



Normalización

formas normales

Presentación



El proceso de normalización fue introducido por Codd en el año 1974. Allí se habían creado la primera forma normal (1FN), segunda forma normal (2FN) y tercera forma normal (3FN). Posteriormente Boyce y Codd crearon la forma normal de Boyce Codd (FNBC), haciendo más restrictiva la normalización.

Beneficios de la normalización

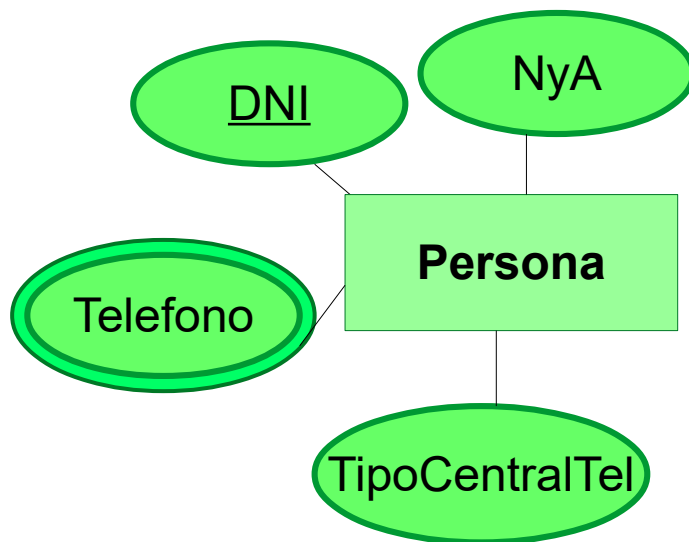


- *Menos espacio de almacenamiento*
- *Actualizaciones más rápidas*
- *Menor inconsistencia de datos*
- *Relaciones más claras*
- *Procedimientos más sencillos para la inserción de datos*
- *Estructura de datos flexible*

Un mal diseño puede ocasionar:

- *Redundancia*
- *Anomalías (inserción, actualización, eliminación)*

Introducción



1. ¿Cómo lo pasamos al modelo relacional?

Persona (DNI, NyA, TipoCentralTel)

Telefono (DNI, Nro)

2. ¿Cuáles son sus dependencias funcionales?

$F_{\text{persona}} = \{DNI \rightarrow NyA; \text{DNI} \rightarrow \text{TipoCentralTel}\}$

$F_{\text{telefono}} = \{DNI, Nro \rightarrow DNI, Nro\}$

3. ¿Qué sucede con la dependencia $DNI \rightarrow TipoCentralTel$? ¿Es correcta? Ó debería ser que el $Tel \rightarrow TipoCentralTel$?

4. ¿Y si cambiamos el diseño?

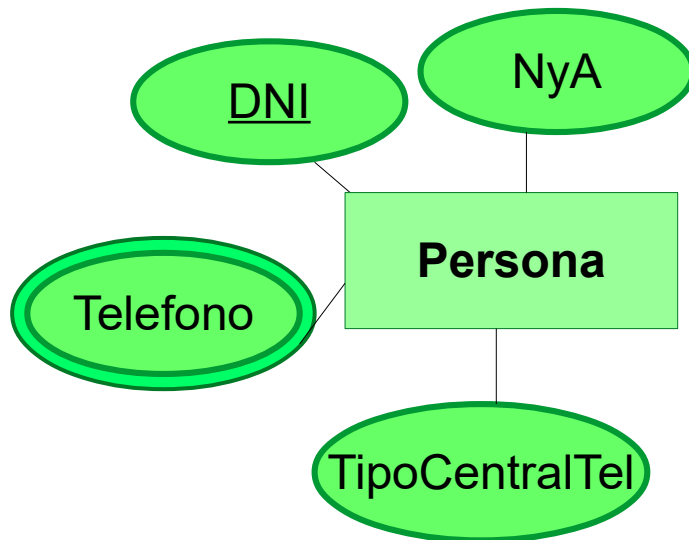
Persona (DNI, NyA) $F_p = \{DNI \rightarrow NyA\}$

Telefono (DNI, Nro, TipoCentralTel)

$F_t = \{DNI, Nro \rightarrow DNI, Nro, TipoCentralTel; Nro \rightarrow TipoCentralTel\}$

5. ¿Qué sucede ahora si varias personas poseen el mismo número de teléfono?

Introducción



6. ¿Cómo lo cambiaríamos para que no genere información redundante?

Persona (DNI, NyA)
 $F_p = \{DNI \rightarrow NyA\}$

Telefono (DNI, Nro)
 $F_t = \{DNI, Tel \rightarrow DNI, Tel\}$

TipoCentral (Nro, TipoCentralTel)
 $F_{tc} = \{Nro \rightarrow TipoCentralTel\}$

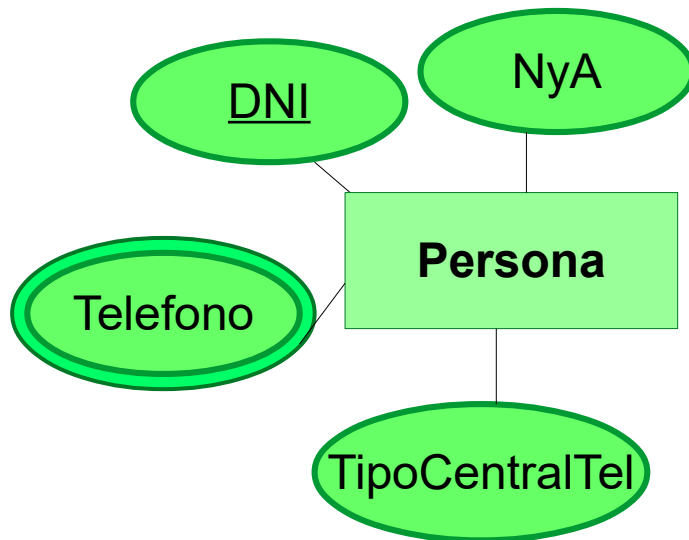
7. ¿Qué método empleamos para realizar este cambio?

DESCOMPOSICIÓN

8. ¿En qué nos basamos para realizar la descomposición?

En las dependencias funcionales.

Introducción



6. ¿Cómo lo cambiaríamos para que no genere información redundante?

Persona (DNI, NyA)
 $F_p = \{DNI \rightarrow NyA\}$

Telefono (DNI, Nro)
 $F_t = \{DNI, Tel \rightarrow DNI, Tel\}$

TipoCentral (Nro, TipoCentralTel)
 $F_{tc} = \{Nro \rightarrow TipoCentralTel\}$

7. ¿Qué método empleamos para realizar este cambio?

DESCOMPOSICIÓN

8. ¿En qué nos basamos para realizar la descomposición?

En las dependencias funcionales.



Formas Normales

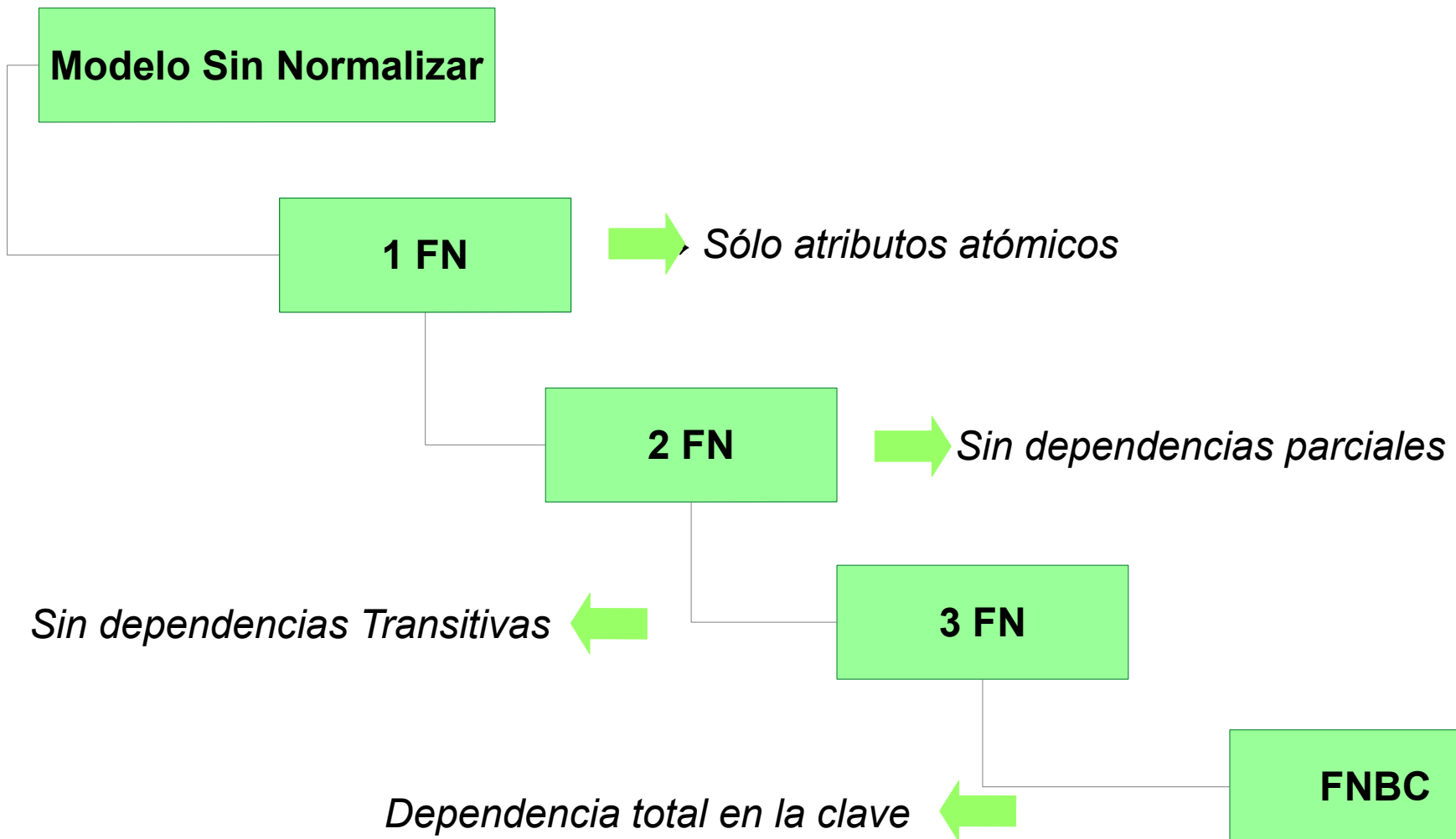
El proceso de la normalización se basa en la descomposición del esquema, utilizando las dependencias funcionales. Existen varias formas normales. Podemos adecuar nuestro modelo a cada una de ellas, sabiendo que a medida que aumentamos la forma normal, estaremos agregando más restricciones a nuestro modelo.



Formas Normales

El proceso de la normalización se basa en la descomposición del esquema, utilizando las dependencias funcionales. Existen varias formas normales. Podemos adecuar nuestro modelo a cada una de ellas, sabiendo que a medida que aumentamos la forma normal, estaremos agregando más restricciones a nuestro modelo.

Formas Normales





Formas Normales

Para toda Dependencia Funcional de R , tal que $X \rightarrow Y$:

1 FN

→ Sólo atributos atómicos

No deben existir atributos multivaluados

2 FN

→ Sin dependencias parciales

X es superlave ó
Y es primo ó
X no pertenece a ninguna clave

3 FN

→ Sin dependencias Transitivas

X es superlave ó
Y es primo

FNBC

→ Dependencia total en la clave

X es superlave

Identificar Formas Normales



1	R(A,B,C)	$F=\{A \rightarrow B\}$?
2	R(A,B,C)	$F=\{AB \rightarrow C, C \rightarrow B\}$?
3	R(A,B,C)	$F=\{A \rightarrow B, B \rightarrow C\}$?
4	R(A,B)	$F1=\{A \rightarrow B\}$ $F2=\{B \rightarrow A\}$ $F3=\{A \rightarrow B, B \rightarrow A\}$ $F4=\{\}$?



Descomposición

La descomposición de un modelo relacional, puede provocar:

- *Pérdida de Dependencias funcionales*
- *Pérdida de Información*

Pérdida de dependencias funcionales



Se dice que una descomposición conserva todas sus dependencias funcionales si:

$$F_d = \{ F_1 \cup F_2 \cup F_3 \dots \cup F_i \}^+ = F^+$$

Es decir, si se pueden obtener todas las dependencias funcionales originales, del conjunto de dependencias funcionales de la descomposición.

Pérdida de dependencias funcionales



Algoritmo:

$x \rightarrow y$

$r = x$

while (existan cambios en r) do
for each R_i in (descomposición)

$t = (r \cap R_i)^+ \cap R_i$

$r = r \cup t$



Si “ r ” posee todos los atributos “ y ”, entonces la dependencia se ha conservado

Pérdida de dependencias funcionales



En la verificación de las Dfs, podrían suceder dos casos:

1) $f \in (F1 \cup F2 \cup \dots F_i)$: que la dependencia funcional f se encuentre en las $U F_i$. Si esto sucede, ya sabemos que esa dependencia funcional, no se ha perdido.

2) $f \notin (F1 \cup F2 \cup \dots F_i)$: que la dependencia funcional de F no se encuentre en la $U F_i$. Entonces podría ser obtenida si utilizamos axiomas o reglas y si lo logramos, sería una dependencia funcional derivada. Con lo cual, tampoco se habría perdido. Si tampoco se puede derivar, concluimos que esa df se ha perdido.

Pérdida de dependencias funcionales



Veamos algunos ejemplos:

*$R (DNI, NyA, Domicilio, Mail)$
 $F = \{DNI \rightarrow NyA, Domicilio\}$*

Podríamos descomponer en:

*$R1 (DNI, NyA, Domicilio) F1 = \{DNI \rightarrow NyA, Domicilio\}$
 $R2 (DNI, Mail) F2 = \{ \}$*

$(F1 \cup F2)^+ = F^+ \rightarrow$ Se preservaron las dependencias



Pérdida de dependencias funcionales

$R (\text{DNI}, \text{NyA}, \text{Empresa}, \text{MailCorporativo})$

$F = \{\text{DNI} \rightarrow \text{NyA}, \text{Empresa}; \text{Empresa} \rightarrow \text{MailCorporativo}; \text{DNI} \rightarrow \text{MailCoporativo}\}$

Podríamos descomponer en:

$R1 (\underline{\text{DNI}}, \text{NyA}, \text{Empresa})$ $F1 = \{\text{DNI} \rightarrow \text{NyA}, \text{Empresa}\}$

$R2 (\text{Empresa}, \underline{\text{MailCoporativo}})$ $F2 = \{\text{Empresa} \rightarrow \text{MailCoporativo}\}$

$(F1 \cup F2) = \{\text{DNI} \rightarrow \text{NyA}, \text{Empresa}; \text{Empresa} \rightarrow \text{MailCoporativo}\}$

¿Falta $\text{DNI} \rightarrow \text{MailCoporativo}$? A través de la transtividad la dependencia se podría obtener. Por lo cual, también se conservan las dependencias funcionales. Se cumple que:

$$\{F1 \cup F2\}^+ = F^+$$

$\text{DNI}^+(\{F1 \cup F2\}) = \{\text{DNI}, \text{NyA}, \text{Empresa}, \text{MailCoporativo}\}$

$\text{DNI}^+(F) = \{\text{DNI}, \text{NyA}, \text{Empresa}, \text{MailCoporativo}\}$



Pérdida de dependencias funcionales

$R (\text{DNI}, \text{NyA}, \text{Empresa}, \text{MailCorporativo})$

$F = \{\text{DNI} \rightarrow \text{NyA}, \text{Empresa}; \text{Empresa} \rightarrow \text{MailCorporativo}; \text{DNI} \rightarrow \text{MailCoporativo}\}$

Podríamos descomponer en:

$R1 (\underline{\text{DNI}}, \text{NyA}, \text{Empresa})$ $F1 = \{\text{DNI} \rightarrow \text{NyA}, \text{Empresa}\}$

$R2 (\text{Empresa}, \text{MailCoporativo})$ $F2 = \{\text{Empresa} \rightarrow \text{MailCoporativo}\}$

$(F1 \cup F2) = \{\text{DNI} \rightarrow \text{NyA}, \text{Empresa}; \text{Empresa} \rightarrow \text{MailCoporativo}\}$

¿Falta $\text{DNI} \rightarrow \text{MailCoporativo}$? A través de la transtividad la dependencia se podría obtener. Por lo cual, también se conservan las dependencias funcionales. Se cumple que:

$$\{F1 \cup F2\}^+ = F^+$$

$\text{DNI}^+(\{F1 \cup F2\}) = \{\text{DNI}, \text{NyA}, \text{Empresa}, \text{MailCoporativo}\}$

$\text{DNI}^+(F) = \{\text{DNI}, \text{NyA}, \text{Empresa}, \text{MailCoporativo}\}$



Pérdida de dependencias funcionales

$R(A, B, C)$

$F = \{A \rightarrow B; B \rightarrow C; C \rightarrow B; A \rightarrow C\}$

Podríamos descomponer en:

$R_1(A, C) \quad F_1 = \{A \rightarrow C\}$

$R_2(B, C) \quad F_2 = \{B \rightarrow C; C \rightarrow B\}$

$(F_1 \cup F_2) = \{A \rightarrow C; B \rightarrow C; C \rightarrow B\}$

¿Falta $A \rightarrow B$ y $A \rightarrow C$? ¿Se perdieron Dfs?

No, porque esas dependencias se pueden derivar de las dependencias de F_1 y F_2 .

$A^{+}_{(F_1 \cup F_2)} = \{A, C, B\} \quad / A \rightarrow B? A \rightarrow C \text{ y } C \rightarrow B \Rightarrow A \rightarrow B \text{ (x transitiv.)}$

$A^{+}_{(F)} = \{A, C, B\}$



Pérdida de dependencias funcionales

Dado:

$R(A, B, C)$

$F = \{A \rightarrow C; BC \rightarrow A\}$

Podríamos descomponer en:

$R_1(A, C) \quad F_1 = \{A \rightarrow C\}$

$R_2(A, B) \quad F_2 = \{\}$

$(F_1 \cup F_2) = \{A \rightarrow C\}$

¿Falta $BC \rightarrow A$? ¿Se perdieron Dfs?

Sí, porque esas dependencias No se pueden derivar de las dependencias de F_1 y F_2 .

$BC^+(F_1 \cup F_2) = \{B, C\}$

$BC^+(F) = \{B, C, A\}$



Pérdida de Información

Se dice que una descomposición posee pérdida de información, cuando no se puede reconstruir la tupla original a partir de la tupla de la cada R_i

$$r = r1 \mid X \mid r2 \mid X \mid r3 \dots \mid X \mid r_i \quad / \quad r_i = \pi_{R_i}(r)$$

En toda descomposición, cada tupla debería poder reconstruirse. Si no pudiera reconstruirse se dice que esa descomposición es inválida.



Pérdida de Información

Veamos algunos ejemplos:

R (DNI, NyA, Empresa, MailCorp)

$F = \{DNI \rightarrow NyA, Empresa; Empresa \rightarrow MailCorp\}$

Podríamos descomponer en:

R_1 (DNI, NyA, Empresa) $F_1 = \{DNI \rightarrow NyA, Empresa\}$

R_2 (Empresa, MailCorp) $F_2 = \{Empresa \rightarrow MailCorp\}$

DNI	NyA	Empresa	MailCorp
1234	Juan	E1 S.A.	e1@e1
9999	Ana	E2 S.A.	e2@e2
5656	Lola	E1 S.A.	e1@e1

DNI	NyA	Empresa
1234	Juan	E1 S.A.
9999	Ana	E2 S.A.
5656	Lola	E1 S.A.

Empresa	MailCorp
E1 S.A.	e1@e1
E2 S.A.	e2@e2

$R(1234, \text{Juan}, \text{E1 S.A.}, \text{e1@e1}) = R_1(1234, \text{Juan}, \text{E1 S.A.}) \mid X \mid R_2(\text{E1 S.A.}, \text{e1@e1})$

No hubo pérdida de Información



Pérdida de Información

Veamos algunos ejemplos:

$R(\text{DNI}, \text{NyA}, \text{Empresa}, \text{MailCorp})$

$F = \{\text{DNI} \rightarrow \text{NyA}, \text{Empresa}; \text{Empresa} \rightarrow \text{MailCorp}\}$

Podríamos descomponer en:

$R_1(\text{DNI}, \text{NyA}, \text{Empresa}) \quad F_1 = \{\text{DNI} \rightarrow \text{NyA}, \text{Empresa}\}$

$R_2(\text{MailCorp}) \quad F_2 = \{ \}$

DNI	NyA	Empresa	MailCorp
1234	Juan	E1 S.A.	e1@e1
9999	Ana	E2 S.A.	e2@e2
5656	Lola	E1 S.A.	e1@e1

DNI	NyA	Empresa
1234	Juan	E1 S.A.
9999	Ana	E2 S.A.
5656	Lola	E1 S.A.

MailCorp
e1@e1
e2@e2

$R(1234, \text{Juan}, \text{E1 S.A.}, \text{e1@e1}) \quad \Leftrightarrow \quad R_1(1234, \text{Juan}, \text{E1 S.A.}) \mid X \mid R_2(\text{e1@e1}) = \{ \}$

Hubo pérdida de Información

Algoritmos de Descomposición



Existen dos algoritmos que nos permiten realizar la descomposición de una relación:

- *Algoritmo de 3FN:*
 - *No existe pérdida de dependencias funcionales*
 - *No existe pérdida de información*
- *Algoritmo de FNBC:*
 - *Puede o no existir pérdida de dependencias funcionales*
 - *No existe pérdida de información*



Algoritmo de 3FN

- 1) Calcular F_{min}
- 2) Para cada miembro izquierdo X que aparezca en F_{min} , crear un esquema de relación $(XUA_1UA_2...UA_n)$ donde $X \rightarrow A_1, X \rightarrow A_2, .. X \rightarrow A_n$ sean todas D_f de F_{min}
- 3) Colocar los atributos restantes, no incluidos en paso 2, en un sólo esquema de relación
- 4) Si ninguno de los esquemas resultantes contiene una clave candidata de R , crear un esquema adicional que contenga una clave de R .



Ejemplo Algoritmo de 3FN

R(A,B,C,D,E,F,G)

F={A → BC, AD → G, AC → E, C → A, B → C, F → B}

C.C. = {DF}

1 FN

1) Calcular Fmin

Fmin = {A → B, AD → G, A → E, C → A, B → C, F → B}

2) Para cada miembro izquierdo X que aparezca en Fmin, crear un esquema de relación (XUA1UA2....UAN) donde $X \rightarrow A1, X \rightarrow A2, \dots, X \rightarrow An$ sean todas Df de Fmin

R1 (A,B,E) F1={A → B, A → E}

R2 (A,D,G) F2={AD → G}

R3 (A,C) F3={C → A}

R4 (B,C) F4={B → C}

R5 (B,F) F5={F → B}

3) Colocar los atributos restantes, no incluidos en paso 2, en un sólo esquema de relación

No agrega ningún esquema

4) Si ninguno de los esquemas resultantes contiene una clave candidata de R, crear un esquema adicional que contenga una clave de R.

Agrega **R6 (D,F) F6={}**



Ejemplo Algoritmo de 3FN

$R(C,P,D,I,K,N)$
 $F=\{PC \rightarrow D, PC \rightarrow I, DI \rightarrow K, K \rightarrow I\}$

C.C. = {PCN}
1 FN

1) Calcular F_{min}

$F_{min} = \{PC \rightarrow D, PC \rightarrow I, DI \rightarrow K, K \rightarrow I\}$

2) Para cada miembro izquierdo X que aparezca en F_{min} , crear un esquema de relación ($XUA_1UA_2...UA_n$) donde $X \rightarrow A_1, X \rightarrow A_2, .. X \rightarrow A_n$ sean todas D_f de F_{min}

$R_1(P,C,D,I)$ **$F_1=\{PC \rightarrow D, PC \rightarrow I\}$**
 $R_2(D,I,K)$ **$F_2=\{DI \rightarrow K, K \rightarrow I\}$**

$CC_1=\{PC\}$
 $CC_2=\{DI,DK\}$

3) Colocar los atributos restantes, no incluidos en paso 2, en un sólo esquema de relación

Agrega **$R_3(N)$** **$F_3=\{\}$** **$CC_3=\{N\}$**

4) Si ninguno de los esquemas resultantes contiene una clave candidata de R , crear un esquema adicional que contenga una clave de R .

Agrega **$R_4(P,C,N)$** **$F_4=\{\}$** **$CC_4=\{PCN\}$**



Algoritmo de FNBC

1) Para cada DF $X \rightarrow Y$ en F , que no sea trivial y que viole las condiciones de FNBC, hacer 2 esquemas:

$$R1 = (X \cup Y) \text{ y } R2 = (R - Y)$$

Para alguna de las dependencias funcionales de F , no pudiera incluirse en alguno de los subesquemas, esa Dfs podría perderse. Para evitar la pérdida de Dfs se podría inferir alguna Df equivalente que pueda ir en $F1$ ó $F2$.

2) Si los subesquemas resultantes siguen sin cumplir las condiciones de FNBC, volver al punto 1.



Ejemplo de Algoritmo de FNBC

$R(M, N, O, P, Q, S)$

$F = \{M \rightarrow ON, N \rightarrow MO, O \rightarrow MN, MQ \rightarrow SO, OP \rightarrow Q\}$

$C.C. = \{PM, PN, PO\}$

2 FN

Si bien no es obligatorio, se calcula el F_{min} para comenzar por el menor F posible:

$F_{min} = \{M \rightarrow N, N \rightarrow O, O \rightarrow M, MQ \rightarrow S, OP \rightarrow Q\}$

1) Para cada DF $X \rightarrow Y$ en F , que no sea trivial y que viole las condiciones de FNBC, hacer 2 esquemas: $R_1 = (X \cup Y)$ y $R_2 = (R - Y)$

$F_{min} = \{M \rightarrow N, N \rightarrow O, O \rightarrow M, MQ \rightarrow S, OP \rightarrow Q\}$

$R_1(M, N)$
 $F_1 = \{M \rightarrow N\}$
 $CC_1 = \{M\}$

$R_2(M, O, P, Q, S)$
 $F_2 = \{O \rightarrow M, MQ \rightarrow S, OP \rightarrow Q\}$
 $CC_2 = \{PO\}$

$R_3(O, M)$
 $F_3 = \{O \rightarrow M\}$
 $CC_3 = \{O\}$

$R_4(O, P, Q, S)$
 $F_4 = \{OP \rightarrow Q, OQ \rightarrow S\}$
 $CC_4 = \{PO\}$

$R_5(O, Q, S)$
 $F_5 = \{OQ \rightarrow S\}$
 $CC_5 = \{OQ\}$

$R_6(O, P, Q)$
 $F_6 = \{OP \rightarrow Q\}$
 $CC_6 = \{OP\}$

$R_1 \cup R_3 \cup R_5 \cup R_6$

Algoritmo de pérdida de Información



- *Método Tableau*
- *Teorema de Health*



Método Tableau

Dado un esquema $R(A_1, A_2, \dots, A_n)$ y $D = \{R_1, R_2 \dots R_n\}$, armar una grilla con lo siguiente:

- 1) Generar una columna por cada atributo de R
- 2) Generar una fila por cada relación de la descomposición R_i
- 3) En cada celda completar con a_j o b_{ij} , dependiendo:
 - a_j si $A \in R_i$
 - b_{ij} si $A \notin R_i$
- 4) Para cada $df\ X \rightarrow A$ de F , verificar si se cumple en la grilla. Si se cumple, se deben cambiar los valores de a_j ó b_{ij} .
- 5) El procedimiento finalizará cuando se haya iterado y no haya cambios en el tableau.
- 6) Si encontramos una fila que posea todas " a_i ", esto significará que no se perdió información. De lo contrario, la descomposición sería incorrecta.



Método Tableau: Ejemplo

$R(A,B,C,D,E) / F=\{C \rightarrow E, E \rightarrow D, B \rightarrow E, AD \rightarrow B\}$
 $R1(A,B,C) \quad R2(C,D) \quad R3(A,D,E)$

Tableau:

Paso1: Armar la grilla

	A	B	C	D	E
R1	a1	a2	a3	b14	b15
R2	b21	b22	a3	a4	b25
R3	a1	b32	b33	a4	a5

Paso 2: Iterar por las dfs

$C \rightarrow E$

	A	B	C	D	E
R1	a1	a2	a3	b14	b15
R2	b21	b22	a3	a4	b25 b15
R3	a1	b32	b33	a4	a5



Método Tableau: Ejemplo

$R(A,B,C,D,E) / F = \{C \rightarrow E, E \rightarrow D, AD \rightarrow B, B \rightarrow E\}$

$R1(A,B,C) \quad R2(C,D) \quad R3(A,D,E)$

Paso 2: Iterar por las dfs

$E \rightarrow D$

	A	B	C	D	E
R1	a1	a2	a3	b14 a4	b15
R2	b21	b22	a3	a4	b25 b15
R3	a1	b32	b33	a4	a5

$AD \rightarrow B$

	A	B	C	D	E
R1	a1	a2	a3	b14 a4	b15
R2	b21	b22	a3	a4	b25 b15
R3	a1	b32 a2	b33	a4	a5

$B \rightarrow E$

	A	B	C	D	E
R1	a1	a2	a3	b14 a4	b15 a5
R2	b21	b22	a3	a4	b25 b15
R3	a1	b32 a2	b33	a4	a5



Método Tableau: Ejemplo

- $R(A,B,C) F=\{AC \rightarrow B, B \rightarrow C\}$

FNBC

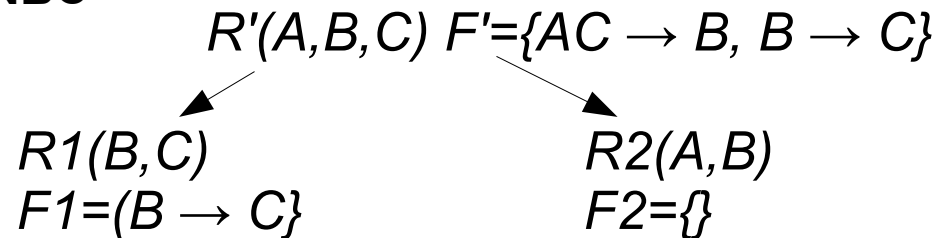


Tableau:

Paso1: Armar la grilla

	A	B	C
R1	b11	a2	a3
R2	a2	a2	b23

Paso 2: Iterar por las dfs $F1 \cup F2$

	A	B	C
R1	b11	a2	a3
R2	a2	a2	b23 a3

Como se encontró una fila con todas a, se dice que NO hay pérdida de Información



Teorema de Health

- *Permite determinar si una descomposición posee o no pérdida de información, pero sólo se podrá utilizar si la relación posee dos subesquemas.*

$$\begin{aligned} & \text{La DF } (R1 \cap R2) \rightarrow (R1 - R2) \in F^+ \\ & \text{La DF } (R1 \cap R2) \rightarrow (R2 - R1) \in F^+ \end{aligned} \quad \text{ó}$$

- **Ejemplo:**

$R(Y,Z,W,X) \ F=\{Y \rightarrow Z, W \rightarrow X\}$

$R1(Y, Z, W) \ F1=\{Y \rightarrow Z\} \ / \ R2(W, X) \ F2=\{W \rightarrow X\}$

$$(R1 \cap R2) \rightarrow (R1 - R2) \qquad W \rightarrow YZ \qquad \notin F^+$$

$$(R1 \cap R2) \rightarrow (R2 - R1) \qquad W \rightarrow X \qquad \in F^+$$



No existe pérdida de Información

Consultas

