

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 "CICLISTAS"

La Tabla Original (Sin Normalizar)

Partimos de una única tabla que mezcla información sobre torneos, etapas y los tiempos de los ciclistas. Esta estructura causa mucha redundancia de datos (los nombres de los ciclistas y los kilómetros de las etapas se repiten).

torneo	numeEtapa	codCorredor	nombreCiclista	tiempoMin	recorridoKm
Vuelta a España	1	125	Indurain	125	250
Vuelta a España	1	135	Perico	200	250
Tour de Francia	1	135	Perico	150	200
Tour de Francia	1	188	Contador	100	200

CICLISTAS_ORIGINAL (torneo, numeEtapa, codCorredor, nombreCiclista, tiempoMin, recorridoKm)

Fase 1: Primera Forma Normal (1FN)

El Objetivo

Identificar la **Clave Primaria (PK)** y asegurar que todos los valores de la tabla sean **atómicos** (una sola pieza de información por celda, sin listas ni grupos).

Análisis

La tabla ya está en 1FN porque cada celda contiene un valor único. La clave primaria que identifica unívocamente cada fila (cada participación de un ciclista en una etapa) es la combinación de los tres primeros atributos.

Resultado (Tabla en 1FN)

CICLISTAS_1FN (torneo, numeEtapa, codCorredor, nombreCiclista, tiempoMin, recorridoKm)

Fase 2: Segunda Forma Normal (2FN)

El Objetivo

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

Análisis

Buscamos atributos que no necesiten la PK completa para ser determinados:

- nombreCiclista: ¿Depende de `(torneo, numeEtapa, codCorredor)`? No. Sabiendo que `codCorredor` es 135, sabemos que el nombre es "Perico", sin importar el torneo o la etapa.
→ **Dependencia Parcial: {codCorredor} → nombreCiclista**
- recorridoKm: ¿Depende de la clave completa? No. Sabiendo que estamos en la etapa 1 de la "Vuelta a España", sabemos que el recorrido es de 250 Km, sin importar qué ciclista la corra.
→ **Dependencia Parcial: {torneo, numeEtapa} → recorridoKm**
- tiempoMin: ¿Depende de la clave completa? Sí. El tiempo de 125 minutos solo tiene sentido para Indurain (125) en la etapa 1 de la Vuelta a España. Depende de los tres. ¡Este atributo está bien!

Resultado (División de tablas)

Para "curar" las dependencias parciales, extraemos los atributos problemáticos a nuevas tablas:

CICLISTAS (codCorredor, nombreCiclista)

ETAPAS (torneo, numeEtapa, recorridoKm)

PARTICIPACIONES (torneo*, numeEtapa*, codCorredor*, tiempoMin)

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 CICLISTAS:** El único atributo no clave (`nombreCiclista`) depende directamente de la PK (`codCorredor`). No hay dependencias transitivas. ✓ **Cumple 3FN.**
- Tabla ETAPAS:** El único atributo no clave (`recorridoKm`) depende directamente de la PK (`torneo, numeEtapa`). No hay dependencias transitivas. ✓ **Cumple 3FN.**
- Tabla PARTICIPACIONES:** El único atributo no clave (`tiempoMin`) depende directamente de la PK (`torneo, numeEtapa, codCorredor`). No hay dependencias transitivas. ✓ **Cumple 3FN.**

Resultado Final (Tablas en 3FN)

En este caso, la descomposición que hicimos para cumplir la 2FN fue suficiente para alcanzar también la 3FN. El modelo final, limpio y sin redundancia, queda así:

CICLISTAS (codCorredor, nombreCiclista)
// Guarda la información única de cada ciclista.

ETAPAS (torneo, numeEtapa, recorridoKm)
// Guarda la información única de cada etapa de cada torneo.

PARTICIPACIONES (torneo, numeEtapa, codCorredor, tiempoMin)
// Tabla intermedia que relaciona las otras dos y guarda el dato único de la participación: el tiempo.