

### Normalización de Bases de Datos

La normalización es un proceso en el diseño de bases de datos relacionales que tiene como objetivo organizar los datos para minimizar la redundancia y evitar problemas de inconsistencia. Este proceso se lleva a cabo mediante la aplicación de una serie de reglas o "formas normales".

#### **Primera Forma Normal (1FN)**

**Regla:** Una tabla está en 1FN si todos los valores de sus columnas son atómicos (indivisibles), y cada columna contiene un único valor por registro.

**Objetivo:** Eliminar grupos repetitivos y asegurar que cada campo almacene un solo valor.

**Ejemplo:** Supongamos que tenemos la siguiente tabla de Estudiantes:

<u>id_estudiante</u>	nombre	cursos
1	Juan	Matemáticas, Historia
2	María	Física
3	Pedro	Matemáticas, Química

**Problema:** La columna "cursos" contiene múltiples valores, lo que viola la 1FN.

**Solución:** Dividir los valores de "cursos" en filas separadas para que cada celda contenga un solo valor.

<u>id_estudiante</u>	nombre	<u>curso</u>
1	Juan	Matemáticas
1	Juan	Historia
2	María	Física
3	Pedro	Matemáticas
3	Pedro	Química

Ahora, la tabla cumple con la 1FN porque cada celda tiene un solo valor.

## Segunda Forma Normal (2FN)

**Regla:** Una tabla está en 2FN si está en 1FN y todos los atributos no clave dependen completamente de la clave primaria, es decir, no debe haber dependencias parciales.

**Objetivo:** Eliminar la dependencia parcial de los atributos en la clave primaria.

**Ejemplo:** Partimos de la tabla que ya está en 1FN:

<u>id_estudiante</u>	<u>curso</u>	nombre	profesor
1	Matemáticas	Juan	Pérez
1	Historia	Juan	García
2	Física	María	López
3	Matemáticas	Pedro	Pérez
3	Química	Pedro	Jiménez

**Problema:** Aquí, el atributo "nombre" depende solo de "id\_estudiante" (dependencia parcial), y "profesor" depende solo de "curso". Esto significa que tenemos dependencias parciales, lo que viola la 2FN.

**Solución:** Dividir la tabla en dos para eliminar las dependencias parciales.

### Tabla Estudiantes:

<u>id_estudiante</u>	nombre
1	Juan
2	María
3	Pedro

### Tabla Cursos:

<u>curso</u>	<u>id_estudiante</u>	profesor
Matemáticas	1	Pérez
Historia	1	García
Física	2	López
Matemáticas	3	Pérez
Química	3	Jiménez

Ahora, ambas tablas están en 2FN porque cada atributo no clave depende completamente de la clave primaria de su respectiva tabla.

### Tercera Forma Normal (3FN)

**Regla:** Una tabla está en 3FN si está en 2FN y no tiene dependencias transitivas, es decir, no debe haber ningún atributo no clave que dependa de otro atributo no clave.

**Objetivo:** Eliminar las dependencias transitivas para asegurar que los atributos no clave dependan solo de la clave primaria.

**Ejemplo:** Consideremos la tabla Cursos en 2FN:

curso	id_estudiante	profesor	departamento
Matemáticas	1	Pérez	Ciencias
Historia	1	García	Humanidades
Física	2	López	Ciencias
Matemáticas	3	Pérez	Ciencias
Química	3	Jiménez	Ciencias

**Problema:** El atributo "departamento" depende de "profesor" (y no directamente de "curso" o "id\_estudiante"), lo que genera una dependencia transitiva.

**Solución:** Dividir la tabla en dos para eliminar la dependencia transitiva.

#### Tabla Cursos (Pivot):

curso	id_estudiante
Matemáticas	1
Historia	1
Física	2
Matemáticas	3
Química	3

#### Tabla Profesores:

profesor	departamento
Pérez	Ciencias
García	Humanidades
López	Ciencias
Jiménez	Ciencias

De esta manera, hemos eliminado la dependencia transitiva y ahora ambas tablas están en 3FN. Ahora ambas tablas están en 3FN.

#### Resumiendo:

- **1FN:** Elimina grupos repetitivos y asegura que cada campo contenga un solo valor.
- **2FN:** Elimina dependencias parciales, asegurando que todos los atributos no clave dependan completamente de la clave primaria.
- **3FN:** Elimina dependencias transitivas, asegurando que no haya atributos no clave que dependan de otros atributos no clave.