

```
use loja_de_roupas
```

```
go
```

```
create trigger atualizartotalvendainsert
```

```
on item_venda for insert
```

```
as
```

```
    update venda
```

```
        set total = total + (select i.qtd * r.preco_venda from roupa r inner join  
inserted i on r.codigo = i.codigo_roupa)
```

```
        where codigo_venda = (select codigo_venda from inserted)
```

```
        if @@rowcount != 0
```

```
            update roupa
```

```
                set qtd_estoque = qtd_estoque - (select qtd from  
inserted)
```

```
                where codigo = (select codigo_roupa from  
inserted)
```

```
        if @@rowcount = 0
```

```
            rollback transaction
```

```
go
```

```
create trigger atualizartotalvendadelete
```

```
on item_venda for delete
```

```
as
```

```
    update venda
```

```
        set total = total - (select d.qtd * r.preco_venda from roupa r inner join  
deleted d on r.codigo = d.codigo_roupa)
```

```
        where codigo_venda = (select codigo_venda from deleted)
```

```
        if @@rowcount != 0
```

```
            update roupa
```

```
set qtd_estoque = qtd_estoque + (select qtd from
deleted)
```

```
where codigo = (select codigo_roupa from deleted)
```

```
if @@rowcount = 0
```

```
rollback transaction
```

```
go
```

```
create trigger atualizartotalvendaupdate
```

```
on item_venda for update
```

```
as
```

```
update venda
```

```
set total = total + (select (i.qtd - (select qtd from deleted)) *
r.preco_venda from roupa r inner join inserted i on r.codigo = i.codigo_roupa)
```

```
where codigo_venda = (select codigo_venda from
inserted)
```

```
if @@rowcount != 0
```

```
update roupa
```

```
set qtd_estoque = qtd_estoque + (select qtd from
deleted) - (select qtd from inserted)
```

```
where codigo = (select codigo_roupa from
inserted)
```

```
if @@rowcount = 0
```

```
rollback transaction
```

```
go
```

```
create trigger atualizarpagamentovendainsert
```

```
on pagamento_venda for insert
```

```
as
```

```

update venda

    set total = total - (select i.valor * fp.pct_desconto/100 from
forma_pagamento fp inner join inserted i on fp.codigo =
i.codigo_forma_pagamento)

    where codigo_venda = (select codigo_venda from inserted)

if @@rowcount = 0

    rollback transaction

```

go

create trigger atualizarpagamentovendadelete

on pagamento_venda for delete

as

```

update venda

    set total = total + (select d.valor * fp.pct_desconto/100 from
forma_pagamento fp inner join deleted d on fp.codigo =
d.codigo_forma_pagamento)

    where codigo_venda = (select codigo_venda from deleted)

if @@rowcount = 0

    rollback transaction

```

go

create trigger atualizarpagamentovendaupdate

on pagamento_venda for update

as

```

    (select (i.valor - (select valor from deleted)) * fp.pct_desconto/100 from
forma_pagamento fp inner join inserted i on fp.codigo =
i.codigo_forma_pagamento)

update venda

    set total = total - (select (i.valor - (select valor from deleted)) *
fp.pct_desconto/100 from forma_pagamento fp inner join inserted i on
fp.codigo = i.codigo_forma_pagamento)

```

```

        where codigo_venda = (select codigo_venda from inserted)

if @@rowcount = 0
    rollback transaction

go

create trigger atualizarestoquepedidoinsert
on item_pedido for insert
as

    declare @new_total decimal(10,2) = (select total from pedido where
codigo_pedido = (select codigo_pedido from inserted)) + (select qtd *
preco_custo from inserted);

    update pedido

        set total = @new_total where codigo_pedido = (select
codigo_pedido from inserted);

    if @@ROWCOUNT != 0
        begin

            if (select data_entrega from pedido where codigo_pedido =
(select codigo_pedido from inserted)) is NULL

                print 'A data de entrega é null, o pedido não foi entregue'

            else

                update roupa

                    set qtd_estoque = qtd_estoque + (select qtd from
inserted)

                        where codigo = (select codigo_roupa from
inserted);

                if @@ROWCOUNT != 0

                    if (select codigo_despesa from pedido where
codigo_pedido = (select codigo_pedido from inserted)) is null

                        print @new_total

                    else

```

```

        update despesa set valor = @new_total
                                where codigo = (select codigo_despesa
from pedido where codigo_pedido = (select codigo_pedido from inserted));
    end
    else
        rollback transaction

go

create trigger atualizareestoquepedidodelete
on item_pedido for delete
as
    declare @new_total decimal(10,2) = (select total from pedido where
codigo_pedido = (select codigo_pedido from deleted)) - (select qtd *
preco_custo from deleted);

    update pedido

        set total = @new_total where codigo_pedido = (select
codigo_pedido from deleted);

    if @@ROWCOUNT != 0
    begin

        if (select data_entrega from pedido where codigo_pedido =
(select codigo_pedido from deleted)) is NULL

            print 'A data de entrega não foi alterada'

        else

            update roupa

                set qtd_estoque = qtd_estoque - (select qtd from
deleted)

                    where codigo = (select codigo_roupa from
deleted);

        if @@ROWCOUNT != 0

            if (select codigo_despesa from pedido where
codigo_pedido = (select codigo_pedido from deleted)) is null

```

```

        print @new_total
    else
        update despesa set valor = @new_total
            where codigo = (select codigo_despesa
from pedido where codigo_pedido = (select codigo_pedido from deleted));
    end
else
    rollback transaction

go

create trigger atualizarestoquepedidoupdate
on item_pedido for update
as
    declare @new_total decimal(10,2) = (select total from pedido where
codigo_pedido = (select codigo_pedido from deleted)) - (select qtd *
preco_custo from deleted) + (select qtd * preco_custo from inserted);

    update pedido

        set total = @new_total where codigo_pedido = (select
codigo_pedido from inserted);

    if @@ROWCOUNT != 0
    begin
        if (select data_entrega from pedido where codigo_pedido =
(select codigo_pedido from inserted)) is NULL

            print 'A data de entrega é null, o pedido não foi entregue'

        else

            update roupa

                set qtd_estoque = qtd_estoque - (select qtd from
deleted) + (select qtd from inserted)

                    where codigo = (select codigo_roupa from
inserted);
    end

```

```

        if @@ROWCOUNT != 0
            if (select codigo_despesa from pedido where
codigo_pedido = (select codigo_pedido from inserted)) is null
                print @new_total
            else
                update despesa set valor = @new_total
                    where codigo = (select codigo_despesa
from pedido where codigo_pedido = (select codigo_pedido from inserted));
        end
    else
        rollback transaction

```

go

alter trigger atualizarestoqueupdatepedido

on pedido for update

as

```

    declare @qtd int, @codigo_roupa int
    if (select data_entrega from inserted) is NULL
        print 'A data de entrega é null. o produto não foi entregue'
    else
        begin
            if (select codigo_despesa from inserted) is NULL
                print 'Não há despesa cadastrada'
            else
                update despesa set valor = (select total from inserted)
where codigo = (select codigo_despesa from inserted)
            if (select data_entrega from deleted) is NULL
                begin

```

```

DECLARE cursor_item_pedido CURSOR for
        select qtd, codigo_roupa from item_pedido
where codigo_pedido = (select codigo_pedido from inserted)

OPEN cursor_item_pedido

FETCH NEXT FROM cursor_item_pedido INTO
@qtd, @codigo_roupa

WHILE @@FETCH_STATUS = 0
begin
        print @@FETCH_STATUS
        update roupa set qtd_estoque =
qtd_estoque + @qtd where codigo = @codigo_roupa

        FETCH NEXT FROM cursor_item_pedido
INTO @qtd, @codigo_roupa
end

CLOSE cursor_item_pedido

DEALLOCATE cursor_item_pedido

end

end

```

- **VIEWS:**

```
use grupo4
```

```
go
```


create view select_cliente as

```
select * from pessoa p  
join pessoa_fisica f on p.codigo = f.codigo_fisica  
join cliente c on p.codigo = c.codigo_cliente;
```

go

create view select_funcionario as

```
select * from pessoa p  
join pessoa_fisica f on f.codigo_fisica = p.codigo  
join funcionario fu on fu.codigo_funcionario = f.codigo_fisica;
```

go

create view select_fornecedor as

```
select * from pessoa p  
join fornecedor f on p.codigo = f.codigo_fornecedor;
```

go

create view select_roupa_categoria as

```
select * from roupa r  
join categoria c on r.codigo_categoria = c.codigo;
```

go

create view select_roupa_cliente_funcionario as

```
select * from venda v  
join pessoa pf on pf.codigo = v.codigo_funcionario  
join pessoa_fisica pff on pff.codigo_fisica = v.codigo_funcionario  
join funcionario f on v.codigo_funcionario = f.codigo_funcionario
```

```
join pessoa pc on pc.codigo = v.codigo_cliente  
join pessoa_fisica pfc on pfc.codigo_fisica = v.codigo_cliente  
join cliente c on v.codigo_cliente = c.codigo_cliente
```

```
go
```

```
create view select_itemvenda_venda_roupa as
```

```
select * from item_venda iv  
join venda v on iv.codigo_venda = v.codigo_venda  
join roupa r on iv.codigo_roupa = r.codigo
```

```
go
```

```
create view select_pagamento_venda__venda_roupa as
```

```
select * from pagamento_venda pv  
join venda v on v.codigo_venda = pv.codigo_venda  
join item_venda iv on iv.codigo_venda = v.codigo_venda  
join roupa r on iv.codigo_roupa = r.codigo  
join categoria ca on r.codigo_categoria = ca.codigo;
```

```
go
```

```
create view select_despesa_funcionario as
```

```
select * from despesa d
```

```
inner join funcionario f on d.codigo_funcionario = f.codigo_funcionario;
```

```
go
```

```
create view select_despesa_null as
```

```
select * from despesa where dt_pagamento is null;
```

go

create view select_pedido_funcionario_fornecedor as

select * from pedido p

join pessoa pfu on pfu.codigo = p.codigo_funcionario

join pessoa_fisica pf on pf.codigo_fisica = p.codigo_funcionario

join funcionario fu on p.codigo_funcionario = fu.codigo_funcionario

join pessoa pfo on pfo.codigo = p.codigo_fornecedor

join fornecedor fo on p.codigo_fornecedor = fo.codigo_fornecedor;

go

create view select_item_pedido_roupa as

select * from item_pedido ip

join roupa r on r.codigo = ip.codigo_roupa;

go

create view select_pedido_not_entregue as

select * from pedido where data_entrega is not null;

go

create view select_pedido_entregue as

select * from pedido where data_entrega is null;

go

create view select_fornecimento_fornecedor_funcionario as

select * from fornecimento f

join roupa r on f.codigo_roupa = r.codigo

join pessoa p on p.codigo = f.codigo_fornecedor

```
join fornecedor fo on f.codigo_fornecedor = fo.codigo_fornecedor;
```

```
go
```

```
create view select_items_not_vendidos as
```

```
select * from roupa r
```

```
left outer join item_venda iv on r.codigo = iv.codigo_roupa where  
iv.codigo_roupa is null;
```

```
go
```

```
create view select_vendas_por_funcionario as
```

```
select count(codigo_roupa) as qtd, sum(v.total) as total from item_venda iv
```

```
inner join venda v on v.codigo_venda = iv.codigo_venda
```

```
inner join funcionario f on f.codigo_funcionario = v.codigo_funcionario
```

```
group by v.codigo_funcionario;
```

```
go
```

```
create view select_forma_pagamento as
```

```
select codigo_forma_pagamento, count(codigo_forma_pagamento) as qtd,  
sum(pv.valor) as total from pagamento_venda pv
```

```
inner join roupa r on r.codigo = pv.codigo_venda
```

```
group by codigo_forma_pagamento;
```

```
go
```

```
create view select_prazo_entrega as
```

```
select f.codigo_fornecedor, DATEDIFF(d, data_pedido, data_entrega) as  
entrega, f.prazo_entrega from pedido p
```

```
inner join fornecedor f on f.codigo_fornecedor = p.codigo_fornecedor;
```

```
use grupo4
```

go

create trigger atualizartotalvendainsert

on item_venda for insert

as

update venda

set total = total + (select i.qtd * r.preco_venda from roupa r inner join
inserted i on r.codigo = i.codigo_roupa)

where codigo_venda = (select codigo_venda from inserted)

if @@rowcount != 0

update roupa

set qtd_estoque = qtd_estoque - (select qtd from
inserted)

where codigo = (select codigo_roupa from
inserted)

if @@rowcount = 0

rollback transaction

go

create trigger atualizartotalvendadelete

on item_venda for delete

as

update venda

set total = total - (select d.qtd * r.preco_venda from roupa r inner join
deleted d on r.codigo = d.codigo_roupa)

where codigo_venda = (select codigo_venda from deleted)

if @@rowcount != 0

update roupa

set qtd_estoque = qtd_estoque + (select qtd from
deleted)

```

                                where codigo = (select codigo_roupa from deleted)

if @@rowcount = 0
    rollback transaction

go

create trigger atualizartotalvendaupdate
on item_venda for update
as

    update venda

        set total = total + (select (i.qtd - (select qtd from deleted)) *
r.preco_venda from roupa r inner join inserted i on r.codigo = i.codigo_roupa)

        where  codigo_venda = (select  codigo_venda  from
inserted)

    if @@rowcount != 0

        update roupa

            set qtd_estoque = qtd_estoque + (select qtd from
deleted) - (select qtd from inserted)

        where codigo = (select codigo_roupa from
inserted)

    if @@rowcount = 0

        rollback transaction

go

create trigger atualizarpagamentovendainsert
on pagamento_venda for insert
as

    update venda

```

```
        set total = total - (select i.valor * fp.pct_desconto/100 from
forma_pagamento fp inner join inserted i on fp.codigo =
i.codigo_forma_pagamento)
```

```
        where codigo_venda = (select codigo_venda from inserted)
```

```
    if @@rowcount = 0
```

```
        rollback transaction
```

```
go
```

```
create trigger atualizarpagamentovendadelete
```

```
on pagamento_venda for delete
```

```
as
```

```
    update venda
```

```
        set total = total + (select d.valor * fp.pct_desconto/100 from
forma_pagamento fp inner join deleted d on fp.codigo =
d.codigo_forma_pagamento)
```

```
        where codigo_venda = (select codigo_venda from deleted)
```

```
    if @@rowcount = 0
```

```
        rollback transaction
```

```
go
```

```
create trigger atualizarpagamentovendaupdate
```

```
on pagamento_venda for update
```

```
as
```

```
        (select (i.valor - (select valor from deleted)) * fp.pct_desconto/100 from
forma_pagamento fp inner join inserted i on fp.codigo =
i.codigo_forma_pagamento)
```

```
    update venda
```

```
        set total = total - (select (i.valor - (select valor from deleted)) *
fp.pct_desconto/100 from forma_pagamento fp inner join inserted i on
fp.codigo = i.codigo_forma_pagamento)
```

```
        where codigo_venda = (select codigo_venda from inserted)
```

```
if @@rowcount = 0
```

```
rollback transaction
```

```
go
```

```
create trigger atualizareestoquepedidoinsert
```

```
on item_pedido for insert
```

```
as
```

```
declare @new_total decimal(10,2) = (select total from pedido where  
codigo_pedido = (select codigo_pedido from inserted)) + (select qtd *  
preco_custo from inserted);
```

```
update pedido
```

```
set total = @new_total where codigo_pedido = (select  
codigo_pedido from inserted);
```

```
if @@ROWCOUNT != 0
```

```
begin
```

```
if (select data_entrega from pedido where codigo_pedido =  
(select codigo_pedido from inserted)) is NULL
```

```
print 'A data de entrega é null, o pedido não foi entregue'
```

```
else
```

```
update roupa
```

```
set qtd_estoque = qtd_estoque + (select qtd from  
inserted)
```

```
where codigo = (select codigo_roupa from  
inserted);
```

```
if @@ROWCOUNT != 0
```

```
if (select codigo_despesa from pedido where  
codigo_pedido = (select codigo_pedido from inserted)) is null
```

```
print @new_total
```

```
else
```

```
update despesa set valor = @new_total
```



```
                                where codigo = (select codigo_despesa
from pedido where codigo_pedido = (select codigo_pedido from inserted));
```

```
end
```

```
else
```

```
rollback transaction
```

```
go
```

```
create trigger atualizarestoquepedidodelete
```

```
on item_pedido for delete
```

```
as
```

```
declare @new_total decimal(10,2) = (select total from pedido where
codigo_pedido = (select codigo_pedido from deleted)) - (select qtd *
preco_custo from deleted);
```

```
update pedido
```

```
set total = @new_total where codigo_pedido = (select
codigo_pedido from deleted);
```

```
if @@ROWCOUNT != 0
```

```
begin
```

```
if (select data_entrega from pedido where codigo_pedido =
(select codigo_pedido from deleted)) is NULL
```

```
print 'A data de entrega não foi alterada'
```

```
else
```

```
update roupa
```

```
set qtd_estoque = qtd_estoque - (select qtd from
deleted)
```

```
where codigo = (select codigo_roupa from
deleted);
```

```
if @@ROWCOUNT != 0
```

```
if (select codigo_despesa from pedido where
codigo_pedido = (select codigo_pedido from deleted)) is null
```

```
print @new_total
```

```

else

        update despesa set valor = @new_total

                                where codigo = (select codigo_despesa
from pedido where codigo_pedido = (select codigo_pedido from deleted));

end

else

        rollback transaction

go

create trigger atualizareestoquepedidoupdate

on item_pedido for update

as

        declare @new_total decimal(10,2) = (select total from pedido where
codigo_pedido = (select codigo_pedido from deleted)) - (select qtd *
preco_custo from deleted) + (select qtd * preco_custo from inserted);

        update pedido

                set total = @new_total where codigo_pedido = (select
codigo_pedido from inserted);

        if @@ROWCOUNT != 0

                begin

                        if (select data_entrega from pedido where codigo_pedido =
(select codigo_pedido from inserted)) is NULL

                                print 'A data de entrega é null, o pedido não foi entregue'

                        else

                                update roupa

                                        set qtd_estoque = qtd_estoque - (select qtd from
deleted) + (select qtd from inserted)

                                                where codigo = (select codigo_roupa from
inserted);

                                if @@ROWCOUNT != 0

```

```

        if (select codigo_despesa from pedido where
codigo_pedido = (select codigo_pedido from inserted)) is null

            print @new_total

        else

            update despesa set valor = @new_total

                where codigo = (select codigo_despesa
from pedido where codigo_pedido = (select codigo_pedido from inserted));

        end

        else

            rollback transaction

```

go

alter trigger atualizarestoqueupdatepedido

on pedido for update

as

```

        declare @qtd int, @codigo_roupa int

        if (select data_entrega from inserted) is NULL

            print 'A data de entrega é null. o produto não foi entregue'

        else

            begin

                if (select codigo_despesa from inserted) is NULL

                    print 'Não há despesa cadastrada'

                else

                    update despesa set valor = (select total from inserted)
where codigo = (select codigo_despesa from inserted)

                    if (select data_entrega from deleted) is NULL

                        begin

                            DECLARE cursor_item_pedido CURSOR for

```

```
select qtd, codigo_roupa from item_pedido
where codigo_pedido = (select codigo_pedido from inserted)
```

```
OPEN cursor_item_pedido
```

```
FETCH NEXT FROM cursor_item_pedido INTO
@qtd, @codigo_roupa
```

```
WHILE @@FETCH_STATUS = 0
```

```
begin
```

```
print @@FETCH_STATUS
```

```
update roupa set qtd_estoque =
qtd_estoque + @qtd where codigo = @codigo_roupa
```

```
FETCH NEXT FROM cursor_item_pedido
INTO @qtd, @codigo_roupa
```

```
end
```

```
CLOSE cursor_item_pedido
```

```
DEALLOCATE cursor_item_pedido
```

```
end
```

```
end
```

- **VIEWS:**

```
use grupo4
```

```
go
```

```
create view select_cliente as
```

```
select * from pessoa p
join pessoa_fisica f on p.codigo = f.codigo_fisica
join cliente c on p.codigo = c.codigo_cliente;
```

go

create view select_funcionario as

```
select * from pessoa p
join pessoa_fisica f on f.codigo_fisica = p.codigo
join funcionario fu on fu.codigo_funcionario = f.codigo_fisica;
```

go

create view select_fornecedor as

```
select * from pessoa p
join fornecedor f on p.codigo = f.codigo_fornecedor;
```

go

create view select_roupa_categoria as

```
select * from roupa r
join categoria c on r.codigo_categoria = c.codigo;
```

go

create view select_roupa_cliente_funcionario as

```
select * from venda v
join pessoa pf on pf.codigo = v.codigo_funcionario
join pessoa_fisica pff on pff.codigo_fisica = v.codigo_funcionario
join funcionario f on v.codigo_funcionario = f.codigo_funcionario
join pessoa pc on pc.codigo = v.codigo_cliente
```

```
join pessoa_fisica pfc on pfc.codigo_fisica = v.codigo_cliente
```

```
join cliente c on v.codigo_cliente = c.codigo_cliente
```

```
go
```

```
create view select_itemvenda_venda_roupa as
```

```
select * from item_venda iv
```

```
join venda v on iv.codigo_venda = v.codigo_venda
```

```
join roupa r on iv.codigo_roupa = r.codigo
```

```
go
```

```
create view select_pagamento_venda__venda_roupa as
```

```
select * from pagamento_venda pv
```

```
join venda v on v.codigo_venda = pv.codigo_venda
```

```
join item_venda iv on iv.codigo_venda = v.codigo_venda
```

```
join roupa r on iv.codigo_roupa = r.codigo
```

```
join categoria ca on r.codigo_categoria = ca.codigo;
```

```
go
```

```
create view select_despesa_funcionario as
```

```
select * from despesa d
```

```
inner join funcionario f on d.codigo_funcionario = f.codigo_funcionario;
```

```
go
```

```
create view select_despesa_null as
```

```
select * from despesa where dt_pagamento is null;
```

```
go
```

```
create view select_pedido_funcionario_fornecedor as
select * from pedido p
join pessoa pfu on pfu.codigo = p.codigo_funcionario
join pessoa_fisica pf on pf.codigo_fisica = p.codigo_funcionario
join funcionario fu on p.codigo_funcionario = fu.codigo_funcionario
join pessoa pfo on pfo.codigo = p.codigo_fornecedor
join fornecedor fo on p.codigo_fornecedor = fo.codigo_fornecedor;
```

go

create view select_item_pedido_roupa as

```
select * from item_pedido ip
join roupa r on r.codigo = ip.codigo_roupa;
```

go

```
create view select_pedido_not_entregue as
select * from pedido where data_entrega is not null;
```

go

create view select_pedido_entregue as

```
select * from pedido where data_entrega is null;
```

go

create view select_fornecimento_fornecedor_funcionario as

```
select * from fornecimento f
join roupa r on f.codigo_roupa = r.codigo
join pessoa p on p.codigo = f.codigo_fornecedor
join fornecedor fo on f.codigo_fornecedor = fo.codigo_fornecedor;
```

go

create view select_items_not_vendidos as

select * from roupa r

left outer join item_venda iv on r.codigo = iv.codigo_roupa where
iv.codigo_roupa is null;

go

create view select_vendas_por_funcionario as

select count(codigo_roupa) as qtd, sum(v.total) as total from item_venda iv

inner join venda v on v.codigo_venda = iv.codigo_venda

inner join funcionario f on f.codigo_funcionario = v.codigo_funcionario

group by v.codigo_funcionario;

go

create view select_forma_pagamento as

select codigo_forma_pagamento, count(codigo_forma_pagamento) as qtd,
sum(pv.valor) as total from pagamento_venda pv

inner join roupa r on r.codigo = pv.codigo_venda

group by codigo_forma_pagamento;

go

create view select_prazo_entrega as

select f.codigo_fornecedor, DATEDIFF(d, data_pedido, data_entrega) as
entrega, f.prazo_entrega from pedido p

inner join fornecedor f on f.codigo_fornecedor = p.codigo_fornecedor;