

Diseño Lógico de bases de datos

Pasos previos a la Transformación E/R a relacional	2
Transformación de relaciones	2
Relaciones binarias N:M	2
Relaciones binarias 1:N	3
Norma general	3
Excepciones	3
CASO 1 (0,1) - NUEVA TABLA DE LA RELACIÓN	3
CASO 2 N:M - NUEVA TABLA DE LA RELACIÓN	3
CASO 3 FECHA - NUEVA TABLA DE LA RELACIÓN	3
Relaciones Binarias 1:1	3
Relaciones Binarias 1:1 - Participaciones (1,1) (1,1)	4
Relaciones Binarias 1:1 - Participaciones (0,1) (0,1)	4
Relaciones Binarias 1:1 - Participaciones (1,1) (0,1)	4
Relaciones Reflexivas	4
Reflexivas 1:1	4
Reflexivas 1:N	4
Reflexivas M:N	5
Transformación de las Entidades Débiles	5
Transformación de Jerarquías	6
CASO A	6
CASO B (COMODÍN)	6
CASO C	7
Normalización	8
Dependencia funcional (Es una dependencia directa de un atributo con otro):	8
Dependencia transitiva (Dependencia indirecta entre atributos)	8
Formas Normales	9
Primera forma normal (1FN)	9
Segunda forma normal (2FN)	10
Tercera forma normal (3FN)	11
Forma normal de Boyce Codd	12
Diccionario de Datos	14

Pasos previos a la Transformación E/R a relacional

Para cumplir con la 1FN hay que eliminar los **Atributos con valores múltiples** y los **Atributos compuestos**.

Los **Atributos con valores múltiples** pasan a transformarse en entidades débiles.

Los **atributos compuestos** se descomponen en atributos simples

Entidades pasan a ser tablas (con el nombre en plural*).

Atributos pasan a ser columnas.

Identificadores principales pasan a ser claves primarias.

Identificadores candidatos pasan a ser claves candidatas.

Transformación de relaciones

La idea inicial es transformar cada relación en una tabla, pero hay que distinguir según el tipo de relación:

Relaciones uno a muchos 1:N

Relaciones uno a uno 1:1

Relaciones muchos a muchos N:M

Relaciones reflexivas

Relaciones binarias N:M

crear una tabla aparte que contendrá una clave foránea a cada una de las tablas correspondientes además de los atributos de la relación. La clave primaria será compuesta por las dos claves foráneas.

En los casos en los que la relación contenga atributo fecha, será necesario incluirlo como identificador principal.

Relaciones binarias 1:N

Cuando la relación entre dos entidades fuertes es de uno a muchos, hay dos modos:

Norma general: Mediante clave ajena.

3 Excepciones: Mediante una tabla aparte.

Norma general

La entidad que tiene la N importa la clave la que tiene el 1 como foránea, además de los atributos de la relación.

Excepciones

CASO 1 (0,1) - NUEVA TABLA DE LA RELACIÓN

Cuando una participación sea (0,1) y tengamos muchos casos con 0. En este caso el número de ocurrencia la entidad que propaga la clave es pequeño y pueden haber muchos nulos.

CASO 2 N:M - NUEVA TABLA DE LA RELACIÓN

Cuando es posible que la relación pueda convertirse en el futuro en N:M (el lado que tiene el 1 pase a ser N).

CASO 3 FECHA - NUEVA TABLA DE LA RELACIÓN

Cuando la relación tenga atributos propios y sobre todo cuando tenga una fecha. Esta fecha formará parte de la clave primaria de la relación.

Relaciones Binarias 1:1

Es preciso revisarlas, ya que es posible que se hayan identificado dos entidades que representen el mismo concepto pero con nombres diferentes (sinónimos). Si este es el caso, ambas entidades deben integrarse en una sola.

Hay tres formas distintas de representar, en el esquema lógico:

Las participaciones de ambas entidades son (1,1)

Las participaciones de ambas entidades son (0,1)

Una tiene participación (1,1) y otra (0,1)

Relaciones Binarias 1:1 - Participaciones (1,1) (1,1)

Propagar la clave de la que vaya a recibir menos consultas a la que vaya a recibir más consultas, teniendo en cuenta a cuál de ellas se le efectúan los accesos más frecuentes.

Relaciones Binarias 1:1 - Participaciones (0,1) (0,1)

La relación se transforma en una tabla para evitar los valores nulos.

Relaciones Binarias 1:1 - Participaciones (1,1) (0,1)

Se propaga la clave de la entidad que tiene cardinalidad (1,1) a la que tiene (0,1).
Habrà que controlar además que la clave propagada como clave ajena no pueda tomar valores nulos.

NOTA: En las relaciones N:M y 1:N si hay fecha en la relación, creamos una tabla con ella.
En la 1:1, si hay fecha en la relación, no hay tabla y esta es absorbida sin ser clave primaria.

Relaciones Reflexivas

Regla general: se convierte en dos tablas: una para la entidad y otra para la relación.

Se pueden presentar los siguientes casos:

Relación 1:1

Relación 1:N

Relación N:M

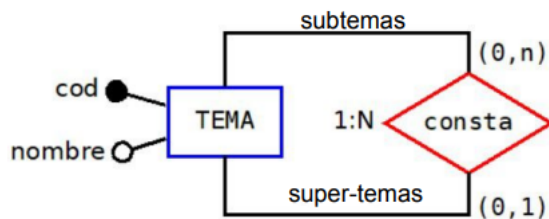
Reflexivas 1:1

No se crea una tabla para la relación. Sino que la clave de la entidad se repite, con lo que la tabla resultante tendrá ese atributo dos veces, una como clave primaria y otra como clave ajena de ella misma.

Reflexivas 1:N

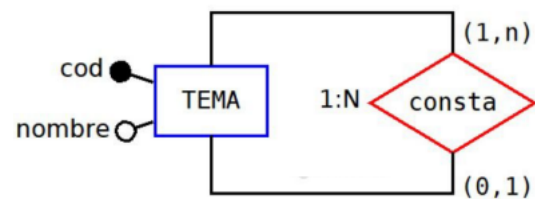
Habrà que atender al lado de MUCHOS (N):

- A) Si no es obligatoria (0,n): se crea una nueva tabla cuya clave primaria será la primaria de la entidad del lado muchos (de forma foránea) (codTemaH) y se propaga la clave a la nueva tabla como clave ajena (entendiendo que la clave del otro lado, el del 1, siendo también foránea pero no principal).
- B) Si es siempre obligatoria (1,n): no se crea una nueva tabla



TEMA (codTemaH, nombre)

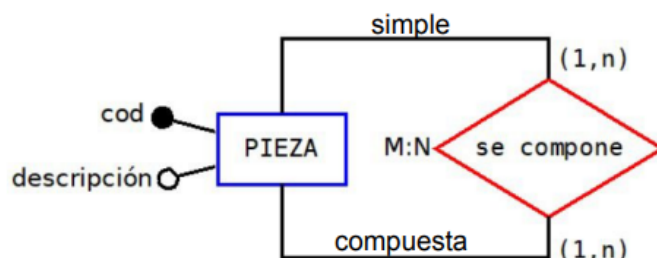
Composición_temas (codTemaH*, codTemaP*)



TEMA (codTemaH, nombre, codTemaP*)

Reflexivas M:N

Se crea una tabla para la relación, que contendrá dos veces la clave primaria del lado de muchos más los atributos de la relación. La clave será la combinación de las dos claves.



PIEZAS (codPiezaS, Descripción)

COMPOSICIÓN (codPiezaC*, codPiezaS*)

(codPiezaC es la pieza que se compone de otras)

Cada pieza se compone de muchas piezas que, a su vez están compuestas por piezas.

Transformación de las Entidades Débiles

Recuerda que las relaciones entre entidades fuerte-débil tienen una cardinalidad 1:N y la participación del lado 1 será (1,1).

Se tratan como las relaciones 1:N, la entidad que tiene la N importa la clave la que tiene el 1 como foránea, además de los atributos de la relación. Esta clave foránea junto a la débil, serán la clave primaria.

Transformación de Jerarquías

Hay tres opciones distintas atendiendo a su tipo o (total o parcial, y exclusiva o superpuesta) y del tipo y frecuencia en los accesos a los datos.

Caso A (Matas hijos): posible cuando cardinalidad = (1,1) o (0,1) [El caso 1:N equivale a hacer una tabla para todo, osea es el Caso B]

Caso B (Tabla a cada uno): Vale para todo.

Caso C (Matas padre): Solo con (1,1)

CASO A

Integrar todas las entidades en una sola tabla (PADRE), junto a todos los atributos y añadir un atributo discriminatorio para indicar el subconjunto.

Si la jerarquía es superpuesta (min,n) el tipo deberá ser multivaluado o incluir cada subtipo como un atributo.

Es aplicable cuando los hijos tienen pocos atributos y si en caso de tener relaciones, estas son compartidas.

Si la jerarquía es total: el tipo no admite nulos, si es parcial, el tipo no admite nulos.

Si entre los subtipos puede haber:

Solapamiento/Superpuesto = Se forman grupos repetitivos, por tanto será necesario crear una nueva tabla que asocie el atributo discriminante con el supertipo.

Exclusividad = No es necesaria una tabla nueva.

CASO B (COMODÍN)

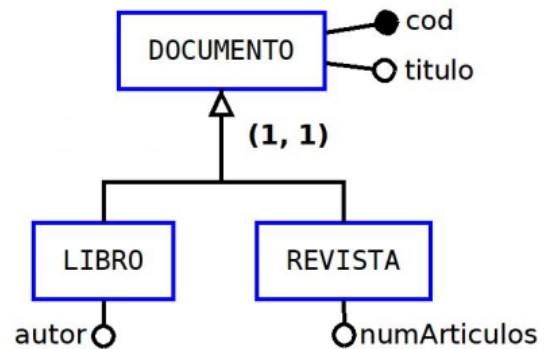
Crea una tabla para la entidad madre y una tabla para cada entidad hija.

Las hijas heredan como clave foránea la clave primaria de la madre, siendo esta su clave primaria.

DOCUMENTO (codigo, titulo)

LIBRO (codigo*, autor)

REVISTA (codigo*, numArticulos)



Esto se puede hacer para cualquier tipo de jerarquía (total, parcial, exclusiva o superpuesta). Se usa cuando hay muchos atributos distintos entre los subtipos y cuando se quieren mantener los atributos comunes en una tabla.

CASO C

Creas una tabla para cada hija y en ella pones los atributos de la madre.

Esta representación sólo puede hacerse para jerarquías totales y exclusivas (1, 1).

Este caso se suele aplicar cuando los subtipos tienen muchos atributos distintos y cuando los accesos a los datos de los subtipos siempre afectan a los atributos comunes.

Transformación de Jerarquías - Ventajas e inconvenientes

- Opción A:
 - Es la más rápida por tener que acceder a una sola entidad.
 - Se quedan atributos vacíos

DOCUMENTO (codigo, titulo, autor, numArticulos, **tipo**)

- Opción B:
 - La menos eficiente
 - La mejor desde un punto de vista semántico

DOCUMENTO (codigo, titulo)
LIBRO (codigo*, autor)
REVISTA (codigo*, numArticulos)

- Opción C:
 - Más eficiente en consultas que afecten a todos los atributos de un subtipo
 - Menos eficiente en consultas que afecten a los atributos comunes
 - Introduce redundancias
 - Es la que pierde más semántica

LIBRO (codigo, titulo, autor)
REVISTA (codigo, titulo, numArticulos)

Normalización

Técnica que busca dar eficiencia y fiabilidad a una BD relacional.

Su objetivo es:

1. Mejorar la velocidad en el manejo de la información
2. Eliminar la redundancia de datos
3. Ordenar datos

Al realizar un diseño en el modelo relacional, se pueden llegar a diferentes alternativas, que no serán equivalentes. La normalización elimina este problema, dividiendo las tablas en dos o más.

Dependencia funcional (Es una dependencia directa de un atributo con otro):

La que existe entre dos atributos de una tabla si para cada valor del primer atributo existe un sólo valor del segundo. Ejemplo es que por un DNI habrá un nombre y una dirección.

DNI → Nombre

DNI → Dirección

DETERMINANTE → DEPENDIENTE

Normalización - EJEMPLO Dependencia funcional

LIBRO (Título_libro, Num_ejemplar, Autor, Editorial, Precio)

Editorial y Precio dependen funcionalmente del conjunto de atributos que forman la clave primaria, pero no dependen de uno de ellos por separado. Dependen funcionalmente de la clave primaria.

Autor depende funcionalmente sólo y exclusivamente de Título_libro, por lo que no presenta una dependencia funcional completa de los atributos que forman la clave.

El modelo relacional sólo requiere un conjunto de tablas en primera forma normal (en caso contrario no se pueden implementar)

Dependencia transitiva (Dependencia indirecta entre atributos)

Dados tres atributos A, B y C, se dice que existe una dependencia transitiva entre A y C, si B depende funcionalmente de A y C depende funcionalmente de B.

Formas Normales

Sólo consideraremos las tres primeras formas normales:

- Primera forma normal (1FN)
- Segunda forma normal (2FN)
- Tercera forma normal (3FN)
- Forma normal de Boyce Codd (FNBC)

Primera forma normal (1FN)

Se impide que un atributo de una tupla pueda tomar más de un valor (Multivaluado) (Esta transformación ya la hemos realizamos al considerar los atributos multivaluados como entidades débiles)

Primera forma normal (1FN)



- Se impide que un atributo de una tupla pueda tomar más de un valor (Multivaluado)
- Es inherente en el esquema relacional, toda tabla realmente relacional la cumple
- Por lo que su cumplimiento es obligatorio

Esta transformación ya la hemos realizamos al considerar los atributos multivaluados como entidades débiles

TRABAJADOR

NOMBRE	Departamento
Elias	Informática
David	Coordinación
	Mantenimiento

NO CUMPLE

TRABAJADOR

NOMBRE	Departamento
Elias	Informática
David	Coordinación
David	Mantenimiento

SÍ CUMPLE AUNQUE NO ES CORRECTO

Todos los atributos múltiples se deben transformar en un tipo de **entidad débil** por existencia. (creando una nueva tabla)

Primera forma normal (1FN)



Empleados (cod, nombre, apellidos....);

Departamentos(cod, codEmpleado*, nombre ...);

Departamentos.codEmpleado hace referencia a Empleados.cod y no es null

Segunda forma normal (2FN)

Ocurre si una tabla está en primera forma normal (1FN) y además cada atributo que no forma parte de la clave primaria es completamente dependiente de la clave primaria.

Si una tabla está en 1FN y su clave es simple, también está en 2FN.

Pero si la tabla está en 1FN y su clave es compuesta puede que haya que trabajar para llevarla a 2FN.

Para pasar de 1FN a 2FN hay que eliminar las dependencias parciales de la clave primaria. Para ello si un atributo no depende de la clave completa, se aparta a otra tabla junto a la parte de la clave de la que depende, y en la 1ª tabla se queda en vez de la parte de la clave, su referencia como clave foránea.

- La siguiente tabla no satisface la segunda forma normal
 - (no es 2FN)

¿Que pasó si queremos modificar el nombre de un Alumno? Tendríamos un conflicto

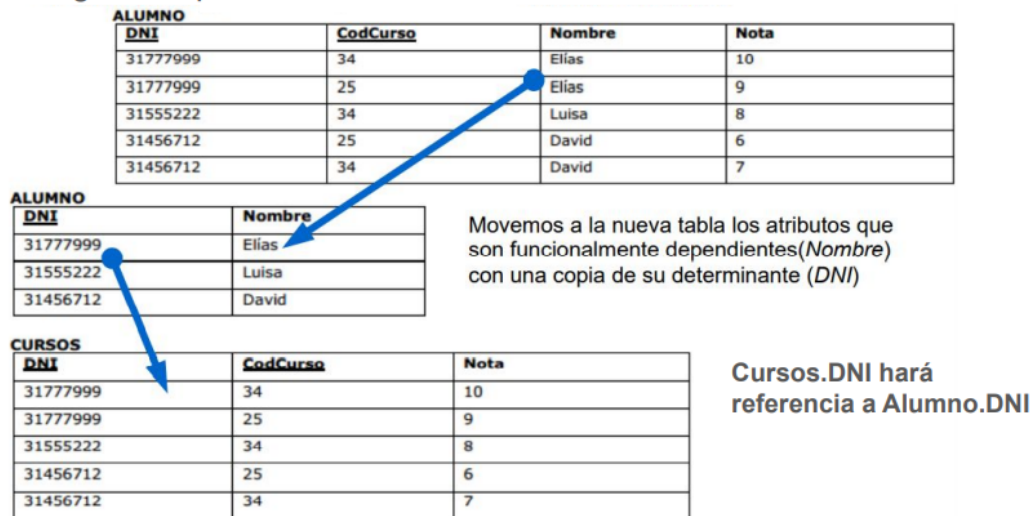
ALUMNO			
<u>DNI</u>	<u>CodCurso</u>	Nombre	Nota
31777999	34	Elías	10
31777999	25	Elías	9
31555222	34	Luisa	8
31456712	25	David	6
31456712	34	David	7

- DNI y código de curso forman la clave primaria compuesta
- sólo la nota tiene dependencia funcional completa con respecto a la clave primaria
- El nombre depende de forma completa del DNI.

Segunda forma normal (2FN)



- Para arreglarlo separamos la tabla en dos:



Tercera forma normal (3FN)

NOTA: Cuando tienes una tabla en 2FN, todavía pueden haber dependencias transitivas, esto es lo que se quiere eliminar en la 3FN.

Se da cuando una tabla está en 2FN y además no existen dependencias transitivas a la clave primaria.

Para eliminar estas dependencias transitivas, se eliminan los atributos que dependen transitivamente y se ponen en una nueva tabla con una COPIA de su determinante (Atributo o atributos no clave del que depende)

- Una tabla no está en 3FN si:
 - Hay algún atributo no clave que depende funcionalmente de forma transitiva de otro campo no clave.
- La siguiente tabla no está en 3FN

ALUMNO			
<u>DNI</u>	Nombre	<u>CodProvincia</u>	Provincia
31777999	Elías	11	Cádiz
31777111	Pepe	41	Sevilla
31555222	Rosa	29	Málaga
31717171	Juana	11	Cádiz
12002003	Manuela	08	Madrid

- La provincia depende funcionalmente del código de provincia
- Tenemos que sacar el nombre de la provincia a otra tabla

Tercera forma normal (3FN)

- Se arregla separando en dos tablas

<u>DNI</u>	Nombre	CodProvincia
31777999	Elías	11
31777111	Pepe	41
31555222	Rosa	29
31717171	Juana	11
12002003	Manuela	08

<u>CodProvincia</u>	Provincia
11	Cádiz
29	Málaga
08	Madrid
41	Sevilla

<u>DNI</u>	Nombre	CodProvincia	Provincia
31777999	Elías	11	Cádiz
31777111	Pepe	41	Sevilla
31555222	Rosa	29	Málaga
31717171	Juana	11	Cádiz
12002003	Manuela	08	Madrid

Forma normal de Boyce Codd

Una tabla está en Forma normal Boyce Codd solo si está en 3FN y los atributos que dependen de otros, lo hacen de claves candidatas.

Aquellas tablas en la que todos sus atributos forman parte de la clave primaria, están en FNBC.

Para normalizar a FNBC tendremos que descomponer la tabla inicial en dos, siendo cuidadosos para evitar la pérdida de información en dicha descomposición.

Ejemplo



Empleados

<u>id</u>	Nombre	ProyectoID	NombreProyecto	Horas Trabajadas
1	Luca	12354	Proyecto A	101
2	Jonny	55975	Proyecto B	5000
3	Ángel	12354	Proyecto A	50
4	Ana	12354	Proyecto A	120

Cumple con las formas normales¿?

Empleados

<u>id</u>	Nombre	ProyectoID	NombreProyecto	Horas
1	Lucas	12354	Proyecto A	101
2	Jonny	55975	Proyecto B	5000
3	Ángel	12354	Proyecto A	50
4	Ana	12354	Proyecto A	120

1FN, 2FN: OK

3FN y BCFN: NO OK

Ejemplo - SOLUCIÓN



Empleados

<u>id</u>	Nombre
1	Lucas
2	Jonny
3	Ángel
4	Ana

Proyectos

<u>cod</u>	NombreProyecto
12354	Proyecto A
55975	Proyecto B

Trabajos

<u>idEmpleado</u>	<u>codProyecto</u>	horas
1	12354	101
2	55975	5000
3	12354	50
4	12354	120

Diccionario de Datos

Un diccionario de datos o catálogo contiene la descripción de los datos de la base de datos.

Plantilla:

- Campo:
 - Nombre del campo
- Tipo.
 - El tipo de datos y rango de valores que puede tomar
- NULL.
 - Si el campo puede ser nulo
- Predeterminado.
 - Valor predeterminado si procede. Si se trata de un valor automático indicarlo.
- Descripción (Unidad - Ejemplo).
 - Descripción del atributo (Indicar la Unidad si procede y Un ejemplo de un dato real que puede contener)

Tipo. El tipo de datos y rango de valores que puede tomar

- Char (indicar el número máximo de caracteres) – Cadena de texto de tamaño fijo
- Varchar(número máximo de caracteres) – Cadena de texto de tamaño variable
- LongText (textos muy largos)
- Date - Fecha
- int - Número entero (indicar número máximo de dígitos si procede)
- Float - Número decimal (indicar número máximo de dígitos si procede)
- Booleano – Verdadero o Falso
- Blob (para tipos binarios como fotos o vídeos)

CLIENTE

Campo	Tipo (Tamaño)	NULL	Predeterminado	Descripción (Unidad - Ejemplo)
<u>Dni</u>	<u>Char(9)</u>	NO		Número de DNI vigente con letra incluida. ("31034792H")
nombre	<u>Varchar(30)</u>	NO		Nombre de persona ("Hugo")
apellidos	<u>Varchar(40)</u>	NO		Apellidos de persona "Pérez López de la Sierra"
<u>tlf</u>	<u>Char(9)</u>	NO		Número de teléfono de persona "928222526"
peso	<u>int(3)</u>	SI	<u>NULL</u>	Peso de persona (Kg - "70")
altura	<u>int(3)</u>	SI	<u>NULL</u>	Altura de persona (cm - "180")
<u>fechaNac</u>	Date	SI	<u>NULL</u>	Fecha en la que nace una persona ("25-08-2011")
precio	<u>Float</u>	SI	<u>NULL</u>	Precio de algo (€ - 12,5)
desempleado	Boolean	SI	<u>NULL</u>	Indica si la persona no esta trabajando. ("True")
foto	<u>Blob</u>	SI	<u>NULL</u>	Guarda la foto carnet de la persona (imagen)
<u>descripcion</u>	<u>Longtext</u>	SI	<u>NULL</u>	Descripción de la vida laboral de la persona ("Empezó a trabajar en 1995 en.....")
<u>codProvincia</u>	<u>int</u>	NO	Automático	Código de la provincia ("12")
cargo	<u>Varchar(9)</u>	NO	"Empleado"	Cargo de la persona dentro de la empresa {"Director", "Encargado", "Empleado"}

RESTRICCIONES

- dni es Clave primaria
- codProvincia es clave foránea de PROVINCIA.cod

Además hay que decir cuales son las claves primarias y si las foráneas aceptan nulos, una pista de eso segundo es si tienen un 0 en las participaciones.