

```
--  
=====  
===  
-- ① Procedure: Cadastrar Novo Usuário com Validação  
--  
=====  
===  
CREATE PROCEDURE PR_INSERIR_USUARIO  
    @NOME VARCHAR(255),  
    @EMAIL VARCHAR(150),  
    @SENHA VARCHAR(255),  
    @DATA_CADASTRO DATETIME,  
    @PLANO_ID INT  
AS  
BEGIN  
    BEGIN TRY  
        BEGIN TRANSACTION;  
  
        IF NOT EXISTS (SELECT 1 FROM PLANO_ASSINATURA WHERE PLANO_ID = @PLANO_ID)  
            RAISERROR('Erro: Plano de assinatura informado não existe.', 16, 1);  
  
        IF LEN(@SENHA) < 8  
            RAISERROR('Erro: A senha deve conter no mínimo 8 caracteres.', 16,  
1);  
  
        INSERT INTO USUARIO (NOME, EMAIL, SENHA, DATA_CADASTRO, PLANO_ID)  
        VALUES (@NOME, @EMAIL, @SENHA, @DATA_CADASTRO, @PLANO_ID);  
  
        COMMIT;  
    END TRY  
    BEGIN CATCH  
        ROLLBACK;  
        PRINT ERROR_MESSAGE();  
    END CATCH  
END;  
GO  
  
EXEC PR_INSERIR_USUARIO  
    @NOME = 'João Silva',  
    @EMAIL = 'joao@email.com',  
    @SENHA = 'senha1234',  
    @DATA_CADASTRO = GETDATE(),  
    @PLANO_ID = 1;  
GO
```

```
--  
=====  
===  
-- ② Procedure: Listar Filmes por Plano  
--  
=====  
===  
CREATE PROCEDURE PR_LISTAR_FILMES_PLANO  
    @PLANO_ID INT  
AS  
BEGIN  
    BEGIN TRY  
        BEGIN TRANSACTION;  
  
        IF NOT EXISTS (SELECT 1 FROM PLANO_ASSINATURA WHERE PLANO_ID = @PLANO_ID)  
            RAISERROR('Erro: Plano de assinatura não encontrado.', 16, 1);  
  
        SELECT F.TITULO, F.DURACAO  
        FROM FILME F  
        INNER JOIN FILME_PLANO_ASSINATURA FP ON FP.FILME_ID = F.FILME_ID  
        WHERE FP.PLANO_ID = @PLANO_ID;  
  
        COMMIT;  
    END TRY  
    BEGIN CATCH  
        ROLLBACK;  
        PRINT ERROR_MESSAGE();  
    END CATCH  
END;  
GO  
  
EXEC PR_LISTAR_FILMES_PLANO @PLANO_ID = 1;  
GO  
  
--  
=====  
===  
-- ③ Procedure: Atualizar Plano com Transação  
--  
=====  
===  
CREATE PROCEDURE PR_ATUALIZAR_PLANO  
    @PLANO_ID INT,  
    @NOVO_PRECO DECIMAL(10,2),  
    @NOVA_QTD_DISPOSITIVOS INT  
AS  
BEGIN
```

```

BEGIN TRY
    BEGIN TRANSACTION;

    IF NOT EXISTS (SELECT 1 FROM PLANO_ASSINATURA WHERE PLANO_ID = @PLANO_ID)
        RAISERROR('Erro: Plano informado não existe.', 16, 1);

    IF @NOVO_PRECO <= 0
        RAISERROR('Erro: O preço deve ser maior que zero.', 16, 1);

    IF @NOVA_QTD_DISPOSITIVOS <= 0
        RAISERROR('Erro: A quantidade de dispositivos deve ser maior que
zero.', 16, 1);

    UPDATE PLANO_ASSINATURA
    SET PRECO = @NOVO_PRECO,
        QTD_DISPOSITIVOS = @NOVA_QTD_DISPOSITIVOS
    WHERE PLANO_ID = @PLANO_ID;

    COMMIT;
END TRY
BEGIN CATCH
    ROLLBACK;
    PRINT ERROR_MESSAGE();
END CATCH
END;
GO

```

```

EXEC PR_ATUALIZAR_PLANO
    @PLANO_ID = 1,
    @NOVO_PRECO = 39.90,
    @NOVA_QTD_DISPOSITIVOS = 4;
GO

```

```

-- =====
=====  

-- ④ Procedure: Remover Usuário com Restrição
-- =====
=====  

CREATE PROCEDURE PR_REMOVER_USUARIO
    @USUARIO_ID INT
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION;

```

```

IF NOT EXISTS (SELECT 1 FROM USUARIO WHERE USUARIO_ID = @USUARIO_ID)
    RAISERROR('Erro: Usuário informado não existe.', 16, 1);

IF EXISTS (SELECT 1 FROM VISUALIZACAO WHERE USUARIO_ID = @USUARIO_ID)
    RAISERROR('Erro: Não é possível excluir. O usuário possui
visualizações.', 16, 1);

IF EXISTS (SELECT 1 FROM AVALIACAO WHERE USUARIO_ID = @USUARIO_ID)
    RAISERROR('Erro: Não é possível excluir. O usuário possui
avaliações.', 16, 1);

DELETE FROM USUARIO WHERE USUARIO_ID = @USUARIO_ID;

COMMIT;
END TRY
BEGIN CATCH
    ROLLBACK;
    PRINT ERROR_MESSAGE();
END CATCH
END;
GO

EXEC PR_REMOVER_USUARIO @USUARIO_ID = 1;
GO

-- =====
-- ⑤ Procedure: Inserir Filme com Validação (Desafio)
-- =====

CREATE PROCEDURE PR_INSERIR_FILME
    @TITULO VARCHAR(255),
    @DURACAO INT,
    @ANO_LANCAMENTO INT,
    @CLASSIFICACAO VARCHAR(10)
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION;

        IF @DURACAO <= 0
            RAISERROR('Erro: A duração do filme deve ser maior que zero.', 16,
1);

        IF @CLASSIFICACAO NOT IN ('L', '10+', '12+', '14+', '16+', '18+')

```

```

RAISERROR('Erro: Classificação indicativa inválida.', 16, 1);

INSERT INTO FILME (TITULO, DURACAO, ANO_LANCAMENTO, CLASSIFICACAO)
VALUES (@TITULO, @DURACAO, @ANO_LANCAMENTO, @CLASSIFICACAO);

COMMIT;
END TRY
BEGIN CATCH
    ROLLBACK;
    PRINT ERROR_MESSAGE();
END CATCH
END;
GO

EXEC PR_INSERIR_FILME
    @TITULO = 'O Poderoso Chefão',
    @DURACAO = 10500,
    @ANO_LANCAMENTO = 1972,
    @CLASSIFICACAO = '18+';
GO

-- =====
-- Procedure: Inserir Avaliação com Verificação
-- =====

CREATE PROCEDURE PR_INSERIR_AVALIACAO
    @USUARIO_ID INT,
    @FILME_ID INT,
    @VISUALIZACAO_ID INT,
    @NOTA DECIMAL(2,1),
    @COMENTARIO VARCHAR(500)
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION;

        IF @NOTA < 0.0 OR @NOTA > 5.0
            RAISERROR('Erro: A nota deve estar entre 0.0 e 5.0.', 16, 1);

        IF NOT EXISTS (
            SELECT 1 FROM VISUALIZACAO
            WHERE VISUALIZACAO_ID = @VISUALIZACAO_ID
                AND USUARIO_ID = @USUARIO_ID
                AND FILME_ID = @FILME_ID
        )
    END TRY
    BEGIN CATCH
        ROLLBACK;
        PRINT ERROR_MESSAGE();
    END CATCH
END;
GO

```

```
)  
    RAISERROR('Erro: Visualização não encontrada ou não corresponde ao  
usuário e filme.', 16, 1);  
  
    INSERT INTO AVALIACAO (USUARIO_ID, FILME_ID, VISUALIZACAO_ID, NOTA,  
COMENTARIO)  
        VALUES (@USUARIO_ID, @FILME_ID, @VISUALIZACAO_ID, @NOTA, @COMENTARIO);  
  
    COMMIT;  
END TRY  
BEGIN CATCH  
    ROLLBACK;  
    PRINT ERROR_MESSAGE();  
END CATCH  
END;  
GO
```

```
EXEC PR_INSERIR_AVALIACAO  
    @USUARIO_ID = 1,  
    @FILME_ID = 1,  
    @VISUALIZACAO_ID = 1,  
    @NOTA = 4.5,  
    @COMENTARIO = 'Excelente filme!';
```

```
GO
```

```
--  
=====  
=====  
-- ⑦ Procedure: Registrar Visualização com Validação  
--  
=====  
=====  
CREATE PROCEDURE PR_INSERIR_VISUALIZACAO  
    @USUARIO_ID INT,  
    @FILME_ID INT,  
    @DATA_HORA DATETIME,  
    @TEMPO_ASSISTIDO INT  
AS  
BEGIN  
    BEGIN TRY  
        BEGIN TRANSACTION;  
  
        IF NOT EXISTS (SELECT 1 FROM USUARIO WHERE USUARIO_ID = @USUARIO_ID)  
            RAISERROR('Erro: Usuário informado não existe.', 16, 1);  
  
        IF NOT EXISTS (SELECT 1 FROM FILME WHERE FILME_ID = @FILME_ID)  
            RAISERROR('Erro: Filme informado não existe.', 16, 1);
```

```

IF @TEMPO_ASSISTIDO < 0
    RAISERROR('Erro: Tempo assistido inválido.', 16, 1);

INSERT INTO VISUALIZACAO (USUARIO_ID, FILME_ID, DATA_HORA,
TEMPO_ASSISTIDO)
VALUES (@USUARIO_ID, @FILME_ID, @DATA_HORA, @TEMPO_ASSISTIDO);

COMMIT;

END TRY
BEGIN CATCH
    ROLLBACK;
    PRINT ERROR_MESSAGE();
END CATCH
END;
GO

EXEC PR_INSERIR_VISUALIZACAO
    @USUARIO_ID = 1,
    @FILME_ID = 1,
    @DATA_HORA = GETDATE(),
    @TEMPO_ASSISTIDO = 3600;
GO

-- =====
-- [8] Procedure: Obter Média de Notas por Filme
-- =====
CREATE PROCEDURE PR_MEDIA_NOTA_FILME
    @FILME_ID INT
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION;

        IF NOT EXISTS (SELECT 1 FROM FILME WHERE FILME_ID = @FILME_ID)
            RAISERROR('Erro: Filme informado não existe.', 16, 1);

        IF NOT EXISTS (SELECT 1 FROM AVALIACAO WHERE FILME_ID = @FILME_ID)
            RAISERROR('Erro: Nenhuma avaliação registrada para este filme.', 16,
1);

        SELECT AVG(NOTA) AS MEDIA_NOTAS
        FROM AVALIACAO
    END TRY

```

```
WHERE FILME_ID = @FILME_ID;

COMMIT;
END TRY
BEGIN CATCH
    ROLLBACK;
    PRINT ERROR_MESSAGE();
END CATCH
END;
GO
```

```
EXEC PR_MEDIA_NOTA_FILME @FILME_ID = 1;
GO
```