**Criando o banco de dados e a Stored Procedure**

Vamos criar um banco de dados como exemplo para mostrar a criação da nossa stored procedure.

Eu poderia usar o SQL Server Management Studio ou outra ferramenta mas vou usar o Visual Basic Express 2008 Edition.

Abra o Visual Basic Express 2010 Edition e no menu View clique em Other Windows -> DataBase Explorer para exibir a janela do Database Source exibindo as conexões existentes:

|  |
| --- |
| http://www.macoratti.net/13/04/vbn_stor1.gif |

A seguir clique com o botão direito do mouse sobre **Data Connections** e a seguir em **Add Connection...**

Na janela **Add Connection** escolha a opção para definir o **Data Source**. Estou usando a opção **Microsoft SQL Server DataBase File;**

Informe o nome do banco de dados que no exemplo será chamado de **Cadastro** e clique no botão OK;

|  |
| --- |
| http://www.macoratti.net/13/04/vbn_stor2.gif |

Será solicitada a confirmação para criar o banco de dados. Clique no botão **Sim**;

|  |
| --- |
| http://www.macoratti.net/13/04/vbn_stor3.gif |

O banco de dados será criado e exibido na janela**DataBase Source**. Vamos agora criar a tabela usada como exemplo no artigo.

Clique com o botão direito sobre **Tables** e selecione **Add New Table**;

|  |
| --- |
| http://www.macoratti.net/13/04/vbn_stor5.gif |

A seguir vamos criar a tabela **Funcionarios** com a estrutura mostrada na figura abaixo:

|  |
| --- |
| http://www.macoratti.net/13/04/vbn_stor6.gif |

Para incluir alguns dados na tabela clique com o botão direito sobre a tabela e selecione **Show Table Data** e digite os valores diretamente na tabela.

|  |
| --- |
| http://www.macoratti.net/13/04/vbn_stor7.gif |

Agora que já temos o banco de dados**Cadastro.mdf** e a tabela **Funcionarios** já podemos criar nossa super poderosa Stored Procedure usando os comandos T-SQL

Clique com o botão direito do mouse sobre **Stored Procedures** e a seguir clique em**Add New Stored Procedure;**

Será aberta a janela para criação da stored procedure com um esqueleto da mesma já usando o comando **CREATE PROCEDURE**;

|  |  |
| --- | --- |
| http://www.macoratti.net/13/04/vbn_stor8.gif | http://www.macoratti.net/13/04/vbn_stor9.gif |

Abaixo temos a stored procedure **ManutencaoFuncionario** criada no banco de dados **Cadastro.mdf** e que permite i*ncluir(Insert), selecionar(Select), atualizar (Update) e deletar (Delete)* registros da tabela **Funcionarios**.

|  |
| --- |
| http://www.macoratti.net/13/04/vbn_stora.gif |

**Criando a aplicação Windows Forms para testar a Stored Procedure**

Eu vou usar um projeto Windows Forms bem simples apenas para mostrar que podemos usar a nossa stored procedure para realizar as operações **CRUD** em uma tabela do SQL Server;

Ainda no Visual Basic 2008 Express Edition crie um novo projeto do tipo Windows Forms Application com o nome **ManutencaoFuncionarios**;

Defina o formulário form1.vb os controles conforme mostram a figura abaixo onde temos:

* 4 Labels
* 4 TextBox - txtCodigo, txtNome, txtSobrenome e txtEmail
* 4 Buttons - btnIncluir,btnAtualizar, btnExcluir e btnSelecionar
* 1 DataGridView - gdvFuncionarios

|  |
| --- |
| http://www.macoratti.net/13/04/vbn_storb.gif |

Defina o seguinte namespace no formulário **form1.vb:**

|  |
| --- |
| Imports System.Data.SqlClient |

A seguir, logo após a declaração da classe form1 defina as variáveis abaixo que representam a string de conexão com o banco de dados e a conexão **SqlConnection**:

|  |
| --- |
| Dim strSql As String = "Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\f5361091\Documents\Cadastro.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True" Dim conexao As SqlConnection |

Agora em cada evento **Click** dos botões de comando vamos definir o respectivo código que usa a nossa Stored Procedure para a finalidade específica.

1- No evento **Click** do botão de comando **btnIncluir** temos o código onde usamos a stored procedure **ManutencaoFuncionario** para incluir um novo registro:

|  |
| --- |
| Private Sub btnIncluir\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnIncluir.Click         Try             'cria a conexão com o banco de dados Teste             conexao = New SqlConnection(strSql)             conexao.Open()             Dim cmd As New SqlCommand("**ManutencaoFuncionario**", conexao)             'define o tipo stored procedure             c**md.CommandType = CommandType.StoredProcedure**             'atribui os valores dos parametros             cmd.Parameters.AddWithValue("@Action", "Insert")             cmd.Parameters.AddWithValue("@nome", txtNome.Text)             cmd.Parameters.AddWithValue("@sobrenome", txtSobrenome.Text)             cmd.Parameters.AddWithValue("@email", txtEmail.Text)             'executa a stored procedure            **cmd.ExecuteNonQuery()**             MsgBox("Dados incluídos com sucesso!")         Catch ex As Exception             MsgBox("Erro : " & ex.Message)         Finally             conexao.Close()         End Try     End Sub |

2- No evento **Click** do botão de comando **btnAtualizar** temos o código onde usamos a stored procedure **ManutencaoFuncionario** para atualizar um registro:

|  |
| --- |
| Private Sub btnAtualizar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAtualizar.Click         Try             'cria a conexão com o banco de dados Teste             conexao = New SqlConnection(strSql)             conexao.Open()             Dim cmd As New SqlCommand("**ManutencaoFuncionario**", conexao)             'define o tipo stored procedure            **cmd.CommandType = CommandType.StoredProcedure**             'atribui os valores dos parametros             cmd.Parameters.AddWithValue("@Action", "Update")             cmd.Parameters.AddWithValue("@id", txtCodigo.Text)             cmd.Parameters.AddWithValue("@nome", txtNome.Text)             cmd.Parameters.AddWithValue("@sobrenome", txtSobrenome.Text)             cmd.Parameters.AddWithValue("@email", txtEmail.Text)             'executa a stored procedure       **cmd.ExecuteNonQuery()**             MsgBox("Dados atualizado com sucesso!")         Catch ex As Exception             MsgBox("Erro : " & ex.Message)         Finally             conexao.Close()         End Try     End Sub |

3- No evento **Click** do botão de comando **btnExcluir** temos o código onde usamos a stored procedure **ManutencaoFuncionario** para excluir um registro:

|  |
| --- |
| Private Sub btnExcluir\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnExcluir.Click         Try             'cria a conexão com o banco de dados Teste             conexao = New SqlConnection(strSql)             conexao.Open()             Dim cmd As New SqlCommand("**ManutencaoFuncionario**", conexao)             'define o tipo stored procedure            **cmd.CommandType = CommandType.StoredProcedure**             'atribui os valores dos parametros             cmd.Parameters.AddWithValue("@Action", "Delete")             cmd.Parameters.AddWithValue("@id", txtCodigo.Text)             'executa a stored procedure            **cmd.ExecuteNonQuery()**             MsgBox("Dados excluídos com sucesso!")         Catch ex As Exception             MsgBox("Erro : " & ex.Message)         Finally             conexao.Close()         End Try     End Sub |

4- No evento **Click** do botão de comando **btnSelecionar** temos o código onde usamos a stored procedure **ManutencaoFuncionario** para selecionar os registros da tabela **Funcionarios**:

|  |
| --- |
| Private Sub btnSelecionar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSelecionar.Click         Try             'cria a conexão com o banco de dados Teste             conexao = New SqlConnection(strSql)             conexao.Open()             'especifia a stored procedure usada             Dim da As New SqlDataAdapter("**ManutencaoFuncionario**", conexao)             'define o tipo stored procedure             da.SelectCommand.CommandType =**CommandType.StoredProcedure**             'define o parametro             da.SelectCommand.Parameters.Add(New SqlParameter("@Action", SqlDbType.VarChar, 10))             'Atribui o valor do parametro             da.SelectCommand.Parameters("@Action").Value = "Select"             'cria e preenche o dataset         **Dim ds As New DataSet             da.Fill(ds, "Funcionarios")**             'exibe o rsutlado           **gdvFuncionarios.DataSource = ds.Tables(0)**         Catch ex As Exception             MsgBox("Erro : " & ex.Message)         Finally             conexao.Close()         End Try     End Sub |

Em cada operação definimos os parâmetros necessários para que a stored procedure seja executada;

Abaixo vemos um exemplo de execução do projeto onde estamos incluindo um novo registro:

|  |
| --- |
| http://www.macoratti.net/13/04/vbn_storc.gif |