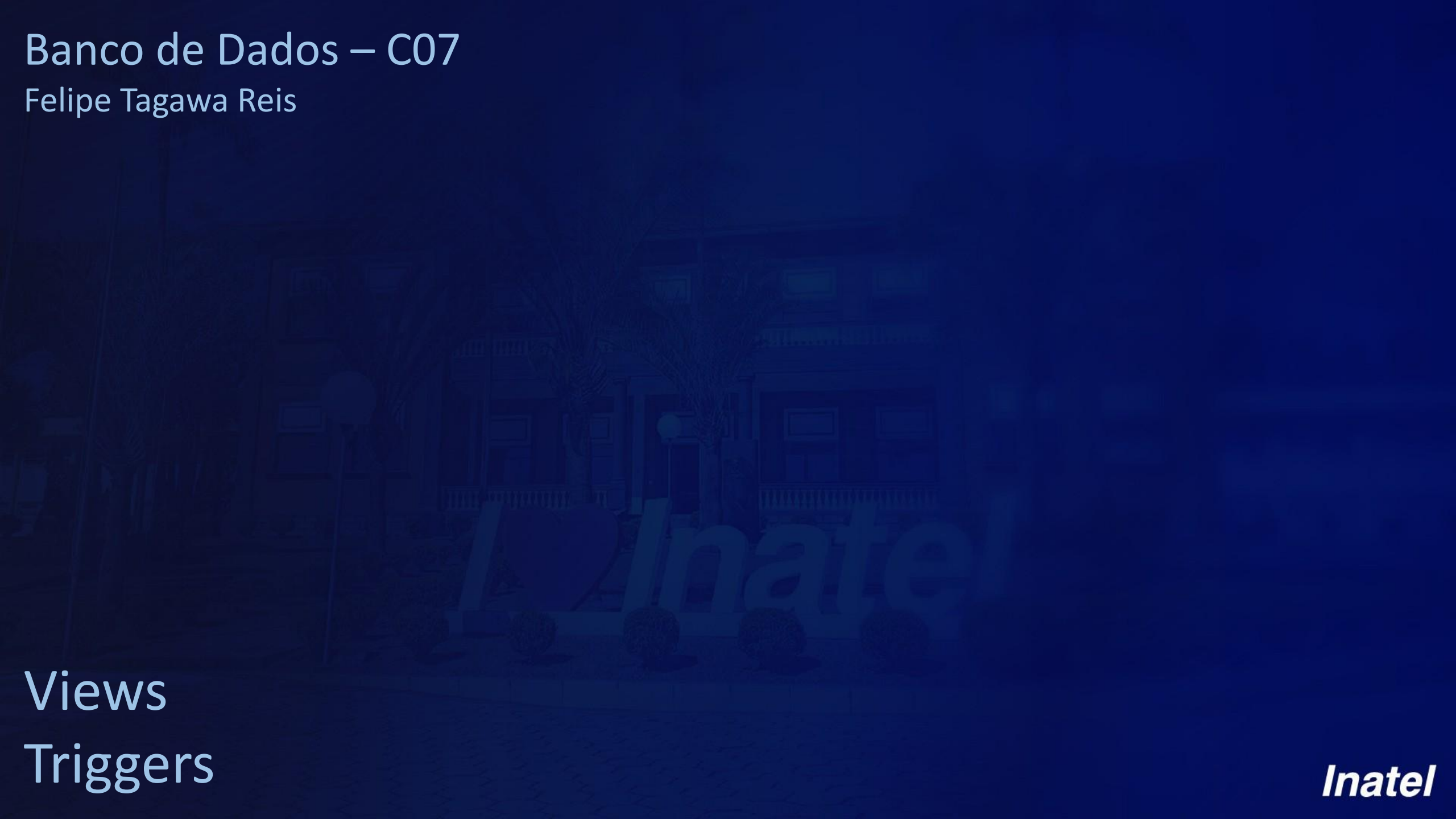


# Banco de Dados – C07

Felipe Tagawa Reis



Views

Triggers

# Triggers

definição

Triggers são estruturas utilizadas para executar verificações de valores ou fazer cálculos sobre os valores informados mediante ativação de comandos (Insert, Update ou Delete) em uma tabela associada;

O gatilho pode ser ativado antes ou depois da declaração ser realizada.

# Triggers

estrutura

```
CREATE [Definer = {user | CURRENT_USER}] TRIGGER (triggerName)(triggerTime)  
(triggerEvent) ON (tableName) FOR EACH ROW (triggerStatement)
```

# Triggers

estrutura

Opcional, mas não será utilizado

```
CREATE [Definer = {user | CURRENT_USER}] TRIGGER (triggerName)(triggerTime)  
(triggerEvent) ON (tableName) FOR EACH ROW (triggerStatement)
```

# Triggers

estrutura

Definição do  
nome do Trigger

Opcional, mas não será utilizado

~~CREATE [Definer = {user | CURRENT\_USER}]~~ **TRIGGER** (triggerName)(triggerTime)  
(triggerEvent) **ON** (tableName) **FOR EACH ROW** (triggerStatement)

# Triggers

estrutura

Definição do  
nome do Trigger

Define a ativação  
do Trigger(BEFORE  
ou AFTER)

Opcional, mas não será utilizado

```
CREATE [Definer = {user | CURRENT_USER}] TRIGGER (triggerName)(triggerTime)  
(triggerEvent) ON (tableName) FOR EACH ROW (triggerStatement)
```

# Triggers

estrutura

Definição do  
nome do Trigger

Define a ativação  
do Trigger(BEFORE  
ou AFTER)

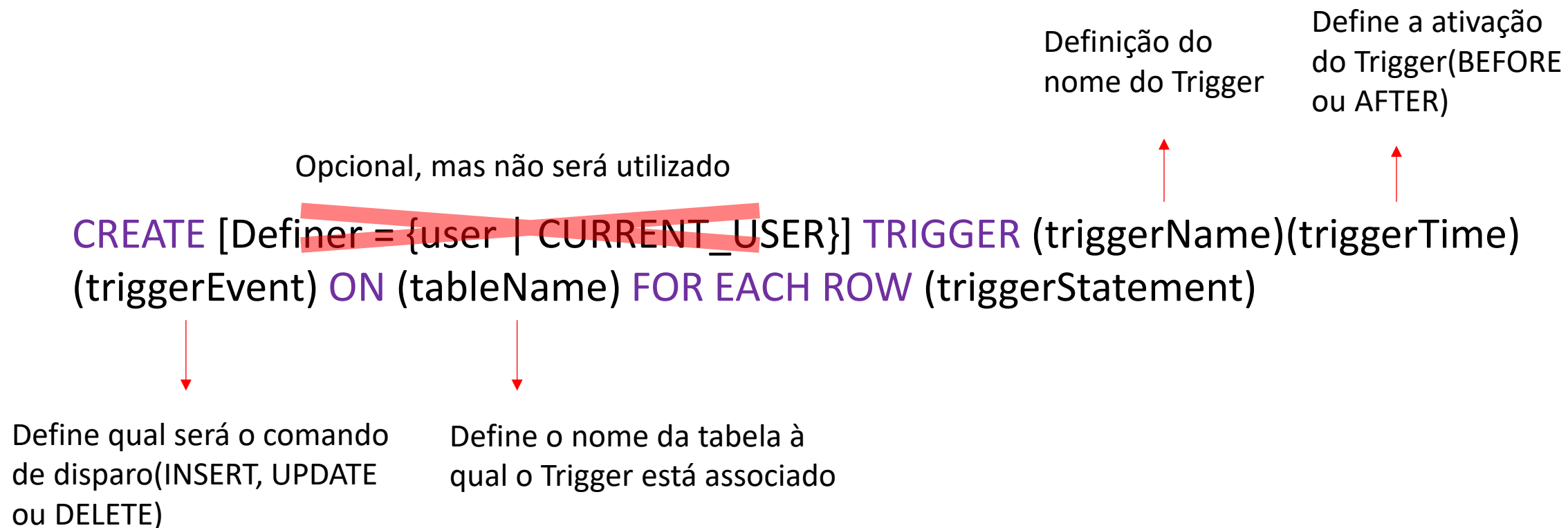
Opcional, mas não será utilizado

```
CREATE [Definer = {user | CURRENT_USER}] TRIGGER (triggerName)(triggerTime)  
(triggerEvent) ON (tableName) FOR EACH ROW (triggerStatement)
```

Define qual será o comando  
de disparo(INSERT, UPDATE  
ou DELETE)

# Triggers

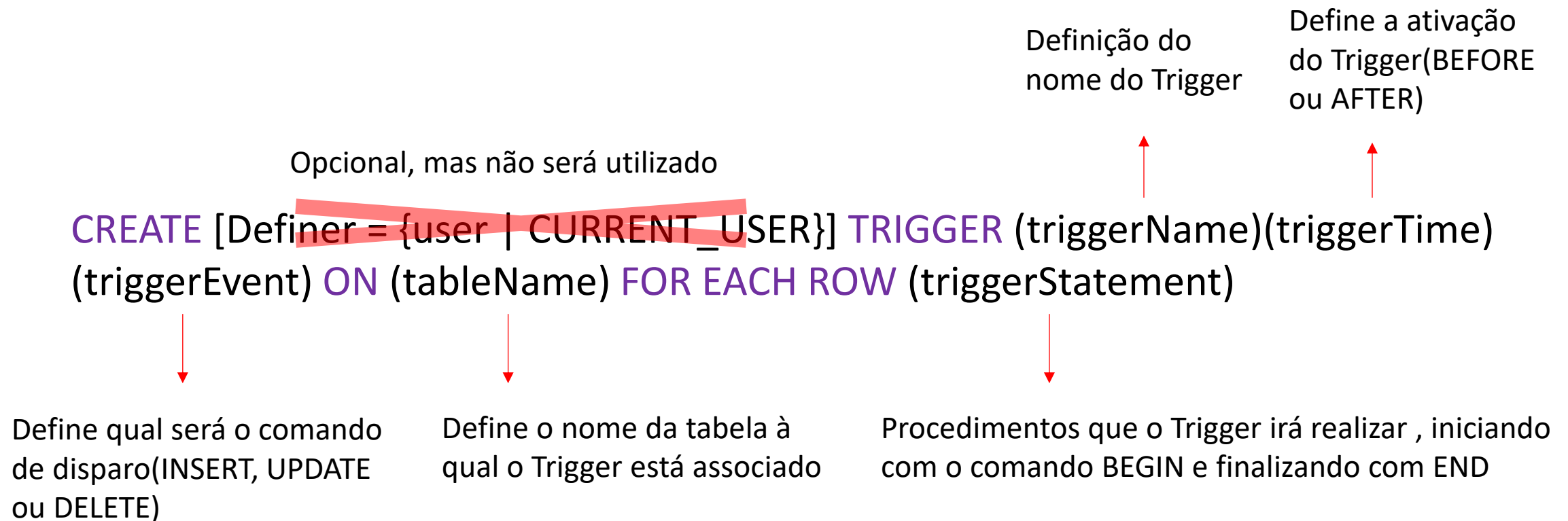
estrutura





# Triggers

estrutura



# Triggers

exemplos

```
-- Criação da tabela conta
CREATE TABLE IF NOT EXISTS conta(
    numero INT PRIMARY KEY,
    total FLOAT
);

-- Trigger para calcular produto antes de inserção
CREATE TRIGGER int_produto BEFORE INSERT
ON conta
FOR EACH ROW SET @produto = @produto * NEW.total;

-- Inicialização da variável produto (valor neutro para
multiplicação)
SET @produto = 1;

-- Inserção de dados de teste
INSERT INTO conta() VALUES
(1,20),
(2,180),
(3,-50);

-- Verificação do resultado
SELECT @produto AS 'Produto Total';
```

Comandos OLD e NEW:

OLD – Usado para recuperar o valor de um campo antes de um UPDATE ou DELETE;

NEW – Usado para recuperar o valor de um campo depois de um UPDATE ou INSERT.

# Triggers

exemplos

```
DELIMITER $$

CREATE TRIGGER desconto_check BEFORE UPDATE
ON conta
FOR EACH ROW
BEGIN

    IF NEW.total < 50 THEN
        SET NEW.total = NEW.total * 0.9; -- 10% desconto
    ELSEIF NEW.total > 150 THEN
        SET NEW.total = NEW.total * 0.8; -- 20% desconto
    END IF;

END; $$
DELIMITER ;

-- Testando o sistema de desconto
UPDATE conta SET total = 40 WHERE numero = 1; -- Desconto
10%
UPDATE conta SET total = 200 WHERE numero = 2; -- Desconto
20%

SELECT * FROM conta;
```

Número	Total Original	Total com Desconto
1	R\$ 40,00	R\$ 36,00
2	R\$ 200,00	R\$ 160,00
3	R\$ 100,00	R\$ 100,00

# Views

definição

Views são estruturas que simplificam a especificação de certas consultas, de forma a evitar que alterações indevidas ocorram diretamente nas tabelas

# Views

estrutura

```
CREATE VIEW nomeView AS (SELECT * FROM nomeTabela);  
DROP VIEW nomeView;
```

# Views

exemplo



vendas



id\_venda INT(11)



cod\_produto INT(11)



nome\_cliente VARCHAR(30)



quantidade INT(11)



produtos



cod\_produto INT(11)



nome\_produto VARCHAR(25)



preco DECIMAL(10,2)

Cliente	Produto	Quantidade	Preço Unitário	Total da Venda
João Silva	Notebook Dell	2	R\$ 2.500,00	R\$ 5.000,00
Maria Santos	Mouse Gamer	5	R\$ 120,00	R\$ 600,00
Carlos Lima	Teclado RGB	3	R\$ 180,00	R\$ 540,00
Ana Costa	Monitor 24"	1	R\$ 850,00	R\$ 850,00

```
CREATE VIEW vendas_detalhadas AS (  
  SELECT  
    v.nome_cliente AS 'Cliente',  
    p.nome_produto AS 'Produto',  
    v.quantidade AS 'Quantidade',  
    p.preco AS 'Preço Unitário',  
    (v.quantidade * p.preco) AS 'Total da Venda'  
  FROM  
    vendas v,  
    produtos p  
  WHERE  
    v.cod_produto = p.cod_produto  
);
```

*-- Consultando a view*

```
SELECT * FROM vendas_detalhadas;
```

# Exercício

```
DROP DATABASE IF EXISTS loja;
CREATE DATABASE loja;
USE loja;

SET GLOBAL log_bin_trust_function_creators = 1; -- Para
permitir a criação de triggers

CREATE TABLE compra(
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    preco FLOAT,
    pagamento FLOAT,
    troco FLOAT
);

INSERT INTO compra(preco, pagamento) VALUES(9.5, 10.25);
INSERT INTO compra(preco, pagamento) VALUES(25.0, 25.0);
INSERT INTO compra(preco, pagamento) VALUES(5.99, 5.99);
INSERT INTO compra(preco, pagamento) VALUES(10.99, 10.99);
INSERT INTO compra(preco, pagamento) VALUES(15.99, 15.99);
```

- Crie um Trigger que seja executado antes do INSERT , que calcule e atribua o valor do troco (valor pago - preço);
- Crie uma View que busca a quantidade de compras, usando COUNT(), cujo preço é maior ou igual a 10.

AULA  
CONCLUÍDA!