

Actividad de Aprendizaje 3

Proceso de transformación de datos y carga en el data mart final

**Asignatura
Bases de Datos II**

**GRUPO
POSICA2501B010004**

**PROFESOR
Víctor Hugo Mercado Ramos**

**ESTUDIANTE
Carlos Andrés Vega Marín**

**Institución Universitaria Digital de Antioquia
Facultad de Ingeniería y Ciencias Agropecuarias
Especialización en Analítica y Big Data
Medellín
2025**



INTRODUCCIÓN

El modelo estrella es una técnica para modelar y organizar los datos de un modelo entidad relación con el fin de permitir realizar consultas más rápidas y eficientes, al igual que el acceso a los datos cuando se realiza la construcción de tableros de visualización para la empresa.

Para este entregable trabajaremos con el modelo entidad relación de una base de datos llamada Jardinería, la cual transformaremos en un modelo dimensional en el cual se identifiquen las tablas de hechos y dimensiones para optimizar el análisis de datos y la generación de reportes.

También, se realizará la extracción de los datos de las tablas de la base de datos de Jardinería como primer paso de construcción de la ETL.

OBJETIVOS

- Diseñar un modelo estrella que permita realizar el análisis de ventas y transacciones en la base de datos Jardinería.
- Construir la tabla de hechos de tal forma que permita centralizar la información de las ventas.
- Identificar las dimensiones que se relacionarán con la tabla de hechos.
- Relacionar las dimensiones con la tabla de hechos que se defina.
- Crear una base de datos de Staging para alojar los datos extraídos de la base de datos Jardinería.
- Configurar las conexiones necesarias para la extracción de información de las tablas incluidas en el modelo estrella.

PLANTEAMIENTO DEL PROBLEMA

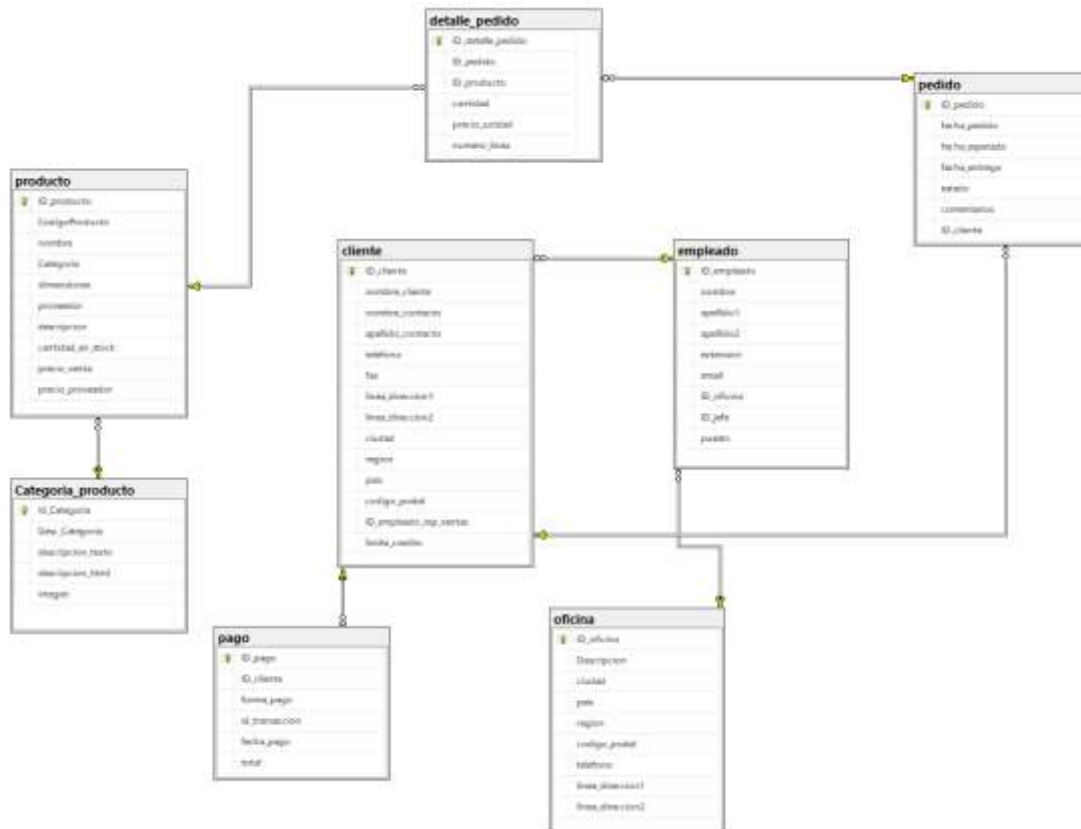
La base de datos Jardinería está diseñada sobre un modelo entidad relación cuyo objetivo es registrar las ventas, los clientes, los productos y los pagos del negocio. Sin embargo, para poder realizar un análisis y consulta eficiente de los datos se requiere de un modelo de datos más simplificado que permita ejecutar consultas que presente las ventas y transacciones con mayor rapidez y menor consumo de recursos.

Para resolver este problema, se propone diseñar un modelo dimensional tipo estrella, donde se definirán una tabla de hechos y varias tablas de dimensiones con sus respectivas medidas. Esta estructura permitirá mejorar la eficiencia de las consultas y facilitar la creación de tableros de control, optimizando así la toma de decisiones estratégicas.

Además, se requiere implementar un proceso ETL que extraiga los datos de la tabla de *Jardinería* para luego realizar las fases de transformación y carga. El primer paso será configurar las fuentes de origen y el destino de los datos, asegurando que se cuente con la información necesaria para completar el resto de los procesos de la ETL.

ANÁLISIS DEL PROBLEMA

La base de datos Jardinería contiene información clave sobre ventas, productos, categorías, oficinas, clientes, empleados y pagos. A partir del modelo entidad relación, podemos concluir lo siguiente:



- Ventas: registradas en las tablas pedido y detalle_pedido.
- Cliente: identificados en la tabla cliente.
- Producto: definidos en las tablas producto y categoria_producto.
- Empleado: relacionados con la tabla clientes y oficina.
- Pago: tiene relación con el cliente.

Limitación del modelo relacional actual: el objetivo principal del modelo entidad relación es organizar los datos de una forma normalizada al guardar la información, pero para realizar consultas y análisis de datos este modelo es lento y por tanto se debe trabajar sobre un modelo dimensional que permita optimizar recursos de procesamiento.

DESCRIPCIÓN MODELO ESTRELLA

Se propone el siguiente modelo estrella para el análisis de eficiente de los datos, a continuación se presenta la table de hechos y las dimensiones.

Tabla de Hechos: venta_transacciones

La tabla de hechos almacena las transacciones de las ventas realizadas.

Variable	Tipo de Dato	Descripción
ID_venta (PK)	int	Identificador único de la venta
ID_pedido (FK)	int	Identificador del pedido realizado
ID_cliente (FK)	int	Relación con el cliente que realizó la compra
ID_producto (FK)	int	Relación con el producto vendido
ID_empleado (FK)	int	Relación con el empleado que realizó la venta
ID_pago (FK)	int	Relación con la forma de pago utilizada
fecha_venta	date	Fecha en que se realizó la venta
cantidad	int	Cantidad de productos vendidos
precio_unitario	decimal	Precio de cada producto vendido
total_venta	decimal	Monto total de la venta

Tablas de Dimensiones: dim_tiempo – Dimensión de tiempo

Esta dimensión permite realizar análisis temporales de las ventas (diario, semanal, mensual, y anual).

Variable	Tipo de Dato	Descripción
ID_tiempo (PK)	int	Identificador único de la fecha
fecha_venta	date	Fecha específica de la transacción
año	int	Año en que se realizó la venta
mes	int	Mes en que se realizó la venta
semana	int	Número de semana del año
día	int	Día del mes

dim_empleado – Dimensión de Empleado

Variable	Tipo de Dato	Descripción
ID_empleado (PK)	int	Identificador único del empleado
nombre	varchar	Nombre del empleado
apellido1	varchar	Primer apellido del empleado
apellido2	varchar	Segundo apellido del empleado
extension	int	Número de extensión telefónica del empleado
email	varchar	Correo electrónico del empleado
ID_Jefe	int	Identificador del jefe del empleado
ID_Oficina	int	Identificador de la oficina donde trabaja el empleado
puesto	varchar	Puesto del empleado

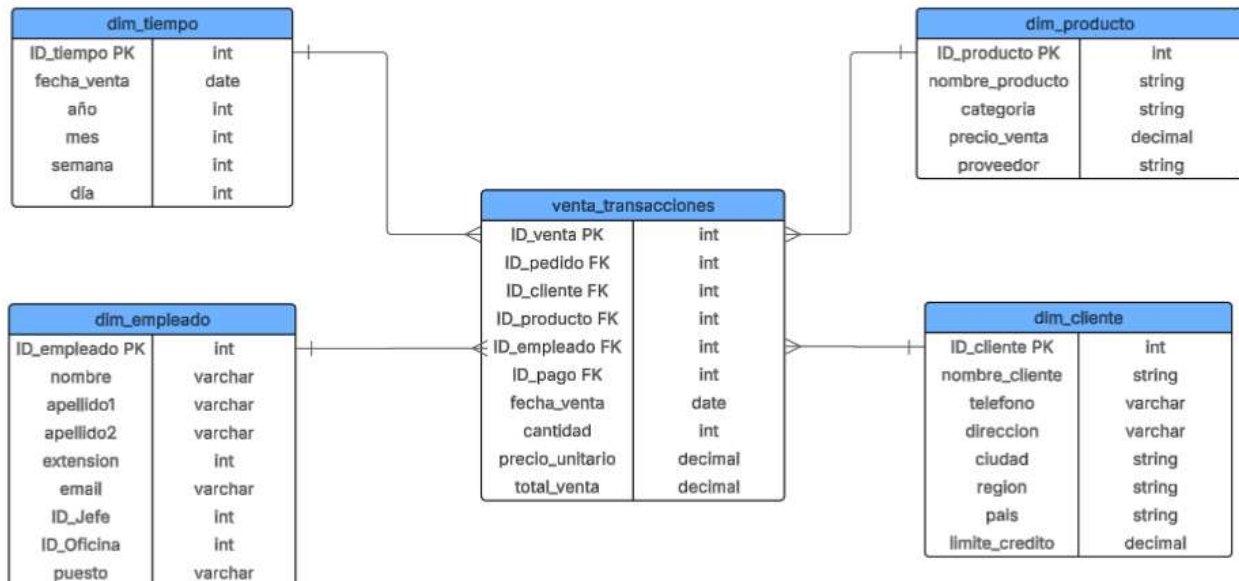
dim_cliente – Dimensión del Cliente

Variable	Tipo de Dato	Descripción
ID_cliente (PK)	int	Identificador único del cliente
nombre_cliente	varchar	Nombre del cliente
telefono	varchar	Teléfono del cliente
direccion	varchar	Dirección del cliente
ciudad	varchar	Ciudad de residencia del cliente
región	varchar	Región del cliente
país	varchar	País del cliente
limite_credito	int	Límite de crédito asignado al cliente

dim_producto – Dimensión del Producto

Variable	Tipo de Dato	Descripción
ID_producto (PK)	int	Identificador único del producto
nombre_producto	varchar	Nombre del producto
categoría	varchar	Categoría a la que pertenece el producto
precio_venta	int	Precio de venta del producto
proveedor	varchar	Proveedor del producto

MODELO ESTRELLA IMPLEMENTADO



CONSULTAS SQL

Dimensión empleado

```
SELECT [ID_empleado]
      ,[nombre]
      ,[apellido1]
      ,[apellido2]
      ,[extension]
      ,[email]
      ,[ID_oficina]
      ,[ID_jefe]
      ,[puesto]
FROM [Jardineria].[dbo].[empleado]
```

```
CREATE TABLE "ST_Empleado" (
  "ID_empleado" int,
  "nombre" nvarchar(50),
  "apellido1" nvarchar(50),
  "apellido2" nvarchar(50),
  "extension" nvarchar(10),
  "email" nvarchar(100),
  "ID_oficina" int,
  "ID_jefe" int,
  "puesto" nvarchar(50))
```

Dimensión producto

```
SELECT [ID_producto]
      ,[nombre]
      ,[Categoria]
      ,[proveedor]
      ,[precio_venta]
FROM [Jardineria].[dbo].[producto]
```

```
CREATE TABLE "Destino_ST_Producto" (
  "ID_producto" int,
  "nombre" nvarchar(70),
  "Categoria" int,
  "proveedor" nvarchar(50),
  "precio_venta" numeric(15,2))
```


Dimensión Cliente

```
SELECT [ID_cliente]
      ,[nombre_cliente]
      ,[telefono]
      ,[linea_direccion1]
      ,[ciudad]
      ,[region]
      ,[pais]
      ,[limite_credito]
FROM [Jardineria].[dbo].[cliente]
```

```
CREATE TABLE "ST_Cliente" (
  "ID_cliente" int,
  "nombre_cliente" nvarchar(50),
  "telefono" nvarchar(15),
  "linea_direccion1" nvarchar(50),
  "ciudad" nvarchar(50),
  "region" nvarchar(50),
  "pais" nvarchar(50),
  "limite_credito" numeric(15,2))
```

Dimensión tiempo

```
SELECT DISTINCT [fecha_pedido]
FROM [Jardineria].[dbo].[pedido]
WHERE [fecha_pedido] IS NOT NULL
```

UNION ALL

```
SELECT DISTINCT [fecha_esperada]
FROM [Jardineria].[dbo].[pedido]
WHERE [fecha_esperada] IS NOT NULL
```

UNION ALL

```
SELECT DISTINCT [fecha_entrega]
FROM [Jardineria].[dbo].[pedido]
WHERE [fecha_entrega] IS NOT NULL
```

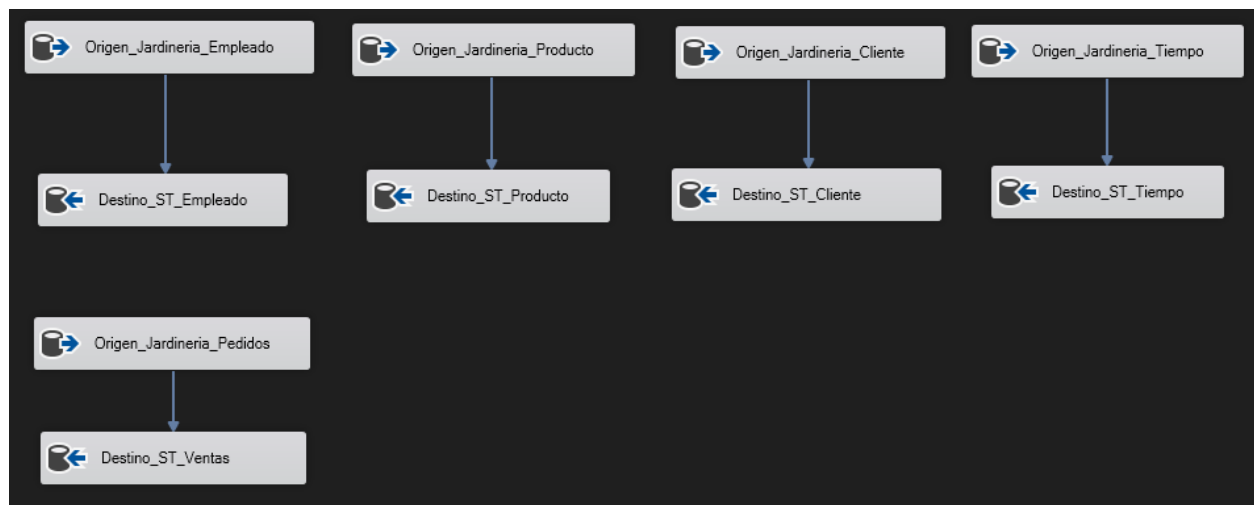
```
ORDER BY [fecha_pedido]
```

```
CREATE TABLE "ST_Tiempo" (  
    "fechas_tbl_pedidos" date)
```

Dimensión Pedido

```
SELECT [ID_pedido]  
    ,[fecha_pedido]  
    ,[fecha_esperada]  
    ,[fecha_entrega]  
    ,[estado]  
    ,[comentarios]  
    ,[ID_cliente]  
FROM [Jardineria].[dbo].[pedido]
```

```
CREATE TABLE "ST_Ventas" (  
    "ID_pedido" int,  
    "fecha_pedido" date,  
    "fecha_esperada" date,  
    "fecha_entrega" date,  
    "estado" nvarchar(15),  
    "comentarios" nvarchar(max),  
    "ID_cliente" int)
```





Descripción del análisis.

Al ejecutar la solución, todos los orígenes de las tablas configurados en la base de datos de Jardinería se conectan con el proceso de destino en la base de datos de Staging. De esta manera, se lleva a cabo la transferencia de información correspondiente a la primera fase de la ETL, denominada extracción de datos.

Sentencias proceso de ETL

Dimensión Tiempo

En el proceso de extracción de la dimensión de tiempo se garantiza un registro único de cada fecha utilizando la sentencia DISTINCT.

```

SELECT DISTINCT fecha_pedido
FROM      ST_Tiempo
ORDER BY 1
  
```

Derived Column Transformation Editor

Specify the expressions used to create new column values, and indicate whether the values update existing columns or populate new columns.

Variables y parámetros

Columns

fecha_pedido

Funciones de fecha y hora

DATEADD(«datepart», «number», «date»)

DATEDIFF(«datepart», «startdate», «enddate»)

DATEPART(«datepart», «date»)

DAY(«date»)

GETDATE()

GETUTCDATE()

MONTH(«date»)

YEAR(«date»)

Descripción:

Derived Column Name	Derived Column	Expression	Data Type	Le
Dia	<add as new column>	DAY(fecha_pedido)	entero de cuatro bytes ...	
Mes	<add as new column>	MONTH(fecha_pedido)	entero de cuatro bytes ...	
Anio	<add as new column>	YEAR(fecha_pedido)	entero de cuatro bytes ...	
NumeroSemana	<add as new column>	DATEPART("WW", [fecha_pedido])	entero de cuatro bytes ...	
Trimestre	<add as new column>	DATEPART("quarter", [fecha_pedido])	entero de cuatro bytes ...	
DiaAnio	<add as new column>	DATEPART("dy", [fecha_pedido])	entero de cuatro bytes ...	
DiaSemana	<add as new column>	DATEPART("dw", [fecha_pedido])	entero de cuatro bytes ...	

Configure Error Output... OK Cancel Ayuda

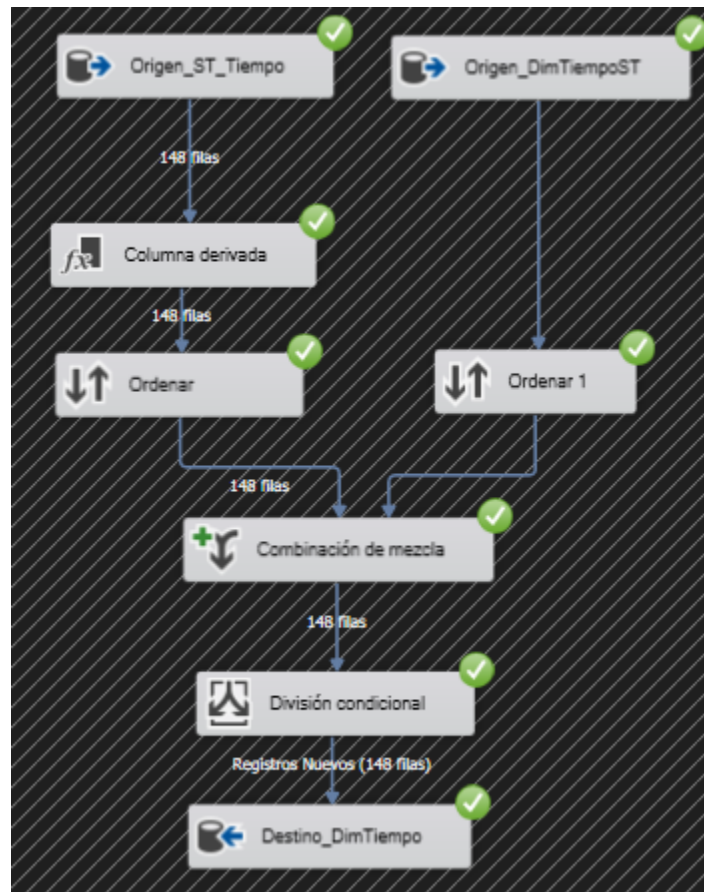
Descripción de la derivación de columnas:

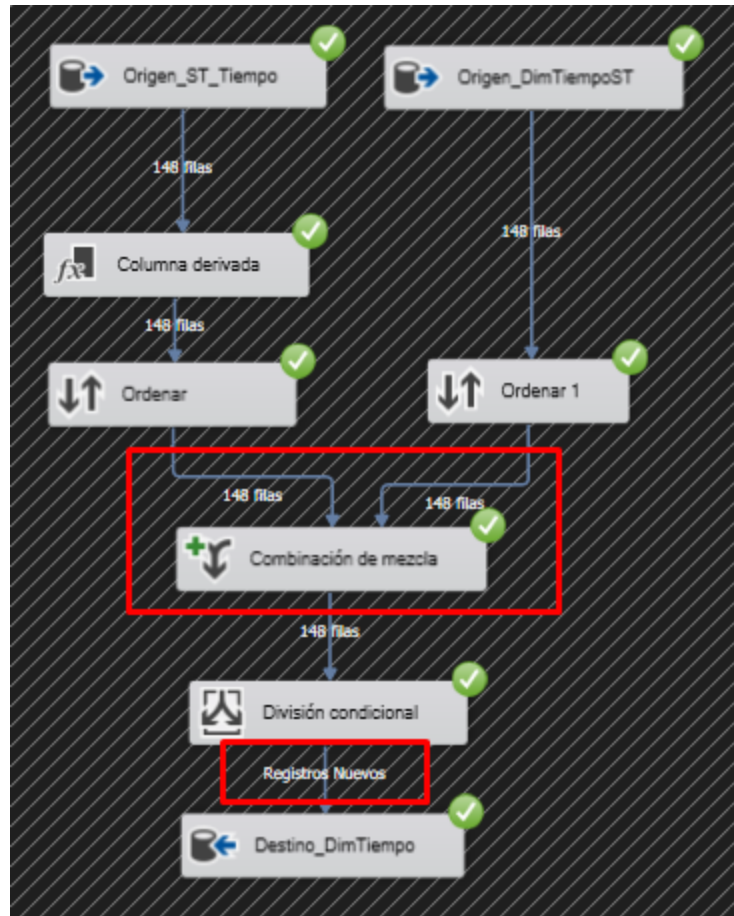
Nombre de la Columna	Expresión Usada	Descripción
Dia	DAY(fecha_pedido)	Extrae el número del día del mes (1-31).
Mes	MONTH(fecha_pedido)	Extrae el número del mes (1-12).
Anio	YEAR(fecha_pedido)	Extrae el año completo (ej. 2024).
NumeroSemana	DATEPART("ww", [fecha_pedido])	Número de la semana del año (1-53 según configuración regional).
Trimestre	DATEPART("quarter", [fecha_pedido])	Trimestre del año (1 a 4).
DiaAnio	DATEPART("dy", [fecha_pedido])	Día del año (1 a 365 o 366 si es bisiesto).
DiaSemana	DATEPART("dw", [fecha_pedido])	Día de la semana (1 a 7, dependiendo del primer día configurado en el sistema: domingo o lunes).

Este proceso busca actualizar o insertar datos en la Dimensión de Tiempo (Destino_DimTiempo), comparando registros provenientes de la tabla de origen (Origen_ST_Tiempo) con los ya existentes en la dimensión (Origen_DimTiempoST). El objetivo es evitar duplicados y mantener la dimensión actualizada con registros nuevos.

Este flujo Dim_Tiempo realiza las siguientes acciones:

- Extrae datos nuevos de tiempo.
- Los transforma y ordena.
- Los compara con los datos ya existentes en la tabla de dimensión de tiempo.
- Filtra los que son realmente nuevos.
- Inserta solo los registros nuevos en la dimensión DimTiempo, evitando duplicados y manteniéndola actualizada.





No se insertaron registros nuevos en la tabla DimTiempo porque ninguno de los 148 registros fue identificado como "nuevo" en el paso División Condicional.

Esto significa que todos los registros que llegaron desde el origen ya existen en la tabla destino.

Por este motivo, el paso del flujo que se encarga de insertar solo los registros nuevos no recibió ningún dato, y como resultado, no se inserta nada.

Dimensión Producto

```

SELECT    ST_Producto.*
FROM      ST_Producto
  
```

Derived Column Transformation Editor

Specify the expressions used to create new column values, and indicate whether the values update existing columns or populate new columns.

Variables y parámetros

Columns

- nombre
- proveedor
- ID_producto
- Categoría
- precio_venta

Funciones matemáticas

Funciones de cadena

Funciones de fecha y hora

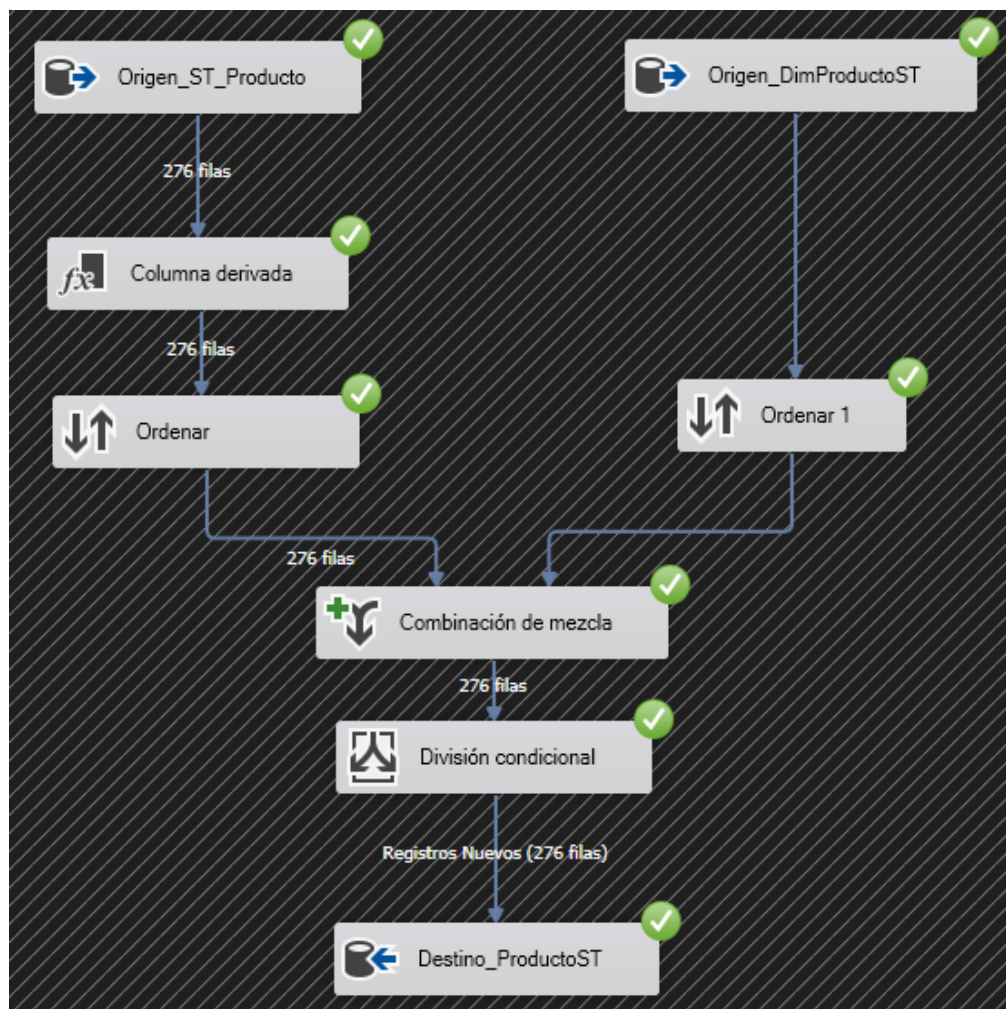
Funciones NULL

Conversiones de tipo


Operadores

Descripción:

Derived Column Name	Derived Column	Expression	Data Type	Le
NombreProducto	<add as new column>	REPLACENULL(nombre,"Sin dato")	cadena Unicode [DT_W...	1
NombreProveedor	<add as new column>	REPLACENULL(proveedor,"Sin dato")	cadena Unicode [DT_W...	1



Dimensión Cliente


Derived Column Transformation Editor

Specify the expressions used to create new column values, and indicate whether the values update existing columns or populate new columns.

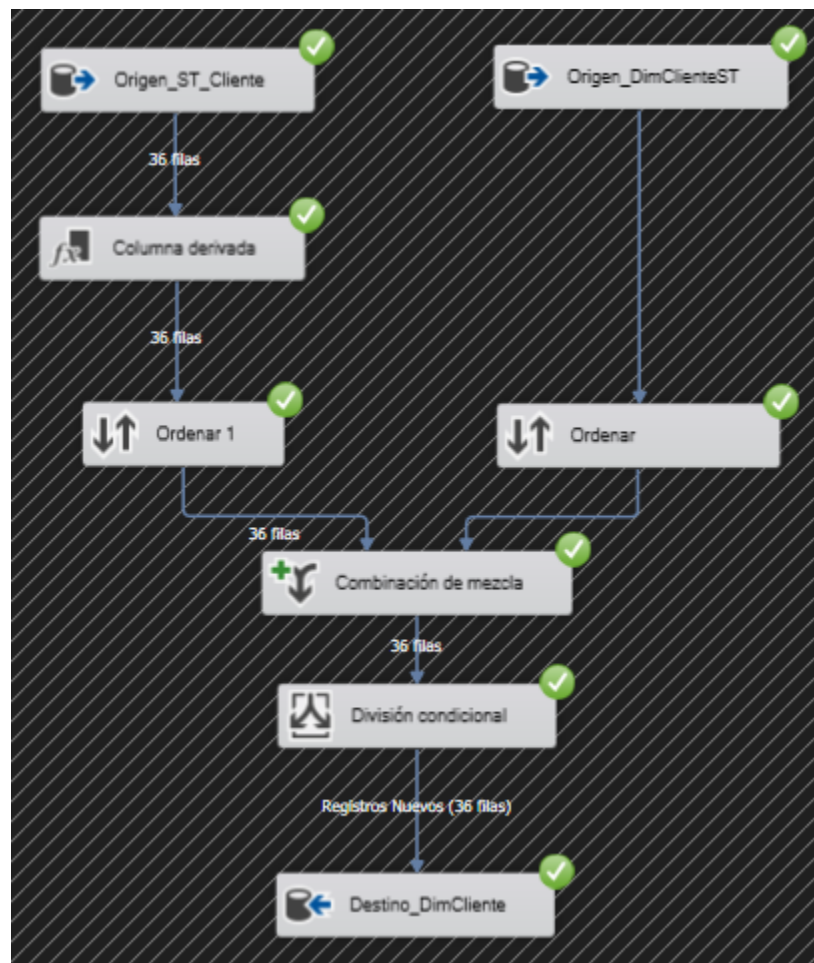
+ Variables y parámetros
+ Columns
+ nombre_cliente
+ linea_direccion1
+ ciudad
+ region
+ pais
+ ID_cliente
+ telefono
+ limite_credito

+ Funciones matemáticas
+ Funciones de cadena
+ Funciones de fecha y hora
+ Funciones NULL
+ Conversiones de tipo
+ Operadores

Descripción:

Derived Column Name	Derived Column	Expression	Data Type	Le
nombre_cliente	Reemplazar 'nombre_c...	REPLACENULL(nombre_cliente,"Sin Dato")	cadena Unicode [DT_W...	5
linea_direccion1	Reemplazar 'linea_dire...	REPLACENULL(linea_direccion1,"Sin Dato")	cadena Unicode [DT_W...	5
ciudad	Reemplazar 'ciudad'	REPLACENULL(ciudad,"Sin Dato")	cadena Unicode [DT_W...	5
region	Reemplazar 'region'	REPLACENULL(region,"Sin Dato")	cadena Unicode [DT_W...	5
pais	Reemplazar 'pais'	REPLACENULL(pais,"Sin Dato")	cadena Unicode [DT_W...	5

Configure Error Output...
OK
Cancel
Ayuda



Dim Empleado

Derived Column Transformation Editor

Specify the expressions used to create new column values, and indicate whether the values update existing columns or populate new columns.

+

 Variables y parámetros

+

 Columns

+

 Funciones matemáticas

+

 Funciones de cadena

+

 Funciones de fecha y hora

+

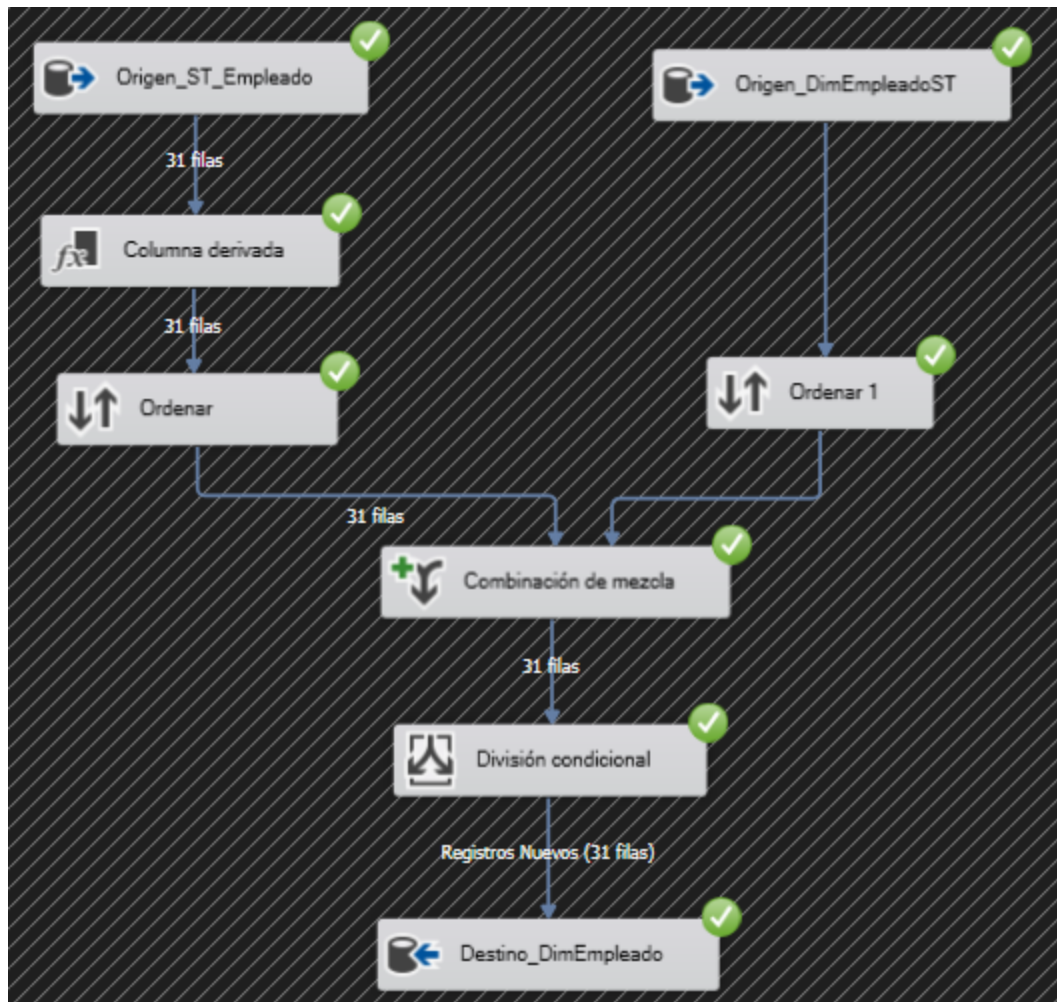
 Funciones NULL

+

 Conversiones de tipo

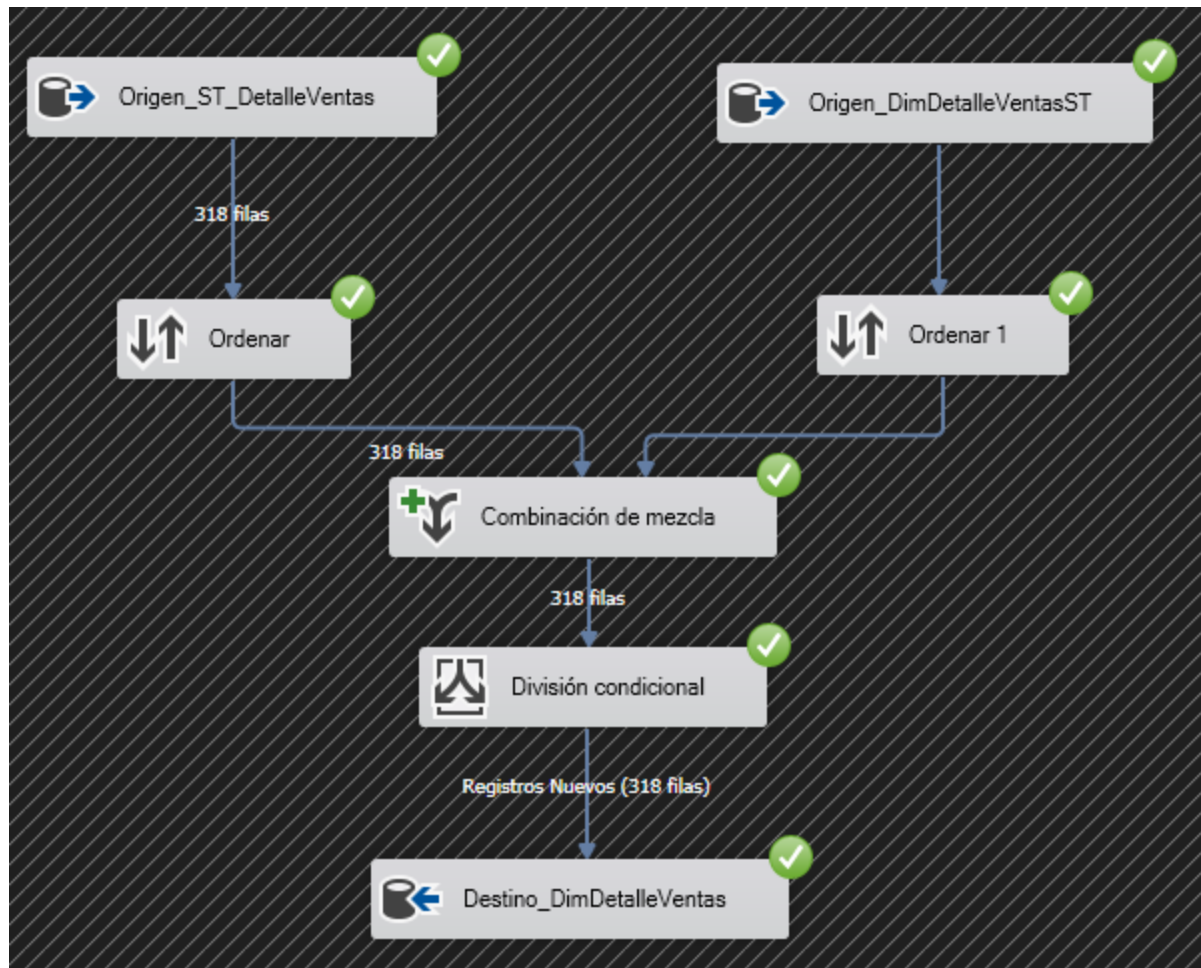
+

 Operadores



Dimensión Detalle de Ventas

Para esta tabla no se identifica ninguna transformación a realizar.



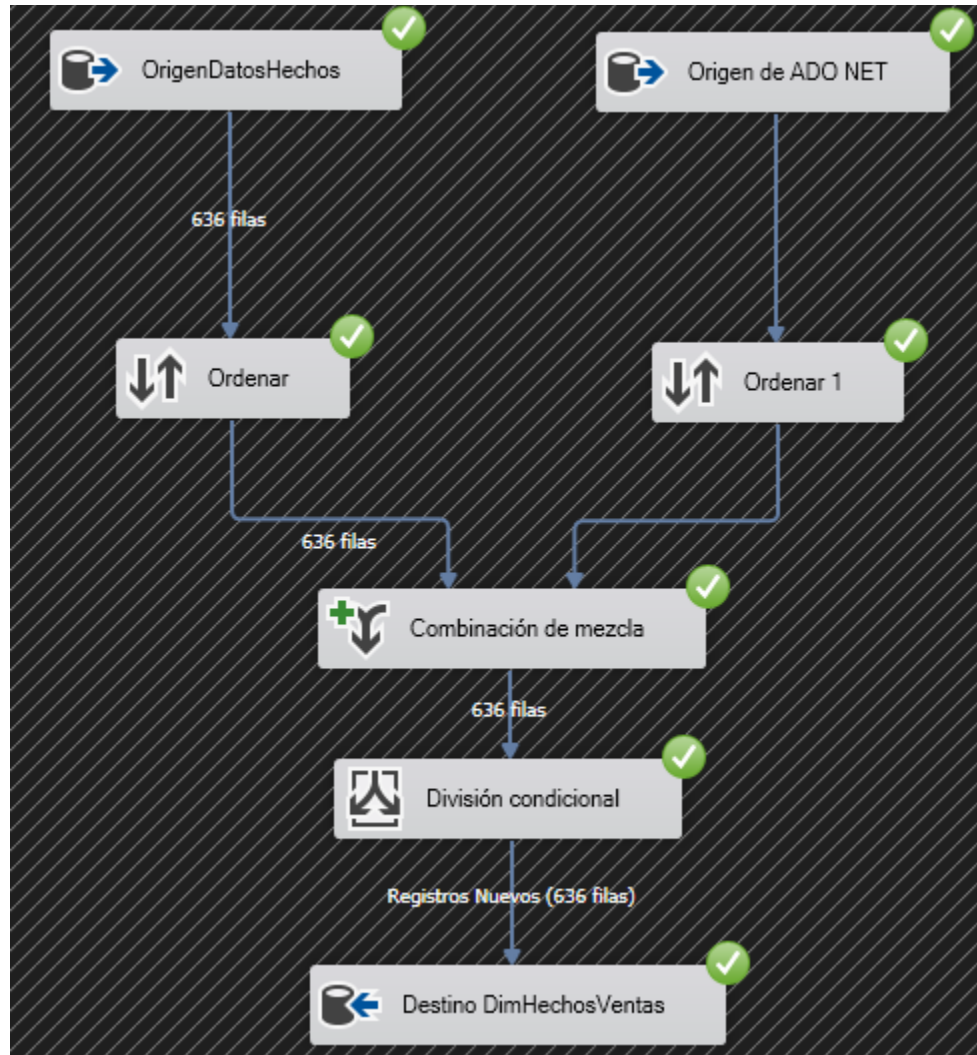
Construcción tabla de Hechos:

```

SELECT
    EV.ID_pedido AS ID_pedido_encabezado,
    DV.ID_pedido AS ID_pedido_detalle,
    OD.IDDimTiempo AS ID_fecha_pedido,
    EV.fecha_pedido AS fecha_pedido_original,
    EV.estado,
    EV.comentarios,
    EV.ID_cliente,
    DV.ID_detalle_pedido,
    DV.ID_producto,
    DV.cantidad,
    DV.precio_unidad
FROM
    [dbo].[ST_Ventas] EV
    INNER JOIN [dbo].[ST_Detalle_Ventas] DV ON EV.ID_pedido = DV.ID_pedido
    INNER JOIN [dbo].[DimTiempoST] OD ON EV.fecha_pedido = OD.fecha_pedido
  
```

```

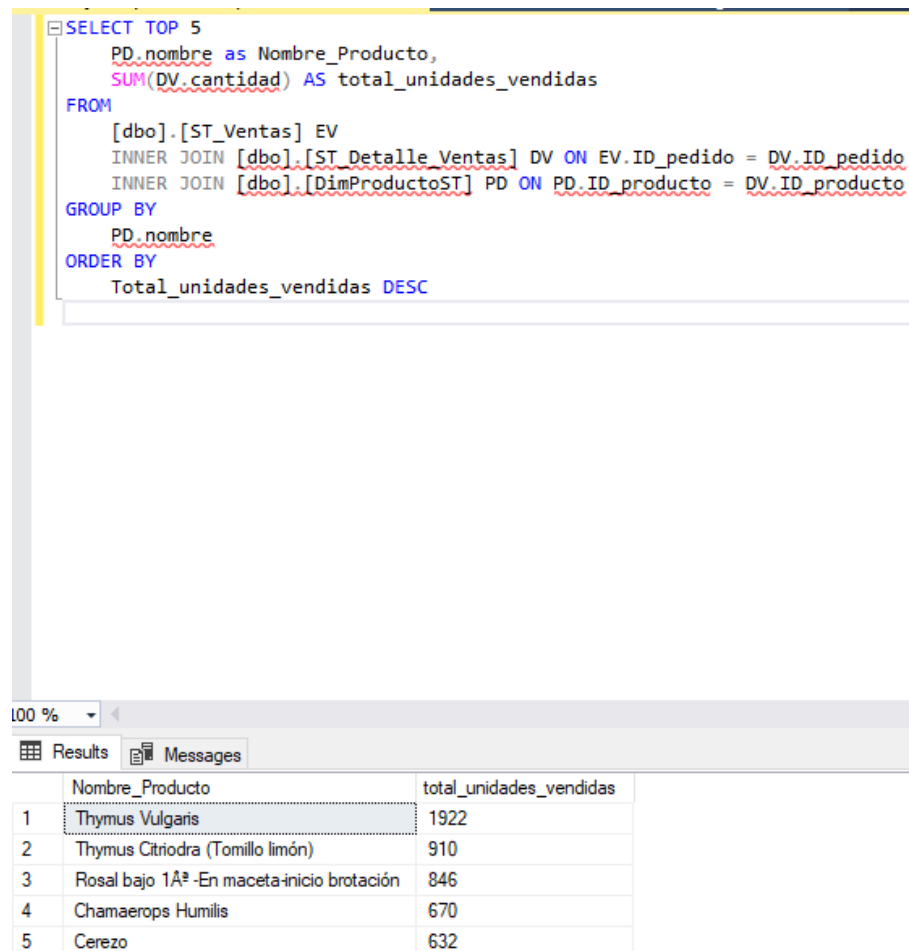
INNER JOIN [dbo].[DimClienteST] DC ON DC.ID_cliente = EV.ID_cliente
INNER JOIN [dbo].[DimProductoST] PD ON PD.ID_producto = DV.ID_producto
ORDER BY
    EV.ID_pedido
    
```



Consulta SQL

Esta consulta identifica los 5 productos con mayor cantidad de unidades vendidas

```
SELECT TOP 5
    PD.nombre as Nombre_Producto,
    SUM(DV.cantidad) AS total_unidades_vendidas
FROM
    [dbo].[ST_Ventas] EV
    INNER JOIN [dbo].[ST_Detalle_Ventas] DV ON EV.ID_pedido = DV.ID_pedido
    INNER JOIN [dbo].[DimProductoST] PD ON PD.ID_producto = DV.ID_producto
GROUP BY
    PD.nombre
ORDER BY
    Total_unidades_vendidas DESC
```



The screenshot shows a SQL query window with the following text:

```
SELECT TOP 5
    PD.nombre as Nombre_Producto,
    SUM(DV.cantidad) AS total_unidades_vendidas
FROM
    [dbo].[ST_Ventas] EV
    INNER JOIN [dbo].[ST_Detalle_Ventas] DV ON EV.ID_pedido = DV.ID_pedido
    INNER JOIN [dbo].[DimProductoST] PD ON PD.ID_producto = DV.ID_producto
GROUP BY
    PD.nombre
ORDER BY
    Total_unidades_vendidas DESC
```

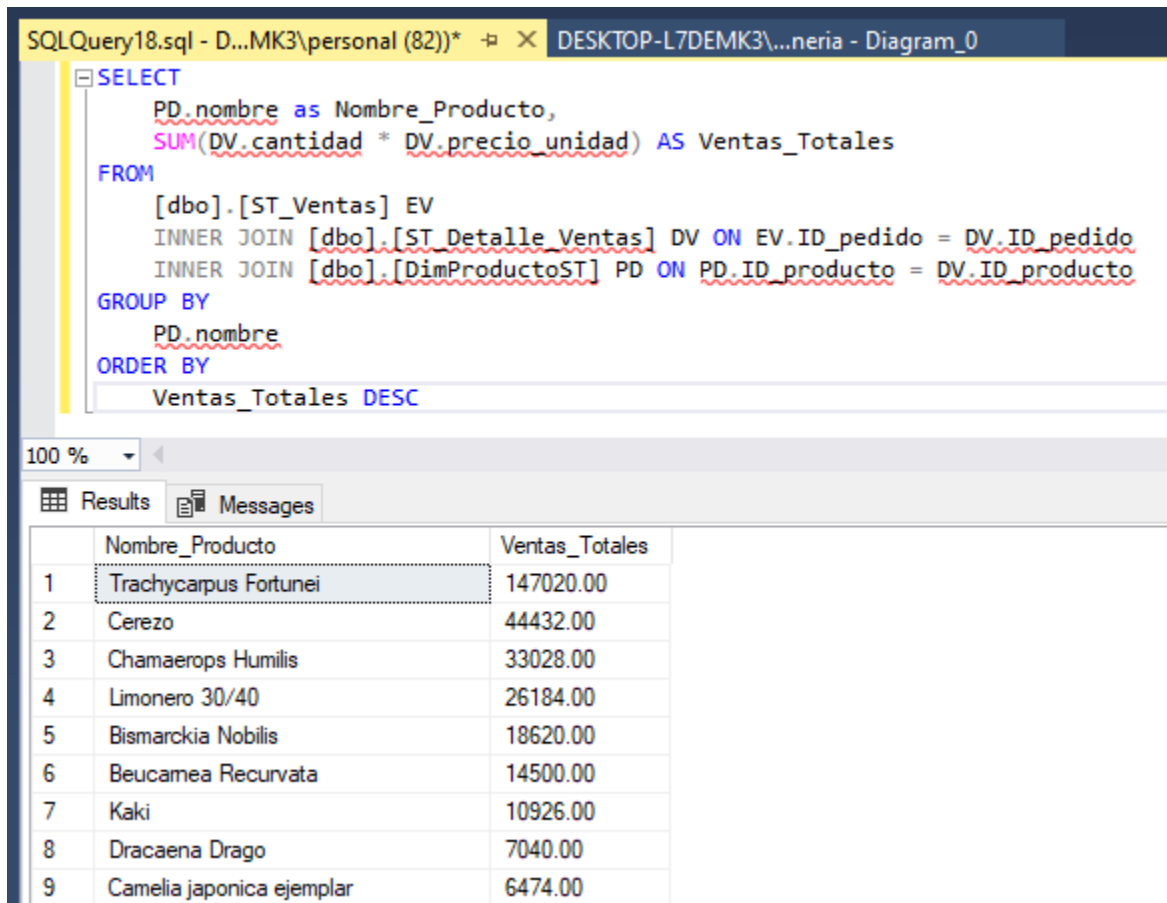
Below the query window, the 'Results' tab is active, displaying the following data:

	Nombre_Producto	total_unidades_vendidas
1	Thymus Vulgaris	1922
2	Thymus Citriodra (Tomillo limón)	910
3	Rosal bajo 1Aº -En maceta-inicio brotación	846
4	Chamaerops Humilis	670
5	Cerezo	632

El producto más vendido fue Thymus Vulgaris con 1.922 unidades.

Consulta para identificar el producto con mayor ingreso en ventas

```
SELECT
    PD.nombre as Nombre_Producto,
    SUM(DV.cantidad * DV.precio_unidad) AS Ventas_Totales
FROM
    [dbo].[ST_Ventas] EV
    INNER JOIN [dbo].[ST_Detalle_Ventas] DV ON EV.ID_pedido = DV.ID_pedido
    INNER JOIN [dbo].[DimProductoST] PD ON PD.ID_producto = DV.ID_producto
GROUP BY
    PD.nombre
ORDER BY
    Ventas_Totales DESC
```



The screenshot shows a SQL query window with the following text:

```
SELECT
    PD.nombre as Nombre_Producto,
    SUM(DV.cantidad * DV.precio_unidad) AS Ventas_Totales
FROM
    [dbo].[ST_Ventas] EV
    INNER JOIN [dbo].[ST_Detalle_Ventas] DV ON EV.ID_pedido = DV.ID_pedido
    INNER JOIN [dbo].[DimProductoST] PD ON PD.ID_producto = DV.ID_producto
GROUP BY
    PD.nombre
ORDER BY
    Ventas_Totales DESC
```

Below the query window, the 'Results' tab is active, displaying a table with 9 rows and 2 columns: 'Nombre_Producto' and 'Ventas_Totales'. The results are ordered by 'Ventas_Totales' in descending order.

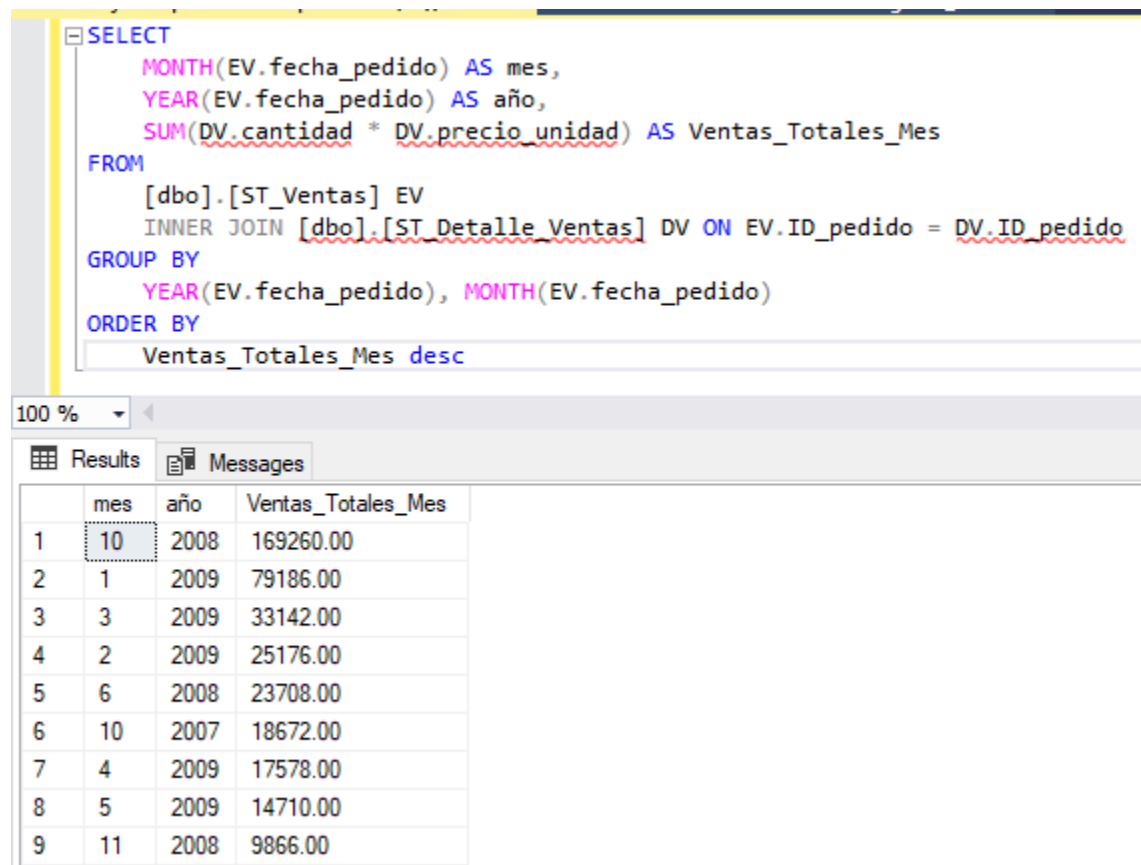
	Nombre_Producto	Ventas_Totales
1	Trachycarpus Fortunei	147020.00
2	Cerezo	44432.00
3	Chamaerops Humilis	33028.00
4	Limonero 30/40	26184.00
5	Bismarckia Nobilis	18620.00
6	Beucamea Recurvata	14500.00
7	Kaki	10926.00
8	Dracaena Drago	7040.00
9	Camelia japonica ejemplar	6474.00

El producto con mayor ingreso en ventas fue Trachycarpus Fortunei

Consulta para identificar el mes y el año que más ventas se realizaron

```

SELECT
    MONTH(EV.fecha_pedido) AS mes,
    YEAR(EV.fecha_pedido) AS año,
    SUM(DV.cantidad * DV.precio_unidad) AS Ventas_Totales_Mes
FROM
    [dbo].[ST_Ventas] EV
    INNER JOIN [dbo].[ST_Detalle_Ventas] DV ON EV.ID_pedido = DV.ID_pedido
GROUP BY
    YEAR(EV.fecha_pedido), MONTH(EV.fecha_pedido)
ORDER BY
    Ventas_Totales_Mes desc
    
```



The screenshot shows a SQL query window with the following text:

```

SELECT
    MONTH(EV.fecha_pedido) AS mes,
    YEAR(EV.fecha_pedido) AS año,
    SUM(DV.cantidad * DV.precio_unidad) AS Ventas_Totales_Mes
FROM
    [dbo].[ST_Ventas] EV
    INNER JOIN [dbo].[ST_Detalle_Ventas] DV ON EV.ID_pedido = DV.ID_pedido
GROUP BY
    YEAR(EV.fecha_pedido), MONTH(EV.fecha_pedido)
ORDER BY
    Ventas_Totales_Mes desc
    
```

Below the query window, the 'Results' tab is active, displaying a table with 9 rows and 4 columns: 'mes', 'año', and 'Ventas_Totales_Mes'. The first row is highlighted, showing '10' for the month and '2008' for the year, with a total sales value of 169260.00.

	mes	año	Ventas_Totales_Mes
1	10	2008	169260.00
2	1	2009	79186.00
3	3	2009	33142.00
4	2	2009	25176.00
5	6	2008	23708.00
6	10	2007	18672.00
7	4	2009	17578.00
8	5	2009	14710.00
9	11	2008	9866.00

En octubre de 2008 fue el año con más ventas realizadas, con un total de 169.260.