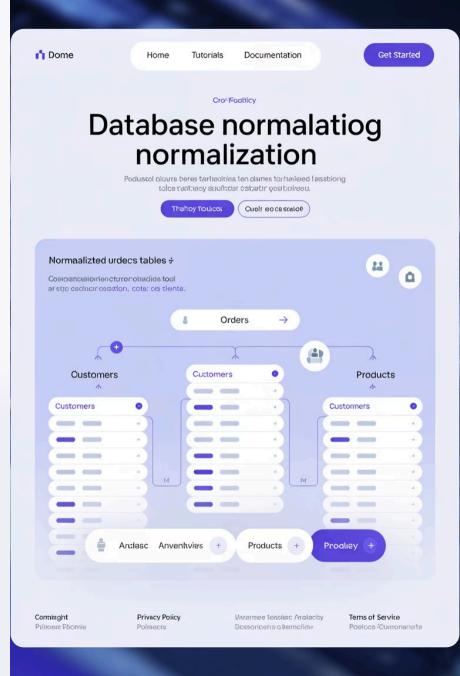
Normalización de Bases de Datos

La normalización es un proceso fundamental en el diseño de bases de datos. Divide grandes tablas en más pequeñas y establece relaciones entre ellas.

UTN - Tecnicatura Universitaria en Programación





¿Qué es la Normalización?

La normalización es un enfoque sistemático para descomponer tablas con el fin de eliminar la redundancia de datos y las características indeseables como las anomalías de inserción, actualización y eliminación.

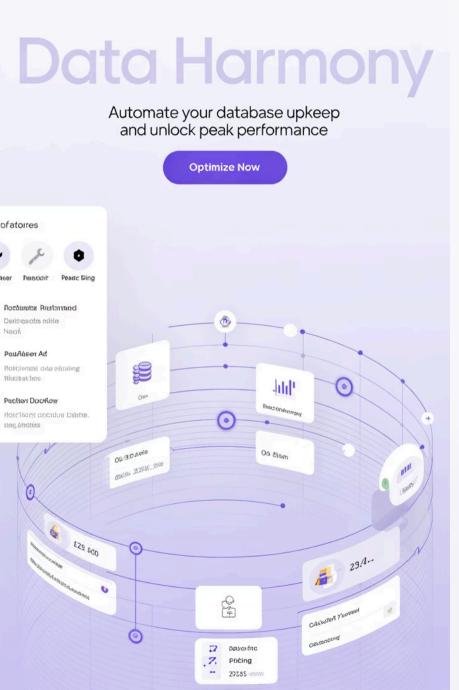
ony

Suppo

ricing

ricing





Ventajas de la Normalización



Reducción de redundancia

Elimina la repetición innecesaria de datos, haciendo la base de datos más eficiente en almacenamiento.



Mejora de integridad

Reduce errores e inconsistencias al organizar los datos de manera estructurada.



Fácil mantenimiento

Simplifica operaciones como actualización o eliminación sin afectar otras partes de la base.

Desventajas de la Normalización



Mayor complejidad

Las consultas se vuelven más complejas al requerir operaciones JOIN para unir tablas.



Impacto en rendimiento

El aumento de tablas y relaciones puede afectar el rendimiento en bases grandes.



Más tablas

Puede resultar en muchas tablas pequeñas, complicando la administración de la base.



Primera Forma Normal (1FN)

Reglas Fundamentales

- Cada registro debe ser único
- Cada celda debe contener un solo valor

La primera forma normal elimina grupos repetitivos y garantiza que cada intersección de fila-columna contenga exactamente un valor.



Ejemplo de Violación de 1FN

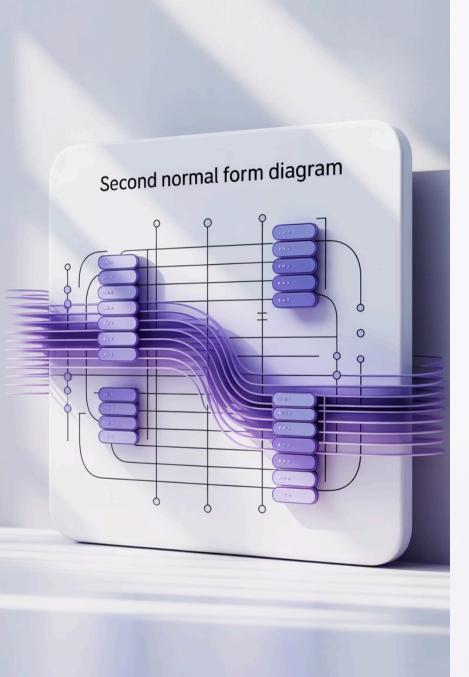
Tabla No Normalizada

id_estudiante	nombre	curso
1	Diego	R, Python
2	Claudia	Data analysis

Esta tabla viola la 1FN porque contiene múltiples valores en una celda (curso).

Tabla en 1FN

id_estudiante	nombre	curso
1	Diego	R
1	Diego	Python
2	Claudia	Data analysis



Segunda Forma Normal (2FN)

1 Requisitos

- Cumplir con la primera forma normal
- Evitar dependencias parciales con las claves primarias

2 Casos Especiales

- Si la PK tiene una sola columna, automáticamente cumple 2FN
- Si la PK es compuesta, cada columna debe depender de TODAS las columnas de la PK

Ejemplo de Violación de 2FN

Tabla que Viola 2FN

En esta tabla, id_profesor y profesor dependen solo de id_curso, no de id_estudiante.

id_estu diante (PK)	id_curs o (PK)	id_profe sor	profeso r	progres o
235	2001	560	Jorge	0.55
455	2345	658	Angela	0.10
767	6548	999	Fiona	1.00



Solución para 2FN

1

Tabla Original

Contiene dependencias parciales que violan la 2FN.

2

Dividir en Tablas

Separar en dos tablas más pequeñas para eliminar dependencias parciales.

3

Resultado 2FN

Cada tabla ahora cumple con la segunda forma normal.



Tablas en Segunda Forma Normal

Tabla de Progreso

id_estudiante (PK)	id_curso (PK)	progreso
235	2001	0.55
455	2345	0.10
767	6548	1.00

Tabla de Cursos

id_curso (PK)	id_profesor	profesor
2001	560	Jorge
2345	658	Angela
6548	999	Fiona

Tercera Forma Normal (3FN)

1 Requisitos

- Cumplir con la segunda forma normal
- Eliminar dependencias transitivas

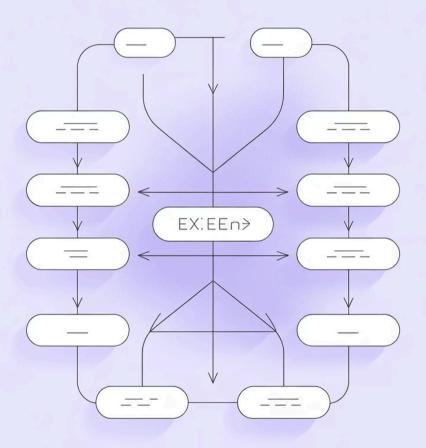
2 Dependencias Transitivas

Los campos que no son parte de la PK no pueden depender de otros campos que no sean PK.

3 Objetivo

Cada atributo debe depender directamente de la clave primaria, no a través de otro atributo.

Third Normal Form Diagram



Ejemplo de Violación de 3FN

Tabla que Viola 3FN

id_curso (PK)	id_profesor	profesor	tecnologia
2001	560	Jorge	Python
2345	658	Angela	R
6548	999	Fiona	SQL

Esta tabla viola la 3FN porque:

- tecnologia depende únicamente de id_curso
- id_profesor y profesor dependen entre sí

Hay dependencias transitivas que deben eliminarse.



Solución para 3FN

1

Tabla Original

Contiene dependencias transitivas que violan la 3FN.

2

Dividir en Tablas

Separar en dos tablas más pequeñas para eliminar dependencias transitivas.

3

Resultado 3FN

Cada tabla ahora cumple con la tercera forma normal.

Tablas en Tercera Forma Normal

Tabla de Cursos

id_curso (PK)	id_profesor	tecnologia
2001	560	Python
2345	658	R
6548	999	SQL

Tabla de Profesores

id_profesor (PK)	nombre
560	Jorge
658	Angela
999	Fiona

Problema de Redundancia

Tabla Aplanada (No Normalizada)

Una tabla aplanada contiene todos los datos en una sola estructura, generando redundancia.

Esto causa problemas de almacenamiento y mantenimiento de datos.



Ejemplo de Redundancia

id_estudiant e	nombre_est udiante	apellido_est udiante	id_curso	tecnologia_ curso	id_profesor	nombre_pr ofesor	apellido_pr ofesor
235	Micaela	Beltrán	2001	Python	560	Jorge	Paez
455	Jonathan	Rodriguez	2345	R	658	Angela	Felici
767	Andrés	Pacheco	6548	SQL	999	Fiona	Valdez
767	Andrés	Pacheco	2001	Python	560	Jorge	Paez

Observe la redundancia: los datos de Andrés Pacheco, el curso Python y el profesor Jorge Paez aparecen duplicados.

Solución Normalizada

Tabla Estudiantes

Contiene solo información de estudiantes y su relación con cursos.

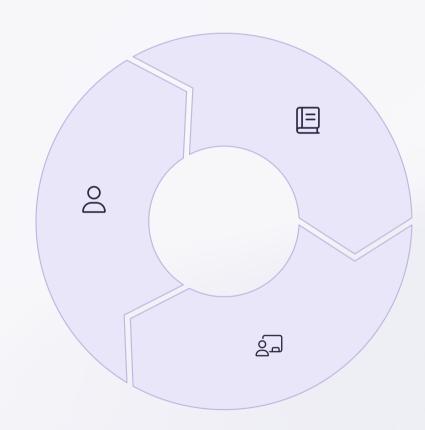


Tabla Cursos

Almacena información de cursos y su relación con profesores.

Tabla Profesores

Guarda únicamente datos de profesores, sin redundancia.

Estructura de Tablas Normalizadas

Tabla Estudiantes

id_estudia nte (PK)	nombre	apellido	id_curso
235	Micaela	Beltrán	2001
455	Jonathan	Rodriguez	2345
767	Andrés	Pacheco	6548

Tabla Cursos

id_curso (PK)	id_profesor	tecnologia
2001	560	Python
2345	658	R
6548	999	SQL

Tabla de Profesores Normalizada

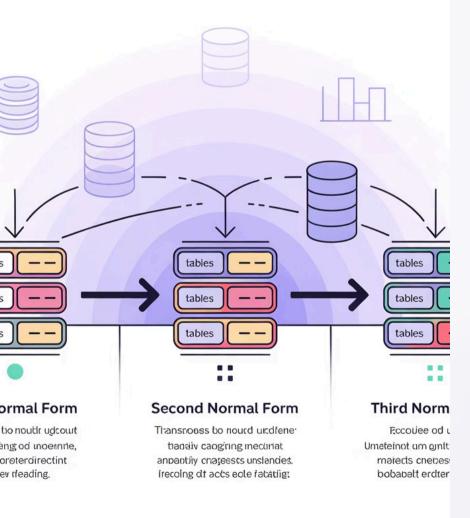
Tabla Profesores

id_profesor (PK)	nombre	apellido
560	Jorge	Paez
658	Angela	Felici
999	Fiona	Valdez

Beneficios

- Eliminación de redundancia
- Mejor integridad de datos
- Facilidad de mantenimiento
- Estructura clara y comprensible

atabase Normalizatio



Resumen de Normalización



La normalización mejora la estructura de la base de datos, reduce la redundancia y facilita el mantenimiento.