

Tema 5: Normalización de bases de datos relacionales.

5.1. La teoría de la normalización.

El proceso de diseño de una base de datos consiste en representar una determinada parcela del mundo real (o universo del discurso) mediante los objetos que proporciona el modelo de datos que estamos utilizando, aplicando para ello las reglas de dicho modelo, como prohibición de un determinado tipo de asociaciones o posibilidad de incluir ciertas restricciones. Cuando se diseña una base de datos mediante el modelo relacional, así como ocurre con otros modelos, tenemos distintas alternativas, es decir, podemos obtener diferentes soluciones plasmadas en distintos esquemas relacionales, no todos ellos equivalentes, ya que unos van a representar la realidad mejor que otros.

Veremos la teoría de la normalización, que permite afrontar el problema del diseño de bases de datos relacionales de una manera rigurosa y objetiva, estudiando la forma de llevar a cabo la normalización de un esquema relacional.

Se va a ver qué propiedades debe tener un esquema relacional para representar adecuadamente la realidad, y cuáles son los problemas que se pueden derivar de un diseño inadecuado.

Veamos un ejemplo: En la figura 1 se muestra una relación denominada *Pedidos*, que almacena datos sobre los pedidos efectuados por los clientes (*RefPed* y *FecPed*) y los artículos solicitados en cada uno de los pedidos (*CodArt*, *DesArt*, *CantArt* y *PVP**Art*). Por cada pedido se indica su referencia y fecha y por cada artículo solicitado, su código, descripción, número de unidades solicitadas y precio unitario. La clave primaria de esta relación está formada por la concatenación de los atributos *RefPed* y *CodArt*. Esta relación presenta diversos problemas, entre los cuales se encuentran los siguientes:

- Gran cantidad de redundancia, ya que la fecha del pedido se repite si en ese pedido han sido solicitados varios artículos. De igual forma, si un artículo ha sido solicitado en varios pedidos, aparecen repetidos los atributos *DesArt* y *PVP**Art*.
- Anomalías de inserción, ya que no sería posible incluir en la base de datos información sobre algún artículo que no estuviese incluido en ningún pedido, al formar el atributo *RefPed* parte de la clave primaria de la relación. Recordemos que por la regla de integridad de la entidad ningún atributo que forme parte de la clave primaria de una relación puede tomar valor nulo. Por otro lado, la inserción de un pedido que incluyese varios artículos obligaría a incluir varias tuplas en la base de datos.

- Anomalías de modificación, ya que si quisiésemos modificar el precio o la descripción de un artículo solicitado en varios pedidos, lo tendríamos que hacer en varias tuplas de la relación para no generar inconsistencias. Lo mismo ocurriría si quisiésemos modificar la fecha de un pedido en el que se han solicitado varios artículos.
- Anomalías de borrado, ya que si quisiéramos eliminar un pedido de la base de datos y uno de los artículos en él incluido solo es solicitado en dicho pedido, desaparecerían también los datos de ese artículo de la base de datos. Además, la eliminación de un pedido de la base de datos implicaría borrar de esta relación tantas tuplas como artículos sean solicitados en el mismo. De manera análoga, si queremos eliminar un artículo de la base de datos y ocurre que ese artículo es solicitado en varios pedidos, habría que borrar una línea por cada uno de esos pedidos. Además, si un artículo solo está solicitado en un pedido, la eliminación de dicho artículo implicaría también eliminar los datos del pedido de la base de datos.

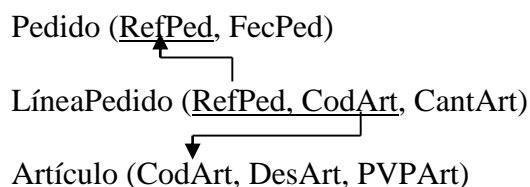
RefPed	FecPed	CodArt	DesArt	CantArt	PVPArt
P0001	16/02/2014	A0043	Bolígrafo azul fino	10	0,78
P0001	16/02/2014	A0078	Bolígrafo rojo normal	12	1,05
P0002	18/02/2014	A0043	Bolígrafo azul fino	5	0,78
P0003	23/02/2014	A0075	Lápiz 2B	20	0,55
P0004	25/02/2014	A0012	Goma de borrar	15	0,15
P0004	25/02/2014	A0043	Bolígrafo azul fino	5	0,78
P0004	25/02/2014	A0089	Sacapuntas	50	0,25

Figura 1: Relación *Pedidos*.

Esta relación presenta todos estos problemas debido a que se almacenan hechos distintos (pedidos y artículos) en una misma relación.

Existe un método formal aplicable a todo esquema relacional que nos permite determinar si un esquema relacional se adecua a la realidad y, en caso de que no sea así, nos indica cómo transformarlo para conseguir que el mismo sea un reflejo lo más fiel posible del mundo real.

El método formal al que me refiero en el párrafo anterior es la teoría de la normalización, con la que se consiguen esquemas relacionales exentos de redundancias y que, por tanto, no presentan los problemas indicados anteriormente. Si hubiésemos aplicado el proceso de normalización a la anterior relación, habríamos obtenido el siguiente esquema relacional:



En este esquema relacional se han almacenado hechos distintos en relaciones diferentes.

La teoría de la normalización puede definirse como una técnica formal para organizar datos que nos ayuda a determinar qué está equivocado en un diseño y nos enseña la manera de corregirlo.

La teoría de la normalización se basa en el concepto de forma normal. Un esquema relacional se encuentra en una determinada forma normal si cumple un conjunto concreto de restricciones.

Existen en total seis formas normales:

- Primera forma normal (1FN).
- Segunda forma normal (2FN).
- Tercera forma normal (3FN).
- Forma normal de Boyce/Codd (FNBC).
- Cuarta forma normal (4FN).
- Quinta forma normal (5FN).

Como se puede observar en la siguiente figura, de todo el universo de relaciones solo algunas están en 1FN, de estas solo algunas se encuentran en 2FN, de estas únicamente una parte está en 3FN, y así sucesivamente; es decir, la 2FN impone más restricciones que la 1FN, la 3FN más que la 2FN, etc., siendo la 5FN la que impone restricciones más fuertes.

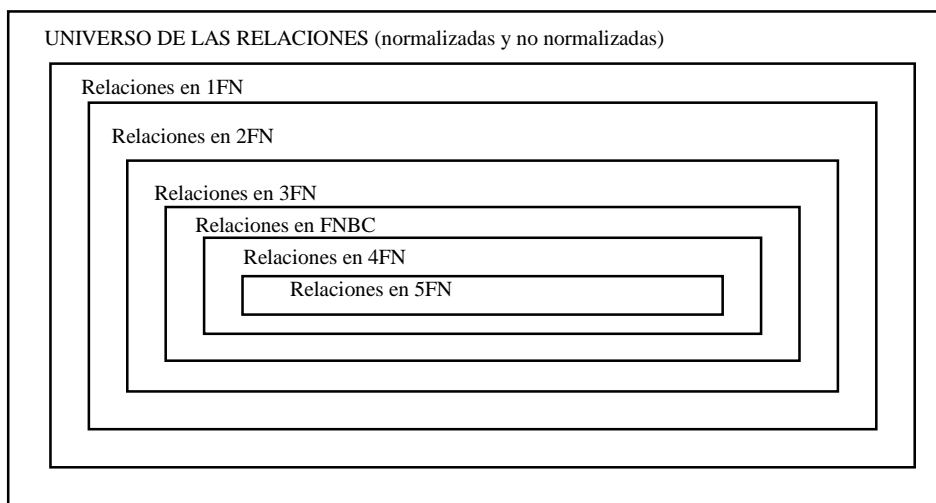


Figura 2: Representación esquemática de las formas normales.

El estudio de las formas normales requiere el conocimiento previo del concepto de dependencia funcional, que va a ser explicado a continuación.

5.2. Dependencias funcionales. Tipos.

Existen varios tipos de dependencias funcionales, que se van a exponer en los siguientes subapartados.

5.2.1. Dependencia funcional.

Dados los subconjuntos de atributos X e Y de una relación, se dice que Y depende funcionalmente de X o que X determina o implica a Y si y solo si cada valor de X tiene asociado un único valor de Y . Representamos esta dependencia de la siguiente forma:

$$X \rightarrow Y$$

Por ejemplo, en la relación Artículo (CodArt, DesArt, PVPArt), el código del artículo determina la descripción del mismo y su precio puesto que dado un código de artículo (que identifica a un artículo), ese artículo tendrá una sola descripción y un único precio:

$$\text{CodArt} \rightarrow \text{DesArt}$$

$$\text{CodArt} \rightarrow \text{PVPArt}$$

Por su parte, en la relación LíneaPedido (RefPed, CodArt, CantArt) la pareja de atributos (*RefPed*, *CodArt*) determinan *CantArt*, porque dado un pedido identificado por su referencia (*RefPed*) y un artículo identificado por su código (*CodArt*) solo hay una cantidad,

es decir, en una línea de pedido solo se solicita un número determinado de unidades de un artículo dado. Por ello:

$$(\text{RefPed}, \text{CodArt}) \rightarrow \text{CantArt}$$

Una herramienta útil para mostrar las dependencias funcionales es el grafo o diagrama de dependencias funcionales, mediante el cual se representa un conjunto de atributos y las dependencias funcionales existentes entre ellos. En estos grafos aparecen los nombres de los atributos unidos por flechas, las cuales indican las dependencias funcionales.

En la figura 3 se muestran las dependencias funcionales para las relaciones *Artículo* y *LíneaPedido*, y también para la relación *Pedido* (RefPed, FecPed), en la cual existe la siguiente dependencia funcional:

$$\text{RefPed} \rightarrow \text{FecPed}$$

porque para un pedido identificado por su referencia hay una única fecha en que este se realizó.

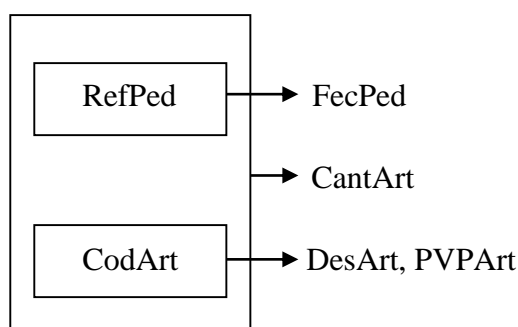


Figura 3: Diagrama de dependencias funcionales.

5.2.2. Dependencia funcional completa.

Dados los subconjuntos de atributos X e Y de una relación (constando X de varios atributos), se dice que Y tiene una dependencia funcional plena o completa de X si depende funcionalmente de X , pero no depende de ningún subconjunto de X , lo que se representa por:

$$X \Rightarrow Y$$

Por ejemplo, en la relación *LíneaPedido* (RefPed, CodArt, CantArt) que ya hemos estudiado, nos podemos plantear si la dependencia funcional

$$(\text{RefPed}, \text{CodArt}) \rightarrow \text{CantArt}$$

es plena o completa. Lo será si el atributo *CantArt* depende del par de atributos (*RefPed*, *CodArt*) y no de uno de ellos por separado, es decir, si no son verdad las dos siguientes dependencias funcionales:

$$\text{RefPed} \rightarrow \text{CantArt}$$

$$\text{CodArt} \rightarrow \text{CantArt}$$

La primera de estas dos dependencias funcionales no es cierta porque dado un pedido identificado por su referencia (*RefPed*) puede haber varias cantidades de artículos solicitadas; de hecho, ocurrirá esto siempre que el pedido incluya varias líneas de pedido, es decir, siempre que en el pedido se soliciten dos o más artículos diferentes.

Con respecto a la segunda dependencia funcional, esta será verdadera si dado un artículo identificado por su código (*CodArt*) solo puede haber una cantidad solicitada de ese artículo. Sin embargo, dado que un artículo puede ser solicitado en diferentes pedidos y en cada uno de dichos pedidos se pueden solicitar cantidades diversas de dicho artículo, la dependencia funcional indicada tampoco es cierta.

Como las dos dependencias funcionales parciales analizadas son falsas, podemos decir que la dependencia funcional

$$(\text{RefPed}, \text{CodArt}) \rightarrow \text{CantArt}$$

es plena o completa, lo que se representa:

$$(\text{RefPed}, \text{CodArt}) \Rightarrow \text{CantArt}$$

En un diagrama de dependencias funcionales, como el de la figura 3, las dependencias funcionales totales se representan partiendo la flecha de un rectángulo en cuyo interior aparece más de un atributo.

5.2.3. Dependencia funcional mutua o interdependencia.

Si en una relación se dan las dependencias funcionales $X \rightarrow Y$ e $Y \rightarrow X$ simultáneamente, entonces se dice que entre los atributos X e Y hay una dependencia funcional mutua o interdependencia, y se representa así:

$$X \leftrightarrow Y$$

Por ejemplo, dada la relación Libro (CodLib, ISBN, Título, Páginas, Editorial), se dan las dependencias funcionales:

$$\text{CodLib} \rightarrow \text{ISBN}$$

$$\text{ISBN} \rightarrow \text{CodLib}$$

porque dado un libro identificado por un código, este tiene un solo ISBN y, de manera análoga, dado un libro identificado por su ISBN, tiene un código único. Esto se debe a que ambos atributos son claves candidatas. Pues bien, al darse las dependencias funcionales en

ambos sentidos, se puede decir que los atributos *CodLib* e *ISBN* mantienen una dependencia funcional mutua o interdependencia:

$$\text{CodLib} \leftrightarrow \text{ISBN}$$

5.2.4. Dependencia funcional transitiva.

Sea una relación $R(X, Y, Z)$ en la que existen las siguientes dependencias funcionales:

$$X \rightarrow Y$$

$$Y \rightarrow Z$$

$$Y \not\rightarrow X$$

Se dice entonces que Z tiene una dependencia funcional transitiva respecto de X a través de Y y se representa:

$$X \twoheadrightarrow Z$$

Consideremos la relación Coche (Matrícula, Marca, Modelo, Color)

En esta relación se dan las siguientes dependencias funcionales:

$$\text{Matrícula} \rightarrow \text{Marca}$$

$$\text{Matrícula} \rightarrow \text{Modelo}$$

$$\text{Matrícula} \rightarrow \text{Color}$$

porque dado un vehículo identificado por su matrícula, este tiene una sola marca (Renault o Ford, por ejemplo), un único modelo (Megane o Fiesta, por ejemplo) y un solo color.

Además, también se da la siguiente dependencia funcional:

$$\text{Modelo} \rightarrow \text{Marca}$$

puesto que dado un modelo de vehículo, le corresponde una única marca (por ejemplo, todos los Meganes son Renault).

Todas estas dependencias funcionales también se pueden representar mediante el siguiente grafo:

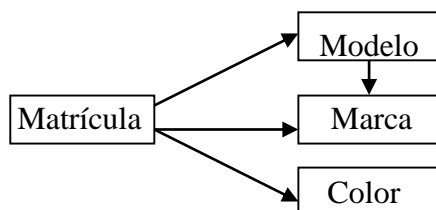


Figura 4: Diagrama de dependencias funcionales.

Dado que se dan las siguientes dependencias funcionales:

$\text{Matrícula} \rightarrow \text{Modelo}$

$\text{Modelo} \rightarrow \text{Marca}$

$\text{Modelo} \not\rightarrow \text{Matrícula}$

se puede decir que *Marca* tiene una dependencia funcional transitiva respecto de *Matrícula* a través de *Modelo* y se representa:

$\text{Matrícula} \twoheadrightarrow \text{Marca}$

5.3. Primera forma normal (1FN).

Una relación se encuentra en 1FN si cada uno de sus componentes es atómico, es decir, si no presenta grupos repetitivos. Podemos definir un grupo repetitivo como un atributo o conjunto de atributos con valores múltiples.

Para eliminar un grupo repetitivo, es decir, para pasar una relación a 1FN, se deben eliminar de la relación de partida los atributos que constituyen el grupo repetitivo, creando una nueva relación con la clave primaria de la relación original y los atributos del grupo repetitivo. La clave de la nueva relación suele consistir en la concatenación de la clave primaria de la relación original con la clave del grupo repetitivo.

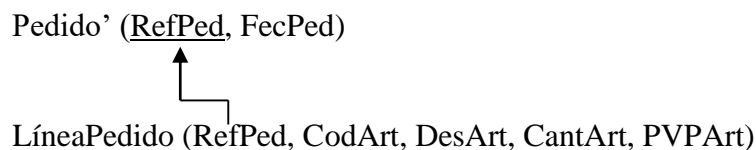
Por ejemplo, disponemos de la siguiente relación:

Pedido (*RefPed*, *FecPed*, *CodArt*, *DesArt*, *CantArt*, *PVPART*)

en la que se muestra la información relativa a un pedido (*RefPed* y *FecPed*) y a los artículos que en él se solicitan, indicando por cada uno de los artículos pedidos, su código (*CodArt*), descripción (*DesArt*), número de unidades solicitadas (*CantArt*) y precio (*PVPART*). Los atributos *CodArt*, *DesArt*, *CantArt* y *PVPART* constituyen un grupo repetitivo porque si consideramos como clave primaria de la relación el atributo *RefPed*, para un mismo pedido identificado por su referencia (*RefPed*), se pueden solicitar varios artículos, cada uno de los cuales tendrá su *CodArt*, *DesArt*, *CantArt* y *PVPART*.

Al existir un grupo repetitivo, esta relación no se encuentra en 1FN. Para pasarla a 1FN, hemos de eliminar de la relación *Pedido* los cuatro atributos que constituyen el grupo repetitivo, generando así una nueva relación, que podemos llamar *Pedido'*. Además, tendremos que crear una nueva relación con los cuatro atributos del grupo repetitivo más la clave primaria de la relación de partida (*RefPed*), siendo la clave de esta nueva relación

surgida la clave de la relación de partida (*RefPed*) más el atributo clave del grupo repetitivo (*CodArt*). Nos quedaría por tanto el siguiente esquema relacional en 1FN:



5.4. Segunda forma normal (2FN).

Una relación se encuentra en 2FN si estando en 1FN, cada atributo que no forme parte de una clave candidata mantiene una dependencia funcional total respecto de dicha clave candidata, es decir, todo atributo no clave debe depender de toda la clave y no solo de parte de ella.

Para pasar una relación a 2FN, se debe eliminar de la relación el atributo que genera la dependencia parcial y crear una nueva relación con ese atributo y con el/los atributo/s de que depende como clave primaria.

Siempre que una relación en 1FN presenta una clave primaria compuesta por un solo atributo, ya se encuentra automáticamente en 2FN. También se encontrarán en 2FN las relaciones en 1FN que no presenten atributos no clave.

Si tomamos el esquema relacional obtenido en el apartado anterior, la relación *Pedido'* ya se encuentra en 2FN por estar en 1FN y tener una clave primaria que consta de un solo atributo. En lo que respecta a la relación *LíneaPedido*, para que esta se encuentre en 2FN se deberá cumplir que los atributos no clave, es decir, *DesArt*, *CantArt* y *PVPArt*, dependan de la totalidad de la clave, es decir, que tengan una dependencia funcional total respecto del par de atributos (*RefPed*, *CodArt*):

$$(\text{RefPed}, \text{CodArt}) \Rightarrow \text{DesArt}$$

$$(\text{RefPed}, \text{CodArt}) \Rightarrow \text{CantArt}$$

$$(\text{RefPed}, \text{CodArt}) \Rightarrow \text{PVPArt}$$

La primera dependencia funcional total no es cierta porque la descripción del artículo solo depende del código del mismo y no del pedido en el que se solicite, es decir, es verdad la dependencia parcial $\text{CodArt} \rightarrow \text{DesArt}$. Por este motivo, ya podemos afirmar que la relación *LíneaPedido* no se encuentra en 2FN.

La segunda dependencia funcional total sí es verdadera porque la cantidad que de un artículo se solicita en un pedido no viene determinada únicamente por el pedido en cuestión o el artículo que se solicita, sino por ambos hechos. Dicho de otra forma, $\text{RefPed} \not\rightarrow \text{CantArt}$, dado que para un pedido se pueden solicitar diferentes cantidades de diversos artículos; de modo análogo, $\text{CodArt} \not\rightarrow \text{CantArt}$, porque un artículo puede ser solicitado en varios pedidos, para cada uno de los cuales se solicitará una determinada cantidad de dicho artículo.

La tercera dependencia funcional total será verdadera en caso de que el precio de los artículos no siempre sea el mismo para todos los pedidos. Si, como es más habitual, consideramos que el precio de los artículos es el mismo para todos los clientes y pedidos, entonces se cumple la dependencia funcional $\text{CodArt} \rightarrow \text{PVPArt}$, con lo que no se cumple tampoco que el atributo no clave *PVPArt* depende de la totalidad de la clave, por lo que este sería otro motivo por el cual la relación no se encuentra en 2FN.

Otra manera de observar si una relación se encuentra en 2FN es analizar su diagrama de dependencias funcionales.

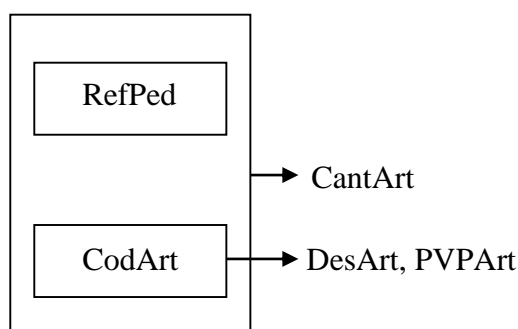
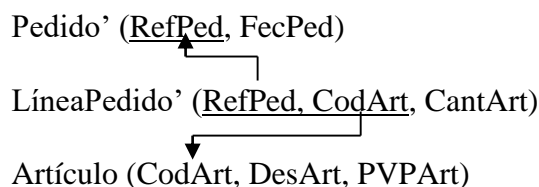


Figura 5: Diagrama de dependencias funcionales.

Si en él hay alguna flecha que atraviesa el espacio dejado entre los atributos que componen la clave compuesta y los atributos no clave, entonces es que hay algún atributo que no depende de la totalidad de la clave, como ocurre en este caso con los atributos *DesArt* y *PVPArt*, por lo que la relación correspondiente no se encuentra en 2FN,

Pasemos por tanto la relación *LíneaPedido* a 2FN: Para ello hemos de eliminar de esta relación cada uno de los atributos que no dependen de la totalidad de la clave (*DesArt* y *PVPArt*) y crear en principio por cada uno de ellos una nueva relación con cada uno de estos atributos y aquel del que dependen (*CodArt*). Como en este caso, ambos atributos dependen del mismo (*CodArt*), no es necesario crear una relación nueva por cada uno de ellos, sino solamente una que albergue los dos atributos y aquel del que dependen como clave. El esquema relacional en 2FN nos quedará como sigue:



5.5. Tercera forma normal (3FN).

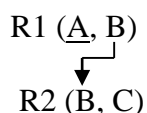
Una relación se encuentra en 3FN si estando en 2FN, cada atributo que no forme parte de una clave candidata depende directamente de ella, es decir, si no hay dependencias transitivas. Toda relación en 2FN con menos de dos atributos no clave ya se encuentra automáticamente en 3FN.

Para eliminar las dependencias transitivas se elimina de la relación que no está en 3FN el atributo que genera la dependencia transitiva y se crea una tabla con el/los atributo/s transitivo/s y el atributo del que depende o por medio del cual mantiene/n la transitividad.

Así, si tenemos la relación $R(\underline{A}, B, C)$ con las siguientes dependencias funcionales:

$$\nexists A \rightarrow B \rightarrow C$$

existe una dependencia funcional transitiva de C respecto de A porque C depende de B y no directamente de A , que es la clave. Se solucionaría este problema, es decir, se pasaría la relación a 3FN, con el siguiente esquema relacional:



Consideremos la relación *Coche* (Matrícula, Marca, Modelo, Color) a la que ya nos referimos en el apartado 5.2.4. En esta relación, *Marca* tiene una dependencia funcional transitiva respecto de *Matrícula* a través de *Modelo* porque:

$$\nexists \text{Matrícula} \rightarrow \text{Modelo} \rightarrow \text{Marca}$$

Por este motivo, esta relación no se encuentra en 3FN y para pasarla a esta forma normal lo que hemos de hacer es eliminar de la relación *Coche* el atributo transitivo (*Marca*) y crear una nueva relación con dicho atributo más aquel del que depende, siendo este último la clave primaria de esta nueva relación. El resultado sería el siguiente esquema relacional.

