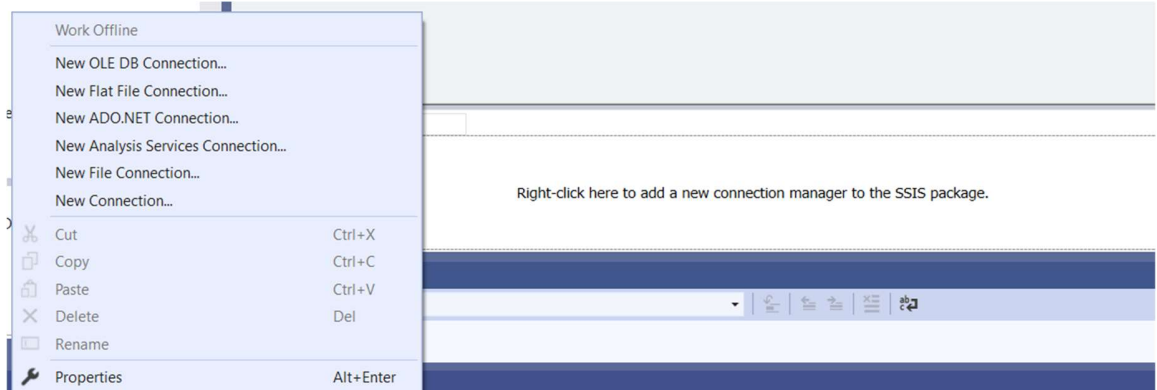
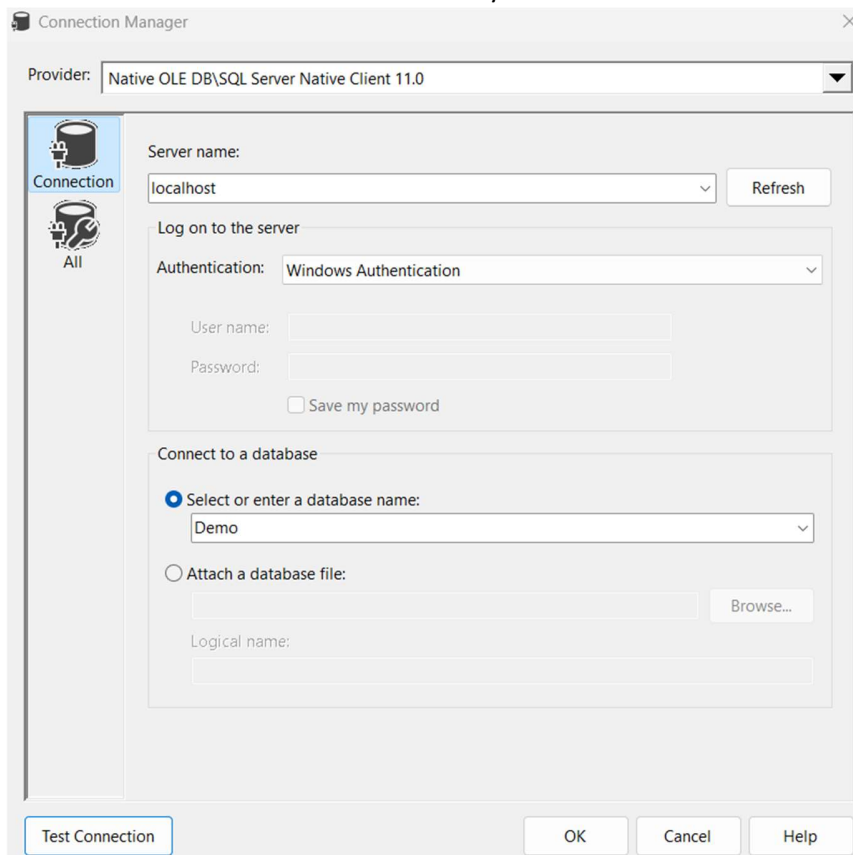


Clase Practica Preparando Carga del Data Warehouse

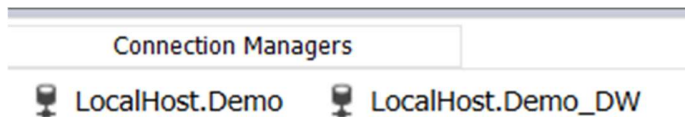
1. Crear un proyecto de SSIS y al paquete por defecto cambiar el nombre a Load_DW
2. Una de las primeras tareas que se debe hacer en un proyecto es crear las conexiones que nos permiten comunicarnos con los orígenes de datos. Así que crearemos dos conexiones OLEDB. Clic derecho en el área del connection manager y nos aparece esta ventana.



3. Seleccionamos New OLE DB Connection...y hacemos click al botón New

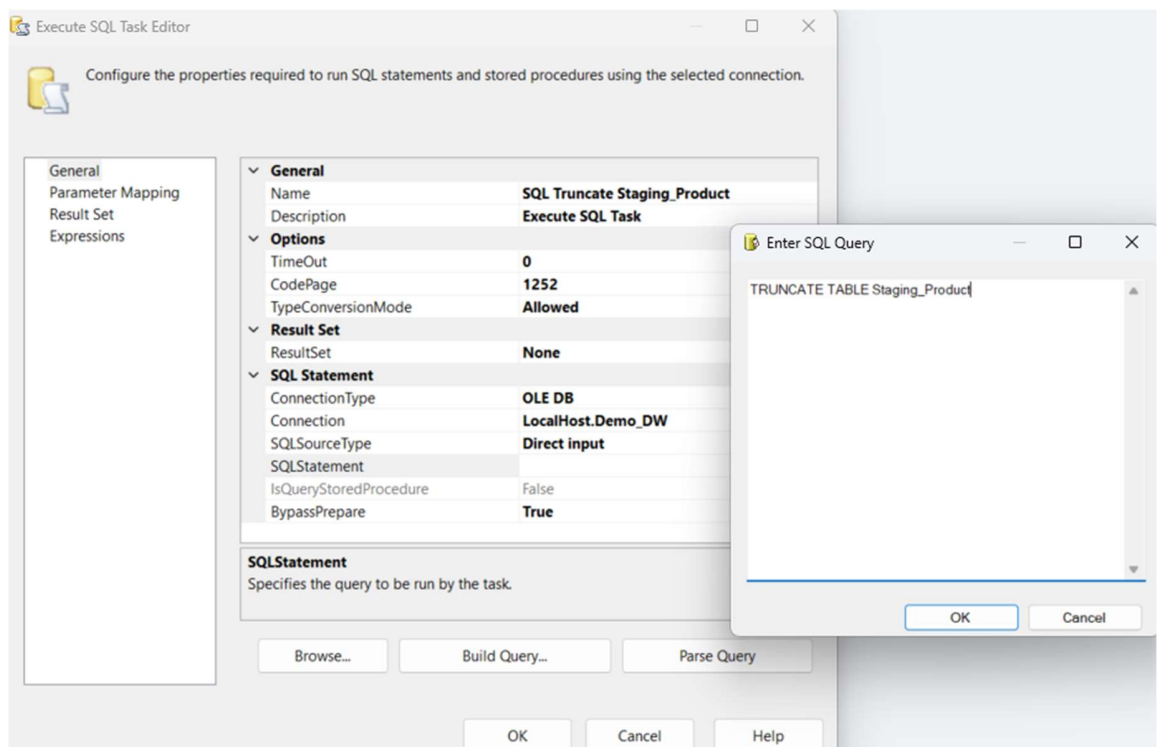


- Después de agregar una conexión a la base de datos Demo y otra conexión a la base de datos Demo_DW, nos debe quedar el connection manager de esta forma.

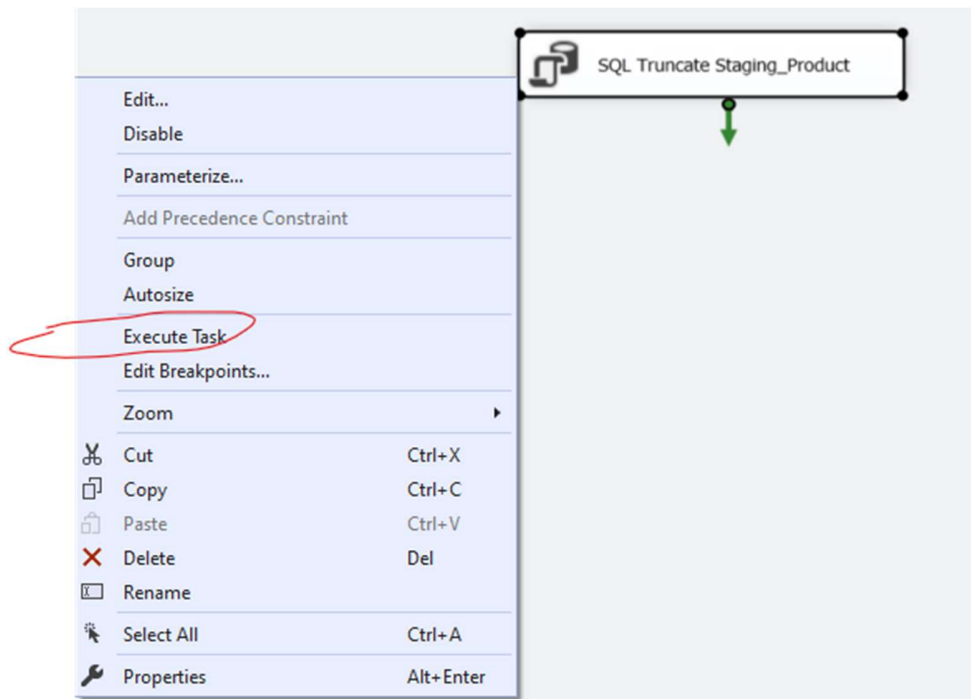


- Creando elemento del control de flujo.

Arrastramos desde el Toolbox una Execute SQL Task, hacia el panel central, clicamos dos veces y nos aparece la siguiente pantalla, en este caso empezaremos con truncar la tabla de Staging_Products. En la sección General cambiamos el Name a “SQL Truncate Staging_Product”, en la sección de SQL Statement, ConnectionType debe ser OLEDB, que es a como tenemos nuestras conexiones, SQL Source Type queda establecido en Direct Input y en SQL Statement van a establecer el comando de la figura en el lado derecho. Hecho esto, presionamos OK en ambas pantallas, para cerrar todo.



- Podemos probar la tarea, clicando la tarea y eligiendo del menú secundario la opción de Execute Task , o clic secundario en el nombre del paquete en el explorador de soluciones y con el botón secundario elegimos Execute Package, que ejecutaría todas las tareas que se encuentran en el paquete, en este caso da igual porque solo tenemos una.



Si la tarea se ejecuta con éxito, obtendremos esta imagen.



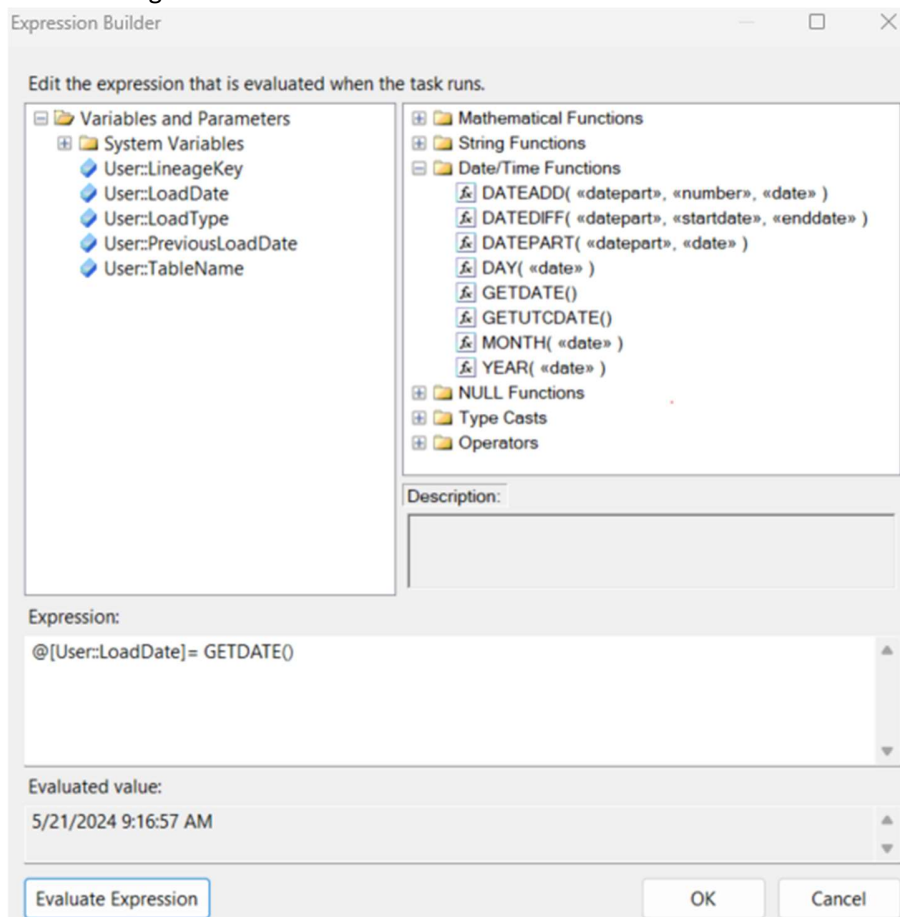
Y mientras esto sucede, estamos en modo depuración, así que no podemos hacer ningún cambio al paquete.



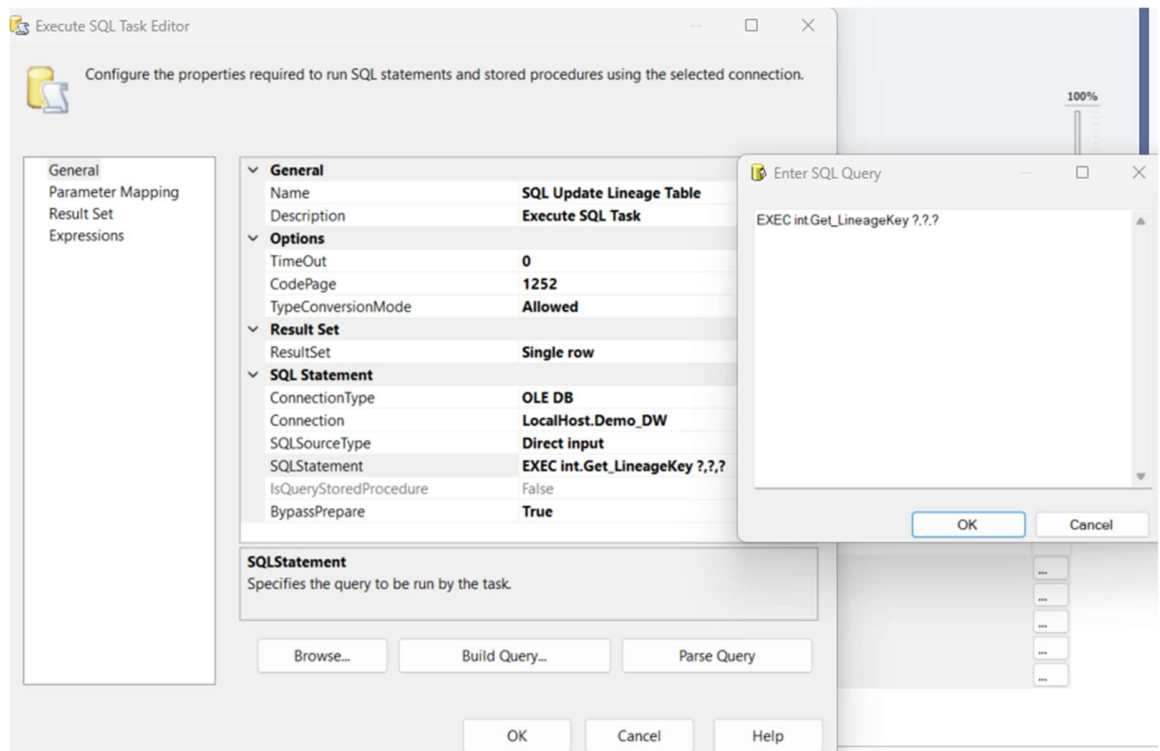
7. Vamos a empezar a trabajar con variables, así que vamos al menú contextual y elegimos la opción de menú secundario variables, en la parte inferior aparecerá una ventana donde agregaremos las variables que aparecen a continuación.

Variables				
Name	Scope	Data type	Value	Expression
LoadDate	Package	DateTime		
LineageKey	Package	Int32	-1	
TableName	Package	String	Dim_Product	
LoadType	Package	String	F	
PreviousLoadDate	Package	DateTime		

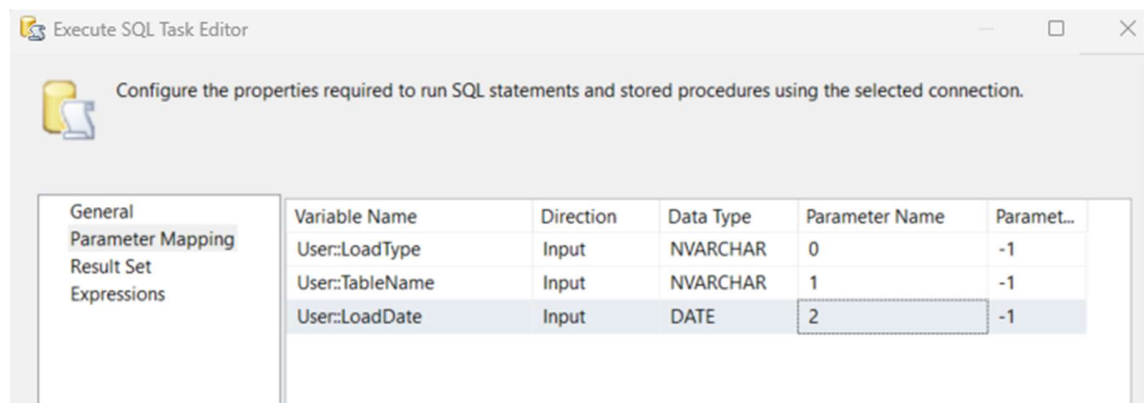
8. Ahora vamos a guardar la fecha de la carga, para eso arrastramos al área de diseño un componente del área common llamado Expression Task, este tipo de componente hace exactamente eso, asignarles valor a las variables, esta tarea la renombramos a Obtener la fecha de carga.



9. Ejecución de una SQL Task con parámetros, empezaremos arrastrando una SQL Task, cambiamos el nombre a SQL Update Lineage Table, revisamos el procedimiento int.Get_LineageKey, el cual tiene tres parámetros, LoadType, TableName y LastLoadedDate

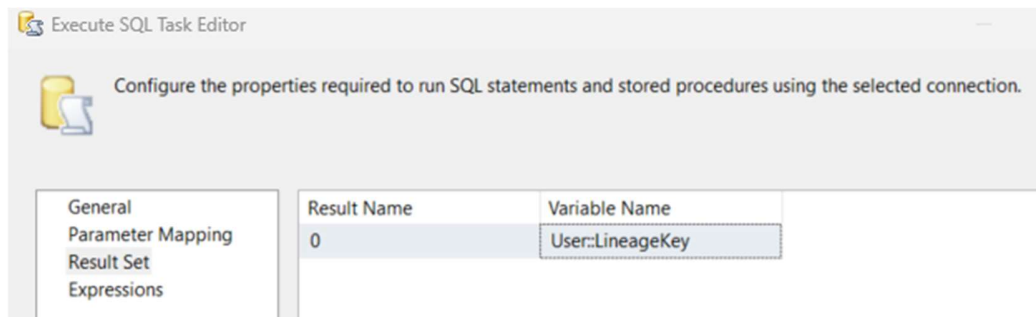


En la pestaña de Parameter Mapping donde establecemos los parámetros de entrada al procedimiento debe verse como en la imagen.



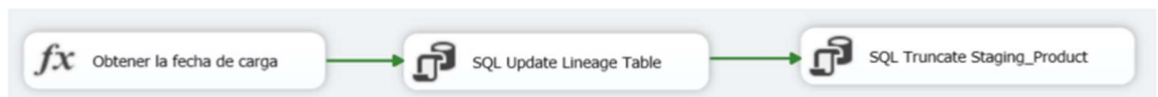
El nombre de parámetro depende del proveedor que usemos, cuando usamos OLEDB, se deben dar las posiciones comenzando con 0. (Recuerden los símbolos ?,?,? del comando que lo ejecuta)

Pasamos a la pestaña del conjunto de resultados, donde vamos a recoger lo que devuelve el procedimiento almacenado.

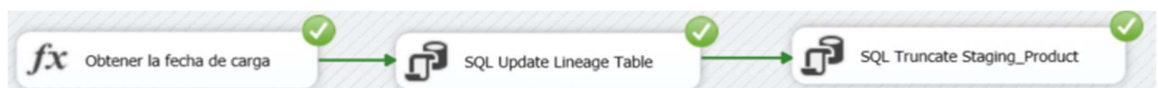


Con eso presionamos Ok.

Ordenamos las tareas para que se vean de la siguiente forma (me refiero al orden, no a la forma horizontal)



Después de ejecutar el paquete debería de ejecutarse sin problemas.



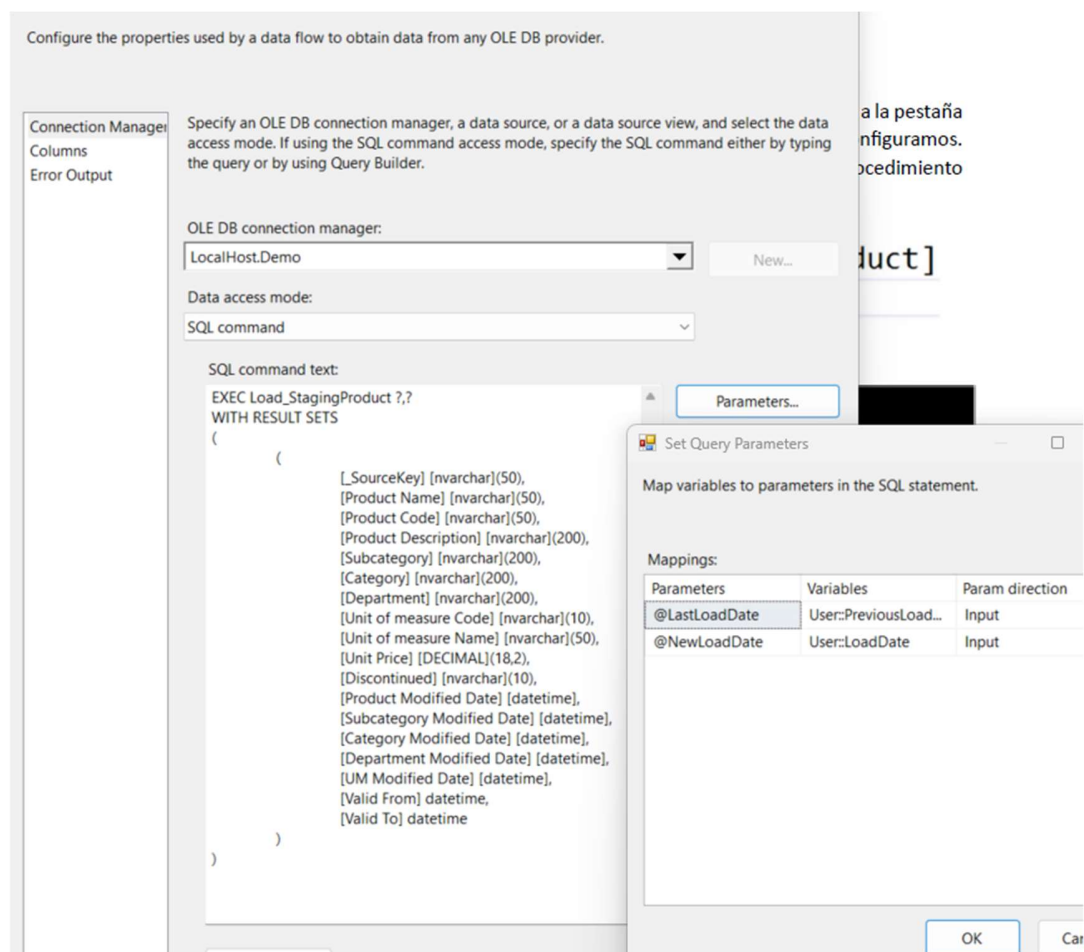
En la tabla Lineage debería de haber insertado un registro la ejecución del paquete.

10. En este paso van a hacer lo siguiente

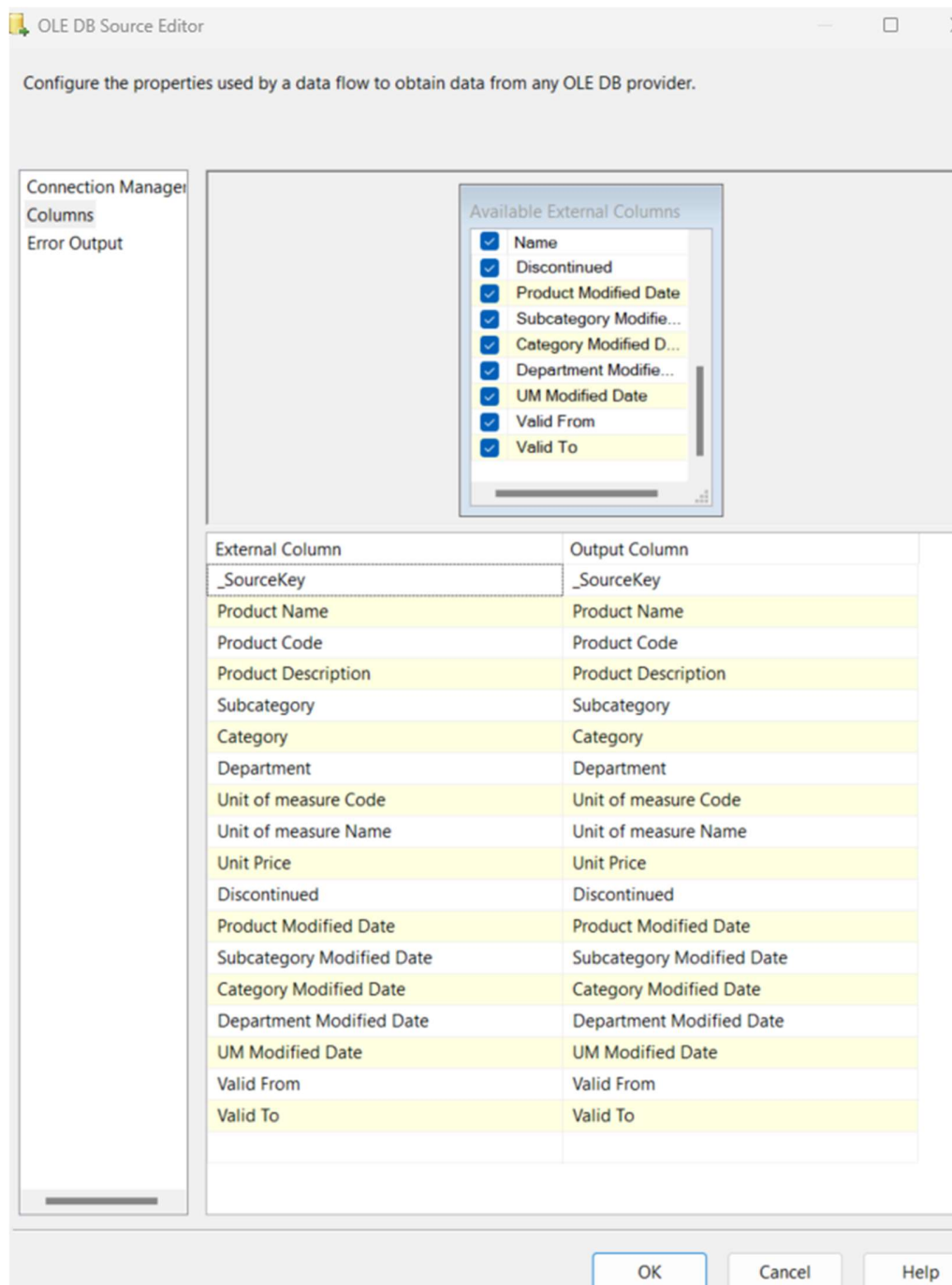
- a. Crear una SQL Task que ejecute el procedimiento [int]. [Get_LastLoadedDate]
- b. Guardar el resultado en la variable: [User::PreviousLoadDate]
- c. Ejecutar la tarea después que la tabla de staging es truncada.
- d. Renombrar la tarea a "SQL Obtener la fecha de la carga previa"

11. Continuando, vamos a agregar un Data Flow Task, hacemos doble clic y pasamos a la pestaña del flujo de datos, arrastramos desde Other Sources un OLE DB Source, y lo configuramos. Nos aseguramos que los parámetros se correspondan con los del procedimiento almacenado.

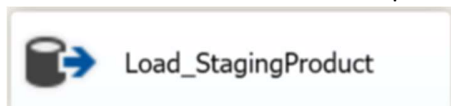
```
ALTER PROCEDURE [dbo].[Load_StagingProduct]
@LastLoadDate datetime,
@NewLoadDate datetime
```



12. Después de establecer los parámetros, podemos ver la correspondencia entre las columnas en la pestaña Columns.



13. Finalmente renombramos el componente a "Load_StagingProduct"



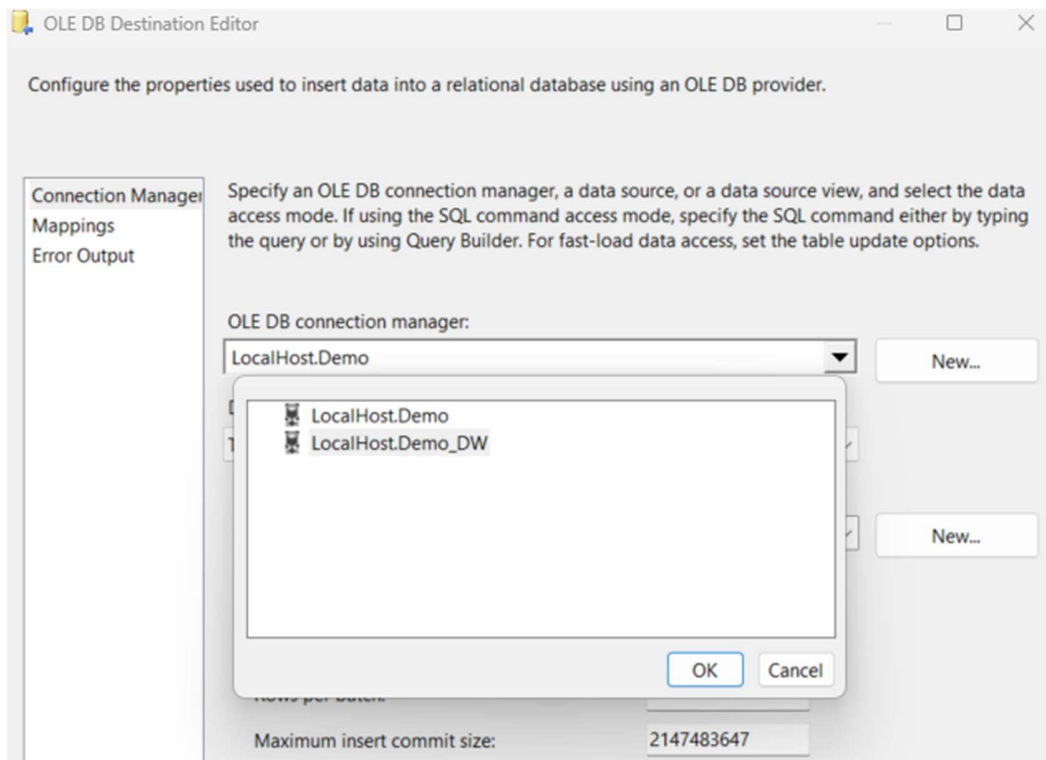
14. Bien ahora el paso natural es que agreguemos un destino para esos datos, esta será la tabla Staging_Product en la base de datos del data warehouse. Para eso arrastraremos un OLE DB Destination desde el grupo Other Destinations.



15. Ahora vamos a conectar el componente de source usando la flecha azul o purpura, a nuestro componente de destino, esto para que nuestro componente destino, nos de información sobre que columnas podemos acceder.



16. Doble clic en nuestro destino, y nos aparece la ventana, donde vamos a cambiar nuestra conexión a la conexión del data Warehouse, en este caso Localhost.Demo_DW



Como nuestro destino es una tabla en este caso Staging_Product, el modo de acceso, es tabla o vista, y buscamos nuestra tabla objetivo que seria la que mencionamos antes, a como se ve en la imagen.

OLE DB Destination Editor

Configure the properties used to insert data into a relational database using an OLE DB provider.

Connection Manager

Mappings

Error Output

Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder. For fast-load data access, set the table update options.

OLE DB connection manager:

localhost.Demo_DW

New...

Data access mode:

Table or view

Name of the table or the view:

[dbo].[Dim_Customer]

[dbo].[Dim_Customer]

[dbo].[Dim_Date]

[dbo].[Dim_Employee]

[dbo].[Dim_Location]

[dbo].[Dim_PaymentType]

[dbo].[Dim_Product]

[dbo].[Dim_Promotion]

[dbo].[Fact_Sales]

[dbo].[Staging_Customer]

[dbo].[Staging_Employee]

[dbo].[Staging_Location]

[dbo].[Staging_PaymentType]

[dbo].[Staging_Product]

[dbo].[Staging_Promotion]

[dbo].[Staging_Sales]

[dbo].[sysdiagrams]

[int].[IncrementalLoads]

[int].[Lineage]

New...

Map the columns on the

OK Cancel Help

17. La pestaña de Mappings, podemos verificar cual es la correspondencia entre las columnas de origen(Input) y las de destino (Destination Column)

Connection Manager
Mappings
Error Output

Available Input Columns

Name
_SourceKey
Product Name
Product Code
Product Description
Subcategory
Category
Department
Unit of measure Code
Unit of measure Name
Unit Price
Discontinued
Product Modified Date
Subcategory Modified Date
Category Modified Date
Department Modified Date
UM Modified Date
Valid From
Valid To

Available Destination Columns

Name
Product Key
_Source Key
Product Name
Product Code
Product Description
Product Subcategory
Product Category
Product Department
Unit Of Measure Code
Unit Of Measure Name
Unit Price
Discontinued
Product Modified Date
Subcategory Modified Date
Category Modified Date
Department Modified Date
UM Modified Date
Valid From
Valid To

Input Column	Destination Column
<ignore>	Product Key
_SourceKey	_Source Key
Product Name	Product Name
Product Code	Product Code
Product Description	Product Description
Subcategory	Product Subcategory
Category	Product Category
Department	Product Department
Unit of measure Code	Unit Of Measure Code
Unit of measure Name	Unit Of Measure Name
Unit Price	Unit Price
Discontinued	Discontinued
Product Modified Date	Product Modified Date
Subcategory Modified Date	Subcategory Modified Date
Category Modified Date	Category Modified Date
Department Modified Date	Department Modified Date
UM Modified Date	UM Modified Date
Valid From	Valid From
Valid To	Valid To

18. Verificamos la correspondencia del mapeo de las columnas, siempre es bueno hacer esa validación, hecho esto, cerramos la configuración del ole db destination y vamos al control Flow a establecer una restricción de precedencia del “SQL Obtener la fecha de carga previa” al componente “DF Populate Staging”.



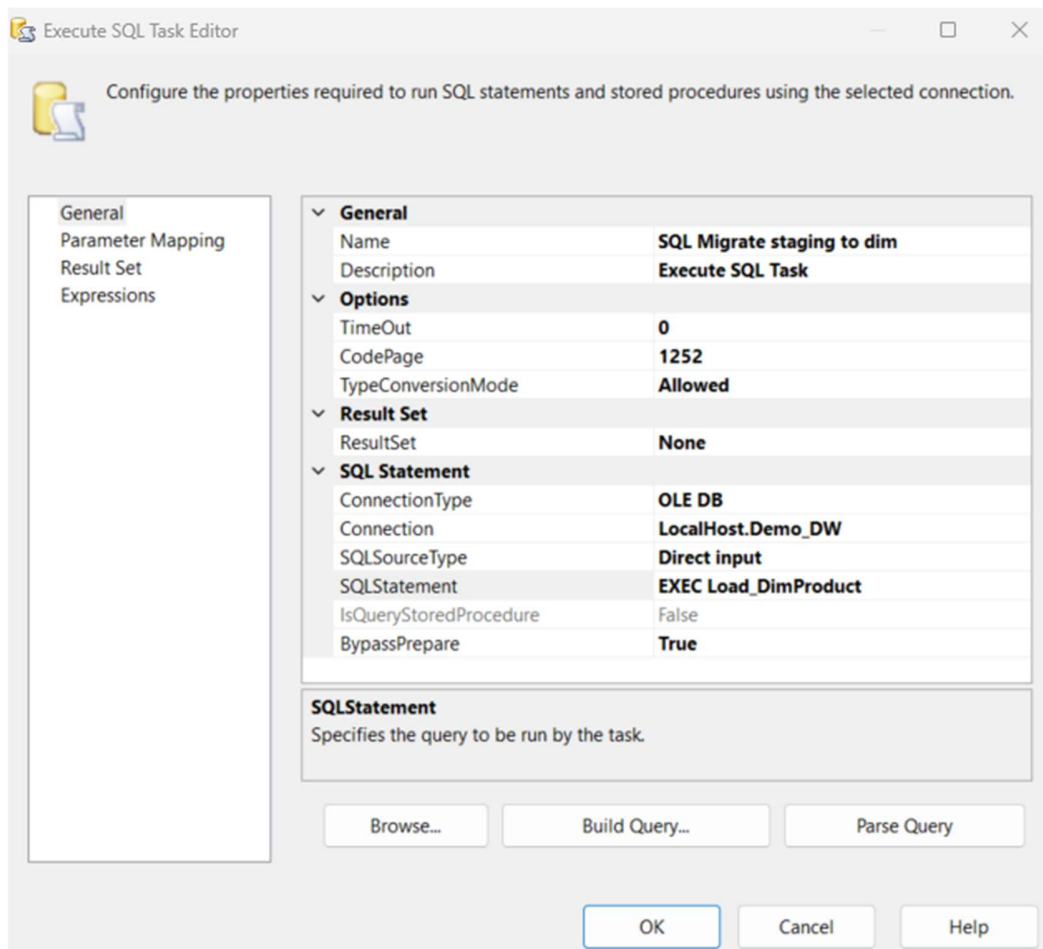
19. Después del paso anterior deberíamos de tener todo nuestro flujo conectado, en este punto al ejecutar nuestro paquete, deberíamos ver algo como esto



Y en la pestaña del flujo de datos podríamos ver cuántos registro movió dicho flujo, a como se ve en la imagen.



20. Para verificar la validez de esos números nos vamos a la tabla Staging_Product de la base de datos Demo_DW y deberían estar los 171 registros que movió nuestro paquete, en este caso de la base de datos transaccional a nuestra área de staging de la base de datos analítica.
21. Agregaremos un ultimo paso al Control Flow que será el movimiento de los datos de Staging_Product a Dim_Product, para esto arrastramos un Execute SQL Task al Control Flow y clicamos en él, para la configuración, la cual nos debe quedar como se ve en la imagen.



22. La pestaña del **Control Flow** debería verse tal y como se ve en la imagen siguiente



23. Solo nos quedaría agregar una restricción de precedencia, limpiar las tablas que serán afectadas por la operación usando el comando, de paso comprobamos que no contengan datos, esto con el fin de no duplicar los datos, además de obligar a las surrogate keys a que empiecen en el 1 nuevamente.

```
TRUNCATE TABLE DBO.STAGING_PRODUCT
TRUNCATE TABLE DBO.DIM_Product

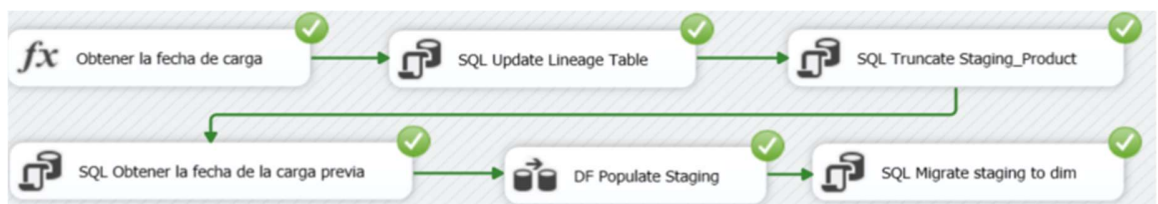
SELECT * FROM DBO.STAGING_PRODUCT
SELECT * FROM DBO.DIM_Product

delete FROM INT.Lineage
delete from int.IncrementalLoads

DBCC CHECKIDENT ('int.Lineage', RESEED, 0)
DBCC CHECKIDENT ('int.IncrementalLoads', RESEED, 0)
DBCC CHECKIDENT ('DBO.STAGING_PRODUCT', RESEED, 0)
DBCC CHECKIDENT ('DBO.DIM_Product', RESEED, 0)

SELECT * FROM INT.Lineage
SELECT * from int.IncrementalLoads
```

Eliminados los datos que puedan tener nuestras tablas y verificado que realmente no contienen datos, incluyendo las tablas de metadata el paquete listo para ejecutar se vería de esta forma en su pestaña Control Flow.



24. Verificando el trabajo realizado por el paquete, podemos ver los registros de las tablas de meta y el conteo de los registros de staging y la dimensión involucrada en este caso, Productos.

```
SELECT * FROM INT.Lineage
SELECT * from int.IncrementalLoads
SELECT COUNT(*) FROM DBO.Staging_Product
SELECT COUNT(1) FROM DBO.DIM_Product
```

LineageKey	TableName	StartLoad	FinishLoad	LastLoadedDate	Status	Type
1	Dim_Product	2024-05-23 13:09:15.697	2024-05-23 13:09:15.967	2024-05-23 13:09:16.000	S	F

LoadDateKey	TableName	LoadDate
1	Dim_Product	2024-05-23 13:09:16.000

(No column name)
171

(No column name)
172

¡Éxito!