

OBJETIVO

- Proponer una solución normalizada a la tabla entregada.
- Llegar a la tercera forma normal, detallando todos los pasos.

SOLUCIÓN

La tabla original **Inventario** contiene los siguientes campos:

- codigo_producto
- producto
- local
- Precio
- existencia
- stock
- ubicacion
- numero_bodega
- vendedor
- rut_vendedor
- numero_boleta
- cantidad_vendida
- rut_cliente
- nombre_cliente

0) Notación: #clave_primaria | clave_foránea | Tabla con alias T

1) [1NF] → Atomicidad: un campo (intersección de una columna con un registro) no puede tener más de dos valores en la tabla. En este caso, notamos que la columna vendedor puede contener varios valores por registro, por lo tanto luego de aplicar **1NF** distinguimos dos tablas:

Inventario2

#codigo_producto	existencia	numero_boleta
producto	stock	cantidad_vendida
local	ubicacion	rut_cliente
precio	numero_bodega	nombre_cliente

Vendedor

#rut
nombre

2) [2NF] → [1NF] + eliminación de dependencia parcial: todas las columnas dependen únicamente de la PK de la tabla. Esto no se cumple en la tabla **Inventario2**, ya que muchas columnas no dependen de **#codigo_producto**. Luego de aplicar **2NF**, obtenemos las siguientes tablas:

Productos	Ventas	Bodegas	Clientes	Vendedores
#codigo	#numero_boleta	#numero	#rut	#rut
nombre	cantidad_vendida	existencia	nombre	nombre
precio	<i>V.nombre</i>	stock		
<i>B.stock</i>	<i>C.nombre</i>	<i>P.nombre</i>		
	<i>P.nombre</i>	local		
		provincia		

Es decir, una tabla para los productos con su respectivo código y nombre, el precio que tiene y el stock asociado; otra tabla para la información de las bodegas, con su número asociado, la existencia de stock, el stock actual de la bodega y el producto que contiene, junto con el local y provincia a la que pertenece; otra tabla para las ventas, referenciadas con su número de boleta, la cantidad de producto vendida, el nombre de los involucrados y el producto que se transa; una tabla para clientes y otra para vendedores, cada una con su respectivo RUT y nombre. Vemos que en todas estas tablas, todas sus columnas dependen únicamente de su respectiva PK.

3) [3NF] → [2NF] + eliminación de dependencia transitiva: la tabla sólo contiene columnas que dependen de forma no-transitiva de su PK. En este caso, por ejemplo, el precio de un producto depende de su nombre, y este nombre de su código. Podríamos entonces separar la tabla **Productos** en dos y llamar a *precio* como una FK. En otro ejemplo, en una bodega el stock depende de la existencia de stock, y esta depende de cada bodega. Esto significaría nuevamente fragmentar la tabla **Bodegas** en dos y llamar a *stock* como una FK. Sin embargo, creo que no todas estas nuevas tablas aportarían igual valor al modelamiento, y decidiré crear sólo una tabla más. De acuerdo a ello, el modelamiento final estaría dado por:

Productos	Ventas	Bodegas	Clientes	Vendedores	Locales
#codigo	#numero_boleta	#numero	#rut	#rut	#nombre
nombre	cantidad_vendida	existencia	nombre	nombre	provincia
precio	<i>V.nombre</i>	stock			
<i>B.stock</i>	<i>C.nombre</i>	<i>P.nombre</i>			
	<i>P.nombre</i>				
	<i>L.nombre</i>				

Y un diagrama ULM simplificado final sería:

