

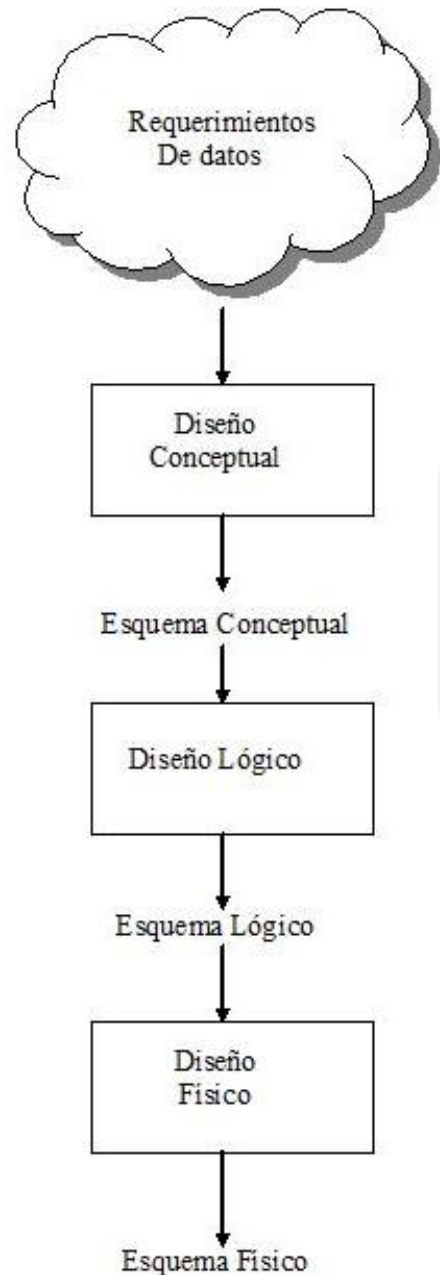
BBDD

Bases de Datos – Normalización



1. Contextualización
2. Introducción a la normalización
3. Definición del proceso de normalización
4. Problemas en relaciones “no normalizadas”
5. Introducción a las formas normales
6. Refinamiento del diseño de BD
7. Dependencias funcionales
8. Formas normales de Boyce Cod
9. Ayuda, ejemplos y actividades
10. Desnormalización

Modelización de BD



El modelado transcurre en tres fases:

1) Modelo conceptual: Modelo expresivo y entendible por el usuario experto en negocio (pero inexperto en informática). Representa el dominio del problema tal y como el usuario lo concibe. Nosotros trabajamos el **modelo E/R**.

2) Modelo lógico: Más técnico y que tienen traducción directa al modelo físico que entiende el futuro SGBD. Este modelo depende del tipo de BD (no es igual modelar una BD orientada a objetos que una relacional). Nosotros trabajamos el **Modelo relacional**.

3) Modelo físico: Resultado de aplicar el modelo lógico a un SGBD concreto. Habitualmente se expresa en un lenguaje de programación de BBDD como SQL. La transformación del Modelo Relacional al modelo físico se lleva a cabo mediante el sub-lenguaje **DDL (lenguaje de definición de datos)** de SQL.

Finalmente, a partir del modelo físico se llevará a cabo la **implementación de la BD**.

Introducción

La **normalización** permite obtener un conjunto adecuado de relaciones en nuestro modelo relacional (descomponiendo grandes relaciones en relaciones más pequeñas) de tal forma que:

- El esquema de la base de datos incluya el mínimo número de atributos necesarios para dar soporte a los requerimientos del sistema (redundancia mínima: salvo los atributos que forman parte de claves foráneas, los demás se representarán una única vez en la base de datos).
- Mejorar la independencia de los datos.
- Resulta mas fácil acceder a la base de datos y, sobre todo, mantener los datos de la base de datos (Simplifica el proceso de inserción, modificación y borrado de datos).



Normalización

La **normalización** de bases de datos es un proceso que consiste en designar y aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional con el objetivo de minimizar la redundancia y dependencia de los datos, eliminando duplicidades o dependencias innecesarias y facilitando así el uso y mantenimiento posterior de la base de datos.



También se conoce como normalización al proceso de obligar a los atributos de un diseño de base de datos a cumplir ciertas **formas normales**.

Problemas sin normalización

En las **relaciones no normalizadas**, surgen problemas de gestión tales como ...

- Tendremos datos repetidos innecesariamente.
La nacionalidad se repite en cada tupla a pesar de que depende del autor.
- Las actualizaciones de un determinado atributo supondrán numerosas actualizaciones.
Cambiar de nombre una editorial supondrá actualizar multitud de tuplas
- Los borrados de determinados registros supondrán la pérdida de otros datos.
Si borramos el libro El gnomo, perdemos la información del autora Ana.
- Tendremos atributos con más de un valor.
El libro La casa tiene dos años de edición porque tiene 2 ediciones.

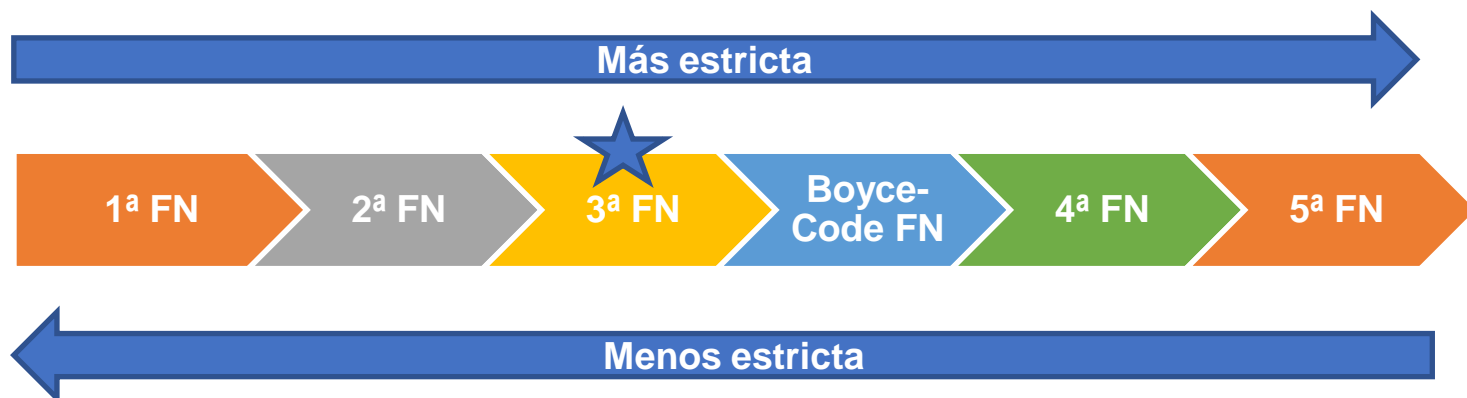
Autor	Nacionalidad	Código	Título	Editorial	Año
Pepe	española	123445	La casa	Pintadas	1987, 1988
Pepe	española	123453	Manzanas	Pintadas	2003
Pepe	española	423653	Picasso	Pintadas	2015
Ana	rusa	739840	El gnomo	Pintadas	2019
María	francesa	345731	Peter Pan	Planeta	2005
María	francesa	758328	100 recetas	Planeta	2017
Oskar	alemana	643200	Susurro	ABC	2017

Formas normales

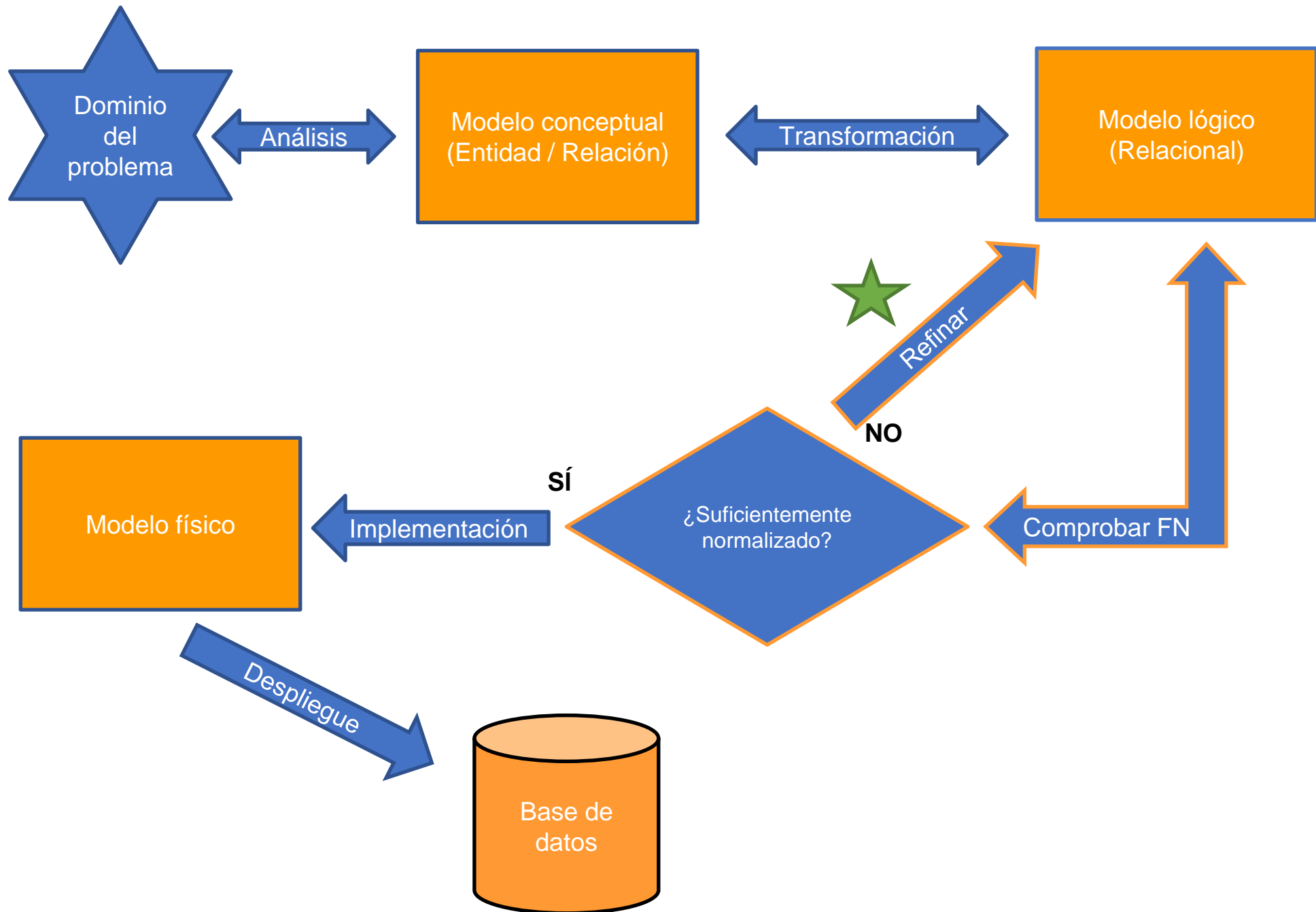
La **forma normal** de una base de datos es usada como parámetro de calidad de la misma. Permite medir su calidad y verificar si cumple los criterios de calidad deseados.

Cada forma normal exige que se cumplan determinadas restricciones sobre el conjunto de atributos de nuestro diseño lógico.

Existen 6 formas normales, cada una de las cuales agrupa a sus antecesoras.



Refinamiento de un diseño de BD



Dependencias funcionales

- Se dice que un atributo Y de una relación «**depende funcionalmente**» de otro atributo X de la misma relación si a todo valor de X le corresponde siempre el mismo valor de Y.
- Se representa como $X \rightarrow Y$
- No es recíproco, $Y \nrightarrow X$
- No tiene porque ser un solo atributo, puede ser un conjunto.

Por ejemplo $(\text{Nombre}, \text{Apellido1}, \text{Apellido2}) \rightarrow (\text{CP}, \text{Calle})$

PERSONAS

<u>Nombre</u>	<u>Apellido1</u>	<u>Apellido2</u>	CP	Calle
Atenea	Mylona	Anagnostopoulos	28910	La Paz
Bob	Smith	Stonebridge	28080	Sol
Sasha	Suznetsov	Petrov	28910	Avda. Constitución
Erika	Mylona	Anagnostopoulos	28920	La Paz

Dependencias funcionales completas

- Se dice que un atributo Y de una relación «tiene una **dependencia funcional completa**» con respecto a un conjunto de atributos X (x_1, x_2, \dots, x_n) de la misma relación si Y depende de todos los atributos de X a la vez pero no de ningún subconjunto de ellos.
- Se representa como $X \Rightarrow Y$
- No es recíproco, $Y \not\Rightarrow X$
- No tiene porque ser un solo atributo, puede ser un conjunto.

Ejemplo: Tenemos aeropuerto, pista, hora de aterrizaje y avión.

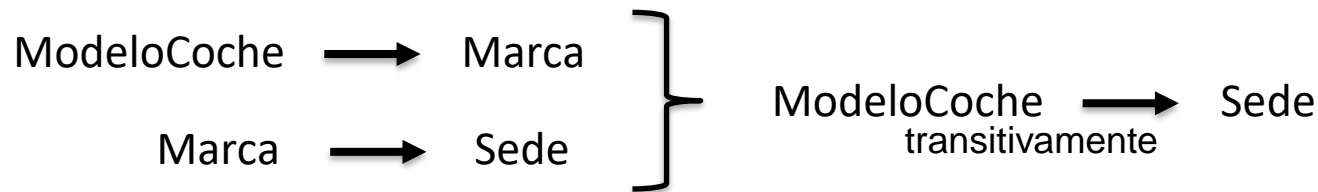
(Aeropuerto, Pista, Hora) \Rightarrow Avión

<u>Aeropuerto</u>	<u>Pista</u>	<u>Hora</u>	Avión
Adolfo Suárez	1	13:00	B747 - 23456
Prat	1	15:55	A380 - 11223
Adolfo Suárez	1	15:55	B747 - 23456
Adolfo Suárez	1	20:00	727-11111

Dependencia funcional transitiva

- Se dice que un atributo Z de una relación «tiene una **dependencia funcional transitiva**» con X a través de Y si $X \longrightarrow Y$ e $Y \longrightarrow Z$ siendo X, Y, Z atributos de una misma relación.

Ejemplo: Tenemos modelos de coche, marcas, sedes y motores



ModeloCoche	Marca	Sede	Motor
Ateca D	Seat	España	diésel
BMW X6 eléctrico	BMW	Alemania	eléctrico
Clio GTI	Renault	Francia	gasolina
Clase A GT	Mercedes	Alemania	gasolina
León TD	Seat	España	diésel

Formas normales de Boyce Codd

1FN: Todos los valores que componen cada atributo en la tupla **son atómicos**. No puede existir un atributo en una tupla que pueda tener dos o más valores. Debemos separar los valores llevándolos a una relación nueva. Los atributos deben depender funcionalmente de la clave.

2FN: 1FN + cada atributo de una relación que no está en la clave dependen funcionalmente de forma completa de la clave primaria de la relación. Debemos sacar los atributos que dependan de otros distintos de la clave. Si la clave es simple y está en 1FN, también lo estará en 2FN.

3FN: 2FN + cada atributo que no está en la clave primaria NO depende transitivamente de ella. Hay que eliminar dependencias transitivas sacando los atributos que dependan transitivamente de la clave. Es decir, no existen transitividades entre atributos que no pertenezcan a la clave.

FNBC: 3FN aplicada, no solo a la clave primaria sino también a las claves candidatas.

Ayuda sobre las formas normales

1FN: Como regla a aplicar, cuando se produce la separación de datos de la tabla original en una nueva tabla ésta, además de los atributos necesarios, traslada la clave primaria de la tabla original como parte de su nueva clave primaria, y por tanto estará formada generalmente por dos atributos.

2FN: La idea intuitiva de la 2FN es identificar todas las tablas con una clave compuesta, pues todas las tablas con clave simple están por defecto en 2FN si ya estuvieran en 1FN, y comprobar que cada uno de los campos de esta tabla dependen de la clave completa.

3FN: Si un atributo que no es clave depende de otro atributo que no es clave, la tabla posiblemente contiene datos acerca de más de una entidad, contradiciendo el principio de que cada tabla almacene información de una entidad.

Ejemplo I

¿Está en 1FN?

<u>DNI</u>	Nombre	Edad	Tfno
51111111	Mary	34	911111111 911111112
52222222	Erwin	23	911111113
53333333	Duncan	19	911111114
54444444	Ann	55	911111115

¡No! Existen atributos multivaluados (Tfno)

<u>DNI</u>	Nombre	Edad
51111111	Mary	34
52222222	Erwin	23
53333333	Duncan	19
54444444	Ann	55

<u>DNI</u>	<u>Tfno</u>
51111111	911111111
51111111	911111112
52222222	911111113
53333333	911111114
54444444	911111115

Ejemplo II

¿Está en 2FN?

<u>CodCliente</u>	<u>CodProducto</u>	FechaCompra	NomProd
C01	P01	12/11/2022	Ipad mini
C02	P02	19/04/2020	Imac 24"
C01	P01	01/10/2022	Ipad mini
C04	P04	29/09/2022	AppleTV
C05	P05	06/01/2021	Iphone Xs

¡No! Existen atributos que no tienen dependencia funcional completa sobre la clave primaria (NombreProd)

<u>CodCliente</u>	<u>CodProducto</u>	FechaCompra
C01	P01	12/11/2022
C02	P02	19/04/2020
C01	P01	01/10/2022
C04	P04	29/09/2022
C05	P05	06/01/2021

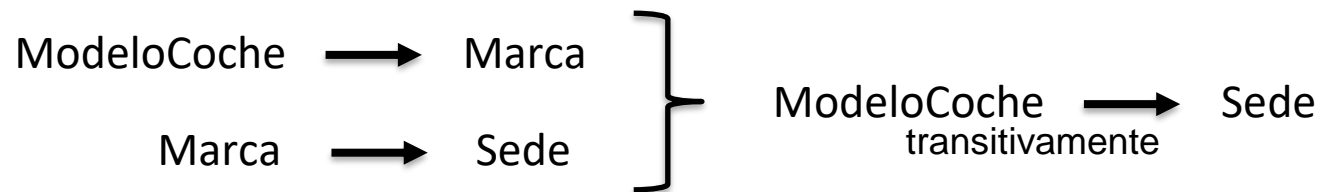
<u>CodProducto</u>	NombreProd
P01	Ipad mini
P02	Imac 24"
P04	AppleTV
P05	Iphone Xs

Ejemplo III

¿Está en 3FN?

<u>ModeloCoche</u>	Marca	Sede	Motor
Ateca D	Seat	España	diésel
BMW X6 eléctrico	BMW	Alemania	eléctrico
Clio GTI	Renault	Francia	gasolina
Clase A GT	Mercedes	Alemania	gasolina
León TD	Seat	España	diésel

¡No! Existen atributos que tienen dependencia funcional transitiva sobre la clave primaria (Sede)



<u>ModeloCoche</u>	Marca	Motor
Ateca	Seat	diésel
BMW X6	BMW	eléctrico
Clio	Renault	gasolina
Clase A	Mercedes	gasolina
León	Seat	diésel

<u>Marca</u>	Sede
Seat	España
BMW	Alemania
Renault	Francia
Mercedes	Alemania

Actividad I

Refina en 1FN la siguiente relación

Autor	Nacionalidad	<u>Código</u>	Título	Editorial	Año
Pepe	española	123445	La casa	Pintadas	1987, 1988
Pepe	española	123453	Manzanas	Pintadas	2003
Pepe	española	423653	Picasso	Pintadas	2015
Ana	rusa	739840	El gnomo	Pintadas	2019
María	francesa	345731	Peter Pan	Planeta	2005
María	francesa	758328	100 recetas	Planeta	2017
Oskar	alemana	643200	Susurro	ABC	2017

Autor	Nacionalidad	<u>Código</u>	Título	Editorial
Pepe	española	123445	La casa	Pintadas
Pepe	española	123453	Manzanas	Pintadas
Pepe	española	423653	Picasso	Pintadas
Ana	rusa	739840	El gnomo	Pintadas
María	francesa	345731	Peter Pan	Planeta
María	francesa	758328	100 recetas	Planeta
Oskar	alemana	643200	Susurro	ABC

<u>Código</u>	<u>Año</u>
123445	1987
123445	1988
123453	2003
423653	2015
739840	2019
345731	2005
758328	2017

Actividad II

Refina mediante normalización hasta 3FN la relación

¿Está en 1FN?

<u>Id_orden</u>	Fecha	Id_cliente	Nom_cliente	Estado	<u>Num_art</u>	<u>Nom_art</u>	Cantidad	Precio
2301	23/02/11	101	Martin	Caracas	3786	Red	3	35,00
2301	23/02/11	101	Martin	Caracas	4011	Raqueta	6	65,00
2301	23/02/11	101	Martin	Caracas	9132	Paq-3	8	4,75
2302	25/02/11	107	Herman	Coro	5794	Paq-6	4	5,00
2303	27/02/11	110	Pedro	Maracay	4011	Raqueta	2	65,00
2303	27/02/11	110	Pedro	Maracay	3141	Funda	2	10,00

No hay atributos multivaluados (ok). Pero NO tiene clave primaria. Elegiremos una:

Claves candidatas: opción 1 {Id_orden, Num_art} opción 2 {Id_orden, Nom_art}

Clave primaria: {Id_orden, Num_art}

ARTICULOS_POR_ORDEN

<u>Id_orden</u>	Fecha	Id_cliente	Nom_cliente	Estado	<u>Num_art</u>	Nom_art	Cantidad	Precio
2301	23/02/11	101	Martin	Caracas	3786	Red	3	35,00

Actividad II

Refina mediante normalización la siguiente relación

¿Está en 2FN?

<u>Id_orden</u>	Fecha	Id_cliente	Nom_cliente	Estado	<u>Num_art</u>	Nom_art	Cantidad	Precio
2301	23/02/11	101	Martin	Caracas	3785	Red	3	35,00
2301	23/02/11	101	Martin	Caracas	4011	Raqueta	6	65,00
2301	23/02/11	101	Martin	Caracas	9132	Paq-3	8	4,75
2302	25/02/11	107	Herman	Coro	5794	Paq-6	4	5,00
2303	27/02/11	110	Pedro	Maracay	4011	Raqueta	2	65,00
2303	27/02/11	110	Pedro	Maracay	3141	Funda	2	10,00

NO, existen atributos que no tienen dependencia funcional completa con la clave por lo que eliminamos las dependencias funcionales sobre la clave primaria que no sean completas:

ARTICULOS_POR_ORDEN (Id_orden, Num_art, Cantidad)

ORDENES (Id_orden, Fech, Id_cliente, Nom_cliente, Estado)

ARTICULOS (Num_art, Nom_art, Precio)

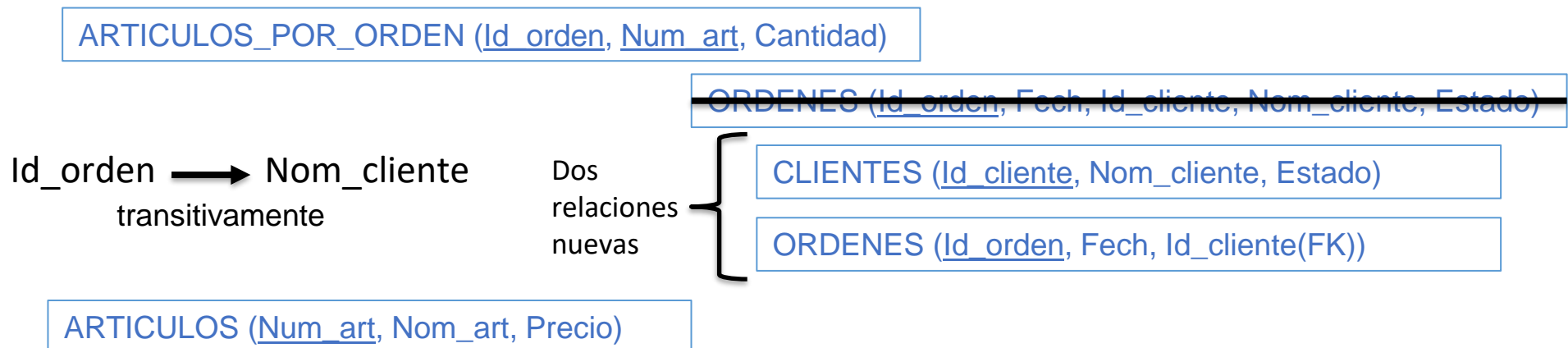
Actividad II

Refina mediante normalización la siguiente relación

¿Está en 3FN?

<u>Id_orden</u>	Fecha	Id_cliente	Nom_cliente	Estado	<u>Num_art</u>	Nom_art	Cantidad	Precio
2301	23/02/11	101	Martin	Caracas	3786	Red	3	35,00
2301	23/02/11	101	Martin	Caracas	4011	Raqueta	6	65,00
2301	23/02/11	101	Martin	Caracas	9132	Paq-3	8	4,75
2302	25/02/11	107	Herman	Coro	5794	Paq-6	4	5,00
2303	27/02/11	110	Pedro	Maracay	4011	Raqueta	2	65,00
2303	27/02/11	110	Pedro	Maracay	3141	Funda	2	10,00

NO, existen atributos que tienen dependencia funcional transitiva con la clave por lo que eliminamos dichas dependencias:



Actividad III

Refina mediante normalización la siguiente relación

Codigo_profe	Nombre	Apellido	Hijos	Cumpleanos
P01	Joaquín	Pérez	Luis, Carlos	23/05/2003, 18/12/2007
P02	María	Rodríguez	Andrea	09/09/2009
P03	Ramón	Menéndez	Sonia, Lucía	10/03/2007, 11/04/2009
P04	Ángela	García	Pedro, Juan, Carlos	28/12/2015, 30/06/2020, 01/11/2022
P05	Manuel	Díaz	Martín, Elena	03/01/2020, 08/05/2021

Desnormalización

- En ocasiones durante la reestructuración de relaciones, se produce un **proceso de desnormalización** en el cual se busca optimizar el desarrollo de la BD mejorando los tiempos de respuesta (por ejemplo, disponer de atributos que estén frecuentemente presentes en una consulta, dentro de la misma tabla aunque ello suponga tener información redundante o transitividad).

Nota: siempre se respetará la 1FN.

Nota: cumplir la tercera forma normal, aunque en teoría es deseable, **no siempre es práctico** y puede penalizar el rendimiento de la base de datos. Si tenemos una tabla "Clientes" y queremos eliminar todas las dependencias posibles entre los campos, tendremos que crear numerosas tablas independientes para las ciudades, códigos postales, ... Eso supondrá muchas tablas pequeñas que podrán degradar el rendimiento o superar la capacidad de memoria o de archivos abiertos. En la práctica, puede ser más factible aplicar la tercera forma normal sólo a los datos que cambian con frecuencia.