**ATIVIDADE 11-2. TRABALHANDO COM CONTROLES DATA BOUND**

Nesta atividade, você irá se começar a se familiarizar com o seguinte:

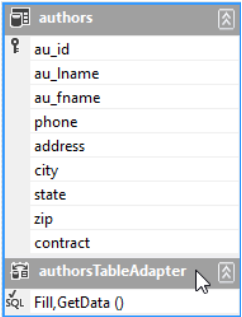
* Unindo um DataGrid para um DataTable
* Atualizando os dados usando um DataAdapter

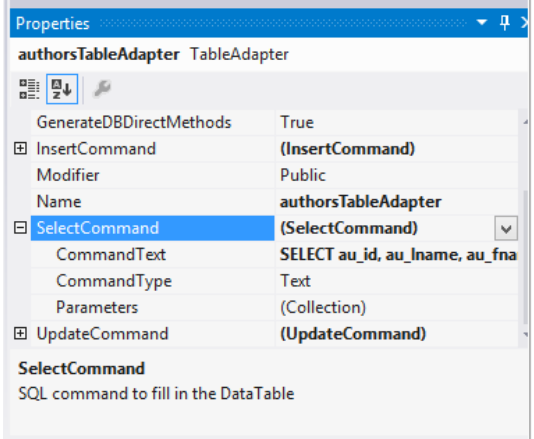
**Unindo um DataGrid para um DataTable**

Para unir um DataGrid para um objeto DataTable, siga estes passos:

**Criando um DataSet**

1. Inicie o *Visual Studio*. Selecione *File* -> *New* ->*Project*.
2. Escolhe a aplicação *WPF*. Renomeie o projeto para *Activity11\_2* e clique no botão *OK*.
3. Depois de carregar o projeto, localize a janela *Data Sources* localizado no lado esquerdo da tela. Clique no link *Add New Data Source*.
4. Na caixa de configuração *Data Source Configuration*, escolha um tipo de *data source* do banco de dados *(database)* e clique em *Next* para continuar.
5. Escolha na janela *Choose a Database Model*, selecione o *DataSet* e clique em *Next*.
6. Na janela *Choose your Data Connection*, selecione ou crie uma conexão para o banco de dados *(database) Pubs* e clique em *Next*.
7. Na próxima tela, salve a conexão para o arquivo de configuração.
8. Na janela *Choose Your Database Objects*, expanda a janela *Tables* e selecione a tabela *authors*.
9. Aviso que a janela *Solutions Explorer* um arquivo *pubsDataSet.xsd* tem sido adicionado para o arquivo. Este arquivo representa um objeto do *dataset* fortemente tipado baseado no banco de dados *(database) pubs*. Dê um duplo clique na *Solution Explorer* para lançar o editor visual do *dataset*.
10. O visual editor contém uma tabela *author*. Selecione a *authorsTableAdapter*, como mostra na figura abaixo. Na janela, avisa que os comandos *select*, *insert*, *update*, e o deIete tem sido gerado por você, como mostra a figura abaixo:





**Criando um visual de janela**

1. Abra o *MainWindow* na janela no *XAML Editor*. Troque o *title (título)* de *“Window”* para *“Phone List”*.
2. Dentro das tags Grid, adicione um controle *DockPanel*. Dentro do *DockPanel*, adicione uma *StackPanel*.

*<Grid>*

*<DockPanel>*

*<StackPanel DockPanel.Dock=“Top” Orientation=“Horizontal”>*

*</StackPanel>*

*</DockPanel>*

*</Grid>*

1. Dentro do *StackPanel*, adicione dois botões - um para *getting data (obtendo dados)* e um para *updating data (atualizando dados)*. Adicione o *handler* do evento *click* para cada botão.

*<StackPanel DockPanel.Dock=“Top” Orientation=“Horizontal”>*

*<Button Name=“btnGetData” Content=“Get Data” Click=“btnGetData\_Click” />*

*<Button Name=“btnSaveData” Content=“Save Data” />*

*</StackPanel>*

1. Fora do *StackPanel*, mas dentro do *DockPanel*, adicione um *DataGrid*.

*<DataGrid Name=“dgAuthors” AutoGenerateColumns = “True” DockPanel.Dock=“Botton” />*

*</DockPanel>*

*</Grid>*

**Carrega o DataGrid**

1. Abra o arquivo *MainWindow.xaml.cs* na janela de Editor de código.
2. Adicione três variáveis de nível de classe do tipo *pubsDataSet*, *authorsTableAdapter* e *authorsDataTable*.

*public partial class MainWindow : Window*

*{*

*pubsDataSet \_dsPubs;*

*pubsDataSetTableAdapters.authorsTableAdapter \_taAuthors;*

*pubsDataSet.authorsDataTable \_dtAuthors;*

1. No evento *btnGetData\_Click*, adicione o código para preencher a tabela *\_taAuthor* e este conjunto é igual para o *DataContext* do grid *dgAuthors*.

*private void btnGetData\_Click(object sender, RoutedEventArgs e)*

*{*

*\_dsPubs = new pubsDataSet();*

*\_taAuthors = new pubsDataSetTableAdapters.authorsTableAdapter();*

*\_dtAuthors = new pubsDataSet.authorsDataTable();*

*\_taAuthors.Fill(\_dtAuthors);*

*this.dgAuthors.DataContext = \_dtAuthors;*

*}*

1. Troque de volta para o *“XAML Editor Window”* e adicione o *binding ItemsSources* para o código XAML do *DataGrid*. Isto irá unir isto para o *DataContext*.

*<DataGrid Name=“dgAuthors” AutoGenerateColumns=“True”*

*DockPanel.Dock=“Bottom” ItemsSource=“{Binding}” />*

1. Selecione *Debug* -> *Start*. Teste a aplicação clicando o botão *Get Data*. O *DataGrid* irá carregar com os dados do *Authors* (autores), veja na figura abaixo. Aviso que desde a propriedade *AutoGenerateColumns* do *DataGrid* é um conjunto para verdade, e o *grid* carrega com todas as colunas na tabela. Os cabeçalhos das colunas do grid também são os mesmos nomes das colunas da tabela *authors* (autores).
2. Depois de visualizar a janela, pare o depurador.



**Atualizando os dados**

1. Abra o arquivo *MainWindow.xaml.cs* na janela *Code Editor*. Adicione o seguinte código para atualizar os dados no evento *handler* *btnSaveData\_Click*. Este código usa os *table adapters* do comando *update* para enviar as mudanças de volta para o *banco de dados (database)*. Você deve também alterar o *XAML* do *btnSaveData’s* para chamar este evento quando isto é clicado.

*private void btnSaveData\_Click(object sender, RoutedEventArgs e)*

*{*

*try*

*{*

*\_taAuthors.Update(\_dtAuthors);*

*MessageBox.Show(“Data Saved.”,*

*“Information”, MessageBoxButton.OK, MessageBoxImage.Information);*

*}*

*catch (Exception ex)*

*{*

*MessageBox.Show(“Could not save data!”,*

*“Warning”, MessageBoxButton.OK,*

*MessageBoxImage.Warning);*

*}*

*}*

1. Atualize do código *Grid´s XAML* para apenas mostrar o *first name* (primeiro nome), *last name* (útlimo nome) e a coluna de *phone* (telefone).

*<DataGrid Name=“dgAuthors” AutoGenerateColumns=“False”*

*DockPanel.Dock=“Bottom” ItemsSource=”{Binding}”>*

*<DataGrid.Columns>*

*<DataGridTextColumns Header=”Last Name”*

*Binding=“{Binding Path=’au\_lname’}” />*

*<DataGridTextColumns Header=”First Name”*

*Binding=“{Binding Path=’au\_fname’}” />*

*<DataGridTextColumns Header=”Phone”*

*Binding=“{Binding Path=’phone’}” />*

*</DataGrid.Columns>*

*<DataGrid>*

1. Selecione *Debug* -> *Start*. Teste a aplicação clicando no botão *“Get Data”*. Atualize alguns dos nomes. Clique no botão *“Save Data”* e então clique no botão *“Get Data”* para verificar os nomes quando salvados para o *banco de dados (database)*.
2. Depois de testar, feche o depurador e saia do *Visual Studio*.