

Título: Creación de una base de datos de Staging

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: 16/09/2025

Introducción

En el marco de la gestión de información empresarial, las bases de datos juegan un papel fundamental en la organización, almacenamiento y análisis de datos relevantes. El presente trabajo tiene como propósito la creación de una base de datos **Staging**, que sirva como espacio intermedio para almacenar datos provenientes de la base de datos **Jardinería**, a fin de optimizar procesos posteriores de análisis y toma de decisiones.

Objetivos

Objetivo general

Diseñar y construir una base de datos Staging para almacenar información relevante extraída de la base Jardinería, garantizando consistencia y calidad en los datos migrados.

Objetivos específicos

- Analizar los datos almacenados en la base Jardinería para determinar cuáles son relevantes.
- Diseñar la estructura de la base de datos Staging con base en el análisis realizado.
- Desarrollar consultas SQL para trasladar los datos desde Jardinería a Staging.
- Validar la correcta carga de los registros en la base de datos Staging.
- Construir respaldos de ambas bases de datos para asegurar la preservación de la información.

Planteamiento del problema

La base de datos Jardinería contiene información de productos, clientes, pedidos y categorías. Sin embargo, esta estructura no está optimizada para un proceso de análisis ágil, ya que contiene datos y tablas que no son relevantes para la etapa de análisis. Se requiere crear una base intermedia (Staging) que contenga únicamente los datos relevantes y limpios para ser posteriormente usados en sistemas de análisis y reportes.

Análisis del problema

Tras revisar la base de datos **Jardinería**, se identificaron las siguientes tablas y campos útiles para la **Staging**:

Tablas relevantes	Campos útiles
producto	CodigoProducto, Nombre, Gama, CantidadEnStock, PrecioVenta
gama_producto (categorías)	Gama, DescripcionTexto.
pedido	CodigoPedido, FechaPedido, CodigoCliente
detalle_pedido	CodigoPedido, CodigoProducto, Cantidad, PrecioUnidad.
cliente	CodigoCliente, NombreCliente, Ciudad, Pais, LimiteCredito.

Las tablas oficina y empleado, aunque útiles para gestión administrativa, no son críticas para este proceso de análisis inicial y no se trasladan a la base Staging.

Propuesta de solución

Correcciones respecto a la entrega previa

- Se incluyen ahora los campos de **pago** como parte de la información relevante.
- Se corrige la relación entre **producto** y **categoria_producto**.
- Se ajusta el mapeo para que las claves primarias coincidan con las foráneas en la base **Staging**.

Estructura de la base de datos Staging

#Tabla Stg_Producto

```
CREATE TABLE Stg_Producto (  
    CodigoProducto VARCHAR(15) PRIMARY KEY,  
    Nombre VARCHAR(70),  
    Gama VARCHAR(50),  
    CantidadEnStock SMALLINT,  
    PrecioVenta DECIMAL(15,2)  
);
```

#Tabla Stg_Categoria (derivada de gama_producto)

```
CREATE TABLE Stg_Categoria (  
    Gama VARCHAR(50) PRIMARY KEY,  
    DescripcionTexto TEXT  
);
```

#Tabla Stg_Pedido

```
CREATE TABLE Stg_Pedido (  
    CodigoPedido INT PRIMARY KEY,  
    FechaPedido DATE,  
    CodigoCliente INT  
);
```

#Tabla Stg_Detalle_Pedido

```
CREATE TABLE Stg_Detalle_Pedido (  
    CodigoPedido INT,  
    CodigoProducto VARCHAR(15),  
    Cantidad INT,  
    PrecioUnidad DECIMAL(15,2),  
    PRIMARY KEY (CodigoPedido, CodigoProducto)  
);
```

#Tabla Stg_Cliente

```
CREATE TABLE Stg_Cliente (  
    CodigoCliente INT PRIMARY KEY,  
    NombreCliente VARCHAR(50),  
    Ciudad VARCHAR(50),  
    Pais VARCHAR(50),  
    LimiteCredito DECIMAL(15,2)  
);
```

Consultas de migración de datos

Cargar tabla categorías

```
INSERT INTO jardineria_stg.dbo.Stg_Categoria (Gama, DescripcionTexto)
SELECT
    cp.gama,
    cp.descripcion
FROM jardineria.dbo.Categoria_producto cp;
```

Cargar tabla Productos

```
INSERT INTO jardineria_stg.dbo.Stg_Producto (CodigoProducto, Nombre, Gama,
CantidadEnStock, PrecioVenta)
SELECT
    p.CodigoProducto,
    p.nombre,
    cp.gama,
    p.cantidad_en_stock,
    p.precio_venta
FROM jardineria.dbo.producto p
JOIN jardineria.dbo.Categoria_producto cp
    ON p.Categoria = cp.ID_Categoria;
```

Cargar tabla Clientes

```
INSERT INTO jardineria_stg.dbo.Stg_Cliente (CodigoCliente, NombreCliente, Ciudad,Pais,
LimiteCredito)
SELECT
    c.ID_cliente,
    c.nombre_cliente,
```

```
c.ciudad,  
c.pais,  
c.limite_credito  
FROM jardineria.dbo.cliente c;
```

Cargar tabla Pedido

```
INSERT INTO jardineria_stg.dbo.Stg_Pedido (CodigoPedido, FechaPedido, CodigoCliente)  
SELECT  
    pe.ID_pedido,  
    pe.fecha_pedido,  
    pe.ID_cliente  
FROM jardineria.dbo.pedido pe;
```

Cargar tabla Detalle Pedidos

```
INSERT INTO jardineria_stg.dbo.Stg_Detalle_Pedido (CodigoPedido, CodigoProducto,  
Cantidad, PrecioUnidad, NumeroLinea)  
SELECT  
    dp.ID_pedido,  
    p.CodigoProducto,  
    dp.cantidad,  
    dp.precio_unidad,  
    dp.numero_linea  
FROM jardineria.dbo.detalle_pedido dp  
JOIN jardineria.dbo.producto p  
    ON dp.ID_producto = p.ID_producto;
```

Validación de carga

Se ejecutaron consultas de las nuevas tablas para validar que los datos queden almacenados correctamente en la Base de datos *Staging*.

PC:jardineria_stg - Stg_Detalle_Pedido SQLQuery4.sql - loc..._stg (PC\User (62)) SQLQuery3.sql - loc..._stg (PC\User (59)) SQLQuery1.sql - loc..._stg (PC\User (60))

```
use jardineria_stg
select * from Stg_Categoria
select * from Stg_Cliente
select * from Stg_Detalle_Pedido
select * from Stg_Pedido
select * from Stg_Producto
```

100 % Results Messages

Gama	DescripcionTesto
1	Herbaceas
2	Herminantes
3	Aromaticas
4	Frutales
5	Ornamentales

CodigoCliente	NombreCliente	Ciudad	País	LímiteCredito
1	GolfFair Garden	San Francisco	USA	3000.00
2	Gardening Associates	Miami	USA	6000.00
3	Gerudo Valley	New York	USA	12000.00
4	Tendo Garden	Miami	USA	60000.00
5	Larrea S.A.	Fuenlabrada	Spain	154310.00
6	Beragua	Madrid	Spain	20000.00
7	Club Golf Puente del Hierro	Madrid	Spain	40000.00
8	Naturagua	Madrid	Spain	32000.00

CodigoPedido	CodigoProducto	Cantidad	PrecioUnidad	NumeroLinea
13	11679	5	14.00	1
17	11679	5	14.00	1
20	11679	14	14.00	1
32	11679	1	14.00	1
38	11679	5	14.00	1
54	11679	3	14.00	1
94	11679	12	14.00	1
115	21636	5	14.00	1

CodigoPedido	FechaPedido	CodigoCliente
1	2006-01-17	5
2	2007-10-23	5
3	2008-08-20	5
4	2009-01-20	5
5	2008-11-09	1
6	2008-12-22	1
7	2009-01-15	3
8	2009-01-20	1

CodigoProducto	Nombre	Gama	CantidadEnStock	PrecioVenta
11679	Sierra de Poda 400MM	2	15	14.00
21636	Pala	2	15	14.00
22225	Rozador de Jardín	2	15	12.00
30310	Azadón	2	15	12.00
ARI-001	Ayres	3	140	1.00
ARI-002	Lavadora Dental	3	140	1.00
ARI-003	Máquina	3	140	1.00

Query executed successfully. localhost (16.0 RTM) PC\User (60) jardineria_stg 00:00:00 750 rows

Respaldos

- **Back Up de jardineria:**

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

g

- **Back Up de jardineria_stg:**

<https://drive.google.com/file/d/1MJnnRTiZiOwedW0tkLNakxA2k-wYzkVy/view?usp=sharing>

Conclusiones

La creación de la base **Staging** permitió seleccionar y organizar de manera eficiente los datos más relevantes de la base de datos **Jardinería**. Esto facilita su uso en análisis posteriores, reduce la complejidad del modelo original y asegura una mejor trazabilidad de clientes, productos, pedidos y pagos.

Anexos

- Script con queries de creación y migración de la base de datos staging:

https://drive.google.com/file/d/1L7wAvTIXCO_sbYnEyuu1Rz2NJoUIKhzy/view?usp=sharing

Bibliografía

Academia De Datos. (2020, July 8). *Cómo hacer una copia de seguridad (backup) y restauración rápida con SQL Server* [Video]. YouTube.

<https://www.youtube.com/watch?v=9I09LRj73Rk>

hdeleon.net. (2019, May 30). *Sql Server nivel Intermedio | Copiar tablas con información entre base de datos | #4* [Video]. YouTube. <https://www.youtube.com/watch?v=Mo7elepKASI>