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**

Volvemos a nuestra tienda. El dueño te pide: *"Quiero saber qué producto se vende más los lunes por la mañana en la sucursal Norte, pero solo de clientes mayores de 30 años"*.

Tienes tu Excel gigante con 50,000 filas.

* Intentas filtrar por "Lunes"... pero las fechas están como texto: "Lunes 10", "10/02/2025".
* Intentas filtrar por "Norte"... pero a veces dice "Sucursal Nte", "Norte", "Sede 1".
* Intentas saber la edad... pero solo tienes el nombre.

**La Reflexión:**

Una base de datos no solo sirve para **guardar** datos, sirve para **responder preguntas** (Queries). Si no normalizamos y estandarizamos, no podemos responder preguntas de negocio.

Ya sabemos *cortar* tablas (Normalizar). Ahora aprenderemos a *nombrarlas* y definirlas profesionalmente.

**Convenciones de Nomenclatura (Naming Conventions)**

Cuando pasemos nuestros dibujos a un entorno profesional, no puedes llamar a la tabla "cosas\_que\_vendemos".

1. **Idioma:** Inglés o Español (Decisión de equipo). Nosotros usaremos Ingles.
2. **Singular vs Plural:**
   * *Tabla:* Clientes (Plural, contiene muchos clientes).
   * *Columna:* Nombre (Singular, es un solo valor).
3. **Formato:**
   * CamelCase: fechaNacimiento (Recomendado en C#/SQL Server).
   * snake\_case: fecha\_nacimiento (Recomendado en Python/Postgres).
   * **Regla:** ¡Sin espacios ni tildes! Dirección -> Direccion.

**Tipos de Datos (Conceptual)**

Aunque no escribamos código hoy, en el diagrama debemos definir qué *tipo* de dato va en la caja.

* **Texto:** Nombres, Direcciones, Códigos alfanuméricos.
* **Entero (INT):** Cantidades, Edades, Años. (Cosas que se cuentan).
* **Decimal/Moneda:** Precios, Salarios, Promedios. (Cosas que se miden).
* **Fecha/Hora:** Cumpleaños, Fecha de Registro.
* **Booleano (Bit):** Activo/Inactivo, Si/No.

**Llaves Naturales vs. Llaves Subrogadas (Surrogate Keys)**

* **Llave Natural:** Un dato que ya existe y es único (Ej: Cédula, ISBN del libro).

***Problema:*** Si el gobierno cambia el formato de la cédula, se rompe tu base de datos.

* **Llave Subrogada (ID):** Un número inventado por nosotros (1, 2, 3...) que no significa nada para el usuario, solo para el sistema.

***Recomendación:*** Usar siempre ID autoincrementable (ClienteID, ProductoID). Es más seguro y rápido.

**¿Almacenar el dato calculado o calcularlo cada vez? - La Regla General:**

Nunca almacenes lo que puedes calcular. En bases de datos transaccionales (OLTP), almacenar datos calculados se considera una mala práctica porque viola la Integridad de los Datos y la 3ª Forma Normal.

**El problema de la Desincronización**: Imagina que tienes una tabla Ventas con columnas: Cantidad, PrecioUnitario y Total.

* Fila: Cantidad: 2, Precio: 100, Total: 200.
* Mañana, alguien actualiza la Cantidad a 3.
* Si el programador olvida actualizar también el Total, la base de datos queda corrupta: Cantidad: 3, Precio: 100, Total: 200

**La Excepción - El Snapshot - Histórico**

Hay un caso donde **SÍ** debes guardar el valor calculado: **Facturación Histórica**.

* Si vendes un producto a $100 hoy, y mañana sube a $150, la factura de ayer NO debe cambiar.
* En este caso, guardas el PrecioHistorico en el detalle de la venta para "congelar" el valor en el tiempo.

**¿Dónde poner la Llave Foránea (FK)?**

Esta decisión depende exclusivamente de la **Cardinalidad** (1:1, 1:N, N:M).

**Caso A: Relación Uno a Muchos (1:N) – La mayoría de los casos**

La llave foránea **SIEMPRE va en el lado del "Muchos"**. Piensa en la relación Padre e Hijos.

* Un Padre tiene muchos Hijos.
* Un Hijo tiene un solo Padre (biológico).

**¿Dónde va la llave?** En la tabla **Hijos**. El hijo necesita saber quién es su padre. El padre no puede tener una columna "HijoID" porque tiene muchos hijos (violaría la 1FN).

* **Ejemplo:** Cliente (1) y Pedidos (N).
  + La tabla Pedidos lleva la columna ClienteID.

**Caso B: Relación Uno a Uno (1:1)**

Puede ir en cualquiera de las dos, pero se prefiere en la tabla **"Débil"** o la que se crea después.

* **Ejemplo:** Empleado y UsuarioDelSistema.
  + Un empleado tiene un usuario. Un usuario es de un empleado.
  + Generalmente, UsuarioDelSistema lleva la FK EmpleadoID, porque el empleado existe primero y el usuario es un accesorio.

**Caso C: Relación Muchos a Muchos (N:M)**

**En ninguna de las dos** Se crea una **Tercera Tabla** (Tabla de Rompimiento o Intermedia).

* **Ejemplo:** Estudiantes y Materias.
* Tabla Inscripciones lleva EstudianteID y MateriaID.

**Demo - El Videoclub (Streaming)**

Vamos a diseñar la base de datos para una app tipo Netflix.

**Datos Crudos (Lo que el cliente cree que es una tabla):**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Usuario | Email | Películas\_Vistas | Actores\_Favoritos | Plan\_Suscripción | Precio\_Plan | Fecha\_Pago |

**Paso 1: Normalización**

1. **1FN:** Películas\_Vistas es una lista. **¡Cuidado!**

Debemos separar. Un usuario ve *muchas* películas. Una película es vista por *muchos* usuarios.

1. **2FN y 3FN:**
   * Precio\_Plan depende del Plan\_Suscripción. -> Tabla Planes.
   * Los Actores no dependen del usuario, dependen de la Película. -> Tabla Actores.

**Modelado (El Diagrama Final) -** Vamos a dibujar las "tablas" con sus tipos de datos.

**Tabla: Usuarios**

* UsuarioID (PK - Entero)
* Email (Texto)
* Nombre (Texto)
* PlanID (FK - Entero) -> Relación con Planes.

**Tabla: Planes**

* PlanID (PK - Entero)
* Nombre (Texto - Ej: "Premium")
* Precio (Moneda)

**Tabla: Peliculas**

* PeliculaID (PK - Entero)
* Titulo (Texto)
* Año (Entero)
* DuracionMinutos (Entero)

**Tabla: HistorialVistas** (La tabla que rompe la relación M:N)

* HistorialID (PK - Entero)
* UsuarioID (FK)
* PeliculaID (FK)
* FechaVisualizacion (Fecha)
* MinutoDondeQuedo (Entero)

**Práctica - Taller de E-Commerce**

Vamos a diseñar el cerebro de una tienda de ropa online. Toma una hoja de papel horizontal.

**Requerimientos**

1. Tenemos **Ropa** (Camisas, Pantalones). Cada prenda tiene Nombre, Precio y Descripción.
2. OJO: Una misma camisa viene en **Tallas** (S, M, L) y **Colores** (Rojo, Azul).
3. Tenemos **Clientes** que hacen **Pedidos**.
4. Un pedido puede tener varias prendas.
5. Necesitamos manejar el **Stock** (¿Cuántas camisas Rojas talla M me quedan?).

**Solución**

Aquí la trampa está en el punto 2 y 5. Si pones "Color" en la tabla "Producto", tendrías que crear "Camisa Polo Roja", "Camisa Polo Azul". Eso repite datos.

1. **Entidad ProductosBase:** (Nombre: "Camisa Polo", Desc, PrecioBase).
2. **Entidad Variantes o Inventario:** Aquí ocurre la magia.
   * InventarioID
   * ProductoID (FK)
   * Talla (Texto)
   * Color (Texto)
   * CantidadStock (Entero)

***Explicación:*** Así, el producto "Camisa Polo" existe una vez, pero tiene muchas variantes en el inventario.

1. **Entidad Pedidos:** (Fecha, Cliente).
2. **Entidad DetallePedido:** (PedidoID, InventarioID, Cantidad, PrecioUnitario).

**Ejercicio Independiente**

Diseño Completo de Base de Datos Académica. **Herramienta:** Papel, Draw.io o Excel (Tablas separadas).

**Enunciado:** El colegio "San Knowledge" necesita organizar su información.

* Tienen **Alumnos** (Datos personales, Acudiente).
* Tienen **Profesores** (Datos personales, Profesión).
* Tienen **Grados/Cursos** (Primero A, Segundo B).
* Tienen **Materias** (Matemáticas, Español).
* **Reglas de Negocio:**
  + Un alumno pertenece a un solo Grado actual.
  + Un Grado tiene muchas Materias asignadas.
  + Una Materia en un Grado específico es dictada por un Profesor. (Ej: Matemáticas de 5to A la dicta el Profe Juan, pero Matemáticas de 5to B la dicta la Profe Ana).
  + Se necesitan guardar las **Notas** de cada alumno por materia y periodo (Periodo 1, 2, 3).

**Resultado del ejercicio:**

Diagrama de Tablas (Entidad-Relación) que muestre:

1. Nombre de la Tabla.
2. Columnas (Atributos) con su tipo de dato sugerido (Texto, Número).
3. Líneas que conecten las tablas (Relaciones).

**Pregunta de reflexión:**

Miren su diagrama del colegio. ¿Qué pasaría si el colegio decide que un alumno puede ver materias de grados diferentes (adelantar materias)?

***Respuesta****:* Nuestro diseño actual (Alumno -> Grado) se rompería. Tendríamos que cambiar la relación.

**Aprendizaje:** El diseño de la base de datos depende 100% de las **Reglas del Negocio**. Si el negocio cambia, la base de datos debe evolucionar.

**Instalación - El Motor (SQL Server Developer Edition)**

El "Motor" es el cerebro que guarda los datos. Sin esto, no hay base de datos. Ingresamos a la siguiente página:

<https://www.microsoft.com/es-es/sql-server/sql-server-downloads>

**Selección de Edición:**

* Bajen hasta encontrar las Descargas principales.
* Verán varias opciones, no enfocaremos en: "Developer" y "Express".
* **Elegimos:** **Developer**. (Es la versión completa Enterprise, pero gratuita para uso educativo/desarrollo).
* Clic en **Descargar la edición Enterprise Developer**.

**El Instalador**

* Ejecuten el archivo descargado (SQL202X-SSEI-Dev.exe).
* Les aparecerán 3 opciones: *Básica*, *Personalizada*, *Descargar medios*.
* Seleccionen **BÁSICA**. (Es la más rápida y configura todo lo necesario por defecto).

**Proceso de Instalación:**

* Clic en **Aceptar** los términos de licencia.
* Clic en **Instalar** (Dejen la ruta por defecto en C:).
* ***Esperar****:* El sistema descargará unos 1000 MB y luego instalará. Esto puede tardar unos minutos.

**¡NO CIERRES LA VENTANA AÚN!**

Cuando termine, verán una pantalla llena de rutas y nombres técnicos.

**IMPORTANTE:** En la parte inferior de esa ventana hay un botón que dice **"Instalar SSMS"**. Haremos clic ahí.

**La Interfaz (SQL Server Management Studio - SSMS)**

El SSMS es la herramienta visual con la que nosotros hablamos con el Motor.

1. **Descarga:**
   * Al hacer clic en el botón anterior, los llevará a una web de Microsoft.

<https://learn.microsoft.com/es-es/ssms/install/install>

Texto

El contenido generado por IA puede ser incorrecto.

1. **Instalación:**
   * Ejecuten el instalador (SSMS-Setup-ENU.exe).
   * Esta instalación es sencilla: Clic en **Install**.
   * En el paquete de idiomas, recomiendo que dejemos también el de inglés.
   * ***Nota:*** Puede pedir reiniciar el equipo al finalizar. Si lo pide, háganlo.

Logotipo

El contenido generado por IA puede ser incorrecto.