



Artículo

Formas normales en DB relacionales

7



Israel Vázquez Morales ⌚ 26 de Abril de 2019

8

La normalización en las bases de datos relacionales es uno de esos temas que, por un lado es sumamente importante y por el otro suena algo esotérico. Vamos a tratar de entender las formas normales (FN) de una manera simple para que puedas aplicarlas en tus proyectos profesionales.

11

Primera Forma Normal (1FN)

12

Esta FN nos ayuda a eliminar los valores repetidos y no atómicos dentro de una base de datos.

13

Normalmente, una tabla está en primera forma normal si:

14

- Todos los atributos son atómicos. Un atributo es atómico si los elementos del dominio son simples e indivisibles.
- No debe existir variación en el número de columnas.
- Los campos no clave deben identificarse por la clave (dependencia funcional).
- Debe existir una independencia del orden tanto de las filas como de las columnas; es decir, si los datos cambian de orden no deben cambiar sus significados.

15

16

Se traduce básicamente a que si tenemos campos compuestos como por ejemplo "nombre_completo" que en realidad contiene varios datos distintos, en este caso podría ser "nombre", "apellido_paterno", "apellido_materno", etc.

También debemos asegurarnos que las columnas son las mismas para todos los



odos los campos que no se consideran clave deben depender de manera única por el o los campos que si son clave.

Los campos deben ser tales que si reordenamos los registros o reordenamos las columnas, cada dato no pierda el significado.

Segunda Forma Normal (2FN)

Esta FN nos ayuda a diferenciar los datos en diversas entidades.

Formalmente, una tabla está en segunda forma normal si:

- Está en 1FN
- Sí los atributos que no forman parte de ninguna clave dependen de forma completa de la clave principal. Es decir, que no existen dependencias parciales.
- Todos los atributos que no son clave principal deben depender únicamente de la clave principal.

Lo anterior quiere decir que sí tenemos datos que pertenecen a diversas entidades, cada entidad debe tener un campo clave separado. Por ejemplo:

alumnos				
alumno_id	alumno	nivel_curso	nombre_curso	materia
1	Juanito	Maestría	Data engineering	MySQL
1	Juanito	Maestría	Data engineering	Python
2	Pepito	Licenciatura	Programación	MySQL
2	Pepito	Licenciatura	Programación	Python

En la tabla anterior tenemos por lo menos dos entidades que debemos separar



RDB ¿Qué

anterior, quedaría de la siguiente manera:

alumnos			
alumno_id	alumno	nivel_curso	nombre_curso
1	Juanito	Maestría	Data engineering
2	Pepito	Licenciatura	Programación

materias		
materia_id	alumno_id	materia
1	1	MySQL
2	1	Python
3	2	MySQL
4	2	Python

Tercera Forma Normal (3FN)

Esta FN nos ayuda a separar conceptualmente las entidades que no son dependientes.

Formalmente, una tabla está en tercera forma normal si:

Se encuentra en 2FN

No existe ninguna dependencia funcional transitiva en los atributos que no son clave

Esta FN se traduce en que aquellos datos que no pertenecen a la entidad deben tener una independencia de las demás y debe tener un campo clave propio. Continuando con el ejemplo anterior, al aplicar la 3FN separamos la tabla alumnos ya que contiene datos de los cursos en ella quedando de la siguiente



RDB ¿Qué

alumnos		
alumno_id	alumno	curso_id
1	Juanito	1
2	Pepito	2

cursos		
curso_id	nivel_curso	nombre_curso
1	Maestría	Data engineering
2	Licenciatura	Programación

materias		
materia_id	alumno_id	materia
1	1	MySQL
2	1	Python
3	2	MySQL
4	2	Python

Cuarta Forma Normal (4FN)

Esta FN nos trata de atomizar los datos multivaluados de manera que no tengamos datos repetidos entre rows.

Formalmente, una tabla está en cuarta forma normal si:



RDB ¿Qué

- Los campos multivaluados se identifican por una clave única

Esta FN trata de eliminar registros duplicados en una entidad, es decir que cada registro tenga un contenido único y de necesitar repetir la data en los resultados se realiza a través de claves foráneas.

Aplicado al ejemplo anterior la tabla materia se independiza y se relaciona con el alumno a través de una tabla transitiva o pivote, de tal manera que si cambiamos el nombre de la materia solamente hay que cambiarla una vez y se propagara a cualquier referencia que haya de ella.



alumnos		
alumno_id	alumno	curso_id
1	Juanito	1
2	Pepito	2

cursos		
curso_id	nivel_curso	nombre_curso
1	Maestría	Data engineering
2	Licenciatura	Programación

materias	
materia_id	materia
1	MySQL
2	Python

materias_por_alumno		
mpa_id	materia_id	alumno_id
1	1	1
2	2	1
3	1	2
4	2	2



reconocer y mantener la consistencia de los datos.

Algunos autores precisan una 5FN que hace referencia a que después de realizar esta normalización a través de uniones (JOIN) permita regresar a la data original de la cual partió.



Escribe aquí tu comentario

+ 2



Julian Alejandro Lopez Romellon Estudiante • [hace 5 horas](#)

Pienso que sería mejor asignarle las materias a los cursos y que los alumnos estén registrados en los cursos, de esa manera no se repetiría el id de cada alumno en la tabla de materias. ¿Qué tan correcto sería esto?



0



Said Hernandez Estudiante • [anteayer](#)

Que bueno que lo plasmaron en texto



0



Daniel Noreña Estudiante • [hace 7 días](#)

una duda? que tan eficiente puede llegar a ser tener en memoria y rendimiento el hecho de tener 4 tablas con diferentes atributos que solo 1? La lógica de mi mente me dice que se va necesitar un poco mas de computo pero segun lo leído es erroneo.



2



ACorvachoH • [hace 4 días](#)

Llevo un tiempo ejecutando queries de manera practica y la intuición me dice que es mucho mas eficiente hacer llamados por campos (field, columnas, en una o varias tablas), que hacer operaciones entre filas (que es lo que tendrías que hacer de tener todo en una sola tabla). Esto debido a que un "SELECT field1, field2, ... fieldn FROM table1" llama a toda la columna solo indicando su posición.

En cambio, si queremos analizar datos dentro de un campo, debemos hacer un recorrido fila por fila, ya sea agrupando o utilizando sentencias CASE WHEN, etc.

Recuerda que es mucho mas probable que nuestra data tenga una cantidad absurda de observaciones (filas) en lugar de características (campos o columnas).

4



RDB ¿Qué