BASES DE DATOS II

PREICA2501B010095

S30 -EVIDENCIA DE APRENDIZAJE 3

PROCESO DE TRANSFORMACIÓN DE DATOS Y CARGA EN EL DATA MART FINAL

REALIZADO POR:

Grupo:

BASEDEDATOSIIPRE 5

JEREMY IVAN PEDRAZA HERNANDEZ

PRESENTADO A:

INSTRUCTOR

VICTOR HUGO MERCADO RAMOS

INSTITUCIÓN UNIVERSITARIA DIGITAL DE ANTIOQUIA

2025

Contenido

[Planteamiento del Problema 3](#_Toc191411633)

[Introducción 4](#_Toc191411634)

[Objetivos 4](#_Toc191411635)

[Análisis de la base de datos (Problema) 5](#_Toc191411636)

[Identificación de campos necesarios 6](#_Toc191411637)

[Diseño del modelo estrella 6](#_Toc191411638)

[Representación Gráfica del modelo estrella 7](#_Toc191411639)

[Código MYSQL para creación de modelo dimensional **¡Error! Marcador no definido.**](#_Toc191411640)

[Conclusiones 27](#_Toc191411641)

[Bibliografía 32](#_Toc191411642)

# Planteamiento del Problema

Desarrollo de proceso de transformación de datos y carga en el *Data Mart* final.

1. Preparación:
   1. Revisar el modelo Estrella definido en el proyecto de análisis de la base de datos Jardinería para comprender la estructura y las relaciones entre las tablas de dimensiones y la tabla de hechos.
   2. Verificar la disponibilidad y consistencia de la base de datos de *staging* previamente creada para el proyecto.
2. Extracción de datos desde la base de datos origen hacia la base de datos de *Staging*:
   1. Utilizar consultas SQL para extraer datos relevantes de la base de datos origen y cargarlos en las tablas correspondientes de la base de datos de *staging.*
   2. Verificar la integridad y consistencia de los datos extraídos para asegurar que cumplan con los requisitos del modelo Estrella.
3. Transformación de datos según las necesidades analíticas:
   1. Aplicar técnicas de transformación de datos, como limpieza, normalización y enriquecimiento, para preparar los datos de acuerdo con las necesidades analíticas específicas.
   2. Realizar la transformación de los datos utilizando consultas SQL u herramientas de ETL (*Extract, Transform, Load*) según sea necesario para garantizar la calidad y coherencia de los datos.
4. Carga de registros en el *Data Mart* final:
   1. Diseñar consultas SQL o scripts de carga para insertar los registros transformados desde la base de datos de *staging* en las tablas del *data mart*final.
   2. Ejecutar las consultas de carga y verificar que los datos se hayan insertado correctamente en el *data mart* final.
5. Documentación y presentación:
   1. Documentar todo el proceso de transformación de datos y carga en un informe detallado que incluya una descripción de las etapas realizadas, las consultas SQL utilizadas.
   2. Presentar el informe de manera clara y concisa, asegurándose de incluir referencias al modelo Estrella y las técnicas de ETL aplicadas.

# Introducción

El modelo dimensional es una técnica clave en el diseño de almacenes de datos (data warehouses), permitiendo una consulta y análisis eficiente de grandes volúmenes de información. Desarrollado por Ralph Kimball (Kimball, 2025), este enfoque organiza los datos en tablas de hechos, que almacenan métricas cuantitativas, y tablas de dimensiones, que ofrecen contexto descriptivo. Su estructura simplificada y centrada en el usuario lo convierte en una herramienta esencial para transformar datos operacionales en información valiosa que respalda la toma de decisiones estratégicas.

Una de sus principales ventajas es la capacidad de procesar consultas complejas de manera ágil y eficiente, optimizando las operaciones de agregación y filtrado (Ahmed, 2024). Además, su diseño intuitivo facilita su uso tanto para profesionales técnicos como para usuarios de negocio, lo que simplifica la generación de reportes y análisis personalizados. En un entorno empresarial donde la velocidad y precisión son fundamentales, el modelo dimensional se posiciona como un recurso imprescindible para el análisis de datos y la toma de decisiones informadas.

# Objetivos

1. Identificar cómo construir un modelo dimensional a partir de un esquema transaccional.
2. Construir un modelo dimensional.
3. Identificar factores de mejora.
4. Codificar un esquema dimensional.

# Análisis de la base de datos (Problema)

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Ilustración 1 Modelo Transaccional Jardinería

1. Se ha cargado la base de datos en la instancia MYSQL y se insertan los datos, así mismo el código cuenta con relaciones en llaves foranes preestablecidas. El producto de esto se muestra en la presente imagen (Ilustración 1)

## Identificación de campos necesarios

1. Observamos que la tabla ***“detalle\_pedido”*** es la tabla en la que se concentra el valor transaccional a nivel observable en el esquema de la base de datos de jardinería. En ella encontramos, el detalle del pedido, el identificador del pedido, el identificador del producto y la cantidad.
2. Así mismo observamos que las tablas cliente, producto, empleado, oficina y pago son tablas importantes en este esquema por tanto procederemos a tomarlas en cuenta a la hora de construir el diagrama dimensional.

## Diseño del modelo estrella

1. Para comenzar a construir la base de datos Data Mart, necesitaremos primero preparar la base de datos de recepción de datos originarios desde el staging, por tanto, usaremos el siguiente código SQL para crear la base de datos

|  |
| --- |
| DROP DATABASE IF EXISTS DBO\_DATAMART\_Jardineria;  CREATE DATABASE DBO\_DATAMART\_Jardineria;  GO  USE DBO\_DATAMART\_Jardineria; |

1. Teniendo lista la base de datos podremos desde el SISS realizar la transformación y traslado de la información desde la Base de datos Staggin el cual ya insertamos en la evidencia pasada.
2. Retomando a su vez el desarrollo del modelo estrella que hicimos en la actividad 1, retomamos que las dimensiones serán las siguientes, producto, empleado, oficina y pago que tomarán los nombres de:
   1. **dim\_tiempo** (Basada en fechas de pedido y pago)
   2. **dim\_cliente** (Basada en la tabla cliente)
   3. **dim\_producto** (Basada en la tabla producto)
   4. **dim\_empleado** (Basada en la tabla empleado)
   5. **dim\_oficina** (Ubicación de empleados)
   6. **dim\_pago** (Basada en las transacciones de pago)

**EXTRAS:**

* 1. **dim\_pedido** (Basada en el estado de pedido y método de envío)

1. Las relaciones entre la tabla de hechos y las dimensiones serán de la siguiente manera:
   1. FK > ID\_CLIENTE **dim\_cliente** -> ID\_CLIENTE
   2. FK > ID\_PRODUCTO **dim\_producto** -> ID\_PRODUCTO
   3. FK > ID\_EMPLEADO **dim\_empleado** -> ID\_EMPLEADO
   4. FK > ID\_OFICINA **dim\_oficina** -> ID\_OFICINA
   5. FK > ID\_TIEMPO **dim\_tiempo** -> ID\_TIEMPO
   6. FK > ID\_PAGO **dim\_pago** -> ID\_PAGO

EXTRAS\_

* 1. FK > ID\_PEDIDO **dim\_pedido** -> ID\_PEDIDO

## Representación Gráfica del modelo estrella

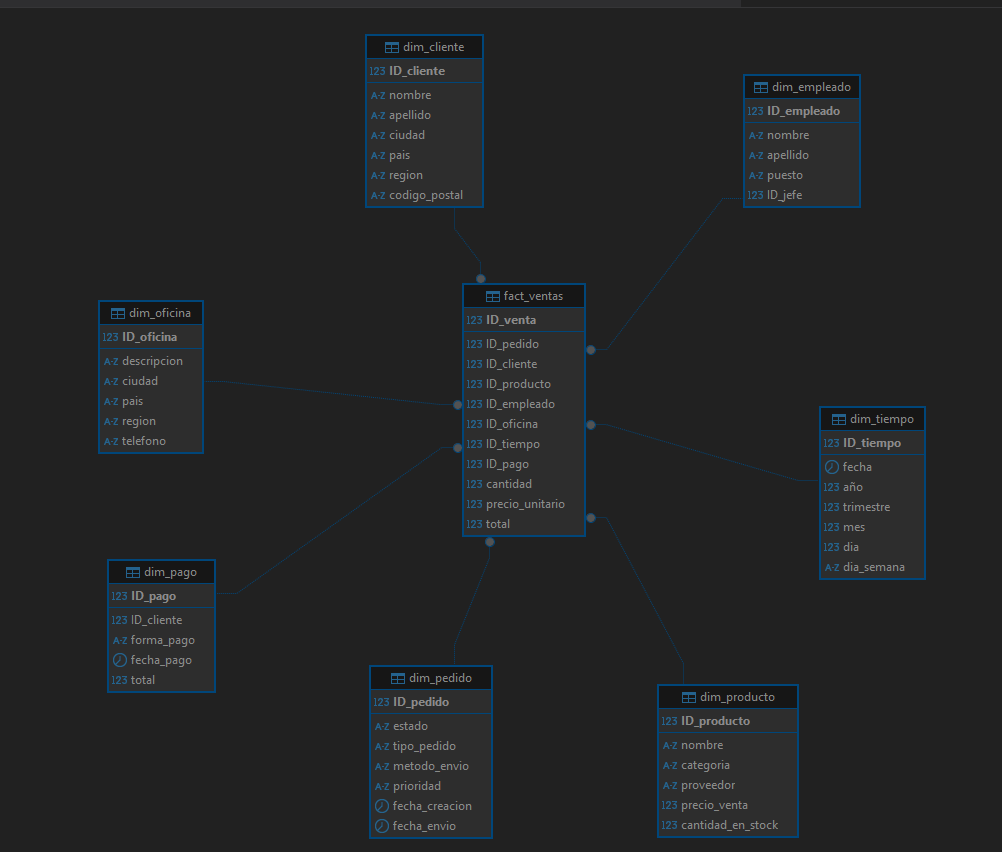
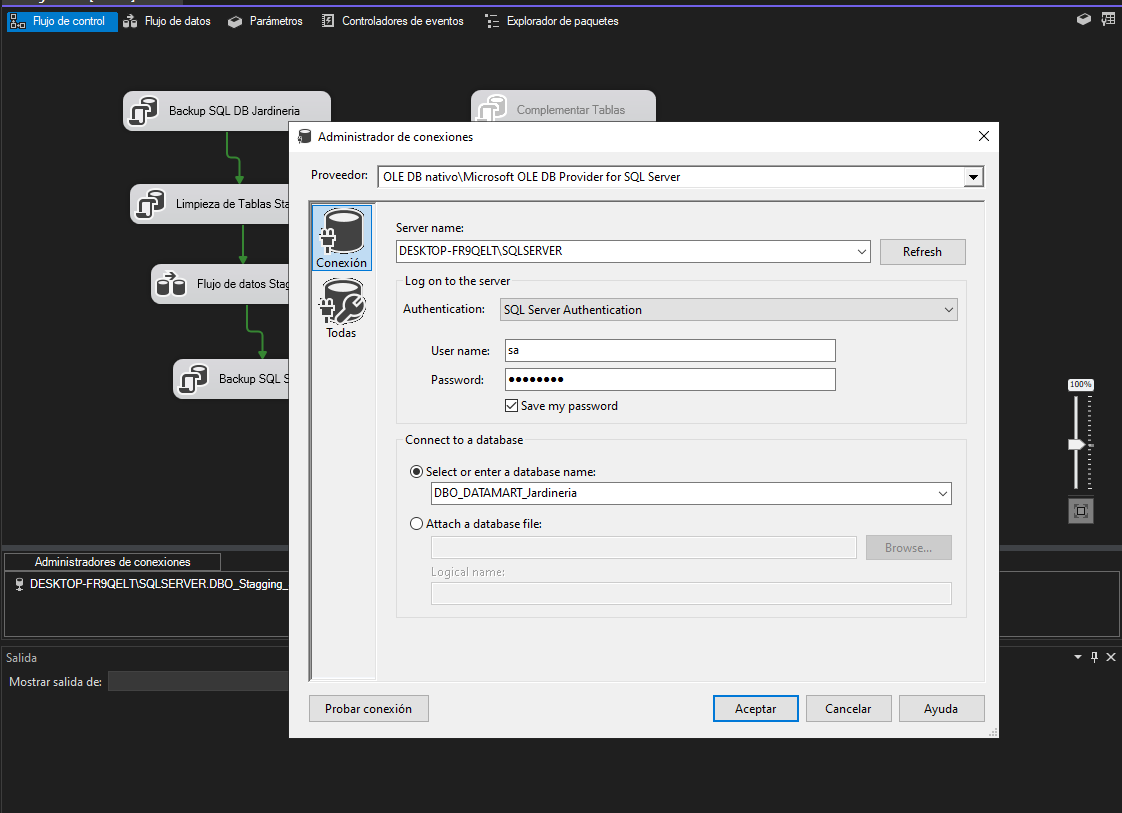


Ilustración 2 Modelo Dimensional

## Construcción de tablas desde herramienta SSIS



Teniendo la base de datos ya creada por parte del área de base de datos podremos comenzar a programar las tareas de traslado, pero primero necesitaremos construir las tablas que recibirán la información.

## Crear tabla de dimensiones

### Tabla dimensión cliente

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13 | **CREATE** **TABLE** dim\_cliente (  ID\_cliente INT **PRIMARY** **KEY**,  nombre\_cliente VARCHAR(50),  nombre\_completo\_contacto VARCHAR(61),  telefono VARCHAR(15),  fax VARCHAR(15),  direccion VARCHAR(100),  ciudad VARCHAR(50),  region VARCHAR(50),  pais VARCHAR(50),  codigo\_postal VARCHAR(10),  cliente\_vip BIT  ); | |

#### Transformaciónes a realizar

* nombre\_completo\_contacto = nombre\_contacto + ' ' + apellido\_contacto
* direccion = concat(linea\_direccion1, ' ', linea\_direccion2)
* cliente\_vip = 1 si limite\_credito > 100000

## Tabla Dimensión producto

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10 | **CREATE** **TABLE** dim\_producto (  ID\_producto INT **PRIMARY** **KEY**,  CodigoProducto VARCHAR(15),  nombre VARCHAR(70),  Categoria VARCHAR(50),  dimensiones VARCHAR(25),  proveedor VARCHAR(50),  cantidad\_en\_stock SMALLINT,  precio\_venta NUMERIC(15,2)  ); | |

#### Transformaciones a realizar

* La categoría viene de la tabla Categoria\_producto.

## Tabla dimensión tiempo

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8 | **CREATE** **TABLE** dim\_tiempo (  ID\_fecha DATE **PRIMARY** **KEY**,  anio INT,  mes INT,  dia INT,  nombre\_mes VARCHAR(20),  trimestre INT  ); | |

## Tabla dimensión empleado

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7 | **CREATE** **TABLE** dim\_empleado (  ID\_empleado INT **PRIMARY** **KEY**,  nombre\_completo VARCHAR(101),  email VARCHAR(100),  puesto VARCHAR(50),  ID\_oficina INT  ); | |

#### Transformaciones a realizar

* Concatenar el nombre\_completo = nombre + ' ' + apellido1 + ' ' + apellido2

## Tabla dimensión oficina

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10 | **CREATE** **TABLE** dim\_oficina (  ID\_oficina INT **PRIMARY** **KEY**,  Descripcion VARCHAR(10),  ciudad VARCHAR(30),  pais VARCHAR(50),  region VARCHAR(50),  codigo\_postal VARCHAR(10),  telefono VARCHAR(20),  direccion VARCHAR(100)  ); | |

#### Transformaciones a realizar

* Conversión de dirección = concat(linea\_direccion1, ' ', linea\_direccion2)

## Tabla de hechos ventas

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11 | **CREATE** **TABLE** fact\_ventas (  ID\_venta INT **IDENTITY**(1,1) **PRIMARY** **KEY**,  ID\_pedido INT,  ID\_cliente INT,  ID\_producto INT,  ID\_empleado INT,  ID\_fecha DATE,  cantidad INT,  precio\_unitario NUMERIC(15,2),  total\_venta NUMERIC(15,2)  ); | |

# Ejecutamos la tarea y validamos los resultados

Imagen de la pantalla de un computador

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Con esto ya tenemos todo preparado para realizar la transformación y envío de datos al Data Mart

# Flujo de Data Flow Task

# Cargue de Dimensión oficina

Diagrama

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Para la tabla de oficina vamos a hacer una selección de los campos previo a la transformación de los datos, usaremos la siguiente consulta

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9 | **SELECT**  ID\_oficina,  ciudad,  pais,  region,  telefono,  linea\_direccion1 ,  linea\_direccion2  **FROM** jardineria.dbo.oficina | |

Vamos a concatenar las líneas de direcciones en una sola la conservaremos y remplazaremos la línea dirección 1 y esa se asignara a la dimensión como columna **dimensión\_completa**

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

A partir de los datos resultantes obtendremos una sola dirección, valores por defectos en caso de null

Una captura de pantalla de una computadora

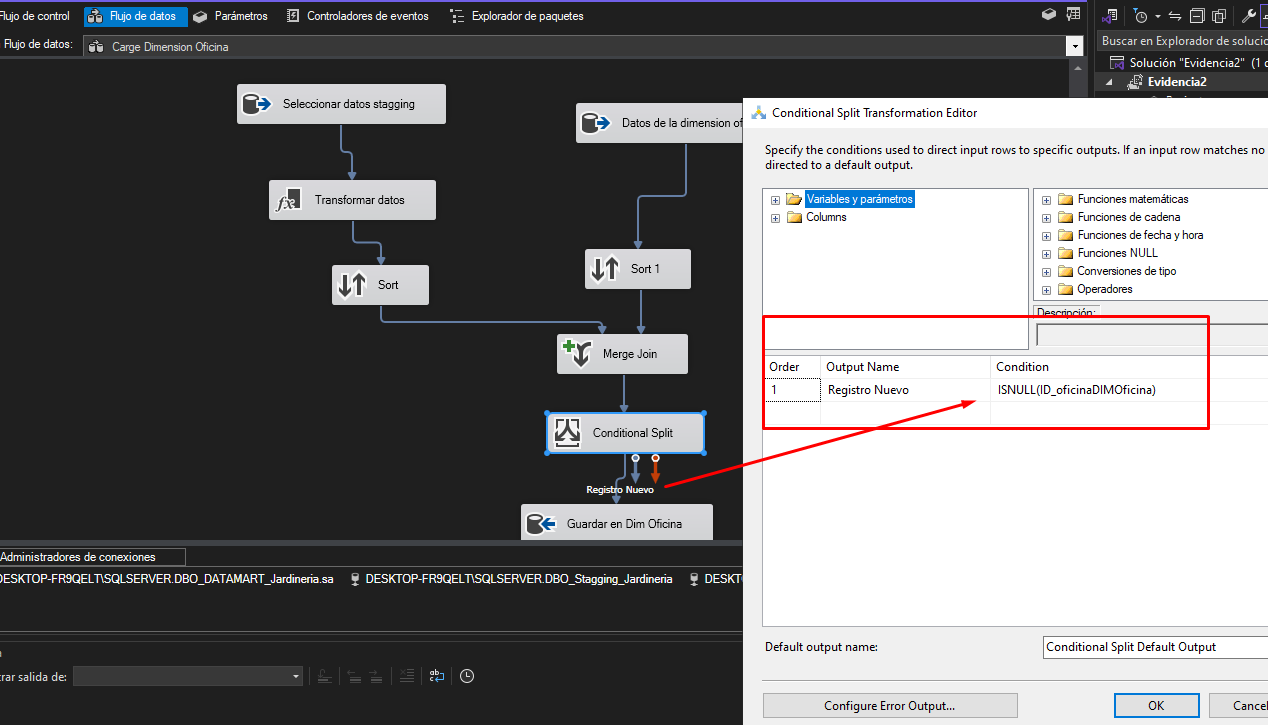
El contenido generado por IA puede ser incorrecto.

Agregamos un sort para ordenar los datos por país

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Adicionalmente debemos verificar si el registro es nuevo, Esto lo haremos por medio del **merge Join** y el condicional Split, uniendo los registros e insertando solo si este último es nuevo



Interfaz de usuario gráfica, Diagrama

El contenido generado por IA puede ser incorrecto.

De esta forma tendremos la transformación de la data desde stagging a la dimensión Oficina del modelo Estrella de nuestro Data Mart.

Procederemos a repetir el proceso con las demás tablas de dimensión

# Cargue dimensión empleado

Rápidamente haremos el mismo proceso

Consultamos todos los datos

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Concatenamos todos los datos del nombre para obtener un nombre completo

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Ordenamos por el id del jefe inmediato

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Aplicamos la validación de registro nuevo y la escritura

Interfaz de usuario gráfica, Diagrama, Aplicación

El contenido generado por IA puede ser incorrecto.

# Cargue Dimensión Cliente

Repetimos el proceso como hemos hecho anteriormente

Seleccionamos los datos

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Concatenamos el nombre uniendo apellido y nombre y concatenamos la línea de dirección y los guardamos en los atributos **nombre\_contacto** y **línea\_direccion1** correspondientemente

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Ordenamos por país

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

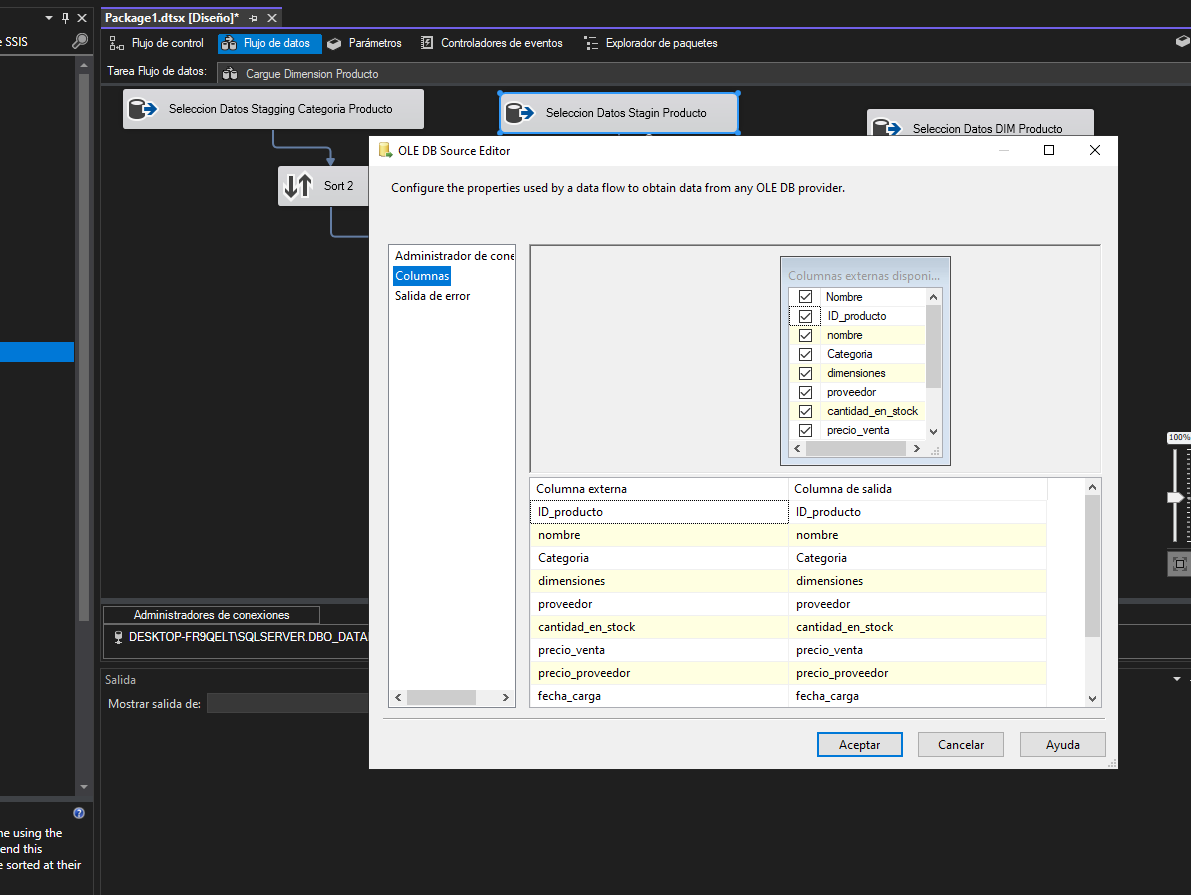
Registramos los registros nuevos

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

# Cargue dimensión producto

Seleccionamos los datos de categoría producto y producto para hacer merge con la descripción, los datos de categoría tendrán la descripción



Ordenamos por categoría

Una captura de pantalla de una red social

El contenido generado por IA puede ser incorrecto.

Filtramos y hacemos un registro de solo los datos nuevos

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

# Cargue dimensión fecha

Para la dimensión tiempo hay que ser mas cuidadoso, el proceso buscará obtener las fechas correctas de los pedidos y los pagos para obtener correctamente el registro de eventos.

El flujo será el siguiente

* Uno para SELECT fecha\_pedido FROM pedido
* Otro para SELECT fecha\_esperada FROM pedido
* Otro para SELECT fecha\_entrega FROM pedido WHERE fecha\_entrega IS NOT NULL
* Otro para SELECT fecha\_pago FROM pago

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Codigo SQL

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3 | **SELECT** **DISTINCT** fecha\_pedido as fecha  **FROM** [DB Stagging Pedido]  **WHERE** (fecha\_pedido **IS** **NOT** **NULL**) | |

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3 | **SELECT** **DISTINCT** fecha\_esperada as fecha  **FROM** [DB Stagging Pedido]  **WHERE** (fecha\_esperada **IS** **NOT** **NULL**) | |

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3 | **SELECT** **DISTINCT** fecha\_entrega as fecha  **FROM** [DB Stagging Pedido]  **WHERE** (fecha\_entrega **IS** **NOT** **NULL**) | |

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3 | **SELECT** **DISTINCT** fecha\_pago **AS** fecha  **FROM** [DB Stagging Pedido]  **WHERE** fecha\_pago **IS** **NOT** **NULL** | |

Hacemos las correspondientes uniones

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Hacemos la transformación de la data

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5 | **YEAR**((DT\_DATE)fecha)  ((**MONTH**((DT\_DATE)fecha) - 1) / 3) + 1  **MONTH**((DT\_DATE)fecha)  **DAY**((DT\_DATE)fecha)  DATEPART("weekday",(DT\_DATE)fecha) | |

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

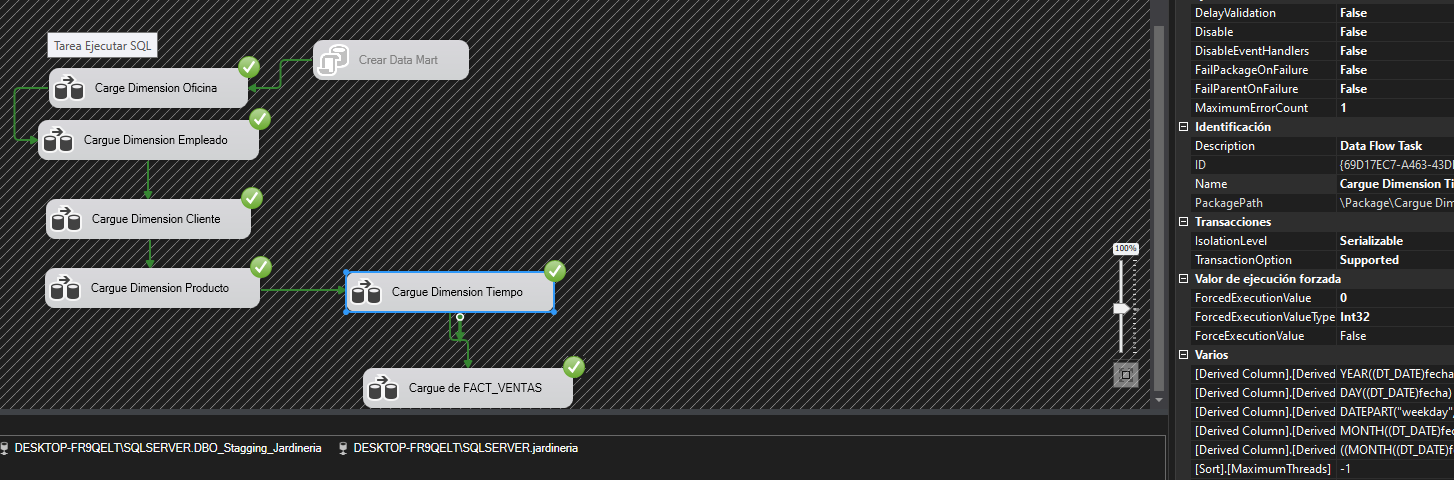
Finalmente hacemos la relación de los datos de las fechas

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

# Cargue FACT\_VENTAS

Finalmente validamos el cargue de los datos y su registro en la base de datos



Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

# Conclusiones

El modelo dimensional desempeña un papel fundamental en el diseño de almacenes de datos, ya que ofrece una estructura clara y eficiente para organizar y analizar grandes volúmenes de información. Al separar los hechos, que representan métricas cuantificables, de las dimensiones, que aportan contexto, no solo optimiza el rendimiento de las consultas, sino que también facilita la interpretación de los datos por parte de los usuarios. Gracias a esta organización, se convierte en una herramienta clave para la inteligencia empresarial, permitiendo a las organizaciones convertir datos en información valiosa para la toma de decisiones.

# Bibliografía

Ahmed, I. (7 de 06 de 2024). *Astera.com*. Obtenido de Astera.com: https://www.astera.com/es/knowledge-center/dimensional-modeling-guide/

Kimball, R. (14 de 02 de 2025). *Wikipedia.org*. Obtenido de Wikepedia: https://es.wikipedia.org/wiki/Modelado\_dimensional