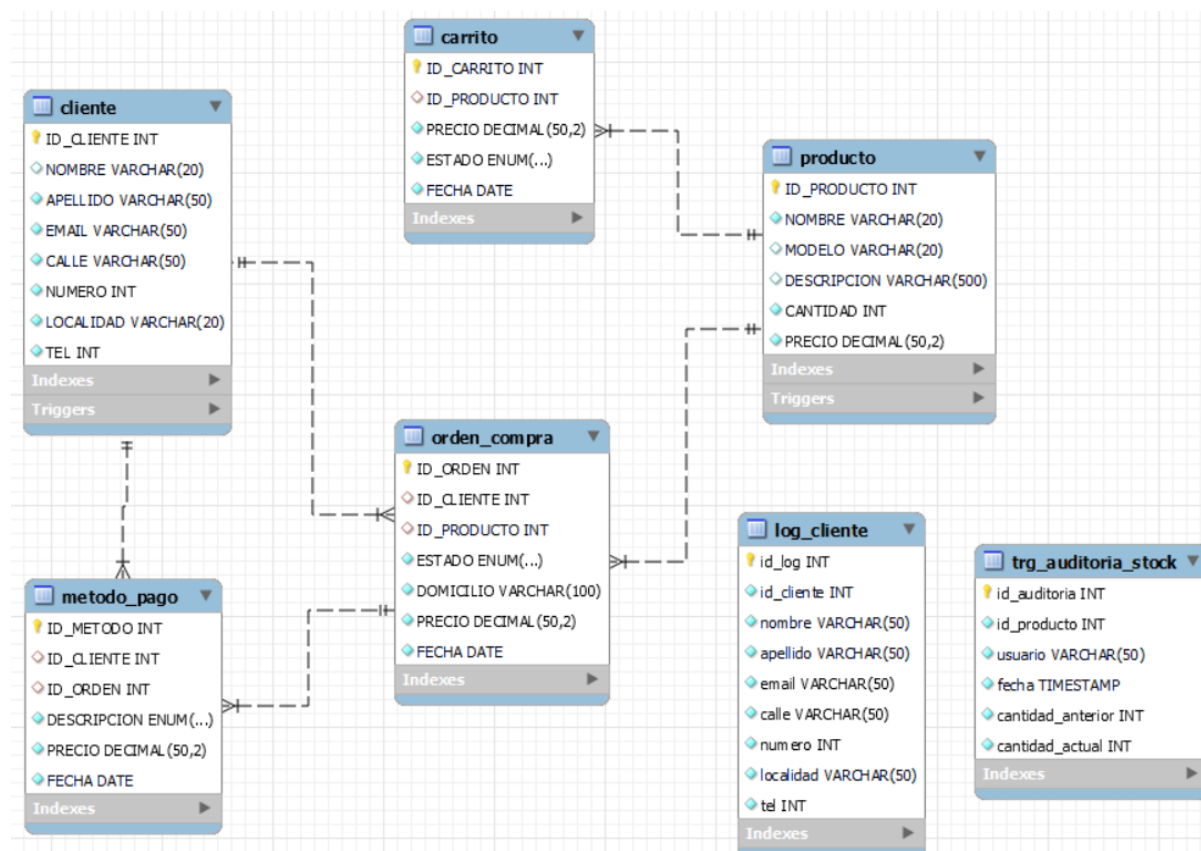


Informe para la base de datos de Ecommerce

1. [Diagrama de Entidad Relación](#)
2. [Introducción](#)
3. [Objetivo](#)
4. [Situación Problemática](#)
5. [Modelo de Negocio](#)
6. [Tablas](#)
7. [Vistas \(Views\)](#)
8. [Funciones \(Functions\)](#)
9. [Procedimientos Almacenados \(Stored Procedures\)](#)
10. [Disparadores \(Triggers\)](#)
11. [Lenguaje de Control de Datos \(Data Control Language - DCL\)](#)
12. [Lenguaje de Control de Transacciones \(Transaction Control Language - TCL\)](#)
13. [Herramientas](#)

Diagrama de Entidad Relación

Un Diagrama de Entidad-Relación es un esquema que muestra cómo las diferentes tablas de la base de datos están conectadas y relacionadas entre sí. Es una forma clara de visualizar cómo organizar y almacenar la información en una base de datos.



Introducción

El proyecto es realizado a partir de una base de datos de un ecommerce de electrodomésticos. Las tablas incluyen a los clientes registrados, productos en stock, carritos creados, órdenes de compra generadas y métodos de pago utilizados por los clientes.

Objetivo

La finalidad es por un lado que el usuario tenga registro de los productos que ya compró y por el otro que el e-commerce lleve un registro de los productos vendidos, cantidad en stock, variación de precios, etc.

Situación Problemática

A partir de las tablas creadas y la relación entre ellas pudimos resolver diferentes problemáticas tales como: compras realizadas por los clientes, registro de usuarios, visualizar de manera más ágil la variación del stock de los productos y productos faltantes. También pudimos observar y diferenciar los métodos de pagos más elegidos por los clientes, así como también aquellas órdenes de compra que se cancelan y no llegan a confirmarse.

Modelo de Negocio

La idea es a partir del análisis y almacenamiento de la información relevante de los clientes, como historial de compras, preferencias, método de pago, comportamiento de navegación y datos demográficos, hacer recomendaciones personalizadas para mejorar las ventas y la experiencia del cliente.

Tablas

Las tablas son objetos en las bases de datos que se utilizan para almacenar y administrar los datos. Se organizan en filas y en columnas y pueden relacionarse con otras tablas para obtener información específica. En este proyecto se utilizaron las siguientes tablas:

Tabla Cliente

Para la tabla cliente, se utilizaron los siguientes campos:

Id_cliente: Número único de identificación para cada cliente. Tipo de dato numérico.

Nombre: Nombre del cliente. Tipo de dato alfanumérico.

Apellido: Apellido del cliente. Tipo de dato alfanumérico.

Email: Correo electrónico del cliente. Tipo de dato alfanumérico.

Calle: Nombre de la calle donde vive el cliente. Tipo de dato alfanumérico.

Número: Número de casa o departamento del cliente en la calle mencionada. Tipo de dato numérico.

Localidad: Localidad o barrio donde vive el cliente. Tipo de dato alfanumérico.

Teléfono: Número de teléfono de contacto del cliente. Tipo de dato numérico.

Tabla Producto:

Para esta tabla se utilizaron los siguientes campos:

Id_producto: Número único de identificación para cada producto. Tipo de dato numérico.

Nombre: Nombre del producto. Tipo de dato alfanumérico.

Modelo: Nombre o código del modelo del producto. Tipo de dato alfanumérico.

Descripción: Breve descripción del producto. Tipo de dato alfanumérico.

Cantidad: Cantidad actual en stock. Tipo de dato numérico.

Precio: Precio del producto. Tipo de dato números con decimales.

Tabla Carrito:

Para esta tabla se utilizaron los siguientes campos:

Id_carrito: Número único de identificación para el carrito creado. Tipo de dato numérico.

Id_producto: Número único de identificación para cada producto (relacionado a la tabla de productos). Tipo de dato numérico.

Precio: Precio del producto. Tipo de dato números con decimales.

Estado: Estado actual del carrito (puede ser Completo, En Proceso, Cancelado). Tipo de dato enumeración (opciones ya establecidas para elegir).

Fecha: Fecha de creación del carrito. Tipo de dato fecha.

Tabla Orden de Compra:

Para la tabla de Orden de Compra se utilizaron los siguientes campos:

Id_orden: Número único de identificación para la orden de compra creada. Tipo de dato numérico.

Id_cliente: Número único de identificación para el cliente que genera la orden de compra (relacionado a la tabla Cliente). Tipo de dato alfanumérico.

Id_producto: Número único de identificación para cada producto (relacionado a la tabla de productos). Tipo de dato numérico.

Estado: Estado actual de la orden de compra (puede ser Completa, En Proceso, Cancelada). Tipo de dato enumeración (opciones ya establecidas para elegir).

Domicilio: Domicilio de entrega del producto. Tipo de dato alfanumérico.

Precio: Precio total del carrito y productos seleccionados. Tipo de dato números con decimales.

Fecha: Fecha de creación de la orden de compra. Tipo de dato fecha.

Tabla Método de Pago:

Para la tabla de Método de Pago se utilizaron los siguientes campos:

Id_metodo: Número único de identificación para el método de pago seleccionado por el cliente. Tipo de dato numérico.

Id_cliente: Número único de identificación para el cliente que genera la orden de compra (relacionado a la tabla Cliente). Tipo de dato alfanumérico.

Id_orden: Número único de identificación para la orden de compra creada (relacionado a la tabla Orden de Compra). Tipo de dato numérico.

Descripción: Tipo de método de pago elegido por el cliente (Débito, Crédito, Cancelada). Tipo de dato enumeración (opciones ya establecidas para elegir).

Precio: Precio total a pagar. Tipo de dato números con decimales.

Fecha: Fecha de pago. Tipo de dato fecha.

Vistas (views)

Es una consulta que crea una vista con datos de otras tablas de la base de datos. Esto permite a los usuarios acceder a información específica sin necesidad de ver toda la base de datos. Las vistas también ayudan a proteger datos privados al ocultarlos y hacen que el proceso de búsqueda y gestión de datos sea más sencillo y seguro.

En este proyecto se utilizaron las siguientes vistas:

ESTADO_OC_CANCEL: Vista para ver las ordenes de compras con estado "Cancelada".

CLIENTES_CABA: Ver los clientes que residen en CABA.

PRODUCTOS_FALTA: Ver los productos que estén por debajo de 30 unidades en stock

DEBITO_2023: Ver cuántos clientes eligieron "Débito" como método de pago en el 2023.

OC_COMPLETA_CREDITO_2023: Ver las órdenes de compra completadas y pagadas en crédito durante el 2023.

Funciones (functions)

Una función es como una rutina diseñada para tomar ciertos datos, procesarlos y devolver un resultado. Pueden utilizarse en una consulta y solo devuelven un único valor, no un conjunto de registros. Sirve para simplificar y reutilizar código, haciendo el trabajo más fácil y eficiente.

En este proyecto se utilizaron las siguientes funciones:

clientes_caba: ver la cantidad de clientes en CABA.

precio_final_aumento: calcular el precio final de un producto a partir de su id; en caso de querer hacer un aumento porcentual del precio.

total_cancelados_2023: función para ver el monto total de las órdenes de compra canceladas en 2023.

Procedimiento Almacenado (stored procedure)

Un "stored procedure" es como un pequeño programa guardado en la base de datos. Su función es realizar tareas específicas y complejas, como actualizaciones o consultas complicadas. Se usa para

simplificar el código y mejorar el rendimiento, ya que se ejecuta directamente en la base de datos y puede ser reutilizado en diferentes partes del sistema.

En este proyecto se utilizaron los siguientes “stored procedures”:

InsertarCliente: para agregar un cliente de manera manual a la base de datos.

oc_estados: obtener las Órdenes de Compra según el estado en el que se encuentre. puede ser Completa, En Proceso, Cancelada.

sp_ordenar_clientes: para ordenar la tabla clientes según el parámetro deseado y pudiendo elegir entre hacerlo de modo ascendente o descendente.

stock_productos: ver la cantidad de productos que tienen stock menor a “X”. La cantidad puede modificarse según la necesidad.

Disparadores (triggers)

Es una funcionalidad que la base de datos ejecuta de manera automática cuando se realiza una operación previamente indicada (inserciones, actualizaciones o eliminaciones de datos). Pueden ser utilizados para auditorías o control de registro de clientes nuevos, entre otros.

En este proyecto se utilizaron los siguientes “triggers”:

trg_auditoria_stock: disparador que audita cada vez que hay una variación en el stock de los productos.

trg_log_cliente: disparador que audita cada vez que se registra un nuevo cliente.

Lenguaje de Control de Datos (Data Control Language - DCL)

Se utiliza dentro de las bases de datos para crear roles y otorgar permisos, así como también el control al acceso a la base de datos. Se centra en el control de la seguridad y la autorización de usuarios para realizar diversas operaciones en las tablas y bases de datos.

En este proyecto se crearon dos usuarios a modo de prueba, cada uno con distintos accesos y permisos sobre las tablas.

Lenguaje de Control de Transacciones (Transaction Control Language - TCL)

Es un conjunto de comandos que se usan para manejar y asegurar las transacciones en la base de datos. Las transacciones son un grupo de operaciones que se ejecutan juntas: se puede confirmar (commit) o deshacer (rollback) estas transacciones. Su objetivo es garantizar que las transacciones en la base de datos se realicen de manera adecuada, permitiendo confirmar o deshacer los cambios realizados en forma de grupos de operaciones.

En este proyecto se utilizaron dos transacciones a modo de prueba; el primero para deshacer la acción (rollback) y/o confirmarla (commit) luego de la inserción de un cliente nuevo. El segundo para deshacer la acción (rollback) y/o confirmarla (commit) después de la inserción de dos grupos de productos en su respectiva tabla.

Herramientas

Para la realización de este proyecto se utilizaron MYSQL Workbench y GitHub.