

d-



UD 3-2. DISEÑO LÓGICO. PARTE 2

Base de Datos
CFGs DAW

Francisco Aldarias Raya
paco.aldarias@ceedcv.es

2019/2020

Fecha 27/11/19

Versión:191127.1111

Licencia



Reconocimiento - NoComercial - Compartirlgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:



Importante



Atención



Interesante

Revisiones

7/11/2019. Ejemplo 1fn.

ÍNDICE DE CONTENIDO

1. INTRODUCCIÓN.....	4
2.DEPENDENCIA FUNCIONAL.....	4
2.1 Dependencia funcional.....	4
2.2 Dependencia funcional completa.....	5
2.3 Dependencia funcional transitiva.....	5
3.3. NORMALIZACIÓN.....	6
3.1 Primera Forma Normal - 1FN.....	6
3.2 Crear el diagrama de dependencias.....	9
3.3 Segunda Forma Normal - 2FN.....	9
3.4 Tercera Forma Normal - 3FN.....	11
3.5 Eliminación de Redundancias.....	13

UD03-2. DISEÑO LÓGICO. PARTE 2

1. INTRODUCCIÓN

□ La normalización es un proceso de refinamiento para comprobar la calidad de nuestro modelo verificando que las relaciones o tablas del modelo Relacional obtenido no tienen redundancias ni inconsistencias.

Posiblemente las primeras veces nuestro modelo Relacional cambie algo al ser normalizado, pero poco a poco os daréis cuenta que al ir creando vuestro modelo de datos, inconscientemente, ya estaréis aplicando la normalización a lo largo del proceso y al final, al comprobar si está normalizado, comprobaréis que cumple todas las reglas necesarias.

El proceso de normalización consiste en la aplicación de una serie de reglas que nos sirven para verificar, y en algunas ocasiones modificar, nuestro modelo Relacional.

Para ello, vamos a aplicar las siguientes fases de la normalización (existen más fases pero se salen de los objetivos de este curso):

- Primer forma normal - 1FN
- Segunda forma normal - 2FN
- Tercera forma normal - 3FN

El proceso de normalización se realizará de la siguiente forma, primero comprobaremos que todas nuestras relaciones o tablas están en 1FN y si no es así realizaremos los cambios necesarios para que así sea. Una vez lo tenemos todo en 1FN aplicaremos las reglas para ver si están en 2FN y así sucesivamente. De tal forma que si decimos que nuestro modelo está en 3FN ya sabemos que cumple 1FN, 2FN y 3FN.

Antes de comenzar a ver las reglas de la normalización debemos conocer el concepto de dependencia funcional y sus tipos.

2. DEPENDENCIA FUNCIONAL

2.1 Dependencia funcional

□ Un atributo B depende funcionalmente de otro atributo A si a cada valor de A le corresponde un único valor de B. Se dice que A implica B o lo que es lo mismo $A \rightarrow B$. En este caso A recibe el nombre de implicante.

Por ejemplo en la siguiente tabla Personas podemos tener:

PERSONAS (DNI, Nombre, Fecha_Nacimiento);

Podemos decir que $DNI \rightarrow Nombre$, ya que un DNI se corresponde con un único Nombre, es decir, a través del DNI podemos localizar el nombre de la persona a la que pertenece.

Luego cuando tenemos un atributo A que implica B ($A \rightarrow B$) es lo mismo que decir que B depende funcionalmente de A.

2.2 Dependencia funcional completa

— Dado un conjunto de atributos A formado por a_1, a_2, a_3, \dots (estos atributos formarán una clave primaria compuesta) existe una dependencia funcional completa cuando B depende de A pero no de un subconjunto de A.

Es decir que para obtener B son necesarios todos los elementos del conjunto A, o dicho de otra forma B depende de todos los atributos que forman la clave primaria A y no puede depender solamente de parte de ellos.

Por ejemplo cuando tenemos una tabla de Compras de la siguiente forma:

COMPRAS (Referencia_Producto, Código_Proveedor, Dirección_Proveedor, Cantidad, Precio);

En este caso nuestra clave primaria está formada por dos campos, la Referencia_Producto y el Código_Proveedor. Pues bien, debemos comprobar si todos los demás atributos tienen una dependencia funcional de toda la clave primaria.

Podemos observar que la Dirección_Proveedor, no dependerá de ambos, sino solamente del Código_Proveedor, luego para ese atributo no existiría dependencia funcional completa.

Sin embargo el resto de los campos, tanto la Cantidad, como el Precio acordado dependen del conjunto, es decir, dependen completamente de la clave primaria, luego aquí sí tenemos una dependencia funcional completa.

2.3 Dependencia funcional transitiva

— Este tipo de dependencia implica a tres atributos. Cuando existe una dependencia $A \rightarrow B$ y a su vez tenemos que $B \rightarrow C$, podemos decir que existe una dependencia funcional transitiva entre A y C.

Por ejemplo, si tenemos una tabla de Productos como la siguiente:

PRODUCTOS (Referencia_Producto, Nombre, Precio, Stock, Fabricante, País)

Aquí podemos encontrar que el Fabricante depende de la Referencia_Producto, es decir.

$Referencia_Producto \rightarrow Fabricante$

Y por otro lado el País también puede depender del Fabricante, luego:

Fabricante → País.

Con estas dos dependencias podemos afirmar que el País depende transitivamente de la Referencia_Producto.

3.3. NORMALIZACIÓN

Vamos a ver cómo aplicar las formas normales hasta conseguir que nuestro Modelo Relacional esté normalizado.

3.1 Primera Forma Normal - 1FN.

Es una forma normal inherente al esquema relacional, por lo que su cumplimiento es obligatorio; es decir toda tabla realmente relacional la cumple.

Una relación R esta en 1FN, si todos los dominios tienen valores atómicos y no es derivada.

Se deberá eliminar:

- 1.- Los grupos repetitivos.
- 2.- La información derivada.

Eliminación de grupos repetitivos:

Supongamos que tenemos la siguiente tabla Centros.

Código_Centro	Nombre	Localidad	Teléfono
45005467	IES Ribera del Tajo	Talavera de la Reina	925722233 - 925722804
45005239	IES Azarquiel	Toledo	925267843 - 925637843
36003748	IES El Plantío	Huelva	973847284

Centros (Código_Centro, Nombre, Localidad, {Teléfono})

Podemos observar como en el campo teléfono hay ocurrencias en las que hay más de un valor. Este campo rompe la regla de la 1FN. La forma de resolverlo es creando una nueva tabla formada por el campo que contiene múltiples valores y la clave principal de la tabla. Quedando de la siguiente forma:

Tabla Centros.

Código_Centro	Nombre	Localidad
45005467	IES Ribera del Tajo	Talavera de la Reina
45005239	IES Azarquiel	Toledo
36003748	IES El Plantío	Huelva

Y una nueva tabla que llamaremos Teléfonos.

Código_Centro	Teléfono
45005467	925722233
45005467	925722804
45005239	925267843
45005239	925637843
36003748	973847284

Se puede observar que un centro tiene muchos teléfonos y un teléfono es de un centro como mucho.

Centros (Código_Centro, Nombre, Localidad)

CP:{Código_Centro}

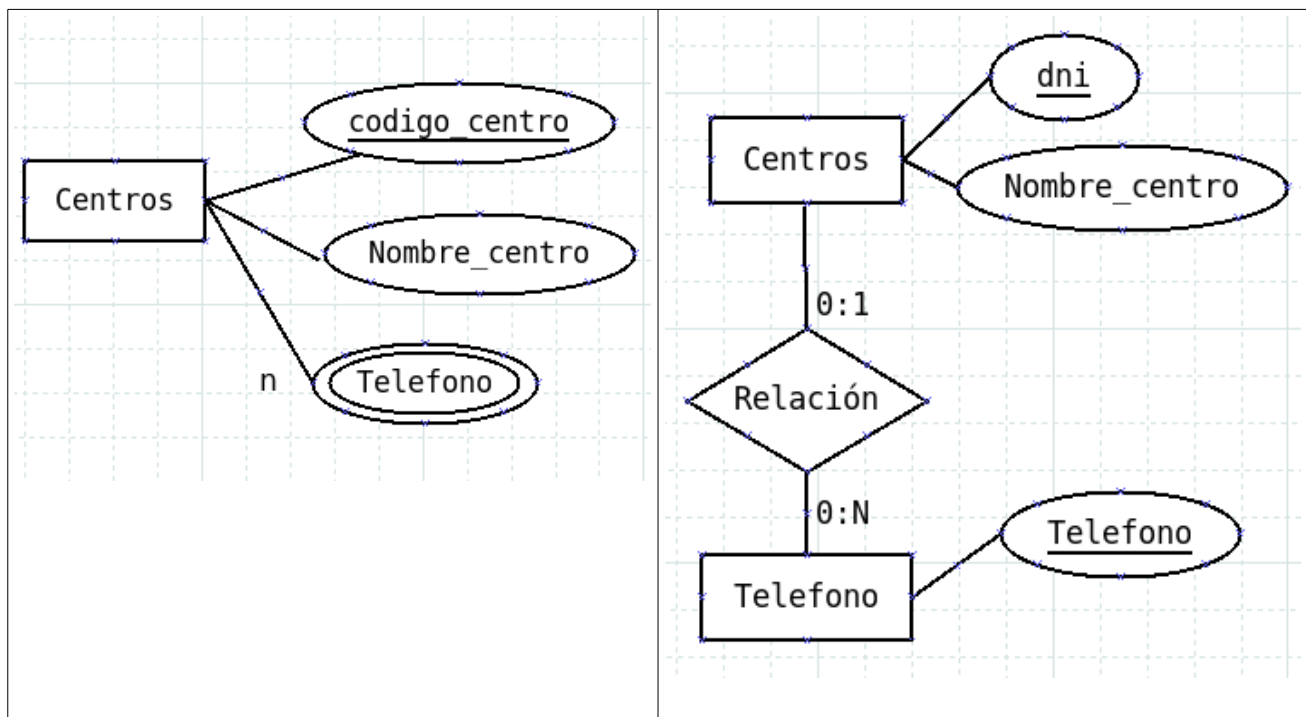
Teléfonos(Teléfono, Código_Centro)

CP:{Teléfono}

CA:{ Código_Centro} → Centros

Gráficamente sería:

No en 1FN	En 1FN
-----------	--------



Notación: Podemos expresar la relación $E(\underline{a1}, a2)$ como $E = @a1 + a2$.

Este atributo teléfono en la entidad Centros, n-evaluado se podría haber hecho como entidad y una relación, y así estaría en 1FN.

Eliminación de información derivada.

Dada la relación $E: E = @a1 + a2$ cuya Información derivada es que $a2 = a3 + a4$.

Se eliminaría la información derivada sustituyendo $a2$ por $a3 + a4$:

$$E = @a1 + a3 + a4$$

Ejemplo1:

Persona = @dni + fecha-nacimiento + edad

edad = año-actual() - fecha-nacimiento

1FN =

Persona = @dni + fecha-nacimiento

3.2 Crear el diagrama de dependencias

Debemos meter en cajas los atributos que determinan funcionalmente a otros atributos. Es decir, representar las dependencias funcionales, que nos permitirán normalizar con más comodidad.

Cada relación tiene su propio diagrama de dependencias.

Ejemplo:

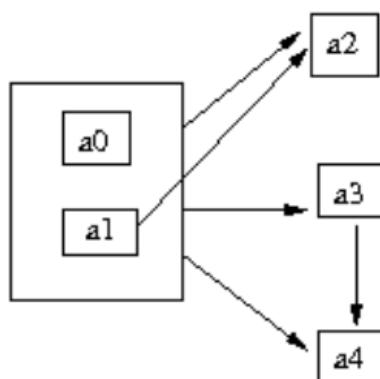
$R = @a0 + @a1 + a2 + a3 + a4$

$a1 \twoheadrightarrow a2$

$a3 \twoheadrightarrow a4$

El diagrama de dependencias funcionales sería:

DIAGRAMA DE
DEPENDENCIAS



3.3 Segunda Forma Normal - 2FN.

Una estructura R está en 2FN, Si está en 1FN., y si todos sus atributos, no clave de R, son plenamente dependientes del conjunto de atributos que conforman la clave.

Ejemplo que no esta en 2FN:

$R = @a0 + @b0 + r1 + r2$

$a0 + b0 \twoheadrightarrow r1, r2$

$b0 \twoheadrightarrow r1$

Las flechas indican que determinan funcionalmente, es decir:

$a0+b0$ determina funcionalmente a $r1$

b0 determina funcionalmente a r1

R no está en 2FN, pq r1 no depende de forma completa de a0+b0, sino que sólo depende de b0.

Paso a 2FN:

Por cada DF no resuelta, se crea una nueva estructura:

R1 = @a0 + @b0 + r2

R2 = @b0 + r1

Las estructuras con claves primarias compuestas, pueden ser sospechosas de no estar en 2FN.
Las estructuras con un atributo en la clave primarias, están siempre en 2FN.

Ejemplo:

Nota = @dni_alumno + @cod_asignatura + nota + nombre_alumno + nombre_asignatura

DF:

@dni_alumno + @codasig \rightarrow nota

@dni_alumno \rightarrow nombrealumno

@cod_asignatura \rightarrow nombre_asignatura

Matricula no esta en 2FN.

Aquí habría repeticiones innecesarias de atributos y daría pie a una incongruencia de la información.

Paso a 2FN:

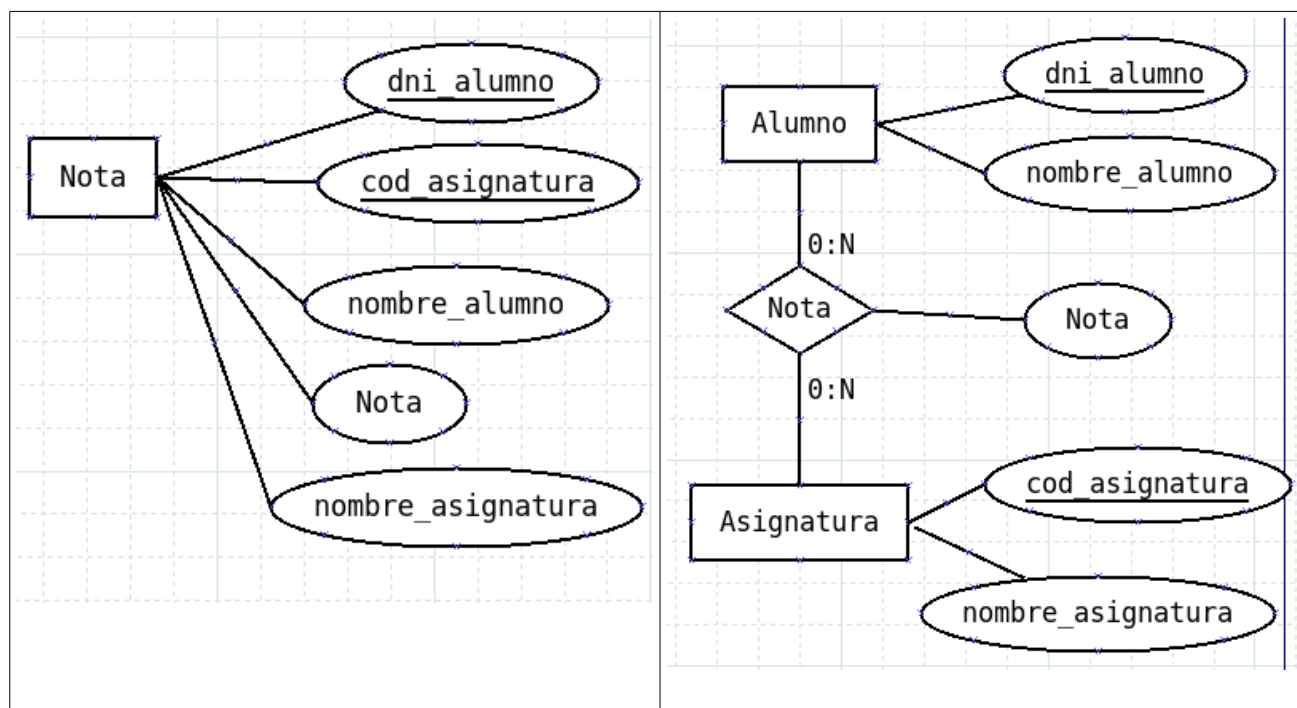
Nota = @dni_alumno + @codasig + nota

Alumno = @dni_alumno + nombre_alumno

Asignatura = @cod_asig + nombre_asignatura

Gráficamente sería:

No en 2FN	En 2FN
-----------	--------



Como se puede apreciar, si el diseño Entidad Relación estuviera bien, no habría que pasarlo a 2FN.

3.4 Tercera Forma Normal - 3FN.

Una estructura R está en 3FN, Si está en 2FN, y si todos los atributos no clave (principal y alternativos) de R, no tienen dependencias funcionales transitivas entre sí o dependencias entre sí.

Estructura q no está en 3FN:

$R = @a_0 + a_1 + a_2 + a_3$

$a_0 \rightarrow a_1, a_2, a_3$

$a_1 \rightarrow a_2$

Aquí existe una dependencia transitiva, ya que hay DF entre atributos no clave.

Paso a 3FN:

$R_1 = @a_0 + a_1 + a_3$

$R_2 = @a_1 + a_2$

Si a_1 fuera clave alternativa, estaría en 3FN, por ser un dependencia natural.

En este caso tendríamos las siguientes DF:

$a0 \rightarrow a1, a2, a3$

$a1 \rightarrow a0, a2, a3$

Ejemplo:

Alumno = @dni_alumno + nombre_alumno + codigo_grupo + nombre_grupo.

codigo_grupo \rightarrow nombre_grupo

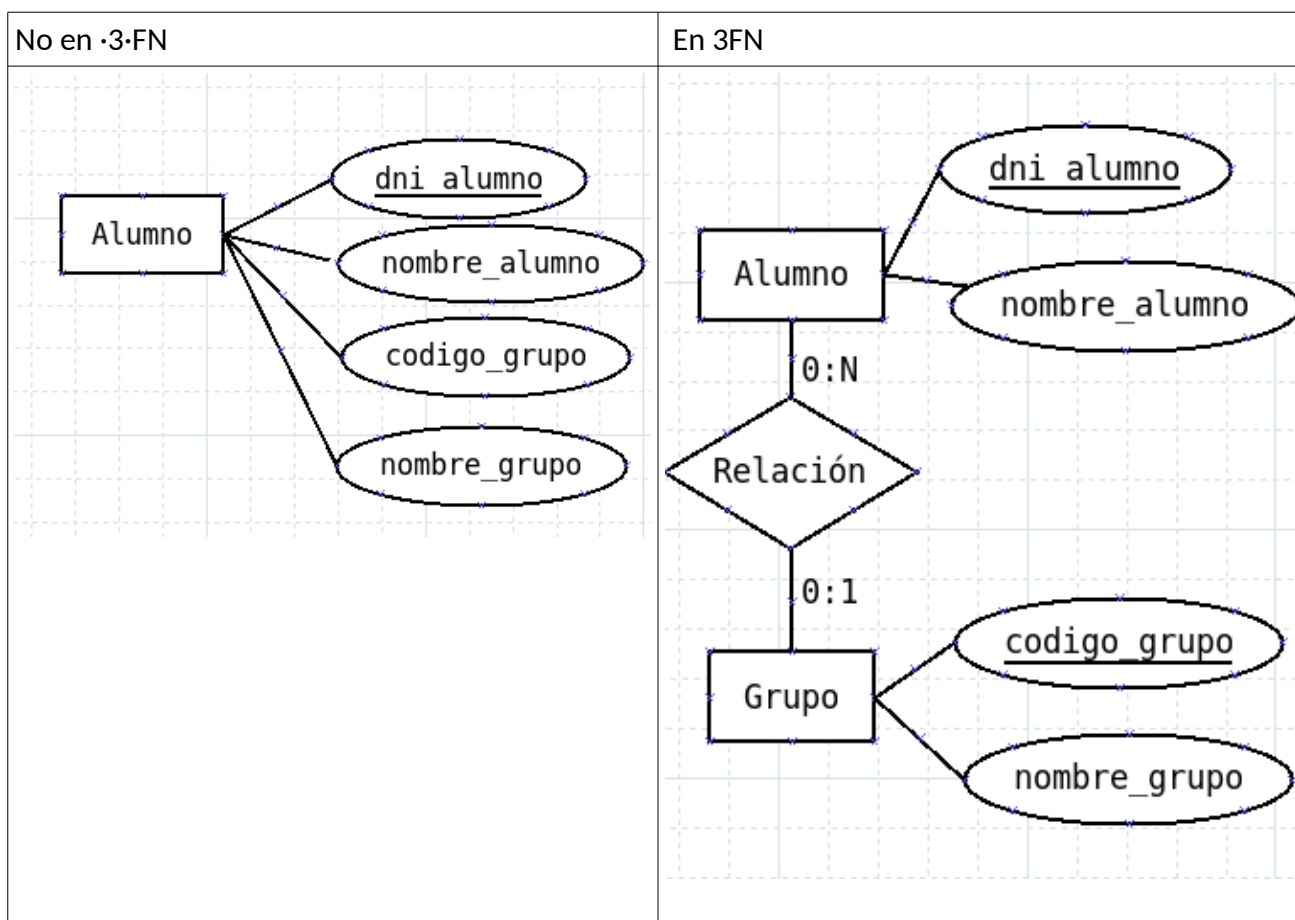
No esta en 3FN, pq hay DF Transitivas.

Quedaría así en 3fn partiendo de que un alumno sólo puede estar en un grupo :

Alumno = @dni_alumno + nombre_alumno + codigo_grupo

Grupo = @codigo_grupo + nombre_grupo

Gráficamente sería:



Si la relación entre Alumno y Grupo fuera N:N entonces habria que crear una tabla nueva

llamada Alumno-Grupo, y no habría la clave ajena `cod_grupo` en Alumno.

3.5 Eliminación de Redundancias

Es esta fase debemos eliminar aquellas relaciones que tengan la misma clave primaria, y atributos duplicados.

Ejemplo:

$A1 = @a0 + x + y$

$A2 = @a0 + x + b$

Después de eliminar redundancias quedaría:

$A = @a0 + x + y + b$