

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Laboratorio de Seminario de Sistemas

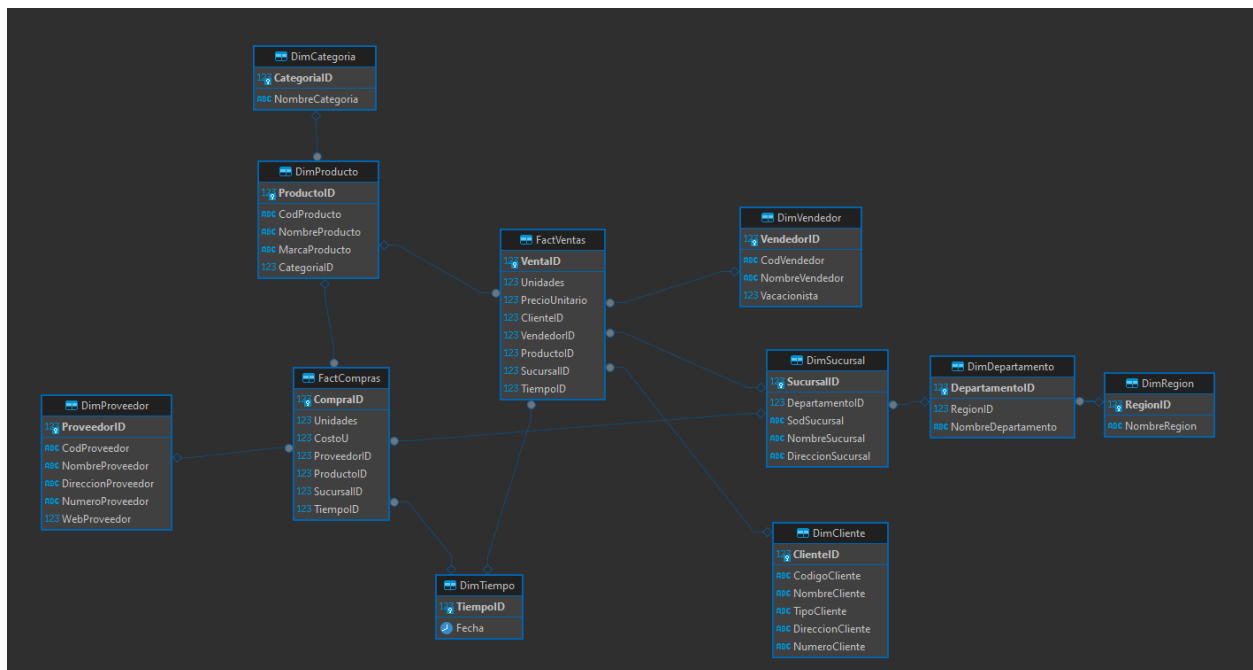
Jorge Antonio Pérez Ordóñez - 201900810

Proyecto: Fase 1

Nombre

SG-Food DW

Modelo del Data Warehouse



El modelo del Data Warehouse sigue la estructura de un esquema de tipo constelación. Es de tipo constelación porque maneja dos tablas de hechos, una para compras y otra para ventas. Las tablas de hechos tienen algunas dimensiones en común y otras que son únicas para cada tabla de hechos. En total hay 6 dimensiones y algunas tienen más de un nivel por los que se podrán filtrar en el momento de análisis de datos.

Tabla Dimensional DimTiempo

La tabla DimTiempo hace referencia a la fecha en la que ocurrió el hecho. Es una dimensión de un solo nivel y su tabla tiene la llava primaria y su fecha correspondiente.

Nombre	Tipo
TiempoID	INT
Fecha	DATE

Tabla Dimensional DimProveedor

La tabla DimProveedor hace referencia al proveedor del cual se compraron los productos. Es una dimensión de un solo nivel y aparece solo en la tabla de FactCompras.

Nombre	Tipo
ProveedorID	INT
CodProveedor	CHAR(5)
NombreProveedor	NVARCHAR(255)
DireccionProveedor	NVARCHAR(255)
NumeroProveedor	CHAR(8)
WebProveedor	BOOL

Tabla Dimensional DimCategoria

La tabla DimCategoria hace referencia a las categorías en las que pueden estar los productos. Esta tabla es un nivel de la dimensión producto.

Nombre	Tipo
CategorialID	INT
NombreCategoria	NVARCHAR(255)

Tabla Dimensional DimProducto

La tabla DimProducto hace referencia a los productos que aparecen en las compras y ventas de la organización. Esta dimensión cuenta con el nivel de categoría.

Nombre	Tipo
ProductID	INT
CodProducto	CHAR(8)
NombreProducto	NVARCHAR(255)
MarcaProducto	NVARCHAR(255)
CategorialID	INT

Tabla Dimensional DimCliente

La tabla DimCliente hace referencia a los clientes a los que se les venden los productos. Esta es una dimensión de un solo nivel.

Nombre	Tipo
ClienteID	INT
CodigoCliente	CHAR(5)
NombreCliente	NVARCHAR(255)
TipoCliente	NVARCHAR(255)
DireccionCliente	NVARCHAR(255)
NumeroCliente	CHAR(8)

Tabla Dimensional DimVendedor

La tabla DimVendedor hace referencia a los vendedores que ejecutan la venta de los productos a los clientes. Es una dimensión de un solo nivel.

Nombre	Tipo
VendedorID	INT
CodigoVendedor	CHAR(5)
NombreVendedor	NVARCHAR(255)
Vacacionista	BOOL

Tabla Dimensional DimRegion

La tabla DimRegion hace referencia a las regiones en donde se encuentran las sucursales. Es una nivel de la dimensión Sucursal.

Nombre	Tipo
RegionID	INT
NombreRegion	NVARCHAR(255)

Tabla Dimensional DimDepartamento

La tabla DimDepartamento hace referencia a los departamentos en donde se encuentran las sucursales. Es un nivel de la dimensión sucursal.

Nombre	Tipo
DepartamentID	INT
NombreDepartamento	NVARCHAR (255)

Tabla Dimensional DimSucursal

La tabla DimSucursal hace referencia a las sucursales en donde se hacen las compras de productos y las ventas de estos. Es una dimensión con 2 niveles.

Nombre	Tipo
SucursalID	INT
SodSucursal	CHAR(5)
Nombre Sucursal	NVARCHAR(255)
DireccionSucursal	NVARCHAR(255)
DepartamentoID	INT

Tabla de Hechos FactCompras

Nombre	Tipo
ComprasID	INT
CostoU	FLOAT
Unidades	INT
FechaID	INT
ProveedorID	INT
SucursalID	INT
ProductoID	INT

Tabla de Hechos FactVentas

Nombre	Tipo
VentasID	INT
PrecioUnitario	FLOAT
Unidades	INT
FechaID	INT
VendedorID	INT
SucursalID	INT
ProductoID	INT
ClienteID	INT

Proceso ETL

Para llevar a cabo el proceso de ETL desde los archivos fuente hasta el modelo del Data Warehouse, se utilizó SQL Server Integration Services. El siguiente esquema muestra el flujo de control por el que pasan las etapas del proceso ETL.



1. Eliminar Temporales SQL Server

En este paso, se eliminan todos los registros que existan en las tablas pivote de SQL Server con el propósito de que las tablas estén vacías para empezar a poblarlas y que no haya registros repetidos. En las tablas de SQL Server se encuentran los registros de los archivos SGFood01.cop y SGFood01.vent.

2. Eliminar Temporales MySQL

En este paso, se eliminan todos los registros que existan en las tablas pivote de MySQL con el propósito de que las tablas estén vacías para empezar a poblarlas y que no haya

registros repetidos. En las tablas de SQL Server se encuentran los registros de los archivos SGFood02.cop y SGFood02.vent.

3. Eliminar Modelo

En este paso se eliminan todos los registros existentes en las tablas del Data Warehouse con el propósito de que se encuentren vacías antes de empezar a poblarlas y que no haya registros repetidos.

4. Exportar a Tablas Pivote

En este paso, se hace la parte de extracción de datos de los archivos SGFood.comp y SGFood.vent. Los registros se obtienen tal y como se encuentran en los archivos y se insertan en tablas pivote dentro de las bases de datos SQL Server y MySQL por medio de un script en C#.

```
0 references
public void Main()
{
    // TODO: Add your code here
    try
    {
        string delimitador = Dts.Variables["User::delimitador"].Value.ToString();
        string extensionVentas = Dts.Variables["User::extensionVentas"].Value.ToString();
        string extensionCompras = Dts.Variables["User::extensionCompras"].Value.ToString();
        string folderErrores = Dts.Variables["User::folderErrores"].Value.ToString();
        string folderOrigen = Dts.Variables["User::folderOrigen"].Value.ToString();
        string tablaTempVentas = Dts.Variables["User::tablaTempVentas"].Value.ToString();
        string tablaTempCompras = Dts.Variables["User::tablaTempCompras"].Value.ToString();

        // Obtener rutas
        string rutaCompra1 = Directory.GetFiles(folderOrigen, "SGFood01" + extensionCompras)[0];
        string rutaVenta1 = Directory.GetFiles(folderOrigen, "SGFood01" + extensionVentas)[0];

        string rutaCompra2 = Directory.GetFiles(folderOrigen, "SGFood02" + extensionCompras)[0];
        string rutaVenta2 = Directory.GetFiles(folderOrigen, "SGFood02" + extensionVentas)[0];

        string sqlServerString = "localhost.jorge";
        string mysqlString = "localhost.semi2proyecto.root";

        string[] compras = { rutaCompra1, rutaCompra2 };
        string[] ventas = { rutaVenta1, rutaVenta2 };
        string[] conexiones = { sqlServerString, mysqlString };

        for (int idx = 0; idx < 2; idx++)
        {
            SqlConnection sqlServerConn = null;
            MySqlConnection myConn = null;

            if (idx == 0)
            {
                sqlServerConn = new SqlConnection();
                sqlServerConn = (SqlConnection)(Dts.Connections[conexiones[0]].AcquireConnection(Dts.Transaction) as SqlConnection);
            }
            else
            {
                myConn = (MySqlConnection)(Dts.Connections[conexiones[1]].AcquireConnection(Dts.Transaction) as MySqlConnection);
            }

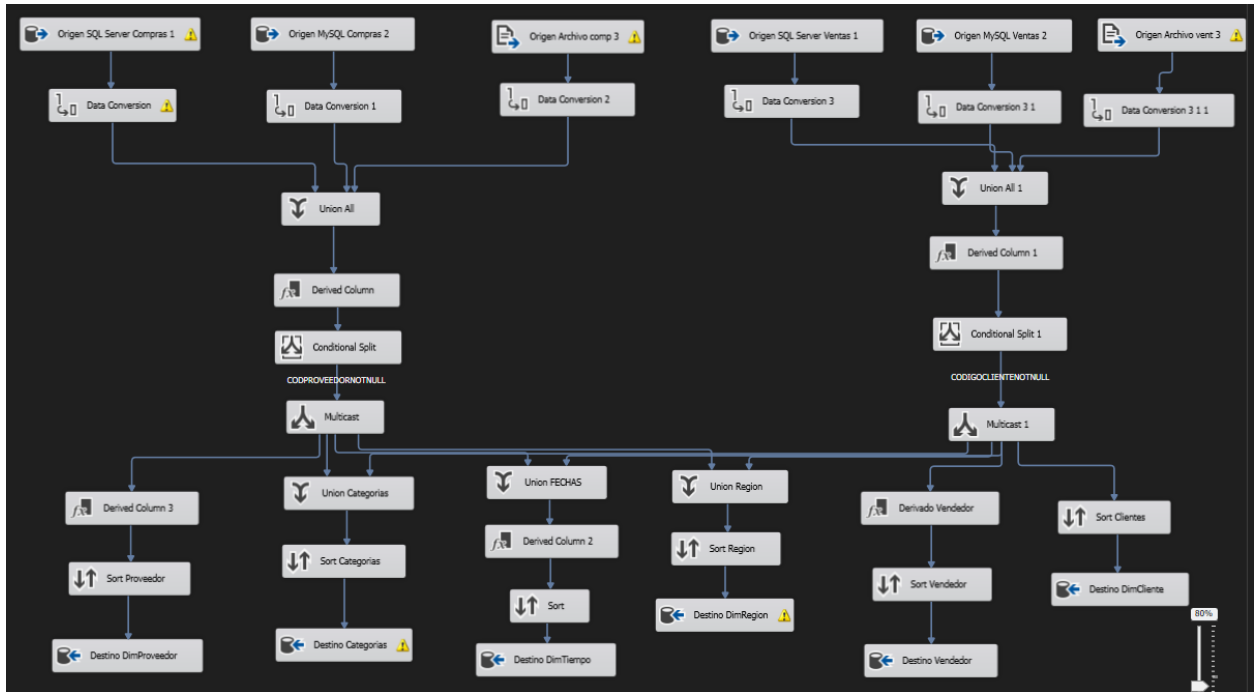
            // Leer compras
            int contadorCompras = 0;
            string lineaCompras;
            List<string> valoresCompras = new List<string>();

            string queryComprasInsert = "INSERT INTO " + tablaTempCompras + " (Fecha, CodProveedor, NombreProveedor, DireccionProveedor, NumeroProveedor, WebProveedor, CodProducto, NombreProd";
            System.IO.StreamReader archivoCompra = new System.IO.StreamReader(compras[idx]);
            while ((lineaCompras = archivoCompra.ReadLine()) != null)
            {
                if (contadorCompras == 0)
                {
                    contadorCompras = 1;
                    continue;
                }
            }
        }
    }
}
```

5. Poblar Tablas Independientes

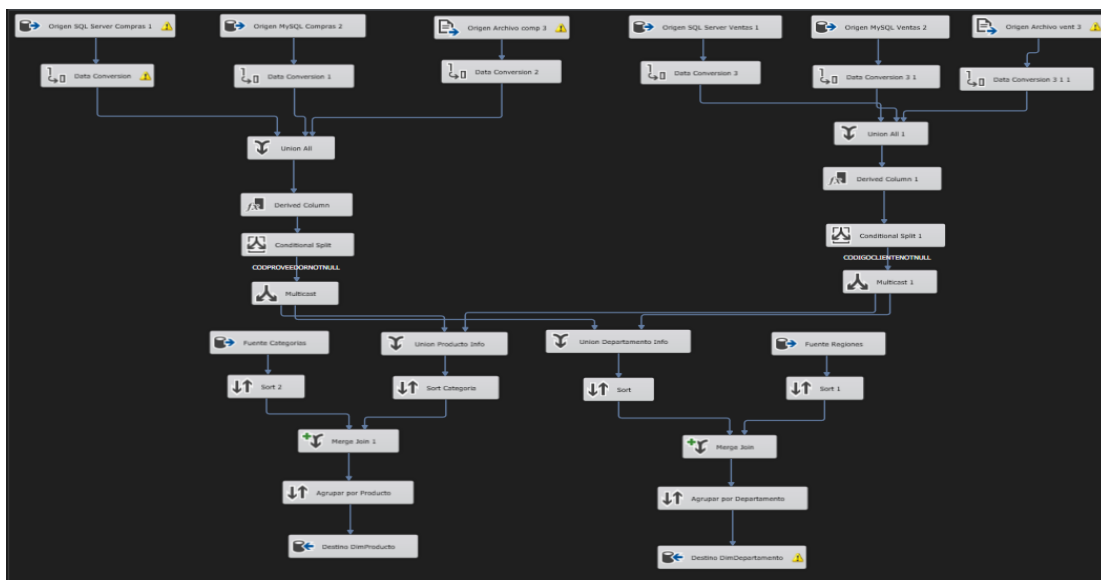
En este paso, se hace la parte de Transform y Load del proceso ETL. Para empezar, se obtienen los datos de las fuentes disponibles: Tablas Pivote SQL Server, Tablas Pivote MySQL y archivos SGFood03. Se unen, transforman y limpian para que la información sea consistente y tenga sentido al momento de analizarla. Por ejemplo, se ignoran los

registros que les haga falta información importante como el código del cliente o código de sucursal. Se transforman a tipos de datos que puedan ser manejados de mejor manera y se empiezan a poblar las tablas independientes. Las tablas independientes en esta caso son las tablas del modelo que no tienen llaves foranes.



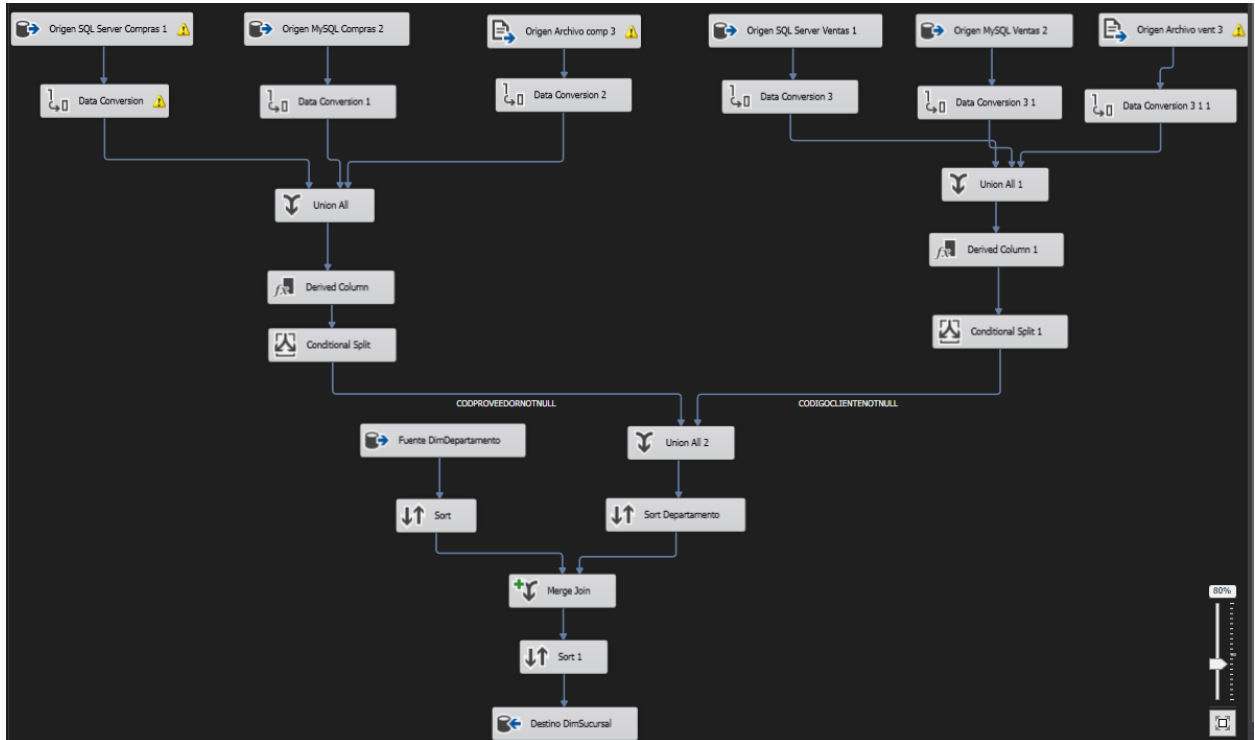
6. Poblar Tablas Dependientes 1

En este paso, se hace el mismo proceso que en paso anterior, sin embargo, las tablas que se utilizan para insertar datos son las que están disponibles tras haber insertado valores en las tablas independientes.



7. Poblar Tablas Dependientes 2

En este paso se hace el mismo proceso que en el paso anterior, con la diferencia que se poblaron las tablas que están disponibles tras haber insertado en las tablas del paso anterior.



8. Poblar Tablas Hechos

Este paso es de los más importantes. Aquí se insertan los datos en las tablas de hechos ya que todas las tablas dimensionales están disponibles para ser referenciadas en las tablas de hechos. Con las herramientas de SSIS, específicamente los Merge Join, se puede hacer referencias a las dimensiones y se insertan los datos en las tablas de hechos.



9. Conteo de Tablas

El último paso consiste en revisar si las tablas del modelo se poblaron exitosamente. Para ello, se ejecuta un script de SQL en donde se selecciona el conteo de registros de cada tabla del modelo y se guardan en variables. Después de ejecutar todo el proceso, las variables almacenaran el conteo de registros en cada tabla del modelo.

Name	Value	Type
» User::CountDimCliente	{249}	Int32
» User::CountDimDepartamento	{22}	Int32
» User::CountDimFecha	{305}	Int32
» User::CountDimProducto	{195}	Int32
» User::CountDimProveedor	{268}	Int32
» User::CountDimRegion	{22}	Int32
» User::CountDimSucursal	{105}	Int32
» User::CountDimVendedor	{51}	Int32
» User::CountFactCompras	{269}	Int32
» User::CountFactVentas	{257}	Int32
» System::CreationDate	{27/08/2024 20:20:44}	DateTime
» System::CreatorComputerName	{DESKTOP-5MM7711}	String