

En la entrega final de proyecto se encontrará el desarrollo de la Problemática, Diagrama de Entidad – Relación, Listado de Tablas, Vistas, Funciones, Stored Procedures y Triggers del Trabajo Integral.

Proyecto Final - SQL.

Nicolás Rosellini

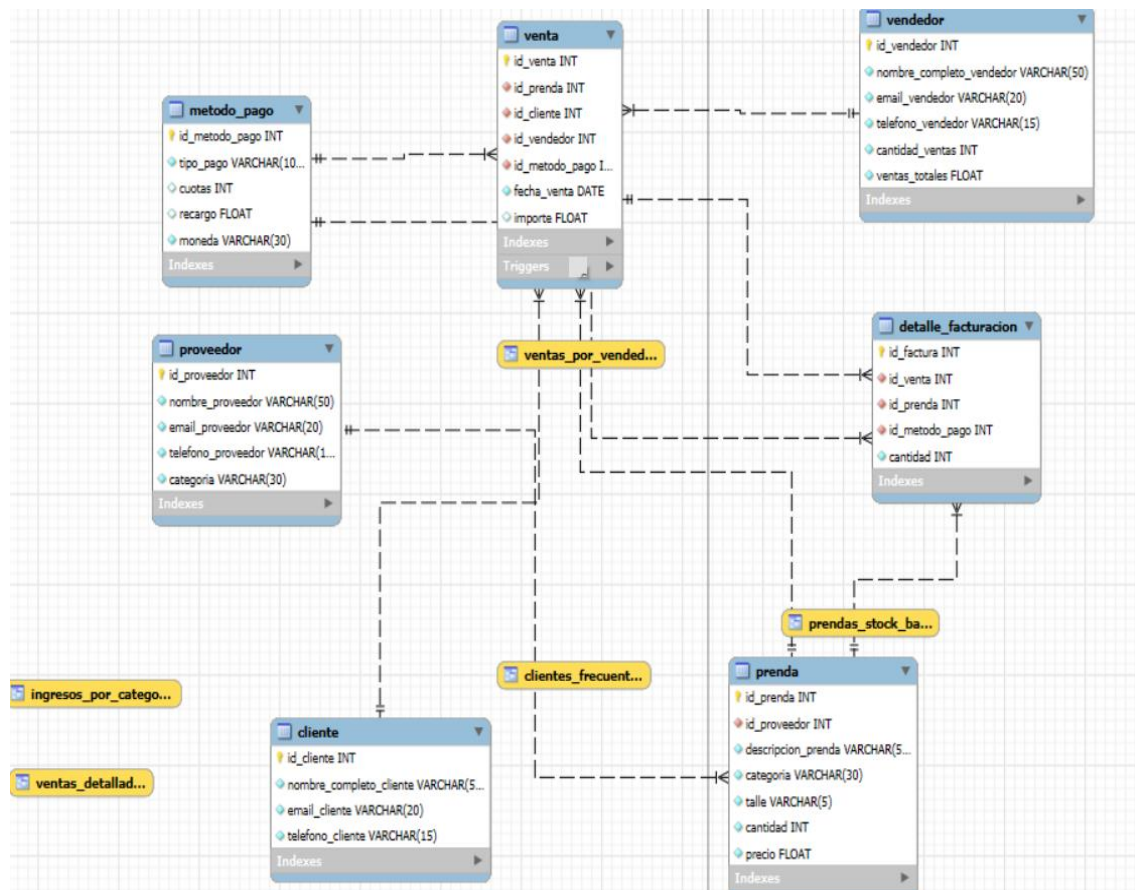
Problemática

La idea de este trabajo es lograr resolver un problema muy común que suelen tener los negocios de todo tipo, pero en este caso es de venta de ropa ya sea de manera online como física.

Lo que busca resolver este trabajo es encontrar analizar, predecir y tomar decisiones a través de consultas en SQL sobre quiénes son sus potenciales clientes, como administran sus compras con los proveedores que tienen y cuales son potenciales a ser considerados los proveedores primarios; cuales son los días que más venden, bajo que modalidad, a quienes, con que promociones, quienes son sus mejores vendedores, entre otros.

En resumen, analizar el entorno a través de las tablas creadas y de esta manera poder predecir comportamientos futuros como así también la toma de decisiones para elegir el camino a recorrer.

Diagrama de Entidad – Relación



Listado de Tablas

Tabla Prendas

Campo	Tipo Dato	PK	FK	NULL
id_prenda	INT	X		
Id_proveedor	INT		X	
descripcion_prenda	VARCHAR (50)			
categoria	VARCHAR (30)			
talle	VARCHAR (5)			
cantidad	INT			
precio	FLOAT			

Tabla Clientes

Campo	Tipo Dato	PK	FK	NULL
id_cliente	INT	X		
Nombre_completo_cliente	VARCHAR (50)		X	
email_cliente	VARCHAR (20)			
telefono_cliente	VARCHAR (15)			

Tabla Proveedores

Campo	Tipo Dato	PK	FK	NULL
id_proveedor	INT	X		
nombre_proveedor	VARCHAR (50)		X	
email_proveedor	VARCHAR (20)			
telefono_proveedor	VARCHAR (15)			
categoria	VARCHAR (30)			

Tabla Vendedores

Campo	Tipo Dato	PK	FK	NULL
id_vendedor	INT	X		
nombre_completo_vendedor	VARCHAR (50)		X	
email_vendedor	VARCHAR (20)			
telefono_vendedor	VARCHAR (15)			
cantidad_ventas	INT			
ventas_totales	FLOAT			

Tabla Métodos de Pago

Campo	Tipo Dato	PK	FK	NULL
id_metodo_pago	INT	X		
tipo_pago	VARCHAR (100)		X	
cuotas	INT			
recargo	FLOAT			
moneda	VARCHAR (30)			

Tabla Ventas

Campo	Tipo Dato	PK	FK	NULL
id_venta	INT	X		
id_prenda	INT		X	
id_cliente	INT		X	
id_vendedor	INT		X	
id_metodo_pago	INT		X	
fecha_venta	DATE			
importe	FLOAT			

Tabla Detalle Facturación

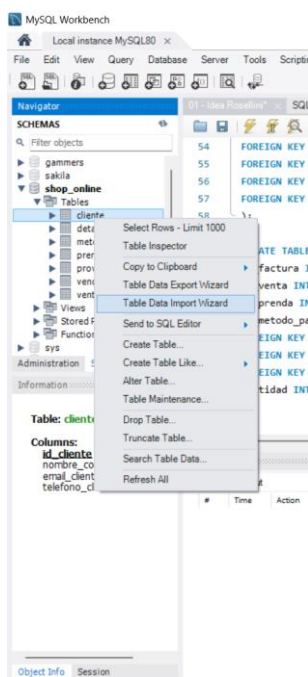
Campo	Tipo Dato	PK	FK	NULL
id_factura	INT	X		
id_venta	INT		X	
id_prenda	INT		X	
id_cliente	INT		X	
id_vendedor	INT		X	
id_metodo_pago	DATE		X	
cantidad	FLOAT			

Inserción de datos en tablas

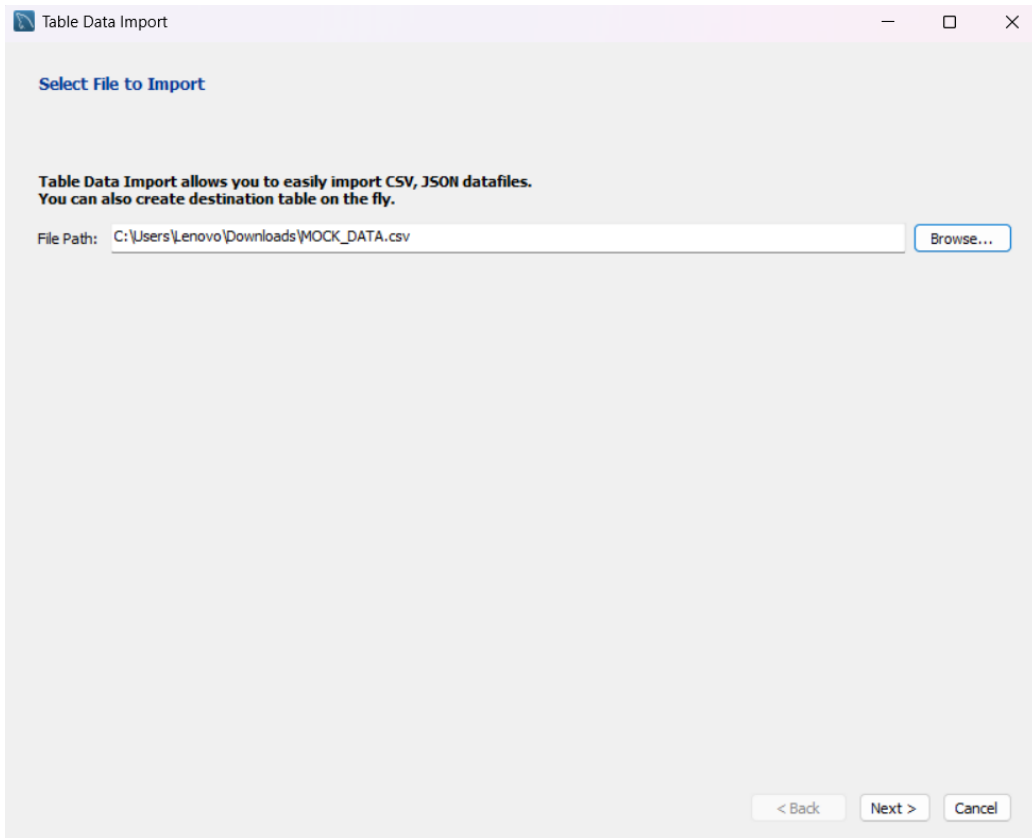
Paso 1: Creación de tablas en Mockaroo (Generador de datos randoms)

Paso 2: Descarga de archivo.

Paso 3: Table Data Import Wizard.

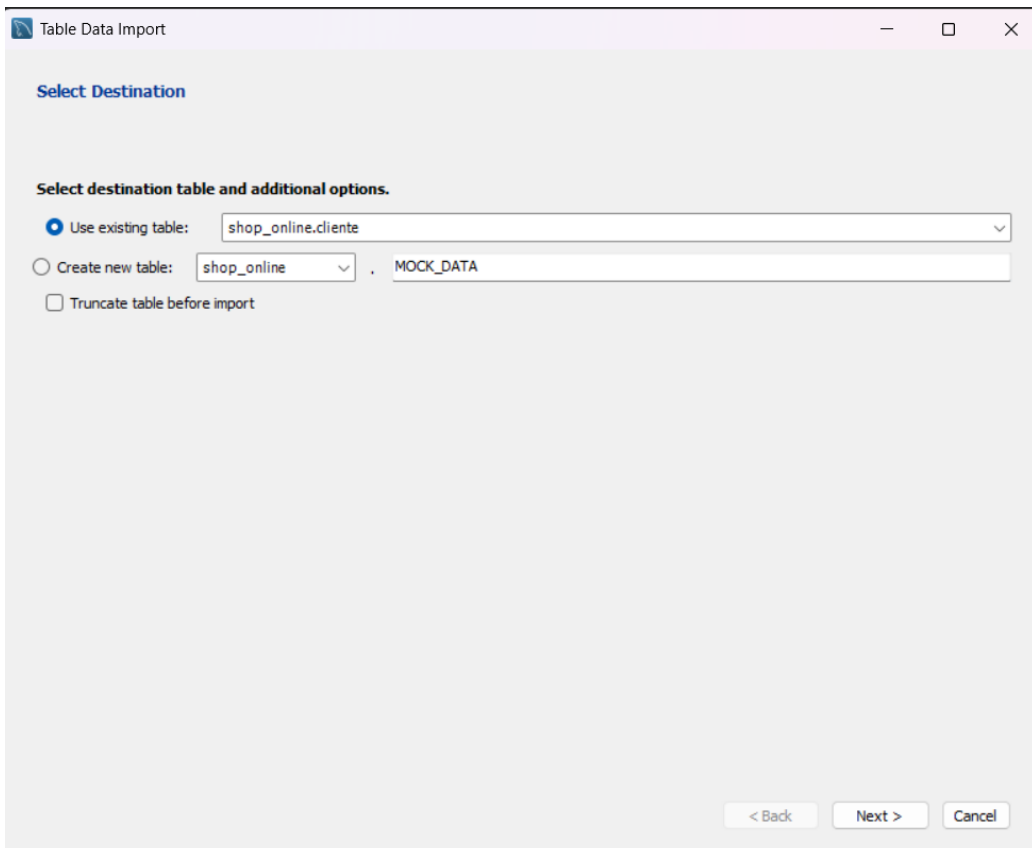


Paso 4: Buscar y cargar archivo descargado.



The screenshot shows the 'Table Data Import' dialog box with the title bar 'Table Data Import'. The main heading is 'Select File to Import'. Below this, a text box explains: 'Table Data Import allows you to easily import CSV, JSON datafiles. You can also create destination table on the fly.' A 'File Path:' label is followed by a text input field containing 'C:\Users\Lenovo\Downloads\MOCK_DATA.csv' and a 'Browse...' button. At the bottom right, there are three buttons: '< Back', 'Next >', and 'Cancel'.

Paso 5: Usar tabla existente




The screenshot shows the 'Table Data Import' dialog box with the title bar 'Table Data Import'. The main heading is 'Select Destination'. Below this, a text box explains: 'Select destination table and additional options.' There are three options: 'Use existing table:' (selected with a radio button) with a dropdown menu showing 'shop_online.cliente'; 'Create new table:' (unselected with a radio button) with a dropdown menu showing 'shop_online' and a text input field containing 'MOCK_DATA'; and 'Truncate table before import' (unselected with a checkbox). At the bottom right, there are three buttons: '< Back', 'Next >', and 'Cancel'.

Paso 6: Matchear los títulos de las columnas.

Table Data Import

Configure Import Settings

Detected file format: csv 

Encoding:

Columns:

<input checked="" type="checkbox"/> Source Column	Dest Column
<input checked="" type="checkbox"/> id	<input type="text" value="id_cliente"/>
<input checked="" type="checkbox"/> first_name	<input type="text" value="nombre_completo_client"/>
<input checked="" type="checkbox"/> last_name	<input type="text" value="email_cliente"/>
<input checked="" type="checkbox"/> email	<input type="text" value="telefono_cliente"/>
<input checked="" type="checkbox"/> gender	<input type="text" value="id_cliente"/>
<input checked="" type="checkbox"/> ip_address	<input type="text" value="id_cliente"/>

id	first_name	last_name	email	gender	ip_address
1	Atlante	Werrilow	awerrilow0...	Female	30.56.202.92
2	Brendan	Puddefoot	bpuddefoot...	Male	127.80.199...
3	Koren	Darrington	kdarrington...	Female	46.100.82....
4	Chery	Knowlys	cknowlys3...	Female	17.183.196...
5	Caria	Voller	cvoller4@w...	Female	246.217.17...

< Back Next > Cancel

Paso 7: Importar los datos.

Table Data Import

Import Data

The following tasks will now be performed. Please monitor the execution.

☐ Prepare Import
☐ Import data file

Click [Next >] to execute.

Show Logs

< Back Next > Cancel

Paso 8: Verificar que se hayan cargado los datos en la tabla correspondiente a través de un `// SELECT * FROM nombre_tabla.`

Vistas

Vista 1: ventas_detalladas

- **Objetivo:** Centralizar información clave de las ventas.
- **Tablas:** venta, cliente, prenda, vendedor, metodo_pago.
- **Beneficio:** Facilita reportes y dashboards con datos de ventas completos.

Vista 2: prendas_stock_bajo

- **Objetivo:** Identificar prendas con stock bajo.
- **Tabla:** prenda.
- **Beneficio:** Permite anticipar reposición y evitar quiebres de stock.

Vista 3: ventas_por_vendedor

- **Objetivo:** Resumir desempeño por vendedor.
- **Tablas:** venta, vendedor.
- **Beneficio:** Útil para evaluaciones, bonificaciones y gestión del equipo comercial.

Vista 4: ingresos_por_categoria

- **Objetivo:** Ver ingresos agrupados por categoría de prenda.
- **Tablas:** venta, prenda.
- **Beneficio:** Ayuda a tomar decisiones sobre qué líneas de productos potenciar.

Vista 5: clientes_frecuentes

- **Objetivo:** Detectar clientes con alta recurrencia de compras.
- **Tablas:** cliente, venta.
- **Beneficio:** Facilita la segmentación para campañas o programas de fidelización.

Funciones

Función 1: calcular_precio_final

- **Objetivo:** Calcular el precio final aplicando un recargo.
- **Uso:** Se puede utilizar en reportes o cálculos dinámicos de importes.
- **Tablas relacionadas:** metodo_pago, prenda.

Función 2: obtener_nombre_cliente

- **Objetivo:** Obtener el nombre del cliente a partir de su ID.
- **Beneficio:** Simplifica llamadas en otros procesos o reportes.
- **Tabla:** cliente.

Stored Procedures

Stored Procedure 1: registrar_venta

- **Objetivo:** Registrar una venta de forma estructurada.
- **Tablas involucradas:** venta.
- **Beneficio:** Automatiza el proceso de carga desde formularios o apps.

Store Procedure 2: actualizar_stock_prenda

- **Objetivo:** Disminuir el stock tras registrar una venta.
- **Tabla:** prenda.
- **Beneficio:** Asegura la integridad del stock en tiempo real.

Triggers

Trigger 1: antes_de_insertar_venta

- **Objetivo:** Prevenir la venta de productos sin stock.
- **Tablas:** venta, prenda.
- **Beneficio:** Mantiene la integridad del negocio.

Trigger 2: despues_de_insertar_venta

- **Objetivo:** Descontar automáticamente el stock tras una venta.
- **Tabla:** prenda.
- **Beneficio:** Automatiza actualización del inventario.

Descripción de los Objetos de la Base de Datos

Vistas

VENTAS_DETALLADAS: Centraliza información clave de cada venta, combinando datos de cliente, prenda, vendedor, método de pago, fecha e importe. Facilita la creación de reportes y dashboards de ventas completas.

PRENDAS_STOCK_BAJO: Lista las prendas con cantidades menores a 10 unidades. Permite anticipar necesidades de reposición y evitar quiebres de stock.

VENTAS_POR_VENDEDOR: Resume el desempeño individual de cada vendedor, mostrando la cantidad de ventas y el total facturado. Útil para evaluaciones de desempeño y gestión de incentivos.

INGRESOS_POR_CATEGORIA: Muestra los ingresos totales agrupados por categoría de prenda. Ayuda en la toma de decisiones estratégicas sobre qué líneas de productos potenciar.

CLIENTES_FRECUENTES: Identifica clientes con dos o más compras registradas. Facilita campañas de fidelización y segmentación comercial.

Funciones

CALCULAR_PRECIO_FINAL: Devuelve el precio final de un producto aplicando un recargo en base a un porcentaje. Útil para calcular valores en reportes y proyecciones.

Tablas relacionadas: metodo_pago, prenda.

OBTENER_NOMBRE_CLIENTE: Retorna el nombre completo de un cliente a partir de su ID. Simplifica reportes y consultas en las que solo se cuenta con la clave primaria.

Tabla relacionada: cliente.

Stored Procedures

REGISTRAR_VENTA: Inserta un nuevo registro en la tabla venta de forma estructurada, evitando errores en la carga de datos desde aplicaciones o formularios.

ACTUALIZAR_STOCK_PRENDA: Reduce la cantidad disponible de una prenda tras una venta. Mantiene el inventario actualizado en tiempo real.

Triggers

CHECK_STOCK: Verifica la disponibilidad de stock antes de registrar una venta. Su objetivo es prevenir ventas de productos sin inventario, protegiendo la integridad del negocio.

DESCONTAR_STOCK: Disminuye automáticamente la cantidad de una prenda en 1 unidad después de una venta, evitando la actualización manual del inventario.