

**Título:** Informe de Proceso ETL y Modelo Estrella para Jardinería

**Autores:**

Acebedo Londoño Camilo

Hernandez Lopera Jeronimo

Pérez Severiche Jorge Luis

**Institución Universitaria Digital de Antioquia**

**Materia:** Bases de Datos II

**Docente:** Antonio Jesús Valderrama

**Fecha:** 30 de septiembre de 2025

## **Introducción**

El presente informe documenta el proceso de construcción de un Data Mart con modelo estrella para la base de datos de Jardinería. El procedimiento incluyó la restauración de la base de datos, la creación de la base de datos *staging* (stg), el diseño del modelo estrella (dimensiones y hechos), y el desarrollo de scripts ETL (Extract, Transform, Load) en Transact-SQL (T-SQL). El objetivo fue garantizar la integridad, calidad y consistencia de los datos, aplicando buenas prácticas en ingeniería de datos y habilitando un entorno analítico para identificar métricas clave, como el producto más vendido y las tendencias de ventas.

## **Objetivos**

### ***Objetivo general***

Diseñar, implementar y validar un datamart con modelo estrella para el área de ventas de la empresa de Jardinería, con el fin de facilitar el análisis de información y la toma de decisiones estratégicas a partir de datos consolidados y organizados.

### ***Objetivos específicos***

- Analizar los datos de la empresa de Jardinería para identificar las entidades principales y las relaciones necesarias para el modelado dimensional.
- Diseñar el modelo estrella definiendo la tabla de hechos y las dimensiones que permitan el análisis de ventas por cliente, producto, categoría y tiempo.
- Aplicar técnicas de transformación y normalización de datos para garantizar consistencia y calidad en la información cargada al datamart.

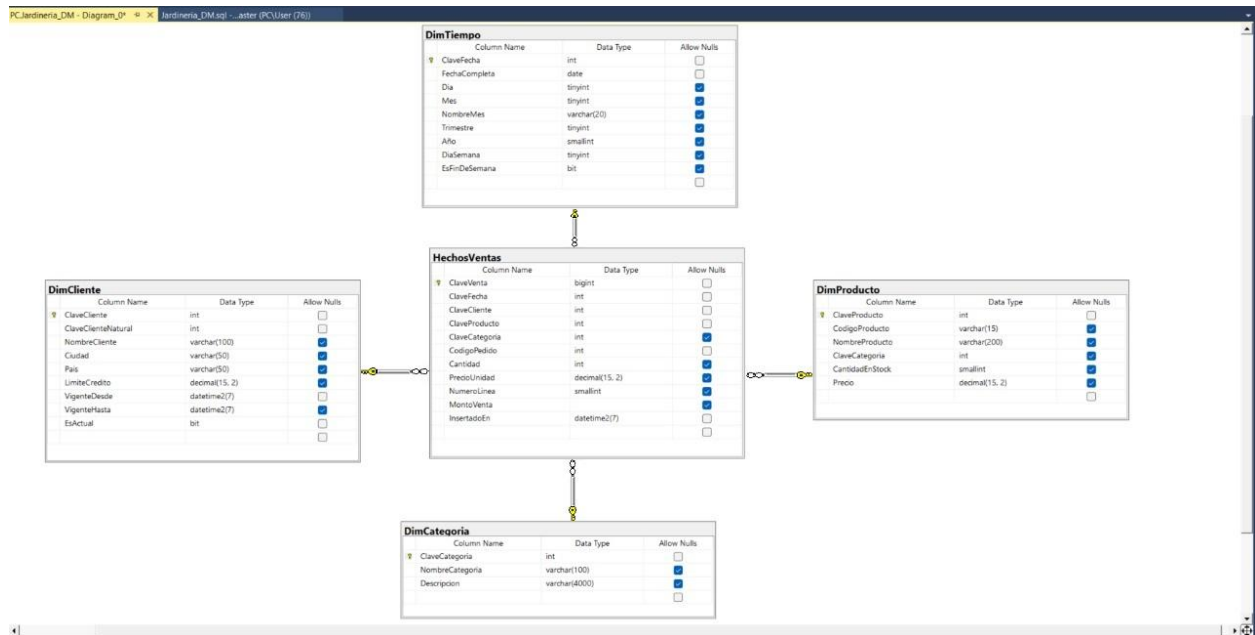
- Implementar el datamart en SQL Server mediante la restauración, carga y estructuración de tablas de hechos y dimensiones.
- Validar la correcta carga e integridad de los datos mediante consultas SQL que verifiquen la relación entre hechos y dimensiones.
- Documentar el proceso de diseño, implementación y validación, incluyendo diagramas, consultas y justificaciones técnicas, para su presentación académica.

## Metodología

El proceso se estructuró en las siguientes fases:

1. Restauración de la base de datos de producción Jardinería.
2. Creación de la base de datos de *staging* (jardineria\_stg).
3. Verificación de consistencia y disponibilidad de los datos en *staging*.
4. Definición del modelo estrella en la base **Jardineria\_DM**.
5. Implementación de procesos ETL para trasladar, limpiar y transformar los datos.
6. Inserción de registros en las dimensiones y en la tabla de hechos de ventas.
7. Validación de la carga, integridad referencial y consistencia de los datos

## Modelo Estrella



## Scripts de Creación

-- DimTiempo

```
CREATE TABLE dbo.DimTiempo (
    ClaveFecha INT PRIMARY KEY, -- AAAAMMDD
    FechaCompleta DATE NOT NULL,
    Dia TINYINT,
    Mes TINYINT,
    NombreMes VARCHAR(20),
    Trimestre TINYINT,
    Año SMALLINT,
    DiaSemana TINYINT,
    EsFinDeSemana BIT
);
```

-- DimCliente (SCD Tipo 2)

```
CREATE TABLE dbo.DimCliente (
```

```
ClaveCliente    INT IDENTITY(1,1) PRIMARY KEY,  
ClaveClienteNatural INT NOT NULL,  
NombreCliente   VARCHAR(100),  
Ciudad          VARCHAR(50),  
Pais            VARCHAR(50),  
LimiteCredito   DECIMAL(15,2),  
VigenteDesde    DATETIME2 NOT NULL DEFAULT SYSUTCDATETIME(),  
VigenteHasta    DATETIME2 NULL,  
EsActual        BIT NOT NULL DEFAULT 1  
);
```

-- DimCategoria

```
CREATE TABLE dbo.DimCategoria (  
    ClaveCategoria INT IDENTITY(1,1) PRIMARY KEY,  
    NombreCategoria VARCHAR(50) UNIQUE,  
    Descripcion    VARCHAR(4000)  
);
```

-- DimProducto

```
CREATE TABLE dbo.DimProducto (  
    ClaveProducto  INT IDENTITY(1,1) PRIMARY KEY,  
    CodigoProducto VARCHAR(15) UNIQUE,  
    NombreProducto VARCHAR(200),  
    ClaveCategoria INT NULL,  
    CantidadEnStock SMALLINT,  
    Precio         DECIMAL(15,2)  
);
```

-- HechosVentas

```
CREATE TABLE dbo.HechosVentas (  
    ClaveVenta  BIGINT IDENTITY(1,1) PRIMARY KEY,  
    ClaveFecha  INT NOT NULL,
```

ClaveCliente INT NOT NULL,  
ClaveProducto INT NOT NULL,  
ClaveCategoria INT NULL,  
CodigoPedido INT NOT NULL,  
Cantidad INT,  
PrecioUnidad DECIMAL(15,2),  
NumeroLinea SMALLINT NULL,  
MontoVenta AS (Cantidad \* PrecioUnidad) PERSISTED,  
InsertadoEn DATETIME2 NOT NULL DEFAULT SYSUTCDATETIME());

- **Tabla de hechos:** HechosVentas.
- **Tablas de dimensiones:** DimTiempo, DimCliente, DimCategoria, DimProducto.

Las relaciones se establecen mediante claves sustitutas, asegurando integridad y eficiencia en consultas analíticas.

## Scripts ETL

### *Extracción*

Se extrajeron datos de las tablas de la base de datos de producción hacia la base *staging*. Este paso garantizó independencia del origen y sirvió como punto intermedio para aplicar transformaciones.

## ***Transformación***

- Se aplicaron transformaciones clave:
- Limpieza de cadenas con LTRIM/RTRIM.
- Normalización de claves (ClaveFecha en formato yyyyymmdd).
- Enriquecimiento con campos calculados (MontoVenta).
- Manejo de Slowly Changing Dimension (SCD Tipo 2) para clientes, preservando histórico de cambios en ciudad, país o límite de crédito.

## **Carga**

Se cargaron los datos transformados desde *staging* hacia las dimensiones y la tabla de hechos en el Data Mart final. Se implementaron verificaciones para evitar duplicados y asegurar idempotencia.

## ***Carga de DimTiempo***

```
DECLARE @inicio DATE = '2018-01-01', @fin DATE = '2030-12-31';
```

```
;WITH Fechas AS (
```

```
    SELECT @inicio AS f
```

```
    UNION ALL
```

```
    SELECT DATEADD(DAY,1,f) FROM Fechas WHERE f < @fin
```

```
)
```

```
INSERT INTO dbo.DimTiempo (ClaveFecha, FechaCompleta, Dia, Mes, NombreMes,  
Trimestre, Anio, DiaSemana, EsFinDeSemana)
```

```
SELECT
```

```
    CONVERT(INT, FORMAT(f,'yyyyMMdd')),
```

```
    f,
```

```

DATEPART(DAY,f),
DATEPART(MONTH,f),
DATENAME(MONTH,f),
DATEPART(QUARTER,f),
DATEPART(YEAR,f),
DATEPART(WEEKDAY,f),
CASE WHEN DATEPART(WEEKDAY,f) IN (1,7) THEN 1 ELSE 0 END
FROM Fechas
OPTION (MAXRECURSION 0);Carga de DimCategoria

```

```

INSERT INTO dbo.DimCategoria (NombreCategoria, Descripcion)
SELECT DISTINCT
    LTRIM(RTRIM(Gama)),
    LTRIM(RTRIM(DescripcionTexto))
FROM jardineria_stg.dbo.Stg_Categoria s
WHERE NOT EXISTS (
    SELECT 1 FROM dbo.DimCategoria d WHERE d.NombreCategoria =
    LTRIM(RTRIM(s.Gama))
);

```

### ***Carga de DimProducto***

```

INSERT INTO dbo.DimProducto (CodigoProducto, NombreProducto, ClaveCategoria,
CantidadEnStock, Precio)
SELECT
    LTRIM(RTRIM(p.CodigoProducto)),
    LTRIM(RTRIM(p.Nombre)),
    d.ClaveCategoria,
    p.CantidadEnStock,
    p.PrecioVenta
FROM jardineria_stg.dbo.Stg_Producto p

```



```

LEFT JOIN dbo.DimCategoria d ON d.NombreCategoria = LTRIM(RTRIM(p.Gama))
WHERE NOT EXISTS (
    SELECT 1 FROM dbo.DimProducto dp WHERE dp.CodigoProducto =
LTRIM(RTRIM(p.CodigoProducto))
);

```

### **Carga de DimCliente (SCD Tipo 2)**

-- Insertar clientes nuevos

```

INSERT INTO dbo.DimCliente (ClaveClienteNatural, NombreCliente, Ciudad, Pais,
LimiteCredito, VigenteDesde, VigenteHasta, EsActual)
SELECT
    s.CodigoCliente,
    LTRIM(RTRIM(s.NombreCliente)),
    LTRIM(RTRIM(s.Ciudad)),
    LTRIM(RTRIM(s.Pais)),
    s.LimiteCredito,
    SYSUTCDATETIME(),
    NULL,
    1
FROM jardineria_stg.dbo.Stg_Cliente s
WHERE NOT EXISTS (
    SELECT 1 FROM dbo.DimCliente dc WHERE dc.ClaveClienteNatural = s.CodigoCliente
AND dc.EsActual = 1
);

```

-- Detectar cambios y actualizar versiones

```

SELECT
    s.CodigoCliente, s.NombreCliente, s.Ciudad, s.Pais, s.LimiteCredito
INTO #ClientesCambiados
FROM jardineria_stg.dbo.Stg_Cliente s

```

```
JOIN dbo.DimCliente dc ON dc.ClaveClienteNatural = s.CodigoCliente AND dc.EsActual = 1
WHERE
```

```
    ISNULL(LTRIM(RTRIM(dc.NombreCliente)),") <>
ISNULL(LTRIM(RTRIM(s.NombreCliente)),")
    OR ISNULL(LTRIM(RTRIM(dc.Ciudad)),") <> ISNULL(LTRIM(RTRIM(s.Ciudad)),")
    OR ISNULL(LTRIM(RTRIM(dc.Pais)),") <> ISNULL(LTRIM(RTRIM(s.Pais)),")
    OR ISNULL(dc.LimiteCredito,0) <> ISNULL(s.LimiteCredito,0);
```

```
UPDATE dc
```

```
SET dc.VigenteHasta = SYSUTCDATETIME(), dc.EsActual = 0
```

```
FROM dbo.DimCliente dc
```

```
JOIN #ClientesCambiados cc ON dc.ClaveClienteNatural = cc.CodigoCliente
```

```
WHERE dc.EsActual = 1;
```

```
INSERT INTO dbo.DimCliente (ClaveClienteNatural, NombreCliente, Ciudad, Pais,
LimiteCredito, VigenteDesde, VigenteHasta, EsActual)
```

```
SELECT
```

```
    s.CodigoCliente,
    LTRIM(RTRIM(s.NombreCliente)),
    LTRIM(RTRIM(s.Ciudad)),
    LTRIM(RTRIM(s.Pais)),
    s.LimiteCredito,
    SYSUTCDATETIME(),
    NULL,
    1
```

```
FROM jardineria_stg.dbo.Stg_Cliente s
```

```
JOIN #ClientesCambiados cc ON s.CodigoCliente = cc.CodigoCliente;
```

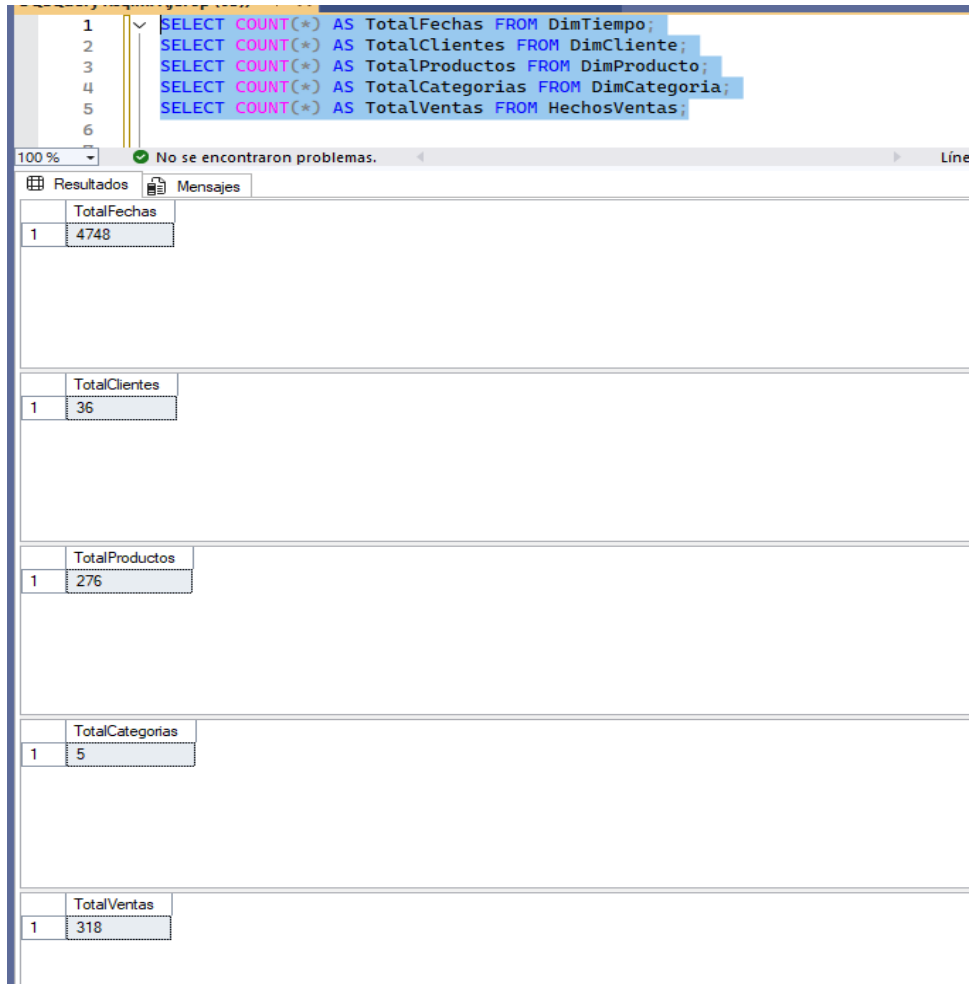
```
DROP TABLE #ClientesCambiados;
```

### ***Carga de HechosVentas***

```
INSERT INTO dbo.HechosVentas (ClaveFecha, ClaveCliente, ClaveProducto, ClaveCategoria,
CodigoPedido, Cantidad, PrecioUnidad, NumeroLinea)
SELECT
    CONVERT(INT, FORMAT(p.FechaPedido,'yyyyMMdd')),
    dc.ClaveCliente,
    dp.ClaveProducto,
    dcat.ClaveCategoria,
    p.CodigoPedido,
    dpd.Cantidad,
    dpd.PrecioUnidad,
    dpd.NumeroLinea
FROM jardineria_stg.dbo.Stg_Detalle_Pedido dpd
JOIN jardineria_stg.dbo.Stg_Pedido p ON dpd.CodigoPedido = p.CodigoPedido
LEFT JOIN dbo.DimCliente dc ON dc.ClaveClienteNatural = p.CodigoCliente AND
dc.EsActual = 1
LEFT JOIN dbo.DimProducto dp ON dp.CodigoProducto = dpd.CodigoProducto
LEFT JOIN dbo.DimCategoria dcat ON dcat.NombreCategoria = (
    SELECT TOP 1 Gama FROM jardineria_stg.dbo.Stg_Producto sp WHERE sp.CodigoProducto
= dpd.CodigoProducto
)
WHERE NOT EXISTS (
    SELECT 1 FROM dbo.HechosVentas hv
    WHERE hv.CodigoPedido = dpd.CodigoPedido
    AND hv.NumeroLinea = dpd.NumeroLinea
    AND hv.ClaveProducto = dp.ClaveProducto
);
```

## Validación de carga

Se ejecutaron consultas de las nuevas tablas para validar que los datos queden almacenados correctamente en el Data Mart



The screenshot displays a SQL Server Enterprise Manager interface. At the top, a query window shows five SQL queries, each using the COUNT(\*) function to verify data in different tables. Below the query window, a status bar indicates 'No se encontraron problemas.' (No problems found). The 'Resultados' (Results) pane shows the output of these queries, with each query result displayed in a separate table.

	TotalFechas
1	4748

	TotalClientes
1	36

	TotalProductos
1	276

	TotalCategorias
1	5

	TotalVentas
1	318

1	SELECT * FROM DimTiempo;
2	SELECT * FROM DimCliente;
3	SELECT * FROM DimProducto;
4	SELECT * FROM DimCategoria;
5	SELECT * FROM HechosVentas;
6	

ClaveFecha	FechaCompleta	Dia	Mes	NombreMes	Trimestre	Año	DiaSemana	EsFinDeSemana
1	20180101	2018-01-01	1	January	1	2018	2	0
2	20180102	2018-01-02	2	January	1	2018	3	0
3	20180103	2018-01-03	3	January	1	2018	4	0
4	20180104	2018-01-04	4	January	1	2018	5	0
5	20180105	2018-01-05	5	January	1	2018	6	0
6	20180106	2018-01-06	6	January	1	2018	7	1
7	20180107	2018-01-07	7	January	1	2018	1	1
8	20180108	2018-01-08	8	January	1	2018	2	0

ClaveCliente	ClaveClienteNatural	NombreCliente	Ciudad	Pais	LimiteCredito	VigenteDesde	VigenteHasta	EsA
1	1	GoldFish Garden	San Francisco	USA	3000.00	2025-09-30 04:52:49.8225434	NULL	1
2	2	Gardening Associates	Miami	USA	6000.00	2025-09-30 04:52:49.8225434	NULL	1
3	3	Gerudo Valley	New York	USA	12000.00	2025-09-30 04:52:49.8225434	NULL	1
4	4	Tendo Garden	Miami	USA	600000.00	2025-09-30 04:52:49.8225434	NULL	1
5	5	...	...	...	...	...	...	...

ClaveProducto	CodigoProducto	NombreProducto	ClaveCategoria	CantidadEnStock	Precio
1	1	Sierra de Poda 400MM	2	15	14.00
2	2	Pala	2	15	14.00
3	3	Rastrillo de Jardín	2	15	12.00
4	4	Azadón	2	15	12.00
5	5	Ajedrea	3	140	1.00
6	6	AR-002 Lavadora Dentata	3	140	1.00

ClaveCategoria	NombreCategoria	Descripcion
1	1	Herbaceas
2	2	Herramientas
3	3	Aromaticas
4	4	Frutales
5	5	Ornamental...

ClaveVenta	ClaveFecha	ClaveCliente	ClaveProducto	ClaveCategoria	CodigoPedido	Cantidad	PrecioUnidad	NumeroLinea	MontoVenta	InsertadoEn
1	1	20090106	7	1	2	13	5	14.00	70.00	2025-09-30 04:...
2	2	20090120	9	1	2	17	5	14.00	70.00	2025-09-30 04:...
3	3	20081230	5	1	2	20	14	14.00	196.00	2025-09-30 04:...
4	4	20080310	4	1	2	32	1	14.00	14.00	2025-09-30 04:...
5	5	20090312	19	1	2	38	5	14.00	70.00	2025-09-30 04:...
6	6	20090105	13	1	2	54	3	14.00	42.00	2025-09-30 04:...
7	7	20090406	30	1	2	94	12	14.00	168.00	2025-09-30 04:...

## Resultados Analíticos

De acuerdo al proceso ETL y el modelo estrella diseñado, se puede concluir que los datos sí están preparados para el análisis porque:

- Se construyó un modelo estrella con dimensiones de Tiempo, Cliente (SCD Tipo 2), Categoría, Producto y una tabla de hechos de ventas.
- Se aplicaron procesos de limpieza, transformación y control de duplicados (ejemplo: uso de LTRIM/RTRIM, control de idempotencia en cargas).
- Se garantizó la integridad referencial entre hechos y dimensiones.
- Se manejó la historización de clientes (SCD Tipo 2), lo que permite analizar cambios en datos maestros a lo largo del tiempo.

Gracias a el Data Mart consolidado, se pueden realizar consultas analíticas para mejorar el rendimiento del negocio, en este caso vamos a identificar el producto más vendido, los clientes más rentables y las ventas por categoría de producto:

### *Identificación del producto más vendido*

```
SELECT TOP 1
```

```
    dp.NombreProducto,
```

```
    SUM(hv.Cantidad) AS TotalVendido
```

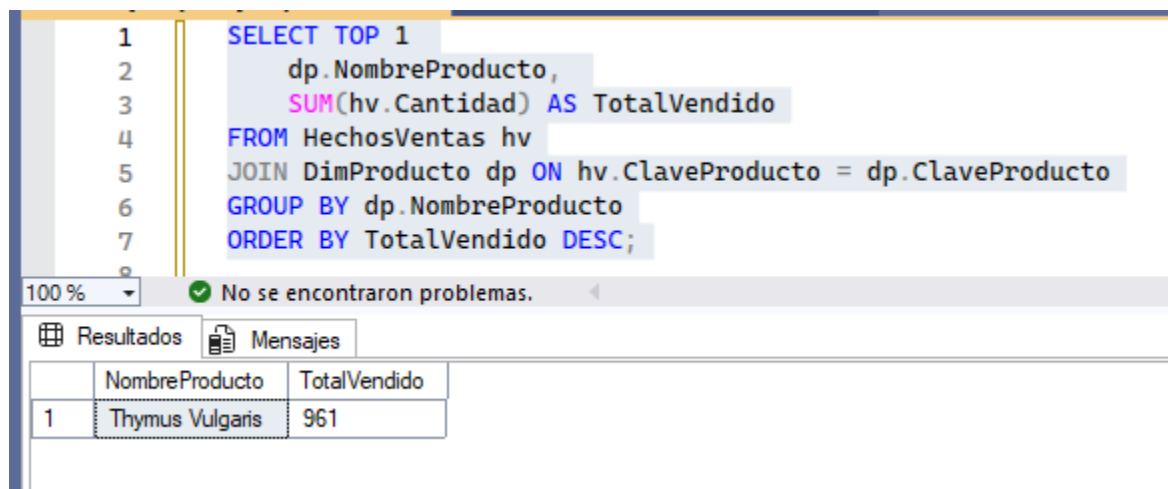
```
FROM HechosVentas hv
```

```
JOIN DimProducto dp ON hv.ClaveProducto = dp.ClaveProducto
```

```
GROUP BY dp.NombreProducto
```

```
ORDER BY TotalVendido DESC;
```

### *Resultado*



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

```
1 SELECT TOP 1
2     dp.NombreProducto,
3     SUM(hv.Cantidad) AS TotalVendido
4 FROM HechosVentas hv
5 JOIN DimProducto dp ON hv.ClaveProducto = dp.ClaveProducto
6 GROUP BY dp.NombreProducto
7 ORDER BY TotalVendido DESC;
```

Below the query, a status bar indicates "No se encontraron problemas." (No problems were found).

At the bottom, the "Resultados" (Results) tab is active, displaying a table with the following data:

	NombreProducto	TotalVendido
1	Thymus Vulgaris	961

### *Cientes más rentables:*

```
SELECT TOP 10 dc.NombreCliente, SUM(hv.MontoVenta) AS TotalCompras
```

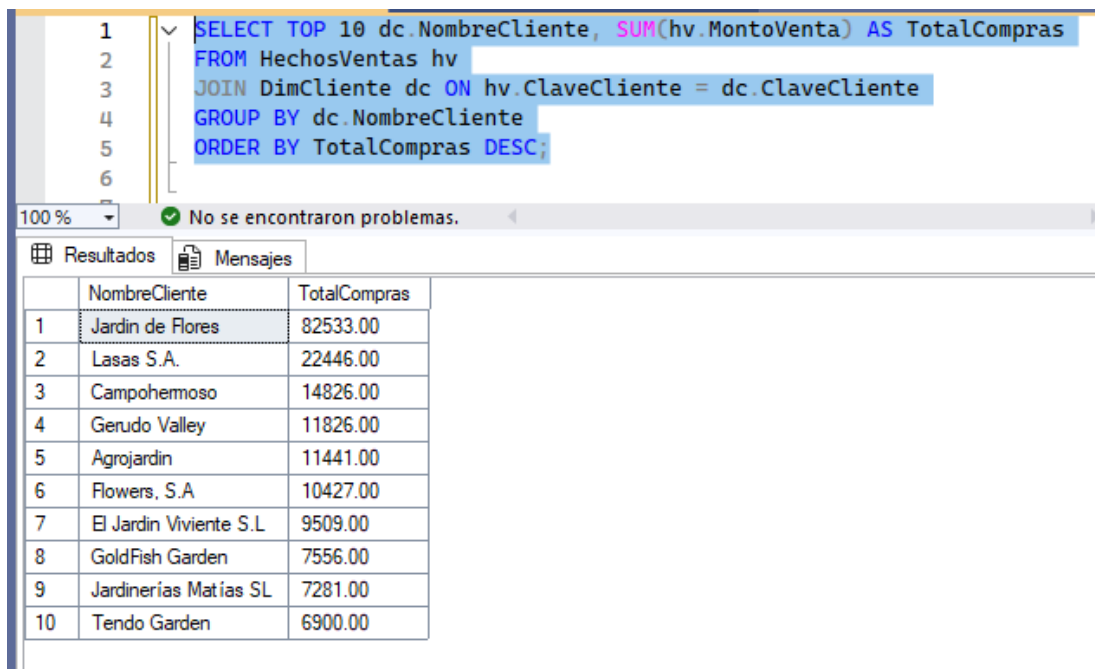
```
FROM HechosVentas hv
```

```
JOIN DimCliente dc ON hv.ClaveCliente = dc.ClaveCliente
```

```
GROUP BY dc.NombreCliente
```

```
ORDER BY TotalCompras DESC;
```

### *Resultado*



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

```
1 SELECT TOP 10 dc.NombreCliente, SUM(hv.MontoVenta) AS TotalCompras
2 FROM HechosVentas hv
3 JOIN DimCliente dc ON hv.ClaveCliente = dc.ClaveCliente
4 GROUP BY dc.NombreCliente
5 ORDER BY TotalCompras DESC;
6
```

Below the query window, a status bar indicates "No se encontraron problemas." (No problems were found).

At the bottom, the "Resultados" (Results) tab is active, displaying a table with the following data:

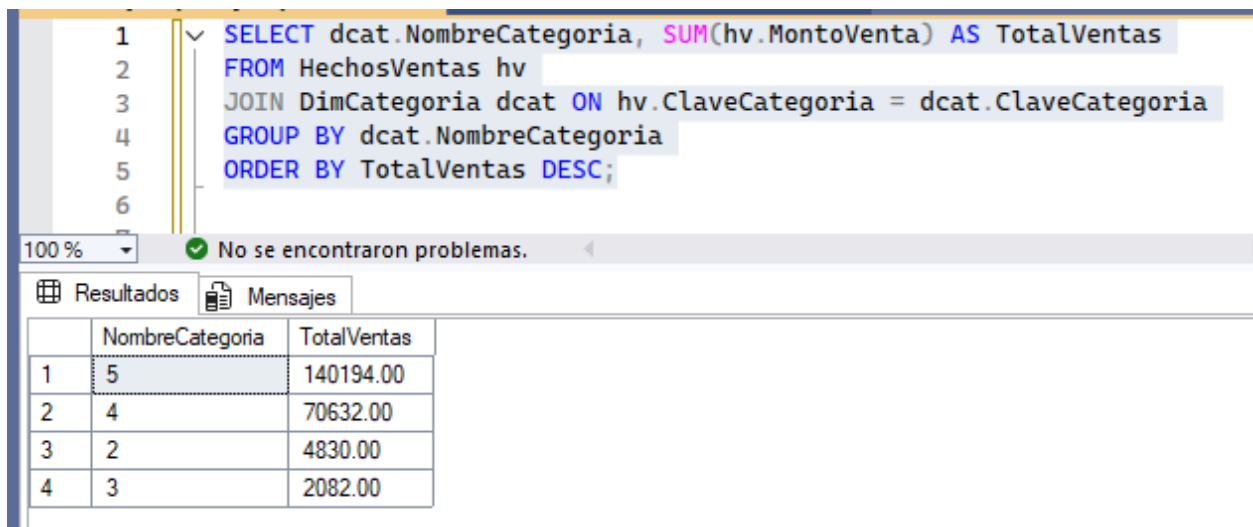
	NombreCliente	TotalCompras
1	Jardin de Flores	82533.00
2	Lasas S.A.	22446.00
3	Campo hemoso	14826.00
4	Gerudo Valley	11826.00
5	Agrojardin	11441.00
6	Flowers, S.A	10427.00
7	El Jardin Viviente S.L	9509.00
8	GoldFish Garden	7556.00
9	Jardinerías Matías SL	7281.00
10	Tendo Garden	6900.00



### *Ventas por categoría de producto*

```
SELECT dcat.NombreCategoria, SUM(hv.MontoVenta) AS TotalVentas  
  
FROM HechosVentas hv  
  
JOIN DimCategoria dcat ON hv.ClaveCategoria = dcat.ClaveCategoria  
  
GROUP BY dcat.NombreCategoria  
  
ORDER BY TotalVentas DESC;
```

### *Resultado*



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

```
1 SELECT dcat.NombreCategoria, SUM(hv.MontoVenta) AS TotalVentas  
2 FROM HechosVentas hv  
3 JOIN DimCategoria dcat ON hv.ClaveCategoria = dcat.ClaveCategoria  
4 GROUP BY dcat.NombreCategoria  
5 ORDER BY TotalVentas DESC;  
6
```

Below the query window, a status bar indicates "No se encontraron problemas." (No problems were found).

At the bottom, the "Resultados" (Results) tab is active, displaying the following table:

	NombreCategoria	TotalVentas
1	5	140194.00
2	4	70632.00
3	2	4830.00
4	3	2082.00

## **RespalDOS**

### ***Back Up de jardineria:***

<https://drive.google.com/file/d/1jdsqkc4xcLrUhRecEr8kU61z7cEUb9rv/view?usp=sharing>

### ***Back Up de jardineria\_stg:***

<https://drive.google.com/file/d/1MJnnRTiZiOwedW0tkLNakxA2kwYzkVy/view?usp=sharing>

### ***Back Up de jardineria\_dm:***

<https://drive.google.com/file/d/1FaO1Hg4kYgaDyemZKjSrlSFyipyd3irz/view?usp=sharing>

## **Conclusiones**

El proceso de construcción del Data Mart de Jardinería permitió consolidar la información de clientes, productos, categorías y pedidos en un esquema de modelo estrella, optimizado para el análisis de ventas, además, se implementaron prácticas de Slowly Changing Dimensión (SCD Tipo 2) para la dimensión de clientes y se aseguró la idempotencia en la carga de hechos. El Data Mart habilita consultas analíticas robustas que apoyan la toma de decisiones, como la identificación de productos más vendidos, tendencias temporales y comportamiento de clientes.

## Bibliografía

*AAMPE - 8 Key Steps to Effectively Build a Data Mart.* (n.d.). <https://aampe.com/blog/8-key-steps-to-effectively-build-a-data-mart>

Yudner Paredes. (2016, December 12). 2. *Diseño y Desarrollo de un DataMart en SQL Server* [Video]. YouTube. <https://www.youtube.com/watch?v=lovGK4E-n1Y>