

Actividad: EA3_Grupo#_ProcesoETL

Asignatura: Bases de Datos II

PREICA2401B010093

PREICA2401B010076

Docente: Víctor Hugo Mercado

Estudiante:

Elizabeth Alzate

Jimy Luis Mora Russy

Vanessa Foronda M.

Ingeniería Software y Datos

Institución: IU Digital de Antioquia

Fecha: 26 mayo 2024



OBJETIVO

Construir consultas SQL para la creación de una base de datos staging basada en el modelo relacional de la base de datos *Jardinería*. [Consúltala aquí](#).

1. Preparación:

- a. Revisar el modelo Estrella definido en el proyecto de análisis de la base de datos Jardinería para comprender la estructura y las relaciones entre las tablas de dimensiones y la tabla de hechos. [Consúltala aquí](#).

- b. Verificar la disponibilidad y consistencia de la base de datos de staging previamente creada para el proyecto.

2. Extracción de datos desde la base de datos origen hacia la BD de Staging:

- a. Utilizar consultas SQL para extraer datos relevantes de la base de datos origen y cargarlos en las tablas correspondientes de la base de datos de staging.

- b. Verificar la integridad y consistencia de los datos extraídos para asegurar que cumplan con los requisitos del modelo Estrella.

3. Transformación de datos según las necesidades analíticas:

- a. Aplicar técnicas de transformación de datos, como limpieza, normalización y enriquecimiento, para preparar los datos de acuerdo con las necesidades analíticas específicas.
- b. Realizar la transformación de los datos utilizando consultas SQL y herramientas de ETL (Extract, Transform, Load) según sea necesario para garantizar la calidad y coherencia de los datos.

4. Carga de registros en el Data Mart final:

- a. Diseñar consultas SQL o scripts de carga para insertar los registros transformados desde la base de datos de staging en las tablas del data mart final.
- b. Ejecutar las consultas de carga y verificar que los datos se hayan insertado correctamente en el data mart final.

5. Documentación y presentación:

- a. Documentar todo el proceso de transformación de datos y carga en un informe detallado que incluya una descripción de las etapas realizadas, las consultas SQL utilizadas.
- b. Presentar el informe de manera clara y concisa, asegurándose de incluir referencias al modelo Estrella y las técnicas de ETL aplicadas.

ÍNDICE: BASE DE DATOS STAGING

| | |
|---|-----------|
| 1. Introducción | 4 |
| 2. Objetivo | 4 |
| 3. Descripción del proceso | 4 |
| 4. Capturas del proceso | 5 |
| Preparación: | 5 |
| Gráfico | 6 |
| Extracción de datos desde la base de datos origen hacia la BD de Staging: | 7 |
| Transformación de datos según las necesidades analíticas: | 9 |
| Carga de registros en el Data Mart final: | 14 |
| 5. Anexos | 15 |

DESARROLLO

1. Introducción

A lo largo del curso, hemos explorado la relevancia de los procesos ETL (Extracción, Transformación y Carga) en el mundo de la gestión de datos. Estos procesos desempeñan un papel fundamental al convertir información desestructurada en datos útiles y coherentes para la toma de decisiones en las organizaciones.

En la siguiente actividad, nos centraremos en la fase de transformación de datos. Estos datos han sido extraídos de la base original de jardinería y ahora forman parte de nuestra base de datos en etapa de staging. Esta etapa es crucial, ya que prepara los datos para su posterior carga en el data warehouse, donde serán utilizados para análisis, informes y estrategias empresariales.

2. Objetivo

Mejorar la calidad, coherencia y utilidad de los datos de la base staging para su carga y siguiente análisis desde el punto de vista empresarial.

3. Descripción del proceso

En el proceso de transformación de datos, se toman las tablas y los campos previamente elegidos de jardinería a staging y es aquí donde sucede la transformación de los datos. Se inicia el flujo con la limpieza de datos, que consiste en corregir los errores identificados como datos faltantes, duplicados o nulos, tal como resulta en los campos de fecha de la tabla tiempo y comentarios de la tabla pedido; se le aplican las reglas de normalización y con esto, están listos para el siguiente paso. En este, se identifican los formatos de cada dato según la fuente de extracción, esto para convertirlos en un formato estándar, compatible con el destino en donde se almacenarán. También, se definen y aplican las reglas de negocio, se valida que cumpla con las restricciones, es decir que los datos sean confiables y consistentes con el origen del negocio.

Luego de esto se inicia la transformación, donde se definen las columnas finales que estarán en las dimensiones finales, aquí se ordenan los datos por el campo elegido, normalmente por el id o por fecha según el caso. Después del ordenamiento, se realiza un merge join, que se encarga de combinar los valores coincidentes de las filas de dos o más tablas y esto se da porque, la tabla destino donde quedarán nuestros datos

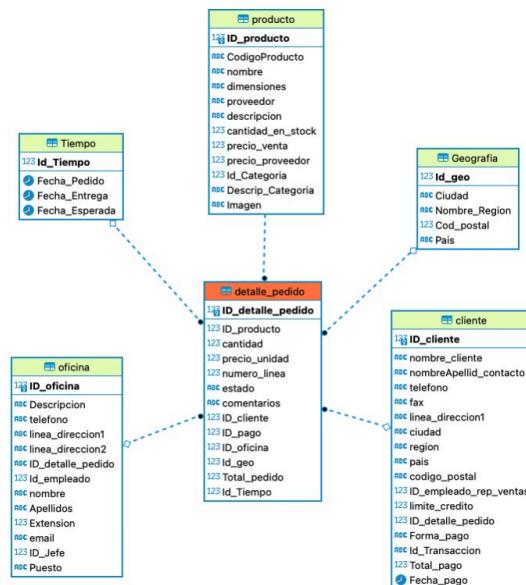
normalizados, es la tabla de origen para nuestro data warehouse. Así mismo, esta función nos ayuda a actualizar los registros de nuestras tablas, insertar o eliminarlos, de forma que quede la misma información tanto en la tabla de origen como en la de destino.

Para el último paso, se agrega un split condicional, el cual es un componente de integración de datos en el que controla la entrada y salida de datos entre las tablas de origen y destino, evalúa las condiciones y dirige los flujos a cada uno de ellas, optimizando el rendimiento y diferenciando cada proceso en el procesamiento de los datos. Con este proceso en marcha, se procede a la conexión de nuestros datos transformados a las tablas de la base final llamada data warehouse.

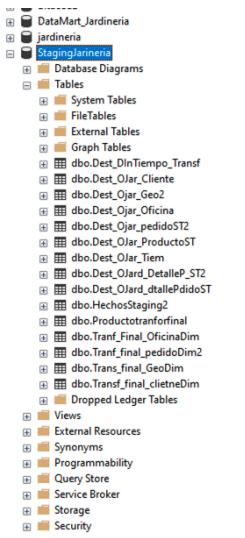
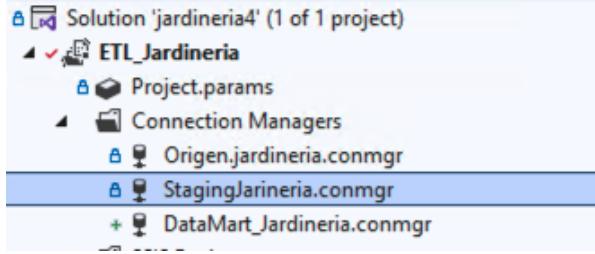
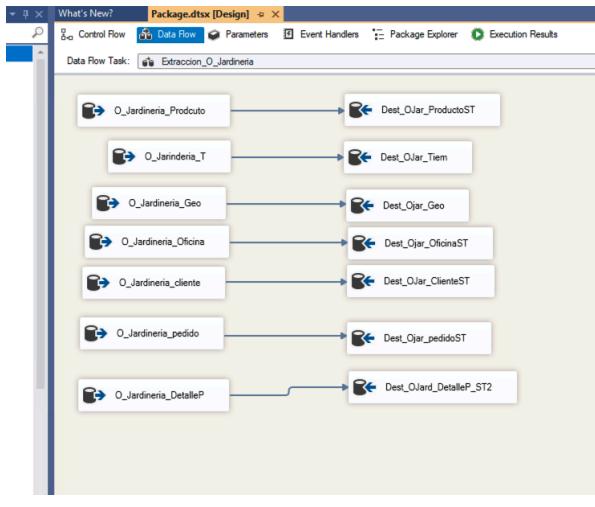
4. Capturas del proceso

Preparación:

Se revisa el modelo Estrella definido en el análisis de la DB Jardinería, se comprende la estructura y sus relaciones como un modelo estrella donde cuenta con 5 tablas de dimensiones: Tiempo, Geografía, Producto, Oficina, Cliente, Pedido, y una tabla central que es la tabla hechos donde se concentrará por medio de consulta SQL toda la información reunida, seleccionada y transformada de los datos.

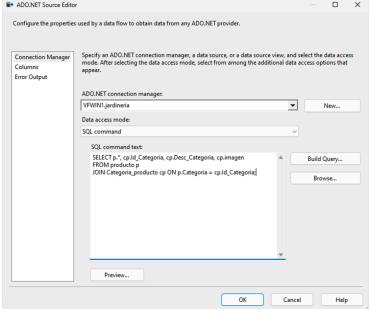
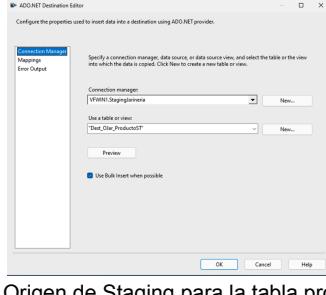
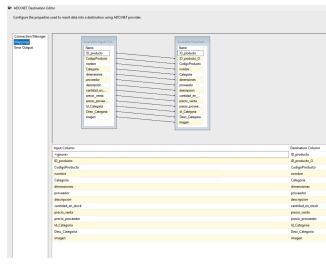
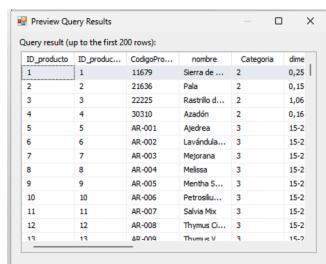


Se valida y se verifica la disponibilidad y consistencia de la bd de staging previamente creada para el proyecto:

| Gráfico | Descripción |
|---|--|
|  | <p>Se crea Base de datos Staging, con sus respectivas tablas que recibirán la extracción de la bd original Jardinería.</p> |
|  | <p>Se realizan conexiones de las bases de datos: Origen jardinería, Staging y Data mart.</p> |
|  | <p>Se crea el data flujo con las tablas de origen Jardinería y la Extracción con a las tablas de Origen en Staging.</p> |

Extracción de datos desde la base de datos origen hacia la BD de Staging:

Se realizan consultas SQL para extraer los datos relevantes de las BD original Jardinera y se cargan a la correspondiente BD de Staging, se verifica la integridad y consistencia de los datos extraídos para asegurar que cumplan con los requisitos del modelo estrella, a continuación algunas imágenes del proceso de extracción:

| Consultas SQL para extracción de datos | Proceso: |
|--|--|
|  | |
| Tabla Origen: Producto | <ul style="list-style-type: none">  <p>Creación y apuntador a la Tabla Origen de Staging para la tabla producto.</p> |
| | <ul style="list-style-type: none">  <p>Mapping de los campos extraídos</p> |
| | <ul style="list-style-type: none">  <p>Previsualización de datos extraídos.</p> |

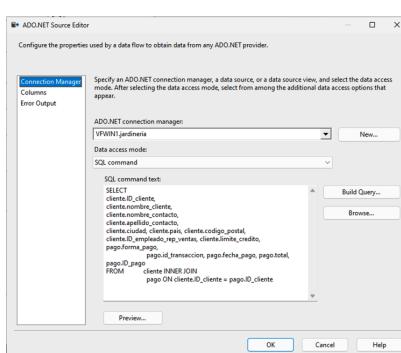
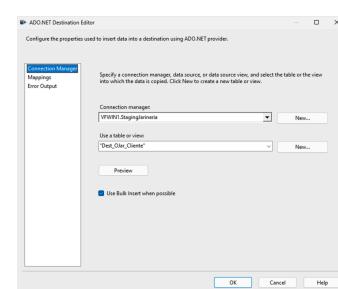
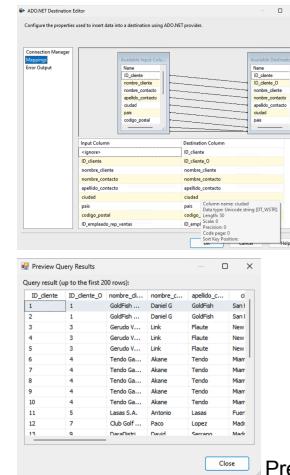


Tabla Origen Cliente



Creación y apuntador a la Tabla Origen de Staging para la tabla producto.



Mapping de los campos extraídos

Previsualización de datos extraídos.

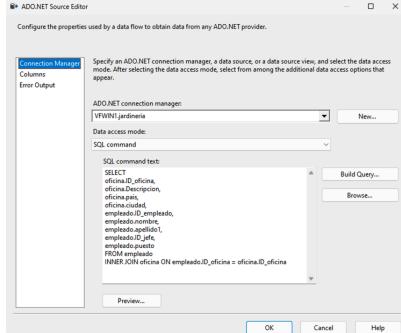
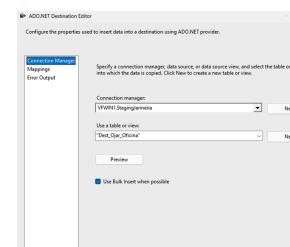
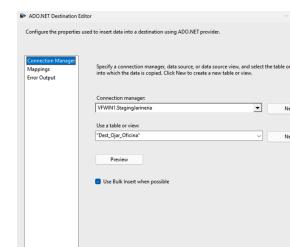
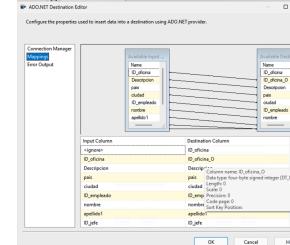


Tabla Origen Oficina



Creación y apuntador a la Tabla Origen de Staging para la tabla producto.



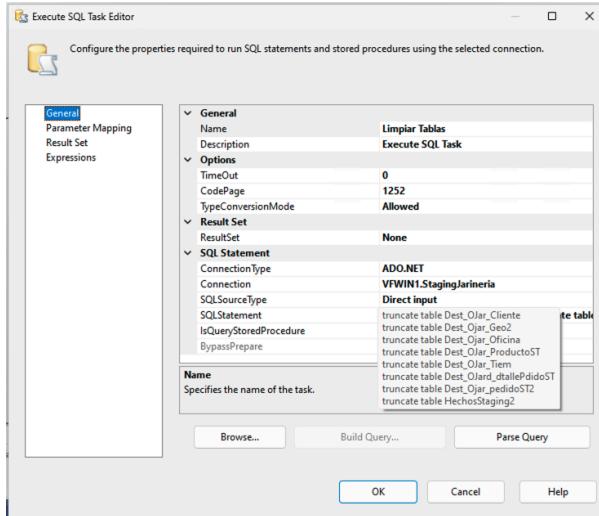
Mapping de los campos extraídos



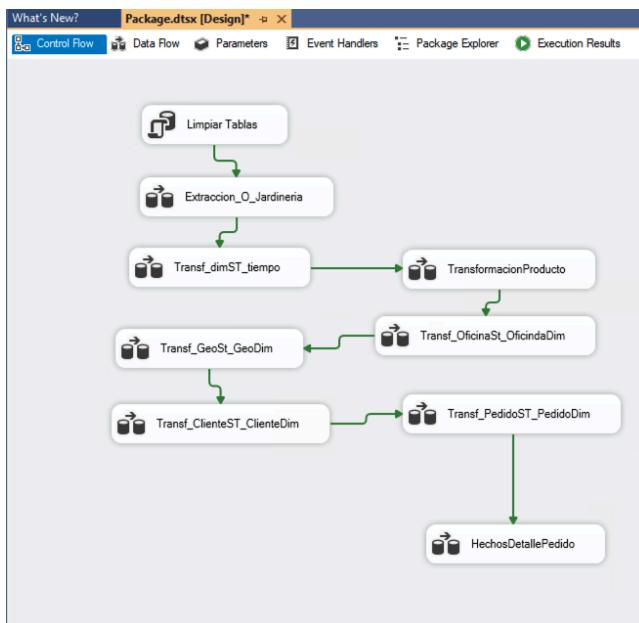
Previsualización de datos extraídos.

Transformación de datos según las necesidades analíticas:

Se realiza la limpieza de todas las tablas del origen del bd.staging con SQL Statement Truncate table:



Se realiza la transformación de los datos utilizando Consultas SQL y herramientas de ETL acorde con nuestro análisis del negocio, se garantiza la calidad y coherencia de los datos.



Cuando ingresamos a alguno de los data flow de transformación de alguna de tabla hallamos el flujo de transformación de los datos, de lado izquierdo se toman los datos de origen de las tablas staging, se realizan diferentes transformaciones en sus datos, se realiza el marge con join left que permitirá unir los

datos de tabla origen con tabla destino, y se validará por un split si corresponde a registros nuevos o no para incluirlo en la tabla destino del staging.

Transformación de algunas tablas:

TABLA TRANSFORMACIÓN Y CARGA PRODUCTO:

| | | |
|--|---|--|
| <p>Diagrama del flujo tabla Producto</p> | <p>Fuente de origen Staging</p> | <p>Derivada realizada donde se reemplaza el null del campo image por un 0</p> |
| <p>Mapping de conexión con la tabla origen de datamart.</p> | <p>Configuración del condicional split que no permite ingresar a la tabla destino registros duplicados, solo nuevos registros.</p> | <p>Creación y direccionamiento de datos a la tabla destino db.dataMart</p> |

| | | |
|--|--|--|
| <p>Flujo de datos corriendo correctamente.</p> | <p>Carga de dato en la tabla y consultas</p> | |
|--|--|--|

TABLA TRANSFORMACIÓN Y CARGA OFICINA:

| | | |
|--|---|---|
| <p>Diagrama del flujo tabla Oficina</p> | <p>Fuente de origen Staging</p> | <p>Derivada transformación de datos al campo ciudad.oficina a Upper y reemplaza el idJefe por 0 cuando este sea null.</p> |
| <p>Merge Join Transformation Editor</p> | <p>Conditional Split Transformation Editor</p> | <p>ADO.NET Destination Editor</p> |

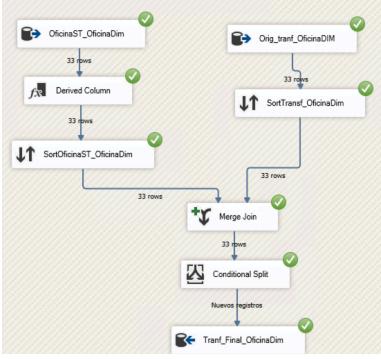
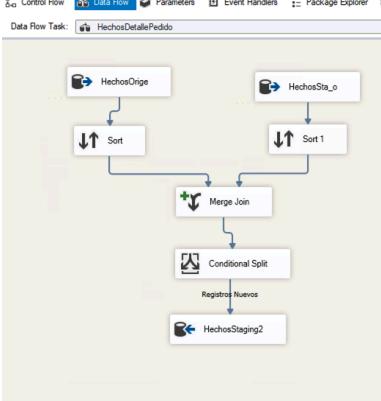
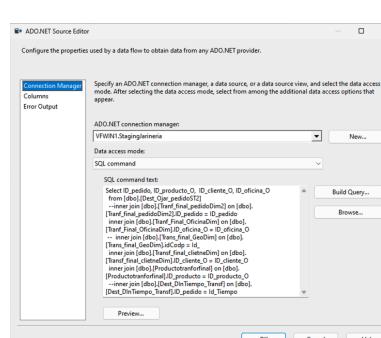
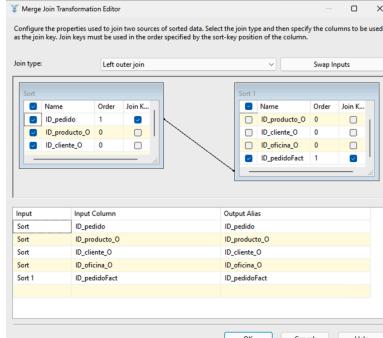
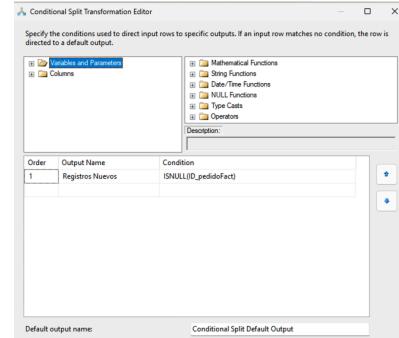
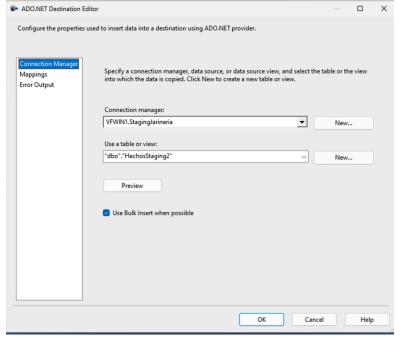
| | | |
|--|---|--|
| Mapping de conexión con la tabla origen de datamart. | Configuración del condicional split que no permite ingresar a la tabla destino registros duplicados, solo nuevos registros. | Creación y direccionamiento de datos a la tabla destino en db.dataMart |
|  <p>Flujo de datos corriendo correctamente.</p> | | |

TABLA TRANSFORMACIÓN Y CARGA HECHOS:

| | | |
|---|---|--|
| <p>Diagrama del flujo tabla Hechos</p>  | <p>Fuente de origen Consulta SQL*</p>  <pre> ADO.NET connection manager: VFWNT\Stageplm Data access mode: SQL command SQL command text: Select ID_pedido, ID_producto, O_ID_cliente, O_ID_oficina_O from [dbo].[HechosDrige] --inner join [dbo].[HechosFinal_pedido] on [dbo].[HechosDrige].ID_pedido = [HechosFinal_pedido].ID_pedido inner join [dbo].[HechosFinal_Prod] on [dbo].[HechosDrige].ID_pedido = [HechosFinal_Prod].ID_pedido inner join [dbo].[HechosFinal_Cli] on [dbo].[HechosDrige].ID_pedido = [HechosFinal_Cli].ID_pedido inner join [dbo].[HechosFinal_Geo] on [dbo].[HechosDrige].ID_pedido = [HechosFinal_Geo].ID_pedido inner join [dbo].[HechosFinal_Tiempo] on [dbo].[HechosDrige].ID_pedido = [HechosFinal_Tiempo].ID_pedido inner join [dbo].[HechosFinal_ProdCli] on [dbo].[HechosDrige].ID_pedido = [HechosFinal_ProdCli].ID_pedido inner join [dbo].[HechosFinal_ProdGeo] on [dbo].[HechosDrige].ID_pedido = [HechosFinal_ProdGeo].ID_pedido [HechosFinal_ProdCli].ID_producto = ID_producto, O_ID_cliente = O_ID_cliente, O_ID_oficina_O = O_ID_oficina_O [HechosFinal_ProdGeo].ID_producto = ID_producto, O_ID_cliente = O_ID_cliente, O_ID_oficina_O = O_ID_oficina_O [HechosFinal_Tiempo].ID_pedido = ID_pedido, O_ID_cliente = O_ID_cliente, O_ID_oficina_O = O_ID_oficina_O [HechosFinal_ProdCli].O_ID_producto = O_ID_producto, O_ID_cliente = O_ID_cliente, O_ID_oficina_O = O_ID_oficina_O [HechosFinal_ProdGeo].O_ID_producto = O_ID_producto, O_ID_cliente = O_ID_cliente, O_ID_oficina_O = O_ID_oficina_O [HechosFinal_Tiempo].O_ID_pedido = ID_pedido, O_ID_cliente = O_ID_cliente, O_ID_oficina_O = O_ID_oficina_O </pre> | |
|---|---|--|

| | | |
|---|---|---|
|  <p>Mapping de conexión con la tabla origen de datamart.</p> |  <p>Configuración del condicional split que no permite ingresar a la tabla destino registros duplicados, solo nuevos registros.</p> |  <p>Creación y direccionamiento de datos a la tabla destino en db.dataMart</p> |
|  <p>Flujo de datos corriendo correctamente.</p> |  <p>Consulta que integra datos de la tabla hechos.</p> | |

* . Consulta SQL de la creación de la tabla de Hechos:

```
Select ID_detalle_pedido, ID_detalle_pedido_O, ID_pedido_O AS pedidtabladped,
ID_producto_O, ID_cliente_O, ID_oficina_O,
cantidad, idCdp
from [dbo].[Dest_OJard_DetalleP_ST2]
inner join [dbo].[Tranf_final_pedidoDim2] on
[dbo].[Tranf_final_pedidoDim2].ID_pedido = ID_pedido_O
inner join [dbo].[Tranf_Final_OficinaDim] on
[dbo].[Tranf_Final_OficinaDim].ID_oficina_O = ID_oficina_O
inner join [dbo].[Trans_final_GeoDim] on [dbo].[Trans_final_GeoDim].idCdp= idCdp
inner join [dbo].[Transf_final_clienteDim] on
[dbo].[Transf_final_clienteDim].ID_cliente_O = ID_cliente_O
```

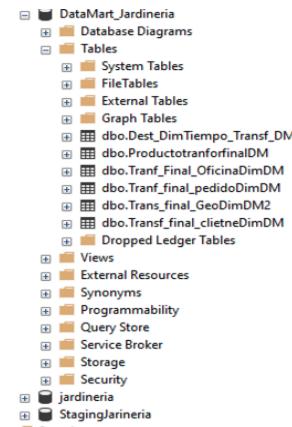
```

inner join [dbo].[Productotranforfinal] on [dbo].[Productotranforfinal].ID_producto
= ID_producto_O
-- inner join [dbo].[Dest_DInTiempo_Transf] on
[dbo].[Dest_DInTiempo_Transf].ID_pedido_O = Id_Tiempo

```

Carga de registros en el Data Mart final:

Se ejecuta y se confirma que los datos están insertados correctamente en el data mart final.



SQLQuery2.asp - VWFWin1\vfmu (57)*

```

    , [Id_cliente]
    , [Id_oficina]
    , [Id_pago]
    , [estado]
    , [comentarios]
FROM [StagingJardineria].[dbo].[Dest_OJard_DetalleP_ST2]

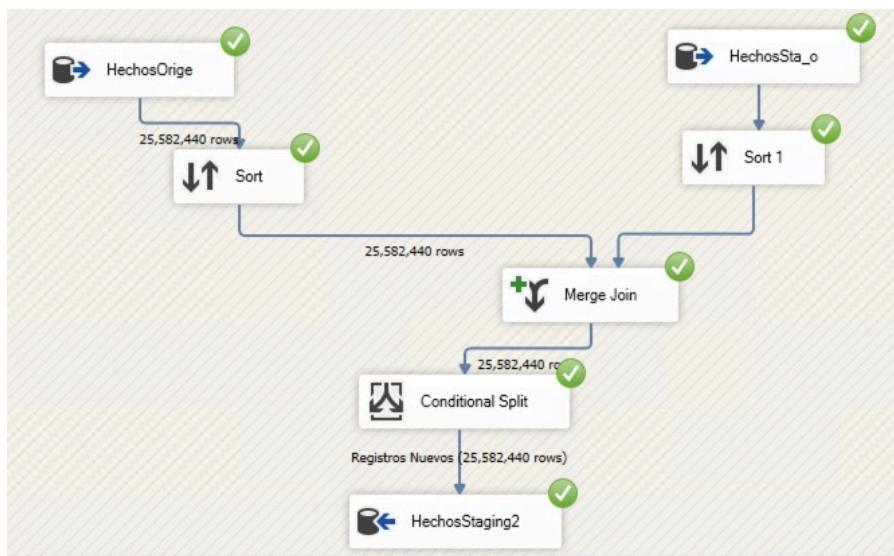
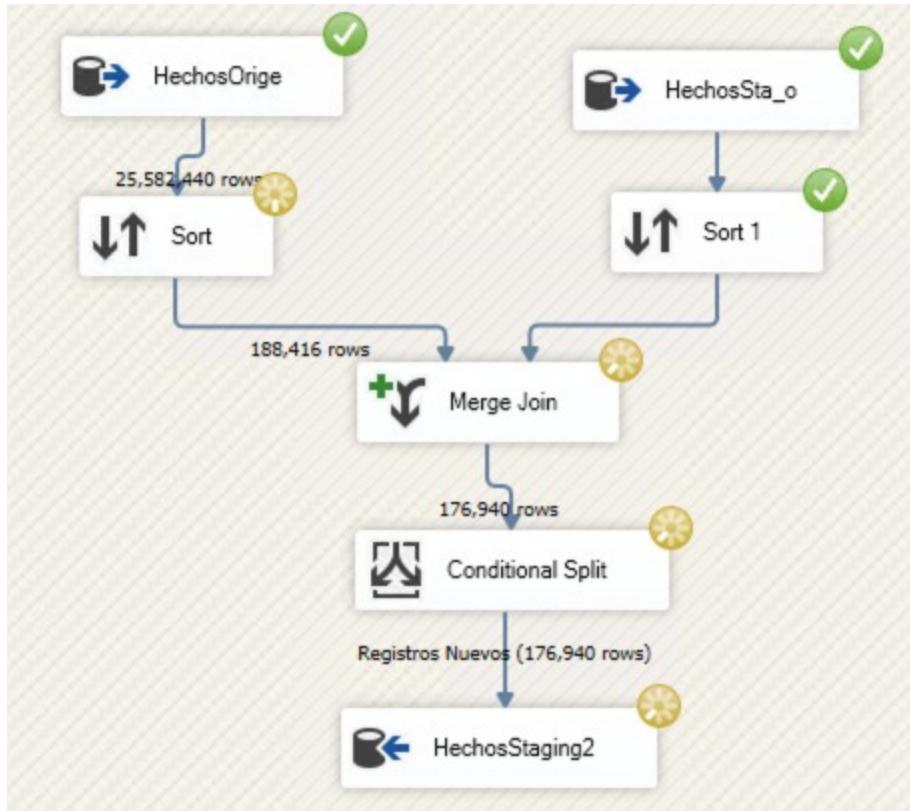
Select ID_detalle_pedido, ID_detalle_pedido_O, pedidatabledped, ID_producto_O, ID_cliente_O, ID_oficina_O,
       cantidad, idCdp
  from [dbo].[Dest_OJard_DetalleP_ST2]
inner join [dbo].[Tranf_final_pedidoDim2] on [dbo].[Tranf_final_pedidoDim2].ID_pedido = ID_pedido_O
inner join [dbo].[Tranf_Final_OficinaDim] on [dbo].[Tranf_Final_OficinaDim].ID_oficina_O = ID_oficina_O
inner join [dbo].[Trans_final_GeoDim] on [dbo].[Trans_final_GeoDim].idCdp = idCdp
inner join [dbo].[Tranf_final_clienteDim] on [dbo].[Tranf_final_clienteDim].ID_cliente_O = ID_cliente_O
inner join [dbo].[Productotranforfinal] on [dbo].[Productotranforfinal].ID_producto = ID_producto_O
-- inner join [dbo].[Dest_DInTiempo_Transf] on [dbo].[Dest_DInTiempo_Transf].ID_pedido_O = Id_Tiempo

```

Results

| ID_detalle_pedido | ID_detalle_pedido_O | pedidatabledped | ID_producto_O | ID_cliente_O | ID_oficina_O | cantidad | idCdp |
|-------------------|---------------------|-----------------|---------------|--------------|--------------|----------|-------|
| 1 | 1 | 1 | 50 | 1 | 8 | 10 | 1 |
| 2 | 2 | 1 | 50 | 1 | 8 | 40 | 1 |
| 3 | 3 | 1 | 50 | 1 | 8 | 25 | 1 |
| 4 | 4 | 1 | 50 | 1 | 8 | 15 | 1 |
| 5 | 5 | 1 | 50 | 1 | 8 | 23 | 1 |
| 6 | 6 | 1 | 50 | 1 | 8 | 3 | 1 |
| 7 | 7 | 1 | 50 | 1 | 8 | 7 | 1 |
| 8 | 8 | 1 | 50 | 1 | 8 | 50 | 1 |
| 9 | 9 | 1 | 50 | 1 | 8 | 20 | 1 |
| 10 | 10 | 1 | 50 | 1 | 8 | 12 | 1 |
| 11 | 11 | 1 | 50 | 1 | 8 | 67 | 1 |
| 12 | 12 | 1 | 50 | 1 | 8 | 5 | 1 |
| 13 | 13 | 1 | 50 | 1 | 8 | 120 | 1 |
| 14 | 14 | 1 | 50 | 1 | 8 | 32 | 1 |
| 15 | 15 | 1 | 50 | 1 | 8 | 11 | 1 |
| 16 | 16 | 1 | 50 | 1 | 8 | 30 | 1 |
| 17 | 17 | 1 | 50 | 1 | 8 | 15 | 1 |
| 18 | 18 | 1 | 50 | 1 | 8 | 24 | 1 |
| 19 | 19 | 1 | 50 | 1 | 8 | 12 | 1 |

Activate Windows
Go to Settings to activate Windows.



5. Anexos

Enlace archivos: Base de datos staging, jardinería y proyecto en Visual Studio.
<https://drive.google.com/drive/folders/1UI8VwoINPxieE4wyOLM6xyHyhDbV5k0->

Entrega 3:

[Vanesafm1/jardineria4 \(github.com\)](https://github.com/Vanesafm1/jardineria4)

Actualizado mayo 26 2024 9:27pm

Entrega 4

[Vanesafm1/jardineria4 \(github.com\)](https://github.com/Vanesafm1/jardineria4)

Actualizado mayo 26 2024 9:27pm