

Normalización

Sitio: [Agencia de Aprendizaje a lo largo de la Vida](#)
Curso: Administración de Base de Datos 1° F
Libro: Normalización

Imprimido por: MARIO DAVID GONZALEZ BENITEZ
Día: miércoles, 28 de agosto de 2024, 19:23

Tabla de contenidos

1. Normalización

1.1. Ejemplo de Normalización

¿Qué es la normalización?



Lo importante en el uso y manejo de una base de datos es [evitar la redundancia de datos y garantizar el fácil acceso](#) a los mismos. Para lograrlo es que Codd [[Edgar Frank Codd](#). Científico informático inglés conocido por sus aportes a la teoría de bases de datos Relacionales] enuncia la metodología de “normalización”.

La normalización es un mecanismo que permite que un conjunto de tablas cumpla una serie de propiedades que eviten:

- [Redundancia](#) de datos
- [Anomalías](#) de actualización
- [Pérdidas](#) de Integridad de datos

El objetivo de la normalización es construir una BD que minimice la redundancia de información; para ello es necesario reagrupar los atributos de cada tabla del modelo.

El proceso de diseño genera redundancia de información, y en algunos casos esa redundancia es necesaria.

Por ejemplo:

Cuando se resuelve una [relación](#) muchos a muchos del esquema lógico, se genera una nueva entidad con las claves primarias de las entidades relacionadas. Este tipo de redundancia es la mínima y necesaria y no afecta al proceso de normalización. Por el contrario, existen otros casos de redundancia, que generan [anomalías](#) de actualización cuando se opera sobre la BD.

Estas anomalías de clasifican en tres grupos: de inserción, de borrado y de modificación.

Anomalías de Inserción

Supongamos que contamos con una tabla que almacena los datos laborales de los empleados de una empresa.

EMPLEADO (idEmpleado Nombre Salario FIngreso idDepto NomDepto)

Si se agrega un nuevo empleado, se debe indicar toda la información, incluyendo repetir el nombre del departamento donde trabaja, aunque sea un departamento ya existente en la tabla. Se debe tener especial cuidado en describir al departamento de forma similar a lo que se hubiese hecho anteriormente.

Por ejemplo:

(dEmpleado	Nombre	Salario	FIngreso	idDepto	NomDepto
1	Marcos	9500	2006/08/12	1	Contabilidad
2	Julian	11000	2007/12/05	2	Eléctrica y Computación
3	Rosario	12000	2010/11/15	1	Contabilidad
4	Maria	11000	2003/01/02	3	Ventas
5	Mirta	9500	2010/09/16	2	Computación y Eléctrica

Se observa que la información almacenada es incorrecta, ya que el departamento con idDepto 2 no puede tener dos nombres diferentes. Esta situación se produce al generar redundancia de información. Cada vez que se agrega un empleado a un departamento existente, debe ingresarse el nombre del departamento, lo cual es innecesario.

Anomalías de Borrado

Las de borrado se producen en situaciones similares.

Supongamos que en el [ejemplo anterior](#) se elimina al empleado María, que trabaja en el departamento de Ventas. Al borrar esa tupla, en la misma operación se borra información del departamento donde trabaja, y como era la única empleada registrada para ese departamento, se pierde “Ventas” como departamento de la organización.

Anomalías de Modificación

Supongamos que el departamento de Electrónica y Computación del ejemplo anterior, se modifica y pasa a llamarse departamento de Tecnología. Este cambio debe producirse necesariamente sobre cada una de las tuplas que corresponden a empleados del viejo departamento de Electrónica y Computación, lo que indica que se debe cambiar varios registros de la tabla. Además, si por cualquier inconveniente no se produjeran todos los cambios, quedarían tuplas donde al iddepto =2 y el nombre del departamento sería Tecnología.

- La metodología está constituida por un conjunto de normas, las primeras fueron numeradas, existiendo una que recibió el nombre de Boyce- Codd.
- Las formas normales más usadas son 1ºFN, 2ºFN y 3ºFN.
- Esta [metodología](#) es totalmente lineal ya que para que una [relación](#) esté en 2ºFN debe estar antes en 1ºFN y así sucesivamente.

¿Se puede aplicar directamente la metodología?

La respuesta es Sí, teniendo en claro algunos conceptos, como ser:

- **Dependencia funcional:** La dependencia funcional se establece entre Atributos en una [Relación](#). Por lo tanto, dados dos atributos [a](#) y [b](#) pertenecientes a la [Relación Q](#), se define dependencia funcional en [Q](#) si al valor de [b](#) está asociado a cada valor de [a](#).
- **Valor atómico:** los valores de los atributos [no](#) son multivalorados
- **Dependencia de los atributos no clave con la Clave Primaria:** cuando la clave primaria es compuesta puede suceder que algún atributo no clave dependa de uno de los atributos de la clave primaria, pero no de la totalidad de ellos. Cuando la clave primaria está compuesta por un solo atributo la dependencia es verdadera.

Primera Forma normal

Para decir que una [relación](#) está en 1ºFN se deben cumplir las siguientes condiciones:

- Debe existir una clave principal (primaria).
- Todos los dominios simples contienen valores atómicos.

Segunda Forma normal

Para decir que una [relación](#) está en 2ºFN se deben cumplir las siguientes condiciones:

- Debe estar en 1ºFN.
- Todos los atributos no clave dependen funcionalmente de la clave primaria.

Tercera Forma normal

Una tabla está en tercera forma normal (3FN) si, y solo si, está en 2FN y, además, cada atributo que no forma parte de la clave primaria no depende transitivamente de la clave.

Ejemplo de normalización



Un dato sin normalizar no cumple con ninguna regla de normalización. Para explicar con un ejemplo en qué consiste cada una de las reglas, vamos a considerar los datos de la siguiente tabla:

ordenes (id_orden, fecha, id_cliente, nom_cliente, estado, num_art, nom_art, cant, precio)

Ordenes								
<u>Id_orden</u>	Fecha	<u>Id_cliente</u>	<u>Nom_cliente</u>	Estado	<u>Num_art</u>	<u>nom_art</u>	<u>cant</u>	Precio
2301	23/02/11	101	Martin	Caracas	3786	Red	3	35,00
2301	23/02/11	101	Martin	Caracas	4011	Raqueta	6	65,00
2301	23/02/11	101	Martin	Caracas	9132	Paq-3	8	4,75
2302	25/02/11	107	Herman	Coro	5794	Paq-6	4	5,00
2303	27/02/11	110	Pedro	Maracay	4011	Raqueta	2	65,00
2303	27/02/11	110	Pedro	Maracay	3141	Funda	2	10,00

Primera Formal normal (1FN)

Al examinar estos registros, podemos darnos cuenta que contienen un grupo repetido para NUM_ART, NOM_ART, CANT y PRECIO.
La 1FN prohíbe los grupos repetidos, por lo tanto tenemos que convertir a la primera forma normal.

Los pasos a seguir son:

- Tenemos que eliminar los grupos repetidos.
- Tenemos que crear una nueva tabla con la PK de la tabla base y el grupo repetido.

Los registros quedan ahora conformados en dos tablas que llamaremos *ordenes* y *articulos_ordenes*

ordenes (id_orden, fecha, id_cliente, nom_cliente, estado)
Articulos_ordenes (id_orden, num_art, nom_art, cant, precio)

Ordenes				
<u>Id_orden</u>	Fecha	<u>Id_cliente</u>	<u>Nom_cliente</u>	Estado
2301	23/02/11	101	Martin	Caracas
2302	25/02/11	107	Herman	Coro
2303	27/02/11	110	Pedro	Maracay

Articulos_ordenes				
<u>Id_orden</u>	<u>Num_art</u>	<u>nom_art</u>	<u>cant</u>	Precio
2301	3786	Red	3	35,00
2301	4011	Raqueta	6	65,00
2301	9132	Paq-3	8	4,75
2302	5794	Paq-6	4	5,00
2303	4011	Raqueta	2	65,00
2303	3141	Funda	2	10,00

Segunda Formal normal (2FN)

Ahora procederemos a aplicar la segunda formal normal, es decir, tenemos que eliminar cualquier columna no llave que no dependa de la llave primaria de la tabla.

Los pasos a seguir son:

- Determinar cuáles columnas que no son llave no dependen de la llave primaria de la tabla.
- Eliminar esas columnas de la tabla base.
- Crear una segunda tabla con esas columnas y la(s) columna(s) de la PK de la cual dependen.

- La tabla [ordenes](#) está en 2FN. Cualquier valor único de [id_orden](#) determina un sólo valor para cada columna. Por lo tanto, todas las columnas son dependientes de la llave primaria [id_orden](#) .
- Por su parte, la tabla [articulos_ordenes](#) no se encuentra en 2FN ya que las columnas [precio](#) y [norm_art](#) son dependientes de [num_art](#), pero no son dependientes de [id_orden](#). Lo que haremos a continuación es eliminar estas columnas de la tabla [articulos_ordenes](#) y crear una tabla [articulos](#) con dichas columnas y la llave primaria de la que dependen.

Las tablas quedan ahora de la siguiente manera:

[Articulos_ordenes](#) (id_orden, num_art, cant)

Articulos_ordenes		
<u>Id_orden</u>	<u>Num_art</u>	<u>cant</u>
2301	3786	3
2301	4011	6
2301	9132	8
2302	5794	4
2303	4011	2
2303	3141	2

[Articulos](#) (num_art, nom_art, precio)

Articulos		
<u>Num_art</u>	<u>nom_art</u>	<u>Precio</u>
3786	Red	35,00
4011	Raqueta	65,00
9132	Paq-3	4,75
5794	Paq-6	5,00
3141	Funda	10,00

Tercera Forma normal (3FN)

La tercera forma normal nos dice que tenemos que eliminar cualquier columna no llave que sea dependiente de otra columna no llave.

Los pasos a seguir son:

- Determinar las columnas que son dependientes de otra columna no llave.
- Eliminar esas columnas de la tabla base.
- Crear una segunda tabla con esas columnas y con la columna no llave de la cual son dependientes.

Al observar las tablas que hemos creado, nos damos cuenta que tanto la tabla:

- [articulos](#), como la tabla [articulos_ordenes](#) se encuentran en 3FN.
- Sin embargo la tabla [ordenes](#) no lo está, ya que [nom_cliente](#) y [estado](#) son dependientes de [id_cliente](#), y esta columna no es la llave primaria.
- Para normalizar esta tabla, moveremos las columnas no llave y la columna llave de la cual dependen dentro de una nueva tabla [clientes](#).
- Las nuevas tablas [clientes](#) y [ordenes](#) se muestran a continuación.

[ordenes](#) (id_orden, fecha, id_cliente)

Ordenes		
<u>Id_orden</u>	<u>Fecha</u>	<u>Id_cliente</u>
2301	23/02/11	101
2302	25/02/11	107
2303	27/02/11	110

[Clientes](#) (id_cliente, nom_cliente, estado)

<u>Id_cliente</u>	<u>Nom_cliente</u>	<u>Estado</u>
101	Martin	Caracas
107	Herman	Coro
110	Pedro	Maracay



Por lo tanto, la base de datos queda de la siguiente manera:

- Ordenes ([id_orden](#), [fecha](#), [id_cliente](#))
- Clientes ([id_cliente](#), [nom_cliente](#), [estado](#))
- Articulos ([num_art](#), [nom_art](#), [precio](#))
- Articulos_ordenes ([id_orden](#), [num_art](#), [cant](#))