

## Ejercicios de normalización

Supongamos que, en una Base de Datos de una empresa de venta de computadoras, la información sobre la facturación se encuentra almacenada de la siguiente forma:

Nro. Factura	Fecha	Código cliente	Nombre cliente	Código de producto	Descripción	Cantidad	Precio unitario	Total factura
1	10/01/04	10	Juan Pérez	25	Mouse	5	10	
1	10/01/04	10	Juan Pérez	22	Teclado	10	15	
1	10/01/04	10	Juan Pérez	01	CPU	2	650	
1	10/01/04	10	Juan Pérez					<b>1500</b>
2	20/02/01	36	Julián Ail	12	Pad	10	2	
2	20/02/01	03	Julián Ail	99	Monitor	3	200	
2	20/02/01	03	Julián Ail					<b>620</b>

Normaliza a 3FN la siguiente relación.

---

En primer lugar debemos comprobar en qué nivel de normalización nos encontramos (si es que la tabla ya está normalizada). Para ello debemos comprobar desde el nivel más bajo de normalización hacia el nivel superior.

Partimos de la siguiente relación:

FACTURA(nro\_factura, fecha, cod\_cliente, nom\_cliente, cod\_prod, descripción, cantidad, precio\_unitario, total\_factura)

Analizamos que en cada factura puede haber varios códigos de producto, varias descripciones, cantidades, precios unitarios. Por esto, debemos hacer otra relación.

FACTURA(nro\_factura, fecha, cod\_cliente, nom\_cliente, total\_factura)

PRODUCTO(cod\_prod, descripción, precio\_unitario)

FACT\_PROD(nro\_factura, cod\_prod, cantidad)

Por tanto, ahora se cumple la condición anterior y tanto las relaciones como la base de datos se encuentran en 1FN.

**2FN:** Una base de datos está en 2FN si, y sólo sí, está en 1FN y además todos los atributos que no forman parte de la clave principal tienen dependencia funcional completa de ella.

Analizamos las dependencias:

nro\_factura → fecha, cod\_cliente, nom\_cliente, total\_factura

cod\_prod → descripción, precio\_unitario

nro\_factura, cod\_prod → cantidad

Comprobamos que se cumple la condición, por lo que las relaciones ya están normalizadas a 2FN.

FACTURA(nro\_factura, fecha, cod\_cliente, nom\_cliente, total\_factura)

PRODUCTO(cod\_prod, descripción, precio\_unitario)

FACT\_PROD(nro\_factura, cod\_prod, cantidad)

**3FN:** Una base de datos está en 3FN si, y sólo sí, está en 2FN y además no existen dependencias transitivas. Todas las dependencias deben ser respecto de la clave principal.

Analizando las dependencias funcionales anteriores detectamos transitividad en las siguientes dependencias:

nro\_factura → cod\_cliente

cod\_cliente → nom\_cliente

Por tanto, se debe crear una nueva relación para eliminar las dependencias transitivas, obteniendo el siguiente resultado en 3FN:

FACTURA(nro\_factura, fecha, cod\_cliente, total)

PRODUCTO(cod\_prod, desc, precio)

COMPRA(nro\_factura, cod\_prod, cant)

CLIENTE(cod\_cliente, nom\_cliente)

Ahora la base de datos se encuentra normalizada en 3FN, ya que todas sus relaciones lo están.