Wedding Planner

# Połączenie między serwerami

## Struktura danych

Aby ułatwić połączenie między serwerami postarano się, aby obie bazy danych przechowywały dane o podobnych typach:

|  |  |  |
| --- | --- | --- |
| **GŁÓWNA** | **LOKALNA** |  |
| ArticleId | ArticleId |  |
| ArticleName | ArticleName |  |
| BarCode | BarCode |  |
| ArticlesNumber | ArticlesNumber |  |
| ShopName | ShopName |  |
| ShopAdress | ShopAdress |  |
| Website | Website |  |
|  | Occupied |  |
|  |  |  |
|  |  |  |
| GuestId | GuestId |  |
| WeddingId\_FK1 |  |  |
| Adults | Name1 | Name2 |
| Teens | NumberOfTeens |  |
| Kids | NumberOfKids |  |
|  | Password |  |
|  | FamilyId |  |
|  | Mail |  |
|  |  |  |
|  |  |  |
| WeddingId |  |  |
| Bride |  |  |
| Groom |  |  |
| Place |  |  |
| Date |  |  |

## Sposób połączenia

Do projektu WeddingsPlanner została podłączona baza danych pobrana z projektu YourWedding przy pomocy programu Microsoft SQL Server.

Poniżej pokazano fragment kodu z pliku numer.aspx.cs[[1]](#footnote-1), który umożliwia przekształcenie zawartości jednej tabeli (z oddziału) do drugiej (z centrali):

private int UpdateGuestList(DataRow dr, DataTable dt2, SqlCommand cmd)

{

int nAdults = 0;

int nTeens = 0;

int nKids = 0;

foreach (DataRow dr2 in dt2.Rows)

{

if (dr2.Field<int?>("NumberOfKids") != null)

nKids += dr2.Field<int>("NumberOfKids");

if (dr2.Field<int?>("NumberOfTeens") != null)

nTeens += dr2.Field<int>("NumberOfTeens");

if (dr2.Field<string>("Name1") != "")

nAdults += 1;

if (dr2.Field<string>("Name2") != "")

nAdults += 1;

}

cmd.CommandText = "delete from Guest where WeddingId\_FK1=@WeddingId\_FK1;" +

"insert into Guest(WeddingId\_FK1,Adults,Teens,Kids) " +

"values(@WeddingId\_FK1,@Adults,@Teens,@Kids)";

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

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

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

cmd.Parameters.Add("@WeddingId\_FK1", SqlDbType.VarChar).Value = dr.Field<int>("WeddingId");

GetData(strConnStringW, cmd);

return 0;

}

Poniżej pokazano fragment pliku presents.aspx.cs[[2]](#footnote-2), który służy do synchronizacji zawartości tabel Article oraz Present pomiędzy centralą a oddziałem:

protected void UpdateDatabase(object sender, EventArgs e)

{

int allAdded = -1;

while (allAdded != 0)

{

SqlConnection con2 = new SqlConnection(strConnStringY);

con2.Open();

SqlCommand cmd2 = new SqlCommand("Select ArticleId,ArticleName,BarCode,ArticlesNumber,ShopName,ShopAdress,Website,Occupied FROM Present");

DataTable dt2 = GetData(strConnStringY, cmd2);

con2.Close();

SqlConnection con = new SqlConnection(strConnStringW);

con.Open();

SqlCommand cmd = new SqlCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = "Select ArticleId,ArticleName,BarCode,ArticlesNumber,ShopName,ShopAdress,Website FROM Article";

DataTable dt = GetData(strConnStringW, cmd);

allAdded = CheckIfPresent(dt, dt2, cmd);

con.Close();

}

Page\_Load(sender, e);

}

private int CheckIfPresent(DataTable dt, DataTable dt2, SqlCommand cmd)

{

foreach (DataRow dr2 in dt2.Rows)

{

int i = 0;

foreach (DataRow dr in dt.Rows)

{

if (dr.Field<string>("BarCode") == dr2.Field<string>("BarCode"))

i = 1;

}

if (i == 0)

{

cmd.CommandText = "insert into Article(ArticleName,ShopName,BarCode,ArticlesNumber,ShopAdress,Website) " +

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

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

cmd.Parameters.Add("@ArticleName", SqlDbType.VarChar).Value = dr2.Field<string>("ArticleName");

cmd.Parameters.Add("@ShopName", SqlDbType.VarChar).Value = dr2.Field<string>("ShopName");

cmd.Parameters.Add("@BarCode", SqlDbType.VarChar).Value = dr2.Field<string>("BarCode");

cmd.Parameters.Add("@ArticlesNumber", SqlDbType.VarChar).Value = dr2.Field<int>("ArticlesNumber");

cmd.Parameters.Add("@ShopAdress", SqlDbType.VarChar).Value = dr2.Field<string>("ShopAdress");

cmd.Parameters.Add("@Website", SqlDbType.VarChar).Value = dr2.Field<string>("Website");

GetData(strConnStringW, cmd);

return -1;

}

}

return 0;

}

private DataTable GetData(String strConnString, 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;

}

## Materiały źródłowe

* <https://www.youtube.com/watch?v=SrsCz2q2zLo>
* <https://www.codeproject.com/Articles/490448/Transfer-data-or-script-between-two-SQL-Server-dat>

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/15777745/how-does-comparison-operator-works-with-null-int>
* <https://stackoverflow.com/questions/9223911/execute-javascript-function-when-clicking-on-asp-net-button>
* <https://stackoverflow.com/questions/18961938/populate-data-table-from-data-reader>
* <https://stackoverflow.com/questions/13870843/turning-a-sqlcommand-with-parameters-into-a-datatable>
* <https://stackoverflow.com/questions/13677318/how-to-run-multiple-sql-commands-in-a-single-sql-connection>

1. <https://github.com/Hryniewska/PAI/blob/master/WeddingsPlanner/WeddingsPlanner/number.aspx.cs> [↑](#footnote-ref-1)
2. <https://github.com/Hryniewska/PAI/blob/master/WeddingsPlanner/WeddingsPlanner/presents.aspx.cs> [↑](#footnote-ref-2)