

NORMALIZACION

Definición.

La normalización es una técnica que se utiliza para crear relaciones lógicas apropiadas entre tablas de una base de datos. La normalización se adoptó porque el viejo estilo de poner todos los datos en un solo lugar, como un archivo o una tabla de la base de datos, era ineficiente y conducía a errores de lógica cuando se trataba de manipular los datos.

El proceso de normalización parte de las formas normales definidas por Edgar Frank Codd (1970) creador de las bases de datos relacionales. Primeramente, Codd formuló las tres primeras formas normales (1FN, 2FN, 3FN); posteriormente, unas anomalías detectadas forzaron a crear una forma normal más completa que la 3FN, es la FNBC (forma normal de Boyce y Codd), después Fagin definió la 4FN y 5FN.

La normalización es el proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser más simples y más estables, son más fáciles de mantener.

Los seres humanos tenemos la tendencia de simplificar las cosas al máximo. Lo hacemos con casi todo, desde los animales hasta con los automóviles. Vemos una imagen de gran tamaño y la hacemos más simple agrupando cosas similares juntas. Las guías que la normalización provee crean el marco de referencia para simplificar una estructura de datos compleja.

La normalización se utiliza para mejorar el esquema lógico, de modo que satisfaga ciertas restricciones que eviten la duplicidad de datos.

Objetivos de la Normalización

Minimizar la redundancia.

Minimizar el mantenimiento de datos.

Minimizar el impacto de futuros cambios (anomalías de actualización y anomalías de borrado) de datos, e ingreso de información (anomalías de inserción).

NORMALIZACION

Ventajas de la Normalización

Evita anomalías en inserciones, modificaciones y borrados.

Mejora la independencia de datos.

No establece restricciones artificiales en la estructura de los datos.

Facilidad de uso.

Flexibilidad.

Precisión.

Seguridad. Facilidad de implementación.

Independencia de datos.

Claridad.

Facilidad de gestión.

Mínima redundancia.

Máximo rendimiento de las aplicaciones.

Existen 5 Formas Normales y una variante de la 4ta:

- 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)

Cada una de estas formas tiene sus propias reglas.

En general, las primeras tres formas normales son suficientes para cubrir las necesidades de la mayoría de las bases de datos.

Antes de iniciar con la teoría referente a normalización es necesario conocer los conceptos de Dependencias Funcionales, los mismos que se utilizarán en la teoría de Normalización.

Formas Normales

Primera Forma Normal (1FN)

La regla de la Primera Forma Normal establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas y además los atributos que contienen múltiples valores se deben descomponer.

Poner la base de datos en la Primera Forma Normal resuelve el problema de los encabezados de columna múltiples. Muy a menudo, los diseñadores de bases de datos inexpertos harán algo similar a la tabla no normalizada. Una y otra vez, crearán columnas que representen los mismos datos. La normalización ayuda a clarificar la base de datos y a organizarla en partes más pequeñas y más fáciles de entender. En lugar de tener que entender una tabla gigantesca y monolítica que tiene muchos diferentes aspectos, sólo tenemos que entender los objetos pequeños y más tangibles, así como las relaciones que guardan con otros objetos también pequeños.

Definición formal: “ una relación está en 1FN, si cumple la propiedad de que sus dominios no tienen elementos que, a su vez, sean conjuntos.”

Si una relación no está en 1FN, hay que eliminar de ella los grupos repetitivos. Un grupo repetitivo será el atributo o grupo de atributos que tiene múltiples valores para cada tupla de la relación.

Por ejemplo: En la siguiente tabla se encuentran los datos del alquiler de diferentes películas. Esta tabla se encuentra en Forma Normal 0. (Sin normalizar).

Alquiler-Película(CodigoPel, TituloPel, EstudioPel, TipoPel, IdiomaPel, CodigoCli, NombreCli, FechaAlq, FechaDevol, NroAlq, NroVendedor, NombreVendedor)

Para que esta tabla pase a 1FN se debe eliminar las columnas que contienen atributos con múltiples valores y también los grupos repetitivos. Por ejemplo, la columna que contiene múltiples valores, que es NombreCli contiene el nombre y el apellido del cliente, esta columna se debe dividir en columnas individuales que guarden valores indivisibles, como NombreCli, y ApellidoCli. Se puede hacer lo mismo con la columna NombreVendedor.

NORMALIZACION

La tabla quedaría:

Alquiler-Película(CodigoPel, TituloPel, EstudioPel, TipoPel, IdiomaPel, CodigoClie, NombreCli, ApellidoCli, FechaAlq, FechaDevol, NroAlq, NroVen, NombreVen, ApellidoVen)

Además, se deben eliminar los grupos repetitivos. En este ejemplo, si al momento de alquilar, el cliente decide llevarse más de una película, esto generará la repetición de los datos del cliente y los datos del vendedor por cada película que alquile.

La tabla final en 1FN es:

Alquiler-Película(NroAlq, CodigoPel, TituloPel, EstudioPel, TipoPel, IdiomaPel)
Alquiler(NroAlq, CodigoClie, NombreCli, ApellidoCli, FechaAlq, FechaDevol, NroVen, NombreVen, ApellidoVen)

Segunda Forma Normal (2FN)

Una tabla se dice que está en **2FN** si y sólo si cumple dos condiciones:

1. Se encuentra en 1FN
2. Todo atributo no-primario (aquellos que no pertenecen a la clave principal, los que se encuentran fuera de la caja) dependen totalmente (tiene una dependencia funcional total) de la clave completa y, por tanto, no de una parte de ella.

Esta forma normal sólo se considera si la clave principal es compuesta y, por tanto, está formada por varios atributos. Si la clave principal de la tabla está formada por un único atributo, entonces la tabla ya se encuentra en 2FN.

La definición formal es: “se dice que una relación R está en 2FN, si además de estar en 1FN, cualquiera de sus atributos no-primarios tienen una dependencia funcional plena con cada una de las claves candidatas de R”.

NORMALIZACION

Si una tabla T tiene como atributos A, B, C, D y la clave es A.B cumpliéndose las dependencias: $A.B \rightarrow C$ $B \rightarrow D$

Se observa que la tabla no se encuentra en 2FN puesto que el atributo D no tiene una dependencia funcional total con la clave entera A.B, sino con una parte de la clave (B). Para convertir una tabla que no está en segunda forma normal a 2FN, se realiza una proyección y se crea:

- 1) Una tabla con la clave y todas sus dependencias totales con los atributos secundarios afectados
- 2) Otra tabla con la parte de la clave que tiene dependencias, junto con los atributos secundarios implicados

La clave de la nueva tabla será la antigua parte de la clave.

En nuestro ejemplo, tomamos la relación que tiene clave compuesta y procedemos a separar los datos propios de la película de los datos del alquiler:

Películas(CodigoPel, TituloPel, EstudioPel, TipoPel, IdiomaPel)

Películas-Alquiladas(NroAlg, CodigoPel)

NORMALIZACION

Tercera Forma Normal (3FN)

La definición formal es: “una relación R está en 3FN si esta en 2FN y además ninguno de sus campos no-primarios tiene dependencias transitivas respecto de las claves candidatas de R.”

Esto quiere decir que un atributo no-primario sólo se debe conocer a través de la clave principal o claves secundarias de la tabla y no por medio de otro atributo no primario.

Por tanto, para pasar una tabla que no cumple la tercera forma normal a 3FN, se realiza una proyección y se genera:

- 1) Una tabla con la clave y todos los atributos no primarios que no son transitivos
- 2) Otra tabla con los atributos transitivos y el atributo no primario (que será la clave de la nueva tabla) por medio del cual mantienen la transitividad

Lógicamente, en la primera tabla T1, el atributo C es clave ajena con respecto de T2 y de ese modo todos los atributos quedan relacionados entre sí. Es lo que se denomina interrelación entre la tabla T1 y T2.

En nuestro ejemplo a partir de la tabla 1FN:

Alquiler(NroAlq, CodigoCli, NombreCli, ApellidoCli, FechaAlq, FechaDevol, NroVen, NombreVen, ApellidoVen)

Se generarán las siguientes tablas:

Clientes(CodigoCli, NombreCli, ApellidoCli)

Vendedores(NroVen, NombreVen, ApellidoVen)

Alquiler(NroAlq, CodigoCli, FechaAlq, FechaDevol, NroVen)

Estando en 3FN se considera un esquema de base de datos normalizado.

Forma Normal de Boyce Codd (FNBC)

Tras la creación de la 3FN se observó posteriormente que se encontraban algunas anomalías que no eran abordadas. Son casos de tablas que estando en 3FN mantienen una dependencia de un atributo secundario con parte de la clave. Para poder manejar esa dependencia en las aplicaciones es imprescindible manejar una gran cantidad de registros innecesarios (aquéllos donde se mantiene fija la parte de la clave que depende y va variando el resto de la clave) La nueva definición se debe a Boyce y Codd:

"Una tabla T está en FNBC si y sólo si las únicas DF elementales son aquellas en las que la clave principal (y claves secundarias) determinan un atributo".

La definición engloba la 3FN puesto que las dependencias transitivas existen por medio de atributos secundarios que no eran clave.

Pero esta definición realmente se creó para evitar los casos anómalos que no se evitaban con la 3FN y que aparecen cuando a partir de un atributo no primario se conoce una parte de la clave.

Si la clave está formada por un sólo atributo, la tabla está en FNBC (si ya estaba en 3FN) como sucedía con la 2FN.

Por lo tanto, para que una tabla que está en 3FN y no cumple la norma de Boyce-Codd se encuentre en FNBC se realiza una proyección procediendo de la siguiente manera:

1. Se crea una tabla con la parte de la clave que es independiente (A) y todos los atributos no primarios (C)
2. Se crea otra tabla con la parte de la clave restante y el atributo secundario del que depende, y será éste último la clave de la nueva tabla

Cuarta Forma Normal (4FN)

La 4FN la generó Fagin tras el teorema que demostró y que dice:

"Una tabla T con atributos A , B y C se puede descomponer sin pérdidas en sus dos proyecciones $T1(A,B)$ y $T2(A,C)$ si y solo si la Dependencia Multivaluada $A \twoheadrightarrow B \mid C$ se cumple en T "

De ese modo, se define la 4FN de la siguiente forma:

Una tabla T se dice que está en 4FN si cumple dos condiciones:

- 1) Está en FNBC.
- 2) Las únicas Dependencias Multivaluadas existentes son las Dependencias Funcionales de la clave con los atributos secundarios.

Quinta Forma Normal (5FN)

Para que una tabla se encuentra en 5FN se deben cumplir dos condiciones:

- 1) Se encuentra en 4FN
- 2) Toda Dependencia de Join viene implicada por las claves (principal o secundarias) de la tabla.

Es decir, que la tabla estará en 5FN si es posible generar unas proyecciones y al realizar su join, los atributos comunes que realizan la operación (atributos de join) están formados por claves (principal o secundaria) de la tabla.