**BASE DE DATOS**

**Proceso de Normalización**

La normalización es el proceso de optimizar una entidad respecto a sus atributos, de tal manera de evitar:

1. Redundancia de datos: datos repetidos sin sentido.
2. Pérdida de integridad: instancias de una entidad que no tienen correspondencia con las instancias de sus entidades relacionadas.
3. Ambigüedades: la información que se obtiene de un conjunto de instancias es inconsistente con la realidad.
4. Pérdida de información: instancias de una entidad sin ninguna relación, quedando almacenadas, pero sin sentido alguno extraíble.
5. Anomalías en general.

Suponga el siguiente registro de ventas

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TRABAJADOR** | **NOMBRE** | **APELLIDO** | **RUN** | **CIUDAD** | **NRO BOLETA** | **NOMBRE PRODUCTO** | **FECHA VENTA** |
| 100234 | CARLOS | PEREZ | 11234234-K | VALPARAÍSO | 1 | TELEVISOR | 01-12-2021 |
| 100234 | CARLOS | PEREZ | 11234234-K | VALPARAÍSO | 2 | RADIO | 02-12-2021 |
| 101555 | RODRIGO | LORCA | 23555666-0 | CHILLÁN | 3 | TELEVISOR | 02-02-2022 |
| 101555 | RODRIGO | LORCA | 23555666-0 | CHILLÁN | 4 | COMPUTADOR | 03-05-2022 |

Es un registro completamente NO normalizado, esto podría causar los siguientes problemas:

* Redundancia: los datos de los trabajadores se repiten en cada venta
* Problemas de actualización: si se cambia la ciudad de un trabajador sin cambiarlas todas,se crea una inconsistencia.
* Problemas de inserción: en caso de querer registrar un vendedor sin ventas.
* Problemas de eliminación: ya que al borrar una venta tendríamos que eliminar los datos de un trabajador.

La teoría de la normalización supone que un conjunto de atributos (A) y las dependencias entre los atributos (D) conforman un esquema relacional R(A, D); se modifican sucesivamente para cumplir con las siguientes características

Un esquema relacional debe cumplir con tres propiedades: conservación de la información, conservación de las dependencias y mínima redundancia de datos.

**Conservación de la información.**

El proceso de normalización se debe llevar a cabo sin que se produzcan pérdidas de información en la base de datos. Para que esto se dé, se debe cumplir:

* + 1. Conservación de los atributos: el modelo resultante debe tener la misma cantidad de atributos. Si hay atributos comunes se pueden fusionar, pero el conteo final debe ser el mismo.
    2. Conservación del contenido: las relaciones resultantes del proceso de normalización deben entregar como resultado el mismo conjunto de datos.

Por ejemplo:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TRABAJADOR** | **NOMBRE** | **APELLIDO** | **RUN** | **CIUDAD** |
| 100234 | CARLOS | PEREZ | 11234234-K | VALPARAÍSO |
| 101555 | RODRIGO | LORCA | 23555666-0 | CHILLÁN |

|  |  |
| --- | --- |
| **NRO BOLETA** | **NOMBRE PRODUCTO** |
| 1 | TELEVISOR |
| 2 | RADIO |
| 3 | TELEVISOR |
| 4 | COMPUTADOR |

|  |  |  |
| --- | --- | --- |
| **BOLETA** | **TRABAJADOR** | **FECHA VENTA** |
| 1 | 100234 | 01-12-2017 |
| 2 | 100234 | 02-12-2017 |
| 3 | 101555 | 02-12-2017 |
| 4 | 101555 | 03-12-2017 |

**Conservación de las dependencias.**

Las dependencias son restricciones que están presentes porque el proceso de abstracción las estableció para el modelo. Cuando se realice el proceso de normalización estas dependencias deben permanecer, por lo que, al modificar el esquema, este debe resultar con el mismo conjunto de dependencias.

**Mínima redundancia de datos (normalización).**

Un modelo relacional está normalizado si cumple con una serie de restricciones o características que indiquen que cumple con ellas; mientras mayor sea la normalización, menores serán los problemas que pueden aparecer.

Existen seis niveles de normalización, desde el más elemental llamado primera forma normal (1FN) hasta el máximo nivel de normalización, llamado quinta forma normal (5FN), después de la 3FN existe una llamada FNBC (Forma Normal Boyce Codd, debidoa quienes implementaron esta, que es una extensión de la 3FN). Podríamos decir entonces que las formas normales existentes son 1FN, 2FN, 3FN, FNBC, 4FN y 5FN. También es cierto que una base de datos está “relativamente” correcta si logra alcanzar la 3FN (o también podríamos decir que es el mínimo aceptable).

**Primera Forma Normal (1FN).**

Tiene directa relación con la cantidad de atributos que posee una relación (registro), por ejemplo: un atributo de un registro puede tener un solo valor posible.

En la tabla de abajo, si bien es cierto es una tabla de datos, los valores posibles para el campo Teléfono tienen dos para cada trabajador, esto está completamente desnormalizado.

|  |  |  |
| --- | --- | --- |
| RUN | Nombre | Teléfonos |
| 22000123-k | Juan Castro | 999555000 |
| 998555001 |
| 16345345-0 | Luis Solar | 988000123 |
| 989898989 |

Si quisiéramos aplicar 1FN, cada atributo de cada registro debe tener solo un valor, esto quedaría de la siguiente manera:

|  |  |  |
| --- | --- | --- |
| RUN | Nombre | Teléfonos |
| 22000123-k | Juan Castro | 999555000 |
| 22000123-k | Juan Castro | 998555001 |
| 16345345-0 | Luis Solar | 988000123 |
| 16345345-0 | Luis Solar | 989898989 |

Se dice que una relación está en 1FN cuando cada atributo toma solo un valor de su dominio.

**Segunda forma normal (2FN).**

Se basa en la dependencia de los atributos no principales sobre los atributos principales. De tal manera que cada atributo no principal depende directamente de todos los atributos principales y no de un subconjunto de ellos.

Cuando la clave de una relación es simple, es decir, contiene solo un atributo, entonces la relación está en 2FN.

Una relación está en 2FN, cuando está en 1FN y además cumple los requisitos para la 2FN. Veamos el siguiente ejemplo: suponga la relación VENTAS\_TRABAJADOR (A, D), donde:

A = {**RUN TRABAJADOR, CÓDIGO PRODUCTO**, FECHA VENTA, CARGO TRABAJADOR}

D = {**RUN TRABAJADOR, CÓDIGO PRODUCTO**  FECHA VENTA; **RUN TRABAJADOR**  CARGOTRABAJADOR}.

En el ejemplo el atributo FECHA VENTA depende de RUN TRABAJADOR y CÓDIGO PRODUCTO. De tal manera que asocia la fecha de venta con quién vendió y qué vendió. Y CARGO TRABAJADOR depende de RUN TRABAJADOR, tenemos acá entonces el caso en que el atributo CARGO depende solo de una parte de la clave (la clave es la combinación de RUT TRABAJADOR y CÓDIGO PRODUCTO)

|  |  |  |  |
| --- | --- | --- | --- |
| **RUN** | **CÓDIGO PRODUCTO** | **FECHA VENTA** | **CARGO TRABAJADOR** |
| 12123123-K | 780100012345-0 | 01-12-2017 | VENDEDOR TERRENO |
| 12123123-K | 780100012555-1 | 02-12-2017 | VENDEDOR TERRENO |
| 20333222-0 | 780100012345-0 | 03-12-2017 | VENDEDOR MESÓN |
| 20333222-0 | 780100012555-1 | 03-12-2017 | VENDEDOR TERRENO |

Las inconsistencias que se podrían producir es que haya ventas para un mismot rabajador con cargos distintos. También podríamos decir en esta relación que un vendedor no podría vender el mismo producto más de una vez, pero este tipo de problemas lo abordaremos más adelante.

La relación VENTAS\_TRABAJADOR descrita más arriba, no está en 2FN. Para lograr que esté en 2FN, es necesario realizar las siguientes transformaciones:

VENTA (A1, D1) y TRABAJADOR (A2, D2), donde:

A1 = {**RUN TRABAJADOR, CÓDIGO PRODUCTO**, FECHA VENTA} A2 = {**RUN TRABAJADOR**, CARGO}

D1 = {**RUN TRABAJADOR, CÓDIGO PRODUCTO**  FECHA VENTA} D2= {**RUN TRABAJADOR**  CARGO}

Dividimos la primera relación VENTAS\_TRABAJADOR, en dos que sí cumplen con la restricción de que los atributos que no son clave dependan directamente de los atributos clave. Las nuevas relaciones VENTA y TRABAJADOR están en 2FN.

|  |  |  |
| --- | --- | --- |
| **RUN** | **CÓDIGO PRODUCTO** | **FECHA VENTA** |
| 12123123-K | 780100012345-0 | 01-12-2017 |
| 12123123-K | 780100012555-1 | 02-12-2017 |
| 20333222-0 | 780100012345-0 | 03-12-2017 |
| 20333222-0 | 780100012555-1 | 03-12-2017 |

|  |  |
| --- | --- |
| **RUN** | **CARGO TRABAJADOR** |
| 12123123-K | VENDEDOR TERRENO |
| 20333222-0 | VENDEDOR MESÓN |

De esta forma, resolvemos el problema de los dos cargos del trabajador, sin embargo, aún nos queda pendiente el problema de poder registrar varias ventas para un mismo producto.

**Tercera forma normal (3FN)**

Una relación está en 3FN si está en 2FN y además, cumple con que ningún atributo no principal dependa transitivamente de una clave de la relación. Por ejemplo:

La relación TRABAJADOR (A1, D1), donde:

A1 = {**RUN\_TRABAJADOR**, CÓDIGO SUCURSAL, NOMBRE SUCURSAL}

D1 = {**RUN\_TRABAJADOR**  CÓDIGO SUCURSAL; CÓDIGO SUCURSAL  NOMBRE SUCURSAL}

|  |  |  |
| --- | --- | --- |
| TRABAJADOR | | |
| RUN | CÓDIGO SUCURSAL | NOMBRE SUCURSAL |
| 12345678-0 | 100 | SANTA MARÍA |
| 10111222-K | 200 | TARAPACÁ |
| 20332445-9 | 300 | VICTORIA |

En este caso, el código de la sucursal depende de la clase RUN del trabajador, y el nombre de la sucursal depende del atributo no clave nombre de la sucursal. Podríamos decir que el nombre de la sucursal depende transitivamente del RUN del trabajador, por lo que esta relación no está en 3FN.

Al igual que en el punto anterior, podemos transformar las relaciones para que estén en 3FN, de la siguiente manera:

TRABAJADOR (A1, D1) y SUCURSAL (A2, D2), donde:

A1= {**RUN**, SUCURSAL} y D1 = {**RUN**  SUCURSAL}

A2= {**SUCURSAL**, NOMBRE SUCURSAL} y D2= {**SUCURSAL**  NOMBRE SUCURSAL}

|  |  |
| --- | --- |
| TRABAJADOR | |
| RUN | CÓDIGO SUCURSAL |
| 12345678-0 | 100 |
| 10111222-K | 200 |
| 20332445-9 | 300 |

|  |  |
| --- | --- |
| SUCURSAL | |
| CÓDIGO SUCURSAL | NOMBRE SUCURSAL |
| 100 | SANTA MARÍA |
| 200 | TARAPACÁ |
| 300 | VICTORIA |

**Forma normal de Boyce-Codd (BCFN).**

Las tres formas normales anteriores eliminan las dependencias parciales y transitivas de los atributos, la BCFN elimina dependencias funcionales entre ellos. Cuando una relación tiene dos o más claves candidatas y los atributos generan dependencia de cualquiera de las claves candidatas podríamos decir que está en 3FN, pero no en BCFN. Por ejemplo:

TRABAJADOR (A1, D1). Donde:

A1 = {**ID\_TRABAJADOR**, RUN, NOMBRE}; D1 = {ID\_TRABAJADOR, RUN  NOMBRE}

|  |  |  |
| --- | --- | --- |
| TRABAJADOR | | |
| **ID\_TRABAJADOR** | **RUN** | **NOMBRE** |
| 100 | 12345678-0 | JUAN PÉREZ |
| 200 | 10111222-K | DANIEL BOONE |

La relación TRABAJADOR está en 3FN, pero podría tener inconsistencias de actualización debido a que un trabajador podría tener dos ID diferentes con el mismo RUN. Para normalizar esto según BCFN, debiéramos separar en dos, de la siguiente manera:

TRABAJADOR (A1, D1) y TRABAJADOR\_DATOS (A2, D2), donde:

A1 = {ID\_TRABAJADOR, RUN}; D1 = {ID\_TRABAJADOR  RUN}

A2= {RUN, NOMBRE}; D2 = {RUN  NOMBRE}

|  |  |
| --- | --- |
| **TRABAJADOR** | |
| **ID\_TRABAJADOR** | **RUN** |
| 100 | 12345678-0 |
| 200 | 10111222-K |

|  |  |
| --- | --- |
| **TRABAJADOR\_DATOS** | |
| **RUN** | **NOMBRE** |
| 12345678-0 | JUAN PÉREZ |
| 10111222-K | DANIEL BOONE |

Cuarta forma normal 4FN.

La cuarta forma normal impide que se produzcan redundancias en los atributos multivaluados y que sus dependencias sean eficientes. Como ya vimos anteriormente un atributo multivaluado es aquel que puede tener varias instancias posibles. En una tabla con atributos multivaluados y dependencias multivaluadas se produce redundancia (no deseada). Por ejemplo:

SUCURSAL (A1, D1), donde:

A1 = {NOMBRE SUCURSAL, TIPO DE PRODUCTO, ZONA DE DESPACHO}

D1= {NOMBRE SUCURSAL  TIPO DE PRODUCTO, ZONA DE DESPACHO; TIPO DE PRODUCTO  NOMBRE SUCURSAL, ZONA DE DESPACHO; ZONA DE DESPACHO  NOMBRE SUCURSAL, TIPO DE PRODUCTO}

Existe una dependencia multivaluada, todos los atributos son parte de la clave, por lo que al agregar una nueva característica, por ejemplo, una nueva zona de despacho, habrá que repetir el registro por cada sucursal y por cada tipo de producto de cada sucursal.

|  |  |  |
| --- | --- | --- |
| **SUCURSAL** | **TIPO** | **ZONA** |
| CONCEPCIÓN | ELECTRÓNICA | NORTE |
| CONCEPCIÓN | ELECTRÓNICA | SUR |
| CONCEPCIÓN | LÍNEA BLANCA | NORTE |
| CONCEPCIÓN | LÍNEA BLANCA | SUR |

Si se agrega una nueva sucursal, se va a repetir el tipo y la zona de despacho:

|  |  |  |
| --- | --- | --- |
| **SUCURSAL** | **TIPO** | **ZONA** |
| CONCEPCIÓN | ELECTRÓNICA | NORTE |
| CONCEPCIÓN | ELECTRÓNICA | SUR |
| CONCEPCIÓN | LINEA BLANCA | NORTE |
| CONCEPCIÓN | LÍNEA BLANCA | SUR |
| CHILLÁN | ELECTRÓNICA | NORTE |
| CHILLÁN | ELECTRÓNICA | SUR |
| CHILLÁN | LÍNEA BLANCA | NORTE |
| CHILLÁN | LÍNEA BLANCA | SUR |

Esta tabla está en 3FN pero no está en 4FN para solucionar el problema de redundancia, hay que realizar la siguiente transformación:

|  |  |
| --- | --- |
| **SUCURSAL\_TIPO** | |
| **SUCURSAL** | **TIPO** |
| CONCEPCIÓN | ELECTRÓNICA |
| CONCEPCIÓN | LÍNEA BLANCA |
| CHILLÁN | ELECTRÓNICA |
| CHILLÁN | LÍNEA BLANCA |

|  |  |
| --- | --- |
| **SUCURSAL\_DESPACHO** | |
| **SUCURSAL** | **DESPACHO** |
| CONCEPCIÓN | SUR |
| CONCEPCIÓN | NORTE |
| CHILLÁN | SUR |
| CHILLÁN | NORTE |

Quinta forma normal (5FN).

La quinta forma normal también es conocida como proyección – unión, esto debido a que los datos se obtienen por medio de la unión (JOIN) entre varias tablas y la proyección de los datos significativos de ellas.

Considere el siguiente ejemplo, donde en una tabla de asignación de ciudades y productos a un vendedor, que representa por ejemplo que el vendedor 12345678-0 vende galletas y helados en la ciudad de Santiago, está configurada de la siguiente manera:

|  |  |  |
| --- | --- | --- |
| **VENDEDOR** | **PRODUCTO** | **CIUDAD** |
| JUAN PEREZ | GALLETAS | ANTOFAGASTA |
| JUAN PEREZ | GALLETAS | CHILLÁN |
| SOFIA VERGARA | HELADOS | OSORNO |
| SOFIA VERGARA | HELADOS | ANTOFAGASTA |
| SOFIA VERGARA | HELADOS | CHILLÁN |
| LUIS RIFFO | HELADOS | VALPARAÍSO |
| LUIS RIFFO | GALLETAS | ANTOFAGASTA |
| LUIS RIFFO | GALLETAS | SANTIAGO |
| LUIS RIFFO | DULCES | VALDIVIA |

Si queremos optimizar esta tabla dejándola en 5FN es necesario dividirla en tres sub tablas, que conserven las dependencias, de la siguiente manera:

En la primera, una tabla que contenga los vendedores y las ciudades a las que atiende:

|  |  |
| --- | --- |
| **VENDEDOR** | **CIUDAD** |
| JUAN PEREZ | ANTOFAGASTA |
| JUAN PEREZ | CHILLÁN |
| SOFIA VERGARA | OSORNO |
| SOFIA VERGARA | ANTOFAGASTA |
| SOFIA VERGARA | CHILLÁN |
| LUIS RIFFO | VALPARAÍSO |
| LUIS RIFFO | ANTOFAGASTA |
| LUIS RIFFO | SANTIAGO |
| LUIS RIFFO | VALDIVIA |

En la segunda, una tabla que contenga los productos y las ciudades en las que se vende, de la siguiente manera:

|  |  |
| --- | --- |
| **PRODUCTO** | **CIUDAD** |
| GALLETAS | ANTOFAGASTA |
| GALLETAS | CHILLÁN |
| GALLETAS | SANTIAGO |
| HELADOS | OSORNO |
| HELADOS | ANTOFAGASTA |
| HELADOS | CHILLÁN |
| DULCES | VALDIVIA |

Por último, una tabla donde tengamos a los vendedores y los productos que venden.

|  |  |
| --- | --- |
| **VENDEDOR** | **PRODUCTO** |
| JUAN PEREZ | GALLETAS |
| SOFIA VERGARA | HELADOS |
| LUIS RIFFO | HELADOS |
| LUIS RIFFO | GALLETAS |
| LUIS RIFFO | DULCES |

De esta manera, si agregamos, por ejemplo, una nueva ciudad para Luis Riffo, digamos Chillán, entonces estamos indicando, según el modelo, que Luis Riffo ahora además, venderá galletas y helados en Chillán; si hubiéramos usado la estructura original, habría que hacer dos registros nuevos.