



# UD1

## DISEÑO CONCEPTUAL DE BASES DE DATOS

MP\_0484  
Bases de Datos

1.5 Modelo  
relacional II

## Introducción

A partir de ahora, vamos a desarrollar las técnicas básicas para llegar al modelo relacional: la primera de ellas se basará en la transformación del modelo E/R a relacional y la segunda en la aplicación del proceso de formalización.

En cualquiera de los casos, estamos ante un proceso diseñado específicamente para el diseño de bases de datos, hecho que no sucedía con el anterior modelo E/R el cual es considerado de propósito general (si bien es usado en buena medida en el ámbito de las bases de datos).

## Paso del modelo E/R al modelo relacional

### Entidades

Cada una de las entidades del modelo E/R pasan a ser tablas del modelo relacional (figura 1), siguiendo las siguientes consideraciones:

- Las entidades pasan a ser tablas.
- Los atributos pasan a ser columnas.
- Los atributos clave primaria, clave candidata, clave alternativa, tienen el mismo sentido.
- El atributo clave ajena (foránea) es un concepto nuevo que va a estar relacionado con las relaciones y entidades del diagrama E/R.

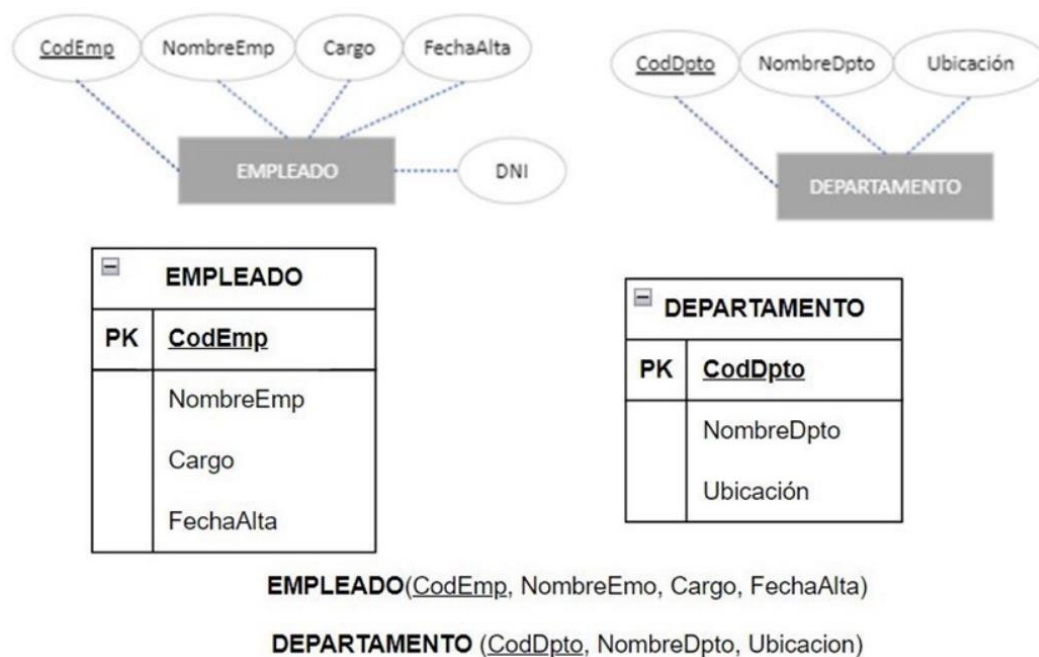


Figura 1. Ejemplo de paso de entidades a modelo relacional

### Relaciones de uno a varios o de 0 a varios

En las relaciones de uno a varios, la entidad a la que llega  $n$  hereda la clave de la entidad 1 como clave ajena (figura 2). Si la relación es de uno a varios, hay que exigir integridad referencial; si la relación es de 0 a  $n$ , no hay que exigir integridad referencial.

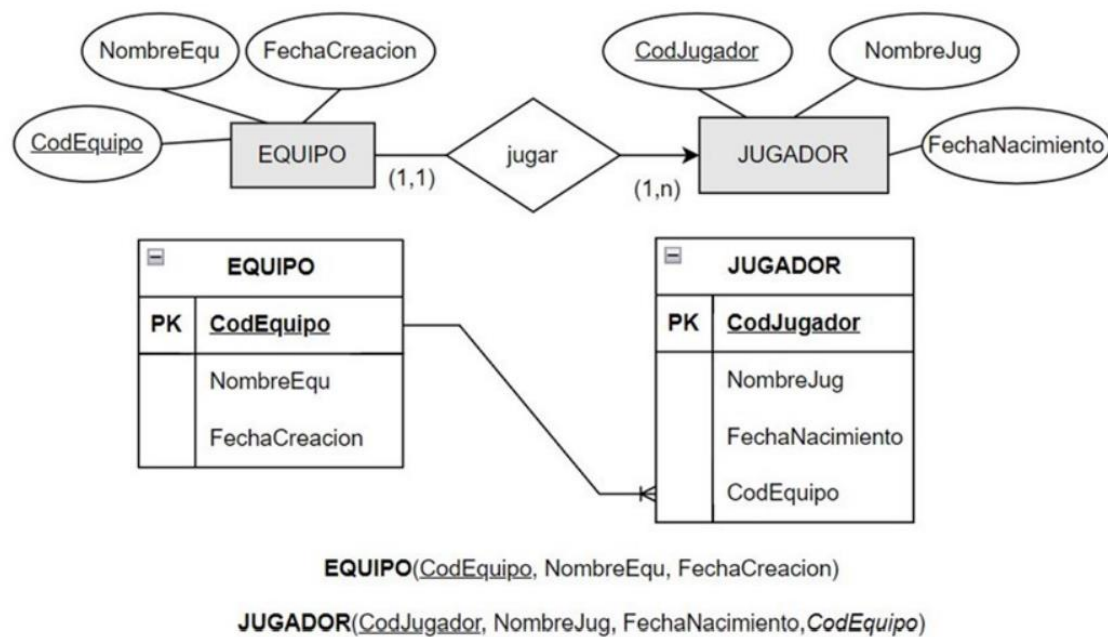


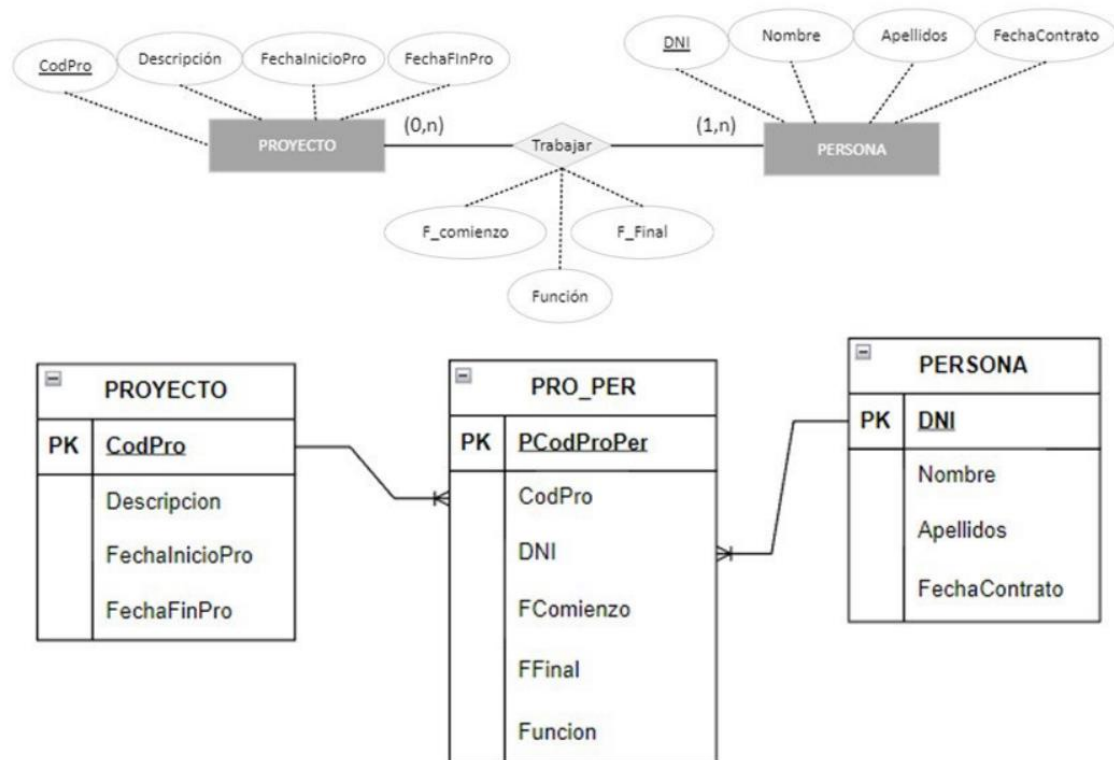
Figura 2. Transformación de relación de uno a  $n$

### Relación de $n$ a $n$ , de varios a varios

En las relaciones de  $n$  a  $n$  (de varios a varios, de  $n:m$ ), las entidades pasan a ser tablas y siempre aparece una tabla nueva entre medio de ellas, que contiene los atributos que aparecen en el diagrama, las claves ajenas de las dos entidades que había, y o añadimos una clave primaria para esta tabla o asignamos entre los campos aquel campo o grupo de campos que sirva como clave. Se debe exigir integridad referencial.

Tal y como se muestra en la figura 3, se han añadido a la tabla intermedia los atributos que estaban en la relación, como por ejemplo las fechas de inicio y final. En este caso, esta decisión de diseño nos permite mantener un histórico de las personas que han pasado por un determinado proyecto, así como de los proyectos en los que ha participado una determinada persona.

De ahí la importancia que se le ha dado en el pasado a ubicar atributos en las relaciones  $n$  a  $n$ , pues ahora pueden ser trasladados al modelo relacional de manera trivial.

Figura 3. Transformación de relación de  $n$  a  $n$ 

### Relación reflexiva

En las relaciones reflexivas de 1 a  $n$ , la entidad hereda la clave de sí misma (figura 4). Si tuviésemos una reflexiva de  $n$  a  $n$ , habría que crear una tabla nueva, que heredaría dos veces la clave.

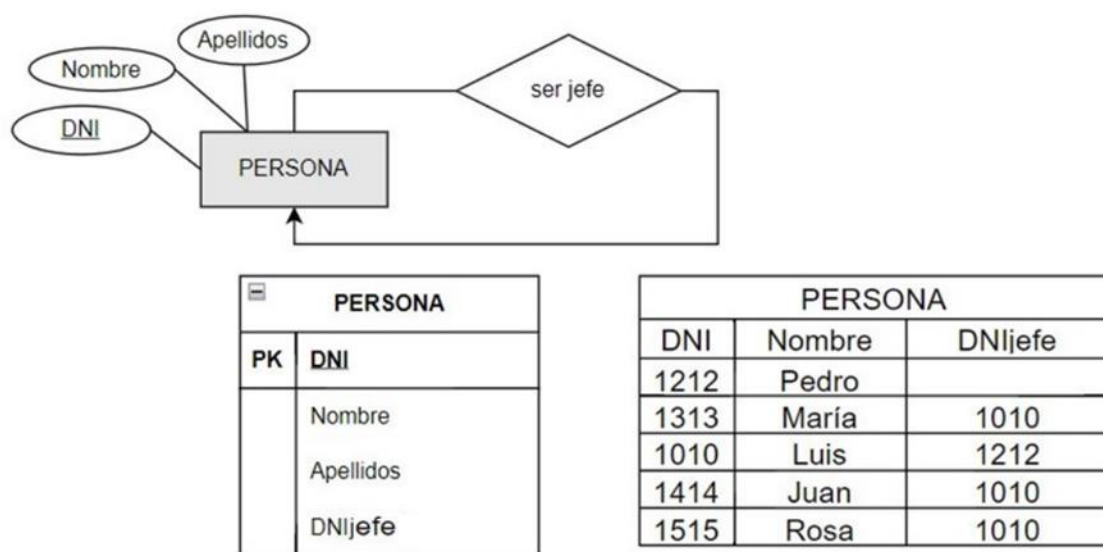


Figura 4. Transformación de relación reflexiva

### Relaciones de 1 a 1 o de 0 a 1

En el caso de tener dos entidades relacionadas entre sí, cada una con su atributo clave deberemos valorar cuál de las entidades tiene más peso en ese diagrama, cuál se relaciona con más entidades, cuál contiene datos que se van a consultar con mayor probabilidad, dicha entidad la trataríamos como la fuerte y la otra como la débil (figura 5). Tendríamos que heredar la clave de la entidad fuerte en la entidad débil como clave ajena.

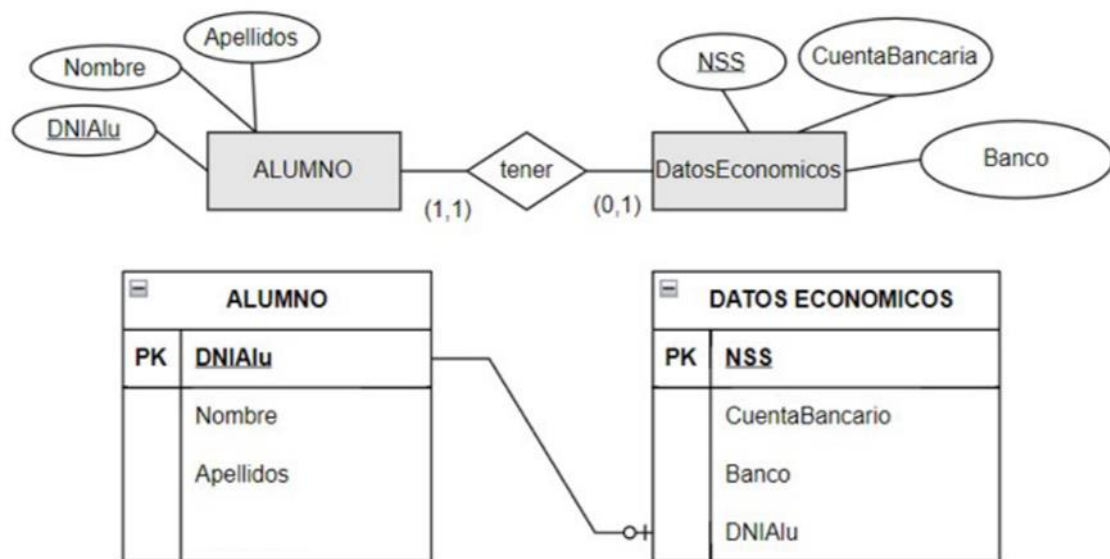


Figura 5. Transformación de relación uno a uno.

## Normalización

Las formas normales son parte de una teoría de normalización iniciada por Edgar F. Codd y desarrollada por otros autores como Boyce y Fagin. En 1970, Codd definió la primera forma normal, seguida por la segunda, tercera, Boyce-Codd, cuarta y quinta forma normal. Una tabla puede estar en la primera forma normal sin estar en la segunda, pero no al revés.

Es decir, las formas normales superiores son más restrictivas, cumpliendo cada una de ellas todas las condiciones de las anteriores. Aunque la teoría de formas normales es completamente matemática, aquí se presenta de manera más intuitiva. Muchos diseñadores consideran suficiente alcanzar la forma Boyce-Codd, ya que la cuarta y, especialmente, la quinta forma normal es muy debatida.

Algunos argumentan que hay BD que, al estar en la quinta forma normal, son peores que en la tercera. Sin embargo, se considera esencial que el diseño alcance al menos la forma normal de Boyce-Codd. Antes de empezar a definir las primeras tres formas normales (FN) vamos a hablar de las dependencias funcionales. Se dice que un conjunto de atributos (Y) depende funcionalmente de otro conjunto de atributos (X) si para cada valor de X hay un único valor posible para Y ( $X \rightarrow Y$ ).

Por ejemplo, el nombre de una persona depende funcionalmente del DNI, es decir, para un DNI concreto solo hay un nombre posible. Si miramos la dependencia entre un departamento de una empresa y el DNI del empleado, podemos afirmar que el departamento no depende del DNI, ya que para un mismo departamento hay más de un DNI. Un atributo A es dependiente funcional completo de un conjunto de atributos B si y solo si A es dependiente funcional de B y no de ningún subconjunto propio de B. En una tabla EMPLEADOS, el NSS y el nombre del empleado provocan una dependencia funcional sobre el resto de los atributos como apellido, dirección, etc. Pero no es completa, ya que solo con el NSS también se produce una dependencia funcional.

Se puede afirmar que NSS, produce una dependencia funcional completa sobre apellidos, dirección. Se dice que existe una dependencia funcional elemental cuando X e Y forman una dependencia funcional completa y además Y es un atributo único. Existe también la posibilidad de tener una dependencia funcional transitiva cuando ocurre que el atributo A determina a un atributo B y el B determina a un atributo C, entonces A transitivamente determina al C. Esto implica que existe una relación indirecta entre A y C a través de B.

### Primera forma normal (1FN)

Una tabla se encuentra en 1FN si impide que un atributo de una tupla pueda tomar más de un valor. Como se puede observar en la figura 6, en la primera tabla Trabajador la tupla Andrea tiene asociados dos departamentos distintos. En la 1FN se solventa esto y la clave principal sería DNI y Departamento: ambos campos formarían la clave principal.

#### TRABAJADOR

<u>DNI</u>	Nombre	Departamento
12121212B	María	Mantenimiento
13131313R	Andrea	Dirección Secretaría
14141414S	Luis	Secretaría

#### TRABAJADOR 1FN

<u>DNI</u>	Nombre	<u>Departamento</u>
12121212B	María	Mantenimiento
13131313R	Andrea	Dirección
13131313R	Andrea	Secretaría
14141414S	Luis	Secretaría

Figura 6. Tabla Trabajador y su 1FN

### Segunda forma normal (2FN)

Una tabla está en 2FN si estando en 1FN se cumple que cada atributo que no sea clave depende funcionalmente de forma completa de cualquiera de las claves. El atributo o conjunto de atributos que forman la clave deben hacer dependientes al resto de atributos de esta; si hay atributos que solo dependen de una parte de la clave, esos atributos formarían otra tabla.

En el ejemplo de la figura 7, teniendo una clave primaria compuesta (DNI, CodCurso), hay que fijarse si el resto de los atributos dependen funcionalmente de todos los campos clave. En este ejemplo, el Nombre y el Apellido dependen solo del DNI, por lo tanto, eso hay que sacarlo a otra tabla.



<u>DNI</u>	<u>CodCurso</u>	Nombre	Apellido	Nota
12121219A	34	Pedro	Valiente	9
12121219A	25	Pedro	Valiente	8
3457775G	34	Ana	Fernández	6
5674378J	25	Sara	Crespo	7
5674378J	34	Sara	Crespo	6

## ALUMNO

<u>DNI</u>	Nombre	Apellido
12121219A	Pedro	Valiente
3457775G	Ana	Fernández
5674378J	Sara	Crespo

## ASISTENCIA

<u>DNI</u>	<u>CodCurso</u>	Nota
12121219A	34	9
12121219A	25	8
3457775G	34	6
5674378J	25	7
5674378J	34	6

Figura 7. Asistencia 2FN

## Tercera forma normal (3FN)

Una tabla está en 3FN cuando estando en 2FN, ninguno de los atributos de la tabla depende funcionalmente de otro que no sea clave. En el caso propuesto en la figura 8, Provincia depende de CodProvincia.

Para solucionar esto, se debe separar la información en dos tablas: por un lado, la información de cada alumno a nivel de datos personales, así como CodProvincia. Adicionalmente, se debe crear una nueva tabla (Provincia) que relacione los CodProvincia con sus respectivas provincias. De esta manera se mejora la integridad de la información y se reduce el espacio consumido en memoria.



## ALUMNO

Provincia	<u>DNI</u>	Nombre	Apellido1	CodProvincia
Palencia	12121349A	Salvador	Velasco	34
Palencia	12121219A	Pedro	Valiente	34
Valladolid	3457775G	Ana	Fernández	47
Valladolid	5674378J	Sara	Crespo	47
Barcelona	3456858S	Marina	Serrat	08

## ALUMNOS

<u>DNI</u>	Nombre	Apellido1	CodProvincia
12121349A	Salvador	Velasco	34
12121219A	Pedro	Valiente	34
3457775G	Ana	Fernández	47
5674378J	Sara	Crespo	47
3456858S	Marina	Serrat	08

## PROVINCIA

<u>CodProvincia</u>	Provincia
34	Palencia
47	Valladolid
08	Barcelona

Figura 8. Alumno 3FN