

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;
```