

Bases de Datos: Formas Normales (Optimizar Tablas)

Autor: Carlos Carmona

Este artículo está bajo licencia CopyLeft. Se permite pues su libre distribución con la única condición de incluir el nombre del autor y un enlace a la web original.

La normalización es un proceso que pretende conseguir tablas con una estructura óptima y eficaz. El proceso de normalización está basado en lograr la independencia de los datos respecto a las aplicaciones que los usan.

Antes de empezar el proceso, se han de conocer las tablas que intervendrán y las relaciones que las unen. Si no se conocen a partir del análisis previo, se buscan todos los nombres (sustantivos) que han sido empleados en la definición del problema. Algunos de esos nombres serán las entidades, otros dependerán de ellas y serán los atributos. Otros no formarán parte ni de las entidades ni de los atributos, son parte del lenguaje necesario para describir el problema a solucionar mediante la creación de una base de datos.

Ejemplo

<<...a cada cliente, al pasar por Caja... se marcan por la caja registradora los artículos que ha comprado. Con los datos de los artículos se hace una factura por el importe total de las mercancías adquiridas que se imprime y se entrega al cliente. Los datos de la factura se almacenan para su posterior tratamiento informático que comprende...>>

Las tablas son sustantivos, por lo que tenemos los siguientes: cliente, Caja, caja registradora, artículos, datos de los artículos, factura, importe total, mercancías adquiridas, datos de la factura. De estos nombres, algunos son atributos de otros: datos de los artículos y artículos, datos de la factura, importe total y factura. De cada cliente no se piden datos, por lo que aunque sea una tabla, si no se necesitan sus datos, no se creará esa entidad. Caja con mayúscula se refiere a un objeto con el que se realizan procesos, por lo que no se necesita almacenar información de ellos. De cada una de las cajas registradoras, tal vez se necesite para las facturas, el número de caja, por lo que se considera una entidad más. Mercancías adquiridas y artículos que ha comprado son sinónimos, por lo que solo se tratará de artículos.

Las tablas encontradas tras el análisis son: artículos, factura y caja registradora. Caja registradora se puede considerar un atributo de factura, por lo que tenemos dos tablas.

Las relaciones se pueden encontrar conociendo todos los verbos que aparecen en la definición del problema. Se eliminan aquellos verbos que son necesarios para el lenguaje y se buscan aquellos que implican dos o más entidades (sustantivos) que ya se han encontrado.

En el ejemplo han aparecido los verbos: pasar, se marcan, ha comprado, se hace una factura, imprime, entrega, almacena. De estos verbos, los que asocian entidades son: marcar, comprar. Los verbos pasar, hacer factura, imprimir, entregar, almacenar, se refieren a procesos que se van a realizar, no a asociaciones entre entidades.

Se han obtenido las siguientes entidades con sus relaciones: clientes, comprar artículos y marcar artículos en factura. Como no se necesitan los datos de los clientes, queda la relación marcada (en la caja registradora) que une las tablas artículos, y factura. La operación marcar en la caja registradora significa que los artículos se incluyen en una factura que se entregará al cliente para su liquidación, consiguiéndose obtener el modelo entidad-relación siguiente:

[imagen perdida]

Hay cinco niveles de normalización, siendo cada vez más complejo el proceso de obtención de tablas normalizadas. Para bases de datos relativamente sencillas se puede terminar la normalización en el tercer nivel o tercera forma normal.

El proceso de normalización se basa en la descomposición sin pérdida de las tablas que están en una forma normal inferior, obteniéndose una forma normal superior. El proceso de descomposición sin pérdida, significa que se ha de dividir o descomponer la tabla en otras con menor cantidad de atributos sin que haya pérdida de información.

Formas normales y dependencias funcionales.

Primera Forma Normal o 1FN:

La Primera Forma Normal, o 1FN, es la más elemental de todas. Una tabla está en 1FN si el valor que contiene un atributo de un registro, un campo, es único y elemental. En cada uno de los atributos sólo se puede incluir un dato, aunque sea compuesto, pero no se pueden incluir una lista de datos. Por ejemplo, no se pueden incluir en el atributo Dirección el domicilio habitual y el de vacaciones; habría que crear dos registros que se diferenciarán por el atributo Dirección:

Tabla de una base de datos						
NIF	Ape	Nom	Dir	CPost	Pobl	Prov
1	García	Francisco	C/Marín 16	33698	Oviedo	Asturias
2	Sanchez	Luisa	C/Tenerías 34 C/Ramorta 65	85458 54585	Cigales Bueu	Valladolid Pontevedra

Esta tabla no está en 1FN, ya que el cliente con Id 2 tiene dos direcciones. Para poder tener esta tabla en 1FN se hace el siguiente cambio:

Tabla de una base de datos						
NIF	Ape	Nom	Dir	CPost	Pobl	Prov
1	García	Francisco	C/Marín 16	33698	Oviedo	Asturias
2	Sanchez	Luisa	C/Tenerías 34	85458	Cigales	Valladolid
2	Sanchez	Luisa	C/Ramorta 65	54585	Bueu	Pontevedra

Segunda Forma Normal o 2FN:

Se dice que un atributo o conjunto de atributos tiene dependencia funcional de otro u otros si a cada uno de los primeros le corresponde sólo uno de los segundos.

Por ejemplo, hay una dependencia funcional entre CIF y el atributo Razón Social, ya que a cada CIF le corresponde una única Razón Social.

Una tabla está en Segunda Forma Normal o 2FN cuando está en 1FN y todo atributo que no pertenece a la clave primaria tiene una dependencia funcional de la clave completa y no de parte de ella. Luego, si la clave principal está formada por un solo atributo y ya está en 1FN, ya estará en 2FN.

Para transformar una tabla con dependencias funcionales, cuya clave está formada por más de un campo, en una tabla en 2FN se necesitan crear tablas nuevas para eliminar las dependencias funcionales, las tablas nuevas tendrán los atributos que dependen funcionalmente de la clave y los que forman la parte de la clave de la que dependen. Una vez creadas las nuevas tablas, se eliminan de la tabla primera los atributos que tenían dependencias funcionales.

En el ejemplo anterior, tanto el nombre como los apellidos dependen del NIF. Se crea una nueva tabla que contiene los atributos: NIF, nombre y apellidos, eliminándose de la tabla cliente los atributos nombre y apellidos, quedando las siguientes tablas:

Tabla en segunda forma normal				
NIF	Dir	CPost	Pobl	Prov
1	C/ Marín nº16	33698	Oviedo	Asturias
2	C/ Tenerías nº34	85458	Cigales	Valladolid
2	C/ Ramorta nº65	54585	Bueu	Pontevedra

Tabla en segunda forma normal

NIF	Ape	Nom
-----	-----	-----

1	García	Francisco
2	Sanchez	Luisa

Tercera Forma Normal o 3FN:

Se dice que hay dependencia funcional transitiva entre dos atributos cuando un atributo que no pertenece a la clave primaria permite conocer el valor de otro atributo.

Por ejemplo: dada la tabla clientes, entre los atributos provincia y prefijo telefónico hay una dependencia funcional transitiva, ya que el primero permite conocer el valor del segundo.

Una tabla está en Tercera Forma Normal o 3FN si está en 2FN y no existen atributos que no pertenezcan a la clave primaria que puedan ser conocidos mediante otro atributo que no forma parte de la clave primaria, es decir, no hay dependencias funcionales transitivas.

Siguiendo con el ejemplo anterior, cuando hay dependencias funcionales transitivas, se crea una nueva tabla con los atributos que tienen dependencia funcional transitiva, eliminándose el atributo dependiente de la tabla original.

Si nos fijamos en esta tabla:

Tabla en segunda forma normal				
NIF	Dir	CPost	Pobl	Prov
1	C/ Marín nº16	33698	Oviedo	Asturias
2	C/ Tenerías nº34	85458	Cigales	Valladolid
2	C/ Ramorta nº65	54585	Bueu	Pontevedra

La dirección, la población y la provincia dependen del código postal, que no forma parte de la clave primaria. Descomponiendo sin pérdida una vez más, obtenemos estas dos tablas:

Tabla en tercera forma normal

NIF	Dir
-----	-----

1	C/ Marín nº16
2	C/ Tenerías nº34
2	C/ Ramorta nº65

Tabla en tercera forma normal			
CPost	Dir	Pobl	Prov
33698	C/ Marín nº16	Oviedo	Asturias
85458	C/ Tenerías nº34	Cigales	Valladolid
54585	C/ Ramorta nº65	Bueu	Pontevedra

Para solucionar algunos problemas de dependencias funcionales, que no se podían resolver solo con la normalización en 3FN, se han propuesto tres formas normales adicionales. La normalización más allá de 3FN queda al juicio del diseñador de la base de datos. A partir de esa forma normal, la eliminación de dependencias funcionales pasa por la creación de tablas con multitud de información redundante, con un posible aumento de tamaño, por lo que se ha de optar entre una optimización del diseño y una optimización del tamaño. Llegándose a diversas soluciones de compromiso entre ambos parámetros. Salvo excepciones, con la 3FN o a lo sumo, la FNBC (que veremos a continuación) es más que suficiente, y llevar la normalización más allá será más perjudicial que beneficioso.

Forma Normal de Boyce-Codd o FNBC:

Una tabla está en Forma Normal de Boyce-Codd o FNBC si solo existen dependencias funcionales elementales que dependan de la clave primaria o de cualquier clave alternativa. Si la clave primaria está formada por un solo atributo y está en 3FN, ya está en FNBC.

Un ejemplo típico para mostrar una tabla que, estando en 3FN, mantiene dependencias funcionales, sin relación con el ejemplo seguido hasta este momento, es una tabla que posee los atributos dirección, código postal y población, suponiendo que a poblaciones diferentes le corresponden códigos postales distintos.

Tabla en tercera forma normal		
CPost	Dir	Pobl
30009	C/ Pantano Camarillas nº16	Murcia
48596	Av. Buenos Aires nº12	Madrid

En este caso hay dependencia entre el código postal y la población, ya que, conocido el código postal se puede conocer la población, y conocida la dirección y la población, se conoce el código postal. Para transformar la tabla en una tabla en FNBC se crea una tabla de códigos postales y poblaciones, eliminando de la tabla original la población, obteniéndose dos tablas, una con los atributos dirección y código postal y otra con el código postal y la población:

Tabla en forma normal de Boyce-Codd	
CPost	Dir
30009	C/ Pantano Camarillas nº16
48596	Av. Buenos Aires nº12

Tabla en forma normal de Boyce-Codd	
CPost	Pobl
30009	Murcia
48596	Madrid

Cuarta Forma Normal o 4FN:

Existe dependencia funcional multivalorada o de múltiples valores si, dados tres atributos de una tabla, si para cada valor del primer atributo existen múltiples valores en el segundo atributo y no hay ninguna relación entre el tercer atributo y el primero, a no ser a través del segundo atributo.

Una tabla está en Cuarta Forma Normal o 4FN si está en FNBC y las únicas dependencias funcionales multivaloradas que existen son las dependencias funcionales de la clave con los atributos que no forman parte de la misma. Estas dependencias multievaluadas de la clave con los atributos que no forman parte de la misma son dependencias triviales, por lo que algunos autores dicen que no existen dependencias multievaluadas en 4FN.

Supongamos que los atributos de la tabla transporte son conductor, tipo de vehículo y tipo de carga, formando los tres campos la clave primaria. A cada conductor se le puede asignar un vehículo u otro y cada vehículo puede transportar varios tipos de carga.

Tabla que no esta en cuarta forma normal
Transporte

Conductor	Tipo Vehículo	Tipo Carga
-----------	---------------	------------

Juan	Furgoneta	Perecederos
Marcos	Furgoneta	Perecederos
Juan	Furgoneta	Muebles
Marcos	Furgoneta	Muebles
Juan	Camión	Mudanza
Marcos	Camión	Mudanza

Con estas condiciones, los conductores son independientes de la carga; el tipo de vehículos depende del conductor y el tipo de vehículo depende de la carga. En este caso hay dependencias funcionales multivaloradas, ya que algunos atributos que forman la clave dependen de otro atributo que también la forman.

Para conseguir que esta tabla esté en 4FN se necesita crear dos nuevas tablas en lugar de la tabla actual, manteniéndose en cada una de ellas una dependencia múltiple. La primera tabla tendrá los atributos conductor y tipo de vehículo y la segunda, tipo de vehículo y tipo de carga. De este modo la tabla en 4FN debido a que la clave primaria de ambas tablas son todos los campos que la forman. Resultado:

Tabla en cuarta forma normal
Tipo Vehículo Tipo Carga

Furgoneta	Perecederos
Furgoneta	Perecederos
Furgoneta	Muebles
Furgoneta	Muebles
Camión	Mudanza
Camión	Mudanza

Tabla en cuarta forma normal	
Conductor	Tipo Vehículo
Juan	Furgoneta
Marcos	Furgoneta
Juan	Furgoneta
Marcos	Furgoneta
Juan	Camión
Marcos	Camión

Quinta Forma Normal o 5FN:

Se dice que hay dependencia de JOIN, de unión o de producto si una tabla tiene dependencia de *unión con varias de sus *proyecciones y se puede obtener la tabla por medio de la unión de dichas proyecciones.

Proyección

Creación de una tabla cuyos elementos forman un subconjunto de una tabla dada. Se incluyen todas las filas y algunas columnas.

Unión

Formar, a partir de dos tablas, una nueva con todos los campos de una de ellas y los registros de ambas, excepto los repetidos.

Ambas tablas han de tener el mismo grado y las mismas columnas.

Una tabla esta en Quinta Forma Normal (5FN) o Forma Normal de Proyección-Unión si está en 4FN y las únicas dependencias que existen son las dependencias de unión de una tabla con sus proyecciones relacionándose entre las distintas proyecciones mediante la clave primaria o cualquier clave alternativa. La 5FN se emplea cuando en una misma tabla tenemos mucha información redundante, con pocos atributos o cuando una tabla posee una gran cantidad de atributos y se hace por ello inmanejable.

Para conseguir que una tabla 4FN con gran cantidad de atributos esté en 5FN, se parte la tabla original en tantas tablas como se desee, teniendo cada una de ellas en común con las demás los campos que forman la clave primaria en la tabla original.

Ejemplo para el caso de una tabla que posee una gran cantidad de atributos:

Id	Datos Familiares				Datos Profesionales			Tabla Datos Personales			Datos Clínicos	
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12
1												

En este caso tenemos una empresa donde se guardan los datos personales, familiares, profesionales y clínicos de cada empleado en una única tabla llamada Empleados. Si esta tabla está ya en 4FN, se puede partir en las tablas empleados-personal, empleados-familia, empleados-profesional, empleados-clínicos; de este modo, la velocidad de acceso y la gestión de datos por cada departamento de la empresa se simplifica, al no tenerse que crear ningún tipo de restricción sobre determinados atributos que no han de ser vistos por el personal que no los necesite.

El resultado sería:

Tabla en quinta forma normal			
Id		Datos Familiares	
1	D1	D2	D3

Tabla en quinta forma normal
Id Datos Profesionales

1	D4	D5	D6
---	----	----	----

Tabla en quinta forma normal
Id Datos Personales

1	D7	D8	D9
---	----	----	----

Tabla en quinta forma normal
Id Datos Clínicos

1	D10	D11	D12
---	-----	-----	-----

Ejemplo para el caso de una tabla que posee mucha información redundante, con pocos atributos:

Tabla que no esta en quinta forma normal
Biblioteca

Título	Fecha	Socio
T1	FT	S1
T2	FU	S2
T3	FV	S1
T4	FG	S4
T1	FH	S3
T2	FT	S4
T3	FV	S3

Si se tiene una tabla de préstamo de libros de una biblioteca, con los atributos título, fecha de préstamo y número de socios que ha tomado prestado el libro, existen multitud de registros que se crean diariamente en esa tabla, pero para cada libro o para cada socio habrá pocos registros, con lo que una consulta para esa tabla como: ¿Cuáles son los libros leídos por un determinado socio?, puede tener una velocidad de respuesta elevada. Si esta tabla se parte en las tablas título-fecha, título-socio y socio-fecha, cualquier consulta similar a la anterior tendrá un tiempo de respuesta tolerable, y cuando sea necesario, se podrán realizar consultas que impliquen los datos de las tres tablas.

El resultado sería pues:

Tabla en quinta forma normal
Título-Fecha

Título	Fecha
T1	FT
T2	FU
T3	FV
T4	FG
T1	FH
T2	FT
T3	FV

Tabla en quinta forma normal
Título-Socio

Título	Socio
T1	S1
T2	S2
T3	S1
T4	S4
T1	S3
T2	S4
T3	S3

Tabla en quinta forma normal
Fecha-Socio

Fecha	Socio
FT	S1
FU	S2
FV	S1
FG	S4
FH	S3
FT	S4
FV	S3

Y con esto concluyo el artículo, espero haber resuelto todas las dudas posibles sobre optimización de tablas mediante las formas normales.

PD: Mi más sincera enhorabuena a todos los que hayáis tenido el valor de leerlos todo el artículo...

Este artículo está bajo licencia CopyLeft. Se permite pues su libre distribución con la única condición de incluir el nombre del autor y un enlace a la web original.