

# UD 2. Diseño Lógico de Bases de Datos

## Parte II: El modelo relacional. Normalización



# Índice

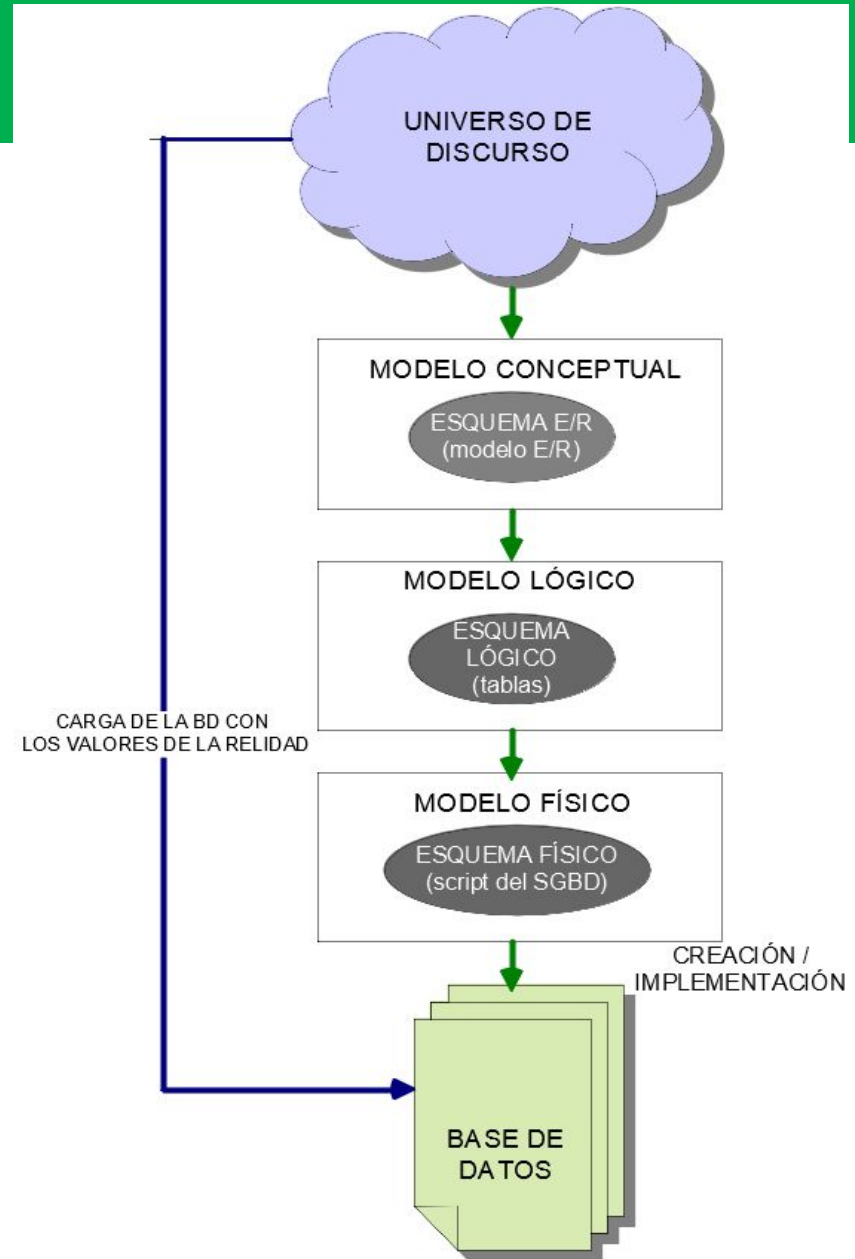
1. Introducción
2. Modelo Relacional
3. Restricciones en el Modelo Relacional
  - 3.1 Definición
  - 3.2 Unicidad de tuplas
  - 3.3 Restricciones de dominio
  - 3.4 Integridad de entidades
  - 3.5 Integridad referencial
  - 3.6 Datos requeridos
  - 3.7 Reglas de negocio
  - 3.8 Sintaxis para el paso a tablas
  - 3.9 Transformación del modelo E/R al modelo relacional
4. Normalización

# 1. Introducción

**Esquema conceptual**= MER + restricciones de negocio

**Esquema lógico**= Esquema relacional + restricciones (integridad, dominio, negocio...)

**Esquema físico**= Scripts de creación de tablas con sus restricciones de integridad, de acceso a disco...



## 2. Modelo Relacional

- Propuesto por Codd en 1970.
- Modelo para Sistemas de Gestión de Bases de Datos.
- Modelo seguido por la mayoría de las BBDDs actuales. **Objetivos:**
  - Independencia física de los datos: La forma de almacenar los datos no debe influir en su manipulación lógica.
  - Independencia lógica de programas y datos: Las aplicaciones que utilizan la base de datos no deben ser modificadas porque se modifiquen elementos de la base de datos.
  - Flexibilidad en cuanto a la forma de presentar los datos: La base de datos ofrece distintas vistas en función de los usuarios y aplicaciones.
  - Uniformidad de las estructuras lógicas de datos: Las estructuras lógicas siempre tienen una única forma conceptual (las tablas).
  - Sencillez: Facilidad de manejo.
- Proporciona un sólido fundamento teórico que permite desarrollar un álgebra y un cálculo relacionales, así como la teoría de normalización.

## 2. Modelo Relacional

Todo SGBD relacional se caracteriza por:

- Presentar sus datos externamente como tablas.
- Disponer de un lenguaje que opere sobre dichas tablas.
- Disponer de interfaces que permitan el acceso desde terminales interactivos y programas de aplicación.
- Disponer de herramientas estándar de respaldo y recuperación y de desarrollo.

## 2. Modelo Relacional

Codd publica en 1985 sus **doce reglas**. De ellas al menos deben satisfacerse seis para que una BD pueda considerarse como totalmente relacional.

1. **Regla de información:** Toda la información está representada a nivel lógico mediante valores en tablas.
2. **Regla de acceso garantizado:** Se puede acceder lógicamente a cada dato mediante la combinación del nombre de la tabla que lo contiene, el nombre de la columna y el valor de la clave primaria de la fila donde está.
3. **Tratamiento sistemático de valores nulos:** Un SGBDR (Sistema de Gestión de Base de Datos Relacional) ha de soportar valores nulos para representar la ausencia de información de forma automática, independientemente del tipo de dato.
4. **Catálogo activo on-line basado en el modelo relacional:** La descripción de la base de datos se representa a nivel lógico como datos ordinarios, de forma que un usuario autorizado pueda utilizar el mismo lenguaje relacional para manejar sus datos y para consultar el catálogo .

## 2. Modelo Relacional

**5. Regla de sublenguaje completo de datos:** Un SGBDR tiene que soportar, al menos, un lenguaje cuyas sentencias tengan una sintaxis bien definida y tenga las operaciones siguientes:

- ▣ Definición de datos
- ▣ Restricciones de integridad
- ▣ Manejo de datos (interactivo o por programa).
- ▣ Definición de vistas
- ▣ Autorizaciones
- ▣ Gestión de transacciones

**6. Regla de actualización de vista:** Las vistas teóricas actualizables son también actualizables por el sistema.

**7. Inserción, modificación y borrado de alto nivel:** La posibilidad de manejar una relación de la base de datos con un único operando no sólo se aplica a la recuperación de datos, sino también a la inserción, la modificación y el borrado de éstos.

## 2. Modelo Relacional

**8. Independencia física de los datos:** Los programas de aplicación y las actividades terminales permanecen lógicamente inalterados cualquiera que sean los cambios efectuados ya sea a las representaciones de almacenamiento o a los métodos de acceso.

**9. Independencia lógica de los datos:** Los programas de aplicación y las actividades terminales permanecen lógicamente inalterados cuando se efectúen sobre las tablas de base cambios preservadores de la información de cualquier tipo que teóricamente permita alteraciones.

**10. Independencia de integridad:** Las restricciones de integridad específicas para una base de datos relacional particular deben ser definibles en el sublenguaje de datos relacional y almacenables en el catálogo, no en los programas de aplicación.

**11. Independencia de distribución:** Un SGBD relacional tiene independencia de distribución.

**12. Regla de no subversión:** Si un sistema relacional tiene un lenguaje de bajo nivel (un solo registro cada vez), dicho lenguaje no puede ser utilizado para subvertir o suprimir las reglas de integridad y las restricciones expresadas en el lenguaje relacional de nivel superior (múltiples registros a la vez).



## 2. Modelo Relacional

- La estructura básica y única del modelo relacional es la **relación** (también llamada **tabla**), que sirve para representar tanto los objetos como las asociaciones entre ellos. Los atributos son las propiedades de las relaciones, y se definen sobre los dominios.
- Para que una tabla forme parte de una estructura relacional, debe cumplir una serie de condiciones.

## 2. Modelo Relacional

- **Condiciones:**

- Debe tener un único tipo de fila.
- Cada fila debe ser única y no pueden existir filas duplicadas.
- Cada columna debe ser única y no pueden existir columnas duplicadas.
- Cada columna debe estar identificada por un nombre específico.
- El valor de una columna para una fila debe ser único.
- Los valores de una columna deben pertenecer al dominio que representa.

## 2. Modelo Relacional

- Una **relación** se compone de un nombre de relación  $R$ , de un conjunto de  $n$  atributos  $\{A_i\}$  y de un conjunto de  $n$  dominios (no necesariamente distintos)  $\{D_i\}$ , donde cada atributo será definido sobre un dominio:

$$R (A_1:D_1, A_2:D_2, \dots A_n:D_n)$$

- Una relación se representa utilizando una tabla donde:
  - Las columnas de la tabla son los atributos que expresan las propiedades de la relación. El número de atributos se llama **grado** de la relación.
  - Cada fila de la tabla, llamada **tupla**, es un elemento del conjunto que forma la relación. El número de tuplas se llama **cardinalidad** de la relación. La cardinalidad varía en el transcurso del tiempo.

## 2. Modelo Relacional

- El universo del discurso (UD) de una base de datos relacional está compuesto por un conjunto de dominios  $\{D_i\}$  y de relaciones  $\{R_i\}$  definidas sobre los dominios.
- Un **dominio** es un conjunto nominado, finito y homogéneo (del mismo tipo) de valores atómicos (indivisibles). Formatos:
  - Extensión: Dando sus posibles valores.
  - Intensión: Tipo de datos.
- Un **Atributo (A)** es la interpretación de un determinado dominio en la relación (tabla), es decir, el papel que desempeña en la misma; si D es el dominio de A se denota:
  - $D = \text{Dom}(A)$

## 3. Restricciones en el Modelo Relacional

### 3.1 Definición

- Son **restricciones** sobre los datos que se deben de cumplir para que la BD no se quede en un estado incoherente.
- Hay cinco tipos:
  - Unicidad de tuplas
  - Restricciones de dominio
  - Integridad de entidades
  - Reglas de validación
    - Datos requeridos
    - Reglas de negocio
  - Integridad referencial

# 3. Restricciones en el Modelo Relacional

## 3.2 Unicidad de tuplas

- La **restricción de unicidad de tuplas** de una relación cumple con las siguientes condiciones:
  - Los datos deben ser únicos (unicidad).
  - Los datos deben ser mínimos y cumplir la regla de No redundancia o Minimalidad.
- Ejemplo:

EMPLEADO
# dni
nombre
apellidos
salario

EMPLEADO (#dni, nombre, apellidos, salario)

**RI de unicidad de tuplas:**

- dni no NULO, tuplas únicas y mínimas
- El resto de atributos podrían ser NULOS (no es el caso), y no cumplir con unicidad y no redundancia.

## 3. Restricciones en el Modelo Relacional

### 3.3 Restricciones de dominio

- Todos los atributos deben contener valores dentro de un dominio definido.
- Ejemplo:

EMPLEADO
# dni
nombre
apellidos
salario

EMPLEADO (#dni, nombre, apellidos, salario)

#### RI de dominios:

- dni es cadena [9]
- nombre es cadena [50]
- apellido es cadena [100]
- salario es real  $> 0$

## 3. Restricciones en el Modelo Relacional

### 3.4 Integridad de entidades

- El **identificador** de una entidad **no puede ser nulo**, por lo tanto, las claves primarias son no nulas.

- Ejemplo:

EMPLEADO
# dni
nombre
apellidos
salario

EMPLEADO (#dni, nombre, apellidos, salario)

RI de integridad de entidades:

- dni es clave primaria: no nula

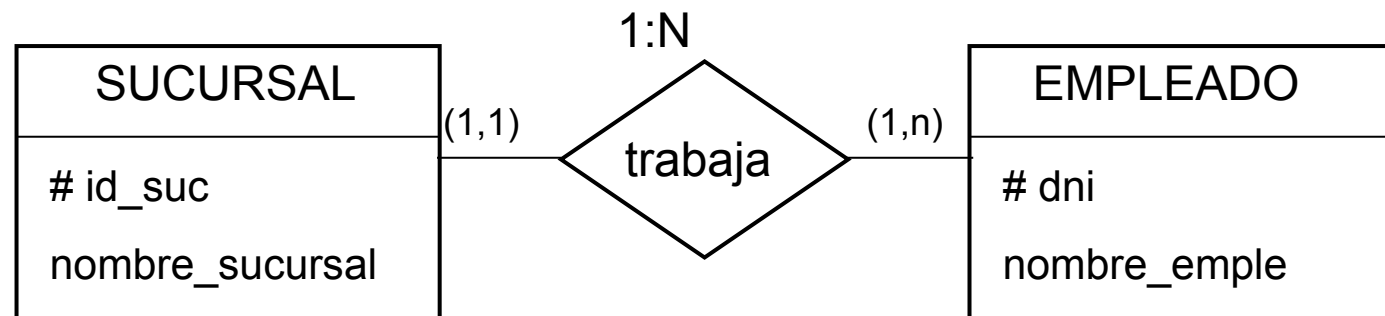


### 3. Restricciones en el Modelo Relacional

#### 3.5 Integridad referencial

- Sabemos que una clave ajena enlaza cada registro de la relación HIJO con el registro de la relación PADRE que tiene el mismo valor en su clave primaria. La **integridad referencial** dice que: *“si una clave ajena tiene un valor (si no es nula), ese valor debe ser uno de los valores de la clave primaria a la que referencia”*.

- Ejemplo:



id_suc	nombre_suc
1	Guadalquivir
2	Guadarrama

dni	nombre	id_suc
21799890A	Pepe	1
28399089J	María	1
23422654M	Juan	3

No existe la sucursal 3. Violación de la restricción de integridad referencial.

## 3. Restricciones en el Modelo Relacional

### 3.5 Integridad referencial

- Otra situación:
  - ¿Qué ocurre si borro la sucursal 1? Violación de la Restricción de Integridad Referencial.
  - ¿Qué ocurre si actualizo el `id_suc=1` y lo cambio a `id_suc=4`? Violación de la Restricción de Integridad Referencial.
- Ante la posibilidad de borrar o actualizar la clave en un registro referenciado tenemos distintas formas de actuar:
  - Restringerlo.
  - Propagarlo.

## 3. Restricciones en el Modelo Relacional

### 3.5 Integridad referencial

- **Restricción (restrict):** no se puede borrar/actualizar registros referenciados.
  - Ejemplo: EMPLEADO(#dni, nombre, id\_suc), id\_suc es clave ajena que referencia al campo 'id\_suc' de SUCURSAL.
  - Si intento borrar/actualizar una clave en SUCURSAL y tiene empleados asociados NO ME DEJA. Habría que borrar primero los empleados asociados y después borrar o actualizar la sucursal.
  - En SQL:
    - `constraint fk_sucursal foreign key(id_suc) references SUCURSAL (id_suc) on delete/update restrict`

## 3. Restricciones en el Modelo Relacional

### 3.5 Integridad referencial

- **En cascada (cascade):** se propaga el borrado/actualizado. Si borro/actualizo una sucursal se borran/actualizan todos sus empleados.
  - Poner a nulo (SET NULL): sólo para el caso del borrado. Se borra la tupla deseada y todos los campos (claves ajenas) que la refieren se ponen a NULL. Si borramos una sucursal, NO se borran sus empleados, pero se ponen a NULL todas sus claves ajenas.
  - Valor por defecto (SET DEFAULT): sólo para el borrado. Igual que antes pero ahora se les da a esos campos un valor por defecto que debe ser especificado.
  - No comprobar (NO ACTION): si se intenta borrar/actualizar NO se realiza comprobaciones de restricciones de integridad referencial.

## 3. Restricciones en el Modelo Relacional

### 3.6 Datos requeridos

- Algunos atributos deben contener valores en todo momento, es decir, no pueden ser nulos (**datos requeridos**).
- Ejemplo:

EMPLEADO
# dni
nombre
apellidos
salario

EMPLEADO (#dni, nombre, apellidos, salario)

**RI de datos requeridos:**

dni no NULO

nombre no NULO

apellido no NULO

salario no NULO

## 3. Restricciones en el Modelo Relacional

### 3.7 Reglas de negocio

- Son las restricciones que impone el **negocio de la empresa**.
- Ejemplo:
  - Un socio no puede alquilar más de tres películas
  - El pedido mínimo debe superar los 10 euros
  - El precio de venta no puede superar un 15% del precio de fábrica
  - ...
- Las reglas de negocio junto a las reglas de datos requeridos constituyen las reglas de validación.

# 3. Restricciones en el Modelo Relacional.

## 3.8 Sintaxis para el paso a tablas

Nombre\_tabla (#campo1, ..., campoN, ...) siendo

campo1 es tipo\_dato [no nulo] [comprueba (condición\_lógica)],

campo2 es tipo\_dato [no nulo] [comprueba (condición\_lógica)],

...

PK (campo1, campo2, ...),

FK (campo1, campo2, ...) de nombre\_tabla

con (operación acción, ...)

### Notación:

- Las palabras reservadas se muestran en **negrita**.
- Lo que se encierra entre corchetes es opcional.
- Los tipos de datos más usuales son: entero, autoincrementado, real, carácter, cadena, fecha, hora, fecha-hora, lógico.
- Las operaciones posibles son: actualización y borrado.
- Las acciones posibles son: restringido/a, en cascada, poner a nulo, poner por defecto valor xx, sin comprobación.

# 3. Restricciones en el Modelo Relacional

## 3.8 Sintaxis para el paso a tablas

Ejemplos:

Sucursal (#id\_suc, nombre, dirección, teléfono) siendo

id\_suc es entero no nulo,

nombre es cadena no nulo,

direccion es cadena no nulo,

teléfono es entero,

PK (id\_suc)

Empleado (#id\_emp, nombre, apellidos, dirección, salario, id\_sucursal) siendo

id\_emp es entero no nulo,

nombre es cadena no nulo,

apellidos es cadena no nulo,

salario es real comprueba (salario > 600),

id\_sucursal es entero,

PK (id\_emp),

FK (id\_sucursal) de Sucursal

con (actualización en cascada, borrado restringido)



## 3. Restricciones en el Modelo Relacional

### 3.9 Transformación del modelo E/R al modelo relacional

Los pasos o reglas a seguir se describen a continuación:

- **Transformación de dominios:** es equivalente en ambos modelos  
atributo -> atributo
- **Transformación de entidades:**  
entidad -> relación
- **Transformación de atributos:**
  - Identificador principal -> Clave primaria
  - Identificador alternativo -> Clave secundaria
  - Atributos no clave -> Columnas de la tabla
  - Atributos multivaluados -> Nueva tabla cuyo identificador es el de la entidad original más el propio atributo

## 3. Restricciones en el Modelo Relacional

### 3.9 Transformación del modelo E/R al modelo relacional

- **Transformaciones de relaciones:**

- **N:M:** Nueva tabla cuya clave es la unión de las claves.
- **1:N:** Dos soluciones:
  - Propagación de clave del lado 1 al lado n.
  - Nueva tabla cuya clave es la unión de ambas.
- **1:1:** Tres soluciones:
  - Nueva tabla con cardinalidad (0,1) en ambas.
  - Propagación de clave del lado 1 al lado 0 con cardinalidad (1,1) en una entidad.
  - Propagación de clave en cualquier sentido con cardinalidad (1,1) en ambas

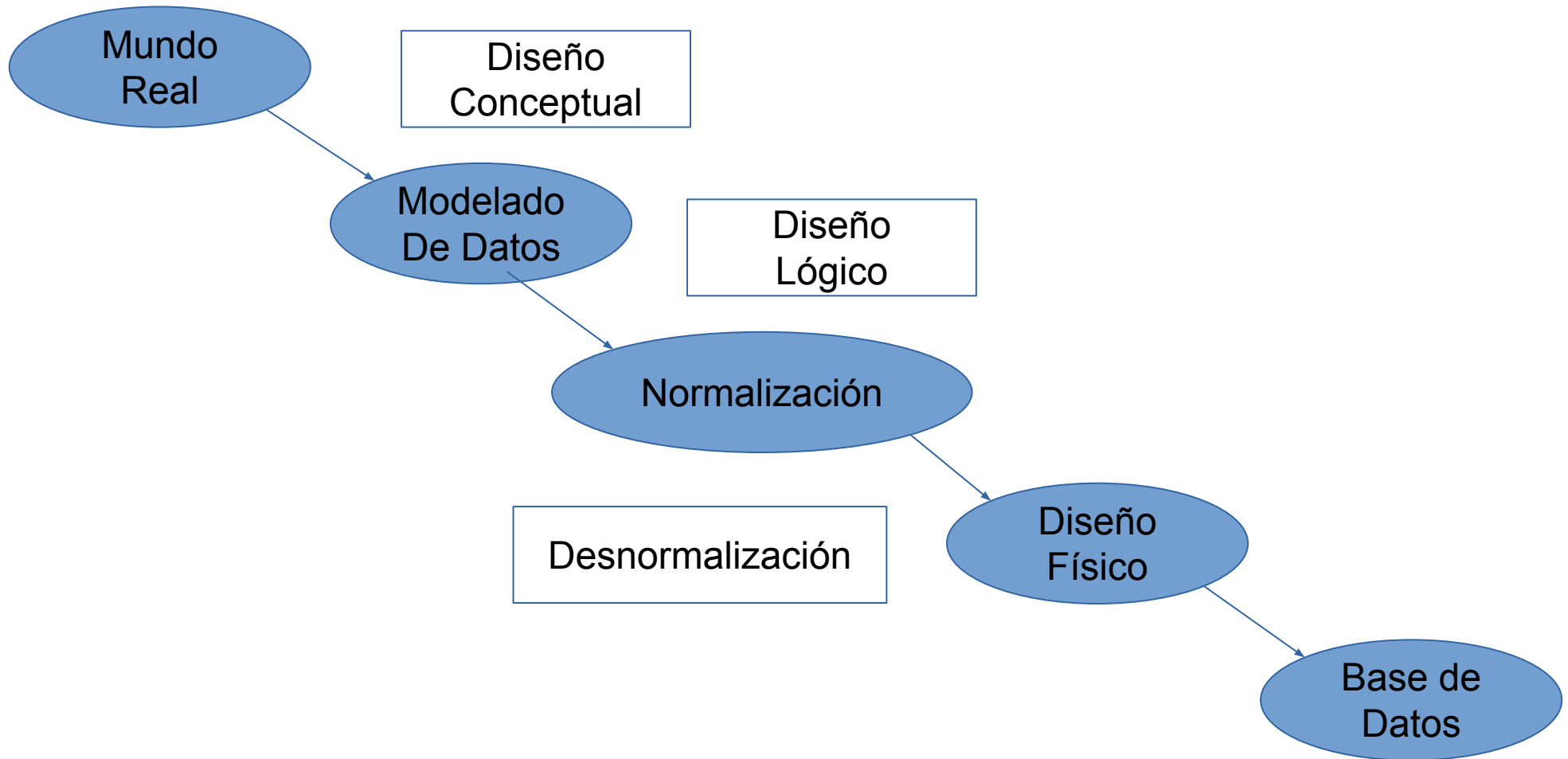
## 3. Restricciones en el Modelo Relacional

### 3.9 Transformación del modelo E/R al modelo relacional

- **Transformaciones de jerarquías:** Tres soluciones posibles:
  - Transformar cada entidad en una tabla donde los subtipos heredan los atributos del supertipo.
  - Formar una única relación con valores nulos para los campos específicos de los subtipos.
  - Formar una relación por cada subtipo integrando los campos comunes del supertipo.

*Para más ampliación véase el Anexo del Algoritmo de Transformación Paso a Tablas disponible en la plataforma educativa.*

# 4. Normalización



# 4. Normalización

## ¿Qué es la Normalización de una Base de Datos?

- Puede definirse como una técnica formal para organizar los datos.
  - Nos ayuda a determinar qué está equivocado en un diseño y la manera de corregirlo.
  - Las anomalías de diseño a las que hacemos referencia se producen sólo en los procesos de actualización y nunca en los de consulta.
- La aplicación de la teoría de la normalización evita problemas que se puedan producir en las actualizaciones pero, al mismo tiempo, penaliza las consultas.
  - El proceso de normalización aumenta el número de tablas.
  - Una posible consulta puede ocasionar el acceso a varias tablas.

*Enunciado ME/R + Restricciones → Esquema de tablas → Reglas de Normalización → Esquema de tablas Normalizadas.*

# 4. Normalización



Visión de la Teoría de Normalización

# 4. Normalización

## Otros conceptos de utilidad

- **Clave candidata:** Todo campo o conjunto de campos que identifican inequívocamente a un registro o tupla.
  - EJ Empleado: DNI, Cod Empleado, etc
- **Clave primaria (PK):** Es la clave candidata que elegimos.
- **Clave ajena o foránea (FK):** Es un campo de una tabla que es la clave principal en otra <sup>→</sup>tabla relacionada con la actual.
- **Dependencia:** A B (B depende de A) si conociendo el valor de A, sólo hay un valor posible de B. Ej: Los atributos dependen de la clave candidata y de la primaria.

# 4. Normalización

## Formas de normalización

Los teóricos de la base de datos relacionales han dividido la normalización en varias reglas llamadas formas normales.

Para que una base de datos esté en la segunda forma normal, debe también estar en la primera forma normal, y para que esté en la tercera forma normal, debe cumplir los requisitos para la primera y segunda forma normal.

Hay también formas adicionales de normalización, pero éstas se aplican raramente. De hecho, puede, ocasionalmente, ser práctico violar incluso las primeras tres formas de normalización.



# 4. Normalización

## 1ª Forma Normal

- Una BBDD está en **1ª Forma Normal** si todos sus dominios son atómicos, es decir, en cada campo solo hay un valor y además no se repiten grupos de campos.

Ejemplo:

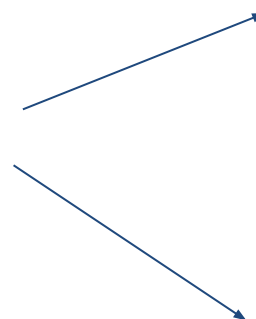
#cod_arti culo	descripci on	medi das
39	Tornillo	3'5 5 7
40	Tuerca	2 5
41	Arandela	2'5 3 9

Un mismo atributo toma varios valores en un mismo instante. No está en 1FN.

## 4. Normalización

- **Solución (1ª FN):** Los campos que se repiten, se pasan a una nueva tabla junto con la clave primaria de la tabla origen.

#cod_articulo	descripcion	medidas	#cod_articulo	descripcion
39	Tornillo	3'5 5 7	39	Tornillo
40	Tuerca	2 5	40	Tuerca
41	Arandela	2'5 3 9	41	Arandela



#cod_articulo	medidas
39	3'5
39	5
39	7
40	2
40	5
...	...

*Otro posible ejemplo es el dato teléfono cuando puede haber más de uno. En casos como los campos nombre también se viola la primera forma normal.*

# 4. Normalización

## 2ª Forma Normal

- Una BBDD está en **2ª Forma Normal** si está en 1ª FN y además, todos los atributos que no son clave dependen de todos los atributos de la clave primaria y no de una parte de ella.
- Se deduce que toda relación cuya clave esté formada por un único atributo está, al menos, en 2ª FN ( siempre que la clave esté bien formada).

- Ejemplos:

*CLIENTE( #dni, nombre, apellidos) → 2FN La 2FN se cumple automáticamente cuando la clave es atómica.*

*PRÉSTAMOS( #num\_socio, nombre\_socio, #cod\_libro, fecha\_préstamo, editorial, país) 'editorial' sólo depende de 'cod\_libro', no de la clave completa, al igual que 'nombre\_socio' que solo depende de 'num\_socio'*

## 4. Normalización

- **Solución (2ª FN):** Se sacan los atributos no claves que dependen solo de la parte de la clave primaria a una nueva tabla con la parte de la clave primaria de la que dependen.

*PRÉSTAMOS( #num\_socio, #cod\_libro, fecha\_préstamo)*

*SOCIO(#num\_socio, nombre\_socio)*

*LIBROS(#cod\_libro, editorial, país)*

# 4. Normalización

## 3ª Forma Normal

- Una BBDD está en **3ª Forma Normal** si está en 2ª FN y se comprueba que los atributos que no forman parte de ninguna clave candidata facilitan información sólo acerca de las claves y no lo hacen acerca de otros atributos no incluidos en estas.

Ejemplo:

*LIBROS(#cod\_libro, editorial, país) El atributo 'país' facilita información acerca de la editorial que publica el libro, y no del libro en sí.*

## 4. Normalización

- **Solución (3ª FN):** Se sacarán a una tabla nueva los atributos no claves dependientes de otro atributo no clave junto a este último.

Relaciones:

*LIBROS(#cod\_libro, editorial)*

*EDITORIAL (#editorial, país)*

# 4. Normalización

## Definiciones poco rigurosas pero explicativas

- 1ªFN: Una relación está en 1ªFN si contiene para cada intersección un solo valor y no un conjunto de ellos.
- 2ªFN: Una relación está en 2ªFN si está en 1ªFN y cada atributo depende de toda la clave.
- 3ªFN: Una relación está en 3ªFN si está en 2ªFN y cada atributo depende sólo de la clave.

**¡OJO! Un conjunto de relaciones sacado de un MER correcto estará en Tercera Forma Normal.**

*La Forma Normal de Boyce-Codd y la Cuarta y Quinta Formas Normales no las estudiaremos en este curso.*