



Institución Universitaria Digital de Antioquia

Facultad de Ingeniería

Ingeniería de software y datos

BASE DE DATOS II

**S30 - Evidencia de aprendizaje 2. Creación de
una base de datos de Staging**

Autor:

Robinson Zapata Villada

CC:1017242373

Medellín – Colombia

16 de sep. de 2025

Introducción

Para asegurar que los datos de la base operacional Jardinería lleguen limpios, consistentes y preparados a los procesos de análisis, se construyó una base de datos staging intermedia. Esta capa staging actúa como zona de aterrizaje para extraer, depurar y estructurar la información clave-oficinas, empleados, clientes, pedidos, detalle de pedidos, productos, categorías y pagos antes de su posterior carga en cualquier sistema destino. Con la bd staging, evitamos duplicados, corregimos inconsistencias y garantizamos la integridad de los datos.

Objetivos

- **General:** Garantizar la disponibilidad de una capa staging limpia, consistente y estructurada que facilite los procesos ETL y asegure la calidad de los datos extraídos de la base de datos Jardinería.
- **Específicos:**
 - Diseñar y crear la base de datos staging_jardineria con tablas stg_* que recojan únicamente los campos esenciales de la fuente operacional.
 - Desarrollar y ejecutar scripts SQL para extraer, transformar y cargar (ETL) la información desde jardineria hacia staging_jardineria.
 - Depurar y normalizar los datos durante la migración, corrigiendo valores nulos, duplicados o inconsistencias de formato.
 - Validar la integridad y consistencia de los registros mediante conteos y verificaciones de claves foráneas entre origen y staging.
 - Generar respaldos completos de las bases de datos origen y staging para asegurar puntos de restauración confiables.
 - Documentar detalladamente cada paso del proceso, incluyendo scripts, resultados de validación y evidencia de backups, para facilitar la reproducibilidad.

Planteamiento del Problema

La base relacional Jardinería gestiona operaciones diarias, pero adolece de:

- Campos nulos o vacíos en direcciones y comentarios.
- Registros duplicados en clientes.
- Descripciones largas e inconsistentes.
- Relaciones complejas que dificultan los procesos ETL directos.

Sin una capa staging, las cargas posteriores a sistemas de informes o data warehouse se vuelven propensas a errores y demoras.

Análisis del Problema

Al explorar las ocho tablas relacionales, descubrimos:

- oficina: 9 sedes con ciudad, país, código postal.
- empleado: 44 registros con jerarquía y cargo.
- cliente: 90 clientes con datos de contacto y crédito.
- pedido: 260 órdenes con fechas y estados.
- detalle_pedido: 1 090 ítems con cantidades y precios.
- producto: 350 artículos con categoría y precios.
- Categoria_producto: 5 categorías.
- pago: 36 transacciones.

Se detecta lo siguiente:

- Valores nulos en linea_direccion2, fecha_entrega y comentarios.
- Repetición de clientes con igual nombre.
- Tipos TEXT e HTML para descripciones que conviene truncar o ignorar.

Solución

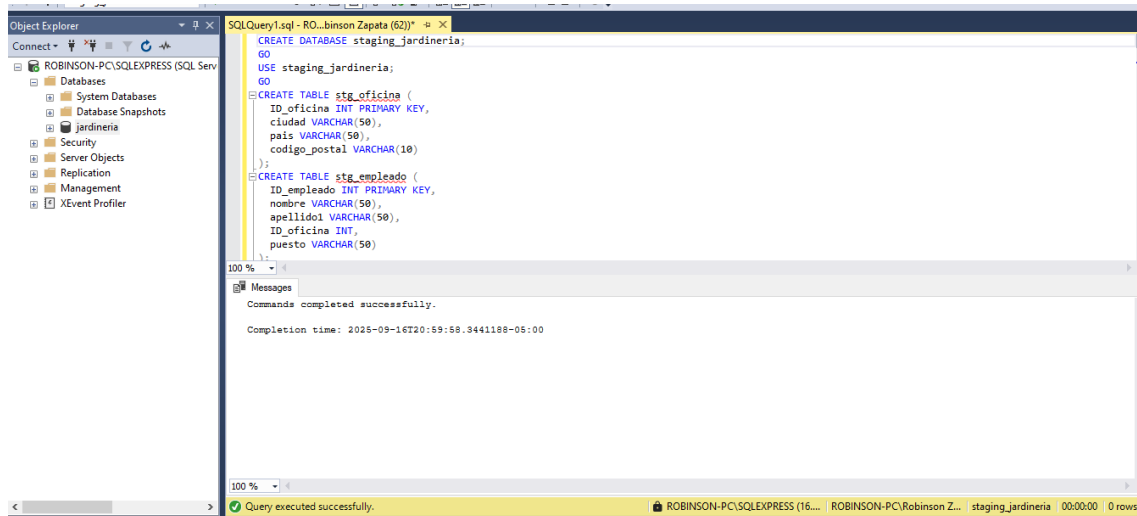
Diseño de la tabla staging; se crea la base staging_jardineria y ocho tablas stg_*

Origen	Staging	Campos retenidos
oficina	stg_oficina	ID_oficina, ciudad, pais, codigo_postal
empleado	stg_empleado	ID_empleado, nombre, apellido1, ID_oficina, puesto
cliente	stg_cliente	ID_cliente, nombre_cliente, ciudad, pais, limite_credito
pedido	stg_pedido	ID_pedido, fecha_pedido, estado, ID_cliente
detalle_pedido	stg_detalle_pedido	ID_detalle_pedido, ID_pedido, ID_producto, cantidad, precio_unidad
producto	stg_producto	ID_producto, CodigoProducto, nombre, Categoria, precio_venta
Categoria_producto	stg_categoria_producto	Id_Categoria, Desc_Categoria
pago	stg_pago	ID_pago, ID_cliente, forma_pago, fecha_pago, total

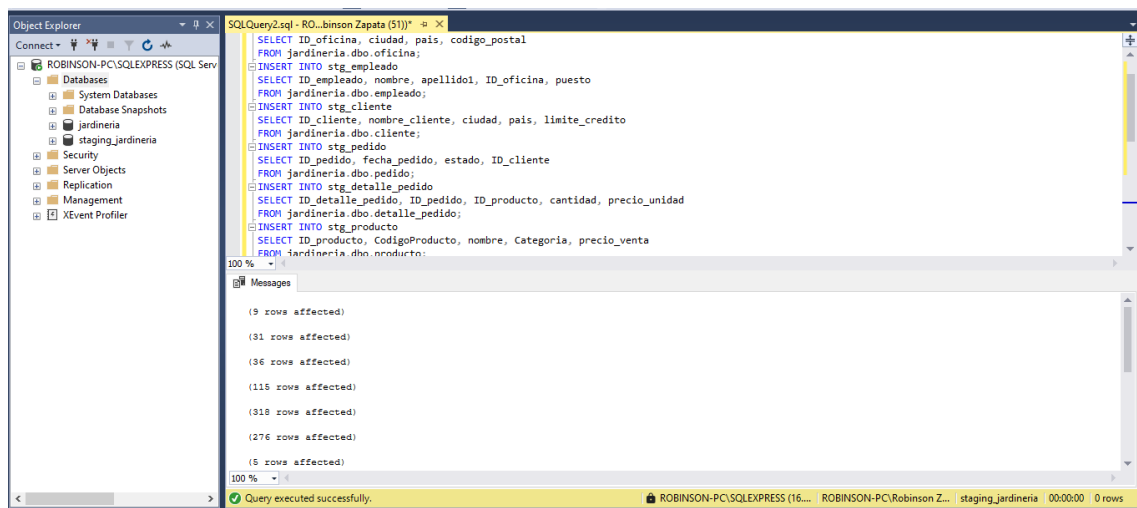
Se crea de esta manera para reducir la complejidad y prepara datos limpios para cualquier uso posterior

Proceso ETL y validación de datos:

- Creación de staging_jardineria: Ver el anexo *Crear_stating.sql*



- Migración de datos: Ver el anexo *Migración.sql*



- Comparación de datos entre las BD

SQLQuery2.sql - RO...binson Zapata (51))

```

SELECT 'stg_oficina' AS Tabla, COUNT(*) AS Registros FROM stg_oficina
UNION ALL
SELECT 'stg_empleado', COUNT(*) FROM stg_empleado
UNION ALL
SELECT 'stg_cliente', COUNT(*) FROM stg_cliente
UNION ALL
SELECT 'stg_pedido', COUNT(*) FROM stg_pedido
UNION ALL
SELECT 'stg_detalle_pedido', COUNT(*) FROM stg_detalle_pedido
UNION ALL
SELECT 'stg_producto', COUNT(*) FROM stg_producto
UNION ALL
SELECT 'stg_categoria_producto', COUNT(*) FROM stg_categoria_producto
UNION ALL
SELECT 'stg_pago', COUNT(*) FROM stg_pago;

```

Tabla	Registros
stg_oficina	9
stg_empleado	31
stg_cliente	36
stg_pedido	115
stg_detalle_pedido	318
stg_producto	276
stg_categoria_producto	5
stg_pago	26

Query executed successfully.

SQLQuery2.sql - RO...binson Zapata (51))

```

SELECT 'oficina', COUNT(*) FROM dbo.oficina
UNION ALL
SELECT 'empleado', COUNT(*) FROM dbo.empleado
UNION ALL
SELECT 'cliente', COUNT(*) FROM dbo.cliente
UNION ALL
SELECT 'pedido', COUNT(*) FROM dbo.pedido
UNION ALL
SELECT 'detalle_pedido', COUNT(*) FROM dbo.detalle_pedido
UNION ALL
SELECT 'producto', COUNT(*) FROM dbo.producto
UNION ALL
SELECT 'Categoria_producto', COUNT(*) FROM dbo.Categoria_producto
UNION ALL
SELECT 'pago', COUNT(*) FROM dbo.pago;

```

(No column name)	(No column name)
oficina	9
empleado	31
cliente	36
pedido	115
detalle_pedido	318
producto	276
Categoria_producto	5
pago	26

Query executed successfully.

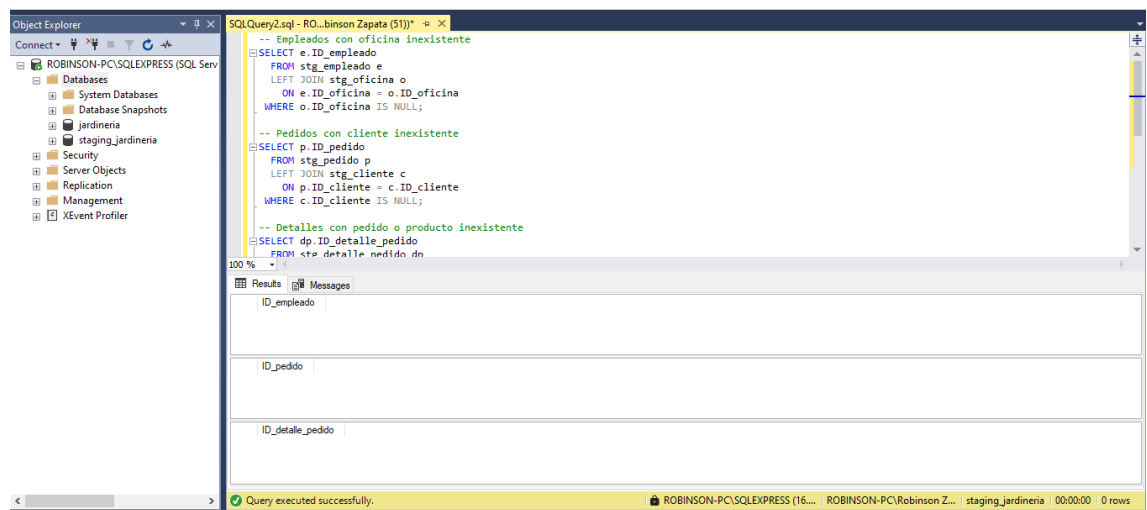
	(No column name)	(No column name)
1	oficina	9
2	empleado	31
3	cliente	36
4	pedido	115
5	detalle_pedido	318
6	producto	276
7	Categoria_producto	5
8	pago	26

BD jardineria

	Tabla	Registros
1	stg_oficina	9
2	stg_empleado	31
3	stg_cliente	36
4	stg_pedido	115
5	stg_detalle_pedido	318
6	stg_producto	276
7	stg_categoria_producto	5
8	stg_pago	26

BD staging_jardineria

- Integridad referencial:

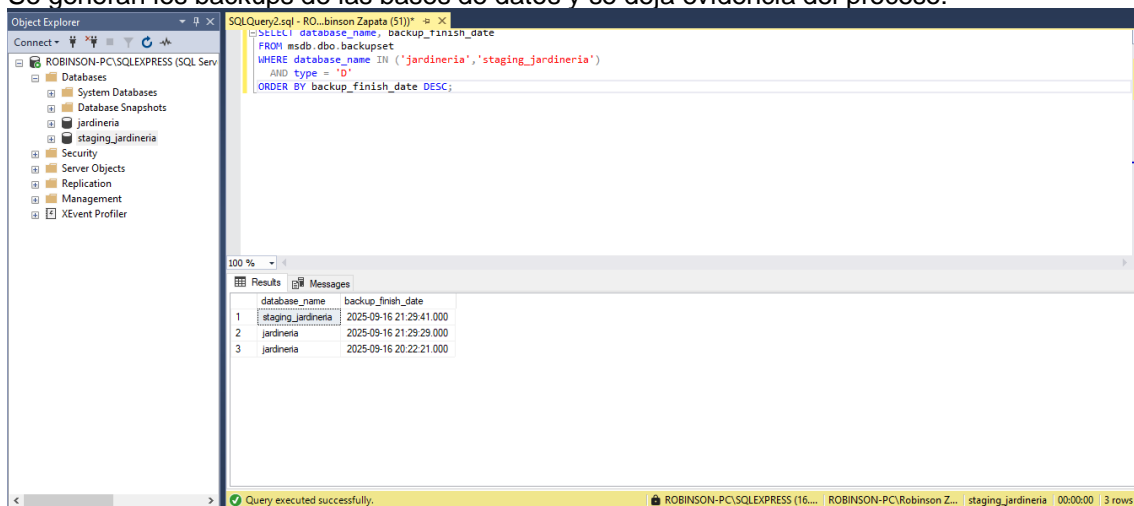


Se validaron valores huérfanos:

- Empleados sin oficina = 0
- Pedidos sin cliente = 0
- Detalles con pedido o producto inexistente = 0

Generación de backup

Se generan los backups de las bases de datos y se deja evidencia del proceso:



Conclusiones:

Al realizar esta actividad se pone en práctica lo aprendido en clase y permite comprender de una manera tangible la importancia de normalizar y limpiar los datos antes de utilizarlos en cualquier tipo de aplicativo.

Más allá de la creación de tablas y sentencias INSERT, el ejercicio reforzó la necesidad de validar cada paso: comparar conteos, detectar registros huérfanos, garantizar la integridad referencial y, finalmente, generar respaldos confiables. Ese ciclo de extracción, carga y verificación consolidó nuestra confianza tanto en el resultado técnico como en la trazabilidad de todo el proceso, lo que es clave para mantener la gobernanza de datos.

Bibliografía:

- Hernández, M. (2020). *Diseño de Bases de Datos Relacionales*. Editorial Académica.
- Microsoft Docs. (2025). *Patterns for staging and ETL processes*.
<https://docs.microsoft.com>