

**SUMINISTROS** (ARTICULO, PROV, N\_UNIDADES, DESCRIPCIÓN, NOMBRE, F\_PEDIDO, COLOR, TAMAÑO, PAGO, #PAGO, PRECIO, ZONA, PESO, F\_SUMINISTRO, F\_NAC, #ZONA, TRANSPORTE).

Donde los campos tienen el siguiente significado:

- ARTICULO: Código asignado al artículo (único en la BD).
- PROV: Código asignado al proveedor (único en la BD).
- N\_UNIDADES: Número de unidades del artículo suministradas por el proveedor.
- DESCRIPCIÓN: Descripción del artículo.
- NOMBRE: Nombre del proveedor.
- F\_PEDIDO: Fecha en la que se hizo el pedido al proveedor.
- COLOR: Color del artículo.
- TAMAÑO: Tamaño del artículo.
- PAGO: Forma de pago. Ejemplo: cheque, contado, tarjeta,...
- #PAGO: Código de la forma de pago: 'c' cheque, 't' tarjeta,...
- PRECIO: Precio del artículo.
- ZONA: Zona en la que el proveedor suministra el artículo.
- PESO: Peso del artículo.
- F\_SUMINISTRO: Fecha en la que el proveedor suministra el artículo.
- F\_NAC Fecha de nacimiento del proveedor.
- #ZONA: Código de la zona en la que distribuye el proveedor.
- TRANSPORTE: Campo lógico que indica si el transporte está incluido en el pedido.

Se tienen en cuenta las siguientes restricciones:

- Se podrán realizar pedidos a un proveedor de un mismo artículo, pero siempre en días diferentes.
- Cada proveedor puede pagar de diferentes formas cada artículo.

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.

**1FN:** Una base de datos se encuentra en primera forma normal si, y sólo si, en todas sus tablas, cada atributo es atómico.

Partimos de la siguiente relación:

**SUMINISTROS** (ARTICULO, PROV, N\_UNIDADES, DESCRIPCIÓN, NOMBRE, F\_PEDIDO, COLOR, TAMAÑO, PAGO, #PAGO, PRECIO, ZONA, PESO, F\_SUMINISTRO, F\_NAC, #ZONA, TRANSPORTE).

Analizamos que No se cumple la 1FN, ya que para un artículo y un proveedor pueden haber diferentes números de unidades suministradas, con diferentes nombres, descripciones, colores, tamaños; de la misma forma que para un artículo que suministre un proveedor podrá haber diferentes precios y pagos, fechas y zonas de suministro.

*Nota: los atributos ARTICULO y PROV, ya vienen marcados como PKs por el enunciado*

Debemos transformar la relación llevando los atributos NO atómicos a una nueva tabla.

**PRODUCTOS** (ARTICULO, PROV, DESCRIPCIÓN, COLOR, PRECIO, TAMAÑO, PESO, NOMBRE, F\_NAC).

**SUMINISTRO** (F\_PEDIDO, F\_SUMINISTRO, N\_UNIDADES, #PAGO, PAGO, #ZONA, ZONA, TRANSPORTE)

*Nota: Una vez sacados los atributos no atómicos, descubrimos que éstos se han convertido en realidad, en la nueva relación de **SUMINISTROS** y que la antigua relación es ahora **PRODUCTOS***

Corrección:

Nota: el enunciado nos da una pista de que la F\_Pedido es una PK

**SUMINISTRO** (ARTICULO, PROV, F\_PEDIDO, F\_SUMINISTRO, N\_UNIDADES, DESCRIPCIÓN, COLOR, PRECIO, TAMAÑO, PESO, #PAGO, PAGO, #ZONA, ZONA, TRANSPORTE, NOMBRE, F\_NAC).

Ya está en 1FN.

Ahora analizaríamos la nueva relación **SUMINISTRO**

**SUMINISTRO** (F\_PEDIDO, F\_SUMINISTRO, N\_UNIDADES, #PAGO, PAGO, #ZONA, ZONA, TRANSPORTE)

*Nota: comprobamos que los atributos son atómicos, así que no hace falta modificar nada más.*

Y ahora ya tendríamos el resultado de la 1FN, con las nuevas relaciones establecidas:

**PRODUCTOS** (ARTICULO, PROV, DESCRIPCIÓN, COLOR, PRECIO, TAMAÑO, PESO, NOMBRE, F\_NAC).

**SUMINISTRO** (F\_PEDIDO, F\_SUMINISTRO, N\_UNIDADES, #PAGO, PAGO, #ZONA, ZONA, TRANSPORTE)

Tan sólo faltaría unir las con alguna PK

**PRODUCTOS** (ARTICULO, PROV, DESCRIPCIÓN, COLOR, PRECIO, TAMAÑO, PESO, NOMBRE, F\_NAC).

**SUMINISTRO** (ARTICULO, F\_PEDIDO, F\_SUMINISTRO, N\_UNIDADES, #PAGO, PAGO, #ZONA, ZONA, TRANSPORTE)

*Nota: hemos unido las relaciones poniendo la PK de ARTICULO en la relación de SUMINISTRO*

En realidad, lo que hemos obtenido es:

**SUMINISTRO** (ARTICULO, PROV, F\_PEDIDO, F\_SUMINISTRO, N\_UNIDADES, DESCRIPCIÓN, COLOR, PRECIO, TAMAÑO, PESO, #PAGO, PAGO, #ZONA, ZONA, TRANSPORTE, NOMBRE, F\_NAC).

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

---

A continuación, comprobamos si la BBDD está ya en 2FN (para ello todas sus tablas tienen que estarlo)

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

Partimos de:

**PRODUCTOS** (ARTICULO, PROV, DESCRIPCIÓN, COLOR, PRECIO, TAMAÑO, PESO, NOMBRE, F\_NAC).

**SUMINISTRO** (ARTICULO, F\_PEDIDO, F\_SUMINISTRO, N\_UNIDADES, #PAGO, PAGO, #ZONA, ZONA, TRANSPORTE)

Partimos de:

**SUMINISTRO** (ARTICULO, PROV, F\_PEDIDO, F\_SUMINISTRO, N\_UNIDADES, DESCRIPCIÓN, COLOR, PRECIO, TAMAÑO, PESO, #PAGO, PAGO, #ZONA, ZONA, TRANSPORTE, NOMBRE, F\_NAC).

Analizamos las siguientes dependencias:

ARTICULO, PROV → DESCRIPCIÓN, COLOR, PRECIO, TAMAÑO, PESO, NOMBRE, F\_NAC

F\_PEDIDO → F\_SUMINISTRO, N\_UNIDADES, #PAGO, PAGO, #ZONA, ZONA, TRANSPORTE

NO está en 2FN porque los atributos NO dependen de la clave completa.

- DESCRIPCIÓN, COLOR, PRECIO, TAMAÑO, PESO, NOMBRE, F\_NAC, no dependen todas del ARTICULO y del PROV a la vez, sino que NOMBRE y F\_NAC dependen sólo del PROV
- F\_SUMINISTRO, N\_UNIDADES, #PAGO, PAGO, #ZONA, ZONA, TRANSPORTE si dependen todas de la F\_PEDIDO por lo que no habría que hacer ninguna modificación

De modo que las dependencias serían:

ARTICULO → DESCRIPCIÓN, COLOR, PRECIO, TAMAÑO, PESO

PROV → NOMBRE, F\_NAC

ARTICULO, PROV, F\_PEDIDO → F\_SUMINISTRO, N\_UNIDADES, #PAGO, PAGO, #ZONA, ZONA, TRANSPORTE

Con lo que se llegaría a las siguientes relaciones:

**PRODUCTOS** (ARTICULO, DESCRIPCIÓN, COLOR, PRECIO, TAMAÑO, PESO)

**PROVEEDOR** (PROV, NOMBRE, F\_NAC)

**SUMINISTRO** (ARTICULO, PROV, F\_PEDIDO, F\_SUMINISTRO, N\_UNIDADES, #PAGO, PAGO, #ZONA, ZONA, TRANSPORTE)

Ahora necesitaríamos crear una nueva relación para unir las tres relaciones que ya tenemos:

**PRODUCTOS** (ARTICULO, DESCRIPCIÓN, COLOR, PRECIO, TAMAÑO, PESO)

**PROVEEDOR** (PROV, NOMBRE, F\_NAC)

**SUMINISTRO** (F\_PEDIDO, F\_SUMINISTRO, N\_UNIDADES, #PAGO, PAGO, #ZONA, ZONA, TRANSPORTE)

**CONSULTA** (ARTICULO, PROV, F\_PEDIDO)

---

**PRODUCTOS** (ARTICULO, DESCRIPCIÓN, COLOR, PRECIO, TAMAÑO, PESO)

**PROVEEDOR** (PROV, NOMBRE, F\_NAC)

**SUMINISTRO** (ARTICULO, PROV, F\_PEDIDO, F\_SUMINISTRO, N\_UNIDADES, #PAGO, PAGO, #ZONA, ZONA, TRANSPORTE)

*Nota: Ya estaba bn unido y relacionado ¡!*

*Nota: para ello hemos creado la relación CONSULTA con la cual rellenando esos tres campos que son las PKs de cada relación anterior, podríamos obtener todos los datos de las tablas.*

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

---

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

Partimos de:

**PRODUCTOS** (ARTICULO, DESCRIPCIÓN, COLOR, PRECIO, TAMAÑO, PESO)

**PROVEEDOR** (PROV, NOMBRE, F\_NAC)

**SUMINISTRO** (F\_PEDIDO, F\_SUMINISTRO, N\_UNIDADES, #PAGO, PAGO, #ZONA, ZONA, TRANSPORTE)

**CONSULTA** (ARTICULO, PROV, F\_PEDIDO)

---

**PRODUCTOS** (ARTICULO, DESCRIPCIÓN, COLOR, PRECIO, TAMAÑO, PESO)

**PROVEEDOR** (PROV, NOMBRE, F\_NAC)

**SUMINISTRO** (ARTICULO, PROV, F\_PEDIDO, F\_SUMINISTRO, N\_UNIDADES, #PAGO, PAGO, #ZONA, ZONA, TRANSPORTE)

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

$A(\text{ARTICULO}, \text{PROV}, \text{F\_PEDIDO}) \rightarrow B(\text{\#PAGO})$

$B(\text{\#PAGO}) \rightarrow C(\text{PAGO})$

$A(\text{ARTICULO}, \text{PROV}, \text{F\_PEDIDO}) \rightarrow C(\text{PAGO})$

Por otra parte, también tenemos la siguiente dependencia funcional con transitividad:

$A(\text{ARTICULO}, \text{PROV}, \text{F\_PEDIDO}) \rightarrow B(\text{\#ZONA})$

$B(\text{\#ZONA}) \rightarrow C(\text{ZONA})$

$A(\text{ARTICULO}, \text{PROV}, \text{F\_PEDIDO}) \rightarrow C(\text{ZONA})$

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

**FACTURA** (#PAGO, PAGO)

**DESCARGA** (#ZONA, ZONA)

Con lo que a modo resumen, habríamos llegado a:

**PRODUCTOS** (ARTICULO, DESCRIPCIÓN, COLOR, PRECIO, TAMAÑO, PESO)

**PROVEEDOR** (PROV, NOMBRE, F\_NAC)

**SUMINISTRO** (F\_PEDIDO, F\_SUMINISTRO, N\_UNIDADES, #PAGO, #ZONA, TRANSPORTE)

**FACTURA** (#PAGO, PAGO)

**DESCARGA** (#ZONA, ZONA)

**CONSULTA** (ARTICULO, PROV, F\_PEDIDO)

*Nota: Al haber creado estas dos nuevas relaciones, tuvimos que quitar los atributos PAGO y ZONA de la relación de SUMINISTRO*

---

**PRODUCTOS** (ARTICULO, DESCRIPCIÓN, COLOR, PRECIO, TAMAÑO, PESO)

**PROVEEDOR** (PROV, NOMBRE, F\_NAC)

**SUMINISTRO** (ARTICULO, PROV, F\_PEDIDO, F\_SUMINISTRO, N\_UNIDADES, #PAGO, PAGO, #ZONA, ZONA, TRANSPORTE)

**FACTURA** (#PAGO, PAGO)

**DESCARGA** (#ZONA, ZONA)

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