Apunte Teórico de Dependencias Funcionales y Normalización

El presente apunte es un resumen de algunos conceptos dados en las Unidades de Dependencias Funcionales y Normalización, para mayor información **consulten la bibliografía** recomendada por la cátedra.

DEFINICION DE DEPENDENCIA FUNCIONAL

Una dependencia funcional es una restricción entre dos conjuntos de atributos de una relación.

Sea R un esquema de relación, sea r una instancia cualquiera de R y sean X e Y dos subconjuntos de R. La dependencia funcional **X → Y** (léase X determina a Y) se cumple si y sólo si para cualesquiera dos tuplas t1 y t2 de r, tales que:

t1[X] = t2[X], entonces necesariamente t1[Y] = t2[Y]

Es decir, si dos tuplas cualesquiera tienen igual valor en X, deben tener igual valor en Y, para que se cumpla la dependencia funcional.

Esto significa que los valores componentes de Y de una tupla de r dependen de los valores del componente X, o están determinados por ellos; o bien, que los valores del componente X de una tupla determinan de manera única los valores del componente Y.

Ejemplo:

Examen (DNI, Nombre, Apellido, CodMateria, NombreMateria, FechaExamen, Nota)

Se cumplen las siguientes dependencias funcionales:

F= { DNI → Nombre, Apellido ;
CodMateria → NombreMateria ;
DNI, CodMateria, FechaExamen → Nota }

A fin de comprender mejor el concepto, podemos trazar un paralelismo entre las dependencias funcionales y el concepto de una función matemática del tipo f(x)=y. Recordemos que según la definición de función para un valor determinado de X la función debe retornar un único valor de Y. Puede darse el caso que un mismo valor de Y pueda ser obtenido con distintos valores de X, pero no se admite que ocurra a la inversa.

f(x) = y

f(2) = 3
f(5) = 3
Esto es admitido en una función.

f(2) = 3
f(2) = 5
Esto NO es admitido en una función.

X → Y

X	Y
2	3
5	3

Esto es admitido en la dependencia funcional dada.

X	Y
2	3
2	5

Esto NO es admitido en la dependencia funcional dada.

Dependencia Funcional Trivial
Decimos que una Dependencia es Trivial cuando es obvia, por ejemplo X → X.

CLAVE Y SUPERCLAVE

Dado un esquema R(A1, A2, ... An) y un conjunto de dependencias funcionales F asociado, se dice que X $\subseteq$ R es una CLAVE para el esquema R si se cumplen las siguientes dos condiciones:

1) X determina a todos los atributos de R, X → A1, A2, … An

2) No existe ningún Z (subconjunto de X) que determine a todos los atributos de R (condición de minimalidad).

Por otra parte, decimos que Y es superclave de R si cumple la condición 1.

Se cumple, que toda clave es superclave, pero no a la inversa.

Ejemplo:

Empleado (Legajo, Nombre, DNI)

Claves Candidatas = { Legajo ;
DNI }

Superclaves = { Legajo, Nombre ;
DNI, Nombre ;
Legajo, DNI ;
Legajo, Nombre, DNI ;
Legajo ;
DNI }

AXIOMAS DE ARMSTRONG

Reflexividad

Si Y $\subseteq$ X => se cumple X→Y

Aumento

Dada X→Y se puede inferir XZ→YZ

Transitividad

Dadas X→Y y Y→Z se puede inferir X→Z

REGLAS ADICIONALES/DERIVADAS

Descomposición

Dada X→YZ se pueden inferir X→Y y X→Z

Unión

Dadas X→Y y X→Z se puede inferir X→YZ

Pseudotransitividad

Dadas X→Y y WY→Z se puede inferir WX→Z

DEMOSTRACIONES

Cada uno de los Axiomas de Armstrong puede ser demostrado en base a la definición de dependencia funcional:

Demostración de la Reflexividad:

Supongamos que $Y \subseteq X$ y que existen dos tuplas t1 y t2 en algún ejemplar de la relación r de R tales que t1[X]=t2[X]. Entonces t1[Y]=t2[Y] porque $Y \subseteq X$; por lo tanto, debe cumplirse $X \rightarrow Y$ en r.

Demostración del Aumento (por contradicción):

Supongamos que $X \rightarrow Y$ se cumple en algún ejemplar r de R, pero que $XZ \rightarrow YZ$ no se cumple. En tal caso, deben existir dos tuplas t1 y t2 en r tales que (1) t1[X]=t2[X], (2) t1[Y]=t2[Y], (3) t1[XZ]=t2[XZ] y (4) t1[YZ]<>t2[YZ]. Esto no es posible porque a partir de (1) y (3) se deduce (5) t1[Z]=t2[Z], y a partir de (2) y (5) se deduce (6) t1[YZ]=t2[YZ], lo que contradice (4).

Demostración de la Transitividad:

Supongamos que se cumplen (1) $X \rightarrow Y$ y (2) $Y \rightarrow Z$ en una relación r. Entonces, para cualesquier dos tuplas t1 y t2 en r tales que t1[X]=t2[X], debemos tener (3) t1[Y]=t2[Y] (por la suposición de (1)), y por tanto también debemos tener (4) t1[Z]=t2[Z], (por (3) y a partir de la suposición (2)); por lo tanto, se debe cumplir $X \rightarrow Z$ en r.

Cada una de las reglas adicionales puede ser demostrada a partir de los Axiomas de Armstrong:

Demostración de la Descomposición:

1) $X \rightarrow YZ$ Dada
2) $YZ \rightarrow Y$ por Reflexividad (donde $Y \subseteq YZ$)
3) $X \rightarrow Y$ Transitividad entre 1 y 2

Demostración de la Unión:

1) $X \rightarrow Y$ Dada
2) $X \rightarrow Z$ Dada
3) $X \rightarrow XY$ Aumento con X en 1 (XX=X)
4) $XY \rightarrow YZ$ Aumento con Y en 2
5) $X \rightarrow YZ$ Transitividad entre 3 y 4

Demostración de la Pseudotransitividad:

1) $X \rightarrow Y$ Dada
2) $WX \rightarrow WY$ Aumento con W en 1
3) $WY \rightarrow Z$ Dada
4) $WX \rightarrow Z$ Transitividad entre 3 y 2

CLAUSURA DE UN CONJUNTO DE DEPENDENCIAS

Dado un conjunto de DF llamado F, denominamos clausura de F (F^+) al conjunto de **todas las dependencias posibles** que se pueden inferir de F.
Por ejemplo:

R(A, B, C) F={ A→B, BC→A }

F^+={ A→A,
B→B,
C→C,
AB→AB, AB→A, AB→B,
AC→AC, AC→A, AC→C,
BC→BC, BC→B, BC→C,
ABC→ABC, ABC→AB, ABC→AC, ABC→BC, ABC→A, ABC→B, ABC→C,
A→**B**, A→AB, AC→B, AC→AB, AC→BC, AC→ABC,
BC→**A**, BC→AB, BC→AC, BC→ABC
}

Triviales (obvias)

Inferidas

<u>Nota</u>: Las DF son relaciones del tipo {conjunto de atributos}→{conjunto de atributos}, por lo tanto, el orden de los atributos no importa (AB→A es lo mismo que BA→A).

CLAUSURA DE UN CONJUNTO DE ATRIBUTOS

Sea X un conjunto de atributos del esquema de relación R, y F un conjunto de DF que se cumplen en R, llamamos Clausura de X en F (X^+_F) al conjunto de todos los atributos determinados funcionalmente por X.
Ejemplo:

R(A, B, C) F = { A→B, BC→A }

A^+_F = AB
B^+_F = B
C^+_F = C
AB^+_F = AB
AC^+_F = ABC
BC^+_F = ABC
ABC^+_F = ABC

Con esto podemos calcular fácilmente el conjunto F^+ (haciendo todas las combinaciones)

Algoritmo para calcular X^+

```
X+ : = X;
viejoX+ := {};

Mientras X+<>viejoX+ hacer
        viejoX+ := X+;
        Para cada dependencia Y→Z en F hacer
                Si Y ⊆ X+ entonces X+ := X+ U Z
Fin Mientras
```

EQUIVALENCIA ENTRE DOS CONJUNTOS DE DEPENDENCIAS FUNCIONALES

Dos conjuntos de dependencias funcionales F y G son equivalentes cuando sus clausuras son iguales:

$F \equiv G$ (F es equivalente a G) <=> $F^+ = G^+$

Para probar que dos conjuntos F y G son equivalentes, sin necesidad de calcular F^+ ni G^+, se debería probar que toda dependencia de F puede inferirse en G ($F \subseteq G^+$), y recíprocamente, toda dependencia de G se puede inferir en F ($G \subseteq F^+$).

$$F^+ = G^+ \iff F \subseteq G^+ \text{ y } G \subseteq F^+$$

Si se cumple que $F \subseteq G^+$ decimos que "G cubre a F" y si se cumple que $G \subseteq F^+$ decimos que "F cubre a G".

Podemos determinar si F cubre a G calculando X^+_F para cada dependencia $X \rightarrow Y$ de G, y comprobando que $Y \subseteq X^+_F$

Veamos un ejemplo:

R (A, B, C, D)
$F = \{ A \rightarrow B, A \rightarrow C, B \rightarrow A, D \rightarrow A \}$
$G = \{ D \rightarrow ABC, B \rightarrow A, B \rightarrow C, A \rightarrow B \}$

¿Son F y G equivalentes?

$$F \overset{?}{\equiv} G$$

$$\downarrow$$

$$F^+ \overset{?}{=} G^+$$

$$F \overset{?}{\subseteq} G^+ \qquad\qquad G \overset{?}{\subseteq} F^+$$

$A^+_G = A\mathbf{BC}$	$D^+_F = D\mathbf{ABC}$
$B^+_G = B\mathbf{A}C$	$B^+_F = B\mathbf{AC}$
$D^+_G = D\mathbf{A}BC$	$A^+_F = A\mathbf{B}C$

Respuesta: Sí, F y G son equivalentes.

Conjunto Mínimo de Dependencias Funcionales (Fmin)

Un conjunto de dependencias funcionales es Mínimo si cumple las siguientes tres condiciones:

1) Todas sus dependencias funcionales tienen un solo atributo en su parte derecha (determinado).
2) No podemos reemplazar ninguna dependencia funcional $X \rightarrow A$ por otra $Y \rightarrow A$, donde Y C X, y seguir teniendo un conjunto equivalente.
3) No podemos quitarle ninguna dependencia funcional y seguir teniendo un conjunto equivalente.

A Fmin, también se lo llama "Cobertura minimal" y es un conjunto de dependencias funcionales sin redundancias.

Un conjunto F puede tener varias coberturas minimales.

Algoritmo para el Cálculo de Fmin

Dado un conjunto F, realizar los siguientes tres pasos:

Paso 1) Descomponer parte derecha
Reemplazar cada DF $X \rightarrow A_1, A_2, .., A_N$ por N dependencias del tipo $X \rightarrow A_1, X \rightarrow A_2, .., X \rightarrow A_N$

Llamaremos F_1 al conjunto resultante luego de aplicar este primer paso.

Paso 2) Eliminar redundancia en parte izquierda
Tomar las DF con más de un atributo en la parte izquierda y hacer:

Por cada dependencia $W \rightarrow Y$,
para cada subconjunto A que esté incluido W,
calcular $(W\text{-}A)^+$ en F1
si $(W\text{-}A)^+_{F_1}$ contiene a Y => reemplazar $W \rightarrow Y$ por $(W\text{-}A) \rightarrow Y$

Llamaremos F_2 al conjunto resultante luego de aplicar este segundo paso.

Paso 3) Eliminar dependencias redundantes
Definir $F_3=F_2$
Por cada dependencia $X \rightarrow Y$ de F_3
Calcular X^+ en $F_3 - \{X \rightarrow Y\}$,
Si X^+ contiene a Y => eliminar $X \rightarrow Y$ de F_3

El conjunto obtenido luego de aplicar los tres pasos será Fmin.

Veamos un ejemplo:

R(ABCDE) $F=\{A \rightarrow BCD, AB \rightarrow DE, BE \rightarrow AC\}$

Paso 1)
$F_1 = \{ A \rightarrow B, A \rightarrow C, A \rightarrow D, AB \rightarrow D, AB \rightarrow E, BE \rightarrow A, BE \rightarrow C \}$

Paso 2)

$A^+_{F_1} = ABC\underline{\mathbf{D}}E$

$AB \rightarrow D$ => reemplazar por $A \rightarrow D$

$B^+_{F_1} = B$

$A^+_{F_1} = ABCD\underline{\mathbf{E}}$

$AB \rightarrow E$ => reemplazar por $A \rightarrow E$

$B^+_{F_1} = B$

$B^+_{F_1} = B$

$BE \rightarrow A$

$E^+_{F_1} = E$

$B^+_{F_1} = B$

$BE \rightarrow C$

$E^+_{F_1} = E$

$F_2 = \{ A \rightarrow B, A \rightarrow C, A \rightarrow D, A \rightarrow E, BE \rightarrow A, BE \rightarrow C \}$

Paso 3)

$A \rightarrow B$	$A^+_{F_2 - \{A \rightarrow B\}} = ACDE$	no es redundante
$A \rightarrow C$	$A^+_{F_2 - \{A \rightarrow C\}} = AB\underline{\mathbf{C}}DE$	$A \rightarrow C$ es Redundante
$A \rightarrow D$	$A^+_{F_2 - \{A \rightarrow C\} - \{A \rightarrow D\}} = ABEC$	no es redundante
$A \rightarrow E$	$A^+_{F_2 - \{A \rightarrow C\} - \{A \rightarrow E\}} = ABD$	no es redundante
$BE \rightarrow A$	$BE^+_{F_2 - \{A \rightarrow C\} - \{BE \rightarrow A\}} = BEC$	no es redundante
$BE \rightarrow C$	$BE^+_{F_2 - \{A \rightarrow C\} - \{BE \rightarrow C\}} = BEAD$	no es redundante

$F_3 = \{ A \rightarrow B, A \rightarrow D, A \rightarrow E, BE \rightarrow A, BE \rightarrow C \}$

$$Fmin = F_3$$

$$\boxed{Fmin = \{ A \rightarrow B, A \rightarrow D, A \rightarrow E, BE \rightarrow A, BE \rightarrow C \}}$$

FORMAS NORMALES

Las siguientes son “definiciones prácticas” creadas como simplificaciones de las definiciones teóricas que aparecen en la bibliografía. Si bien son equivalentes, se recomienda consultar la bibliografía (o la siguiente página del presente apunte) para conocer las definiciones formales de cada forma normal.

Primera Forma Normal

1FN

Atributos con valores atómicos (o indivisibles).
Esta Forma Normal no admite los atributos multivaluados, los atributos compuestos y sus combinaciones.

Segunda Forma Normal

2FN

$\forall\ X \rightarrow Y,$
- Si Y es un atributo no primo, entonces X no debe ser un subconjunto de alguna CC (*) (es decir, Y no debe depender en forma parcial de alguna CC)
- o
- Y es primo

(*) Decir que X no debe ser primo, no sería estrictamente correcto, porque primos son los atributos, no los subconjuntos.

Tercera Forma Normal

3FN

$\forall\ X \rightarrow Y,$
- X es Superclave
- o
- Y es primo

Forma Normal de Boyce - Codd

FNBC $\forall\ X \rightarrow Y,$ X es Superclave

IMPORTANTE: Todas las Formas Normales aceptan DF X→Y donde Y $\subseteq$ X (trivial). Por eso, lo mejor es usar Fmin al momento de evaluar en qué forma normal se encuentra un esquema dado, ya que Fmin no contiene DF triviales.

Atributo Primo

Un atributo del esquema de relación R se denomina atributo primo de R si es miembro de cualquier clave candidata de R.

Por ejemplo:

EMPLEADO (<u>tipo_doc</u>, <u>nro_doc</u>, nombre, apellido, fecha_nacimiento)

En este caso, los atributos tipo_doc y nro_doc son primos. Los restantes atributos son no primos.

Definiciones teóricas de las Formas Normales

Primera Forma Normal
Un esquema de relación se encuentra en primera forma normal si todos sus atributos admiten únicamente valores atómicos (indivisibles).
La primera forma normal no acepta atributos compuestos ni atributos multivaluados.

Segunda Forma Normal
Se basa en el concepto de <u>dependencia parcial</u>.

- Una DF X→Y es una **DF total** si la eliminación de cualquier atributo A de X hace que la dependencia deje de ser válida.
- Una DF X→Y es una **DF parcial** si es posible eliminar un atributo A perteneciente a X de X y la dependencia sigue siendo válida, es decir, se cumple {X-A}→Y.

Ejemplo:

EMPLEADO (<u>tipo_doc</u>, <u>nro_doc</u>, nombre, apellido, salario)

tipo_doc, nro_doc, apellido → salario
Es una DF parcial (ya que se puede quitar el atributo apellido del determinante y la DF sigue siendo válida).

tipo_doc, nro_doc → salario
Es una DF total (ya que si se quita alguno de los dos atributos del determinante, la DF deja de ser válida).

Un esquema de relación R cumple 2FN si ningún atributo no primo de R depende parcialmente de cualquier clave candidata de R.

Ejemplo:

ASIGNADO_A (<u>legajo_empleado</u>, <u>cod_proyecto</u>, fecha_desde, nombre_proyecto)

El esquema de relación ASIGNADO_A no está en 2FN porque el atributo nombre_proyecto no es primo y depende parcialmente de la clave.

cod_proyecto → nombre_proyecto

Tercera Forma Normal
Se basa en el concepto de <u>dependencia transitiva</u>.

Una DF X→Y en un esquema de relación R es una **dependencia transitiva** si existe un conjunto de atributos Z que no sea un subconjunto de cualquier clave candidata de R y se cumplen tanto X→Z como Z→Y.

Ejemplo:

EMPLEADO (<u>legajo</u>, apellido, nombre, calle, nro, CP, ciudad, provincia)

Se cumplen:
legajo → CP y CP → ciudad, provincia
Luego, la DF legajo → ciudad es una DF transitiva.

Un esquema de relación R cumple 3FN si cumple 2FN y ningún atributo no primo de R depende transitivamente de alguna clave candidata de R.

Forma Normal de Boyce-Codd
Un esquema de relación R está en FNBC si para toda DF no trivial X→Y de R se cumple que X es una superclave de R.

Descomposición sin Pérdida de Información

Sea un esquema de relación R con un conjunto de dependencias funcionales F y una descomposición D = {R1, R2, …, Rm} de R, decimos que D es una descomposición sin Pérdida de Información si, para cada instancia r de R que satisface las dependencias funcionales de F, se verifica que:

$$r = \pi_{R1}(r) \;|x|\; \pi_{R2}(r) \;|x|\; \ldots \;|x|\; \pi_{Rm}(r)$$

Es decir, si luego de la descomposición es posible reconstruir la relación inicial r mediante la junta natural de cada subesquema Ri, entonces decimos que no hubo pérdida de información.
Si al hacer la junta, se pierden algunas tuplas o bien se obtienen tuplas adicionales con información errónea (tuplas espurias), entonces decimos que sí hubo pérdida de información.

También puede ocurrir que al hacer la junta natural de los subesquemas, la relación resultante tenga menos atributos que el r original, en ese caso también decimos que hubo pérdida de información.

Veamos un ejemplo:

Supongamos el esquema de relación R(Patente, Modelo, Color, Marca, Motor, Combustible) y la siguiente instancia "r" de dicho esquema:

r

Patente	Modelo	Color	Marca	Motor	Combustible
IKU-496	206 Generation	Rojo	Peugeot	1.4	Nafta
HRV-709	207 Compact	Blanco	Peugeot	1.6	Nafta
GVR-286	Vectra GT	Negro	Chevrolet	2.4	Nafta
ILP-456	207 Compact	Azul	Peugeot	1.8	Diesel
HBN-142	206 Generation	Gris	Peugeot	1.4	Nafta

Evaluaremos si la siguiente descomposición de R es con o sin pérdida de información:
R1(Patente, Modelo, Color) R2(Modelo, Marca) R3(Modelo, Motor, Combustible)

Según la definición, no habrá P.I. si se cumple: $r = \pi_{R1}(r) \;|x|\; \pi_{R2}(r) \;|x|\; \pi_{R3}(r)$

$\pi_{R1}(r)$

Patente	Modelo	Color
IKU-496	206 Generation	Rojo
HRV-709	207 Compact	Blanco
GVR-286	Vectra GT	Negro
ILP-456	207 Compact	Azul
HBN-142	206 Generation	Gris

$\pi_{R2}(r)$

Modelo	Marca
206 Generation	Peugeot
207 Compact	Peugeot
Vectra GT	Chevrolet

$\pi_{R3}(r)$

Modelo	Motor	Combustible
206 Generation	1.4	Nafta
207 Compact	1.6	Nafta
Vectra GT	2.4	Nafta
207 Compact	1.8	Diesel

La junta natural de las tres tablas anteriores da como resultado:

$\pi_{R1}(r) \;|x|\; \pi_{R2}(r) \;|x|\; \pi_{R3}(r)$

Patente	Modelo	Color	Marca	Motor	Combustible	
IKU-496	206 Generation	Rojo	Peugeot	1.4	Nafta	
HRV-709	207 Compact	Blanco	Peugeot	1.6	Nafta	
HRV-709	**207 Compact**	**Blanco**	**Peugeot**	**1.8**	**Diesel**	Tupla espuria
GVR-286	Vectra GT	Negro	Chevrolet	2.4	Nafta	
ILP-456	**207 Compact**	**Azul**	**Peugeot**	**1.4**	**Nafta**	Tupla espuria
ILP-456	207 Compact	Azul	Peugeot	1.8	Diesel	
HBN-142	206 Generation	Gris	Peugeot	1.4	Nafta	

Finalmente podemos ver que no se cumple $r = \pi_{R1}(r) \;|x|\; \pi_{R2}(r) \;|x|\; \pi_{R3}(r)$
Por lo tanto la descomposición es con PERDIDA DE INFORMACIÓN.

MÉTODO DEL TABLEAU

Este método se utiliza para verificar si una descomposición es con o sin pérdida de información.

Dado el esquema R(A1, A2, …, An) y la descomposición D={R1, R2, …, Rm} de R, se debe armar un tableau T de la siguiente manera:

1) T tendrá n columnas, una para cada atributo de R.
2) T tendrá m filas, una para cada esquema Ri de la descomposición.
3) Dadas la fila i y la columna j (para el esquema Ri y atributo Aj), el contenido del tableau será:

   a_j si $A_j \in R_i$
   o bien
   b_{ij} si $A_j \notin R_i$

Las variables del tipo “aj“ se denominan variables distinguidas y son únicas por columna, pudiendo repetirse en diferentes filas.
Las variables del tipo “bij” se denominan variables no distinguidas e inicialmente son todas diferentes.

<u>Procedimiento:</u>

1) Construir el tableau T para la descomposición propuesta. Lo llamaremos T_0 o tableau inicial.

2) Mientras haya cambios sobre T, hacer:
   Por cada df X→A de F hacer:
      Por cada par de filas t_i, t_h de T hacer:
         Si ($t_i[X] = t_h[X]$) and ($t_i[A]$ <> $t_h[A]$) entonces
            Igualar los valores de $t_i[A]$ y $t_h[A]$ teniendo en cuenta que:
            - Si uno de esos valores es una variable distinguida (a) y el otro es una variable no distinguida (b) se debe reemplazar la no distinguida por la distinguida.
            - Si ambos valores son variables no distinguidas, se debe escoger una de las variables y reemplazarla por la otra.

   Fin Mientras

3) Finalmente, si luego de hacer los cambios, obtenemos al menos una fila que conste exclusivamente de variables distinguidas (a), entonces la descomposición es sin pérdida de información, en caso contrario, la descomposición es con pérdida de información.

Veamos un ejemplo:

Dado el esquema de relación **R(ABCDE)** con **F={ E → D, C → E, B → E, AD → B }**, utilizaremos el método del Tableau para verificar si la siguiente descomposición es Sin Pérdida de Información:
R1(ABC) R2(CD) R3(ADE)

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

Tableau inicial

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

C → E

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

E → D

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

AD → B

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

B → E

RESULTADO: No hay Pérdida de Información, ya que se logra obtener una fila con todas variables distinguidas.
NOTA: En este ejemplo fue necesario iterar 3 veces entre las DF para lograr conseguir una fila con todas variables distinguidas.

TEOREMA DE "HEATH"

(Página 434 del libro de Elmasri-Navathe)

Una descomposición D={R1, R2} de R es Sin Perdida de Información, si y solo si:

- La DF (R1∩R2) → (R1-R2) pertenece a F^+

O bien,

- La DF (R1∩R2) → (R2-R1) pertenece a F^+

<u>Nota</u>: Este teorema solo es útil para descomposiciones de dos subesquemas, a diferencia del método del Tableau que permite verificar la Pérdida de Información de cualquier descomposición.

Descomposición sin Pérdida de Dependencias Funcionales

Sea un esquema de relación R con un conjunto de dependencias funcionales F y una descomposición D = { R1, R2, …, Rm} de R. Sea Fi el conjunto de todas las dependencias funcionales no triviales de F^+ que mencionan solamente atributos del subesquema Ri de la descomposición.

$$Fi = \{ X \rightarrow Y \ / \ X \rightarrow Y \in F^+ \text{ y } X \rightarrow Y \text{ no es trivial y } XY \subseteq Ri \}$$

Decimos que D es una descomposición <u>sin Pérdida de Dependencias Funcionales</u> si se verifica que:

$$F \equiv F_1 \cup F_2 \cup \ldots \cup F_m$$

Veamos un ejemplo:

Supongamos el esquema R (ABCDEF) y el conjunto F= {A→C, BC→A, D→F, AE→F, DC→E}.
Luego, supongamos que R se descompone en los siguientes 4 subesquemas:

- R_1 (AC)
- R_2 (DF)
- R_3 (ADE)
- R_4 (ABD)

Cada subesquema tendrá su propio conjunto de dependencias funcionales:

F_1 = {A→C}
F_2 = {D→F}
F_3 = {AD→E}

Ya que:
A^+_F = {AC}
D^+_F = {DF}
E^+_F = {E}
AD^+_F = {ADCF**E**}
AE^+_F = {AECF}
DE^+_F = {DEF}

F_4 = { }

Ya que:
A^+_F = {AC}
B^+_F = {B}
E^+_F = {E}
AB^+_F = {ABC}
AD^+_F = {ADCFE}
BD^+_F = {BDF}

Ahora veremos si hubo pérdida de dependencias funcionales. Para ello, hay que verificar si el conjunto F original es equivalente a la unión de los cuatro conjuntos F_i de los subesquemas.

$$F \overset{?}{\equiv} F_1 \cup F_2 \cup F_3 \cup F_4$$

$$\{A \rightarrow C, BC \rightarrow A, D \rightarrow F, AE \rightarrow F, DC \rightarrow E\} \overset{?}{\equiv} \{A \rightarrow C\} \cup \{D \rightarrow F\} \cup \{AD \rightarrow E\} \cup \{\}$$

$$\{A \rightarrow C, BC \rightarrow A, D \rightarrow F, AE \rightarrow F, DC \rightarrow E\} \overset{?}{\equiv} \{A \rightarrow C, D \rightarrow F, AD \rightarrow E\}$$

Llamaremos G al conjunto resultante de la unión de los 4 conjuntos F_1, F_2, F_3 y F_4. Para verificar si F y G son equivalentes se debe verificar si F cubre a G y viceversa.
En este caso, a simple vista vemos que G no cubre a F, ya que por ejemplo la dependencia BC→A de F no existe en G+.

BC^+ en G = {BC}

Por lo tanto, podemos decir que en este caso la descomposición tiene Pérdida de Dependencias Funcionales.

ALGORITMO PARA 3FN
(Sin PI y sin PDF)

Pasos:
1) Calcular Fmin

2) Para cada miembro izquierdo X que aparezca en Fmin, crear un esquema de relación {X unión A1 unión A2 unión An} donde X→A1, X→A2, … , X→An sean todas DF de Fmin.

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

Veamos un ejemplo:
Dado R (ABCDEFG) y F= {A→BC, AD→G, AC→E, C→A, B→C, F→B}

Primero calculamos el Fmin:
Fmin ={A→B, AD→G, A→E, C→A, B→C, F→B}

Luego, analizamos en qué forma normal se encuentra R, para ello calculamos sus Claves Candidatas:
CC={DF}
Analizando las dependencias funcionales de Fmin, vemos que R se encuentra en 1FN (ya que F→B viola 2FN).

Finalmente, descomponemos R en subesquemas que cumplan 3FN:

R1(ABE)
F1={A→B, A→E}

R2(ADG)
F2={AD→G}

R3(AC)
F3={C→A}

R4(BC)
F4={B→C}

R5(BF)
F5={F→B}

R6(DF)
F6={}

Nota: El último esquema R6 se agrega porque ninguno de los subesquemas anteriores contiene la clave candidata DF.

ALGORITMO PARA FNBC
(Sin PI)

Dado el esquema R y el conj. de DF F (no es necesario trabajar con Fmin, pero si es aconsejable):
Tomar cualquier DF X→Y perteneciente a F <u>que no sea trivial</u> y que viole FNBC, y hacer 2 subesquemas:

R1=**(X U Y)** y R2=**(R - Y)**

Si alguno de los dos subesquemas obtenidos sigue violando FNBC, aplicar nuevamente el punto anterior sobre dicho esquema. Repetir ese paso, e ir dividiendo en subesquemas, tantas veces como sea necesario, hasta lograr obtener todos subesquemas que cumplan FNBC.

División de las Dependencias Funcionales:
Las dependencias de F deberán repartirse a F1 o F2 según corresponda (puede ser que alguna dependencia no pueda incluirse en ninguno de los dos subconjuntos y en ese caso la misma se perderá, pero esa pérdida puede salvarse intentando inferir alguna otra dependencia equivalente en base a ella, que si pueda ubicarse en F1 o F2).

Veamos un ejemplo:

Dado R(MNOPQS) y F={M→ON, N→MO, O→NM, MQ→SO, OP→Q}

En primer lugar, calcularemos Fmin para simplificar el desarrollo:

F1={ M→O, M→N, N→M, N→O, O→N, O→M, MQ→S, MQ→O, OP→Q }
F2={ M→O, M→N, N→M, N→O, O→N, O→M, MQ→S, OP→Q}
F3={ M→N, N→O, O→M, MQ→S, OP→Q } = Fmin

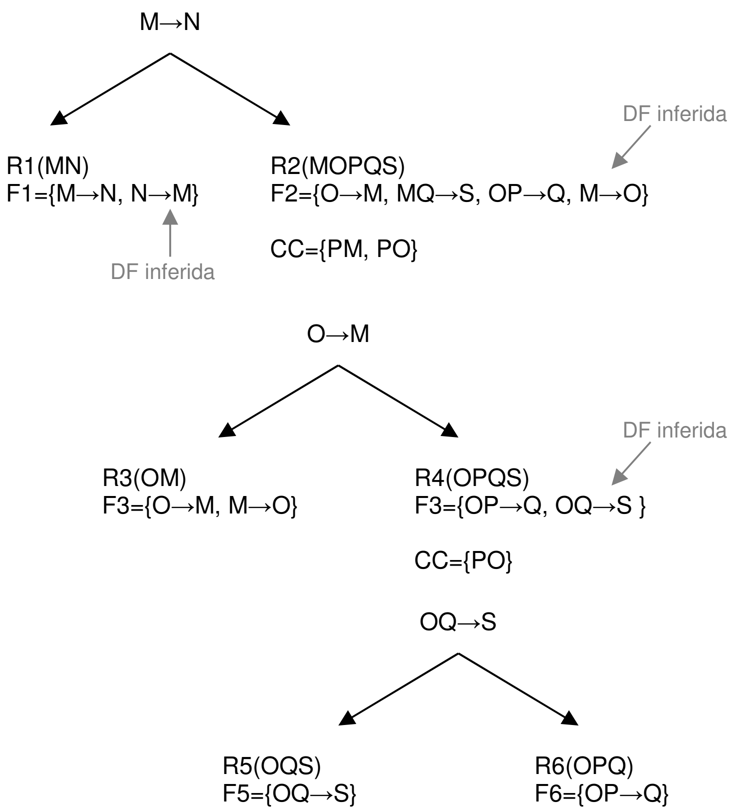
¿En qué forma normal se encuentra R?
Para responder a esa pregunta debemos calcular las Claves Candidatas y luego evaluar una a una las dependencias funcionales.

CC = {PM, PN, PO}

M→N	cumple 3FN
N→O	cumple 3FN
O→M	cumple 3FN
MQ→S	cumple 2FN
OP→Q	cumple FNBC

Repuesta: R se encuentra en 2FN

Ahora aplicaremos el Algoritmo para obtener una descomposición de R que cumpla FNBC.



La descomposición obtenida es:
R1, R3, R5 y R6 con F1, F3, F5 y F6

En este caso, al inferir las DF indicadas, se logra que no haya pérdida de dependencias funcionales.