Projeto de Banco de Dados - Loja Virtual (Versão Corrigida)

# 1. Criação das Tabelas (Estoque como DECIMAL)

CREATE DATABASE loja;

USE loja;

-- CLIENTES  
CREATE TABLE clientes (  
 id\_cliente INT PRIMARY KEY AUTO\_INCREMENT,  
 nome VARCHAR(100),  
 email VARCHAR(100),  
 telefone VARCHAR(20),  
 endereco VARCHAR(150)  
);  
  
-- CATEGORIAS  
CREATE TABLE categorias (  
 id\_categoria INT PRIMARY KEY AUTO\_INCREMENT,  
 nome\_categoria VARCHAR(50)  
);  
  
-- PRODUTOS  
CREATE TABLE produtos (  
 id\_produto INT PRIMARY KEY AUTO\_INCREMENT,  
 nome\_produto VARCHAR(100),  
 preco DECIMAL(10,2),  
 estoque\_produto DECIMAL,  
 id\_categoria INT,  
 FOREIGN KEY (id\_categoria) REFERENCES categorias(id\_categoria)  
);  
  
-- USUÁRIOS  
CREATE TABLE usuarios (  
 id\_usuario INT PRIMARY KEY AUTO\_INCREMENT,  
 nome\_usuario VARCHAR(100),  
 login VARCHAR(50),  
 senha VARCHAR(100)  
);  
  
-- VENDAS  
CREATE TABLE vendas (  
 id\_venda INT PRIMARY KEY AUTO\_INCREMENT,  
 id\_cliente INT,  
 id\_usuario INT,  
 data\_venda DATETIME DEFAULT CURRENT\_TIMESTAMP,  
 valor\_total DECIMAL(10,2),  
 FOREIGN KEY (id\_cliente) REFERENCES clientes(id\_cliente),  
 FOREIGN KEY (id\_usuario) REFERENCES usuarios(id\_usuario)  
);  
  
-- ITENS DA VENDA  
CREATE TABLE itens\_venda (  
 id\_item INT PRIMARY KEY AUTO\_INCREMENT,  
 id\_venda INT,  
 id\_produto INT,  
 quantidade INT,  
 preco\_unitario DECIMAL(10,2),  
 FOREIGN KEY (id\_venda) REFERENCES vendas(id\_venda),  
 FOREIGN KEY (id\_produto) REFERENCES produtos(id\_produto)  
);

# 2. INSERTs de Exemplo

-- CLIENTES  
INSERT INTO clientes (nome, email, telefone, endereco) VALUES  
('Ana Paula', 'ana.paula@email.com', '(11) 98888-1234', 'Rua das Rosas, 123'),  
('Carlos Souza', 'carlos.souza@email.com', '(11) 97777-4321', 'Av. Brasil, 456'),  
('Juliana Lima', 'juliana.lima@email.com', '(11) 96666-5678', 'Rua Verde, 789');  
  
-- CATEGORIAS  
INSERT INTO categorias (nome\_categoria) VALUES  
('Roupas'),  
('Calçados'),  
('Acessórios');  
  
-- PRODUTOS  
INSERT INTO produtos (nome\_produto, preco, estoque\_produto, id\_categoria) VALUES  
('Camiseta Branca', 49.90, 20, 1),  
('Tênis Esportivo', 199.90, 15, 2),  
('Relógio Digital', 99.90, 10, 3);  
  
-- USUÁRIOS  
INSERT INTO usuarios (nome\_usuario, login, senha) VALUES  
('Maria Vendedora', 'maria.vend', 'senha123'),  
('João Atendente', 'joao.atend', 'senha456');

SELECT \* FROM clientes;

-- SELECT \* FROM categorias;

-- SELECT \* FROM produtos;

-- SELECT \* FROM usuarios;

# 3. Procedure para Registrar Venda

DELIMITER $$  
  
CREATE PROCEDURE registrar\_venda (  
 IN p\_id\_cliente INT,  
 IN p\_id\_usuario INT,  
 IN p\_valor\_total DECIMAL(10,2)  
)  
BEGIN  
 INSERT INTO vendas (id\_cliente, id\_usuario, valor\_total)  
 VALUES (p\_id\_cliente, p\_id\_usuario, p\_valor\_total);  
  
 SET @id\_venda := LAST\_INSERT\_ID();  
END$$  
  
DELIMITER ;

# 4. Triggers

-- Deduzir estoque  
DELIMITER $$  
CREATE TRIGGER tr\_deduzir\_estoque  
AFTER INSERT ON itens\_venda  
FOR EACH ROW  
BEGIN  
 UPDATE produtos  
 SET estoque\_produto = estoque\_produto - NEW.quantidade  
 WHERE id\_produto = NEW.id\_produto;  
END$$  
DELIMITER ;  
  
-- Impedir venda com estoque insuficiente  
DELIMITER $$  
CREATE TRIGGER tr\_verificar\_estoque  
BEFORE INSERT ON itens\_venda  
FOR EACH ROW  
BEGIN  
 DECLARE estoque\_atual DECIMAL;  
  
 SELECT estoque\_produto INTO estoque\_atual  
 FROM produtos  
 WHERE id\_produto = NEW.id\_produto;  
  
 IF estoque\_atual < NEW.quantidade THEN  
 SIGNAL SQLSTATE '45000'  
 SET MESSAGE\_TEXT = 'Estoque insuficiente para esse produto.';  
 END IF;  
END$$  
DELIMITER ;

# 5. Views (Com Estoque como DECIMAL)

-- Estoque baixo  
CREATE VIEW vw\_estoque\_baixo AS  
SELECT nome\_produto, estoque\_produto  
FROM produtos  
WHERE CAST(estoque\_produto AS UNSIGNED) <= 5;  
  
-- Vendas por cliente  
CREATE VIEW vw\_vendas\_por\_cliente AS  
SELECT c.nome, COUNT(v.id\_venda) AS total\_vendas, SUM(v.valor\_total) AS total\_gasto  
FROM clientes c  
JOIN vendas v ON c.id\_cliente = v.id\_cliente  
GROUP BY c.id\_cliente;  
  
-- Faturamento mensal  
CREATE VIEW vw\_faturamento\_mensal AS  
SELECT DATE\_FORMAT(data\_venda, '%Y-%m') AS mes, SUM(valor\_total) AS faturamento  
FROM vendas  
GROUP BY mes  
ORDER BY mes;