

```

1  USE [master]
2  GO
3
4  SET ANSI_NULLS ON
5  GO
6  SET QUOTED_IDENTIFIER ON
7  GO
8
9  ALTER PROCEDURE [dbo].[SP_Cria_DB_XPERIUN_HealthLab]
10 AS
11 BEGIN
12     -- Declaração das variáveis
13     DECLARE
14         @PastaDB NVARCHAR(80), -- Pasta onde será criado o Banco de Dados
15         @PastaArqs NVARCHAR(80), -- Pasta onde estão os arquivos a serem importados
16         @ExtArqs NVARCHAR(4), -- Extensão dos arquivos a serem importados
17         @DATABASE NVARCHAR(20), -- Nome do Banco de Dados
18         @DB_TB_NOME NVARCHAR(MAX), -- Concatena @DATABASE + '.dbo' + @TB_NOME
19         @CMD_SQL NVARCHAR(MAX), -- Variável que contém comando a ser executado
20         @QTD_TABELAS INT, -- Quantidade Total de Tabelas a serem criadas
21         @NUM_TABELA INT, -- Numero da Tabela
22         @TB_NOME NVARCHAR(30), -- Nome da Tabela a ser criada
23         @NOME_ARQ NVARCHAR(30), -- Nome do arquivo a ser importado
24         @NOME_COL NVARCHAR(MAX), -- Definição dos Campos na Tabela Virtual
25         @LSTCAMPOS
26
27     -- Configuração do nome do Banco de Dados e Diretórios
28     SET @PastaDB = 'D:\SGBD\MS-SQL-Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\'
29     SET @PastaArqs = 'D:\Users\User\OneDrive\Git-Dados\Projetos\Teste\'
30     SET @ExtArqs = '.CSV'
31     SET @DATABASE = 'DB_Testes'
32
33     -- Criação do Banco de Dados
34     IF NOT EXISTS (SELECT 1 FROM sys.databases WHERE [name] = @DATABASE)
35     BEGIN
36         SET @CMD_SQL = N'
37         CREATE DATABASE ' + @DATABASE + ' ON (
38         NAME = ''' + @DATABASE + '_Data'',
39         FILENAME = ''' + @PastaDB + @DATABASE + '.mdf'',
40         SIZE = 5MB,
41         MAXSIZE = 25MB,
42         FILEGROWTH = 15%
43         )
44         LOG ON (
45         NAME = ''' + @DATABASE + '_Log'',
46         FILENAME = ''' + @PastaDB + @DATABASE + '_log.ldf'',
47         SIZE = 5MB,
48         MAXSIZE = 25MB,
49         FILEGROWTH = 5MB
50         )'
51         EXEC sp_executesql @CMD_SQL
52         IF @@ERROR <> 0
53         BEGIN
54             PRINT 'NÃO foi possível criar o Banco de dados: ' + @DATABASE +
55             '. Favor verificar com o Responsável.'
56             RETURN
57         END
58         ELSE
59             PRINT 'Banco de dados: ' + @DATABASE + ' criado com sucesso!'
60         END
61         ELSE
62             PRINT 'Banco de dados: ' + @DATABASE + ' já EXISTE!'
63
64     -- Criação da Tabela Virtual com Relação de Tabelas
65     DECLARE
66         @LST_TABELAS
67     TABLE (
68         ID_Tabela INT,
69         NomeTabela VARCHAR(60),
70         NomeArqImportar VARCHAR(100)
71     )
72     INSERT INTO
73     @LST_TABELAS

```

```

72  -->VALUES
73  -->(1, 'T_Classe_Procedimento', 'tab_classe_procedimento')
74
75  -->IF @@ERROR <> 0
76  -->BEGIN
77  -->PRINT 'NÃO foi possível criar Tabela Virtual: Relação das Tabelas.
      Favor verificar com o Responsável.'
78  -->RETURN
79  -->END
80  -->PRINT 'Tabela Virtual: Relação das Tabelas criada com sucesso!'
81
82  -->-- Criação da Tabela Virtual com Relação de Campos
83  -->DECLARE
84  -->@LST_CAMPOS
85  -->TABLE (
86  -->    >ID_TabelaCampo INT,
87  -->    >Nome_DefCampo NVARCHAR(MAX)
88  -->)
89  -->INSERT INTO
90  -->@LST_CAMPOS
91  -->VALUES
92  -->    >-- T_Classe_Procedimento
93  -->    >(1, '[Cod_ClasseProc] [int] NOT NULL, ' +
94  -->    >'[Descr_ClasseProc] [nvarchar] (15) NOT NULL'
95  -->    >)
96  -->IF @@ERROR <> 0
97  -->BEGIN
98  -->PRINT 'NÃO foi possível criar Tabela Virtual: Relação dos Campos das
      Tabelas. Favor verificar com o Responsável.'
99  -->RETURN
100 -->END
101 -->PRINT 'Tabela Virtual : Relação dos Campos das Tabelas criada com sucesso!'
102
103 -->-- Loop de Criação de Tabelas
104 -->SET @NUM_TABELA = 1
105 -->SELECT @QTD_TABELAS = COUNT(*) FROM @LST_TABELAS
106 -->
107 -->WHILE @NUM_TABELA <= @QTD_TABELAS
108 -->BEGIN
109 -->    >SELECT @TB_NOME = NomeTabela, @NOME_ARQ = NomeArqImportar FROM
      @LST_TABELAS WHERE ID_Tabela = @NUM_TABELA
110 -->    >SELECT @NOME_COL = Nome_DefCampo FROM @LST_CAMPOS WHERE
      ID_TabelaCampo = @NUM_TABELA
111
112 -->    >SET @DB_TB_NOME = QUOTENAME(@DATABASE) + '.' + QUOTENAME('dbo') + '.'
      + QUOTENAME(@TB_NOME)
113
114 -->    >-- Verifica se a tabela existe. Se não existir Cria
115 -->    >SET @CMD_SQL = N'
116 -->    >IF NOT EXISTS (SELECT 1 FROM ' + QUOTENAME(@DATABASE) +
      N'.sys.tables WHERE name = ' + QUOTENAME(@TB_NOME) + N'
117 -->    >BEGIN
118 -->    >    >PRINT 'A tabela: ' + @DB_TB_NOME + ' não existe'
119 -->    >
120 -->    >DECLARE @SQL NVARCHAR(MAX)
121 -->    >SET @SQL = N'CREATE TABLE ' + QUOTENAME(@DB_TB_NOME) +
      N' (' + @NOME_COL + N') '
122 -->
123 -->    >EXEC sp_executesql @SQL
124 -->
125 -->    >IF @@ERROR <> 0
126 -->    >BEGIN
127 -->    >    >PRINT 'NÃO foi possível criar Tabela: ' +
      @TB_NOME + '. Favor verificar com o Responsável.'
128 -->    >RETURN
129 -->    >END
130 -->    >ELSE
131 -->    >    >PRINT 'Tabela: ' + @TB_NOME + ' criada com sucesso!'
132 -->    >END
133 -->    >ELSE
134 -->    >    >PRINT 'Tabela: ' + @TB_NOME + ' já existe.'
135 -->    >EXEC sp_executesql @CMD_SQL
136

```

```

137  ----->-- Importa os arquivos CSV
138  ----->SET @CMD_SQL =
139  ----->'BULK INSERT ' +
140  ----->@TB_NOME +
141  ----->' FROM ' +
142  ----->' "' + @PastaArqs + @NOME_ARQ + @ExtArqs + "' +
143  ----->' WITH (
144  ----->FIELDTERMINATOR = ',',',',
145  ----->ROWTERMINATOR = '\\n',
146  ----->FIRSTROW = 2,
147  ----->CODEPAGE = '65001',
148  ----->FORMAT = 'CSV'
149  ----->)'
150  ----->PRINT @CMD_SQL
151  ----->SET @NUM_TABELA += 1
152  ----->END
153  ----->PRINT 'Rotina finalizada!'
154  END

```