Федеральное государственное бюджетное образовательное учреждение высшего образования

«Национальный исследовательский университет «Московский энергетический институт»

**Типовой расчёт**

по курсу **«Базы данных»**

Студент: Оськин А. С.

Группа: А-08-18

**Москва 2020**

1. **Тема типового расчёта**

***Деревня староверов Горловка.***

Жители деревни Горловка изготавливают товары натурального производства и занимаются их продажей. Также, деревня староверов интересует общественность, в деревне есть комплексов домов, предназначенных для сдачи туристам. Доход приносит и сувенирная лавка (товары в которой изготавливаются жителями).

1. **Описание бизнес процесса**

В деревне живут жители. Некоторые жители изготавливают товар натурального производства специально для деревни.  
Деревня получает контракт извне на определённый тип товара (молоко, говядина, шерсть).  
На каждый контракт есть житель, который контролирует его выполнение.

Также, есть группа домов для туристов, за каждым из которых следят жители.  
Эти дома сдаются по договору с туристом.

Деревня получает доход и с сувенирной лавки (все сувениры делаются жителями).

Необходимо хранить и обрабатывать информацию, связанную с

- контрактами деревни и внешних организаций (к какому сроку, в каком объёме товар должен быть готов, какой житель следит за выполнением)

- контрактами деревни и жителями (какой товар производит определённый житель и в каком объёме)

- содержимым сувенирной лавки (цена товара и изготовитель)

- домами для туристов (какой житель следит за домом, сколько комнат в доме, есть ли горячая вода, цена за сутки пребывания)

- туристическими контрактами (кто снимает дом, срок на который снимает, сколько людей будут проживать в доме)

Нужно добавлять и удалять информацию, связанную с

- жителями

- контрактами

- содержимом сувенирной лавки

- домами для туристов

**3. Группы пользователей**

С базой данных будут работать три типа пользователей:

1. Старейшина. Может редактировать всю информацию. Может использовать функции для обработки данных.
2. Помощник старейшины. Может регистрировать новые контракты, добавлять сувениры в сувенирную лавку, менять информацию о домах.
3. Житель. Может посмотреть информацию о себе и своих контрактах.

**4. Концептуальное проектирование ER-диаграмма. Таблицы, индексы, ключи.**

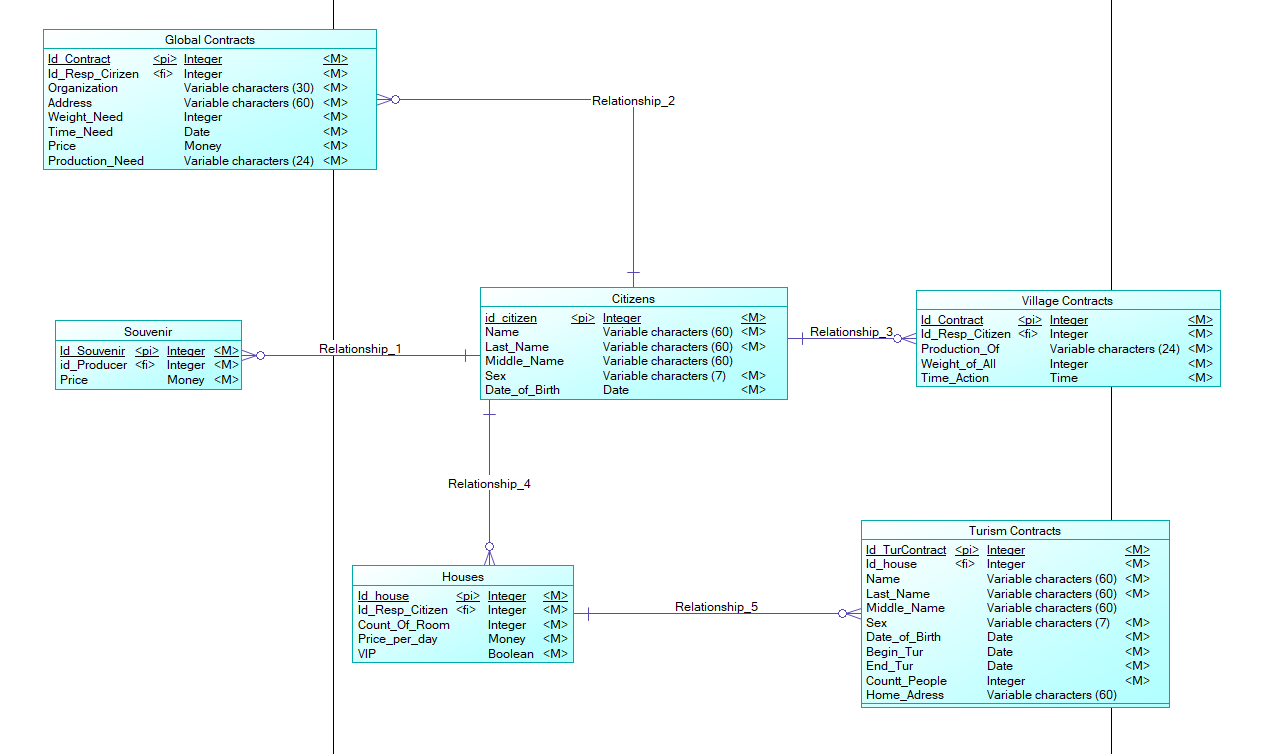


Рис. 1. ER-диаграмма

Таблица Citizens содержит информацию о жителях: Имя, Фамилия, Отчество, гендер, дата рождения.

Таблица Souvenir содержит информацию о сувенирной лавке: цену товара и внешний ключ, ссылающийся на таблицу Citizens.

Таблица Village Contracts содержит информацию о жителях-производителях:   
id\_Resp\_Citizen – внешний ключ, ссылающийся на Citizens, Production\_Of – информация о том, товар какого типа изготавливается, Weight\_of\_All – объем готового товара (в кг).

Таблица Global Contracts содержит информацию о контрактах деревни с внешними организациями: название организации, её адрес, какой объём необходим (в кг), к какому сроку, цена, внешний ключ, ссылающийся на жителя контролирующего выполнение заказа, из таблицы Citizens.

Таблица Houses хранит информацию о домах: число комнат в доме, цена за день пребывания, наличие горячей воды, внешний ключ, ссылающийся на жителя, смотрящего за домом из таблицы Citizens.

Таблица Turism Contracts содержит информацию о туристических контрактах: Фамилия, Имя и Отчество того, кто снимает дом, его пол, дата рождения, дата заселения, дата выселения, число людей, которые будут жить в доме, домашний адрес, внешний ключ, ссылающийся на дом из таблицы Houses, на который оформляется контракт.

Связь между таблицей Citizens и таблицами Houses, Village Contracts, Global Contracts, Souvenir – один-ко-многим, так как один житель может следить за многими домами, производить более одного товара, следить за выполнением более одного контракта, быть производителем многих сувениров из сувенирной лавки.

Связь между таблицей Houses и Turism Contracts один-к-одному, так как на один контракт можно снять только один дом.

Ограничения:

Для добавления жителя в таблицу жителей, необходимо, чтобы ему было от 16 лет.

Минимальный объём товара для заказа – 5 кг.

Снять дом можно минимум на 6 дней.

Индексы:

Для ускорения поиска сделаем неуникальные индексы:

Индекс по номеру дома.

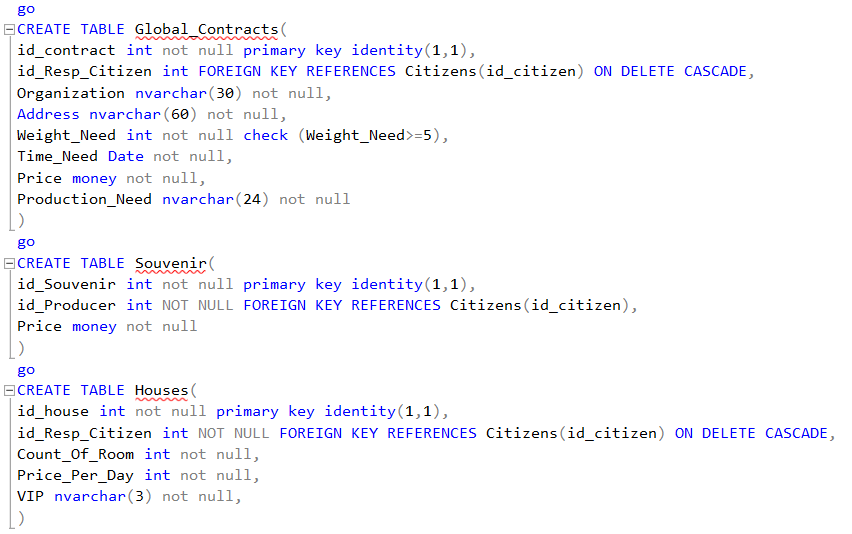
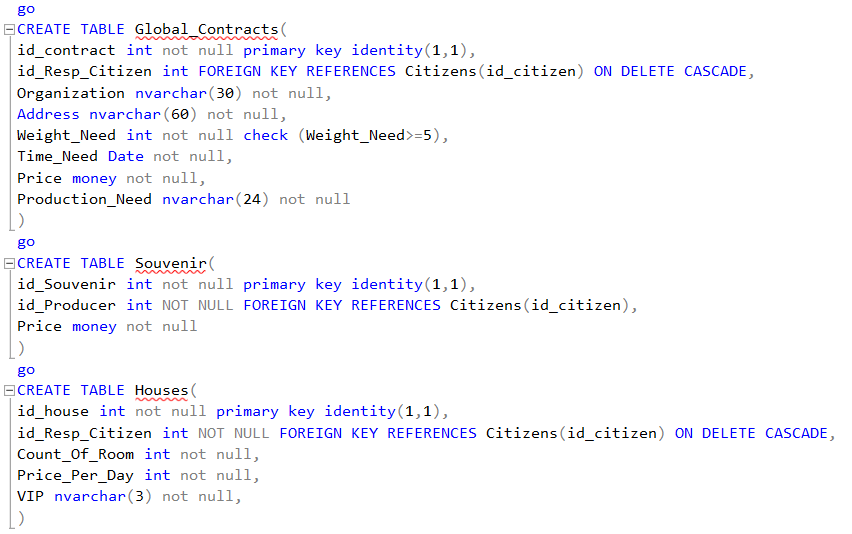
Индекс по наличию горячей в доме.

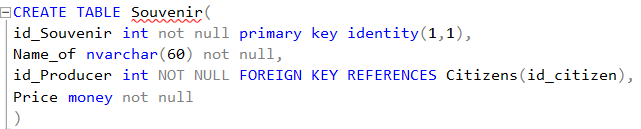
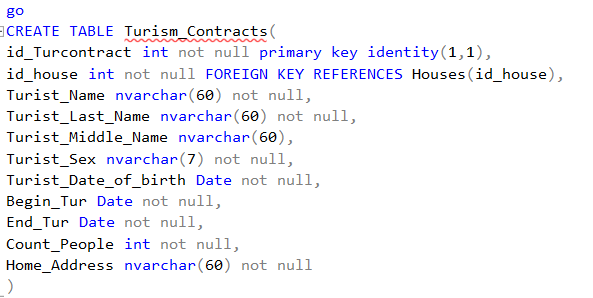
Индекс по количеству комнат в доме.

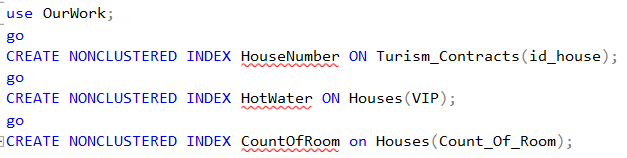
**6. Создание физической модели БД  
Создание БД  
**

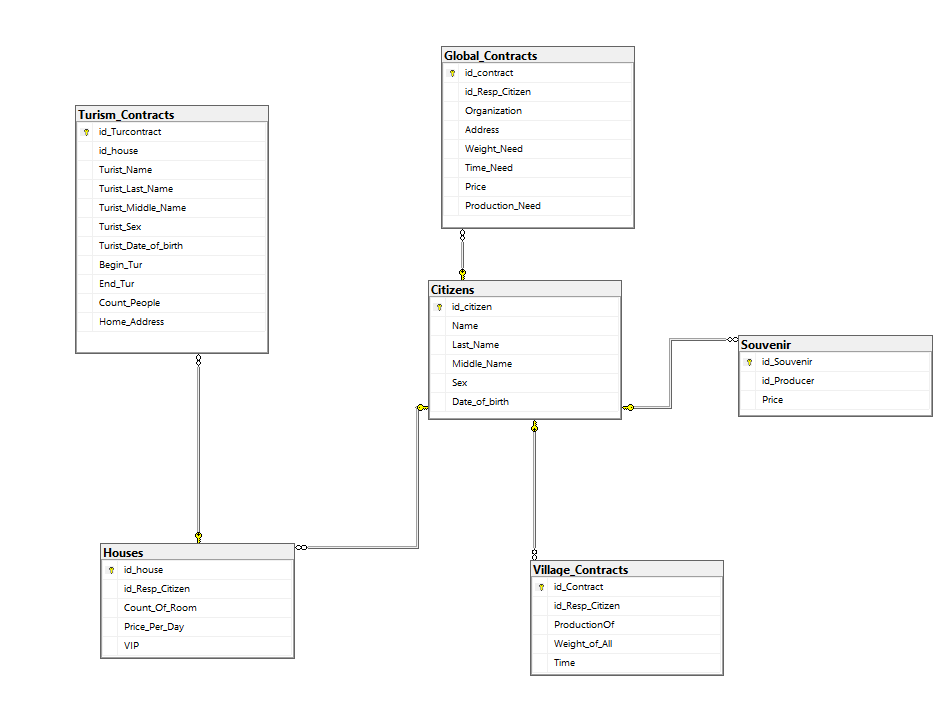
**Создание таблиц**

Создание таблиц Citizens, Village\_Contracts, Houses, Global\_Contracts, Souvenir, Turism Contracts с необходимыми ограничениями. ****

****

**  
  
  
Создание индексов**

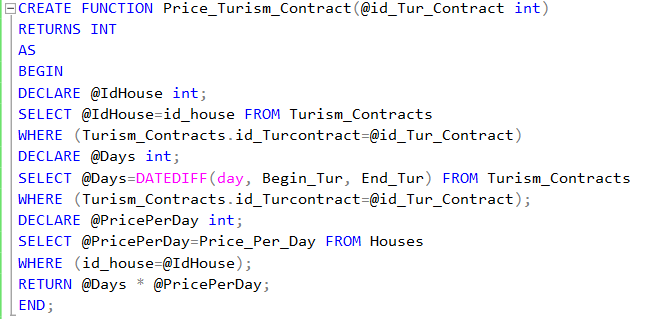
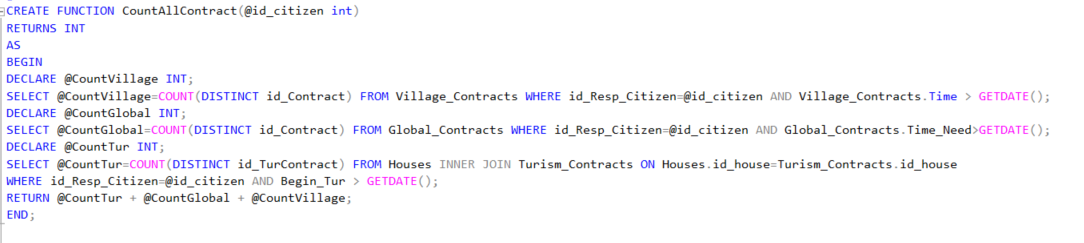
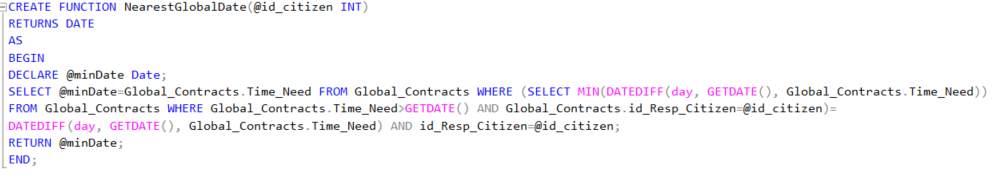
Создание необходимых индексов. ****

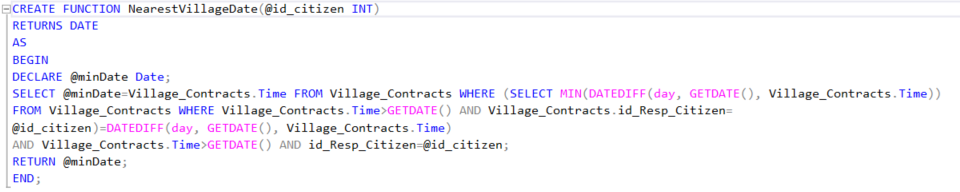
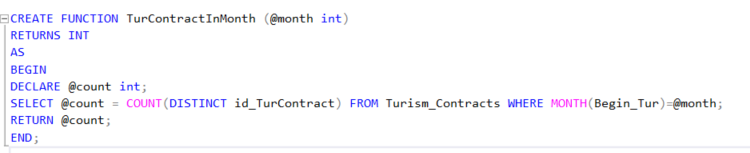
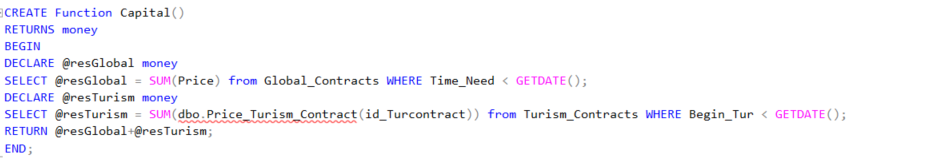
****Рис.2. Визуальное представление созданной базы данных в MS SQL

**6. Создание функций и представления.**

**Создание скалярных функции**

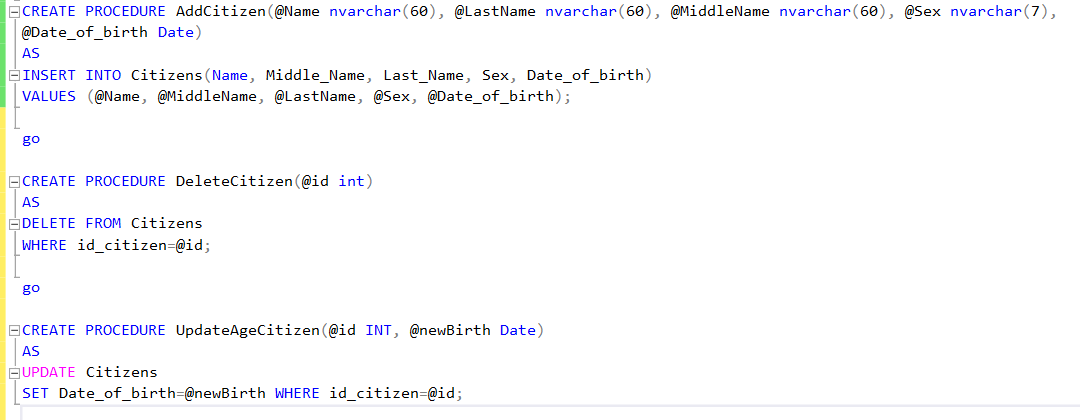
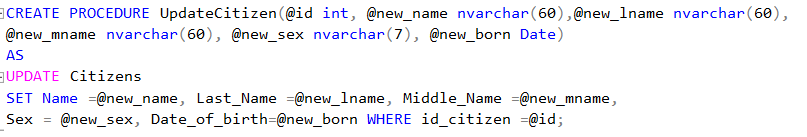
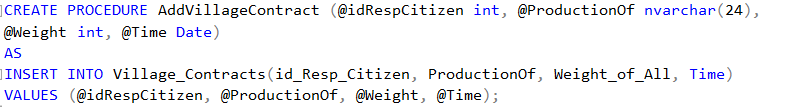
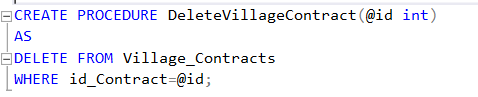
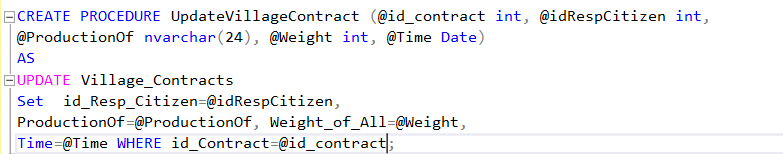
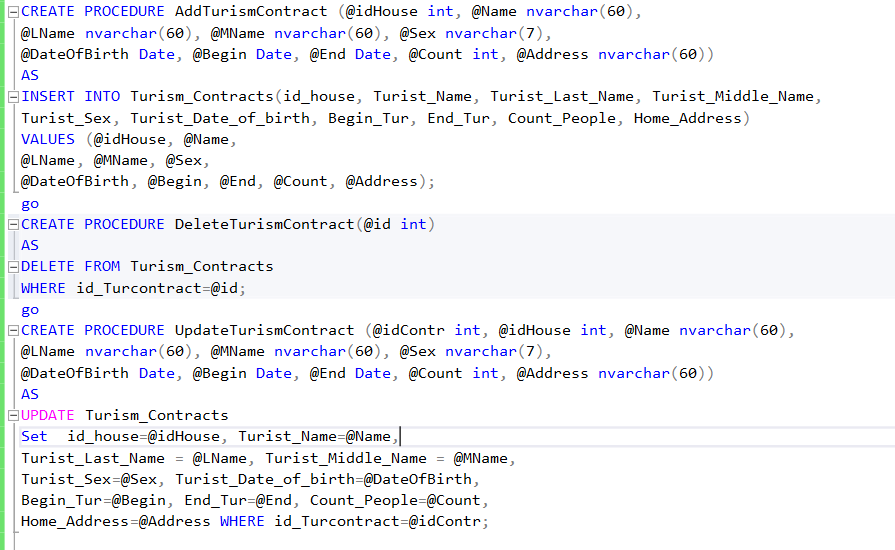
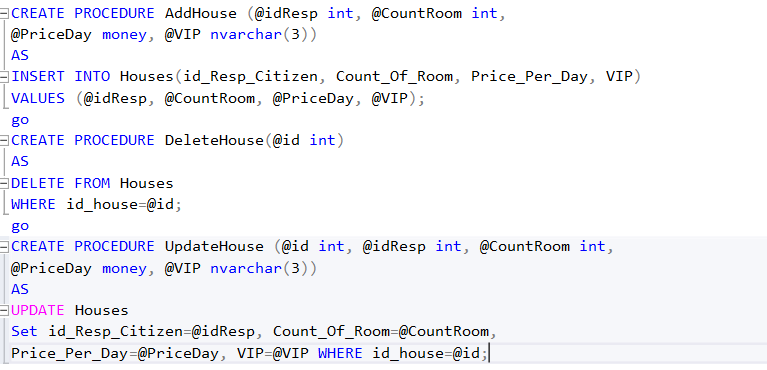
Создание функции Price\_Turism\_Contract, данная функция помогает вычислить полную стоимость туристического контракта по его ID.

****Создание функции CountAllContract. Эта функция возвращает число контрактов, сроки которых ещё не прошли. ****  
Функция NearestGlobalDate возвращает ближайший срок глобального контракта для жителя. ****

Функция NearestVillageDate возвращает ближайший срок контракта с деревней для жителя. ****  
Функция TurContractInMonth возвращает число туристов, которые когда-либо заселялись в указанном месяце. ****  
Функция Capital возвращает общую заработанную сумму за всё время. **  
Создание табличной функции**Функция WhoWork возвращает ФИО жителей, работающих в указанной области ****

**7. Необходимые процедуры для манипуляции с БД**

Ниже приведены процедуры для добавления, удаления, обновления данных в таблицы Citizen, Global\_Contracts, Village\_Contracts, Turism\_Contracts, Houses

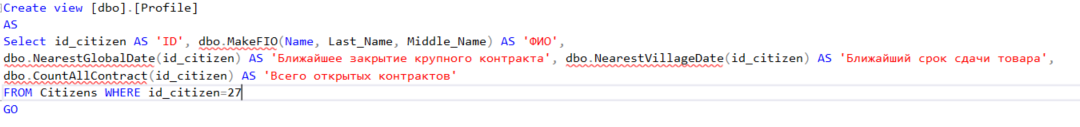
**  
  
  
  
  
  
  
  
**

**8**. **Создание необходимых представлений**

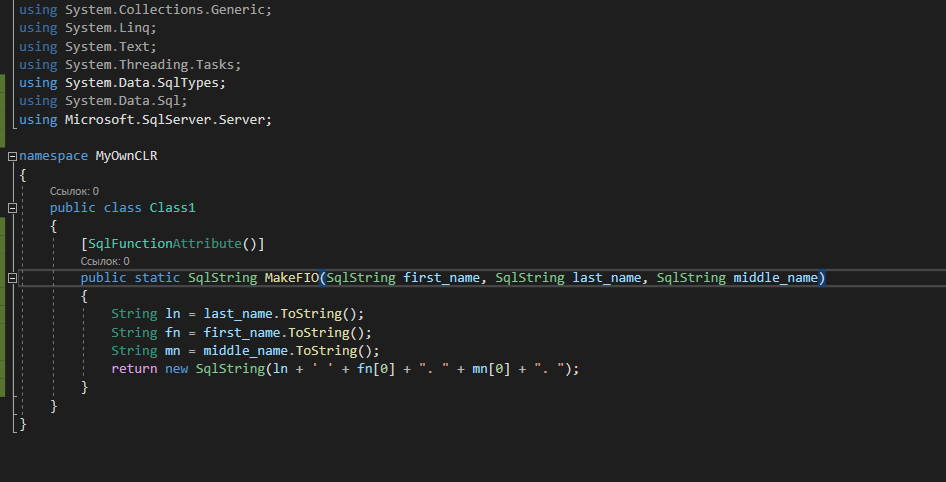
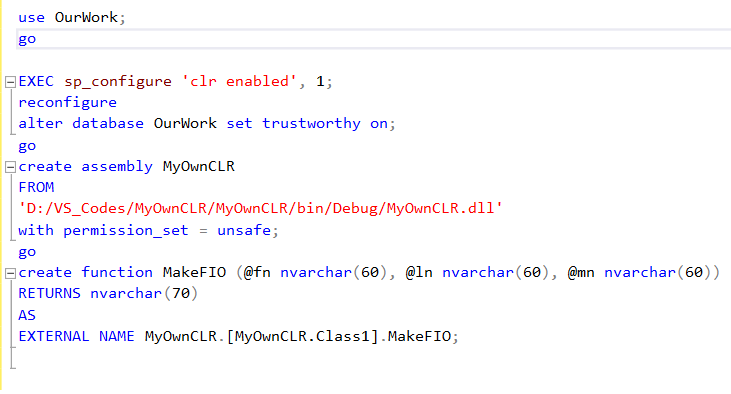
Представление Profiles создано для старейшины и его помощника.

****

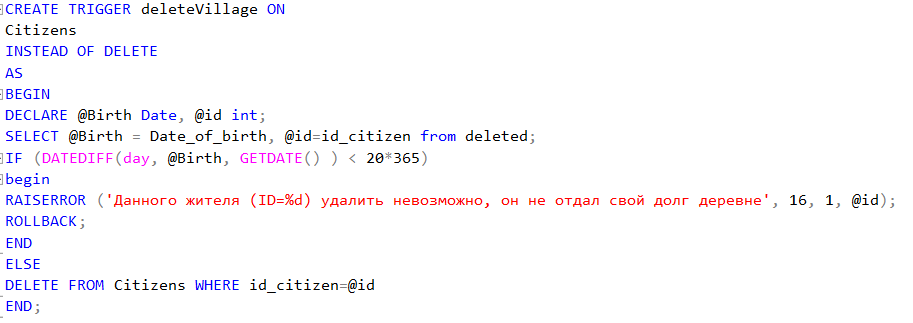
Представление Profile создано для жителя.

****

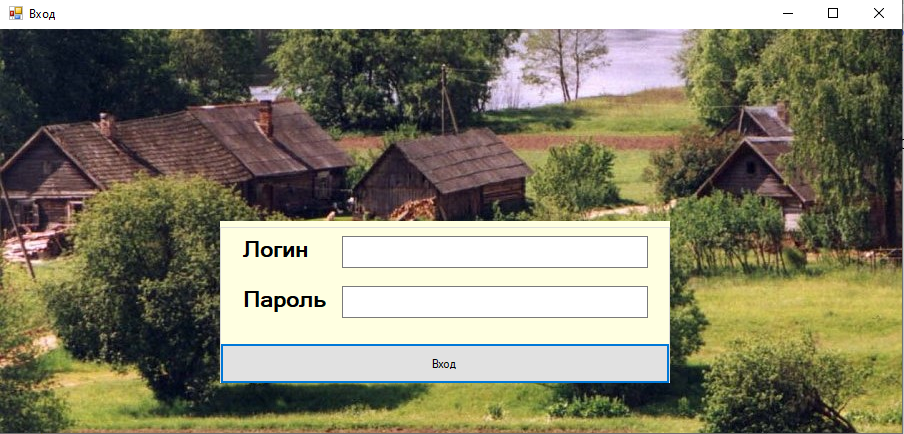
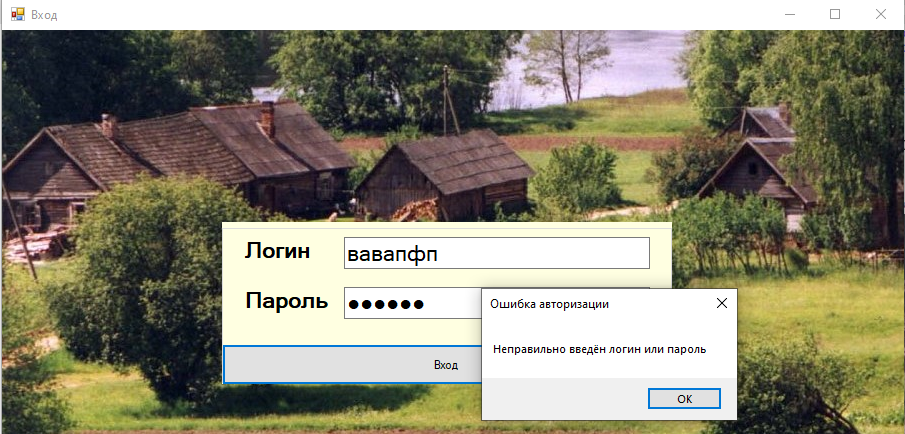
**9. Создать CLR-функцию**

Напишем CLR-функцию, которая генерирует ФИО в формате «Фамилия И. О.» из Имени, Фамилии и Отчества. **  
**

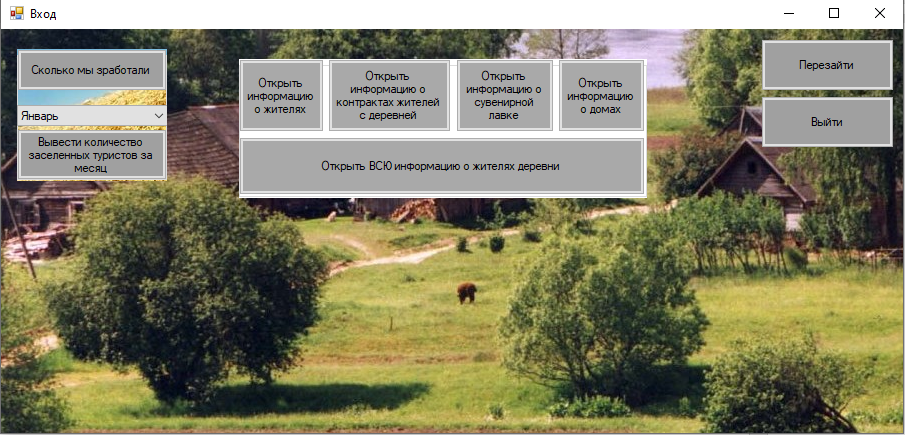
**10. Создать триггер**

Создадим триггер, запрещающий удалять жителя, если ему меньше 20 лет. ****

**11. Создать клиент-серверное приложение на С#**

****Рис.11.1 Окно входа  
 ****Рис.11.2. Результат попытки входа при некорректно заполненных полях

**Старейшина**

****Рис.11.3. Интерфейс старейшины

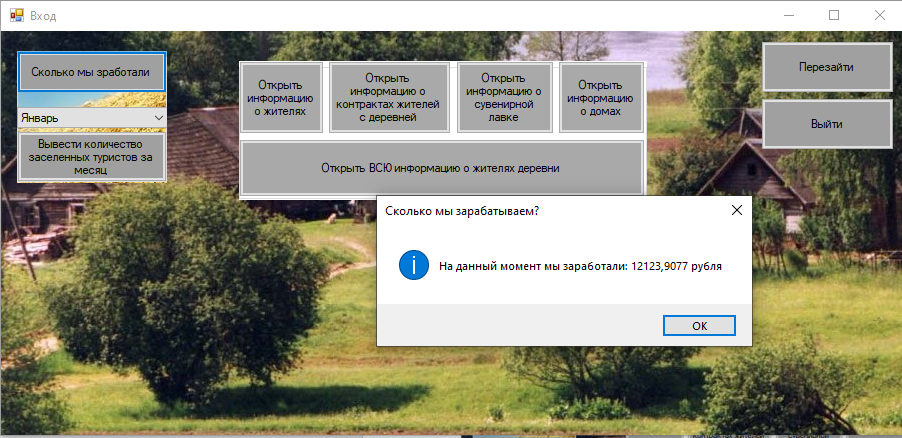
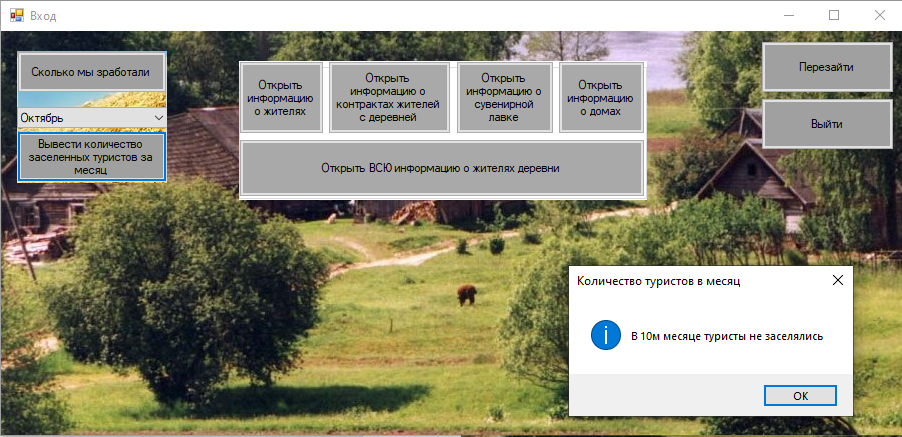
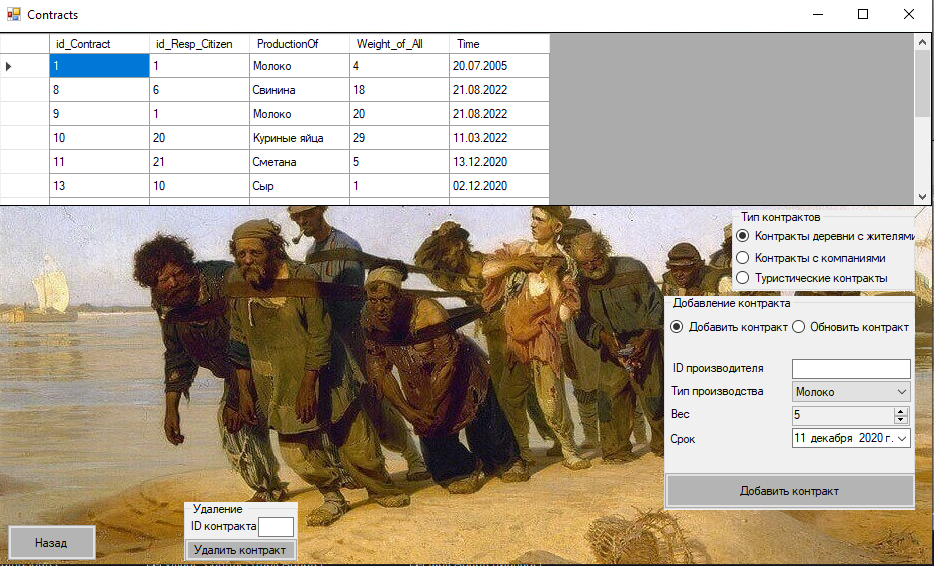
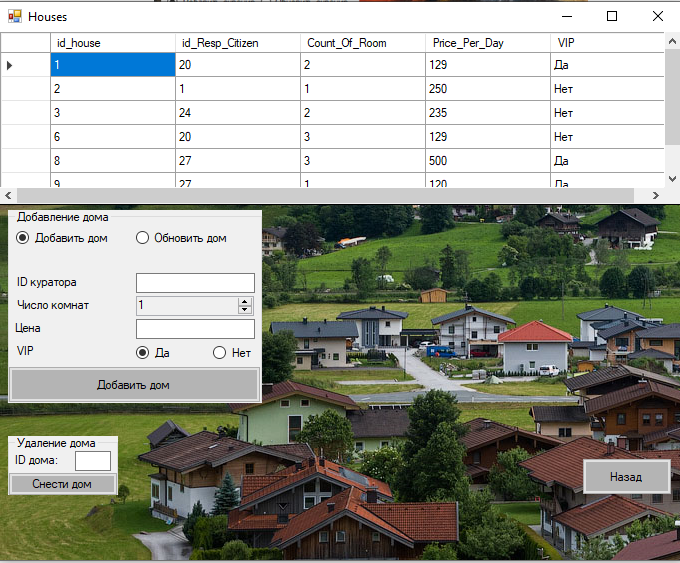
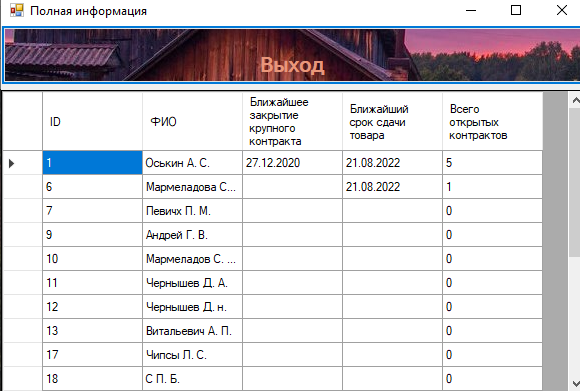
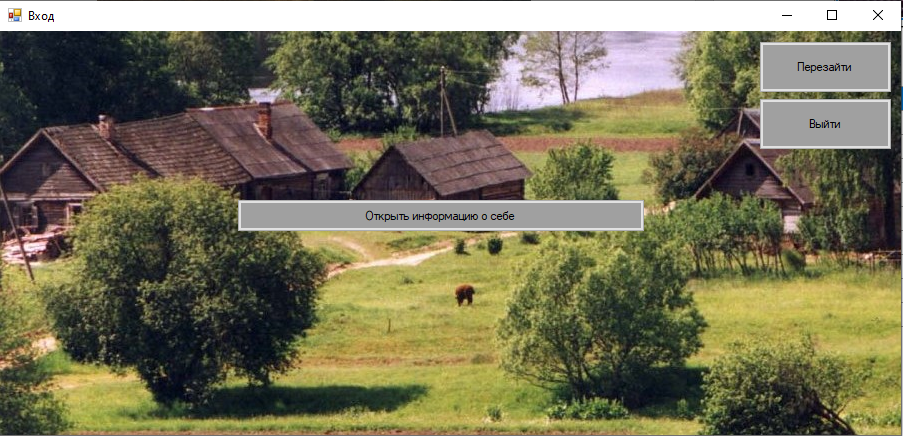
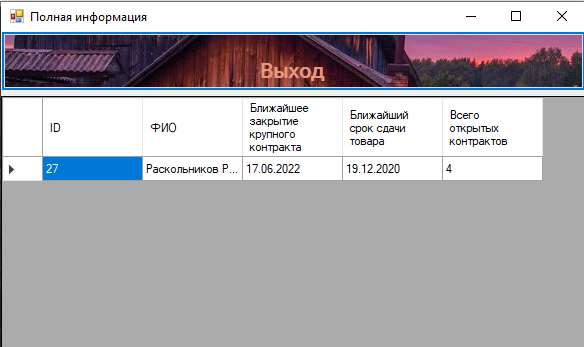
  
Рис.11.4. Результат нажатия на кнопку «Сколько мы заработали»  
  
  
Рис.11.5. Результат нажатия на кнопку «Вывести количество заселенных туристов за месяц», месяц задаётся в comboBox

  
Рис.11.6. Информация о жителях для старейшины  
  
Рис.11.7. Информация о контрактах для старейшины  
  
Рис.11.8. Информация о сувенирах для старейшины  
  
Рис.11.9. Информация о домах для старейшины  
  
Рис.11.10. Информация о профилях жителей для старейшины

**Помощник старейшины**Помощник старейшины обладает практически теми же данными, что и сам старейшина за исключением возможности удаления каких-либо данных (на формах отсутствуют groupBox, отвечающие за удаление), возможности какого-либо изменения таблицы жителей, возможности обновления контрактов (radioButton, отвечающие ща обновление имеют свойство Enabled=false), возможности обновления информации о сувенирах, возможности добавления новых домов.  
Также помощник старейшины не имеет groupBox с кнопками «Сколько мы заработали» и «Вывести количество заселенных туристов за месяц»

**Житель**

  
Рис.11.11. Интерфейс жителя  
****  
Рис.11.12. Информация, доступная жителю

**Приложение  
Код С#  
Файл «Form1.cs»**using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

//Старейшина

namespace OuurWork

{

public partial class mainForm : Form

{

string connectString = "Data Source=RRESHRRPC;Initial Catalog=OurWork;Integrated Security = True";

public mainForm()

{

InitializeComponent();

GroupLogin.Visible = true;

AdminBox.Visible = false;

Exit.Visible = false;

somefunctions.Visible = false;

CitizInfoBut.Visible = false;

Logout.Visible = false;

}

bool admin;

//Старейшина

//логин - Oskin Andrey

//пароль - 2000.20.07

// Помощник

//логин - Good Boy

//пароль - iamhelper

// житель

//логин - Grandfather Sergey

//пароль - ilikepotato

private void buttonIn\_Click(object sender, EventArgs e)

{

if ((loginText.Text == "Oskin Andrey") && (pass\_text.Text == "2000.20.07"))

{

AdminBox.Visible = true;

Exit.Visible = true;

Logout.Visible = true;

GroupLogin.Visible = false;

admin = true;

somefunctions.Visible = true;

monthBox.SelectedItem = "Январь";

}

else if ((loginText.Text == "Good Boy") && (pass\_text.Text == "iamhelper"))

{

AdminBox.Visible = true;

GroupLogin.Visible = false;

Exit.Visible = true;

Logout.Visible = true;

admin = false;

}

else if ((loginText.Text == "Rodion Romanovich") && (pass\_text.Text == "ilikepotato"))

{

Exit.Visible = true;

Logout.Visible = true;

GroupLogin.Visible = false;

CitizInfoBut.Visible = true;

} else

{

MessageBox.Show("Неправильно введён логин или пароль", "Ошибка авторизации", MessageBoxButtons.OK);

}

}

private void AdminCitizens\_Click(object sender, EventArgs e)

{

Citizens citizensForm = new Citizens(admin);

citizensForm.Show();

}

private void AdminVILContracts\_Click(object sender, EventArgs e)

{

Village\_Contracts formVC = new Village\_Contracts(admin);

formVC.Show();

}

private void ADMINView\_Click(object sender, EventArgs e)

{

All\_Info formALL = new All\_Info("admin");

formALL.Show();

}

private void contextMenuStrip1\_Opening(object sender, CancelEventArgs e)

{

}

private void AdminHouses\_Click(object sender, EventArgs e)

{

Houses form = new Houses(admin);

form.Show();

}

private void AdminSouvenir\_Click(object sender, EventArgs e)

{

Souvenir form = new Souvenir(admin);

form.Show();

}

private void AdminBox\_Enter(object sender, EventArgs e)

{

}

private void Exit\_Click(object sender, EventArgs e)

{

this.Close();

}

private void Logout\_Click(object sender, EventArgs e)

{

Exit.Visible = false;

Logout.Visible = false;

GroupLogin.Visible = true;

AdminBox.Visible = false;

CitizInfoBut.Visible = false;

somefunctions.Visible = false;

}

private void CitizInfoBut\_Click(object sender, EventArgs e)

{

All\_Info form = new All\_Info("citizen");

form.Show();

}

private void HowMuch\_Click(object sender, EventArgs e)

{

SqlConnection connection = new SqlConnection(connectString);

connection.Open();

string command = "SELECT dbo.Capital();";

var res = new SqlCommand(command, connection);

var data = res.ExecuteScalar();

string finalres = "На данный момент мы зарaботали: " + data.ToString() + " рубля";

MessageBox.Show(finalres, "Сколько мы зарабатываем?", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

private void button1\_Click(object sender, EventArgs e)

{

int month = monthBox.Items.IndexOf(monthBox.SelectedItem) + 1;

SqlConnection connection = new SqlConnection(connectString);

connection.Open();

string command = "SELECT dbo.TurContractInMonth(" + month.ToString() + ");";

var res = new SqlCommand(command, connection);

var data = res.ExecuteScalar();

string finalres;

if (Convert.ToInt32(data) > 0)

{

finalres = "В " + month + "м месяце заселилось туристов:" + data.ToString();

}

else

finalres = "В " + month + "м месяце туристы не заселялись";

MessageBox.Show(finalres, "Количество туристов в месяц", MessageBoxButtons.OK, MessageBoxIcon.Information); }

}

}

**Файл «Citizens.cs»**using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace OuurWork

{

public partial class Citizens : Form

{

string connectString= "Data Source=RRESHRRPC;Initial Catalog=OurWork;Integrated Security = True";

public Citizens(bool admin)

{

InitializeComponent();

idForUp.Visible = false;

label7.Visible = false;

sexCitizenMale.Checked = true;

AdminAddCitizen.Text = AddUpBut.Text;

if (!admin)

{

DeleteBoxAdmin.Visible = false;

AdminAddCitizen.Visible = false;

}

}

private void citizensBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.citizensBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.ourWorkDataSet);

}

private void Citizens\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "ourWorkDataSet.Citizens". При необходимости она может быть перемещена или удалена.

this.citizensTableAdapter.Fill(this.ourWorkDataSet.Citizens);

}

private void button1\_Click(object sender, EventArgs e)

{

if ((String.Empty == adminMiddleNameAdd.Text)||(String.Empty == adminLastnameAdd.Text)

||(String.Empty == adminAdd.Text))

{

MessageBox.Show("Заполните поля корректно");

return;

}

using (SqlConnection connection = new SqlConnection(connectString))

{

connection.Open();

if (AddCitizen.Checked)

{

string sqlExp = "AddCitizen";

SqlCommand command = new SqlCommand(sqlExp, connection);

command.CommandType = System.Data.CommandType.StoredProcedure; //хранимые процедуры

SqlParameter Name = new SqlParameter

{

ParameterName = "@Name",

Value = adminAdd.Text

};

SqlParameter LastName = new SqlParameter

{

ParameterName = "@LastName",

Value = adminLastnameAdd.Text

};

SqlParameter MiddleName = new SqlParameter

{

ParameterName = "@MiddleName",

Value = adminMiddleNameAdd.Text

};

string s = "";

if (sexCitizenMale.Checked)

{

s = "Мужчина";

} else

{

s = "Женщина";

}

SqlParameter Sex = new SqlParameter

{

ParameterName = "@Sex",

Value = s

};

SqlParameter DateOfBirth = new SqlParameter

{

ParameterName = "@Date\_of\_birth",

Value = HelpfulClass.GetValidDateForSql(AdminDateOfBirthAdd)

};

command.Parameters.Add(Name);

command.Parameters.Add(LastName);

command.Parameters.Add(MiddleName);

command.Parameters.Add(Sex);

command.Parameters.Add(DateOfBirth);

command.ExecuteNonQuery();

} else

{

if (String.Empty == idForUp.Text)

{

MessageBox.Show("Заполните поля корректно");

return;

}

string sqlExp = "UpdateCitizen";

SqlCommand command = new SqlCommand(sqlExp, connection);

command.CommandType = System.Data.CommandType.StoredProcedure; //хранимые процедуры

SqlParameter pr1 = new SqlParameter

{

ParameterName = "@new\_name",

Value = adminAdd.Text

};

SqlParameter pr2 = new SqlParameter

{

ParameterName = "@new\_lname",

Value = adminLastnameAdd.Text

};

SqlParameter pr3 = new SqlParameter

{

ParameterName = "@new\_mname",

Value = adminMiddleNameAdd.Text

};

string s = "";

if (sexCitizenMale.Checked)

{

s = "Мужчина";

} else

{

s = "Женщина";

}

SqlParameter pr4 = new SqlParameter

{

ParameterName = "@new\_sex",

Value = s

};

SqlParameter pr5 = new SqlParameter

{

ParameterName = "@new\_born",

Value = HelpfulClass.GetValidDateForSql(AdminDateOfBirthAdd)

};

SqlParameter pr6 = new SqlParameter

{

ParameterName = "@id",

Value = Convert.ToInt32(idForUp.Text)

};

command.Parameters.Add(pr1);

command.Parameters.Add(pr2);

command.Parameters.Add(pr3);

command.Parameters.Add(pr4);

command.Parameters.Add(pr5);

command.Parameters.Add(pr6);

command.ExecuteNonQuery();

}

HelpfulClass.UpdateTable("Citizens", connection, citizensDataGridView);

connection.Close();

}

}

private void BackButton\_Click(object sender, EventArgs e)

{

this.Close();

}

private void deleteAdmin\_Click(object sender, EventArgs e)

{

if (idDeleted.Text == String.Empty)

{

MessageBox.Show("Заполните поля конректно");

return;

}

using (SqlConnection connection = new SqlConnection(connectString))

{

connection.Open();

string sqlExp = "DeleteCitizen";

SqlCommand command = new SqlCommand(sqlExp, connection);

command.CommandType = System.Data.CommandType.StoredProcedure; //хранимые процедуры

SqlParameter id = new SqlParameter

{

ParameterName = "@id",

Value = Convert.ToInt32(idDeleted.Text)

};

command.Parameters.Add(id);

command.ExecuteNonQuery();

HelpfulClass.UpdateTable("Citizens", connection, citizensDataGridView);

connection.Close();

}

}

private void UpdateCitizen\_CheckedChanged(object sender, EventArgs e)

{

if (UpdateCitizen.Checked)

{

AddUpBut.Text = "Обновить жителя";

idForUp.Visible = true;

label7.Visible = true;

AdminAddCitizen.Text = AddUpBut.Text;

}

}

private void AddCitizen\_CheckedChanged(object sender, EventArgs e)

{

if (AddCitizen.Checked)

{

AddUpBut.Text = "Добавить жителя";

idForUp.Visible = false;

label7.Visible = false;

AdminAddCitizen.Text = AddUpBut.Text;

}

}

private void DeleteBoxAdmin\_Enter(object sender, EventArgs e)

{

}

private void idForUp\_KeyPress(object sender, KeyPressEventArgs e)

{

var symb = e.KeyChar;

if (!Char.IsDigit(symb) && !Char.IsControl(symb))

{

e.Handled=true;

}

}

private void idDeleted\_KeyPress(object sender, KeyPressEventArgs e)

{

var symb = e.KeyChar;

if (!Char.IsDigit(symb) && !Char.IsControl(symb))

{

e.Handled = true;

}

}

}

}

**Файл «Village\_Contracts.cs»**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace OuurWork

{

public partial class Village\_Contracts : Form

{

string connectString = "Data Source=RRESHRRPC;Initial Catalog=OurWork;Integrated Security = True";

public Village\_Contracts(bool admin)

{

InitializeComponent();

radioVillageContract.Checked = true;

groupGlobalBox.Visible = false;

groupVilContractAddUp.Visible = true;

groupAddUpTurBox.Visible = false;

radioAddVilContract.Checked = true;

label7.Visible = false;

tboxVilContractID.Visible = false;

radioAddTurContr.Checked = true;

idTurContract.Visible = false;

labelForIDTur.Visible = false;

radioAddGlobal.Checked = true;

label4.Visible = false;

IdContrGlobal.Visible = false;

TypeGlobal.SelectedItem = "Молоко";

ProductionVil.SelectedItem = "Молоко";

if (!admin)

{

radioUpdGlobal.Enabled = false;

DeleteBoxAdmin.Visible = false;

radioUpTurContr.Enabled = false;

radioUpdateVilContract.Enabled = false;

}

using (SqlConnection connection = new SqlConnection(connectString))

{

connection.Open();

HelpfulClass.UpdateTable("Village\_contracts", connection, DataGridForContracts);

connection.Close();

}

}

private void village\_ContractsBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.village\_ContractsBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.ourWorkDataSet);

}

private void Village\_Contracts\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "ourWorkDataSet.Village\_Contracts". При необходимости она может быть перемещена или удалена.

this.village\_ContractsTableAdapter.Fill(this.ourWorkDataSet.Village\_Contracts);

}

private void button1\_Click(object sender, EventArgs e)

{

}

private void radioVillageContract\_CheckedChanged(object sender, EventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectString))

{

connection.Open();

HelpfulClass.UpdateTable("Village\_contracts", connection, DataGridForContracts);

connection.Close();

if (radioVillageContract.Checked)

{

groupGlobalBox.Visible = false;

groupVilContractAddUp.Visible = true;

groupAddUpTurBox.Visible = false;

}

}

}

private void radioGlobalContracts\_CheckedChanged(object sender, EventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectString))

{

connection.Open();

HelpfulClass.UpdateTable("Global\_contracts", connection, DataGridForContracts);

connection.Close();

}

if (radioGlobalContracts.Checked)

{

groupGlobalBox.Visible = true;

groupVilContractAddUp.Visible = false;

groupAddUpTurBox.Visible = false;

}

}

private void radioTurismContract\_CheckedChanged(object sender, EventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectString))

{

connection.Open();

HelpfulClass.UpdateTable("Turism\_contracts", connection, DataGridForContracts);

connection.Close();

}

if (radioTurismContract.Checked)

{

groupGlobalBox.Visible = false;

groupVilContractAddUp.Visible = false;

groupAddUpTurBox.Visible = true;

}

}

private void AddUpContractVil\_Click(object sender, EventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectString))

{

if (tBoxVilIDResp.Text=="")

{

MessageBox.Show("Заполните необходимые поля", "Пустое поле", MessageBoxButtons.OK);

return;

}

connection.Open();

SqlCommand command;

if (radioAddVilContract.Checked)

{

string sqlExp = "AddVillageContract";

command = new SqlCommand(sqlExp, connection);

command.CommandType = System.Data.CommandType.StoredProcedure; //хранимые процедуры

SqlParameter prm1 = new SqlParameter

{

ParameterName = "@idRespCitizen",

Value = Convert.ToInt32(tBoxVilIDResp.Text)

};

SqlParameter prm2 = new SqlParameter

{

ParameterName = "@ProductionOf",

Value = ProductionVil.SelectedItem.ToString()

};

SqlParameter prm3 = new SqlParameter

{

ParameterName = "@Weight",

Value = Convert.ToInt32(WeightVil.Value)

};

SqlParameter prm4 = new SqlParameter

{

ParameterName = "@Time",

Value = HelpfulClass.GetValidDateForSql(DateVil)

};

command.Parameters.Add(prm1);

command.Parameters.Add(prm2);

command.Parameters.Add(prm3);

command.Parameters.Add(prm4);

}

else

{

if (tboxVilContractID.Text == "")

{

MessageBox.Show("Заполните необходимые поля", "Пустое поле", MessageBoxButtons.OK);

return;

}

string sqlExp = "UpdateVillageContract";

command = new SqlCommand(sqlExp, connection);

command.CommandType = System.Data.CommandType.StoredProcedure; //хранимые процедуры

SqlParameter prm1 = new SqlParameter

{

ParameterName = "@idRespCitizen",

Value = Convert.ToInt32(tBoxVilIDResp.Text)

};

SqlParameter prm2 = new SqlParameter

{

ParameterName = "@ProductionOf",

Value = ProductionVil.SelectedItem.ToString()

};

SqlParameter prm3 = new SqlParameter

{

ParameterName = "@Weight",

Value = Convert.ToInt32(WeightVil.Value)

};

SqlParameter prm4 = new SqlParameter

{

ParameterName = "@Time",

Value = HelpfulClass.GetValidDateForSql(DateVil)

};

SqlParameter prm5 = new SqlParameter

{

ParameterName = "@id\_Contract",

Value = Convert.ToInt32(tboxVilContractID.Text)

};

command.Parameters.Add(prm1);

command.Parameters.Add(prm2);

command.Parameters.Add(prm3);

command.Parameters.Add(prm4);

command.Parameters.Add(prm5);

}

command.ExecuteNonQuery();

HelpfulClass.UpdateTable("Village\_Contracts", connection, DataGridForContracts);

connection.Close();

}

}

private void radioUpdateVilContract\_CheckedChanged(object sender, EventArgs e)

{

if (radioUpdateVilContract.Checked)

{

label7.Visible = true;

tboxVilContractID.Visible = true;

groupVilContractAddUp.Text = "Обновление контракта";

AddUpContractVil.Text = "Обновить контракт";

}

}

private void radioAddVilContract\_CheckedChanged(object sender, EventArgs e)

{

if (radioVillageContract.Checked)

{

label7.Visible = false;

tboxVilContractID.Visible = false;

groupVilContractAddUp.Text = "Добавление контракта";

AddUpContractVil.Text = "Добавить контракт";

}

}

private void AddUpGlobal\_Click(object sender, EventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectString))

{

connection.Open();

if ((IdRespGlobal.Text == "")|| ("" == OrganizationGlobal.Text) || ("" == AdressGlobal.Text) || ("" == priceGlobal.Text)){

MessageBox.Show("Заполните необходимые поля", "Пустое поле", MessageBoxButtons.OK);

return;

}

SqlCommand command;

if (radioAddGlobal.Checked)

{

string sqlExp = "AddGlobalContract";

command = new SqlCommand(sqlExp, connection);

command.CommandType = System.Data.CommandType.StoredProcedure; //хранимые процедуры

SqlParameter prm1 = new SqlParameter

{

ParameterName = "@idRespCitizen",

Value = Convert.ToInt32(IdRespGlobal.Text)

};

SqlParameter prm2 = new SqlParameter

{

ParameterName = "@ProductionOf",

Value = TypeGlobal.SelectedItem.ToString()

};

SqlParameter prm3 = new SqlParameter

{

ParameterName = "@Weight",

Value = Convert.ToInt32(WeightGlobal.Value)

};

SqlParameter prm4 = new SqlParameter

{

ParameterName = "@Time",

Value = HelpfulClass.GetValidDateForSql(dateGlobal)

};

SqlParameter prm5 = new SqlParameter

{

ParameterName = "@Organ",

Value = OrganizationGlobal.Text

};

SqlParameter prm6 = new SqlParameter

{

ParameterName = "@Address",

Value = AdressGlobal.Text

};

SqlParameter prm7 = new SqlParameter

{

ParameterName = "@Price",

Value = Convert.ToDecimal(priceGlobal.Text)

};

command.Parameters.Add(prm1);

command.Parameters.Add(prm2);

command.Parameters.Add(prm3);

command.Parameters.Add(prm4);

command.Parameters.Add(prm5);

command.Parameters.Add(prm6);

command.Parameters.Add(prm7);

}

else

{

if (IdContrGlobal.Text == "")

{

MessageBox.Show("Заполните необходимые поля", "Пустое поле", MessageBoxButtons.OK);

return;

}

string sqlExp = "UpdateGlobalContract";

command = new SqlCommand(sqlExp, connection);

command.CommandType = System.Data.CommandType.StoredProcedure; //хранимые процедуры

SqlParameter prm1 = new SqlParameter

{

ParameterName = "@idRespCitizen",

Value = Convert.ToInt32(IdRespGlobal.Text)

};

SqlParameter prm2 = new SqlParameter

{

ParameterName = "@ProductionOf",

Value = TypeGlobal.SelectedItem.ToString()

};

SqlParameter prm3 = new SqlParameter

{

ParameterName = "@Weight",

Value = Convert.ToInt32(WeightGlobal.Value)

};

SqlParameter prm4 = new SqlParameter

{

ParameterName = "@Time",

Value = HelpfulClass.GetValidDateForSql(dateGlobal)

};

SqlParameter prm5 = new SqlParameter

{

ParameterName = "@Organ",

Value = OrganizationGlobal.Text

};

SqlParameter prm6 = new SqlParameter

{

ParameterName = "@Address",

Value = AdressGlobal.Text

};

SqlParameter prm7 = new SqlParameter

{

ParameterName = "@Price",

Value = Convert.ToDecimal(priceGlobal.Text)

};

SqlParameter prm8 = new SqlParameter

{

ParameterName = "@id\_contract",

Value = Convert.ToInt32(IdContrGlobal.Text)

};

command.Parameters.Add(prm1);

command.Parameters.Add(prm2);

command.Parameters.Add(prm3);

command.Parameters.Add(prm4);

command.Parameters.Add(prm5);

command.Parameters.Add(prm6);

command.Parameters.Add(prm7);

command.Parameters.Add(prm8);

}

command.ExecuteNonQuery();

HelpfulClass.UpdateTable("Global\_Contracts", connection, DataGridForContracts);

connection.Close();

}

}

private void radioAddGlobal\_CheckedChanged(object sender, EventArgs e)

{

if (radioAddGlobal.Checked)

{

label4.Visible = false;

IdContrGlobal.Visible = false;

AddUpGlobal.Text = "Добавление контракта";

groupGlobalBox.Text = "Добавить контракт";

}

}

private void radioUpdGlobal\_CheckedChanged(object sender, EventArgs e)

{

if (radioUpdGlobal.Checked)

{

label4.Visible = true;

IdContrGlobal.Visible = true;

AddUpGlobal.Text = "Обновление контракта";

groupGlobalBox.Text = "Обновить контракт";

}

}

private void AddUpTur\_Click(object sender, EventArgs e)

{

if ((idHouseTur.Text=="")||(""==NameTur.Text)||(LastnameTur.Text=="")||(MiddleNameTur.Text=="")||(TurAddress.Text=="")){

MessageBox.Show("Не оставляйте поля пустыми", "Внимание", MessageBoxButtons.OK);

return;

}

using (SqlConnection connection = new SqlConnection(connectString))

{

connection.Open();

SqlCommand command;

if (radioAddTurContr.Checked)

{

string sqlExp = "AddTurismContract";

command = new SqlCommand(sqlExp, connection);

command.CommandType = System.Data.CommandType.StoredProcedure; //хранимые процедуры

SqlParameter prm1 = new SqlParameter

{

ParameterName = "@idHouse",

Value = Convert.ToInt32(idHouseTur.Text)

};

SqlParameter prm2 = new SqlParameter

{

ParameterName = "@Name",

Value = NameTur.Text

};

SqlParameter prm3 = new SqlParameter

{

ParameterName = "@LName",

Value = LastnameTur.Text

};

SqlParameter prm4 = new SqlParameter

{

ParameterName = "@MName",

Value = MiddleNameTur.Text

};

string sx = "";

if (radioSexTurMale.Checked) { sx = "Мужчина"; }

else { sx = "Женщина"; }

SqlParameter prm5 = new SqlParameter

{

ParameterName = "@Sex",

Value = sx

};

SqlParameter prm6 = new SqlParameter

{

ParameterName = "@DateOfBirth",

Value = HelpfulClass.GetValidDateForSql(DateOfBirthTur)

};

SqlParameter prm7 = new SqlParameter

{

ParameterName = "@Begin",

Value = HelpfulClass.GetValidDateForSql(dateBegTur)

};

SqlParameter prm8 = new SqlParameter

{

ParameterName = "@End",

Value = HelpfulClass.GetValidDateForSql(dateEndTur)

};

SqlParameter prm9 = new SqlParameter

{

ParameterName = "@Count",

Value = CountTur.Value

};

SqlParameter prm10 = new SqlParameter

{

ParameterName = "@Address",

Value = TurAddress.Text

};

command.Parameters.Add(prm1);

command.Parameters.Add(prm2);

command.Parameters.Add(prm3);

command.Parameters.Add(prm4);

command.Parameters.Add(prm5);

command.Parameters.Add(prm6);

command.Parameters.Add(prm7);

command.Parameters.Add(prm8);

command.Parameters.Add(prm9);

command.Parameters.Add(prm10);

}

else

{

if (""== idTurContract.Text)

{

MessageBox.Show("Не оставляйте поля пустыми", "Внимание", MessageBoxButtons.OK);

return;

}

string sqlExp = "UpdateTurismContract";

command = new SqlCommand(sqlExp, connection);

command.CommandType = System.Data.CommandType.StoredProcedure; //хранимые процедуры

SqlParameter prm1 = new SqlParameter

{

ParameterName = "@idHouse",

Value = Convert.ToInt32(idHouseTur.Text)

};

SqlParameter prm2 = new SqlParameter

{

ParameterName = "@Name",

Value = NameTur.Text

};

SqlParameter prm3 = new SqlParameter

{

ParameterName = "@LName",

Value = LastnameTur.Text

};

SqlParameter prm4 = new SqlParameter

{

ParameterName = "@MName",

Value = MiddleNameTur.Text

};

string sx = "";

if (radioSexTurMale.Checked) { sx = "Мужчина"; }

else { sx = "Женщина"; }

SqlParameter prm5 = new SqlParameter

{

ParameterName = "@Sex",

Value = sx

};

SqlParameter prm6 = new SqlParameter

{

ParameterName = "@DateOfBirth",

Value = HelpfulClass.GetValidDateForSql(DateOfBirthTur)

};

SqlParameter prm7 = new SqlParameter

{

ParameterName = "@Begin",

Value = HelpfulClass.GetValidDateForSql(dateBegTur)

};

SqlParameter prm8 = new SqlParameter

{

ParameterName = "@End",

Value = HelpfulClass.GetValidDateForSql(dateEndTur)

};

SqlParameter prm9 = new SqlParameter

{

ParameterName = "@Count",

Value = CountTur.Value

};

SqlParameter prm10 = new SqlParameter

{

ParameterName = "@Address",

Value = TurAddress.Text

};

SqlParameter prm11 = new SqlParameter

{

ParameterName = "@idContr",

Value = Convert.ToInt32(idTurContract.Text)

};

command.Parameters.Add(prm1);

command.Parameters.Add(prm2);

command.Parameters.Add(prm3);

command.Parameters.Add(prm4);

command.Parameters.Add(prm5);

command.Parameters.Add(prm6);

command.Parameters.Add(prm7);

command.Parameters.Add(prm8);

command.Parameters.Add(prm9);

command.Parameters.Add(prm10);

command.Parameters.Add(prm11);

}

command.ExecuteNonQuery();

HelpfulClass.UpdateTable("Turism\_Contracts", connection, DataGridForContracts);

connection.Close();

}

}

private void radioUpTurContr\_CheckedChanged(object sender, EventArgs e)

{

if (radioUpTurContr.Checked)

{

idTurContract.Visible = true;

labelForIDTur.Visible = true;

groupAddUpTurBox.Text = "Обновление контракта";

AddUpTur.Text = "Обновление контракта";

}

}

private void radioAddTurContr\_CheckedChanged(object sender, EventArgs e)

{

if (radioAddTurContr.Checked)

{

idTurContract.Visible = false;

labelForIDTur.Visible = false;

groupAddUpTurBox.Text = "Добавление контракта";

AddUpTur.Text = "Добавление контракта";

}

}

private void BackButton\_Click(object sender, EventArgs e)

{

this.Close();

}

private void deleteHouse\_Click(object sender, EventArgs e)

{

if (idDeleted.Text == String.Empty)

{

MessageBox.Show("Заполняйте поля корректно");

return;

}

using (SqlConnection connection = new SqlConnection(connectString))

{

connection.Open();

string sqlExp = "Delete";

string table = "";

if (radioGlobalContracts.Checked)

{

sqlExp += "GlobalContract";

table = "Global\_Contracts";

}

if (radioVillageContract.Checked)

{

sqlExp += "VillageContract";

table = "Village\_Contracts";

}

if (radioTurismContract.Checked)

{

sqlExp += "TurismContract";

table = "Turism\_Contracts";

}

SqlCommand command = new SqlCommand(sqlExp, connection);

command.CommandType = System.Data.CommandType.StoredProcedure; //хранимые процедуры

SqlParameter id = new SqlParameter

{

ParameterName = "@id",

Value = Convert.ToInt32(idDeleted.Text)

};

command.Parameters.Add(id);

command.ExecuteNonQuery();

HelpfulClass.UpdateTable(table, connection, DataGridForContracts);

}

}

private void IdContrGlobal\_KeyPress(object sender, KeyPressEventArgs e)

{

var symb = e.KeyChar;

if (!Char.IsDigit(symb) && !Char.IsControl(symb))

{

e.Handled = true;

}

}

private void IdRespGlobal\_KeyPress(object sender, KeyPressEventArgs e)

{

var symb = e.KeyChar;

if (!Char.IsDigit(symb) && !Char.IsControl(symb))

{

e.Handled = true;

}

}

private void idTurContract\_KeyPress(object sender, KeyPressEventArgs e)

{

var symb = e.KeyChar;

if (!Char.IsDigit(symb) && !Char.IsControl(symb))

{

e.Handled = true;

}

}

private void idHouseTur\_KeyPress(object sender, KeyPressEventArgs e)

{

var symb = e.KeyChar;

if (!Char.IsDigit(symb) && !Char.IsControl(symb))

{

e.Handled = true;

}

}

private void idDeleted\_KeyPress(object sender, KeyPressEventArgs e)

{

var symb = e.KeyChar;

if (!Char.IsDigit(symb) && !Char.IsControl(symb))

{

e.Handled = true;

}

}

private void tboxVilContractID\_KeyPress(object sender, KeyPressEventArgs e)

{

var symb = e.KeyChar;

if (!Char.IsDigit(symb) && !Char.IsControl(symb))

{

e.Handled = true;

}

}

}

**}**  
**Файл “Houses.cs”**using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace OuurWork

{

public partial class Houses : Form

{

string connectString = "Data Source=RRESHRRPC;Initial Catalog=OurWork;Integrated Security = True";

public Houses(bool admin)

{

InitializeComponent();

if (admin)

{

radioAdd.Checked = true;

labelIdHouse.Visible = false;

IdHouse.Visible = false;

}

radioVip.Checked = true;

using (SqlConnection connection = new SqlConnection(connectString))

{

connection.Open();

HelpfulClass.UpdateTable("Houses", connection, housesDataGridView);

connection.Close();

}

if (!admin)

{

radioUp.Checked = true;

radioAdd.Enabled = false;

labelIdHouse.Visible = true;

IdHouse.Visible =true;

DeleteBoxAdmin.Visible = false;

}

}

private void label6\_Click(object sender, EventArgs e)

{

}

private void AddUpHouse\_Click(object sender, EventArgs e)

{

if ((String.Empty == price.Text)|| (String.Empty == IdResp.Text)){

MessageBox.Show("Заполните поля корректно", "Прошу", MessageBoxButtons.OK);

return;

}

using (SqlConnection connection = new SqlConnection(connectString))

{

connection.Open();

if (radioAdd.Checked)

{

string sqlExp = "AddHouse";

SqlCommand command = new SqlCommand(sqlExp, connection);

command.CommandType = System.Data.CommandType.StoredProcedure; //хранимые процедуры

SqlParameter prm1 = new SqlParameter

{

ParameterName = "@idResp",

Value = Convert.ToInt32(IdResp.Text)

};

SqlParameter prm2 = new SqlParameter

{

ParameterName = "@CountRoom",

Value = Count.Value

};

SqlParameter prm3 = new SqlParameter

{

ParameterName = "@PriceDay",

Value = Convert.ToDecimal(price.Text)

};

string s = "";

if (radioVip.Checked)

{

s = "Да";

}

else

{

s = "Нет";

}

SqlParameter prm4 = new SqlParameter

{

ParameterName = "@Vip",

Value = s

};

command.Parameters.Add(prm1);

command.Parameters.Add(prm2);

command.Parameters.Add(prm3);

command.Parameters.Add(prm4);

command.ExecuteNonQuery();

}

else

{

if (String.Empty == IdHouse.Text){

MessageBox.Show("Заполните поля корректно", "Прошу", MessageBoxButtons.OK);

return;

}

string sqlExp = "UpdateHouse";

SqlCommand command = new SqlCommand(sqlExp, connection);

command.CommandType = System.Data.CommandType.StoredProcedure; //хранимые процедуры

SqlParameter prm1 = new SqlParameter

{

ParameterName = "@idResp",

Value = Convert.ToInt32(IdResp.Text)

};

SqlParameter prm2 = new SqlParameter

{

ParameterName = "@CountRoom",

Value = Count.Value

};

SqlParameter prm3 = new SqlParameter

{

ParameterName = "@PriceDay",

Value = Convert.ToDecimal(price.Text)

};

string s = "";

if (radioVip.Checked)

{

s = "Да";

}

else

{

s = "Нет";

}

SqlParameter prm4 = new SqlParameter

{

ParameterName = "@Vip",

Value = s

};

SqlParameter prm5 = new SqlParameter

{

ParameterName = "@id",

Value = Convert.ToInt32(IdHouse.Text)

};

command.Parameters.Add(prm1);

command.Parameters.Add(prm2);

command.Parameters.Add(prm3);

command.Parameters.Add(prm4);

command.Parameters.Add(prm5);

command.ExecuteNonQuery();

}

HelpfulClass.UpdateTable("Houses", connection, housesDataGridView);

connection.Close();

}

}

private void radioUp\_CheckedChanged(object sender, EventArgs e)

{

if (radioUp.Checked)

{

labelIdHouse.Visible = true;

IdHouse.Visible = true;

groupInterface.Text = "Обновление дома";

AddUpHouse.Text = "Обновить инфо о доме";

}

}

private void radioAdd\_CheckedChanged(object sender, EventArgs e)

{

if (radioAdd.Checked)

{

groupInterface.Text = "Добавление дома";

AddUpHouse.Text = "Добавить дом";

labelIdHouse.Visible = false;

IdHouse.Visible = false;

}

}

private void deleteHouse\_Click(object sender, EventArgs e)

{

if (idDeleted.Text == String.Empty)

{

MessageBox.Show("Заполните поля корректно");

return;

}

using (SqlConnection connection = new SqlConnection(connectString))

{

connection.Open();

string sqlExp = "DeleteHouse";

SqlCommand command = new SqlCommand(sqlExp, connection);

command.CommandType = System.Data.CommandType.StoredProcedure; //хранимые процедуры

SqlParameter id = new SqlParameter

{

ParameterName = "@id",

Value = Convert.ToInt32(idDeleted.Text)

};

command.Parameters.Add(id);

command.ExecuteNonQuery();

HelpfulClass.UpdateTable("Houses", connection, housesDataGridView);

connection.Close();

}

}

private void BackButton\_Click(object sender, EventArgs e)

{

this.Close();

}

private void IdHouse\_KeyPress(object sender, KeyPressEventArgs e)

{

var symb = e.KeyChar;

if (!Char.IsDigit(symb) && !Char.IsControl(symb) )

{

e.Handled = true;

}

}

private void IdResp\_KeyPress(object sender, KeyPressEventArgs e)

{

var symb = e.KeyChar;

if (!Char.IsDigit(symb) && !Char.IsControl(symb) )

{

e.Handled = true;

}

}

private void idDeleted\_KeyPress(object sender, KeyPressEventArgs e)

{

var symb = e.KeyChar;

if (!Char.IsDigit(symb) && !Char.IsControl(symb) )

{

e.Handled = true;

}

}

private void price\_TextChanged(object sender, EventArgs e)

{

}

private void price\_KeyPress(object sender, KeyPressEventArgs e)

{

var symb = e.KeyChar;

if (!Char.IsDigit(symb) && !Char.IsControl(symb) && symb != ',')

{

e.Handled = true;

}

}

}

}

**Файл «Souvenir.cs»**using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace OuurWork

{

public partial class Souvenir : Form

{

string connectString = "Data Source=RRESHRRPC;Initial Catalog=OurWork;Integrated Security = True";

public Souvenir(bool admin)

{

InitializeComponent();

labelIdSouvenir.Visible = false;

IdSouvenir.Visible = false;

radioAdd.Checked = true;

groupInterface.Text = "Добавление сувенира";

AddUpSouvenir.Text = "Добавить сувенир";

if (!admin)

{

radioUp.Enabled = false;

DeleteBoxAdmin.Visible = false;

}

}

private void souvenirBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.souvenirBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.ourWorkDataSet);

}

private void Souvenir\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "ourWorkDataSet.Souvenir". При необходимости она может быть перемещена или удалена.

this.souvenirTableAdapter.Fill(this.ourWorkDataSet.Souvenir);

}

private void AddUpHouse\_Click(object sender, EventArgs e)

{

if ((String.Empty==IdResp.Text)|| (String.Empty == price.Text)|| (String.Empty == nameSouvenir.Text))

{

MessageBox.Show("Заполните поля корректно", "Пожайлуста", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

if (radioAdd.Checked)

{

using (SqlConnection connection = new SqlConnection(connectString))

{

connection.Open();

string command = "INSERT INTO Souvenir(id\_Producer, Price, Name\_of) VALUES (";

command += Convert.ToInt32(IdResp.Text) + ", " + (Convert.ToDecimal(price.Text)).ToString().Replace(",", ".") + ", " + '\'' + nameSouvenir.Text + '\'' + ");";

SqlDataAdapter SDA = new SqlDataAdapter(command, connection);

DataSet DS = new DataSet();

SDA.Fill(DS, "Souvenir");

souvenirDataGridView.DataSource = DS.Tables["Souvenir"];

HelpfulClass.UpdateTable("Souvenir", connection, souvenirDataGridView);

}

}

else

{

if (String.Empty == IdSouvenir.Text)

{

MessageBox.Show("Заполните поля корректно", "Прошу", MessageBoxButtons.OK);

return;

}

using (SqlConnection connection = new SqlConnection(connectString))

{

connection.Open();

string command = "UPDATE Souvenir SET ";

command += "Id\_Producer=" + Convert.ToInt32(IdResp.Text) + ", Price=" + (Convert.ToDecimal(price.Text)).ToString().Replace(",", ".")

+ ",Name\_of= " + '\'' + nameSouvenir.Text + '\'' +

"WHERE id\_souvenir=" + Convert.ToInt32(IdSouvenir.Text) + ";";

SqlDataAdapter SDA = new SqlDataAdapter(command, connection);

DataSet DS = new DataSet();

SDA.Fill(DS, "Souvenir");

souvenirDataGridView.DataSource = DS.Tables["Souvenir"];

HelpfulClass.UpdateTable("Souvenir", connection, souvenirDataGridView);

}

}

}

private void deleteHouse\_Click(object sender, EventArgs e)

{

if (idDeleted.Text == String.Empty)

{

MessageBox.Show("Заполните поля корректно", "Умоляю", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

using (SqlConnection connection = new SqlConnection(connectString))

{

connection.Open();

string command = "DELETE FROM Souvenir WHERE id\_souvenir=" + Convert.ToInt32(idDeleted.Text);

SqlDataAdapter SDA = new SqlDataAdapter(command, connection);

DataSet DS = new DataSet();

SDA.Fill(DS, "Souvenir");

souvenirDataGridView.DataSource = DS.Tables["Souvenir"];

HelpfulClass.UpdateTable("Souvenir", connection, souvenirDataGridView);

}

}

private void radioAdd\_CheckedChanged(object sender, EventArgs e)

{

if (radioAdd.Checked)

{

labelIdSouvenir.Visible = false;

IdSouvenir.Visible = false;

groupInterface.Text = "Добавление сувенира";

AddUpSouvenir.Text = "Добавить сувенир";

}

}

private void radioUp\_CheckedChanged(object sender, EventArgs e)

{

if (radioUp.Checked)

{

labelIdSouvenir.Visible = true;

IdSouvenir.Visible = true;

groupInterface.Text = "Обновление сувенира";

AddUpSouvenir.Text = "Обновить сувенир";

}

}

private void BackButton\_Click(object sender, EventArgs e)

{

this.Close();

}

private void idDeleted\_KeyPress(object sender, KeyPressEventArgs e)

{

var symb = e.KeyChar;

if (!Char.IsDigit(symb) && !Char.IsControl(symb))

{

e.Handled = true;

}

}

private void IdSouvenir\_TextChanged(object sender, EventArgs e)

{

}

private void IdResp\_TextChanged(object sender, EventArgs e)

{

}

private void IdResp\_KeyPress(object sender, KeyPressEventArgs e)

{

var symb = e.KeyChar;

if (!Char.IsDigit(symb) && !Char.IsControl(symb))

{

e.Handled = true;

}

}

private void IdSouvenir\_KeyPress(object sender, KeyPressEventArgs e)

{

var symb = e.KeyChar;

if (!Char.IsDigit(symb) && !Char.IsControl(symb))

{

e.Handled = true;

}

}

private void price\_KeyPress(object sender, KeyPressEventArgs e)

{

var symb = e.KeyChar;

if (!Char.IsDigit(symb) && !Char.IsControl(symb) && symb != ',')

{

e.Handled = true;

}

}

}

}

**Файл «All\_info.cs»**using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace OuurWork

{

public partial class All\_Info : Form

{

string connectString = "Data Source=RRESHRRPC;Initial Catalog=OurWork;Integrated Security = True";

public All\_Info(string info)

{

InitializeComponent();

if (info == "admin")

{

using (SqlConnection connection = new SqlConnection(connectString))

{

connection.Open();

string command = "SELECT \* FROM Profiles";

SqlDataAdapter SDA = new SqlDataAdapter(command, connection);

DataSet DS = new DataSet();

SDA.Fill(DS, "Profiles");

dataGridView.DataSource = DS.Tables["Profiles"];

HelpfulClass.UpdateTable("Profiles", connection, dataGridView);

}

} else if (info=="citizen")

{

using (SqlConnection connection = new SqlConnection(connectString))

{

connection.Open();

string command = "SELECT \* FROM Profile";

SqlDataAdapter SDA = new SqlDataAdapter(command, connection);

DataSet DS = new DataSet();

SDA.Fill(DS, "Profile");

dataGridView.DataSource = DS.Tables["Profile"];

HelpfulClass.UpdateTable("Profile", connection, dataGridView);

}

}

}

private void Back\_Click(object sender, EventArgs e)

{

this.Close();

}

private void citizensProfilesDataGridView\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

}

}

**Файл «Helpfulclass.cs»**using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace OuurWork

{

class HelpfulClass

{

public static void UpdateTable(string nameTable, SqlConnection connection, System.Windows.Forms.DataGridView DGV)

{

SqlDataAdapter SDA = new SqlDataAdapter("SELECT \* FROM " + nameTable, connection);

DataSet DS = new DataSet();

SDA.Fill(DS, nameTable);

DGV.DataSource = DS.Tables[nameTable];

}

public static string GetValidDateForSql(System.Windows.Forms.DateTimePicker date)

{

string res = date.Value.Year.ToString();

if (date.Value.Month < 10)

{

res += '0' + date.Value.Month.ToString();

}

else

{

res += date.Value.Month.ToString();

}

if (date.Value.Day < 10)

{

res += '0' + date.Value.Day.ToString();

}

else

{

res += date.Value.Day.ToString();

}

return res;

}

}

}

**Код MS SQL**

use master;

go

create database OurWork

ON (

name ='OurWork',

filename = 'D:\Учёба, год третий\ourwork.mdf',

size =5MB,

maxsize=25MB,

filegrowth = 10%)

log on (

name ='OurWork\_log',

filename = 'D:\Учёба, год третий\ourwork\_log.ldf',

size =5MB,

maxsize=15MB,

filegrowth = 3MB);

go

use OurWork;

go

CREATE TABLE Citizens(

id\_citizen int not null primary key identity(1,1),

Name nvarchar(60) not null,

Last\_Name nvarchar(60) not null,

Middle\_Name nvarchar(60),

Sex nvarchar(7) not null,

Date\_of\_birth Date not null CHECK (datediff(day, Date\_of\_birth, getdate()) >= 16 \* 365)

)

go

CREATE TABLE Village\_Contracts(

id\_Contract int not null primary key identity(1,1),

id\_Resp\_Citizen int NOT NULL FOREIGN KEY REFERENCES Citizens(id\_citizen) ON DELETE CASCADE,

ProductionOf nvarchar(24) not null,

Weight\_of\_All int not null,

Time Date not null,

)

go

CREATE TABLE Global\_Contracts(

id\_contract int not null primary key identity(1,1),

id\_Resp\_Citizen int FOREIGN KEY REFERENCES Citizens(id\_citizen) ON DELETE CASCADE,

Organization nvarchar(30) not null,

Address nvarchar(60) not null,

Weight\_Need int not null check (Weight\_Need>=5),

Time\_Need Date not null,

Price money not null,

Production\_Need nvarchar(24) not null

)

go

CREATE TABLE Souvenir(

id\_Souvenir int not null primary key identity(1,1),

id\_Producer int NOT NULL FOREIGN KEY REFERENCES Citizens(id\_citizen),

Price money not null

)

go

CREATE TABLE Houses(

id\_house int not null primary key identity(1,1),

id\_Resp\_Citizen int NOT NULL FOREIGN KEY REFERENCES Citizens(id\_citizen) ON DELETE CASCADE,

Count\_Of\_Room int not null,

Price\_Per\_Day int not null,

VIP nvarchar(3) not null,

)

go

CREATE TABLE Turism\_Contracts(

id\_Turcontract int not null primary key identity(1,1),

id\_house int not null FOREIGN KEY REFERENCES Houses(id\_house),

Turist\_Name nvarchar(60) not null,

Turist\_Last\_Name nvarchar(60) not null,

Turist\_Middle\_Name nvarchar(60),

Turist\_Sex nvarchar(7) not null,

Turist\_Date\_of\_birth Date not null,

Begin\_Tur Date not null,

End\_Tur Date not null,

Count\_People int not null,

Home\_Address nvarchar(60) not null

)

use OurWork;

go

CREATE NONCLUSTERED INDEX HouseNumber ON Turism\_Contracts(id\_house);

go

CREATE NONCLUSTERED INDEX HotWater ON Houses(VIP);

go

CREATE NONCLUSTERED INDEX CountOfRoom on Houses(Count\_Of\_Room);

use OurWork;

go

CREATE FUNCTION Price\_Turism\_Contract(@id\_Tur\_Contract int) /\*скалярная функция, возвращающая сколько всего денег

было получено с контракта\*/

RETURNS INT

AS

BEGIN

DECLARE @IdHouse int;

SELECT @IdHouse=id\_house FROM Turism\_Contracts

WHERE (Turism\_Contracts.id\_Turcontract=@id\_Tur\_Contract)

DECLARE @Days int;

SELECT @Days=DATEDIFF(day, Begin\_Tur, End\_Tur) FROM Turism\_Contracts

WHERE (Turism\_Contracts.id\_Turcontract=@id\_Tur\_Contract);

DECLARE @PricePerDay int;

SELECT @PricePerDay=Price\_Per\_Day FROM Houses

WHERE (id\_house=@IdHouse);

RETURN @Days \* @PricePerDay;

END;

CREATE PROCEDURE AddCitizen(@Name nvarchar(60), @LastName nvarchar(60), @MiddleName nvarchar(60), @Sex nvarchar(7),

@Date\_of\_birth Date)

AS

INSERT INTO Citizens(Name, Middle\_Name, Last\_Name, Sex, Date\_of\_birth)

VALUES (@Name, @MiddleName, @LastName, @Sex, @Date\_of\_birth);

CREATE PROCEDURE DeleteCitizen(@id int)

AS

DELETE FROM Citizens

WHERE id\_citizen=@id;

CREATE PROCEDURE UpdateCitizen(@id int, @new\_name nvarchar(60),@new\_lname nvarchar(60),

@new\_mname nvarchar(60), @new\_sex nvarchar(7), @new\_born Date)

AS

UPDATE Citizens

SET Name =@new\_name, Last\_Name =@new\_lname, Middle\_Name =@new\_mname,

Sex = @new\_sex, Date\_of\_birth=@new\_born WHERE id\_citizen =@id;

Select \* from Citizens;

CREATE PROCEDURE AddVillageContract (@idRespCitizen int, @ProductionOf nvarchar(24),

@Weight int, @Time Date)

AS

INSERT INTO Village\_Contracts(id\_Resp\_Citizen, ProductionOf, Weight\_of\_All, Time)

VALUES (@idRespCitizen, @ProductionOf, @Weight, @Time);

CREATE PROCEDURE DeleteVillageContract(@id int)

AS

DELETE FROM Village\_Contracts

WHERE id\_Contract=@id;

CREATE PROCEDURE UpdateVillageContract (@id\_contract int, @idRespCitizen int,

@ProductionOf nvarchar(24), @Weight int, @Time Date)

AS

UPDATE Village\_Contracts

Set id\_Resp\_Citizen=@idRespCitizen,

ProductionOf=@ProductionOf, Weight\_of\_All=@Weight,

Time=@Time WHERE id\_Contract=@id\_contract;

CREATE PROCEDURE AddGlobalContract (@idRespCitizen int, @ProductionOf nvarchar(24),

@Weight int, @Time Date, @Organ nvarchar(30), @Address nvarchar(60),

@Price money)

AS

INSERT INTO Global\_Contracts(id\_Resp\_Citizen, Production\_Need, Weight\_Need, Time\_Need,

Organization, Address, Price)

VALUES (@idRespCitizen, @ProductionOf, @Weight, @Time, @Organ, @Address, @Price);

CREATE PROCEDURE DeleteGlobalContract(@id int)

AS

DELETE FROM Global\_Contracts

WHERE id\_Contract=@id;

CREATE PROCEDURE UpdateGlobalContract (@id\_contract int, @idRespCitizen int, @ProductionOf nvarchar(24),

@Weight int, @Time Date, @Organ nvarchar(30), @Address nvarchar(60),

@Price money)

AS

UPDATE Global\_Contracts

Set id\_Resp\_Citizen=@idRespCitizen,

Production\_Need=@ProductionOf, Weight\_Need=@Weight,

Time\_Need=@Time, Organization=@Organ, Address=@Address,

Price=@Price WHERE id\_Contract=@id\_contract;

CREATE PROCEDURE AddTurismContract (@idHouse int, @Name nvarchar(60),

@LName nvarchar(60), @MName nvarchar(60), @Sex nvarchar(7),

@DateOfBirth Date, @Begin Date, @End Date, @Count int, @Address nvarchar(60))

AS

INSERT INTO Turism\_Contracts(id\_house, Turist\_Name, Turist\_Last\_Name, Turist\_Middle\_Name,

Turist\_Sex, Turist\_Date\_of\_birth, Begin\_Tur, End\_Tur, Count\_People, Home\_Address)

VALUES (@idHouse, @Name,

@LName, @MName, @Sex,

@DateOfBirth, @Begin, @End, @Count, @Address);

CREATE PROCEDURE DeleteTurismContract(@id int)

AS

DELETE FROM Turism\_Contracts

WHERE id\_Turcontract=@id;

CREATE PROCEDURE UpdateTurismContract (@idContr int, @idHouse int, @Name nvarchar(60),

@LName nvarchar(60), @MName nvarchar(60), @Sex nvarchar(7),

@DateOfBirth Date, @Begin Date, @End Date, @Count int, @Address nvarchar(60))

AS

UPDATE Turism\_Contracts

Set id\_house=@idHouse, Turist\_Name=@Name,

Turist\_Last\_Name = @LName, Turist\_Middle\_Name = @MName,

Turist\_Sex=@Sex, Turist\_Date\_of\_birth=@DateOfBirth,

Begin\_Tur=@Begin, End\_Tur=@End, Count\_People=@Count,

Home\_Address=@Address WHERE id\_Turcontract=@idContr;

CREATE PROCEDURE AddHouse (@idResp int, @CountRoom int,

@PriceDay money, @VIP nvarchar(3))

AS

INSERT INTO Houses(id\_Resp\_Citizen, Count\_Of\_Room, Price\_Per\_Day, VIP)

VALUES (@idResp, @CountRoom, @PriceDay, @VIP);

CREATE PROCEDURE DeleteHouse(@id int)

AS

DELETE FROM Houses

WHERE id\_house=@id;

CREATE PROCEDURE UpdateHouse (@id int, @idResp int, @CountRoom int,

@PriceDay money, @VIP nvarchar(3))

AS

UPDATE Houses

Set id\_Resp\_Citizen=@idResp, Count\_Of\_Room=@CountRoom,

Price\_Per\_Day=@PriceDay, VIP=@VIP WHERE id\_house=@id;

CREATE Function Capital()

RETURNS money

BEGIN

DECLARE @resGlobal money

SELECT @resGlobal = SUM(Price) from Global\_Contracts WHERE Time\_Need < GETDATE();

DECLARE @resTurism money

SELECT @resTurism = SUM(dbo.Price\_Turism\_Contract(id\_Turcontract)) from Turism\_Contracts WHERE Begin\_Tur < GETDATE();

RETURN @resGlobal+@resTurism;

END;

CREATE FUNCTION TurContractInMonth (@month int)

RETURNS INT

AS

BEGIN

DECLARE @count int;

SELECT @count = COUNT(\*) FROM Turism\_Contracts WHERE MONTH(Begin\_Tur)=@month;

RETURN @count;

END;

GO

CREATE FUNCTION WhoWork(@Where nvarchar(24))

RETURNS @result table (FIO nvarchar(180))

AS

BEGIN

DECLARE @localTabl table(id INT);

INSERT INTO @localTabl(id)

SELECT Village\_Contracts.id\_Resp\_Citizen FROM Village\_Contracts WHERE

Village\_Contracts.ProductionOf=@Where;

INSERT INTO @result (FIO)

SELECT Citizens.Last\_Name + ' ' + left(Citizens.Name,1) + '. ' + LEFT(Citizens.Middle\_Name,1) + '. ' as 'ФИО'

FROM Citizens, @localTabl WHERE id\_citizen = id;

return;

END;

GO

CREATE FUNCTION CountAllContract(@id\_citizen int)

RETURNS INT

AS

BEGIN

DECLARE @CountVillage INT;

SELECT @CountVillage=COUNT(DISTINCT id\_Contract) FROM Village\_Contracts WHERE id\_Resp\_Citizen=@id\_citizen AND Village\_Contracts.Time > GETDATE();

DECLARE @CountGlobal INT;

SELECT @CountGlobal=COUNT(DISTINCT id\_Contract) FROM Global\_Contracts WHERE id\_Resp\_Citizen=@id\_citizen AND Global\_Contracts.Time\_Need>GETDATE();

DECLARE @CountTur INT;

SELECT @CountTur=COUNT(DISTINCT id\_TurContract) FROM (Houses INNER JOIN Turism\_Contracts ON Houses.id\_house=Turism\_Contracts.id\_house)

WHERE id\_Resp\_Citizen=@id\_citizen AND Begin\_Tur > GETDATE();

RETURN @CountTur + @CountGlobal + @CountVillage;

END;

GO

CREATE FUNCTION NearestGlobalDate(@id\_citizen INT)

RETURNS DATE

AS

BEGIN

DECLARE @minDate Date;

SELECT @minDate=Global\_Contracts.Time\_Need FROM Global\_Contracts WHERE (SELECT MIN(DATEDIFF(day, GETDATE(), Global\_Contracts.Time\_Need))

FROM Global\_Contracts WHERE Global\_Contracts.Time\_Need>GETDATE() AND Global\_Contracts.id\_Resp\_Citizen=@id\_citizen)=DATEDIFF(day, GETDATE(), Global\_Contracts.Time\_Need)

AND id\_Resp\_Citizen=@id\_citizen;

RETURN @minDate;

END;

GO

CREATE FUNCTION NearestVillageDate(@id\_citizen INT)

RETURNS DATE

AS

BEGIN

DECLARE @minDate Date;

SELECT @minDate=Village\_Contracts.Time FROM Village\_Contracts WHERE (SELECT MIN(DATEDIFF(day, GETDATE(), Village\_Contracts.Time))

FROM Village\_Contracts WHERE Village\_Contracts.Time>GETDATE() AND Village\_Contracts.id\_Resp\_Citizen=

@id\_citizen)=DATEDIFF(day, GETDATE(), Village\_Contracts.Time)

AND Village\_Contracts.Time>GETDATE() AND id\_Resp\_Citizen=@id\_citizen;

RETURN @minDate;

END;

GO

CREATE FUNCTION Price\_Turism\_Contract(@id\_Tur\_contract int) /\*скалярная функция, возвращающая сколько всего денег

было получено с контракта\*/

RETURNS INT

AS

BEGIN

DECLARE @IdHouse int;

SELECT @IdHouse=id\_house FROM Turism\_Contracts

WHERE (Turism\_Contracts.id\_Turcontract=@id\_Tur\_Contract)

DECLARE @Days int;

SELECT @Days=DATEDIFF(day, Begin\_Tur, End\_Tur) FROM Turism\_Contracts

WHERE (Turism\_Contracts.id\_Turcontract=@id\_Tur\_Contract);

DECLARE @PricePerDay money;

SELECT @PricePerDay = Price\_Per\_Day FROM Houses

WHERE (id\_house=@IdHouse);

RETURN @Days \* @PricePerDay;

END;

GO

EXEC sp\_configure 'clr enabled', 1;

reconfigure

alter database OurWork set trustworthy on;

create assembly MyOwnCLR

FROM

'D:/VS\_Codes/MyOwnCLR/MyOwnCLR/bin/Debug/MyOwnCLR.dll'

with permission\_set = unsafe;

create function MakeFIO (@fn nvarchar(60), @ln nvarchar(60), @mn nvarchar(60))

RETURNS nvarchar(70)

AS

EXTERNAL NAME MyOwnCLR.[MyOwnCLR.Class1].MakeFIO;

CREATE TRIGGER deleteVillage ON

Citizens

INSTEAD OF DELETE

AS

BEGIN

DECLARE @Birth Date, @id int;

SELECT @Birth = Date\_of\_birth, @id=id\_citizen from deleted;

IF (DATEDIFF(day, @Birth, GETDATE() ) < 20\*365)

begin

RAISERROR ('Данного жителя (ID=%d) удалить невозможно, он не отдал свой долг деревне', 16, 1, @id);

ROLLBACK;

END

ELSE

DELETE FROM Citizens WHERE id\_citizen=@id

END;

Create view Profiles

AS

Select id\_citizen AS 'ID', dbo.MakeFIO(Name, Last\_Name, Middle\_Name) AS 'ФИО',

dbo.NearestGlobalDate(id\_citizen) AS 'Ближайшее закрытие крупного контракта', dbo.NearestVillageDate(id\_citizen) AS 'Ближайший срок сдачи товара',

dbo.CountAllContract(id\_citizen) AS 'Всего открытых контрактов'

FROM Citizens

GO

Create view Profile

AS

Select id\_citizen AS 'ID', dbo.MakeFIO(Name, Last\_Name, Middle\_Name) AS 'ФИО',

dbo.NearestGlobalDate(id\_citizen) AS 'Ближайшее закрытие крупного контракта', dbo.NearestVillageDate(id\_citizen) AS 'Ближайший срок сдачи товара',

dbo.CountAllContract(id\_citizen) AS 'Всего открытых контрактов'

FROM Citizens WHERE id\_citizen=27

GO