1. Introducción

La base de datos **Jardinería** contiene información sobre clientes, empleados, productos y pedidos de una empresa ficticia de distribución. Para mejorar los procesos de análisis y reportes, se construye una **base de datos Staging** que centraliza los datos relevantes y sirve como capa intermedia antes de cargarlos en un Data Warehouse o usarlos en analítica.

---

2. Objetivos

🎯 Objetivo General

Construir una base de datos **Staging** basada en el modelo relacional de Jardinería para optimizar la integración y análisis de los datos.

🎯 Objetivos Específicos

- Analizar las tablas de Jardinería y seleccionar los datos relevantes.
- Diseñar la estructura de tablas en la base Staging.
- Construir las consultas SQL para crear la base Staging y migrar los datos desde Jardinería.
- Validar que los datos se almacenen correctamente.
- Realizar un respaldo de ambas bases de datos.

---

3. Planteamiento del Problema

La base **Jardinería** tiene la información distribuida en múltiples tablas con un modelo pensado para operaciones transaccionales. Esto dificulta su explotación directa para análisis y reportes. Por ello, se requiere una **base intermedia (Staging)** que organice los datos clave y permita consultas más rápidas y limpias.

---

4. Análisis del Problema

Tras revisar Jardinería, se identificó que las tablas relevantes son:

- **Clientes**: idCliente, nombreCliente, ciudad, país.
- **Empleados**: idEmpleado, nombre, puesto, oficina.
- **Productos**: idProducto, nombre, gama, cantidadEnStock, precioVenta.
- **Pedidos**: idPedido, fechaPedido, estado, idCliente.

- **DetallesPedido**: idPedido, idProducto, cantidad, precioUnidad.

Estos datos se trasladan a la base **Staging**, omitiendo campos redundantes o de poco valor analítico.

---

5. Propuesta de Solución

🔷 **Creación de la base de datos Staging**

```
-- Crear base de datos Staging
CREATE DATABASE Jardineria_Staging;
USE Jardineria_Staging;
```

🔷 **Tablas de Staging y migración de datos**

Clientes

```
CREATE TABLE Clientes_Staging (
   idCliente INT PRIMARY KEY,
   nombreCliente VARCHAR(100),
   ciudad VARCHAR(50),
   pais VARCHAR(50)
);

INSERT INTO Clientes_Staging (idCliente, nombreCliente, ciudad, pais)
SELECT idCliente, nombreCliente, ciudad, pais
FROM Jardineria.dbo.Clientes;
```

Empleados

```
CREATE TABLE Empleados_Staging (
   idEmpleado INT PRIMARY KEY,
   nombre VARCHAR(100),
   puesto VARCHAR(50),
   oficina VARCHAR(50)
);
```

```
INSERT INTO Empleados_Staging (idEmpleado, nombre, puesto, oficina)
SELECT idEmpleado, CONCAT(nombre, ' ', apellido1), puesto, codigoOficina
FROM Jardineria.dbo.Empleados;
```

Productos

```
CREATE TABLE Productos_Staging (
    idProducto VARCHAR(15) PRIMARY KEY,
    nombre VARCHAR(100),
    gama VARCHAR(50),
    cantidadEnStock INT,
    precioVenta DECIMAL(10,2)
);

INSERT INTO Productos_Staging (idProducto, nombre, gama, cantidadEnStock,
precioVenta)
SELECT idProducto, nombre, gama, cantidadEnStock, precioVenta
FROM Jardineria.dbo.Productos;
```

Pedidos

```
CREATE TABLE Pedidos_Staging (
    idPedido INT PRIMARY KEY,
    fechaPedido DATE,
    estado VARCHAR(15),
    idCliente INT
);

INSERT INTO Pedidos_Staging (idPedido, fechaPedido, estado, idCliente)
SELECT idPedido, fechaPedido, estado, idCliente
FROM Jardineria.dbo.Pedidos;
```

Detalles de Pedidos

```
CREATE TABLE DetallesPedido_Staging (
   idPedido INT,
   idProducto VARCHAR(15),
   cantidad INT,
   precioUnidad DECIMAL(10,2),
   PRIMARY KEY (idPedido, idProducto)
);


INSERT INTO DetallesPedido_Staging (idPedido, idProducto, cantidad, precioUnidad)
SELECT idPedido, idProducto, cantidad, precioUnidad
FROM Jardineria.dbo.DetallePedidos;
```

---

6. Validación de Datos

Para comprobar la correcta migración se compararon los conteos entre Jardinería y Staging:

```
SELECT COUNT(*) FROM Jardineria.dbo.Clientes;
SELECT COUNT(*) FROM Jardineria_Staging.dbo.Clientes_Staging;
```

👉 Los resultados mostraron que el número de registros es igual en ambas bases.

---

7. Backup de Bases de Datos

```
-- Backup de la base original
BACKUP DATABASE Jardineria
TO DISK = 'C:\Backup\Jardineria.bak';


-- Backup de la base Staging
BACKUP DATABASE Jardineria_Staging
TO DISK = 'C:\Backup\Jardineria_Staging.bak';
```

---

8. Conclusiones

- Se construyó la base Staging a partir de Jardinería, con tablas simplificadas.
- Los datos se migraron correctamente, manteniendo integridad y consistencia.
- El modelo Staging facilita análisis posteriores y es útil como base intermedia para un Data Warehouse.

---

9. Bibliografía

- Coronel, C., & Morris, S. (2019). *Database Systems: Design, Implementation, & Management*. Cengage Learning.
- Microsoft. (2025). *BACKUP (Transact-SQL)*. Recuperado de https://learn.microsoft.com/sql