

# UN EJEMPLO SIMPLE DE NORMALIZACIÓN DE BASES DE DATOS RELACIONALES (HASTA 3FN)\*

Miguel-Angel Sicilia

This work is produced by The Connexions Project and licensed under the  
Creative Commons Attribution License †

## Abstract

Se describe un ejemplo sencillo (de una sola tabla) de aplicación de la normalización de bases de datos relacionales.

## 1 El proceso de normalización de bases de datos relacionales

La normalización de bases de datos relacionales toma un esquema relacional y le aplica un conjunto de técnicas para producir un nuevo esquema que representa la misma información pero contiene menos redundancias y evita posibles anomalías en las inserciones, actualizaciones y borrados.

### 1.1 Breve recordatorio del modelo (formal) relacional

El modelo relacional de bases de datos se basa en un modelo formal especificado de acuerdo a la teoría de conjuntos. Una base de datos relacional puede considerarse como un conjunto de relaciones o tablas de la forma  $R(A_1, \dots, A_n)$ , donde  $R$  es el nombre de la relación, que se define por una serie de atributos  $A_i$ .

Sobre las tablas relacionales se pueden definir diferentes restricciones. La integridad de entidad es una restricción que nos indica que cada entidad representada por una tupla tiene que ser diferente de las demás en su relación, es decir, debe haber algunos atributos cuyos valores identifiquen unívocamente las tuplas. La integridad referencial indica que una clave ajena solo debe contener valores que o bien sean nulos, o bien existan en la relación referenciada por la clave ajena.

### 1.2 El proceso de normalización

El proceso de normalización consiste en comprobar en secuencia si el esquema original está en 1FN, 2FN y 3FN, analizando las dependencias funcionales en cada paso.

---

\*Version 1.1: Nov 19, 2008 12:10 pm -0600

†<http://creativecommons.org/licenses/by/2.0/>

### 1.3 Un ejemplo completo

Tenemos una empresa pública donde los puestos de trabajo están regulados por el Estado, de modo que las condiciones salariales están determinadas por el puesto. Se ha creado el siguiente esquema relacional

EMPLEADOS(*nss*, nombre, puesto, salario, emails) con *nss* como clave primaria.

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
...	...	...	...	...

Table 1

### 1.4 Primera forma normal (1FN)

Una tabla está en 1FN si sus atributos contienen valores atómicos. En el ejemplo, podemos ver que el atributo **emails** puede contener más de un valor, por lo que viola 1FN.

En general, tenemos una relación  $R$  con clave primaria  $K$ . Si un atributo  $M$  viola la condición de 1FN, tenemos dos opciones.

#### 1.4.1 Solución 1: duplicar los registros con valores repetidos

En general, esta solución pasa por sustituir  $R$  por una nueva relación modificada  $R'$ , en la cual:

- El atributo  $M$  que violaba 1FN se elimina.
- Se incluye un nuevo atributo  $M'$  que solo puede contener valores simples, de modo que si  $R'[M']$  es uno de los valores que teníamos en  $R[M]$ , entonces  $R'[K] = R[K]$ . En otras palabras, para una tupla con  $n$  valores duplicados en  $M$ , en la nueva relación habrá  $n$  tuplas, que sólo varían en que cada una de ellas guarda uno de los valores que había en  $M$ .
- La clave primaria de  $R'$  es  $(K, M')$ , dado que podrá haber valores de  $K$  repetidos, para los valores multivaluados en  $M$ .

Siguiendo el ejemplo, tendríamos el siguiente esquema para la nueva tabla EMPLEADOS' (**a**) con clave primaria (*nss*, email):

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

Table 2

### 1.4.2 Solución 2: separar el atributo que viola 1FN en una tabla

En general, esta solución pasa por:

sustituir R por una nueva relación modificada R' que no contiene el atributo M.

Crear una nueva relación N(K, M'), es decir, una relación con una clave ajena K referenciando R', junto al atributo M', que es la variante mono-valuada del atributo M.

La nueva relación N tiene como clave (K, M').

Siguiendo el ejemplo, tendríamos el siguiente esquema para la nueva tabla EMPLEADOS' (b)

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

**Table 3**

Y además tendríamos una nueva tabla EMAILS con clave primaria (nss, email):

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

**Table 4**

### 1.5 Segunda forma normal (2FN)

Un esquema está en 2FN si:

Está en 1FN.

Todos sus atributos que no son de la clave principal tienen dependencia funcional completa respecto de todas las claves existentes en el esquema. En otras palabras, para determinar cada atributo no clave se necesita la clave primaria completa, no vale con una subclave.

La 2FN se aplica a las relaciones que tienen claves primarias compuestas por dos o más atributos. Si una relación está en 1FN y su clave primaria es simple (tiene un solo atributo), entonces también está en 2FN. Por tanto, de las soluciones anteriores, la tabla EMPLEADOS' (b) está en 1FN (y la tabla EMAILS no tiene atributos no clave), por lo que el esquema está en 2FN. Sin embargo, tenemos que examinar las dependencias funcionales de los atributos no clave de EMPLEADOS' (a). Las dependencias funcionales que tenemos son las siguientes:

nss->nombre, salario, email

puesto->salario

Como la clave es (nss, email), las dependencias de nombre, salario y email son **incompletas**, por lo que la relación no está en 2FN.

En general, tendremos que observar los atributos no clave que dependan de parte de la clave.

Para solucionar este problema, tenemos que hacer lo siguiente para los grupos de atributos con dependencia incompleta M:

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'.

La clave primaria de la nueva relación será K'.

Siguiendo el ejemplo anterior, crearíamos una nueva relación con los atributos que tienen dependencia incompleta:

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

**Table 5**

Y al eliminar de la tabla original estos atributos nos quedaría:

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

**Table 6**

Como vemos, la solución a la que llegamos es la misma que en la otra opción de solución para el problema de 1FN.

### 1.6 Tercera forma normal (3FN)

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

está en 2FN

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.

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. 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:

**nss  $\rightarrow$  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
...	...	...

**Table 7**

En la nueva tabla PUESTOS, la clave sería el puesto, que también queda como clave ajena referenciando la tabla EMPLEADOS. El resto de las tablas quedan como estaban.