

A. Criar uma procedure que atualize os salários dos vendedores com faixa de comissão = 'B' :

A procedure deverá receber como parâmetro de entrada um código que definirá a porcentagem de aumento. Se

- Código = 2 aumentar salário_fixo em 20%
- Código = 3 aumentar salário_fixo em 30%
- Código = 4 aumentar salário_fixo em 40%
- Código tiver um valor diferente destes o salário deverá permanecer o mesmo

Iniciar e terminar a procedure efetuando um comando select na tabela vendedor.

```
create PROCEDURE alterarValor2(@codigo as INT)
```

```
AS
```

```
BEGIN
```

```
IF @codigo = 2
```

```
BEGIN
```

```
SELECT salario_fixo * 1.2, nome_vendedor
```

```
FROM dbo.vendedor
```

```
WHERE faixa_comissao = 'B'
```

```
END;
```

```
IF @codigo = 3
```

```
BEGIN
```

```
SELECT salario_fixo * 1.3, nome_vendedor
```

```
FROM dbo.vendedor
```

```
WHERE faixa_comissao = 'B'
```

```
END;
```

```
IF @codigo = 4
```

```

BEGIN

    SELECT salario_fixo * 1.4, nome_vendedor

    FROM dbo.vendedor

    WHERE faixa_comissao = 'B'

END;

```

```

END;

```

```

=====

```

B. Criar uma procedure que liste uma classificação do produto de acordo com o especificado abaixo:

- Se valor_unitario < 2, classificação = 'Baixo'
- Se valor_unitario entre 2 e 10, classificação = 'Moderado'
- Se valor_unitario > 10, classificação = 'caro'

O código do produto deve ser passado como parâmetro da procedure

Listar o código do produto, a descrição, o valor unitário e a classificação

```

CREATE PROCEDURE verificarValorUnitario

```

```

    @codigo AS INT

```

```

AS

```

```

BEGIN

```

```

    SELECT

```

```

        cod_produto,

```

```

        descricao,

```

```

        Valor_unitario AS valor_unitario,

```

```

    CASE

```

```

        WHEN Valor_unitario < 2 THEN 'Baixo'

        WHEN Valor_unitario >= 2 AND Valor_unitario < 10 THEN 'Moderado'

        WHEN Valor_unitario > 10 THEN 'Caro'

    END AS classificacao

FROM dbo.produto

WHERE cod_produto = @codigo;

END;

```

=====

C. Criar uma procedure para atualizar o valor unitário de um produto específico na tabela de produtos

A procedure deverá receber como parâmetros de entrada o código do produto que deverá ter seu preço atualizado e a taxa de aumento(por exemplo 0.20,0.10 etc)

Listar os dados do produto especificado, no início e no fim da procedure

```

ALTER PROCEDURE atualizarValor2

    @codigoProduto AS INT,
    @aumento AS INT,
    @valorAtualizado AS DECIMAL(10, 2) OUTPUT
AS
BEGIN
    UPDATE dbo.produto
    SET Valor_unitario = Valor_unitario * @aumento
    WHERE cod_produto = @codigoProduto;

    SELECT @valorAtualizado = Valor_unitario
    FROM dbo.produto
    WHERE cod_produto = @codigoProduto;

    -- Mostrar o valor atualizado
    SELECT @valorAtualizado AS ValorAtualizado;
END;

```

=====

D. Escreva uma stored procedure que realize o seguinte:

- Obter o código de um vendedor como parâmetro de entrada da procedure
- Testar se o código do vendedor existe na tabela vendedor, se não existir imprimir uma mensagem e sair da procedure
- Testar se o código do vendedor existe na tabela de pedidos, se não existir imprimir uma mensagem de erro e sair da procedure
- Calcular o valor total de suas vendas e exibi-lo

```
ALTER PROCEDURE obterCodigo2(@codigoVendedor as INT )
AS
BEGIN
IF EXISTS(SELECT * from dbo.vendedor WHERE cod_vendedor =@codigoVendedor)
BEGIN
    PRINT 'Existe o codigo do vendedor na tabela vendedor'
    SELECT v.nome_vendedor,p.cod_cliente,c.Nome_cliente,SUM(pr.Valor_unitario) AS
'TOTAL'
    FROM  dbo.pedido AS p
    INNER JOIN dbo.vendedor AS v
    ON p.cod_vendedor = v.cod_vendedor
    INNER JOIN dbo.cliente AS c
    ON p.cod_cliente = c.cod_cliente
    INNER JOIN dbo.item_pedido as it
    ON it.num_pedido = p.Num_pedido
    INNER JOIN produto pr
    ON pr.cod_produto = it.cod_produto
    WHERE v.cod_vendedor =@codigoVendedor
    GROUP BY v.nome_vendedor, p.cod_cliente, c.Nome_cliente
END;
ELSE
BEGIN
    PRINT 'codigo nao existe na tabela vendedor'
END;
IF EXISTS(SELECT * FROM dbo.pedido WHERE cod_vendedor = @codigoVendedor)
BEGIN
    PRINT 'Existe o codigo do vendedor na tabela pedido'
END;
ELSE
BEGIN
    PRINT 'codigo nao existe na tabela pedido'
```

END;

END;

=====

Versão com CASE

A. Criar uma procedure que atualize os salários dos vendedores com faixa de comissão = 'B' :

A procedure deverá receber como parâmetro de entrada um código que definirá a porcentagem de aumento. Se

- Código = 2 aumentar salário_fixo em 20%
- Código = 3 aumentar salário_fixo em 30%
- Código = 4 aumentar salário_fixo em 40%
- Código tiver um valor diferente destes o salário deverá permanecer o mesmo

Iniciar e terminar a procedure efetuando um comando select na tabela vendedor.

CREATE PROCEDURE verificarValorUnitario

@codigo **AS INT**

AS

BEGIN

SELECT

cod_produto,

descricao,

Valor_unitario **AS** valor_unitario,

CASE

```
WHEN Valor_unitario < 2 THEN 'Baixo'

WHEN Valor_unitario >= 2 AND Valor_unitario < 10 THEN 'Moderado'

WHEN Valor_unitario > 10 THEN 'Caro'

END AS classificacao

FROM dbo.produto

WHERE cod_produto = @codigo;

END;
```