_6
R_4	a_1	b_{42}	a_3	b_{44}	b_{45}	b_{46}
R_5	a_1	b_{52}	b_{53}	b_{54}	a_5	b_{56}

#P $\rightarrow$ NombreP
#H $\rightarrow$ NombreH
#E $\rightarrow$ NombreE
#P,#H $\rightarrow$ #P,#H
#P,#E $\rightarrow$ #P,#E

Comprobando la PLJ

	P	NP	H	NH	E	NE
R_1	a_1	a_2	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	a_3	a_4	b_{25}	b_{26}
R_3	b_{31}	b_{32}	b_{33}	b_{34}	a_5	a_6
R_4	a_1	b_{42}	a_3	b_{44}	b_{45}	b_{46}
R_5	a_1	b_{52}	b_{53}	b_{54}	a_5	b_{56}

#P → NombreP
#H → NombreH
#E → NombreE
#P,#H → #P,#H
#P,#E → #P,#E

Comprobando la PLJ

	P	NP	H	NH	E	NE
R_1	a_1	a_2	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	a_3	a_4	b_{25}	b_{26}
R_3	b_{31}	b_{32}	b_{33}	b_{34}	a_5	a_6
R_4	a_1	a_2	a_3	b_{44}	b_{45}	b_{46}
R_5	a_1	a_2	b_{53}	b_{54}	a_5	b_{56}

#P $\rightarrow$ NombreP
#H $\rightarrow$ NombreH
#E $\rightarrow$ NombreE
#P,#H $\rightarrow$ #P,#H
#P,#E $\rightarrow$ #P,#E

Comprobando la PLJ

	P	NP	H	NH	E	NE
R_1	a_1	a_2	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	a_3	a_4	b_{25}	b_{26}
R_3	b_{31}	b_{32}	b_{33}	b_{34}	a_5	a_6
R_4	a_1	a_2	a_3	b_{44}	b_{45}	b_{46}
R_5	a_1	a_2	b_{53}	b_{54}	a_5	b_{56}

#P $\rightarrow$ NombreP

#H $\rightarrow$ NombreH

#E $\rightarrow$ NombreE

#P,#H $\rightarrow$ #P,#H

#P,#E $\rightarrow$ #P,#E

Comprobando la PLJ

	P	NP	H	NH	E	NE
R_1	a_1	a_2	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	a_3	a_4	b_{25}	b_{26}
R_3	b_{31}	b_{32}	b_{33}	b_{34}	a_5	a_6
R_4	a_1	a_2	a_3	a_4	b_{45}	b_{46}
R_5	a_1	a_2	b_{53}	b_{54}	a_5	b_{56}

#P $\rightarrow$ NombreP
#H $\rightarrow$ NombreH
#E $\rightarrow$ NombreE
#P,#H $\rightarrow$ #P,#H
#P,#E $\rightarrow$ #P,#E

Comprobando la PLJ

	P	NP	H	NH	E	NE
R_1	a_1	a_2	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	a_3	a_4	b_{25}	b_{26}
R_3	b_{31}	b_{32}	b_{33}	b_{34}	a_5	a_6
R_4	a_1	a_2	a_3	a_4	b_{45}	b_{46}
R_5	a_1	a_2	b_{53}	b_{54}	a_5	b_{56}

#P $\rightarrow$ NombreP
#H $\rightarrow$ NombreH
#E $\rightarrow$ NombreE
#P,#H $\rightarrow$ #P,#H
#P,#E $\rightarrow$ #P,#E

Comprobando la PLJ

	P	NP	H	NH	E	NE
R_1	a_1	a_2	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	a_3	a_4	b_{25}	b_{26}
R_3	b_{31}	b_{32}	b_{33}	b_{34}	a_5	a_6
R_4	a_1	a_2	a_3	a_4	b_{45}	b_{46}
R_5	a_1	a_2	b_{53}	b_{54}	a_5	a_6

#P → NombreP
#H → NombreH
#E → NombreE
#P,#H → #P,#H
#P,#E → #P,#E

Comprobando la PLJ

	P	NP	H	NH	E	NE
R_1	a_1	a_2	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	a_3	a_4	b_{25}	b_{26}
R_3	b_{31}	b_{32}	b_{33}	b_{34}	a_5	a_6
R_4	a_1	a_2	a_3	a_4	b_{45}	b_{46}
R_5	a_1	a_2	b_{53}	b_{54}	a_5	a_6

#P $\rightarrow$ NombreP
#H $\rightarrow$ NombreH
#E $\rightarrow$ NombreE
#P,#H $\rightarrow$ #P,#H
#P,#E $\rightarrow$ #P,#E

Comprobando la PLJ

	P	NP	H	NH	E	NE
R_1	a_1	a_2	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	a_3	a_4	b_{25}	b_{26}
R_3	b_{31}	b_{32}	b_{33}	b_{34}	a_5	a_6
R_4	a_1	a_2	a_3	a_4	b_{45}	b_{46}
R_5	a_1	a_2	b_{53}	b_{54}	a_5	a_6

#P $\rightarrow$ NombreP
#H $\rightarrow$ NombreH
#E $\rightarrow$ NombreE
#P,#H $\rightarrow$ #P,#H
#P,#E $\rightarrow$ #P,#E

No se cumple la PLJ

La PLJ es un poco engañosa...

Requiere que se pueda reconstruir la relación universal pero...

¿Siempre tiene sentido tener una relación universal?

Otra forma de comprobar la PLJ

Teorema de Ullman

Si $\rho = (R_1, R_2)$ es una descomposición de $R(U, F)$ entonces ρ cumple con la PLJ respecto a F si y sólo si:

$$R_1 \cap R_2 \to R_1 - R_2 \quad \vee \quad R_1 \cap R_2 \to R_2 - R_1$$

Un parche para el algoritmo de 3FN que cumple la PPDF

Lema de Ullman

Sea ρ una descomposición en 3FN para $R(U, F)$ construida utilizando el algoritmo para obtener una descomposición en 3FN que cumple la PPDF, y sea X una llave del esquema $R(U, F)$. Entonces, $\sigma = \rho \cup X$ es una descomposición de $R(U, F)$ con todos sus esquemas relacionales en 3FN que cumple la PPDF, pero que además cumple con la PLJ.

Un parche para el algoritmo de 3FN que cumple la PPDF

Lema de Ullman

Sea ρ una descomposición en 3FN para $R(U, F)$ construida utilizando el algoritmo para obtener una descomposición en 3FN que cumple la PPDF, y sea X una llave del esquema $R(U, F)$. Entonces, $\sigma = \rho \cup X$ es una descomposición de $R(U, F)$ con todos sus esquemas relacionales en 3FN que cumple la PPDF, pero que además cumple con la PLJ.

Una descomposición obtenida con el algoritmo de 3FN y PPDF siempre puede ser convertida en un diseño correcto


ALGORITMO DE 3FN QUE CUMPLE PPDF PERO NO PLJ
LEMA DE ULLMAN
imgflip.com

Más allá de la 3FN

Forma Normal de Boyce Codd

Un esquema relacional $R(U, F)$ está en BCFN si cada uno de sus determinantes es una superllave o llave candidata del esquema.

Superllave

Dado un esquema relacional $R(U, F)$ un atributo $X \subseteq U$ es superllave de R si $X_F^+ = U$ y existe un atributo $Y \subset X$ tal que $Y_F^+ = U$.

Más allá de la 3FN

Forma Normal de Boyce Codd

Un esquema relacional $R(U, F)$ está en BCFN si cada uno de sus determinantes es una superllave o llave candidata del esquema.

Superllave

Dado un esquema relacional $R(U, F)$ un atributo $X \subseteq U$ es superllave de R si $X_F^+ = U$ y existe un atributo $Y \subset X$ tal que $Y_F^+ = U$.

Una superllave no es un atributo primo ni tampoco un atributo no primo

Algoritmo para obtener una descomposición en BCFN

Entrada: Un esquema relacional $R(U, F)$, donde F es un cubrimiento minimal.
Salida: Una descomposición $\rho = (R_1, R_2, ..., R_n)$ tal que cumple la propiedad PLJ y cada uno de sus esquemas relacionales R_i está en BCFN con respecto a $\Pi_{R_i}(F), \forall i = 1...n$.

Algoritmo para obtener una descomposición en BCFN

Método: Se construye iterativamente una descomposición ρ para $R(U, F)$ tal que en todo momento ρ cumpla con la PLJ con respecto a F.

1. Inicialmente $S = R$ (Raíz del árbol de descomposición)
2. Se recorre el árbol seleccionando un nodo hoja que contenga una dependencia funcional $X \to A$ que viola BCFN.
3. Sea S_0, el nodo hoja que no se encuentra en BCFN, se reemplaza por dos esquemas relacionales:
   - $S_1(U_1 = X \cup \{A\}, F_1 = \Pi_{U_1}(F))$
   - $S_2(U_2 = U_0 - \{A\}, F_2 = \Pi_{U_2}(F))$
4. Mientres resten nodos en el árbol por analizar se retorna a 2

Aplicando el algoritmo

$U_0 = U$
$F_0 = F$
Llave: HS

$U = \{CGTHRS\}$
$F = \{CS \to G, C \to T$
$HT \to R, HR \to C, HS \to R\}$

Aplicando el algoritmo

$U_0 = U$
$F_0 = F$
Llave: HS

$U_1 = \{CSG\}$
$F_1 = \{CS \rightarrow G\}$
Llave: CS

BCFN

$U_2 = \{CTHRS\}$
$F_2 = \{C \rightarrow T, HT \rightarrow R,$
$HR \rightarrow C, HS \rightarrow R\}$
Llaves: HS, HT

$U = \{CGTHRS\}$
$F = \{CS \rightarrow G, C \rightarrow T$
$HT \rightarrow R, HR \rightarrow C, HS \rightarrow R\}$

Aplicando el algoritmo

$U = \{CGTHRS\}$
$F = \{CS \to G, C \to T$
$HT \to R, HR \to C, HS \to R\}$

$U_0 = U$
$F_0 = F$
Llave: HS

$U_1 = \{CSG\}$
$F_1 = \{CS \to G\}$
Llave: CS

BCFN

$U_2 = \{CTHRS\}$
$F_2 = \{C \to T, HT \to R,$
$HR \to C, HS \to R\}$
Llaves: HS, HT

$U_3 = \{CT\}$
$F_3 = \{C \to T\}$
Llave: C

BCFN

$U_4 = \{CHRS\}$
$F_4 = \{CH \to R$
$HR \to C, HS \to R\}$
Llave: HS

Aplicando el algoritmo

$U = \{CGTHRS\}$
$F = \{CS \to G, C \to T$
$HT \to R, HR \to C, HS \to R\}$

$U_0 = U$
$F_0 = F$
Llave: HS

$U_1 = \{CSG\}$
$F_1 = \{CS \to G\}$
Llave: CS

BCFN

$U_2 = \{CTHRS\}$
$F_2 = \{C \to T, HT \to R,$
$HR \to C, HS \to R\}$
Llaves: HS, HT

$U_3 = \{CT\}$
$F_3 = \{C \to T\}$
Llave: C

BCFN

$U_4 = \{CHRS\}$
$F_4 = \{CH \to R$
$HR \to C, HS \to R\}$
Llave: HS

$U_5 = \{CHR\}$
$F_5 = \{CH \to R,$
$HR \to C\}$
Llaves: CH, HR

BCFN

$U_6 = \{CHS\}$
$F_6 = \{HS \to C\}$
Llave: HS

BCFN

Y ahora...

¿Se cumplirá la PPDF?

Recordando...

Propiedad de Preservación de Dependencias Funcionales

Si para un $R(U, F)$ se tiene la descomposición $\rho = (R_1, R_2, ..., R_k)$, se dice que ρ cumple la Propiedad de Preservación de Dependencias Funcionales (PPDF) con respecto al conjunto de dependencias funcionales F si:

$$F \equiv \bigcup_{i=1}^{k} \Pi_{R_i}(F)$$

Recordando...

Propiedad de Preservación de Dependencias Funcionales

Si para un $R(U, F)$ se tiene la descomposición $\rho = (R_1, R_2, ..., R_k)$, se dice que ρ cumple la Propiedad de Preservación de Dependencias Funcionales (PPDF) con respecto al conjunto de dependencias funcionales F si:

$$F \equiv \bigcup_{i=1}^{k} \Pi_{R_i}(F)$$

¿Cómo comprobar la equivalencia entre dos conjuntos de dependencias funcionales?

Equivalencia de conjuntos de dependencias funcionales

$$F \equiv G \Leftrightarrow F^+ = G^+$$

- Se debe considerar cada $X \rightarrow Y$ en F y determinar si X_G^+ contiene a Y.
- Se debe considerar cada $Z \rightarrow W$ en G y determinar si Z_F^+ contiene a W.

Equivalencia de conjuntos de dependencias funcionales

$$F \equiv G \Leftrightarrow F^+ = G^+$$

- Se debe considerar cada $X \to Y$ en F y determinar si X_G^+ contiene a Y.
- Se debe considerar cada $Z \to W$ en G y determinar si Z_F^+ contiene a W.

¿Y si no podemos calcular G?

R_i-operación

Si se tiene un esquema $R(U, F)$ y una descomposición $\rho = (R_1, R_2, ..., R_k)$ y $Z \subseteq U$, una R_i-operación consisten en añadir a Z aquellos atributos simples $A, A \subseteq U$, tales que: $(Z \cap R_i) \to A$ esté en $\Pi_{R_i}(F)$.

$$Z = Z \cup ((Z \cap R_i)^+ \cap R_i), \text{ donde } (Z \cap R_i)^+ \text{ sobre } F$$

Algoritmo para determinar si se cumple la PPDF

Entrada:

- Un esquema relacional $R(U, F)$
- Una descomposición $\rho = (R_1(U_1, F_1), R_2(U_2, F_2), ..., R_k(U_k, F_k))$

Salida: Una decisión de si ρ cumple con la PPDF

Método:

Por cada $X \to Y$ hacer
    $Z = X$
    Mientras ocurra un cambio en Z hacer:
        Para $i = 1...k$ hacer
         $Z = Z \cup ((Z \cap R_i)^+ \cap R_i)$
    Al concluir $Z = X_G^+$
    Si $Y \nsubseteq Z$ entonces ρ no cumple la PPDF
Si $Y \subseteq Z$ para todas las DF de F entonces ρ cumple la PPDF

Aplicando el algoritmo

$HT \to R$

Ejecución

Aplicando el algoritmo

$HT \to R$

Ejecución

1. $Z = HT$

Aplicando el algoritmo

$HT \rightarrow R$

Ejecución

1. $Z = HT$
2. $Z = HT \cup ((HT \cap CSG)^{+} \cap CSG) = HT$

Aplicando el algoritmo

$HT \rightarrow R$

Ejecución

1. $Z = HT$
2. $Z = HT \cup ((HT \cap CSG)^{+} \cap CSG) = HT$
3. $Z = HT \cup ((HT \cap CT)^{+} \cap CT) = HT$

Aplicando el algoritmo

$HT \rightarrow R$

Ejecución

1. $Z = HT$
2. $Z = HT \cup ((HT \cap CSG)^{+} \cap CSG) = HT$
3. $Z = HT \cup ((HT \cap CT)^{+} \cap CT) = HT$
4. $Z = HT \cup ((HT \cap CHR)^{+} \cap CHR) = HT$

Aplicando el algoritmo

$HT \to R$

Ejecución

1. $Z = HT$
2. $Z = HT \cup ((HT \cap CSG)^+ \cap CSG) = HT$
3. $Z = HT \cup ((HT \cap CT)^+ \cap CT) = HT$
4. $Z = HT \cup ((HT \cap CHR)^+ \cap CHR) = HT$
5. $Z = HT \cup ((HT \cap CHS)^+ \cap CHS) = HT$

Aplicando el algoritmo

$HT \to R$

Ejecución

1. $Z = HT$
2. $Z = HT \cup ((HT \cap CSG)^+ \cap CSG) = HT$
3. $Z = HT \cup ((HT \cap CT)^+ \cap CT) = HT$
4. $Z = HT \cup ((HT \cap CHR)^+ \cap CHR) = HT$
5. $Z = HT \cup ((HT \cap CHS)^+ \cap CHS) = HT$

No se cumple la PPDF

Resumiendo

Algoritmo	FN	PLJ	PPDF
Algoritmo de 3FN	3FN	Sí (por Lema de Ullman)	Sí (Por construcción)
Algoritmo de BCFN	BCFN	Sí (por construcción)	No necesariamente

Resumiendo

Algoritmo	FN	PLJ	PPDF
Algoritmo de 3FN	3FN	Sí (por Lema de Ullman)	Sí (Por construcción)
Algoritmo de BCFN	BCFN	Sí (por construcción)	No necesariamente

El algoritmo de 3FN garantiza poder obtener un diseño correcto

Resumiendo

Más anomalías :(

Sean el esquema relacional $R(U, F)$, con

$$U = \{Persona, TipoEstablecimiento, EstablecimientoMásCercano\} \quad \text{y}$$

$$F = \{$$

$$Persona\ \ TipoEstablecimiento \rightarrow EstablecimientoMásCercano;$$

$$EstablecimientoMásCercano \rightarrow TipoEstablecimiento$$

$$\}$$

y una instancia de R:

Persona	**Tipo de Establecimiento**	**Establecimiento Más Cercano**
Claudia	Óptica	Almendares
Claudia	Peluquería	Luly Salón
Javier	Librería	Cuba Va
Alejandra	Panadería	La Cubana
Alejandra	Peluquería	Riudi Peluqueros
Alejandra	Óptica	Almendares

Algo que se tiene que hacer al menos una vez en la vida

Dado un esquema relacional $R(U, F)$, donde:

- $U = \{A, B, C, D, E\}$,
- la descomposición $\rho = (R_1, R_2, R_3, R_4, R_5)$,
- $R_1 = \{A, D\}, R_2 = \{A, B\}, R_3 = \{B, E\}, R_4 = \{C, D, E\}, R_5 = \{A, E\}$,
- y el conjunto de dependencias funcionales:

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

Verifique si se cumple PLJ aplicando el algoritmo correspondiente.

Esto también se tiene que hacer al menos una vez en la vida

Considere el conjunto de atributos $U = \{A, B, C, D\}$ con descomposición $\rho = \{AB, BC, CD\}$ y el conjunto de dependencias $F = \{A \to B,\ B \to C,\ C \to D,\ D \to A\}$. Verifique si se cumple la PPDF aplicando el algoritmo correspondiente.

Bases de Datos I

Teoría del diseño: PLJ, PPDF y BCFN

Lic. Andy Ledesma García
Lic. Víctor M. Cardentey Fundora
Dra. C. Lucina García Hernández

Departamento de Computación
Facultad de Matemática y Computación
Universidad de La Habana

11 de agosto de 2024