

Entrega N°2

Empresa de Muebles

INTRODUCCIÓN

La base de datos relacional que propongo desarrollar para este proyecto está basada en una empresa dedicada a la venta de muebles y artículos de decoración para el hogar. La compañía ofrece una amplia variedad de productos, como mesas, sillas, sofás, estanterías y muebles a medida, que provienen tanto de fabricantes nacionales como de importadores internacionales. Además, las ventas se realizan de forma mayorista a locales de decoración y minorista a clientes finales, tanto en tiendas físicas como a través de una plataforma online.

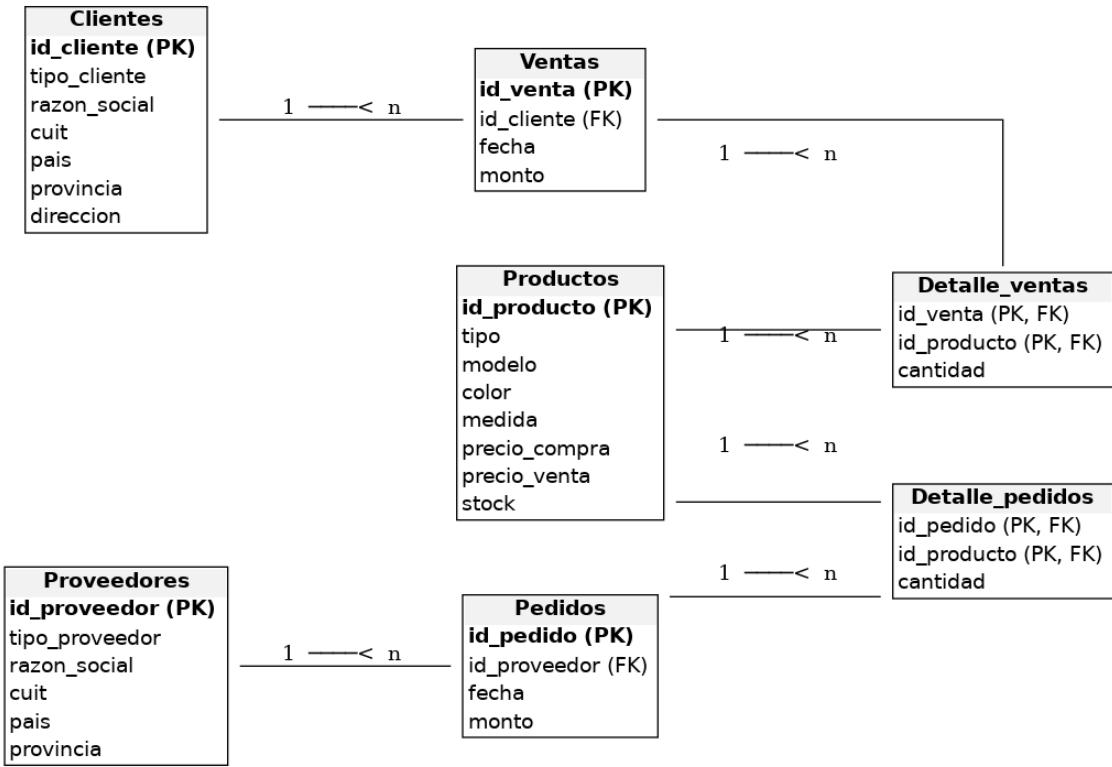
OBJETIVO

La base de datos buscará centralizar información de proveedores, productos, clientes, pedidos de compra y ventas, asegurando trazabilidad de inventario y operaciones. La estructura permitirá gestionar de manera integrada a los proveedores, los productos, los clientes, así como los pedidos de compra y las ventas, garantizando una visión completa de las operaciones del negocio.

SITUACIÓN PROBLEMÁTICA

La empresa de muebles enfrenta dificultades para gestionar de manera unificada la información de proveedores, productos, pedidos y ventas. Actualmente, estos datos se encuentran dispersos en distintos registros, lo que complica la trazabilidad de los productos, la administración del inventario y el control de operaciones comerciales.

DIAGRAMA ENTIDAD RELACIÓN



El siguiente diagrama E-R representa la estructura actualizada de la base de datos diseñada para la empresa de muebles. En esta segunda versión se mantienen las entidades principales: Clientes, Proveedores, Productos, Pedidos y Ventas, y se incorporan las tablas Detalle_pedidos y Detalle_ventas para reflejar de forma más precisa las relaciones de muchos a muchos entre los productos y las operaciones comerciales.

Las relaciones muestran que cada cliente puede realizar múltiples ventas y que cada proveedor puede registrar varios pedidos. Además, cada pedido y cada venta pueden incluir distintos productos, permitiendo detallar las cantidades y precios asociados a cada transacción. De esta manera, el modelo logra una trazabilidad completa de las operaciones de compra y venta, facilitando el control de stock, el cálculo de montos y el análisis de productos.

LISTADO DE TABLAS

Tabla Clientes

Campo	Tipo Dato	PK FK NULL	
id_cliente	INT	X	NO
tipo_cliente	ENUM('mayorista','minorista')		NO

Campo	Tipo Dato	PK FK NULL
razon_social	VARCHAR(255)	NO
cuit	VARCHAR(50)	NO
pais	VARCHAR(100)	SÍ
provincia	VARCHAR(100)	SÍ
direccion	VARCHAR(255)	SÍ

Tabla Proveedores

Campo	Tipo Dato	PK FK NULL
id_proveedor	INT	X NO
tipo_proveedor	ENUM('nacional','internacional')	NO
razon_social	VARCHAR(255)	NO
cuit	VARCHAR(50)	NO
pais	VARCHAR(100)	SÍ
provincia	VARCHAR(100)	SÍ

Tabla Productos

Campo	Tipo Dato	PK FK NULL
id_producto	INT	X NO
tipo	VARCHAR(50)	NO
modelo	VARCHAR(100)	NO
color	VARCHAR(50)	SÍ
medida	VARCHAR(50)	SÍ
precio_compra	DECIMAL(10,2)	NO
precio_venta	DECIMAL(10,2)	NO
stock	INT	SÍ

Tabla Pedidos

Campo	Tipo Dato	PK	FK	NULL
id_pedido	INT	X	NO	
id_proveedor	INT		X	SÍ
fecha	DATE			NO
monto	DECIMAL(10,2)			NO

Tabla Ventas

Campo	Tipo Dato	PK	FK	NULL
id_venta	INT	X	NO	
id_cliente	INT		X	SÍ
fecha	DATE			NO
monto	DECIMAL(10,2)			NO

Tabla Detalle_pedidos

Campo	Tipo Dato	PK	FK	NULL
id_pedido	INT	X	X	NO
id_producto	INT	X	X	NO
cantidad	INT			NO

Tabla Detalle_ventas

Campo	Tipo Dato	PK	FK	NULL
id_venta	INT	X	X	NO
id_producto	INT	X	X	NO
cantidad	INT			NO

LISTADO DE VISTAS

1) vista_detalles_pedidos

Descripción: muestra el detalle de cada pedido realizado a proveedores, incluyendo cada producto del pedido, su cantidad, el precio de compra vigente y el costo total por ítem.

Objetivo de uso: facilitar el control de costos por pedido, auditorías de compras y conciliación con facturas de proveedor.

Tablas que la componen:

Pedidos (pe) – cabecera del pedido (id_pedido, fecha, monto). Detalle_pedidos (dp) – productos y cantidades por pedido.

Productos (pr) – precios de compra y atributos del producto.

2) vista_detalles_ventas

Descripción: desglosa cada venta a nivel de producto, indicando cliente, producto, cantidad, precio de venta y total por ítem.

Objetivo de uso: permitir análisis rápidos de ingresos por venta, control de precios aplicados y verificación del cálculo de monto en la cabecera.

Tablas que la componen:

Ventas (ve) – cabecera de la venta (id_venta, fecha, monto).

Clientes (cl) – datos del cliente.

Detalle_ventas (dv) – productos y cantidades por venta.

Productos (pr) – precios de venta y atributos del producto.

3) vista_historial_ventas_cliente

Descripción: consolida el historial de ventas por cliente, mostrando cada operación con sus productos, cantidades y el total por producto, junto al monto de la venta.

Objetivo de uso: analizar el comportamiento de compra por cliente (frecuencia, mix de productos, ticket promedio) y generar reportes comerciales.

Tablas que la componen:

Clientes (cl) – cliente y tipo (mayorista/minorista).

Ventas (ve) – cabecera de venta.

Detalle_ventas (dv) – productos y cantidades por venta.

Productos (pr) – precio de venta y atributos.

4) vista_historial_pedidos_proveedor

Descripción: concentra el historial de pedidos por proveedor, con el detalle de productos y cantidades, precio de compra y subtotal por ítem, además del monto del pedido.

Objetivo de uso: evaluar desempeño de proveedores (volumen, mix, costos), planificar reaprovisionamiento y negociar condiciones.

Tablas que la componen:

Proveedores (pv) – datos del proveedor.

Pedidos (pe) – cabecera de pedido.

Detalle_pedidos (dp) – productos y cantidades por pedido.

Productos (pr) – precio de compra y atributos.

5) vista_productos_mas_vendidos

Descripción: ranking de productos por unidades vendidas y monto total facturado.

Objetivo de uso: identificar productos “estrella” y tendencias de demanda para decisiones de stock, pricing y campañas comerciales.

Tablas que la componen:

Productos (pr) – base de productos.

Detalle_ventas (dv) – cantidades vendidas por producto.

(Se agrupa y suma por producto; el precio proviene de Productos).

LISTADO DE FUNCIONES

1) Función: calcular_ventas_producto

Descripción: esta función recibe como parámetro el identificador de un producto (producto_id) y calcula el total de ventas acumuladas para ese producto. Multiplica la cantidad vendida por el precio de venta registrado en la tabla Productos. En caso de que el producto no haya sido vendido, la función devuelve el valor 0 para evitar resultados nulos.

Objetivo: permitir obtener de forma automática el total facturado por cada producto, facilitando la elaboración de reportes de ingresos, el análisis de productos más rentables y el control de desempeño comercial sin necesidad de escribir consultas manuales complejas.

Tablas que manipula:

Detalle_ventas: contiene la cantidad vendida de cada producto en cada venta.

Productos: aporta el precio de venta utilizado en el cálculo del total.

2) Función: calcular_valor_stock_producto

Descripción: esta función calcula el valor económico del stock de un producto determinado (producto_id). Para ello, multiplica el stock actual por el precio de compra correspondiente. Si el producto no existe o su stock es nulo, la función devuelve el valor 0, asegurando que no se generen resultados inválidos.

Objetivo: brindar una herramienta rápida para estimar el valor total invertido en inventario, lo cual es útil para reportes contables, control de existencias y toma de decisiones sobre reposición de productos.

Tablas que manipula:

Productos: utiliza los campos stock y precio_compra para calcular el valor total almacenado.

LISTADO DE STORED PROCEDURES

1) Stored Procedure: insertar_pedido

Descripción: procedimiento para registrar un pedido a un proveedor en dos pasos:

1. Inserta la cabecera del pedido en la tabla Pedidos.
2. Inserta dos renglones de detalle en Detalle_pedidos (producto y cantidad para cada ítem).

Objetivo / Beneficio: estandarizar y simplificar la carga operativa de pedidos, evitando que el usuario tenga que recordar múltiples INSERTs y asegurando la consistencia entre la cabecera del pedido y sus ítems. Facilita la generación de reportes de compras y el uso de vistas como vista_detalles_pedidos.

Tablas que lo componen / con las que interactúa:

Pedidos: crea el registro de cabecera (campos: id_proveedor, fecha, monto).

Detalle_pedidos: inserta dos líneas de producto por pedido (campos: id_pedido, id_producto, cantidad).

2) Stored Procedure: insertar_venta

Descripción: Procedimiento para registrar una venta a un cliente en dos pasos:

1. Inserta la cabecera de la venta en la tabla Ventas.
2. Inserta dos renglones de detalle en Detalle_ventas (producto y cantidad para cada ítem). Al crear los detalles se activan los triggers configurados sobre Detalle_ventas, que recalculan el monto de la cabecera y descuentan stock en Productos.

Objetivo / Beneficio: Automatizar el alta de ventas asegurando que, al cargar los renglones, la base actualice monto total y stock sin pasos manuales adicionales. Reduce errores de cálculo y mantiene el inventario actualizado para reportes como vista_detalles_ventas, vista_historial_ventas_cliente y vista_productos_mas_vendidos.

Tablas que lo componen / con las que interactúa:

Ventas: crea el registro de cabecera (campos: id_cliente, fecha, monto).

Detalle_ventas: inserta dos líneas de producto por venta (campos: id_venta, id_producto, cantidad).

Productos: se actualiza vía triggers al descontar stock según la cantidad vendida.

LISTADO DE TRIGGERS

1) Trigger: after_detalle_ventas_insert

Descripción: se ejecuta automáticamente después de insertar un nuevo registro en la tabla

Detalle_ventas. Calcula el monto total de la venta sumando la cantidad por el precio de venta de cada producto asociado, y actualiza el campo monto en la tabla Ventas. Además, descuenta del stock la cantidad vendida en la tabla Productos, reflejando la reducción del inventario.

Objetivo: Asegurar la coherencia automática entre las tablas de ventas, productos y stock. Evita que el usuario tenga que recalcular manualmente montos o actualizar existencias, garantizando la integridad de los datos.

Tablas afectadas / situaciones donde se acciona:

Detalle_ventas: se dispara tras un INSERT.

Ventas: actualiza el monto total (monto).

Productos: descuenta stock (stock = stock - cantidad).

2) Trigger: after_detalle_ventas_update

Descripción: se activa después de una actualización (UPDATE) en la tabla Detalle_ventas. Recalcula el monto total de la venta afectada, multiplicando la cantidad actualizada por el precio de venta de cada producto. Luego, vuelve a actualizar el stock del producto restando la nueva cantidad registrada.

Objetivo: mantener actualizados los totales y el inventario ante cambios en las cantidades vendidas. Previene inconsistencias en el stock o montos si un usuario modifica un registro de venta existente.

Tablas afectadas / situaciones donde se acciona:

Detalle_ventas: se dispara tras un UPDATE.

Ventas: recalcula el monto total (monto).

Productos: ajusta el stock descontando la nueva cantidad (stock = stock - NEW.cantidad).