МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

НАЦИОНАЛЬНЫЙ АЭРОКОСМИЧЕСКИЙ УНИВЕРСИТЕТ

ИМ. Н. Е. ЖУКОВСКОГО

«ХАРЬКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ»

ДОМАШНЯЯ РАБОТА

ВАРИАНТ №12

Выполнила: студентка группы 335а

Специальности 122 «Информатика»

Скицан Ольга

Проверил: ст. преподаватель каф. 304

Коробчинский К.П.

Харьков – 2018

**Цель работы:** Создать программу на языке С#, используя уже созданную базу данных.

**Задание:**

На автопредприятии имеются машины с известными государственными номерами - все грузовые, с одинаковым бензином и перечнем запчастей. Водители совершают рейсы, о которых нужно знать: откуда и куда выполнялся рейс, какой груз, дату отправления в рейс и прибытия назад. Есть рейсы, в которые выезжают два водителя. Водители ездят на любой машине. Если в рейсе случилась поломка, то водитель должен сообщить о ее типе, после чего со склада выдается стандартный набор запчастей для устранения поломки. Все запчасти имеют фиксированную цену. Набор запчастей для ремонта зависит от вида поломки. В одном рейсе может быть более одной поломки.

Запросы:

* Получить список рейсов в указанном диапазоне дат;
* Найти, сколько дней в рейсах провел каждый водитель, а также водителей, которые провели в рейсах максимальное количество дней, а также водителей, в рейсах которых никогда не случалась поломка заданного вида;
* Подсчитать, в какую сумму денег обошелся ремонт поломок каждого вида, и найти тот вид поломок, который встречается наиболее часто;
* Подсчитать среднее количество поломок заданного вида в расчете на один рейс (суммарное количество поломок заданного типа разделить на общее количество рейсов);
* Найти, на каких машинах заданный вид поломки никогда не случался.

Транзакции:

* + - Выполнить новый рейс, учитывая сведения о поломках;
* Купить новую машину.

**База данных**

CREATE TABLE Reis (

id\_reisa int IDENTITY Primary key,

vremya\_pribivaniya DATETIME NOT NULL,

vremya\_otpravleniya DATETIME NULL,

)

CREATE TABLE Voditeli (

id\_Voditelya int IDENTITY Primary key,

FIO nVARCHAR(40) NOT NULL,

Telefon VARCHAR(15) NULL,

Adress nVARCHAR(25) NULL,

)

CREATE TABLE Gruz (

id\_gruza int IDENTITY Primary key,

id\_reisa int NOT NULL FOREIGN KEY REFERENCES Reis(id\_reisa),

id\_Voditelya int NOT NULL FOREIGN KEY REFERENCES Voditeli(id\_Voditelya),

kol\_vo\_gruza int NOT NULL,

)

CREATE TABLE Tehnika (

id\_mashin int IDENTITY Primary key,

model nVARCHAR(30) NOT NULL,

id\_Voditelya int NOT NULL FOREIGN KEY REFERENCES Voditeli(id\_Voditelya),

)

CREATE TABLE Jobs (

id\_Jobs int IDENTITY Primary key,

FIO\_Jobs nVARCHAR(50) NOT NULL,

Kontaktnie\_dannie nVARCHAR(15) NOT NULL,

)

CREATE TABLE Polomka (

id\_polomki int IDENTITY Primary key,

id\_mashin int NOT NULL FOREIGN KEY REFERENCES Tehnika(id\_mashin),

id\_jobs int NOT NULL FOREIGN KEY REFERENCES Jobs(id\_Jobs),

vremya\_polomki DATETIME NULL

)

CREATE TABLE Detali (

id\_detali int IDENTITY Primary key,

id\_polomki int NOT NULL FOREIGN KEY REFERENCES Polomka(id\_polomki),

id\_mashin int NOT NULL FOREIGN KEY REFERENCES Tehnika(id\_mashin),

kol\_vo\_detaley int NOT NULL

)

CREATE TABLE Dostavka (

id\_Dostavka int IDENTITY Primary key,

id\_gruza int NOT NULL FOREIGN KEY REFERENCES Gruz(id\_gruza),

id\_mashin int NOT NULL FOREIGN KEY REFERENCES Tehnika(id\_mashin),

vremya\_pribitiya DATETIME NOT NULL

)

***Заполнение таблиц:***

INSERT INTO Reis

Values

('2018-09-14 11:20:13','2018-09-14 22:34:11'),

('2018-09-11 22:30:56','2018-09-11 10:20:20'),

('2018-09-22 15:14:56','2018-09-22 09:00:00'),

('2018-09-29 23:55:33','2018-09-29 19:05:20'),

('2018-09-05 09:30:44','2018-09-05 01:45:00'),

('2018-10-16 09:17:00','2018-10-16 12:00:10')

go

INSERT INTO Voditeli

Values

('Иванов Иван Иванович','+380960252271','Чкалова 14'),

('Смирнов Алексей Николаевич','+380931333333','Чернышевского 125'),

('Васильев Валерий Петрович','+380660555551','Дружбы Народов 116'),

('Морозов Николай Николаевич','+380999999999','Сумская 56'),

('Николаев Егор Егорович','+380559110345','Садовая 67'),

('Алексеев Семен Александрович','+380960555678','Центральная 45')

go

INSERT INTO Gruz

Values

(13,1,5),

(14,2,1),

(15,3,10),

(16,4,1),

(17,5,1),

(18,6,1)

go

INSERT INTO Tehnika

Values

('Henschel 33D1',1),

('Урал 375Д',2),

('Toyota Grand Hiace',3),

('Toyota Dyna 150',4),

('Nissan Cabstar',5),

('Renault Magnum',6)

go

INSERT INTO Jobs

Values

('Калилин Игорь Викторович','+380950345678'),

('Маслов Прохор Петрович','+380994234556'),

('Абрамов Андрей Алексеевич','+380678567223'),

('Щербаков Никита Сергеевич','+380950567233'),

('Данилов Андрей Андреевич','+380930567890'),

('Овчинников Дмитрий Егорович','+380930777888')

Go

INSERT INTO Polomka

Values

(1,1,NULL),

(2,2,'2018-09-11 18:23:56'),

(3,3,NULL),

(4,4,NULL),

(5,5,NULL),

(6,6,'2018-10-16 09:17:00')

Go

INSERT INTO Detali

Values

(1,1,5),

(2,2,1),

(3,3,1),

(4,4,11),

(5,5,2),

(6,6,5)

go

INSERT INTO Dostavka

Values

(2,1,'2018-09-14 11:20:13'),

(3,2,'2018-09-11 22:30:56'),

(4,3,'2018-09-22 15:14:56'),

(5,5,'2018-09-29 23:55:33'),

(6,5,'2018-09-05 09:30:44'),

(7,6,'2018-10-16 09:17:00')

go

***Процедуры и функции созданные для заданных в задании запросах:***

1. Получить список рейсов в указанном диапазоне дат

Create PROCEDURE Zadanie111

@PayDay1 DATETIME,@PayDay2 DATETIME

AS

BEGIN

SELECT Gruz.id\_reisa

FROM Reis

INNER JOIN Gruz ON Gruz.id\_reisa=Reis.id\_reisa

INNER JOIN Voditeli ON Voditeli.id\_Voditelya=Gruz.id\_Voditelya

Where Reis.vremya\_otpravleniya between @PayDay1 and @PayDay2

GO

EXEC Zadanie111 '2017-01-01 00:00:00','2018-10-30 00:00:00'

1. Найти, сколько дней в рейсах провел каждый водитель, а также водителей, которые провели в рейсах максимальное количество дней, а также водителей, в рейсах которых никогда не случалась поломка заданного вида

* Найти, сколько дней в рейсах провел каждый водитель

Сreate proc Proc1

AS

BEGIN

SELECT DATEDIFF(day,Reis.vremya\_otpravleniya,Reis.vremya\_pribivaniya) AS 'DURATION\_DAY', Voditeli.FIO

FROM Reis

INNER JOIN Gruz ON Gruz.id\_reisa=Reis.id\_reisa

INNER JOIN Voditeli ON Voditeli.id\_Voditelya=Gruz.id\_Voditelya

EXEC Proc1

* Найти водителей, которые провели в рейсах максимальное количество дней

CREATE TABLE MaxDay

(

DDay int,

Name nVARCHAR(100)

--)

Create proc Proc3

AS

BEGIN

SELECT MAX (DATEDIFF(day,Reis.vremya\_otpravleniya,Reis.vremya\_pribivaniya) ) ,Voditeli.FIO

FROM Reis

INNER JOIN Gruz ON Gruz.id\_reisa=Reis.id\_reisa

INNER JOIN Voditeli ON Voditeli.id\_Voditelya=Gruz.id\_Voditelya group by Voditeli.FIO

end

EXEC Proc3

SELECT \* FROM MaxDay

* Найти водителей, в рейсах которых никогда не случалась поломка заданного вида

Create function Zadanie3\_3(@vid\_p NVARCHAR(60))

Returns table

As

Return

(

SELECT Voditeli.FIO

FROM Tehnika

INNER JOIN Voditeli ON Voditeli.id\_Voditelya=Tehnika.id\_Voditelya

INNER JOIN Polomka ON Tehnika.id\_mashin =Polomka.id\_mashin

Where Polomka.vid\_polomki<>@vid\_p

);

GO

SELECT \* from Zadanie3\_3('Vid 1')

1. Подсчитать суммарное количество поломок заданного типа разделить на общее количество рейсов

CREATE procedure Zadanie444(@vid\_polomki NVARCHAR(80))

AS

BEGIN

DECLARE @kol int

DECLARE @vsego\_reisov int

DECLARE @avg\_polomok real

SET @kol=(

--

SELECT

count(\*) As '- ' from Polomka

where Polomka.vid\_polomki=@vid\_polomki)

----

SET @vsego\_reisov=(

SELECT count(\*) As '- ' from Reis)

----

SET @avg\_polomok=(

@kol/@vsego\_reisov)

PRINT @kol

PRINT @vsego\_reisov

PRINT @avg\_polomok

end

exec Zadanie444 'Vid 2'

1. Подсчитать, в какую сумму денег обошелся ремонт поломок каждого вида, и найти тот вид поломок, который встречается наиболее часто

CREATE procedure Zadanie3\_33333(@vid\_polomki NVARCHAR(50))

AS

BEGIN

DECLARE @summa int

DECLARE @polomka nVARCHAR(60)

SET @summa=(

Select SUM(Polomka.Stoimosti) From Polomka

Where Polomka.vid\_polomki=@vid\_polomki)

SET @polomka=

(select vid\_polomki from Polomka

group by vid\_polomki

having count(\*) >= ALL(select count(\*) from Polomka group by vid\_polomki))

PRINT @summa

PRINT @polomka

end

exec Zadanie3\_33333 'Vid 2'

1. Найти, на каких машинах заданный вид поломки никогда не случался.

CREATE FUNCTION dbo.ufn\_CarReports ( @CrashType nvarchar(50) )

RETURNS @Reports TABLE

(

CarID int primary key identity,

CarModel nvarchar(30) NOT NULL

)

AS

BEGIN

INSERT @Reports

SELECT T.model

FROM Polomka AS P

RIGHT OUTER JOIN Tehnika AS T

ON P.id\_mashin = T.id\_mashin

WHERE vid\_polomki <> @CrashType

RETURN

END

SELECT \* FROM [dbo].[ufn\_CarReports] ('Vid 2')

GO

**Программный код на C#**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApplication1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

dateTimePicker1.Format = DateTimePickerFormat.Time;

dateTimePicker2.Format = DateTimePickerFormat.Time;

}

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

{

string connectionString = @"Data Source= A203-14; Initial Catalog = АвтоБД\_1;Integrated Security=True";

string sqlExpression = "INSERT INTO [Tehnika] (model,id\_Voditelya) VALUES (@model,@id\_Voditelya)";

using (SqlConnection connection = new SqlConnection(connectionString))

{

try

{

connection.Open();

SqlCommand command = new SqlCommand(sqlExpression, connection);

command.Parameters.AddWithValue("model", textBox2.Text);

command.Parameters.AddWithValue("id\_Voditelya", textBox1.Text);

if (command.ExecuteNonQuery() == 1)

MessageBox.Show("Запись успешно добавлена!");

connection.Close();

}

catch (SqlException ex)

{

MessageBox.Show(ex.Message);

}

}

}

private void button4\_Click(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

string connectionString = @"Data Source= A203-14; Initial Catalog = АвтоБД\_1;Integrated Security=True";

string sqlExpression = "UPDATE [Detali] SET [id\_polomki]=@id\_polomki, [id\_mashin]=@id\_mashin,[kol\_vo\_detaley]=@kol\_vo\_detaley WHERE [id\_detali]=@id\_detali";

using (SqlConnection connection = new SqlConnection(connectionString))

{

try

{

connection.Open();

SqlCommand command = new SqlCommand(sqlExpression, connection);

command.Parameters.AddWithValue("id\_detali", textBox3.Text);

command.Parameters.AddWithValue("id\_polomki", textBox4.Text);

command.Parameters.AddWithValue("id\_mashin", textBox5.Text);

command.Parameters.AddWithValue("kol\_vo\_detaley", textBox6.Text);

if (command.ExecuteNonQuery() == 1)

MessageBox.Show("Запись успешно изменена!");

connection.Close();

}

catch (SqlException ex)

{

MessageBox.Show(ex.Message);

}

}

}

private void button3\_Click(object sender, EventArgs e)

{

string connectionString = @"Data Source= A203-14; Initial Catalog = АвтоБД\_1;Integrated Security=True";

string sqlExpression = "DELETE FROM [Jobs] WHERE id\_Jobs=@id\_Jobs";

using (SqlConnection connection = new SqlConnection(connectionString))

{

try

{

connection.Open();

SqlCommand command = new SqlCommand(sqlExpression, connection);

command.Parameters.AddWithValue("id\_Jobs", textBox7.Text);

if (command.ExecuteNonQuery() == 1)

MessageBox.Show("Запись успешно удалена!");

connection.Close();

}

catch (SqlException ex)

{

MessageBox.Show(ex.Message);

}

}

}

private void button8\_Click(object sender, EventArgs e)

{

string connectString = @"Data Source= A203-14; Initial Catalog = АвтоБД\_1;Integrated Security=True";

SqlConnection myConnection = new SqlConnection(connectString);

myConnection.Open();

string query = "SELECT \* FROM [Tehnika]";

SqlCommand command = new SqlCommand(query, myConnection);

SqlDataReader reader = command.ExecuteReader();

List<string[]> data = new List<string[]>();

while (reader.Read())

{

data.Add(new string[3]);

data[data.Count - 1][0] = reader[0].ToString();

data[data.Count - 1][1] = reader[1].ToString();

data[data.Count - 1][2] = reader[2].ToString();

}

reader.Close();

myConnection.Close();

foreach (string[] s in data)

dataGridView1.Rows.Add(s);

}

private void button9\_Click(object sender, EventArgs e)

{

string connectString = @"Data Source= A203-14; Initial Catalog = АвтоБД\_1;Integrated Security=True";

SqlConnection myConnection = new SqlConnection(connectString);

myConnection.Open();

string query = "SELECT \* FROM [Detali]";

SqlCommand command = new SqlCommand(query, myConnection);

SqlDataReader reader = command.ExecuteReader();

List<string[]> data = new List<string[]>();

while (reader.Read())

{

data.Add(new string[4]);

data[data.Count - 1][0] = reader[0].ToString();

data[data.Count - 1][1] = reader[1].ToString();

data[data.Count - 1][2] = reader[2].ToString();

data[data.Count - 1][3] = reader[3].ToString();

}

reader.Close();

myConnection.Close();

foreach (string[] s in data)

dataGridView2.Rows.Add(s);

}

private void button4\_Click\_1(object sender, EventArgs e)

{

string connectString = @"Data Source= A203-14; Initial Catalog = АвтоБД\_1;Integrated Security=True";

SqlConnection myConnection = new SqlConnection(connectString);

myConnection.Open();

string query = "SELECT \* FROM [Jobs]";

SqlCommand command = new SqlCommand(query, myConnection);

SqlDataReader reader = command.ExecuteReader();

List<string[]> data = new List<string[]>();

while (reader.Read())

{

data.Add(new string[3]);

data[data.Count - 1][0] = reader[0].ToString();

data[data.Count - 1][1] = reader[1].ToString();

data[data.Count - 1][2] = reader[2].ToString();

}

reader.Close();

myConnection.Close();

foreach (string[] s in data)

dataGridView3.Rows.Add(s);

}

private void button5\_Click(object sender, EventArgs e)

{

string connectString = @"Data Source= A203-14; Initial Catalog = АвтоБД\_1;Integrated Security=True";

string sqlExpression = "dbo.Zadanie112";

SqlConnection connection = new SqlConnection(connectString);

connection.Open();

SqlCommand command = new SqlCommand(sqlExpression, connection);

command.CommandType = System.Data.CommandType.StoredProcedure;

command.Parameters.AddWithValue("PayDay1", "2017-01-01 00:00:00");

command.Parameters.AddWithValue("PayDay2", "2017-03-30 00:00:00");

SqlDataReader reader = command.ExecuteReader();

List<string[]> data = new List<string[]>();

while (reader.Read())

{

data.Add(new string[2]);

data[data.Count - 1][0] = reader[0].ToString();

data[data.Count - 1][1] = reader[1].ToString();

}

reader.Close();

connection.Close();

foreach (string[] s in data)

dataGridView4.Rows.Add(s);

}

private void button7\_Click(object sender, EventArgs e)

{

string connectString = @"Data Source= A203-14; Initial Catalog = АвтоБД\_1;Integrated Security=True";

string sqlExpression = "dbo.Proc1";

SqlConnection connection = new SqlConnection(connectString);

connection.Open();

SqlCommand command = new SqlCommand(sqlExpression, connection);

command.CommandType = System.Data.CommandType.StoredProcedure;

SqlDataReader reader = command.ExecuteReader();

List<string[]> data = new List<string[]>();

while (reader.Read())

{

data.Add(new string[2]);

data[data.Count - 1][0] = reader[0].ToString();

data[data.Count - 1][1] = reader[1].ToString();

}

reader.Close();

connection.Close();

foreach (string[] s in data)

dataGridView5.Rows.Add(s);

}

private void button6\_Click(object sender, EventArgs e)

{

string connectString = @"Data Source= A203-14; Initial Catalog = АвтоБД\_1;Integrated Security=True";

string sqlExpression = "SELECT \* FROM [MaxDay]";

SqlConnection connection = new SqlConnection(connectString);

connection.Open();

SqlCommand command = new SqlCommand(sqlExpression, connection);

SqlDataReader reader = command.ExecuteReader();

List<string[]> data = new List<string[]>();

while (reader.Read())

{

data.Add(new string[2]);

data[data.Count - 1][0] = reader[0].ToString();

data[data.Count - 1][1] = reader[1].ToString();

}

reader.Close();

connection.Close();

foreach (string[] s in data)

dataGridView6.Rows.Add(s);

}

private void button10\_Click(object sender, EventArgs e)

{

string connectString = @"Data Source= A203-14; Initial Catalog = АвтоБД\_1;Integrated Security=True";

string sqlExpression = "SELECT \* FROM dbo.Zadanie3\_3(@vid\_p)";

SqlConnection connection = new SqlConnection(connectString);

connection.Open();

SqlCommand command = new SqlCommand(sqlExpression, connection);

command.Parameters.AddWithValue("vid\_p", textBox8.Text);

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

SqlDataReader reader = command.ExecuteReader();

List<string[]> data = new List<string[]>();

while (reader.Read())

{

data.Add(new string[1]);

data[data.Count - 1][0] = reader[0].ToString();

}

reader.Close();

connection.Close();

foreach (string[] s in data)

dataGridView7.Rows.Add(s);

}

private void button12\_Click(object sender, EventArgs e)

{

string connectString = @"Data Source= A203-14; Initial Catalog = АвтоБД\_1;Integrated Security=True";

string sqlExpression = "SELECT \* FROM [dbo].[ufn\_CarReports](@vid\_polomki)";

SqlConnection connection = new SqlConnection(connectString);

connection.Open();

SqlCommand command = new SqlCommand(sqlExpression, connection);

command.Parameters.AddWithValue("vid\_polomki", textBox10.Text);

SqlDataReader reader = command.ExecuteReader();

List<string[]> data = new List<string[]>();

while (reader.Read())

{

data.Add(new string[2]);

data[data.Count - 1][0] = reader[0].ToString();

data[data.Count - 1][1] = reader[1].ToString();

}

reader.Close();

connection.Close();

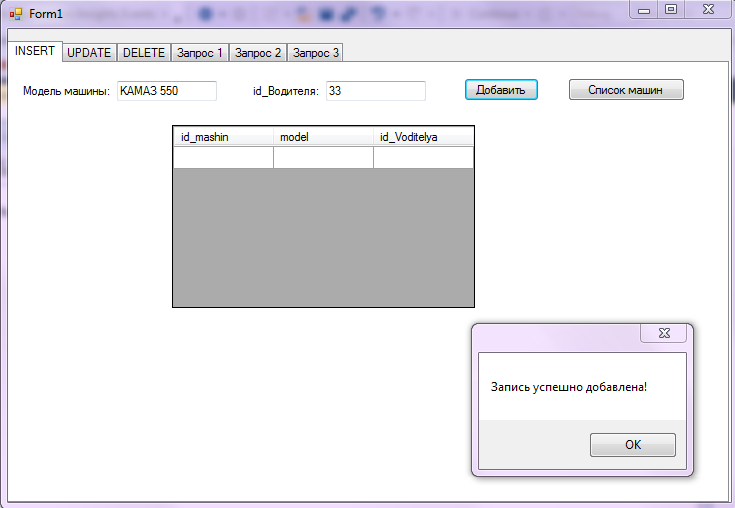
foreach (string[] s in data)

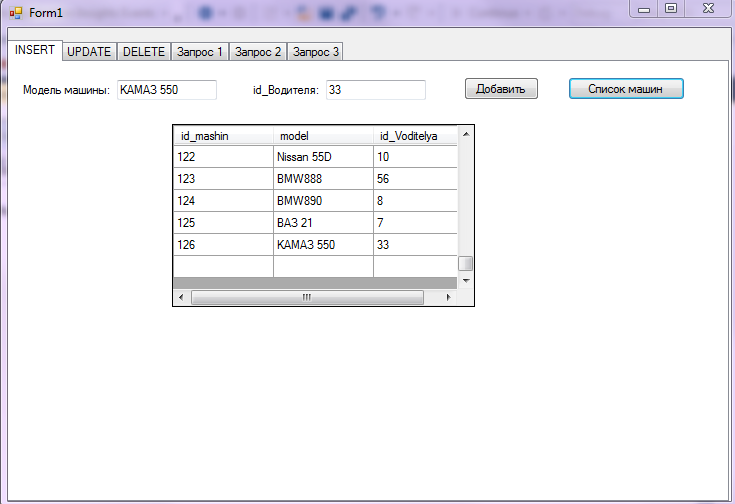
dataGridView8.Rows.Add(s);

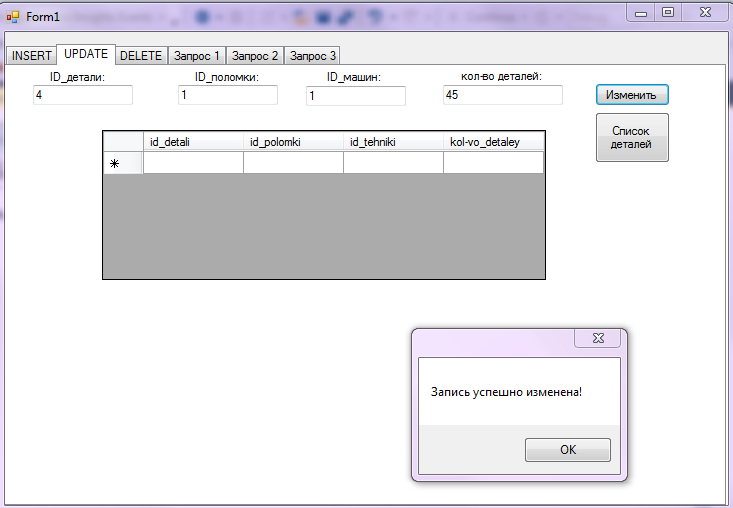
}

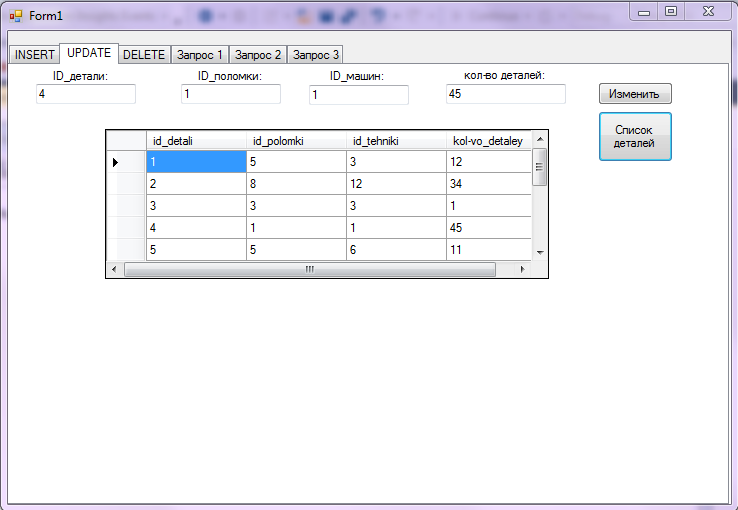
}

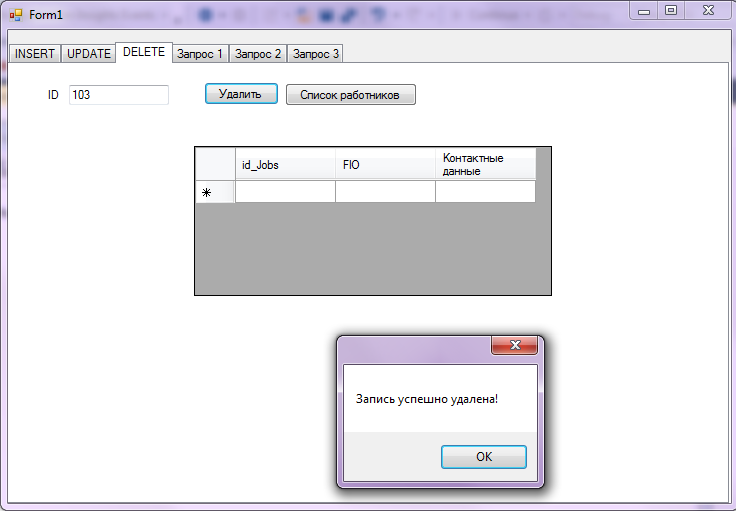
}

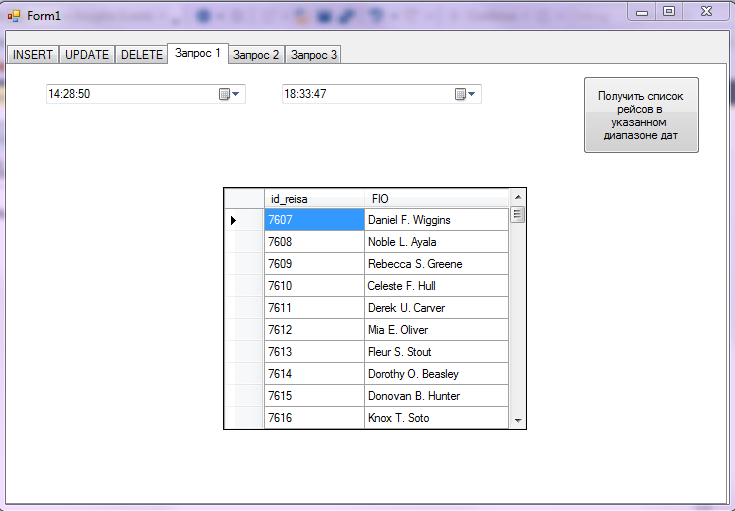


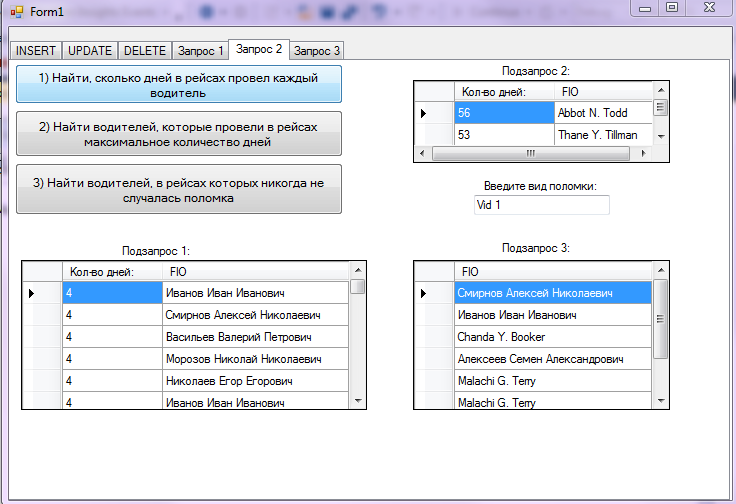


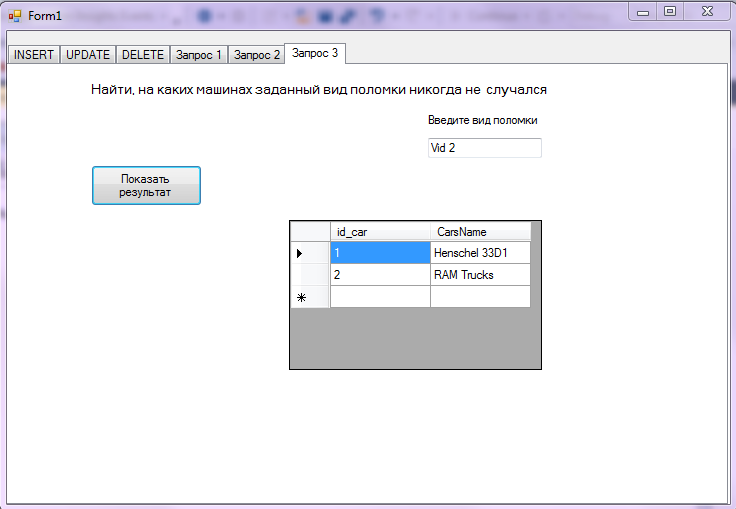












**Вывод*:*** выполнена программная реализация заданного задания на языке С#.