

# Introducción

- El diseño de una base de datos se realiza en tres fases:
  - Diseño conceptual: Representar la información con independencia de usuarios y aplicaciones.
  - Diseño lógico: Transformar el diseño conceptual al modelo de datos del SGBD.
  - Diseño físico: Implementar de forma eficiente el diseño lógico. Es completamente dependiente del SGBD y el ordenador.

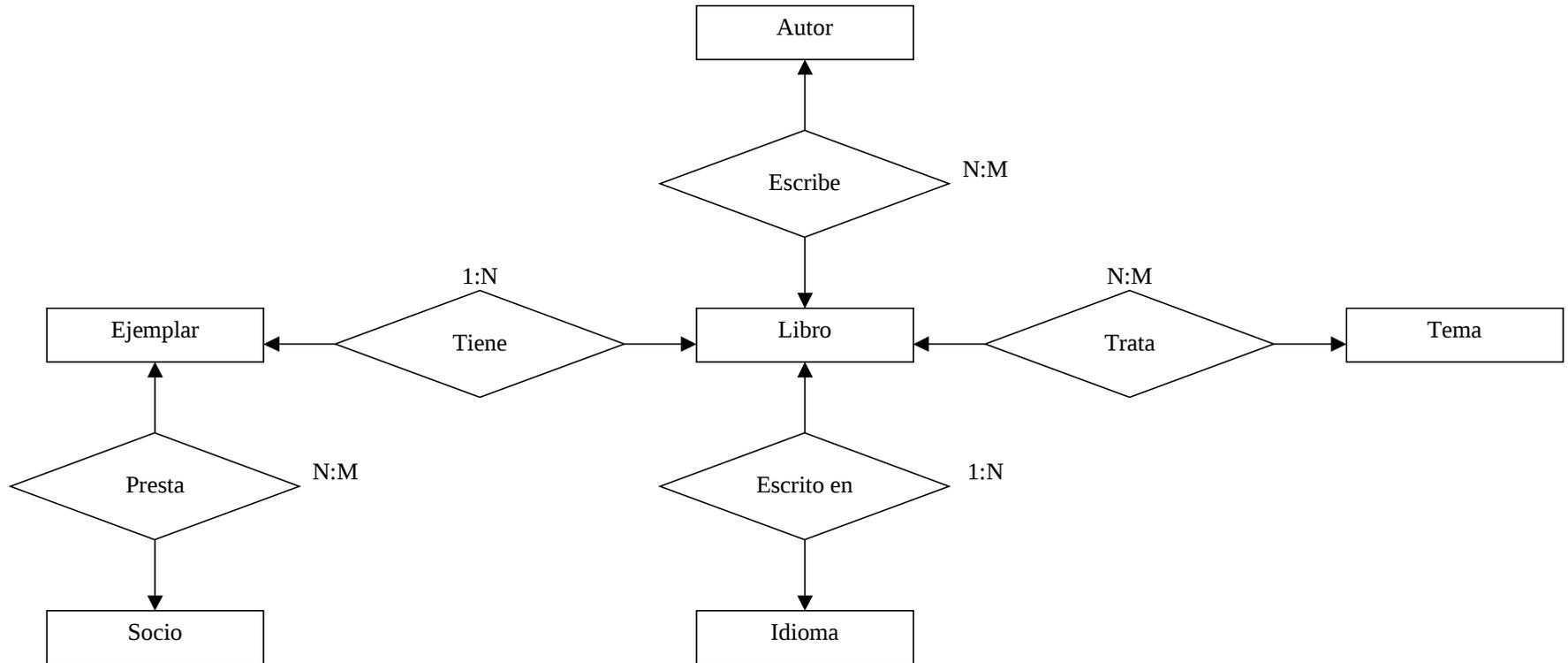
# Diseño conceptual (I)

- Idea intuitiva: Obtener la información necesaria para saber que se desea que represente la BD.
- Suele realizarse mediante consulta con empleados, etc.
- Ejemplo de una biblioteca:
  - Fichas con las características de los libros.
  - Fichas con los datos de los lectores.
  - Fichas de prestamos.
  - Información adicional:
    - De cada libro pueden existir varios ejemplares.
    - Se desea información del idioma de un libro.
    - Se desea reflejar los temas que trata un libro.
    - Se desea conocer el nombre de los autores.

## Diseño conceptual (II)

- Los elementos usados en el diseño conceptual son:
  - Cuadrados para las entidades.
    - Entidad son los elementos que existen realmente, tanto físicos (autores, libros) como lógicos (idiomas, temas).
  - Rombos para las relaciones entre entidades.
    - La cardinalidad indica las posibilidades de relación entre entidades:
      - 1:1 Una entidad A con una entidad B y viceversa.
      - 1:N Una entidad A con N entidades B, pero solo una entidad B con una entidad A.
      - N:M N entidades A con M entidades B y viceversa.

# Diseño conceptual (III)



## Diseño lógico (I)

- La conversión del diseño conceptual al lógico se basa en tres reglas básicas:
  - Toda entidad se convierte en tabla.
  - Toda relación 1:N se convierte en una propagación de clave (primaria o foránea).
    - Excepcionalmente se crea una tabla intermedia.
  - Toda relación N:M se convierte en una tabla intermedia.

## Diseño lógico (II)

- Aplicando la primera regla:

AUTOR

<u>Codigo autor</u> Nombre
-------------------------------

LIBRO

<u>Codigo libro</u> Titulo Año
--------------------------------------

EJEMPLAR

<u>Codigo ejemplar</u>
------------------------

IDIOMA

<u>Codigo idioma</u> Descripción
-------------------------------------

TEMA

<u>Codigo tema</u> Descripción
-----------------------------------

SOCIO

<u>DNI</u> Nombre Domicilio Telefono
---

## Diseño lógico (III)

- Aplicando la segunda regla:

AUTOR

<u>Codigo autor</u> Nombre
-------------------------------

LIBRO

<u>Codigo libro</u> Titulo Año Codigo_idioma
---

EJEMPLAR

<u>Codigo libro</u> <u>Codigo ejemplar</u>
---

IDIOMA

<u>Codigo idioma</u> Descripción
-------------------------------------

TEMA

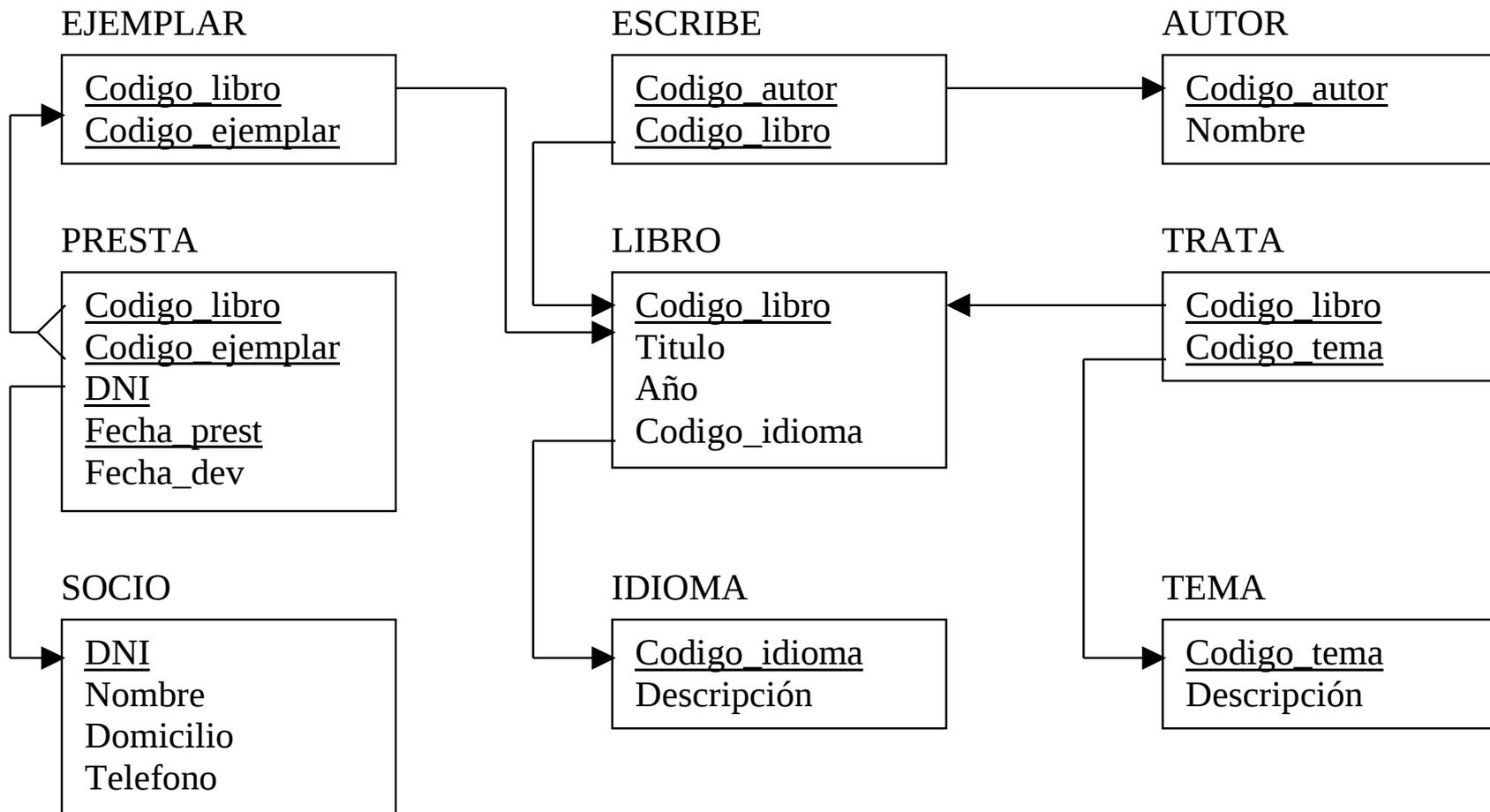
<u>Codigo tema</u> Descripción
-----------------------------------

SOCIO

<u>DNI</u> Nombre Domicilio Telefono
---

# Diseño lógico (IV)

- Aplicando la tercera regla:



## Teoría de la normalización (I)

- En un diseño de una BD. se obtienen unas tablas, pero no podemos estar seguros de que no presenten problemas:
  - Incapacidad de almacenar ciertos hechos.
  - Redundancias.
  - Ambigüedades.
  - Pérdida de información.
  - ...
- Las reglas formales que forman la teoría de la normalización permiten detectar y corregir esos errores.
- Existen 6 FN, aunque lo normal es aplicar las 3 primeras FN.

## Teoría de la normalización (II)

- Tomaremos como ejemplo la facturación a un cliente.

FACTURA

***Codigo\_factura***  
Codigo\_cliente  
Nombre\_cliente  
Direccion\_cliente  
Poblacion\_cliente  
Fecha\_factura  
Forma\_pago  
Codigo\_articulo\_1  
Descripcion\_1  
Cantidad\_1  
Importe\_1  
Tipo\_IVA\_1  
...  
Codigo\_articulo\_N  
Descripcion\_N  
Cantidad\_N  
Importe\_N  
Tipo\_IVA\_N

## Primera forma normal (1FN)

- Una base de datos esta en 1FN si:
  - Cada atributo de una tabla contiene un valor atómico (simple).

### FACTURA

Codigo\_factura  
Codigo\_cliente  
Nombre\_cliente  
Direccion\_cliente  
Poblacion\_cliente  
Fecha\_factura  
Forma\_pago

### DETALLE\_FACTURA

Codigo\_factura  
Codigo\_articulo  
Descripcion  
Cantidad  
Importe  
Tipo\_IVA

## Segunda forma normal (2FN)

- Una base de datos esta en 2FN si:
  - Esta en 1FN.
  - Cada atributo no clave depende de la clave completa y no de parte de ella.
    - Toda tabla con clave formada por un solo atributo cumple con esta propiedad.

FACTURA

<u>Codigo_factura</u>
Codigo_cliente
Nombre_cliente
Direccion_cliente
Poblacion_cliente
Fecha_factura
Forma_pago
Tipo_IVA

DETALLE\_FACTURA

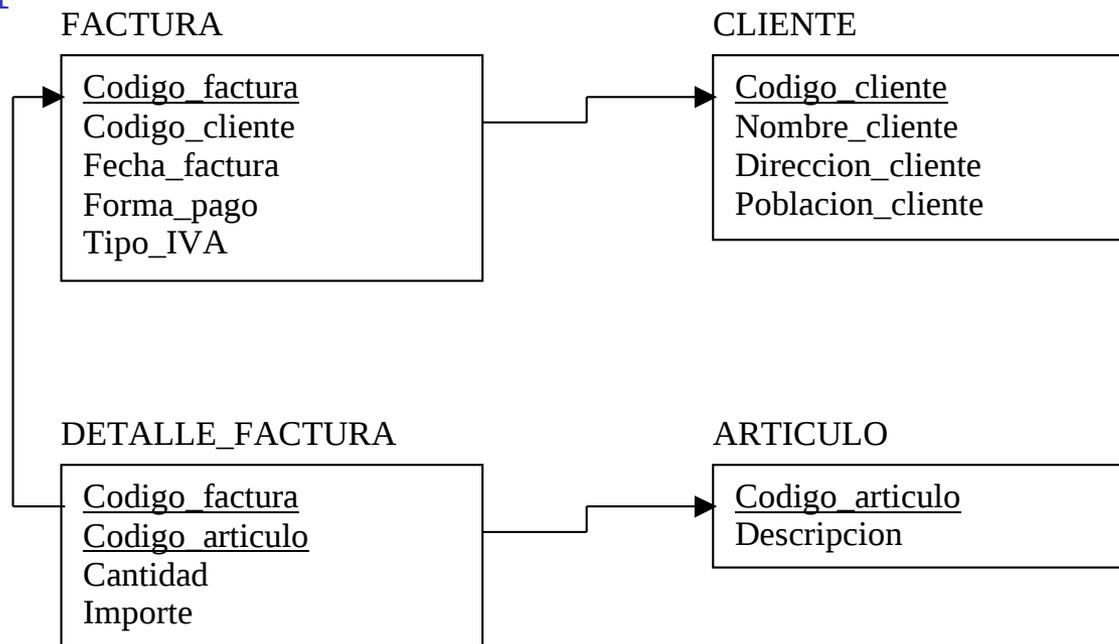
<u>Codigo_factura</u>
<u>Codigo_articulo</u>
Cantidad
Importe

ARTICULO

<u>Codigo_articulo</u>
Descripcion

## Tercera forma normal (3FN)

- Una base de datos esta en 3FN si:
  - Esta en 2FN.
  - Todos los atributos que no son claves son independientes entre si.



## Consideraciones finales

- Si se realiza un diseño conceptual y lógico:
  - Aplicar las formas normales nos comprueba que es correcto.
  - Ayuda a corregir los errores existentes.
- Sin embargo, las FN implican descomponer tablas en otras más pequeñas.
  - Problema de integridad de la base de datos.
  - Disminución del rendimiento del sistema.
- Por tanto es necesario llegar a un compromiso.