

-- Nota: este script aún no contempla/incluye lo que serían las tablas/entidades generados por las herramientas, lenguajes, frameworks, tecnologías, apis, etc, que usemos. Este solo se centra en lo que sería la base de datos y modelo de negocio general, osea lo que será específico y necesario para el proyecto independientemente de donde o que se le aplique.
-- SCRIPT MODERNO (este toca revisarlo bien en busca de errores pero este script ya debería estar más semejante a todo lo que dicen los requisitos de información de la matriz)

--
=====

=====

-- PROYECTO RED - SCRIPT SQL OPTIMIZADO Y NORMALIZADO
-- Basado en Matriz de Requerimientos y Ficha Técnica
-- Normalización: 1NF, 2NF, 3NF aplicada
-- Priorización: MoSCoW + Eisenhower (MVP primero)

--
=====

=====

SET FOREIGN_KEY_CHECKS = 0;
DROP DATABASE IF EXISTS red;
CREATE DATABASE red CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
USE red;

--
=====

=====

-- FASE 1: ENTIDADES MUST (MVP - Prioridad Alta/Urgente)

--
=====

=====

-- -----
-- MÓDULO: GESTIÓN DE USUARIOS
-- Prioridad: CRÍTICA (MUST) - Base del sistema
-- -----

-- Tabla USUARIO (RI-001)
-- Cumple 3NF: Todos los atributos dependen únicamente de la PK
CREATE TABLE USUARIO (
 ID INT AUTO_INCREMENT PRIMARY KEY COMMENT 'PK autoincremental',
 usuario VARCHAR(100) NOT NULL COMMENT 'Nombre de usuario',
 correo VARCHAR(100) UNIQUE NOT NULL COMMENT 'Email único',
 contrasena VARCHAR(255) NOT NULL COMMENT 'Contraseña hasheada (mín 8
chars)',
 estado ENUM('Activo', 'Inactivo', 'Bloqueado') DEFAULT 'Activo' COMMENT 'Estado del
usuario',
 rol ENUM('Administrador', 'Revisor', 'Usuario') DEFAULT 'Usuario' COMMENT 'Rol del
usuario',

```
    fecha_registro DATETIME DEFAULT CURRENT_TIMESTAMP COMMENT 'Fecha de registro',
```

```
    fecha_ultima_sesion DATETIME NULL COMMENT 'Última sesión',  
    email_verificado BOOLEAN DEFAULT FALSE COMMENT 'Email verificado',  
    intentos_fallidos INT DEFAULT 0 COMMENT 'Intentos fallidos de login',  
    fecha_bloqueo DATETIME NULL COMMENT 'Fecha de bloqueo si aplica',  
    eliminado BOOLEAN DEFAULT FALSE COMMENT 'Soft delete',  
    fecha_eliminacion DATETIME NULL COMMENT 'Fecha eliminación lógica',
```

```
    INDEX idx_correo (correo),  
    INDEX idx_rol (rol),  
    INDEX idx_estado (estado),  
    INDEX idx_email_verificado (email_verificado)
```

```
) COMMENT = 'Usuarios del sistema con roles y estados (RI-001)';
```

```
-- Tabla SESION (RI-002)
```

```
-- 3NF: Depende de usuario_id (FK) y tiene sus propios atributos únicos
```

```
CREATE TABLE SESION (
```

```
    ID INT AUTO_INCREMENT PRIMARY KEY,  
    usuario_id INT NOT NULL,  
    token_sesion VARCHAR(255) UNIQUE NOT NULL COMMENT 'Token único de sesión',  
    fecha_inicio DATETIME DEFAULT CURRENT_TIMESTAMP,  
    fecha_expiracion DATETIME NOT NULL,  
    ip_address VARCHAR(45) COMMENT 'IP del usuario',  
    user_agent TEXT COMMENT 'Navegador/dispositivo',  
    activa BOOLEAN DEFAULT TRUE,
```

```
    FOREIGN KEY (usuario_id) REFERENCES USUARIO(ID) ON DELETE CASCADE,  
    INDEX idx_token (token_sesion),  
    INDEX idx_usuario (usuario_id),  
    INDEX idx_activa (activa)
```

```
) COMMENT = 'Sesiones activas de usuarios (RI-002)';
```

```
-- Tabla TOKEN_VERIFICACION (RI-003)
```

```
-- 3NF: Tokens únicos asociados a usuarios para verificaciones
```

```
CREATE TABLE TOKEN_VERIFICACION (
```

```
    ID INT AUTO_INCREMENT PRIMARY KEY,  
    usuario_id INT NOT NULL,  
    token VARCHAR(255) UNIQUE NOT NULL COMMENT 'Token único',  
    tipo ENUM('Verificacion_Email', 'Recuperacion_Password', 'Cambio_Email') NOT NULL,  
    fecha_creacion DATETIME DEFAULT CURRENT_TIMESTAMP,  
    fecha_expiracion DATETIME NOT NULL COMMENT 'Validez de 24 horas',  
    usado BOOLEAN DEFAULT FALSE,
```

```
    FOREIGN KEY (usuario_id) REFERENCES USUARIO(ID) ON DELETE CASCADE,  
    INDEX idx_token (token),  
    INDEX idx_tipo (tipo),  
    INDEX idx_expiracion (fecha_expiracion)
```

) COMMENT = 'Tokens de verificación con validez de 24h (RI-003, RN-003)';

-- Tabla CAMBIO_EMAIL (RI-004)

-- 3NF: Historial de cambios de email con verificación

```
CREATE TABLE CAMBIO_EMAIL (  
  ID INT AUTO_INCREMENT PRIMARY KEY,  
  usuario_id INT NOT NULL,  
  email_anterior VARCHAR(100) NOT NULL,  
  email_nuevo VARCHAR(100) NOT NULL,  
  token_verificacion VARCHAR(255) NOT NULL,  
  fecha_solicitud DATETIME DEFAULT CURRENT_TIMESTAMP,  
  verificado BOOLEAN DEFAULT FALSE,  
  fecha_verificacion DATETIME NULL,
```

```
  FOREIGN KEY (usuario_id) REFERENCES USUARIO(ID) ON DELETE CASCADE,  
  INDEX idx_usuario (usuario_id)
```

) COMMENT = 'Cambios de email con verificación (RI-004, RN-004)';

-- MÓDULO: GESTIÓN DE PRODUCTOS (ADMIN)
-- Prioridad: CRÍTICA (MUST) - Core del negocio

-- Tabla PRODUCTO (RI-012)

-- 3NF: Atributos base del producto sin dependencias transitivas

```
CREATE TABLE PRODUCTO (  
  ID INT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(100) UNIQUE NOT NULL COMMENT 'Nombre único del producto',  
  descripcion TEXT COMMENT 'Máximo 500 caracteres',  
  precio_base DECIMAL(10,2) NOT NULL COMMENT 'Precio base > 0',  
  fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  activo BOOLEAN DEFAULT TRUE,  
  aprobado BOOLEAN DEFAULT FALSE COMMENT 'Aprobado para catálogo público',  
  fecha_aprobacion DATETIME NULL,  
  usuario_id_creador INT NULL COMMENT 'Quien creó el producto',  
  usuario_id_aprobador INT NULL COMMENT 'Quien aprobó el producto',
```

```
  CONSTRAINT chk_precio_base CHECK (precio_base > 0),  
  CONSTRAINT chk_descripcion_length CHECK (CHAR_LENGTH(descripcion) <= 500),  
  FOREIGN KEY (usuario_id_creador) REFERENCES USUARIO(ID) ON DELETE SET  
NULL,  
  FOREIGN KEY (usuario_id_aprobador) REFERENCES USUARIO(ID) ON DELETE SET  
NULL,
```

```
  INDEX idx_nombre (nombre),  
  INDEX idx_precio (precio_base),  
  INDEX idx_aprobado (aprobado),  
  INDEX idx_activo (activo),
```

```
INDEX idx_fecha_creacion (fecha_creacion)
) COMMENT = 'Productos base del catálogo (RI-012, RN-012)';
```

```
-- Tabla IMAGEN_PRODUCTO (RI-013, RI-014)
-- 3NF: Imágenes asociadas a productos con orden y principal
```

```
CREATE TABLE IMAGEN_PRODUCTO (
  ID INT AUTO_INCREMENT PRIMARY KEY,
  producto_id INT NOT NULL,
  ruta_imagen VARCHAR(500) NOT NULL COMMENT 'Ruta del archivo',
  alt_text VARCHAR(200) COMMENT 'Texto alternativo para accesibilidad',
  principal BOOLEAN DEFAULT FALSE COMMENT 'Imagen principal del producto',
  orden INT DEFAULT 0 COMMENT 'Orden en galería',
  fecha_carga TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
```

```
FOREIGN KEY (producto_id) REFERENCES PRODUCTO(ID) ON DELETE CASCADE,
```

```
INDEX idx_producto (producto_id),
INDEX idx_principal (producto_id, principal),
```

```
-- Constraint: Solo una imagen principal por producto
CONSTRAINT unique_principal_per_product UNIQUE (producto_id, principal)
) COMMENT = 'Imágenes de productos (RI-013, RI-014, RN-013, RN-014)';
```

```
-- Tabla VARIANTE (RI-015, RI-016)
-- 3NF: Variantes de producto (talla + color + stock + precio opcional)
```

```
CREATE TABLE VARIANTE (
  ID INT AUTO_INCREMENT PRIMARY KEY,
  producto_id INT NOT NULL,
  talla ENUM('S', 'M', 'L', 'XL') NOT NULL COMMENT 'Tallas disponibles',
  color_hex VARCHAR(7) NOT NULL COMMENT 'Color en formato #RRGGBB',
  color_nombre VARCHAR(50) COMMENT 'Nombre descriptivo del color',
  stock INT DEFAULT 0 COMMENT 'Stock disponible >= 0',
  precio_variante DECIMAL(10,2) NULL COMMENT 'Precio específico o NULL usa
precio_base',
  fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
```

```
CONSTRAINT chk_stock CHECK (stock >= 0),
CONSTRAINT chk_precio_variante CHECK (precio_variante IS NULL OR precio_variante
> 0),
CONSTRAINT chk_color_hex CHECK (color_hex REGEXP '^#[0-9A-Fa-f]{6}$'),
```

```
FOREIGN KEY (producto_id) REFERENCES PRODUCTO(ID) ON DELETE CASCADE,
```

```
-- Constraint: Combinación única de producto + talla + color
CONSTRAINT unique_variante UNIQUE (producto_id, talla, color_hex),
```

```
INDEX idx_producto (producto_id),
INDEX idx_talla (talla),
```

```

        INDEX idx_color (color_hex),
        INDEX idx_stock (stock)
    ) COMMENT = 'Variantes de productos con talla/color/stock (RI-015, RI-016, RN-015, RN-016)';

```

```

-----
-- MÓDULO: CARRITO Y PEDIDOS
-- Prioridad: CRÍTICA (MUST) - Funcionalidad e-commerce básica
-----

```

```

-- Tabla CARRITO (RI-083)
-- 3NF: Un carrito por usuario activo
CREATE TABLE CARRITO (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    usuario_id INT UNIQUE NOT NULL COMMENT 'Un carrito por usuario',
    fecha_creacion DATETIME DEFAULT CURRENT_TIMESTAMP,
    fecha_actualizacion DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE
    CURRENT_TIMESTAMP,
    estado ENUM('Activo', 'Pendiente', 'Procesado', 'Vacio') DEFAULT 'Activo',

    FOREIGN KEY (usuario_id) REFERENCES USUARIO(ID) ON DELETE CASCADE,
    INDEX idx_usuario (usuario_id),
    INDEX idx_estado (estado)
) COMMENT = 'Carritos de compra de usuarios (RI-083)';

```

```

-- Tabla CARRITO_ITEM (RI-084, RI-085)
-- 3NF: Items individuales del carrito con variante específica
CREATE TABLE CARRITO_ITEM (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    carrito_id INT NOT NULL,
    variante_id INT NOT NULL COMMENT 'Variante específica (talla + color)',
    cantidad INT NOT NULL DEFAULT 1 COMMENT 'Cantidad mínima 1',
    precio_unitario DECIMAL(10,2) NOT NULL COMMENT 'Precio al agregar',
    subtotal DECIMAL(12,2) GENERATED ALWAYS AS (precio_unitario * cantidad)
    STORED,
    fecha_agregado TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    estado_item ENUM('Activo', 'Pendiente', 'Eliminado') DEFAULT 'Activo',

    CONSTRAINT chk_cantidad CHECK (cantidad >= 1),

    FOREIGN KEY (carrito_id) REFERENCES CARRITO(ID) ON DELETE CASCADE,
    FOREIGN KEY (variante_id) REFERENCES VARIANTE(ID) ON DELETE RESTRICT,

    INDEX idx_carrito (carrito_id),
    INDEX idx_variante (variante_id)
) COMMENT = 'Items del carrito con variantes (RI-084, RI-085, RN-084)';

```

```

-- Tabla PEDIDO (RI-088)

```

-- 3NF: Pedidos finalizados con información de pago y envío

```
CREATE TABLE PEDIDO (  
  ID INT AUTO_INCREMENT PRIMARY KEY,  
  usuario_id INT NOT NULL,  
  carrito_id INT NULL COMMENT 'Carrito que originó el pedido',  
  numero_pedido VARCHAR(30) UNIQUE NOT NULL COMMENT 'Número único  
incremental',  
  total DECIMAL(12,2) NOT NULL,  
  metodo_pago VARCHAR(50) DEFAULT 'Wompi' COMMENT 'Método de pago usado',  
  fecha_pedido DATETIME DEFAULT CURRENT_TIMESTAMP,  
  estado ENUM('Pendiente', 'Pagado', 'Produccion', 'Enviado', 'Cancelado') DEFAULT  
'Pendiente',  
  
  FOREIGN KEY (usuario_id) REFERENCES USUARIO(ID),  
  FOREIGN KEY (carrito_id) REFERENCES CARRITO(ID) ON DELETE SET NULL,  
  
  INDEX idx_usuario (usuario_id),  
  INDEX idx_numero_pedido (numero_pedido),  
  INDEX idx_estado (estado),  
  INDEX idx_fecha (fecha_pedido)  
) COMMENT = 'Pedidos procesados (RI-088, RF-088)';
```

-- Tabla DATOS_ENVIO (RI-094)

-- 3NF: Información de envío separada del pedido

```
CREATE TABLE DATOS_ENVIO (  
  ID INT AUTO_INCREMENT PRIMARY KEY,  
  pedido_id INT UNIQUE NOT NULL COMMENT 'Relación 1:1 con pedido',  
  nombre VARCHAR(100) NOT NULL,  
  correo VARCHAR(100) NOT NULL,  
  telefono VARCHAR(20) NOT NULL,  
  direccion VARCHAR(255) NOT NULL,  
  ciudad VARCHAR(100) NOT NULL,  
  codigo_postal VARCHAR(10) NOT NULL,  
  
  FOREIGN KEY (pedido_id) REFERENCES PEDIDO(ID) ON DELETE CASCADE  
) COMMENT = 'Datos de envío de pedidos (RI-094, RN-094)';
```

-- Tabla TRANSACCION (RI-089)

-- 3NF: Transacciones de pago con Wompi

```
CREATE TABLE TRANSACCION (  
  ID INT AUTO_INCREMENT PRIMARY KEY,  
  pedido_id INT NOT NULL,  
  referencia_wompi VARCHAR(255) UNIQUE COMMENT 'Referencia de Wompi',  
  estado_pago ENUM('Pendiente', 'Pagado', 'Fallido', 'Cancelado') DEFAULT 'Pendiente',  
  fecha DATETIME DEFAULT CURRENT_TIMESTAMP,  
  valor_pagado DECIMAL(12,2),  
  metodo_pago_detalle VARCHAR(100) COMMENT 'Tarjeta, PSE, etc.',
```

```

FOREIGN KEY (pedido_id) REFERENCES PEDIDO(ID) ON DELETE CASCADE,

INDEX idx_pedido (pedido_id),
INDEX idx_referencia (referencia_wompi),
INDEX idx_estado (estado_pago)
) COMMENT = 'Transacciones de pago con Wompi (RI-089, RN-089)';

-- Tabla PEDIDO_ITEM
-- 3NF: Items del pedido (copia de carrito al confirmar)
CREATE TABLE PEDIDO_ITEM (
  ID INT AUTO_INCREMENT PRIMARY KEY,
  pedido_id INT NOT NULL,
  variante_id INT NOT NULL,
  cantidad INT NOT NULL,
  precio_unitario DECIMAL(10,2) NOT NULL,
  subtotal DECIMAL(12,2) GENERATED ALWAYS AS (precio_unitario * cantidad)
STORED,

  FOREIGN KEY (pedido_id) REFERENCES PEDIDO(ID) ON DELETE CASCADE,
  FOREIGN KEY (variante_id) REFERENCES VARIANTE(ID) ON DELETE RESTRICT,

  INDEX idx_pedido (pedido_id),
  INDEX idx_variante (variante_id)
) COMMENT = 'Items de pedidos finalizados';

--
=====
=====
-- FASE 2: ENTIDADES SHOULD (Funcionalidad Completa)
-- Prioridad: ALTA (Después del MVP)
--
=====
=====

-----
-- MÓDULO: AUDITORÍA Y ADMINISTRACIÓN
-- Prioridad: ALTA (SHOULD) - Necesario para gestión
-----

-- Tabla LOG_AUDITORIA (RI-008)
-- 3NF: Registro de acciones administrativas
CREATE TABLE LOG_AUDITORIA (
  ID INT AUTO_INCREMENT PRIMARY KEY,
  usuario_admin_id INT NULL COMMENT 'Admin que realizó la acción',
  usuario_afectado_id INT NULL COMMENT 'Usuario afectado por la acción',
  tabla_afectada VARCHAR(100) NOT NULL,
  accion VARCHAR(255) NOT NULL COMMENT 'Descripción de la acción',
  datos_anteriores JSON COMMENT 'Estado anterior en JSON',

```

```
datos_nuevos JSON COMMENT 'Estado nuevo en JSON',
fecha_accion DATETIME DEFAULT CURRENT_TIMESTAMP,
ip_address VARCHAR(45),
```

```
FOREIGN KEY (usuario_admin_id) REFERENCES USUARIO(ID) ON DELETE SET
NULL,
FOREIGN KEY (usuario_afectado_id) REFERENCES USUARIO(ID) ON DELETE SET
NULL,
```

```
INDEX idx_admin (usuario_admin_id),
INDEX idx_afectado (usuario_afectado_id),
INDEX idx_fecha (fecha_accion),
INDEX idx_tabla (tabla_afectada)
) COMMENT = 'Log de auditoría de acciones administrativas (RI-008, RN-008)';
```

```
-- Tabla ESTADO_USUARIO (RI-009)
-- 3NF: Historial de cambios de estado de usuarios
```

```
CREATE TABLE ESTADO_USUARIO (
  ID INT AUTO_INCREMENT PRIMARY KEY,
  usuario_id INT NOT NULL,
  estado_anterior ENUM('Activo', 'Inactivo', 'Bloqueado') NOT NULL,
  estado_nuevo ENUM('Activo', 'Inactivo', 'Bloqueado') NOT NULL,
  motivo TEXT,
  fecha_cambio DATETIME DEFAULT CURRENT_TIMESTAMP,
  admin_id INT NULL COMMENT 'Admin que realizó el cambio',
```

```
FOREIGN KEY (usuario_id) REFERENCES USUARIO(ID) ON DELETE CASCADE,
FOREIGN KEY (admin_id) REFERENCES USUARIO(ID) ON DELETE SET NULL,
```

```
INDEX idx_usuario (usuario_id),
INDEX idx_fecha (fecha_cambio)
) COMMENT = 'Historial de cambios de estado de usuarios (RI-009, RN-009)';
```

```
-- -----
-- MÓDULO: CATEGORÍAS Y DISEÑOS
-- Prioridad: ALTA (SHOULD) - Para personalización 3D
-- -----
```

```
-- Tabla CATEGORIA_DISENO (RI-034)
-- 3NF: Categorías para clasificar diseños
CREATE TABLE CATEGORIA_DISENO (
  ID INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) UNIQUE NOT NULL,
  descripcion TEXT,
  activo BOOLEAN DEFAULT TRUE,
```

```
INDEX idx_nombre (nombre),
INDEX idx_activo (activo)
```


) COMMENT = 'Categorías de diseños del marketplace (RI-034, RN-034)';

-- Tabla DISENO

-- 3NF: Diseños creados por usuarios o predefinidos

```
CREATE TABLE DISENO (  
  ID INT AUTO_INCREMENT PRIMARY KEY,  
  usuario_id INT NULL COMMENT 'Creador del diseño (NULL = diseño predefinido)',  
  categoria_diseno_id INT NULL,  
  nombre VARCHAR(200) NOT NULL,  
  descripcion TEXT,  
  ruta_archivo VARCHAR(500) NOT NULL COMMENT 'Ruta del archivo de diseño',  
  publico BOOLEAN DEFAULT FALSE COMMENT 'Visible en marketplace',  
  aprobado BOOLEAN DEFAULT FALSE COMMENT 'Aprobado para uso público',  
  fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  activo BOOLEAN DEFAULT TRUE,  
  
  FOREIGN KEY (usuario_id) REFERENCES USUARIO(ID) ON DELETE SET NULL,  
  FOREIGN KEY (categoria_diseno_id) REFERENCES CATEGORIA_DISENO(ID) ON  
DELETE SET NULL,  
  
  INDEX idx_usuario (usuario_id),  
  INDEX idx_categoria (categoria_diseno_id),  
  INDEX idx_publico (publico),  
  INDEX idx_aprobado (aprobado)
```

) COMMENT = 'Diseños para personalización de productos';

-- Tabla MOTIVO_DESAPROBACION (RI-032)

-- 3NF: Historial de desaprobaciones con motivos

```
CREATE TABLE MOTIVO_DESAPROBACION (  
  ID INT AUTO_INCREMENT PRIMARY KEY,  
  producto_id INT NULL COMMENT 'Producto desaprobado',  
  diseno_id INT NULL COMMENT 'Diseño desaprobado',  
  motivo VARCHAR(200) NOT NULL,  
  usuario_id_revisor INT NOT NULL,  
  fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  
  FOREIGN KEY (producto_id) REFERENCES PRODUCTO(ID) ON DELETE CASCADE,  
  FOREIGN KEY (diseno_id) REFERENCES DISENO(ID) ON DELETE CASCADE,  
  FOREIGN KEY (usuario_id_revisor) REFERENCES USUARIO(ID) ON DELETE  
CASCADE,  
  
  INDEX idx_producto (producto_id),  
  INDEX idx_diseno (diseno_id)
```

) COMMENT = 'Motivos de desaprobación de productos/diseños (RI-032, RN-032)';

-- -----

-- MÓDULO: PERSONALIZACIÓN 3D

-- Prioridad: ALTA (SHOULD) - Core feature del proyecto

```

-----

-- Tabla MODELO_3D (RI-064)
-- 3NF: Modelos 3D base para camisetas
CREATE TABLE MODELO_3D (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    diseno_id INT NULL COMMENT 'Diseño aplicado al modelo',
    producto_id INT NULL COMMENT 'Producto base del modelo',
    ruta_modelo VARCHAR(500) NOT NULL COMMENT 'Ruta del archivo 3D',
    formato ENUM('GLTF', 'OBJ', 'FBX') DEFAULT 'GLTF',
    variacion VARCHAR(100) COMMENT 'Ej: Talla M - Blanco',
    fecha_render TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    activo BOOLEAN DEFAULT TRUE,

    FOREIGN KEY (diseno_id) REFERENCES DISENO(ID) ON DELETE SET NULL,
    FOREIGN KEY (producto_id) REFERENCES PRODUCTO(ID) ON DELETE CASCADE,

    INDEX idx_diseno (diseno_id),
    INDEX idx_producto (producto_id)
) COMMENT = 'Modelos 3D base para visualización (RI-064, RI-065, RN-064, RN-065)';

-- Tabla CONFIGURACION (RI-075)
-- 3NF: Configuraciones guardadas de personalización 3D
CREATE TABLE CONFIGURACION (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    usuario_id INT NOT NULL,
    modelo_3d_id INT NULL,
    nombre VARCHAR(100) NOT NULL COMMENT 'Nombre de la configuración',
    json_config TEXT NOT NULL COMMENT 'Configuración en JSON',
    fecha_guardado DATETIME DEFAULT CURRENT_TIMESTAMP,

    FOREIGN KEY (usuario_id) REFERENCES USUARIO(ID) ON DELETE CASCADE,
    FOREIGN KEY (modelo_3d_id) REFERENCES MODELO_3D(ID) ON DELETE SET
    NULL,

    INDEX idx_usuario (usuario_id)
) COMMENT = 'Configuraciones guardadas de diseños 3D (RI-075, RN-075)';

--
=====
=====
-- FASE 3: ENTIDADES COULD (Nice to Have)
-- Prioridad: MEDIA-BAJA (Funcionalidades adicionales)
--
=====
=====

-- Tabla CONTACTO (RI-067)

```

```
-- 3NF: Formulario de contacto
CREATE TABLE CONTACTO (
  ID INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  correo_electronico VARCHAR(100) NOT NULL,
  mensaje TEXT NOT NULL,
  fecha_envio DATETIME DEFAULT CURRENT_TIMESTAMP,
  estado ENUM('Pendiente', 'Respondido', 'Archivado') DEFAULT 'Pendiente',

  CONSTRAINT chk_mensaje_length CHECK (CHAR_LENGTH(mensaje) BETWEEN 10
AND 1000),

  INDEX idx_estado (estado),
  INDEX idx_fecha (fecha_envio)
) COMMENT = 'Consultas de contacto (RI-067, RN-067, RN-069)';
```

```
-- Tabla CALIFICACION
-- 3NF: Calificaciones de productos por clientes
CREATE TABLE CALIFICACION (
  ID INT AUTO_INCREMENT PRIMARY KEY,
  usuario_id INT NOT NULL,
  producto_id INT NOT NULL,
  estrellas TINYINT NOT NULL COMMENT 'Calificación 1-5',
  comentario TEXT,
  fecha_calificacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

  CONSTRAINT chk_estrellas CHECK (estrellas BETWEEN 1 AND 5),
  CONSTRAINT unique_calificacion UNIQUE (usuario_id, producto_id),

  FOREIGN KEY (usuario_id) REFERENCES USUARIO(ID) ON DELETE CASCADE,
  FOREIGN KEY (producto_id) REFERENCES PRODUCTO(ID) ON DELETE CASCADE,

  INDEX idx_producto (producto_id),
  INDEX idx_estrellas (estrellas)
) COMMENT = 'Calificaciones de productos (RF-050, RF-054)';
```

```
-- Tabla HISTORIAL_VISUALIZACION (RI-047)
-- 3NF: Registro de visualizaciones de productos
CREATE TABLE HISTORIAL_VISUALIZACION (
  ID INT AUTO_INCREMENT PRIMARY KEY,
  usuario_id INT NULL COMMENT 'NULL si no está logueado',
  producto_id INT NOT NULL,
  fecha_visualizacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  duracion_segundos INT NULL,

  FOREIGN KEY (usuario_id) REFERENCES USUARIO(ID) ON DELETE SET NULL,
  FOREIGN KEY (producto_id) REFERENCES PRODUCTO(ID) ON DELETE CASCADE,
```

```

    INDEX idx_usuario (usuario_id),
    INDEX idx_producto (producto_id),
    INDEX idx_fecha (fecha_visualizacion)
) COMMENT = 'Historial de visualizaciones para analytics (RI-047, RN-047)';

-- Tabla FACTURA_SIMPLE (RI-093)
-- 3NF: Facturas simples de pedidos
CREATE TABLE FACTURA_SIMPLE (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    pedido_id INT UNIQUE NOT NULL,
    fecha_generada DATETIME DEFAULT CURRENT_TIMESTAMP,
    total_final DECIMAL(12,2) NOT NULL,
    ruta_pdf VARCHAR(500) COMMENT 'Ruta del PDF generado',

    FOREIGN KEY (pedido_id) REFERENCES PEDIDO(ID) ON DELETE CASCADE,

    INDEX idx_pedido (pedido_id)
) COMMENT = 'Facturas simples de pedidos (RI-093, RN-093)';

--
=====
=====
-- TRIGGERS PARA MANTENER INTEGRIDAD Y AUTOMATIZACIÓN
--
=====
=====

DELIMITER //

-- Trigger: Validar stock antes de agregar al carrito
CREATE TRIGGER tr_validar_stock_carrito
BEFORE INSERT ON CARRITO_ITEM
FOR EACH ROW
BEGIN
    DECLARE stock_disponible INT;

    SELECT stock INTO stock_disponible
    FROM VARIANTE
    WHERE ID = NEW.variante_id;

    IF NEW.cantidad > stock_disponible THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Stock insuficiente para la variante seleccionada';
    END IF;
END//

-- Trigger: Actualizar stock al confirmar pedido
CREATE TRIGGER tr_actualizar_stock_pedido

```

```

AFTER INSERT ON PEDIDO_ITEM
FOR EACH ROW
BEGIN
    UPDATE VARIANTE
    SET stock = stock - NEW.cantidad
    WHERE ID = NEW.variante_id;
END//

```

```

-- Trigger: Calcular calificación promedio de productos
CREATE TRIGGER tr_calificacion_promedio
AFTER INSERT ON CALIFICACION
FOR EACH ROW
BEGIN
    -- Aquí podrías agregar lógica para actualizar una tabla de estadísticas
    -- Por ahora se calcula dinámicamente con queries
    NULL;
END//

```

```

-- Trigger: Registrar cambios de estado de usuario en LOG_AUDITORIA
CREATE TRIGGER tr_log_cambio_estado_usuario
AFTER UPDATE ON USUARIO
FOR EACH ROW
BEGIN
    IF OLD.estado != NEW.estado THEN
        INSERT INTO LOG_AUDITORIA (
            usuario_afectado_id,
            tabla_afectada,
            accion,
            datos_anteriores,
            datos_nuevos,
            fecha_accion
        ) VALUES (
            NEW.ID,
            'USUARIO',
            CONCAT('Cambio de estado: ', OLD.estado, ' -> ', NEW.estado),
            JSON_OBJECT('estado', OLD.estado),
            JSON_OBJECT('estado', NEW.estado),
            NOW()
        );
    END IF;
END//

```

```

DELIMITER ;

```

```

--
=====
=====
-- DATOS DE PRUEBA INICIALES

```

```

--
=====

=====

-- Insertar usuario administrador inicial
INSERT INTO USUARIO (usuario, correo, contrasena, rol, estado, email_verificado)
VALUES
('admin', 'admin@red.com', SHA2('Admin123!', 256), 'Administrador', 'Activo', TRUE),
('revisor', 'revisor@red.com', SHA2('Revisor123!', 256), 'Revisor', 'Activo', TRUE),
('cliente1', 'cliente1@test.com', SHA2('Cliente123!', 256), 'Usuario', 'Activo', TRUE);

-- Insertar categorías de diseño básicas
INSERT INTO CATEGORIA_DISENO (nombre, descripcion) VALUES
('Abstracto', 'Diseños abstractos y artísticos'),
('Tipografía', 'Diseños basados en texto'),
('Naturaleza', 'Diseños inspirados en la naturaleza'),
('Gaming', 'Diseños relacionados con videojuegos'),
('Minimalista', 'Diseños simples y minimalistas');

-- Insertar productos base
INSERT INTO PRODUCTO (nombre, descripcion, precio_base, activo, aprobado,
usuario_id_creador) VALUES
('Camiseta Básica Blanca', 'Camiseta 100% algodón, perfecta para personalizar', 45000.00,
TRUE, TRUE, 1),
('Camiseta Básica Negra', 'Camiseta 100% algodón color negro', 45000.00, TRUE, TRUE,
1),
('Camiseta Premium', 'Camiseta de alta calidad con mejor acabado', 65000.00, TRUE,
TRUE, 1);

-- Insertar imágenes de productos
INSERT INTO IMAGEN_PRODUCTO (producto_id, ruta_imagen, alt_text, principal, orden)
VALUES
(1, '/images/productos/camiseta-blanca-frontal.jpg', 'Camiseta blanca vista frontal', TRUE,
1),
(1, '/images/productos/camiseta-blanca-trasera.jpg', 'Camiseta blanca vista trasera', FALSE,
2),
(2, '/images/productos/camiseta-negra-frontal.jpg', 'Camiseta negra vista frontal', TRUE, 1),
(3, '/images/productos/camiseta-premium-frontal.jpg', 'Camiseta premium vista frontal',
TRUE, 1);

-- Insertar variantes (tallas y colores)
INSERT INTO VARIANTE (producto_id, talla, color_hex, color_nombre, stock) VALUES
-- Camiseta Básica Blanca
(1, 'S', '#FFFFFF', 'Blanco', 50),
(1, 'M', '#FFFFFF', 'Blanco', 75),
(1, 'L', '#FFFFFF', 'Blanco', 60),
(1, 'XL', '#FFFFFF', 'Blanco', 40),
-- Camiseta Básica Negra

```

```

(2, 'S', '#000000', 'Negro', 45),
(2, 'M', '#000000', 'Negro', 80),
(2, 'L', '#000000', 'Negro', 55),
(2, 'XL', '#000000', 'Negro', 35),
-- Camiseta Premium (múltiples colores)
(3, 'S', '#FFFFFF', 'Blanco', 30),
(3, 'M', '#FFFFFF', 'Blanco', 45),
(3, 'L', '#FFFFFF', 'Blanco', 35),
(3, 'S', '#000000', 'Negro', 25),
(3, 'M', '#000000', 'Negro', 40),
(3, 'L', '#000000', 'Negro', 30);

```

```

--
=====
=====
-- VISTAS ÚTILES PARA CONSULTAS FRECUENTES
--
=====
=====

```

```

-- Vista: Productos con stock total y precio mínimo
CREATE VIEW vista_productos_catalogo AS
SELECT
    p.ID,
    p.nombre,
    p.descripcion,
    p.precio_base,
    p.aprobado,
    p.activo,
    COALESCE(MIN(v.precio_variante), p.precio_base) AS precio_minimo,
    SUM(v.stock) AS stock_total,
    i.ruta_imagen AS imagen_principal,
    COALESCE(AVG(c.estrellas), 0) AS calificacion_promedio,
    COUNT(DISTINCT c.ID) AS num_calificaciones
FROM PRODUCTO p
LEFT JOIN VARIANTE v ON p.ID = v.producto_id
LEFT JOIN IMAGEN_PRODUCTO i ON p.ID = i.producto_id AND i.principal = TRUE
LEFT JOIN CALIFICACION c ON p.ID = c.producto_id
WHERE p.activo = TRUE AND p.aprobado = TRUE
GROUP BY p.ID, p.nombre, p.descripcion, p.precio_base, p.aprobado, p.activo,
i.ruta_imagen;

```

```

-- Vista: Carrito con detalles completos
CREATE VIEW vista_carrito_detalle AS
SELECT
    ci.ID AS item_id,
    ci.carrito_id,
    c.usuario_id,

```

```

    p.ID AS producto_id,
    p.nombre AS producto_nombre,
    v.ID AS variante_id,
    v.talla,
    v.color_nombre,
    ci.cantidad,
    ci.precio_unitario,
    ci.subtotal,
    i.ruta_imagen AS imagen_producto
FROM CARRITO_ITEM ci
JOIN CARRITO c ON ci.carrito_id = c.ID
JOIN VARIANTE v ON ci.variante_id = v.ID
JOIN PRODUCTO p ON v.producto_id = p.ID
LEFT JOIN IMAGEN_PRODUCTO i ON p.ID = i.producto_id AND i.principal = TRUE
WHERE ci.estado_item = 'Activo' AND c.estado = 'Activo';

```

-- Vista: Pedidos con resumen

```
CREATE VIEW vista_pedidos_resumen AS
```

```
SELECT
```

```

    ped.ID AS pedido_id,
    ped.numero_pedido,
    ped.usuario_id,
    u.usuario AS nombre_usuario,
    u.correo AS correo_usuario,
    ped.total,
    ped.estado AS estado_pedido,
    ped.fecha_pedido,
    COUNT(pi.ID) AS cantidad_items,
    t.estado_pago,
    t.referencia_wompi,
    de.ciudad,
    de.direccion

```

```
FROM PEDIDO ped
```

```
JOIN USUARIO u ON ped.usuario_id = u.ID
```

```
LEFT JOIN PEDIDO_ITEM pi ON ped.ID = pi.pedido_id
```

```
LEFT JOIN TRANSACCION t ON ped.ID = t.pedido_id
```

```
LEFT JOIN DATOS_ENVIO de ON ped.ID = de.pedido_id
```

```

GROUP BY ped.ID, ped.numero_pedido, ped.usuario_id, u.usuario, u.correo,
    ped.total, ped.estado, ped.fecha_pedido, t.estado_pago,
    t.referencia_wompi, de.ciudad, de.direccion;

```

```
SET FOREIGN_KEY_CHECKS = 1;
```

```
--
```

```
=====
```

```
=====
```

```
-- FIN DEL SCRIPT
```


--

=====

/*

RESUMEN DE PRIORIZACIÓN (MoSCoW + Eisenhower):

FASE 1 - MVP (MUST + Urgente/Importante):

- ✓ Usuario, Sesión, Token_Verificacion, Cambio_Email
- ✓ Producto, Imagen_Producto, Variante
- ✓ Carrito, Carrito_Item, Pedido, Pedido_Item
- ✓ Datos_Envio, Transaccion

FASE 2 - Funcionalidad Completa (SHOULD + Importante):

- ✓ Log_Auditoria, Estado_Usuario
- ✓ Categoria_Disenio, Diseno, Motivo_Desaprobacion
- ✓ Modelo_3D, Configuracion

FASE 3 - Features Adicionales (COULD + Nice to Have):

- ✓ Contacto
- ✓ Calificacion
- ✓ Historial_Visualizacion
- ✓ Factura_Simple

EXCLUIDAS (WON'T):

- ✗ Sistema de gamificación (nivel_vip, puntos, recompensas)
- ✗ Sistema complejo de descuentos generales
- ✗ Múltiples proveedores de envío
- ✗ Analytics avanzado integrado

NORMALIZACIÓN:

- ✓ 1NF: Todos los atributos son atómicos
- ✓ 2NF: No hay dependencias parciales
- ✓ 3NF: No hay dependencias transitivas

TOTAL DE TABLAS: 25 tablas principales + 3 vistas

- Core (MVP): 13 tablas
- Completo: 18 tablas
- Adicional: 7 tablas

*/