**30 Работа с базами данных. Технология ADO.NET**

Задание 1. В любой СУБД создайте базу данных DBTur\_firm (Структура и описание таблиц базы данных DBTur\_firm – стр.18, Схема данных базы DBTur\_firm– стр. 21). Для примера рассмотрите создание базы данных в MSSQL Server стр. 22-27. На стр. 28 представлена схема базы данных.

Таблица 30.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  |  |

Анализ результатов:



Рисунок 30.1 – Результат работы программы

Задание 2. Создайте .Net приложение (конкольное WinForm, WPF любое на выбор) и выполните подключение к разработанной базе данных, для этого изучите материал главы 4, пункт 4.1. Создайте и выполните SQL запросы: выберите все туры; удалите тур; добавьте туриста; измените туриста

Листинг программы:

using System.Data;

using System.Windows;

using System.Data.SQLite;

using System.IO;

namespace Task1

{

public partial class MainWindow : Window

{

private readonly string connectionString = @"DataSource=../../../Database/DBTur\_firm.db;Version=3;";

public MainWindow()

{

InitializeComponent();

LoadAllTourists();

}

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

{

LoadAllTours();

}

private void LoadAllTours()

{

using (SQLiteConnection connection = new(connectionString))

{

connection.Open();

string query = "SELECT \* FROM Туры";

SQLiteDataAdapter adapter = new(query, connection);

DataTable dataTable = new();

adapter.Fill(dataTable);

dataGridTours.ItemsSource = dataTable.DefaultView;

}

}

private void LoadAllTourists()

{

using (SQLiteConnection connection = new(connectionString))

{

MessageBox.Show(Path.GetFullPath(@"../../../Database/DBTur\_firm.db"));

connection.Open();

string query = "SELECT \* FROM Туристы";

SQLiteDataAdapter adapter = new SQLiteDataAdapter(query, connection);

DataTable dataTable = new DataTable();

adapter.Fill(dataTable);

dataGridTourists.ItemsSource = dataTable.DefaultView;

}

}

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

{

DeleteTouristWindow deleteTouristWindow = new();

if (deleteTouristWindow.ShowDialog() == true)

{

int touristId = deleteTouristWindow.TourCode;

using (SQLiteConnection connection = new(connectionString))

{

connection.Open();

string query = "DELETE FROM Туры WHERE Код\_тура = @Код\_тура";

SQLiteCommand command = new(query, connection);

command.Parameters.AddWithValue("@Код\_тура", touristId);

int rowsAffected = command.ExecuteNonQuery();

if (rowsAffected > 0)

{

MessageBox.Show("Тур удален.");

}

else

{

MessageBox.Show("Тур с таким кодом не найден.");

}

}

LoadAllTours();

}

}

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

{

AddTouristWindow addTouristWindow = new();

if (addTouristWindow.ShowDialog() == true)

{

string name = addTouristWindow.Name;

string surname = addTouristWindow.Surname;

string middleName = addTouristWindow.MiddleName;

using (SQLiteConnection connection = new(connectionString))

{

connection.Open();

string query = "INSERT INTO Туристы (Фамилия, Имя, Отчество) VALUES (@Фамилия, @Имя, @Отчество)";

SQLiteCommand command = new SQLiteCommand(query, connection);

command.Parameters.AddWithValue("@Фамилия", surname);

command.Parameters.AddWithValue("@Отчество", middleName);

command.Parameters.AddWithValue("@Имя", name);

command.ExecuteNonQuery();

}

MessageBox.Show("Турист добавлен.");

LoadAllTourists();

}

}

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

{

UpdateTouristWindow updateTouristWindow = new UpdateTouristWindow();

if (updateTouristWindow.ShowDialog() == true)

{

int touristId = updateTouristWindow.TouristCode;

string firstName = updateTouristWindow.Name;

string lastName = updateTouristWindow.Surname;

string patronymic = updateTouristWindow.MiddleName;

using (SQLiteConnection connection = new SQLiteConnection(connectionString))

{

connection.Open();

string query = "UPDATE Туристы SET Фамилия = @Фамилия, Имя = @Имя, Отчество = @Отчество WHERE Код\_туриста = @Код\_туриста";

SQLiteCommand command = new SQLiteCommand(query, connection);

command.Parameters.AddWithValue("@Код\_туриста", touristId);

command.Parameters.AddWithValue("@Фамилия", lastName);

command.Parameters.AddWithValue("@Имя", firstName);

command.Parameters.AddWithValue("@Отчество", patronymic);

int rowsAffected = command.ExecuteNonQuery();

if (rowsAffected > 0)

{

MessageBox.Show("Турист обновлен.");

}

else

{

MessageBox.Show("Турист с таким кодом не найден.");

}

}

LoadAllTourists();

}

}

}

}

Таблица 30.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  |  |

Анализ результатов:

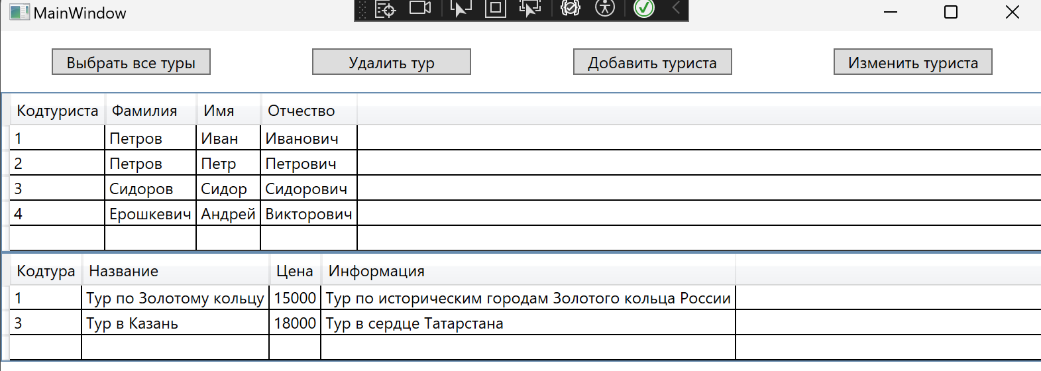


Рисунок 30.2 – Результат работы программы