

Trabajo Práctico Gestión de datos

“Fábrica de sillones”

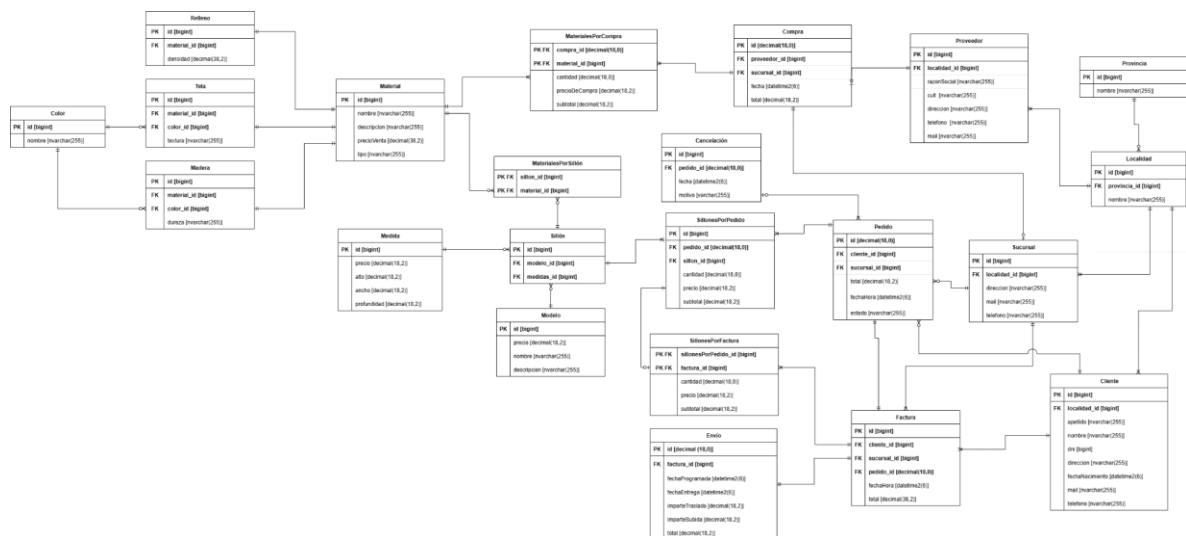
Índice

Modelo Relacional.....	3
DER.....	3
Consideraciones tomadas.....	3
Modificaciones del DER.....	5
Stored Procedures.....	8
Modelo BI.....	8
DER BI.....	9
Consideraciones tomadas.....	9
Stored Procedures BI.....	11

Modelo Relacional

Para la migración de datos de la tabla Maestra a la nueva base de datos se cargaron los datos de la misma en las diferentes tablas creadas. Estas tablas se relacionan según los datos proporcionados, mediante la creación de tablas intermedias y conexiones de Primary Keys y Foreign Keys

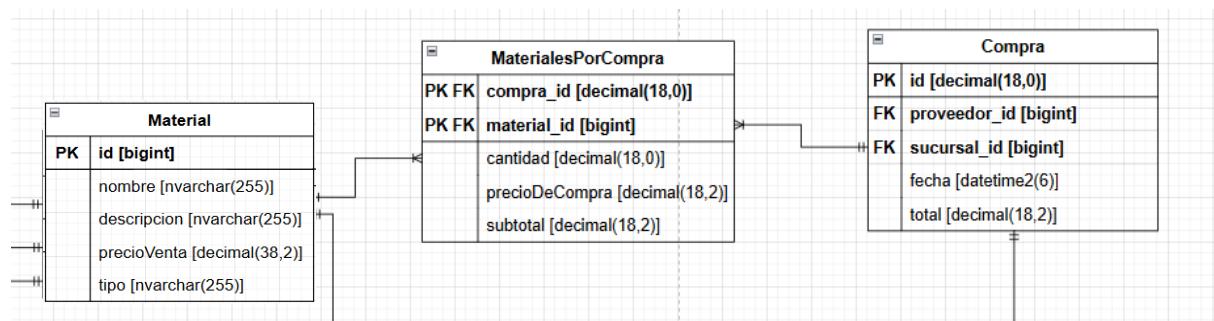
DER



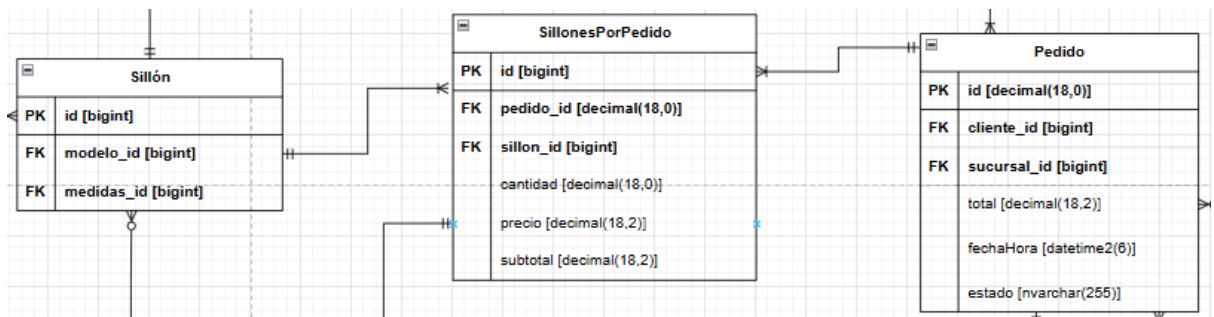
(*) Como se solicita en la consigna, y por motivos de resolución de la imagen, se adjunta un archivo aparte con el DER en formato JPEG.

Consideraciones tomadas

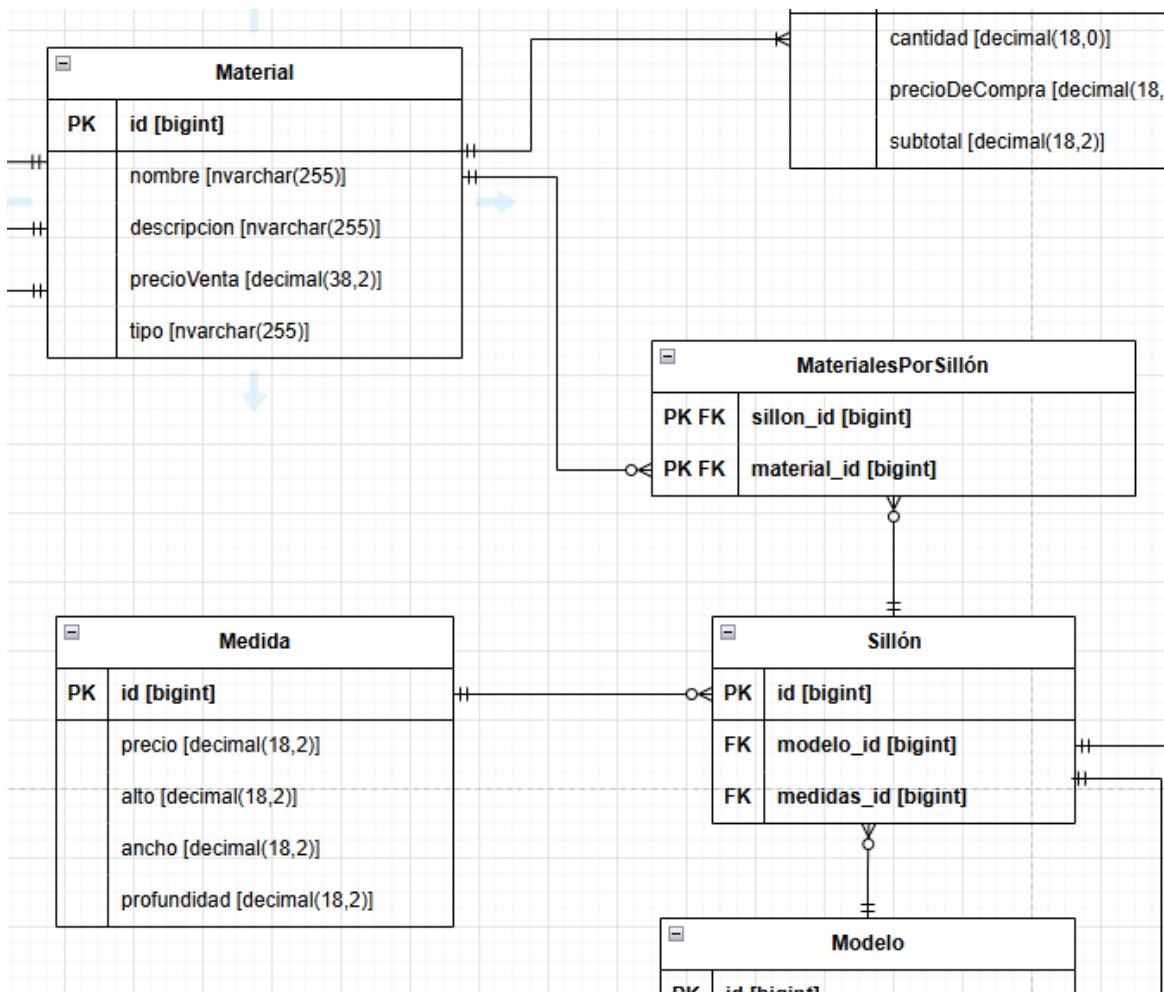
- Se creó una tabla intermedia entre las entidades Compra y Material para representar el detalle de la compra



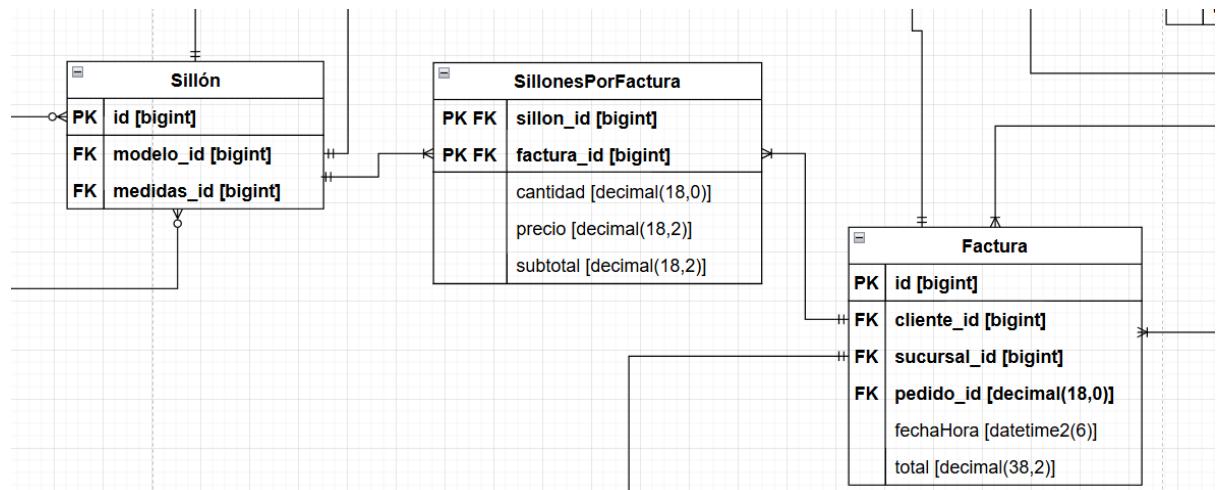
- Se creó una tabla intermedia entre las entidades Sillón y Pedido para representar el detalle del pedido



- Se creó una tabla intermedia entre las entidades Sillón y Material para crear la relación entre los distintos materiales que tendrá un sillón con el sillón correspondiente

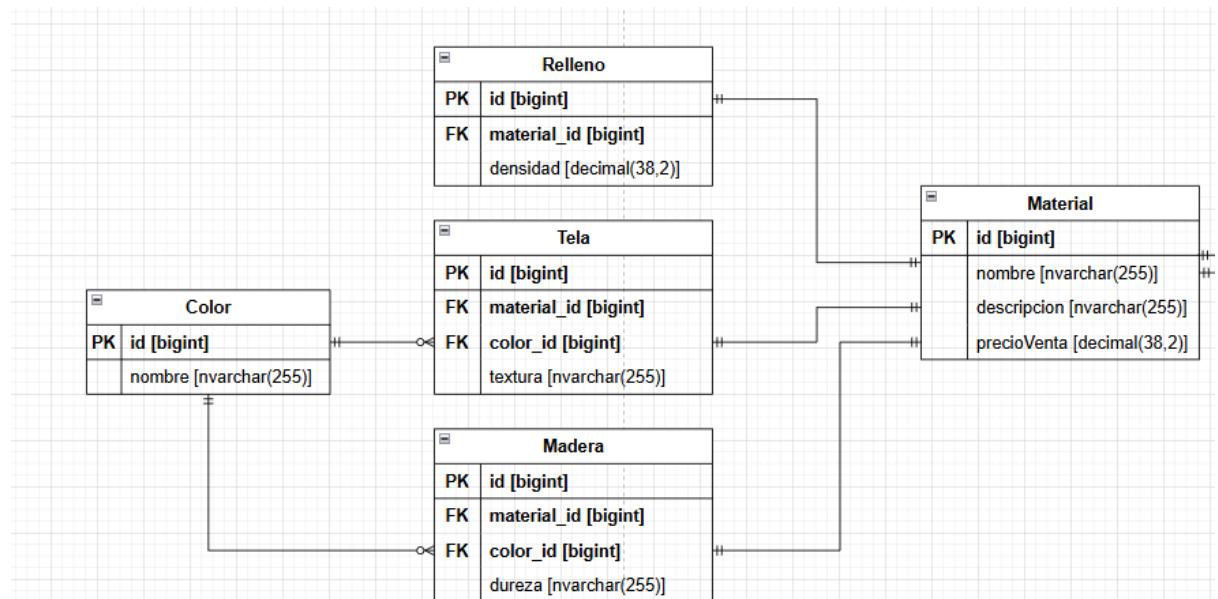


- Se creó la tabla intermedia SillonesPorFactura para representar el detalle de la factura



Modificaciones del DER

- Actualmente la entidad Materiales se encuentra representada de esta forma, como un “Joined” o “Herencia”.

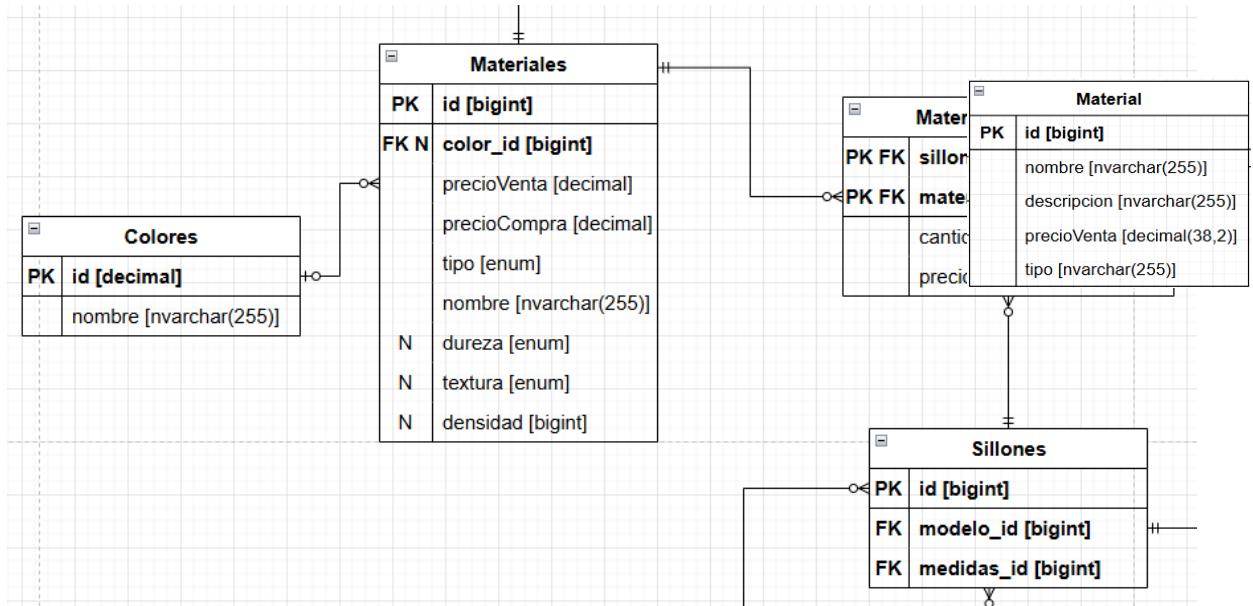


Material anteriormente era una única tabla con el campo “Tipo”.

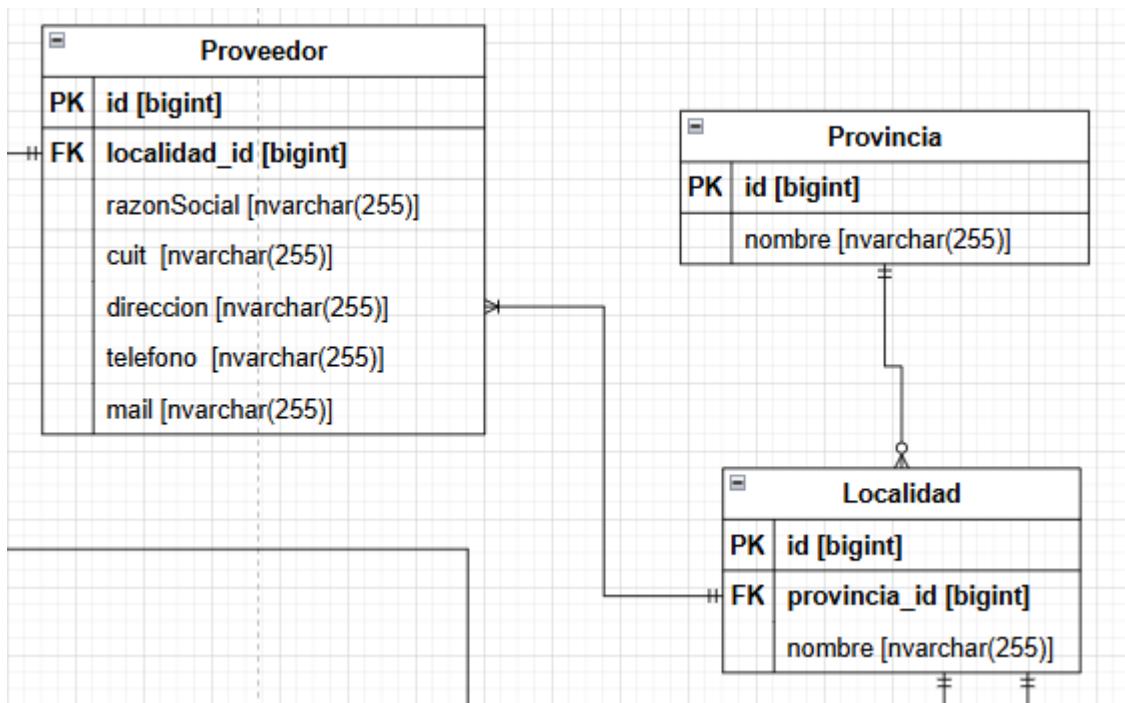
Como los distintos tipos de materiales no tenían los mismos campos, se contempló la posibilidad de que estos campos quedarán en NULL en caso de no ser utilizados.

También se modificó la entidad MaterialesPorSillon. Actualmente solo es una tabla intermedia con los identificadores de las tablas

que relaciona. Se quitaron los campos “cantidad” y “precioDeVenta” dado que no tenían utilidad.

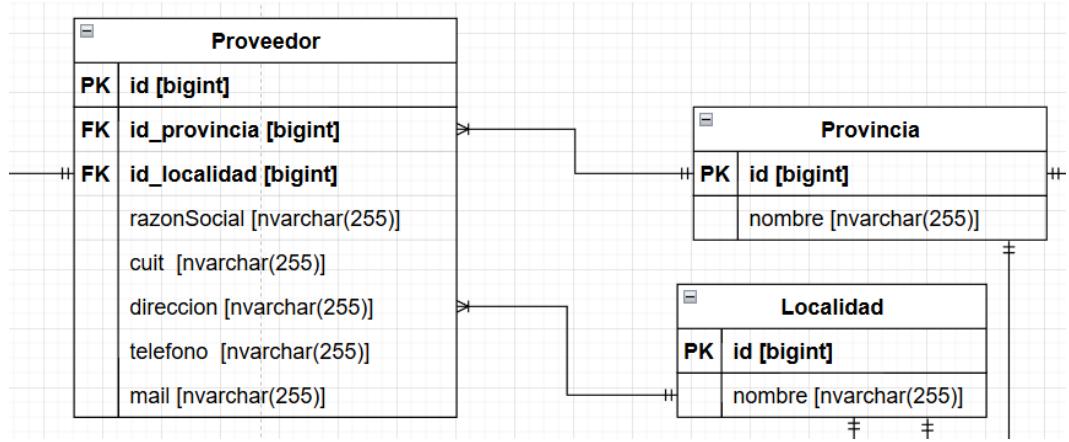


- La entidad Localidad se relaciona con Provincia para poder crear la dimensión Ubicación, en el Modelo BI, que relaciona a cada localidad con su respectiva provincia. Ahora las entidades que tenían relación con Provincia y Localidad, solo tendrán relación con Localidad de la siguiente forma:

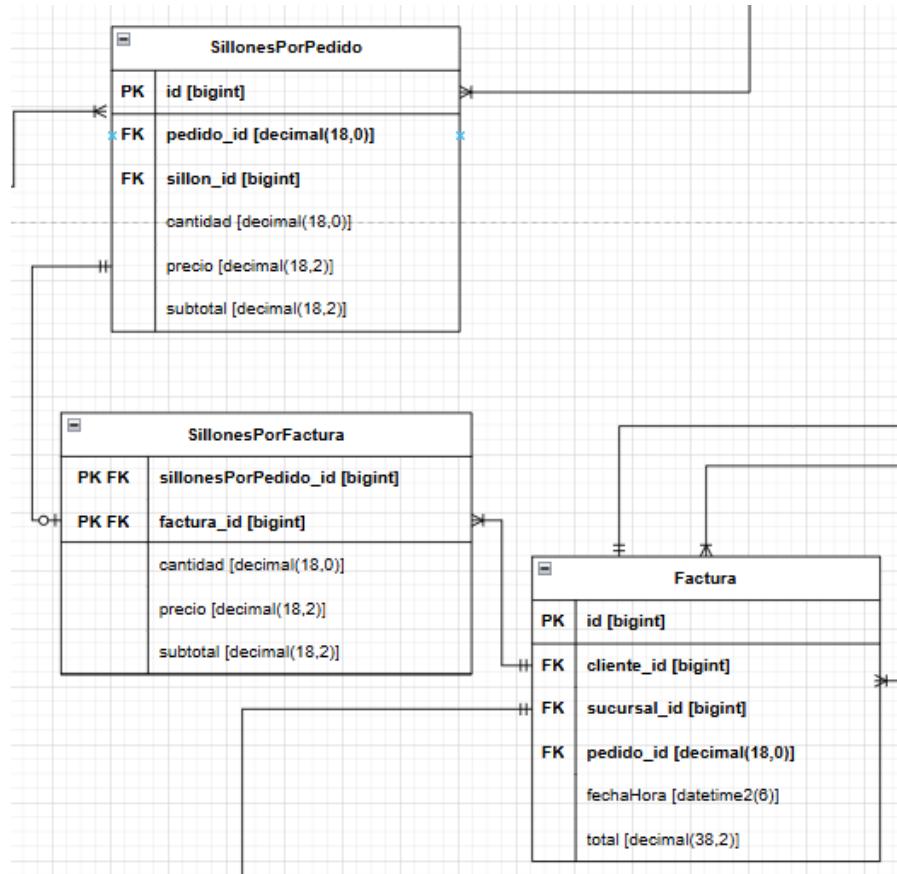


Anteriormente Localidad y Provincia no se relacionaban entre sí, sino que cada entidad se relacionaba independientemente con

cada localidad y con cada provincia, por lo que se generaba un problema cuando queríamos relacionar las mismas al momento de crear la dimensión ubicación.



- Para la entrega final se decide cambiar el modelo y ubicar a SillonesPorFactura entre SillonesPorPedido y Factura porque necesitábamos la información de cada sillón del detalle del pedido por factura.



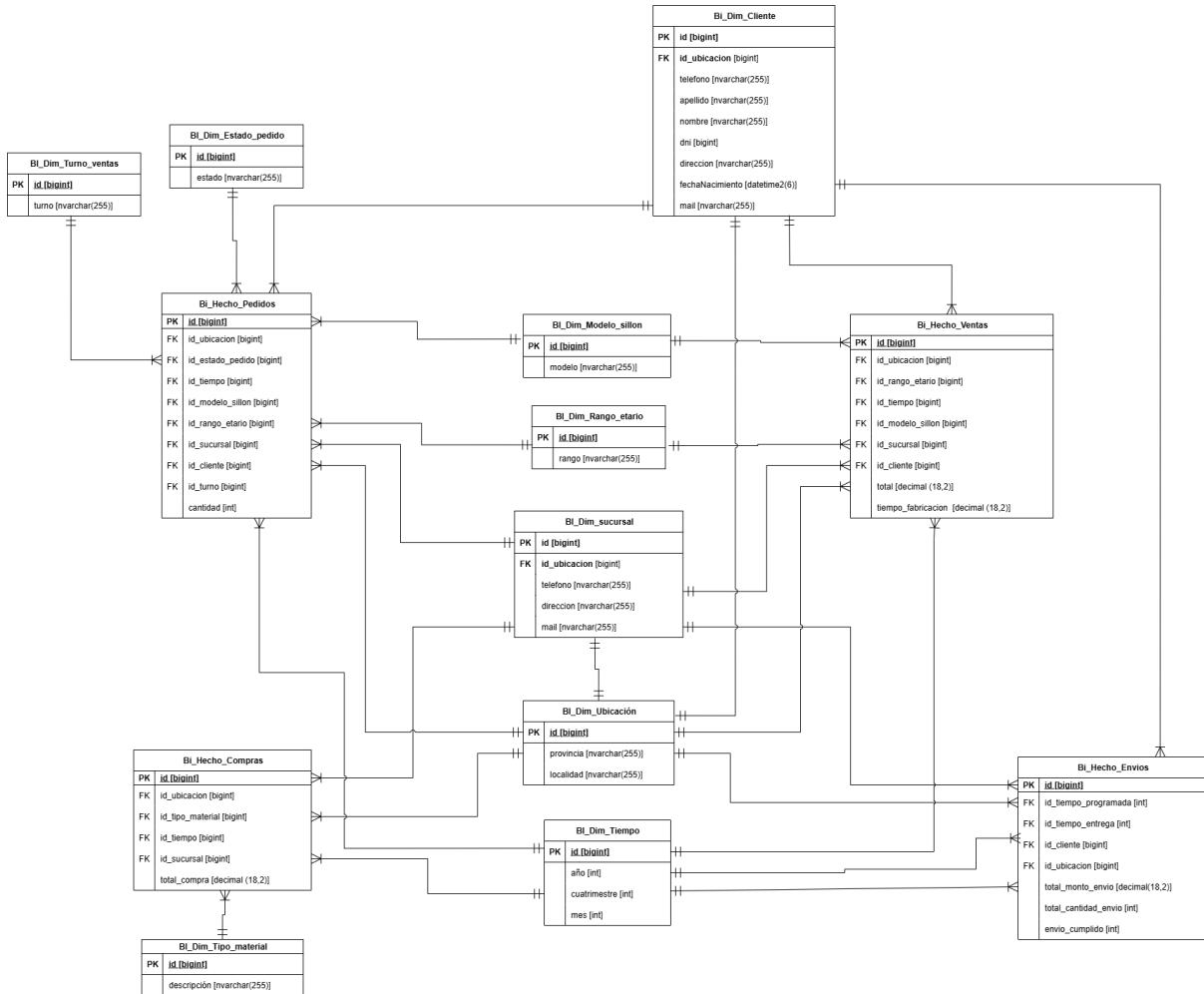
Stored Procedures

- **Creación de esquema:** verificación de existencia y creación del esquema SOMOS.QUERY.LORDS.
- **Creación de tablas:** verificación de existencia y creación de las tablas que coinciden con cada una de las entidades del DER.
- **Inserción de datos:** existe un stored procedure por cada tabla el cual toma los datos de las columnas correspondientes de la tabla maestra y de las tablas del nuevo esquema, en caso de ser necesario, y los inserta en la tabla.

Modelo BI

Para la migración de datos de la base de datos relacional creada SOMOS_QUERY_LORDS a la nueva base de datos multidimensional se cargaron los datos necesarios de la misma en las diferentes tablas creadas. En algunas tablas se desnormalizaron los datos para mayor eficiencia al momento de crear las vistas. Estas tablas se relacionan entre sí mediante la creación de tablas de hechos utilizando PKs y FKs.

DER BI



(*) Como se solicita en la consigna, y por motivos de resolución de la imagen, se adjunta un archivo aparte con el DER_BI en formato JPEG.

Consideraciones tomadas

- Se crean 4 tablas de hechos, según la información necesaria a analizar para la creación de los tableros de control a nivel gerencial:
 - **BI_HECHOS_ENVÍOS:**
Para las vistas 9 y 10 se crea una tabla de hechos donde se reflejan los envíos efectuados. Esta tabla contiene el monto total del envío, la cantidad total de artículos del envío y un atributo que identifica si el envío cumple con la fecha estimada de entrega.

El envío tiene asociado el tiempo programado de envío, el tiempo de entrega, el cliente y la ubicación.

- BI_HECHOS_COMPRAS:

Para las vistas 1, 7 y 8 se crea una tabla de hechos donde se reflejan las compras efectuadas. Esta tabla contiene el costo total de la compra.

La compra tiene asociada el tiempo(momento donde ocurre), la ubicación, el tipo de material y la sucursal que realiza la compra.

- BI_HECHOS_PEDIDOS:

Para las vistas 4 y 5 se crea una tabla de hechos donde se reflejan los pedidos efectuados. Esta tabla contiene la cantidad total de pedidos.

El pedido tiene asociado el tiempo programado en el cual se genera un pedido, la ubicación, el estado del pedido, el modelo del sillón, el cliente, el rango etario, la sucursal y el turno.

- BI_HECHOS_VENTAS:

Para las vistas 1, 2, 3 y 6 se crea una tabla de hechos donde se reflejan las ventas efectuadas.

En esta tabla representamos las ventas que se realizan satisfactoriamente. Incluimos las dimensiones de tiempo, sucursal, cliente, modelo de sillón, turno de venta, ubicación y rango etario. También agregamos una métrica de tiempo de fabricación, la cual representa la diferencia entre la fecha de pedido y la fecha de facturación.

- Se agregaron nuevas dimensiones para un análisis de datos más eficiente:

- Tiempo: los datos de esta dimensión son generados por el procedimiento “sp_generar_dim_tiempo”, el cual genera

- todas las fechas (año, cuatrimestre, mes) desde la primer fecha de facturación hasta la última.
- Turno ventas: se distinguen 2 turnos en los que se realizan las ventas, para poder calcular estadísticas según el horario. Para obtener el id correspondiente al turno de la venta, creamos la función “ObtenerIdTurno(@horaVenta)”.
- Ubicación: se busca facilitar y agrupar en una tabla la localidad y su provincia asociada.
- Rango etario: tal como pide el enunciado, es importante poder diferenciar las operaciones según el rango etario del cliente. Para facilitarlo, se crea esta dimensión para definir de manera única los rangos. Para obtener el id correspondiente al rango etario del cliente, creamos la función “ObtenerIdRangoEtario(@fechaNacimiento)”.

Stored Procedures BI

- **Creación de tablas:** verificación de existencia y creación de las tablas que coinciden con cada una de las entidades del DER_BI.
- **Inserción de datos:** existe un stored procedure por cada tabla el cual toma los datos desde el modelo relacional y los inserta en la tabla en el caso de que sean dimensiones.
Para los hechos se insertan las FKs necesarias para relacionar los datos y puede tener algún o algunos atributos extras para una mayor eficiencia al momento de crear las vistas.
- **Creación de Views:** se implementaron vistas sobre las tablas de hechos y dimensiones para poder responder los distintos requerimientos de negocio planteados.