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

Elaborado por:

Damián Pérez Salvatierra

Catalina Vinasco Arredondo

Yilma Valeria Espitia Sanabria

Marleny del Pilar Ochoa Tapias

Docente:

Víctor Hugo Mercado

Curso:

Base de Datos II

Grupo: 76

Ingeniería de Software y Datos

Facultad de Ingeniería y de Ciencias Agropecuarias

Institución Universitaria Digital de Antioquia

2024

**Introducción**

Las empresas generan gran cantidad de datos cada día, siendo estos un activo muy valioso. Para su análisis es fundamental utilizar la inteligencia de negocio y la organización para dar facilidad al momento de tomar decisiones. El modelo estrella es una estructura dentro del Data Warehouse, diseñada para optimizar y facilitar el análisis de datos en entornos empresariales. Un Data Mart convierte la información sin procesar en contenido estructurado y significativo para un departamento empresarial específico (AWS). Los datos se organizan en una tabla central de hechos que contiene las métricas del objeto a analizar y tablas de dimensiones que asemejan los brazos de una estrella, representando cada uno de los ejes en un espacio multidimensional. Una de las grandes ventajas de un modelo estrella Data Mart es la eficiencia en el análisis, flexibilidad, mejora del rendimiento empresarial, optimización del proceso de toma de decisiones y compatibilidad con herramientas de análisis. Esta clase de modelo facilita la exploración de datos, la generación de informes e identificación de tendencias y patrones clave que pueden ayudar a mejorar el rendimiento empresarial, debido a que permite optimizar el proceso de uso de la información, a través de la agrupación de factores de interés que inciden en un hecho en particular. (Zerpa, Garcia, & Izquierdo, 2020). En el proceso de búsqueda de información en la base de datos proporcionada, es necesario identificar los aspectos de mayor relevancia. Es por esto que, en el siguiente trabajo nos enfocaremos en la construcción de un modelo estrella Data Mart utilizando la base de datos de jardinería y se llevará a cabo el análisis de las categorías para identificar el producto más vendido, el año de mayores ventas y la categoría con más productos.

Implementar una base de datos staging permite añadir columnas a las tablas de datos que van desde un tipo date, clave externa a otra tabla que almacena metadatos más detallados; estos metadatos pueden no solo informar de cuándo están incorporados los datos en datawarehouse sino también de modificaciones posteriores.

La gestión de datos desempeña un papel importante en la optimización de procesos y estrategias empresariales. Este trabajo está relacionado estrechamente con la inteligencia de negocios, partiendo del modelo estrella realizado anteriormente en una base de datos staging, tomando las dimensiones más importantes e implementándolo en Visual Community que nos proporciona una plataforma dinámica para la visualización y análisis de datos críticos, observaremos cómo esta implementación facilita la extracción de información ofreciendo una mejor toma de decisiones y eficiencia operativa.

**Objetivo**

Construir consultas SQL para la creación de una base de datos staging basada en el modelo relacional de la base de datos “Jardinería” aprovechando la capacidad de Visual Community para el análisis de datos. Teniendo en cuenta los requerimientos de la evidencia dos y las correcciones previas de la actividad anterior.

**Planteamiento del problema**

Si bien en la base de datos de Jardinería, se encuentra la relación entre varias tablas relacionadas al cliente, empleado, producto y geografía, estas tablas no proporcionan la información adecuada para lograr responder las preguntas del negocio. En primer lugar, la falta de conexión entre estas tablas y la dispersión de los datos de la base de datos de Jardinería hace que sea una tarea compleja el lograr identificar con la granularidad necesaria para saber cuál es el producto más vendido y cuál es la categoría con más productos. En segundo lugar, la falta de un modelo estrella dificulta la correcta interpretación y análisis de la información disponible no solo de las preguntas del negocio, sino también la identificación de posibles problemas, relaciones, patrones e incluso oportunidades que llegarán a presentar entre los datos.

Por otro lado, la falta de un modelo Staging hace que se carezca de un entorno dedicado para la consolidación, limpieza y transformación de los datos antes de su integración en el modelo estrella. La falta de este paso puede llevar a inconsistencias en los datos, duplicando registros y dificultando el proceso de ETL. Por lo tanto, la ausencia de un modelo Staging representa un obstáculo significativo para la orientación de información precisa y significativa a partir de la base de datos jardinería.

**Análisis del Problema**

De acuerdo con el análisis minuciosamente realizado dentro de nuestro grupo, pudimos observar que, gracias al modelo de estrella, hay una forma de identificar las tres categorías solicitadas en la evidencia solicitada. Ya que el problema radica en la falta de conexión entre las tablas y la dispersión de datos en la base de datos de Jardinería, al implementar un modelo estrella con facturaVenta y dimensiones resolvería estas deficiencias. Para abordar la dispersión de datos en la base de datos de Jardinería y facilitar la implementación del modelo estrella, proponemos la creación de una base de datos Staging la cual servirá como un entorno intermedio donde se puedan consolidar, limpiar y transformar los datos de origen. Mediante el uso de base de datos de Staging podremos realizar procesos de ETL de manera eficiente y así garantizando una mejor estructura de la información.

**Propuestas de la solución con:**

**Correcciones a la entrega 1**

Se hace la corrección de la gráfica correspondiente al modelo estrella con su respectivo script.

**Modelo estrella, base de datos de Jardinería**

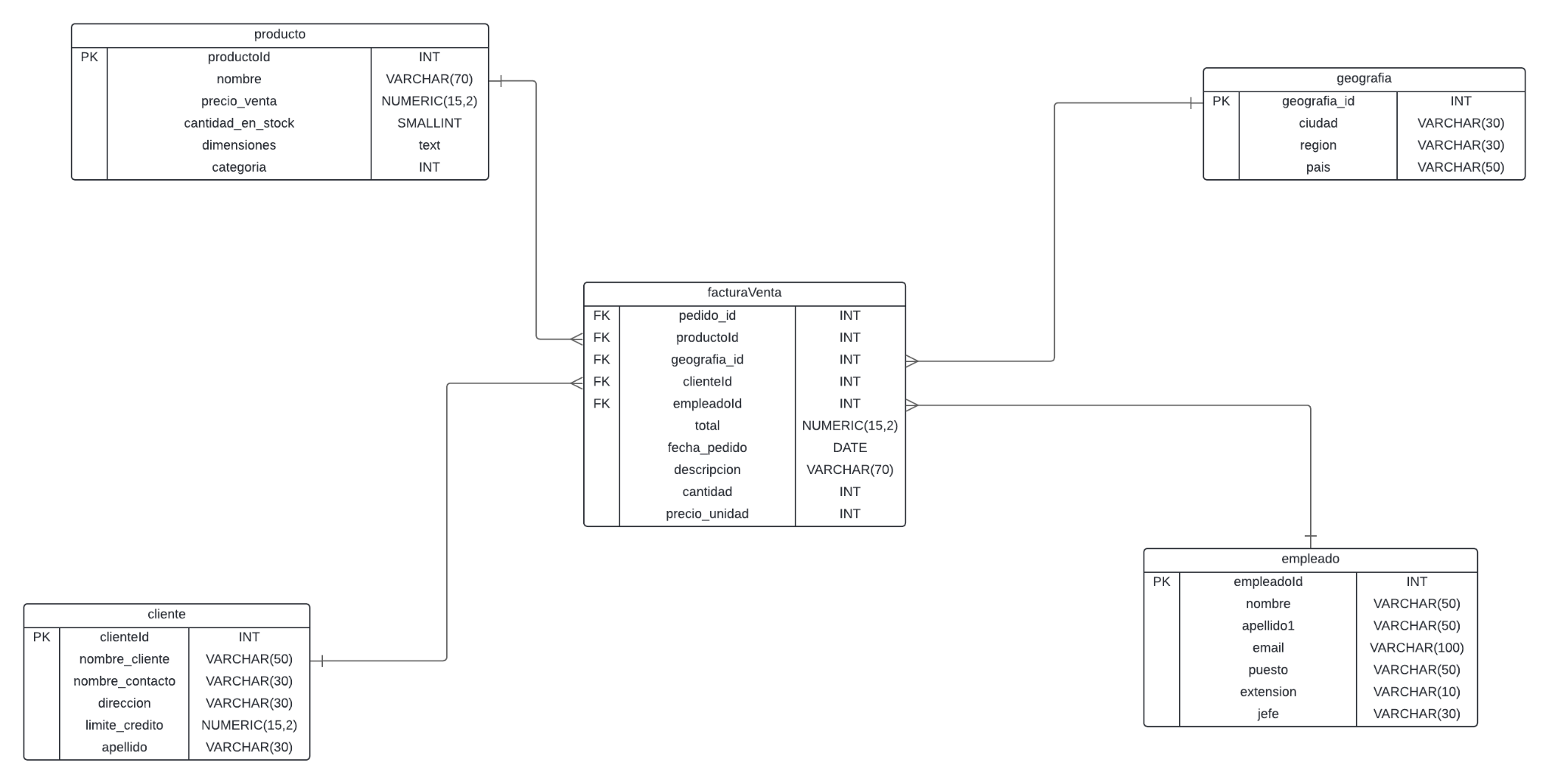


Imagen 1. Modelo estrella de Jardinería

**Script**

Este script se utilizó como plantilla para la creación de tablas del modelo estrella

CREATE TABLE [empleado] (

[empleadoId] INT,

[nombre] VARCHAR(50),

[apellido1] VARCHAR(50),

[email] VARCHAR(50),

[puesto] VARCHAR(50),

[extension] VARCHAR(10),

[jefe] VARCHAR(30),

PRIMARY KEY ([empleadoId])

);

CREATE TABLE [cliente] (

[clienteId] INT,

[nombreCliente] VARCHAR(30),

[contacto] VARCHAR(15),

[genero] VARCHAR(15),

[edad] INT,

[cedula] VARCHAR(30),

[direccion] VARCHAR(30),

[limiteCredito] NUMERIC(15,2),

[comportamiento] VARCHAR(30),

[apellido] VARCHAR(30),

[email] VARCHAR(50),

PRIMARY KEY ([clienteId])

);

CREATE TABLE [geografia] (

[geografiaId] INT,

[ciudad] VARCHAR(30),

[region] VARCHAR(30),

[pais] VARCHAR(30),

PRIMARY KEY ([geografiaId])

);

CREATE TABLE [producto] (

[codigoProducto] INT,

[nombre] VARCHAR(70),

[precio\_venta] NUMERIC(15,2),

[cantidad\_en\_stock] SMALLINT,

[dimensiones] text,

[categoria] INT,

PRIMARY KEY ([codigoProducto])

);

CREATE TABLE [facturaVenta] (

[pedidoId] INT,

[productoId] INT,

[oficinaId] INT,

[clienteId] INT,

[empleadoId] INT,

[total] NUMERIC(15,2),

[fecha\_pedido] DATE,

[descripcion] VARCHAR(70),

[cantidad] INT,

[precio\_unidad] INT,

CONSTRAINT [FK\_facturaVenta.oficinaId]

FOREIGN KEY ([oficinaId])

REFERENCES [geografia]([geografiaId]),

CONSTRAINT [FK\_facturaVenta.clienteId]

FOREIGN KEY ([clienteId])

REFERENCES [cliente]([nombreCliente]),

CONSTRAINT [FK\_facturaVenta.productoId]

FOREIGN KEY ([productoId])

REFERENCES [producto]([nombre])

);

**Descripción del análisis realizado a los datos *Jardinería* y cómo estos se trasladaron a la base de datos *Staging.***

* Revisión y corrección del modelo estrella de acuerdo a las observaciones dadas por el profesor, garantizando la precisión y coherencia con los requerimientos establecidos. Dicho proceso también fue replicado en el script.
* Se hizo verificación de los campos de las dimensiones del modelo estrella, para asegurarse de que estén correctamente representados en la base de datos de Jardinería, garantizando la integridad de los datos.
* Se hace identificación de las posibles consultas de SQL, para obtener los datos necesarios que se requiere en cada dimensión, todo lo anterior mediante la plataforma de SQL Management Studio.
* Creación de una base de datos de destino llamada “staging076”.
* Después de identificar las consultas que se tendrán en cuenta en el proceso, se procede a crear un nuevo proyecto en el visual Community, mediante el uso de una plantilla en blanco llamada Integration Services Project.
* Se configura la conexión a la base de datos de origen que se llama jardinería y la conexión a la base de datos de destino “staging076”.

**Descripción paso a paso**

Se crea un nuevo flujo de Datos llamado “Extracción Jardinería 076”:

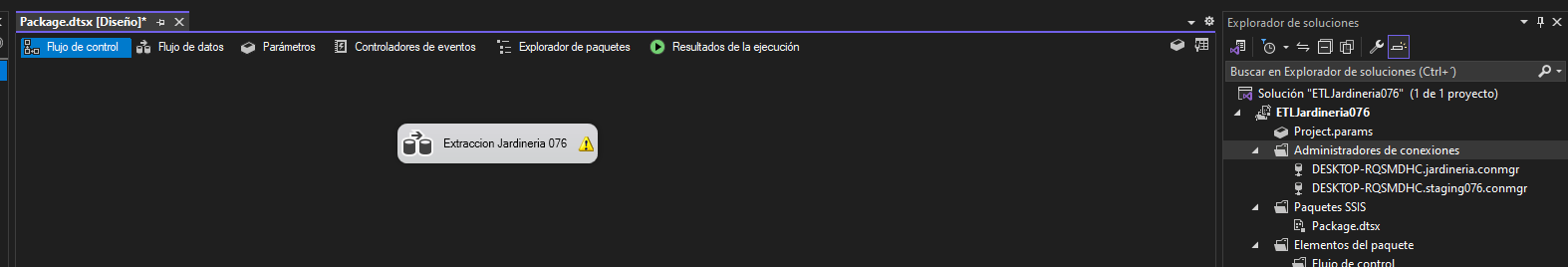


Imagen 2. Creacion de Data Flow en Visual Community 2022

Se realiza la extracción de los datos proveniente de la DB de jardinería, teniendo en cuenta el modelo estrella que realizamos en la primera evidencia:

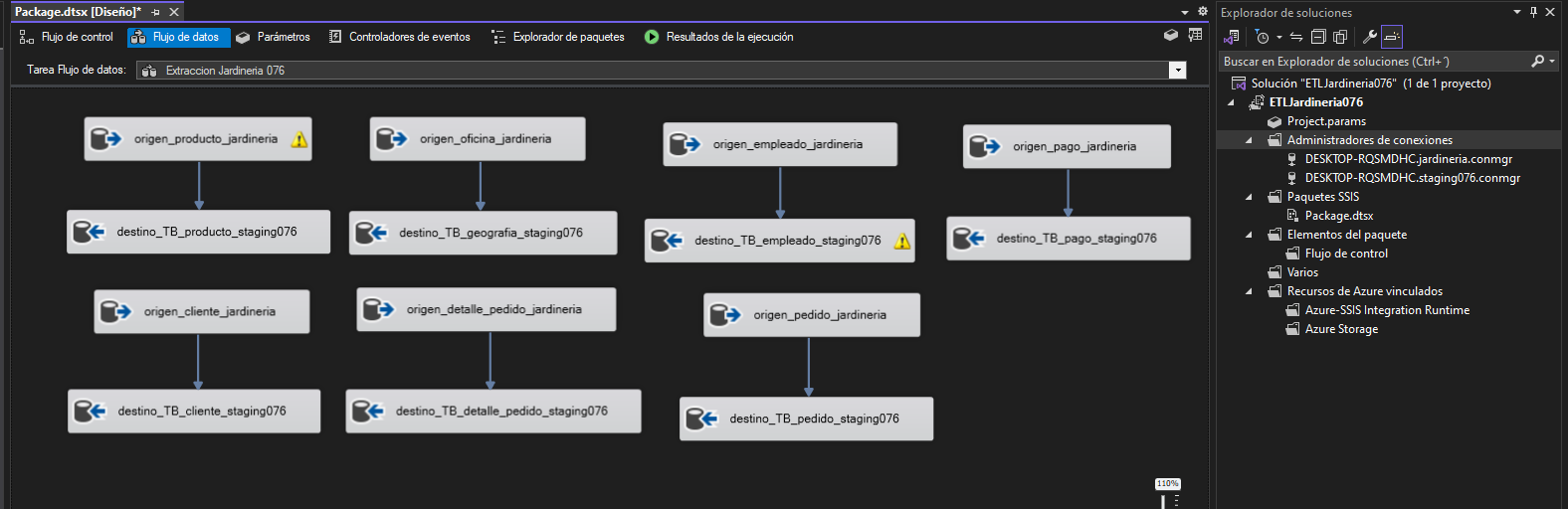


Imagen 3. Creaciones de tabla de origen y destino en Visual Community 2022

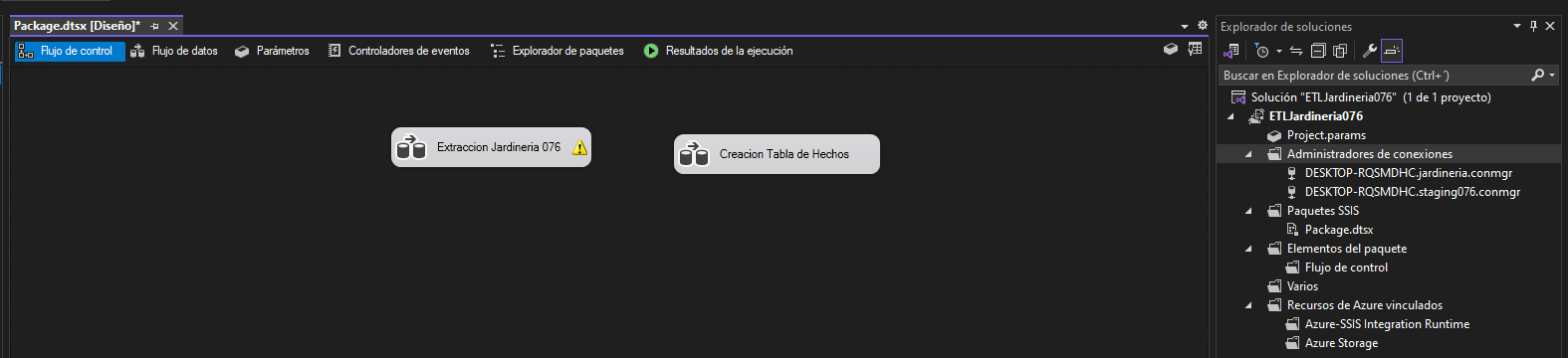
Se crea un nuevo flujo de Datos para la creación de la tabla de hechos llamada “Creación de Tabla de Hechos”:

Imagen 4. Creación de un nuevo Data Flow para la tabla de hechos

Dentro de este flujo de Datos, se extrae de las tablas de la base de datos de staging076 la información necesaria para la creación de la tabla de hechos:

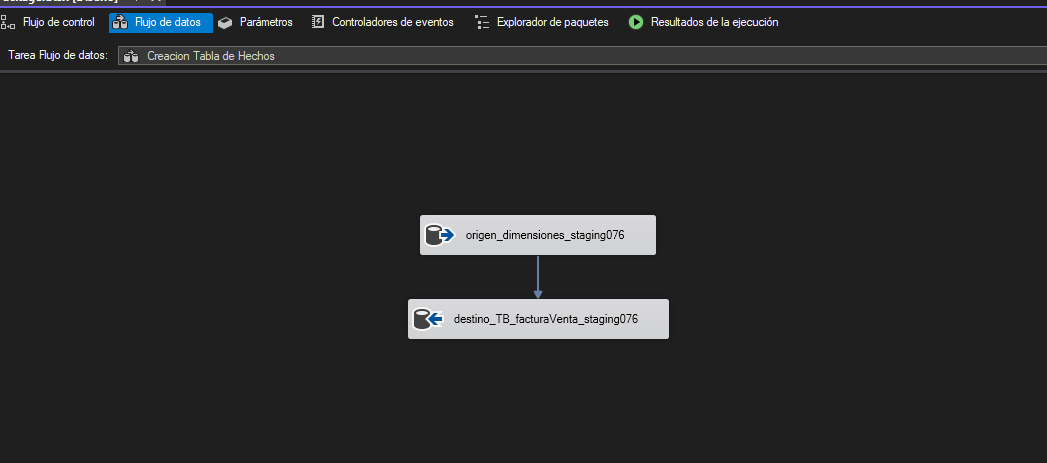


Imagen 5. Creación de tabla de destino y origen para tabla de hecho

Finalmente, luego de corroborar la validación de los datos para cada una de las tablas de la base de datos de staging076, se crea un backup de la base de datos origen, “jardinería” y de la base de datos de staging o de destino que se llama “staging076”:

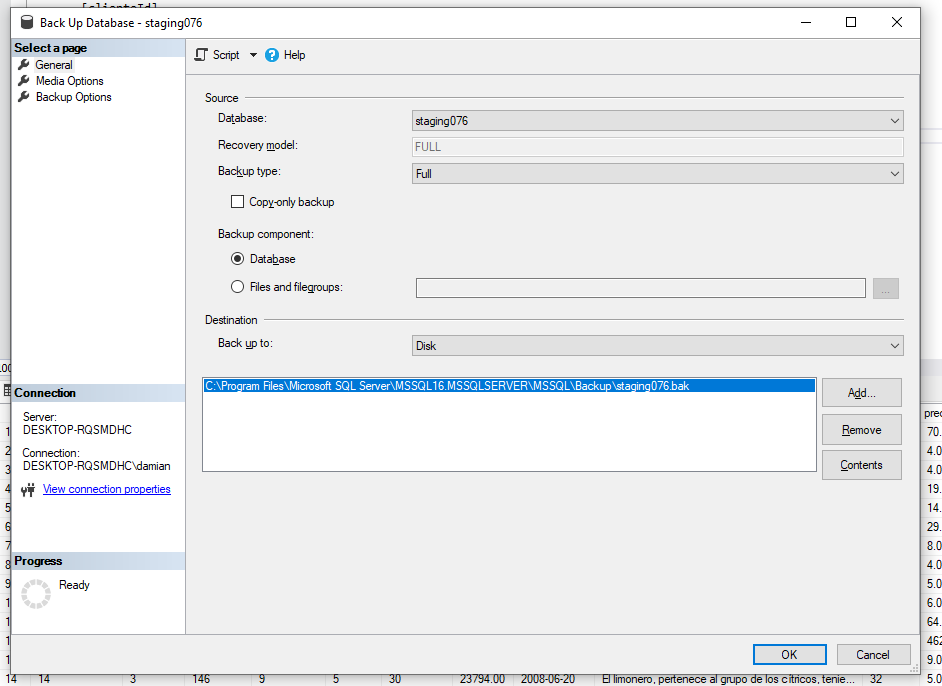


Imagen 6. Creación de Backup para base de datos “staging076”

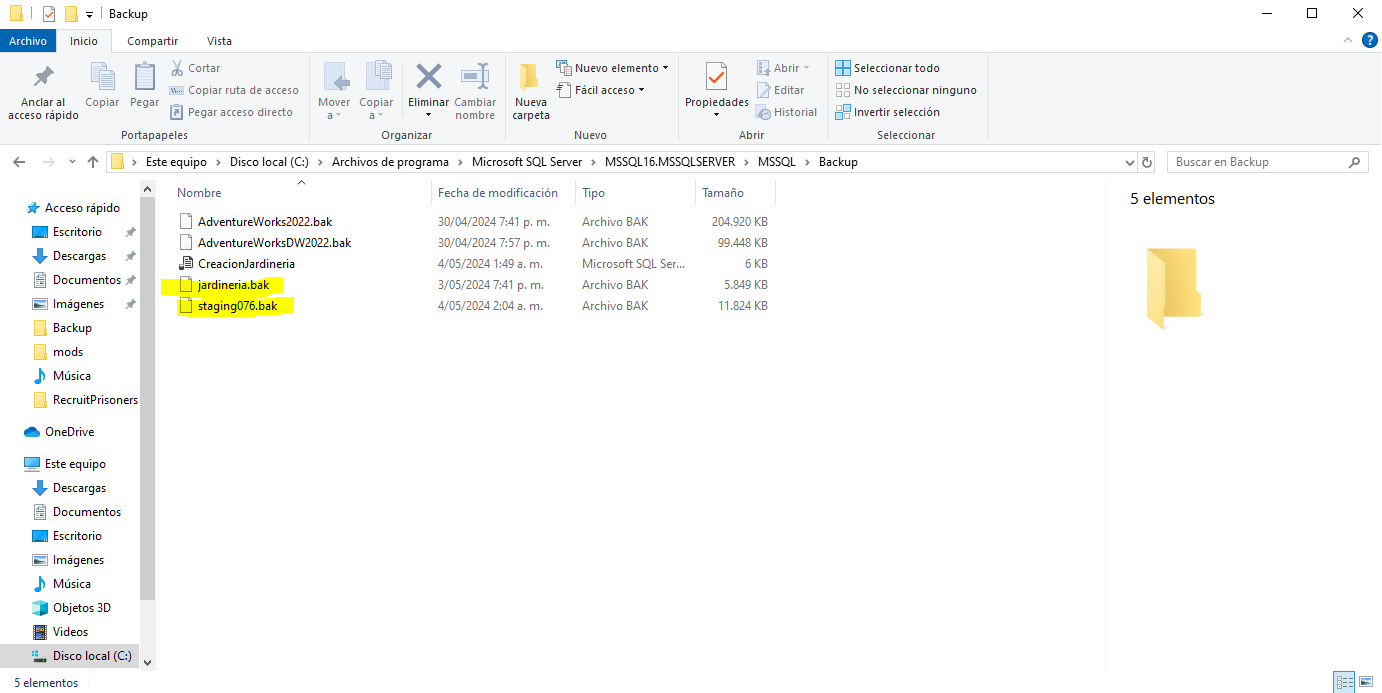


Imagen 7. Verificación de Backups creados

**Scripts**

Los siguientes queries se utilizaron para realizar las consultas necesarias para extraer los datos para ser usados en el modelo estrella. Hay que tener en cuenta que aquí no se encuentra la inserción de datos hacia las tablas ya que dicho proceso se llevo a cabo por medio del Visual Community 2022.

-- Consultas en la BD de jardineria para la creacion de tablas de destino

USE jardineria

-- Seleccion Tabla de Origen Producto

SELECT P.precio\_venta, P.nombre, P.cantidad\_en\_stock, P.dimensiones, P.Categoria, P.descripcion

FROM dbo.producto P

ORDER BY P.nombre ASC

-- Seleccion Tabla de Origen Oficina

Select O.ciudad, O.region, O.pais, O.ID\_oficina

FROM dbo.oficina O

ORDER BY O.ciudad ASC

-- Seleccion Tabla de Origen Cliente

Select C.nombre\_cliente,C.apellido\_contacto as apellido ,C.nombre\_contacto, C.linea\_direccion1 as direccion, C.limite\_credito, C.ID\_empleado\_rep\_ventas

FROM dbo.cliente C

ORDER BY C.nombre\_cliente ASC

-- Seleccion Tabla de Origen Empleado

Select E.nombre, E.apellido1,E.email , E.puesto, E.extension, E.ID\_jefe, E.ID\_oficina

FROM dbo.empleado E

ORDER BY E.nombre ASC

-- Seleccion Tabla de Origen Detalle Pedido

Select PE.ID\_pedido, PE.cantidad, PE.precio\_unidad, PE.ID\_producto

FROM dbo.detalle\_pedido PE

-- Seleccion Tabla de Origen Pedido

Select P.ID\_pedido, P.ID\_cliente, P.fecha\_pedido

FROM dbo.pedido P

-- Seleccion Tabla de Origen Pago

Select P.ID\_cliente, P.total

FROM dbo.pago P

-- Creacion de la BD de destino y creacion de sus respectivas tablas

DROP DATABASE IF EXISTS staging076;

CREATE DATABASE staging076;

USE staging076;

-- Creacion Tabla de pago para DB staging076

CREATE TABLE "destino\_TB\_pago\_staging076" (

"pago\_id" int identity(1,1),

"ID\_cliente" int,

"total" numeric(15,2)

primary key("pago\_id")

)

-- Creacion Tabla de empleado para DB staging076

CREATE TABLE "destino\_TB\_empleado\_staging076" (

"empleadoId" int identity(1,1),

"nombre" nvarchar(50),

"apellido1" nvarchar(50),

"email" nvarchar(100),

"puesto" nvarchar(50),

"extension" nvarchar(10),

"ID\_jefe" int,

"ID\_oficina" int

primary key("empleadoId")

)

-- Creacion Tabla de detalle\_pedido para DB staging076

CREATE TABLE "destino\_TB\_detalle\_pedido\_staging076" (

"detalle\_pedido\_id" int identity(1,1),

"ID\_pedido" int,

"cantidad" int,

"precio\_unidad" numeric(15,2),

"ID\_producto" int

primary key( "detalle\_pedido\_id")

)

-- Creacion Tabla de pedido para DB staging076

CREATE TABLE "destino\_TB\_pedido\_staging076" (

"pedido\_id" int identity(1,1),

"ID\_pedido" int,

"ID\_cliente" int,

"fecha\_pedido" date

primary key("pedido\_id")

)

-- Creacion Tabla de geografia para DB staging076

CREATE TABLE "destino\_TB\_geografia\_staging076" (

"geografia\_id" int identity(1,1),

"ciudad" nvarchar(30),

"region" nvarchar(50),

"pais" nvarchar(50),

"ID\_oficina" int

primary key("geografia\_id")

)

-- Creacion Tabla de producto para DB staging076

CREATE TABLE "destino\_TB\_producto\_staging076" (

"productoId" int identity(1,1),

"precio\_venta" numeric(15,2),

"nombre" nvarchar(70),

"cantidad\_en\_stock" smallint,

"dimensiones" nvarchar(25),

"Categoria" int,

"descripcion" text

primary key("productoId")

)

-- Creacion Tabla de destino Cliente para DB staging076

CREATE TABLE "destino\_TB\_cliente\_staging076" (

"clienteId" int identity(1,1),

"nombre\_cliente" nvarchar(50),

"apellido" nvarchar(30),

"nombre\_contacto" nvarchar(30),

"direccion" nvarchar(50),

"limite\_credito" numeric(15,2),

"ID\_empleado\_rep\_ventas" int

primary key("clienteId")

)

-- Creacion Tabla de hecho para staging076 de Factura de Ventas

CREATE TABLE "destino\_TB\_facturaVenta\_staging076" (

"facturaVenta\_id" int identity(1,1),

"pedido\_id" int,

"productoId" int,

"geografia\_id" int,

"clienteId" int,

"empleadoId" int,

"total" numeric(15,2),

"fecha\_pedido" date,

"descripcion" nvarchar(max),

"cantidad" int,

"precio\_unidad" numeric(15,2)

primary key("facturaVenta\_id")

)

-- Seleccion de datos de diferentes tablas de staging076 para la posterior creacion

-– de la tabla de hechos de factura de venta:

SELECT PE.pedido\_id,P.productoId, G.geografia\_id,C.clienteId,E.empleadoId, PA.total,PE.fecha\_pedido, PR.descripcion, DP.cantidad, DP.precio\_unidad

FROM destino\_TB\_cliente\_staging076 C

INNER JOIN destino\_TB\_pedido\_staging076 PE ON PE.ID\_cliente = C.clienteId

INNER JOIN destino\_TB\_detalle\_pedido\_staging076 DP ON DP.ID\_pedido = PE.ID\_pedido

INNER JOIN destino\_TB\_producto\_staging076 P ON P.productoId = DP.ID\_producto

INNER JOIN destino\_TB\_empleado\_staging076 E ON E.empleadoId = C.ID\_empleado\_rep\_ventas

INNER JOIN destino\_TB\_pago\_staging076 PA ON PA.ID\_cliente = C.clienteId

INNER JOIN destino\_TB\_producto\_staging076 PR ON PR.productoId = DP.ID\_producto

INNER JOIN destino\_TB\_geografia\_staging076 G ON G.ID\_oficina = E.ID\_oficina

**Referencias**

Base de datos Jardinería Iniciar sesión en Canvas. (s. f.). https://iudigital.instructure.com/courses/15580/files/4737073/download

Calzada, J. M. (2020, noviembre 5). *Seis razones por las que usar staging -*. Consultoria Certia. Formación | Consultoria | Desarrollo; Consultoria Certia. <https://www.certia.net/seis-razones-por-las-que-usar-staging/>

AWS. (s.f.). *¿ Que es un data mart?- Explicación de los data marts- AWS.* Obtenido de <https://aws.amazon.com/es/what-is/data-mart/>

Zerpa, H., Garcia, R., & Izquierdo, H. (2020). Datamart Basado en el modelo estrella para la implementación de indicadores clave de desempeño como salida del big data.