

3.2. Làm việc với GridView trong ASP.NET

GridView là điều khiển hiển thị dữ liệu giàu tính năng nhất trong ASP.NET. Bên cạnh việc hiển thị dữ liệu, ta có thể thực hiện các thao tác như select, sort, page, và edit.

3.2.1. Phân trang và sắp xếp

Khi ứng dụng thực hiện trên lượng dữ liệu lớn, việc hiển thị toàn bộ dữ liệu sẽ không hiệu quả. Paging là kỹ thuật phân trang, chia các bản ghi dữ liệu thành những trang riêng biệt. Trong điều khiển DataGrid, kích thước mặc định mỗi trang là 10 bản ghi

Phân trang: DataGrid, GridView hỗ trợ 2 chế độ phân trang

Default paging và Custom paging

- Chế độ Default paging, toàn bộ dữ liệu được nạp từ nguồn và DataGrid lấy toàn bộ dữ liệu, phân trang theo kích thước mặc định (10) và hiển thị.
- Chế độ custom paging, ta phải viết chương trình lấy từng nhóm bản ghi để hiển thị trong mỗi trang và kích thước trang có thể được thiết lập. Custom paging thực hiện hiệu quả và nhanh hơn default paging

Sử dụng chế độ default paging của GridView:

- AllowPaging = **true**

- PageSize = 10

- PageIndex = 0

- Các sự kiện thường xử lý

SelectedIndexChanged, PageIndexChanged: thiết lập PageIndex

GridView1.PageIndex= e.NewPageIndex;

Sắp xếp: DataGrid, GridView hỗ trợ sắp xếp theo thứ tự và tên trường

- AllowSorting = **true**. Tất cả các columns sẽ được thiết lập clickable

- Hàm sự kiện


```
protected void GridView1_Sorting(object sender, GridViewSortEventArgs e)
{
}
```

GridViewSortEventArgs: hỗ trợ SortDirection và SortExpression.

SortDirection Sắp xếp theo thứ tự ascending và descending.

SortExpression Tên column ta muốn sắp xếp.

Cấu trúc bảng CSDL Employee

	Column Name	Data Type	Allow Nulls
	EmpID	int	<input type="checkbox"/>
	Name	nvarchar(50)	<input checked="" type="checkbox"/>
	Gender	nvarchar(50)	<input checked="" type="checkbox"/>
	Salary	float	<input checked="" type="checkbox"/>
	Address	nvarchar(50)	<input checked="" type="checkbox"/>
	DEPID	varchar(50)	<input checked="" type="checkbox"/>

```
<asp:SqlDataSource ID="EmployeeDataSource" Runat="server"
SelectCommand="SELECT * FROM [Employee]"
ConnectionString= "<%%$ ConnectionStrings:EmpConnectionString %>">
</asp:SqlDataSource>
```

PagingAndSorting.aspx

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Web.UI.WebControls;

public partial class GridViewDemo : System.Web.UI.Page
{
    SqlConnection conn;
    SqlDataAdapter adapter;
    DataSet ds;
    string str="EmpID";

    protected void Page_Load(object sender, EventArgs e) {
        if(!IsPostBack)
        {
            FillGridView(str);
        }
    }

    protected void FillGridView(string str) {
        string cs = ConfigurationManager.ConnectionStrings
            ["conString"].ConnectionString;
```

```

try {
    conn = new SqlConnection(cs);
    adapter = new SqlDataAdapter
        ("select * from tblEmps order by "+str, conn);
    ds = new DataSet();
    adapter.Fill(ds);
    GridView1.DataSource = ds;
    GridView1.DataBind();
}
catch (Exception ex) {
    Label1.Text = "ERROR :: " + ex.Message;
}
finally {
    ds.Dispose();
    conn.Dispose();
}
}

protected void GridView1_Sorting(object sender, GridViewSortEventArgs e) {
    str = e.SortExpression;
    FillGridView(str);
}

protected void GridView1_PageIndexChanging(object sender,
    GridViewPageEventArgs e) {

    GridView1.PageIndex= e.NewPageIndex;
    FillGridView(str);
}
}

```

Đặt PageSize = 4, kết quả

```

<form id="form1" runat="server">
    <asp:GridView ID="GridView1" Runat="server"
        PageIndexChanged = "GridView1_PageIndexChanging"
        AutoGenerateColumns="False"
        AllowSorting="True" BorderWidth="2px" BackColor="White"
        GridLines="None" CellPadding="3"
        CellSpacing="1" BorderStyle="Ridge" BorderColor="White"
        AllowPaging="True" PageSize="4">
        <FooterStyle ForeColor="Black"
            BackColor="#C6C3C6"></FooterStyle>

```

```

<PagerStyle ForeColor="Black" HorizontalAlign="Right"
  BackColor="#C6C3C6"></PagerStyle>
<HeaderStyle ForeColor="#E7E7FF" Font-Bold="True"
  BackColor="#4A3C8C"></HeaderStyle>
<Columns>
  <asp:BoundField HeaderText="Emp ID"
    DataField="EmpID" SortExpression="EmpID">
  </asp:BoundField>
  <asp:BoundField HeaderText="Name"
    DataField="Name" SortExpression="Name"
    DataFormatString="{0:c}">
    <ItemStyle HorizontalAlign="Right"></ItemStyle>
  </asp:BoundField>
  <asp:BoundField HeaderText="Gender"
    DataField="Gender"
    SortExpression="Gender">
    <ItemStyle HorizontalAlign="Right"></ItemStyle>
  </asp:BoundField>
  <asp:BoundField HeaderText="Salary"
    DataField="Salary SortExpression="Salary"></asp:BoundField>
    </asp:BoundField>
  <asp:BoundField HeaderText="Dept ID"
    DataField="DEPID SortExpression="DEPID"></asp:BoundField>
</Columns>
<SelectedRowStyle ForeColor="White" Font-Bold="True"
  BackColor="#9471DE"></SelectedRowStyle>
<RowStyle ForeColor="Black" BackColor="#DEDFDE"></RowStyle>
</asp:GridView>

```

Bạn đang xem trang

<%=productsGridView.PageIndex + 1%> /

<%=productsGridView.PageCount%>

</form>

<u>EmpID</u>	<u>Name</u>	<u>Gender</u>	<u>Salary</u>	<u>Address</u>	<u>DEPID</u>
1	Raj	Male	25000	Pune	1
2	Ramesh	Male	20000	Nagpur	1
3	David	Male	35000	USA	2
4	Zeny	Female	42000	Canada	2
1 2					

Click trang tiếp theo

<u>EmpID</u>	<u>Name</u>	<u>Gender</u>	<u>Salary</u>	<u>Address</u>	<u>DEPID</u>
5	Nishant	Male	50000	Pune	3
6	Digvijay	Male	45000	New York	3
7	Nandani	Female	23000	USA	3
1 2					

Click tên trường để sắp xếp các bản ghi.

<u>EmpID</u>	<u>Name</u>	<u>Gender</u>	<u>Salary</u>	<u>Address</u>	<u>DEPID</u>
5	Nishant	Male	50000	Pune	3
6	Digvijay	Male	45000	New York	3
7	Nandani	Female	23000	USA	3
1 2					

3.2.2. Sử dụng các trường trong GridView

GridView các loại trường dữ liệu sau:

- BoundField
- CheckBoxField
- CommandField
- ButtonField
- HyperLinkField
- ImageField
- TemplateField

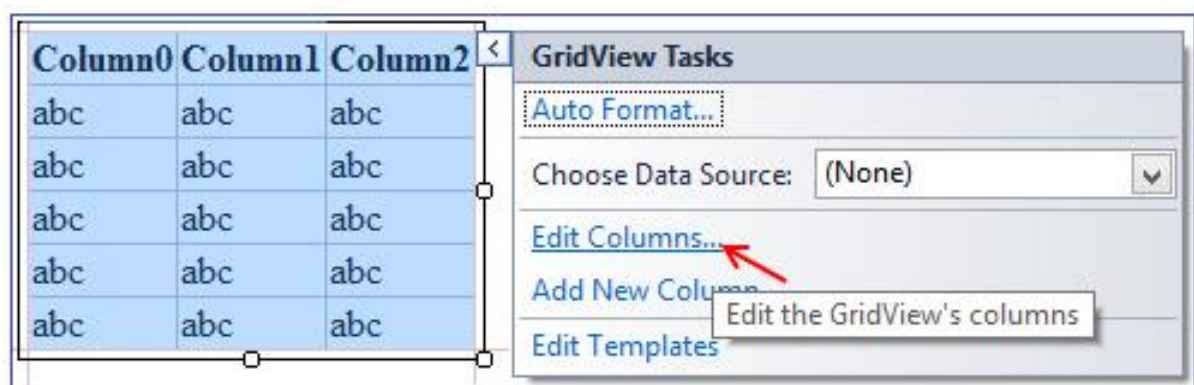
BoundField

BoundField là thuộc tính dùng để hiển thị giá trị một trường của bảng dữ liệu. BoundField cho phép ta điều khiển sự hiển thị của trường và các giá trị. Ví dụ, tên trường của bảng dữ liệu là EmpID nhưng ta có thể hiển thị trên trang web là Employee ID bằng thuộc tính HeaderText. Ta có thể thay đổi thứ tự các trường bằng cách thay đổi BoundField.

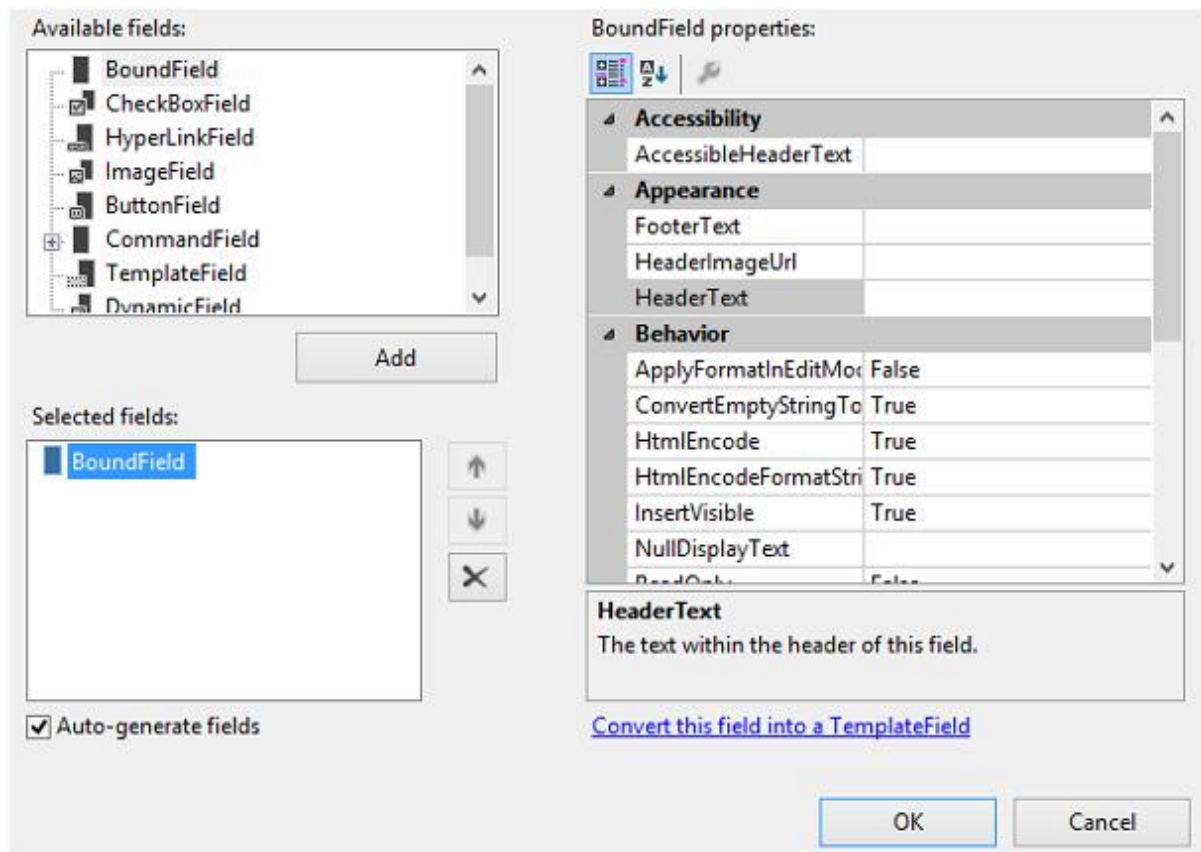
BoundField Class

- **DataField:** Liên kết với trường dữ liệu của bảng.
- **DataFormatString:** Định dạng chuỗi.
- **HeaderText:** Đặt nhãn cho trường.
- **ReadOnly:** Chỉ đọc không cập nhật.
- **Visible:** Xác định sự hiển thị.
- **NullDisplayText:** Hiển thị giá trị NULL nếu không có dữ liệu.

Sử dụng smart tag để thiết lập GridView BoundField



Chọn Edit rồi nhập HeaderText, DataField. Bỏ đánh dấu Auto-generate.



GridView hỗ trợ thuộc tính `AutoGenerateColumns`, mặc định có giá trị `true`. Nếu bật `AutoGenerateColumns`, thì cả hai trường sẽ được hiển thị trong `BoundFields`

Hoặc viết mã trong trang ASPX:

`<Columns>`

`<asp:BoundField DataField="EmpID" HeaderText="Employee ID" />`

`<asp:BoundField DataField="DepID" HeaderText="Department ID" />`

`<asp:BoundField DataField="Name" HeaderText="Employee Name" />`

`</Columns>`

Sử dụng CommandFields

CommandField được sử dụng để thực hiện các tác vụ Edit, Delete, Update, Cancel, và Select một cách đơn giản. Ta không cần bật các thuộc tính `AutoGenerateEditButton` hoặc `AutoGenerateDeleteButton` trong GridView khi sử dụng `CommandField`.

`CommandField` hỗ trợ thuộc tính `ButtonType` để xác định kiểu của nút hiển thị bởi `CommandField`. Ta có thể thiết lập là `Button`, `Image`, hoặc `Link`. Chế độ mặc định được thiết lập là `Link`.

Nút **Edit** sử dụng để cập nhật dữ liệu từ nguồn, nếu ta click nút Edit nó sẽ tự động được thay thế bởi nút **Update** và nút **Cancel**. Nút **Delete** dùng để xóa bản ghi từ nguồn.

```

<Columns>
    <asp:CommandField ShowEditButton="True" />
    <asp:CommandField ShowDeleteButton="True" />
</Columns>

```

CommandFields.cs

```

using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Web.UI.WebControls;

public partial class GridViewDemo : System.Web.UI.Page
{
    SqlConnection conn;
    SqlDataAdapter adapter;
    DataSet ds;
    SqlCommand cmd;
    string cs;
    protected void Page_Load(object sender, EventArgs e)
    {
        if(!IsPostBack)
        {
            FillGridView();
        }
    }
    protected void FillGridView()
    {
        cs =
ConfigurationManager.ConnectionStrings["conString"].ConnectionString;
        try
        {
            conn = new SqlConnection(cs);
            adapter = new SqlDataAdapter("select * from tblEmps", conn);
            ds = new DataSet();
            adapter.Fill(ds);
            GridView1.DataSource = ds;
            GridView1.DataBind();
        }
        catch (Exception ex)
        {
            Label1.Text = "ERROR :: " + ex.Message;
        }
    }
}

```



```

        finally
        {
            ds.Dispose();
            conn.Dispose();
        }
    }

protected void GridView1_RowEditing(object sender,
    GridViewEditEventArgs e)
{
    GridView1.EditIndex = e.NewEditIndex;
    FillGridView();
}

protected void GridView1_RowUpdating(object sender,
    GridViewUpdateEventArgs e)
{
    int userid = Convert.ToInt32(GridView1.DataKeys[e.RowIndex].Value.
        ToString());
    GridViewRow row = (GridViewRow)GridView1.Rows[e.RowIndex];
    TextBox txtName = (TextBox)row.Cells[2].Controls[0];
    TextBox txtGender = (TextBox)row.Cells[3].Controls[0];
    TextBox txtSalary = (TextBox)row.Cells[4].Controls[0];
    TextBox txtAddress = (TextBox)row.Cells[5].Controls[0];
    TextBox txtDepartmentID = (TextBox)row.Cells[6].Controls[0];

    string updateQuery = "update tblEmps set name=" + txtName.Text
        + ", gender=" + txtGender.Text + ",Salary=" + txtSalary.Text
        + ",Address=" + txtAddress.Text + ",DepID="
        + txtDepartmentID.Text + " where EmpID=" + userid;
    cs = ConfigurationManager.ConnectionStrings
        ["conString"].ConnectionString;
    conn = new SqlConnection(cs);
    cmd = new SqlCommand(updateQuery,conn);
    conn.Open();
    cmd.ExecuteNonQuery();
    conn.Close();
    GridView1.EditIndex = -1;
    FillGridView();
}

protected void GridView1_RowCancelingEdit(object sender,

```

```

GridViewCancelEventArgs e)
{
    GridView1.EditIndex = -1;
    FillGridView();
}
protected void GridView1_RowDeleting(object sender,
GridViewDeleteEventArgs e)
{
    int userid = Convert.ToInt32(GridView1.DataKeys[e.RowIndex].Value.
        ToString());
    string deleteQuery = "delete from tblEmps where empID="+userid;
    cs = ConfigurationManager.ConnectionStrings
        ["conString"].ConnectionString;
    conn = new SqlConnection(cs);
    cmd = new SqlCommand(deleteQuery, conn);
    conn.Open();
    cmd.ExecuteNonQuery();
    conn.Close();
    GridView1.EditIndex = -1;
    FillGridView();
}
}

```

Kết quả:

		EmpID	Name	Gender	Salary	Address	DEPID
Edit	Delete	1	Raj	Male	25000	Pune	1
Edit	Delete	2	Ramesh Gupta	Male	20000	Nagpur	1
Edit	Delete	3	David	Male	20000	USA	2
Edit	Delete	4	Zeny	Female	42000	Canada	2
Edit	Delete	5	Nishant	Male	50000	Pune	3
Edit	Delete	6	Digvijay	Male	45000	New York	3

Click Edit, trong trường hợp này EmpID được sử dụng là DataKeys, nên được thiết lập non-editable. Ta có thể thiết lập thuộc tính DataKeys của GridView.

		EmpID	Name	Gender	Salary	Address	DEPID
Edit	Delete	1	Raj	Male	25000	Pune	1
Edit	Delete	2	Ramesh Gupta	Male	20000	Nagpur	1
Update Cancel		3	David	Male	20000	USA	2
Edit	Delete	4	Zeny	Female	42000	Canada	2
Edit	Delete	5	Nishant	Male	50000	Pune	3
Edit	Delete	6	Digvijay	Male	45000	New York	3

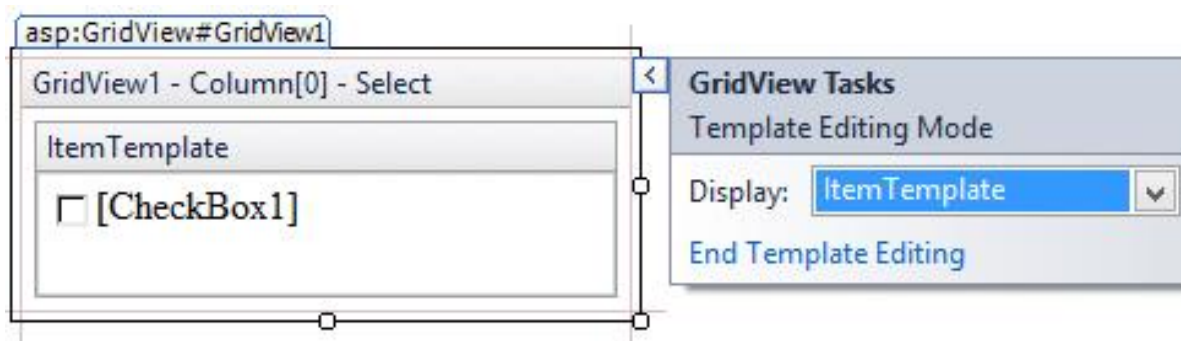
3.2.3. Sử dụng TemplateField in GridView

Sử dụng TemplateField, ta có thể đặt các thẻ HTML, biểu thức liên kết dữ liệu động DataBinding, hoặc điều khiển ASP.NET trong GridView. Ta có thể thay đổi giao diện và thêm xác thực logic cho các trường.

Ví dụ user có thể đánh dấu các hộp chọn để lựa chọn một hoặc nhiều bản ghi để xóa. Để thực hiện ta thêm các TemplateField vào GridView.

Sử dụng smart tag của điều khiển GridView

Click Edit Column, chọn TemplateField và click nút Add. Tiếp theo click liên kết Edit Templates thêm checkbox vào ItemTemplate sau đó chọn End Template Editing.



Hoặc viết mã trong trang ASPX

```
<Columns>
    <asp:TemplateField HeaderText="Select">
        <ItemTemplate>
            <asp:CheckBox ID="CheckBox1" runat="server" /> <br />
        </ItemTemplate>
    </asp:TemplateField>
</Columns>
```

TemplateField trong GridView hỗ trợ 6 loại templates:

- **AlternatingItemTemplate:** Hiển thị định dạng nội dung dòng chẵn lẻ.
- **EditItemTemplate:** Sử dụng khi user edit bản ghi.
- **FooterTemplate:** Hiển thị nhãn column footer.
- **HeaderTemplate:** Hiển thị nội dung là nhãn column header.
- **InsertItemTemplate:** Khi thêm bản ghi mới,
- **ItemTemplate:** Dùng cho tất cả các bản ghi trong GridView.

```
protected void btnDelete_Click(object sender, EventArgs e)
{
    foreach (GridViewRow gridrow in GridView1.Rows)
    {
        CheckBox chkDelete = (CheckBox)gridrow.
            FindControl("CheckBox1");
        if (chkDelete.Checked)
        {
            int empid = Convert.ToInt32(gridrow.Cells[1].Text);
            DeleteRecord(empid);
        }
    }
    FillGridView();
}
```

```
protected void DeleteRecord(int ID)
{
    string deleteQuery = "delete from tblEmps where EmpID="+ID;
    cs = ConfigurationManager.ConnectionStrings
        ["conString"].ConnectionString;
    conn = new SqlConnection(cs);
    cmd = new SqlCommand(deleteQuery,conn);
    conn.Open();
    cmd.ExecuteNonQuery();
    conn.Close();
}
```

Kết quả:

Select	EmpID	Name	Gender	Salary	Address	DEPID
<input type="checkbox"/>	1	Raj	Male	25000	Pune	1
<input type="checkbox"/>	2	Ramesh Gupta	Male	20000	Nagpur	1
<input type="checkbox"/>	3	David	Male	20000	USA	2
<input type="checkbox"/>	4	Zeny	Female	42000	Canada	3
<input type="checkbox"/>	5	Nishant	Male	50000	Pune	3
<input type="checkbox"/>	6	Digvijay	Male	45000	New York	3

Delete

3.2.4. Sử dụng sự kiện RowDataBound

Sự kiện RowDataBound event phát sinh khi data row được nạp cho GridView. RowDataBound được sử dụng khi ta muốn thay đổi hiển thị của một tập hợp các bản ghi trong GridView.

Ví dụ ta muốn highlight với màu Light Green, cho các bản ghi , "Nhân viên có lương từ 50000 trở lên.

```
Double salaryTotal = 0;
protected void GridView1_RowDataBound(object sender,
    GridViewRowEventArgs e)
{
    if (e.Row.RowType == DataControlRowType.DataRow)
    {
        Double sal = (Double)DataBinder.Eval(e.Row.DataItem, "Salary");
        salaryTotal += sal;
        if (sal >= 50000)
            e.Row.BackColor = System.Drawing.Color.LightGreen;
    }
}
```

EmpID	Name	Gender	Salary	Address	DEPID
1	Raj	Male	25000	Pune	1
2	Ramesh Gupta	Male	20000	Nagpur	1
3	David	Male	20000	USA	2
4	Zeny	Female	42000	Canada	3
5	Nishant	Male	50000	Pune	3
6	Digvijay	Male	45000	New York	3

The second parameter of RowDataBound event provides lots of important properties.

```
protected void GridView1_RowDataBound(object sender,
GridViewRowEventArgs e)
{
}
```

Các thuộc tính của GridViewRowEventArgs:

- **Cells:** Tập hợp các ô trong dòng của bảng được nạp dữ liệu.
- **DataItem:** Data item sẽ được nạp dữ liệu cho dòng của bảng.
- **RowIndex:** Chỉ số index của dòng được nạp dữ liệu.
- **RowState:** Trạng thái của dòng. Alternate, Normal, Selected, và Edit.
- **RowType:** Kiểu của dòng trong bảng sẽ được nạp dữ liệu. Gồm các giá trị DataRow, Footer, Header, NullRow, Pager, and Separator.