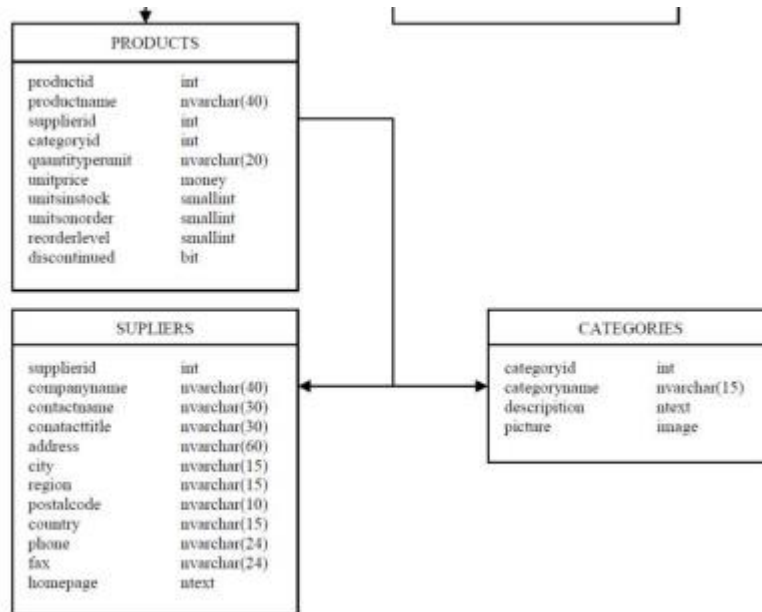


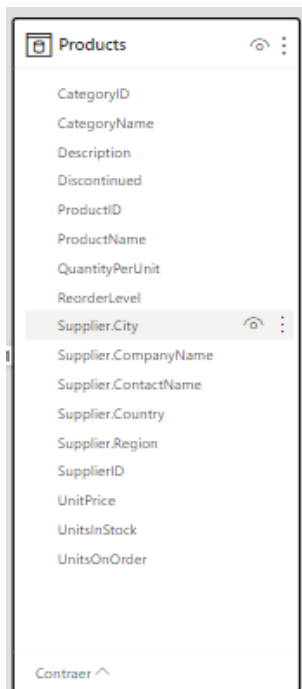
## DESCRIPCION DE LOS PROCESOS REALIZADOS EN LA INTERFAZ DE POWER BI

SE unifican las tablas PRODUCTS, SUPLIERS y CATEGORIEN en una sola "PRODUCTS" esto para lograr un modelo lo más parecido a una ESTRELLA

TABLAS ANTERIORES:



ACTUAL:



Esto para facilitar la búsqueda de los índices a una tabla en lugar de 3.

Se unifican las tablas orders y orders\_detalle, esto para crear una sola tabla de hechos y lograr un modelo estrella

TABLAS INICIALES:

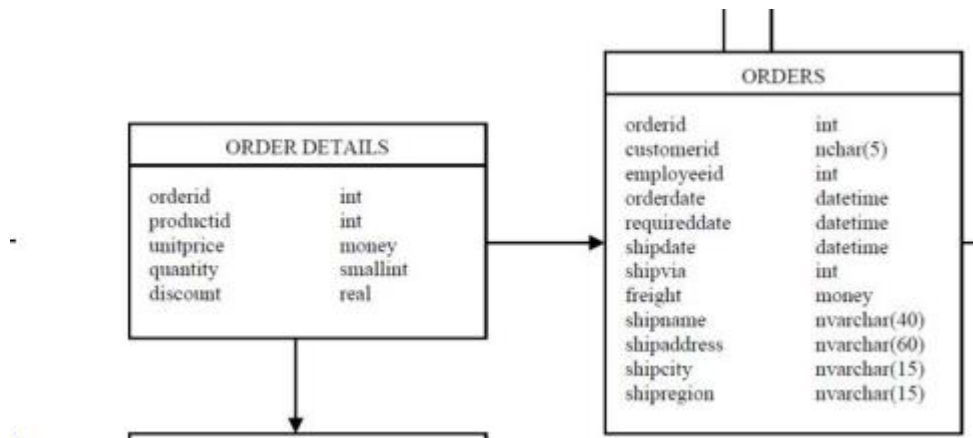


TABLA FINAL:

The screenshot shows the final 'Orders' table structure. The columns are: CustomerID, EmployeeID, Freight, Order\_Details.Discount, Order\_Details.ProductID, Order\_Details.Quantity, Order\_Details.UnitPrice, OrderDate, OrderID, RequiredDate, ShipAddress, ShipCity, ShipCountry, ShipName, ShippedDate, ShipperID, ShipRegion, and CANTIDAD. The table is marked with a star (\*) and a plus sign (+).

Column	Data Type
CustomerID	int
EmployeeID	int
Freight	money
Order_Details.Discount	real
Order_Details.ProductID	int
Order_Details.Quantity	smallint
Order_Details.UnitPrice	money
OrderDate	datetime
OrderID	int
RequiredDate	datetime
ShipAddress	nvarchar(60)
ShipCity	nvarchar(15)
ShipCountry	nvarchar(15)
ShipName	nvarchar(40)
ShippedDate	datetime
ShipperID	int
ShipRegion	nvarchar(15)
CANTIDAD	smallint

Se crea la DIMENSION CALENDARIO:

## Calendar

Opciones

```
let
    // Definir la fecha de inicio y fin para el rango de calendario
    FechaInicio = #date(1996, 1, 1),
    FechaFin = #date(1998, 12, 31),

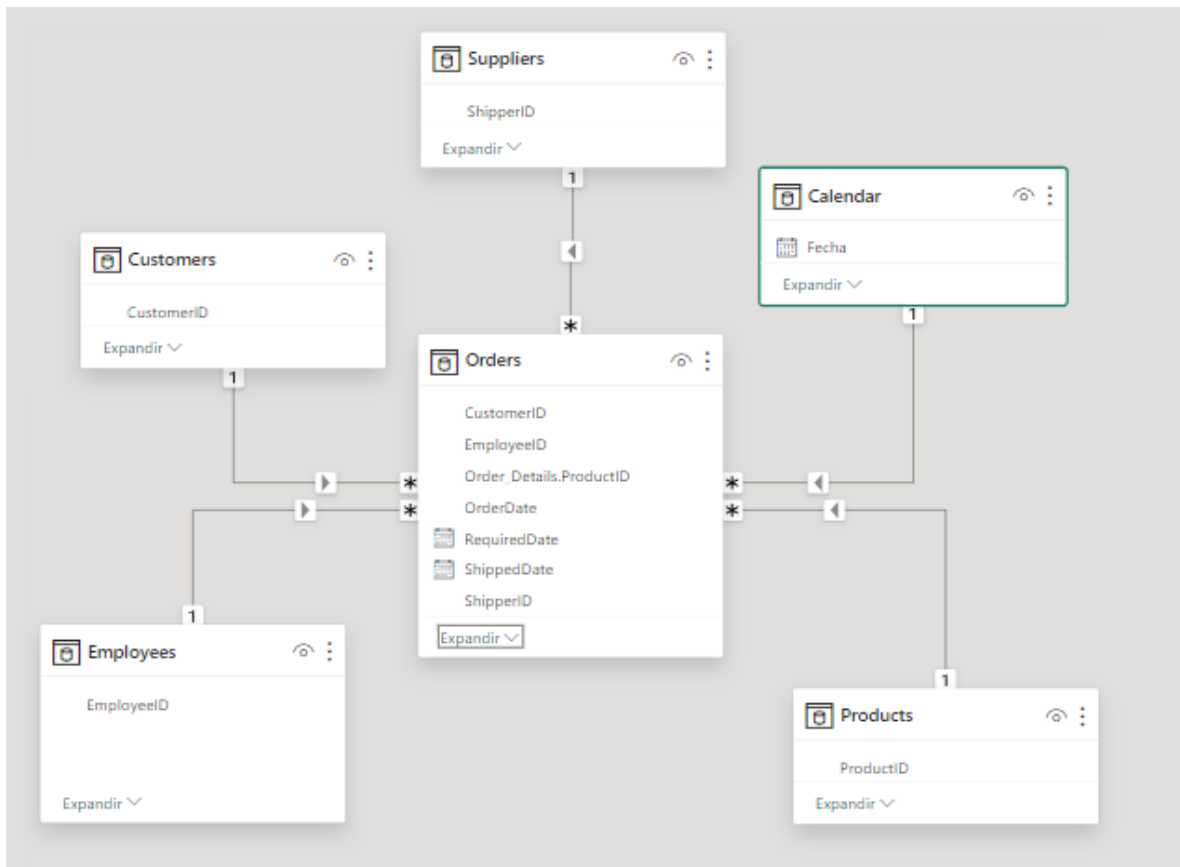
    // Generar una lista de fechas desde la fecha de inicio hasta la fecha de fin
    ListaFechas = List.Dates(FechaInicio, Number.From(FechaFin - FechaInicio) + 1, #duration(1, 0, 0, 0)),

    // Convertir la lista de fechas en una tabla
    TablaCalendario = Table.FromList(ListaFechas, Splitter.SplitByNothing(), {"Fecha"}, null, ExtraValues.Error),

    // Agregar columnas para día, mes, año y trimestre
    #"Añadir Año" = Table.AddColumn(TablaCalendario, "Año", each Date.Year([Fecha])),
    #"Añadir Mes" = Table.AddColumn(#"Añadir Año", "Mes", each Date.Month([Fecha])),
    #"Añadir Nombre Mes" = Table.AddColumn(#"Añadir Mes", "Nombre Mes", each Date.ToText([Fecha], "MMMM")),
    #"Añadir Día" = Table.AddColumn(#"Añadir Nombre Mes", "Día", each Date.Day([Fecha])),
    #"Añadir Trimestre" = Table.AddColumn(#"Añadir Día", "Trimestre", each Date.QuarterOfYear([Fecha])),
    #"Tipo cambiado" = Table.TransformColumnTypes(#"Añadir Trimestre",{{"Fecha", type date}, {"Mes", Int64.Type}})
in
    #"Tipo cambiado"
```

Esto para facilitar el análisis temporal por variables como trimestres, semestre, etc.

Logrando el siguiente modelo:



**Modelo Estrella:** El diseño del modelo se basa en un esquema de estrella, donde tienes una tabla de hechos (Orders.) en el centro y tablas de dimensiones alrededor (Calendar, Customers etc.). Este diseño simplifica las consultas y optimiza el rendimiento del modelo.

**Conexiones Bien Establecidas:** Las relaciones entre las tablas están correctamente establecidas. Esto garantiza que las consultas sean eficientes y que puedas realizar análisis de manera rápida y precisas.

**Eliminación de Datos Duplicados:** Se han eliminado los datos duplicados en las diferentes tablas. Esto reduce la redundancia de los datos y el tamaño del modelo, lo que mejora el rendimiento general y la eficiencia del modelo.

**Uso de Medidas DAX:** Se han utilizado medidas DAX para calcular métricas específicas y dinámicas en lugar de almacenar resultados precalculados. Esto ayuda a reducir la cantidad de datos almacenados y mejora la flexibilidad del modelo.

**Filtros y Inteligencia de Tiempo:** Se han implementado filtros y funciones de inteligencia de tiempo para permitir una interacción fácil y dinámica con los datos. Esto facilita la exploración de los datos y proporciona una experiencia de usuario mejorada.

En resumen, el modelo está optimizado para un rendimiento eficiente al minimizar la redundancia de los datos, establecer relaciones adecuadas entre las tablas, utilizar medidas DAX para cálculos dinámicos y proporcionar funcionalidades de filtrado e inteligencia de tiempo para una mejor interacción con los datos.