

---

CARRERA DE INGENIERÍA DE SISTEMAS

# INGENIERIA DE DATOS

## Modelo Lógico

INGENIERIA DE DATOS

ÁREA DE INGENIERÍA DE SOFTWARE

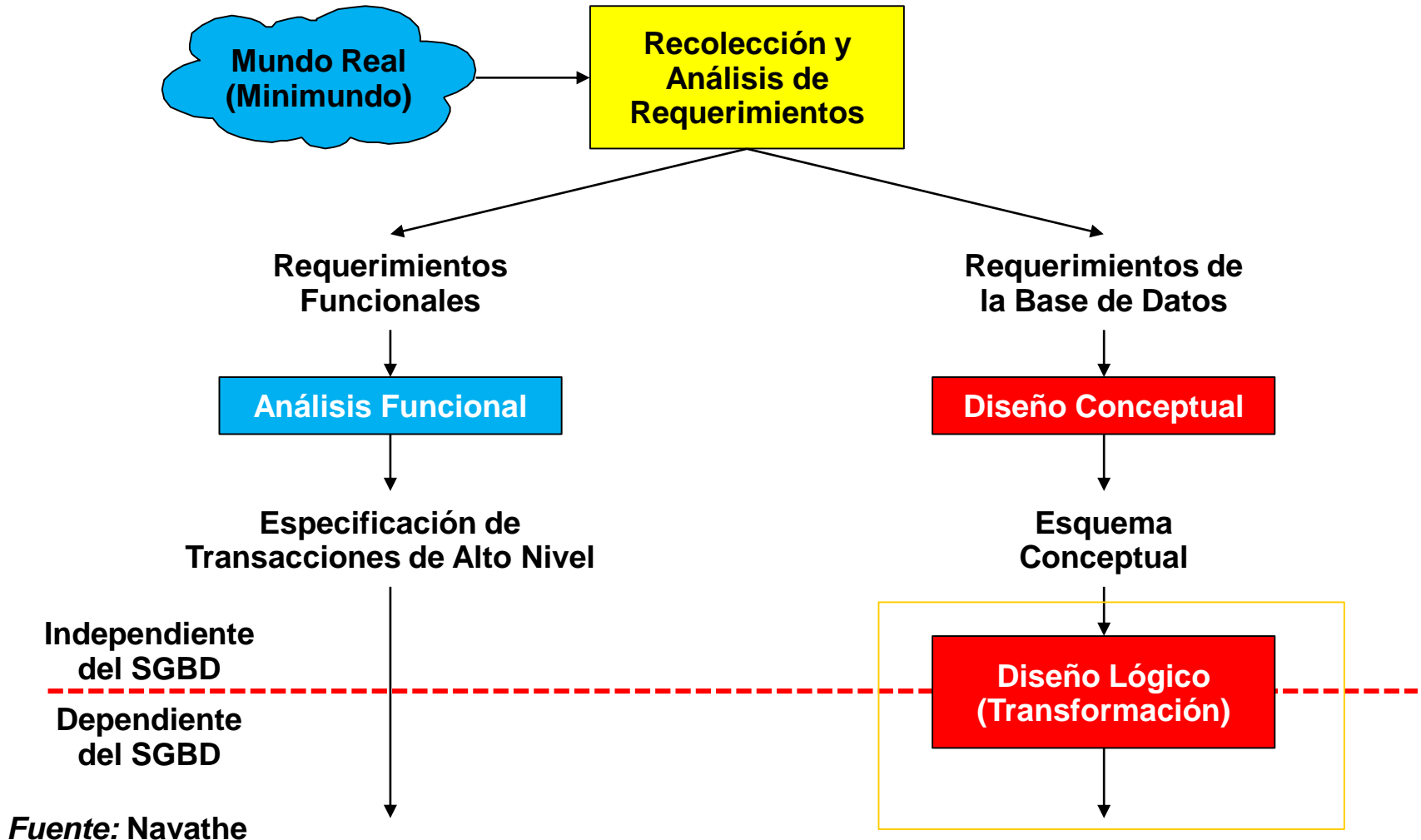
# Agenda

---

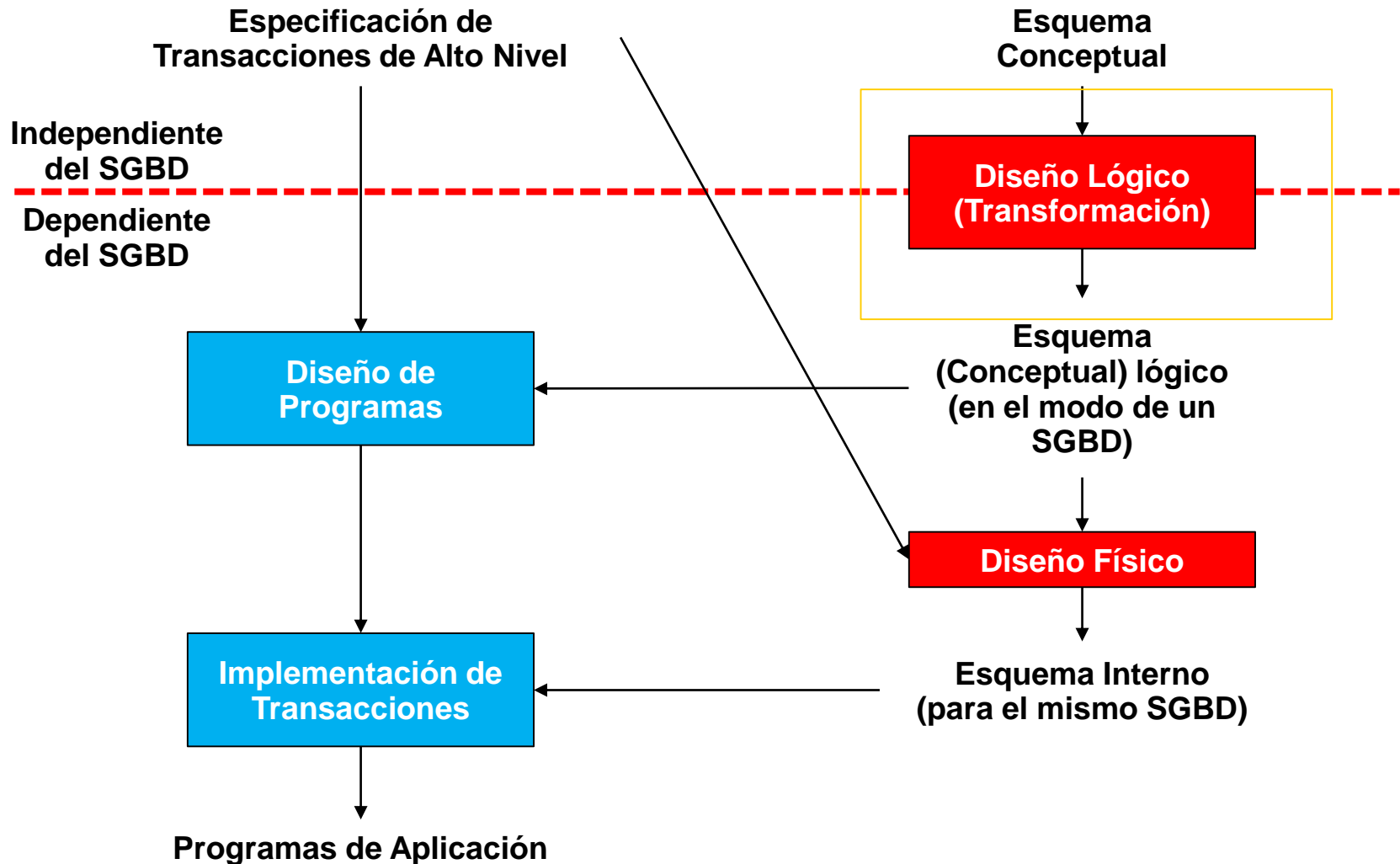
- Enfoque por Descomposición
- Dependencias Funcionales
- Formas Normales
  - Primera Forma Normal
  - Segunda Forma Normal
  - Tercera Forma Normal

# Diseño Lógico

# Proceso de Diseño de una BD



# Proceso de Diseño de una BD



*Fuente: Navathe*

# Diseño Lógico

- Mientras que el objetivo fundamental del diseño conceptual es completitud y la expresividad de los esquemas conceptuales,
- El objetivo del diseño lógico es obtener una **representación** que use, del modo más **eficiente** posible, los recursos que el modelo de SGBD posee **para estructurar los datos y para modelar las restricciones**.

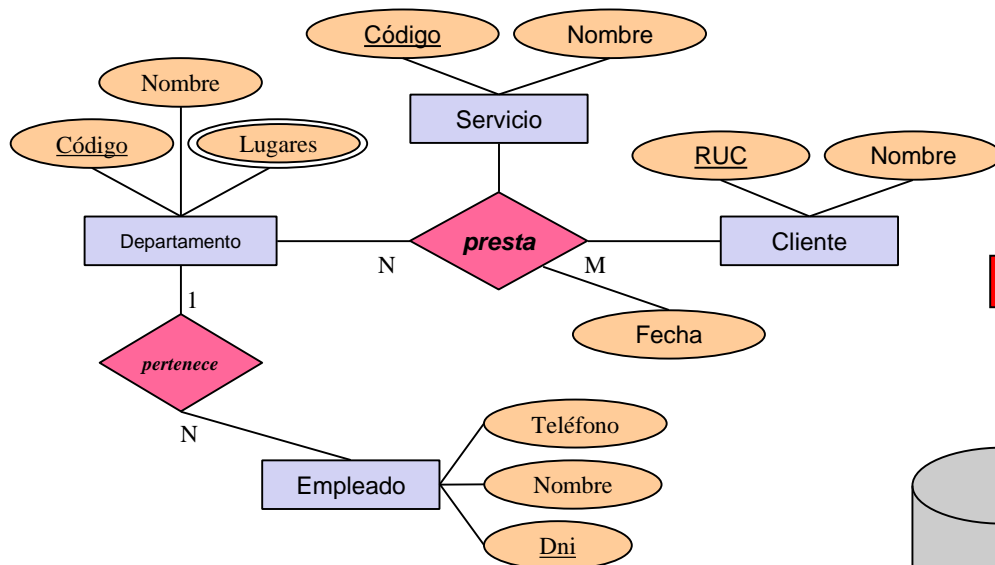
# Diseño Lógico

## Objetivos:

- ***Eliminar redundancias***, conseguir máxima simplicidad, evitar cargas suplementarias de programación.
- ***Conseguir una estructura lógica adecuada***, un equilibrio entre requisitos de usuario y eficiencia de implementación.

# Diseño Descendente

El Diseño Descendente se basa en el uso de los modelos semánticos de datos.



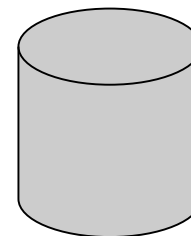
Departamento (Código, Nombre)

Empleado (DNI, Nombre, Teléfono, CodDpto)

Cliente (RUC, Nombre)

Servicio (Código, Nombre)

Presta (CódDpto, CodServ, RUC, Fecha)



Base de Datos  
Relacional



# Diseño Descendente

---

Transformación

**Esquema Conceptual → Esquema Lógico de Datos**

El **objetivo** del diseño lógico es convertir los **esquemas conceptuales** en **un esquema lógico** que se ajuste al modelo de SGBD sobre el que se vaya a implementar el sistema.

Aquí se trata el diseño de bases de datos relacionales, en esta etapa se obtiene un **conjunto de relaciones** (tablas) que representen los datos de interés.

# Enfoque por Descomposición

Relación Universal

Dependencias  
Funcionales

+

R	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
	...	...	...	...	...	...	...	...	...	...	...	...

R1	A1	A2	A3	A4	A5	A6
	...	...	...	...	...	...

R2	A7	A8	A9	A10	A11	A12
	...	...	...	...	...	...

R1.1	A1	A2	A3
	...	...	...

R1.2	A4	A5	A6
	...	...	...

R2.1	A7	A8	A9
	...	...	...

R2.2	A10	A11	A12
	...	...	...

Entidades y Asociaciones del Mundo Real

Enfoque por Descomposición o  
Síntesis Relacional

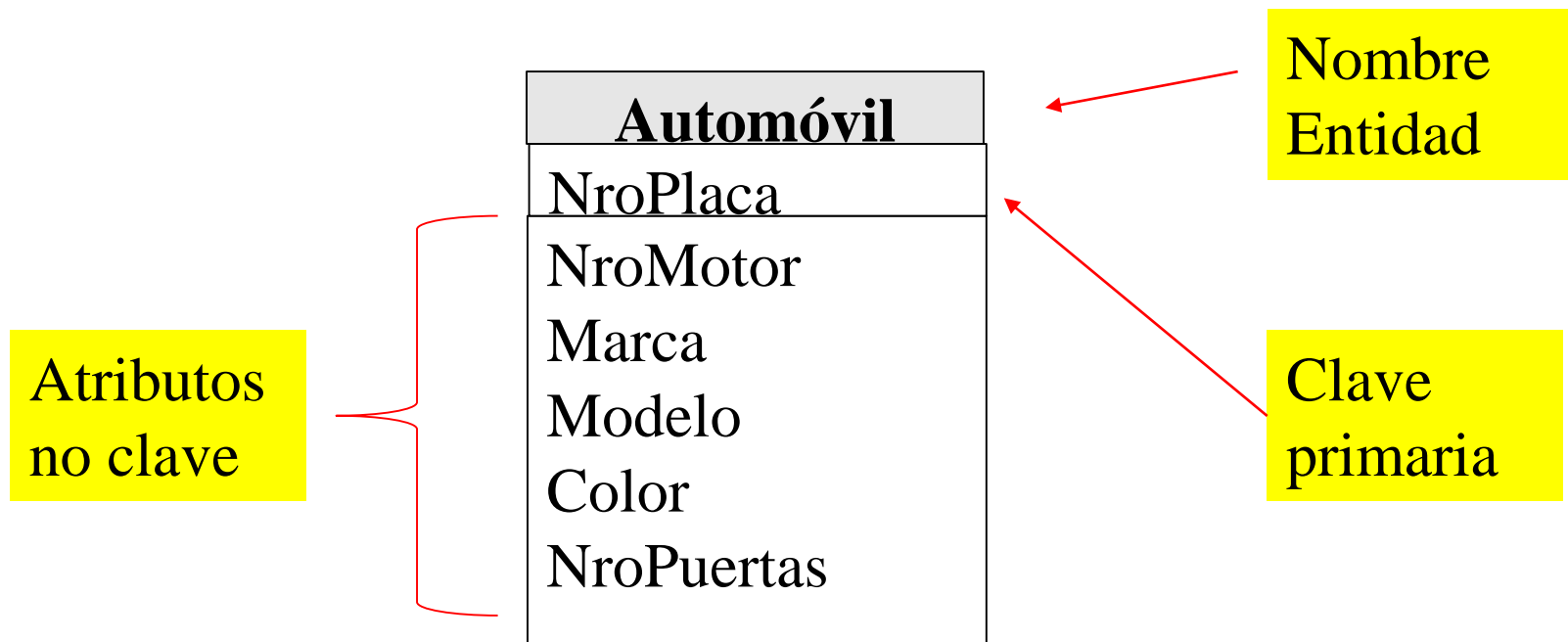
# Enfoque por Descomposición

Definir **relaciones universales** compuestas de todos los atributos de la base de datos y luego descomponerlas, utilizando el proceso de normalización de las relaciones, en subrelaciones que no sufran anomalías

Es un proceso de **refinamiento paso a paso**, que lleva al **aislamiento de las entidades y asociaciones** del mundo real [Codd, 1979]

Se basa las operaciones fundamentales del álgebra relacional (Proyección / Reunión Natural)

# Diagrama Lógico



# Terminología Base Datos Relacional

- Cada tabla tiene filas y columnas.

**Tabla EMPRESA (Relación)**

The diagram illustrates a table with four columns: ID, NAME, PHONE, and CITY\_ID. The first column, ID, is highlighted with a red box and labeled as the Primary Key. The last column, CITY\_ID, is highlighted with a green oval and labeled as the Foreign Key. The first row is highlighted with a yellow box and labeled as a Fila (Tupla). The first column is also labeled as a Columna (Atributo). The table contains the following data:

ID	NAME	PHONE	CITY_ID
201	Unisports	55-2066101	12
202	Simms Atheletics	81-20101	14
203	Delhi Sports	91-10351	14
204	Womansport	1-206-104-0103	11

- Se puede manipular datos de las filas usando comandos SQL.

# PRIMARY KEY

---

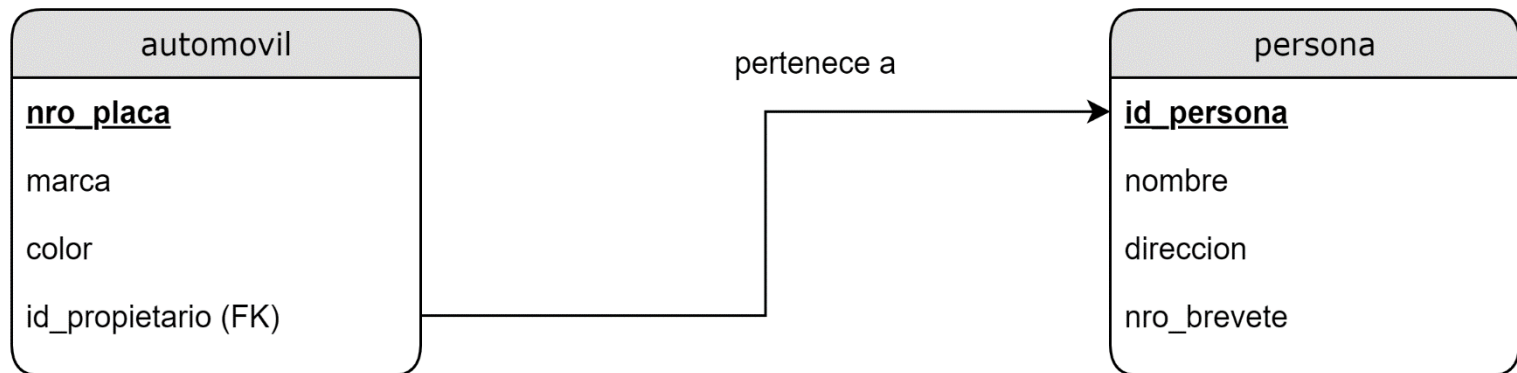
- Sólo se permite la definición de un **Primary Key** por cada tabla
- Garantiza que no hayan filas duplicadas en la tabla
- No se permiten valores nulos
- Automáticamente crea un índice UNIQUE

# FOREIGN KEY

---

- Designa una columna o combinación de columnas como un **Foreign Key**
- Establece una relación entre el **Primary Key** en la misma tabla o con otras tablas
- Debe existir un valor en la tabla Padre a donde hace referencia el **Foreign Key**

# Diagrama Lógico (ERD Plus)



**Una persona podría ser dueña de varios automóviles, de uno solo o de ninguno.**

**Un automóvil es propiedad de una sola persona.**



# Relación Identificada y No Identificada

---

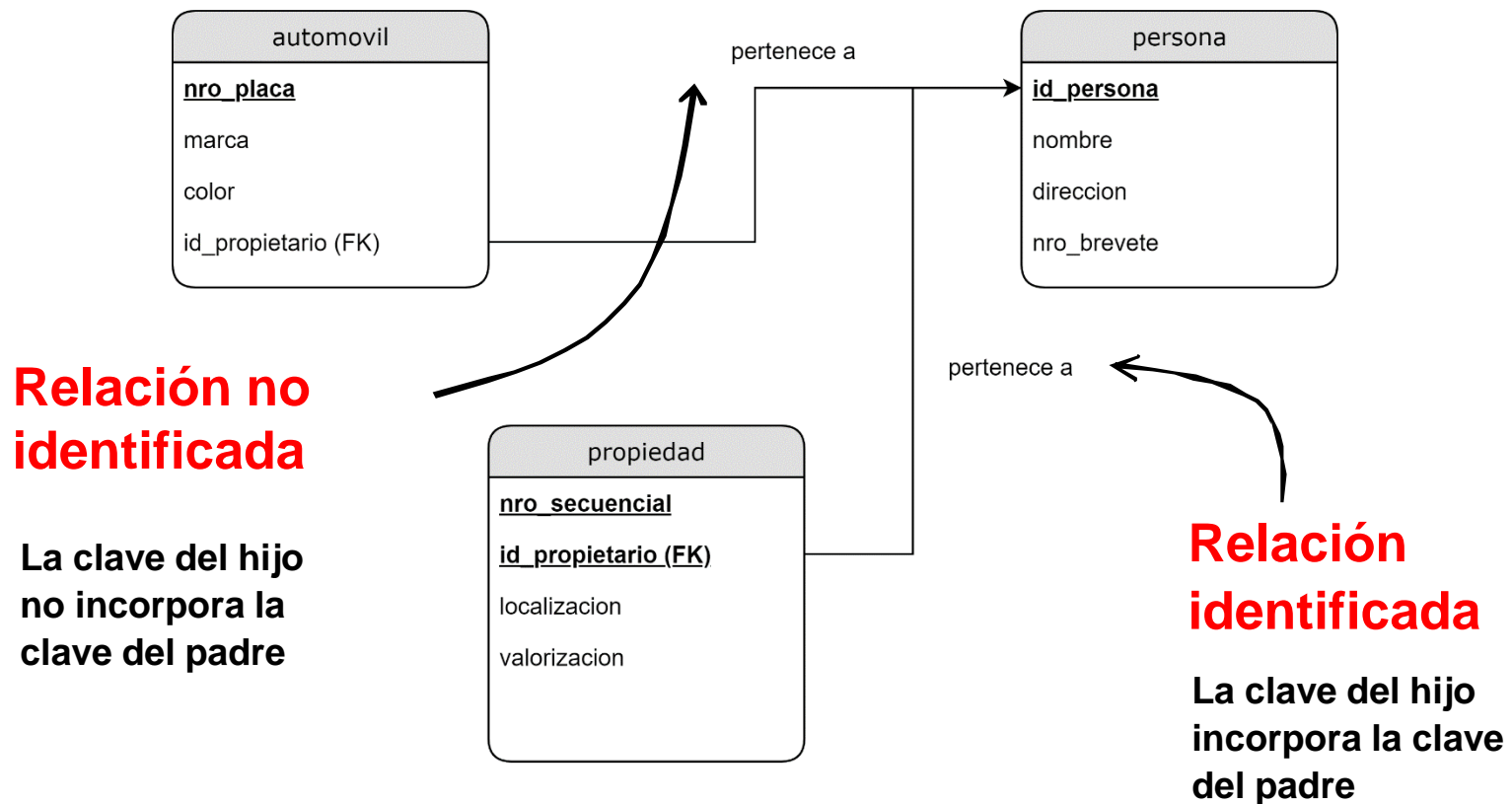
- **RELACION IDENTIFICADA (Obligatoriedad)**

Una relación identificada representa la obligatoriedad de la existencia de una entidad (fuerte) para que se produzca ocurrencia de elementos en otra entidad (débil), se trabajan con las claves para relaciones de ambas entidades. La PK de la entidad padre se agrega a la entidad hijo como FK, dentro de la zona de claves y con **línea de relación continua**.

- **RELACION NO IDENTIFICADA (No obligatoriedad)**

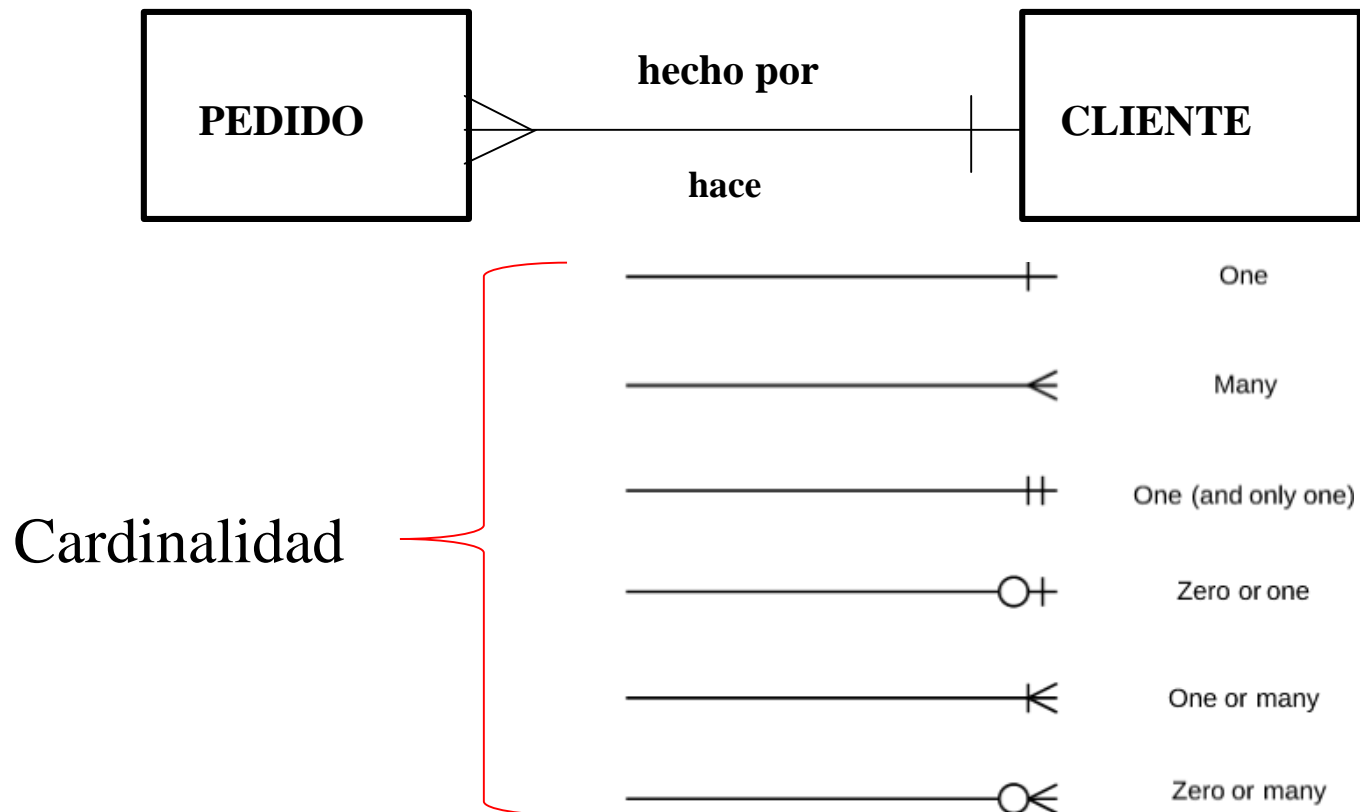
En una relación no identificada no es obligatoria identificar la PK de la entidad padre para identificar algún elemento de la entidad hijo, se **representa** con una **línea discontinua**. La PK de la entidad padre se agrega a la entidad hijo como FK fuera de la zona de claves.

# Relación Identificada y No Identificada

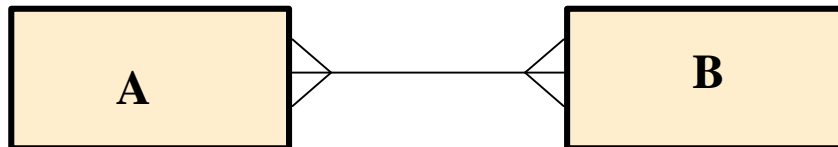


# Notación IE

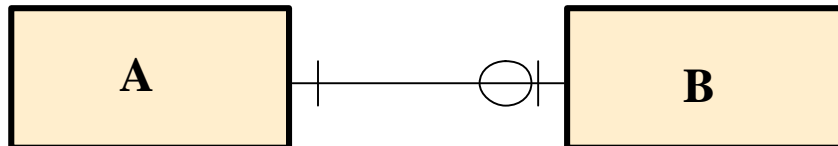
## Information Engineering



# Notación IE



**M : M**  
**Muchos a Muchos**

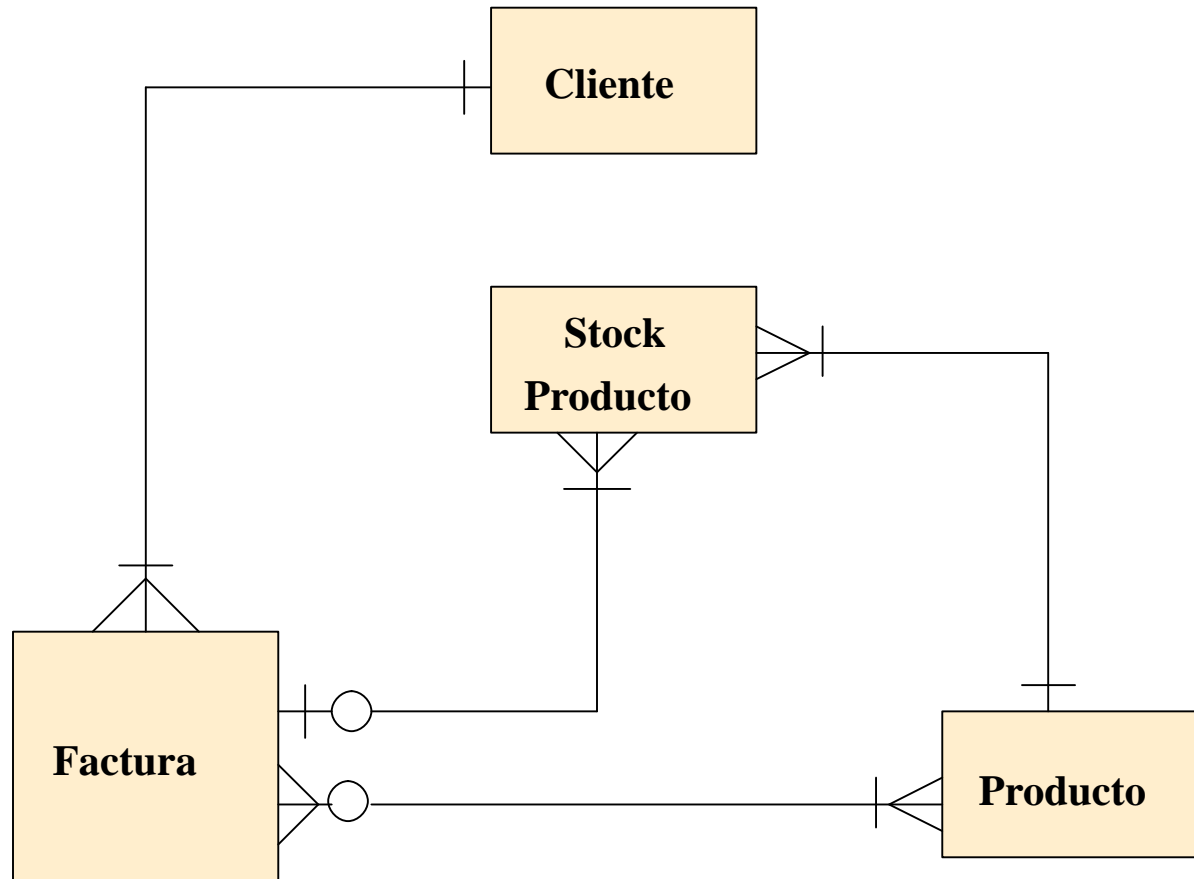


**1 : 0..1**  
**Uno a Cero o Uno**



**1 : M**  
**Uno a Muchos**

# Notación IE

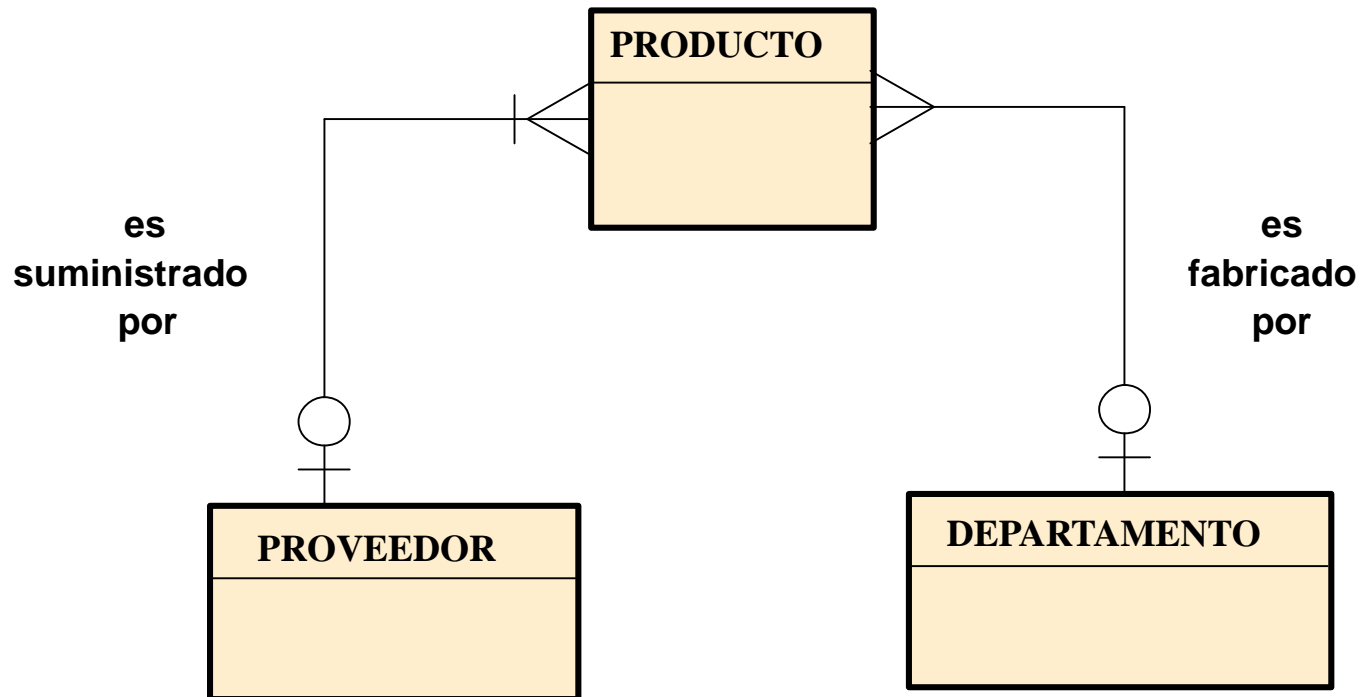


# Relaciones Mutuamente Exclusivas

---

Este tipo de relaciones se presentan si existen relaciones entre una entidad **A** y las entidades **B** y **C**, y la existencia de un apareamiento basado en una de las relaciones excluye la existencia de un apareamiento basado en la otra, se dice

# Relaciones Mutuamente Exclusivas



“ Dado que un producto es suministrado por un proveedor o fabricado por un departamento, pero no por ambos.

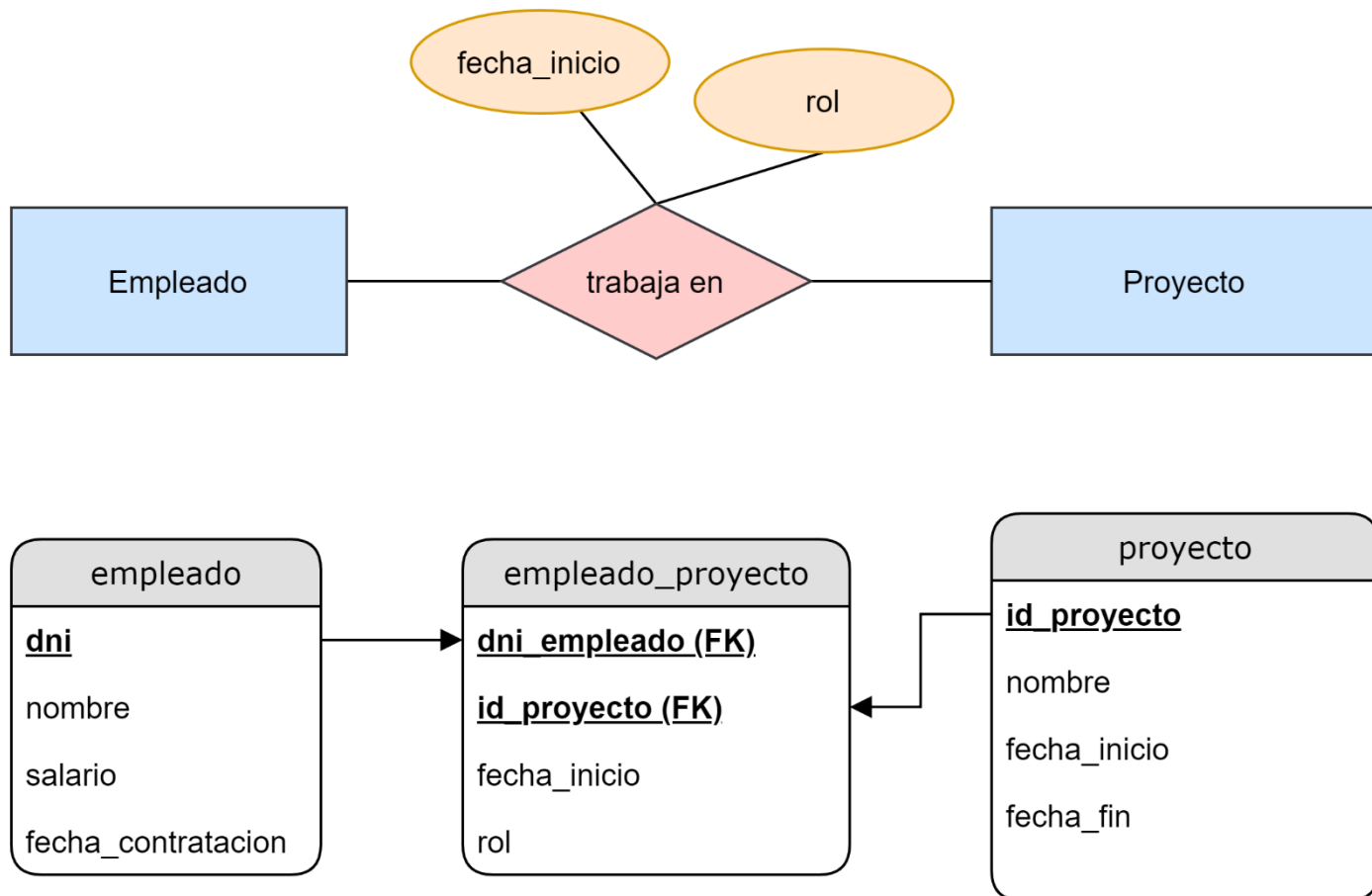
# Resolviendo Relaciones N:M

---

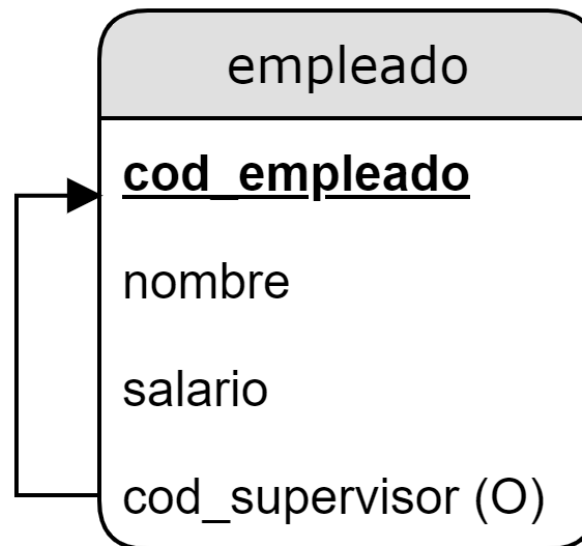
- Este tipo de relaciones no pueden ser directamente implementadas en el modelo relacional. Para resolver esto se introduce el concepto de **entidad de intersección** o **entidad de enlace**.
- La nueva entidad deriva su PK de ambas entidades relacionadas.
- Esta nueva entidad, es conocida con el nombre de entidad de enlace, asociativa o de intersección.



# Resolviendo Relaciones N:M



# Relaciones Reflexivas o Recursivas



**\* Este tipo de relación es siempre opcional en uno de sus extremos.**

# Dependencias Funcionales

# Dependencias Funcionales

- Son ***restricciones de integridad*** que permiten conocer que ***interrelaciones*** existen entre dos o más atributos del mundo real.
- Son propiedades ***inherentes*** al ***contenido semántico*** de los datos, que se han de cumplir para cualquier extensión del esquema de relación.
- Informalmente, Y **depende funcionalmente** de X si ***cada valor*** de X tiene asociado ***siempre el mismo*** valor de Y en una relación R que contiene a X y Y como atributos.

# Dependencia funcional

- Una dependencia funcional es una relación entre atributos de una misma relación (tabla).
- Sea R una relación y sean X e Y subconjuntos(**atributos**) arbitrarios de R. Entonces decimos que **Y es funcionalmente dependiente de X**, en símbolos:

$$X \twoheadrightarrow Y$$

- (se lee **X determina funcionalmente a Y**, o simplemente X flecha Y)
  - Cuando dos tuplas de R concuerdan en su valor de X, también concuerdan en su valor Y.

# Dependencias Funcionales

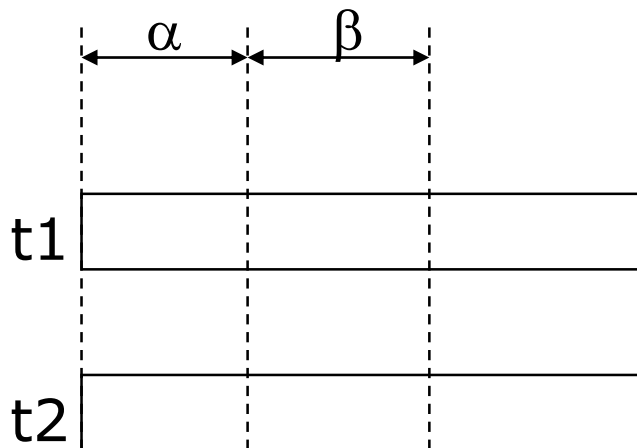
Sean  $a$  y  $b$  atributos de la relación  $R$ .

Decimos que  $a$  **determina funcionalmente a  $b$**  en  $R$ , denotado por  $a \rightarrow b$

**Si y sólo si :**

Para todos los pares de tuplas  $t1$ ,  $t2$  de la relación  $R$ , tales que  $t1[a] = t2[a]$  también se cumple que  $t1[b] = t2[b]$

Ejemplo:  $dni \rightarrow nombre$ .



*Si  $t1$  y  $t2$  coinciden en el atributo  $a$ , Entonces **deben coincidir** también en el atributo  $b$ .*

# Dependencia Funcional

## Ejemplo

- En una relación Empleado si conocemos el valor de FechaDeNacimiento podemos conocer el valor de Edad.

$$\{ \text{fechaDeNacimiento} \} \rightarrow \{ \text{edad} \}$$

- Se puede leer de dos formas:
  - FechaDeNacimiento determina a Edad o
  - Edad es dependiente de FechaDeNacimiento.

# Características de las Dependencias Funcionales

## Dependencia funcional parcial

- Si "Y" es dependiente de "X" y además, es dependiente de un subconjunto propio de "X".
  - Ejemplo:  
 $\{ \text{dni, nombre} \} \rightarrow \{ \text{dirección} \}$ 
    - Si con el dni más el nombre se determina su dirección. Entonces, sólo con el dni también se determina el nombre de una persona, entonces



# Características de las Dependencias Funcionales

## Dependencia funcional completa

- Si "Y" es dependiente de "X" pero no es dependiente de ningún subconjunto propio de "X".
  - Ejemplo:
  - $\{ \text{dni} \} \twoheadrightarrow \{ \text{nombre} \}$ 
    - Si el nombre de una persona están incluidos en el DNI, entonces con el DNI podemos determinar su nombre.

# Características de las Dependencias Funcionales

## Dependencia funcional Transitiva

- Sean X, Y, Z subconjuntos de la misma relación. Si Y depende funcionalmente de X y Z de Y, se dice que Z depende transitivamente de X.

$$X \twoheadrightarrow Y \twoheadrightarrow Z \Rightarrow X \twoheadrightarrow Z$$

Ejemplo:

$\{ \text{fechaDeNacimiento} \} \rightarrow \{ \text{edad} \}$

$\{ \text{edad} \} \rightarrow \{ \text{licenciaConducir} \}$

Entonces:

$\{ \text{fechaDeNacimiento} \} \rightarrow \{ \text{licenciaConducir} \}$

# Normalización

# Normalización

---

El proceso de **normalización** consiste en aplicar una serie de reglas a las relaciones obtenidas en el modelo relacional.

Las bases de datos relacionales se normalizan para:

- **Evitar la redundancia de los datos.**
- **Evitar problemas de actualización de los datos en las tablas.**
- **Proteger la integridad de los datos.**

# Normalización

---

El esquema relacional es un **modelo de la realidad** bajo la forma de una colección de relaciones para que:

- La creación, modificación y supresión de datos de **forma eficaz**. Para esto es indispensable eliminar toda **redundancia innecesaria**. Idealmente, ante la ocurrencia de **un evento** se desea que éste se traduzca en el manejo de **una única tupla** en la extensión del esquema relacional.
- La modificación del esquema relacional por la evolución de la percepción de la realidad, sea lo más simple posible.
- La comprensión de la realidad sea facilitada por el esquema relacional (Ver el esquema -> Comprender la Realidad).

# Normalización

---

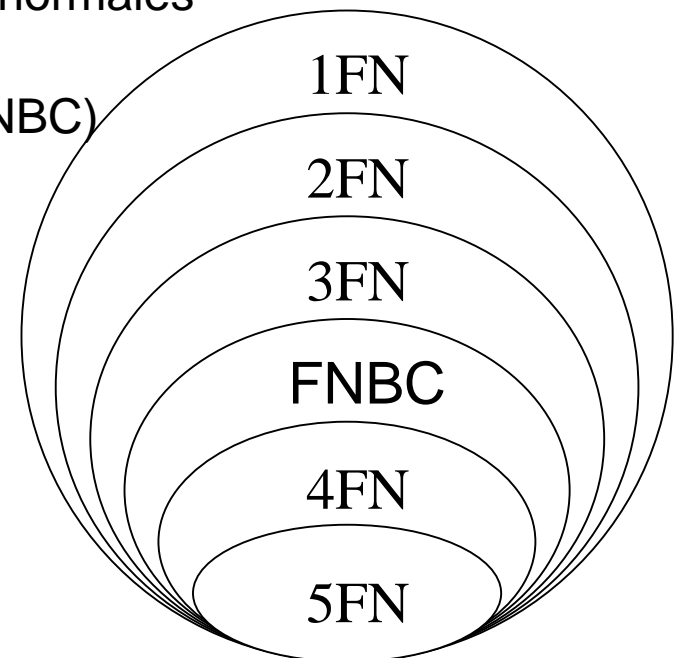
**Normalizar una base de datos es:**

Es encontrar una **descomposición adecuada** de la “relación universal” de la base de datos que nos permite **cumplir con los criterios de** eficacia, ausencia de redundancia, evolución, comprensión, flexibilidad enunciados anteriormente

# Normalización

## Técnica formal para organizar datos basada en formas normales

- Primera, segunda y tercera formas normales  
*Codd 1970*
- Forma normal de Boyce y Codd (FNBC)  
*Boyce – Codd 1974*
- Cuarta forma normal (4FN)  
*Fagin 1977*
- Quinta forma normal (5FN)  
*Fagin 1979*



# Formas normales

---

- El **proceso de normalización** gira en torno al concepto de **formas normales**.
- Se dice que una relación se encuentra en una determinada forma normal si satisface un conjunto de **condiciones establecidas**.
- Toda **relación normalizada** está en 1FN, es decir: **normalizada** y **1FN** significa lo mismo.



# Formas normales

---

- **1NF** (Primera forma normal) - Remover atributos multivalorados.
- **2NF** (Segunda forma normal) - Remover dependencias parciales
- **3NF** (Tercera forma normal) - Remover dependencias transitivas.
- (forma normal Boyce-Codd) - Remover anomalías sobrantes que resultan de los múltiples candidatos a key.
- **4NF** (Cuarta forma normal) - Remover dependencias con multivalores.
- **5NF** (Quinta forma normal) - Remover anomalías que hayan quedado.

**Rara vez utilizamos más de las primeras 3 formas al normalizar**

# Primera, Segunda y Tercera Forma Normal

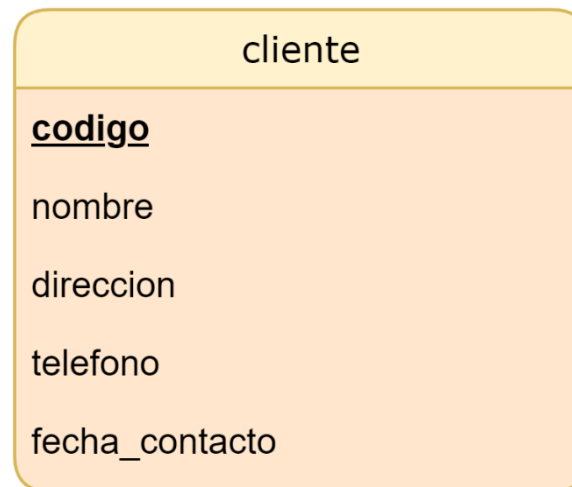
---

## Primera Forma Normal (1NF)

- Una relación está en primera forma normal si y solo si todos los dominios subyacentes contienen solo valores escalares.

# Ejemplo -1NF

¿Cumple la entidad CLIENTE con la 1NF?



¿Todos los atributos cumplen con la condición de no tener múltiples valores?

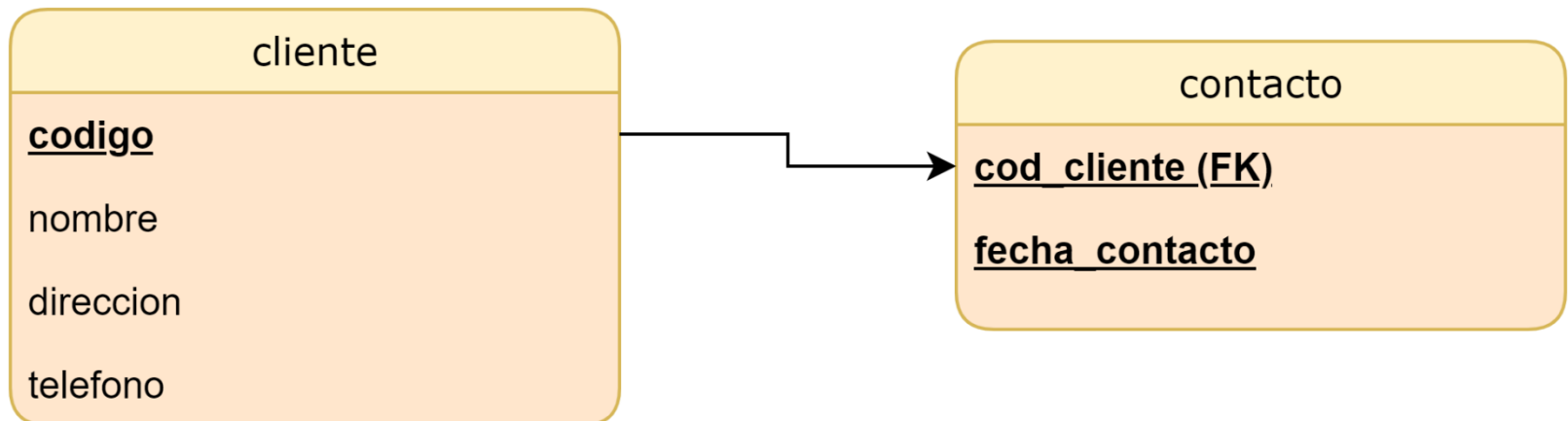
# Ejemplo -1NF

---

- El atributo **fecha\_contacto** puede tener más de un valor ya que uno puede contactar a un cliente en más de una ocasión.
- Por lo tanto el atributo **fecha\_contacto** tiene múltiples valores (aunque sólo puede almacenar la última fecha de contacto)
- **Solución:** Crear una entidad adicional llamada CONTACTO que permita almacenar todos los contactos que se tengan con el cliente.

# Ejemplo -1NF

## SOLUCIÓN



# Normalización - solución

---

## 1NF

**SOLUCIÓN:** Si un atributo tiene múltiples valores, cree una entidad adicional y relaciónela con la entidad original usando una relación M:1

# Primera, Segunda y Tercera Forma Normal

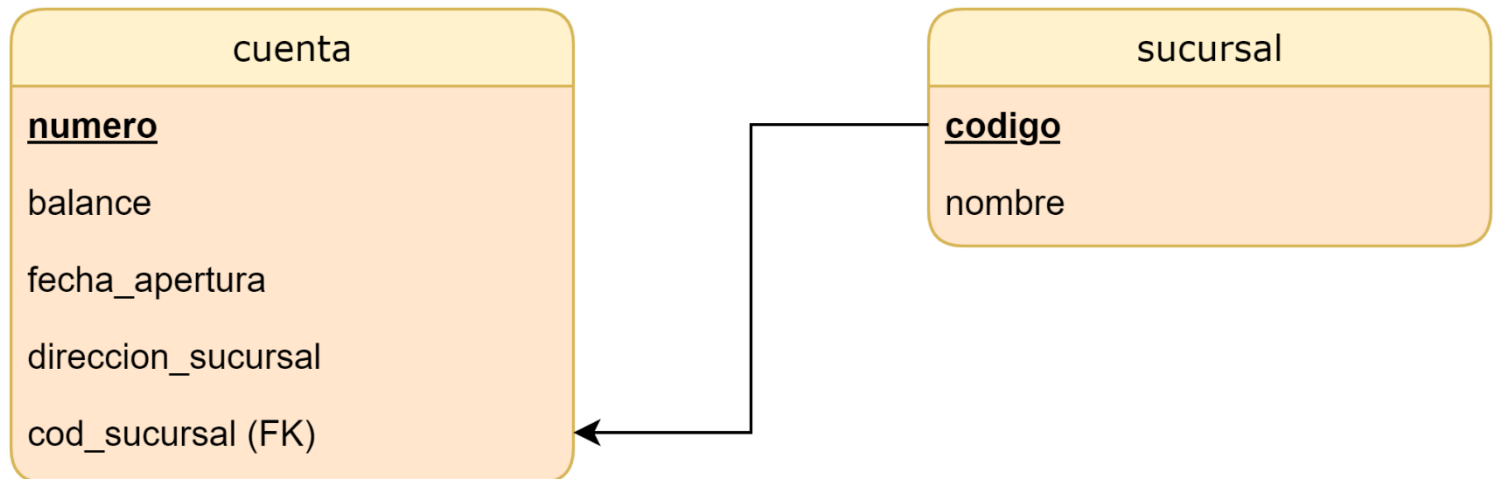
---

## Segunda Forma Normal (2NF)

Una relación está en segunda forma normal (2NF) si y sólo si está en 1NF y todos los atributos no clave dependen por completo de la clave primaria.

# Ejemplo -2NF

¿Cumple esta relación con la 2NF?





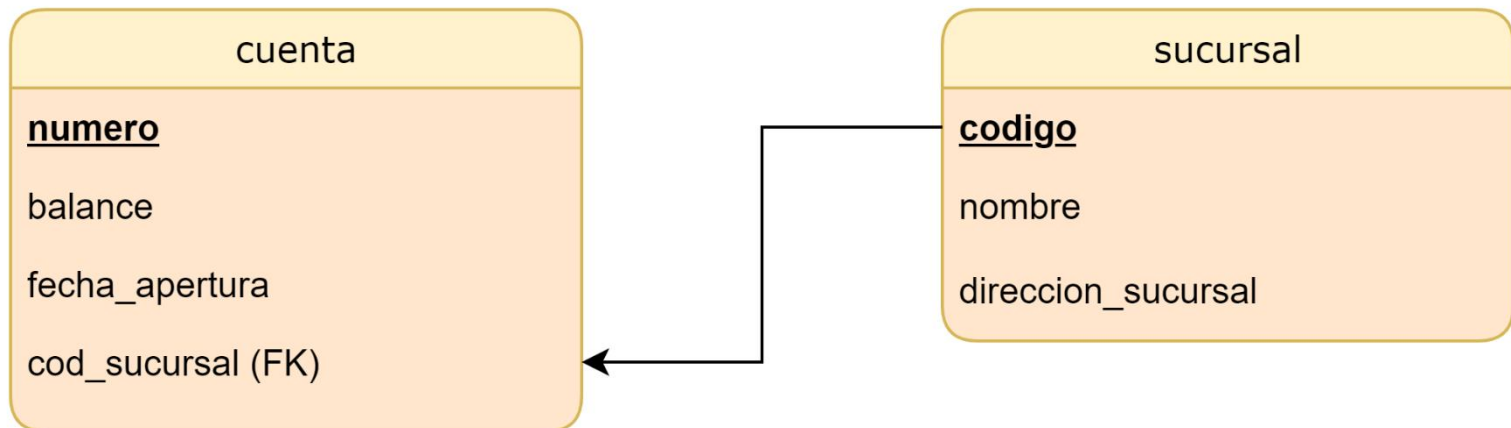
## Ejemplo -2NF

---

- Debe cotejar cada atributo y ver si tiene una relación directa con el *primary key* . Haga este cotejo en ambas entidades.
- En el caso del atributo dirección sucursal, nos percatamos de que pertenece a la entidad SUCURSAL y no a la entidad CUENTA

# Ejemplo -2NF

## SOLUCIÓN



**Pasar el atributo fecha apertura a la entidad SUCURSAL.**

# Normalización - solución

---

## 2NF

**SOLUCIÓN:** Si un atributo no depende completamente de la PK de la entidad, está mal localizado y debe mudarse a otra entidad.

# Primera, Segunda y Tercera Forma Normal

---

## Tercera Forma Normal (3NF)

Una relación está en tercera forma normal (3NF) si y sólo si está en 2NF y todos los atributos no clave dependen de manera no transitiva de la clave primaria.

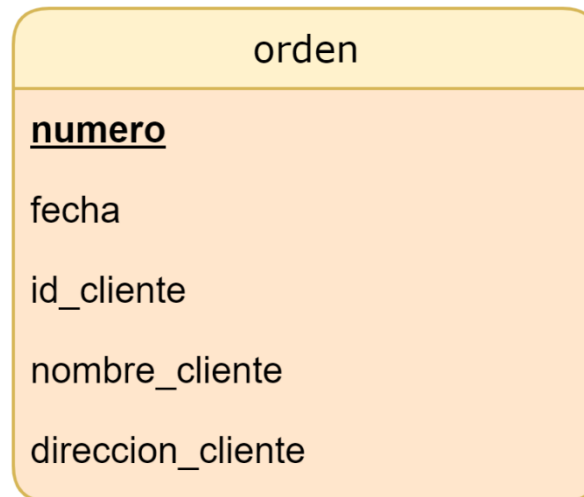
# Tercera Forma Normal (3NF)

---

- Debe estar en 2NF y además ***no debe tener dependencias transitivas*** (dependencias funcionales o atributos que no son *primary-key*)
- **Nota:** Se llama transitivo debido a que la *primary key* es un determinante para otro atributo, el cual a su vez es un determinante de un tercero.
- **Solución:** un determinante *Non-key* con dependencias transitivas va a una nueva tabla. Los determinantes *non-keys* vienen a ser *primary key* en la nueva tabla y se quedan con *foreign key* en la vieja tabla.

## Ejemplo - 3NF

¿Hay algún atributo que no sea PK dependiendo de otro que no pueda servir como PK alternativo en la entidad ORDEN?



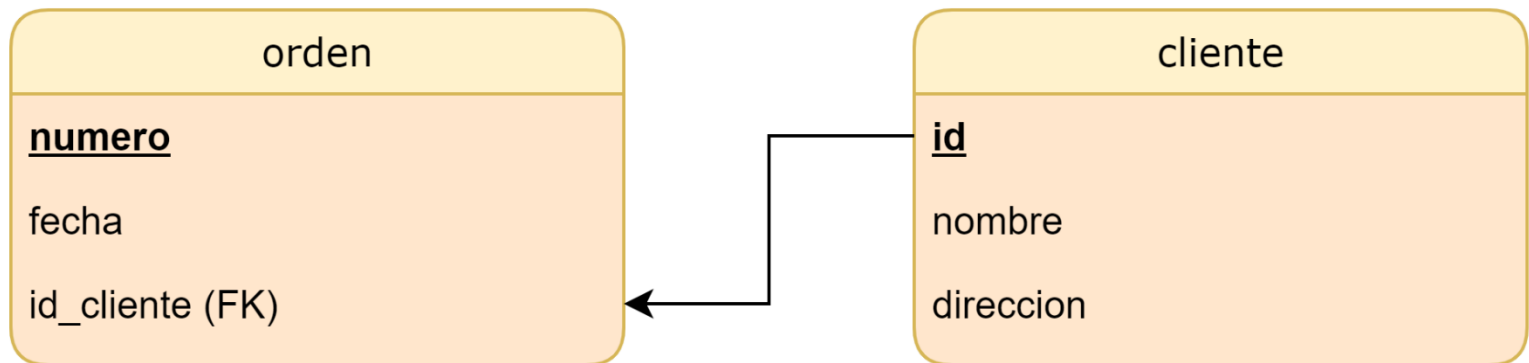
## Ejemplo - 3NF

---

- Los atributos **nombre\_cliente** y **dirección\_cliente** dependen del atributo **id\_cliente**.
- **id\_cliente** no es un PK alternativo de la entidad ORDEN
- Por lo tanto, se debe crear una entidad aparte llamada CLIENTE en la cual se ubican los atributos relacionados al cliente.

# Ejemplo - 3NF

## SOLUCIÓN



**Pasar los atributos del cliente a la entidad CLIENTE.**



# Normalización - solución

---

## 3NF

**SOLUCIÓN:** Si un atributo no depende por completo de la PK de la entidad, está mal localizado y debe mudarse a otra entidad.

# Forma Normal

## Boyce&Codd - BCNF

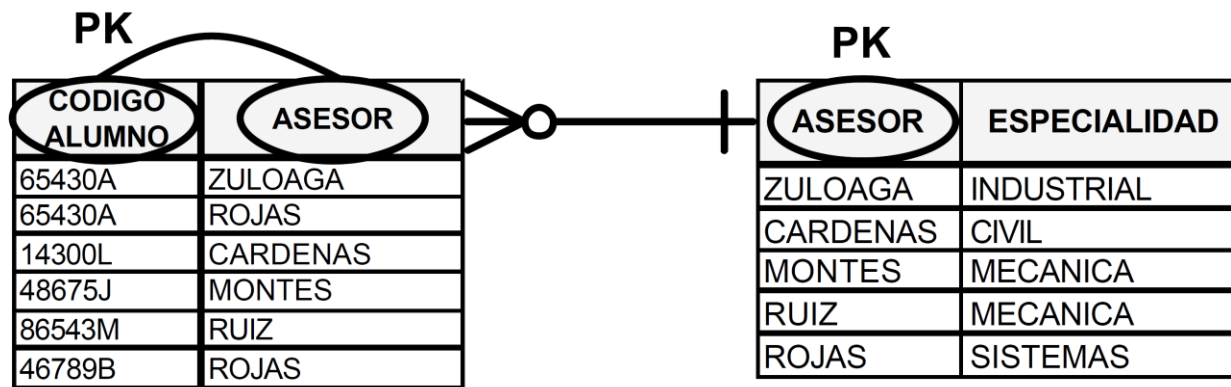
---

- Cuando una relación tiene mas de una clave candidata, se pueden presentar anomalías aun cuando la relación este en 3FN.
- R.F. Boyce y E.F. Codd identificaron esta deficiencia y propusieron una extraña definición de 3FN que soluciona el problema, introduciendo el concepto de determinante.
- Un determinante es cualquier atributo (simple o compuesto) sobre el cual algún otro atributo es funcionalmente dependiente (total).

# Forma Normal Boyce&Codd - BCNF

CODIGO ALUMNO	ESPECIALIDAD	ASESOR
65430A	INDUSTRIAL	ZULOAGA
65430A	SISTEMAS	ROJAS
14300L	CIVIL	CARDENAS
48675J	MECANICA	MONTES
86543M	MECANICA	RUIZ
46789B	SISTEMAS	ROJAS

- La relación esta en 3FN.
- Cada alumno puede especializarse en varias materias.
- Por cada especialidad un alumno sólo puede tener un asesor
- Cada especialidad tiene varios asesores
- Cada asesor asesora sólo una especialidad.
- Existen dos claves candidatas (codalum+espec) y (codalum+asesor).



- Con esto se eliminan las posibles anomalías de inserción, eliminación y actualización.

# Cuarta Forma Normal – 4FN

---

- Una relación esta en 4FN si esta en la BCFN y no contiene dependencias multivaluadas.
- Existe una dependencia multivaluada cuando hay tres atributos (A,B y C) en una relación, tal que por cada valor de A existe un bien definido conjunto de valores de B y un bien definido conjunto de valores de C, sin embargo el conjunto de valores de B es independiente del conjunto C y viceversa

# Cuarta Forma Normal – 4FN

<u>COMPUTADORA</u>	<u>PAQUETE SOFTWARE</u>	<u>TIENDA</u>
APPLE	WRITER	PCSHOP
APPLE	FOX	PCSHOP
APPLE	WRITER	DIGISHOP
IBM	WORD	CIBERSTORE
NCR	LOTUS	DIGISHOP
NCR	WORDPERFECT	DIGISHOP
NCR	LOTUS	CIBERSTORE

- La relación esta en la BCFN.
- La clave primaria de la relación se expresa en función de los tres atributos.
- Por cada computadora existe un conjunto de paquetes y un conjunto de tiendas que las venden.
- Los paquetes y las tiendas son independientes.

<u>COMPUTADORA</u>	<u>PAQUETE SOFTWARE</u>
APPLE	WRITER
APPLE	FOX
IBM	WORD
NCR	WORDPERFECT
NCR	LOTUS

<u>COMPUTADORA</u>	<u>TIENDA</u>
APPLE	PCSHOP
APPLE	DIGISHOP
IBM	CIBERSTORE
NCR	DIGISHOP
NCR	CIBERSTORE

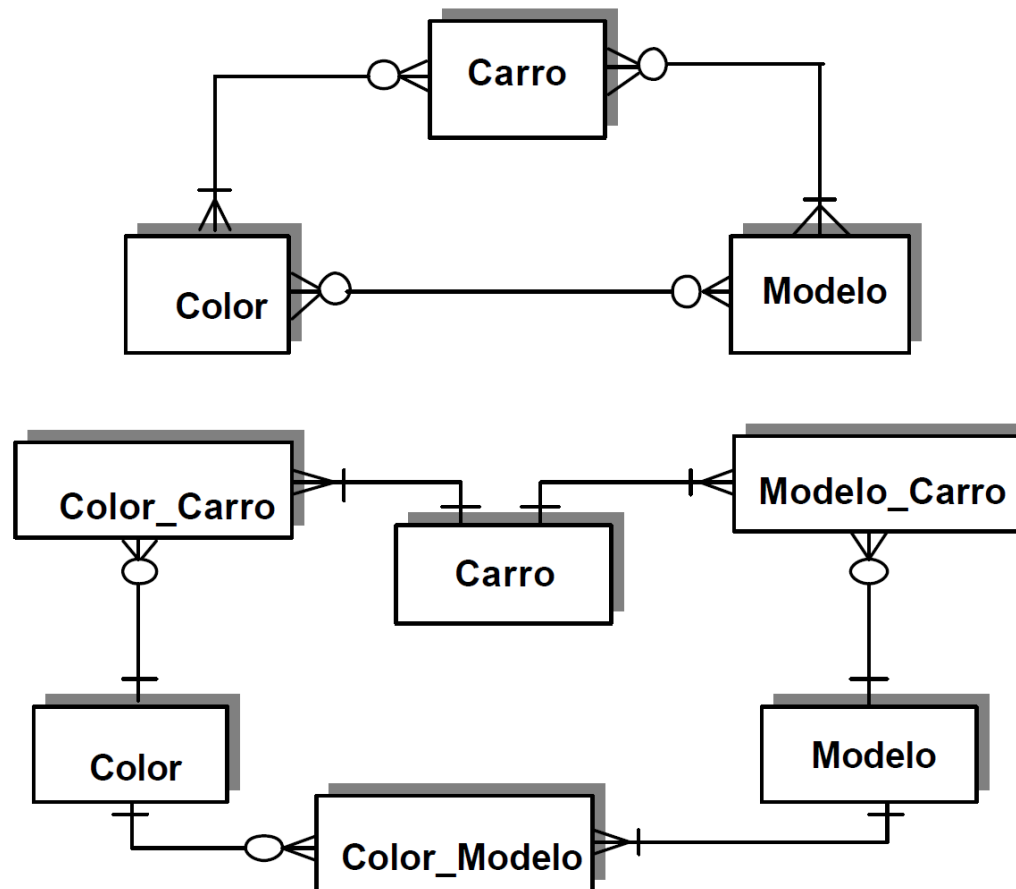
- Para eliminar las anomalías dividimos la relación en dos entidades.

# Quinta Forma Normal – 5FN

---

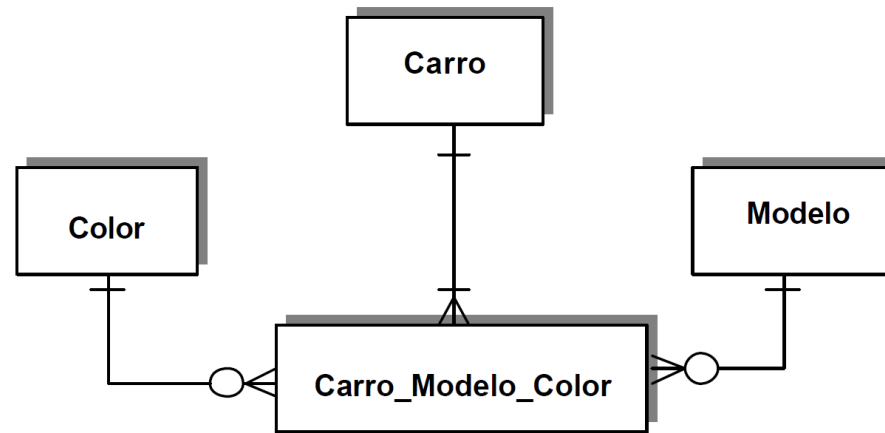
- Permite hacer frente a un tipo de dependencia denominada dependencia de unión (Join dependency).
- Suele presentarse cuando resolvemos tres (o mas) entidades, todas relacionadas con una relación muchos-a-muchos a las otras.
- Es algunas veces referida como Join-Projection Normal Form (JPNF).
- Estas relaciones son raras en la práctica.

# Quinta Forma Normal – 5FN



**Solución con Entidades Asociativas**

# Quinta Forma Normal – 5FN

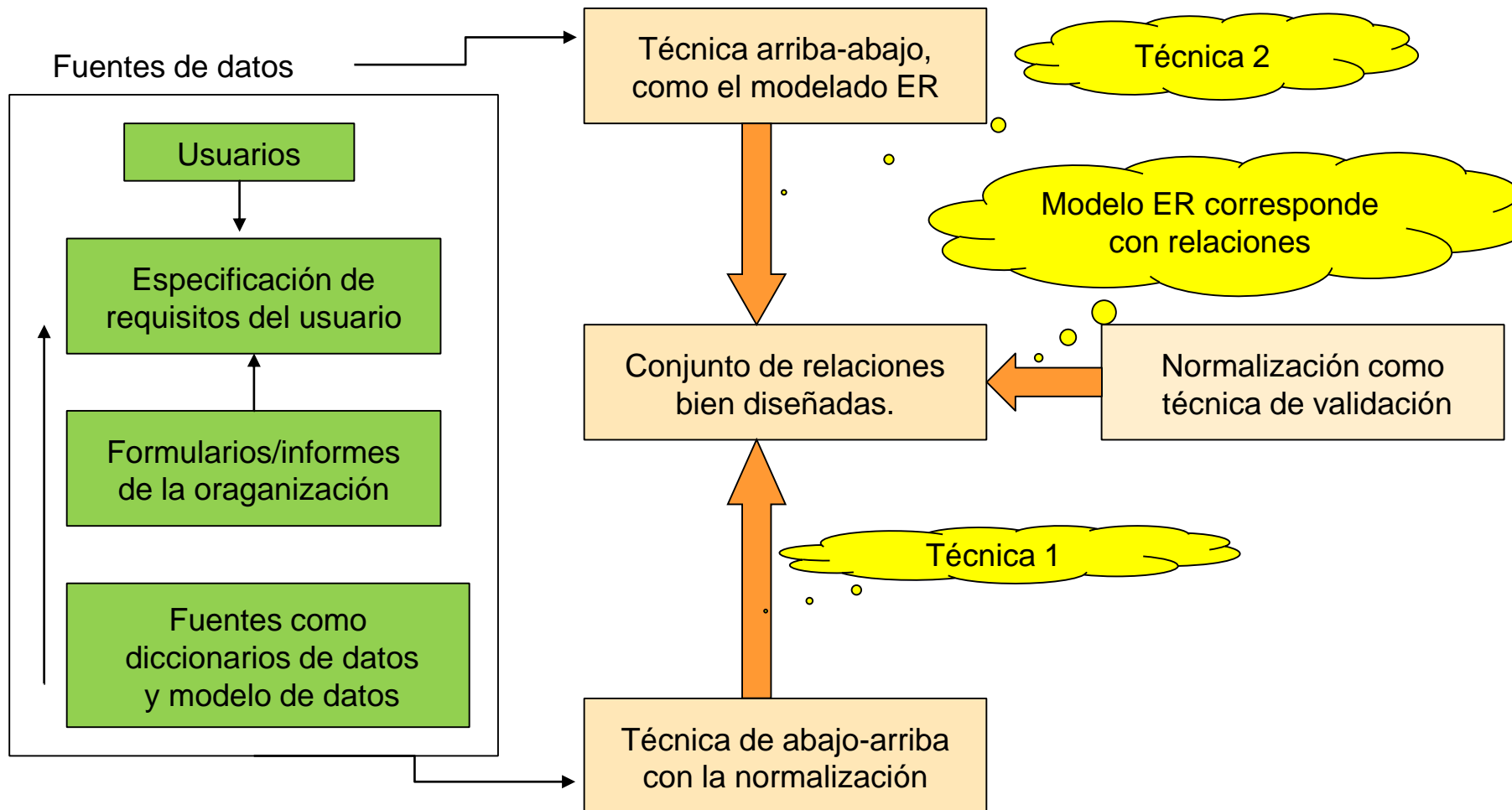


## **Solución a través de la Tabla Asociativa - 5NF**

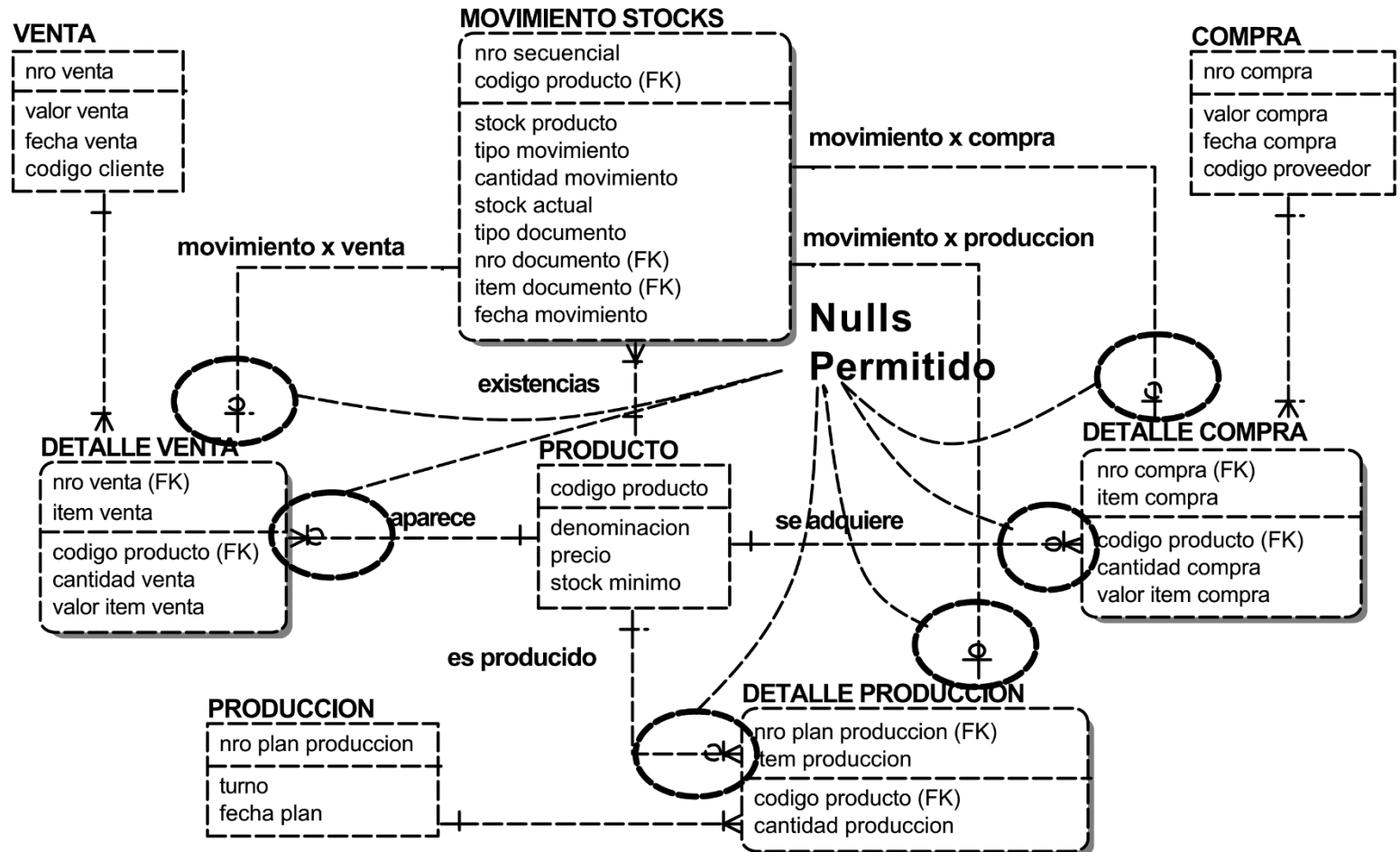
- La solución correcta a este problema es mucho mas simple, hay que introducir una simple entidad asociativa enlazando las otras tres, a la que podemos denominar Carro\_Color\_Modelo.



# Uso de la normalización



# Esquema Normalizado- Ejemplo



# Resumen

---

- La normalización es el proceso de **reorganizar los datos** de acuerdo a las formas normales para establecer un modelo de datos relacional.
- La normalización garantiza el **acceso** en forma sencilla a la información y poder realizar el **mantenimiento** eficientemente.