

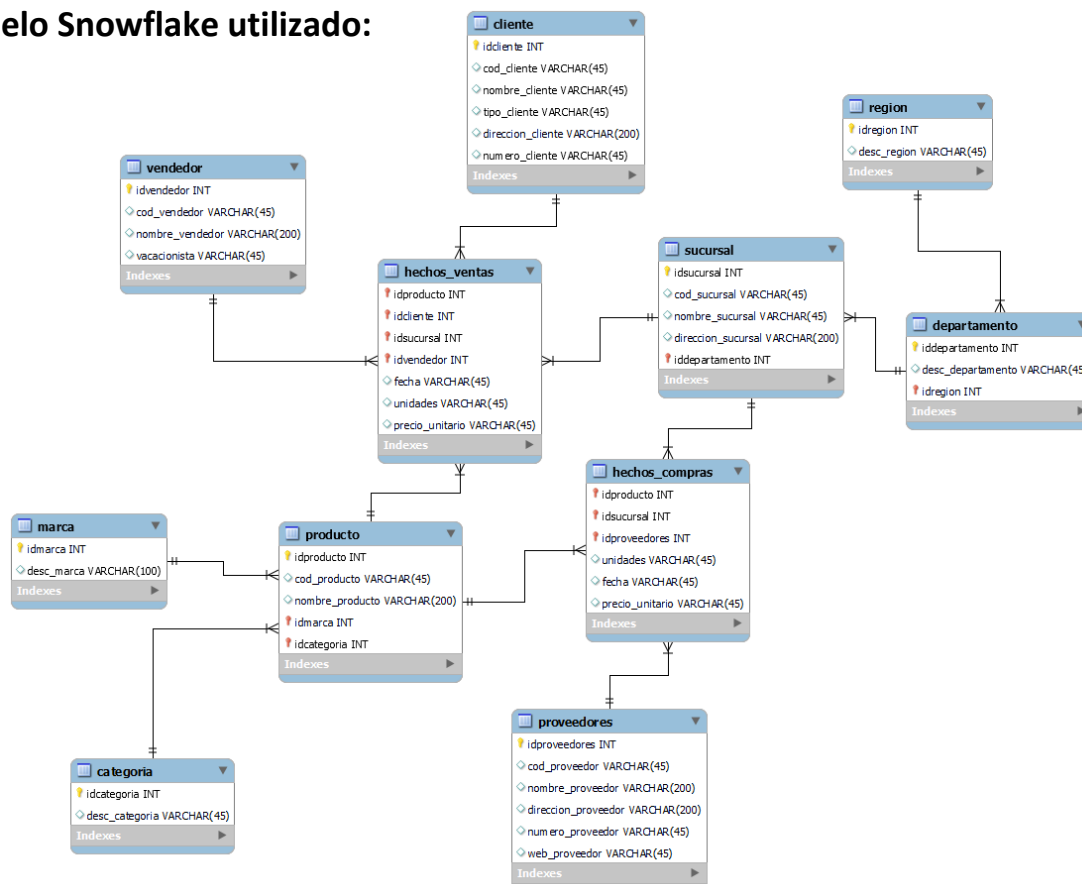
SG_Food

Bajo un analisis previo sobre el Desarrollo del problema y sobre los resultados que la empresa requiere para cumplir sus necesidades, se determino que es necesaria la creacion de un modelo que complazca toda la informacion ingresada, de manera que esta pueda ser manejada por una base de datos dedicada a la recoleccion de los resultados en base al tiempo el cual se le denomina Datawarehouse.

Para la elaboracion de este Datawarehouse es necesario construir su modelo correspondiente para que de esta manera, se pueda concluir cuales son las dimensiones requeridas para solventar esta situacion y posteriormente tener los datos totales de una manera accesible y entendible de consultar.

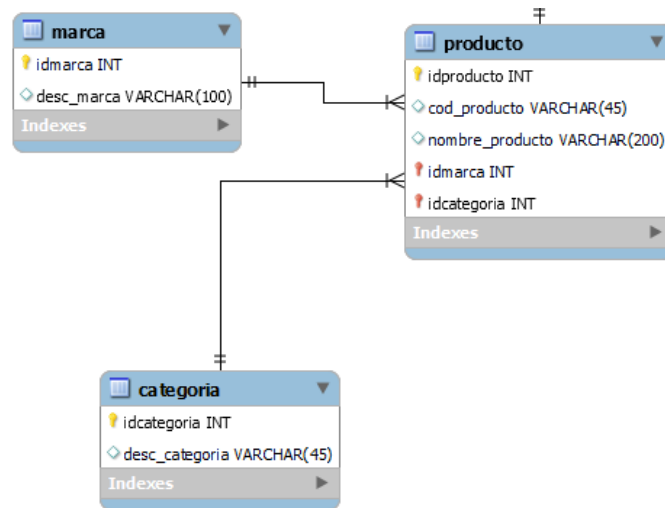
El modelo en el que se basara el datawarehouse correspondiente se le denomina de tipo snowflake o constelacion, la razon por la cual es necesaria la utilizacion de este modelo, es debido a que se debe de saber como es el rendimiento del departamento de ventas y del departamento de compras al mismo tiempo, dicha actividad puede ser conseguida mediante la creacion de dimensiones denominadas hechos que estaran enlazadas con las dimensiones de los datos que se requieran para poder llenar esta dimension de hechos correspondientes a las compras y ventas y asi poder visualizar de una mayor manera esas metricas que describen el rendimiento de la empresa a lo largo del tiempo.

Modelo Snowflake utilizado:



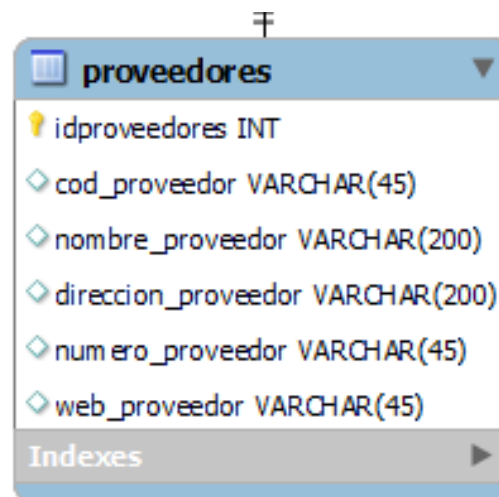
Dimensiones:

Producto → marca → categoria



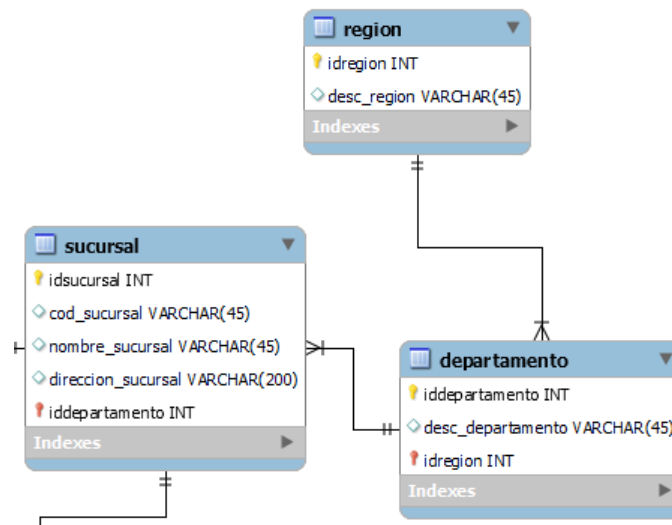
Esta dimension se encarga de encapsular los datos del product conteniendo lo que es sum arca y categoria correspondiente, la razon por la cual existen 2 dimensiones extra enlazadas con producto es por que se necesita cierta normalizacion para evitar un exeso de reduntancia entre datos y poder conseguir lecturas mas precisas.

Proveedores



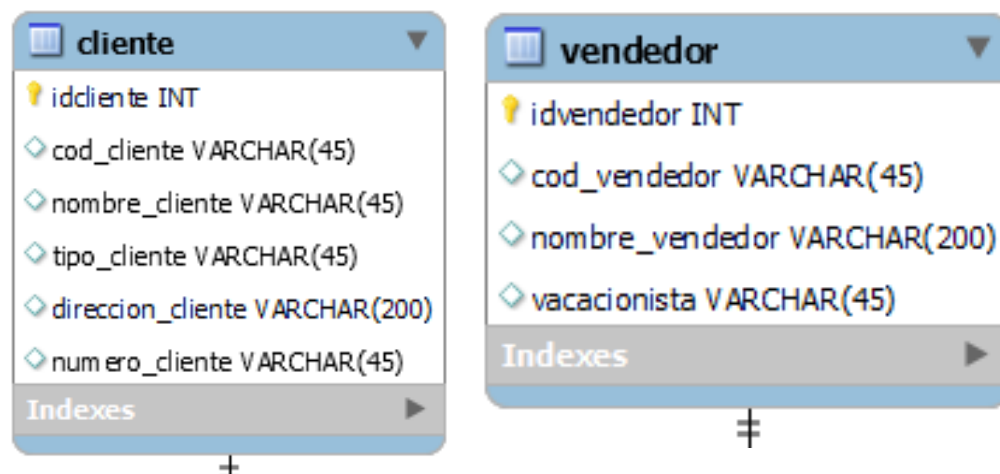
Dimension encargada de almacenar los datos de los proveedores

Sucursal -> departamento -> region



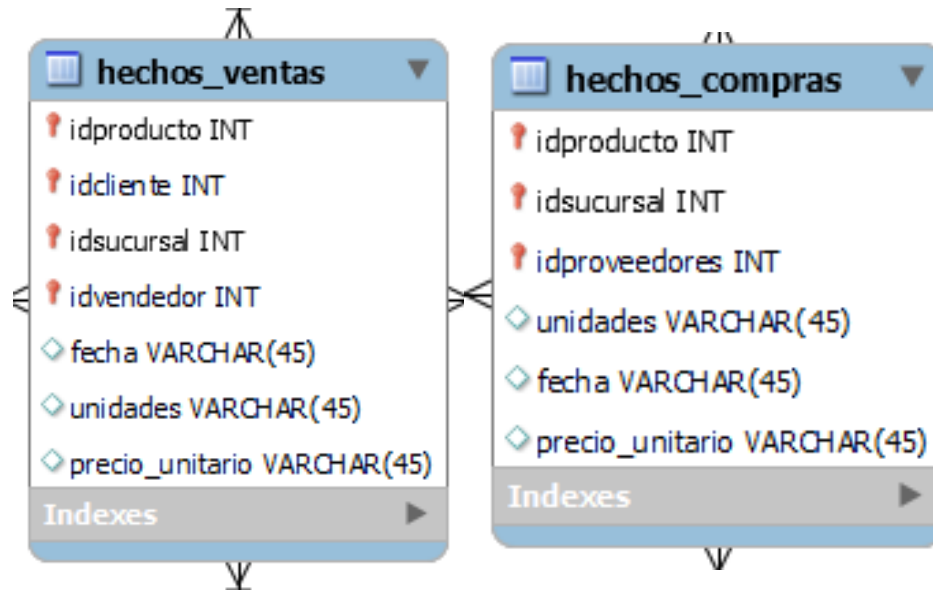
La dimension succursal esta enlazada en conjunto con las dimensiones de departamento y region correspondientes a la ubicacion de la tienda, esto con el objetivo de tener un dato exacto por cada tienda o sucursal que esta almacenada o registrada en el Sistema.

Cliente, Vendedor



Las dimensiones Cliente y Vendedor correspondientes unicamente almacenaran los datos normalizados de cada cliente y vendedor

Hechos_ventas, Hechos_compras



Estas dimensiones se encargan de almacenar las metricas correspondientes con cada una de las dimensiones que tengan una union directa con la informacion que se desea visualizar en base a metricas.

PROCESO ETL







Luego de armar lo que sera el modelo datawarehouse a utilizar, se procede a hacer la creacion y organizacion de tablas en los distintos origenes a los que se le desea almacenar temporalmente los datos para luego ser transformados, esa primera seccion se le llama extraccion.


Primero se deben crear lo que son las tablas pivote que contendran el area de stage en donde se recurrira para cargar los datos, en esta ocasion se requieren de 2 bases de datos, las dos llamadas **seminario2_201800987** ubicadas en MYSQL y SQL SERVER, dichas tablas se crean con el siguiente script sql:

```
create table temporal_general_v (
  fecha varchar(200),
  codigo_cliente varchar(200),
  nombre_cliente varchar(200),
  tipo_cliente varchar(200),
  direccion_cliente varchar(200),
  numero_cliente varchar(200),
  cod_vendedor varchar(200),
  nombre_vendedor varchar(200),
  vacacionista varchar(200),
  cod_producto varchar(200),
  nombre_producto varchar(200),
  marca_producto varchar(200),
  categoria varchar(200),
  sod_sucursal varchar(200),
  nombre_sucursal varchar(200),
  direccion_sucursal varchar(200),
  region varchar(200),
  departamento varchar(200),
  unidades varchar(200),
  precio_unitario varchar(200),
);

create table temporal_general_c (
  fecha varchar(200),
  codigo_proveedor varchar(200),
  nombre_proveedor varchar(200),
  direccion_proveedor varchar(200),
  numero_proveedor varchar(200),
  web_proveedor varchar(200),
  cod_producto varchar(200),
  nombre_producto varchar(200),
  marca_producto varchar(200),
  categoria varchar(200),
  sod_sucursal varchar(200),
  nombre_sucursal varchar(200),
  direccion_sucursal varchar(200),
  region varchar(200),
  departamento varchar(200),
  unidades varchar(200),
  precio_unitario varchar(200),
);
```

Estas tablas seran pobladas con archivos de entrada con extension .sg que se cargaran con un proceso que se explicara posteriormente

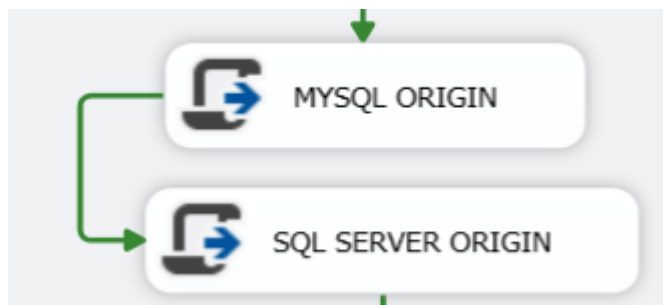
-  SGFood01.comp
-  SGFood01.vent
-  SGFood02.comp
-  SGFood02.vent
-  SGFood03.comp
-  SGFood03.vent

Por ultimo se procede a crear lo que es el modelo snowflake que contendra todos los datos en SQL SERVER, dicho modelo se ejecuta con el script  `model_script_server` contenido en la raiz de este Proyecto.

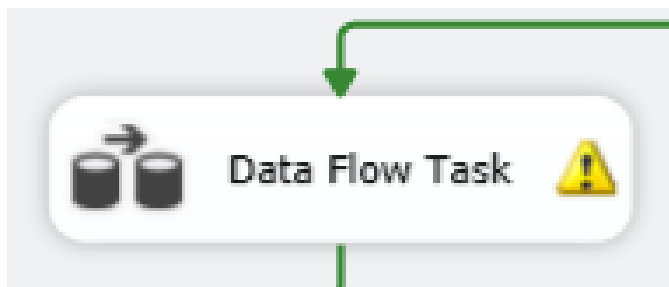
Para realizar todo este proceso ETL se requiere de la utilizacion de las herramientas SSIS proporcionadas por Visual Studio 2019, este Proyecto se comprondra de:



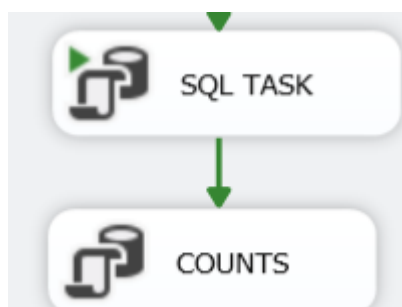
2 componentes T-SQL STATEMENT los cuales funcionaran como ejecutores de scripts garantizando la entrada fluida de los nuevos datos que se requieran ingresar, cada statement realiza la funcion correspondiente a su titulo.



2 componentes SCRIPT TASK los cuales mediante un script codificado en C# se encargaran de obtener los datos de los archivos .sg que se desean cargar y posteriormente introducirlos en las tablas pivotte anteriormente descritas.

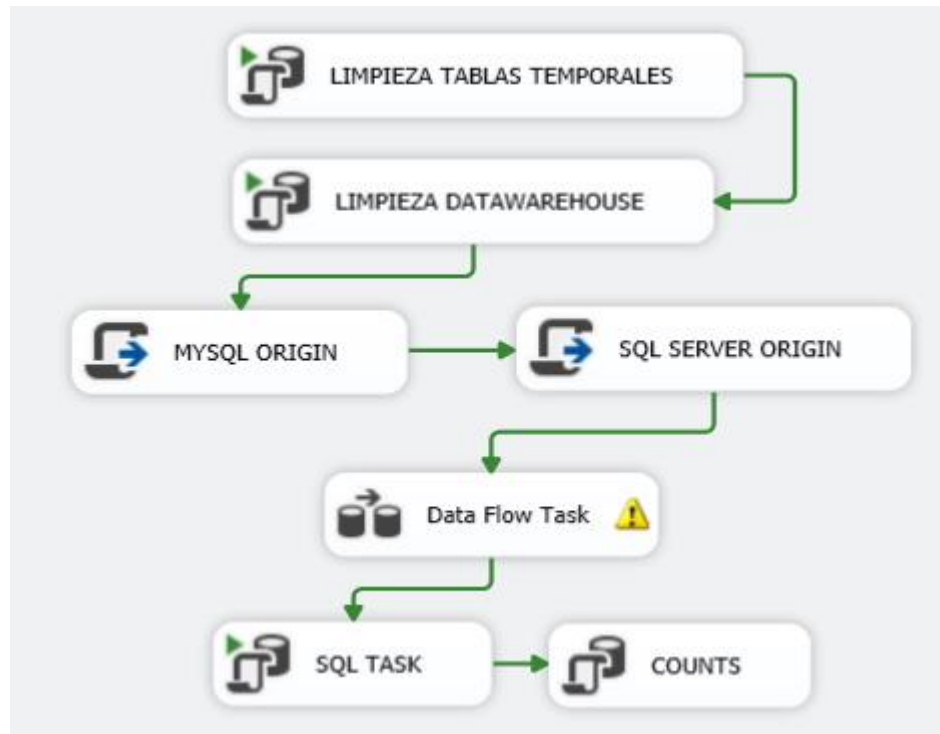


2 componente DATA FLOW TASK encargado de realizar el proceso de transformacion y carga de los datos hacia las tablas dimensionales creadas en el datawarehouse.



2 componentes T-SQL STATEMENT que se encargan de enlazar las dimensiones correspondientes con sus debidas llaves subrogadas y con un script que al ser ejecutado, mostrara lo que son la cuenta de registros introducidos en el datawarehouse.

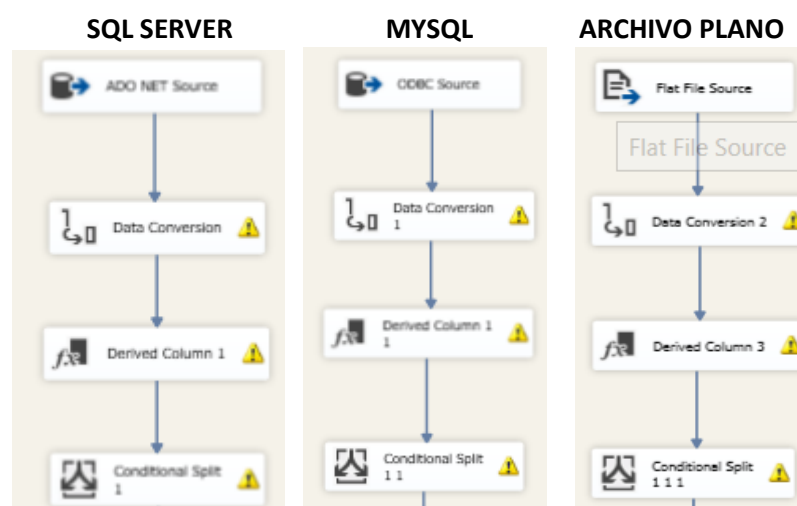
VISTA GENERAL ETL:



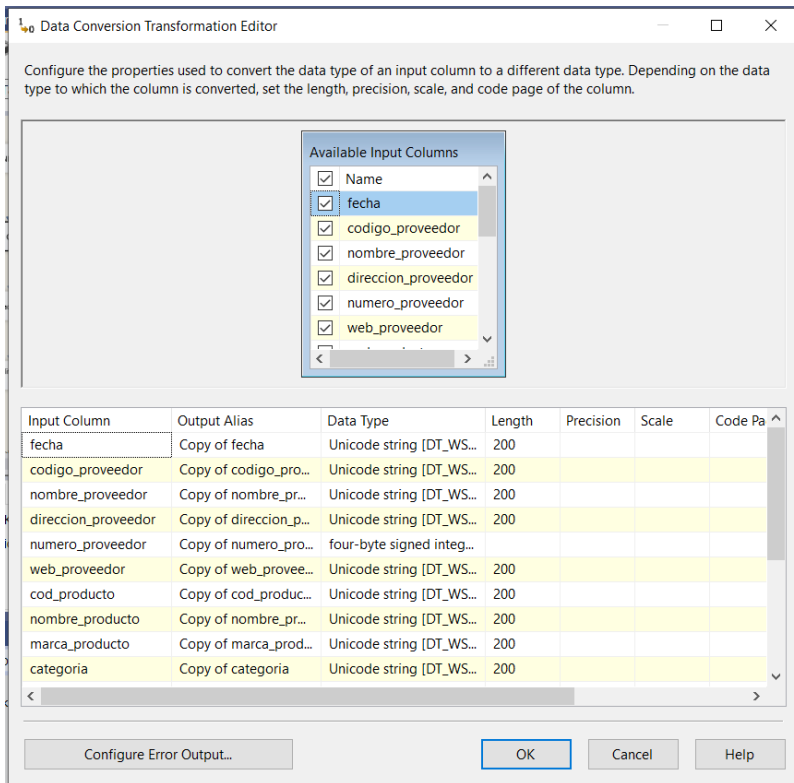
Proceso de transformacion

Esta seccion es la mas larga y tardada ya que en esta es donde se realizara el parseo de la informacion para que esta no tenga problemas al ser introducida en el datawarehouse.

La transformacion se llevo a cabo de la siguiente manera, en donde se separaron 3 componentes, 1 que recopila info del destino MYSQL , 1 que Tambien recopila solo que desde SQL SERVER y 1 que se encarga de cargar los datos directamente de un archivo plano, este proceso se muestra de la siguiente manera:

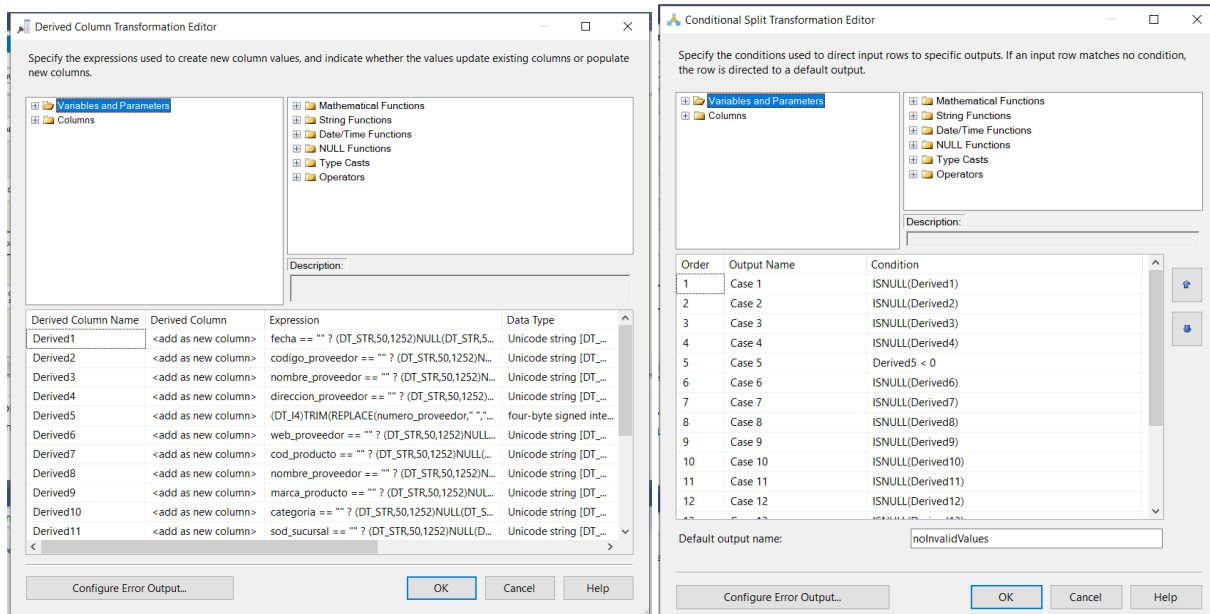


Cada transformacion se lleva a cabo en el componente Data Conversion en donde se especifica hacia que tipo de valor el registro ingresado sera parseado

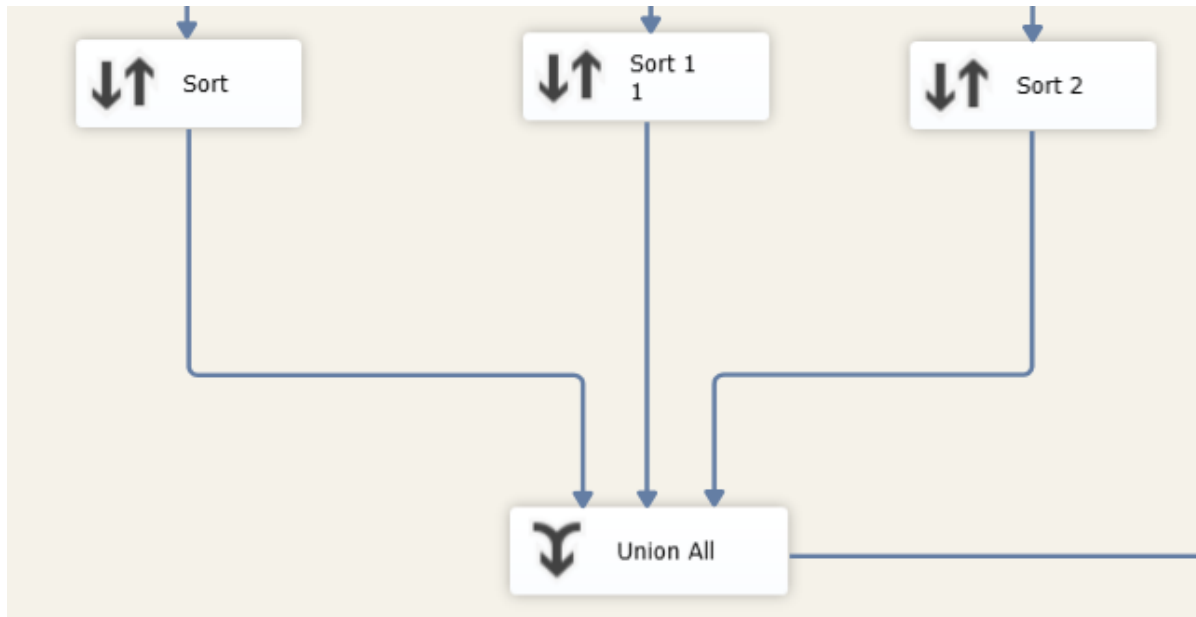


Como se puede apreciar, en primera instancia se seleccionaron todas las columnas de origen de cada componente y se crea una copia de ellas conteniendo el tipo de dato al cual sera parseado, dicha metodologia se aplico para los 3 origenes

Posteriormente se aplica una limpieza de los datos para evitar trabajar con datos nulos o incongruentes, dicha funcionalidad se lleva a cabo en los componentes Derived Column en donde se especifican las columnas a ser evaluadas y un Conditional Split en donde se valida que esa columna contenga informacion valida para procesar, la configuracion es la siguiente para cada componente:



Por ultimo se enlaza a un componente sort, el cual eliminara registros duplicados en cada uno de los origenes y posteriormente unira todos esos registros con la herramienta Union All en donde como su palabra lo dice, sacara todos los registros de las tablas enlazados entre si.



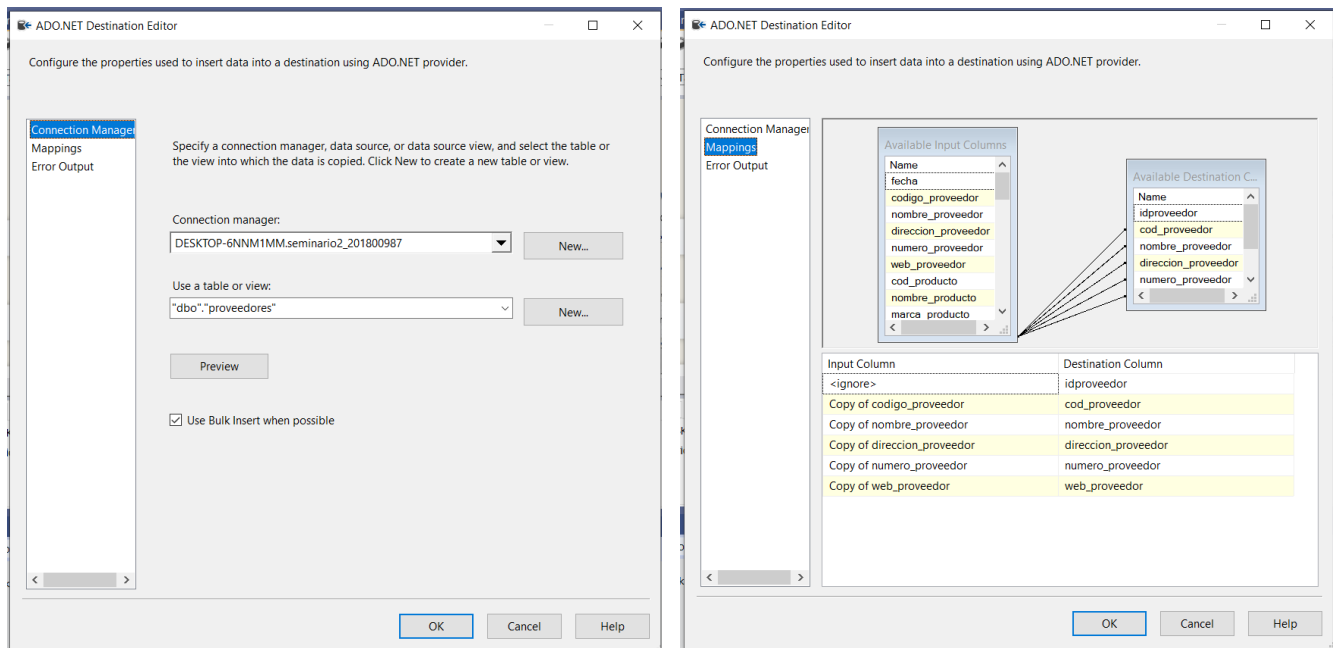
Proceso de carga

Este es el proceso mas largo, debido a que aqui es donde se separan los datos por dimensiones y se registran en el modelo datawarehouse.

En resumidas cuentas este proceso se lleva a cabo por la separacion de varios Multicast y Sorts los cuales funcionaran como direccionadores especificos de informacion los cuales tendran como destino una tabla en especifico del data warehouse.

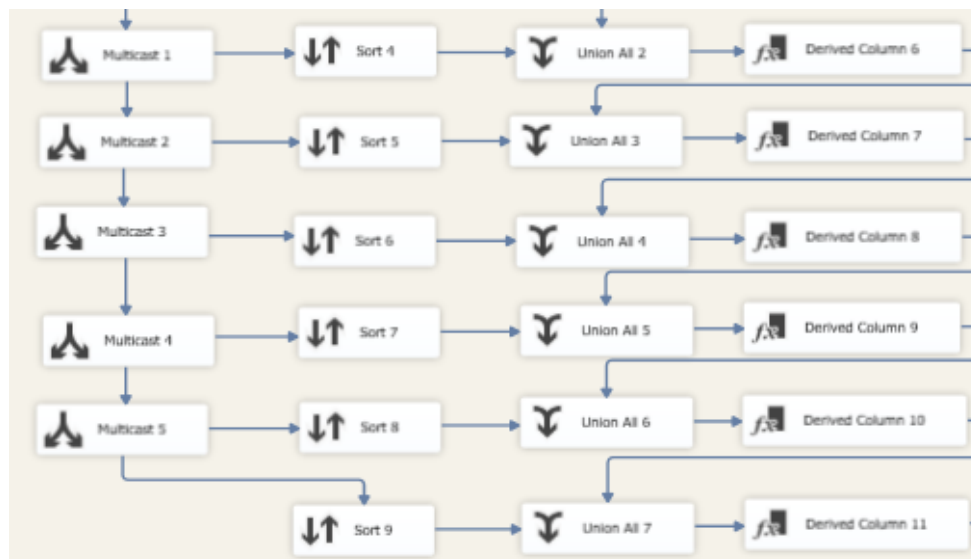


Cada dimension fue registrada de la siguiente manera en donde El Derived Column se encarga de limpiar cada registro que desee ser ingresado y en el Sort se eliminan duplicados, por ultimo asi como se mencionaba con anterioridad, esos tienen un destino a una base de datos, en este caso se desea ingresar los datos a la tabla que esta especificada en el componente ADO NET destination el cual se configura de la siguiente manera:



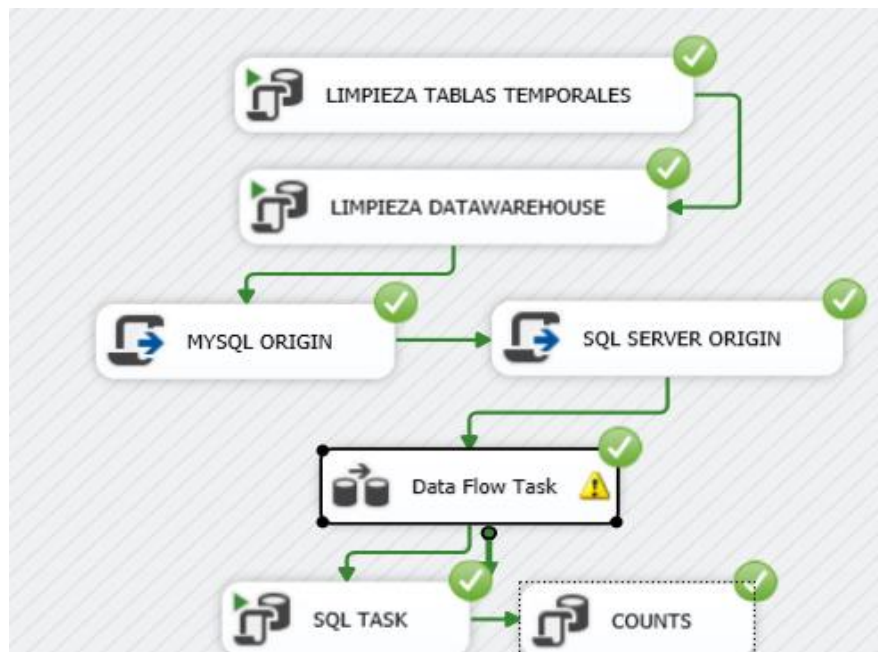
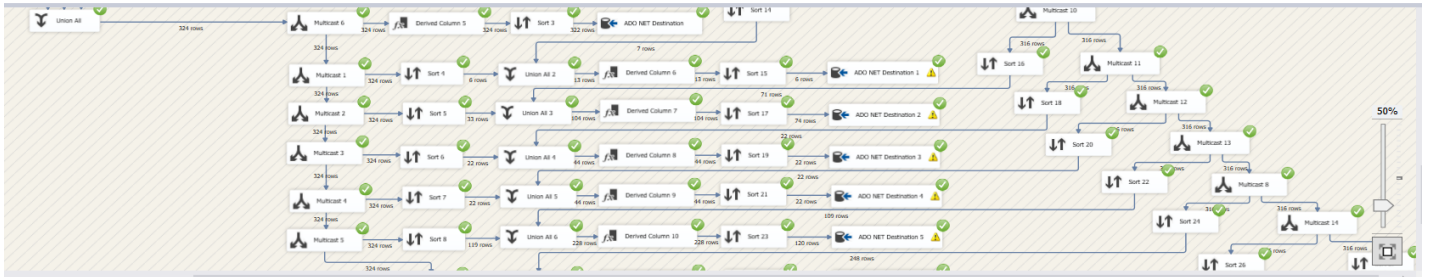
Para cada destino se debe realizar un mapeo de columnas en donde se especifica que columnas son las que contienen los registros y hacia que columnas en la tabla del data warehouse esos deben ser introducidos.

VISTA GENERA DE LA CARGA DE INFORMACION PARA LAS DIMENSIONES



Para realizar todo ese proceso en tiempo real basta con correr el modelo y esperar si fue exitosa la ejecucion.

VISTA GENERAL DE EJECUCION



Como se aprecia, Visual Studio indica que el proceso ETL se ejecuto de manera exitosa por lo que se puede proceder a comprobar los resultados del mismo.

Lo que queda solamente verificar que si fueron cargados los datos, a continuacion se haran consultas count hacia las tablas para verificar si constan de registros almacenados:

```
select count(*) as "TOTAL PROVEEDORES" from proveedores;  
select count(*) as "TOTAL CLIENTES" from cliente;  
select count(*) as "TOTAL VENDEDORES" from vendedor;  
select count(*) as "TOTAL PRODUCTOS" from producto;  
select count(*) as "TOTAL SUCURSALES" from sucursal;  
select count(*) as "TOTAL COMPRAS" from hechos_compras;  
select count(*) as "TOTAL VENTAS" from hechos_ventas;
```

	TOTAL PROVEEDORES
1	322
	TOTAL CLIENTES
1	294
	TOTAL VENDEDORES
1	60
	TOTAL PRODUCTOS
1	229
	TOTAL SUCURSALES
1	120
	TOTAL COMPRAS
1	324
	TOTAL VENTAS
1	319

Dados los resultados correspondientes con esto se concluye que el proceso ETL se ejecuto exitosamente y se comprueba que el modelo de datawarehouse aplicado al mismo fue hecho de la manera ideal para que pueda cumplir con un correcto funcionamiento en la base de datos donde se encuentra almacenado el cual es SQL SERVER.