



Nombre: Mauricio Bautista Rubiano
Carrera: Ingeniería de Software y Datos.
Grupo: PREICA2502B010064

Docente: Antonio Jesús Valderrama Jaramillo
Materia: Bases de Datos II

Introducción

En este documento tengo la intención de presentar el diseño de un modelo estrella para un almacén de datos (Data Warehouse) construido a partir de la base “Jardinería”, pero adaptada localmente, en Windows 11 a PostgreSQL 12.0. Lo que deseo es poder habilitar el análisis eficiente de ventas, clientes, productos, representantes y oficinas a lo largo del tiempo. Se detalla la tabla de hechos principal (ventas por línea de pedido), las dimensiones asociadas, las relaciones e integridad referencial, y se proporciona un diagrama del modelo, además de definiciones de campos y tipos de datos.

Averiguando, creo que un modelo estrella facilita análisis de desempeño, tendencias y segmentaciones, reduciendo complejidad, mejorando así los tiempos de respuesta.

Objetivos

- Diseñar una tabla de hechos para ventas a nivel de línea de pedido.
- Definir dimensiones relevantes (tiempo, cliente, producto, empleado, oficina, estado) y sus atributos.
- Establecer relaciones que aseguren integridad referencial.
- Ilustrar el modelo con un diagrama legible.

Planteamiento del problema

La base “Jardinería” contiene entidades como pedidos, detalles de pedido, clientes, productos, empleados y oficinas, diseñadas para la captura de información de las operaciones del negocio. Para análisis de negocio (p. ej., ventas por categoría y mes, desempeño por representante y oficina), con lo cual, creo que se requiere un repositorio analítico que lleve un mejor control.

Análisis del problema

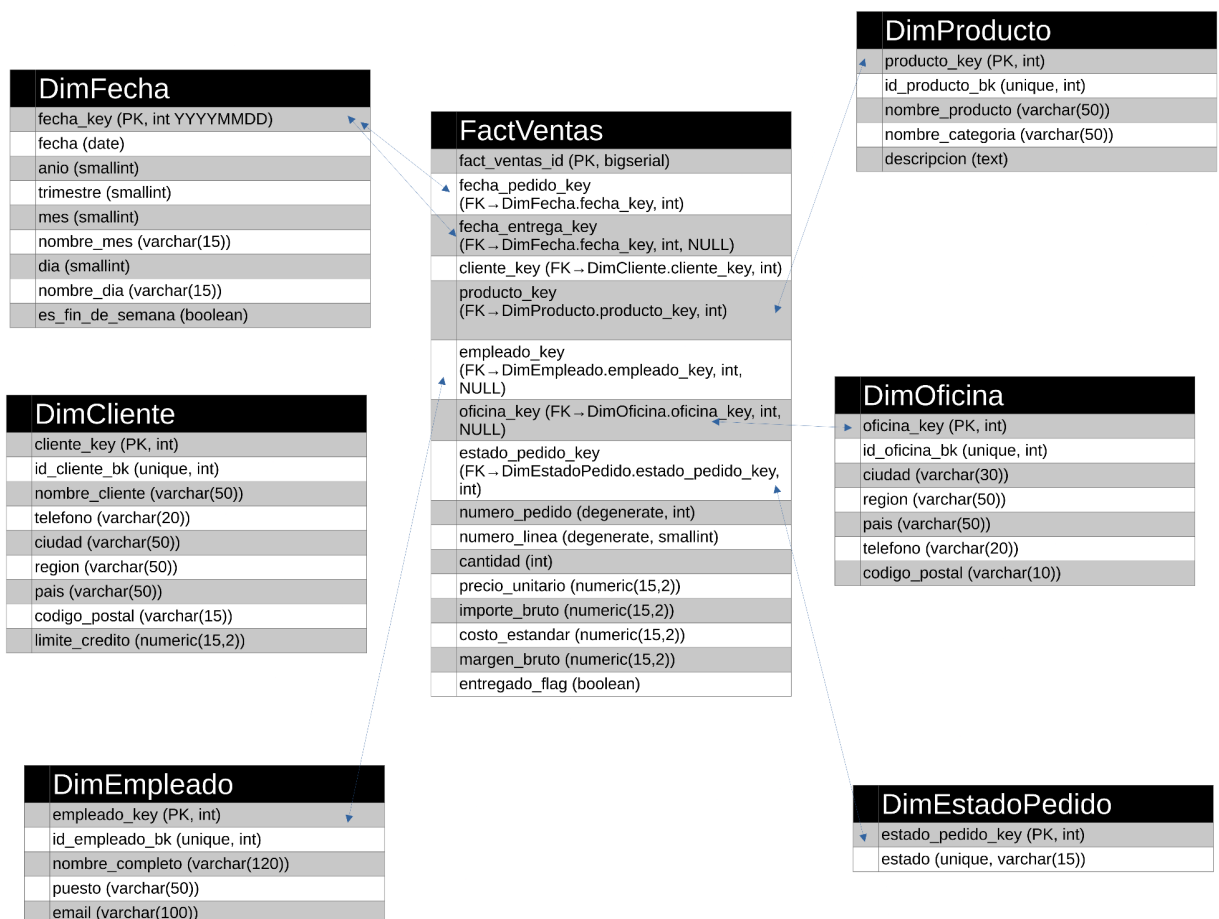
- Granularidad:** la analítica de ventas necesita el nivel de detalle de cada línea de pedido.
- Métricas:** cantidad, importe, costo y margen son medidas fundamentales; el precio unitario debe ser el del momento de la venta (del detalle).
- Dimensiones:** tiempo (para series y estacionalidad), cliente (segmentación geográfica y de valor), producto (incluida su categoría), empleado y oficina (rendimiento comercial), estado de pedido (ciclo y cumplimiento).
- Integridad:** las FKs desde hechos a dimensiones deben existir; se recomiendan miembros “Unknown” para datos faltantes.

•**Evolución:** para simplificar, se propone SCD Tipo 1 (sobrescritura). En fases futuras, puede adoptarse SCD Tipo 2 para historizar cambios.

Descripción del modelo estrella propuesto.

El modelo de estrella diseñado para “DW Jardinería” tiene como su objetivo habilitar el análisis de ventas con un esquema que a mi forma de ver es sencillo, consistente y de alto rendimiento. La granularidad de la tabla de hechos principal es “una fila por línea de pedido” (cada registro del detalle de pedido). Esto en teoría debería permitir analizar cantidades, importes y márgenes al nivel más detallado y agregar fácilmente por cualquier dimensión.

Diseño (Imagen) del modelo estrella donde se puedan observar las dimensiones, la tabla de hechos, sus campos, tipos de datos y relaciones.



Lista de dimensiones propuestas.

DimFecha	fecha_key int (PK) fecha date anio smallint trimestre smallint mes smallint nombre_mes varchar(15) dia smallint nombre_dia varchar(15) es_fin_de_semana boolean.
DimCliente	cliente_key int (PK) id_cliente_bk int (UNQ) nombre_cliente varchar(50) telefono varchar(20) ciudad varchar(50) region varchar(50) pais varchar(50) codigo_postal varchar(15) limite_credito numeric(15,2).
DimProducto	producto_key int (PK) id_producto_bk int (UNQ) nombre_producto varchar(50) nombre_categoria varchar(50) descripcion text.
DimEmpleado	empleado_key int (PK) id_empleado_bk int (UNQ) nombre_completo varchar(120) puesto varchar(50) email varchar(100).
DimOficina	oficina_key int (PK) id_oficina_bk int (UNQ) ciudad varchar(30), region varchar(50) pais varchar(50) telefono varchar(20) codigo_postal varchar(10)
DimEstadoPedido	estado_pedido_key int (PK) estado varchar(15) (UNQ).

Detalla la tabla de hechos, con sus campos y tipos de datos

Granularidad: una fila por línea de pedido (cada registro de detalle_pedido).

Clave primaria:

fact_ventas_id: bigserial. Identificador único de la fila (PK).

Claves foráneas:

fecha_pedido_key: int. FK a dim_fecha(fecha_key). Fecha en la que se realizó el pedido. NOT NULL.

fecha_entrega_key: int. FK a dim_fecha(fecha_key). Fecha en la que se entregó (si aplica). Puede ser NULL.

cliente_key: int. FK a dim_cliente(cliente_key). Cliente asociado al pedido. NOT NULL.

producto_key: int. FK a dim_producto(producto_key). Producto vendido. NOT NULL.

empleado_key: int. FK a dim_empleado(empleado_key). Representante de ventas (si existe). Puede ser NULL.

oficina_key: int. FK a dim_oficina(oficina_key). Oficina del representante. Puede ser NULL.

estado_pedido_key: int. FK a dim_estado_pedido(estado_pedido_key). Estado del pedido. NOT NULL.

Dimensiones degeneradas (se quedan en la tabla de hechos):

numero_pedido: int. Número del pedido del sistema transaccional. NOT NULL.

numero_linea: smallint. Número de línea dentro del pedido. NOT NULL.

Medidas (métricas):

cantidad: int. Unidades vendidas en la línea. NOT NULL (idealmente > 0).

precio_unitario: numeric(15,2). Precio por unidad usado en la línea. NOT NULL (≥ 0).

importe_bruto: numeric(15,2). Resultado de cantidad \times precio_unitario. NOT NULL.

costo_estandar: numeric(15,2). Aproximación de costo (cantidad \times precio_proveedor). NOT NULL.

margen_bruto: numeric(15,2). Importe_bruto – costo_estandar. NOT NULL.

entregado_flag: boolean. TRUE si el pedido fue entregado (fecha_entrega no es NULL). NOT NULL.

Conclusiones

Con este trabajo intenté llevar la base transaccional de “Jardinería” a un modelo estrella sencillo para análisis. La tabla de hechos fact_ventas, lleva una granularidad a nivel de línea de pedido, que al mismo tiempo me permitió definir con claridad las medidas principales (cantidad, importe, costo y margen) y relacionarlas con dimensiones que considero clave: fecha, cliente, producto, empleado, oficina y estado del pedido. Siento que esta estructura cumple el objetivo de facilitar consultas de desempeño y tendencias sin tener que unir demasiadas tablas.

Aunque el diseño es básico, creo que mantiene una buena coherencia: usé claves sustitutas en las dimensiones, y procuré asegurar la integridad referencial. También traté de que los tipos de datos fueran consistentes con PostgreSQL. Elegí PostgreSQL debido a que estoy implementando Odoo en la empresa de mi papá, además de que es el tipo de Postgres que mejor se me da debido a ello, creo que recordar cómo plantear una base de datos sin el código python y librerías de Odoo como intermediarios probablemente podrá beneficiarme para poder crear mejores aplicaciones, módulos, y a sobre todo poder ser más crítico con las decisiones.

Si bien reconozco que es una primera versión para este ejercicio. Sé que probablemente hay varias cosas por mejorar, y que harán falta reglas de calidad de datos más estrictas, tal vez nuevas columnas/datos que no haya anticipado y que en un contexto real puedan ser útiles. Aun así, pienso que este punto de partida es útil para filtrar y manejar datos que puedan ayudar a la hipotética Jardinería de este ejercicio a crear estrategias y tomar decisiones que impacten positivamente a sus ventas y/o a la rentabilidad de sus procesos.

Bibliografía.

Debido a que diseñé el diagrama en LibreOffice Draw, incluiré un PDF con la imagen en la subida de este trabajo, por lo que, para este caso no incluiré links hacia LucidChart o aplicaciones web similares.