

## LISTA 2 MATHEUS

*criação de tabelas:*

```
CREATE TABLE regiao(  
    cod_regiao INT IDENTITY(1,1) PRIMARY KEY,  
    Nome_regiao VARCHAR (15)  
  
);  
  
CREATE TABLE departamento (  
    cod_departamento INT IDENTITY(1,1) PRIMARY KEY,  
    Nome_departamento VARCHAR(15),  
    cod_regiao INT,  
    CONSTRAINT FK_departamento_regiao FOREIGN KEY (cod_regiao)  
REFERENCES regiao(cod_regiao)  
);
```

*Verificando possíveis erros na inserção de dados:*

```
ALTER PROCEDURE copulandoTabela(@nomeRegiao VARCHAR(45))  
AS  
BEGIN  
    BEGIN TRANSACTION;  
  
    BEGIN TRY  
        INSERT INTO regiao(Nome_regiao) VALUES(@nomeRegiao);  
        IF @@ERROR > 0  
        BEGIN  
            ROLLBACK; -- Desfaz a transação em caso de erro  
        END  
        ELSE  
        BEGIN
```

```

        COMMIT; -- Confirma a transação se não houver
erros
    END
END TRY
BEGIN CATCH
    ROLLBACK; -- Em caso de erro, desfaz a transação
END CATCH
END;

```

=====

```

CREATE PROCEDURE copulandoDepartamento(@nomeDepartamento
VARCHAR(45))
AS
BEGIN
    BEGIN TRANSACTION;

    BEGIN TRY
        INSERT INTO departamento( Nome_departamento)
VALUES (@nomeDepartamento);
        IF @@ERROR > 0
        BEGIN
            ROLLBACK; -- Desfaz a transação em caso de erro
        END
        ELSE
        BEGIN
            COMMIT; -- Confirma a transação se não houver
erros
        END
    END TRY
    BEGIN CATCH
        ROLLBACK; -- Em caso de erro, desfaz a transação
    END CATCH
END;

```

```

CREATE PROCEDURE AlterarCodigoRegiaoDepartamento
    @codigoDepartamento INT,
    @nomeRegiao VARCHAR(45)
AS
BEGIN
    BEGIN TRANSACTION;

    DECLARE @codRegiao INT;

    -- Obter o código da região a partir do nome informado
    SELECT @codRegiao = cod_regiao
    FROM regiao
    WHERE Nome_regiao = @nomeRegiao;

    -- Verificar se o nome da região existe na tabela regiao
    IF @codRegiao IS NULL
    BEGIN
        PRINT 'Nome da regiao não existente na tabela.';
        ROLLBACK;
        RETURN;
    END;

    -- Verificar se há duplicidade de nome de região
    IF @@ROWCOUNT > 1
    BEGIN
        PRINT 'O nome da regiao está duplicado.';
        ROLLBACK;
        RETURN;
    END;

    -- Alterar o código da região na tabela departamento
    UPDATE departamento
    SET cod_regiao = @codRegiao
    WHERE cod_departamento = @codigoDepartamento;

```

```
-- Verificar se ocorreu algum erro na atualização
IF @@ERROR <> 0
BEGIN
    PRINT 'Algum erro ocorreu.';
    ROLLBACK;
    RETURN;
END;

-- Efetivar a transação
COMMIT;
PRINT 'Alteração do código da regioa concluída com
sucesso.';
END;
```