

1FN: Primera Forma Normal

La **normalización** en bases de datos es un proceso para estructurar los datos con el objetivo de minimizar la redundancia y evitar problemas de inconsistencia. La **Primera Forma Normal (1FN)** es el primer paso en este proceso y establece ciertos requisitos básicos para que una tabla esté adecuadamente estructurada.

Para que una base de datos esté en **Primera Forma Normal (1FN)**, debe cumplir con las siguientes reglas:

- 1. **Atomicidad:** Cada columna debe contener valores atómicos, es decir, cada celda de la tabla debe tener solo un valor, sin listas o conjuntos de valores. Por ejemplo, en lugar de tener una columna de "teléfonos" donde aparezcan varios números separados por comas, cada número de teléfono debería estar en una fila o columna independiente.
- 2. **Valores de columnas del mismo tipo:** Todos los valores de una columna deben ser del mismo tipo de dato. Por ejemplo, si una columna es de tipo "Fecha", todos los valores en esa columna deben ser fechas.
- 3. **Identificación única:** La tabla debe tener una clave primaria o un identificador único que diferencie cada fila, asegurando que no haya duplicados.

Ejemplo de una tabla que no está en 1FN y cómo sería una versión normalizada:

Cliente	Teléfonos
Juan	1234, 5678
Ana	7890

Para normalizar esta tabla en 1FN:

Cliente	Teléfono
Juan	1234
Juan	5678
Ana	7890

En esta versión normalizada, cada celda contiene un solo valor (atomización) y se eliminan listas de valores en una misma celda. Esta estructura hace que la base de datos sea más consistente y más fácil de gestionar.

2FN: Segunda Forma Normal

La **Segunda Forma Normal (2FN)** es el siguiente paso en el proceso de normalización después de la Primera Forma Normal. Para que una tabla esté en 2FN, debe cumplir con los requisitos de la 1FN y agregar algunos criterios adicionales relacionados con las dependencias de las columnas en la clave primaria.

Una tabla está en **Segunda Forma Normal (2FN)** si cumple con estos requisitos:

1. **Estar en Primera Forma Normal (1FN):** La tabla ya debe estar normalizada en 1FN, es decir, no debe tener valores repetidos en una misma celda, ni listas, y debe tener un identificador único (clave primaria).
2. **No tener dependencias parciales de la clave primaria:** Esto significa que todas las columnas en la tabla deben depender de toda la clave primaria, no de una parte de ella. Esto aplica solo a tablas con claves primarias compuestas (es decir, claves formadas por más de una columna).
 - En una tabla con una clave primaria compuesta, cada columna que no sea clave debe depender de todas las columnas que forman la clave primaria.
 - Si alguna columna depende solo de una parte de la clave compuesta y no de toda ella, se dice que hay una **dependencia parcial**, lo cual viola la 2FN.

Ejemplo

Supongamos una tabla de "Pedidos" que está en 1FN pero no en 2FN:

ID Pedido	ID Producto	Nombre Producto	Cantidad
1	A123	Camisa	10
1	B456	Pantalón	5
2	A123	Camisa	3

- En este caso, la clave primaria es la combinación de **ID Pedido** e **ID Producto**.
- La columna **Nombre Producto** depende solo de **ID Producto** y no de toda la clave primaria (**ID Pedido**, **ID Producto**), lo cual es una **dependencia parcial**.

Para llevar esta tabla a 2FN, debemos crear una nueva tabla para los productos, eliminando las dependencias parciales:

Tabla Pedidos:

ID Pedido	ID Producto	Cantidad
1	A123	10
1	B456	5
2	A123	3

Tabla Productos:

ID Producto	Nombre Producto
A123	Camisa
B456	Pantalón

Ahora, la tabla de pedidos en 2FN solo tiene columnas que dependen de toda la clave primaria (**ID Pedido** e **ID Producto**). La información del nombre del producto se ha separado en otra tabla, eliminando la dependencia parcial y optimizando la estructura de la base de datos.

3FN: Tercera Forma Normal

La **Tercera Forma Normal (3FN)** es el siguiente paso en el proceso de normalización de bases de datos, después de haber cumplido con los requisitos de la Segunda Forma Normal (2FN). La 3FN agrega una regla adicional que evita redundancias relacionadas con las **dependencias transitivas**.

Una tabla está en **Tercera Forma Normal (3FN)** si cumple con los siguientes requisitos:

1. **Estar en Segunda Forma Normal (2FN)**: La tabla debe estar normalizada en 2FN, lo que significa que todas las columnas dependen de la clave primaria completa y no tienen dependencias parciales.
2. **No tener dependencias transitivas**: Una dependencia transitiva ocurre cuando una columna que no es clave depende de otra columna que tampoco es clave, y esta última depende de la clave primaria. En otras palabras, ninguna columna que no sea clave debe depender indirectamente de la clave primaria a través de otra columna que tampoco sea clave.

Ejemplo

Supongamos que tenemos una tabla "Empleados" que ya cumple con la 2FN pero no con la 3FN:

ID Empleado	Nombre Empleado	Departamento	Gerente Departamento
1	Juan Pérez	Ventas	Carlos Martínez
2	Ana López	Compras	María García
3	Pedro Ruiz	Ventas	Carlos Martínez

En esta tabla:

- La clave primaria es **ID Empleado**.
- La columna **Gerente Departamento** depende del **Departamento** y no directamente de **ID Empleado**.
- Esto significa que hay una **dependencia transitiva**: **Gerente Departamento** depende de **Departamento**, y **Departamento** depende de **ID Empleado**.

Para llevar esta tabla a 3FN, creamos una nueva tabla para los departamentos, eliminando la dependencia transitiva.

Tabla Empleados:

ID Empleado	Nombre Empleado	Departamento
1	Juan Pérez	Ventas
2	Ana López	Compras
3	Pedro Ruiz	Ventas

Tabla Departamentos:

Departamento	Gerente Departamento
Ventas	Carlos Martínez
Compras	María García

Ahora, en la tabla de "Empleados", todas las columnas dependen directamente de la clave primaria **ID Empleado**, sin depender de otras columnas. En la tabla "Departamentos", cada departamento tiene su gerente correspondiente, eliminando redundancias y manteniendo la base de datos estructurada en 3FN.

Tips para diseño 3FN.

Una relación m a n -> Crear una tabla intermedia: ejemplo

Chinook: InvoiceItems.