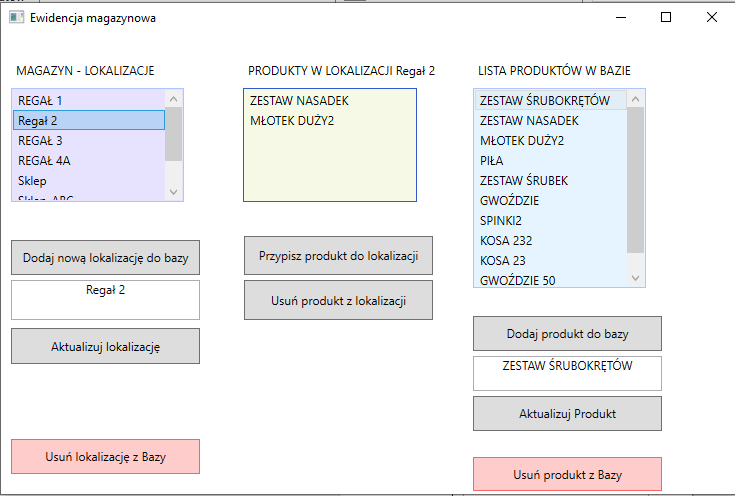
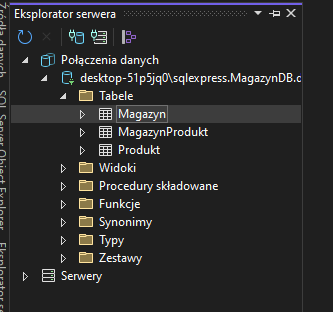
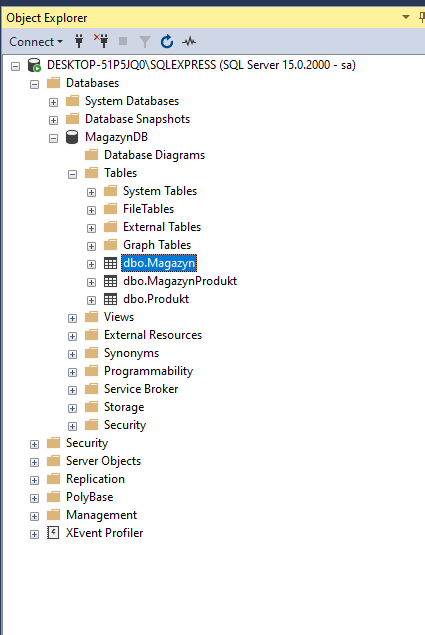
PROGRAM CRUD: DODAWANIE PRODUKTÓW NA STAN DO OKREŚLONEJ LOKALIZACJI, FUNKCJE:



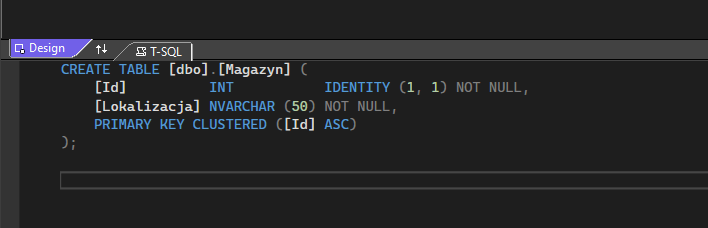
APLIKACJA ZAWIERA BAZĘ DANYCH – ZAINSTALUJ SQL EXPRESS



WIDOK Z SQLEXPRESS:



WŁAŚCIWOŚCI BAZY:



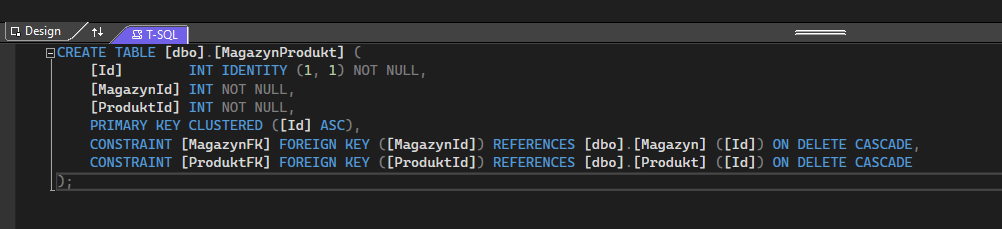
CREATE TABLE [dbo].[Magazyn] (

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

[Lokalizacja] NVARCHAR (50) NOT NULL,

PRIMARY KEY CLUSTERED ([Id] ASC)

);



CREATE TABLE [dbo].[MagazynProdukt] (

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

[MagazynId] INT NOT NULL,

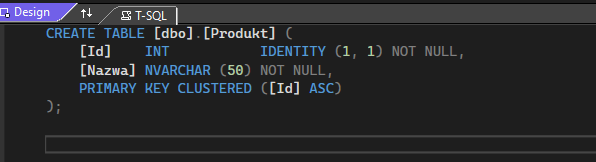
[ProduktId] INT NOT NULL,

PRIMARY KEY CLUSTERED ([Id] ASC),

CONSTRAINT [MagazynFK] FOREIGN KEY ([MagazynId]) REFERENCES [dbo].[Magazyn] ([Id]) ON DELETE CASCADE,

CONSTRAINT [ProduktFK] FOREIGN KEY ([ProduktId]) REFERENCES [dbo].[Produkt] ([Id]) ON DELETE CASCADE

);



CREATE TABLE [dbo].[Produkt] (

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

[Nazwa] NVARCHAR (50) NOT NULL,

PRIMARY KEY CLUSTERED ([Id] ASC)

);

OKNO XAML: MainWindow.xaml

<Window x:Class="APLIKACJA\_SQL.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:APLIKACJA\_SQL"

mc:Ignorable="d"

Title="Ewidencja magazynowa" Height="500" Width="750">

<Grid>

<Label Content="MAGAZYN - LOKALIZACJE" HorizontalAlignment="Left" Margin="10,24,0,0" VerticalAlignment="Top"/>

<ListBox Name="listaLokalizacji" Height="NaN" Margin="10,55,551,292" Width="NaN" SelectionChanged="listaLokalizacji\_SelectionChanged" Background="#FFE7E3FF" BorderBrush="#FFB0C4FF"/>

<Label Name="LabelLokalizacja" Content="WYBIERZ LOKALIZACJĘ" HorizontalAlignment="Left" Margin="242,24,0,0" VerticalAlignment="Top"/>

<ListBox Name="listaProduktLokalizacji" Height="NaN" Margin="242,55,318,292" Width="NaN" BorderBrush="#FF3359CA" Background="#FFF6F9E6"/>

<ListBox Name="listaAllProdukt" Height="NaN" Margin="472,55,89,206" Width="NaN" Background="#FFE5F4FF" BorderBrush="#FFB0C4FF" SelectionChanged="listaAllProdukt\_SelectionChanged"/>

<Button Content="Usuń lokalizację z Bazy" HorizontalAlignment="Left" Margin="10,406,0,0" VerticalAlignment="Top" Height="35" Width="189" Click="Button\_Click\_UsunLokalizacje" BorderBrush="#FFF96262" Background="#FFFFCCCC"/>

<Button Content="Usuń produkt z lokalizacji" HorizontalAlignment="Left" Margin="243,247,0,0" VerticalAlignment="Top" Height="40" Width="189" Click="Button\_Click\_UsunProduktzLokalizacji" />

<Button Content="Dodaj nową lokalizację do bazy" HorizontalAlignment="Left" Margin="10,207,0,0" VerticalAlignment="Top" Height="35" Width="189" Click="Button\_Click\_DodajLokalizacjedoBazy"/>

<Button Content="Dodaj produkt do bazy" HorizontalAlignment="Left" Margin="472,283,0,0" VerticalAlignment="Top" Height="35" Width="189" Click="Button\_Click\_DodajProduktdoBazy"/>

<Button Content="Przypisz produkt do lokalizacji" HorizontalAlignment="Left" Margin="243,203,0,0" VerticalAlignment="Top" Height="39" Width="189" Click="Button\_Click\_DodajProduktdoLokalizacji"/>

<Button Content="Usuń produkt z Bazy" HorizontalAlignment="Left" Margin="472,424,0,0" VerticalAlignment="Top" Height="34" Width="189" Click="Button\_Click\_UsunProduktzBazy" Background="#FFFFCCCC" BorderBrush="#FFF96666"/>

<Button Content="Aktualizuj lokalizację" HorizontalAlignment="Left" Margin="10,295,0,0" VerticalAlignment="Top" Height="36" Width="189" Click="Button\_Click\_AktualizacjaLokalizacji"/>

<Button Content="Aktualizuj Produkt" HorizontalAlignment="Left" Margin="472,363,0,0" VerticalAlignment="Top" Height="35" Width="189" Click="Button\_Click"/>

<TextBox Name="textBox1" HorizontalAlignment="Left" Margin="10,247,0,0" Text="ZMIEŃ LUB DODAJ LOKALIZACJĘ" VerticalAlignment="Top" Width="189" Height="40" TextAlignment="Center" TextWrapping="Wrap"/>

<TextBox Name="textBox2" HorizontalAlignment="Left" Margin="472,323,0,0" TextWrapping="Wrap" Text="ZMIEŃ LUB DODAJ PRODUKT" VerticalAlignment="Top" Width="189" Height="35" TextAlignment="Center"/>

<Label Content="LISTA PRODUKTÓW W BAZIE" HorizontalAlignment="Left" Margin="472,24,0,0" VerticalAlignment="Top"/>

</Grid>

</Window>

OKNO: MainWindow.xaml.cs

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace APLIKACJA\_SQL

{

/// <summary>

/// Logika interakcji dla klasy MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

SqlConnection sqlConnection;

public MainWindow()

{

InitializeComponent();

//Połączenie do bazy danych SQL o nazwie : "MagazynDBConnectionString"

string connectionString = ConfigurationManager.ConnectionStrings["WPF\_MAGAZYN.Properties.Settings.MagazynDBConnectionString"].ConnectionString;

sqlConnection = new SqlConnection(connectionString);

ShowAllProdukt();

ShowMagazyn();

}

//### OKNO TRZECIE

private void ShowAllProdukt()

{

try

{

string query = "select \* from Produkt";

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(query, sqlConnection);

using (sqlDataAdapter)

{

DataTable produktTable = new DataTable();

sqlDataAdapter.Fill(produktTable);

listaAllProdukt.DisplayMemberPath = "Nazwa";

listaAllProdukt.SelectedValuePath = "Id";

listaAllProdukt.ItemsSource = produktTable.DefaultView;

}

}

catch (Exception e)

{

// MessageBox.Show(e.ToString());

}

}

//Tworzenie zapytania SQL

private void ShowMagazyn()

{

try

{

string query = "select \* from Magazyn";

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(query, sqlConnection);

using (sqlDataAdapter)

{

//Tworzenie Tabeli

DataTable magazynTable = new DataTable();

//Fill(DataSet) - Dodaje lub odświeża wiersze w obiekcie DataSet , aby dopasować je do tych w źródle danych.

sqlDataAdapter.Fill(magazynTable);

//dodatnie do ListBox "listaLokalizacji" informacji z tabeli, które mają być wyświetlone - w tym przypadku :Lokalizacja

listaLokalizacji.DisplayMemberPath = "Lokalizacja";

//dodatnie do ListBox "listaLokalizacji" przekazania zaznaczonej wartości - w tym przypadku: Id

listaLokalizacji.SelectedValuePath = "Id";

//dodatnie do ListBox "listaLokalizacji" domyslnego widoku magazynTable (query = "select \* from Magazyn")

listaLokalizacji.ItemsSource = magazynTable.DefaultView;

}

}

catch (Exception e)

{

MessageBox.Show(e.ToString());

}

}

//############# OKNO DRUGIE - POJAWIANIE SIĘ PRODUKTÓW W "listaProduktLokalizacji" PO WYBRANIU MAGAZYNIU

private void ShowProduktLokalizacji()

{

try

{

// utworzenie zaptania i połączenie do bazy

string query = "select \* from Produkt p inner join MagazynProdukt mp on p.Id = mp.ProduktId where mp.MagazynId = @MagazynId";

SqlCommand sqlCommand = new SqlCommand(query, sqlConnection);

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand);

using (sqlDataAdapter)

{

sqlCommand.Parameters.AddWithValue("@MagazynId", listaLokalizacji.SelectedValue);

//Tworzenie Tabeli

DataTable produktTable = new DataTable();

//Fill(DataSet) - Dodaje lub odświeża wiersze w obiekcie DataSet , aby dopasować je do tych w źródle danych.

sqlDataAdapter.Fill(produktTable);

//dodatnie do ListBox "listaLokalizacji" informacji z tabeli, które mają być wyświetlone - w tym przypadku :Lokalizacja

listaProduktLokalizacji.DisplayMemberPath = "Nazwa";

//dodatnie do ListBox "listaLokalizacji" przekazania zaznaczonej wartości - w tym przypadku: Id

listaProduktLokalizacji.SelectedValuePath = "Id";

//dodatnie do ListBox "listaLokalizacji" domyslnego widoku magazynTable (query = "select \* from Magazyn")

listaProduktLokalizacji.ItemsSource = produktTable.DefaultView;

}

}

catch (Exception e)

{

// MessageBox.Show(e.ToString());

}

}

private void listaLokalizacji\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

// Wyświetla produkt

ShowProduktLokalizacji();

// AKTUALIZACJA TEXTU W TEXT\_BOXIE

ShowSelectedinTextBox1();

}

// PRZYCISK

private void Button\_Click\_UsunLokalizacje(object sender, RoutedEventArgs e)

{

try

{

string query = "delete from Magazyn where id = @MagazynId";

SqlCommand sqlCommand = new SqlCommand(query, sqlConnection);

sqlConnection.Open();

sqlCommand.Parameters.AddWithValue("@MagazynId", listaLokalizacji.SelectedValue);

sqlCommand.ExecuteScalar();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

finally

{

sqlConnection.Close();

ShowMagazyn();

}

}

// PRZYCISK DODAJ LOKALIZACJĘ

private void Button\_Click\_DodajLokalizacjedoBazy(object sender, RoutedEventArgs e)

{

try

{

string query = "insert into Magazyn values (@Lokalizacja)";

SqlCommand sqlCommand = new SqlCommand(query, sqlConnection);

sqlConnection.Open();

sqlCommand.Parameters.AddWithValue("@Lokalizacja", textBox1.Text);

sqlCommand.ExecuteScalar();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

finally

{

sqlConnection.Close();

ShowMagazyn();

}

}

// PRZYCISK DODAJ PRODUKT DO LOKALIZACJI

private void Button\_Click\_DodajProduktdoLokalizacji(object sender, RoutedEventArgs e)

{

try

{

string query = "insert into MagazynProdukt values (@MagazynId, @ProduktId)";

SqlCommand sqlCommand = new SqlCommand(query, sqlConnection);

sqlConnection.Open();

sqlCommand.Parameters.AddWithValue("@MagazynId", listaLokalizacji.SelectedValue);

sqlCommand.Parameters.AddWithValue("@ProduktId", listaAllProdukt.SelectedValue);

sqlCommand.ExecuteScalar();

}

catch (Exception ex)

{

MessageBox.Show("Wybierz produkt z bazy oraz lokalizację");

}

finally

{

sqlConnection.Close();

ShowProduktLokalizacji();

}

}

// PRZYCISK DODAJ PRODUKT DO BAZY

private void Button\_Click\_DodajProduktdoBazy(object sender, RoutedEventArgs e)

{

try

{

string query = "insert into Produkt values (@Nazwa)";

SqlCommand sqlCommand = new SqlCommand(query, sqlConnection);

sqlConnection.Open();

sqlCommand.Parameters.AddWithValue("@Nazwa", textBox2.Text);

sqlCommand.ExecuteScalar();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

finally

{

sqlConnection.Close();

ShowAllProdukt();

}

}

// USUŃ PRODUKT Z BAZY DANYCH

private void Button\_Click\_UsunProduktzBazy(object sender, RoutedEventArgs e)

{

try

{

string query = "delete from Produkt where id = @ProduktId";

SqlCommand sqlCommand = new SqlCommand(query, sqlConnection);

sqlConnection.Open();

sqlCommand.Parameters.AddWithValue("@ProduktId", listaAllProdukt.SelectedValue);

sqlCommand.ExecuteScalar();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

finally

{

sqlConnection.Close();

ShowAllProdukt();

}

}

// // USUŃ PRODUKT Z lokalizacji

private void Button\_Click\_UsunProduktzLokalizacji(object sender, RoutedEventArgs e)

{

try

{

string query = "delete TOP(1) from MagazynProdukt where MagazynId = @MagazynId AND ProduktId = @ProduktId";

SqlCommand sqlCommand = new SqlCommand(query, sqlConnection);

sqlConnection.Open();

sqlCommand.Parameters.AddWithValue("@MagazynId", listaLokalizacji.SelectedValue);

sqlCommand.Parameters.AddWithValue("@ProduktId", listaProduktLokalizacji.SelectedValue);

sqlCommand.ExecuteScalar();

}

catch (Exception ex)

{

MessageBox.Show("Wybierz produkt z lokalizacji aby usunąć");

}

finally

{

sqlConnection.Close();

ShowProduktLokalizacji();

}

}

// AKTUALIZACJA TEXTU W TEXT\_BOXIE 1

private void ShowSelectedinTextBox1()

{

try

{

// utworzenie zaptania i połączenie do bazy

string query = "select Lokalizacja from Magazyn where Id = @MagazynId";

SqlCommand sqlCommand = new SqlCommand(query, sqlConnection);

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand);

using (sqlDataAdapter)

{

sqlCommand.Parameters.AddWithValue("@MagazynId", listaLokalizacji.SelectedValue);

//Tworzenie Tabeli

DataTable MagazynDataTable = new DataTable();

//Fill(DataSet) - Dodaje lub odświeża wiersze w obiekcie DataSet , aby dopasować je do tych w źródle danych.

sqlDataAdapter.Fill(MagazynDataTable);

//Zmiana tekstu w textbox

textBox1.Text = MagazynDataTable.Rows[0]["Lokalizacja"].ToString();

//Zmiana tekstu w LABEL

LabelLokalizacja.Content = "PRODUKTY W LOKALIZACJI " + MagazynDataTable.Rows[0]["Lokalizacja"].ToString();

}

}

catch (Exception e)

{

// MessageBox.Show(e.ToString());

}

}

// AKTUALIZACJA TEXTU W TEXT\_BOXIE 2

private void ShowSelectedinTextBox2()

{

try

{

// utworzenie zaptania i połączenie do bazy

string query = "select Nazwa from Produkt where Id = @ProduktId";

SqlCommand sqlCommand = new SqlCommand(query, sqlConnection);

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand);

using (sqlDataAdapter)

{

sqlCommand.Parameters.AddWithValue("@ProduktId", listaAllProdukt.SelectedValue);

//Tworzenie Tabeli

DataTable ProduktDataTable = new DataTable();

//Fill(DataSet) - Dodaje lub odświeża wiersze w obiekcie DataSet , aby dopasować je do tych w źródle danych.

sqlDataAdapter.Fill(ProduktDataTable);

textBox2.Text = ProduktDataTable.Rows[0]["Nazwa"].ToString();

}

}

catch (Exception e)

{

// MessageBox.Show(e.ToString());

}

}

private void listaAllProdukt\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

// AKTUALIZACJA TEXTU W TEXT\_BOXIE 2

ShowSelectedinTextBox2();

}

// Przycick\_AktualizacjaLokalizacji

private void Button\_Click\_AktualizacjaLokalizacji(object sender, RoutedEventArgs e)

{

try

{

string query = "update Magazyn Set Lokalizacja = @Lokalizacja where Id = @MagazynId";

SqlCommand sqlCommand = new SqlCommand(query, sqlConnection);

sqlConnection.Open();

sqlCommand.Parameters.AddWithValue("@MagazynId", listaLokalizacji.SelectedValue);

sqlCommand.Parameters.AddWithValue("@Lokalizacja", textBox1.Text);

sqlCommand.ExecuteScalar();

}

catch (Exception ex)

{

MessageBox.Show("Zaznacz LOKALIZACJĘ aby dokonać AKTUALIZACJI");

}

finally

{

sqlConnection.Close();

ShowMagazyn();

}

}

// Przycick\_Aktualizacja PRODUKTÓW

private void Button\_Click(object sender, RoutedEventArgs e)

{

try

{

string query = "update Produkt Set Nazwa = @Nazwa where Id = @ProduktId";

SqlCommand sqlCommand = new SqlCommand(query, sqlConnection);

sqlConnection.Open();

sqlCommand.Parameters.AddWithValue("@ProduktId", listaAllProdukt.SelectedValue);

sqlCommand.Parameters.AddWithValue("@Nazwa", textBox2.Text);

sqlCommand.ExecuteScalar();

}

catch (Exception ex)

{

MessageBox.Show("Zaznacz PRODUKT aby dokonać AKTUALIZACJI");

}

finally

{

sqlConnection.Close();

ShowAllProdukt();

}

}

}

}