

# AULA 15

## UTILIZANDO TRIGGERS PARA CONTROLE DE ESTOQUE

Utilizar o BD – trig\_06\_10.bak

## Exercício

- Ao efetuar uma venda, atualizar a quantidade na tabela estoque e verificar se atingiu o nível mínimo. Se atingir, exibir uma mensagem de estoque baixo e não permitir quantidade acima do limite de estoque.

# SOLUÇÃO – CRIANDO A FUNÇÃO

```
CREATE OR REPLACE FUNCTION atualiza_estoque()  
  RETURNS trigger AS  
$BODY$  
DECLARE  
  qt integer;  
  tot integer;  
  tot_estoque integer;  
  est_min integer;  
  id bigint;
```

# SOLUÇÃO – CRIANDO A FUNÇÃO

BEGIN

qt := new.qtdade;

id := new.fk\_iditem;

tot := estoque.qtdade - new.qtdade FROM estoque WHERE estoque.iditem = id;

tot\_estoque := estoque.qtdade FROM estoque WHERE estoque.iditem = id;

est\_min := estoque.qt\_min FROM estoque WHERE estoque.iditem = id;

IF tot >= 0 THEN

UPDATE estoque SET qtdade = qtdade - qt WHERE iditem = id;

IF tot < est\_min THEN

RAISE NOTICE 'ESTOQUE ABAIXO DO LIMITE!!!';

END IF;

# SOLUÇÃO – CRIANDO A FUNÇÃO

ELSE

RAISE NOTICE 'VALOR ACIMA DO LIMITE DE ESTOQUE ,  
QUANTIDADE PARA VENDA ATUALIZADO PARA (%)', tot\_estoque;

UPDATE estoque SET qtdade = 0 WHERE iditem = id;

UPDATE venda SET qtdade = tot\_estoque WHERE new.idvenda =  
idvenda;

END IF;

RETURN NULL;

END;

\$BODY\$

LANGUAGE 'plpgsql';

## SOLUÇÃO – CRIANDO A TRIGGER

```
CREATE TRIGGER trigger_atualiza_estoque  
  AFTER INSERT ON venda FOR EACH ROW  
  EXECUTE  
  PROCEDURE atualiza_estoque();
```

## SOLUÇÃO – INSERINDO VALORES

```
INSERT INTO venda(idvenda, fk_idvendedor,  
fk_idcliente, valor, datavenda, qtdade, tot,  
fk_iditem)  
VALUES (25, 1, 1, 45, '06/10/2010', 30, 0, 2);
```