

PROBLEMA: Obtengo muchas DFs y no se si están bien, y no puedo descomponer la relación en todas las tablas que necesitaría con esas DFs porque varias DFs son muy parecidas y sería redundante.

PASOS para obtener respuesta a varias preguntas por separado

ZZ.- ¿Cómo encontrar Dependencias Funcionales? → de una relación o de toda la BD

ZZ.a) Reunir todos los atributos necesarios para la relación (puedo poner atributos de toda la BD).

ZZ.b) Para cada atributo, decidir qué DFs hay siguiendo: (llamo F al conjunto de todas las DFs)

- Qué atributo(s) dependen de ese atributo, aplicando la definición de DF
- Influyen la semántica de cada atributo y sus relaciones (pueden ser diferentes DFs)
 - ¿Qué deben cumplir? Relaciones, requisitos, necesidades
- Si no está claro: probar con valores concretos de los atributos.

ZZ.c) Seguir haciendo DFs, probando con combinaciones de atributos: en parejas, en tríos, etc.

A.- ¿Cómo hago el cierre X+ de un cjto de atributos X? → ver las transparencias de Teoría.

Para conseguir todos los atributos que dependen de esos atributos X

B.- ¿Cómo saber si en una DF sobra algún atributo del lado IZQ? (atributo extraño o ajeno)

1.- Toda DF se puede descomponer en varias DFs: cada una con *un solo atributo* en el lado DCHO

AB → CD es equivalente a tener dos nuevas AB → C y AB → D

→ Hacer esto con cada DF que tenga en el lado DCHO más de un atributo

2.- Para saber si sobra algún atributo del lado IZQ de una DF.

Usando el resultado de *paso 1*: p.e., si tengo la DF: ABD → N

2.1 Quito un atributo del lado izq, p.e. el B. y

- Hago el cierre (respecto a las DFs del paso 1.) de los atributos del lado izq que quedan p.e. AB
- Si el lado DCHO, la N, no está en el cierre que acabo de hacer: ese atributo NO sobra
- Si el lado DCHO, la N, sí está en dicho cierre: ese atributo se puede quitar de la DF

2.2 Repito el paso 2.1 con todos los atributos del lado IZQ

→ El resultado de los pasos 1 y 2 nos da la DF *mínima*: muy útil para el resto de los procesos

C.- ¿Cómo saber si sobra alguna DF completa, porque se puede inferir con las otras?

3.- para cada DF del *resultado del paso 2*, hago lo siguiente

3.1 Hago el cierre del lado izq (**sin** usar esa DF)

3.2 Aplicamos esta condición:

- Si en ese cierre está el atributo del lado dcho de la DF: esa DF sobra,
 - quitarla y seguir haciendo 3.1 y 3.2 usando las DFs sin la que hemos quitado
- Si en ese cierre **no está** el atributo del lado dcho de la DF: esa DF se conserva
 - seguir haciendo 3.1 y 3.2 usando las mismas DFs

D.- ¿Cómo obtengo el conjto equivalente *minimal* de F? (llamado también *cierre minimal de F*)

- Busco que **no** sobren atributos **ni** DFs: haciendo B y C (F son todas las DFs)

E.- ¿Cómo saber si un cjto de atributos X son clave principal (CP, o PK) o clave candidata(CC)?

E.1- Aplicamos los pasos 1, 2 y 3 (B y C) a TODAS las DFs, para obtener el *conjto mínimo de DFs*

E.2 - Hacemos (A) el cierre X+ de esos atributos X, usando ese conjto mínimo de DFs obtenido

- Si el resto de atributos de la relación están en X+ : entonces es CP (o PK)

La definición: XYZ es **CP** si tenemos una DF, que sea mínima y sea así :

XYZ → *todos los atrib de la relación*

F.-¿Cómo obtener todas las CCs? Haciendo el paso E.-2 para Combinaciones de atributos:

Combinando atributos y probando si el cierre contiene todos los demás:

- Primero solo un atributo, luego dos que **no** sean ya CC, luego tres, etc.

Algunas CCs pueden obtenerse por este atajo:

- Obtener las CC obvias: haciendo el cierre de los lados izq de cada DF y ver si tienen todos los atrib.
 - Obtener las CC compuestas: añadiendo el atributo que falte del cierre que NO tenga todos los atrib.
- Si un atrib. no está en ningún lado izq, seguro que es parte de una CC.

(SIGUE →) ¿Cómo determinar que tablas o relaciones obtengo?

- Para empezar: tengo todos los atributos en una sola relación inicial
- Encuentro todas las DFs que hay entre ellos. (ver **ZZ**)
- Hago el *cierre minimal de todas las DFs* (ver **D.-** que aplica **B.-** y **C.-**)
- Encuentro una CP (o PK) de la relación inicial (ver **E.-** y **F.-**), necesaria para el último paso.
- Con las DFs del cierre minimal: agrupo en una sola DF, todas las DFs que tengan el lado izq. igual
- creo una tabla o relación con sus atributos para cada una de las DFs agrupadas del paso anterior
- Asigno CP (o PK) a cada relación:
 - o Asigno los atributos del lado izq de la DF correspondiente.
- Último paso: Si la CP que encontré para la relación inicial no tiene todos sus atributos en una sola relación de las nuevas, tengo que crear una relación nueva solo con los atributos de esa CP y todos ellos formarán CP de esa nueva relación.

G.- ¿Cómo saber si una tabla está en 3ª FN?

G-1.- Obtengo el conjunto equivalente minimal de F o cierre minimal haciendo B y C

G-3.- Para cada DF, $X \rightarrow Y$, que he obtenido compruebo si cumple alguna de las dos condiciones:

- *Que X es superclave*
 - Compruebo si el cierre X^+ tiene todos los atrib. de la relación
- *Que Y es parte de una CC (Y es primo)*
 - Compruebo si Y está en alguna de las CCs que obtuve en el paso F-2

H.- ¿Cómo se si una tabla está en FNBC?

- Ninguna DF, $X \rightarrow Y$ no puede cumplir la 2ª condición de 3ª FN: Y no puede ser parte de una CC

I.- ¿Cómo se si se ha perdido alguna DF al descomponer una relación R?

La relación R se descompone en un conjunto de relaciones $S = \{S_1, \dots, S_n\}$

- **Caso 1.-** Si todos los atributos de la DF están en una S_i : la DF se conserva seguro
- **Caso 2.-** Si los atributos de una DF $X \rightarrow Y$ están dispersos entre varias S_i , es *dudosa* y se aplica este Algoritmo para cada DF dudosa por separado:

Entradas:

- una sola DF dudosa: $X \rightarrow Y$
- $S = \{S_1, \dots, S_n\}$ descomposición de un esquema
- F conjunto *minimal* de dep. func. Del esquema original

Salida :

- Resultado: true si S conserva F o false en otro caso.

Algoritmo:

Resultado = true

- 1.- Para una dep. func. *dudosa* $X \rightarrow Y \in F$ hacer

$Z_{Antiguo} = \emptyset$

$Z_{Nuevo} = X$

- 2.- Mientras $Z_{Antiguo} \neq Z_{Nuevo}$ hacer

$Z_{Antiguo} = Z_{Nuevo}$

- 3.- Para $i := 1$ hasta n Hacer

$Z_{Nuevo} := Z_{Nuevo} \cup ((S_i \cap Z_{Nuevo})^+ \cap S_i)$

- 4.- Si $Y \notin Z_{Nuevo}$ Entonces Resultado: *False* . (en caso contrario *True*)