**Evidencia de aprendizaje 3. Proceso de transformación de datos y carga en el data mart final**

**Mateo Lara Aristizábal**

**Juan Diego Urrego Gutiérrez**

**Simón Lara Aristizábal**

**IUDigital**

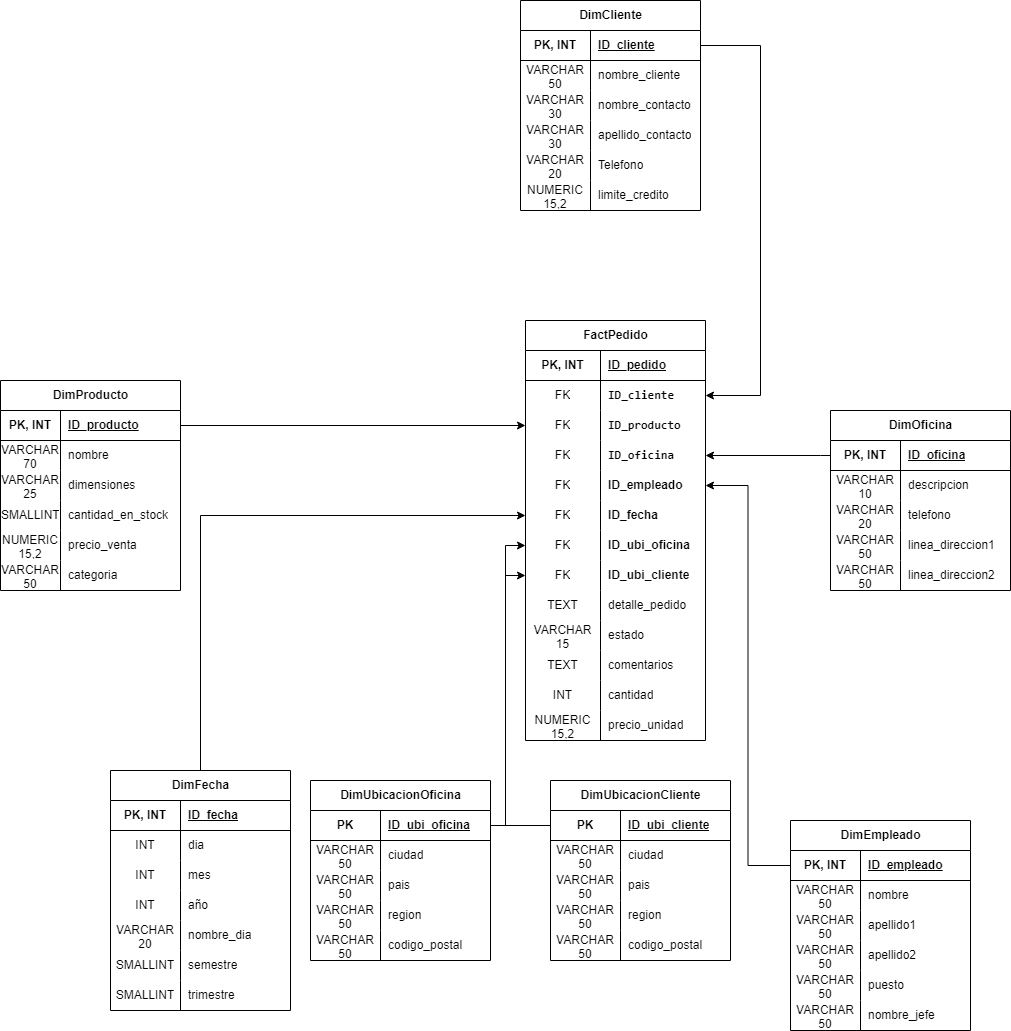
**Bases de Datos II - PREICA2402B010070**

**Victor Hugo Mercado**

**05 de octubre de 2024**

**Introducción**

En este documento se va a mostrar el análisis del modelo estrella para la base de datos jardinería, y como fue el proceso de pasar los datos hacia la tabla staging para de esta forma poder transformarlos y moverlos a la base de datos datamart final, se explicarán también los SQLS usados en la carga y transformación de los respectivos datos.



**Análisis del modelo estrella**

Al mirar el modelo estrella para crear la base de datos staging no dimos cuenta que teníamos que crear 3 tablas para transformar los datos las cuales son (Fecha, Ubicación oficina y Ubicación cliente) y además habían unas tablas que no se iban a usar en el datamart final, asi que luego de analizar y comprender este modelo estrella se procedió a la creación de la base de datos staging

**SQL Creación tabla staging**  
-- Creación de la base de datos staging para transformación de datos

CREATE DATABASE jardineria\_intermedia;

USE jardineria\_intermedia;

-- Tabla para clientes

CREATE TABLE Clientes\_Intermedia (

ID\_cliente INT PRIMARY KEY,

nombre\_cliente VARCHAR(50) NOT NULL,

nombre\_contacto VARCHAR(30),

apellido\_contacto VARCHAR(30),

telefono VARCHAR(20),

limite\_credito NUMERIC(15,2)

);

-- Tabla para productos

CREATE TABLE Producto\_Intermedia (

ID\_producto INT PRIMARY KEY,

nombre VARCHAR(70) NOT NULL,

dimensiones VARCHAR(25),

cantidad\_en\_stock SMALLINT,

precio\_venta NUMERIC(15,2),

Categoria VARCHAR(50)

);

-- Tabla para oficinas

CREATE TABLE Oficinas\_Intermedia (

ID\_oficina INT PRIMARY KEY,

Descripcion VARCHAR(10) NOT NULL,

ciudad VARCHAR(30) NOT NULL,

pais VARCHAR(50) NOT NULL,

region VARCHAR(50),

codigo\_postal VARCHAR(10),

telefono VARCHAR(20) NOT NULL,

linea\_direccion1 VARCHAR(50) NOT NULL,

linea\_direccion2 VARCHAR(50)

);

-- Tabla para fechas

CREATE TABLE Fecha\_Intermedia (

ID\_fecha INT PRIMARY KEY,

Dia INT,

Mes INT,

Año INT,

nombre\_dia VARCHAR(20),

semestre SMALLINT,

trimestre SMALLINT

);

-- Tabla para empleados

CREATE TABLE Empleado\_Intermedia (

ID\_empleado INT PRIMARY KEY,

Nombre VARCHAR(50) NOT NULL,

apellido1 VARCHAR(50) NOT NULL,

apellido2 VARCHAR(50),

Puesto VARCHAR(50),

nombre\_jefe VARCHAR(50)

);

-- Tabla para ubicación de clientes

CREATE TABLE Ubicacion\_Cliente\_Intermedia (

ID\_ubi\_cliente INT PRIMARY KEY,

Ciudad VARCHAR(50),

Region VARCHAR(50),

Pais VARCHAR(50),

codigo\_postal VARCHAR(10)

);

-- Tabla para ubicación de oficinas

CREATE TABLE Ubicacion\_Oficina\_Intermedia (

ID\_ubi\_oficina INT PRIMARY KEY,

Ciudad VARCHAR(50),

Region VARCHAR(50),

Pais VARCHAR(50),

codigo\_postal VARCHAR(10)

);

-- Tabla de hechos para pedidos

CREATE TABLE Pedidos\_Intermedia (

ID\_pedido INT PRIMARY KEY,

ID\_cliente INT,

ID\_producto INT,

ID\_oficina INT,

ID\_empleado INT,

ID\_fecha INT,

ID\_ubi\_cliente INT,

ID\_ubi\_oficina INT,

detalle\_pedido TEXT,

Estado VARCHAR(15),

Comentarios TEXT,

Cantidad INT,

precio\_unidad NUMERIC(15,2),

FOREIGN KEY (ID\_cliente) REFERENCES Clientes\_Intermedia(ID\_cliente),

FOREIGN KEY (ID\_producto) REFERENCES Producto\_Intermedia(ID\_producto),

FOREIGN KEY (ID\_oficina) REFERENCES Oficinas\_Intermedia(ID\_oficina),

FOREIGN KEY (ID\_empleado) REFERENCES Empleado\_Intermedia(ID\_empleado),

FOREIGN KEY (ID\_fecha) REFERENCES Fecha\_Intermedia(ID\_fecha),

FOREIGN KEY (ID\_ubi\_cliente) REFERENCES Ubicacion\_Cliente\_Intermedia(ID\_ubi\_cliente),

FOREIGN KEY (ID\_ubi\_oficina) REFERENCES Ubicacion\_Oficina\_Intermedia(ID\_ubi\_oficina)

);

Como se observa en el código SQL la base de datos staging se llama jardineria\_intermedia y en esta tenemos el modelo estrella pero sin las restricciones de los datos entre las tablas para asi poder mover los datos libremente

**SQL Agregar datos y transformarlos**

-- Insertar datos de clientes desde jardineria a jardineria\_intermedia

INSERT INTO jardineria\_intermedia.dbo.Clientes\_Intermedia (ID\_cliente, nombre\_cliente, nombre\_contacto, apellido\_contacto, telefono, limite\_credito)

SELECT ID\_cliente, nombre\_cliente, nombre\_contacto, apellido\_contacto, telefono, limite\_credito

FROM jardineria.dbo.cliente;

-- Insertar datos de productos desde jardineria a jardineria\_intermedia

INSERT INTO jardineria\_intermedia.dbo.Producto\_Intermedia (ID\_producto, nombre, dimensiones, cantidad\_en\_stock, precio\_venta, Categoria)

SELECT ID\_producto, nombre, dimensiones, cantidad\_en\_stock, precio\_venta, Categoria

FROM jardineria.dbo.producto;

-- Insertar datos de oficinas desde jardineria a jardineria\_intermedia

INSERT INTO jardineria\_intermedia.dbo.Oficinas\_Intermedia (ID\_oficina, telefono, linea\_direccion1, linea\_direccion2)

SELECT ID\_oficina, Descripcion, telefono, linea\_direccion1, linea\_direccion2

FROM jardineria.dbo.oficina;

-- Insertar datos de fechas desde jardineria a jardineria\_intermedia

INSERT INTO jardineria\_intermedia.dbo.Fecha\_Intermedia (ID\_fecha, Dia, Mes, Año, nombre\_dia, semestre, trimestre)

SELECT ID\_pedido, DAY(fecha\_pedido), MONTH(fecha\_pedido), YEAR(fecha\_pedido),

DATENAME(WEEKDAY, fecha\_pedido) AS nombre\_dia,

CASE WHEN MONTH(fecha\_pedido) BETWEEN 1 AND 6 THEN 1 ELSE 2 END AS semestre,

DATEPART(QUARTER, fecha\_pedido) AS trimestre

FROM jardineria.dbo.pedido;

-- Insertar datos de empleados desde jardineria a jardineria\_intermedia

INSERT INTO jardineria\_intermedia.dbo.Empleado\_Intermedia (ID\_empleado, Nombre, apellido1, apellido2, Puesto, nombre\_jefe)

SELECT ID\_empleado, nombre, apellido1, apellido2, puesto, (SELECT CONCAT(nombre,' ',apellido1,' ',apellido2) FROM jardineria.dbo.empleado WHERE ID\_empleado = e.ID\_jefe)

FROM jardineria.dbo.empleado e;

-- Insertar datos de ubicación de clientes desde jardineria a jardineria\_intermedia

INSERT INTO jardineria\_intermedia.dbo.Ubicacion\_Cliente\_Intermedia (ID\_ubi\_cliente, Ciudad, Region, Pais, codigo\_postal)

SELECT ID\_cliente, ciudad, region, pais, codigo\_postal

FROM jardineria.dbo.cliente;

-- Insertar datos de ubicación de oficinas desde jardineria a jardineria\_intermedia

INSERT INTO jardineria\_intermedia.dbo.Ubicacion\_Oficina\_Intermedia (ID\_ubi\_oficina, Ciudad, Region, Pais, codigo\_postal)

SELECT ID\_oficina, ciudad, region, pais, codigo\_postal

FROM jardineria.dbo.oficina;

En este codigo se mueven los datos de la tabla jardinería a la tabla staging, aunque como se observa se están transformando los datos:  
**tabla oficinas**

solo se mueven (ID\_oficina, telefono, linea\_direccion1, linea\_direccion2)

**Tabla clientes**

se mueve (ID\_cliente, nombre\_cliente, nombre\_contacto, apellido\_contacto, telefono, limite\_credito)

**Tabla fecha**

se agarran los datos de fecha pedido y se separan por dia, mes y año, además de que se agrega que dia de la semana fue el pedido, además de que al momento de que el mes este entre los 6 primeros del año, se agrega semestre 1, si no seria el 2do semestre del año y con los trimestres se pone el quarter para que los 12 meses estén separados por cada 3 meses.

**Tabla empleados**

El cambio de esta tabla fue en el nombre jefe para que no fuera un id, si no, se puso el nombre completo del jefe o supervisor del empleado, ya que la consulta busca el id del empleado y agrega al jefe de este  
  
**tablas ubicación**

Se agrego en las tablas la (Ciudad, Region, Pais, codigo\_postal) de los clientes y las oficinas respectivamente y así haya mas orden en las tablas.  
  
**Nota:** No se logro usar las herramientas ETL para que este proceso fuera mas fácil ya que al usar la aplicación de visual studio no se lograba hacer la conexión de origen con la de destino en las bases de datos, asi que todo se hizo con consultas SQL en SSMS