

Claves:

Para garantizar la integridad de los datos dentro de una tabla se selecciona/n el/los campos que identifiquen unívocamente cada fila.

Las tablas suelen representar algún tipo de entidad, por ejemplo, productos en la tabla **Productos**, y la clave principal puede ser tan simple como un campo individual que contenga los identificadores únicos para los objetos de este tipo de entidad. Podríamos considerar el nombre del producto como un identificador único, pero es habitual que en una tabla los nombres no sean únicos. En consecuencia, para identificar de forma unívoca cada producto colocamos un valor único en el campo **Codigo**. Esto evitará toda ambigüedad al hacer referencia a un producto específico.

Para crear una clave primaria (PRIMARY KEY) en el diagrama de base de datos o en la pantalla de diseño de la tabla, procedemos de la siguiente forma:

Seleccionamos el/los campo/s que serán partícipes de la clave principal de la tabla, es decir, los campos que identificarán a la entidad.

Luego pulsamos el icono de la barra de herramientas Establecer Clave Primaria, o botón derecho -> "Establecer Clave Principal".

Normalización de la Base de datos

La normalización es un término que deriva de la metodología que se utiliza para evitar redundancia de datos y el fácil acceso y actualización de estos. Dicha metodología fue enunciada por CODD. Esta consistía en definir un conjunto de normas a las que se las llamó formas normales. Cada forma fue numerada.

1º forma normal

La 1º FN solicita que se cumplan dos condiciones sobre la relación (entidad o tabla):

Debe existir una clave primaria

Todos los dominios simples contienen únicamente valores atómicos.

Sobre la existencia de la clave primaria a esta altura no hace falta realizar ninguna explicación. Y sobre la segunda condición que dice "todos los dominios simples contienen únicamente valores atómicos" quiere decir "los campos de la tabla no pueden contener grupos de datos repetitivos".

Apliquemos en el ejemplo:

NroFactura	CodProd	Descripcion	PUProd	Cantidad	CodCliente	Nombre	FechaVta
3	2	Martillo	20.4	15	201	Juan	10/03/06
3	5	Clavo	0.8	300	201	Juan	10/03/06
3	9	Taladro	120	2	201	Juan	10/03/06
3	15	Destornillador	7.5	8	201	Juan	10/03/06

Tabla 1: Para la tabla **Venta**, se muestra un conjunto de filas posibles.

Si se presta especial atención observará que los campos **NroFactura**, **CodCliente**, **Nombre**, **FechaVta** se repiten en la tabla por cada aparición diferente de **CodProd**,

Cantidad, PUProd y Descripción. En realidad estos últimos campos (referentes al producto) no están cumpliendo con la segunda condición definida para que una relación se encuentre en 1º FN y por eso, los campos correspondientes a la venta y al cliente se repiten en cada fila, esto es lo que llamamos redundancia de datos. Supongamos que el cliente en cuestión se cambiara el nombre, usted estaría obligado con este esquema de representación a actualizar las celdas en las filas de la tabla Ventas donde aparezca el cliente en cuestión.

¿Cómo hacemos para que nuestra relación Ventas quede en 1º FN?

- 1 - Eliminar de la relación (tabla) original el o los atributos que no posean únicamente valores atómicos (o sea los que provocan grupos repetitivos).
- 2 - Seleccionar la clave primaria de la relación original.
- 3 - Crear una nueva tabla que posea los campos que provocaban grupos repetitivos y fueron eliminados de la tabla original.
- 4 - El campo que es clave primaria en la tabla original será parte o formará parte de la clave primaria de la nueva tabla.
- 5 - Dar un nombre a la nueva tabla y definir su clave primaria.

En el ejemplo,

- 1 - Eliminamos de nuestra tabla **Ventas** los siguientes campos: **CodProd, Cantidad, PUProd, Descripción** (y agregamos el campo **Total**).
 - 2 - Seleccionamos como clave primaria de la tabla **Ventas** el campo **NroFactura**.
 - 3 - Definiremos una nueva tabla con los siguientes campos: **CodProd, Cantidad, PUProd, Descripción**.
 - 4 - Agregaremos a la nueva tabla el campo **NroFactura** que formará parte de la clave primaria de la nueva tabla.
 - 5 - Definiremos la tabla con el nombre **Detalle_Ventas** y tomaremos como clave primaria de ésta a los campos **NroFactura y CodProd**.
- Como consecuencia de esto nos queda un modelo relacional como el que se muestra a continuación:

Ventas	
PK	<u>NroFactura</u>
	CodCliente
	Nombre
	FechaVta
	Total

Detalle_Ventas	
PK	<u>NroFactura</u>
PK	<u>CodProd</u>
	Descripcion
	PUProd
	Cantidad

Respecto a los datos:

Ventas

Ventas

NroFactura	CodCliente	Nombre	FechaVta	Total
3	201	Juan	10/03/06	XXXX

Detalle_Ventas

NroFactura	CodProd	Descripcion	PUProd	Cantidad
3	2	Martillo	20.4	15
3	5	Clavo	0.8	300
3	9	Taladro	120	2
3	15	Destornillador	7.5	8

Ambas tablas han quedado en 1º FN con sus correspondientes claves primarias definidas y se muestra cómo se preservan los datos en ambas tablas sin pérdida de información con respecto a las filas (tuplas) de la tabla Ventas original.

2ª forma normal

Para que una tabla (relación) se encuentre en 2ª FN debe satisfacer las siguientes condiciones:

Debe estar en 1º FN.

Todos los atributos no clave dependen funcionalmente de la clave primaria.

En nuestro ejemplo las tablas (relaciones) **Ventas** y **Detalle_Ventas** que se encuentran en 1º FN no se encuentran en 2ª FN. Observemos en **Detalle_Ventas**, el campo **Descripcion** depende funcionalmente del campo **CodProd** que forma parte de la clave primaria de dicha tabla, pero no depende funcionalmente del campo **NroFactura** que también es uno de los campos que forman parte de la clave primaria de la tabla. En cambio los campos **Cantidad** y **PUProd** dependen funcionalmente de ambos campos, que forman la clave primaria. Veremos ahora como logramos que nuestra tabla **Detalle_Ventas** quede en 2ª FN.

- 1 – Eliminar de la tabla (relación) original el o los campos que no dependan funcionalmente por completo de la clave primaria.
 - 2 – Crear una nueva tabla que posea como campos el o los campos eliminados de la tabla original.
 - 3 – Agregar a dicha tabla el campo de la clave primaria de la tabla original del cual dependían funcionalmente los campos no claves de la tabla original.
 - 4 – El campo del punto anterior será tomado como clave primaria de la nueva tabla.
 - 5 – Darle un nombre a la nueva tabla.
- Ahora que sabemos cómo lograr que una tabla se encuentre en 2ª FN aplicaremos dicha técnica sobre la tabla **Detalle_Ventas**.

¿Cómo hacemos para que nuestra tabla **Detalle_Ventas** quede en 2º FN?

- 1 – Sacaremos de la tabla **Detalle_Ventas** el campo **Descripcion**.
- 2 – Crearemos una nueva tabla que contenga los campos **Descripcion** y **PUProd**.
- 3 – Agregamos a dicha tabla el campo **CodProd**.
- 4 – Definiremos el campo **CodProd** como clave primaria de dicha tabla.
- 5 – Llamaremos a dicha tabla **Productos**.

Con lo que nos queda nuestro modelo relacional de la siguiente forma:

Ventas	
PK	<u>NroFactura</u>
	CodCliente
	Nombre
	FechaVta
	Total

Detalle_Ventas	
PK	<u>NroFactura</u>
PK	<u>CodProd</u>
	PUProd
	Cantidad

Productos	
PK	<u>CodProd</u>
	Descripcion
	PUProd

Respecto a los datos:

Ventas

NroFactura	CodCliente	Nombre	FechaVta	Total
3	201	Juan	10/03/06	XXXX

Detalle_Ventas

NroFactura	CodProd	PUProd	Cantidad
3	2	20.4	15
3	5	0.8	300
3	9	120	2
3	15	7.5	8

Productos

CodProd	Descripcion	PUProd
2	Martillo	20.4
5	Clavo	0.8
9	Taladro	120
15	Destornillador	7.5

¿Nuestras tres tablas **Ventas**, **Detalle_Ventas** y **Productos** han quedado en 2ª FN?

Para esto deberían mostrar los valores de cada tabla en 2ª FN, con los valores derivados de las filas originales de la tabla **Ventas** del inicio, sin pérdida de información de estos, sin redundancia, gracias a la normalización.

Ventas	
PK	<u>NroFactura</u>
	CodCliente
	FechaVta
	Total

Detalle_Ventas	
PK	<u>NroFactura</u>
PK	<u>CodProd</u>
	PUProd
	Cantidad

Clientes	
PK	<u>CodCliente</u>
	Nombre

Productos	
PK	<u>CodProd</u>
	Descripcion
	PUProd