

Conexão Java/MySql – parte 3 (botões manutenção dados e pesquisa)

1 – Botões de ação para efetuar a “Manutenção dos dados”: novo, cadastrar, alterar e excluir:

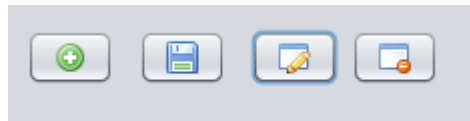


Figura 1 - Botões para Manutenção dos registros

a) Botão Novo:

Função: Limpar as caixas de texto para novas entradas de dados.

```
418 private void BtnNovoActionPerformed(java.awt.event.ActionEvent evt) {  
419     txtCod.setText(""); // limpa a caixa de texto em questão  
420     txtNome.setText("");  
421     txtNasc.setText("");  
422     txtFone.setText("");  
423     txtEmail.setText("");  
424     txtCod.requestFocus(); // posiciona o cursor neste campo para digitação  
425 }
```

Figura 2 - Código do botão Novo

b) Botão Cadastrar:

Função: Incluir na tabela um novo registro a partir dos dados digitados nas caixas de texto:

```
303 private void btnGravarActionPerformed(java.awt.event.ActionEvent evt) {  
304     String nome = txtNome.getText();  
305     String data_nasc = txtNasc.getText();  
306     String telefone = txtFone.getText();  
307     String email = txtEmail.getText();  
308  
309     try {  
310         String insert_sql="insert into tbclientes (nome,telefone, email, dt_nasc) values ('" + nome + "','" + telefone + "','" + email +  
311         con_cliente.statement.executeUpdate(insert_sql);  
312         JOptionPane.showMessageDialog(null,"Gravação realizada com sucesso!!","Mensagem do Programa",JOptionPane.INFORMATION_MESSAGE);  
313         con_cliente.executaSQL("select * from tbclientes order by cod");  
314         con_cliente.resultset.first();  
315         preencherTabela();  
316         mostrar_Dados();  
317     }  
318     catch(SQLException errosql){  
319         JOptionPane.showMessageDialog(null,"\n Erro na gravação :\n "+errosql,"Mensagem do Programa",JOptionPane.INFORMATION_MESSAGE);  
320     }  
321 }
```

Figura 3 - Código do botão de Cadastro

Linha 310 completa:

String insert_sql="insert into tbclientes (nome,telefone, email, dt_nasc) values ('" + nome + "','" + telefone + "','" + email + "','" + data_nasc + "')";

- **Linhas 304 a 307:** declara as variáveis e atribui a elas o valor de cada caixa de texto respectivamente, através do método “getText()”.
- **Linha 310:** define o comando SQL de inserção, onde são informados nome da tabela, campos, e, na sequência as variáveis resultantes das caixas de texto, respectivamente.
- **Linha 311:** executa (no ambiente Java - virtualmente) o comando de inserção.

- **Linha 313:** executa (no ambiente Java - virtualmente) o comando de seleção dos dados da tabela, e na próxima linha posiciona no 1º registro.
- **Linha 315:** executa o método “preencherTabela”, responsável por apresentar dados na Jtable.
- **Linha 316:** executa o método “mostrar_Dados”, responsável por apresentar dados nas caixas de texto.

c) Botão Alterar:

Função: Alterar na tabela os dados de um registro a partir dos dados digitados nas caixas de texto. Para encontrar o registro na tabela será usado o campo “código”, onde acontecerá um filtro e as alterações só ocorrerão quando o SQL encontrar na tabela o registro que tenha o código coincidente.

```

323 private void btnAlterarActionPerformed(java.awt.event.ActionEvent evt) {
324     String nome = txtNome.getText();
325     String data_nasc = txtNasc.getText();
326     String telefone = txtFone.getText();
327     String email = txtEmail.getText();
328     String sql="";
329     String msg="";
330
331     try {
332         if(txtCod.getText().equals("")){
333             sql="insert into tblclientes (nome,telefone, email, dt_nasc) values ('" + nome + "','" + telefone + "','" + email + "','" + data_nasc + "')";
334             msg="Gravação de um novo registro";
335         }else{
336             sql="update tblclientes set nome='" + nome + "',telefone='" + telefone + "', email='" + email + "', dt_nasc='" + data_nasc + "' where cod = " + txtCod.getText();
337             msg="Alteração de registro";
338         }
339
340         con_cliente.statement.executeUpdate(sql);
341         JOptionPane.showMessageDialog(null,msg+" realizada com sucesso!!","Mensagem do Programa",JOptionPane.INFORMATION_MESSAGE);
342
343         con_cliente.executaSQL("select * from tblclientes order by cod");
344         con_cliente.resultset.first();
345         preencherTabela();
346         mostrar_Dados();
347
348     }catch(SQLException errosql){
349         JOptionPane.showMessageDialog(null,"\n Erro na gravação :\n "+errosql,"Mensagem do Programa",JOptionPane.INFORMATION_MESSAGE);
350     }
351 }

```

Figura 4 - Código do botão da Alteração

Esta rotina é bem parecida com a anterior:

- **Linhas 324 a 327:** declara as variáveis e atribui a elas o valor de cada caixa de texto respectivamente, através do método “getText()”.
- **Linha 332,** possui um if porque se o usuário não digitar nenhum código na caixa de texto, então este botão fará uma inclusão e não alteração.
- **Linha 336:** define o comando SQL de alteração, onde é informado o nome da tabela, e, após o “set” estão sendo feitas atribuições para cada campo com sua respectiva variável, extraída das caixas de texto. É nesta hora que os atributos passam a ter o novo valor.
- **Linha 340:** executa (no ambiente Java - virtualmente) o comando de inserção e/ou alteração.
- **Linha 343:** executa (no ambiente Java - virtualmente) o comando de seleção dos dados da tabela, e na próxima linha posiciona no 1º registro.
- **Linha 345:** executa o método “preencherTabela”, responsável por apresentar dados na Jtable.
- **Linha 346:** executa o método “mostrar_Dados”, responsável por apresentar dados nas caixas de texto.

d) Botão Excluir:

Função: Excluir na tabela um registro a partir do código da caixa de texto. Para encontrar o registro na tabela será usado o campo “código”, onde acontecerá um filtro e a exclusão só ocorrerá quando o SQL encontrar na tabela o registro que tenha o código coincidente.

```
353 private void btnExcluirActionPerformed(java.awt.event.ActionEvent evt) {  
354     String sql="";  
355     try {  
356         int resposta = JOptionPane.showConfirmDialog(rootPane, "Deseja excluir o registro: ", "Confirmar Exclusão", JOptionPane.YES_NO_OPTION, 3);  
357         if (resposta==0){  
358             sql = "delete from tbclientes where cod = " + txtCod.getText();  
359             int excluir = con_cliente.statement.executeUpdate(sql);  
360             if (excluir==1){  
361                 JOptionPane.showMessageDialog(null, "Exclusão realizada com sucesso!!", "Mensagem do Programa", JOptionPane.INFORMATION_MESSAGE);  
362                 con_cliente.executeQuery("select * from tbclientes order by cod");  
363                 con_cliente.resultset.first();  
364                 preencherTabela();  
365                 posicionarRegistro();  
366             }  
367             else{  
368                 JOptionPane.showMessageDialog(null, "Operação cancelada pelo usuário!!", "Mensagem do Programa", JOptionPane.INFORMATION_MESSAGE);  
369             }  
370         }  
371     } catch (SQLException excecacao){  
372         JOptionPane.showMessageDialog(null, "Erro na exclusão: "+excecacao, "Mensagem do Programa", JOptionPane.INFORMATION_MESSAGE);  
373     }  
374 }  
375
```

Figura 5 - Código do botão da Exclusão

Aqui nesta rotina, será feita a pergunta ao usuário se realmente “deseja excluir”, se sim, então a exclusão será oficializada.

- **Linha 358:** executa (no ambiente Java - virtualmente) o comando de deletar registros, onde o parâmetro para o filtro de registros será o conteúdo da caixa de texto do código.

A sequência é igual as rotinas anteriores.

2 – Pesquisa de um determinado registro:

Função: Filtrar os registros da JTable, mostrando somente aqueles que atendem as teclas digitadas nesta caixa de texto.

a) Desenhar no formulário, abaixo da tabela, estes objetos:

Pesquisa por nome do Cliente:

Figura 6 - Pesquisar Cliente

Exemplo 1: Sem filtro, ou seja, sem digitação na caixa de texto (apresenta todos registros):

Código	Nome	Data Nascim...	Telefone	Email
1	Antonio F. Cruz	17/03/1939	1111-1111	antonio@antonio.com.br
2	Edneia R F	27/03/1969	2222-2222	edneia@edneia.com.br
3	Gustavo Alameda	27/05/2000	3333-3333	hustavo@gustavo.com.br
4	Ana Maria da Silva	01/01/2001	4444-4444	ana@aaa.com.br
5	Bento Gonçalves	02/02/2002	5555-5555	bento@bbb.com.br
6	Guilherme Souza e Silva	01/01/2001	6666-6666	guigui@gui.com
8	Edneia R F	27/03/1969	2222-2222	edneia@edneia.com.br

Pesquisa por nome do Cliente:

Figura 7 - Tabela sem filtro de pesquisa

Exemplo 2: Com filtro, ou seja, com digitação da letra “g” na caixa de texto (apresenta somente os registros cujo nome comece com a letra “g”:

Código	Nome	Data Nascim...	Telefone	Email
3	Gustavo Alameda	27/05/2000	3333-3333	hustavo@gustavo.com.br
6	Guilherme Souza e Silva	01/01/2001	6666-6666	guigui@gui.com
10	Gustavo Alameda hhhh	27/05/2000	3333-3333	hustavo@gustavo.com.br

Pesquisa por nome do Cliente:

Figura 8- Tabela com filtro através da pesquisa da letra G

b) Acione o evento “KeyReleased” da caixa de texto:

```

434 private void txt_pesquisaKeyReleased(java.awt.event.KeyEvent evt) {
435     try{
436         String pesquisa = "select * from tbclientes where nome like '" + txt_pesquisa.getText() + "%'";
437         con_cliente.executaSQL(pesquisa);
438
439         if(con_cliente.resultset.first()){
440             preencherTabela();
441         }
442         else{
443             JOptionPane.showMessageDialog(null, "\n Não existe dados com este parâmetro!!", "Mensagem do Programa", JOI
444         }
445     }catch(SQLException errosql){
446         JOptionPane.showMessageDialog(null, "\n Os dados digitados não foram localizados!! :\n "+errosql, "Mensagem de
447     }
448 }
449

```

Este evento executa o código sempre que um novo caractere for digitado. Então a cada letra teclada a pesquisa na tabela é feita através do filtro da linha 436:

where nome like ''' + txt_pesquisa.getText() + "%'";

O percentual significa que “qualquer complemento” será colocado a(s) letra(s) digitadas. Então digitado a letra “G”, o comando filtrou todos nomes começados com “G”.


- Linha 443 completa:

JOptionPane.showMessageDialog(null, "\n Não existe dados com este parâmetro!!", "Mensagem do Programa", JOptionPane.INFORMATION_MESSAGE);

- Linha 446 completa:

JOptionPane.showMessageDialog(null, "\n Os dados digitados não foram localizados!! :\n "+errosql, "Mensagem do Programa", JOptionPane.INFORMATION_MESSAGE);

3 – Formulário finalizado:

 Cadastro de Clientes










Código:

Nome:

Data Nascimento:

Telefone:

Email:

Código	Nome	Data Nascim...	Telefone	Email
1	Antonio F. Cruz	17/03/1939	1111-1111	antonio@antonio.com.br
2	Edneia R F	27/03/1969	2222-2222	edneia@edneia.com.br
3	Gustavo Alameda	27/05/2000	3333-3333	hustavo@gustavo.com.br
4	Ana Maria da Silva	01/01/2001	4444-4444	ana@aaa.com.br
5	Bento Gonçalves	02/02/2002	5555-5555	bento@bbb.com.br
6	Guilherme Souza e Silva	01/01/2001	6666-6666	guigui@gui.com
8	Edneia R F	27/03/1969	2222-2222	edneia@edneia.com.br

Pesquisa por nome do Cliente:

Figura 9- Formulário Completo

Agora cabe a você deixar o visual mais bonito, para isto, use a criatividade.