

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

ПРОГРАММНОЕ ПОДКЛЮЧЕНИЕ К СЕРВЕРУ POSTGRES SQL

Отчёт о лабораторной работе № 4
по дисциплине «Базы данных»

Выполнил: студент гр. 431-3

_____ Андреев Д.П.

«___» _____ 2023 г.

Проверил: ассистент каф. АСУ

_____ Яблонский Я. В.

«___» _____ 2023 г.

Томск 2023

Оглавление

1	Цель лабораторной работы.....	3
2	Описание требований к отчётам.....	4
3	Листинг.....	5
4	Описание процесса реализации.....	2
5	Выводы.....	2

1 Цель лабораторной работы

Научиться создавать программы, взаимодействующие с сервером СУБД.

2 Описание требований к отчётам

На рисунке 2.1 показано как должен выглядеть отчёт.

Отчёт о задолжниках			
Гр. 764		Задолжников	3
Волков П.Г.		Всего долгов	3
	Дисциплина	Оценка	
	ТПМП	НЕУДОВЛ	
	ПМКР	НЕУДОВЛ	
Медведев С.О.	СПМ	Н/А	
		Всего долгов	6
	Дисциплина	Оценка	
	ТПМП	Н/А	
Гарькина Т.П.	
		Всего долгов	1
	Дисциплина	Оценка	
	ТПМП	Н/А	

Рисунок 2.1 – Форма отчёта о задолжниках

3 ЛИСТИНГ

```

using System;
using Npgsql;

namespace lab_4
{
    class Program
    {
        static void Main(string[] args)
        {
            // Укажите свои данные для
            // подключения к PostgreSQL
            string connectionString =
                "Server=localhost; Port=5432; Database=decanat;
                User Id=postgres; Password=12345678;";

            // Список атрибутов таблиц, для
            // удобства в дальнейшем
            string[][] attributes = { new string[]
            { "GrNum", "Cours", "Qt"}, //STUD_GROUP
                                     new string[]
            { "StNum", "StNam", "GrNum", "Addr", "Tel"},
            //STUD
                                     new string[] { "Abbr",
            "DisNam"}, //DIS
                                     new string[] { "Abbr",
            "GrNum", "ExDat"}, //Exam
                                     new string[]
            { "StNum", "Abbr", "Dat", "Ball" } //BALL
            };

            // Создание подключения
            var connection = new
            Npgsql.NpgsqlConnection(connectionString);
            connection.Open();
            Console.WriteLine("Соединение с
            PostgreSQL установлено.");

            bool flag = true;

            while (flag)
            {
                Console.WriteLine("Выберите
                действие:\n\t1. Внести новые данные." +
                                     "\n\t2.
                Вывести отчёт." +
                                     "\n\t3.
                Выйти");
                int mode;
                mode =
                Convert.ToInt32(Console.ReadLine());

                switch (mode)
                {
                    case 1:
                        Console.WriteLine("Выберите
                        таблицу:\n\t1. STUD_GROUP." +
                                             "\n\t2.
                        STUD." +
                                             "\n\t3.
                        DIS" +
                                             "\n\t4.
                        Exam" +
                                             "\n\t5.
                        BALL" +
                                             "\n\t6.
                        Вернуться");
                        mode =
                        Convert.ToInt32(Console.ReadLine());
                        //int count_attr = 0;
                        var cmd =
                        connection.CreateCommand();
                        if (mode == 1)
                        {
                            //Вносим данные
                            string insertQuery =
                            "INSERT INTO STUD_GROUP (GrNum, Cours,
                            Qt) VALUES (@GrNum, @Cours,@Qt)";

```

```

        cmd.CommandText =
insertQuery;

        // Параметры для вставки
значений в запрос

        Console.WriteLine($"Введи
те значение атрибута {attributes[mode - 1][0]}:");

cmd.Parameters.AddWithValue("GrNum",
Console.ReadLine());

        Console.WriteLine($
"Введите значение атрибута {attributes[mode - 1]
[1]}:");

cmd.Parameters.AddWithValue
lue("Cours", Console.ReadLine());

        Console.WriteLine($
"Введите значение атрибута {attributes[mode - 1]
[2]}:");

cmd.Parameters.AddWithValue
lue("Qt", Convert.ToInt32(Console.ReadLine()));

    }
    if (mode == 2)
    {
        //Вносим данные

        string insertQuery =
"INSERT INTO STUD (StNum, StNam,
GrNum,Addr,Tel) VALUES (@StNum,
@StNam,@GrNum,@Addr,@Tel)";

        cmd.CommandText =
insertQuery;

        // Параметры для вставки
значений в запрос

        Console.WriteLine($"Введи
те значение атрибута {attributes[mode - 1][0]}:");

cmd.Parameters.AddWithValue("StNum",
Console.ReadLine());

```

```

        Console.WriteLine($
"Введите значение атрибута {attributes[mode - 1]
[1]}:");

cmd.Parameters.AddWithValue
lue("StNam", Console.ReadLine());

        Console.WriteLine($
"Введите значение атрибута {attributes[mode - 1]
[2]}:");

cmd.Parameters.AddWithValue
lue("GrNum", Console.ReadLine());

        Console.WriteLine($
"Введите значение атрибута {attributes[mode - 1]
[3]}:");

cmd.Parameters.AddWithValue
lue("Addr", Console.ReadLine());

        Console.WriteLine($
"Введите значение атрибута {attributes[mode - 1]
[4]}:");

cmd.Parameters.AddWithValue
lue("Tel", Console.ReadLine());

    }
    if (mode == 3)
    {
        //Вносим данные

        string insertQuery =
"INSERT INTO DIS (Abbr, DisNam) VALUES
(@Abbr, @DisNam)";

        cmd.CommandText =
insertQuery;

        // Параметры для вставки
значений в запрос

        Console.WriteLine($"Введи
те значение атрибута {attributes[mode - 1][0]}:");

cmd.Parameters.AddWithValue("Abbr",
Console.ReadLine());

        Console.WriteLine($
"Введите значение атрибута {attributes[mode - 1]
[1]}:");

```

```

        cmd.Parameters.AddWithValueVa
lue("DisNam", Console.ReadLine());
    }
    if (mode == 4)
    {
        //Вносим данные
        string insertQuery =
"INSERT INTO Exam (Abbr, GrNum,ExDat)
VALUES (@Abbr, @GrNum, @ExDat)";

        cmd.CommandText =
insertQuery;

        // Параметры для вставки
значений в запрос

        Console.WriteLine($"Введи
те значение атрибута {attributes[mode - 1][0]}:");

        cmd.Parameters.AddWithValueValue("Abbr",
Console.ReadLine());

        Console.WriteLine($
"Введите значение атрибута {attributes[mode - 1]
[1]}:");

        cmd.Parameters.AddWithVa
lue("GrNum", Console.ReadLine());

        Console.WriteLine($
"Введите значение атрибута {attributes[mode - 1]
[2]}:");

        cmd.Parameters.AddWithVa
lue("ExDat",
Convert.ToDateTime(Console.ReadLine()));
    }
    if (mode == 5)
    {
        //Вносим данные
        string insertQuery =
"INSERT INTO BALL (StNum, Abbr,Dat,Ball)
VALUES (@StNum, @Abbr, @Dat, @Ball)";

        cmd.CommandText =
insertQuery;

```

```

        // Параметры для вставки
значений в запрос

        Console.WriteLine($"Введи
те значение атрибута {attributes[mode - 1][0]}:");

        cmd.Parameters.AddWithValueValue("StNum",
Console.ReadLine());

        Console.WriteLine($
"Введите значение атрибута {attributes[mode - 1]
[1]}:");

        cmd.Parameters.AddWithVa
lue("Abbr", Console.ReadLine());

        Console.WriteLine($
"Введите значение атрибута {attributes[mode - 1]
[2]}:");

        cmd.Parameters.AddWithVa
lue("Dat",
Convert.ToDateTime(Console.ReadLine()));

        Console.WriteLine($
"Введите значение атрибута {attributes[mode - 1]
[3]}:");

        cmd.Parameters.AddWithVa
lue("Ball", Convert.ToInt32(Console.ReadLine()));
    }

    // Выполнение запроса на
вставку данных

    int rowsAffected =
cmd.ExecuteNonQuery();

    if (rowsAffected > 0)
    {
        Console.WriteLine("Данные
успешно добавлены в таблицу.");
    }
    else
    {
        Console.WriteLine("Не
удалось добавить данные в таблицу.");
    }
    break;

```

```

        case 2:
            string[] grNum= new string[4];
            int[] countDolgGroup= new
int[4];

            string selectGrNumQuery =
"SELECT stud_group.grnum,count(*) " +
                                "from
stud_group " +
                                "join stud on
stud_group.grnum = stud.grnum " +
                                "join exam on
stud_group.grnum = exam.grnum " +
                                "left join ball
on exam.abbr = ball.abbr and stud.stnum =
ball.stnum " +
                                "join est on
ball.ball = est.ball " +
                                "where
ball.ball = 0 or ball.ball = 2 " +
                                "group by
stud_group.grnum;";

            cmd = new
NpgsqlCommand(selectGrNumQuery, connection);
            NpgsqlDataReader reader =
cmd.ExecuteReader();

            for(int i=0; reader.Read();i++)
            {
                grNum[i] =
reader.GetString(0);
                countDolgGroup[i] =
reader.GetInt32(1);
            }
            reader.Close();

            int countDolgSumm =
countDolgGroup[0] + countDolgGroup[1] +
countDolgGroup[2] + countDolgGroup[3];

            string[,] Report =new
string[countDolgSumm, 4];

```

```

            string selectReportQuery =
"SELECT stud_group.grnum,stud.stnam,
exam.abbr,est.wrd " +
                                "from
stud_group " +
                                "join stud on
stud_group.grnum = stud.grnum " +
                                "join exam on
stud_group.grnum = exam.grnum " +
                                "left join ball
on exam.abbr = ball.abbr and stud.stnum =
ball.stnum " +
                                "join est on
ball.ball = est.ball " +
                                "where
ball.ball = 0 or ball.ball = 2 " +
                                "group by
stud_group.grnum,stud.stnam, exam.abbr,est.wrd;";

            var cmdR =
connection.CreateCommand();

            cmdR = new
NpgsqlCommand(selectReportQuery, connection);
            NpgsqlDataReader readerR =
cmdR.ExecuteReader();

            for (int i = 0;readerR.Read(); i+
+)
            {
                Report[i, 0] =
readerR.GetString(0);
                Report[i, 1] =
readerR.GetString(1);
                Report[i, 2] =
readerR.GetString(2);
                Report[i, 3] =
readerR.GetString(3);
                if(Report[i, 3] == "?/?")
                {
                    Report[i, 3] = "H/A";
                }
                if (Report[i, 3] != "?/?")

```



```

        {
            Report[i, 3] =
"НЕУДОВЛ";
        }
    }

    int buf = 0;
    Console.WriteLine("-----
-----
Отчет-----");
    for(int i = 0; i < 4; i++)
    {
        Console.WriteLine("Гр. {0}
Задолжники
{1}", grNum[i], countDolgGroup[i]);
        Console.WriteLine("-----
-----");
    }

    Console.WriteLine("
ФИО          Дисциплина          Оценка
");
    for(int j=buf;
j<countDolgSumm && grNum[i] == Report[j, 0]; j++
+)
    {
        Console.WriteLine(" {0}
{1}          {2}", Report[j, 1], Report[j, 2], Report[j,
3]);
    }

```

```

        buf = j;
    }
    buf++;

    Console.WriteLine("-----
-----");
    Console.WriteLine();
    Console.WriteLine();
}
break;

case 3:
    flag = false;
    break;
}

// Закрытие подключения
connection.Close();
Console.WriteLine("Соединение с
PostgreSQL закрыто.");
}
}
}

```

4 Описание процесса реализации

Для начала подключаем библиотеку Npgsql. После указываем свои данные для подключения к БД. Далее выполняем подключение к БД и переходим к основному блоку программы, где нас спрашивают какое действие мы хотим выполнить (рисунок 4.1).

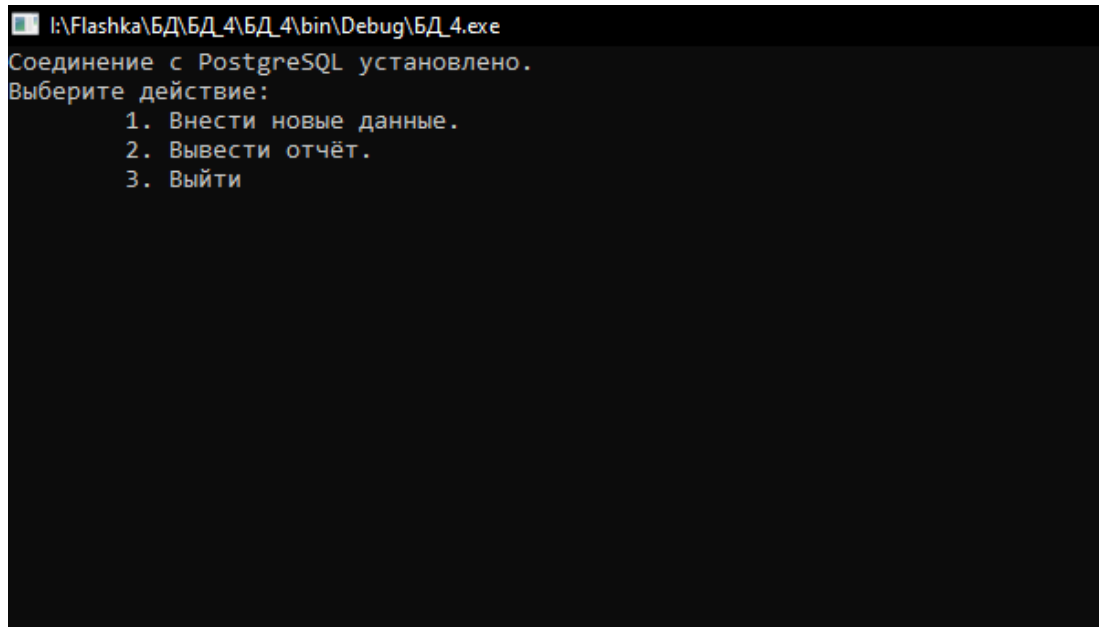


Рисунок 4.1 – Соединение с PostgreSQL и меню выбора действий

Если мы выберем “Внести новые данные”, то программа предложит таблицы на выбор (рисунок 4.2).

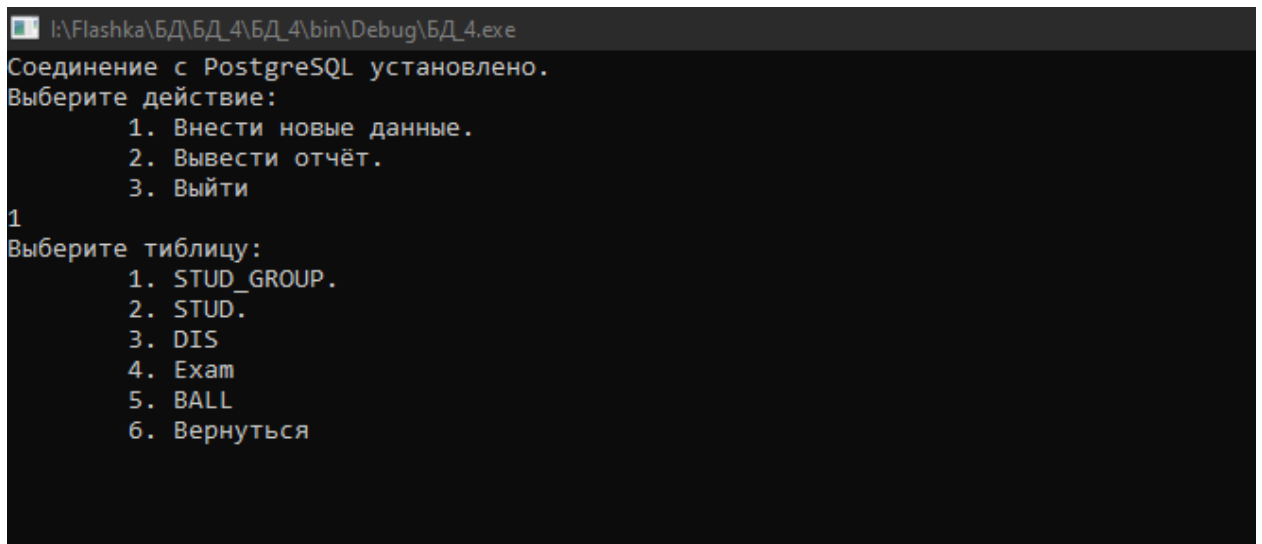


Рисунок 4.2 – Меню выбора таблицы

После выбора таблицы вносим данные в выбранную таблицу (рисунок 4.3).

```
I:\Flashka\БД\БД_4\БД_4\bin\Debug\БД_4.exe
Соединение с PostgreSQL установлено.
Выберите действие:
    1. Внести новые данные.
    2. Вывести отчёт.
    3. Выйти
1
Выберите таблицу:
    1. STUD_GROUP.
    2. STUD.
    3. DIS
    4. Exam
    5. BALL
    6. Вернуться
1
Введите значение атрибута GrNum:
431-6
Введите значение атрибута Cours:
4
Введите значение атрибута Qt:
18
Данные успешно добавлены в таблицу.
Выберите действие:
    1. Внести новые данные.
    2. Вывести отчёт.
    3. Выйти
```

Рисунок 4.3 – Ввод данных в таблицу

Данный метод реализуется с помощью команд и вноса туда INSERT с вводом в него наших переменных.

Если мы выберем “Вывести отчёт”, то программа сразу выведет отчёт на экран (рисунок 4.4).

-----Отчет-----		
Гр. 431-1	Задолжники	3
ФИО	Дисциплина	Оценка
Игнатьев Игорь Егоровна	СиТ	НЕУДОВЛ
Морозов Марк Фёдорович	СиТ	НЕУДОВЛ
Смирнова Юлия Никитична	СиТ	НЕУДОВЛ
Гр. 431-2	Задолжники	1
ФИО	Дисциплина	Оценка
Орлова Антонина Максимовна	БД	НЕУДОВЛ
Гр. 431-3	Задолжники	3
ФИО	Дисциплина	Оценка
Андреев Дмитрий Павлович	ОС	НЕУДОВЛ
Озерова Елена Ивановна	СиТ	НЕУДОВЛ
Родионовна Полина Дмитриевна	СиТ	НЕУДОВЛ
Гр. 431-4	Задолжники	2
ФИО	Дисциплина	Оценка
Соколов Семён Данилович	СВС	НЕУДОВЛ
Фролов Елезавета Марковна	СВС	НЕУДОВЛ

Рисунок 4.4 – Отчёт об задолжниках

Данный метод реализуется так же с помощью команд, но туда мы вносим SELECT. Сначала запрос на группы с должниками и количество должников в каждой группе. После чего мы отправляем запрос уже на информацию о самих должниках.

5 Выводы

В ходе лабораторной работы были получены навыки работы с базами данных при разработке программ и получены навыки работы с библиотекой Npgsql.