



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

Modelo Relacional.....	3
Modelo de datos.....	3
Entidades y sus relaciones.....	4
(1) Ubicacion.....	4
(2) Cliente.....	4
(3) Sucursal.....	5
(4) Proveedor.....	6
(5) Factura.....	6
(6) Compra.....	7
(7) Pedido.....	7
(8) Medida.....	8
(9) Modelo.....	8
(10) Sillon.....	9
(11) DetallePedido.....	9
(12) TipoMaterial.....	10
(13) Material.....	10
(14) Tela.....	11
(15) Madera.....	11
(16) Relleno.....	11
(17) DetalleCompra.....	12
(18) PedidoCancelacion.....	12
(19) Envio.....	12
(20) DetalleFactura.....	13
(21) MaterialPorSillon.....	13
Inconsistencias.....	14
Normalización.....	16
Diseño.....	17
Modelo de Inteligencia de Negocios.....	19
Modelo de datos: Dimensiones.....	19
BI_Dimension_Tiempo.....	19
BI_Dimension_Cliente.....	19
BI_Dimension_Ubicacion.....	20
BI_Dimension_Sucursal.....	20
BI_Dimension_Modelo_Sillon.....	20
BI_Dimension_Tipo_Material.....	21
BI_Dimension_Estado_Pedido.....	21
BI_Dimension_Turno_Venta.....	21
Modelo de datos: Hechos.....	22
BI_Hechos_Ventas.....	22
BI_Hechos_Compras.....	23
BI_Hechos_Pedidos.....	23
BI_Hechos_Envios.....	24
Vistas.....	24
1. Ganancias.....	24
2. Factura promedio mensual.....	24

3. Rendimiento de modelos.....	24
4. Volumen de pedidos.....	25
5. Conversión de pedidos.....	25
6. Tiempo promedio de fabricación.....	25
7. Promedio de compras.....	25
8. Compras por Tipo de Material.....	26
9. Porcentaje de cumplimiento de envíos.....	26
10. Localidades que pagan mayor costo de envío.....	26
Diseño.....	27

Modelo Relacional

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

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

- 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 definir las 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.

	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:
 - Ubicacion: 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.

Modelo de Inteligencia de Negocios

Generamos un modelo de datos y un script que al ejecutarse crea un modelo de inteligencia de negocios que migra los datos del sistema transaccional con el fin de generar las vistas necesarias para resolver las consultas de negocio.

Modelo de datos: Dimensiones

Creamos las dimensiones que se pedían en el enunciado:

- Tiempo
- Ubicación
- Rango etario clientes
- Turno ventas
- Tipo material (tela, madera y relleno)
- Modelo sillón
- Estado pedido

Y además, creamos otra que consideramos conveniente para el armado de las vistas solicitadas:

- Sucursal (ya que en las vistas 1, 4, 5, 6 y 8 se nos piden consultas agrupadas por sucursal)

BI_Dimension_Tiempo

- Tiempo_Id: para identificar los distintos tiempos de nuestro dominio por los que queremos consultar, utilizamos un IDENTITY(1,1).
- Tiempo_Año, Tiempo_Cuatrimestre y Tiempo_Mes: los obtenemos mediante las fechas del modelo transaccional, y los utilizamos en casi todas las vistas ya que se suele pedir agrupar por mes, año o cuatrimestre en la mayoría de ellas.

Atributo	Tipo de Dato	Tipo de Clave
Tiempo_Id	bigint	Primary Key
Tiempo_Año	smallint	
Tiempo_Cuatrimestre	tinyint	
Tiempo_Mes	tinyint	

BI_Dimension_Cliente

- Cliente_Id identifica a cada cliente ya que consiste en la migración de la PK de la tabla Cliente del modelo transaccional.
- Cliente_Rango_Etario: consiste en la clasificación de las edades de los clientes en:
 - <25
 - 25-35
 - 35-50
 - >50

Atributo	Tipo de Dato	Tipo de Clave
Cliente_Id	bigint	Primary Key
Cliente_Rango_Etario	varchar(10)	

BI_Dimension_Ubicacion

- A través de la Primary Key Ubicacion_Id (IDENTITY(1,1)), esta dimensión se relaciona con BI_Hechos_Ventas y BI_Hechos_Envios.
- Ubicacion_Provincia y Ubicacion_Localidad contienen las provincias y localidades de sucursales, proveedores y clientes obtenidos en el modelo transaccional.

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

BI_Dimension_Sucursal

- Sucursal_Id: identifica a cada sucursal ya que consiste en la migración de la PK de la tabla Sucursal del modelo transaccional.
- Sucursal_Telefono y Sucursal_Mail: los introdujimos para que haya algo más acerca de las sucursales, pero la verdad es que únicamente la utilizamos para agrupar por su id en las distintas vistas que se requiere, por lo cual esto es un poco innecesario.

Atributo	Tipo de Dato	Tipo de Clave
Sucursal_Id	bigint	Primary Key
Sucursal_Telefono	nvarchar(255)	
Sucursal_Mail	nvarchar(255)	

BI_Dimension_Modelo_Sillon

- Modelo_Sillon_Id: identifica a cada modelo de sillón ya que consiste en la migración de la PK de la tabla Modelo del modelo transaccional.
- Modelo_Sillon_Descripcion: una cadena de caracteres que representa a la descripción del modelo de sillón.

Atributo	Tipo de Dato	Tipo de Clave
Modelo_Sillon_Id	bigint	Primary Key

Modelo_Sillon_Descripcion	nvarchar(255)	
---------------------------	---------------	--

BI_Dimension_Tipo_Material

- Tipo_Material_Id: identifica a cada tipo de material ya que consiste en la migración de la PK de la tabla Tipo_Material del modelo transaccional.
- Tipo_Material_Tipo: vendría ser el tipo en sí, es decir “madera”, “tela” o “relleno”, sirve para agrupar en la **vista 8**.

Atributo	Tipo de Dato	Tipo de Clave
Tipo_Material_Id	bigint	Primary Key
Tipo_Material_Tipo	nvarchar(255)	

BI_Dimension_Estado_Pedido

- Estado_Pedido_Id: identifica a cada estado del pedido mediante un IDENTITY(1,1), ya que aquí agrupamos por el momento los estados “ENTREGADO” y “CANCELADO”.
- Estado_Pedido_Estado: una cadena de caracteres que representa si el pedido se encuentra entregado o cancelado, y es útil para agrupar en la **vista 5** por estado de pedido.

Atributo	Tipo de Dato	Tipo de Clave
Estado_Pedido_Id	bigint	Primary Key
Estado_Pedido_Estado	nvarchar(255)	

BI_Dimension_Turno_Venta

- Turno_Venta_Id: para identificar a los distintos turnos de venta de nuestro dominio por los que queremos consultar, utilizamos un IDENTITY(1,1).
- Turno_Venta_Horario: consiste en la clasificación de las franjas horarias de los turnos de venta en:
 - 08:00 - 14:00
 - 14:00 - 20:00

Aclaración: Sabemos que solo se piden identificar aquellos pedidos que se hayan realizado entre esas franjas horarias, pero viendo los datos de la tabla, vimos que hay pedidos que no entran en ninguno de esos intervalos, los cuales fueron filtrados y no tenidos en cuenta a la hora de migrar la dimensión, para lo cual hicimos uso de una función que calcula la franja horaria en base a la fecha del pedido, y si no encuadra con esos intervalos, la marca como ‘Franja Inválida’ para luego filtrar.

Atributo	Tipo de Dato	Tipo de Clave
Turno_Venta_Id	bigint	Primary Key

Turno_Venta_Horario	varchar(20)	
---------------------	-------------	--

Modelo de datos: Hechos

El modelo de datos que creamos está compuesto por hechos y dimensiones, estructurando así la información de la base de datos transaccional en un modelo estrella con el fin de facilitar el análisis de los datos. Todas las relaciones entre hechos y dimensiones son del mismo tipo, es decir, una dimensión puede estar en muchos o ningún hecho, y un hecho se relaciona con una dimensión obligatoria.

BI_Hechos_Ventas

- **Primary Key:** identificamos cada venta a través de la PK compuesta por Tiempo_Id, Sucursal_Id, Ubicacion_Id, Cliente_Id y Modelo_Sillon_Id.
- **Foreign Key:** Tiempo_Id, Sucursal_Id, Ubicacion_Id, Cliente_Id y Modelo_Sillon_Id son claves foráneas que relacionan esta tabla con las de dimensión.
- **Ventas_Monto:** representa el monto total vendido.
- **Ventas_Cantidad:** representa la cantidad que se vendió en unidades de ese modelo.
- **Ventas_Tiempo_Fabricacion:** representa el tiempo promedio que tarda en fabricarse un sillón desde que se realiza el pedido hasta que se emite la factura.

Esta tabla la utilizamos para representar:

- **Vista 1:** conocer las ganancias mensuales de cada sucursal.
- **Vista 2:** calcular el valor promedio de las facturas por provincia por cuatrimestre de cada año.
- **Vista 3:** conocer los 3 modelos de sillón con mayores ventas por cada cuatrimestre de cada año, según localidad de la sucursal y rango etario.
- **Vista 6:** conocer el tiempo promedio de fabricación entre que se realiza un pedido hasta que se factura, por sucursal, por cuatrimestre.

Atributo	Tipo de Dato	Tipo de Clave
Tiempo_Id	bigint	Primary Key, Foreign Key
Sucursal_Id	bigint	Primary Key, Foreign Key
Ubicacion_Id	bigint	Primary Key, Foreign Key
Cliente_Id	bigint	Primary Key, Foreign Key
Modelo_Sillon_Id	bigint	Primary Key, Foreign Key
Ventas_Monto	decimal(28,2)	
Ventas_Cantidad	decimal(18,0)	
Ventas_Tiempo_Fabricacion	tinyint	

BI_Hechos_Compras

- Primary Key: identificamos cada compra a través de la PK compuesta por Tiempo_Id, Tipo_Material_Id y Sucursal_Id.
- Foreign Key: Tiempo_Id, Tipo_Material_Id y Sucursal_Id son claves foráneas que relacionan esta tabla con las de dimensión.
- Compras_Monto: representa el monto invertido en compras (egresos).

Esta tabla la utilizamos para representar:

- **Vista 1**: conocer las ganancias mensuales de cada sucursal
- **Vista 7**: calcular el importe promedio de compras por mes.
- **Vista 8**: calcular el importe total gastado por tipo de material, sucursal y cuatrimestre.

Atributo	Tipo de Dato	Tipo de Clave
Tiempo_Id	bigint	Primary Key, Foreign Key
Tipo_Material_Id	bigint	Primary Key, Foreign Key
Sucursal_Id	bigint	Primary Key, Foreign Key
Compras_Monto	decimal(28,2)	

BI_Hechos_Pedidos

- Primary Key: identificamos cada pedido a través de la PK compuesta por Tiempo_Id, Sucursal_Id, Estado_Pedido_Id y Turno_Venta_Id.
- Foreign Key: Tiempo_Id, Sucursal_Id, Estado_Pedido_Id y Turno_Venta_Id son claves foráneas que relacionan esta tabla con las de dimensión.
- Pedidos_Cantidad: representa la cantidad de pedidos recibidos.

Utilizamos esta tabla para representar:

- **Vista 4**: donde se requiere obtener la cantidad de pedidos registrados por turno, por sucursal, según el mes de cada año.
- **Vista 5**: donde se requiere obtener porcentaje de pedidos, según estado, por cuatrimestre y sucursal.

Atributo	Tipo de Dato	Tipo de Clave
Tiempo_Id	bigint	Primary Key, Foreign Key
Sucursal_Id	bigint	Primary Key, Foreign Key
Estado_Pedido_Id	bigint	Primary Key, Foreign Key
Turno_Venta_Id	bigint	Primary Key, Foreign Key
Pedidos_Cantidad	int	

BI_Hechos_Envios

- Primary Key: identificamos cada envío a través de la PK compuesta por Tiempo_Id y Ubicacion_Id.
- Foreign Key: Tiempo_Id y Ubicacion_Id son claves foráneas que relacionan esta tabla con las de dimensión.
- Envios_Cantidad_Total y Envios_Cantidad_En_Fecha: los utilizamos para calcular la **vista 9**, donde se requiere conocer el porcentaje de cumplimiento en fecha de envíos.
- Envios_Monto: representa el monto que fue enviado.

Atributo	Tipo de Dato	Tipo de Clave
Tiempo_Id	bigint	Primary Key, Foreign Key
Ubicacion_Id	bigint	Primary Key, Foreign Key
Envios_Cantidad_Total	int	
Envios_Cantidad_En_Fecha	int	
Envios_Monto	decimal(28,2)	

Vistas

1. Ganancias

Para conocer las ganancias mensuales de cada sucursal, calculamos la diferencia entre el total de ingresos (Ventas_Monto de BI_Hechos_Ventas) y el total de egresos (Compras_Monto de BI_Hechos_Compras). Para esto fue necesario que ambas tablas de Hechos tuvieran relación con

- BI_Dimension_Sucursal: para identificar la sucursal.
- BI_Dimension_Tiempo: para agrupar por mes.

2. Factura promedio mensual

Se obtiene el promedio mensual de facturación por provincia, cuatrimestre y año. Se utiliza la tabla BI_Hechos_Ventas y se divide el AVG(Ventas_Monto) entre 4 (cantidad de meses por cuatrimestre).

3. Rendimiento de modelos

Se obtiene el Top 3 de modelos de sillón más vendidos (por cantidad) agrupados por:

- Cuatrimestre
- Año
- Localidad de la sucursal
- Rango etario del cliente

Se usa ROW_NUMBER() con PARTITION BY para numerar los modelos dentro de cada grupo. Luego se filtran los primeros 3.

A pesar de que hicimos un subselect en el from, no se nos ocurría otra manera de hacer un top 3 para cada cuatrimestre, año, localidad y rango etario, la otra forma de hacerlo con top 1 anidados nos pareció mucho más engorrosa.

4. Volumen de pedidos

Muestra la cantidad total de pedidos (SUM(Pedidos_Cantidad)) registrados por turno, sucursal, mes y año.

Se agrupa por:

- Franja horaria del turno (Turno_Venta_Horario)
- Sucursal
- Mes
- Año

5. Conversión de pedidos

Se calcula el porcentaje de pedidos por estado (Estado_Pedido_Estado) en cada combinación de cuatrimestre y sucursal.

Se agrupa por:

- Estado del pedido
- Cuatrimestre
- Sucursal

$\text{SUM(Pedidos_Cantidad)} * 100.0 / \text{TotalPedidosEnEseGrupo}$.

Se usa una subconsulta para obtener el total de pedidos por cuatrimestre y sucursal.

6. Tiempo promedio de fabricación

Muestra el tiempo promedio entre la fecha del pedido y la fecha de facturación, agrupado por sucursal y cuatrimestre.

Promedio del campo Ventas_Tiempo_Fabricacion, calculado previamente al migrar ventas.

Tablas utilizadas:

- BI_Hechos_Ventas
- BI_Dimension_Tiempo
- BI_Dimension_Sucursal

Agrupado por:

- Sucursal
- Cuatrimestre

7. Promedio de compras

Promedio del monto de compras realizadas por mes. Se calcula con el $\text{AVG(Compras_Monto)}$

Tablas utilizadas:

- BI_Hechos_Compras
- BI_Dimension_Tiempo

Agrupado por:

- Mes

8. Compras por Tipo de Material

Importe total de compras por tipo de material, sucursal y cuatrimestre. Se calcula con el SUM(Compras_Monto).

Se agrupa por:

Tablas utilizadas:

- BI_Hechos_Compras
- BI_Dimension_Tipo_Material
- BI_Dimension_Tiempo
- BI_Dimension_Sucursal

Agrupado por:

- Tipo de material
- Sucursal
- Cuatrimestre

9. Porcentaje de cumplimiento de envíos

Porcentaje de envíos que fueron entregados en fecha en cada mes. Se calcula como $\text{Envios_Cantidad_En_Fecha} * 100.0 / \text{Envios_Cantidad_Total}$

Tablas utilizadas:

- BI_Hechos_Envios
- BI_Dimension_Tiempo

Agrupado por:

- Tiempo_Mes

10. Localidades que pagan mayor costo de envío

Muestra el top 3 localidades con mayor costo promedio de envío. Se calcula como promedio de Envios_Monto por localidad, ordenado descendientemente por dicho promedio.

Tablas utilizadas:

- BI_Hechos_Envios
- BI_Dimension_Ubicacion

Agrupado por:

- Localidad

Diseño

- **Uso de stored procedures:**
Se optó por encapsular tanto la creación de tablas como la migración de datos en procedimientos almacenados (procedures), al igual que en la entrega anterior, con el objetivo de mantener una estructura modular, reutilizable y fácilmente ejecutable del proceso completo.
Esta decisión, si bien no era requerida explícitamente en esta etapa del trabajo, se fundamenta en los siguientes beneficios:
 - ❖ **Automatización del despliegue:** Permite ejecutar el proceso completo (creación + carga de datos) con una única instrucción EXEC, lo cual es especialmente útil en entornos de prueba o corrección.
 - ❖ **Reutilización del código:** Al estar empaquetado en procedimientos, el código puede reutilizarse en distintos contextos sin copiar y pegar bloques enteros.
 - ❖ **Organización del proyecto:** Mejora la claridad del código al separar las responsabilidades y mantener cada paso del proceso bien definido.
 - ❖ **Consistencia con entregas previas:** Se mantiene el mismo criterio técnico utilizado en la etapa anterior, favoreciendo la homogeneidad y coherencia del proyecto completo.

En caso de ser necesario realizar ajustes o volver a cargar los datos, el uso de procedimientos facilita la repetición controlada de los procesos sin riesgos de omitir pasos o introducir errores manuales.

- Creamos una función para que, dada una fecha de nacimiento, calcular el rango etario del cliente, utilizada en la migración de la dimensión Cliente.
- Creamos una función para que, dada una fecha, calcule la franja horaria a la que pertenece, y si no entra en el rango que consideramos(08:00 - 14:00 o bien 14:00 - 20:00), se la marca como franja inválida y luego se la filtra en la migración de la dimensión Turno_Venta.
- 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.
- Creamos índices para acceder a los datos de las tablas de manera más eficiente:
 - **Dimension_Tiempo:** utilizando Tiempo_Año, Tiempo_Cuatrimestre y Tiempo_Mes.
 - **Dimension_Cliente:** utilizando Cliente_Rango_Etario.
 - **Dimension_Ubicacion:** utilizando Ubicacion_Provincia y Ubicacion_Localidad.
 - **Dimension_Turno_Venta:** utilizando Turno_Venta_Horario.
 - **Hechos_Ventas:** utilizando Ventas_Monto, Ventas_Cantidad, Ventas_Tiempo_Fabricacion.
 - **Hechos_Compras:** utilizando Compras_Monto.
 - **Hechos_Pedidos:** utilizando Pedidos_Cantidad.
 - **Hechos_Envios:** utilizando Envios_Cantidad_Total, Envios_Cantidad_En_Fecha, Envios_Monto.
- Para las sumatorias de montos en los hechos de ventas, compras y envíos utilizamos como tipo de dato decimal(28,2), ya que en el modelo transaccional los montos unitarios

son decimal(18,2) pero creemos que al estar sumando muchos de ellos necesitaríamos mayor apreciación para poder almacenar números muy grandes.



