

**МИНОБРНАУКИ РОССИИ**  
**федеральное государственное бюджетное образовательное учреждение**  
**высшего образования**  
**«Московский государственный технологический университет «СТАНКИН»**  
**(ФГБОУ ВО МГТУ «СТАНКИН»)**

---

**Институт**  
информационных технологий

**Кафедра**  
информационных систем

**Отчет по лабораторной работе №10**

по дисциплине «**Управление данными**»

на тему: «Вызов хранимых процедур и пользовательских функций для работы с БД из  
клиентского приложения в MS Visual Studio»

**Студент**  
группа ИДБ–22–06

**Мустафаева П.М.**

---

подпись

**Руководитель**  
старший преподаватель

**Быстрикова В. А.**

---

подпись

## **ЦЕЛЬ РАБОТЫ**

Целью работы является овладение практическими навыками вызова хранимых процедур и пользовательских функций в клиентском приложении для работы с базой данных с применением встроенных инструментов на MS Visual Studio.

## ХОД РАБОТЫ

### ФОРМА Form1

Создание главной кнопочной формы. На основной форме была размещена надпись (Label) и три кнопки (Button). У всех кнопок были изменены имена на btnProc, btnScalarFun и btnTableFun. Основная форма открывается при запуске приложения. Кнопка «Вызов процедуры» открывает форму frmProc. Кнопка «Вызов скалярной функции» открывает форму frmScalarFun. Кнопка «Вызов табличной функции» открывает форму frmTableFun.

На рис. 1 представлена форма в режиме конструктора.

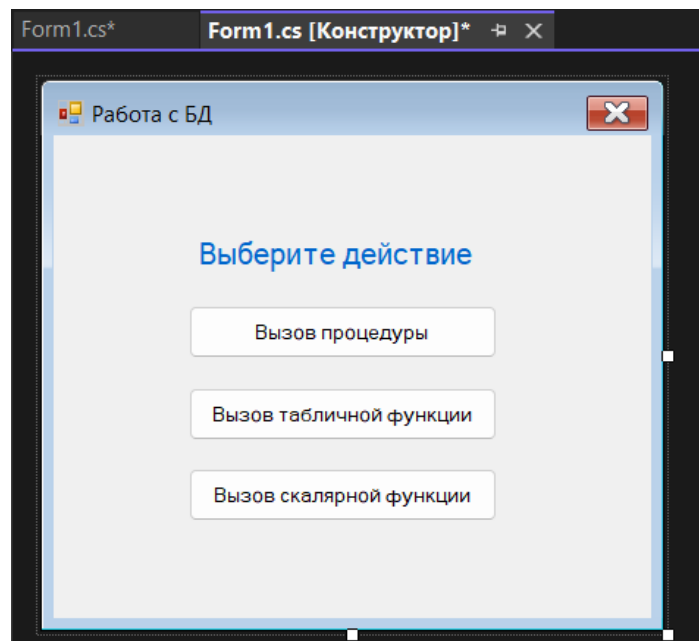


Рис. 1. Форма Form1 в режиме конструктора

Листинг кода программы:

```
namespace MustafaevaApp
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnProc_Click(object sender, EventArgs e)
```

```

    {
        Form frm = new frmProc();
        frm.Show();
    }

    private void frmScalarFun_Click(object sender, EventArgs e)
    {
        Form frm = new frmTableFun();
        frm.Show();
    }

    private void btnTableFun_Click(object sender, EventArgs e)
    {
        Form frm = new frmScalarFun();
        frm.Show();
    }
}

```

Результат выполнения задания представлен на рис. 2.

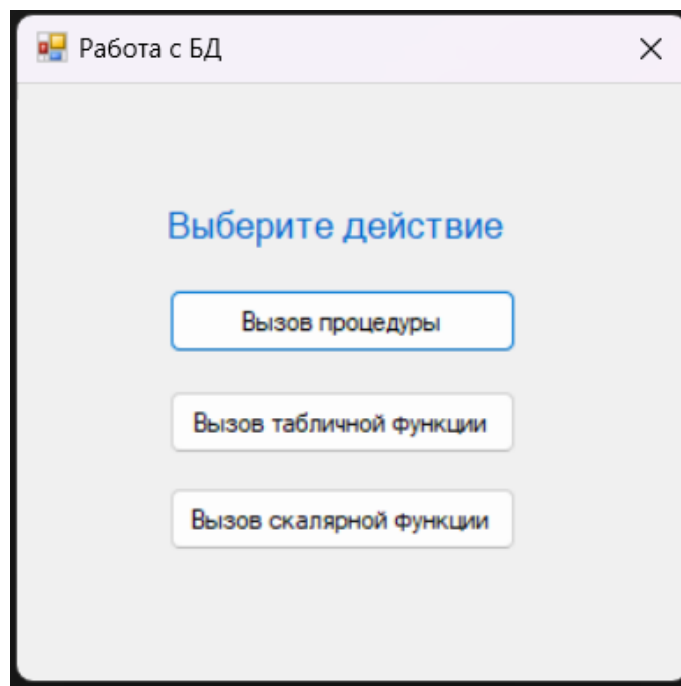


Рис. 2. Общий вид формы Form1

## ФОРМА frmProc

С помощью хранимой процедуры выполнить перевод лектора Воронова В.В. на кафедру ИС, предварительно проверив его наличие в базе данных и отсутствие на новой кафедре полных однофамильцев. В основной программе вызвать данную процедуру и прокомментировать результаты (выдать сообщение об успешности выполнения перевода или о возникшей ошибке).

Листинг кода создания процедуры:

```
CREATE PROCEDURE TransferL
    @fio VARCHAR(30), @kafedra VARCHAR(10), @flag VARCHAR(50)
OUTPUT //входные параметры: ФИО лектора, кафедра и флаг, содержащий
сообщение о результате операции
AS
BEGIN
    DECLARE @Lect_id INT //переменная, для хранения Id лектора
    //запрос для поиска лектора по его ФИО
    SELECT @Lect_id = Id_Lect
    FROM Lecturer
    WHERE FIO = @fio;
    //проверка, если лектор не найден выводится сообщение
    IF @Lect_id IS NULL
    BEGIN
        SET @flag = 'Лектор с заданным фио не найден';
        RETURN;
    END
    //проверка, есть ли на новой кафедре однофамилец
    IF EXISTS (
        SELECT *
        FROM Lecturer
        WHERE FIO = @fio AND Kafedra = @kafedra
    )
```

```

BEGIN

    SET @flag = 'На новой кафедре есть однофамилец'; //вывод
сообщения о том, что на кафедре есть однофамилец

    RETURN;

END

//обновление кафедры лектора на новую

UPDATE Lecturer

SET Kafedra = @kafedra

WHERE Id_Lect = @lect_id;

SET @flag = 'Лектор успешно переведен на кафедру'; //вывод
сообщения об успешном переводе лектора

END;

```

Исходные данные для процедуры представлены в приложении А.

Форма frmProc в режиме конструктора, а также ее свойства представлены на рис. 3-8.

	Id_Lect	FIO	Stage	Kafedra
*				

Рис. 3. Форма frmProc в режиме конструктора

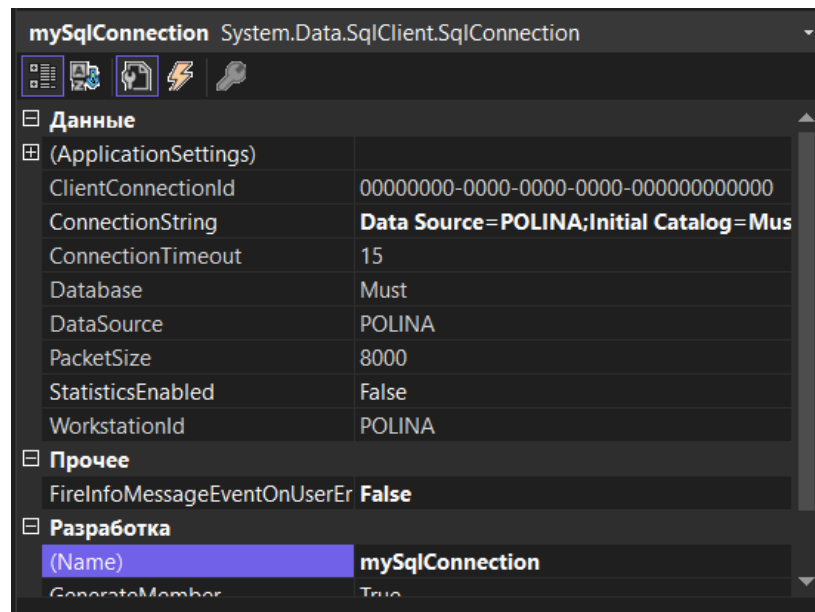


Рис. 4. Свойства коллекции mySqlConnection для хранимой процедуры

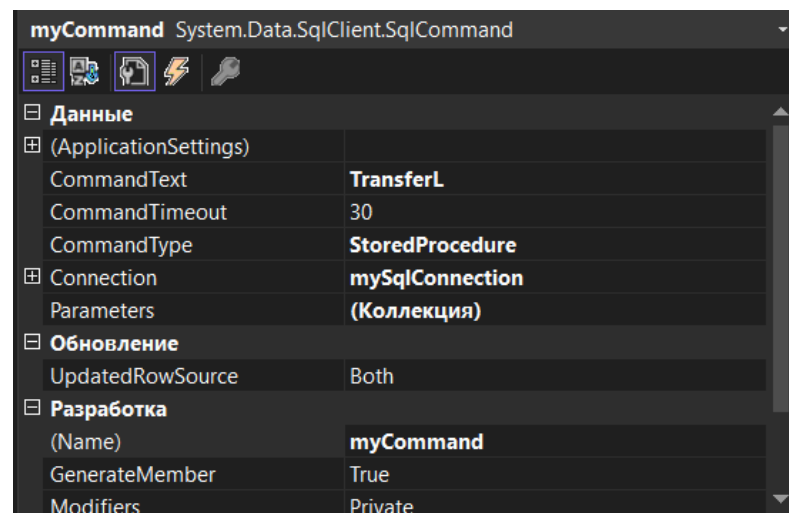


Рис. 5. Свойства myCommand для хранимой процедуры

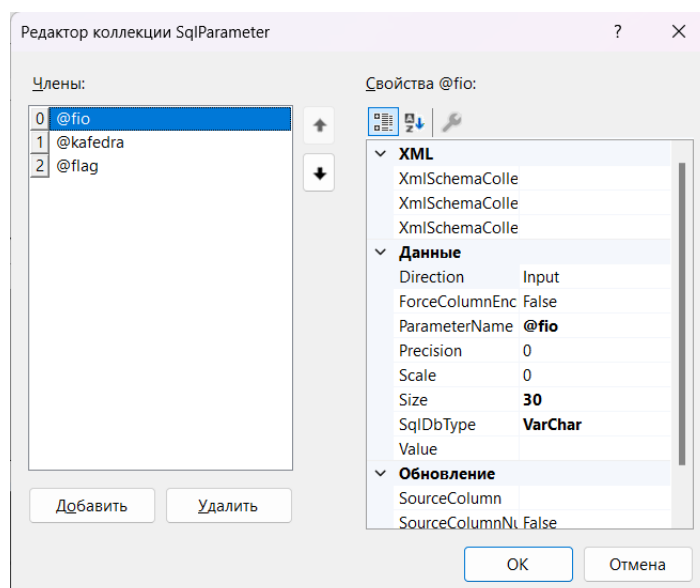


Рис. 6. Окно настройки параметров объекта myCommand

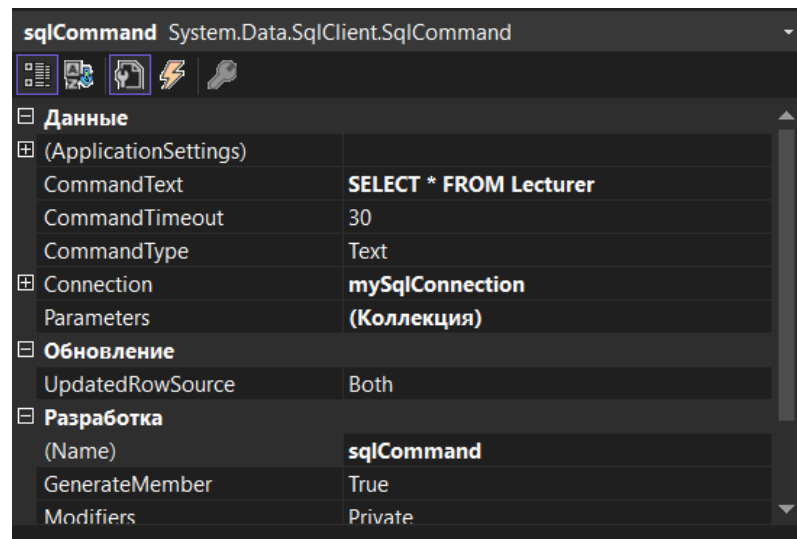


Рис. 7. Свойства sqlCommand для хранимой процедуры

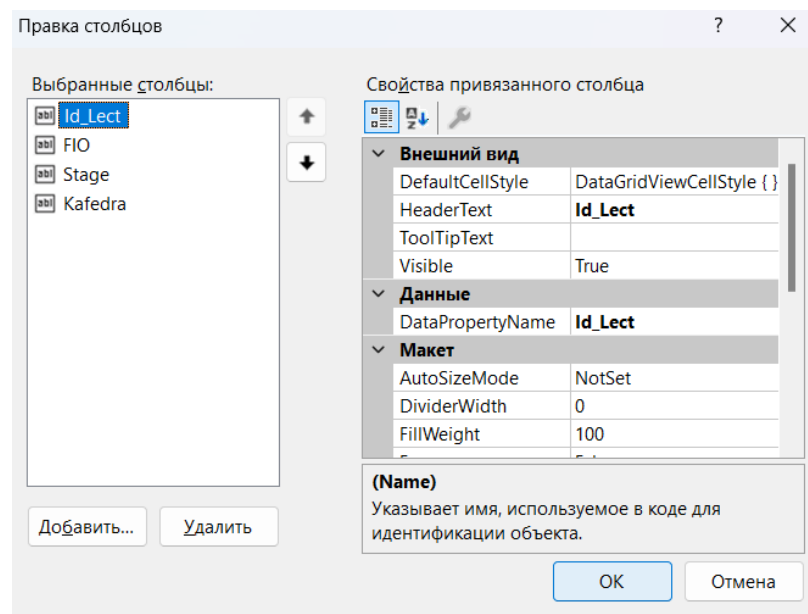


Рис. 8. Редактор столбцов для dataGridViewRes до применения процедуры

Листинг кода программы:

```
namespace MustafaevaApp
{
    public partial class frmProc : Form
    {
        public frmProc()
        {
            InitializeComponent();
        }
        private void frmProc_Load(object sender, EventArgs e)
        {
            this.lecturerTableAdapter.Fill(this.mustDataSet.Lecturer);
        }
    }
}
```



```

    }
    private void btnExecute_Click(object sender, EventArgs e)
    {
        myCommand.Parameters["@fio"].Value = txtFIO.Text;
        myCommand.Parameters["@kafedra"].Value = txtKafedra.Text;
        mySqlConnection.Open();
        myCommand.ExecuteNonQuery();
        mySqlConnection.Close();
        try
        {
            String result = (String)myCommand.Parameters["@flag"].Value;
            MessageBox.Show(result);
        }
        catch (Exception)
        {
            MessageBox.Show("Некорректные исходные данные");
        }
        mySqlConnection.Open();
        var temp = new DataTable();
        temp.Load(sqlCommand.ExecuteReader());
        dataGridViewRes.DataSource = temp;
        mySqlConnection.Close();
    }
}
}

```

Результат выполнения задания представлен на рис. 9-11.

	Id_Lect	FIO	Stage	Kafedra
▶	10	Быстрикова В.А.	27	ИС
	11	Разумовский А.И.	29	ИСиТ
	12	Елисеева Ю.В.	33	ПМ
	13	Саркисова И.О.	29	ИТиВС
	14	Иванова А.А.	26	ПМ
*				

Рис. 9. Форма frmProc после вызова процедуры до перевода лектора на кафедру

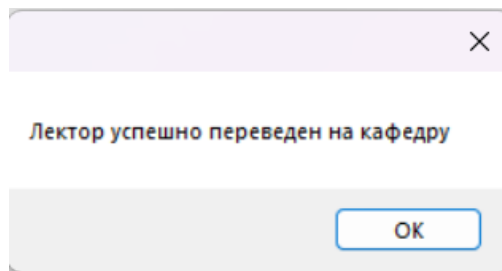


Рис. 10. Уведомление о выполнении хранимой процедуры

Процедура

ФИО лектора

Перевести на кафедру

	Id_Lect	FIO	Stage	Kafedra
▶	10	Быстрикова В.А.	27	ИС
	11	Разумовский А.И.	29	ИСиТ
	12	Елисеева Ю.В.	33	ПМ
	13	Саркисова И.О.	29	ИС
	14	Иванова А.А.	26	ПМ
*				

Рис. 11. Форма frmProc после вызова поцедуры после перевода лектора на кафедру

## ФОРМА frmTableFun

С помощью функции определить, кто среди студентов 3-го курса сдал указанный экзамен на отлично. В основной программе вывести их фамилии и группы для дисциплины Математика.

Листинг кода создания функции:

```
CREATE FUNCTION ReturnMark (@Kurs INT, @Subj VARCHAR(10))
    RETURNS @stud TABLE (Fio VARCHAR(100), N_group
    VARCHAR(20)) //создание таблицы с параметрами: ФИО и номер группы
AS
BEGIN
    INSERT INTO @stud (Fio, N_group) //занесение данных в табличную
    переменную @stud
    SELECT Student.FIO, Student.NameGroup
    FROM Student
    JOIN Exam ON Student.Id_Student = Exam.Id_Student // объединение с
    таблицей Exam по полю Id_Student
    JOIN StudGroup ON Student.NameGroup = StudGroup.NameGroup //
    Объединение с таблицей StudGroup по полю NameGroup
    WHERE StudGroup.Kurs = @Kurs
    AND Exam.Subject = @Subj
    AND Exam.Mark BETWEEN 45 AND 54; //выбираются только
    отличники
    RETURN;
END;
```

Исходные данные для функции представлены в приложении А.

Форма frmTableFun в режиме конструктора, а также свойства формы представлены на рис. 12-15.

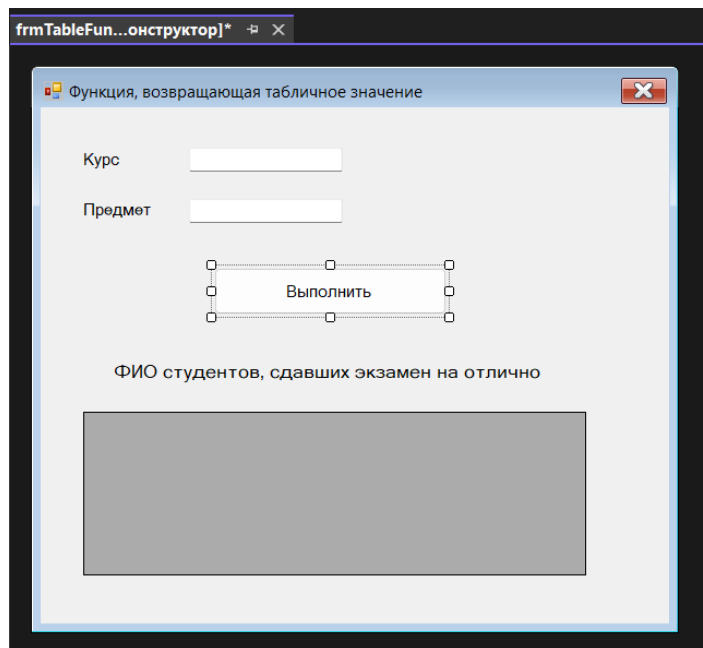


Рис. 12. Форма frmTableFun в режиме конструктора

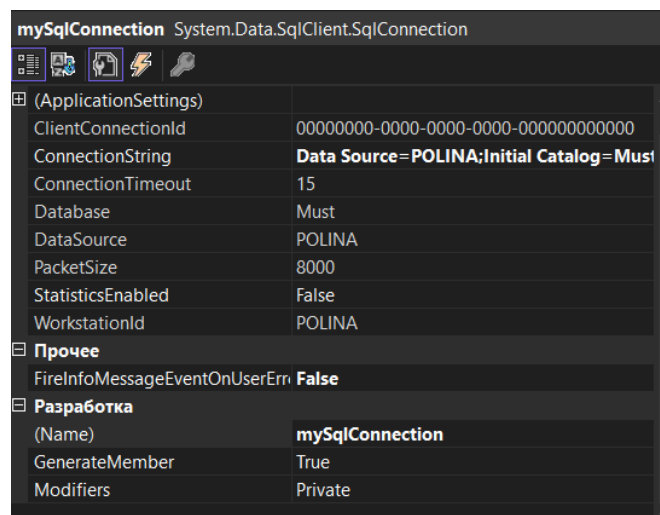


Рис. 13. Свойства mySqlConnection для табличной функции

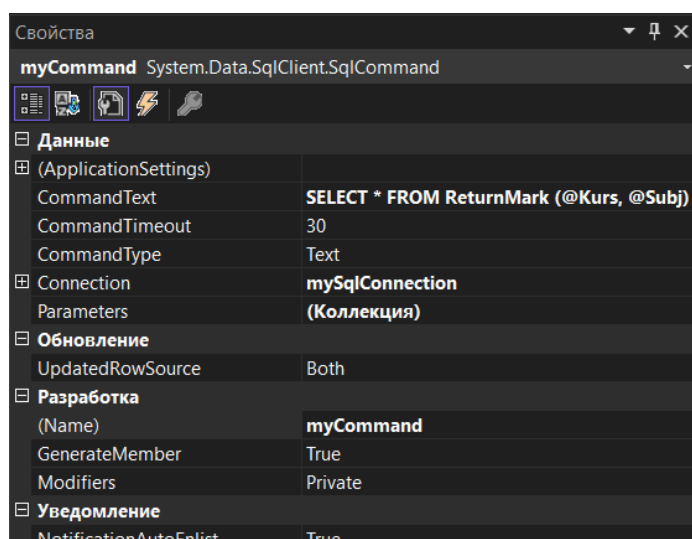


Рис. 14. Свойства myCommand для табличной функции

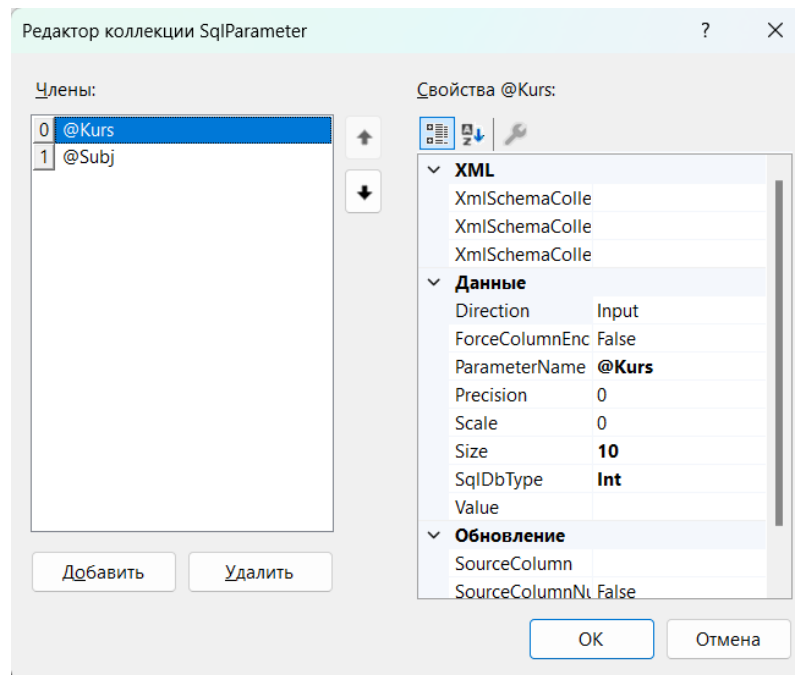


Рис. 15. Окно настройки параметров объекта myCommand

Листинг кода программы:

```
namespace MustafaevaApp
{
    public partial class frmTableFun : Form
    {
        public frmTableFun()
        {
            InitializeComponent();
        }

        private void btnExecute_Click(object sender, EventArgs e)
        {
            myCommand.Parameters["@Kurs"].Value = txtKurs.Text;
            myCommand.Parameters["@Subj"].Value = txtSubj.Text;

            // открыть соединение с БД
            mySqlConnection.Open();

            // создать временную таблицу temp
            var temp = new DataTable();
            // выполнить табличную функцию и вернуть таблицу в объект
            Reader
            // заполнить таблицу temp данными из Reader
            temp.Load(myCommand.ExecuteReader());
            //установить связь с объектом типа dataGridView
            dataGridViewRes.DataSource = temp;
            // закрыть соединение с БД
```

```

        MySqlConnection.Close();
    }
}

```

Результат работы формы представлен на рис. 16.

Функция, возвращающая табличное значение

Курс: 3

Предмет: Математика

Выполнить

ФИО студентов, сдавших экзамен на отлично

	Fio	N_group
►	Ильин К.Г.	ИДБ-21-09
	Ларченко В.П.	АДБ-21-08
	Бигаева А.Д.	АДБ-21-08
*		

Рис. 16. Форма frmTableFun после вызова табличной функции

## ФОРМА frmScalarFun

Создать функцию, возвращающую общее количество студентов отличников 3-го курса, используя при этом функцию из п.2.5 (в). В основной программе вызвать данную функцию для дисциплины Управление данными.

Листинг кода создания функции:

```
// Создание функции CountStud, которая принимает предмет (@Subj) и
возвращает количество студентов

CREATE FUNCTION CountStud (@Subj VARCHAR(50))
RETURNS INT
AS
BEGIN
    DECLARE @Count INT; // Объявление переменной @Count для
хранения количества студентов

    SELECT @Count = COUNT(*)
    FROM ReturnMark(3, @Subj); //используем функцию для подсчета
кол-ва студентов 3-го курса и сдали экзамен по предмету @subj с оценкой 45-
54

    RETURN @Count; // Возвращаем результат - количество студентов
END;
```

Исходные данные для процедуры представлены в приложении А.

Форма ScalarFun в режиме конструктора, а также свойства формы представлены на рис. 17-20.

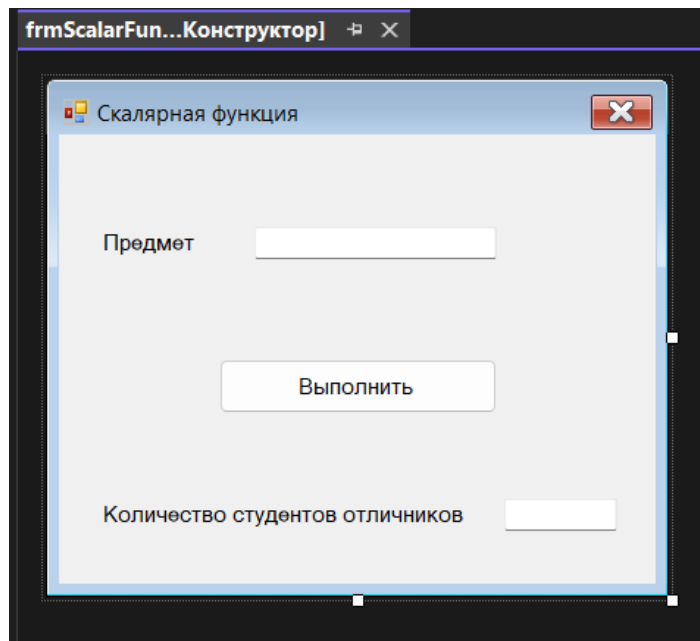


Рис. 17. Форма frmScalarFun в режиме конструктора

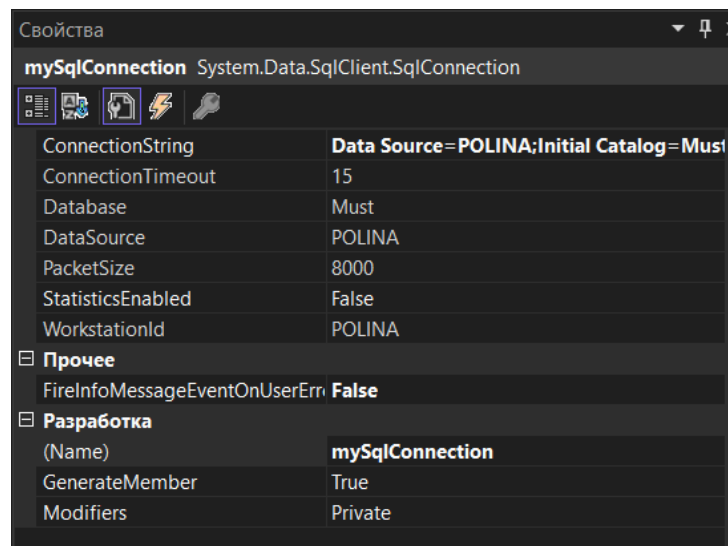


Рис. 18. Свойства mySqlCollection для табличной функции

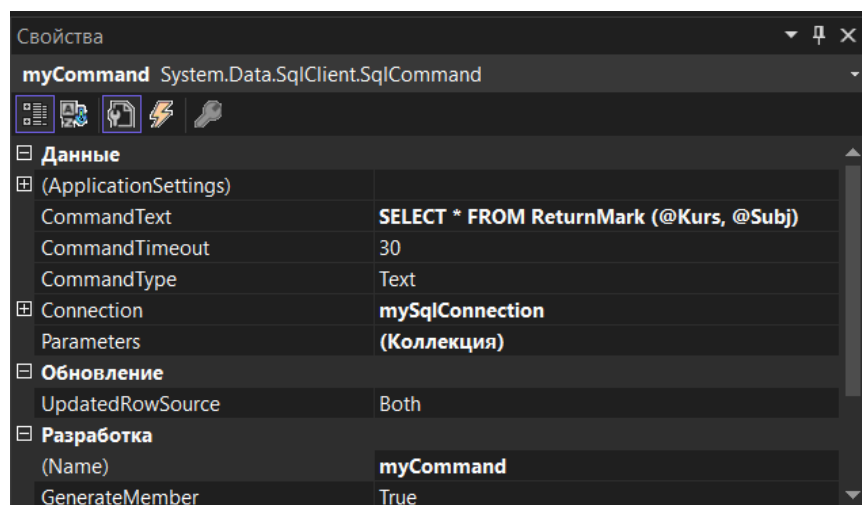


Рис. 19. Свойства myCommand для табличной функции



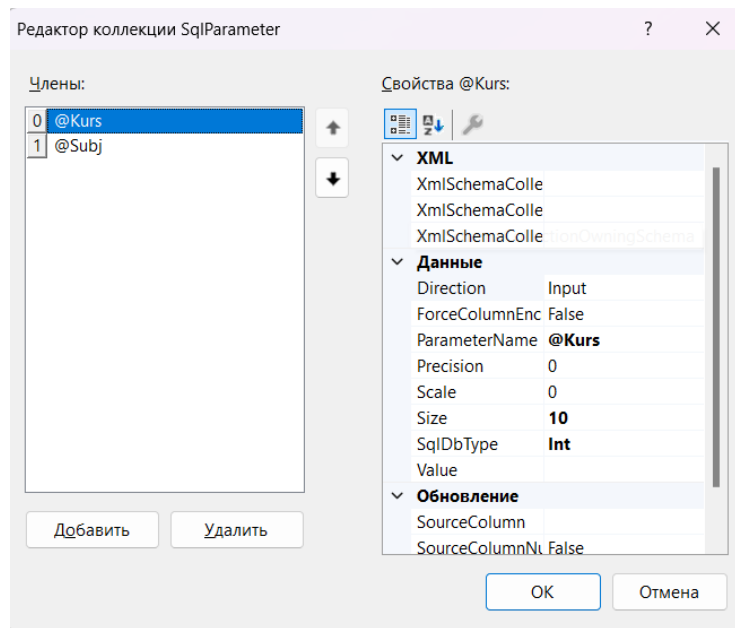


Рис. 20. Окно настройки параметров объекта myCommand

Листинг кода программы:

```
namespace MustafaevaApp
{
    public partial class frmScalarFun : Form
    {
        public frmScalarFun()
        {
            InitializeComponent();
        }

        private void btnExecute_Click(object sender, EventArgs e)
        {
            // присвоить значения входным параметрам процедуры
            myCommand.Parameters["@Subj"].Value = txtSubj.Text;
            // открыть соединение с БД
            mySqlConnection.Open();
            // выполнить sql-выражение (функцию) и преобразовать
результат в строку
            txtResult.Text = myCommand.ExecuteScalar().ToString();
            // закрыть соединение с БД
            mySqlConnection.Close();
        }
    }
}
```

Результат работы формы представлен на рис. 21.

Скалярная функция

Предмет

Количество студентов отличников

Рис. 21. Форма frmScalarFun после вызова табличной функции

## **ВЫВОД**

В ходе лабораторной работы были изучены способы вызова хранимых процедур и пользовательских функций в клиентском приложении для работы с базой данных с применением встроенных элементов на MS Visual Studio.

Были созданы формы для вызова процедуры, табличной и скалярной функций. Для хранимой процедуры была добавлена таблица с исходными данными из таблицы Lecturer, для наглядного отображения результата работы процедуры. В результате, после перевода лектора на другую кафедру данные в таблице изменялись и отображались обновленные данные. Для табличной и скалярной функций в качестве текста команды для вызова функций на форму был добавлен запрос.

Благодаря использованию встроенных элементов для работы с базой данных удалось наглядно продемонстрировать изменения, происходящие при вызове процедуры, табличной и скалярной функций.

Исходные данные для процедур и функций

	Id_Student	Subject	Mark	Exam_Date	Id_Lect
	107	УД	38	2024-01-08	10
	107	ЭВМ	45	2024-01-16	13
	108	Математика	27	2024-01-20	12
	108	ООП	39	2024-01-14	11
	109	Математика	54	2024-01-20	12
	109	ООП	25	2024-01-14	11
	110	УД	33	2024-01-08	10
	110	ЭВМ	50	2024-01-16	13
	111	УД	40	2024-01-12	10
	111	ЭВМ	36	2024-01-16	13
	112	Математика	35	2024-01-21	12
	112	ООП	46	2024-01-15	11
	113	Математика	25	2024-01-21	12
	113	ООП	40	2024-01-15	11
	114	Математика	52	2024-01-21	12
	114	УД	46	2024-01-12	10
	115	ООП	42	2024-01-15	11
	115	УД	30	2023-01-12	10
	116	Математика	40	2024-01-14	12
	116	УД	39	2024-01-15	10
	117	Математика	45	2024-01-24	12
	117	ООП	25	2024-01-19	11
	118	Математика	27	2024-01-24	12
	118	ООП	54	2024-01-19	11
	119	Математика	33	2024-01-24	12
	119	ООП	25	2024-01-19	11
	120	Математика	48	2024-01-24	12
	120	ООП	54	2024-01-19	11
*	NULL	NULL	NULL	NULL	NULL

Рис. А.1. Исходные данные таблицы «Exam»

	Id_Student	FIO	Birthday	Gender	NameGroup	Stip
►	101	Иванова П.Т.	2004-09-23	ж	ИДБ-22-06	3200
	102	Смирнова Е.А.	2004-09-07	ж	ИДБ-22-06	4100
	103	Петрова А.А.	2002-07-21	ж	ИДБ-22-06	8000
	104	Иванько Н.С.	2004-03-24	м	ИДБ-22-06	2500
	105	Чернышев С.С.	2004-03-15	м	ИДБ-22-06	5300
	106	Петренко А.М	2005-06-05	м	ИДБ-23-03	1500
	107	Серова М.Н.	2005-11-12	ж	ИДБ-23-03	4000
	108	Клинов Д.А.	2005-04-30	м	ИДБ-23-03	2000
	109	Летов М.Д.	2005-12-01	м	ИДБ-23-03	3500
	110	Морозова Л.С.	2005-07-01	ж	ИДБ-23-03	6000
	111	Пахомов Я.Я.	2003-03-13	м	ИДБ-23-03	4200
	112	Сергеева С.С.	2003-05-04	ж	ИДБ-21-09	4000
	113	Волков Н.В.	2003-08-08	м	ИДБ-21-09	3200
	114	Ильин К.Г.	2003-10-16	м	ИДБ-21-09	5000
	115	Зоркина Г.С.	2003-01-31	ж	ИДБ-21-09	2500
	116	Седов А.А.	2002-04-04	м	АДБ-21-08	4600
	117	Ларченко В.Л.	2002-02-25	м	АДБ-21-08	5300
	118	Грозный И.О.	2002-11-08	м	АДБ-21-08	1500
	119	Минаева И.Н	2002-11-11	ж	АДБ-21-08	2000
	120	Бигаева А.Д.	2002-06-16	ж	АДБ-21-08	5900
*	NULL	NULL	NULL	NULL	NULL	NULL

Рис. А.2. Исходные данные таблицы «Student»

	NameGroup	Kurs	Kafedra
►	АДБ-21-08	3	ПМ
	ИДБ-21-09	3	ИС
	ИДБ-22-06	2	ИС
	ИДБ-23-03	1	ИС
*	NULL	NULL	NULL

Рис. А.3. Исходные данные таблицы «StudGroup»

	Id_Lect	FIO	Stage	Kafedra
►	10	Быстриков...	27	ПМ
	11	Разумовски...	29	ИС
	12	Елисеева ...	33	ПМ
	13	Саркисова ...	29	ИТиВС
	14	Иванова А....	26	NULL
*	NULL	NULL	NULL	NULL

Рис. А.4. Исходные данные таблицы «Lecturer»