

Clase 4

Alexis Ovalle

9 de abril de 2025

Indice

- 1 Normalizacion
- 2 Ejemplo
- 3 Web scrapping

¿Qué es la Normalización?

- Proceso de organización de los datos en una base de datos para reducir la redundancia y mejorar la integridad de los datos.
- Implica dividir tablas grandes en tablas más pequeñas y definir relaciones entre ellas.
- Ayuda a minimizar problemas de actualización y eliminación, facilitando el mantenimiento de la base de datos.

¿Por qué Normalizar una Base de Datos?

- Evita la duplicidad de datos y garantiza mínima redundancia.
- Optimiza el espacio físico de almacenamiento.
- Facilita el acceso e interpretación de los datos.
- Protege la integridad de los datos previniendo eliminaciones indeseadas.

Primera Forma Normal (1NF)

DESARMAR ARREGLOS = atomicidad de datos

- Una tabla está en 1NF si:
 - Todos los atributos contienen valores atómicos (indivisibles).
 - No existen grupos repetitivos ni listas dentro de las celdas.

Ejemplo 1 de Primera Forma Normal (1NF)

Tabla NO normalizada (violando 1NF):

ID	Nombre	Teléfonos
1	Juan	12345678, 87654321
2	Ana	11223344

Tabla normalizada (cumple 1NF):

ID	Nombre	Teléfono
1	Juan	12345678
1	Juan	87654321
2	Ana	11223344

Ejemplo 2: Valores de distinto tipo en la columna

Violación a la 1NF:

ID Empleado	Nombre
1	Juan
2	25
3	Ana

Ejemplo 3: Productos y cantidades en una sola columna (No cumple 1NF)

Tabla de Pedidos

ID Pedido	Cliente	Productos
101	Carlos López	TV-2, Laptop-1
102	María Ruiz	Smartphone-1

Ejemplo 4: Múltiples columnas para un mismo tipo de dato

Violación a la 1NF:

ID Cliente	Nombre	FPago1	FPago2	FPago3
1	Ileana	Tarjeta	Efectivo	—
2	Luis	Efectivo	—	—

Se usan varias columnas para un mismo atributo (forma de pago).

Solución:

Tabla Clientes:

ID Cliente	Nombre
1	Ileana
2	Luis

Tabla FormasDePago:

ID Cliente	FPago
1	Tarjeta
1	Efectivo
2	Efectivo

Segunda Forma Normal (2NF)

Si no depende = saquese

- Una tabla está en 2NF si:
 - Está en 1NF.
 - Todos los atributos no clave dependen completamente de la clave primaria, evitando dependencias parciales.
- **Nota:** Aplica principalmente a tablas con claves primarias compuestas.

La forma normal 2, consiste en tener una llave primaria ÚNICA dentro de nuestra tabla, y todo dato que no dependa de ella tendrá irse a otra tabla.

No es necesario que sea una llave primaria dentro de una de las tablas para que sea foránea, pero si es necesario en ocasiones hacer referencia al mismo dato, y para evitar duplicidad de muchos más datos se emigra este dato a otra tabla.

Ejemplo de violación a la 2FN

Tabla: Ventas (violación a la 2FN)

<u>IDVenta</u>	<u>IDProducto</u>	NombreProducto	Cantidad	Fecha
1	101	Lápiz	5	2023-01-01
1	102	Cuaderno	3	2023-01-01
2	101	Lápiz	2	2023-01-02

Violación: "NombreProducto" depende solo de IDProducto, no de la clave compuesta.

Normalización a 2FN: Separación de dependencias parciales

Tabla: Ventas

IDVenta	IDProducto	Cantidad	Fecha
1	101	5	2023-01-01
1	102	3	2023-01-01
2	101	2	2023-01-02

Tabla: Productos

IDProducto	NombreProducto
101	Lápiz
102	Cuaderno

Violación a la 2FN en tabla de calificaciones

Tabla: Calificaciones (violación a la 2FN)

IDAlumno	IDCurso	NombreAlumno	Nota
1	A1	Ana Pérez	90
1	B2	Ana Pérez	88
2	A1	Luis Gómez	95

Violación: "NombreAlumno" depende solo de IDAlumno, no de la clave compuesta.

Solución a 2FN: Separación de alumno y nota

Tabla: Calificaciones

IDAlumno	IDCurso	Nota
1	A1	90
1	B2	88
2	A1	95

Tabla: Alumnos

IDAlumno	NombreAlumno
1	Ana Pérez
2	Luis Gómez

Violación a la Segunda Forma Normal (2FN)

Tabla: Empleados

IDEmpleado	Nombre	Sexo
1	Luis Martínez	Masculino
2	Ana Gómez	Femenino
3	Carlos Ruiz	Masculino
4	Laura Díaz	Femenino

Violación: El campo "Sexo" no depende de toda la clave (IDEmpleado), y su valor se repite. Es redundante.

Solución en 2FN: Tabla separada para Sexo

Tabla: Empleados

IDEmpleado	Nombre	IDSexo
1	Luis Martínez	M
2	Ana Gómez	F
3	Carlos Ruiz	M
4	Laura Díaz	F

Tabla: Sexo

IDSexo	Descripción
M	Masculino
F	Femenino

Tercera Forma Normal (3NF)

Dependencia directa o indirecta?

- Una tabla está en 3NF si:
 - Está en 2NF.
 - No existen dependencias transitivas entre atributos no clave.

Dependencias Funcionales

- Una **dependencia funcional** se da cuando el valor de un atributo (o conjunto de atributos) determina de forma única el valor de otro atributo.
- **Notación:** Si A determina a B , se escribe: $A \rightarrow B$
- **Ejemplo:**
Si cada empleado tiene un ID único y ese ID determina su nombre:
 $ID_{Empleado} \rightarrow Nombre$

Dependencias Transitivas

- Una **dependencia transitiva** ocurre cuando un atributo A determina a B y B determina a C , entonces A determina a C de forma transitiva.
- **Notación:** Si $A \rightarrow B$ y $B \rightarrow C$, entonces $A \rightarrow C$
- **Estas dependencias violan la 3FN**, ya que C no depende directamente de la clave primaria, sino a través de otro atributo no clave.

Ejemplo de Dependencia Transitiva

Tabla: Empleados

IDEmpleado	Nombre	Departamento	<u>Ubicación</u>
1	Juan Pérez	Ventas	CDMX
2	Ana Díaz	TI	Monterrey

Dependencias:

- $\text{IDEmpleado} \rightarrow \text{Departamento}$
- $\text{Departamento} \rightarrow \text{Ubicación}$
- $\Rightarrow \text{IDEmpleado} \rightarrow \text{Ubicación}$ (transitiva)

Esto rompe la 3FN porque 'Ubicación' no depende directamente de la clave primaria.

Violación a la Tercera Forma Normal (3FN)

Tabla: Empleados

IDEmpleado	Nombre	Departamento	Ubicación
1	Juan Pérez	Ventas	Ciudad de México
2	María Gómez	Ventas	Ciudad de México
3	Carlos Ruiz	TI	Monterrey
4	Laura Díaz	TI	Monterrey

Violación: La ubicación depende del departamento, no directamente del empleado, creando una dependencia transitiva.

Solución en 3FN: Separar tabla de Departamentos

Tabla: Empleados

IDEmpleado	Nombre	IDDepartamento
1	Juan Pérez	101
2	María Gómez	101
3	Carlos Ruiz	102
4	Laura Díaz	102

Tabla: Departamentos

IDDepartamento	NombreDepartamento	Ubicación
101	Ventas	Ciudad de México
102	TI	Monterrey

Ejemplo de Tercera Forma Normal (3FN)

- Supongamos que tenemos las siguientes tablas:

Tabla: Facturas

Número Factura	Municipio	Fecha	Apellido	Nombre	Número Cliente	Calle	Número	Código Postal
335	Sevilla	01.08.2022	Rodríguez	Facundo	21	Calle Rivero	41	41004
336	Sevilla	02.08.2022	Martínez	Natalia	22	Calle San Eloy	82	41001
337	Huelva	02.08.2022	García	Carlos	23	Avenida de Italia	245	21003

Ejemplo de Tercera Forma Normal (3FN) - Continuación

Tabla: Items

Número Factura	Ítem Factura	Artículo	Código Artículo	Precio
335	1	Tecladoinalámbrico	2 – 0023 – D	245
335	2	CableredUTPRJ45	4 – 0023 – D	0,90
335	3	CámarawebUSB	5 – 0023 – D	165
336	1	Auricularesinalámbricos	1 – 0023 – D	175
336	2	MiniAdaptadorUSB – CUSB2,0	3 – 0023 – D	0,75
337	1	Tecladoinalámbrico	2 – 0023 – D	245

Ejemplo de Tercera Forma Normal (3FN) - Normalización

- En la tabla Facturas, los atributos como nombre, apellido, calle, etc., dependen de número cliente, lo que genera una dependencia transitiva.
- En la tabla Items, los atributos artículo y precio dependen de código artículo, pero no directamente de la clave primaria.
- Para cumplir con 3FN, normalizamos en las siguientes tablas:

Facturas Normalizada

Facturas Normalizada

Número Factura	Fecha	Número Cliente
335	01,08,2022	21
336	02,08,2022	22
337	02,08,2022	23

Clientes Normalizados

✓

Id
cod Pos

Clientes

Número Cliente	Apellido	Nombre	Calle	Número	Código Postal	Municipio
21	Rodríguez	Facundo	CalleRivero	41	41004	Sevilla
22	Martínez	Natalia	CalleSanEloy	82	41001	Sevilla
23	García	Carlos	AvenidaItalia	245	21003	Huelva

cod Pos
Id | mun

Items Normalizados

Items

Número Factura	Ítem Factura	Código Artículo	Unidades
335	1	2 – 0023 – <i>D</i>	10
335	2	4 – 0023 – <i>D</i>	12
335	3	5 – 0023 – <i>D</i>	1
336	1	1 – 0023 – <i>D</i>	2
336	2	3 – 0023 – <i>D</i>	2
337	1	2 – 0023 – <i>D</i>	25

Artículos Normalizados

Artículos

Código Artículo	Artículo	Precio
2 – 0023 – D	Teclado inalámbrico	245
4 – 0023 – D	Cable red UTP RJ45	0,90
5 – 0023 – D	Cámara web USB	165
1 – 0023 – D	Auriculares inalámbricos	175
3 – 0023 – D	Mini Adaptador USB – C USB 2,0	0,75

Boyce-Codd Normal Form (BCNF)

La Boyce-Codd Normal Form (BCNF) es una versión más estricta de la Tercera Forma Normal (3FN). Una relación está en BCNF si, para cada una de sus dependencias funcionales, la clave primaria es la única que determina todos los atributos de la relación.

Regla: Si una relación tiene una dependencia funcional $A \rightarrow B$, entonces A debe ser una superclave.

Ejemplo de Boyce-Codd Normal Form (BCNF)

Consideremos la siguiente tabla:

Curso	Estudiante	Profesor
<i>Matemáticas</i>	<i>Juan</i>	<i>Sra.López</i>
<i>Física</i>	<i>Pedro</i>	<i>Sr.Martínez</i>
<i>Matemáticas</i>	<i>Ana</i>	<i>Sra.López</i>
<i>Física</i>	<i>Laura</i>	<i>Sr.Martínez</i>

En esta tabla, la dependencia funcional $\text{Curso} \rightarrow \text{Profesor}$ existe, pero Curso no es una superclave (porque no determina de manera única a cada fila). Esta tabla no está en BCNF.

Para normalizarla, creamos dos tablas:

Curso	Profesor
<i>Matemáticas</i>	<i>Sra.López</i>
<i>Física</i>	<i>Sr.Martínez</i>

Ejemplo BCNF

Y otra para los estudiantes inscritos:

Curso	Estudiante
<i>Matemáticas</i>	<i>Juan</i>
<i>Física</i>	<i>Pedro</i>
<i>Matemáticas</i>	<i>Ana</i>
<i>Física</i>	<i>Laura</i>

Ahora, ambas tablas están en BCNF, ya que no existen dependencias donde un atributo no clave determine un atributo que no sea parte de una clave.

Tabla: Personas

id	Nombre	Sexo
1	<i>Juan</i>	<i>Masculino</i>
2	<i>Ana</i>	<i>Femenino</i>
3	<i>Luis</i>	<i>Masculino</i>

Tabla: Personas Normalizada

id	id_sexo
1	1
2	2
3	1

Tabla: Sexo

id_sexo	Sexo
1	<i>Masculino</i>
2	<i>Femenino</i>

Cuarta Forma Normal (4NF)

Solo depende de uno y de nadie mas.

- Una tabla está en 4NF si:
 - Está en 3NF.
 - No contiene dependencias multivaluadas; es decir, no existen conjuntos de valores multivaluados que sean independientes entre sí pero que se almacenan en la misma tabla. Sea A, B y C eventos independientes se tiene: $A \twoheadrightarrow B$
 $A \twoheadrightarrow C$

Tabla: Áreas de Reparto Pizzería

P_K *F_K* *F_K*

Pizzería	Variedad de Pizza	Área de Entrega
PizzaPlaneta	PanGrueso	Centro
PizzaPlaneta	PanGrueso	Guerrero
PizzaPlaneta	PanGrueso	Juárez
PizzaPlaneta	PanRelleno	Centro
PizzaPlaneta	PanRelleno	Guerrero
PizzaPlaneta	PanRelleno	Juárez
PizzasToño	PanDelgado	Juárez
PizzasToño	PanRelleno	Juárez
PizzasLorena	PanGrueso	Centro
PizzasLorena	PanGrueso	Guerrero
PizzasLorena	PanDelgado	Centro
PizzasLorena	PanDelgado	Guerrero

Tabla: Variedades por Pizzería

 P_K T_K

Pizzería	Variedad de Pizza
<i>PizzaPlaneta</i>	<i>PanGrueso</i>
<i>PizzaPlaneta</i>	<i>PanRelleno</i>
<i>PizzasToño</i>	<i>PanDelgado</i>
<i>PizzasToño</i>	<i>PanRelleno</i>
<i>PizzasLorena</i>	<i>PanGrueso</i>
<i>PizzasLorena</i>	<i>PanDelgado</i>

Tabla: Áreas de Entrega por Pizzería

P_k f_k

Pizzería	Área de Entrega
<i>PizzaPlaneta</i>	<i>Centro</i>
<i>PizzaPlaneta</i>	<i>Guerrero</i>
<i>PizzaPlaneta</i>	<i>Juárez</i>
<i>PizzasToño</i>	<i>Juárez</i>
<i>PizzasLorena</i>	<i>Centro</i>
<i>PizzasLorena</i>	<i>Guerrero</i>

Quinta Forma Normal (5NF)

Mejor por partes que entero

- Una tabla está en 5NF si:
 - Está en 4NF.
 - forma normal de **proyección-uniión**, es un nivel de normalización de bases de datos diseñado para reducir redundancia en las bases de datos relacionales que guardan hechos multi-valores aislando semánticamente relaciones múltiples relacionadas.

Tabla Original (Vendedor, Empresa, Producto)



Vendedor	Empresa	Producto
V1	E1	Aspiradora
V1	E1	Panera
V1	E2	Tijeras de podar
V1	E2	Aspiradora
V2	E2	Panera
V2	E2	Base de sombrilla
V2	E2	Aspiradora
V3	E2	Telescopio
V3	E1	Aspiradora
V3	E1	Lámpara de lava
V3	E3	Corbatero

$L_1 - L_2 - L_3$

División en Tres Tablas

- **Tabla VP:** Relaciona Vendedor y Producto

Vendedor	Producto
V1	Aspiradora
V1	Panera
V1	Tijeras de podar
V2	Aspiradora
V2	Panera
V2	Base de sombrilla
V3	Telescopio
V3	Aspiradora
V3	Lámpara de lava
V3	Corbatero

Frame Title

Tabla VE: Relaciona Vendedor y Empresa

Vendedor	Empresa
V1	E1
V1	E2
V2	E2
V3	E1
V3	E2
V3	E3

Tabla EP: Relaciona Empresa y Producto

Empresa	Producto
E1	Aspiradora
E1	Panera
E1	Lámpara de lava
E2	Tijeras de podar
E2	Aspiradora
E2	Panera
E2	Base de sombrilla
E2	Telescopio

Sexta Forma Normal (6NF)

$\text{año} - \text{su} - \text{eq}$ | $\frac{\text{Tiempo}}{\text{su}}$ - año - plant
 Plor/eq - Jugador

- Se centra en descomponer las tablas hasta el nivel en que cada una represente un solo hecho, especialmente útil en bases de datos temporales donde los datos cambian con el tiempo.
- Nota:** Es menos común en aplicaciones prácticas y se utiliza en contextos específicos donde se requiere un alto nivel de descomposición.

Tabla Original: Estudiantes y Resultados

Student_ID	Student_FirstName	Student_LastName	Marks
S01	Tom	Alter	90
S02	Jacob	Watson	80
S03	Harry	Smith	85

Descomposición en 6NF

Ahora descompondremos la tabla en varias tablas para cumplir con la 6FN.

- **Tabla: StudentFirstName**

Student_ID	Student_FirstName
S01	Tom
S02	Jacob
S03	Harry

- **Tabla: StudentLastName**

Student_ID	Student_LastName
S01	Alter
S02	Watson
S03	Smith

- **Tabla: StudentResult**

Student_ID	Marks
S01	90
S02	80
S03	85

Codigo postal mexicano

Utilicemos un archivo en el que trabajaremos para normalizar: el de los códigos postales de México. La oficina de correos de México ofrece una versión en modo de texto donde vienen poco más de 145 mil registros con los ~~códigos postales~~ y las poblaciones en la República Mexicana.

Web scrapping

- Es una técnica para extraer información de sitios web de forma automática.
- Se utiliza para recolectar datos públicos desde páginas HTML.
- Permite automatizar la lectura de contenido visible en páginas web.
- Herramientas comunes: **requests**, **BeautifulSoup**, **Selenium**.
- Ejemplos de uso:
 - Recolectar noticias.
 - Obtener datos de precios.
 - Capturar tablas o reportes públicos.

Web Scraping con BeautifulSoup

- Es una librería de Python para analizar documentos HTML y XML.
- Facilita la navegación por el árbol DOM de una página web.
- Ideal para páginas estáticas (no generadas dinámicamente con JavaScript). Único
- Permite buscar etiquetas, clases, atributos y extraer texto.
- Comúnmente se usa junto con requests.

Automatización con Selenium

BI

- Framework para automatizar navegadores web.
- Permite interactuar con sitios dinámicos (JavaScript, AJAX).
- Simula el comportamiento de un usuario: clics, scroll, formularios.
- Se puede usar con Chrome, Firefox, Edge, entre otros.
- Ideal cuando BeautifulSoup no puede acceder a ciertos datos.

Simulación de Datos con Faker

- Librería de Python para generar datos falsos de prueba de forma realista.
- Útil para simular:
 - Nombres, direcciones, correos, teléfonos.
 - Fechas, ubicaciones, textos aleatorios.
- Soporta múltiples idiomas, incluyendo español.
- Ayuda a poblar bases de datos sin necesidad de datos reales sensibles.