

Creación de Data Mart final

Jefree Tamayo Herrera

Facultad de Ingeniería de Software y Datos, Institución Universitaria Digital De
Antioquia

Antonio Jesús Valderrama

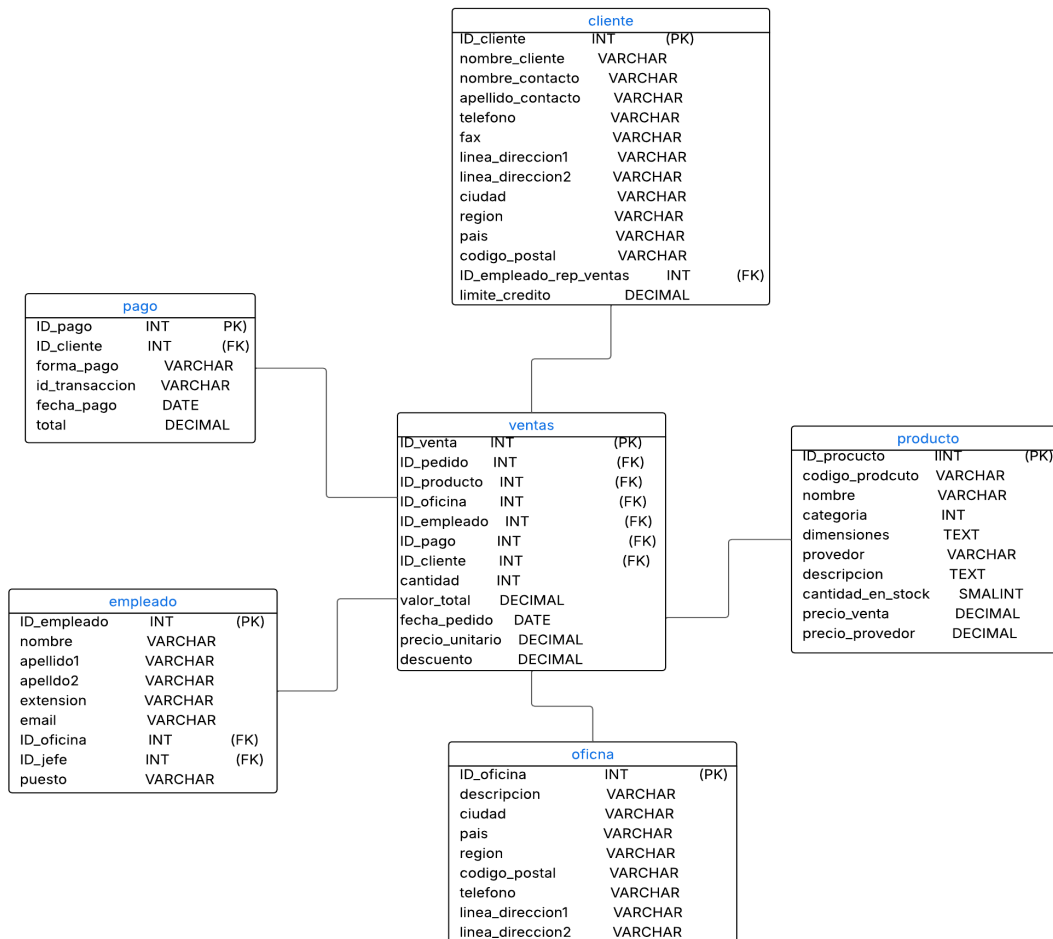
30 de septiembre del 2025

Proceso de creación de data mart

En este documento documentaremos el proceso de creación de toda nuestra base de datos, aquí podemos encontrar desde la preparación con nuestro modelo estrella, la transformación y extracción de los datos y por ultimo vemos nuestro data mart final.

Preparación

Como primer punto para empezar nuestro proceso, creamos un modelo estrella como base para empezar toda nuestra base de datos, teníamos como referencia la base de datos jardinería y de allí creamos el siguiente modelo estrella.



Aquí podemos ver nuestras tablas de dimensiones y nuestra tabla de hechos, como primero se definió la base de hechos, que era una única tabla llamada ventas que representa cada transacción por venta y de allí salían nuestras tablas de dimensiones como empleado, cliente, pago, producto etc., nuestra tabla de hechos contiene métricas numéricas lo cual ayuda a diferentes análisis como total, descuento, precio unitario y las tablas de dimensiones contextualizan las ventas ya que se relacionan con los productos, la oficina, el empleado que vendió, el cliente que compro y su respectivo pago. Ahora pasamos a la siguiente fase.

Extracción de datos

Ahora con nuestro esqueleto creado que es el modelo estrella, proseguimos a construir tablas staging con el objetivo de que esta área de staging actúa como puente, donde se cargan los registros de nuestras base principal (jardinería) tal como viene para luego ser transformados y normalizados según nuestro modelo estrella, en esta etapa los datos pueden contener duplicados o nulos y función principal es mantener la trazabilidad hacia el origen.

```
68 • CREATE TABLE stg_pago (
69     id_stg INT AUTO_INCREMENT PRIMARY KEY,
70     bk_pago INT,
71     bk_cliente INT,
72     forma_pago VARCHAR(40),
73     bk_transaccion VARCHAR(50),
74     fecha_pago DATE,
75     total NUMERIC(15,2),
76     origen VARCHAR(50),
77     fecha_carga DATETIME,
78     estado_registro VARCHAR(50)
79 );
```

```
41 • CREATE TABLE stg_pedido (
42     id_stg INT AUTO_INCREMENT PRIMARY KEY,
43     bk_pedido INT,
44     fecha_pedido DATE,
45     fecha_esperada DATE,
46     fecha_entrega DATE,
47     estado VARCHAR(15),
48     comentarios TEXT,
49     bk_cliente INT,
50     origen VARCHAR(50),
51     fecha_carga DATETIME,
52     estado_registro VARCHAR(50)
53 );
--
```

En esta fase hicimos un cambio importante, cambiamos la tabla de hechos de ventas que teníamos en el modelo estrella por dos tablas de hechos que tienen relación entre sí, que son la tabla de pago y pedido ya que vimos que para hacer análisis de negocio era más productivo tener estas dos tablas, una donde podemos analizar los pagos realizados por los clientes y en la tabla los pedidos realizados por cada cliente, también podemos que añadimos las Business key en cada tabla que es el identificador de cada negocio, persona como la bk de cliente o la bk de pago, estas tablas son un espejo de nuestras base de datos original.

```

98 • INSERT INTO stg_empleado (bk_empleado, nombre, apellido1, apellido2, extension, ,
99 SELECT
100 ID_empleado AS bk_empleado,
101 nombre,
102 apellido1,
103 apellido2,
104 extension,
105 email,
106 ID_oficina,
107 ID_jefe,
108 puesto,
109 NOW() AS fecha_carga,
110 "OLAP jardineria" AS origen,
111 "activo" AS estado_registro
112 FROM jardineria.empleado;
113

```

Este es un ejemplo donde extraemos los datos crudos de nuestro base de datos principal y los pasamos tal cual a nuestras tablas staging.

	id_stg	bk_empleado	nombre	apellido1	apellido2	extension	email	ID_oficina	ID_jefe	puesto	fecha_carga
▶	1	1	Marcos	Magaña	Perez	3897	marcos@jardineria.es	8	NULL	Director General	2025-09-15 23:54:39
	2	2	Ruben	López	Martinez	2899	rlopez@jardineria.es	8	1	Subdirector Marketing	2025-09-15 23:54:39
	3	3	Alberto	Soria	Carrasco	2837	asoria@jardineria.es	8	2	Subdirector Ventas	2025-09-15 23:54:39
	4	4	Maria	Solís	Jerez	2847	msolis@jardineria.es	8	2	Secretaria	2025-09-15 23:54:39
	5	5	Felipe	Rosas	Marquez	2844	frosas@jardineria.es	8	3	Representante Ventas	2025-09-15 23:54:39
	6	6	Juan Carlos	Ortiz	Serrano	2845	cortiz@jardineria.es	8	3	Representante Ventas	2025-09-15 23:54:39
	7	7	Carlos	Soria	Jimenez	2444	csoria@jardineria.es	4	3	Director Oficina	2025-09-15 23:54:39
	8	8	Mariano	López	Murcia	2442	mlopez@jardineria.es	4	7	Representante Ventas	2025-09-15 23:54:39

Aquí podemos el resultado de nuestros datos pasado a las tablas staging, ahora con esta fase realizada podemos pasar a la siguiente.

Data Mart final.

Ahora llegamos a nuestra fase final que es transformar nuestros datos de las tablas staging que son la réplica de nuestros datos originales para presentarlos de manera organizado, sin duplicados, datos nulos u otro tipo de dato pero preservando nuestra trazabilidad, como primer paso es crear nuestras tablas de dimensiones y hechos finales para después realizar consultas de transformación para deshacer los duplicados y ordenar nuestros datos.

```
6 • ○ CREATE TABLE dim_empleado (  
7     id_empleado_dim INT AUTO_INCREMENT PRIMARY KEY NOT NULL,  
8     bk_empleado INT NOT NULL,  
9     nombre VARCHAR(50) NOT NULL,  
10    apellido1 VARCHAR(50) NOT NULL,  
11    apellido2 VARCHAR(50) NOT NULL,  
12    email VARCHAR(50) NOT NULL,  
13    id_jefe INT  
14 );  
--
```

Este es un ejemplo de nuestra tabla de dimensión empleado, vemos que sigue manteniendo su respectiva Business key para mantener trazabilidad y los atributos que queremos presentar de manera ordenar, las tablas de la data mart final no necesariamente tiene que tener los mismo atributos que nuestras tablas staging si no que podemos agregar solo los que queremos que se presenten.

	id_empleado_dim	bk_empleado	nombre	apellido1	apellido2	email	id_jefe
▶	1	1	Marcos	Magaña	Perez	marcos@jardineria.es	NULL
	2	2	Ruben	López	Martinez	rlopez@jardineria.es	1
	3	3	Alberto	Soria	Carrasco	asoria@jardineria.es	2
	4	4	Maria	Solís	Jerez	msolis@jardineria.es	2
	5	5	Felipe	Rosas	Marquez	frosas@jardineria.es	3
	6	6	Juan Carlos	Ortiz	Serrano	cortiz@jardineria.es	3
	7	7	Carlos	Soria	Jimenez	csoria@jardineria.es	3
	8	8	Mariano	López	Murcia	mlopez@jardineria.es	7
	9	9	Lucio	Campoamor	Martín	lcampoamor@jardineria.es	7
	10	10	Hilario	Rodriguez	Huertas	hrodriguez@jardineria.es	7

Esta es la tabla de dimensión empleado final con los datos organizados de manera clara y podemos ver que tiene menos valores que las tablas anteriores ya que en este caso solo queremos algunos atributos o características para hacer el análisis de negocios o simplemente para mostrarlo a nuestro público. En esta parte del proceso decidimos añadir otra tabla de dimensiones que es la tabla de tiempo, que nos ayuda a ver los pedidos realizados por año o consultar las ventas por semestre, trimestre, mes o año.

```

42 • CREATE TABLE dim_tiempo (
43     id_tiempo INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
44     fecha DATE NOT NULL UNIQUE,
45     anio INT NOT NULL,
46     semestre TINYINT NOT NULL,
47     trimestre TINYINT NOT NULL,
48     mes TINYINT NOT NULL,
49     nombre_mes VARCHAR(20) NOT NULL,
50     dia_mes TINYINT NOT NULL,
51     nombre_dia VARCHAR(20) NOT NULL,
52     numero_semana TINYINT NOT NULL
53 );

```

Y por último solo nos queda agregar todos nuestros datos limpios a nuestras tablas finales.

```

148 • INSERT INTO fact_pago (id_cliente_dim, id_tiempo, bk_pago, bk_cliente, bk_transaccion, forma_pago, monto_pago)
149 SELECT
150     dc.id_cliente_dim,
151     dt.id_tiempo,
152     sp.bk_pago,
153     sp.bk_cliente,
154     sp.bk_transaccion,
155     sp.forma_pago,
156     sp.total
157 FROM stg_jardineria.stg_pago sp
158 JOIN dim_cliente dc
159 ON dc.id_cliente_dim = sp.bk_cliente
160 JOIN dim_tiempo dt
161 ON sp.fecha_pago = dt.fecha;

```

Ese es un ejemplo de insertado de nuestros datos limpios hacia nuestra tabla de hechos pago final.

Conclusiones

La construcción del modelo estrella de ventas a partir de una base de datos me permitió transformar los datos operacionales de información analítica organizada y confiable. Las tablas Staging funcionaron como un espejo de la base de datos fuente, preservando los datos originales con sus duplicados y valores nulos, mientras que en el data mart final se normalizaron, depuraron y estructuraron en dimensiones y hechos para ser presentadas de manera coherente y organizada sin olvidar de preservar nuestra trazabilidad.