

ING. ELLUZ LEMUS

TEORÍA DE LA NORMALIZACIÓN

¿QUÉ ES LA NORMALIZACIÓN DE LOS DATOS?

La normalización de bases de datos es un proceso que consiste en designar y aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional. Con objeto de minimizar la redundancia de datos, facilitando su gestión posterior.

¿RETOS DE LA NORMALIZACIÓN?

Uno de los retos en el diseño de la base de datos es el de obtener una estructura estable y lógica tal que:

1. El sistema de base de datos no sufra de anomalías de almacenamiento.
2. El modelo lógico pueda modificarse fácilmente para admitir nuevos requerimientos.

RIESGOS DE LA NORMALIZACIÓN

Existen diversos riesgos en el diseño de las bases de datos relacionales que afecten la funcionalidad de la misma, los riesgos generalmente son la redundancia de información y la inconsistencia de datos.

LA NORMALIZACIÓN SE LLEVA A CABO POR CUATRO RAZONES

- Estructurar los datos de forma que se puedan representar las relaciones pertinentes entre los datos.
- Permitir la recuperación sencilla de los datos en respuesta a las solicitudes de consultas y reportes.

- Simplificar el mantenimiento de los datos actualizándolos, insertándolos y borrándolos.
- Reducir la necesidad de reestructurar o reorganizar los datos cuando surjan nuevas aplicaciones.

PROCEDIMIENTO A SEGUIR PARA APLICAR LA NORMALIZACIÓN

- Identifica una clave primaria para la entidad sin normalizar (antes de iniciar el proceso)
- Identifica y resuelve las violaciones a la primera forma normal (1FN), removiendo todos los grupos repetitivos.
- Identifica y resuelve las violaciones a la segunda forma normal (2FN), asegurándote de que cada atributo que no sea clave dependa de la totalidad de la clave primaria
- Identifica y resuelve las violaciones a la tercera forma normal (3FN), asegurándote de que cada atributo no clave no dependa de otro atributo que no sea la clave primaria

BENEFICIOS O VENTAJAS DE APLICAR LA NORMALIZACIÓN

PRIMERA FORMA NORMAL (1FN)

Una tabla o entidad R está en primera forma normal (1FN), si y sólo si cada elemento de R es atómico (indivisible), es decir, no contiene atributos multivaluados. Debe existir una dependencia funcional simple.

Interpretación: Para llegar a la primera forma normal (1FN) se aíslan en entidades nuevas y únicas, todos los atributos (o grupos de atributos) repetitivos (recurrentes), partiendo la entidad en dos o más entidades.

Procedimiento para aplicar la 1FN: Formula la siguiente pregunta para cada atributo contenido en la entidad:

¿Se repite este atributo más de una vez para cualquier registro de esta entidad?

Si la respuesta es **afirmativa**, procede de la siguiente manera:

1. Crea una nueva entidad y asígnale un nombre
2. Mueve los atributos repetidos a la nueva entidad
3. Selecciona entre estos atributos una clave primaria o sustituta
4. Establece una relación desde la entidad original hacia la nueva entidad. La clave primaria de la entidad original migra hacia la nueva entidad, convirtiéndose en una clave foránea.

SEGUNDA FORMA NORMAL (2FN)

Una tabla o entidad R está en segunda forma normal (2FN), si y sólo si está en 1FN y **cada atributo no primario (atributo no clave)** de R depende en forma funcionalmente completa de la clave primaria de ésta.

Interpretación: Para llegar a la segunda forma normal (2FN) se aíslan en entidades nuevas y únicas, los atributos que **no dependan de todos los atributos claves de la entidad** (este paso se aplica a entidades con *claves concatenadas*). Debe existir una dependencia funcionalmente completa.

Procedimiento para aplicar la 2FN: Para aquellas entidades cuya clave primaria está formada por dos o más atributos, Formula la siguiente pregunta para cada atributo que no forme parte de la clave primaria:

¿Es este atributo dependiente de la totalidad de la clave primaria?

(No dependencias parciales)

Si la respuesta es **negativa**, procede de la siguiente manera:

1. Crea una nueva entidad y asígnale un nombre

2. Mueve los atributos que tienen la misma dependencia parcial a la nueva entidad
3. Usa un atributo determinante como la nueva clave primaria
4. Crea una relación desde la nueva entidad hacia la entidad original

TERCERA FORMA NORMAL (3FN)

Una tabla o entidad R está en tercera forma normal (3FN), si y sólo si está en 2FN y cada **atributo no primario** de R **no es transitivamente dependiente** de cada clave primaria.

Interpretación: Para llegar a la tercera forma normal (3FN) se aíslan en entidades nuevas y únicas, los atributos que dependen de atributos que no son clave primaria de la entidad.

Procedimiento para aplicar la 3FN: Pregúntate lo siguiente para cada atributo que **no forme parte** de la clave primaria:

***¿Depende este atributo de algún otro atributo que no forme parte de la clave primaria?
(No interdependencias entre atributos que no formen parte de la clave primaria)***

Si la respuesta es **afirmativa**, procede de la siguiente manera:

1. Crea una nueva entidad y asígnale un nombre
2. Mueve los atributos que tienen la misma interdependencia a la nueva entidad
3. Usa un atributo determinante como la nueva clave primaria
4. Crea una relación desde la nueva entidad hacia la original

VENTAJAS DE LA NORMALIZACIÓN DE DATOS: El proceso de normalización de una base de datos tiene numerosas ventajas. A continuación, se enumeran algunos beneficios clave:

- Se puede eliminar la duplicación.

- Al normalizar, podemos reducir los valores nulos.
- Da como resultado una base de datos más pequeña (ya que hay menos duplicación de datos).
- Minimizar/evitar problemas con la modificación de datos.
- Facilita las consultas.
- La estructura de la base de datos es más comprensible y sencilla.
- Se pueden agregar datos existentes a la base de datos sin que esto tenga ningún impacto.
- Debido a que la tabla es compacta y pueden caber más filas en la página de datos, la búsqueda, la clasificación y la indexación pueden ser más rápidas.

DESVENTAJAS DE LA NORMALIZACIÓN DE DATOS: El proceso de normalización de una base de datos presenta numerosas desventajas. A continuación, se enumeran algunas de ellas:

- Disminuye la calidad y la integridad de los datos
- Puede introducir anomalías e inconsistencias en los datos
- Los tiempos de respuesta suelen ralentizarse a medida que la base de datos crece

BENEFICIOS DE LA NORMALIZACIÓN DE LAS BASE DE DATOS

- Evitar la creación y actualización de relaciones y dependencias no deseadas entre datos
- Prevenir borrados indeseados de datos
- Optimizar el espacio de almacenamiento.
- Reducir el tiempo y complejidad de revisión de las bases de datos cuando es necesario introducir nuevos tipos de datos.

EJERCICIOS NORMALIZACIÓN

1. EJEMPLO 1

Vamos a considerar los datos de la siguiente tabla.

`ordenes` (Id_Orden, Fecha, Id_Cliente, Nom_Cliente, Ciudad, Id_art, Nom_art, Cant, Precio)

Ordenes

Id_orden	Fecha	Id_cliente	Nom_cliente	Ciudad	Id_art	Nom_art	Cant	Precio
2301	23/02/11	101	Martin	Riobamba	3786	Red	3	35,00
2301	23/02/11	101	Martin	Riobamba	4011	Raqueta	6	65,00
2301	23/02/11	101	Martin	Riobamba	9132	Paq-3	8	4,75
2302	25/02/11	107	Herman	Ambato	5794	Paq-6	4	5,00
2303	27/02/11	110	Pedro	Quito	4011	Raqueta	2	65,00
2303	27/02/11	110	Pedro	Quito	3141	Funda	2	10,00

1.1 PRIMERA FORMA NORMAL (1FN)

- Una relación está en primera forma normal si todos los dominios contienen sólo valores atómicos.
- Un dominio es atómico si se considera que los elementos del dominio son unidades indivisibles
- La regla de la 1FN establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas.

Ejemplo:

Al examinar estos registros, podemos darnos cuenta que contienen un grupo repetidos ID_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 (clave) 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, Ciudad) `articulos_ordenes` (Id_orden, Id_Art, Nom_art, Cant, Precio)

ordenes

Id_orden	Fecha	Id_cliente	Nom_cliente	Ciudad
2301	23/02/11	101	Martin	Riobamba
2302	25/02/11	107	Herman	Ambato
2303	27/02/11	110	Pedro	Quito

articulos_ordenes

Id_orden	Id_Art	Nom_art	Cant	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

1.2 SEGUNDA FORMA NORMA (2FN)

Una relación está en segunda forma normal si y sólo si está en 1FN y todos los atributos no clave dependen por completo de cualquier clave candidata.

Ejemplo:

Ahora procederemos a aplicar la segunda forma normal al ejemplo, es decir, tenemos que eliminar cualquier columna no clave, que no dependa de la clave primaria de la tabla.

Los pasos a seguir son:

- Determinar cuáles columnas que no son clave, no dependen de la clave 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 `NOM_ART` son dependientes de `ID_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`, `Id_Art`, `Cant`)

`artículos` (`Id_Art`, `Nom_art`, `Precio`)

`articulos_ordenes`

Id_orden	Id_Art	Cant
2301	3786	3
2301	4011	6
2301	9132	8
2302	5794	4
2303	4011	2
2303	3141	2

`artículos`

Id_Art	Nom_art	Precio
3786	Red	35,00
4011	Raqueta	65,00
9132	Paq-3	4,75
5794	Paq-6	5,00
3141	Funda	10,00

1.3 TERCERA FORMA NORMA (3FN)

Una relación está en tercera forma normal si y sólo si está en 2FN y además cada atributo del esquema de relación que no está en la clave primaria sólo depende funcionalmente de la clave primaria, y no de ningún otro atributo

Ejemplo:

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

Los pasos a seguir son:

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

Al observar las tablas, 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 CIUDAD son dependientes de ID_CLIENTE, y esta columna no es la llave primaria.

Para normalizar esta tabla, moveremos las columnas no clave y la columna clave 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)
clientes (Id_cliente, Nom_cliente, Ciudad)
```

ordenes

Id_orden	Fecha	Id_cliente
2301	23/02/11	101
2302	25/02/11	107
2303	27/02/11	110

clientes

Id_cliente	Nom_cliente	Ciudad
101	Martin	Riobamba
107	Herman	Ambato
110	Pedro	Quito

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

```
Ordenes (Id_orden, Fecha,
Id_cliente) Clientes (Id_cliente,
Nom_cliente, Ciudad) Articulos (
Id_Art, Nom_art, Precio)
Articulos_ordenes (Id_orden, Id_Art,
Cant)
```