Conjunto de DF’s F = { A 🡪 C, D 🡪 A, CD 🡪 A, DAE 🡪 B } Esquema R(ABCDE)

df = Dependencia funcional

**Cubrimiento mínimo**

**1. Del lado derecho agregamos todo lo que podemos determinar con el lado izquierdo:**

Para ello calculamos la clausura del lado izquierdo

A+ = AC D+=DAC CD+=CDA DAE+=DAEBC

Luego el cubrimiento cerrado en atributos es:

M1 = {A 🡪 AC , D 🡪 DAC, CD 🡪 CDA, DAE 🡪 DAEBC}

**2.Eliminamos dependencias redundantes**, esto es ver si podemos determinar una dependencia si la sacamos del conjunto, **calculando la clausura del lado izquierdo, pero sin su dependencia**:

¿A 🡪 AC es redundante?

**¿Podemos determinar con A en M1 la dependencia A->AC, sin A->AC?**

A+M1\{A->AC} = A ⊉AC, como A no incluye a AC entonces no es redundante.

D+M1\ {D 🡪 DAC} =D ⊉ DAC, como D no incluye DAC entonces no es redundante.

CD+M1\{CD->CDA} =CDA ⊇ CDA, como CDA incluye CDA entonces es redundante

DAE+M1\{DAE->DAEBC} =DAEC ⊉ DAEBC, como DAEC no incluye DAEBC entonces no es redundante

Luego el conjunto sin las dependencias redundantes seria:

M2 = {A 🡪 AC , D 🡪 DAC, DAE 🡪 DAEBC}

**Cubrimiento mínimo Reducido**

**3.Reducimos a izquierda**: Eliminamos atributos extraños a izquierda, sea una df XA->Y, A es extraño a izquierda si podemos generar X->Y.

En los casos de A 🡪 AC , D 🡪 DAC no puede haber extraños a izquierda porque tanto la A como la D están solitas.

Veamos el caso DAE 🡪 DAEBC:

¿D es extraño a izquierda? Vemos que podemos agregar a partir de AE

AE+M2 = AEC ⊉ DAEBC, entonces D no es extraño a izquierda

¿A es extraño a izquierda?

DE+M2 = DEACEB ⊇ DAEBC, entonces A es extraño a izquierda

Modificamos el conjunto:

M3 = {A → AC, D → ACD, DE → ABCDE}

¿E es extraño a izquierda?

DA+M2 = DAC ⊉ DAEBC, entonces E no es extraño a izquierda

El cubrimiento reducido a izquierda queda: M3 = {A → AC, D → ACD, DE → ABCDE}

**4.Reducimos a derecha**: Eliminar los atributos extraños a derecha

Algunos atributos son trivialmente extraños a derecha y pueden eliminarse directamente. Ejemplo: A->AC quedaría A->C

Eliminamos los atributos trivialmente extraños a derecha de M3:

M4 = {A->C, D->AC, DE -> ABC}

**En D->AC ¿A es extraño a derecha?** **“Tachamos” la A y vemos que generamos con D**

D+m4\{D->AC} u {D->C} = DC ⊉ AC, entonces A no es extraño a derecha

Lo que quiere decir D+M4\{D->AC} u {D->C} es que vamos a ver que se puede generar con D sin la A en la derecha en el conjunto M4.

Resumido para ver si un atributo (letra) es extraño a derecha, lo “tachamos” en la dependencia y vemos que se genera con el conjunto sin ese atributo.

En D->AC ¿C es extraño a derecha?

D+M4\{D->AC} u{D->A} =DAC ⊇ AC, entonces C es extraño a derecha porque sin C puedo seguir determinando lo mismo

M5 = {A->C, D->A, DE -> ABC}

Continuamos con DE->ABC

EN DE->ABC ¿A es extraño a derecha?

DE+M5\{DE->ABC} u{DE->BC} = DEABC ⊇ ABC, A es extraño a derecha

M6 = {A->C, D->A, DE ->BC}

¿B es extraño a derecha?

DE+M5\{DE->BC} u{DE->C} = DECA ⊉ BC

¿C es extraño a derecha?

DE+M5\{DE->BC} u{DE->B} = DEBAC ⊇ BC, C es extraño a derecha

Luego el **Cubrimiento Mínimo Reducido** obtenido es:

M7 = {A → C, D → A, DE->B}

**Algoritmo:**

**Cubrimiento mínimo**

1. Del lado derecho agregamos todo lo que podemos determinar con el lado izquierdo

2. Eliminamos dependencias redundantes o sea calculamos la clausura del lado izquierdo, pero sin su dependencia:

**Cubrimiento mínimo Reducido**

3. Reducimos a izquierda, sea una df XA->Y, A es extraño a izquierda si podemos generar X->Y, si lo es modificamos el conjunto y seguimos operando con el conjunto modificado.

4. Reducimos a derecha, sea una df D->AC ¿A es extraño a derecha? “Tachamos” la A y vemos que generamos con D

a.Eliminamos los atributos trivialmente extraños a derecha

**Cálculo de llaves**

Sea F = {KMS 🡪 TN, LP 🡪 MTS, LT 🡪 K, LN 🡪 S, MT 🡪 L, S 🡪 L, KT 🡪 L} un conjunto mínimo reducido definido sobre el esquema R(KLMNPSTU). Para encontrar todas las llaves candidatas distinguimos tres conjuntos de atributos del esquema R(KLMNPSTU):

Creamos tabla para los atributos que siempre van a formar parte de las llaves, talvez formen parte o nunca:

|  |  |  |
| --- | --- | --- |
| Siempre | Talvez | Nunca |
| P U | K M S T N L |  |

En siempre van los atributos que solo aparecen a izquierda o no están en ninguna df. Estos **siempre formaran parte de la llave.**

En talvez van los atributos que aparecen a izquierda y derecha

En nunca solo los que aparecen a derecha

Arrancamos desde los conjuntos más chicos a más grandes:

Calculamos la clausura de PU respecto del conjunto F

PU+F = PU ⊉ R por lo tanto no es llave

3 atributos, hacemos combinaciones con PU que sabemos que tiene que estar en una llave y le agregamos un atributo:

PUK+F = PUK ⊉ R, por lo tanto, no es llave

PUM+F = PUM ⊉ R, por lo tanto, no es llave

P U S+F =P U SLM T KN ⊇ R ⇒ es llave

P U T +F =P U T ⇒ no es llave

P U N +F =P U N ⇒ no es llave

PUL+F = PULMTSKN ⊇ R, por lo tanto, es llave

4 atributos, hacemos combinaciones con los que no son llaves:

P U KM +F =P U KM ⇒ no es llave

P U KN +F =P U KN ⇒ no es llave

P U KT +F =P U KT LM ST N ⇒ es llave

P U M N +F =P U M N ⇒ no es llave

P U M T +F =P U M T LKSN ⇒ es llave

P U N T +F =P U N T ⇒ no es llave

Combinaciones de 5 atributos que no contengan una llave de 4 atributos:

P U KM N +F =P U KM ⇒ no es llave

P U KM T, P U KN T y P U M N T no se consideran porque contienen las llaves P U KT y P U MT, es decir, son super llaves.

Note que no se pueden generar llaves de 6 atributos dado que hay una sola combinación de 5 atributos que no es llave y las demás combinaciones de 5 son super llaves. Luego podemos asegurar que ya se calcularon todas las llaves candidatas.

**Terminamos cuando llegamos a combinaciones que son super llaves de otras**

Las llaves son: P U L, P U S, P U KT, P U M T