

Introducción:

La presente propuesta se centra en el diseño y la implementación de una base de datos integral para respaldar las operaciones de una tienda de ropa en línea. Esta tienda en línea se dedica a la venta de una amplia gama de productos de moda, y su visión es proporcionar a sus clientes una experiencia de compra personalizada y eficiente.

La base de datos planificada permitirá la recopilación, el almacenamiento y la gestión eficaz de datos críticos relacionados con clientes, productos, compras, proveedores y repartidores. Con esta base de datos, la empresa busca mejorar la eficiencia operativa, ofrecer un servicio más personalizado y tomar decisiones informadas para su crecimiento y éxito continuo en el competitivo mercado de la moda en línea.

Objetivos:

Los objetivos principales de esta base de datos son los siguientes:

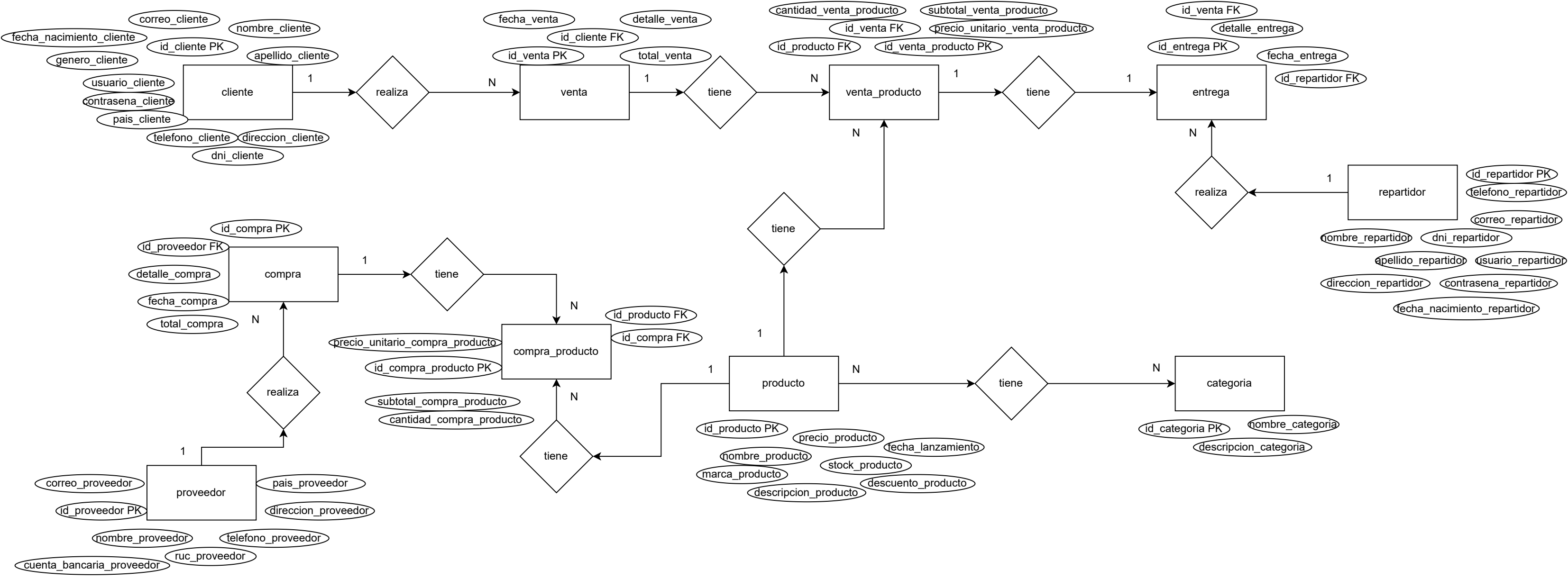
- Optimización de la gestión de clientes: La base de datos permitirá el registro detallado de información de clientes, incluyendo datos de registro, historial de compras y preferencias. Esto facilitará la personalización de la experiencia del cliente y la segmentación para promociones específicas.
- Control de inventario eficiente: La gestión de productos se verá mejorada mediante el seguimiento de datos clave, como nombre, precio, stock y marca. Esta información permitirá estimar las compras semanales, garantizando una oferta constante de productos populares.
- Relaciones con proveedores mejoradas: La base de datos registrará información de proveedores y compras realizadas. Esto proporcionará una visión clara de las ganancias, facilitará la planificación de adquisiciones y mejorará la toma de decisiones en cuanto a qué proveedor seleccionar para productos específicos.
- Seguimiento de entregas y logística eficiente: La base de datos permitirá el registro de repartidores, entregas, direcciones de clientes asociadas y detalles de las entregas. Esto garantizará un seguimiento efectivo de las órdenes y permitirá gestionar la logística de manera más eficiente.

Situación Problemática:

La empresa de venta de ropa en línea enfrenta varios desafíos:

Desafío: La falta de una base de datos centralizada y efectiva dificulta la personalización de la experiencia del cliente y la gestión eficiente del inventario. Además, la falta de seguimiento de proveedores y repartidores afecta la eficiencia operativa y la toma de decisiones.

Solución: La implementación de esta base de datos resolverá estos problemas al permitir la recopilación y el almacenamiento de datos cruciales, mejorando la experiencia del cliente, la gestión del inventario, las relaciones con proveedores y la logística de entregas. Esto permitirá a la empresa superar estos desafíos y crecer de manera más efectiva en el mercado de la moda en línea.



cliente				
Nombre	Abv.	Tipo de dato	Clave	Otros
id_cliente	id	int	pk	
nombre_cliente	nombre	varchar(50)		not null
apellido_cliente	apellido	varchar(50)		not null
telefono_cliente	tel	varchar(20)		not null
dni_cliente	dni	varhcar(20)		not null
direccion_cliente	direccion	varchar(100)		not null
usuario_cliente	usuario	varchar(100)		not null, unique
contrasena_cliente	contrasena	varchar(100)		not null,unique
pais_cliente	pais	varchar(50)		not null
correo_cliente	correo	varchar(50)		not null
fecha_nacimiento_cliente	nacimiento	date		not null
genero_cliente	genero	varchar(10)		not null

Se almacenan datos sobre el cliente, nombre, apellido y el contacto del cliente

producto				
Nombre	Abv.	Tipo de dato	Clave	Otros
id_producto	id	int	pk	
nombre_producto	nombre	varchar(100)		not null
precio_producto	precio	decimal(10,2)		not null
stock_producto	stock	int		not null
descripcion_producto	descripcion	text		
marca_producto	marca	varchar(50)		not null
descuento_producto	descuento	decimal(5,2)		
fecha_lanzamiento	lanzamiento	date		

Se almacenan datos de cada producto, precio, nombre y la cantidad del producto que se tiene en la tienda, stock.

categoria_producto				
Nombre	Abv.	Tipo de dato	Clave	Otros
id_categoria_producto	id	int	pk	
id_producto	producto	int	fk	
id_categoria	categoria	int	fk	

tabla que sirve para la relacion de cada producto con su respectiva categoria

categoria				
Nombre	Abv.	Tipo de dato	Clave	Otros
id_categoria	id	int	pk	
nombre_categoria	nombre	varchar(100)		not null
descripcion_categoria	descripcio	text		

en esta tabla se guarda la categoria de cada producto, con la finalidad de poder tener organizado cada producto

venta				
Nombre	Abv.	Tipo de dato	Clave	Otros
id_venta	id	int	pk	
id_cliente	cliente	int	fk	

total_venta	total	decimal(5,2)		not null
fecha_venta	fecha	date		not null
detalle_venta	descripcion	text		

Tabla en la que se almacena los datos de cada venta, total que se vendio y la cantidad de productos que se vendio

venta_producto				
Nombre	Abv.	Tipo de dato	Clave	Otros
id_venta_producto	id	int	pk	
id_venta	venta	int	fk	
id_producto	producto	int	fk	
precio_unitario_venta_producto	unidad	decimal(5,2)		not null
subtotal_venta_producto	subtotal	decimal(5,2)		not null
cantidad_venta_producto	cantidad	int		not null

tabla que une los productos con la tabla venta, almacena la cantidad y subtotal de cada producto, con respecto a la venta que se realizo

proveedor				
Nombre	Abv.	Tipo de dato	Clave	Otros
id_proveedor	id	int	pk	
nombre_proveedor	nombre	varchar(50)		not null
correo_proveedor	correo	varchar(50)		not null
telefono_proveedor	tel	varchar(20)		not null
direccion_proveedor	direccion	varchar(100)		not null
pais_proveedor	pais	varchar(50)		not null
ruc_proveedor	ruc	varchar(20)		not null
cuenta_bancaria_proveedor	cuenta	varchar(20)		not null

tabla que almacenara los datos de los proveedores, y asi obtener una mejor organización al comprar los productos a vender

compra				
Nombre	Abv.	Tipo de dato	Clave	Otros
id_compra	compra	int	pk	
id_proveedor	proveedor	int	fk	
fecha_compra	fecha	date		not null
detalle_compra	detalle	text		
total_compra	total	decimal(5,2)		not null

Se almacenara datos de la compra, los productos que se compraron, la fecha y el proveedor al que se le compro.

compra_producto				
Nombre	Abv.	Tipo de dato	Clave	Otros
id_compra_producto	id	int	pk	
id_compra	compra	int	fk	
id_producto	producto	int	fk	
cantidad_compra_producto	cantidad	int		not null
precio_unitario_compra_producto	unidad	decimal(5,2)		not null
subtotal_compra_producto	subtotal	decimal(5,2)		not null

tabla que une los productos con la tabla compra, almacena la cantidad y subtotal de cada producto, con respecto a la compra que se realizo

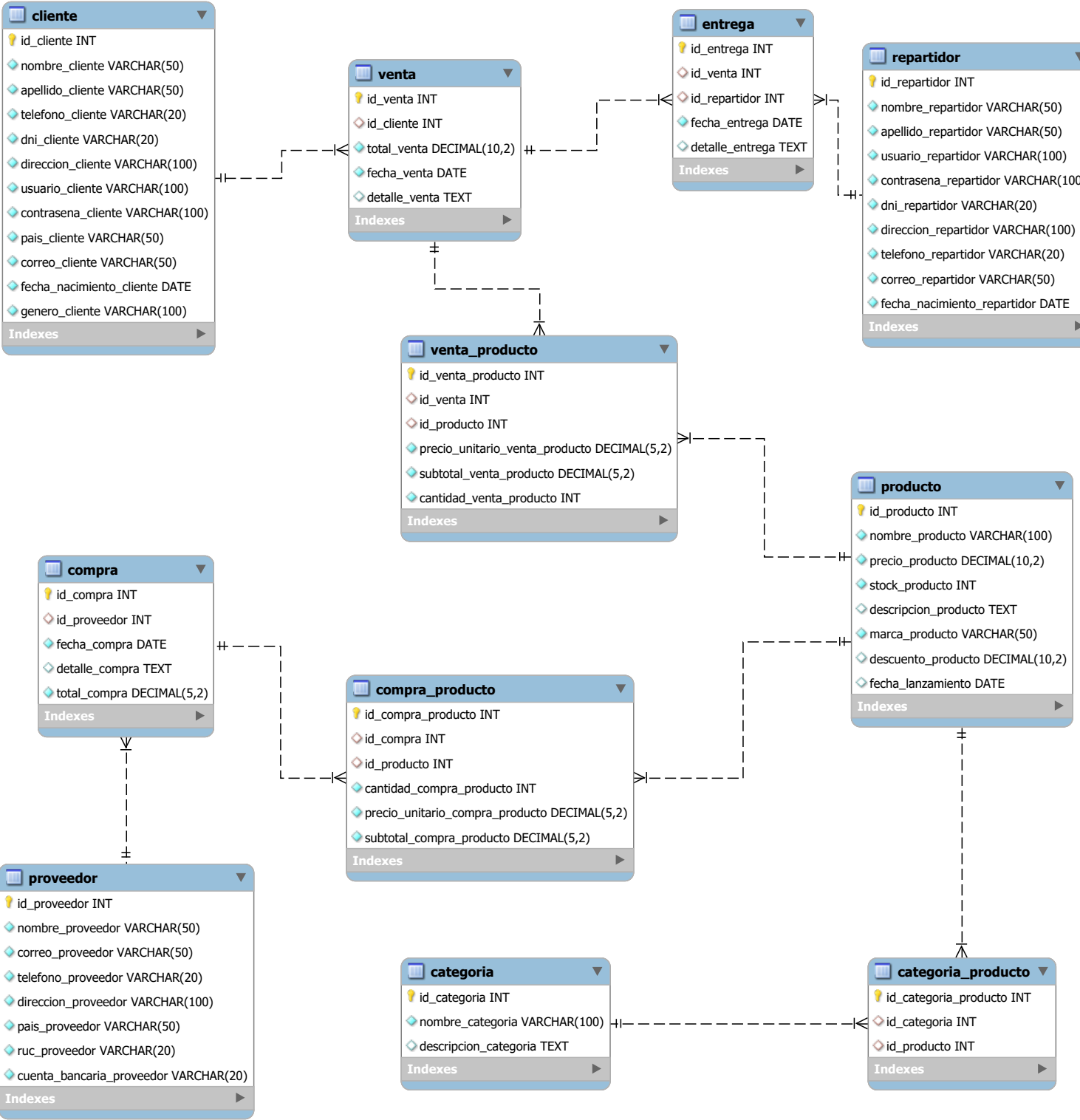
repartidor				
------------	--	--	--	--

Nombre	Abv.	Tipo de dato	Clave	Otros
id_repartidor	id	int	pk	
nombre_repartidor	nombre	varchar(50)		not null
apellido_repartidor	apellido	varchar(50)		not null
usuario_repartidor	usuario	varchar(100)		not null, unique
contrasena_repartidor	contrasena	varchar(100)		not null, unique
dni_repartidor	dni	varhcar(20)		not null
direccion_repartidor	direccion	varchar(100)		not null
telefono_repartidor	tel	varchar(20)		not null
correo_repartidor	correo	varchar(50)		not null
fecha_nacimiento_repartidor	nacimiento	date		not null

tabla en la cual se guarda la informacion de cada repartidor, su nombre, dni, direccion,etc.

entrega				
Nombre	Abv.	Tipo de dato	Clave	Otros
id_entrega	id	int	pk	
id_venta	venta	int	fk	
id_repartidor	repartidor	int	fk	
fecha_entrega	fecha	date		not null
detalle_entrega	detalle	text		

se almacenaran los datos de cada entrega que se realice, y el repartidor que lo realizo, con el fin de tener un seguimiento de las entregas



Vistas:

1.-Vista entregas_ultimo_mes:

Esta vista te permite visualizar las últimas entregas realizadas por cada repartidor en el último mes. Tiene como objetivo obtener un registro de las entregas realizadas por los repartidores en el último mes para el seguimiento y control de las entregas. Esta vista combina datos de las tablas entrega y repartidor.

2.-Vista compras_ultimo_mes:

Esta vista muestra las compras realizadas en el último mes, incluyendo el proveedor, la cantidad de productos comprados y el total de compra. Tiene como objetivo obtener un resumen de las compras realizadas en el último mes para el análisis y la gestión de inventario. Esta vista utiliza datos de las tablas compra, proveedor, y compra_producto.

3.-Vista info_producto:

Esta vista proporciona información sobre el stock de todos los productos, incluyendo la categoría a la que pertenecen y la fecha de lanzamiento. Tiene como objetivo ofrecer una visión general de la información de los productos, incluyendo su categoría y fecha de lanzamiento. Esta vista combina datos de las tablas producto, categoria_producto, y categoria.

4.-Vista ventas_ultimo_mes:

Esta vista muestra las ventas realizadas en el último mes, incluyendo el nombre del cliente, su usuario, la cantidad de productos vendidos y el total de venta. Tiene como objetivo obtener un resumen de las ventas realizadas en el último mes para el análisis de ventas y el seguimiento de clientes. Esta vista utiliza datos de las tablas venta, cliente, y venta_producto.

5.-Vista stock_bajo_producto:

Esta vista presenta productos que tienen un stock inferior a 10 unidades, junto con información sobre los proveedores que venden dichos productos. Tiene como objetivo identificar productos con bajo stock para tomar decisiones sobre reabastecimiento y mantener una comunicación con los proveedores correspondientes. Esta vista combina datos de las tablas producto, compra_producto, compra, proveedor, categoria_producto, y categoria.

Funciones:

1.-Función total_compras_cliente:

Esta función calcula el total de dinero gastado y la cantidad de compras realizadas por un cliente dado su número de identificación (DNI). Tiene como objetivo proporcionar a los usuarios la capacidad de obtener información detallada sobre las compras de un cliente específico en función de su DNI. Esta función interactúa con la tabla cliente y la tabla venta para calcular el total de compras y la cantidad de compras realizadas por el cliente con el DNI proporcionado.

2.-Función total_ventas_fecha_determinada:

Esta función calcula el monto total de ventas y la cantidad total de ventas dentro de un rango de fechas determinado. Tiene como objetivo permitir a los usuarios obtener información resumida sobre las ventas realizadas en un período específico. Esta función interactúa con la tabla venta para calcular el monto total de ventas y la cantidad total de ventas dentro del rango de fechas especificado.

Stored Procedure:

1.-Stored Procedure total_venta_dia:

Este procedimiento calcula el total de ventas realizadas en un día específico. Tiene como objetivo permitir a los usuarios obtener rápidamente el total de ventas para un día en particular, lo que puede ser útil para el seguimiento y análisis de las ventas diarias. Interactúa con la tabla venta para recuperar las ventas realizadas en el día especificado.

2.-Stored Procedure actualizar_stock_venta:

Este procedimiento actualiza el stock de productos después de realizar una venta, reduciendo la cantidad de productos disponibles en el inventario. Tiene como objetivo automatizar la actualización del inventario de productos después de cada venta, lo que garantiza que el stock refleje con precisión las existencias disponibles. Interactúa con las tablas venta, venta_producto y producto para actualizar el stock de productos.

3.-Stored Procedure ordenar_tabla:

Este procedimiento ordena una tabla específica según un campo de ordenamiento y un criterio de orden (ascendente o descendente). Tiene como objetivo facilitar la recuperación de datos ordenados de manera específica según las necesidades del usuario, lo que mejora la capacidad de análisis de datos. Este procedimiento puede utilizarse con cualquier tabla en la base de datos, ya que toma el nombre de la tabla como parámetro. Por lo tanto, no está directamente vinculado a una tabla específica, pero puede ser aplicado a varias tablas según sea necesario.

Informe 1: Informe de ventas por cliente

Este informe mostrará un resumen de las ventas realizadas por cada cliente, incluyendo el número total de ventas y el monto total gastado. Puedes utilizar la siguiente consulta SQL para generar este informe:

```
SELECT c.nombre_cliente, COUNT(v.id_venta) AS total_ventas, SUM(v.total_venta) AS monto_total_gastado
FROM cliente c
LEFT JOIN venta v ON c.id_cliente = v.id_cliente
GROUP BY c.nombre_cliente;
```

Informe 2: Top 5 de productos más vendidos

Este informe mostrará los cinco productos más vendidos junto con la cantidad total vendida de cada uno. Puedes utilizar la siguiente consulta SQL para generar este informe:


```
SELECT p.nombre_producto, SUM(vp.cantidad_venta_producto) AS cantidad_total_vendida
FROM producto p
LEFT JOIN venta_producto vp ON p.id_producto = vp.id_producto
GROUP BY p.nombre_producto
ORDER BY cantidad_total_vendida DESC
LIMIT 5;
```

Herramientas y Tecnologías Utilizadas:

Microsoft Excel: Utilizado para organizar y mostrar las tablas de datos.

Draw.io: Utilizado para crear el diagrama de entidad-relación (ERD).

SQL (Structured Query Language): Lenguaje utilizado para definir la estructura de la base de datos y realizar consultas.