Chuleta de Normalización Relacional (1NF, 2NF, 3NF)

El objetivo de normalizar es eliminar la redundancia y evitar problemas al insertar, modificar o borrar datos (anomalías).

Ejemplo Práctico: Normalización de la tabla "CARRERAS"

La Tabla Original (Sin Normalizar)

Partimos de una única tabla que mezcla información sobre carreras, asignaturas y la duración específica de cada una. Esta estructura provoca redundancia: el nombre de la carrera "Historia del arte" se repite, al igual que el nombre de la asignatura "Inglés".

codCarrera	codAsignatura	nombreCarrera	nombreAsignatura	numHorasAsignatura
00AB	011	Historia del arte	Inglés	300
00AB	024	Historia del arte	Lengua Castellana	450
00CC	011	Filología Hispánica	Inglés	350
00CC	099	Filología Hispánica	Literatura española	400

```
CARRERAS_ORIGINAL (codCarrera, codAsignatura, nombreCarrera, nombreAsignatura, numHorasAsignatura)
```

Fase 1: Primera Forma Normal (1FN)

El Objetivo

Identificar la **Clave Primaria (PK)** que identifique unívocamente cada fila y asegurar que todos los valores sean **atómicos** (un solo dato por celda).

Análisis

La tabla ya cumple con la atomicidad. Para la PK, ni `codCarrera` ni `codAsignatura` son suficientes por sí solos. La combinación de ambos, `(codCarrera, codAsignatura)`, sí que identifica de forma única cada fila (la relación entre una carrera y una asignatura).

Resultado (Tabla en 1FN)

```
CARRERAS_1FN (codCarrera, codAsignatura, nombreCarrera, nombreAsignatura, numHorasAsignatura)
```

Fase 2: Segunda Forma Normal (2FN)

El Objetivo

Eliminar las **dependencias parciales**. Cada atributo que no es clave debe depender de la **clave primaria completa**, no solo de una parte de ella.

Análisis

Buscamos atributos que no necesiten la PK completa `(codCarrera, codAsignatura)` para ser determinados:

- `nombreCarrera`: Sabiendo que `codCarrera` es "00AB", ya sabemos que el nombre es "Historia del arte". No necesitamos saber la asignatura.
  → **Dependencia Parcial: {codCarrera} → nombreCarrera**
- `nombreAsignatura`: Sabiendo que `codAsignatura` es "011", ya sabemos que el nombre es "Inglés". No necesitamos saber la carrera.
  → **Dependencia Parcial: {codAsignatura} → nombreAsignatura**
- `numHorasAsignatura`: Para saber que la duración es de 300 horas, necesitamos saber ambas cosas: que es la carrera "00AB" y la asignatura "011". Depende de la clave completa. ¡Este atributo está bien!

Resultado (División de tablas)

Para "curar" las dependencias parciales, extraemos los atributos problemáticos a sus propias tablas, donde la parte de la clave de la que dependen se convierte en su nueva PK.

```
CARRERAS (codCarrera, nombreCarrera)

ASIGNATURAS (codAsignatura, nombreAsignatura)

PENSUM (codCarrera*, codAsignatura*, numHorasAsignatura)
```

Fase 3: Tercera Forma Normal (3FN)

El Objetivo

Eliminar las **dependencias transitivas**. Esto significa que un atributo que no es clave no puede depender de otro atributo que tampoco es clave.

Análisis

Revisamos las tres nuevas tablas que hemos creado en la Fase 2:

- **Tabla CARRERAS:** El único atributo no clave (`nombreCarrera`) depende directamente de la PK (`codCarrera`). No hay dependencias transitivas. ✓ **Cumple 3FN.**
- **Tabla ASIGNATURAS:** El único atributo no clave (`nombreAsignatura`) depende directamente de la PK (`codAsignatura`). No hay dependencias transitivas. ✓ **Cumple 3FN.**
- **Tabla PENSUM:** El único atributo no clave (`numHorasAsignatura`) depende directamente de la PK (`codCarrera, codAsignatura`). No hay dependencias transitivas. ✓ **Cumple 3FN.**

Resultado Final (Tablas en 3FN)

La descomposición realizada para cumplir la 2FN ha sido suficiente para alcanzar también la 3FN. El modelo final, limpio y sin redundancia, queda así:

```
CARRERAS (codCarrera, nombreCarrera)
// Guarda la información única de cada carrera.

ASIGNATURAS (codAsignatura, nombreAsignatura)
// Guarda la información única de cada asignatura.

PENSUM (codCarrera, codAsignatura, numHorasAsignatura)
// Tabla intermedia que relaciona las otras dos y guarda el dato que depende de ambas: las horas.
```