**Triggers**

DELIMITER $$

CREATE TRIGGER `Tgr\_Estoquepedidos\_Insert` BEFORE UPDATE ON `pedido`

FOR EACH ROW BEGIN

IF NEW.quantidade > (

Select quantidade

from produto P

where NEW.produtoID = P.id

)THEN

signal sqlstate '45000' set message\_text ='A quantidade solicitada é maior do que a que há em estoque';

END IF;

END $$

DELIMITER ;

**--------**

DELIMITER $$

CREATE TRIGGER `Tgr\_ItensVenda\_Insert` AFTER INSERT ON `venda`

FOR EACH ROW BEGIN

DECLARE idprod INT;

DECLARE quantprod INT;

SET idprod = 0;

SET quantprod = 0;

SELECT produto.ID into idprod

FROM pedido

INNER JOIN produto

ON pedido.produtoID = produto.ID

WHERE pedido.id = NEW.pedidoID;

SELECT pedido.quantidade into quantprod

FROM pedido

INNER JOIN venda

ON venda.pedidoID = pedido.ID

WHERE pedido.ID = NEW.pedidoID;

UPDATE produto

SET quantidade = quantidade - quantprod

WHERE ID = idprod;

END $$

DELIMITER ;

**Function**

CREATE FUNCTION fn\_somarProdutoQntdPreco (preco float, quantidade int)

RETURNS float

RETURN preco \* quantidade;

**Cursor**

DELIMITER $$

CREATE PROCEDURE totalbruto (OUT resultado FLOAT)

BEGIN

DECLARE existe\_mais\_linhas INT DEFAULT 0;

DECLARE total\_produto FLOAT DEFAULT 0;

DECLARE total\_bruto FLOAT DEFAULT 0;

DECLARE meuCursor CURSOR FOR

SELECT fn\_somarProdutoQntdPreco(produto.preco, pedido.quantidade)

FROM pedido

INNER JOIN produto

ON pedido.produtoID = produto.ID;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET existe\_mais\_linhas=1;

OPEN meuCursor;

meuLoop: LOOP

FETCH meuCursor INTO total\_produto;

IF existe\_mais\_linhas = 1 THEN

LEAVE meuLoop;

END IF;

SET total\_bruto = total\_bruto + total\_produto;

END LOOP meuLoop;

SET resultado = total\_bruto;

CLOSE meuCursor;

END $$

DELIMITER ;

CALL totalbruto(@variavel\_temporaria);

SELECT @variavel\_temporaria;

**View**

CREATE VIEW vw\_vendas

AS SELECT pedido.ID as id, produto.nome as nomeproduto, pedido.quantidade as quantidadecarrinho, produto.quantidade as quantidadeproduto, categoria.nome as nomecategoria, produto.preco as preco, fn\_somarProdutoQntdPreco(produto.preco, pedido.quantidade) as precoquantidade

FROM pedido

INNER JOIN produto

ON pedido.produtoID = produto.ID

INNER JOIN categoria

ON produto.categoriaID = categoria.ID;