



Universidad Tecnológica Nacional  
Facultad Regional Buenos Aires

# Bases de datos

## Trabajo Práctico: Fábrica de sillones

### Estrategia

**Número de grupo:** 3

**Nombre del grupo:** LOS\_POLLOS\_HERMANOS

**Integrantes del grupo:**

Apellido	Nombre	Legajo
Herzkovich	Agustín	213.787-2
Piacentini	Nicolás	214.001-9
Sorrentino	Tadeo	214.153-0
Maturano	Gabriela	172.649-3

## **Índice**

<b>Modelo de datos.....</b>	<b>2</b>
<b>Entidades y sus relaciones.....</b>	<b>3</b>
(1) Ubicacion.....	3
(2) Cliente.....	3
(3) Sucursal.....	4
(4) Proveedor.....	5
(5) Factura.....	5
(6) Compra.....	6
(7) Pedido.....	6
(8) Medida.....	7
(9) Modelo.....	7
(10) Sillon.....	8
(11) DetallePedido.....	8
(12) TipoMaterial.....	9
(13) Material.....	9
(14) Tela.....	10
(15) Madera.....	11
(16) Relleno.....	11
(17) DetalleCompra.....	12
(18) PedidoCancelacion.....	12
(19) Envio.....	12
(20) DetalleFactura.....	13
(21) MaterialPorSillon.....	13
<b>Inconsistencias.....</b>	<b>14</b>
<b>Normalización.....</b>	<b>16</b>
<b>Diseño.....</b>	<b>17</b>

## Modelo de datos

Se creó un modelo de datos que organiza y normaliza los datos de la tabla provista por la cátedra. Para esto creamos 19 nuevas tablas con PK y FK para relacionarlas entre ellas. En el siguiente cuadro se detalla cómo se relacionan entre sí. Ejemplo: PK (1) se relaciona con FK (1)

Entidad	Primary Key	Foreign Key
Ubicacion	Ubicacion_Codigo (bigint) <b>(1)</b>	
Cliente	Cliente_Codigo (bigint) <b>(2)</b>	Cliente_Ubicacion (bigint) <b>(1)</b>
Sucursal	Sucursal_Codigo (bigint) <b>(3)</b>	Sucursal_Ubicacion (bigint) <b>(1)</b>
Proveedor	Proveedor_Codigo (bigint) <b>(4)</b>	Proveedor_Ubicacion (bigint) <b>(1)</b>
Factura	Factura_Numero (bigint) <b>(5)</b>	Factura_Cliente (bigint) <b>(2)</b> Factura_Sucursal (bigint) <b>(3)</b>
Compra	Compra_Numero (decimal(18,0)) <b>(6)</b>	Compra_Sucursal (bigint) <b>(3)</b> Compra_Proveedor (bigint) <b>(4)</b>
Pedido	Pedido_Numero (decimal(18,0)) <b>(7)</b>	Pedido_Cliente (bigint) <b>(2)</b> Pedido_Sucursal (bigint) <b>(3)</b>
Medida	Medida_Codigo (bigint) <b>(8)</b>	
Modelo	Modelo_Codigo (bigint) <b>(9)</b>	
Sillon	Sillon_Codigo (bigint) <b>(10)</b>	Sillon_Medida (bigint) <b>(8)</b> Sillon_Modelo (bigint) <b>(9)</b>
DetallePedido	Detalle_Pedido_Numero (bigint) <b>(11)</b>	Detalle_Pedido_Sillon (bigint) <b>(10)</b> Detalle_Pedido_Pedido (decimal(18,0)) <b>(7)</b>
TipoMaterial	TipoMaterial_Codigo (bigint) <b>(12)</b>	
Material	Material_Codigo (bigint) <b>(13)</b>	Material_Tipo (bigint) <b>(12)</b>
Tela	Tela_Codigo (bigint)	Tela_Material (bigint) <b>(13)</b>
Madera	Madera_Codigo (bigint)	Madera_Material (bigint) <b>(13)</b>
Relleno	Relleno_Codigo (bigint)	Relleno_Material (bigint) <b>(13)</b>
DetalleCompra	Detalle_Compra_Numero (bigint)	Detalle_Compra_Compra (decimal(18,0)) <b>(6)</b> Detalle_Compra_Material (bigint) <b>(13)</b>
PedidoCancelacion	Pedido_Cancelacion_Numero (bigint)	Pedido_Cancelacion_Pedido (decimal(18,0)) <b>(7)</b>
Envio	Envio_Numero (decimal(18,0))	Envio_Factura (bigint) <b>(5)</b>
DetalleFactura	Detalle_Factura_Numero (bigint)	Detalle_Factura_Factura (bigint) <b>(5)</b> Detalle_Factura_Detalle_Pedido (bigint) <b>(11)</b>
MaterialPorSillon	MaterialPorSillon_Codigo (bigint)	MaterialPorSillon_Material (bigint) <b>(13)</b> MaterialPorSillon_Sillon (bigint) <b>(10)</b>

## Entidades y sus relaciones

### (1) Ubicacion

Esta entidad representa la Ubicacion y contiene los atributos Provincia, Localidad y Direccion. Tiene relación con las entidades Proveedor, Cliente y Sucursal ya que en todas se requiere registrar estos datos.

- Relación con Sucursal: Una *Ubicacion* puede tener una *Sucursal* y una *Sucursal* tiene una *Ubicacion*.
- Relación con Proveedor: Una *Ubicacion* puede tener un *Proveedor* y un *Proveedor* tiene una *Ubicacion*.
- Relación con Cliente: Una *Ubicacion* puede tener un *Cliente* y un *Cliente* tiene una *Ubicacion*.

Atributo	Tipo de Dato	Tipo de Clave
Ubicacion_Codigo	bigint	Primary Key
Ubicacion_Provincia	nvarchar(255)	
Ubicacion_Localidad	nvarchar(255)	
Ubicacion_Direccion	nvarchar(255)	

### (2) Cliente

Esta entidad representa al cliente y contiene toda la información del cliente requerida para la realización de un pedido, facturación y gestión de envíos.

Creamos la PK *Cliente\_Codigo* para identificar de forma unívoca a la entidad *Cliente* ya que se encontraron casos en donde dos clientes comparten Dni en la tabla Maestra.

- Relación con Pedido: Cliente se relaciona con la entidad Pedido a través de la PK *Cliente\_Codigo*. En esta relación, Un *Pedido* tiene un único *Cliente* obligatorio pero *Cliente* tiene uno o muchos *Pedido*.
- Relación con Factura: Cliente se relaciona con la entidad Factura a través de la PK *Cliente\_Codigo*. En esta relación, una *Factura* tiene un *Cliente* y un *Cliente* puede tener muchas *Factura*.
- Relación con Ubicación: Cliente se relaciona con Ubicación a través de la FK *Cliente\_Ubicacion*, donde una *Ubicacion* puede tener un *Cliente* y un *Cliente* tiene una *Ubicacion*.

Atributo	Tipo de Dato	Tipo de Clave
Cliente_Codigo	bigint	Primary Key
Cliente_Ubicacion	bigint	Foreign Key
Cliente_Dni	bigint	
Cliente_Nombre	nvarchar(255)	
Cliente_Apellido	nvarchar(255)	
Cliente_Fecha_Nacimiento	datetime2(6)	
Cliente_Mail	nvarchar(255)	
Cliente_Telefono	nvarchar(255)	

### (3) Sucursal

La fábrica cuenta con varias sucursales. Esta entidad representa a cada una de ellas para poder registrar en cuál de ellas fue realizada cada factura, pedido o compra a proveedor.

- Relación con Factura: Sucursal se relaciona con Factura a través de la PK Sucursal\_Codigo. En esta relación, una *Factura* tiene una *Sucursal obligatoria* y una *Sucursal* puede tener muchas o ninguna *Factura*.
- Relación con Compra: Sucursal se relaciona con Compra a través de la PK Sucursal\_Codigo. En esta relación, una *Compra* tiene una *Sucursal* y una *Sucursal* puede tener muchas *Compra* o ninguna.
- Relación con Pedido: Sucursal se relaciona con pedido a través de la PK Sucursal\_Codigo. En esta relación, un *Pedido* tiene una única *Sucursal* y una *Sucursal* puede relacionarse con muchos *Pedido* o ninguno.
- Relación con Ubicación: Sucursal se relaciona con Ubicacion a través de la FK Sucursal\_Ubicacion. En esta relación, una *Ubicacion* puede tener una o ninguna *Sucursal* y una *Sucursal* tiene una única *Ubicacion*.

Atributo	Tipo de Dato	Tipo de Clave
Sucursal_Codigo	bigint	Primary Key
Sucursal_Ubicacion	bigint	Foreign Key
Sucursal_Telefono	nvarchar(255)	
Sucursal_Mail	nvarchar(255)	

#### (4) Proveedor

Esta entidad representa a cada proveedor al que se le realizan compras de los distintos tipos de materiales: tela, madera y relleno.

Si bien *Proveedor\_Cuit* no se repite, decidimos crear una Primary Key *Proveedor\_Codigo* para asegurarnos de manejarlos mejor y cerciorarnos de que en un futuro no se repitan.

- Relación con Compra: Proveedor se relaciona con Compra a través de su PK *Proveedor\_Codigo*. En esta relación una *Compra* tiene un *Proveedor* y un *Proveedor* puede tener muchas *Compra*.
- Relación con Ubicación: Proveedor se relaciona con Ubicación a través de su FK *Proveedor\_Ubicacion*. Una *Ubicacion* puede tener uno o ningún *Proveedor* y un *Proveedor* tiene una *Ubicación*.

Atributo	Tipo de Dato	Tipo de Clave
Proveedor_Codigo	bigint	Primary Key
Proveedor_Ubicacion	bigint	Foreign Key
Proveedor_Cuit	nvarchar(255)	
Proveedor_RazonSocial	nvarchar(255)	
Proveedor_Telefono	nvarchar(255)	
Proveedor_Mail	nvarchar(255)	

#### (5) Factura

Esta entidad representa la factura que se registra cuando el pedido está completado y el cliente realiza el pago. Todas las facturas registran la información contenida en esta entidad.

- Relación con DetalleFactura: Factura se relaciona con la entidad DetalleFactura a través de la PK *Factura\_Numero*, donde una *Factura* tiene al menos un *DetalleFactura* y un *DetalleFactura* tiene una única *Factura*.
- Relación con Envio: Factura se relaciona con la entidad Envio a través de la PK *Factura\_Numero*, donde una *Factura* tiene un *Envio* y un *Envio* tiene una *Factura*.
- Relación con Sucursal: Factura se relaciona con la entidad Sucursal a través de la FK *Factura\_Sucursal*, donde: una *Factura* tiene una *Sucursal* y una *Sucursal* puede tener cero o muchas *Factura*.
- Relación con Cliente: Factura se relaciona con un Cliente a través de la FK *Factura\_Cliente*, donde una *Factura* tiene un *Cliente* y un *Cliente* puede tener muchas *Factura*.

Atributo	Tipo de Dato	Tipo de Clave
Factura_Numero	bigint	Primary Key
Factura_Cliente	bigint	Foreign Key
Factura_Sucursal	bigint	Foreign Key
Factura_Fecha	datetime2(6)	
Factura_Total	decimal(38,2)	

## (6) Compra

Esta entidad representa cada compra que realiza la fábrica de sillones a sus proveedores. Por cada compra se registra la información contenida en esta entidad.

- Relación con DetalleCompra: Una *Compra* tiene uno o muchos *DetalleCompra* y un *DetalleCompra* tiene una *Compra* obligatoria.
- Relación con Proveedor: Una *Compra* tiene un *Proveedor* y un *Proveedor* tiene una o ninguna *Compra*.
- Relación con Sucursal: Una *Compra* tiene una *Sucursal* y una *Sucursal* tiene muchas o ninguna *Compra*.

Atributo	Tipo de Dato	Tipo de Clave
Compra_Numero	decimal(18,0)	Primary Key
Compra_Sucursal	bigint	Foreign Key
Compra_Proveedor	bigint	Foreign Key
Compra_Fecha	datetime2(6)	
Compra_Total	decimal(18,2)	

## (7) Pedido

Esta entidad representa a cada pedido que realiza un cliente

- Relación con PedidoCancelacion: Un *Pedido* puede tener un *PedidoCancelacion* o ninguno y un *PedidoCancelacion* está asociado sí o sí a un *Pedido* que ya existe.
- Relación con DetallePedido: Un *Pedido* tiene uno o muchos *DetallePedido* ya que se realiza por *Sillon* y un *DetallePedido* está asociado a un único *Pedido* obligatorio.
- Relación con Cliente: Un *Pedido* tiene un único *Cliente* obligatorio pero *Cliente* tiene uno o muchos *Pedido*.
- Relación con Sucursal: Un *Pedido* tiene una única *Sucursal* y una *Sucursal* puede relacionarse con muchos *Pedido* o ninguno.

Atributo	Tipo de Dato	Tipo de Clave
Pedido_Numero	decimal(18,0)	Primary Key
Pedido_Cliente	bigint	Foreign Key
Pedido_Sucursal	bigint	Foreign Key
Pedido_Fecha	datetime2(6)	
Pedido_Total	decimal(18,2)	
Pedido_Estado	nvarchar(255)	

### (8) Medida

Esta entidad representa a cada medida que es elegida por el cliente: ancho, alto y profundidad del sillón. Las medidas se pueden aplicar a cualquier modelo de sillón y cada una tiene un precio.

- Relación con Sillon: Un *Sillon* tiene una *Medida* y una *Medida* puede relacionarse con muchos *Sillon*.

Atributo	Tipo de Dato	Tipo de Clave
Medida_Codigo	bigint	Primary Key
Medida_Ancho	decimal(18,2)	
Medida_Alto	decimal(18,2)	
Medida_Profundidad	decimal(18,2)	
Medida_Precio	decimal(18,2)	

### (9) Modelo

Esta entidad representa a cada modelo de sillón. Cada uno de los modelos tiene un precio base.

El atributo *Modelo\_Codigo\_Numero* en el modelo que diseñamos se corresponde con el atributo *Sillon\_Modelo* de la tabla maestra.

- Relación con Sillon: Un *Sillon* tiene un *Modelo* y un *Modelo* puede relacionarse con muchos *Sillon*.

Atributo	Tipo de Dato	Tipo de Clave
Modelo_Codigo	bigint	Primary Key
Modelo_Codigo_Numero	nvarchar(255)	
Modelo_Descripcion	nvarchar(255)	



Modelo_Precio_Base	decimal(18,2)	
--------------------	---------------	--

### (10) Sillon

Esta entidad representa a cada sillón, donde el cliente es quien elige las medidas, modelo, y materiales que tendrá el mismo.

Creamos una entidad intermedia *MaterialPorSillon* entre las entidades *Sillon* y *Material* que permite modelar correctamente la relación de muchos a muchos entre estas entidades.

- Relación con DetallePedido: Un *DetallePedido* corresponde a un *Sillon* y un *Sillon* puede tener muchos *DetallePedido*.
- Relación con MaterialPorSillon: Un *Sillon* tiene uno o muchos *MaterialPorSillon*. Un *MaterialPorSillon* tiene un *Sillon*.
- Relación con Modelo: Un *Sillon* tiene un *Modelo* y un *Modelo* puede relacionarse con muchos *Sillon*.
- Relación con Medida: Un *Sillon* tiene una *Medida* y una *Medida* puede relacionarse con muchos *Sillon*.

Atributo	Tipo de Dato	Tipo de Clave
Sillon_Codigo	bigint	Primary Key
Sillon_Medida	bigint	Foreign Key
Sillon_Modelo	bigint	Foreign Key

### (11) DetallePedido

Cada pedido puede tener varios sillones distintos, por cada sillón distinto a fabricar, se registra la información contenida en esta entidad.

- Relación con DetalleFactura: Un *DetallePedido* puede tener 0 o 1 *DetalleFactura*. *DetalleFactura* está relacionado sí o sí a un *DetallePedido* que ya existe.
- Relación con Sillon: Un *DetallePedido* tiene un *Sillon* y un *Sillon* puede relacionarse con muchos *DetallePedido*.
- Relación con Pedido: Un *Pedido* tiene uno o muchos *DetallePedido* ya que se realiza por *Sillon* y un *DetallePedido* está asociado a un único *Pedido* obligatorio.

Atributo	Tipo de Dato	Tipo de Clave
Detalle_Pedido_Numero	bigint	Primary Key
Detalle_Pedido_Sillon	bigint	Foreign Key
Detalle_Pedido_Pedido	decimal(18,0)	Foreign Key
Detalle_Pedido_Cantidad	bigint	
Detalle_Pedido_Precio	decimal(18,2)	

Detalle_Pedido_Subtotal	decimal(18,2)	
-------------------------	---------------	--

## (12) TipoMaterial

Esta entidad se relaciona con *Material*.

El atributo de esta entidad llamado *TipoMaterial\_Tipo* se corresponde con el atributo *Material\_Tipo* de la tabla maestra.

- Relación con Material: Un *TipoMaterial* puede tener muchos *Material*. Un *Material* sólo puede tener un *TipoMaterial*.

Atributo	Tipo de Dato	Tipo de Clave
TipoMaterial_Codigo	bigint	Primary Key
TipoMaterial_Tipo	nvarchar(255)	
TipoMaterial_Nombre	nvarchar(255)	
TipoMaterial_Descripcion	nvarchar(255)	

## (13) Material

Esta entidad representa a cada material que puede elegir el cliente para su sillón. Entre los materiales se encuentran distintos tipos de telas, maderas y rellenos. Cada uno de los materiales tiene un precio y características.

Creamos una entidad intermedia *MaterialPorSillon* entre las entidades *Sillon* y *Material* que permite modelar correctamente la relación de muchos a muchos entre estas entidades.

- Relación con DetalleCompra: Un *DetalleCompra* tiene un *Material* y un *Material* puede tener muchos *DetalleCompra* o ninguno.
- Relación con MaterialPorSillon: Un *Material* tiene uno o muchos *MaterialPorSillon*, y un *MaterialPorSillon* tiene un único *Material*.
- Relación con TipoMaterial: Un *TipoMaterial* puede tener uno o muchos *Material*. Un *Material* sólo puede tener un *TipoMaterial*.
- Relación con Tela: Un *Material* tiene una o ninguna *Tela* y una *Tela* tiene un *Material*.
- Relación con Madera: Un *Material* tiene una o ninguna *Madera* y una *Madera* tiene un *Material*.
- Relación con Relleno: Un *Material* tiene uno o ningún *Relleno* y un *Relleno* tiene un *Material*.

Atributo	Tipo de Dato	Tipo de Clave
Material_Codigo	bigint	Primary Key
Material_Tipo	bigint	Foreign Key

Material_Precio	decimal(38,2)	
-----------------	---------------	--

#### (14) Tela

Esta entidad representa a cada una de las telas con la que se fabrican los sillones y contiene el color y la textura de las mismas.

- Relación con Material: Un *Material* tiene una o ninguna *Tela* y una *Tela* tiene un *Material*.

Atributo	Tipo de Dato	Tipo de Clave
Tela_Codigo	bigint	Primary Key
Tela_Material	bigint	Foreign Key
Tela_Color	nvarchar(255)	
Tela_Textura	nvarchar(255)	

### (15) Madera

Esta entidad representa a cada una de las maderas con la que se fabrican los sillones y contiene el color y la dureza de las mismas.

- Relación con Material: Un *Material* tiene una o ninguna *Madera* y una *Madera* tiene un *Material*.

Atributo	Tipo de Dato	Tipo de Clave
Madera_Codigo	bigint	Primary Key
Madera_Material	bigint	Foreign Key
Madera_Color	nvarchar(255)	
Madera_Dureza	nvarchar(255)	

### (16) Relleno

Esta entidad representa a cada una de los rellenos con los que se fabrican los sillones y contiene información sobre la densidad de los mismos.

- Relación con Material: Un *Material* tiene uno o ningún *Relleno* y un *Relleno* tiene un *Material*.

Atributo	Tipo de Dato	Tipo de Clave
Relleno_Codigo	bigint	Primary Key
Relleno_Material	bigint	Foreign Key
Relleno_Densidad	decimal(38,2)	

### (17) DetalleCompra

Esta entidad representa el detalle de compra de cada material que se compra al proveedor. Es una entidad intermedia entre las entidades Compra y Material para poder representar la relación de muchos a muchos entre ellas.

Atributo	Tipo de Dato	Tipo de Clave
Detalle_Compra_Numero	bigint	Primary Key
Detalle_Compra_Compra	decimal(18,0)	Foreign Key
Detalle_Compra_Material	bigint	Foreign Key
Detalle_Compra_Cantidad	decimal(18,0)	
Detalle_Compra_Precio	decimal(18,2)	
Detalle_Compra_Subtotal	decimal(18,2)	

### (18) PedidoCancelacion

Esta entidad representa la información de cada pedido cancelado.

- Relación con Pedido: un Pedido tiene uno o ningún PedidoCancelacion, y un PedidoCancelacion tiene un único Pedido.

Atributo	Tipo de Dato	Tipo de Clave
Pedido_Cancelacion_Numero	bigint	Primary Key
Pedido_Cancelacion_Pedido	decimal(18,0)	Foreign Key
Pedido_Cancelacion_Fecha	datetime2(6)	
Pedido_Cancelacion_Motivo	varchar(255)	

### (19) Envio

Esta entidad contiene la información de envío relacionada a cada venta.

- Relación con Factura: una Factura tiene una único Envio, al igual que cada Envio tiene una única Factura.

Atributo	Tipo de Dato	Tipo de Clave
Envio_Numero	decimal(18,0)	Primary Key
Envio_Factura	bigint	Foreign Key
Envio_Fecha_Programada	datetime2(6)	
Envio_Fecha_Entrega	datetime2(6)	
Envio_Importe_Traslado	decimal(18,2)	
Envio_Importe_Subida	decimal(18,2)	

Envio_Total	decimal(18,2)	
-------------	---------------	--

## (20) DetalleFactura

Esta entidad representa al detalle de la factura, conteniendo información sobre los ítems de la factura. Relacionándose tanto con la Factura como con el DetallePedido al que están asociados a los ítems facturados.

- Relación con Factura: una Factura puede tener uno o muchos DetalleFactura, un DetalleFactura puede pertenecer sólo una Factura.
- Relación con DetallePedido: un DetallePedido puede tener cero o un DetalleFactura. Un DetalleFactura puede tener un único DetallePedido.

Atributo	Tipo de Dato	Tipo de Clave
Detalle_Factura_Numero	bigint	Primary Key
Detalle_Factura_Factura	bigint	Foreign Key
Detalle_Factura_DetallePedido	bigint	Foreign Key
Detalle_Factura_Cantidad	decimal(18,0)	
Detalle_Factura_Precio	decimal(18,2)	
Detalle_Factura_Subtotal	decimal(18,2)	

## (21) MaterialPorSillon

Creamos una entidad intermedia *MaterialPorSillon* entre las entidades *Sillon* y *Material* que permite modelar correctamente la relación de muchos a muchos entre estas entidades.

- Relación con Sillon: Un Sillon puede tener uno o muchos MaterialPorSillon. Un MaterialPorSillon puede estar únicamente en un Sillon.
- Relación con Material: Un Material puede estar en uno o muchos MaterialPorSillon. Un MaterialPorSillon puede corresponderse únicamente con un Material.

Atributo	Tipo de Dato	Tipo de Clave
MaterialPorSillon_Codigo	bigint	Primary Key
MaterialPorSillon_Material	bigint	Foreign Key
MaterialPorSillon_Sillon	bigint	Foreign Key

Aclaración: Todas las PK que creamos nosotros (es decir, no estaban en la Tabla Maestra) las hicimos de tipo *bigint* para poder definirlos como *Autoincrement*.

## Inconsistencias

- Se encontraron emails, nombres de calles, provincias, localidades y nombres de clientes, proveedores y sucursales que contienen espacios, símbolos especiales, punto y coma y tildes. Estos errores sintácticos tendrán un impacto en la base de datos ya que, por ejemplo, al hacer consultas, en el caso de querer seleccionar a los clientes de “Santiago del Estero”, habrá complicaciones al no estar escrito de esa forma.

Santia; Del Estero

Avenida Crisólo; Laralde N° 7479

- Todos los pedidos y facturas están realizados en fechas futuras, entre 2026 y 2028. Esto podría tener un impacto en la base de datos al no estar cronológicamente ordenado. Cuando lleguen esas fechas, se tendrán nuevamente pedidos y facturas con la misma fecha en la que ya se facturó o generó un pedido. Podrá originar problemas al buscarlas o intentar diferenciar cuáles corresponden a la fecha correcta.

Results			Messages		
	factura_fecha			pedido_fecha	
1	2026-01-11 08:00:00.000000			2026-01-01 08:00:00.000000	
95873	2028-07-24 20:00:00.000000			2028-06-29 20:00:00.000000	

- Hay clientes que tienen el mismo DNI que otra persona, esto puede impactar en la base de datos ya que, al querer buscar un cliente por DNI, se tendrán múltiples resultados, no sabiendo cuál es el correcto.

	cliente_Dni	cliente_nombre	cliente_apellido
1	2718249	LAYA	Lucero
2	2718249	PIERRE	Álvarez
3	16085930	CLODOMIRO	Silva
4	16085930	JEHIEL	Rojas
5	21173566	FLABIAN	Para
6	21173566	PERCIVAL	Riquelme

- La mayoría de los clientes tienen un DNI que no corresponde con su fecha de nacimiento, esto puede impactar en la base de datos ya que, al querer buscar un cliente por DNI, el DNI del cliente puede no corresponder con el DNI que está en la base de datos.

Cliente_FechaNacimiento	cliente_dni
1994-10-17 20:01:59.190000	1316729
1994-10-17 20:01:59.190000	1316729

- Encontramos casos en los que *Detalle\_Compra\_Precio \* Detalle\_Compra\_Cantidad* no coincide completamente con *Detalle\_Compra\_SubTotal*, sino que hay una leve diferencia en el resultado del cálculo en comparación con los datos cargados en la Tabla Maestra. Aquí mostramos el subtotal ya cargado en comparación con el calculado y a continuación la diferencia.

Results		Messages				
	Detalle_compra_precio	detalle_compra_cantidad	detalle_compra_subtotal	Resultado_Calculado	Diferencia	
1	1054.28	2515	2651524.37	2651514.20	10.17	
2	1054.28	2511	2647307.24	2647297.08	10.16	
3	1054.28	2466	2599864.46	2599854.48	9.98	
4	1054.28	2465	2598810.17	2598800.20	9.97	
5	1054.28	2454	2587213.05	2587203.12	9.93	
6	1054.28	2442	2574561.64	2574551.76	9.88	
7	1054.28	2437	2569290.22	2569280.36	9.86	
8	1054.28	2417	2548204.54	2548194.76	9.78	
9	1054.28	2405	2535553.13	2535543.40	9.73	
10	1054.28	2394	2523956.00	2523946.32	9.68	
11	1054.28	2385	2514467.45	2514457.80	9.65	
12	1054.28	2385	2514467.45	2514457.80	9.65	
13	1054.28	2354	2481784.64	2481775.12	9.52	
14	1054.28	2348	2475458.94	2475449.44	9.50	
15	1054.28	2273	2396387.64	2396378.44	9.20	
16	1054.28	2246	2367921.97	2367912.88	9.09	
17	1054.28	2229	2349999.14	2349990.12	9.02	
18	1054.28	2201	2320479.18	2320470.28	8.90	
19	1054.28	2185	2303610.64	2303601.80	8.84	
20	1054.28	2169	2286742.09	2286733.32	8.77	
21	1054.28	2130	2245625.02	2245616.40	8.62	
22	1054.28	2083	2196073.67	2196065.24	8.43	
23	1054.28	2069	2181313.69	2181305.32	8.37	
24	1250.01	2659	3323784.91	3323776.59	8.32	
25	1054.28	2055	2166553.71	2166545.40	8.31	
26	1054.28	2049	2160228.01	2160219.72	8.29	
27	1054.28	2017	2126490.92	2126482.76	8.16	
28	1054.28	2015	2124382.35	2124374.20	8.15	
29	1250.01	2592	3240034.03	3240025.92	8.11	
30	1054.28	1990	2098025.25	2098017.20	8.05	
31	1250.01	2564	3205033.66	3205025.64	8.02	
32	1250.01	2547	3183783.44	3183775.47	7.97	
33	1054.28	1950	2055853.89	2055846.00	7.89	

Esto también ocurre con los demás detalles, pero aquí lo ejemplificamos con Detalle\_Compra. Si bien la diferencia no es muy significativa, lo aclaramos por si acaso.



## Normalización

- Consideramos la opción de normalizar las direcciones a través de la creación de una entidad *Direccion* que tuviera los atributos *nombre\_calle* y *altura\_calle* para separar el nombre de la calle de la altura en las direcciones, pero al no saber si la calle iba a estar siempre separada del número por un “ N° ”, decidimos dejarlo como está en la tabla maestra.

	cliente_Direccion	nombre_calle_normalizado	altura_calle_normalizado
1	Valdenegro N° 9489	Valdenegro	9489
2	Valdenegro N° 9489	Valdenegro	9489
3	Valdenegro N° 9489	Valdenegro	9489
4	Valdenegro N° 9489	Valdenegro	9489
5	Valdenegro N° 9489	Valdenegro	9489
6	Avenida Figueroa Alcorta N° 6197	Avenida Figueroa Alcorta	6197
7	Avenida Figueroa Alcorta N° 6197	Avenida Figueroa Alcorta	6197
8	Avenida Figueroa Alcorta N° 6197	Avenida Figueroa Alcorta	6197
9	Avenida Figueroa Alcorta N° 6197	Avenida Figueroa Alcorta	6197
10	Avenida Brasil N° 7374	Avenida Brasil	7374
11	Avenida Brasil N° 7374	Avenida Brasil	7374

- Redundancias: hay campos que tienen el mismo nombre que el atributo que representa donde se podría almacenar únicamente el valor, liberando así el espacio en memoria que ocuparía el nombre del campo almacenado como varchar en cada fila.

	proveedor_razonSocial
1	Razon Social N°7
2	Razon Social N°7
17	Razon Social N°5
18	Razon Social N°5
19	Razon Social N°4

Lo mismo ocurre en *sillon\_modelo\_codigo*, *sillon\_modelo* y *sillon\_modelo\_descripcion*, que repiten el nombre del atributo en cada registro

sillon_modelo_codigo	sillon_modelo	sillon_modelo_Descripcion
435448	Modelo N°: 435448	Descripcion Modelo N°: 435448
612514	Modelo N°: 612514	Descripcion Modelo N°: 612514
612514	Modelo N°: 612514	Descripcion Modelo N°: 612514
474559	Modelo N°: 474559	Descripcion Modelo N°: 474559

## Diseño

- Decidimos separar la entidad Modelo de la entidad Medida para evitar redundancias en los atributos del sillón. Mientras que Modelo tiene la descripción del diseño del sillón, Medida almacena las dimensiones físicas (alto, ancho y profundidad), para evitar que se repitan los mismos datos en varios modelos. Esto también permite reutilizar medidas, y mantiene independientes las modificaciones sobre diseño estético y estructura física. Además, esto fue conveniente ya que cada medida está previamente cargada en el sistema, con un precio propio, al igual que los modelos.
- Creamos las entidades DetallePedido, DetalleFactura y DetalleCompra para descomponer estructuras repetitivas dentro de Pedido, Factura y Compra respectivamente. Esto permite modelar correctamente las relaciones de uno a muchos y facilita el cálculo de precios y subtotales.
- Creamos la entidad Material para representar de forma genérica los insumos utilizados dentro de los detalles de compras. De esta manera almacenamos los atributos comunes como *material\_tipo*, y *material\_precio* y evitamos repetirlos en múltiples lugares.
- Creamos la entidad TipoMaterial para agrupar los atributos Tipo, Nombre y Descripción, y así evitamos repetirlos, y nos ahorramos posibles errores de tipeo como “Madera” y “Mareda”.
- Consideramos la posibilidad de no incluir los subtotales en las tablas ya que son valores calculables. Los conservamos porque forman parte de los datos de la tabla Maestra.
- Creamos las entidades Relleno, Tela y Madera que son los tipos de materiales disponibles. Consideramos que estos son subtipos de Material, pero como a nivel código no puede implementarse tal relación, hicimos que cada uno de ellos tenga una FK hacia la tabla Material.
- Decidimos por estandarización, primero crear tablas, después las Primary Key y por último las Foreign Key con el fin de que no haya conflicto al establecer las relaciones.
- Hicimos Stored Procedures para:
  - Creación de tablas (uno por cada tabla y luego uno que envuelve a todos esos otros).
  - Migración de datos hacia las tablas creadas (uno por cada tabla y luego uno que envuelve a todos esos otros).
- Creamos checks para realizar verificaciones:
  - TipoMaterial: para verificar que el valor coincida con ‘Tela’, ‘Madera’ o ‘Relleno’.
  - Medida: para verificar que las medidas y el precio sean mayores a 0.
  - Modelo: para verificar que el precio sea mayor a 0.
  - DetallePedido: para verificar que la cantidad y precio sea mayor a 0.
  - DetalleFactura: para verificar que la cantidad y precio sea mayor a 0.
  - DetalleCompra: para verificar que la cantidad y precio sea mayor a 0.
- Creamos índices para acceder a los datos de las tablas de manera más eficiente:

- Ubicación: utilizando Ubicacion\_Provincia, Ubicacion\_Localidad, Ubicacion\_Direccion.
  - Cliente: utilizando Cliente\_Dni, Cliente\_Nombre, Cliente\_Apellido.
  - Proveedor: utilizando Proveedor\_Cuit, Proveedor\_RazonSocial.
  - DetallePedido: utilizando Detalle\_Pedido\_Sillon, Detalle\_Pedido\_Pedido.
  - TipoMaterial: utilizando TipoMaterial\_Tipo, TipoMaterial\_Nombre, TipoMaterial\_Descripcion.
  - Medida: utilizando Medida\_Ancho, Medida\_Alto, Medida\_Profundidad, Medida\_Precio.
  - Material: utilizando Material\_Tipo.
  - Tela: utilizando Tela\_Color, Tela\_Textura.
  - Madera: utilizando Madera\_Color, Madera\_Dureza.
  - Relleno: utilizando Relleno\_Densidad.
  - DetalleCompra: utilizando Detalle\_Compra\_Compra, Detalle\_Compra\_Material.
  - PedidoCancelacion: utilizando Pedido\_Cancelacion\_Pedido, Pedido\_Cancelacion\_Fecha.
  - DetalleFactura: utilizando Detalle\_Factura\_Factura, Detalle\_Factura\_Detalle\_Pedido.
  - MaterialPorSillon: utilizando MaterialPorSillon\_Material, MaterialPorSillon\_Sillon.
- Creamos triggers:
    - ActualizarTotalDelPedido: actualiza el Pedido\_Total cada vez que se hace UPDATE, INSERT o DELETE de un DetallePedido.
    - ActualizarTotalDeCompra: actualiza el Compra\_Total cada vez que se hace UPDATE, INSERT o DELETE de un DetalleCompra.
    - ActualizarTotalDeEnvio: actualiza el Envio\_Total cada vez que se hace UPDATE de un Envio.
    - ActualizarTotalDeFactura: actualiza el Factura\_Total cada vez que se hace UPDATE, INSERT o DELETE de un DetalleFactura.
  - Hicimos un subselect en el JOIN de la migración de DetalleFactura, ya que nos encontramos que al migrar, nos daban más filas de las que tenían que dar, y esto ocurre ya que en la tabla maestra hay más de una fila en la cual coinciden Factura\_Numero, Pedido\_Numero, Detalle\_Factura\_Cantidad, Detalle\_Factura\_Precio y Detalle\_Factura\_Subtotal, por ende no teníamos forma de vincular un Detalle\_Factura con cada Detalle\_Pedido. También habíamos considerado usar ROW\_NUMBER() pero nos pareció mucho más complejo. Entendemos que no es lo más performante pero, si no es de esta manera, no se nos ocurrió otra de vincularlos, ya que según la tabla maestra, deberían haber 61092 detalles factura diferentes, y sin hacer esto que hacemos nosotros nos daba 61228 anteriormente.

