Criar um banco de dados chamado: sistema\_vendas e rodar o script abaixo:

-- Cria banco de dados

CREATE DATABASE sistema\_vendas;

USE sistema\_vendas;

-- Tabela de clientes  
CREATE TABLE clientes (

    id\_cliente INT AUTO\_INCREMENT PRIMARY KEY,

    nome VARCHAR(100) NOT NULL,

    email VARCHAR(100) UNIQUE,

    data\_nascimento DATE,

CHECK (YEAR(data\_nascimento) <= YEAR(CURDATE()) – 18)

);

-- Tabela de produtos  
CREATE TABLE produtos (  
    id\_produto INT AUTO\_INCREMENT PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    preco DECIMAL(10,2) NOT NULL CHECK (preco > 0),  
    estoque INT DEFAULT 0 CHECK (estoque >= 0)  
);

-- Tabela de pedidos  
CREATE TABLE pedidos (  
    id\_pedido INT AUTO\_INCREMENT PRIMARY KEY,  
    id\_cliente INT,  
    data\_pedido DATETIME DEFAULT CURRENT\_TIMESTAMP,  
    FOREIGN KEY (id\_cliente) REFERENCES clientes(id\_cliente)  
);

-- Tabela de itens do pedido  
CREATE TABLE itens\_pedido (  
    id\_item INT AUTO\_INCREMENT PRIMARY KEY,  
    id\_pedido INT,  
    id\_produto INT,  
    quantidade INT CHECK (quantidade > 0),  
    preco\_unitario DECIMAL(10,2),  
    FOREIGN KEY (id\_pedido) REFERENCES pedidos(id\_pedido),  
    FOREIGN KEY (id\_produto) REFERENCES produtos(id\_produto)  
);

-- VIEW: Pedidos com total  
CREATE VIEW view\_pedidos\_totais AS  
SELECT   
    p.id\_pedido,  
    c.nome AS cliente,  
    p.data\_pedido,  
    SUM(i.quantidade \* i.preco\_unitario) AS total\_pedido  
FROM pedidos p  
JOIN clientes c ON p.id\_cliente = c.id\_cliente  
JOIN itens\_pedido i ON p.id\_pedido = i.id\_pedido  
GROUP BY p.id\_pedido, c.nome, p.data\_pedido;

-- TRIGGER: Atualiza estoque após inserção de item  
DELIMITER //  
CREATE TRIGGER trg\_atualiza\_estoque  
AFTER INSERT ON itens\_pedido  
FOR EACH ROW  
BEGIN  
    UPDATE produtos  
    SET estoque = estoque - NEW.quantidade  
    WHERE id\_produto = NEW.id\_produto;  
END;  
//  
DELIMITER ;

-- FUNCTION: Calcular idade do cliente  
DELIMITER //  
CREATE FUNCTION calcular\_idade(data\_nasc DATE)  
RETURNS INT  
DETERMINISTIC  
RETURN TIMESTAMPDIFF(YEAR, data\_nasc, CURDATE());  
//  
DELIMITER ;

DELIMITER //

CREATE TRIGGER validar\_idade

BEFORE INSERT ON clientes

FOR EACH ROW

BEGIN

IF TIMESTAMPDIFF(YEAR, NEW.data\_nascimento, CURDATE()) < 18 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Cliente deve ter pelo menos 18 anos.';

END IF;

END

//

DELIMITER ;

Inserindo dados no banco:

INSERT INTO clientes (nome,email,data\_nascimento) VALUES ('Maria Rosa','maria.rosa@gmail.com','2000-10-03');

INSERT INTO clientes (nome,email,data\_nascimento) VALUES ('Hugo Tavares','hugo.tavares@outlook.com','1998-04-06');

INSERT INTO clientes (nome,email,data\_nascimento) VALUES ('Tales Soares','tales.soares@aol.com','2014-06-09');

# Mensagem do MySQL: Documentação

#1644 - Cliente deve ter pelo menos 18 anos.

INSERT INTO produtos (nome,preco,estoque) VALUES ('Sabonete',2,100);

INSERT INTO produtos (nome,preco,estoque) VALUES ('Maçã',1,200);

INSERT INTO pedidos (id\_cliente,data\_pedido) VALUES (1, '2025-08-06');

INSERT INTO pedidos (id\_cliente,data\_pedido) VALUES (2, '2025-08-06 01:22:38');

INSERT INTO itens\_pedido (id\_pedido,id\_produto,quantidade,preco\_unitario) VALUES (1,1,20,2);

INSERT INTO itens\_pedido (id\_pedido,id\_produto,quantidade,preco\_unitario) VALUES (2,2,50,1);

Selecionando VIEW:

USE sistema\_vendas;

SELECT \* FROM clientes;

SELECT \* FROM pedidos;

SELECT \* FROM produtos;

SELECT \* FROM view\_pedidos\_totais;