

Informe Final – Proceso de Transformación de Datos y Carga al Data Mart

Proyecto: Base de Datos Jardinería

Curso: Bases de Datos II (Antonio Jesús Valderrama) - PREICA2502B010064

Estudiante: Alejandro Arango Calderon

Semestre: 4° Ingeniería de Software

Fecha: 24/09/2025

1. Introducción

El presente informe documenta el proceso de **transformación de datos y carga en el Data Mart final** correspondiente al caso de estudio de la **Base de Datos Jardinería**.

El objetivo es consolidar la información de ventas en un **modelo en estrella**, facilitando el análisis de los factores clave del negocio: clientes, productos, empleados, oficinas y tiempo.

2. Preparación

2.1 Modelo Estrella

El modelo definido incluye:

- **Tabla de Hechos:**
 - FactVentas → Contiene las medidas de negocio (total de ventas).
- **Dimensiones:**
 - DimCliente
 - DimProducto
 - DimEmpleado
 - DimOficina
 - DimTiempo

Cada dimensión tiene una clave sustituta (surrogate key) que se relaciona con la tabla de hechos.

2.2 Base de datos de Staging

La base de datos de Staging se utilizó como área intermedia donde se cargaron los datos extraídos del sistema origen, aplicando transformaciones antes de llevarlos al Data Mart final.

3. Extracción de datos (Origen → Staging)

Se emplearon consultas SQL para extraer la información relevante desde la base de datos operacional de Jardinería hacia las tablas de staging.

-- Extracción de clientes

```
INSERT INTO Staging.Cliente_Staging (id_cliente, nombre_cliente, pais, ciudad)
SELECT CodigoCliente, NombreCliente, Pais, Ciudad
FROM Jardineria.Clientes;
```

-- Extracción de productos

```
INSERT INTO Staging.Producto_Staging (id_producto, nombre, gama, precio_venta)
SELECT CodigoProducto, Nombre, Gama, PrecioVenta
FROM Jardineria.Productos;
```

-- Extracción de pedidos con total calculado

```
INSERT INTO Staging.Pedido_Staging (id_pedido, id_cliente, id_empleado,
fecha_pedido, total)
SELECT p.CodigoPedido, p.CodigoCliente, e.CodigoEmpleado, p.FechaPedido,
       SUM(dp.Cantidad * dp.PrecioUnidad) AS total
FROM Jardineria.Pedidos p
JOIN Jardineria.DetallePedidos dp ON p.CodigoPedido = dp.CodigoPedido
JOIN Jardineria.Empleados e ON p.CodigoEmpleado = e.CodigoEmpleado
GROUP BY p.CodigoPedido, p.CodigoCliente, e.CodigoEmpleado, p.FechaPedido;
```

4. Transformación de datos

En la fase de transformación se aplicaron las siguientes técnicas:

1. Limpieza de datos:

- Normalización de mayúsculas/minúsculas en nombres.
- Eliminación de valores nulos o inconsistentes.

2. Normalización de fechas:

- Conversión al formato YYYY-MM-DD.
- Creación de tabla de tiempo con atributos derivados (año, mes, trimestre).

3. Claves sustitutas (surrogate keys):

- Se generaron claves numéricas artificiales para cada dimensión.

-- Normalización de clientes

```
UPDATE Staging.Cliente_Staging
```

```
SET nombre_cliente = UPPER(nombre_cliente);
```

-- Creación de dimensión tiempo

```
INSERT INTO Staging.DimTiempo_Staging (fecha, anio, mes, trimestre)
```

```
SELECT FechaPedido,
```

```
    YEAR(FechaPedido),
```

```
    MONTH(FechaPedido),
```

```
    CASE
```

```
        WHEN MONTH(FechaPedido) BETWEEN 1 AND 3 THEN 1
```

```
        WHEN MONTH(FechaPedido) BETWEEN 4 AND 6 THEN 2
```

```
        WHEN MONTH(FechaPedido) BETWEEN 7 AND 9 THEN 3
```

```
        ELSE 4
```

```
    END
```

```
FROM Staging.Pedido_Staging;
```

5. Carga de datos en el Data Mart (Staging → Final)

Finalmente, se insertaron los registros transformados en el Data Mart final.

-- Carga de clientes

```
INSERT INTO DataMart.DimCliente (sk_cliente, id_cliente, nombre_cliente, pais, ciudad)
```

```
SELECT ROW_NUMBER() OVER (ORDER BY id_cliente) AS sk_cliente,
```

```

        id_cliente, nombre_cliente, pais, ciudad
FROM Staging.Cliente_Staging;

-- Carga de tiempo
INSERT INTO DataMart.DimTiempo (sk_tiempo, fecha, anio, mes, trimestre)
SELECT ROW_NUMBER() OVER (ORDER BY fecha) AS sk_tiempo,
        fecha, anio, mes, trimestre
FROM Staging.DimTiempo_Staging;

-- Carga de hechos
INSERT INTO DataMart.FactVentas (sk_cliente, sk_tiempo, sk_producto, sk_empleado,
sk_oficina, total_venta)
SELECT dc.sk_cliente, dt.sk_tiempo, dp.sk_producto, de.sk_empleado, dof.sk_oficina,
p.total
FROM Staging.Pedido_Staging p
JOIN DataMart.DimCliente dc ON p.id_cliente = dc.id_cliente
JOIN DataMart.DimTiempo dt ON p.fecha_pedido = dt.fecha
JOIN DataMart.DimProducto dp ON p.id_producto = dp.id_producto
JOIN DataMart.DimEmpleado de ON p.id_empleado = de.id_empleado
JOIN DataMart.DimOficina dof ON de.id_oficina = dof.id_oficina;

```

6. Conclusiones

- El proceso ETL permitió consolidar la información de la base de datos Jardinería en un **modelo en estrella optimizado para análisis**.
- La separación en **staging → transformación → Data Mart** aseguró la calidad y consistencia de los datos.
- Con el modelo estrella, ahora es posible analizar las ventas por cliente, producto, oficina, empleado y periodo de tiempo de manera eficiente.