Logotipo

El contenido generado por IA puede ser incorrecto.



Santiago Yosa González

**Ingeniero Sistemas de Información**

Logotipo

El contenido generado por IA puede ser incorrecto.}

* Normalización de una base de datos (formas normales).
* 1ª Forma Normal
* 2ª Forma Normal
* 3ª Forma Normal
* Aplicación práctica de normalización

**Unidad I**

**Normalización de una base de datos**

**Situación**

Imagina que trabajas en una ferretería y el sistema actual guarda las ventas en una sola "Súper Tabla" de Excel o SQL llamada RegistroVentas.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID\_Venta | Cliente | Direccion\_Cliente | Productos\_Comprados | Total | Vendedor | Sucursal\_Vendedor |
| 101 | Juan Pérez | Calle 123 | Martillo, Clavos (50), Sierra | 50000 | Ana López | Centro |
| 102 | Juan Pérez | Calle 123 | Taladro | 80000 | Carlos Ruiz | Norte |
| 103 | María Gómez | Cra 45 | Martillo | 15000 | Ana López | Centro |

**El Problema:**

1. **Redundancia de Datos:** "Ana López" y "Centro" se repiten. Si Ana se muda a la sucursal Sur, ¿tienes que actualizar 1,000 filas antiguas? (Anomalía de Actualización).
2. **Atomicidad Violada:** En Productos\_Comprados tenemos una lista separada por comas. ¿Cómo haces una consulta SQL para saber *cuántos* martillos se vendieron? Es imposible/ineficiente con esa celda sucia.
3. **Inconsistencia:** Si escribes "Calle 123" en una fila y "Cll 123" en otra, para la base de datos son dos direcciones distintas.

**¿Qué es la normalización?**

La normalización es un proceso utilizado para organizar los datos en una base de datos. Su objetivo es minimizar la redundancia (datos duplicados) y mejorar la integridad de los datos.

**Beneficios de la normalización:**

* Evita la duplicidad de información.
* Facilita el mantenimiento de los datos.
* Aumenta la consistencia y precisión.
* Mejora el rendimiento en algunas consultas.

Es un proceso de optimización en bases de datos que busca reducir redundancias y anomalías dividiendo tablas grandes en estructuras más pequeñas y relacionadas.

Se incluye la creación de tablas y el establecimiento de relaciones entre ellas según reglas diseñadas tanto para proteger los datos como para hacer que la base de datos sea más flexible al eliminar la redundancia y las dependencias incoherentes.

En términos más sencillos, consiste en descomponer una tabla grande y compleja en tablas más pequeñas y sencillas, manteniendo las relaciones entre los datos.

**Fuentes:**

[Fundamentos de la normalización de bases de datos](https://learn.microsoft.com/es-es/office/troubleshoot/access/database-normalization-description)

**Integridad de los datos**

Imagina una base de datos que contiene información sobre los clientes. Sin normalización, si un cliente cambia de edad, tendríamos que actualizarla en varios sitios, lo que aumentaría el riesgo de incoherencias. Al normalizar los datos, podemos tener tablas separadas vinculadas por un identificador único que garantizará que los datos sigan siendo precisos y coherentes.

**Consulta de eficacia**

Consideremos una base de datos compleja con múltiples tablas relacionadas que almacena información redundante. En este escenario, las consultas que implican uniones se vuelven más complicadas y consumen más recursos. La normalización ayudará a simplificar las consultas al dividir los datos en tablas más pequeñas, en las que cada tabla contendrá sólo la información relevante, reduciendo así la necesidad de complejas uniones.

**Optimización del almacenamiento**

Un problema importante de los datos redundantes es que ocupan espacio de almacenamiento innecesario. Por ejemplo, si almacenamos los mismos detalles del producto en cada registro de pedido, se produce una duplicación. Con la normalización, puedes eliminar la redundancia dividiendo los datos en tablas separadas.

**Requisitos de la normalización**

Para que las tablas de nuestra BD estén normalizadas deben cumplir las siguientes reglas:

* Cada tabla debe tener su nombre único.
* No puede haber dos filas iguales.
* No se permiten los duplicados.
* Todos los datos en una columna deben ser del mismo tipo.

**Tabla Inicial no normalizada: Sistema de Pedidos de una Tienda**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Factura | Fecha | Cliente | Dirección | Productos | Vendedor | Zona\_Vendedor |
| F001 | 01/02 | Juan Pérez | Cl 10 #20 | Leche, Pan, Huevos | Ana | Norte |
| F002 | 01/02 | María Gil | Cr 5 #80 | Leche | Ana | Norte |
| F003 | 02/02 | Juan Pérez | Calle 10 Num 20 | Arroz, Pan | Carlos | Sur |

**Primera Forma Normal (1FN) - Atomicidad**

* Cada celda de la tabla debe contener un valor único y atómico (indivisible).
* No puede haber grupos repetidos (listas) en una columna.
* Debe existir una Llave Primaria (PK)
* Cada columna debe tener un solo tipo de dato.

**¿Qué errores lógicos encontramos? (no de software, sino de información).**

**La Celda "Sucia":** En la fila 1, columna "Productos", hay tres cosas a la vez.

¿Si quiero saber cuánto Pan vendí en total, cómo sumo esa celda?

**La Inconsistencia:** Juan Pérez aparece dos veces. En la primera dice "Cl 10" y en la segunda "Calle 10".

Para nosotros es comprensible, es lo mismo, pero para una máquina son dos direcciones distintas.

**La Redundancia:** "Ana" siempre es del "Norte".

Si Ana se cambia de zona, ¿tengo que buscar las 5,000 facturas viejas de Ana y cambiarlas una por una?

**La Transformación:**

Para arreglar la tabla de arriba, debemos "explotar" la lista de productos. Si Juan compró 3 cosas, Juan tendrá 3 filas.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Factura | Fecha | Cliente | Dirección | Productos | Vendedor | Zona\_Vendedor |
| F001 | 01/02 | Juan Pérez | Cl 10 #20 | Pan | Ana | Norte |
| F001 | 01/02 | Juan Pérez | Cl 10 #20 | Leche | Ana | Norte |
| F001 | 01/02 | Juan Pérez | Cl 10 #20 | Huevos | Ana | Norte |
| F002 | 01/02 | María Gil | Cr 5 #80 | Leche | Ana | Norte |
| F003 | 02/02 | Juan Pérez | Calle 10 Num 20 | Arroz | Carlos | Sur |
| F003 | 02/02 | Juan Pérez | Calle 10 Num 20 | Pan | Carlos | Sur |

**El concepto de Llave Primaria (PK):**

Ahora, Factura (F001) ya no es único, se repite 3 veces. Para identificar una fila única necesitamos la combinación **Factura + Producto**. A esto le llamamos **Llave Compuesta**.

**Aplicando la Segunda Forma Normal (2FN)**

* Debe estar en 1FN.
* Elimina dependencias parciales: ningún campo no clave debe depender solo de una parte de la clave primaria compuesta
* Todo atributo debe hablar de la **Llave Completa**. Si una columna solo describe a *una parte* de la llave, debe irse a otra tabla.

Nuestra llave es Factura + Producto.

* **Columna Cantidad:** ¿Depende de la Factura y el Producto? Sí. (Es cuánta leche llevé en esa factura). -> **SE QUEDA.**
* **Columna Cliente:** ¿El nombre del cliente depende del Producto? No. El cliente es dueño de la Factura, sin importar qué producto lleve. -> **SE VA.**

**La Transformación:**

Separamos en dos tablas.

Tabla 1: DETALLE (Lo que pasó)

|  |  |  |
| --- | --- | --- |
| Factura | Producto | Cantidad |
| F001 | Leche | 1 |
| F001 | Pan | 1 |

Tabla 2: ENCABEZADO (Quién y Cuándo)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Factura | Cliente | Fecha | Vendedor | Zona\_Vendedor |
| F001 | Juan | 01/02 | Ana | Norte |

**Aplicación de 3FN (eliminar dependencias transitivas):**

* Estar en 2FN.
* Elimina **dependencias transitivas**: El atributo debe depender de la llave, toda la llave y nada más que la llave.
* (A depende de B, y B depende de C)

Miremos la Tabla 2 (Encabezado). La llave es Factura.

* Vendedor depende de la Factura (quién la atendió).

**Zona\_Vendedor:**

**¿La zona depende de la factura?**

No, la zona depende de **Ana**. Ana trabaja en el Norte. Si Ana atiende la factura 500, seguirá siendo del Norte.

* Esto es una **Dependencia Transitiva**.

**La Transformación**

Sacamos al vendedor a su propia tabla maestra

Tabla 3: VENDEDORES

|  |  |
| --- | --- |
| Vendedor | Zona |
| Ana | Norte |
| Carlos | Sur |

Tabla 2 (Refinada): ENCABEZADO

|  |  |  |  |
| --- | --- | --- | --- |
| Factura | Cliente | Fecha | Vendedor\_ID |
| F001 | Juan | 01/02 | Ana |

**Otro Ejemplo de Normalización**

**Primera forma normal (1NF)**

Este nivel de normalización garantiza que cada columna de tus datos contenga sólo valores atómicos.

**Requisitos:**

* Eliminar grupos repetitivos.
* Tener valores atómicos (no listas ni conjuntos en una sola celda).
* Cada columna debe tener un solo tipo de dato.

**Tabla inicial**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ClienteNombre | ClienteTel | Productos | Cantidades | PreciosUnitarios | FechaPedido |
| Ana Pérez | 3114567890 | Teclado, Mouse | 1,2 | 100000,50000 | 2024-05-10 |
| Luis Gómez | 3001234567 | Monitor | 1 | 600000 | 2024-05-11 |
| Ana Pérez | 3114567890 | Teclado, Webcam | 1,1 | 100000,150000 | 2024-05-12 |
| Carlos Ríos | 3129876543 | Monitor, Mouse, Teclado | 1,1,1 | 600000,50000,100000 | 2024-05-13 |

En la tabla tenemos las siguientes situaciones a resolver:

No existe un dato único que identifique cada una de las filas, una llave primaria.

Porque contiene **campos multivaluados**:

El campo **Productos** tiene **más de un valor** en la misma celda (ej. "Teclado, Mouse").

Lo mismo ocurre con **Cantidades** y **PreciosUnitarios**.

Solucionando y aplicando tenemos:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PedidoID | ClienteNombre | ClienteTel | Producto | Cantidad | PrecioUnitario | FechaPedido |
| 1 | Ana Pérez | 3114567890 | Teclado | 1 | 100000 | 2024-05-10 |
| 1 | Ana Pérez | 3114567890 | Mouse | 2 | 50000 | 2024-05-10 |
| 2 | Luis Gómez | 3001234567 | Monitor | 1 | 600000 | 2024-05-11 |
| 3 | Ana Pérez | 3114567890 | Teclado | 1 | 100000 | 2024-05-12 |
| 3 | Ana Pérez | 3114567890 | Webcam | 1 | 150000 | 2024-05-12 |
| 4 | Carlos Ríos | 3129876543 | Monitor | 1 | 600000 | 2024-05-13 |
| 4 | Carlos Ríos | 3129876543 | Mouse | 1 | 50000 | 2024-05-13 |
| 4 | Carlos Ríos | 3129876543 | Teclado | 1 | 100000 | 2024-05-13 |

**Aplicando la Segunda Forma Normal (2FN)**

Criterio de 2FN:

* Debe estar en 1FN.
* Elimina dependencias parciales: ningún campo no clave debe depender solo de una parte de la clave primaria compuesta.

En este caso, la clave compuesta sería: (PedidoID, Producto)

Pero:

* ClienteNombre, ClienteTel y FechaPedido dependen **solo** de PedidoID.

Entonces **separamos en tablas.** Con esto:

* Se eliminan dependencias parciales.
* Cada tabla cumple un único propósito.
* Ya estaríamos en **2FN**.

|  |  |  |  |
| --- | --- | --- | --- |
| PedidoID | ProductoID | Cantidad | PrecioUnitario |
| 1 | 1 | 1 | 100000 |
| 1 | 2 | 2 | 50000 |
| 2 | 3 | 1 | 600000 |
| 3 | 1 | 1 | 100000 |
| 3 | 4 | 1 | 150000 |

|  |  |
| --- | --- |
| ProductoID | NombreProducto |
| 1 | Teclado |
| 2 | Mouse |
| 3 | Monitor |
| 4 | Webcam |

|  |  |  |
| --- | --- | --- |
| PedidoID | ClienteID | FechaPedido |
| 1 | 1 | 2024-05-10 |
| 2 | 2 | 2024-05-11 |
| 3 | 1 | 2024-05-12 |

|  |  |  |
| --- | --- | --- |
| ClienteID | ClienteNombre | ClienteTel |
| 1 | Ana Pérez | 3114567890 |
| 2 | Luis Gómez | 3001234567 |

**Aplicando la Tercera Forma Normal (3FN)**

**Criterio de 3FN:**

* Estar en 2FN.
* Elimina **dependencias transitivas**: un campo no clave no debe depender de otro campo no clave.

En nuestro ejemplo ya **no hay campos no-clave que dependan de otros campos no-clave**.

Por lo tanto, el modelo ya cumple con **3FN**.

**Preguntas Frecuentes**

**Al dividir la tabla en tres pedazos, ¿no se hace más difícil leer la información?**

Para un humano, sí. Para el sistema, es perfecto. El sistema une las tablas en milisegundos cuando lo necesites. Lo que ganamos es que si Ana se cambia de zona, solo modificamos *una* celda en la Tabla 3, y automáticamente todas las facturas quedan actualizadas. Priorizamos la **integridad** sobre la legibilidad humana directa.

**¿Cómo sé cuándo parar de dividir?**

Cuando cada tabla habla de una sola cosa.

* Una tabla solo habla de Productos.
* Otra solo habla de Clientes.
* Otra solo habla de Ventas. Si en la tabla de "Ventas" tienes la "Fecha de nacimiento del vendedor", te has detenido demasiado pronto.

**Demo**

**Caso:** Sistema de Notas Escolares. **Estado Inicial (UNF):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Santiago | Materia: Matemáticas | Profesor: Sr. X | Oficina\_Prof: 202 | Nota\_Final: 4.5 |
| Santiago | Materia: Historia | Profesor: Sra. Y | Oficina\_Prof: 101 | Nota\_Final: 3.0 |
| Santiago | Materia: Física | Profesor: Sr. Z | Oficina\_Prof: 303 | Nota\_Final: 3.5 |

**Paso 1: Aplicar 1FN (Tablas Planas)** Imaginemos que Santiago ve 3 materias. La tabla se ve así:

1. Santiago - Matemáticas - Sr. X - Of 202 - 4.5
2. Santiago - Historia - Sra. Y - Of 101 - 3.0
3. Santiago - Física - Sra. Z - Of 101 - 3.5

**Paso 2: Aplicar 2FN (Dependencias Parciales)** Llave: Estudiante + Materia.

¿El Profesor depende del Estudiante? No. Depende de la Materia.

**Acción:**

Sacamos las materias aparte.

* *Tabla Materias:*

|  |  |  |
| --- | --- | --- |
| Materia | Profesor | Oficina\_Prof |

* *Tabla Notas:*

|  |  |  |
| --- | --- | --- |
| Estudiante | Materia | Nota |

**Paso 3:** Aplicar 3FN (Dependencias Transitivas)

Miremos la tabla *Materias*:

|  |  |  |
| --- | --- | --- |
| Materia | Profesor | Oficina\_Prof |

La oficina depende del Profesor, no de la materia. Si el profesor da 5 materias, repetiremos "Oficina 202" 5 veces.

**Acción:**

Sacamos a los profesores.

|  |  |
| --- | --- |
| Profesor | Oficina |

|  |  |
| --- | --- |
| Materia | Profesor\_ID |

**Práctica Acompañada**

**Ejercicio: "La Clínica Médica"**

**Enunciado:** El hospital tiene este formato en Excel para las citas:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CitaID | Fecha | Paciente | Edad\_Paciente | Alergias\_Paciente | Doctor | Especialidad\_Doctor | Consultorio | Diagnostico |

**Instrucciones:**

Toma tijeras imaginarias. Dibuja en una hoja o en Excel cómo quedarían las tablas separadas aplicando las 3 formas normales.

***Pista:***

Un paciente tiene edad y alergias siempre, no solo en la cita.

El doctor tiene una especialidad siempre.

**Detectar Entidades Fuertes:** Paciente y Doctor.

* Tabla Pacientes: Paciente, Edad, Alergias.
* Tabla Doctores: Doctor, Especialidad, Consultorio (Asumiendo que el consultorio es fijo del doctor).

**Detectar el Evento (La relación):** La Cita.

* Tabla Citas: CitaID, Fecha, Paciente\_ID, Doctor\_ID, Diagnostico.

**Demo Independiente**

**Enunciado:**

Una empresa de transporte de carga registra sus envíos así:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Guía\_Envío | Fecha | Camión\_Placa | Marca\_Camión | Modelo\_Camión | Chofer | Licencia\_Chofer | Ciudad\_Origen | Ciudad\_Destino | Peso\_Carga |

**Instrucciones:** Normaliza hasta 3FN. Entrega el dibujo de las tablas resultantes con sus columnas.

**(Solución al final del ejercicio):**

* Tabla Camiones: Placa, Marca, Modelo.
* Tabla Choferes: Chofer, Licencia. (Ojo: Si un camión siempre lo maneja el mismo chofer, podría haber relación, pero asumamos que rotan -> Tabla independiente).
* Tabla Rutas (Opcional si se repiten mucho): Origen, Destino.
* Tabla Envios (La central): Guía, Fecha, Placa\_Camion, ID\_Chofer, Peso, Origen, Destino.

**Ejercicio Independiente - El Hotel**

**Contexto:** El "Hotel Paraíso" registra sus huéspedes en un libro de actas manual.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Habitación | Tipo\_Habitacion (Simple/Doble) | Precio\_Noche | Huesped\_Principal | Documento | Telefono | Fecha\_Llegada | Fecha\_Salida | Noches\_Total |

**Reto:** Diseña el esquema de tablas para digitalizar este hotel.

**Debes resolver:**

1. Que el precio de la habitación depende del tipo de habitación, no del huésped.
2. Que un huésped puede venir muchas veces (no repetir sus datos).
3. Calcula si "Noches\_Total" debería guardarse o calcularse (Pista: Dato calculado vs Dato almacenado).

**Criterios de Logro:**

* Creación de Tabla Huespedes.
* Creación de Tabla Habitaciones (con su tipo y precio base).
* Creación de Tabla Reservas o Estadías que una todo.
* Eliminación de columnas calculadas (Noches\_Total no se guarda, se calcula restando fechas en el reporte).

Logotipo

El contenido generado por IA puede ser incorrecto.