

Normalización del modelo Relacional

Esquema del proceso de Normalización:

1FN

Una tabla está en 1FN si y solo si los valores que componen los atributos de una fila tienen un solo valor.

id_trabajador	nombre	empresa	url	dir_emp
1	Luis	BRG	brg1.com brg2.com	c/Sagasta
2	Andrés	GNC	gnc1.com gnc2.com	c/Lepanto
3	Fernando	BRG	brg1.com brg2.com	c/Sagasta

Tabla 1

Las columnas repetidas deben eliminarse y si es preciso colocarse en tablas separadas.

id_trabajador	nombre	empresa	url_1	url_2	dir_emp
1	Luis	BRG	brg1.com	brg2.com	c/Sagasta
2	Andrés	GNC	gnc1.com	gnc2.com	c/Lepanto
3	Fernando	BRG	brg1.com	brg2.com	c/Sagasta

Tabla 2

Solución para cumplir con la **1FN**: crear un registro de cada trabajador para cada URL de cada empresa.

id_trabajador	nombre	empresa	url	dir_emp
1	Luis	BRG	brg1.com	c/Sagasta
1	Luis	BRG	brg2.com	c/Sagasta
2	Andrés	GNC	gnc1.com	c/Lepanto
2	Andrés	GNC	gnc2.com	c/Lepanto
3	Fernando	BRG	brg1.com	c/Sagasta
3	Fernando	BRG	brg2.com	c/Sagasta

Tabla 3

Aparecen nuevos problemas:

- Cada vez que introducimos un nuevo registro en la tabla usuario hay que duplicar el nombre de la empresa y del usuario.
- La BD crecerá mucho y será fácil que se corrompa si escribimos mal alguno de los datos redundantes.

2FN

Una tabla está en 2FN si y solo si se cumplen dos condiciones:

- Está en 1FN.
- **Todo atributo secundario (atributo que no pertenece a la clave primaria) depende totalmente (tiene una dependencia funcional total) de la clave completa, y por lo tanto, no de una parte de ella.**

NOTA: Se dice que existe una dependencia funcional entre los atributos A y B si para cada valor de A existe un único valor posible de B

Se descompone la tabla en dos:

1. Proyección de la clave primaria junto con los atributos que tienen valores atómicos.
2. Nueva clave con los atributos que tienen valores múltiples (Es necesario una nueva clave primaria).

En nuestro ejemplo la clave primaria estaría compuesta por los atributos **id_trabajador y url** , ya que con esta combinación podríamos identificar unívocamente a cualquier tupla de la tabla.

En el ejemplo , pe, el atributo empresa no depende completamente (de todos los atributos) de la clave principal. (sólo depende de parte de la url)

Hay que separar los grupos de datos relacionados en tablas separadas y relacionadas entre sí mediante uno o varios campos.

id_trabajador	nombre	empresa	dir_emp
1	Luis	BRG	c/Sagasta
2	Andrés	GNC	c/Lepanto
3	Fernando	BRG	c/Sagasta

Nom_empresa	url_empresa
BRG	brg1.com
BRG	brg2.com
GNC	gnc1.com
GNC	gnc2.com

Tabla 4

Ahora se pueden añadir más trabajadores sin tener que repetir los datos de cada URL de la misma. Se usará la clave primaria nombre_empresa para relacionar ambas tablas.

¿Qué ocurre si queremos añadir 200 empleados a una empresa?. Tendríamos el nombre de la empresa y su dirección duplicándose por cada nuevo empleado. →

REDUNDANCIA, OCUPACIÓN DE ESPACIO INNECESARIO Y POSIBLES ERRORES DE COHERENCIA DE DATOS.

3FN

Una tabla está en 3FN si y solo si se cumplen dos condiciones:

- Está en 2FN.
- No existen atributos no primarios (que no forman parte de la clave primaria) que son transitivamente dependientes de clave candidata (cada posible clave primaria de la tabla).

Se descompone la tabla en dos:

1. Una tabla con la clave primaria y todos los atributos no primarios que no son transitivos.
2. Una tabla con los atributos transitivos y el atributo no primario (que será la clave primaria de la nueva tabla).

Una tabla está normalizada en esta forma si todos los atributos que no son clave o parte de la misma son funcionalmente dependientes por completo de la clave primaria y no hay dependencias transitivas.

NOTA: una dependencia transitiva es aquella en la cual existen columnas que no son clave que dependen de otras columnas que tampoco lo son.

Para pasar a 3FN habrá que eliminar todos aquellos campos que no dependan directamente de la clave.

En el ejemplo la dir_empresa no tiene nada que ver con id_trabajador salvo de forma transitiva, así que lo separaremos formando nuevas tablas:

id_trabajador	nombre	empresa
1	Luis	BRG
2	Andrés	GNC
3	Fernando	BRG

Tabla :
Trabajadores

Nom_empresa	url_empresa
BRG	brg1.com
BRG	brg2.com
GNC	gnc1.com
GNC	gnc2.com

Tabla:
empresas_URL

Tabla:
empresas

empresas	dir_empresas
BRG	c/Sagasta
GNC	c/Lepanto

Tabla 5

FNBC

Una tabla está en FNBC si y solo si está en 3FN y las únicas dependencias funcionales son aquellas en que la clave primaria (y las claves secundarias) determinan un atributo.

Una tabla está en FNBC si ningún atributo, forme parte de la clave o no, depende funcionalmente de otro atributo que no sea clave.

Este hecho se produce cuando hay dos posibilidades para elegir clave primaria (o existen dos claves candidatas) y ambas posibilidades tienen atributos en común.

Si ya partimos de la 3FN no puede existir ningún atributo no clave que dependa de otro que a su vez tampoco lo sea. Sólo habría que comprobar si algún atributo clave depende funcionalmente de algún atributo no clave

Se descompone la tabla en dos:

1. Una tabla con todos los atributos menos la parte de la clave dependiente del atributo secundario. La clave primaria esta formada por el resto de la clave y el atributo secundario que tenía una dependencia con parte de la clave.
2. Una tabla con el atributo de la clave dependiente y el atributo secundario, donde el atributo secundario pasa a ser la nueva clave primaria de la tabla.

Hay que comprobar que no se produce pérdida de información al pasar una tabla a FNBC. Si se pierde es mejor dejar el modelo en 3FN.

En nuestro ejemplo está ya en FNBC.

Otro ejemplo:

Proveedor	Producto	Nave	Sector
1	1	100	A
2	1	100	B
2	2	110	X
3	5	120	F
4	6	121	B
5	1	150	A

Tabla 6

Tenemos ciertos productos que se guardan en naves. En cada nave, se guarda un producto en concreto y un producto pertenece a varios proveedores (cada proveedor tiene reservado un espacio dentro de cada nave). Así pues el modelo relacional podría ser:

Almacén (**proveedor**, **producto**, nave, sector)

Este modelo no cumple con la FNBC: hay un atributo clave, *producto* que depende funcionalmente de otro atributo no clave, *nave*. Como dice el enunciado “en cada nave se guarda un producto en concreto”, por lo que un valor de *nave* determina el único valor que puede tomar *producto*.

Solución: sacar fuera de la tabla los atributos involucrados y dejar aquellos que sean necesarios para mantener la relación entre la tabla original y la nueva.

Almacén (**proveedor**, nave, sector)

Nave (**nave**, producto)