

# TEMA 3: NORMALIZACIÓN DE BASES DE DATOS

# Índice

# Normalización

---

- 1- ¿Qué es la Normalización de Bases de Datos?
- 2- ¿Cuál es el propósito de la Normalización?
- 3- Tipos de dependencias
- 4- Ejemplo de dependencias
- 5- 1FN, 2FN, 3FN y FNBC.
- 6- Tabla comparativa 1FN, 2FN, 3FN y FNBC.
- 7- Primera Forma Normal (1FN)
- 8- Segunda Forma Normal (2NF)
- 9- Tercera Forma Normal (3FN)
- 10-Desnormalizar

# ¿Qué es la normalización de Bases de Datos?

- ❑ La normalización de base de datos es un principio de diseño de base de datos para organizar los datos de una manera consistente y estructurada.
- ❑ Consiste en simplificar las bases de datos siguiendo unas normas.

# ¿Cuál es el propósito de la normalización?

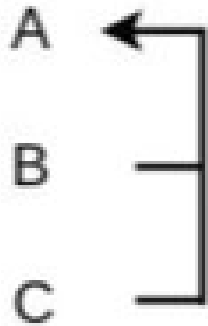
- ❑ Evitar redundancia de datos y mantener la integridad de la base de datos.
- ❑ Eliminar características indeseables asociados con la inserción, eliminación y actualización
- ❑ Eliminar valores duplicados.
- ❑ Organizar los datos de una manera consistente → varias tablas enlazadas con relaciones
- ❑ Los administradores de base de datos son capaces de lograr estas relaciones usando las llaves primarias, llaves foráneas, y llaves compuestas.

# Tipos de dependencias: funcional y transitiva

## Dependencia funcional

Los atributos B y C dependen funcionalmente de la clave primaria (A)

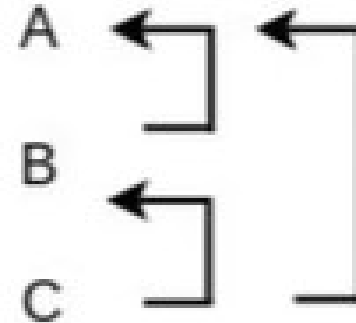
Para que exista B debe existir A y lo mismo pasa con C, para que exista C debe existir A, es decir, B y C dependen funcionalmente de A



## Dependencia transitiva

El atributo B depende funcionalmente de la clave primaria (A).

El atributo C depende de B que no es clave, pero como B depende de A, C también depende de A, es decir, C tiene una dependencia transitiva de la clave primaria (A)



# Ejemplo de dependencias

<u>idCliente</u>	nomCliente	apellido	numSerie	ordenador
1111	Manuel	García	AAAAAA	Portátil Asus Gaming
2222	Ana	López	BBBBBB	PC Sobremesa Fujitsu
3333	Carmen	Fernández	CCCCCC	Portátil MSI Gaming
4444	Pedro	Alonso	DDDDDD	Portátil Medion
5555	Elena	Pérez	EEEEEE	PC Sobremesa Lenovo

- ❑ Los atributos *nomCliente*, *apellido* y *numSerie* dependen **funcionalmente** de la clave *idCliente*.
- ❑ El atributo *ordenador* depende de *numSerie*, como *numSerie* depende de *idCliente*, el atributo *ordenador* depende **transitivamente** de la clave primaria *idCliente*.

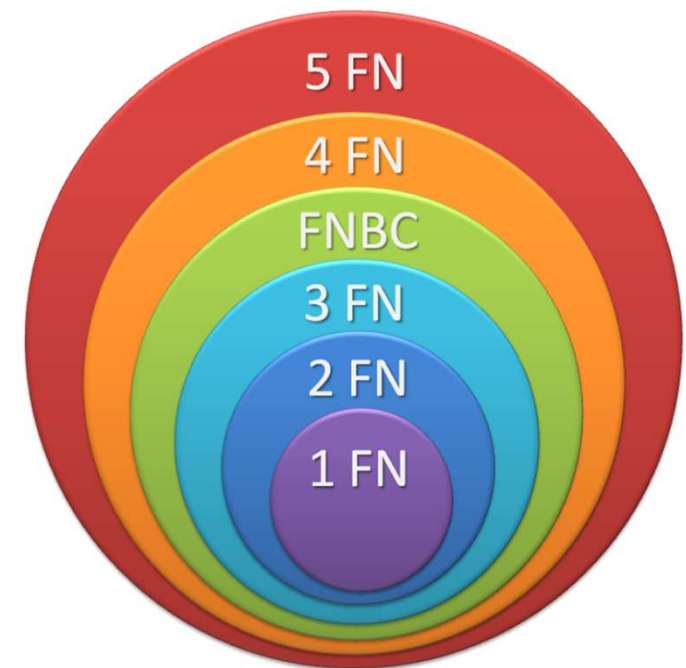
# 1FN, 2FN, 3FN y FNBC

Tres primeros tipos de normalización de base de datos:

- 1FN (primera forma normal)
- 2FN (segunda forma normal)
- 3FN (tercera forma normal)

Hay también:

- FNBC (forma normal Boyce-Codd)
- 4FN (cuarta forma normal)
- 5FN (quinta forma normal)
- 6FN (sexta forma normal)



En este tema estudiaremos hasta la 3FN

# 1FN , 2FN, 3FN y FNBC

<b>1FN</b>	<p>Las tablas tienen clave primaria.</p> <p>La clave primaria no tiene valores nulos.</p> <p>Cada atributo debe contener valores atómicos.</p> <p>No debe haber grupos repetidos de datos.</p> <p>Cada fila de la misma tabla debe ser única.</p>
<b>2FN</b>	<p>Debe estar en 1FN.</p> <p>La relación debe tener clave primaria preferentemente simple.</p> <p>Cada atributo que no es clave debe depender totalmente de TODA la clave principal. (dependen de toda la clave primaria, no sólo de una parte).</p> <p>Se basa en el concepto de dependencias funcionales.</p>
<b>3FN</b>	<p>Debe estar en 2FN.</p> <p>Todos los atributos no clave deben depender de un atributo que sea clave (no pueden depender transitivamente de la clave).</p> <p>Se basa en el concepto de dependencias transitivas.</p>
<b>FNBC</b>	<p>Una tabla está en Forma Normal de Boyce-Codd si está en 3FN.</p> <p>Los únicos atributos que dependen de otros atributos (si los hay), son claves candidatas.</p> <p>Es muy difícil que una tabla esté en 3FN y no lo esté en FNBC, se da en pocas ocasiones.</p>



# La Primera Forma Normal – 1FN

Para que una tabla esté en 1FN debe cumplir lo siguiente:

- ☐ Campos/atributos atómicos (una celda no debe contener más de un valor)
- ☐ Debe haber una clave primaria para identificación
- ☐ No puede haber filas duplicadas
- ☐ Los datos de una columna deben ser del mismo tipo

# ¿Está en 1FN?

TRABAJADOR		
<u>dni</u>	nombre	departamento
11111111-A	Carlos	Mantenimiento
22222222-B	Alicia	Dirección, Gestión

¿Todos los atributos son atómicos? No

La tabla tiene atributos que no son atómicos, tiene más de un valor en cada celda. Por tanto, no está en 1FN

TRABAJADOR		
<u>dni</u>	nombre	departamento
11111111-A	Carlos	Mantenimiento
22222222-B	Alicia	Dirección
22222222-B	Alicia	Gestión

Creamos una nueva fila para que todos los atributos sean atómicos. Ahora vemos que se generan grupos repetidos.

No está en 1FN. Separamos en 2 tablas

- Las tablas tienen clave primaria
- Atributos atómicos
- No hay grupos repetidos
- Ambas tablas están en 1FN

TRABAJADOR	
<u>dni</u>	nombre
11111111-A	Carlos
22222222-B	Alicia

TRABAJADOR-DEPARTAMENTO	
<u>dni</u>	<u>departamento</u>
11111111-A	Mantenimiento
22222222-B	Dirección
22222222-B	Gestión

# Normalización: ¿está en 1FN?

ORDENES								
<u>idOrden</u>	fecha	idCliente	nomCliente	ciudad	<u>idArticulo</u>	nomArticulo	cantidad	Precio
2301	23/02/2023	101	Carlos	Oviedo	1111	Ratón	3	15,00
2301	23/02/2023	101	Carlos	Oviedo	2222	Teclado	6	20,00
2301	23/02/2023	101	Carlos	Oviedo	3333	Altavoz	8	18,00
2302	25/02/2023	107	Alicia	Gijón	4444	Auriculares	4	35,00
2303	27/02/2023	110	Pedro	Avilés	2222	Teclado	2	20,00
2303	27/02/2023	110	Pedro	Avilés	5555	Alfombrilla	2	10,00

¿Todos los atributos son atómicos?  
**Sí**

**¿Hay grupos repetidos? Sí.** Aunque todos los atributos son atómicos, no está en 1FN porque existe grupos repetidos. Tenemos que separar en dos tablas. Para separar las tablas debemos añadir la clave primaria de la tabla ORDENES a la tabla ARTÍCULOS-ORDENES para poder relacionar ambas tablas.

ORDENES				
<u>idOrden</u>	fecha	idCliente	nomCliente	ciudad
2301	23/02/2023	101	Carlos	Oviedo
2302	25/02/2023	107	Alicia	Gijón
2303	27/02/2023	110	Pedro	Avilés

ARTICULOS-ORDENES				
<u>idOrden</u>	<u>idArticulo</u>	nomArticulo	cantidad	Precio
2301	1111	Ratón	3	15,00
2301	2222	Teclado	6	20,00
2301	3333	Altavoz	8	18,00
2302	4444	Auriculares	4	35,00
2303	2222	Teclado	2	20,00
2303	5555	Alfombrilla	2	10,00

# Normalización: ¿está en 2FN?

ORDENES				
idOrden	fecha	idCliente	nomCliente	ciudad
2301	23/02/2023	101	Carlos	Oviedo
2302	25/02/2023	107	Alicia	Gijón
2303	27/02/2023	110	Pedro	Avilés

Una tabla está en 2FN cuando todos los atributos no clave dependen de la clave primaria completa, no sirve conque dependan de una parte de ella solamente.

La tabla ORDENES está en 2FN. Todos los atributos no clave dependen de la clave primaria *idOrden*.

La tabla ARTÍCULOS-ORDENES no está en 2FN porque *nomArticulo* y *precio* dependen de *idArticulo*, pero no dependen de *idOrden*. Tendremos que crear una nueva tabla con estos atributos y la clave de la que dependen.

ARTICULOS-ORDENES				
idOrden	idArticulo	nomArticulo	cantidad	Precio
2301	1111	Ratón	3	15,00
2301	2222	Teclado	6	20,00
2301	3333	Altavoz	8	18,00
2302	4444	Auriculares	4	35,00
2303	2222	Teclado	2	20,00
2303	5555	Alfombrilla	2	10,00



ARTICULOS		
idArticulo	nomArticulo	Precio
1111	Ratón	15,00
2222	Teclado	20,00
3333	Altavoz	18,00
4444	Auriculares	35,00
5555	Alfombrilla	10,00

ARTICULOS-ORDENES		
idOrden	idArticulo	cantidad
2301	1111	3
2301	2222	6
2301	3333	8
2302	4444	4
2303	2222	2
2303	5555	2



# Normalización: ¿está en 3FN?

ORDENES				
<u>idOrden</u>	fecha	idCliente	nomCliente	ciudad
2301	23/02/2023	101	Carlos	Oviedo
2302	25/02/2023	107	Alicia	Gijón
2303	27/02/2023	110	Pedro	Avilés

Para estar en 3FN tenemos que separar en otra tabla todos los atributos no clave que dependan de otra columna no clave y eliminarlos de la tabla base.

En la nueva tabla añadiremos esos atributos dependientes y el atributo no clave del cual dependen.

ARTICULOS		
<u>idArticulo</u>	nomArticulo	Precio
1111	Ratón	15,00
2222	Teclado	20,00
3333	Altavoz	18,00
4444	Auriculares	35,00
5555	Alfombrilla	10,00

1- ARTICULOS: está en 3FN porque *nomArticulo* y *precio* dependen de la clave primaria *idArticulo*.

2- ARTICULOS-ORDENES: está en 3FN porque el atributo no clave *cantidad* depende de la clave primaria *idOrden-idArticulo*.

ARTICULOS-ORDENES		
<u>idOrden</u>	<u>idArticulo</u>	cantidad
2301	1111	3
2301	2222	6
2301	3333	8
2302	4444	4
2303	2222	2
2303	5555	2

3- ORDENES: no está en 3FN porque *nomCliente* y *ciudad* dependen del atributo *idCliente*, que no es clave primaria. Para normalizar la tabla, movemos los atributos no clave a una nueva tabla CLIENTES junto con la clave de la que dependen.

# Normalización: ¿está en 3FN?

ORDENES				
<u>idOrden</u>	fecha	idCliente	nomCliente	ciudad
2301	23/02/2023	101	Carlos	Oviedo
2302	25/02/2023	107	Alicia	Gijón
2303	27/02/2023	110	Pedro	Avilés



ORDENES		
<u>idOrden</u>	fecha	idCliente
2301	23/02/2023	101
2302	25/02/2023	107
2303	27/02/2023	110

CLIENTES		
<u>idCliente</u>	NomCliente	ciudad
101	Carlos	Oviedo
107	Alicia	Gijón
110	Pedro	Avilés

ORDENES no está en 3FN.

Para normalizar la tabla, movemos los atributos no clave a una nueva tabla CLIENTES junto con la clave de la que dependen.

Ahora tenemos cuatro tablas:

ORDENES(idOrden, fecha, idCliente)

CLIENTES (idCliente, nomCliente, ciudad)

ARTICULOS(idArticulo, nomArticulo, precio)

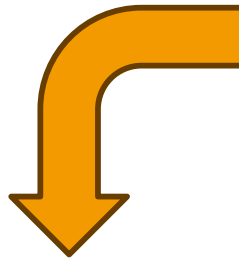
ARTICULOS-ORDENES(idOrden, idArticulo, cantidad)

ARTICULOS		
<u>idArticulo</u>	nomArticulo	Precio
1111	Ratón	15,00
2222	Teclado	20,00
3333	Altavoz	18,00
4444	Auriculares	35,00
5555	Alfombrilla	10,00

ARTICULOS-ORDENES		
<u>idOrden</u>	<u>idArticulo</u>	cantidad
2301	1111	3
2301	2222	6
2301	3333	8
2302	4444	4
2303	2222	2
2303	5555	2

# Normalización: resultado final

Tabla base sin normalizar



ORDENES								
<u>idOrden</u>	fecha	idCliente	nomCliente	ciudad	<u>idArticulo</u>	nomArticulo	cantidad	Precio
2301	23/02/2023	101	Carlos	Oviedo	1111	Ratón	3	15,00
2301	23/02/2023	101	Carlos	Oviedo	2222	Teclado	6	20,00
2301	23/02/2023	101	Carlos	Oviedo	3333	Altavoz	8	18,00
2302	25/02/2023	107	Alicia	Gijón	4444	Auriculares	4	35,00
2303	27/02/2023	110	Pedro	Avilés	2222	Teclado	2	20,00
2303	27/02/2023	110	Pedro	Avilés	5555	Alfombrilla	2	10,00

Tablas normalizadas:

CLIENTES		
<u>idCliente</u>	NomCliente	ciudad
101	Carlos	Oviedo
107	Alicia	Gijón
110	Pedro	Avilés

ARTICULOS		
<u>idArticulo</u>	nomArticulo	Precio
1111	Ratón	15,00
2222	Teclado	20,00
3333	Altavoz	18,00
4444	Auriculares	35,00
5555	Alfombrilla	10,00

ARTICULOS-ORDENES		
<u>idOrden</u>	<u>idArticulo</u>	cantidad
2301	1111	3
2301	2222	6
2301	3333	8
2302	4444	4
2303	2222	2
2303	5555	2

ORDENES		
<u>idOrden</u>	fecha	idCliente
2301	23/02/2023	101
2302	25/02/2023	107
2303	27/02/2023	110

ORDENES(idOrden, fecha, idCliente)

CLIENTES (idCliente, nomCliente, ciudad)

ARTICULOS(idArticulo, nomArticulo, precio)

ARTICULOS-ORDENES(idOrden, idArticulo, cantidad)

# Desnormalizar

- ❑ La desnormalización consiste en optimizar el funcionamiento de una base de datos por medio de agregar datos redundantes.
- ❑ Un modelo de datos desnormalizado no es lo mismo que un modelo de datos que no ha sido normalizado.
- ❑ La desnormalización solo se hará una vez alcanzado un nivel satisfactorio de normalización y de que hayan sido creadas las restricciones requeridas para ocuparse de las anomalías inherentes en el diseño.
- ❑ A veces puede ser necesario desnormalizar para facilitar el uso de la base de datos y mejorar el rendimiento de las consultas.