

Normalización de bases de datos relacionales

La normalización de una base de datos relacional es una herramienta que nos ayuda a evitar o reducir la redundancia de información y evita posibles anomalías al momento de hacer inserciones, actualizaciones y borrados, aparte nos ayuda a resolver los problemas de lógica para las acciones antes descritas.

Breve recordatorio del modelo relacional

El modelo relacional de bases de datos, se basa en un modelo formal especificado de acuerdo a la teoría de conjuntos. Una base de datos relacional puede considerarse como un conjunto de relaciones, entidades o tablas de la forma $R(A_1, \dots, A_n)$, donde R es el nombre de la relación, tabla o entidad que se define por una serie de atributos o campos A_i .

Sobre las tablas relacionales se pueden definir diferentes restricciones. La integridad de entidad es una restricción que nos indica que cada entidad representada por un registro tiene que ser diferente de las demás en su relación, es decir, debe haber algunos atributos cuyos conjuntos de valores identifiquen unívocamente los registros, a esto se le conoce como **llave primaria (Primary Key)** en la tabla. También nos toparemos con la integridad referencial, la cual indica que una clave ajena solo debe contener valores válidos en la tabla de la cual depende o se hace referencia (tabla papá), esta clave ajena es conocida como **llave foránea (Foreign Key)**.

El proceso de normalización

El proceso de normalización consiste en comprobar en secuencia si el esquema original está en **1FN, 2FN y 3FN**, analizando las dependencias funcionales en cada paso. Una base de datos normalizada hasta la 3FN, está más que suficientemente normalizada para trabajar adecuadamente. A continuación, se describe brevemente en que consiste cada una de las formas normales.

Primera Forma Normal: (1FN) Incluye la eliminación de todos los grupos repetidos (atributos o campos que generan un grupo dentro de los datos para ese registro). Una tabla está en Primera Forma Normal si:

- Todos los atributos son atómicos. Un atributo es atómico si los elementos del dominio son simples e indivisibles.
- La tabla contiene una clave primaria única.
- La clave primaria no contiene atributos nulos.

- No debe existir variación en el número de columnas.
- Los Campos no clave deben identificarse por la clave (Dependencia Funcional)

Segunda Forma Normal: (2FN) Asegura que todas las columnas, atributos o campos que no son llave sean completamente dependientes de la llave primaria (PK) (que no se generen grupos y conlleven a duplicar los registros o generar grupos repetidos).

Debemos seguir los siguientes pasos:

1. Crear tablas separadas para aquellos grupos de datos que se aplican a varios registros.
2. Relacionar estas tablas mediante una clave externa.

Sabremos si nuestra base de datos tiene en la segunda forma normal si ésta previamente cumple con las normas de la Primera forma Normal y **si sus atributos no principales dependen de forma completa de la clave principal**. Es decir que no existen dependencias parciales.

Tercera Forma Normal: (3FN) Elimina cualquier dependencia transitiva. Una dependencia transitiva es aquella en la cual las columnas que no son llaves dependen de otras columnas que tampoco son llave.

Debemos considerar los siguientes puntos:

1. Eliminar aquellos campos que no dependan de la llave.
2. Ninguna columna puede depender de una columna que no tenga una llave.
3. No puede haber datos derivados.

Veamos un ejemplo completo: Tenemos una empresa donde los puestos de trabajo que desempeñan los empleados están regulados para definir el salario, de modo que las condiciones salariales o sueldos que reciben dichos empleados están determinadas por el puesto, es decir, dependiendo del puesto eso es lo que gana el empleado.

Para ejemplificar las formas normales, se ha creado el siguiente esquema relacional.

Se cuenta con la entidad o tabla EMPLEADOS la cual contiene los siguientes atributos (no_emp – es el número de empleado en la empresa, nombre – nombre completo del empleado (obvio este atributo debe de estar descompuesto en nombre, paterno y materno, puesto – puesto en el que trabaja el empleado, salario – salario que se le paga al empleado,

email – correos del empleado) con el campo o atributo no_emp como llave primaria PK (primary key).

No_emp	Nombre	Paterno	Puesto	Salario	Email
111	Juan	Pérez	Jefe de Área	3000	juanp@ecn.es; jefe2@ecn.es
222	José	Sánchez	Administrativo	1500	jsanchez@ecn.es
333	Ana	Díaz	Administrativo	1500	adiaz@ecn.es; ana32@gmail.com
...

Tabla: EMPLEADOS

Como se puede observar en la tabla EMPLEADOS, algunos de ellos tienen dos correos, este campo forma un grupo repetido, y como lo indica la (1FN) “Incluye la eliminación de todos los grupos repetidos” (atributos o campos que generan un grupo dentro de los datos para ese registro).

La 1FN nos dice, que una tabla está en 1FN si sus atributos contienen valores atómicos (**que sean únicos, que en ése atributo solo exista un solo dato o valor**). En el ejemplo, podemos ver que el atributo email puede contener más de un valor, por lo que viola 1FN, éste valor no es atómico, no es único, único es el nombre, el paterno, el materno, el salario, etc, etc.

En general, tenemos una relación R con clave primaria K. Si un atributo M viola la condición de 1FN, tenemos dos opciones para solucionar y normalizar la tabla.

Solución 1: Duplicar los registros con valores repetidos

Lo cual no es correcto, dado que tendríamos registros duplicados por cada elemento (correo) que tenga el empleado, si uno de ellos tuviera 5 correos, tendríamos 5 veces el mismo registro para poder tener registrado sus correos.

No_emp	Nombre	Paterno	Puesto	salario	Email
111	Juan	Pérez	Jefe de Área	3000	juanp@ecn.es
111	Juan	Pérez	Jefe de Área	3000	jefe2@ecn.es
222	José	Sánchez	Administrativo	1500	jsanchez@ecn.es
333	Ana	Díaz	Administrativo	1500	adiaz@ecn.es

333	Ana	Díaz	Administrativo	1500	ana32@gmail.com
...
Tabla: EMPLEADOS					

Solución 2: Separar el atributo que viola 1FN en una nueva tabla, lo cual nos lleva a la 2FN.

Lo que hacemos, es crear una nueva tabla, en la nueva tabla incluimos el campo PK de la tabla origen (papá) y el campo o atributo que genera el conjunto de datos en dicha tabla, en éste caso el correo que pueden ser N correos. La tabla papá ya no contiene el campo que provoca la falla de la 1FN, así evitamos la duplicidad de los registros.

Siguiendo el ejemplo y tomando en consideración lo que indica la 1FN, al quitar el atributo Email de la tabla Empleados tendríamos el siguiente esquema para la nueva tabla EMPLEADOS'(b)

no_emp	Nombre	Paterno	Puesto	Salario
111	Juan	Pérez	Jefe de Área	3000
222	José	Sánchez	Administrativo	1500
333	Ana	Díaz	Administrativo	1500
...
Tabla: EMPLEADOS				

En la nueva tabla de EMPLEADOS y no tenemos el campo Email y tampoco la duplicidad de registros.

Y además tendríamos una nueva tabla EMAILS con los campos o atributos (no_emp – para saber que empleado es y el campo email – el correo que tiene el empleado): el campo o atributo no_emp, en la tabla CORREOS es una llave foránea y aquí si es válido que se duplique. La relación que se tiene entre estas tablas es del tipo 1:N, es decir, un empleado puede tener uno o más registros, en nuestro caso, el empleado puede tener 1 o más correos.

No_emp	Email
111	juanp@ecn.es
111	jefe2@ecn.es
222	jsanchez@ecn.es

333	adiaz@ecn.es
333	ana32@gmail.com
...	...

Tabla CORREOS

Segunda forma normal (2FN)

Un esquema está en 2FN si: Está en 1FN, es decir, hemos quitados los grupos repetidos de la tabla.

Lo que hacemos, es crear una nueva tabla, en la nueva tabla incluimos el campo PK de la tabla origen (papá) y el campo o atributo que genera el conjunto de datos en dicha tabla (la tabla origen), en éste caso el correo que pueden ser N correos. La tabla papá ya no contiene el campo que provoca la falla de la 1FN, así evitamos la duplicidad de los registros.

Siguiendo el ejemplo y tomando en consideración lo que indica la 1FN, al quitar el atributo Email de la tabla Empleados tendríamos el siguiente esquema para la nueva tabla EMPLEADOS'(b)

no_emp	Nombre	Paterno	Puesto	Salario
111	Juan	Pérez	Jefe de Área	3000
222	José	Sánchez	Administrativo	1500
333	Ana	Díaz	Administrativo	1500
...

Tabla: EMPLEADOS

En la nueva tabla de empleados y no tenemos el campo Email y tampoco la duplicidad de registros.

Y además tendríamos una nueva tabla EMAILS con los campos o atributos (no_emp – para saber que empleado es y el campo email – el correo que tiene el empleado): el campo o atributo no_emp, en la tabla EMAILS es una llave foránea y aquí si es válido que se duplique. La relación que se tiene entre estas tablas es del tipo 1:N, es decir, un empleado puede tener uno o más registros, en nuestro caso, el empleado puede tener 1 o más correos.

No_emp	Email
111	juanp@ecn.es
111	iefe2@ecn.es
222	jsanchez@ecn.es
333	adiaz@ecn.es
333	ana32@gmail.com
...	...

Tabla: CORREOS

Tercera forma normal (3FN)

Una relación está en tercera forma normal si, y sólo si: está en 2FN y, además, cada atributo que no está incluido en la clave primaria no depende transitivamente de la clave primaria.

Por lo tanto, a partir de un esquema en 2FN, tenemos que buscar dependencias funcionales entre atributos que no estén en la clave.

En general, tenemos que buscar dependencias transitivas de la clave, es decir, secuencias de dependencias como la siguiente: $K \rightarrow A$ y $A \rightarrow B$, donde A y B no pertenecen a la clave. La solución a este tipo de dependencias está en separar en una tabla adicional N el/los atributos B, y poner como clave primaria de N el atributo que define la transitividad A.

Siguiendo el ejemplo anterior, podemos detectar la siguiente transitividad, aquellos atributos cuyo dato forman un catálogo y su descripción al momento de escribirlo o capturarlo por el usuario final puede dar pie a que se escriba de múltiples formas, generando problemas al momento de hacer búsquedas o comparaciones bajo esos datos.

No_emp->puesto (en éste atributo se puede escribir de diferente forma)

puesto->salario

Por lo tanto la descomposición sería la siguiente:

No_emp	Nombre	Paterno	Puesto
111	Juan	Pérez	Jefe de Área
222	José	Sánchez	Administrativo
333	Ana	Díaz	Administrativo
...
Tabla: EMPLEADOS			

Para solucionar éste problema, se tiene que crear una nueva tabla la cual contiene dos atributos (Puesto – clave del puesto que tiene el empleado y el atributo Descripción – que es la descripción del puesto que desempeña el empleado en la organización), en ésta tabla el atributo Puesto es la PK .

En la tabla de EMPLEADOS el campo o atributo Puesto no se quita, ése atributo se deja y queda como llave foránea (Foreign Key) haciendo referencia a la tabla PUESTOS. En otras palabras, el atributo Puesto es heredado de la nueva tabla llamada PUESTOS, pudiendo decir que la tabla PUESTOS es un catálogo. Esto ayuda a evitar que los usuarios escriban datos incorrectos o de distinta manera o forma, y ello nos genere problemas al momento de hacer consultas a la base de datos, por ejemplo, no es lo mismo escribir, Hola, a escribir HOLA.

El atributo Puesto en la entidad EMPLEADOS y Puesto en la entidad PUESTOS son los que hacen la relación entre las entidades.

No_emp	Nombre	Paterno	Puesto
111	Juan	Pérez	1
222	José	Sánchez	2
333	Ana	Díaz	2
...
Tabla EMPLEADOS			

Puesto	Descripción
1	Jefe de área
2	Administrativo
3	Finanzas
...	...
	Tabla: PUESTOS

La relación entre estas entidades es, un PUESTO tiene N empleados.

Cuando su base de datos cumpla con al menos estas tres formas normales, ya podemos trabajar con ella de manera adecuada y lógicamente.