

UNIDAD 4. NORMALIZACIÓN

ÍNDICE

- 4.- Normalización
 - 4.1.- Conceptos
 - 4.2.- Formas Normales

4.- Normalización

4.1.- Conceptos

Habitualmente, el diseño lógico de una base de datos termina en el paso del modelo entidad-relación al modelo relacional. No obstante, siempre que se diseña un sistema, no solo una base de datos, sino también cualquier tipo de solución informática, se ha de medir la calidad de la misma, y si no cumple determinados criterios de calidad, hay que realizar, de forma iterativa, sucesivos refinamientos en el diseño, para alcanzar la calidad deseada.

Uno de los parámetros que mide la calidad de una base de datos es la **forma normal** en la que se encuentra su diseño. Esta forma normal puede alcanzarse cumpliendo ciertas restricciones que impone cada forma normal al conjunto de atributos de un diseño. El proceso de obligar a los atributos de un diseño a cumplir ciertas formas normales se llama **normalización**.

La normalización persigue estos objetivos:

- Evitar la **redundancia** de los datos.
- Evitar problemas de **actualización** de los datos en las tablas.
- Proteger la **integridad** de los datos.

Uno de los conceptos fundamentales en la normalización es el de **dependencia funcional**. Una dependencia funcional se da como una *relación entre atributos de una misma relación* (tabla).

Definición:

- Si **X** e **Y** son atributos de la relación **R**, se dice que **Y** es funcionalmente dependiente de **X** (se denota por $X \rightarrow Y$) si cada valor de **X** tiene asociado un solo valor de **Y** (X e Y pueden constar de uno o varios atributos). Esto es lo mismo que decir que **si a todo valor de X le corresponde siempre el mismo valor de Y**.

COD_ALUMNO	ASIGNATURA	NOM_ALUM	APE_ALUM	NOTA
1111	LENGUA I	PEPE	GARCÍA	5
1111	IDIOMA	PEPE	GARCÍA	5
2222	IDIOMA	MARÍA	SUAREZ	7
2222	CIENCIAS	MARÍA	SUAREZ	7
3333	PLÁSTICA	JUAN	GIL	6
3333	MATEMÁTICAS I	JUAN	GIL	6
4444	LENGUA II	FRANCISCO	MONTOYA	4
4444	MATEMÁTICAS I	FRANCISCO	MONTOYA	6
4444	CIENCIAS	FRANCISCO	MONTOYA	8

En este ejemplo, NOM_ALUM y APE_ALUM dependen funcionalmente de COD_ALUM, pero no dependen de ASIGNATURA.

La dependencia funcional es una noción semántica. Si hay o no dependencias funcionales entre atributos no lo determina una serie abstracta de reglas, sino, más bien, los modelos mentales del usuario y las reglas de negocio de la organización o empresa para la que se desarrolla el sistema de información.

En el proceso de normalización se debe ir comprobando que cada relación (tabla) cumple una serie de reglas que se basan en la **clave primaria** y las **dependencias funcionales**. Cada regla que se cumple aumenta el grado de normalización. Si una regla no se cumple, **la relación se debe descomponer en varias relaciones que sí la cumplan**.

4.2.- Formas Normales

Reglas de normalización: Se dice que una relación está en una forma normal determinada si satisface un cierto conjunto específico de restricciones impuestas por la regla de normalización correspondiente.

Primera forma normal. 1FN.

Se dice que una relación está en 1FN **si y sólo si los valores que componen cada atributo de una tupla son atómicos**, es decir, cada atributo de la relación toma un único valor del dominio correspondiente, o lo que es lo mismo no existen grupos repetitivos.

Esta tabla ALUMNO **NO** está en 1FN:

<u>COD_ALUMNO</u>	NOMBRE	APELLIDO	TLF	DIRECCION
1111	PEPE	GARCÍA	678-900800 91-2233441 91-1231232	C/Las cañas 45
2222	MARÍA	SUÁREZ	91-7008001	C/Mayor 12
3333	JUAN	GIL	91-7562324 660-111222	C/La plaza
4444	FRANCISCO	MONTOYA	678-556443	C/La arboleda

Solución: Para conseguir que la tabla esté en 1FN, se eliminan los grupos repetitivos (TLF) y se crea una nueva relación (tabla) junto con la clave inicial, en la que tanto la clave inicial como el TLF formarán la clave de dicha tabla:

TABLA ALUMNO

<u>COD_ALUMNO</u>	NOMBRE	APELLIDO	DIRECCION
1111	PEPE	GARCÍA	C/Las cañas 45
2222	MARÍA	SUÁREZ	C/Mayor 12
3333	JUAN	GIL	C/La plaza
4444	FRANCISCO	MONTOYA	C/La arboleda

TABLA TELÉFONO

<u>COD_ALUMNO(FK)</u>	TLF
1111	678-900800
1111	91-2233441
1111	91-1231232
2222	91-7008001

Segunda forma normal. 2FN.

Se dice que una relación se encuentra en 2FN si y sólo si satisface la 1FN, y **cada atributo de la relación que no está en la clave depende funcionalmente de forma completa de la clave primaria de la relación.**

Esta forma normal tiene sentido desde el momento en que la clave principal es una clave compuesta por más de un atributo. Las tablas (relaciones) con una **clave primaria simple** que están en 1FN, lo están también en 2FN.

COD_ALUMNO	ASIGNATURA	NOM_ALUM	APE_ALUM	NOTA
1111	LENGUA I	PEPE	GARCÍA	5
1111	IDIOMA	PEPE	GARCÍA	5
2222	IDIOMA	MARÍA	SUAREZ	7
2222	CIENCIAS	MARÍA	SUAREZ	7
3333	PLÁSTICA	JUAN	GIL	6
3333	MATEMÁTICAS I	JUAN	GIL	6
4444	LENGUA II	FRANCISCO	MONTOYA	4
4444	MATEMÁTICAS I	FRANCISCO	MONTOYA	6
4444	CIENCIAS	FRANCISCO	MONTOYA	8

Esta relación no estaría en 2FN, puesto que, por ejemplo, el atributo NOM_ALUM no depende funcionalmente de parte de la clave primaria (depende de COD_ALUMNO, pero no de ASIGNATURA).

Otro ejemplo:

COMPRAS (Código-Producto, Código-Proveedor, Nombre-Producto, Cantidad, Fecha-Compra).

En el ejemplo anterior, Nombre-Producto depende funcionalmente sólo de Código-Producto, por lo que esta tabla no estaría en 2FN, ya que no existe dependencia funcional completa.

¿Qué implica esto? Estaríamos, por ejemplo, incurriendo en una redundancia respecto a los datos del producto. Con cada registro de información de compra de un producto estaríamos registrando innecesariamente su nombre, abriendo la posibilidad a registrarlo en alguna ocasión con alguna errata o con alguna diferencia ortográfica.

Solución: Tenemos que hacer lo siguiente, para el atributo o grupos de atributos con dependencia incompleta **M** en la relación **R**, cuya clave primaria (compuesta) es **K**:

- Eliminar de **R** el atributo **M**.
- Crear una nueva relación **N** con el atributo **M** y la parte de la clave primaria **K** de la que depende, que llamaremos **K'**.

En nuestro ejemplo:

COMPRAS (Código-Producto(FK), Código-Proveedor, Cantidad,
Fecha-Compra).
(hemos eliminado Nombre-Producto)

Creamos

PRODUCTOS (Código-Producto, Nombre-Producto)

Ahora el nombre del producto se almacena una única vez, sin riesgos.

Observa que Código-Producto en COMPRAS es ahora Clave Ajena (FK)

Tercera Forma Normal. 3FN

Un diseño se encuentra en 3FN si está en FN2 y además, **no hay ningún atributo no clave que depende de forma transitiva de la clave.**

Recordamos la propiedad transitiva: Si $A \rightarrow B$ y $B \rightarrow C$, entonces $A \rightarrow C$

En general, tenemos que buscar **dependencias transitivas de la clave**, es decir, secuencias de dependencias como la siguiente: $K \rightarrow A$ y $A \rightarrow B$, donde **A** y **B** no pertenecen a la clave.

Ejemplo:

PRODUCTOS (Código-Producto, Nombre, Fabricante, País).

Código Producto \rightarrow Fabricante (**$K \rightarrow A$**)

Fabricante \rightarrow País (**$A \rightarrow B$**)

Y sin embargo, Código-Producto --/--> País (**K**-/->**B**)

País depende transitivamente de **Código-Producto**, por tanto, no está en tercera forma normal.

Dicho de otra forma: Si examinamos los atributos **no-clave**, vemos si alguno de ellos puede repetirse. Si es el caso, vemos si hay algún otro atributo no-clave que tome siempre el mismo valor para las repeticiones del primero. Eso nos va a indicar que ese atributo no-clave depende del anterior, existiendo así una dependencia transitiva de la clave principal que hay que reparar. Míralo en este ejemplo:

<u>Num-socio</u>	Nombre	Ciudad	Comunidad-Aut
1	Juan García	Sevilla	Andalucía
2	Luisa Gómez	Cartagena	Murcia
3	Manuela Pérez	Pamplona	Navarra
4	Pedro Pérez	Pamplona	Navarra
5	Juana Martín	Cádiz	Andalucía

Vemos que en el atributo “Ciudad”, las repeticiones siempre implican la repetición del atributo “Comunidad-Aut” (por ejemplo *Pamplona* y *Navarra*), por lo que este último depende funcionalmente de “Ciudad”. La tabla no está en 3FN.

Solución: Separar en una tabla adicional **N** el/los atributos **B**, y poner como clave primaria de **N** el atributo que define la transitividad (**A**).

PRODUCTOS (Código-Producto, Nombre, Fabricante(FK))
(Eliminamos atributo País, y hacemos Fabricante clave ajena)

FABRICANTES (Fabricante, País)
(Ahora Fabricante es clave ajena en PRODUCTOS,)

O en el otro caso:

SOCIOS (Num-socio, Nombre, Ciudad(FK))

CIUDADES (Ciudad, Comunidad-Aut)

Fijate en que el objetivo de la normalización es evitar **anomalías y redundancias**. Imagina que hubiese 700 socios de Sevilla. Estaríamos introduciendo 700 veces el literal “Andalucía” en cada uno de los registros de socios de Sevilla. Estamos desperdiciando espacio de almacenamiento, por no decir el trabajo que implicaría corregir 700 registros en caso que nos hubiésemos confundido de comunidad autónoma. Una vez normalizada la tabla, sin embargo, sólo necesitamos introducir una vez la relación entre Sevilla y Andalucía.

Otro ejemplo:

Supongamos que tenemos una tabla de ganadores de torneos de tenis.

GANADORES-TORNEOS-TENIS

(Nombre-Torneo, Año, Nombre-Ganador, Nacionalidad)

Esta tabla no está en 3FN porque el atributo **Nacionalidad**, que no es de la clave, depende del **Nombre del Ganador** (aunque también dependa de la clave). Digamos que **nacionalidad aporta información sobre el ganador, pero no sobre la clave**. Es una dependencia transitiva porque **Nacionalidad** depende de **ganador** que a su vez depende de **Torneo-Año**.

¿Cómo solucionaríamos este ejemplo?

Existen otras formas normales (FN de Boyce-Codd, 4FN y 5FN). Dichas FN incluyen conceptos complejos y, puesto que un esquema se considera aceptable cuando cumple 3FN (y por tanto 2FN y 1FN también), no las estudiaremos.

Un ejemplo completo. Normalizar hasta 3FN

nss	nombre	puesto	salario	emails
111	Juan Pérez	Jefe de Área	3000	juanp@ecn.es; jefe2@ecn.es
222	José Sánchez	Administrativo	1500	jsanchez@ecn.es
333	Ana Díaz	Administrativo	1500	adiaz@ecn.es; ana32@gmail.com
...

(Clave primaria: NSS)

1FN. No está en 1FN porque el atributo emails tiene valores múltiples. Separamos en 2 tablas, como vimos:

Empleados (PK: NSS)

nss	nombre	puesto	salario
111	Juan Pérez	Jefe de Área	3000
222	José Sánchez	Administrativo	1500
333	Ana Díaz	Administrativo	1500
...

y emails (PK: nss+email) (Además, nss es ahora FK en esta nueva tabla)

nss	email
111	juanp@ecn.es
111	jefe2@ecn.es
222	jsanchez@ecn.es
333	adiaz@ecn.es
333	ana32@gmail.com
...	...

EMPLEADO (**nss**, nombre, puesto, salario)

EMAIL (**nss(FK),email**)

2FN. Al no existir una clave compuesta en empleados, y al no tener emails más atributos que los de la clave,, nuestro esquema estaría en 2FN.

EMPLEADO (nss, nombre, puesto, salario)

EMAIL (nss(FK),email)

3FN. Una relación está en tercera forma normal si, y sólo si:

- a) está en 2FN
- b) y, además, cada atributo que no está incluido en la clave primaria no depende transitivamente de la clave primaria.

Por lo tanto, a partir de un esquema en 2FN, tenemos que buscar dependencias funcionales entre atributos que no estén en la clave. Nos olvidamos de la tabla email y examinamos empleados.

En general, tenemos que buscar dependencias transitivas de la clave, es decir, secuencias de dependencias como la siguiente: **K**->**A** y **A**->**B**, donde **A** y **B** no pertenecen a la clave. La solución a este tipo de dependencias está en separar en una tabla adicional **N** el/los atributos **B**, y poner como clave primaria de **N** el atributo que define la transitividad (**A**).

Siguiendo el ejemplo anterior, podemos detectar la siguiente transitividad: el salario está íntimamente relacionado con el puesto, ya que es el puesto el que marca un determinado salario. Si cambiásemos el puesto de Juan Pérez, también tendríamos que cambiar su salario.

nss->puesto

puesto->salario

Por lo tanto la descomposición sería la siguiente:

nss	nombre	puesto
111	Juan Pérez	Jefe de Área
222	José Sánchez	Administrativo
333	Ana Díaz	Administrativo
...

puesto	salario
Jefe de Área	3000
Administrativo	1500

(*puesto* es ahora FK en la primera tabla y PK en la nueva)

3FN

EMPLEADO (nss, nombre, puesto(FK))

EMAIL (nss(FK),email)

PUESTO(puesto,salario)