Wedding Planner

# Baza danych

## Moduł główny

Moduł główny zawiera trzy tabele:

#### Article

CREATE TABLE [dbo].[Article] (

[ArticleId] INT IDENTITY (1, 1) NOT NULL,

[ArticleName] NVARCHAR (50) NOT NULL,

[BarCode] NVARCHAR (50) NULL,

[ArticlesNumber] INT NULL,

[ShopName] NVARCHAR (50) NULL,

[ShopAdress] NVARCHAR (50) NULL,

[Website] NVARCHAR (50) NULL,

PRIMARY KEY CLUSTERED ([ArticleId] ASC)

);

#### Guest

CREATE TABLE [dbo].[Guest] (

[GuestId] INT IDENTITY (1, 1) NOT NULL,

[WeddingId\_FK1] INT NOT NULL,

[Adults] INT NULL,

[Teens] INT NULL,

[Kids] INT NULL,

PRIMARY KEY CLUSTERED ([GuestId] ASC, [WeddingId\_FK1] ASC)

);

#### Wedding

CREATE TABLE [dbo].[Wedding] (

[WeddingId] INT IDENTITY (1, 1) NOT NULL,

[Bride] NVARCHAR (50) NOT NULL,

[Groom] NVARCHAR (50) NOT NULL,

[Place] NVARCHAR (50) NULL,

[Date] DATE NULL,

PRIMARY KEY CLUSTERED ([WeddingId] ASC)

);

Każda z nich zawiera przynajmniej jeden rekord, aby odpowiednia tabela pojawiała się w widoku projektu.

Poniżej pokazano plik numer.aspx.cs, który służy do wyszukiwania, a następnie wyświetlania zawartości na stronie:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

namespace WeddingsPlanner

{

public partial class number : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=D:\Pulpit\PAI\WeddingsPlanner\WeddingsPlanner\App\_Data\MyDatabase.mdf;Integrated Security=True");

protected void Page\_Load(object sender, EventArgs e)

{

con.Open();

con.Close();

}

}

}

## Moduł lokalny

W module lokalnym są tylko dwie tabele:

#### Guest

CREATE TABLE [dbo].[Guest] (

[GuestId] INT IDENTITY (1, 1) NOT NULL,

[FamilyId] INT NOT NULL,

[Name1] NVARCHAR (50) NOT NULL,

[Name2] NVARCHAR (50) NULL,

[Mail] NVARCHAR (50) NULL,

[Password] NVARCHAR (50) NULL,

[NumberOfKids] INT NULL,

[NumberOfTeens] INT NULL,

PRIMARY KEY CLUSTERED ([GuestId] ASC)

);

#### Present

CREATE TABLE [dbo].[Present] (

[ArticleId] INT IDENTITY (1, 1) NOT NULL,

[ArticleName] NVARCHAR (50) NOT NULL,

[BarCode] NVARCHAR (50) NULL,

[ArticlesNumber] INT NULL,

[ShopName] NVARCHAR (50) NULL,

[ShopAdress] NVARCHAR (50) NULL,

[Website] NVARCHAR (50) NULL,

[Occupied] INT DEFAULT ((0)) NOT NULL,

PRIMARY KEY CLUSTERED ([ArticleId] ASC)

);

Przykład kodu z pliku presents.aspx.cs, który służy do wyświetlania, edytowania, wstawiania, usuwania elementów z bazy danych oraz do umożliwia paginację:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

using System.Data;

using System.Configuration;

using System.Web.Security;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

using System.Collections;

using System.IO.Compression;

namespace YourWedding

{

public partial class presents : System.Web.UI.Page

{

private String strConnString = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;

protected void Page\_Load(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(strConnString);

con.Open();

if (!IsPostBack)

{

BindData();

}

con.Close();

}

private void BindData()

{

string strQuery = "select ArticleId,ArticleName,BarCode,ArticlesNumber,ShopName,ShopAdress,Website,Occupied" +

" from Present";

SqlCommand cmd = new SqlCommand(strQuery);

GridView1.DataSource = GetData(cmd);

GridView1.DataBind();

}

private DataTable GetData(SqlCommand cmd)

{

DataTable dt = new DataTable();

SqlConnection con = new SqlConnection(strConnString);

SqlDataAdapter sda = new SqlDataAdapter();

cmd.CommandType = CommandType.Text;

cmd.Connection = con;

con.Open();

sda.SelectCommand = cmd;

sda.Fill(dt);

return dt;

}

protected void OnPaging(object sender, GridViewPageEventArgs e)

{

BindData();

GridView1.PageIndex = e.NewPageIndex;

GridView1.DataBind();

}

protected void AddNewCustomer(object sender, EventArgs e)

{

string ArticleId = ((TextBox)GridView1.FooterRow.FindControl("txtArticleId")).Text;

string ArticleName = ((TextBox)GridView1.FooterRow.FindControl("txtArticleName")).Text;

string ShopName = ((TextBox)GridView1.FooterRow.FindControl("txtShopName")).Text;

string BarCode = ((TextBox)GridView1.FooterRow.FindControl("txtBarCode")).Text;

string ArticlesNumber = ((TextBox)GridView1.FooterRow.FindControl("txtArticlesNumber")).Text;

string ShopAdress = ((TextBox)GridView1.FooterRow.FindControl("txtShopAdress")).Text;

string Website = ((TextBox)GridView1.FooterRow.FindControl("txtWebsite")).Text;

string Occupied = ((TextBox)GridView1.FooterRow.FindControl("txtOccupied")).Text;

SqlConnection con = new SqlConnection(strConnString);

SqlCommand cmd = new SqlCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = "SET IDENTITY\_INSERT Present ON;" +

"insert into Present(ArticleId,ArticleName,ShopName,BarCode,ArticlesNumber,ShopAdress,Website,Occupied) " +

"values(@ArticleId,@ArticleName,@ShopName,@BarCode,@ArticlesNumber,@ShopAdress,@Website,@Occupied);" +

"SET IDENTITY\_INSERT Present OFF;" +

"select ArticleId,ArticleName,ShopName,BarCode,ArticlesNumber,ShopAdress,Website,Occupied from Present";

cmd.Parameters.Add("@ArticleId", SqlDbType.VarChar).Value = ArticleId;

cmd.Parameters.Add("@ArticleName", SqlDbType.VarChar).Value = ArticleName;

cmd.Parameters.Add("@ShopName", SqlDbType.VarChar).Value = ShopName;

cmd.Parameters.Add("@BarCode", SqlDbType.VarChar).Value = BarCode;

cmd.Parameters.Add("@ArticlesNumber", SqlDbType.VarChar).Value = ArticlesNumber;

cmd.Parameters.Add("@ShopAdress", SqlDbType.VarChar).Value = ShopAdress;

cmd.Parameters.Add("@Website", SqlDbType.VarChar).Value = Website;

cmd.Parameters.Add("@Occupied", SqlDbType.VarChar).Value = Occupied;

GridView1.DataSource = GetData(cmd);

GridView1.DataBind();

}

protected void DeleteCustomer(object sender, EventArgs e)

{

LinkButton lnkRemove = (LinkButton)sender;

SqlConnection con = new SqlConnection(strConnString);

SqlCommand cmd = new SqlCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = "SET IDENTITY\_INSERT Present ON;" +

"delete from Present where " +

"ArticleId=@ArticleId;" +

"SET IDENTITY\_INSERT Present OFF;" +

"select ArticleId,ArticleName,ShopName,BarCode,ArticlesNumber,ShopAdress,Website,Occupied from Present";

cmd.Parameters.Add("@ArticleId", SqlDbType.VarChar).Value = lnkRemove.CommandArgument;

GridView1.DataSource = GetData(cmd);

GridView1.DataBind();

}

protected void EditCustomer(object sender, GridViewEditEventArgs e)

{

GridView1.EditIndex = e.NewEditIndex;

BindData();

}

protected void CancelEdit(object sender, GridViewCancelEditEventArgs e)

{

GridView1.EditIndex = -1;

BindData();

}

protected void UpdateCustomer(object sender, GridViewUpdateEventArgs e)

{

string ArticleId = ((Label)GridView1.Rows[e.RowIndex].FindControl("lblArticleId")).Text;

string ArticleName = ((TextBox)GridView1.Rows[e.RowIndex].FindControl("txtArticleName")).Text;

string ShopName = ((TextBox)GridView1.Rows[e.RowIndex].FindControl("txtShopName")).Text;

string BarCode = ((TextBox)GridView1.Rows[e.RowIndex].FindControl("txtBarCode")).Text;

string ArticlesNumber = ((TextBox)GridView1.Rows[e.RowIndex].FindControl("txtArticlesNumber")).Text;

string ShopAdress = ((TextBox)GridView1.Rows[e.RowIndex].FindControl("txtShopAdress")).Text;

string Website = ((TextBox)GridView1.Rows[e.RowIndex].FindControl("txtWebsite")).Text;

string Occupied = ((TextBox)GridView1.Rows[e.RowIndex].FindControl("txtOccupied")).Text;

SqlConnection con = new SqlConnection(strConnString);

SqlCommand cmd = new SqlCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = "SET IDENTITY\_INSERT Present ON;" +

"delete from Present where ArticleId=@ArticleId;" +

"insert into Present(ArticleId,ArticleName,ShopName,BarCode,ArticlesNumber,ShopAdress,Website,Occupied) " +

"values(@ArticleId,@ArticleName,@ShopName,@BarCode,@ArticlesNumber,@ShopAdress,@Website,@Occupied);" +

"SET IDENTITY\_INSERT Present OFF;" +

"select ArticleId,ArticleName,BarCode,ArticlesNumber,ShopName,ShopAdress,Website,Occupied from Present";

cmd.Parameters.Add("@ArticleId", SqlDbType.VarChar).Value = ArticleId;

cmd.Parameters.Add("@ArticleName", SqlDbType.VarChar).Value = ArticleName;

cmd.Parameters.Add("@ShopName", SqlDbType.VarChar).Value = ShopName;

cmd.Parameters.Add("@BarCode", SqlDbType.VarChar).Value = BarCode;

cmd.Parameters.Add("@ArticlesNumber", SqlDbType.VarChar).Value = ArticlesNumber;

cmd.Parameters.Add("@ShopAdress", SqlDbType.VarChar).Value = ShopAdress;

cmd.Parameters.Add("@Website", SqlDbType.VarChar).Value = Website;

cmd.Parameters.Add("@Occupied", SqlDbType.VarChar).Value = Occupied;

GridView1.EditIndex = -1;

GridView1.DataSource = GetData(cmd);

GridView1.DataBind();

}

}

}

## Materiały źródłowe

* <https://www.aspsnippets.com/Articles/GridView---Add-Edit-Update-Delete-and-Paging-the-AJAX-way.aspx>
* <https://msdn.microsoft.com/en-us/library/tw738475.aspx>
* <https://www.youtube.com/watch?v=AuVpmu9CKRY>

Poniższe linki odnoszą się do stron serwisu Stack Overflow, zawierających rozwiązania problemów, które pojawiły się podczas łączenia widoków z bazą danych:

* <https://stackoverflow.com/questions/19155775/how-to-update-identity-column-in-sql-server>
* <https://stackoverflow.com/questions/11574964/master-page-only-one-instance-of-a-scriptmanager-can-be-added-to-the-page>
* <https://stackoverflow.com/questions/41257016/the-name-gridview1-does-not-exist-in-the-current-context>
* <https://stackoverflow.com/questions/5817261/select-statement-to-select-records-from-two-tables-based-on-id-entered-in-textbo?rq=1>