Evidencia de aprendizaje 3. Proceso de transformación de datos

Andrés Camilo Arango Calle

Docente:

Victor Hugo Mercado Ramos

Curso:

Bases de Datos II

Programa Ingeniería de Software y Datos

Facultad Ingenierías y Ciencias Agropecuarias

Institución Universitaria Digital de Antioquia

2024

Introducción

Una base de datos de Staging desempeña un papel crucial en el proceso de desarrollo de software y gestión de datos. Actuando como un ambiente controlado, ofrece un espacio designado para realizar pruebas exhaustivas y validar cambios antes de que se desplieguen en la base de datos de producción. Al simular las condiciones de producción, pero en un entorno seguro, estas bases de datos permiten a los equipos de desarrollo detectar y corregir errores potenciales, mientras optimizan el rendimiento de las aplicaciones, facilitando así una transición sin problemas hacia la producción.

Además de su función en la garantía de calidad y estabilidad del software, las bases de datos de Staging desempeñan un papel vital en la gestión de riesgos. Al separar el proceso de prueba y desarrollo del entorno de producción, se reduce el impacto de posibles errores en los datos y procesos empresariales críticos. Esta separación también fomenta la innovación, ya que proporciona un espacio seguro para explorar nuevas funcionalidades y tecnologías sin comprometer la seguridad de los datos en producción.

Las bases de datos de staging son esenciales en la inteligencia empresarial, actuando como un puente entre los datos brutos y la información útil. No solo garantizan la calidad y estabilidad del software, sino que también son el punto de partida para la transformación de datos en dimensiones y tablas de hechos, los componentes básicos de un data warehouse.

En el entorno de staging, los datos se limpian, estandarizan y enriquecen, preparándolos para su estructuración en dimensiones y tablas de hechos. Este proceso meticuloso asegura la calidad y fiabilidad de la información que alimentará los análisis y reportes, permitiendo a las organizaciones tomar decisiones informadas y basadas en datos sólidos.

Objetivos

1. Validar los datos antes de la carga en el modelo dimensional para garantizar su integridad y calidad.
2. Facilitar la transformación y ETL de los datos, permitiendo realizar las manipulaciones necesarias antes de cargarlos en el modelo dimensional.
3. Probar y validar el modelo dimensional en un entorno separado, asegurando su correcto funcionamiento antes de implementarlo en producción.

Planeamiento y análisis del problema

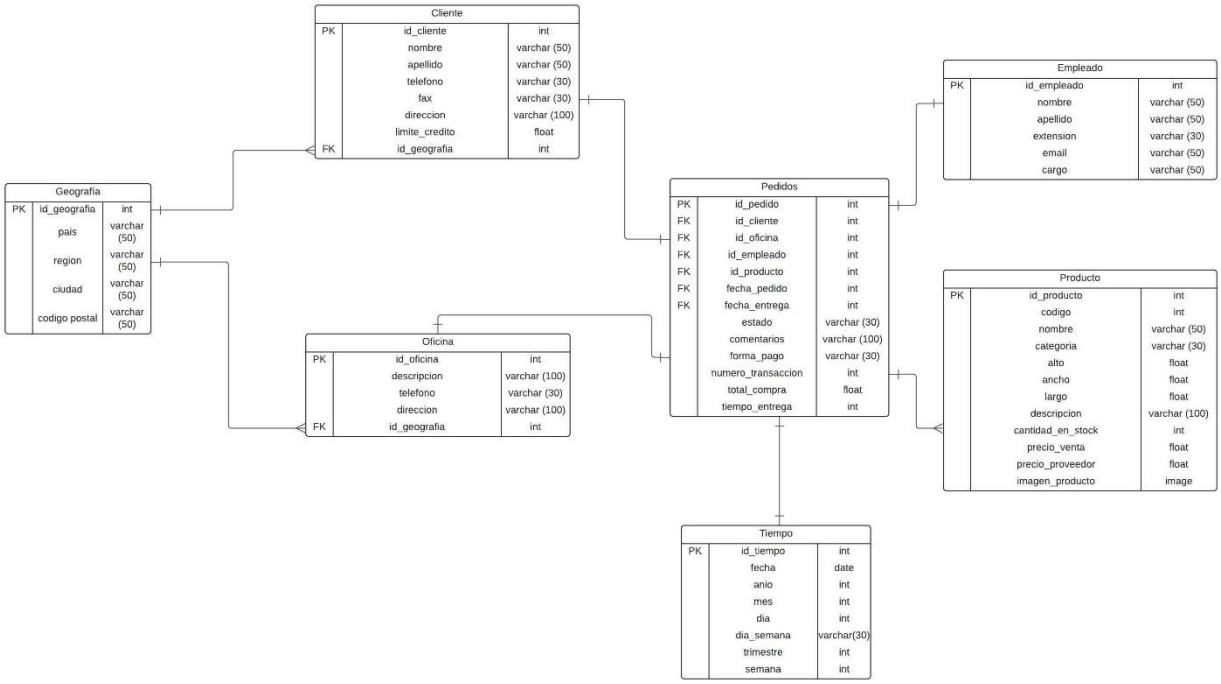
En el presente escenario, nos encontramos con el desafío de crear una base de datos de Staging a partir de una fuente de datos que consta de múltiples tablas. El objetivo principal es identificar y elegir los datos más pertinentes de cada tabla de origen para luego convertirlos en dimensiones que serán utilizadas para alimentar un modelo dimensional.

Al analizar este problema, nos enfrentamos a varios aspectos importantes:

- La base de datos original está compuesta por diversas tablas con diferentes tipos de datos y relaciones entre ellas. Esta complejidad agrega un nivel adicional de dificultad al proceso de selección y transformación de los datos, ya que implica la identificación de las tablas y campos más relevantes para el modelo dimensional.
- Es crucial determinar qué datos son más significativos para el modelo dimensional. Esto requiere una evaluación detallada de los requisitos del negocio y las necesidades de los usuarios finales para determinar qué información es esencial para la toma de decisiones y el análisis de datos.
- Una vez que se han identificado los datos relevantes, es necesario establecer un proceso eficiente para seleccionar y transformar estos datos en dimensiones. Esto implica la implementación de técnicas de limpieza, normalización y enriquecimiento de datos para garantizar su integridad y calidad en la base de datos de Staging.

Solución

1. Correcciones entrega 1



- **Análisis de dimensiones:**
  - **Dimensión Cliente:**
    - Id_cliente (PK)
    - Nombre_cliente
    - Teléfono
    - Linea_direccion1
    - Linea_direccion2
    - Id_geografia(FK)
    - Vendedor
    - Limite_credito

- **Dimensión Geografia:**
  - Id_geografia(PK)
  - País
  - Ciudad
  - Región
  - Código_postal
- **Dimensión Oficina:**
  - Id_oficina (PK)
  - Descripción
  - Id_geografia(FK)
  - Telefono
  - Dirección
- **Dimensión Empleado:**
  - id_empleado (PK)
  - nombre
  - extensión
  - email
  - oficina
  - jefe
  - puesto
- **Dimensión Producto:**
  - Id_producto (PK)
  - Código

        - Nombre
        - Categoría
        - Dimensiones
        - Proveedor
        - Descripción
        - Cantidad_en_stock
        - Precio_venta
        - Precio_proveedor
    - **Dimensión Tiempo**
        - Id_tiempo (PK)
        - Fecha
        - Año
        - Mes
        - Día
        - DíaSemanaNum
        - NumeroSemanaContable
        - Trimestre
        - DiaDelAnio
- **Tabla de hechos (Pedidos):**
    - Id_cliente (FK)
    - Id_oficina (FK)
    - Id_empleado (FK)
    - Id_producto (FK)

- Fecha_pedido (FK)
- Id_pedido (PK)
- Estado
- Comentarios
- Cantidad
- Precio_unidad

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

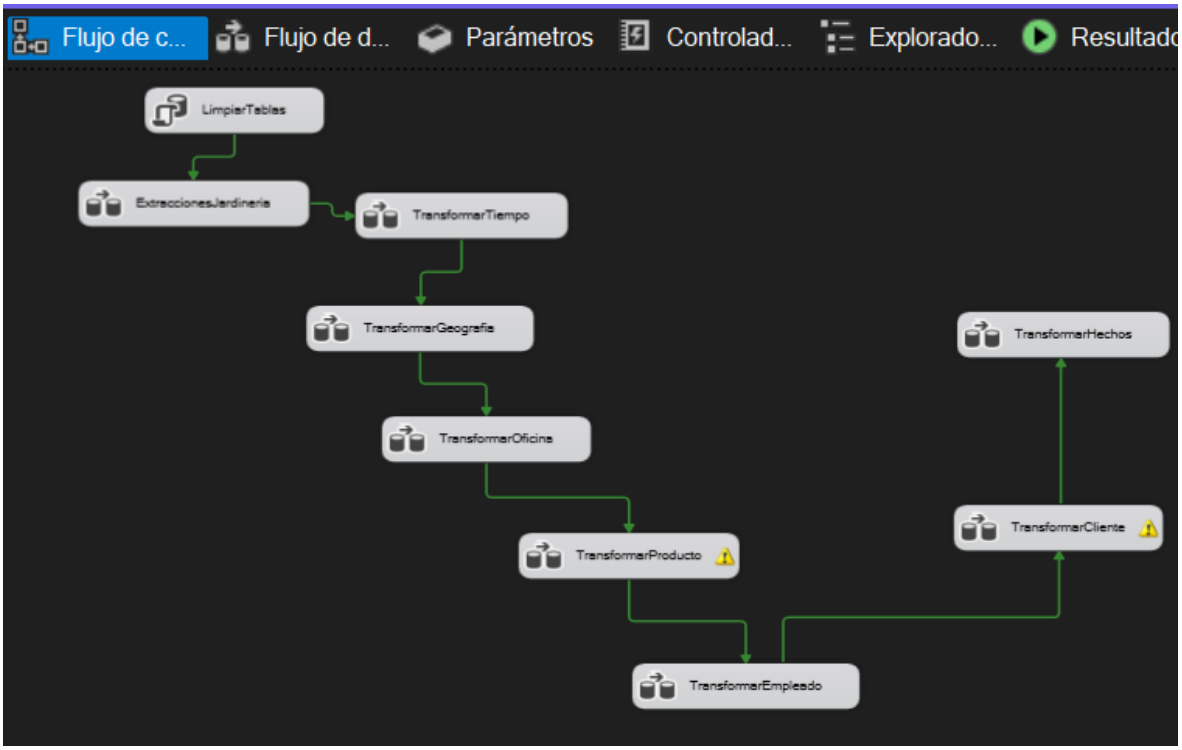
- **Tabla cliente:** Se descartaron dos columnas de esta tabla, ya que nos interesa la información únicamente de la persona que compra, por lo que se descarta el nombre y apellido del contacto, además también se prefirió no utilizar la columna de fax, puesto que en la actualidad no es de mucho uso.
- **Tabla Categoria_producto:** Se descartaron las columnas descripción_html e imagen, ya que no son utilizados.
- **Tabla Detalle_pedido:** Se descarta la columna numero_linea, ya que no es relevante como detalle del pedido para una tabla de hechos futura de Pedido.
- **Tabla Empleado:** Todas las columnas son relevantes, ya que tienen información personal y de contacto del empleado.
- **Tabla Oficina:** Se descarta únicamente la columna de línea_direccion2, puesto que sólo es relevante la dirección principal, junto con los otros datos de ubicación y contacto de las oficinas.
- **Tabla Pago:** Se seleccionaron todas las columnas, debido a que es importante la información del pago para una futura tabla de hechos de Pedido.
- **Tabla Pedido:** Se seleccionaron todas las columnas, debido a que es importante toda la información del pedido para una futura tabla de hechos de Pedido.
- **Tabla Producto:** Se seleccionaron todas las columnas, debido a que es importante toda la información del producto para una futura dimensión de Producto en conjunto con la tabla Categoria_producto
- **Tabla Tiempo:** Se selecciona únicamente la columna fecha_pedido de la tabla Pedido, para la dimensión futura de tiempo, que permita reemplazar las fechas que estarían en las demás dimensiones
-

2. Script de las consultas para crear la base de datos Staging

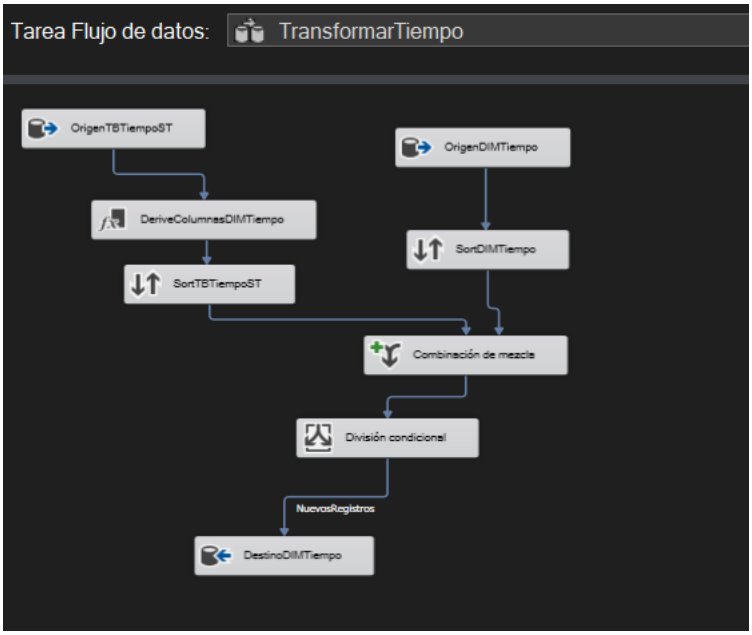
```
--TableCliente
SELECT C.ID_cliente,C.nombre_cliente, C.telefono, C.linea_direccion1,
C.linea_direccion2, C.ciudad, C.region, C.pais, C.codigo_postal,
C.ID_empleado_rep_ventas, C.limite_credito
FROM jardineria.dbo.cliente C
ORDER BY C.ID_cliente ASC

--TableCategoria_producto
SELECT CP.Id_Categoria,CP.Desc_Categoria,CP.descripcion_texto
FROM jardineria.dbo.Categoria_producto CP

--TableDetalle_pedido
SELECT
DP.ID_detalle_pedido,DP.ID_pedido,DP.ID_producto,DP.cantidad,DP.precio_unidad
FROM jardineria.dbo.detalle_pedido DP

--TableEmpleado
SELECT
E.ID_empleado,E.nombre,E.apellido1,E.apellido2,E.extension,E.email,E.ID_oficina,E.
ID_jefe,E.puesto
FROM jardineria.dbo.empleado E
ORDER BY E.ID_empleado ASC

--TableOficina
SELECT O.ID_oficina,O.Descripcion, O.ciudad,
O.pais,O.region,O.codigo_postal,O.telefono,O.linea_direccion1
FROM jardineria.dbo.oficina O
ORDER BY O.ID_oficina ASC

--TablePago
SELECT P.ID_pago,P.ID_cliente,P.forma_pago,P.id_transaccion,P.fecha_pago,P.total
FROM jardineria.dbo.pago P

--TablePedido
SELECT
Pd.ID_pedido,Pd.fecha_pedido,Pd.fecha_esperada,Pd.fecha_entrega,Pd.estado,Pd.comen
tarios,Pd.ID_cliente
FROM jardineria.dbo.pedido Pd

--TableProducto
SELECT
Pr.ID_producto,Pr.CodigoProducto,Pr.nombre,Pr.Categoria,Pr.dimensiones,Pr.proveedo
r,Pr.descripcion,Pr.cantidad_en_stock,Pr.precio_venta,Pr.precio_proveedor
FROM jardineria.dbo.producto Pr

--Table Tiempo
SELECT T.fecha_pedido
FROM jardineria.dbo.pedido T
ORDER BY 1 ASC
```

Proceso de transformación de datos

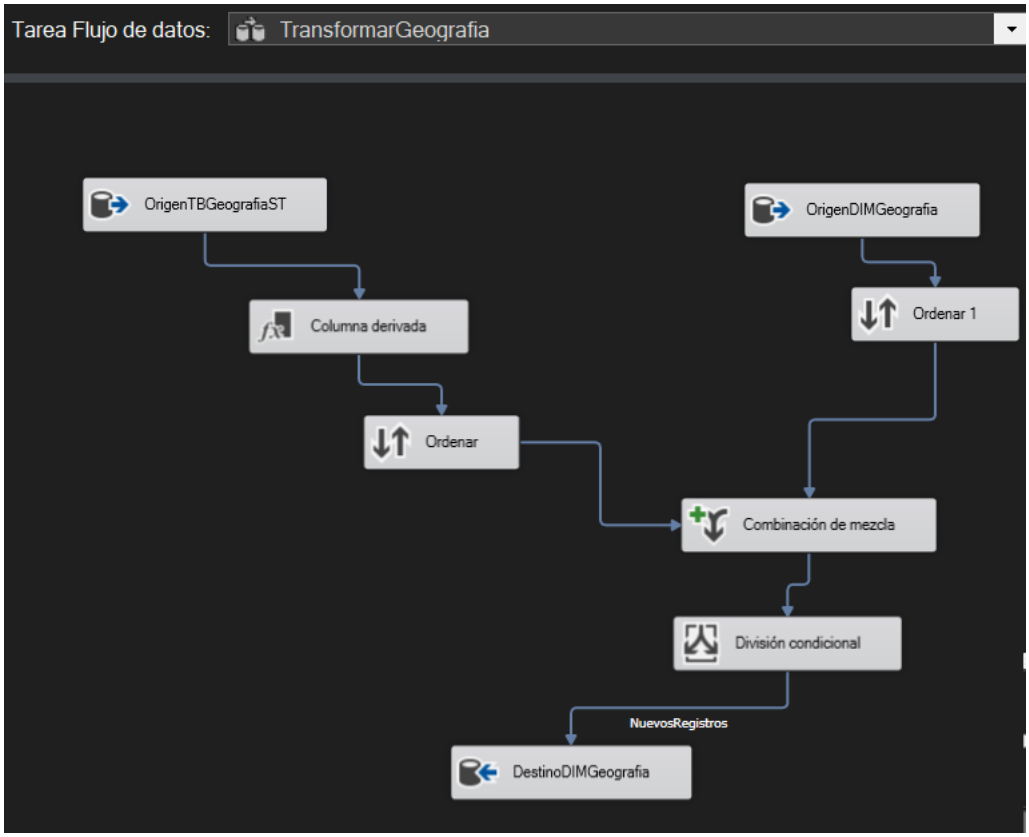
- **Flujo de control**



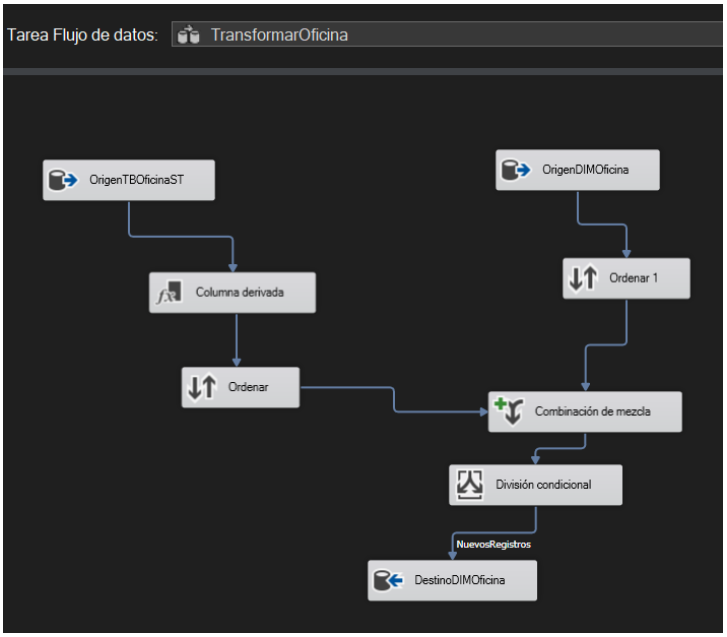
- **Transformación Tiempo**



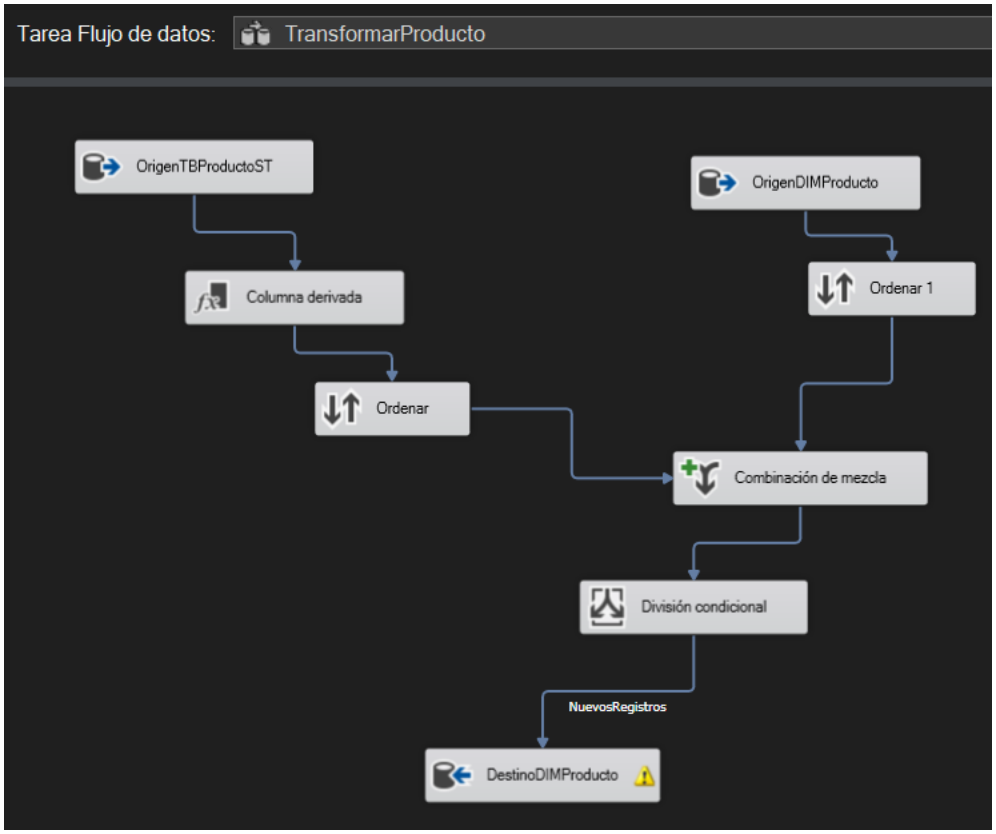
- **Transformación Geografía**



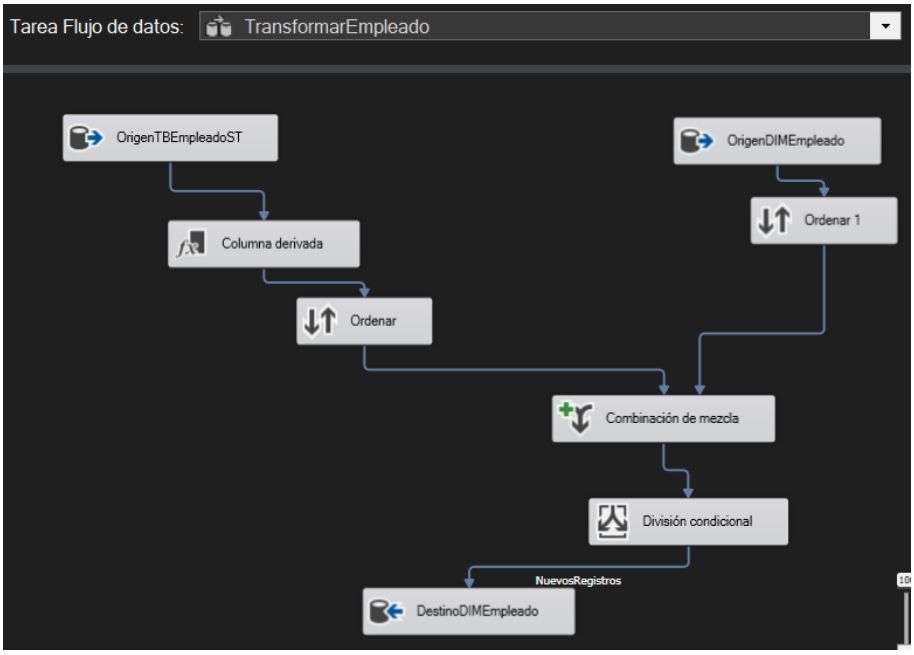
- **Transformación Oficina**



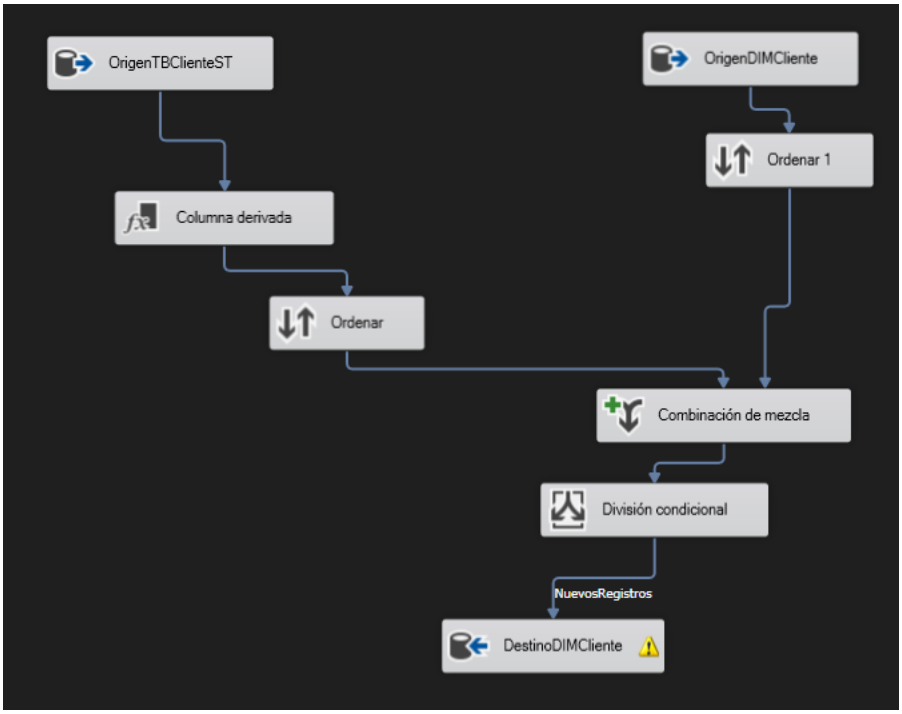
- **Transformación Producto**



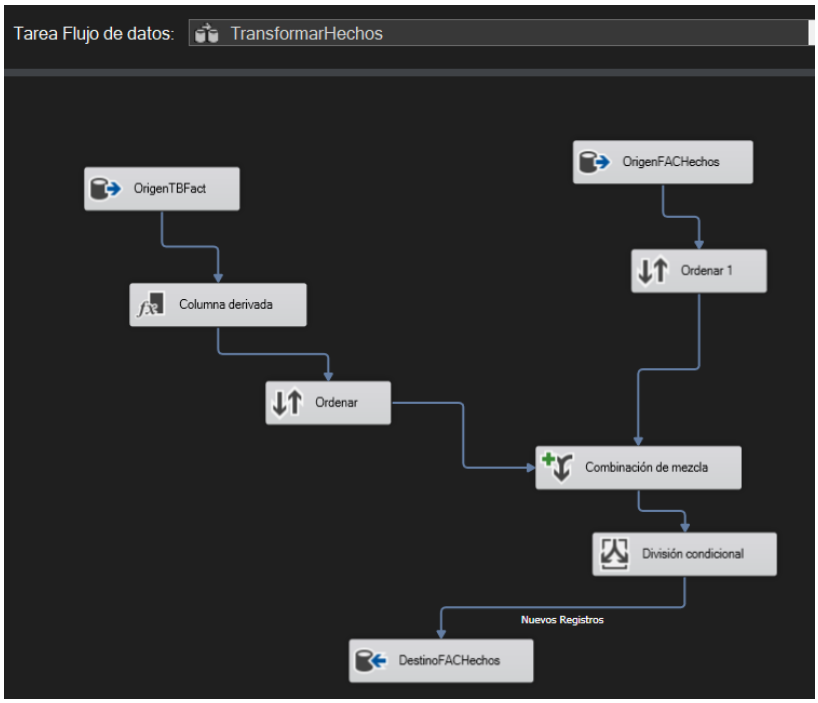
- **Transformación Empleado**



- **Transformación Cliente**

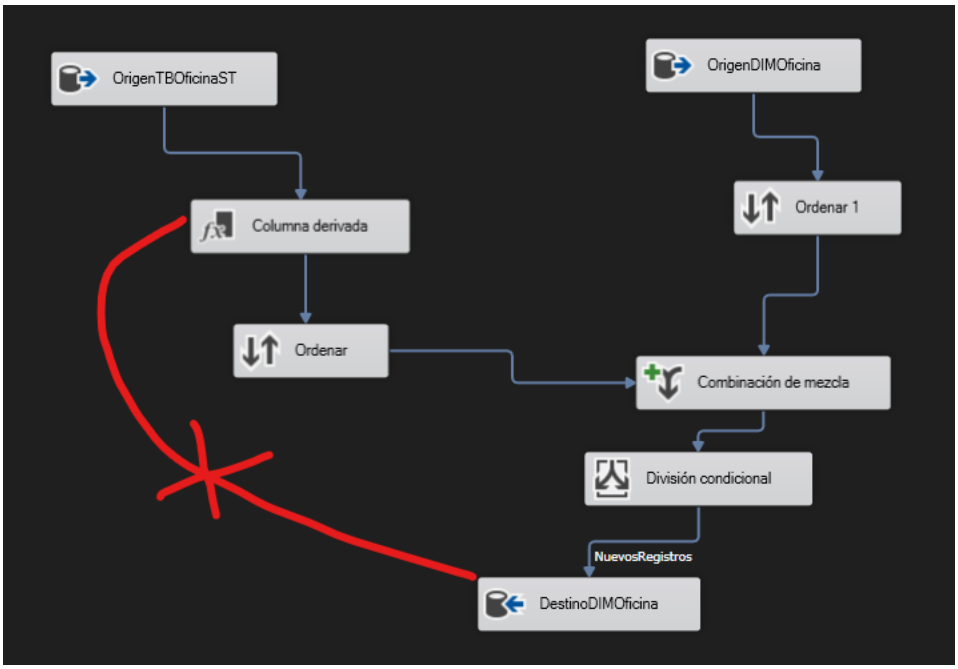


- **Transformación Tabla de Hechos**



Observaciones

- ✓ Las únicas dimensiones que generé por medio de SQL Server Integration Services (SSIS) fueron las de tiempo y geografía
- ✓ Previamente, fue necesario generar una tabla Geografia en la base de datos Staging, extrayendo campos de otras tablas como Oficina y Cliente por medio de la extracción de datos.
- ✓ Las demás dimensiones fueron generadas a través de consultas SQL, por medio de INNER JOIN, LEFT JOIN
- ✓ El proceso para generar cada una de las dimensiones consistía en tomar como origen la tabla correspondiente de la BD Staging, luego por medio de Columna Derivada se transformaban los valores de algunas columnas (mayúsculas, cambiarlo en caso de ser NULL,etc.), para luego crear la dimensión. Luego se borraba la relación entre la Columna Derivada y el Destino de la dimensión, para ingresar un nuevo origen, en base a la dimensión ya creada, después se ordenaban ambos sectores, generalmente, de forma ascendente de acuerdo al id de cada tabla, para continuar se seleccionaba la opción de Combinación de mezcla, para la comparación de ambos lados desde la izquierda, y seleccionando las columnas transformadas y descartando las originales, además de poner un id adicional para la tabla, que permitirá llevar control de los registros nuevos que vayan ingresando en cada tabla, esto en conjunto con un elemento de división condicional que realiza el filtro de los nuevos valores, para que después, al relacionarla con la dimensión, se puedan cargar estos registros nuevos. Se ve más detallado en el siguiente ejemplo de la dimensión Oficina:



Consultas SQL

A continuación se muestran algunas de las consultas SQL que se utilizaron para generar las dimensiones:

- ✓ **Dimensión Oficina:**

```
SELECT
O.ID_oficina,O.ID_oficina_O,G.id_geografia,O.Descripcion,O.telefono,O.linea_direccion1
FROM TBOficinaST O INNER JOIN DestinoTBGeografiaST G
ON O.ciudad=G.ciudad AND O.codigo_postal=G.codigo_postal AND O.pais=G.pais AND

O.region=O.region
```

- ✓ **Dimensión Producto:**

```
SELECT P.ID_producto,P.ID_producto_O,P.CodigoProducto,P.nombre,C.Desc_Categoria as
Categoria,P.dimensiones,P.proveedor,P.descripcion,P.cantidad_en_stock,P.precio_venta,P.pr
ecio_proveedor
FROM TBProductoST P INNER JOIN TBCategoriaProductoST C
ON P.Categoria=C.Id_Categoria
```

- ✓ **Dimensión Empleado**

```
SELECT E.ID_empleado,E.ID_empleado_O,(ISNULL(E.nombre,' ')+' '+ISNULL(E.apellido1,' ')+'
'+ISNULL(E.apellido2, ' ')) as nombre,E.extension,E.email, O.Descripcion as oficina,
(ISNULL(J.nombre,' ')+' '+ISNULL(J.apellido1,' ')+' '+ISNULL(J.apellido2, ' ')) as
jefe,E.puesto
FROM TBEmpleadoST E
LEFT JOIN TBEmpleadoST J ON E.ID_jefe=J.ID_empleado
INNER JOIN TBOficinaST O ON E.ID_oficina=O.ID_oficina
```

- ✓ **Dimensión Cliente**

```
SELECT
C.ID_cliente,C.ID_cliente_O,C.nombre_cliente,C.telefono,C.linea_direccion1,C.linea_direcc
ion2,G.id_geografia as id_geografia,(ISNULL(E.nombre,' ')+' '+ISNULL(E.apellido1,' ')+'
'+ISNULL(E.apellido2,' ')) as vendedor,C.limite_credito
FROM TBClienteST C
INNER JOIN DestinoTBGeografiaST G ON C.ciudad=G.ciudad AND
C.codigo_postal=G.codigo_postal AND C.pais=G.pais AND C.region=G.region
INNER JOIN TBEmpleadoST E ON C.ID_empleado_rep_ventas=E.ID_empleado
```

- ✓ **Tabla de Hechos**

```
SELECT P.ID_pedido,P.ID_pedido_O,T.ID_tiempo as fecha_pedido,C.ID_cliente as
id_cliente,Pr.ID_producto as id_producto,E.ID_empleado as id_empleado,O.ID_oficina as
id_oficina,
P.estado,P.comentarios,D.cantidad, D.precio_unidad
FROM TBPedidoST P
INNER JOIN TBTiempoST T ON P.fecha_pedido=T.fecha_pedido
INNER JOIN TBClienteST C ON P.ID_cliente=C.ID_cliente
INNER JOIN TBDetallePedidoST D ON P.ID_pedido=D.ID_pedido
INNER JOIN TBProductoST Pr ON Pr.ID_producto=D.ID_producto
INNER JOIN TBEmpleadoST E ON E.ID_empleado=C.ID_empleado_rep_ventas
INNER JOIN TBOficinaST O ON O.ID_oficina=E.ID_oficina
```

Anexos

Enlace Modelo estrella: https://lucid.app/lucidchart/3c255804-6a57-4eea-996e-ef63ef2ba8a2/edit?viewport_loc=-1321%2C-1471%2C4743%2C1913%2C0_0&invitationId=inv_89f14898-1944-41db-a0e9-67e5baee3515