

Evidencia de aprendizaje 4.

Proyecto integrador-Repositorio de todas las actividades

Luis Fernando Navarro Mercado

Shara Milena Mosquera Goez

Julio Cesar Muñoz Rios

Jessica Lizeth Moncada Gallego

Institución Universitaria Digital de Antioquia

Base de datos II

Victor Hugo Mercado

PREICA2401B010094

2024

Introducción

En la actualidad la capacidad de organizar y analizar grandes volúmenes de datos es de gran importancia para cualquier empresa que desee ser competitiva. Los datos sin analizar y sin un orden específico, pueden ayudar a la toma de decisiones no acertadas.

Este proyecto se enfoca en transformar los datos de la base de datos Jardinería para construir un entorno de datos estructurado, para poder así darle el uso requerido a los datos.

El proyecto se divide en tres etapas clave:

Construcción de un Modelo Estrella para un Data Mart: En esta etapa, diseñamos y creamos un modelo estrella que permite analizar y validar datos relacionados con las ventas, logrando así saber el producto más vendido, a qué categoría pertenece, en qué año se obtuvo mayores ventas, entre otros.

Creación de una Base de Datos Staging: Construimos consultas SQL para trasladar los datos más importantes desde la base de datos Jardinería a una base de datos Staging. Importante el análisis para tener claro los datos requeridos y crear la base de datos de una manera eficaz.

Desarrollo de un Proceso ETL Completo: En la última etapa, implementamos un proceso de extracción, transformación y carga (ETL) que mueve los datos desde su origen en la base de datos Jardinería, pasando por la base de datos Staging, hasta el data mart final.

Objetivo general:

Desarrollar un proceso integral de gestión de datos que inicie con la recolección y transformación de datos de la base de datos Jardinería, para luego realizar su almacenamiento en un data mart optimizado, utilizando un modelo estrella y una base de datos Staging para asegurar que los datos sean de alta calidad, coherentes y útiles para su análisis.

Objetivos específico:

- Diseñar un modelo de datos que refleje la estructura de los registros de jardinería que van a almacenarse en la base de datos Staging, definiendo así los tipos de datos correctos y logrando coherencia entre los mismos.
- Construir y ejecutar consultas que permitan traer los registros y validar que los datos se encuentren correctos en la base de datos.
- Implementar un proceso de transformación de datos en el cual se validan los datos requeridos, asegurando su calidad y coherencia antes de ser almacenados en el data mart final.

Planteamiento del problema

Solo un pequeño porcentaje de la información es ordenada, lo que puede causar errores si no cuidamos la calidad de los datos.

Al analizar la base de datos jardinería logramos identificar ciertos datos relevantes para los cuales necesitamos: facilitar la extracción de dichos datos, poder hacer consultas más rápidas y eficientes y mejorar la calidad de los datos. Por ello nos planteamos la opción de usar una base de datos staging que nos ayude a solucionar el problema antes mencionado.

Análisis del problema

La base de datos jardinería requiere hacer un proceso de staging el cual consiste en utilizar como un área temporal de almacenamiento y procesamiento durante el proceso de extracción, transformación y carga (ETL) de datos, pero se debe enfocar en datos importantes que respondan o resuelvan preguntas clave, como por ejemplo: ¿Cuál fue el año con más ventas?, ¿cuál fue el producto más vendido? y ¿cuál es la categoría con más productos?. Si queremos responder a estas preguntas, debemos realizar un proceso staging donde se puedan extraer datos que respondan a estas preguntas.

Correcciones a la entrega 1.

Imagen 1.

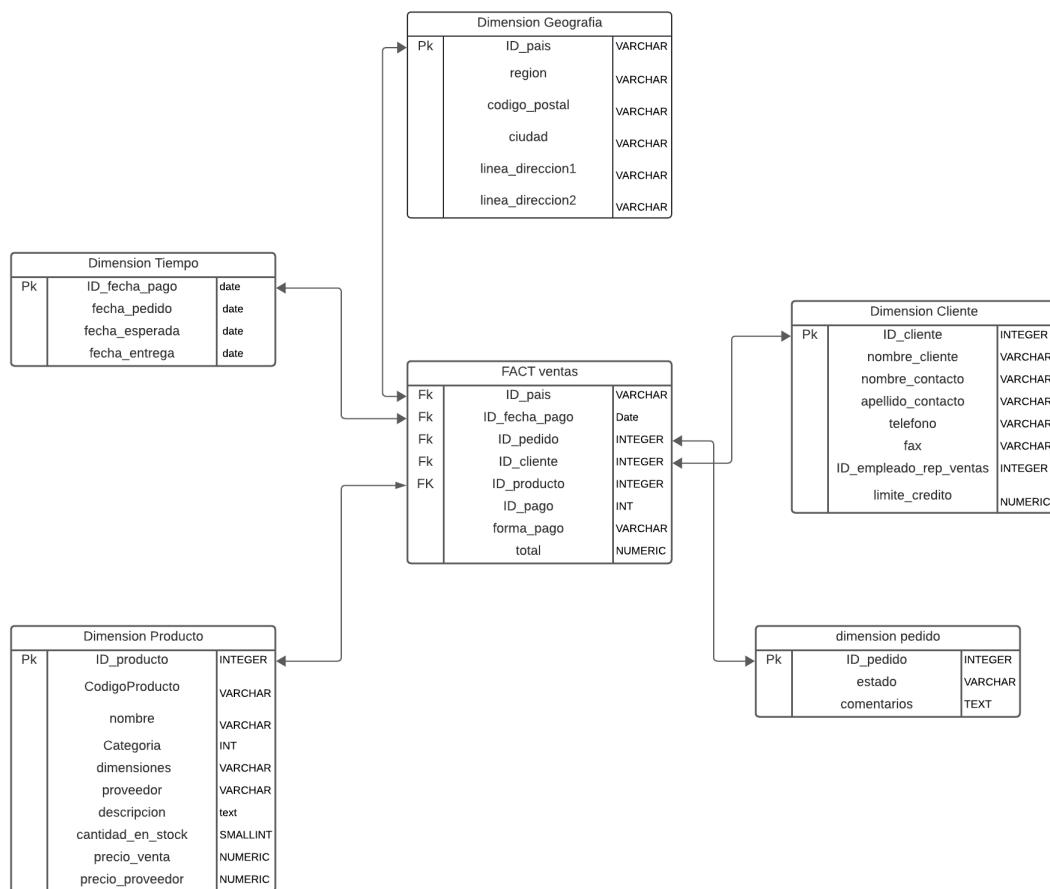
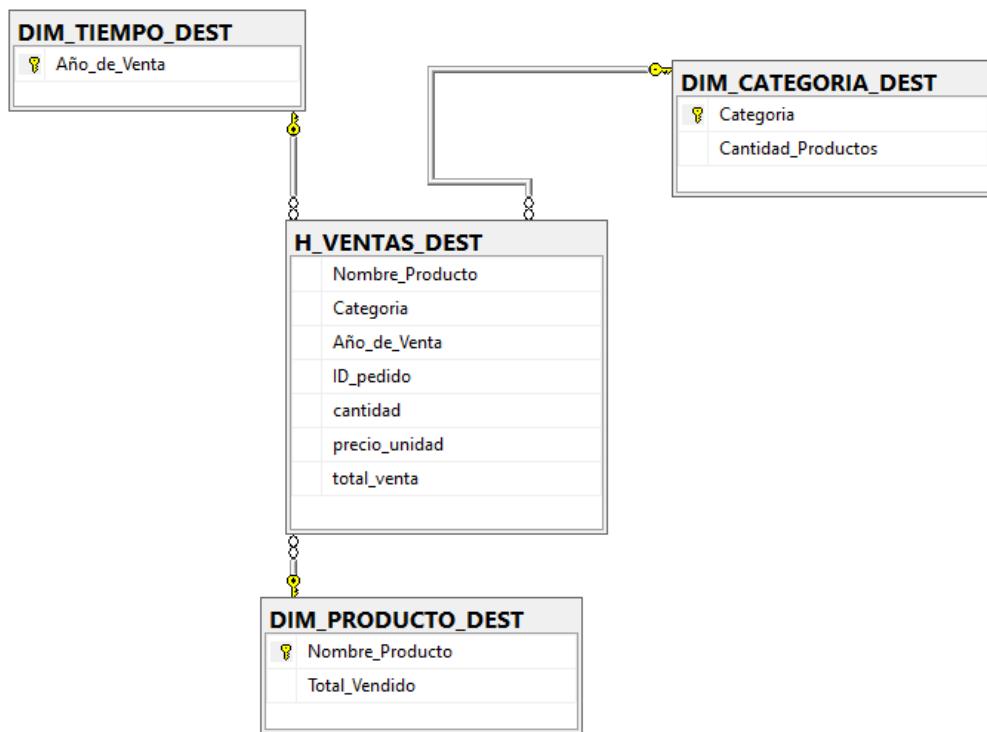


Imagen 2.



Correcciones.

En la imagen 1 se puede apreciar que está formada por 5 dimensiones las cuales son producto, tiempo, geografía, cliente, pedido y la tabla de hechos ventas. En la imagen 2 está formada por las dimensiones producto, tiempo, categoría y la tabla de hechos ventas.

En la primera imagen del modelo estrella se generaliza demasiado, donde se recopilan datos de muchas tablas de la base de datos jardinería que no son relevantes para reflejar o resolver las preguntas en concreto que necesita la empresa jardinería. En la imagen dos, se extrae información verdaderamente relevante. Las consultas se centraron en: ¿Cuál fue el año con más ventas?, ¿cuál es la categoría con más productos? y ¿cuál es el producto más vendido? Esta información se encuentran en las tablas de producto, categoria_producto, detalle_pedido y pedido.

Descripción del análisis realizado a los datos Jardinería y cómo estos se trasladaron a la base de datos Staging.

El análisis se centró en la información más relevante de la base de datos jardinería que resolvieran las preguntas anteriormente mencionadas y las que extrajeran sus posibles respuestas. En la base de datos origen ETLjardineria se busca recopilar datos de la tabla producto, detalle_pedido, pedido y categoria_producto, después de saber específicamente cual son esos datos se hacen consulta para elaborar las dimensiones y la tabla de hechos en la creación se formaron las dimensiones tiempo, producto, categoría y ventas.

Descripción de la consulta para la creación de la dimensión tiempo:

La consulta se realizó sobre la tabla pedido, que contiene la información de productos pedidos se extrae datos de la columna “fecha_pedido”, que almacena la fecha en que se realizó cada pedido. En esta columna utilizares una consulta para extraer el año de la fecha almacenada en “fecha_pedido”. Esto convierte la fecha en un valor numérico que representa el año. Esta consulta SQL extrae y muestra los años en los que se realizaron pedidos, agrupado con la fecha, el mes y el dia y ordenado de manera descendente (del más reciente al más antiguo). Esto proporciona una visión general de la distribución de pedidos a lo largo de los años en la tabla DIM_TIEMPO_DEST y en la tabla de hechos.

Descripción de la consulta para la creación de la dimensión producto:

Esta consulta recopila información de las tablas producto y detalle_pedido, calcula la cantidad total vendida de cada producto, agrupa los resultados por nombre de producto y los ordena de mayor a menor cantidad vendida. Esto proporciona una lista clasificada de productos según sus ventas totales.

Descripción de la consulta para la creación de la dimensión categoría:

Esta consulta busca obtener una lista de categorías de productos junto con la cantidad de productos asociados a cada una de esas categorías. Para ello, utiliza una combinación (LEFT JOIN) entre la tabla Categoria_producto y producto, donde cada categoría en Categoria_producto puede tener cero o más productos en producto. Luego, utiliza la función de agregación COUNT para contar la cantidad de productos (ID_producto) para cada categoría. Estos resultados se agrupan (GROUP BY) por la descripción de la categoría (Desc_Categoría) y se ordenan (ORDER BY) de manera descendente según la cantidad de productos asociados a cada categoría. Por lo tanto, al ejecutar esta consulta, obtendremos una lista ordenada de categorías de productos junto con la cantidad de productos que tienen en cada una.

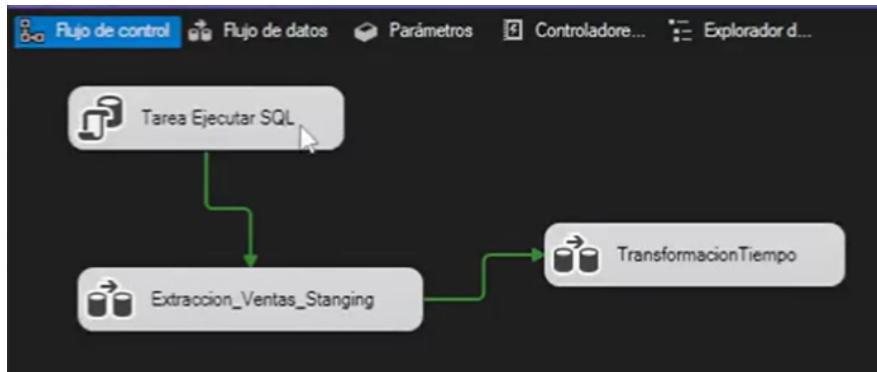
Descripción de la consulta para la creación de la tabla hechos ventas:

Esta consulta integra los primary key de cada dimensión y atributos propios de la tabla de hechos ventas, proporciona una visión detallada de las ventas de productos, incluyendo el nombre del producto, su categoría, el año de venta, la información del pedido, la cantidad vendida, el precio unitario y el monto total de la venta, ordenado por el monto total de la venta en orden descendente.

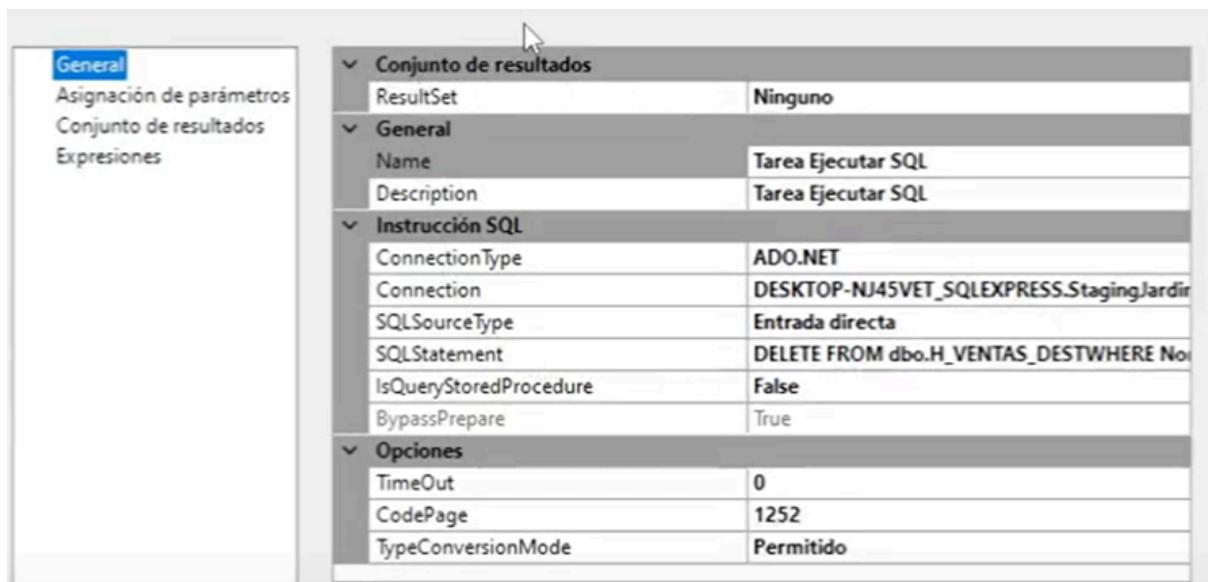
Después de hacer este análisis, ingresamos a Visual Studio Community, donde creamos el proyecto de integración entre las dos bases de datos y la conexión entre ambas. ETL jardinería sería la base de datos de origen y Staging jardinería el destino, donde se extraerán los datos al ejecutar el proyecto. Luego de hacer la extracción procedemos a hacer la transformación de los datos.

Transformación de dimensión tiempo.

En la parte de flujo de control, se crea una tarea de ejecución para evitar errores a la hora de ejecutar el proyecto.



Al abrir Tarea ejecutar SQL se pone la conexión que es de tipo ADO.NET con la conexión a la base de datos StagingJardineria, en la instrucción de SQLStatement se pone la consulta.



```
DELETE FROM dbo.H_VENTAS_DEST
WHERE Nombre_Producto IN (SELECT Nombre_Producto
FROM DIM_PRODUCTO_DEST
OR Categoría IN (SELECT Categoría FROM
DIM_CATEGORIA_DEST)
OR fecha_pedido IN (SELECT fecha_pedido FROM
DIM TIEMPO_DEST);
```

Consulta: que al iniciar el proyecto se eliminen los registros de la tabla venta, los datos que se encuentren en ella, y que al momento de iniciar la ejecución no registre errores..

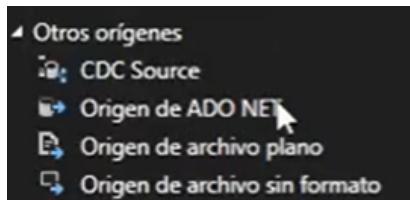
```
DELETE FROM DIM_PRODUCTO_DEST;
DELETE FROM DIM_CATEGORIA_DEST;
DELETE FROM DIM_TIEMPO_DEST;
```

Elimina los datos de la tabla producto, categoría y tiempo, esto para que en cada ejecución se eliminan los datos que ya están y así poder extraerlos de nuevo.

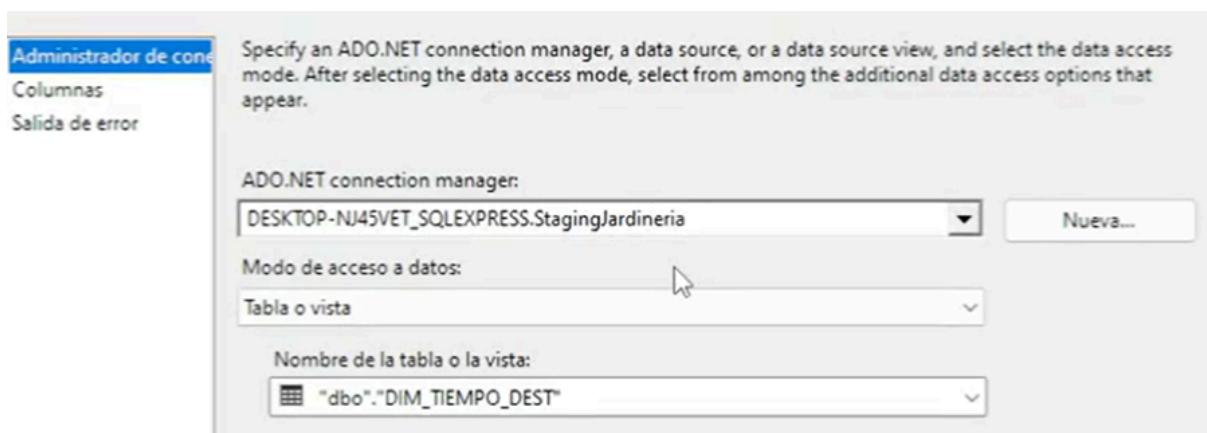


En la parte de Extracción_Ventas_Stangin, se encuentran la dimensión categoría, producto, tiempo y la tabla ventas.

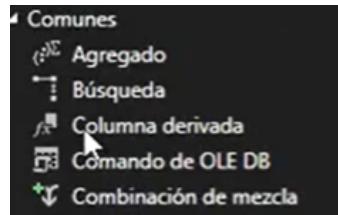
En la transformación tiempo, se encuentra la transformación de la dimensión tiempo.



Desde el cuadro de herramientas se toma Origen de ADO NET



Se apunta a que base de datos se va a utilizar, en este caso solo será con StagingJardineria que fue la que se creó nueva con la extracción de los datos de las dimensiones y se localiza la tabla de tiempo de la StagingJardineria, que tiene por nombre: DIM_TIEMPO_DEST.



Luego se usa una función llamada: columnas derivadas

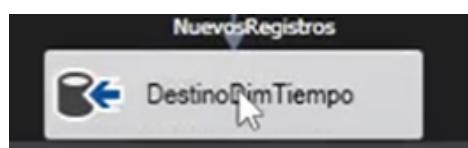
Derived Column Name	Derived Column	Expression	Data Type	Length
Anio	<add as new column>	YEAR(fecha_pedido)	entero de cuatro bytes ...	
Mes	<add as new column>	MONTH(fecha_pedido)	entero de cuatro bytes ...	
Dia	<add as new column>	DAY(fecha_pedido)	entero de cuatro bytes ...	
DiasemanaNum	<add as new column>	DATEPART("dw",fecha_pedido)	entero de cuatro bytes ...	
Trimestre	<add as new column>	DATEPART("quarter",fecha_pedido)	entero de cuatro bytes ...	
DiaAño	<add as new column>	DATEPART("dy",fecha_pedido)	entero de cuatro bytes ...	

En la parte donde indica columnas, se registra la columna que hay de la base de datos de la tabla DIM_TIEMPO_DEST, en este caso hay solo una llamada: fecha_pedido.

Por lo que es necesario agregar otras columnas, teniendo en cuenta los datos más detallados que quisiéramos validar.

Derived Column Name	Derived Column	Expression	Data Type
Anio	<add as new column>	YEAR([fecha_pedido]) R(fecha_pedido)	entero de cuatro bytes ...
Mes	<add as new column>	MONTH(fecha_pedido)	entero de cuatro bytes ...
Dia	<add as new column>	DAY(fecha_pedido)	entero de cuatro bytes ...
DiasemanaNum	<add as new column>	DATEPART("dw",fecha_pedido)	entero de cuatro bytes ...
Trimestre	<add as new column>	DATEPART("quarter",fecha_pedido)	entero de cuatro bytes ...
DiaAño	<add as new column>	DATEPART("dy",fecha_pedido)	entero de cuatro bytes ...

Al año se le la fecha_pedido y debe mostrar el año solamente de esa fecha y así para cada una de las columnas.



Specify a connection manager, data source, or data source view, and select the table or the view into which the data is copied. Click New to create a new table or view.

Connection manager:

DESKTOP-NJ45VET_SQLEXPRESS.StagingJardineria

Use a table or view:

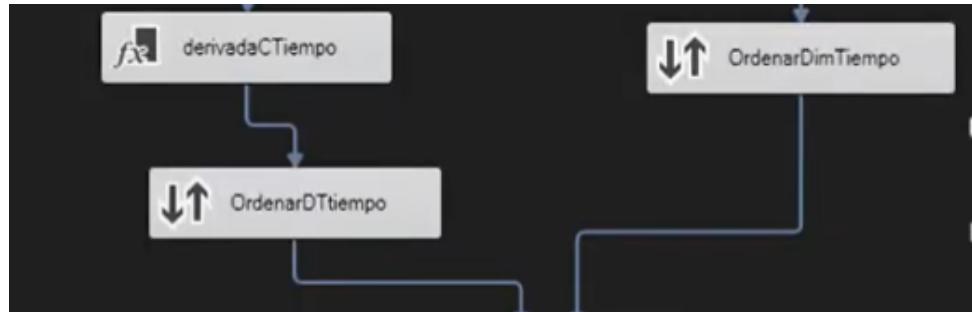
"DimTiempo"

Use Bulk Insert when possible

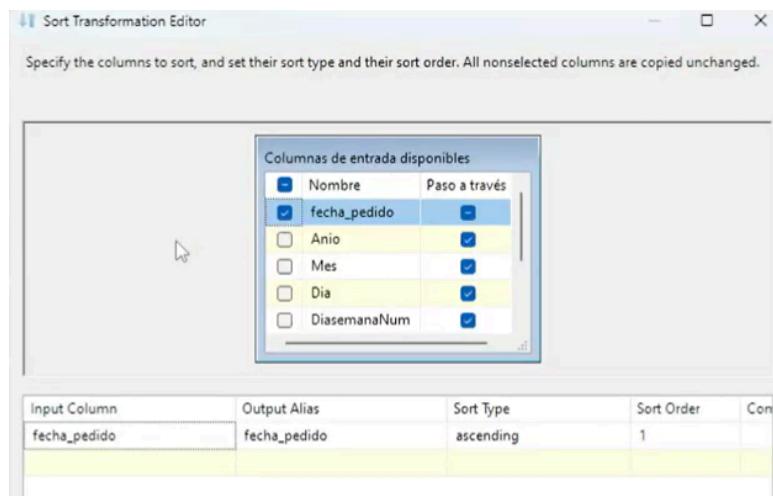
Luego de tener la configuración de todas las columnas, vamos a la parte de DestinoDimTiempo, en el cual se apunta de nuevo a la base de datos StagingJardineria, para elegir la tabla damos clic en New y se crea la siguiente tabla.

```
CREATE TABLE “DestinoDimTiempo” (
    “fecha_pedido” date,
    “Anio” int,
    “Mes” int,
    “Dia” int,
    “DiasemanaNum” int,
    “Trimestre” int,
    “DiaAño” int,
    “fecha_pedidoDimTiempo” date
)
```

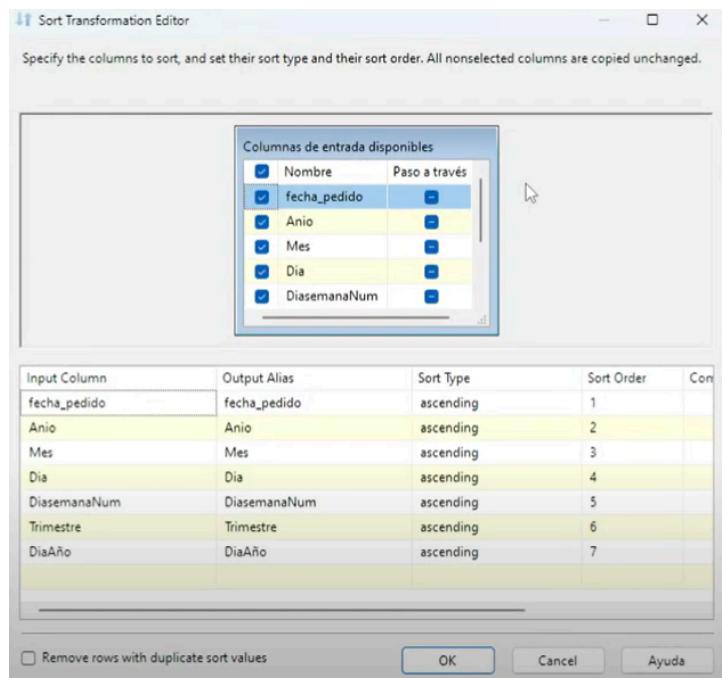
Luego utilizamos la herramienta ordenar, ponemos dos; uno con el nombre “OrdenarDTiempo” y el otro “OrdenarDimTiempo”.



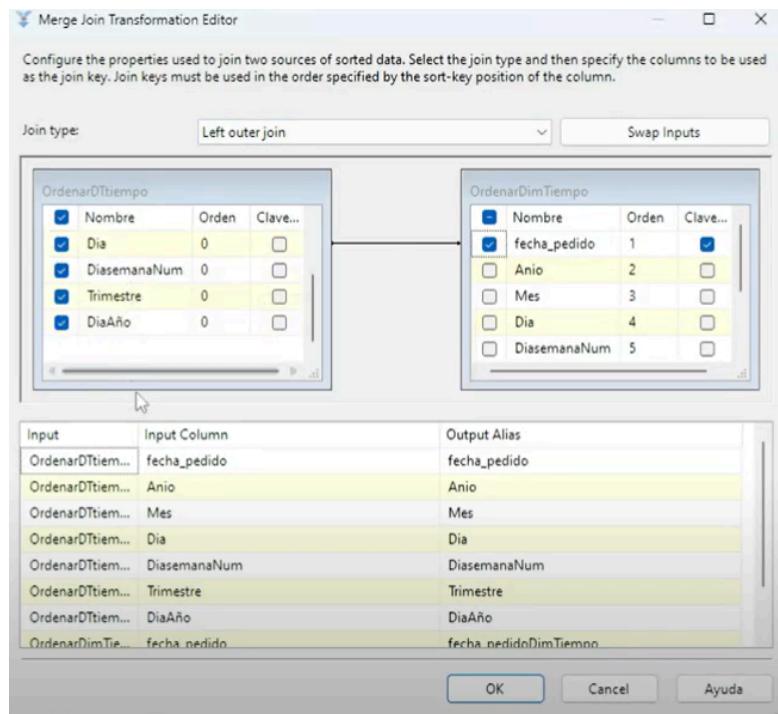
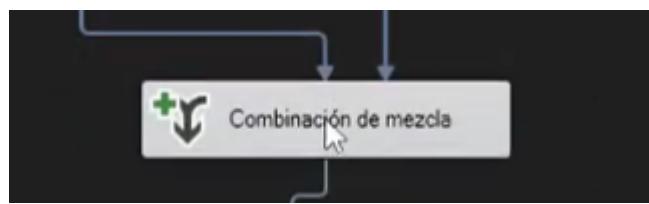
En la función de “OrdenarDTiempo” escogemos qué datos nos va a traer (fecha_pedido)



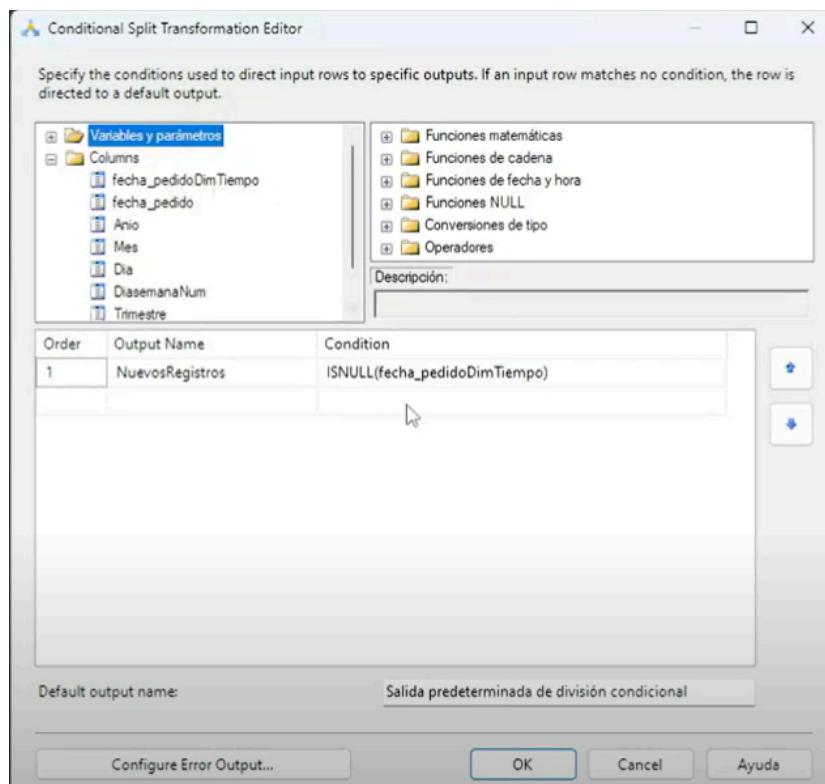
y en la función de “OrdenarDimTiempo” escogemos que nos traiga todo (fecha_pedido, Anio, Mes, Dia, DiasemanaNum, Trimestre, DiaAño).



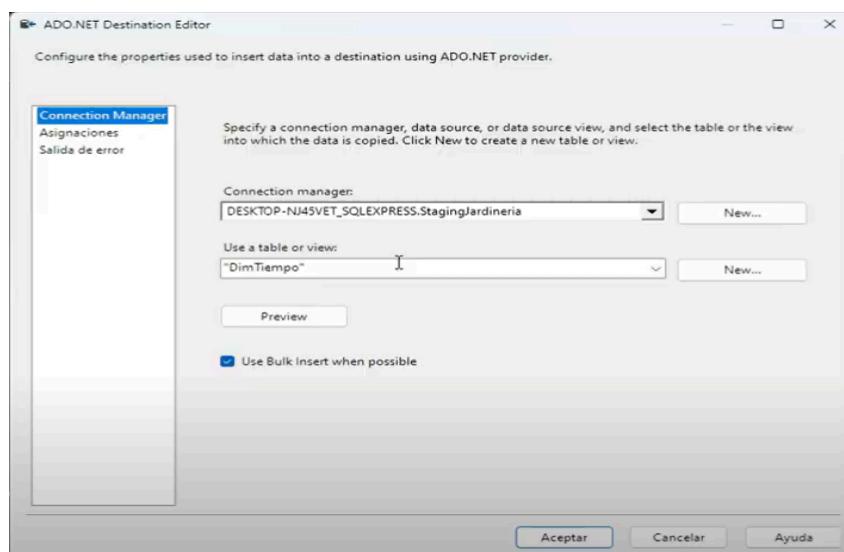
Cuando tenemos bien configurada la herramienta de ordenar utilizaremos la herramienta “Combinación de mezcla”, ahí uniremos “OrdenarDTiempo” y “OrdenarDimTiempo; esta herramienta nos muestra como la unión de las dos tablas con sus respectivos datos.



Luego necesitaremos la herramienta División condicional, aquí pondremos una condición, que nos servirá para cuando agreguemos nuevos datos en las bases de datos “ETLJardinería”, es decir si hay nuevos datos en estas bases de datos, se debe agregar y deben aparecer como NuevosRegistros; pero esta condición solo aplica si hay nuevos datos de la fecha_pedido. Entonces se realiza la configuración de la condición, indicando cómo se mostrarán los registros (NuevosRegistros) y ISNULL (si no hay nuevos datos no los trae y si hay nuevos los trae) y se escoge la columna que de donde se traerá los datos.



Por último utilizamos la herramienta destino, en donde traemos la dimensión que necesitamos y así haga la transformación correspondiente, primero realizamos la configuración; escogemos la bases de datos donde se va a conectar “StagingJardineria” y traemos la dimensión tiempo “DimTiempo”.



Cuando lo ejecutamos vemos que en las bases de datos StagingJardineria, en las tablas aparece dbo.DimTiempo, que esto muestra la transformación de la tabla de la dimensión tiempo, muestra los datos de fecha_pedido y los nuevos datos que nosotros queremos (fecha_pedido, Anio, Mes, Dia, DiasemanaNum, Trimestre, DiaAño).

The screenshot shows the SQL Server Management Studio interface. On the left, the Object Explorer displays the database structure for 'DESKTOP-NJ45VET\SQLEXPRESS (SQL)'. It lists several databases, including 'ETLjardineria', 'jardineria', and 'StagingJardineria'. Under 'StagingJardineria', there are tables like 'dbo.DIM_CATEGORIA_DEST', 'dbo.DIM_PRODUCTO_DEST', 'dbo.DIM_TIEMPO_DEST', and 'dbo.fecha_pedido'. A table named 'dbo.DimTiempo' is highlighted in yellow. On the right, a query window titled 'SQLQuery1.sql - DE...-NJ45VET\User (67)' contains the following SQL code:

```
SELECT TOP (1000) [fecha_pedido]
      ,[Anio]
      ,[Mes]
      ,[Dia]
      ,[DiasemanaNum]
      ,[Trimestre]
      ,[DiaAño]
  FROM [StagingJardineria].[dbo].[DimTiempo]
```

Below the code, the 'Results' tab is selected, showing a grid of data. The columns are labeled: fecha_pedido, Anio, Mes, Dia, DiasemanaNum, Trimestre, and DiaAño. The data starts with the date 2006-01-17 and continues sequentially.

	fecha_pedido	Anio	Mes	Dia	DiasemanaNum	Trimestre	DiaAño
1	2006-01-17	2006	1	17	2	1	17
2	2006-05-25	2006	5	25	4	2	145
3	2007-01-07	2007	1	7	7	1	7
4	2007-03-19	2007	3	19	1	1	78
5	2007-05-20	2007	5	20	7	2	140
6	2007-06-20	2007	6	20	3	2	171
7	2007-10-05	2007	10	5	5	4	278
8	2007-10-23	2007	10	23	2	4	296
9	2008-01-04	2008	1	4	5	1	4
10	2008-03-05	2008	3	5	3	1	65
11	2008-03-10	2008	3	10	1	1	70

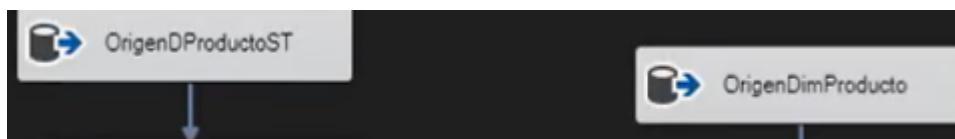
Para empezar con las demás transformación, primero debemos ir a la parte de flujo de control y crear una nueva consulta.

The screenshot shows the 'Editor de la tarea Ejecutar SQL' dialog box. In the center, there is a large text area containing an SQL script. The script starts with a DELETE statement from the 'dbo.H_VENTAS_DEST' table, filtering by 'Nombre_Producto' and 'Nombre_Proveedor'. It then performs three DELETE operations on the 'DIM_PRODUCTO_DEST', 'DIM_CATEGORIA_DEST', and 'DIM_TIEMPO_DEST' tables. Below the script, there are tabs for 'General', 'Conjunto de resultados', 'Instrucción SQL', and 'Opciones'. The 'General' tab shows the connection is set to 'ADO.NET' and the connection is 'DESKTOP-NJ45VET_SQLEXPRESS.StagingJardineria'. The 'Instrucción SQL' tab contains the full SQL script. The 'Opciones' tab has settings for 'Name' and 'BypassPrepare'. At the bottom, there are buttons for 'Examinar...', 'Generar consulta...', 'Analizar consulta', 'Aceptar', 'Cancelar', and 'Ayuda'.

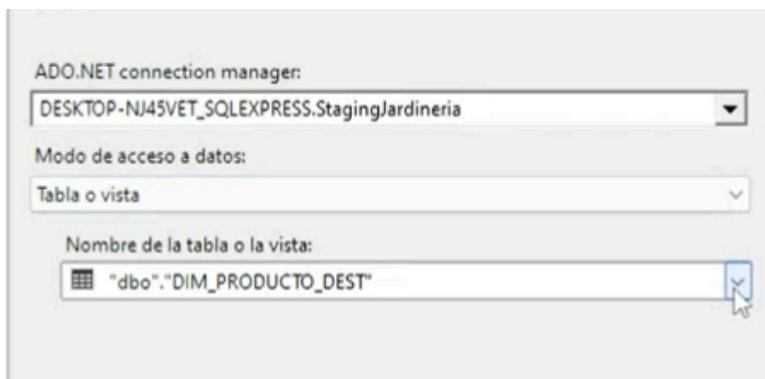
```
DELETE FROM dbo.H_VENTAS_DEST  
WHERE Nombre_Producto IN (SELECT Nombre_Producto  
FROM DIM_PRODUCTO_DEST)  
    OR Categoría IN (SELECT Categoría FROM  
DIM_CATEGORIA_DEST)  
    OR fecha_pedido IN (SELECT fecha_pedido FROM  
DIM TIEMPO_DEST);  
  
DELETE FROM DIM_PRODUCTO_DEST;  
DELETE FROM DIM_CATEGORIA_DEST;  
DELETE FROM DIM TIEMPO_DEST;  
  
DELETE FROM DimCategoria  
DELETE FROM DimProducto  
DELETE FROM DimTiempo  
DELETE FROM FactsVentas
```

Transformación dimensión productos

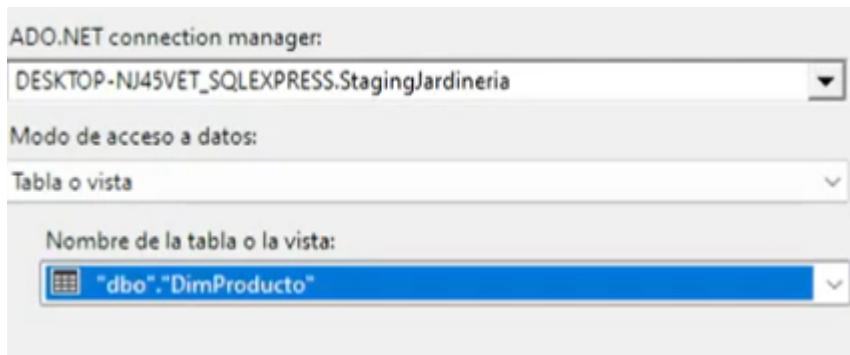
En la sección de flujo de datos, escogemos la herramienta origen de ADO NET y se le asigna a uno el nombre de “Origen DProductoST” y al otro “OrigenDimProducto”.



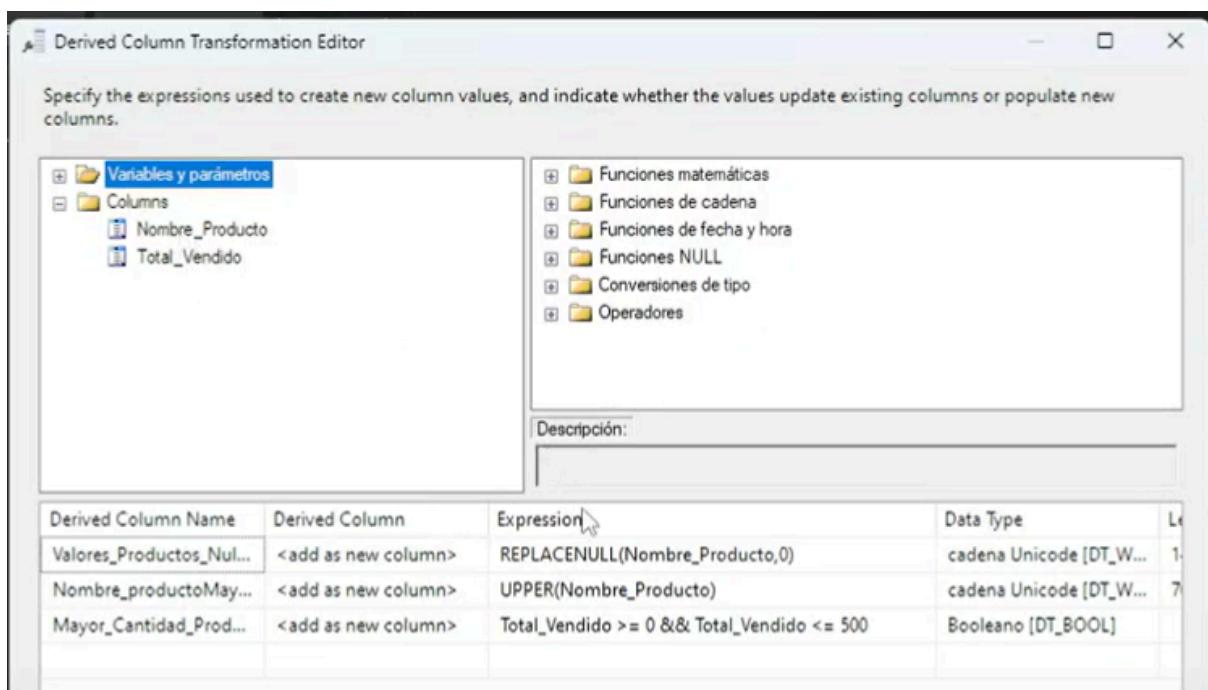
Realizamos la configuración escogemos desde qué bases de datos nos vamos a conectar y donde nos vamos a conectar, en este caso sería “DIM_PRODUCTO_DEST”; para el primero



y para el segundo es “DimProducto”



Después de verificar que no hay errores en la conexión, utilizamos la herramienta columna derivada, le asignamos el nombre de “derivadaCProducto”. Cuando configuramos la funcionalidad, le indicamos que pase los datos Null a 0, que convierta los nombre de los productos en mayúscula y nos traiga los datos de la columna Total_Vendido mayor a 0.



Después utilizamos la herramienta de ordenar y le asignamos el nombre “OrdenarTBProducto” y “OrdenarDimProducto”



Lo configuramos “OrdenarTBProducto” indicando la columna que nos va a traer (Nombre_Producto),

Columnas de entrada disponibles

	Nombre	Paso a través
<input checked="" type="checkbox"/>	Nombre	
<input checked="" type="checkbox"/>	Nombre_Producto	
<input type="checkbox"/>	Total_Vendido	Nombre de columna: Nombre_Producto Tipo de datos: cadena Unicode [DT_WSTR] Longitud: 70 Escala: 0 Precisión: 0 Componente de origen: OrigenDProductoST
<input type="checkbox"/>	Valores_Produ	
<input type="checkbox"/>	Nombre_prod	

Input Column Output Alias Sort Type Sort Order Com

Nombre_Producto	Nombre_Producto	ascending	1	
-----------------	-----------------	-----------	---	--

y para “OrdenarDimProducto” escogemos todas las columnas.

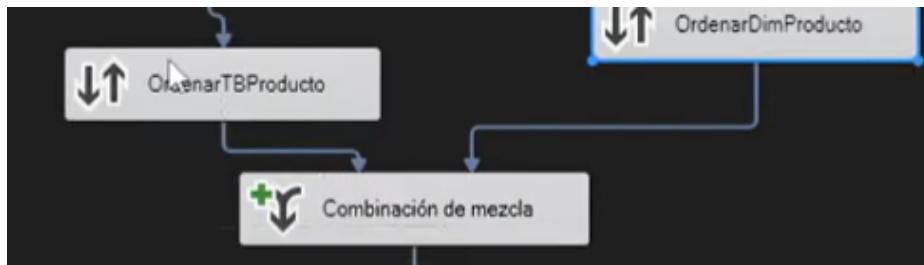
Columnas de entrada disponibles

	Nombre	Paso a través
<input checked="" type="checkbox"/>	Nombre	
<input checked="" type="checkbox"/>	Nombre_Producto	
<input checked="" type="checkbox"/>	Total_Vendido	
<input checked="" type="checkbox"/>	Valores_Productos_Nulos	
<input checked="" type="checkbox"/>	Nombre_productoMayus	

Input Column Output Alias Sort Type Sort Order Com

Nombre_Producto	Nombre_Producto	ascending	1	
Total_Vendido	Total_Vendido	ascending	2	
Valores_Productos_Nulos	Valores_Productos_Nulos	ascending	3	
Nombre_productoMayus	Nombre_productoMayus	ascending	4	
Mayor_Cantidad_Productos	Mayor_Cantidad_Productos	ascending	5	

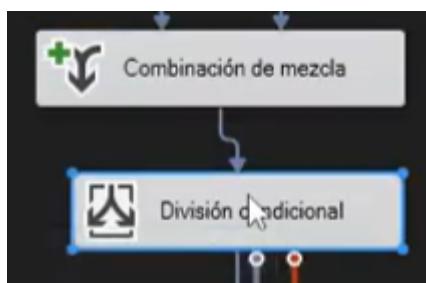
Luego realizamos la combinación de mezcla, primero conectando “OrdenarTBProducto” y luego “OrdenarDimProducto”



Configuramos por cuál lado se va a conectar, escogiendo el de la izquierda. También seleccionamos los datos que vamos traer de cada tabla

Input	Input Column	Output Alias
OrdenarTBPro...	Nombre_Producto	Nombre_Producto
OrdenarTBPro...	Total_Vendido	Total_Vendido
OrdenarTBPro...	Valores_Productos_Nulos	Valores_Productos_Nulos
OrdenarTBPro...	Nombre_productoMayus	Nombre_productoMayus
OrdenarTBPro...	Mayor_Cantidad_Productos	Mayor_Cantidad_Productos
OrdenarDimPr...	Nombre_Producto	Nombre_ProductoDim

Luego utilizamos la herramienta división condicional y realizamos su respectiva conexión



Realizamos una nueva condición, indicando que los registros nuevos aparezcan con el nombre “RegistrosNuevos” y que no nos mande si hay un registro null

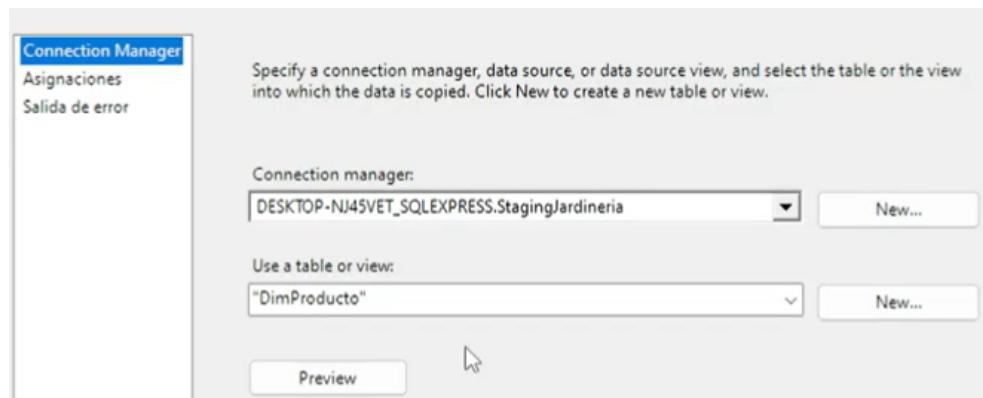
Specify the conditions used to direct input rows to specific outputs. If an input row matches no condition, the row is directed to a default output.

Order	Output Name	Condition
1	RegistrosNuevos	ISNULL(Nombre_ProductoDim)

Luego conectamos la división con el destino y nos fijamos que aparece la condición en medio de la conexión

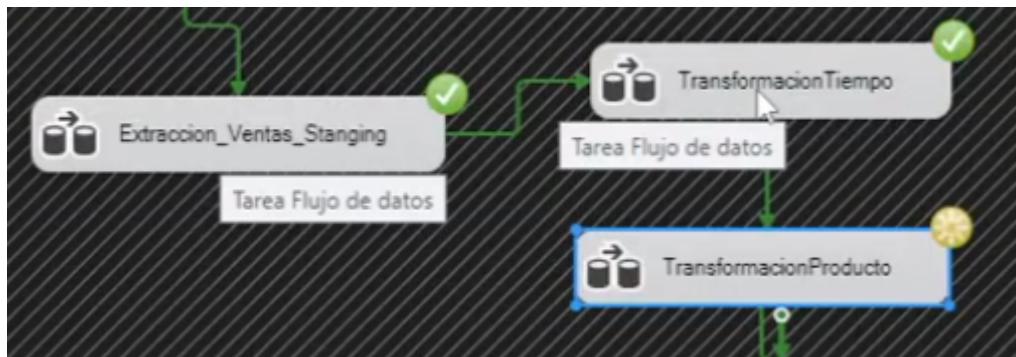


Luego nos muestra cual es la base de datos que nos vamos a conectar y a que dimensión haremos la transformación.

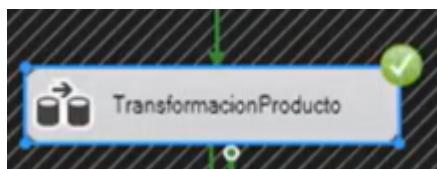




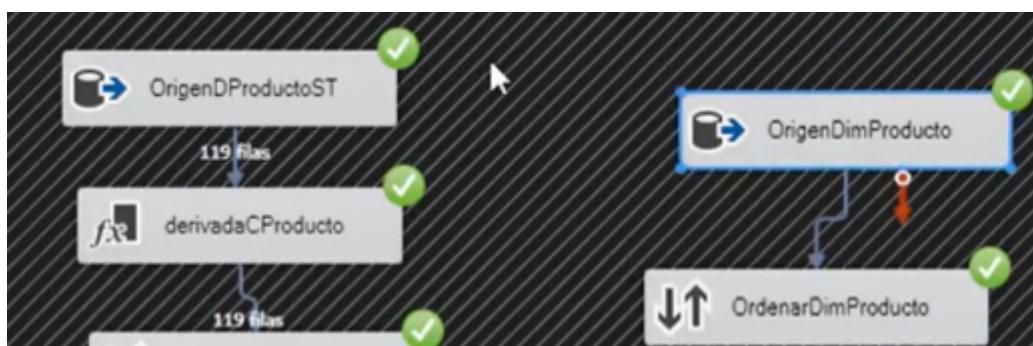
Realizamos la ejecución y se observamos nos va indicando en qué proceso se encuentra, desde que realiza la extracción de los datos de la ventas, luego hace la transformación de tiempo y pasa realiza la transformación de productos

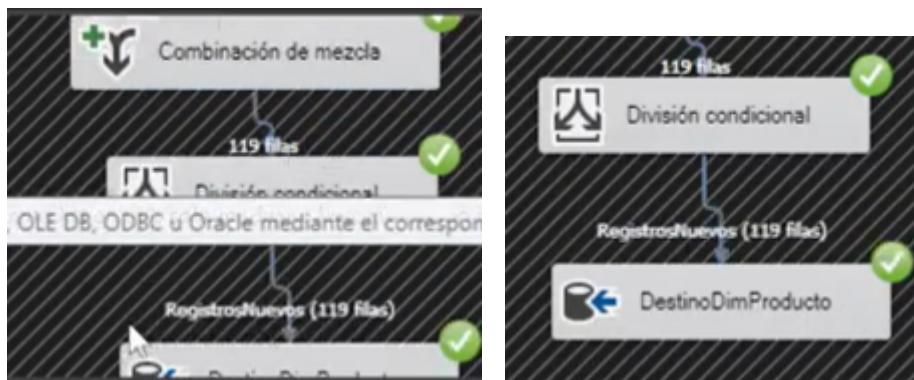


Cuando observamos que la transformación producto tiene un chulo verde indica que ya está listo



y podemos observar en cada uno de los procesos cuántos datos ejecuto.





Luego observamos en la base de datos StangingJardineria, en la tabla DimProducto, los nuevos datos de la transformación (Valores_Productos_Nulos, Nombre_productosMayus, Mayor_Cantidad_Productos)

```

SELECT TOP (1000) [Nombre_Producto]
      ,[Total_Vendido]
      ,[Valores_Productos_Nulos]
      ,[Nombre_productoMayus]
      ,[Mayor_Cantidad_Productos]
  FROM [StagingJardineria].[dbo].[DimProducto]
  
```

	Nombre_Producto	Total_Vendido	Valores_Productos_Nulos	Nombre_productoMayus	Mayor_Cantidad_Productos
1	Acer Negundo	4	Acer Negundo	ACER NEGUNDO	1
2	Acer platanoides	33	Acer platanoides	ACER PLATANOIDES	1
3	Acer Pseudoplatanus	262	Acer Pseudoplatanus	ACER PSEUDOPLATANUS	1
4	Ajedrea	135	Ajedrea	AJEDREA	1
5	Albaricoquero	62	Albaricoquero	ALBARICOQUERO	1
6	Archontophoenix Cunninghamiana	50	Archontophoenix Cunninghamiana	ARCHONTOPHOENIX CUNNINGHAMIANA	1
7	Azadon	220	Azadon	AZADON	1
8	Beucamea Recurvata	150	Beucamea Recurvata	BEUCARNEA RECURVATA	1
9	Bismarckia Nobilis	35	Bismarckia Nobilis	BISMARCKIA NOBILIS	1
10	Bougamvillea roja, naranja	18	Bougamvillea roja, naranja	BOUGAMVILLEA ROJA, NARANJA	1
11	Bougamvillea Sanderiana Tutor	23	Bougamvillea Sanderiana Tutor	BOUGAMVILLEA SANDERIANA TUTOR	1

Transformación de la dimensión categoría.

Para esta transformación utilizamos la misma herramientas que las anteriores transformaciones, utilizamos Origen de ADO NET, y se le asigna el nombre una de “OrigenDCategoríaST” y “OrigenDimCategoria”, contiene el mismo origen de conexión solo cambia la tabla que es DIM_CATEGORIA_DEST.

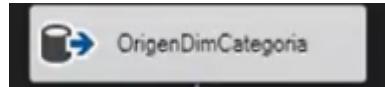


Luego configuramos la derivada, asignamos el nombre derivadaCCategoría, y agregamos las nuevas columnas que va a tener la transformación, le asignamos parámetros, de cómo queremos ver los datos.

Derived Column Name	Derived Column	Expression	Data Type	Length
Categoría_Mayusculas	<add as new column>	UPPER(Categoría)	cadena Unicode [DT_WSTR]	50
Categoría_	<add as new column>	REPLACENULL(Categoría,"N/A")	cadena Unicode [DT_WSTR]	100
Cantidad_Productos_O...	<add as new column>	REPLACENULL(Cantidad_Productos,"N/A")	entero de cuatro bytes [DT_I4]	

Luego realizamos la configuración, para ordenar la categoría, le ponemos el nombre de “OrdenarTBCategoría” y escogemos qué columna utilizaremos (Categoría)

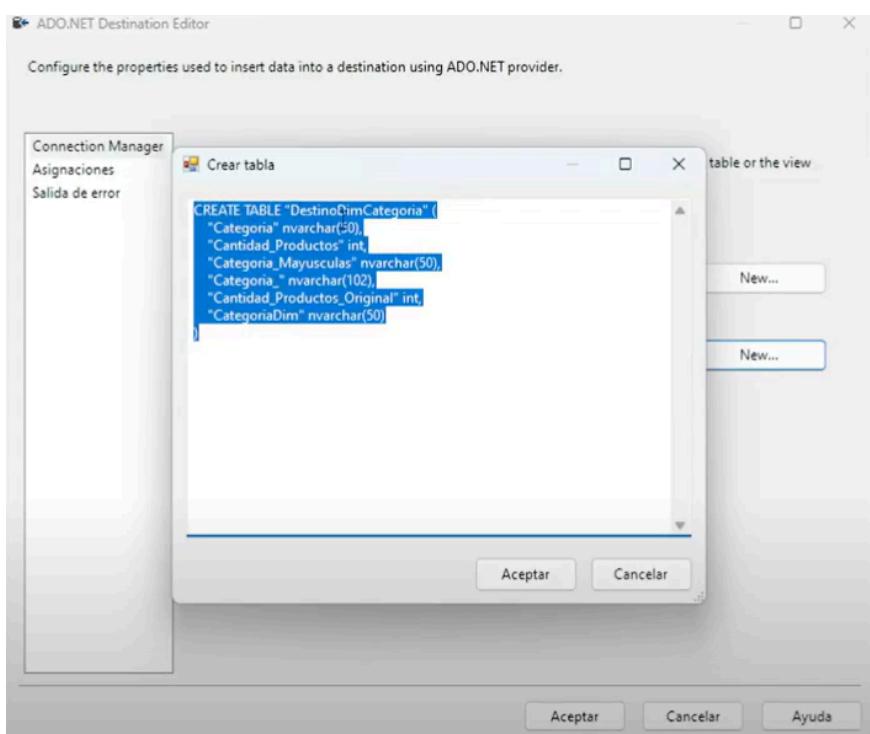
Input Column	Output Alias	Sort Type	Sort Order	Con
Categoría	Categoría	ascending	1	



Luego volvemos a crear un Origen, con el nombre **OrigenDimCategoria**, como anteriormente ya habíamos hecho el proceso para crear “DimCateogria” (conectamos la herramienta de destino nombra “DestinoDimCategoria” con la derivada),



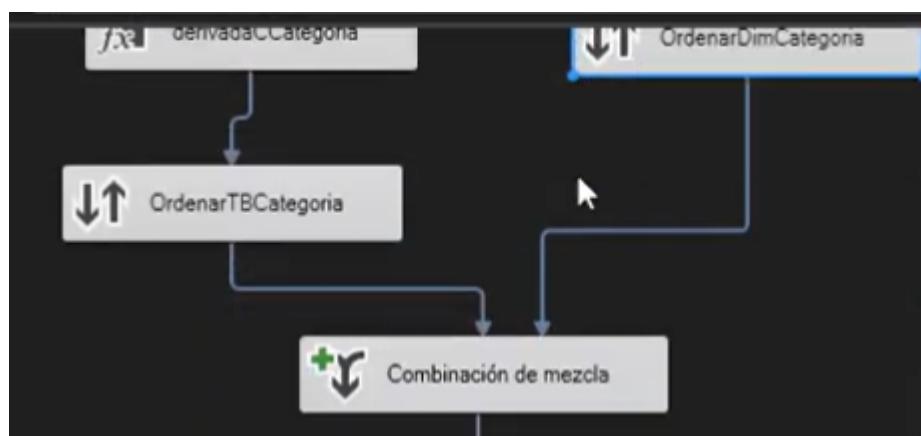
Cuando realizamos la configuración, escogemos la conexión de la bases de datos con la tabla



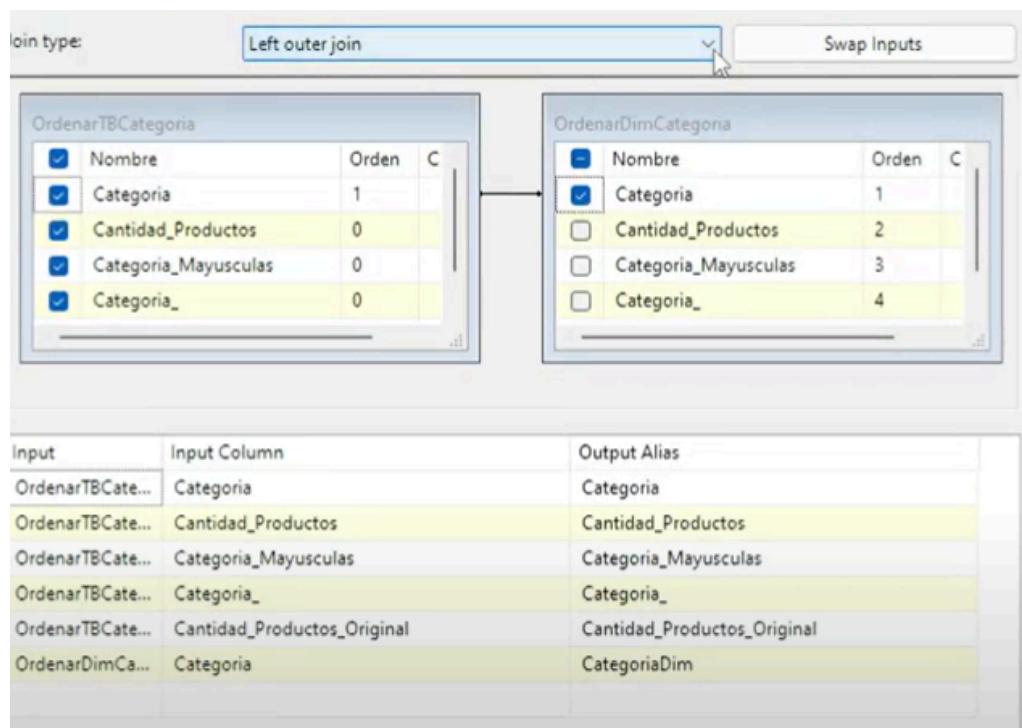
Luego nuevamente utilizamos la herramienta ordenar para crear “OrdenarDimCategoria” y escogemos todas las columnas que vamos a utilizar

Input Column	Output Alias	Sort Type	Sort Order	Conn.
Categoría	Categoría	ascending	1	
Cantidad_Productos	Cantidad_Productos	ascending	2	
Categoría_Mayusculas	Categoría_Mayusculas	ascending	3	
Categoría_	Categoría_	ascending	4	
Cantidad_Productos_Original	Cantidad_Productos_Original	ascending	5	

Luego seguimos realizando la conexión de las dos herramientas ordenar con la herramienta combinación de mezcla



Configuramos la combinación, indicando de dónde hacia dónde va a comenzar, en este caso la izquierda, y hacemos lo contrario de la dimensión anterior, de la tabla “OrdenarTBCatgoria” escogemos todas las columnas y de “OrdenarDimCategoria” escogemos la columna categoría



Continuamos poniendo la condición, utilizando la herramienta  **División condicional**, indicamos que utilizaremos la categoriaDim, como variable ISNULL para los NuevosRegistros.

Order	Output Name	Condition
1	NuevosRegistros	ISNULL(CategoríaDim)

y por último conectamos división condicional con DimesioDimCategoria, que como sabemos es donde se va depositar los datos que se va a extraer. Configuramos escogiendo la conexión de la base de datos y la tabla que se usará para la transformación.



into which the data is copied. Click New to create a new table or view.

Connection manager: DESKTOP-NJ4SVET_SQLEXPRESS.StagingJardineria

Use a table or view: "DimCategoria"

Use Bulk Insert when possible

Luego ejecutamos y miramos cómo realiza los pasos, y lo observamos en la base de datos StagingJardineria, en la tabla DimCategoria, los datos que tenía y los nuevos (Categoria_Mayuscula, Categoria_, Cantidad_Productos_Original)

```

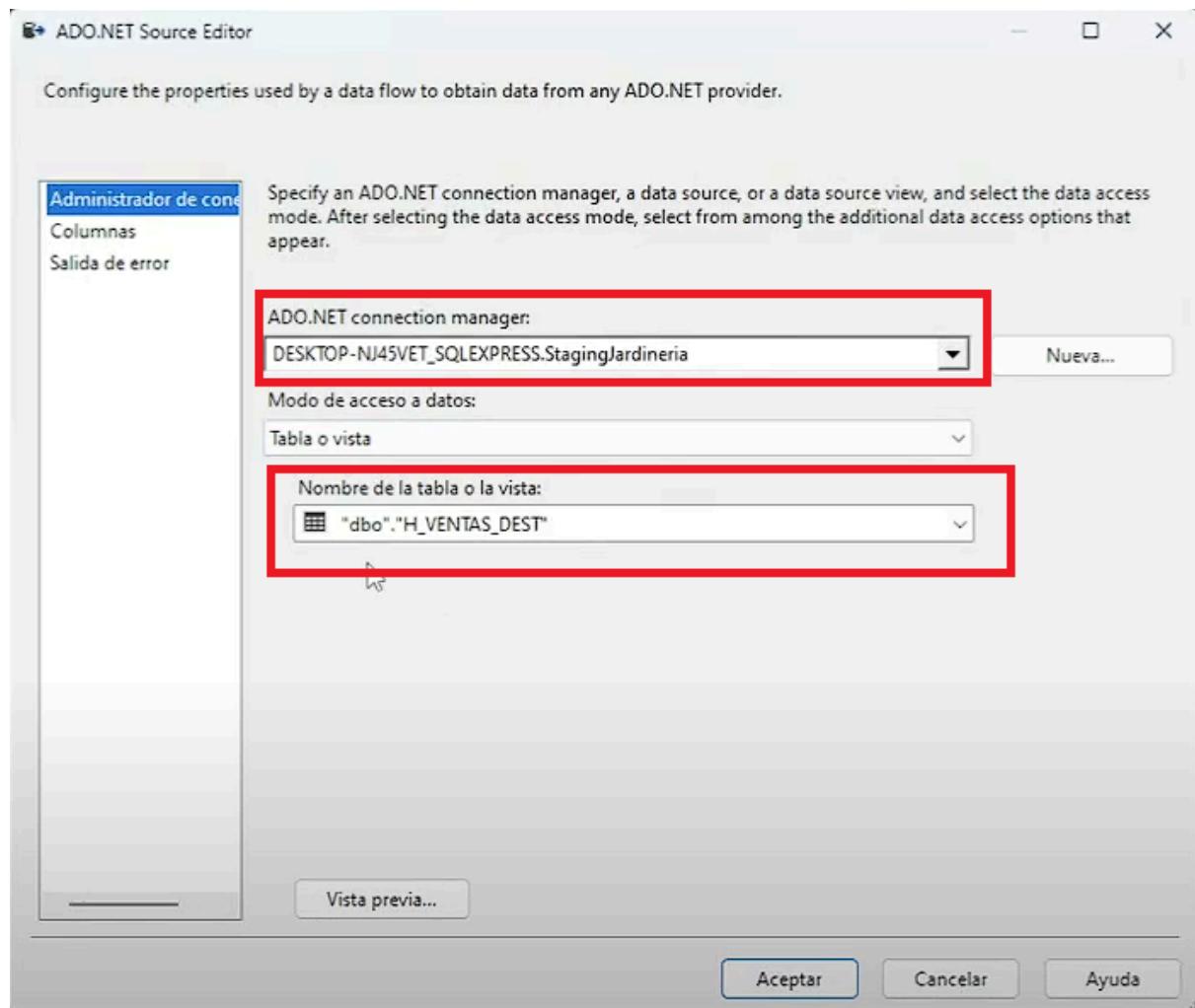
SELECT TOP (1000) [Categoria]
      ,[Cantidad_Productos]
      ,[Categoria_Mayusculas]
      ,[Categoria_]
      ,[Cantidad_Productos_Original]
  FROM [StagingJardineria].[dbo].[DimCategoria]

```

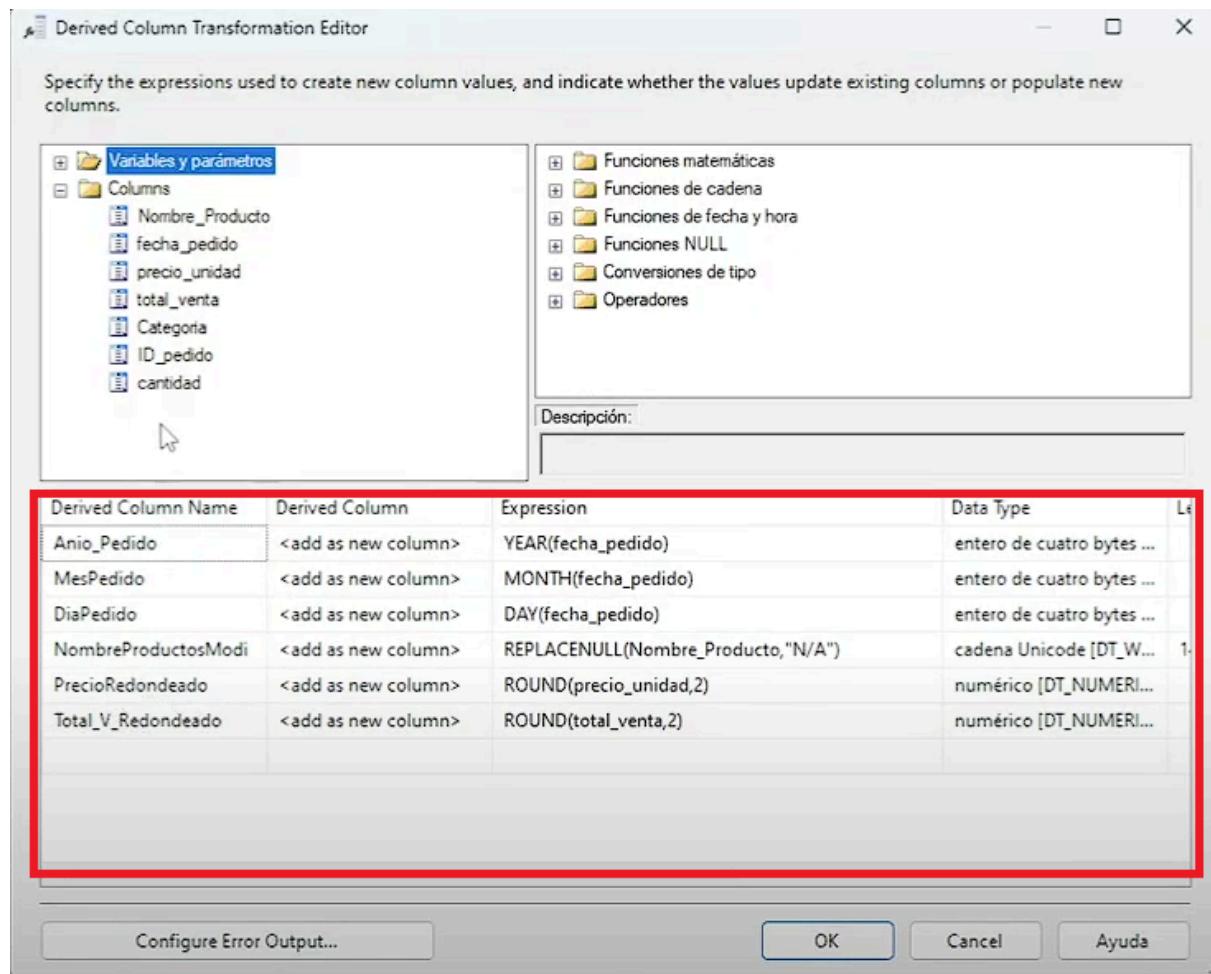
	Categoria	Cantidad_Productos	Categoria_Mayusculas	Categoria_	Cantidad_Productos_Original
1	Aromaticas	30	AROMATICAS	Aromaticas	30
2	Frutales	78	FRUTALES	Frutales	78
3	Herbaceas	4	HERBACEAS	Herbaceas	4
4	Herramientas	10	HERRAMIENTAS	Herramientas	10
5	Omamentales	154	ORNAMENTALES	Omamentales	154

Transformación dimensión hechos

Accedemos a los datos que emplearemos.

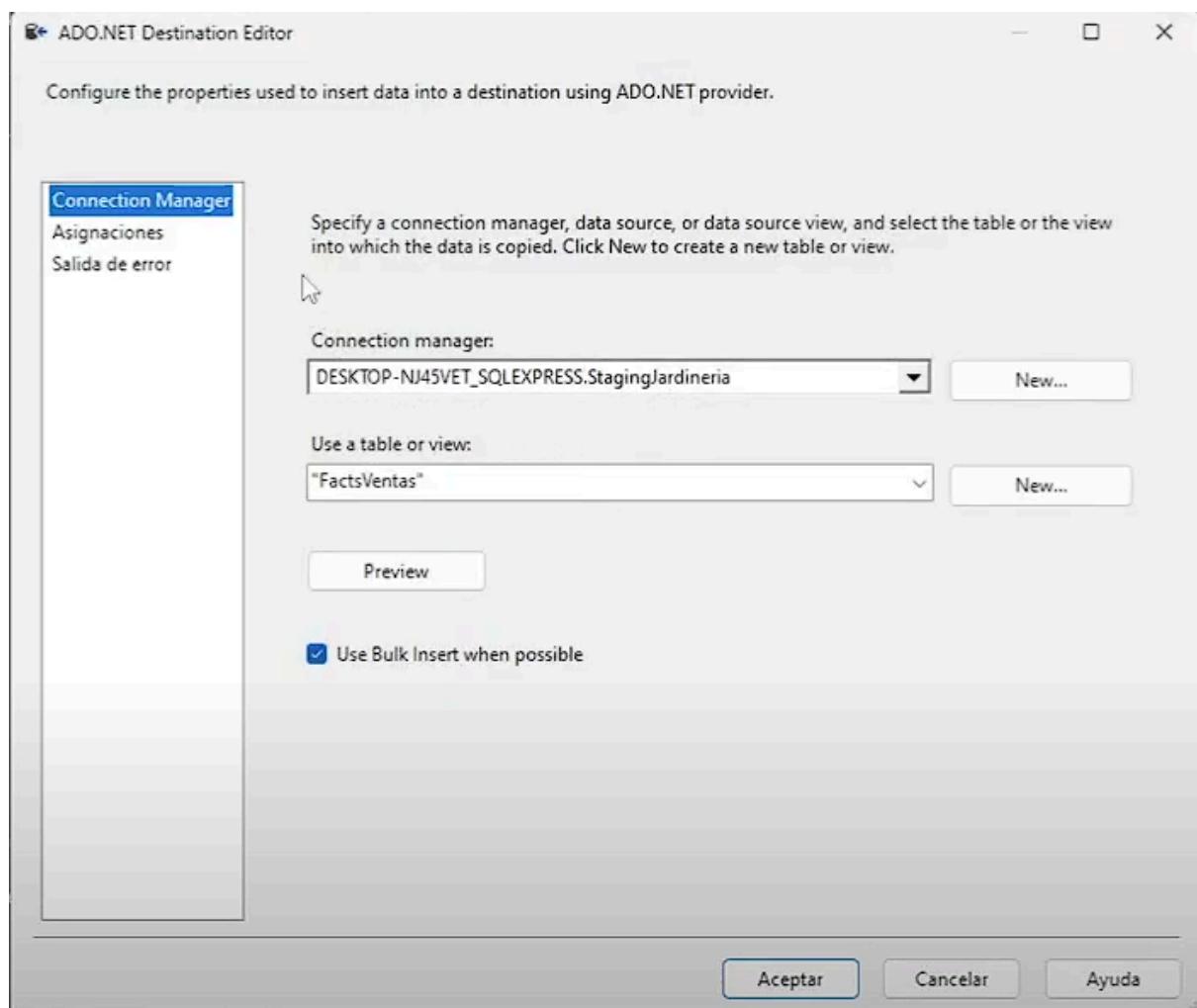


A continuación, realizamos las configuraciones que queremos para la transformación como se muestra en la siguiente imagen.

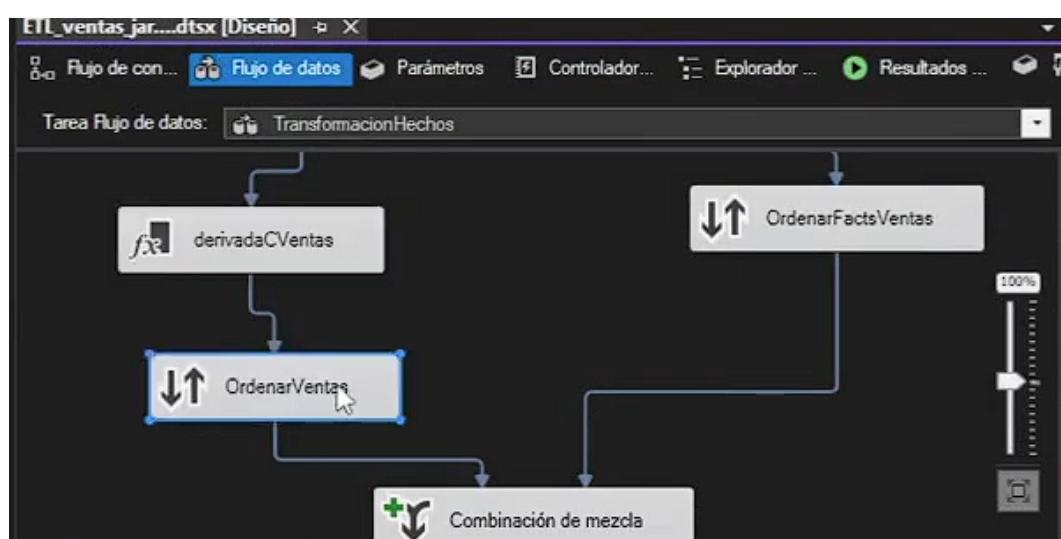


Procedemos a crear la tabla de destino donde tendremos todos los datos de la transformación. Realizamos el procedimiento ya antes mencionado el cual mostraremos en las siguientes imágenes.

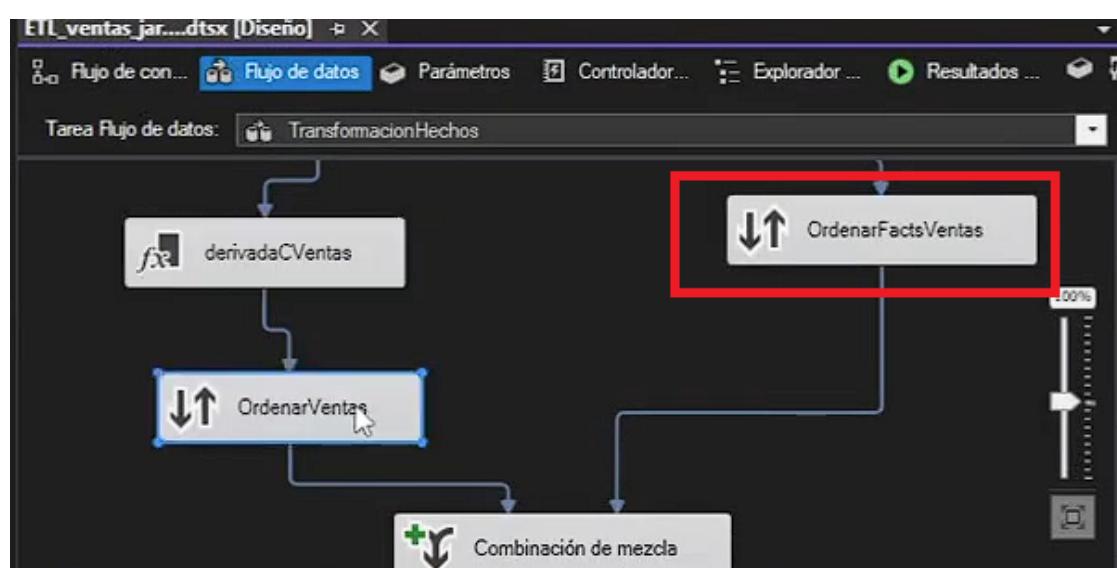
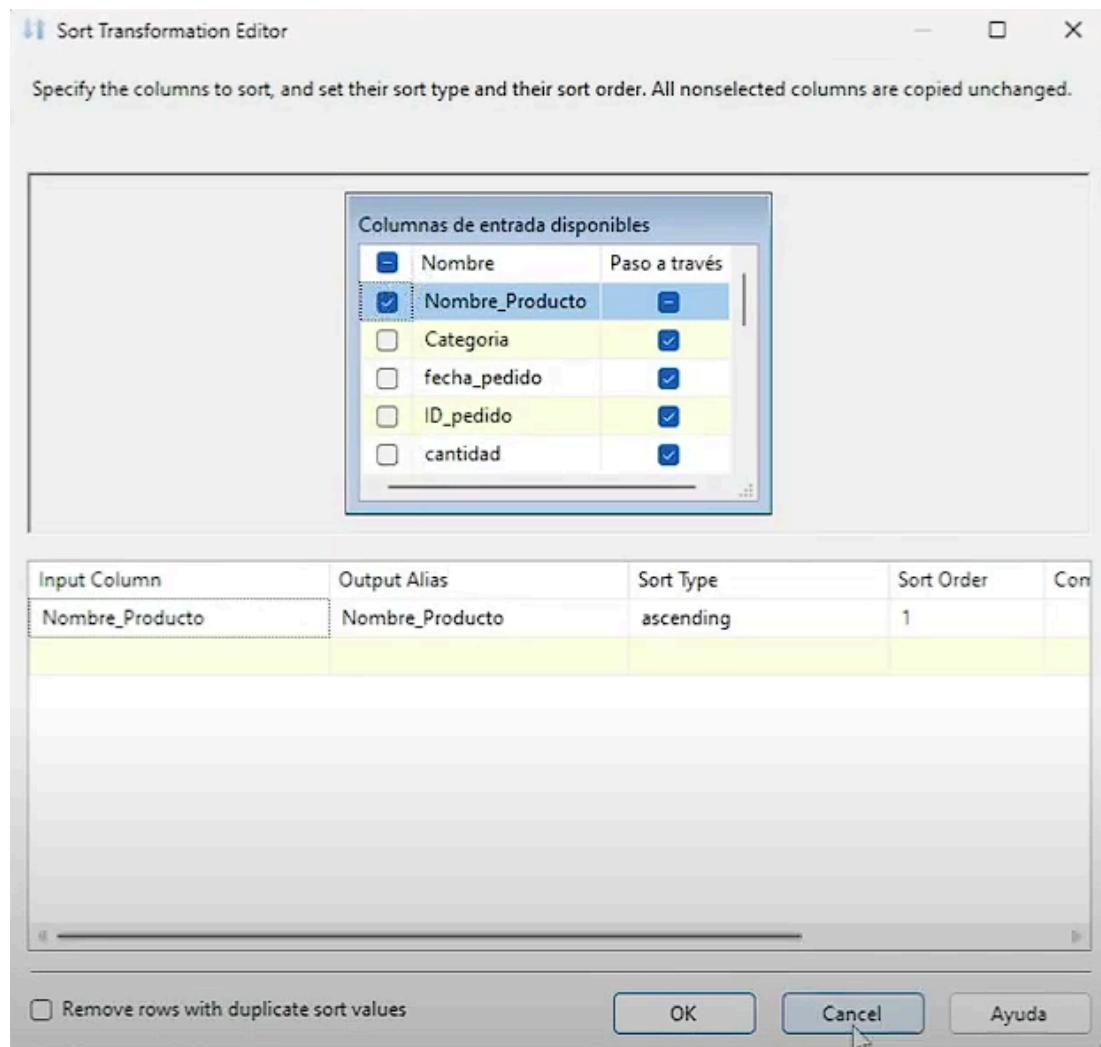
Accedemos a los datos



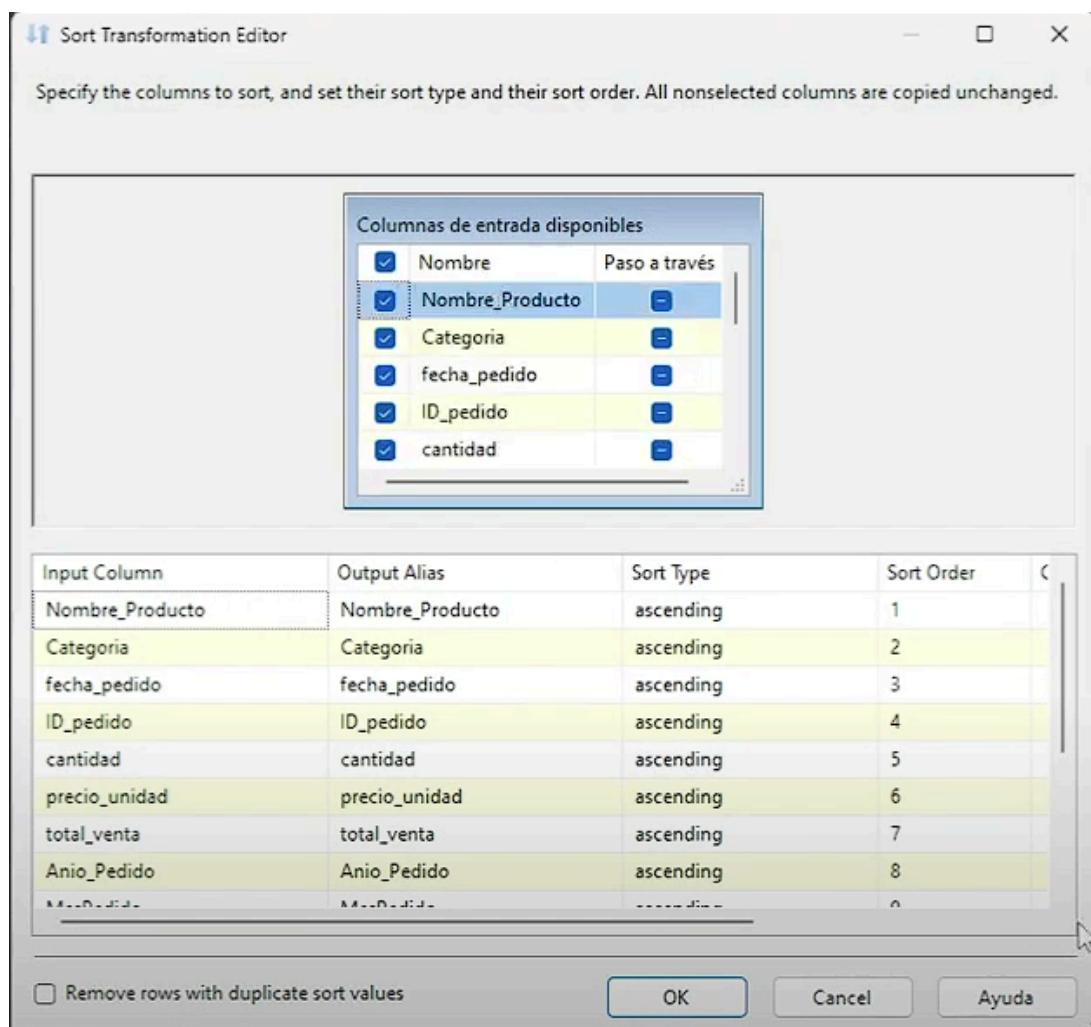
Ordenamos.



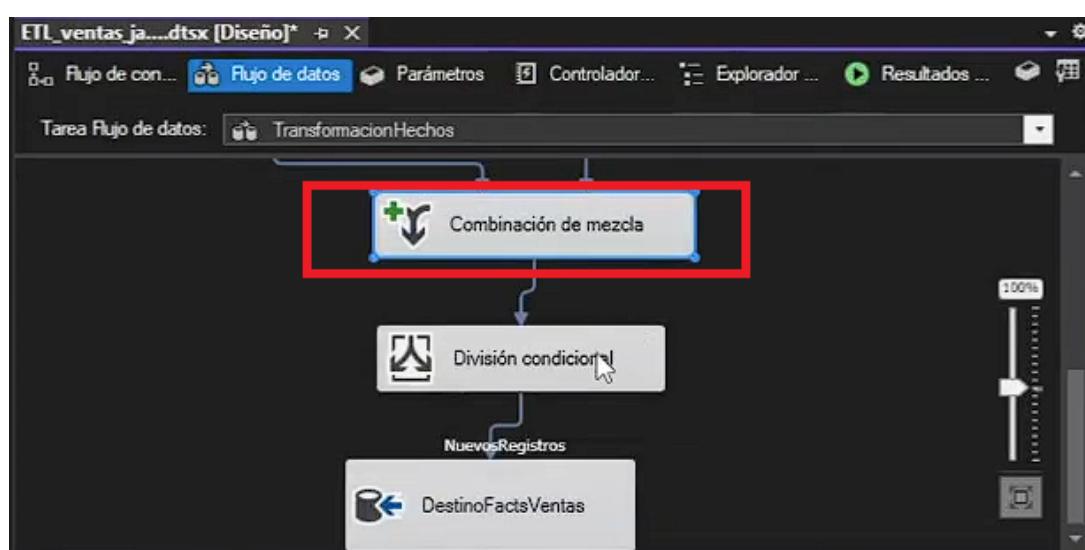
Seleccionamos Nombre_Producto.



Para esta opción seleccionamos todas las filas.



A continuación, configuraremos el Merge Join.



Merge Join Transformation Editor

Configure the properties used to join two sources of sorted data. Select the join type and then specify the columns to be used as the join key. Join keys must be used in the order specified by the sort-key position of the column.

Join type: **Left outer join** Swap Inputs

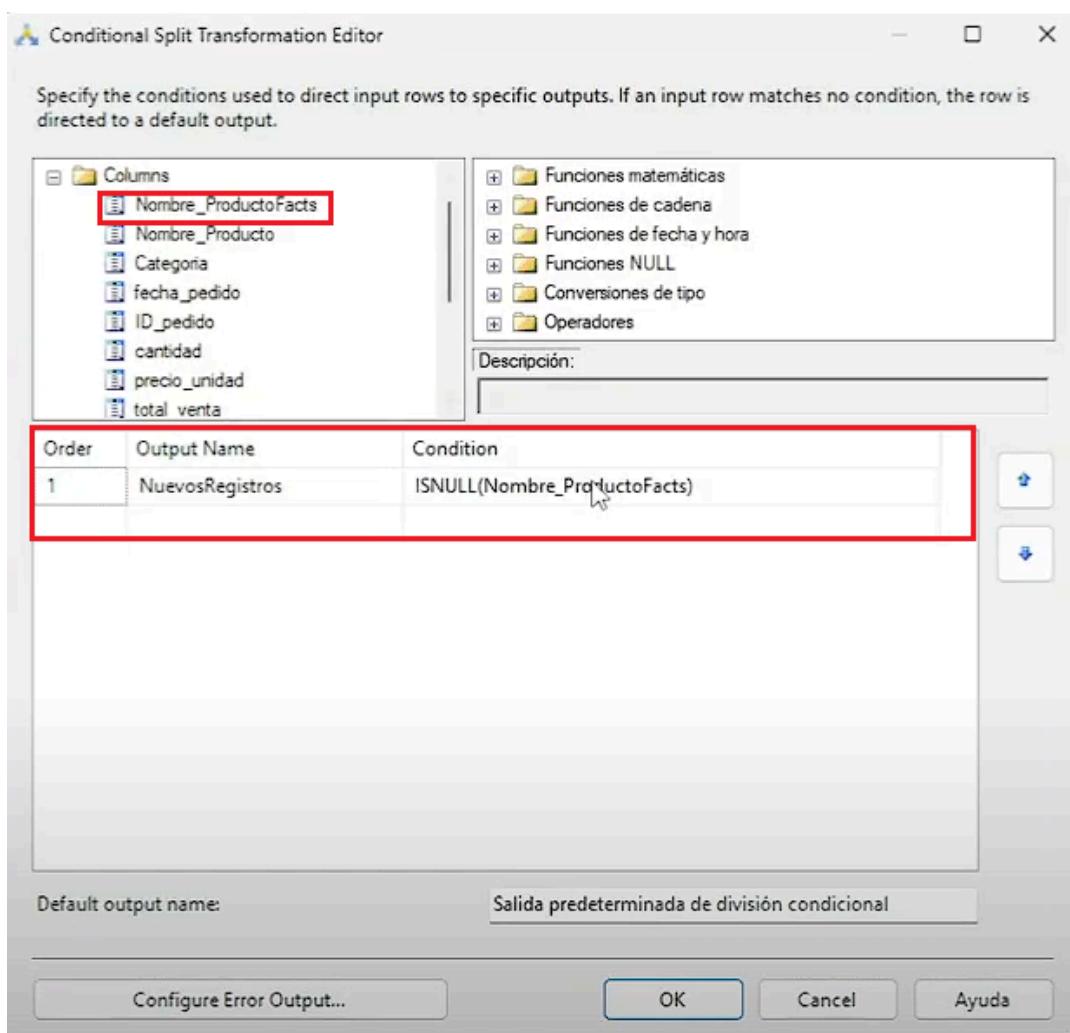
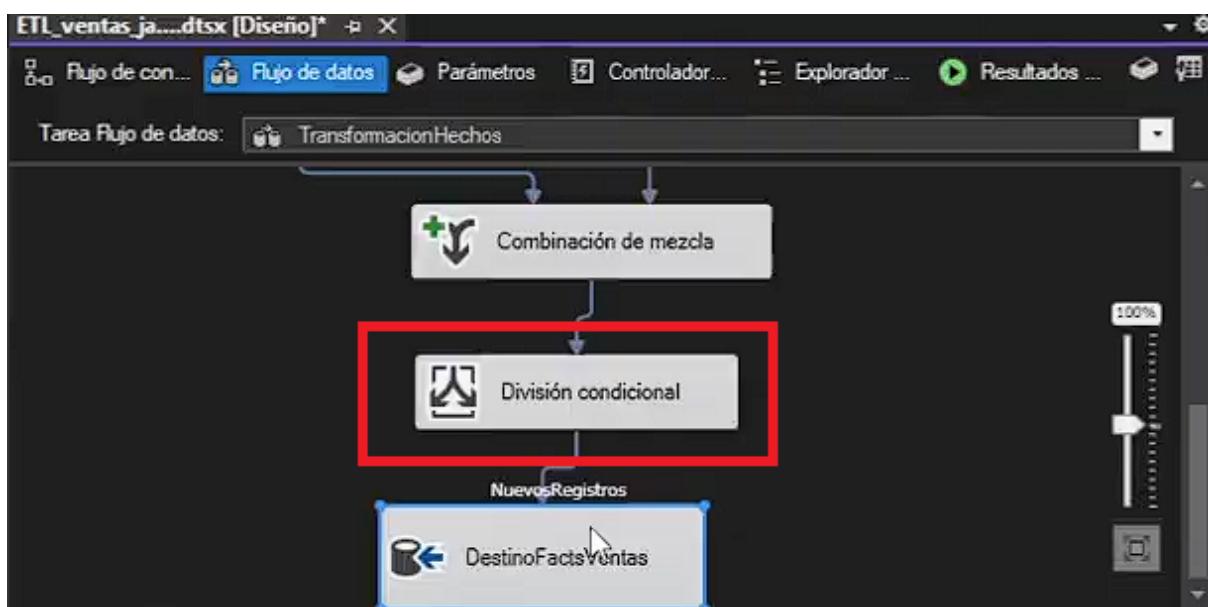
OrdenarVentas OrdenarFactsVentas

	Input	Input Column	Output Alias
OrdenarVentas	Nombre_Producto		Nombre_Producto
OrdenarVentas	Categoría		Categoría
OrdenarVentas	fecha_pedido		fecha_pedido
OrdenarVentas	ID_pedido		ID_pedido
OrdenarVentas	cantidad		cantidad
OrdenarVentas	precio_unidad		precio_unidad
OrdenarVentas	total_venta		total_venta
OrdenarVentas	Anio_Pedido		Anio_Pedido

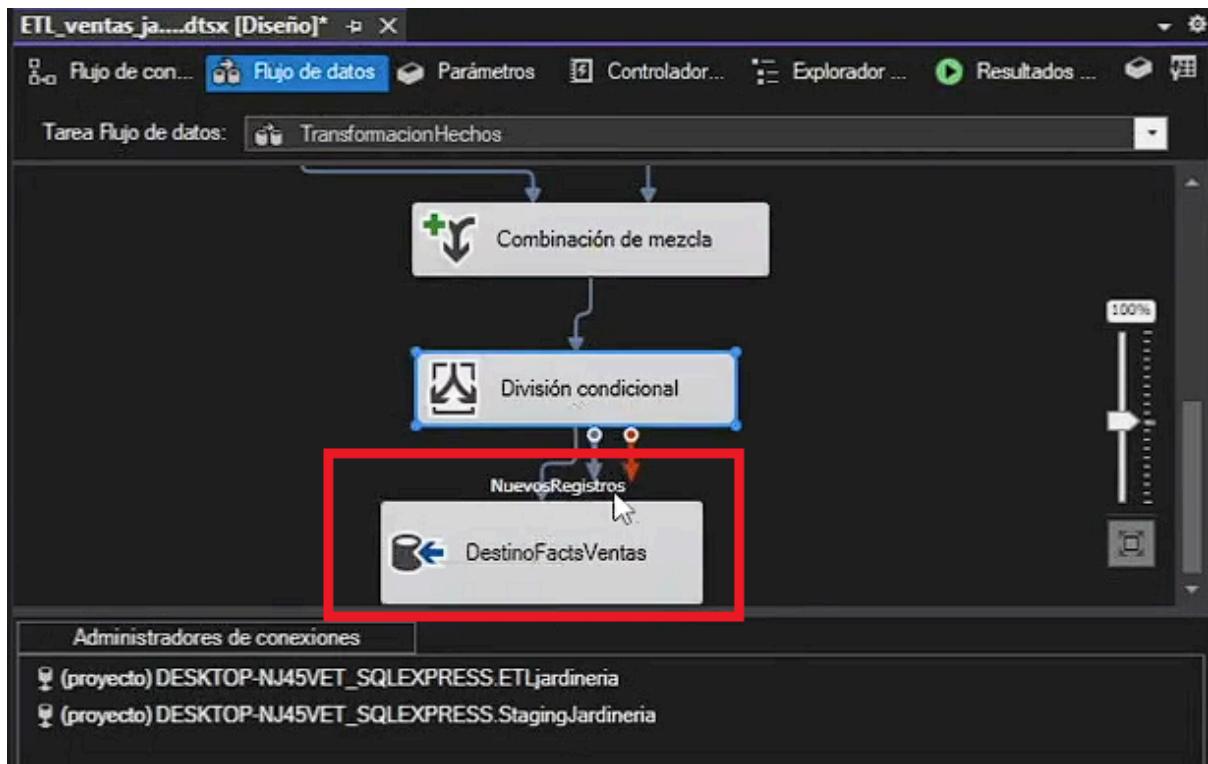
OK Cancel Ayuda

The screenshot shows the 'Merge Join Transformation Editor' window. At the top, it says 'Merge Join Transformation Editor' and provides instructions: 'Configure the properties used to join two sources of sorted data. Select the join type and then specify the columns to be used as the join key. Join keys must be used in the order specified by the sort-key position of the column.' Below this, the 'Join type' is set to 'Left outer join', which is highlighted with a red box. There are two input tables: 'OrdenarVentas' and 'OrdenarFactsVentas'. Both tables have columns: 'Nombre', 'Nombre_Producto', 'Categoría', 'fecha_pedido', 'ID_pedido', and 'cantidad'. In 'OrdenarVentas', 'Nombre_Producto' is marked as a join key (checked) and has an 'Orden' value of 1. In 'OrdenarFactsVentas', 'Nombre_Producto' is also checked and has an 'Orden' value of 1. A connection line links the two tables. Below the tables is a table mapping 'Input' columns to 'Output Alias' columns. The 'Input' column names are: 'Nombre_Producto', 'Categoría', 'fecha_pedido', 'ID_pedido', 'cantidad', 'precio_unidad', 'total_venta', and 'Anio_Pedido'. The 'Output Alias' column names are: 'Nombre_Producto', 'Categoría', 'fecha_pedido', 'ID_pedido', 'cantidad', 'precio_unidad', 'total_venta', and 'Anio_Pedido'. At the bottom are buttons for 'OK', 'Cancel', and 'Ayuda'.

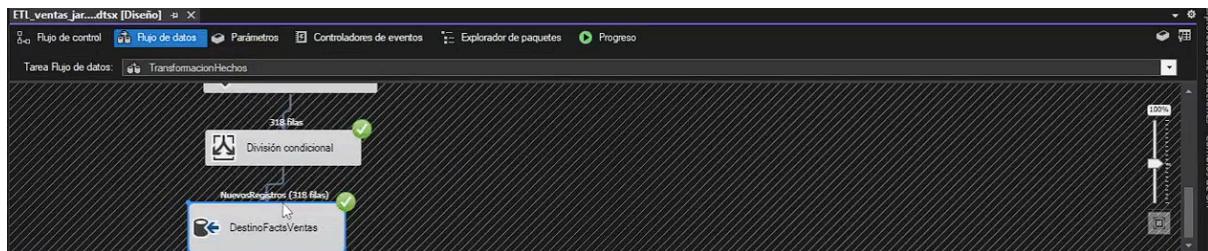
Continuamos con la configuración del Conditional Split



Y finalmente obtenemos el nuevo registro.

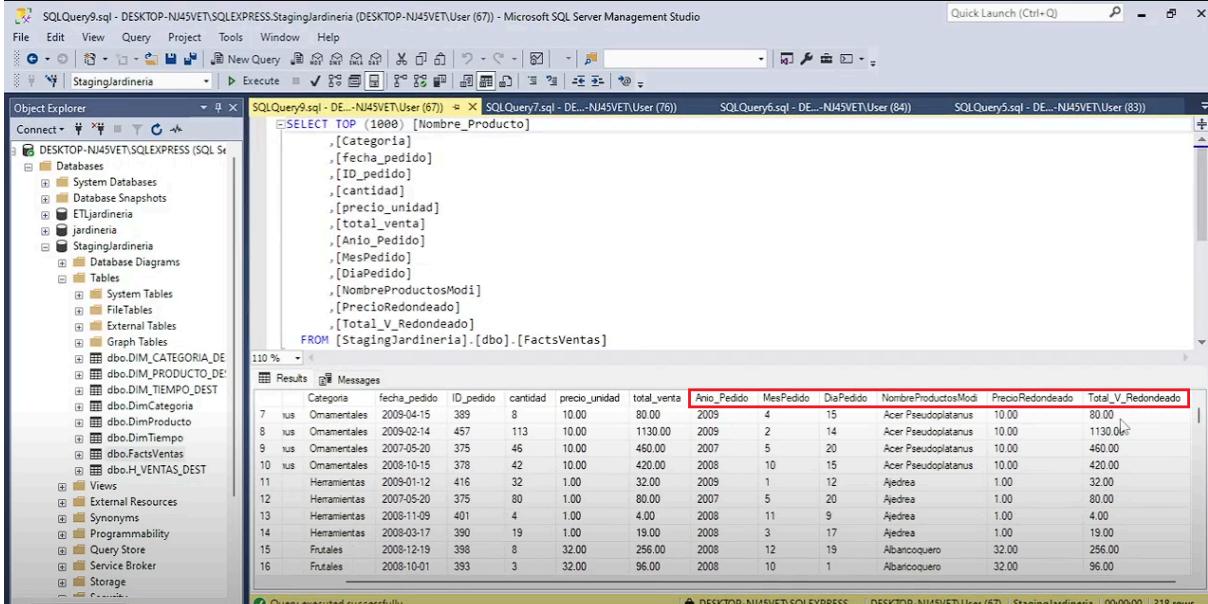


Procedemos a ejecutar para que nos corra las configuraciones que hemos realizado.



Vemos que la ejecución se completó correctamente.

Procedemos a revisar que nuestra transformación se aplicó correctamente en la base de datos.



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'StagingJardineria' is selected. In the center pane, a query window displays a SELECT statement. The results grid shows 318 rows of data from the 'FactsVentas' table, including columns like Categoría, fecha_pedido, ID_pedido, cantidad, precio_unidad, total_venta, Anio_Pedido, MesPedido, DiaPedido, NombreProductosModi, PrecioRedondeado, and Total_V_Redondeado. The data includes various product categories, purchase dates, and quantities, along with calculated fields for the year, month, day of the month, modified product name, rounded price, and total rounded value.

	Categoría	fecha_pedido	ID_pedido	cantidad	precio_unidad	total_venta	Anio_Pedido	MesPedido	DiaPedido	NombreProductosModi	PrecioRedondeado	Total_V_Redondeado	
7	ius	Ornamentales	2009-04-15	389	8	10.00	80.00	2009	4	15	Acer Pseudoplatanus	10.00	80.00
8	ius	Ornamentales	2009-02-14	457	113	10.00	1130.00	2009	2	14	Acer Pseudoplatanus	10.00	1130.00
9	ius	Ornamentales	2007-05-20	375	46	10.00	460.00	2007	5	20	Acer Pseudoplatanus	10.00	460.00
10	ius	Ornamentales	2008-10-15	378	42	10.00	420.00	2008	10	15	Acer Pseudoplatanus	10.00	420.00
11		Herramientas	2009-01-12	416	32	1.00	32.00	2009	1	12	Ajedrea	1.00	32.00
12		Herramientas	2007-05-20	375	80	1.00	80.00	2007	5	20	Ajedrea	1.00	80.00
13		Herramientas	2008-11-09	401	4	1.00	4.00	2008	11	9	Ajedrea	1.00	4.00
14		Herramientas	2008-03-17	390	19	1.00	19.00	2008	3	17	Ajedrea	1.00	19.00
15		Frutales	2008-12-19	398	8	32.00	256.00	2008	12	19	Albaricoquero	32.00	256.00
16		Frutales	2008-10-01	393	3	32.00	96.00	2008	10	1	Albaricoquero	32.00	96.00

Conclusiones

Al implementar el modelo de estrella en el Data Mart proporcionado nos encontramos que esta nos ofrece una ventaja para el análisis de datos ya que, por medio de las tablas dimensión nos facilita la comprensión del negocio y los requisitos de los usuarios. Al reducir el número de uniones necesarias para poder obtener la información deseada podemos mejorar el rendimiento de las consultas y permite la flexibilidad y la escalabilidad del diseño, ya que podemos modificar o agregar dimensiones y medidas sin afectar las demás tablas.

Es de gran importancia destacar el enfoque estructurado en la gestión de datos, dado que permite visualizar como un buen manejo de la información puede contribuir a una mejor toma de decisiones, teniendo en cuenta datos relevantes y usando la información específica para cada necesidad.

Bibliografía.

- PowerData. (20 de 08 de 2013). *Staging: la salvaguarda de los procesos ETL.*
Obtenido de
<https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/bid/312643/staging-la-salvaguarda-de-los-procesos-etl>.
- Navarro, L. F. (2024). Modelo estrella [1].
https://lucid.app/lucidchart/1bcb2143-fe2e-43c5-b683-bea31549df64/edit?page=0_0&invitationId=inv_85cbb478-ecef-4e20-87d9-c9ef57d971f1#

IBM. (08 de 03 de 2021). Esquemas de estrella. Obtenido de
<https://www.ibm.com/docs/es/ida/9.1.2?topic=modeling-data>

Calzada, J. M. (05 de 11 de 2020). *Seis razones por las que usar staging.*
Obtenido de <https://www.certia.net/seis-razones-por-las-que-usar-staging/>