Universidad Nacional de La Matanza

Catedra de Base de Datos

Clase de Normalización (parte 1 de 2)

Introducción a la Normalización

La Normalización es un proceso mediante el cual se puede depurar un diseño de base de datos.

Permite eliminar ciertos defectos y características indeseables de los esquemas.

Para entender bien cuáles son esos defectos que pueden tener los esquemas de datos, vamos a ver a continuación algunos ejemplos de problemas que se pueden presentar.

Introducción a la Normalización

<u>Problemas de un esquema NO Normalizado</u>

Existen muchos problemas o inconvenientes, nosotros solo vamos a ver algunos de ellos:

- REDUNDANCIA
- ANOMALIAS DE ACTUALIZACIÓN
- ANOMALIAS DE INSERCIÓN
- ANOMALIAS DE ELIMINACIÓN
- PERDIDA DE INFORMACIÓN

Introducción a la Normalización

<u>Supongamos el siguiente esquema:</u>

UNIVERSIDAD (dni_alumno, nya_alumno, materia, fecha_inscripción, fecha_examen, nota)

A simple vista se puede ver que este esquema no es correcto.

Vamos a analizar sus problemas…

Introducción a la Normalización

UNIVERSIDAD (dni_alumno, nya_alumno, materia, fecha_inscripción, fecha_examen, nota)

<u>REDUNDANCIA</u>

Si un alumno rindió 10 exámenes, vamos a repetir 10 veces su nombre cuando bastaría con tener solo el DNI.

Si un alumno cursó una materia y luego rindió 3 veces el examen final (hasta aprobarla), vamos a repetir la fecha de inscripción 3 veces.

Esto ocurre porque estamos mezclando información de <u>hechos independientes</u>.

Introducción a la Normalización

UNIVERSIDAD (dni_alumno, nya_alumno, materia, fecha_inscripción, fecha_examen, nota)

<u>ANOMALIAS DE ACTUALIZACIÓN</u>

Si hay que modificar el nombre de un alumno, tendremos que actualizar varias filas (en todas las que figure ese alumno). Mientras que si hubiese estado bien normalizado, el nombre debería estar solo en una fila y solo sería necesario modificar un único valor.

Introducción a la Normalización

UNIVERSIDAD (dni_alumno, nya_alumno, materia, fecha_inscripción, fecha_examen, nota)

ANOMALIAS DE INSERCIÓN

Si tenemos que dar de alta un nuevo alumno, que aun no cursó ni rindió ninguna materia, vamos a tener que dejar con valor NULO los otros atributos.

Idem si tenemos que dar de alta una materia sin alumnos.

Introducción a la Normalización

UNIVERSIDAD (dni_alumno, nya_alumno, materia, fecha_inscripción, fecha_examen, nota)

ANOMALIAS DE ELIMINACIÓN

Si tenemos que eliminar un Alumno y no queda otra fila con esa Materia, se pierde.

O bien, si se decide eliminar a todos los exámenes desaprobados (que tienen nota menor a 4) se eliminaran muchas filas y tal vez se pierdan alumnos que solo tienen exámenes desaprobados.

Introducción a la Normalización

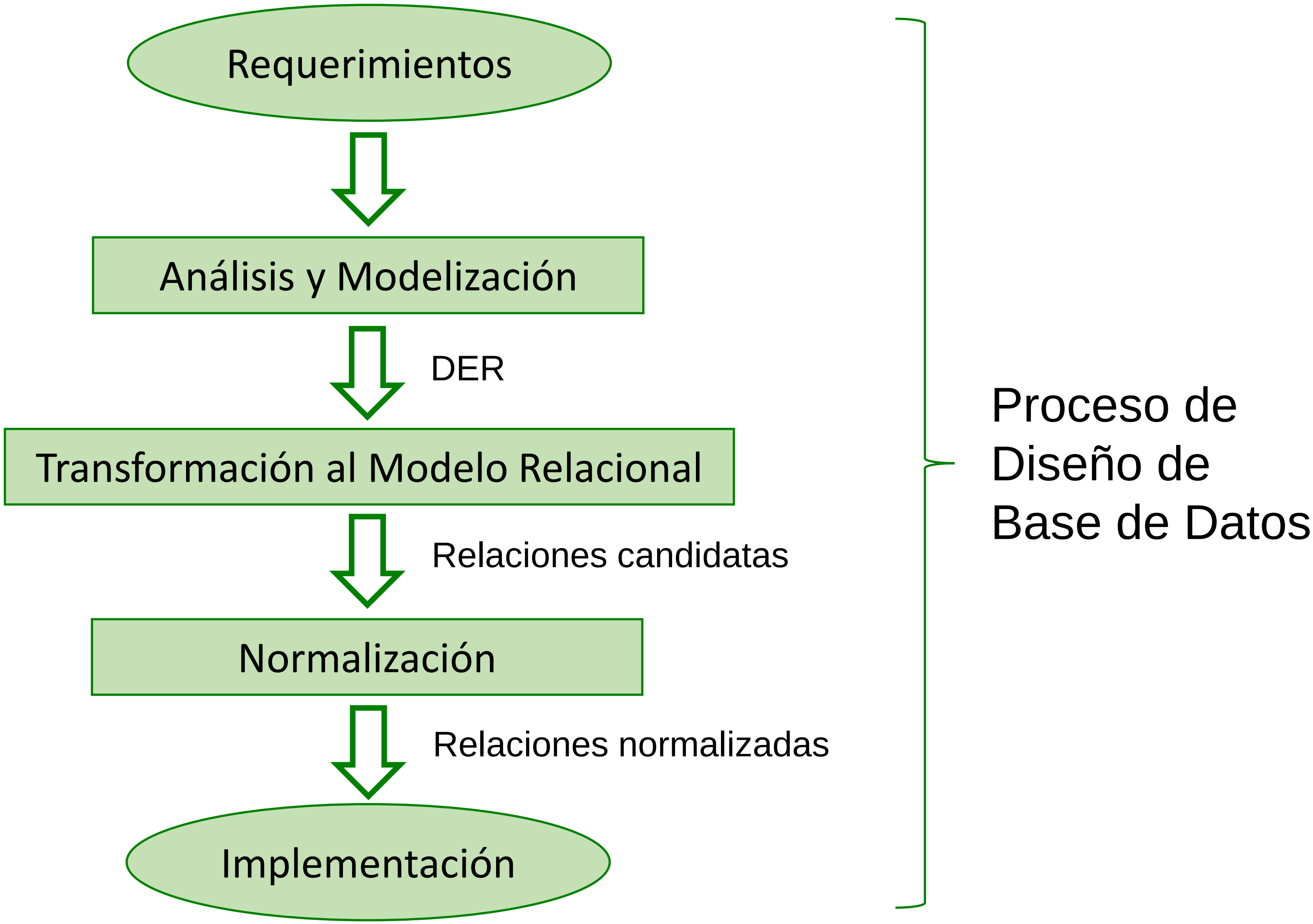
UNIVERSIDAD (dni_alumno, nya_alumno, materia, fecha_inscripción, fecha_examen, nota)

PÉRDIDA DE INFORMACIÓN

Este problema aparece en el caso de una descomposición errónea de un esquema en varios subesquemas.

Mas adelante veremos detalladamente en que consiste la Pérdida de Información.

Introducción a la Normalización



Normalización

DEPENDENCIA FUNCIONAL

Definición:

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.

Normalización

$$X \rightarrow Y$$

Normalización

X determina a Y

$$X \rightarrow Y$$

Determinante
O
Parte Izquierda

Determinado
O
Parte Derecha

Normalización

Ejemplo:

EXAMEN (DNI_Alumno, Nombre, Apellido, Cod_Materia, Nombre_Materia, Fecha_Examen, Nota)

Se cumplen las siguientes dependencias funcionales:

F= { ....

Normalización

Ejemplo:

EXAMEN (DNI_Alumno, Nombre, Apellido, Cod_Materia, Nombre_Materia, Fecha_Examen, Nota)

Se cumplen las siguientes dependencias funcionales:

F= { DNI_Alumno → Nombre, Apellido

Normalización

Ejemplo:

EXAMEN (DNI_Alumno, Nombre, Apellido, Cod_Materia, Nombre_Materia, Fecha_Examen, Nota)

Se cumplen las siguientes dependencias funcionales:

F= { DNI_Alumno → Nombre, Apellido ;

Cod_Materia → Nombre_Materia

Normalización

Ejemplo:

EXAMEN (DNI_Alumno, Nombre, Apellido, Cod_Materia, Nombre_Materia, Fecha_Examen, Nota)

Se cumplen las siguientes dependencias funcionales:

F= { DNI_Alumno → Nombre, Apellido ;

Cod_Materia → Nombre_Materia ;

DNI_Alumno, Cod_Materia, Fecha_Examen → Nota }

Normalización

<u>DEPENDENCIA FUNCIONAL **TRIVIAL**</u>

Decimos que una dependencia funcional es <u>**Trivial**</u> cuando es obvia.

Por ejemplo $X \rightarrow X$

Todo conjunto de atributos se determina a si mismo o a un conjunto de atributos menor (que este contenido en el primero).

$$XY \rightarrow X$$

Normalización

CUESTIONES A TENER EN CUENTA:

Las dependencias funcionales vinculan dos conjuntos de atributos.

Según la teoría de conjuntos:

- El orden de los elementos de un conjunto no importa.

  XY → Z es exactamente igual que YX → Z

- Los conjuntos no pueden tener elementos repetidos.

  no podemos tener XX → Y

  de las dos X automáticamente se suprime una y queda X → Y

Normalización

CLAVE Y SUPERCLAVE

Dado un esquema de relación R(A1, A2, ... An) y un conjunto de dependencias funcionales F asociado, se dice que el conjunto de atributos X perteneciente a R es una **CLAVE** de 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 el conjunto de atributos Y es **SUPERCLAVE** de R si cumple la condición 1.

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

Normalización

CLAVE Y SUPERCLAVE

Ejemplo:

EMPLEADO (Legajo, Nombre, DNI)

Claves = {

Normalización

CLAVE Y SUPERCLAVE

Ejemplo:

EMPLEADO (Legajo, Nombre, DNI)

Claves = { Legajo ;
DNI }

Normalización

CLAVE Y SUPERCLAVE

Ejemplo:

EMPLEADO (Legajo, Nombre, DNI)

Claves Candidatas = { Legajo ;
DNI }

Normalización

CLAVE Y SUPERCLAVE

Clave Primaria
Primary Key (PK)

Ejemplo:

EMPLEADO (Legajo, Nombre, DNI)

Claves Candidatas = { Legajo ;
DNI }

El Diseñador de la Base de Datos tiene que elegir a una de las claves candidatas como Clave Primaria

Normalización

<u>CLAVE Y SUPERCLAVE</u>

Ejemplo:

EMPLEADO (<u>Legajo</u>, Nombre, DNI)

Claves Candidatas = { Legajo ;
DNI }

Superclaves = {

Normalización

CLAVE Y SUPERCLAVE

Ejemplo:

EMPLEADO (Legajo, Nombre, DNI)

Claves Candidatas = { Legajo ;
DNI }

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

Normalización

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

Normalización

REGLAS ADICIONALES/DERIVADAS

Descomposición

Dada $X \rightarrow YZ$ se pueden inferir $X \rightarrow Y$ y $X \rightarrow Z$

Unión

Dadas $X \rightarrow Y$ y $X \rightarrow Z$ se puede inferir $X \rightarrow YZ$

Pseudotransitividad

Dadas $X \rightarrow Y$ y $WY \rightarrow Z$ se puede inferir $WX \rightarrow Z$

Normalización

REGLAS ADICIONALES/DERIVADAS

Las Reglas Adicionales se pueden demostrar a partir de los Axiomas de Armstrong.

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

Ver las otras demostraciones en el apunte de Normalización…

Normalización

CLAUSURA DE UN CONJUNTO DE DEPENDENCIAS (F^+)

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

Normalización

CLAUSURA DE UN CONJUNTO DE DEPENDENCIAS (F^+)

Ejemplo: R(A, B, C) F={ A→B, BC→A }

F^+={

Todas las dependencias Triviales

+

Las dependencias de F

+

Dependencias inferidas

Normalización

CLAUSURA DE UN CONJUNTO DE DEPENDENCIAS (F^+)

Ejemplo: R(A, B, C) F={ A→B, BC→A }

F^+={ A→A,

B→B,

C→C,

Normalización

CLAUSURA DE UN CONJUNTO DE DEPENDENCIAS (F^+)

Ejemplo: R(A, B, C) F={ A→B, BC→A }

F^+={ A→A,
B→B,
C→C,
AB→AB

CLAUSURA DE UN CONJUNTO DE DEPENDENCIAS (F^+)

Ejemplo: R(A, B, C) F={ A→B, BC→A }

F^+={ A→A,

B→B,

C→C,

AB→AB, AB→A, AB→B,

Normalización

CLAUSURA DE UN CONJUNTO DE DEPENDENCIAS (F^+)

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,

CLAUSURA DE UN CONJUNTO DE DEPENDENCIAS (F^+)

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

Normalización

CLAUSURA DE UN CONJUNTO DE DEPENDENCIAS (F^+)

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,

CLAUSURA DE UN CONJUNTO DE DEPENDENCIAS (F^+)

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,

BC→A

CLAUSURA DE UN CONJUNTO DE DEPENDENCIAS (F^+)

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

BC→A

Normalización

CLAUSURA DE UN CONJUNTO DE DEPENDENCIAS (F^+)

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

BC→A

CLAUSURA DE UN CONJUNTO DE DEPENDENCIAS (F^+)

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→BC

BC→A

Normalización

CLAUSURA DE UN CONJUNTO DE DEPENDENCIAS (F^+)

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→BC, AC→ABC

BC→A

CLAUSURA DE UN CONJUNTO DE DEPENDENCIAS (F^+)

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→BC, AC→ABC, AC→AB, AC→B,

BC→A

Normalización

<u>CLAUSURA DE UN CONJUNTO DE DEPENDENCIAS (F^+)</u>

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→BC, AC→ABC, AC→AB, AC→B,
BC→A, BC→AB, BC→AC, BC→ABC
}

Normalización

CLAUSURA DE UN CONJUNTO DE DEPENDENCIAS (F^+)

Ejemplo: R(A, B, C) F={ A→B, BC→A }

Azul: Triviales
Rojo: Provenientes de F
Verde: Inferidas

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→BC, AC→ABC, AC→AB, AC→B,
BC→A, BC→AB, BC→AC, BC→ABC
}

Normalización

CLAUSURA DE UN CONJUNTO DE ATRIBUTOS (X^+)

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

Normalización

<u>CLAUSURA DE UN CONJUNTO DE ATRIBUTOS (X^+)</u>

Ejemplo:

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

$A^+_F = AB$

$B^+_F = B$

$C^+_F = C$

$AB^+_F = AB$

$AC^+_F = ABC$

$BC^+_F = ABC$

$ABC^+_F = ABC$

Normalización

CLAUSURA DE UN CONJUNTO DE ATRIBUTOS (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 F+ fácilmente, (haciendo todas las combinaciones)

Normalización

EQUIVALENCIA ENTRE DOS CONJUNTOS DE DEPENDENCIAS FUNCIONALES

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

$$F \equiv G \quad \Leftrightarrow \quad F^+ = G^+$$

Calcular F^+ y G^+ puede ser muy largo y tedioso…

Normalización

EQUIVALENCIA ENTRE DOS CONJUNTOS DE DEPENDENCIAS FUNCIONALES

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 ($G \subseteq F^+$).

$$F^+ = G^+$$

$F \subseteq G^+$ (G cubre a F)

$G \subseteq F^+$ (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 $Y \subseteq X^+_F$

Normalización

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?

$$F \stackrel{?}{\equiv} G \longrightarrow F^+ \stackrel{?}{=} G^+$$

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

$A^+_G = A\mathbf{BC}$

$B^+_G = B\mathbf{A}C$

$D^+_G = D\mathbf{A}BC$

$D^+_F = D\mathbf{ABC}$

$B^+_F = B\mathbf{AC}$

$A^+_F = A\mathbf{B}C$

Respuesta: Sí, F y G son equivalentes.

Normalización

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→A por otra Y→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.

Normalización

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.

Algoritmo para el Cálculo de Fmin

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.

Normalización

Algoritmo para el Cálculo de Fmin

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

Cálculo de Fmin

Ejemplo:

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

Paso 1)

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

Normalización

Cálculo de Fmin

Paso 2)

AB→D

A^+F_1 = ABC**D**E

B^+F_1 = B

=> Reemplazar por A→D

AB→E

A^+F_1 = ABCD**E**

B^+F_1 = B

=> Reemplazar por A→E

BE→A

B^+F_1 = B

E^+F_1 = E

BE→C

B^+F_1 = B

E^+F_1 = E

F_2 = {A→B, A→C, A→D, A→E, BE→A, BE→C }

Normalización

Cálculo de Fmin

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

Paso 3)

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

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

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