```
CREATE TABLE trigger_pessoa (codigo int not null auto_increment,
nome varchar(45) not null,
email varchar(45) not null,
telefone varchar(45) not null,
PRIMARY KEY (codigo));
CREATE TABLE trigger produto (codigo int not null auto increment,
nome varchar(45) not null,
estoque int not null,
preco float not null,
PRIMARY KEY (codigo));
CREATE TABLE trigger_compra (codigo int not null auto_increment,
codigo_produto int not null,
codigo pessoa int not null,
quantidade int not null,
valor_unitario float not null,
valor total float not null,
PRIMARY KEY(codigo),
FOREIGN KEY (codigo produto) REFERENCES trigger produto(codigo),
FOREIGN KEY (codigo_pessoa) REFERENCES trigger_pessoa(codigo));
CREATE TABLE trigger venda (codigo int not null auto increment,
codigo produto int not null,
codigo pessoa int not null,
quantidade int not null,
valor unitario float not null,
valor total float not null,
PRIMARY KEY(codigo),
FOREIGN KEY (codigo_produto) REFERENCES trigger_produto(codigo),
FOREIGN KEY (codigo pessoa) REFERENCES trigger pessoa(codigo));
1) Crie uma trigger, que dispare antes de um evento de inserção na tabela pessoa, para
verificar se o atributo email possui a letra @. Se não possuir cancele a inserção
apresentando uma mensagem de erro.
delimiter $$
drop trigger if exists verifica arroba$$
create trigger verifica arroba before insert on trigger pessoa for each
row
begin
declare msg varchar(255);
if (new.email not like '%@%') then
set msg = "Email Invalido";
```

```
signal sqlstate '45000' set message_text = msg;
end if;
end$$
delimiter;
```

```
INSERT INTO trigger_pessoa(nome, email, telefone) values ('helena', 'kdsshdkh', '8994739')

Mensagens do MySQL: 
#1644 - Email Invalido

Executa comando(s) SQL na base de dados info18_helena: 

INSERT INTO trigger_pessoa(nome, email, telefone) values ('helena', 'kdsshdkh', '8994739');
```

2) Faça uma trigger que atualize automaticamente o estoque da tabela produto quando determinado produto for comprado. Também gere automaticamente o valor da coluna "valor_total" na tabela compra.

```
delimiter $$
drop trigger if exists atualiza_valor$$
create trigger atualiza_valor before insert on trigger_compra for each
row
begin
    declare estoquee int;
    declare codigoo int;
    declare precoo float;
    declare aux int;
        select codigo, estoque, preco into codigoo, estoquee, precoo from trigger_produto
where new.codigo_produto=codigo;
        set aux = estoquee + new.quantidade;
```

update trigger_produto set estoque=aux where codigo = codigoo; set new.valor_total = new.quantidade * new.valor_unitario;

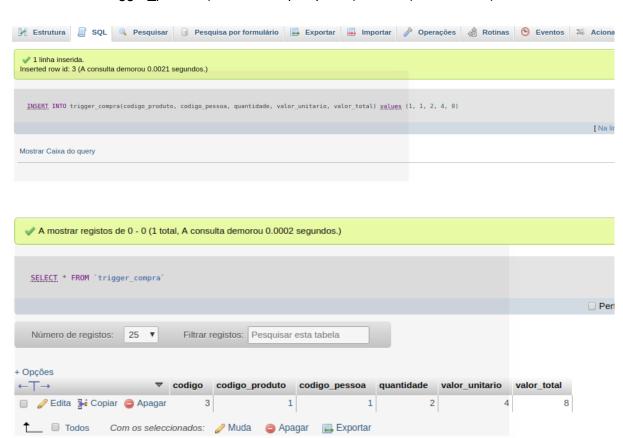
end\$\$ delimiter;

INSERT INTO trigger_compra(codigo_produto, codigo_pessoa, quantidade, valor_unitario, valor_total) values (1, 1, 2, 4, 0);

INSERT INTO trigger_produto(nome, estoque, preco) values ('guarana', 20, 4);

INSERT INTO trigger_produto(nome, estoque, preco) values ('fanta', 40, 5);

INSERT INTO trigger_produto(nome, estoque, preco) values ('coca', 12, 7);



3) Crie uma trigger que no momento de registro de uma venda, de baixa no estoque do produto vendido, considere como valor unitário do produto o preço armazenado na tabela produto acrescido de 30%, e também faça o calculo automático do valor total da venda. Garanta que a quantidade vendida esteja disponível em estoque, se não tiver retorne uma mensagem e cancela a venda.

delimiter \$\$
drop trigger if exists registro_venda\$\$
create trigger registro_venda before insert on trigger_venda for each
row
begin

```
declare msg varchar(255);
       declare quantidadee int;
       declare codigoo int;
       declare precoo float;
       declare porcentagem float;
       declare total float;
       declare aux int;
       select estoque, codigo, preco into quantidadee, codigoo, precoo from trigger_produto
where new.codigo_produto=codigo;
       set aux = quantidadee - new.quantidade;
       update trigger_produto set estoque=aux where codigo = codigoo;
       set porcentagem = (precoo*30)/100;
       set new.valor_unitario = porcentagem + precoo;
       set new.valor_total = new.valor_unitario * new.quantidade;
       if (aux<0) then
              set msg = "Quantidade Invalida";
              signal sqlstate '45000' set message_text = msg;
       end if;
```

end\$\$ delimiter;

```
Erro

Comando SQL:

INSERT INTO trigger_venda(codigo_produto, codigo_pessoa, quantidade, valor_unitario, valor_total) values (3, 1, 50, 3, 0)

Mensagens do MySQL: 
#1644 - Quantidade Invalida

Executa comando(s) SQL na base de dados info18_helena: 
#185ERT INTO trigger_venda(codigo_produto, codigo_pessoa, quantidade, valor_unitario, valor_total) values (3, 1, 50, 3, 0);

Clear
```





4) Crei uma rotina que incremente o preço de todos os produtos em 1% a cada minuto durante 5 minutos a partir de um momento específico.

```
delimiter $$
drop event if exists nome_evento $$
create event nome_evento
on schedule every 1 minute starts now() ends now() + interval 5 minute
on completion preserve
enable
do
begin
update trigger_produto set preco=(preco*0.01)+preco;
```

end \$\$ delimiter;

ESTOQUES

