

ESTRATEGIA - QUERYOSOS

Curso: K3012

Número de grupo: 8

| Alumno | Legajo |
|---------------------------------|-----------|
| Gabriel Borré | 203.425-6 |
| Maria Florencia Sanchez | 175.781-7 |
| María Sol Garaventa | 209.622-5 |
| Sebastián Santiago Ayala Osorio | 213.487-1 |

| | |
|--|-----------|
| DER..... | 2 |
| Inconsistencia hallada en la tabla maestra..... | 2 |
| 1)Gestión de Pedidos..... | 3 |
| Entidad Pedido y Estado..... | 3 |
| Entidad Pedido y Cliente..... | 4 |
| Entidad Pedido y Motivo_Cancelacion_Pedido..... | 4 |
| Entidad Pedido y Sucursal..... | 5 |
| Entidad Direccion, Localidad y Provincia..... | 6 |
| Entidad Pedido y entidad ItemDetallePedido..... | 7 |
| Entidad Sillon..... | 8 |
| Entidad Material y Material_Sillon..... | 9 |
| Entidad Material y Entidades Tela, Madera y Relleno..... | 10 |
| 2)Facturación..... | 10 |
| Entidad Factura y Entidad ItemDetalleFactura..... | 10 |
| 3) Envío..... | 11 |
| 4)Compra..... | 12 |
| Migración de datos..... | 13 |
| Modelo de Inteligencia de Negocios (BI)..... | 18 |
| Dimensión BITiempo..... | 18 |
| Dimensión BITurno..... | 18 |
| Dimensión BISucursal..... | 19 |
| Dimensión BI_EstadoPedido..... | 19 |
| Dimensión BIModelo..... | 20 |
| Dimensión BIMaterial..... | 20 |
| Tabla de Hecho BI_Pedido..... | 21 |
| Tabla de Hecho BI_Compra..... | 22 |
| Tabla de Hecho BI_Envio..... | 23 |
| Tabla de Hecho BI_Facturacion..... | 24 |
| Aclaraciones:..... | 25 |

DER

Inconsistencia hallada en la tabla maestra

En la tabla maestra existen detalles de pedidos que tienen varios códigos de sillones distintos con el mismo modelo, materiales y dimensiones que, a su vez, poseen el mismo precio. En otras palabras, existen detalles de pedidos que aparecen repetidos, únicamente variando en ellos el sillón de código que tienen asociado

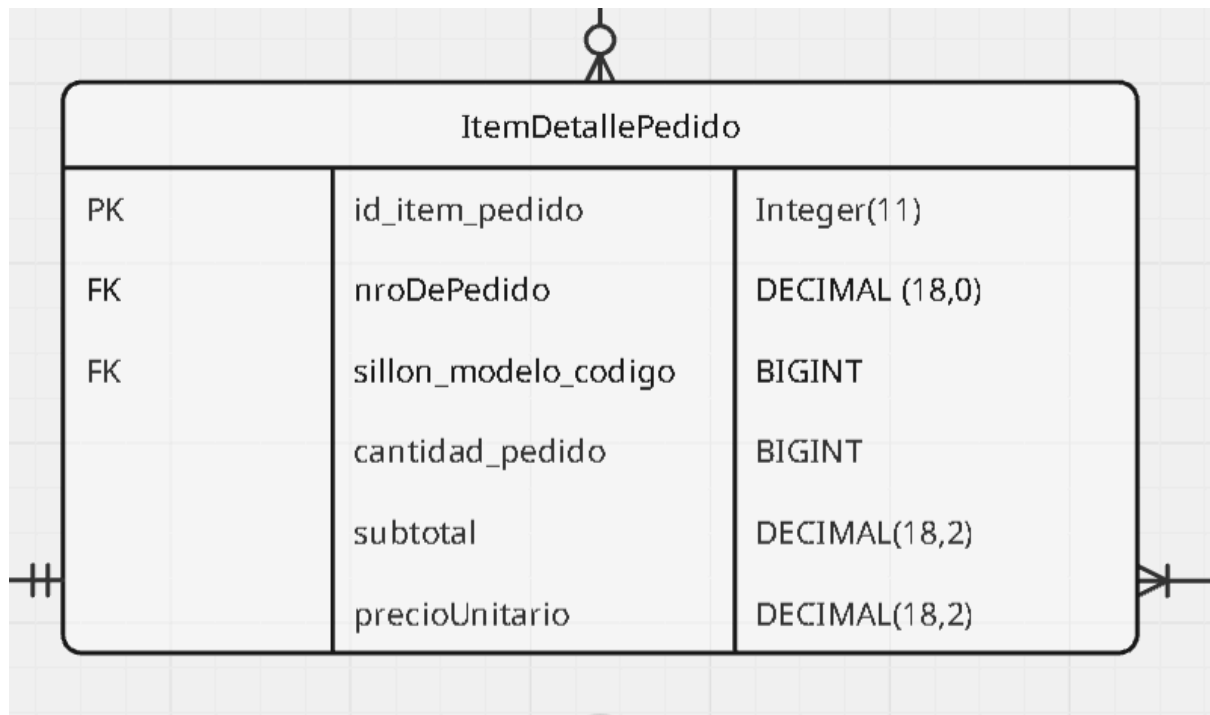
Para algunos casos, no se facturan cada uno de los sillones con distinto código, sino que se facturan menos de ellos, siempre coincidiendo la cantidad del detalle factura y el subtotal con la cantidad de uno de los detalles compra.

Por eso mismo, para salvar la inconsistencia, suponemos que si un ítemdetalle pedido tiene asociado el mismo modelo de sillón, la misma cantidad de unidades, el mismo subtotal y, además, pertenece al mismo pedido, entonces ese va a ser un registro de la tabla ItemDetallePedido, con su propio id.

De esta manera, independientemente del código del sillón, todos los pedidos que tienen facturas asociadas, van a poder estar referenciados desde los registros de la tabla ItemDetallefactura.

Ilustramos lo expuesto anteriormente con el ejemplo del Pedido número 56375800, el cual tiene dos detalles de pedido que tienen las mismas características (cantidad igual a 6, precio igual a 31365.54, mismo modelo, etc), pero difieren en el código de sillón, y por eso, solamente es facturado uno de ellos (penúltimo registro de la tabla)

| 100 % | | | | | | | | | | | |
|------------------|----------------------------|---------------|--------------|-------------------------|-----------------------|-------------------------|----------------------|-------------------|-------------------------------|----------------------|--------------|
| Results Messages | | | | | | | | | | | |
| mero | Pedido_Fecha | Pedido_Estado | Pedido_Total | Detalle_Pedido_Cantidad | Detalle_Pedido_Precio | Detalle_Pedido_SubTotal | Sillon_Modelo_Codigo | Sillon_Modelo | Sillon_Modelo_Descripcion | Sillon_Modelo_Precio | Sillon_Codig |
| 1 | 2027-02-11 14:00:00.000000 | ENTREGADO | 354659.51 | 5 | 31365.64 | 156828.18 | 406739 | Modelo N°: 406739 | Description Modelo N°: 406739 | 16554.20 | 551320 |
| 2 | 2027-02-11 14:00:00.000000 | ENTREGADO | 354659.51 | 5 | 31365.64 | 156828.18 | 406739 | Modelo N°: 406739 | Description Modelo N°: 406739 | 16554.20 | 551320 |
| 3 | 2027-02-11 14:00:00.000000 | ENTREGADO | 354659.51 | 5 | 31365.64 | 156828.18 | 406739 | Modelo N°: 406739 | Description Modelo N°: 406739 | 16554.20 | 551320 |
| 4 | 2027-02-11 14:00:00.000000 | ENTREGADO | 354659.51 | 5 | 31365.64 | 156828.18 | 406739 | Modelo N°: 406739 | Description Modelo N°: 406739 | 16554.20 | 551321 |
| 5 | 2027-02-11 14:00:00.000000 | ENTREGADO | 354659.51 | 5 | 31365.64 | 156828.18 | 406739 | Modelo N°: 406739 | Description Modelo N°: 406739 | 16554.20 | 551321 |
| 6 | 2027-02-11 14:00:00.000000 | ENTREGADO | 354659.51 | 5 | 31365.64 | 156828.18 | 406739 | Modelo N°: 406739 | Description Modelo N°: 406739 | 16554.20 | 551321 |
| 7 | 2027-02-11 14:00:00.000000 | ENTREGADO | 354659.51 | 1 | 41003.15 | 41003.15 | 700792 | Modelo N°: 700792 | Description Modelo N°: 700792 | 18847.86 | 551322 |
| 8 | 2027-02-11 14:00:00.000000 | ENTREGADO | 354659.51 | 1 | 41003.15 | 41003.15 | 700792 | Modelo N°: 700792 | Description Modelo N°: 700792 | 18847.86 | 551322 |
| 9 | 2027-02-11 14:00:00.000000 | ENTREGADO | 354659.51 | 1 | 41003.15 | 41003.15 | 700792 | Modelo N°: 700792 | Description Modelo N°: 700792 | 18847.86 | 551322 |
| 10 | 2027-02-11 14:00:00.000000 | ENTREGADO | 354659.51 | 5 | 31365.64 | 156828.18 | NULL | NULL | NULL | NULL | NULL |
| 11 | 2027-02-11 14:00:00.000000 | ENTREGADO | 354659.51 | 1 | 41003.15 | 41003.15 | NULL | NULL | NULL | NULL | NULL |



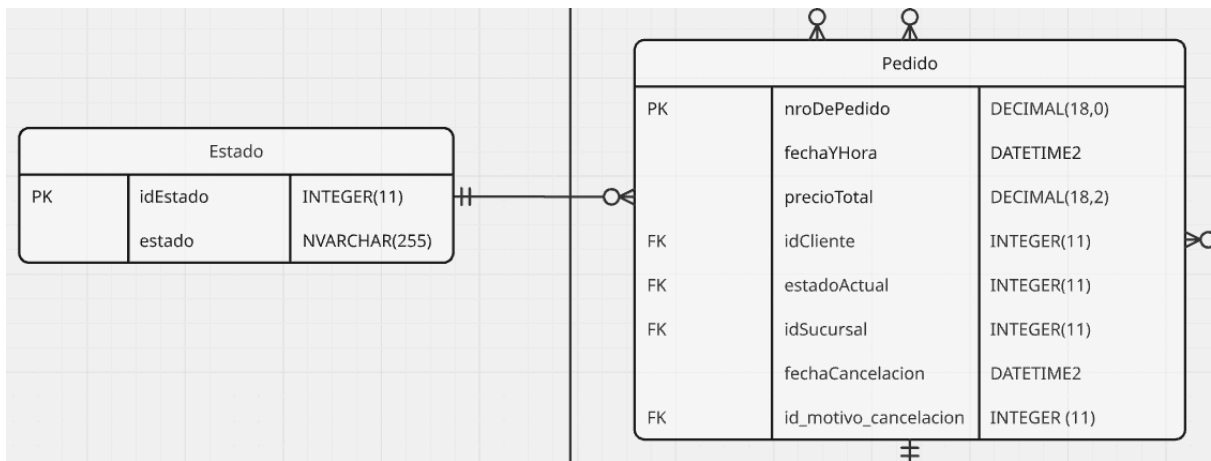
Con esta tabla ItemDetallePedido, salvamos la inconsistencia y garantizamos la integridad de los datos.

1)Gestión de Pedidos

Entidad Pedido y Estado

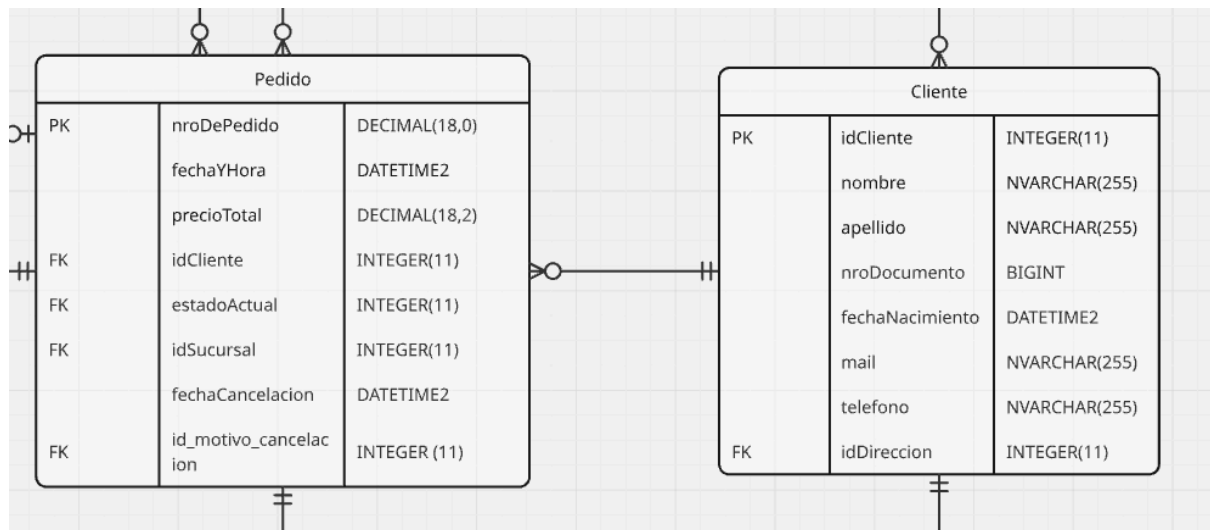
Pedido es la entidad central de la gestión de pagos y fue modelada de esta manera ya que debe guardar en sus campos las referencias (FK) asociadas a qué cliente fue el que lo realizó y qué estado actual tiene. Por cuestiones de consistencia, se modeló la entidad Estado, la cual tiene como campo al tipo de estado en el que se encuentra el pedido (Pendiente, Entregado o Cancelado).

Asimismo, dentro de la entidad Pedido se agregó un campo “precio” que representa el precio total del pedido. Si bien este puede ser calculado a partir de campos ubicados en otras tablas, es necesario incluirlo en la mencionada entidad de manera desnormalizada, debido a una cuestión de performance. Además, si los precios de las otras tablas fueran modificados, nos interesa conservar el monto correcto al momento del pedido.



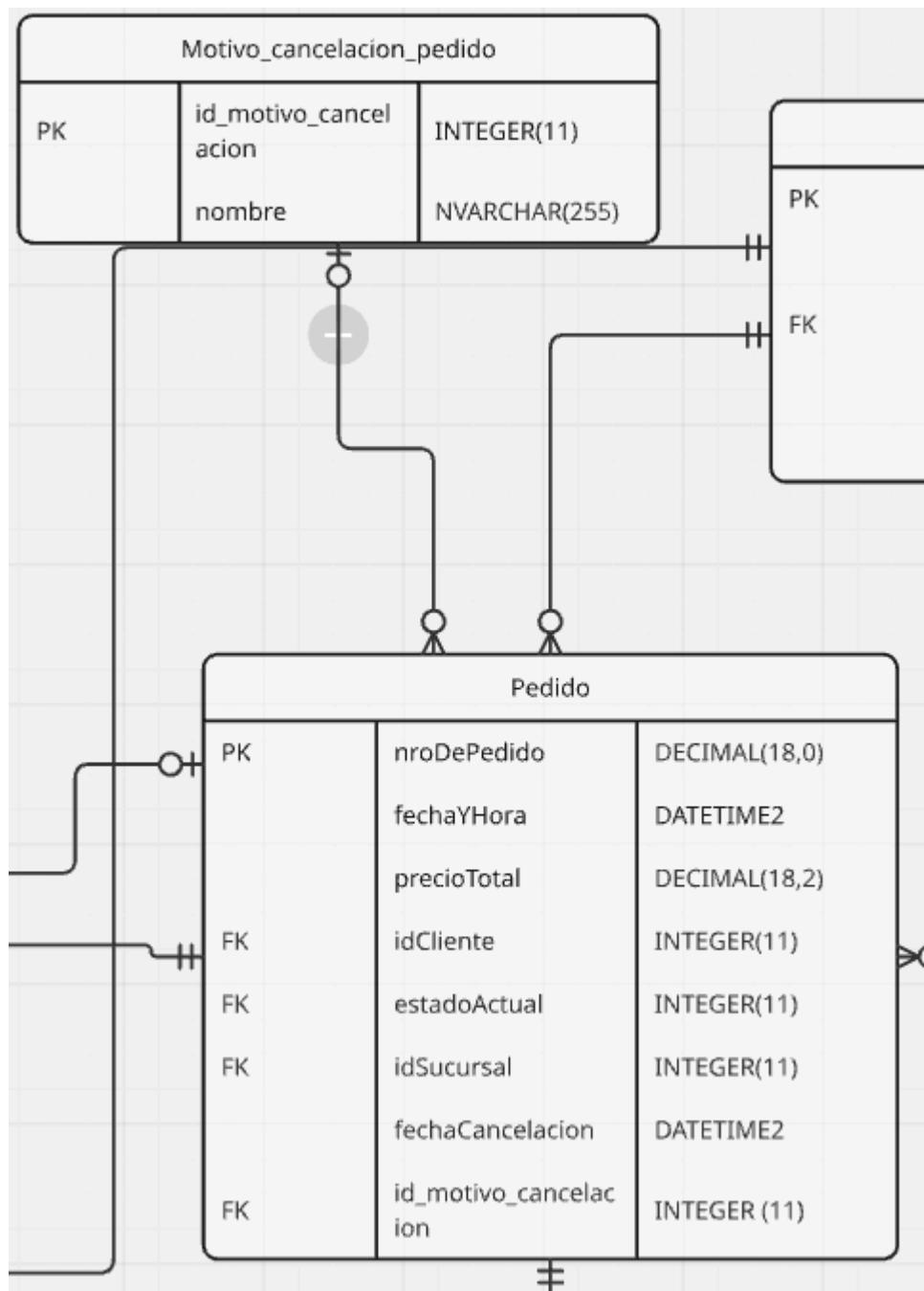
Entidad Pedido y Cliente

Como se mencionó previamente, es necesario guardar en la tabla Pedido una FK a la tabla Cliente, con el objetivo de saber qué cliente realizó determinado periodo.



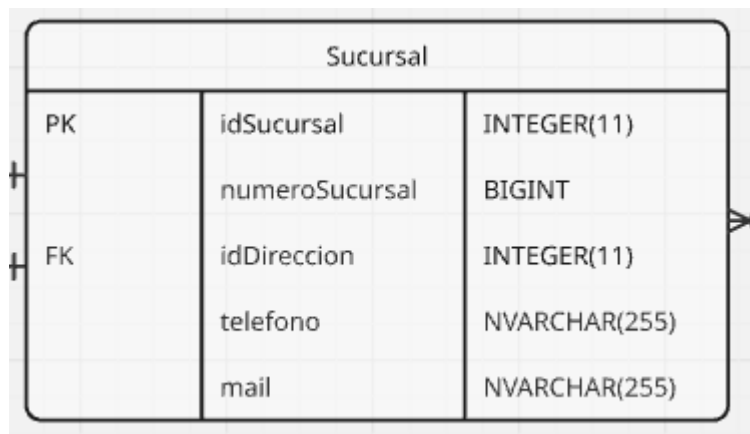
Entidad Pedido y Motivo_Cancelacion_Pedido

Para evitar redundancia de datos, se decidió modelar el motivo de cancelación del pedido como una entidad. De esta manera, el Pedido tiene una FK al motivo, para saber cuál fue el motivo que ocasionó la cancelación del pedido. En caso de que no exista motivo de cancelación, este campo estará en NULL.



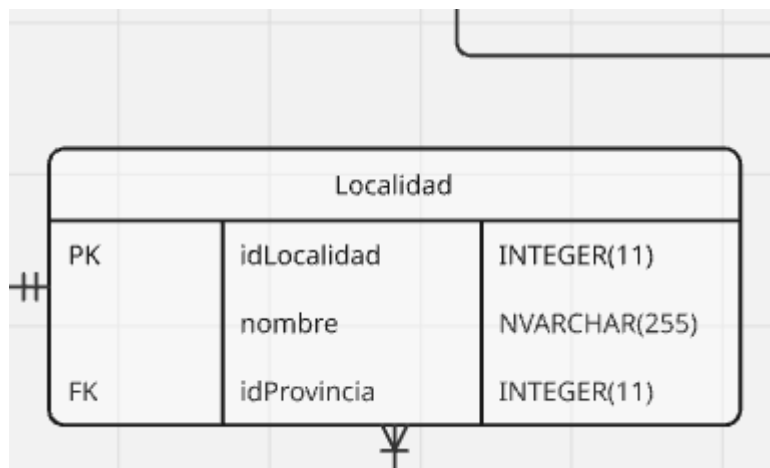
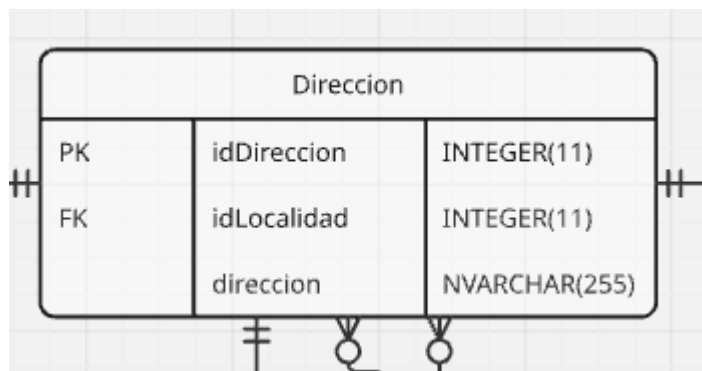
Entidad Pedido y Sucursal

El sistema necesita registrar la sucursal en la que se realizó el pedido, por lo que es necesario crear una tabla "Sucursal" que sea referenciada en la tabla Pedido a través de una FK.



Entidad Direccion, Localidad y Provincia

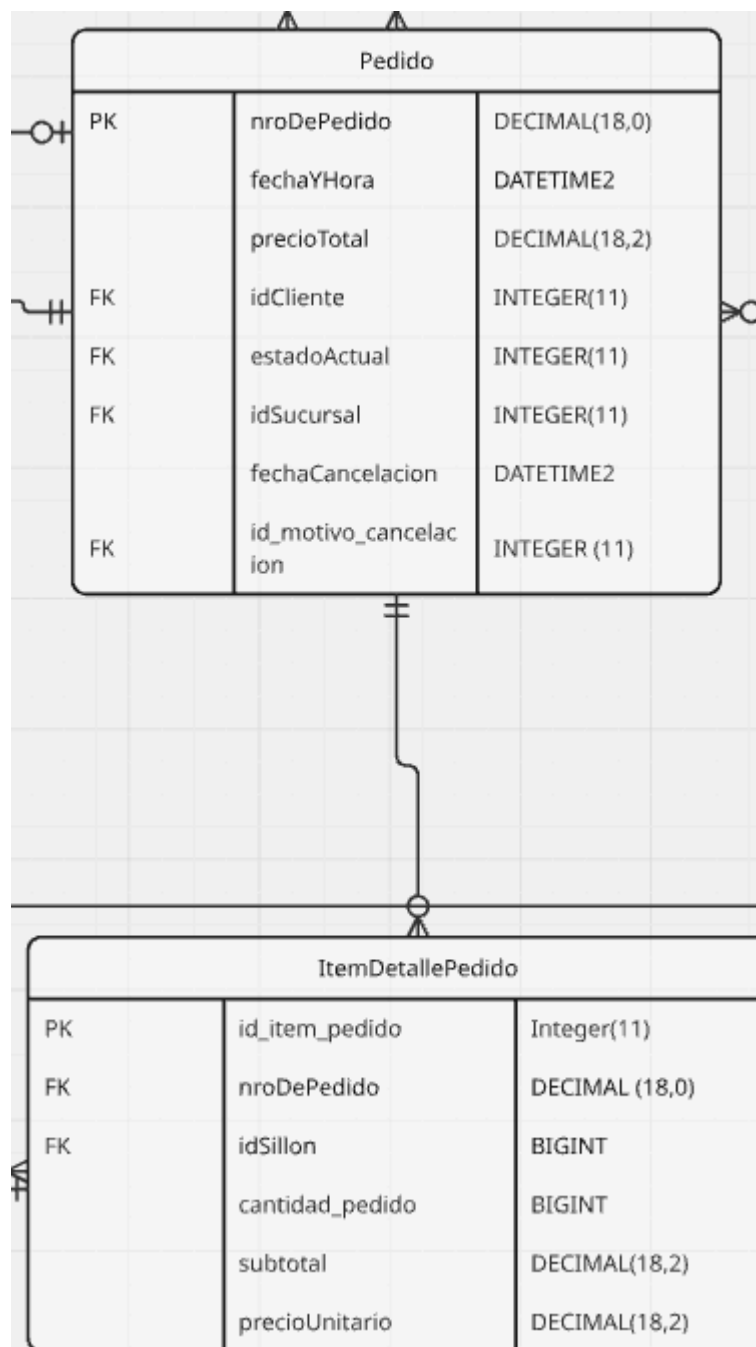
A su vez, por una cuestión de consistencia de datos y para normalizar el modelo, se decidió modelar a la dirección como una entidad. La tabla "Direccion" tendrá campo VARCHAR(255) que contenga la dirección en texto y, además, contará con una FK a la entidad Localidad, que a su vez tendrá una FK a la entidad Provincia. Estas dos tablas se modelaron de esta manera por la misma razón que la tabla Dirección: la consistencia de datos.





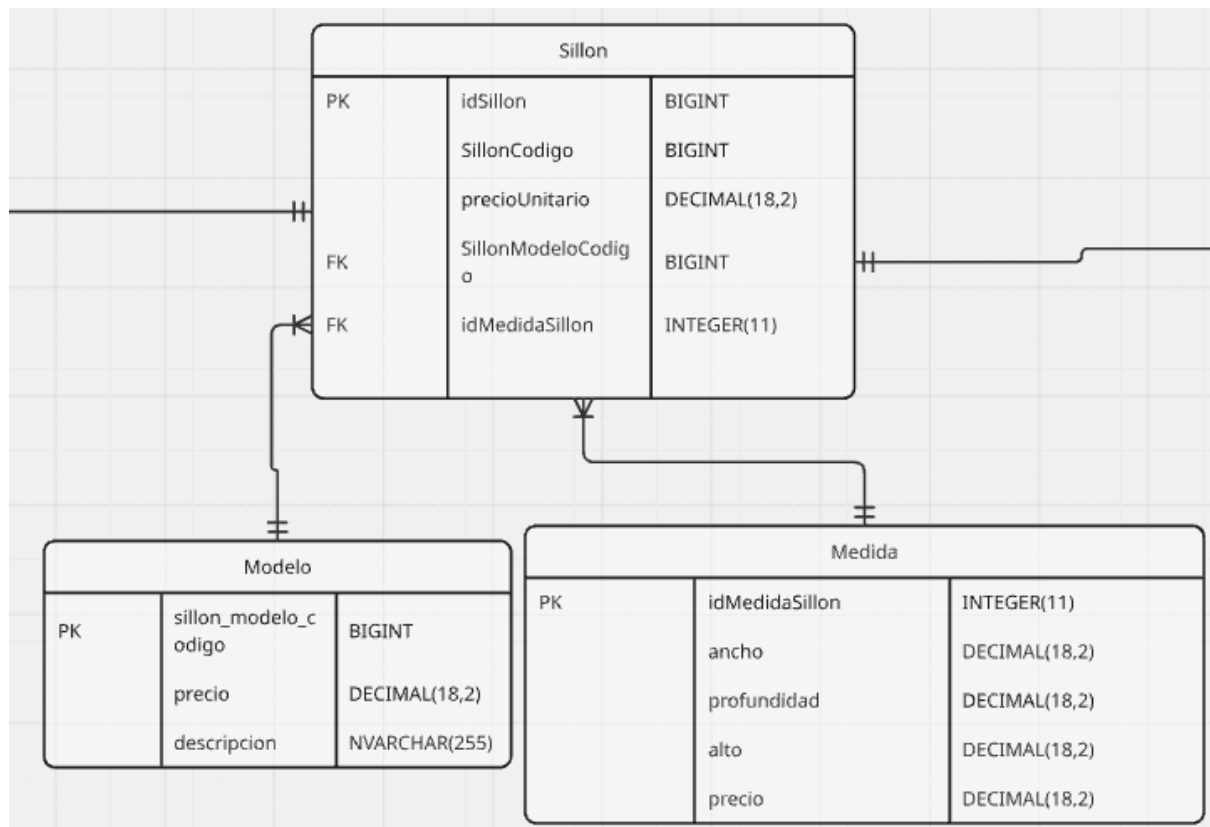
Entidad Pedido y entidad ItemDetallePedido

Un pedido se relaciona con la entidad ItemDetallePedido, tabla la cual tiene los renglones asociados a un pedido, los cuales especifican qué sillón se pidió, qué cantidad de estos se solicitó, cuál fue su precio unitario, entre otras. Un pedido puede tener muchos renglones (uno por cada sillón distinto) y, a su vez, un renglón (una fila de la tabla ItemDetallePedido) pertenece a un único pedido y, por ese motivo, tiene una FK a esta tabla.



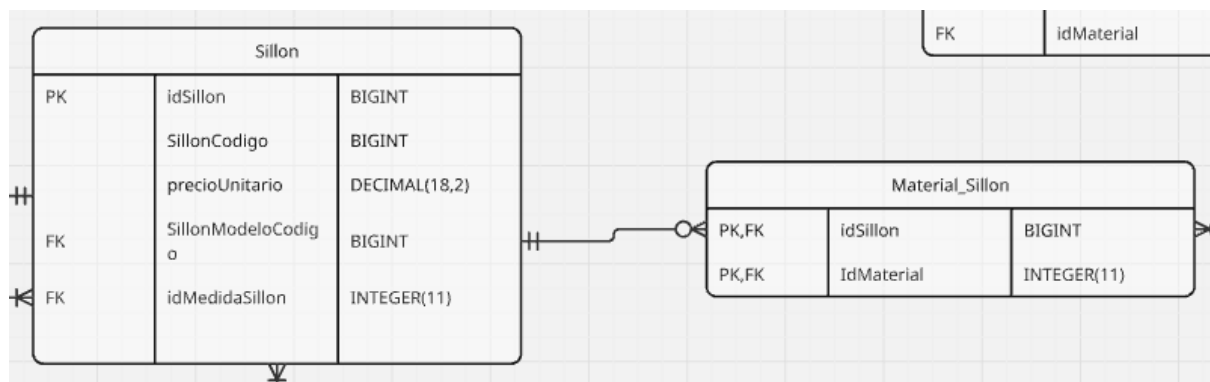
Entidad Sillon

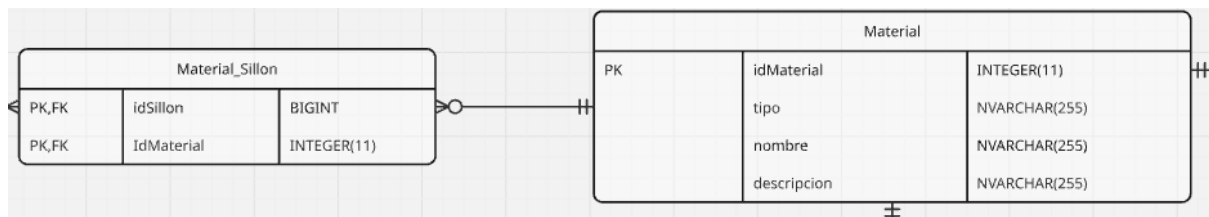
La entidad Sillon representa un sillón en particular que el cliente solicita, el cual tiene un determinado modelo, ciertas dimensiones y está constituido por determinados materiales. Por consistencia de datos y para respetar la tercera forma normal, cada una de estas tres características fue representada con una tabla. De esta manera, el sillón tiene una FK a cada una de ellas, para mantener una referencia de qué modelo, qué dimensiones y qué materiales tiene.



Entidad Material y Material_Sillon

Los materiales están preconfigurados en el Sistema y debemos guardar un registro de ellos, por lo que es necesario una entidad Material. El campo “tipo” hace alusión a si el material es de madera, lana o relleno de sillón. Asimismo, el campo “nombre” hace alusión al nombre del material en cuestión (por ejemplo, material “tipo” tela, “nombre” poliester). Asimismo, debido a que un sillón puede estar compuesto por muchos materiales distintos y un material puede constituir muchos sillones distintos, es necesario crear una tabla intermedia Material_Sillon, cuya PK se halle compuesta por la FK de Material y la FK de Sillón



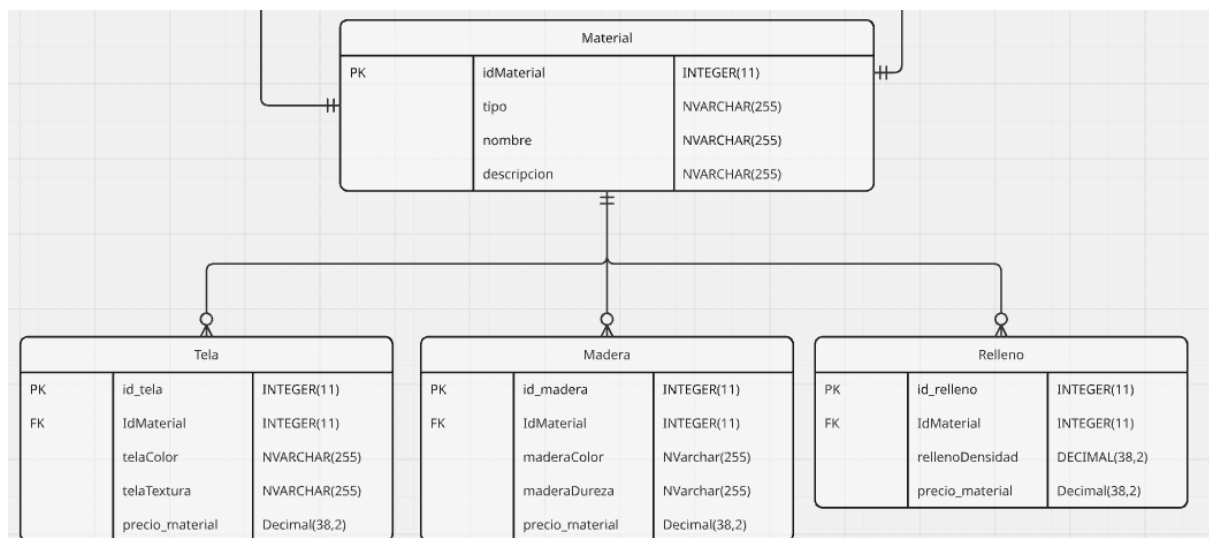


Entidad Material y Entidades Tela, Madera y Relleno

Sabemos que materiales de características particulares distintas pueden tener las mismas características generales. Por ejemplo, dos telas de poliéster pueden tener distintos colores, pero ambas no dejan de ser telas de poliéster.

Es por esto que por cada tipo material de características generales iguales, se crearon tablas que hacen referencia a las características particulares de ese material. Como dos materiales particulares distintos pueden tener las mismas características generales, es necesario referenciar a esas características generales mediante una FK.

A su vez, decidimos hacer el precio relativo al material en particular y no a las características generales, dado que dos telas del mismo tipo pero de distinto color, pueden tener precios asociados distintos.



2)Facturación

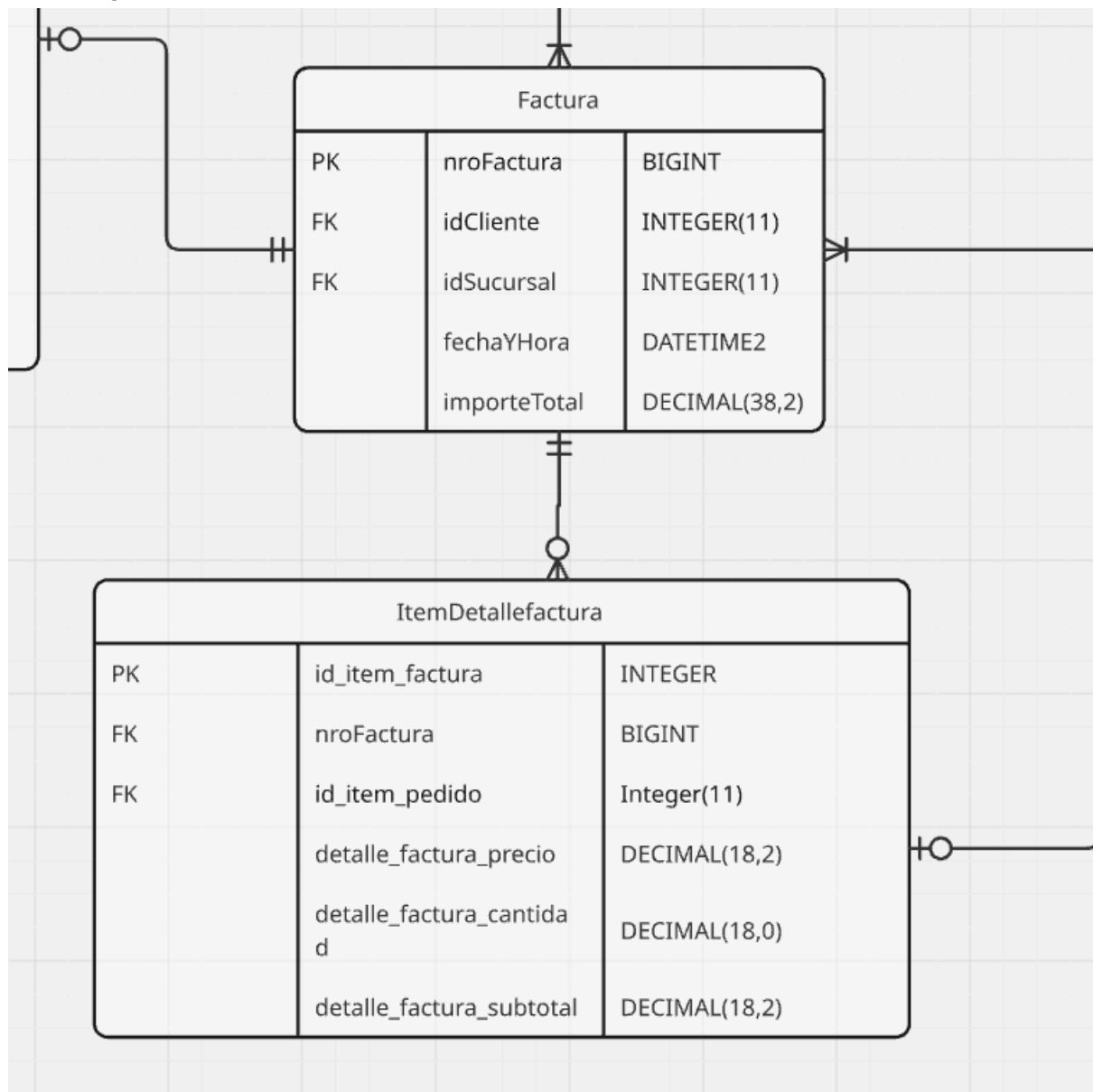
Entidad Factura y Entidad ItemDetalleFactura

Se modeló la entidad Factura con todos sus campos correspondientes tal como indica el enunciado. A su vez, una factura puede tener muchos renglones (uno por cada producto abonado), por lo que es necesario crear una tabla ItemDetalleFactura con la cantidad de

unidades compradas de un determinado producto, el precio unitario y el subtotal de ese renglón (precio unitario por cantidad de unidades).

Asimismo, se debe hacer referencia mediante una FK al itemDetallePedido para saber a qué renglón del pedido corresponde ese renglón de la factura.

Por una cuestión de performance, decidimos desnormalizar el campo importeTotal en la tabla Factura, el cálculo será la suma del precio unitario por la cantidad de unidades de cada renglón que conforme a la factura.

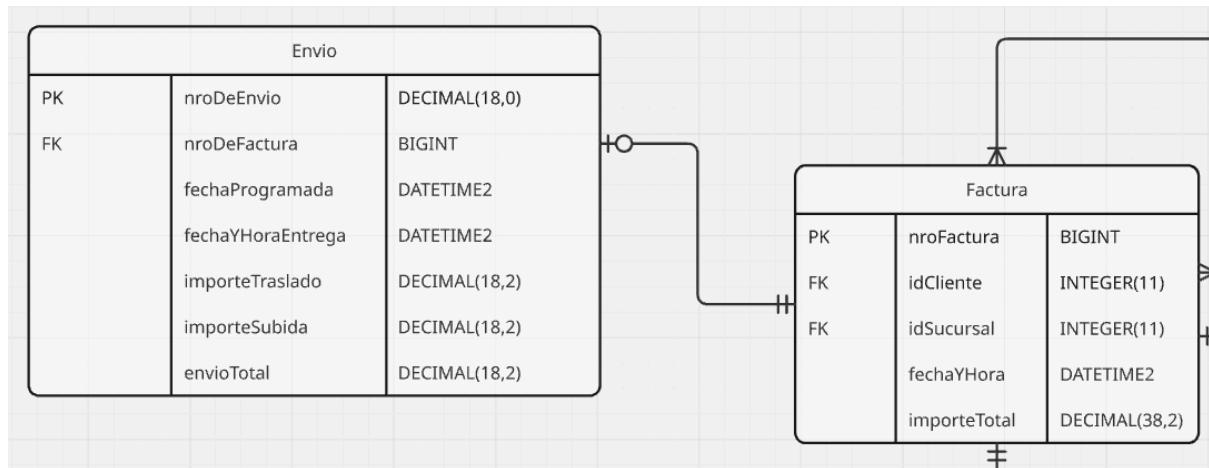


Si bien las tablas ItemDetallePedido e ItemDetalleFactura son parecidas, entendemos que conceptualmente son diferentes y por lo tanto deben tener cada una su propia tabla.

3) Envío

La Entidad Envio tiene todos los campos especificados a la consigna. El campo fechaYHoraEntrega indica la fecha y hora en la que se entregó el pedido, mientras que fecha programada hace referencia a la fecha en la que se debía entregar el pedido.

A su vez, un envío tiene la FK de la factura al cual está asociado, porque los envíos se realizan luego de que el cliente abonó su compra.



4) Compra

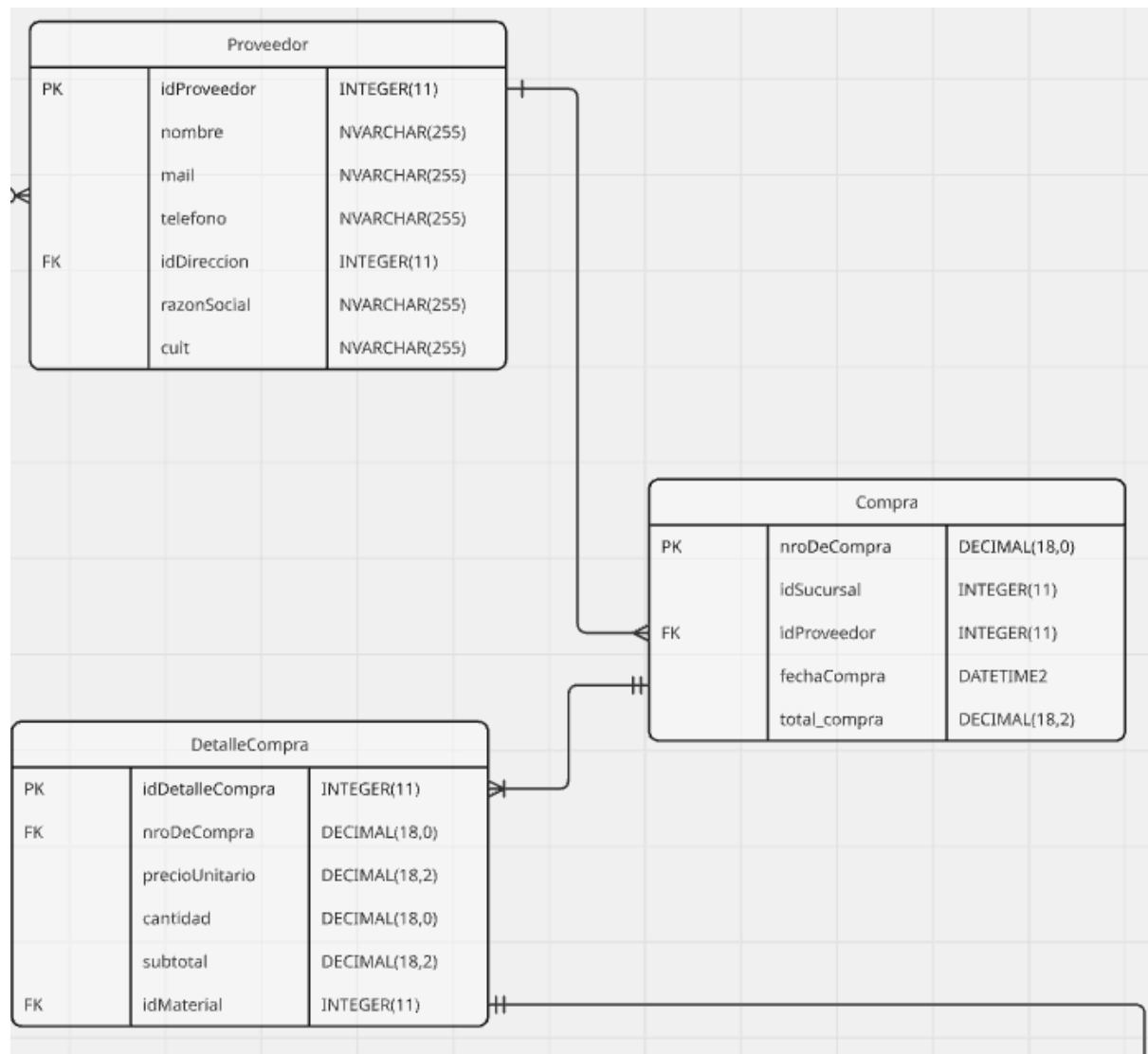
La tabla compra referencia a las compras que la fábrica de sillones realiza a sus proveedores.

La entidad Compra tiene un número el cual identifica unívocamente a una compra. A su vez, tiene una FK a la sucursal que realizó la compra y el proveedor al cual se hizo la compra.

Una compra puede tener muchos renglones (uno por cada material comprado), por eso se decidió modelar la entidad DetalleCompra, por el mismo motivo que se modeló ItemDetalleFactura e ItemDetallePedido.

Por una cuestión de performance, el totalCompra aparece como campo en la entidad Compra.

Por último, detalleCompra tiene una FK al material que fue comprado en ese renglón.



Migración de datos

Dropeo de tablas

- Se elimina cualquier tabla previa antes de volver a correr las consultas, esto permite redefinir la estructura desde cero y evita conflictos de integridad en cada ejecución.

```

3 IF OBJECT_ID('QUERYOSOS.ItemDetallefactura', 'U') IS NOT NULL
4     DROP TABLE QUERYOSOS.ItemDetallefactura;

3 IF OBJECT_ID('QUERYOSOS.ItemDetallePedido', 'U') IS NOT NULL
4     DROP TABLE QUERYOSOS.ItemDetallePedido;

3 IF OBJECT_ID('QUERYOSOS.DetalleCompra', 'U') IS NOT NULL
4     DROP TABLE QUERYOSOS.DetalleCompra;

3 IF OBJECT_ID('QUERYOSOS.MaterialSillon', 'U') IS NOT NULL
4     DROP TABLE QUERYOSOS.MaterialSillon;

3 IF OBJECT_ID('QUERYOSOS.Sillon', 'U') IS NOT NULL
4     DROP TABLE QUERYOSOS.Sillon;

3 IF OBJECT_ID('QUERYOSOS.Relleno', 'U') IS NOT NULL
4     DROP TABLE QUERYOSOS.Relleno;

3 IF OBJECT_ID('QUERYOSOS.Madera', 'U') IS NOT NULL
4     DROP TABLE QUERYOSOS.Madera;

3 IF OBJECT_ID('QUERYOSOS.Tela', 'U') IS NOT NULL
4     DROP TABLE QUERYOSOS.Tela;

3 IF OBJECT_ID('QUERYOSOS.Material', 'U') IS NOT NULL
4     DROP TABLE QUERYOSOS.Material;

3 IF OBJECT_ID('QUERYOSOS.Medida', 'U') IS NOT NULL
4     DROP TABLE QUERYOSOS.Medida;

```

Creación de las tablas

- Se define la estructura completa de cada tabla con columnas y restricciones para asegurar que la tabla destino esté correctamente configurada antes de agregar los datos.

```

1
2 CREATE TABLE QUERYOSOS.Pedido(
3     nroDePedido decimal(18,0) CONSTRAINT PK_Pedido PRIMARY KEY,
4     fechaYHora DateTime2(6),
5     precioTotal decimal(18,2),
6     idCliente Integer,
7     estadoActual Integer,
8     idSucursal Integer,
9     fechaCancelacion DateTime2(6),
10    idMotivoCancelacion Integer
11 )
12 GO
13
14
15 CREATE TABLE QUERYOSOS.ItemDetallePedido (
16     id_item_pedido Integer Identity(1,1) CONSTRAINT PK_Item_Detalle_Pedido PRIMARY KEY,
17     nroDePedido decimal(18,0),
18     idSillon BigInt,
19     cantidad_pedido BigInt,
20     subtotal decimal(18,2),
21     precioUnitario decimal(18,2)
22 )
23
24 CREATE TABLE QUERYOSOS.Factura (
25     nroFactura BIGINT CONSTRAINT PK_Factura PRIMARY KEY,
26     idCliente Integer,
27     idSucursal Integer,
28     fechaYHora DateTime2(6),
29     importeTotal decimal(38,2)
30 )

```

Creación de FKs

- Las claves foráneas se crean con **ALTER TABLE** después de poblar las tablas para asegurar que las columnas de dependencia apunten a registros válidos en las tablas padre, sin interrumpir el proceso de carga de datos.

```
--CREACION DE FOREIGN KEYS
--FOREIGN KEYS PARA PEDIDO---
ALTER TABLE QUERYOSOS.Pedido
ADD CONSTRAINT FK_Pedido_Cliente
FOREIGN KEY (idCliente) REFERENCES QUERYOSOS.Cliente(idCliente)
ALTER TABLE QUERYOSOS.Pedido
ADD CONSTRAINT FK_Pedido_Sucursal
FOREIGN KEY (idSucursal) REFERENCES QUERYOSOS.Sucursal(idSucursal)
ALTER TABLE QUERYOSOS.Pedido
ADD CONSTRAINT FK_Pedido_Estado
FOREIGN KEY (estadoActual) REFERENCES QUERYOSOS.Estado(idEstado)
ALTER TABLE QUERYOSOS.Pedido
ADD CONSTRAINT FK_Pedido_Motivo_Cancelacion
FOREIGN KEY (idMotivoCancelacion) REFERENCES QUERYOSOS.Motivo_cancelacion_pedido(id_motivo_cancelacion)
```

Dropeo procedures

- Se incluye para garantizar que cada procedimiento se elimine automáticamente antes de recrearlo, evitando errores de duplicados al iterar el despliegue.

```
DROP PROCEDURE IF EXISTS Migrar_Provincia
DROP PROCEDURE IF EXISTS Migrar_Localidad
DROP PROCEDURE IF EXISTS Migrar_Direccion
DROP PROCEDURE IF EXISTS Migrar_Sucursal
DROP PROCEDURE IF EXISTS Migrar_Cliente
DROP PROCEDURE IF EXISTS Migrar_Proveedor
DROP PROCEDURE IF EXISTS Migrar_Pedido
DROP PROCEDURE IF EXISTS Migrar_Compra
DROP PROCEDURE IF EXISTS Migrar_Factura
DROP PROCEDURE IF EXISTS Migrar_Modelo_Sillon
DROP PROCEDURE IF EXISTS Migrar_Medida_Sillon
DROP PROCEDURE IF EXISTS Migrar_Material
DROP PROCEDURE IF EXISTS Migrar_Tela
DROP PROCEDURE IF EXISTS Migrar_Madera
DROP PROCEDURE IF EXISTS Migrar_Relleno
DROP PROCEDURE IF EXISTS Migrar_Sillon
DROP PROCEDURE IF EXISTS Migrar_Material_Sillon
DROP PROCEDURE IF EXISTS Migrar_Item_Detalle_Pedido
DROP PROCEDURE IF EXISTS Migrar_Envio
DROP PROCEDURE IF EXISTS Migrar_Detalle_Compra
DROP PROCEDURE IF EXISTS Migrar_Item_Detalle_Factura
GO
```

Uso de procedimientos almacenados


```

3 CREATE PROCEDURE Migrar_Provincia
AS
3 BEGIN

3     INSERT INTO QUERYOSOS.Provincia (nombre)
3     SELECT DISTINCT
3         Maestra.Sucursal_Provincia
3     FROM gd_esquema.Maestra
3     WHERE Maestra.Sucursal_Provincia IS NOT NULL
3         AND NOT EXISTS (
3             SELECT 1
3             FROM QUERYOSOS.Provincia P
3             WHERE P.nombre = Maestra.Sucursal_Provincia
3         );

3     INSERT INTO QUERYOSOS.Provincia (nombre)
3     SELECT DISTINCT
3         Maestra.Cliente_Provincia
3     FROM gd_esquema.Maestra
3     WHERE Maestra.Cliente_Provincia IS NOT NULL
3         AND NOT EXISTS (
3             SELECT 1
3             FROM QUERYOSOS.Provincia P
3             WHERE P.nombre = Maestra.Cliente_Provincia
3         );

3     INSERT INTO QUERYOSOS.Provincia (nombre)
3     SELECT DISTINCT
3         Maestra.Proveedor_Provincia
3     FROM gd_esquema.Maestra
3     WHERE Maestra.Proveedor_Provincia IS NOT NULL
3         AND NOT EXISTS (
3             SELECT 1
3             FROM QUERYOSOS.Provincia P
3             WHERE P.nombre = Maestra.Proveedor_Provincia
3         );
3 END;

GO

```

- Cada tabla del esquema destino se puebla mediante un `CREATE PROCEDURE Migrar_NombreTabla`
- De esta manera, es posible ejecutar cada carga de forma independiente y volver a correrla sin introducir duplicados (gracias a los filtros empleados).

Inserciones con `SELECT DISTINCT` y/o `NOT EXISTS`

- Para las tablas “*maestras*” (por ejemplo, Provincia, Localidad, Dirección, Cliente, Proveedor, etc.), se siguió siempre el patrón:

```

INSERT INTO QUERYOSOS.Provincia (nombre)
SELECT DISTINCT
    Maestra.Sucursal_Provincia
FROM gd_esquema.Maestra
WHERE Maestra.Sucursal_Provincia IS NOT NULL
    AND NOT EXISTS (
        SELECT 1
        FROM QUERYOSOS.Provincia P
        WHERE P.nombre = Maestra.Sucursal_Provincia
    );

```

- `SELECT DISTINCT` evita que se inserten filas duplicadas cuando la misma combinación aparece varias veces en la fuente.
- `NOT EXISTS (...)` comprueba, antes de insertar, que la clave lógica (o combinación de claves) no exista ya en el destino, permitiendo volver a ejecutar el mismo procedimiento sin que se dupliquen registros.

Orden de ejecución según dependencia

- Se migraron todas las tablas padre, y luego las que dependen de ellas. Finalmente se ejecutan secuencialmente:

```
EXEC Migrar_Provincia
EXEC Migrar_Localidad
EXEC Migrar_Direccion
EXEC Migrar_Sucursal
EXEC Migrar_Cliente
EXEC Migrar_Proveedor
EXEC Migrar_Pedido
EXEC Migrar_Compra
EXEC Migrar_Factura
EXEC Migrar_Envio
EXEC Migrar_Modelo_Sillon
EXEC Migrar_Medida_Sillon
EXEC Migrar_Material
EXEC Migrar_Tela
EXEC Migrar_Madera
EXEC Migrar_Relleno
EXEC Migrar_Sillon
EXEC Migrar_Material_Sillon
EXEC Migrar_Item_Detalle_Pedido
EXEC Migrar_Detalle_Compra
EXEC Migrar_Item_Detalle_Factura
```

Modelo de Inteligencia de Negocios (BI)

Para el modelo de inteligencia de negocios, decidimos crear las siguientes dimensiones como entidades, las cuales van a estar referenciadas en las tablas de hechos:

Dimensión BITiempo

La cual contiene referencias al año, cuatrimestre y mes que le corresponda

| BI_Tiempo | | |
|-----------|--------------|--------------|
| PK | idTiempo | INTEGER |
| | año | INTEGER |
| | cuatrimestre | VARCHAR(255) |
| | mes | INTEGER |

Dimensión BITurno

Contiene referencia al horario en donde comienza el turno y al horario donde finaliza el turno

| BI_Turno | | |
|----------|---------|---------|
| PK | idTurno | INTEGER |
| | desde | TIME |
| | hasta | TIME |

Dimensión BIUbicación

Contiene referencia a la localidad,dirección y provincia de la ubicación

| BI_Ubicacion | | |
|--------------|-------------|--------------|
| PK | idUbicacion | INTEGER |
| | provincia | VARCHAR(255) |
| | localidad | VARCHAR(255) |
| | direccion | VARCHAR(255) |

Dimensión BISucursal

Contiene referencia al numero de la sucursal y a la dirección de la sucursal

| BI_Sucursal | | |
|-------------|----------------|--------------|
| PK | idSucursal | INTEGER |
| | numeroSucursal | BIGINT |
| | direccion | VARCHAR(255) |

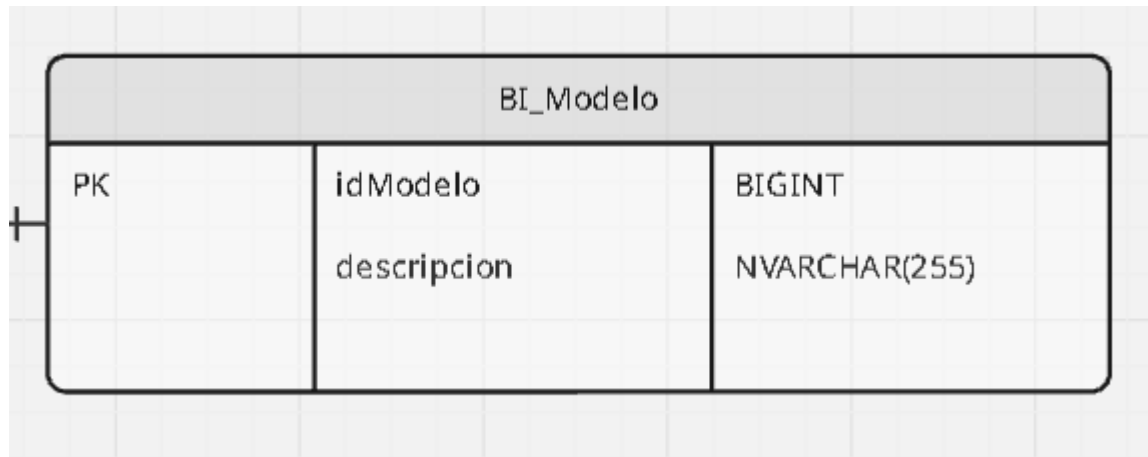
Dimensión BI_EstadoPedido

Contiene el string que describe al estado del hecho “pedido”

| BI_EstadoPedido | | |
|-----------------|------------|-------------|
| PK | idEstadoBI | INTEGER |
| | estado | VARCHAR(50) |

Dimensión BIModelo

Contiene referencia al modelo de sillón, lo cual sirve para que luego a partir de la tabla de hechos “facturación” podamos calcular los modelos de sillón más vendidos para cada cuatrimestre de cada año y por localidad



Dimensión BIMaterial

Contiene referencias al tipo de material, la cual va a estar referenciada por la tabla de hechos “compra” para obtener el importe total gastado por tipo de material

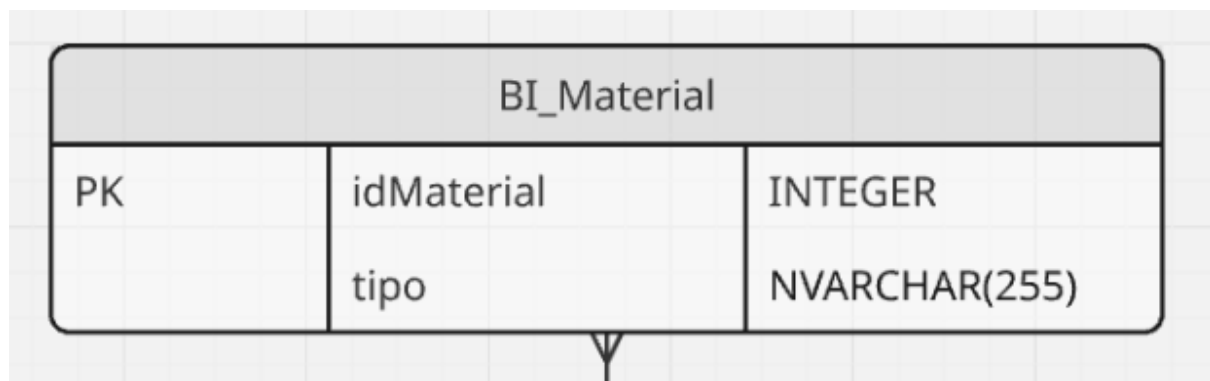


Tabla de Hecho BI_Pedido

Relación con dimensiones:

- **idSucursal** → **BI_Sucursal**: identifica la sucursal que recibió el pedido.
- **idTiempo** → **BI_Tiempo**: permite analizar pedidos por año, cuatrimestre y mes.
- **idTurno** → **BI_Turno**: categoriza el pedido en una franja horaria (ej: 08:00–14:00).
- **idUbicacion** → **BI_Ubicacion**: ubicación geográfica (provincia, localidad, dirección).
- **idEstadoBI** → **BI_EstadoPedido**: estado del pedido (Ej: entregado, cancelado).

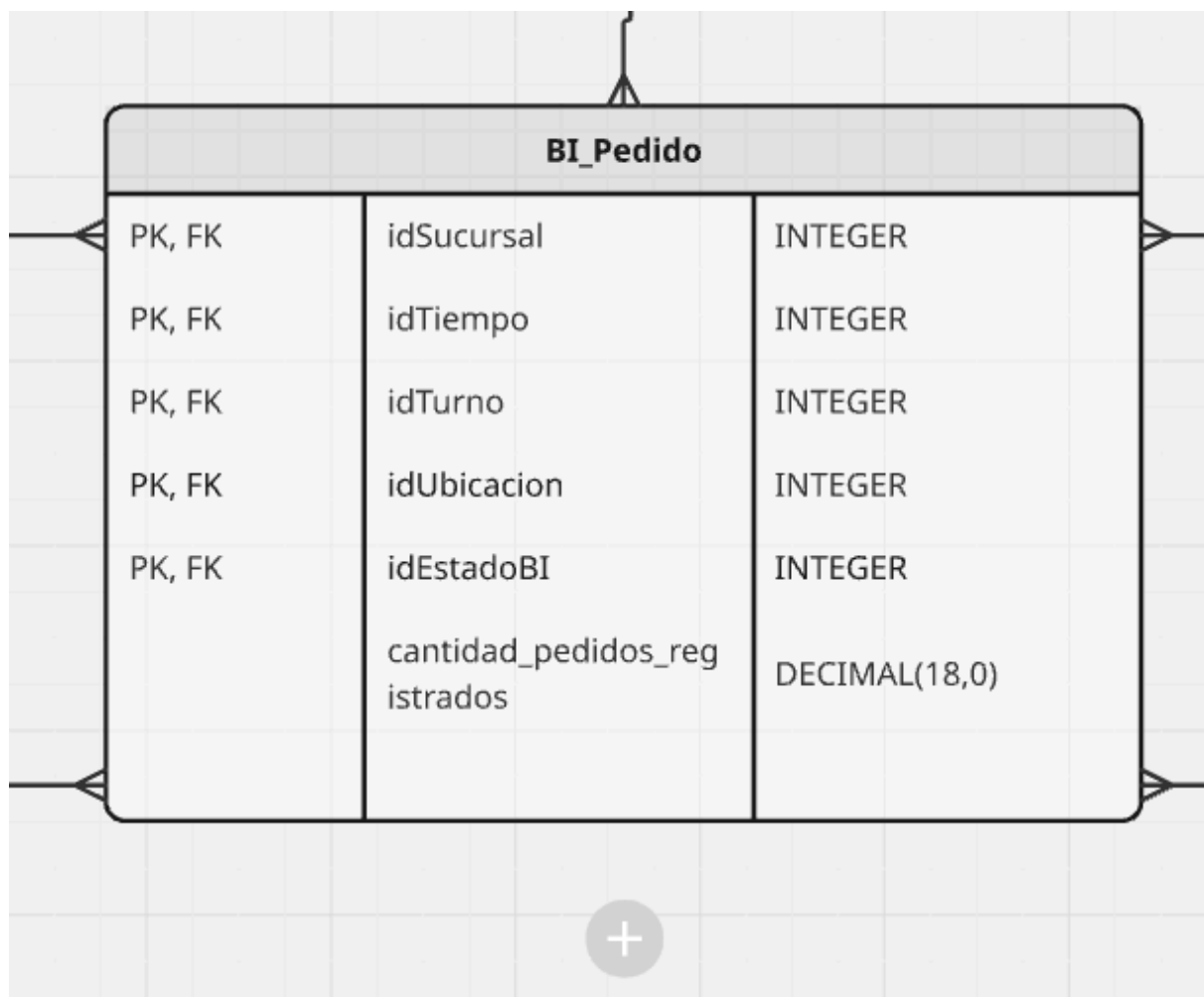


Tabla de Hecho BI_Compra

Almacena los **importes de compras** de materiales, asociando cada gasto con sucursal, período y tipo de material.

Agrega los campos “importe_total_gastado” y “cantidad_compras” por cada combinación de dimensiones.

Relación con dimensiones:

- **idTiempo** → **BI_Tiempo**
- **idSucursal** → **BI_Sucursal**
- **idUbicacion** → **BI_Ubicacion**
- **idMaterial** → **BI_Material**: permite distinguir entre tela, madera o relleno.

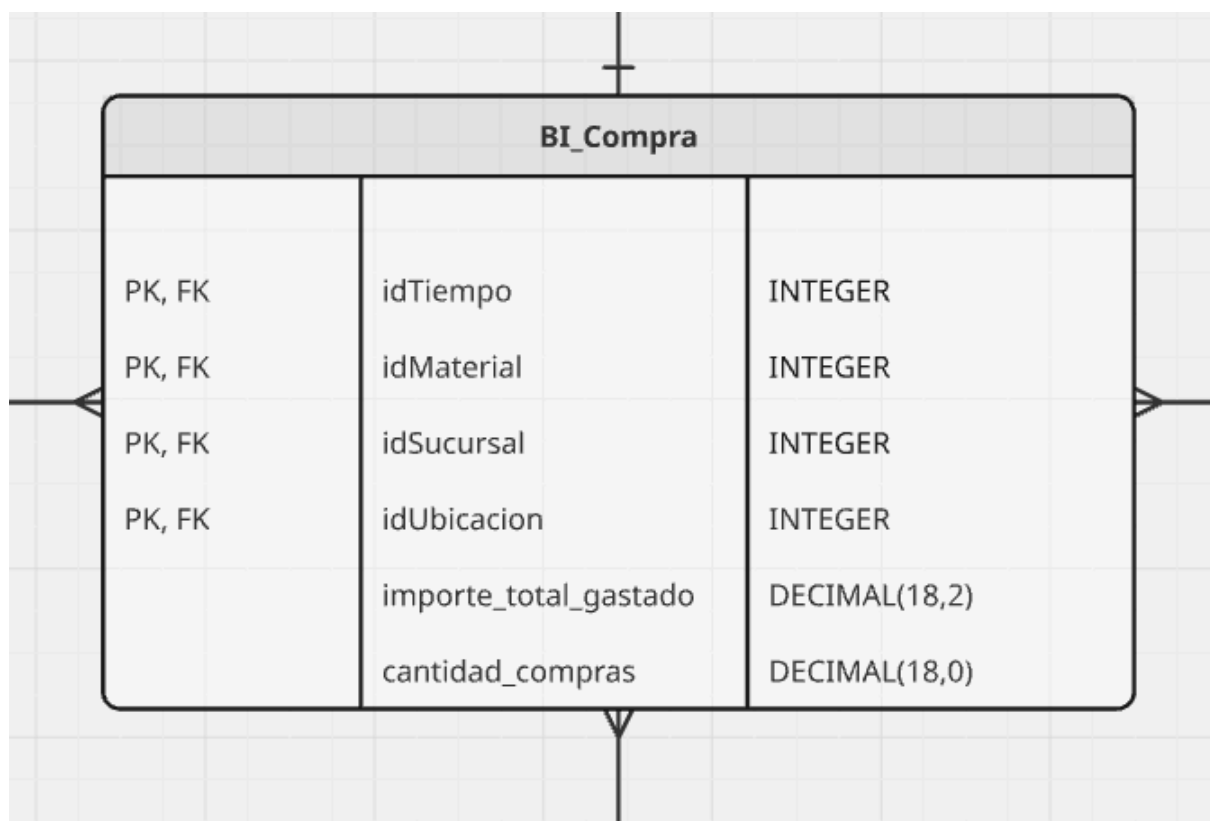


Tabla de Hecho BI_Envio

Agrupar los envíos realizados por cada combinación de dimensiones y como campos adicionales tenemos la cantidad de envíos y los envíos cumplidos, por cada combinación de dimensiones

Relación con dimensiones:

- **idTiempo** → **BI_Tiempo**
- **idUbicacion** → **BI_Ubicacion**



Tabla de Hecho BI Facturacion

Agrupar la facturación por cada combinación de las dimensiones.

Agrega los campos “total ingresos”, “cantidad_facturas” y “tiempo_promedio_fabricacion” para sumarizar esa información por cada combinación de dimensiones.

Relación con dimensiones:

- **idSucursal** → **BI_Sucursal**
- **idTiempo** → **BI_Tiempo**
- **idUbicacion** → **BI_Ubicacion**
- **idModelo** → **BI_Modelo**: tipo de sillón vendido.
- **idRangoEtario** → **BI_RangoEtario**: grupo de edad del cliente.

