

#YoPrograma 2da edición

← Volver al menu principal

Material de lectura

Bases de datos

Normalización de Datos

La normalización es la técnica de estandarización y validación de datos que se utiliza para diseñar las tablas y establecer las relaciones entre ellas. Esto nos permite una mayor organización para eliminar la redundancia de datos y proteger la integridad de los mismos. Sin la normalización, tendríamos datos redundantes los cuales traen aparejados problemas de mantenimiento, además de ocupar espacio en el disco duro, como también, dependencias incoherentes.

Las formas normales se pueden clasificar en:

1. Primera Forma Normal (1FN)
2. Segunda Forma Normal (2FN)
3. Tercera Forma Normal (3FN)
4. Cuarta Forma Normal (4FN)
5. Quinta Forma Normal (5FN)

Vamos a realizar el proceso de normalización sobre la siguiente tabla:

Id_Cliente	Nombre	Apellido	Email	Provincia	Localidad	CVU	EstadoCuenta
1	JORGE	GOMEZ	jorge@test.com; jgomez@test.com	SALTA	EL JARDIN	122334455	HABILITADA
2	LUISA	PEREZ	luisaperez@test.com	CÓRDOBA	CARLOS PAZ	212223334	HABILITADA
3	MARTHA	GONZALES	martha@test.com	TUCUMAN	TAFI VIEJO	312233445	HABILITADA
4	JUAN	PEREZ	juan@test.com	CÓRDOBA	LA FALDA	185274196	HABILITADA
5	PASCUAL	VARGAS	vargasp@test.com; pvargas@hotmail.com	CÓRDOBA	LA FALDA	321654789	HABILITADA

Aclaración: CVU es Clave virtual uniforme.

En esta tabla, que es la normalidad en la vida cotidiana de las personas, para los desarrolladores es casi una pesadilla de datos.

Podemos encontrar los siguientes problemas:

1. Datos duplicados como nombres de provincia, estados de cuenta, etc
2. Campos con varios datos
3. Datos relacionados
4. Podríamos tener errores en la relación de Provincias y Localidades. Por algún error una localidad podría haber quedado junto a una provincia que no le correspondía.

Ayuda

Texto a voz

5. Por algún error humano, alguien pudo haber escrito mal el nombre de una provincia o localidad. Encontrar y solucionar en la base de datos ese incidente puede ser muy incómodo
6. No tenemos control si hay emails o CVUs duplicados
7. Los procesos de búsqueda y actualización son ineficientes. Por ejemplo, si quisiéramos buscar y reemplazar el email de un usuario como Pascual Vargas.

Realicemos el proceso de normalización y veamos qué va ocurriendo con cada uno de estos problemas.

Primera Forma Normal (1FN)

Una tabla está en primera forma normal cuando todos los datos son atómicos, es decir, cada atributo tiene su propia celda y todas las columnas contienen el mismo tipo de datos. Sirve para eliminar los grupos repetidos.

Como vemos en el siguiente ejemplo, la primera forma normal no se cumple por el campo Email. En la primera y en la quinta fila tenemos dos emails:

Id_Cliente	Nombre	Apellido	Email	Provincia	Localidad	CVU	EstadoCuenta
1	JORGE	GOMEZ	jorge@test.com; jgomez@test.com	SALTA	EL JARDIN	122334455	HABILITADA
2	LUISA	PEREZ	luisaperez@test.com	CÓRDOBA	CARLOS PAZ	212223334	HABILITADA
3	MARTHA	GONZALES	martha@test.com	TUCUMAN	TAFI VIEJO	312233445	HABILITADA
4	JUAN	PEREZ	juan@test.com	CÓRDOBA	LA FALDA	185274196	HABILITADA
5	PASCUAL	VARGAS	vargasp@test.com; pvargas@hotmail.com	CÓRDOBA	LA FALDA	321654789	HABILITADA

Para solucionarlo, podemos crear un registro por cada email o eliminar el atributo que no cumple dicha condición. En la siguiente imagen vemos cómo quedaría la tabla si separamos el email y creamos un nuevo registro para cumplir con la condición. Ahora tenemos un email por registro:

Id_Cliente	Nombre	Apellido	Email	Provincia	Localidad	CVU	EstadoCuenta
1	JORGE	GOMEZ	jgomez@test.com	SALTA	EL JARDIN	122334455	HABILITADA
1	JORGE	GOMEZ	jorge@test.com	SALTA	EL JARDIN	122334455	HABILITADA
2	LUISA	PEREZ	luisaperez@test.com	CÓRDOBA	CARLOS PAZ	212223334	HABILITADA
3	MARTHA	GONZALES	martha@test.com	TUCUMAN	TAFI VIEJO	312233445	HABILITADA
4	JUAN	PEREZ	juan@test.com	CÓRDOBA	LA FALDA	185274196	HABILITADA
5	PASCUAL	VARGAS	pvargas@hotmail.com	CÓRDOBA	LA FALDA	321654789	HABILITADA
5	PASCUAL	VARGAS	vargasp@test.com	CÓRDOBA	LA FALDA	321654789	HABILITADA

Sin embargo se nos presenta un nuevo problema. Vemos que JORGE GOMEZ aparece dos veces, y lo mismo ocurre con PASCUAL VARGAS.

Entonces para cumplir con la primera forma normal eliminando el atributo email, tendremos que crear una tabla emails que contenga las columnas email, que funcionara como clave primaria (PK), y una columna Id_Cliente, que funcionara como Clave Foránea que apunta hacia el id del cliente.

Cientes

Id_Cliente	Nombre	Apellido	Provincia	Localidad	CVU	EstadoCuenta
1	JORGE	GOMEZ	SALTA	EL JARDIN	122334455	HABILITADA
2	LUISA	PEREZ	CÓRDOBA	CARLOS PAZ	212223334	HABILITADA
3	MARTHA	GONZALES	TUCUMAN	TAFI VIEJO	312233445	HABILITADA
4	JUAN	PEREZ	CÓRDOBA	LA FALDA	185274196	HABILITADA
5	PASCUAL	VARGAS	CÓRDOBA	LA FALDA	321654789	HABILITADA

Emails

Email	Id_Cliente
jorge@test.com	1
jgomez@test.com	1
luisaperez@test.com	2
martha@test.com	3
juan@test.com	4
vargasp@test.com	5
pvargas@hotmail.com	5

El resultado final es que tenemos una tabla con todos los datos de los usuarios, y una segunda tabla que nos permite relacionar los emails con los datos de usuario. Así si un usuario tiene N emails, en esta nueva tabla tendremos N mails relacionados con el mismo id de usuario.

En este primer paso hemos eliminado algunos de los problemas mencionados anteriormente:

1. Campos con varios datos. Cada columna ahora contiene un único valor
2. Los procesos de búsqueda y actualización son inefficientes. Por ejemplo si quisiéramos buscar reemplazar el email de un usuario como Pascual Vargas.
 1. El sistema es eficiente para buscar usuario por email
 2. Si queremos editar, agregar o borrar el email de un usuario solamente tenemos que realizar la operación sobre la tabla Emails

Segunda Forma Normal (2FN)

Para estar en la segunda forma normal, se agregan atributos que no forman parte de ninguna clave a las condiciones de la primera forma normal. Su funcionamiento depende de la clave primaria. Ésta sirve para eliminar los datos redundantes.

Después de pasar por el proceso de 1FN la tabla de clientes nos quedó de la siguiente forma

Cientes

Id_Cliente	Nombre	Apellido	Provincia	Localidad	CVU	EstadoCuenta
1	JORGE	GOMEZ	SALTA	EL JARDIN	122334455	HABILITADA
2	LUISA	PEREZ	CÓRDOBA	CARLOS PAZ	212223334	HABILITADA
3	MARTHA	GONZALES	TUCUMAN	TAFI VIEJO	312233445	HABILITADA
4	JUAN	PEREZ	CÓRDOBA	LA FALDA	185274196	HABILITADA
5	PASCUAL	VARGAS	CÓRDOBA	LA FALDA	321654789	HABILITADA

El problema que vamos a resolver es la falta de normalización de las columnas Provincia y Localidad.

Para empezar será necesario

1. Crear las tablas Provincias y Localidades
2. Eliminar la columna Provincia de la tabla Clientes

Ayuda

Texto a voz

¿Por qué eliminamos la columna provincia? Porque el dato de la provincia lo podemos obtener sabiendo la localidad. Todas las localidades pertenecen a una única provincia, así es que si se sabe la localidad, se sabe la provincia.

Entonces, ahora obtendremos la segunda forma normal, ya que obtendremos la provincia desde la tabla localidades y la localidad desde la tabla clientes. Además, creamos la tabla EstadosCuentas con Id_Estado (PK) y Nombre.

Al aplicar la segunda forma normal nos quedarían las tablas Clientes, Provincias y Localidades:

Provincias

Id_Provincia	Nombre
1	CÓRDOBA
2	SALTA
3	TUCUMAN

Localidades

Id_Localidad	Nombre	Id_Provincia
1	EL JARDÍN	2
2	CARLOS PAZ	1
3	TAFI VIEJO	3
4	LA FALDA	1

Clientes

Id_Cliente	Nombre	Apellido	Id_Localidad	CVU	Id_EstadoCuenta
1	JORGE	GOMEZ	1	122334455	1
2	LUISA	PEREZ	2	212223334	1
3	MARTHA	GONZALES	3	312233445	1
4	JUAN	PEREZ	4	185274196	1
5	PASCUAL	VARGAS	4	321654789	1

Observemos la lógica de estas 3 tablas:

1. Tenemos una tabla Provincias con el nombre de la provincia y su respectivos id.
2. Una tabla Localidades donde cada localidad tiene su id, el nombre de la localidad y un id de provincia que se corresponde con la tabla Provincias
3. Tabla Clientes donde ya no tenemos la columna provincias, y solamente tenemos la columna Localidad con el id de la localidad.

Ahora volvamos a los problemas que mencionamos anteriormente

1. Podríamos tener errores en la relación de Provincias y Localidades. Por algún error una localidad pudo haber quedado junto a una provincia que no le correspondía. Este problema ya no existe, todas las relaciones de provincia y localidades existen en una única tabla, y de ser necesario cambiar una relación la tarea se simplifica a editar el registro de una única tabla.
1. Por algún error humano, alguien pudo haber escrito mal el nombre de una provincia o localidad. Encontrar y solucionar en la base de datos ese incidente puede ser muy incómodo. Nuevamente este problema se soluciona editando un único registro, en una única tabla.

La importancia de una buena normalización es que además de prevenir errores y ser eficiente en el uso del sistema, nos ayuda a realizar tareas de mantenimiento mucho más simples.

Para mantener la consistencia del proceso de normalización, aprovechamos y separamos los estados de cuenta en una tabla aparte. Si no te diste cuenta, en la tabla clientes ahora hay una columna Id_Estado_Cuentas y en la siguiente imagen esta es su correspondiente tabla de relaciones:

EstadosCuentas

Id_Estado	Nombre
1	HABILITADA
2	DESHABILITADA

Ayuda

Texto a voz

Tercera Forma Normal (3FN)

Para cumplirse la tercera forma normal, a las dos condiciones anteriores hay que agregarle que los atributos que no son clave no pueden depender de manera transitiva de una clave candidata. Aquí se deben eliminar las columnas que no dependen de la clave principal.

Esta forma normal es considerada como el estándar a cumplir por los esquemas relacionales.

Siguiendo el ejemplo, tenemos que agregar la tabla Cuentas la cual tiene los atributos CVU (PK), Id_Estado e Id_Cliente y en la tabla Clientes eliminaremos a columna CVU y Id_EstadoCuenta quedando las tablas de la siguiente forma:

EstadosCuentas

Id_Estado	Nombre
1	HABILITADA
2	DESHABILITADA

Provincias

Id_Provincia	Nombre
1	CÓRDOBA
2	SALTA
3	TUCUMAN

Localidades

Id_Localidad	Nombre	Id_Provincia
1	EL JARDÍN	2
2	CARLOS PAZ	1
3	TAFI VIEJO	3
4	LA FALDA	1

Cuentas

CVU	Id_Estado	Id_Cliente
122334455	1	1
212223334	1	2
312233445	1	3
185274196	1	4
321654789	1	5

Clientes

Id_Cliente	Nombre	Apellido	Id_Localidad
1	JORGE	GOMEZ	1
2	LUISA	PEREZ	2
3	MARTHA	GONZALES	3
4	JUAN	PEREZ	4
5	PASCUAL	VARGAS	4

Con el resultado obtenido, en la tabla Cuentas ahora tenemos la posibilidad de mantener la información del CVU y el estado de la cuenta de una forma mas eficiente y desacoplada de la tabla Clientes. La cual se ha reducido significativamente a los datos que son relevantes para los detalles de un cliente como nombre, apellido y localidad.

En este punto podemos observar que una única tabla que teníamos al inicio, se ha convertido en 5 tablas. En cada tabla encontramos datos representativos, de forma ordenada y relacionada.

Cuarta Forma Normal (4FN)

El objetivo de la cuarta forma normal, es evitar que nuestras tablas posean dependencias multivaluadas. Las dependencias multivaluadas son aquellas donde la existencia de dos o más relaciones N:M (Muchos a muchos) causando redundancia.

Como nuestro ejemplo de momento no cuenta con ninguna instancia de relación de muchos a muchos (N:M) tenemos que asumir que necesitamos habilitar los movimientos que tiene cada cliente con uno o más tipos de moneda.

Para esto crearemos la tabla Monedas, la cual tendría los atributos Id_Moneda y Nombre, y una tabla ClientesMonedas que tendrá las columnas Id_Cliente e Id_Moneda, en donde asignaremos con qué tipos de monedas puede operar cada cliente:

Ayuda

Texto a voz

Monedas

Id_Moneda	Nombre
1	PESO
2	DOLAR
3	BITCOIN

ClientesMonedas

Id_Cliente	Id_Moneda
1	1
1	2
2	1
2	3
3	1
3	2
4	1
5	1
5	2
5	3

Cómo vemos en la tabla anterior, ahora nuestros clientes tienen la posibilidad de manejar diferentes tipos de moneda. En la tabla ClientesMoneda vemos que un cliente puede tener muchas monedas, y que una moneda puede ser usada por muchos clientes (N:M).

Quinta Forma Normal (5FN o forma normal de Proyección-Unión)

Una relación se encuentra en 5FN si ya pasó por la normalización 4FN y además, las dependencias existentes son las dependencias de Join o de unión con sus proyecciones. Esto ocurre cuando podemos obtener la tabla original por medio de la unión de sus proyecciones, relacionándose entre sí a través de la clave primaria o alguna de sus claves alternativas.

Esto quiere decir que si mediante la unión de los campos de las tablas, podemos recuperar la tabla inicial. Si hacemos el ejercicio de ir conectando todos los id de las tablas, veremos que es posible reconstruir la tabla inicial.

Te dejamos un cuestionario para repasar algunos conceptos vistos en esta sección:

¿Qué es la normalización?

- ☐ La normalización es la técnica de validación para que todas las tablas tengan los mismos tipos de datos
- ☐ Es la técnica de estandarización y validación de datos que se utiliza para diseñar las tablas y establecer
- ☐ La normalización es la técnica de estandarización y validación para que todas las tablas tengan la mism

✓ Comprobar

Ayuda

Texto a voz



Actividad previa

◀ Diseño de Base de Datos

Siguiente actividad

Programando Bases de Datos ▶

Ayuda

Texto a voz