

# Banco de Dados

Prof. Anésio

## Aula 09 – Resposta da Atividade de Trigger

1. Escreva um trigger que realize a atualização do campo valor médio do Imóvel a cada nova oferta cadastrada, alterada ou excluída.

```
Alter Table Imovel
Add vl_MediaOferta money

Create Trigger AtualizaMedia on Oferta
For Insert, Update, Delete
As
Begin
Declare @media money, @Imovel int
If Exists (Select * From Deleted)
    set @Imovel = (Select cd_imovel from Deleted
                    group by cd_Imovel)
Else
    set @Imovel = (Select cd_imovel from Inserted
                    group by cd_Imovel)
Set @media = (Select avg(vl_oferta) from Oferta
              where cd_Imovel = @Imovel)
Update Imovel Set vl_MediaOferta = @media
              where cd_Imovel = @Imovel
End
```

2. Escreva um trigger que não permita a alteração de dados na tabela Faixa\_Imovel a sua exclusão.

```
create trigger tr_Protege on Faixa_Imovel
instead of delete
as
begin
    declare @faixa varchar(30)
    select @faixa = nm_faixa from deleted
    print 'Registro ' + @faixa + ' não pode ser Excluido'
end
```

3. Fazer um procedimento que no ato da inclusão de uma nova oferta, o banco atualize automaticamente a quantidade de ofertas no campo qt\_Ofertas da tabela Imóvel.

```
Create Trigger tr_ContaOferta on Oferta
For Insert As
Begin
    Declare @Imovel int, @Cont int
    Set @Imovel = (Select cd_Imovel From Inserted)
    Select @Cont = Count(*)
    From Oferta
    Where cd_Imovel = @Imovel
    Update Imovel
    Set qt_Ofertas = @Cont
    Where cd_Imovel = @Imovel
End
```

4. No ato da gravação do contrato, o banco deverá gerar automaticamente os registros de mensalidades do parcelamento com os Imóveis, mesmo em pagamento a vista, deverá gerar um registro.

```
Create Trigger tr_ParcelasContrato on Contrato
For Insert
As
Begin
    Declare @Valor money, @Contrato int, @Parc int,
            @Venc datetime, @Cont int
    Set @Contrato = (Select cd_Contrato From Inserted)
    Set @Parc = (Select qt_Parcela From Inserted)
    Set @Valor = (Select vl_Contrato From Inserted) / @Parc
    Set @Venc = (Select dt_Contrato From Inserted)
    Set @Cont = 0
    While @Parc > @Cont
    Begin
        Set @Cont = @Cont + 1
        Insert Into Parcelas
            Values (@Contrato, @Cont, @Valor,
                Dateadd(M,@Cont,@Venc),null,0)
    End
End
```

5. Fazer um procedimento que no ato da alteração da data do pagamento de uma parcela, o banco atualize automaticamente o valor da multa, caso existir atraso. Calcular o valor da multa em 3% por dia de atraso. Caso a data de pagamento seja nula, o banco deve zerar o valor da multa.

```
Create Trigger tr_CalculaMulta on Parcelas
For Update As
Begin
    Declare @Valor money, @Venc datetime, @Pag datetime,
            @Dias int, @ValParc money, @Contrato int, @Parcela int
    Set @Venc = (Select dt_Vencimento From Deleted)
    Set @Pag = (Select dt_Pagamento From Inserted)
    Set @Contrato = (Select cd_Contrato From Inserted)
    Set @Parcela = (Select cd_Parcela From Inserted)
    Set @Valor = 0
    If @Pag > @Venc
    Begin
        Set @ValParc = (Select vl_Parcela From Inserted)
        Set @Dias = DATEDIFF(Day,@Venc,@Pag)
        Set @Valor = @Dias * @ValParc * 0.03
    End
    Update Parcelas Set vl_Multa = @Valor
    Where cd_Contrato = @Contrato and
        qt_Parcela = @Parcela
End
```

6. Fazer um procedimento para gerar uma tabela temporária (Inadimplentes) informando o código do contrato, o número da parcela, a data de vencimento e o valor a ser pago dos Imóveis que estão inadimplentes no mês anterior ao mês corrente, para gerar a cobrança dos pagamentos em atraso.

```
Create Procedure sp_Inadimplentes
As
Begin
    Delete From Inadimplentes
    Declare @Data datetime
    Set @Data = Dateadd(month,-1,getdate())
    Insert Into Inadimplentes
        Select C.cd_Imovel, P.cd_Parcela,
               P.dt_Vencimento, P.vl_Parcela
        From Parcelas P, Contrato C
        Where P.cd_Contrato = C.cd_Contrato and
               P.dt_Pagamento is null and
               P.dt_Vencimento < @Data
end
```