-- =====================================================

-- COMANDOS PARA CREAR LA BASE DE DATOS IDÉNTICA

-- Sistema E-commerce Atilio Marola - PPS

-- =====================================================

-- Crear la base de datos principal

CREATE DATABASE IF NOT EXISTS atilio\_marola;

USE atilio\_marola;

-- =====================================================

-- 1. TABLA DE ROLES

-- =====================================================

CREATE TABLE IF NOT EXISTS roles (

    idRol INT AUTO\_INCREMENT PRIMARY KEY,

    nombreRol VARCHAR(50) NOT NULL UNIQUE,

    descripcion TEXT

);

-- Insertar roles del sistema

INSERT INTO roles (idRol, nombreRol, descripcion) VALUES

(1, 'Cliente', 'Usuario que puede comprar productos y solicitar servicios'),

(2, 'Vendedor', 'Usuario que puede gestionar ventas y productos'),

(3, 'Administrador', 'Usuario con acceso completo al sistema');

-- =====================================================

-- 2. TABLA DE USUARIOS (Base del sistema)

-- =====================================================

CREATE TABLE IF NOT EXISTS usuarios (

    idUsuario INT AUTO\_INCREMENT PRIMARY KEY,

    nombre VARCHAR(100) NOT NULL,

    apellido VARCHAR(100) NOT NULL,

    email VARCHAR(255) NOT NULL UNIQUE,

    password VARCHAR(255) NOT NULL,

    idRol INT NOT NULL,

    fechaRegistro TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

    activo BOOLEAN DEFAULT TRUE,

    FOREIGN KEY (idRol) REFERENCES roles(idRol)

);

-- =====================================================

-- 3. TABLA DE CLIENTES (Información adicional)

-- =====================================================

CREATE TABLE IF NOT EXISTS clientes (

    idCliente INT AUTO\_INCREMENT PRIMARY KEY,

    idUsuario INT NOT NULL UNIQUE,

    direccion VARCHAR(500),

    telefono VARCHAR(20),

    fechaNacimiento DATE,

    FOREIGN KEY (idUsuario) REFERENCES usuarios(idUsuario) ON DELETE CASCADE

);

-- =====================================================

-- 4. TABLA DE SUCURSALES

-- =====================================================

CREATE TABLE IF NOT EXISTS sucursales (

    idSucursal INT AUTO\_INCREMENT PRIMARY KEY,

    nombre VARCHAR(100) NOT NULL,

    direccion VARCHAR(500) NOT NULL,

    telefono VARCHAR(20),

    email VARCHAR(255),

    horario VARCHAR(200),

    activa BOOLEAN DEFAULT TRUE

);

-- =====================================================

-- 5. TABLA DE PRODUCTOS (Con imagen principal)

-- =====================================================

CREATE TABLE IF NOT EXISTS productos (

    idProducto INT AUTO\_INCREMENT PRIMARY KEY,

    nombre VARCHAR(255) NOT NULL,

    descripcion TEXT,

    precio DECIMAL(10,2) NOT NULL,

    stockTotal INT DEFAULT 0,

    imagen VARCHAR(255), -- Imagen principal (compatibilidad)

    categoria VARCHAR(100),

    activo BOOLEAN DEFAULT TRUE,

    fechaCreacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

    fechaActualizacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

-- =====================================================

-- 6. TABLA DE MÚLTIPLES IMÁGENES POR PRODUCTO

-- =====================================================

CREATE TABLE IF NOT EXISTS producto\_imagenes (

    id INT AUTO\_INCREMENT PRIMARY KEY,

    producto\_id INT NOT NULL,

    imagen VARCHAR(255) NOT NULL,

    orden INT DEFAULT 0,

    created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

    FOREIGN KEY (producto\_id) REFERENCES productos(idProducto) ON DELETE CASCADE,

    INDEX idx\_producto\_orden (producto\_id, orden)

);

-- =====================================================

-- 7. TABLA DE STOCK POR SUCURSAL

-- =====================================================

CREATE TABLE IF NOT EXISTS stock\_sucursal (

    idStock INT AUTO\_INCREMENT PRIMARY KEY,

    idSucursal INT NOT NULL,

    idProducto INT NOT NULL,

    stockDisponible INT DEFAULT 0,

    stockMinimo INT DEFAULT 5,

    fechaActualizacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

    FOREIGN KEY (idSucursal) REFERENCES sucursales(idSucursal),

    FOREIGN KEY (idProducto) REFERENCES productos(idProducto),

    UNIQUE KEY unique\_sucursal\_producto (idSucursal, idProducto)

);

-- =====================================================

-- 8. TABLA DE SOLICITUDES DE SERVICIOS POST-VENTA

-- =====================================================

CREATE TABLE IF NOT EXISTS solicitudes\_servicio\_postventa (

    idSolicitud INT AUTO\_INCREMENT PRIMARY KEY,

    idUsuario INT NOT NULL,

    tipoServicio ENUM('instalacion', 'mantenimiento', 'garantia') NOT NULL,

    descripcion TEXT NOT NULL,

    direccion VARCHAR(500) NOT NULL,

    telefono VARCHAR(20),

    fechaPreferida DATE,

    horaPreferida TIME,

    estado ENUM('pendiente', 'confirmado', 'en\_proceso', 'completado', 'cancelado') DEFAULT 'pendiente',

    observacionesAdmin TEXT,

    fechaCreacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

    fechaActualizacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

    FOREIGN KEY (idUsuario) REFERENCES usuarios(idUsuario)

);

-- =====================================================

-- 9. TABLA DE PEDIDOS

-- =====================================================

CREATE TABLE IF NOT EXISTS pedidos (

    idPedido INT AUTO\_INCREMENT PRIMARY KEY,

    idCliente INT NOT NULL,

    idSucursal INT,

    total DECIMAL(10,2) NOT NULL,

    estado ENUM('pendiente', 'confirmado', 'preparando', 'enviado', 'entregado', 'cancelado') DEFAULT 'pendiente',

    metodoPago VARCHAR(50),

    direccionEnvio VARCHAR(500),

    observaciones TEXT,

    fechaPedido TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

    fechaActualizacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

    FOREIGN KEY (idCliente) REFERENCES usuarios(idUsuario),

    FOREIGN KEY (idSucursal) REFERENCES sucursales(idSucursal)

);

-- =====================================================

-- 10. TABLA DE DETALLES DE PEDIDOS

-- =====================================================

CREATE TABLE IF NOT EXISTS detalle\_pedidos (

    idDetalle INT AUTO\_INCREMENT PRIMARY KEY,

    idPedido INT NOT NULL,

    idProducto INT NOT NULL,

    cantidad INT NOT NULL,

    precioUnitario DECIMAL(10,2) NOT NULL,

    subtotal DECIMAL(10,2) NOT NULL,

    FOREIGN KEY (idPedido) REFERENCES pedidos(idPedido) ON DELETE CASCADE,

    FOREIGN KEY (idProducto) REFERENCES productos(idProducto)

);

-- =====================================================

-- 11. TABLA DE HISTORIAL (Auditoría)

-- =====================================================

CREATE TABLE IF NOT EXISTS historial (

    idHistorial INT AUTO\_INCREMENT PRIMARY KEY,

    tabla VARCHAR(50) NOT NULL,

    idRegistro INT NOT NULL,

    accion ENUM('crear', 'actualizar', 'eliminar') NOT NULL,

    usuario VARCHAR(255),

    descripcion TEXT,

    fechaAccion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- =====================================================

-- DATOS INICIALES ESENCIALES

-- =====================================================

-- Usuario administrador inicial (password: admin123)

INSERT INTO usuarios (nombre, apellido, email, password, idRol) VALUES

('Admin', 'Sistema', 'admin@atiliomarola.com', '$2a$10$N9qo8uLOickgx2ZMRZoMye1IYZIP/CqVvOUMLEL6jE0ZQZZ6zLzze', 3);

-- Sucursales principales

INSERT INTO sucursales (idSucursal, nombre, direccion, telefono, email, horario) VALUES

(1, 'Sucursal Centro', 'Av. Rivadavia 1000, CABA', '+54 11 4000-1000', 'centro@atiliomarola.com', 'Lun-Vie: 9:00-18:00, Sáb: 9:00-13:00'),

(2, 'Sucursal Norte', 'Av. Cabildo 2000, CABA', '+54 11 4000-2000', 'norte@atiliomarola.com', 'Lun-Vie: 8:30-19:00, Sáb: 9:00-14:00'),

(3, 'Sucursal Sur', 'Av. Avellaneda 3000, Avellaneda', '+54 11 4000-3000', 'sur@atiliomarola.com', 'Lun-Vie: 9:00-17:30');

-- =====================================================

-- ÍNDICES PARA OPTIMIZACIÓN

-- =====================================================

-- Índices para usuarios

CREATE INDEX idx\_usuarios\_email ON usuarios(email);

CREATE INDEX idx\_usuarios\_rol ON usuarios(idRol);

-- Índices para productos

CREATE INDEX idx\_productos\_categoria ON productos(categoria);

CREATE INDEX idx\_productos\_activo ON productos(activo);

CREATE INDEX idx\_productos\_precio ON productos(precio);

-- Índices para pedidos

CREATE INDEX idx\_pedidos\_cliente ON pedidos(idCliente);

CREATE INDEX idx\_pedidos\_estado ON pedidos(estado);

CREATE INDEX idx\_pedidos\_fecha ON pedidos(fechaPedido);

-- Índices para servicios

CREATE INDEX idx\_servicios\_usuario ON solicitudes\_servicio\_postventa(idUsuario);

CREATE INDEX idx\_servicios\_estado ON solicitudes\_servicio\_postventa(estado);

CREATE INDEX idx\_servicios\_tipo ON solicitudes\_servicio\_postventa(tipoServicio);

CREATE INDEX idx\_servicios\_fecha\_creacion ON solicitudes\_servicio\_postventa(fechaCreacion);

-- =====================================================

-- TRIGGERS PARA MANTENER CONSISTENCIA

-- =====================================================

-- Trigger para actualizar stockTotal cuando cambia stock\_sucursal

DELIMITER //

CREATE TRIGGER actualizar\_stock\_total

AFTER UPDATE ON stock\_sucursal

FOR EACH ROW

BEGIN

    UPDATE productos

    SET stockTotal = (

        SELECT COALESCE(SUM(stockDisponible), 0)

        FROM stock\_sucursal

        WHERE idProducto = NEW.idProducto

    )

    WHERE idProducto = NEW.idProducto;

END//

DELIMITER ;

-- Trigger para actualizar total del pedido

DELIMITER //

CREATE TRIGGER actualizar\_total\_pedido

AFTER INSERT ON detalle\_pedidos

FOR EACH ROW

BEGIN

    UPDATE pedidos

    SET total = (

        SELECT SUM(subtotal)

        FROM detalle\_pedidos

        WHERE idPedido = NEW.idPedido

    )

    WHERE idPedido = NEW.idPedido;

END//

DELIMITER ;

-- =====================================================

-- CONFIGURACIÓN DE CONEXIÓN

-- =====================================================

-- Configuración para tu archivo DB.js:

/\*

const connection = mysql.createConnection({

  host: '127.0.0.1',

  user: 'root',

  password: '12345678', // Cambiar por tu password

  database: 'atilio\_marola',

});

\*/

-- =====================================================

-- COMANDOS DE VERIFICACIÓN

-- =====================================================

-- Verificar que todas las tablas se crearon correctamente:

-- SHOW TABLES;

-- Verificar estructura de tabla principal:

-- DESCRIBE usuarios;

-- DESCRIBE productos;

-- DESCRIBE solicitudes\_servicio\_postventa;

-- Verificar datos iniciales:

-- SELECT \* FROM roles;

-- SELECT \* FROM usuarios;

-- SELECT \* FROM sucursales;

-- =====================================================

-- NOTAS IMPORTANTES

-- =====================================================

/\*

1. Este script crea la base de datos exactamente como la usas en tu proyecto

2. Incluye soporte para múltiples imágenes por producto

3. Mantiene compatibilidad con tu código existente

4. Incluye triggers para mantener consistencia automática

5. Usuario admin inicial: admin@atiliomarola.com / admin123

INSTRUCCIONES DE USO:

1. Abrir MySQL Workbench o terminal de MySQL

2. Ejecutar todo este script

3. Verificar que se crearon todas las tablas

4. Actualizar password del admin si es necesario

\*/