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

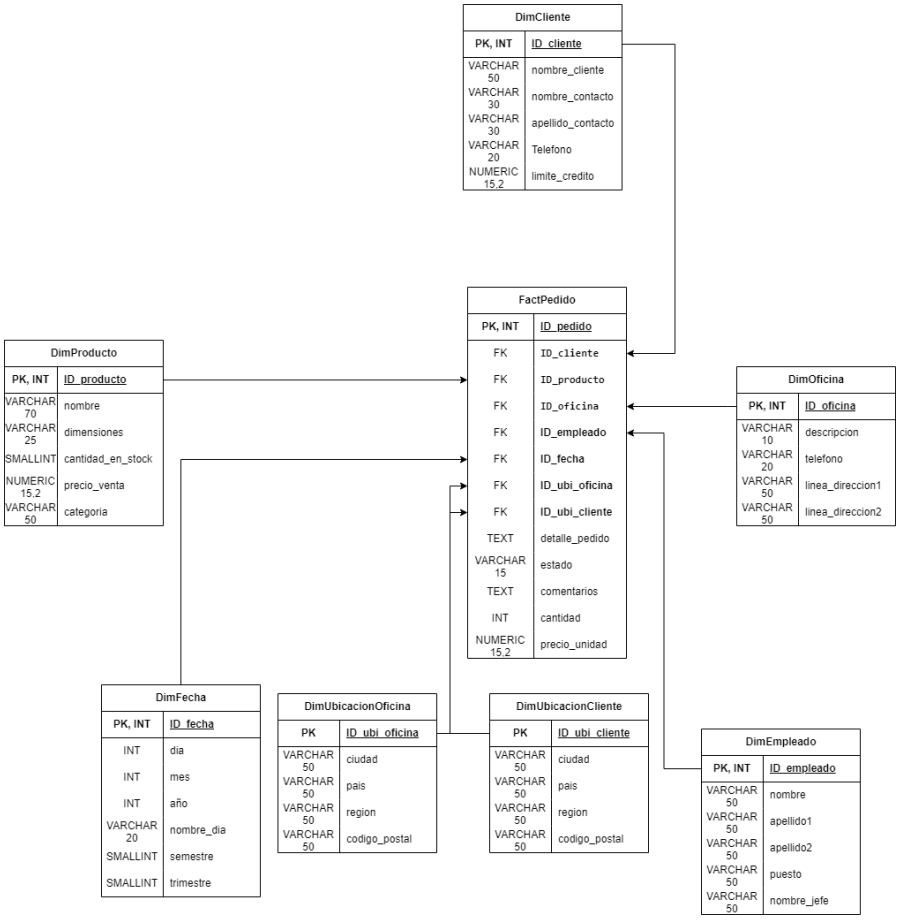
Bases de Datos II - PREICA2402B010070

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

```sql
-- 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)
);
```

```sql
-- 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