# Proyecto DB SQL

## Introducción

### Descripción de la Situación del Negocio

En el núcleo de la creación de nuestro marketplace yace la esencial tarea de diseñar y construir una base de datos robusta que permitirá respaldar la gestión eficiente de productos, la personalización de la experiencia del usuario, la seguridad de datos y un rendimiento óptimo. En este proceso, nos embarcamos en la tarea de analizar requisitos específicos, diseñar la estructura, implementar con un enfoque escalable. Con cada paso, nuestro objetivo es forjar una base de datos que no solo cumpla con las necesidades actuales de nuestro marketplace, sino que también siente las bases para su crecimiento sostenible y la entrega de una experiencia excepcional a nuestros usuarios y socios comerciales.

### Objetivo de la Base de Datos

* Almacenar información detallada sobre cada producto, como nombre, descripción, precio, imágenes, categoría, atributos específicos, etc.
* Mantener un seguimiento actualizado del inventario para evitar problemas de sobreventa o falta de existencias.
* Almacenar datos de usuarios registrados, incluyendo nombres, direcciones, información de contacto y preferencias.
* Registrar información financiera relacionada con las transacciones, como precios, impuestos, descuentos, métodos de pago y estado de pago.

### Problemas a resolver

1. Gestión de Inventario: Controlar y actualizar en tiempo real el inventario de productos. La base de datos permitirá almacenar información sobre la disponibilidad de productos, evitando problemas como la sobreventa o la falta de existencias.
2. Seguimiento de Transacciones: Gestionar y realizar un seguimiento de las transacciones de compra. La base de datos permitirá almacenar datos de transacciones, incluyendo detalles financieros y estados de pago, para facilitar el seguimiento y la resolución de problemas relacionados con los pedidos.
3. Personalización de Experiencia: Ofrecer una experiencia personalizada a los usuarios. La base de datos permitirá almacenar historiales de compras y preferencias de usuarios, lo que permite personalizar recomendaciones y promociones.
4. Eficiencia en Búsquedas y Filtrados: Realizar búsquedas y filtrados eficientes en un catálogo extenso. La base de datos permitirá utilizar índices y consultas eficientes en la base de datos para mejorar la velocidad de recuperación de datos y proporcionar una experiencia de usuario fluida.
5. Seguridad y Privacidad: Garantizar la seguridad y privacidad de los datos almacenados. La base de datos permitirá implementar medidas de seguridad en la base de datos para proteger la integridad y confidencialidad de la información, así como gestionar los permisos de acceso de los usuarios.
6. Escalabilidad: Manejar el aumento en el volumen de datos y transacciones. La base de datos permitirá diseñar la base de datos de manera escalable para adaptarse al crecimiento del marketplace sin comprometer el rendimiento.

## Entidades y relaciones

### Diagrama Entidad-Relación

### Descripción de tablas

#### Tabla: usuarios

Esta tabla almacena información sobre los usuarios registrados en el marketplace.

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo de Clave | Nombre del Campo | Tipo de Datos | Descripción |
| PK | id\_usuario | INT | Identificador único del usuario. |
|  | Tipo\_documento | VARCHAR (45) | Tipo de documento de identidad. |
|  | dni | VARCHAR (45) | Número documento de identidad. |
|  | nombre | VARCHAR (45) | Nombre completo del usuario. |
|  | apellido | VARCHAR (45) | Apellido completo del usuario. |
|  | correo | VARCHAR (80) | Correo electrónico del usuario |
|  | contraseña | VARCHAR (45) | Contraseña del usuario. |
|  | telefono | VARCHAR (45) | Teléfono del usuario. |
|  | Fecha\_nacimiento | VARCHAR (80) | Fecha de nacimiento. |

#### Tabla: historial\_compras

Esta tabla almacena las compras realizadas de los usuarios.

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo de Clave | Nombre del Campo | Tipo de Datos | Descripción |
| PK | id\_historial | INT | Identificador único del historial de compras |
| FK | id\_usuario | INT | ID del usuario asociado. |
|  | fecha | TIMESTAMP | Calle y número de la dirección. |
|  | estado | VARCHAR (20) | Ciudad de la dirección. |

#### Tabla: detalle\_historial

Esta tabla almacena información sobre los usuarios registrados en el marketplace.

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo de Clave | Nombre del Campo | Tipo de Datos | Descripción |
| PK | id\_detalle\_historial | INT | Identificador único del detalle del historial. |
| FK | id\_historial | INT | ID del historial de compras asociado. |
| FK | id\_usuario | INT | ID del usuario asociado. |
| FK | id\_producto | INT | ID del producto asociado. |
| FK | id\_categoria | INT | ID de la categoría del producto asociado. |
|  | cantidad | INT | Teléfono del usuario. |
|  | precio | DECIMAL (10, 2) | Precio de cada producto. |
|  | comentario | VARCHAR (45) | Comentarios detalles del historial. |

#### Tabla: Productos

Esta tabla almacena información sobre los productos disponibles en el marketplace.

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo de Clave | Nombre del Campo | Tipo de Datos | Descripción |
| PK | id\_producto | INT | Identificador único del producto. |
| FK | id\_categoria | INT | ID de la categoría del producto asociado. |
| FK | id\_proveedor | INT | ID del proveedor asociado. |
|  | nombre | VARCHAR (45) | Nombre del producto |
|  | descripcion | VARCHAR (45) | Descripción detalla del producto. |
|  | precio | DECIMAL (10, 2) | Precio de cada producto. |
|  | Cantidad\_stock | INT | Cantidad en stock de cada producto. |

#### Tabla: Categorias

Esta tabla almacena información sobre los productos disponibles en el marketplace.

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo de Clave | Nombre del Campo | Tipo de Datos | Descripción |
| PK | id\_categoria | INT | Identificador único de la categoría del producto. |
|  | nombre | INT | Nombre de la categoría del producto. |
|  | descripcion | INT | Descripción de la categoría. |
|  | fecha\_creacion | TIMESTAMP | Fecha de Creación de una nueva categoría. |

#### Tabla: Comentarios

Esta tabla almacena información sobre los comentarios de cada usuario por cada producto.

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo de Clave | Nombre del Campo | Tipo de Datos | Descripción |
| PK | id\_comentario | INT | Identificador único del comentario. |
| FK | id\_producto | INT | ID del producto asociado. |
| FK | id\_categoria | INT | ID de la categoría asociada. |
|  | comentario | VARCHAR (45) | Contenido real del comentario. |
|  | fecha | TIMESTAMP | Tiempo que se hizo el comentario. |
|  | estado | VARCHAR (45) | Estado del comentario. |
|  | calificacion | VARCHAR (45) | Calificación o puntaje de los comentarios. |

#### Tabla: Promociones

Esta tabla almacena información sobre las promociones o descuentos que la empresa ofrece a sus clientes.

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo de Clave | Nombre del Campo | Tipo de Datos | Descripción |
| PK | id\_promocion | INT | Identificador único de la promoción. |
| FK | id\_producto | INT | ID del producto asociado. |
|  | nombre | INT | Nombre de la promoción. |
|  | descuento | VARCHAR (45) | Descuento aplicado a la promoción. |
|  | fecha\_Inicio | VARCHAR (45) | Fecha inicial de la promoción. |
|  | fecha\_Fin | DECIMAL (10, 2) | Fecha máxima de la promoción. |

#### Tabla: Proveedor

Esta tabla almacena información de los proveedores fuente importante en el suministro de productos y servicios.

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo de Clave | Nombre del Campo | Tipo de Datos | Descripción |
| PK | id\_proveedor | INT | Identificador único del proveedor. |
|  | nombre | VARCHAR (45) | Nombre del proveedor. |
|  | contacto | VARCHAR (45) | Número fijo o móvil del proveedor. |
|  | direccion | VARCHAR (45) | Ubicación de la empresa. |
|  | estado | VARCHAR (45) | Estado del proveedor: activo, inactivo, suspendido. |
|  | Info\_Financiera | VARCHAR (45) | Datos financieros relevantes. |
|  | tipo | VARCHAR (45) | Indica el tipo: bienes o servicios. |

#### Tabla: Envío

Esta tabla almacena información de los envíos de cualquier producto solicitado por el usuario.

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo de Clave | Nombre del Campo | Tipo de Datos | Descripción |
| PK | id\_envio | INT | Identificador único del envío. |
|  | direccion | VARCHAR (45) | Dirección donde será entregado el producto. |
|  | estado | VARCHAR (45) | Estado del envío: en tránsito, entregado, pendiente. |
|  | Fecha\_entrega | TIMESTAMP | Fecha de entrega del producto. |
|  | metodo | VARCHAR (45) | Método de envío: estándar, express, recogido en tienda. |
|  | courrier | VARCHAR (45) | Datos contacto empresa mensajera. |
|  | costo | DECIMAL (10,2) | Costo por el transporte. |
|  | tiempo\_entrega | TIMESTAMP | Tiempo en minutos tomará entrega del producto. |

#### Tabla: Carrito

Esta tabla almacena información de los envíos de cualquier producto solicitado por el usuario.

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo de Clave | Nombre del Campo | Tipo de Datos | Descripción |
| PK | id\_carrito | INT | Identificador único por cada carrito de compras. |
| KF | Id\_orden | INT | ID de la orden asociada. |
|  | fecha\_Entrega | TIMESTAMP | Fecha y hora que se creó el carrito. |

#### Tabla: Detalle Carrito

Esta tabla almacena información detalla del carrito de compras.

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo de Clave | Nombre del Campo | Tipo de Datos | Descripción |
| PK | id\_detalleCarrito | INT | Identificador único por cada artículo en el carrito. |
| FK | id\_carrito | INT | ID del usuario asociado. |
| FK | id\_orden | INT | ID del producto asociado. |
| FK | id\_producto | INT | ID del producto asociado. |
| FK | id\_categoria | INT | ID de la categoría asociada. |
| FK | id\_proveedor | INT | ID del proveedor asociado. |
|  | cantidad | INT | Cantidad de productos. |
|  | precio\_total | DECIMAL (10,2) | Precio total del producto. |

#### Tabla: Ordenes

Esta tabla almacena gestión de pedidos sobre cada transacción de compra.

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo de Clave | Nombre del Campo | Tipo de Datos | Descripción |
| PK | id\_orden | INT | Identificador único por cada artículo en el carrito. |
| FK | id\_promocion | INT | ID de la promoción asociada. |
|  | fecha | TIMESTAMP | Fecha de creación de la orden. |
|  | estado | VARCHAR (45) | Estado de la orden creada. |

#### Tabla: Detalle Orden

Esta tabla almacena información detallada de las transacciones de gestión de pedidos.

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo de Clave | Nombre del Campo | Tipo de Datos | Descripción |
| PK | id\_detalle | INT | Identificador único por cada artículo de la orden. |
| FK | id\_producto | INT | ID del producto asociado. |
| FK | id\_categoria | INT | ID de la categoría asociado. |
| FK | id\_usuario | INT | ID del usuario asociado. |
| FK | id\_envio | INT | ID del envío asociado. |
| FK | id\_promocion | INT | ID de la promoción asociada. |
| FK | Id\_orden | INT | ID del orden asociado. |
|  | cantidad | INT | Cantidad de unidades del producto incluido en la orden. |
|  | precio\_total | DECIMAL (10,2) | Costo total del artículo. |

## Scripts

### Creación de tablas

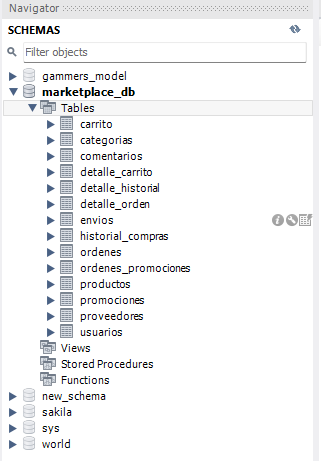
*-- Crear la base de datos del marketplace*  
CREATE DATABASE IF NOT EXISTS marketplace\_db;  
  
USE marketplace\_db;  
  
*-- Tabla de Usuarios*  
CREATE TABLE IF NOT EXISTS usuarios  
  (  
     id\_usuario       *INT* auto\_increment PRIMARY KEY,  
     tipo\_documento   *VARCHAR*(50) NOT NULL,  
     dni              *INT* NOT NULL UNIQUE,  
     nombre           *VARCHAR*(50) NOT NULL,  
     apellido         *VARCHAR*(255) NOT NULL,  
     correo           *VARCHAR*(255) NOT NULL UNIQUE,  
     contrasena       *VARCHAR*(50) NOT NULL,  
     telefono         *VARCHAR*(50),  
     fecha\_nacimiento *DATETIME*  
  );  
  
*-- Tabla de Historial de compras*  
CREATE TABLE IF NOT EXISTS historial\_compras  
  (  
     id\_historial *INT* auto\_increment PRIMARY KEY,  
     fecha        *DATETIME* NOT NULL,  
     estado       *VARCHAR*(255),  
     id\_usuario   *INT*,  
     FOREIGN KEY(id\_usuario) REFERENCES usuarios(id\_usuario)  
  );

*-- Tabla de Categorias*  
CREATE TABLE IF NOT EXISTS categorias  
  (  
     id\_categoria   *INT* auto\_increment PRIMARY KEY,  
     nombre         *VARCHAR*(45) NOT NULL,  
     descripcion    *VARCHAR*(255),  
     fecha\_creacion *DATETIME*  
  );  
  
*-- Tabla de Productos*  
CREATE TABLE IF NOT EXISTS productos  
  (  
     id\_producto *INT* auto\_increment PRIMARY KEY,  
     nombre      *VARCHAR*(255) NOT NULL,  
     description *VARCHAR*(255),  
     price       *DECIMAL*(10, 2) NOT NULL,  
     cantidad    *INT*  
  );  
  
*-- Tabla del Detalle del historial de compras*  
CREATE TABLE IF NOT EXISTS detalle\_historial  
  (  
     id\_detalle\_historial *INT* auto\_increment PRIMARY KEY,  
     price                *DECIMAL*(10, 2) NOT NULL,  
     cantidad             *INT* NOT NULL,  
     comentario           *VARCHAR*(255),  
     id\_historial         *INT*,  
     id\_usuario           *INT*,  
     id\_producto          *INT*,  
     id\_categoria         *INT*,  
     FOREIGN KEY(id\_historial) REFERENCES historial\_compras(id\_historial),  
     FOREIGN KEY(id\_usuario) REFERENCES usuarios(id\_usuario),  
     FOREIGN KEY(id\_categoria) REFERENCES categorias(id\_categoria),  
     FOREIGN KEY(id\_producto) REFERENCES productos(id\_producto)  );  
*-- Tabla de Proveedores*  
CREATE TABLE IF NOT EXISTS proveedores  
  (  
     id\_proveedor    *INT* auto\_increment PRIMARY KEY,  
     nombre          *VARCHAR*(45) NOT NULL,  
     contacto        *DECIMAL*(10, 2) NOT NULL,  
     direccion       *VARCHAR*(90),  
     estado          *VARCHAR*(45),  
     info\_financiera *VARCHAR*(255),  
     tipo            *VARCHAR*(45) NOT NULL  
  );  
  
*-- Tabla de Promociones*  
CREATE TABLE IF NOT EXISTS promociones  
  (  
     id\_promocion *INT* auto\_increment PRIMARY KEY,  
     nombre       *VARCHAR*(45) NOT NULL,  
     descuento    *DECIMAL*(10, 2) NOT NULL,  
     fecha\_inicio *DATETIME*,  
     fecha\_fin    *DATETIME*,  
     id\_producto  *INT*,  
     id\_categoria *INT*,  
     id\_proveedor *INT*,  
     FOREIGN KEY(id\_producto) REFERENCES productos(id\_producto),  
     FOREIGN KEY(id\_categoria) REFERENCES categorias(id\_categoria),  
     FOREIGN KEY(id\_proveedor) REFERENCES proveedores(id\_proveedor)  
  );

*-- Tabla de Ordenes*  
CREATE TABLE IF NOT EXISTS ordenes  
  (  
     id\_orden     *INT* auto\_increment PRIMARY KEY,  
     fecha        *DATETIME*,  
     estado       *VARCHAR*(255),  
     id\_promocion *INT*,  
     FOREIGN KEY(id\_promocion) REFERENCES promociones(id\_promocion)  
  );

*-- Tabla de Envíos*  
CREATE TABLE IF NOT EXISTS envios  
  (  
     id\_envio       *INT* auto\_increment PRIMARY KEY,  
     direccion      *VARCHAR*(50) NOT NULL,  
     estado         *VARCHAR*(50) NOT NULL,  
     fecha\_entrega  *DATETIME* NOT NULL,  
     metodo         *VARCHAR*(255) NOT NULL UNIQUE,  
     courrier       *VARCHAR*(50) NOT NULL,  
     costo          *DECIMAL*(10, 2) NOT NULL,  
     tiempo\_entrega *DATETIME*  
  );

*-- Tabla Detalle de la Orden*  
CREATE TABLE IF NOT EXISTS detalle\_orden  
  (  
     id\_detalle   *INT* auto\_increment PRIMARY KEY,  
     cantidad     *INT*,  
     precio\_total *DECIMAL*(10, 2) NOT NULL,  
     id\_producto  *INT*,  
     id\_categoria *INT*,  
     id\_usuario   *INT*,  
     id\_envio     *INT*,  
     id\_orden     *INT*,  
     FOREIGN KEY(id\_producto) REFERENCES productos(id\_producto),  
     FOREIGN KEY(id\_categoria) REFERENCES categorias(id\_categoria),  
     FOREIGN KEY(id\_envio) REFERENCES envios(id\_envio),  
     FOREIGN KEY(id\_orden) REFERENCES ordenes(id\_orden)  
  );  
  
*-- Tabla de Carrito de compras*  
CREATE TABLE IF NOT EXISTS carrito  
  (  
     id\_carrito     *INT* auto\_increment PRIMARY KEY,  
     fecha\_creacion *DATETIME*,  
     id\_orden       *INT*,  
     FOREIGN KEY(id\_orden) REFERENCES ordenes(id\_orden)  
  );  
  
*-- Tabla Detalle del carrito de compras*  
CREATE TABLE IF NOT EXISTS detalle\_carrito  
  (  
     id\_detalle\_carrito *INT* auto\_increment PRIMARY KEY,  
     cantidad           *INT*,  
     precio\_total       *DECIMAL*(10, 2) NOT NULL,  
     id\_carrito         *INT*,  
     id\_orden           *INT*,  
     id\_producto        *INT*,  
     id\_categoria       *INT*,  
     id\_proveedor       *INT*,  
     FOREIGN KEY(id\_carrito) REFERENCES carrito(id\_carrito),  
     FOREIGN KEY(id\_orden) REFERENCES ordenes(id\_orden),  
     FOREIGN KEY(id\_producto) REFERENCES productos(id\_producto),  
     FOREIGN KEY(id\_categoria) REFERENCES categorias(id\_categoria),  
     FOREIGN KEY(id\_proveedor) REFERENCES proveedores(id\_proveedor)  
  );  
  
*-- Tabla de Comentarios*  
CREATE TABLE IF NOT EXISTS comentarios  
  (  
     id\_comentario *INT* auto\_increment PRIMARY KEY,  
     comentario    *VARCHAR*(255) NOT NULL,  
     fecha         *DATETIME*,  
     estado        *VARCHAR*(45),  
     calificacion  *VARCHAR*(45),  
     id\_producto   *INT*,  
     id\_categoria  *INT*,  
     FOREIGN KEY(id\_producto) REFERENCES productos(id\_producto),  
     FOREIGN KEY(id\_categoria) REFERENCES categorias(id\_categoria)  
  );



### Inserción de datos

INSERT INTO usuarios (id\_usuario,tipo\_documento, dni, nombre, apellido, correo, contrasena, telefono, fecha\_nacimiento) VALUES  
(1,'DNI', 123456789, 'Juan', 'Pérez', 'juan.perez@email.com', 'contraseña123', '987654321', '1990-05-15 00:00:00'),  
(2,'Pasaporte', 987654321, 'Ana', 'Gómez', 'ana.gomez@email.com', 'clave\_segura', '654321987', '1985-08-22 00:00:00'),  
(3,'DNI', 567890123, 'Pedro', 'Martínez', 'pedro.martinez@email.com', 'password123', '789012345', '1992-02-10 00:00:00'),  
(4,'Carné de Identidad', 456789012, 'Laura', 'García', 'laura.garcia@email.com', 'seguridad456', '012345678', '1988-11-30 00:00:00'),  
(5,'DNI', 345678901, 'Miguel', 'López', 'miguel.lopez@email.com', 'contraseña456', '345678901', '1995-07-05 00:00:00'),  
(6,'Pasaporte', 234567890, 'Carolina', 'Sánchez', 'carolina.sanchez@email.com', 'clave\_segura789', '567890123', '1998-04-18 00:00:00'),  
(7,'DNI', 876543210, 'Daniel', 'Fernández', 'daniel.fernandez@email.com', 'secure\_pass', '678901234', '1983-09-12 00:00:00'),  
(8,'Carné de Identidad', 765432109, 'Isabel', 'Díaz', 'isabel.diaz@email.com', 'password789', '890123456', '1991-12-25 00:00:00'),  
(9,'DNI', 654321098, 'Lucas', 'Hernández', 'lucas.hernandez@email.com', 'contraseña789', '901234567', '1980-06-08 00:00:00'),  
(10,'Pasaporte', 543210987, 'María', 'Jiménez', 'maria.jimenez@email.com', 'seguro\_pass', '234567890', '1987-03-03 00:00:00');

INSERT INTO historial\_compras (id\_historial, fecha, estado, id\_usuario)

VALUES  
(1, '2024-06-01 10:15:00', 'Completado', 1),  
(2, '2024-06-02 12:30:00', 'Pendiente', 2),  
(3, '2024-06-03 14:45:00', 'Cancelado', 3),  
(4, '2024-06-04 09:00:00', 'Completado', 4),  
(5, '2024-06-05 11:20:00', 'Pendiente', 5),  
(6, '2024-06-06 16:00:00', 'Completado', 6),  
(7, '2024-06-07 08:45:00', 'Cancelado', 7),  
(8, '2024-06-08 13:10:00', 'Completado', 8),  
(9, '2024-06-09 15:30:00', 'Pendiente', 9),  
(10, '2024-06-10 10:00:00', 'Completado', 10);

INSERT INTO categorias (id\_categoria, nombre, descripcion ,fecha\_creacion )

VALUES  
(1, 'Electrónicos','Productos electrónicos como teléfonos, computadoras y accesorios',**NOW**()),  
(2, 'Ropa','Ropa de moda para hombres, mujeres y niños',**NOW**()),  
(3, 'Hogar y Jardín','Artículos para el hogar, muebles y suministros de jardinería.',**NOW**()),  
(4, 'Deportes','Equipos y accesorios para deportes y actividades al aire libre.',**NOW**()),  
(5, 'Libros','Libros de diversas categorías',**NOW**()),  
(6, 'Juguetes','Juguetes para niños de todas las edades',**NOW**()),  
(7, 'Joyería','Joyas y accesorios',**NOW**()),  
(8, 'Alimentación','Productos alimenticios y gourmet',**NOW**()),  
(9, 'Salud y Belleza','Productos para el cuidado personal',**NOW**()),  
(10, 'Automotriz','Accesorios y productos para automóviles',**NOW**());

INSERT INTO productos (id\_producto, nombre, description, price, cantidad)

VALUES  
(1, 'Smartphone X', 'Potente smartphone con cámara dual', 599.99, 50),  
(2, 'Camiseta de Algodón', 'Camiseta cómoda y de alta calidad', 19.99, 100),  
(3, 'Juego de Sartenes', 'Set de sartenes antiadherentes', 89.95, 30),  
(4, 'Balón de Fútbol', 'Balón oficial para partidos', 29.99, 20),  
(5, 'Libro "Aventuras en el Espacio"', 'Novela de ciencia ficción', 14.50, 80),  
(6, 'Muñeca Interactiva', 'Muñeca que habla y realiza acciones', 39.99, 15),  
(7, 'Collar de Diamantes', 'Collar elegante con diamantes', 499.99, 10),  
(8, 'Pack de Snacks Saludables', 'Variado pack de snacks saludables', 29.95, 50),  
(9, 'Kit de Cuidado Facial', 'Productos para el cuidado facial', 59.99, 25),  
(10, 'Limpiaparabrisas Universal', 'Limpiaparabrisas de calidad para automóviles', 12.75, 40);

INSERT INTO detalle\_historial (price, cantidad, comentario, id\_historial, id\_usuario, id\_producto, id\_categoria) VALUES  
(59.99, 2, 'Buena compra', 1, 3, 1, 1),  
(19.99, 1, 'Talla correcta', 1, 2, 2, 2),  
(89.95, 1, 'Excelente calidad', 1, 1, 3, 3),  
(29.99, 3, 'Divertido para jugar en equipo', 2, 4, 4, 4),  
(14.50, 1, 'Interesante trama', 2, 3, 5, 5),  
(39.99, 1, 'Ideal para regalar', 3, 1, 6, 6),  
(499.99, 1, 'Muy elegante', 3, 2, 7, 7),  
(29.95, 2, 'Deliciosos snacks', 4, 4, 8, 8),  
(59.99, 1, 'Noté mejoras en mi piel', 4, 3, 9, 9),  
(12.75, 1, 'Fácil instalación', 5, 1, 10, 10);

INSERT INTO proveedores (id\_proveedor,nombre, contacto, direccion, estado, info\_financiera, tipo) VALUES  
(1,'Proveedor A', 999888777, 'Calle Principal 123', 'Activo', 'Cuenta Bancaria: XXXX-XXXX-XXXX-XXXX', 'Electrónicos'),  
(2,'Proveedor B', 888777666, 'Avenida Secundaria 456', 'Activo', 'Cuenta Bancaria: YYYY-YYYY-YYYY-YYYY', 'Ropa'),  
(3,'Proveedor C', 777666555, 'Plaza Comercial 789', 'Inactivo', 'Cuenta Bancaria: ZZZZ-ZZZZ-ZZZZ-ZZZZ', 'Hogar'),  
(4,'Proveedor D', 666555444, 'Calle Peatonal 101', 'Activo', 'Cuenta Bancaria: WWWW-WWWW-WWWW-WWWW', 'Deportes'),  
(5,'Proveedor E', 555444333, 'Calle de los Libros 202', 'Inactivo', 'Cuenta Bancaria: VVVV-VVVV-VVVV-VVVV', 'Libros'),  
(6,'Proveedor F', 444333222, 'Esquina de los Juguetes 303', 'Activo', 'Cuenta Bancaria: UUUU-UUUU-UUUU-UUUU', 'Juguetes'),  
(7,'Proveedor G', 333222111, 'Avenida de las Joyas 404', 'Activo', 'Cuenta Bancaria: TTTT-TTTT-TTTT-TTTT', 'Joyería'),  
(8,'Proveedor H', 222111000, 'Calle de los Alimentos 505', 'Inactivo', 'Cuenta Bancaria: SSSS-SSSS-SSSS-SSSS', 'Alimentación'),  
(9,'Proveedor I', 111000999, 'Paseo de la Salud 606', 'Activo', 'Cuenta Bancaria: RRRR-RRRR-RRRR-RRRR', 'Salud y Belleza'),  
(10,'Proveedor J', 999000888, 'Bulevar Automotriz 707', 'Activo', 'Cuenta Bancaria: QQQQ-QQQQ-QQQQ-QQQQ', 'Automotriz');

INSERT INTO promociones (id\_promocion, nombre, descuento, fecha\_inicio, fecha\_fin, id\_producto, id\_categoria, id\_proveedor) VALUES  
(1, 'Oferta Verano', 0.15, '2024-06-01 00:00:00', '2024-06-30 23:59:59', 1, 1, 1),  
(2, 'Descuento Ropa', 0.20, '2024-07-01 00:00:00', '2024-07-15 23:59:59', 2, 2, 2),  
(3, 'Liquidación Hogar', 0.30, '2024-08-01 00:00:00', '2024-08-31 23:59:59', 3, 3, 3),  
(4, 'Especial Deportes', 0.25, '2024-09-01 00:00:00', '2024-09-15 23:59:59', 4, 4, 4),  
(5, 'Promo Libros', 0.18, '2024-10-01 00:00:00', '2024-10-31 23:59:59', 5, 5, 5),  
(6, 'Juguetes Rebajados', 0.22, '2024-11-01 00:00:00', '2024-11-15 23:59:59', 6, 6, 6),  
(7, 'Descuento Joyería', 0.15, '2024-12-01 00:00:00', '2024-12-31 23:59:59', 7, 7, 7),  
(8, 'Oferta Alimentación', 0.25, '2025-01-01 00:00:00', '2025-01-15 23:59:59', 8, 8, 8),  
(9, 'Cuidado Facial en Oferta', 0.20, '2025-02-01 00:00:00', '2025-02-28 23:59:59', 9, 9, 9),  
(10, 'Accesorios Automotrices', 0.15, '2025-03-01 00:00:00', '2025-03-15 23:59:59', 10, 10, 10);

INSERT INTO ordenes (id\_orden, fecha, estado, id\_promocion) VALUES  
(1, '2024-06-01 10:15:00', 'En Proceso', 1),  
(2, '2024-06-02 12:30:00', 'Enviado', 2),  
(3, '2024-06-03 14:45:00', 'Entregado', 3),  
(4, '2024-06-04 09:00:00', 'En Proceso', 4),  
(5, '2024-06-05 11:20:00', 'Enviado', 5),  
(6, '2024-06-06 16:00:00', 'Entregado', 6),  
(7, '2024-06-07 08:45:00', 'En Proceso', 7),  
(8, '2024-06-08 13:10:00', 'Enviado', 8),  
(9, '2024-06-09 15:30:00', 'Entregado', 9),  
(10, '2024-06-10 10:00:00', 'Enviado', 10);

INSERT INTO envios (id\_envio, direccion, estado, fecha\_entrega, metodo, courrier, costo, tiempo\_entrega) VALUES  
(1, 'Calle Principal 123', 'En Proceso', '2024-06-15 14:00:00', 'Estándar', 'EnvíoExpress', 15.99, NULL),  
(2, 'Avenida Secundaria 456', 'Enviado', '2024-06-16 10:30:00', 'Express', 'RápidoEnvíos', 25.50, '2024-06-18 14:00:00'),  
(3, 'Plaza Comercial 789', 'Entregado', '2024-06-17 11:45:00', 'Estándar', 'MegaLogística', 12.75, '2024-06-19 12:30:00'),  
(4, 'Calle Peatonal 101', 'En Proceso', '2024-06-18 09:30:00', 'Express', 'EntregaVeloz', 20.00, NULL),  
(5, 'Calle de los Libros 202', 'Enviado', '2024-06-19 12:15:00', 'Estándar', 'LibrosExpress', 18.99, '2024-06-21 15:00:00'),  
(6, 'Esquina de los Juguetes 303', 'Entregado', '2024-06-20 15:30:00', 'Express', 'JugueteRápido', 22.50, '2024-06-22 10:45:00'),  
(7, 'Avenida de las Joyas 404', 'Enviado', '2024-06-21 14:45:00', 'Estándar', 'JoyasEnvíos', 30.00, '2024-06-23 13:00:00'),  
(8, 'Calle de los Alimentos 505', 'En Proceso', '2024-06-22 10:00:00', 'Express', 'RicoEnvío', 15.75, NULL),  
(9, 'Paseo de la Salud 606', 'Entregado', '2024-06-23 12:30:00', 'Estándar', 'SaludableExpress', 28.99, '2024-06-25 09:15:00'),  
(10, 'Bulevar Automotriz 707', 'En Proceso', '2024-06-24 09:45:00', 'Express', 'AutoRápido', 19.50, NULL);

INSERT INTO detalle\_orden (id\_detalle, cantidad, precio\_total, id\_producto, id\_categoria, id\_usuario, id\_envio, id\_orden) VALUES  
(1, 2, 35.99, 1, 1, 1, 1, 1),  
(2, 1, 25.50, 2, 2, 2, 2, 2),  
(3, 3, 45.75, 3, 3, 3, 3, 3),  
(4, 4, 80.00, 4, 4, 4, 4, 4),  
(5, 2, 37.98, 5, 5, 5, 5, 5),  
(6, 1, 44.99, 6, 6, 6, 6, 6),  
(7, 3, 90.00, 7, 7, 7, 7, 7),  
(8, 2, 31.50, 8, 8, 8, 8, 8),  
(9, 1, 28.99, 9, 9, 9, 9, 9),  
(10, 4, 65.00, 10, 10, 10, 10, 10);

INSERT INTO carrito (id\_carrito, fecha\_creacion, id\_orden) VALUES  
(1, '2024-06-01 10:15:00', 1),  
(2, '2024-06-02 12:30:00', 2),  
(3, '2024-06-03 14:45:00', 3),  
(4, '2024-06-04 09:00:00', 4),  
(5, '2024-06-05 11:20:00', 5),  
(6, '2024-06-06 16:00:00', 6),  
(7, '2024-06-07 08:45:00', 7),  
(8, '2024-06-08 13:10:00', 8),  
(9, '2024-06-09 15:30:00', 9),  
(10, '2024-06-10 10:00:00', 10);

INSERT INTO detalle\_carrito (id\_detalle\_carrito, cantidad, precio\_total, id\_carrito, id\_orden, id\_producto, id\_categoria, id\_proveedor) VALUES  
(1, 2, 35.99, 1, 1, 1, 1, 1),  
(2, 1, 25.50, 2, 2, 2, 2, 2),  
(3, 3, 45.75, 3, 3, 3, 3, 3),  
(4, 4, 80.00, 4, 4, 4, 4, 4),  
(5, 2, 37.98, 5, 5, 5, 5, 5),  
(6, 1, 44.99, 6, 6, 6, 6, 6),  
(7, 3, 90.00, 7, 7, 7, 7, 7),  
(8, 2, 31.50, 8, 8, 8, 8, 8),  
(9, 1, 28.99, 9, 9, 9, 9, 9),  
(10, 4, 65.00, 10, 10, 10, 10, 10);

INSERT INTO comentarios (id\_comentario, comentario, fecha, estado, calificacion, id\_producto, id\_categoria) VALUES  
(1, '¡Excelente producto, estoy muy satisfecho!', '2024-06-01 14:30:00', 'Aprobado', '5 estrellas', 1, 1),  
(2, 'Buen servicio al cliente, rápido y eficiente.', '2024-06-02 11:45:00', 'Aprobado', '4 estrellas', 2, 2),  
(3, 'Producto de calidad, lo recomiendo.', '2024-06-03 09:20:00', 'Pendiente', '5 estrellas', 3, 3),  
(4, 'No recibí mi pedido a tiempo, servicio insatisfactorio.', '2024-06-04 16:00:00', 'Rechazado', '2 estrellas', 4, 4),  
(5, '¡Me encanta! Compré otro para regalar.', '2024-06-05 12:15:00', 'Aprobado', '5 estrellas', 5, 5),  
(6, 'El producto no cumplió con mis expectativas.', '2024-06-06 14:45:00', 'Aprobado', '3 estrellas', 6, 6),  
(7, 'Servicio de entrega excelente, llegó antes de lo esperado.', '2024-06-07 10:30:00', 'Aprobado', '4 estrellas', 7, 7),  
(8, 'Calidad regular, esperaba más.', '2024-06-08 13:00:00', 'Rechazado', '2 estrellas', 8, 8),  
(9, '¡Me encanta este producto! Compraría de nuevo.', '2024-06-09 15:30:00', 'Pendiente', '5 estrellas', 9, 9),  
(10, 'Buen servicio, producto conforme a la descripción.', '2024-06-10 08:45:00', 'Aprobado', '4 estrellas', 10, 10);

### Vistas

*-- 1. Historial de compras por cada Usuario*

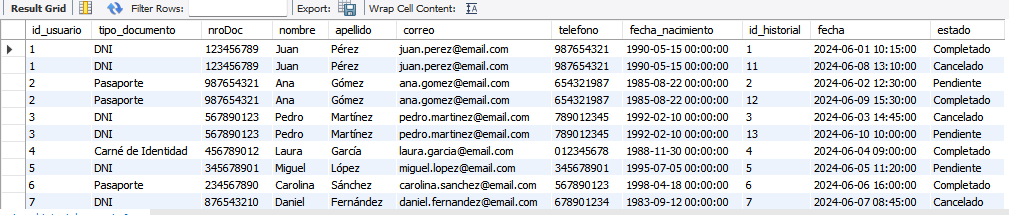
CREATE OR REPLACE VIEW view\_historial\_usuario AS

SELECT u.id\_usuario, u.tipo\_documento, u.nroDoc, u.nombre, u.apellido, u.correo, u.telefono, u.fecha\_nacimiento,h.id\_historial, h.fecha, h.estado

FROM usuarios u

JOIN historial\_compras h ON u.id\_usuario = h.id\_usuario;

SELECT \* FROM view\_historial\_usuario;



*-- 2. Detalle del historial de compras por carrito*

CREATE OR REPLACE VIEW view\_carrito\_detalles AS

SELECT dc.id\_detalle\_carrito, dc.cantidad, dc.precio\_unitario,

c.id\_carrito, c.fecha\_creacion,

p.id\_producto, p.nombre AS nombre\_producto, p.description,

pr.id\_proveedor, pr.nombre AS nombre\_proveedor,

COUNT(\*) AS total\_detalles,

SUM(dc.precio\_unitario \* dc.cantidad) AS total\_precio

FROM detalle\_carrito dc

JOIN carrito c ON dc.id\_carrito = c.id\_carrito

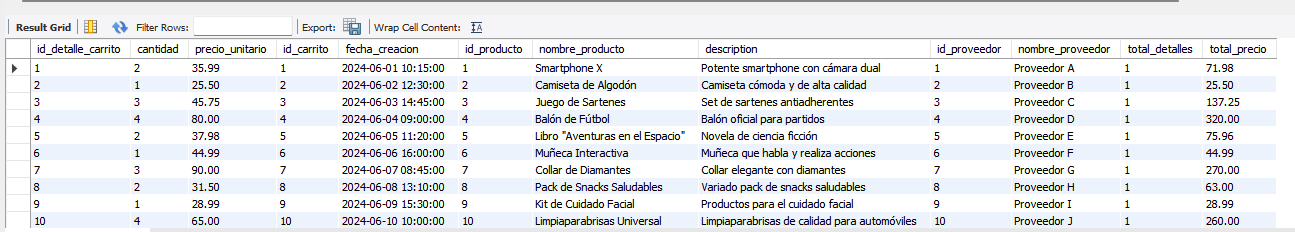
JOIN productos p ON dc.id\_producto = p.id\_producto

JOIN proveedores pr ON dc.id\_proveedor = pr.id\_proveedor

GROUP BY dc.id\_detalle\_carrito, c.id\_carrito, p.id\_producto

ORDER BY c.fecha\_creacion;

SELECT \* FROM VistaCarritoDetalles;



*-- 3. Total de Ventas Por Categoría*

CREATE OR REPLACE VIEW vista\_total\_venta\_por\_categoria AS

SELECT c.id\_categoria, c.nombre AS nombre\_categoria,

COUNT(dh.id\_detalle\_historial) AS total\_ventas,

SUM(dh.price \* dh.cantidad) AS ingreso\_total

FROM categorias c

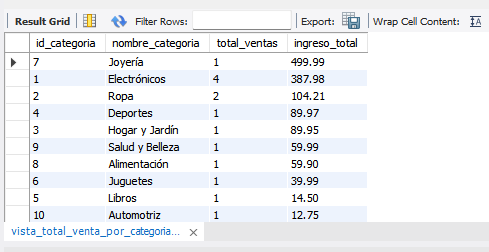
LEFT JOIN productos p ON c.id\_categoria = p.id\_categoria

LEFT JOIN detalle\_historial dh ON p.id\_producto = dh.id\_producto

GROUP BY c.id\_categoria, c.nombre

ORDER BY ingreso\_total DESC;

SELECT \* FROM vista\_total\_venta\_por\_categoria;



### Funciones

USE marketplace\_db;  
*-- 1. Historial de compras por Usuario*

DELIMITER //

CREATE FUNCTION obtener\_historial\_compras\_por\_usuario (usuario\_id int)  
RETURNS varchar(255)

DETERMINISTIC  
BEGIN

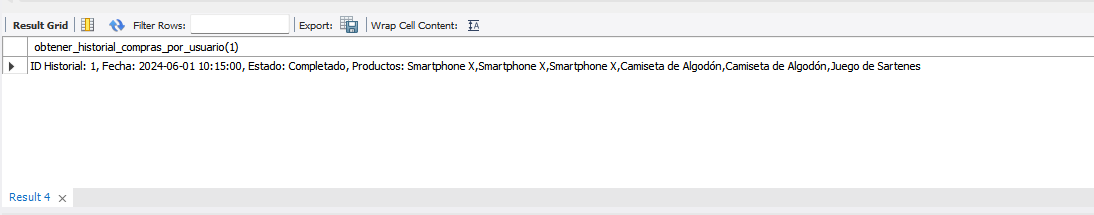
DECLARE resultado *VARCHAR*(255);

SELECT **Group\_concat**(**Concat**('ID Historial: ', hc.id\_historial, ', Fecha: ', hc.fecha, ', Estado: ', hc.estado, ', Productos: ', p.productos))  
  INTO   resultado  
  FROM   historial\_compras hc  
  JOIN  
         (  
                  SELECT   dh.id\_historial,  
                           **Group\_concat**(p.nombre) AS productos  
                  FROM     detalle\_historial dh  
                  JOIN     productos p  
                  ON       dh.id\_producto = p.id\_producto  
                  GROUP BY dh.id\_historial ) p  
  ON     hc.id\_historial = p.id\_historial  
  WHERE  hc.id\_usuario = usuario\_id;

RETURN resultado;

END //  
DELIMITER ;

SELECT obtener\_historial\_compras\_por\_usuario (1);



*- 2.  Productos por categoría específica*

DELIMITER //

CREATE FUNCTION obtener\_productos\_por\_categoria

(categoria\_nombre varchar(45))  
RETURNS varchar(255)

DETERMINISTIC  
BEGIN

DECLARE resultado *VARCHAR*(255);

SELECT **Group\_concat**(p.nombre)  
  INTO   resultado  
  FROM   productos p  
  JOIN   categorias c  
  ON     p.id\_categoria = c.id\_categoria  
  WHERE  c.nombre = categoria\_nombre;

RETURN resultado;

END //  
DELIMITER ;

SELECT obtener\_productos\_por\_categoria ('Electrónicos') AS Resultado;

**

*- 3. Calcular Descuento Total promoción por Ordenes*

DELIMITER //

CREATE FUNCTION calcular\_descuento\_total(orden\_id int)  
RETURNS decimal(10, 2)

DETERMINISTIC  
BEGIN

DECLARE descuento\_total *DECIMAL*(10, 2);

SELECT *Sum*(p.descuento)  
  INTO   descuento\_total  
  FROM   promociones p  
  JOIN   ordenes\_promociones op  
  ON     p.id\_promocion= op.id\_promocion  
  JOIN   ordenes o  
  ON     op.id\_orden= o.id\_orden  
  WHERE  o.id\_orden = orden\_id;

RETURN descuento\_total;

END //  
DELIMITER ;

SELECT **calcular\_descuento\_total**(1);



### Stored Procedures

USE marketplace\_db;

*-- 1. Calcular el Monto total por Orden*

DELIMITER //

CREATE PROCEDURE calcular\_total\_orden(in orden\_id int)  
BEGIN

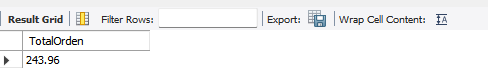
DECLARE total *DECIMAL*(10, 2);

SELECT *Sum*(do.precio\_unitario\*do.cantidad)  
  INTO   total  
  FROM   detalle\_orden DO  
  WHERE  do.id\_orden = orden\_id;

SELECT total AS TotalOrden;

END //  
DELIMITER ;

CALLcalcular\_total\_orden(1);



*- 2. Realizar el orden de una data Escogiendo el Nombre de la tabla, La Columna y Orden*

DELIMITER //

CREATE PROCEDURE ordenar\_tabla

(IN  p\_nombre\_tabla varchar(255),

IN p\_campo\_ordenamiento varchar(255),  
 IN p\_orden              varchar(4) )  
BEGIN

SET @sql\_query = **Concat**('SELECT \* FROM ', p\_nombre\_tabla, ' ORDER BY ', p\_campo\_ordenamiento, ' ', p\_orden);

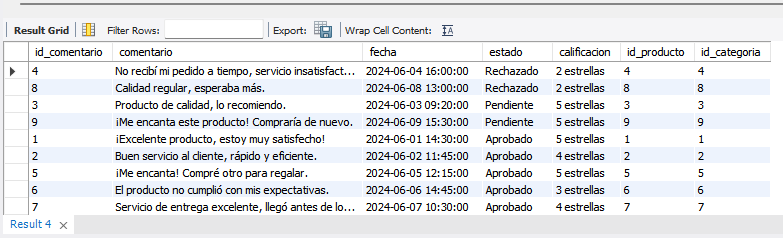
PREPARE stmt FROM @sql\_query;

EXECUTE **Stmt**;

DEALLOCATE prepare stmt;

END //  
DELIMITER ;

CALL ordenar\_tabla('comentarios','estado','DESC');

**

*- 3. Eliminar un item del carrito de compras*

DELIMITER //

CREATE PROCEDURE eliminar\_item\_carrito

(

 IN carrito\_id INT ,  
 IN detalle\_carrito\_id INT

)  
BEGIN

DECLARE existe\_carrito *INT*;

SELECT *Count*(\*)  
  INTO   existe\_carrito  
  FROM   carrito  
  WHERE  id\_carrito = carrito\_id;  
    
  *-- Si el carrito existe, eliminar el detalle del carrito*

IF existe\_carrito > 0 then  
  DELETE  
  FROM   detalle\_carrito  
  WHERE  id\_detalle\_carrito = detalle\_carrito\_id;

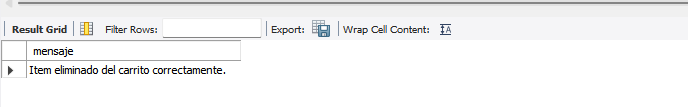
SELECT 'Item eliminado del carrito correctamente.' AS mensaje;

ELSE

SELECT 'El carrito especificado no existe.' AS mensaje;

END IF;

END //  
DELIMITER ;

CALL eliminar\_item\_carrito (10, 10);  
**

*- 4. Crear un item del carrito de compras*

DELIMITER //

CREATE PROCEDURE agregar\_item\_carrito

(

IN  carrito\_id      int,

  IN producto\_id     int,

  IN cantidad        int,

  IN precio\_unitario decimal(10, 2)

)  
BEGIN

DECLARE existe\_carrito *INT*;  
  *-- Verificar si el carrito existe*

SELECT *Count*(\*)  
  INTO   existe\_carrito  
  FROM   carrito  
  WHERE  id\_carrito = carrito\_id;  
    
  *-- Si el carrito existe, agregar el detalle al carrito*

IF existe\_carrito > 0 then  
  INSERT INTO detalle\_carrito  
              (  
                          id\_carrito,  
                          id\_producto,  
                          cantidad,  
                          precio\_unitario  
              )  
              VALUES  
              (  
                          carrito\_id,  
                          producto\_id,  
                          cantidad,  
                          precio\_unitario  
              );

SELECT 'Item agregado al carrito correctamente.' AS mensaje;

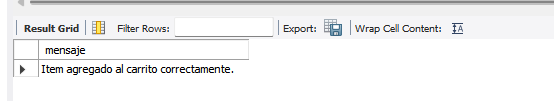
ELSE

SELECT 'El carrito especificado no existe.' AS mensaje;

END IF;

END //  
DELIMITER ;

CALL agregar\_item\_carrito (10, 8,3, 29.95);



### 6. Triggers

USE marketplace\_db;  
*-- 1.*  
*-- Creación de la Tabla log de productos*

CREATE TABLE IF NOT EXISTS log\_productos

(

log\_id int auto\_increment PRIMARY KEY,

id\_producto int NOT NULL,

usuario\_modificacion varchar(50),

fecha\_modificacion date NOT NULL,

hora\_modificacion time NOT NULL,

accion\_realizada varchar(255) NOT NULL,

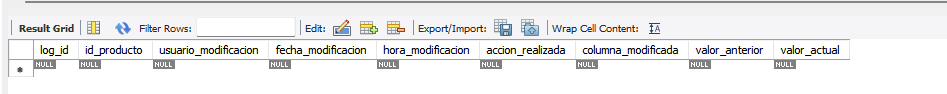
columna\_modificada varchar(255) NULL,

valor\_anterior varchar(255) NULL,

valor\_actual varchar(255) NULL

 );

SELECT \* FROM log\_productos;



*- Creación de un Trigger BEFORE UPDATE*

DELIMITER //

CREATE TRIGGER before\_update\_log\_productos

BEFORE UPDATE on productos

FOR EACH ROW

BEGIN

INSERT INTO log\_productos (id\_producto, usuario\_modificacion, fecha\_modificacion, hora\_modificacion, accion\_realizada, valor\_anterior, valor\_actual)

VALUES (NEW.id\_producto, 'nombre\_de\_usuario', CURDATE(), CURTIME(), 'before update', NEW.nombre, NEW.nombre);

END //  
DELIMITER ;

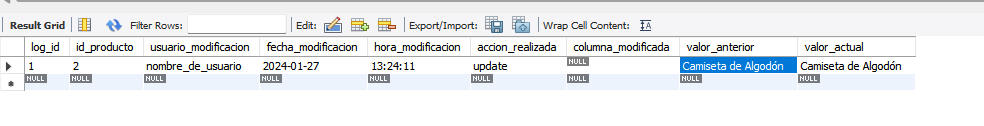
UPDATE productos

SET price =23.51

WHERE id\_producto=2

SELECT \* FROM productos;

SELECT \* FROM log\_productos;

**

*- Creación Trigger AFTER UPDATE*

DELIMITER //

CREATE TRIGGER after\_update\_producto\_log\_trigger

AFTER UPDATE ON productos

FOR EACH ROW

BEGIN

DECLARE columna\_modificada *VARCHAR*(255);

DECLARE valor\_anterior     *VARCHAR*(255);

DECLARE valor\_actual       *VARCHAR*(255);

*-- Determinar la columna modificada*

IF old.price <> new.price then  
SET columna\_modificada = 'price';

SET valor\_anterior = old.price;

SET valor\_actual = new.price;

ELSEIF old.cantidad <> new.cantidad THEN SET columna\_modificada = 'cantidad';

SET valor\_anterior = old.cantidad;

SET valor\_actual = new.cantidad;

ELSEIF old.description <> new.description THEN SET columna\_modificada = 'description';

SET valor\_anterior = old.description;

SET valor\_actual = new.description;

ELSEIF old.nombre <> new.nombre

THEN SET columna\_modificada = 'nombre';

SET valor\_anterior = old.nombre;

SET valor\_actual = new.nombre;

END IF;

*-- Insertar en el log solo si alguna columna ha cambiado*

IF columna\_modificada IS NOT NULL THEN

INSERT INTO log\_productos (id\_producto, usuario\_modificacion, fecha\_modificacion, hora\_modificacion, accion\_realizada, columna\_modificada, valor\_anterior, valor\_actual)

VALUES (NEW.id\_producto, CURRENT\_USER(), CURDATE(), CURTIME(), 'After update', columna\_modificada, valor\_anterior, valor\_actual);

END IF;

END //  
DELIMITER ;

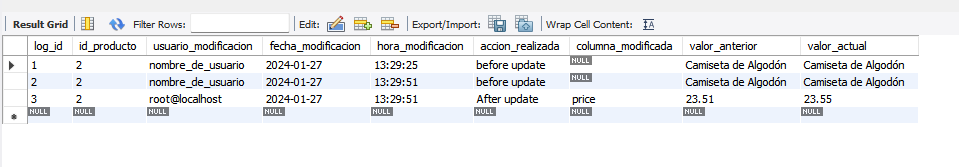
UPDATE productos

SET price =23.55

WHERE id\_producto=2;

SELECT \* FROM productos;

SELECT \* FROM log\_productos;



*-- 2. Creación de la Tabla log de usuarios*

CREATE TABLE IF NOT EXISTS log\_usuarios (

log\_id int auto\_increment PRIMARY KEY,

id\_usuario int NOT NULL,

campo\_modificado VARCHAR(50) NOT NULL,

valor\_anterior VARCHAR(255),

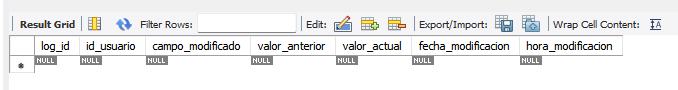
valor\_actual VARCHAR(255),

fecha\_modificacion date NOT NULL,

hora\_modificacion time NOT NULL

);

SELECT \* FROM log\_usuarios ;



*-- Crear trigger BEFORE UPDATE*

DELIMITER //

CREATE TRIGGER before\_update\_log\_usuarios

BEFORE UPDATE ON usuarios

FOR EACH ROW

BEGIN

IF NEW.tipo\_documento <> OLD.tipo\_documento THEN

INSERT INTO log\_usuarios (id\_usuario, campo\_modificado, valor\_anterior, valor\_actual, fecha\_modificacion, hora\_modificacion)

VALUES (OLD.id\_usuario, 'tipo\_documento', OLD.tipo\_documento, NEW.tipo\_documento, CURDATE(), CURTIME());

END IF;

IF NEW.nroDoc <> OLD.nroDoc THEN

INSERT INTO log\_usuarios (id\_usuario, campo\_modificado, valor\_anterior, valor\_actual, fecha\_modificacion, hora\_modificacion)

VALUES (OLD.id\_usuario, 'nroDoc', OLD.nroDoc, NEW.nroDoc, CURDATE(), CURTIME());

END IF;

END;

//

DELIMITER ;

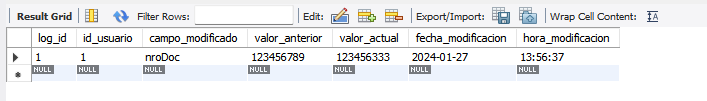
SELECT \* FROM usuarios;

SELECT \* FROM log\_usuarios;

UPDATE usuarios

SET nroDoc="123456333"

WHERE id\_usuario=1



*- Crear trigger AFTER INSERT*

DELIMITER //

CREATE TRIGGER after\_insert\_log\_usuarios   
AFTER INSERT ON usuarios

FOR EACH ROW

BEGIN

INSERT INTO log\_usuarios (id\_usuario, campo\_modificado, valor\_anterior , valor\_actual , fecha\_modificacion, hora\_modificacion)

VALUES (NEW.id\_usuario, 'nuevo\_usuario\_insertado', NULL, NULL, CURDATE(), CURTIME());

END;

//

DELIMITER ;

INSERT INTO usuarios (id\_usuario,tipo\_documento, nroDoc, nombre, apellido, correo, contrasena, telefono, fecha\_nacimiento) VALUES

(11,'DNI', 123456111, 'José', 'Pérez', 'jose.perez@email.com', 'contraseña123', '987654321', '1990-05-15 00:00:00');

SELECT \* FROM log\_usuarios;

