

Optimización del Diseño de BBDD Relacionales. Normalización

En el modelo relacional, el diseño de una base de datos, se puede abordar de dos formas distintas:

1. Obteniendo el esquema relacional directamente del universo del discurso (UD), es decir, de la observación directa de la realidad **obtenemos todos los atributos y dependencias**.
2. Realizando en dos fases el diseño del esquema relacional:
 - 1º. Diseño conceptual, por ejemplo **Modelo Entidad/Relación**
 - 2º. Transformar el modelo E/R en un esquema relacional, es decir, pasar de E/R a modelo relacional mediante unas determinadas reglas de transformación **(paso a tablas)**

En cualquiera de los dos casos se pueden presentar distintos problemas y nosotros lo que queremos es:

- Evitar redundancias
- Evitar problemas de actualización
- Proteger la integridad de los datos

Para minimizar estos posibles problemas vamos a estudiar un método que nos permita llegar a un esquema relacional correcto:

➤ TEORÍA FORMAL DE LA NORMALIZACIÓN DE ESQUEMAS RELACIONALES

Dado un conjunto A de atributos, y el conjunto de dependencias funcionales existentes entre ellos, que constituyen un esquema de relación $R(A, DF)$ (esquema origen), se trata de transformar, mediante sucesivas proyecciones, este esquema de partida en un conjunto de n esquemas de relación (esquemas resultantes) $\{R_i(A_i, DF_i)\}_{i=1}^n$ que cumplan unas determinadas condiciones.

Queremos un conjunto de esquemas $\{R_i\}$ que sea **equivalente** a R y lo **mejore**

Para que los esquemas sean equivalentes se ha de cumplir:

- 1º. Conservación de la información
- 2º. Conservación de las dependencias

Además deberá darse

- 3º. Mínima redundancia de los datos (normalización de las relaciones)

TEMA 4: Optimización del Diseño de BBDD Relacionales. Normalización.

Si las relaciones resultantes están en formas normales más avanzadas que R, se dice que $\{R_i\}$ es mejor que R.

Otros objetivos son:

- Minimización de dependencias, tanto en el número de dependencias como los atributos que hay en ellas.
- Minimización de los esquemas resultantes, tanto en número como en los atributos que hay en ellos.

Más fácil:

ESQUEMA

ORIGEN

R (DNI, Nombre, Apellidos, Teléfono, codasig, nombreAsig, nota)

DNI → Nombre, Apellidos, Teléfono

A cada DNI le corresponde un único Nombre

A cada DNI le corresponde un único Apellidos

A cada DNI le corresponde un único Teléfono

Codasig → nombreAsig

DNI, Codasig → nota

Esto está en solo 1FN (Primera Forma Normal)

R (DNI, Nombre, Apellidos, Teléfono, codasig, nombreAsig, nota)

1111	Pepe	Pérez	686777444	1	BBDD	7
1111	Pepe	Pérez	686777444	2	Marcas	6
1111	Pepe	Pérez	686777444	3	Sistemas	9
2222	Ana	Martín	656222111	1	BB DD	6
2222	Ana	Martín	656222111	2	Marcas	8
2222	Ana	Martín	656222111	3	Sistemas	8
1111	Pepe	Pere	686777443	4	FOL	6

Esta manera de guardar la información de los datos de los alumnos junto en la misma tabla que las asignaturas y sus notas NO es buena manera porque veo información repetida y que además puede estar mal.

NO ES UN BUENA RELACION (TABLA)

Con lo que ya hemos ido viendo somos capaces de “pensar” una manera mejor de guardar esta información.

La idea va a ser deducir que una Relación (Tabla) no está en la mejor manera posible y la vamos a **descomponer** en otras relaciones o tablas que mejoren la anterior.

TEMA 4: Optimización del Diseño de BBDD Relacionales. Normalización.
ESQUEMA
RESULTANTE
R1-Alumnos (DNI, Nombre, Apellidos, Teléfono)

DNI → Nombre, Apellidos, Teléfono

1111	Pepe	Pérez	686777444
2222	Ana	Martín	656222111

R1 está en 1FN, 2FN, 3FN y BCFN

R2-Asignaturas (codasig, nombreAsig)

Codasig → nombreAsig

1	BBDD
2	Marcas
3	Sistemas
4	FOL

R2 está en 1FN, 2FN, 3FN y BCFN

R3-Notas (DNI, codasig, nota)

DNI, Codasig → nota

1111	1	7
1111	2	6
1111	3	9
2222	1	6
2222	2	8
2222	3	8
1111	4	6

R3 está en 1FN, 2FN, 3FN y BCFN

A partir de ahora vamos a “descomponer” relaciones en otras “mejores”

ESQUEMA RESULTANTE es equivalente y mucho mejor porque sus relaciones están en formas normales más avanzadas.

Seguimos...

➤ **CONSERVACIÓN DE LA INFORMACIÓN**

La información contenida en la relación origen R debe ser la misma que la contenida en las relaciones resultantes { R_i }, es la llamada equivalencia de datos.

Para ello se debe cumplir:

- Conservación de atributos: La unión de atributos de { R_i } serán los atributos de R
- Al menos una de las claves candidatas de la relación original se encuentra en una de las relaciones resultantes R_i

TEMA 4: Optimización del Diseño de BBDD Relacionales. Normalización.

➤ **CONSERVACIÓN DE LAS DEPENDENCIAS**

El conjunto de dependencias funcionales de R debe ser equivalente a la unión de los conjuntos de dependencias funcionales de $\{R_i\}$

Es decir, el cierre de DF y el cierre de la unión de DF_i coinciden

$$(U DF_i)^+ = DF^+$$

Volvemos al Ejemplo

R(dni, nombre, codasig, nomasig, nota)

DF={ dni -> nombre
 codasig -> nomasig
 dni, codasig -> nota }

Supongamos que descomponemos en R1 (dni, nombre, nota) y R2 (codasig, nomasig, nota)

En R1 quedarían las d.f. {dni -> nombre}

En R2 quedarían las d.f. {codasig -> nomasig}

Perdemos la dependencia dni, codasig -> nota

Si hacemos R1 (dni, nombre), R2 (codasig, nomasig) y R3 (dni, codasig, nota) no perdemos dependencias.

En R1 {dni -> nombre}

En R2 {codasig -> nomasig}

En R3 {dni, codasig -> nota}

Más fácil:

Ejemplo:

R(A,B,C,D,E,F,G,H)

A->B,C

D->E

E->F

A,D -> G,H

Veamos qué sucede en las siguientes descomposiciones:

A) R1 (A,B,C,D)

R2(D,E,F)

R3(A,D,G)

¿Están todos los atributos que estaban en R?

Me he dejado la H, no está en ninguna relación. ¡¡¡ Horror !!!

No puedo perder atributos

TEMA 4: Optimización del Diseño de BBDD Relacionales. Normalización.

- B) R1 (A,B,C,D) R2(D,E,F) R3(A,D,G,H)
 ¿Están todos los atributos que estaban en R? Sí, están todos.
No he perdido atributos.
- C) R1 (A,B,C) R2(D,E,F) R3(A,D,G,H)
 ¿Están todos los atributos que estaban en R? Sí, están todos.
No he perdido atributos.

El siguiente paso IMPORTANTISIMO es TAMPOCO PERDER D.F.

- B) R1 (A,B,C,D) R2(D,E,F) R3(A,D,G,H)
 ¿Están todos los atributos que estaban en R? Sí, están todos.
No he perdido atributos.
 ¿Están todas las d.f.?
 A->B,C se quedaría en R1

 D->E se quedaría en R2
 E->F se quedaría en R2

 A,D -> G,H se quedaría en R3

No he perdido dependencias funcionales.

Esta descomposición sería:

R1 (A,B,C,D)	R2(D,E,F)	R3(A,D,G,H)
A->B,C	D->E	A,D -> G,H
	E->F	

NO QUIERE DECIR QUE ESTA DESCOMPOSICION SEA LA MEJOR, a lo mejor hay otras mejores, pero por ahora tengo una descomposición que no pierde atributos ni pierde dependencias funcionales.

Ejemplo:

R (A,B,C,D,E,F) ESQUEMA ORIGEN

A->B

A->C

C,D ->E

E->F

Claves candidatas: **(A,D)**⁺=A,D,B,C,E,F

TEMA 4: Optimización del Diseño de BBDD Relacionales. Normalización.
FORMA 1: ESQUEMAS RESULTANTES
R1 (A,B,C)
A->B
A->C
R2 (C,D,E)
C,D ->E
R3 (E,F)
E->F

¿Pierdo atributos? No, no pierdo

¿Pierdo d.f.? No, están todas.

¿Está la clave de la relación original en alguna de las relaciones resultantes?
NO, entonces no va a ser una buena descomposición

FORMA 2: ESQUEMAS RESULTANTES
R1 (A,B,C,D)
A->B
A->C
R2 (E,F)
E->F

¿Pierdo atributos? No, no pierdo

¿Pierdo d.f.? Pierdo **C,D ->E**

¿Está la clave de la relación original en alguna de las relaciones resultantes?

Sí está, en R1

Para que una transformación sea buena se tiene que dar:
1º Conservación de la información:

* No perder ningún atributo

* La clave de la relación original tiene que estar en alguna de las relaciones resultantes.

2º Conservación de las dependencias funcionales

Seguimos...

Ejercicio 1:

Dada la relación R(A,B,C,D) con DF={ A ->B C -> D}

Comprobar si se conserva la información y las dependencias si descomponemos en:

- | | | | |
|----|---------|-----------|---------|
| a. | R1(A,B) | R2(C,D) | |
| b. | R1(A,B) | R2(C,D) | R3(A,C) |
| c. | R1(A,D) | R2(B,C,D) | |
| d. | R1(A,C) | R2(A,B) | R3(A,D) |

TEMA 4: Optimización del Diseño de BBDD Relacionales. Normalización.
Ejercicio 2:

Dada la relación $R(A,B,C,D, E)$

con $DF=\{A \rightarrow B, C \quad D \rightarrow A, E\}$

Comprobar si se conserva la información y las dependencias si descomponemos en:

- $R_1(A,C) \quad R_2(A,D,B,E)$
- $R_1(A,B) \quad R_2(A,C) \quad R_3(A,D,E)$
- $R_1(A,D,E) \quad R_2(A,B,C)$

Ejercicio 3:

Dada la relación $R(A,B,C,D,E)$ con dependencias

$DF=\{B \rightarrow A \quad B \rightarrow C, E \quad E \rightarrow D \quad E \rightarrow C\}$

Hemos decidido descomponerla en las siguientes relaciones

$R_1(A,B,E) \quad R_2(C,D,E)$

¿Hay pérdida de información? ¿Y de dependencias?

Ejercicio 4:

$R(A,B,C,D,E,F,G,H)$

$A \rightarrow B, C$

$B \rightarrow A$

$D \rightarrow E, F$

$A, D \rightarrow G, H$

Hacer un estudio de lo que sucede si se realizan las siguientes descomposiciones.

- $R_1(A,B) \quad R_2(A,C) \quad R_3(A,D,G) \quad R_4(D,E,F,H)$
- $R_1(A,B,C,D) \quad R_2(E,F,G,H)$
- $R_1(A,B,C) \quad R_2(D,E,F) \quad R_3(A,D,G,H)$
- $R_1(A,B,C,D) \quad R_2(D,E,F) \quad R_3(G,H) \quad R_4(A,D)$

TEMA 4: Optimización del Diseño de BBDD Relacionales. Normalización.

➤ **PROCESO DE NORMALIZACIÓN**

Una relación está en una determinada **forma normal** si satisface unas condiciones específicas.

Los problemas de diseño, inserción, actualización, borrado, van solventándose según tengamos una relación en una forma normal más avanzada.

Al proceso de descomponer cualquier relación en dos o más relaciones se llama proceso de normalización y consiste sustituir la relación original por otras sin perder información y procurando no perder dependencias funcionales.

Las relaciones resultantes estarán en formas normales más avanzadas que la original, lo que nos llevará a:

- Evitar redundancias
- Evitar problemas de actualización
- Proteger la integridad de los datos

➤ **FORMAS NORMALES**
Primera Forma Normal (1FN)

Una relación se encuentra en 1FN si no contiene atributos multivalor, es decir, si no hay ninguna fila que tenga algún atributo con más de un valor.

DNI	Nombre	Apellidos	Teléfono
1111111F	Pepe	Pérez	686133334
2222222G	Ana	Luarca	675331188

DNI	Nombre	Apellidos	Teléfono
1111111F	Pepe	Pérez	686133334 644666222 632222333
2222222G	Ana	Luarca	675331188

Atributo
Multivalor

En las tablas con las que trabajamos solo podemos “meter” un valor en cada “hueco”

En nuestras tablas (modelo Relacional) NO Podemos meter atributos multivalor.

TEMA 4: Optimización del Diseño de BBDD Relacionales. Normalización.

En el modelo relacional esto no puede darse nunca.

En el modelo relacional cualquier relación está siempre en al menos 1FN

Definición:

Atributo principal o atributo clave es aquel que pertenece a alguna clave candidata de la relación

Atributo no principal o atributo no clave es aquel que no pertenece a ninguna clave candidata de la relación.

A partir de ahora cuando tengamos una relación y se nos pregunte en qué forma normal está tenemos que hallar sus atributos clave y no clave.

Ejemplo 1:

$R(A,B,C,D)$ con $DF=\{A \rightarrow B \quad C \rightarrow D\}$

Clave : (A, C)

At. Clave o principal: A, C

At. NO clave o no principal : B, D

Segunda Forma Normal (2FN)

Una relación se encuentra en 2FN si:

- Está en 1FN
- Cada atributo no principal depende funcionalmente de toda la clave primaria y no de solo una parte de la clave.

En el ejemplo anterior

La clave es (A, C). Tengo que mirar si los atributos NO clave dependen de TODA la clave o dependen de solo una parte de la clave.

B es un atributo no principal

¿B depende de toda la clave o solo de una parte de la clave?

B solo depende de A

Si encuentro algún atributo no principal que depende de SOLO una parte de la clave NO ESTÁ en 2FN

$R(A,B,C,D)$ con $DF=\{A \rightarrow B \quad C \rightarrow D\}$ **está solo en 1 FN** y **no está en 2FN**

TEMA 4: Optimización del Diseño de BBDD Relacionales. Normalización.
Ejemplo 2:

R(Dni, Nombre, Teléfono, Dirección)

Dni - > Nombre, Teléfono, Dirección

Teléfono - > Dni

Claves candidatas: (Dni) (Teléfono)

2FN porque cualquier atributo no principal depende de toda la clave en su totalidad

Ejemplo 3:

R(dni, nombre, codasig, nomasig, nota)

DF={ dni -> nombre
codasig -> nomasig
dni, codasig -> nota }

Clave (dni, codasig)

No está en 2FN porque el atributo *nombre* depende de una parte de la clave (*dni*) y no de la clave en su totalidad. Lo mismo sucede con *nomasig*

Ejercicio 5:

Comprobar si están en 2FN las siguientes relaciones:

a. R(A,B,C,D,E)

DF = { A -> B,C D -> E }

b. R(A,B,C,D,E)

DF = { A,B -> C,D D -> E }

c. R(A,B,C,D,E)

DF = { A,B -> C,D D -> E E -> F }

d. R(A,B,C,D)

DF = { A, B -> C, D C,D -> A,B }

e. R(A,B,C,D,E,F,G,H)

DF = { A -> B, C B -> A D -> E,F A,D -> G,H }

TEMA 4: Optimización del Diseño de BBDD Relacionales. Normalización.

Observaciones:

- Una relación binaria siempre está en 2FN
- Una relación cuyas claves son simples (un sólo atributo) está siempre en 2FN
- Una relación en la que todos sus atributos son principales está en 2FN

Proceso de Normalización:

Cuando tenemos una relación en 1FN y no en 2FN podemos "normalizar" haciendo una nueva relación con las dependencias funcionales que están provocando que la relación no esté en 2FN.

Por ejemplo, en nuestro ejemplo

R(dni, nombre, codasig, nomasig, nota) 1FN
 DF={ dni -> nombre
 codasig -> nomasig
 dni, codasig -> nota }

Si hacemos: R1 (Dni, Nombre) dni -> nombre 2FN
 R2(codasig, nomasig) codasig -> nomasig 2FN
 R3 (dni, codasig, nota) dni, codasig -> nota 2FN

Tercera Forma Normal (3FN)

Una relación se encuentra en 3FN si:

- Está en 2FN
- No existen dependencias funcionales entre atributos no principales (no clave).
 Es decir, ningún atributo no principal depende de otro atributo no principal.

Ejemplo 1:

R(A,B,C) A->B,C
 Clave (A) Está en 2FN
 At. No clave (B,C)
 ¿existen d.f. entre B y C? No, entonces está en 3FN

Ejemplo 2:

R(A,B,C)
 A->B B->C
 Clave (A)
 Está en 2FN
 At. No clave (B,C)

Sí existen d.f. entre B y C, luego NO está en 3FN

TEMA 4: Optimización del Diseño de BBDD Relacionales. Normalización.**Ejemplo 3:**

R(Dni, Nombre, Dirección, Cod Postal)

Dni - > Nombre, Dirección

Dirección - > Cod Postal

Claves candidatas: (Dni)

2FN

3FN no porque hay dependencias funcionales entre atributos no clave

Ejemplo 4:

R(A,B,C,D,E,F)

A -> B

B -> A

C->D,E

D,E -> C

Hallar todas las claves candidatas: (A,C,F) (B,C,F) (A,D,E,F) (B,D,E,F)

At. No clave: Ninguno

Está en 2FN pq no hay at. No clave.

¿Está en 3FN?

¿existen d.f. entre atributos no clave? Claro, no existen d.f. pq no hay at. No clave con lo cual SÍ está en 3FN.

Si NO HAY at. No clave, no podrán "fastidiar" la 2FN ni la 3FN, luego estará en al menos 3FN.

Ejercicio 6:

Comprobar en el ejercicio 5 que relaciones están en 3FN

Normalizar para pasar a 3FN las relaciones que no lo estén.

TEMA 4: Optimización del Diseño de BBDD Relacionales. Normalización.
Forma Normal de Boyce-Codd (FNBC o BCFN)

Una relación está en FNBC si está en 1FN y cualquier determinante (parte izquierda de una dependencia funcional) es clave candidata.

Ejemplo 1:
 $R(A,B,C,D)$
 $DF = \{ A, B \rightarrow C, D \quad C, D \rightarrow A, B \}$

Claves (A,B) (C,D) Está en BCFN

Ejemplo 2:
 $R(A,B,C,D)$
 $DF = \{ A, C \rightarrow D \quad A \rightarrow B \quad B \rightarrow A \}$

Claves (A,C) (B,C) NO Está en BCFN

Observaciones:

- Una relación en BCFN está siempre en 3FN y por lo tanto en 2FN
- Una relación en 3FN no siempre está en BCFN
- El paso de 3FN a BCFN puede originar pérdida de dependencias.

Ejemplo:

 $R(\text{cod_estd}, \text{cod_prof}, \text{materia})$
 $\text{cod_estd}, \text{materia} \rightarrow \text{cod_prof}$
 $\text{cod_prof} \rightarrow \text{materia}$

(suponemos que un profesor solo da una materia pero hay varios profesores que dan la misma materia)

claves candidatas (cod_estd, materia) (cod_estd, cod_prof)

Está en 3FN pero no en BCFN porque cod_prof no es clave candidata

Para que pasar a BCFN tendríamos que hacer

 $R1(\text{cod_prof}, \text{materia}) \text{ cod_prof} \rightarrow \text{materia}$
 $R2(\text{cod_estd}, \text{cod_prof})$ sin dependencias

Se ha perdido $\text{cod_estd}, \text{materia} \rightarrow \text{cod_prof}$ y no es posible deducirla

TEMA 4: Optimización del Diseño de BBDD Relacionales. Normalización.
Ejercicio 7:

Dadas las siguientes relaciones con sus correspondientes dependencias funcionales, decir en qué forma normal se encuentran explicando claramente por qué está en esa forma normal y no en la siguiente

- | | | | | |
|-------------------------|---|-----------------------|---------------------|-------------------------------------|
| a. $R(A,B,C,D,E,F)$ | $A \rightarrow B$ | $B \rightarrow A$ | $C \rightarrow D,E$ | $E \rightarrow F$ |
| b. $R(A,B,C,D,E)$ | $A,B \rightarrow C,D$ | $C,D \rightarrow A,B$ | $A \rightarrow E$ | $E \rightarrow A$ |
| c. $R(A,B,C,D,E,F)$ | $A,B \rightarrow C,A,B \rightarrow D,E$ | $D,E \rightarrow F$ | | |
| d. $R(A,B,C,D,E,F,G,H)$ | $A \rightarrow B,C,E$ | $B,E \rightarrow A$ | $D \rightarrow F$ | $G \rightarrow H$ $H \rightarrow G$ |

Ejercicio 8:

		CLAVES Candidatas	Forma Normal	¿Por qué?
a.	$R(A,B,C,D,E,F)$ $A \rightarrow B$ $B \rightarrow A$ $C \rightarrow D,E$ $D,E \rightarrow C$ $A \rightarrow F$			
b.	$R(A,B,C,D,E,F)$ $A \rightarrow B$ $B \rightarrow A$ $C \rightarrow D,E$ $D,E \rightarrow C$			
c.	$R(A,B,C,D,E,F)$ $A,B \rightarrow D$ $D \rightarrow A,B,C$ $C \rightarrow E,F$			
d.	$R(A,B,C,D,E,F)$ $A,C \rightarrow B$ $B \rightarrow E,F,D$			
e.	$R(A,B,C,D,E,F)$ $A \rightarrow B$ $B \rightarrow A$ $A,C \rightarrow D,E$ $E \rightarrow F$			
f.	$R(A,B,C,D,E)$ $A, B \rightarrow C,D$ $C,D \rightarrow E$ $E \rightarrow A,B$			

TEMA 4: Optimización del Diseño de BBDD Relacionales. Normalización.
Proceso de Normalización

Dada una relación $R(A, DF)$ queremos que esta esté en la forma normal más avanzada posible.

Si no está en FNBC podemos aplicar el siguiente método para pasar a un conjunto de relaciones $R_i(A_i, DF_i)$ que sea equivalente al original, conserve la información y las dependencias y cada R_i esté en FNBC o al menos 3FN.

- 1º. Se agrupan dependencias que tengan el mismo determinante
- 2º. Se establece una relación R_i por cada grupo, con los atributos y dependencias implicadas
- 3º. Si existen atributos independientes, que no aparecen en ninguna d.f., se forma una relación con ellos.
- 4º. Si la clave de la relación inicial no se encuentra en ninguna R_i , se añade esta como una relación.

Ejemplo 1:
ESQUEMA
ORIGEN

$R(\text{dni, nombre, dir, tfno, codasig, nomasig, nota})$

$DF=\{ \text{dni} \rightarrow \text{nombre}$

$\text{dni} \rightarrow \text{dir}$

$\text{dni} \rightarrow \text{tfno}$

$\text{codasig} \rightarrow \text{nomasig}$

$\text{dni, codasig} \rightarrow \text{nota} \}$

Clave (dni, codasig)

1FN

Normalizamos:

ESQUEMA
RESULTANTE

$R1(\text{dni, nombre, dir, tfno})$

$DF=\{ \text{dni} \rightarrow \text{nombre, dir, tfno} \}$

Clave (dni) BCFN

$R2(\text{codasig, nomasig})$

$DF=\{ \text{codasig} \rightarrow \text{nomasig} \}$

Clave (codasig) BCFN

$R3(\text{dni, codasig, nota})$

$DF=\{ \text{dni, codasig} \rightarrow \text{nota} \}$

Clave (dni, codasig) BCFN

TEMA 4: Optimización del Diseño de BBDD Relacionales. Normalización.

Hemos pasado de R a R1, R2 y R3 con sus correspondientes dependencias funcionales.

R1, R2 y R3 están en BCFN

No se han perdido dependencias ni información.

Cuando hacemos una descomposición (normalizamos) tenemos que:

1. No perder información
 - a. No perder atributos
 - b. No perder la clave de la relación original
2. No perder dependencias funcionales
3. Las relaciones resultantes tienen que estar en BCFN o al menos 3FN

Ejemplo 2:

R(A,B,C,D,E,F)

DF={ A,B → C,D D→E E→F }

Clave (A,B) 2FN

Normalizamos:

R1 (A,B,C,D)
DF={ A,B→C,D }
Clave (A,B) BCFN

R2 (D,E)
DF={ D→E }
Clave (D) BCFN

R3(E,F)
DF={ E→F }
Clave (E) BCFN

Hemos pasado de R a R1, R2 y R3 con sus correspondientes dependencias funcionales.

R1, R2 y R3 están en BCFN

TEMA 4: Optimización del Diseño de BBDD Relacionales. Normalización.

No se han perdido dependencias ni información.

Ejemplo 3:

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

$DF=\{ A,B \rightarrow C \quad D \rightarrow E \quad D \rightarrow F \}$

Clave (A,B,D) 1FN

Normalizamos:

$R_1(A,B,C)$
 $DF=\{ A,B \rightarrow C \}$
 Clave (A,B) BCFN

$R_2(D,E,F)$
 $DF=\{ D \rightarrow E, F \}$
 Clave (D) BCFN

$R_3(A,B,D)$
 $DF=\{ \}$
 Clave (A,B,D) BCFN

Hemos añadido R3 para no perder información (la clave de la relación original).

Ejemplo 4:

Ahora vamos a ver un caso particular de una relación que NO se puede descomponer y llegar a BCFN sin perder dependencias funcionales.

$R(A,B,C) \quad A \rightarrow B$
 $C, B \rightarrow A$

Claves candidatas: (C,B) (C,A) At. No clave: ninguno

Está en 3FN pq no hay at. No clave que anulen ni la 2FN ni la 3FN

¿Está en BCFN? No, no está.

Vamos a intentar descomponer para llegar a BCFN.

TEMA 4: Optimización del Diseño de BBDD Relacionales. Normalización.

Una forma:

R1 (A,B) A->B BCFN

R2 (C,B)

Otra forma:

R1 (A,B) A->B BCFN

R2 (C,A)

De cualquiera de estas dos maneras perdemos la d.f. C,B -> A

Estamos ante un caso en el que si llegamos a BCFN es perdiendo d.f.

Ejercicio 8:

Normalizar hasta llegar a BCFN

- a. R (A, B, C, D, E, F) DF = { A -> B C,A -> D B,D -> E E -> F }
- b. R(A,B,C,D,E,F,G,H) DF={ A->C C->D,H E,F->B B->E,F,G }
- c. R (A, B, C, D, E, F, G) DF = { A -> B,F A,C -> D, E C,B -> AF -> G }

NORMALIZACIÓN por SINTESIS

Partimos ahora de la observación directa de la realidad, normalmente de un enunciado del que deducimos atributos y dependencias funcionales.

Los pasos a seguir serán los siguientes:

- 1º. Se determinan atributos y dependencias funcionales
- 2º. Se eliminan las dependencias funcionales redundantes
- 3º. Se determina la clave de la relación original
- 4º. Se agrupan dependencias que tengan el mismo determinante
- 5º. Se establece una relación R_i por cada grupo, con los atributos y dependencias implicadas
- 6º. Si existen atributos independientes, que no aparecen en ninguna d.f., se forma una relación con ellos.
- 7º. Si la clave de la relación inicial no se encuentra en ninguna R_i , se añade esta como una relación.

Quedarán las relaciones en al menos 3FN

No habrá pérdida de información ni de dependencias funcionales

Puede producirse cierta redundancia

EJERCICIOS NORMALIZACIÓN POR SÍNTESIS

1. Disponemos de la siguiente información para mantener la base de datos de un taller de coches. Por un lado tenemos los mecánicos, caracterizados por un **DNI_MEC** y de los que guardamos su nombre **NOM_MEC** y función principal **FUN** en el taller. Por cada coche que llega se guarda su matrícula **MAT**, el dni del dueño, **DNI_DU**, nombre y teléfono del dueño, **NOM_DU**, **TF_DU**. Cada **DNI_DU** tiene asignado un número de cliente **NUM_CLI**. Un propietario podría llevar al taller distintos coches. Cuando un coche llega al taller se le asigna un mecánico para que arregle la avería y se guarda en qué **FECHA** se produce la avería. Un coche puede venir al taller en distintas ocasiones para ser arreglado y por cada una de estas ocasiones se hará una factura con su número de factura **NUM_FAC** y un determinado importe **IMPORTE**
2. Se está celebrando un campeonato de atletismo y gimnasia en diversas instalaciones municipales de la ciudad. Participan deportistas de varias Comunidades Autónomas a los que se les ha asignado un **DORSAL**, independientemente de la prueba o pruebas en las que compitan, que los identifica durante todo el campeonato. De los deportistas conocemos su **DNI**, **NOMBRE**

TEMA 4: Optimización del Diseño de BBDD Relacionales. Normalización.

y **COMUNIDAD** de la que provienen. Hay varias pruebas y de ellas se conoce un código **COD_PRUEBA**, una descripción **DESC**, el día en que se realiza **DIA**, el tipo al que pertenece **TIPO_PRUEBA** y el **LUGAR** en el que se lleva a cabo. Los deportistas podrán participar en más de una prueba pero nunca en el mismo día. En un lugar y día determinado podrán realizarse varias pruebas. El control de las competiciones lo realizan árbitros de los que conocemos el dni **DNI_ARB**, nombre **NOM_ARB**, su comunidad **COMUNIDAD** y el tipo de prueba que arbitran. Una misma prueba puede tener más de un árbitro y un árbitro serlo de varias pruebas incluso en el mismo día, pero un árbitro sólo arbitra un tipo de prueba.

3. Un parque de atracciones dispone de zonas separadas donde hay diversas atracciones, cada zona tiene un código **COD_Z** que la identifica, el nombre **NOM_Z** que será único y el número de atracciones de que dispone **NUM**. Una atracción tiene un número **NUM_A**, un nombre **NOM_A** y una fecha de instalación **FEC_I** y la atracción está en una única zona. El parque de atracciones tiene contratados operarios de mantenimiento que tienen un **NUM_E** que le caracteriza, un **DNI_E** que será único y un nombre de empleado **NOM_E**. Cada empleado se dedica al mantenimiento de varias atracciones de una misma zona y cuando se estropea una atracción se guardará la fecha de la avería **FEC_AV** y la fecha del arreglo **FEC_AR** y el empleado que la arregla