

1. MODELO RELACIONAL

R : Esquema de relación

A₁, A₂, ..., A_n o A, B, C,..., etc. atributos de R

r: Instancia de R

t₁, t₂, ..., etc. tuplas de r

2. DEPENDENCIAS FUNCIONALES (DF)

2.1. DEFINICION DE DF

Decimos que vale $X \twoheadrightarrow Y$ en R si para toda r se verifica que si $t_1(X)=t_2(X)$ entonces necesariamente $t_1(Y)=t_2(Y)$

Decimos que “X determina funcionalmente Y”, o que “Y es determinado funcionalmente por X”

X, conjunto de atributos de R, lado izquierdo.

Y, conjunto de atributos de R, lado derecho

X e Y no tienen que ser necesariamente disjuntos

t₁ y t₂ dos tuplas cualesquiera de r

Si r cumple con todas las dependencias funcionales entonces decimos que r es LEGAL

Las dependencias funcionales las establece el diseñador de la BD

EJEMPLO 1:

Supongamos que queremos registrar para una facultad los datos personales de los alumnos, las materias en las que se inscribieron y los exámenes que rindieron.

Para esto definimos el siguiente esquema de relación:

FACULTAD (LU, NOMBRE, MATERIA, IFEC, EFEC, NOTA)

y el siguiente conjunto de dependencias funcionales F:

LU \twoheadrightarrow NOMBRE (no puede haber dos alumnos con el mismo LU)

LU, MATERIA \twoheadrightarrow IFEC (se puede inscribir una sola vez en cada materia)

LU, MATERIA, EFEC \twoheadrightarrow NOTA (hay una sola nota por examen)

NOTA: Al no estar LU, MATERIA \twoheadrightarrow EFEC se puede rendir varias veces la misma materia

2.2. INFERENCIAS DE F

Decimos que si F INFIERE f ($F \models f$) toda r que satisface F debe necesariamente satisfacer también f

EJEMPLO 2:

Del conjunto de dependencias funcionales del Ejemplo 1, podemos inferir:

$F \models$ LU, MATERIA \twoheadrightarrow NOMBRE, IFEC

2.3. REGLAS DE INFERENCIA (AXIOMAS DE ARMSTRONG)

- 1) Reflexividad: Si $Y \subseteq X$ entonces $X \twoheadrightarrow Y$
- 2) Aumento: Para cualquier W , si $X \twoheadrightarrow Y$ entonces $XW \twoheadrightarrow WY$
- 3) Transitividad: Si $X \twoheadrightarrow Y$ e $Y \twoheadrightarrow Z$ entonces $X \twoheadrightarrow Z$

De 1) inferimos las triviales

Se sigue que siempre se cumple $X \twoheadrightarrow X$

Por inercia se tiende a pensar que Si $X \twoheadrightarrow Y$ entonces $Y \twoheadrightarrow X$, pero esto, aunque a veces puede ser verdadero, en general es falso.

2.4. CLAUSURA DE F (F^+)

Conjunto de todas las dependencias funcionales que pueden inferirse de F aplicando los axiomas

$$F^+ = \{X \twoheadrightarrow Y \mid F \models X \twoheadrightarrow Y\}$$

EJEMPLO 3:

$R(A,B)$
 $F: A \twoheadrightarrow B$

$$F^+ = \{A \twoheadrightarrow B, A \twoheadrightarrow A, B \twoheadrightarrow B, AB \twoheadrightarrow A, AB \twoheadrightarrow B, AB \twoheadrightarrow AB, A \twoheadrightarrow AB\}$$

EJEMPLO 3.1:

$R(A,B,C)$
 $F = \{AB \twoheadrightarrow C, C \twoheadrightarrow B\}$

$$F^+ = \{A \twoheadrightarrow A, AB \twoheadrightarrow A, AC \twoheadrightarrow A, ABC \twoheadrightarrow A, B \twoheadrightarrow B, AB \twoheadrightarrow B, BC \twoheadrightarrow B, ABC \twoheadrightarrow B, C \twoheadrightarrow C, AC \twoheadrightarrow C, BC \twoheadrightarrow C, ABC \twoheadrightarrow C, AB \twoheadrightarrow AB, ABC \twoheadrightarrow AB, AC \twoheadrightarrow AC, ABC \twoheadrightarrow AC, BC \twoheadrightarrow BC, ABC \twoheadrightarrow BC, ABC \twoheadrightarrow ABC, AB \twoheadrightarrow C, AB \twoheadrightarrow AC, AB \twoheadrightarrow BC, AB \twoheadrightarrow ABC, C \twoheadrightarrow B, C \twoheadrightarrow BC, AC \twoheadrightarrow B, AC \twoheadrightarrow AB\}$$

2.5. REGLAS ADICIONALES

- 4) Unión: $X \twoheadrightarrow Y$ y $X \twoheadrightarrow Z$ entonces $X \twoheadrightarrow YZ$
- 5) Pseudotransitividad: Para cualquier W , $X \twoheadrightarrow Y$ e $YW \twoheadrightarrow Z$ entonces $XW \twoheadrightarrow Z$
- 6) Descomposición: $X \twoheadrightarrow YZ$ entonces $X \twoheadrightarrow Y$ y $X \twoheadrightarrow Z$

Las reglas adicionales se demuestran aplicando los axiomas

EJEMPLO 4:

Demostraremos la regla de unión:

1. $X \twoheadrightarrow Y$ (dada)
2. $X \twoheadrightarrow Z$ (dada)
3. $X \twoheadrightarrow XY$ (aumento de 1 con X)
4. $XY \twoheadrightarrow YZ$ (aumento de 2 con Y)
5. $X \twoheadrightarrow YZ$ (transitividad de 3 y 4)

Si es falsa se demuestra con una instancia que sea contraejemplo:

EJEMPLO 5:

$\{X \twoheadrightarrow Z, Y \twoheadrightarrow Z\} \models X \twoheadrightarrow Y \text{ ?}$

X	Y	Z

1	2	5
1	3	5
2	2	5
2	3	5

2.6. CLAUSURA DE UN CONJUNTO DE ATRIBUTOS X (X+)

X+ con respecto a F, es el conjunto de todos los atributos A tal que $X \twoheadrightarrow A$, o sea:

$X+ = \{A \in R \mid F \models X \twoheadrightarrow A\}$

Una forma de calcular X+ es computar una secuencia de conjuntos de atributos X0, X1, ... aplicando las siguientes reglas:

- 1) X0 es X
 - 2) Xi+1 es Xi Unión el conjunto de atributos A tal que hay alguna dependencia funcional $Y \twoheadrightarrow Z$ en F, A está en Z e $Y \subseteq Xi$
- Aplicamos repetidas veces la regla (2) hasta que $Xi = Xi+1$

NOTA: Como $X = X0 \subseteq \dots \subseteq Xi \subseteq R$, y R es finito, eventualmente llegaremos a que $Xi = Xi+1$, que es la parada del algoritmo.

2.7. PROPIEDAD DE LA CLASURA

Dados F y $X \twoheadrightarrow Y$, luego $F \models X \twoheadrightarrow Y$ Sii $Y \subseteq X+$

Si $X+ = R$ entonces X es superclave de R

EJEMPLO 6:

Tomando el mismo R y F del Ejemplo 3, tenemos:

R(A,B)
F: $A \twoheadrightarrow B$
 $F+ = \{A \twoheadrightarrow B, A \twoheadrightarrow A, B \twoheadrightarrow B, AB \twoheadrightarrow A, AB \twoheadrightarrow B, AB \twoheadrightarrow AB, A \twoheadrightarrow AB\}$

$B+ = B$
 $A+ = AB = R$, por lo tanto A es clave

EJEMPLO 7:

R(A,B,C,D,E)
F= $\{AB \twoheadrightarrow C, C \twoheadrightarrow D, BD \twoheadrightarrow E\}$
 $AB+ \text{ ?}$

$X0 = AB$ por $AB \twoheadrightarrow C$
 $X1 = ABC$ por $C \twoheadrightarrow D$
 $X2 = ABCD$ por $BD \twoheadrightarrow E$
 $X3 = ABCDE$

$X+ = ABCDE = R$, por lo tanto AB es superclave

2.8. SUPERCLAVE Y CLAVE

Si $X \twoheadrightarrow R$ entonces decimos que X es SUPERCLAVE

Si además no existe ningún $Z \subset X$ tal que $Z \twoheadrightarrow R$ entonces X también es CLAVE

En el Ejemplo 1, la clave es $\{LU, MATERIA, EFEC\}$ ¿Por qué?

R puede tener una o varias claves a las que llamaremos en general *claves candidatas* (CC)

Un ejercicio típico es hallar todas las claves de R .

Una forma de hacerlo es computando primero los atributos que no están en ningún lado derecho, llamémoslo X . Si $X^+ = R$, X es la única CC. Sino, hay que probar todos los casos.

Es decir, comenzamos con $X \cup A$, para cada A . Si $XA^+ = R$, XA es CC, y todos los que incluyan a XA serán superclave. Si $XA^+ \neq R$, agregamos un atributo más a XA , sea este B , y computamos XAB^+ , y así sucesivamente.

2.9. EQUIVALENCIA DE CONJUNTOS DE DEPENDENCIAS FUNCIONALES

Decimos que dos conjuntos de dependencias funcionales F y G sobre R son equivalentes ($F \equiv G$) si

$$F^+ = G^+$$

También si $F \models G$ y $G \models F$ (si F cubre a G y G cubre a F)

EJEMPLO 8:

$R(A,B,C)$

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

$$G = \{B \twoheadrightarrow A, C \twoheadrightarrow B, A \twoheadrightarrow C\}$$

$$F \equiv G$$

2.10. CUBRIMIENTO MINIMAL

Dado F buscamos un F_m tal que $F_m \equiv F$ y además F_m tiene:

- 1) Todo lado derecho tiene un único atributo (regla de descomposición)
- 2) Todo lado izquierdo es reducido (no tiene atributos redundantes)
($B \subset X$ es redundante para $X \twoheadrightarrow A$ si $A \in (X - \{B\})^+$)
- 3) No contiene dependencias funcionales redundantes (en general las que se obtienen por transitividad, $X \twoheadrightarrow A$ es redundante si $(F - \{X \twoheadrightarrow A\}) \equiv F$)

Puede haber varios cubrimientos minimales para un mismo F

EJEMPLO 9:

$R(A,B,C,D)$

$$F = \{A \twoheadrightarrow BD, B \twoheadrightarrow C, C \twoheadrightarrow D, BC \twoheadrightarrow D\}$$

- 1) $G = \{A \twoheadrightarrow B, A \twoheadrightarrow D, B \twoheadrightarrow C, C \twoheadrightarrow D, BC \twoheadrightarrow D\}$
- 2) $G = \{A \twoheadrightarrow B, A \twoheadrightarrow D, B \twoheadrightarrow C, C \twoheadrightarrow D, C \twoheadrightarrow D\}$, B es redundante en $BC \twoheadrightarrow D$
- 3) $G = \{A \twoheadrightarrow B, B \twoheadrightarrow C, C \twoheadrightarrow D\}$, $A \twoheadrightarrow D$ es redundante y $C \twoheadrightarrow D$ esta duplicada

$$F_{\min} := G$$

3. PERDIDA DE INFORMACION

Analizaremos ahora los problemas (*anomalías*) que se pueden presentar en un esquema y las posibles soluciones.

EJEMPLO 10:

Volvamos al esquema del Ejemplo 1:

FACULTAD (LU, NOMBRE, MATERIA, IFEC, EFEC, NOTA)

Analicemos los problemas que podría tener el esquema:

1) Redundancia de información: El nombre del alumno se repite por cada materia en la que se inscribe y por cada examen que rinda. Lo mismo pasa con el nombre de la materia y la fecha de inscripción si la rinde varias veces.

2) Anomalías de actualización: Si una alumna se casa y decide cambiar por su apellido de casada tenemos que actualizar varias tuplas y podríamos cometer errores u omisiones.

3) Anomalías de inserción: No podemos dar de alta a un alumno hasta que se haya inscripto en la primer materia o aún peor hasta que haya rendido el primer examen. Podríamos hacerlo pero tendríamos que poner valores nulos en campos que forman la clave como Materia, IFec y EFec.

FACULTAD (lu1, nom1, -, -, -, -)

Si se toma un examen y no se presenta nadie también tendríamos que poner valores nulos en algunos campos de la clave.

FACULTAD (-, -, -, mat1, -, efec1, -)

4) Anomalías de bajas: Si en algún momento decidimos borrar los datos de los exámenes rendidos por un alumno perderemos los datos personales del mismo.

Por lo mencionado más arriba decidimos descomponer FACULTAD en:

ALUMNO (LU, NOMBRE)

RESULTADO (LU, MATERIA, NOTA)

EXAMEN (MATERIA, EFEC)

INSCRIPTO (LU, MATERIA, IFEC)

Supongamos que hacemos la consulta:

¿En qué fechas dio examen el alumno con LU= “123/01”?

Para responder esto podríamos hacer la siguiente consulta:

$\sigma_{LU="123/01"}(RESULTADO \bowtie MATERIA \bowtie EXAMEN)$

El resultado no es correcto porque la junta asocia al alumno con todas fechas de examen de las materias que haya rendido.

El resultado tiene muchas más tuplas (espurias) que las de la respuesta correcta.

Decimos que esta descomposición es con PERDIDA DE INFORMACION.

La descomposición correcta es:

ALUMNO (LU, NOMBRE)

EXAMEN (LU, MATERIA, EFEC, NOTA)

INSCRIPTO (LU, MATERIA, IFEC)

¿Por qué?

3.1. FORMALIZANDO

Sea $R = (A_1, A_2, \dots, A_n)$ y F conjunto de dependencias funcionales

Una descomposición ρ de R

$$\rho = \{R_1, R_2, \dots, R_k\}$$

Tal que

$$\bigcup_{i=1, k} R_i = R$$

Decimos que ρ es una descomposición sin pérdida (lossless join, SPI) si para cada instancia r de R que satisface F , se verifica:

$$r = \bigcup_{i=1, k} \Pi_{R_i}(r)$$

Entonces es importante determinar, dados R , F y ρ , si ρ es SIN PERDIDA (lossless, SPI)

EJEMPLO 11:

Sean $R = (A, B, C)$

$F = \{A \twoheadrightarrow B\}$

$\rho = (AB, BC)$

No es lossless join con respecto a F , por ejemplo:

$$r = \{a_1 \ b_1 \ c_1, \ a_2 \ b_1 \ c_2\}$$

$$\Pi_{AB}(r) = \{a_1 \ b_1, \ a_2 \ b_1\}$$

$$\Pi_{BC}(r) = \{b_1 \ c_1, \ b_1 \ c_2\}$$

$$\Pi_{AB}(r) \cup \Pi_{BC}(r) = \{a_1 \ b_1 \ c_1, \ a_1 \ b_1 \ c_2, \ a_2 \ b_1 \ c_1, \ a_2 \ b_1 \ c_2\}$$

Que es un superconjunto de r .

Si en cambio hacemos $\rho = (AB, AC)$ veremos que es LOSSLESS JOIN.

3.2. PROPIEDAD DE DESCOMPOSICION BINARIA

Decimos que una descomposición ρ de R , $\rho = (R_1, R_2)$ es Lossless Join con respecto a un conjunto de dependencias funcionales F , Sii

- 1) La dependencia funcional $(R_1 \cap R_2) \twoheadrightarrow (R_1 - R_2)$ está en F^+
o
- 2) La dependencia funcional $(R_1 \cap R_2) \twoheadrightarrow (R_2 - R_1)$ está en F^+

O sea que si la intersección de los atributos de R_1 y R_2 forman una superclave para uno de los dos esquemas, entonces ρ es SIN PERDIDA

Ver Ejemplo 11.

3.3. ALGORITMO DEL TABLEAU

Hay un algoritmo más general para determinar si una descomposición es lossless. Es el algoritmo del Tableau:

Dados R , F y $\rho = (R_1, R_2, \dots, R_k)$, inicialmente se construye un tableau T inicial T_0 donde las columnas son los atributos y las filas los subesquemas de ρ .

Luego completamos el tableau con símbolos distinguidos (aj) si $A_j \in R_i$ o con no distinguidos (bij) si $A_j \notin R_i$.
 Luego vamos modificando el tableau aplicando las dependencias funcionales de la siguiente forma: Para cada $X \twoheadrightarrow A$ si (fila_i[X]=fila_h[X]) y (fila_i[A]≠fila_h[A]), entonces:
 i) si fila_i[A]=aj hacemos fila_h[A]=aj o
 ii) si fila_i[A]=bij y fila_h[A]=bhj hacemos fila_h[A]=bij
 Así vamos generando una serie T0, T1, T2, ..., T*.
 Paramos cuando no se puedan hacer más cambios en el tableau final (T*) por aplicación de las dependencias funcionales.
 Si T* contiene una fila con todos símbolos distinguidos entonces ρ es SPI, de lo contrario ρ es con pérdida.

T0:

	A1	A2	.	.	An

R1					
R1					
.					
.					
Rk					

Veamos algunos ejemplos:

EJEMPLO 11.1:

R= ABCD
 F= { A-->B, AC-->D}
 ρ = (AB, ACD)

	A	B	C	D	

AB	a1	a2	b13	b14	
ACD	a1	b22	a3	a4	A-->B (b22=a2)
	A	B	C	D	

AB	a1	a2	b13	b14	
ACD	a1	a2	a3	a4	< ==

Vemos que la segunda fila tiene todos símbolos distinguidos y por lo tanto ρ es SPI.

EJEMPLO 11.2:

R= ABCDE
 F= { A-->C, B-->C, C-->D, DE-->C, CE-->A}
 ρ = (AD, AB, BE, CDE, AE)

	A	B	C	D	E	

AD	a1	b12	b13	a4	b15	
AB	a1	a2	b23	b24	b25	
BE	b31	a2	b33	b34	a5	
CDE	b41	b42	a3	a4	a5	
AE	a1	b52	b53	b54	a5	A-->C (b23=b53=b13) B-->C (b33=b13)

	A	B	C	D	E	
AD	a1	b12	b13	a4	b15	
AB	a1	a2	b13	b24	b25	
BE	b31	a2	b13	b34	a5	
CDE	b41	b42	a3	a4	a5	
AE	a1	b52	b13	b54	a5	
						C-- >D (b24=b34=b54=a4)
						DE-- >C (b13=a3)
						CE-- >A (b31=b41=a1)
	A	B	C	D	E	
AD	a1	b12	a3	a4	b15	
AB	a1	a2	a3	a4	b25	
BE	a1	a2	a3	a4	a5	
CDE	a1	b42	a3	a4	a5	
AE	a1	b52	a3	a4	a5	
						< ==

Vemos que la tercera fila tiene todos símbolos distinguidos y por lo tanto ρ es SPI.

4. PERDIDA DE DEPENDENCIAS FUNCIONALES

Además de preservar la información (SPI) una descomposición ρ debería preservar las dependencias funcionales (SPDF)

4.1. PROYECCION DE UN CONJUNTO DE DEPENDENCIAS FUNCIONALES

Dados R , y $\rho = (R_1, R_2, \dots, R_k)$ decimos que la proyección de F sobre un conjunto de atributos Z ($\Pi_Z(F)$) es el conjunto de dependencias funcionales $X \twoheadrightarrow Y \in F^+$ tal que $XY \subseteq Z$

ρ preserva F si la unión de todas las dependencias funcionales en $\Pi_{R_i}(F)$ para $i = 1, 2, \dots, k$ implican F , o sea:

$$F^+ = (\cup_{i=1, k} \Pi_{R_i}(F))^+$$

NOTA: ρ podría ser SPI pero no SPDF o al revés

EJEMPLO 12:

a)

$R(A, B, C)$

$F = \{AB \twoheadrightarrow C, C \twoheadrightarrow A\}$

$\rho = (BC, AC)$

ρ es SPI pero sin embargo no es SPDF (no se preserva $AB \twoheadrightarrow C$)

b)

$R(A, B, C, D)$

$F = \{A \twoheadrightarrow B, C \twoheadrightarrow D\}$

$\rho = (AB, CD)$

ρ no es SPI pero sin embargo es SPDF

EJEMPLO 13: (para hacer énfasis en que hay proyectar F^+)

Dados:

$R(A, B, C, D)$

$F = \{A \twoheadrightarrow B, B \twoheadrightarrow C, C \twoheadrightarrow D, D \twoheadrightarrow A\}$

$\rho = (AB, BC, CD)$

ρ es SPDF?

Vemos que en F^+ cada atributo determina a todos los otros

Intuitivamente podemos suponer que cuando proyectamos F se pierde $D \twoheadrightarrow A$ pero esto no es así porque cuando decimos que proyectamos F estamos proyectando F^+

Si hacemos los cálculos veremos que finalmente:

$\{B \twoheadrightarrow A, C \twoheadrightarrow B, D \twoheadrightarrow C\} \models D \twoheadrightarrow A$

4.2. ALGORITMO PARA TESTEAR PERDIDA DE DEPENDENCIAS

Hay un algoritmo para probar si ρ preserva las dependencias funcionales sin necesidad de calcular F^+

La idea es computar $Z \cup ((Z \cap R_i)^+ \cap R_i)$ donde Z inicialmente es igual al lado izquierdo de la dependencia funcional $X \twoheadrightarrow Y$ que se desea testear.

(La clausura de $(Z \cap R_i)$ es con respecto a F)

Dados R , F y ρ , queremos verificar si se preserva $X \twoheadrightarrow Y$

Procedimiento:

```
Z := X;  
while Z cambie do  
for i := 1 to k do  
  Z := Z  $\cup ((Z \cap R_i)^+ \cap R_i)$ ;
```

Si Z es el resultado final y, además, $Y \subseteq Z$ luego $X \twoheadrightarrow Y \in (\bigcup_{i=1,\dots,k} \Pi R_i(F))^+$
Si esto se cumple para toda dependencia funcional $X \twoheadrightarrow Y$ entonces podemos afirmar que ρ es SPDF.

EJEMPLO 13.1:

Sea $R = (A, B, C, D, E)$
 $F = \{AB \twoheadrightarrow C, B \twoheadrightarrow D, C \twoheadrightarrow E, D \twoheadrightarrow C, E \twoheadrightarrow B\}$
 $\rho = (ABC, CD, BDE)$

Queremos verificar si se preserva $C \twoheadrightarrow E$

Aplicamos el algoritmo:

Comenzamos haciendo $Z = C$

Luego iteramos ($R_1=ABC$, $R_2=CD$, $R_3=BDE$):

$$C \cup ((C \cap ABC)^+ \cap ABC) = BC$$

$$BC \cup ((BC \cap CD)^+ \cap CD) = BCD$$

$$BCD \cup ((BCD \cap BDE)^+ \cap BDE) = BCDE$$

Vemos que si seguimos iterando no se producen más cambios al valor obtenido $Z = BCDE$. También verificamos que se cumple $Y \subseteq Z$, ya que $E \subseteq BCDE$.
Luego $C \twoheadrightarrow E$ se preserva.

5. FORMAS NORMALES

Se obtienen por DESCOMPOSICION de R (esquema universal) o por SINTESIS a partir de las dependencias funcionales

La idea es evitar la redundancia de información que es lo que produce las anomalías de altas, bajas y modificación.

Por otro lado las formas normales nos dan un piso para el proceso de descomposición (cuando todos los subesquemas quedan en la FN deseada paramos)

5.1. CLAVES

Como ya mencionamos un esquema R puede tener una o varias claves candidatas (CC)

La *clave primaria* (PK) se elige arbitrariamente entre las CC. Al resto las llamaremos *claves secundarias*

5.2. ATRIBUTOS PRIMOS

Decimos que un atributo es PRIMO si es miembro de ALGUNA clave candidata

Si no, decimos que es NO PRIMO

5.3. DEPENDENCIA FUNCIONAL PARCIAL (ATRIBUTOS PARCIALMENTE DEPENDIENTES)

Decimos que $X \twoheadrightarrow Y$ es dependencia funcional PARCIAL (o que Y depende parcialmente de X) si para algún $B \in X$ $(X - \{B\}) \twoheadrightarrow Y$

Si no, decimos que la dependencia funcional es TOTAL (o que Y depende totalmente de X)

5.4. SEGUNDA FORMA NORMAL (2FN)

R esta en 2FN si TODO atributo no primo A en R NO es parcialmente dependiente de ALGUNA clave de R.

O en forma equivalente:

R está en 2FN si TODO atributo NO primo A en R es TOTALMENTE dependiente de TODAS las claves de R.

5.5. TERCERA FORMA NORMAL (3FN)

R esta en 3FN si para TODA dependencia funcional no trivial $X \twoheadrightarrow A$ sobre R,

- a) X es superclave de R o
- b) A es primo

O en forma equivalente:

R está en 3FN si para TODA dependencia funcional no trivial $X \twoheadrightarrow Y$ sobre R, o bien X es una superclave de R o Y es un subconjunto de alguna clave de R.

5.6. FORMA NORMAL DE BOYCE-CODD (FNBC)

R está en FNBC si para TODA dependencia funcional no trivial $X \twoheadrightarrow A$ sobre R, X es una superclave de R

O en forma equivalente:

R está en FNBC si para TODA dependencia funcional $X \twoheadrightarrow Y$ en F^+ , o bien $Y \subseteq X$ o X es una superclave de R.

Algunas propiedades para tener en cuenta:

- La validez debe ser para F^+ (PROPIEDAD para FNBC y 3FN: si todas las dependencias funcionales tienen lado derecho simple, entonces no hay violación en F^+)
- Si R está en FNBC entonces también está en 3FN y en 2FN y si R está en 3FN entonces también está en 2FN
- Todo esquema se puede descomponer en 3FN que sea SPI y SPDF
- Hay esquemas que no se pueden descomponer en FNBC y que sean SPDF
- Todo esquema de 2 atributos esta en FNBC. ¿Por qué?

EJEMPLO 14:

Veremos un ejemplo en donde no se puede hallar una descomposición en FNBC que sea SPDF

DIRECCION (BARRIO, CIUDAD, CPOSTAL)

Dependencias Funcionales:

CPOSTAL \twoheadrightarrow CIUDAD

BARRIO, CIUDAD \twoheadrightarrow CPOSTAL

Claves:

{CPOSTAL, BARRIO}

{BARRIO, CIUDAD}

La dependencia funcional CPOSTAL \twoheadrightarrow CIUDAD viola la FNBC, pero ninguna descomposición de este esquema va a preservar la dependencia funcional BARRIO, CIUDAD \twoheadrightarrow CPOSTAL

En forma más general, dados $R(A,B,C)$ y $F = \{AB \twoheadrightarrow C, C \twoheadrightarrow A \text{ o } C \twoheadrightarrow B\}$ no es posible hallar una descomposición de R que esté en FNBC y que sea SPDF.

EJEMPLO 15:

De descomposición sin usar algoritmos:

Volvemos a considerar el Ejemplo 1:

FACULTAD (LU, NOMBRE, MATERIA, IFEC, EFEC, NOTA)

F:

LU \twoheadrightarrow NOMBRE (el nro. de libreta es único)

LU, MATERIA \twoheadrightarrow IFEC (se puede inscribir una sola vez en cada materia)

LU, MATERIA, EFEC \twoheadrightarrow NOTA (hay una sola nota por examen)

Clave: {LU, MATERIA, EFEC}

No está ni en 2FN

Descomponemos:

ALUMNO (LU, NOMBRE), clave: {LU}, está en FNBC

RESTO (LU, MATERIA; IFEC; EFEC; NOTA), clave: {LU,MATERIA,EFEC}, no está ni en 2FN

Volvemos a descomponer:

INSCRIPTO (LU, MATERIA, IFEC) clave: {LU, MATERIA}, está en FNBC

EXAMEN (LU, MATERIA, EFEC, NOTA), clave: {LU,MATERIA,EFEC}, está en FNBC

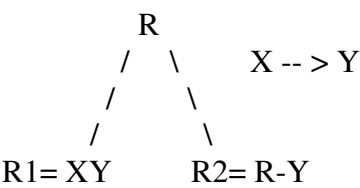
Vemos que ρ = (ALUMNO, INSCRIPTO, EXAMEN) está en FNBC

¿ ρ es SPI? ¿ ρ es SPDF?

5.7. ALGORITMO PARA OBTENER UNA DESCOMPOSICION EN FNBC SPI

Este algoritmo obtiene una descomposición ρ de R partiendo R aplicando la propiedad de descomposición binaria. El ρ resultante es siempre SPI, pero a veces no es SPDF.

Si hay una dependencia funcional $X \twoheadrightarrow Y$ que viola FNBC se parte R en R1 y R2 de la siguiente forma:

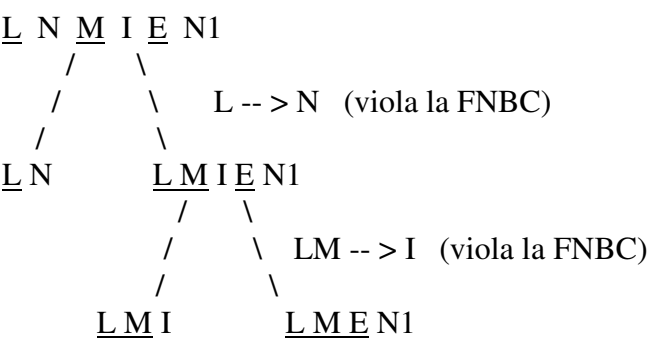


Vemos que esta descomposición es SPI porque aplica la regla de descomposición binaria:
 $R1 \cap R2 = X$
 $R1 - R2 = Y$

EJEMPLO 16:

Volvemos a considerar el Ejemplo 1:

FACULTAD (L, N, M, I, E, N1)
F= {L-->N, LM-->I, LME-->N1}
CLAVE: {L,M,E}



ρ = (LN, LMI, LMEN1) esta en FNBC y es SPI

5.8. ALGORITMO PARA OBTENER UNA DESCOMPOSICION EN 3FN SPI Y SPDF

Este algoritmo obtiene una descomposición ρ de R por SINTESIS a partir de una COBERTURA MINIMAL de F. El ρ resultante es SPI y SPDF.

- 1) Cada dependencia funcional se convierte en un esquema (las que tienen igual lado izquierdo se juntan)
- 2) Si ninguno de los esquemas resultantes contiene una clave se agrega uno con los atributos de alguna clave

EJEMPLO 17:

Veamos otra vez el Ejemplo 1:

FACULTAD (L, N, M, I, E, N1)
F= {L-->N, LM-->I, LME-->N1}
CLAVE: {L,M,E}

F es cobertura minimal porque:

Los lados derechos son de un solo atributo

No hay atributos redundantes en los lados izquierdos: L+ = LN, M+ = M, E+ = E

No hay dependencias funcionales redundantes

Entonces $\rho = (LN, LMI, LMEN1)$ está en 3FN, es SPDF (están todas las dependencias funcionales) y también es SPI porque R3=LMEN1 contiene todos los atributos de la clave.

Nota del docente:

Gran parte de los ejemplos de este apunte fueron tomados de los siguientes libros:

“Introducción a las Bases de Datos Relacionales”, de Alberto Mendelzon y Juan Ale, 2000.

“Principles of Database and Knowledge-Base Systems”, Volume I, de Jeffrey Ullman, 1988.

El algoritmo para hallar todas las claves es una sugerencia de Alejandro Vaisman.