

Evidencia de Aprendizaje 2
Creación de una Base de Datos Staging

Justin Beckham Cardona

Jorge Armando Rodríguez

Lina Johana Seguro

Facultad de Ingeniería y Ciencias Agropecuarias,

Institución Universitaria Digital de Antioquia

064: Bases de Datos II

Antonio Jesús Valderrama

16 de septiembre de 2025

Introducción

En la actualidad, las organizaciones necesitan procesar grandes cantidades de datos transaccionales e información útil para la toma de decisiones. Para ello, resulta necesario crear áreas intermedias como las bases de datos staging, las cuales permiten centralizar, limpiar y armonizar la información antes de cargarla al Data Warehouse o al Data Mart.

En este documento se presenta la construcción de una base de datos Staging para la base de datos origen Jardinería. La Staging será utilizada como área intermedia para almacenar los datos de origen limpios y armonizados antes de su carga al Data Warehouse o data mart. Se describen el análisis de los datos, el diseño de las tablas, los scripts SQL para poblarlas, las validaciones y los backups solicitados.

Objetivos

Objetivo General:

Construir una base de datos Staging a partir de Jardinería, trasladando y validando los datos necesarios para alimentar el Data Mart de ventas.

Objetivos Específicos:

1. Analizar las tablas y campos de la base de datos Jardinería para identificar los datos relevantes que deben ser trasladados a la base de datos Staging.
2. Diseñar la estructura de las tablas staging y construir las consultas SQL necesarias para poblarlas y validar su correcta carga.
3. Generar respaldos de ambas bases de datos y documentar el proceso de construcción de la base Staging conforme a las normas APA, anexando los scripts como evidencia.

Planteamiento del Problema

La base de datos Jardinería la cual fue proporcionada para la actividad, fue diseñada originalmente para la gestión transaccional de operaciones como pedidos, clientes y productos. Este es un modelo adecuado para trabajar diariamente, pero no resulta eficiente si se quiere realizar análisis de negocio, ya que, para consultar información específica como el producto más vendido, la categoría con mayor cantidad de ventas o el año más exitoso requiere combinar información de varias partes de la base de datos, lo que complica el proceso y puede afectar el rendimiento. Además, esto dificulta la toma de decisiones en áreas como ventas, gestión de clientes y administración de inventario.

Debido a ello, para construir un Data Warehouse o un Data Mart se requiere una capa intermedia llamada Staging. Esta permite centralizar los datos relevantes para las transformaciones posteriores, validar la información y rehacer las cargas sin afectar la base de origen. Además, facilita la auditoría y la recuperación mediante respaldos.

Análisis del Problema

La base de datos origen Jardinería contiene tablas transaccionales y maestras típicas de un negocio de jardinería, como clientes, productos, pedidos, detalle de pedidos, empleados, oficinas, pagos y categorías de producto. El objetivo de esta actividad es construir una base de datos Staging que centralice los datos relevantes sin transformar aún a un modelo dimensional, preserve la trazabilidad mediante los mismos IDs de origen, permita validar y re-ejecutar los procesos ETL sin afectar la base de producción, y facilite las posteriores cargas al Data Warehouse.

Durante el análisis inicial se detectaron varios problemas que podrían afectar la correcta construcción de la Staging: intentos de insertar valores en columnas IDENTITY sin configurar SET IDENTITY_INSERT, inserciones con distinto número de columnas entre la lista del INSERT y la del SELECT, conflictos de claves foráneas al cargar hechos antes que dimensiones maestras, uso incorrecto de nombres de columnas y objetos referenciados en distintos contextos de base de datos, y falta de una secuencia de carga adecuada (primero tablas maestras y luego transaccionales). Estos problemas dificultan una carga reproducible y auditada de los datos en la Staging, lo que resalta la necesidad de un diseño y ejecución cuidadosos del proceso.

Propuesta de Solución

Para construir la base de datos Staging se realizó un análisis detallado de las tablas de la base de datos Jardinería, con el objetivo de identificar los campos que resultan relevantes para el Data Mart de ventas. Se seleccionaron las tablas maestras y transaccionales que contienen la información necesaria para responder las preguntas de negocio planteadas: producto más vendido, categoría con más productos y año con mayores ventas. Cada tabla staging fue diseñada para reflejar únicamente los campos esenciales, conservando los identificadores originales y asegurando la trazabilidad desde la fuente hasta la capa intermedia. Además, se creó una base de Staging separada, denominada Jardineria_Staging, lo que evita riesgos sobre la base de datos productiva y facilita la realización de respaldos puntuales de la capa intermedia. Esta selección permite centralizar los datos relevantes, facilitar la validación de la carga y optimizar el posterior traslado al Data Warehouse.

Correcciones a la entrega uno:

- **IDENTITY:** en vez de usar columnas IDENTITY en las tablas Staging, se definieron PK INT normales para permitir insertar las claves originales sin SET IDENTITY_INSERT.
- **Coincidencia de columnas:** se corrigieron todos los INSERT INTO ... (cols) SELECT ... para que la cantidad y orden de columnas coincidiera exactamente con el SELECT.
- **Orden de carga:** se estableció que las tablas maestras (categoría, producto, cliente, empleado, oficina) se cargan antes que las transaccionales (pedido, detalle_pedido, pago) para no violar integridad referencial.

- Nombres correctos: se adaptaron los nombres de columnas/objetos al DDL provisto (ej.: `CodigoProducto`, `Desc_Categoria`), evitando `Invalid object name` y `Invalid column name`.
- Validaciones: se implementaron consultas de conteo y detección de huérfanos tras la carga, permitiendo detectar y resolver problemas antes de continuar.

Descripción del análisis realizado a los datos Jardinería y traslado a BD Staging

Para construir la base de datos Staging se realizó un análisis detallado de las tablas de la base de datos Jardinería, con el objetivo de identificar los campos que resultan relevantes para el Data Mart de ventas. Se seleccionaron las tablas maestras y transaccionales que contienen la información necesaria para responder las preguntas de negocio planteadas: producto más vendido, categoría con más productos y año con mayores ventas. Cada tabla Staging fue diseñada para reflejar únicamente los campos esenciales, conservando los identificadores originales y asegurando la trazabilidad desde la fuente hasta la capa intermedia. Además, se creó una base de Staging separada, denominada `Jardineria_Staging`, lo que evita riesgos sobre la base de datos productiva y facilita la realización de respaldos puntuales de la capa intermedia. Los datos se trasladaron mediante consultas `INSERT`, `SELECT`, copiando únicamente los campos relevantes y realizando la carga primero en las tablas maestras y luego en las transaccionales, con validaciones intermedias de conteo y registros huérfanos, asegurando la integridad y preparación de los datos para su posterior uso en el Data Warehouse.

1. Tablas relevantes y campos seleccionados:

Se revisó la estructura y el contenido de las tablas en jardinería y se decidió trasladar los siguientes campos a Staging:

- **Cliente** se trasladaron a la Staging únicamente los *campos ID_cliente, nombre_cliente, ciudad y pais*, con el fin de realizar análisis por cliente y ubicación geográfica, manteniendo la trazabilidad de los registros hacia el Data Mart. Se omite el campo adicional, *limite_credito*, ya que no son necesarios para responder las preguntas de negocio planteadas.
- **Producto** se incluyeron los campos *ID_producto, CodigoProducto, nombre, Categoria y precio_venta*, permitiendo identificar cada producto de manera única, clasificarlo por categoría y disponer de la información de precios para los análisis de ventas. Se eliminaron campos opcionales como *dimensiones, proveedor, precio_proveedor y cantidad_en_stock* para simplificar la Staging, dado que no influyen directamente en las métricas principales del Data Mart.
- **Empleado** se trasladaron los campos *ID_empleado, nombre, apellido y ID_oficina*, lo que permite analizar las ventas por vendedor, mientras que el campo *puesto* se omite por no ser esenciales para el modelo estrella.
- **Pedido** se conservaron *ID_pedido, fecha_pedido e ID_cliente*, necesarios para relacionar los pedidos con los clientes y la dimensión tiempo, eliminando columnas como *fecha_esperada, fecha_entrega y estado*, que no son necesarias para las métricas de ventas principales.
- **Detalle_Pedido** se seleccionaron *ID_detalle_pedido, ID_pedido, ID_producto, cantidad y precio_unidad*, permitiendo calcular las métricas de ventas de manera precisa. Se omitió *numero_linea*, ya que no aporta al análisis del Data Mart.
- **Oficina** se seleccionaron los campos *ID_oficina, ciudad, país y región*, lo que permite realizar análisis de carácter geográfico y vincular a los empleados con su

respectiva localización. Se omitieron otros posibles campos administrativos, ya que no son necesarios para el modelo estrella ni para las métricas principales del Data Mart.

2. Mapeo de columnas entre tablas origen y Staging

Tabla 1

Mapeo de columnas entre tablas de origen y tablas Staging

Tabla Staging	Columnas Staging	SELECT desde base de datos origen (dbo)
Stg_Cliente	id_cliente, nombre_cliente, ciudad, pais, limite_credito	SELECT ID_cliente, nombre_cliente, ciudad, pais, limite_credito FROM dbo.cliente
Stg_Producto	id_producto, nombre_producto, categoria, precio_venta	SELECT ID_producto, nombre, c.Desc_Categoria, precio_venta FROM dbo.producto p JOIN dbo.Categoria_producto c ON c.Id_Categoria = p.Categoria
Stg_Pedido	id_pedido, fecha_pedido, estado, id_cliente	SELECT ID_pedido, fecha_pedido, estado, ID_cliente FROM dbo.pedido
Stg_DetallePedido	id_pedido, id_producto, cantidad, precio_unidad	SELECT ID_pedido, ID_producto, cantidad, precio_unidad FROM dbo.detalle_pedido
Stg_Empleado	id_empleado, nombre, apellido, puesto, id_oficina	SELECT ID_empleado, nombre, apellido1, puesto, ID_oficina FROM dbo.empleado
Stg_Oficina	id_oficina, ciudad, pais, region	SELECT ID_oficina, ciudad, pais, region FROM dbo.oficina

Nota. En los scripts posteriores algunas tablas Staging serán ampliadas con más columnas (por ejemplo: direcciones o correo electrónico) para conservar información adicional; sin embargo, para la entrega EA2 se mantuvieron únicamente las columnas aquí presentadas.

3. Orden de Carga

Se siguió un orden sistemático para la carga de las tablas en la base de datos

Jardineria_Staging, garantizando la integridad de los datos y facilitando el seguimiento del proceso.

- Cargar primero las tablas maestras: *Stg_Producto*, *Stg_Cliente*, *Stg_Empleado*, *Stg_Oficina*.
- Cargar las tablas transaccionales: *Stg_Pedido*, *Stg_DetallePedido*.
- Validaciones intermedias (conteos, cheques de huérfanos) entre pasos.
- Ajuste de INSERTS: asegurar que la lista de columnas en el INSERT coincida exactamente con las columnas retornadas por el SELECT.
- Uso de nombres y esquemas correctos: referenciar *jardineria.dbo.tabla* y *Jardineria_Staging.dbo.Stg_tabla* si se ejecuta desde otra BD, o usar USE para posicionarse en la BD correcta.

4. Validaciones

- Comparaciones de COUNT (*) entre *jardineria.dbo.xxx* y *Jardineria_Staging.dbo.Stg_xxx*. Se comprobó que los números coinciden: clientes 36/36, productos 276/276, pedidos 115/115, detalle_pedido 318/318, empleados 31/31, oficinas 9/9.
- Consultas para detectar registros huérfanos (detalle sin pedido, pedido sin cliente).
 - Backups: Se generaron los backups completos de ambas bases (origen y staging) y se adjuntaron en los anexos para la evidencia.

Anexos

En esta sección se incluyen las capturas de pantalla generadas durante el desarrollo del proyecto, con el fin de evidenciar los procesos realizados. Se incluyen los respaldos (*backups*) de las dos bases de datos empleadas, así como el documento con los scripts de creación y carga de la base de datos Staging. Estos anexos permiten garantizar la trazabilidad, la reproducibilidad del proceso y la verificación de la estructura propuesta en el modelo de datos.

- Creación de base de datos Jardinería_ Staging:

```
-- Tabla Staging Clientes
CREATE TABLE stg_clientes (
    id_cliente INT,
    nombre_cliente VARCHAR(100),
    ciudad VARCHAR(50),
    pais VARCHAR(50),
    limite_credito DECIMAL(10,2)
);

-- Tabla Staging Productos
CREATE TABLE stg_productos (
    id_producto INT,
    nombre_producto VARCHAR(100),
    gama VARCHAR(50),
    cantidad_en_stock INT,
    precio_venta DECIMAL(10,2)
);

-- Tabla Staging Pedidos
CREATE TABLE stg_pedidos (
    id_pedido INT,
    fecha_pedido DATE,
    id_cliente INT,
    id_empleado INT
);

-- Tabla Staging Detalle Pedido
CREATE TABLE stg_detalle_pedido (
    id_pedido INT,
    id_producto INT,
    cantidad INT,
    precio_unidad DECIMAL(10,2)
);

-- Tabla Staging Empleados
CREATE TABLE stg_empleados (
    id_empleado INT,
    nombre VARCHAR(50),
    apellido VARCHAR(50),
    puesto VARCHAR(50),
    id_oficina INT
);
```

```
-- Tabla Staging Oficinas
CREATE TABLE stg_oficinas (
    id_oficina INT,
    ciudad VARCHAR(50),
    pais VARCHAR(50),
    region VARCHAR(50)
);
```

- Poblado Datos BD Jardinería _Staging:

```

-- Poblar Stg_Cliente
✓ INSERT INTO dbo.Stg_Cliente (id_cliente, nombre_cliente, ciudad, pais, limite_credito)
  SELECT
    ID_cliente,
    nombre_cliente,
    ciudad,
    pais,
    limite_credito
  FROM dbo.cliente;

-- Poblar Stg_Producto
✓ INSERT INTO dbo.Stg_Producto (id_producto, nombre_producto, categoria, precio_venta)
  SELECT
    ID_producto,
    nombre,
    c.Desc_Categoria,
    precio_venta
  FROM dbo.producto p
  JOIN dbo.Categoria_producto c ON c.Id_Categoria = p.Categoria;

-- Poblar Stg_Pedido
✓ INSERT INTO dbo.Stg_Pedido (id_pedido, fecha_pedido, estado, id_cliente)
  SELECT
    ID_pedido,
    fecha_pedido,
    estado,
    ID_cliente
  FROM dbo.pedido;

-- Poblar Stg_DetallePedido
✓ INSERT INTO dbo.Stg_DetallePedido (id_pedido, id_producto, cantidad, precio_unidad)
  SELECT
    ID_pedido,
    ID_producto,
    cantidad,
    precio_unidad
  FROM dbo.detalle_pedido;

-- Poblar Stg_Empleado
✓ INSERT INTO dbo.Stg_Empleado (id_empleado, nombre, apellido1, puesto, id_oficina)
  SELECT
    ID_empleado,
    nombre,
    apellido1,
    puesto,
    ID_oficina
  FROM dbo.empleado;

```

```
-- Poblar Stg_Oficina
INSERT INTO dbo.Stg_Oficina (id_oficina, ciudad, pais, region)
SELECT
    ID_oficina,
    ciudad,
    pais,
    region
FROM dbo.oficina;
```

- Validaciones Tablas Jardinería y Jardinería_Staging:

```
-- Validación de Clientes
SELECT 'Clientes' AS Tabla,
    (SELECT COUNT(*) FROM dbo.cliente) AS Registros_Original,
    (SELECT COUNT(*) FROM dbo.Stg_Cliente) AS Registros_Staging;

-- Validación de Productos
SELECT 'Productos' AS Tabla,
    (SELECT COUNT(*) FROM dbo.producto) AS Registros_Original,
    (SELECT COUNT(*) FROM dbo.Stg_Producto) AS Registros_Staging;

-- Validación de Pedidos
SELECT 'Pedidos' AS Tabla,
    (SELECT COUNT(*) FROM dbo.pedido) AS Registros_Original,
    (SELECT COUNT(*) FROM dbo.Stg_Pedido) AS Registros_Staging;

-- Validación de Detalle de Pedidos
SELECT 'Detalle Pedido' AS Tabla,
    (SELECT COUNT(*) FROM dbo.detalle_pedido) AS Registros_Original,
    (SELECT COUNT(*) FROM dbo.Stg_DetallePedido) AS Registros_Staging;

-- Validación de Empleados
SELECT 'Empleados' AS Tabla,
    (SELECT COUNT(*) FROM dbo.empleado) AS Registros_Original,
    (SELECT COUNT(*) FROM dbo.Stg_Empleado) AS Registros_Staging;

-- Validación de Oficinas
SELECT 'Oficinas' AS Tabla,
    (SELECT COUNT(*) FROM dbo.oficina) AS Registros_Original,
    (SELECT COUNT(*) FROM dbo.Stg_Oficina) AS Registros_Staging;
```


- Resultados obtenidos a las Validaciones:

100 %			
6 0			
Resultados Mensajes			
	Tabla	Registros_Original	Registros_Staging
1	Cientes	36	36
	Tabla	Registros_Original	Registros_Staging
1	Productos	276	276
	Tabla	Registros_Original	Registros_Staging
1	Pedidos	115	115
	Tabla	Registros_Original	Registros_Staging
1	Detalle Pedido	318	318
	Tabla	Registros_Original	Registros_Staging
1	Empleados	31	31
	Tabla	Registros_Original	Registros_Staging
1	Oficinas	9	9

- Link a Backups de ambas Bases de datos:

[Link BK Jardinería](#)

[Link BK Jardienría-Staging](#)

- Evidencia de consultas que dan respuesta adecuadamente a las tres preguntas trabajadas: producto más vendido, categoría con mas productos y año con mayores ventas:

```

SELECT TOP 1
    p.id_producto,
    p.nombre_producto AS NombreProducto,
    SUM(d.cantidad) AS Total_Unidades_Vendidas
FROM dbo.Stg_DetallePedido d
JOIN dbo.Stg_Producto p ON d.id_producto = p.id_producto
GROUP BY p.id_producto, p.nombre_producto
ORDER BY Total_Unidades_Vendidas DESC;

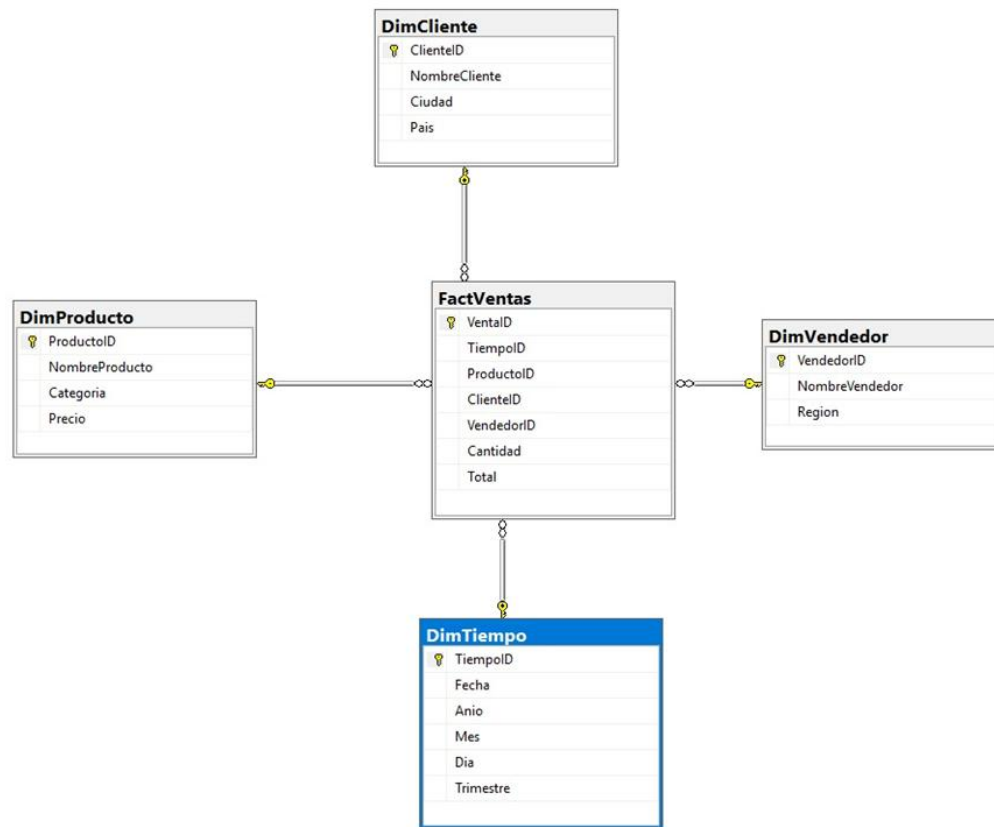
SELECT TOP 1
    categoria AS Categoria_Nombre,
    COUNT(*) AS Numero_Productos
FROM dbo.Stg_Producto
GROUP BY categoria
ORDER BY Numero_Productos DESC;

SELECT TOP 1
    YEAR(p.fecha_pedido) AS Anio,
    SUM(d.cantidad * d.precio_unidad) AS Total_Ventas
FROM dbo.Stg_DetallePedido d
JOIN dbo.Stg_Pedido p ON d.id_pedido = p.id_pedido
GROUP BY YEAR(p.fecha_pedido)
ORDER BY Total_Ventas DESC;

```

Resultados		Mensajes	
	id_producto	NombreProducto	Total_Unidades_Vendidas
1	13	Thymus Vulgaris	961
	Categoria_Nombre	Numero_Productos	
1	Ornamentales	154	
	Anio	Total_Ventas	
1	2008	118558.00	

- De acuerdo a la Base de Datos Jardinería_Staging se realizó el modelo estrella



Conclusiones

- **Objetivo 1:** Analizar las tablas y campos de la base de datos Jardinería para identificar los datos relevantes que deben ser trasladados a la base de datos Staging.

Conclusión

El análisis permitió seleccionar únicamente los campos necesarios para responder a las preguntas de negocio, garantizando la trazabilidad y simplificación de la base Staging.

- **Objetivo 2:** Diseñar la estructura de las tablas Staging y construir las consultas SQL necesarias para poblarlas y validar su correcta carga.

Conclusión

Se definió la estructura de la base Staging y, mediante consultas SQL, se logró la carga correcta de los datos, validada con conteos y controles de integridad.

- **Objetivo 3:** Generar respaldos de ambas bases de datos y documentar el proceso de construcción de la base Staging conforme a las normas APA, anexando los scripts como evidencia.

Conclusión

Se generaron los respaldos y se documentó el proceso siguiendo las normas APA, anexando capturas y scripts como evidencia del trabajo realizado.

Bibliografía

Kimball, R., & Ross, M. (2013). *The data warehouse toolkit: The definitive guide to dimensional modeling* (3.^a ed.). Wiley.

Inmon, W. H. (2005). *Building the data warehouse* (4.^a ed.). Wiley.

Microsoft. (s. f.). *Backup and restore of SQL Server databases*. Microsoft Docs.

Recuperado de <https://learn.microsoft.com/sql/backup-restore/backup-and-restore-sql-server-databases>