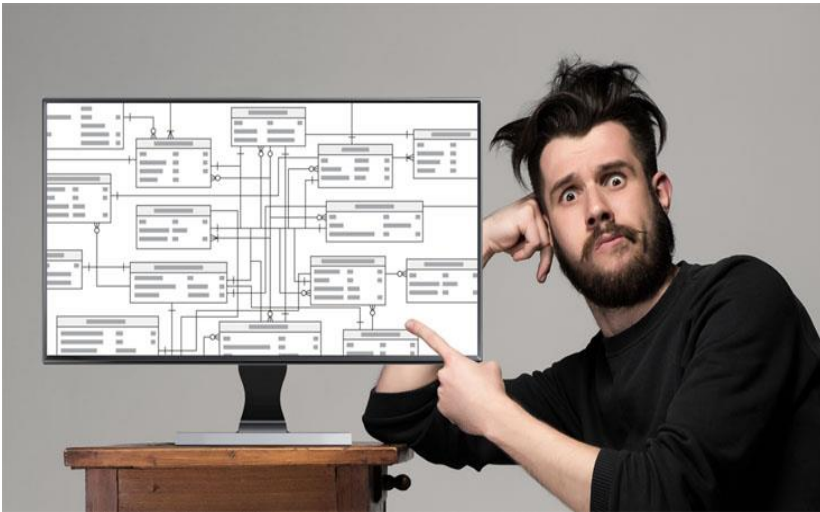


# GESTIÓN DE BASES DE DATOS

## DISEÑO LÓGICO: NORMALIZACIÓN



Francisco Pérez  
franciscoprdv@gmail.com

[Esta obra está bajo una licencia de Creative Commons  
Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional](#)

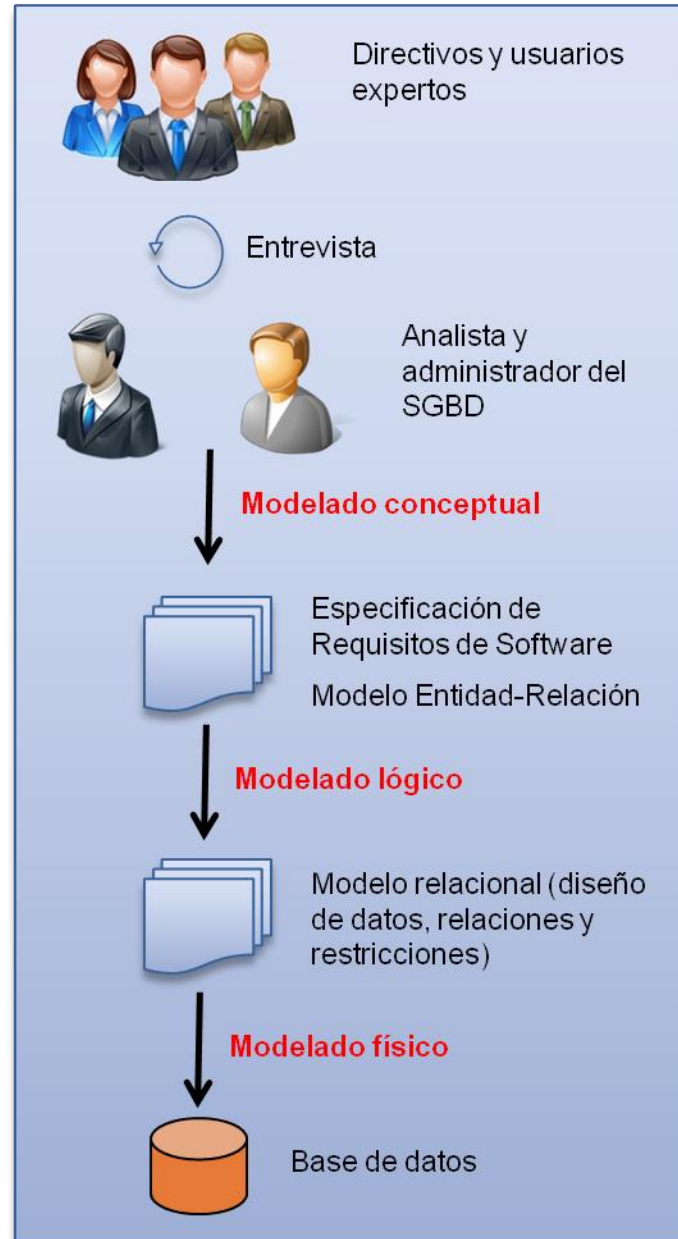
# CONTENIDOS

1. NOMENCLATURA UTILIZADA EN EL MODELADO DE DATOS
2. NORMALIZACIÓN Y FORMAS NORMALES
3. PRIMERA FORMA NORMAL
4. SEGUNDA FORMA NORMAL
5. TERCERA FORMA NORMAL
6. FORMA NORMAL DE BOYCE-CODD
7. DENORMALIZACIÓN

# NORMALIZACIÓN

## 1. NOMENCLATURA UTILIZADA EN EL MODELADO DE DATOS

ERS	Modelo Entidad-Relación	Modelo Lógico	Modelo Físico
Lenguaje natural	Entidades	Relaciones	Tablas
	Relaciones		
	Atributos	Atributos	Columnas
	Ocurrencia	Tuplas	Filas
	Atributo identificador principal	Atributo clave	Clave primaria



# NORMALIZACIÓN

## 2. NORMALIZACIÓN Y FORMAS NORMALES

Durante la etapa de modelado lógico, tras la obtención del modelo relacional, suele ser necesario realizar una etapa de normalización, que permite obtener un modelo lógico optimizado.

La normalización consiste en una serie de refinamientos realizados sobre el modelo lógico.

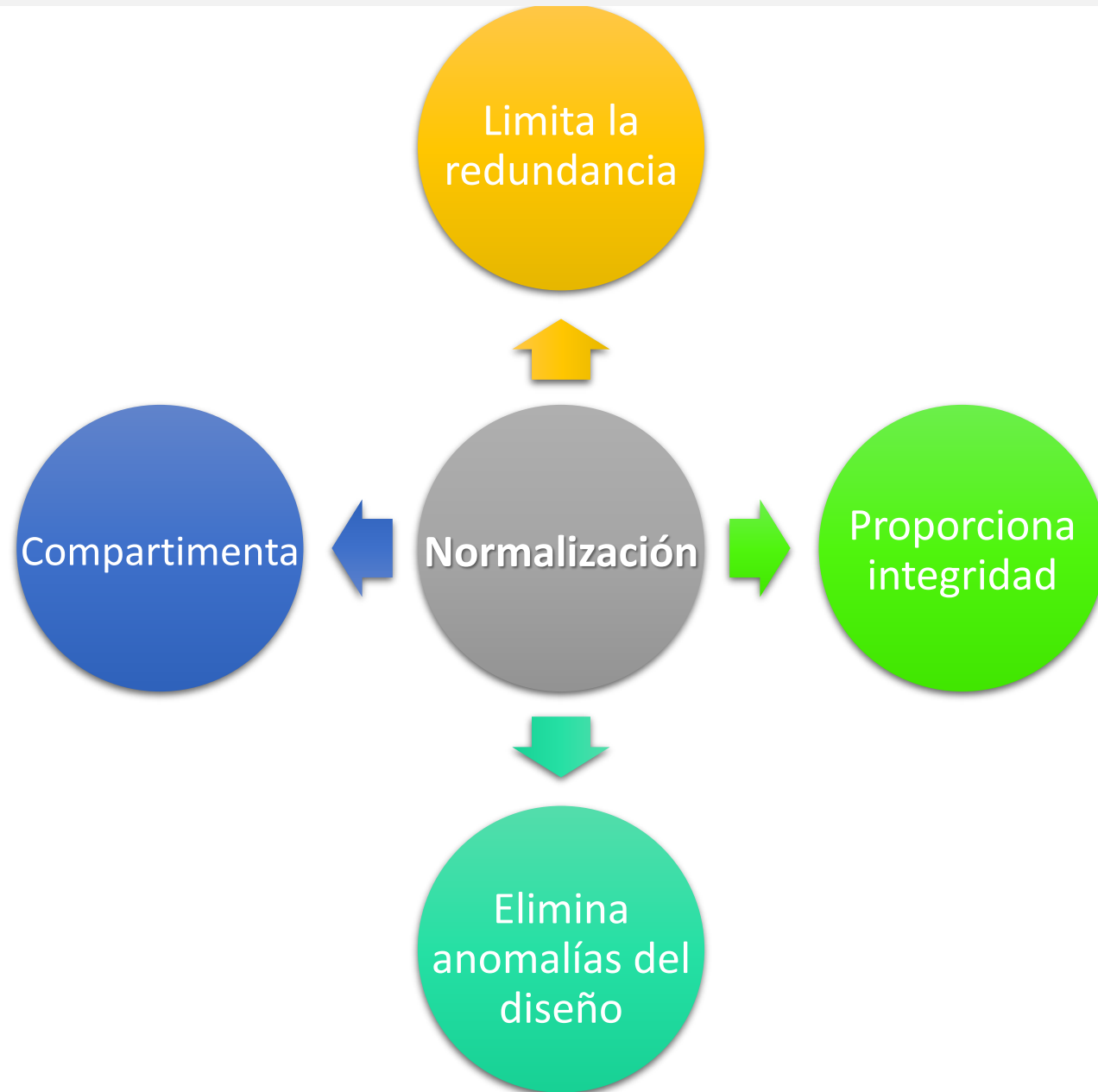
Una **forma normal** o forma normalizada es un parámetro que permite medir la calidad del diseño de una base de datos.

Para llevar una base de datos a una forma normal hay que imponer una serie de restricciones a los atributos del modelo relacional.

Existen 6 *formas normales*.

Conforme se alcanzan formas normales superiores, el diseño mejora conteniendo la redundancia, compartimentando la información y eliminando anomalías.

Para alcanzar una forma normal de orden 'n', primero es necesario haber alcanzado la forma normal de orden inferior, 'n-1'.



#### DEPENDENCIA FUNCIONAL

Para normalizar una relación es necesario aplicar el concepto de dependencia funcional

Dada una relación que contiene los atributos 'X' e 'Y', se dice que 'Y' **depende funcionalmente** de 'X' (o que 'X' **determina** a 'Y', representándolo como  $X \rightarrow Y$ ), si para cada valor de 'X' hay asociado un único valor de 'Y'.

Esto es equivalente a decir que si  $X \rightarrow Y$ , para cada valor de 'X', existe un único valor de 'Y'

Si  $X \rightarrow Y$ , cuando dos tuplas de una relación tienen el mismo valor para un atributo 'X', forzosamente han de tener el mismo valor para el atributo 'Y' (lo contrario no tiene porqué ser cierto).

## EJEMPLO DE DEPENDENCIA FUNCIONAL

- En la siguiente tabla se muestran datos de películas:

<u>Cód_Película</u>	Título	Año	Duración
1	Star Wars I	1980	120
6	Star Trek XI	2007	133
7	Back to the future	1985	97
9	Matrix Revolutions	2007	102

- Si nos fijamos en la primera fila, el atributo **Cód\_Película=1** identifica a una película con un título, año y duración determinada. Siempre que se haga referencia al atributo **Cód\_Película** con valor 1, tendremos que los valores de los atributos **Título**, **Año** y **Duración** son siempre los mismos.
- Por tanto, **Cód\_Película** **determina** a **Título**, **Año** y a **Duración** (o lo que es lo mismo, **Título**, **Año** y **Duración** tienen dependencia funcional respecto a **Cód\_Película**):

**Cód\_Película** → **Título**

**Cód\_Película** → **Año**

**Cód\_Película** → **Duración**



## CONTRA EJEMPLO DE DEPENDENCIA FUNCIONAL

- En la siguiente tabla se muestran datos de películas:

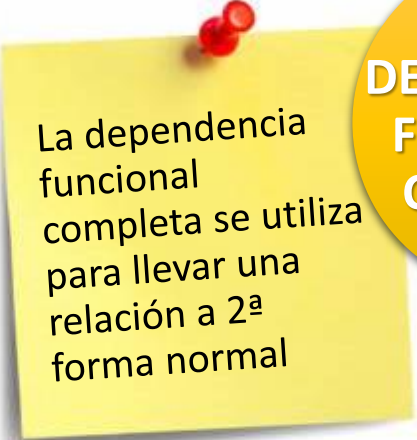
<u>Cód_Película</u>	Título	Año	Duración
1	Star Wars I	1980	120
6	Star Trek XI	2007	133
7	Back to the future	1985	97
9	Matrix Revolutions	2007	102

- Si buscamos todas las filas en las que **Año=2007**, los valores de *Cód\_Película*, *Título* y *Duración* no son los mismos siempre en todas las filas (hay o puede haber varias películas distintas en ese mismo año). Por tanto:

**Año**  $\nrightarrow$  **Cód\_Película**

**Año**  $\nrightarrow$  **Título**

**Año**  $\nrightarrow$  **Duración**



La dependencia funcional completa se utiliza para llevar una relación a 2ª forma normal

#### DEPENDENCIA FUNCIONAL COMPLETA

Dada una relación que contiene los atributos:

$X_1, X_2, X_3, \dots, X_i, \dots, X_j, \dots, X_n$

siendo  $\{\underline{X_1}, \dots, \underline{X_i}\}$  el atributo clave de la relación, se dice que *el atributo  $X_j$  tiene dependencia funcional completa respecto de la clave si toda la clave determina al atributo  $X_j$  pero ninguno de los atributos de la clave  $\{\underline{X_1}, \dots, \underline{X_i}\}$  por separado (ni un subconjunto de ellos) determina a  $X_j$*

Sólo es necesario verificar la existencia de dependencia funcional completa cuando la clave está formada por dos o más atributos.

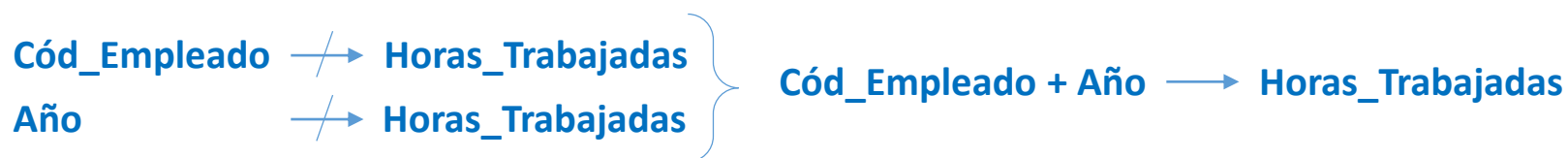
Una clave formada por un único atributo siempre tiene dependencia funcional respecto a los demás atributos.

## EJEMPLO DE DEPENDENCIA FUNCIONAL COMPLETA

- En la siguiente tabla se muestran las horas trabajadas anualmente por unos empleados:

<u>Cód_Empleado</u>	<u>Año</u>	Horas_Trabajadas
1457	2016	1023
1501	2013	254
985	2016	5
985	2012	341

- Para verificar si existe una dependencia funcional completa comprobamos lo siguiente:
  - El atributo clave está formado por la combinación de dos o más atributos: **Cod\_Empleado** y **Año**.
  - Por sí sólo, el atributo **Cod\_Empleado** no determina al atributo **Horas\_Trabajadas**.
  - Por sí sólo, el atributo **Año** no determina al atributo **Horas\_Trabajadas**.
  - Conjuntamente, **Cod\_Empleado** y **Año** sí determinan al atributo **Horas\_Trabajadas**



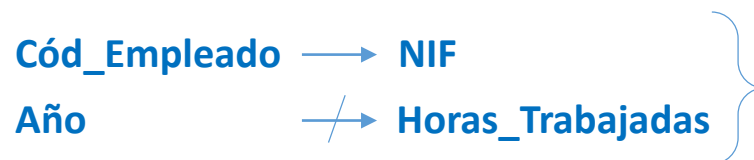
Por tanto, el atributo **Horas\_Trabajadas** tiene dependencia funcional completa respecto a la clave.

## CONTRA EJEMPLO DE DEPENDENCIA FUNCIONAL COMPLETA

- En la siguiente tabla se muestran las horas trabajadas anualmente por unos empleados. Para generar una anomalía, en la tabla se ha incluido la columna 'NIF':

<u>Cód_Empleado</u>	<u>Año</u>	NIF	Horas_Trabajadas
1457	2016	12345678Z	1023
1501	2013	23232323A	254
985	2016	46468585T	5
985	2012	46468585T	341

- Para verificar si existe una dependencia funcional completa comprobamos lo siguiente:
  - El atributo clave está formado por la combinación de dos o más atributos: **Cod\_Empleado** y **Año**.
  - Por sí sólo, el atributo **Cod\_Empleado** determina al atributo **NIF**. *Esta dependencia parcial impide que haya dependencia funcional completa*
  - Por sí sólo, el atributo **Año** no determina al atributo **NIF**.



Por tanto, no se cumplen las condiciones para establecer una dependencia funcional completa entre el atributo **NIF** y la clave.

La dependencia funcional transitiva se utiliza para llevar una relación a 3ª forma normal

#### DEPENDENCIA FUNCIONAL TRANSITIVA

Si 'X', 'Y' y 'Z' son atributos de una relación, se dice que 'X' *determina a 'Z' de manera transitiva a través de 'Y'*, si la dependencia se establece indirectamente a través de un atributo intermedio 'Y', es decir:

- $X \longrightarrow Z$

si se cumple lo siguiente:

- $X \longrightarrow Y$

- $Y \longrightarrow Z$

- $Y \not\longrightarrow X$

Sólo es necesario verificar la existencia de dependencia funcional transitiva cuando la relación contiene más de un atributo no-clave.

Una relación con un único atributo no-clave nunca tiene dependencia funcional transitiva.

## EJEMPLO DE DEPENDENCIA FUNCIONAL TRANSITIVA

- En la siguiente tabla se muestra la población de distintas áreas postales (áreas geográficas delimitadas por un código postal)

...	Continente	País	ZIP (C.P.)	Habitantes
...	1	34	30205	250.000
...	1	34	30005	500.000
...	2	76	99501	1.500
...	2	83	170137	2.527.600
...	3	7	BN1 1DA	26.980

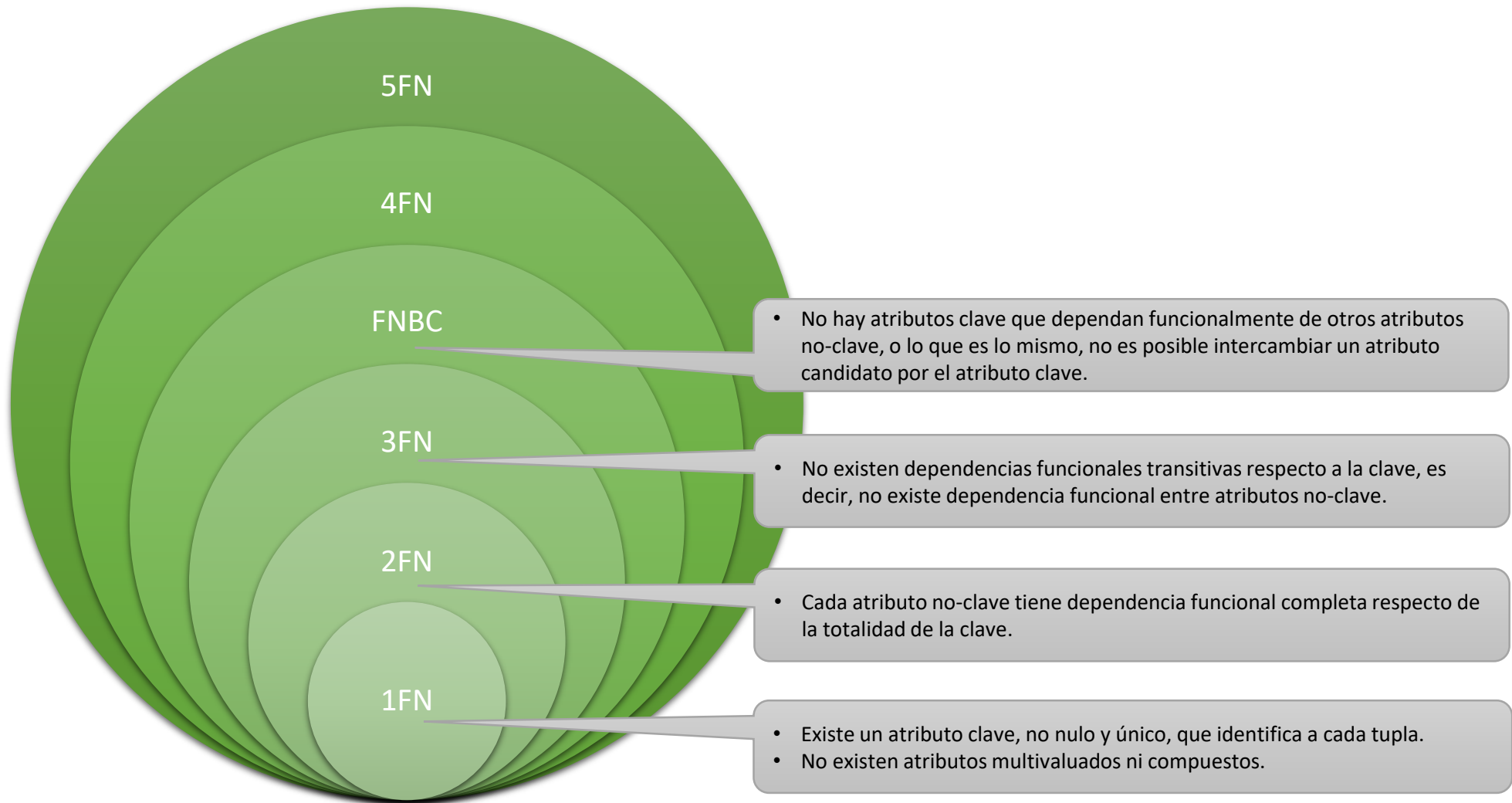
- Para verificar si existe una dependencia funcional transitiva comprobamos lo siguiente:

- El atributo *ZIP* (el código postal de una zona) determina al atributo *País*:
- El atributo *País* determina al atributo *Continente*:
- El atributo *País* no determina al atributo *ZIP*:

*ZIP* → *País*  
*País* → *Continente*  
*País* ↛ *ZIP*

En estas condiciones se cumple la dependencia transitiva:

*ZIP* → *Continente*



- Una relación se encuentra en Primera Forma Normal si se cumple lo siguiente:



- No hay dos tuplas (filas) iguales, es decir, existe una clave única y no nula que identifica a cada tupla, formada por un **atributo clave** (o un conjunto de ellos).
- Cada atributo sólo puede tomar un único valor atómico (es decir, no hay atributos **multivaluados** ni atributos **compuestos**).

- En caso de existir atributos multivaluados o compuestos, es necesario modificar la relación para que cada atributo represente un valor atómico.
- La Primera Forma Normal es un requisito esencial para una relación: constituye la base de la integridad referencial y compartimenta la información de una relación en elementos indivisibles.



- La eliminación de atributos multivaluados puede realizarse siguiendo una de estas estrategias:

### ALTERNATIVA 1: SEGMENTAR EL ATRIBUTO MULTIVALUADO

Tabla con atributos multivaluados (Apellidos):

<u>Cód Empleado</u>	Nombre	Apellidos
1457	Balbino	Bechamel Pérez
1501	Jetulio	Pencas Morcillo
985	Cleofás	Tijerilla
985	Genaro	Tormo Cansino

Normalización



<u>Cód Empleado</u>	Nombre	Apellido1	Apellido2
1457	Balbino	Bechamel	Pérez
1501	Jetulio	Pencas	Morcillo
985	Cleofás	Tijerilla	<b>NULL</b>
985	Genaro	Tormo	Cansino

Esta opción es adecuada si se sabe que el atributo multivaluado siempre toma un número concreto, finito, de valores mayoritariamente no nulos

## ALTERNATIVA 2: CREAR UNA NUEVA RELACIÓN

Tabla con atributos multivaluados (Actor):

<u>Cod_Peli</u>	Película	Año	Actor
1	La amenaza fantasma	1999	Ewan McGregor Leam Neeson Natalie Portman
2	Blade Runner	1982	Harrison Ford Sean Young
3	Avatar	2009	Sigourney Weaver Zoe Saldana

Normalización



2 Tablas en 1FN:

<u>Cod_Peli</u>	Película	Año
1	La amenaza fantasma	1999
2	Blade Runner	1982
3	Avatar	2009

<u>Cod_Peli</u>	<u>Actor</u>
1	Ewan McGregor
1	Leam Neeson
1	Natalie Portman
2	Harrison Ford
2	Sean Young
3	Sigourney Weaver
3	Zoe Saldana

Esta opción puede ser adecuada cuando el número de valores que puede tomar el atributo multivaluado no se conoce, es muy variable o toma valores nulos

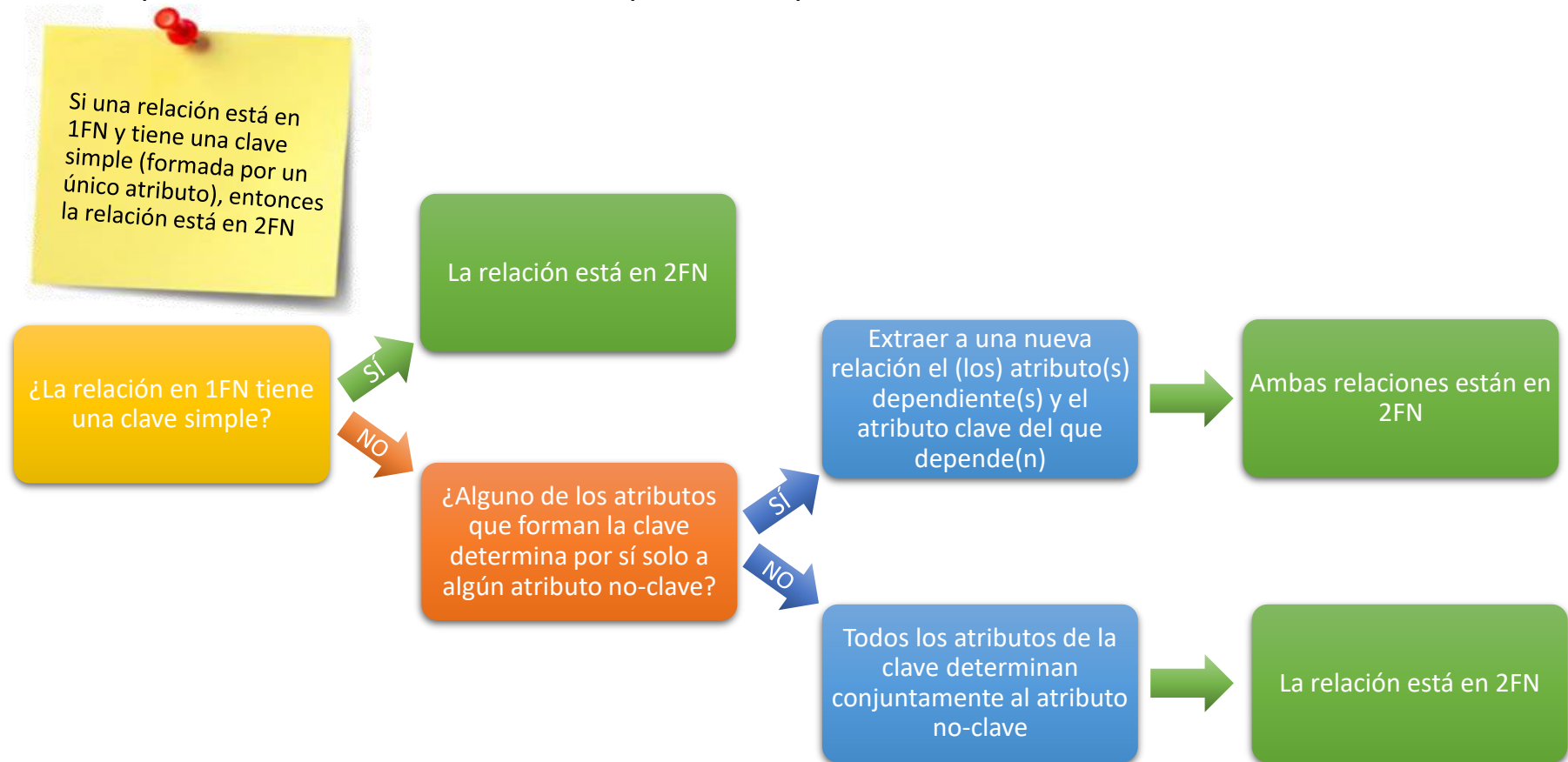
- Una relación se encuentra en Segunda Forma Normal si se cumple lo siguiente:



- La relación está en 1FN.
- Cada atributo no-clave tiene dependencia funcional completa de la clave.

- En caso de no existir dependencia funcional completa respecto a la clave (es decir, en caso de que exista dependencia funcional parcial), es necesario modificar la relación, extrayendo los atributos con dependencia parcial y el atributo del que dependen a una nueva relación, para obtener así un conjunto de relaciones en las que sólo hay dependencia funcional completa respecto a la clave
- La Segunda Forma Normal compartimenta la información impidiendo que una relación represente información de entidades o hechos distintos.

- Si una relación presenta alguna dependencia funcional parcial de la clave, es necesario modificar la relación para extraer los atributos con dependencia parcial a una nueva relación:

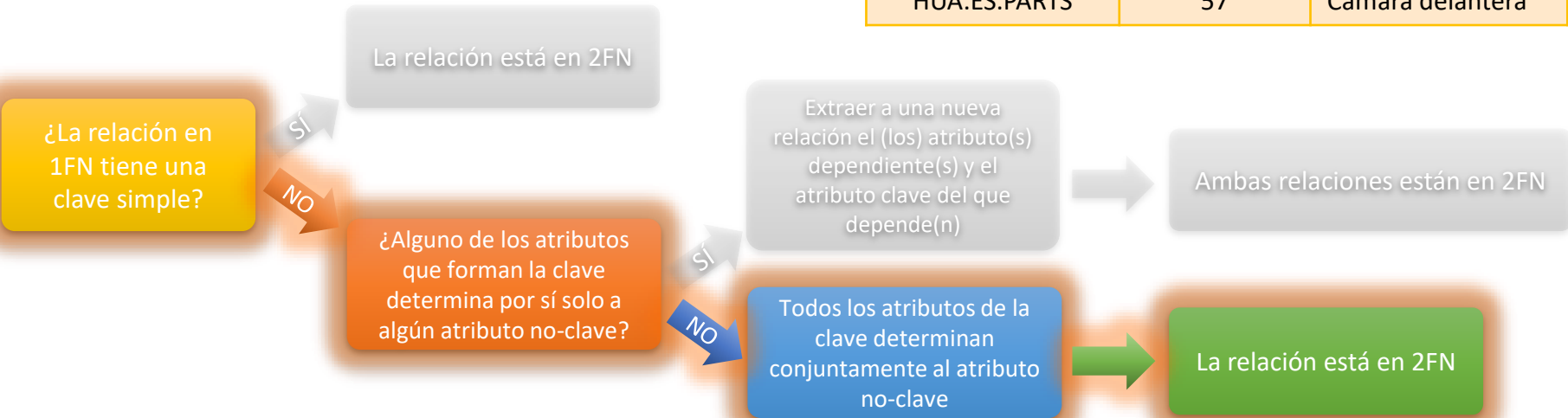


EJEMPLO DE RELACIÓN EN 2FN (DEPENDENCIA FUNCIONAL COMPLETA)

- La siguiente tabla contiene información sobre las piezas de repuesto utilizadas en un servicio técnico de reparación de móviles. Cada pieza tiene un fabricante, identificado por un código único y cada fabricante identifica a sus piezas mediante un número de pieza único para ese fabricante:

Tabla 'Repuestos' en 2FN:

Cód_Fabricante	Cod_Pieza	Descripción
LG_MP	130056	Pantalla capacitiva
LG_MP	130057	Pantalla resistiva
SAM-MOB-PART	0001-34-96	Batería 3000mAh
NOK-MOB-56	57	Módulo WiFi
HUA.ES.PARTS	57	Cámara delantera



EJEMPLO DE RELACIÓN CON DEPENDENCIA FUNCIONAL PARCIAL

- La siguiente tabla contiene información sobre las piezas de repuesto utilizadas en un servicio técnico de reparación de móviles. Cada pieza tiene un fabricante, identificado por un código único y cada fabricante identifica a sus piezas mediante un número de pieza único para ese fabricante. Además, para cada fabricante, se almacena una dirección de email para solicitar más repuestos:

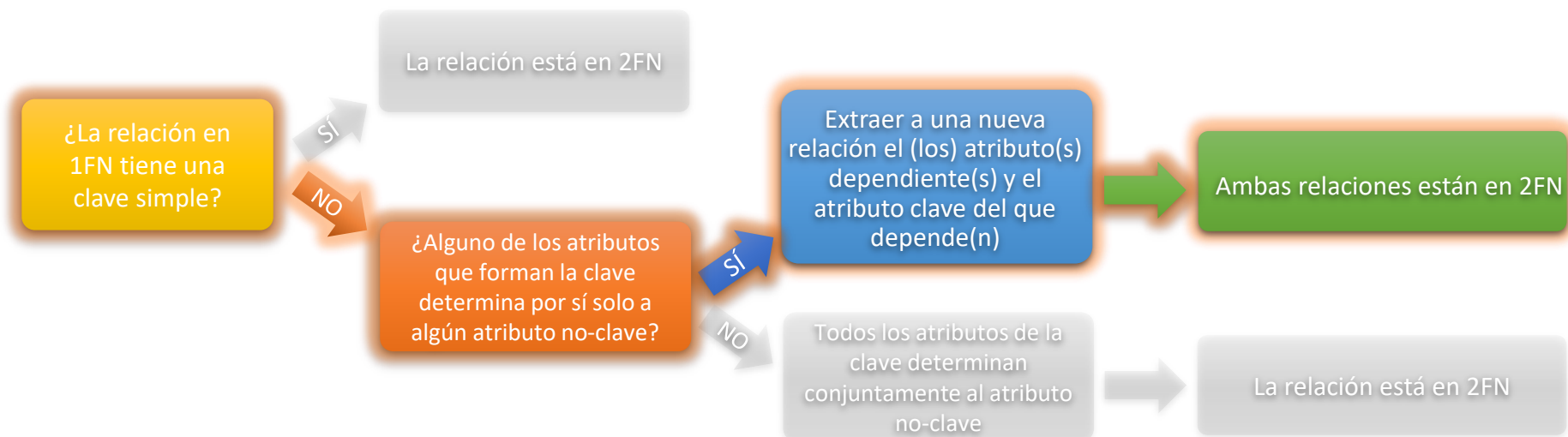
Tabla 'Repuestos' con dependencia funcional parcial:

<u>Cód_Fabricante</u>	<u>Cod_Pieza</u>	Descripción	Email_Solicitud_Repuestos
LG_MP	130056	Pantalla capacitiva	<a href="mailto:lg_mobile_parts@lg.com">lg_mobile_parts@lg.com</a>
LG_MP	130057	Pantalla resistiva	<a href="mailto:lg_mobile_parts@lg.com">lg_mobile_parts@lg.com</a>
SAM-MOB-PART	0001-34-96	Batería 3000mAh	<a href="mailto:samsung.electronics@samsung.com">samsung.electronics@samsung.com</a>
NOK-MOB-56	57	Módulo WiFi	<a href="mailto:nokia.lumia.parts@microsoft.com">nokia.lumia.parts@microsoft.com</a>
HUA.ES.PARTS	57	Cámara delantera	<a href="mailto:huawei_parts@huawei.com">huawei_parts@huawei.com</a>

## EJEMPLO DE RELACIÓN CON DEPENDENCIA FUNCIONAL PARCIAL

Tabla 'Repuestos' con dependencia funcional parcial:

<u>Cód_Fabricante</u>	<u>Cod_Pieza</u>	Descripción	Email_Solicitud_Repuestos
LG_MP	130056	Pantalla capacitiva	<a href="mailto:lg_mobile_parts@lg.com">lg_mobile_parts@lg.com</a>
LG_MP	130057	Pantalla resistiva	<a href="mailto:lg_mobile_parts@lg.com">lg_mobile_parts@lg.com</a>
SAM-MOB-PART	0001-34-96	Batería 3000mAh	<a href="mailto:samsung.electronics@samsung.com">samsung.electronics@samsung.com</a>
NOK-MOB-56	57	Módulo WiFi	<a href="mailto:nokia.lumia.parts@microsoft.com">nokia.lumia.parts@microsoft.com</a>
HUA.ES.PARTS	57	Cámara delantera	<a href="mailto:huawei_parts@huawei.com">huawei_parts@huawei.com</a>



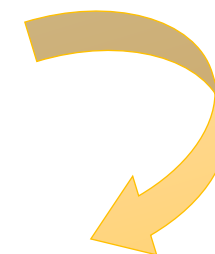
**Cód\_Fabricante** → **Email\_Solicitud\_Repuestos**

## EJEMPLO DE RELACIÓN CON DEPENDENCIA FUNCIONAL PARCIAL

- Para eliminar la dependencia funcional parcial, se extrae a otra tabla el (los) atributo(s) dependiente(s) así como el atributo clave del cual dependen:

Tabla 'Repuestos' con dependencia funcional parcial:

<u>Cód_Fabricante</u>	<u>Cod_Pieza</u>	Descripción	Email_Solicitud_Repuestos
LG_MP	130056	Pantalla capacitiva	<a href="mailto:lg_mobile_parts@lg.com">lg_mobile_parts@lg.com</a>
LG_MP	130057	Pantalla resistiva	<a href="mailto:lg_mobile_parts@lg.com">lg_mobile_parts@lg.com</a>
SAM-MOB-PART	0001-34-96	Batería 3000mAh	<a href="mailto:samsung.electronics@samsung.com">samsung.electronics@samsung.com</a>
NOK-MOB-56	57	Módulo WiFi	<a href="mailto:nokia.lumia.parts@microsoft.com">nokia.lumia.parts@microsoft.com</a>
HUA.ES.PARTS	57	Cámara delantera	<a href="mailto:huawei_parts@huawei.com">huawei_parts@huawei.com</a>



Normalización

Tabla 'Repuestos' en 2FN:

<u>Cód_Fabricante</u>	<u>Cod_Pieza</u>	Descripción
LG_MP	130056	Pantalla capacitiva
LG_MP	130057	Pantalla resistiva
SAM-MOB-PART	0001-34-96	Batería 3000mAh
NOK-MOB-56	57	Módulo WiFi
HUA.ES.PARTS	57	Cámara delantera

Tabla 'Email\_Solicitudes' en 2FN:

<u>Cód_Fabricante</u>	Email_Solicitud_Repuestos
LG_MP	<a href="mailto:lg_mobile_parts@lg.com">lg_mobile_parts@lg.com</a>
LG_MP	<a href="mailto:lg_mobile_parts@lg.com">lg_mobile_parts@lg.com</a>
SAM-MOB-PART	<a href="mailto:samsung.electronics@samsung.com">samsung.electronics@samsung.com</a>
NOK-MOB-56	<a href="mailto:nokia.lumia.parts@microsoft.com">nokia.lumia.parts@microsoft.com</a>
HUA.ES.PARTS	<a href="mailto:huawei_parts@huawei.com">huawei_parts@huawei.com</a>



- Una relación se encuentra en Tercera Forma Normal si se cumple lo siguiente:



- La relación está en 2FN.
- No existen dependencias funcionales transitivas, es decir, no existe dependencia funcional entre atributos no-clave.

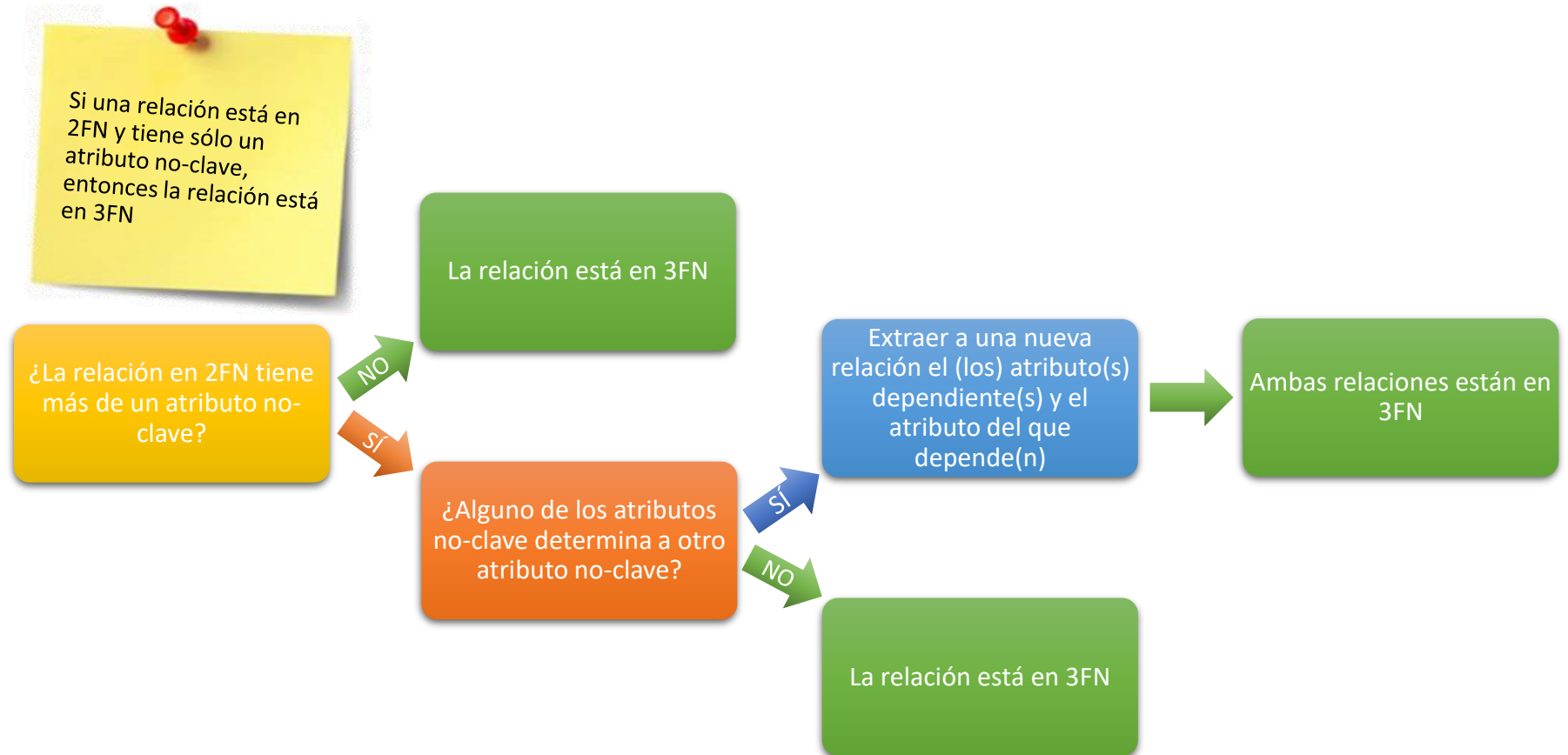
- En caso de existir dependencia funcional transitiva (es decir, en caso de que exista dependencia funcional entre atributos no-clave), es necesario modificar la relación, extrayendo los atributos no-clave dependientes y los atributos determinantes de los que dependen a una nueva relación, para obtener así un conjunto de relaciones en las que no hay dependencia entre atributos no-clave.
- Una relación en Tercera Forma Normal generalmente está libre de anomalías de inserción, modificación y borrado.

# NORMALIZACIÓN

## 5. TERCERA FORMA NORMAL (3FN)

### 5.1. ELIMINACIÓN DE DEPENDENCIAS FUNCIONALES TRANSITIVAS

- Si una relación presenta alguna dependencia funcional transitiva a través de la clave, es necesario modificar la relación para extraer los atributos no-clave dependientes a una nueva relación:



## EJEMPLO DE RELACIÓN CON DEPENDENCIA TRANSITIVA

- La siguiente tabla contiene información sobre los empleados de una empresa: nombre, apellidos y datos del departamento en el que trabajan. A los efectos de este ejemplo se considera que los apellidos y el teléfono son valores atómicos:

Tabla 'Empleados' en 2FN:

<u>Cód_Empleado</u>	Nombre	Apellidos	Departamento	Tlf_Dpto
50001	Balbino	Bechamel Pencas	SIS01	#125
796	Jetulio	Pencas Morcillo	ADMIN	#256
3335	Cleofás	Tijerilla	DES01.02	#113

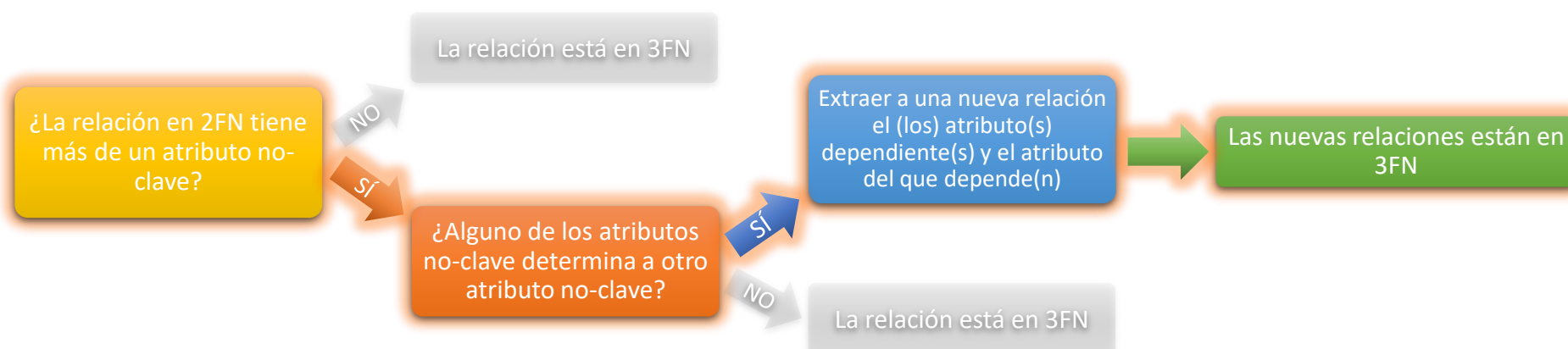


- El atributo no-clave '**Departamento**' determina al atributo no-clave '**Tlf\_Dpto**' y no determina a la clave, '**Cód\_Empleado**'. En esta situación se cumple una dependencia transitiva por la que  $\text{Cod\_Empleado} \longrightarrow \text{Tlf\_Dpto}$ . Esta dependencia viola la 3FN, por lo que es necesario normalizar la relación.

## EJEMPLO DE RELACIÓN CON DEPENDENCIA TRANSITIVA

Tabla 'Empleados' en 2FN:

<u>Cód_Empleado</u>	Nombre	Apellidos	Departamento	Tlf_Dpto
50001	Balbino	Bechamel Pencas	SIS01	#125
796	Jetulio	Pencas Morcillo	ADMIN	#256
3335	Cleofás	Tijerilla	DES01.02	#113



## EJEMPLO DE RELACIÓN CON DEPENDENCIA TRANSITIVA

Tabla 'Empleados' en 2FN:

<u>Cód_Empleado</u>	Nombre	Apellidos	Departamento	Tlf_Dpto
50001	Balbino	Bechamel Pencas	SIS01	#125
796	Jetulio	Pencas Morcillo	ADMIN	#256
3335	Cleofás	Tijerilla	DES01.02	#113

Normalización

Tabla 'Empleados' en 3FN:

<u>Cód_Empleado</u>	Nombre	Apellidos	Departamento
50001	Balbino	Bechamel Pencas	SIS01
796	Jetulio	Pencas Morcillo	ADMIN
3335	Cleofás	Tijerilla	DES01.02

Tabla 'Departamentos' en 3FN:

<u>Departamento</u>	Tlf_Dpto
SIS01	#125
ADMIN	#256
DES01.02	#113

- Una relación se encuentra en Forma Normal de Boyce-Codd si se cumple lo siguiente:



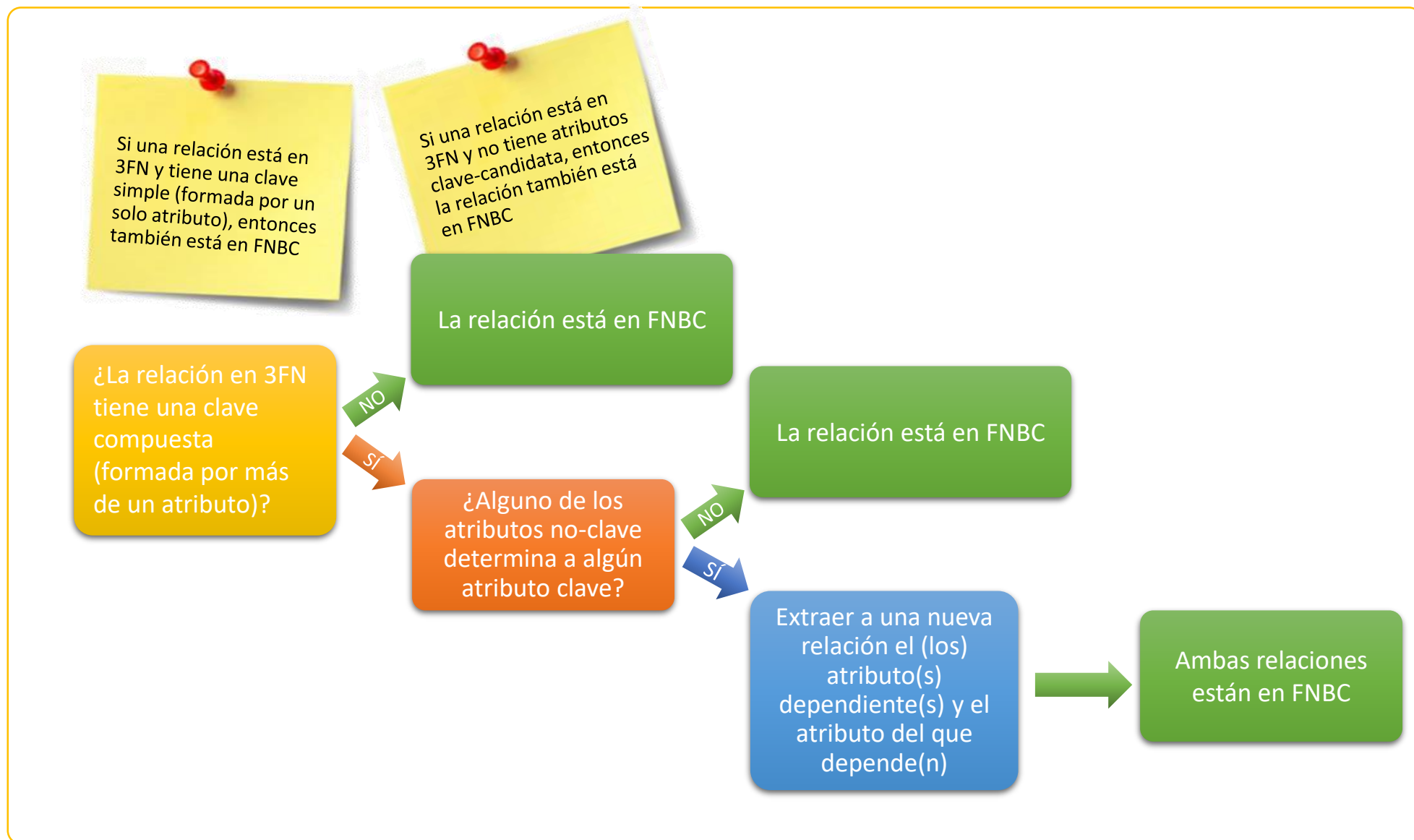
- La relación está en 3FN.
- No existen claves compuestas formadas por atributos que dependan de atributos no-clave. Esto equivale a verificar que no existen claves candidatas que puedan ser intercambiadas por la clave.

- En caso de que algún atributo clave dependa funcionalmente de un atributo no-clave, es necesario modificar la relación, extrayendo el atributo no-clave y el atributo clave del que depende a una nueva relación.
- Una relación en Forma Normal de Boyce-Codd está libre de anomalías de inserción, modificación y borrado y proporciona más compartimentación y consistencia que la 3FN. En la mayoría de casos, una tabla en 3FN también está en FNBC, aunque no se puede generalizar.

# NORMALIZACIÓN

## 6. FORMA NORMAL DE BOYCE-CODD (FNBC)

### 6.1. ELIMINACIÓN DE DEPENDENCIAS FUNCIONALES DE ATRIBUTOS NO-CLAVE RESPECTO A ATRIBUTOS CLAVE



#### EJEMPLO DE RELACIÓN CON DEPENDENCIA DE UN ATRIBUTO CLAVE RESPECTO A UN ATRIBUTO NO-CLAVE

- La siguiente tabla contiene información sobre los proyectos en los que trabajan unos empleados y el servidor en el que se almacenan los informes de los proyectos. A los efectos de este ejemplo se considera que cada proyecto está almacenado en un único servidor y que un servidor almacena sólo un proyecto :

Tabla 'Proyectos' en 3FN:

<u>Cód Empleado</u>	<u>Cod Proyecto</u>	Servidor
50001	DES.WEB.01	192.168.0.7
50001	MANT.SIS.02	192.168.0.25
3335	NETWORKNG.07	192.168.0.134

- En esta situación se cumple la dependencia **Servidor**  $\longrightarrow$  **Cod Proyecto**, o lo que es lo mismo, el atributo no-clave '**Servidor**' se comporta como clave candidata y puede reemplazar al atributo clave '**Cod Proyecto**'. Esta dependencia viola la FNBC, por lo que es necesario normalizar la relación.



#### EJEMPLO DE RELACIÓN CON DEPENDENCIA DE UN ATRIBUTO CLAVE RESPECTO A UN ATRIBUTO NO-CLAVE

Tabla 'Proyectos' en 3FN:

<u>Cód Empleado</u>	<u>Cod Proyecto</u>	Servidor
50001	DES.WEB.01	192.168.0.7
50001	MANT.SIS.02	192.168.0.25
3335	NETWORKNG.07	192.168.0.134

¿La relación en 3FN  
tiene una clave  
compuesta  
(formada por más  
de un atributo)?

La relación está en FNBC



¿Alguno de los  
atributos no-clave  
determina a algún  
atributo clave?

La relación está en FNBC



Extraer a una nueva  
relación el (los)  
atributo(s)  
dependiente(s) y el  
atributo del que  
depende(n)

Ambas relaciones  
están en FNBC



EJEMPLO DE RELACIÓN CON DEPENDENCIA DE UN ATRIBUTO CLAVE RESPECTO A UN ATRIBUTO NO-CLAVE

Tabla 'Proyectos' en 3FN:

<u>Cód Empleado</u>	<u>Cod Proyecto</u>	Servidor
50001	DES.WEB.01	192.168.0.7
50001	MANT.SIS.02	192.168.0.25
3335	NETWORKNG.07	192.168.0.134

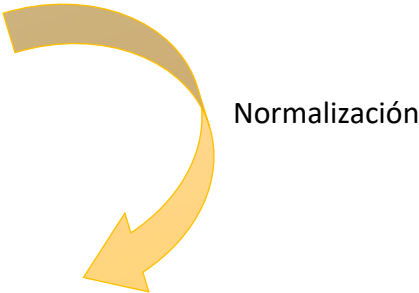


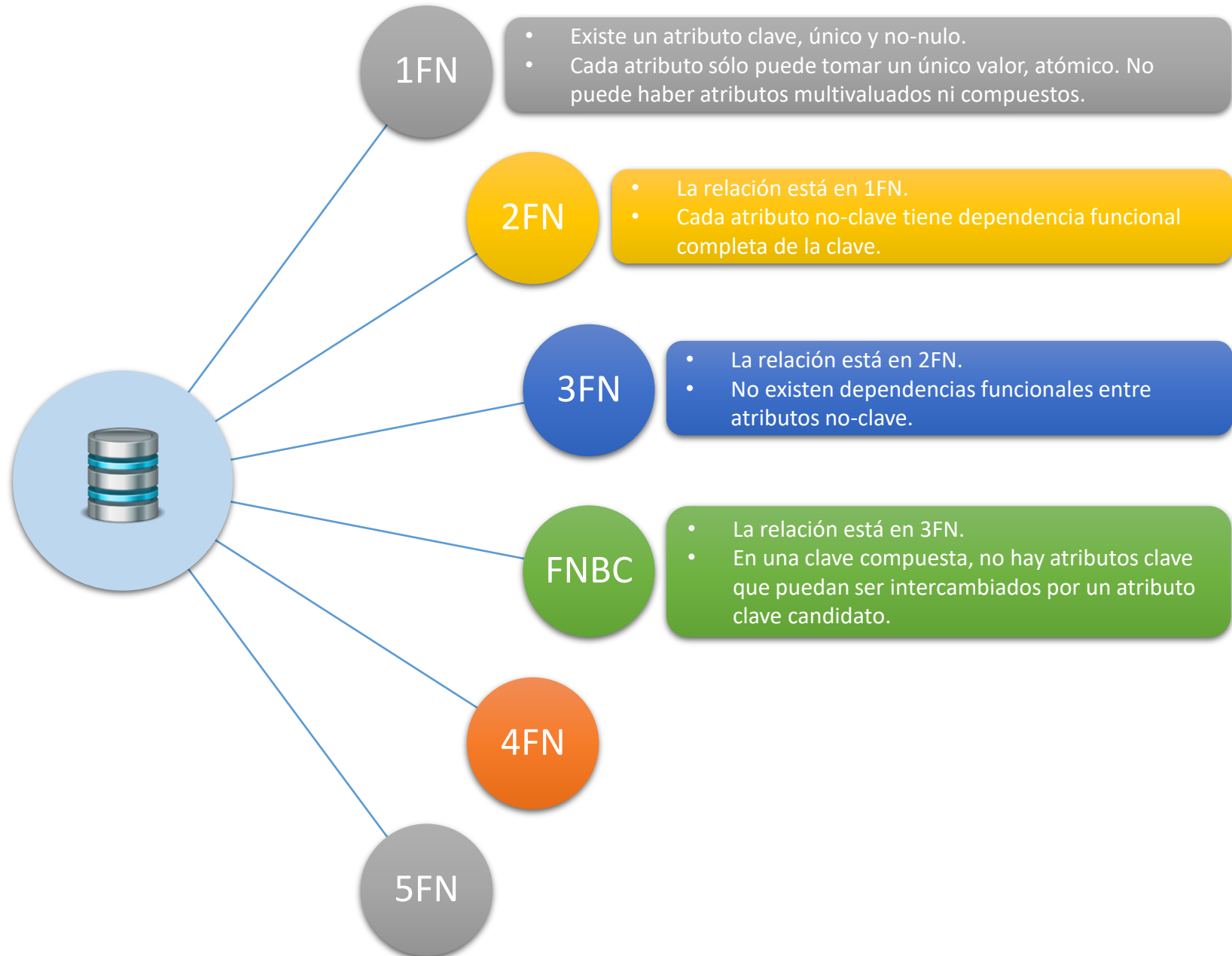
Tabla 'AsignaciónEmpleados' en 3FN:

<u>Cód Empleado</u>	<u>Cod Proyecto</u>
50001	DES.WEB.01
50001	MANT.SIS.02
3335	NETWORKNG.07

Tabla 'AsignaciónServidores' en 3FN:

<u>Servidor</u>	<u>Cod Proyecto</u>
DES.WEB.01	192.168.0.7
MANT.SIS.02	192.168.0.25
NETWORKNG.07	192.168.0.134

## NORMALIZACIÓN



La normalización del modelo relacional es un proceso de refinamiento que mejora las propiedades formales del diseño.

Sin embargo, conforme aumenta la forma normal alcanzada para una relación, aumenta la compartimentación y, por tanto, el número de relaciones creadas.

Un número elevado de tablas dificulta la comprensión del esquema general de la base de datos. Para evitar este problema, pueden utilizarse Vistas, como veremos en la siguiente unidad, que aíslan a las aplicaciones y a los usuarios de la complejidad del diseño.

Habitualmente, una base de datos se mantiene en 3FN o FNBC para garantizar un equilibrio entre el diseño correcto y el rendimiento de las consultas.

Para mejorar el rendimiento puede optarse por reducir el grado de normalización del diseño, introduciendo cierta redundancia controlada, pero manteniendo la integridad y consistencia.

Las vistas abstraen la complejidad del diseño pero aún así, cuando se realizan consultas para acceder a la información almacenada, el rendimiento de la consulta decrece al aumentar el número de tablas implicadas en la consulta.

Esta reducción en el grado de normalización recibe el nombre de '**denormalización**'.